

การบีบอัดข้อมูลแบบมีการสูญเสียต่ำชนิดความซับซ้อนต่ำ

A LOW COMPLEXITY TRANSFORM WITH LOW LOSS COMPRESSION

สายน้ำฝน หอมจันทร์

SAINAMPHON HOMJUN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2545

ISBN 974-324-097-7

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การบีบอัดข้อมูลแบบมีการสูญเสียต่ำชนิดความซับซ้อนต่ำ

A LOW COMPLEXITY TRANSFORM WITH LOW LOSS COMPRESSION



สายน้ำฝน หอมจันทร์

SAINAMPHON HOMJUN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2545

ISBN 974-324-097-7

A LOW COMPLEXITY TRANSFORM WITH LOW LOSS COMPRESSION

SAINAMPHON HOMJUN

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2002

ISBN 974-324-097-7

COPYRIGHT 2002

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

หัวข้อวิทยานิพนธ์	การบีบอัดข้อมูลแบบมีการสูญเสียต่ำชนิดความซับซ้อนต่ำ
นักศึกษา	นางสาวสายน้ำฝน หอมจันทร์
รหัสประจำตัว	43061151
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2545
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.กอบชัย เดชหาญ

บทคัดย่อ

เป็นการนำเสนอวิธีการบีบอัดข้อมูลภาพแบบที่มีการสูญเสียต่ำ (Low loss compression) ซึ่งอาศัยข้อดีของทั้งการแปลง (Transformation) และ การบีบอัดแบบอาศัยข้อมูลแวดล้อม (context-based compression) โดยที่วิธีที่นำเสนอนี้ช่วยลดความซ้ำซ้อนระหว่างจุดภาพ (interpixel redundancy) และ ความซ้ำซ้อนของรหัสข้อมูล (coding redundancy) ซึ่งอาศัยการแปลงแบบดิสครีตโคไซน์ (DCT: Discrete Cosine Transform) ทำให้ความสัมพันธ์ระหว่างจุดภาพลดลง ในขณะที่การเข้ารหัสแบบ Golomb-Rice เมื่อประยุกต์ใช้กับการบีบอัดแบบอาศัยข้อมูลแวดล้อม และด้วยการแมปปีง (mapping) ของสัมประสิทธิ์ DCT ทำให้การเข้ารหัสมีประสิทธิภาพสูง ผลที่ได้จากวิธีนี้ได้แสดงให้เห็นถึงอัตราการบีบอัดข้อมูลที่สูงกว่าเมื่อเทียบกับ วิธีการบีบอัดภาพแบบอาศัยข้อมูลแวดล้อม และ แบบ JPEG-baseline โดยเฉพาะกับข้อมูลที่มีความต่อเนื่องต่ำ

Thesis Title	A Low Complexity Transform with Low Loss Compression
Student	Miss Sainamphon Homjun
Student ID	43061151
Degree	Master of Engineering
Programme	Electrical Engineering
Year	2002
Thesis Advisor	Assoc.Prof.Dr.Kobchai Dejhan

ABSTRACT

This paper concerns about a new low loss image compression scheme that utilizes the advantage of both transformation and context-based compression. The interpixel and coding redundancy reduction can be achieved by this purposed method. Discrete Cosine Transformation (DCT) is used to decorreate the interpixel relation, while Golomb-Rice Coding gives the high performance of context-based compression with remapping of DCT coefficients. The results show higher compression ratio when compared with both context-based and JPEG-baseline, especially, for low continuous tone image.

กิตติกรรมประกาศ

การทำวิจัยนี้สำเร็จด้วยดี ผู้วิจัยขอกราบขอบพระคุณ รศ.ดร.กอบชัย เดชหาญ ซึ่งเป็นอาจารย์ผู้ควบคุมการทำวิจัยที่กรุณาให้แนวความคิดและคำปรึกษาแนะนำในการดำเนินงานวิจัย ตลอดจนการแก้ปัญหาต่างๆอันเป็นประโยชน์ต่องานวิจัย

ผู้วิจัยขอขอบพระคุณอาจารย์ทุกท่านที่ให้ความรู้และคำแนะนำต่างๆ บิดา-มารดาของข้าพเจ้า คุณ สมปอง วิเศษพาณิชย์กิจ ที่กรุณาให้คำปรึกษาและแนะนำการทำวิทยานิพนธ์และให้ความช่วยเหลือในด้านต่างๆ มูลนิธิเพื่อการศึกษาคอมพิวเตอร์และการสื่อสารที่ให้ทุนในการศึกษา และขอขอบคุณเพื่อนๆ พี่ๆและน้องๆทุกคนที่เป็นกำลังใจให้ด้วยดีเสมอมา ประโยชน์อันใดที่เกิดจากงานวิจัยนี้ย่อมเป็นผลมาจากความกรุณาของท่านดังกล่าวข้างต้น ผู้วิจัยรู้สึกซาบซึ้งเป็นอย่างยิ่ง จึงใคร่ขอขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

สาขาน้ำฝน หอมจันทร์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญภาพ.....	VIII
บทที่ 1 บทนำ.....	1
1.1 คำนำ.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	1
1.3 ขอบเขตของการวิจัย.....	1
บทที่ 2 การบีบอัดข้อมูลภาพ.....	3
2.1 หลักการพื้นฐาน.....	3
2.1.1 ความซ้ำซ้อนของรหัสข้อมูล.....	4
2.1.2 ความซ้ำซ้อนของข้อมูลระหว่างจุดภาพ.....	6
2.1.3 ความซ้ำซ้อนที่เกินจำเป็นในการรับรู้ของมนุษย์.....	9
2.2 ขบวนการบีบอัดข้อมูลภาพ.....	10
2.2.1 ส่วนการเข้ารหัส.....	11
2.2.2 ส่วนการถอดรหัส.....	13
2.3 การจำแนกประเภทของการบีบอัดข้อมูลภาพ.....	13
2.3.1 การเข้ารหัสเอนโทรปี.....	14
2.3.1.1 การเข้ารหัสแบบรันเรจจ์.....	14
2.3.1.2 การเข้ารหัสฮัฟแมน.....	17
2.3.1.3 การเข้ารหัสเลขคณิต.....	21
2.3.2 การเข้ารหัสซอร์ส.....	23
2.3.2.1 การเข้ารหัสด้วยการทำนาย.....	23
2.3.2.2 การเข้ารหัสด้วยการแปลง.....	23
2.3.3 การเข้ารหัสแบบไฮบริด.....	24

สารบัญ (ต่อ)

	หน้า
2.4 เทคนิคการบีบอัดข้อมูลแบบต่างๆ.....	25
2.4.1 มาตรฐานการบีบอัดข้อมูลภาพนิ่ง.....	25
2.4.2 การบีบอัดข้อมูลภาพด้วยการเข้ารหัสการควอนไทซ์แบบเวกเตอร์.....	27
2.4.3 การบีบอัดข้อมูลภาพแบบเฟคทอล.....	28
2.4.4 การบีบอัดข้อมูลภาพด้วยการแปลงเวฟเล็ต.....	30
2.5 การวัดประสิทธิภาพการบีบอัดข้อมูลภาพ.....	30
2.5.1 ค่าความผิดพลาดกำลังสองเฉลี่ย.....	30
2.5.2 SNR และ PSNR.....	31
2.5.3 การวัดประสิทธิภาพของการบีบอัดข้อมูลภาพด้วยความเข้าใจ.....	33
2.5.3.1 อัตราบิท.....	33
2.5.3.2 อัตราการบีบอัดข้อมูล.....	33
บทที่ 3 การแปลงดิสครีตโคไซน์.....	34
3.1 คุณสมบัติและนิยามของการแปลงดิสครีตโคไซน์.....	34
3.1.1 การแปลงโคไซน์ฟูเรียร์.....	34
3.1.2 นิยาม.....	37
3.2 การแปลงดิสครีตโคไซน์ 2 มิติ.....	39
3.3 การปรับปรุง และจัดลำดับค่าสัมประสิทธิ์ดิสครีตโคไซน์ในการบีบอัดข้อมูลภาพ.....	42
3.3.1 การปรับปรุงค่าสัมประสิทธิ์.....	44
3.3.2 การจัดลำดับค่าสัมประสิทธิ์.....	45
บทที่ 4 การบีบอัดข้อมูลแบบอาศัยข้อมูลเวดล้อม.....	50
4.1 บทนำ.....	50
4.2 ทฤษฎีการบีบอัดข้อมูลแบบอาศัยข้อมูลเวดล้อม.....	52
4.2.1 แบบจำลองข้อมูลเวดล้อมบนคอซอลเทมเพลต.....	52
4.2.2 การเลือกโหมดการทำงาน.....	52
4.2.3 การทำนายค่า.....	53
4.2.4 การออกแบบแบบจำลองข้อมูลเวดล้อม.....	54
4.2.4.1 การหาค่าพารามิเตอร์.....	54

สารบัญ (ต่อ)

	หน้า
4.2.4.2 การหาข้อมูลเวดล้อม.....	55
4.2.4.3 การแก้ไขค่าผิดพลาดจากการทำนายแบบปรับตัวได้.....	57
4.3 การเข้ารหัสในโหมคปกติ.....	59
4.3.1 รหัสโกลอมป์-ไรซ์.....	59
4.3.2 การหาค่า k ที่เหมาะสมสำหรับการเข้ารหัส.....	60
4.3.3 การจำกัดความยาวของรหัส.....	61
4.4 การเข้ารหัสในโหมครัน.....	62
4.5 การรีเซตค่าที่เกี่ยวข้องกับ context.....	63
4.6 สรุปขั้นตอนในการบีบอัดข้อมูลแบบอาศัยข้อมูลเวดล้อม.....	64
4.6.1 สรุปขั้นตอนหลักในโหมคปกติ.....	64
4.6.2 สรุปขั้นตอนในโหมครัน.....	65
บทที่ 5 การบีบอัดข้อมูลแบบมีการสูญเสียต่ำชนิดความซับซ้อนต่ำ.....	66
5.1 ขบวนการบีบอัดข้อมูลที่น่าสนใจ.....	66
5.2 ผลการทดลอง.....	67
5.3 สรุปผลการทดลอง.....	78
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ.....	79
6.1 สรุปผลการวิจัย.....	79
6.2 ปัญหาที่เกิดขึ้นและข้อเสนอแนะ.....	80
เอกสารอ้างอิง.....	81
ภาคผนวก ก. ผลงานวิจัยที่ได้รับการตีพิมพ์.....	82
ภาคผนวก ข. โปรแกรม.....	83
ประวัติผู้เขียน.....	107

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงตัวอย่างของรหัสความยาวไม่คงที่.....	5
2.2 การจำแนกการเข้ารหัส และวิธีการบีบอัดข้อมูลแบบต่างๆ.....	14
2.3 แสดงการแบ่งขอบเขตตามความน่าจะเป็นสะสม.....	21
3.1(a) แสดงข้อมูลขนาด 8×8	41
(b) แสดงผลลัพธ์ที่ได้จากการแปลงดิคคริต โคชานน์.....	41
3.2 ตารางเปรียบเทียบค่าผิดพลาด (Mean square error) ของภาพผลลัพธ์ที่ได้จากการปรับขนาดแซม ประสิทธิภาพดิคคริต โคชานน์โดยการใช้ค่า n ต่างกัน.....	45
5.1 แสดงการเปรียบเทียบวิธีการบีบอัดข้อมูลภาพแบบต่างๆ.....	78

สารบัญภาพ

ภาพที่	หน้า
2.1 (a) (b) แสดงข้อมูลภาพ (c) (d) แสดงการแจกแจงของระดับความสว่างที่พบในภาพ (e) (f) แสดงเปรียบเทียบค่าสัมประสิทธิ์ Auto Correlation ระหว่างภาพสองภาพ.....	7
2.2 แสดงตัวอย่างของ Run – Length Coding.....	8
2.3 แสดงตัวอย่างการทดลองของ Ernest Mach.....	9
2.4 แสดงขบวนการทั้งหมดในการบีบอัดข้อมูลภาพ.....	10
2.5 (a) แสดงขบวนการของ Source Encoder และ (b) ขบวนการของ Source Decoder.....	12
2.6 ประเภทของการบีบอัดข้อมูลภาพ.....	13
2.7 ฮัฟแมนทรีเมื่อผ่านไป 2 รอบ.....	18
2.8 แผนภาพของฮัฟแมนทรี.....	18
2.9 ตัวอย่างการเข้ารหัสข้อมูล.....	19
2.10 การถอดรหัสเมื่อได้รับรหัส '0'.....	19
2.11 การถอดรหัสเมื่อได้รับรหัส '1'.....	20
2.12 การถอดรหัสเมื่อได้รับรหัส '11'.....	20
2.13 การถอดรหัสเมื่อได้รับรหัส '111'.....	20
2.14 แสดงการแบ่งขอบเขตตามข้อมูลที่เพิ่มขึ้น.....	22
2.15 JPEG ชนิด Sequential Baseline (a) การเข้ารหัส (b) การถอดรหัส.....	26
2.16 (a) การเข้ารหัสของค่า DC (b) ลำดับการซิกแซกของการเข้ารหัสเอ็นโทรปี.....	27
2.17 ขั้นตอนการควอนไทซ์แบบเวกเตอร์ (a) การเข้ารหัส.....	27
(b) การถอดรหัส.....	28
2.18 การบีบอัดและการขยายข้อมูลภาพด้วยการแปลงเวฟเล็ต.....	30
3.1 แสดงส่วนประกอบของผลลัพธ์ที่ได้จากการแปลงดิสครีตโคซายน์ 2 มิติ.....	40
3.2 แสดงการกระจายของสัมประสิทธิ์ของการแปลงแบบฟูริเยร์ และแบบดิสครีตโคซายน์.....	42
3.3 แสดงการแปลงดิสครีตโคซายน์แบบ 2 มิติในลักษณะภาพ 3 มิติ.....	43
3.4 แสดงการแปลงแบบดิสครีตโคไซน์ในแต่ละบล็อก 8×8	43
3.5 แสดงการจัดลำดับค่าสัมประสิทธิ์ดิสครีตโคซายน์.....	45
3.6 (a)-(c) แสดงภาพผลการทดลองของภาพต้นแบบ Baboon ในการจัดลำดับค่าสัมประสิทธิ์.....	46
3.7 (a)-(c) แสดงภาพผลการทดลองของภาพต้นแบบ Lena ในการจัดลำดับค่าสัมประสิทธิ์.....	47

สารบัญญภาพ (ต่อ)

ภาพที่	หน้า
3.8 (a)-(c) แสดงภาพผลการทดลองของภาพต้นแบบภาพถ่ายจากดาวเทียม Lansat TM ในการจัดลำดับค่าสัมประสิทธิ์.....	48
3.9 (a)-(c) แสดงภาพผลการทดลองของภาพต้นแบบภาพถ่ายจากดาวเทียม Jers-1 SAR ในการจัดลำดับค่าสัมประสิทธิ์.....	49
4.1 แสดงแผนผังการบีบอัดข้อมูลแบบอาศัยข้อมูลแวดล้อมร่วมกับการเข้ารหัสแบบ Golomb –Rice.....	51
4.2 จุดภาพเสมือนที่ถูกกำหนดก่อนการบีบอัด และcausal templateที่ใช้ในแบบจำลองแวดล้อมและ การทำนายค่า.....	52
4.3 แสดงการกระจายแบบ TSGD.....	55
4.4 แสดงตัวอย่างการควอนไทซ์.....	56
4.5 แสดงการทำงานของขบวนการไบอัส.....	58
4.6 แสดงแผนผังของรหัสโกลอมบ์.....	59
4.7 แสดงการเข้ารหัสในโหมครัน.....	63
5.1 แสดงการบีบอัดข้อมูลแบบอาศัยข้อมูลแวดล้อมร่วมกับการแปลงดิสครีตโคซายน์.....	66
5.2 แสดงการนำข้อมูลกลับจากการบีบอัดข้อมูลแบบอาศัยข้อมูลแวดล้อมร่วมกับการแปลงแบบดิสครีตโคซายน์.....	66
5.3 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบ Baboon ด้วยวิธีการบีบอัดข้อมูลที่ต่างกัน.....	68
5.4 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบ Airplane ด้วยวิธีการบีบอัดข้อมูลที่ต่างกัน.....	69
5.5 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบ Lena ด้วยวิธีการบีบอัดข้อมูลที่ต่างกัน.....	70
5.6 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบ Man ด้วยวิธีการบีบอัดข้อมูลที่ต่างกัน.....	71
5.7 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบ Peppers ด้วยวิธีการบีบอัดข้อมูลที่ต่างกัน.....	72
5.8 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบ camera man ด้วยวิธีการบีบอัดข้อมูลที่ต่างกัน.....	73

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
5.9 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบจากดาวเทียม Lansat TM ด้วยวิธี การบีบอัดข้อมูลที่ต่างกัน.....	74
5.10 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบจากดาวเทียม Jers-1 OPS ด้วยวิธี การบีบอัดข้อมูลที่ต่างกัน.....	75
5.11 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบจากดาวเทียม ERS-1 SAR ด้วยวิธี การบีบอัดข้อมูลที่ต่างกัน.....	76
5.12 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบจากดาวเทียม Jers-1 SAR ด้วยวิธี การบีบอัดข้อมูลที่ต่างกัน.....	77

บทที่ 1

บทนำ

1.1 คำนำ

ทุกวันนี้คอมพิวเตอร์ได้เข้ามามีบทบาทมากขึ้น โดยมีการประยุกต์ใช้งานในด้านต่างๆ รวมทั้งทางด้านการสื่อสารข้อมูล และใช้ในการบันทึกข้อมูลต่างๆ เป็นต้น และข้อมูลส่วนหนึ่งที่ถูกนำมาใช้กับคอมพิวเตอร์นี้ จะเป็นข้อมูลภาพ โดยมีแนวโน้มที่จะมีการใช้เพิ่มขึ้น เนื่องจากข้อมูลภาพนี้ จะสามารถสื่อความหมายได้ดีกว่า แต่การใช้งานข้อมูลภาพนี้มีข้อเสียคือปริมาณของข้อมูลที่มีจำนวนมาก จะเป็นอุปสรรคให้การสื่อสารกระทำได้ช้า และสิ้นเปลืองเนื้อที่ในการจัดเก็บมาก

ขบวนการบีบอัดข้อมูลภาพจึงเป็นส่วนสำคัญที่จะเพิ่มความสามารถในการทำงานของอุปกรณ์ที่มีอยู่ ความมุ่งหมายของการบีบอัดข้อมูลภาพเพื่อต้องการลดพื้นที่ในการจัดเก็บ และ ลดค่าใช้จ่ายในการสื่อสารข้อมูล โดยยังคงรักษาคุณภาพของภาพที่ได้ไว้ เทคนิคของการบีบอัดข้อมูลมีหลายประเภท มีทั้งการบีบอัดข้อมูลแบบไม่มีการสูญเสียซึ่งได้อัตราการบีบอัดข้อมูลต่ำ และการบีบอัดข้อมูลแบบมีการสูญเสียซึ่งได้อัตราการบีบอัดข้อมูลสูง ต่อมาได้มีการพัฒนาเทคนิคการบีบอัดข้อมูลอย่างต่อเนื่อง โดยการผสมผสานเทคนิคต่างๆ เข้าด้วยกัน เพื่อให้ได้ประสิทธิภาพการบีบอัดข้อมูลสูงขึ้น ทั้งเรื่องการสูญเสีย และ อัตราการบีบอัดข้อมูล ซึ่งต้องอาศัยอัลกอริทึมที่ยุ่งยากซับซ้อน

ดังนั้นในวิทยานิพนธ์ฉบับนี้ จึงได้นำเสนอเทคนิคการบีบอัดข้อมูลแบบมีการสูญเสียต่ำชนิดความซับซ้อนต่ำโดยประยุกต์ข้อดีของการแปลงแบบคอสคริตโคซายน์ และการบีบอัดข้อมูลแบบอาศัยข้อมูลเวดล้อมด้วยการเข้ารหัส Golomb-Rice ซึ่งเป็นวิธีที่มีอัลกอริทึมที่ซับซ้อนต่ำ ให้ทำงานร่วมกันทำให้เทคนิควิธีการนี้มีประสิทธิภาพสูงขึ้น

1.2 วัตถุประสงค์ของการวิจัย

เป็นการเสนอเทคนิคการบีบอัดข้อมูลภาพแบบมีการสูญเสียต่ำชนิดความซับซ้อนต่ำ โดยจะเป็นการศึกษาถึงการแปลงแบบคอสคริตโคซายน์ และการบีบอัดแบบอาศัยข้อมูลเวดล้อม โดยมีการวัดประสิทธิภาพของการบีบอัดข้อมูลภาพด้วยค่าความผิดพลาดกำลังสองเฉลี่ย (Mean square error, MSE) และค่าอัตราการบีบอัดข้อมูล (Compression ratio, CR)

1.3 ขอบเขตของการวิจัย

ในวิทยานิพนธ์ฉบับนี้ ได้นำเสนอเทคนิคการบีบอัดข้อมูลแบบมีการสูญเสียต่ำชนิดความซับซ้อนต่ำ โดยเนื้อหาและรายละเอียดของการวิจัย แบ่งออกได้เป็น 6 บทดังต่อไปนี้

บทที่ 1 บทนำ

กล่าวถึงความจำเป็นและความสำคัญองงานวิจัย วัตถุประสงค์ของการทำวิจัยและเนื้อหาของวิทยานิพนธ์

บทที่ 2 การบีบอัดข้อมูลภาพ

กล่าวถึงหลักการพื้นฐานของการบีบอัดข้อมูลภาพ เพื่อให้เกิดความเข้าใจขั้นตอนต่างๆ ของการบีบอัดข้อมูลภาพ โดยจะอธิบายถึงขบวนการบีบอัดข้อมูลภาพ การจำแนกประเภทของการบีบอัดข้อมูลภาพ เทคนิคการบีบอัดข้อมูลภาพด้วยวิธีต่างๆ และการวัดประสิทธิภาพของการบีบอัดข้อมูลภาพ

บทที่ 3 การแปลงคิสคริตโคซายน์

กล่าวถึงคุณสมบัติและนิยามของการแปลงคิสคริตโคซายน์ การแปลงคิสคริตโคซายน์ 2 มิติ การปรับปรุงและการจัดลำดับค่าสัมประสิทธิ์คิสคริตโคซายน์ในการบีบอัดข้อมูลภาพ สุดท้ายเป็นตัวอย่างเป็นตัวอย่างภาพการจัดลำดับค่าสัมประสิทธิ์

บทที่ 4 การบีบอัดแบบอาศัยข้อมูลแวดล้อม

กล่าวถึงทฤษฎีการบีบอัดข้อมูลแบบอาศัยข้อมูลแวดล้อม ซึ่งประกอบด้วย แบบจำลองข้อมูลแวดล้อมบนคอสเทลเทมเพลท การเลือกโหนดการทำงาน การทำนายค่า และการออกแบบแบบจำลองข้อมูลแวดล้อม รายละเอียดการเข้ารหัสในโหนดปกติ รายละเอียดการเข้ารหัสในโหนดรัน การรีเซตค่าที่เกี่ยวข้องกับ context สุดท้ายเป็นการสรุปขั้นตอนการเข้ารหัสในโหนดรัน และโหนดปกติพอสังเขป

บทที่ 5 การบีบอัดข้อมูลแบบมีการสูญเสียต่ำชนิดความซับซ้อนต่ำ

กล่าวถึงขบวนการบีบอัดข้อมูลที่น่าเสนอ ทั้งส่วนการบีบอัดและการแปลงกลับผลการทดลองเปรียบเทียบประสิทธิภาพการบีบอัดข้อมูลที่น่าเสนอในวิทยานิพนธ์กับ วิธีการบีบอัดแบบอาศัยข้อมูลแวดล้อม และ การบีบอัดข้อมูลแบบ JPEG-baseline สุดท้ายเป็นการสรุปผลการทดลอง

บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ

กล่าวถึงผลสรุปจากการวิจัย ปัญหาที่เกิดขึ้นและข้อเสนอแนะสำหรับแนวทางในการพัฒนาต่อไป

บทที่ 2

การบีบอัดข้อมูลภาพ

2.1 หลักการพื้นฐาน

ขบวนการบีบอัดข้อมูล (Data compression) จะหมายถึง ขบวนการที่ใช้ในการทำให้ข้อมูล (Data) ที่ต้องใช้แทนข่าวสารหนึ่ง ๆ นั้นลดลง ซึ่งสามารถเปรียบเทียบข้อมูลได้กับตัวหนังสือที่จะสื่อความหมายถึงเนื้อหาสาระภายในหนังสือเล่มหนึ่ง ๆ นั้นเอง ในกรณีของหนังสือสองเล่มที่แต่งโดยคนละคนกันแต่มีเนื้อหาที่เหมือนกันนั้น จำนวนตัวอักษร (ข้อมูล) ที่ใช้ในการบอกเล่าจะไม่เท่ากันก็ได้ นั่นก็แสดงว่าหนังสือที่ผู้แต่งใช้จำนวนของตัวอักษรที่มากกว่าจะต้องมีคำหรือประโยคบางประโยค ที่ส่วนเกินความจำเป็น เช่น อาจเป็นประโยคที่บอกเล่าสิ่งที่ได้กล่าวถึงมาแล้วก่อนหน้านี้ หรือเป็นคำ/ประโยคที่ไม่ได้สื่อความหมายใด ๆ

การบีบอัดข้อมูลภาพก็เช่นเดียวกัน ข้อมูลของระดับความสว่างจุดภาพแต่ละจุดรวมกันเพื่อ สื่อถึงความหมายของภาพ ก็จะมีส่วนที่เกินความจำเป็นที่สามารถตัดออกไปได้

ถ้ากำหนดให้ n_1 และ n_2 เป็นจำนวนของข้อมูลที่ใช้ในการสื่อความหมายของภาพ ๆ หนึ่ง อัตราส่วนการบีบอัดของข้อมูล (Data compression ratio) จะคำนวณได้คือ

$$\text{Data compression ratio} = \frac{n_1}{n_2} \quad (2.1)$$

และอีกวิธีการหนึ่งในการวัดค่าการบีบอัดข้อมูลภาพที่นิยมใช้กันคือ การวัดจำนวนบิต (Bit) ของข้อมูลที่ต้องใช้แทนระดับความสว่างของจุดภาพใด ๆ ของภาพ ๆ นั้น (Bit Per Pixel : bpp) โดยทั่วไปแล้วภาพระดับสีเทา 256 ระดับจะต้องใช้จำนวนบิตข้อมูลต่อหนึ่งจุดภาพเท่ากับ 8 บิต (8 bpp) เมื่อนำมาผ่านขบวนการบีบอัดข้อมูลแล้วจำนวนบิตที่ต้องใช้แทนระดับความสว่างนี้จะลดลง อาจเหลือเพียง 1.1 bpp ได้โดยที่ระดับความสว่างยังคงมีได้ 256 ระดับ เท่าเดิม

วิธีการบีบอัดข้อมูลภาพ จะสามารถแบ่งออกได้ตามหลักการที่ใช้ในการตัดข้อมูลในส่วนที่ไม่จำเป็นในภาพออกไป ได้สามวิธีการใหญ่ คือ

- 1) ความซ้ำซ้อนของรหัสข้อมูล (Coding Redundancy)
- 2) ความซ้ำซ้อนของข้อมูลระหว่างจุดภาพ (Interpixel Redundancy)
- 3) ความซ้ำซ้อนที่เกินจำเป็นในการรับรู้ของมนุษย์ (Psychovisual Redundancy)

จะสามารถอธิบายรายละเอียดในแต่ละหัวข้อได้ดังต่อไปนี้ [1]

2.1.1 ความซ้ำซ้อนของรหัสข้อมูล (Coding Redundancy)

จากข้อมูลภาพซึ่งประกอบขึ้นจากจุดภาพที่มีระดับความสว่างต่าง ๆ กัน ข้อมูลของแต่ละจุดภาพนี้จะถูกแทนด้วยรหัสที่แตกต่างกันไปตามระดับความสว่างเช่น ถ้าเป็นข้อมูลของภาพระดับสีเทา 256 ระดับแล้วนั้น โดยทั่วไป แล้วจะใช้รหัสเลขฐานสอง (Binary number) ขนาด 8 บิต ในการแทนข้อมูลของหนึ่งจุดภาพตามระดับความสว่างเป็นต้น

ในภาพหนึ่ง ๆ จำนวนข้อมูลของระดับความสว่างแต่ละระดับ จะมีเกิดขึ้นมากน้อยไม่เท่ากัน โดยจะสังเกตได้จาก การทำ ฮิสโตแกรม (Histogram) ของภาพ ซึ่งจะเห็นได้ว่าจำนวนของข้อมูลหรือ รหัส ในแต่ละระดับความสว่างจะมีไม่เท่ากัน

จาก Histogram จะสามารถหาค่าความน่าจะเป็น (P_r) ในการเกิดขึ้นของรหัสต่าง ๆ (r_k) ได้ตามสมการ

$$P_r(r_k) = \frac{n_k}{n}; \quad k = 0, 1, \dots, L-1 \quad (2.2)$$

เมื่อ

L คือ จำนวนของระดับความสว่างทั้งหมด

n_k คือ จำนวนที่พบรหัสที่ระดับความสว่าง k

n คือ จำนวนของข้อมูลทั้งหมด

ถ้าให้ขนาดความยาวของรหัส (จำนวนบิต) ที่ใช้ในการแทนระดับความสว่าง (r_k) แต่ละระดับเป็น $l(r_k)$ ดังนั้นค่าเฉลี่ยความยาวของรหัส (L_{avg}) ที่ใช้จะคำนวณได้จาก

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) \cdot P_r(r_k) \quad (2.3)$$

เมื่อ

L_{avg} คือ ค่าความยาวเฉลี่ยของรหัสต่อหนึ่งจุดภาพ (bpp)

ดังนั้น จำนวนบิตของข้อมูลที่ต้องใช้ในการเข้ารหัสภาพ ซึ่งมีขนาด n จุดภาพ จะเท่ากับ $L_{avg} * n$

ในการเข้ารหัสระดับความสว่าง โดยใช้เลขฐานสองขนาด m บิต จะทำให้ค่าของ L_{avg} เท่ากับ m ด้วยเพราะ การใช้รหัสเลขฐานสองขนาด m บิต ค่าของ $l(r_k)$ จะเท่ากับ m คงที่ เมื่อนำมาแทนค่าในสมการที่ 2.3 ค่าคงที่ m จะสามารถยกออกมาได้ และผลรวมของค่าความน่าจะเป็นทั้งหมดจะเท่ากับหนึ่ง ดังแสดงได้ในตารางที่ 2.1

ตารางที่ 2.1 แสดงตัวอย่างของรหัสความยาวไม่คงที่

r_k	$P_r(r_k)$	แบบที่ 1	$l_1(r_k)$	แบบที่ 2	$l_2(r_k)$
r_0	0.19	000	3	11	2
r_1	0.25	001	3	01	2
r_2	0.21	010	3	10	2
r_3	0.16	011	3	001	3
r_4	0.08	100	3	0001	4
r_5	0.06	101	3	00001	5
r_6	0.03	110	3	000001	6
r_7	0.02	111	3	000000	6

จากข้อมูลภาพที่มี 8 ระดับความสว่าง ซึ่งในแต่ละระดับมีค่าความน่าจะเป็นที่จะเกิดขึ้นคือ p_r ในตารางที่ 2.1 ถ้าเลือกใช้รหัสเลขฐานสอง ขนาด 3 บิต เป็นตัวแทนของระดับความสว่างแต่ละระดับ (แบบที่ 1) จะได้ค่าความยาวเฉลี่ยของรหัสต่อหนึ่งจุดภาพ (bpp) คือ $L_{avg} = 3$ บิต จากค่าของ $l_1(r_k) = 3$ บิตคงที่ แต่ถ้าเลือกใช้รหัสที่มีขนาดความยาวของรหัสที่ไม่คงที่ (Variable Length Coding, แบบที่ 2) ค่าความยาวเฉลี่ยของรหัสต่อหนึ่งจุดภาพ L_{avg} จะเป็น

$$\begin{aligned}
 L_{avg} &= \sum_{k=0}^7 l_2(r_k) \cdot P_r(r_k) \\
 &= 2(0.19) + 2(0.25) + 2(0.21) + 3(0.16) + \\
 &\quad 4(0.08) + 5(0.06) + 6(0.03) + 6(0.02) \\
 &= 2.7 \text{ Bit}
 \end{aligned}$$

จากสมการที่ 2.1 จะได้ว่าค่าอัตราการบีบอัดข้อมูล (Data compression ratio) จะเท่ากับ $3/2.7$ หรือ 1.1 หรือลดลงประมาณ 10 เปอร์เซ็นต์จากการเข้ารหัสในแบบที่ 1

จากตัวอย่างข้างต้น การเข้ารหัสในแบบที่ 2 นี้ จะเป็นการเข้ารหัสโดยใช้จำนวนบิตที่น้อย กับข้อมูลระดับความสว่างที่มีระดับความน่าจะเป็นสูง และจำนวนบิตที่มากขึ้นกับข้อมูลที่มีค่าความน่าจะเป็นต่ำลงมา การเข้ารหัสแบบนี้จะเรียกว่า Variable-Length Coding

ดังนั้น จึงอาจสรุปได้ว่าการเข้ารหัสระดับความสว่าง (Grey-Level) ของภาพที่ใช้จำนวนรหัสมากกว่าที่จำเป็นแล้ว จะเรียกว่ารหัสที่มีความซ้ำซ้อนกัน (Coding Redundancy)

โดยทั่วไปแล้วการซ้ำซ้อนกันของรหัสที่ใช้ จะเกิดขึ้นเมื่อมีการใช้รหัส (Code) แทนเซตของเหตุการณ์ต่าง ๆ เช่นค่าระดับความสว่างของจุดภาพ โดยไม่ได้คำนึงถึงค่าความน่าจะเป็นในการเกิด

ขึ้นของเหตุการณ์นั้น ๆ ซึ่งส่วนใหญ่แล้วภาพระดับสีเทาจะมีลักษณะการกระจาย (Histogram) ของระดับความสว่างต่าง ๆ ไม่เท่ากัน เนื่องจากภาพโดยมากจะเป็นเพียงบางส่วนของวัตถุที่มีระดับความสว่างใกล้เคียงกัน

ดังนั้นการใช้รหัสเลขฐานสอง (Binary Number) ในการแทนระดับความสว่างก็จะไม่เป็นวิธีการที่จะให้ได้จำนวนของข้อมูลที่น้อยที่สุด

2.1.2 ความซ้ำซ้อนของข้อมูลระหว่างจุดภาพ (Interpixel Redundancy)

Interpixel Redundancy จะสามารถอธิบายได้จากตัวอย่างในภาพที่ 2.1 (a) และ (b) ทั้งสองรูปนี้จะมีค่าของ Histogram ที่เหมือนกันดังแสดงในภาพที่ 2.1 (c) และ (d) ตามลำดับ เมื่อสังเกตลักษณะของการแจกแจง (Histogram) ของระดับความสว่างนี้ จะเห็นได้ว่าภาพจะประกอบขึ้นจากจุดภาพที่มีระดับความสว่างอยู่สามช่วงใหญ่ ๆ ด้วยกัน ซึ่งจากการที่ค่าของความน่าจะเป็นของระดับความสว่างไม่เท่ากันนี้ สามารถที่จะนำหลักการของ Variable – Length Coding มาใช้เพื่อลดขนาดของรหัสที่ต้องใช้ลงได้

แต่อย่างไรก็ตามขบวนการเข้ารหัสระดับความสว่างนี้ ไม่ได้เปลี่ยนแปลงข้อมูลของจุดภาพเลย ซึ่งที่จริงแล้วข้อมูลของจุดภาพแต่ละจุดนั้น จะมีความสัมพันธ์ซึ่งกันและกันอยู่ ความสัมพันธ์กันของจุดภาพนี้จะเกิดขึ้นได้จาก โครงสร้างหรือ ลักษณะทางเลขาคณิตของวัตถุที่อยู่ในภาพนั้น ๆ

จากการคำนวณหาสัมประสิทธิ์ Auto Correlation ที่ผ่านการ Normalize แล้ว ของเส้นภาพหนึ่งเส้นในภาพ (a) และ (b) โดยใช้สมการของ Normalized Auto Correlation คือ

$$\gamma(\Delta n) = \frac{A(\Delta n)}{A(0)} \quad (2.4)$$

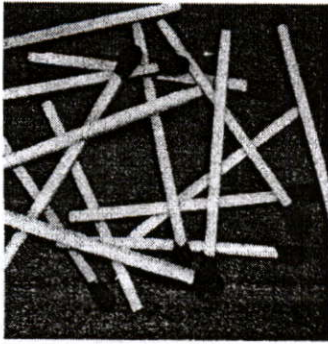
เมื่อ

$$A(\Delta n) = \frac{1}{N - \Delta n} \sum_{y=0}^{N-1-\Delta n} f(x, y) \cdot f(x, y + \Delta n) \quad (2.5)$$

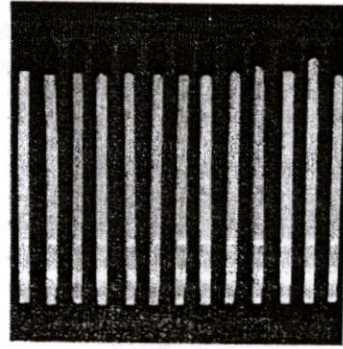
x ตำแหน่งของเส้นภาพที่นำมาคำนวณ

Δn ระยะต่างของจุดภาพที่นำมาเปรียบเทียบ

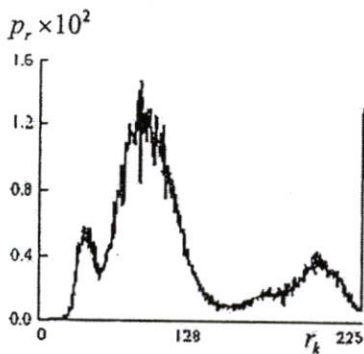
ผลที่ได้ของสัมประสิทธิ์ Auto Correlation ที่ได้โดยการเปลี่ยนค่าของ Δn จะได้ดังแสดงในภาพที่ 2.1 (e) และ (f) ตามลำดับ ซึ่งแสดงให้เห็น ถึงความแตกต่างกันคือ ที่ค่าของ Δn เท่ากับ 45 และ 90 จุดภาพ ข้อมูลจะมีความสัมพันธ์กันมาก (highly correlated) ซึ่งที่ค่าของ Δn เท่ากับ 45 และ 90 จุดภาพนี้ เป็นระยะเดียวกันกับความห่างของก้านไม้ขีดไฟในภาพพอดี้ และนอกจากนี้ จะเห็นได้ว่าค่าของสัมประสิทธิ์ γ ของเส้นภาพทั้งสองนี้ ที่ค่าของ Δn เท่ากับหนึ่ง จะเป็น 0.9922 และ 0.9928



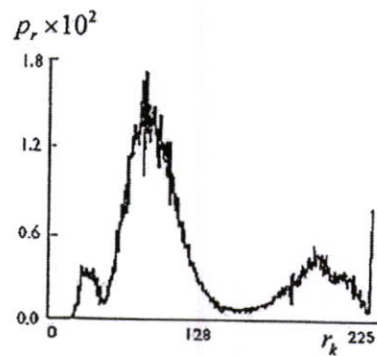
(a)



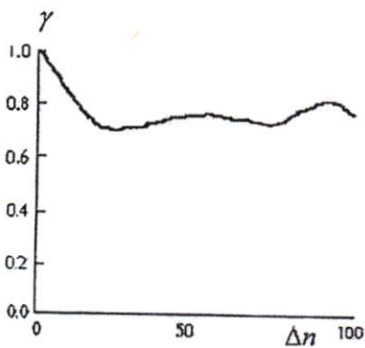
(b)



