

การบีบอัดภาพของกล้องจุลทรรศน์อิเล็กตรอน  
โดยใช้การแปลงเวฟเล็ตแบบดิสครีท

นาย ธีรวัฒน์ ทัดพิทักษ์กุล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมไฟฟ้า  
มหาวิทยาลัยเทคโนโลยีสุรนารี  
ปีการศึกษา 2544  
ISBN 974-7988-74-7

**EM Image Compression**  
**Using the Discrete Wavelet Transform**

**Mr. Nattanun Thatphithakkul**

**A Thesis Submitted in Partial Fulfillment of the Requirements**  
**for the Degree of Master of Engineering Program in Electrical Engineering**

**Suranaree University of Technology**

**Academic Year 2001**

**ISBN 974-7988-74-7**

หัวข้อวิทยานิพนธ์

การบีบอัดข้อมูลภาพของกล้องจุลทรรศน์อิเล็กตรอน

โดยใช้การแปลงเวฟเลตแบบดิสครีต

(EM Image Compression Using the Discrete Wavelet Transform)

สภามหาวิทยาลัย มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วน  
หนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

.....

(รองศาสตราจารย์ ดร.สราวุฒิ สุจิตจร)

ประธานกรรมการ

.....

(อาจารย์ ดร.กิตติ อรรถกิจมงคล)

อาจารย์ที่ปรึกษา/กรรมการ

.....

(ผู้ช่วยศาสตราจารย์ ดร.ยุวดี มานะเกษม)

กรรมการ

.....

(รองศาสตราจารย์ ดร.เกษม ปราบริบูรณ์)

รองอธิการบดีฝ่ายวิจัยและบริการวิชาการ

รักษาการแทนรองอธิการบดีฝ่ายวิชาการ

.....

(รองศาสตราจารย์ น.อ.ดร.วรพจน์ ขำพิศ)

คณบดี สำนักวิชาวิศวกรรมศาสตร์

ณัฐนันท์ ทัดพิทักษ์กุล : การบีบอัดภาพของกล้องจุลทรรศน์อิเล็กตรอนโดยใช้การแปลงเวฟเลตแบบดิสครีท

(EM Image Compression Using the Discrete Wavelet Transform)

อ.ที่ปรึกษา : ดร. กิตติ อรรถกัจจวมงคล, 199 หน้า.

ISBN 974-7988-74-7

งานวิจัยนี้ ได้ดำเนินการบีบอัดข้อมูลภาพจากกล้องจุลทรรศน์อิเล็กตรอน (Electron Microscope, EM) ด้วยวิธีการแปลงเวฟเลตแบบดิสครีท โดยภาพ EM เมื่อนำมาแปลงด้วยวิธีเวฟเลตแบบดิสครีท ข้อมูลที่เป็นพื้นผิววัตถุจะอยู่ที่สัมประสิทธิ์ที่แบนด์ย่อยความถี่ต่ำ และข้อมูลที่เป็นขอบหรือลายเส้นของวัตถุจะอยู่ที่สัมประสิทธิ์ที่แบนด์ย่อยความถี่สูง ดังนั้นการบีบอัดข้อมูลภาพ EM ด้วยการแปลงเวฟเลตจะต้องให้ความสำคัญกับสัมประสิทธิ์ทุกแบนด์ย่อย อัลกอริทึม Set Partitioning in Hierarchical Tree (SPIHT) เป็นอัลกอริทึมที่เข้ารหัส โดยให้ความสำคัญกับขนาดของสัมประสิทธิ์และไม่สนใจว่าสัมประสิทธิ์ตัวนั้นจะอยู่ในระดับแบนด์ย่อยใด วิทยานิพนธ์นี้ได้เสนอวิธีการพัฒนาอัลกอริทึม SPIHT ให้สามารถบีบอัดข้อมูลภาพได้เพิ่มขึ้น โดยการเพิ่ม List of Forbidden Coefficients (LFC) และเพิ่มเงื่อนไขการเข้ารหัสและการถอดรหัส พร้อมทั้งหาเวฟเลตแม่ที่เหมาะสมกับอัลกอริทึมที่ทำการพัฒนา และทำการหาข้อสรุปในการหาระดับอัตราบิตที่เหมาะสมของการบีบอัดข้อมูลภาพ EM จากการประเมินผลด้วยแบบสอบถาม

ผลการบีบอัดภาพ EM ด้วยการแปลงเวฟเลตแบบดิสครีท ที่ใช้  $bi9-7$  เป็นเวฟเลตแม่ในการแปลงเวฟเลต แล้วมาเข้ารหัสด้วยอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง จะให้ผลการบีบอัดข้อมูลภาพ EM ที่ดีที่สุด และระดับอัตราบิตที่เหมาะสมสำหรับการบีบอัดภาพ EM คือ ภาพที่มีรายละเอียดภาพต่ำจะมีอัตราบิตเท่ากับ 0.25 bpp และภาพที่มีรายละเอียดสูงจะมีอัตราบิตเท่ากับ 1 bpp ซึ่งภาพที่อัตราบิตดังกล่าวจะเป็นภาพที่พอจะนำมาใช้ทดแทนภาพต้นแบบได้ และการนำข้อมูลที่ผ่านการบีบอัดด้วยอัลกอริทึม SPIHT ที่ผ่านการปรับปรุงมาเข้ารหัสเลขคณิต จะทำให้ข้อมูลมีขนาดเล็กลง

สาขาวิชาวิศวกรรมไฟฟ้า

ปีการศึกษา 2544

ลายมือชื่อนักศึกษา \_\_\_\_\_

ลายมือชื่ออาจารย์ที่ปรึกษา \_\_\_\_\_

NATTANUN THATPHITHAKKUL : EM IMAGE COMPRESSION USING THE  
DISCRETE WAVELET TRANSFORM

THESIS ADVISOR : KITTI ATTAKITMONGCOL, Ph.D. 199 pp

ISBN 974-7988-74-7

This thesis presents the EM image compression using the discrete wavelet transform. When the discrete wavelet transform is applied to an EM image, plain surfaces of the object will be in the low-frequency subband and edges will be in the high-frequency subbands. Thus, compression algorithm for the EM image must take every subband of wavelet coefficients into account. A powerful image compression algorithm we consider is the Set Partitioning in Hierarchical Tree (SPIHT). This coding algorithm exploits the self-similarity of the wavelet coefficients across different scales and searches for the high magnitude coefficients in every subband. This thesis also proposes an improvement of the SPIHT algorithm to increase compression capability by adding the List of Forbidden Coefficients (LFC) and extending the encoding and decoding conditions. Furthermore, we search for the best wavelet basis for the EM image and evaluate the performance of the improved algorithm to find the suitable bit rate. The evaluation is based on the survey of scientists in Thailand.

Numerical results show that the bi9-7 wavelet gives the best quality of the decompressed EM images at the same bit rate using the improved algorithm. From the survey, we find that images which have many details can be effectively compressed at bit rate of 1 bpp while the images with less details can be compressed at bit rate of 0.25 bpp. In addition, we implement the arithmetic coding with the bit stream obtained from the improved algorithm. This makes the compression algorithm more efficient.

สาขาวิชาวิศวกรรมไฟฟ้า  
ปีการศึกษา 2544

ลายมือชื่อนักศึกษา \_\_\_\_\_  
ลายมือชื่ออาจารย์ที่ปรึกษา \_\_\_\_\_

## กิตติกรรมประกาศ

ผู้วิจัยขอขอบพระคุณ บุพการี ที่อบรม สั่งสอน และเลี้ยงดู รวมทั้งให้การสนับสนุนอย่างดี มาโดยตลอด

ขอขอบพระคุณ ดร. กิตติ อรรถกิจมงคล อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้คำปรึกษาแนะนำ ให้ความรู้อันเป็นประโยชน์ต่อการทำวิทยานิพนธ์เป็นอย่างมาก และสละเวลาตรวจสอบ ทำให้วิทยานิพนธ์มีความถูกต้องสมบูรณ์มากยิ่งขึ้น

ขอขอบพระคุณ รองศาสตราจารย์ ดร. สราวุธ สุจิตจร ที่ให้คำปรึกษาแนะนำในการนำภาพ EM มาใช้เป็นข้อมูลในการบีบอัดข้อมูล ช่วยสนับสนุนในการจัดทำและจัดส่งแบบประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลไฟล์ภาพ ทำให้ได้ผลงานวิจัยที่ดียิ่งขึ้น

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. ยุวดี มานะเกษม ที่ให้คำปรึกษาแนะนำในการทำแบบประเมิน และช่วยทำการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลไฟล์ภาพ

ขอขอบคุณ รองศาสตราจารย์ วิรุฬห์ มังกละวิรัช, ดร. วีระยุทธ ลอประยูร, รองศาสตราจารย์ ดร. ประพีร์ เศรษฐรักษ์, รองศาสตราจารย์ ดร. ชัยทิพย์ วนิชานนท์, ดร. วิศิษฐ์ ทวีปรั้งยีพร, รองศาสตราจารย์ เกลียวพันธ์ เถกิงพล, ฝ่ายเครื่องมือวิทยาศาสตร์ฯ สถาบันวิจัยและพัฒนาฯ มหาวิทยาลัยเกษตรศาสตร์, ฝ่ายปฏิบัติการวิจัยฯ สถาบันวิจัยและพัฒนาฯ มหาวิทยาลัยเกษตรศาสตร์, คุณ กรวรรณ รัตนไชย และคุณ ประสิทธิ์ งามสันเทียะ ที่ช่วยทำการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลไฟล์ภาพ

ขอขอบคุณ คุณ อนุชิต เรื่องวิทยานนท์ สำหรับการจัดหาข้อมูลภาพ EM ที่นำมาใช้ในการทดสอบการบีบอัดข้อมูลภาพ และช่วยทำการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลไฟล์ภาพ

นอกจากนี้ผู้เขียนขอขอบคุณ เพื่อนๆ พี่ๆ ร่วมเรียนระดับปริญญาโท และพี่ๆ ที่ศูนย์เครื่องมือฯ สำหรับกำลังใจและให้ความช่วยเหลือทุกอย่างแก่ผู้เขียน จนกระทั่งวิทยานิพนธ์ฉบับนี้เสร็จโดยสมบูรณ์

นาย ณัฐนันท์ ทัดพิทักษ์กุล

4 เมษายน 2545

## สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย).....	ก
บทคัดย่อ (ภาษาอังกฤษ).....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญภาพ.....	ญ

### บทที่

<b>1 บทนำ.....</b>	<b>1</b>
1.1 ความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์การวิจัย.....	2
1.3 ข้อตกลงเบื้องต้น.....	2
1.4 ขอบเขตของการวิจัย.....	2
1.5 ขั้นตอนการดำเนินงาน.....	2
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.7 การจัดรูปเล่มวิทยานิพนธ์.....	3
<b>2 กล้องจุลทรรศน์อิเล็กตรอน.....</b>	<b>5</b>
2.1 บทนำ.....	5
2.2 ความเป็นมาของกล้องจุลทรรศน์.....	5
2.3 หลักการเบื้องต้นของกล้องจุลทรรศน์อิเล็กตรอน.....	6
2.4 โครงสร้างของกล้องจุลทรรศน์อิเล็กตรอน.....	8
2.5 หลักการทำงานของกล้องจุลทรรศน์อิเล็กตรอน.....	8
2.6 ระบบภาพของกล้องจุลทรรศน์อิเล็กตรอน.....	11
2.6.1 กระบวนการทางภาพของกล้องจุลทรรศน์อิเล็กตรอน	
แบบส่องกราดแบบอนาล็อก.....	12

## สารบัญ (ต่อ)

หน้า

2.6.2	ข้อจำกัดของระบบสร้างภาพแบบอนาล็อก ในกระบวนการวิเคราะห์ภาพ.....	15
2.6.3	กระบวนการทางกายภาพของกล้องจุลทรรศน์อิเล็กตรอน แบบส่องกราดระบบเชิงตัวเลข.....	16
2.7	กล้องจุลทรรศน์อิเล็กตรอนแบบส่องกราดรุ่น JSM-5800 LV.....	16
2.7.1	ความสามารถในการแจกแจง.....	16
2.7.2	ลักษณะการใช้งาน.....	16
2.7.3	ประเภทของชิ้นตัวอย่าง.....	17
2.8	สรุป.....	18
<b>3</b>	<b>หลักการเบื้องต้นในการบีบอัดข้อมูลภาพ.....</b>	<b>19</b>
3.1	บทนำ.....	19
3.2	การบีบอัดข้อมูลภาพแบบที่ไม่มีการสูญเสีย.....	19
3.2.1	การเข้ารหัสแบบรันเลนจซ์.....	19
3.2.2	การเข้ารหัสฮัฟแมน.....	20
3.2.3	การเข้ารหัสเลขคณิต.....	21
3.3	การบีบอัดข้อมูลภาพแบบที่มีการสูญเสีย.....	23
3.3.1	การเข้ารหัสโดยการแปลง.....	23
3.3.2	การเข้ารหัสควอนไทซ์แบบเวกเตอร์.....	25
3.3.3	การเข้ารหัสแบนด์ย่อย.....	26
3.4	การวัดประสิทธิภาพการบีบอัดข้อมูลภาพ.....	27
3.4.1	การวัดเชิงปริมาณ.....	27
3.4.2	การวัดเชิงคุณภาพ.....	28
3.5	สรุป.....	29
<b>4</b>	<b>การแปลงเวฟเล็ต.....</b>	<b>30</b>
4.1	บทนำ.....	30



## สารบัญ (ต่อ)

หน้า

4.2	ความเป็นมาของการแปลงเวฟเล็ต.....	30
4.3	ทฤษฎีพื้นฐานของการแปลงเวฟเล็ต.....	33
4.3.1	ลักษณะของเวฟเล็ต.....	33
4.3.2	ทฤษฎีพื้นฐานเวฟเล็ตในการสเกลและการเลื่อนตำแหน่ง.....	34
4.4	การแปลงเวฟเล็ตแบบดิสครีทใน 1 มิติ.....	41
4.5	การแปลงเวฟเล็ตแบบดิสครีทใน 2 มิติ.....	45
4.6	สรุป.....	50
5	<b>การบีบอัดข้อมูลภาพด้วยอัลกอริทึม Set Partitioning in Hierarchical Tree (SPIHT).....</b>	<b>51</b>
5.1	บทนำ.....	51
5.2	อัลกอริทึม SPIHT.....	51
5.2.1	ความสัมพันธ์แบบ Spatial Orientation Trees.....	53
5.2.2	อัลกอริทึมการเข้ารหัส SPIHT.....	54
5.2.3	การจัดเก็บเป็นไฟล์การบีบอัดข้อมูล.....	56
5.2.4	อัลกอริทึมการถอดรหัส SPIHT.....	56
5.3	การพัฒนาอัลกอริทึม SPIHT.....	57
5.3.1	อัลกอริทึมการเข้ารหัสอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง.....	59
5.3.2	การจัดเก็บเป็นไฟล์การบีบอัดข้อมูลอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง.....	62
5.3.3	อัลกอริทึมการถอดรหัสอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง.....	62
5.4	การเข้ารหัสเลขคณิต.....	62
5.5	สรุป.....	68
6	<b>ผลการทดลอง.....</b>	<b>70</b>
6.1	บทนำ.....	70

## สารบัญ (ต่อ)

	หน้า
6.2 การหาตระกูลเวฟเล็ดแม่.....	70
6.2.1 ผลการทดสอบหาตระกูลเวฟเล็ดแม่.....	73
6.3 การบีบอัดข้อมูลภาพด้วยอัลกอริทึม SPIHT และอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง.....	74
6.3.1 ผลการทดสอบการบีบอัดข้อมูลภาพด้วยอัลกอริทึม SPIHT และอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง.....	75
6.4 การบีบอัดข้อมูลภาพด้วยอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง และการเข้ารหัสเลขคณิต.....	77
6.4.1 ผลการทดสอบการบีบอัดข้อมูลภาพด้วยอัลกอริทึม SPIHT ที่ผ่านการปรับปรุงและการเข้ารหัสเลขคณิต.....	77
6.5 การหาข้อสรุปในการหาระดับอัตราบิต ในการบีบอัดข้อมูลภาพ EM.....	79
6.5.1 ผลการหาข้อสรุปในการหาระดับอัตราบิต ในการบีบอัดข้อมูลภาพ EM.....	80
6.6 สรุป.....	88
<b>7 สรุปและข้อเสนอแนะ.....</b>	<b>89</b>
7.1 บทนำ.....	89
7.2 สรุป.....	89
7.3 ข้อเสนอแนะ.....	90
เอกสารอ้างอิง.....	91
ภาคผนวก	
ภาคผนวก ก แบบสอบถามเพื่อประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลไฟล์ภาพ สำหรับการแสดงผลของกล้องจุลทรรศน์อิเล็กตรอน.....	93

## สารบัญ (ต่อ)

หน้า

ภาคผนวก ข ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลไฟล์ภาพ.....	126
ภาคผนวก ค รายละเอียดโปรแกรมการเข้ารหัสอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง.....	144
ภาคผนวก ง รายละเอียดโปรแกรมการถอดรหัสอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง.....	162
ภาคผนวก จ รายละเอียดโปรแกรมการเข้ารหัสและถอดรหัสเลขคณิต.....	173
ภาคผนวก ฉ รายละเอียดโปรแกรมอ่านและเขียนไฟล์ภาพ ที่ได้จากกล้องจุลทรรศน์อิเล็กตรอน.....	178
ภาคผนวก ช บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในขณะศึกษา.....	181
ประวัติผู้เขียน.....	199

## สารบัญตาราง

ตารางที่	หน้า
5.1	ตัวอย่างสัญลักษณ์และจำนวนของสัญลักษณ์ของข้อมูลที่นำมาเข้ารหัสเลขคณิต.....64
5.2	ขั้นตอนการเข้ารหัสเลขคณิต.....65
5.3	ข้อมูลของสัญลักษณ์ที่ใช้ถอดรหัส.....67
5.4	ขั้นตอนการถอดรหัสเลขคณิต.....68
6.1	แสดงผลการบีบอัดข้อมูลภาพ EM ที่ตระกูลเวฟเล็ตแม่ต่างๆ.....73
6.2	แสดงผลการบีบอัดข้อมูลภาพ EM ด้วยอัลกอริทึม SPIHT และอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง.....76
6.3	แสดงผลการบีบอัดข้อมูลภาพ EM ด้วยอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง และการเข้ารหัสเลขคณิต.....78
6.4	แสดงระดับอัตราบิดที่เหมาะสมในการบีบอัดข้อมูลภาพ EM.....88
ข.1	ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละบุคคล.....128
ข.2	ผลรวมการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละภาพ.....140

## สารบัญภาพ

ภาพที่	หน้า
2.1	แสดงการเปรียบเทียบโครงสร้าง OM กับ EM.....7
2.2	แผนภาพเบื้องต้นของระบบกล้องจุลทรรศน์อิเล็กตรอน.....8
2.3	ภาพตัดขวางของคอลัมน์กล้องจุลทรรศน์แบบ TEM รุ่น JEM - 2010 ของบริษัท JEOL.....10
2.4	ภาพตัดขวางของคอลัมน์กล้องจุลทรรศน์แบบ SEM รุ่น 35CF ของบริษัท JEOL.....11
2.5	แผนภาพการทำงานของกล้องจุลทรรศน์อิเล็กตรอน แบบส่องกราดระบบอนาล็อก.....12
2.6	แผนภาพการทำงานของระบบสร้างภาพ.....13
2.7	ลักษณะของสัญญาณภาพของ SEM.....15
2.8	ภาพตัดขวางของคอลัมน์กล้องจุลทรรศน์แบบ SEM รุ่น JSM – 5800 LV ของบริษัท JEOL.....17
2.9	ภาพของกล้องจุลทรรศน์แบบ SEM รุ่น JSM – 5800 LV ของบริษัท JEOL.....18
3.1	แสดงตัวอย่างรหัสแบบรันเลนจ์.....20
3.2	แสดงตัวอย่างการเข้ารหัสฮัฟแมน.....20
3.3	แสดงการเข้ารหัสเลขคณิต.....22
3.4	แผนภาพแสดงการเรียงลำดับแบบซิกแซก.....25
3.5	แผนภาพการเข้ารหัสเวกเตอร์ควอนไทซ์.....26
3.6	แผนภาพการถอดรหัสเวกเตอร์ควอนไทซ์.....26
4.1	แสดงลักษณะของระนาบเวลาความถี่แบบคงที่.....31
4.2	แสดงลักษณะของระนาบเวลาและความถี่ของ WT.....32
4.3	ตัวอย่างแสดงลักษณะเวฟเล็ต.....33

## สารบัญภาพ (ต่อ)

ภาพที่		หน้า
4.4	ลักษณะของสัญญาณรูปคลื่น โคซายน์.....	33
4.5	แสดงลักษณะของฟังก์ชันหน้าต่าง.....	34
4.6	แสดงลักษณะของเวฟเล็ทแม่ที่สเกลและเลื่อนตำแหน่ง ไปที่ค่า a และ b ต่างๆ.....	35
4.7	แสดงการแยกองค์ประกอบของเวฟเล็ทและการสร้างกลับ จากองค์ประกอบด้วยเวฟเล็ท.....	36
4.8	แสดงลักษณะของสัญญาณที่มีหลายความถี่ที่ตำแหน่งเวลาต่างกัน.....	36
4.9	ตระกูลเวฟเล็ทแม่.....	40
4.10	แผนภาพการวิเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์ แบบ 1 มิติ.....	41
4.11	แผนภาพการลดข้อมูลลงครึ่งหนึ่ง.....	41
4.12	แผนภาพการสังเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์ แบบ 1 มิติ.....	42
4.13	แผนภาพการเพิ่มข้อมูลขึ้นเป็นสองเท่า.....	42
4.14	แผนภาพการวิเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์ แบบ 3 แบนด์ย่อย.....	43
4.15	แผนภาพการสังเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์ แบบ 3 แบนด์ย่อย.....	43
4.16	ตัวอย่างการวิเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์แบบ 3 แบนด์ย่อย.....	44
4.17	ตัวอย่างการสังเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์แบบ 3 แบนด์ย่อย.....	44
4.18	แผนภาพการแยกองค์ประกอบด้วยเวฟเล็ท.....	45
4.19	แผนภาพการสร้างกลับจากองค์ประกอบด้วยเวฟเล็ท.....	46
4.20	แสดงภาพแบนด์ที่ผ่านการแปลงเวฟเล็ทหลายครั้ง.....	47
4.21	ตัวอย่างภาพที่ผ่านการแปลงเวฟเล็ทด้วยวิธี การแยกองค์ประกอบด้วยเวฟเล็ท.....	49
5.1	แผนภาพแสดงขั้นตอนการเข้ารหัสและขั้นตอนการถอดรหัส.....	52
5.2	แผนภาพแสดงความสัมพันธ์แบบ Spatial Orientation Trees.....	53
5.3	แผนภาพกำหนดตำแหน่งของการเข้ารหัส.....	54
5.4	แผนภาพแสดงการเข้ารหัสขั้นตอนที่ 2 ของอัลกอริทึม SPIHT.....	57

## สารบัญภาพ (ต่อ)

ภาพที่	หน้า
5.5	แผนภาพแสดงการแบ่งแบนด์ย่อยและกำหนด Label ของ LFC.....58
5.6	แผนภาพแสดงการเข้ารหัสขั้นตอนที่ 2 ของอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง.....61
6.1	ภาพต้นแบบ (pollen of zucchini).....72
6.2	ภาพต้นแบบที่ใช้ทดสอบการบีบอัดข้อมูล.....75
6.3	แผนภาพแสดงขั้นตอนการทดสอบการเข้ารหัสเลขคณิต.....77
6.4	แผนภูมิแสดงผลการประเมินของภาพ ant.....81
6.5	แผนภูมิแสดงผลการประเมินของภาพ sand.....82
6.6	แผนภูมิแสดงผลการประเมินของภาพ LED.....83
6.7	แผนภูมิแสดงผลการประเมินของภาพ skin.....84
6.8	แผนภูมิแสดงผลการประเมินของภาพ pollen of zucchini .....85
6.9	แผนภูมิแสดงผลการประเมินของภาพรายละเอียดต่ำ.....86
6.10	แผนภูมิแสดงผลการประเมินของภาพรายละเอียดสูง.....87

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันการศึกษาค้นคว้าความจริงของธรรมชาติในโลกขนาดจิ๋ว ให้เห็นพื้นผิวลักษณะความแตกต่างของโครงสร้างนั้นๆ ที่ไม่สามารถมองเห็นได้ด้วยตาเปล่า เช่น ขนาดของเกรน (grain) การกระจายของอนุภาค ความพรุน ความหนาแน่น หรือร่องรอยอันเกิดจากความผิดปกติของผิววัสดุ สามารถทำได้อย่างมีประสิทธิภาพ โดยอาศัยภาพไมโครกราฟที่ได้จากกล้องจุลทรรศน์อิเล็กตรอน (electron microscope: EM) ภาพไมโครกราฟที่ได้จากเครื่อง EM มีอยู่ด้วยกัน 2 ประเภทคือ ภาพที่ได้จากการบันทึกด้วยระบบถ่ายภาพบนฟิล์มหรือกล้องโพลารอยด์ (polaroid) ภาพที่ได้นั้นมักไม่สม่ำเสมอ เนื่องจากสายตาไม่สามารถแยกความเปรียบต่าง (contrast) ของภาพรวมถึงความแตกต่างของรูปลักษณะตามบริเวณต่างๆ ได้ชัดเจน จึงทำให้เกิดความคลาดเคลื่อนในการวิเคราะห์สูง ส่วนอีกประเภทหนึ่งคือภาพที่ได้จากการเปลี่ยนสัญญาณภาพจากระบบสแกนความเร็วต่ำ (low speed scan) ของ EM ที่อยู่ในรูปของสัญญาณอนาล็อก (analog signal) จะถูกแปลงเป็นสัญญาณเชิงตัวเลข (digital signal) และบันทึกไว้ในหน่วยความจำ ซึ่งภาพนี้ยังสามารถนำไปผ่านกระบวนการปรับแต่งรายละเอียดภาพด้วยวิธีการกรองสิ่งรบกวน และผ่านกระบวนการชดเชยสัญญาณให้มีความคมชัดของภาพดีขึ้น จึงทำให้ภาพไมโครกราฟในระบบเชิงตัวเลขมีความคมชัด และมีความเชื่อมั่นในผลการวิเคราะห์ได้สูงขึ้น (เดโช ทองอร่าม และ สุวิทย์ ปุณณชัย-ยะ, 2538)

งานวิจัยต่างๆ ที่ใช้เครื่อง EM จำเป็นต้องมีการบันทึกภาพ เพื่อใช้เป็นข้อมูลในการวิเคราะห์รายละเอียดของงาน สำหรับงานวิจัยบางอย่างต้องใช้ระยะเวลายาวนานในการทำงานนั้น จำเป็นจะต้องใช้เนื้อที่ในการบันทึกข้อมูลสูง ซึ่งทำให้สิ้นเปลืองค่าใช้จ่ายเพิ่มมากขึ้นไปด้วย ดังนั้นในงานวิจัยนี้จึงนำเสนออัลกอริทึมการบีบอัดข้อมูลภาพ ที่เหมาะสมกับภาพไมโครกราฟในระบบเชิงตัวเลข ที่ได้จากเครื่อง EM



## 1.2 วัตถุประสงค์การวิจัย

1. เพื่อศึกษาและออกแบบอัลกอริทึมการบีบอัดข้อมูลภาพจากเครื่อง EM โดยใช้วิธีการแปลงเวฟเลต (wavelet transform) และการพัฒนาอัลกอริทึม Set Partitioning in Hierarchical Trees (SPIHT) ในการเข้ารหัส และการถอดรหัส
2. เพื่อหาข้อสรุปในการใช้งานที่เหมาะสมของการบีบอัดข้อมูลภาพจากเครื่อง EM โดยใช้วิธีการแปลงเวฟเลต และใช้อัลกอริทึมที่ได้พัฒนา ในการเข้ารหัส และการถอดรหัส
3. เพื่อเป็นแนวทางในการพัฒนาขบวนการบีบอัดข้อมูลภาพ ให้กับผู้ที่สนใจได้ทำการศึกษา และพัฒนาให้ดียิ่งขึ้น

## 1.3 ข้อตกลงเบื้องต้น

1. อัลกอริทึมการบีบอัดข้อมูลภาพ ที่นำมาพัฒนาคือ SPIHT (Amir and William, 1996)
2. ภาพที่ใช้ในการบีบอัดข้อมูลคือ ภาพที่ได้จากเครื่อง EM รุ่น JSM-5800 LV
3. โปรแกรมการบีบอัดข้อมูลภาพ เขียนด้วยโปรแกรม Visual C++
4. โปรแกรมการอ่านและเขียนไฟล์ภาพที่ได้จากเครื่อง EM เขียนด้วยโปรแกรม MATLAB

## 1.4 ขอบเขตของงานวิจัย

1. พัฒนาอัลกอริทึม SPIHT ให้มีประสิทธิภาพการบีบอัดข้อมูลภาพเพิ่มขึ้น
2. พัฒนาโปรแกรมการบีบอัดข้อมูลภาพที่ได้จากเครื่อง EM รุ่น JSM-5800 LV โดยใช้วิธีการแปลงเวฟเลต และใช้อัลกอริทึมที่ได้พัฒนาในการเข้ารหัส และการถอดรหัส
3. การหาข้อสรุปในการหาระดับของการบีบอัดข้อมูลภาพจากเครื่อง EM ที่เหมาะสม จาก การประเมินผลจากแบบสอบถาม

## 1.5 ขั้นตอนการดำเนินงาน

1. ดำเนินการศึกษากการแปลงข้อมูลภาพ ด้วยวิธีการแปลงเวฟเลตแบบดิสครีท (discrete wavelets transform: DWT) แบบ 2 มิติ ใช้หลักการแยกองค์ประกอบเป็นแบนด์ย่อย (subband decomposition)
2. ดำเนินการศึกษากการลดข้อมูลภาพด้วยวิธี SPIHT
3. ดำเนินการพัฒนาอัลกอริทึมการบีบอัดข้อมูลภาพด้วยวิธี SPIHT
4. ดำเนินการศึกษากการเข้ารหัสเลขคณิต และนำมาใช้งานร่วมกับอัลกอริทึม SPIHT ที่ทำ การพัฒนาเพื่อให้การบีบอัดข้อมูลภาพมีประสิทธิภาพเพิ่มขึ้น

5. คำเนินการหาข้อสรุปในการหาระดับที่เหมาะสมกับการบีบอัดข้อมูลภาพจากเครื่อง EM นำมาหาข้อสรุปที่ได้จากการประเมินผลจากแบบสอบถามเพื่อประเมินผลด้วยวิธีวัดความเหมือนระหว่างภาพต้นแบบ กับภาพที่ผ่านการบีบอัดข้อมูล จากกลุ่มเป้าหมายที่ปฏิบัติงานเกี่ยวข้องกับกล้องจุลทรรศน์อิเล็กตรอนคือ อาจารย์ นักวิทยาศาสตร์ นักวิจัย ช่างภาพการแพทย์ เป็นต้น ผลการประเมินจะนำมาวิเคราะห์ทางสถิติ

## 1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้เรียนรู้ทฤษฎีและการประยุกต์ใช้การแปลงเวฟเล็ด ในการบีบอัดข้อมูลภาพ
2. ได้อัลกอริทึมการบีบอัดข้อมูลภาพของเครื่อง EM โดยใช้วิธีการแปลงเวฟเล็ด
3. ได้ระดับการบีบอัดข้อมูลภาพของเครื่อง EM ที่เหมาะสม โดยใช้วิธีการที่ได้นำเสนอ
4. ได้เป็นแนวทางในการพัฒนาขบวนการบีบอัดข้อมูลภาพ ให้กับผู้อื่นได้ทำการศึกษาและพัฒนาให้ดียิ่งขึ้นต่อไปในอนาคต

## 1.7 การจัดรูปเล่มวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ประกอบด้วย 7 บท และ 7 ภาคผนวก

บทที่ 1 เป็นบทนำ กล่าวถึงความสำคัญของปัญหา วัตถุประสงค์ ข้อตกลงเบื้องต้น ขอบเขตของการวิจัย ขั้นตอนการดำเนินงานและประโยชน์ที่คาดว่าจะได้รับจากการวิจัย รวมทั้งเนื้อหาพอลังเขตที่เป็นองค์ประกอบของวิทยานิพนธ์ฉบับนี้

บทที่ 2 กล่าวถึงความเป็นมาของกล้องจุลทรรศน์ หลักการทำงานเบื้องต้นของ EM โครงสร้างของ EM ระบบภาพของ EM และความสามารถในการแจกแจง ลักษณะการใช้งานของ SEM รุ่น JSM – 5800 LV พร้อมทั้งสรุป

บทที่ 3 กล่าวถึงการบีบอัดข้อมูลภาพแบบที่ไม่มีการสูญเสีย การบีบอัดข้อมูลภาพแบบที่มีการสูญเสีย และการวัดประสิทธิภาพการบีบอัดข้อมูลภาพ พร้อมทั้งสรุป

บทที่ 4 กล่าวถึงความเป็นมาของการแปลงเวฟเล็ด ทฤษฎีพื้นฐานของการแปลงเวฟเล็ด การแปลงเวฟเล็ดแบบดิสครีทใน 1 มิติ และการแปลงเวฟเล็ดแบบดิสครีทใน 2 มิติ พร้อมทั้งสรุป

บทที่ 5 กล่าวถึงอัลกอริทึม SPIHT การพัฒนาอัลกอริทึม SPIHT และการเข้ารหัสเลขคณิต พร้อมทั้งสรุป

บทที่ 6 กล่าวถึงการหาตระกูลเวฟเล็ดแม่ การบีบอัดข้อมูลภาพด้วยอัลกอริทึม SPIHT และอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง การบีบอัดข้อมูลภาพด้วยอัลกอริทึม SPIHT ที่ผ่านการ

ปรับปรุงและการเข้ารหัสเลขคณิต และการหาข้อสรุปในการหาระดับอัตราบิต ในการบีบอัดข้อมูลภาพ EM พร้อมทั้งสรุป

บทที่ 7 กล่าวถึงบทสรุป และข้อเสนอแนะ

ภาคผนวก ก. แบบสอบถามเพื่อประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลไฟล์ภาพ สำหรับแสดงผลของกล้องจุลทรรศน์อิเล็กตรอน

ภาคผนวก ข. ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลไฟล์ภาพ

ภาคผนวก ค. รายละเอียดโปรแกรมการเข้ารหัสอัลกอริทึม SPIHT

ภาคผนวก ง. รายละเอียดโปรแกรมการถอดรหัสอัลกอริทึม SPIHT

ภาคผนวก จ. รายละเอียดโปรแกรมการเข้ารหัสและถอดรหัสเลขคณิต

ภาคผนวก ฉ. รายละเอียดโปรแกรมอ่านและเขียนไฟล์ภาพ ที่ได้จากกล้องจุลทรรศน์อิเล็กตรอน

ภาคผนวก ช. บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในขณะศึกษา

## บทที่ 2

### กล้องจุลทรรศน์อิเล็กตรอน

#### 2.1 บทนำ

ปัจจุบันการศึกษาค้นคว้าสิ่งเร้นลับทางชีวภาพและกายภาพที่ลึกลับซับซ้อน ให้เห็นพื้นผิว ลักษณะความแตกต่างภายในโครงสร้างขนาดเล็กในระดับไมโคร ซึ่งไม่สามารถมองเห็นได้ด้วยตาเปล่าได้ ดังนั้นการวิเคราะห์รายละเอียดของโครงสร้างระดับไมโครบนพื้นผิววัสดุ เป็นสิ่งจำเป็นอย่างยิ่งในการควบคุมคุณภาพของงานด้านโลหกรรมและวัสดุศาสตร์ การผลิตสิ่งประดิษฐ์ทางไมโครอิเล็กทรอนิกส์ ที่นับวันยิ่งต้องความละเอียดแม่นยำสูงขึ้น สิ่งหนึ่งที่ใช้ในการวิเคราะห์คือ ภาพไมโครกราฟที่ได้จาก กล้องจุลทรรศน์อิเล็กตรอน (electron microscope: EM)

#### 2.2 ความเป็นมาของกล้องจุลทรรศน์

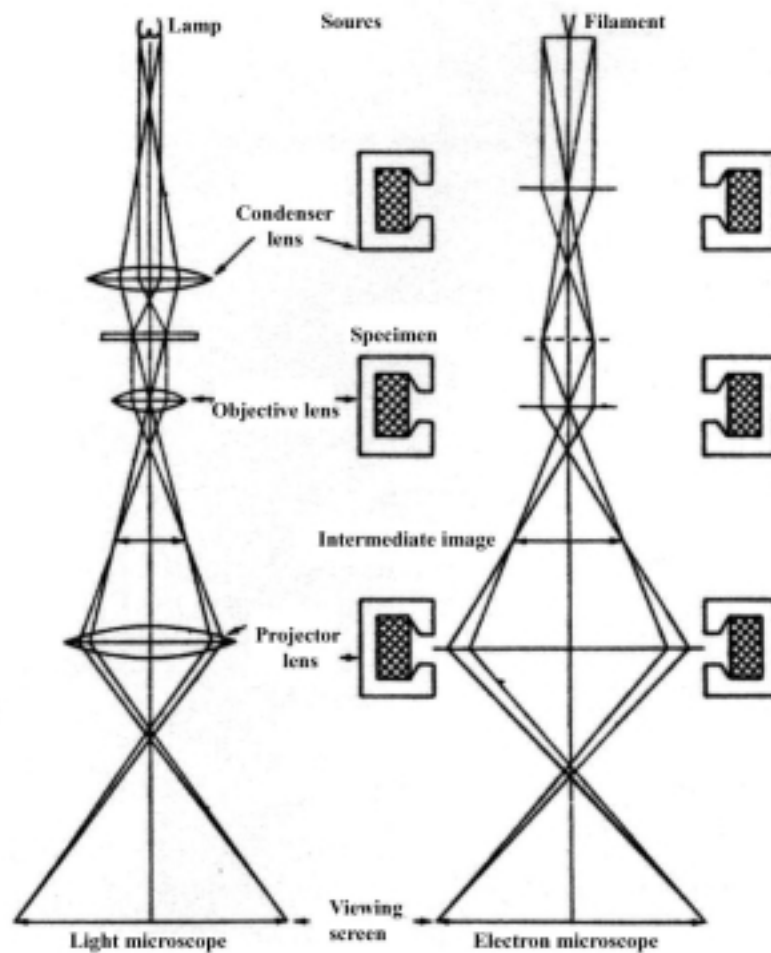
ในระยะแรกของการพัฒนากล้องจุลทรรศน์ให้สามารถมองเห็น โครงสร้างขนาดเล็กด้วยตาเปล่า นั้น นักวิทยาศาสตร์อาศัยแสงสว่างเป็นสื่อกลางของการมองเห็นผ่านระบบกำลังขยายของเลนส์แก้ว (optical lens) เรียกว่ากล้องจุลทรรศน์แบบแสง (optical microscope: OM) กล้องจุลทรรศน์ชนิดนี้ได้รับการออกแบบการทำงานใน 2 แบบคือ แบบแรกใช้เทคนิคการจัดระบบเลนส์แบบลำแสงส่องผ่าน (transmission type) ช่วยให้เห็นโครงสร้างขนาดเล็กภายในแผ่นเนื้อเยื่อและฟิล์มบางของวัตถุ และอีกแบบหนึ่งใช้เทคนิคการจัดระบบเลนส์ลำแสงสะท้อน (reflection type) ซึ่งช่วยให้สามารถมองเห็นโครงสร้างขนาดเล็กบนพื้นผิวของวัสดุ แต่อย่างไรก็ตาม OM นั้น มีขีดจำกัดในด้านกำลังขยายและความสามารถในการแจกแจงโครงสร้างขนาดเล็ก (resolving power) เนื่องจากความยาวคลื่นของแสงที่ส่องกระทบบนโครงสร้างของชิ้นวัตถุตัวอย่าง ในช่วงคลื่นที่ตาเปล่ามองเห็นมีขนาดกว้างกว่ารายละเอียดของโครงสร้างขนาดเล็กที่ต้องการจะศึกษา ด้วยเหตุนี้นักวิทยาศาสตร์จึงคิดพัฒนาวิธีการที่จะเพิ่มความสามารถของกล้องจุลทรรศน์ให้มีความสามารถมากขึ้น โดยการค้นหาสื่อกลางที่ช่วยในการมองเห็นภาพโครงสร้างระดับไมโคร จนกระทั่งปี พ.ศ. 2467 De Broglie ได้พบว่าอิเล็กตรอนที่ถูกเร่งด้วยสนามไฟฟ้าในสุญญากาศให้มีพลังงานสูงขึ้น จะมีพฤติกรรมทางกายภาพคล้ายคลื่นแสงและสามารถบังคับให้คลื่นอิเล็กตรอนมีขนาดความยาวคลื่นสั้นกว่าคลื่นแสงในช่วงที่ตามองเห็นได้ โดยเรียกก้องจุลทรรศน์ที่ได้นี้ว่า กล้องจุลทรรศน์อิเล็กตรอน

ประมาณ ปี พ.ศ. 2474 โดย Ernst Ruska ได้สร้าง EM ขึ้นเป็นครั้งแรก โดยใช้เทคนิคระบบเลนส์แบบลำอิเล็กตรอนส่องผ่าน และเรียก EM ชนิดนี้ว่า กล้องจุลทรรศน์อิเล็กตรอนแบบส่องผ่าน (transmission electron microscope: TEM) ได้มีการพัฒนาให้ EM มีกำลังขยายสูงมากกว่าล้านเท่า และมีขีดความสามารถในการแจกแจงได้เล็กกว่า 2 อังสตรอม (angstrom) และยังได้รับการออกแบบโครงสร้างของ EM ให้สามารถศึกษาโครงสร้างโมเลกุลบนพื้นผิววัสดุด้วยเทคนิคระบบเลนส์แบบลำอิเล็กตรอนสแกน และเรียก EM ชนิดนี้ว่า กล้องจุลทรรศน์อิเล็กตรอนแบบส่องกราด (scanning electron microscope: SEM) นอกจากนี้ EM จะมีขีดความสามารถในการมองเห็นสูงกว่า OM และผลของลำอิเล็กตรอนพลังงานสูงที่ตกกระทบบนชิ้นตัวอย่างยังเหนี่ยวนำให้เกิดปรากฏการณ์ทางกายภาพของชิ้นตัวอย่าง เช่น การเรืองแสงของชิ้นตัวอย่าง ทำให้ระบบของ EM สามารถเพิ่มการวิเคราะห์ธาตุเชิงคุณภาพ และปริมาณควบคู่ไปกับการศึกษาโครงสร้างและผิวตัวอย่างจากภาพไมโครกราฟ จึงทำให้สามารถศึกษารายละเอียดของชิ้นวัตถุได้มากขึ้น ดังนั้นปัจจุบัน EM จึงเป็นเครื่องมือวิจัยหลักที่จำเป็นและมีบทบาทสำคัญต่อการศึกษาวิจัยและพัฒนาเทคโนโลยีขั้นสูงเป็นอย่างมาก (วิรุฬห์ มัลลละวิรัช และ สุวิทย์ ปุณณชัยยะ, 2534)

### 2.3 หลักการเบื้องต้นของกล้องจุลทรรศน์อิเล็กตรอน

การพัฒนา EM เพื่อให้เกิดการมองเห็นโครงสร้างโมเลกุลด้วยตาเปล่าอย่างชัดเจนนั้น มีองค์ประกอบที่สำคัญ 3 ประการคือ ความสามารถในการแจกแจง (resolving power) กำลังขยาย (magnification) และความคมชัดของภาพ องค์ประกอบทั้ง 3 ประการที่กล่าวมานี้ขึ้นกับโครงสร้างของกล้องจุลทรรศน์ตั้งแต่แหล่งกำเนิดของสื่อกลางที่ช่วยให้เกิดการมองเห็น ระบบเลนส์ที่ทำให้เกิดกำลังขยาย และอุปกรณ์ประกอบที่ช่วยลดความผิดเพี้ยนของภาพรวมทั้งความคมชัดไม่ว่าจะเป็น OM หรือ EM ก็ตาม จะมีโครงสร้างพื้นฐานที่คล้ายกันดังแสดงในภาพ 2.1 โดยภาพที่ 2.1 แสดงโครงสร้างของ OM และ EM ที่จัดระบบเลนส์แบบลำแสงและลำอิเล็กตรอนส่องผ่าน จะเห็นว่ามีส่วนประกอบที่ทำให้เกิดการมองเห็นภาพคล้ายคลึงกัน สามารถเทียบหน้าที่การทำงานของส่วนประกอบต่างๆ ได้ แต่ในเรื่องการทำงานของกล้องจุลทรรศน์ทั้งสองชนิดจะมีความแตกต่างกัน เนื่องจากแหล่งกำเนิดของสื่อกลางในการทำให้เกิดภาพของ OM ใช้หลอดไฟฟ้าซึ่งให้ลำแสงผ่านระบบเลนส์แก้วในคอลัมน์ที่มีอากาศได้ ซึ่งการปรับเปลี่ยนกำลังขยายและการปรับความคมชัดอาศัยระบบกลในการควบคุมทั้งหมด ในส่วน EM แหล่งกำเนิดอิเล็กตรอนจะต้องทำงานในคอลัมน์สุญญากาศ โดยจะถูกบังคับด้วยสนามไฟฟ้าศักดาสูงเร่งให้อิเล็กตรอนมีพลังงานสูงและมีพฤติกรรมคล้ายคลื่นแสง ซึ่งการปรับกำลังขยายและความคมชัดอาศัยการบังคับให้ลำอิเล็กตรอนผ่านสนามแม่เหล็กซึ่งทำหน้าที่คล้ายเลนส์ การปรับเปลี่ยนกำลังขยายและความ

คมชัดของระบบเลนส์อิเล็กตรอนเป็นการปรับปริมาณกระแสไฟฟ้าในขดลวดแม่เหล็กไฟฟ้า เพื่อให้เกิดการเปลี่ยนแปลงความหนาแน่นของเส้นแรงแม่เหล็ก การใส่ชิ้นตัวอย่างต้องใส่ในห้องสุญญากาศ รวมทั้งระบบการมองเห็นภาพต้องอาศัยตัวกลางในการเปลี่ยนสัญญาณอิเล็กตรอนที่เกิดจากอันตรกิริยาในโครงสร้างไมโครของชิ้นตัวอย่างให้ปรากฏเป็นภาพที่สามารถมองเห็นด้วยตาเปล่า ด้วยเหตุนี้ EM จึงประกอบด้วยระบบต่างๆ มากมาย เช่น ระบบสุญญากาศ ระบบกำเนิดไฟฟ้าศักดาสูง ระบบควบคุมอิเล็กตรอนิกส์ ระบบเลนส์อิเล็กตรอน เป็นต้น จึงทำให้มีขนาดใหญ่กว่า OM และมีการใช้งานยุ่งยากซับซ้อนกว่า แต่นักวิทยาศาสตร์ทั้งสาขาชีวภาพและกายภาพก็มีความจำเป็นต้องใช้ EM เพราะมีขีดความสามารถเหนือกว่า OM หลายเท่า



ภาพที่ 2.1 แสดงการเปรียบเทียบโครงสร้าง OM กับ EM

(วิรุพห์ มัลลละวิรัช และสุวิทย์ ปุณณชัยยะ, 2534)

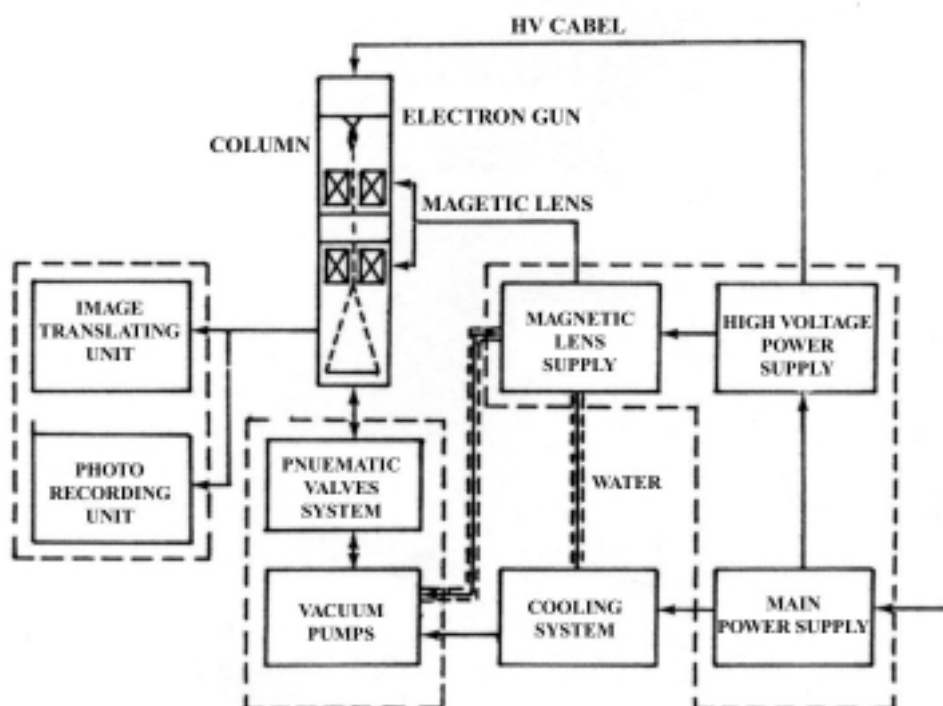
## 2.4 โครงสร้างของกล้องจุลทรรศน์อิเล็กตรอน

EM แบ่งออกเป็น 2 แบบคือ TEM และ SEM กล้องทั้งสองแบบจะมีโครงสร้างภายนอกคล้ายกัน (วิรุพท์ มัลลละวิรัช และ สุวิทย์ ปุณณชัยยะ, 2534) ประกอบด้วยส่วนสำคัญ 5 ส่วนคือ

1. ระบบอิเล็กตรอนออปติก (electron optics system)
2. ระบบสุญญากาศ (vacuum system)
3. ระบบระบายความร้อน (cooling system)
4. ระบบจ่ายกำลังไฟฟ้า (power supply system)
5. ระบบสร้างภาพและบันทึกภาพ (image translation and photo recording)

ส่วนที่แตกต่างจะอยู่ที่โครงสร้างของอิเล็กตรอนออปติกภายในคอลัมน์สุญญากาศ ซึ่งจะกล่าวถึงความแตกต่างในหัวข้อต่อไป

## 2.5 หลักการทำงานของกล้องจุลทรรศน์อิเล็กตรอน

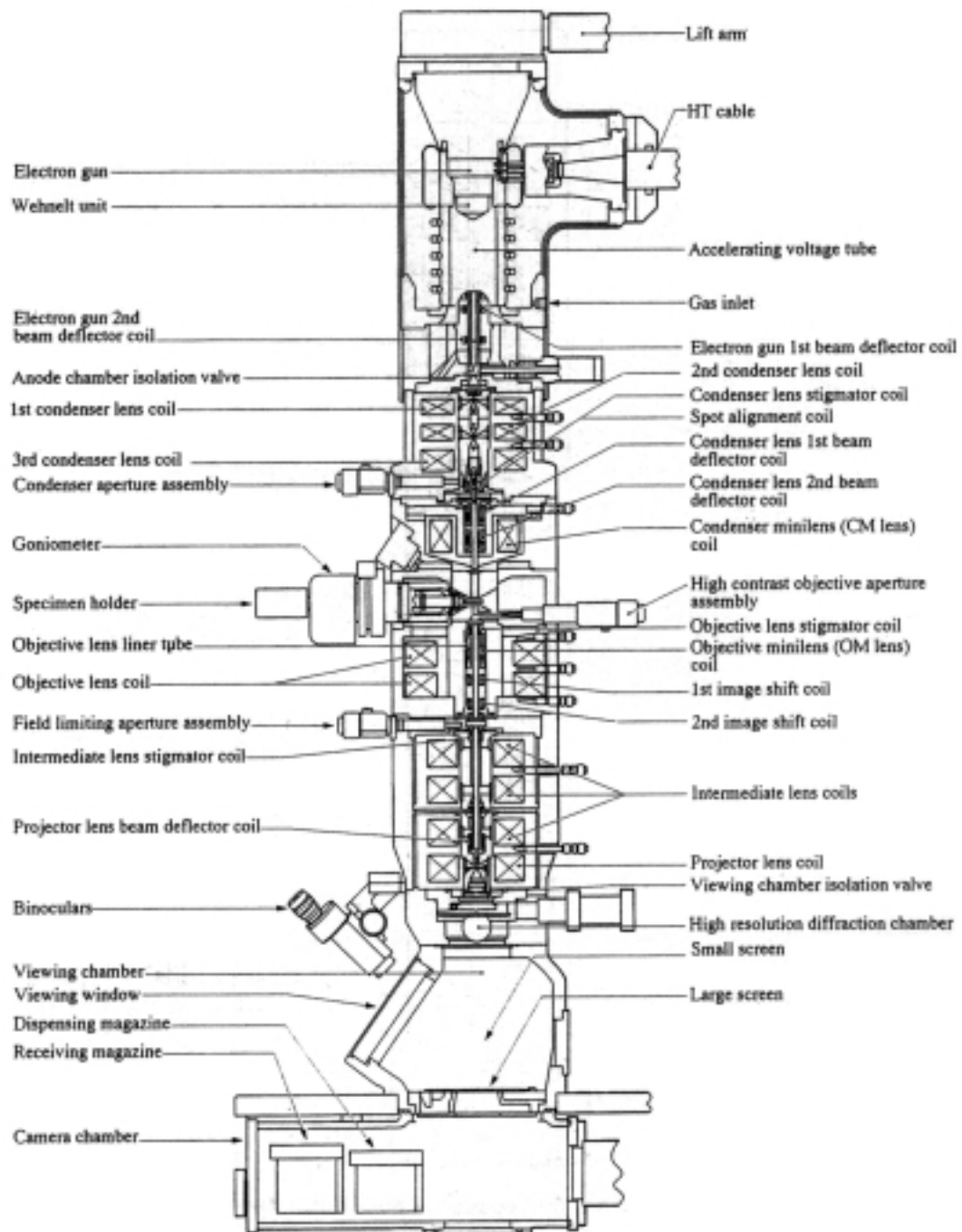


ภาพที่ 2.2 แผนภาพเบื้องต้นของระบบกล้องจุลทรรศน์อิเล็กตรอน  
(วิรุพท์ มัลลละวิรัช และสุวิทย์ ปุณณชัยยะ, 2534)

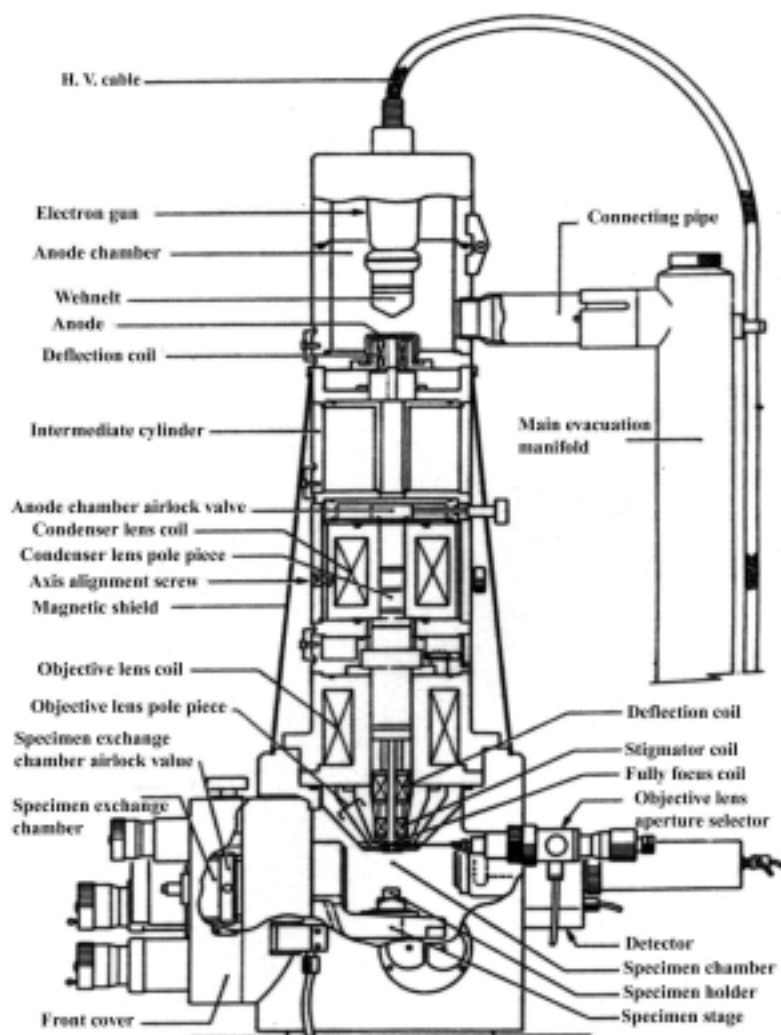
ภาพที่ 2.2 แสดงแผนภาพเบื้องต้นของระบบต่างๆ ภายนอก EM ซึ่งทำงานเชื่อมโยงกันตามลำดับขั้นที่ถูกรอกแบบไว้ กล่าวคือ EM จะทำงานได้นั้น ระบบสุญญากาศจะต้องสร้างภาพสุญญากาศภายในคอลัมน์ซึ่งบรรจุระบบอิเล็กตรอนออปติกก่อน จากนั้นระบบจ่ายกำลังไฟฟ้าหลัก 2 ส่วนคือ แหล่งจ่ายศักดาไฟฟ้าสูงและแหล่งจ่ายไฟฟ้าสำหรับเลนส์แม่เหล็กจึงจะทำงานได้ แหล่งจ่ายศักดาไฟฟ้าสูงจะทำหน้าที่สร้างสนามไฟฟ้าแรงให้กับระบบอิเล็กตรอนออปติกเพื่อสร้างลำอิเล็กตรอนความเร็วสูง และแหล่งจ่ายกระแสไฟฟ้าควบคุมการทำงานของระบบเลนส์อิเล็กตรอนเพื่อสร้างกำลังขยายภาพขึ้นในระบบอิเล็กตรอนออปติกทั้งระบบสุญญากาศและแหล่งจ่ายกระแสไฟฟ้าให้กับเลนส์อิเล็กตรอนจะประกอบด้วยอุปกรณ์ที่เกิดความร้อนสูง ได้แก่ เครื่องสูบอากาศแบบดิวเทอเรียม ทรานซิสเตอร์กำลังสำหรับขับกระแสในขดลวดแม่เหล็ก เป็นต้น ดังนั้นจึงจำเป็นต้องมีระบบระบายความร้อนด้วยน้ำเย็นเพื่อรักษาอุณหภูมิของการทำงานให้อยู่ที่อุณหภูมิห้อง ชิ้นตัวอย่างที่ต้องการจะศึกษาโครงสร้างจะต้องใส่ในห้องสุญญากาศภายในคอลัมน์ที่จัดไว้ตามสภาพโครงสร้างของระบบเลนส์ ภาพสัญญาณอิเล็กตรอนที่เกิดขึ้นในระบบอิเล็กตรอนออปติกต้องอาศัยระบบการสร้างภาพเพื่อให้สามารถมองเห็นภาพขยายนั้นด้วยตาเปล่า และสามารถบันทึกลงบนแผ่นฟิล์มได้

แม้ว่าโครงสร้างภายนอกของ TEM และ SEM จะเหมือนกันก็ตาม แต่เมื่อพิจารณาด้านการจัดระบบเลนส์อิเล็กตรอนแล้ว โครงสร้างอิเล็กตรอนออปติกภายในคอลัมน์สุญญากาศจะแตกต่างกันมากทีเดียว ดังแสดงในภาพตัดขวางของคอลัมน์กล้อง TEM เป็นดังภาพที่ 2.3 และภาพตัดขวางของคอลัมน์กล้อง SEM เป็นดังภาพที่ 2.4 จากภาพที่ 2.3 จะเห็นได้ว่า TEM จะมีการวางชิ้นตัวอย่างจะอยู่ที่เลนส์คอนเดนเซอร์ กำลังขยายของภาพจะเกิดโดยระบบเลนส์ออฟเฟกตีฟ เลนส์อินเตอร์มีเดียทและฉายภาพขยายลงบนฉากเรืองแสงด้วยเลนส์โปรเจกเตอร์ การบันทึกภาพสามารถบันทึกโดยตรงบนแผ่นฟิล์มซึ่งบรรจุไว้ได้แผ่นเรืองแสง ในขณะที่จากภาพที่ 2.4 จะเห็นได้ว่า SEM จะมีการวางชิ้นตัวอย่างจะอยู่ที่เลนส์ออฟเฟกตีฟ ลำอิเล็กตรอนในระบบเลนส์จะถูกโฟกัสลงบนผิวของชิ้นตัวอย่าง และถูกควบคุมให้เคลื่อนที่ตามบริเวณที่ต้องการศึกษาโครงสร้างด้วยระบบสแกน และสร้างสัญญาณภาพขยายอิเล็กตรอนบนจอภาพ การบันทึกภาพจะบันทึกจากจอภาพแคโทดเรย์





ภาพที่ 2.3 ภาพตัดขวางของคอลัมน์กล้องจุลทรรศน์แบบ TEM รุ่น JEM - 2010 ของบริษัท JEOL



ภาพที่ 2.4 ภาพตัดขวางของคอลัมน์กล้องจุลทรรศน์แบบ SEM รุ่น 35CF ของบริษัท JEOL (วิรุพท์ มัลลละวิรัช และ สุวิทย์ ปุณณชัยยะ, 2534)

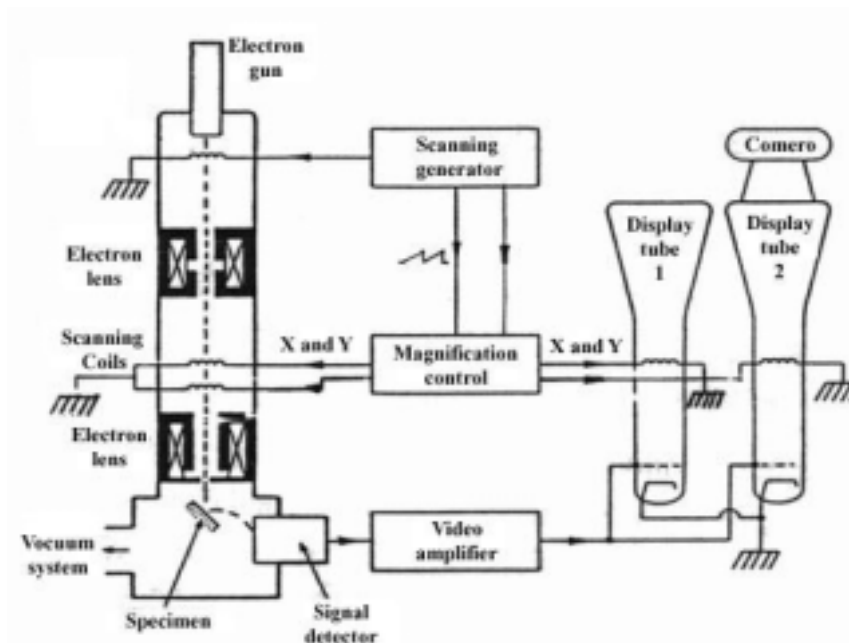
## 2.6 ระบบภาพของกล้องจุลทรรศน์อิเล็กตรอน

เนื่องจากระบบภาพของ SEM และ TEM เหมือนกัน (วิรุพท์ มัลลละวิรัช และ สุวิทย์ ปุณณชัยยะ, 2534) และในงานวิทยานิพนธ์ก็ได้ใช้ภาพจาก SEM เป็นส่วนใหญ่ ดังนั้นจะขอกล่าวเฉพาะระบบภาพของ SEM ภาพไมโครกราฟของ SEM ที่ปรากฏบนจอ เกิดจากระบวนการควบคุมการสแกนลำอิเล็กตรอนที่มีขนาดปลายลำเล็กมากให้เคลื่อนที่บนระนาบ X-Y ของพื้นที่ผิวตัวอย่างทำให้เกิดอันตรกิริยา (interaction) ระหว่างอิเล็กตรอนจากลำอิเล็กตรอนกับอะตอมธาตุในชิ้นตัวอย่าง ทำให้เกิดสัญญาณอิเล็กตรอน (electron signals) ขึ้น ตามคุณสมบัติของธาตุ

องค์ประกอบในชิ้นตัวอย่างนั้น สัญญาณอิเล็กทรอนิกส์เหล่านี้จะถูกเปลี่ยนให้เป็นสัญญาณภาพด้วยอุปกรณ์วัดที่เหมาะสม และส่งผ่านสัญญาณให้ระบบแสดงภาพบนจอแคโทดเรย์ (เดโช ทองอร่าม และ สุวิทย์ ปุณณชัยยะ, 2538)

### 2.6.1 กระบวนการทางภาพของกล้องจุลทรรศน์อิเล็กตรอนแบบส่องกราดแบบอนาล็อก

การกำเนิดภาพไมโครกราฟของระบบ SEM แบบดั้งเดิมนั้นใช้ระบบอิเล็กทรอนิกส์ทั้งหมดควบคุมกระบวนการทางภาพ โดยสัญญาณภาพชนิดอนาล็อกที่ได้รับจากผลของลำอิเล็กตรอนทำอันตรกิริยากับพื้นผิวชิ้นตัวอย่าง จะเกิดขึ้นในช่วงเวลาเดียวกับการแสดงภาพบนจอแคโทดเรย์ตำแหน่งต่อตำแหน่ง ไม่มีการเก็บข้อมูลภาพในหน่วยความจำนอกจากบันทึกภาพลงบนฟิล์ม จึงมักเรียก EM ชนิดนี้ว่า กล้องจุลทรรศน์อิเล็กตรอนแบบส่องกราดระบบอนาล็อก ซึ่งมีแผนภาพการทำงานของระบบในภาพที่ 2.5

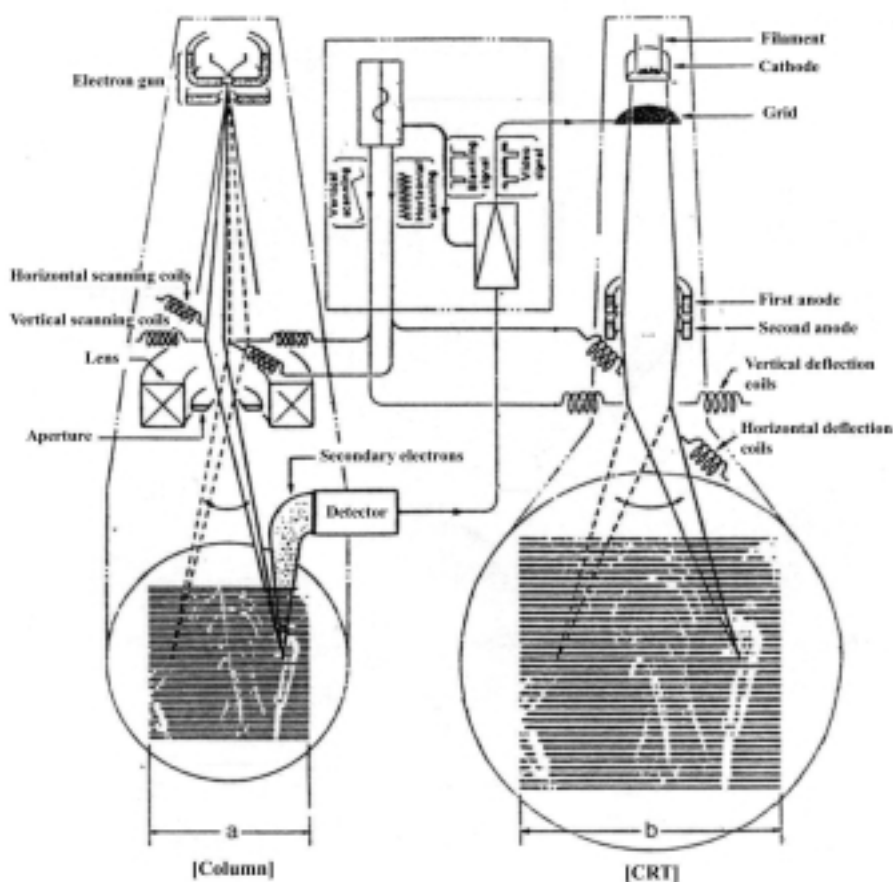


ภาพที่ 2.5 แผนภาพการทำงานของกล้องจุลทรรศน์อิเล็กตรอนแบบส่องกราดระบบอนาล็อก (เดโช ทองอร่าม และ สุวิทย์ ปุณณชัยยะ, 2538)

กล้องชนิดนี้ประกอบด้วยส่วนสำคัญสำหรับกระบวนการทางภาพได้แก่

1. ระบบกำเนิดสัญญาณสแกน (scanning generator) เป็นส่วนที่ทำหน้าที่กำเนิดคลื่นสัญญาณรูปฟันเลื่อย สำหรับควบคุมการสแกนของลำอิเล็กตรอนในคอลัมน์ของ EM และลำอิเล็กตรอนในหลอดแคโทดเรย์ให้เคลื่อนที่บนระนาบ X-Y ของตัวอย่าง และจอภาพตามลำดับ

2. ระบบควบคุมการปรับกำลังขยาย (magnification control) เป็นส่วนที่ทำหน้าที่จัดสัดส่วนขนาดของอัตราสัญญาณสแกนที่ส่งไปควบคุมการสแกน บนพื้นที่จอภาพต่อสัญญาณสแกนบนพื้นผิวตัวอย่าง เพื่อควบคุมกำลังขยายของ EM โดยพื้นที่ของระนาบ X-Y บนพื้นผิวตัวอย่างได้รับสัญญาณควบคุมการสแกนที่มีบริเวณเล็กกว่าพื้นที่การสแกนบนระนาบ X-Y ของจอแคโทดเรย์จึงเกิดกำลังขยายภาพขึ้นดังในภาพที่ 2.6



ภาพที่ 2.6 แผนภาพการทำงานของระบบสร้างภาพ

(เดโช ทองอร่าม และ สุวิทย์ ปุณณชัยยะ, 2538)

3. ระบบขยายสัญญาณภาพ (video amplifier) เป็นส่วนที่ทำหน้าที่ขยายสัญญาณภาพจากอุปกรณ์วัดสัญญาณอิเล็กทรอนิกส์ เพื่อให้มีสัญญาณแรงขึ้นพอที่จะควบคุมวงจรกริดของจอแคโทดเรย์ และได้ปริมาณอิเล็กทรอนิกส์บนพื้นที่จอภาพสัมพันธ์กับปริมาณสัญญาณอิเล็กทรอนิกส์จากชิ้นตัวอย่างเกิดความเปรียบต่างภาพบนจอแสดงภาพชัดเจน

4. ระบบแสดงภาพและบันทึกภาพ (CRT display and recording unit) เป็นส่วนที่ทำหน้าที่แสดงภาพไมโครกราฟและถ่ายภาพเพื่อนำไปวิเคราะห์รายละเอียด โดยระบบแสดงภาพบนจอแคโทดรีจะมีลักษณะการทำงานให้เล็กลงดังนี้

4.1 การแสดงภาพระบบโทรทัศน์ (TV mode) เป็นการแสดงภาพด้วยความเร็วในการสแกนสูง เกิดเป็นภาพเต็มพื้นที่ต่อเนื่องเช่นเดียวกับโทรทัศน์ ใช้ในกรณีที่ปรับกำลังขยายของกล้องต่ำเพื่อเลือกบริเวณพื้นผิวที่ต้องการวิเคราะห์ก่อนเพิ่มกำลังขยายของกล้อง

4.2 การแสดงภาพแบบความเร็วสแกนสูง (rapid scan) เป็นการแสดงภาพที่มีความเร็วในการสแกนช้ากว่าระบบโทรทัศน์ ใช้สำหรับการปรับความสว่าง ความเปรียบต่างภาพ และความคมชัดเบื้องต้นที่กำลังขยายสูง และแสดงภาพเฉพาะพื้นที่ที่ไม่เต็มจอภาพ

4.3 การแสดงภาพแบบความเร็วสแกนต่ำ (slow speed scan) ใช้สำหรับปรับความผิดเพี้ยน และความคมชัดของภาพขั้นสุดท้าย ก่อนการบันทึกภาพโดยตรงจากจอภาพ

การทำงานของระบบกำเนิดภาพ เริ่มจากปลายลำอิเล็กตรอนที่ถูกระบบเลนส์อิเล็กตรอนโฟกัสลงบนผิวชิ้นตัวอย่างเฉพาะจุดและเกิดสัญญาณอิเล็กตรอนขึ้น ระบบควบคุมการสแกนจะบังคับให้ลำอิเล็กตรอนเคลื่อนที่บนพื้นผิวไมโครเป็นเส้นสแกนบนระนาบ X-Y สัญญาณที่เกิดจากการอันตรกิริยาบนผิวชิ้นตัวอย่างจุดต่อจุดจะถูกเปลี่ยนเป็นสัญญาณภาพด้วยอุปกรณ์วัดส่งให้ระบบสแกนภาพของจอแคโทดรีซึ่งทำงานเข้าจังหวะกันเส้นต่อเส้นต่อเนื่องกันไป สัญญาณภาพที่เกิดจากการสแกนแต่ละเส้นจะปรากฏเป็นภาพบนจอภาพในเวลาเดียวกัน ดังแสดงรูปสัญญาณที่บันทึกลงบนภาพไมโครกราฟในภาพที่ 2.7 แต่เนื่องจากขณะที่ SEM ทำงานที่กำลังขยายสูงจะไม่สามารถสแกนด้วยความเร็วสูงได้ เพราะจะเกิดความผิดเพี้ยนของภาพ อันเป็นผลจากการเปลี่ยนแปลงสนามแม่เหล็กของแกนขดลวดสแกนจึงต้องใช้ความเร็วในการสแกนต่ำ ดังนั้นภาพไมโครกราฟที่แสดงผลบนจอภาพจึงปรากฏเฉพาะส่วนที่ผ่านการสแกน และคงการเรืองแสงด้วยความเฉื่อยของสารที่ฉาบจอเรืองแสง (persistence of screen) ในช่วงเวลาอันสั้นเท่านั้น ด้วยเหตุนี้ภาพเต็มพื้นที่ที่สแกนจึงเกิดจากการบันทึกภาพด้วยระบบถ่ายภาพบนฟิล์ม หรือกล้องโพลารอยด์เท่านั้น



ภาพที่ 2.7 ลักษณะของสัญญาณภาพของ SEM

(เดโช ทองอร่าม และ สุวิทย์ ปุณณชัยยะ, 2538)

### 2.6.2 ข้อจำกัดของระบบสร้างภาพแบบอนาล็อกในกระบวนการวิเคราะห์ภาพ

จากการทำงานของระบบสร้างภาพของ SEM ระบบอนาล็อกนั้น แม้ว่าผู้ใช้งานจะมีประสบการณ์และความชำนาญมากเพียงใดก็ตาม แต่การใช้งานเพื่อให้ภาพไมโครกราฟมีคุณภาพที่ดียังคงมีปัญหา (เดโช ทองอร่าม และ สุวิทย์ ปุณณชัยยะ, 2538) ดังนี้

1. ภาพไมโครกราฟที่ปรากฏบนจอแคโทดเรย์ของ SEM ระบบอนาล็อกนั้นมิได้เป็นภาพนิ่งทั้งภาพ แม้ว่าจะมีระบบสแกนความเร็วสูงในรูปแบบของระบบโทรทัศน์แต่จะใช้ได้เฉพาะกำลังขยายของกล้องที่ต่ำกว่า 3000 เท่า หรือในระบบสแกนเร็วก็จะมีพื้นที่แสดงภาพเล็กกว่าปกติ

2. การปรับความเปรียบต่างและความสว่างของภาพไมโครกราฟที่ปรากฏบนจอภาพ เฉพาะส่วนที่สแกนผ่านทำได้ลำบาก บางครั้งต้องทดลองถ่ายภาพหลายครั้ง ทำให้เสียเวลาและสิ้นเปลืองค่าใช้จ่ายมาก

3. การสแกนภาพไมโครกราฟข้ามบนพื้นที่ไมโครของชิ้นตัวอย่างที่กำลังขยายสูงเป็นเวลานาน จะทำให้ผิวชิ้นตัวอย่างสกปรกและสูญเสียรายละเอียดภาพ โดยเฉพาะชิ้นตัวอย่างที่ไม่นำไฟฟ้าจะทำให้เกิดปัญหา charging ของประจุจากลำอิเล็กตรอนรุนแรง

4. แม้ว่าการปรับปรุงคุณภาพของกระบวนการสร้างภาพแบบอนาล็อกจะสามารถทำได้ด้วยการใช้วงจรอิเล็กทรอนิกส์ เช่นการใช้วงจรขยายแบบไม่เป็นเชิงเส้น หรือวงจรกรองสัญญาณรบกวน เป็นต้น แต่ที่กำลังขยายสูงการปรับแต่งต้องใช้เวลาและลำบากมาก

