สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาและทดสอบอัลกอริทึมสำหรับการลดขนาดข้อมูลรูปภาพ

โดย

นายชูสม	กุลานุจารี	รหัสประจำตัว 33501008
นางสาวสายหิม	ชิ้มสกุล	รหัสประจำตัว 33501034
นางสาวสิริพร	หงษ์ทอง	รหัสประจำตัว 33501036

วพ.
ว689ก
2536
เลขหมู่.....
เลขทะเบียน.....
วัน,เดือน,ปี.....
612558199

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต

สาขาคณิตศาสตร์ประยุกต์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2536

ปัญหาพิเศษ

เรื่อง

การพัฒนาและทดสอบอัลกอริทึมสำหรับการลดขนาดข้อมูลรูปภาพ

โดย

นายชูสม	กุลานุจารี	รหัสประจำตัว 33501008
นางสาวสายพิณ	ชื้มสกุล	รหัสประจำตัว 33501034
นางสาวสิริพร	หงษ์ทอง	รหัสประจำตัว 33501036

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต

สาขาคณิตศาสตร์ประยุกต์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2536

A SPECIAL PROJECT

OF

IMPLEMENTATION AND TESTING ALGORITHMS FOR IMAGE COMPRESSION

BY

MR. CHUSOM	KULANUCHAREE	ID. 33501008
MISS SAIPHIN	SIMSAKUL	ID. 33501034
MISS SIRIPORN	HONGTHONG	ID. 33501036

A Special Project Submitted in Partial Fulfillment

of the Requirement for the Degree of bachelor of Science

Major of Applied Mathematics

Department of Mathematics and Computer Science

Faculty of Science

King Mongkut's Institute of Technology Ladkrabang

1993

ปัญหาพิเศษเรื่อง การพัฒนาและทดสอบอัลกอริทึมสำหรับการลดขนาดข้อมูลรูปภาพ

(Implementation and testing algorithms for image compression)

ชื่อนักศึกษา นายชูสม กุลานุจารี รหัสประจำตัว 33501008

นางสาวสายทิพ ชื้มสกุล รหัสประจำตัว 33501034

นางสาวสิริพร หงษ์ทอง รหัสประจำตัว 33501036

ชื่ออาจารย์ที่ปรึกษา อาจารย์ ศรัณย์ อินทโกสุม

อาจารย์ วีระ บุญจริง

ปัญหาพิเศษฉบับนี้ กรรมการสอบปัญหาพิเศษ ได้ตรวจพิจารณาแล้ว เห็นชอบแล้ว จึงอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตร์บัณฑิต สาขาคณิตศาสตร์ ประยุกต์ ประจำปีการศึกษา 2536

(รศ.ดร. ไมตรี โพธิ์สุข)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

(ผศ. ผ่องพรรณ รัตนธนาวันต์) ประธานกรรมการสอบปัญหาพิเศษ

(อาจารย์ วีระชัย ตันยะสิทธิ์) กรรมการสอบปัญหาพิเศษ

(อาจารย์ ศรัณย์ อินทโกสุม) อาจารย์ที่ปรึกษาปัญหาพิเศษ

(อาจารย์ วีระ บุญจริง) อาจารย์ที่ปรึกษาปัญหาพิเศษ

ปัญหาพิเศษเรื่อง การพัฒนาและทดสอบอัลกอริทึมสำหรับการลดขนาดข้อมูลรูปภาพ
(Implementation and testing algorithms for image compression)

ชื่อนักศึกษา นายชูสม กุลานุจารี รหัสประจำตัว 33501008
นางสาวสายพิณ ชิ้มสกุล รหัสประจำตัว 33501034
นางสาวสิริพร หงษ์ทอง รหัสประจำตัว 33501036

ชื่ออาจารย์ที่ปรึกษา อาจารย์ ศรัณย์ อินทโกสุม
อาจารย์ วีระ บุญจริง

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์

ปีการศึกษา 2536

บทคัดย่อ

ปัญหาพิเศษฉบับนี้ เสนอ การพัฒนาและทดสอบอัลกอริทึมสำหรับการลดขนาดข้อมูลรูปภาพโดยใช้อัลกอริทึมทั้งหมด 4 ชนิด คือ รันเลงท์, ฮัฟฟ์แมน, แอล แซด ดับบลิว และ บี ที ซี เมื่อทำการศึกษาลักษณะการจัดเก็บข้อมูลที่เป็นภาพของอัลกอริทึมทั้ง 4 ชนิดแล้ว จะทำการทดสอบ และเปรียบเทียบประสิทธิภาพของอัลกอริทึมในด้าน ความเร็วของการลดขนาดข้อมูล, เปอร์เซ็นต์ของการลดขนาดข้อมูล และ คุณภาพของภาพที่ได้หลังจากการลดขนาดข้อมูล ซึ่งผลการเปรียบเทียบพบว่า ถ้าพิจารณาด้านความเร็วควรใช้อัลกอริทึมของ รันเลงท์, ถ้าพิจารณาเปอร์เซ็นต์ของการลดขนาดข้อมูลควรใช้อัลกอริทึมของ แอล แซด ดับบลิว และ ถ้าพิจารณาด้านคุณภาพของภาพที่ได้หลังจากการลดขนาดข้อมูลจะใช้อัลกอริทึมของรันเลงท์ หรือ ฮัฟฟ์แมน หรือแอล แซด ดับบลิว ก็ได้เพราะคุณภาพของภาพก่อนและหลังการลดขนาดข้อมูลจะไม่แตกต่างกัน

Special Project Title	Implementation and testing algorithms for image compression
Name	Mr. Chusom Kulanujaree Id.33501008
	Miss Saiphin Simsakul Id.33501034
	Miss Siriporn Hongthong Id.33501036
Adviser	Mr.Sarun Intakosoom
	Mr.Weera Boonjing
Department	Mathematics and Computer Science
Academic Year	1993

Abstract

This special project is implementation and testing algorithms for image compression. First, we selected 4 algorithms (Run-Length, Huffman, LZW and BTC) and studied about the arrangement of image. Then each algorithm is implemented and tested for image compression. Next we compered each algorithm about the speed, percent of image compression and the quantity of image after decompression.

Finally, we found that Run-Length has the lowest speed, LZW has the lowest percent of image compression and Run-Length, Huffman and LZW have the same quantity as the initial image.

กิตติกรรมประกาศ

ปัญหาพิเศษนี้สำเร็จได้เพราะความช่วยเหลือของบุคคลต่าง ๆ เหล่านี้

1. อาจารย์ ศรัณย์ อินทโกสุม

ท่านช่วยแนะนำวิธีที่ใช้ในการลดขนาดข้อมูล, ช่วยให้แนวคิดในการเขียนโปรแกรม รวมทั้งให้คำปรึกษา และช่วยหาเอกสารต่าง ๆ ที่เกี่ยวข้องกับการทำปัญหาพิเศษ

2. อาจารย์ วีระ บุญจริง

ท่านช่วยแนะนำวิธีการจัดทำเอกสาร รวมทั้งให้คำปรึกษาที่เกี่ยวกับปัญหาพิเศษ

ขอขอบคุณในความกรุณาเป็นอย่างสูง

นายชูสม กุลานุจารี
นางสาวสายหิณ ชื้มสกุล
นางสาวสิริพร หงษ์ทอง

ผู้ทำปัญหาพิเศษ

สารบัญรูป

หน้า

รูปที่ 2.1 การลดขนาดข้อมูลและการขยายกลับ 3
รูปที่ 2.2 ทรีของฮัฟฟ์แมนในการลดขนาดข้อมูล 7
รูปที่ 2.3 ทรีของฮัฟฟ์แมนในการขยายกลับข้อมูล 8
รูปที่ 4.1 ภาพที่มีลักษณะเป็นแถบสีหลายสี 21
รูปที่ 4.2 ภาพที่มีสีเดียวกันอย่างต่อเนื่อง 22
รูปที่ 4.3 ภาพที่มีสีซ้ำกับหลายที่แต่ไม่ต่อเนื่อง 23