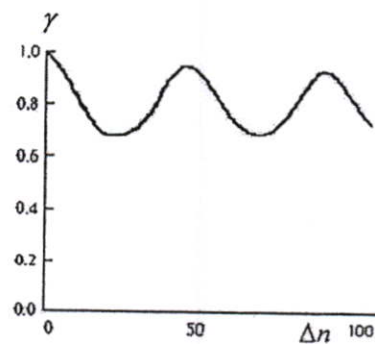
(c)



(d)



(e)



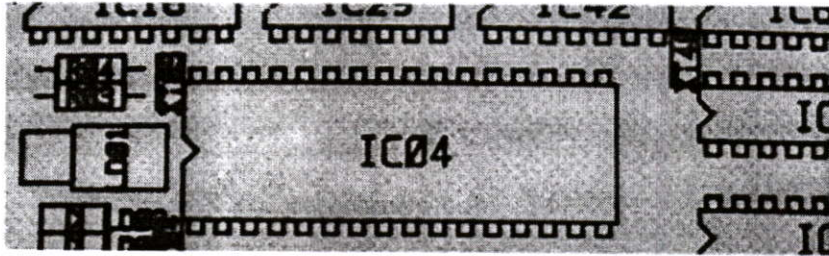
(f)

ภาพที่ 2.1 (a)(b) แสดงข้อมูลภาพ (c)(d) แสดงการแจกแจงของระดับความสว่างที่พบในภาพ (e)(f) แสดงเปรียบเทียบค่าสัมประสิทธิ์ Auto Correlation ระหว่างภาพสองภาพ

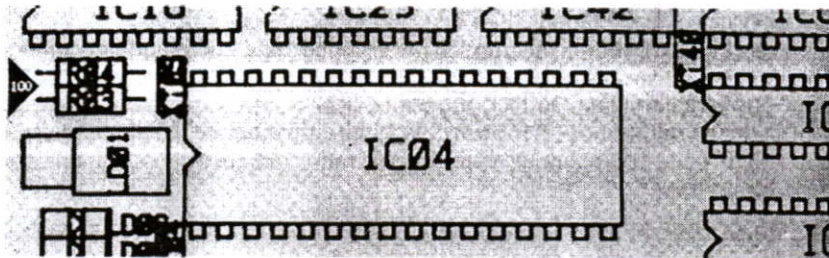
จากตัวอย่างข้างต้นจะพบว่า การซ้ำกันของข้อมูลภาพจะเกิดขึ้นได้จากความสัมพันธ์กันระหว่างจุดภาพที่อยู่ติด ๆ กัน ซึ่งจะมีค่าที่ใกล้เคียงกัน ดังนั้นค่าของข้อมูลที่จุดภาพใด ๆ จะสามารถตัดทิ้งออกไปได้ และจะสามารถสร้างกลับมาใหม่โดยการประมาณค่าจากจุดภาพที่อยู่รอบ ๆ นั้นเอง และจากการที่ข้อมูลในแต่ละจุดภาพสามารถที่จะคำนวณกลับออกมาใหม่ได้นั้นแสดงว่าข้อมูล

ในจุดภาพหนึ่ง ๆ นั้นจะมีความสำคัญมากต่อความหมายทั้งหมดของภาพ ด้วยเหตุนี้ จึงอาจกล่าวได้ว่าข้อมูลในแต่ละจุดภาพนั้นมีความซ้ำซ้อนกันเอง (Interpixel Redundancy)

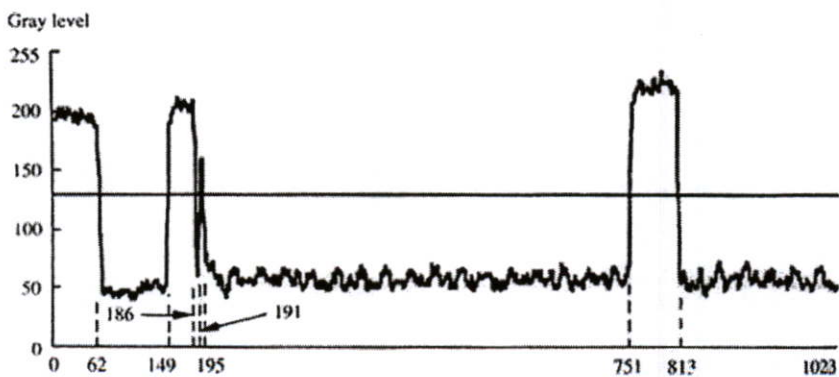
การที่จะลด Interpixel Redundancy ที่เกิดขึ้นในภาพ ข้อมูลภาพซึ่งเป็น อาร์เรย์สองมิติของระดับความสว่าง จะต้องถูกแปลงให้อยู่ในรูปแบบที่เหมาะสมกว่า ตัวอย่างเช่น การแทนข้อมูลภาพด้วยผลต่างระหว่างจุดภาพสองจุดที่อยู่ติดกันก็จะแทนข้อมูลภาพทั้งหมดได้ ซึ่งจากที่ข้อมูลระดับความสว่างนั้นมีค่าใกล้เคียงกัน จะทำให้จำนวนแบบรหัสที่ต้องใช้แทนข้อมูลที่ถูกลดแล้วลดลง



(a)



(b)



(c)

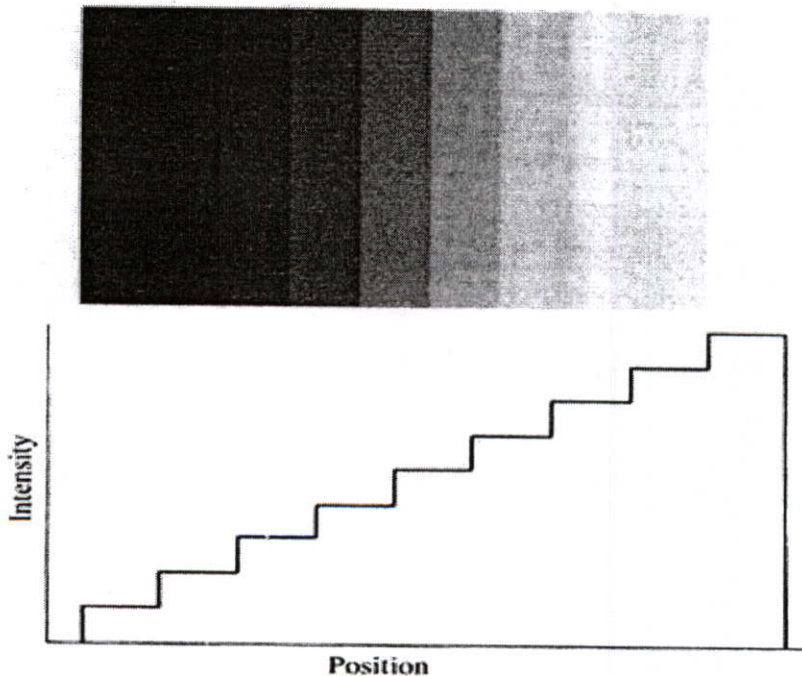
ภาพที่ 2.2 แสดงตัวอย่างของ Run – Length Coding

ขบวนการแปลงข้อมูลเพื่อใช้ในการลด Interpixel Redundancy นี้จะเรียกว่าส่งถ่าย (Mapping) และจะเรียกขบวนการส่งถ่าย นี้ว่าสามารถย้อนกลับได้ (Reversible) ถ้าข้อมูลของภาพต้นแบบสามารถสร้างกลับออกมาได้ จากข้อมูลที่ผ่านมาขบวนการส่งถ่ายแล้ว

การทำ Mapping สามารถแสดงได้ดังตัวอย่างจากภาพที่ 2.2 (a) เป็นภาพของส่วนประกอบของวงจรีเล็กทรอนิกส์ ขนาด 1×3 นิ้ว ที่ความละเอียด 330 dpi (Dot Per Inch: จุดต่อนิ้ว) และภาพที่ 2.2 (b) เป็นภาพที่มีสองระดับความสว่าง (Binary Image) การ Mapping ข้อมูลของภาพให้เป็น Binary Image จะทำโดยใช้วิธีการตั้งค่า Threshold ซึ่งแสดงในภาพที่ 2.2 (c) ซึ่งเป็นค่าระดับความสว่างของเส้นภาพหนึ่งเส้น จะเห็นได้ว่า Binary Image ที่ได้จะมีลักษณะที่สังเกตได้คือจะมีจุดภาพที่มีระดับความสว่างเดียวกันอยู่ติดกันเป็นจำนวนมาก ด้วยเหตุนี้จะมีวิธีการที่จะเก็บข้อมูลลักษณะนี้ให้น้อยลงได้โดย จากข้อมูลระดับความสว่างของจุดภาพในเส้นภาพ คือ $f(x,0) f(x,1) f(x,2) \dots f(x, N-1)$ ให้เป็นคู่ลำดับของข้อมูล $(g_1, r_1) (g_2, r_2) (g_3, r_3) \dots$ เมื่อ g หมายถึงค่าของระดับความสว่าง และ r_i หมายถึงจำนวนที่พบติดต่อกัน วิธีการดังกล่าวนี้เรียกว่า Run-Length Coding ข้อมูลเมื่อผ่านขบวนการส่งถ่ายนี้แล้วจะมีความซ้ำซ้อนกันน้อยลง และไม่สามารถมองออกเป็นภาพได้

2.1.3 ความซ้ำซ้อนที่เกินจำเป็นในการรับรู้ของมนุษย์ (Psychovisual Redundancy)

จากการรับรู้ภาพวัตถุของตามมนุษย์ที่ไม่ได้เป็นสัดส่วนโดยตรงกับระดับความสว่าง หรือแสงที่สะท้อนมาจากวัตถุเท่านั้นแต่จะขึ้นอยู่กักระดับความสว่างของสิ่งแวดล้อมรอบ ๆ ด้วย ดังตัวอย่างจากการทดลองของ Ernest Mach [1] ดังภาพที่ 2.3



ภาพที่ 2.3 แสดงตัวอย่างการทดลองของ Ernest Mach

เส้นกราฟด้านล่างจะแสดงถึงระดับความสว่างของภาพที่อยู่ด้านบน แต่จะพบว่าในการรับรู้ความสว่างของตามมนุษย์นั้น เมื่อสังเกตภาพข้างต้นจะเป็นเห็นแถบที่มีความมืดลงที่ตำแหน่งด้าน ขวาของแถบ และสว่างขึ้นทางด้านซ้าย จากการทดลองนี้จะแสดงให้เห็นว่า แถบที่มีระดับความสว่างคงที่ จะมีการเปลี่ยนแปลง ไปในการรับรู้ของตามมนุษย์

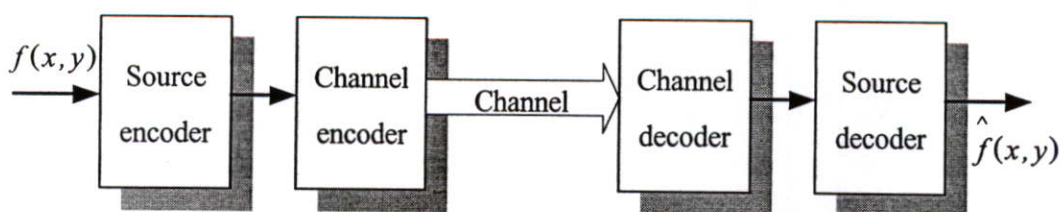
ปรากฏการณ์ลักษณะนี้เกิดขึ้นจากความจริงที่ว่า ตาของมนุษย์นั้น ตอบสนองต่อระดับความสว่างในส่วนต่าง ๆ ของภาพได้ไม่เท่ากัน ระดับความสว่างของภาพบางส่วนไม่ได้มีความหมายใด ๆ ต่อการรับรู้ข้อมูลภาพนั้น ๆ ซึ่งรายละเอียดของภาพในส่วนที่ไม่มีความสำคัญนี้ จะเรียกว่าเป็น Psychovisual Redundant ที่จะสามารถตัดออกไปจากข้อมูลภาพได้ โดยไม่ทำให้ความหมายของภาพในการรับรู้เปลี่ยนไป เนื่องจากในการรับรู้ความหมายของภาพ โดยมนุษย์นั้นไม่ได้เป็นการพิจารณาถึงระดับความสว่างของแต่ละจุดภาพ โดยทั่วไปตาของมนุษย์จะสังเกตถึงคุณลักษณะที่ จะแบ่งแยกวัตถุ หรือสิ่งต่าง ๆ ออกจากกัน เช่น ขอบของรูป หรือบริเวณของพื้นผิว เป็นต้น รวมเข้าด้วยกันเป็นวัตถุ หรือรูปร่างต่าง ๆ และสมองจะทำการแยกแยะวัตถุ หรือรูปร่างนั้น ๆ ออกมาโดยเปรียบเทียบเข้ากับสิ่งที่เคยรู้จักมาแล้ว

Psychovisual Redundancy จะมีหลักการพื้นฐาน ที่ไม่เหมือนกัน การลดความซ้ำซ้อนกันทั้งสองแบบที่ได้อธิบายไปแล้ว Psychovisual Redundancy นี้จะเกี่ยวข้องโดยตรงกับการมองเห็นของตามมนุษย์ ข้อมูลส่วนที่ซ้ำซ้อนกันนั้นก็เนื่องมาจาก มันไม่ได้มีความหมายในการรับรู้ตัวเอง ซึ่งการตัดข้อมูลในส่วนนี้ทิ้งไปนั้น จะทำให้ข้อมูลของภาพในเชิงปริมาณเสียไป

ขบวนการนี้เช่นขบวนการของ Quantization ซึ่งจะหมายถึงขบวนการ Mapping ข้อมูลอินพุตที่มีจำนวนหลาย ๆ แบบ ไปเป็นข้อมูลที่มีจำนวนแบบที่ลดลง และขบวนการ Quantization นี้จะเป็นขบวนการที่ไม่สามารถกระทำย้อนกลับได้ (Irreversible) เนื่องจากข้อมูลส่วนที่ถูกตัดออกไปแล้วนั้น จะไม่สามารถนำกลับมาได้

2.2 ขบวนการบีบอัดข้อมูลภาพ

จากหัวข้อที่ผ่านมา ได้กล่าวถึงวิธีการในการลดความซ้ำซ้อนกันที่มีในภาพ ซึ่งมีหลักการอยู่สามแบบด้วยกัน ในทางปฏิบัติแล้ว การลดขนาดของข้อมูลภาพ จะเป็นการนำ เอาหลักการการทั้งสามนี้มาประยุกต์ใช้งานร่วมกัน เป็นขบวนการบีบอัดข้อมูลภาพที่ต้องการ



ภาพที่ 2.4 แสดงขบวนการทั้งหมดในการบีบอัดข้อมูลภาพ

จากภาพที่ 2.4 จะเป็นขบวนการทั้งหมดของการบีบอัดข้อมูลภาพ โดยจะแยกออกเป็นสองส่วน คือ ขั้นตอนของขบวนการเข้ารหัส (Encoder) และ ขั้นตอนของการถอดรหัส (Decoder) ข้อมูลภาพ ต้นแบบ (Input Image: $f(x, y)$) จะผ่านขบวนการของ Encoder ซึ่งจะเข้ารหัสแทนของข้อมูลนี้ ที่มีจำนวนลดลง หลังจากเข้ารหัสแทนข้อมูลนี้ผ่านตัวกลาง (Channel) มายังภาค Decoder ที่จะทำการสร้าง ภาพผลลัพธ์ (Output Image: $\hat{f}(x, y)$) กลับออกมาเพื่อนำไปใช้งานต่อไป

โดยทั่วไปแล้ว $f(x, y)$ ที่ได้กลับออกมานี้อาจเหมือนหรือไม่เหมือนกับ $f(x, y)$ ก็ได้ ขึ้นอยู่กับขบวนการที่ใช้ว่าเป็นแบบที่มีความผิดพลาดหรือไม่ ถ้าเป็นขบวนการแบบที่มีความผิดพลาดแล้วนั้น $f(x, y)$ ที่ได้ก็จะมี ความผิดพลาดเกิดขึ้น

จากภาพที่ 2.4 ในส่วนของ Encoder และ Decoder จะประกอบขึ้นมาจากส่วนย่อย ๆ สองส่วนที่มีหน้าที่ ๆ แตกต่าง กันคือ สำหรับในส่วนของ Encoder จะประกอบไปด้วย Source Encoder ซึ่งทำหน้าที่ในการลดการซ้ำซ้อนกันของข้อมูลภาพ และ Channel Encoder ซึ่งทำหน้าที่ในการเพิ่มความต้านทานของข้อมูล ต่อสัญญาณรบกวนที่จะเกิดขึ้นในการส่งข้อมูลผ่านตัวกลาง และเช่นเดียวกัน สำหรับในส่วนของ Decoder ซึ่งประกอบไปด้วย Channel Decoder และตามด้วย Source Decoder ตามลำดับ

ในกรณีที่ตัวกลางในการส่งข้อมูล เป็นแบบที่ไม่มีความผิดพลาด หรือสัญญาณรบกวนเกิดขึ้นแล้ว เช่นในการเก็บข้อมูลในรูปแบบของ ดิจิตอล ในอุปกรณ์บันทึกข้อมูล ส่วนของ Channel Encoder และ Channel Decoder จะไม่มีความจำเป็น จึงจะเหลือเฉพาะแต่ Source Encoder และ Source Decoder ตามลำดับ

2.2.1 ส่วนการเข้ารหัส

ในส่วนเข้ารหัส (Source Encoder) นี้จะมีหน้าที่สำหรับ การลดความซ้ำซ้อนกันของข้อมูลในภาพทั้งในส่วนของ Coding Redundancy Interpixel Redundancy และ Psychovisual Redundancy

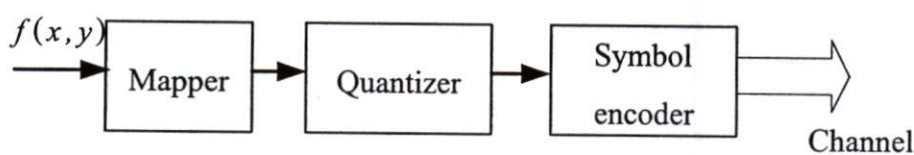
ในการออกแบบขบวนการของ Source Encoder นั้นลักษณะของการนำไปประยุกต์ใช้งานและคุณภาพของภาพ ที่ต้องการจะเป็นตัวกำหนดถึงวิธีการที่เหมาะสมจะนำมาใช้ เช่น ข้อมูลภาพที่ต้องการนำไปประมวลผลต่อไป ควรต้องใช้ขบวนการบีบอัดข้อมูลแบบที่ไม่มีความผิดพลาด และสำหรับในการใช้งานทั่วไปแล้วนั้น ภาพอาจมีความผิดพลาดได้บ้าง ตามแต่คุณภาพของภาพที่ต้องการเป็นต้น

โดยส่วนใหญ่แล้วสำหรับการใช้งานทั่วไป ส่วนของ Source Encoder จะประกอบไปด้วยสาม ส่วนใหญ่ ดังแสดงในภาพที่ 2.5 (a) และสำหรับส่วนของ Source Decoder จะแสดงในภาพที่ 2.5 (b)

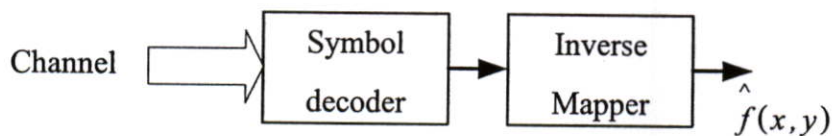
จากองค์ประกอบของ Source Encoder นี้ ในส่วนแรก ซึ่งเรียกว่า Mapper จะทำหน้าที่ในการแปลงข้อมูลภาพที่เข้ามาให้อยู่ในรูปแบบที่เหมาะสมในการลด Interpixel Redundancy

โดยที่การทำ Mapping นี้ อาจเป็นการลด Interpixel Redundancy โดยตรง หรือไม่ก็จะเป็นการทำให้ ส่วนที่เป็น Interpixel Redundancy นี้สามารถที่จะเข้าถึงหรือ ลดลงได้ โดยขบวนการขั้นต่อ ๆ ไป โดยทั่วไปแล้วขบวนการส่งถ่ายนี้จะเป็นขบวนการที่สามารถกระทำย้อนกลับได้ (Reversible)

ข้อมูลที่ได้หลังจากที่ผ่านขบวนการส่งถ่ายแล้ว อาจมีขนาดที่ลดลงหรือไม่ก็ได้ สำหรับตัวอย่างของขบวนการส่งถ่ายเช่น ขบวนการของ Run-length Coding ซึ่งเป็นวิธีการส่งถ่ายวิธีหนึ่ง ที่เป็นการลดจำนวนของข้อมูลลงโดยตรง หรือขบวนการของการแปลงข้อมูลด้วย DCT (Discrete Cosine Transform) และ DWT (Discrete Wavelet Transform) ซึ่งข้อมูลที่ผ่านการแปลงแล้ว จะมีจำนวนเท่าเดิม แต่รูปแบบการเรียงของข้อมูลจะเปลี่ยนไปให้ง่ายต่อการจัดการลด Interpixel Redundancy ตามที่ต้องการ



(a) Source encoder



(b) Source decoder

ภาพที่ 2.5 (a) แสดงขบวนการของ Source Encoder และ (b) ขบวนการของ Source Decoder

สำหรับในส่วนที่สองของ Block Diagram ในภาพที่ 2.5(a) หรือ Quantizer ซึ่ง จะเป็นการลดความละเอียดของข้อมูลที่ได้จากการส่งถ่ายแล้ว การลดความละเอียดของข้อมูลนี้เป็นผลให้ข้อมูลมีความผิดพลาดเกิดขึ้น ขบวนการ Quantization จะเป็นการลด Psychovisual Redundancy ดังในหัวข้อที่ผ่านมาแล้ว ซึ่งเป็นขบวนการที่ไม่สามารถกระทำย้อนกลับได้ (Irreversible) ดังนั้น ถ้าต้องการบีบอัดข้อมูลภาพแบบที่ไม่มี ความผิดพลาด ส่วนของการทำ Quantization นี้จะไม่มี

และสำหรับส่วนสุดท้าย หรือ Symbol Coder ในส่วนนี้จะเป็นการแทนข้อมูลที่ได้จากการ Quantize ด้วยรหัสที่เหมาะสม ซึ่งขบวนการนี้จะเป็นการลด Coding Redundancy นั้นเอง โดยทั่วไปแล้วการใช้ Variable - length Coding จะเป็นวิธีการที่มีประสิทธิภาพ เนื่องจากมันจะใช้รหัสที่สั้นที่สุด แทนข้อมูลที่พบเป็นจำนวนมาก ซึ่งจะทำให้จำนวนของรหัสที่ต้องใช้แทนข้อมูลทั้งหมดนี้ลดลง

หลังจากที่ข้อมูลภาพ ผ่านขบวนการทั้งสามนี้แล้ว ความซ้ำซ้อนกัน ที่มีในข้อมูลภาพจะลดลง และจำนวนของรหัสข้อมูลทั้งหมดจะลดลงด้วย

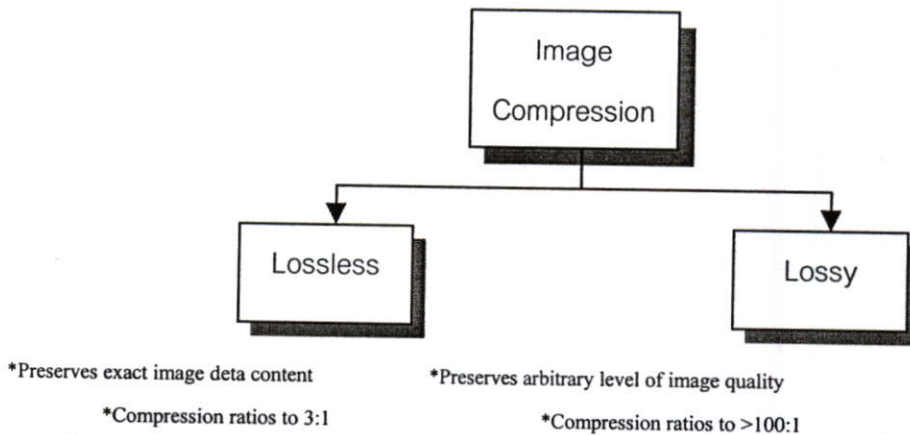
2.2.2 ส่วนการถอดรหัส

เมื่อต้องการนำข้อมูลภาพนี้มาใช้งาน จะกระทำได้โดยใช้ขบวนการของ Source Decoder ในการสร้างข้อมูลภาพกลับออกมา ซึ่งจะมีขบวนการดังแสดงในภาพที่ 2.5 (b) โดยจะประกอบด้วย ส่วนย่อย ๆ เพียงสองส่วนคือ

ในส่วนแรก หรือ Symbol Decoder จะทำหน้าที่ในการแปลงรหัส กลับออกมาเป็นข้อมูล จากนั้นข้อมูลนี้จะผ่านไปยังส่วนที่สองหรือ Inverse Mapper ซึ่งจะทำการแปลงข้อมูลนี้ กลับออกมาเป็นข้อมูลของภาพที่ต้องการ จะเห็นได้ว่าในขบวนการของ Source Decoder นี้จะเป็นขบวนการที่กระทำย้อนกลับของ Source Encoder แต่จะไม่มีส่วนของการทำ Inverse Quantization เพราะว่า ขบวนการ Quantization นี้เป็นขบวนการที่ไม่สามารถกระทำย้อนกลับได้

2.3 การจำแนกประเภทของการบีบอัดข้อมูลภาพ

โดยทั่วไปรูปแบบการบีบอัดข้อมูลภาพ สามารถแบ่งเป็น 2 ประเภทดังนี้



ภาพที่ 2.6 ประเภทของการบีบอัดข้อมูลภาพ

1) การบีบอัดข้อมูลแบบไม่มีการสูญเสีย (Lossless compression) เป็นวิธีการที่ข้อมูลภาพเมื่อผ่านขั้นตอนการบีบอัดแล้วผลลัพธ์ของข้อมูลภาพที่ได้จะเหมือนภาพต้นแบบ (Original image) ทุกประการ นิยมใช้กับข้อมูลที่มีความสำคัญมากและสูญเสียไม่ได้ เช่น ข้อมูลภาพที่ใช้วินิจฉัยทางการแพทย์ เป็นต้น

2) การบีบอัดข้อมูลแบบมีการสูญเสีย (Lossy compression) เป็นวิธีการที่ผลลัพธ์ของข้อมูลภาพที่ได้จากการขยายข้อมูลภาพกลับมามีความแตกต่างจากภาพต้นแบบ โดยยอมให้มีการสูญเสียข้อมูลได้บ้าง นิยมใช้ในการประมวลผลภาพทั่วไปเป็นอย่างมาก เนื่องจากตาของคนเราไม่สามารถตรวจจับ

ข้อมูลในส่วนที่มีการสูญเสียได้ อย่างไรก็ตามคุณภาพของภาพที่ได้มีความสัมพันธ์กับอัตราการบีบอัดข้อมูลด้วย(Compression ratio) นั่นคือ ข้อมูลภาพมีอัตราการลดขนาดที่สูง จะได้ขนาดของข้อมูลภาพที่มีขนาดเล็ก แต่ทำให้คุณภาพของภาพที่ได้แย่ลงไปด้วย

วิธีการบีบอัดข้อมูลภาพมีหลายวิธี แต่ละวิธีมีความเหมาะสมในลักษณะของการนำไปใช้งานที่แตกต่างกัน สามารถจำแนกความแตกต่างของวิธีการบีบอัดข้อมูลภาพได้ดังตารางที่ 2.2

ตารางที่ 2.2 การจำแนกการเข้ารหัส และวิธีการบีบอัดข้อมูลแบบต่างๆ

Entropy Coding	Run-length Coding	
	Huffman Coding	
	Arithmetic Coding	
Source Coding	Prediction	DPCM
		DM
	Transformation	FFT
		DCT
	Layered Coding	Bit Position
		Subsampling
		Sub-band Coding
Vector Quantisation		
Hybrid Coding	JPEG	
	MPEG	
	H.261	
	DVI RTV,DVI PLV	

2.3.1 การเข้ารหัสเอนโทรปี (Entropy coding)

การเข้ารหัสแบบนี้จะไม่คำนึงถึงคุณสมบัติเฉพาะของข้อมูลที่นำมาผ่านขบวนการ โดยสายข้อมูลจะถูกบีบอัดข้อมูลโดยการใช้ลำดับของข้อมูลทางดิจิทัล การเข้ารหัสแบบนี้เป็นการเข้ารหัสแบบไม่มีการสูญเสีย ได้แก่

2.3.1.1 การเข้ารหัสแบบรันเลงจ์ (Run-length coding)

การบีบอัดข้อมูลภาพด้วยวิธีรันเลงจ์ (Runlength Compression) นั้น จะอาศัยลักษณะโดยทั่วไปของภาพที่จะต้องมีส่วนของฉากหลัง (Background) และพื้นหน้า (Foreground) ซึ่งในส่วนของฉากหลังนั้นจะมีรายละเอียดของภาพไม่มากนัก ส่วนนี้เองจะมีการเปลี่ยนแปลงของข้อมูลค่อนข้างน้อยเมื่อเทียบกับส่วนของพื้นหน้าซึ่งมีรายละเอียดและการเปลี่ยนแปลงของข้อมูลมากกว่า โดยที่ส่วนที่

มีการเปลี่ยนแปลงของข้อมูลน้อยๆ นี้เองที่เราสามารถที่จำนำวิธีการเข้ารหัสแบบรันเลขจ้มาประยุกต์ใช้งานได้อย่างมีประสิทธิภาพ

การเข้ารหัสแบบนี้ จะจัดข้อมูลของภาพแบบเดิมให้อยู่ในรูปของคู่ลำดับ (G,L) โดยที่ G แทนระดับความเข้ม หรือระดับสีเทา และ L แทนความยาวของข้อมูลหรือ จำนวนจุดที่มีระดับสีเทา G ซึ่งการเข้ารหัสแบบนี้มีอยู่ด้วยกัน 2 วิธีใหญ่ ๆ คือ

1) จะทำโดยทำการอ่านข้อมูลเข้ามาโดยนับจำนวนข้อมูลที่ซ้ำกันกับข้อมูลนั้นเข้ามาด้วย แล้วแปลงข้อมูลที่รับเข้าไปเป็น 2 ไบต์ คือ ไบต์แรกจะเก็บจำนวนตัวข้อมูลที่ซ้ำกัน และจะมีไบต์ที่สองเก็บค่าของระดับสีที่ซ้ำกันนั่นเองไว้ โดยใน 1 ชุด ข้อมูลเอาท์พุท (2 ไบต์) จะนับจำนวนจุดที่ซ้ำกันไว้ 256 จุดสี ซึ่งจะสามารถยกตัวอย่างของการเข้ารหัสข้อมูลแบบรันเลขจ้ได้ เช่น

สมมติข้อมูลที่จะนำเข้ารหัสแบบรันเลขจ้มีค่าเป็น (ค่าที่แสดงเป็นตัวเลขฐาน 16)

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

จะเห็นได้ว่าเราสามารถลดข้อมูลจาก 19 ไบต์ เหลือ 12 ไบต์ได้ ซึ่งหากว่าข้อมูลมีค่าที่ซ้ำกันถึง 256 จุดแล้วก็จะทำให้เราลดขนาดของข้อมูลลงได้อย่างมหาศาล แต่ในขณะเดียวกันหากเกิดกรณีที่แย่ที่สุดของการใช้วิธีนี้ซึ่งก็คือ กรณีที่ข้อมูลแต่ละค่าไม่ซ้ำกันกับค่าข้างเคียงเลย ซึ่งจะทำให้ผลของการเข้ารหัสจะได้รหัสที่ยาวเป็นสองเท่าของข้อมูลอินพุท

2) นั้นเกิดเนื่องมาจากความบกพร่องของวิธีการทำรันเลขจ้เอ็น โค้ดคิงวิธีแรก ตรงที่โปรแกรมจะทำการเข้ารหัสข้อมูล เมื่อนับจำนวนข้อมูลที่ได้ตั้งแต่ 1 ถึง 255 จุด ดังนั้นถ้าเกิดข้อมูลของเรามีจุดสีที่มีค่าโคด ๆ (ไม่ซ้ำกับจุดข้างเคียงเลย) อยู่มากมายจะทำให้การเข้ารหัสไม่ทำให้ข้อมูลเล็กลงมากนัก จึงได้มีแนวคิดที่จะทำการแก้ไขปัญหาของการทำรันเลขจ้เอ็น โค้ดคิงของวิธีแรกขึ้นมา ซึ่งมีหลักการอยู่ 2 ข้อดังนี้

2.1) จะไม่ทำการเข้ารหัสข้อมูลกับส่วนที่มีจำนวนข้อมูลซ้ำกันน้อยกว่า 3 ค่า

2.2) ให้มีการทำเครื่องหมายเพื่อที่จะแยกส่วนที่มีการลดขนาดของข้อมูลกับส่วนที่ไม่มีการลดขนาดของข้อมูลออกจากกัน โดยใช้บิตๆ หนึ่งเป็นตัวบอกว่าข้อมูลนั้นมีการเข้ารหัสหรือไม่

จากหลักการดังกล่าวนี้ จะใช้บิตที่ 7 ของไบต์ที่ใช้ในการบอกขนาด เป็นตัวบอกว่าข้อมูลต่อจากนี้ไปมีการเข้ารหัสของข้อมูลหรือไม่ ส่วนบิตที่เหลือก็ยังคงใช้ในการบอกขนาดต่อไป เมื่อเราทำการนับพิกเซลที่มีค่าซ้ำกันได้ตั้งแต่ 3 ค่าขึ้นไป บิตที่ 7 ของไบต์บอกขนาดจะถูกเซตให้เป็น 1 และเราจะเริ่มทำการนับ 1 ตั้งแต่พิกเซลที่ซ้ำกันตั้งแต่ตัวที่ 4 เป็นต้นไป ทำให้บิตบอกขนาดนั้นมีค่าได้ตั้งแต่ 3 ถึง 130 หรืออาจจะบอกได้ว่าใน 2 ไบต์ของการเข้ารหัสข้อมูลนี้เราอาจจะสามารถทำการเก็บข้อมูลได้ถึง 130 ไบต์ ส่วนในกรณีที่ไม่มีมีการเข้ารหัสนั้น บิตที่ 7 ของไบต์ของขนาด ก็จะถูกเซตให้เป็น 0 และ 7 บิตที่เหลือจะบอกจำนวนของข้อมูลที่มีค่าไม่ซ้ำกันนั้น โดยค่าที่ตามมาจะเป็นข้อ

มูลที่ไม่มีการเข้ารหัส ซึ่งจะมีค่าระหว่าง 0 ถึง 127 โดยจำนวนจริงจะมีค่าเท่ากับไบนารีออกขนาดบวกรวมหนึ่ง ดังตัวอย่างเช่น

ค่าของข้อมูลที่จะนำมาทำการเข้ารหัสเป็น (ค่าที่แสดงเป็นเลขฐาน 16)

AC AC AC AC 15 15 15 15 15 15 45 45 78 20 10 10 10 10 ซึ่งจะทำการเข้ารหัสได้เป็น 81 AC 83 15 03 45 45 78 20 81 10 ซึ่งความหมายก็คือ 81 AC หมายถึงมีการเข้ารหัส 4 ไบนารีที่มีค่าเท่ากับ AC, 83 15 หมายถึง มีการเข้ารหัส 8 ไบนารีที่มีค่าเท่ากับ 15, 03 45 45 78 20 หมายถึงมีการเข้ารหัส 4 ไบนารี ซึ่งค่าที่นำมาเข้ารหัสมีค่าเท่ากับ 45 45 78 และ 20, 81 10 หมายถึงมีการนำข้อมูลมาเข้ารหัสจำนวน 4 ไบนารี ซึ่งข้อมูลที่นำมาเข้ารหัสนั้นที่ค่าเท่ากับ 10

การลดขนาดของข้อมูลลงด้วยวิธีการรันเลงจั้นนั้นจะไม่มี ความคลาดเคลื่อนของภาพเกิดขึ้นภายหลังจากที่ทำการถอดรหัส การใช้วิธีนี้ต้องพิจารณาถึงลักษณะของภาพต้นแบบว่ามีความเหมาะสมหรือไม่ ซึ่งภาพที่เหมาะสมควรมีรายละเอียดของภาพน้อย เช่น ภาพเอกสาร ภาพจากกราฟจากตาราง และเราสามารถที่จะเพิ่มประสิทธิภาพของการเข้ารหัสข้อมูลได้ โดยการปรับปรุงข้อมูลให้มีความแตกต่างกันของระดับสีเทาให้น้อยลงได้

ข่าวสาร ปริมาณข่าวสารเฉลี่ยและส่วนที่ซ้ำ

ข่าวสาร (Information) ในความหมายของทฤษฎีข่าวสารจะสามารถกล่าวได้ว่า ข่าวสารคือ ปริมาณความไม่แน่นอนของเหตุการณ์ (Measure of uncertainty) เช่น สำหรับเหตุการณ์ (Event) ที่เกิดขึ้นก่อนข้างแน่นอนเหตุการณ์นั้นจะมีค่าทางข่าวสารน้อย แต่สำหรับเหตุการณ์ที่เกิดขึ้นไม่มากครั้งนั้นจะมีคุณค่าทางข่าวสารมากกว่า ซึ่งการวัดปริมาณข่าวสารของเหตุการณ์หนึ่ง ๆ นั้น จะสามารถคำนวณได้จาก

$$I_i = -\log_2 P_i \quad (\text{บิต}) \quad (2.6)$$

โดยที่ P_i คือความน่าจะเป็นในการเกิดเหตุการณ์ I

ปริมาณข่าวสารของเหตุการณ์ I สามารถที่จะบอกได้ว่าควรจะใช้พื้นที่ของหน่วยความจำในการเก็บเหตุการณ์ I เท่าไร ยกตัวอย่างเช่น เหตุการณ์ x มีค่าความน่าจะเป็นในการเกิดเท่ากับ 0.125 แล้วแสดงว่า พื้นที่ ๆ จะต้องใช้ในการเก็บเหตุการณ์ x มีค่าเท่ากับ 3 บิต ปริมาณข่าวสารเฉลี่ยของเหตุการณ์ทั้งหมด (Entropy : H) สามารถคำนวณได้จาก

$$H = -(P_1 \log_2 P_1 + P_2 \log_2 P_2 + \dots + P_M \log_2 P_M) \quad (2.7)$$

โดยที่ M คือ เหตุการณ์ที่เป็นไปได้ทั้งหมด

ปริมาณข่าวสารเฉลี่ยสามารถบอกเราได้ว่าควรเก็บข้อมูลทั้งหมดด้วยหน่วยบันทึกผลเท่าไร เช่น ข้อมูล 'e' มีค่าความน่าจะเป็นเท่ากับ 0.0625 ดังนั้นข้อมูล 'e' จะใช้พื้นที่ในการเก็บข้อมูลขนาด 20 บิต ซึ่งค่าของข่าวสารเฉลี่ยนั้นมีเงื่อนไขที่น่าสนใจอยู่ 2 กรณีคือ

1) เมื่อ $M=1$ คือมีเหตุการณ์เดียว ดังนั้น $P_i=1$ ทำให้ $H=0$ หมายความว่าไม่มีค่าของข่าวสารเฉลี่ยของเหตุการณ์ที่ทราบแน่นอนแล้ว