### 2.6.3 กระบวนการทางกายภาพของกล้องจุลทรรศน์อิเล็กตรอนแบบส่องกราดระบบเชิงตัวเลข

จากข้อจำกัดที่กล่าวมา ได้มีการพัฒนาและปรับปรุงคุณภาพของภาพและวิธีวิเคราะห์ภาพจาก SEM แบบอนาล็อก โดยการประยุกต์เทคโนโลยีระบบเชิงตัวเลขและโปรแกรมคอมพิวเตอร์ เพื่อใช้ปรับการทำงานของ SEM ให้ทำงานในระบบเชิงตัวเลขได้ และเรียก EM ชนิดนี้ว่า กล้องจุลทรรศน์อิเล็กตรอนแบบส่องกราดระบบเชิงตัวเลข โดยมีการทำงานดังนี้คือ จะทำการสแกนภาพความเร็วต่ำบนพื้นที่ไม่ใคร่เพียงครั้งเดียว และทำการเก็บข้อมูลภาพที่ได้ลงบนหน่วยความจำ เพื่อแสดงภาพข้อมูลจากหน่วยความจำได้อย่างอิสระเป็นเวลานานเท่าที่ต้องการ โดยไม่ต้องอาศัยกระบวนการอัดขยายภาพในห้องมืด นอกจากนี้ยังสามารถวิเคราะห์รายละเอียดภาพด้วยกระบวนการประมวลผลของคอมพิวเตอร์ได้อีกด้วย

## 2.7 กล้องจุลทรรศน์อิเล็กตรอนแบบส่องกราดรุ่น JSM – 5800 LV

ภาพตัดขวางของคอลัมน์กล้อง SEM เป็นดังภาพที่ 2.8 และมีความสามารถ ลักษณะการใช้งาน และประเภทของชิ้นตัวอย่าง ดังนี้

### 2.7.1 ความสามารถในการแจกแจง

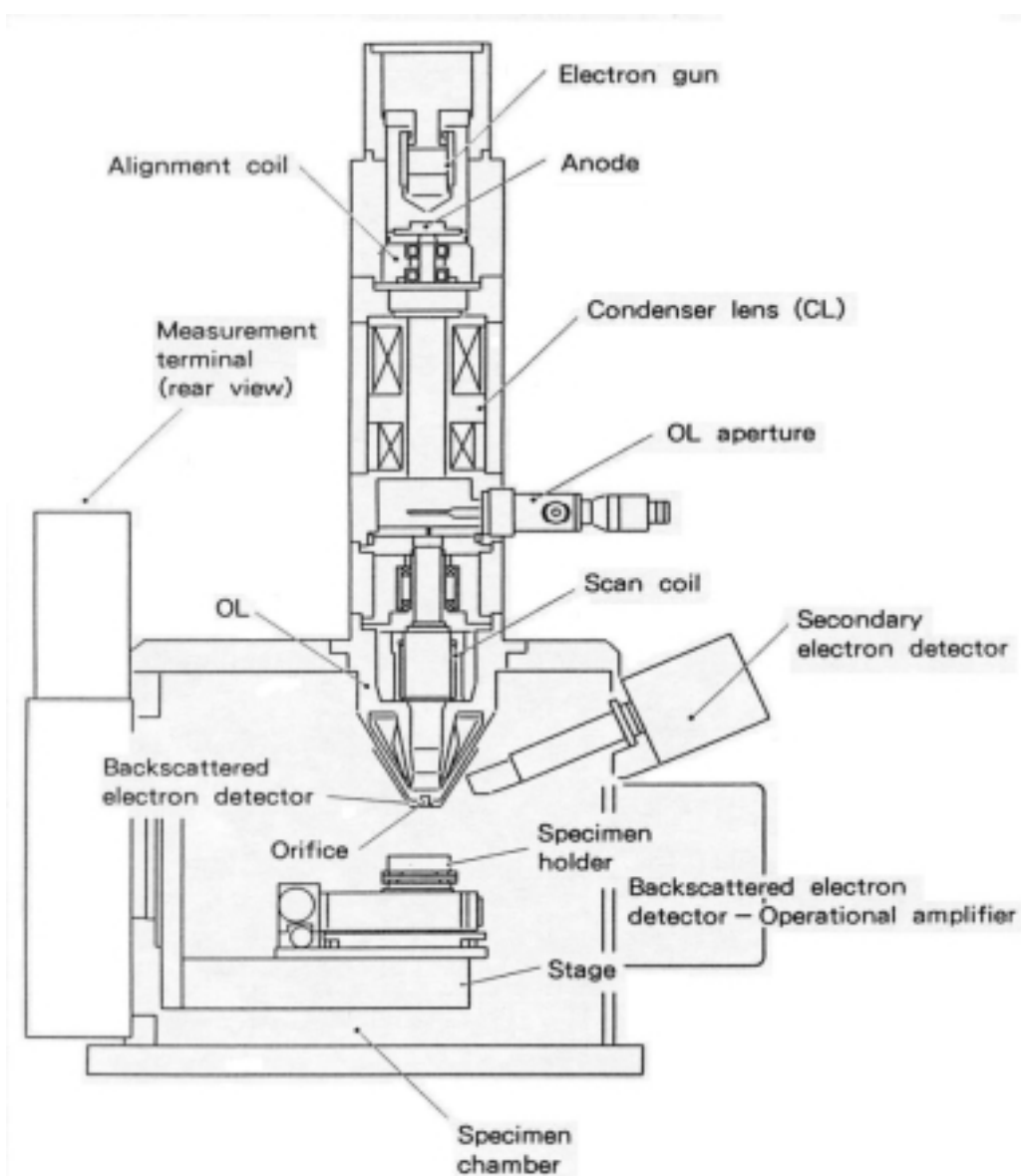
- ที่ระบบสุญญากาศต่ำ (low vacuum mode) เท่ากับ 5.5 นาโนเมตร ที่ตำแหน่งเลนส์ห่างจากวัตถุ 5 มิลลิเมตร และแรงดันไฟฟ้า 30 KV ในการแปลงสัญญาณอิเล็กตรอนกระเจิงกลับ (back scattering electron)
- ที่ระบบสุญญากาศสูง (high vacuum mode) เท่ากับ 3.5 นาโนเมตร ที่ตำแหน่งเลนส์ห่างจากวัตถุ 8 มิลลิเมตร และแรงดันไฟฟ้า 30 KV ในการแปลงสัญญาณอิเล็กตรอนทุติยภูมิ (secondary electron)

### 2.7.2 ลักษณะการใช้งาน

- ตรวจสอบลักษณะผิวภายนอกของตัวอย่าง
- ตรวจสอบการเรียงตัวของผลึกด้วยระบบการรับสัญญาณเลี้ยวเบนของอิเล็กตรอนกระเจิงกลับ (electron backscattered diffraction: EBSD)
- ตรวจสอบการเปลี่ยนแปลงตัวอย่างจากการดึง

### 2.7.3 ประเภทของชิ้นตัวอย่าง

- โลหะ
- เซรามิก
- พอลิเมอร์
- เซลเนื้อเยื่อที่ผ่านการดึ่งน้ำออกจากเซลล์



ภาพที่ 2.8 ภาพตัดขวางของคอลัมน์กล้องจุลทรรศน์แบบ SEM  
รุ่น JSM – 5800 LV ของบริษัท JEOL





ภาพที่ 2.9 ภาพของกล้องจุลทรรศน์แบบ SEM  
รุ่น JSM – 5800 LV ของบริษัท JEOL

## 2.8 สรุป

การวิเคราะห์รายละเอียดของโครงสร้างระดับไมโครบนพื้นผิววัสดุ เช่นขนาดของเกรน การกระจายของอนุภาค ความพรุน ความหนาแน่น หรือร่องรอยอันเกิดจากความผิดปกติของผิววัสดุ เป็นสิ่งจำเป็นอย่างยิ่งในการควบคุมคุณภาพของงานด้านโลหกรรมและวัสดุศาสตร์ การผลิตประดิษฐ์ทางไมโครอิเล็กทรอนิกส์ ที่นับวันยังต้องการความละเอียดแม่นยำสูงขึ้น ทำให้การวิเคราะห์ผลต่างๆด้วยภาพจาก EM มีความสำคัญมากขึ้น ทำให้ภาพ EM เป็นภาพที่มีรายละเอียดมาก และไฟล์ภาพจะมีขนาดใหญ่ ทำให้ในงานวิจัยต่างๆ ที่ใช้ภาพ EM ในการวิเคราะห์รายละเอียดของงาน ที่ต้องใช้ระยะเวลานานในการทำงานนั้นจำเป็นต้องใช้เนื้อที่ในการบันทึกข้อมูลสูง ดังนั้นถ้ามีการนำภาพ EM มาบีบอัดข้อมูล จะทำให้ช่วยลดปัญหานี้ได้

## บทที่ 3

### หลักการเบื้องต้นในการบีบอัดข้อมูลภาพ

#### 3.1 บทนำ

การลดข้อมูลภาพเป็นกระบวนการลดจำนวนของบิตที่จะใช้ในการแสดงภาพ การลดข้อมูลภาพจะคำนึงถึงข้อมูลทางสถิติของภาพ ร่วมกับความสัมพันธ์ที่ยอมรับได้ด้วยการสังเกตด้วยตาของมนุษย์ โดยการลดจำนวนของบิตที่ต้องการใช้ในการแสดงภาพ ทำให้การลดข้อมูลภาพสามารถนำไปประยุกต์ใช้ในงานอื่นๆ ได้อย่างมีประสิทธิภาพ การจัดเก็บข้อมูลสามารถพัฒนาให้ดีขึ้นได้ด้วยการใช้การลดข้อมูลเพื่อทำการลดความต้องการหน่วยความจำ

การจำแนกการบีบอัดข้อมูลสามารถแบ่งออกได้เป็น 2 กลุ่มคือ การบีบอัดข้อมูลแบบไม่มีการสูญเสีย (lossless compression) และการบีบอัดข้อมูลแบบที่มีการสูญเสีย (lossy compression)

#### 3.2 การบีบอัดข้อมูลภาพแบบที่ไม่มีการสูญเสีย

การบีบอัดข้อมูลภาพแบบที่ไม่มีการสูญเสีย เป็นวิธีการที่นำเอาข้อมูลภาพ ไปผ่านขั้นตอนการบีบอัดข้อมูลภาพ ซึ่งภาพที่ได้จะเหมือนภาพต้นแบบทุกประการ นิยมใช้กับข้อมูลที่มีความสำคัญและไม่ต้องการให้มีการสูญเสีย เช่น ข้อมูลภาพที่ใช้วินิจฉัยทางการแพทย์ เป็นต้น

การบีบอัดข้อมูลภาพแบบไม่มีการสูญเสีย จะเป็นการเข้ารหัสเอนโทรปี (entropy coding) โดยเอนโทรปีจะใช้อธิบายความไม่แน่นอนของข้อมูล และแสดงค่าเฉลี่ยของจำนวนบิตที่แทนข้อมูล ภาพจะถูกลดข้อมูลโดยการเข้ารหัสข้อมูลจุดภาพที่เกิดขึ้นบ่อยด้วยจำนวนบิตที่น้อย และเข้ารหัสข้อมูลจุดภาพที่เกิดขึ้นน้อยกว่าด้วยจำนวนบิตที่มากขึ้นตามลำดับ ซึ่งโดยเฉลี่ยแล้วจะเป็นการลดจำนวนบิตที่ใช้แทนข้อมูล วิธีการบีบอัดข้อมูลภาพแบบที่ไม่มีการสูญเสียได้แก่

##### 3.2.1 การเข้ารหัสแบบรันเลนจอร์ (run-length coding)

การเข้ารหัสแบบรันเลนจอร์ (Steven, 1992) เป็นวิธีการที่ง่ายที่สุดในการเข้ารหัสข้อมูล โดยอาศัยพื้นฐานของข้อมูลภาพในระดับพิกเซลที่อยู่ติดกันจะมีการเปลี่ยนแปลงของระดับความเข้มไม่มากและมีจำนวนที่ซ้ำกัน จะทำการเก็บจำนวนของพิกเซลที่ซ้ำกันแทนการเก็บค่าระดับความเข้มของแต่ละพิกเซลเหล่านั้น เป็นดังภาพที่ 3.1

ถ้าข้อมูลภาพแต่ละพิกเซลเป็นดังนี้

00 00 00 11 11 11 11 33 33 33 33 33 55 55 55 55 56 56 57 .....

เมื่อทำการเข้ารหัสแบบรันเลนจ์ จะได้ข้อมูลใหม่ดังนี้

00 03 11 04 33 05 55 04 56 02 57 01 .....

ภาพที่ 3.1 แสดงตัวอย่างรหัสแบบรันเลนจ์

### 3.2.2 การเข้ารหัสฮัฟแมน (huffman coding)

การเข้ารหัสฮัฟแมน (Steven, 1992) มีหลักการสำคัญของการเข้ารหัสคือ ข้อมูลที่มีความน่าจะเป็นสูง จะถูกเข้ารหัสด้วยจำนวนบิตที่น้อยกว่าข้อมูลที่มีความน่าจะเป็นต่ำกว่า

การเข้ารหัสฮัฟแมน มีขั้นตอนการเข้ารหัสดังนี้

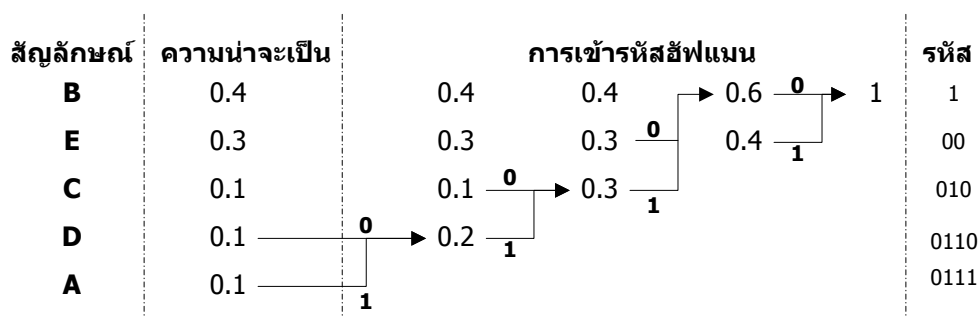
ขั้นตอนที่ 1 ทำการหาความน่าจะเป็นของข้อมูลแต่ละตัว และเรียงข้อมูลตามความน่าจะเป็นจากน้อยไปหามาก

ขั้นตอนที่ 2 นำความน่าจะเป็นที่มีค่าต่ำสุด 2 ค่ามารวมกันเพื่อสร้างเป็นโหนดใหม่ แล้วนำความน่าจะเป็นใหม่ที่ได้อาเรียงลำดับใหม่ ร่วมกับความน่าจะเป็นเดิมที่เหลือ

ขั้นตอนที่ 3 กำหนดค่าไบนารีให้กับโหนดคู่ที่ถูกกำหนดให้เป็นเส้นทางของโหนดใหม่นั้น โดยโหนดหนึ่งมีค่า 0 ส่วนอีกโหนดมีค่า 1

ขั้นตอนที่ 4 ทำซ้ำตามขั้นตอนที่ 2 และ 3 จนกระทั่งได้ความน่าจะเป็นสุดท้ายเป็น 1

ตัวอย่างของการสร้างรหัสฮัฟแมน แสดงดังภาพที่ 3.2



ภาพที่ 3.2 แสดงตัวอย่างการเข้ารหัสฮัฟแมน

### 3.2.3 การเข้ารหัสเลขคณิต (arithmetic coding)

การเข้ารหัสเลขคณิต (Rafael, and Richard, 1993) ข้อมูลจะถูกแทนที่ด้วยช่วงระยะห่างของจำนวนจริงระหว่าง 0 และ 1 โดยข้อมูลแต่ละตัวจะลดขนาดของระยะห่างตามความน่าจะเป็นของข้อมูลที่นำมาเข้ารหัส ข้อมูลที่มีความน่าจะเป็นสูงจะทำให้ระยะห่างถูกลดลงได้น้อยกว่าข้อมูลที่มีความน่าจะเป็นต่ำ ดังนั้นข้อมูลที่มีความน่าจะเป็นสูงจะใช้จำนวนบิตในการเข้ารหัสน้อยกว่า

การเข้ารหัสเลขคณิตมีขั้นตอนการเข้ารหัสดังนี้

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

ขั้นตอนที่ 2 นำสัญลักษณ์ของข้อมูลมาพิจารณา โดยแบ่งเป็นอีก 2 ขั้นตอนย่อยคือ

2.1 ทำการแบ่งช่วงขอบเขตปัจจุบันให้เป็นช่วงขอบเขตย่อย ตามความน่าจะเป็นของสัญลักษณ์ โดยมีสมการหาขอบเขตของแต่ละสัญลักษณ์ดังนี้

$$P_L(i) = \sum_{s=1}^{i-1} p(s) \quad (3.1)$$

$$P_H(i) = \sum_{s=1}^i p(s) \quad (3.2)$$

เมื่อ  $P_L(i)$  คือความน่าจะเป็นของขอบเขตล่างของสัญลักษณ์ตัวที่  $i$

$P_H(i)$  คือความน่าจะเป็นของขอบเขตบนของสัญลักษณ์ตัวที่  $i$

$p(s)$  คือความน่าจะเป็นของสัญลักษณ์ตัวที่  $s$

2.2 เลือกช่วงขอบเขตย่อยใหม่ที่สอดคล้องกับสัญลักษณ์ที่นำมาเข้ารหัสของข้อมูลตัวใหม่ แล้วนำช่วงขอบเขตดังกล่าวมาแทนให้เป็นช่วงขอบเขตปัจจุบันแทน การเข้ารหัสจะทำไปจนกว่าจะสิ้นสุดชุดของข้อมูลที่นำมาเข้ารหัส หรือเมื่อพบสัญลักษณ์ที่แสดงถึงจุดสิ้นสุดของข้อมูล โดยมีสมการหาขอบเขตใหม่ดังนี้

$$L_{\text{new}} = L_{\text{old}} + P_L(i)(H_{\text{old}} - L_{\text{old}}) \quad (3.3)$$

$$H_{\text{new}} = L_{\text{old}} + P_H(i)(H_{\text{old}} - L_{\text{old}}) \quad (3.4)$$

เมื่อ  $L_{\text{new}}$  และ  $H_{\text{new}}$  คือขอบเขตใหม่

$L_{\text{old}}$  และ  $H_{\text{old}}$  คือขอบเขตปัจจุบัน

ขั้นตอนที่ 3 นำค่าช่วงขอบเขตสุดท้ายที่ได้ในขั้นตอนที่ 2 มาทำการแปลงเป็นข้อมูลไบนารี ซึ่งข้อมูลไบนารีที่ได้จะเป็นค่าที่ได้จากการเข้ารหัส โดยมีสมการหาค่าจำนวนไบนารีดังนี้

$$n = -\log_2(p_t) \quad (3.5)$$

เมื่อ  $p_t$  คือค่าเฉลี่ยของขอบเขตปัจจุบัน

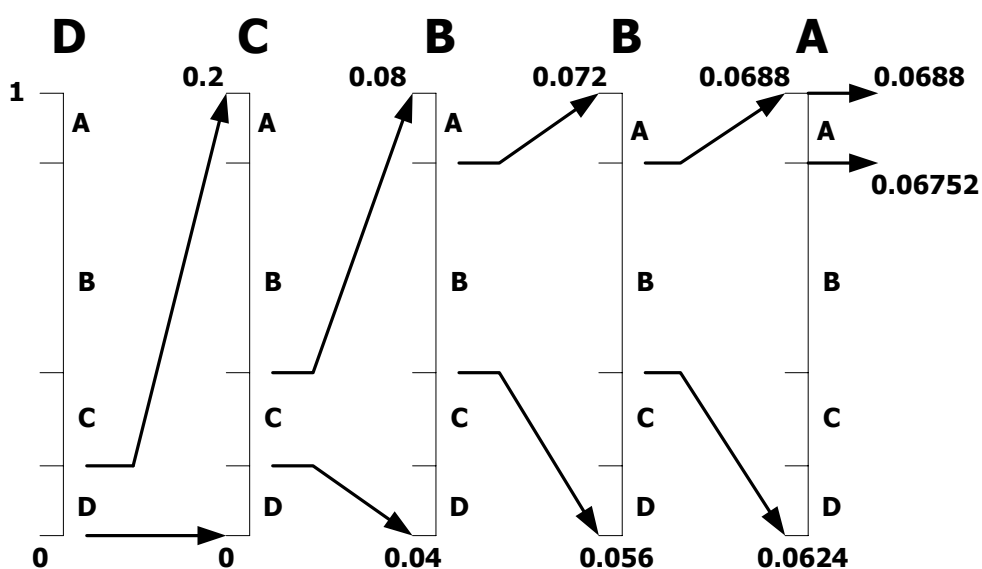
$n$  คือจำนวนบิตของข้อมูลที่ได้จากการเข้ารหัส

ซึ่งข้อมูลที่ได้จากการเข้ารหัสเลขคณิตคือข้อมูลไบนารีที่ได้จากการแปลงเลขฐานสองของ  $p_t$  จำนวนเท่ากับ  $n$  บิต

ตัวอย่างการเข้ารหัสเลขคณิต แสดงดังภาพที่ 3.3 และกำหนดให้สัญลักษณ์ มีค่าความน่าจะเป็นดังนี้

- A มีค่าความน่าจะเป็นเท่ากับ 0.2
- B มีค่าความน่าจะเป็นเท่ากับ 0.4
- C มีค่าความน่าจะเป็นเท่ากับ 0.2
- D มีค่าความน่าจะเป็นเท่ากับ 0.2

และกำหนดข้อมูลในการเข้ารหัสคือ DCBBA



ภาพที่ 3.3 แสดงการเข้ารหัสเลขคณิต

จากภาพที่ 3.3 จะได้ขอบเขตของการเข้ารหัสสัญลักษณ์ชุดนี้คือ  $[0.06752, 0.0688)$  และจำนวนบิตที่ใช้แทนสัญลักษณ์นี้เท่ากับ  $-\log_2(0.06816) = 3.87$  บิต ให้ทำการปัดขึ้นเพื่อทำให้เป็นจำนวนเต็ม จะได้จำนวนบิตที่ใช้แทนสัญลักษณ์ชุดนี้เท่ากับ 4 บิต

### 3.3 การบีบอัดข้อมูลภาพแบบที่มีการสูญเสีย

เป็นวิธีการที่ยอมให้ภาพที่ได้จากการบีบอัดข้อมูลมีความแตกต่างจากภาพต้นแบบ โดยยอมให้มีการสูญเสียข้อมูลบ้าง นิยมใช้ในการประมวลผลภาพทั่วไปเป็นอย่างมาก เนื่องจากตาของมนุษย์ไม่สามารถตรวจจับข้อมูลในส่วนที่มีการสูญเสียได้ อย่างไรก็ตามคุณภาพของภาพที่ได้มีความสัมพันธ์กับอัตราการบีบอัดข้อมูลภาพด้วย โดยข้อมูลภาพที่มีอัตราการบีบอัดที่สูงจะมีขนาดของข้อมูลภาพที่มีขนาดเล็ก แต่อาจทำให้มีคุณภาพของภาพที่ผ่านการบีบอัดลดลงตามไปด้วย วิธีการบีบอัดข้อมูลภาพแบบที่มีการสูญเสียได้แก่

#### 3.3.1 การเข้ารหัสโดยการแปลง (transform coding)

การบีบอัดข้อมูลด้วยการเข้ารหัสการแปลง จะทำการแปลงข้อมูลอินพุตที่อยู่ในรูปของสเปซโดเมน (spatial domain) ให้อยู่ในรูปของสเปกตรัม (spectral) หรือเป็นโดเมนความถี่นั่นเอง โดยใช้วิธีการแปลงแบบต่างๆ เช่น การแปลงฟูเรียร์ (fourier transform) ในการแปลงข้อมูลที่อยู่ในรูปของสเปซโดเมนให้อยู่ในรูปสัมประสิทธิ์ของพลังงานความถี่ โดยที่ความถี่ต่ำๆจะมีพลังงานสูง และที่ความถี่สูงๆ พลังงานจะลดลงไป สำหรับข้อมูลภาพโดยทั่วไปนั้นจะประกอบด้วยส่วนที่เป็นความถี่สูง คือส่วนที่เป็นขอบภาพ และส่วนที่เป็นความถี่ต่ำ คือส่วนที่เป็นพื้นผิวของภาพ ซึ่งจำนวนบิตที่ใช้ในการเข้ารหัสของแต่ละช่วงความถี่จะไม่เท่ากัน ดังนั้นเมื่อต้องการลดข้อมูลให้ได้มาก ค่าของพลังงานความถี่สูงจะถูกตัดทิ้งไป ทำให้รายละเอียดส่วนที่เป็นขอบของภาพขาดหายไป ภาพที่ได้จะออกมัวๆ (blur) ขาดความคมชัด

การแปลงที่ใช้ในการบีบอัดข้อมูลภาพมีอยู่หลายวิธี เช่น การแปลงฟูเรียร์แบบเร็ว (fast fourier transform), การแปลงวอลซ์-ฮาดามาร์ด (walsh-hadamard transform), การแปลงคอสครีทโคไซน์ (discrete cosine transform: DCT) และการแปลงฮาร์ (haar transform) เป็นต้น ซึ่งแต่ละวิธีมีข้อดีและเสียแตกต่างกันไป แต่วิธีที่นิยมใช้ในการบีบอัดข้อมูลภาพมากที่สุดคือ การแปลงคอสครีทโคไซน์ เนื่องจากเป็นวิธีที่สามารถคำนวณได้ง่ายและเร็ว ตัวอย่างของการบีบอัดข้อมูลด้วยวิธีนี้คือ การบีบอัดข้อมูลโดยเทคนิคของ Joint Photographic Expert Group (JPEG)

การบีบอัดข้อมูลแบบ JPEG แบบมาตรฐาน (ศิริพร เดชะสีตารักษ์, 2543) มีขั้นตอนการบีบอัดข้อมูลดังนี้

ขั้นตอนที่ 1 ทำการแบ่งข้อมูลออกเป็นบล็อกย่อยๆ และทำการแปลงข้อมูลแต่ละบล็อกให้อยู่ในโดเมนความถี่ด้วยวิธี DCT ซึ่งข้อมูลที่ได้จากการแปลงจะเรียกว่าสัมประสิทธิ์ DCT มีอยู่ด้วยกัน 2 ส่วนคือ ค่าสัมประสิทธิ์ DC และค่าสัมประสิทธิ์ AC และมีสมการแปลง DCT ดังสมการที่ 3.6 และมีสมการแปลงกลับ DCT ดังสมการที่ 3.7 จากสมการที่ 3.6 ค่า DC คือข้อมูลตัวแรกที่ได้จากการแปลง DCT และข้อมูลที่เหลือจะเป็นค่า AC

$$C(u,v) = \frac{1}{\sqrt{2N}} a(u)a(v) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) \cos\left[\frac{(2x+1)v\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2M}\right] \quad (3.6)$$

$$f(x,y) = \frac{1}{\sqrt{2N}} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} a(u)a(v)C(u,v) \cos\left[\frac{(2x+1)v\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2M}\right] \quad (3.7)$$

เมื่อ  $C(u,v)$  คือสัมประสิทธิ์ DCT

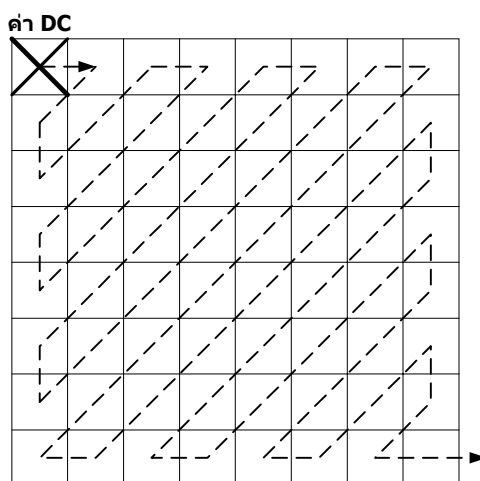
$f(x,y)$  คือภาพต้นแบบที่มีขนาดภาพเท่ากับ  $N \times M$  พิกเซล

$$a(i) = \frac{1}{\sqrt{2}}, i = 0$$

$$a(i) = 1, i > 0$$

ขั้นตอนที่ 2 ทำการควอนไทซ์ค่าสัมประสิทธิ์ DCT โดยทำการควอนไทซ์ด้วยค่าสัมประสิทธิ์ของการควอนไทซ์และทำการปิดเศษขึ้นให้เป็นค่าจำนวนเต็ม ทำให้ข้อมูลมีค่าเป็นศูนย์มากขึ้น เพื่อให้การเข้ารหัสมีประสิทธิภาพมากยิ่งขึ้น

ขั้นตอนที่ 3 การเข้ารหัสฮัฟแมน โดยค่าสัมประสิทธิ์ DC ใช้การเข้ารหัสของผลต่างระหว่างค่าสัมประสิทธิ์ DC เนื่องจากค่าสัมประสิทธิ์ DC แต่ละบล็อกจะมีค่าใกล้เคียงกัน สำหรับค่า AC จะเรียงลำดับแบบซิกแซก (zigzag sequence) เป็นดังภาพที่ 3.4 เพื่อให้ค่าสัมประสิทธิ์ที่มีความถี่ต่ำที่มีค่าส่วนใหญ่ไม่เป็นศูนย์อยู่ติดกัน และค่าสัมประสิทธิ์ความถี่สูงที่มีค่าส่วนใหญ่เป็นศูนย์อยู่ติดกันเพื่อสะดวกในการเข้ารหัสแบบรันเลนจ์



ภาพที่ 3.4 แผนภาพแสดงการเรียงลำดับแบบซิกแซก

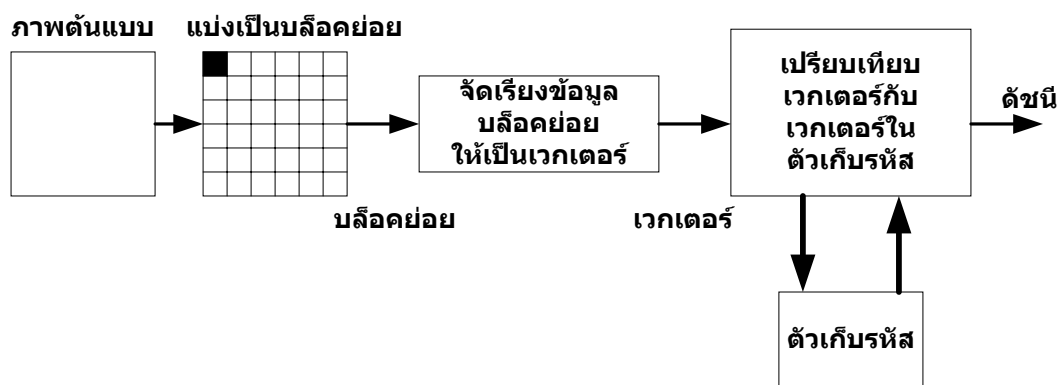
### 3.3.2 การเข้ารหัสควอนไทซ์แบบเวกเตอร์

การเข้ารหัสควอนไทซ์แบบเวกเตอร์ (เกศกนก กฤตยาฤทธิวัฒน์, 2543) มีแผนภาพการเข้ารหัสเป็นดังภาพที่ 3.5 จากภาพจะเห็นว่าการเข้ารหัสชนิดนี้มีลักษณะเช่นเดียวกับ JPEG ในตอนเริ่มต้นคือการแบ่งข้อมูลออกเป็นบล็อกย่อยๆ ขนาด  $N \times N = K$  พิกเซล และเรียงให้อยู่ในรูปเวกเตอร์ขนาด  $K$  มิติ และอาศัยหลักการแทนที่บล็อกย่อยของข้อมูลด้วยเลขดัชนีของชุดข้อมูลที่ใช้แทนบล็อกย่อยเหล่านั้น มักจะใช้เลขดัชนีเป็นเลขจำนวนนับและมีค่าจำกัดอยู่ระดับหนึ่ง ในการเข้ารหัสจะเปรียบเทียบอินพุตเวกเตอร์กับเวกเตอร์รหัสทุกตัวที่เก็บอยู่ในตัวเก็บรหัส โดยเปรียบเทียบหาเวกเตอร์รหัสที่แทนด้วยเวกเตอร์ที่มีความผิดเพี้ยนจากอินพุตเวกเตอร์น้อยที่สุด ผลลัพธ์ที่ได้ก็คือเลขดัชนีที่ใช้แทนเวกเตอร์รหัสที่มีความผิดเพี้ยนน้อยที่สุด

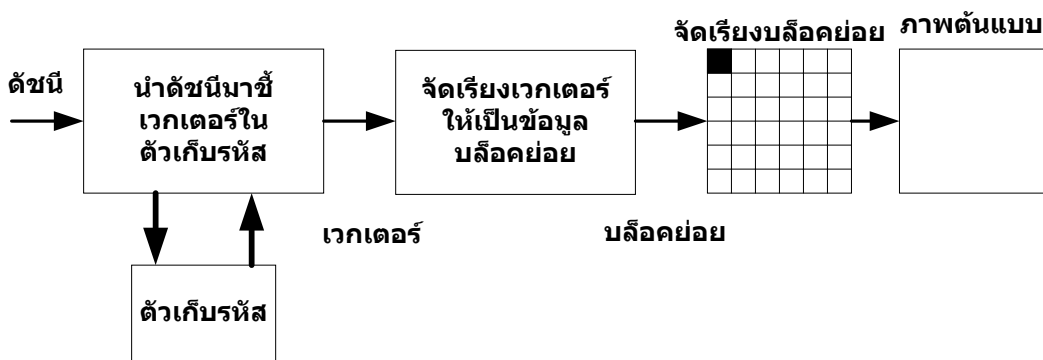
มีแผนภาพการถอดรหัสเป็นดังภาพที่ 3.6 ในขั้นตอนถอดรหัสจะนำค่าดัชนีมาซึ่งข้อมูลเวกเตอร์รหัสที่เก็บอยู่ในตัวเก็บรหัสเดียวกับขั้นตอนเข้ารหัส ผลลัพธ์ที่ได้จะเป็นเอาต์พุตเวกเตอร์ จากนั้นนำเอาต์พุตเวกเตอร์ที่ได้มาจัดเรียงใหม่ให้เป็นบล็อกย่อยของภาพดั้งเดิม

มีการพัฒนาการบีบอัดข้อมูลชนิดนี้ให้มีประสิทธิภาพเพิ่มมากขึ้น โดยการนำมาใช้งานร่วมกับเทคนิคการเข้ารหัสแบบการแปลงหรือ เทคนิคการเข้ารหัสแบบด้อย





ภาพที่ 3.5 แผนภาพการเข้ารหัสเวกเตอร์ควอนไทซ์



ภาพที่ 3.6 แผนภาพการถอดรหัสเวกเตอร์ควอนไทซ์

### 3.3.3 การเข้ารหัสแบนด์ย่อย (subband coding)

ในการเข้ารหัสแบนด์ย่อย เป็นการเข้ารหัสโดยใช้วิธีการแปลงเวฟเล็ต (wavelet transform) (Burrus, Ramesh, and Haitao, 1998) มีลักษณะการแปลงดังนี้คือ ภาพต้นแบบจะถูกกรองตามช่วงของความถี่เพื่อสร้างเป็นภาพย่อย ซึ่งภาพเหล่านี้ถูกเรียกว่าภาพแบนด์ย่อย (subbands) เนื่องจากแต่ละภาพแบนด์ย่อยได้ถูกลดความกว้างแถบ (bandwidth) เมื่อเปรียบเทียบกับภาพต้นแบบ ทำให้สามารถที่จะทำการลดจำนวนการสุ่ม (downsampling) ลงได้ ซึ่งสามารถทำได้ด้วยกระบวนการในการกรอง (filtering) ส่วนการลดจำนวนการสุ่มนี้เรียกว่าเป็นขั้นตอนการวิเคราะห์ (analysis stage) หลังจากนั้นภาพแบนด์ย่อยที่ได้จะถูกเข้ารหัสด้วยตัวเข้ารหัสต่างๆ และในการสร้างภาพกลับคืน สามารถทำได้โดยการเพิ่มจำนวนการสุ่ม (upsampling) ของภาพแบนด์ย่อยที่ได้ถอดรหัสแล้ว จากนั้นนำมาผ่านตัวกรองแล้วรวมภาพแต่ละแบนด์ย่อยเข้าด้วยกัน เพื่อสร้างภาพกลับคืน เรียกขั้นตอนนี้ว่า ขั้นตอนการสังเคราะห์ (synthesis stage)

การเข้ารหัสภาพแบนด์ย่อยจะไม่ทำให้เกิดการลดลงของข้อมูล เนื่องจากจำนวนข้อมูลในแต่ละแบนด์ย่อย เมื่อนำมารวมกันแล้วจะเท่ากับข้อมูลของภาพต้นแบบ แต่จะมีข้อคืออยู่ที่ข้อมูล

ในแต่ละแบนด์ย่อยนั้นสามารถนำไปเข้ารหัสได้มีประสิทธิภาพการบีบอัดข้อมูลดีกว่าการเข้ารหัสด้วยภาพต้นแบบโดยตรง และในงานวิทยานิพนธ์นี้ได้เลือกใช้อัลกอริทึม Set Partitioning in Hierarchical Tree (SPIHT) มาใช้ในการเข้ารหัส ซึ่งให้ผลของการบีบอัดข้อมูลภาพได้ดีกว่า JPEG (AI, 2000)

### 3.4 การวัดประสิทธิภาพการบีบอัดข้อมูลภาพ

ในการบีบอัดข้อมูลภาพแบบมีการสูญเสีย ประสิทธิภาพของเทคนิคต่างๆ ไม่สามารถสรุปได้จากอัตราการบีบอัดข้อมูลเพียงอย่างเดียว แต่ต้องพิจารณาถึงความสูญเสียข้อมูลที่เกิดขึ้นว่ายอมรับได้หรือไม่ เกณฑ์ที่ใช้ในการวัดประสิทธิภาพของการบีบอัดข้อมูลภาพสามารถแบ่งออกเป็น 2 ประเภทดังนี้

#### 3.4.1 การวัดเชิงปริมาณ

การวัดค่าความผิดพลาดที่เกิดจากการบีบอัดข้อมูลสามารถกระทำได้หลายวิธีการคือ

##### 3.4.1.1 ค่าความผิดพลาดกำลังสองเฉลี่ย (mean square error: MSE)

ค่าความผิดพลาดกำลังสองเฉลี่ยของการบีบอัดข้อมูลภาพขนาด  $M \times N$  หาได้ดังนี้

$$MSE = \frac{1}{M * N} \sum_{x=1}^M \sum_{y=1}^N \left[ f(x, y) - f_i(x, y) \right]^2 \quad (3.8)$$

โดยที่  $M$  คือจำนวนพิกเซลตามความกว้างของภาพ

$N$  คือจำนวนพิกเซลตามความสูงของภาพ

$f(x,y)$  คือค่าของพิกเซลที่ตำแหน่ง  $(x,y)$  ของภาพต้นแบบ

$f_i(x,y)$  คือค่าของพิกเซลที่ตำแหน่ง  $(x,y)$  ของภาพที่สร้างกลับคืนมาใหม่

ค่าความผิดพลาดกำลังสองเฉลี่ยถ้ามีค่าน้อย แสดงว่าภาพที่ได้จากการสร้างกลับจากการบีบอัดข้อมูลมีความผิดเพี้ยนไปจากภาพต้นแบบน้อย และถ้าค่าความผิดพลาดกำลังสองเฉลี่ยมีค่ามาก แสดงว่าภาพที่ได้จากการสร้างกลับจากการบีบอัดข้อมูลมีความผิดเพี้ยนไปจากภาพต้นแบบมาก

### 3.4.1.2 อัตราส่วนของสัญญาณต่อสัญญาณรบกวนสูงสุด (peak-signal-to-noise ratio: PSNR)

ค่าอัตราส่วนของสัญญาณต่อสัญญาณรบกวนสูงสุดของการบีบอัดข้อมูลภาพขนาด  $M \times N$  ของภาพระดับความเทา (gray scale) หาได้ดังนี้

$$\text{PSNR} = 10 \log \frac{255^2}{\text{MSE}} \quad (\text{dB}) \quad (3.9)$$

ค่าอัตราส่วนของสัญญาณต่อสัญญาณรบกวนสูงสุดถ้ามีค่าน้อย แสดงว่าภาพที่ได้จากการสร้างกลับจากการบีบอัดข้อมูลมีความผิดเพี้ยนไปจากภาพต้นแบบมาก และถ้าอัตราส่วนของสัญญาณต่อสัญญาณรบกวนสูงสุดมีค่ามาก แสดงว่าภาพที่ได้จากการสร้างกลับจากการบีบอัดข้อมูลมีความผิดเพี้ยนไปจากภาพต้นแบบน้อย

### 3.4.1.3 อัตราบิต (bit rate)

อัตราบิตคือค่าเฉลี่ยของจำนวนบิตต่อพิกเซล (bits per pixel: bpp) ของภาพที่ถูกลดข้อมูลโดยสามารถคำนวณได้จากอัตราส่วนระหว่างจำนวนบิตของภาพที่ผ่านการบีบอัดข้อมูลทั้งหมดต่อจำนวนพิกเซลของภาพต้นแบบ หาได้ดังนี้

$$\text{Bit rate} = \frac{\text{จำนวนบิตทั้งหมดของภาพที่ผ่านการบีบอัดข้อมูล}}{\text{จำนวนพิกเซลทั้งหมดของภาพต้นแบบ}} \quad (3.10)$$

### 3.4.1.4 อัตราการบีบอัดข้อมูลภาพ (compression ration: CR)

อัตราการบีบอัดข้อมูลภาพคืออัตราส่วนระหว่างจำนวนบิตข้อมูลภาพต้นแบบและจำนวนบิตข้อมูลของภาพที่ผ่านการบีบอัดข้อมูลภาพ หาได้ดังนี้

$$\text{CR} = \frac{\text{จำนวนบิตทั้งหมดของภาพต้นแบบ}}{\text{จำนวนบิตทั้งหมดของภาพที่ผ่านการบีบอัดข้อมูล}} \quad (3.11)$$

## 3.4.2 การวัดเชิงคุณภาพ

เป็นวิธีการพื้นฐานแต่ให้ผลที่น่าเชื่อถือมาก โดยใช้สายตาของผู้ที่ไม่เกี่ยวข้องทำการตัดสินคุณภาพของภาพในฐานะที่เป็นผู้ใช้ภาพในการวิจัย และผู้ใช้เครื่องมือในการถ่ายภาพ ที่มีความสามารถในการสังเกตความบกพร่องเล็กน้อยที่ผู้ใช้ทั่วไปอาจมองข้าม

ในงานวิทยานิพนธ์นี้ได้ใช้วิธีการประเมินผลจากแบบสอบถามเพื่อประเมินผล ด้วยวิธีวัดความเหมือนระหว่างภาพต้นแบบกับภาพที่ผ่านการบีบอัดข้อมูล จากกลุ่มเป้าหมายที่ปฏิบัติงานเกี่ยวข้องกับกล้องจุลทรรศน์อิเล็กตรอนคือ อาจารย์ นักวิทยาศาสตร์ นักวิจัย ช่างภาพการแพทย์ เป็นต้น ผลการประเมินจะนำมาวิเคราะห์ทางสถิติ

### 3.5 สรุป

การบีบอัดข้อมูลภาพในปัจจุบัน มีหลายวิธีที่แตกต่างกันไป สำหรับงานวิทยานิพนธ์นี้เลือกการบีบอัดภาพแบบมีการสูญเสีย แบบการเข้ารหัสแบบด้อย เนื่องจากให้อัตราการบีบอัดข้อมูลที่สูงกว่าการบีบอัดข้อมูลแบบไม่มีการสูญเสีย และเป็นที่ยอมรับใช้งานในปัจจุบัน แต่ในขณะเดียวกันก็ต้องยอมรับกับคุณภาพของภาพจากการบีบอัดข้อมูลที่แย่งหรือเกิดความผิดเพี้ยนของภาพขึ้น ดังนั้นในการบีบอัดข้อมูลภาพ EM จะต้องมีการหาอัตราบิดหรือหาอัตราการบีบอัดข้อมูลที่เหมาะสม เพื่อให้ภาพที่ได้จากการบีบอัดข้อมูลภาพสามารถนำไปวิเคราะห์ข้อมูลต่อไปได้

## บทที่ 4

### การแปลงเวฟเล็ต

#### 4.1 บทนำ

บทนี้จะกล่าวถึงความเป็นมาของการแปลงเวฟเล็ต (wavelet transform: WT) ทฤษฎีพื้นฐานของการแปลงเวฟเล็ต การแปลงเวฟเล็ตแบบดิสครีต (discrete wavelet transform: DWT) แบบ 1 มิติ ด้วยหลักการฟิลเตอร์แบงก์ (filter bank) และการแปลง DWT แบบ 2 มิติ ด้วยหลักการแยกองค์ประกอบเป็นแบนด์ย่อย (subband decomposition)

#### 4.2 ความเป็นมาของการแปลงเวฟเล็ต

การแปลงฟูรีเยร์ (fourier transform: FT) เป็นการแปลงสัญญาณ  $f(t)$  จากโดเมนเวลาไปเป็นสัญญาณ  $f(\omega)$  ซึ่งอยู่ในโดเมนความถี่ หรือเรียกว่าเป็นสเปกตรัม (spectrum) ของสัญญาณ  $f(t)$  นั้นเอง โดยหาได้จากสมการคือ

$$f(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt \quad (4.1)$$

จากสมการจะเห็นได้ว่าการแปลงตลอดย่านความถี่  $(-\infty, \infty)$  ดังนั้นเมื่อเกิดการเปลี่ยนแปลงของสัญญาณอินพุตที่ระยะเวลาใดๆ ก็จะส่งผลกระทบต่อตลอดย่านของสเปกตรัม  $f(\omega)$  ด้วย นอกจากนี้การใช้ FT ยังไม่สามารถเลือกทำการวิเคราะห์สัญญาณที่เวลาใดเวลาหนึ่งได้ หรือเฉพาะช่วงความถี่ใดความถี่หนึ่งได้ ดังนั้น FT จึงให้ข้อมูลที่มากเกินไปจนจำเป็น นอกจากนั้นในทางปฏิบัติสัญญาณอินพุตจะมีการเปลี่ยนแปลงที่ไม่คงที่ (non stationary signal) ซึ่งถ้าใช้ FT กับสัญญาณประเภทนี้จะทำให้สิ้นเปลืองเวลาในการคำนวณ ดังนั้น FT จึงเหมาะสมกับสัญญาณที่มีการเปลี่ยนแปลงคงที่ (stationary signal)

Gabor Transform (GT) เป็นการแปลงแบบแรกที่นำมาแก้ปัญหาเรื่องการวิเคราะห์สัญญาณเฉพาะช่วงเวลาและช่วงความถี่ที่ต้องการ โดยกำหนดผ่านฟังก์ชันหน้าต่างต่าง (window function: wf) ซึ่ง wf นี้จะใช้เป็น Gaussian Function เท่านั้น และมีสมการคือ

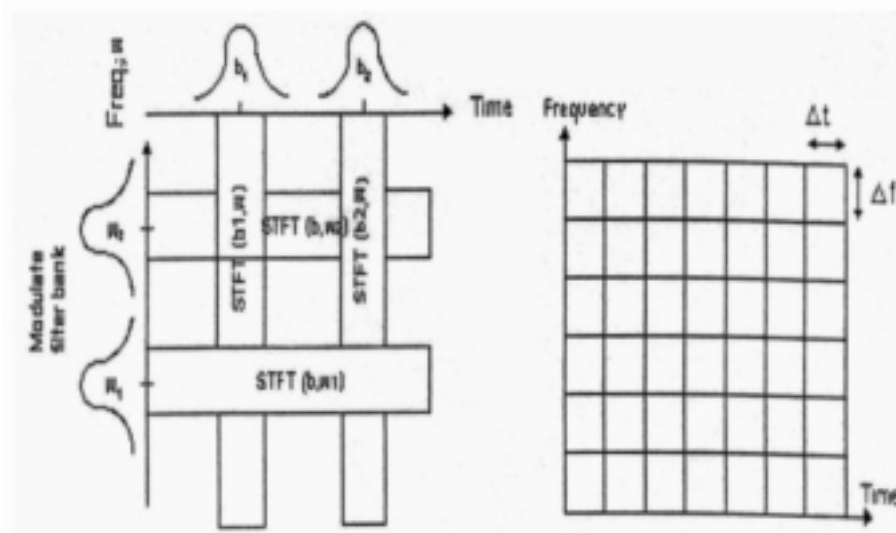
$$g_{\alpha}(t) = \frac{1}{2\sqrt{\pi\alpha}} e^{-\frac{t^2}{4\alpha}} \tag{4.2}$$

$$(G_{\beta}^{\alpha}f)(\omega) = \int_{-\infty}^{\infty} (e^{-i\omega t}f(t))g_{\alpha}(t-\beta)dt \tag{4.3}$$

จากสมการที่ 4.2 เป็น Gaussian Window Function และสมการที่ 4.3 เป็นสมการ GT สามารถกำหนดตำแหน่งเวลาการวิเคราะห์สัญญาณด้วยการกำหนดที่พารามิเตอร์  $\beta$  และเลือกช่วงความถี่ด้วยการกำหนดพารามิเตอร์  $\alpha$  และต่อมาได้มีการพัฒนาเป็น Short Time Fourier Transform (STFT) ซึ่งสามารถทำให้เลือกใช้ wf ได้ ทำให้เกิดความยืดหยุ่นเพิ่มมากขึ้น เพราะสามารถเลือก wf ให้เหมาะสมกับสัญญาณอินพุตที่จะวิเคราะห์ได้ดีขึ้น และมีสมการคือ

$$(G_{\alpha}f)(\omega) = \int_{-\infty}^{\infty} (e^{-i\omega t}f(t))\overline{w(t-\beta)}dt \tag{4.4}$$

สมการที่ 4.4 เป็นสมการ STFT โดยที่ตำแหน่งเวลาในการวิเคราะห์สัญญาณด้วยการกำหนดที่พารามิเตอร์  $\beta$  และช่วงความถี่ในการวิเคราะห์กำหนดด้วยความถี่ของ wf ที่ใช้ แต่จะสังเกตได้ว่า ทั้ง GT และ STFT จะมีลักษณะของช่วงเวลาและช่วงความถี่ดังภาพที่ 4.1



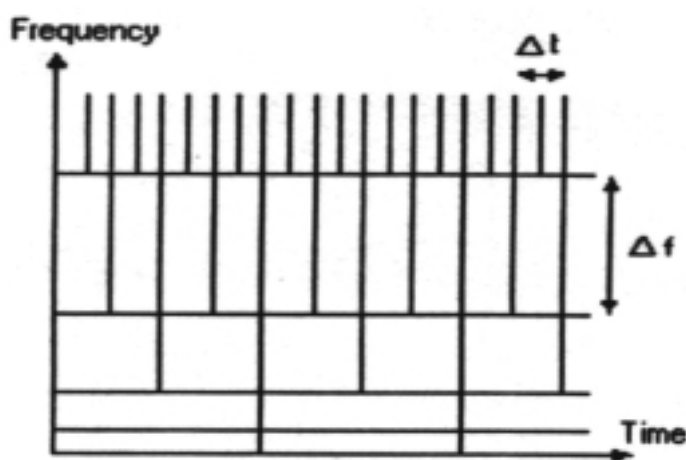
ภาพที่ 4.1 แสดงลักษณะของระนาบเวลาความถี่แบบคงที่ (ชัยชาญ มัคคูน, 2539)

เนื่องจากสัญญาณในทางปฏิบัติ นั้น สัญญาณความถี่สูงจะมีช่วงคาบเวลาที่แคบในขณะที่ สัญญาณความถี่ต่ำจะมีช่วงคาบเวลาที่กว้าง ดังนั้น GT และ STFT อาจทำให้เกิดปัญหาการแปลง ข้อมูลให้ออกมามีจำนวนข้อมูลมาก หรือน้อยเกินไป ดังนั้นช่วงเวลาที่ควรจะต้องมีการเปลี่ยนแปลง สัมพันธ์กับช่วงความถี่ที่จะทำการวิเคราะห์ โดยที่ช่วงความถี่ที่สูงจะต้องมีช่วงเวลาในการวิเคราะห์ น้อยลง และในขณะที่การวิเคราะห์ช่วงความถี่ต่ำจะต้องใช้ช่วงเวลาที่ยาวกว่าในการวิเคราะห์ สัญญาณ จากเหตุผลการแก้ปัญหาดังกล่าวทำให้มีการพัฒนาการแปลงเวฟเลต ซึ่งจะมีการแปลงผ่าน wf แต่ wf ที่ใช้สามารถเลื่อนตำแหน่งและปรับเปลี่ยนช่วงเวลาให้เหมาะสมกับช่วงความถี่ที่จะ วิเคราะห์ได้ โดยเรียก wf แบบนี้ว่าฟังก์ชันเวฟเลต (wavelet function) และมีสมการคือ

$$(W_g f)(a, b) = |a|^{-0.5} \int_{-\infty}^{\infty} f(t) g^* \left( \frac{t-b}{a} \right) dt \quad (4.5)$$

เมื่อ  $g(t)$  คือเวฟเลตแม่ (Mother wavelet) โดยที่  $*$  หมายถึงค่าสังยุคเชิงซ้อน

$g(t)$  จะถูกเลื่อนตำแหน่ง (เปลี่ยนช่วงเวลา) และถูกสเกล (เปลี่ยนความถี่) ด้วยพารามิเตอร์  $a$  และ  $b$  ตามลำดับ โดยที่ช่วงเวลาและความถี่ในการแปลงจะสัมพันธ์ตามที่ได้กล่าวมาแล้ว ดังแสดง ในภาพที่ 4.2 ซึ่งจะเห็นว่า ระยะเวลาและความถี่ จะมีลักษณะการเปลี่ยนแปลงช่วงเวลาและ ความถี่ที่สัมพันธ์กันคือ ในช่วงความถี่สูงจะใช้ช่วงเวลาในการวิเคราะห์น้อย แต่ในช่วงความถี่ต่ำจะ ใช้ช่วงเวลาในการวิเคราะห์มาก ซึ่งก็เหมาะสมกับสัญญาณในทางปฏิบัติ และนอกจากนี้  $g(t)$  ยังมี ได้หลายแบบจึงทำให้การแปลงเวฟเลตมีความยืดหยุ่นและสามารถเลือกใช้ให้เหมาะสมกับงานที่นำไป ใช้ได้เป็นอย่างดี



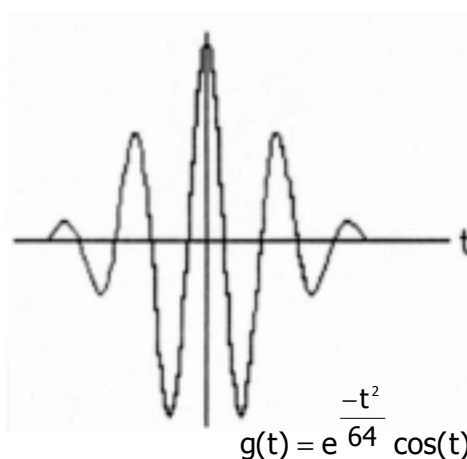
ภาพที่ 4.2 แสดงลักษณะของระยะเวลาและความถี่ ของ WT (ชัยชาญ มัคคูน, 2539)

### 4.3 ทฤษฎีพื้นฐานของการแปลงเวฟเล็ต

ทฤษฎีเวฟเล็ต (wavelet theory) เป็นคณิตศาสตร์ที่ใช้อธิบายการสร้างโมเดลของสัญญาณระบบ มีลักษณะและทฤษฎีพื้นฐานของการแปลงเวฟเล็ต ดังนี้

#### 4.3.1 ลักษณะของเวฟเล็ต

เวฟเล็ตประกอบขึ้นด้วยเซตของสัญญาณเฉพาะที่มีลักษณะเป็นคลื่นเล็กๆ โดยลักษณะของเวฟเล็ตจะเป็นคลื่นที่มีการเปลี่ยนแปลงอย่างต่อเนื่อง และขนาดของคลื่นจะลดลงสู่ศูนย์กลางอย่างรวดเร็วทั้งสองด้าน เป็นดังภาพที่ 4.3 ซึ่งภาพที่ 4.3 นี้เป็นตัวอย่างเวฟเล็ตชนิดหนึ่งเรียกว่าเวฟเล็ตแบบ Morlet จะสังเกตได้ว่ามีคุณลักษณะเป็นไปตามที่กล่าวมาทั้ง 2 เงื่อนไข คือมีการออสซิลเลตด้วยรูปทรงของคลื่นไซน์หรือโคไซน์ ดังภาพที่ 4.4 และการลดลงอย่างรวดเร็วตามลักษณะ  $wf$  ดังภาพที่ 4.5

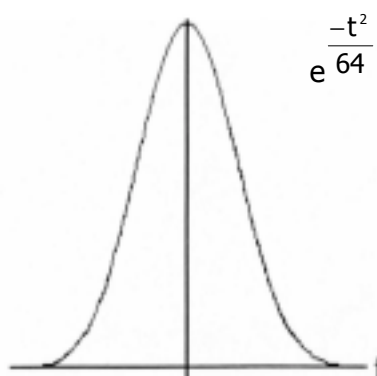


ภาพที่ 4.3 ตัวอย่างแสดงลักษณะเวฟเล็ต



ภาพที่ 4.4 ลักษณะของสัญญาณรูปคลื่น โคไซน์





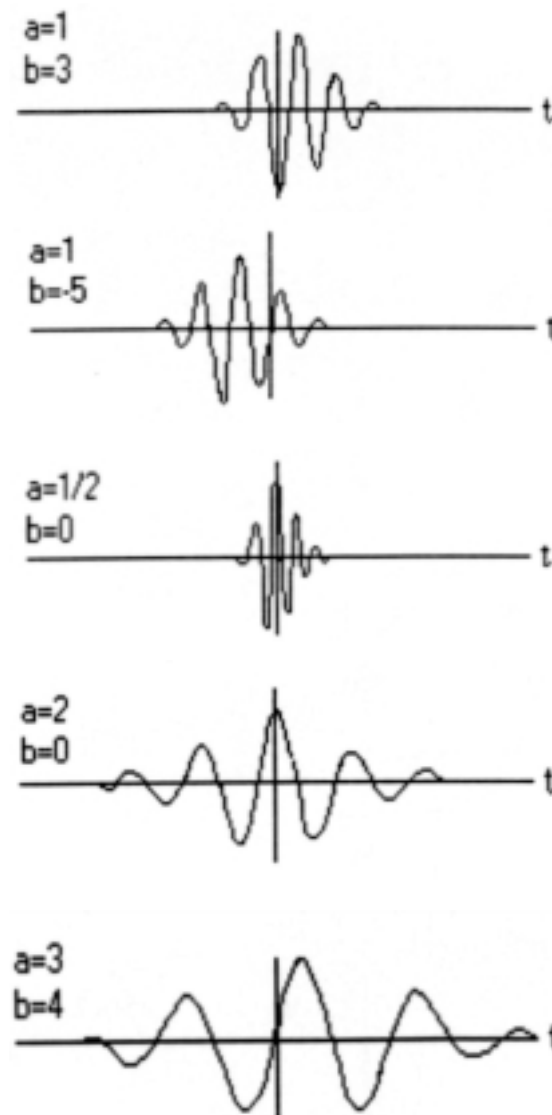
ภาพที่ 4.5 แสดงลักษณะของฟังก์ชันหน้าต่าง

การนำเวฟเล็ตหลายๆ อันมารวมกันเป็นเซต เพื่อจะใช้ในการอธิบายสัญญาณใดๆ โดยที่คลื่นเวฟเล็ตแต่ละตัวมีโครงสร้างมาจากฟังก์ชันเดียวกัน ซึ่งฟังก์ชันนี้จะเป็นเวฟเล็ตต้นกำเนิดที่เรียกว่าเวฟเล็ตแม่ โดยคลื่นแต่ละคลื่นภายในเซตของเวฟเล็ตจะเกิดจากการสเกล การสเกลในที่นี้หมายถึง การบีบเข้า (compressing) หรือการขยายออก (dilation) โดยใช้พารามิเตอร์  $a$  แทนการบีบหรือขยายอันเป็นการเปลี่ยนความถี่ และการเลื่อนตำแหน่งของเวฟเล็ตแม่ โดยใช้พารามิเตอร์  $b$  แทนการเลื่อนตำแหน่งของการเกิดคลื่นบนแกนเวลา เป็นดังภาพที่ 4.6 แสดงให้เห็นถึงเวฟเล็ตที่ถูกสเกลและเลื่อนตำแหน่งไปที่  $a$  และ  $b$  ต่างๆ กัน โดยที่ยังคงอยู่ในเซตของเวฟเล็ตที่สัมพันธ์กับเวฟเล็ตแม่ในภาพที่ 4.3 นอกจากนี้เวฟเล็ตที่ถูกสเกลจะต้องถูกนอร์มอลไลซ์ด้วยค่า  $a^{-0.5}$  เสมอ เพื่อให้เวฟเล็ตที่ถูกสเกลไปแล้วยังคงมีพลังงานเช่นเดียวกับเวฟเล็ตแม่

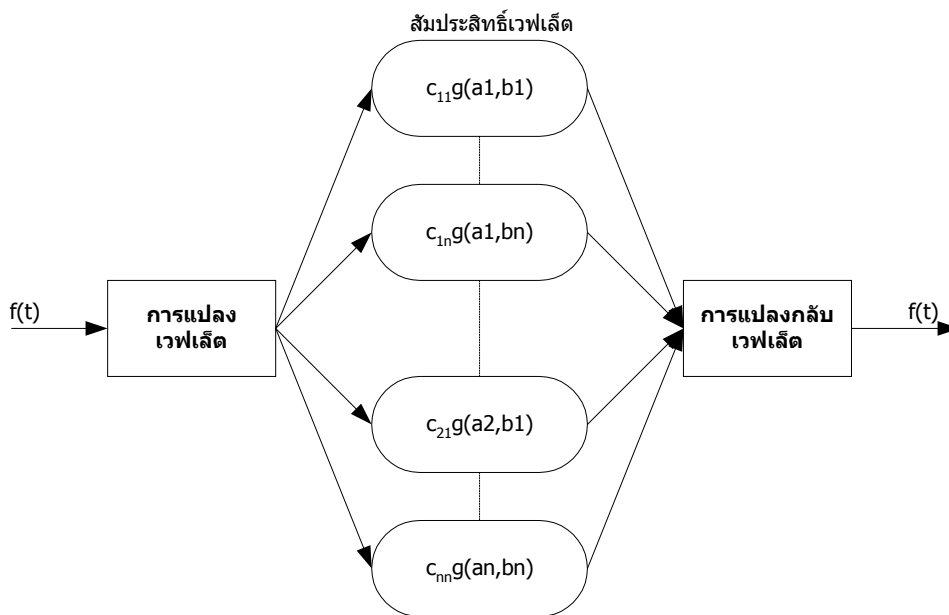
#### 4.3.2 ทฤษฎีพื้นฐานเวฟเล็ตในการสเกลและการเลื่อนตำแหน่ง

การอธิบายสิ่งหนึ่งสิ่งใดในทฤษฎีเวฟเล็ตจะเปรียบเสมือนกับการแตกสิ่งเหล่านั้นออกเป็นส่วนประกอบเล็กๆ โดยที่แต่ละส่วนประกอบนั้นจะอยู่ในรูปของเวฟเล็ตที่ถูกสเกลและเลื่อนตำแหน่ง และมีค่าถ่วงน้ำหนักต่างๆ กัน คุณอยู่กับเวฟเล็ตแต่ละตัว ดังนั้นจึงเปรียบเสมือนว่าคลื่นเวฟเล็ตเป็นฟังก์ชันพื้นฐาน (basis function, bf) ในการสร้างสัญญาณใดๆ โดยขบวนการแตกกระจายชิ้นส่วนนี้เรียกว่า การแยกองค์ประกอบด้วยเวฟเล็ต (wavelet decomposition) ซึ่งก็คือขั้นตอนการแปลงเวฟเล็ตนั่นเอง ในทำนองเดียวกัน การสร้างกลับจากองค์ประกอบด้วยเวฟเล็ต (wavelet reconstruction) ก็คือการแปลงกลับเวฟเล็ต (inverse wavelet transform: IWT) โดยจะเป็นการนำเอาส่วนประกอบย่อยๆ เหล่านี้มาเรียงต่อกันเพื่อประกอบเป็นสัญญาณเดิม ดังภาพที่ 4.7 แสดงการแยกองค์ประกอบของสัญญาณ  $f(t)$  ออกเป็นคลื่นเวฟเล็ตที่ความถี่และตำแหน่งต่างกัน

จำนวน  $n \times n$  ตัว และทำการสร้างกลับจากองค์ประกอบด้วยเวฟเล็ต จากคลื่นเวฟเล็ตที่ความถี่และตำแหน่งต่างกันจำนวน  $n \times n$  ตัว

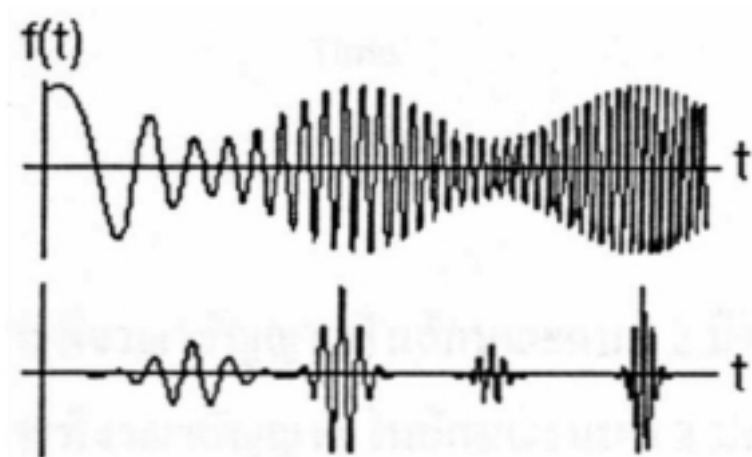


ภาพที่ 4.6 แสดงลักษณะของเวฟเล็ตแม่ที่ถูกสเกลและเลื่อนตำแหน่งไปที่ค่า  $a$  และ  $b$  ต่างๆ (เกศกนก กฤตยาภาศิริวัฒน์ 2543)



ภาพที่ 4.7 แสดงการแยกองค์ประกอบของเวฟเล็ตและการสร้างกลับจากองค์ประกอบด้วยเวฟเล็ต

และแสดงการดึงสัญญาณ  $f(t)$  ที่ตำแหน่งเวลาต่างๆ กัน ออกมาจากสัญญาณ  $f(t)$  ที่ประกอบด้วยคลื่นหลายความถี่ เป็นดังภาพที่ 4.8



ภาพที่ 4.8 แสดงลักษณะของสัญญาณที่มีหลายความถี่ที่ตำแหน่งเวลาต่างกัน (ชัยชาญ มัคคูน, 2539)

จากการที่สามารถทำการสร้างเซตของเวฟเล็ตได้จากการสเกลและการเลื่อนตำแหน่งของเวฟเล็ตแม่นั้น การสเกลและการเลื่อนตำแหน่งดังกล่าวจะถูกเรียกรวมกันว่า affine operation ซึ่งจะสามารถเขียนเป็นสมการคณิตศาสตร์ในการส่ง (mapping) ตัวแปรอิสระ  $t$  ไปยังตัวแปรอิสระตัวใหม่  $t'$  ดังแสดงได้ตามสมการคือ

$$t' = \frac{t-b}{a} \quad (4.6)$$

สามารถเขียนสมการสำหรับการแปลงเวฟเล็ตแบบต่อเนื่อง (continuous wavelet transform: CWT) ที่ใช้ในการกระจายสัญญาณ  $f(t)$  ด้วยเวฟเล็ตแม่  $g(t)$  คือ

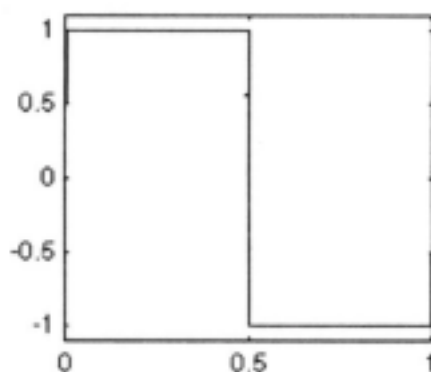
$$(W_g f)(a,b) = |a|^{-0.5} \int_{-\infty}^{\infty} f(t)g^*\left(\frac{t-b}{a}\right)dt \quad (4.7)$$

โดยที่  $(W_g f)(a,b)$  คือสัมประสิทธิ์ของเวฟเล็ต (wavelet coefficient)

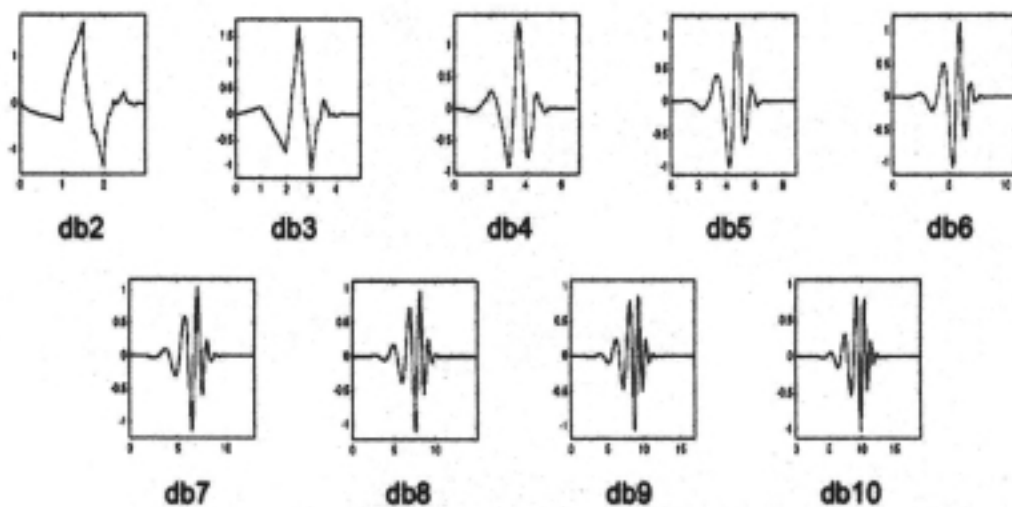
a คือ พารามิเตอร์ของการสเกล

b คือ พารามิเตอร์ของการเลื่อนตำแหน่ง

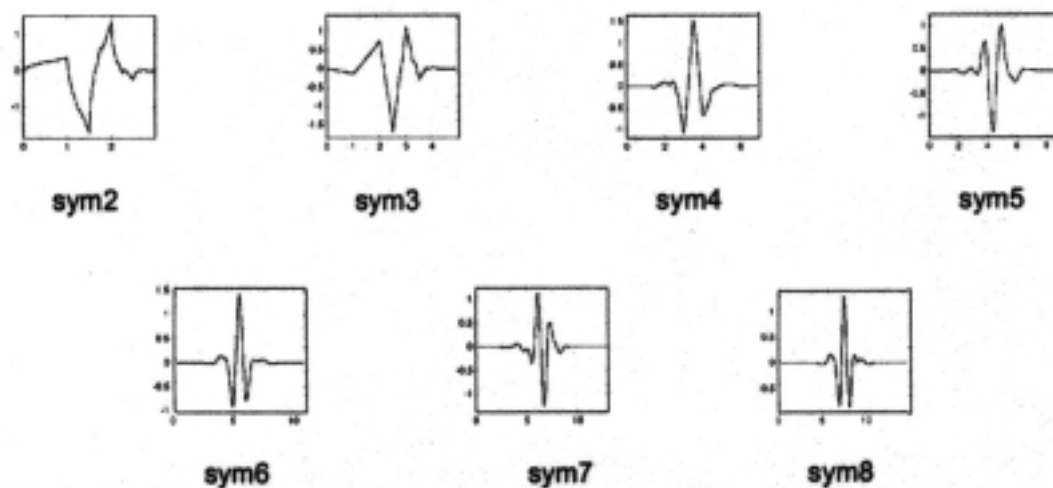
เวฟเล็ตแม่  $g(t)$  ยังมีได้หลายรูปแบบ ดังภาพที่ 4.9 ที่แสดงตัวอย่างของตระกูลเวฟเล็ตแม่ชนิดต่างๆ จึงทำให้การแปลงเวฟเล็ตมีความยืดหยุ่นและสามารถนำไปประยุกต์ใช้กับงานได้อย่างดี



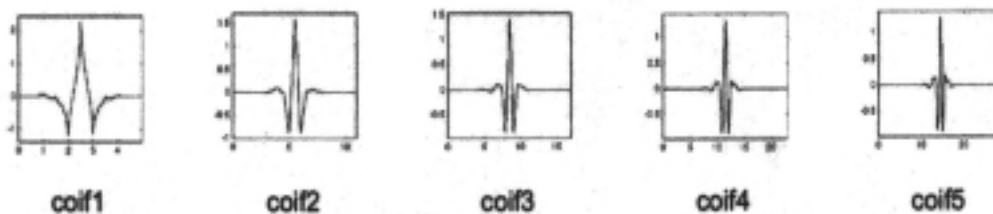
ภาพที่ 4.9.a ตระกูล Haar



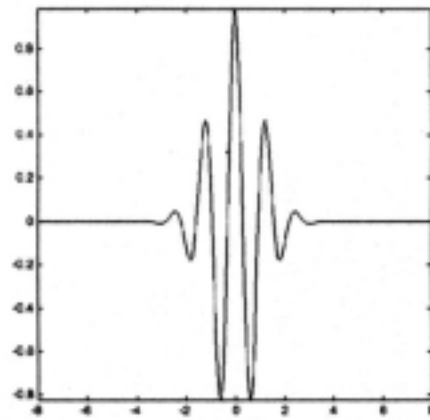
ภาพที่ 4.9.b ตระกูล Daubechies



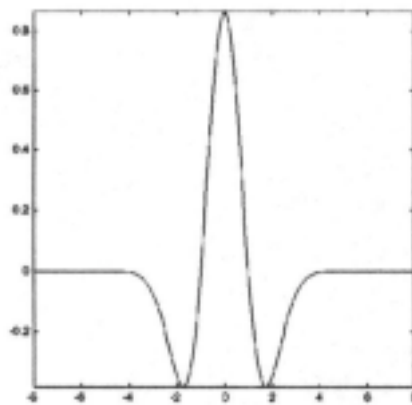
ภาพที่ 4.9.c ตระกูล Symlets



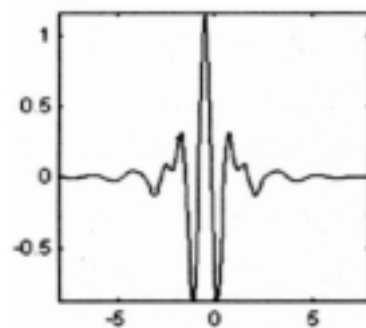
ภาพที่ 4.9.d ตระกูล Coiflets



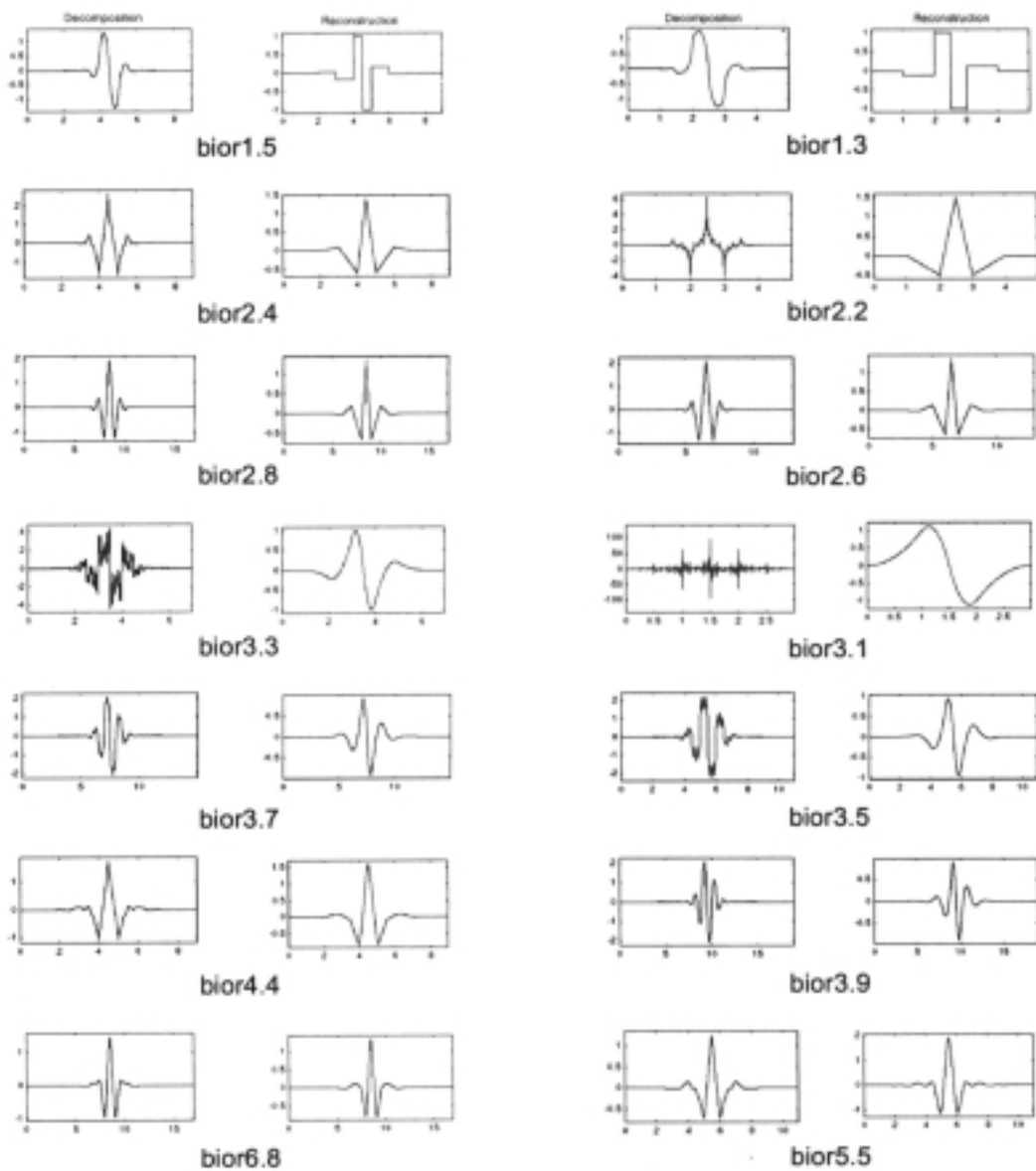
ภาพที่ 4.9.e ตระกูล Morlet



ภาพที่ 4.9.f ตระกูล Mexican hat



ภาพที่ 4.9.g ตระกูล Meyer



ภาพที่ 4.9.h ตระกูล Biorthogonal

ภาพที่ 4.9 ตระกูลเวฟเล็ตแม่ (Michel, Yves, Georges, and Jean, 1996)