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญรูป ..... ง

บทที่ 1 บทนำ 1

ความสำคัญและที่มาของปัญหา 1

วัตถุประสงค์ของปัญหา 1

ขอบเขตของปัญหา 1

ขั้นตอนในการดำเนินงาน 1

ประโยชน์ที่คาดว่าจะได้รับ 2

บทที่ 2 ทฤษฎีการลดขนาดข้อมูล 3

แนวคิดในการลดขนาดข้อมูล 3

การลดขนาดข้อมูลโดยใช้วิธีรัน เลงท์ (Run-Length) 4

การลดขนาดข้อมูลโดยใช้วิธีฮัฟฟ์แมน (Huffman) 5

การลดขนาดข้อมูลโดยใช้วิธีแอล แซด ดับบลิว (LZW) 9

การลดขนาดข้อมูลโดยใช้วิธี บี ที ซี (BTC) 13

บทที่ 3 การพัฒนาโปรแกรมเพื่อทดสอบอัลกอริทึม 16

วิธีการและขั้นตอนการพัฒนาโปรแกรม 16

วิธีการทดสอบอัลกอริทึม 17

หน้า

บทที่ 4 การเปรียบเทียบประสิทธิภาพของอัลกอริทึม 19

เปอร์เซ็นต์และเวลาในการลดขนาดข้อมูล 19

คุณภาพของข้อมูล 24

บทที่ 5 สรุป และ ข้อเสนอแนะ 25

สรุป 25

ข้อเสนอแนะ 26

เอกสารอ้างอิง

บทที่ 1

บทนำ

ความสำคัญและที่มาของปัญหา

ในปัจจุบันคอมพิวเตอร์ฮาร์ดแวร์ ได้มีการพัฒนาก้าวหน้าไปอย่างมาก ทั้งในด้านความสามารถ ความเร็ว และ ความจุของหน่วยความจำ สำหรับด้านซอฟท์แวร์ได้มีการพัฒนาให้ผู้ใช้ ใช้งานได้โดยง่ายขึ้น โดยนำเอาเทคนิคที่เรียกว่ากุย(GUI :Graphic User Interface) เทคนิคนี้เป็นการนำภาพขึ้นมาแทนตัวอักษรเพื่อติดต่อกับผู้ใช้ แต่ทั้งนี้ภาพต่างๆ นั้นต้องใช้หน่วยความจำเพื่อใช้ในการเก็บข้อมูลเป็นจำนวนมาก ได้มีนักคอมพิวเตอร์และนักคณิตศาสตร์จำนวนมาก ได้คิดวิธีการที่จะเก็บภาพเหล่านั้นโดยใช้หน่วยความจำน้อยที่สุด โดยมีวิธีการต่างๆ กันหลายวิธี แต่ละวิธีใช้ได้อย่างเหมาะสมกับข้อมูลที่มีลักษณะต่างๆ กันจึงต้องศึกษาว่าข้อมูลภาพแต่ละภาพนั้นเหมาะสมที่จะใช้วิธีการใดในการเก็บภาพ จึงจะได้ขนาดของข้อมูลรูปภาพที่ใช้เนื้อที่หน่วยความจำน้อยที่สุด และ ใช้เวลาน้อยที่สุด

วัตถุประสงค์ของปัญหา

เพื่อศึกษา เปรียบเทียบอัลกอริทึม (algorithm) ที่ใช้ในการลดขนาดข้อมูลภาพและวิเคราะห์เพื่อหาอัลกอริทึมที่เหมาะสมกับภาพแต่ละประเภท

ขอบเขตของปัญหา

การศึกษานี้จะศึกษา เปรียบเทียบ อัลกอริทึม จำกัดอยู่เฉพาะวิธีต่อไปนี้คือ
รันเลงท์เอ็นโคดเดอร์ (Run-Length Encoder), ฮัฟฟ์แมน (Huffman),
แอล แซด ดับบลิว (LZW) และ บี ที ซี (BTC)

ขั้นตอนในการดำเนินงาน

1. ศึกษาทฤษฎีการลดขนาดข้อมูล
2. เขียนโปรแกรมทดสอบอัลกอริทึมต่างๆ โดยมีตัวแปร คือ ความเร็ว, ขนาดข้อมูล,

คุณภาพของรูปภาพหลังการคืนข้อมูลเดิม เป็นต้น

3. เปรียบเทียบความสามารถในการทำงานของอัลกอริทึมชนิดต่างๆ

4. สรุปความสามารถของอัลกอริทึมชนิดต่างๆ ว่าเหมาะสมกับข้อมูลภาพแบบใด

ประโยชน์ที่คาดว่าจะได้รับ

1. ช่วยเพิ่มประสิทธิภาพในการเก็บข้อมูลรูปภาพ

2. ทำให้สามารถเลือกอัลกอริทึมในการลดขนาดข้อมูลรูปภาพที่เหมาะสมกับภาพแต่ละประเภทได้

3. เป็นแนวทางในการศึกษาและพัฒนาอัลกอริทึมเพื่อลดขนาดข้อมูลรูปภาพต่อไป

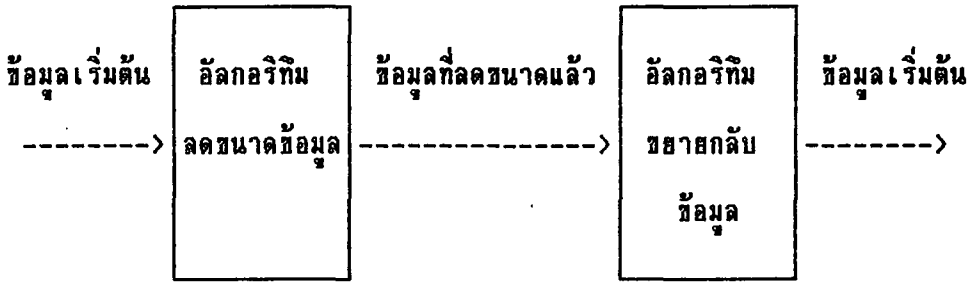
บทที่ 2

ทฤษฎีการลดขนาดข้อมูล

แนวคิดในการลดขนาดข้อมูล

การลดขนาดข้อมูล เป็นการทำให้ข้อมูลมีขนาดเล็กลง เพื่อประโยชน์คือการใช้เนื้อที่ในการจัดเก็บน้อยลง การลดเวลาเข้าถึงข้อมูล และ สามารถประมวลผลได้เร็วขึ้น แต่อย่างไรก็ตามการนำข้อมูลกลับมาใช้งานต้องมีการขยายกลับ เพื่อให้ได้ข้อมูลที่มีสาระครบถ้วนตามข้อมูลเริ่มต้น

การลดขนาดข้อมูล และ การขยายกลับนั้นโดยทั่วไปทำโดยนำข้อมูลมาผ่านอัลกอริทึม ดังแสดงในรูปที่ 2.1 เทคนิคการลด และ การขยายกลับจะเป็นไปตามอัลกอริทึม ซึ่งในที่นี้จะกล่าวถึงอัลกอริทึมสำหรับภาพคือ วิธีรันเลงท์ (Run-Length), วิธีฮัฟฟ์แมน (Huffman), วิธีแอล แซด ดับบลิว (LZW) และ วิธีบี ที ซี (BTC)



รูปที่ 2.1 การลดขนาดข้อมูลและการขยายกลับ

การลดขนาดข้อมูลโดยวิธีรัน เลงท์ (Run-Length)