2) เมื่อ $P_i=0$ ทำให้ $H=0$ หมายความว่าไม่มีค่าของข่าวสารเฉลี่ยของเหตุการณ์ที่เป็นไปไม่ได้ นั่นก็คือ $H=-\log_2 P=0$ (เมื่อ P เข้าใกล้ 0)

ส่วนที่ซ้ำเกิดขึ้นเนื่องจาก การกระจายของข้อมูลไม่เท่ากัน คือค่าความน่าจะเป็นของเหตุการณ์ทุกเหตุการณ์นั้น ไม่เท่ากัน แต่ถ้การกระจายของข้อมูลทุกตัวเท่ากันแล้วจะได้ว่า $H=\log_2 M$ ซึ่ง H จะมีค่าสูงสุด แต่โดยทั่วไปแล้วข้อมูลนั้นมักมีการกระจายไม่เท่ากัน

ดังนั้นเทคนิคการบีบอัดข้อมูลนั้นก็จะใช้ประโยชน์จากการกระจายไม่เท่ากันของข้อมูลเพื่อทำการลดขนาดของข้อมูล โดยการเข้ารหัสข้อมูลด้วยรหัสที่เท่ากับ หรือต่างจากค่าเอนโทรปีน้อยที่สุด ซึ่งตัวอย่างการเข้ารหัสข้อมูลแบบนี้คือ การเข้ารหัสข้อมูลแบบฮัฟแมน (Huffman Coding) และการเข้ารหัสข้อมูลแบบชานอน ฟาโน (Shanon Fano Coding)

2.3.1.2 การเข้ารหัสฮัฟแมน (Huffman coding)

การทำงานของเข้ารหัสแบบนี้อยู่บนพื้นฐานที่ว่า ในประโยคหรือไฟล์ข้อมูลที่เป็นภาษาอังกฤษหนึ่ง ๆ จะมีการใช้อักษรภาษาอังกฤษซ้ำไปซ้ำมาเสมอ และบางตัวก็ไม่ได้ใช้บ่อยสักเท่าไร ซึ่งอักษรภาษาอังกฤษตัวหนึ่งจะถูกแทนด้วยตัวเลขฐานสองจำนวน 8 บิต สำหรับหลักการสำคัญของการลดข้อมูลแบบฮัฟแมนนี้ คือ การลดจำนวนบิตที่ใช้แทนตัวอักษรที่มีการใช้บ่อย ๆ และเพิ่มจำนวนบิตที่ใช้แทนตัวอักษรที่มีการใช้น้อย ๆ

ในปัจจุบันนี้มีโปรแกรมแอฟพลิเคชันหลาย ๆ โปรแกรม ที่ได้นำวิธีการลดขนาดของฮัฟแมนนี้ไปใช้ เช่น การบีบอัดข้อมูลแบบ MNP-5 ของโมเด็ม ซึ่งได้ใช้การบีบอัดข้อมูลของฮัฟแมนแบบไดนามิก การเข้ารหัสแบบชานอน-ฟาโน (Shanon-Fano Coding) การเข้ารหัสบางส่วนของโปรแกรม PKZIP และการทำงานในตอนท้ายของการบีบอัดข้อมูลรูปภาพแบบ JPEG (Joint Photographic Expert Group)

การสร้างรหัสของฮัฟแมนนั้นก่อนอื่นต้องทำการอ่านข้อมูลก่อนหนึ่งรอบ เพื่อเป็นการหาความถี่ของแต่ละข้อมูล จากนั้นทำการเรียงลำดับของข้อมูลที่มีความถี่มากอยู่ด้านหนึ่ง และข้อมูลที่มีความถี่น้อยอยู่อีกด้านหนึ่ง แล้วจึงกำหนดให้ข้อมูลแต่ละตัวคือ โหนด (Node) ใด ๆ ของทรี (Tree) แล้วจึงนำมาสร้างทรี ตามวิธีการดังต่อไปนี้ นำเอาข้อมูล หรือ โหนดที่มีความถี่น้อยที่สุดมาเป็น ไรต์ โหนด (Child node) ของพารেন্ট โหนด (Parent node) ซึ่งความถี่ของพารেন্ট โหนดนี้จะเท่ากับผลรวมของความถี่ของไรต์ โหนดทั้งสอง

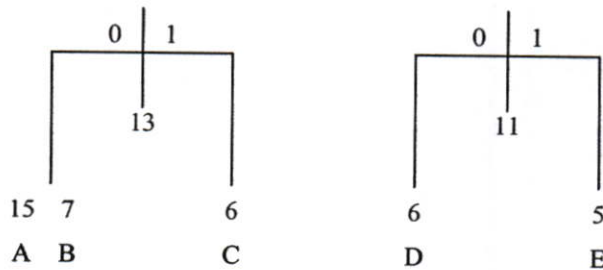
1) พารেন্টโนดจะถูกนำไปรวมกับโนดอื่น ๆ ที่เหลือและไซล์โนดที่ถูกใช้แล้วทั้งสองโนด จะไม่ถูกนำมาใช้อีก

2) ไซล์โนดหนึ่งจะถูกกำหนดให้เป็นบิต 0 และอีกโนดหนึ่งให้เป็นบิต 1

3) กระทำซ้ำตั้งแต่ 1. จนกระทั่งเหลือโนดเดียวที่เรียกว่ารากของทรี ยกตัวอย่างเช่นข้อมูลมีลักษณะดังนี้

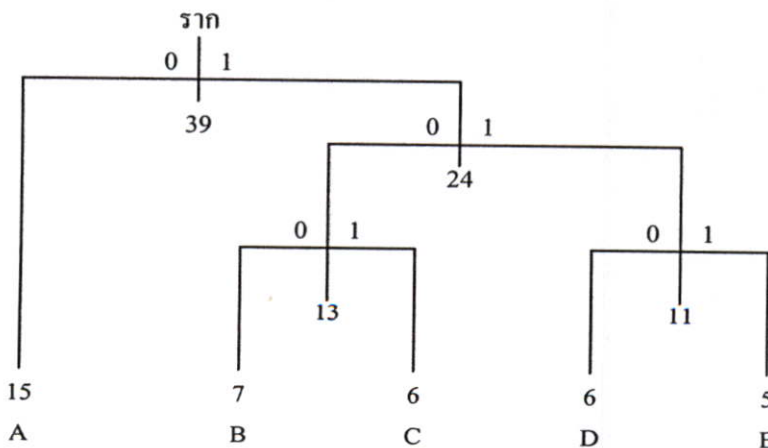
'AEDBAACBADCBECADACEDADACAEBABABEADABAD'

ความถี่ของข้อมูล A=15, B=7, C=6, D=6, E=5 โหนดที่มีน้ำหนักน้อยที่สุดคือ D และ E ทั้งสองโนดจะถูกเชื่อมให้เป็นพารেন্টโนดซึ่งมีน้ำหนักเท่ากับ 11(6+5) แล้วเราจึงกำหนดให้ โหนด D มีค่าเป็น 0 และ โหนด E มีค่าเป็น 1 รอบที่สอง B และ C จะถูกนำมาเชื่อมเป็นพารেন্টโนดใหม่อีก ซึ่งมีน้ำหนักเป็น 13 (7+6) ซึ่งจะเป็นดังภาพที่ 2.7



ภาพที่ 2.7 ฮัฟแมนทรีเมื่อผ่านไป 2 รอบ

รอบต่อมา 2 โหนดที่มีน้ำหนักน้อยที่สุด คือพารেন্টโนดของ B/C กับ D/E จะถูกเชื่อมกันเป็นพารেন্টโนดใหม่ หลังจากนั้นจะเหลือเพียง 2 โหนด เมื่อนำมารวมกันจะได้รากของทรีซึ่งมีลักษณะดังภาพที่ 2.8



ภาพที่ 2.8 แผนภาพของเมนทรี

การหารหัสของข้อมูลกระทำได้โดยการเดินทางไปยังเส้นทางที่ข้อมูลตัวนั้นผ่าน เมื่อผ่านโนดใดก็ให้จำเอาไว้ ซึ่งรหัสของข้อมูลก็คือค่าของโนดที่จำไว้ตามลำดับ ดังนั้นรหัสของข้อมูลคือ

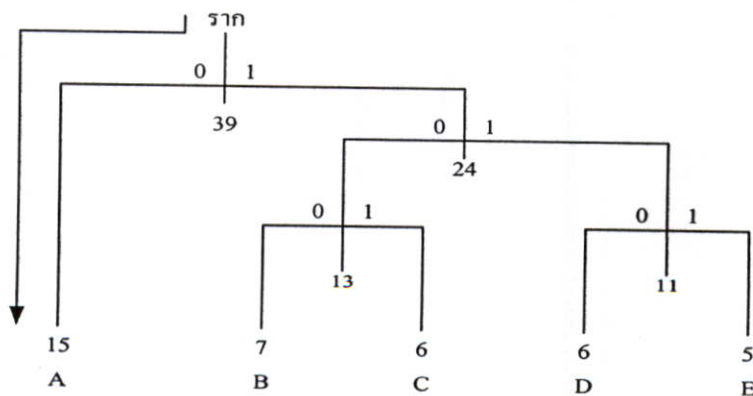
- A = 0
- B = 100
- C = 101
- D = 110
- E = 111

หลังจากที่ได้รหัสของข้อมูลแล้วจึงทำการอ่านข้อมูลอีกรอบเพื่อที่จะทำการลดขนาดของข้อมูลทั้งหมด โดยการแทนข้อมูลด้วยรหัสของข้อมูล ดังนั้นจะได้รหัสของข้อมูลเรียงกันในลักษณะเป็นลำดับเช่นเดียวกันกับลำดับของข้อมูล ดังภาพที่ 2.9

“0 111 110 100 0 0 101.....” รหัสของข้อมูลมีลักษณะเป็นบิต
 “A E D B A A C.....” ข้อมูลมีลักษณะเป็นไบต์

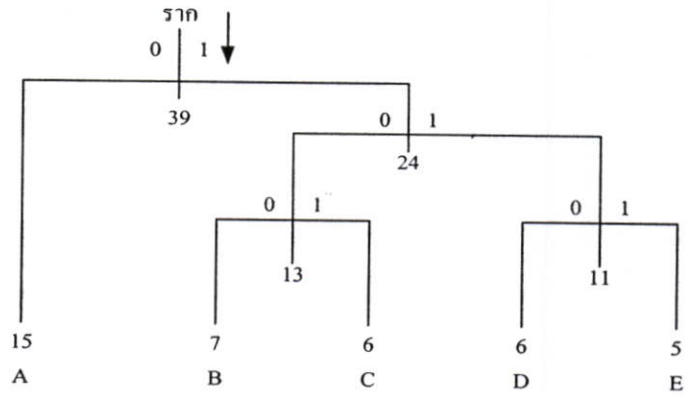
ภาพที่ 2.9 ตัวอย่างการเข้ารหัสข้อมูล

สำหรับการขยายข้อมูลกลับจะทำได้โดย การสร้างทรีขึ้นมาก่อน โดยที่ทรีนี้คือทรีอันเดียวกับที่ใช้ในการเข้ารหัสข้อมูล โดยจะทำการอ่านรหัสเข้ามาทีละบิตและเริ่มเดินทางจากรากไปยังกิ่งที่มีค่าเท่ากับบิต 1 บิตที่ป้อนเข้ามา และกระทำเช่นนี้ต่อไปเรื่อย ๆ โดยย่อลงไปอีกก็จะได้ข้อมูลกลับคืนมา 1 ตัว หลังจากนั้นก็เริ่มเดินทางจากรากใหม่จนกระทั่งได้ข้อมูลกลับคืนมาจนครบหมดจากรหัสของข้อมูลข้างต้น ค่าของบิตแรกคือ 0 ดังนั้น จึงสามารถที่จะถอดรหัสออกมาได้เป็น A ดังภาพที่ 2.10



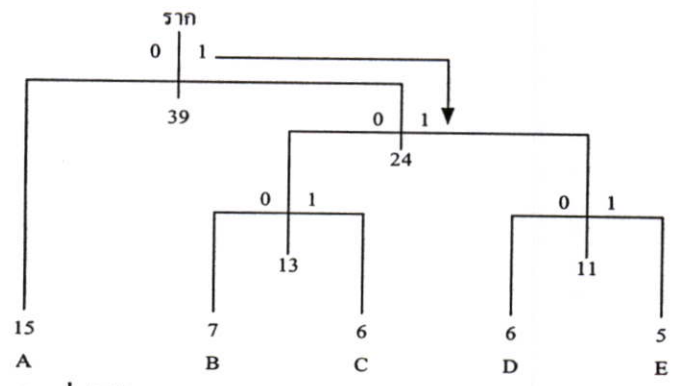
รูปที่ 2.10 การถอดรหัสเมื่อได้รับรหัส '0'

จากนั้นเมื่ออ่านรหัสข้อมูลในบิตที่ 2 คือ 1 เมื่อเดินทางจากรากไปยังทรีจะมีลักษณะดังภาพที่ 2.11



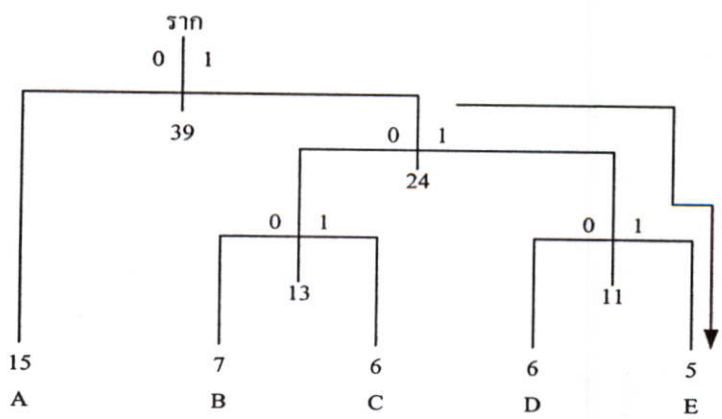
ภาพที่ 2.11 การถอดรหัสเมื่อได้รับรหัส '1'

จากนั้นจะทำการอ่านข้อมูลในบิตที่ 3 คือ 1 เมื่อเดินทางจากรากไปยังทรีจะมีลักษณะตามภาพที่ 2.12



ภาพที่ 2.12 การถอดรหัสเมื่อได้รับรหัส '11'

จากนั้นจะทำการอ่านข้อมูลในบิตที่ 4 คือ 1 เมื่อเดินทางจากรากไปยังทรีจะมีลักษณะตามภาพที่ 2.13



ภาพที่ 2.13 การถอดรหัสเมื่อได้รับรหัส '111'

ก็จะสามารถถอดรหัสได้เป็น E และจะกระทำเช่นนี้ต่อไปจนกระทั่งไม่มีข้อมูลเหลือ จึงจะได้ข้อมูลเดิมกลับมาทุกประการ

2.3.1.3 การเข้ารหัสเลขคณิต (Arithmetic Coding)

ในการเข้ารหัสเลขคณิตนั้นข่าวสารต้นแบบจะถูกแทนด้วยช่วงระยะห่างของจำนวนจริงระหว่าง 0 และ 1 โดยข้อมูลแต่ละตัวของข่าวสารจะทำให้ช่วงระยะห่างนั้นแคบลง ซึ่งระยะห่างที่เล็กลงนี้จะทำให้จำนวนบิตมากขึ้น ตัวข้อมูลที่ต่อเนื่องกันของข่าวสารจะลดขนาดของระยะห่างตามความน่าจะเป็นของข้อมูล ข่าวสารที่มีความน่าจะเป็นสูงจะทำให้ระยะห่างถูกลดลงได้น้อยกว่าข่าวสารที่มีความน่าจะเป็นต่ำ ดังนั้นจึงทำให้ข่าวสารที่มีความน่าจะเป็นสูงใช้จำนวนบิตในการเข้ารหัสน้อยกว่า

ตารางที่ 2.3 แสดงการแบ่งขอบเขตตามความน่าจะเป็นสะสม

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

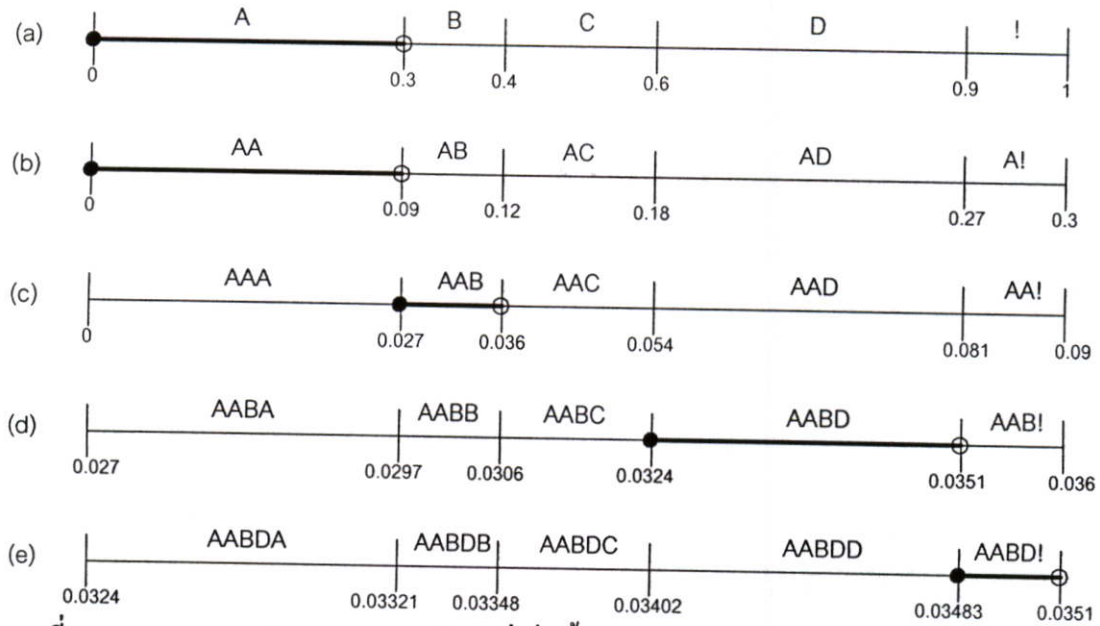
ขบวนการจะเริ่มจากข้อมูลต้นแบบที่ไม่ได้มีการเรียงลำดับความน่าจะเป็น จากนั้นจะทำการแบ่งช่วงระยะห่างระหว่าง 0 กับ 1 เป็นช่วงย่อยๆ ตามความน่าจะเป็นสะสม (cumulative probability) อาทิเช่น กำหนดให้ข้อมูลต้นแบบคือ {A,B,C,D,!} มีความน่าจะเป็นเป็น {0.3,0.1,0.2,0.3,0.1} ตามลำดับ การแบ่งช่วงระยะห่างระหว่าง 0 กับ 1 แสดงไว้ในตารางที่ 2.3

ตัวอย่างการเข้ารหัส AABD! แสดงไว้ในภาพที่ 2.14 โดยมีขั้นตอนดังต่อไปนี้

- 1) A ตัวแรกจะลดขอบเขตของระยะห่างระหว่าง 0 กับ 1 ลงเป็น [0,0.3) ตามภาพที่ 2.14 (a)
- 2) A ตัวที่สองจะลดขอบเขตของระยะห่างระหว่าง 0 กับ 0.3 ลงเป็น [0,0.09) ตามภาพที่ 2.14 (b)
- 3) B จะลดขอบเขตของระยะห่างระหว่าง 0 กับ 0.09 ลงเป็น [0.027,0.036) ตามภาพที่ 2.14 (c)
- 4) D จะลดขอบเขตระยะห่างระหว่าง 0.027 กับ 0.036 ลงเป็น [0.0324,0.0351) ตามภาพที่ 2.14 (d)

5) ! จะลดขอบเขตระยะห่างระหว่าง 0.0324 กับ 0.0351 ลงเป็น $[0.03483, 0.0351]$ ตามภาพที่ 2.14(e)

โดยที่ $[0, 0.3]$ หมายถึงระยะห่างที่อยู่ในช่วงมากกว่าหรือเท่ากับ 0 และ น้อยกว่า 0.3 เป็นต้น ดังนั้นข้อมูล AABD! สามารถจะเข้ารหัสได้โดยการแทนด้วยช่วงระยะห่างหรือตัวเลขจำนวนใดๆ ที่อยู่ในช่วง $[0.03483, 0.0351]$



ภาพที่ 2.14 แสดงการแบ่งขอบเขตตามข้อมูลที่เพิ่มขึ้น

สมมติว่าตัวถอดรหัสได้รับข้อมูลรหัสเป็น $i = 0.0350$ ดังนั้นขบวนการถอดรหัสจะมีขั้นตอนดังต่อไปนี้

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

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

5) แบ่งช่วง [0.0324,0.0351) ตามความน่าจะเป็นแล้วพบว่า $i = 0.0350$ อยู่ในช่วง [0.03483,0.0351) ซึ่งเป็นช่วงของตัวอักษร ! ดังนั้นจะได้ตัวอักษรตัวที่สามคือ ! จากนั้นจะลดขอบเขตของ [0.0324,0.0351) ลงเป็น [0.03483,0.0351)

ในทางปฏิบัติแล้วมีอยู่หลายปัจจัยที่ทำให้การเข้ารหัสและถอดรหัสเลขคณิตมีความซับซ้อนกว่าการเข้ารหัสฮัฟแมน แต่ก็จะสามารถลดข้อมูลที่มากกว่า

2.3.2 การเข้ารหัสซอร์ส (Source coding)

พิจารณาความสัมพันธ์ของข้อมูลหรือองค์ประกอบของข้อมูลเริ่มต้นเป็นหลัก โดยส่วนใหญ่เป็นการบีบอัดข้อมูลที่มีการสูญเสีย โดยการเข้ารหัสแบบนี้จะให้อัตราการบีบอัดข้อมูลที่สูงกว่าแบบการเข้ารหัสแบบเอนโทรปี ได้แก่

2.3.2.1 การเข้ารหัสด้วยการทำนาย (Predictive coding)

การบีบอัดข้อมูลภาพด้วยวิธีพรีดิกทีฟโค้ดดิ้ง (Predictive coding) นี้ เป็นการอาศัยคุณสมบัติของข้อมูลภาพที่มักจะมีค่าซ้ำ ๆ กัน และเมื่อข้อมูลอินพุตถูกกำหนดให้มีความเกี่ยวพันกัน นั่นก็คือจุดภาพที่มีตำแหน่งอยู่ใกล้ ๆ กัน มักจะมีค่าแอมพลิจูดที่ใกล้เคียงกันหรือเท่ากัน ดังนั้นจึงสามารถที่จะใช้ค่าของจุดภาพหนึ่งจุด หรือหลาย ๆ จุดที่ผ่านมาในไลน์ (Line) นั้น ไลน์ก่อนหน้านั้น หรือในเฟรมที่ผ่านมา เป็นตัวคาดคะเนหรือแทนค่าของจุดภาพปัจจุบัน ซึ่งโดยธรรมชาติทางสถิติของข้อมูลภาพ เราสามารถที่จะคาดคะเนค่าของข้อมูลได้โดยไม่ผิดพลาดมากนัก จากค่าที่ได้จากการคาดคะเนนี้เอาไปลบกับค่าจริงของจุดภาพ จะได้เป็นค่าความแตกต่างระหว่างค่าจริงกับค่าที่ได้ทำการคาดคะเนเอาไว้ ซึ่งค่านี้อาจมีขนาดเล็ก และค่าของผลต่างนี้จะถูกนำไปเข้ารหัสเพื่อที่จะเก็บไว้ใช้ในตอนถอดรหัสพร้อมกับค่าที่เราคาดคะเนเอาไว้ ดังนั้นในการที่จะเก็บค่าลงในหน่วยความจำหรือต้องการส่งไปในช่องทางส่งสัญญาณ ก็จะใช้ค่าทั้งสองค่านี้นี้ เมื่อถึงตอนที่ให้นำเอาภาพเดิมกลับมาหรือถึงขั้นตอนในการถอดรหัส ก็จะนำเอาค่าที่ได้คาดคะเนไว้ในตอนแรกบวกกับค่าของผลต่างของจุดภาพนั้น ก็จะได้ค่าของจุดภาพนั้น ๆ ซึ่งค่าความผิดพลาดเกิดขึ้นในตอนถอดรหัสนั้นเกิดขึ้นเพียงกรณีเดียวในขั้นตอนการจัดระดับสัญญาณ (Quantization) ของค่าความแตกต่างของแอมพลิจูดเท่านั้น โดยที่วิธีการนี้เป็นวิธีการที่ง่ายต่อการสร้างระบบ และสามารถที่จะทำการบีบอัดภาพลงให้เหลือประมาณ 1 – 2 bpp เท่านั้น

2.3.2.2 การเข้ารหัสด้วยการแปลง (Transform coding)

การบีบอัดข้อมูลภาพโดยวิธีการ ทรานส์ฟอร์มโค้ดดิ้ง (Transform coding) นั้นเป็นวิธีการที่มีความซับซ้อน และมีขั้นตอนที่มากกว่าการบีบอัดข้อมูลภาพด้วยวิธีการพรีดิกทีฟโค้ดดิ้ง หลักการของวิธีการทรานส์ฟอร์มโค้ดดิ้งนั้นจะทำการแปลงข้อมูลอินพุตที่อยู่ในสเปซเชิงโดเมน (Spatial

domain) ให้อยู่ในรูปของสเปกตรัมในฟรีควเอนซีโดเมน (Frequency domain) โดยอาศัยวิธีการแปลงข้อมูลรูปภาพในแบบต่าง ๆ เช่น การแปลงฟูริเยอร์แบบไม่ต่อเนื่อง (Discrete Fourier Transform : DFT) การแปลงโคไซน์แบบไม่ต่อเนื่อง (Discrete Cosine Transform : DCT) ซึ่งจะเป็นการเปลี่ยนแปลงข้อมูลที่อยู่ในรูปของสเปกตรัมโดเมน ให้อยู่ในรูปของสัมประสิทธิ์พลังงานที่ความถี่ต่าง ๆ โดยที่ความถี่ต่ำ ๆ จะมีพลังงานสูง ที่ความถี่สูงพลังงานจะลดลงไป การเข้ารหัสจึงใช้จำนวนบิตสำหรับแต่ละช่วงความถี่ที่ไม่เท่ากัน เมื่อต้องการอัตราการส่งข้อมูล (Bit rate) สูง ๆ ค่าของพลังงานความถี่สูงจะถูกตัดทิ้งไปเป็นส่วนใหญ่ เป็นเหตุให้รายละเอียดในส่วนที่เป็นขอบของภาพนั้นหายไป ทำให้ได้ภาพที่เบลอ ขาดความคมชัด

การแปลงที่นิยมใช้ในการบีบอัดข้อมูลภาพนั้นมีอยู่ด้วยกันหลายวิธี เช่น การแปลงฟูริเยอร์แบบรวดเร็ว (Fast Fourier Transform) การแปลงฮาดามาร์ดแบบรวดเร็ว (Fast Hadamard Transform) การแปลงโคไซน์แบบไม่ต่อเนื่องอย่างรวดเร็ว (Fast Discrete Cosine Transform) เป็นต้น ซึ่งในแต่ละวิธีก็มีข้อดี และข้อเสียแตกต่างกันออกไป ซึ่งวิธีการแปลงที่นิยมใช้กันมากได้แก่ การแปลงโคไซน์แบบไม่ต่อเนื่อง (Discrete Cosine Transform) เพราะเป็นวิธีที่สามารถคำนวณได้ง่าย เนื่องจากการคำนวณเฉพาะส่วนของค่าจริง โดยที่จะไม่มีค่าจินตภาพ

วิธีทรานส์ฟอร์มโค้ดดิ้งถึงแม้ว่าจะเป็นวิธีการที่ยู่ยากแต่ก็สามารถสร้างระบบได้ด้วยอุปกรณ์ทางฮาร์ดแวร์ (Hardware) คณิตศาสตร์ที่เร็วสูงได้ง่าย และเป็นระบบที่มีความยืดหยุ่น (Flexible) สามารถที่จะลดขนาดของข้อมูลให้ลดลงเหลือประมาณ 0.5 ถึง 1 bpp ซึ่งเป็นช่วงอัตราการลดลงของข้อมูลที่สามารถสร้างภาพเดิมกลับมาได้สมบูรณ์เพียงพอต่อการนำไปใช้งานในด้านต่าง ๆ

การเข้ารหัสเลเยอร์ (Layer coding) นั้นประกอบด้วย การเข้ารหัสด้วยบิตโพซิชั่น การเข้ารหัสด้วยการซัพแซมปลิง และการเข้าแบนด์ย่อย นอกจากนี้การเข้ารหัสซอร์สซิงประกอบด้วย การเข้ารหัสด้วยการควอนไทซ์แบบเวกเตอร์

2.3.3 การเข้ารหัสแบบไฮบริด (Hybrid coding)

ไฮบริดโค้ดดิ้ง (Hybrid coding) เป็นเทคนิคที่นำข้อดีของพรีดิกทีฟโค้ดดิ้งและ ทรานส์ฟอร์มโค้ดดิ้งมาใช้ร่วมกัน ทั้งนี้เพราะบางครั้งการบีบอัดข้อมูลภาพด้วยทรานส์ฟอร์มโค้ดดิ้งนั้นไม่สามารถที่จะให้รายละเอียดของภาพได้เพียงพอ ดังนั้นการเพิ่มรายละเอียดของภาพจึงต้องใช้ความสัมพันธ์ของจุดภาพที่มีอยู่ทั้งทางแนวนอน และทางแนวตั้ง โดยใช้การแปลงแบบมิตติเดียวในทิศทางแกนใดแกนหนึ่งของภาพ เช่น การแปลงกับข้อมูลภาพในทางแนวนอน ต่อจากนั้นจึงใช้พรีดิกทีฟโค้ดดิ้งกับสัมประสิทธิ์ของการแปลงที่ได้ เพื่อที่จะใช้ในการคาดคะเนค่าของกลุ่มของสัมประสิทธิ์ที่เหมือน ๆ กัน ซึ่งก็เหมือนกับการทำพรีดิกทีฟโค้ดดิ้งนั่นเอง และในกรณีของการทำการแปลงแบบ 2 มิติ สามารถที่จะทำได้เร็วขึ้นซึ่งอาจจะใช้การทำพรีดิกทีฟโค้ดดิ้ง กับค่าสัมประสิทธิ์ของการแปลงของภาพในบริเวณเดียวกัน แต่เป็นเฟรมที่แตกต่างกันที่อยู่ติด ๆ กัน การใช้ไฮบริดโค้ดดิ้ง ใน

ลักษณะนี้สามารถที่จะเป็นไปได้ในอีกลักษณะหนึ่งคือ ใช้การทำพรีดิคทีฟโค้ดดิ้งในขั้นตอนแรกก่อน แล้วจึงทำการแปลง แต่ในกรณีนี้ต้องการระบบที่ซับซ้อนมากกว่า

การบีบอัดข้อมูลภาพด้วยวิธีการไฮบริดโค้ดดิ้งนี้อาจจะพูดได้ว่ามีคุณสมบัติอยู่ระหว่างวิธีพรีดิคทีฟโค้ด และ ทรานส์ฟอร์มโค้ดดิ้ง ดังนั้นอัตราส่วนของการบีบอัดสัญญาณนั้นจึงไม่สามารถที่จะทำให้ต่ำกว่าวิธีทรานส์ฟอร์มโค้ดดิ้งได้ เพียงแต่สามารถสร้างได้ง่ายกว่า โดยที่อัตราการบีบอัดข้อมูลนั้นจะอยู่ที่ราว ๆ 1 bpp ซึ่งสามารถที่จะสร้างภาพเดิมกลับมาใหม่ได้คุณภาพของภาพที่ดีพอสมควร

2.4 เทคนิคการบีบอัดข้อมูลแบบต่างๆ

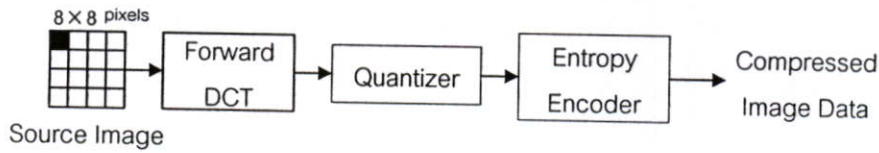
2.4.1 มาตรฐานการบีบอัดข้อมูลภาพนิ่ง

JPEG เป็นมาตรฐานการลดขนาดและขยายข้อมูลภาพนิ่งที่กำหนดขึ้นโดย Joint Photographic Expert Group ซึ่งเป็นความร่วมมือกันระหว่าง ISO (International Organization for Standardization) และ ITU (International Telecommunication Union) โดยมีวัตถุประสงค์ที่จะพัฒนามาตรฐานการบีบอัดข้อมูลภาพนิ่ง [2]

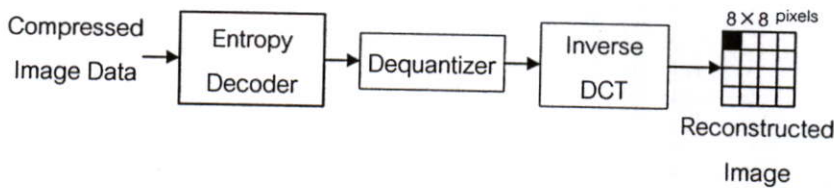
- 1) ผู้ใช้สามารถปรับเปลี่ยนอัตราส่วนของการบีบอัดข้อมูลภาพได้
- 2) สามารถนำไปใช้งานกับภาพดิจิทัลต่างๆได้
- 3) มีขั้นตอนการคำนวณที่ไม่ยุ่งยากซับซ้อนเพื่อนำไปใช้งานต่างๆได้อย่างกว้างขวาง
- 4) JPEG ประกอบด้วยการทำงาน 4 โหมดดังนี้
 - 4.1) การเข้ารหัสแบบ Sequential
 - 4.2) การเข้ารหัสแบบ Progressive
 - 4.3) การเข้ารหัสแบบ Lossless
 - 4.4) การเข้ารหัสแบบ Hierarchical

JPEG มีการบีบอัดข้อมูลแบบมีการสูญเสีย 3 โหมดและแบบไม่มีการสูญเสีย 1 โหมด ในหัวข้อนี้จะเน้นการบีบอัดข้อมูลภาพ JPEG ชนิด Sequential Baseline JPEG จะให้อัตราการบีบอัดข้อมูลภาพที่ดี เมื่อทำการบีบอัดข้อมูลภาพด้วยอัตราการลดขนาดที่ 20:1 ผลภาพที่ได้ไม่สามารถสังเกตเห็นความแตกต่างได้เมื่อเปรียบเทียบกับภาพต้นแบบ การบีบอัดข้อมูลภาพ JPEG สามารถกำหนดระดับคุณภาพของภาพด้วยค่า Q (Quality factor) โดยค่า Q ที่มีค่าสูงจะให้คุณภาพของภาพที่ได้สูง แต่จะมีขนาดของไฟล์ที่ใหญ่กว่าเมื่อเปรียบเทียบกับค่า Q ที่มีค่าต่ำกว่า ซึ่งจะให้ขนาดของไฟล์เล็กกว่าแต่คุณภาพของภาพที่ได้แย่กว่า และเป็นความสัมพันธ์พื้นฐานของคุณภาพและอัตราการบีบอัดข้อมูล ในการบีบอัดข้อมูลแบบมีการสูญเสีย JPEG มีขั้นตอนการบีบอัดข้อมูลแบบสมมาตร

(Symmetrical algorithm) เนื่องจากมีขั้นตอนการลดขนาดและขั้นตอนการขยายข้อมูลเท่ากันดังภาพที่ 2.15 ดังนั้นเวลาที่ใช้ย่อมเท่ากันด้วย



(a)



(b)

ภาพที่ 2.15 JPEG ชนิด Sequential Baseline (a) การเข้ารหัส (b) การถอดรหัส

การบีบอัดข้อมูลภาพ JPEG ชนิด Sequential Baseline ประกอบด้วยขั้นตอนการทำงานพื้นฐาน 5 ขั้นตอนดังนี้

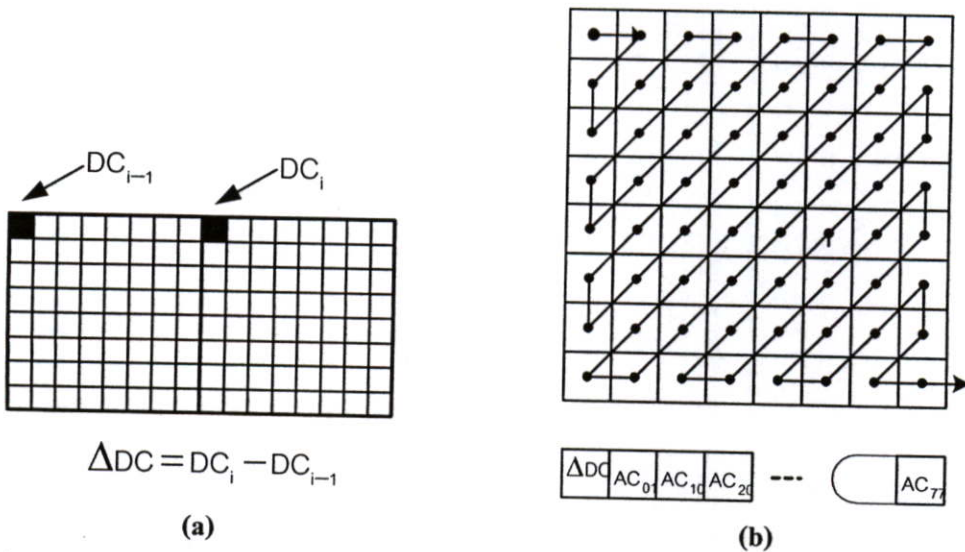
1) การแปลงระบบสี RGB ไปเป็นระบบสี YCbCr ยกเว้นในกรณีของภาพหลายระดับสีเทาเพื่อแยกเป็นองค์ประกอบของแสง (Luminance) และองค์ประกอบของสี (Chrominance) เนื่องจากเรามองเห็นของคนเรามีข้อจำกัดในการรับรู้การเปลี่ยนแปลงองค์ประกอบของแสงได้ดีกว่าองค์ประกอบของสี ทำให้ JPEG พยายามลดข้อมูลในส่วนที่เป็นองค์ประกอบของสีมากกว่าองค์ประกอบของแสง

2) การลดข้อมูลในส่วนที่เป็นองค์ประกอบของสี ขณะที่องค์ประกอบของแสงมีความละเอียดแบบเต็ม (Full resolution) การลดข้อมูล (Subsampling) ในส่วนที่เป็นองค์ประกอบของสีลง 2 พิกเซลในทิศทางแนวนอนและแนวตั้งเป็นรูปแบบที่นิยมมากอย่างหนึ่ง โดยการหาค่าเฉลี่ยของบล็อกข้อมูลขนาด 2x2 พิกเซล ขั้นตอนนี้จะมีการสูญเสียข้อมูลภาพและลดข้อมูลลงไปครึ่งหนึ่งของข้อมูลเดิม

3) การแบ่งข้อมูลออกเป็นบล็อกย่อยๆ ขนาด 8x8 พิกเซล และทำการแปลงข้อมูลให้อยู่ในโดเมนของความถี่ด้วยการแปลง Forward DCT ข้อมูลตัวแรกของบล็อกเรียกว่าค่า DC เป็นค่าเฉลี่ยของข้อมูลภายในบล็อกนั้นๆ และข้อมูลที่เหลือจำนวน 63 ค่าเรียกว่าค่า AC