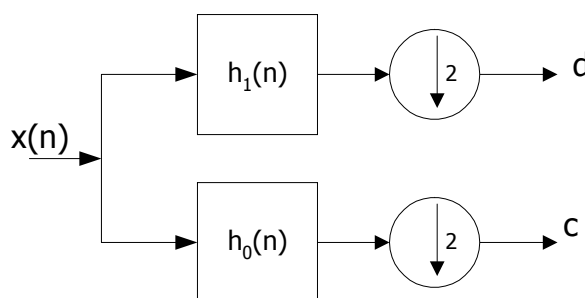
#### 4.4 การแปลงเวฟเล็ดแบบดิสครีทใน 1 มิติ

การแปลงเวฟเล็ดแบบดิสครีทแบบ 1 มิติ โดยใช้หลักการของฟิลเตอร์แบงก์ (filter bank) (Olivier, and Martin, 1991) มีหลักการวิเคราะห์ (analysis) สัญญาณด้วยฟิลเตอร์แบงก์เป็นดังภาพที่ 4.10 กำหนดให้  $x(n)$  คือข้อมูลอินพุต  $h_0(n)$  คือตัวฟิลเตอร์ที่กรองความถี่ต่ำ และ  $h_1(n)$  คือตัวฟิลเตอร์ที่กรองความถี่สูง และมีแผนภาพการลดข้อมูลลงครึ่งหนึ่ง (downsampling) เป็นดังภาพที่ 4.11 สามารถเขียนเป็น สมการการวิเคราะห์ได้ดังสมการที่ 4.8 และ 4.9

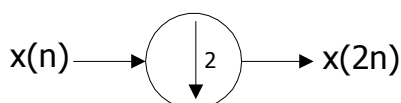
$$C(k) = \sum_m h_0(2k - m)x(m) \quad (4.8)$$

$$d(k) = \sum_m h_1(2k - m)x(m) \quad (4.9)$$

เมื่อ  $c$  คือ สัมประสิทธิ์เวฟเล็ดขององค์ประกอบความถี่ต่ำ  
 $d$  คือ สัมประสิทธิ์เวฟเล็ดขององค์ประกอบความถี่สูง



ภาพที่ 4.10 แผนภาพการวิเคราะห์สัญญาณด้วยฟิลเตอร์แบงก์แบบ 1 มิติ

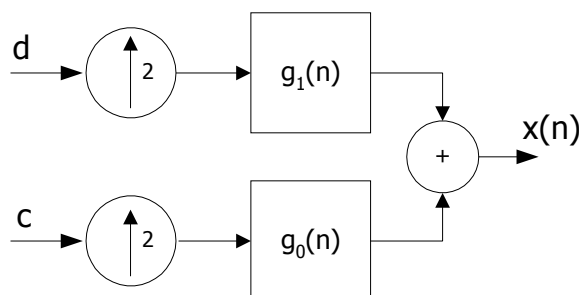


ภาพที่ 4.11 แผนภาพการลดข้อมูลลงครึ่งหนึ่ง

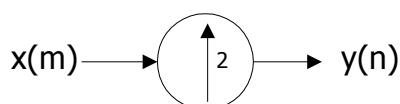
และมีหลักการสังเคราะห์ (synthesis) สัญญาณด้วยฟิลเตอร์แบงก์เป็นดังภาพที่ 4.12 กำหนดให้  $g_0(n)$  คือ ตัวฟิลเตอร์ที่กรองความถี่ต่ำ และ  $g_1(n)$  คือ ตัวฟิลเตอร์ที่กรองความถี่สูง และมีแผนภาพการเพิ่มข้อมูลขึ้นเป็นสองเท่า (upsampling) เป็นดังภาพที่ 4.13 สามารถเขียนเป็นสมการการสังเคราะห์ได้ดังสมการที่ 4.10



$$x(n) = \sum_m c(m)g_0(n-2m) + \sum_m d(m)g_1(n-2m) \quad (4.10)$$



ภาพที่ 4.12 แผนภาพการสังเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์ แบบ 1 มิติ



$$y(n) = \begin{cases} x(l), & n = 2l \\ 0, & \text{อื่นๆ} \end{cases}$$

ภาพที่ 4.13 แผนภาพการเพิ่มข้อมูลขึ้นเป็นสองเท่า

การวิเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์แบบหลายแบนด์ย่อย สามารถทำได้โดยการนำเอา การวิเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์แบบ 1 มิติแบบ 1 แบนด์ย่อยมาประยุกต์ใช้ได้ดังสมการที่ 4.11 และ 4.12 และมีแผนภาพการวิเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์แบบหลายแบนด์ย่อย ดังภาพที่ 4.14 ซึ่งภาพที่ 4.14 เป็นการวิเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์แบบ 3 แบนด์ย่อย

$$C_j(k) = \sum_m h_0(2k-m)c_{j+1}(m) \quad (4.11)$$

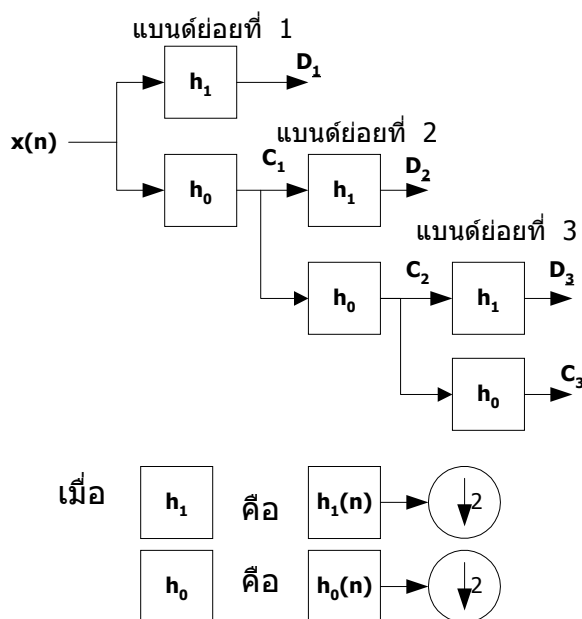
$$d_j(k) = \sum_m h_1(2k-m)c_{j+1}(m) \quad (4.12)$$

เมื่อ  $c_j$  คือ สัมประสิทธิ์เวฟเลตขององค์ประกอบความถี่ต่ำ ที่แบนด์ย่อยที่  $j$   
 $d_j$  คือ สัมประสิทธิ์เวฟเลตขององค์ประกอบความถี่สูง ที่แบนด์ย่อยที่  $j$

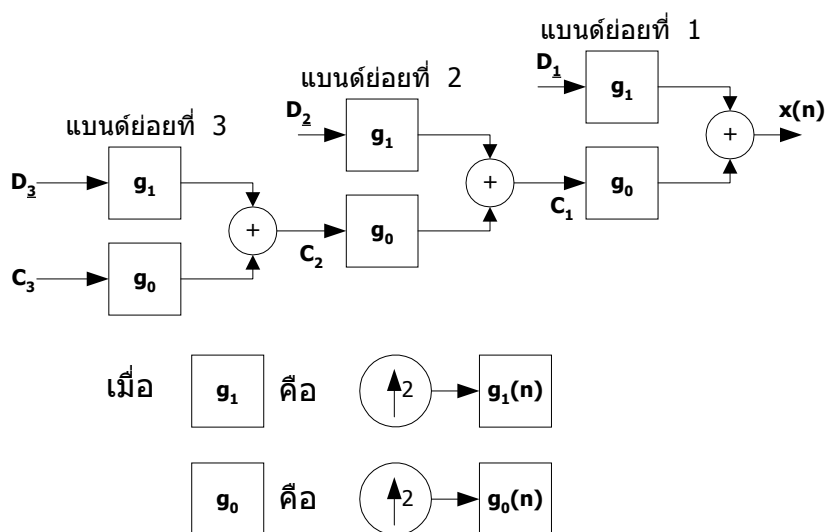
การสังเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์แบบหลายแบนด์ย่อย สามารถทำได้โดยการนำเอา การสังเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์แบบ 1 มิติ แบบ 1 แบนด์ย่อย มาประยุกต์ใช้ได้ ดังสมการที่ 4.13 และ 4.14 และมีแผนภาพสังเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์แบบหลายแบนด์ย่อย ดังภาพที่ 4.15 ซึ่งภาพที่ 4.15 เป็นการสังเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์แบบ 3 แบนด์ย่อย

$$x(n) = \sum_m c_1(m)g_0(n-2m) + \sum_m d_1(m)g_1(n-2m) \tag{4.13}$$

$$c_{j+1}(k) = \sum_p c_j(p)g_0(k-2p) + \sum_p d_j(p)g_1(k-2p) \tag{4.14}$$

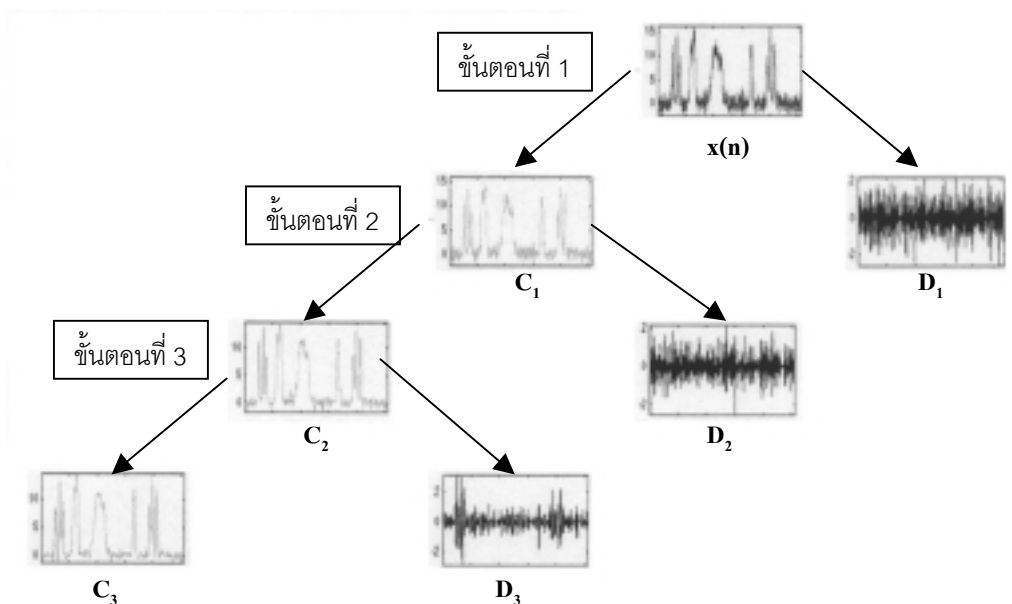


ภาพที่ 4.14 แผนภาพการวิเคราะห์สัญญาณด้วยฟิลเตอร์แบงก์ แบบ 3 แบนด์ย่อย



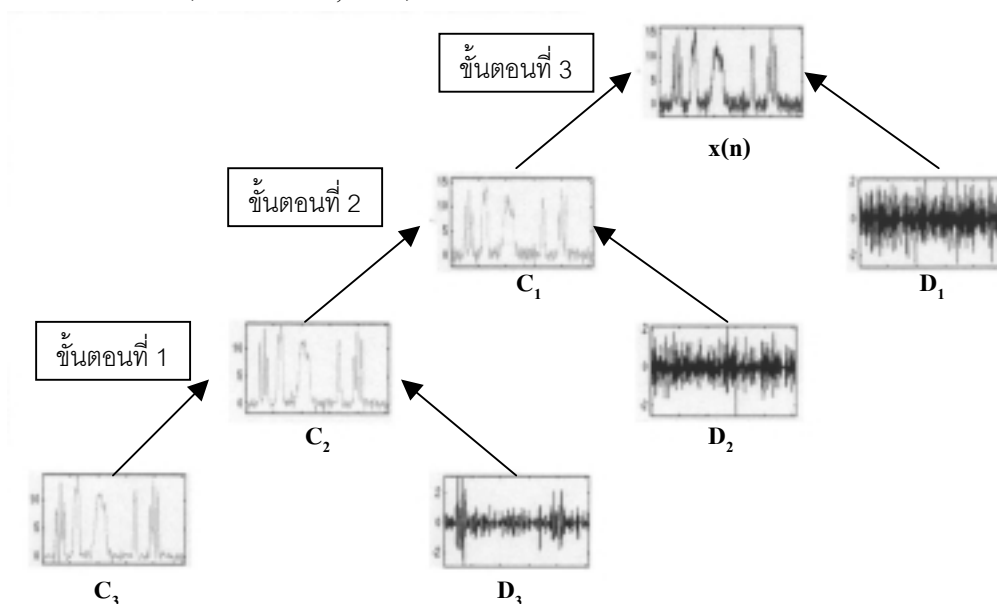
ภาพที่ 4.15 แผนภาพการสังเคราะห์สัญญาณด้วยฟิลเตอร์แบงก์ แบบ 3 แบนด์ย่อย

มีตัวอย่างการวิเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์แบบ 3 แบนด์ย่อยเป็นดังภาพที่ 4.16 และมีตัวอย่างการสังเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์แบบ 3 แบนด์ย่อยเป็นดังภาพที่ 4.17



ภาพที่ 4.16 ตัวอย่างการวิเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์แบบ 3 แบนด์ย่อย

(Michel et al., 1996)

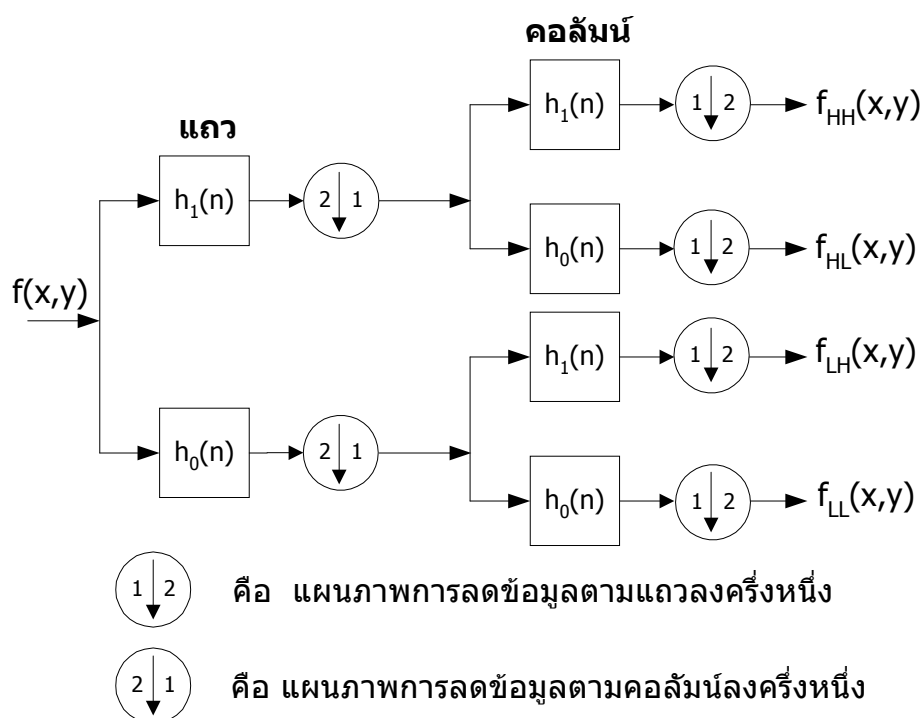


ภาพที่ 4.17 ตัวอย่างการสังเคราะห์สัญญาณด้วยฟิลเตอร์เบงค์แบบ 3 แบนด์ย่อย

(Michel et al., 1996)

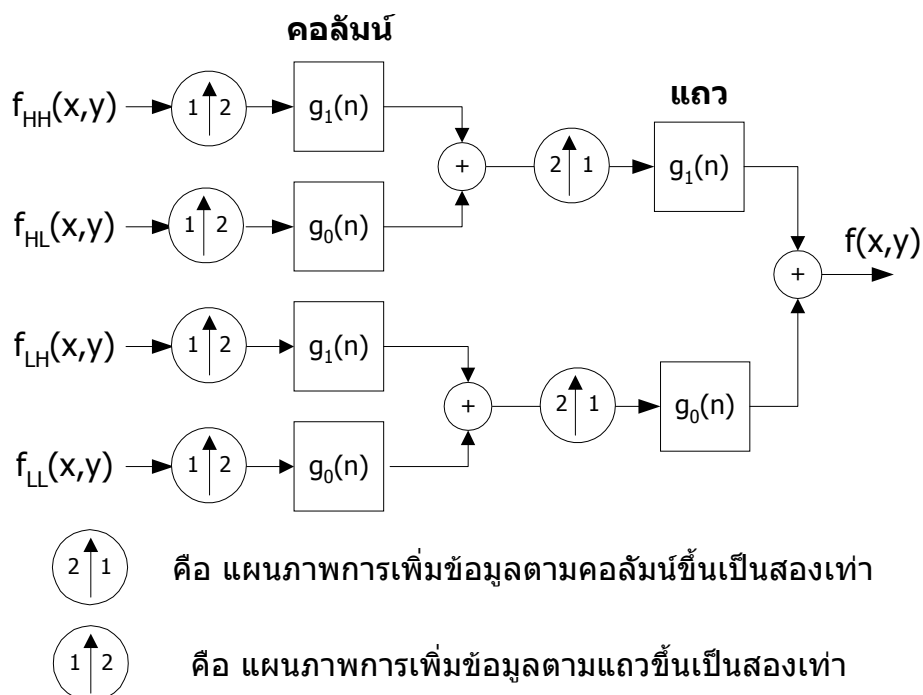
#### 4.5 การแปลงเวฟเล็ดแบบคัสครีทใน 2 มิติ

การแปลงเวฟเล็ดแบบคัสครีทแบบ 2 มิติ ใช้หลักการแยกองค์ประกอบเป็นแบนด์ย่อย (subband decomposition) (Stephane, 1989; Marc, Michel, Pierre, and Ingrid, 1992) มีหลักการประมวลผลภาพ โดยใช้การแปลงเวฟเล็ดออกเป็นสองขั้นตอนคือ การแยกองค์ประกอบด้วยเวฟเล็ด มีวิธีการแปลงเป็นดังภาพที่ 4.18 กำหนดให้  $f(x,y)$  คือภาพต้นแบบ  $f_{LL}(x,y)$ ,  $f_{LH}(x,y)$ ,  $f_{HL}(x,y)$  และ  $f_{HH}(x,y)$  คือสัมประสิทธิ์เวฟเล็ด และการสร้างกลับจากองค์ประกอบด้วยเวฟเล็ด เป็นดังภาพที่ 4.19



ภาพที่ 4.18 แผนภาพการแยกองค์ประกอบด้วยเวฟเล็ด

จากภาพที่ 4.18 ภาพต้นแบบจะถูกกรองไปตามแนวแถวด้วยฟิลเตอร์  $h_0(n)$  และ  $h_1(n)$  จากนั้นนำผลที่ได้มาลดจำนวนข้อมูลลงครึ่งหนึ่ง โดยการเก็บข้อมูล 1 คอลัมน์จากข้อมูล 2 คอลัมน์ จะได้เอาต์พุตความถี่สูงและความถี่ต่ำ จากนั้นเอาต์พุตทั้งสองจะถูกกรองตามแนวคอลัมน์ด้วยฟิลเตอร์  $h_0(n)$  และ  $h_1(n)$  และลดจำนวนข้อมูลลงครึ่งหนึ่ง โดยการเก็บข้อมูลมา 1 แถวจากข้อมูล 2 แถว จะได้ภาพของสัมประสิทธิ์แบนด์ย่อยความถี่ต่ำคือ  $f_{LL}(x,y)$  จำนวนหนึ่งภาพ และภาพของสัมประสิทธิ์แบนด์ย่อยความถี่สูงจำนวน 3 ภาพคือ  $f_{HL}(x,y)$ ,  $f_{LH}(x,y)$  และ  $f_{HH}(x,y)$  โดยแต่ละภาพจะมีขนาดลดลงเป็นหนึ่งในสี่ของภาพต้นแบบ



ภาพที่ 4.19 แผนภาพการสร้างกลับจากองค์ประกอบด้วยเวฟเล็ต

จากภาพที่ 4.19 การสร้างภาพขึ้นมาใหม่จากองค์ประกอบแบนด์ย่อย เริ่มจากภาพของสัมประสิทธิ์แบนด์ย่อยทั้ง 4 ภาพจะถูกเพิ่มจำนวนข้อมูลเป็น 2 เท่า โดยการเติมศูนย์เข้าไปในระหว่างแต่ละคู่ของแถว หลังจากเพิ่มจำนวนข้อมูลแล้วผลลัพธ์ที่ได้จะถูกกรองไปตามคอลัมน์ด้วยฟิลเตอร์  $g_0(n)$  และ  $g_1(n)$  และบวกเข้าด้วยกันตามภาพที่ 4.19 จากนั้นนำผลลัพธ์ที่ได้มาเติมศูนย์เข้าไปในระหว่างแต่ละคู่ของคอลัมน์ หลังจากเพิ่มจำนวนข้อมูลแล้วผลลัพธ์ที่ได้จะถูกกรองไปตามแถวด้วยฟิลเตอร์  $g_0(n)$  และ  $g_1(n)$  และบวกเข้าด้วยกันตามภาพที่ 4.19 ผลลัพธ์ที่ได้คือภาพต้นแบบที่สร้างกลับขึ้นมาใหม่

การแปลงเวฟเล็ตของข้อมูลภาพแบบหลายครั้ง เป็นดังภาพที่ 4.20 ซึ่งภาพที่ 4.20 จะแสดงภาพแบนด์ย่อยที่ผ่านการแปลงเวฟเล็ตหลายครั้ง และมีตัวอย่างภาพที่ผ่านการแปลงเวฟเล็ตด้วยวิธีการแยกองค์ประกอบด้วยเวฟเล็ต 3 ครั้ง เป็นดังภาพที่ 4.21

$f1_{LL}(x,y)$	$f1_{LH}(x,y)$
$f1_{HL}(x,y)$	$f1_{HH}(x,y)$

ภาพแบนด์ย่อยที่ได้จากการแปลงเวฟเล็ต 1 ครั้ง

$f2_{LL}(x,y)$	$f2_{LH}(x,y)$	$f1_{LH}(x,y)$
$f2_{HL}(x,y)$	$f2_{HH}(x,y)$	
$f1_{HL}(x,y)$		$f1_{HH}(x,y)$

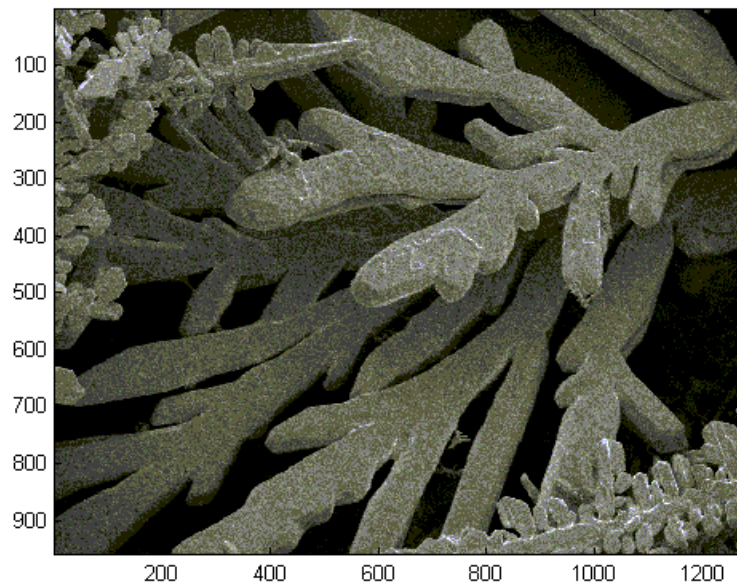
ภาพแบนด์ย่อยที่ได้จากการแปลงเวฟเล็ต 2 ครั้ง

$f3_{LL}$	$f3_{LH}$	$f2_{LH}(x,y)$	$f1_{LH}(x,y)$
$f3_{HL}$	$f3_{HH}$		
$f2_{HL}(x,y)$		$f2_{HH}(x,y)$	
$f1_{HL}(x,y)$		$f1_{HH}(x,y)$	

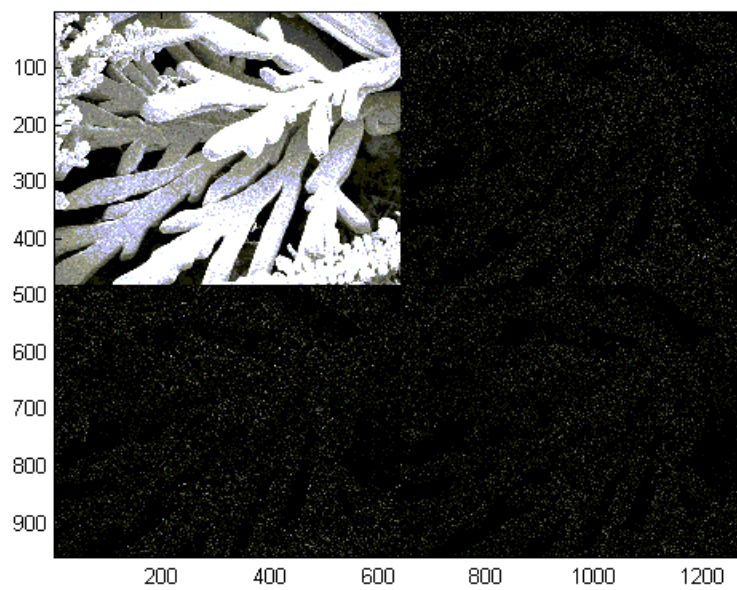
ภาพแบนด์ย่อยที่ได้จากการแปลงเวฟเล็ต 3 ครั้ง

ภาพที่ 4.20 แสดงภาพแบนด์ที่ผ่านการแปลงเวฟเล็ตหลายครั้ง

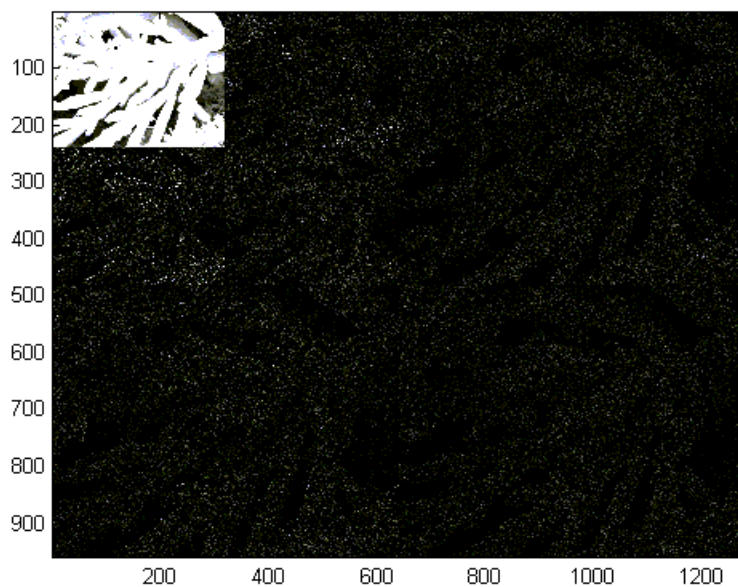
Original Image



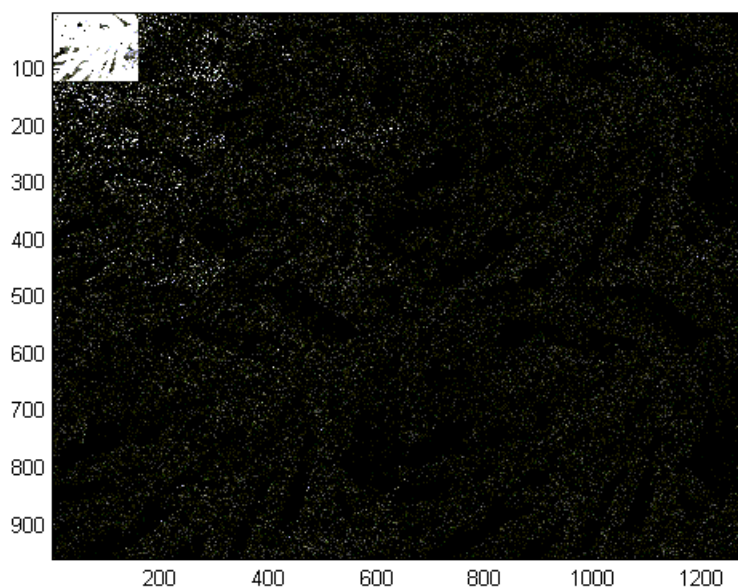
ภาพที่ 4.21.a ภาพต้นแบบ Paga ขนาด 960x1280x8 บิต



ภาพที่ 4.21.b ภาพที่ผ่านการแปลงเวฟเล็ตด้วยวิธีการแยกองค์ประกอบด้วยเวฟเล็ต 1 ครั้ง



ภาพที่ 4.21.c ภาพที่ผ่านการแปลงเวฟเล็ตด้วยวิธีการแยกองค์ประกอบด้วยเวฟเล็ต 2 ครั้ง



ภาพที่ 4.21.d ภาพที่ผ่านการแปลงเวฟเล็ตด้วยวิธีการแยกองค์ประกอบด้วยเวฟเล็ต 3 ครั้ง

ภาพที่ 4.21 ตัวอย่างภาพที่ผ่านการแปลงเวฟเล็ตด้วยวิธีการแยกองค์ประกอบด้วยเวฟเล็ต



#### 4.6 สรุป

ในบทนี้ได้กล่าวถึงความเป็นมาของการแปลงเวฟเล็ด ทฤษฎีพื้นฐานของการแปลงเวฟเล็ด โดยอธิบายถึงลักษณะของเวฟเล็ด อธิบายทฤษฎีพื้นฐานของการสเกลและการเลื่อนตำแหน่ง และแสดงภาพของตระกูลเวฟเล็ดแม่ต่างๆ ที่นิยมใช้งานในปัจจุบัน นอกจากนั้นยังได้กล่าวถึงการแปลงเวฟเล็ดดิสครีทใน 1 มิติ ด้วยวิธีการฟิลเตอร์แบงก์ โดยการแปลงเวฟเล็ดใช้วิธีการวิเคราะห์สัญญาณด้วยฟิลเตอร์แบงก์และการแปลงกลับเวฟเล็ดใช้วิธีสังเคราะห์สัญญาณด้วยฟิลเตอร์แบงก์ และการแปลงเวฟเล็ดแบบดิสครีทใน 2 มิติ โดยการแปลงเวฟเล็ดใช้วิธีการแยกองค์ประกอบด้วยเวฟเล็ด และการแปลงกลับเวฟเล็ดใช้วิธีการสร้างกลับจากองค์ประกอบด้วยเวฟเล็ด

## บทที่ 5

### การบีบอัดข้อมูลภาพ EM ด้วยอัลกอริทึม

#### Set Partitioning in Hierarchical Tree (SPIHT)

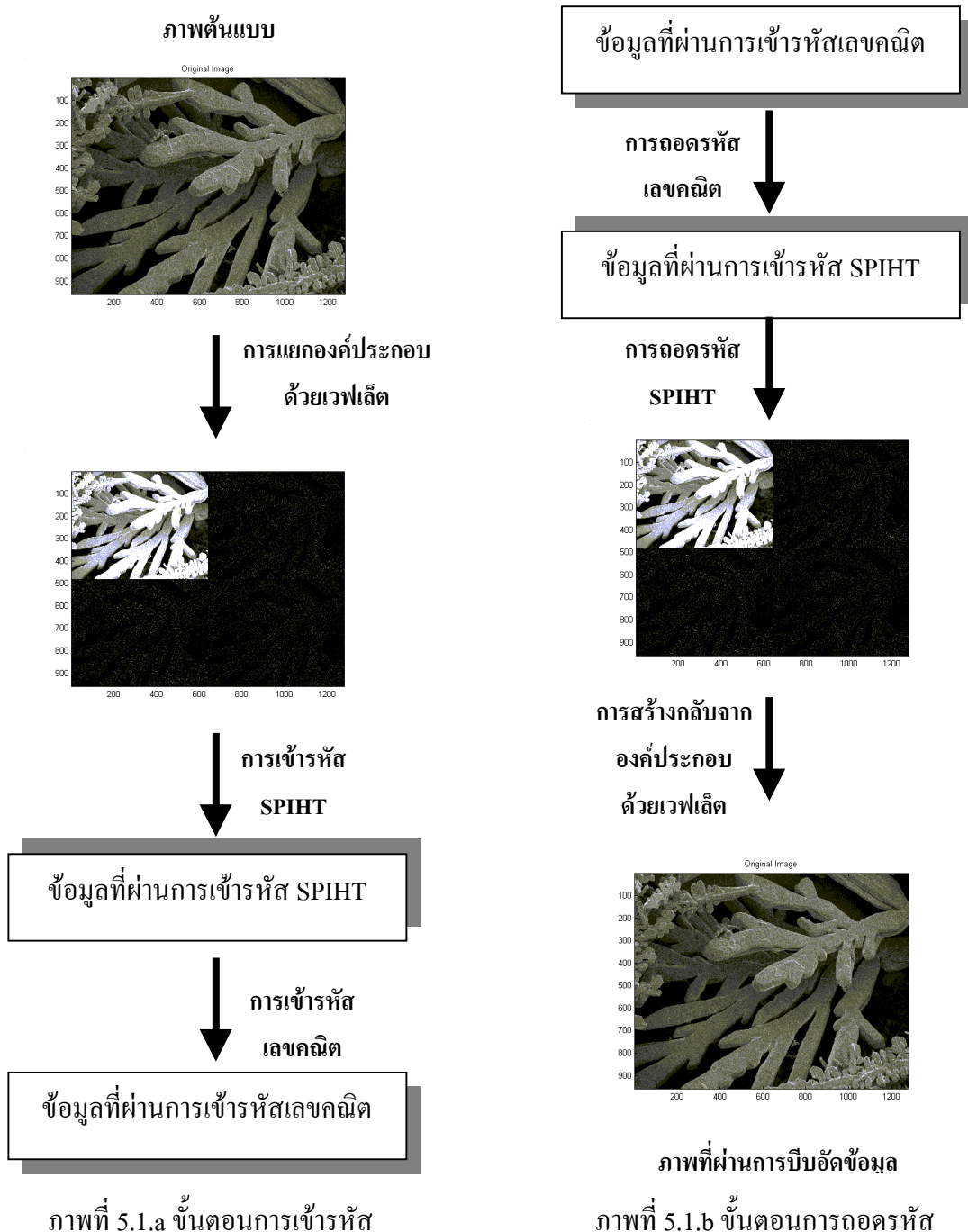
##### 5.1 บทนำ

ภาพของกล้อง EM เมื่อนำมาแปลงด้วยวิธีเวฟเลตแบบดิสครีต ข้อมูลที่เป็นพื้นผิววัตถุจะอยู่ที่สัมประสิทธิ์ที่แบนด์ย่อยความถี่ต่ำ และข้อมูลที่เป็นขอบหรือลายเส้นของวัตถุจะอยู่ที่สัมประสิทธิ์ที่แบนด์ย่อยความถี่สูง ดังนั้นการบีบอัดข้อมูลภาพ EM ด้วยการแปลงเวฟเลตจะต้องให้ความสำคัญกับสัมประสิทธิ์ทุกแบนด์ย่อย เพื่อทำให้ภาพที่ผ่านการบีบอัดข้อมูลมีรายละเอียดของภาพเหมือนภาพต้นแบบมากที่สุด อัลกอริทึม SPIHT เป็นอัลกอริทึมหนึ่งที่เหมาะสม เนื่องจากอัลกอริทึมนี้เข้ารหัสโดยให้ความสำคัญกับขนาดของสัมประสิทธิ์ และไม่สนใจว่าสัมประสิทธิ์ตัวนั้นจะอยู่ในระดับแบนด์ย่อยใด

##### 5.2 อัลกอริทึม SPIHT

Embedded Zerotrees Wavelet (EZW) วิธีนี้นำเสนอครั้งแรกโดย Jerome M. Shapiro (1993) ให้อัตราการลดข้อมูลที่ดีกว่าการบีบอัดข้อมูลภาพด้วยวิธี JPEG ซึ่งเป็นวิธีที่ได้รับการพัฒนาขึ้นมาจากสองหน่วยงานคือ International Organization for Standardization (ISO) และ International Telecommunication Union (ITU) และสามารถนำไปใช้ได้กับรูปภาพทุกประเภท โดยไม่ต้องปรับแก้อัลกอริทึม เพราะอัลกอริทึมนี้จะเข้ารหัสข้อมูลแบบปรับเปลี่ยนได้เอง (adaptive coding) (Jerome, 1993) นอกจากนั้นการลดข้อมูลภาพแบบ EZW ยังไม่เกิดปัญหาเรื่องการเกิดบล็อกอาร์ติแฟกต์ (block artifacts) คือการเกิดบล็อกที่บริเวณขอบของวัตถุและลายเส้นอย่างชัดเจน (ศิริพร เศษศิริรักษ์, 2543; สุขสันต์ จิรเชวง และ วุฒิพงศ์ อารกุล 2544) เพราะ EZW จะเข้ารหัสโดยให้ความสำคัญกับขนาดของสัมประสิทธิ์ และไม่สนใจว่าสัมประสิทธิ์ตัวนั้นจะอยู่ในระดับความถี่ใด (ขวัญฤทัย ไพรีพ่ายฤทธิ์, 2542) ได้มีการพัฒนาอัลกอริทึมการเข้ารหัสด้วย EZW ให้มีประสิทธิภาพเพิ่มขึ้นมากมายหลายวิธี แต่ในงานวิจัยนี้เลือกใช้อัลกอริทึมการเข้ารหัสด้วยวิธี SPIHT ซึ่งให้อัตราการบีบอัดข้อมูลที่ดีกว่าการบีบอัดข้อมูลภาพด้วยวิธี EZW (Amir and William, 1996; Al, 2000) นอกจากนั้นการบีบอัดข้อมูลภาพด้วยวิธี SPIHT เป็นอัลกอริทึมที่เร็วและมีประสิทธิภาพ

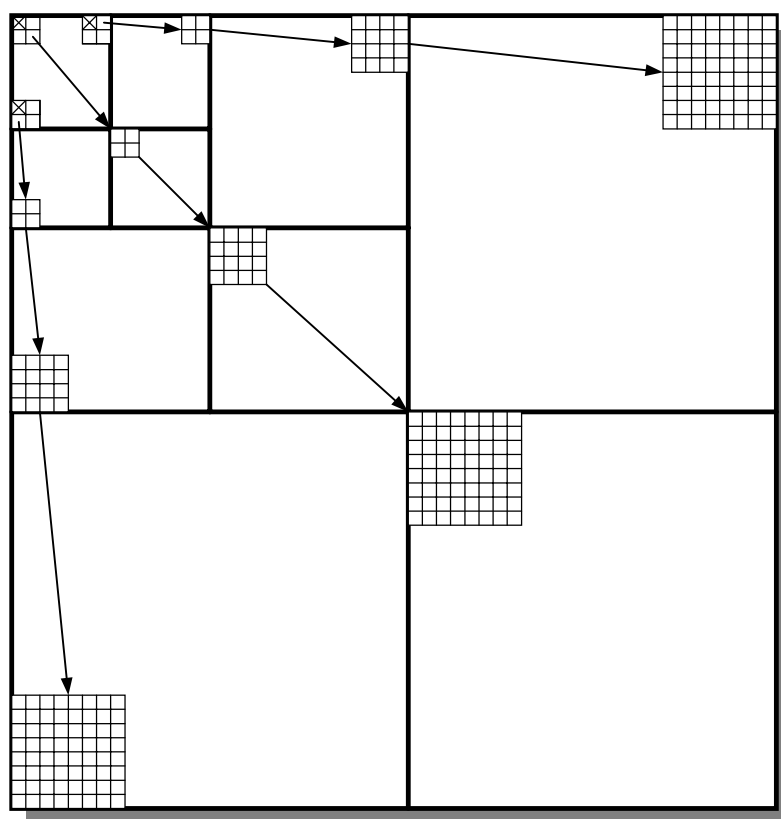
(Brian, and Thomas, 2001) มีขั้นตอนการเข้ารหัส (encode) เป็นดังภาพที่ 5.1.a และมีขั้นตอนการถอดรหัส (decode) เป็นดังภาพที่ 5.1.b



ภาพที่ 5.1 แผนภาพแสดงขั้นตอนการเข้ารหัสและขั้นตอนการถอดรหัส

### 5.2.1 ความสัมพันธ์แบบ Spatial Orientation Trees

โดยปกติพลังงานของภาพ (image energy) ที่ทำการแปลงเวฟเล็ตแบบดิสครีท พลังงานส่วนใหญ่จะอยู่ในส่วนความถี่ต่ำ หรือส่วนภาพของสัมประสิทธิ์แบนด์ย่อยความถี่ต่ำคือ  $f_{LL}(x,y)$  ดังนั้นสามารถคาดหวังได้ว่าขนาดของสัมประสิทธิ์จากแบนด์ย่อยระดับการแปลงที่สูง ไปยังระดับแบนด์ย่อยระดับการแปลงที่ต่ำกว่าของการแปลงเวฟเล็ต ตามทิศทางของ Spatial ตามความสัมพันธ์แบบ Spatial Orientation Trees เป็นดังภาพที่ 5.2 ขนาดของสัมประสิทธิ์จะมีค่ามากขึ้น (Amir and William, 1993)

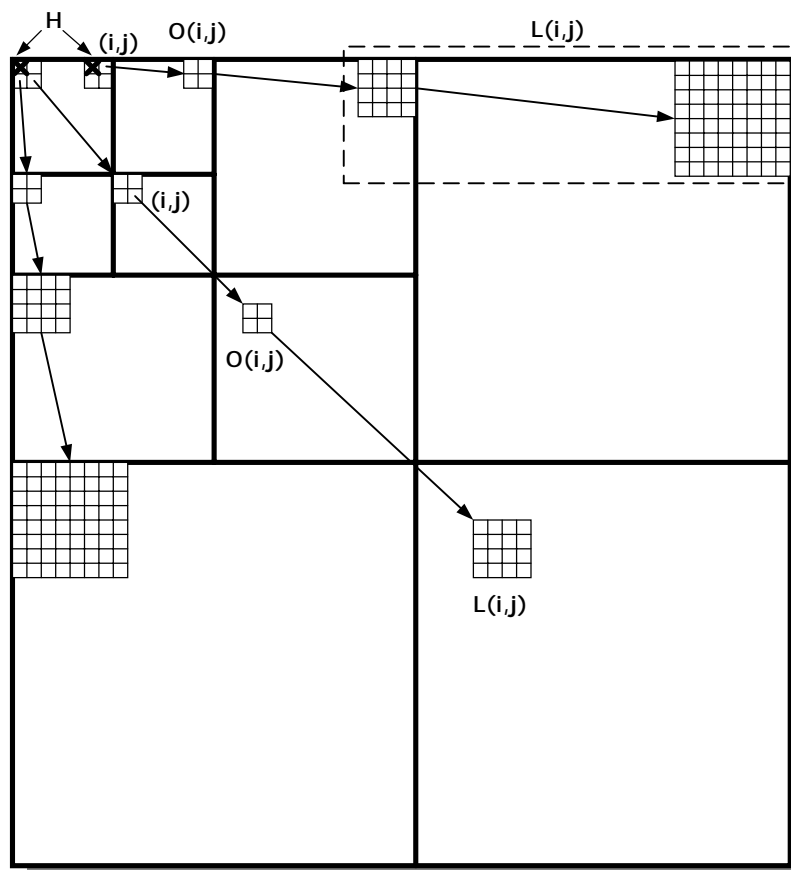


ภาพที่ 5.2 แผนภาพแสดงความสัมพันธ์แบบ Spatial Orientation Trees

การกำหนดตำแหน่ง (coordinates) ของการเข้ารหัส ตามความสัมพันธ์แบบ spatial orientation trees เป็นดังภาพที่ 5.3 และมีตำแหน่งที่ต้องกำหนดดังนี้

- $(i,j)$  คือ ตำแหน่งของสัมประสิทธิ์หรือโหนด (node)
- $O(i,j)$  คือ ตำแหน่งของการสืบทอด (offspring) ของโหนด และ 1 โหนดมีการสืบทอด 4 ตำแหน่ง

- $L(i,j)$  คือ ตำแหน่งของการสืบทอดของ  $O(i,j)$  โดยมีการสืบทอดไปจนกว่าจะถึงแบนด์ย่อยต่ำสุด
- $D(i,j) = O(i,j) + L(i,j)$  คือ ตำแหน่งของการสืบทอดทั้งหมดของโหนด
- $H$  คือ ตำแหน่งทั้งหมดที่อยู่ในแบนด์ย่อยสูงสุด ที่เป็นต้นกำเนิดโหนด และแต่ละตำแหน่งจะมีตำแหน่งของการสืบทอด 3 ตำแหน่ง



ภาพที่ 5.3 แผนภาพกำหนดตำแหน่งของการเข้ารหัส

### 5.2.2 อัลกอริทึมการเข้ารหัส SPIHT

ในการเข้ารหัสและถอดรหัส จะมีตารางในการเข้ารหัสอยู่ด้วยกัน 3 ตาราง คือ

1. List of Insignificant Set (LIS) คือ ตารางที่ใช้เก็บตำแหน่งของกลุ่มสัมประสิทธิ์เวฟเล็ตที่ยังไม่มีความสำคัญ ซึ่ง LIS นี้จะมีอยู่ด้วยกัน 2 ชนิดคือ ชนิด A และ ชนิด B ซึ่งค่าที่เก็บใน LIS ชนิด A คือ  $D(i,j)$  และค่าที่เก็บใน LIS ชนิด B คือ  $L(i,j)$
2. List of Insignificant Pixel (LIP) คือ ตารางที่ใช้เก็บตำแหน่งของสัมประสิทธิ์เวฟเล็ตที่ยังไม่มีความสำคัญ ซึ่งค่าที่เก็บคือ  $(i,j)$  ที่ไม่มีความสำคัญ

3. List of Significant Pixel (LSP) คือ ตารางที่ใช้เก็บตำแหน่งของสัมประสิทธิ์เวฟเล็ตที่มีความสำคัญ ซึ่งค่าที่เก็บคือ  $(i,j)$  ที่มีความสำคัญ

มีฟังก์ชันการเข้ารหัส  $S_T(k)$  เป็นดังสมการ

$$S_T(k) = \begin{cases} 1, & \max_{(i,j) \in k} |c(i,j)| \geq T \\ 0, & \text{อื่นๆ} \end{cases} \quad (5.1)$$

$$n \leq \lfloor \log_2(\max |c(i,j)|) \rfloor \quad (5.2)$$

$$T = 2^n \quad (5.3)$$

เมื่อ  $k$  คือ สัมประสิทธิ์  $(c(i,j))$  หรือกลุ่มของสัมประสิทธิ์ของเวฟเล็ต  $(D(i,j))$  หรือ  $L(i,j)$

การเข้ารหัสของอัลกอริทึม SPIHT มีขั้นตอนการเข้ารหัสอยู่ด้วยกัน 4 ขั้นตอนดังนี้  
**ขั้นตอนที่ 1** กำหนดค่าเริ่มต้น

- หาค่า  $n$  และ  $T$  ตามสมการที่ 5.2 และสมการที่ 5.3 ตามลำดับ
- LSP กำหนดเป็นตารางว่าง
- LIP กำหนดเป็นค่าสมาชิกของ  $H$
- LIS กำหนดเป็นค่าสมาชิกของ  $H$  และกำหนดให้เป็นชนิด A

**ขั้นตอนที่ 2** การจัดลำดับ (sorting) มีขั้นตอนตามภาพที่ 5.4

- สำหรับแต่ละ  $(i,j)$  ใน LIP
  - ส่งค่า  $S_T(c(i,j))$  ออก
  - ถ้าบิตเท่ากับ 1 ให้ส่งบิตเช็คเครื่องหมาย (กำหนดให้ถ้า  $c(i,j)$  มีเครื่องหมายเป็นบวก ให้ส่ง 1 และถ้า  $c(i,j)$  มีเครื่องหมายลบให้ส่ง 0)
- สำหรับแต่ละ  $(i,j)$  ใน LIS
  - ถ้าเป็น LIS ชนิด A
    - \* ส่งค่า  $S_T(D(i,j))$  ออก
    - \* ถ้า  $S_T(D(i,j))$  เท่ากับ 1
      - สำหรับแต่ละ  $(k,l)$  ของ  $O(i,j)$ 
        - ~ ส่งค่า  $S_T(c(k,l))$  ออก
        - ~ ถ้าค่า  $S_T(c(k,l))$  เท่ากับ 1 ให้ส่งบิตเช็คเครื่องหมาย และให้ส่ง  $(k,l)$  ไป LSP

~ ถ้าค่า  $S_T(c(k,l))$  เท่ากับ 0 ให้ส่ง  $(k,l)$  ไป LIP

\* ถ้า  $L(i,j)$  มีสมาชิก ให้ส่ง  $(i,j)$  ไปท้าย LIS แล้วกำหนดให้เป็นชนิด B

- ถ้าเป็น LIS ชนิด B

\* ส่งค่า  $S_T(L(i,j))$  ออก

~ ถ้าค่า  $S_T(L(i,j))$  เท่ากับ 1 ให้ส่งแต่ละสมาชิกของ  $O(i,j)$  ไปท้าย LIS แล้ว

กำหนดให้เป็นชนิด A และย้าย  $(i,j)$  ออกจาก LIS

### ขั้นตอนที่ 3 Quantization

- $n = n - 1$
- สำหรับแต่ละ  $(i,j)$  ใน LSP
  - ส่งค่าบิตที่  $n$  ของ  $c(i,j)$  ออก

### ขั้นตอนที่ 4 Update

- ทำซ้ำขั้นตอนที่ 2 ถึงขั้นตอนที่ 3 และยุติเมื่อมีการส่งข้อมูลการบีบอัดข้อมูลครบ

ตามต้องการ

### 5.2.3 การจัดเก็บเป็นไฟล์การบีบอัดข้อมูล

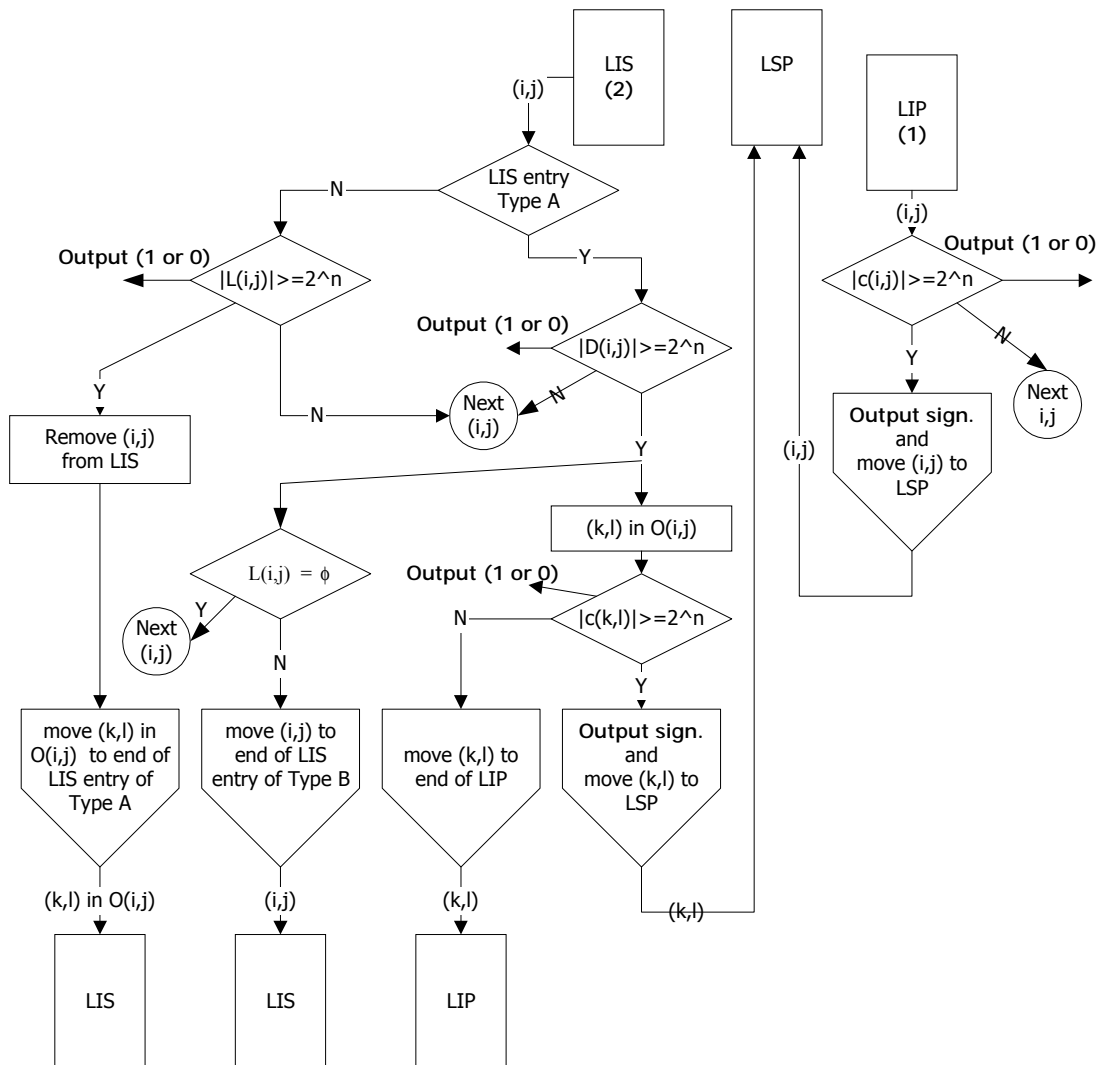
การจัดเก็บไฟล์แบ่งออกเป็น 2 ส่วนคือ ไฟล์ส่วนหัว และไฟล์ข้อมูลที่ได้จากการเข้ารหัส โดยไฟล์แต่ละส่วนจะมีการจัดเก็บค่าดังนี้

1. ไฟล์ส่วนหัวจะทำการจัดเก็บข้อมูลคือ
  - 1.1 จำนวนข้อมูลทั้งหมดที่ทำการบีบอัด
  - 1.2 ค่า  $n$
  - 1.3 จำนวนคอลัมน์และจำนวนแถวของภาพ
  - 1.4 จำนวนระดับการแปลงเวฟเล็ต
2. ไฟล์ข้อมูลจะทำการจัดเก็บข้อมูลทั้งหมดที่ได้จากการเข้ารหัสการบีบอัดข้อมูล

ด้วยวิธี SPIHT

### 5.2.4 อัลกอริทึมการถอดรหัส SPIHT

ในการถอดรหัสจะใช้อัลกอริทึมเดียวกับการเข้ารหัส เพียงแต่เปลี่ยนจากการส่งค่าเป็นการรับค่า และนำค่านั้นมาพิจารณาเหมือนการเข้ารหัส



ภาพที่ 5.4 แผนภาพแสดงการเข้ารหัสขั้นตอนที่ 2 ของอัลกอริทึม SPIHT

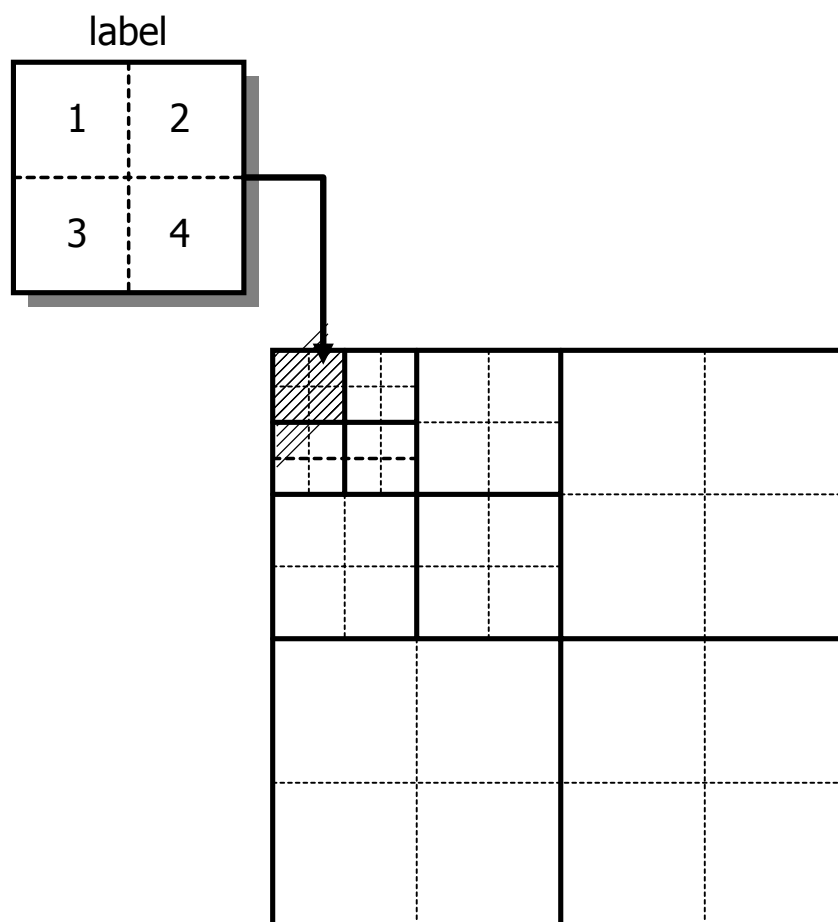
### 5.3 การพัฒนาอัลกอริทึม SPIHT

จากอัลกอริทึมที่กล่าวมามีจุดที่สามารถปรับปรุงพัฒนาได้ 2 จุดกล่าวคือ  
**กรณีที่ 1** คือ  $O(i,j)$  ไม่มีระดับความสำคัญแต่  $L(i,j)$  มีระดับความสำคัญ จะมีการส่งบิตออกมา 2 บิต (ไม่นับรวมการเช็คแต่ละสมาชิกของ  $O(i,j)$ ) ในกรณีนี้สามารถปรับอัลกอริทึมให้ส่งออกมาเพียง 1 บิต ทำได้โดยการเพิ่มการเช็คการส่ง  $(k,l)$  เข้า LIP คือถ้ามีการส่ง  $(k,l)$  ทั้งหมดของ  $O(i,j)$  แสดงว่า  
 ได้เกิดกรณีนี้ขึ้น ดังนั้นสามารถทำการส่ง  $(k,l)$  เข้า LIS ให้เป็นชนิด A ได้เลย



กรณีที่ 2 คือ  $O(i,j)$  มีระดับความสำคัญแต่  $L(i,j)$  ไม่มีระดับความสำคัญ จะมีการส่งบิตออกมาเหมือนกับข้อ 1 ในกรณีนี้สามารถปรับอัลกอริทึมให้ส่งข้อมูลให้น้อยลงได้ ทำได้โดยการเพิ่ม List of Forbidden Coefficients (LFC) และมีขั้นตอนการหาดังนี้

- ทำการแบ่งสัมประสิทธิ์เวฟเล็ตแต่ละแบนด์ย่อยออกเป็น 4 ส่วน โดยเรียกแต่ละส่วนที่ทำการแบ่งว่า label โดยแต่ละแบนด์ย่อยที่มีการแบ่งจะมีค่าตำแหน่ง label ไม่ซ้ำกัน เป็นดังภาพที่ 5.5



ภาพที่ 5.5 แผนภาพแสดงการแบ่งแบนด์ย่อยและกำหนด label ของ LFC

- หาค่าสัมประสิทธิ์สูงสุดของแต่ละ label
- หาค่าระดับของการควอนไทซ์ในแต่ละ label และนำมาทำเป็น LFC โดยหาได้จาก

สมการ 5.4

$$LFC(w) \leq \lfloor \log_2(\max(c_{\text{label}}(w))) \rfloor \quad (5.4)$$

เมื่อ  $w$  คือ ตำแหน่งของ label

$C_{\text{label}}(w)$  คือ สัมประสิทธิ์เวฟเล็ตทั้งหมดที่  $w$

$LFC(w)$  คือ ระดับการควอนไทซ์ ที่  $w$

โดย  $LFC$  นี้จะเป็นตัวบ่งบอกสถานะว่า  $O(i,j)$  และ  $L(i,j)$  ควรมีการส่งบิตออกมาหรือไม่ โดยจะดูว่า  $O(i,j)$  หรือ  $L(i,j)$  นี้อยู่ที่ label ไหน และมีระดับการควอนไทซ์ที่เท่าใด ดังนั้นถ้าระดับของการควอนไทซ์ที่มีอยู่มีค่าน้อยกว่าระดับการควอนไทซ์ของรอบการทำงาน ก็ไม่ต้องทำการเช็ค  $O(i,j)$  หรือ  $L(i,j)$  สามารถเขียนเป็นฟังก์ชันได้ดังสมการที่ 5.5

$$F_n(p) = \begin{cases} 1, & \max_{w \in p} LFC(w) \leq n \\ 0, & \text{อื่นๆ} \end{cases} \quad (5.5)$$

เมื่อ  $F_n(p)$  คือ ฟังก์ชันที่ใช้เป็นเงื่อนไขในการเช็คค่าระดับการควอนไทซ์ของ  $O(i,j)$  หรือ  $L(i,j)$

$p$  คือ ค่า  $LFC$  ทั้งหมดที่เป็นของสมาชิกใน  $O(i,j)$  หรือ  $L(i,j)$

### 5.3.1 อัลกอริทึมการเข้ารหัสอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง

การเข้ารหัสของอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง มีขั้นตอนการเข้ารหัสอยู่ด้วยกัน 4 ขั้นตอนดังนี้

**ขั้นตอนที่ 1** กำหนดค่าเริ่มต้น

- หาค่า  $n$  และ  $T$  ตามสมการที่ 5.2 และสมการที่ 5.3 ตามลำดับ
- LSP กำหนดเป็นตารางว่าง
- LIP กำหนดเป็นค่าสมาชิกของ  $H$
- LIS กำหนดเป็นค่าสมาชิกของ  $H$  และกำหนดให้เป็นชนิด  $A$
- LFC กำหนดค่าเป็นไปตามสมการที่ 5.4

**ขั้นตอนที่ 2** การจัดลำดับ มีขั้นตอนตามภาพที่ 5.6

- สำหรับแต่ละ  $(i,j)$  ใน LIP
  - ส่งค่า  $S_T(c(i,j))$  ออก
  - ถ้าบิตเท่ากับ 1 ให้ส่งบิตเช็คเครื่องหมาย

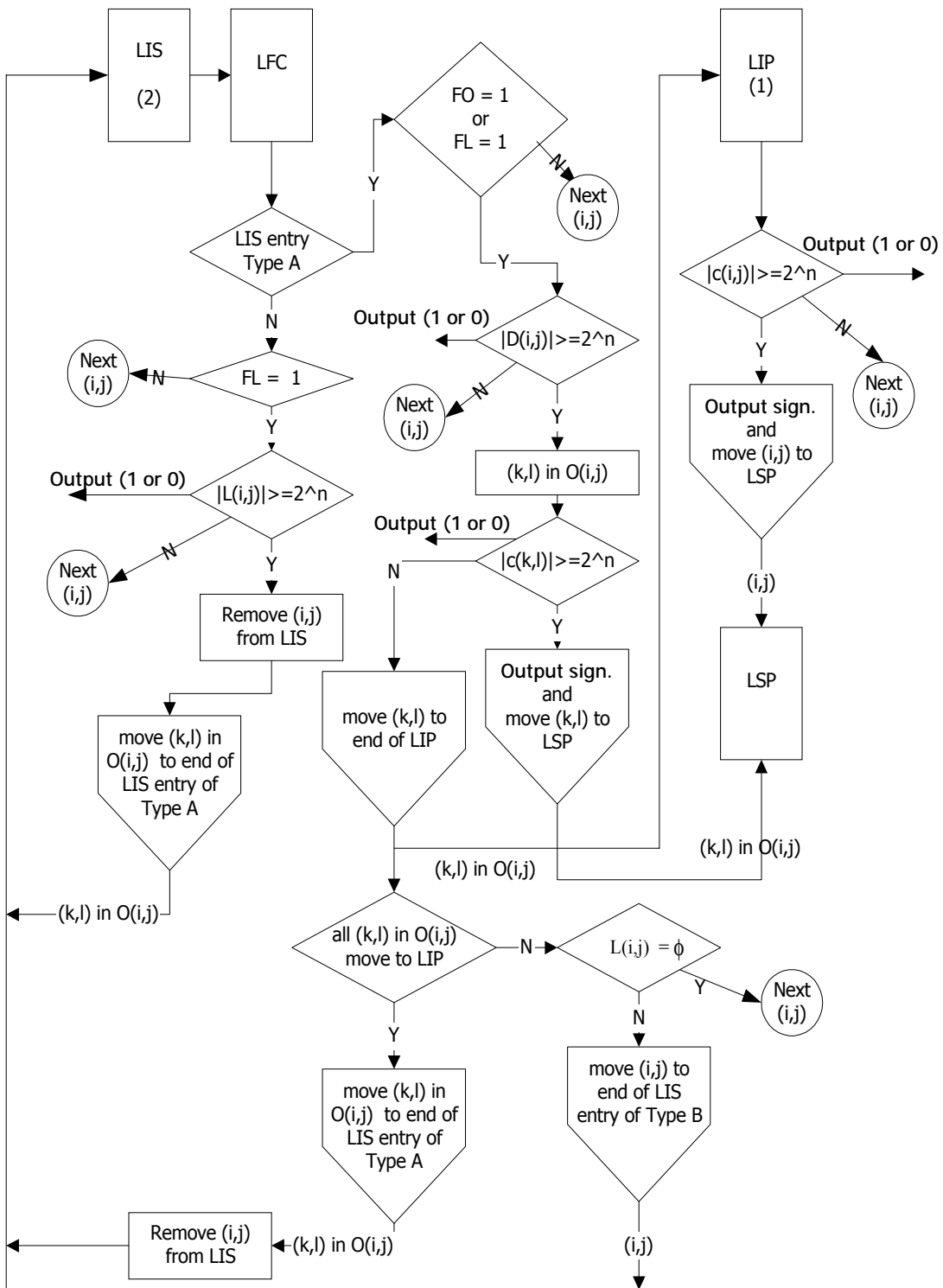
- สำหรับแต่ละ  $(i,j)$  ใน LIS
  - เช็คค่า FO จาก  $F_n(O(i,j))$
  - เช็คค่า FL จาก  $F_n(L(i,j))$
  - ถ้าเป็น LIS ชนิด A และ FO หรือ FL เท่ากับ 1
    - \* ส่งค่า  $S_T(D(i,j))$  ออก
    - \* ถ้า  $S_T(D(i,j))$  เท่ากับ 1
      - สำหรับแต่ละ  $(k,l)$  ของ  $O(i,j)$ 
        - ~ ส่งค่า  $S_T(c(k,l))$  ออก
        - ~ ถ้าค่า  $S_T(c(k,l))$  เท่ากับ 1 ให้ส่งบิตเช็คเครื่องหมาย และให้ส่ง  $(k,l)$  ไป LSP
        - ~ ถ้าค่า  $S_T(c(k,l))$  เท่ากับ 0 ให้ส่ง  $(k,l)$  ไป LIP
    - \* ถ้า  $L(i,j)$  มีสมาชิก
      - ~ ถ้าสมาชิกทุกตัวของ  $O(i,j)$  ถูกส่งไป LIP ให้ทำการส่งสมาชิกทุกตัวของ  $O(i,j)$  ไปท้าย LIS และกำหนดให้เป็นชนิด A
      - ~ ถ้าสมาชิกบางตัวของ  $O(i,j)$  ไม่ถูกส่งไป LIP ให้ทำการส่ง  $(i,j)$  ไปท้าย LIS แล้วกำหนดให้เป็นชนิด B
  - ถ้าเป็น LIS ชนิด B และ FL เท่ากับ 1
    - \* ส่งค่า  $S_T(L(i,j))$  ออก
      - ~ ถ้าค่า  $S_T(L(i,j))$  เท่ากับ 1 ให้ส่งแต่ละสมาชิกของ  $O(i,j)$  ไปท้าย LIS แล้วกำหนดให้เป็นชนิด A และทำการย้าย  $(i,j)$  ออกจาก LIS

### ขั้นตอนที่ 3 Quantization

- $n = n - 1$
- สำหรับแต่ละ  $(i,j)$  ใน LSP
  - ส่งค่าบิตที่  $n$  ของ  $c(i,j)$  ออก

### ขั้นตอนที่ 4 Update

- ทำซ้ำขั้นตอนที่ 2 ถึงขั้นตอนที่ 3 และยุติเมื่อมีการส่งข้อมูลการบีบอัดข้อมูลครบตามต้องการ



ภาพที่ 5.6 แผนภาพแสดงการเข้ารหัสขั้นตอนที่ 2 ของอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง

### 5.3.2 การจัดเก็บเป็นไฟล์การบีบอัดข้อมูลอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง

การจัดเก็บไฟล์แบ่งออกเป็น 2 ส่วนคือ ไฟล์ส่วนหัว และไฟล์ข้อมูลที่ได้จากการเข้ารหัส โดยไฟล์แต่ละส่วนจะมีการจัดเก็บค่าดังนี้

#### 1. ไฟล์ส่วนหัวจะทำการจัดเก็บข้อมูลคือ

1.1 จำนวนข้อมูลทั้งหมดที่ทำการบีบอัด

1.2 ค่า  $n$

1.3 จำนวนคอลัมน์และจำนวนแถวของภาพ

1.4 จำนวนระดับการแปลงเวฟเล็ต

1.5 ค่า LFC

2. ไฟล์ข้อมูลจะทำการจัดเก็บข้อมูลทั้งหมดที่ได้จากการเข้ารหัสการบีบอัดข้อมูลด้วยวิธี SPIHT ที่ผ่านการปรับปรุง

### 5.3.3 อัลกอริทึมการถอดรหัสอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง

ในการถอดรหัสจะใช้อัลกอริทึมเดียวกับการเข้ารหัส เพียงแต่เปลี่ยนจากการส่งค่าเป็นการรับค่า และนำค่านั้นมาพิจารณาเหมือนการเข้ารหัส

## 5.4 การเข้ารหัสเลขคณิต

เนื่องจากวิธีการเข้ารหัสเลขคณิตจะอาศัยค่าของขอบเขตความน่าจะเป็นในการเกิดสัญลักษณ์ของข้อมูลแต่ละสัญลักษณ์ ซึ่งจะมีค่าอยู่ในช่วง  $[0,1)$  ซึ่งถ้ามีการเข้ารหัสกับข้อมูลที่มีขนาดยาวจะทำให้ค่าของระยะขอบเขตมีค่าน้อยมาก ทำให้เกิดโพลตติ้งพอยต์ (floating point) ซึ่งเป็นปัญหาในการเขียนโปรแกรม (Ian, Radford, and John, 1987; Paul, and Jeffrey, 1994) แต่ปัญหานี้ได้ถูกแก้ไขได้โดยการเปลี่ยนช่วงขอบเขตแบบทศนิยม ไปเป็นช่วงขอบเขตแบบจำนวนเต็ม (David, 1998) ซึ่งจะมีขั้นตอนการเข้ารหัสดังนี้

#### ขั้นตอนการเข้ารหัส

ขั้นตอนที่ 1 ทำการกำหนดช่วงขอบเขตเริ่มต้น ซึ่งจะมีอยู่ด้วยกัน 2 ชุดคือ ขอบเขตแบบทศนิยมค่า  $[LL, HH)$  มีค่าเป็น  $[0,1)$  และขอบเขตแบบจำนวนเต็มค่า  $[L,H)$  มีค่าเป็น  $[0000,9999)$

ขั้นตอนที่ 2 นำสัญลักษณ์ของข้อมูลมาพิจารณา โดยแบ่งเป็นอีก 3 ขั้นตอนย่อยคือ

2.1 ทำการปรับช่วงขอบเขตใหม่ที่สอดคล้องกับสัญลักษณ์ที่นำมาเข้ารหัสของข้อมูลตัวใหม่ แล้วนำช่วงขอบเขตดังกล่าวมาแทนให้เป็นช่วงขอบเขตปัจจุบันแทน โดยทำการหาขอบเขต

แบบทศนิยมก่อน แล้วนำค่าขอบเขตแบบทศนิยมไปหาค่าขอบเขตแบบจำนวนเต็มต่อ ซึ่งมีวิธีการหาค่าขอบเขตทั้งสองแบบดังนี้

ขอบเขตเป็นแบบทศนิยม

$$LL = LL^{\wedge} + (HH^{\wedge} - LL^{\wedge}) * P_L(S) \quad (5.6)$$

$$HH = LL^{\wedge} + (HH^{\wedge} - LL^{\wedge}) * P_H(S) \quad (5.7)$$

เมื่อ LL และ HH คือ ขอบเขตใหม่

LL<sup>^</sup> และ HH<sup>^</sup> คือ ขอบเขตเก่า

P<sub>L</sub>(S) คือ ความน่าจะเป็นของขอบเขตล่างของสัญลักษณ์ S

P<sub>H</sub>(S) คือ ความน่าจะเป็นของขอบเขตบนของสัญลักษณ์ S

ขอบเขตเป็นแบบจำนวนเต็ม

L = ค่าหลังจุดทศนิยมของ LL 4 ตัวแรก

H = ค่าหลังจุดทศนิยมของ HH 4 ตัวแรก - 1

ตัวอย่างเช่น

LL แบบทศนิยมเท่ากับ 0.5 จะได้ L แบบจำนวนเต็มเท่ากับ 5000

HH แบบทศนิยมเท่ากับ 0.75 จะได้ L แบบจำนวนเต็มเท่ากับ 7499

2.2 การพิจารณาการส่งเอาต์พุตที่ได้จากเข้ารหัสในขั้นตอนที่ 2.1 ออก มีเงื่อนไขคือตัวเลขตำแหน่งแรกทางซ้ายของทั้ง L และ H แบบจำนวนเต็มมีค่าซ้ำกันให้ทำการส่งตัวเลขนั้นออกมาและทำการปรับ L และ H แบบจำนวนเต็มใหม่ โดยการตัดตัวเลขตำแหน่งแรกทางซ้ายออกและทำการเลื่อนตำแหน่งถัดมาขึ้นมาแทน ส่วนตำแหน่งสุดท้ายจะมีการใส่ค่าคือ ค่าของ L จะใส่ 0 เป็นค่าสุดท้าย และค่าของ H จะใส่ 9 เป็นตำแหน่งสุดท้าย

ตัวอย่างเช่น

$$L = XYYY \text{ และ } H = XZZZ$$

เอาต์พุต คือ X

L ใหม่มีค่าเท่ากับ YYY0

H ใหม่มีค่าเท่ากับ ZZZ9

ในกรณีที่มีการส่งเอาต์พุตออกมา นอกจากจะต้องทำการปรับค่า L และ H แบบตัวเลขแล้วจะต้องทำการปรับค่า LL และ HH แบบทศนิยมด้วย การปรับทำได้โดยการเติมตำแหน่งทศนิยมไว้หน้าข้อมูล LL ได้เลย ส่วน HH จะต้องทำการบวก 1 เสียก่อนจึงค่อยเติมตำแหน่งทศนิยมได้