อัลกอริทึมการลดขนาดข้อมูล

การลดขนาดข้อมูลโดยวิธีนี้มีอัลกอริทึมดังนี้

1. อ่านค่าจากไฟล์ ถ้าค่าที่อ่านมามีการซ้ำติด ๆ กัน ก็จะนำค่าจำนวนที่ซ้ำนั้นมา or กับ c0H แล้วเก็บลงไฟล์ ตามด้วยค่าที่มีการซ้ำ เช่น 01 01 01 จะได้เป็น c3 01 แต่จำนวนซ้ำจะต้องไม่เกิน 63 ไบต์ ถ้าเกิน จะเก็บค่าซ้ำที่ค่า ffH (3fH or c0H) ลงไปแล้วตามด้วยค่าของข้อมูล จากนั้นเริ่มต้นนับ 1 ใหม่

2. ถ้าค่าที่อ่านมาไม่ซ้ำกันเลยให้เปรียบเทียบค่านั้นกับ c0H

   ถ้าน้อยกว่า c0H ให้เก็บลงไฟล์ไปเลย

   ถ้ามากกว่าหรือเท่ากับ c0H ให้ เก็บค่า c1 แล้วตามด้วยค่าของข้อมูล

ตัวอย่างที่ 2.1 ถ้าข้อมูลเข้ามีลักษณะดังนี้ คือ 01 01 01 05 c0 c0 c0 c7 จะได้ ข้อมูลออกดังนี้ คือ c3 01 05 c3 c0 c1 c7

อัลกอริทึมการขยายกลับข้อมูล

การขยายกลับข้อมูลโดยวิธีนี้มีอัลกอริทึมดังนี้

1. อ่านข้อมูลมาทีละ 1 ไบต์ แล้วเปรียบเทียบกับ c0H

   ถ้ามากกว่า (คือ นำมา and กับ c0H แล้วได้ c0H) ให้ and กับ 3fH

   จะได้ค่าที่ซ้ำออกมา และ ข้อมูลที่ตามมาก็ คือ ข้อมูลที่จะต้องขยายออกมา ซ้ำ ๆ กันตามจำนานที่ได้ในตอนแรก

   ถ้าน้อยกว่า ก็จะเขียนค่านั้นลงในบัฟเฟอร์

2. ทำจนกระทั่งจบไฟล์ก็จะได้ข้อมูลทั้งหมดคืนมา

ตัวอย่างที่ 2.2 ถ้าข้อมูลเข้ามีลักษณะดังนี้ คือ c3 01 05 c3 c0 c1 c7 จะได้ ข้อมูลออกดังนี้ คือ 01 01 01 05 c0 c0 c0 c7

การลดขนาดข้อมูลโดยวิธีฮัฟฟ์แมน (Huffman-Compression)

แนวคิด

วิธีการลดขนาดข้อมูลรูปภาพแบบฮัฟฟ์แมน มีอยู่ 2 วิธีคือ แบบไม่ยืดหยุ่น (static) และแบบยืดหยุ่น (dynamic) สำหรับการทำงานแบบไม่ยืดหยุ่นนี้ การเข้ารหัสจะใช้ตารางเก็บค่าสถิติ ซึ่งเป็นตารางที่บอกว่ามี "ความถี่" ของจุดสีแต่ละจุดอยู่เท่าไหร่ ซึ่งเราจะต้องทำการสแกนอ่านไฟล์ก่อน 1 รอบ จึงจะได้ตารางเก็บค่าสถิตินี้มา จากนั้นจึงค่อยทำการลดขนาดข้อมูลรูปภาพในครั้งต่อไป โดยเราจะนำตารางเก็บค่าสถิติมาใช้เป็นหลักเกณฑ์ในการเข้ารหัส หรือถอดรหัส โดยตัวลดขนาดข้อมูลรูปภาพ (compressor) และขยายขนาดข้อมูลรูปภาพ (decompress) จะทำหน้าที่สร้าง "ทรี" หรือต้นไม้สำหรับตารางค่าสถิตินี้อีกครั้งหนึ่ง สำหรับแบบยืดหยุ่นจะเป็นการลดขนาดข้อมูลรูปภาพชนิดอิสระ (on the fly) คือจะสามารถสร้าง "ทรี" หรือต้นไม้ของฮัฟฟ์แมนในขณะที่อ่าน แล้วทำการลดขนาดข้อมูลรูปภาพได้ทันที ในที่นี้เราจะกล่าวถึงเฉพาะการลดขนาดฮัฟฟ์แมนแบบไม่ยืดหยุ่น

อัลกอริทึมการลดขนาดข้อมูล

1. ทำการสแกนอ่านไฟล์ครั้งที่ 1 และวิเคราะห์ไฟล์ข้อมูลรูปภาพที่จะทำการลดขนาดข้อมูล

2. ใช้อัลกอริทึมในการสร้าง "อาร์เรย์" (array) สำหรับเก็บค่าความถี่ของจุดสีแต่ละจุด

3. สร้างทรีของฮัฟฟ์แมนจากอาร์เรย์ที่เก็บค่าความถี่ โดยสร้างจากโหนดย่อยไปโหนดหลัก จุดประสงค์ในการสร้างก็คือ สร้างความสัมพันธ์ระหว่างจุดสีแต่ละจุดที่ได้อ่านเข้ามากับค่าบิตสตริงใหม่ ซึ่งจะใช้เป็นรหัสของจุดสีใช้แทนจุดสีแต่ละจุดโดยจุดสีที่มีค่าความถี่มากก็จะแทนด้วยบิตสตริงที่มีจำนวนบิตน้อย ๆ และจุดสีที่มีค่าความถี่น้อยก็จะแทนด้วยบิตสตริงที่มีจำนวนบิตมาก

4. ทำการสแกนอ่านไฟล์ครั้งที่ 2 แล้วแทนจุดสีซึ่งมีขนาด 8 บิต/จุดสี ด้วยค่าบิตสตริงใหม่ที่ได้จากทรี ซึ่งขนาดบิตของจุดสีแต่ละจุดจะไม่เท่ากัน (ขึ้นอยู่กับความถี่ของจุดสี แต่ละจุด)

5. ทำการเขียนไฟล์ลงในดิสก์

อัลกอริทึมในการสร้างทรีสำหรับการลดขนาดข้อมูลรูปภาพ

จะสร้างทรีจากโหนดย่อยไปโหนดหลัก จากตารางเก็บค่าความถี่ของจุดสีให้แต่ละโหนดแทนจุดสี ดังนั้นแต่ละโหนดจะมีค่าความถี่ตามตารางที่เก็บ โดยทำการสร้างทรีดังนี้

1. เลือกโหนดอิสระ 2 โหนดที่มีค่าความถี่ต่ำที่สุด

2. โหนดหลัก (parent node) ที่ได้จาก 2 โหนดนี้จะถูกสร้างขึ้นให้มีความถี่เท่ากับผลรวมของค่าความถี่ของ 2 โหนดย่อย (child node)

3. โหนดหลักนี้จะถูกเพิ่มเข้าไปในรายการของโหนดอิสระ และ 2 โหนดย่อยจะถูกย้ายออกจากรายการของโหนดอิสระ