4) การควอนไทซ์ค่าสัมประสิทธิ์ DCT โดยทำการควอนไทซ์บล็อกย่อยๆ ด้วยค่าสัมประสิทธิ์ของการควอนไทซ์และทำการปิดเศษค่าให้เป็นเลขจำนวนเต็ม ทำให้ข้อมูลมีค่าเป็นศูนย์มากเพื่อสะดวกในการเข้ารหัสต่อไปใน JPEG ไม่มีการกำหนดตารางการควอนไทซ์ที่แน่นอน

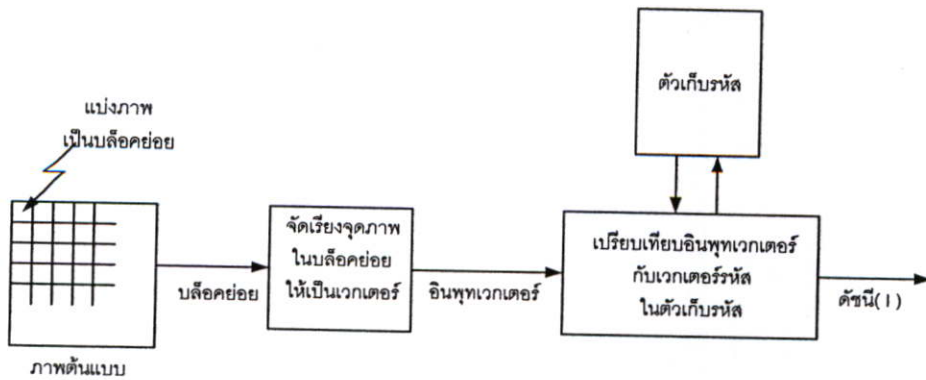
5) การเข้ารหัสเอนโทรปี ในที่นี้ใช้การเข้ารหัสแบบฮัฟแมน โดยค่า DC ใช้การเข้ารหัสของผลต่างระหว่างค่า DC ของบล็อกก่อนหน้าและบล็อกปัจจุบันเนื่องจากค่า DC ของบล็อกที่ใกล้เคียงกันจะมีความสัมพันธ์กันอย่างมาก สำหรับค่า AC จะเรียงลำดับแบบซิกแซก (Zigzag sequence) เพื่อให้ค่าสัมประสิทธิ์ที่มีความถี่ต่ำที่มีค่าส่วนใหญ่ไม่เป็นศูนย์อยู่ติดกันและค่าสัมประสิทธิ์ความถี่สูงที่มีค่าส่วนใหญ่เป็นศูนย์อยู่ติดกันเพื่อสะดวกในการเข้ารหัสแบบรันเลงจ้ดดังภาพที่ 2.16 (a) และ (b) ตามลำดับ



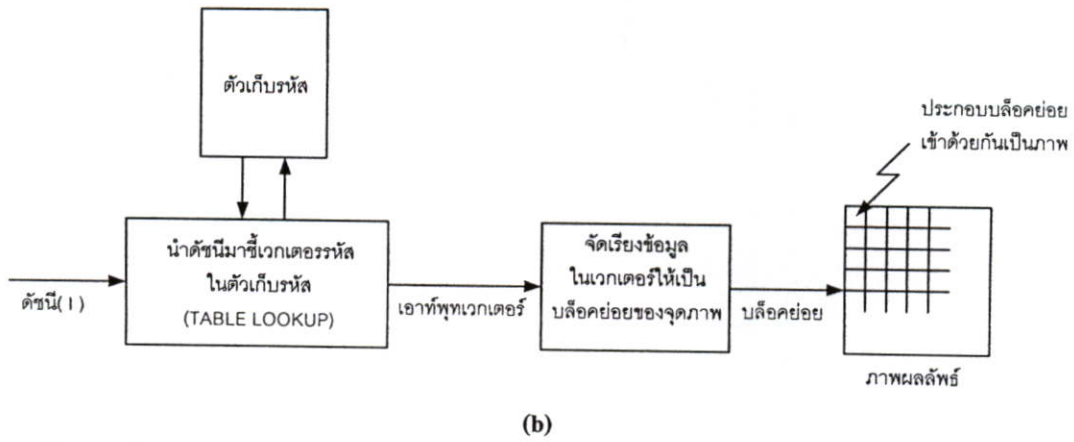
ภาพที่ 2.16 (a) การเข้ารหัสของค่า DC (b) ลำดับการซิกแซกของการเข้ารหัสเอนโทรปี

2.4.2 การบีบอัดข้อมูลภาพด้วยการเข้ารหัสการควอนไทซ์แบบเวกเตอร์ [3]

การควอนไทซ์แบบเวกเตอร์มีลักษณะเช่นเดียวกับ JPEG คือการแบ่งข้อมูลออกเป็นบล็อกย่อยๆ ขนาด $n \times n$ พิกเซล และอาศัยหลักการแทนที่บล็อกย่อยของข้อมูลด้วยดัชนีของชุดข้อมูลที่ใช้แทนบล็อกย่อยเหล่านั้น มักจะใช้เลขดัชนีเป็นเลขจำนวนนับและมีค่าจำกัดอยู่ระดับหนึ่ง



(a)



ภาพที่ 2.17 ขั้นตอนการควอนไตซ์แบบเวกเตอร์ (a) การเข้ารหัส (b) การถอดรหัส

ขั้นตอนในการเข้ารหัสการควอนไตซ์แบบเวกเตอร์นั้น ข้อมูลภาพจะถูกแยกองค์ประกอบให้เป็นเวกเตอร์ขนาด K มิติ เวกเตอร์เหล่านี้สามารถสร้างได้หลายวิธี เช่น การแบ่งข้อมูลภาพออกเป็นบล็อกย่อยๆ ขนาด $n \times n = K$ พิกเซล และทำการเรียงให้อยู่ในรูปเวกเตอร์ขนาด K มิติได้ เป็นต้น จำนวนบล็อกย่อยที่แบ่งได้พอดีจะเท่ากับจำนวนอินพุทเวกเตอร์ ในการเข้ารหัสจะเปรียบเทียบอินพุทเวกเตอร์กับเวกเตอร์รหัสทุกตัวที่ถูกเก็บอยู่ในตัวเก็บรหัส โดยเปรียบเทียบหาเวกเตอร์รหัสที่แทนด้วยเวกเตอร์ที่มีความเพี้ยนจากอินพุทเวกเตอร์น้อยที่สุด ผลลัพธ์ที่ได้ก็คือเลขดัชนีที่ใช้แทนเวกเตอร์รหัสที่มีความเพี้ยนน้อยที่สุด ดังภาพที่ 2.17 (a) ในขั้นตอนการถอดรหัสจะเป็นการนำเอาค่าดัชนีมาชี้ข้อมูลเวกเตอร์รหัสที่เก็บอยู่ในตัวเก็บรหัสเดียวกันกับขั้นตอนการเข้ารหัส ผลลัพธ์ที่ได้จะเป็นเอาท์พุทเวกเตอร์ จากนั้นนำเอาท์พุทเวกเตอร์ที่ได้มาจัดเรียงใหม่ให้เป็นบล็อกย่อยของภาพดังภาพที่ 2.17 (b)

2.4.3 การบีบอัดข้อมูลภาพแบบแฟคทอล (Fractal)

การบีบอัดข้อมูลภาพแฟคทอลจะใช้พื้นฐานของการแบ่งภาพออกเป็นส่วนย่อยๆ (Fractal) และกำหนดให้แต่ละส่วนย่อยที่มีลักษณะเหมือนกันหรือคล้ายกันแสดงในแต่ละพื้นที่ บล็อกที่มีขนาดเล็กกว่าจะทำการหาส่วนที่เหมือนกันภายในภาพได้ง่ายกว่า ภาพจะถูกแบ่งออกเป็นโดเมนและช่วงของบล็อกที่แตกต่างกัน แต่ละบล็อกของโดเมนหลังจากทำการแปลงข้อมูลจะมีช่วงของบล็อกที่ตรงกันหรือใกล้เคียงมากที่สุด

การบีบอัดข้อมูลภาพแฟคทอลเกี่ยวข้องกับการแปลงข้อมูลภาพซึ่งเป็นฟังก์ชันทางคณิตศาสตร์ ประกอบด้วย การหมุน (Rotation) การปรับขนาด (Scaling) การลาดเอียง (Skew) การเลื่อน (Translation) ตัวอย่างของการแปลงข้อมูลภาพ 2 มิติ สามารถหาได้จากสมการต่อไปนี้

$$W(x, y) = (ax + by + e, cx + dy + f) \quad (2.8)$$

หรืออยู่ในรูปของเมตริกซ์ได้ดังนี้

$$W \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} = \begin{bmatrix} ax + by + e \\ cx + dy + f \end{bmatrix} \quad (2.9)$$

เมื่อเมตริกซ์ $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ แสดงการหมุน การลาดเอียงและการเปลี่ยนขนาด ส่วนเมตริกซ์ $\begin{bmatrix} e \\ f \end{bmatrix}$ แสดงการเลื่อน การแปลงข้อมูลจะย้ายจากจุด (0,0) (1,0) (0,1) (1,1) ไปยัง (e,f) $(a+e,c+f)$ $(b+e,d+f)$ $(a+e,c+d+f)$ ตามลำดับ เมื่อค่า a b d และ f เป็นค่าสัมประสิทธิ์ของการแปลง

การบีบอัดข้อมูลภาพแฟคทอลประกอบด้วย การหาชุดการแปลงข้อมูลไปทางขวาสำหรับอัลกอริธึมการบีบอัดข้อมูลภาพแฟคทอลมีขั้นตอนการทำงานดังนี้

- 1) ในตอนแรกภาพจะแบ่งออกเป็นส่วนๆ โดยไม่ซ้อนทับกัน กลุ่มโดเมนของพื้นที่ที่ต้องครอบคลุมตลอดทั้งภาพสามารถมีขนาดหรือรูปร่างใดก็ได้
- 2) รวบรวมช่วงของพื้นที่ที่กำหนด ซึ่งมีขนาดใหญ่กว่าโดเมนของพื้นที่และสามารถซ้อนทับกันได้ตลอดทั้งภาพ
- 3) เลือกระดับของการแปลงข้อมูลที่ใช้งานในช่วงของพื้นที่ที่ใกล้เคียงกับพื้นที่ของโดเมนข้อมูลของการแปลง 3 มิติแต่ละค่าจะมีความสัมพันธ์กัน
- 4) จัดเก็บข้อมูลในรูปแบบของ FIF (Fractal Image Format) ประกอบด้วยส่วนหัว (Header) และรายละเอียด (Information) ของข้อมูลในการเลือกพื้นที่ตามด้วยกลุ่มรายชื่อของค่าสัมประสิทธิ์ที่เลือกในแต่ละพื้นที่ของโดเมน

ขั้นตอนการบีบอัดข้อมูลจะให้ไฟล์ข้อมูลที่ไม่ขึ้นกับความละเอียดของภาพต้นแบบ (Original image) การกำหนดค่าพารามิเตอร์จะระบุเป็นสมการสำหรับภาพนั้นๆ เช่น สมการเส้นตรง หาได้ดังนี้ $y=ax+b$ โดยทราบค่าสัมประสิทธิ์ของ a และ b เลยเส้นสามารถจะวาดที่ความละเอียดใดๆ ก็ได้ เช่นเดียวกับค่าสัมประสิทธิ์ของภาพ อัลกอริธึมการขยายข้อมูลภาพจะสร้างแบบจำลองของแฟคทอลหรือส่วนย่อยของภาพต้นแบบที่ความละเอียดใดๆ

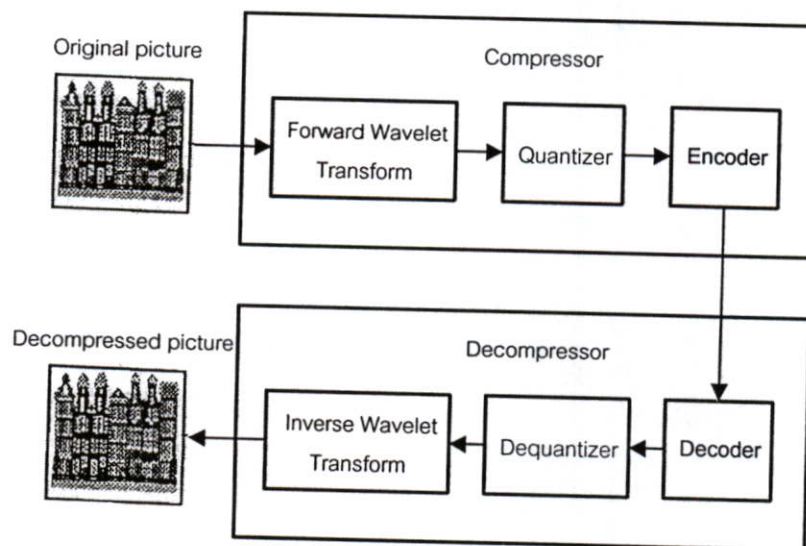
ขั้นตอนการขยายข้อมูลภาพเป็นขั้นตอนการทำซ้ำ ในแต่ละพื้นที่ของโดเมนจะแทนที่ด้วยข้อมูลที่ผ่านการแปลงจากช่วงของพื้นที่ที่เหมาะสมโดยใช้ค่าสัมประสิทธิ์ที่เก็บไว้สำหรับโดเมนนั้นๆ ขั้นตอนการลดขนาดและขยายข้อมูลภาพแฟคทอลเป็นลักษณะอสมมาตร (Asymmetric) นั่นคือต้องเวลาการคำนวณในขั้นตอนการลดขนาดมากกว่าการขยายข้อมูล

สรุปได้ว่าการบีบอัดข้อมูลภาพแฟคทอลมีลักษณะตรงข้ามกับ JPEG โดยที่การบีบอัดข้อมูลภาพแฟคทอลได้ให้การขยายข้อมูลภาพที่รวดเร็วและไม่ขึ้นกับความละเอียด นอกจากนี้ยังให้อัตรา

การบีบอัดข้อมูลที่สูง มีประโยชน์ในการใช้งานระบบมัลติมีเดียแบบสมมาตรที่ให้ความสำคัญกับการเข้าถึงข้อมูลภาพที่รวดเร็วได้

2.4.4 การบีบอัดข้อมูลภาพด้วยการแปลงเวฟเล็ต

ช่วงเวลา 7 ปีที่ผ่านมา การบีบอัดข้อมูลภาพและวิดีโอโดยใช้พื้นฐานการแปลงเวฟเล็ตได้กลายเป็นหัวข้อที่น่าสนใจในงานวิจัยเป็นอย่างมาก ได้มีการนำเสนออัลกอริทึมการบีบอัดข้อมูลภาพด้วยการแปลงเวฟเล็ตมากมาย และมีรูปแบบที่แตกต่างกันตั้งแต่การเข้ารหัสเอ็นโทรปี การควอนไทซ์แบบเวกเตอร์ การแปลงข้อมูลแบบปรับตัวเองได้ (Adaptive transformation) จนถึงการเข้ารหัสแบบต้นไม้ (Tree coding) และการเข้ารหัสโดยใช้พื้นฐานของขอบผล (Edge-based coding) บล็อกไดอะแกรมของอัลกอริทึมการบีบอัดข้อมูลภาพโดยใช้พื้นฐานการแปลงเวฟเล็ตแสดงดังภาพที่ 2.18 การลดขนาดโดยใช้พื้นฐานการแปลงเวฟเล็ต ประกอบด้วย 3 ขั้นตอนดังนี้ (1) FWT (Forward wavelet transform) (2) การควอนไทซ์ค่าสัมประสิทธิ์ของการแปลง (3) การเข้ารหัสค่าสัมประสิทธิ์การควอนไทซ์



ภาพที่ 2.18 การบีบอัดและการขยายข้อมูลภาพด้วยการแปลงเวฟเล็ต

2.5 การวัดประสิทธิภาพการบีบอัดข้อมูลภาพ

ในการวัดประสิทธิภาพของการบีบอัดข้อมูลในเชิงคณิตศาสตร์สามารถทำได้หลายวิธีการคือ

2.5.1 ค่าความผิดพลาดกำลังสองเฉลี่ย (Mean square error, MSE)

เราสามารถหาค่าผิดพลาดกำลังสองเฉลี่ยในการวัดประสิทธิภาพของการบีบอัดข้อมูล คือถ้าความผิดพลาดกำลังสองเฉลี่ยมีค่าน้อย แสดงว่าภาพที่ได้จากการแปลงกลับภาพที่ถูกบีบอัดข้อมูล

แล้วมีความผิดเพี้ยนไปจากภาพต้นแบบน้อย ซึ่งเราสามารถหาค่าความผิดพลาดกำลังสองเฉลี่ยของการบีบอัดข้อมูลภาพขนาด $M \times N$ ได้ดังนี้

$$MSE = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N [f(x,y) - \hat{f}(x,y)]^2 \quad (2.10)$$

โดยที่ M คือจำนวนจุดภาพตามความกว้างของภาพ

N คือจำนวนจุดภาพตามความสูงของภาพ

$f(x,y)$ คือ ระดับความเทา (Gray level) ที่ตำแหน่ง (x,y) ของภาพต้นแบบ

$\hat{f}(x,y)$ คือ ระดับความเทาที่ตำแหน่ง (x,y) ของภาพที่ได้สร้างกลับขึ้นมาใหม่

ในบางครั้งเราอาจจะพบว่ามีการใช้ค่ารากที่สองของความผิดพลาดกำลังสองเฉลี่ย (Root mean square error, $RMSE$) ในการวัดประสิทธิภาพการบีบอัดข้อมูลแทนค่า MSE ซึ่งค่า $RMSE$ สามารถหาได้จาก

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N [f(x,y) - \hat{f}(x,y)]^2} \quad (2.11)$$

2.5.2 SNR และ PSNR

อีกวิธีการหนึ่งของการอธิบายประสิทธิภาพของอัลกอริทึมคือ ค่าสัดส่วนสัญญาณต่อสัญญาณรบกวน (signal-to-noise ratio, SNR) ในบางครั้งก็เรียกว่า ค่าสัดส่วนสัญญาณต่อสัญญาณรบกวนการควอนไทซ์ (signal-to-quantization-noise ratio) ซึ่งในการวัดค่าจะมีความสัมพันธ์ของกำลังของสัญญาณกับความเพี้ยนของสัญญาณ สามารถหาได้จากการนำค่าเฉลี่ยของภาพต้นแบบยกกำลังสองแล้วหารด้วย MSE โดยมีหน่วยเป็นเดซิเบล (dB) ดังนี้

$$SNR = 10 \log_{10} \frac{\frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N [f(x,y)]^2}{MSE} \quad (\text{dB}) \quad (2.12)$$

$$SNR = 10 \log_{10} \frac{\frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N [f(x,y)]^2}{\frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N [f(x,y) - \hat{f}(x,y)]^2} \quad (\text{dB}) \quad (2.13)$$

การวัดที่คล้ายๆ กันคือ ค่าสัดส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด (peak-signal-to-noise ratio, PSNR) โดยสามารถหาได้ดังนี้

$$PSNR = 10 \log_{10} \frac{Peak^2}{MSE} \quad (\text{dB}) \quad (2.14)$$

เมื่อ Peak แทนค่าสูงสุดของขอบเขตของข้อมูลทั้ง SNR และ PSNR เป็นวิธีที่นิยมใช้อธิบายคุณภาพของการบีบอัดข้อมูลภาพ

สำหรับที่มีจุดภาพแทนด้วยข้อมูลขนาด n บิต หรืออาจกล่าวได้ว่ามีจำนวนระดับความเทา (gray scale) เป็น 2^n จะได้

$$PSNR = 10 \log_{10} \frac{(2^n - 1)^2}{MSE} \quad (\text{dB}) \quad (2.15)$$

ดังนั้นภาพที่มีจุดภาพแทนด้วยข้อมูลขนาด 8 บิต หรือมีจำนวนระดับความเทาเป็น 256 จะได้

$$PSNR = 10 \log_{10} \frac{(2^8 - 1)^2}{MSE} \quad (2.16)$$

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad (2.17)$$

$$PSNR = 20 \log_{10} \frac{255}{RMSE} \quad (2.18)$$

2.5.3 การวัดประสิทธิภาพของการบีบอัดข้อมูลภาพด้วยความเข้าใจ (Perceptual Measures)

เป็นการยากที่จะใช้คณิตศาสตร์ในการบอกปริมาณความผิดพลาดตามความเข้าใจของมนุษย์ และการวัดเชิงคณิตศาสตร์ก็ไม่มีความสัมพันธ์โดยตรงกับคุณภาพของภาพ หมายความว่าอัลกอริธึมอาจจะทำให้เกิดความผิดพลาดเชิงคณิตศาสตร์มากกว่า แต่เมื่อสังเกตด้วยตาแล้วอาจจะมีคุณภาพดีกว่า ซึ่งกรณีนี้สามารถเกิดขึ้นได้ถ้าอัลกอริธึมให้เกิดความผิดพลาดในพื้นที่หรือความถี่มากขึ้นซึ่งตาไม่สามารถวัดได้ โดยทั่วไปรหัสที่สร้างขึ้นด้วยการควอนไทซ์แบบหยาบๆ เช่น ผลจากบล็อก (block effect) ผลจากทางเดิน (countour effect) ผลจากวงแหวน (ringing effect) เป็นต้น จะไม่แสดงความเพี้ยนทางคณิตศาสตร์ให้เห็นอย่างชัดเจน แต่สามารถที่จะสังเกตเห็นได้ง่ายด้วยสายตามนุษย์

2.5.3.1 อัตราบิต

อัตราบิต (bit rate) คือค่าเฉลี่ยของจำนวนบิตต่อจุดภาพ (bit per pixel,BPP) ของภาพที่ถูกบีบอัดข้อมูล โดยสามารถคำนวณได้จากอัตราส่วนระหว่างจำนวนบิตข้อมูลทั้งหมดของภาพที่ถูกบีบอัดข้อมูล ต่อจำนวนจุดภาพทั้งหมดของภาพต้นแบบ ดังสมการต่อไปนี้

$$\text{Bit Rate (bits per pixel,BPP)} = \frac{\text{จำนวนบิตทั้งหมดของภาพที่ถูกบีบอัดข้อมูล}}{\text{จำนวนจุดภาพทั้งหมดของภาพต้นแบบ}} \quad (2.19)$$

2.5.3.2 อัตราการบีบอัดข้อมูล

อัตราการบีบอัดข้อมูลคือ อัตราส่วนระหว่างจำนวนบิตข้อมูลที่ใช้แทนภาพต้นแบบ และจำนวนบิตข้อมูลที่ใช้แทนภาพที่ถูกทำการบีบอัดข้อมูล โดยสามารถหาได้ดังสมการต่อไปนี้

$$\text{อัตราการบีบอัดข้อมูล} = \frac{\text{จำนวนบิตข้อมูลของภาพต้นแบบ}}{\text{จำนวนบิตข้อมูลของภาพที่ถูกบีบอัดข้อมูล}} \quad (2.20)$$

บทที่ 3

การแปลงดิสครีตโคไซน์

3.1 คุณสมบัติและนิยามของการแปลงดิสครีตโคไซน์ (Discrete Cosine Transform, DCT)

การลดความซับซ้อนของการคำนวณทางคณิตศาสตร์ทั่วไปจะใช้วิธีการแปลงให้เป็นค่าที่เหมาะสมกับแต่ละการใช้งานเพื่อลดเวลาในการประมวลผล การแปลงฟูรีเยร์ถูกนำมาใช้ในการแยกส่วนประกอบของสัญญาณทางเวลาให้อยู่ในรูปของความถี่ในย่านต่าง ๆ ซึ่งได้มีการนำมาประยุกต์ใช้ในงานทางด้านการประมวลผลภาพในปัจจุบัน ในบทนี้จะกล่าวถึงคุณสมบัติและการพัฒนาสู่ DCT

3.1.1 การแปลงโคไซน์ฟูรีเยร์ (Fourier Cosine Transform, FCT)

จากข้อกำหนดของการแปลงฟูรีเยร์ [4] โดยกำหนดให้ ฟังก์ชัน $x(t)$ สำหรับค่า $-\infty < t < \infty$ จะได้สมการแปลงฟูรีเยร์ของฟังก์ชัน $x(t)$ เป็น

$$X(\omega) \equiv F[x(t)] = \left(\frac{1}{2\pi}\right)^{1/2} \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt \quad (3.1)$$

และการแปลงกลับฟูรีเยร์ของฟังก์ชัน $x(t)$ เป็น

$$x(t) \equiv F^{-1}[X(\omega)] = \left(\frac{1}{2\pi}\right)^{1/2} \int_{-\infty}^{\infty} X(\omega)e^{j\omega t} dt \quad (3.2)$$

ทำการสร้างฟังก์ชัน $y(t)$ ในรูปของฟังก์ชัน $x(t)$ โดยกำหนดค่า $t \geq 0$ ตามนี้

$$y(t) = \begin{cases} x(t) & t \geq 0 \\ x(-t) & t \leq 0 \end{cases}$$

จะได้การแปลงฟูรีเยร์ของฟังก์ชัน $y(t)$ ดังนี้

$$\begin{aligned}
 F[y(t)] &= \left(\frac{1}{2\pi}\right)^{1/2} \left\{ \int_0^{\infty} x(t)e^{-j\omega t} dt + \int_{-\infty}^0 x(-t)e^{-j\omega t} dt \right\} \\
 &= \left(\frac{1}{2\pi}\right)^{1/2} \int_0^{\infty} x(t)[e^{-j\omega t} + e^{j\omega t}] dt \\
 &= \left(\frac{2}{\pi}\right)^{1/2} \int_0^{\infty} x(t)\cos(\omega t) dt
 \end{aligned} \tag{3.3}$$

จากสมการที่ (3.3) ซึ่งเป็นสมการของการแปลงโคซายน์ฟูเรียร์ (FCT) ดังนั้นการแปลงโคซายน์ฟูเรียร์ของสมการ $x(t)$ โดยกำหนดให้

$$X_c(\omega) \equiv F_c[x(t)] = \left(\frac{2}{\pi}\right)^{1/2} \int_0^{\infty} x(t)\cos(\omega t) dt \tag{3.4}$$

และหาฟังก์ชันการแปลงกลับของสมการที่ (3.3) ดังนี้

$$y(t) = x(t) \equiv F_c^{-1}[X_c(\omega)] = \left(\frac{2}{\pi}\right)^{1/2} \int_0^{\infty} X_c(\omega)\cos(\omega t) d\omega \quad (t \geq 0) \tag{3.5}$$

จากสมการที่ (3.4) และ (3.5) เป็นฟังก์ชันการแปลงสัมพันธ์กันและจะได้คุณสมบัติหลายประการตามนี้

1) คุณสมบัติการแปลงกลับ (Inversion)

$$F_c \equiv F_c^{-1} \tag{3.6}$$

ซึ่งแสดงดังสมการที่ (3.4) และ (3.5) ได้ว่า

$$F_c^{-1}\{F_c[x(t)]\} = x(t) \quad t \geq 0$$

2) ความเป็นเชิงเส้น (Linearity)

$$F_c[\alpha x(t) + \beta y(t)] = \alpha X_c(\omega) + \beta Y_c(\omega) \quad (3.7)$$

เมื่อ α และ β เป็นค่าคงที่

3) การขยายในเวลา (Scaling in time)

$$\begin{aligned} F_c[x(at)] &= \left(\frac{2}{\pi}\right)^{1/2} \int_0^{\infty} x(at) \cos(\omega t) dt \\ &= a^{-1} \left(\frac{2}{\pi}\right)^{1/2} \int_0^{\infty} x(\tau) \cos\left(\frac{\omega \tau}{a}\right) d\tau \\ &= \frac{1}{a} X_c\left(\frac{\omega}{a}\right), \quad a > 0 \end{aligned} \quad (3.8)$$

4) เลื่อนในเวลา (Shift in time)

$$\begin{aligned} F_c[x(t-\alpha)] &= \left(\frac{2}{\pi}\right)^{1/2} \int_0^{\infty} x(t-\alpha) \cos(\omega t) dt \\ &= \left(\frac{2}{\pi}\right)^{1/2} \int_0^{\infty} x(\tau) \cos[\omega(\tau+\alpha)] d\tau \\ &= \cos(\omega\alpha) F_c[x(t)] - \sin(\omega\alpha) F_s[x(t)] \end{aligned} \quad (3.9a)$$

เมื่อ F_s เป็นการแปลงซายน์ฟูเรียร์ (FST) มีสมการตามนี้

$$F_s[x(t)] \equiv \left(\frac{2}{\pi}\right)^{1/2} \int_0^{\infty} x(t) \sin(\omega t) dt \quad (3.9b)$$

โดยกำหนดว่า $x(t) = 0$ สำหรับค่า $t < 0$

5) อนุพันธ์ในเวลา (Differentiation in time)

$$\begin{aligned}
 F_c \left[\frac{d}{dt} x(t) \right] &= \left(\frac{2}{\pi} \right)^{1/2} \int_0^{\infty} \left[\frac{d}{dt} x(t) \right] \cos(\omega t) dt \\
 &= \left(\frac{2}{\pi} \right)^{1/2} \left\{ [x(t) \cos(\omega t)]_0^{\infty} + \omega \int_0^{\infty} x(t) \sin(\omega t) dt \right\} \quad (3.10) \\
 &= - \left(\frac{2}{\pi} \right)^{1/2} x(0) + \omega F_s [x(t)]
 \end{aligned}$$

โดยสมมุติว่าค่า $x(t)$ มีค่าเป็นศูนย์เมื่อ t เข้าสู่ค่าอนันต์

6) คุณสมบัติการประสาน (The convolution property)

จากการประสานกันของฟังก์ชันสองฟังก์ชัน $x(t)$ และ $y(t)$ จะได้ว่า

$$x(t) * y(t) \equiv \int_0^{\infty} x(t-u) y(u) du \quad (3.11)$$

การแปลงโคซายน์ฟูเรียร์ของสมการที่ (3.11) คือ

$$\begin{aligned}
 F_c [x(t) * y(t)] &= \left(\frac{2}{\pi} \right)^{1/2} \int_0^{\infty} \int_0^{\infty} x(t-u) y(u) \cos(\omega t) du dt \\
 &= \left(\frac{2}{\pi} \right)^{1/2} [X_c(\omega) Y_c(\omega) - X_s(\omega) Y_s(\omega)] \quad (3.12)
 \end{aligned}$$

3.1.2 นิยาม (Definitions)

จากสมการที่ (3.4) การแปลงโคซายน์ฟูเรียร์มีแกนหลัก (Kernel) โดยกำหนดได้ดังนี้

$$K_c(\omega, t) = \cos(\omega t) \quad (3.13a)$$

กำหนดให้ $\omega_m = 2\pi m \delta f$ และ $t_n = n \delta t$ เป็นการสุ่มเชิงมุมความถี่และเวลา เมื่อ δf และ δt แทนช่วงเวลาของการสุ่มหนึ่งหน่วยความถี่และเวลาตามลำดับ โดยค่า m และ n เป็นค่าจำนวนเต็ม สามารถเขียนสมการที่ (3.13a) ให้อยู่ในรูปใหม่ได้ดังนี้

$$K_c(\omega_m, t_n) = K_c(2\pi m \delta f, n \delta t) = \cos(2\pi m n \delta f \delta t) = K_c(m, n) \quad (3.13b)$$

โดยกำหนดต่อไปว่า $\delta f \delta t = 1/(2N)$ เมื่อ N เป็นจำนวนเต็มจะได้ว่า

$$K_c(m, n) = \cos\left(\frac{\pi mn}{N}\right) \quad (3.14)$$

สิ่งที่ได้สมการที่ (3.14) จะเป็นแกนหลักของโคไซน์ฟูเรียร์ช่วง (Discretized Fourier Cosine) ซึ่งมีคุณสมบัติข้อ 1) ถึงข้อ 6) เช่นเดียวกับการแปลงโคไซน์ฟูเรียร์ แต่จะง่ายกว่าในการคำนวณและการนำไปใช้งาน จากสมการที่ (3.14) สามารถกำหนดให้อยู่ในรูปของ Matrix การแปลงขนาด $N + 1, N + 1$ โดยแสดงสัญลักษณ์เป็น $[M]$ โดยมีค่า m และ n เป็นส่วนประกอบ แต่ละค่าใน Matrix ตามนี้

$$[M]_{mn} = \cos\left(\frac{\pi mn}{N}\right) \quad m, n = 0, 1, \dots, N. \quad (3.15)$$

เมื่อทำการคูณ Matrix $[M]$ เข้ากับ vector $x = [x(0), x(1), x(2), \dots, x(N)]^T$ จะได้ vector $X = [X(0), X(1), \dots, X(N)]^T$

$$X = [M]x \quad (3.16)$$

เมื่อ

$$X(m) = \sum_{n=0}^N \cos\left(\frac{\pi mn}{N}\right) x(n) \quad m = 0, 1, \dots, N \quad (3.17)$$

สังเกตเห็นได้ว่า x คือค่าของการสุ่มสัญญาณเป็นจำนวน N ครั้ง ส่วนสมการที่ (3.17) เป็นสมการของการแปลงดิคริตโคไซน์ (Discrete Cosine Transform) ซึ่งนำไปใช้กันมากในงานประมวลผลสัญญาณดิจิทัล (Digital Signal Processing) โดยได้มีการกำหนดนิยามของ DCT ออกเป็น 4 กลุ่ม [4] ด้วยกันดังนี้

1) DCT-I:

$$[C'_{N+1}]_{mn} = \left(\frac{2}{N}\right)^{1/2} \left[k_m k_n \cos\left(\frac{mn\pi}{N}\right) \right] \quad m, n = 0, 1, \dots, N;$$

2) DCT-II;

$$[C_N^{II}]_{mn} = \left(\frac{2}{N}\right)^{1/2} \left[k_m \cos \left(\frac{\left(n + \frac{1}{2}\right)m\pi}{N} \right) \right] \quad m, n = 0, 1, \dots, N-1;$$

3) DCT-III;

$$[C_N^{III}]_{mn} = \left(\frac{2}{N}\right)^{1/2} \left[k_n \cos \left(\frac{\left(m + \frac{1}{2}\right)n\pi}{N} \right) \right] \quad m, n = 0, 1, \dots, N-1;$$

4) DCT-IV;

$$[C_N^{IV}]_{mn} = \left(\frac{2}{N}\right)^{1/2} \left[\cos \left(\frac{\left(m + \frac{1}{2}\right)\left(n + \frac{1}{2}\right)\pi}{N} \right) \right] \quad m, n = 0, 1, \dots, N-1;$$

เมื่อ

$$k_j = \begin{cases} 1 & \text{เมื่อ } j \neq 0 \text{ หรือ } N \\ \frac{1}{\sqrt{2}} & \text{เมื่อ } j = 0 \text{ หรือ } N \end{cases}$$

3.2 การแปลงดิสครีตโคซายน์ 2 มิติ

การแปลงดิสครีตโคซายน์ 2 มิติ นิยมใช้กันมากในการบีบอัดข้อมูลภาพ เนื่องจากลดขั้นตอนการประมวลผลทางคณิตศาสตร์แล้ว ยังให้ค่า MSE เทียบเคียงได้กับ K-L Transforms โดยสมการการแปลงดิสครีตโคซายน์ 2 มิติ จาก [1] แสดงได้ดังนี้

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right] \quad (3.18)$$

เมื่อ

u, v คือ ตัวแปรของ discrete frequency มีค่าเป็น $0, 1, 2, \dots, N-1$

x, y คือ ตำแหน่งของจุดภาพภายในภาพย่อยขนาด $N \times N$ มีค่าเป็น $0, 1, 2, \dots, N-1$

$C(u, v)$ คือ สัมประสิทธิ์ของการแปลง

$f(x, y)$ คือ ข้อมูลเริ่มต้น และ ภาพผลลัพธ์

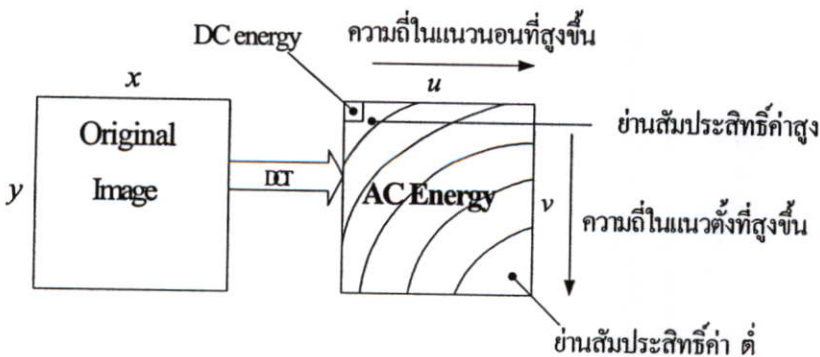
และ α ถูกกำหนดให้เป็น

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{สำหรับ } u=0 \\ \sqrt{\frac{2}{N}} & \text{สำหรับ } u=1, 2, \dots, N-1 \end{cases}$$

และจะได้สมการแปลงกลับดังนี้

$$f(x, y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \alpha(u) \alpha(v) C(u, v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad (3.19)$$

ผลลัพธ์ที่ได้จากการแปลงดิสครีตโคซายน์ 2 มิติ ของข้อมูลภาพ จะประกอบด้วย 2 ส่วน คือ ส่วนพลังงาน DC (DC component) หรือระดับความสว่าง และ ส่วนพลังงาน AC (AC Component) หรือระดับความคมชัด โดยพลังงาน DC จะเป็นค่าเฉลี่ยของจุดภาพทั้งหมดจะปรากฏที่มุมซ้ายบนของภาพผลลัพธ์หรือ ที่จุด $0, 0$ ส่วนพลังงานของ AC จะกระจายแบ่งแยกเป็นชั้น ๆ โดยมีค่าสัมประสิทธิ์จากมากไปหาน้อย ส่วนที่อยู่ใกล้จุดพลังงาน DC จะมีค่าสัมประสิทธิ์สูงสุด ส่วนที่อยู่ไกลจากจุดพลังงาน DC จะมีค่าสัมประสิทธิ์ต่ำสุด ดังแสดงได้ดังภาพที่ 3.1 และ ตารางที่ 3.1 แสดงตัวอย่างผลของการแปลง DCT



ภาพที่ 3.1 แสดงส่วนประกอบของผลลัพธ์ที่ได้จากการแปลงดิสครีตโคซายน์ 2 มิติ

ตารางที่ 3.1 (a) แสดงข้อมูลขนาด 8*8 (b) แสดงผลลัพธ์ที่ได้จากการแปลงดิสคริตโคซายน์