ตัวอย่างเช่น

$$L = 5000 \text{ และ } H = 8999$$

$$LL \text{ ใหม่มีค่าเท่ากับ } 0.5$$

$$HH \text{ ใหม่มีค่าเท่ากับ } 0.9$$

2.3 เช็การสิ้นสุดชุดของข้อมูลที่นำมาเข้ารหัส หรือเมื่อพบสัญลักษณ์ที่แสดงถึงจุดสิ้นสุดของข้อมูล ในกรณีที่เจอตามเงื่อนไขให้ไปทำขั้นตอนที่ 3 แต่ถ้ายังไม่เจอตามเงื่อนไขให้กลับไปทำขั้นตอนที่ 2.1

ขั้นตอนที่ 3 ทำการส่ง L แบบจำนวนเต็ม ไปเป็นเอาต์พุต

ตัวอย่างการเข้ารหัสเลขคณิต

ถ้ากำหนดให้สัญลักษณ์และจำนวนของสัญลักษณ์มีค่าเป็นดังตารางที่ 5.1 โดยข้อมูลของการนำมาเข้ารหัสมีการจัดเรียงสัญลักษณ์คือ A1A2A3A1A1A5A4A3A1A1 และมีขั้นตอนการแสดงผลการเข้ารหัสเลขคณิตเป็นดังตารางที่ 5.2 จากตารางที่ 5.2 เอาต์พุตของการเข้ารหัสคือ 717533750

ตารางที่ 5.1 ตัวอย่างสัญลักษณ์และจำนวนของสัญลักษณ์ของข้อมูลที่นำมาเข้ารหัสเลขคณิต

สัญลักษณ์	จำนวน	ความน่าจะเป็น	$[P_L(S), P_H(S))$	ความถี่สะสม
A1	5	$5/10 = 0.5$	$[0.5, 1)$	$[5, 10)$
A2	1	$1/10 = 0.1$	$[0.4, 0.5)$	$[4, 5)$
A3	2	$2/10 = 0.2$	$[0.2, 0.4)$	$[2, 4)$
A4	1	$1/10 = 0.1$	$[0.1, 0.2)$	$[1, 2)$
A5	1	$1/10 = 0.1$	$[0.0, 0.1)$	$[0, 1)$

จำนวนสัญลักษณ์ทั้งหมดเท่ากับ 10

ตารางที่ 5.2 ขั้นตอนการเข้ารหัสเลขคณิต

สัญลักษณ์	ช่วงขอบเขต แบบทศนิยม	ช่วงขอบเขต แบบจำนวนเต็ม	เอาต์พุต	การปรับช่วงขอบเขต แบบจำนวนเต็ม
A1	$LL = 0+(1-0)*0.5 = 0.5$ $HH = 0+(1-0)*1.0 = 1.0$	$L = 5000$ $H = 9999$		5000 9999
A2	$LL = 0.5+(1-0.5)*0.4 = 0.7$ $HH = 0.5+(1-0.5)*0.5 = 0.75$	$L = 7000$ $H = 7499$	7	0000 4999
A3	$LL = 0+(0.5-0)*0.2 = 0.1$ $HH = 0+(0.5-0)*0.4 = 0.2$	$L = 1000$ $H = 1999$	1	0000 9999
A1	$LL = 0+(1-0)*0.5 = 0.5$ $HH = 0+(1-0)*1.0 = 1.0$	$L = 5000$ $H = 9999$		5000 9999
A1	$LL = 0.5+(1-0.5)*0.5 = 0.75$ $HH = 0.5+(1-0.5)*1.0 = 1$	$L = 7500$ $H = 9999$		7500 9999
A5	$LL = 0.75+(1-0.75)*0.0 = 0.75$ $HH = 0.75+(1-0.75)*0.1 = 0.775$	$L = 7500$ $H = 7749$	7	5000 7499
A4	$LL = 0.5+(0.75-0.5)*0.1 = 0.525$ $HH = 0.5+(0.75-0.5)*0.5 = 0.55$	$L = 5250$ $H = 5499$	5	2500 4999
A3	$LL = 0.25+(0.5-0.25)*0.2 = 0.3$ $HH = 0.25+(0.5-0.25)*0.4 = 0.35$	$L = 3000$ $H = 3499$	3	0000 4999
A1	$LL = 0+(0.5-0)*0.5 = 0.25$ $HH = 0+(0.5-0)*1.0 = 0.5$	$L = 2500$ $H = 4999$		2500 4999
A1	$LL = 0.25+(0.5-0.25)*0.5 = 0.375$ $HH = 0.25+(0.5-0.25)*1.0 = 0.5$	$L = 3750$ $H = 4999$	3750	4999

การนำการเข้ารหัสเลขคณิตไปใช้ในการบีบอัดข้อมูลที่ได้จากอัลกอริทึม SPIHT ที่ผ่านการปรับปรุงนั้น ลักษณะข้อมูลของอัลกอริทึม SPIHT ที่ผ่านการปรับปรุงจะมีค่าเป็น 0 หรือ 1 เท่านั้น ดังนั้นสัญลักษณ์ในการนำไปเข้ารหัสเลขคณิตจะมีอยู่ด้วยกัน 2 สัญลักษณ์เท่านั้น



ขั้นตอนการถอดรหัส

ขั้นตอนที่ 1 ทำการกำหนดช่วงขอบเขตเริ่มต้น [L,H) เป็นขอบเขตแบบจำนวนเต็มค่า [L,H) มีค่าเป็น [0000,9999)

ขั้นตอนที่ 2 คำนวณค่า index ได้จากสมการดังนี้

$$\text{index} = ((\text{code} - L + 1) * \text{Maxfreq} - 1) / (H - L + 1) \quad (5.8)$$

เมื่อ index คือ ค่าชี้ตำแหน่งของสัญลักษณ์

code คือ ข้อมูลที่ใช้ถอดรหัส

Maxfreq คือ จำนวนข้อมูลทั้งหมดที่นำมาเข้ารหัส

L และ H คือ ขอบเขตการเข้ารหัส

ขั้นตอนที่ 3 ใช้ index ในการหาสัญลักษณ์ โดยนำค่าตัวแรกของ index ที่ได้ไปเทียบกับความถี่สะสมของสัญลักษณ์

เช่น

$$\text{index} = 7.75 \text{ ค่าที่ใช้ในการหาสัญลักษณ์คือ } 7$$

นำไปเทียบกับความถี่สะสมจากรายการเข้ารหัส จะเห็นได้ว่า 7 อยู่ในช่วง [5,10) ดังนั้นสัญลักษณ์ที่ได้จาก index เท่ากับ 7.75 คือ "A1"

ขั้นตอนที่ 4 ทำการปรับช่วงขอบเขตใหม่ที่สอดคล้องกับสัญลักษณ์ที่ได้จากขั้นตอนที่ 3 แล้วนำช่วงขอบเขตดังกล่าวมาแทนให้เป็นช่วงขอบเขตปัจจุบันแทน ซึ่งมีสมการในการปรับช่วงขอบเขตใหม่ดังนี้

$$L = L^{\wedge} + (H^{\wedge} - L^{\wedge} + 1) * P_L(S) \quad (5.9)$$

$$H = [L^{\wedge} + (H^{\wedge} - L^{\wedge} + 1) * P_H(S)] - 1 \quad (5.10)$$

ขั้นตอนที่ 4 ถ้าตัวเลขตำแหน่งแรกทางซ้ายของ L และ H เป็นค่าเดียวกัน ให้ทำการตัดตัวเลขตำแหน่งแรกทางซ้ายของ L และ H ออกและทำการเลื่อนตำแหน่งถัดมาขึ้นมาแทน ส่วนตำแหน่งสุดท้ายจะมีการใส่ค่าคือ ค่าของ L จะใส่ 0 เป็นค่าสุดท้าย และค่าของ H จะใส่ 9 เป็นตำแหน่งสุดท้าย นอกจากนั้นยังต้องทำการปรับค่าของ code ด้วยโดยทำการตัดตัวเลขตำแหน่งแรกทางซ้ายของ code ออกและทำการเลื่อนตำแหน่งถัดมาขึ้นมาแทนให้ครบจำนวนดังเดิม

เช่น

ข้อมูลการถอดรหัสคือ 37159456

$L = 3750$ ,  $H = 3999$  และ  $\text{code} = 3715$

ทำการปรับค่า  $L$ ,  $H$  และ  $\text{code}$  ใหม่ได้คือ

$L = 7500$ ,  $H = 9999$  และ  $\text{code} = 7159$

ขั้นตอนที่ 5 เช็การสิ้นสุดชุดของข้อมูลที่นำมาถอดรหัส โดยพิจารณาจากเอาต์พุตที่ได้จากการถอดรหัสมีจำนวนเท่ากับ  $\text{Maxfreq}$  หรือยัง ถ้าเท่ากันแล้วก็ให้หยุด แต่ถ้ายังไม่เท่าให้กลับไปทำขั้นตอนที่ 2

ตัวอย่างการถอดรหัสเลขคณิต

ถ้ากำหนดให้ข้อมูลที่จะนำมาถอดรหัสเลขคณิตคือค่าเอาต์พุตที่ได้จากตารางที่ 5.2 ซึ่งมีค่าคือ 717533750 มีข้อมูลของสัญลักษณ์ที่ใช้ถอดรหัสเป็นดังตารางที่ 5.3 และมีขั้นตอนการแสดงผลการถอดรหัสเลขคณิตเป็นดังตารางที่ 5.4

ตารางที่ 5.3 ข้อมูลของสัญลักษณ์ที่ใช้ถอดรหัส

สัญลักษณ์	$[P_L(S), P_H(S))$	ความถี่สะสม
A1	[0.5,1)	[5,10)
A2	[0.4,0.5)	[4,5)
A3	[0.2,0.4)	[2,4)
A4	[0.1,0.2)	[1,2)
A5	[0.0,0.1)	[0,1)

จำนวนสัญลักษณ์ทั้งหมดเท่ากับ 10 ( $\text{Maxfreq} = 10$ )

ตารางที่ 5.4 ขั้นตอนการถอดรหัสเลขคณิต

code	Index	ช่วงขอบเขต	เอาต์พุต	การปรับค่า
7175	$[(7175-0+1)*10-1] / (9999-0+1) = 7.18 \rightarrow 7$	$L = 0+(9999-0+1)*0.5 = 5000$ $H = [0+(9999-0+1)*1]-1 = 9999$	A1	-
7175	$[(7175-5000+1)*10-1] / (9999-5000+1) = 4.35 \rightarrow 4$	$L = 5000+(9999-5000+1)*0.4 = \mathbf{7000}$ $H = [5000+(9999-5000+1)*0.5]-1 = \mathbf{7499}$	A2	code = 1753 L = 0000 H = 4999
1753	$[(1753-0+1)*10-1] / (4999-0+1) = 3.51 \rightarrow 3$	$L = 0+(4999-0+1)*0.2 = \mathbf{1000}$ $H = [0+(4999-0+1)*0.4]-1 = \mathbf{1999}$	A3	code = 7533 L = 0000 H = 9999
7533	$[(7533-0+1)*10-1] / (9999-0+1) = 7.53 \rightarrow 7$	$L = 0+(9999-0+1)*0.5 = 5000$ $H = [0+(9999-0+1)*1]-1 = 9999$	A1	-
7533	$[(7533-5000+1)*10-1] / (9999-5000+1) = 5.07 \rightarrow 5$	$L = 5000+(9999-5000+1)*0.5 = 7500$ $H = [5000+(9999-5000+1)*1]-1 = 9999$	A1	-
7533	$[(7533-7500+1)*10-1] / (9999-7500+1) = 0.14 \rightarrow 0$	$L = 7500+(9999-7500+1)*0 = \mathbf{7500}$ $H = [7500+(9999-7500+1)*0.1]-1 = \mathbf{7749}$	A5	code = 5337 L = 5000 H = 7499
5337	$[(5337-5000+1)*10-1] / (7499-5000+1) = 1.35 \rightarrow 1$	$L = 5000+(7499-5000+1)*0.1 = \mathbf{5250}$ $H = [5000+(7499-5000+1)*0.2]-1 = \mathbf{5499}$	A4	code = 3375 L = 2500 H = 4999
3375	$[(3375-2500+1)*10-1] / (4999-2500+1) = 3.50 \rightarrow 3$	$L = 2500+(4999-2500+1)*0.2 = \mathbf{3000}$ $H = [2500+(4999-2500+1)*0.4]-1 = \mathbf{3499}$	A3	code = 3750 L = 0000 H = 4999
3750	$[(3750-0+1)*10-1] / (4999-0+1) = 7.50 \rightarrow 7$	$L = 0+(4999-0+1)*0.5 = 2500$ $H = [0+(4999-0+1)*1]-1 = 4999$	A1	-
3750	$[(3750-2500+1)*10-1] / (4999-2500+1) = 5.00 \rightarrow 5$	$L = 2500+(4999-2500+1)*0.5 = 3750$ $H = [2500+(4999-2500+1)*1]-1 = 4999$	A1	-

## 5.5 สรุป

ในบทนี้กล่าวถึงการเข้ารหัสและถอดรหัสด้วยอัลกอริทึม SPIHT และกล่าวถึงการพัฒนาอัลกอริทึม SPIHT ด้วยการเพิ่มเงื่อนไขการเช็คการส่ง (k,l) เข้า LIP คือถ้ามีการส่ง (k,l) ทั้งหมดของ  $O(i,j)$  แสดงว่าได้เกิดกรณีนี้ขึ้น ดังนั้นสามารถทำการส่ง (k,l) เข้า LIS ให้เป็นชนิด A ได้เลย เพื่อลดการส่งข้อมูลในเงื่อนไข  $O(i,j)$  ไม่มีระดับความสำคัญแต่  $L(i,j)$  มีระดับความสำคัญ และทำการเพิ่ม

LFC และหาค่า FO และ FL เพื่อนำไปตัดสินใจว่าจะทำการลดการส่งข้อมูลในเงื่อนไข  $O(i,j)$  มีระดับความสำคัญแต่  $L(i,j)$  ไม่มีระดับความสำคัญหรือไม่ นอกจากนั้นยังกล่าวถึงการนำการเข้ารหัสเลขคณิตมาใช้กับข้อมูลที่ได้จากอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง เพื่อทำการเพิ่มประสิทธิภาพการบีบอัดข้อมูลให้สูงขึ้น

## บทที่ 6

### ผลการทดลอง

#### 6.1 บทนำ

การบีบอัดข้อมูลภาพ EM ด้วยการแปลงเวฟเล็ดแบบดิสครีท แล้วทำการเข้ารหัสด้วยอัลกอริทึม SPIHT และทำการเพิ่มประสิทธิภาพการบีบอัดข้อมูล ด้วยการนำข้อมูลที่ได้จากอัลกอริทึม SPIHT ไปทำการเข้ารหัสเลขคณิต ในการแปลงเวฟเล็ดถ้ามีการเปลี่ยนเวฟเล็ดแม่ ก็จะทำให้ค่าสัมประสิทธิ์เวฟเล็ดที่ได้แตกต่างกันไป ซึ่งก็จะทำให้ประสิทธิภาพการบีบอัดข้อมูลภาพแตกต่างกันไปด้วย ดังนั้นจึงต้องมีการทดสอบหาเวฟเล็ดแม่ ที่ทำให้ผลการบีบอัดข้อมูลภาพ EM ด้วยอัลกอริทึม SPIHT และอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง ให้ได้ผลการบีบอัดที่ดีที่สุด และทำการทดสอบอัลกอริทึม SPIHT และอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง และทำการหาข้อสรุปในการหาระดับอัตราบิตที่เหมาะสมของการบีบอัดข้อมูลภาพ EM จากการประเมินผลด้วยแบบสอบถาม

#### 6.2 การหาตระกูลเวฟเล็ดแม่

ในขั้นตอนการแปลงเวฟเล็ดการใช้เวฟเล็ดแม่ที่ต่างกัน ย่อมทำให้ได้สัมประสิทธิ์เวฟเล็ดที่ไม่เหมือนกัน ซึ่งทำให้ได้ผลการบีบอัดข้อมูลภาพ EM ที่อัตราการบีบอัดข้อมูลเดียวกันแตกต่างกันออกไป ดังนั้นเพื่อให้ได้ผลการบีบอัดข้อมูลภาพ EM ที่ดีที่สุด ด้วยอัลกอริทึม SPIHT และอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง จึงต้องทำการทดสอบหาตระกูลเวฟเล็ดแม่ที่เหมาะสมสำหรับภาพ EM

ตระกูลเวฟเล็ดแม่ที่ใช้ในการทดสอบ คือ Biorthogonal9-7 (bi9-7), Daubechies4 (db4), Symlets8 (sym8) และ Coiflets5 (coif5) โดยแต่ละตระกูลเวฟเล็ดแม่มีค่าดังนี้

ตระกูล bi9-7 ตัวฟิลเตอร์  $h_0(n)$ ,  $h_1(n)$ ,  $g_0(n)$  และ  $g_1(n)$  มีความสัมพันธ์กันดังนี้ (Jabran, 2001)

$$g_0(n) = (-1)^{(n+1)} h_1(n) \quad (6.1)$$

$$g_1(n) = (-1)^n h_0(n) \quad (6.2)$$

ฟิลเตอร์  $h_0(n)$  ตระกูล bi9-7 (Marc et al., 1992) คือ

[0.03782879857992, -0.02384929751586, -0.11062402748951, 0.37740268810913,  
0.85269865321930, 0.37740268810913, -0.11062402748951, -0.02384929751586,  
0.03782879857992]

ฟิลเตอร์  $h_1(n)$  ตระกูล bi9-7 (Marc et al., 1992) คือ

[0.06453905013246, -0.04068975261660, -0.41809244072573, 0.7884848722061,  
-0.41809244072573, -0.04068975261660, 0.06453905013246]

ตระกูล db4, sym8 และ coif5 ตัวฟิลเตอร์  $h_0(n)$ ,  $h_1(n)$ ,  $g_0(n)$  และ  $g_1(n)$  มีความสัมพันธ์กัน  
ดังนี้ (Jabran, 2001)

$$g_0(n) = h_0(-n) \quad (6.3)$$

$$g_1(n) = (-1)^n h_0(n) \quad (6.4)$$

$$h_1(n) = (-1)^{n+1} h_0(-n) \quad (6.5)$$

ฟิลเตอร์  $h_0(n)$  ตระกูล db4 (Michel et al., 1996) คือ

[-0.01059740178500, 0.03288301166698, 0.03084138183599, -0.18703481171888,  
-0.02798376941698, 0.63088076792959, 0.71484657055254, 0.23037781330886]

ฟิลเตอร์  $h_0(n)$  ตระกูล sym8 (Michel et al., 1996) คือ

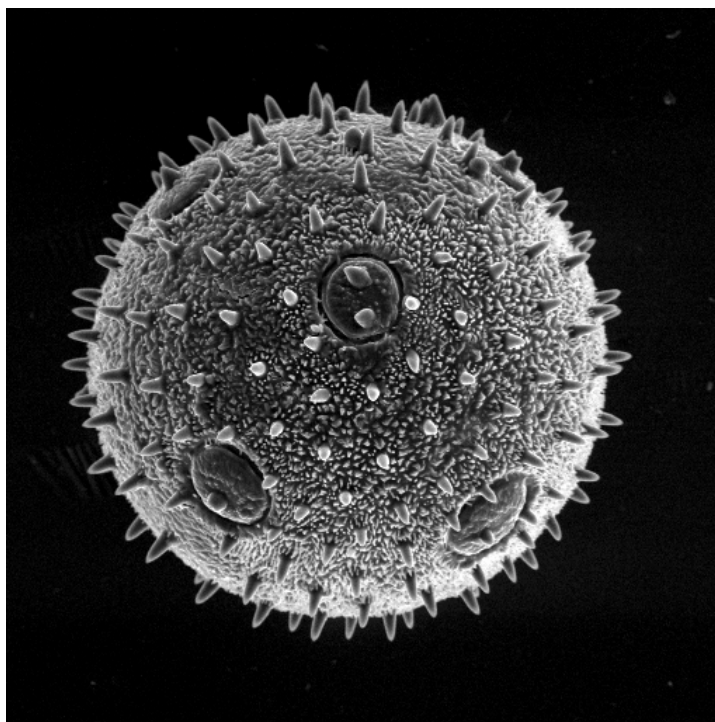
[-0.00338241595101, -0.00054213233179, 0.03169508781149, 0.00760748732492,  
-0.14329423835081, -0.06127335906766, 0.48135965125837, 0.77718575170052,  
0.36444189483533, -0.05194583810771, -0.02721902991706, 0.04913717967361,  
0.00380875201389, -0.01495225833705, -0.00030292051472, 0.00188995033276]

ฟิลเตอร์  $h_0(n)$  ตระกูล coif5 (Michel et al., 1996) คือ

[-0.00000009517657, -0.00000016744289, 0.00000206376185, 0.00000373465518,  
-0.00002131502681, -0.00004134043227, 0.00014054114970, 0.00030225958181,  
-0.00063813134305, -0.00166286370201, 0.00243337321266, 0.00676418544805,  
-0.00916423116248, -0.01976177894257, 0.03268357426711, 0.04128920875018,  
-0.10557420870334, -0.06203596396290, 0.43799162617184, 0.77428960365296,

0.42156620669085, -0.05204316317624, -0.09192001055970, 0.02816802897094,  
 0.02340815678584, -0.01013111751985, -0.00415935878139, 0.00217823635811,  
 0.00035858968790, -0.00021208083980]

ภาพต้นแบบที่ใช้ทดสอบคือ ภาพ pollen of zucchini (ละอองเกสรของ zucchini) เป็นภาพตัวอย่างจากกล้อง EM รุ่น JEM 2010 ของบริษัท JEOL LTD ดังแสดงในภาพที่ 6.1 ซึ่งมีข้อมูลภาพขนาด 512x512 พิกเซล และแต่ละพิกเซลมี 8 บิต และทำการทดสอบที่อัตราบิต 0.5, 1, 1.5 และ 2 bpp และทำการวัดประสิทธิภาพด้วยวิธี PSNR



ภาพที่ 6.1 ภาพต้นแบบ (pollen of zucchini)

การเลือกใช้ภาพ pollen of zucchini เป็นภาพทดสอบหาตระกูลเวฟเล็ต เนื่องจากภาพนี้เป็นภาพตัวอย่างที่มาจากบริษัท ซึ่งถือได้ว่าเป็นภาพที่ถ่ายได้คุณภาพ และภาพนี้เป็นภาพที่มีรายละเอียดภาพสูง ทำให้เหมาะเป็นอย่างยิ่งที่จะใช้ในการทดสอบหาตระกูลเวฟเล็ตแม่

### 6.2.1 ผลการทดสอบหาตระกูลเวฟเล็ตแม่

ผลการทดสอบหาตระกูลเวฟเล็ตแม่ที่ให้ผลการบีบอัดข้อมูลภาพ EM ได้ดีที่สุด พบว่าทั้งอัลกอริทึม SPIHT และอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง ที่ใช้ bi9-7 เป็นเวฟเล็ตแม่ในการแปลงเวฟเล็ต ให้ผลการบีบอัดข้อมูลภาพ EM ได้ดีที่สุด เป็นดังตารางที่ 6.1

ตารางที่ 6.1 แสดงผลการบีบอัดข้อมูลภาพ EM ที่ตระกูลเวฟเล็ตแม่ต่างๆ

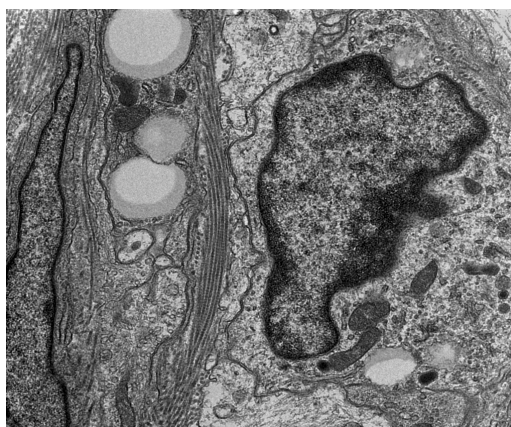
ตระกูลเวฟเล็ตแม่	อัตราบิต (bpp)	PSNR (dB)	
		SPIHT	SPIHT*
bi9-7	0.5	25.9437	26.4209
	1	30.7854	31.4849
	1.5	35.0367	35.2730
	2	38.4622	38.7493
db4	0.5	25.6627	25.7600
	1	30.0797	30.4873
	1.5	30.1865	34.7129
	2	37.3909	38.3268
sym8	0.5	25.7788	25.8553
	1	30.1865	30.5499
	1.5	33.9292	34.6861
	2	37.3499	38.2538
coif5	0.5	25.5975	25.7016
	1	29.9516	30.2969
	1.5	33.6847	34.4694
	2	37.0617	38.0273

SPIHT\* คือ SPIHT ที่ผ่านการปรับปรุง

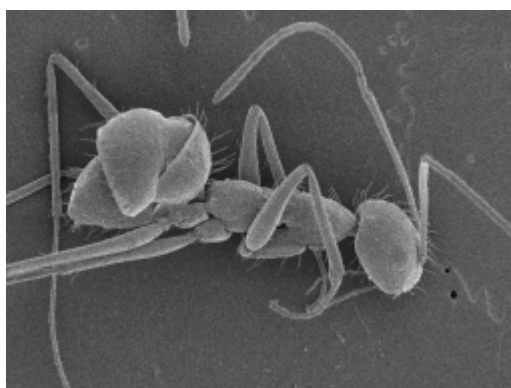


### 6.3 การบีบอัดข้อมูลภาพด้วยอัลกอริทึม SPIHT และอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง

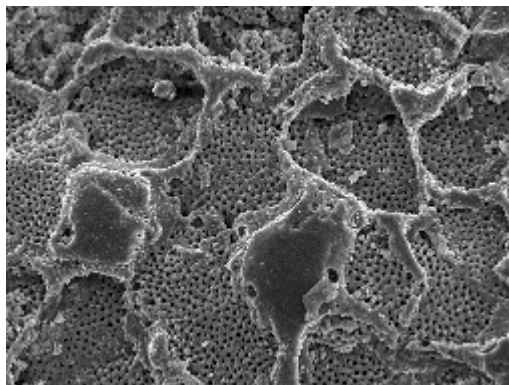
ทำการทดสอบอัลกอริทึม SPIHT และอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง ที่อัตราบิตเดียวกันคือ 0.125, 0.25, 0.5 และ 1 bpp โดยใช้ข้อมูลของภาพต้นแบบจำนวน 5 ภาพ คือ ภาพ pollen of zucchini, ภาพ skin (เป็นภาพตัวอย่างจากกล้อง EM รุ่น JEM 2010 ของบริษัท JEOL LTD) ซึ่งมีข้อมูลภาพขนาด 832x1024 พิกเซล และแต่ละพิกเซลมี 8 บิต และภาพ sand, ภาพ ant และภาพ LED (เป็นภาพที่ถ่ายจากกล้อง EM รุ่น JSM-5800 LV ของบริษัท JEOL LTD) ซึ่งมีข้อมูลภาพขนาด 960x1280 พิกเซล และแต่ละพิกเซลมี 8 บิต ซึ่งภาพที่นำมาทดสอบยกเว้นภาพ pollen of zucchini ได้แสดงไว้ในภาพที่ 6.2 (a-d)



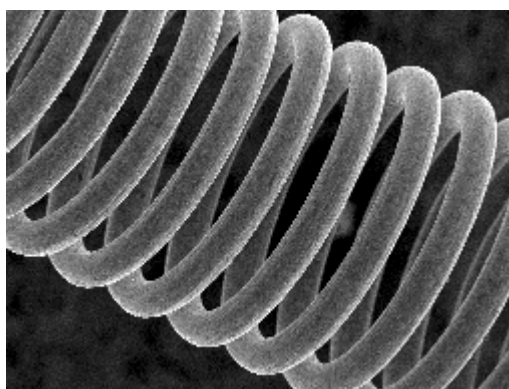
ภาพที่ 6.2.a (skin)



ภาพที่ 6.2.b (ant)



ภาพที่ 6.2.c (sand)



ภาพที่ 6.2.d (LED)

ภาพที่ 6.2 ภาพต้นแบบที่ใช้ทดสอบการบีบอัดข้อมูล

### 6.3.1 ผลการทดสอบการบีบอัดข้อมูลภาพด้วยอัลกอริทึม SPIHT และอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง

ผลการทดสอบการเปรียบเทียบประสิทธิภาพการบีบอัดข้อมูลภาพ EM ด้วยอัลกอริทึม SPIHT และอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง พบว่าอัลกอริทึม SPIHT ที่ผ่านการปรับปรุงให้ผลการบีบอัดข้อมูลภาพ EM ได้ดีกว่า เป็นดังตารางที่ 6.2 และมีภาพที่ผ่านการบีบอัดข้อมูลด้วยอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง เป็นดังภาคผนวก ก

ตารางที่ 6.2 แสดงผลการบีบอัดข้อมูลภาพ EM ด้วยอัลกอริทึม SPIHT  
และอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง

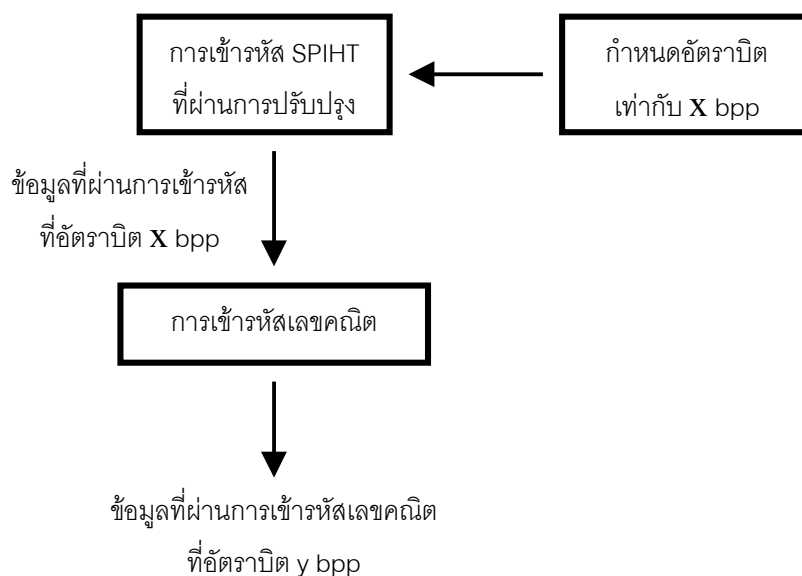
ภาพ	อัตราบิต (bpp)	ขนาดของไฟล์ภาพ ที่ผ่านการบีบอัดข้อมูล	PSNR (dB)	
			SPIHT	SPIHT*
pollen of zucchini	0.125	1/64 ของไฟล์ภาพต้นแบบ	21.0101	21.2506
	0.25	1/32 ของไฟล์ภาพต้นแบบ	22.9363	23.1409
	0.5	1/16 ของไฟล์ภาพต้นแบบ	25.9437	26.4209
	1	1/8 ของไฟล์ภาพต้นแบบ	30.7854	31.4849
skin	0.125	1/64 ของไฟล์ภาพต้นแบบ	20.0459	20.2261
	0.25	1/32 ของไฟล์ภาพต้นแบบ	21.4673	21.6107
	0.5	1/16 ของไฟล์ภาพต้นแบบ	23.6564	23.8868
	1	1/8 ของไฟล์ภาพต้นแบบ	25.9389	26.1148
sand	0.125	1/64 ของไฟล์ภาพต้นแบบ	21.8326	22.1093
	0.25	1/32 ของไฟล์ภาพต้นแบบ	23.3655	23.5302
	0.5	1/16 ของไฟล์ภาพต้นแบบ	25.1323	25.3597
	1	1/8 ของไฟล์ภาพต้นแบบ	27.7760	28.0055
ant	0.125	1/64 ของไฟล์ภาพต้นแบบ	34.1146	31.3520
	0.25	1/32 ของไฟล์ภาพต้นแบบ	34.5026	34.7356
	0.5	1/16 ของไฟล์ภาพต้นแบบ	35.4126	35.4504
	1	1/8 ของไฟล์ภาพต้นแบบ	37.1842	37.2121
LED	0.125	1/64 ของไฟล์ภาพต้นแบบ	26.3506	26.4125
	0.25	1/32 ของไฟล์ภาพต้นแบบ	27.5024	27.6934
	0.5	1/16 ของไฟล์ภาพต้นแบบ	28.8230	29.0162
	1	1/8 ของไฟล์ภาพต้นแบบ	30.7168	30.9553

SPIHT\* คือ SPIHT ที่ผ่านการปรับปรุง

#### 6.4 การบีบอัดข้อมูลภาพด้วยอัลกอริทึม SPIHT ที่ผ่านการปรับปรุงและการเข้ารหัสเลขคณิต

ข้อมูลที่ได้จากการบีบอัดข้อมูลภาพด้วยอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง จะมีค่าเท่ากับ 0 หรือ 1 ซึ่งสามารถนำมาบีบอัดข้อมูลแบบไม่มีการสูญเสียได้อีก เพื่อที่จะได้เพิ่มประสิทธิภาพการบีบอัดข้อมูลให้ได้เพิ่มมากขึ้น และในงานวิจัยนี้เลือกใช้การเข้ารหัสเลขคณิต ซึ่งการเข้ารหัสเลขคณิตนี้ให้ผลการบีบอัดข้อมูลได้ดีกว่าการเข้ารหัสแบบฮัฟแมน (I. H. Witten , R. M. Neal and J. G. Cleary, 1987)

ทำการทดสอบการบีบอัดข้อมูลที่ได้จากการบีบอัดข้อมูลภาพอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง ด้วยการนำมาเข้ารหัสเลขคณิตของภาพต้นแบบจำนวน 5 ภาพคือ ภาพ pollen of zucchini, ภาพ skin, ภาพ sand, ภาพ ant และภาพ LED โดยแต่ละภาพจะทำการทดสอบการบีบอัดข้อมูลที่อัตราบิต 0.125, 0.25, 0.5 และ 1 bpp มีขั้นตอนการทดสอบเป็นดังภาพที่ 6.3



ภาพที่ 6.3 แผนภาพแสดงขั้นตอนการทดสอบการเข้ารหัสเลขคณิต

##### 6.4.1 ผลการทดสอบการบีบอัดข้อมูลภาพด้วยอัลกอริทึม SPIHT ที่ผ่านการปรับปรุงและการเข้ารหัสเลขคณิต

ผลการทดสอบการบีบอัดข้อมูลภาพ EM ด้วยอัลกอริทึม SPIHT ผ่านการปรับปรุงและการเข้ารหัสเลขคณิต พบว่าการเข้ารหัสเลขคณิตสามารถช่วยลดขนาดข้อมูลที่ได้จากอัลกอริทึม SPIHT ที่ผ่านการปรับปรุงได้ ซึ่งจะทำให้ประสิทธิภาพการบีบอัดข้อมูลภาพ EM เพิ่มมากขึ้น เป็นดังตารางที่ 6.3

ตารางที่ 6.3 แสดงผลการบีบอัดข้อมูลภาพ EM ด้วยอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง และการเข้ารหัสเลขคณิต

ภาพ	อัตราบิต (bpp)		% การลดของ ขนาดข้อมูล SPIHT*
	SPIHT*	SPIHT*+	
pollen of zucchini	0.125	0.1215	2.7832 %
	0.25	0.2408	3.6743 %
	0.5	0.4816	3.8696 %
	1	0.9682	3.1707 %
skin	0.125	0.1216	2.7118 %
	0.25	0.2401	3.9250%
	0.5	0.4793	4.1278 %
	1	0.9647	3.5231%
sand	0.125	0.1201	3.8593 %
	0.25	0.2402	3.8906 %
	0.5	0.4804	3.9075 %
	1	0.9521	4.7819 %
ant	0.125	0.1083	13.3229 %
	0.25	0.2458	1.6458 %
	0.5	0.4527	9.4414 %
	1	0.9686	3.1380 %
LED	0.125	0.1232	1.4114 %
	0.25	0.2428	2.8984 %
	0.5	0.4782	5.6341 %
	1	0.9701	2.9882 %

SPIHT\* คือ SPIHT ที่ผ่านการปรับปรุง

SPIHT\*+ คือ SPIHT ที่ผ่านการปรับปรุงและการเข้ารหัสเลขคณิต

## 6.5 การหาข้อสรุปในการหาระดับอัตราบิด ในการบีบอัดข้อมูลภาพ EM

ในการบีบอัดข้อมูลภาพที่ระดับอัตราบิดต่างๆ ภาพที่ได้จากการบีบอัดย่อมมีคุณภาพของภาพแตกต่างกัน นั่นคือภาพจากการบีบอัดที่ระดับอัตราบิดสูง ภาพที่ได้ย่อมมีคุณภาพหรือมีลักษณะใกล้เคียงกับภาพต้นแบบมากกว่าภาพจากการบีบอัดที่ระดับอัตราบิดต่ำกว่า ดังนั้นเพื่อให้ทราบอัตราบิดที่เหมาะสมกับการบีบอัดข้อมูลภาพ EM จึงต้องดำเนินการหาข้อสรุปในการหาระดับที่เหมาะสมกับการบีบอัดข้อมูลภาพ EM ด้วยวิธีการประเมินผลจากแบบสอบถามจากกลุ่มเป้าหมายที่ปฏิบัติงานเกี่ยวข้องกับกล้องจุลทรรศน์อิเล็กตรอนคือ อาจารย์ นักวิทยาศาสตร์ นักวิจัย ช่างภาพการแพทย์ เป็นจำนวนทั้งสิ้น 12 ท่าน

ในการประเมินผลจากแบบสอบถามมีการประเมินดังนี้

1. การประเมินผลในการระบุคุณภาพของภาพมีเกณฑ์การพิจารณา และการให้คะแนนตามคุณภาพของภาพดังนี้

1) หลักเกณฑ์ในการพิจารณาและคำจำกัดความ

ก. ความสว่างของภาพ หมายถึง ระดับของปริมาณแสงที่ดวงตาจับได้โดยไม่เกี่ยวข้องกับสี

ข. ความคมชัดของภาพ หมายถึง ความชัดเจนของลายเส้น หรือขอบภาพ

ค. ความเปรียบต่าง (contrast) หมายถึง ความแตกต่างในความสว่างที่ทำให้สามารถแยกพื้นที่แสดงภาพที่ต่างกันออกจากกันได้

ง. รายละเอียดของภาพ หมายถึง ข้อมูลที่ได้จากภาพ ที่สามารถนำไปใช้งานได้

2) เกณฑ์การให้คะแนนประมาณค่าแบ่งออกเป็น 4 ระดับดังนี้

4 หมายถึง รูปภาพนั้นมีคุณภาพ ดีมาก

3 หมายถึง รูปภาพนั้นมีคุณภาพ ดี

2 หมายถึง รูปภาพนั้นมีคุณภาพ พอใช้

1 หมายถึง รูปภาพนั้นมีคุณภาพ ใช้ไม่ได้

2. การประเมินผลในการนำไปใช้ทดแทนภาพต้นแบบได้มีเกณฑ์ในการประเมินออกเป็น 3 ระดับดังนี้

- ใช้ทดแทนภาพต้นแบบได้อย่างดี

- พอจะใช้ทดแทนภาพต้นแบบได้

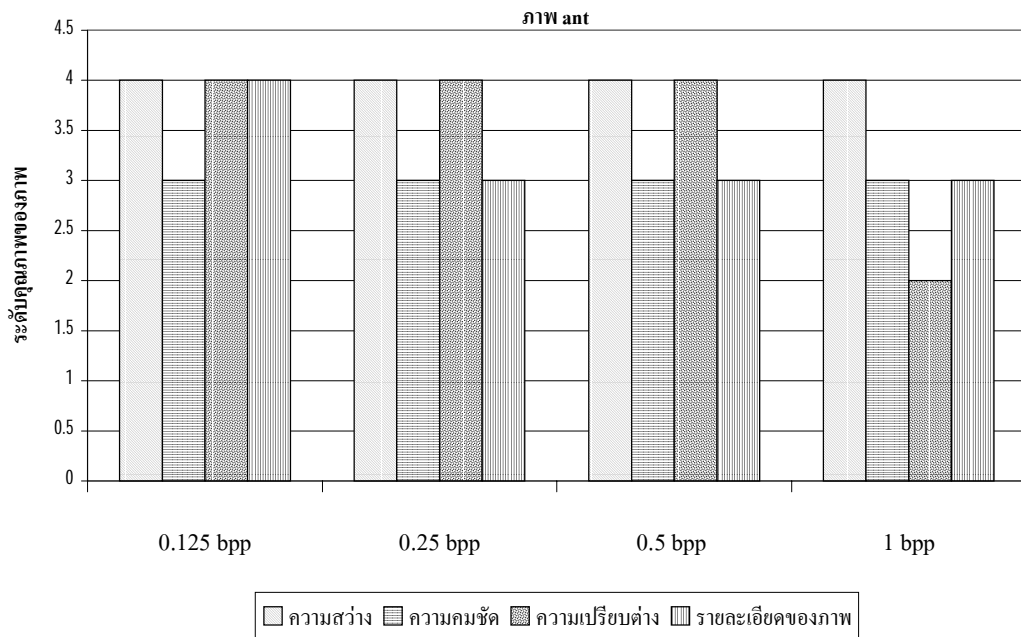
- ไม่อาจยอมให้ใช้ทดแทนภาพต้นแบบได้

ทำการหาข้อสรุปจากภาพที่ผ่านการบีบอัดข้อมูลที่ได้จากการบีบอัดข้อมูลภาพอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง ของภาพต้นแบบจำนวน 5 ภาพคือ ภาพ pollen of zucchini, ภาพ skin, ภาพ sand, ภาพ ant และภาพ LED โดยแต่ละภาพจะทำการบีบอัดข้อมูลที่อัตราบิต 0.125, 0.25, 0.5 และ 1 bpp

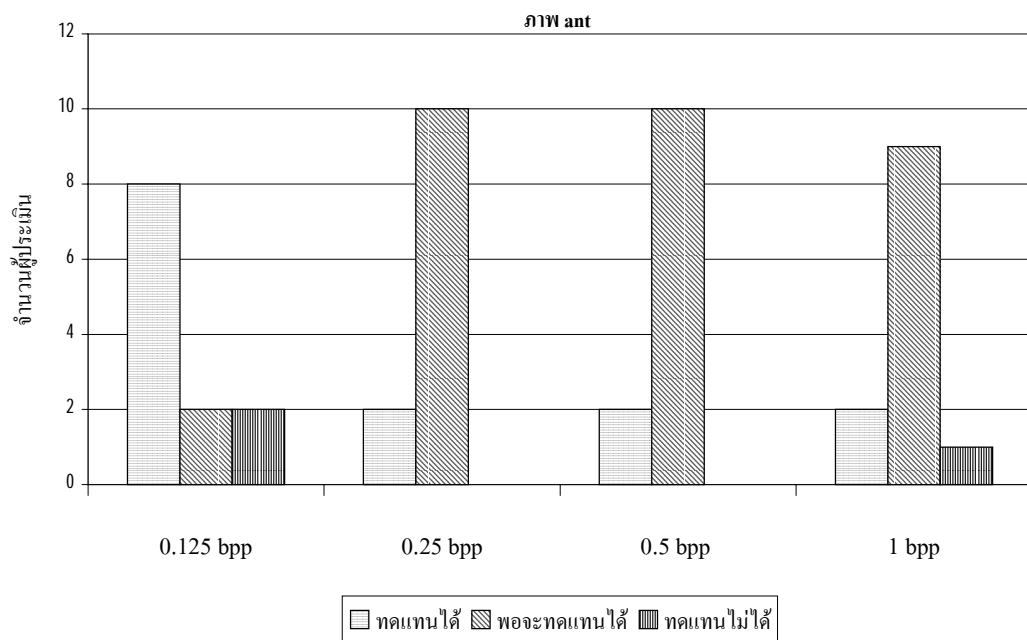
การแสดงผลภาพที่ผ่านการบีบอัดข้อมูล ที่ใช้ในการประเมินจะแสดงผลภาพโดยการพิมพ์ด้วยเครื่องพิมพ์รุ่น HP LaserJet 5P ของบริษัท Hewlett Packard ซึ่งสามารถดูได้ในภาพผนวก ก

### 6.5.1 ผลการหาข้อสรุปในการหาระดับอัตราบิต ในการบีบอัดข้อมูลภาพ EM

ผลการประเมินของภาพ pollen of zucchini, ภาพ skin, ภาพ sand, ภาพ ant และภาพ LED เป็นดังภาพที่ 6.4-6.8 ตามลำดับ ซึ่งข้อมูลภาพที่ 6.4-6.8 เป็นผลสรุปมาจากผลการประเมินจากแบบสอบถามซึ่งอยู่ในภาคผนวก ข จากภาพที่ 6.4-6.8 จะเห็นได้ว่าระดับอัตราบิตของภาพที่ผ่านการบีบอัดข้อมูลที่พอจะนำมาใช้ทดแทนภาพต้นแบบได้นั้นมีความแตกต่างกันคือ ภาพ ant จะอยู่ที่อัตราบิตเท่ากับ 0.125 bpp ภาพ sand จะอยู่ที่อัตราบิตเท่ากับ 0.5 bpp ภาพ LED จะอยู่ที่อัตราบิตเท่ากับ 0.5 bpp ภาพ skin จะอยู่ที่อัตราบิตเท่ากับ 0.5 bpp และภาพ pollen of zucchini จะอยู่ที่อัตราบิตเท่ากับ 1 bpp จากผลที่กล่าวมาจะเห็นได้ว่าอัตราบิตที่เหมาะสมแต่ละภาพจะแตกต่างกัน ดังนั้นในการหาข้อสรุปของแต่ละภาพเพื่อหาข้อสรุปรวมทั้งหมดจึงไม่สามารถหาข้อสรุปได้ แต่ถ้าแบ่งภาพที่ใช้ในการทดสอบออกเป็น 2 กลุ่มคือ กลุ่มแรกเป็นภาพที่มีรายละเอียดต่ำประกอบด้วยภาพ ant และภาพ LED กลุ่มที่สองเป็นกลุ่มภาพที่มีรายละเอียดสูงประกอบด้วยภาพ pollen of zucchini, ภาพ skin และภาพ sand และทำการรวมผลการประเมินของภาพในแต่ละกลุ่ม จะได้ผลการประเมินของภาพที่มีรายละเอียดต่ำเป็นดังภาพที่ 6.9 ซึ่งข้อมูลภาพที่ 6.9 เป็นผลสรุปมาจากผลการประเมินจากแบบสอบถามซึ่งอยู่ในภาคผนวก ข จากภาพที่ 6.9 จะเห็นได้ว่าอัตราบิตของภาพที่ผ่านการบีบอัดข้อมูลที่พอจะนำมาใช้ทดแทนภาพต้นแบบได้จะอยู่ที่อัตราบิตเท่ากับ 0.25 bpp ซึ่งภาพนี้จะมีแสงสว่างอยู่ในระดับคุณภาพดีมาก มีความคมชัดอยู่ในระดับคุณภาพดี มีความเปรียบต่างอยู่ในระดับดี และมีรายละเอียดภาพอยู่ในระดับพอใช้ และผลการประเมินของภาพที่มีรายละเอียดสูงเป็นดังภาพที่ 6.10 ซึ่งข้อมูลภาพที่ 6.10 เป็นผลสรุปมาจากผลการประเมินจากแบบสอบถามซึ่งอยู่ในภาคผนวก ข จากภาพที่ 6.10 จะเห็นได้ว่าอัตราบิตของภาพที่ผ่านการบีบอัดข้อมูลที่พอจะนำมาใช้ทดแทนภาพต้นแบบได้จะอยู่ที่อัตราบิตเท่ากับ 1 bpp ซึ่งภาพนี้จะมีแสงสว่างอยู่ในระดับคุณภาพดีมาก มีความคมชัดอยู่ในระดับคุณภาพดี มีความเปรียบต่างอยู่ในระดับดี และมีรายละเอียดภาพอยู่ในระดับดี



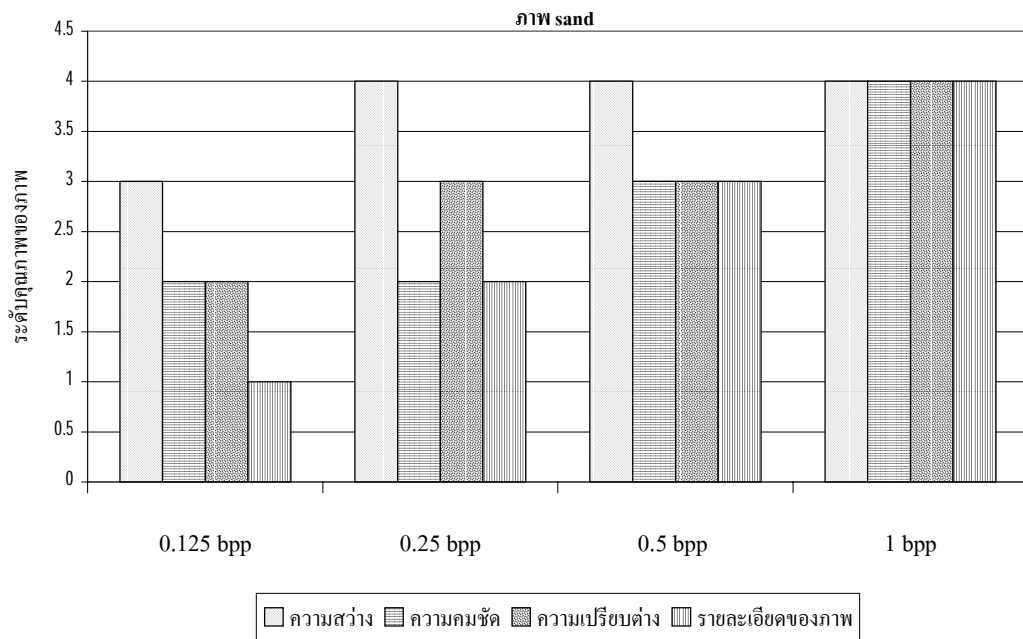
ภาพที่ 6.4.1 แผนภูมิแสดงผลการประเมินในการระบุคุณภาพของภาพ ant



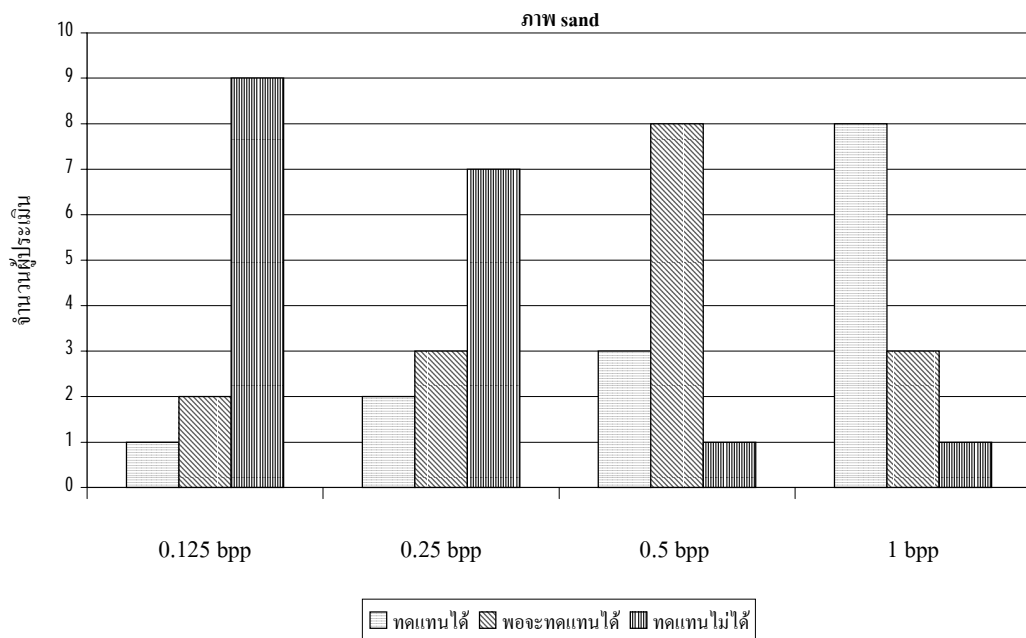
ภาพที่ 6.4.2 แผนภูมิแสดงผลการประเมินในการนำไปใช้ทดแทนภาพต้นแบบของภาพ ant

ภาพที่ 6.4 แผนภูมิแสดงผลการประเมินของภาพ ant



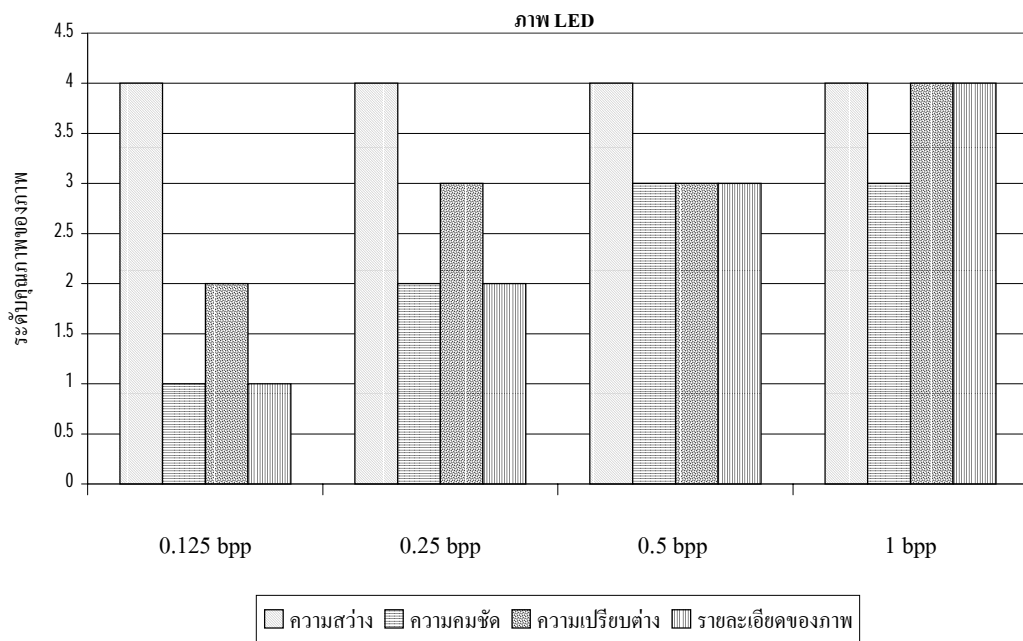


ภาพที่ 6.5.1 แผนภูมิแสดงผลการประเมินในการระบุคุณภาพของภาพ sand

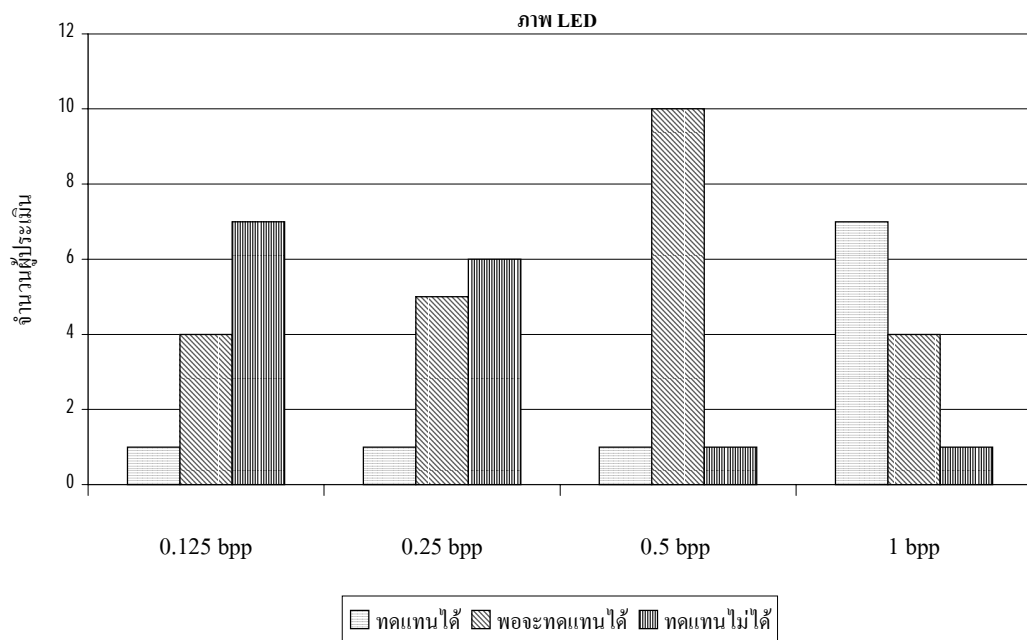


ภาพที่ 6.5.2 แผนภูมิแสดงผลการประเมินในการนำไปใช้ทดแทนภาพต้นแบบของภาพ sand

ภาพที่ 6.5 แผนภูมิแสดงผลการประเมินของภาพ sand

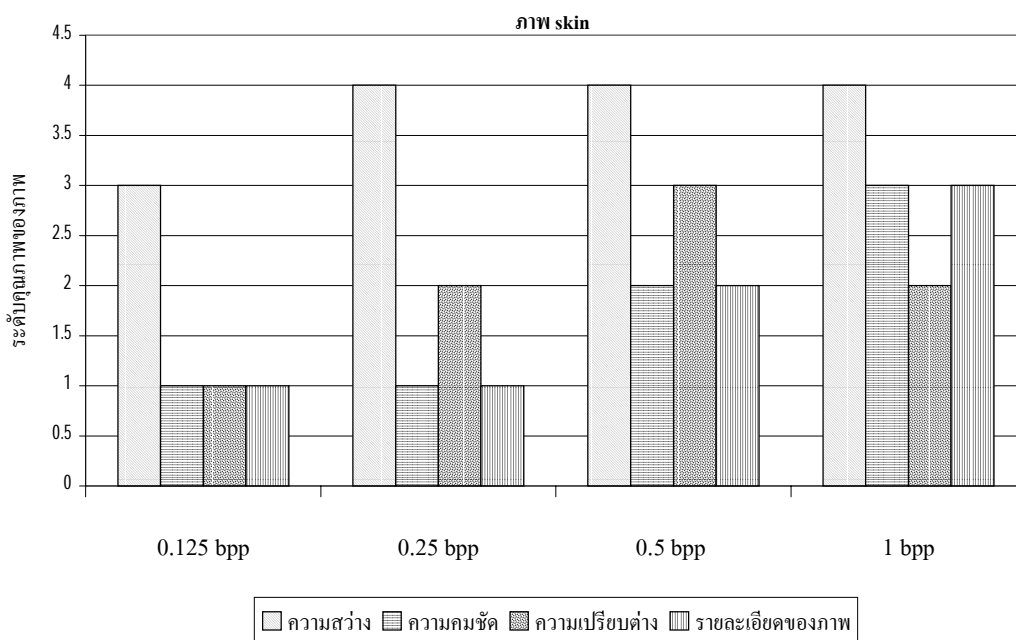


ภาพที่ 6.6.1 แผนภูมิแสดงผลการประเมินในการระบุคุณภาพของภาพ LED

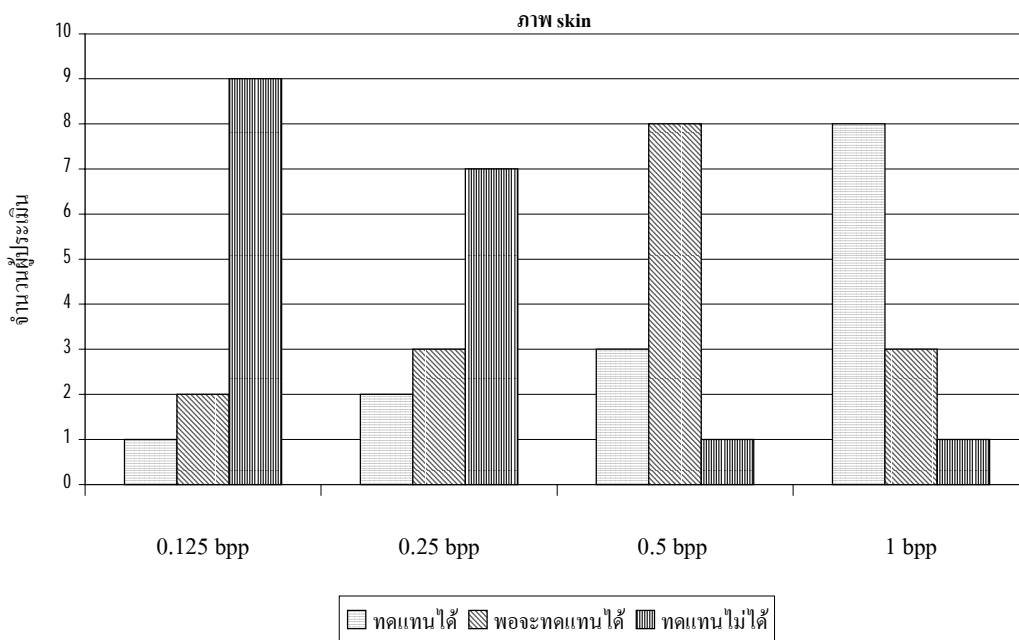


ภาพที่ 6.6.2 แผนภูมิแสดงผลการประเมินในการนำไปใช้ทดแทนภาพต้นแบบของภาพ LED

ภาพที่ 6.6 แผนภูมิแสดงผลการประเมินของภาพ LED

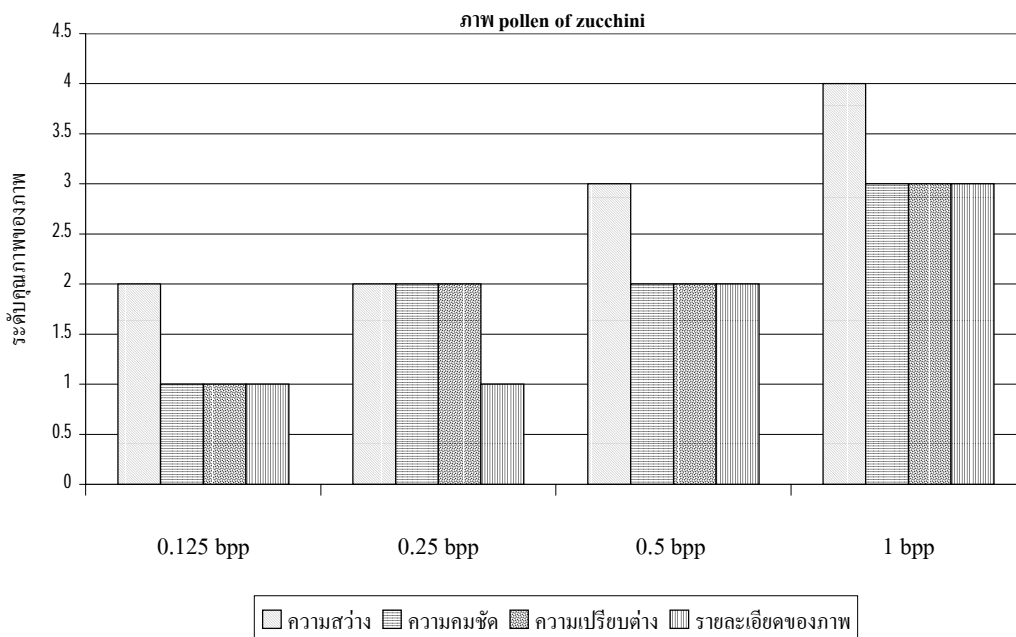


ภาพที่ 6.7.1 แผนภูมิแสดงผลการประเมินในการระบุคุณภาพของภาพ skin

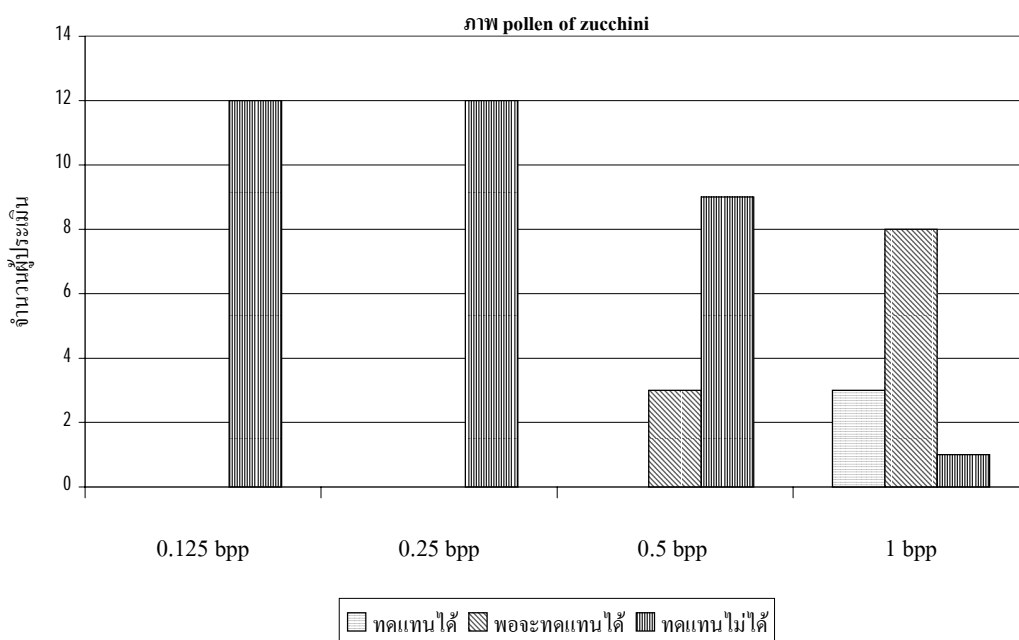


ภาพที่ 6.7.2 แผนภูมิแสดงผลการประเมินในการนำไปใช้ทดแทนภาพต้นแบบของภาพ skin

ภาพที่ 6.7 แผนภูมิแสดงผลการประเมินของภาพ skin

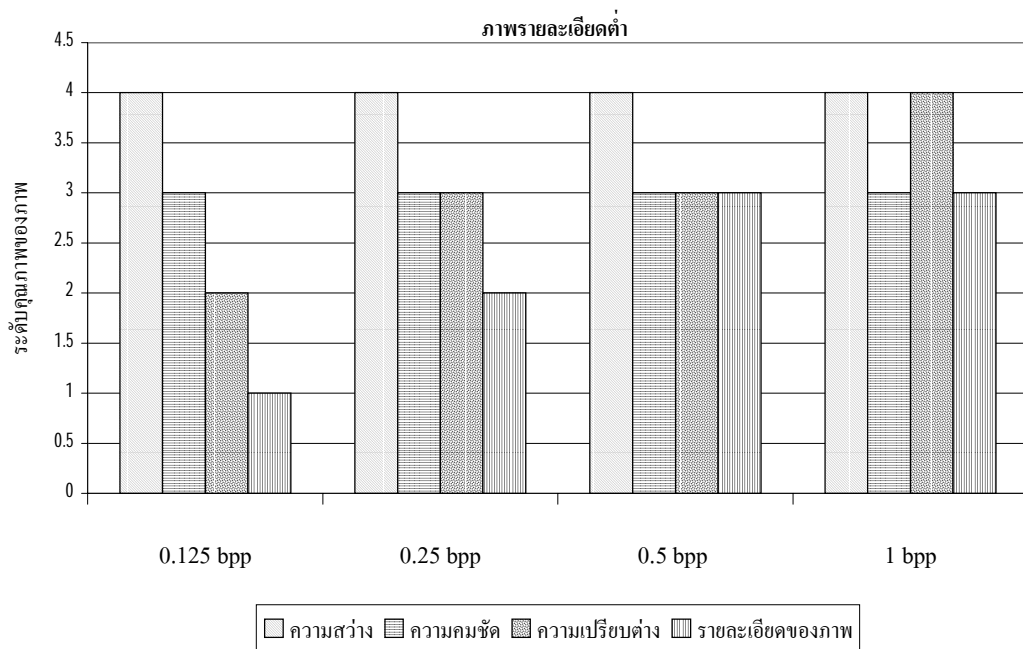


ภาพที่ 6.8.1 แผนภูมิแสดงผลการประเมินในการระบุคุณภาพของภาพ pollen of zucchini

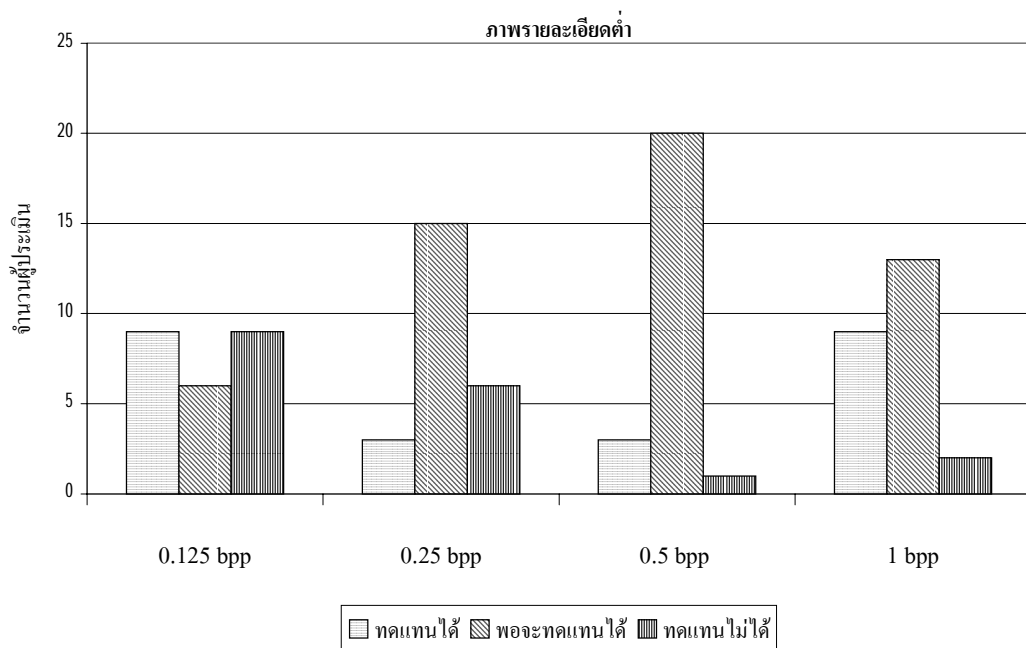


ภาพที่ 6.8.2 แผนภูมิแสดงผลการประเมินในการนำไปใช้ทดแทนภาพต้นแบบของภาพ pollen of zucchini

ภาพที่ 6.8 แผนภูมิแสดงผลการประเมินของภาพ pollen of zucchini

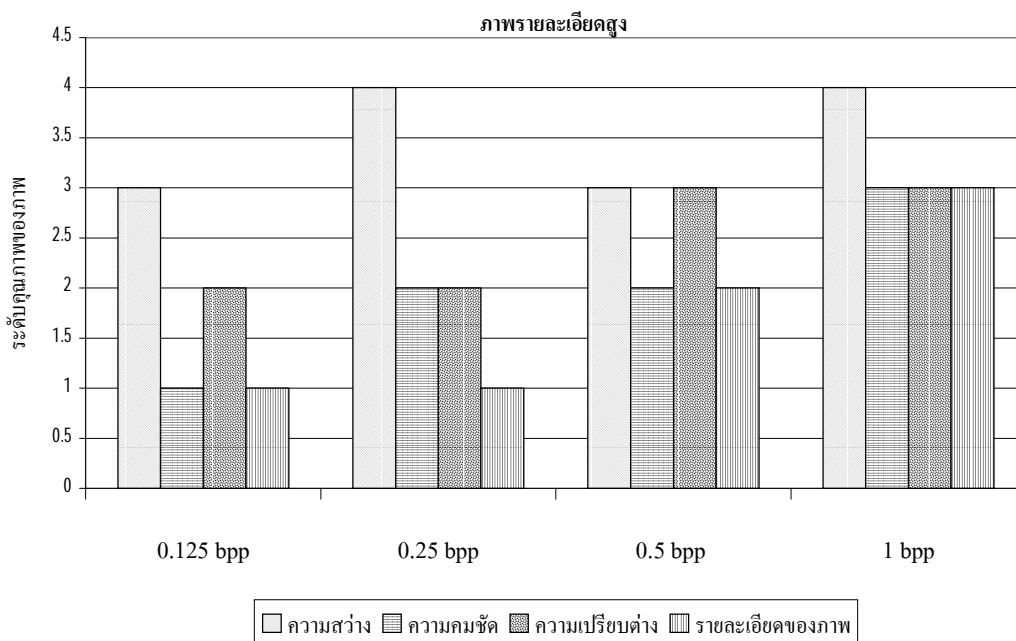


ภาพที่ 6.9.1 แผนภูมิแสดงผลการประเมินในการระบุคุณภาพของภาพรายละเอียดต่ำ

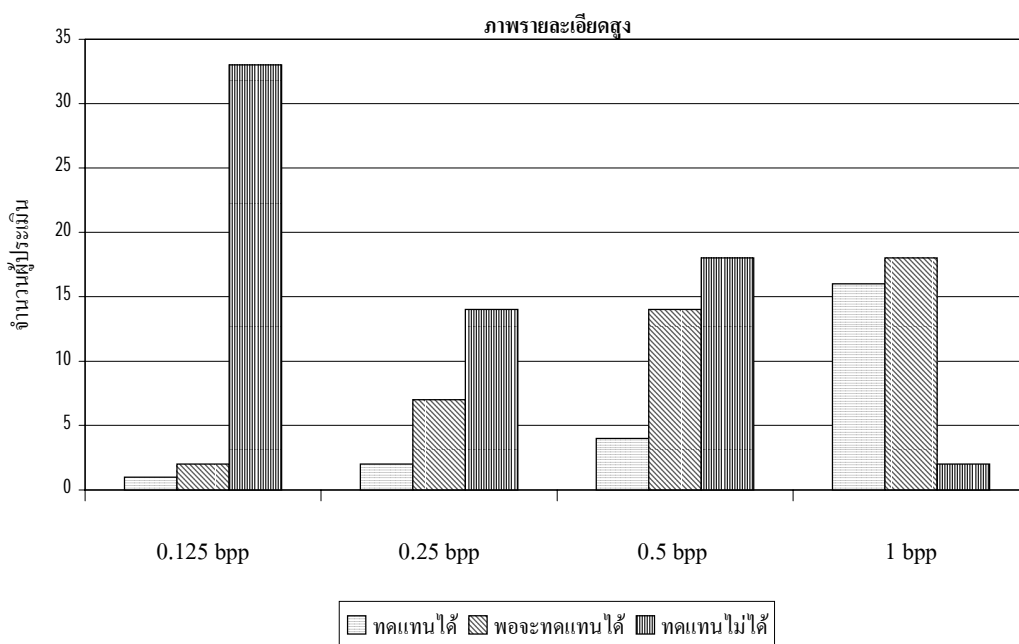


ภาพที่ 6.9.2 แผนภูมิแสดงผลการประเมินในการนำไปใช้ทดแทนภาพต้นแบบของภาพรายละเอียดต่ำ

ภาพที่ 6.9 แผนภูมิแสดงผลการประเมินของภาพรายละเอียดต่ำ



ภาพที่ 6.10.1 แผนภูมิแสดงผลการประเมินในการระบุคุณภาพของภาพรายละเอียดสูง



ภาพที่ 6.10.2 แผนภูมิแสดงผลการประเมินในการนำไปใช้ทดแทนภาพต้นแบบของภาพรายละเอียดสูง

ภาพที่ 6.10 แผนภูมิแสดงผลการประเมินของภาพรายละเอียดสูง

## 6.6 สรุป

จากผลการทดสอบหาตระกูลเวฟเล็ตแม่ที่ให้ผลการบีบอัดข้อมูลภาพ EM ได้ดีที่สุด พบว่าการใช้ bi9-7 เป็นเวฟเล็ตแม่ให้ผลการบีบอัดข้อมูลภาพ EM ได้ดีที่สุด และจากผลการทดสอบการเปรียบเทียบประสิทธิภาพการบีบอัดข้อมูลภาพ EM ด้วยอัลกอริทึม SPIHT และอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง พบว่าอัลกอริทึม SPIHT ที่ผ่านการปรับปรุงให้ผลการบีบอัดข้อมูลภาพ EM ได้ดีกว่า และจากการหาข้อสรุปในการหาระดับอัตราบิตของภาพที่ผ่านการบีบอัดข้อมูล ที่พอจะนำมาใช้ทดแทนภาพต้นแบบได้ แสดงได้ดังตารางที่ 6.4

ตารางที่ 6.4 แสดงระดับอัตราบิตที่เหมาะสมในการบีบอัดข้อมูลภาพ EM

ประเภทของภาพ	อัตราบิต (bpp)	ระดับคุณภาพของภาพ			
		ความสว่าง	ความคมชัด	ความเปรียบต่าง	รายละเอียดภาพ
ภาพรายละเอียดต่ำ	0.25	ดีมาก	ดี	ดี	พอใช้
ภาพรายละเอียดสูง	1	ดีมาก	ดี	ดี	ดี

ระดับอัตราบิตที่แสดงนี้เป็นอัตราบิตที่ต่ำที่สุด ที่ทำให้ภาพที่ผ่านการบีบอัดข้อมูลพอที่จะนำมาใช้แทนภาพต้นแบบได้ แต่ถ้าต้องการให้ได้ผลดีกว่านี้ควรเพิ่มอัตราบิตให้สูงขึ้นก็จะทำให้คุณภาพของภาพดีขึ้น และสามารถเพิ่มประสิทธิภาพของการบีบอัดข้อมูลภาพ EM ด้วยอัลกอริทึม SPIHT ผ่านการปรับปรุงได้อีกด้วยการนำมาเข้ารหัสเลขคณิต ซึ่งจะทำให้ขนาดของข้อมูลที่ได้เล็กลง

## บทที่ 7

### สรุปและข้อเสนอแนะ

#### 7.1 บทนำ

วิทยานิพนธ์นี้ได้ทำการศึกษาการบีบอัดข้อมูลภาพ EM ซึ่งภาพ EM เป็นภาพที่มีรายละเอียดมาก และไฟล์ภาพจะมีขนาดใหญ่ จำเป็นจะต้องใช้เนื้อที่ในการบันทึกข้อมูลสูง ดังนั้นถ้ามีการนำภาพ EM มาบีบอัดข้อมูล จะทำให้ช่วยลดปัญหานี้ได้ งานวิจัยนี้นำเสนอวิธีการบีบอัดข้อมูลภาพที่เหมาะสมสำหรับภาพ EM ด้วยวิธีการแปลงเวฟเล็ดแบบดิสครีท โดยภาพ EM เมื่อนำมาแปลงด้วยวิธีเวฟเล็ดแบบดิสครีท ข้อมูลที่เป็นพื้นผิววัตถุจะอยู่ที่สัมประสิทธิ์ที่แบนด์ย่อยความถี่ต่ำ และข้อมูลที่เป็นขอบหรือลายเส้นของวัตถุจะอยู่ที่สัมประสิทธิ์ที่แบนด์ย่อยความถี่สูง ดังนั้นการบีบอัดข้อมูลภาพ EM ด้วยการแปลงเวฟเล็ดจะต้องให้ความสำคัญกับสัมประสิทธิ์ทุกแบนด์ย่อย เพื่อให้ภาพที่ได้จากการบีบอัดมีความใกล้เคียงกับภาพต้นแบบมากที่สุด อัลกอริทึม SPIHT เป็นอัลกอริทึมหนึ่งที่เหมาะสม เนื่องจากอัลกอริทึมนี้เข้ารหัสโดยให้ความสำคัญกับขนาดของสัมประสิทธิ์และไม่สนใจว่าสัมประสิทธิ์ตัวนั้นจะอยู่ในระดับแบนด์ย่อยใด ในงานวิจัยนี้ยังเสนอวิธีการพัฒนาอัลกอริทึม SPIHT โดยการเพิ่ม LFC และเพิ่มเงื่อนไขการเข้ารหัสและการถอดรหัส และทำการหาตระกูลเวฟเล็ดแม่ที่ให้ผลการบีบอัดข้อมูลภาพ EM ได้ดีที่สุด นอกจากนี้ยังทำการหาข้อสรุปในการหาระดับอัตราบิตที่เหมาะสมของการบีบอัดข้อมูลภาพ EM จากการประเมินผลด้วยแบบสอบถาม นอกจากนี้วิทยานิพนธ์นี้ยังเพิ่มประสิทธิภาพของการบีบอัดข้อมูลภาพ EM ด้วยอัลกอริทึม SPIHT ที่ผ่านการปรับปรุงได้อีก ด้วยการนำข้อมูลที่ผ่านการบีบอัดด้วยอัลกอริทึม SPIHT ที่ผ่านการปรับปรุงมาเข้ารหัสเลขคณิต

#### 7.2 สรุป

การดำเนินงานวิจัยวิทยานิพนธ์นี้บรรลุวัตถุประสงค์ ตามที่ตั้งไว้ทุกประการ มีผลการวิจัยคือ การบีบอัดภาพ EM ด้วยการแปลงเวฟเล็ดแบบดิสครีท ที่ใช้ bi9-7 เป็นเวฟเล็ดแม่ในการแปลงเวฟเล็ด แล้วนำสัมประสิทธิ์เวฟเล็ดมาเข้ารหัสด้วยอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง จะให้ผลการบีบอัดข้อมูลภาพ EM ได้ดีที่สุด และจากการประเมินผลด้วยแบบสอบถามเพื่อหาอัตราบิตที่เหมาะสมสำหรับการบีบอัดภาพ EM ซึ่งสรุปผลได้ว่า อัตราบิตที่ได้จากการบีบอัดภาพ EM ด้วย



อัลกอริทึม SPIHT ที่ผ่านการปรับปรุง ภาพที่มีรายละเอียดภาพต่ำจะมีอัตราบิตเท่ากับ 0.25 bpp หรือขนาดของไฟล์ภาพที่ผ่านการบีบอัดข้อมูลมีขนาดเท่ากับ 1/32 ของไฟล์ภาพต้นแบบ และภาพที่มีรายละเอียดสูงจะมีอัตราบิตเท่ากับ 1 bpp หรือขนาดของไฟล์ภาพที่ผ่านการบีบอัดข้อมูลมีขนาดเท่ากับ 1/8 ของไฟล์ภาพต้นแบบ ซึ่งภาพที่ผ่านการบีบอัดข้อมูลที่อัตราบิตดังกล่าวจะเป็นภาพที่พอจะนำมาใช้ทดแทนภาพต้นแบบได้ แต่ถ้าต้องการให้ภาพที่ได้มีคุณภาพดีขึ้นกว่านี้จะต้องเพิ่มอัตราบิตให้สูงขึ้น และการนำข้อมูลที่ผ่านการบีบอัดด้วยอัลกอริทึม SPIHT ที่ผ่านการปรับปรุงมาเข้ารหัสเลขคณิต จะทำให้ขนาดของข้อมูลที่ได้จากการเข้ารหัสมีขนาดเล็กลงเฉลี่ยเท่ากับ 4.235 %

### 7.3 ข้อเสนอแนะ

1. เนื่องจากโปรแกรมการอ่านและเขียนไฟล์ภาพที่ได้จากเครื่อง EM เขียนด้วยโปรแกรม MATLAB แต่โปรแกรมการบีบอัดข้อมูลภาพ เขียนด้วยโปรแกรม Visual C++ ซึ่งทำให้มีขั้นตอนในการบีบอัดเพิ่มขึ้น ดังนั้นเพื่อลดขั้นตอนนี้ลงควรทำการศึกษาโครงสร้างของไฟล์นามสกุล BMP ซึ่งเป็นนามสกุลของไฟล์ภาพที่ได้จากเครื่อง EM และเขียนโปรแกรมการอ่านและเขียนไฟล์ภาพที่ได้จากเครื่อง EM ด้วยโปรแกรม Visual C++

2. ในการประยุกต์ใช้กับภาพชนิดอื่นเช่น ภาพถ่ายจากดาวเทียม ภาพที่จะนำมาประยุกต์ใช้ต้องเป็นภาพระดับเทา ในกรณีต้องการบีบอัดภาพสีจะต้องทำการศึกษาเพิ่มเติม เพราะโครงสร้างของไฟล์ภาพสีกับภาพระดับเทามีโครงสร้างไม่เหมือนกัน

3. โปรแกรมการบีบอัดข้อมูลด้วยอัลกอริทึม SPIHT ที่ทำการปรับปรุงเป็นเพียงโปรแกรมที่เขียนเพื่อทดสอบอัลกอริทึม SPIHT ที่ทำการปรับปรุงเท่านั้น ทำให้โปรแกรมที่ได้จะใช้เวลาในการบีบอัดข้อมูลค่อนข้างนานเมื่อเทียบกับการบีบอัดข้อมูลด้วยวิธี JPEG ดังนั้นต้องทำการพัฒนาการเขียนโปรแกรมการบีบอัดข้อมูล เพื่อให้ใช้เวลาในการบีบอัดข้อมูลน้อยลง

## เอกสารอ้างอิง

- เกศกนก กฤตยาภาศิริวัฒน์ (2543). การบีบอัดข้อมูลภาพด้วยการเข้ารหัสเวกเตอร์ควอนไทซ์แบบอะแดปทีฟด้วยค่าสัมประสิทธิ์ที่ได้จากการแปลงเวฟเลต. วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบัง.
- ขวัญฤทัย ไพรีพ่ายฤทธิ์. (2542). ตระกูลเวฟเลตที่เหมาะสมสำหรับการเข้ารหัสแบบเอ็มเบดเด็ดซีโรเทรีเวฟเลตในการประยุกต์ใช้งานการแพทย์ทางไกล. วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี.
- ชัยชาญ มัคคูน (2539). การแก้ไขลักษณะของลายเส้นโดยใช้เวฟเลตทรานส์ฟอร์ม. วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบัง.
- เดโช ทองอร่าม และ สุวิทย์ ปุณณชัยยะ (2538). กระบวนการทางภาพและการวิเคราะห์ภาพสำหรับกล้องจุลทรรศน์อิเล็กตรอนแบบสแกน. วารสารเครื่องมือวิจัยวิทยาศาสตร์และเทคโนโลยี. 5 (1): 21-48.
- วิรุพห์ มัลลละวิรัช และ สุวิทย์ ปุณณชัยยะ (2534). กล้องจุลทรรศน์อิเล็กตรอน. วารสารเครื่องมือวิจัยวิทยาศาสตร์และเทคโนโลยี. 1 (2): 129-169.
- ศิริพร เดชะสิลารักษ์. (2543). การลดขอบบดบังในภาพ JPEG ด้วยวิธีเวฟเลตเรซโซลคิงที่ปรับตัวเองได้. วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบัง.
- สุขสันต์ จิรเชวง และ วุฒิพงศ์ อารกุล (2544). การปรับปรุงการบีบอัดภาพวิธี SPIHT โดยใช้การเข้ารหัสบล็อกสำหรับภาพลายเส้นคม. การประชุมวิชาการทางวิศวกรรมไฟฟ้าครั้งที่ 24 (EECON-24) สจล. 1214-1219.
- Al, B. (2000). **Handbook of image & video processing**. New York: Academic Press.
- Amir, S., and William, A. P. (1993). Image compression using the spatial-orientation tree. **IEEE Int. Symp. On Circuits and Systems, Chicaro, IL**, 279-282.

