

การพัฒนาซอฟต์แวร์การแบ่งส่วนภาพและบีบอัดข้อมูลภาพเพื่อเพิ่มประสิทธิภาพการส่งข้อมูล
ผ่านอินเทอร์เน็ต



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
บัณฑิตวิทยาลัย สาขาเทคโนโลยีวิศวกรรม
สถาบันเทคโนโลยีไทย-ญี่ปุ่น
ปีการศึกษา 2555

SOFTWARE DEVELOPMENT ON IMAGE SEGMENTATION AND DATA COMPRESSION
FOR EFFICIENCY ENHANCEMENT IN DATA TRANSMISSION OVER INTERNET

Paisan Sripraram

TNI

A Thesis Submitted in Partial Fulfillment of Requirements
for the Degree of Master of Engineering Technology Program in Engineering
Graduate School

Thai-Nichi Institute of Technology

Academic Year 2012

หัวข้อวิทยานิพนธ์

การพัฒนาซอฟต์แวร์การแบ่งส่วนภาพและบีบ
อัดข้อมูลภาพเพื่อเพิ่มประสิทธิภาพการส่ง
ข้อมูลผ่านอินเทอร์เน็ต

โดย

ไพศาล ศรีพระราม

สาขาวิชา

เทคโนโลยีวิศวกรรม

อาจารย์ที่ปรึกษาวิทยานิพนธ์

ผู้ช่วยศาสตราจารย์ ดร. วรากร ศรีเชวงทรัพย์

บัณฑิตวิทยาลัย สถาบันเทคโนโลยีไทย-ญี่ปุ่น อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็น
ส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

.....คณบดีบัณฑิตวิทยาลัย

(รองศาสตราจารย์ ดร. พิชิต สุขเจริญพงษ์)

วัน.....เดือน.....พ.ศ.....

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ดร. กำพล วรดิษฐ์)

.....กรรมการ

(ดร. วิมล แสนอ้อม)

.....กรรมการ

(ดร. กันติชา กิตติพิรัช)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์

(ผู้ช่วยศาสตราจารย์ ดร. วรากร ศรีเชวงทรัพย์)

ไพศาล ศรีพระราม : การพัฒนาซอฟต์แวร์การแบ่งส่วนภาพและบีบอัดข้อมูลภาพเพื่อเพิ่มประสิทธิภาพการส่งข้อมูลผ่านอินเทอร์เน็ต. อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร. วรากร ศรีเชวงทรัพย์, 54 หน้า.

งานวิจัยนี้นำเสนอการพัฒนาซอฟต์แวร์เพื่อลดขนาดข้อมูลภาพ โดยนำเทคนิคการหาขอบภาพด้วยวิธี Canny และการแบ่งส่วนภาพมาใช้เพื่อแยกและตัดข้อมูลส่วนที่ผู้ใช้ไม่ต้องการออก และนำข้อมูลภาพส่วนที่สนใจมาใช้ จากนั้นนำข้อมูลส่วนที่ได้จากการแบ่งส่วนแล้วมาทำการบีบอัดข้อมูลโดยใช้หลักการของ Zlib Deflate เพื่อทำให้ข้อมูลมีขนาดเล็กลง ซึ่งช่วยให้การส่งข้อมูลผ่านอินเทอร์เน็ตทำได้รวดเร็ว และมีความต่อเนื่องมากขึ้น ซึ่งในงานวิจัยนี้จะแสดงการออกแบบซอฟต์แวร์ที่พัฒนาขึ้นด้วยโปรแกรม MATLAB ผลที่ได้จากการทดสอบ พบว่าการแบ่งส่วนภาพและการบีบอัดทำให้ข้อมูลภาพมีขนาดลดลง และการส่งข้อมูลภาพผ่านทางอินเทอร์เน็ตมีความต่อเนื่องมากขึ้น

บัณฑิตวิทยาลัย

สาขาวิชา เทคโนโลยีวิศวกรรม

ปีการศึกษา 2555

ลายมือชื่อนักศึกษา.....

ลายมือชื่ออาจารย์ที่ปรึกษา.....

PAISAN SRIPARAM : SOFT5464WARE DEVELOPMENT ON IMAGE
SEGMENTATION AND DATA COMPRESSION FOR EFFICIENCY
ENHANCEMENT IN DATA TRANSMISSION OVER INTERNET.
ADVISOR : ASST.PROF.DR. WARAKORN SRICHAVENGSUP, 54 PP.

This thesis describes a software development on image data reduction by using Canny edge detection and image segmentation in order to remove unnecessary data such as image background. Moreover, Zlib Deflate compression algorithm is used to achieve an effective data reduction capability. This software is developed using MATLAB programming language and MATLAB GUI. Results show that the developed software can reduce the transmitted data size and increase the frame rate.

Graduate School

Student's Signature.....

Field of Engineering Technology

Advisor's Signature.....

