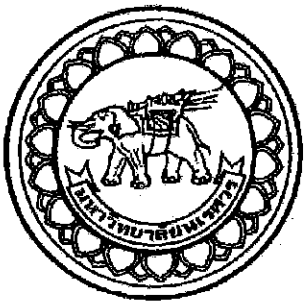
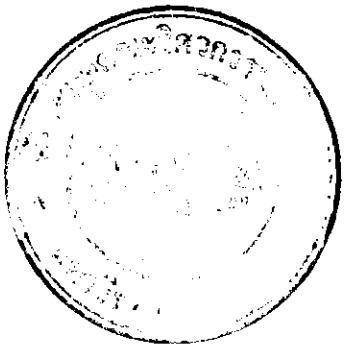


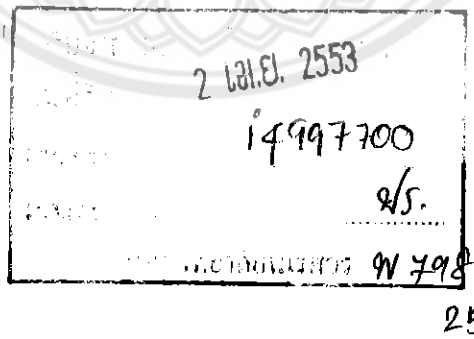


การบีบอัดข้อมูลด้วยวิธีคาลิค

DATA COMPRESSION BY USING CALIC

นายพีระพงษ์ อุดมตระกูลวงศ์ รหัส 48364869

นายเฉลิมเกียรติ รอดเกตุ รหัส 48365057

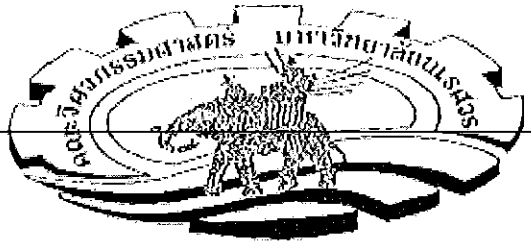


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2551



ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน การบีบอัดข้อมูลด้วยวิธีคาลิค

ผู้ดำเนินโครงงาน นายพีระพงษ์ อุดมตระกูลวงศ์ รหัส 48364869

นายเฉลิมเกียรติ รอดเกตุ รหัส 48365057

อาจารย์ที่ปรึกษา ดร.พนมขวัญ ริยะมงคล

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2551

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ดร.พนมขวัญ ริยะมงคล)

.....กรรมการ

(นางสาวศิริพร เดชะศิลารักษ์)

.....กรรมการ

(นายเศรษฐา ตั้งค้าวานิช)

หัวข้อโครงงาน	การบีบอัดข้อมูลภาพด้วยวิธีคาลิค
ผู้ดำเนินโครงงาน	นายพีระพงษ์ อุดมตระกูลวงศ์ รหัส 48364869
	นายเฉลิมเกียรติ รอดเกตุ รหัส 48365057
อาจารย์ที่ปรึกษา	ดร.พนมขวัญ ริยะมงคล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2551

บทคัดย่อ

โครงงานนี้ได้พัฒนาโปรแกรมบีบอัดข้อมูลภาพด้วยวิธีคาลิค ซึ่งอาศัยหลักการและทฤษฎีการบีบอัดข้อมูลภาพแบบคอนเทกซ์เบส โดยโปรแกรมนี้ได้นำเอาข้อมูลภาพนำมาผ่านกระบวนการบีบอัดข้อมูลภาพด้วยวิธีคาลิค ซึ่งเมื่อนำรูปภาพมาผ่านกระบวนการบีบอัดด้วยวิธีคาลิคโปรแกรมจะทำการบีบอัดภาพจนมีขนาดเล็กลงแต่ไม่มีการสูญหายของข้อมูล นอกจากนี้ได้ทำการเปรียบเทียบการบีบอัดข้อมูล กับ LJPEGและGZIP เพื่อแสดงว่าผลของการบีบอัดข้อมูลแบบคาลิค มีอัตราส่วนการบีบอัด ดีที่สุด

Project Title	DATA COMPRESSION BY USING CALIC
Name	Mr. Perapong Udomtragulwong ID. 48364869
	Mr.Chalermkiat Rodkate ID. 48365057
Project Advisor	Panomkhawn Riyamongkol, Ph.D
Major	Computer Engineering.
Department	Electrical and Computer Engineering.
Academic Year	2008

ABSTRACT

This project has developed the data-compression program with calic strategy. This strategy is generally based on context based concept the program has taken the picture data to go through the process of calic data-compression the encoded picture files have smaller sizes than the original picture files. Moreover, the tested picture files have been compress with LJPEG and GZIP to compare the output data size. The result show that calic strategy give the best compression ratio.

กิตติกรรมประกาศ

โครงงานนี้ประสบความสำเร็จได้ด้วยดีด้วยความกรุณาจากอาจารย์ที่ปรึกษาโครงงานและคณะกรรมการทุกท่าน ขอขอบพระคุณ ดร.พนมขวัญ รียะมงคล ซึ่งเป็นที่ปรึกษาโครงงานนี้ที่คอยให้คำแนะนำและคำปรึกษามาโดยตลอด อีกทั้ง อาจารย์เศรษฐา ตั้งค้าวานิช และ อาจารย์ศิริพร เดชะศิลารักษ์ ที่ให้คำแนะนำในการปรับปรุงโครงงานให้ดียิ่งขึ้น จนทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

ในโอกาสนี้ทางคณะผู้จัดทำโครงงานจึงขอขอบพระคุณทุกๆ ท่านที่มีส่วนร่วมในการทำโครงงานนี้ ตลอดจนผู้คิดค้นทฤษฎีต่างๆ ที่โครงงานฉบับนี้ได้นำความรู้ที่ได้มาพัฒนาระบบจนประสบความสำเร็จ คณะผู้จัดทำขอบกราบขอบพระคุณไว้ ณ ที่นี้

นายพีระพงษ์ อุดมตระกูลวงศ์

นายเฉลิมเกียรติ รอดเกตุ

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ฉ

สารบัญรูป ..... ช

บทที่ 1 บทนำ

1.1ที่มาและความสำคัญของโครงงาน 1

1.2วัตถุประสงค์ของโครงงาน 1

1.3ขอบข่ายของโครงงาน 1

1.4ขั้นตอนการดำเนินโครงงาน 1

1.5แผนการดำเนินโครงการ 2

1.6ผลที่คาดว่าจะได้รับ 2

1.7งบประมาณที่ใช้ 3

บทที่ 2 หลักการและทฤษฎี

2.1 หลักการบีบอัดข้อมูล 4

2.2 การเข้ารหัส 5

2.3 โครงสร้างของคาลิค 7

2.4 การบีบอัดข้อมูลแบบอื่นนอกจากคาลิค 13

บทที่ 3 วิธีการดำเนินการ

3.1 ศึกษาอัลกอริทึมและขั้นตอนการออกแบบอัลกอริทึมของคาลิค 17

3.2 โปรแกรมการเข้ารหัส 18

3.3 โปรแกรมการถอดรหัส 19

สารบัญ(ต่อ)

หน้า

บทที่ 4 ผลการทดลอง

4.1 วัตถุประสงค์การทดลอง 23

4.2 การเรียกใช้โปรแกรม 23

4.3 การทดลองบีบอัดแบบคาลิด 23

4.4 ผลการทดลอง 23

4.5 สรุปผลการทดลอง 33

บทที่ 5 บทสรุป

5.1 สรุปผลการทดลอง 38

5.2 ปัญหาในการทดลองและแนวทางแก้ไข 38

5.3 แนวทางการพัฒนาในอนาคต 38

เอกสารอ้างอิง 39

ประวัติผู้เขียนโครงงาน 40

สารบัญตาราง

ตารางที่ หน้า

1.1 แผนการดำเนินงาน 2
2.1 ขั้นตอนทูไลน์บัฟเฟอ 8
2.2 คอนเทกซ์ฟอเมชั่น 9
2.3 ค่าของ $\hat{I}$ 10
2.4 ควอนไทล์เซชั่น 11
2.5 การเทอนารี 12
4.1 ผลการทดลองที่ 1 23
4.2 ผลการทดลองที่ 2 24
4.3 ผลการทดลองที่ 3 25
4.4 ผลการทดลองที่ 4 27
4.5 ผลสรุปของรูป555x366 34
4.6 ผลสรุปของรูป270x103 35
4.7 ผลสรุปของรูป130x92 36
4.8 ผลสรุปของรูป20x20 36

สารบัญรูป

รูปที่	หน้า
2.1 รูปที่ 1 โครงสร้างที่เป็นองค์ประกอบของคาลิค	8
2.3 พิกเซลใกล้เคียง	13
3.1 ขั้นตอนการดำเนินงาน	16
3.2 แผนภาพโปรแกรมการบีบอัดข้อมูลแบบคาลิค	17
3.3 แผนภาพโปรแกรมการเข้ารหัส	18
3.4 แผนภาพโปรแกรมการถอดรหัส	19
3.5 โฟลชาร์ตการเข้ารหัส	20
3.6 โฟลชาร์ตการถอดรหัส	21
4.1 ไฟล์รูปขนาด 20x20	23
4.2 ไฟล์รูปขนาด 130x92	24
4.3ไฟล์รูปขนาด 270x103	25
4.4 ไฟล์รูปขนาด 555x366	26
4.5 หน้าตาของโปรแกรมการบีบอัดข้อมูลด้วยวิธีคาลิค	28
4.6 ผลการรันโปรแกรมบีบอัดข้อมูลด้วยวิธีคาลิค	28
4.7 รูปที่ได้หลังการ Decode	29
4.8 ฮิสโทแกรมของรูปขนาด 20x20	30
4.9 ฮิสโทแกรมของรูปขนาด 130x92	31
4.10 หน้าตาโปรแกรมบีบอัดแบบGzip	32
4.11 หน้าตาโปรแกรมบีบอัดแบบ LJPEG	32
4.12 รูปแสดงผลหลังการบีบอัดของทุกการบีบอัด	33
4.13 กราฟแสดงค่าCRและเวลาของรูปขนาด555x366	33
4.14 กราฟแสดงค่าCRและเวลาของรูปขนาด270x103	34
4.15 กราฟแสดงค่าCRและเวลาของรูปขนาด130x92	35
4.16 กราฟแสดงค่าCRและเวลาของรูปขนาด20x20	36

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เนื่องจากในปัจจุบันการจัดเก็บข้อมูลต่างๆเป็นสิ่งที่สำคัญมาก แต่ปัจจุบันข้อมูลมีจำนวนมากและมีขนาดใหญ่จึงทำให้มีการเปลืองเนื้อที่ในการเก็บข้อมูล ดังนั้นจึงมีการบีบอัดข้อมูลซึ่งการบีบอัดข้อมูลนั้นมีจุดประสงค์หลักเพื่อให้ขนาดของข้อมูลที่เก็บมีขนาดลดลง การบีบอัดข้อมูลมีประโยชน์ในการลดปริมาณการใช้ทรัพยากร เช่น ประหยัดพื้นที่ของฮาร์ดดิสก์เมื่อเก็บข้อมูล หรือใช้แบนด์วิดธ์ของระบบเครือข่ายน้อยลงเพื่อส่งข้อมูลที่บีบอัดแล้ว โครงงานนี้จึงศึกษาวิธีการบีบอัดข้อมูลภาพชนิดหนึ่งซึ่งเรียกว่า คาลิค การบีบอัดข้อมูลด้วยวิธีการนี้จะไม่มีการสูญเสียข้อมูล และมีความสามารถในการบีบอัดข้อมูลได้ดีเมื่อเทียบกับวิธีการอื่น

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาวิธีการบีบอัดข้อมูลภาพโดยใช้ คาลิค

1.2.2 เพื่อให้ได้โปรแกรมสำหรับบีบอัดข้อมูลภาพด้วยวิธีคาลิค

1.3 ขอบข่ายของโครงงาน

1.3.1 เขียนโปรแกรมเพื่อบีบอัดข้อมูลโดยใช้ คาลิค

1.3.2 เปรียบเทียบกับวิธีบีบอัดข้อมูลชนิด LJPEG และ GZIP

1.4 ขั้นตอนของโครงงาน

1.4.1 ศึกษาวิธีการบีบอัดข้อมูล

1.4.2 เขียนโปรแกรมสำหรับบีบอัดข้อมูลด้วยวิธี คาลิค

1.4.3 ทดสอบโปรแกรมและเปรียบเทียบผลลัพธ์กับวิธีการบีบอัดข้อมูล

1.4.4 สรุปผลการทดลองและทำรายงาน

1.5 แผนการดำเนินงาน

ตารางที่1.1 แผนการดำเนินงาน

ลำดับ	การดำเนินการ	ปี 2551							ปี 2552			
		มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
1	ศึกษาวิธีการบีบอัดข้อมูล	←	—	—	→							
2	เขียนโปรแกรมสำหรับการบีบอัดข้อมูลด้วยวิธี คาลิค		←	—	—	—	→					
3	ทดสอบโปรแกรมและเปรียบเทียบผลลัพธ์กับวิธีการบีบอัดข้อมูลชนิดอื่น				←	—	—	→				
4	สรุปผลการดำเนินโครงงาน					←	—	—	→			
5	จัดพิมพ์โครงงาน							←	—	—	—	→

1.6 ผลที่คาดว่าจะได้รับ

1.สามารถนำความรู้ที่ได้ศึกษามาใช้ในการบีบอัดข้อมูลเพื่อใช้งานจริง

2.สามารถเข้าใจหลักการบีบอัดข้อมูลได้

3. เป็นแหล่งข้อมูลและความรู้ที่เป็นประโยชน์แก่สาธารณชน

1.7 งบประมาณของโครงการ

1. ค่าถ่ายเอกสารและค่าเข้าเล่มรายงานฉบับสมบูรณ์	เป็นเงิน	1,000	บาท
2. ค่าวัสดุอุปกรณ์สำนักงาน	เป็นเงิน	600	บาท
3. ค่าวัสดุอุปกรณ์คอมพิวเตอร์	เป็นเงิน	400	บาท
	รวมเป็นเงินทั้งสิ้น	2,000	บาท

(สองพันบาทถ้วน)

หมายเหตุ ถัวเฉลี่ยทุกรายการ

บทที่2
หลักการและทฤษฎี

2.1 หลักการบีบอัดข้อมูล

ความสนใจในการบีบอัดข้อมูลเริ่มขึ้นเมื่อประมาณ 35 ปีที่แล้วในโลกของอะนาลอกเพื่อลดแบนด์วิธในการส่งสัญญาณวีดีโอ โดยกระบวนการนี้ถูกเรียกว่า การบีบแบนด์วิธ (Bandwidth compression) จากนั้นเมื่อเทคโนโลยีดิจิตอลพัฒนาขึ้นทำให้เกิดการบีบอัดข้อมูลทางดิจิตอลขึ้นและได้รับการพัฒนาอย่างรวดเร็ว