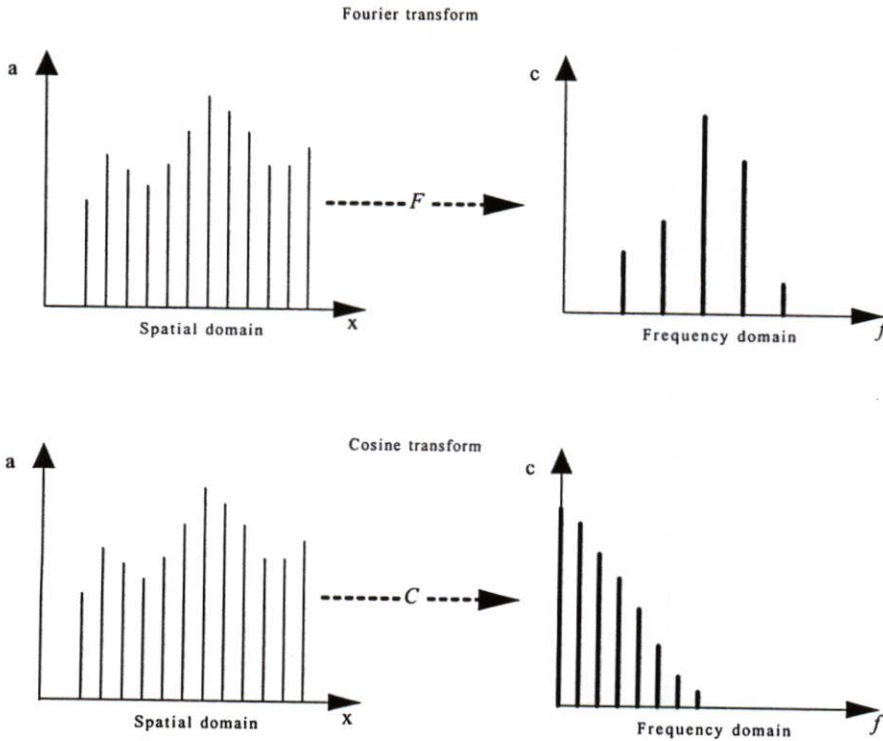
140	144	147	140	140	155	179	175
144	152	140	147	140	148	167	179
125	155	136	167	163	162	152	172
168	145	156	160	152	155	136	160
162	148	156	148	140	136	147	132
147	167	140	155	155	140	136	132
136	156	123	167	162	144	140	147
148	155	136	155	152	147	147	175

(a)

151	-18	15	-9	23	-9	-14	-19
21	-34	26	-9	-11	11	14	7
-10	-24	-2	6	-18	3	-20	-1
-8	-5	14	-15	-8	-3	-3	8
-3	10	8	1	-11	18	18	15
4	-2	-18	8	8	-4	1	-7
9	1	-3	4	-1	-7	-1	-2
0	-8	-2	2	1	4	-6	0

(b)

จากคุณลักษณะที่ได้จากการแปลงดิสคริตโคซายน์จะเห็นได้ว่า เมื่อความถี่ยิ่งสูงขึ้นค่าสัมประสิทธิ์ในส่วนของพลังงาน AC จะมีค่าน้อยลงหรือเข้าใกล้ศูนย์ ดังนั้นเมื่อต้องการบีบอัดข้อมูลภาพก็ทำการส่งเฉพาะส่วนที่มีค่าสัมประสิทธิ์สูงออกไป และตัดส่วนสัมประสิทธิ์ต่ำที่ไม่มีผลกระทบต่อข้อมูลภาพมากนักทิ้งไปก็จะสามารถลดทอนข้อมูลในการส่งภาพได้ นอกจากนี้สามารถพิจารณาลักษณะการกระจายของสัมประสิทธิ์ DCT เทียบกับ สัมประสิทธิ์จากการแปลงแบบฟูเรียร์ ดังแสดงในภาพที่ 3.2 โดย a เป็นขนาดของข้อมูลในสเปเชียลโดเมน และ c เป็นค่าขนาดของกลุ่มพลังงานใน โดเมนความถี่



ภาพที่ 3.2 แสดงการกระจายของสัมประสิทธิ์ของการแปลงแบบฟูเรียร์ และแบบดิสคริต โคซายน์

จากในภาพที่ 3.2 จะเห็นว่าค่าสัมประสิทธิ์การแปลงแบบฟูเรียร์นั้นมีลักษณะการกระจายที่มากกว่าการแปลงแบบดิสคริต โคซายน์ และเมื่อพิจารณาสัมประสิทธิ์ที่ได้ที่ความถี่เดียวกันในแต่ละบล็อกจะมีค่าอยู่ในช่วงที่ใกล้เคียงกัน ในลักษณะเช่นนี้จึงได้นำมาประยุกต์ใช้กับในวิทยานิพนธ์นี้ซึ่งจะกล่าวในหัวข้อถัดไป

3.3 การปรับปรุง และจัดลำดับค่าสัมประสิทธิ์ดิสคริตโคซายน์ในการบีบอัดข้อมูลภาพ

ในระบบ baseline ซึ่งบ่อยครั้งถูกเรียกว่า ระบบ baseline แบบลำดับ ความละเอียดของข้อมูลอินพุท และ เอ้าท์พุทถูกจำกัดด้วยข้อมูล 8 บิต ในขณะที่สัมประสิทธิ์ที่ได้จากการแปลงดิสคริตโคซายน์ อาจมีมากถึง 11 บิต [1]

ขั้นแรกของการแปลงดิสคริต โคซายน์เริ่มจากภาพอินพุทที่เข้ามาจะถูกแบ่งออกเป็นบล็อกย่อยๆ โดยสามารถกำหนดขนาดของบล็อกย่อยได้ ส่วนมากแล้วขนาดของบล็อกย่อยจะมีค่าเป็นเลขยกกำลังสอง เช่น 4×4 , 8×8 , 16×16 ซึ่งแต่ละบล็อกย่อยจะไม่ซ้อนทับกัน ในวิทยานิพนธ์นี้จะแบ่งเป็นบล็อกย่อยขนาด 8×8 จุดภาพ โดยแต่ละบล็อกย่อยจะมี 64 จุดภาพ ซึ่งจะถูกลื่อน (shift) ด้วยการลบออกด้วยจำนวน 2^{n-1} (เพื่อลดค่าที่ใช้คำนวณให้ต่ำลง เป็นผลให้ค่าสัมประสิทธิ์ที่ได้มีค่าไม่สูงมากเกินไป) ขณะที่ 2^n คือ จำนวนระดับสีเทาที่มากที่สุด ดังตัวอย่างเช่น ในวิทยานิพนธ์นี้ได้ใช้ค่าระดับสีเทาที่มากที่สุดเท่ากับ $255 = 2^8$ ถ้าสมมติให้จุดภาพหนึ่ง ($f(1,2)$) มีค่าระดับสีเทาเท่ากับ 0 เมื่อทำการลบด้วย 2^{n-1} จะได้

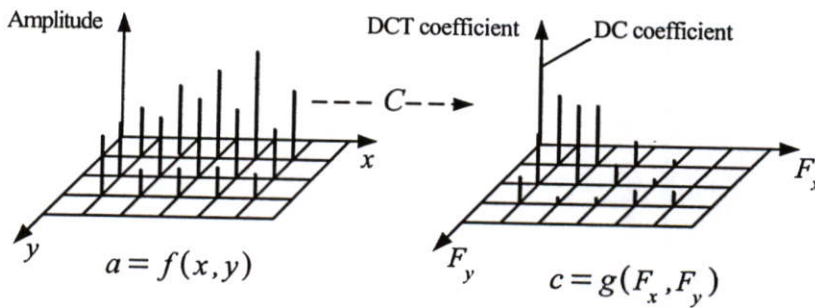
$$\therefore 0 - 2^{8-1} = -128$$

หรือถ้าสมมติให้ $f(3,3) = 255$ จะได้

$$\therefore 255 - 2^{8-1} = 127$$

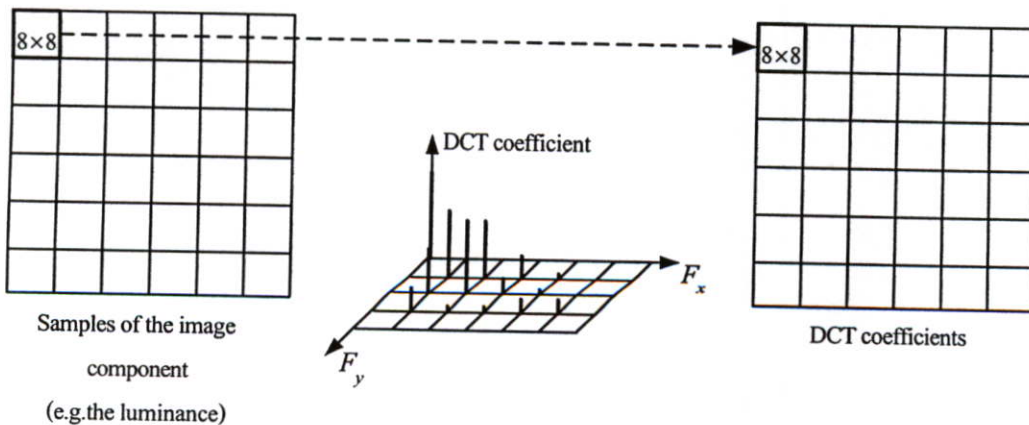
ซึ่งในแต่ละบล็อกย่อยนี้จะใช้การแปลงดิคคริตโคซายน์ แบบ 2 มิติ ดังแสดงในสมการที่ 3.18 ทำนองเดียวกันเมื่อต้องการแปลงกลับ (Inverse DCT) ดังแสดงในสมการที่ 3.19

ในแต่ละบล็อกของ 8×8 นั้นจะมีทั้งหมด 64 ค่า โดยแต่ละค่าจะแสดงขนาดค่าระดับสีเทา หรือ แอมพลิจูด ขนาดของระดับสีเทานี้จะเป็นฟังก์ชันของจุดพิกัด 2 จุด โดยกำหนดให้ $a = f(x,y)$ โดยที่ x,y จะเป็นมิติของสเปาเรียลโดเมน ดังตัวอย่างในภาพลักษณะ 3 มิติในบางส่วนของบล็อก 8×8 ดังแสดงในภาพที่ 3.3



ภาพที่ 3.3 แสดงการแปลงดิคคริตโคซายน์แบบ 2 มิติในลักษณะภาพ 3 มิติ

หลังจากผ่านกระบวนการแปลงด้วยดิคคริตโคไซน์มาจะได้รูปแบบของฟังก์ชันที่อยู่ในโดเมนของความถี่โดยสมมติฟังก์ชัน $C(u,v)$ คือ $C = g(F_x, F_y)$ โดย C จะเป็นค่าของสัมประสิทธิ์ ส่วนค่าของ F_x และ F_y จะแสดงถึงค่าที่เกิดขึ้นในแกนของความถี่ในแต่ละทิศทาง โดยจากภาพที่ 3.4 เป็นการแสดงการแปลงแบบดิคคริตโคไซน์ในแต่ละบล็อก 8×8



ภาพที่ 3.4 แสดงการแปลงแบบดิคคริตโคไซน์ในแต่ละบล็อก 8×8

3.3.1 การปรับปรุงค่าสัมประสิทธิ์

เนื่องจากการบีบอัดข้อมูลที่จะกล่าวถึงในบทถัดไปต้องการอินพุทเป็นจำนวนเต็มขนาด 8 บิต แบบไม่มีเครื่องหมาย แต่ค่าสัมประสิทธิ์ที่ได้จากการแปลงดิสครีตโคซายน์จะมีทั้งค่าลบ และ บวก และยังมีค่าเป็นทศนิยม จึงต้องมีการกำจัดค่าลบและทศนิยม และต้องขยายข้อมูลให้มีขอบเขตอยู่ระหว่าง 0-255 อย่างเหมาะสมโดยมีขั้นตอนดังนี้

1) เพื่อเพิ่มย่านของทศนิยมให้กว้างขึ้น (มีผลในการลดค่าความผิดพลาดในการปิดเศษทศนิยม) จึงต้องทำการหารสัมประสิทธิ์ทั้งหมดด้วย 10 ยกกำลังค่าต่างๆ เช่น $10^1, 10^2, \dots, 10^n$ โดยที่ $n = 1, 2, 3, \dots, n$

2) ทำการปรับค่าสัมประสิทธิ์ที่เป็นลบให้เป็นบวกทั้งหมด ทำโดย จากสัมประสิทธิ์ที่ผ่านขั้นตอนที่ 1 มาแล้ว ให้นำค่าลบที่น้อยที่สุดมาบวกกับสัมประสิทธิ์ทั้งหมด ซึ่งจากจุดนี้ทำให้ค่าน้อยที่สุดปรับจากค่าลบขึ้นมาเป็นศูนย์

3) เป็นการปรับค่าสัมประสิทธิ์ที่ได้จากขั้นที่ 2 ให้มีค่าอยู่ในช่วง 0-255 โดยการทำให้ไดนามิกเรนจ์ [1] จะใช้สูตรการแปลงดัง สมการที่ 3.20

$$s = c \log(1 + |r|) \quad (3.20)$$

เมื่อ

s คือ ค่าสัมประสิทธิ์หลังจากการทำไดนามิกเรนจ์

c คือ เป็นค่าคงที่ที่ใช้ในการสเกลและเป็นฟังก์ชันลอการิทึม หาได้จากสมการที่ 3.21

$$c = \frac{255}{\log(1 + |r_{\max}|)} \quad (3.21)$$

เมื่อ

r คือ ค่าสัมประสิทธิ์ในแต่ละความถี่

r_{\max} คือ ค่าสัมประสิทธิ์ที่มากที่สุดในแต่ละภาพ

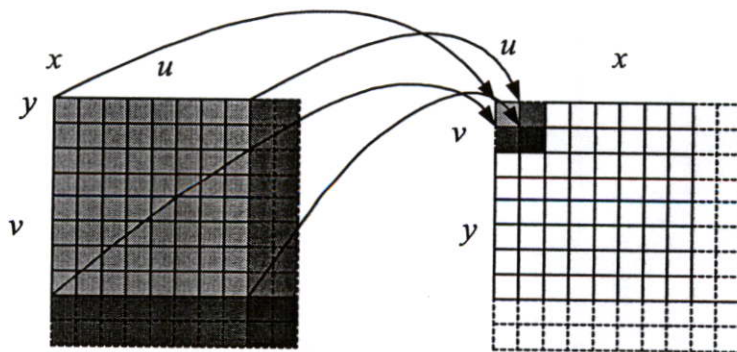
จากขั้นตอนการปรับปรุงค่าสัมประสิทธิ์ทั้ง 3 ขั้นจะพบว่าในขั้นตอนที่ 1 ค่าของ 10^n จะมีค่าไม่จำกัด แต่จากการทดลองเพิ่มค่า n จะทำให้ได้ค่าความผิดพลาด (Mean square error (MSE)) หลังการแปลงกลับดิสครีตโคซายน์ (Inverse DCT) ลดลงเรื่อยๆ จนถึงค่าค่าหนึ่ง ประมาณ $n = 5$ ค่าผิดพลาดที่ได้เกือบจะคงที่ ดังแสดงในตารางที่ 3.2

ตารางที่ 3.2 ตารางเปรียบเทียบค่าผิดพลาด (Mean square error) ของภาพผลลัพธ์ที่ได้จากการปรับ
 ขดแซมปริสทธิคัสคริตโคซานน์โดยใช้ค่า n ต่างกัน

ภาพ ภาพต้นแบบ	ค่า MSE ที่ $DCT/10''$					
	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$
Baboon	30.341	14.634	5.811	2.052	2.050	2.011
Airplane	13.189	7.945	4.953	3.590	3.111	2.070
Lena	60.93	26.95	7.650	3.204	2.867	2.861
Man	43.79	29.327	12.540	5.234	3.032	2.818
Peppers	29.327	25.725	6.960	3.633	3.128	2.72
Camera man	22.11	11.85	6.349	3.529	2.83	2.43
Lansat TM	60.65	30.707	12.21	5.286	3.172	2.65
Jers-1 OPS	30.809	15.32	5.95	2.99	2.911	2.88
ERS-1 SAR	58.344	28.52	12.13	4.69	2.72	2.611
Jers-1 SAR	34.00	16.60	6.45	3.179	2.947	2.951

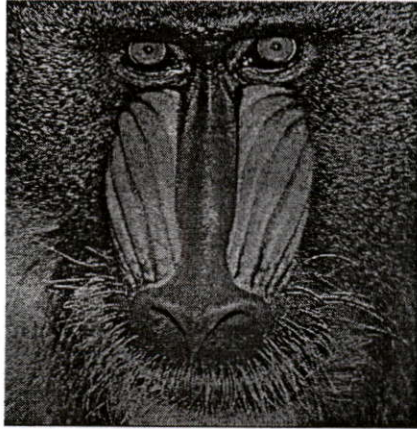
3.3.2 การจัดลำดับค่าแซมปริสทธิ

เนื่องจากประสิทธิภาพของการบีบอัดข้อมูลแบบอาศัยข้อมูลเวกเตอร์ที่จะกล่าวในบทถัดไปจะ
 จำกัดอยู่กับลักษณะข้อมูลภาพอินพุทที่มีความต่อเนื่องมากหรือน้อย ดังนั้นจึงต้องมีการจัดลำดับ
 ค่าแซมปริสทธิใหม่ ให้ข้อมูลค่าแซมปริสทธิที่จะเป็นข้อมูลอินพุทในการบีบอัดข้อมูลมีความต่อ
 เนืองมากขึ้น ทำได้โดยการสลับตำแหน่งจุดภาพที่ (x,y) ของภาพย่อย (u,v) ไปยังภาพย่อยที่
 (x,y) ณ ตำแหน่งจุดภาพ (u,v) ซึ่งสามารถทำได้โดยง่ายโดยการสลับดัชนีที่ใช้จุดภาพ และ
 ภาพย่อยตามลำดับสำหรับภาพขนาด 64×64 จุดภาพ ดูภาพที่ 3.5 ประกอบ

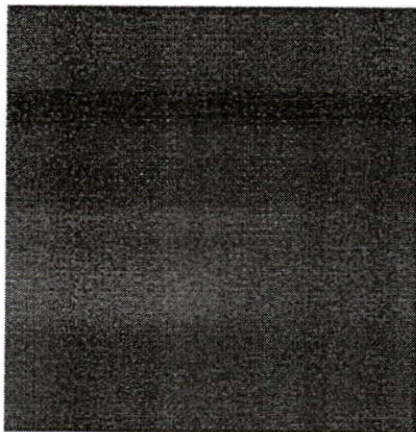


ภาพที่ 3.5 แสดงการจัดลำดับค่าแซมปริสทธิคัสคริตโคซานน์

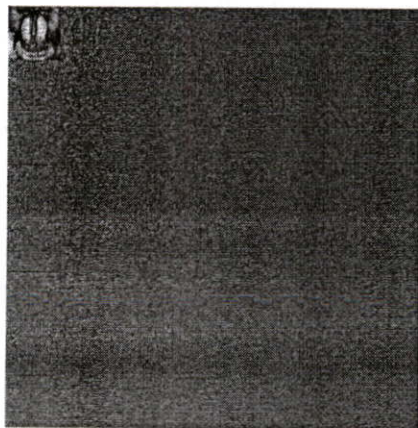
การจัดลำดับค่าสัมประสิทธิ์ดิสกรีตโคซายน์ใหม่นี้ทำให้ค่าสัมประสิทธิ์ที่ความถี่เดียวกันในแต่ละบล็อกของการแปลง (8*8) จะมาเรียงต่อกันในบล็อกใหม่ตามลำดับบล็อก ซึ่งตัวอย่างผลการทดลองการจัดลำดับค่าสัมประสิทธิ์ดิสกรีตโคซายน์ แสดงในภาพที่ 3.6-3.9



(a) ภาพต้นแบบ Baboon ขนาด 512×512×8 บิต



(b) ภาพสัมประสิทธิ์ดิสกรีตโคซายน์ก่อนการจัดลำดับ

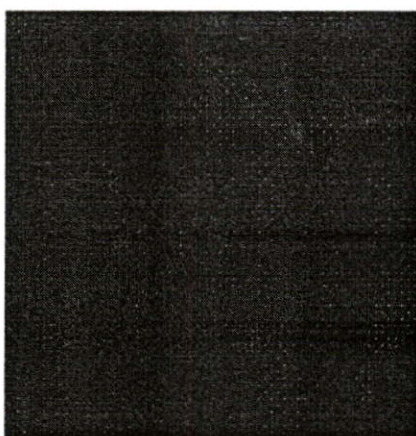


(c) ภาพสัมประสิทธิ์ดิสกรีตโคซายน์หลังการจัดลำดับ

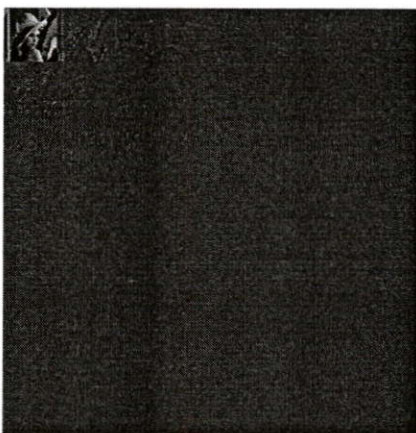
ภาพที่ 3.6 (a)-(c) แสดงผลการทดลองของภาพต้นแบบ Baboon ในการจัดลำดับค่าสัมประสิทธิ์



(a) ภาพต้นแบบ Lena ขนาด 512×512×8 บิต



(b) ภาพสัมพันธ์คิฮิสคริตโคชาชนก่อนการจัดลำดับ

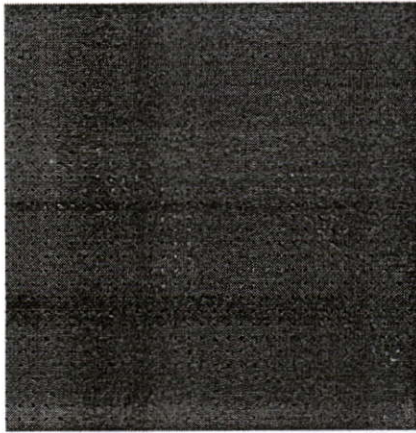


(c) ภาพสัมพันธ์คิฮิสคริตโคชาชนหลังการจัดลำดับ

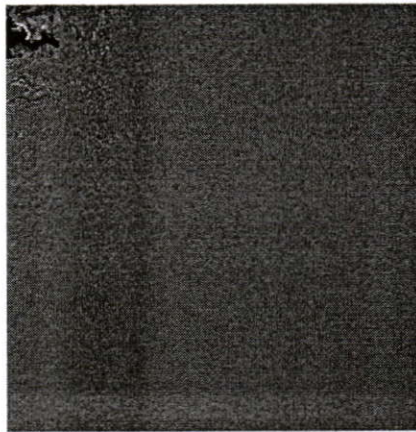
ภาพที่ 3.7 (a)-(c) แสดงภาพผลการทดลองของภาพต้นแบบ Lena ในการจัดลำดับค่าสัมพันธ์



(a) ภาพต้นแบบภาพถ่ายจากดาวเทียม Landsat TM ขนาด 256×256×8 บิต



(b) ภาพสัมพันธ์เชิงสีครีต โคชาชนก่อนการจัดลำดับ

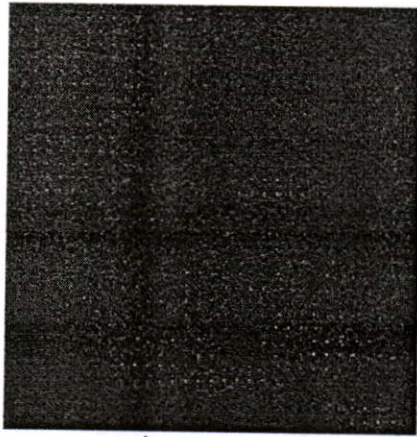


(c) ภาพสัมพันธ์เชิงสีครีต โคชาชนหลังการจัดลำดับ

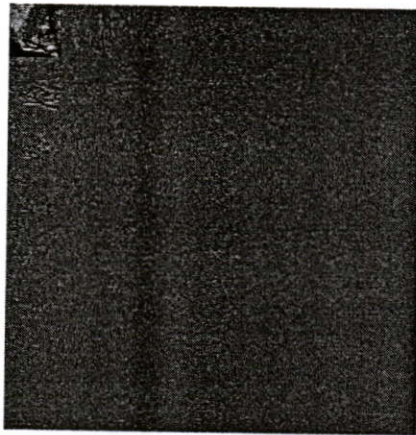
ภาพที่ 3.8 (a)-(c) แสดงภาพผลการทดลองของภาพต้นแบบภาพถ่ายจากดาวเทียม Landsat TM ใน การจัดลำดับค่าสัมพันธ์



(a) ภาพต้นแบบภาพถ่ายจากดาวเทียม Jers-1 SAR ขนาด $256 \times 256 \times 8$ บิต



(b) ภาพสัมพันธ์คิสุคริตโคชายนก่อนการจัดลำดับ



(c) ภาพสัมพันธ์คิสุคริตโคชายนหลังการจัดลำดับ

ภาพที่ 3.9 (a)-(c) แสดงภาพผลการทดลองของภาพต้นแบบภาพถ่ายจากดาวเทียม Jers-1 SAR ในการจัดลำดับค่าสัมพันธ์

หมายเหตุ ในภาพ (b) และ (c) เป็นภาพที่ผ่านการปรับค่าสัมพันธ์แล้ว โดยเลือกใช้ $n = 5$ ซึ่งทำให้ภาพผลลัพธ์หลังการแปลงกลับคิสุคริตโคชายน มีค่าผิดพลาด (MSE) น้อยที่สุด

บทที่ 4

การบีบอัดข้อมูลแบบอาศัยข้อมูลแวดล้อม

4.1 บทนำ

การบีบอัดข้อมูลแบบอาศัยข้อมูลแวดล้อมที่นำเสนอในบทนี้ได้ประยุกต์ร่วมกับการใช้การเข้ารหัสแบบโกลอมบ์-ไรซ์ (Golomb-Rice) วิธีนี้จัดได้ว่าเป็นการบีบอัดข้อมูลแบบไม่มีการสูญเสียและเป็นวิธีการที่ซับซ้อนต่ำ ซึ่งจากหลักการบีบอัดข้อมูลแบบไม่เกิดการผิดพลาดของข้อมูลข่าวสาร (error-free compression) ประสิทธิภาพการบีบอัดข้อมูลจะอยู่ที่ 2-10 เท่า ซึ่งสามารถนำไปใช้ได้ทั้งกับข้อมูลภาพระดับสีเทา และ แบบไบนารี เทคนิคการบีบอัดข้อมูลแบบไม่เกิดการผิดพลาดของข้อมูลข่าวสารนี้ โดยทั่วไปประกอบด้วย 2 ขั้นตอนดังนี้ [1]

1) การสร้างแบบจำลอง (modeling) คือ การกำหนดวิธีการแสดงภาพโดยลดส่วนที่ซ้ำซ้อนระหว่างจุดภาพ

2) การเข้ารหัสแหล่งข้อมูลซึ่งได้จากแบบจำลอง โดยกำจัดความซ้ำซ้อนในการเข้ารหัส

วิธีที่ง่ายที่สุดสำหรับการบีบอัดข้อมูลแบบไม่เกิดการผิดพลาดของข้อมูลข่าวสาร คือ ลดเฉพาะส่วนความซ้ำซ้อนในการเข้ารหัส ซึ่งความซ้ำซ้อนในการเข้ารหัสปรกติจะแสดงให้เห็นในการเข้ารหัสเลขฐาน 2 ของระดับสีเทาในภาพ และสามารถกำจัดออกได้โดยการเข้ารหัสระดับสีเทา ซึ่งต้องอาศัยโครงสร้างของรหัสแบบปรับความยาว (variable-length code) โดยกำหนดค่าความยาวรหัสที่สั้นที่สุด หรือใช้จำนวนบิตที่น้อย กับข้อมูลระดับสีเทาที่มีระดับความน่าจะเป็นสูงสุด และกำหนดความยาวรหัสที่มากขึ้นกับข้อมูลที่มีความน่าจะเป็นต่ำลงมา ซึ่งความยาวรหัสโดยเฉลี่ยต่อหนึ่งจุดภาพหาได้โดย

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k) \quad (4.1)$$

เมื่อ L_{avg} คือ ค่าความยาวเฉลี่ยของรหัสต่อหนึ่งจุดภาพ (bpp)

L คือ จำนวนของระดับสีเทา

r_k คือ ระดับสีเทาที่ k

$l(r_k)$ คือ จำนวนบิตที่ใช้ในการแทนระดับสีเทา (r_k) แต่ละระดับ

$p_r(r_k)$ คือ ความน่าจะเป็นในการเกิดระดับสีเทา

ส่วนการออกแบบแบบจำลองสามารถพิจารณาแบบเป็นระบบ โดยเมื่อได้ค่าจากการแสดงข้อมูลที่ผ่านมา $x^i = x_1 x_2 x_3 \dots x_i$ สิ่งหนึ่งซึ่งต้องการเพื่อทำการประมาณค่าจุดภาพถัดไป x_{i+1} คือการกำหนดเงื่อนไขการกระจายความน่าจะเป็น $P(\cdot|x^i)$ ให้กับมัน ในทางอุดมคติความยาวรหัส (code

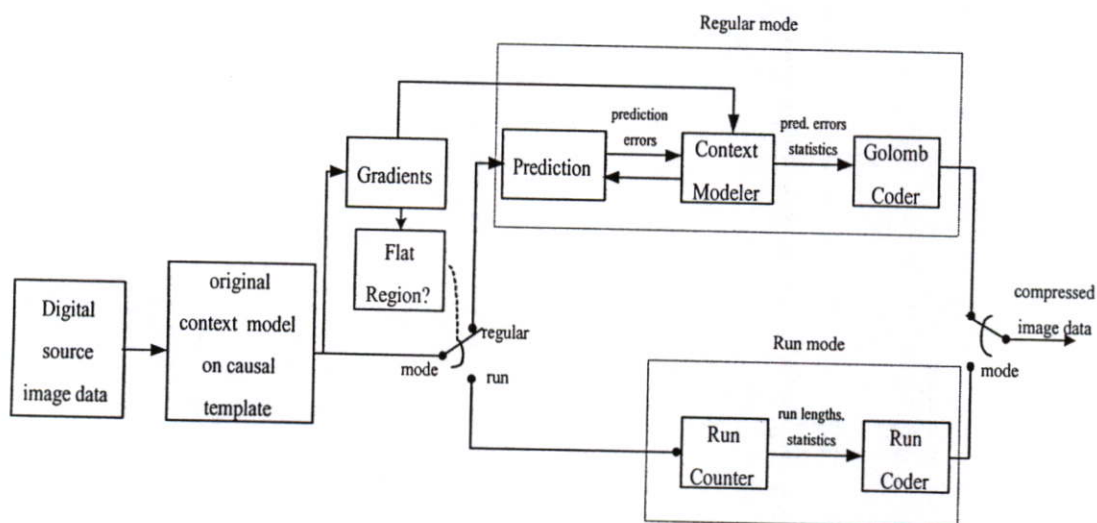
length) ถูกกำหนดโดย x_{i+1} มีค่าความยาวจำนวนบิตเท่ากับ $-\log P(x_{i+1}|x^i)$ ซึ่งเมื่อทำการเฉลี่ยแล้ว จะมีค่าเท่ากับ เอนโทรปี (entropy) ของแบบจำลองความน่าจะเป็นตามทฤษฎีข้อมูลข่าวสาร ทั้งนี้ เพื่อให้จุดภาพถัดไปมีความน่าจะเป็นสูงที่จะมีเอนโทรปีต่ำ ซึ่งสามารถทำได้โดยอาศัยข้อมูลเวดล้อมจำนวนมากขึ้น โดยทั่วไปสามารถแบ่งขั้นตอนต่างๆ ได้ดังนี้

1) ขั้นตอนการทำนายค่า (prediction) โดยทำการหาค่า \hat{x}_{i+1} สำหรับจุดภาพถัดไป x_{i+1} ซึ่งได้จากการคาดเดาจากลำดับที่ผ่านมา (x^i) โดยอาศัย คอซอลเทมเพลท (causal template)

2) การหาเงื่อนไขไปเวดล้อมในการเกิด x_{i+1}

3) การออกแบบแบบจำลองความน่าจะเป็น สำหรับเศษที่เหลือจากการทำนาย หรือ ค่าผิดพลาดคิดได้จาก $\mathcal{E}_{i+1} = x_{i+1} - \hat{x}_{i+1}$ โดยมีเงื่อนไขอยู่บนพื้นฐานของข้อมูลเวดล้อมของ x_{i+1}

ส่วนในวิธีนี้ [5] ได้รวมเอาความไม่ซับซ้อนของการเข้ารหัสแบบปรับตัวได้ของ โกลอมบ์-ไรซ์ และ แทรกทรหัสเพิ่มขยายสำหรับการเข้ารหัสบริเวณข้อมูลภาพที่มีเอนโทรปีต่ำ เพื่อที่จะเพิ่มศักยภาพในการบีบอัดข้อมูลของแบบจำลองข้อมูลเวดล้อม วิธีการคือใช้ตัวทำนายแบบไม่เป็นเชิงเส้นบนพื้นฐานของการตรวจจับขอบภาพ และโดยอาศัยความเรียบง่ายของแบบจำลองข้อมูลเวดล้อมซึ่งหาได้โดยอัตราความลาดเอียง หรือ แกรเดียนท์ (gradient) ที่ถูกควอนไทซ์ ซึ่งเหมาะสมอย่างยิ่งกับการประยุกต์ใช้งานที่ต้องการความซับซ้อนต่ำ อัตราการบีบอัดข้อมูลที่ได้จะมีค่าเท่ากับ หรือ ดีกว่าเมื่อเทียบกับระเบียบวิธีเดียวกันกับการเข้ารหัสแบบเชิงเลขคณิต (arithmetic coding) วิธีดังกล่าวนี้แสดงในภาพที่ 4.1



ภาพที่ 4.1 แสดงแผนผังการบีบอัดข้อมูลแบบอาศัยข้อมูลเวดล้อมร่วมกับการเข้ารหัสแบบ Golomb

-Rice

4.2 ทฤษฎีการบีบอัดข้อมูลแบบอาศัยข้อมูลแวดล้อม (Context based compression)

4.2.1 แบบจำลองข้อมูลแวดล้อมบนคอซอลเทมเพลต (context model on causal template)

ในส่วนของการทำนายค่า และการออกแบบแบบจำลอง [5] กระทำอยู่บนพื้นฐานของคอซอลเทมเพลต ดังแสดงในภาพที่ 4.2 ขณะที่ x แทนจุดภาพปัจจุบัน และ a, b, c และ d เป็นจุดภาพข้างเคียง

index	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$	$x_{1,4}$
2	0	$x_{2,1}$			$x_{2,4}$	$x_{2,4}$
3	$x_{2,1}$	$x_{3,1}$	c	b	d	
4	$x_{3,1}$		a	x		

ภาพที่ 4.2 จุดภาพเสมือนที่ถูกกำหนดก่อนการบีบอัด และcausal templateที่ใช้ในแบบจำลองแวดล้อม และการทำนายค่า

จากภาพที่ 4.2 จุดภาพที่แสดงอยู่ส่วนใน (สีขาว) แสดงถึงภาพที่มีอยู่จริง ขณะที่พื้นที่แรเงาด้านนอกแสดงถึงค่า R_b, R_c และ R_d เมื่อ I_x อยู่ในแถวแรก และจะมีค่า R_d และ R_c เมื่อ I_x อยู่ในคอลัมน์แรก และจะมีค่า R_d เมื่อ I_x อยู่ในคอลัมน์สุดท้าย ส่วนในพื้นที่ที่แรเงาเข้ม คือ คอซอลเทมเพลต จากค่าที่ถูกสร้างขึ้นใหม่ที่จุดภาพ a, b, c, d (R_a, R_b, R_c, R_d ตามลำดับ) จะเป็นข้อมูลแวดล้อมแรกที่ใช้ในการตัดสินใจว่าข้อมูลในจุดภาพ x ควรจะถูกเข้ารหัสในโหมดปกติ หรือ โหมดรัน

4.2.2 การเลือกโหมดการทำงาน

ในการเลือกโหมดการทำงานสามารถทำได้ดังนี้

- 1) คำนวณหาโลคอลเกรเดียนท์ (local gradient) (D_1, D_2 และ D_3) จากสมการที่ 4.2

$$\begin{aligned} D_1 &= R_d - R_b \\ D_2 &= R_b - R_c \\ D_3 &= R_c - R_a \end{aligned} \quad (4.2)$$

- 2) เลือกโหมดการทำงานโดยพิจารณาค่าโลคอลเกรเดียนท์ ซึ่งถ้า $D_1, D_2, D_3 = 0$ จะเลือกเข้าสู่โหมดรัน แต่ถ้า $D_1, D_2, D_3 \neq 0$ จะเลือกเข้าสู่โหมดปกติ และสามารถอธิบายการทำงานของทั้งสองโหมดอย่างคร่าวๆ ดังนี้

(1) โหมดรัน (run mode) จะถูกเลือกเมื่อการประมาณข้อมูลเวกเตอร์ที่ต่อเนื่องกันของจุดภาพที่พิจารณามีความเหมือนกัน หรือใกล้เคียงกัน โดยกระบวนการของแบบจำลองจะเลือกเข้าสู่การทำงานแบบโหมดรัน ซึ่งกระบวนการเข้ารหัสจะข้ามส่วนที่ทำหน้าที่ทำนายค่า และ กระบวนการเข้ารหัสค่าผิดพลาด(error-encoding) ซึ่งในโหมดรันนี้ตัวเข้ารหัสจะทำการมองโดยเริ่มจากจุดภาพ x และจุดภาพที่ตามต่อเนื่องกันมาว่ามีค่าเหมือนกันกับค่าที่ถูกสร้างขึ้นใหม่ ณ จุดภาพที่ตำแหน่ง a หรือไม่ โดยโหมดรันนี้จะสิ้นสุดเมื่อจุดภาพที่กำลังพิจารณามีค่าต่างจากจุดภาพก่อนหน้า หรือ กรณีที่สิ้นสุดของแถวที่พิจารณา ขึ้นกับว่าสิ่งใดเกิดขึ้นก่อน ความยาวของข้อมูลจุดภาพที่ต่อเนื่องกันนี้จะถูกจำแนกการสิ้นสุดโหมดรันว่ามาจากกรณีใด และจะถูกเข้ารหัสแบบรันเรจันบนพื้นฐานของรหัสโคโลมบ์ต่อไป

(2) โหมดปกติ (regular mode) ถูกเลือกเมื่อการประมาณข้อมูลเวกเตอร์มีค่าแตกต่างกัน โดยตัวทำนายค่าจะใช้ค่าของข้อมูลข้างเคียงที่ถูกสร้างขึ้นใหม่ ณ ตำแหน่งจุดภาพ a, b และ c เพื่อการทำนายค่าจุดภาพที่ตำแหน่ง x ซึ่งวิธีนี้ใช้การทำนายแบบตรวจจับขอบภาพอธิบายในหัวข้อถัดไป ค่าผิดพลาดจากการทำนายจะถูกคำนวณได้จากผลต่างระหว่างค่าจริง และค่าทำนาย ณ จุดภาพที่กำลังพิจารณา ซึ่งค่าผิดพลาดนี้จะถูกปรับค่าให้ถูกต้องด้วยค่าข้อมูลเวกเตอร์เพื่อชดเชยความสมมาตรในการทำนาย ค่าผิดพลาดที่ถูกแก้ไขแล้วจะถูกเข้ารหัสโคโลมบ์ต่อไป

4.2.3 การทำนายค่า (Prediction)

ในโหมดการทำงานปกติ จะใช้ตัวทำนายแบบตรวจจับขอบภาพใน [5] โดยค่าทำนาย (P_x) ณ จุดภาพตำแหน่ง x หาได้จาก R_a, R_b และ R_c ณ ตำแหน่งจุดภาพ a, b และ c ตามลำดับ ดังสมการที่ 4.3

$$P_x = \begin{cases} \min(a, b) & \text{ถ้า } c \geq \max(a, b) \\ \max(a, b) & \text{ถ้า } c \leq \min(a, b) \\ a + b + c & \text{otherwise} \end{cases} \quad (4.3)$$