Academic Year 2012

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จลุล่วงตามกำหนดเวลาได้เนื่องจากความช่วยเหลือของหลาย ๆ ท่าน อันดับแรกขอขอบคุณท่านอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร. วรากร ศรีเชวงทรัพย์ ที่ได้เสียสละเวลา เพื่อช่วยเหลือให้คำปรึกษาต่าง ๆ อีกทั้ง ผู้ช่วยศาสตราจารย์ ดร. วิภาวดี วงษ์สุวรรณ และ ดร. วิมล แสนอุ่ม รวมทั้ง เพื่อน ๆ ที่ร่วมทำวิจัยที่คอยช่วยเหลือ แนะนำ ทั้งในเรื่องวิธีการทำงานวิจัย และเรื่องงานที่เกี่ยวข้องต่าง ๆ

ขอขอบคุณ สถาบันเทคโนโลยีไทย-ญี่ปุ่น ที่ได้ให้โอกาสได้มาศึกษาทั้งในห้องเรียน และได้สามารถทำงานวิจัยชิ้นนี้ขึ้นมาได้

ขอขอบคุณ บุคคลที่คอยช่วยเหลือ ทั้งในเรื่องการให้ยืมอุปกรณ์คอมพิวเตอร์ ช่วยเหลือเรื่องสถานที่ในการทดสอบงานวิจัย และเวลาในการช่วยเหลืออื่น ๆ ที่ได้เสียสละมาให้อย่างเต็มใจ

งานวิจัยเล่มนี้ จะไม่สามารถสำเร็จลุล่วงไปได้ ถ้าไม่มี พ่อ แม่ และครอบครัว ที่เป็นแรงผลักดัน ให้ได้ทำงานวิจัยนี้จนสำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ขอขอบคุณทุก ๆ ท่านที่ได้ช่วยเหลือทั้งทางตรงและทางอ้อม ที่ไม่สามารถกล่าวได้ทั้งหมด ณ ที่นี้ ด้วย ขอขอบพระคุณทุก ๆ ท่านเป็นอย่างสูงครับ

ไพศาล ศรีพระราม

TNI

THAI - NICHU INSTITUTE OF TECHNOLOGY

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ	ฉ
สารบัญ	ช
สารบัญตาราง	ฌ
สารบัญรูป	ญ
 บทที่	
1 บทนำ.....	1
1.1 ความสำคัญและที่มาของงานวิจัย.....	1
1.2 ขั้นตอนการทำงานวิจัย.....	1
1.3 ขอบเขตการวิจัย.....	2
1.4 วัตถุประสงค์ของโครงการวิจัย.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย.....	2
2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	3
2.1 ระบบภาพ.....	3
2.2 การแบ่งส่วนภาพ (Image Segmentation).....	6
2.3 การบีบอัดข้อมูล (Data Compression).....	14
2.4 การส่งข้อมูลผ่านระบบเครือข่ายคอมพิวเตอร์.....	19
2.5 ประวัติและเทคโนโลยีที่เกี่ยวข้อง.....	22
3 การออกแบบซอฟต์แวร์.....	25
3.1 การออกแบบซอฟต์แวร์.....	25
3.2 การออกแบบระบบทดสอบและการแสดงผล.....	28
3.3 วิธีการทดสอบประสิทธิภาพของโปรแกรม.....	32
3.4 ผลการทดสอบที่คาดหวัง.....	32

สารบัญ (ต่อ)

บทที่	หน้า
4 ผลการทดสอบและการวิเคราะห์ผล.....	33
4.1 ผลการทดสอบ.....	33
4.2 การวิเคราะห์ผล.....	41
5 สรุปผลการวิจัยและข้อเสนอแนะ.....	42
5.1 สรุปผลการวิจัย.....	42
5.2 ข้อเสนอแนะและแนวทางการปรับปรุงงานวิจัย.....	42
บรรณานุกรม.....	43
ภาคผนวก.....	47
ประวัติย่อผู้วิจัย.....	54



สารบัญตาราง

ตารางที่		หน้า
2.1	การเปรียบเทียบผลการบีบอัดที่ได้จากแต่ละวิธี.....	17
4.1	ขนาดของข้อมูลที่ได้จากการทดสอบ.....	36
4.2	เวลาในการส่งข้อมูลที่ได้จากการทดสอบ.....	39