- Amir, S., and William, A. P. (1996). A new fast and efficient image codec based on set partitioning in hierarchical trees. **IEEE Transaction on Circuit and Systems for Video Technology**. 6 (3): 243-250.
- Brian, A. B., and Thomas, R. F. (2001). Quadtree classification and TCQ image coding. **IEEE Transactions on Circuit and Systems for Video Technology**. 11 (1): 3-8.
- Burrus, C. S., Ramesh, A. G., and Haitao, G. (1998). **Introduction to wavelets and wavelet transforms**. New York: Prentice-Hall International.
- David, S. (1998). **Data compression the complete reference**. London: Springer-Verlag.
- Ian, H. W., Radford, M. N., and John, G. C. (1987). Arithmetic coding for data compression. **Communication of the ACM**. 30 (6): 520-540.
- Jabran, A. (2001). **Optimization of biorthogonal wavelet filters for signal and image compression**. Degree of Candiatu Scientarium at the Department of Informatics: University of Oslo.
- Jerome, M. S. (1993). Embedded image coding using zerotrees of wavelets coefficients. **IEEE Transaction on Signal Processing**. 41 (12): 3445-3462.
- Marc, A., Michel, B., Pierre, M., and Ingrid, D. (1992). Image coding using wavelet transform. **IEEE Transactions on Image Processing**. 1 (2): 205-220.
- Michel, M., Yves, M., Georges, O., and Jean, M. P. (1996). **Wavelet toolbox for use with matlab**. New York: The MathWorks.
- Olivier, R., and Martin, V. (1991). Wavelets and signal processing. **IEEE SP Magazine**. 14-38.
- Paul, G. H., and Jeffrey, S. V. (1994). Arithmetic coding for data compression. **IEEE Proceedings of the IEEE**. 82 (6): 857-865.
- Rafael, G. G., and Richard, E. W. (1993). **Digital image processing**. New York: Addison Wesley.
- Stephane, G. M. (1989). A theory for multiresolution signal decomposition: The wavelet representation. **IEEE Transaction on Pattern and Machine Intelligence**. 11 (7): 674-693.
- Steven, R. (1992). **Coding and information theory**. London: Springer-Verlag.

ภาคผนวก ก

แบบสอบถามเพื่อประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลไฟล์ภาพ  
สำหรับการแสดงผลของกล้องจุลทรรศน์อิเล็กตรอน

**แบบสอบถามเพื่อประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลไฟล์ภาพ สำหรับการแสดงผล  
ของกล้องจุลทรรศน์อิเล็กตรอน**

แบบสอบถามนี้แบ่งเป็น 3 ตอน

**ตอนที่ 1** ข้อมูลส่วนบุคคล

**ตอนที่ 2** การระบุคุณภาพของภาพฯ และความหมาย

**ตอนที่ 3** แบบสอบถามการเปรียบเทียบภาพที่ผ่านการบีบอัดข้อมูลกับภาพต้นแบบ

**ตอนที่ 1** ข้อมูลส่วนบุคคล

โปรดกรอกข้อความลงในช่องว่างหรือกาเครื่องหมาย  ลงในช่อง ( )

1. เพศ

( ) ชาย

( ) หญิง

2. วุฒิการศึกษา

( ) ปริญญาตรี

( ) ปริญญาโท

( ) ปริญญาเอก

3. ตำแหน่ง.....

4. สังกัดภาควิชา/สาขาวิชา/ฝ่าย/กลุ่มงาน.....

.....

5. ประสบการณ์ทำงานที่เกี่ยวข้องกับกล้องจุลทรรศน์อิเล็กตรอน

( ) 1-5 ปี

( ) 6-10 ปี

( ) 10 ปีขึ้นไป

6. ภาระงานที่รับผิดชอบในปัจจุบัน.....

.....

## ตอนที่ 2: การระบุคุณภาพของภาพฯ และความหมาย

- 1) คำจำกัดความหลักเกณฑ์ในการพิจารณา
  - ก. ความสว่างของภาพ หมายถึง ระดับของปริมาณแสงที่ดวงตาจับได้โดยไม่เกี่ยวข้องกั  
สี
  - ข. ความคมชัดของภาพ หมายถึง ความชัดเจนของลายเส้น หรือขอบภาพ
  - ค. ความเปรียบต่าง (contrast) หมายถึง ความแตกต่างในความสว่างที่ทำให้สามารถแยก  
พื้นที่แสดงภาพที่ต่างกันออกจากกันได้
  - ง. รายละเอียดของภาพ หมายถึง ข้อมูลที่ได้จากภาพ ที่สามารถนำไปใช้งานได้
- 2) การให้คะแนน โดยพิจารณาตามลำดับคุณภาพของภาพตามความคิดของท่าน โดยใช้  
เกณฑ์การประมาณค่าแบ่งออกเป็น 4 ระดับดังนี้
  - 4 หมายถึง รูปภาพนั้นมีคุณภาพ ดีมาก
  - 3 หมายถึง รูปภาพนั้นมีคุณภาพ ดี
  - 2 หมายถึง รูปภาพนั้นมีคุณภาพ พอใช้
  - 1 หมายถึง รูปภาพนั้นมีคุณภาพ ใช้ไม่ได้

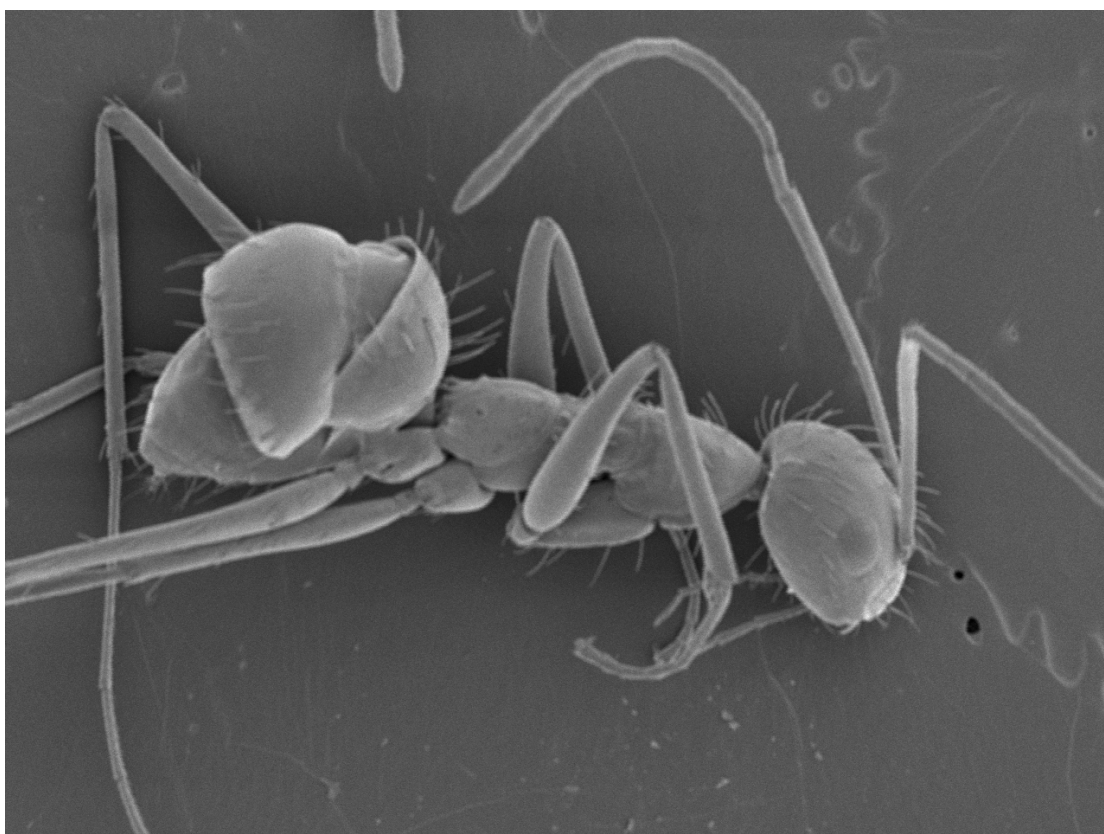
### หมายเหตุ:

- 1) ภาพแต่ละกลุ่มคือ ภาพต้นแบบดังนี้
  - ภาพกลุ่ม A คือ ภาพต้นแบบ ant
  - ภาพกลุ่ม B คือ ภาพต้นแบบ sand
  - ภาพกลุ่ม C คือ ภาพต้นแบบ LED
  - ภาพกลุ่ม D คือ ภาพต้นแบบ skin
  - ภาพกลุ่ม E คือ ภาพต้นแบบ pollen of zucchini
- 2) ภาพแต่ละภาพในแต่ละกลุ่มภาพคือ ภาพที่ผ่านการบีบอัดข้อมูลที่อัตราบิตดังนี้
  - ภาพที่ 1 คือภาพที่ผ่านการบีบอัดข้อมูลที่อัตราบิตเท่ากับ 0.125 bpp
  - ภาพที่ 2 คือภาพที่ผ่านการบีบอัดข้อมูลที่อัตราบิตเท่ากับ 0.25 bpp
  - ภาพที่ 3 คือภาพที่ผ่านการบีบอัดข้อมูลที่อัตราบิตเท่ากับ 0.50 bpp
  - ภาพที่ 4 คือภาพที่ผ่านการบีบอัดข้อมูลที่อัตราบิตเท่ากับ 1.00 bpp

ตอนที่3: แบบสอบถามการเปรียบเทียบภาพที่ผ่านการบีบอัดข้อมูลกับภาพต้นแบบ

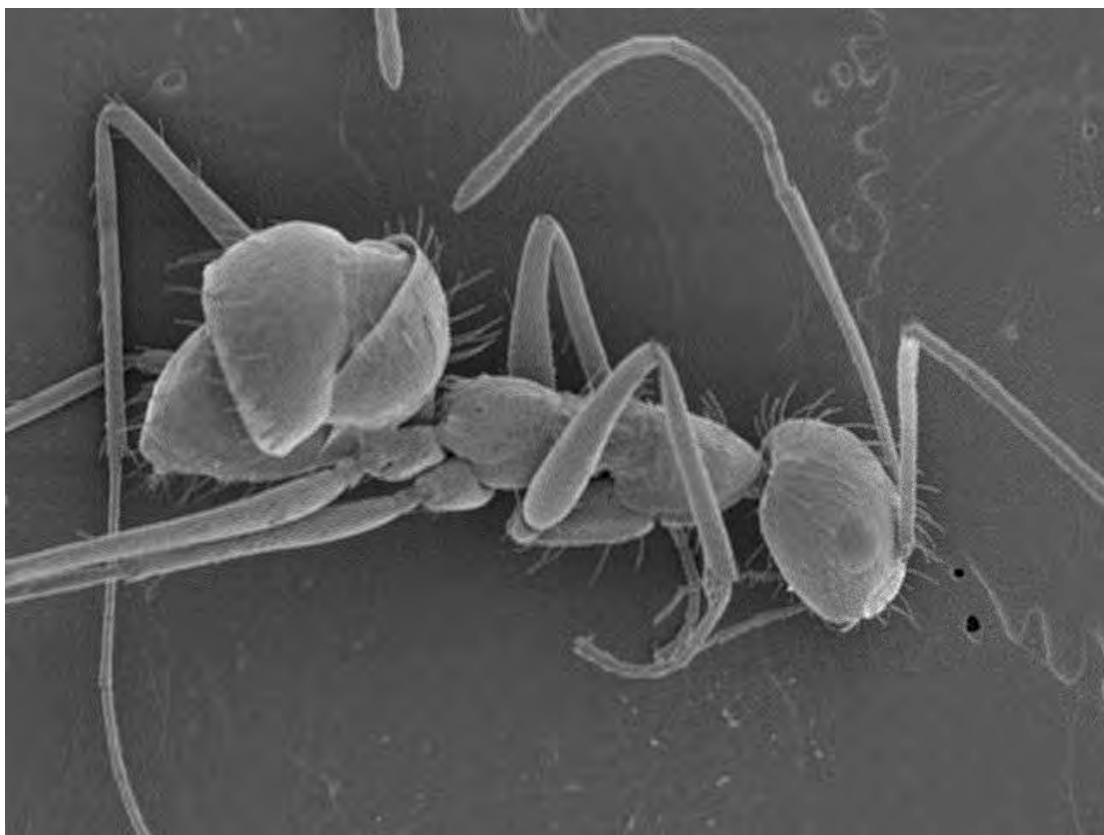
ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ไม่ผ่านการบีบอัดข้อมูล (ภาพต้นแบบ)

กลุ่ม A



ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล

กลุ่ม A ภาพที่ 1

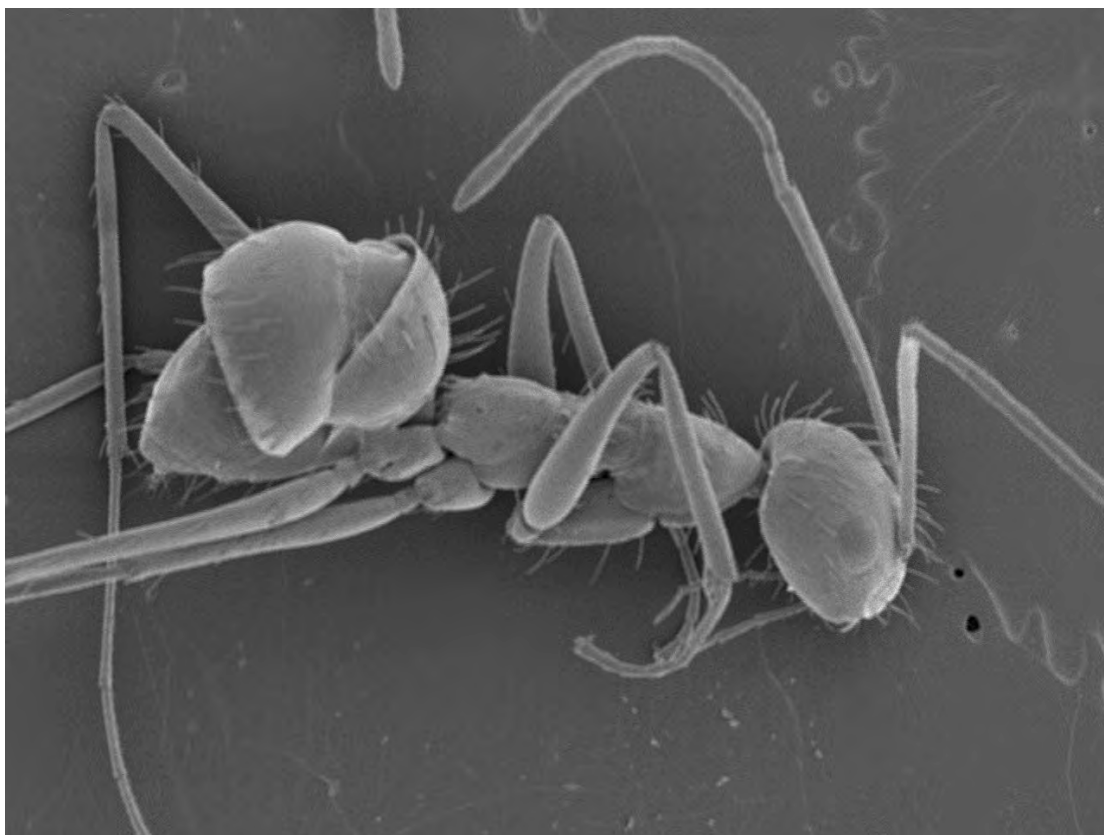


กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบเทียบ				
4. รายละเอียดของภาพ				



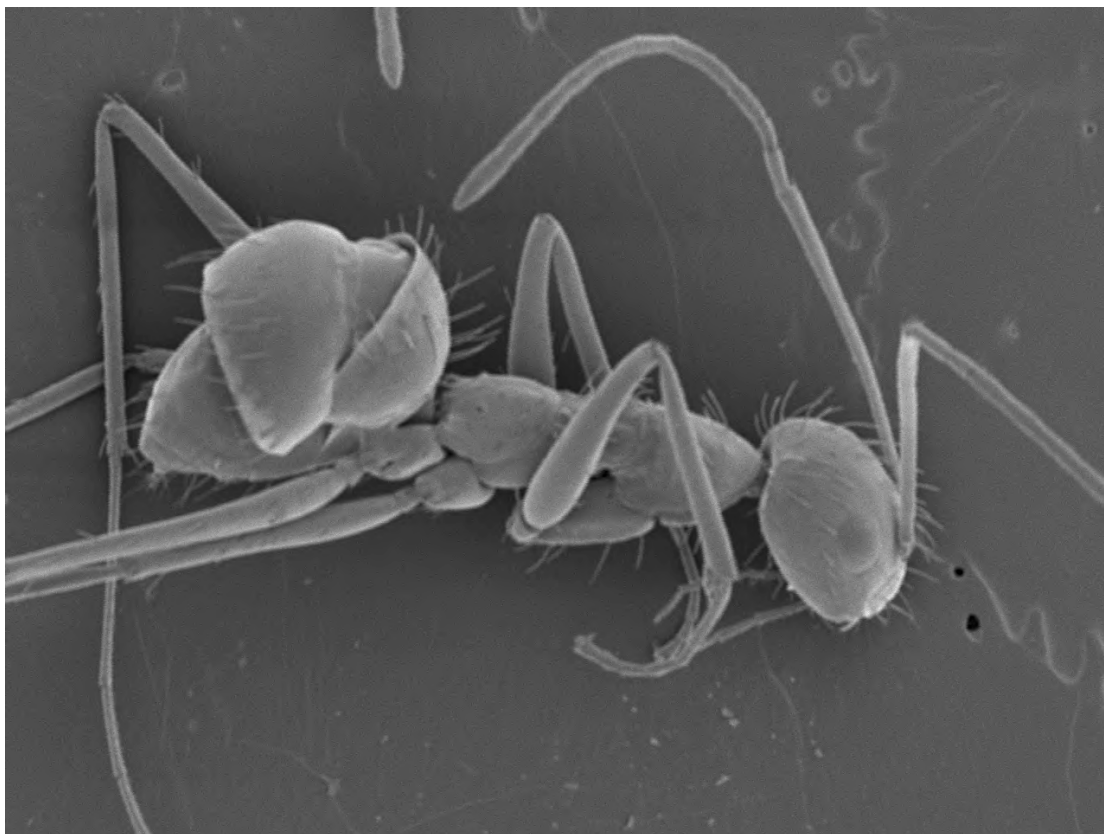
ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล  
กลุ่ม A ภาพที่ 2



กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบเทียบ				
4. รายละเอียดของภาพ				

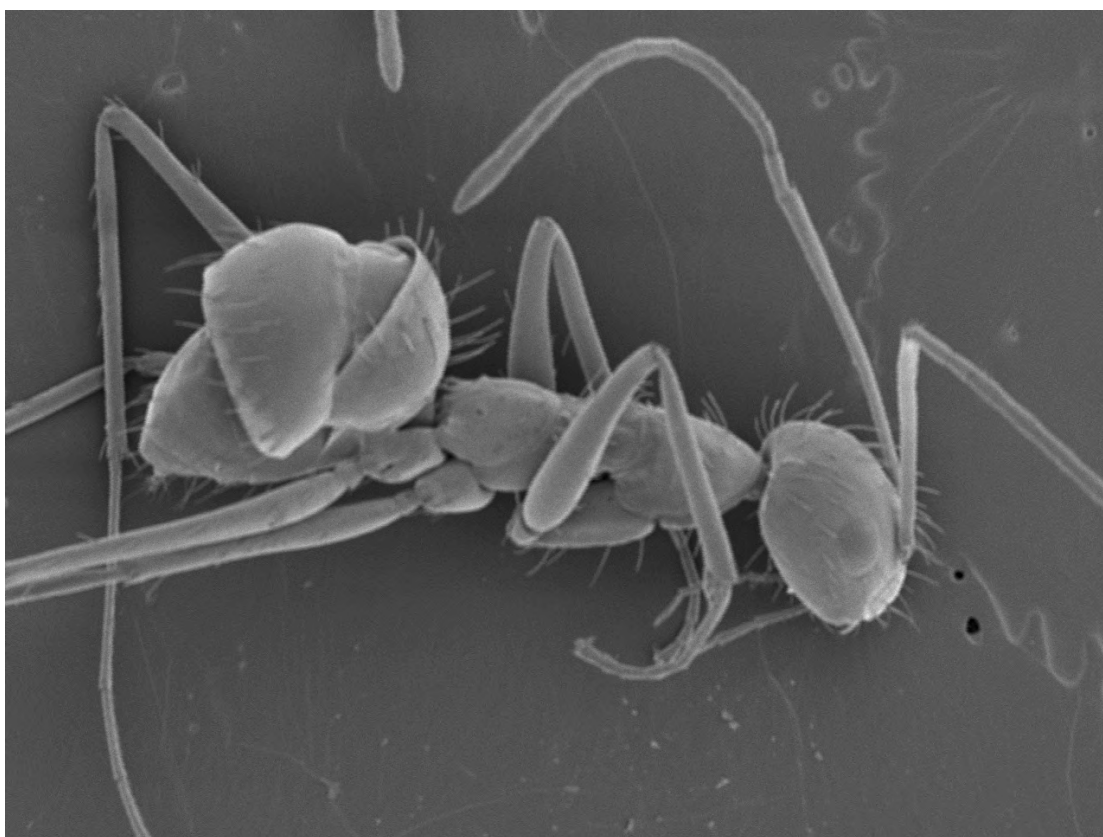
ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล  
กลุ่ม A ภาพที่ 3



กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบเทียบ				
4. รายละเอียดของภาพ				

ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล  
กลุ่ม A ภาพที่ 4



กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบต่าง				
4. รายละเอียดของภาพ				

ให้ท่านพิจารณาว่า ภาพใดสามารถนำไปใช้ทดแทนภาพต้นแบบได้ กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ลำดับภาพ	ใช้ทดแทนภาพต้นแบบได้อย่างดี	พอจะใช้ทดแทนภาพต้นแบบได้	ไม่อาจยอมให้ใช้ทดแทนภาพต้นแบบได้
1. ภาพ: กลุ่ม A ภาพที่ 1			
2. ภาพ: กลุ่ม A ภาพที่ 2			
3. ภาพ: กลุ่ม A ภาพที่ 3			
4. ภาพ: กลุ่ม A ภาพที่ 4			

ข้อเสนอแนะ.....

.....

.....

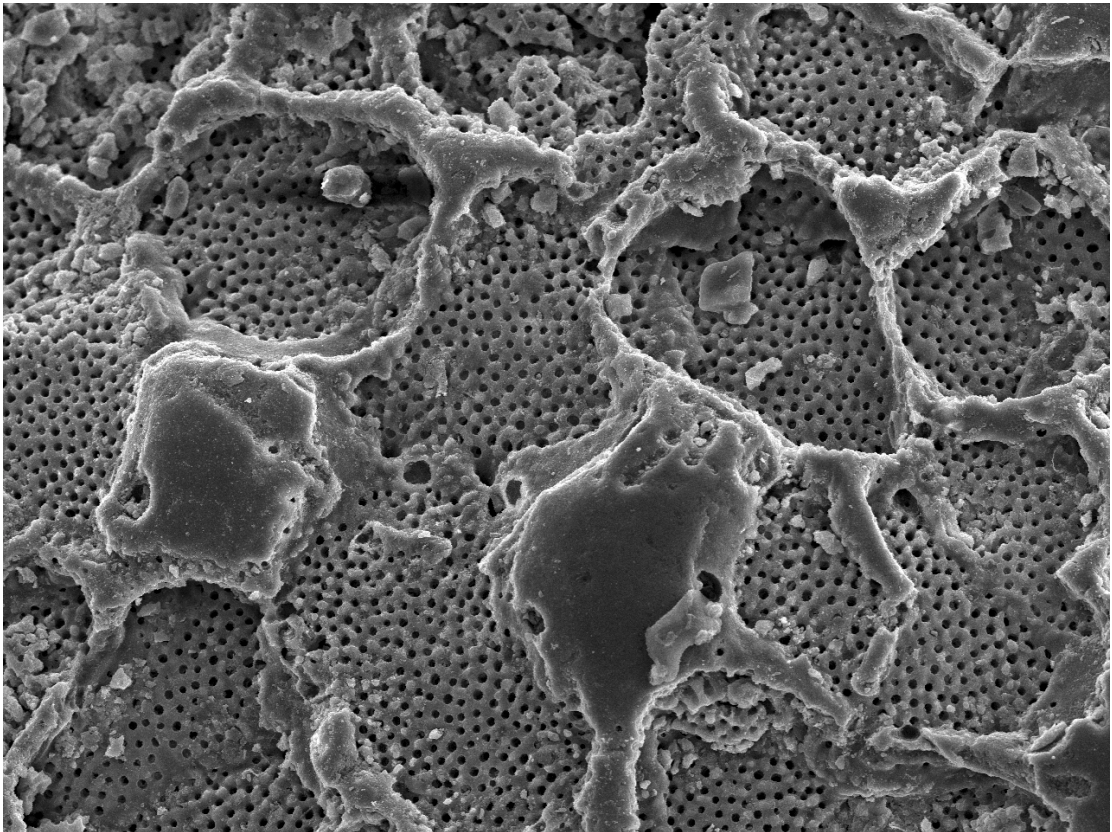
ตัวอย่างภาพที่ใช้วิจัย เป็นภาพที่บันทึกด้วยตัวกล้องจุลทรรศน์อิเล็กตรอนรุ่น JSM-5800 LV ของบริษัท JEOL LTD ณ มหาวิทยาลัยเทคโนโลยีสุรนารี และพิมพ์ด้วยเครื่องพิมพ์รุ่น HP LaserJet 5P ของบริษัท Hewlett Packard

ขอขอบพระคุณที่กรุณาใช้เวลากรอกแบบสอบถามเป็นอย่างสูง

ตอนที่3: แบบสอบถามการเปรียบเทียบภาพที่ผ่านการบีบอัดข้อมูลกับภาพต้นแบบ

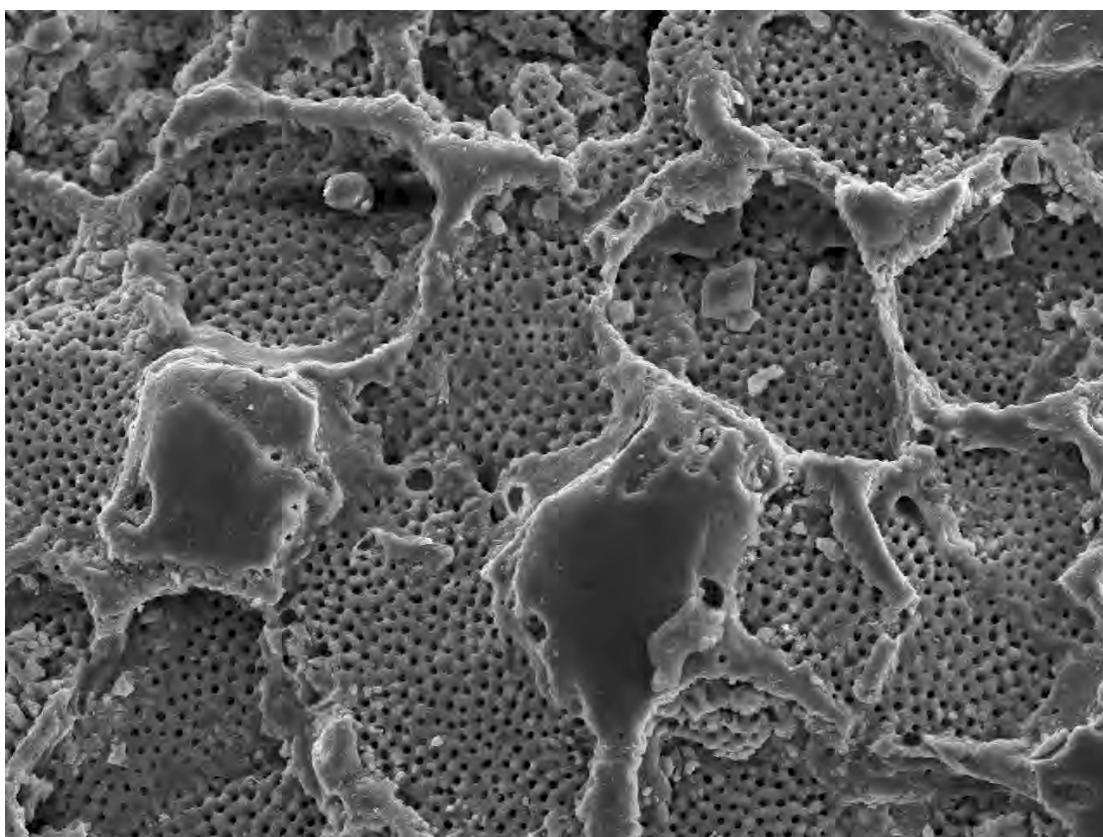
ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ไม่ผ่านการบีบอัดข้อมูล (ภาพต้นแบบ)

กลุ่ม B



ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล

กลุ่ม B ภาพที่ 1

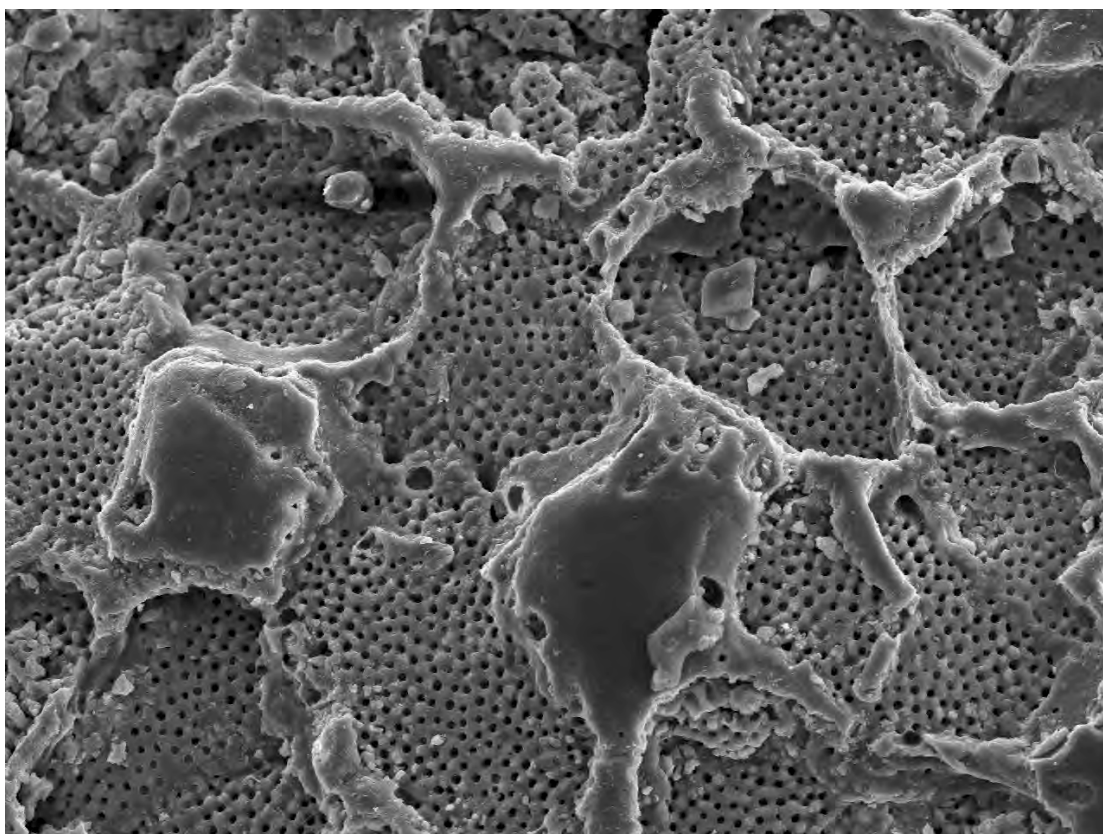


กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบต่าง				
4. รายละเอียดของภาพ				

ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล

กลุ่ม B ภาพที่ 2

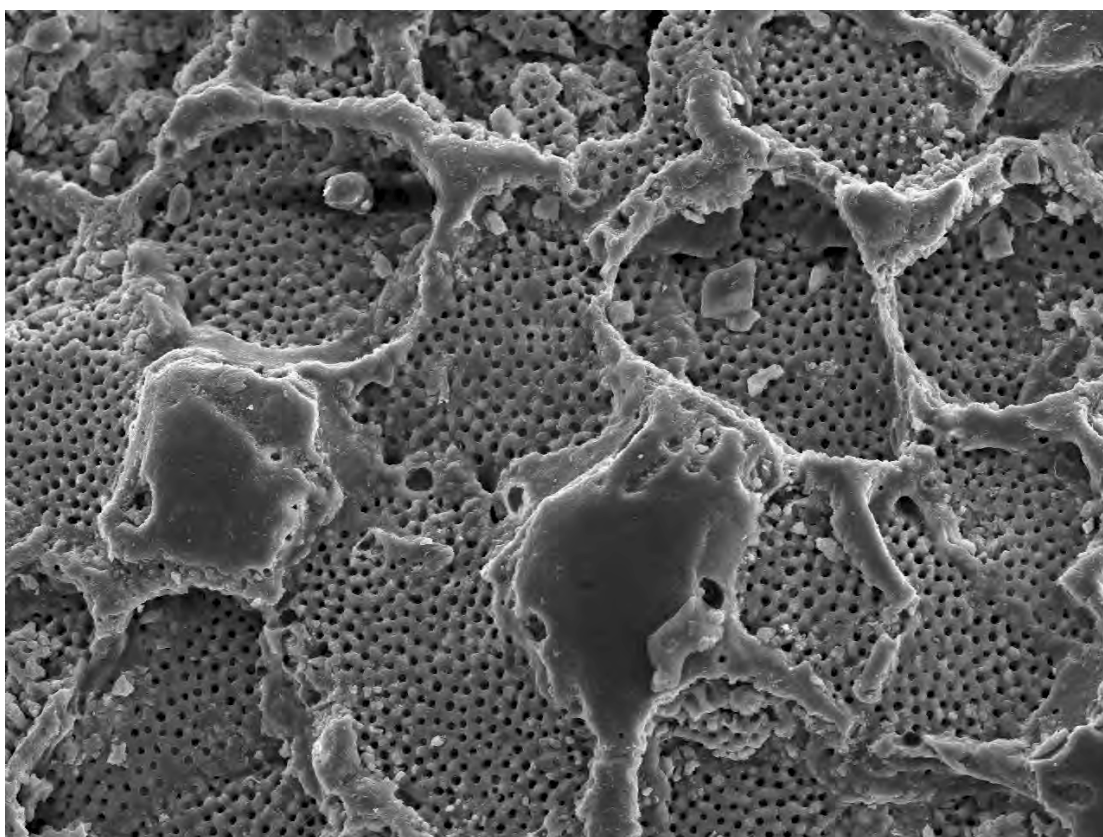


กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบต่าง				
4. รายละเอียดของภาพ				

ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล

กลุ่ม B ภาพที่ 3



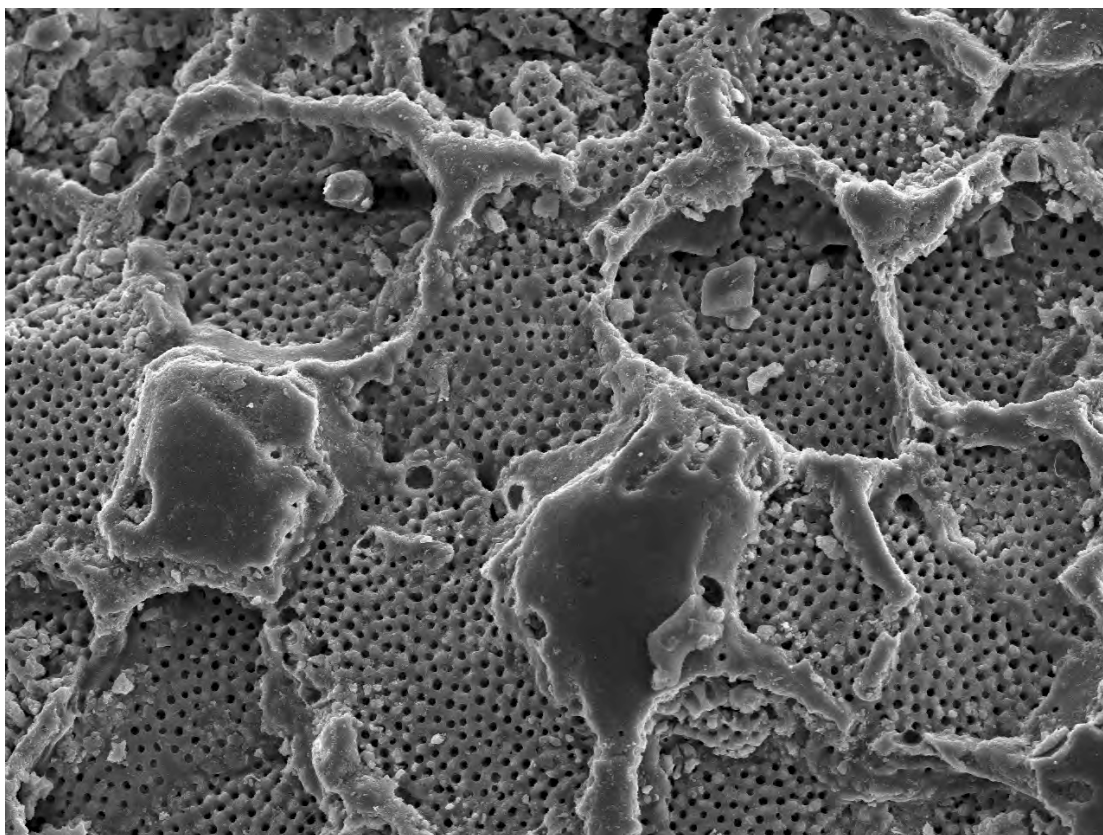
กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบต่าง				
4. รายละเอียดของภาพ				



ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล

กลุ่ม B ภาพที่ 4



กรุณาภาเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบต่าง				
4. รายละเอียดของภาพ				

ให้ท่านพิจารณาว่า ภาพใดสามารถนำไปใช้ทดแทนภาพต้นแบบได้ กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ลำดับภาพ	ใช้ทดแทนภาพต้นแบบได้อย่างดี	พอจะใช้ทดแทนภาพต้นแบบได้	ไม่อาจยอมให้ใช้ทดแทนภาพต้นแบบได้
1. ภาพ: กลุ่ม B ภาพที่ 1			
2. ภาพ: กลุ่ม B ภาพที่ 2			
3. ภาพ: กลุ่ม B ภาพที่ 3			
4. ภาพ: กลุ่ม B ภาพที่ 4			

ข้อเสนอแนะ.....

.....

.....

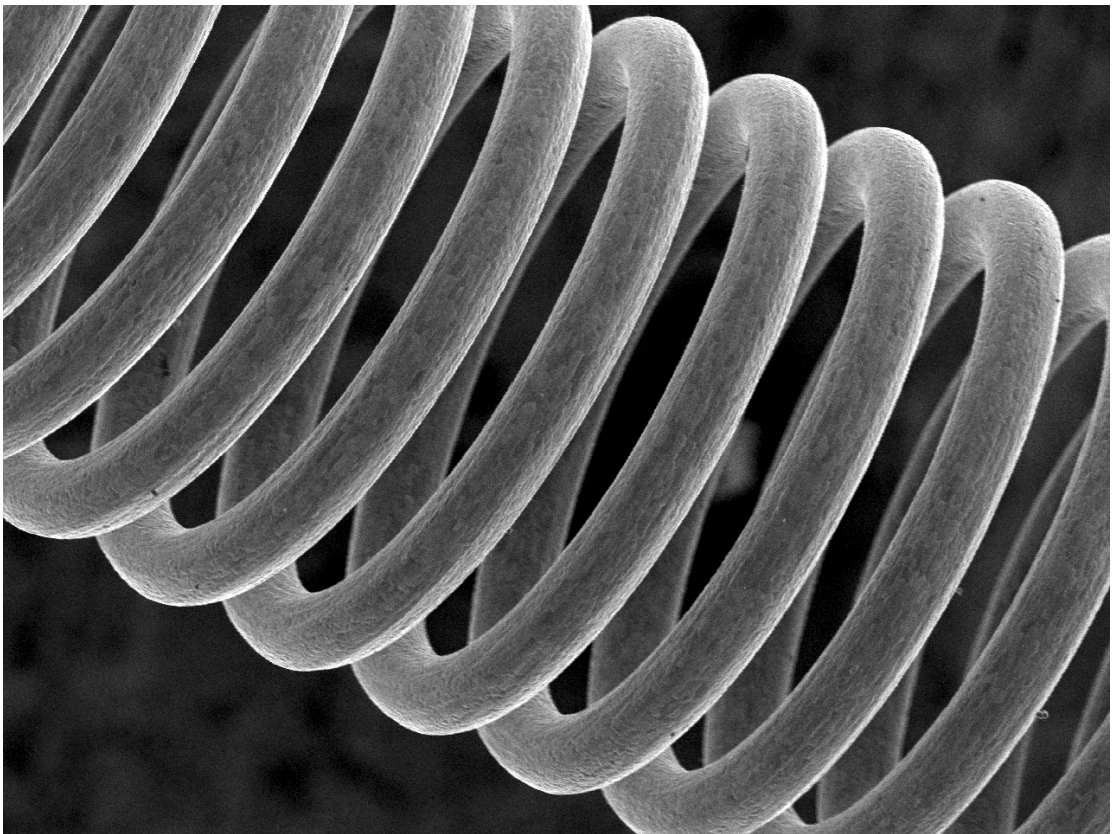
ตัวอย่างภาพที่ใช้วิจัย เป็นภาพที่บันทึกด้วยตัวกล้องจุลทรรศน์อิเล็กตรอนรุ่น JSM-5800 LV ของบริษัท JEOL LTD ณ มหาวิทยาลัยเทคโนโลยีสุรนารี และพิมพ์ด้วยเครื่องพิมพ์รุ่น HP LaserJet 5P ของบริษัท Hewlett Packard

ขอขอบพระคุณที่กรุณาใช้เวลากรอกแบบสอบถามเป็นอย่างสูง

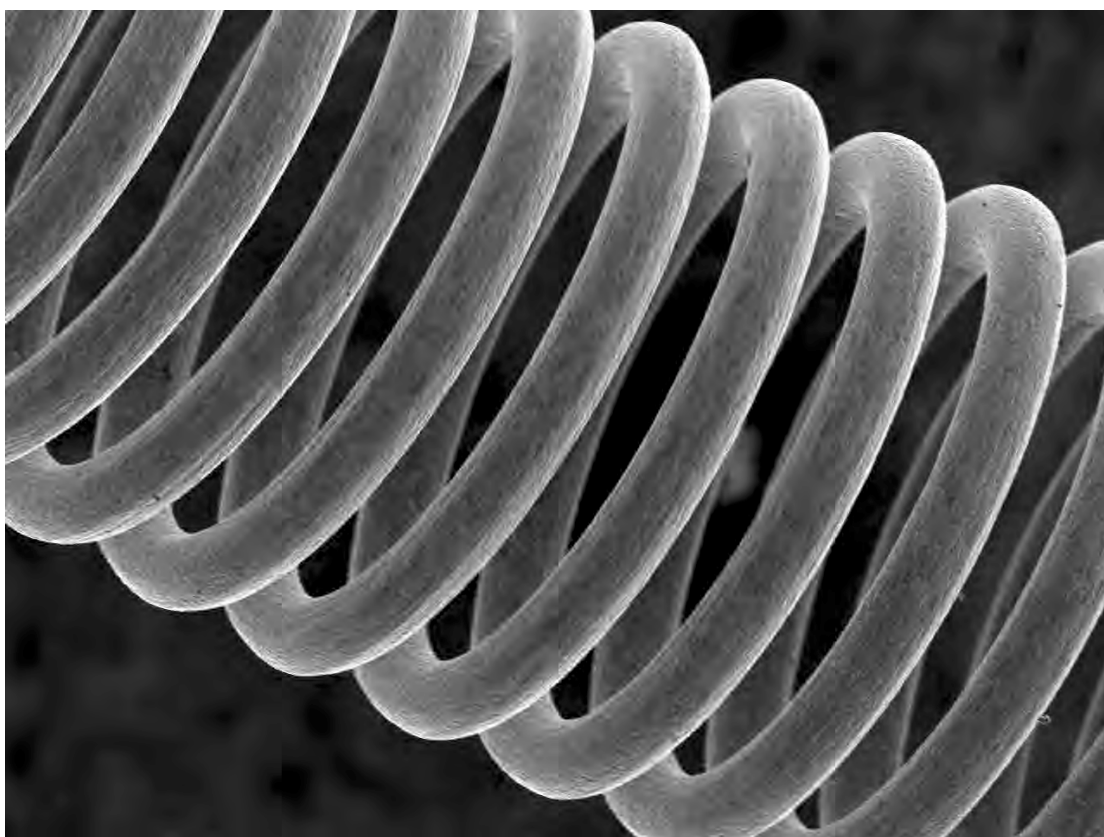
ตอนที่3: แบบสอบถามการเปรียบเทียบภาพที่ผ่านการบีบอัดข้อมูลกับภาพต้นแบบ

ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ไม่ผ่านการบีบอัดข้อมูล (ภาพต้นแบบ)

กลุ่ม C



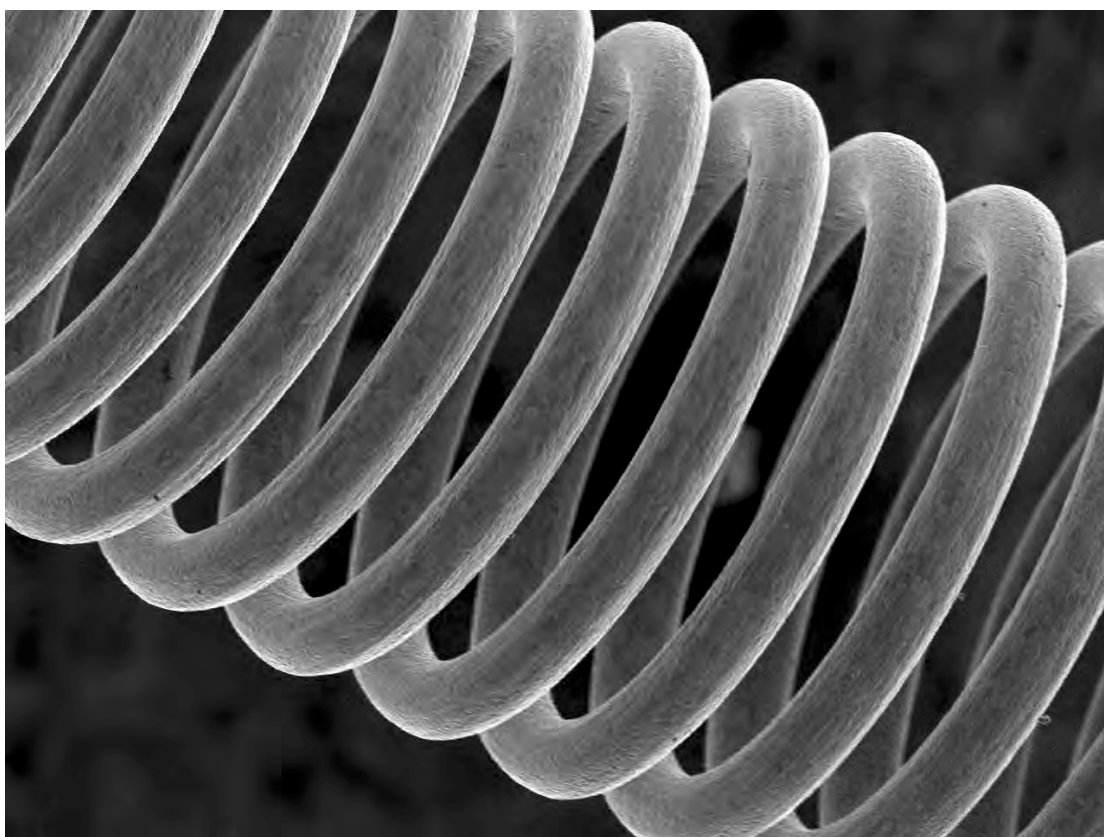
ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล  
กลุ่ม C ภาพที่ 1



กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบต่าง				
4. รายละเอียดของภาพ				

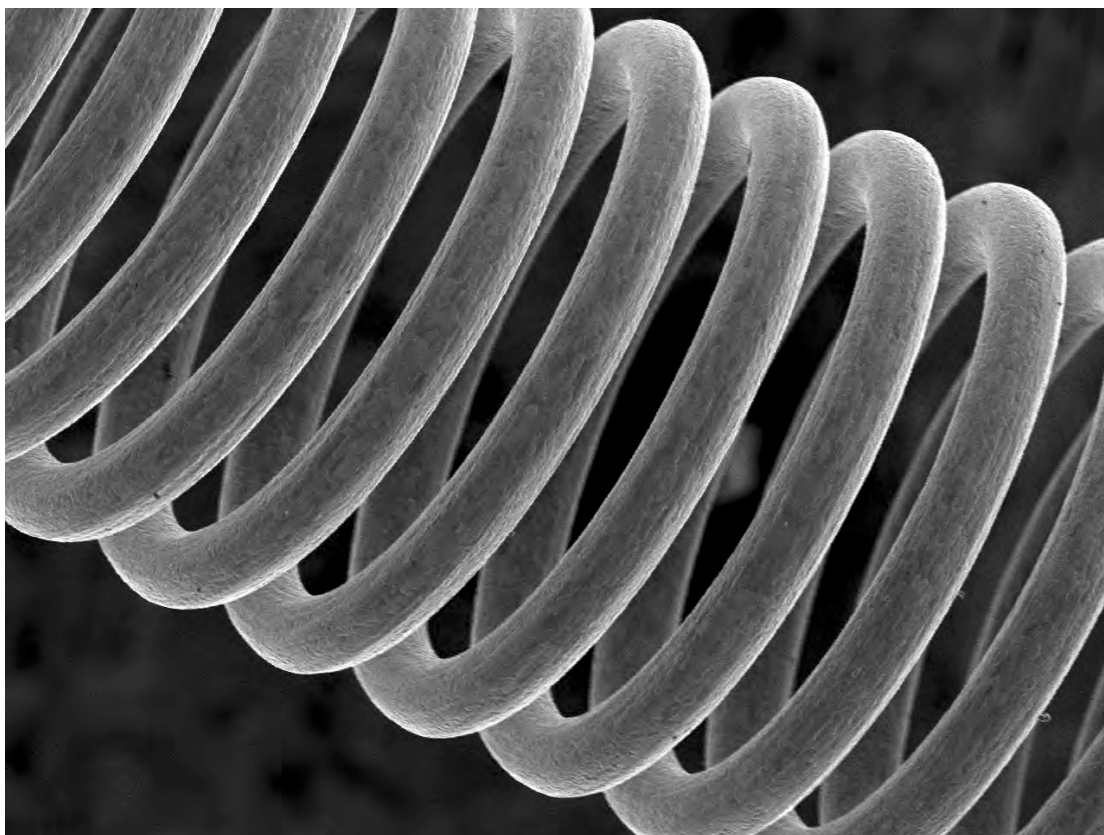
ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล  
กลุ่ม C ภาพที่ 2



กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบต่าง				
4. รายละเอียดของภาพ				

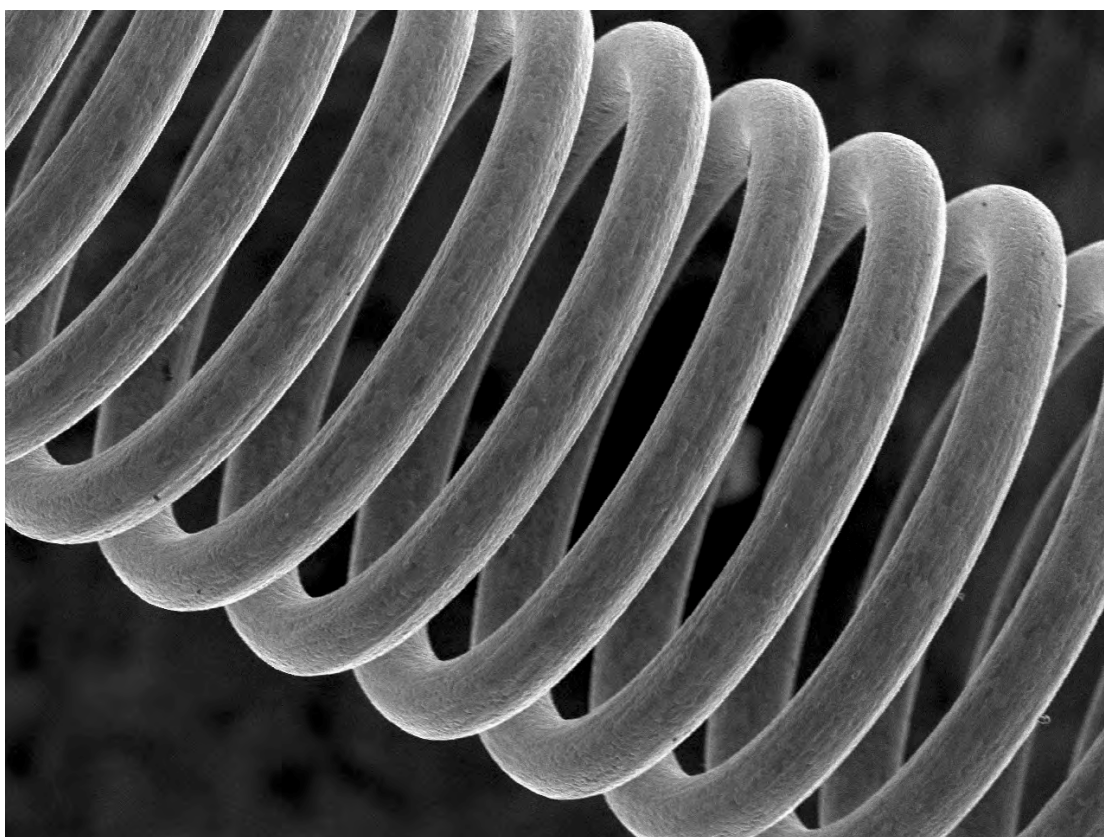
ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล  
กลุ่ม C ภาพที่ 3



กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบต่าง				
4. รายละเอียดของภาพ				

ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล  
กลุ่ม C ภาพที่ 4



กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบเทียบ				
4. รายละเอียดของภาพ				

ให้ท่านพิจารณาว่า ภาพใดสามารถนำไปใช้ทดแทนภาพต้นแบบได้ กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ลำดับภาพ	ใช้ทดแทนภาพต้นแบบได้อย่างดี	พอจะใช้ทดแทนภาพต้นแบบได้	ไม่อาจยอมให้ใช้ทดแทนภาพต้นแบบได้
1. ภาพ: กลุ่ม C ภาพที่ 1			
2. ภาพ: กลุ่ม C ภาพที่ 2			
3. ภาพ: กลุ่ม C ภาพที่ 3			
4. ภาพ: กลุ่ม C ภาพที่ 4			

ข้อเสนอแนะ.....

.....

.....

ตัวอย่างภาพที่ใช้วิจัย เป็นภาพที่บันทึกด้วยตัวกล้องจุลทรรศน์อิเล็กตรอนรุ่น JSM-5800 LV ของบริษัท JEOL LTD ณ มหาวิทยาลัยเทคโนโลยีสุรนารี และพิมพ์ด้วยเครื่องพิมพ์รุ่น HP LaserJet 5P ของบริษัท Hewlett Packard

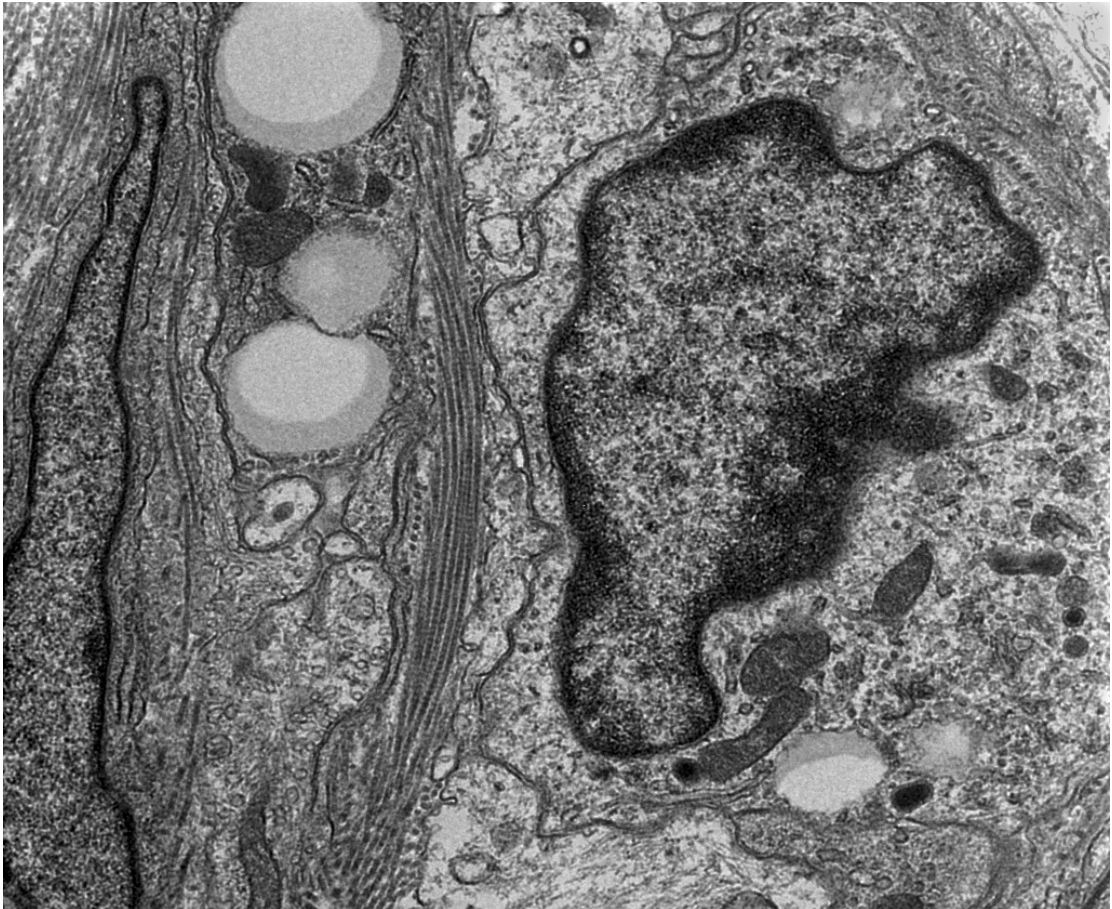
ขอขอบพระคุณที่กรุณาใช้เวลากรอกแบบสอบถามเป็นอย่างสูง



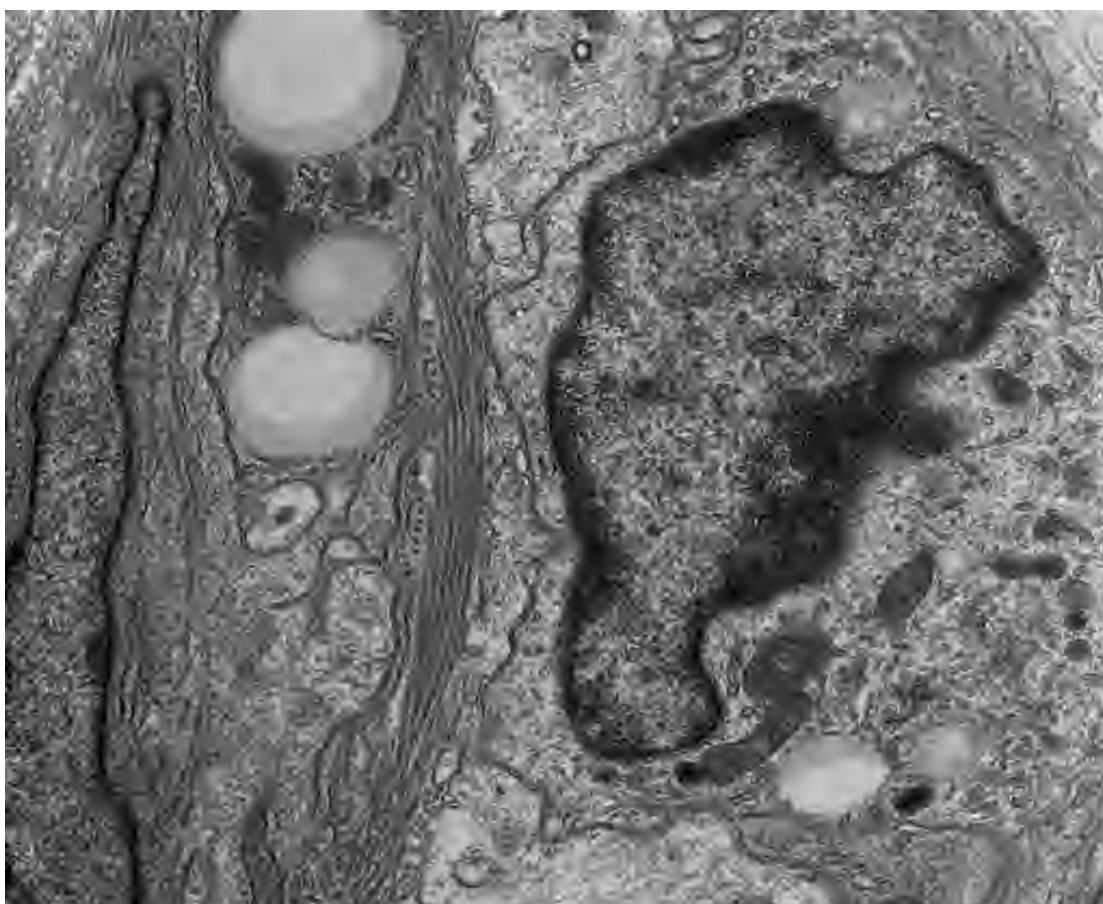
ตอนที่3: แบบสอบถามการเปรียบเทียบภาพที่ผ่านการบีบอัดข้อมูลกับภาพต้นแบบ

ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ไม่ผ่านการบีบอัดข้อมูล (ภาพต้นแบบ)

กลุ่ม D



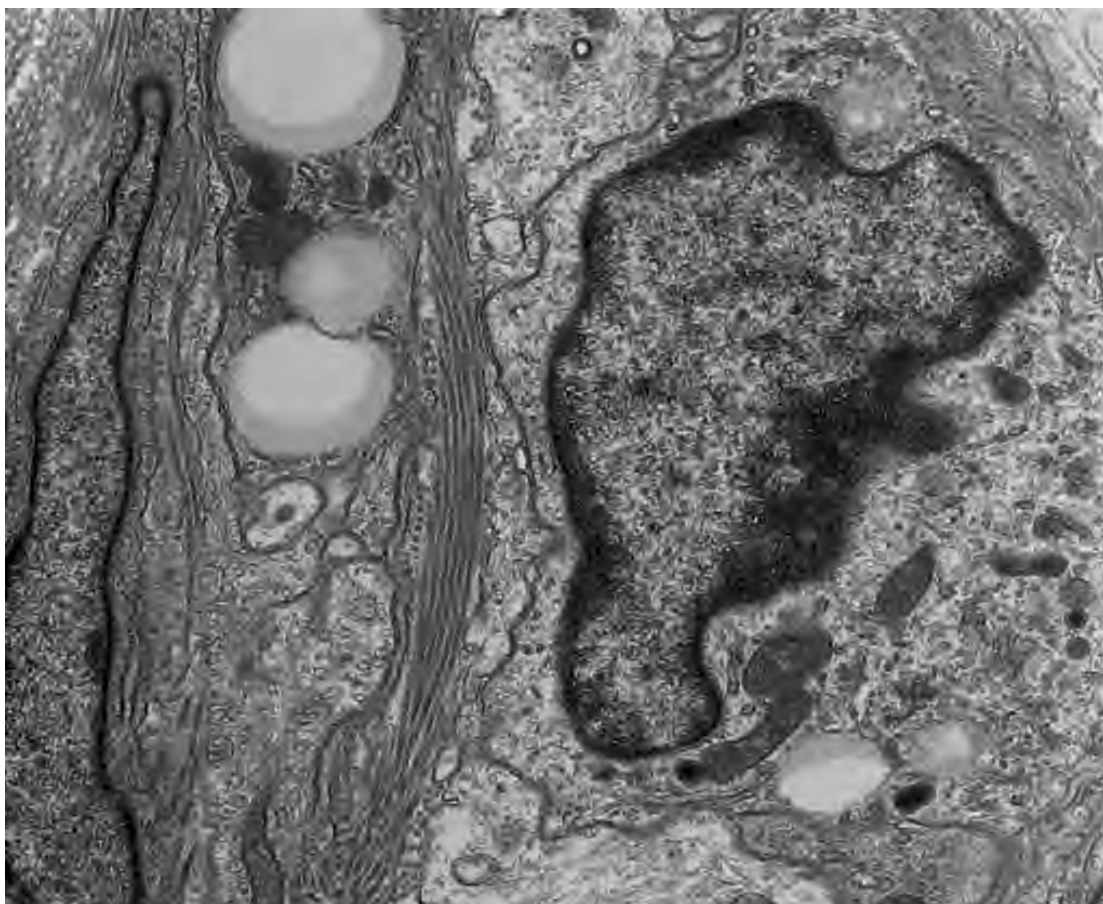
ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล  
กลุ่ม D ภาพที่ 1



กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบต่าง				
4. รายละเอียดของภาพ				

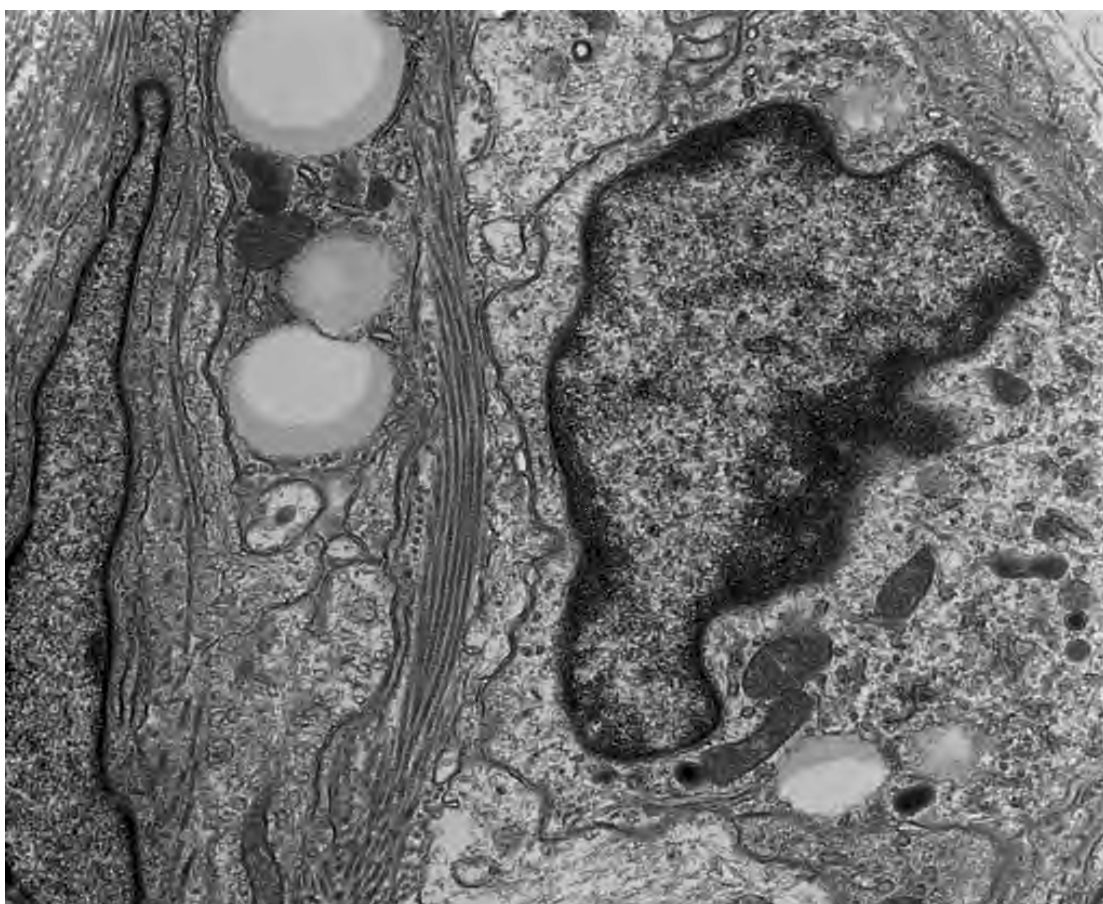
ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล  
กลุ่ม D ภาพที่ 2



กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบต่าง				
4. รายละเอียดของภาพ				

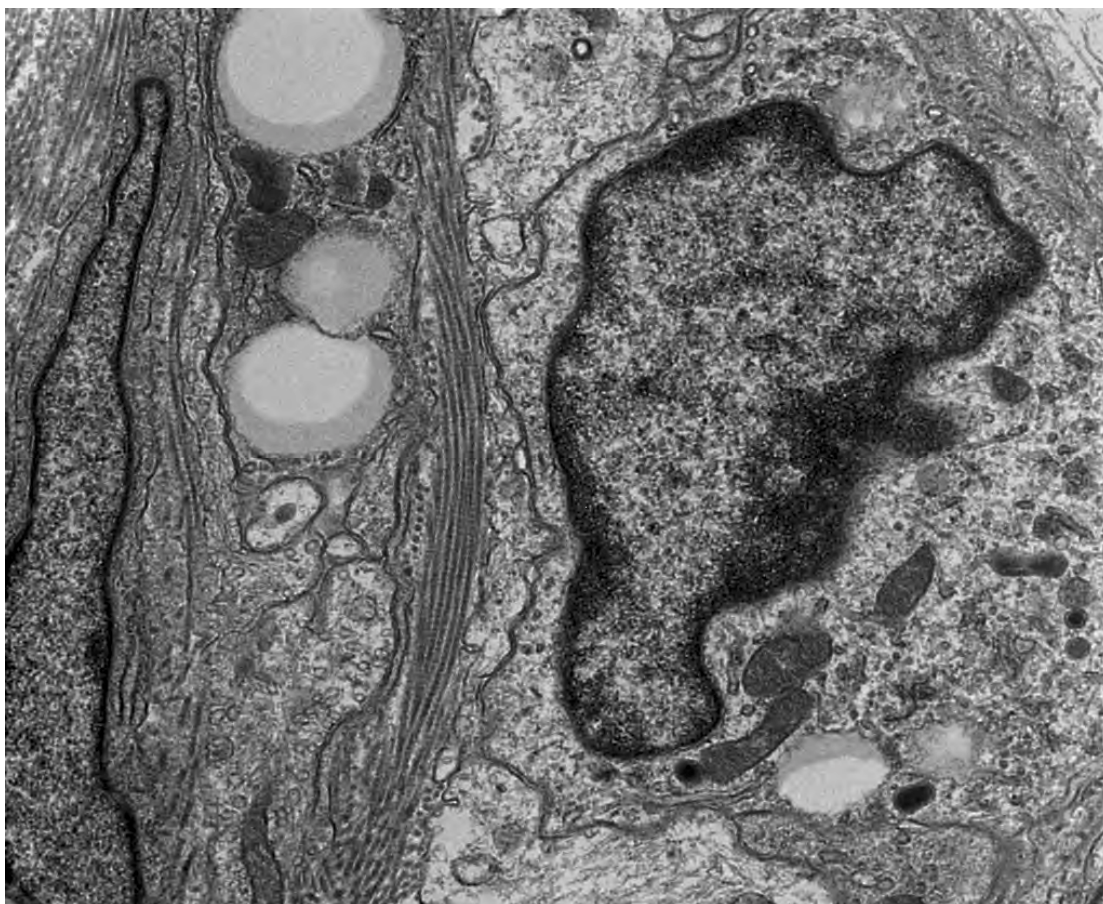
ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล  
กลุ่ม D ภาพที่ 3



กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบต่าง				
4. รายละเอียดของภาพ				

ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล  
 กลุ่ม D ภาพที่ 4



กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบต่าง				
4. รายละเอียดของภาพ				

ให้ท่านพิจารณาว่า ภาพใดสามารถนำไปใช้ทดแทนภาพต้นแบบได้ กรุณาทำเครื่องหมาย  
 ✓ ลงในช่องที่ท่านเห็นสมควร

ลำดับภาพ	ใช้ทดแทนภาพ ต้นแบบได้อย่างดี	พอจะใช้ทดแทน ภาพต้นแบบได้	ไม่อาจยอมให้ใช้ ทดแทนภาพ ต้นแบบได้
1. ภาพ: กลุ่ม D ภาพที่ 1			
2. ภาพ: กลุ่ม D ภาพที่ 2			
3. ภาพ: กลุ่ม D ภาพที่ 3			
4. ภาพ: กลุ่ม D ภาพที่ 4			

ข้อเสนอแนะ.....  
 .....  
 .....