แบบจำลองตัวทำนายนี้สามารถมองได้ว่า ค่าทำนาย (P_x) จะมีค่าเท่ากับ b เมื่อขอบภาพตามแนวตั้งอยู่ทางด้านซ้ายของจุดที่กำลังพิจารณา ในขณะที่ (P_x) จะมีค่าเท่ากับ a เมื่อขอบภาพตามแนวนอนอยู่เหนือจุดที่พิจารณา แล้วจะมีค่าเท่ากับ $a+b-c$ ในกรณีที่ ไม่ปรากฏขอบภาพใดๆ ในตำแหน่งที่กำลังพิจารณา จะสังเกตเห็นว่าแบบจำลองตัวทำนายนี้จะไม่ใช่ค่า d ซึ่งจะถูกใช้ในการออกแบบแบบจำลองข้อมูลเวกเตอร์

การทำนายนี้ถือได้ว่าเป็นค่ากลาง (median) ของทั้งสามค่าของ a, b และ $a+b+c$ ดังนั้นทำนายจึงได้ชื่อใหม่ว่า “median edge detector” (MED) และจากค่าทำนายที่ได้จะมีการปรับขนาดค่าตัวทำนายโดยอาศัยการประมาณการกระจายแบบเรขาคณิต (Geometric distribution)

4.2.4 การออกแบบแบบจำลองข้อมูลแวดล้อม (context modeling)

การลดตัวเลขของพารามิเตอร์เป็นกุญแจสำคัญในการออกแบบแบบจำลอง ตัวเลขทั้งหมดของพารามิเตอร์ในแบบจำลองหาได้โดยอาศัยพารามิเตอร์อิสระที่กำหนดจากการกระจายของรหัสข้อมูลในแต่ละ context และโดยตัวเลขของ context

4.2.4.1 การหาค่าพารามิเตอร์ (parameterization)

การหาค่าพารามิเตอร์ของแบบจำลองสามารถแบ่งเป็นหัวข้อได้ดังนี้

1) แบบจำลองการกระจาย TSGD (two-sided geometric distribution) [6]

แบบจำลองนี้เป็นที่ยอมรับกันอย่างกว้างขวางว่า ในทางสถิติของเศษเหลือจากการทำนายแบบคงที่ในภาพที่มีความต่อเนื่อง (continuous-tone image) เป็นแบบจำลองที่ดีโดยมีศูนย์กลางอยู่ที่ศูนย์กลางตามการกระจายนี้ ความน่าจะเป็นของค่าเศษเหลือจากการทำนาย (\mathcal{E}) เป็นสัดส่วนกับ $\theta^{|\mathcal{E}|}$ โดยที่ $\theta \in (0,1)$ สัดส่วนนี้ควบคุมอัตราการลดลงของกราฟเอ็กซ์โพเนนเชียลทั้งสองข้าง ศูนย์กลางของการกระจาย TSGD อาจมีการเลื่อนไปข้าง แต่จะอยู่ระหว่างค่าจำนวนเต็ม 2 ค่า ดังนั้นแบบจำลองจึงเพิ่มส่วนการชดเชยค่าด้วยพารามิเตอร์ชดเชย (offset parameter (μ)) ที่เหมาะสม โดยให้ μ เป็นค่าที่ไม่เป็นจำนวนเต็ม ซึ่ง $\mu = R - s$ โดยที่ R คือส่วนที่เป็นจำนวนเต็มหรือส่วนไบอัส ส่วน s คือส่วนที่ถูกเลื่อนไป เมื่อ $0 \leq s < 1$ ดังนั้นพารามิเตอร์ของ TSGD เป็นดังสมการที่ 4.4

$$P_{(\theta, \mu)}(\mathcal{E}) = C(\theta, s)\theta^{|\mathcal{E} - R + s|}, \quad \mathcal{E} = 0, \pm 1, \pm 2, \dots \quad (4.4)$$

เมื่อ

$$C(\theta, s) = (1 - \theta) / (\theta^{1-s} + \theta^s) \quad \text{คือ นอมอลไรซ์ แฟกเตอร์}$$

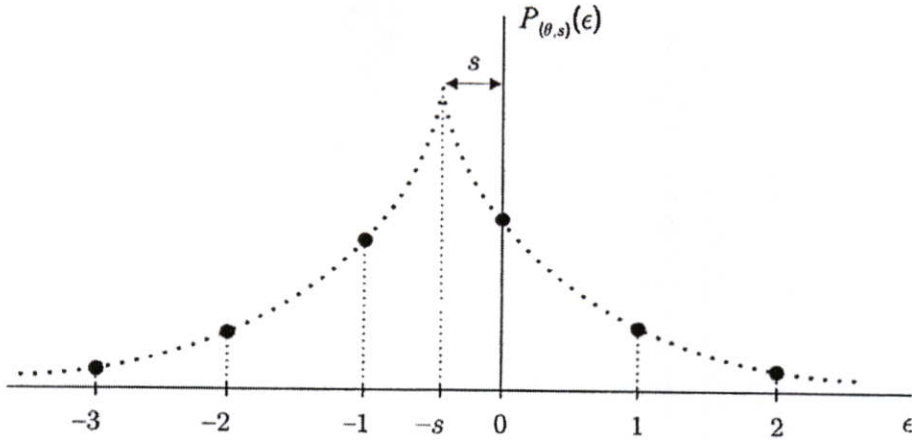
และถ้า ศูนย์กลางอยู่ระหว่าง ค่า 0 กับ 1 แล้วทำให้ $R=0$ จะทำให้สมการของพารามิเตอร์ TSGD มีค่าดังสมการที่ 4.5

$$P_{(\theta, s)}(\mathcal{E}) = C(\theta, s)\theta^{|\mathcal{E} + s|}, \quad \mathcal{E} = 0, \pm 1, \pm 2, \dots \quad (4.5)$$

เมื่อ $0 < \theta < 1$ และ $0 \leq s < 1$

การลดบริวณขอบเขตลงนี้ สำหรับชดเชยให้เข้ากับที่จะนำเสนอในหัวข้อการเข้ารหัส แบบจำลองการกระจายแบบ TSGD ในสมการที่ 4.5 ดูได้จากภาพที่ 4.3

ถ้าศูนย์กลางของ TSGD อยู่ที่ ศูนย์ จะตรงกับ $s = 0$ แต่ถ้า $s = 1/2$ $P_{(\theta, s)}$ จะเป็นการกระจายแบบ bi-modal และจะมีจุดสูงสุดที่เท่ากันอยู่ที่ -1 กับ 0



ภาพที่ 4.3 แสดงการกระจายแบบ TSGD

2) ค่าผิดพลาดที่เหลือจากการทำนาย

ค่าผิดพลาดจากการทำนายที่กล่าวถึงนี้เป็นค่าผิดพลาดเบื้องต้นก่อนการปรับค่าทำนาย หาได้จาก $\mathcal{E} = I_x - P_x$ ซึ่งค่า \mathcal{E} ในสมการที่ 4.5 เป็นค่าอนันต์ แต่ในทางปฏิบัติต้องจำกัดค่า \mathcal{E} ให้อยู่ในช่วง α โดยค่า \mathcal{E} จะถูกนำไปไว้บนช่วง $-P_x \leq \mathcal{E} < \alpha - P_x$ และเมื่อผ่านขบวนการกระจายแบบ TSGD ในสมการที่ 4.5 แล้วสามารถลดค่า \mathcal{E} ให้จำกัดขอบเขตอยู่ระหว่าง $-(\alpha/2) \leq \mathcal{E} < (\alpha/2) - 1$ เมื่อ $\alpha = 2^\beta$ และ β คือ จำนวนบิตข้อมูลที่ใช้แทนภาพ

4.2.4.2 การหาข้อมูลแวดล้อม (context determination)

ข้อมูลแวดล้อมนั้นเป็นเงื่อนไขของการเข้ารหัสเศษเหลือจากการทำนาย มีเป้าหมายเพื่อชดเชยค่าผิดพลาดจากการทำนาย ข้อมูลแวดล้อมนี้สร้างโดยอาศัยโลกอลแกรเดียนท์ที่ได้กล่าวมาแล้ว ซึ่งยี่ระดับการเคลื่อนไหวรอบๆ จุดที่พิจารณาพัฒนาเป็นสถิติของความผิดพลาดจากการทำนาย โดยทฤษฎีความสมมาตร [7] ค่าโลกอลแกรเดียนท์ D_1, D_2 และ D_3 มีอิทธิพลต่อแบบจำลองในทิศทางเดียวกัน ซึ่งแต่ละค่าของโลกอลแกรเดียนท์ จะถูกควอนไตซ์ ให้มีค่าน้อยลง

มีหลักการ คือ ค่าของขอบเขตภายในแต่ละ context จะถูกควอนไตซ์ด้วยค่าที่ต่างกันซึ่งควรจะสามารถปรับเปลี่ยนให้เหมาะสมได้ อย่างไรก็ตามภายใต้เงื่อนไขความซับซ้อนต่ำจึงต้องมีการกำหนดขอบเขตให้ชัดเจน ดังนั้นจากกฎความสมมาตร สามารถกำหนดดัชนีได้เป็น $-T, \dots, -1, 0, 1, \dots, T$ เมื่อ T คือดัชนีที่ใช้แทนขอบเขตการควอนไตซ์ เพราะฉะนั้นจะมีจำนวน context เท่ากับ $(2T+1)^3$ การลดจำนวนของ context ทำโดยการเฝ้าสังเกตความเป็นสมมาตรโดยสมมติว่า

$$\text{Prob}\{\mathcal{E}_{t+1} = \Delta/Q_t = [q_1, q_2, q_3]\} = \text{Prob}\{\mathcal{E}_{t+1} = -\Delta/Q_t = [-q_1, -q_2, -q_3]\}$$

เมื่อ Q_i แทน ค่าทั้งสามของ q

q_i แทน ค่า D ที่ถูกควอนไตซ์ ; $i = 1, 2, 3$

ณ จุดนี้จึงกำหนดตัวเลข และเครื่องหมาย (sign) ของ context เป็นดังนี้

$$\text{Sign} = \begin{cases} -1 & \text{ถ้า } (q_1 < 0) \text{ หรือ } (q_1 = 0 \& q_2 < 0) \text{ หรือ } (q_1 = 0 \& q_2 = 0 \& q_3 < 0) \\ 1 & \text{ถ้า เป็นกรณีอื่น} \end{cases}$$

และกำหนดให้

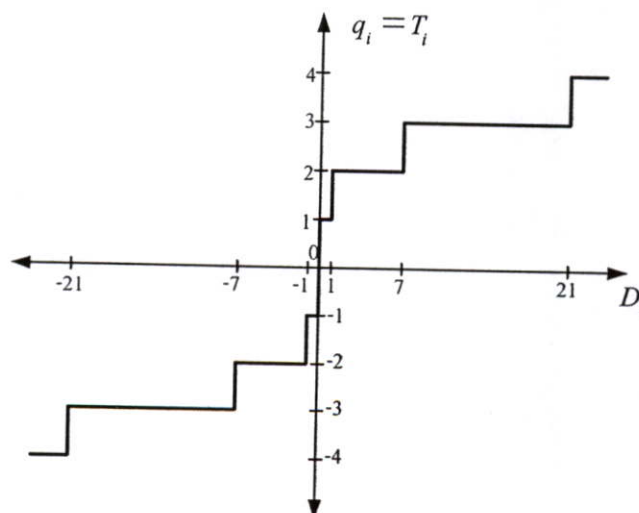
$$q_1 \cdot q_2 \cdot q_3 = -q_1, -q_2, -q_3 \quad \text{ถ้า } (q_1 < 0) \text{ หรือ } (q_1 = 0 \& q_2 < 0) \\ \text{หรือ } (q_1 = 0 \& q_2 = 0 \& q_3 < 0)$$

เครื่องหมายของ context นี้มีความสำคัญในการถอดรหัส ซึ่งเครื่องหมายจำเป็นในการหาค่า \mathcal{E} เริ่มต้น เช่นเดียวกันเงื่อนไขนี้ทำให้จำนวน context ลดลงเหลือ $((2T + 1)^3 + 1)/2$

สำหรับขอบเขตการควอนไตซ์ที่ค่าระดับสี่เทา เท่ากับ 8 บิตต่อพิกเซล[7] จะมีขอบเขตอยู่ที่ $\{0\}, \pm\{1, 2\}, \pm\{3, 4, 5, 6\}, \pm\{7, 8, \dots, 20\}, \pm\{e | e \geq 21\}$ อย่างไรก็ตามค่านี้สามารถปรับเปลี่ยนได้ยกเว้น ค่าที่ 0 ต้องคงที่ไว้ ในบทนี้ได้กำหนดค่า $T = 4$ จะได้จำนวน context เท่ากับ

$$((2 \times 4)^3 + 1)/2 = 365 \text{ context}$$

เงื่อนไขของการควอนไตซ์แสดงในภาพที่ 4.4



ภาพที่ 4.4 แสดงตัวอย่างการควอนไตซ์

ค่าเวกเตอร์ที่ได้จากการควอนไทซ์ $(\pm q_1, \pm q_2, \pm q_3)$ จะถูกรวมเป็นค่าค่าเดียวคือ Q โดยการเรียงลำดับตาม q_1, q_2 และ q_3 เพื่อง่ายต่อการชี้ตำแหน่ง context ซึ่งสมการที่ใช้ในการรวมค่าแสดงในสมการที่ 4.6

$$Q = (81 \times q_1) + (9 \times q_2) + (q_3) + 1 \quad (4.6)$$

เช่น ถ้าสมมติให้ $q_1 = 0, q_2 = 0, q_3 = 0$

$$\begin{aligned} \text{จะได้ } Q &= (81 \times 0) + (9 \times 0) + (0) + 1 \\ &= 1 \quad ; \text{ตำแหน่ง context ที่ } Q \text{ คือ } 1 \end{aligned}$$

อีกกรณีที่ $q_1 = 4, q_2 = 4, q_3 = 4$

$$\begin{aligned} \text{จะได้ } Q &= (81 \times 4) + (9 \times 4) + (4) + 1 \\ &= 365 \quad ; \text{ตำแหน่ง context ที่ } Q \text{ คือ } 365 \end{aligned}$$

4.2.4.3 การแก้ไขค่าผิดพลาดจากการทำนายแบบปรับตัวได้ (adaptive correction)

ในหัวข้อนี้จะกล่าวถึงการแก้ไขค่าผิดพลาดแบบปรับตัวได้ หรือ อาจเรียกว่า bias cancellation ที่กระทำที่ความซับซ้อนต่ำ ได้ใช้หลักการประมาณแบบ maximum-likelihood (ML) การประมาณของ R ในสมการที่ 4.4 จะเป็นตัวสั่งให้ขบวนการ bias cancellation ให้อยู่บนพื้นฐานบนค่ากลางของความผิดพลาดของตัวทำนายที่เกิดขึ้นใน context โดยอาศัยตัวทำนายในสมการ 4.3 พื้นฐานการประมาณบนค่าเฉลี่ยสามารถทำได้โดย เก็บจำนวนนับครั้ง (N)ของแต่ละ context ที่เกิดการผิดพลาด และบวกสะสมความผิดพลาด (A) ของตัวทำนายใน context นั้นๆ ที่เกิดการผิดพลาดขึ้นแล้วมาทำการหาค่าเฉลี่ยดังสมการที่ 4.7

$$C' = \lceil A/N \rceil \quad (4.7)$$

เมื่อ

C' คือ ค่าเฉลี่ยความผิดพลาดที่ใช้แก้ไขค่าผิดพลาดจากการทำนาย จากนั้นนำค่า C' มาบวกกับค่าทำนาย P_x เพื่อชดเชยความผิดพลาดจากการทำนายดังในสมการที่ 4.8

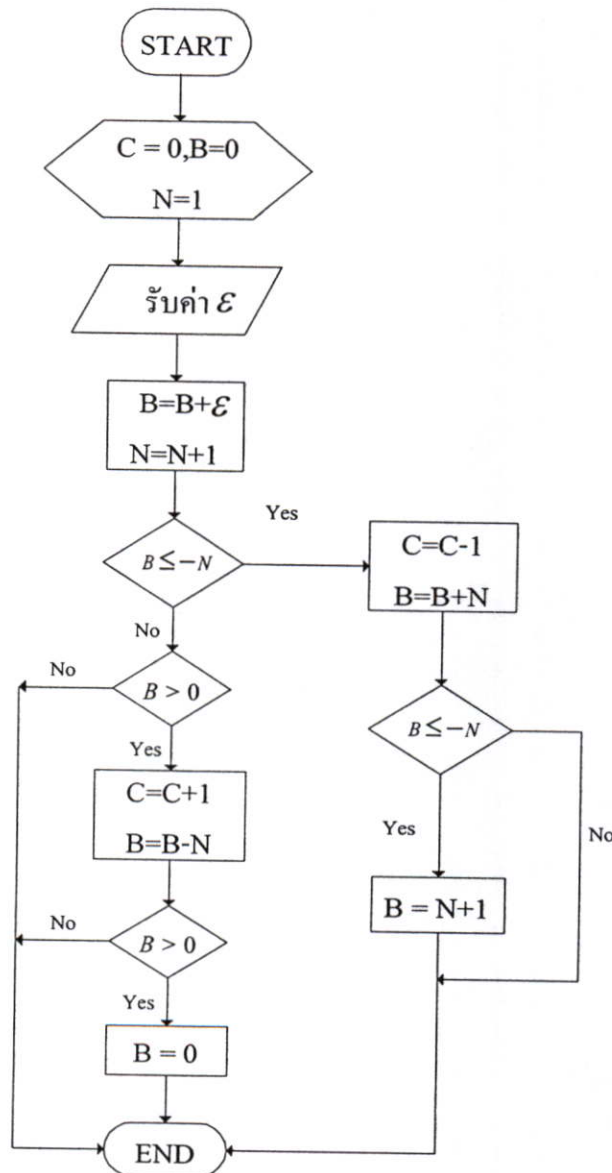
$$P_x = P_x \pm C[Q] \quad (4.8)$$

แต่การหารในสมการที่ 4.7 ไม่สนับสนุนการเป็นอัลกอริทึมแบบซับซ้อนต่ำ (low complexity) และเนื่องด้วยถ้าค่าความผิดพลาดมีค่ามากทำให้ค่า A มีค่ามากและจะส่งผลให้การ return ค่ากลับมาเป็นไปได้ยาก จึงต้องมีการปรับเปลี่ยนรูปแบบสมการใหม่ดังนี้

$$A = N \cdot C' + B' \quad (4.9)$$

เมื่อจำนวนเต็ม B' เป็นไปตาม $-N < B' \leq 0$

จากสมการนั้นเหมือนเป็นการตรวจสอบความผิดพลาดอย่างฉับพลันโดยการตั้งกฎง่ายๆ ว่า การแก้ไขค่าสามารถดำเนินการโดยการส่งค่า B' และ C' และปรับเปลี่ยนค่าทั้งคู่ให้เหมาะสมกับในแต่ละ context วิธีการนี้ขั้นแรกต้องบวกค่า \mathcal{E} กับค่าของ B' ก่อนแล้วค่อยบวก หรือ ลบค่า B' ด้วย N จนกระทั่ง B' มีค่าอยู่ในช่วง $-N < B' \leq 0$ ส่วนตัวเลขที่ใช้ในการบวก หรือ ลบค่า B' ก็คือ N ซึ่งกำหนดจากการปรับค่า C' การทำงานของขบวนการไบอัสสามารถดูได้จาก ไดอะแกรมในภาพที่ 4.5



ภาพที่ 4.5 แสดงการทำงานของขบวนการไบอัส

ในภาพที่ 4.5 ค่า C แทน C' และค่า B แทน B' จากไดอะแกรมพบว่า การแก้ไขค่า C ในแต่ละครั้งที่ $B > 0$ หรือ $B \leq -N$ แล้วค่า B จะถูกปรับตาม C โดยการบวก หรือ ลบ ด้วยค่า N แต่ถ้กรณีที่เปลี่ยน B แล้ว B ยังไม่อยู่ในช่วงที่กำหนดคือ $-N < B \leq 0$ สุดท้ายจะทำการตัดค่าให้อยู่ในช่วงที่กำหนดนั่นเอง เช่นเดียวกันค่า C ที่มีการบวก หรือ ลบ นี้จะถูกจำกัดให้มีค่าอยู่ในช่วง $-128 < C < 127$ ขบวนการนี้จะเฝ้าสังเกตดูให้ค่าเฉลี่ยของเศษเหลือจากการทำนายอยู่ในช่วง $(-1, 0]$ เท่านั้น ส่วนค่า C ที่ใช้เป็นการประมาณของ R (ในหัวข้อการหาค่าพารามิเตอร์) สังเกตว่า $-B/N$ เป็นการประมาณของส่วนที่ไม่สำคัญที่ถูกชิปไป (s)

4.3 การเข้ารหัสในโหมดปกติ (Regular mode encoding)

การเข้ารหัสนี้เป็นการเข้ารหัสค่าเศษเหลือจากการทำนายที่ได้ผ่านการปรับขนาดเซกการไบอัสแล้วจากสมการที่ 4.8

ถ้า I_x คือจุดภาพที่กำลังพิจารณาแล้ว ค่าเศษเหลือจากการทำนายหาได้จากสมการที่ 4.10

$$\bar{E} = I_x - P_x \quad (4.10)$$

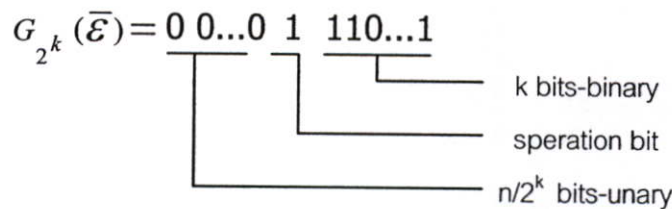
4.3.1 รหัสโกลอมบ์-ไรซ์ (Golomb-Rice)

รหัสโกลอมบ์ (Golomb(G_m)) [8] นี้สามารถคำนวณได้ง่ายโดยไม่ต้องมีตารางรหัสเหมือนเช่นในการเข้ารหัสแบบฮัฟแมน ซึ่งการเข้ารหัสโกลอมบ์ในความหมายคือ การเข้ารหัสค่าที่ไม่เป็นลบของรันเรจจ์ โดยมีรูปแบบคือให้จำนวนเต็มบวก m อยู่ในรูปของ 2 ยกกำลัง k รหัส G_m จะเข้ารหัสค่าจำนวนเต็ม $n \geq 0$ ซึ่งจะประกอบด้วย 2 ส่วนหลัก คือ

1. ส่วนไบนารี (binary) คือ ส่วนเลขฐานสอง จำนวน k บิตที่มีนัยสำคัญต่ำ (LSB)

2. ส่วนยูนารี (unary) คือ ส่วนจำนวนที่เหลือจาก $n/2^k$ โดยแทนด้วยบิต '0'

ทั้งสองส่วนนี้จะมีบิต '1' จำนวน 1 บิตขั้นแยกทั้งสองส่วน ดังนั้นความยาวของรหัสจะเท่ากับ $k + 1 + \lceil n/2^k \rceil$ แผนผังของรหัสแสดงในภาพที่ 4.6



ภาพที่ 4.6 แสดงแผนผังของรหัสโกลอมบ์

อาจกล่าวได้ว่า $G_m = G_{2^k}$ คือ Golomb-power-of-2 (GPO2)

ตัวอย่างการเข้ารหัสโกลอมบ์ ถ้าสมมติ $n = 20 \rightarrow$ '10100' และ $k = 3$

จะได้ว่า ที่ k บิตไบนารี จะมีค่า '100'

ส่วน $n/2^k$ บิต unary จะมีค่า '00' จากค่าบิต '10' = 2 ที่เหลือ
เพราะฉะนั้น จะได้รับรหัส Golomb คือ 001100

ในคุณสมบัติของรหัสที่ดีที่สุด (optimal prefix code) สำหรับ TSGD ใน [6] คือ ช่วงห่างของพารามิเตอร์ (θ, s) ถูกแบ่งภายในย่านขอบเขต และความแตกต่างของ optimal prefix code สอดคล้องกับแต่ละย่าน ในแต่ละย่านขอบเขตมีความเกี่ยวข้องกับลำดับ m ของรหัส Golomb ขณะที่พารามิเตอร์ m คือฟังก์ชันของค่า ของ θ และ s ในขอบเขต บนการอาศัยขอบเขต รหัสที่เหมาะสมที่สุดจากการเข้ารหัสค่าผิดพลาด ($\bar{\mathcal{E}}$) จำนวนเต็ม ทำโดยการประยุกต์แบบอาศัยการปรับปรุงขอบเขตของ G_m กับ $|\bar{\mathcal{E}}|$

จากคุณสมบัติของการเข้ารหัส Golomb ที่จะเข้ารหัสค่าที่ไม่เป็นลบ จึงต้องมีการจัดค่า $\bar{\mathcal{E}}$ ใหม่ มีเงื่อนไขดังสมการที่ 4.11

$$M(\bar{\mathcal{E}}) = 2|\bar{\mathcal{E}}| - u(\bar{\mathcal{E}}) \quad (4.11)$$

โดยมีเงื่อนไขดังสมการที่ 4.12

$$u(\bar{\mathcal{E}}) = \begin{cases} 1 & \text{ถ้า } \bar{\mathcal{E}} < 0 \\ 0 & \text{ถ้า } \bar{\mathcal{E}} \geq 0 \end{cases} \quad (4.12)$$

จากสมการจะสังเกตเห็นว่า ถ้า $\bar{\mathcal{E}} \geq 0$ จะทำให้ได้ $M(\bar{\mathcal{E}})$ เป็นจำนวนคู่
ถ้า $\bar{\mathcal{E}} < 0$ จะทำให้ได้ $M(\bar{\mathcal{E}})$ เป็นจำนวนคี่

การจัดค่าประเภทนี้ได้ใช้โดย Rice [9] เพื่อเข้ารหัสแบบอาศัย TSGD ที่ศูนย์กลางอยู่ที่ศูนย์กลางด้วยการประยุกต์ใช้กับรหัส G_{2^k} เพื่อเข้ารหัสค่า $M(\bar{\mathcal{E}})$ สังเกตว่า $M(\bar{\mathcal{E}})$ เป็นการจัดค่าแบบธรรมดาภายใต้การสมมติในกรณี $s \leq 1/2$ การจัดค่าเศษเหลือจะเป็น $M(\bar{\mathcal{E}})$ แต่ถ้า $s > 1/2$ จะได้ $M'(\bar{\mathcal{E}}) = M(-\bar{\mathcal{E}} - 1)$ ซึ่งมีผลต่อการจัดประเภทของความน่าจะเป็นในกรณีนี้ จากในหัวข้อค่าผิดพลาดที่เหลือจากการทำนายจะมีค่า $\bar{\mathcal{E}}$ อยู่ในช่วง $-(\alpha/2) \leq \bar{\mathcal{E}} \leq (\alpha/2) - 1$ และเมื่อมีการจัดค่า $\bar{\mathcal{E}}$ ใหม่จะทำให้ $M(\bar{\mathcal{E}})$ มีค่าอยู่ในช่วง $0 \leq M(\bar{\mathcal{E}}) \leq \alpha - 1$

4.3.2 การหาค่า k ที่เหมาะสมสำหรับการเข้ารหัส

ในการหาค่า k สำหรับการเข้ารหัส Golomb-ไรซ์ จะมีค่าพารามิเตอร์ 2 ตัวที่เกี่ยวข้องคือ

N คือ จำนวนนับครั้งของเศษเหลือจากการทำนายในแต่ละ context

A คือ ขนาดเศษเหลือจากการทำนายสะสมในแต่ละ context

โดยสามารถหาค่า k ได้จากสมการที่ 4.13

$$k = \min \left\{ k' \mid 2^{k'} N \geq A \right\} \quad (4.13)$$

4.3.3 การจำกัดความยาวของรหัส

ในขบวนการเข้ารหัสที่กล่าวมาแล้ว อาจสร้างความยาวรหัสที่มากเกินไปได้ เช่น ในกรณีที่ $\alpha = 256$, $k=0$, $\bar{\mathcal{E}} = -128 \rightarrow M(\bar{\mathcal{E}}) = 256$ จะสามารถสร้างรหัสได้ถึง 256 บิต จึงต้องมีการจำกัดค่าความยาวนี้ไม่ให้เกิน $L_{\max} = 4\beta$ ขณะที่ $\beta = \lceil \log \alpha \rceil$

สำหรับในกรณีที่ภาพเป็น 8 บิตต่อพิกเซล ค่าความยาวจึงจำกัดอยู่ที่ 32 ซึ่งคิดได้จาก

$$\begin{aligned} \beta &= \log_2 2^8 \\ \beta &= 8 \\ \therefore L_{\max} &= 4\beta = 32 \end{aligned}$$

เทคนิคการจำกัดค่าความยาวรหัสนี้ได้จาก [10] ส่วนที่ทำให้รหัสมีความยาวมากมาจากส่วนของบิตยูนารี ในกรณีทำการเข้ารหัสค่า y , $0 \leq y \leq \alpha$ เมื่อ บิตยูนารี $= q(y) = \lfloor 2^{-k} y \rfloor$ ซึ่งค่า $q(y)$ จำกัดอยู่ในช่วงซึ่งแสดงในสมการที่ 4.14

$$q(y) < L_{\max} - \beta - 1 = q_{\max} \quad (4.14)$$

ขณะที่ สมมติให้ $L_{\max} > \beta + 1$ เช่นเดียวกันค่า k ที่ใช้ต้องมีการจำกัดขอบเขตเช่นกันดังสมการที่ 4.15

$$k \leq \log(\alpha/2) \leq \beta - 1 \quad (4.15)$$

จากสมการที่ 4.14 ถ้า $q(y) \geq q_{\max}$ แล้ว q_{\max} จะถูกเข้ารหัสในส่วนของบิตยูนารี ส่วนในบิตไบนารีจะแทนด้วยค่าบิตไบนารีของ $y-1$ ซึ่งจะมีค่าไม่เกิน β บิต การกระทำเช่นนี้ถือได้ว่าเป็นการเข้ารหัสที่มีความยาวมาก ถึงแม้ว่าจะดูเหมือนส่วนของบิตยูนารีจะมีมากขึ้น แต่มันจะช่วยให้กระบวนการถอดรหัสง่ายขึ้น

4.4 การเข้ารหัสในโหมดรัน (Run mode encoding)

เป็นส่วนของการเข้ารหัสค่าจำนวนนับครั้งที่เท่ากันของตำแหน่งที่กำลังพิจารณา (x) กับค่าในตำแหน่งถัดไป (a) หรือกล่าวได้ว่าเข้ารหัสจำนวนนับครั้งที่ $x = a$ แต่เมื่อใดที่ $x \neq a$ และจบแถวที่พิจารณา การเข้ารหัสจะไปใน ส่วนรันอินเทอร์รัพ การเข้ารหัสในส่วนนี้แตกต่างจาก โหมดปกติตรงที่รหัสที่ใช้แทนจำนวนนับครั้งนี้แทนด้วยบิตที่มีค่า '1' แล้วปิดท้ายด้วย '0' รูปแบบการเข้ารหัสเป็นดังนี้

$$0, 10, 110, \dots, 1^{m-1}0, 1^m$$

โดยที่ m คือจำนวนนับครั้งที่ $x = a$ ก่อนมีการอินเทอร์รัพ

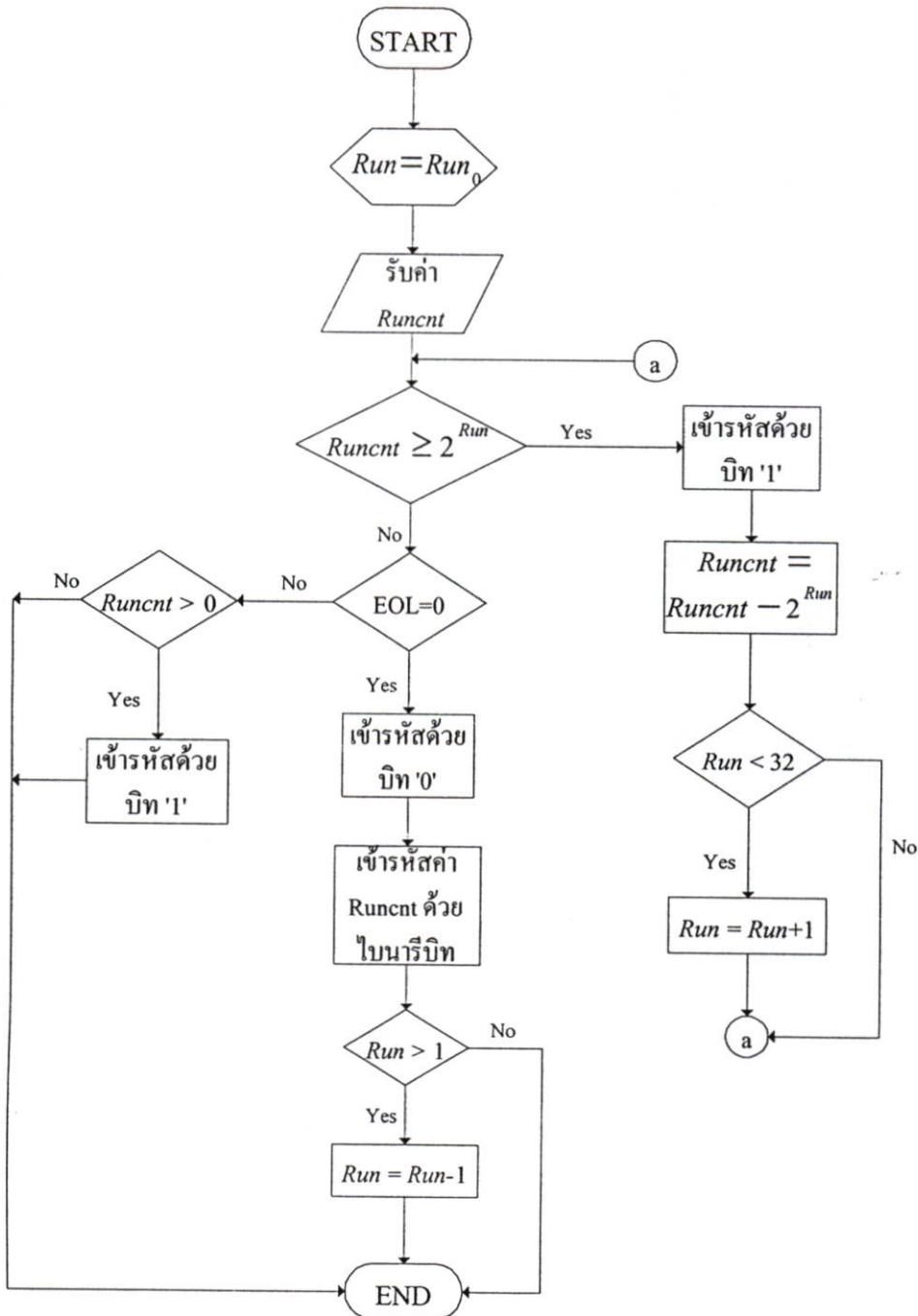
อาจกล่าวได้ว่าการเข้ารหัสแบบนี้เป็นการเข้ารหัสแบบรันเรจ [8] อย่างไรก็ตามการเข้ารหัสวิธีนี้ก็อยู่บนพื้นฐานของรหัสโคลอมบ์ โดยมีการปรับลดค่าของจำนวน m ให้ m มีค่าน้อยลงเพื่อลดความยาวของรหัส โดยใช้ความสัมพันธ์ของลำดับในตารางดัชนี

ตารางลำดับดัชนีในโหมดรัน (Run Index (*Run*)) ที่ใช้ในวิทยานิพนธ์นี้ และใน [7] มีค่าเป็น

$$Run = [0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 9, 10, 11, 12, 13, 14, 15]$$

ในส่วนการเข้ารหัสรันอินเทอร์รัพ จะคล้ายกับในส่วนโหมดปกติ ซึ่งมีไว้บอกค่าความต่างของ x กับ a ตอนที่อยู่ในโหมดรัน การเข้ารหัสในโหมดรันดูได้จากภาพที่ 4.7

จากภาพที่ 4.7 จะเห็นว่ามีการเซตค่าดัชนี *Run* เริ่มต้นให้เป็น *Run* ที่ตำแหน่ง 0 ซึ่งจากลำดับดัชนีมีค่าเท่ากับ 0 และจากนั้นค่าดัชนี *Run* จะมีการปรับเปลี่ยนอยู่ตลอดเวลาเพื่อให้ได้ความยาวรหัสที่เหมาะสมที่สุด ส่วนค่า Run_{cnt} คือ ค่าจำนวนนับครั้งที่ $x = a$ นอกจากนี้การเข้ารหัสยังมีการเพิ่มบิต '0' หรือ '1' เพื่อบ่งบอกว่าการสิ้นสุดแถวที่พิจารณาหรือไม่โดยดูจากค่า EOL ซึ่งค่านี้จะถูกกำหนดให้เป็น 0 ถ้ายังไม่สิ้นสุดแถว ในทางกลับกันกำหนดให้เป็น 1 และในส่วนการเพิ่มรหัสค่า Run_{cnt} ที่เหลือด้วยบิตไบนารี เพื่อบอกถึงสถานะและตำแหน่งของดัชนี *Run* ขึ้นตอนทั้งหมดนี้เพื่อความสะดวกในการพิจารณาในส่วนการถอดรหัส



ภาพที่ 4.7 แสดงการเข้ารหัสในโหนดต้นไม้

4.5 การรีเซตค่าที่เกี่ยวข้องกับ context

เนื่องจากข้อมูลทางสถิติของ context ที่พิจารณา ในแต่ละ context มีการปรับเปลี่ยนรูปแบบไม่คงที่ จึงต้องมีการรีเซตค่าตัวแปร N , A และ B เป็นช่วงๆ โดยมีเงื่อนไขจากค่า N_0 ซึ่งในบทนี้ได้ใช้ $N_0 = 64$ [7] ถือเป็นช่วงที่ทำงานได้ดี การรีเซตมีเงื่อนไขดังนี้

กรณี ถ้า $N = N_0$ แล้วจะรีเซตค่าต่างๆได้เป็น

$$A = \frac{A}{2}$$

$$B = \frac{B}{2}$$

$$N = \frac{N}{2}$$

4.6 สรุปขั้นตอนในการบีบอัดข้อมูลแบบอาศัยข้อมูลแวดล้อม

4.6.1 สรุปขั้นตอนหลักในโหมคปกติ

ขั้นตอนในส่วนนี้เป็นขั้นตอนหลักซึ่งจะประกอบด้วย ส่วนการกำหนดค่าเริ่มต้น การหาข้อมูลแวดล้อม การเลือกโหมคการทำงาน และการเข้ารหัสในโหมคปกติ สามารถสรุปเป็นขั้นตอนดังนี้

ขั้นตอนกำหนดค่าเริ่มต้น

1). กำหนดค่า $L_{\max} = 2(\beta_{\max} + \max\{8, \beta_{\max}\})$ ขณะที่ $\beta_{\max} = \max\{2, \lceil \log \alpha \rceil\}$

เมื่อ β คือ จำนวนบิตที่ใช้แทนข้อมูล และ $\alpha = 2^\beta$

2). กำหนดให้จำนวนลำดับของ A, B, C และ N มีจำนวนลำดับเท่ากับค่าจำนวน context คือ 365 และให้ค่าขนาดของ $B = C = 0, N = 1$ ส่วน $A = \max\{2, \lceil (\alpha + 32) / 64 \rceil\}$ กำหนดจำนวนลำดับ A และ N เพิ่มอีก 2 ลำดับเพื่อใช้ใน รันอินเทอร์รัพ