หลักการในการบีบอัดข้อมูลภาพคือ พยายามลดหรือกำจัดส่วนข้อมูลที่เกินความจำเป็นหรือซ้ำซ้อนกัน โดยยังคงข้อมูลไว้เหมือนเดิมซึ่งจะทำให้ข้อมูลภาพมีขนาดเล็กลงจากเดิม และสามารถนำภาพกลับมาแสดงภายหลักโดยผ่านกระบวนการคลาย

ข้อมูลจริงมักถูกแปลงเป็นสำเนาแล้วส่งไปในลักษณะสัญญาณดิจิตอลในการบันทึกรูปภาพในหน่วยความจำ นั้นจะใช้เนื้อที่ในการจัดเก็บมาก เทคนิคในการทำซอร์สโค้ด (เรียกว่า data compression หรือ compression) สามารถใช้การเข้ารหัสและจะได้ข้อมูลใหม่ออกมา วิธีการทำ การบีบอัด สามารถประยุกต์ใช้ได้ที่เห็นชัดนั้นมี 2 ระบบคือ ระบบการสื่อสารโดยการเข้ารหัสข้อมูลเราสามารถที่จะส่งสารสนเทศไปในเวลาที่สั้นกว่าเดิมซึ่งเราสามารถส่งแบนด์วิธเล็กๆแทนที่จะส่งรหัสข้อมูลไปแทนระบบการบันทึก เราสามารถลดเนื้อที่ในการบีบอัดข้อมูล

ในการบีบอัดรูปภาพนั้นจะมีความแตกต่างออกไปจากการบีบอัดข้อมูลเลขในระบบเลขฐานสองแน่นอน โปรแกรมบีบอัดข้อมูลทั่วไปสามารถนำมาใช้ในการบีบอัดข้อมูลรูปภาพได้ แต่ผลที่ได้จากการใช้โปรแกรมรูปที่ได้จะมีประสิทธิภาพต่ำกว่าเพราะว่ารูปภาพจะมีคุณสมบัติเกี่ยวกับทางสถิติที่สามารถนำไปใช้โดยการเข้ารหัสตามชนิดของการออกแบบอย่างไรก็ตาม บางส่วนที่ดีของรูปต้องเสียหายไปในการบันทึกเป็นแบนด์วิธเล็กๆหรือบันทึกในพื้นที่น้อยกว่า อย่างไรก็ตามเราสามารถใช้วิธีดารบีบอัดข้อมูลแบบมีการสูญเสียบางส่วนในพื้นที่ส่วนที่เสียหายไปได้

การบีบอัดแบบสูญเสียคุณภาพ เป็นการบีบอัดข้อมูลและหลังจากการบีบอัดข้อมูลจะได้ข้อมูลที่เหมือนกับข้อมูลเดิม ในกรณีเมื่อมีการจัดการตารางข้อมูลเลขฐานสอง และเอกสารอื่นๆเมื่อทำการบีบอัดข้อมูลแล้วผลที่ได้จากการคลายการบีบอัด สำเนาข้อมูลนั้นจะได้ข้อมูลที่เหมือนข้อมูลจริงออกมา ซึ่งความแตกต่างระหว่างรูปต้นฉบับกับรูปที่ทำการบีบอัดแล้วจะมีความเหมือนกันมาก

ประเภทของการบีบอัดข้อมูลภาพแบ่งได้เป็น 2 แบบคือ

- การบีบอัดข้อมูลแบบไม่มีการสูญเสีย
- การบีบอัดข้อมูลแบบมีการสูญเสียบางส่วน

2.2 การเข้ารหัส (Coding)

กระบวนการที่กำหนดให้การจัดลำดับของเลขฐานสองไปเป็นสัญลักษณ์เรียกว่าการเข้ารหัส (Coding) เซตของการจัดลำดับของเลขฐานสองเรียกว่ารหัส และเลขที่เป็นเลขเฉพาะของเซตเรียกว่า รหัสคำ

การเข้ารหัสที่น่าสนใจคือการเข้ารหัสเอโทรปี (Entropy coding) คือการลดความฟุ่มเฟือยด้วยวิธีการบีบอัดข้อมูลโดยไม่สนใจถึงความหมายของข้อมูล เพียงแต่มองข้อมูลเป็นบิต 0 หรือ 1 ซึ่งในการบีบอัดภาพที่นิยมอยู่ คือ การเข้ารหัสแบบฮัฟฟ์แมน (Huffman coding)

การเข้ารหัสแบบฮัฟฟ์แมน กระบวนการนี้ถูกพัฒนาขึ้นโดย เดวิท ฮัฟฟ์แมน (David Huffman) ซึ่งเป็นสมาชิกในทฤษฎีข่าวสารแรกของ โรเบิร์ต ฟาโน (Robert Fano) ที่ มหาลัยเอ็มไอที การเกิดรหัสโดยใช้กระบวนการนี้จึงเรียกว่ารหัสของฮัฟฟ์แมน เป็นรหัสแบบ รหัสไร้ส่วนนำ และเป็นสถานะการที่ใช้แบบตัวอย่าง (เซตของความน่าจะเป็น)

กระบวนการของฮัฟฟ์แมนมีข้อสังเกตพื้นฐาน 2 ข้อคือ

1. ในข้อมูลปัจจุบัน สัญลักษณ์ที่มีอัตราการเกิดบ่อยครั้ง (มีความน่าจะเป็นในการเกิดสูง) จะมีจำนวนบิตของรหัสข้อมูลสั้นลง

2. ในรหัสข้อมูลปัจจุบัน สัญลักษณ์สองตัวที่มีการปรากฏน้อยที่สุดมีความยาวเท่ากัน

วิธีการสังเกต คือถ้าสัญลักษณ์มีอัตราการเกิดสูงขึ้น จะมีรหัสคำยาวกว่าจำนวนเฉลี่ยบิตต่อสัญลักษณ์ มากกว่าในเงื่อนไขการเปลี่ยนตำแหน่ง ดังนั้นจึงมีการกำหนดให้ รหัสคำที่ยาวกว่าเป็นสัญลักษณ์ที่มีความถี่ในการเกิดสูงกว่า ไม่สามารถระบุเป็นรหัสปัจจุบัน

ในการสังเกตอีกวิธีหนึ่ง พิจารณาตำแหน่งในการคาดคะเน จากรหัสที่เข้ามาที่มีรหัสคำเหมือนกันมีความถี่ในการเกิดต่ำสัญลักษณ์จะมีความยาวเท่ากัน สมมุติให้ความยาวของรหัสคำคือ k บิต ยาวกว่ารหัสคำที่สั้น รหัสคำที่สั้นกว่าจะไม่เป็นคำนำหน้าของรหัสคำที่ยาวกว่า หมายความว่าถ้าเราตกอยู่ที่ k บิตตัวสุดท้ายของรหัสคำที่ยาวกว่ารหัสคำตัวหนึ่งจะมีความชัดเจนขึ้น

เนื่องจากรหัสคำเหมือนกันที่มีอัตราการเกิดต่ำที่สุดในสัญลักษณ์ที่เป็นกลุ่มของตัวอักขระไม่มีรหัสคำตัวใดที่ยาวกว่าตัวนี้ได้ ดังนั้นจึงไม่มีผลกระทบอันใดในการทำให้รหัสคำสั้นลง โดยจะเริ่มที่การเข้ารหัสพรีฟิก ของตัวอื่น เมื่อทำการตกที่ k บิต เราจะให้รหัสใหม่ที่มีความยาวเฉลี่ยสั้นกว่าเดิม กระบวนการเข้ารหัสแบบฮัฟฟ์แมน จะมีวิธีการรวมค่าการเกิดที่ต่ำที่สุด จะมีความแตกต่างที่บิตสุดท้ายเท่านั้น นั่นคือ ถ้า และ คือสัญลักษณ์ 2 ตัวสุดท้ายที่มีอัตราการเกิดต่ำที่สุดในสัญลักษณ์ของตัวอักขระ และถ้ารหัสคำของ คือ m*0 รหัสคำของ รหัสของจะเป็น m*1 โดยที่ m คือค่าของ 1s และ 0s และ *

ข้อมูลแต่ละตัวจะถูกแปลงไปเป็นชุดของตัวเลข 0 และ 1 ซึ่งจะเป็นตัวแสดงถึงเส้นทางของมันในแผนภูมิรูปต้นไม้ของ การเข้ารหัสของ ฮัฟฟ์แมน โดยข้อมูลแต่ละตัวจะมีความยาวของชุด

ตัวเลขดังกล่าวต่างกันไปขึ้นอยู่กับค่าความถี่ของมันที่ปรากฏในข้อมูลทั้งหมด ข้อมูลที่มีความถี่สูงสุด จะมีความยาวของชุดตัวเลขน้อยสุดคือ 2 ตัว ส่วนข้อมูลที่มีความถี่ต่ำสุดจะมีความยาวของชุดตัวเลข 2n-1 ตัวเมื่อ n คือค่าบิตของการเก็บข้อมูลปกติ ยกตัวอย่างเช่น ข้อมูล 8 บิตมีข้อมูลที่เป็นไปได้ 256 ค่า (0-255) เมื่อเข้ารหัสจะมีความยาวของชุดตัวเลขเป็นได้ตั้งแต่ 2 ตัวจนถึง 128 ตัว

ในการถอดรหัสข้อมูลดูจากเส้นทางของข้อมูลเริ่มจากส่วนยอดลงไปเรื่อยๆ จนครบตัวเลขที่เป็นไปได้สำหรับข้อมูลตัวหนึ่งๆ จากนั้นจึงเริ่มดูส่วนยอดลงมาสำหรับข้อมูลตัวถัดไปเช่นนี้ไปเรื่อยๆถึงแม้การเข้ารหัสแบบฮัฟฟ์แมน มักจะช่วยทำให้ความจุของข้อมูลลดลงไม่มากนักแต่มันจะช่วยรักษาข้อมูลเดิมไว้ได้ทั้งหมดเมื่อถอดรหัสมา

การหาความผิดพลาดของการบีบอัดข้อมูล เราสามารถใช้ค่าของ อัตราระหว่างกำลังของสัญญาณเทียบกับสัญญาณ(SNR) และ Peak Signal to Noise Ratio (PSNR) โดยจะเป็นตัววัดคุณภาพของการบีบอัดข้อมูลว่า เมื่อบีบอัดข้อมูลแล้วคุณภาพของภาพที่ได้ใหม่จะมีการสูญเสียมากน้อยเพียงใด SNR ถูกนิยามในหน่วย (db) ดังต่อไปนี้

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

โดยที่ s (m,n) และ s (m,n) คือ ค่าข้อมูลภาพต้นฉบับและภาพที่ถูกคลายหลังการบีบอัดแล้วตามลำดับสำหรับ m และ n เป็นดัชนีตามแนวนอนและแนวตั้งของภาพทั้ง 2 หรืออาจวัดจากความผิดพลาดของการบีบอัดข้อมูลในรูปแบบของ Psnr ดังต่อไปนี้

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

อัตราส่วนของการบีบอัดข้อมูล(Compression Ratio: CR) คือการเปรียบเทียบค่าระหว่างขนาดข้อมูลต้นฉบับ (n1) กับข้อมูลที่ถูกบีบอัดแล้ว (n2)

CR = n1 / n2

2.3 โครงสร้างของ คาลิค

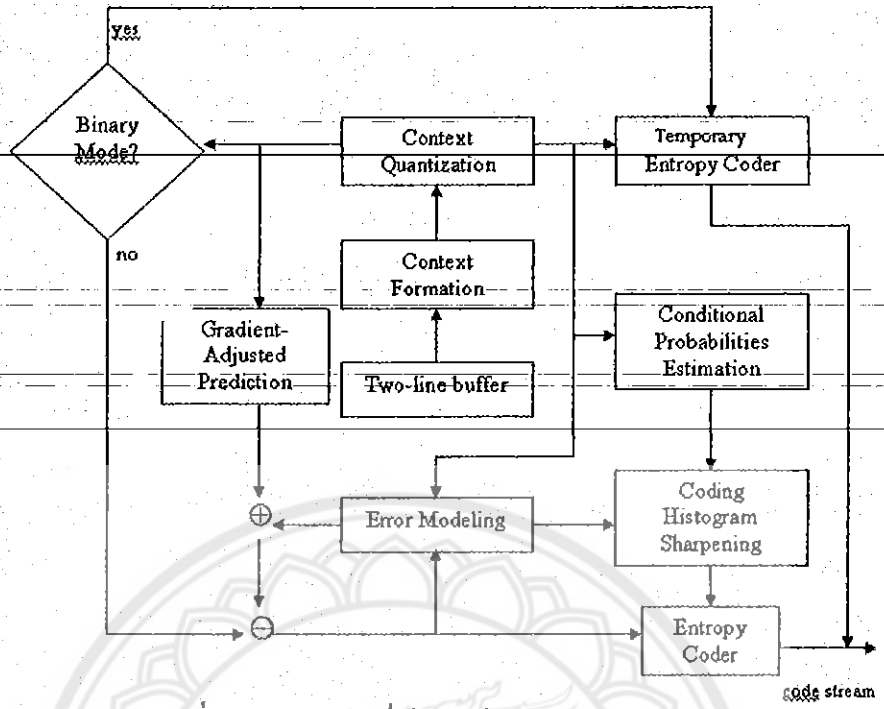
โครงสร้างของคาลิกเมื่อเปรียบเทียบกับ การบีบอัดข้อมูลแบบไม่มีการสูญเสีย อื่นๆ คาลิค มีความเกี่ยวข้องที่ซับซ้อนในที่นี้เราจะมาดูโครงสร้างของคาลิค คาลิคคือการเข้ารหัสและถอดรหัส ในครั้งเดียวโดยการทำแบบราสเตอแสกเนอร์ มันใช้การแสกนไลน์ แบบเก่าของโด๊ต เพื่อทำการคาดคะเนสิ่งแวดล้อมรอบด้านในลำดับให้ได้มาซึ่งประสิทธิภาพสูงสุดในภาพที่มีค่าความแตกต่างของเกรย์สเกล เพียงสองระดับการเข้ารหัส คาลิก จะทำงานเป็นสองแบบคือ ไบนารี่ และ คอนทินิวอัสโทน ระบบจะเลือกทำงานหนึ่งในสองทันทีในขั้นตอนการโค๊ดดิ่งสภาพแวดล้อมของพิกเซล ขณะนั้นไบนารี่โหมด จะถูกดึงออกมาเมื่อความแตกต่างรายละเอียดของพิกเซล น้อยกว่า2ในที่นี้จะมีน้อยมากในการที่จะมี 0 หรือ 1 Grayscale บางตัวที่อยู่ในคอนตินูอัส โทน จะสามารถได้ใน ไบนารี่โหมด ในการนำข้อมูลมาใช้ให้เกิดประโยชน์ ในการเข้ารหัส จะมีรายละเอียดที่แตกต่างกัน 32 แบบใน ไบนารี่โหมด โดยพื้นฐานของ คอนทินิวอัสโทน จะมีข้อหลักๆอยู่ 4 ประการ