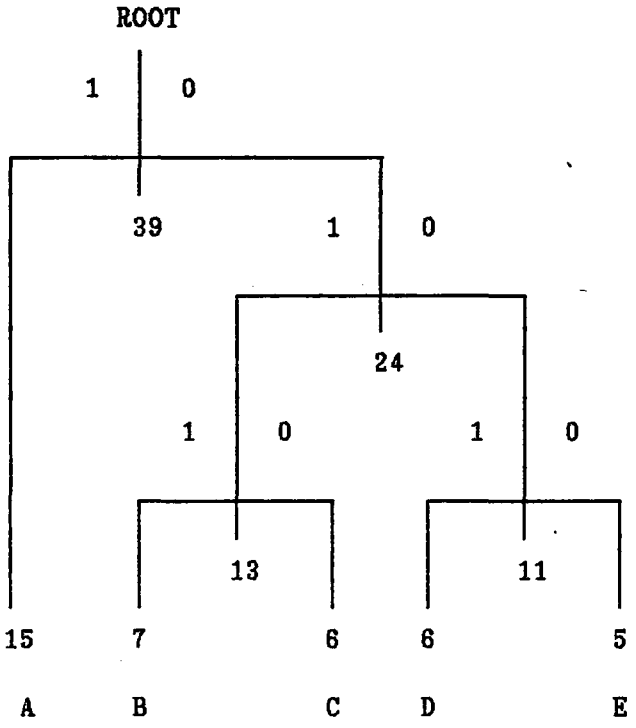
4. กำหนดให้โหนดย่อยซ้ายมือมีค่าบิตสตริงเป็น 1 และ โหนดย่อยขวามือมีค่าบิตสตริงเป็น 0

5. กลับไปทำขั้นตอนที่ 1 ใหม่ ทำเช่นนี้จนกระทั่งเหลือโหนดอิสระเพียง 1โหนด ซึ่งโหนดนี้ก็คือราก (root) ของทรีนั่นเอง

ตัวอย่างที่ 2.3 สมมติให้อักษรภาษาอังกฤษแทนแต่ละจุดสี ซึ่งมีค่าความถี่ดังนี้ จงทำการลดขนาดข้อมูลรูปภาพ

ความถี่	:	15	7	6	6	5
จุดสี	:	A	B	C	D	E

ทำการสร้างทรีของฮัฟฟ์แมนจากโหนดย่อยไปโหนดหลัก จะได้ทรีดังรูปที่ 2.2



รูปที่ 2.2 ทรีของฮัฟฟ์แมนในการลดขนาดข้อมูล

แทนค่าบิตสตริงให้กับจุดสีเหล่านี้ โดยเริ่มจากโหนดย่อย (leaf node) ไปที่ราก (root) ของฮัฟฟ์แมนทรี ซึ่งเราจะพุช (push) บิตไปที่สแตค (stack) แล้วทำการป๊อป (pop) บิตเหล่านี้ขึ้นมา ก็จะได้ค่าบิตสตริงของจุดสีเหล่านี้ คือ

A มีค่าบิตสตริงเท่ากับ 1

B มีค่าบิตสตริงเท่ากับ 011

C มีค่าบิตสตริงเท่ากับ 010

D มีค่าบิตสตริงเท่ากับ 001

E มีค่าบิตสตริงเท่ากับ 000

อัลกอริทึมการขยายกลับข้อมูล

1. ทำการอ่านตารางอาร์เรย์ความถี่

2. สร้างทรีของฮัฟฟ์แมนจากอาร์เรย์ที่เก็บค่าความถี่ โดยสร้างจากโหนดหลักไปโหนดย่อย

3. อ่านไฟล์ข้อมูลที่ถูกลดขนาดทีละบิตต่อครั้ง โดยเริ่มจากส่วนที่เป็นโหนดหลักของทรี

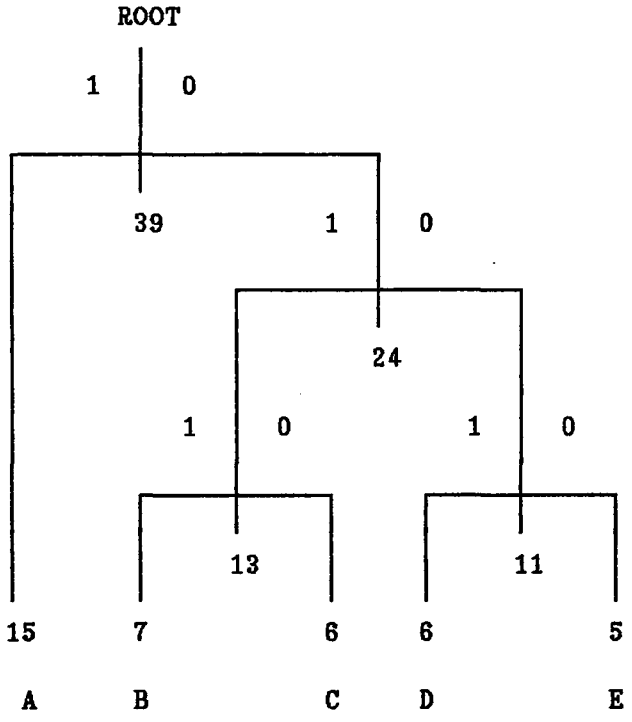
4. เช็คว่าค่าของข้อมูลที่อ่านเข้ามาอยู่กิ่งซ้าย หรือ กิ่งขวาของทรี การอ่านจะเป็นอย่างนี้ไปเรื่อย ๆ จนกว่าจะไม่มีโหนดย่อยลงต่อไปอีก

5. ทำการใส่จุดสีไปที่ปลายโหนดย่อยนั้น

ตัวอย่างที่ 2.4 จากข้อมูลรูปภาพที่ได้ทำการลดขนาดไปแล้ว ในตัวอย่างที่ 2.3 ซึ่งมีค่าความถี่ดังนี้ จงทำการขยายกลับข้อมูลรูปภาพ

ความถี่	:	15	7	6	6	5
ค่าบิตสตริง	:	1	011	010	001	000

ทำการสร้างทรีของฮัฟฟ์แมนจากโหนดหลักไปโหนดย่อย จะได้ทรีดังรูปที่ 2.3



รูปที่ 2.3 ทรีของฮัฟฟ์แมนในการขยายกลับข้อมูล

แทนค่าจุดสีให้กับบิตสตริงเหล่านี้ โดยเริ่มจากโหนดหลักไปที่โหนดย่อยของฮัฟฟ์แมนทรี ก็จะได้ค่าจุดสีของบิตสตริงเหล่านี้ที่โหนดปลายสุด คือ

ค่าบิตสตริง 1 คือ จุดสี A

ค่าบิตสตริง 011 คือ จุดสี B

ค่าบิตสตริง 010 คือ จุดสี C

ค่าบิตสตริง 001 คือ จุดสี D

ค่าบิตสตริง 000 คือ จุดสี E

การลดขนาดข้อมูลโดยวิธีแอล แซด ดับบลิว (LZW)

แนวคิด ปกติข้อมูลขนาด 8 บิตจะแทนรหัส 256 ถ้าเราใช้ข้อมูลขนาด 10 บิตก็จะสามารถแทนข้อมูลได้ถึง 1024 ตัว ทำให้สามารถสร้างอักษรใหม่นอกเหนือจากที่เคยกำหนดไว้แล้ว โดยอักษรใหม่ที่สร้างนี้จะเป็นกลุ่มคำในข้อมูลนั้น และ จะให้กลุ่มคำนั้นมีรหัสต่อจากเดิมที่เราเพิ่มขึ้นมา เมื่อมีการซ้ำของกลุ่มคำซึ่งมีรหัสแล้ว เราก็จะเก็บข้อมูลของกลุ่มคำนั้นเป็นรหัสที่กำหนดไว้แทนที่จะเก็บรหัสอักษรทีละตัว ซึ่งวิธีนี้จะเป็นการลดตัวอักษรในกลุ่มคำนั้นให้เหลือเพียงตัวเดียว