3). กำหนดดัชนี *Run* เริ่มต้นที่ตำแหน่งลำดับ 0

4). กำหนดให้ x เป็นพิกเซลแรกในภาพ

ขั้นตอนที่ 1

คำนวณค่าโลกอลเกรเดียนท์ $\rightarrow D_1 = R_d - R_b, D_2 = R_b - R_c, D_3 = R_c - R_a$

ขั้นตอนที่ 2

เลือกโหมคการทำงาน โดยมีเงื่อนไขว่าถ้า $D_1 = D_2 = D_3 = 0 \rightarrow$ เข้าสู่โหมครันแต่ถ้าไม่ใช่จะเข้าสู่โหมคปกติ

ขั้นตอนที่ 3

ควอนไทซ์ค่าโลกอลเกรเดียนท์ (ดูใน หัวข้อ 4.2.4.2)

ขั้นตอนที่ 4

แทนโลกอลเกรเดียนท์ด้วย $q_i, i=1,2,3$ จากนั้นกลับเครื่องหมายค่า q_i และจัดและรวมค่า q_i ใหม่ไปเป็นค่า Q ตามเงื่อนไขและสมการในหัวข้อ 4.2.4.2

ขั้นตอนที่ 5

ทำการทำนายค่าตามสมการที่ 4.3

ขั้นตอนที่ 6

แก้ไขค่าทำนายโดยการบวก หรือ ลบ ค่าของ C ตามเงื่อนไขใน context นั้นๆ ด้วยสมการ $P_x = P_x \pm C[Q]$ แล้วต้องจำกัดค่าทำนายที่แก้ไขแล้วให้มีค่าอยู่ในช่วง $[0, \alpha-1]$

ขั้นตอนที่ 7

คำนวณเศษเหลือจากการทำนายด้วย $\bar{E} = I_x - P_x$ และถ้าเกี่ยวข้องกับเครื่องหมาย context ที่ เป็นลบให้ปรับค่า $-\bar{E} \rightarrow \bar{E}$ แล้วปรับลดค่า \bar{E} ให้อยู่ในช่วง $[-\lfloor \alpha/2 \rfloor, \lceil \alpha/2 \rceil - 1]$

ขั้นตอนที่ 8

คำนวณค่า k ในสมการที่ 4.13

ขั้นตอนที่ 9

แมปค่า $\bar{E} \rightarrow M(\bar{E})$ หรือ ถ้า $k=0$ และ $2B \leq -N \rightarrow M'(\bar{E})$

ขั้นตอนที่ 10

นำค่า $M(\bar{E})$ มาเข้ารหัสด้วยรหัสโคโลมบ์-ไรซ์ โดยใช้ค่า k ที่คำนวณได้ในขั้นที่ 8 และ พิจารณารหัสที่มีค่าเกิน L_{\max} ให้เข้ารหัสในหัวข้อที่ 4.3.3

ขั้นตอนที่ 11

ปรับค่าดัชนีในแต่ละ context โดยบวก \bar{E} กับ B และบวก $|\bar{E}|$ กับ A แล้วพิจารณาว่าถ้า $N = N_0 = 64$ ต้องทำการปรับให้ A, B และ $N = \frac{A}{2}, \frac{B}{2}, \frac{N}{2}$

ขั้นตอนที่ 12

ปรับค่า B และ C ตามเงื่อนไขในบล็อกไดอะแกรมในภาพที่ 4.5

ขั้นตอนที่ 13

กลับไปยังขั้นตอนที่ 1 และรับค่าข้อมูลในอินพุตถัดไป

4.6.2 สรุปขั้นตอนในโหมดรัน

ขั้นตอนที่ 1

อ่านค่าของ sample ใหม่จนกระทั่ง $x \neq a$ หรือจนสิ้นสุดแถวในภาพที่พิจารณา

ขั้นตอนที่ 2

ให้ $m = 2^s$ แทนพารามิเตอร์ปัจจุบันขององค์ประกอบรหัส Golomb และเข้ารหัสแบบรันแรงจี้ ด้วยบิต '1' แล้วปรับค่าดัชนี Run ตามบล็อกไดอะแกรมในภาพที่ 4.7

ขั้นตอนที่ 3

ถ้าโหมดรันถูกขัดจังหวะด้วยการสิ้นสุดแถว ให้เพิ่มบิต '1' ในบิตเอาต์พุตของรันโหมดแล้ว กลับไปยังขั้นตอนที่ 1 ในโหมดปกติ แต่ถ้ายังไม่สิ้นสุดแถวให้เพิ่มบิต '0' แทน และตามด้วยไบ นารีบิตที่แทนค่าที่เหลือของรันแรงจี้จากการใช้ g บิต จากนั้นลดดัชนี Run

ขั้นตอนที่ 4

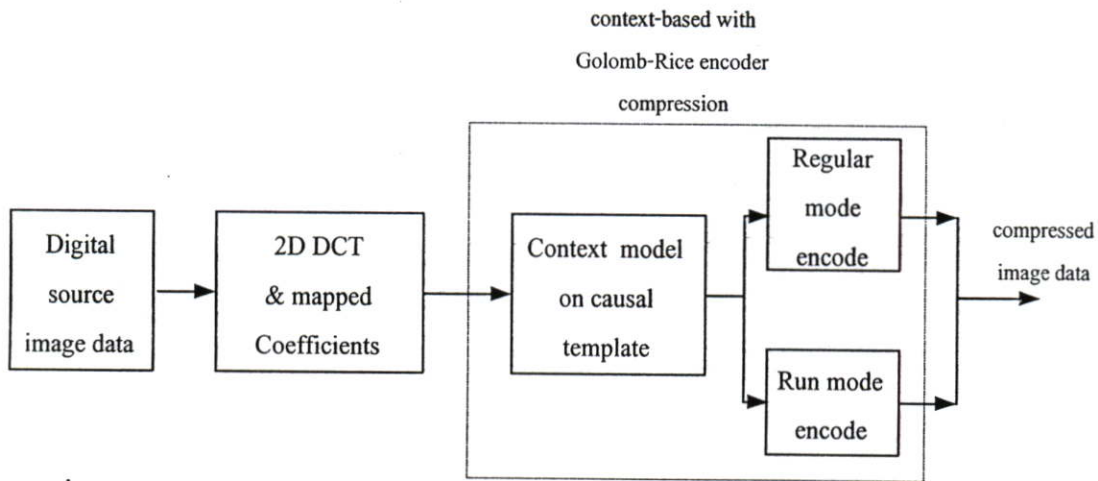
เข้ารหัสรันอินเทอร์รัพ และกลับไปยังขั้นตอนที่ 1 ในโหมดปกติ

บทที่ 5

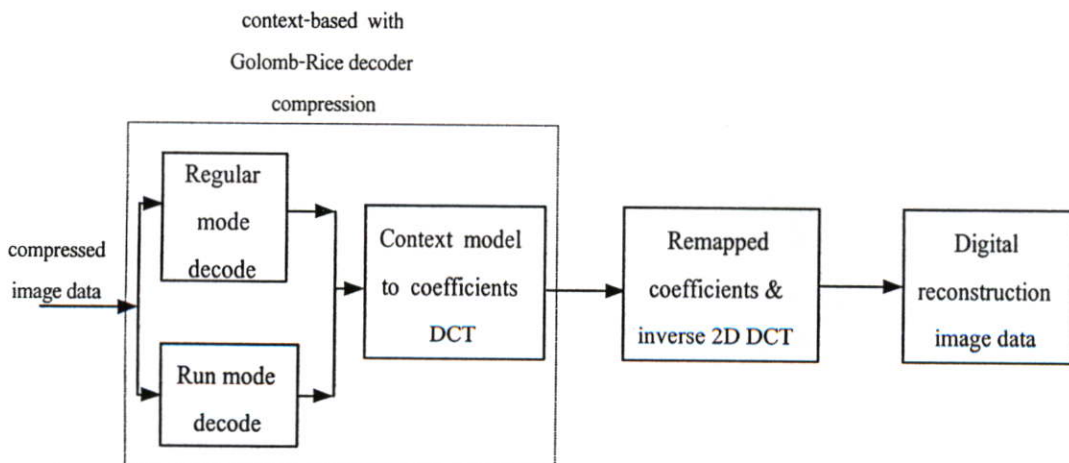
การบีบอัดข้อมูลแบบมีการสูญเสียต่ำชนิดความซับซ้อนต่ำ

5.1 ขบวนการบีบอัดข้อมูลที่น่าสนใจ

จากอัลกอริธึมที่ซับซ้อนต่ำของวิธีการบีบอัดข้อมูลแบบอาศัยข้อมูลแวดล้อมในบทที่ 4 ซึ่งชี้แจงจำกัดของประสิทธิภาพในการบีบอัดข้อมูลขึ้นอยู่กับความต่อเนื่องของภาพ ในบทนี้จึงได้นำการแปลงแบบดิสครีตโคไซน์ ในบทที่ 3 มาประยุกต์ใช้ร่วมกับการบีบอัดข้อมูลแบบอาศัยข้อมูลแวดล้อม เพื่อให้เพิ่มขีดความสามารถในการลดข้อมูลให้ใช้ได้กับทั้งภาพที่มีความต่อเนื่องต่ำ โดยวิธีการบีบอัดข้อมูลในบทนี้มีขั้นตอนการทำงานดังภาพที่ 5.1 และขั้นตอนการแปลงกลับดังภาพที่ 5.2



ภาพที่ 5.1 แสดงการบีบอัดข้อมูลแบบอาศัยข้อมูลแวดล้อมร่วมกับการแปลงดิสครีตโคไซน์



ภาพที่ 5.2 แสดงการนำข้อมูลกลับจากการบีบอัดข้อมูลแบบอาศัยข้อมูลแวดล้อมร่วมกับการแปลงแบบดิสครีตโคไซน์

ในกระบวนการแปลงคิงภาพที่ 5.1 เปรียบเสมือนการปรับปรุงภาพก่อนทำการบีบอัดข้อมูล ซึ่งมีขั้นตอนโดยสรุปพอสังเขปดังนี้ กระบวนการจะเริ่มต้นจากการนำข้อมูลภาพต้นแบบมาทำการแปลงแบบดิสครีตโคซายน์แบบ 2 มิติ โดยใช้บล็อกขนาด 8×8 และจากนั้นทำการปรับปรุงค่าสัมประสิทธิ์ดิสครีตโคซายน์ และ จัดลำดับค่าสัมประสิทธิ์ใหม่เพื่อให้ภาพมีความต่อเนื่องมากขึ้น ดูรายละเอียดเพิ่มเติมได้ในบทที่ 3

- 1) แบ่งภาพต้นแบบเป็นบล็อกเท่าๆ กันขนาด 8×8 พิกเซล
- 2) ทำการแปลงแบบดิสครีตโคซายน์ 2 มิติ กับบล็อก 8×8 ทุกบล็อก
- 3) ปรับปรุงค่าสัมประสิทธิ์เพื่อให้เหมาะสมกับการใช้งาน คือ ให้มีค่าอยู่ในช่วง 0-255
- 4) จัดลำดับค่าสัมประสิทธิ์ใหม่

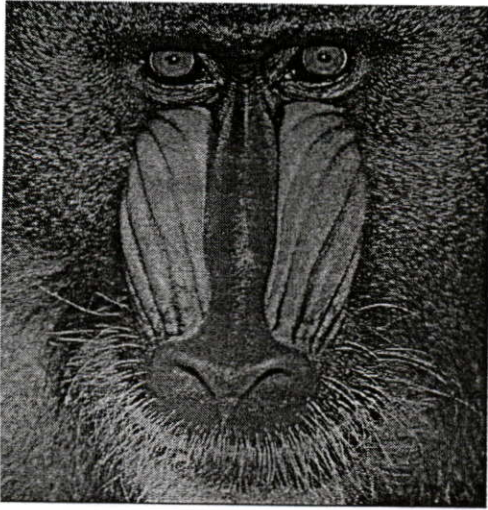
หลังจากผ่านขบวนการทั้ง 4 มาแล้ว จากนั้นเข้าสู่ขบวนการบีบอัดข้อมูลแบบอาศัยข้อมูลแวดล้อมขั้นตอนของขบวนการบีบอัดข้อมูลนี้ดูรายละเอียดเพิ่มเติมในบทที่ 4 ซึ่งอาจจะสรุปได้ดังนี้

- 1) กำหนดข้อมูลแวดล้อมเริ่มต้นด้วย causal template
- 2) เลือกโหมดการทำงานด้วยการคำนวณ local gradient
 - 2.1) โหมดปกติ
 - 2.1.1) ทำการทำนายค่า และปรับชดเชยค่าที่ได้จากการทำนาย
 - 2.1.2) คำนวณเศษเหลือจากการทำนายและเข้ารหัสค่าเศษเหลือด้วยรหัส Golomb-ไรซ์
 - 2.2) โหมดรัน
 - 2.2.1) นับจำนวนครั้งที่ $x = a$
 - 2.2.2) เข้ารหัสแบบรันเรนจ์บนพื้นฐานของรหัส Golomb

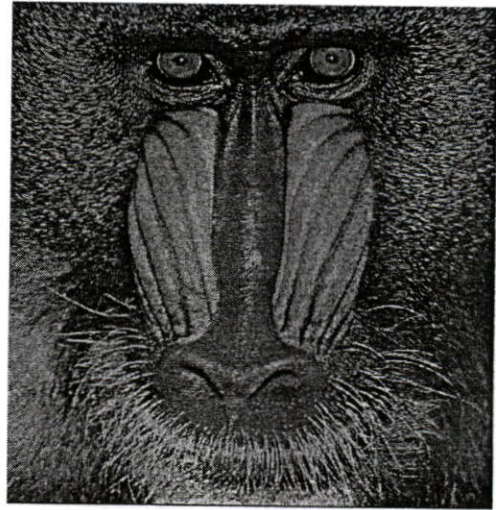
5.2 ผลการทดลอง

ในวิทยานิพนธ์ฉบับนี้ได้ทำการทดลองบีบอัดข้อมูลภาพขนาด 512×512 และ 256×256 รายละเอียดต่อจุดภาพขนาด 8 บิตต่อพิกเซล ซึ่งภาพขนาด 512×512 มีทั้งหมด 5 ภาพ คือ ภาพ Baboon , ภาพ Airplane , ภาพ Lena , ภาพ Man และ ภาพ Peppers ส่วนภาพขนาด 256×256 มีทั้งหมด 5 ภาพ คือ ภาพ Camera man , ภาพถ่ายดาวเทียมแสดงรายละเอียดพื้นที่บางส่วนของเกาะฮ่องกง (จากดาวเทียม Lansat TM) , ภาพถ่ายดาวเทียมแสดงรายละเอียดของนิวออร์ลีอันในหลุยส์เซียนา (จากดาวเทียม Jers-1 OPS) , ภาพถ่ายดาวเทียมแสดงรายละเอียดชุมชนมนุษย์ในแม่น้ำมิสซิสซิปปี และแม่น้ำอิลลินอยล์ (จากดาวเทียม ERS-1 SAR) และภาพถ่ายดาวเทียมแสดงรายละเอียดชายหาดนิวพอร์ตในแคลิฟอร์เนีย(จากดาวเทียม JERS-1 SAR) ด้วยวิธีการบีบอัดข้อมูลแบบต่างๆ เช่น วิธีการบีบอัดข้อมูลแบบอาศัยข้อมูลแวดล้อม[5] วิธีการบีบอัดข้อมูลแบบ JPEG-baseline ซึ่งทดลองโดยใช้โปรแกรม Microsoft Photo Editor โดยได้เลือกค่า Quality Factor (Q.F) ต่างๆ ที่ทำ

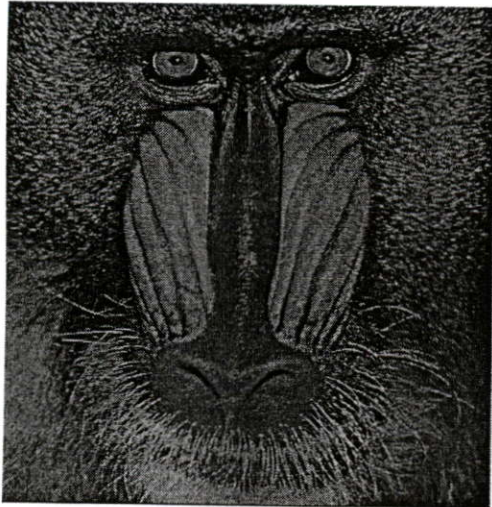
ให้ค่า (Mean square error (MSE)) มีค่าใกล้เคียงกับวิธีการบีบอัดข้อมูลแบบมีการสูญเสียต่ำชนิดความซับซ้อนต่ำเพื่อเปรียบเทียบประสิทธิภาพการบีบอัดในกรณีที่มีค่า MSE ใกล้เคียงกัน และสุดท้ายเป็นวิธีการบีบอัดข้อมูลแบบมีการสูญเสียต่ำชนิดความซับซ้อนต่ำที่ได้นำเสนอในบทนี้ ผลลัพธ์แสดงไว้ในภาพที่ (5.3 (a)-(d)) - (5.12 (a)-(d)) จากนั้นได้นำผลลัพธ์ทั้งสามวิธีมาทำการคำนวณค่า MSE และ (Compression ratio (CR)) เปรียบเทียบกัน ซึ่งแสดงผลในตารางที่ 5.1



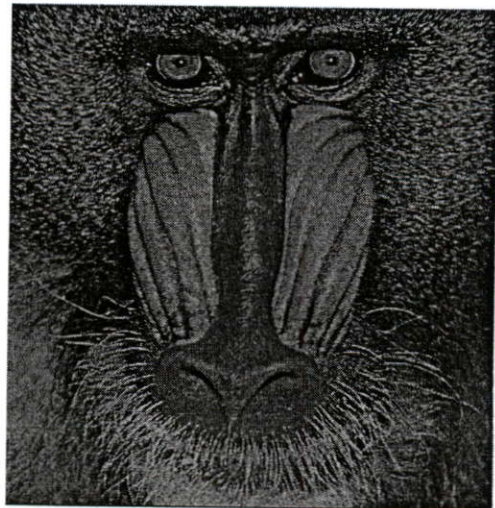
(a) ภาพต้นแบบ Baboon
ขนาด 512×512 , 8 bpp



(b) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบอาศัยข้อมูลเวดล้อม [5] มีค่า $CR=1.376, MSE=0$

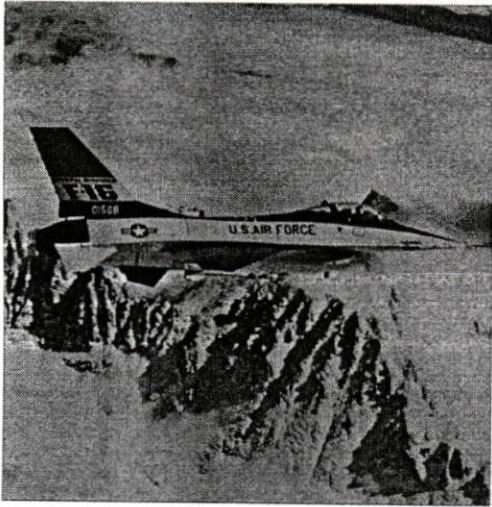


(c) ภาพผลลัพธ์จากวิธี JPEG-Baseline
 $QF=94%$, $CR=1.755$, $MSE=3.06$



(d) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบมีการสูญเสียต่ำชนิดความซับซ้อนต่ำ
มีค่า $CR=2.154$, $MSE=2.52$

ภาพที่ 5.3 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบ Baboon ด้วยวิธีการบีบอัดข้อมูลที่แตกต่างกัน



(a) ภาพต้นแบบ Airplane
ขนาด 512×512,8 bpp



(b) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบอาศัย
ข้อมูลเวดล้อม [5]
มีค่า CR=2.218,MSE=0



(c) ภาพผลลัพธ์จากวิธี JPEG-Baseline
QF=90% ,CR=3.590 ,MSE=2.18



(d) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบมีการ
สูญเสียต่ำชนิดความซับซ้อนต่ำ
มีค่า CR=4.228,MSE=2.07

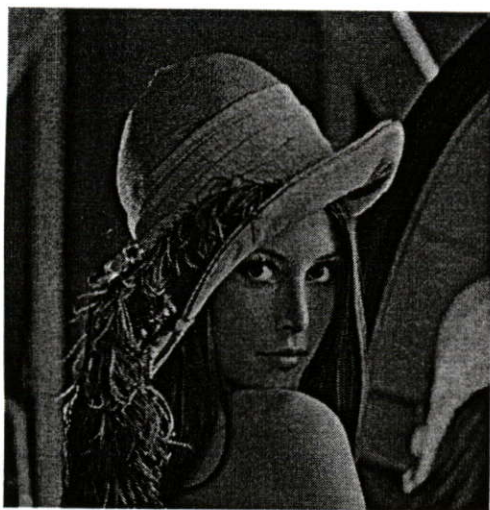
ภาพที่ 5.4 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบ Airplane ด้วยวิธีการบีบอัดข้อมูล
ที่ต่างกัน



(a) ภาพต้นแบบ Lena
ขนาด 512×512,8 bpp



(b) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบอาศัย
ข้อมูลเวดล้อม [5]
มีค่า CR=1.888,MSE=0

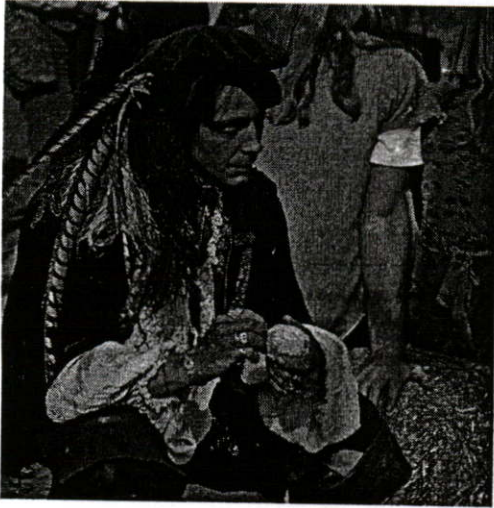


(c) ภาพผลลัพธ์จากวิธี JPEG-Baseline
QF=92% ,CR=3.19 ,MSE=3.35

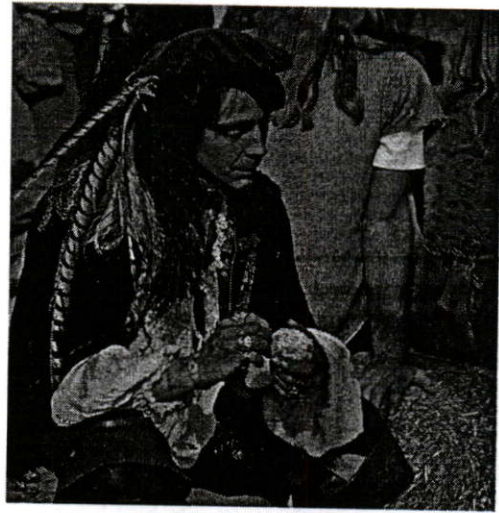


(d) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบมีการ
สูญเสียค่าชนิดความซับซ้อนต่ำ
มีค่า CR=3.94,MSE=3.204

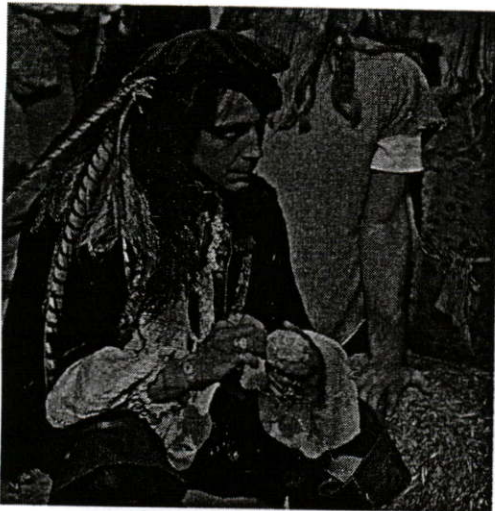
ภาพที่ 5.5 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบ Lena ด้วยวิธีการบีบอัดข้อมูลที่แตกต่างกัน



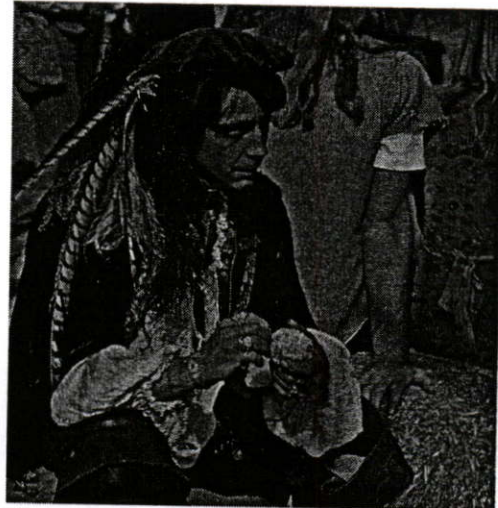
(a) ภาพต้นแบบ Man
ขนาด 512×512,8 bpp



(b) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบอาศัย
ข้อมูลแวดล้อม [5]
มีค่า CR=1.519,MSE=0

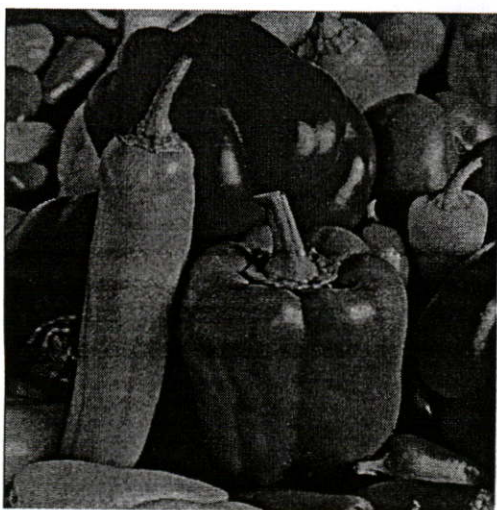


(c) ภาพผลลัพธ์จากวิธี JPEG-Baseline
QF=93% ,CR=1.911 ,MSE=2.99

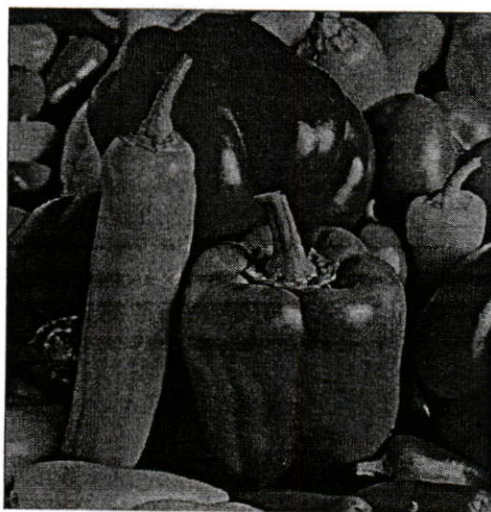


(d) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบมีการ
สูญเสียค่าขนาดความซับซ้อนต่ำ
มีค่า CR=2.540,MSE=2.818

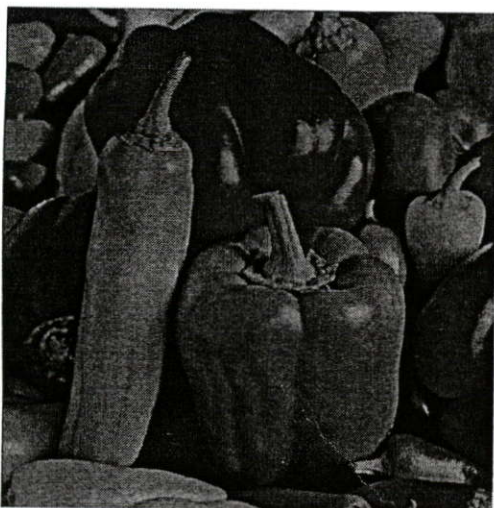
ภาพที่ 5.6 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบ Man ด้วยวิธีวิธีการบีบอัดข้อมูล
ที่ต่างกัน



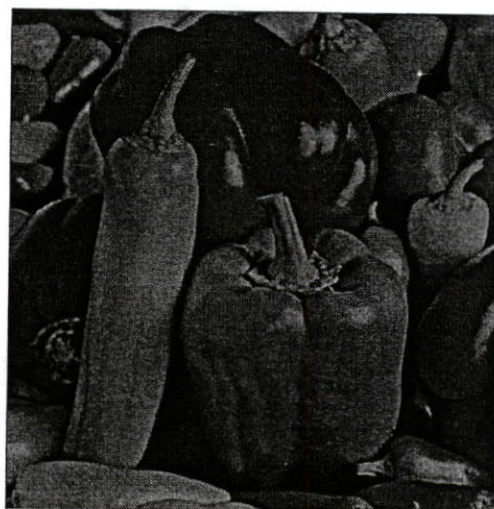
(a) ภาพต้นแบบ Peppers
ขนาด 512×512,8 bpp



(b) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบอาศัย
ข้อมูลเวคัล้อม [5]
มีค่า CR=1.865,MSE=0



(c) ภาพผลลัพธ์จากวิธี JPEG-Baseline
QF=93% ,CR=2.629 ,MSE=2.76



(d) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบมีการ
สูญเสียต่ำชนิดความซับซ้อนต่ำ
มีค่า CR=3.43,MSE=2.72

ภาพที่ 5.7 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบ Peppers ด้วยวิธีการบีบอัดข้อมูล
ที่ต่างกัน



(a) ภาพต้นแบบ Camera man
ขนาด 256×256,8 bpp



(b) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบอาศัย
ข้อมูลเวดล้อม [5]
มีค่า CR=1.688,MSE=0



(c) ภาพผลลัพธ์จากวิธี JPEG-Baseline
QF=93% ,CR=2.225 ,MSE=2.501



(d) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบมีการ
สูญเสียต่ำชนิดความซับซ้อนต่ำ
มีค่า CR=2.590,MSE=2.43

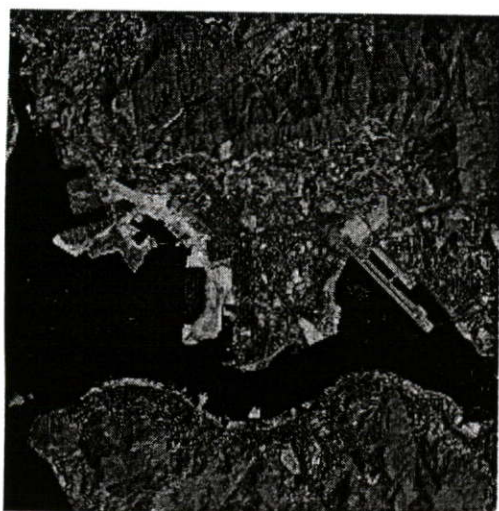
ภาพที่ 5.8 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบ Camera man ด้วยวิธีการบีบอัดข้อมูลที่แตกต่างกัน



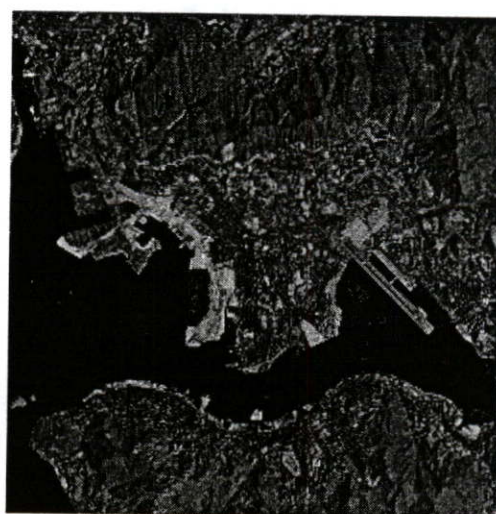
(a) ภาพต้นแบบจากดาวเทียม Lansat TM
ขนาด 256×256,8 bpp



(b) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบอาศัย
ข้อมูลเวดล้อม [5]
มีค่า CR=1.345,MSE=0

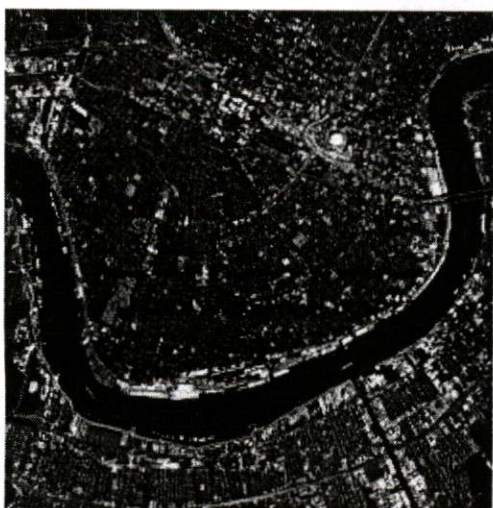


(c) ภาพผลลัพธ์จากวิธี JPEG-Baseline
QF=93% ,CR=1.61 ,MSE=2.726



(d) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบมีการ
สูญเสียค่าชนิดความชัดซ้อนดำ
มีค่า CR=1.969,MSE=2.65

ภาพที่ 5.9 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบจากดาวเทียม Lansat TM ด้วยวิธี
การบีบอัดข้อมูลที่แตกต่างกัน



(a) ภาพต้นแบบ Jers-1 OPS
ขนาด 256×256 , 8 bpp



(b) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบอาศัย
ข้อมูลเวดล้อม [5]
มีค่า $CR=1.25, MSE=0$



(c) ภาพผลลัพธ์จากวิธี JPEG-Baseline
 $QF=93\%$, $CR=1.52$, $MSE=3.09$

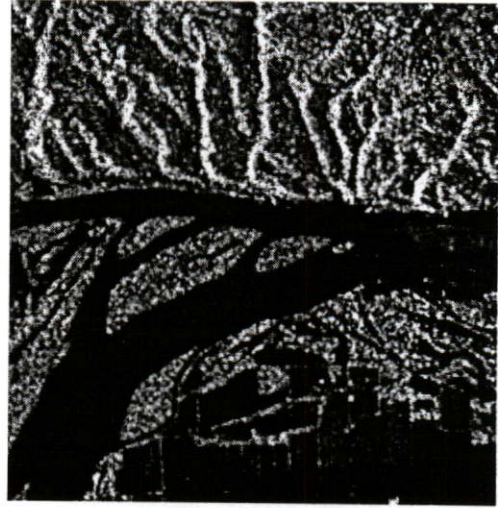


(d) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบมีการ
สูญเสียต่ำชนิดความซับซ้อนต่ำ
มีค่า $CR=1.817, MSE=2.99$

ภาพที่ 5.10 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบจากดาวเทียม Jers-1 OPS ด้วยวิธี
การบีบอัดข้อมูลที่แตกต่างกัน



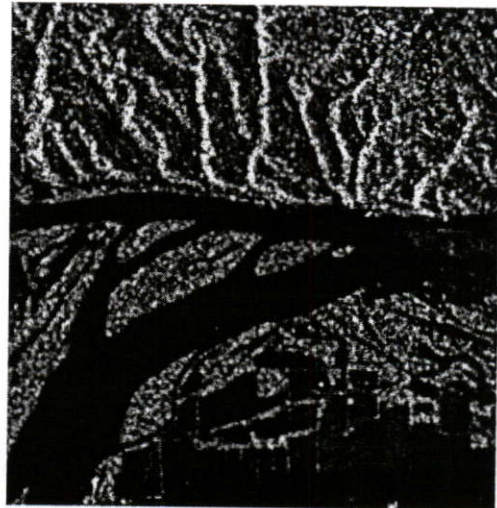
(a) ภาพต้นแบบจากดาวเทียม ERS-1 SAR
ขนาด 256×256 , 8 bpp



(b) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบอาศัย
ข้อมูลเวดล้อม [5]
มีค่า $CR=1.394, MSE=0$

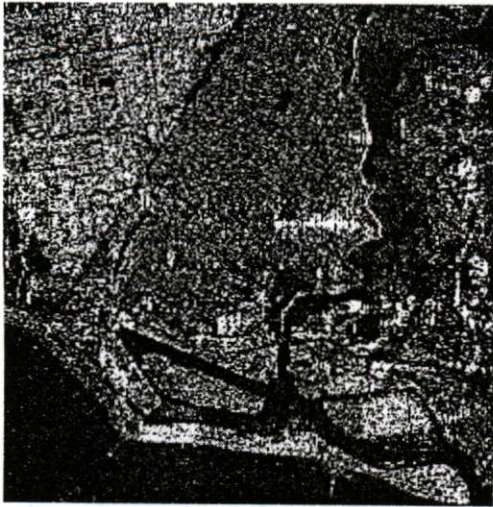


(c) ภาพผลลัพธ์จากวิธี JPEG-Baseline
 $QF=93\%$, $CR=1.607$, $MSE=2.714$



(d) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบมีการ
สูญเสียค่าชนิดความซับซ้อนต่ำ
มีค่า $CR=1.958, MSE=2.611$

ภาพที่ 5.11 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบจากดาวเทียม ERS-1 SAR ด้วย
วิธีการบีบอัดข้อมูลที่แตกต่างกัน



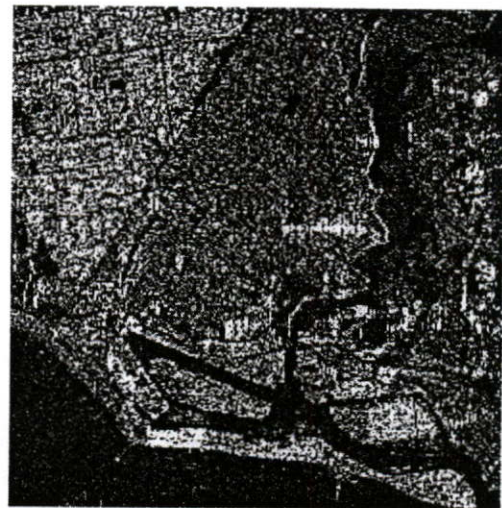
(a) ภาพต้นแบบจากดาวเทียม Jer-1 SAR
ขนาด 256×256 , 8 bpp



(b) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบอาศัย
ข้อมูลเวดล้อม [5]
มีค่า $CR=1.014, MSE=0$



(c) ภาพผลลัพธ์จากวิธี JPEG-Baseline
 $QF=93\%$, $CR=1.253$, $MSE=3.174$



(d) ภาพผลลัพธ์จากวิธีบีบอัดข้อมูลแบบมีการ
สูญเสียต่ำชนิดความซับซ้อนต่ำ
มีค่า $CR=1.45, MSE=2.951$

ภาพที่ 5.12 (a)-(d) แสดงผลการทดลองบีบอัดข้อมูลภาพต้นแบบจากดาวเทียม Jers-1 SAR ด้วยวิธี
การบีบอัดข้อมูลที่แตกต่างกัน

ตารางที่ 5.1 แสดงการเปรียบเทียบวิธีการบีบอัดข้อมูลภาพแบบต่างๆ

ภาพต้นแบบ	วิธีการบีบอัดข้อมูล					
	แบบอาศัยข้อมูลแวดล้อม[5]		แบบ JPEG -baseline		แบบมีการสูญเสียต่ำ ชนิดความซับซ้อนต่ำ	
	CR	MSE	CR	MSE	CR	MSE
Baboon	1.376	0	1.755	3.06	2.154	2.52
Airplane	2.218	0	3.590	2.18	4.228	2.07
Lena	1.888	0	3.190	3.35	3.944	3.20
Man	1.519	0	1.911	2.99	2.540	2.81
Peppers	1.865	0	2.629	2.76	3.433	2.72
Camera man	1.688	0	2.225	2.50	2.590	2.43
Lansat TM	1.345	0	1.610	2.72	1.969	2.65
Jers-1 OPS	1.255	0	1.522	3.09	1.817	2.99
Ers-1 SAR	1.394	0	1.607	2.71	1.958	2.61
Jers-1 SAR	1.014	0	1.253	3.17	1.450	2.95