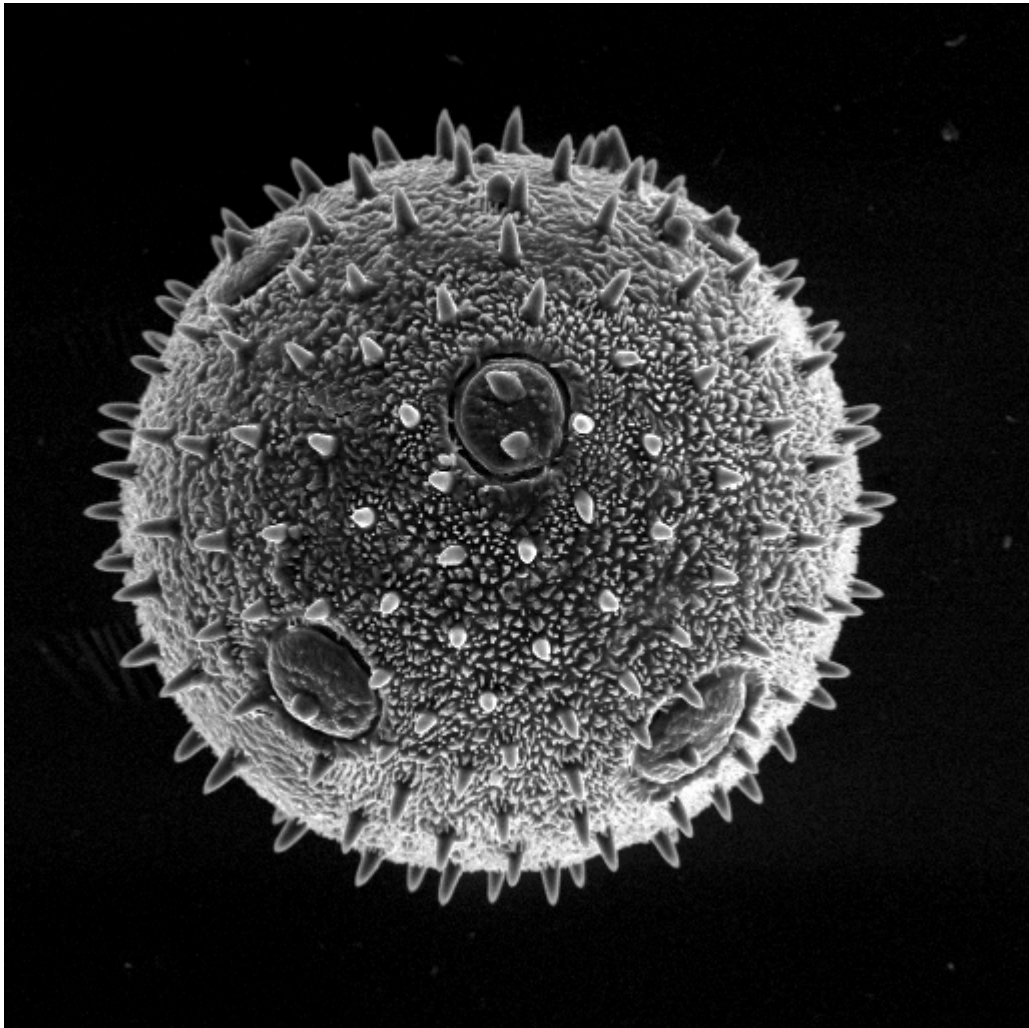
ตัวอย่างภาพที่ใช้วิจัย เป็นภาพตัวอย่างจากกล้องจุลทรรศน์อิเล็กตรอนรุ่น JEM 2010 ของบริษัท  
 JEOL LTD และพิมพ์ด้วยเครื่องพิมพ์รุ่น HP LaserJet 5P ของบริษัท Hewlett Packard

ขอขอบพระคุณที่กรุณาใช้เวลากรอกแบบสอบถามเป็นอย่างสูง

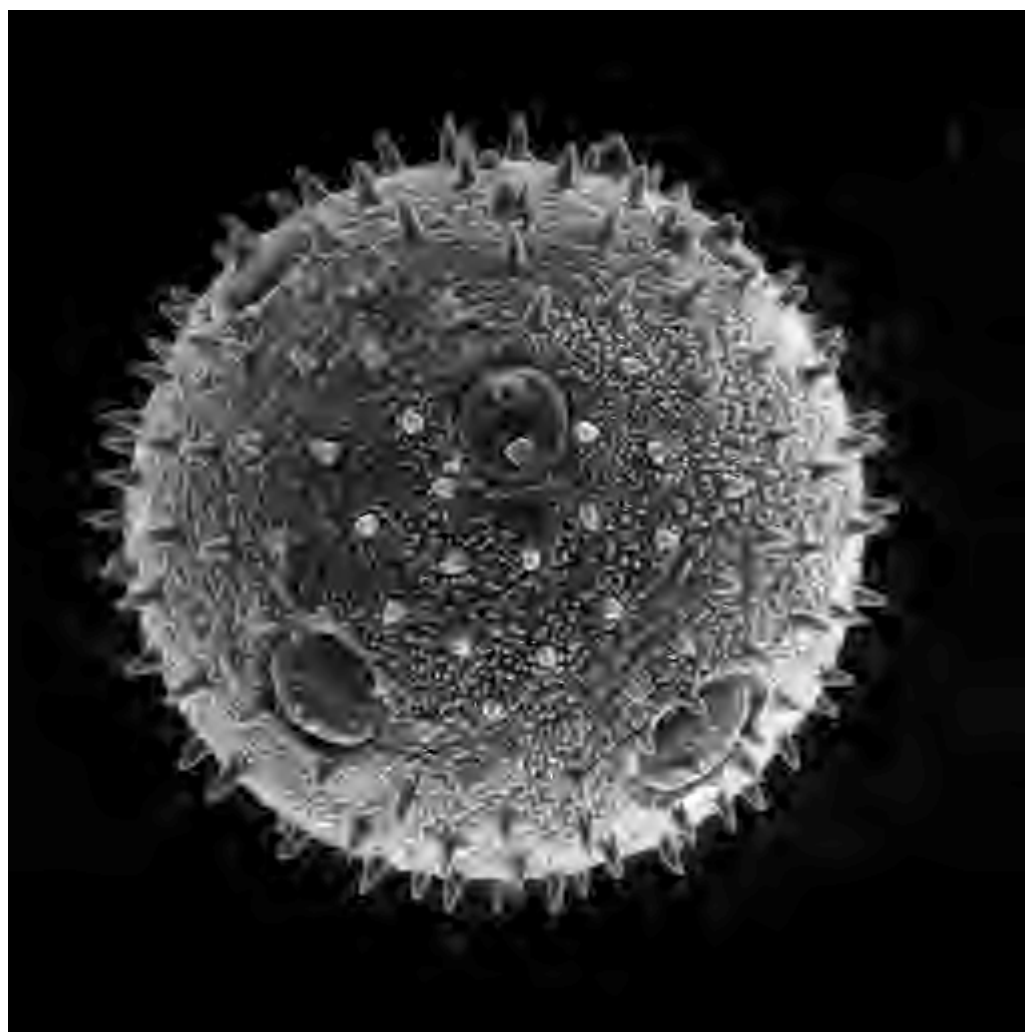
ตอนที่3: แบบสอบถามการเปรียบเทียบภาพที่ผ่านการบีบอัดข้อมูลกับภาพต้นแบบ

ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ไม่ผ่านการบีบอัดข้อมูล (ภาพต้นแบบ)

กลุ่ม E



ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล  
กลุ่ม E ภาพที่ 1

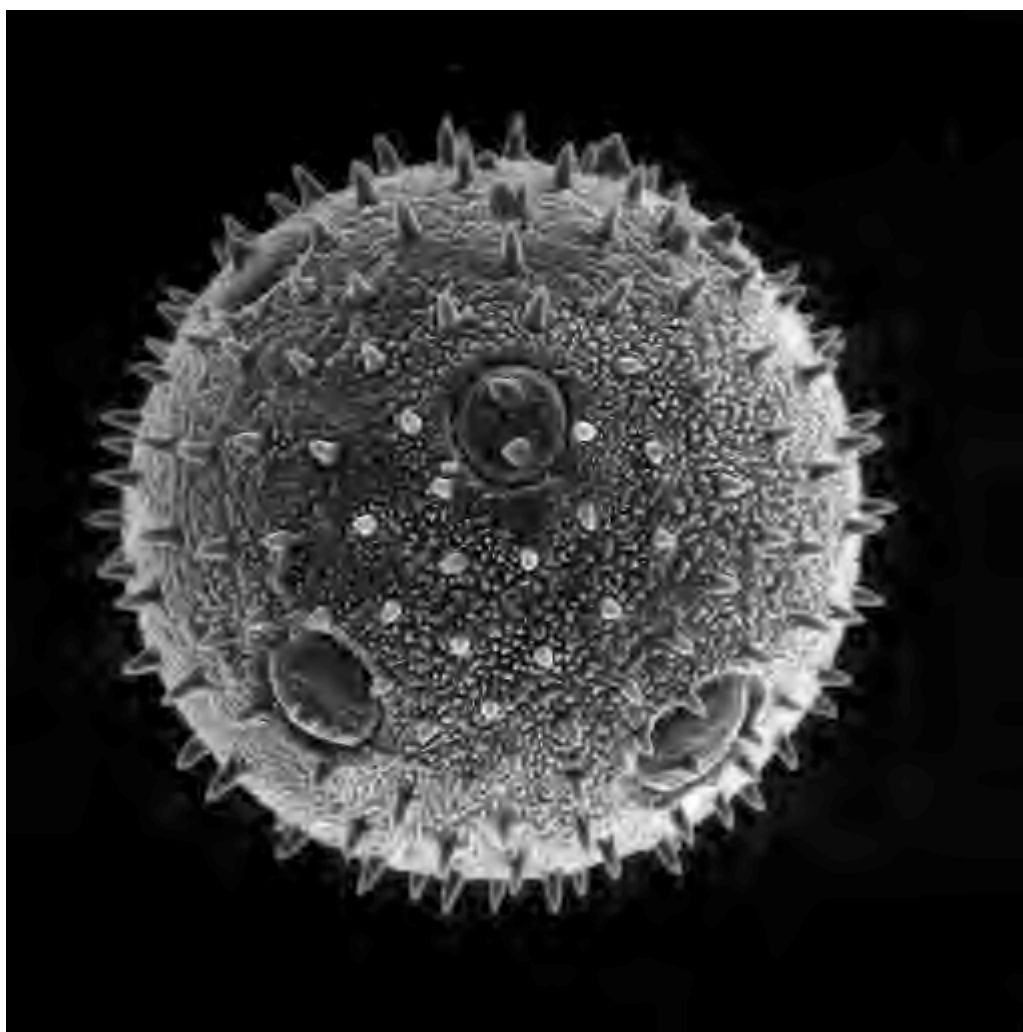


กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบเทียบ				
4. รายละเอียดของภาพ				



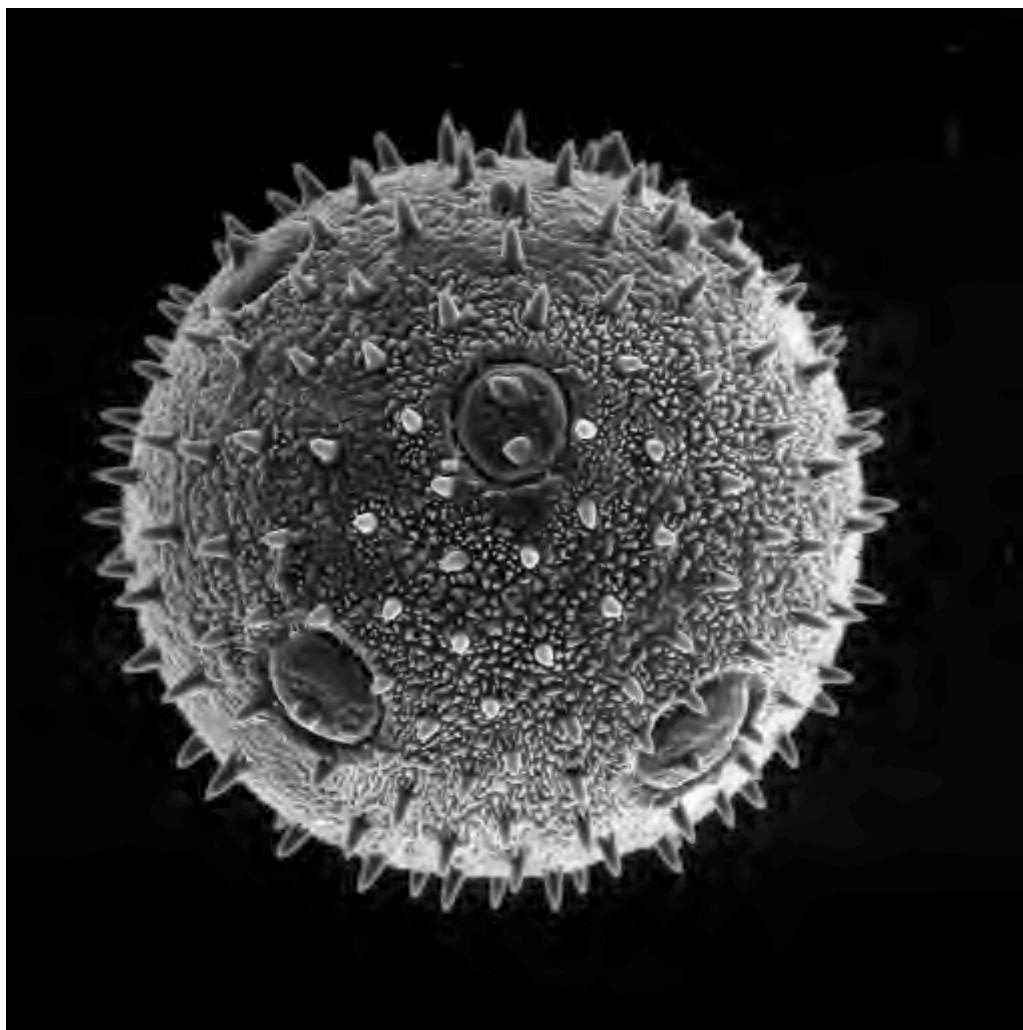
ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล  
กลุ่ม E ภาพที่ 2



กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบเทียบ				
4. รายละเอียดของภาพ				

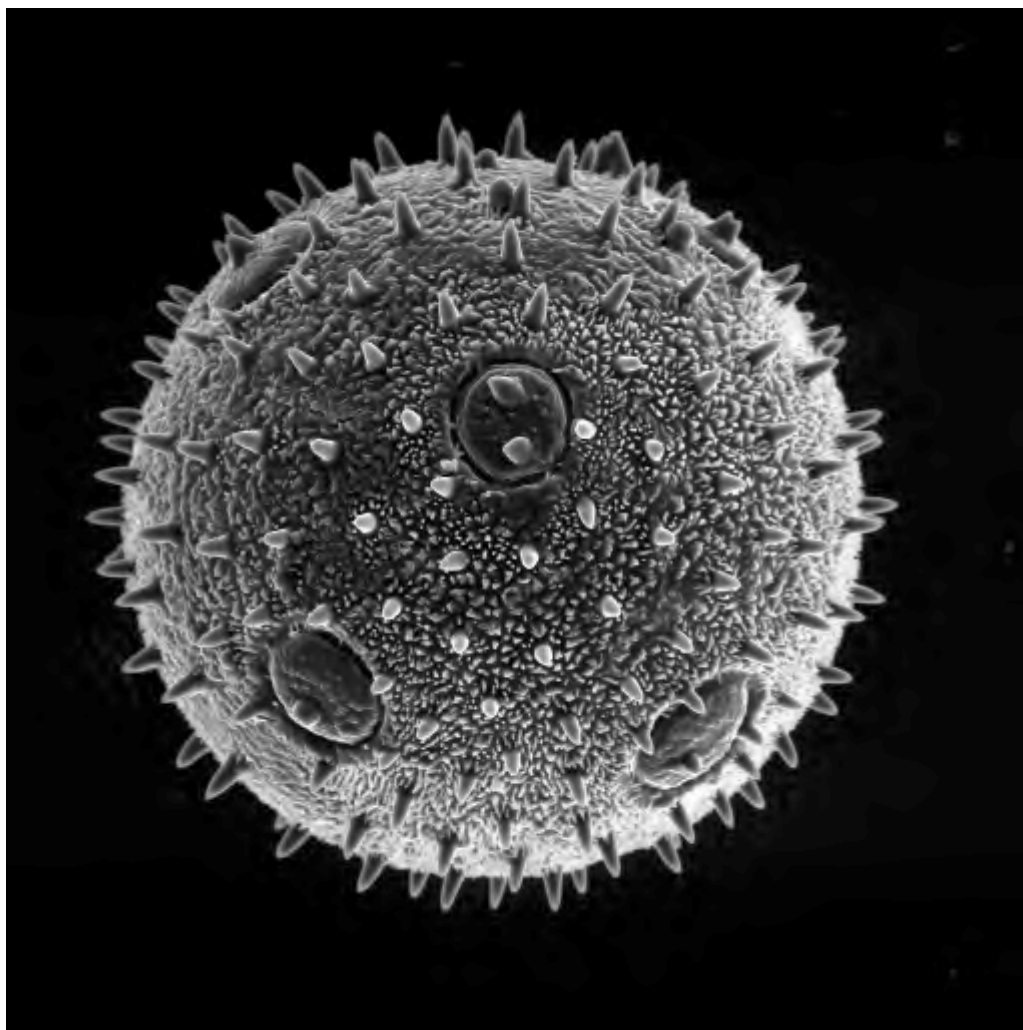
ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล  
กลุ่ม E ภาพที่ 3



กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบเทียบ				
4. รายละเอียดของภาพ				

ภาพจากกล้องจุลทรรศน์อิเล็กตรอนที่ผ่านการบีบอัดข้อมูล  
กลุ่ม E ภาพที่ 4



กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ข้อพิจารณา	ระดับคุณภาพของภาพ			
	4	3	2	1
1. ความสว่าง				
2. ความคมชัด				
3. ความเปรียบต่าง				
4. รายละเอียดของภาพ				

ให้ท่านพิจารณาว่า ภาพใดสามารถนำไปใช้ทดแทนภาพต้นแบบได้ กรุณาทำเครื่องหมาย ✓ ลงในช่องที่ท่านเห็นสมควร

ลำดับภาพ	ใช้ทดแทนภาพต้นแบบได้อย่างดี	พอจะใช้ทดแทนภาพต้นแบบได้	ไม่อาจยอมให้ใช้ทดแทนภาพต้นแบบได้
1. ภาพ: กลุ่ม E ภาพที่ 1			
2. ภาพ: กลุ่ม E ภาพที่ 2			
3. ภาพ: กลุ่ม E ภาพที่ 3			
4. ภาพ: กลุ่ม E ภาพที่ 4			

ข้อเสนอแนะ.....  
 .....  
 .....

ตัวอย่างภาพที่ใช้วิจัย เป็นภาพตัวอย่างจากกล้องจุลทรรศน์อิเล็กตรอนรุ่น JEM 2010 ของบริษัท JEOL LTD และพิมพ์ด้วยเครื่องพิมพ์รุ่น HP LaserJet 5P ของบริษัท Hewlett Packard

ขอขอบพระคุณที่กรุณาใช้เวลากรอกแบบสอบถามเป็นอย่างสูง

## ภาคผนวก ข

ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลไฟล์ภาพ

ลำดับรายชื่อของผู้ทำการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลไฟล์ภาพ มีรายชื่อผู้ทำการประเมินดังนี้

- ท่านที่ 1 รศ. วิรุพห์ มังคละวิรัช
- ท่านที่ 2 ผศ. ดร. ยุวดี มานะเกษม
- ท่านที่ 3 คุณ กรวรรณ รัตนไชย
- ท่านที่ 4 คุณ ประสิทธิ์ งามสันเทียะ
- ท่านที่ 5 อ. ดร. วีระยุทธ ลอประยูร
- ท่านที่ 6 คุณ อนุชิต เรืองวิทยานนท์
- ท่านที่ 7 รศ. ดร. ประพีร์ เศรษฐรักษ์
- ท่านที่ 8 รศ. ดร. ชัยทิพย์ วนิชานนท์
- ท่านที่ 9 อ. ดร. วิศิษฐ ทวีปรั้งยีพร
- ท่านที่ 10 รศ. เกลียวพันธ์ เถลิงพล
- ท่านที่ 11 ฝ่ายเครื่องมือวิทยาศาสตร์ฯ สถาบันวิจัยและพัฒนาฯ มหาวิทยาลัยเกษตร-  
ศาสตร์
- ท่านที่ 12 ฝ่ายปฏิบัติการวิจัยฯ สถาบันวิจัยและพัฒนาฯ มหาวิทยาลัยเกษตรศาสตร์  
กำแพงแสน

#### หมายเหตุ

- 1) ผลการประเมินของแต่ละท่านที่ทำการประเมินเรียงตามลำดับรายชื่อ
- 2) ความหมายของระดับการใช้ทดแทนคือ
  - 1- หมายถึง ใช้ทดแทนภาพต้นแบบได้อย่างดี
  - 2- หมายถึง พอดีจะใช้ทดแทนภาพต้นแบบได้
  - 3- หมายถึง ไม่อาจยอมให้ใช้ทดแทนภาพต้นแบบได้
- 3) ภาพแต่ละกลุ่มภาพเป็นไปตามภาคผนวก ก.
- 4) ภาพรายละเอียดต่ำคือ ภาพ ant และภาพ LED
- 5) ภาพรายละเอียดสูงคือ ภาพ sand, ภาพ skin และภาพ pollen of zucchini

ตารางที่ ข.1 ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละบุคคล

ข้อพิจารณา	ผลการประเมินท่านที่ 1				ผลการประเมินท่านที่ 2				ผลการประเมินท่านที่ 3				
	ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				
	4	3	2	1	4	3	2	1	4	3	2	1	
ความสว่าง	✓				✓							✓	กลุ่ม A ภาพ 1
ความคมชัด		✓				✓						✓	
ความเปรียบต่าง		✓			✓							✓	
รายละเอียดของภาพ		✓			✓							✓	
ความสว่าง	✓					✓				✓			กลุ่ม A ภาพ 2
ความคมชัด		✓					✓		✓	✓			
ความเปรียบต่าง		✓					✓		✓	✓			
รายละเอียดของภาพ		✓					✓		✓	✓			
ความสว่าง	✓						✓				✓		กลุ่ม A ภาพ 3
ความคมชัด		✓					✓			✓	✓		
ความเปรียบต่าง	✓						✓			✓	✓		
รายละเอียดของภาพ	✓						✓			✓	✓		
ความสว่าง	✓						✓		✓				กลุ่ม A ภาพ 4
ความคมชัด		✓					✓		✓	✓			
ความเปรียบต่าง		✓					✓		✓	✓			
รายละเอียดของภาพ		✓					✓		✓	✓			
ความสว่าง	✓					✓						✓	กลุ่ม B ภาพ 1
ความคมชัด			✓				✓					✓	
ความเปรียบต่าง		✓					✓					✓	
รายละเอียดของภาพ			✓				✓					✓	
ความสว่าง	✓					✓						✓	กลุ่ม B ภาพ 2
ความคมชัด			✓				✓					✓	
ความเปรียบต่าง		✓					✓					✓	
รายละเอียดของภาพ			✓				✓					✓	
ความสว่าง	✓					✓						✓	กลุ่ม B ภาพ 3
ความคมชัด		✓					✓					✓	
ความเปรียบต่าง		✓					✓					✓	
รายละเอียดของภาพ		✓					✓					✓	
ความสว่าง	✓						✓			✓			กลุ่ม B ภาพ 4
ความคมชัด		✓					✓			✓			
ความเปรียบต่าง		✓					✓			✓			
รายละเอียดของภาพ			✓				✓			✓			
ความสว่าง	✓						✓					✓	กลุ่ม C ภาพ 1
ความคมชัด		✓					✓					✓	
ความเปรียบต่าง		✓					✓					✓	
รายละเอียดของภาพ		✓					✓					✓	

ตารางที่ ข.1 ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละบุคคล (ต่อ)

ข้อพิจารณา	ผลการประเมินท่านที่ 1				ผลการประเมินท่านที่ 2				ผลการประเมินท่านที่ 3				
	ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				
	4	3	2	1	4	3	2	1	4	3	2	1	
ความสว่าง	✓				✓						✓		กลุ่ม C ภาพ 2
ความคมชัด		✓				✓				✓	✓		
ความเปรียบต่าง			✓			✓				✓	✓		
รายละเอียดของภาพ		✓					✓			✓	✓		
ความสว่าง	✓						✓			✓			กลุ่ม C ภาพ 3
ความคมชัด		✓				✓				✓	✓		
ความเปรียบต่าง		✓				✓				✓	✓		
รายละเอียดของภาพ		✓				✓				✓	✓		
ความสว่าง	✓						✓			✓			กลุ่ม C ภาพ 4
ความคมชัด		✓				✓				✓	✓		
ความเปรียบต่าง			✓			✓				✓	✓		
รายละเอียดของภาพ		✓				✓				✓	✓		
ความสว่าง	✓						✓					✓	กลุ่ม D ภาพ 1
ความคมชัด			✓				✓				✓	✓	
ความเปรียบต่าง			✓				✓				✓	✓	
รายละเอียดของภาพ			✓				✓				✓	✓	
ความสว่าง	✓						✓					✓	กลุ่ม D ภาพ 2
ความคมชัด			✓				✓				✓	✓	
ความเปรียบต่าง			✓				✓				✓	✓	
รายละเอียดของภาพ			✓				✓				✓	✓	
ความสว่าง	✓						✓				✓		กลุ่ม D ภาพ 3
ความคมชัด			✓				✓				✓	✓	
ความเปรียบต่าง			✓				✓				✓	✓	
รายละเอียดของภาพ		✓					✓				✓	✓	
ความสว่าง	✓						✓				✓		กลุ่ม D ภาพ 4
ความคมชัด		✓					✓				✓	✓	
ความเปรียบต่าง			✓				✓				✓	✓	
รายละเอียดของภาพ			✓				✓				✓	✓	
ความสว่าง		✓					✓					✓	กลุ่ม E ภาพ 1
ความคมชัด			✓				✓				✓	✓	
ความเปรียบต่าง			✓				✓				✓	✓	
รายละเอียดของภาพ			✓				✓				✓	✓	
ความสว่าง		✓					✓					✓	กลุ่ม E ภาพ 2
ความคมชัด			✓				✓				✓	✓	
ความเปรียบต่าง			✓				✓				✓	✓	
รายละเอียดของภาพ			✓				✓				✓	✓	



ตารางที่ ข.1 ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละบุคคล (ต่อ)

ข้อพิจารณา	ผลการประเมินท่านที่ 1				ผลการประเมินท่านที่ 2				ผลการประเมินท่านที่ 3				
	ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				
	4	3	2	1	4	3	2	1	4	3	2	1	
ความสว่าง	✓					✓						✓	กลุ่ม E ภาพ 3
ความคมชัด			✓				✓					✓	
ความเปรียบต่าง			✓				✓					✓	
รายละเอียดของภาพ		✓					✓					✓	
ความสว่าง	✓					✓						✓	กลุ่ม E ภาพ 4
ความคมชัด			✓			✓						✓	
ความเปรียบต่าง		✓				✓						✓	
รายละเอียดของภาพ		✓				✓						✓	

ตารางที่ ข.1 ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละบุคคล (ต่อ)

ลำดับภาพ	ผลการประเมินท่านที่ 1			ผลการประเมินท่านที่ 2			ผลการประเมินท่านที่ 3		
	ระดับการใช้ทดแทน			ระดับการใช้ทดแทน			ระดับการใช้ทดแทน		
	3	2	1	3	2	1	3	2	1
กลุ่ม A ภาพ 1		✓		✓					✓
กลุ่ม A ภาพ 2		✓			✓		✓		
กลุ่ม A ภาพ 3	✓				✓			✓	
กลุ่ม A ภาพ 4		✓			✓		✓		
กลุ่ม B ภาพ 1			✓		✓				✓
กลุ่ม B ภาพ 2			✓	✓					✓
กลุ่ม B ภาพ 3		✓		✓				✓	
กลุ่ม B ภาพ 4			✓	✓			✓		
กลุ่ม C ภาพ 1		✓		✓					✓
กลุ่ม C ภาพ 2			✓	✓					✓
กลุ่ม C ภาพ 3		✓			✓			✓	
กลุ่ม C ภาพ 4			✓	✓			✓		
กลุ่ม D ภาพ 1			✓			✓			✓
กลุ่ม D ภาพ 2			✓			✓			✓
กลุ่ม D ภาพ 3			✓		✓			✓	
กลุ่ม D ภาพ 4		✓			✓		✓		
กลุ่ม E ภาพ 1			✓			✓			✓
กลุ่ม E ภาพ 2			✓			✓			✓
กลุ่ม E ภาพ 3			✓		✓				✓
กลุ่ม E ภาพ 4		✓		✓				✓	

ตารางที่ ข.1 ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละบุคคล (ต่อ)

ข้อพิจารณา	ผลการประเมินท่านที่ 4				ผลการประเมินท่านที่ 5				ผลการประเมินท่านที่ 6				
	ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				
	4	3	2	1	4	3	2	1	4	3	2	1	
ความสว่าง	✓				✓						✓		กลุ่ม A ภาพ 1
ความคมชัด	✓					✓						✓	
ความเปรียบต่าง	✓				✓					✓			
รายละเอียดของภาพ	✓				✓						✓		
ความสว่าง		✓			✓					✓			กลุ่ม A ภาพ 2
ความคมชัด	✓					✓				✓			
ความเปรียบต่าง	✓					✓				✓			
รายละเอียดของภาพ	✓				✓					✓			
ความสว่าง			✓		✓					✓			กลุ่ม A ภาพ 3
ความคมชัด			✓		✓						✓		
ความเปรียบต่าง		✓				✓					✓		
รายละเอียดของภาพ		✓				✓				✓			
ความสว่าง		✓			✓					✓			กลุ่ม A ภาพ 4
ความคมชัด			✓			✓					✓		
ความเปรียบต่าง			✓			✓					✓		
รายละเอียดของภาพ			✓			✓				✓			
ความสว่าง		✓			✓					✓			กลุ่ม B ภาพ 1
ความคมชัด			✓			✓						✓	
ความเปรียบต่าง			✓			✓				✓			
รายละเอียดของภาพ		✓				✓						✓	
ความสว่าง		✓			✓					✓			กลุ่ม B ภาพ 2
ความคมชัด		✓				✓						✓	
ความเปรียบต่าง		✓				✓				✓			
รายละเอียดของภาพ		✓				✓						✓	
ความสว่าง			✓		✓					✓			กลุ่ม B ภาพ 3
ความคมชัด		✓				✓					✓		
ความเปรียบต่าง		✓				✓				✓			
รายละเอียดของภาพ			✓			✓					✓		
ความสว่าง	✓				✓					✓			กลุ่ม B ภาพ 4
ความคมชัด	✓					✓				✓			
ความเปรียบต่าง		✓				✓						✓	
รายละเอียดของภาพ		✓				✓				✓			
ความสว่าง		✓			✓					✓			กลุ่ม C ภาพ 1
ความคมชัด			✓			✓						✓	
ความเปรียบต่าง			✓			✓					✓		
รายละเอียดของภาพ			✓			✓						✓	

ตารางที่ ข.1 ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละบุคคล (ต่อ)

ข้อพิจารณา	ผลการประเมินท่านที่ 4				ผลการประเมินท่านที่ 5				ผลการประเมินท่านที่ 6				
	ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				
	4	3	2	1	4	3	2	1	4	3	2	1	
ความสว่าง	✓				✓					✓			กลุ่ม C ภาพ 2
ความคมชัด			✓			✓					✓		
ความเปรียบต่าง			✓			✓					✓		
รายละเอียดของภาพ			✓			✓					✓		
ความสว่าง	✓				✓					✓			กลุ่ม C ภาพ 3
ความคมชัด		✓				✓				✓			
ความเปรียบต่าง		✓				✓				✓			
รายละเอียดของภาพ		✓				✓				✓			
ความสว่าง	✓				✓					✓			กลุ่ม C ภาพ 4
ความคมชัด	✓					✓				✓			
ความเปรียบต่าง	✓					✓				✓			
รายละเอียดของภาพ	✓					✓				✓			
ความสว่าง				✓		✓					✓		กลุ่ม D ภาพ 1
ความคมชัด				✓			✓					✓	
ความเปรียบต่าง			✓			✓				✓		✓	
รายละเอียดของภาพ			✓			✓				✓		✓	
ความสว่าง			✓			✓					✓		กลุ่ม D ภาพ 2
ความคมชัด			✓				✓					✓	
ความเปรียบต่าง			✓				✓					✓	
รายละเอียดของภาพ			✓				✓					✓	
ความสว่าง			✓			✓				✓			กลุ่ม D ภาพ 3
ความคมชัด		✓					✓				✓		
ความเปรียบต่าง			✓				✓			✓			
รายละเอียดของภาพ			✓				✓			✓			
ความสว่าง			✓			✓				✓			กลุ่ม D ภาพ 4
ความคมชัด		✓					✓			✓			
ความเปรียบต่าง		✓					✓			✓			
รายละเอียดของภาพ		✓					✓			✓			
ความสว่าง				✓		✓					✓		กลุ่ม E ภาพ 1
ความคมชัด				✓			✓					✓	
ความเปรียบต่าง				✓		✓						✓	
รายละเอียดของภาพ				✓			✓					✓	
ความสว่าง			✓			✓					✓		กลุ่ม E ภาพ 2
ความคมชัด			✓				✓					✓	
ความเปรียบต่าง			✓			✓						✓	
รายละเอียดของภาพ			✓				✓					✓	

ตารางที่ ข.1 ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละบุคคล (ต่อ)

ข้อพิจารณา	ผลการประเมินท่านที่ 4				ผลการประเมินท่านที่ 5				ผลการประเมินท่านที่ 6				
	ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				
	4	3	2	1	4	3	2	1	4	3	2	1	
ความสว่าง		✓				✓				✓			กลุ่ม E ภาพ 3
ความคมชัด		✓					✓					✓	
ความเปรียบต่าง			✓				✓				✓		
รายละเอียดของภาพ			✓					✓			✓		
ความสว่าง		✓				✓				✓			กลุ่ม E ภาพ 4
ความคมชัด		✓					✓				✓		
ความเปรียบต่าง		✓				✓				✓			
รายละเอียดของภาพ		✓				✓					✓		

ตารางที่ ข.1 ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละบุคคล (ต่อ)

ลำดับภาพ	ผลการประเมินท่านที่ 4			ผลการประเมินท่านที่ 5			ผลการประเมินท่านที่ 6		
	ระดับการใช้ทดแทน			ระดับการใช้ทดแทน			ระดับการใช้ทดแทน		
	3	2	1	3	2	1	3	2	1
กลุ่ม A ภาพ 1	✓			✓					✓
กลุ่ม A ภาพ 2		✓			✓			✓	
กลุ่ม A ภาพ 3		✓			✓		✓		
กลุ่ม A ภาพ 4			✓		✓			✓	
กลุ่ม B ภาพ 1			✓			✓			✓
กลุ่ม B ภาพ 2		✓			✓			✓	
กลุ่ม B ภาพ 3		✓			✓			✓	
กลุ่ม B ภาพ 4	✓				✓		✓		
กลุ่ม C ภาพ 1			✓		✓				✓
กลุ่ม C ภาพ 2		✓			✓				✓
กลุ่ม C ภาพ 3		✓			✓			✓	
กลุ่ม C ภาพ 4	✓				✓		✓		
กลุ่ม D ภาพ 1			✓			✓			✓
กลุ่ม D ภาพ 2		✓			✓				✓
กลุ่ม D ภาพ 3		✓			✓				✓
กลุ่ม D ภาพ 4	✓				✓			✓	
กลุ่ม E ภาพ 1			✓			✓			✓
กลุ่ม E ภาพ 2			✓			✓			✓
กลุ่ม E ภาพ 3		✓			✓				✓
กลุ่ม E ภาพ 4	✓				✓			✓	

ตารางที่ ข.1 ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละบุคคล (ต่อ)

ข้อพิจารณา	ผลการประเมินท่านที่ 7				ผลการประเมินท่านที่ 8				ผลการประเมินท่านที่ 9				
	ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				
	4	3	2	1	4	3	2	1	4	3	2	1	
ความสว่าง	✓				✓				✓				กลุ่ม A ภาพ 1
ความคมชัด	✓				✓				✓				
ความเปรียบต่าง	✓				✓				✓				
รายละเอียดของภาพ	✓				✓				✓				
ความสว่าง	✓				✓				✓				กลุ่ม A ภาพ 2
ความคมชัด		✓				✓				✓			
ความเปรียบต่าง	✓				✓					✓	✓		
รายละเอียดของภาพ		✓				✓				✓	✓		
ความสว่าง	✓				✓				✓				กลุ่ม A ภาพ 3
ความคมชัด		✓				✓				✓			
ความเปรียบต่าง	✓				✓					✓	✓		
รายละเอียดของภาพ		✓				✓				✓	✓		
ความสว่าง	✓				✓				✓				กลุ่ม A ภาพ 4
ความคมชัด		✓				✓				✓			
ความเปรียบต่าง	✓				✓					✓	✓		
รายละเอียดของภาพ		✓				✓				✓	✓		
ความสว่าง	✓				✓				✓				กลุ่ม B ภาพ 1
ความคมชัด			✓				✓			✓			
ความเปรียบต่าง			✓				✓			✓	✓		
รายละเอียดของภาพ				✓				✓			✓	✓	
ความสว่าง	✓				✓				✓				กลุ่ม B ภาพ 2
ความคมชัด			✓				✓			✓			
ความเปรียบต่าง			✓				✓			✓	✓		
รายละเอียดของภาพ			✓				✓			✓	✓		
ความสว่าง	✓				✓				✓				กลุ่ม B ภาพ 3
ความคมชัด		✓					✓			✓			
ความเปรียบต่าง		✓					✓			✓	✓		
รายละเอียดของภาพ		✓					✓			✓	✓		
ความสว่าง	✓				✓				✓				กลุ่ม B ภาพ 4
ความคมชัด	✓				✓				✓				
ความเปรียบต่าง	✓				✓				✓				
รายละเอียดของภาพ	✓				✓				✓				
ความสว่าง	✓				✓				✓				กลุ่ม C ภาพ 1
ความคมชัด				✓			✓		✓				
ความเปรียบต่าง			✓				✓			✓	✓		
รายละเอียดของภาพ				✓				✓			✓	✓	

ตารางที่ ข.1 ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละบุคคล (ต่อ)

ข้อพิจารณา	ผลการประเมินท่านที่ 7				ผลการประเมินท่านที่ 8				ผลการประเมินท่านที่ 9				
	ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				
	4	3	2	1	4	3	2	1	4	3	2	1	
ความสว่าง	✓				✓				✓				กลุ่ม C ภาพ 2
ความคมชัด			✓			✓			✓				
ความเปรียบต่าง		✓				✓			✓				
รายละเอียดของภาพ			✓			✓				✓			
ความสว่าง	✓				✓				✓				กลุ่ม C ภาพ 3
ความคมชัด		✓			✓				✓				
ความเปรียบต่าง		✓			✓				✓				
รายละเอียดของภาพ		✓			✓					✓			
ความสว่าง	✓				✓				✓				กลุ่ม C ภาพ 4
ความคมชัด	✓						✓		✓				
ความเปรียบต่าง	✓						✓		✓				
รายละเอียดของภาพ	✓						✓			✓			
ความสว่าง	✓				✓					✓			กลุ่ม D ภาพ 1
ความคมชัด				✓				✓			✓		
ความเปรียบต่าง				✓			✓				✓	✓	
รายละเอียดของภาพ				✓			✓				✓	✓	
ความสว่าง	✓				✓					✓			กลุ่ม D ภาพ 2
ความคมชัด				✓			✓				✓	✓	
ความเปรียบต่าง			✓			✓					✓	✓	
รายละเอียดของภาพ			✓			✓					✓	✓	
ความสว่าง	✓				✓					✓			กลุ่ม D ภาพ 3
ความคมชัด			✓				✓			✓			
ความเปรียบต่าง		✓				✓					✓	✓	
รายละเอียดของภาพ			✓			✓					✓	✓	
ความสว่าง	✓				✓					✓			กลุ่ม D ภาพ 4
ความคมชัด	✓				✓					✓			
ความเปรียบต่าง	✓				✓					✓			
รายละเอียดของภาพ		✓				✓				✓			
ความสว่าง	✓				✓							✓	กลุ่ม E ภาพ 1
ความคมชัด				✓				✓			✓	✓	
ความเปรียบต่าง			✓				✓				✓	✓	
รายละเอียดของภาพ				✓			✓				✓	✓	
ความสว่าง	✓				✓							✓	กลุ่ม E ภาพ 2
ความคมชัด			✓				✓				✓	✓	
ความเปรียบต่าง			✓				✓				✓	✓	
รายละเอียดของภาพ				✓			✓				✓	✓	

ตารางที่ ข.1 ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละบุคคล (ต่อ)

ข้อพิจารณา	ผลการประเมินท่านที่ 7				ผลการประเมินท่านที่ 8				ผลการประเมินท่านที่ 9				
	ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				
	4	3	2	1	4	3	2	1	4	3	2	1	
ความสว่าง	✓				✓							✓	กลุ่ม E ภาพ 3
ความคมชัด		✓				✓						✓	
ความเปรียบต่าง		✓				✓						✓	
รายละเอียดของภาพ			✓				✓					✓	
ความสว่าง	✓				✓							✓	กลุ่ม E ภาพ 4
ความคมชัด	✓				✓							✓	
ความเปรียบต่าง	✓				✓							✓	
รายละเอียดของภาพ		✓				✓						✓	

ตารางที่ ข.1 ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละบุคคล (ต่อ)

ลำดับภาพ	ผลการประเมินท่านที่ 7			ผลการประเมินท่านที่ 8			ผลการประเมินท่านที่ 9		
	ระดับการใช้ทดแทน			ระดับการใช้ทดแทน			ระดับการใช้ทดแทน		
	3	2	1	3	2	1	3	2	1
กลุ่ม A ภาพ 1	✓			✓			✓		
กลุ่ม A ภาพ 2		✓			✓			✓	
กลุ่ม A ภาพ 3		✓			✓			✓	
กลุ่ม A ภาพ 4		✓			✓			✓	
กลุ่ม B ภาพ 1			✓			✓	✓		
กลุ่ม B ภาพ 2			✓			✓	✓		
กลุ่ม B ภาพ 3		✓			✓		✓		
กลุ่ม B ภาพ 4	✓			✓				✓	
กลุ่ม C ภาพ 1			✓			✓		✓	
กลุ่ม C ภาพ 2			✓			✓		✓	
กลุ่ม C ภาพ 3		✓			✓			✓	
กลุ่ม C ภาพ 4	✓			✓				✓	
กลุ่ม D ภาพ 1			✓			✓			✓
กลุ่ม D ภาพ 2		✓			✓				✓
กลุ่ม D ภาพ 3			✓			✓			✓
กลุ่ม D ภาพ 4	✓			✓				✓	
กลุ่ม E ภาพ 1			✓			✓			✓
กลุ่ม E ภาพ 2			✓			✓			✓
กลุ่ม E ภาพ 3			✓			✓			✓
กลุ่ม E ภาพ 4		✓			✓				✓

ตารางที่ ข.1 ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละบุคคล (ต่อ)

ข้อพิจารณา	ผลการประเมินท่านที่ 10				ผลการประเมินท่านที่ 11				ผลการประเมินท่านที่ 12				
	ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				
	4	3	2	1	4	3	2	1	4	3	2	1	
ความสว่าง		✓			✓					✓			กลุ่ม A ภาพ 1
ความคมชัด		✓				✓			✓				
ความเปรียบต่าง		✓			✓					✓			
รายละเอียดของภาพ		✓				✓			✓				
ความสว่าง		✓			✓				✓				กลุ่ม A ภาพ 2
ความคมชัด		✓				✓				✓			
ความเปรียบต่าง		✓			✓				✓				
รายละเอียดของภาพ		✓				✓				✓			
ความสว่าง		✓			✓				✓				กลุ่ม A ภาพ 3
ความคมชัด		✓			✓					✓			
ความเปรียบต่าง			✓		✓				✓				
รายละเอียดของภาพ			✓		✓					✓			
ความสว่าง		✓			✓				✓				กลุ่ม A ภาพ 4
ความคมชัด		✓			✓					✓			
ความเปรียบต่าง			✓		✓				✓				
รายละเอียดของภาพ			✓		✓				✓				
ความสว่าง	✓				✓					✓			กลุ่ม B ภาพ 1
ความคมชัด		✓				✓					✓		
ความเปรียบต่าง	✓				✓					✓		✓	
รายละเอียดของภาพ	✓					✓		✓			✓		
ความสว่าง	✓				✓				✓				กลุ่ม B ภาพ 2
ความคมชัด		✓				✓				✓		✓	
ความเปรียบต่าง	✓				✓				✓		✓		
รายละเอียดของภาพ	✓					✓		✓		✓			
ความสว่าง	✓				✓					✓			กลุ่ม B ภาพ 3
ความคมชัด	✓					✓					✓		
ความเปรียบต่าง	✓				✓				✓				
รายละเอียดของภาพ	✓					✓		✓		✓			
ความสว่าง	✓				✓				✓				กลุ่ม B ภาพ 4
ความคมชัด	✓					✓			✓				
ความเปรียบต่าง	✓				✓				✓				
รายละเอียดของภาพ	✓					✓		✓		✓			
ความสว่าง		✓			✓				✓				กลุ่ม C ภาพ 1
ความคมชัด			✓			✓				✓		✓	
ความเปรียบต่าง			✓			✓				✓			
รายละเอียดของภาพ		✓					✓		✓		✓		



ตารางที่ ข.1 ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละบุคคล (ต่อ)

ข้อพิจารณา	ผลการประเมินท่านที่ 10				ผลการประเมินท่านที่ 11				ผลการประเมินท่านที่ 12				
	ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				
	4	3	2	1	4	3	2	1	4	3	2	1	
ความสว่าง		✓			✓				✓				กลุ่ม C ภาพ 2
ความคมชัด		✓				✓					✓		
ความเปรียบต่าง			✓			✓				✓			
รายละเอียดของภาพ		✓					✓				✓		
ความสว่าง	✓				✓				✓				กลุ่ม C ภาพ 3
ความคมชัด	✓					✓				✓			
ความเปรียบต่าง		✓			✓						✓		
รายละเอียดของภาพ	✓					✓					✓		
ความสว่าง	✓				✓				✓				กลุ่ม C ภาพ 4
ความคมชัด	✓				✓					✓			
ความเปรียบต่าง	✓				✓					✓			
รายละเอียดของภาพ	✓				✓					✓			
ความสว่าง		✓			✓				✓				กลุ่ม D ภาพ 1
ความคมชัด			✓					✓				✓	
ความเปรียบต่าง		✓					✓					✓	
รายละเอียดของภาพ			✓				✓					✓	
ความสว่าง	✓				✓				✓				กลุ่ม D ภาพ 2
ความคมชัด			✓					✓				✓	
ความเปรียบต่าง		✓					✓					✓	
รายละเอียดของภาพ		✓					✓					✓	
ความสว่าง	✓				✓				✓				กลุ่ม D ภาพ 3
ความคมชัด		✓						✓				✓	
ความเปรียบต่าง		✓					✓					✓	
รายละเอียดของภาพ		✓					✓					✓	
ความสว่าง	✓				✓				✓				กลุ่ม D ภาพ 4
ความคมชัด	✓							✓				✓	
ความเปรียบต่าง	✓							✓				✓	
รายละเอียดของภาพ	✓							✓				✓	
ความสว่าง			✓		✓				✓				กลุ่ม E ภาพ 1
ความคมชัด				✓				✓				✓	
ความเปรียบต่าง			✓				✓					✓	
รายละเอียดของภาพ				✓			✓					✓	
ความสว่าง		✓			✓				✓				กลุ่ม E ภาพ 2
ความคมชัด			✓					✓				✓	
ความเปรียบต่าง			✓				✓					✓	
รายละเอียดของภาพ				✓			✓					✓	

ตารางที่ ข.1 ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละบุคคล (ต่อ)

ข้อพิจารณา	ผลการประเมินท่านที่ 10				ผลการประเมินท่านที่ 11				ผลการประเมินท่านที่ 12				
	ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				ระดับคุณภาพของภาพ				
	4	3	2	1	4	3	2	1	4	3	2	1	
ความสว่าง		✓			✓					✓			กลุ่ม E ภาพ 3
ความคมชัด			✓				✓				✓		
ความเปรียบต่าง		✓				✓				✓			
รายละเอียดของภาพ		✓					✓			✓			
ความสว่าง	✓				✓				✓				กลุ่ม E ภาพ 4
ความคมชัด		✓				✓				✓			
ความเปรียบต่าง		✓				✓				✓			
รายละเอียดของภาพ		✓				✓				✓			

ตารางที่ ข.1 ผลการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละบุคคล (ต่อ)

ลำดับภาพ	ผลการประเมินท่านที่ 10			ผลการประเมินท่านที่ 11			ผลการประเมินท่านที่ 12		
	ระดับการใช้ทดแทน			ระดับการใช้ทดแทน			ระดับการใช้ทดแทน		
	3	2	1	3	2	1	3	2	1
กลุ่ม A ภาพ 1	✓				✓		✓		
กลุ่ม A ภาพ 2	✓				✓			✓	
กลุ่ม A ภาพ 3		✓			✓			✓	
กลุ่ม A ภาพ 4		✓		✓				✓	
กลุ่ม B ภาพ 1		✓				✓			✓
กลุ่ม B ภาพ 2		✓				✓			✓
กลุ่ม B ภาพ 3	✓				✓			✓	
กลุ่ม B ภาพ 4	✓				✓		✓		
กลุ่ม C ภาพ 1		✓				✓			✓
กลุ่ม C ภาพ 2		✓				✓			✓
กลุ่ม C ภาพ 3	✓				✓				✓
กลุ่ม C ภาพ 4	✓			✓				✓	
กลุ่ม D ภาพ 1			✓			✓			✓
กลุ่ม D ภาพ 2		✓				✓			✓
กลุ่ม D ภาพ 3	✓					✓			✓
กลุ่ม D ภาพ 4	✓				✓			✓	
กลุ่ม E ภาพ 1			✓			✓			✓
กลุ่ม E ภาพ 2			✓			✓			✓
กลุ่ม E ภาพ 3		✓				✓			✓
กลุ่ม E ภาพ 4	✓				✓			✓	

ตารางที่ ข.2 ผลรวมการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละภาพ

ข้อพิจารณา	ผลรวมการประเมิน กลุ่ม A ภาพที่ 1				ผลรวมการประเมิน กลุ่ม A ภาพที่ 2				ผลรวมการประเมิน กลุ่ม A ภาพที่ 3				ผลรวมการประเมิน กลุ่ม A ภาพที่ 4			
	จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ			
	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1
ความสว่าง	7	3	1	7	4	1	0	1	7	5	0	0	6	4	2	0
ความคมชัด	4	6	0	2	6	4	0	2	2	8	2	0	2	6	4	0
ความเปรียบต่าง	6	5	0	4	3	5	0	1	6	4	2	0	5	3	4	0
รายละเอียดของภาพ	6	3	2	2	6	4	0	1	2	8	2	0	2	7	3	0

ตารางที่ ข.2 ผลรวมการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละภาพ (ต่อ)

ข้อพิจารณา	ผลรวมการประเมิน กลุ่ม B ภาพที่ 1				ผลรวมการประเมิน กลุ่ม B ภาพที่ 2				ผลรวมการประเมิน กลุ่ม B ภาพที่ 3				ผลรวมการประเมิน กลุ่ม B ภาพที่ 4			
	จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ			
	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1
ความสว่าง	5	6	0	9	2	1	0	6	5	1	0	1	7	4	1	0
ความคมชัด	0	1	8	7	4	1	0	1	8	3	0	3	0	3	8	1
ความเปรียบต่าง	2	3	6	7	4	1	0	2	9	1	0	1	2	6	4	0
รายละเอียดของภาพ	1	1	4	5	5	2	0	1	8	3	0	6	1	3	6	2

ตารางที่ ข.2 ผลรวมการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละภาพ (ต่อ)

ข้อพิจารณา	ผลรวมการประเมิน กลุ่ม C ภาพที่ 1				ผลรวมการประเมิน กลุ่ม C ภาพที่ 2				ผลรวมการประเมิน กลุ่ม C ภาพที่ 3				ผลรวมการประเมิน กลุ่ม C ภาพที่ 4			
	จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ			
	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1
ความสว่าง	6	8	3	1	0	8	3	1	0	4	1	1	10	1	1	0
ความคมชัด	0	0	6	6	0	1	10	1	0	3	4	5	6	6	0	0
ความเปรียบต่าง	0	0	7	5	0	1	9	2	0	3	8	1	7	4	1	0
รายละเอียดของภาพ	0	0	3	9	0	1	8	3	0	3	3	6	7	3	2	0

ตารางที่ ข.2 ผลรวมการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละภาพ (ต่อ)

ข้อพิจารณา	ผลรวมการประเมิน กลุ่ม D ภาพที่ 1				ผลรวมการประเมิน กลุ่ม D ภาพที่ 2				ผลรวมการประเมิน กลุ่ม D ภาพที่ 3				ผลรวมการประเมิน กลุ่ม D ภาพที่ 4			
	จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ			
	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1
ความสว่าง	4	4	5	7	3	2	0	5	2	0	2	2	5	3	3	1
ความคมชัด	0	0	0	4	5	3	0	3	9	0	3	9	0	0	5	7
ความเปรียบต่าง	0	1	0	4	4	4	0	6	5	1	5	6	0	1	8	3
รายละเอียดของภาพ	0	0	0	2	5	5	0	3	7	2	4	8	0	1	5	6

ตารางที่ ข.2 ผลรวมการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละภาพ (ต่อ)

ข้อพิจารณา	ผลรวมการประเมิน กลุ่ม E ภาพที่ 1				ผลรวมการประเมิน กลุ่ม E ภาพที่ 2				ผลรวมการประเมิน กลุ่ม E ภาพที่ 3				ผลรวมการประเมิน กลุ่ม E ภาพที่ 4			
	จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ			
	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1
ความสว่าง	3	2	4	3	3	3	4	2	4	6	1	1	6	5	0	1
ความคมชัด	0	0	1	11	0	0	6	6	0	3	7	2	2	6	3	1
ความเปรียบต่าง	0	1	5	6	0	1	8	3	0	4	7	1	2	9	0	1
รายละเอียดของภาพ	0	0	1	11	0	0	3	9	0	2	8	2	0	10	1	1

ตารางที่ ข.2 ผลรวมการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละภาพ (ต่อ)

ข้อพิจารณา	ผลรวมการประเมิน ภาพรายละเอียดต่ำ ภาพที่ 1				ผลรวมการประเมิน ภาพรายละเอียดต่ำ ภาพที่ 2				ผลรวมการประเมิน ภาพรายละเอียดต่ำ ภาพที่ 3				ผลรวมการประเมิน ภาพรายละเอียดต่ำ ภาพที่ 4			
	จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ			
	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1
ความสว่าง	13	7	2	17	5	2	0	2	15	8	1	0	14	7	3	0
ความคมชัด	4	9	4	8	12	4	0	7	2	14	8	0	3	16	5	0
ความเปรียบต่าง	6	8	8	11	7	6	0	2	6	11	7	0	6	12	6	0
รายละเอียดของภาพ	6	6	5	9	9	6	0	7	2	11	11	0	3	15	6	0

ตารางที่ ข.2 ผลรวมการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละภาพ (ต่อ)

ข้อพิจารณา	ผลรวมการประเมิน ภาพรายละเอียดสูง ภาพที่ 1				ผลรวมการประเมิน ภาพรายละเอียดสูง ภาพที่ 2				ผลรวมการประเมิน ภาพรายละเอียดสูง ภาพที่ 3				ผลรวมการประเมิน ภาพรายละเอียดสูง ภาพที่ 4			
	จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ				จำนวนคนประเมิน ระดับคุณภาพของภาพ			
	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1
ความสว่าง	12	12	6	6	15	10	8	3	15	16	4	1	22	10	3	1
ความคมชัด	0	1	12	23	0	3	19	14	1	14	19	2	13	15	7	1
ความเปรียบต่าง	2	5	16	13	2	8	20	6	2	19	13	2	13	17	5	1
รายละเอียดของภาพ	1	1	9	25	1	4	14	17	1	13	18	4	7	20	8	1

ตารางที่ ข.2 ผลรวมการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละภาพ (ต่อ)

ลำดับภาพ	ผลรวมการประเมิน กลุ่ม A			ผลรวมการประเมิน กลุ่ม B			ผลรวมการประเมิน กลุ่ม C		
	จำนวนคนประเมิน ระดับการใช้ทดแทน			จำนวนคนประเมิน ระดับการใช้ทดแทน			จำนวนคนประเมิน ระดับการใช้ทดแทน		
	3	2	1	3	2	1	3	2	1
ภาพ 1	8	2	2	1	2	9	1	4	7
ภาพ 2	2	10	0	2	3	7	1	5	6
ภาพ 3	2	10	0	3	8	1	1	10	1
ภาพ 4	2	9	1	8	3	1	7	4	1

ตารางที่ ข.2 ผลรวมการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละภาพ (ต่อ)

ลำดับภาพ	ผลรวมการประเมิน กลุ่ม D			ผลรวมการประเมิน กลุ่ม E		
	จำนวนคนประเมิน ระดับการใช้ทดแทน			จำนวนคนประเมิน ระดับการใช้ทดแทน		
	3	2	1	3	2	1
ภาพ 1	0	0	12	0	0	12
ภาพ 2	0	4	8	0	0	12
ภาพ 3	1	3	8	0	3	9
ภาพ 4	5	7	0	3	8	1

ตารางที่ ข.2 ผลรวมการประเมินคุณภาพของภาพที่ผ่านการบีบอัดข้อมูลภาพของแต่ละภาพ (ต่อ)

ลำดับภาพ	ผลรวมการประเมิน ภาพรายละเอียดต่ำ			ผลรวมการประเมิน ภาพรายละเอียดสูง		
	จำนวนคนประเมิน ระดับการใช้ทดแทน			จำนวนคนประเมิน ระดับการใช้ทดแทน		
	3	2	1	3	2	1
ภาพ 1	9	6	9	1	2	33
ภาพ 2	3	15	6	2	7	27
ภาพ 3	3	20	1	4	14	18
ภาพ 4	9	13	2	16	18	2

## ภาคผนวก ค

รายละเอียดโปรแกรมการเข้ารหัสอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง

### โปรแกรม SPIHT.h

```

/*-----*/
// จัดทำโดย: นาย ฐิฐนันท์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี
// ,ก.พ. 2545
/*-----*/
//DWT and IDWT (1D and 2D) -> bi9-7
void conv(float* out,float* hn,float* xn,int size_h,int
size_x);
void conv_p(float* hn,float* xn,int size_x,int size_h);
//DWT
void refind_data_bi9(float* out,float* in,int size_x);
void refind_data_bi7(float* out,float* in,int size_x);
void bi_dwt(float* inx,int size_x);
void down_sampling(float* in,int st,int nxt,int type);
void bi_dwt2(float** fdwt,int s_x,int s_y);
//IDWT
void bi_idwt(float* inx,int size_x);
void up_sampling(float* out,float* in,int size_x,int type);
void bi_idwt2(float** fdwt,int s_x,int s_y);
//Encode SPIHT
void coor_offspring(int** O_list,int i,int j,int s_x,int s_y,int
level);
void coor_descendants(int* i,int j,float threshold,float**
fdwt,int s_x,int s_y,int level);
//LFC SPIHT
void LFC_list(int* q_LFC,float** image_in,int s_x,int
s_y,int level);
void cut_image(float** output,float** input,int x_start,int
x_end,int y_start,int y_end);
void dis_max_abs(float* max_out,float** input,int ss_x,int
ss_y);
int type_label(int x,int y,int s_x,int s_y,int level);
int floor_LFC_n(float max_out_n,int max_n);
int log2_floor(float data);
void sn_out(float* output,int* LIP_x,int* LIP_y,int
t,float** fdwt,float num_n);
void cut_LIP(long* count_LIP_new,int* LIP_x,int*
LIP_y,long count_LIP);

```

```

void cut_LIS(long* count_LIS_new,int* LIS_x,int*
LIS_y,int* LIS_type,long count_LIS);
void refinement_pass(long* count_new,char* imout,int*
LSP_abs,long count_LSP,float num_n,long bit_end,long
count);
int find_max_coff(float** fdwt,int s_x,int s_y,int level);
//SPIHT
void SPIHT_Encode(char* LFC_list_type,long*
countp,char* imout,int end_bit,float** fdwt,int s_x,int
s_y,int level);
void SPIHT_Decode(char* LFC_list_type,float**
output_image,char* imout,int bit_end,int s_x,int s_y,int
level,int n);
void i_coor_descendants(int** O_list,int* type_ij,int i,int
j,int s_x,int s_y,int level);
void i_refinement_pass(long* count_new,char* imout,int*
LSP_abs,long count_LSP,float num_n,long bit_end,long
count);
void re_image(int** output_image,int* LSP_x,int*
LSP_y,int* LSP_abs,int* LSP_sign,long count_LSP);
void coor_descendants_s(int* type_ij,int i,int j,float
threshold,float** fdwt,int s_x,int s_y,int level);
void coor_offspring_s(int* countO,int** O_list,int i,int j,int
s_x,int s_y,int level);
void i_coor_descendants_s(int *count_set,int** O_list,int*
type_ij,int i,int j,int s_x,int s_y,int level);
void round_image_pass(float** input,int s_x,int s_y,float
bitrate);
void round_image_abs(float** input,int s_x,int s_y);
void re_image_pass(float** output_image,int* LSP_x,int*
LSP_y,int * LSP_abs,long count_LSP);

```

### โปรแกรม coff.h

```

/*-----*/
// จัดทำโดย: นาย ฐิฐนันท์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี
// ,ก.พ. 2545
/*-----*/
/*decomposition lowpass*/
#define bdl0 0.03782879857992

```



```

#define bdl1 -0.02384929751586
#define bdl2 -0.11062402748951
#define bdl3 0.37740268810913
#define bdl4 0.85269865321930
#define bdl5 0.37740268810913
#define bdl6 -0.11062402748951
#define bdl7 -0.02384929751586
#define bdl8 0.03782879857992
/*decomposition highpass*/
#define bdh0 0.06453905013246
#define bdh1 -0.04068975261660
#define bdh2 -0.41809244072573
#define bdh3 0.78848487220618
#define bdh4 -0.41809244072573
#define bdh5 -0.04068975261660
#define bdh6 0.06453905013246
/*reconsturction lowpass*/
#define brl0 -0.06453905013246
#define brl1 -0.04068975261660
#define brl2 0.41809244072573
#define brl3 0.78848487220618
#define brl4 0.41809244072573
#define brl5 -0.04068975261660
#define brl6 -0.06453905013246
/*reconsturction highpass*/
#define brh0 0.03782879857992
#define brh1 0.02384929751586
#define brh2 -0.11062402748951
#define brh3 -0.37740268810913
#define brh4 0.85269865321930
#define brh5 -0.37740268810913
#define brh6 -0.11062402748951
#define brh7 0.02384929751586
#define brh8 0.03782879857992

โปรแกรม constant.h
/*-----*/
// จัดทำโดย: นาย ฉัฐนันท์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี
// ,ก.พ. 2545

/*-----*/
////////// 2-D //////////
//input and DWT,IDWT
#define r_xx 960 //(row-Image-960)
#define r_yy 1280 //(column-Image-1280)
//////////
#define level_s 5 //Level of Wavelet Tranfrom
#define max_nr -1 //Check No root of o(i,j)
#define step_quan 12 //Set step_Quantization

โปรแกรม main.cpp
/*-----*/
// จัดทำโดย: นาย ฉัฐนันท์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี
// ,ก.พ. 2545
/*-----*/
#include"SPIHT.h"
#include"coeff.h"
main(){
//////////
/* Transfer Data from VC++ to MATLAB
// Set r_x,r_y (size Data input)
//////////
FILE *p_read; long int numread;
long int count_r = 0; int i,r_i,r_j;
//generate size pointer <Dynamic >
unsigned char *point_read = new unsigned char
[r_xx*r_yy]; //for DATA IMAGE
float **fdwt = new float*[r_xx];
float **imin = new float*[r_xx];
for( i=0; i<r_xx; i++)
{fdwt[i] = new float[r_yy]; imin[i] = new float[r_yy];}
p_read = fopen("histioc1.dat","rb");
numread = fread(point_read, sizeof(char), r_xx*r_yy ,
p_read);
for (r_i = 0; r_i < r_yy ; r_i++)
{for (r_j = 0; r_j < r_xx ; r_j++)
{fdwt[r_j][r_i] = (float) point_read[count_r];
count_r = count_r + 1;}}
count_r = 0;

```

```

for (r_i = 0; r_i < r_yy ; r_i++)
{for (r_j = 0; r_j < r_xx ; r_j++)
{min[r_j][r_i] = (float)point_read[count_r];
count_r = count_r + 1;}}
fclose(p_read);
delete[] point_read;
//// DWT
bi_dwt2(fdwt,r_xx,r_yy);
bi_dwt2(fdwt,r_xx/2,r_yy/2);
bi_dwt2(fdwt,r_xx/4,r_yy/4);
bi_dwt2(fdwt,r_xx/8,r_yy/8);
bi_dwt2(fdwt,r_xx/16,r_yy/16);
////////////////////////////////////
* SPIHT -> Encode
////////////////////////////////////*/
//DATA HEADER
int s_x = r_xx; //->Dimension X
int s_y = r_yy; //->Dimension Y
int level = level_s; //-> level wavelet tranfrom
//Initial DATA SPIHT = end_bit (bit)
int bit_end,n; float bitrate;
printf("bit rate = ");
scanf("%f",&bitrate);
bit_end = (int)(s_x*s_y*bitrate);
char* imout = new char[bit_end];
n = find_max_coff(fdwt,s_x,s_y,level);
long* countp = new long[1];
//LFC List
int count_LFC;
count_LFC = (level*4*3) + 4;
char* LFC_list_type = new char[count_LFC];
//Encode SPIHT + LFC
round_image_pass(fdwt,r_xx,r_yy,bitrate);
SPIHT_Encode(LFC_list_type,countp,imout,bit_end,fdwt,s_x,s_y,level);
////////////////////////////////////
// save data
FILE *pimout_write;
// Open file in SUT mode:
pimout_write = fopen("output.sut","wb");

```

```

//generate size pointer <Dynamic >
fwrite(imout,sizeof(char),bit_end,pimout_write);
fclose( pimout_write );
FILE *pheader; pheader = fopen("header.sim","wb");
int header[1];
header[0] = countp[0]; //size of data wavelet SPIHT
fwrite(header,sizeof(int),1,pheader);
header[0] = s_x ; //hight image
fwrite(header,sizeof(int),1,pheader);
header[0] = s_y ; //wide image
fwrite(header,sizeof(int),1,pheader);
header[0] = level ; //level decomposition
fwrite(header,sizeof(int),1,pheader);
header[0] = n; //number step Quantization
fwrite(header,sizeof(int),1,pheader);
fwrite(LFC_list_type,1,count_LFC,pheader);
fclose(pheader);
return(0);}

```

#### โปรแกรม bi\_dwt.cpp

```

/*-----*/
// จัดทำโดย: นาย ฉัฐนันท์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี
// ,ภ.พ. 2545
/*-----*/
#include "SPIHT.h"
#include "coff.h"
////////////////////////////////////
//Bi-orthogonal 9-7
////////////////////////////////////
void bi_idwt2(float** fdwt,int s_x,int s_y){
% Reconstruction 2-D Wavelet Tranform
% yA is input lowpass
% yD is input highpass
% CL is filter coefficient (lowpass)
% CH is filter coefficient (highpass)
% y is Image reconstruction output*/
long int b,g; float *ll = new float[s_x];
float *hh = new float[s_y];
//% columns image
for (b=0; b<s_y ; b++)

```

```

{ for(g=0; g<s_x ; g++){ll[g] = fdwt[g][b];}
bi_idwt(ll,s_x);
for(g=0; g<s_x ; g++){fdwt[g][b] = ll[g];}
%% rows image
for (b=0 ; b<s_x ; b++)
{for(g=0 ; g<s_y ; g++){hh[g] = fdwt[b][g];}
bi_idwt(hh,s_y);
for(g=0 ; g<s_y ; g++){fdwt[b][g] = hh[g];}
delete[] hh; delete[] ll;}
////////////////////////////////////
void bi_dwt2(float** fdwt,int s_x,int s_y){
% Decomposition 2-D Wavelet Transform
% x is Image input
% ld is filter coefficient (lowpass)
% hd is filter coefficient (highpass)
% cll is coff Approximation cll
% chl is coff Horizontail Detail chl
% clh is coff Vertical Detail clh
% chh is coff Diagonal Detail chh*/
int b,g,pp;
%%Decomposition
if (s_x >= s_y){pp = s_x;} else {pp = s_y;}
float *hh = new float[pp];
%% rows image
for (b=0 ; b<s_x ; b++)
{for(g=0 ; g<s_y ; g++){hh[g] = fdwt[b][g];}
bi_dwt(hh,s_y);
for(g=0 ; g<s_y ; g++){fdwt[b][g] = hh[g];}
%% columns image
for (b=0; b<s_y ; b++)
{for(g=0; g<s_x ; g++){hh[g] = fdwt[g][b];}
bi_dwt(hh,s_x);
for(g = 0; g<s_x ; g++){fdwt[g][b] = hh[g];}
delete[] hh;}
////////////////////////////////////
void bi_dwt(float* inx,int size_x){
/*decomposition lowpass*/
float bdl[9] = {(float)bdl0,(float)bdl1,(float)bdl2,(float)
bdl3,(float)bdl4,(float)bdl5,(float)bdl6,(float)bdl7,(float)
bdl8};
/*decomposition highpass*/
float bdh[7] = {(float)bdh0,(float)bdh1,(float)bdh2,(float)
bdh3,(float)bdh4,(float)bdh5,(float)bdh6};
int gl=4; int gh=3;
float *out_l = new float[size_x+8];
float *out_h = new float[size_x+6];
float *o_l = new float[size_x+8];
float *o_h = new float[size_x+6];
refind_data_bi9(out_l,inx,size_x);
refind_data_bi7(out_h,inx,size_x);
conv(o_l,bdl,out_l,9,size_x+8);
int st=8; int nxt=size_x+8;
down_sampling(o_l,st,nxt,1);
conv(o_h,bdh,out_h,7,size_x+6);
st=6; nxt=size_x+6;
down_sampling(o_h,st,nxt,0);
for(int z=0; z<(size_x/2) ; z++){
{inx[z] = o_l[z]; inx[z+(size_x/2)] = o_h[z];}
delete[] out_l; delete[] out_h; delete[] o_l; delete[] o_h;}
////////////////////////////////////
void bi_idwt(float* inx,int size_x)
{float *out_l = new float[size_x+6];
float *out_h = new float[size_x+8];
/*decomposition lowpass*/
float brh[9] = {(float)brh0,(float)brh1,(float)brh2,(float)
brh3,(float)brh4,(float)brh5,(float)brh6,(float)brh7,(float)
brh8};
/*decomposition highpass*/
float brl[7] = {(float)brl0,(float)brl1,(float)brl2,(float)brl3,
(float)brl4,(float)brl5,(float)brl6};
int gh=4; int gl=3;
float *l = new float[(size_x/2)];
float *h = new float[(size_x/2)];
for(int t=0 ; t<(size_x/2) ; t++){[t] = inx[t];
h[t] = inx[t+(size_x/2)];}
up_sampling(inx,l,size_x,1);
refind_data_bi7(out_l,inx,size_x);
up_sampling(inx,h,size_x,0);
refind_data_bi9(out_h,inx,size_x);
delete[] l; delete[] h;