อัลกอริทึมการลดขนาดข้อมูล

การลดขนาดข้อมูลโดยวิธีนี้มีอัลกอริทึมดังนี้

1. อ่านอักษรตัวแรกเข้ามา ให้เป็น w
2. อ่านอักษรตัวถัดไป ให้เป็น K

   ถ้าจบไฟล์ จะเขียนรหัสของ w ลง output และสิ้นสุดการทำงาน
3. ตรวจสอบดูว่าเรากำหนดรหัสให้ wK แล้วหรือยัง โดยตรวจจากตารางรหัส

   ถ้ากำหนดแล้ว ให้ w = wK

   ถ้ายังไม่ได้กำหนด ก็กำหนดรหัสตัวถัดไปให้ wK และ ใส่ลงในตารางรหัส

   เขียนรหัสของ w ลง output จากนั้นให้ w = K
4. ทำขั้นตอนที่ 2 ซ้ำ

ตัวอย่างที่ 2.5 ถ้าข้อมูลเข้า คือ a b a b c b a b a b a a a a a a a จงหาข้อมูลออก

loop	w	K	string ที่เพิ่ม	output
1	a	b	ab	1
2	b	a	ba	2
3	a	b		
	ab			
4	ab	c	abc	4
5	c	b	cb	3
6	b	a		
	ba			
7	ba	b	bab	5
8	b	a		
	ba			
9	ba	b		
	bab			
10	bab	a	baba	8
11	a	a	aa	1
12	a	a		
	aa			
13	aa	a	aaa	10
14	a	a		
	aa			
15	aa	a		

loop	w	K	string ที่เพิ่ม	output
16	aaa aaa a	a	aaaa	11 exit

string table

1	a	7	cb
2	b	8	bab
3	c	9	baba
4	ab	10	aa
5	ba	11	aaa
6	abc	12	aaaa

ดังนั้นจะได้ข้อมูลออก คือ 1 2 4 3 5 8 1 10 11

อัลกอริทึมการขยายกลับข้อมูล

การขยายกลับข้อมูลโดยวิธีนี้มีอัลกอริทึมดังนี้

1. อ่านรหัสตัวแรกเข้ามา ให้เป็น CODE , OLDcode = CODE

2. ตรวจสอบจากตารางรหัสเดิมว่า CODE เป็นรหัสของอักษรใด ถ้า CODE เป็นรหัสของ K ก็เขียน K ลง output และ ให้ FINchar = K

3. อ่านรหัสตัวถัดไป ให้เป็น CODE , INcode = CODE

ถ้าจบไฟล์ ก็สิ้นสุดการทำงาน

ถ้า CODE ไม่มีในตารางรหัส (special case) จะทำดังนี้

เขียน FINchar ลง output

CODE = OLDcode

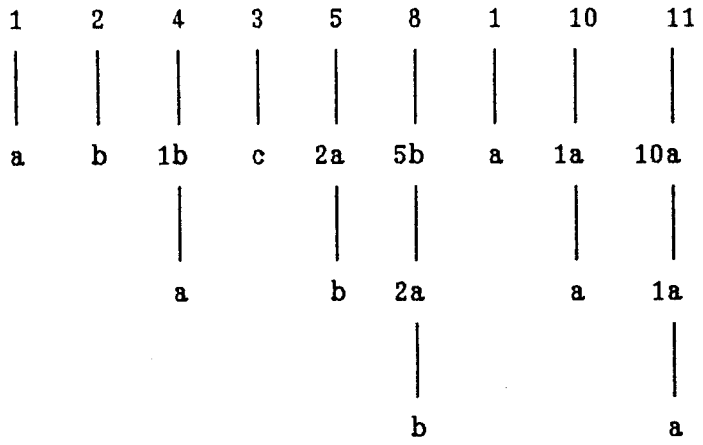
INcode = รหัสของ(OLDcode,FINchar)

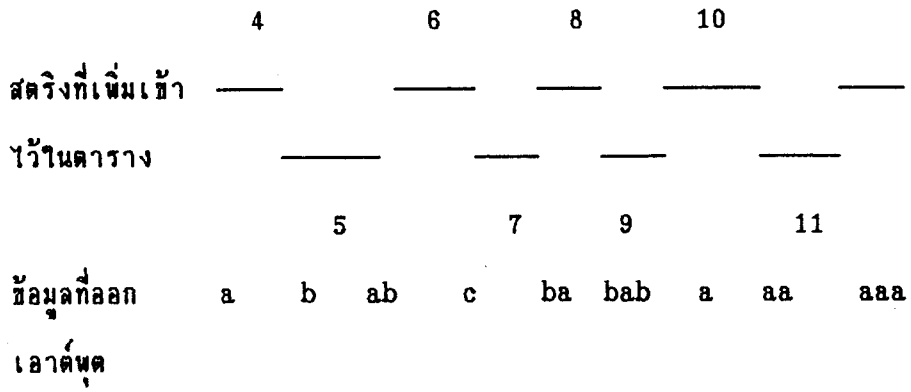
4. ถ้า CODE เป็นรหัสของ wK จะเก็บ K ลง stack และให้ CODE = รหัสของ w ทำซ้ำจนกว่า CODE เป็นรหัสของ K

5. ถ้า CODE เป็นรหัสของ K ให้เขียน K ลง output, K = FINchar จากนั้น pop stack ออกมาเขียนลง output และเขียน รหัสของ(OLDcode,K) ลงในตารางรหัส ให้ OLDcode = INcode

6. ทำซ้ำ ขั้นตอนที่ 3

ตัวอย่างที่ 2.6 ถ้ารหัสที่เข้ามา คือ





string table

(w,K)		(w,K)	
1	a	7	3b
2	b	8	5b
3	c	9	8a
4	1b	10	1a
5	2a	11	10a
6	4c	12	11a

การลดขนาดข้อมูลโดยวิธีบี ที ซี (BTC)

แนวคิด คือ กำหนดบล็อกขนาด 4 X 4 ซึ่งในแต่ละบล็อกจะมีทั้งสิ้น 16 จุดสี และในการเก็บข้อมูล จะเก็บเฉพาะค่าทางสถิติของจุดสีนั้น ซึ่งวิธีนี้จะได้จำนวนจุดสีเท่าเดิม แต่ค่าของจุดสีเดิมจะหายไป คงไว้แต่ค่าทางสถิติของจุดสีนั้น

อัลกอริทึมการลดขนาดข้อมูล

การลดขนาดข้อมูลโดยวิธีนี้มีอัลกอริทึมดังนี้

1. แบ่งจุดสีออกเป็นบล็อก ๆ ละ 16 จุดสี
2. หาค่าเฉลี่ยของค่าแต่ละจุดสี และ ค่าส่วนเบี่ยงเบนมาตรฐาน
3. เปรียบเทียบค่าจุดสีกับค่าเฉลี่ย

   ถ้ามากกว่า ให้บิตนั้นเป็น 1

   ถ้าน้อยกว่า ให้บิตนั้นเป็น 0

ทำจนครบ 16 บิต

4. ให้ค่าที่ใช้แทน บิต 1 คือ ค่าเฉลี่ย + ส่วนเบี่ยงเบนมาตรฐาน และ ค่าที่ใช้แทน บิต 0 คือ ค่าเฉลี่ย - ส่วนเบี่ยงเบนมาตรฐาน
5. เก็บค่าที่ใช้แทนบิต 1 และ ค่าที่ใช้แทนบิต 0
6. ทำซ้ำขั้นตอนที่ 1 จนจบไฟล์