5.3 สรุปผลการทดลอง

จากผลการทดลองพบว่าผลลัพธ์ที่ได้จากการบีบอัดข้อมูลแบบมีการสูญเสียต่ำชนิดความซับซ้อนต่ำที่นำเสนอในวิทยานิพนธ์นี้ มีอัตราการบีบอัดข้อมูล (CR) ที่สูงที่สุดเมื่อเทียบกับอีก 2 วิธี และเมื่อพิจารณาที่อัตราการสูญเสีย (MSE) ที่ใกล้เคียงกันของวิธี JPEG-baseline กับวิธีที่นำเสนอ จะพบว่าประสิทธิภาพในการบีบอัดข้อมูลของวิธีที่นำเสนอนี้ก็สูงกว่า และเมื่อเทียบกับวิธีบีบอัดข้อมูลแบบอาศัยข้อมูลแวดล้อม [5] จะพบว่าค่าอัตราการบีบอัดข้อมูลของวิธีที่นำเสนอนี้มีค่าสูงกว่า แต่มีการสูญเสียของข้อมูลเกิดขึ้น อย่างไรก็ตามเมื่อพิจารณาอัตราการสูญเสียที่เกิดขึ้นนั้นมีค่าน้อยมากซึ่งเกือบจะไม่มีผลกับข้อมูลของภาพผลลัพธ์เลย จึงอาจถือได้ว่าวิธีที่นำเสนอในวิทยานิพนธ์ฉบับนี้เป็นวิธีที่มีประสิทธิภาพดีที่สุด

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

วิทยานิพนธ์ฉบับนี้ได้ทำการศึกษาและวิจัยเกี่ยวกับการบีบอัดข้อมูลด้วยวิธีการบีบอัดแบบมีการสูญเสียค่าชนิดความซับซ้อนต่ำ โดยประยุกต์ใช้การแปลงแบบคิรคิตโคชานน์ และการบีบอัดแบบอาศัยข้อมูลแวดล้อม การวิจัยสามารถสรุปได้เป็น 4 ส่วนคือ

1. การแปลงคิรคิตโคชานน์

การแปลงคิรคิตโคชานน์เพื่อทำให้ภาพที่จะใช้เป็นข้อมูลภาพอินพุทของการบีบอัดแบบอาศัยข้อมูลแวดล้อมมีความต่อเนื่องมากขึ้น ทั้งนี้เพื่อเพิ่มศักยภาพในการบีบอัดข้อมูลให้มากขึ้นด้วย ในส่วนนี้ได้มีการปรับและหาค่าสัมประสิทธิ์เพื่อให้อัตราการสูญเสียข้อมูลของภาพผลลัพธ์มีค่าน้อยที่สุด จากการทดลองได้มีการหาค่าสัมประสิทธิ์ด้วย 10" และได้เลือกใช้ค่า $n = 5$ เพราะทำให้ได้อัตราการสูญเสียของภาพผลลัพธ์ต่ำสุด ด้วยเหตุที่เมื่อทดลองบีบค่า $n > 5$ จะพบว่าอัตราการสูญเสียมีค่าเกือบคงที่

2. การบีบอัดแบบอาศัยข้อมูลแวดล้อม

ในส่วนนี้เป็นกรบีบอัดโดยอาศัยข้อมูลรอบข้างมาพิจารณาเพื่อเข้ารหัสโดยอาศัยทฤษฎีความน่าจะเป็นในการเกิดความผิดพลาด ความสามารถในการบีบอัดข้อมูลขึ้นอยู่กับลักษณะของภาพอินพุท คือ ถ้าภาพมีความต่อเนื่องมากอัตราการบีบอัดที่ได้ก็จะสูง แต่ในทางกลับกันถ้าภาพมีความต่อเนื่องต่ำอัตราการบีบอัดที่ได้ก็มีค่าต่ำด้วย วิธีนี้ได้ใช้การเข้ารหัสแบบ Golomb-Rice ซึ่งไม่มีส่วนของ Header file เพราะเป็นการเข้ารหัสข้อมูลไปที่ละพิกเซลและอาศัยข้อมูลรอบข้างเป็นตัวบ่งบอกขนาดของข้อมูล เช่นเดียวกันในการถอดรหัสก็อาศัยข้อมูลแวดล้อมรอบข้างมาพิจารณาค่าขนาดของข้อมูลอินพุท วิธีจึงจัดได้ว่าเป็นวิธีที่มีอัลกอริธึมในการคำนวณที่ไม่ซับซ้อนมากนัก ปัจจัยที่มีผลต่อความยาวรหัสคือ ค่า k

3. การบีบอัดข้อมูลแบบมีการสูญเสียค่าชนิดความซับซ้อนต่ำ

ในส่วนนี้ได้ใช้คุณสมบัติข้อดีของทั้งการแปลงแบบคิรคิตโคชานน์ และการบีบอัดข้อมูลแบบอาศัยข้อมูลแวดล้อมมาทำงานร่วมกัน เพื่อให้ได้ประสิทธิภาพการบีบอัดข้อมูลที่สูงขึ้นและยังสามารถใช้ได้ดีกับภาพที่มีความต่อเนื่องต่ำได้ ซึ่งวิธีนี้การบีบอัดข้อมูลจะถูกกระทำบนสัมประสิทธิ์ที่ได้จากการแปลงคิรคิตโคชานน์หากข้อมูลที่นำมาบีบอัดมีการกระจายของสเปกตรัมและไม่

ต่อเนื่อง ประสิทธิภาพก็ถูกจำกัด แต่ในทางกลับกัน ข้อมูลที่มีลักษณะไม่ต่อเนื่องแต่มีการซ้ำคาบ เช่น ข้อมูลที่มีลักษณะเป็น texture หรือที่แสดงรายละเอียดของภาพ ซึ่งจะปรากฏเป็นความต่อเนื่องในสเปกตรัมโดเมน จึงเหมาะกับวิธีที่นำเสนอนี้

4. การเปรียบเทียบผลของวิธีที่นำเสนอกับวิธีอื่น

วิธีการบีบอัดข้อมูลแบบมีการสูญเสียค่าชนิดความซับซ้อนต่ำที่นำเสนอในวิทยานิพนธ์นี้ จะให้ค่าอัตราการบีบอัดข้อมูลสูงสุด แต่ค่าความผิดพลาดจะอยู่ระหว่างวิธีการบีบอัดข้อมูลแบบอาศัยข้อมูลแวดล้อม กับวิธี JPEG-Baseline เมื่อพิจารณาจากค่าความผิดพลาดของภาพผลลัพธ์ที่ได้ในวิธีที่นำเสนอนี้ก็ถือว่าอยู่ในช่วงที่มีความสูญเสียน้อยมากซึ่งแทบจะไม่มีผลกับภาพเลย วิธีนี้จึงเป็นวิธีที่ควรรู้หนึ่งในการบีบอัดข้อมูลภาพแบบความสูญเสียค่าและมีความซับซ้อนในการคำนวณต่ำ

จากการทดลองนี้สามารถเป็นแนวทางสำหรับผู้ที่สนใจสามารถนำมาศึกษาค้นคว้าวิจัยเพื่อพัฒนาอัลกอริธึมในการบีบอัดข้อมูลแบบความซับซ้อนต่ำกับข้อมูลภาพต่อไป เช่น นำไปประยุกต์ใช้กับการแปลงเวฟเลท

6.2 ปัญหาที่เกิดขึ้นและข้อเสนอแนะ

ปัญหาที่พบในการทดลอง คือ เนื่องจากในการทดลองทำการบีบอัดข้อมูลนี้ได้ใช้โปรแกรม Matlab ในการเขียนอัลกอริธึมของโปรแกรม ซึ่งมีการทำงานที่ค่อนข้างช้า และในส่วนการแปลงดิสครีตโคซายน์ ถ้าภาพมีขนาดใหญ่จะใช้เวลาในการคำนวณค่อนข้างมาก เพราะ มีการวนลูปรการคำนวณหลายรอบ แต่ถ้ามีการประยุกต์ใช้กับโปรแกรมที่เขียนโดยภาษาอื่น เช่น Delphi , C++ และ Borland C++ เป็นต้น ก็อาจทำให้การคำนวณในโปรแกรมเร็วขึ้น

ในส่วนของการเข้ารหัส Golomb-Rice ปัจจัยที่มีผลต่อความยาวรหัส คือค่า k เช่นกรณีที่ E มีค่าเกินค่า k ที่คำนวณได้มากเกินไป ทำให้ต้องมีการเข้ารหัสแบบจำกัดความยาว ซึ่งมีค่าถึง 32 บิต ถือว่ามีค่ามากเกินไปในแต่ละค่าพิกเซล แต่ถ้ามีการปรับอัลกอริธึมที่คำนวณค่า k ให้เหมาะสมก็อาจจะทำให้ความยาวรหัสมีค่าสั้นลงและส่งผลให้ได้อัตราการบีบอัดข้อมูลสูงขึ้นได้

เอกสารอ้างอิง

- [1] R. C. Gonzales and R. E. Woods, **Digital Image Processing**, Wesley Publishing Company, Sep. 1993.
- [2] R. Crane, **A Simplified Approach to Image Processing**, USA: John Wiley & Sons, Inc, 1997.
- [3] ศิริพงษ์ วงษ์คาร “การลดข้อมูลภาพถ่ายดาวเทียมด้วยการควอนไทซ์แบบเวกเตอร์” วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 2540.
- [4] K. R. Rao and P. Yip, **Discrete Cosine Transform**, San Diego: Academic Press, Inc. 1990.
- [5] M. J. Weinberger, G. Seroussi and G. Sapiro, “LOCO-I: A Low Complexity, Context-based, Lossless Image Compression Algorithm,” Proc. 1996 Data Compression Conference, Utah USA, pp. 140-149, Mar. 1996.
- [6] G. Seroussi and M. J. Weinberger, “Optimal Prefix Codes for Sources with Two-sided Geometric Distributions,” IEEE Trans. Inform. Theory, vol. 46, pp. 121-135, Jan. 2000.
- [7] M. J. Weinberger and G. Seroussi, “The LOCO-I Lossless Image Compression Algorithm : Principles and Standardization into JPEG-LS,” IEEE Trans. image processing, Vol. 9, No. 8, pp. 1309-1324, Aug. 2000.
- [8] S. W. Golomb, “Run-length encodings,” IEEE Trans. Inform. Theory, Vol. IT-12, pp. 399-401, July 1966.
- [9] R. F. Rice, “Some Practical Universal Noiseless Coding Techniques,” Tech. Rep. JPL-79-22, Jet Propulsion Laboratory, Pasadena, CA, Mar. 1979.
- [10] “JPEG-LS with Limited-Length Code Words”, ISO/IEC JTC1/SC29/WG1 Doc.N538, July 1997.

ภาคผนวก ก.

ผลงานวิจัยที่ได้รับการตีพิมพ์

- (1) กอบชัย เศรษฐาญ สายน้ำฝน หอมจันทร์ นภัทร สระเอี่ยม และ อัครพล ตีร์รัตน์, “การบีบอัดข้อมูลของการเข้ารหัสแบบการสูญเสียค่าที่มีความซับซ้อนต่ำ” วิศวกรรมลาดกระบัง ปีที่ 19 ฉบับที่ 1 หน้าที่ 125-130 กันยายน 2544.

ภาคผนวก ข.

โปรแกรม

โปรแกรมหลัก

โปรแกรมประกอบด้วยโปรแกรมหลัก 2 โปรแกรม คือ โปรแกรมการแปลง DCT และ โปรแกรมการบีบอัดข้อมูลแบบอาศัยข้อมูลเวดล้อม

1. โปรแกรมการแปลง DCT (สำหรับภาพ 256*256)

1.1 โปรแกรมการแปลง DCT และการจัดลำดับสัมประสิทธิ์

```
ff=imread('d:\cman.bmp','bmp');
ff=[ff];
f=double(ff);
format long
global gmin;
global gmax;
gmin(1,1:64)=0;
gmax(1,1:64)=0;
ii=0;
[s(1:8,1:8)]=[0];
f=f-(2^7);
r0=0;
c0=0;
for r1=1:8:256
    for c1=1:8:256
        r0=r1+7;
        c0=c1+7;
        a1=[f((r1:r0),(c1:c0))];
[n,m]=size(a1);
%initial variable%
gtemp=0;
```

```

%coefficien setup%
alp(1+0)=sqrt(1/n);
alp((1+1):(n-1+1))=sqrt(2/n);
for u=0:n-1
    for v=0:m-1
        g(u+1,v+1)=0;
    %c(u,v) calculate%
        for x=0:n-1
            for y=0:m-1
                gtemp=a1(x+1,y+1)*cos((2*x+1)*u*pi/(2*n))*cos((2*y+1)*v*pi/(2*m))+ gtemp;
            end
        end
        g(u+1,v+1)=(alp(u+1)*alp(v+1)*gtemp);%round to nearest int
    gtemp=0;
    end
end
g=[g];
g12(r1:r0,c1:c0)=g;
g=0; s=0;
    end
end
g12=[g12];
g12=g12/10;
gg=min(g12);
ii=ii+1;
gmin=min(gg);
ggg=g12+abs(gmin);%is =g+min%
gm=max(ggg);
gmax=max(gm);
for k=1:256
    for l=1:256
        s(k,l)=(255*log10(1+abs(ggg(k,l))))/log10(1+abs(gmax));
    end
end

```

```

        end
    end
    g122=round(s);
    r1=1;
    r0=r1+31;
    for i=1:8
        c1=1;
        c0=c1+31;
        for j=1:8
            [m(1:32,1:32)]=g122(i:8:256,j:8:256);
            if((r1==i) & (c1==j))
                mmm(r1:r0,c1:c0)=[m];
            elseif((r1==i) & (c1~=j))
                r1=r1;r0=r1+31;
                c1=c1+32;c0=c1+31;
                mmm(r1:r0,c1:c0)=[m];
            elseif((r1~=i) & (c1==j))
                mmm(r1:r0,c1:c0)=[m];
            else((r1~=i) & (c1~=j))
                r1=r1;r0=r1+31;
                c1=c0+1;c0=c1+31;
                mmm(r1:r0,c1:c0)=[m];
            end
        end
    end
    r1=r0+1;
    r0=r1+31;
end
g122=[g122];
g1122=uint8(g122);
imwrite(g1122,'d:\cmanlognomap.bmp','bmp')
mmm=[mmm];
mmm1=uint8(mmm);

```

```
imwrite(mmm1,'d:\cmanlogmap.bmp','bmp')
```

1.2 การแปลงกลับ DCT และ การจัดลำดับสัมประสิทธิ์กลับ

```
ww=imread('d:\cmanlogmap.bmp','bmp');
ww=[ww];
w=double(ww);
jj=0;
format long
r1=1;
r0=r1+7;
for i=1:32
    c1=1;
    c0=c1+7;
    for j=1:32
        [rr(1:8,1:8)]=w(i:32:256,j:32:256);
        if((r1==i) & (c1==j))
            rrr(r1:r0,c1:c0)=[rr];
        elseif((r1==i) & (c1~=j))
            r1=r1;r0=r1+7;
            c1=c1+8;c0=c1+7;
            rrr(r1:r0,c1:c0)=[rr];
        elseif((r1~=i) & (c1==j))
            rrr(r1:r0,c1:c0)=[rr];
        else((r1~=i) & (c1~=j))
            r1=r1;r0=r1+7;
            c1=c0+1;c0=c1+7;
            rrr(r1:r0,c1:c0)=[rr];
        end
    end
end
r1=r0+1;
r0=r1+7;
end
```

```

rrr=[rrr];
rrr=double(rrr);
for k=1:256
    for l=1:256
        gr(k,l)=(10^((rrr(k,l)*log10(1+abs(gmax)))/255))-1;
    end
end
grr=gr+gmin;
grr=grr*10;
r0=0;
c0=0;
for r1=1:8:256
    for c1=1:8:256
        r0=r1+7;
        c0=c1+7;
        g13=[grr((r1:r0),(c1:c0))];
[n,m]=size(g13);
%initial variable%
invdcttemp=0;
%coefficien setup%
alp(1+0)=sqrt(1/n);
alp((1+1):(n-1+1))=sqrt(2/n);
for x=0:n-1
    for y=0:m-1
        invdct(x+1,y+1)=0;
        %c(u,v) calculate%
        for u=0:n-1
            for v=0:m-1
                invdcttemp=alp(u+1)*alp(v+1)*g13(u+1,v+1)*cos((2*x+1)*u*pi/(2*n))*cos((2*y+1)*v*pi/
(2*m))+ invdcttemp;
            end
        end
    end
end
end

```

```

invdct(x+1,y+1)=(invdcttemp);
invdcttemp=0;
    end
end
invdct=invdct+2^7;
invdct=[invdct];
invdct122(r1:r0,c1:c0)=invdct;
invdct=0;
end
end
invdct133=uint8(round(inv dct122));
imwrite(inv dct133,'d:\cmanlogremapdec.bmp','bmp')

```

2. โปรแกรมการบีบอัดข้อมูลแบบอาศัยข้อมูลแวดล้อม

2.1 โปรแกรมหลัก

```

%start%
clc;
clear;
global bitcnt;
bitcnt=0;
% open input image file and output binary file%
img=imread('d:\cmanlogmap.bmp'); %gray image 8bit only%
fname='test.bin';
ftest=fopen(fname,'wb+'); %fid = ftest;%
foutputName='output.bin';
[n,m]=size(img);
img=double(img);
global flog;
flog = fopen('log.txt','wr+'); %fid = flog;
initial;
while ~((CurrPos.x==m)&(CurrPos.y==n))
    [x,a,b,c,d,Ix,Ra,Rb,Rc,Rd,EOL,CurrPos]=GetNextSample(CurrPos,img);

```

```

LocalGradientComputation;
fprintf(flog,'Current Pos. x = %d, y = %d\n',CurrPos.x,CurrPos.y);
fprintf(flog,'Rc= %d, Rb= %d, Rd= %d, Ra= %d, Ix= %d\n',Rc,Rb,Rd,Ra,Ix);
% mode selection procedure for lossless coding%
if (D1==0)&(D2 == 0)&(D3 == 0)
RunModeProcessing;
else
fprintf(flog,'***** start Reg mode *****\n');
RegularModeProcessing;
fprintf(flog,'***** end Reg mode *****\n');
end
end
Report

```

2.2 โปรแกรมกำหนดค่าเริ่มต้น (initial)

```

%%A.2 Initialisations and conventions P.17%
T1=1;
T2=7;
T3=21;
MAXVAL = 255;
MAX_C=127;
MIN_C=-128;
NEAR=0;
RANGE = floor(((MAXVAL +(2*NEAR))/((2*NEAR)+1)))+1;
qbpp = ceil(log2(RANGE));
bpp = max(2, ceil(log2(MAXVAL +1)));
LIMIT = 2*(bpp+max(8,bpp));
RESET = 64;
A(1,1:367)= max(2,floor(((RANGE +(2^5))/(2^6))));
N = ones(1,367); %N(0..366) = 1;
B = zeros(1,365); %B(0..364) = 0;
C = zeros(1,365); %C(0..364) = 0;

```

```

RUNindex = 1;
J(1,1:32)=[0,0,0,0,1,1,1,1,1,2,2,2,2,3,3,3,3,4,4,5,5,6,6,7,7,8,9,10,11,12,13,14,15];
Nn(1,366:367)=[0,0];
%start position of img scan%
CurrPos.x=0;
CurrPos.y=1;
EOL=0;

```

2.3 ฟังก์ชันโปรแกรม GetNextSample

```

function [x,a,b,c,d,Ix,Ra,Rb,Rc,Rd,EOL,NextPos]=GetNextSample(CurrPos,imgin);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Read the next sample in the input image and sets the corresponding values of
[x,a,b,c,d,Ix,Ra,Rb,Rc,Rd,EOL,NextPos]. EOL set to be 1, if the sameple read is at the end
of the current image line. Input = CurrPos : Current Position CurrPos.x and CurrPos.y
and imgin : input image. Lossless x,a,b,c,d = Ix,Ra,Rb,Rc,Rd respectively. and Ix=Rx also.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[n,m]=size(imgin); % n = row, m=column%
%assign next position%
if CurrPos.x==m %Current image EOL%
    if CurrPos.y==n
return;
%exit;end of image%
    end
    NextPos.x=1;
    NextPos.y=CurrPos.y+1;
else
    NextPos.x=CurrPos.x+1;
    NextPos.y=CurrPos.y;
end
if NextPos.y==1 % frist line%
    c=0;
    b=0;

```

```

    d=0;
if NextPos.x==1 %frist point of image%
    a=b;
    EOL=0; Ra=a; Rb=b;Rc=c;Rd=d;Ix=imgin(NextPos.y,NextPos.x);x=Ix;
Return;
elseif NextPos.x==m %EOL of frist line5
    EOL=1;
    d=b;
    a=imgin(NextPos.y,NextPos.x-1);
    Ra=a; Rb=b;Rc=c;Rd=d;Ix=imgin(NextPos.y,NextPos.x);x=Ix;
Return;
else
    a=imgin(NextPos.y,NextPos.x-1);
    EOL=0; Ra=a; Rb=b;Rc=c;Rd=d;Ix=imgin(NextPos.y,NextPos.x);x=Ix;
Return;
end
else
    if NextPos.x==1 % frist point not frit line%
        b=imgin(NextPos.y-1,NextPos.x);
        d=imgin(NextPos.y-1,NextPos.x+1);
        a=b;
    if NextPos.y==2
        c=0;
    else
        c=imgin(NextPos.y-2,NextPos.x);
    end
    EOL=0; Ra=a; Rb=b;Rc=c;Rd=d;Ix=imgin(NextPos.y,NextPos.x);x=Ix;
Return;
elseif NextPos.x==m %EOL of not frist line%
    EOL=1;
    b=imgin(NextPos.y-1,NextPos.x);
    d=b;

```

```

a=imgin(NextPos.y,NextPos.x-1);
c=imgin(NextPos.y-1,NextPos.x-1);
Ra=a; Rb=b;Rc=c;Rd=d;Ix=imgin(NextPos.y,NextPos.x);x=Ix;
Return;
else
a=imgin(NextPos.y,NextPos.x-1);
b=imgin(NextPos.y-1,NextPos.x);
c=imgin(NextPos.y-1,NextPos.x-1);
d=imgin(NextPos.y-1,NextPos.x+1);
EOL=0; Ra=a; Rb=b;Rc=c;Rd=d;Ix=imgin(NextPos.y,NextPos.x);x=Ix;
Return;
end
end

```

2.3 โปรแกรมย่อย LocalGradientComputation

%A3.1 Local gradient computation p.19-----step 2.%

```

D1= double(Rd)-double(Rb);
D2= double(Rb)-double(Rc);
D3= double(Rc)-double(Ra);

```

2.4 โปรแกรมย่อย ในโหมดรัน(RunModeProcessing)

2.4.1 โปรแกรมย่อย RunScan

%A.7.1.1 Run Scanning P.25----- step 17 a%.

```

RUNval = Ra;
RUNcnt = 0;
while (abs(double(Ix) - RUNval) <= NEAR)
    RUNcnt = RUNcnt +1;
    Rx = RUNval;
    if (EOL == 1)
        break;
    else
        [x,a,b,c,d,Ix,Ry,Rb,Rc,Rd,EOL,CurrPos]=GetNextSample(CurrPos,img);
    end
end

```

```

end
end
RUNcnt0=RUNcnt;

```

2.4.2 โปรแกรมย่อย RunLengthCoding

```

%A.7.1.2 Run-length coding P.25%
%Fig: A.15 Encoding of run segments of length rm. --- step 17 b.%
fprintf(flog,'*****start runmode *****\n'); %%%%%%%%%%%%%%%
RUNindex0=RUNindex; %for RunInterrupt
while (RUNcnt >= (bitshift(1,J(RUNindex))))
    AppendToBitStream(1,1,ftest);
    RUNcnt = RUNcnt - (bitshift(1,J(RUNindex)));
    if RUNindex < 32
        RUNindex = RUNindex + 1;
    end
end
end
%Fig: A.16 Encoding of run segments of length less than rm.step 17 c.%
if (EOL ==0)
    AppendToBitStream(0,1,ftest);
    AppendToBitStream(RUNcnt,J(RUNindex),ftest);

    if (RUNindex > 1)
        RUNindex = RUNindex -1;
    end
else
    if (RUNcnt >0)
        AppendToBitStream(1,1,ftest);
    end
end
if ((CurrPos.x==m)&(CurrPos.y==n))
    fprintf(flog,'\n**** end of img by run mode *****\n');
Report;

```

end

2.4.3 ฟังก์ชันโปรแกรม AppendToBitStream

```
function y =AppendToBitStream(a,b,fid)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

appends the non-negative number a in binary form to the encoded bit stream,using b bits. Most significant bits are appended first.

```
[y,errmes]=AppendTobitStream(a,b);
```

```
y errcode 0 = fail , 1 = done
```

```
errmes=error message.
```

```
fid = file id.
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
global bitcnt;
```

```
global flog;
```

```
if a<0
```

```
    disp('AppendTobitStrem(a,b) but a<0');
```

```
    y=0;
```

```
    return;
```

```
end
```

```
if b==0
```

```
    return;
```

```
end
```

```
a_binary=dec2base(a,2);
```

```
fprintf(flog,'%c',a_binary);
```

```
bit0=bitand(0,bitcmp(0,8));
```

```
bit1=bitcmp(bit0,8);
```

```
for i=1:b
```

```
    temp=uint8(str2num(a_binary(i)));
```

```
    if temp==0
```

```
        fwrite(fid,bit0,'ubit1');
```

```
    else
```

```
        fwrite(fid,bit1,'ubit1');
```

```

end
    bitcnt=bitcnt+1;
end
y=1;

```

2.4.4 โปรแกรมย่อย RunInterrupt

%A.7.2 Run interruption sample encoding P.26 -----step 18.%

```
if (abs(Ra - Rb) <= NEAR)
```

```
    RItyp = 1;
```

```
else
```

```
    RItyp = 0;
```

```
end
```

%Fig: A.17 Index computation%

```
if (RItyp == 1)
```

```
    Px = Ra;
```

```
else
```

```
    Px = Rb;
```

```
end
```

```
Errval = double(Ix) - double(Rb);
```

%Fig: A.18 Prediction error for a run interruption sample%

```
if ((RItyp == 0) & (Ra > Rb))
```

```
    Errval = -Errval;
```

```
    SIGN = -1;
```

```
else
```

```
    SIGN = 1;
```

```
end
```

```
if (NEAR > 0)
```

```
    Errval = Errval ;           %Errval = Quantize(Errval);%
```

```
    Rx = Ix;                    %Rx = ComputeRx();%
```

```
else
```

```
    Rx = Ix;
```

```
end
```

```
Errval = ModRange(Errval,RANGE);
```

```
%Fig: A.19 Error computation for a run interruption sample.%
```

```
if (RIttype ==0)
```

```
    TEMP = A(366);
```

```
else
```

```
    TEMP = A(367)+(bitshift(N(367),-1));
```

```
end
```

```
%Fig A.20 Computation of the auxiliary variable%
```

```
Q=RIttype+366;
```

```
k=0;
```

```
while (bitshift(N(Q),k) < A(Q))
```

```
    k = k+1;
```

```
end
```

```
% Set Q=Ritype +365  %
```

```
if (( k==0) & (Errval >0) & (2*Nn(Q) < N(Q)))
```

```
    map =1;
```

```
elseif ((Errval < 0) & (2* Nn(Q) >= N(Q)))
```

```
    map =1;
```

```
elseif ((Errval <0) & (k ~= 0))
```

```
    map =1;
```

```
else
```

```
    map=0;
```

```
end
```

```
%Fig A.21 Computation of map far Errval mapping%
```

```
if Errval==0
```

```
    EMErrval=0;
```

```
else
```

```
    EMErrval = 2*abs(Errval)-RIttype-map;
```

```
End
```

%Fig: A.22 Errval mapping for run interruption sample%

%%%

Encode EMErrval following the same procedures as in the regular mode (see A.5.3), but using the limited length Golomb code function $LG(k, \text{glimit})$, where $\text{glimit} = \text{LIMIT} - J[\text{RUNindex}] - 1$ and RUNindex corresponds to the value of the variable before the update specified in A.7.1.2.

%%%

HEMErrval = bitshift(EMErrval, -k); %High 8-k significant bits%

glimit=LIMIT-J(RUNindex0)-1;

if HEMErrval < (glimit - qbpp - 1)

for i = 1:HEMErrval

AppendToBitStream(0,1,ftest);

end

AppendToBitStream(1,1,ftest);

for i = 0:(k-1)

AppendToBitStream(bitget(EMErrval,k-i),1,ftest);

end

else

for i=1:(glimit-qbpp-1)

(AppendToBitStream(0,1,ftest));

end

AppendToBitStream(1,1,ftest);

AppendToBitStream(0,qbpp-length(dec2bin(EMErrval-1)),ftest);

AppendToBitStream((EMErrval-1),length(dec2bin(EMErrval-1)),ftest);

end

fprintf(flog, '\ncoding position x: %d, y: %d\n', CurrPos.x, CurrPos.y);

fprintf(flog, 'RUNval: %d, RUNcnt: %d, RUNindex: %d, Rltype: %d, Errval: %d, TEMP:

%d\n', RUNval, RUNcnt0, RUNindex0-1, Rltype, Errval, TEMP);%

fprintf(flog, 'k: %d, map: %d, EMErrval: %d, Q: %d, A[Q]: %d, N[Q]: %d, Nn[Q]:

%d\n', k, map, EMErrval, Q-1, A(Q), N(Q), Nn(Q)); %

% update variable for run interruption sample%

if (Errval<0)

```

        Nn(Q)= Nn(Q)+1;
end
        A(Q) = A(Q)+(bitshift((EMErrval + 1 - Rltype),-1));
if (N(Q) == RESET)
        A(Q) = bitshift(A(Q),1);
        N(Q) = bitshift(N(Q) ,1);
        Nn(Q) = bitshift(Nn(Q),1);
end
        N(Q) = N(Q)+1;
%Fig A.23 : Update of variables for run interruption sample.%
fprintf(flog,'After update RUNindex: %d, A[Q]: %d, N[Q]: %d, Nn[Q]: %d\n',RUNindex-1,A
(Q),N(Q),Nn(Q)); %%%
fprintf(flog,'\n***** end runmode *****\n'); %%%

```

2.4.5 ฟังก์ชันโปรแกรม Modrange

```

function Errval_out=ModRange(Errval,RANGE);
%A.4.5 Modulo reduction of the prediction error P.22 -----step 10%
if (Errval <0)
        Errval_out = Errval +RANGE;
elseif (Errval >= ((RANGE+1)/2))
        Errval_out = Errval - RANGE;
else
        Errval_out=Errval;
End

```

2.5 โปรแกรมในโหมดปกติ(RegularModeProcessing)

2.5.1 โปรแกรมย่อย LocalGradientQuantization

```

% A.3.3 Local gradient quantization p.20-----step 4.%
if (D1<=-T3) Q1 = -4;
elseif (D1<=-T2) Q1 = -3;
elseif (D1<=-T1) Q1 = -2;
elseif (D1<-NEAR) Q1 = -1;

```

```

elseif (D1<=NEAR) Q1 = 0;
elseif (D1<T1) Q1 = 1;
elseif (D1<T2) Q1 = 2;
elseif (D1<T3) Q1 = 3;
else Q1=4;
end

```

```

if (D2<=-T3) Q2 = -4;
elseif (D2<=-T2) Q2 = -3;
elseif (D2<=-T1) Q2 = -2;
elseif (D2<-NEAR) Q2 = -1;
elseif (D2<=NEAR) Q2 = 0;
elseif (D2<T1) Q2 = 1;
elseif (D2<T2) Q2 = 2;
elseif (D2<T3) Q2 = 3;
else Q2=4;
end

```

```

if (D3<=-T3) Q3 = -4;
elseif (D3<=-T2) Q3 = -3;
elseif (D3<=-T1) Q3 = -2;
elseif (D3<-NEAR) Q3 = -1;
elseif (D3<=NEAR) Q3 = 0;
elseif (D3<T1) Q3 = 1;
elseif (D3<T2) Q3 = 2;
elseif (D3<T3) Q3 = 3;
else Q3=4;
end

```

2.5.2 โปรแกรมย่อย LocalGradientMerging

%A.3.4 Quantized gradient merging p.20 -----step 5.%

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
If the first non-zero element of the vector (Q1, Q2, Q3) is negative, then all the signs of the vector
(Q1, Q2, Q3) shall be reversed to obtain (-Q1, -Q2, -Q3) .In this case, the variable SIGN shall be
set to -1, otherwise it shall be set to +1.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (Q1<0) | ((Q1==0)&(Q2<0)) | ((Q1==0)&(Q2==0)&(Q3<0))
    SIGN = -1;
else
    SIGN =1;
end
if (Q1<0)|((Q1==0) & (Q2<0))|((Q1==0) & (Q2==0) & (Q3<0))
    Q1=-Q1;
    Q2=-Q2;
    Q3=-Q3;
end
if Q1==0 & Q2==0 & Q3==0
    Q=0;
elseif Q1>0
    Q = (Q1-1)*(9^2)+(Q2+4)*(9^1)+(Q3+4)*(9^0)+1;
elseif (Q1==0) & (Q2>0)
    Q = (Q2-1)*(9^1)+(Q3+4)*(9^0)+325;
elseif (Q1==0) & (Q2==0) & (Q3>0)
    Q = (Q3-1)*(4^0) + 361;
End

```

2.5.3 โปรแกรมย่อย Prediction

```

%% A.4 Prediction%
%A.4.1 Edge-detection predictor p.21-----step 6.%
if (Rc>= max(Ra, Rb))
    Px= min(Ra, Rb);
else
    if (Rc<= min(Ra, Rb))

```

```

        Px= max(Ra, Rb);
    else
        Px = Ra + Rb - Rc;
    end
end
%A.4.2 Prediction Correction p.21-----step 7.%
if (SIGN == +1)
    Px= Px+C(Q);
else
    Px=Px-C(Q);
end
if (Px> MAXVAL)
    Px= MAXVAL;
elseif (Px<0)
    Px=0;
end
%A.4.3 Computation of prediction error P.21-----step 8.%
Errval= double(Ix)- double(Px);
if (SIGN == -1)
    Errval=-Errval;
end
Rx=Ix;
%A.4.5 Modulo reduction of the prediction error P.22 -----step 10%
if (Errval <0)
    Errval = Errval +RANGE;
end
if (Errval >= ((RANGE+1)/2))
    Errval = Errval - RANGE;
End

```

2.5.4 โปรแกรมย่อย PredictionErrorCoding

```
%%A.5 Prediction error encoding P.22%
```

%A.5.1 Golobm coding variable computation P.23-----step 11.%

%for (k=0; (N(Q)<<k)<A(Q); k++)%

k=0;

while (bitshift(N(Q),k) < A(Q))

 k = k+1;

end

%A.5.2 Error mapping P.23----- step 12%

if ((NEAR == 0) & (k == 0) & (2*B(Q) <= -N(Q)))

 if (Errval >= 0)

 MErrval = 2* Errval + 1;

 else

 MErrval = -2*(Errval + 1);

 end

else

 if (Errval >= 0)

 MErrval = 2* Errval;

 else

 MErrval = -2* Errval - 1;

 end

end

%A.5.3 Mapped-error encoding P.23 -----step 13%

%%%

The mapped error value, MErrval, shall be encoded with the limited length Golomb %code

function LG(k, LIMIT) defined by the following procedure.

%%%

HMErrval = bitshift(MErrval, -k); %High 8-k significant bits%

if HMErrval < (LIMIT - qbpp - 1)

 for i = 1:HMErrval

 AppendToBitStream(0,1,ftest);

 end

 AppendToBitStream(1,1,ftest);

 for i = 0:(k-1)

```

        AppendToBitStream(bitget(MErrval,k-i),1,ftest);
    end
else
    for i=1:(LIMIT-qbpp-1)
        AppendToBitStream(0,1,ftest);
    end
    AppendToBitStream(1,1,ftest);
    for i=1:qbpp-length(dec2bin(MErrval-1))
        AppendTobitStream(0,1,ftest);
    end
    AppendToBitStream((MErrval-1),length(dec2bin(MErrval-1)),ftest);
end
fprintf(flog,'\n Q1 = %d, Q2 = %d, Q3 = %d ',Q1,Q2,Q3); %%%%%%%%%%%
fprintf(flog,'\n Px: %d, SIGN: %d, Errval(mod): %d, k: %d, MErrval: %d, A[Q]: %d,B[Q]: %d, C
[Q]: %d, N[Q]: %d\n',Px,SIGN,Errval,k,MErrval,A(Q),B(Q),C(Q),N(Q));%%%%%%%%%%

```

2.5.5 โปรแกรมย่อย UpdateVariable

```

%%A.6 Update variables P.23 %
%A.6.1 Update P.24-----step 14 %
B(Q) = B(Q) + Errval*(2*NEAR +1);
A(Q) = A(Q) + abs(Errval);
if (N(Q) == RESET)
    A(Q) = bitshift(A(Q),-1);
    B(Q) = bitshift(B(Q),-1);
    N(Q) = bitshift(N(Q),-1);
end
N(Q) = N(Q)+1;
% A.6.2 Bias computation P.24-----step 15 %
if (B(Q) <= -N(Q))
    B(Q) = B(Q) +N(Q);
    if (C(Q) > MIN_C)
        C(Q) = C(Q)-1;
    end
end

```

```

end
if (B(Q) <= -N(Q))
    B(Q) = -N(Q) + 1;
end
elseif (B(Q)>0)
    B(Q) = B(Q) - N(Q);
    if (C(Q) < MAX_C)
        C(Q) = C(Q) + 1;
    end
    if (B(Q) > 0)
        B(Q) = 0;
    end
end
end
fprintf(flog,'after update A[Q]: %d, B[Q]: %d, C[Q]: %d, N[Q]: %d\n',A(Q),B(Q),C(Q),N(Q));
fprintf(flog,'***** end Reg. mode *****');

```

2.6 โปรแกรมย่อย Report

```

% end of main program%
fprintf('number of encoded bit = %d\n',bitcnt)
fprintf(flog,'number of encoded bit = %d\n',bitcnt);
for i=1:(8-rem(bitcnt,8))
    AppendToBitStream(0,1,ftest);
end
fclose(flog);
fclose(ftest);
report=dir('test.bin') %output file
BitSwap(fname,foutputName);
quit cancel;

```

2.6.1 ฟังก์ชันโปรแกรม BitSwap

```

function foutput=BitSwap(finputName,foutputName)
%function BitSwap(finputName,foutputName)%

```

```
%rotate 8 bit data%
%finputName='test.bin';%
finput=fopen(finputName,'r');

%foutputName='output.bin';%
foutput=fopen(foutputName,'wb+');

while ~feof(finput)
    temp=fread(finput,1,'uint8');
    for i=0:7
        fwrite(foutput,bitget(temp,8-i),'ubit1');
    end
end
fclose(finput);
fclose(foutput);
return
```

ประวัติผู้เขียน

นางสาวสายน้ำฝน หอมจันทร์ เกิดเมื่อวันที่ 25 เมษายน 2521 ที่จังหวัดฉะเชิงเทรา สำเร็จการศึกษาวិทยาศาสตร์บัณฑิตสาขาฟิสิกส์อุตสาหกรรม จากภาควิชาฟิสิกส์อุตสาหกรรมและอุปกรณ์การแพทย์ คณะวิทยาศาสตร์ประยุกต์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ ปีการศึกษา 2541