```

```

conv_p(brl,out_l,7,size_x+6);
conv_p(brh,out_h,9,size_x+8);
for(t = 0; t<size_x ;t++)
{inx[t] = out_l[t+6] + out_h[t+8];}
delete[] out_l; delete[] out_h;}
////////////////////////////////////
void conv(float* out,float* hn,float* xn,int size_x,int
size_h){int check;
for(int t=0; t<size_h ; t++){out[t] = 0;
for(int p=0; p<size_x ; p++){check = t-p;
if (check >= 0){out[t] = out[t] + hn[p]*xn[check];}
else{break;}}}
////////////////////////////////////
void refind_data_bi9(float* out,float* in,int size_x)
{//[fliplr(x(2:gl+1)) x fliplr(x(N-gl:N-1))];
out[0] = in[4]; out[1] = in[3];
out[2] = in[2]; out[3] = in[1];
for(int t=0; t < size_x ; t++) {out[t+4] = in[t];}
out[size_x+4] = in[size_x-2];
out[size_x+5] = in[size_x-3];
out[size_x+6] = in[size_x-4];
out[size_x+7] = in[size_x-5];}
////////////////////////////////////
void refind_data_bi7(float* out,float* in,int size_x)
{out[0] = in[3]; out[1] = in[2]; out[2] = in[1];
for(int t=0; t < size_x ; t++){out[t+3] = in[t];}
out[size_x+3] = in[size_x-2];
out[size_x+4] = in[size_x-3];
out[size_x+5] = in[size_x-4];}
////////////////////////////////////
void up_sampling(float* out,float* in,int size_x,int type)
{//type = 0 -> odd ,= 1 -> even
int check = type; int count = 0;
for(int t=0 ; t<size_x ;t++)
{if(check==1) {out[t] = in[count];
count = count + 1; check = 0;}
else {out[t] = 0; check = 1;}}
////////////////////////////////////
void down_sampling(float* in,int st,int nxt,int type)
{//type = 0 -> odd ,= 1 -> even

```

```

int check;check = type; int count=0;
for(int d=st; d<nxt ; d++)
{if (check==1){in[count]=in[d];
count = count + 1; check = 0;}
else{check = 1;}}
////////////////////////////////////
void conv_p(float* hn,float* xn,int size_x,int size_h)
{int check,t; float* out_s = new float[size_h];
for(t=0; t<size_h ; t++){out_s[t] = 0;
for(int p=0; p<size_x ; p++) {check = t-p;
if (check >= 0){out_s[t] = out_s[t] + hn[p]*xn[check];}
else{break;}}} for(t=0; t<size_h ; t++) {xn[t] = out_s[t];}
delete[] out_s;}

```

### โปรแกรม Encode\_SPIHT\_main.cpp

```

/*-----*/
// จัดทำโดย: นาย ฉัฐนันท์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี
// ,ก.พ. 2545
/*-----*/
#include"SPIHT.h"
void SPIHT_Encode(char* LFC_list_type,long*
countp,char* imout,int bit_end,float** fdwt,int s_x,int
s_y,int level)
{//Initial DATA SPIHT = end_bit (bit) -> imout
//%1)Initialization :
output n = [log2(max(i,j){ci,j})]
int n,nl; float num_n; double number; int first_x,first_y;
//limit of first Layer and finit Layer
//first Layer
number = pow((double)2,(double)level);
//pow(x,y) -> x to y power
nl= (int)number; first_x = s_x/(nl); first_y = s_y/(nl);
float* max_out = new float[4];
dis_max_abs(max_out,fdwt,s_x,s_y);
float max_nn,tii; max_nn = 0;
for (int e = 0 ; e < 4 ; e++)
{ tii = max_out[e];
if (max_nn < tii) { max_nn = tii; }
n = log2_floor(max_nn);

```

```

number    = pow((double)2,(double)n);
num_n     = (float)number;
long count,count_LSP,count_LIP;
long count_LIS,count_LSP_old;
count = 0; count_LSP = 0; count_LIP = 0;
count_LIS = 0; count_LSP_old = 0;
%%Initial LIP size=[s_x*s_y 3]
int* LSP_x = new int[s_x*s_y];
int* LSP_y = new int[s_x*s_y];
int* LSP_abs = new int[s_x*s_y];
%%Initial LIP size=[s_x*s_y 2]
int* LIP_x = new int[s_x*s_y];
int* LIP_y = new int[s_x*s_y];
int i,j,oi,oj;
for (i=0 ; i < first_x ; i++)
{ for (j=0 ; j < first_y ; j++)
{LIP_x[count_LIP] = i;
LIP_y[count_LIP] = j;
count_LIP = count_LIP + 1;}}
%%Initial LIS size=[s_x*s_y 3]
int* LIS_x = new int[s_x*s_y];
int* LIS_y = new int[s_x*s_y];
int* LIS_type = new int[s_x*s_y];
for ( i=0 ; i < first_x ; i++){
for (j=0 ; j < first_y ; j++)
{//Check odd and even
oi = i%2; oj = j%2;
//Check i
if (oi == 0) { oi = 0; } else { oi = 1; }
//Check j
if (oj == 0) { oj = 0; } else { oj = 1; }
if ( (oi+oj)!= 0 ) {
LIS_x[count_LIS] = i; LIS_y[count_LIS] = j;
LIS_type[count_LIS] = 0; %%->set type A
count_LIS = count_LIS +1;}}
%%1.2 Make it LFC
//size of LFC
int count_LFC; count_LFC = (level*4*3) + 4;
int* LFC_type = new int[count_LFC];
LFC_list(LFC_type,fdwt,s_x,s_y,level);
for(e=0; e < count_LFC ; e++)
{ LFC_list_type[e] = (char)LFC_type[e]; }
for (int step_Quantization = 1;
step_Quantization <= step_quan ; step_Quantization++) {
%%2)Sorting pass:
%%2.1 for each entry (i,j) in the LIP
for (int t=0; t < count_LIP; t++)
%%2.1.1 output Sn(i,j);
{if (bit_end <= count){break;}
float* output = new float[3];
sn_out(output,LIP_x,LIP_y,t,fdwt,num_n);
imout[count] = (int)output[0]; count = count + 1;
%%2.1.2) Sn(i,j) = 1 then move (i,j) to the LSP and output
the sign of cij
if ((int)output[0] == 1)
{LSP_x[count_LSP] = LIP_x[t];
LSP_y[count_LSP] = LIP_y[t];
LSP_abs[count_LSP] = (int)output[1];
count_LSP = count_LSP+1;
%%Output the sign of cij
if (output[2] != 0){ imout[count] = 1; }
else{ imout[count] = 0;}
%%END of Output the sign of cij
count = count + 1;
LIP_x[t] = -1; LIP_y[t] = -1; }
%%End of the Sn(i,j)=1
if (bit_end <= count){break;}
}
%%END of is Sn(i,j) = then move (i,j) to the LSP
%%2.1.2) Move LIP(i,j) at sn(i,j)=1
long* count_LIP_new = new long[1];
cut_LIP(count_LIP_new,LIP_x,LIP_y,count_LIP);
count_LIP = count_LIP_new[0];
%% END of Loop for each entry (i,j) in the LIP
%%2.2 for each entry (i,j) in the LIS
%%Initial parameter 2.2
long count_LIS_old;
count_LIS_old = count_LIS;
int change,set_add,ccp;
change = 100;
set_add = 0;

```

```

ccp = 0;
for (long LIS_length = 0 ; LIS_length < 100000000 ;
LIS_length++){
if ((change == 0) && (count_LIS_old <= LIS_length))
{break;}
if (LIS_length==0)
{change = 0;}
if ((count_LIS_old <= LIS_length) && set_add==1)
{ change = change - 1 ; }
%%Find Do(i,j) , LO(i,j)
i = LIS_x[LIS_length]; j = LIS_y[LIS_length];
%%Check LFC list
int LFC_D = 0; %%LFC O(i,j) int LFC_L = 0;
%%LFC L(i,j)
//O_list
int **DOij_list = new int*[4];
for(int ii=0; ii<4; ii++)
{DOij_list[ii] = new int[2];}
int LFC_x,LFC_y,Let_ijk=0;
int* count_set = new int[1];
count_set[0] = 4;
coor_offspring_s(count_set,DOij_list,i,j,s_x,s_y,level);
LFC_x = DOij_list[0][0]; LFC_y = DOij_list[0][1];
int type_LFC_label;
type_LFC_label =type_label(LFC_x,LFC_y,s_x,s_y,level);
if (LFC_type[type_LFC_label] <= step_Quantization)
{LFC_D = 1; %%->No LFC D(i,j)}
//O_listo =
coor_offspring(LFC_x,LFC_y,x_pic,y_pic,level);
int **O_listo = new int*[4];
for(ii=0; ii<4; ii++)
{O_listo[ii] = new int[2];}
////////////////////////////////////
coor_offspring(O_listo,LFC_x,LFC_y,s_x,s_y,level);
int LFC_ox,LFC_oy;
if (O_listo[0][0] != -1){LFC_ox = O_listo[0][0];
LFC_oy = O_listo[0][1];
type_LFC_label =
type_label(LFC_ox,LFC_oy,s_x,s_y,level);
if (LFC_type[type_LFC_label] <= step_Quantization)
{ LFC_L = 1; %%->No LFC L(i,j) }}
int check_work = 0; int value_SD,value_SL,Lij_set,yp;
int *type_ij = new int[3];
if ((LFC_D + LFC_L) != 0)
{coor_descendants_s(type_ij,i,j,num_n,fdwt,s_x,s_y,level);
value_SD = type_ij[0];
value_SL = type_ij[1];
Lij_set = type_ij[2];
check_work = 1;}
float absx,d_fdwt; int check_LIP = 0;
if (check_work == 1) %%2.2.1 if the entry is of type A
{if (LIS_type[LIS_length] == 0) %%Output Sn(D(i,j));}
if (bit_end <= count){goto a;}
imout[count] = value_SD;
count = count + 1;
if (value_SD == 1)
{for (int OD_A=0 ; OD_A < count_set[0] ; OD_A++){
%%Output Sn(OD(k,l))
//DATA DOij_list
d_fdwt = fdwt[DOij_list[OD_A][0]][DOij_list[OD_A][1]];
if (d_fdwt < 0){absx = d_fdwt * -1;}
else {absx = d_fdwt;}
if (absx >= num_n)
%%Sn(O(i,j)) is significant
{imout[count] = 1;
count = count + 1;
LSP_x[count_LSP] = DOij_list[OD_A][0];
LSP_y[count_LSP] = DOij_list[OD_A][1];
LSP_abs[count_LSP] = (int)(absx - num_n);
count_LSP = count_LSP+1;
%%Output sing of c(k,l)
if (d_fdwt >= 0) {imout[count] = 1;}
else {imout[count] = 0;}
count = count + 1;
if (bit_end <= count){goto a;}}
else {imout[count] = 0;
count = count + 1;
LIP_x[count_LIP] = DOij_list[OD_A][0];
LIP_y[count_LIP] = DOij_list[OD_A][1];
count_LIP = count_LIP + 1;

```

```

check_LIP = check_LIP + 1;
if (bit_end <= count){goto a;}}
if (Lij_set == 1)
{if (check_LIP == count_set[0])
{for(yp = 0 ; yp < count_set[0] ; yp++)
{LIS_x[count_LIS] = DOij_list[yp][0];
LIS_y[count_LIS] = DOij_list[yp][1];
LIS_type[count_LIS] = 0;
count_LIS = count_LIS + 1;
change = change + 1;} set_add = 1;
//%remove (i,j) from the LIS_B
LIS_x[LIS_length] = -1;
LIS_y[LIS_length] = -1;
LIS_type[LIS_length] = 0; }
else{if (LFC_L == 1)//%Output Sn(L(i,j))
{if (bit_end <= count){goto a;}
imout[count] = value_SL;
count = count + 1;
if (value_SL == 1)
{for(int yp = 0 ; yp < count_set[0] ; yp++)
{ LIS_x[count_LIS] = DOij_list[yp][0];
LIS_y[count_LIS] = DOij_list[yp][1];
LIS_type[count_LIS] = 0;
count_LIS = count_LIS + 1;
change=change + 1;} //%remove (i,j) from the LIS_B
LIS_x[LIS_length] = -1;
LIS_y[LIS_length] = -1;
LIS_type[LIS_length] = 1;
set_add = 1;}
else{LIS_type[LIS_length] = 1;}
if (bit_end <= count){goto a;}}
else{LIS_type[LIS_length] = 1;}}
else{LIS_x[LIS_length] = -1;
LIS_y[LIS_length] = -1;
LIS_type[LIS_length] = 0; }}
//%2.2.2 if the entry is of type B
else{if (LFC_L == 1)//%Output Sn(L(i,j))
{if (bit_end <= count){goto a;}
imout[count] = value_SL;
count = count + 1;

```

```

if (value_SL == 1)
{for(int yp = 0 ; yp < count_set[0] ; yp++)
{ LIS_x[count_LIS] = DOij_list[yp][0];
LIS_y[count_LIS] = DOij_list[yp][1];
LIS_type[count_LIS] = 0;
count_LIS = count_LIS + 1;
change=change + 1;}
//%remove (i,j) from the LIS_B
LIS_x[LIS_length] = -1;
LIS_y[LIS_length] = -1;
LIS_type[LIS_length] = 1;
set_add = 1;}
if (bit_end <= count){goto a;}}}}
//%remove (i,j)
long* count_LIS_new = new long[1];
cut_LIS(count_LIS_new,LIS_x,LIS_y,LIS_type,count_LIS
); count_LIS = count_LIS_new[0];
//%3 Refinement Pass: for each entry (i,j) in the LSP
long* count_new = new long[1];
refinement_pass(count_new,imout,LSP_abs,count_LSP,
(num_n/2),bit_end,count);
count = count_new[0]; num_n = num_n/2;}
a: countp[0] = count;
delete[] LSP_x;
delete[] LSP_y;
delete[] LSP_abs;
delete[] LIP_x;
delete[] LIP_y;
delete[] LIS_x;
delete[] LIS_y;
delete[] LIS_type;}

```

### โปรแกรม Encode\_SPIHT\_function.cpp

```

/*-----*/
// จัดทำโดย: นาย ฐิฐนันท์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี
// ,ก.พ. 2545
/*-----*/
#include"SPIHT.h"

```

```

void coord_descendants_s(int* type_ij,int i,int j,float
threshold,float** fdwt,int s_x,int s_y,int level)
{ /*Input
%i,j = coordinates
%thres = threshold
%Output
%Value_SD = output Sn(D(i,j));
% = 0 -> insignificant D(i,j)
% = 1 -> significant D(i,j)
%value_SL = output Sn(L(i,j));
% = 0 -> insignificant L(i,j)
% = 1 -> significant L(i,j)
%Lij_set = set of L(i,j)
%= 0 -> No membership
%= 1 -> Yes membership
%type_ij
type_ij[0] = value_SD;
type_ij[1] = value_SL;
type_ij[2] = Lij_set; */
//Initial parameter
int Lij_set = 0;
int value_SD = 0;
int value_SL = 0;
int null = 1;
int *countOO = new int[1];
float opp;
/*%X-> i %Y-> j
%limit of first Layer and finit Layer
%first Layer */
double number;
int p,first_x,first_y,finit_x,finit_y;
//limit of first Layer and finit Layer
//first Layer
number = pow((double)2,(double)level);
p = (int)number;
first_x = s_x/(p);
first_y = s_y/(p);
//%finit Layer
finit_x = s_x/2;
finit_y = s_y/2;
//%Group 1 ->DWT Level 1 ,LH1 ,HL1 ,HH1
if ((i>= finit_x && i<s_x && j>=0 && j<s_y) ||
(i>=0 && i<finit_x && j>=finit_y && j<s_y))
{Lij_set = 0;null = 0; }
if (null != 0){
//O_list
int **O_list = new int*[4];
for(int ii=0; ii<4; ii++){O_list[ii] = new int[2];}
int ss_i,ss_j;
//%First Step search
coord_offspring_s(countOO,O_list,i,j,s_x,s_y,level);
//O_list_new
int *O_list_i = new int[350];
int *O_list_j = new int[350];
//display(O_list,4,2);
for (int n=0 ; n < countOO[0] ; n++)
{ //Lij_set = 1;
ss_i = O_list[n][0];
ss_j = O_list[n][1];
//%Sn(D(i,j))
if (fdwt[ss_i][ss_j] < 0)
{opp = fdwt[ss_i][ss_j] * -1;}
else {opp = fdwt[ss_i][ss_j];}
if ( opp >= threshold){ value_SD = 1;}
O_list_i[n] = O_list[n][0];
O_list_j[n] = O_list[n][1];}
int count_oln,count_oln_old,end_count;
count_oln = countOO[0];
count_oln_old = count_oln;
//Next Step search
if (O_list[0][0] != -1){
for (int pso=0 ; pso < count_oln_old ; pso ++){
{ss_i = O_list_i[pso];
ss_j = O_list_j[pso];
coord_offspring(O_list,ss_i,ss_j,s_x,s_y,level);
if (O_list[0][0] != -1){Lij_set = 1; null = 1;
for (n=0 ; n < 4 ; n++){
ss_i = O_list[n][0];
ss_j = O_list[n][1];
O_list_i[count_oln] = ss_i;

```



```

O_list_j[count_oln] = ss_j;
count_oln = count_oln + 1;
//%Sn(L(i,j)
if (fdwt[ss_i][ss_j] < 0)
{opp = fdwt[ss_i][ss_j] * -1;}
else{opp = fdwt[ss_i][ss_j];}
if ( opp >= threshold){
value_SD = 1;
value_SL = 1;
null = 0;goto a;}}}}
int point=2;
if (null == 1){
for (int ttt = 1; ttt <= 100 ; ttt++){
null = 0;
end_count = count_oln;
for (int psp = count_oln_old ; psp < end_count ;psp ++ )
{ ss_i = O_list_i[psp];
ss_j = O_list_j[psp];
coor_offspring(O_list,ss_i,ss_j,s_x,s_y,level);
if (O_list[0][0] != -1){ Lij_set = 1; null = 1;
for (n=0 ; n < 4 ; n++){if (point <= 3)
{ss_i = O_list[n][0];
ss_j = O_list[n][1];
O_list_i[count_oln] = ss_i;
O_list_j[count_oln] = ss_j;
count_oln = count_oln + 1;}
//%Sn(L(i,j)
if (fdwt[ss_i][ss_j] < 0)
{opp = fdwt[ss_i][ss_j] * -1;}
else{opp = fdwt[ss_i][ss_j];}
if ( opp >= threshold){
value_SD = 1;
value_SL = 1;
null = 0;
goto a; }}}
count_oln_old = end_count;
point = point + 1;
if (null != 1){goto a;}
delete[] O_list_i;
delete[] O_list_j; } }
a: type_ij[0] = value_SD;
type_ij[1] = value_SL;
type_ij[2] = Lij_set;}
////////////////////////////////////
void coor_offspring_s(int* countO,int** O_list,int i,int j,int
s_x,int s_y,int level)
{ i = i + 1;
j = j + 1;
double number;
int count_O = 0;
int p,first_x,first_y,finit_x,finit_y;
//limit of first Layer and finit Layer
//first Layer
number = pow((double)2,(double)level);
p = (int)number;
first_x = s_x/(p);
first_y = s_y/(p);
//finit Layer
finit_x = s_x/2;
finit_y = s_y/2;
//Find odd and even
//output = 1 -> odd
//output = 0 -> even
int oi,oj;
//Check i
oi = i%2; oj = j%2;
if (oi == 0)
{oi = 0;}
else{oi = 1;}
//Check j
if (oj == 0)
{oj = 0;}
else{oj = 1;}
int Check = 1;
//No root if ->Yes L(i,j)
//Level END
int px,py;
if (i>=1 && i<=first_x && j>=1 && j<=first_y)
{if ((oi+oj)!=2){ px=first_x-1;
py=first_y-1;

```

```

if (oi==1 && oj==0){
O_list[0][0] = i;
O_list[0][1] = j+py;}
if (oi==0 && oj==1){
O_list[0][0] = i+px;
O_list[0][1] = j; }
if (oi==0 && oj==0)
{O_list[0][0] = i+px;
O_list[0][1] = j+py;}
Check = 0;}
else{ count_O = 0;
O_list[count_O][0] = i+1-1;
O_list[count_O][1] = j-1;
//%1)
O_list[count_O+1][0] = i-1;
O_list[count_O+1][1] = j+1-1;
//%2)
O_list[count_O+2][0] = i+1-1;
O_list[count_O+2][1] = j-1+1;
count_O = 3; Check = 2; } }
//NO root if -> No L(i,j)
//Level 1 (LH1 HL1 HH1)
int a_1,a_2;
a_1 = (finit_x+1); a_2 = (finit_y+1);
if ((i>= a_1 && i<=s_x && j>=1 && j<=s_y) ||
(i>=1 && i<=finit_x && j>= a_2 && j<=s_y) )
{O_list[0][0] = -1; //(-1)
O_list[0][1] = -1; //(-1)
Check = 2;}
//Root of O(i,j) ->Yes L(i,j)
if (Check == 1){
O_list[0][0]=(int)((i*2)-1);
O_list[0][1]=(int)((j*2)-1);}
if (Check !=2){
O_list[0][0]=-1; O_list[0][1]=-1;
//Find root of O(i,j)
O_list[1][0] = O_list[0][0];
O_list[1][1] = O_list[0][1] + 1;
O_list[2][0] = O_list[0][0] + 1;
O_list[2][1] = O_list[0][1];
O_list[3][0] = O_list[0][0] + 1;
O_list[3][1] = O_list[0][1] + 1;
count_O = 4;}
countO[0] = count_O;}
////////////////////////////////////
int find_max_coff(float** fdwt,int s_x,int s_y,int level)
{int first_x,first_y,n,ni; float max_nn,tii;
double number;
//limit of first Layer and finit Layer
//first Layer
number = pow((double)2,(double)level);
ni = (int)number;
first_x = s_x/(ni);
first_y = s_y/(ni);
//max(max(abs(pic)))
float* max_out = new float[4];
dis_max_abs(max_out,fdwt,s_x,s_y);
max_nn = 0;
for (int e = 0 ; e < 4 ; e++){ tii = max_out[e];
if (max_nn < tii){max_nn = tii;}}
n = log2_floor(max_nn);
delete[] max_out;
return(n); }
////////////////////////////////////
void coor_descendants(int* type_ij,int i,int j,float
threshold,float** fdwt,int s_x,int s_y,int level)
{/*Input
%i,j = coordinates
%thres = threshold
%Output
%Value_SD = output Sn(D(i,j));
% = 0 -> insignificant D(i,j)
% = 1 -> significant D(i,j)
%value_SL = output Sn(L(i,j));
% = 0 -> insignificant L(i,j)
% = 1 -> significant L(i,j)
%Lij_set = set of L(i,j)
% = 0 -> No membership
% = 1 -> Yes membership
%type_ij

```

```

type_ij[0] = value_SD;
type_ij[1] = value_SL;
type_ij[2] = Lij_set; */
//Initial parameter
int Lij_set = 0;
int value_SD = 0;
int value_SL = 0;
int null = 1; float opp;
%limit of first Layer and finit Layer
%first Layer */
double number;
int p,first_x,first_y,finit_x,finit_y;
//limit of first Layer and finit Layer
//first Layer
number = pow((double)2,(double)level);
p = (int)number;
first_x = s_x/p; first_y = s_y/p;
//%finit Layer
finit_x = s_x/2; finit_y = s_y/2;
//%Group 1 ->DWT Level 1 ,LH1 ,HL1 ,HH1
if ((i>= finit_x && i<s_x && j>=0 && j<s_y) || (i>=0 &&
i<finit_x && j>=finit_y && j<s_y))
{Lij_set = 0;null = 0;}
if (null != 0){
//O_list
int **O_list = new int*[4];
for(int ii=0; ii<4; ii++)
{O_list[ii] = new int[2];}
int ss_i,ss_j;
//%First Step search
coor_offspring(O_list,i,j,s_x,s_y,level);
for (int n=0 ; n < 4 ; n++){
ss_i = O_list[n][0]; ss_j = O_list[n][1];
//%Sn(D(i,j))
if (fdwt[ss_i][ss_j] < 0)
{opp = fdwt[ss_i][ss_j] * -1;}
else{opp = fdwt[ss_i][ss_j];}
if ( opp >= threshold){ value_SD = 1; }}
//O_list_new
int *O_list_i = new int[350];
int *O_list_j = new int[350];
int count_oln,count_oln_old,end_count;
O_list_i[0] = O_list[0][0];
O_list_j[0] = O_list[0][1];
O_list_i[1] = O_list[1][0];
O_list_j[1] = O_list[1][1];
O_list_i[2] = O_list[2][0];
O_list_j[2] = O_list[2][1];
O_list_i[3] = O_list[3][0];
O_list_j[3] = O_list[3][1];
count_oln = 4;
count_oln_old = count_oln;
//Next Step search
if (O_list[0][0] != -1){ Lij_set = 1;
for (int pso=0 ; pso < 4 ; pso ++){
ss_i = O_list_i[pso]; ss_j = O_list_j[pso];
coor_offspring(O_list,ss_i,ss_j,s_x,s_y,level);
if (O_list[0][0] != -1){null = 1;
for (n=0 ; n < 4 ; n++){
ss_i = O_list[n][0];
ss_j = O_list[n][1];
O_list_i[count_oln] = ss_i;
O_list_j[count_oln] = ss_j;
count_oln = count_oln + 1;
//%Sn(L(i,j))
if (fdwt[ss_i][ss_j] < 0)
{opp = fdwt[ss_i][ss_j] * -1;}
else{opp = fdwt[ss_i][ss_j];}
if ( opp >= threshold){
value_SD = 1; value_SL = 1;
null = 0; goto a;}}}}
int point=2;
if (null == 1){
for (int ttt = 1; ttt <= 100 ; ttt++){
Lij_set = 1; null = 0;
end_count = count_oln;
for (int psp = count_oln_old ; psp < end_count ; psp ++){
ss_i = O_list_i[psp];
ss_j = O_list_j[psp];
coor_offspring(O_list,ss_i,ss_j,s_x,s_y,level);

```

```

if (O_list[0][0] != -1){null = 1;
for (n=0 ; n < 4 ; n++){
if (point <= 3){
ss_i = O_list[n][0];
ss_j = O_list[n][1];
O_list_i[count_oln] = ss_i;
O_list_j[count_oln] = ss_j;
count_oln = count_oln + 1;};}Sn(L(i,j)
if (fdwt[ss_i][ss_j] < 0)
{opp = fdwt[ss_i][ss_j] * -1;}
else{opp = fdwt[ss_i][ss_j];}
if ( opp >= threshold){ value_SD = 1;
value_SL = 1;
null = 0;
goto a; } } }
count_oln_old = end_count;
point = point + 1;
if (null != 1)
{goto a;} }
delete[] O_list_i;
delete[] O_list_j; } }
a: type_ij[0] = value_SD;
type_ij[1] = value_SL;
type_ij[2] = Lij_set; }
////////////////////////////////////
void coor_offspring(int** O_list,int i,int j,int s_x,int s_y,int
level)
{ i = i + 1; j = j + 1;
double number;
int p,first_x,first_y,finit_x,finit_y;
//limit of first Layer and finit Layer
//first Layer
number = pow((double)2,(double)level);
p = (int)number;
first_x = s_x/(p);
first_y = s_y/(p);
//finit Layer
finit_x = s_x/2;
finit_y = s_y/2;
//Find odd and even
//output = 1 -> odd output = 0 -> even
int oi,oj;
//Check i
oi = i%2; oj = j%2;
if (oi == 0){oi = 0;}
else{oi = 1;}
//Check j
if (oj == 0){oj = 0;}
else{oj = 1;}
int Check = 1;
//No root if ->Yes L(i,j)
//Level END
int px,py;
if (i>=1 && i<=first_x && j>=1 && j<=first_y)
{if ((oi+oj)!=2){px=first_x-1; py=first_y-1;
if (oi==1 && oj==0){
O_list[0][0] = i;
O_list[0][1] = j+py; }
if (oi==0 && oj==1){
O_list[0][0] = i+px;
O_list[0][1] = j; }
if (oi==0 && oj==0) {
O_list[0][0] = i+px;
O_list[0][1] = j+py;}
Check = 0;}
else{ O_list[0][0] = -1;
O_list[0][1] = -1; Check = 2; } }
//NO root if -> No L(i,j)
//Level 1 (LH1 HL1 HH1)
int a_1,a_2;
a_1 = (finit_x+1);
a_2 = (finit_y+1);
if ((i>= a_1 && i<=s_x && j>=1 && j<=s_y) ||
(i>=1 && i<=finit_x && j>= a_2 && j<=s_y) )
{O_list[0][0] = -1; //(-1)
O_list[0][1] = -1; //(-1)
Check = 2;}
//Root of O(i,j) ->Yes L(i,j)
if (Check == 1){
O_list[0][0]=(int)((i*2)-1);

```

```

O_list[0][1]=(int)((j*2)-1);}
if (Check !=2){
O_list[0][0]=1; O_list[0][1]=1;
//Find root of O(i,j)
O_list[1][0] = O_list[0][0];
O_list[1][1] = O_list[0][1] + 1;
O_list[2][0] = O_list[0][0] + 1;
O_list[2][1] = O_list[0][1];
O_list[3][0] = O_list[0][0] + 1;
O_list[3][1] = O_list[0][1] + 1;}}
////////////////////////////////////
void refinement_pass(long* count_new,char* imout,int*
LSP_abs,long count_LSP,float num_n,long bit_end,long
count)
{float ans;
for (int t=0 ; t < count_LSP ; t++){
if (bit_end <= count){break;}
ans = (float)LSP_abs[t];
if ((ans >= num_n) && (ans != 0)){
imout[count] = 1;
LSP_abs[t] = (int)(ans - num_n);}
else{imout[count] = 0;}
count = count + 1;
if (bit_end <= count){break;}}
count_new[0] = count;}
////////////////////////////////////
void cut_LIS(long* count_LIS_new,int* LIS_x,int*
LIS_y,int* LIS_type,long count_LIS)
{ long count=0;
for(long z=0 ; z < count_LIS ; z++){
if(LIS_x[z]!=-1){count = count + 1;}}
int* x_new = new int[count];
int* y_new = new int[count];
int* z_new = new int[count];
count = 0;
for( z=0 ; z < count_LIS ; z++)
{if(LIS_x[z]!=-1)
{x_new[count] = LIS_x[z]; y_new[count] = LIS_y[z];
z_new[count] = LIS_type[z];
LIS_x[z] = -1;
count = count + 1;}}
for( z=0 ; z < count ; z++){
LIS_x[z] = x_new[z];
LIS_y[z] = y_new[z];
count_LIS_new[0] = count;
delete[] x_new;
delete[] y_new;}
////////////////////////////////////
void sn_out(float* output,int* LIP_x,int* LIP_y,int
t,float** fdwt,float num_n)
{ /* output[0] = significant -> 1
No significant -> 0
output[1] = abs(data(i,j)) - num_n
output[2] = sign -> 1 -> (+)
-> 0 -> (-) */
//%coordinates Sn(i,j)
int i,j;

```

```

i = LIP_x[t];
j = LIP_y[t];
//%Output
float data; data = fdwt[i][j];
//Check sign and abs(data)
if (data < 0){output[2] = 0; // (-)
data = data * -1;}
else{output[2] = 1; // (+)}
//Check significant pixel and save data
if (data >= num_n)
{output[0] = 1;//significant}
else{output[0] = 0; //no significant}
data = data - num_n;
output[1] = data; // abs(data) - num_n}

```

### โปรแกรม LFC.cpp

```

/*-----*/
// จัดทำโดย: นาย ฉัฐนันท์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี
// ,ก.พ. 2545
/*-----*/
#include"SPIHT.h"
////////////////////////////////////
//LFC list
////////////////////////////////////
int type_label(int x,int y,int s_x,int s_y,int level)
{ //X-> x //Y-> y
x = x + 1; y = y + 1;
//%limit of first Layer and finit Layer
//%first Layer
double number; int p,first_x,first_y;
//limit of first Layer and finit Layer
//first Layer
number = pow((double)2,(double)level);
p = (int)number;
first_x = s_x/(p);
first_y = s_y/(p);
int tx,ty,label;
//%----- level - 1
//%(LL)1-4

```

```

if ((x<=first_x) && (y<=first_y)){
tx = first_x/2;
ty = first_y/2;
if ((x<=tx) && (y<=ty)){label = 1;}
if ((x<=tx) && (y>ty)){label = 2;}
if ((x>tx) && (y<=ty)){label = 3;}
if ((x>tx) && (y>ty)){label = 4;}}
int ll,a,c,delta;
for (int l=0 ; l <= (level-1) ; l++)
{delta = 12*l; //ll = (2^l);
number = pow((double)2,(double)l);
ll= (int)number; a = (first_x*ll); c = (first_x*2*ll);
//%----- layer
//%(HL)
if ((x<=a) && (y>a && y<=c))
{tx = a/2; ty = (a+c)/2;
if ((x<=tx) && (y<=ty))
{label = 5+delta; break; }
if ((x<=tx) && (y>ty))
{label = 6+delta; break; }
if ((x>tx) && (y<=ty))
{label = 7+delta; break; }
if ((x>tx) && (y>ty))
{label = 8+delta; break; } }
//%(LH)
if ((x>a && x<=c) && (y<=a))
{tx = (a+c)/2; ty = a/2;
if ((x<=tx) && (y<=ty))
{ label = 9+delta; break; }
if ((x<=tx) && (y>ty))
{label = 10+delta; break;
if ((x>tx) && (y<=ty))
{label = 11+delta; break; }
if ((x>tx) && (y>ty))
{label = 12+delta; break; }
//%(HH)
if ((x>a && x<=c) && (y>a && y<=c))
{tx = (a+c)/2; ty = (a+c)/2;
if ((x<=tx) && (y<=ty))
{label = 13+delta; break; }

```

```

if ((x<=tx) && (y>ty))
{ label = 14+delta; break; }
if ((x>tx) && (y<=ty))
{label = 15+delta; break;}
if ((x>tx) && (y>ty))
{label = 16+delta;break;}}
label = label - 1; return(label);}
////////////////////////////////////
void LFC_list(int* q_LFC,float** image_in,int s_x,int
s_y,int level)
{%%limit of first Layer and finit Layer
%%first Layer
double number;
int p,first_x,first_y;
//limit of first Layer and finit Layer
//first Layer
number = pow((double)2,(double)level);
p = (int)number;
first_x = s_x/(p); first_y = s_y/(p);
//size of LFC list
int count_LFC,x1,x2,y1,y2;
count_LFC = (level*4*3) + 4;
float* LFC = new float[count_LFC];
count_LFC = 0;
%%LL
//c11 = image_in(1:first_x,1:first_y);
float** output = new float*[s_x/2];
for(int ii=0; ii<(s_x/2); ii++)
{output[ii] = new float[s_y/2];}
x1 = 0; y1 = 0;
x2 = first_x - 1;
y2 = first_y - 1;
cut_image(output,image_in,0,x2,0,y2);
float* max_out = new float[4];
dis_max_abs(max_out,output,first_x,first_y);
for (int n=0 ; n < 4 ; n++)
{LFC[count_LFC] = max_out[n];
count_LFC = count_LFC + 1; }
int l,a,b,c;
%%Layer 1-level
for (int degree=0; degree <= (level-1) ; degree++)
{number = pow((double)2,(double)degree);
l = (int)number;
a = (first_x*1) - 1;
b = (first_x*1+1) - 1;
c = (first_x*2*1) - 1;
%%HL
cut_image(output,image_in,0,a,b,c);
x1 = a + 1;
y1 = c - b + 1;
dis_max_abs(max_out,output,x1,y1);
for (n=0 ; n < 4 ; n++)
{LFC[count_LFC] = max_out[n];
count_LFC = count_LFC + 1; }
%%LH
cut_image(output,image_in,b,c,0,a);
x1 = c - b + 1; y1 = a;
dis_max_abs(max_out,output,x1,y1);
for (n=0 ; n < 4 ; n++)
{LFC[count_LFC] = max_out[n];
count_LFC = count_LFC + 1; }
%%HH
cut_image(output,image_in,b,c,b,c);
x1 = c - b + 1;
y1 = x1;
dis_max_abs(max_out,output,x1,y1);
for (n=0 ; n < 4 ; n++)
{LFC[count_LFC] = max_out[n];
count_LFC = count_LFC + 1; }
int type_ii,max_n; float tii,max_nn; max_nn = 0;
for (int e = 0 ; e < 4 ; e++){tii = LFC[e];
if (max_nn < tii){max_nn = tii;} }
max_n = log2_floor(max_nn);
for (e = 0 ; e < count_LFC ; e++){ tii = LFC[e];
type_ii = floor_LFC_n(tii,max_n);
q_LFC[e] = type_ii; }
////////////////////////////////////
int floor_LFC_n(float max_out_n,int max_n)
{ int out,output;
out = log2_floor(max_out_n);

```

```

output = max_n - out + 1 ;
return(output);}
////////////////////////////////////
int log2_floor(float data)
{double x,y; int out;
x = (double)data; y = log(x);
x = log(2.0); out = (int)(y/x); return(out);}
////////////////////////////////////
void cut_image(float** output,float** input,int x_start,int
x_end,int y_start,int y_end)
{ int s_i_x,s_i_y; s_i_x = 0;
for (int index_row=x_start;index_row<=x_end;
Index_row++) {s_i_y = 0;
for (int index_col=y_start;index_col<=y_end;index_col++)
{output[s_i_x][s_i_y] = input[index_row][index_col];
s_i_y = s_i_y + 1; }
s_i_x = s_i_x + 1; }
////////////////////////////////////
void dis_max_abs(float* max_out,float** input,int ss_x,int
ss_y)
{ int x,y,il,jl;
float l1,l2,l3,l4,s_data;
l1 = 0; l2 = 0; l3 = 0; l4 = 0;
//1
x = ss_x/2; y = ss_y/2;
for(il = 0; il < x; il++) { for(jl = 0; jl < y; jl++){
if (input[il][jl] < 0){s_data = input[il][jl] * -1;}
else{s_data = input[il][jl];}
if ( l1 < s_data)
{l1 = s_data; } }
//2
for(il = 0; il < x; il++){ for(jl = y; jl < ss_y; jl++){
if (input[il][jl] < 0){ s_data = input[il][jl] * -1;}
else{s_data = input[il][jl];}
if ( l2 < s_data){l2 = s_data; } } }
//3
for(il = x; il < ss_x; il++){ for(jl = 0; jl < y; jl++){
if (input[il][jl] < 0){s_data = input[il][jl] * -1;}
else{s_data = input[il][jl];}
if ( l3 < s_data){l3 = s_data; } } }
//4
for(il = x; il < ss_x; il++){ for(jl = y; jl < ss_y; jl++)
{ if (input[il][jl] < 0)
{s_data = input[il][jl] * -1;}
else{s_data = input[il][jl];}
if ( l4 < s_data){l4 = s_data; } } }
max_out[0] = l1;
max_out[1] = l2;
max_out[2] = l3;
max_out[3] = l4;}
////////////////////////////////////
void round_image(float** input,int s_x,int s_y)
{int store_int;
for(int i = 0 ; i < s_x ; i++){for(int j = 0 ; j < s_y ; j++){
if (input[i][j] >= 0)
{store_int = (int)(input[i][j]+0.5);
input[i][j] = (float)store_int;}
else{store_int = (int)(input[i][j]-0.5);
input[i][j] = (float)store_int;}}} }

```



## ภาคผนวก ง

รายละเอียดโปรแกรมการถอดรหัสอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง

### โปรแกรม SPIHT.h

```

/*-----*/
// จัดทำโดย: นาย ฐิฐนันท์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี
// ,ก.พ. 2545
/*-----*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include"constant.h"
#include<iostream.h>
#define max_nr -1 //Check No root of o(i,j)
//DWT and IDWT (1D and 2D) -> bi9-7
void conv(float* out,float* hn,float* xn,int size_h,int
size_x);
void conv_p(float* hn,float* xn,int size_x,int size_h);
//DWT
void refind_data_bi9(float* out,float* in,int size_x);
void refind_data_bi7(float* out,float* in,int size_x);
void bi_dwt(float* in,int size_x);
void down_sampling(float* in,int st,int nxt,int type);
void bi_dwt2(float** fdwt,int s_x,int s_y);
//IDWT
void bi_idwt(float* in,int size_x);
void up_sampling(float* out,float* in,int size_x,int type);
void bi_idwt2(float** fdwt,int s_x,int s_y);
//Encode SPIHT
void coor_offspring(int** O_list,int i,int j,int s_x,int s_y,int
level);
void coor_descendants(int* in,int i,int j,float threshold,float**
fdwt,int s_x,int s_y,int level);
//LFC SPIHT
void LFC_list(int* q_LFC,float** image_in,int s_x,int
s_y,int level);
void cut_image(float** output,float** input,int x_start,int
x_end,int y_start,int y_end);
void dis_max_abs(float* max_out,float** input,int ss_x,int
ss_y);
int type_label(int x,int y,int s_x,int s_y,int level);
int floor_LFC_n(float max_out_n,int max_n);

```

```

int log2_floor(float data);
void sn_out(float* output,int* LIP_x,int* LIP_y,int
t,float** fdwt,float num_n);
void cut_LIP(long* count_LIP_new,int* LIP_x,int*
LIP_y,long count_LIP);
void cut_LIS(long* count_LIS_new,int* LIS_x,int*
LIS_y,int* LIS_type,long count_LIS);
void refinement_pass(long* count_new,char* imout,int*
LSP_abs,long count_LSP,float num_n,long bit_end,long
count);
int find_max_coff(float** fdwt,int s_x,int s_y,int level);
//DECODE SPIHT
void SPIHT_Decode(char* LFC_list_type,float**
output_image,char* imout,int bit_end,int s_x,int s_y,int
level,int n);
void i_coor_descendants(int** O_list,int* type_ij,int i,int
j,int s_x,int s_y,int level);
void i_refinement_pass(long* count_new,char* imout,int*
LSP_abs,long count_LSP,float num_n,long bit_end,long
count);
void re_image(int** output_image,int* LSP_x,int*
LSP_y,int* LSP_abs,int* LSP_sign,long count_LSP);
void coor_descendants_s(int* type_ij,int i,int j,float
threshold,float** fdwt,int s_x,int s_y,int level);
void coor_offspring_s(int* countO,int** O_list,int i,int j,int
s_x,int s_y,int level);
void i_coor_descendants_s(int* count_set,int** O_list,int*
type_ij,int i,int j,int s_x,int s_y,int level);
void round_image_pass(float** input,int s_x,int s_y,float
bitrate);
void round_image_abs(float** input,int s_x,int s_y);

```

### โปรแกรม coff.h

เหมือนในภาคผนวก ก.

### โปรแกรม main.cpp

```

/*-----*/
// จัดทำโดย: นาย ฐิฐนันท์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี
// ,ก.พ. 2545

```

```

/*-----*/
#include"SPIHT.h"
#include"coeff.h"
main(){
////////////////////////////////////
//Load file Header SPIHT
int bit_end,in1[1]; //Size Data
int s_x,in2[1]; //->Dimension X
int s_y,in3[1]; //->Dimension Y
int level,in4[1]; //-> level wavelet tranfrom
int n,in5[1]; //-> Initial n max(coff)
////////////////////////////////////
FILE *pheader;
pheader = fopen("header.sim","rb");
fread(in1,sizeof(int),1,pheader); //size of data wavelet
SPIHT
fread(in2,sizeof(int),1,pheader); //hight image
fread(in3,sizeof(int),1,pheader); //wide image
fread(in4,sizeof(int),1,pheader); //level decomposition
fread(in5,sizeof(int),1,pheader); //number step Quantization
bit_end = in1[0];
s_x = in2[0];
s_y = in3[0];
level = in4[0];
n = in5[0];
//LFC List
int count_LFC; count_LFC = (level*4*3) + 4;
char* LFC_list_type = new char[count_LFC];
fread(LFC_list_type,sizeof(char),count_LFC,pheader);
fclose(pheader);
////////////////////////////////////
//Load file Data SPIHT
char* imout = new char[bit_end];
FILE *pimout_read;
// Open file in SUT mode:
pimout_read = fopen("output.sut","rb");
//generate size pointer <Dynamic >
fread(imout,sizeof(char),bit_end,pimout_read);
fclose(pimout_read);
////////////////////////////////////
// Set r_x,r_y (size Data input)
////////////////////////////////////
long int count_r = 0; int i,r_i,r_j;
//generate size pointer <Dynamic >
float **fdwt = new float*[s_x];
for( i=0; i<s_x; i++)
{fdwt[i] = new float[s_y];}
for (r_i = 0; r_i < s_y ; r_i++){
for (r_j = 0; r_j < s_x ; r_j++){
{fdwt[r_j][r_i] = (float)0; //data out -> double
count_r = count_r + 1;}}
////////////////////////////////////
/* SPIHT -> Decode
////////////////////////////////////*/
//Decode_SPIHT + LFC
SPIHT_Decode(LFC_list_type,fdwt,imout,bit_end,s_x,s_y,
level,n);
////////////////////////////////////
//// IDWT
bi_idwt2(fdwt,r_xx/16,r_yy/16);
bi_idwt2(fdwt,r_xx/8,r_yy/8);
bi_idwt2(fdwt,r_xx/4,r_yy/4);
bi_idwt2(fdwt,r_xx/2,r_yy/2);
bi_idwt2(fdwt,r_xx,r_yy);
printf("\nbit rate = %0.5f bpp -> bit compress = %d bit\n",
(float)bit_end/(s_x*s_y),bit_end);
round_image_abs(fdwt,r_xx,r_yy);
////////////////////////////////////
// Transfer Data from VC++ to MATLAB // Set
w_x,w_y in constant.h (size Data input)
////////////////////////////////////
FILE *p_write;
long int numwritten;
long int count_w = 0;
//generate size pointer <Dynamic >
unsigned char *w_list = new unsigned char[r_xx*r_yy]; //
Open file in text mode:
p_write = fopen("out_image.dat", "wb" );
for ( int w_i = 0; w_i < s_y; w_i++ ){
for (int w_j = 0; w_j < s_x ; w_j++){

```

```

w_list[count_w] = (unsigned char)(fdwt[w_j][w_i]);
count_w = count_w + 1; } }
numwritten = fwrite(w_list, sizeof(char),s_x*s_y, p_write
);
fclose( p_write );
////////////////////////////////////
//clear Memory pointer
delete[] w_list;
return(0);}

```

### โปรแกรม bi\_dwt.cpp

เหมือนภาคผนวก ก.

### โปรแกรม Decode\_SPIHT\_main.cpp

```

/*-----*/
// จัดทำโดย: นาย ฉัฐนันท์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี
// ,ก.พ. 2545
/*-----*/
#include"SPIHT.h"
void SPIHT_Decode(char* LFC_list_type,float**
output_image,char* imout,int bit_end,int s_x,int s_y,int
level,int n){
//Initial DATA i_SPIHT = output_image (size = s_x*s_y)
int i,j,oi,oj;
for (i=0;i<s_x;i++){for(j=0;j<s_y;j++)
{ output_image[i][j] = 0; } }
//%1Initialization :output n = [log2(max(i,j){ci,j})]
int nl,first_x,first_y; float num_n;
double number;
//limit of first Layer and finit Layer
//first Layer
number = pow((double)2,(double)level);
nl = (int)number;
first_x = s_x/(nl);
first_y = s_y/(nl);
number = pow((double)2,(double)n);
num_n = (float)number;
long count,count_LSP,count_LIP,count_LIS;
count = 0;

```

```

count_LSP = 0;
count_LIP = 0;
count_LIS = 0;
//%Initial LSP size=[s_x*s_y 3]
int* LSP_x = new int[long(s_x*s_y/2)];
int* LSP_y = new int[long(s_x*s_y/2)];
int* LSP_abs = new int[long(s_x*s_y/2)];
//%Initial LIP size=[s_x*s_y 2]
int* LIP_x = new int[s_x*s_y/2];
int* LIP_y = new int[s_x*s_y/2];
for (i=0 ; i < first_x ; i++)
{for (j=0 ; j < first_y ; j++)
{ LIP_x[count_LIP] = i;
LIP_y[count_LIP] = j;
count_LIP = count_LIP + 1; } }
//%Initial LIS size=[s_x*s_y 3]
int* LIS_x = new int[s_x*s_y/2];
int* LIS_y = new int[s_x*s_y/2];
int* LIS_type = new int[s_x*s_y];
for ( i=0 ; i < first_x ; i++)
{ for (j=0 ; j < first_y ; j++){
//Check odd and even
oi = i%2;
oj = j%2;
//Check i
if (oi == 0){ oi = 0; }
else{ oi = 1; }
//Check j
if (oj == 0){ oj = 0; }
else{ oj = 1; }
if ( (oi+oj)!= 0 ) {
LIS_x[count_LIS] = i;
LIS_y[count_LIS] = j;
LIS_type[count_LIS] = 0; //%->set type A
count_LIS = count_LIS +1; } } }
//%1.2 Make it LFC list , size of LFC list
int count_LFC; count_LFC = (level*4*3) + 4;
int* LFC_type = new int[count_LFC];
for(int e=0; e <count_LFC ;e++)
{ LFC_type[e] = (int)LFC_list_type[e]; }

```

```

for (int step_Quantization = 1; step_Quantization <=
step_quan ; step_Quantization++) {
    %%2)Sorting pass:
    %%2.1 for each entry (i,j) in the LIP
    for (int t=0; t < count_LIP; t++)
    %%2.1.1 output Sn(i,j);
    {if (bit_end <= count){break;}
    int output; output = imout[count]; count = count + 1;
    %%2.1.2) Sn(i,j) = 1 then move (i,j) to the LSP and output
    the sign of cij
    if (output == 1) {
        LSP_x[count_LSP] = LIP_x[t];
        LSP_y[count_LSP] = LIP_y[t];
        LSP_abs[count_LSP] = (int)num_n;
        %%Output the sign of cij
        output = imout[count];
        count = count + 1;
        if (output == -1) {
            LSP_abs[count_LSP] = -LSP_abs[count_LSP];}
        %%END of Output the sign of cij
        count_LSP = count_LSP+1;
        LIP_x[t] = -1;
        LIP_y[t] = -1;} %%End of the Sn(i,j)=1
    if (bit_end <= count){break} %%END of is Sn(i,j) = then
    move (i,j) to the LSP
    %%2.1.2) Move LIP(i,j) at sn(i,j)=1
    long* count_LIP_new = new long[1];
    cut_LIP(count_LIP_new,LIP_x,LIP_y,count_LIP);
    count_LIP = count_LIP_new[0];
    %% END of Loop for each entry (i,j) in the LIP
    %%2.2 for each entry (i,j) in the LIS
    %%Initial parameter 2.2
    long count_LIS_old;
    count_LIS_old = count_LIS;
    int change,set_add;
    change = 100;
    set_add = 0;
    for (long LIS_length = 0 ; LIS_length < 100000000 ;
    LIS_length++){
        if ((change == 0) && (count_LIS_old <= LIS_length))
            {break;}
        if (LIS_length==0){change = 0;}
        if ((count_LIS_old <= LIS_length)&& set_add==1)
            {change = change - 1;}
        %%Find Do(i,j) , LO(i,j)
        i = LIS_x[LIS_length];
        j = LIS_y[LIS_length];
        %%Check LFC list
        int LFC_D = 0; %%LFC O(i,j)
        int LFC_L = 0; %%LFC L(i,j)
        %%i_coor_descendants_gen
        int value_SD,value_SL,Lij_set;
        int *type_ij = new int[1]; %%O_list
        int **DOij_list = new int*[4];
        for(int ii=0; ii<4; ii++){ DOij_list[ii] = new int[2]; }
        int LFC_x,LFC_y,Let_ijk=0;
        coor_offspring(DOij_list,i,j,s_x,s_y,level);
        LFC_x = DOij_list[0][0];
        LFC_y = DOij_list[0][1];
        int type_LFC_label;
        type_LFC_label = type_label(LFC_x,LFC_y,s_x,
        s_y,level);
        if (LFC_type[type_LFC_label] <= step_Quantization)
            {LFC_D = 1; %%->No LFC D(i,j)}
        int **O_listo = new int*[4];
        for(ii=0; ii<4; ii++)
            {O_listo[ii] = new int[2];} coor_offspring
            (O_listo,LFC_x,LFC_y,s_x,s_y,level);
        int LFC_ox,LFC_oy;
        if (O_listo[0][0] != -1){Let_ijk = 1;
        LFC_ox = O_listo[0][0];
        LFC_oy = O_listo[0][1];
        type_LFC_label = type_label(LFC_ox,LFC_oy,s_x,
        s_y,level);
        if (LFC_type[type_LFC_label] <= step_Quantization)
            { LFC_L = 1; %%->No LFC L(i,j) }}
        int check_work = 0; int* count_set = new int[1];
        if ((LFC_D + LFC_L) != 0){

```

```

i_coor_descendants_s(count_set,DOij_list,type_ij,i,j,s_x,s_
y,level);
Lij_set = type_ij[0];
check_work = 1; }
int absx,d_fdwt,yp;
int check_LIP = 0;
if (check_work == 1){
  %%2.2.1 if the entry is of type A
  if (LIS_type[LIS_length] == 0)%%Output Sn(D(i,j));
  {if (bit_end <= count){goto a;}
  value_SD = imout[count];
  count = count + 1;
  if (value_SD == 1){
    for (int OD_A=0 ; OD_A < count_set[0] ; OD_A++){
      %%Output Sn(OD(k,l)){absx = imout[count];
      count = count + 1;
      if (absx == 1)%%Sn(O(i,j)) is significant
      {LSP_x[count_LSP] = DOij_list[OD_A][0];
      LSP_y[count_LSP] = DOij_list[OD_A][1];
      LSP_abs[count_LSP] = (int)num_n;
      %%Output sing of c(k,l)
      d_fdwt = imout[count];
      count = count + 1;
      if (d_fdwt == -1)
      {LSP_abs[count_LSP] = -LSP_abs[count_LSP];}
      count_LSP = count_LSP+1;
      if (bit_end <= count){goto a;}}
    else %%Sn(O(i,j)) is insignificant
    {LIP_x[count_LIP] = DOij_list[OD_A][0];
    LIP_y[count_LIP] = DOij_list[OD_A][1];
    count_LIP = count_LIP + 1;
    check_LIP = check_LIP + 1;
    if (bit_end <= count){goto a; } }
    if (Let_ijk == 1)%%remove (i,j) from the LIS (type A)
    {if (check_LIP == count_set[0]){
      %%set type A
      for(yp = 0 ; yp < count_set[0] ; yp++){
        LIS_x[count_LIS] = DOij_list[yp][0];
        LIS_y[count_LIS] = DOij_list[yp][1];
        LIS_type[count_LIS] = 0;
        count_LIS = count_LIS + 1;
        change = change + 1;}
      %%remove (i,j) from the LIS_B
      LIS_x[LIS_length] = -1;
      LIS_y[LIS_length] = -1;
      LIS_type[LIS_length] = 1;
      set_add = 1; }
      else{LIS_type[LIS_length] = 1;}
      if (bit_end <= count){goto a;}
      else{LIS_type[LIS_length] = 1; } }
      else{%%remove (i,j) from the LIS_B
      LIS_x[LIS_length] = -1;
      LIS_y[LIS_length] = -1;
      LIS_type[LIS_length] = 1; } }
      }%%END of entry is of type A
      else { if (LFC_L == 1) { %%Output Sn(L(i,j))
      if (bit_end <= count)
      {goto a;}
      value_SL = imout[count];
      count = count + 1;
      if (value_SL == 1) {
        for(yp = 0 ; yp < count_set[0] ; yp++) {
          LIS_x[count_LIS] = DOij_list[yp][0];

```

```

LIS_y[count_LIS] = DOij_list[yp][1];
LIS_type[count_LIS] = 0;
count_LIS = count_LIS + 1;
change = change + 1; }
set_add = 1;
%%remove (i,j) from the LIS_B
LIS_x[LIS_length] = -1;
LIS_y[LIS_length] = -1;
LIS_type[LIS_length] = 0;
if (bit_end <= count){goto a; }
} } } //END of LOOP LIS
long* count_LIS_new = new long[1];
cut_LIS(count_LIS_new,LIS_x,LIS_y,LIS_type,count_LIS
);
count_LIS = count_LIS_new[0];
%%3 Refinement Pass: for each entry (i,j) in the LSP
long* count_new = new long[1];
i_refinement_pass(count_new,imout,LSP_abs,count_LSP,
(num_n/2),bit_end,count);
count = count_new[0];
%%4 Quantization
num_n = num_n/2;}
a:
re_image_pass(output_image,LSP_x,LSP_y,LSP_abs,coun
t_LSP);
delete[] LSP_x;
delete[] LSP_y;
delete[] LSP_abs;
delete[] LIP_x;
delete[] LIP_y;
delete[] LIS_x;
delete[] LIS_y;
delete[] LIS_type; }

โปรแกรม Decode_SPIHT_function.cpp
/*-----*/
// จัดทำโดย: นาย ฐันนันทน์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี
// ,ก.พ. 2545
/*-----*/

#include"SPIHT.h"
////////////////////////////////////
//SPIHT-Decode-function
////////////////////////////////////
void i_coor_descendants_s(int *count_set,int** O_list,int*
type_ij,int i,int j,int s_x,int s_y,int level)
{ /*Input <i,j = coordinates>
%Output <Lij_set = set of L(i,j)>
% = 0 -> No membership
% = 1 -> Yes membership
%type_ij -> type_ij[0] = Lij_set; */
//Initial parameter
int Lij_set = 1;
int null = 1;
%limit of first Layer and finit Layer
%first Layer */
double number;
int p,first_x,first_y,finit_x,finit_y;
//limit of first Layer and finit Layer
//first Layer
number = pow((double)2,(double)level);
p = (int)number;
first_x = s_x/p;
first_y = s_y/p;
//%finit Layer
finit_x = s_x/2; finit_y = s_y/2;
//%Group 1 ->DWT Level 1 ,LH1 ,HL1 ,HH1
if ((i>= finit_x/2 && i<s_x && j>=0 && j<s_y) || (i>=0
&& i<finit_x/2 && j>=finit_y/2 && j<s_y))
{ Lij_set = 0; } //O_list
coor_offspring_s(count_set,O_list,i,j,s_x,s_y,level);
type_ij[0] = Lij_set; }
////////////////////////////////////
void i_coor_descendants(int** O_list,int* type_ij,int i,int
j,int s_x,int s_y,int level)
{ /*Input
%i,j = coordinates
%Output
%Lij_set = set of L(i,j)
% = 0 -> No membership

```

```

%           = 1 -> Yes membership
%type_ij
type_ij[0] = Lij_set; */
//Initial parameter
int Lij_set = 1;
int null = 1;
%limit of first Layer and finit Layer
%first Layer */
double number; int p,first_x,first_y,finit_x,finit_y;
//limit of first Layer and finit Layer
//first Layer
number = pow((double)2,(double)level);
p = (int)number;
first_x = s_x/(p);
first_y = s_y/(p);
//%finit Layer
finit_x = s_x/2;
finit_y = s_y/2;
//%Group 1 ->DWT Level 1 ,LH1 ,HL1 ,HH1
if ((i>= finit_x && i<s_x && j>=0 && j<s_y) || (i>=0 &&
i<finit_x && j>=finit_y && j<s_y))
{Lij_set = 0; null = 0;}
if (null != 0){coor_offspring(O_list,i,j,s_x,s_y,level);}
type_ij[0] = Lij_set; }
////////////////////////////////////
void i_refinement_pass(long* count_new,char* imout,int*
LSP_abs,long count_LSP,float num_n,long bit_end,long
count){int ans;
for (int t=0 ; t < count_LSP ; t++) {
if (bit_end <= count)
{break;}
ans = imout[count];
count = count + 1;
if (LSP_abs[t] >= 0) {
LSP_abs[t] = (int)(LSP_abs[t] + num_n*ans);}
else{LSP_abs[t] = (int)(LSP_abs[t] - num_n*ans);}
if (bit_end <= count){break;}}
count_new[0] = count;
////////////////////////////////////
void re_image(int** output_image,int* LSP_x,int*
LSP_y,int * LSP_abs,int* LSP_sign,long count_LSP)
{int i,j;
float s_amp;
for(long step = 0; step < count_LSP ; step ++ ) {
i = LSP_x[step];
j = LSP_y[step];
s_amp = LSP_abs[step] * (float)LSP_sign[step];
output_image[i][j] = (int)s_amp; }
////////////////////////////////////
void re_image_pass(float** output_image,int* LSP_x,int*
LSP_y,int * LSP_abs,long count_LSP)
{ int i,j;
for(long step = 0; step < count_LSP ; step ++ ) {
i = LSP_x[step]; j = LSP_y[step];
output_image[i][j] = (float)LSP_abs[step]; } }
////////////////////////////////////
void coor_offspring_s(int* countO,int** O_list,int i,int j,int
s_x,int s_y,int level)
{ i = i + 1; j = j + 1; double number; int count_O = 0;
int p,first_x,first_y,finit_x,finit_y,oi,oj;
//limit of first Layer and finit Layer
//first Layer
number = pow((double)2,(double)level);
p = (int)number;
first_x = s_x/(p);
first_y = s_y/(p);
//finit Layer
finit_x = s_x/2;
finit_y = s_y/2;
//Find odd and even output = 1 -> odd
//output = 0 -> even
//Check i
oi = i%2; oj = j%2;
if (oi == 0){oi = 0;}
else{oi = 1;}
//Check j
if (oj == 0){oj = 0;}
else{oj = 1;}
int Check = 1;

```



```

//No root if ->Yes L(i,j)
//Level END
int px,py;
if (i>=1 && i<=first_x && j>=1 && j<=first_y) {
if ((oi+oj)!=2) {
px=first_x-1;
py=first_y-1;
if (oi==1 && oj==0)
{ O_list[0][0] = i;
O_list[0][1] = j+py; }
if (oi==0 && oj==1)
{ O_list[0][0] = i+px;
O_list[0][1] = j; }
if (oi==0 && oj==0)
{ O_list[0][0] = i+px;
O_list[0][1] = j+py; }
Check = 0; }
else { count_O = 0;
O_list[count_O][0] = i+1-1;
O_list[count_O][1] = j-1;
//%1)
O_list[count_O+1][0] = i-1;
O_list[count_O+1][1] = j+1-1;
//%2)
O_list[count_O+2][0] = i+1-1;
O_list[count_O+2][1] = j-1+1;
count_O = 3;
Check = 2; } }
//NO root if -> No L(i,j)
//Level 1 (LH1 HL1 HH1)
int a_1,a_2;
a_1 = (finit_x+1);
a_2 = (finit_y+1);
if ((i>= a_1 && i<=s_x && j>=1 && j<=s_y) || (i>=1 &&
i<=finit_x && j>= a_2 && j<=s_y) ) {
O_list[0][0] = -1; //(-1)
O_list[0][1] = -1; //(-1)
Check = 2 }
//Root of O(i,j) ->Yes L(i,j)
if (Check == 1) {
O_list[0][0]=(int)((i*2)-1);
O_list[0][1]=(int)((j*2)-1);}
if (Check !=2){
O_list[0][0]=-1;
O_list[0][1]=-1;
//Find root of O(i,j)
//2 3 ;*O(i,j)
//1)
O_list[1][0] = O_list[0][0];
O_list[1][1] = O_list[0][1] + 1;
//2)
O_list[2][0] = O_list[0][0] + 1;
O_list[2][1] = O_list[0][1];
//3)
O_list[3][0] = O_list[0][0] + 1;
O_list[3][1] = O_list[0][1] + 1;
count_O = 4; }
countO[0] = count_O; }
////////////////////////////////////
void coor_offspring(int** O_list,int i,int j,int s_x,int s_y,int
level)
{ i = i + 1; j = j + 1;
double number; int p,first_x,first_y,finit_x,finit_y;
//limit of first Layer and finit Layer
//first Layer
number = pow((double)2,(double)level);
p = (int)number;
first_x = s_x/(p);
first_y = s_y/(p);
//finit Layer
finit_x = s_x/2; finit_y = s_y/2;
//Find odd and even
//output = 1 -> odd output = 0 -> even
int oi,oj;
//Check i
oi = i%2; oj = j%2;
if (oi == 0){oi = 0;}
else{oi = 1;}
//Check j
if (oj == 0){oj = 0;}

```

```

else{oj = 1;}
int Check = 1;
//No root if ->Yes L(i,j)
//Level END
int px,py;
if (i>=1 && i<=first_x && j>=1 && j<=first_y)
{ if ((oi+oj)!=2) {
px=first_x-1;
py=first_y-1;
if (oi==1 && oj==0)
{O_list[0][0] = i;
O_list[0][1] = j+py; }
if (oi==0 && oj==1)
{O_list[0][0] = i+px;
O_list[0][1] = j; }
if (oi==0 && oj==0)
{ O_list[0][0] = i+px;
O_list[0][1] = j+py; }
Check = 0; }
else { O_list[0][0] = -1;
O_list[0][1] = -1
Check = 2; } }
//NO root if -> No L(i,j)
//Level 1 (LH1 HL1 HH1)
int a_1,a_2;
a_1 = (finit_x+1); a_2 = (finit_y+1);
if ((i>= a_1 && i<=s_x && j>=1 && j<=s_y) || (i>=1 &&
i<=finit_x && j>= a_2 && j<=s_y) )
{ O_list[0][0] = -1; //(-1)
O_list[0][1] = -1; //(-1)
Check = 2; }
//Root of O(i,j) ->Yes L(i,j)
if (Check == 1)
{ O_list[0][0]=(int)((i*2)-1);
O_list[0][1]=(int)((j*2)-1); }
if (Check !=2) {
O_list[0][0]=1;
O_list[0][1]=-1;
//Find root of O(i,j)
//2 3 ;*=O(i,j)
//1)
O_list[1][0] = O_list[0][0];
O_list[1][1] = O_list[0][1] + 1;
//2)
O_list[2][0] = O_list[0][0] + 1;
O_list[2][1] = O_list[0][1];
//3)
O_list[3][0] = O_list[0][0] + 1;
O_list[3][1] = O_list[0][1] + 1; } }
////////////////////////////////////
void cut_LIS(long* count_LIS_new,int* LIS_x,int*
LIS_y,int* LIS_type,long count_LIS)
{ long count=0;
for(long z=0 ; z < count_LIS ; z++)
{ if(LIS_x[z]!=-1)
{ count = count + 1; } }
int* x_new = new int[count];
int* y_new = new int[count];
int* z_new = new int[count];
count = 0;
for( z=0 ; z < count_LIS ; z++) {
if(LIS_x[z]!=-1) {
x_new[count] = LIS_x[z];
y_new[count] = LIS_y[z];
z_new[count] = LIS_type[z];
LIS_x[z] = -1;
count = count + 1; } }
for( z=0 ; z < count ; z++)
{LIS_x[z] = x_new[z];
LIS_y[z] = y_new[z];
LIS_type[z] = z_new[z]; }
count_LIS_new[0] = count;
delete[] x_new;
delete[] y_new;
delete[] z_new;}
////////////////////////////////////
void cut_LIP(long* count_LIP_new,int* LIP_x,int*
LIP_y,long count_LIP)
{ long count=0;
for(long z=0 ; z < count_LIP ; z++) {

```

```

if(LIP_x[z]!=-1) { count = count + 1; } }
int* x_new = new int[count];
int* y_new = new int[count];
count = 0;
for( z=0 ; z < count_LIP ; z++)
{ if(LIP_x[z]!=-1) {
x_new[count] = LIP_x[z];
y_new[count] = LIP_y[z];
LIP_x[z] = -1;
count = count + 1; } }
for( z=0 ; z < count ; z++) {
LIP_x[z] = x_new[z];
LIP_y[z] = y_new[z]; }
count_LIP_new[0] = count;
delete[] x_new;
delete[] y_new; }
////////////////////////////////////
void round_image_abs(float** input,int s_x,int s_y)
{ float point=0.5;
int store_int;
for(int i = 0 ; i < s_x ; i++)
{ for(int j = 0 ; j < s_y ; j++) {
if (input[i][j] > 0) { if (input[i][j] <= 255)
{ store_int = (int)(input[i][j] + point);
input[i][j] = (float)store_int; }
else {input[i][j] = (float)(255);} }
if (input[i][j] <= 0)
{input[i][j] = 0; } } } }

```

### โปรแกรม LFC.cpp

เหมือนในภาคผนวก ก.