ตัวอย่างที่ 2.7 ถ้าข้อมูลที่เข้ามา แสดงได้ดังตารางนี้ คือ

1	6	7	9
2	2	8	3
5	5	0	4
2	0	1	5

ค่าเฉลี่ย = 3.75

ค่าส่วนเบี่ยงเบนมาตรฐาน = 2.81

เปรียบเทียบค่าจุดสีกับค่าเฉลี่ยจะได้ดังตารางซึ่งจะเก็บไว้ใน 2 ไบต์แรก

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

ค่าที่ใช้แทน บิต 1 = 6.56 โดยจะเก็บไว้ในไบต์ที่ 3

ค่าที่ใช้แทน บิต 0 = 0.94 โดยจะเก็บไว้ในไบต์ที่ 4

อัลกอริทึมการขยายกลับข้อมูล

การขยายกลับข้อมูลโดยวิธีนี้มีอัลกอริทึมดังนี้

1. อ่านข้อมูลจากไฟล์มาทีละ 4 ไบต์
2. ตรวจสอบค่าใน 2 ไบต์แรก

   ถ้าเป็น 1 ให้เก็บค่านั้นด้วยค่าในไบต์ที่ 3

   ถ้าเป็น 0 ให้เก็บค่านั้นด้วยค่าในไบต์ที่ 4
3. ทำซ้ำขั้นตอนที่ 1 จนจบไฟล์

ตัวอย่างที่ 2.8 จากตัวอย่างที่ 2.7 สามารถขยายข้อมูลกลับได้ดังนี้

0.94	6.56	6.56	6.56
0.94	0.94	6.56	0.94
6.56	6.56	0.94	6.56
0.94	0.94	0.94	6.56

บทที่ 3

การพัฒนาโปรแกรมเพื่อการทดสอบ

วิธีการและขั้นตอนการพัฒนาโปรแกรม

ปัญหาพิเศษเรื่องนี้ได้ทำการพัฒนาโปรแกรม เพื่อการทดสอบอัลกอริทึมที่ใช้ในการลดและขยายกลับข้อมูลรูปภาพ โดยแปลงอัลกอริทึมเป็นโปรแกรมภาษาซี โดยใช้คอมไพเลอร์เทอร์โบซี โดยภายในโปรแกรมแต่ละโปรแกรมจะแทรกตัววัดประสิทธิภาพของแต่ละวิธีได้แก่

1. ตัววัดเวลาที่ใช้ในการลดขนาดข้อมูลรูปภาพ
2. ตัววัดที่เป็นตัวนับจำนวนไบต์ของข้อมูลรูปภาพก่อนที่จะมีการลดขนาด
3. ตัววัดที่เป็นตัวนับจำนวนไบต์ของข้อมูลรูปภาพหลังจากที่มีการลดขนาดแล้ว
4. ตัววัดที่เป็นจำนวนเปอร์เซ็นต์ของการลดขนาด
5. ตัววัดที่บอกคุณภาพของข้อมูลรูปภาพที่มีการลดขนาดแล้ว

การแสดงผลของตัววัดต่าง ๆ จะแสดงดังนี้

* แสดงผลของเวลาที่ใช้ในการลดขนาดข้อมูลรูปภาพ โดยคำนวณจากเวลาที่อ่านได้หลังจากที่ได้ลดขนาดเสร็จเรียบร้อยแล้ว ลบด้วยเวลาที่อ่านได้ก่อนที่จะมีการลดขนาดข้อมูลรูปภาพ
* จำนวนไบต์ของข้อมูลรูปภาพก่อนที่จะมีการลดขนาด
* จำนวนไบต์ของข้อมูลรูปภาพหลังจากที่มีการลดขนาดแล้ว
* จำนวนเปอร์เซ็นต์ของการลดขนาดข้อมูลรูปภาพ โดยคำนวณจากผลต่างของจำนวนไบต์ของข้อมูลรูปภาพก่อนที่จะมีการลดขนาด กับจำนวนไบต์ของข้อมูลรูปภาพหลังจากที่มีการลดขนาดแล้ว หารด้วยจำนวนไบต์ของข้อมูลรูปภาพก่อนที่จะมีการลดขนาด แล้วคูณด้วย 100
* แสดงผลของรูปภาพก่อนที่จะมีการลดขนาด
* แสดงผลของรูปภาพหลังจากที่มีการลดขนาดแล้ว

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

วิธีการทดสอบอัลกอริทึม

ในการทดสอบอัลกอริทึมโดยใช้โปรแกรมนั้น จะมีขั้นตอนการทดสอบดังนี้

1. การเตรียมข้อมูลรูปภาพเข้า ผู้ใช้จะต้องทำการโหลด (load) รูปภาพจากเมนูหลักเสียก่อน โดยรูปภาพที่สามารถโหลดได้จากโปรแกรมนี้ต้องเป็นไฟล์รูปภาพที่มีนามสกุลเป็นจุด บี เอ็ม พี (.BMP)

2. ทำการลดขนาดข้อมูลรูปภาพโดยใช้อัลกอริทึมทั้ง 4 ชนิด คือ

-รันเลงท์ (Run-Length)

-ฮัฟฟ์แมน (Huffman)

-แอล แซด ดับบลิว (LZW)

-บี ที ซี (BTC)

ซึ่งอัลกอริทึมทั้ง 4 ชนิดนี้จะให้ข้อมูลรูปภาพที่ลดขนาดแล้วเป็น

จุด อาร์ แอล อี (.RLE),

จุด ฮัฟฟ์ (.HUF),

จุด แอล แซด ดับบลิว (.LZW) และ

จุด บี ที ซี (.BTC) ตามลำดับ

3. ผลของการลดขนาดข้อมูลรูปภาพนั้น สามารถดูประสิทธิภาพของแต่ละอัลกอริทึมได้โดยผู้ใช้เลือก คอมแพร์ (compare) ในเมนูหลัก ซึ่งจะแสดงขนาดของข้อมูลรูปภาพก่อนและหลังการลดขนาด , เปอร์เซ็นต์ในการลดขนาดข้อมูล และเวลาที่ใช้ในการลดขนาดข้อมูล

4. รูปภาพที่ได้หลังจากการลดขนาดข้อมูลแล้ว มาจากการเลือก ดีคอมเพรส (decompress) ในเมนูหลัก โดยข้อมูลรูปภาพเข้าสำหรับการ ดีคอมเพรส คือไฟล์รูปภาพที่มีนามสกุลเป็น

จุด อาร์ แอล อี (.RLE),

จุด ฮัฟฟ์ (.HUF),

จุด แอล แซด ดับบลิว (.LZW) และ

จุด บี ที ซี (.BTC)

5. คุณภาพของรูปภาพก่อน และ หลังการลดขนาดข้อมูล ผู้ใช้สามารถดูได้จากการเลือก วิว (view) ในเมนูหลัก

บทที่ 4

การเปรียบเทียบประสิทธิภาพของอัลกอริทึม

หลักเกณฑ์ที่ใช้ในการเปรียบเทียบ คือ เวลา, เปอร์เซ็นต์ในการลดขนาดข้อมูล และคุณภาพของข้อมูล

จากการทดสอบอัลกอริทึมชนิดต่าง ๆ พบว่า

1. เปอร์เซ็นต์ และ เวลาในการลดขนาดข้อมูล

แยกพิจารณาตามลักษณะข้อมูลต่อไปนี้

1.1 ขนาดของข้อมูล

ขนาดของข้อมูล หน่วยเป็น ไบต์

อัลกอริทึม	100000-199999		200000-299999		มากกว่า 300000	
	%ในการลดขนาด	เวลา (วินาที)	%ในการลดขนาด	เวลา (วินาที)	%ในการลดขนาด	เวลา (วินาที)
Run-Length	73	3.13	-16	5.93	32	4.23*
Huffman	73	2.89*	39	9.28	45	9.99
LZW	89*	3.71	57	8.95	71	10.24
BTC	75	4.12	75*	4.28*	75*	5.05