GAP, gradient-adjusted prediction. GAP พยามที่จะใช้รายละเอียดของแนวตั้งของข้อมูล เพื่อประมวลผลของความหนาแน่นของ พิกเซล (I) ซึ่งผลลัพธ์คือ I ขั้นตอนนี้เรียกว่า ลิเนีย พรีดิกชั่น

การเลือกรายละเอียดและการสุ่มค่า ในขั้นตอนนี้จะนำออกของความเกี่ยวพันธ์ระหว่างการประมวลที่ผิดพลาดของ GAP โดยภายใต้เงื่อนไขการแยกส่วนที่ผิดพลาดที่แตกต่างกันในการสุ่มหาข้อผิดพลาดผลลัพธ์ ที่ได้จะมีความแตกต่างกัน 8 ระดับ

รายละเอียดของ รูปแบบ ของการคำนวณที่ผิดพลาด, ความผิดพลาดของ รูปแบบ พยามที่จะแยกออกเพื่อจัดเรียงให้อยู่ในพื้นผิวที่แตกต่างและจากนั้นใช้ sample ที่เหมือนกันคือ e- เพื่อปรับปรุง I ผลลัพธ์คือ I~ ขั้นตอนนี้จะเป็นขั้นตอนสุดท้ายนอนลิเนีย

การสูญหายจากการประมวลที่ผิดพลาด ปกติจะใช้หลังจากการประมวลผลแบบจำลองไม่ว่าจะเป็น ฮัฟฟ์แมน โค๊ดดิ่ง หรือ อริสเมติก โค๊ดดิ่ง ก็สามารถนำไปใช้ได้ เพื่อให้ได้ประสิทธิผลที่ดีที่สุด สามารถจะเลือกใช้ อริสเมติก โค๊ดดิ่ง แต่อาจจะไปเกี่ยวข้องกับสิทธิบัตร



รูปที่ 2.1 โครงสร้างที่เป็นองค์ประกอบของ คาลิค

2.3.1 ทูไลน์บัฟเฟอ (Two-line buffer)

สมมุติรูปที่มีขนาด 8 x 8 โดยจะคาดคะเนกับสิ่งแวดล้อมรอบข้างของพิกเซลที่ต้องการหา โดยดูจากจุดที่เราพิจารณาแถวที่ต้องการที่จะหาและแถวที่อยู่ถัดขึ้นไปข้างบนอีกสองแถว

ตารางที่ 2.1ขั้นตอน ทูไลน์บัฟเฟอ

30	2	32	34	29	34	12	12
20	15	99	92	25	8	29	24
31	23	23	32	8	60	45	35
12	18	$\hat{I}$	1	8	0	12	31
12	44	32	22	16	13	32	11
33	10	9	42	12	13	56	24
27	31	12	16	43	48	19	18

โดยที่ค่า I คือค่าเริ่มต้นของพิกเซลนั้น =20

จากรูปเราจะพิจารณาจุดรอบพิกเซล 7 จุดเพื่อหาค่า $\hat{I}$ จุดที่เราพิจารณาก็คือจาก North, West, North-North, West-West, North-West, North-East and North-North-East ตามตารางที่1

2.3.2 คอนเทกซ์ฟอเมชั่น (Context Formation)

Context Formation และ Context Quantization จะมีความเกี่ยวข้องกันและต่อเนื่องกันโดยที่ Context Formation จะเป็นการจัดเรียงกันของข้อมูล จากตารางที่ 2 เราจะแทนตัวเลขในจุดของแต่ละช่องพิกเซลเพื่อที่จะหาขนาดของจุดที่เราต้องการหา $\hat{I}$

ตารางที่ 2.2 คอนเทกซ์ฟอเมชั่น

30	2	32	34	29	34	12	12
20	15	99	92	25	8	29	24
31	23	23	32	8	60	45	35
12	18	$\hat{I}$	1	8	0	12	31
12	44	32	22	16	13	32	11
33	10	9	42	12	13	56	24
27	31	12	16	43	48	19	18

จากตารางที่ 2 จะมีค่าของแต่ละพิกเซลเพื่อนำค่าไปคำนวนหาค่า $\hat{I}$ โดยจุดที่เราต้องการหาคือ $\hat{I}$ คำนวณโดยการใช้สมการ

$d_v = |I_w - I_{nw}| + |I_n - I_{nn}| + |I_{ne} - I_{nne}| = |5| + |76| + |60| = 141$

$d_h = |I_w - I_{ww}| + |I_n - I_{nw}| + |I_n - I_{ne}| = |6| + |0| + |24| = 30$

$D = d_v - d_h$

if $(D > 80)\ \hat{I} = I_w$

elseif $(D < -80)\ \hat{I} = I_n$

elseif

$\hat{I} = (I_w + I_n) = 2 + (I_{ne} - I_{nw}) / 4$

If $(D > 32)\ \hat{I} = (\hat{I} + I_w) / 2$

Elseif $(D > 8)\ \hat{I} = (3\ \hat{I} + I_w) / 4$

Elseif $(D < -32)\ \hat{I} = (\hat{I} + I_n) / 2$

Elseif $(D < -8)\ \hat{I} = (3\ \hat{I} + I_n) / 4$

เมื่อได้ค่า d_v และ d_h เราสามารถหาดา D ได้จาก $D = d_v - d_h = 141-30 = 111$ จากนั้นเช็คเงื่อนไขในซูโดโค๊ต โดย $D = 111$ ดังนั้นจะอยู่ในกรณีที่ ($D > 80$) $\hat{I} = I_w$ โดย $\hat{I}$ จะมีค่า $= 18$ แล้วนำค่าไปใส่ในตาราง

ตารางที่ 2.3 แสดงค่าของ $\hat{I}$

30	2	32	34	29	34	12	12
31	15	65	53	40	18	39	24
30	27	16	15	16	35	35	35
12	18	18	21	7	12	11	31
12	42	22	24	13	13	38	11
33	15	17	31	14	22	45	24
27	31	12	16	43	48	19	18

จากสมการค่า b_k คือผลลัพธ์ที่ได้จากการ Quantization แล้วโดยที่ x_kคือค่าของจุดพิกเซลเริ่มต้น ของพิกเซลนั้น (ตารางที่1) โดยนำค่า x_k ที่ได้มาเปรียบเทียบกับค่า $\hat{I}$ (ตารางที่3) โดย ถ้า x_kมีค่ามากกว่าหรือเท่ากับ $\hat{I}$ ผลลัพธ์ของการ Quantization เท่ากับ 0 แต่ถ้า x_k มีค่าน้อยกว่า $\hat{I}$ ผลลัพธ์จากการ Quantization มีค่าเท่ากับ 1 ซึ่งผลของการquantization จะมีค่าคือ 0 กับ 1 เท่านั้นดังตาราง

2.3.3 Context Quantization

Context Quantization จะเป็นวิธีการลดจำนวนบิตที่ใช้ในการจัดเก็บลงโดยเปลี่ยนค่าเป็นเลข 0กับ 1

เมื่อได้ขนาดของ $\hat{I}$ ครบทุกจุดแล้วจะได้ค่าในตาราง4 จากนั้นทำการ quantizationใส่สมการ

$$(0 \le k < K = 8) : b_{k=} \begin{cases} 0, & if x_k \ge \hat{I} \\ 1, & if x < \hat{I} \end{cases} \qquad (3)$$

จากสมการค่า b_k คือผลลัพธ์ที่ได้จากการ Quantization แล้วโดยที่ x_kคือค่าของจุดพิกเซลเริ่มต้น ของพิกเซลนั้น (ตารางที่1) โดยนำค่า x_k ที่ได้มาเปรียบเทียบกับค่า $\hat{I}$ (ตารางที่3) โดย ถ้า x_kมีค่ามากกว่าหรือเท่ากับ $\hat{I}$ ผลลัพธ์ของการ Quantization เท่ากับ 0 แต่ถ้า x_k มีค่าน้อยกว่า $\hat{I}$ ผลลัพธ์จากการ Quantization มีค่าเท่ากับ 1 ซึ่งผลของการquantization จะมีค่าคือ 0 กับ 1 เท่านั้นดังตาราง

ตารางที่ 2.4 Quantization

0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	0	1	1	0	1	1	0
0	0	1	0	1	0	1	0
0	1	1	0	1	0	0	0
0	0	0	1	0	0	1	0
0	0	0	0	0	0	0	0

เมื่อทำการ Quantization แล้วจะได้ตารางค่าไบนารี่ (ตารางที่ 4) เพื่อนำไปเช็คว่าเป็นไบนารี่หรือไม่

2.3.4 ไบนารี่โมด (Binary mode)

เมื่อทำการ Quantization แล้วจะได้ตารางค่าไบนารี่ (ตารางที่ 4) อัลกอริทึมจะทำการเช็คพิกเซลใส่เตียง I_{ww}, I_{w}, I_{nw}, I_{n}, I_{ne}, I_{nn} ถ้าจุดทั้งหกนี้มีค่าต่างกันไม่เกิน 2 จะเป็นไบนารี่ถ้ามีค่าต่างกันเกิน 2 จะไป continuous tone ในไบนารี่โหมดนั้น จะกำหนด $s_1 = I_w$ และให้ค่าอื่นๆคือ s_2 ใส่รูปสมการ

$$T = \begin{cases} 0, & if\ I = s_1 \\ 1, & if\ I = s_2 \\ 2, & otherwise \end{cases} \qquad (4)$$

โดยค่า T คือค่าที่ได้จากการ ternary code จากสมการ (4) ให้นำค่า s_1 (เท่ากับค่า I_w) มาเช็คว่าเท่ากับค่า I หรือ ไม่ ถ้า s_1 มีค่าเท่ากับ I ให้ T มีค่า = 0 แต่ถ้าค่า s_1 มีค่าไม่เท่ากับ I (เท่ากับ s_2) ให้ T มีค่าเท่ากับ 1 เมื่อทำการเปรียบเทียบค่า T ทุกจุดบนรูปแล้ว จะได้ตารางที่มีค่า ที่ได้จากการ Ternary code ตามตารางที่ 5

ตารางที่2.5 แสดงค่าที่ได้จากการ Ternary

0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	0	1	0	1	1	0	0
0	0	1	1	1	1	1	1
0	1	0	1	1	1	0	0
0	0	0	1	1	0	1	0
0	0	0	0	0	0	0	0

2.3.5 Conditionnal Probabilities Estimation

$$CB = QB(I_w)QB(I_n)QB(I_{nw})QB(I_{ne})QB(I_{ww})QB(I_{nn}) \quad (5)$$

$$Q_B(x) = \begin{cases} 0, if\ x = s_1 \\ 1, if\ x = s_2 \end{cases} \quad (6)$$

คือเงื่อนไขการประมาณเงื่อนไขความน่าจะเป็นที่เกิดขึ้นจากสมการที่ (5) (6) โดยค่าบิตแรกของ CB จะบอกถึงค่า I_w เนื่องจาก I_w มีค่าใกล้เคียงกับ s_1 ดังนั้นเราสามารถเพิกเฉยค่า I_w ดังนั้นผลต่างของไบนารี่คือ $2^5 = 32$ ค่าของผลต่าง คาลิกจะใช้ 32 เงื่อนไขของความน่าจะเป็น คือ $p(T|C_B)$ในการเข้ารหัส

การคำนวณค่าคงที่และจุดเริ่มต้นที่ให้ในซูโดโค้ดคือการเลือกจากการสุ่มโดยจะขึ้นอยู่กับการทดสอบตัวเลขที่มีความเกี่ยวเนื่องกัน อนึ่งสามารถใช้เกณฑ์ของการประเมินที่ต่ำสุดของความผิดพลาดให้เหมาะสมแต่ความในที่นี้ไม่สามารถที่จะทำให้ผลลัพธ์ดีขึ้นมากนักเหตุผลหนึ่งคือผลกระทบจากการย้อนกลับของความผิดพลาดโมเดลลิ่ง

ขั้นตอนนี้จะกระทำแต่การดึงความซ้ำซ้อนที่มีอยู่ในรูปภาพเท่านั้นและจะถูกสังเกตความเปลี่ยนแปลงของการทำนายข้อผิดพลาด $e = I - \hat{I}$ ที่สำคัญความสัมพันธ์ของความเรียบของภาพโดยรอบของพิกเซล x ขั้นต่อไปคือการนำความทับซ้อนออกจากพื้นที่เรียบของภาพโดยคาลิดจะนิยามความเป็นไปได้คือ

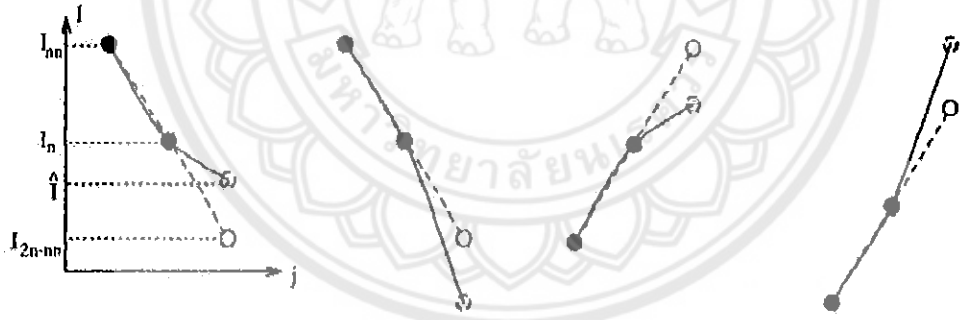
$$\Delta = d_h + d_v + 2|e_w|$$
$$e_w = I_w - \hat{I}_w \quad (9)$$

สาเหตุที่ใส่ ew คือมีแนวโน้มที่จะเกิดความผิดพลาดขนาดใหญ่ติดต่อกันจากสมการสมการเดลต้าโดยนำไปใช้กับการประเมินออฟไลน์ลิทสแควร์ตามความเป็นจริงอาจไม่ค่อยดีนักที่จะนำไปใช้ประโยชน์ที่ได้รับจากการใช้เดลต้าคือเราสามารถกำหนดเงื่อนไขของความผิดพลาดให้อยู่ในลำดับที่แตกต่างกันได้เอนโทพี โค๊ดดิ่งจะใช้ความเป็นไปได้ของความน่าจะเป็นและปรับปรุงให้ดีขึ้นโดยใช้ P(e)รูปที่มี △ จำนวนมากจะเกิดความแตกต่างมาก โดยคาลิคจะเลือกสัญญาณการประมวลผล △ 8 ระดับเช่น Q คือสัญญาณการประมวลผลของ △ Q : △→ 0,1,...,7 เป็นเงื่อนไขของการเอ็นโทปีของความผิดพลาดเมื่อเสร็จสิ้นกระบวนการออกแบบคาลิคจะมีประสิทธิภาพตามเลเวลเริ่มต้นที่เพิ่มขึ้น