**ภาคผนวก จ**

**รายละเอียดโปรแกรมการเข้ารหัสและถอดรหัสเลขคณิต**

## โปรแกรม Arithm.h

```

/*-----*/
// จัดทำโดย: นาย ฉัฐนันท์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี
// ,ก.พ. 2545
/*-----*/

#include <stdlib.h>
#include <stdio.h>

#define MaxSymbols 257
#define CodeValueBits 16
#define TopValue 65535L /* 2^CodeValueBits - 1 */
#define FirstQtr 16384L /* (TopValue + 1) / 4 */
#define Half 32768L /* 2 * FirstQtr */
#define ThirdQtr 49152L /* 3 * FirstQtr */
#define MaxFrequency 4095L /* 2^12 - 1 */
typedef struct { int numb_symb, * freq, * cum_freq, *
index_to_symb, * symb_to_index;} Adaptive_Model;
typedef struct { FILE * out_file; int bit_buffer,
bit_index, bits_to_follow; long low, high, byte_counter;}
Encoder;
typedef struct { FILE * in_file; int bit_buffer, bit_index,
extra_bits; long low, high, value, byte_counter; }
Decoder;
void Create_Model(Adaptive_Model * M, int
numb_symb);
void Reset_Model(Adaptive_Model * M);
void Set_New_Model(Adaptive_Model * M, int
numb_symb);
void Dispose_Model(Adaptive_Model * M);
/* - - < Encoder //////////////////////////////////////////////////*/
void Start_Encoder(Encoder * E, char * file_name);
long Stop_Encoder(Encoder * E);
void Write_Symbol(Encoder * E, Adaptive_Model * M,
int symb);
void Write_Bits(Encoder * E, int bits, int word);
long Bytes_Used(Encoder * E);
/* - - < Decoder //////////////////////////////////////////////////*/
void Start_Decoder(Decoder * D, char * file_name);
void Stop_Decoder(Decoder * E);
int Read_Symbol(Decoder * D, Adaptive_Model * M);

```

```
int Read_Bits(Decoder * E, int bits);
```

```
long Bytes_Read(Decoder * E);
```

## โปรแกรม encode.cpp

```

/*-----*/
// จัดทำโดย: นาย ฉัฐนันท์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี
// ,ก.พ. 2545
/*-----*/

#include "arithm.h"
/* - - Main function //////////////////////////////////////////////////*/
int main(void)
{Adaptive_Model m; Encoder enc;
long befor = 0; long after;
FILE *fp;
int ch;
if((fp = fopen("data.sut", "r")) == NULL )
printf("error can not open");
// step 1
Create_Model( &m, 2); // 2 = total symbol
Start_Encoder(&enc, "data.zim" );
while (!feof(fp) ){//step 2 encode
ch = fgetc(fp);
if (ch == EOF) break; befor++;
Write_Symbol(&enc, &m, ch);
// ch is charactor sent to compress}
// end encode
Stop_Encoder(&enc);
after = Bytes_Used( &enc);
if((enc.out_file = fopen("data.zim", "a")) == NULL)
printf("error can not open file");
fwrite(&after, sizeof(after), 1, enc.out_file);
fclose(enc.out_file);
printf(" Size befor compress %d Bits\n ", befor);
printf("Size after compress %d Bits \n\n ", after*8);
Dispose_Model(&m);}
Start_Decoder(&dec, "data.zim" ); // test.ko is file output for
compress
length = bit_end; //size of data before copress
for( i = 0 ; i < length; i++){//step 2 encode
ch = Read_Symbol(&dec, &m);

```

```

// ch is charector sent to compress
if(fputc(ch,fp) == EOF)
{printf("error write "); exit(1); }
// end encode
Stop_Decoder(&dec);
Dispose_Model(&m);
printf("End of Program \n\n");
return(0);}

```

### โปรแกรม decode.cpp

```

/*-----*/
// จัดทำโดย: นาย ฉัฐนันท์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี
// ,ก.พ. 2545
/*-----*/
#include "arithm.h"
/* - - Main function
////////////////////////////////////*/
int main(void)
{int bit_end,in1[1]; //Size Data
long befor = 0; long i; long lenght =0;
int ch;
Adaptive_Model m; Decoder dec;
FILE *fp;
FILE *pheader;
////////////////////////////////////
//Load file Header SPIHT
////////////////////////////////////
pheader = fopen("header.sim","rb");
fread(in1,4,1,pheader);
//size of data wavelet SPIHTbit_end = in1[0];
fclose(pheader);
////////////////////////////////////
if((fp = fopen("output.sut","w+")) == NULL )
{printf("error can not open"); exit(1);}
// step 1
Create_Model( &m, 2); // 2 = total symbol

```

### โปรแกรม arithm.cpp

```

/*-----*/

```

```

// จัดทำโดย: นาย ฉัฐนันท์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี
// ,ก.พ. 2545
/*-----*/
#include "arithm.h"
static void Error(char * s)
{ fprintf(stderr, "%a -> Error: %s", s);
  fprintf(stderr, "\n Execution terminated...\n\n");
  exit(1);}
////////////////////////////////////
void Create_Model(Adaptive_Model * M, int ns)
{ M->numb_symb = 0; Set_New_Model(M, ns);}
////////////////////////////////////
void Reset_Model(Adaptive_Model * M)
{Set_New_Model(M, M->numb_symb);}
////////////////////////////////////
void Set_New_Model(Adaptive_Model * M, int ns)
{int i, cum;
  if ((ns < 2) || (ns > MaxSymbols))
  Error("invalid < Adaptive_Model > definition");
  if (M->numb_symb != ns) {
  if (M->numb_symb) free((char *) M->freq);
  M->numb_symb = ns; i = ns + 1;
  if ((M->freq = (int *) malloc(4 * i * sizeof(int))) == NULL)
  Error("< Adaptive_Model >: insufficient memory");
  M->cum_freq = M->freq + i;
  M->symb_to_index = M->cum_freq + i;
  M->index_to_symb = M->symb_to_index + i; }
  for (i = 0; i < ns; i++) {
  M->symb_to_index[i] = i + 1;
  M->index_to_symb[i+1] = i; }
  M->freq[0] = 0;
  for (i = 1; i <= ns; i++) M->freq[i] = 1;
  for (cum = 0, i = ns; i >= 0; i--) {
  M->cum_freq[i] = cum; cum += M->freq[i]; }
  //////////////////////////////////////
void Dispose_Model(Adaptive_Model * M)
{ if (M->numb_symb) {
  free((char *) M->freq); M->numb_symb = 0; } }
////////////////////////////////////

```

```

static void Update_Model(Adaptive_Model * M, int
index)
{int i, cum, symb_i, symb_index;
if (M->cum_freq[0] == MaxFrequency)
for (cum = 0, i = M->numb_symb; i >= 0; i--) {
M->freq[i] = (M->freq[i] + 1) >> 1;
M->cum_freq[i] = cum; cum += M->freq[i]; }
for (i = index; M->freq[i] == M->freq[i-1]; i--);
if (i < index) {symb_i = M->index_to_symb[i];
symb_index = M->index_to_symb[index];
M->index_to_symb[i] = symb_index;
M->index_to_symb[index] = symb_i;
M->symb_to_index[symb_i] = index;
M->symb_to_index[symb_index] = i; }
M->freq[i]++;
while (i) M->cum_freq[--i]++;}
////////////////////////////////////
static int Select_Symbol(Adaptive_Model * M, long
value, long * l, long * h)
{int index, cum, symbol; long range;
if ((value < *l) || (value > *h))
Error("invalid < Adaptive_Model > value");
range = *h - *l + 1;
cum = (int) (((value - *l + 1) * M->cum_freq[0] - 1) /
range);
for (index = 1; M->cum_freq[index] > cum; index++);
*h = *l + (range * M->cum_freq[index-1]) / M->
cum_freq[0] - 1;
*l = *l + (range * M->cum_freq[index]) / M->cum_freq
[0];
symbol = M->index_to_symb[index];
Update_Model(M, index);
return symbol;}
////////////////////////////////////
static void New_Interval(Adaptive_Model * M, int
symb, long * l, long * h)
{int index; long range;
if ((symb < 0) || (symb >= M->numb_symb))
Error("invalid < Adaptive_Model > symbol");
index = M->symb_to_index[symb];
range = *h - *l + 1;
*h = *l + (range * M->cum_freq[index-1]) / M->cum_freq[0]
- 1;
*l = *l + (range * M->cum_freq[index]) / M->cum_freq[0];
Update_Model(M, index);}
////////////////////////////////////
static void Output_Byte(Encoder * E)
{E->byte_counter++; E->bit_index = 8;
if (putc(E->bit_buffer, E->out_file) == EOF)
Error("< Encoder > cannot write to file");}
////////////////////////////////////
static void Bit_Plus_Follow(Encoder * E, int b)
{ E->bit_buffer >>= 1;
if (b) E->bit_buffer |= 0x80;
if (--E->bit_index == 0) Output_Byte(E);
while (E->bits_to_follow > 0) {
E->bit_buffer >>= 1; --E->bits_to_follow;
if (!b) E->bit_buffer |= 0x80;
if (--E->bit_index == 0) Output_Byte(E); }}
////////////////////////////////////
static void Update_Encoder(Encoder * E)
{ for (;) {if (E->high < Half) Bit_Plus_Follow(E, 0);
else if (E->low >= Half)
{Bit_Plus_Follow(E, 1);
E->low -= Half; E->high -= Half; }
else if ((E->low >= FirstQtr) && (E->high < ThirdQtr))
{E->bits_to_follow++; E->low -= FirstQtr;
E->high -= FirstQtr; }
else break;
E->low <<= 1; E->high += E->high + 1; }}
////////////////////////////////////
long Bytes_Used(Encoder * E)
{return E->byte_counter;}
////////////////////////////////////
void Start_Encoder(Encoder * E, char * file_name)
{char * msg = "< Encoder > cannot write to file";
if ((E->out_file = fopen(file_name, "wb")) == NULL) Error
(msg);
}

```

```

E->bit_index = 8;
E->bit_buffer = E->bits_to_follow = 0;
E->byte_counter = E->low = 0;
E->high = TopValue;}
long Stop_Encoder(Encoder * E)
{char * msg = "< Encoder > cannot write to file";
 ++E->bits_to_follow; ++E->byte_counter;
Bit_Plus_Follow(E, E->low >= FirstQtr);
if (putc(E->bit_buffer >> E->bit_index, E->out_file)
 == EOF) Error(msg);
if (fclose(E->out_file) == EOF) Error(msg);
return E->byte_counter;}
////////////////////////////////////
void Write_Symbol(Encoder * E, Adaptive_Model * M,
int s)
{New_Interval(M, s, &(E->low), &(E->high));
 Update_Encoder(E);}
////////////////////////////////////
void Write_Bits(Encoder * E, int bits, int word)
{long lm1 = E->low - 1, range = E->high - lm1, prod =
range;
word &= (1 << bits) - 1;
prod *= word;
E->high = lm1 + ((prod + range) >> bits);
E->low += prod >> bits;
Update_Encoder(E);}
////////////////////////////////////
// Decoder
////////////////////////////////////
static void Input_Byte(Decoder * D)
{if ((D->bit_buffer =getc(D->in_file)) == EOF)
if (++D->extra_bits > CodeValueBits - 2)
Error("< Decoder > attempted to read past end of file");
++D->byte_counter; D->bit_index = 8;}
////////////////////////////////////
static void Update_Decoder(Decoder * D)
{for (;;) {if (D->high >= Half)
if (D->low >= Half)
{D->value -= Half; D->low -= Half; D->high -= Half;}
else if ((D->low >= FirstQtr) && (D->high < ThirdQtr)
{D->value -= FirstQtr; D->low -= FirstQtr;
D->high -= FirstQtr; }
elsebreak;
if (!D->bit_index) Input_Byte(D);
D->low <<= 1; D->high += D->high + 1;
D->value += D->value + (D->bit_buffer & 1);
D->bit_buffer >>= 1; --D->bit_index; }}
long Bytes_Read(Decoder * D)
{return D->byte_counter;}
////////////////////////////////////
void Start_Decoder(Decoder * D, char * file_name)
{int i;
char * msg = "< Decoder > cannot read file";
if ((D->in_file = fopen(file_name, "rb")) == NULL)
Error(msg);
D->value = D->low = D->bit_index = D->extra_bits = 0;
D->high = TopValue; D->byte_counter = 0;
for (i = 1; i <= CodeValueBits; i++) {
if (!D->bit_index) Input_Byte(D);
D->value += D->value + (D->bit_buffer & 1);
D->bit_buffer >>= 1; --D->bit_index; }}
////////////////////////////////////
void Stop_Decoder(Decoder * D)
{fclose(D->in_file);}
////////////////////////////////////
int Read_Symbol(Decoder * D, Adaptive_Model * M)
{int symbol = Select_Symbol(M, D->value, &(D->low), &(D->
high));
Update_Decoder(D); return symbol;}
////////////////////////////////////
int Read_Bits(Decoder * D, int bits)
{long lm1 = D->low - 1, range = D->high - lm1, prod =
range;
int word = (int) (((D->value - lm1) << bits) - 1) / range);
prod *= word;
D->high = lm1 + ((prod + range) >> bits);
D->low += prod >> bits;
Update_Decoder(D);
return word; }

```



## ภาคผนวก จ

รายละเอียดโปรแกรมอ่านและเขียนไฟล์ภาพที่ได้จากกล้องจุลทรรศน์อิเล็กตรอน

### โปรแกรม write\_wavelet.m

```

/*-----*/
// จัดทำโดย: นาย ฐฐนันท์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี ,ก.พ. 2545
/*-----*/

%Read BMP file
X = imread('sut3.bmp','bmp');
im_in = double(X);
clear X;

save('original_image','im_in');

%size and level decomposition
x = 960;    %size row
y = 1280;   %size column

%Write Binary File
fid=fopen('D:\dwt\ant1.dat','w');
%Write Data in File
fwrite(fid,im_in,'uint8');
fclose(fid);

```

### โปรแกรม read\_wavelet.m

```

/*-----*/
// จัดทำโดย: นาย ฐฐนันท์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี ,ก.พ. 2545
/*-----*/

%size ana level
sx = 512; %row
sy = 512; %column

f1=fopen('D:\dwt\zucchini.dat','r');
[Pic1,c]=fread(f1,[sx sy],'uint8');
Pic1 = double(Pic1);

```

```

fclose(f1);
imwrite(Pic,gray(256),'E:\zucchini_0125.bmp','bmp')
%read original image
load original_image;
PSNR(Pic,im_in)

```

### ฟังก์ชัน PSNR.m

```

/*-----*/
// จัดทำโดย: นาย ณัฐนันท์ ทัดพิทักษ์กุล
// ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีสุรนารี
// ,ก.พ. 2545
/*-----*/

function PSNR = PSNR(input,output)
%PSNR = PSNR(input,output)
%input = image input
%output = image output
[Nr,Nc]=size(input);
%Check Error
%MSE = Mean square error
error=abs(input-output);
MSE=sum((sum(error.^2)))/(Nr*Nc);
%PSNR = Peak signal-to-noise ratio (dB)
PSNR =10*log10(255^2/MSE);

```

## ภาคผนวก ข

บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในขณะศึกษา

รายชื่อบทความที่ได้รับการตีพิมพ์เผยแพร่ในขณะศึกษา

1) “EM Image Compression Using the Discrete Wavelet Transform” ตีพิมพ์ใน JOURNAL OF ELECTRON MICROSCOPY SOCIETY OF THAILAND. 3<sup>rd</sup> ASEAN Microscopy Conference and 19<sup>th</sup> Annual conference of EMST . 16(1): 74-75. January 2002.

2) “การบีบอัดภาพของกล้องจุลทรรศน์อิเล็กตรอน โดยการใช้การแปลงเวฟเล็ตแบบดิสครีท” ตีพิมพ์ในวารสารวิชาการเนคเทค (ตามหนังสือตอบรับที่ วว 5204.0803/0772 ลงวันที่ 18 กุมภาพันธ์ 2545 จากศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ)

3) “การบีบอัดภาพของกล้องจุลทรรศน์อิเล็กตรอน” ตีพิมพ์ในการประชุมเสนอผลงานวิจัยระดับบัณฑิตศึกษาของประเทศไทยครั้งที่ 3 (ตามหนังสือตอบรับที่ ทม 5145/26 ลงวันที่ 26 มีนาคม 2545 จากมหาวิทยาลัยเทคโนโลยีสุรนารี)

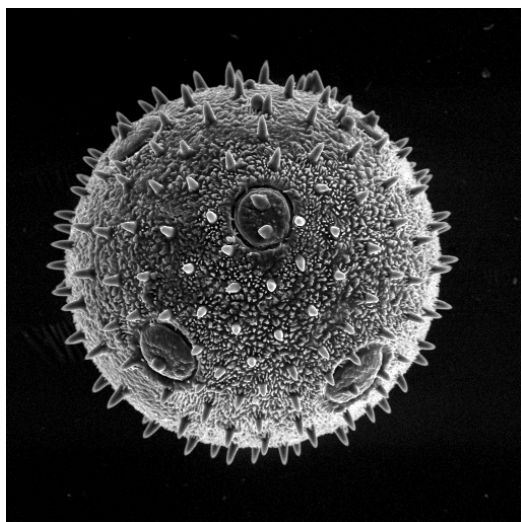
## EM Image Compression Using the Discrete Wavelet Transform

N. Thatphithakkul, K. Attakitmongcol, S. Sujitjorn

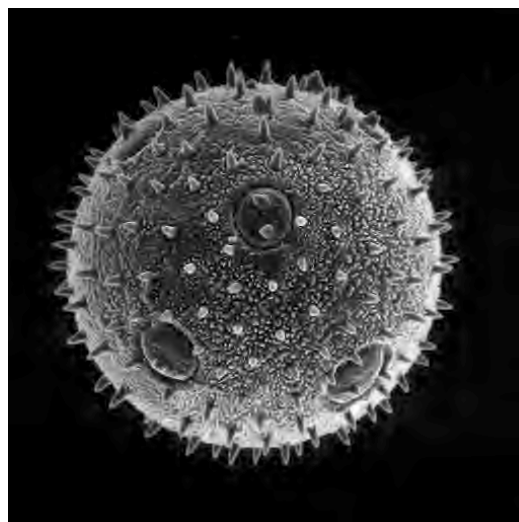
School of Electrical Engineering, Suranaree University of Technology, Thailand

The image from the Electron Microscope (EM) is effectively used to analyze the fine details of the object's surface. When the discrete wavelet transform is applied to the image, the plain surface of the object will be in the low-frequency subband and the edge will be in the high-frequency subbands. Thus, compression algorithm for the EM image must take every subband of the wavelet coefficients into account. A powerful image compression algorithm we consider is the Set Partitioning in Hierarchical Tree (SPIHT). This coding scheme exploits the self-similarity of the wavelet coefficients across different scales and searches for the high magnitude coefficients in every subband so its compression rate is better than the one by the Joint Photographic Group (JPEG) method. Furthermore, SPIHT does not cause Block Artifacts as the JPEG does. In this work, we propose an improvement of the SPIHT algorithm to increase the image compression capability by adding List of Forbidden Coefficients (LFC), extending the encoding and decoding condition, and assessing the image compression efficiency by using the Peak-signal-to-noise-ratio (PSNR) method and the Bit rate method.

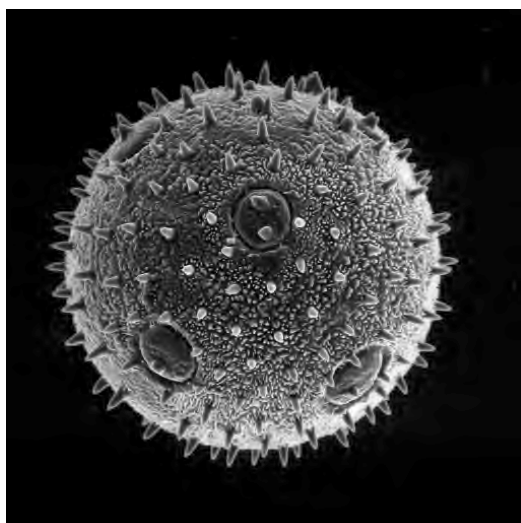
The proposed algorithm for EM image compression has been coded in C. The codes run on PentiumIII, 1 GHz, by 2-10 seconds according to different bit rates used. As shown in figure 1(a-d), the quality of the image in figure 1(d) is almost the same as that of the original in figure 1(a). The size of the compressed image file is 1/8 times the size of the original. With lower bit rates, the smaller compressed files could be obtained with degraded quality of the compressed images. The suitable bit rate should be judged against the required image quality.



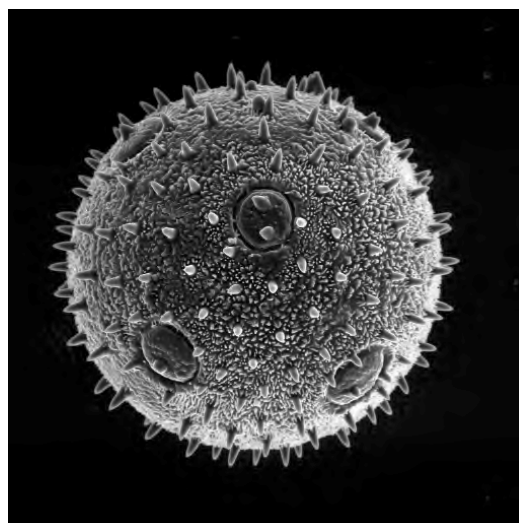
(a) Original Image (Zucchini)



(b) Bit rate = 0.25 bpp,  
PSNR = 23.1409 dB



(c) Bit rate = 0.5 bpp,  
PSNR = 26.4209 dB



(d) Bit rate = 1.00 bpp,  
PSNR = 31.4849 dB

Figure 1: Images obtained using the improved SPIHT algorithm.

## การบีบอัดภาพของกล้องจุลทรรศน์อิเล็กตรอน

### โดยใช้การแปลงเวฟเล็ตแบบดีสครีท

#### EM Image Compression Using the Discrete Wavelet Transform

ณัฐนันท์ ทัดพิทักษ์กุล<sup>1</sup> กิตติ อรรถกิจมงคล<sup>2</sup> สราวุฒิ สุจิตจร<sup>3</sup>

<sup>1</sup>นักศึกษาระดับบัณฑิตศึกษา <sup>2</sup>อาจารย์ <sup>3</sup>รองศาสตราจารย์ สาขาวิชาวิศวกรรมไฟฟ้า สำนัก  
วิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

**ABSTRACT** – The image from the Electron Microscope (EM) is effectively used to analyze the fine details of the object's surface. When the discrete wavelet transform is applied to the image, the plain surface of the object will be in the low-frequency subband and the edge will be in the high-frequency subbands. Thus, compression algorithm for the EM image must take every subband of the wavelet coefficients into account. A powerful image compression algorithm we consider is the Set Partitioning in Hierarchical Tree (SPIHT). This coding scheme exploits the self-similarity of the wavelet coefficients across different scales and searches for the high magnitude coefficients in every subband. In this paper, we propose an improvement of the SPIHT algorithm. Several wavelets are then applied for comparison in order to find the best wavelet basis for the EM image with the improved algorithm.

**KEY WORD** – Image compression, SPIHT, Wavelet

**บทคัดย่อ** – ภาพของกล้องจุลทรรศน์อิเล็กตรอน (Electron Microscope, EM) เป็นภาพที่ใช้วิเคราะห์รายละเอียดของโครงสร้างระดับไมโครบนพื้นวัตถุ ทำให้ภาพ EM เป็นภาพที่มีรายละเอียดมาก เมื่อนำมาแปลงด้วยวิธีเวฟเล็ตแบบดีสครีท ข้อมูลที่เป็นพื้นผิววัตถุจะอยู่ที่สัมประสิทธิ์ที่แบนด์ย่อยความถี่ต่ำ และข้อมูลที่เป็นขอบหรือลายเส้นของวัตถุจะอยู่ที่สัมประสิทธิ์ที่แบนด์ย่อยความถี่สูง ดังนั้นการบีบอัดข้อมูลภาพ EM ด้วยการแปลงเวฟเล็ตจะต้องให้ความสำคัญกับสัมประสิทธิ์ทุกแบนด์ย่อย อัลกอริทึม Set Partitioning in Hierarchical Tree (SPIHT) เป็นอัลกอริทึมหนึ่งที่เหมาะสม เนื่องจากอัลกอริทึมนี้เข้ารหัสโดยให้ความสำคัญกับขนาดของสัมประสิทธิ์และไม่สนใจว่าสัมประสิทธิ์ตัวนั้นจะอยู่ในระดับแบนด์ย่อยใด ในบทความนี้เสนอวิธีการพัฒนาอัลกอริทึม SPIHT ให้สามารถบีบอัดข้อมูลได้เพิ่มขึ้น พร้อมทั้งหาเวฟเล็ตแม่ที่เหมาะสมกับอัลกอริทึมที่ทำการพัฒนา

**คำสำคัญ** – การบีบอัดข้อมูลภาพ, SPIHT, เวฟเล็ต

#### 1. คำนำ

ปัจจุบันการศึกษาค้นคว้าสิ่งเร้นลับทางชีวภาพและกายภาพขนาดจิ๋ว ซึ่งไม่สามารถมองเห็นได้ด้วยตาเปล่าได้ เช่นการวิเคราะห์รายละเอียดของโครงสร้างระดับไมโครบนพื้นผิววัตถุ ไม่ว่าจะเป็นการกระจาย

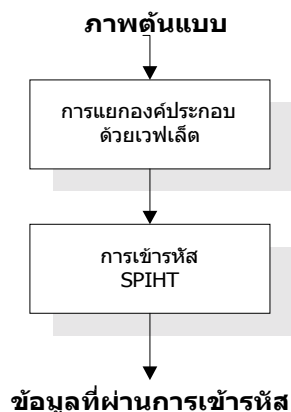
ของอนุภาค ความพรุน ความหนาแน่น หรือร่องรอยอันเกิดจากความผิดปกติของผิววัสดุ เป็นสิ่งจำเป็นอย่างยิ่งในการควบคุมคุณภาพของงานด้านโลหกรรมและวัสดุศาสตร์ และการผลิตประดิษฐกรรมทางไมโครอิเล็กทรอนิกส์ สิ่งหนึ่งที่ใช้ในการวิเคราะห์คือภาพไมโครกราฟที่ได้จากกล้อง EM จะมีอยู่ด้วยกัน 2 ประเภทคือ



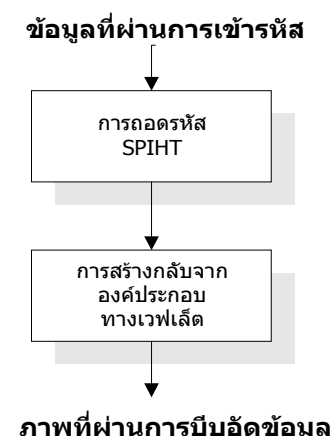
ภาพที่ได้จากกล้องโพลารอยด์ (Polaroid) ภาพที่ได้ นั้นมักไม่สามารถแยกความเปรียบต่าง (contrast) ของภาพ รวมถึงความแตกต่างของรูปลักษณะตาม บริเวณต่างๆ ได้ชัดเจน ทำให้เกิดความคลาดเคลื่อนใน การวิเคราะห์สูง [12] และภาพที่ได้จากการเปลี่ยน สัญญาณภาพจากระบบสแกนความเร็วต่ำของ EM ที่ อยู่ในรูปสัญญาณอนาล็อก จะถูกแปลงเป็นสัญญาณ เชิงตัวเลข และบันทึกเป็นไฟล์ข้อมูล ซึ่งภาพจากไฟล์ ข้อมูลยังสามารถนำไปผ่านกระบวนการปรับแต่งรายละเอียดภาพด้วยวิธีการกรองสิ่งรบกวน และผ่าน กระบวนการชดเชยให้มีความคมชัด และความเปรียบ ต่างของภาพดีขึ้น ทำให้มีความเชื่อมั่นในผลการ วิเคราะห์ที่ได้สูงขึ้น [12]

ภาพของ EM เป็นภาพที่มีรายละเอียดภาพ สูง เมื่อนำมาแปลงเวฟเลตแบบดิสครีท ข้อมูลที่เป็น พื้นผิววัตถุจะอยู่ที่สัมประสิทธิ์ที่แบนด์ย่อยความถี่ต่ำ และ ข้อมูลที่เป็นขอบหรือลายเส้นของวัตถุจะอยู่ที่ สัมประสิทธิ์ที่แบนด์ย่อยความถี่สูง ดังนั้นการบีบอัด ข้อมูลภาพ EM ด้วยการแปลงเวฟเลตจะต้องให้ความสำคัญ กับสัมประสิทธิ์ทุกแบนด์ย่อย อัลกอริทึม Embedded Zerotrees Wavelet (EZW) เป็นอัลกอริทึม หนึ่งในที่เหมาะสม เนื่องจากอัลกอริทึมนี้เข้ารหัสโดยให้ ความสำคัญกับขนาดของสัมประสิทธิ์และไม่สนใจว่า สัมประสิทธิ์ตัวนั้นจะอยู่ในระดับแบนด์ย่อยใด [8,9] และให้อัตราการบีบอัดข้อมูลดีกว่าการบีบอัดข้อมูล ภาพด้วยวิธี Joint Photographic Group (JPEG) [9] และ EZW ยังไม่ทำให้เกิดบล็อกอาร์ตifacts (Block Artifacts) เหมือน JPEG [10,11] ได้มีการพัฒนา EZW ให้มี ประสิทธิภาพเพิ่มมากขึ้นหลายวิธี แต่ใน บทความนี้เลือกใช้อัลกอริทึม Set Partitioning in Hierarchical Tree (SPIHT) ซึ่งให้อัตราการบีบอัดข้อมูลภาพที่ดีกว่า EZW [2,8] และเป็นอัลกอริทึมที่เร็ว และมีประสิทธิภาพ [3] นอกจากนี้ในบทความนี้ยัง ทำการพัฒนาอัลกอริทึม SPIHT ให้มีอัตราการบีบอัด ข้อมูลภาพที่ดีขึ้นด้วยการเพิ่ม List of Forbidden Coefficients (LFC) และเพิ่มเงื่อนไขการเข้ารหัสและการถอดรหัส พร้อมทั้งหา เวฟเลตแม่ที่เหมาะสมกับ อัลกอริทึม SPIHT ที่ทำการพัฒนา

สำหรับการบีบอัดข้อมูลภาพในบทความนี้ ประกอบ ด้วยขั้นตอนการเข้ารหัส (Encode) และขั้นตอนการถอดรหัส (Decode) ซึ่งแต่ละขั้นตอนได้แสดงไว้ในรูปที่ 1 และรูปที่ 2 ตาม ลำดับ



รูปที่ 1. การเข้ารหัสการบีบอัดข้อมูลภาพ

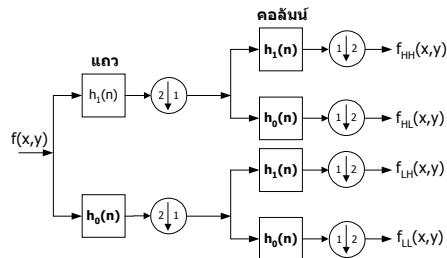


รูปที่ 2. การถอดรหัสการบีบอัดข้อมูลภาพ

## 2. การแปลงเวฟเลตแบบดิสครีท (Discrete Wavelet Transform)

การแปลงเวฟเลตของข้อมูลภาพ เป็นการแปลงเวฟเลตแบบดิสครีทแบบ 2 มิติ [1,5,7] ใช้หลักการแยกองค์ประกอบเป็นแบนด์ย่อย (Subband Decomposition) โดยมีวิธีแยกองค์ประกอบด้วยเวฟเลต (Wavelet Decomposition) เป็นดังรูปที่ 3 กำหนดให้  $f(x,y)$  คือภาพต้นแบบ  $f_{LL}(x,y)$ ,  $f_{LH}(x,y)$ ,  $f_{HL}(x,y)$  และ  $f_{HH}(x,y)$  คือสัมประสิทธิ์เวฟเลต  $h_0(n)$  และ  $h_1(n)$  คือสัมประสิทธิ์ของเวฟเลตของการแยกองค์ประกอบ (ตัวฟิลเตอร์ที่กรองความถี่ต่ำและสูงตามลำดับ) และมีวิธีการสร้างกลับจากองค์ประกอบด้วยเวฟเลต (Wavelet

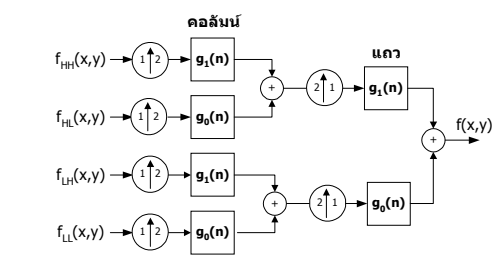
Reconstruction) เป็นดังรูปที่ 4 กำหนดให้  $g_0(n)$  และ  $g_1(n)$  คือสัมประสิทธิ์ของเวฟเล็ตของการสร้างกลับจากองค์ประกอบ (ตัวฟิลเตอร์ที่กรองความถี่ต่ำและสูงตามลำดับ)



$\begin{pmatrix} 1 & 2 \\ \downarrow & \end{pmatrix}$  คือ แผนภาพการลดข้อมูลตามแถวลงครึ่งหนึ่ง

$\begin{pmatrix} 2 & 1 \\ \downarrow & \end{pmatrix}$  คือ แผนภาพการลดข้อมูลตามคอลัมน์ลงครึ่งหนึ่ง

รูปที่. 3 การแยกองค์ประกอบด้วยเวฟเล็ต



$\begin{pmatrix} 2 & 1 \\ \uparrow & \end{pmatrix}$  คือ แผนภาพการเพิ่มข้อมูลตามคอลัมน์ขึ้นเป็นสองเท่า

$\begin{pmatrix} 1 & 2 \\ \uparrow & \end{pmatrix}$  คือ แผนภาพการเพิ่มข้อมูลตามแถวขึ้นเป็นสองเท่า

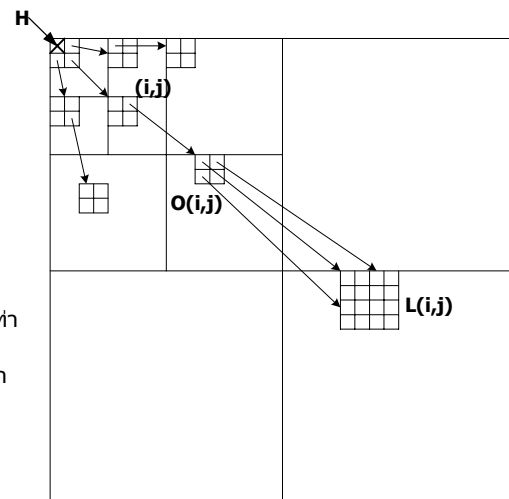
รูปที่. 4 การสร้างกลับจากองค์ประกอบด้วยเวฟเล็ต

### 3. การเข้ารหัส SPIHT [3,8]

ตำแหน่งสัมประสิทธิ์เวฟเล็ตในการเข้ารหัสมีความสัมพันธ์แบบ Spatial Orientation Trees โดยมีความสัมพันธ์เป็นดังรูปที่ 5 โดยปกติพลังงานของภาพ (Image Energy) ส่วนใหญ่จะอยู่ในส่วนที่มีความถี่ต่ำ ดังนั้นเราจะเคลื่อนที่จากแบนด์ย่อยที่สูงไปยังแบนด์ย่อยที่ต่ำกว่า นอกจากนั้นจากการสังเกต spatial ของสัมประสิทธิ์ที่มีความคล้ายกันระหว่างแบนด์ย่อยที่อยู่ในทิศทางเดียวกัน และคาดหวังว่าขนาดของสัมประสิทธิ์จะมีค่าน้อยลง โดยเริ่มจากแบนด์ย่อยสูงไปยังแบนด์ย่อยต่ำตามทิศทาง spatial ที่มีทิศทางเดียวกัน

การกำหนดตำแหน่ง (Coordinates) ของการเข้ารหัส

- $(i,j)$  คือตำแหน่งของสัมประสิทธิ์เวฟเล็ตที่โหนด(node)
- $O(i,j)$  คือกลุ่มสัมประสิทธิ์เวฟเล็ตที่ตำแหน่งของการสืบทอด (offspring) ของโหนด และ 1 โหนดมีการสืบทอด 4 ตำแหน่ง
- $(k,l)$  คือตำแหน่งของสัมประสิทธิ์เวฟเล็ตที่สืบทอดจากโหนด นั่นคือ  $O(i,j)$  จะมี  $(k,l)$  อยู่ 4 ตำแหน่ง
- $L(i,j)$  คือกลุ่มสัมประสิทธิ์เวฟเล็ตที่ตำแหน่งของการสืบทอดของ  $O(i,j)$  โดยมีการสืบทอดไปจนกว่าจะถึงแบนด์ย่อยต่ำสุด
- $D(i,j) = O(i,j) + L(i,j)$  คือตำแหน่งของการสืบทอดทั้งหมดของโหนด
- $H$  คือตำแหน่งทั้งหมดที่อยู่ในแบนด์ย่อยสูงสุด ที่เป็นต้นกำเนิดโหนด และแต่ละตำแหน่งจะมีตำแหน่งของการสืบทอด 3 ตำแหน่ง



รูปที่.5 ความสัมพันธ์แบบ Spatial Orientation Trees

ในการเข้ารหัสและถอดรหัส จะมีการเรียงในการเข้ารหัสอยู่ด้วยกัน 3 ตารางคือ

1. List of Insignificant Set (LIS) คือตารางที่ใช้เก็บตำแหน่งของกลุ่มสัมประสิทธิ์เวฟเล็ตที่ยังไม่มีความสำคัญ ซึ่ง LIS นี้จะมีอยู่ด้วยกัน 2 ชนิดคือ ชนิด A และ ชนิด B ซึ่งค่าที่เก็บใน LIS ชนิด A คือ  $D(i,j)$  และค่าที่เก็บใน LIS ชนิด B คือ  $L(i,j)$
2. List of Insignificant Pixel (LIP) คือตารางที่ใช้เก็บตำแหน่งของสัมประสิทธิ์เวฟเล็ตที่ยังไม่มีความสำคัญ

3. List of Significant Pixel (LSP) คือตารางที่ใช้เก็บตำแหน่งของสัมประสิทธิ์เวฟเล็ตที่มีความสำคัญ

มีฟังก์ชันการเข้ารหัส  $S_T(k)$  เป็นดังสมการ

$$S_T(k) = \begin{cases} 1, & \max_{(i,j) \in k} |c(i,j)| \geq T \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$n \leq \log_2(\max |c(i,j)|) \quad (2)$$

$$T = 2^n \quad (3)$$

เมื่อ  $k$  คือ สัมประสิทธิ์เวฟเล็ต  $((i,j)$  หรือ  $(k,l)$ ) หรือ กลุ่มของสัมประสิทธิ์เวฟเล็ต  $(D(i,j)$  หรือ  $L(i,j)$ )

### 3.1 อัลกอริทึมการเข้ารหัส

มีขั้นตอนดังนี้

- กำหนดค่าเริ่มต้น
  - หาค่า  $n$  และ  $T$  จากสมการที่ 2 และ 3 ตามลำดับ
  - LSP กำหนดเป็นตารางว่าง
  - LIP กำหนดเป็นค่าสมาชิกของ  $H$
  - LIS กำหนดเป็นค่าสมาชิกของ  $H$  และ กำหนดให้เป็นชนิด  $A$
- การจัดลำดับ (Sorting)
  - สำหรับแต่ละ  $(i,j)$  ใน LIP
    - ส่งค่า  $S_T((i,j))$  ออก
    - ถ้าบิตเท่ากับ 1 ให้ส่งบิตเครื่องหมาย
  - สำหรับแต่ละ  $(i,j)$  ใน LIS
    - ถ้าเป็น LIS ชนิด  $A$ 
      - ส่งค่า  $S_T(D(i,j))$  ออก
      - ถ้า  $S_T(D(i,j))$  เท่ากับ 1
        - สำหรับแต่ละ  $(k,l)$  ของ  $O(i,j)$ 
          - ส่งค่า  $S_T((k,l))$  ออก
          - ถ้าค่า  $S_T((k,l))$  เท่ากับ 1 ให้ส่งบิตเครื่องหมาย และให้ส่ง  $(k,l)$  ไป LSP
          - ถ้าค่า  $S_T((k,l))$  เท่ากับ 0 ให้ส่ง  $(k,l)$  ไป LIP

\* ถ้า  $L(i,j)$  มีสมาชิก ให้ส่ง  $(i,j)$  ไปท้าย

LIS แล้วกำหนดให้เป็นชนิด  $B$

- ถ้าเป็น LIS ชนิด  $B$

\* ส่งค่า  $S_T(L(i,j))$  ออก

~ ถ้าค่า  $S_T(L(i,j))$  เท่ากับ 1 ให้ส่งแค่สมาชิกของ  $O(i,j)$  ไปท้าย LIS แล้วกำหนดให้เป็นชนิด  $A$  และย้าย  $(i,j)$  ออกจาก LIS

### 3. Quantization

- $n = n - 1$
- สำหรับแต่ละ  $(i,j)$  ใน LSP
  - ส่งค่าบิตที่  $n$  ของ  $(i,j)$  ออก

### 4. Update

- ทำซ้ำขั้นตอนที่ 2 ถึงขั้นตอนที่ 3 และยุติเมื่อมีการส่งข้อมูลการบีบอัดครบตามต้องการ

### 3.2 การจัดเก็บเป็นไฟล์การบีบอัดข้อมูล

การจัดเก็บไฟล์แบ่งออกเป็น 2 ส่วนคือ ไฟล์ส่วนหัวและไฟล์ข้อมูลที่ได้จากการเข้ารหัส โดยไฟล์แต่ละส่วนจะมีการจัดเก็บค่าดังนี้

- ไฟล์ส่วนหัวจะทำการจัดเก็บข้อมูลคือ
  - จำนวนข้อมูลทั้งหมดที่ทำการบีบอัด
  - ค่า  $n$
  - จำนวนคอลลัมน์ของภาพ
  - จำนวนแถวของภาพ
  - จำนวนระดับการแปลงเวฟเล็ต
- ไฟล์ข้อมูลจะทำการจัดเก็บข้อมูลทั้งหมดที่ได้จากการเข้ารหัสการบีบอัดข้อมูลด้วยวิธี SPIHT

### 3.3 อัลกอริทึมการถอดรหัส

ในการถอดรหัสจะใช้อัลกอริทึมเดียวกับการเข้ารหัส เพียงแต่เปลี่ยนจากการส่งค่าเป็นการรับค่า และนำค่านั้นมาพิจารณาเหมือนการเข้ารหัส

### 4. การพัฒนาอัลกอริทึม SPIHT

จากอัลกอริทึมที่กล่าวมามีจุดที่สามารถปรับปรุงพัฒนาได้ 2 จุด กล่าวคือ

- กรณีที่  $O(i,j)$  ไม่มีระดับความสำคัญแต่  $L(i,j)$  มีระดับความสำคัญ จะมีการส่งบิตออกมา 2 บิต (ไม่นับรวมการเช็คแต่ละสมาชิกของ  $O(i,j)$ ) ในกรณีนี้สามารถปรับอัลกอริทึมให้ส่ง

ออกมาเพียง 1 บิต โดยการเพิ่มการเข้ารหัสการส่ง (k,l) เข้า LIP คือถ้ามีการส่ง (k,l) ทั้งหมดของ O (i,j) แสดงว่าได้เกิดกรณีนี้ขึ้น ดังนั้นสามารถทำการส่ง (k,l) เข้า LIS ให้เป็นชนิด A ได้เลย

2. กรณีที่  $O(i,j)$  มีระดับความสำคัญแต่  $L(i,j)$  ไม่มีระดับความสำคัญ จะมีการส่งบิตออกมาเหมือนกับข้อ 1 ในกรณีนี้สามารถปรับอัลกอริทึมให้ส่งข้อมูลให้น้อยลงได้ โดยการเพิ่ม List of Forbidden Coefficients (LFC) มีลักษณะการหา LFC ดังนี้

- ทำการแบ่งสัมประสิทธิ์เวฟเล็แต่ละแบนด์ย่อยออกเป็น 4 ส่วน โดยเรียกแต่ละส่วนที่ทำการแบ่งว่า Label โดยแต่ละแบนด์ย่อยที่มีการแบ่งจะมีค่าตำแหน่ง Label ไม่ซ้ำกัน
- หาค่าสัมประสิทธิ์สูงสุดของแต่ละ Label
- หาค่าระดับของการควอนไทซ์ในแต่ละ Label และนำมาทำเป็น LFC โดยหาได้จากสมการดังนี้

$$LFC(w) \leq \log_2(\max(c_{label}(w))) \quad (4)$$

เมื่อ  $W$  คือตำแหน่งของ Label

$c_{label}(W)$  คือสัมประสิทธิ์เวฟเล็แต่ทั้งหมดที่  $W$

$LFC(W)$  คือระดับการควอนไทซ์ ที่  $W$

โดย LFC จะเป็นตัวบ่งบอกสถานะว่า  $O(i,j)$  และ  $L(i,j)$  ควรมีการส่งบิตออกมาหรือไม่ โดยจะดูว่า  $O(i,j)$  หรือ  $L(i,j)$  นี้อยู่ที่ Label ไหนและมีระดับการควอนไทซ์ที่เท่าใด ถ้าระดับของการควอนไทซ์ที่มีอยู่มากกว่าระดับการควอนไทซ์ของ  $O(i,j)$  หรือ  $L(i,j)$  ก็ไม่ต้องทำการเข้ารหัส  $O(i,j)$  หรือ  $L(i,j)$  สามารถเขียนเป็นฟังก์ชันได้ดังสมการที่ 3

$$F_n(p) = \begin{cases} 1, & \max_{w \in p} LFC(w) \leq n \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

เมื่อ  $F_n(p)$  คือฟังก์ชันที่ใช้เป็นเงื่อนไขใน

การเข้ารหัสระดับการควอนไทซ์ของ  $O(i,j)$

หรือ  $L(i,j)$

$p$  คือค่า LFC ทั้งหมดที่เป็นของสมาชิกใน

$O(i,j)$  หรือ  $L(i,j)$

#### 4.1 อัลกอริทึมการเข้ารหัสที่มีการพัฒนา

มีการพัฒนาเฉพาะในส่วนการจัดลำดับ ในอัลกอริทึม SPIHT ส่วนอื่นจะเหมือนเดิม ดังนั้นจะแสดงเฉพาะในส่วนที่เป็นการจัดลำดับ และส่วนกำหนดค่าเริ่มต้น มีขั้นตอนดังนี้

##### 1. กำหนดค่าเริ่มต้น

- หาค่า  $n$  และ  $T$  จากสมการที่ 2 และ 3 ตามลำดับ
- LSP กำหนดเป็นตารางว่าง
- LIP กำหนดเป็นค่าสมาชิกของ  $H$
- LIS กำหนดเป็นค่าสมาชิกของ  $H$  และกำหนดให้เป็นชนิด A
- LFC กำหนดค่าเป็นไปตามสมการที่ 4

##### 2. การจัดลำดับ

- สำหรับแต่ละ  $(i,j)$  ใน LIP
  - ส่งค่า  $S_T(i,j)$  ออก
  - ถ้าบิตเท่ากับ 1 ให้ส่งบิตเช็ครีโงหมาย
- สำหรับแต่ละ  $(i,j)$  ใน LIS
  - เช็คค่า FO จาก  $F_n(O(i,j))$
  - เช็คค่า FL จาก  $F_n(L(i,j))$
  - ถ้าเป็น LIS ชนิด A และ FO หรือ FL เท่ากับ 1
    - \* ส่งค่า  $S_T(D(i,j))$  ออก
    - \* ถ้า  $S_T(D(i,j))$  เท่ากับ 1
      - สำหรับแต่ละ  $(k,l)$  ของ  $O(i,j)$ 
        - ~ ส่งค่า  $S_T(k,l)$  ออก
        - ~ ถ้าค่า  $S_T(k,l)$  เท่ากับ 1 ให้ส่งบิตเช็ครีโงหมาย และให้ส่ง  $(k,l)$  ไป LSP
        - ~ ถ้าค่า  $S_T(k,l)$  เท่ากับ 0 ให้ส่ง  $(k,l)$  ไป LIP และเพิ่มค่าตัวเข้ารหัสการส่ง  $(k,l)$
      - \* ถ้า  $L(i,j)$  มีสมาชิก
        - ~ ถ้าสมาชิกทุกตัวของ  $O(i,j)$  ถูกส่งไป LIP ให้ทำการส่งสมาชิกทุกตัวของ  $O(i,j)$  ไปท้าย LIS และกำหนดให้เป็นชนิด A

- ~ ถ้าสมาชิกบางตัวของ  $O(i,j)$  ไม่ถูกส่งไป LIS ให้ทำการส่ง  $(i,j)$  ไปท้าย LIS แล้วกำหนดให้เป็นชนิด B
- ถ้าเป็น LIS ชนิด B และ FL เท่ากับ 1
  - \* ส่งค่า  $S_r(L(i,j))$  ออก
  - ~ ถ้าค่า  $S_r(L(i,j))$  เท่ากับ 1 ให้ส่งแต่ละสมาชิกของ  $O(i,j)$  ไปท้าย LIS แล้ว กำหนดให้เป็นชนิด A และทำการย้าย  $(i,j)$  ออกจาก LIS

#### 4.2 การจัดเก็บเป็นไฟล์การบีบอัดข้อมูล

การจัดเก็บไฟล์แบ่งออกเป็น 2 ส่วนคือ ไฟล์ส่วนหัว และไฟล์ข้อมูลที่ได้จากการเข้ารหัส โดยไฟล์แต่ละส่วนจะมีการจัดเก็บค่าดังนี้

1. ไฟล์ส่วนหัวจะทำการจัดเก็บข้อมูลคือ
  - จำนวนข้อมูลทั้งหมดที่ทำการบีบอัด
  - ค่า n
  - จำนวนคอลัมน์ของภาพ
  - จำนวนแถวของภาพ
  - จำนวนระดับการแปลงเวฟเล็ต
  - ค่า LFC
2. ไฟล์ข้อมูลจะทำการจัดเก็บข้อมูลทั้งหมดที่ได้จากการเข้ารหัสการบีบอัดข้อมูลด้วยวิธี SPIHT

#### 4.3 อัลกอริทึมการถอดรหัสที่มีการพัฒนา

ในการถอดรหัสจะใช้อัลกอริทึมเดียวกับการเข้ารหัส เพียงแต่เปลี่ยนจากการส่งค่าเป็นการรับค่า และนำค่านั้นมาพิจารณาเหมือนการเข้ารหัส

### 5. การทดสอบ

#### 5.1 วิธีการทดสอบ

1. ทำการทดสอบหาตระกูลเวฟเล็ตแม่ ที่ให้ผลการบีบอัดข้อมูลภาพ EM ด้วยอัลกอริทึม SPIHT และ อัลกอริทึม SPIHT ที่ผ่านการปรับปรุง ให้ได้ผลการบีบอัดข้อมูลภาพ EM ได้ดีที่สุด โดย

ใช้ข้อมูลของภาพต้นแบบจำนวน 1 ภาพคือ “Zucchini” (เป็นภาพตัวอย่างจากกล้อง EM รุ่น JEM 2010 ของบริษัท JEOL LTD) ซึ่งมีข้อมูลภาพขนาด 512x512 พิกเซล แต่ละพิกเซลมี 8 บิต และตระกูลเวฟเล็ตแม่ที่ใช้ในการทดสอบมีอยู่ด้วยกัน 4 ตระกูลคือ bi9-7 [1] db4 [8] sym8 [5] และ coif5 [4] และทำการวัดประสิทธิภาพการบีบอัดข้อมูลภาพโดยใช้วิธี อัตราส่วนของสัญญาณต่อสัญญาณรบกวนสูงสุด (Peak signal-to-noise ratio, PSNR)

$$\text{PSNR(dB)} = 10 \log \frac{255^2}{\text{MSE}} \quad (6)$$

$$\text{MSE} = \frac{1}{M * N} \sum_{x=1}^M \sum_{y=1}^N \left[ f(x, y) - \hat{f}(x, y) \right]^2 \quad (7)$$

เมื่อ M คือจำนวนพิกเซลตามความกว้างภาพ

N คือจำนวนพิกเซลตามความสูงภาพ

$f(x,y)$  คือค่าของพิกเซลที่ตำแหน่ง  $(x,y)$  ของ ภาพต้นแบบ

$\hat{f}(x, y)$  คือค่าของพิกเซลที่ตำแหน่ง  $(x,y)$  ของภาพที่ผ่านการบีบอัดข้อมูล

ค่า PSNR จะสะท้อนถึงภาพที่ได้จากการบีบอัดว่ามีความคิดเห็นจากภาพต้นแบบมากน้อยเพียงใด นั่นคือถ้าค่า PSNR มีค่ามากแสดงว่าภาพที่ได้จากการบีบอัดมีความคิดเห็นน้อย และถ้าค่า PSNR มีค่าน้อยแสดงว่าภาพที่ได้จากการบีบอัดมีความคิดเห็นมาก และ วิธีอัตราบิต (Bit rate) คือค่าเฉลี่ยของจำนวนบิตต่อพิกเซล (Bits per pixel, bpp) ของภาพที่ผ่านการบีบอัดข้อมูล

$$\text{Bit rate(bpp)} = \frac{\text{bit compress}}{\text{pixel image}} \quad (8)$$

เมื่อ bit compress คือจำนวนบิตทั้งหมดของภาพที่ผ่านการบีบอัดข้อมูล

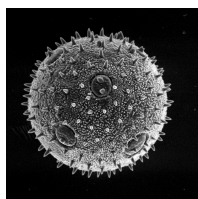
pixel image คือจำนวนพิกเซลทั้งหมดของภาพต้นแบบ

โดยอัตราบิต จะมีความสัมพันธ์กับขนาดของไฟล์ภาพที่ผ่านการบีบอัดข้อมูล เป็นดังตารางที่ 1 และทำการทดสอบที่อัตราบิตเดียวกันคือ 0.5, 1, 1.5 และ 2 bpp

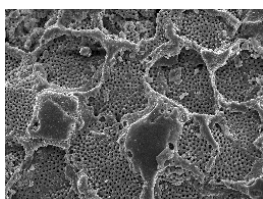
ตารางที่ 1 แสดงความสัมพันธ์ระหว่างอัตราบิต และขนาดของไฟล์ภาพที่ผ่านการบีบอัดข้อมูล

อัตราบิต (bpp)	ขนาดของไฟล์ภาพที่ผ่านการบีบอัดข้อมูล
0.25	1/32 ของไฟล์ภาพต้นแบบ
0.5	1/16 ของไฟล์ภาพต้นแบบ
1	1/8 ของไฟล์ภาพต้นแบบ
2	1/4 ของไฟล์ภาพต้นแบบ

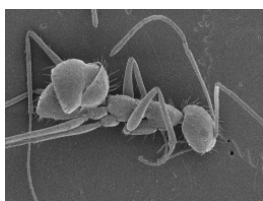
2. ทำการทดสอบอัลกอริทึม SPIHT และอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง ที่อัตราบิตเดียวกันคือ 0.25, 0.5 และ 1 bpp โดยใช้ข้อมูลของภาพต้นแบบจำนวน 4 ภาพคือ “Zucchini” และ “sand” “ant” และ “LED” (เป็นภาพที่ถ่ายจากกล้อง EM รุ่น JSM-5800 LV ของบริษัท JEOL LTD) ซึ่งมีข้อมูลภาพขนาด 960x1280 พิกเซล แต่ละพิกเซลมี 8 บิต ซึ่งภาพที่นำมาทดสอบได้แสดงไว้ในรูปที่ 6(a-d)



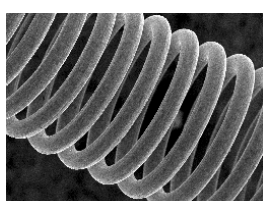
รูปที่ 6.a (Zucchini)



รูปที่ 6.b (sand)



รูปที่ 6.c (ant)



รูปที่ 6.d (LED)

รูปที่ 6 (a-d) รูปภาพที่ใช้ทดสอบการบีบอัดข้อมูล

## 5.2 ผลการทดสอบ

1. ผลการทดสอบหาตระกูลเวฟเล็ตแม่ที่ให้ผลการบีบอัดข้อมูลภาพ EM ได้ดีที่สุด พบว่าทั้งอัลกอริทึม SPIHT และอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง ที่ใช้ bi9-7 เป็นเวฟเล็ตแม่

ในการแปลงเวฟเล็ต ให้ผลการบีบอัดข้อมูลภาพ EM ได้ดีที่สุด เป็นดัง ตารางที่ 2

2. ผลการทดสอบการเปรียบเทียบประสิทธิภาพการบีบอัดข้อมูลภาพ EM ด้วยอัลกอริทึม SPIHT และอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง พบว่าอัลกอริทึม SPIHT ที่ผ่านการปรับปรุงให้ผลการบีบอัดข้อมูลภาพ EM ได้ดีกว่า เป็นดังตารางที่ 3 และมีตัวอย่างภาพที่ผ่านการบีบอัดข้อมูลด้วยอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง เป็นดังรูปที่ 7
3. เวลาที่ใช้ในการบีบอัดข้อมูลระหว่างอัลกอริทึม SPIHT และอัลกอริทึม SPIHT ที่ผ่านการ

ปรับปรุง มีการใช้เวลาในการบีบอัดข้อมูลภาพที่ใกล้เคียงกันมาก เนื่องจากเงื่อนไขในการเข้ารหัสที่เพิ่มขึ้นในอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง ไม่มีความซับซ้อน

ตารางที่ 2. แสดงผลการบีบอัดข้อมูลภาพ EM ที่ตระกูลเวฟเล็ตแม่ต่างๆ SPIHT \* คือ SPIHT ที่ผ่านการปรับปรุง

ตระกูลเวฟเล็ตแม่	อัตราบิต (bpp)	PSNR (dB)	
		SPIHT	SPIHT*
bi9-7	0.5	25.9437	26.4209
	1	30.7854	31.4849
	1.5	35.0367	35.2730
	2	38.4622	38.7493
db4	0.5	25.6627	25.7600
	1	30.0797	30.4873
	1.5	30.1865	34.7129
	2	37.3909	38.3268
Sym8	0.5	25.7788	25.8553
	1	30.1865	30.5499
	1.5	33.9292	34.6861
	2	37.3499	38.2538
Coif 5	0.5	25.5975	25.7016
	1	29.9516	30.2969
	1.5	33.6847	34.4694
	2	37.0617	38.0273

ตารางที่ 3. แสดงผลการบีบอัดข้อมูลภาพ EM ด้วย อัลกอริทึม SPIHT และอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง SPIHT\* คือ SPIHT ที่ผ่านการปรับปรุง

ภาพ	อัตราบิต (bpp)	PSNR (dB)	
		SPIHT	SPIHT*
Zucchini	0.25	22.9363	23.1409
	0.5	25.9437	26.4209
	1	30.7854	31.4849
sand	0.25	23.3655	23.5302
	0.5	25.1323	25.3597
	1	27.7760	28.0055
ant	0.25	34.5026	34.7356
	0.5	35.4126	35.4504
	1	37.1842	37.2121
LED	0.25	27.5024	27.6934
	0.5	28.8230	29.0162
	1	30.7168	30.9553

## 6. สรุป

จากผลการวิจัยที่ได้นำเสนอ จากตารางที่ 1 พบว่าการใช้ bi-9-7 เป็นเวฟเล็ตแม่ ให้ผลการบีบอัดข้อมูลภาพ EM ได้มีประสิทธิภาพมากที่สุด และจากตารางที่ 2 อัลกอริทึม SPIHT ที่ผ่านการปรับปรุง โดยการเพิ่ม LFC และเพิ่มเงื่อนไขการเข้ารหัสและการถอดรหัส ให้ผล การบีบอัดข้อมูลภาพที่ดีขึ้น

สำหรับงานวิจัยในอนาคต จะทำการพัฒนาการบีบอัดข้อมูลให้มีผลการบีบอัดข้อมูลที่ดีขึ้น ด้วยการนำมาเข้ารหัสเลขคณิต (Arithmetic Coding) และทำการหาข้อสรุปในการหาระดับอัตราบิตที่เหมาะสมของการบีบอัดข้อมูลภาพ EM จากการประเมินผลด้วยแบบสอบถาม

## เอกสารอ้างอิง

[1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. "Image coding using wavelet transform" IEEE Transaction on Image Processing. Vol 1, April 1992. pp 205-220.

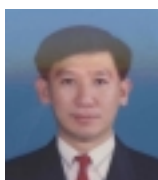
- [2] A. Bovik, "Handbook of Image & Video Processing" New York: Academic Press. 2000
- [3] B. A. Banister and T. R. Fischer. "Quadtree Classification and TCQ Image Coding." IEEE Transactions on Circuit and Systems for Video Technology. Vol. 11, No 1, January 2001. pp 3-8.
- [4] C. S. Burrus and J. E. Odegaard. "Coiflet System and Zero Moments." IEEE Transaction on Signal Processing. Vol 46, No 3, March 1998. pp. 761-766
- [5] C. S. Burrus, R. A. Gopinath and H. Guo. "Introduction to Wavelets and Wavelet Transforms." New York: Prentice-Hall International, Inc. 1998.
- [6] M. J. Lai. "On the Digital-Associated with Daubechies's Wavelets" IEEE Transaction on Signal Processing. VOL 43, September 1995. pp 2203-2205.
- [7] S. G. Mallat. "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation." IEEE Transaction on Pattern and Machine Intelligence. Vol 11, July 1989. pp.674-693.
- [8] A. Said and W. A. Pearlman., "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees." IEEE Transaction on Circuit and Systems for Video Technology. Vol. 6, June 1996. pp 243-250.
- [9] J. M. Shapiro. "Embedded image coding using zerotrees of wavelets coefficients." IEEE Transaction on Signal Processing. Vol 41, Jan 1993. pp 3445 -3462.
- [10] ขวัญฤทัย ไพรีพ่ายฤทธิ์. ครอบคลุมเวฟเล็ตที่เหมาะสมสำหรับการเข้ารหัสแบบเอ็มเบดเด็ด ซีโรทรีเวฟเล็ตในการประยุกต์ใช้งานการแพทย์ทางไกล. วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี. 2542.
- [11] ศิริพร เคชชะศิลาภิรักษ์. การลดขอบบดล็อกในภาพ JPEG ด้วยวิธีเวฟเล็ตเรโซลคั้งที่ปรับตัวเองได้. วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบัง. 2543
- [12] เดโซ ทองอร่าม และ สุวิทย์ ปุณณชัยยะ. กระบวนการทางภาพและการวิเคราะห์ภาพสำหรับกล้องจุลทรรศน์อิเล็กทรอนิกส์แบบสแกน. วารสารเครื่องมือวิจัยวิทยาศาสตร์และเทคโนโลยี. Vol. 5, 1995. หน้า 21 - 48.



นายฉันทันท์ พิศพิทักษ์กุล  
สำเร็จปริญญาตรีในสาขาวิชา  
วิศวกรรมโทรคมนาคม จาก  
มหาวิทยาลัยเทคโนโลยีสุรนารี  
เมื่อ พ.ศ. 2543 ปัจจุบันกำลัง  
ศึกษาระดับปริญญาโทที่มหาวิทยาลัยเทคโนโลยี  
สุรนารี มีความสนใจงานวิจัยทางด้าน Digital Signal  
Processing, Image Processing และ Wavelet  
Transform

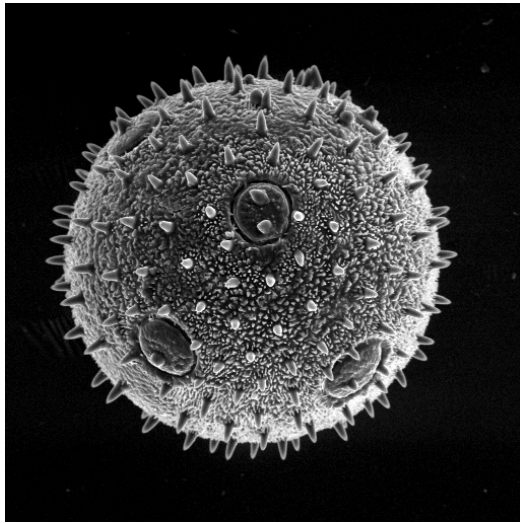


ดร.กิติ อัดถกิจมงคล สำเร็จ  
ปริญญาเอก ในสาขาวิศวกรรม  
ไฟฟ้า จาก Vanderbilt  
University ประเทศสหรัฐ  
อเมริกา เมื่อ พ.ศ. 2542 ปัจจุบัน  
ดำรงตำแหน่งเป็นอาจารย์ประจำสาขาวิชาวิศวกรรม  
ไฟฟ้า สำนักวิศวกรรมศาสตร์ มหาวิทยาลัย  
เทคโนโลยีสุรนารี มีความสนใจงานวิจัยทางด้าน  
Digital Signal Processing, Image Processing และ  
Wavelet Transform

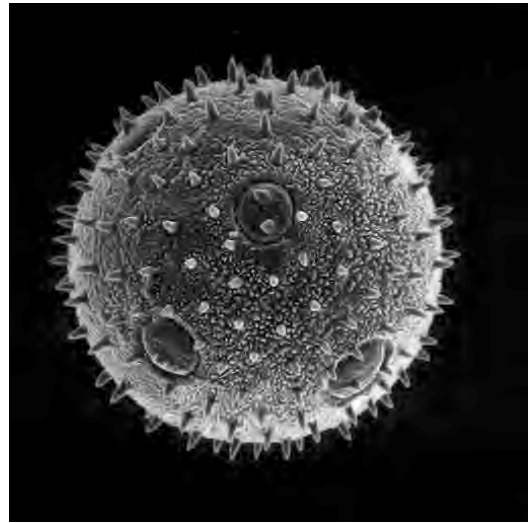


ดร.สรวุฒิ สุจิตจร สำเร็จ  
ปริญญาเอกในสาขาวิศวกรรม  
ไฟฟ้าจากมหาวิทยาลัยเบอร์มิง  
แฮม ประเทศอังกฤษ ปัจจุบัน  
ดำรงตำแหน่งรองศาสตราจารย์  
และหัวหน้าสาขาวิศวกรรมไฟฟ้า สำนักวิชา  
วิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี  
ดำเนินงานวิจัยทางด้านสัญญาณระบบและการ  
ควบคุม

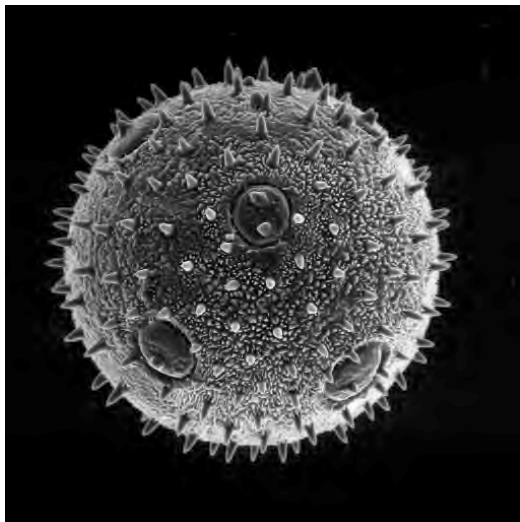




รูปที่ A1 ภาพต้นแบบ (Zucchini)  
ขนาดไฟล์เท่ากับ 262,144 ไบต์



รูปที่ A2 อัตราบิต 0.25 bpp  
ขนาดไฟล์เท่ากับ 8,192 ไบต์

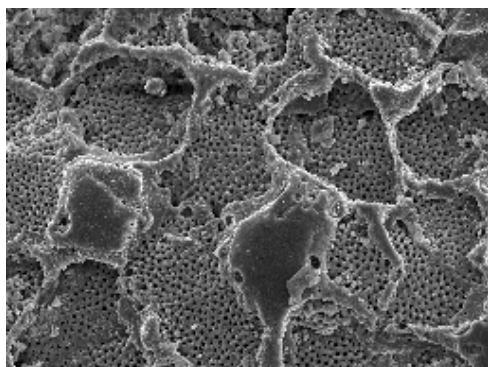


รูปที่ A3 อัตราบิต 0.5 bpp  
ขนาดไฟล์เท่ากับ 16,384 ไบต์

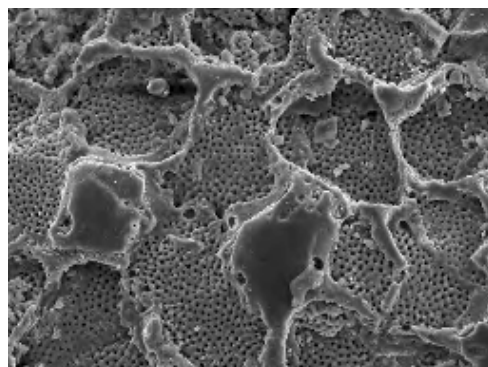


รูปที่ A4 อัตราบิต 1.00 bpp  
ขนาดไฟล์เท่ากับ 32,768 ไบต์

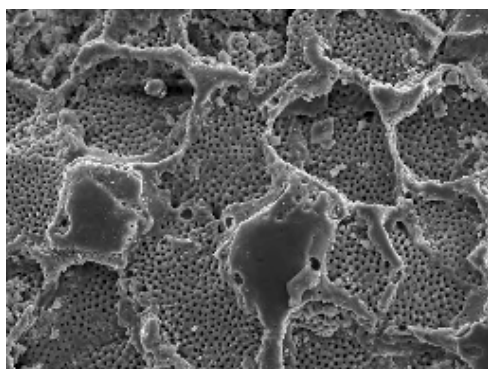
รูปที่ 7.1 รูปภาพต้นแบบ (A1) และภาพที่ผ่านการบีบอัดข้อมูลด้วยอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง (A2-A4)



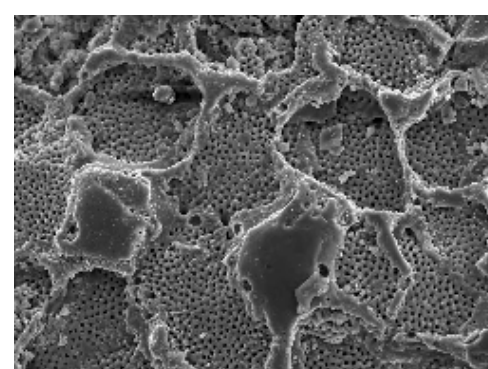
รูปที่ B1 รูปภาพดินแบบ (sand)  
ขนาดไฟล์เท่ากับ 1,228,800 ไบต์



รูปที่ B2 อัตราบิต 0.25 bpp  
ขนาดไฟล์เท่ากับ 38,400 ไบต์



รูปที่ B3 อัตราบิต 0.50 bpp  
ขนาดไฟล์เท่ากับ 76,800 ไบต์



รูปที่ B4 อัตราบิต 1.00 bpp  
ขนาดไฟล์เท่ากับ 153,600 ไบต์

รูปที่ 7.2 รูปภาพดินแบบ (B1) และภาพที่ผ่านการบีบอัดข้อมูลด้วยอัลกอริทึม SPIHT ที่ผ่านการปรับปรุง (B2-B4)



**National Electronics and Computer Technology Center**  
National Science and Technology Development Agency  
73/1 Rama VI Rd., Ratchathewi, Bangkok 10400, THAILAND  
Tel. (662) 644-8150..9 Fax. (662) 644-8137..8  
<http://www.nectec.or.th/> e-mail: [info@nectec.or.th](mailto:info@nectec.or.th)

**ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ**  
สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ  
73/1 ถนนพระรามที่ 6 เขตราชเทวี กรุงเทพฯ 10400  
โทรศัพท์ (662) 644-8150..9 โทรสาร (662) 644-8137..8  
<http://www.nectec.or.th/> e-mail: [info@nectec.or.th](mailto:info@nectec.or.th)

ที่ วว 5204.0803/0772

18 กุมภาพันธ์ 2545

เรื่อง แจ้งผลการพิจารณาบทความเพื่อตีพิมพ์ในวารสารวิชาการเนคเทค

เรียน นายสรวิชัย สุจิตจร

สิ่งที่ส่งมาด้วย สำเนาข้อเสนอแนะจากผู้ประเมิน 3 ชุด

ตามที่ท่านได้จัดส่งบทความเรื่อง “การบีบอัดภาพของกล้องจุลทรรศน์อิเล็กทรอนิกส์โดยใช้การแปลงเวฟเล็ดแบบดิสครีท” เพื่อพิจารณาตีพิมพ์ในวารสารวิชาการเนคเทคนั้น กองบรรณาธิการได้จัดส่งบทความของท่านให้ผู้ทรงคุณวุฒิในสาขาที่เกี่ยวข้องพิจารณาคุณภาพและความเหมาะสมของบทความ ซึ่งบัดนี้ได้รับผลการประเมินจากผู้ทรงคุณวุฒิจำนวน 4 ท่านเรียบร้อยแล้ว จึงใคร่ขอเรียนให้ทราบว่าบทความของท่าน **ได้รับพิจารณาให้ตีพิมพ์ในวารสารฯ** โดยทางผู้ทรงคุณวุฒิมีข้อเสนอแนะดังนี้

- เสนอให้ปรับปรุงการนิยามตัวแปรให้ชัดเจนมากยิ่งขึ้น ดังระบุในเอกสารแนบ
- เสนอให้มีการเปรียบเทียบขนาดของไฟล์ภาพทั้งก่อน และหลังผ่านวิธีนี้
- เสนอให้มีการผสม Genetic Algorithm ในการตัดเวฟเล็ดก่อนการทำ SPIHT ซึ่งน่าจะให้ผลดีขึ้น
- PSNR ขึ้นอยู่กับวิธีการเลือกตัดเวฟเล็ด หากเลือกค่าน้อยและมี coefficient ใดๆ ผลย่อมออกมาคืออยู่แล้ว
- ตรวจสอบและแก้ไขคำสะกดผิดตามเอกสารแนบ
- เสนอให้เปรียบเทียบ complexity ของระบบเดิมกับระบบที่นำเสนอ

จึงเรียนมาเพื่อโปรดทราบ และขอความอนุเคราะห์ให้ท่านปรับปรุงบทความดังกล่าวตามข้อเสนอแนะข้างต้น โดยจัดส่งบทความดังกล่าวในรูปแบบอิเล็กทรอนิกส์ให้กองบรรณาธิการที่ [chulak@nectec.or.th](mailto:chulak@nectec.or.th) ภายในวันที่ 1 มีนาคม 2545 ด้วย จักขอบคุณยิ่ง

ขอแสดงความนับถือ

(นางจุฬารัตน์ ตันประเสริฐ)

บรรณาธิการ

งานวิจัยเทคโนโลยีคลังข้อมูล  
โทรศัพท์ 0 2642 7077 ต่อ 3002  
โทรสาร 0 2642 7146

**ชื่อบทความ:** การบีบอัดภาพของกล้องจุลทรรศน์อิเล็กตรอน

**ผู้แต่ง:** ณัฐนันท์ ทัดพิทักษ์กุล<sup>1</sup> กิตติ อรรถกิจมงคล<sup>1</sup> สราวุฒิ สุจิตจร<sup>1</sup> และ ยุวดี มานะเกษม<sup>2</sup>

**ผู้นำเสนอบทความ:** ณัฐนันท์ ทัดพิทักษ์กุล

**สังกัด:** <sup>1</sup>สาขาวิชาวิศวกรรมไฟฟ้า สำนักวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

<sup>2</sup>สาขาวิชาเทคโนโลยีการผลิตพืช สำนักเทคโนโลยีการเกษตร มหาวิทยาลัยเทคโนโลยีสุรนารี

**อีเมล:** [T.Nattanun@hotmail.com](mailto:T.Nattanun@hotmail.com) **โทร:** (01) 2081443

**กลุ่มวิชา:** กลุ่มวิทยาศาสตร์-วิศวกรรมศาสตร์-เทคโนโลยี

ภาพจากกล้องจุลทรรศน์อิเล็กตรอน (Electron Microscope, EM) เป็นภาพที่ใช้วิเคราะห์รายละเอียดของโครงสร้างระดับไมโครบนพื้นวัตถุ ทำให้ภาพ EM เป็นภาพที่มีรายละเอียดมาก และไฟล์ภาพจะมีขนาดใหญ่ ทำให้ในงานวิจัยต่างๆ ที่ใช้ภาพ EM ในการวิเคราะห์รายละเอียดของงาน ที่ต้องใช้ระยะเวลายาวนานในการทำงานนั้น จำเป็นจะต้องใช้เนื้อที่ในการบันทึกข้อมูลสูง ดังนั้นถ้ามีการนำภาพ EM มาบีบอัดข้อมูล จะทำให้ช่วยลดปัญหานี้ได้ บทความวิจัยนี้นำเสนอวิธีการบีบอัดข้อมูลภาพที่เหมาะสมสำหรับภาพ EM ด้วยวิธีการแปลงเวฟเล็ตแบบดิสครีต โดยภาพ EM เมื่อนำมาแปลงด้วยวิธีเวฟเล็ตแบบดิสครีต ข้อมูลที่เป็นพื้นผิววัตถุจะอยู่ที่สัมประสิทธิ์ที่แบนด์ย่อยความถี่ต่ำ และข้อมูลที่เป็นขอบหรือรายละเอียดของวัตถุจะอยู่ที่สัมประสิทธิ์ที่แบนด์ย่อยความถี่สูง ดังนั้นการบีบอัดข้อมูลภาพ EM ด้วยการแปลงเวฟเล็ตจะต้องให้ความสำคัญกับสัมประสิทธิ์ทุกแบนด์ย่อย เพื่อให้ภาพที่ได้จากการบีบอัดมีความใกล้เคียงกับภาพต้นแบบมากที่สุด อัลกอริทึม Set Partitioning in Hierarchical Tree (SPIHT) เป็นอัลกอริทึมหนึ่งที่เหมาะสม เนื่องจากอัลกอริทึมนี้เข้ารหัสโดยให้ความสำคัญกับขนาดของสัมประสิทธิ์และไม่สนใจว่าสัมประสิทธิ์ตัวนั้นจะอยู่ในระดับแบนด์ย่อยใด และให้ผลการบีบอัดข้อมูลได้ดีกว่าการบีบอัดข้อมูลภาพด้วยวิธี Joint Photographic Expert Group (JPEG) และ SPIHT ยังไม่ทำให้เกิดบล็อกอาร์ติแฟกต์ (Block Artifacts) เหมือน JPEG ในบทความวิจัยนี้ยังเสนอวิธีการพัฒนาอัลกอริทึม SPIHT ให้สามารถบีบอัดข้อมูลได้เพิ่มขึ้น โดยการเพิ่ม List of Forbidden Coefficients (LFC) และเพิ่มเงื่อนไขการเข้ารหัสและการถอดรหัส และวัดประสิทธิภาพการบีบอัดข้อมูลภาพด้วยวิธี อัตราส่วนของสัญญาณต่อสัญญาณรบกวนสูงสุด (Peak Signal-to-Noise Ratio, PSNR) และวิธีอัตราบิต (Bit rate) และทำการหาข้อสรุปในการหาระดับอัตราบิตที่เหมาะสมของการบีบอัดข้อมูลภาพ EM จากการประเมินผลด้วยแบบสอบถาม

ที่ ทม 5145/อ.6



มหาวิทยาลัยเทคโนโลยีสุรนารี  
111 ถนนมหาวิทยาลัย ตำบลสุรนารี  
อำเภอเมือง จังหวัดนครราชสีมา 30000

๒6 มีนาคม 2545

เรื่อง แจ้งผลการพิจารณาบทความคัดย่องานวิจัย เพื่อนำเสนอในการประชุมเสนอผลงานวิจัยระดับบัณฑิตศึกษาของประเทศไทย ครั้งที่ 3  
เรียน นายณัฐนันท์ ทัดพิทักษ์กุล

- สิ่งที่ส่งมาด้วย 1. รูปแบบการจัดทำบทความเพื่อตีพิมพ์ในสื่อรวมบทความวิชาการ (Proceeding)  
2. แบบแสดงความจำนงเข้าร่วมการประชุม

ตามที่ท่านได้ส่งบทความคัดย่อผลงานวิจัย เพื่อนำเสนอในการประชุมเสนอผลงานวิจัยระดับบัณฑิตศึกษาของประเทศไทย ครั้งที่ 3 ระหว่างวันที่ 18 - 19 กรกฎาคม 2545 ณ มหาวิทยาลัยเทคโนโลยีสุรนารี นั้น

คณะกรรมการฝ่ายวิชาการจัดการประชุมฯ ขอแจ้งผลการพิจารณา ดังนี้

ผลงานวิจัยของ **นายณัฐนันท์ ทัดพิทักษ์กุล** เรื่อง **การบีบอัดภาพของกล้องจุลทรรศน์อิเล็กตรอน** ได้รับการพิจารณาให้นำเสนอในการประชุมเสนอผลงานวิจัยระดับบัณฑิตศึกษาของประเทศไทย ครั้งที่ 3 ระหว่างวันที่ 18 - 19 กรกฎาคม 2545 ณ มหาวิทยาลัยเทคโนโลยีสุรนารี โดยจัดอยู่ในกลุ่มสาขาวิจัยด้าน **วิศวกรรมศาสตร์**

ทั้งนี้ ในการพิจารณาจัดกลุ่มการนำเสนอเป็นแบบบรรยายหรือโปสเตอร์ นั้น มหาวิทยาลัยเทคโนโลยีสุรนารี จะแจ้งให้ท่านทราบโดยเร็วต่อไป และขอให้ท่านจัดทำต้นฉบับบทความเพื่อตีพิมพ์ในสื่อรวมบทความวิชาการ (Proceeding) ของการประชุมครั้งนี้ สำหรับเผยแพร่ในการประชุม และเพื่ออ้างอิงทางวิชาการ ซึ่งสามารถเลือกใช้ภาษาไทยหรือภาษาอังกฤษ โดยขอให้จัดส่งต้นฉบับภายในวันที่ 15 มิถุนายน 2545 รายละเอียดการจัดทำและการจัดส่งบทความ ปรากฏตามเอกสารสิ่งที่ส่งมาด้วย 1.

อนึ่ง หากท่านยังไม่ได้ลงทะเบียนหรือชำระค่าลงทะเบียนขอความกรุณาดำเนินการด้วย เพื่อให้การเตรียมการประชุมสามารถดำเนินการไปได้ด้วยความเรียบร้อยและทราบจำนวนผู้เข้าร่วมนำเสนอผลงานวิจัยที่แน่นอน รายละเอียดการลงทะเบียนแจ้งไว้แล้วในเอกสารสิ่งที่ส่งมาด้วย 2. หรือที่ Web Site : <http://www.sut.ac.th/gradresearch3>

จึงเรียนมาเพื่อทราบ และโปรดจัดทำต้นฉบับบทความเพื่อตีพิมพ์ ดังที่เรียนไว้ข้างต้นด้วย

ขอแสดงความนับถือ

(รองศาสตราจารย์ ดร. กฤษณะ สาคริก)

รองอธิการบดีฝ่ายวางแผน มหาวิทยาลัยเทคโนโลยีสุรนารี

ประธานคณะกรรมการฝ่ายวิชาการ

การประชุมเสนอผลงานวิจัยระดับบัณฑิตศึกษาของประเทศไทย ครั้งที่ 3

คณะกรรมการฝ่ายวิชาการฯ  
โทรศัพท์ (044) 224052-3, 224057  
โทรสาร (044) 224050

111 ถนนมหาวิทยาลัย ตำบลสุรนารี อำเภอเมือง จังหวัดนครราชสีมา 30000 โทรศัพท์ (044) 223000 โทรสาร (044) 224070

**SURANAREE UNIVERSITY OF TECHNOLOGY**

111 UNIVERSITY AVENUE, SUB DISTRICT SURANAREE, MUANG DISTRICT, NAKHON RATCHASIMA 30000, THAILAND Tel. (044) 223000 Fax. (044) 224070

### ประวัติผู้เขียน

นายฉัฐนันท์ ทัดพิทักษ์กุล เกิดเมื่อวันที่ 20 พฤศจิกายน พ.ศ. 2521 สำเร็จปริญญาตรีในสาขา วิชาวิศวกรรมโทรคมนาคม จากมหาวิทยาลัยเทคโนโลยีสุรนารี เมื่อ พ.ศ. 2543 เคยสอนวิชา ปฏิบัติการวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ในขณะที่ ศึกษาระดับปริญญาโท มีผลงานวิชาการที่ได้รับตีพิมพ์เผยแพร่ 3 บทความ ดังที่ได้รวบรวมไว้ใน ภาคผนวก ข.