สรุป

ข้อมูลที่มีขนาดอยู่ในช่วง 100000-199999 ไบต์

อัลกอริทึมของ LZW จะลดขนาดข้อมูลได้มากที่สุด และ

อัลกอริทึมของ Huffman จะใช้เวลาในการลดขนาดข้อมูลน้อยที่สุด

ข้อมูลที่มีขนาดอยู่ในช่วง 200000-299999 ไบต์

อัลกอริทึมของ BTC จะลดขนาดข้อมูลได้มากที่สุด และ

อัลกอริทึมของ BTC จะใช้เวลาในการลดขนาดข้อมูลน้อยที่สุด

ข้อมูลที่มีขนาดมากกว่า 300000 ไบต์

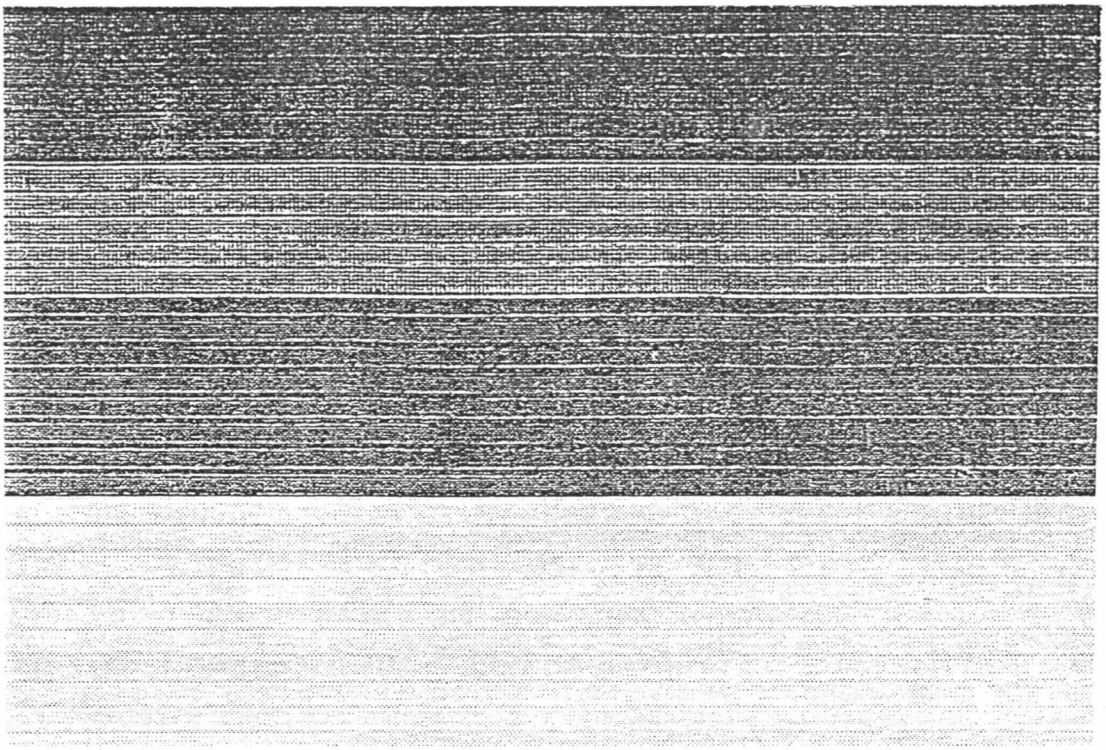
อัลกอริทึมของ BTC จะลดขนาดข้อมูลได้มากที่สุด และ