T : 5; 15; 25; 42; 60; 85; 140

สัญญาณการประมวลผล 8 ระดับสำหรับข้อผิดพลาดสามารถสร้างตัวอย่างในการสร้างแบบจำลองของคอนแทคโมเดลถิ่งจาก p(e|Q(△)อย่างรวดเร็วในการเข้ารหัสและจะช่วยในการประหยัดหน่วยความจำในการเข้ารหัส

แม้ว่า GAP จะคาดคะแนสิ่งแวดล้อมรอบข้างได้อย่างดีแต่สามารถจับบริบทที่แตกต่างกันได้ในการทำนายผลลัพธ์คาลิดจะพยายามคาดคะเนข้อผิดพลาดของแบบจำลอง เช่น การออกแบบโครงสร้างพื้นฐานและและลักษณะพิเศษของรูปภาพ



รูป 2.3 พิกเซลใกล้เคียง

2.4 การบีบอัดข้อมูลแบบอื่นนอกจากคาลิด

2.4.1. JPEG

JPEG (Joint Photographic Experts Group - "เจเพ็ก") คือรูปแบบการบีบอัดแฟ้มภาพแบบไม่สูญเสีย โดยยังให้เสียความละเอียดน้อยที่สุด รูปแบบแฟ้มสำหรับวิธีการนี้ได้แก่ .jpeg, .jpg, .jpe, .jfif, .jfi รูปแบบแฟ้ม JPEG นี้ เป็นรูปแบบแฟ้มที่ใช้กันในการจัดเก็บและแลกเปลี่ยนรูปภาพบนเวิลด์ไวด์เว็บมากที่สุด โดยเฉพาะภาพถ่าย เนื่องจากสามารถเก็บความละเอียดสูงได้โดยใช้ขนาดไฟล์ที่เล็ก สามารถเก็บภาพสีได้หลากหลายระดับความแม่นยำของสี(Bit Depth) ความสามารถใน

การย่อขนาดไฟล์ของแฟ้ม JPEG นั้นเกิดจากการใช้เทคนิคการย่อขนาดภาพแบบการบีบอัดคงข้อมูลหลัก (Lossy Compression) หรือการบีบอัดแบบมีความสูญเสียทำให้ไม่นิยมใช้กับภาพที่เป็นลายเส้นหรือไอคอนต่าง ๆ เนื่องจากจะไม่ได้ประสิทธิภาพเท่าการเก็บในรูปแบบอื่นอย่าง PNG หรือ GIF การบีบอัดของ JPEG นั้นจะใช้เทคนิคที่เรียกว่า DCT (Discrete Cosine Transform) ซึ่งเป็นการแปลงค่าความสว่างของภาพให้อยู่ในรูปแบบเชิงความถี่ (Frequency Domain) ทำให้สามารถเลือกแทนค่าของสัมประสิทธิ์หรือในที่นี้คือแอมพลิจูดของค่าความถี่ต่างๆ ได้โดยอาศัยตัวแปรที่มีนัยสำคัญที่ต่างกันได้ การที่สามารถลดนัยสำคัญของค่าตัวเลขลงไปได้ทำให้สามารถลดขนาดของหน่วยความจำหรือขนาดไฟล์ที่ใช้เก็บตามไปได้

2.4.2. PNG

ภาพกราฟฟิกแบบ PNG หรืออ่านออกเสียงว่า "ปิง" นั้นกำเนิดขึ้นในช่วงประมาณปี 1995 ด้วยเจตนารมย์ที่จะหารูปแบบไฟล์ที่เหมาะสำหรับแสดงผลบนเว็บ และเพื่อมาใช้แทนฟอร์แมต GIFในอีกมิติหนึ่ง คือ มิติทางเทคนิก ข้อดีประการต่อมาของ PNG ก็คือความสามารถในการแสดงผลในเชิงคุณภาพก็คือ PNG รับรองการแสดงผลสีได้มากถึง 48 บิต, การบีบอัดแบบไม่สูญเสีย (ซึ่งทำให้ภาพมีขนาดเล็ก แต่ยังคงความสวยงามได้เทียบเท่าต้นฉบับ และที่น่าสนใจที่สุด คือ การควบคุมการแสดงภาพแบบโปร่งแสงได้ตามต้องการในขณะที่ PNG รองรับ Alpha Transparency ฟอร์แมตแบบ GIF รองรับ เพียง Binary Transparency ซึ่งมีนัยสำคัญสองประการคือGIF นั้นสามารถทำภาพได้แบบมีพื้นหลัง (สมมติค่าเป็น 1) คือเป็นสีทึบ กับ แบบที่เป็นโปร่งใส หรือที่เรียกกันคุ้นปากว่า transparentg ติค่าเป็น 0) คือไร้สีโดยสิ้นเชิง สองอย่างนี้เท่านั้น ดังนั้นจะไม่มีระดับอื่น เช่น 0.2 หรือ 0.85 ได้เลย จึงเรียกคุณสมบัตินี้ว่า Binary Transparency ประการต่อมา คือ หากเราใช้ Transparency จะต้องมีการใส่ค่า matte เพื่อให้ขอบของภาพกลืนไปกับสีของพื้น หากเปรียบเทียบกับ PNG แล้ว ปัญหาทั้งสองประการข้างต้น จะไม่เกิดขึ้น เพราะ PNG สามารถให้ค่าความโปร่งแสงได้หลายระดับ และ PNG แบบเจาะใสนั้น จะไม่ขึ้นอยู่กับสีของพื้นหลัง

2.4.3. GIF

รูปแบบไฟล์ GIF ได้รับการออกแบบโดย CompuServe ซึ่งเป็นระบบเครือข่ายข่าวสารแบบออนไลน์ เพื่อให้บริการแลกเปลี่ยนกราฟิกในรูปแบบ bitmap ที่มีการจัดการทางด้านหน่วยความจำที่มีประสิทธิภาพ ข้อจำกัดของภาพแบบ GIF คือ ความสามารถทางด้านสีซึ่งเป็นแผงสีแบบอินเด็กซ์ (ภาพสีแบบ 24 บิตไม่สามารถใช้ได้) แผงสีสามารถบรรจุได้ 2 ถึง 256 สี ซึ่งถูกสร้างจากข้อมูลสี 24 บิต ไฟล์แบบ GIF ถูกบีบขนาดโดยใช้การบีบขนาด LZW แบบประยุกต์ การขยายไฟล์ข้อมูลแบบ GIF กลับคืน จะช้ากว่าการบีบขนาดแบบ RLE แต่จะเปลืองเนื้อที่หน่วยความจำน้อยกว่า

รูปแบบไฟล์ GIF เป็นภาพซึ่งใช้สีจำกัด ไม่เกิน 256 สี ไม่ใช้ทั้งหมดของสเปกตรัมสีที่แสดงได้บนมอนิเตอร์ เหมาะสำหรับภาพที่ต้องการไฟล์ขนาดเล็ก โหลดเร็ว ไฟล์แบบนี้จึงเหมาะกับงานที่ใช้สีแบบ solid color เช่น โลโก้ หรือ ภาพแบบ Illustration

2.4.4. Gzip

Gzip อาศัยอัลกอริทึมชื่อ DEFLATE ซึ่งผสมผสาน LZ77 กับรหัสฮัฟแมน DEFLATE ออกแบบมาแทนที่ LZW และอัลกอริทึมบีบอัดข้อมูลอื่นที่มีปัญหาเรื่องสิทธิบัตร ทำให้ใช้โปรแกรมเช่น compress ได้จำกัดคำว่า gzip ยังนิยมเรียกรูปแบบไฟล์ที่สร้างโดยโปรแกรม gzip ซึ่งประกอบด้วย ส่วนหัวไฟล์ หรือ เฮดเดอร์ ขนาด 10 ไบต์ มีสัญลักษณ์ของไฟล์ หมายเลขเวอร์ชัน วันและเวลา เฮดเดอร์เพิ่มเติมเช่นชื่อไฟล์เดิม ส่วนนี้อาจไม่มีก็ได้ ข้อมูลไฟล์ที่บีบโดยอัลกอริทึม DEFLATE ส่วนท้ายไฟล์ขนาด 8 ไบต์ ประกอบด้วยค่า CRC-32 และความยาวของข้อมูลเดิมก่อนบีบอัด ถึงแม้ว่าไฟล์รูปแบบนี้สามารถเก็บไฟล์หลายไฟล์ต่อ ๆ กัน gzip มักถูกใช้บีบเพียงไฟล์เดียว โดยไฟล์ต่าง ๆ จะถูกรวมเข้าด้วยกันก่อนโดยใช้โปรแกรม tar แล้วค่อยบีบอัดไฟล์ที่ได้ด้วย gzip ไฟล์ที่ได้มักใช้นามสกุล .tar.gz หรือ .tgz และนิยมเรียกว่า "compressed tarball" gzip เป็นไฟล์คนละชนิดกับ ZIP ซึ่งใช้อัลกอริทึม DEFLATE เช่นเดียวกัน ZIP สามารถเก็บไฟล์หลายไฟล์ในตัวมันเองโดยไม่ต้องพึ่งโปรแกรมอื่นแต่ก็ไม่ได้ ขนาดเล็กเท่าการบีบโดยใช้ tar และ gzip เนื่องจาก ZIP บีบข้อมูลแต่ละไฟล์แยกจากกัน แต่ gzip สามารถใช้ข้อมูลที่ซ้ำซ้อนระหว่างไฟล์มาเป็นประโยชน์ในการบีบอัด อัลกอริทึมใน gzip ยังออกมาในรูปของไลบรารีชื่อ zlib ซึ่งสนับสนุนรูปแบบไฟล์ gzip และรูปแบบเฉพาะของมันเองที่ชื่อ zlib stream รูปแบบ zlib stream, DEFLATE และ gzip กำหนดโดยมาตรฐาน RFC 1950, RFC 1951 และ RFC 1952 ตามลำดับ เนื่องจากส่วนหัวไฟล์ gzip มีวันที่และเวลาที่สร้าง ในการเปรียบเทียบข้อมูลในไฟล์สองไฟล์ว่าเหมือนกันหรือไม่จำเป็นต้องใช้โปรแกรม zcmp หรือ zdiff ถึงจะได้ผลที่ถูกต้อง โปรแกรมสำหรับคลายข้อมูลที่บีบโดย gzip เรียกว่า gunzip

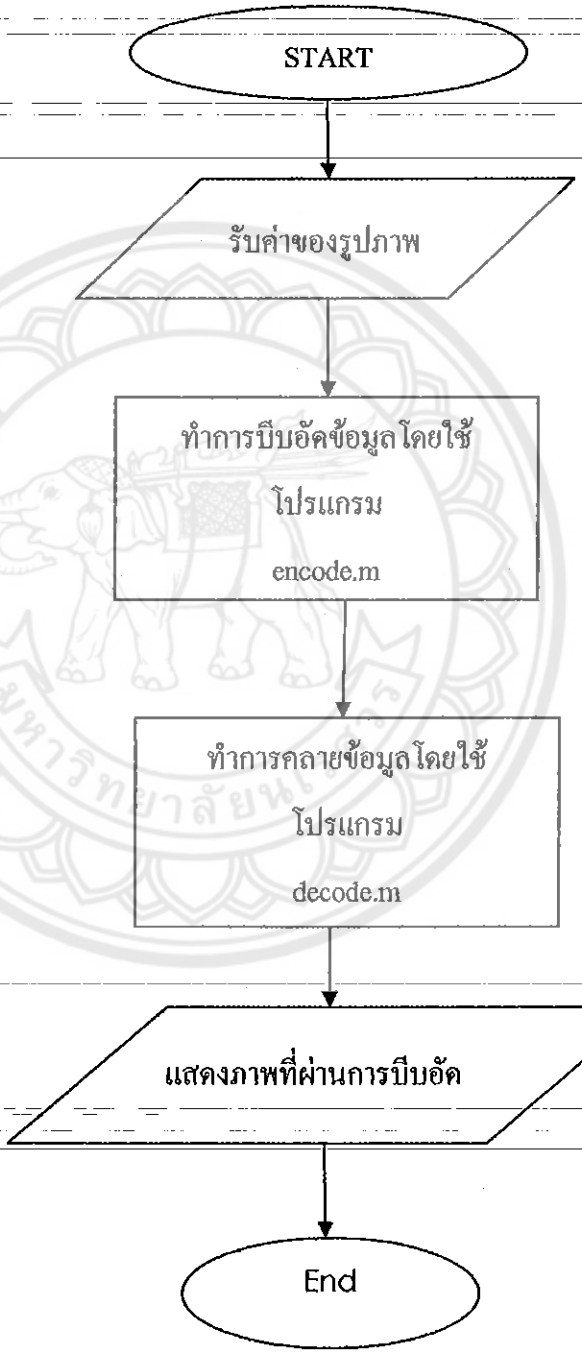
บทที่ 3
ขั้นตอนการดำเนินการ

ในการบีบอัดข้อมูลโดยวิธีการของ คาลิค ได้แบ่งขั้นตอนการดำเนินงานออกเป็น 3 ขั้นตอน โดยเริ่มจากการศึกษาอัลกอริทึมและขั้นตอนการออกแบบอัลกอริทึมของ คาลิค การเขียนโปรแกรมเพื่อทดสอบอัลกอริทึมของ คาลิค และนำผลที่ได้ไปเปรียบเทียบกับอัลกอริทึมอื่น แสดงดังรูปที่ 3.1

รูปที่ 3.1 ขั้นตอนการดำเนินงาน

3.1 ศึกษาอัลกอรึทึมและขั้นตอนการออกแบบอัลกอริทึมของ คาลิค

จากบทที่ 2 ศึกษาทฤษฎีและอัลกอลิทึมที่เกี่ยวข้อง เราสามารถนำทฤษฎีและอัลกอลิทึมที่ได้มาออกแบบโปรแกรมการบีบอัดข้อมูลภาพด้วยวิธีการของ คาลิค และนำมาเขียนเป็น บล็อคไดอะแกรม ได้ดังรูปที่ต่อไปนี้

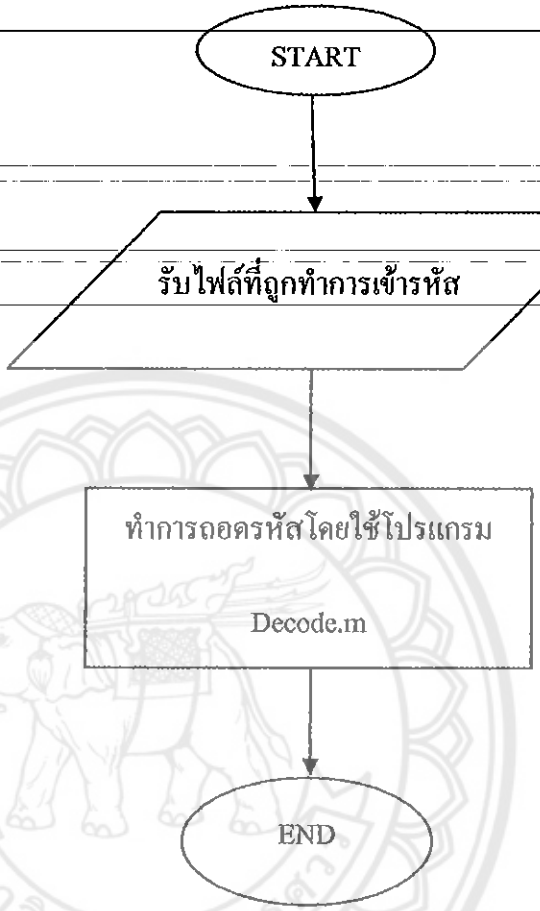


รูปที่ 3.2 แผนภาพโปรแกรมการบีบอัดข้อมูลแบบคาลิก

3.2 โปรแกรมการเข้ารหัส

รูปที่ 3.3 แผนภาพโปรแกรมการเข้ารหัส

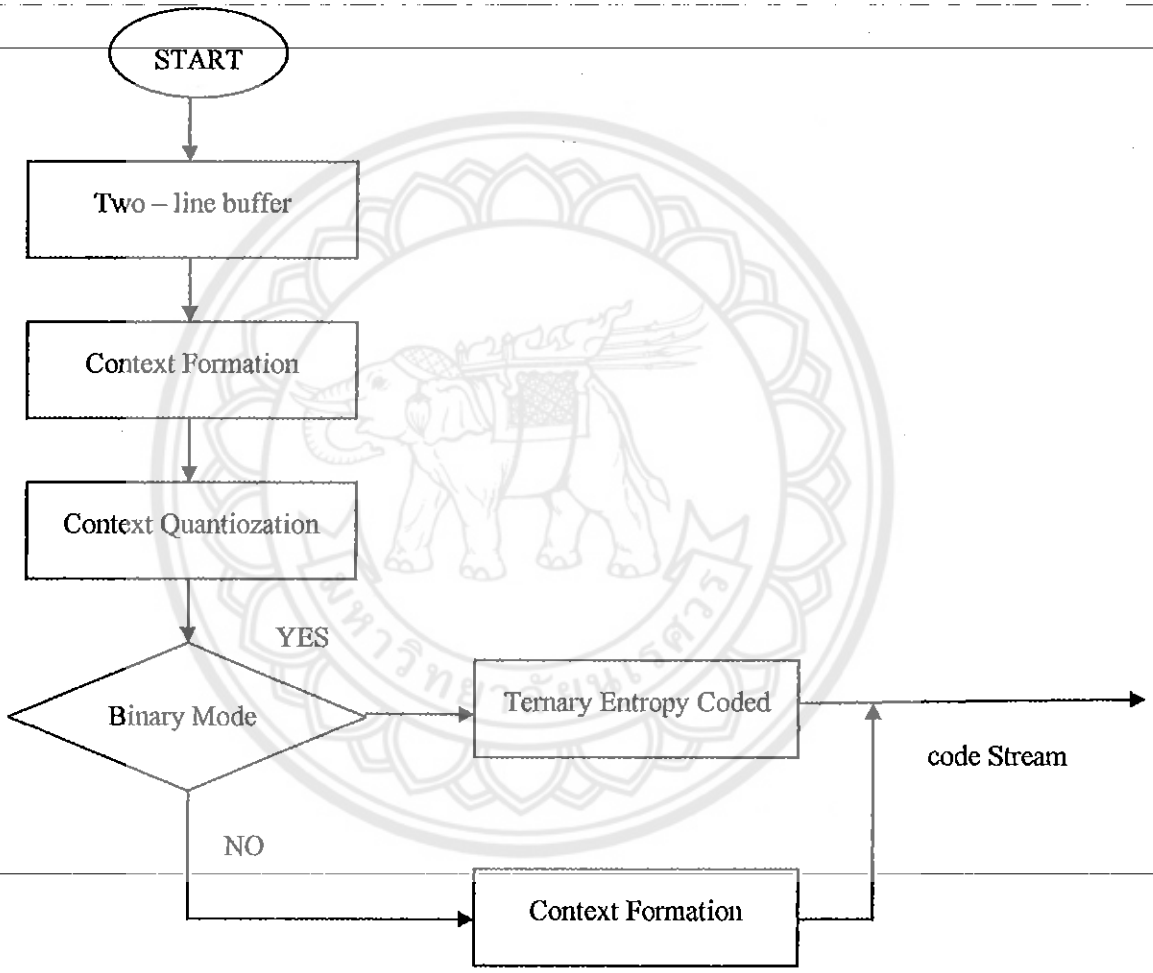
3.3 โปรแกรมการถอดรหัส



รูปที่ 3.4 แผนภาพโปรแกรมการถอดรหัส

3.3.1 ฟังก์ชั่นการเข้ารหัส

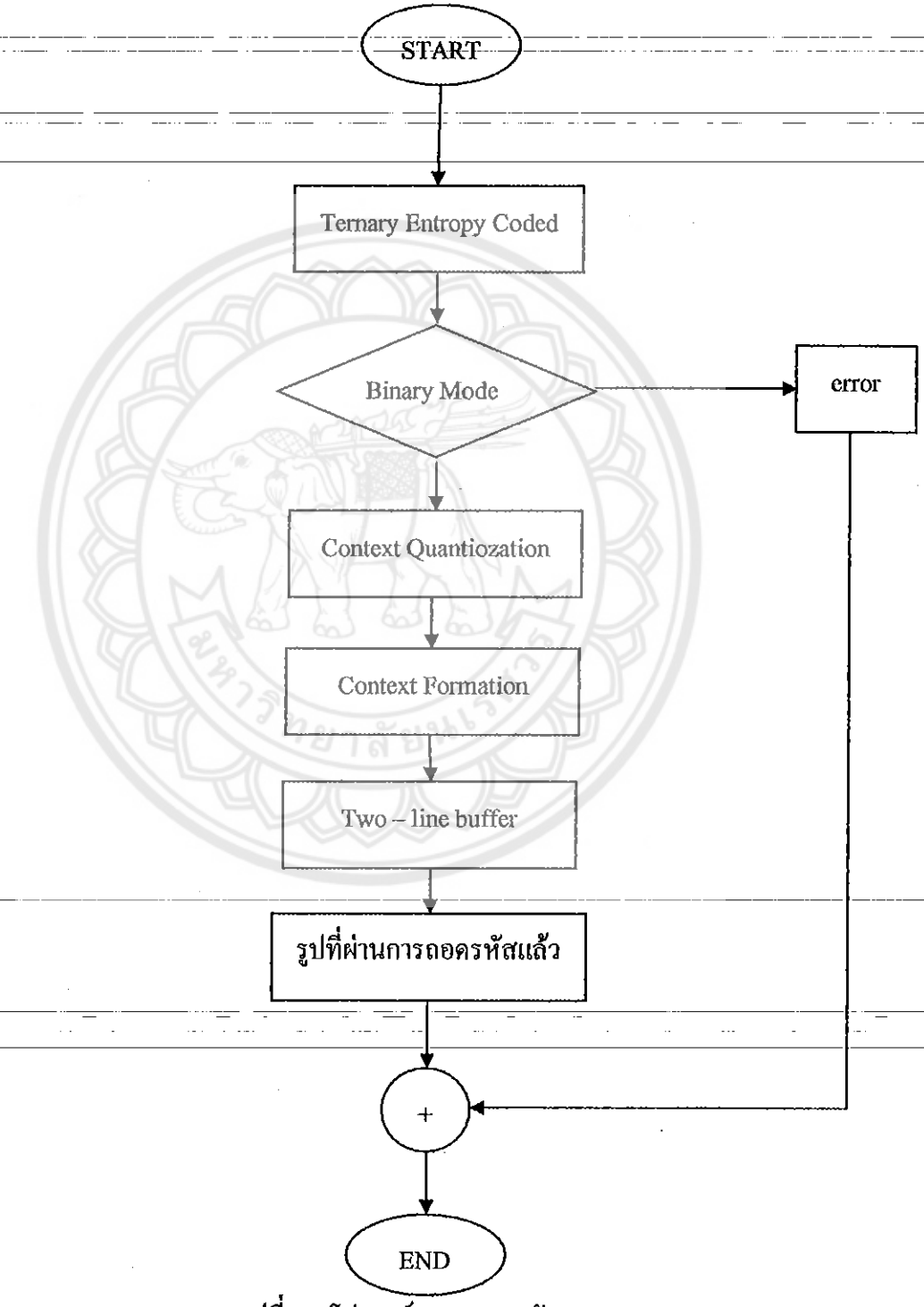
ทำหน้าที่ในการเขียนข้อมูล ขั้นตอนการทำงานของฟังก์ชั่นการเข้ารหัสโดยเริ่มที่ Start และกำหนดค่าเริ่มต้นให้ y และ i = 0 และนำภาพมาหาความยาวเปรียบเทียบขนาดกับอินพุต ถ้าเป็นจริงให้กำหนดค่า j และเปรียบเทียบกับอินพุตจะได้โค๊ดและแสดงดังรูปที่ 3.3 และตัวโปรแกรมดูจากภาคผนวก



รูปที่ 3.5 โฟลชาร์ตการเข้ารหัส

3.3.2 ฟังก์ชั่นการถอดรหัส

ทำหน้าที่ในการเขียนข้อมูล ขั้นตอนการทำงานของฟังก์ชั่นการเข้ารหัสโดยการนำภาพที่ถูกเข้ารหัสมาโดยทำสลับกับวิธีการเข้ารหัสโดยหาขนาดของ dict และนำมาเปรียบเทียบกับกับค่าโค๊ด โดยกำหนดค่าเริ่มต้น i=0 แสดงดังรูปที่ 3.4 และตัวโปรแกรมดูจากภาคผนวก



รูปที่ 3.6 โฟลชาร์ตการถอดรหัส

ในขณะที่เทคนิคการบีบอัดข้อมูลแบบพื้นฐานจะมุ่งเน้นแต่การทำ 1 มิติแต่การบีบภาพที่ดีนั้นจะต้องมีการใช้ประโยชน์ของความพิเศษของข้อมูลภาพในความรู้สึกของความเกี่ยวข้องของ 2 มิติ โดยพื้นฐานเทคนิค การบีบอัดภาพแบบไม่สูญเสียข้อมูล สามารถแยกความแตกต่างออกได้ชัดเจนสองส่วนคือ แบบจำลอง และ โค๊ดดิ่ง ซึ่งจะเห็นกันอยู่บ่อยๆ ภาพถ่ายจะมีการเน้นแบบจุดต่อจุดในบางส่วน (ส่วนใหญ่จะเป็น raster-scan line) ในส่วนของแบบจำลอง จะมุ่งเน้นไปที่การรวมของข้อมูลซึ่งจะขึ้นอยู่กับความน่าจะเป็นของแบบจำลองเพื่อเป็นรูปแบบของการวินิจฉัย ปัญหาที่เกิดขึ้นในแต่ละครั้งของการทำงานและหลังจากการสแกนข้อมูล xt = x1x2::xt อนุมานกับค่า Sample xt+1 โดยการสั่งให้เป็นเงื่อนไขการจัดสรรของความน่าจะเป็น ลำดับต่อมาคือการ โค๊ด ส่วนมากจะจัดการให้สมบูรณ์โดยอริเมติกโค๊ดดิ่งหรือ ฮัฟฟ์แมนโค๊ดดิ่ง ในขั้นตอนนี้การเข้ารหัสจะเป็นเงื่อนไขทุกครั้งไปตั้งแต่ใส่โค้ดจนถึงข้อจำกัดในส่วนที่ไม่สามารถทำได้เมื่อมีข้อจำกัดขั้นตอนในการทำโค้ด ทั้ง ที่ไม่มีสิ่งเกี่ยวข้องและจนถึงการปรับปรุงจนถึงที่สุดแล้วจากภาพที่เป็นต้นแบบเทคนิคและสิ่งที่ ทำได้ดีที่สุดคือแบบจำลองเทคนิคที่ได้ผลดีที่สุดคือการปรับเปลี่ยนความเป็นไปได้ของแบบจำลอง เพื่อให้ได้ผลลัพธ์ออกมาดีโดยพื้นฐานแล้วสามารถแยกออกได้เป็นดังนี้

1. ขั้นตอนการคาดเดา ค่าของ $\hat{X}_{t+1}$ คือการเดาของ Sample x_{t+1} ขึ้นอยู่กับขอบเขตของตัวเลขย่อย (a causal template) ของข้อมูลที่มีอยู่ x_t
2. การคำนวณตามข้อมูลที่มีอยู่คือ x_{t+1} ตามความเข้าใจก็ยังคือ causal template
3. ความเป็นไปได้ของความผิดพลาด $e_{t+1} = x_{t+1} - \hat{X}_{t+1}$ ภายใต้เงื่อนไขของ x_{t+1}

การคาดเดาที่เหลือคือโค้ดขึ้นอยู่กับการกระจายความเป็นไปได้เมื่อมีความเหมือนและสามารถนำมาใช้ได้ระหว่าง การคาดเดา การกระจายของความเป็นไปได้และ และความเข้าใจเพื่อถอดรหัสออกมาซึ่งสามารถแก้ไขจำนวนที่ถูกต้องของรหัส เมื่อภาพมี แบบจำลองที่ดีกว่าที่ดีกว่าอาจจะทำให้มีผลกระทบต่อการบีบอัดเพิ่มขึ้นขณะที่ประสิทธิผลลดลง ข้อแลกเปลี่ยนที่ดีในเรื่องนี้คือการเฝ้าศึกษาคอนตินิวอัสโทน ของภาพคือกุญแจที่จะนำไปสู่การทำให้บรรลุผลสำเร็จสูงสุด

The GAP predictor, ใช้ในคาลิดเพื่อรวมกันระหว่างเส้นแนวตั้งและแนวนอน GAP จะตรวจจับความเปลี่ยนแปลงของเส้นอย่างไรโดยรอบพิกเซล x จากนั้นจึงทำการจัดเรียงแนวโน้มของเส้นที่เปลี่ยนแปลงให้อยู่ในระดับของ ความคมชัด ธรรมดาและไม่ชัด ซึ่งจะให้น้ำหนักที่แตกต่างของพิกเซลใกล้เคียงสำหรับการทำ ลิเนียพริดิกชั่นพิกเซลx

บทที่ 4

การทดลอง

4.1 วัตถุประสงค์การทดลอง

1.ทดลองการบีบอัดข้อมูลภาพโดยวิธีการแบบคาลิค

2.แสดงผลการเปรียบเทียบแบบคาลิค JPEGและ GZIP

4.2 การเรียกใช้โปรแกรม

โปรแกรมการบีบอัดข้อมูลภาพโดยวิธีการแบบคาลิคเรียกใช้โปรแกรมผ่านโปรแกรม MATLAB

4.3 การทดลองบีบอัดแบบคาลิค

การทดลองจะทำการบีบอัดข้อมูลแบบคาลิคโดยใช้โปรแกรม MATLAB เปรียบเทียบกับการบีบอัดแบบ JPEGและ Gzip โดยจะวัดจากอัตราการบีบอัดข้อมูลภาพ โดยไฟล์ตัวอย่างที่ใช้ในการทดลองเป็นไฟล์ที่เป็นไฟล์ภาพ bmp ขนาด 8 bits

4.4 ผลการทดลอง

ผลการทดลองที่ 1

รูปที่4.1 ไฟล์รูปขนาด 20x20

ตารางที่ 4.1 ผลการทดลองที่1

ชนิดของโปรแกรมบีบอัดข้อมูล	Total time (S)	COMPRESSION RATIO
CALIC	0.18	1.205
LJPEG	0.1	1.308
GZIP	0.08	2.011

Start Encoding...