อัลกอริทึมของ Run-Length จะใช้เวลาในการลดขนาดข้อมูลน้อยที่สุด

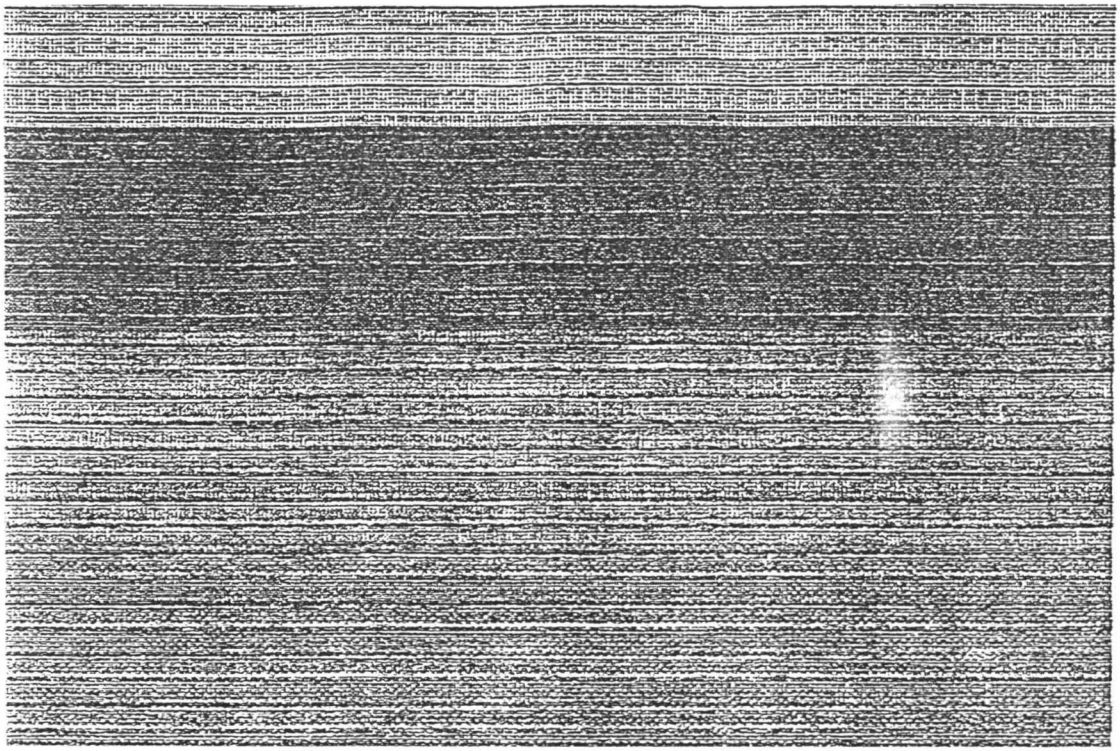
1.2 ลักษณะภาพ

ลักษณะภาพ

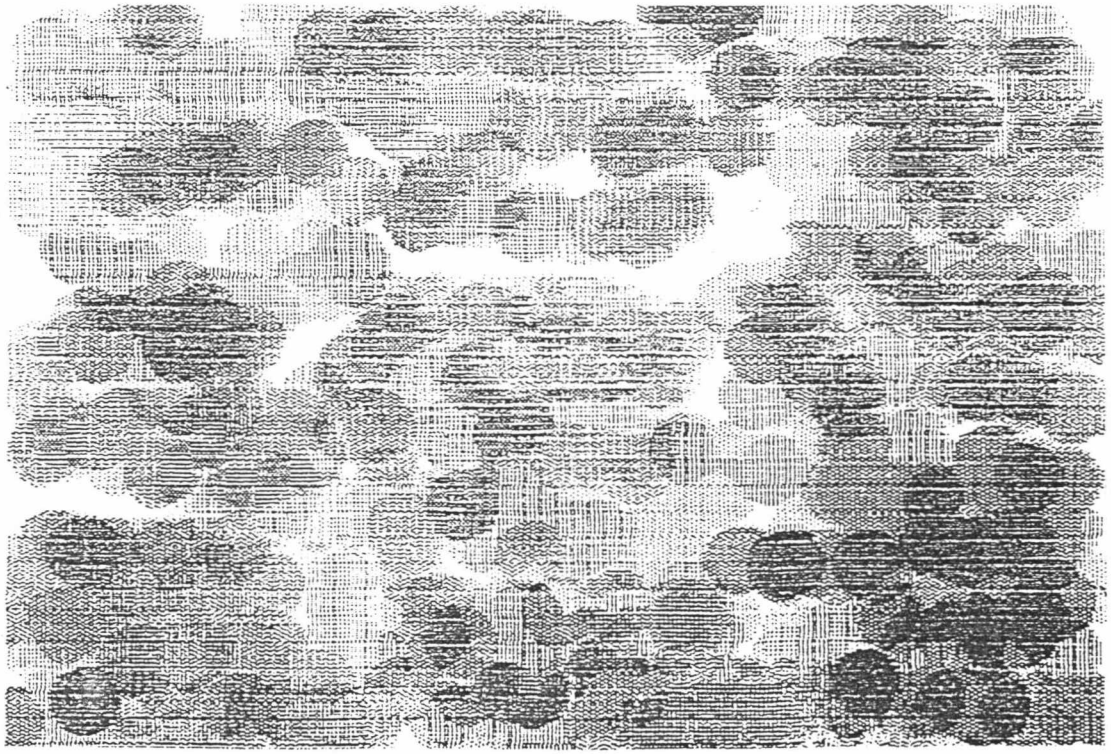
อัลกอริทึม	แถบสีหลายสี		สีเดียวกันต่อเนื่อง		สีซ้ำกันไม่ต่อเนื่อง	
	%ในการลดขนาด	เวลา (วินาที)	%ในการลดขนาด	เวลา (วินาที)	%ในการลดขนาด	เวลา (วินาที)
Run-Length	95	1.54*	95	2.14*	7	4.23*
Huffman	69	5.99	69	6.05	62	6.92
LZW	98*	4.78	98*	3.96	87*	6.04
BTC	75	4.66	75	5.71	75	5.77



รูปที่ 4.1 ภาพที่มีลักษณะเป็นแถบสีหลายสี



รูปที่ 4.2 ภาพที่มีสีเดียวกันอย่างต่อเนื่อง



รูปที่ 4.3 ภาพที่มีสีซ้ำกันหลายที่แต่ไม่ต่อเนื่อง

สรุป

อัลกอริทึมของ LZW จะลดขนาดข้อมูลได้มากที่สุด และ

อัลกอริทึมของ Run-Length จะใช้เวลาในการลดขนาดข้อมูลน้อยที่สุด

2. คุณภาพของข้อมูล

อัลกอริทึมของ Run-Length, Huffman และ LZW เมื่อลดขนาดข้อมูลแล้ว ขยายกลับจะได้ภาพที่มีลักษณะเหมือนภาพเริ่มต้น

อัลกอริทึมของ BTC เมื่อลดขนาดข้อมูลแล้วขยายกลับ จะได้ภาพที่ต่างจากภาพเริ่มต้น แต่ ยังมีเค้าโครงเดิมอยู่

หมายเหตุ

-ผลการทดลองข้างต้นได้มาจากการเฉลี่ยไฟล์รูปภาพจำนวน 5 ไฟล์

-การทดสอบอัลกอริทึมทั้ง 4 ชนิดนี้ทดสอบโดยใช้เครื่องไมโครคอมพิวเตอร์รุ่น 486DX-33 และ ใช้ DOS version 6

บทที่ 5

สรุปและข้อเสนอแนะ

สรุป

ปัญหาพิเศษเรื่อง การพัฒนาและทดสอบอัลกอริทึมสำหรับการลดขนาดข้อมูลรูปภาพ นี้เป็นการศึกษาลักษณะการจัดเก็บข้อมูลที่เป็นภาพ รวมถึงการเปรียบเทียบประสิทธิภาพของอัลกอริทึมทั้ง 4 ชนิดที่กล่าวมาแล้วคือ

1. รันเลงท์ (Run-Length)
2. ฮัฟฟ์แมน (Huffman)
3. แอล แซด ดับบลิว (LZW)
4. บี ที ซี (BTC)

โดยเปรียบเทียบความสามารถของอัลกอริทึมในด้านความเร็ว (เวลาที่ใช้ในการลดขนาดข้อมูลรูปภาพ), เปอร์เซ็นต์ของการลดขนาดข้อมูลรูปภาพและคุณภาพของภาพที่ได้หลังการลดขนาดข้อมูลรูปภาพแล้ว

สำหรับผลที่ได้จากการทดสอบโปรแกรม สามารถสรุปได้ว่า

-ถ้าภาพที่มีขนาดอยู่ในช่วง 100000 - 199999 ไบต์ ควรเลือกใช้อัลกอริทึมของ LZW

-ถ้าภาพมีขนาดมากกว่า 200000 ไบต์ ควรคำนึงถึงจุดประสงค์ของเราว่าต้องการจะเก็บภาพนี้ไว้ใช้ทำอะไร ถ้าต้องการรายละเอียดของภาพเดิมก็ควรเลือกใช้อัลกอริทึมของ LZW แต่ถ้าไม่ต้องการรายละเอียดมากนักก็เลือกใช้อัลกอริทึมของ BTC

-อัลกอริทึมของ Run-Length จะเหมาะสำหรับ ภาพที่มีสีซ้ำกันอย่างต่อเนื่อง

-อัลกอริทึมของ Huffman จะเหมาะสำหรับ ภาพที่มีสีซ้ำกันโดยไม่จำเป็นต้องอยู่ติดกัน

-อัลกอริทึมของ LZW จะเหมาะสำหรับ ภาพที่มีชุดของสีซ้ำกันโดยไม่จำเป็นต้องอยู่ติดกัน

-อัลกอริทึมของ BTC จะเหมาะสำหรับ การเก็บข้อมูลภาพที่ไม่ต้องการรายละเอียดมากนัก

ข้อเสนอแนะ

ปัญหาพิเศษเรื่อง การพัฒนาและทดสอบอัลกอริทึมสำหรับการลดขนาดข้อมูลรูปภาพนี้ เป็นการนำเอาทฤษฎีการลดและการขยายกลับข้อมูลมาศึกษา ในที่นี้นำมาเพียง 4 อัลกอริทึมเท่านั้น ซึ่งที่จริงแล้วยังมีอัลกอริทึมอื่นๆ อีกมาก และ อัลกอริทึมเหล่านั้นอาจมีประสิทธิภาพที่ดีกว่าก็ได้ นอกจากนี้ในส่วนของการพัฒนาโปรแกรม ข้อมูลรูปภาพที่นำเข้ามานั้นใช้ได้เฉพาะรูปภาพที่มี 256 สีเท่านั้น ซึ่งไม่สามารถใช้ได้กับรูปภาพชนิดอื่น ๆ

จึงขอเสนอให้ผู้ที่มีความสนใจในปัญหาพิเศษนี้ ได้ทำการศึกษาและพัฒนาโปรแกรมดังนี้

1. เพิ่มอัลกอริทึมที่ใช้ในการลดขนาดข้อมูล
2. เขียนโปรแกรมเพิ่มเติม เพื่อให้ใช้ได้กับภาพทุกชนิด

เอกสารอ้างอิง

1. Mark Nelson,*The compression Book*,Prentice Hall.

2. Steve Rimmer,*Bit-Mapped Graphics*,McGraw-Hill.

3. วิโรจน์ อัศวรังสี,บุญชู งามไพโรจน์พิบูลย์ "ลด...ลดขนาดข้อมูล"
คอมพิวเตอร์รีวิว 92 (เมษายน 2535): 164

4. สาธิต ทองจีน "การลดขนาดข้อมูลรูปภาพแบบ Block Truncation"
คอมพิวเตอร์รีวิว 103 (มีนาคม 2536): 317

5. สาธิต ทองจีน "รู้จัก Bitmapped Graphic ตอน PCX และ BMP"
คอมพิวเตอร์รีวิว 106 (มิถุนายน 2536): 221