Picture Name: 20x20.bmp

Start timeDate: 8-5-2009--2:57:0.953

Stop timeDate: 8-5-2009--2:57:1.843

Original Size: 1440

Picture Size: 400

จากผลการทดลอง จะเห็นได้ว่าค่า COMPRESSION RATIO ของ คาลิคมีค่าน้อยที่สุดในบรรดาโปรแกรมบีบอัดที่นำมาเปรียบเทียบแต่คาลิคจะใช้ระยะเวลาในการเข้ารหัสนานกว่าการบีบอัดชนิดอื่น โดยขนาดภาพก่อนการบีบอัดข้อมูลซึ่งมีขนาดไฟล์ที่จัดเก็บ 1440 ไบท์เมื่อผ่านการบีบอัดข้อมูลโดยใช้คาลิคได้ไฟล์บีบอัดข้อมูลภาพ 400 ไบท์และทำการเปรียบเทียบภาพทั้งสองได้ค่า SNR และ PSNR เข้าสู่อนันต์ นั้นแสดงให้เห็นว่าการบีบอัดและการคลายการบีบอัดข้อมูลภาพขนาด 20x20 มีลักษณะเหมือนกันทุกประการ

ผลการทดลองที่ 2

รูปที่4.2 ไฟล์รูปขนาด 130x92

ตารางที่ 4.2 ผลการทดลองที่ 2

ชนิดของโปรแกรมบีบอัดข้อมูล	Total time (S)	COMPRESSION RATIO
CALIC	1.46	5.27
LJPEG	0.68	5.58
GZIP	0.58	6.65

1499 1700

ว/ร.
ฬ 7987
2551

Start Encoding...

Picture Name: 130x92.bmp

Start timeDate: 8-5-2009--2:57:1.859

Stop timeDate: 8-5-2009--2:57:16.062

Original Size: 14900

Picture Size: 5960

จากผลการทดลอง จะเห็นได้ว่าค่า COMPRESSION RATIO ของ คาลิคมีค่าน้อยที่สุดในบรรดาโปรแกรมบีบอัดที่นำมาเปรียบเทียบแต่คาลิคจะใช้ระยะเวลาในการเข้ารหัสนานกว่าการบีบอัดชนิดอื่น โดยขนาดภาพก่อนการบีบอัดข้อมูลซึ่งมีขนาดไฟล์ที่จัดเก็บ 12900 ไบท์เมื่อผ่านการบีบอัดข้อมูลโดยใช้คาลิคได้ไฟล์บีบอัดข้อมูลภาพ 11800 ไบท์และทำการเปรียบเทียบภาพทั้งสองได้ค่า SNR และ PSNR เข้าสู่อนันต์ นั้นแสดงให้เห็นว่าการบีบอัดและการคลายการบีบอัดข้อมูลภาพขนาด 130x92มีลักษณะเหมือนกันทุกประการ

ผลการทดลองที่ 3

รูปที่4.3 ไฟล์รูปขนาด 270x103

ตารางที่ 4.3 ผลการทดลองที่ 3

ชนิดของโปรแกรมบีบอัดข้อมูล	Total time (S)	COMPRESSION RATIO
CALIC	3.774	4.934
LJPEG	1.328	5.379
GZIP	0.99	6.49

Start Encoding...

Picture Name: 270x103.bmp

Start timeDate: 8-5-2009--2:57:16.062

Stop timeDate: 8-5-2009--2:58:20.984

Original Size: 29900

Picture Size: 10810

จากผลการทดลอง จะเห็นได้ว่าค่า COMPRESSION RATIO ของ คาลิคมีค่าน้อยที่สุดในบรรดาโปรแกรมบีบอัดที่นำมาเปรียบเทียบแต่คาลิคจะใช้ระยะเวลาในการเข้ารหัสนานกว่าการบีบอัดชนิดอื่นโดยขนาดภาพก่อนการบีบอัดข้อมูลซึ่งมีขนาดไฟล์ที่จัดเก็บ 29900 ไบท์เมื่อผ่านการบีบอัดข้อมูลโดยใช้คาลิคได้ไฟล์บีบอัดข้อมูลภาพ 27810 ไบท์และทำการเปรียบเทียบภาพทั้งสองได้ค่า SNR และ PSNR เข้าสู่อนันต์ นั้นแสดงให้เห็นว่าการบีบอัดและการคลายการบีบอัดข้อมูลภาพขนาด 2270x103 มีลักษณะเหมือนกันทุกประการ

ผลการทดลองที่ 4

รูปที่4.4 ไฟล์รูปขนาด 555x366

ตารางที่ 4.4 ผลการทดลองที่ 4

ชนิดของโปรแกรมบีบอัดข้อมูล	Total time (S)	COMPRESSION RATIO
CALIC	12.611	4.62
LJPEG	3.9	5.11
GZIP	2.98	6.32

Start Encoding...

Picture Name: 1024x768.bmp

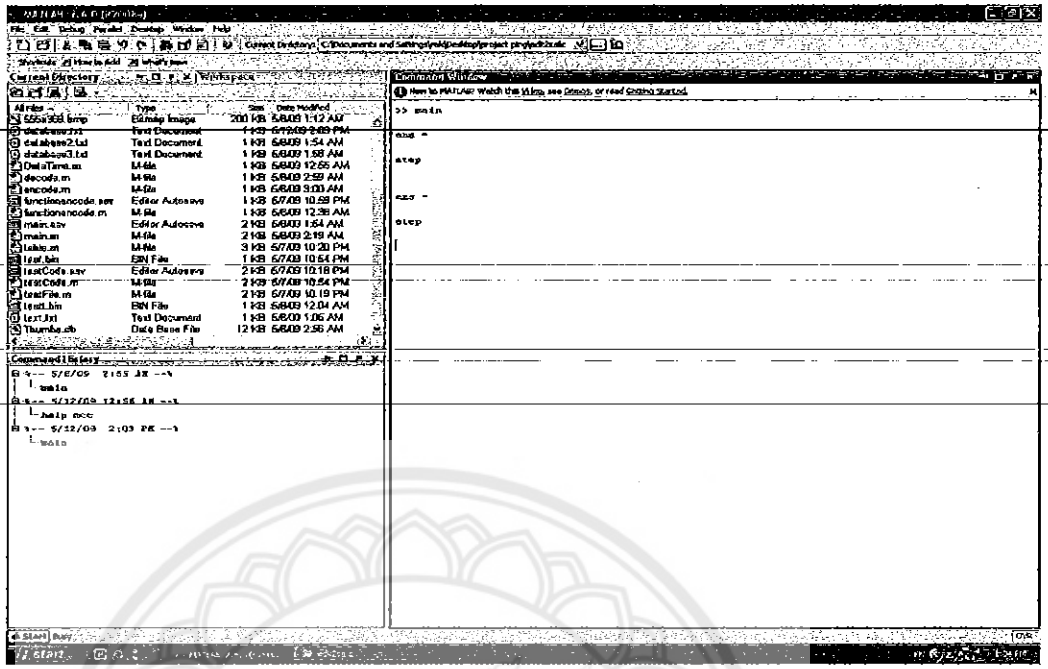
Start timeDate: 8-5-2009--3:53:4.234

Stop timeDate: 8-5-2009--15:45:4.843

Original Size: 204576

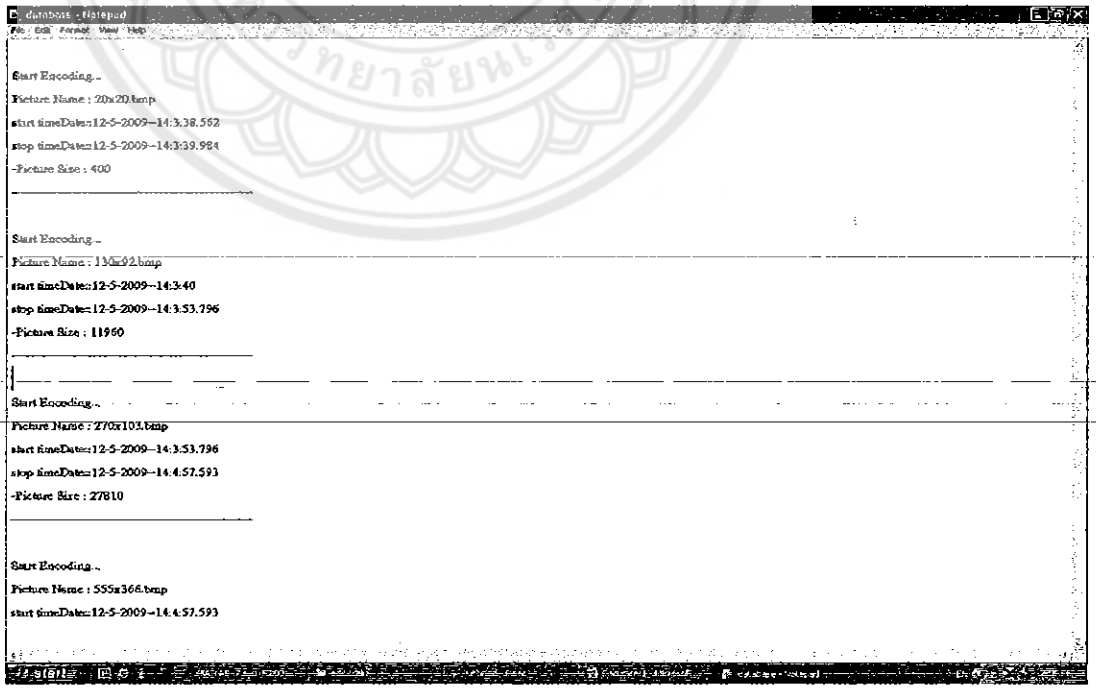
Picture Size: 86432

จากผลการทดลอง จะเห็นได้ว่าค่า COMPRESSION RATIO ของ คาลิดมีค่าน้อยที่สุดในบรรดาโปรแกรมบีบอัดที่นำมาเปรียบเทียบแต่คาลิดจะใช้ระยะเวลาในการเข้ารหัสนานกว่าการบีบอัดชนิดอื่น โดยขนาดภาพก่อนการบีบอัดข้อมูลซึ่งมีขนาดไฟล์ที่จัดเก็บ 29900 ไบท์เมื่อผ่านการบีบอัดข้อมูลโดยใช้คาลิดได้ไฟล์บีบอัดข้อมูลภาพ 27810 ไบท์และทำการเปรียบเทียบภาพทั้งสองได้ค่า SNR และ PSNR เข้าสู่อนันต์ นั้นแสดงให้เห็นว่าการบีบอัดและการคลายการบีบอัดข้อมูลภาพขนาด 2270x103 มีลักษณะเหมือนกันทุกประการ



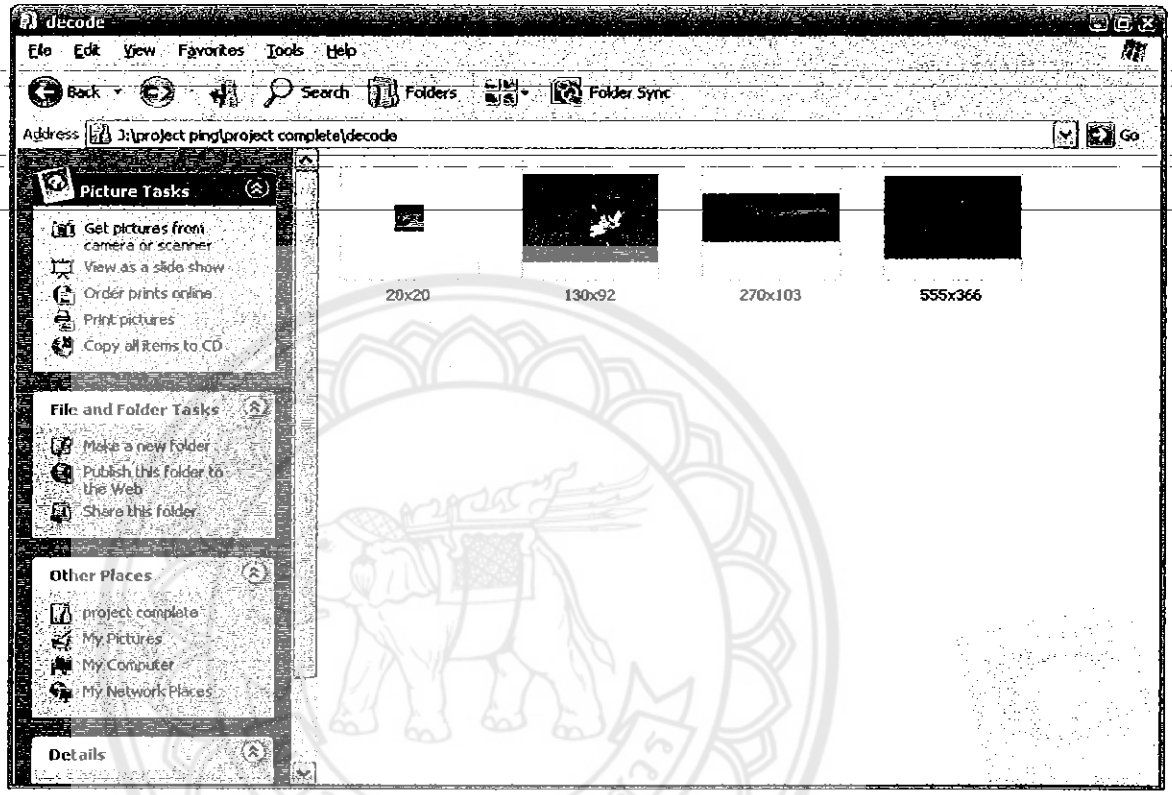
รูปที่ 4.5 หน้าตาของโปรแกรมการบีบอัดข้อมูลด้วยวิธีคาลิค

รูปที่ 4.5 หน้าตาของโปรแกรมการบีบอัดข้อมูลด้วยวิธีคาลิค โดยรันไฟล์ main.m ผ่านโปรแกรมแมทแลป



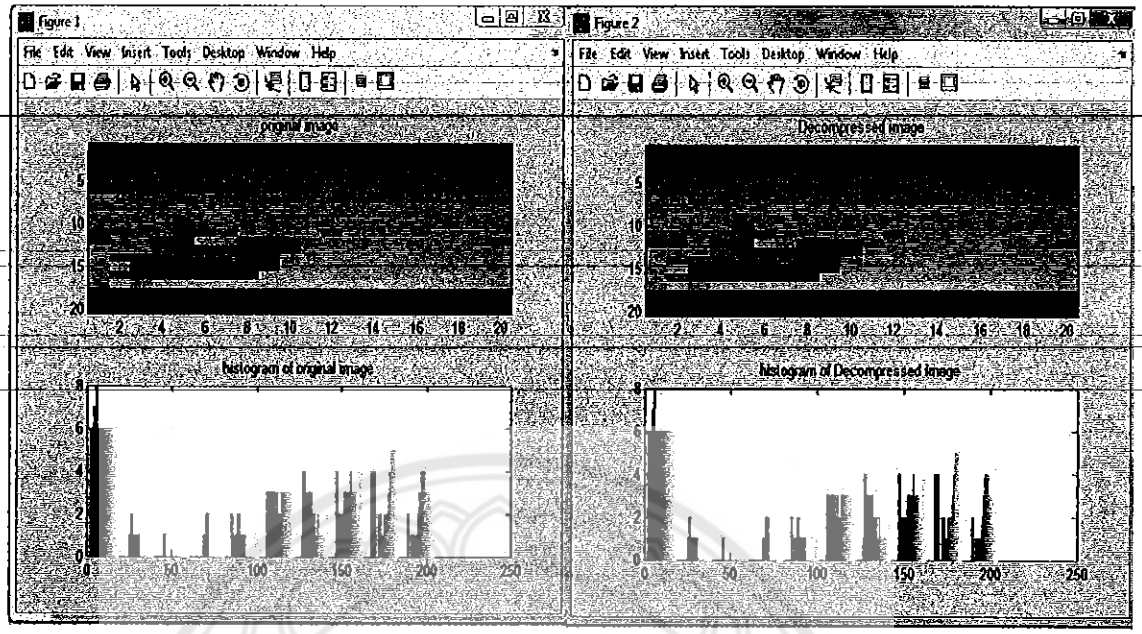
รูปที่ 4.6 ผลการรันโปรแกรมบีบอัดข้อมูลด้วยวิธีคาลิค

รูปที่ 4.6 ผลการรันโปรแกรมบีบอัดข้อมูลด้วยวิธีคาลิคจากรูปรูปที่ 4.5 จะแสดงผลการบีบอัดข้อมูลทั้ง 4 ภาพ ซึ่งจะโชว์ชื่อไฟล์ เวลาที่เริ่มในการบีบอัด เวลาที่ทำการบีบอัดข้อมูลเสร็จ และ ขนาดภาพหลังจากการถูกบีบอัดซึ่งข้อมูลที่ได้จะถูกเก็บเป็นไฟล์ .txt



รูปที่ 4.7 รูปที่ได้หลังการ Decode

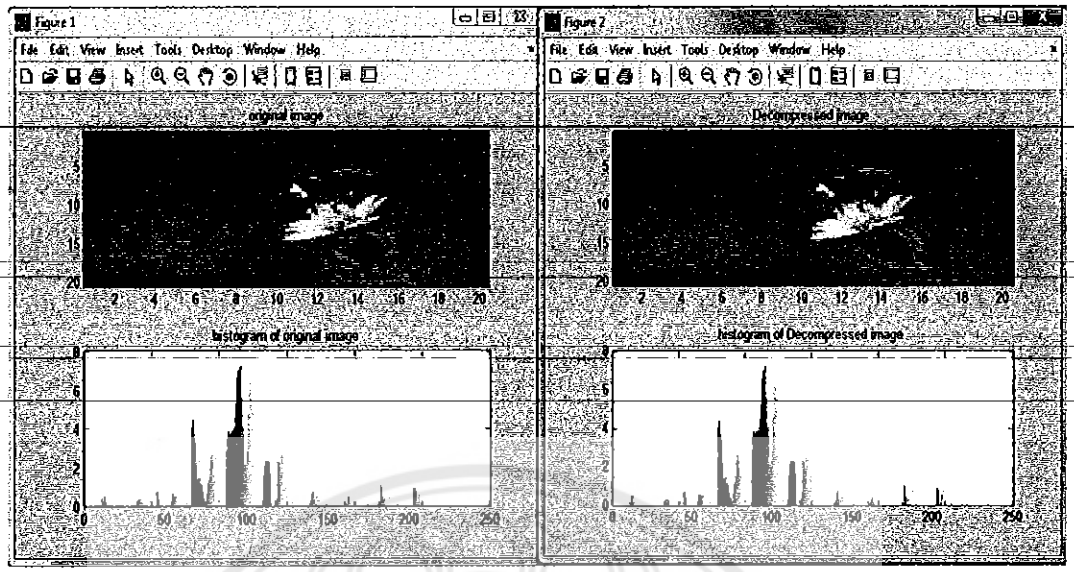
รูปที่ 4.7 รูปที่ได้หลังการ Decode จะได้ไฟล์รูปภาพออกมาเป็นไฟล์ .bmp



(ก) (ข)

รูปที่ 4.8 ฮิสโทแกรมของรูปขนาด 20*20

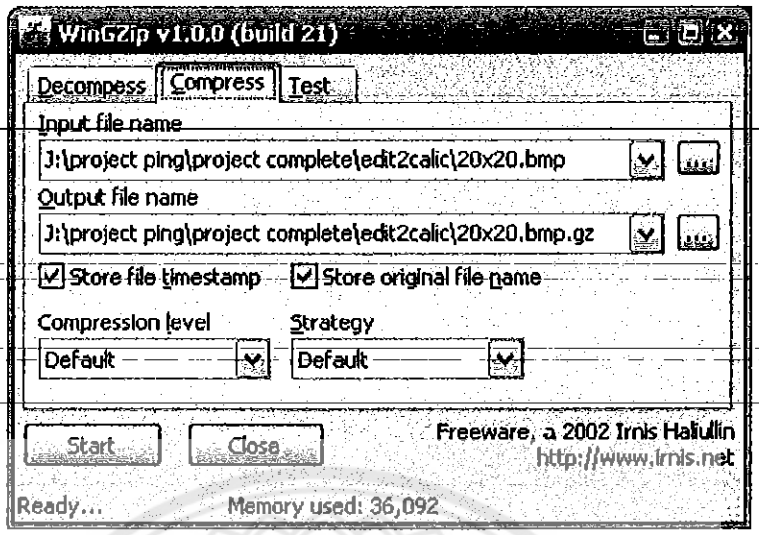
รูปที่ 4.8 เป็นฮิสโทแกรมของรูปขนาด 20*20 โดยในรูปที่ 4.8 (ก) รูปด้านบนจะเป็นรูปภาพที่ยังไม่ได้ผ่านการบีบอัดข้อมูลและด้านล่างจะเป็นรูปของฮิสโทแกรมของภาพที่ยังไม่ผ่านการบีบอัดข้อมูล ในรูปที่ 4.8 (ข) รูปด้านบนจะเป็นรูปภาพที่ผ่านการบีบอัดข้อมูลและได้ทำการคลายการบีบอัดข้อมูลภาพและด้านล่างจะเป็นรูปของฮิสโทแกรมของภาพที่ผ่านการบีบอัดข้อมูลและได้ทำการคลายการบีบอัดข้อมูลภาพเมื่อเปรียบเทียบจากฮิสโทแกรมทั้งสองภาพมีกราฟเหมือนกันทุกประการแสดงว่าภาพก่อนการบีบอัดข้อมูลและการคลายการบีบอัดข้อมูลมีคุณลักษณะเหมือนกันทุกประการ



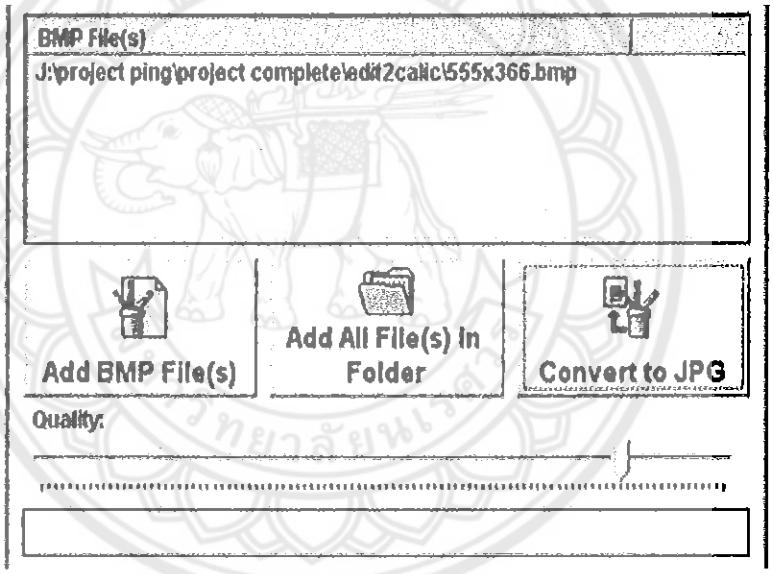
(ก) (ข)

รูปที่ 4.9 ฮิสโทแกรมของรูปขนาด 130*92

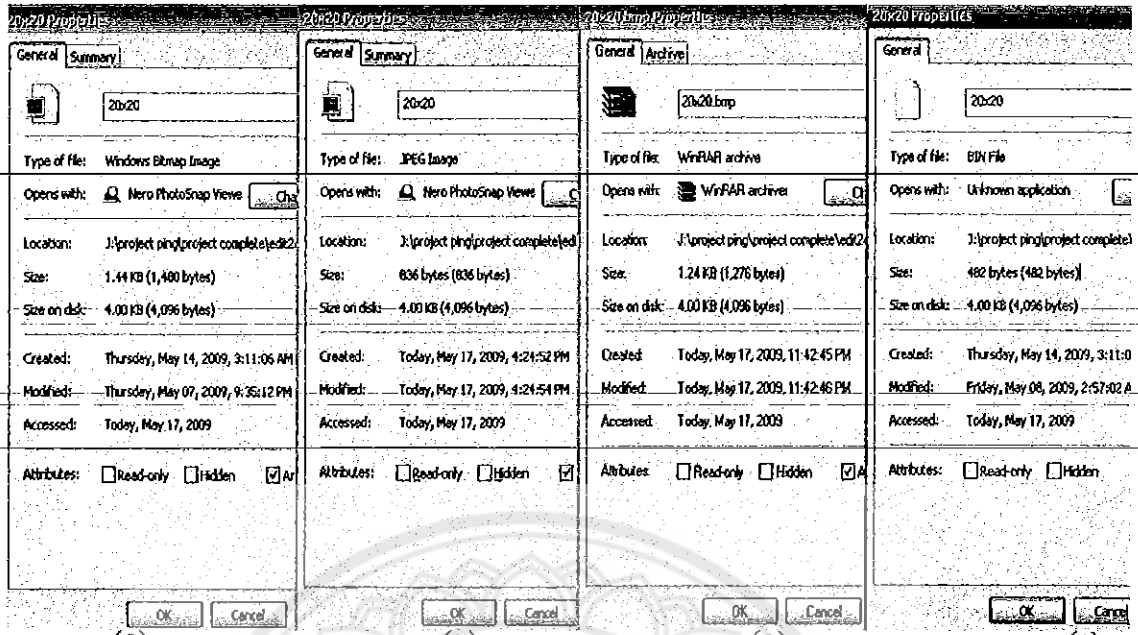
รูปที่ 4.9 เป็นฮิสโทแกรมของรูปขนาด 130*92 โดยในรูปที่ 4.9 (ก) รูปด้านบนจะเป็นรูปภาพที่ยังไม่ได้ผ่านการบีบอัดข้อมูลและด้านล่างจะเป็นรูปของฮิสโทแกรมของภาพที่ยังไม่ผ่านการบีบอัดข้อมูล ในรูปที่ 4.8 (ข) รูปด้านบนจะเป็นรูปภาพที่ผ่านการบีบอัดข้อมูลและ ได้ทำการคลายการบีบอัดข้อมูลภาพและด้านล่างจะเป็นรูปของฮิสโทแกรมของภาพที่ผ่านการบีบอัดข้อมูลและ ได้ทำการคลายการบีบอัดข้อมูลภาพเมื่อเปรียบเทียบจากฮิสโทแกรมทั้งสองภาพมีกราฟเหมือนกันทุกประการแสดงว่าภาพก่อนการบีบอัดข้อมูลและการคลายการบีบอัดข้อมูลมีคุณลักษณะเหมือนกันทุกประการ



รูปที่ 4.10 หน้าตาโปรแกรมบีบอัดแบบGzip



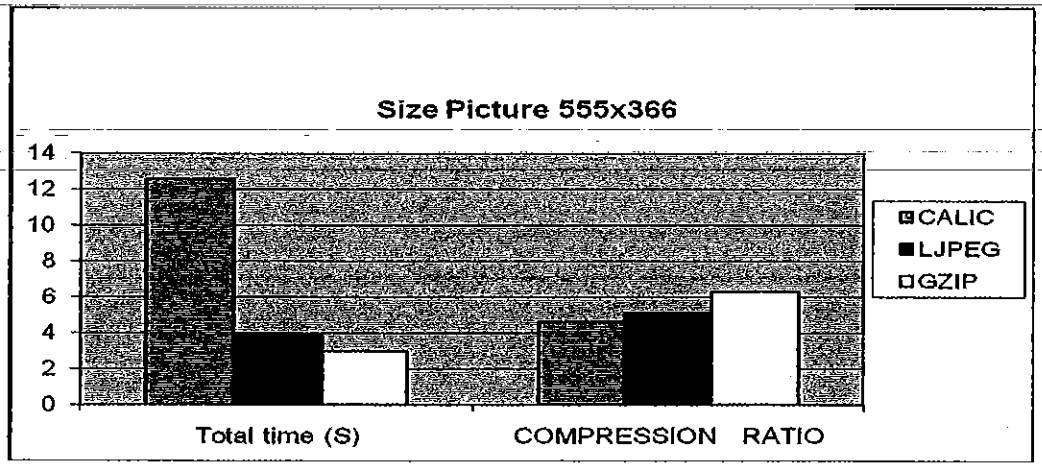
รูปที่ 4.11 หน้าตาโปรแกรมบีบอัดแบบ LJPEG



รูปที่ 4.12 รูปแสดงผลหลังการบีบอัดของทุกการบีบอัด

รูปที่ 4.12 รูปแสดงผลหลังการการบีบอัดของทุกการบีบอัด โดยรูป (ก) จะเป็นภาพต้นแบบจะเป็นภาพที่มีขนาดไฟล์ 1480 ไบท์ภาพที่ได้จะเป็นไฟล์.bmpรูป (ข) จะเป็นภาพที่ใช้โปรแกรม JPEC ในการบีบอัดภาพจะมีขนาดไฟล์ 836 ไบท์ภาพที่ได้จะเป็นไฟล์.JPEG (ค) จะเป็นภาพที่ใช้โปรแกรม GZIPในการบีบอัดภาพจะมีขนาดไฟล์ 1276ไบท์ภาพที่ได้จะเป็นไฟล์.GZIP (ง) จะเป็นภาพที่ใช้โปรแกรม calic ในการบีบอัดภาพจะมีขนาดไฟล์ 482 ไบท์ภาพที่ได้จะเป็นไฟล์.bin

4.5 สรุปผลการทดลอง

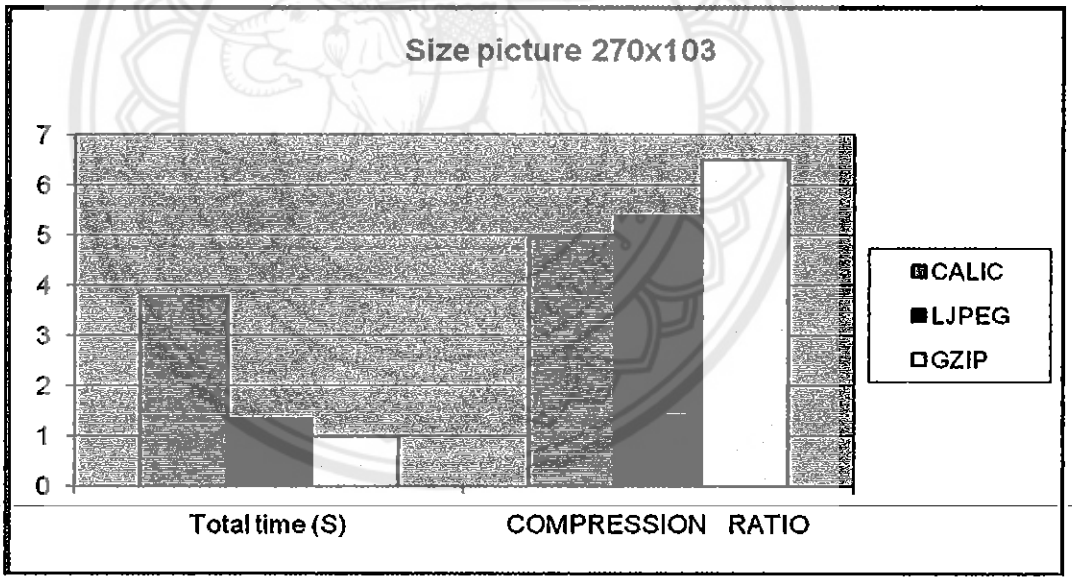


รูปที่ 4.13 กราฟแสดงค่าCRและเวลาของรูปขนาด555 x 366

ตารางที่4.5 ผลสรุปของรูป555x366

ชนิดของโปรแกรมบีบอัดข้อมูล	size picture	Total time (S)	COMPRESSION RATIO
CALIC	555x366	12.611	4.62
LJPEG	555x366	3.9	5.11
GZIP	555x366	2.98	6.32

จากรูปที่ 4.13 แสดงผลการเปรียบเทียบค่าCOMPRESSION RATIOและเวลา ระหว่างโปรแกรมบีบอัด CALIC LJPEG GZIP ของรูปขนาด 555x366 จากกราฟสรุปได้ว่า ค่า COMPRESSION RATIO ของคาลิคมีค่าน้อยที่สุดแต่ก็ใช้ระยะเวลามากที่สุดด้วย

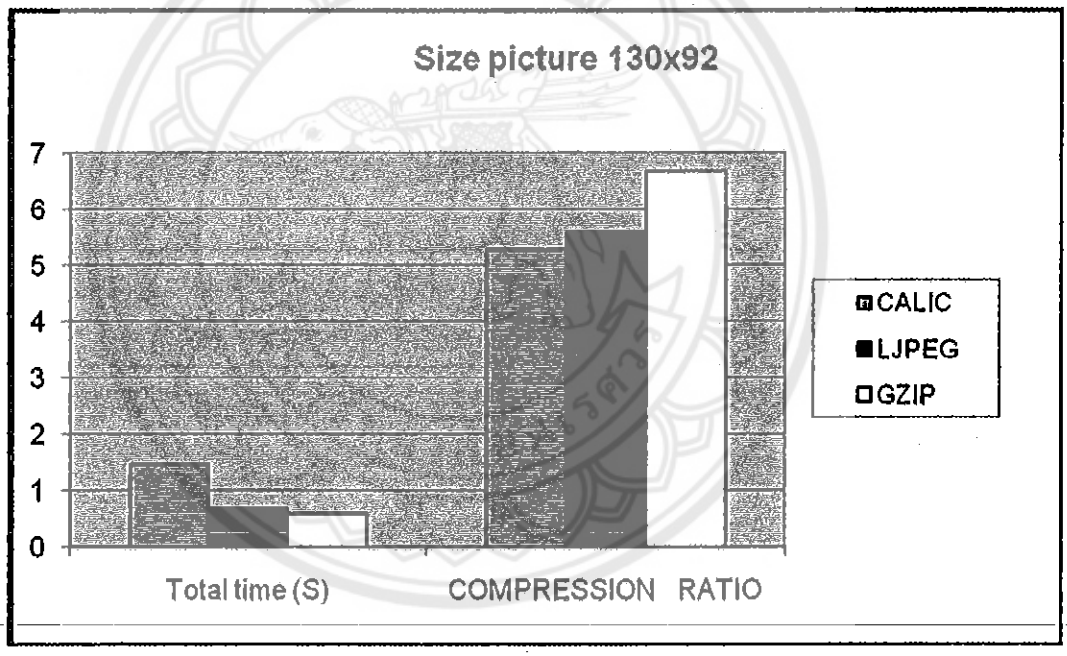


รูปที่ 4.14 กราฟแสดงค่าCRและเวลาของรูปขนาด270x103

ตารางที่4.6 ผลสรุปของรูป270x103

ชนิดของโปรแกรมบีบอัดข้อมูล	size picture	Total time (S)	COMPRESSION RATIO
CALIC	270x103	3.774	4.934
LJPEG	270x103	1.328	5.379
GZIP	270x103	0.99	6.49

จากรูปที่ 4.14 แสดงผลการเปรียบเทียบค่าCOMPRESSION RATIOและเวลา ระหว่างโปรแกรมบีบอัด CALIC LJPEG GZIP ของรูปขนาด 270x103 จากกราฟสรุปได้ว่า ค่า COMPRESSION RATIO ของคาลิคมีค่าน้อยที่สุดแต่ก็ใช้ระยะเวลามากที่สุดด้วย

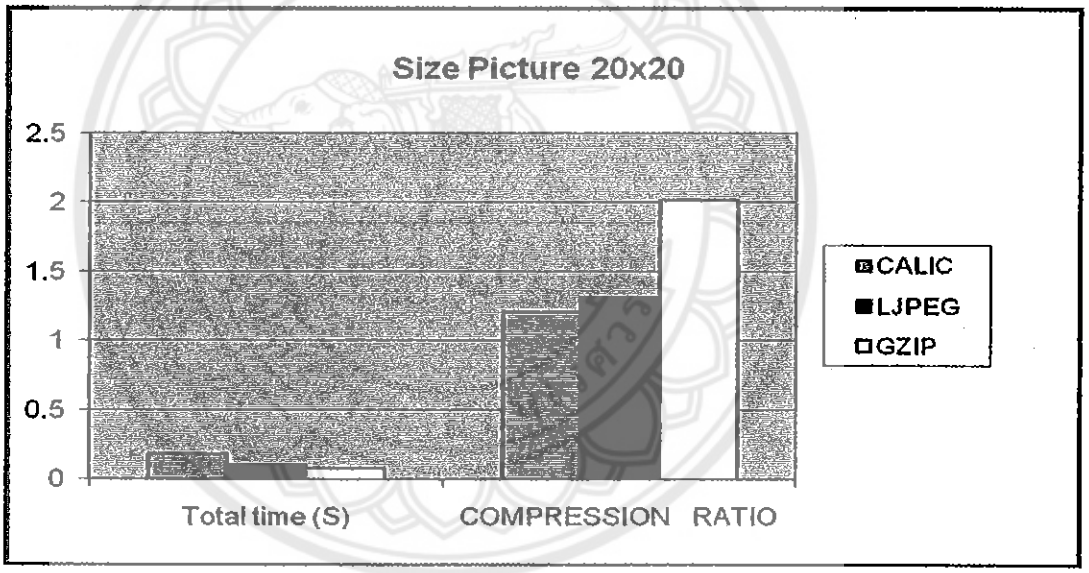


รูปที่ 4.15 กราฟแสดงค่าCRและเวลาของรูปขนาด130x92

ตารางที่4.7 ผลสรุปของรูป130x92

ชนิดของโปรแกรมบีบอัดข้อมูล	size picture	Total time (S)	COMPRESSION RATIO
CALIC	130x92	1.46	5.27
LJPEG	130x92	0.68	5.58
GZIP	130x92	0.58	6.65

รูปที่ 4.15 แสดงผลการเปรียบเทียบค่าCOMPRESSION RATIOและเวลา ระหว่างโปรแกรมบีบอัด CALIC LJPEG GZIP ของรูปขนาด 130x92 จากกราฟสรุปได้ว่า ค่า CR ของคาลิดมีค่าน้อยที่สุดแต่ก็ใช้ระยะเวลามากที่สุดด้วย



รูปที่ 4.16 กราฟแสดงค่าCRและเวลาของรูปขนาด20x20

ตารางที่4.8 ผลสรุปของรูป 20x20

ชนิดของโปรแกรมบีบอัดข้อมูล	size picture	Total time (S)	COMPRESSION RATIO
CALIC	20x20	0.18	1.205
LJPEG	20x20	0.1	1.308
GZIP	20x20	0.08	2.011

รูปที่ 4.16 แสดงผลการเปรียบเทียบค่าCOMPRESSION RATIOและเวลา ระหว่างโปรแกรมบีบอัด CALIC LJPEG GZIP ของรูปขนาด 20x20 จากกราฟสรุปได้ว่า ค่า COMPRESSION RATIO ของคาลิคมีค่าน้อยที่สุดแต่ก็ใช้ระยะเวลามากที่สุดด้วย

บทที่ 5
บทสรุป

5.1 สรุปผลการทดลอง

จากบทที่ 4 ผลการทดลองที่ได้ทำให้ทราบว่า การบีบอัดข้อมูลภาพแบบคาลิคสามารถบีบอัดข้อมูลภาพได้ ค่า compression ratio น้อยกว่า LJPEG และ GZIPประมาณ 0.1 เท่า แต่การบีบอัดข้อมูลแบบคาลิค จะใช้เวลาในการบีบอัดข้อมูลนานกว่า LJPEG และ GZIP ประมาณ 1เท่า

5.2 ปัญหาในการทดลองและแนวทางแก้ไข

ปัญหาที่เกิดขึ้น

ถ้าใช้ไฟล์ภาพขนาดใหญ่จะใช้เวลาในการรันโปรแกรมเวลานาน

แนวทางการแก้ไขปัญหา

เนื่องจากการเข้าบีบอัดข้อมูลภาพนั้นจะทำทีละตัว ทำให้ใช้เวลาในการรันโปรแกรม ดังนั้นจึงมีการกำหนดขนาดภาพให้เล็กลงเพื่อจะใช้เวลาในการรันโปรแกรมน้อยลง โดยกำหนดให้ขนาดอยู่ที่ประมาณ 20x20 pixels

5.3 แนวทางการพัฒนาในอนาคต

เนื่องจากการบีบอัดข้อมูลแบบคาลิคบีบอัดได้เฉพาะข้อมูลภาพ แต่ในการบีบอัดมีข้อมูลชนิดอื่นที่ไม่ใช่เฉพาะข้อมูลภาพอย่างเดียว ดังนั้นควรที่จะเขียนโปรแกรมที่สามารถบีบอัดได้ทั้งข้อมูลภาพและข้อมูลชนิดอื่นด้วย

เอกสารอ้างอิง

[1] มนัสชัย โรจนวิเชียร, อรอุมา อักโข, "การบีบอัดข้อมูลด้วยวิธีฮัฟฟ์แมน" ปริญญานิพนธ์มหาบัณฑิต สาขาวิศวกรรมไฟฟ้าและคอมพิวเตอร์. มหาวิทยาลัยนเรศวร. 2006

[2] Hao Hu. "A Study of CALIC" A paper submitted to the Computer Science & Electrical Engineering Department in partial ful_llment of the requirements for the M.S. degree at University of Maryland Baltimore County. 2004

[3] รศ.ดร.มนัส สังวรศิลป์ .คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์. กรุงเทพมหานคร : อินโฟเพรส 2543.

ประวัติผู้เขียนโครงงาน

ชื่อ นายพีระพงษ์ อุดมตระกูลวงศ์

ภูมิลำเนา 139 ถ.แสงสวรรค์ ต.ชุมแสง อ.ชุมแสง จ.นครสวรรค์ 60120

ประวัติการศึกษา

- จบการศึกษาระดับมัธยมศึกษาจากโรงเรียนชุมแสงชนูทิศ
- ปัจจุบันกำลังในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E – mail : p_maleaw@hotmail.com

ชื่อ นายเฉลิมเกียรติ รอดเกตุ

ภูมิลำเนา 732 หมู่ 16 ต.หนองไผ่ อ.หนองไผ่ จ.เพชรบูรณ์ 67140

ประวัติการศึกษา

- จบการศึกษาระดับมัธยมศึกษาจากโรงเรียนเซนต์โยเซฟศรีเพรชบูรณ์
- ปัจจุบันกำลังในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E – mail : phycho_am@hotmail.com