สารบัญรูป

รูปที่		หน้า
2.1	ตำแหน่งของจุดภาพ.....	4
2.2	เมตริกซ์ของจุดภาพ.....	4
2.3	กราฟฮิสโทแกรม.....	5
2.4	วิธีการแยกองค์ประกอบของรูปภาพโดยใช้กราฟฮิสโทแกรม.....	7
2.5	การหาขอบโดยใช้ฮิสโทแกรมอันดับหนึ่ง.....	9
2.6	การหาขอบโดยใช้ฮิสโทแกรมอันดับสอง.....	10
2.7	ตัวอย่างการเปรียบเทียบผลของการหาขอบภาพด้วยวิธีต่าง ๆ.....	11
2.8	ขั้นตอนวิธีการหาขอบภาพด้วยวิธี Canny Edge Detection.....	12
2.9	ตัวอย่างภาพของแต่ละลำดับขั้นตอนการหาขอบภาพด้วยวิธี Canny.....	14
2.10	ตัวอย่างการหาขอบภาพด้วยวิธีของ Canny กับรูปใบหน้าคน.....	14
2.11	การบีบอัดข้อมูลแบบ Lossless และแบบ Lossy.....	16
2.12	การเข้าและถอดรหัสการบีบอัดข้อมูล.....	18
2.13	เปรียบเทียบระหว่างโปรโตคอล UDP และโปรโตคอล TCP.....	21
3.1	ขั้นตอนของโปรแกรมการลดขนาดข้อมูลภาพ.....	25
3.2	การหาขอบภาพด้วยวิธี Canny.....	26
3.3	การออกแบบการแบ่งส่วนภาพ.....	27
3.4	การออกแบบการบีบอัดข้อมูลภาพ.....	28
3.5	หน้าต่างที่ออกแบบเพื่อการแสดงผล.....	30
3.6	การแสดงผลของการทดสอบ.....	31
4.1	ผลที่ได้จากการแบ่งส่วนภาพ.....	34
4.2	ตัวอย่างภาพผลที่ได้จากการทดสอบกับพื้นหลังสีอ่อน.....	34
4.3	ตัวอย่างภาพผลที่ได้จากการทดสอบกับพื้นหลังสีเข้ม.....	35
4.4	ผลการทดสอบขนาดข้อมูลที่ถูกแบ่งส่วนและบีบอัด.....	39
4.5	ผลการทดสอบเวลาในการส่งข้อมูล.....	40

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของงานวิจัย

ในปัจจุบันการส่งผ่านข้อมูลในระบบเครือข่ายอินเทอร์เน็ตได้เข้ามามีบทบาทต่อชีวิตประจำวันมากขึ้น ซึ่งเป็นผลมาจากเทคโนโลยีดิจิทัลที่ได้ถูกนำมาใช้เป็นจำนวนมาก รวมทั้งการส่งผ่านทางระบบเครือข่ายคอมพิวเตอร์ ด้วยอุปกรณ์พกพา หรือระบบคอมพิวเตอร์เอง ปัญหาที่มีความสำคัญอย่างหนึ่งก็คือข้อจำกัดในเรื่องของอัตราการส่งผ่านข้อมูลบนอินเทอร์เน็ต ส่งผลทำให้การส่งข้อมูลภาพที่มีขนาดใหญ่มีปัญหาในการแสดงผลภาพที่ไม่ต่อเนื่อง ซึ่งอาจทำให้เกิดปัญหาเรื่องความผิดพลาดและไม่ราบรื่นในการสื่อสารได้

ปัจจัยหลักที่มีผลกระทบสำคัญต่อการส่งข้อมูลทางอินเทอร์เน็ตเช่น ความเร็วของสัญญาณอินเทอร์เน็ตซึ่งในบางกรณีที่มีผู้ใช้จำนวนมากในเวลาเดียวกัน อาจทำให้สัญญาณอินเทอร์เน็ตมีความเร็วลดลง ปัจจัยหลักอีกประการหนึ่งคือปริมาณ หรือขนาดของข้อมูลที่ถูกส่งผ่านอินเทอร์เน็ต ก็มีผลโดยตรงต่อการแสดงผลข้อมูลภาพ ซึ่งในปัจจุบันวิธีการต่าง ๆ ของการประมวลผลภาพก็ถูกนำมาใช้ในการแก้ปัญหา เช่นการพัฒนาความเร็วของอัตราการส่งสัญญาณ และการทำข้อมูลให้มีขนาดเล็กลง เพื่อให้เหมาะสมต่อการจัดเก็บข้อมูล และการส่งผ่านข้อมูลภายในระบบเครือข่ายอินเทอร์เน็ต

ในงานวิจัยนี้ได้นำเสนอวิธีการรวมการใช้เทคนิคการแบ่งส่วนภาพร่วมกับการบีบอัดข้อมูลภาพ เพื่อให้ข้อมูลภาพมีขนาดลดลง และทำให้การแสดงผลภาพมีความต่อเนื่องมากขึ้น ส่งผลให้การสื่อสารผ่านระบบเครือข่ายอินเทอร์เน็ตมีประสิทธิภาพมากขึ้น

1.2 ขั้นตอนการทำงานวิจัย

- 1.2.1 ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการแบ่งส่วนภาพ
- 1.2.2 ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการบีบอัดข้อมูลภาพ
- 1.2.2 ศึกษาหลักการที่เกี่ยวข้องกับระบบเครือข่ายอินเทอร์เน็ต
- 1.2.3 ออกแบบซอฟต์แวร์
- 1.2.4 ทดสอบการประมวลผลซอฟต์แวร์
- 1.2.5 ออกแบบ GUI (Graphic User Interface) หรือส่วนต่อประสานกราฟิกกับผู้ใช้ เพื่อแสดงผลลัพธ์ที่ได้
- 1.2.6 เปรียบเทียบผลลัพธ์ แกไขข้อบกพร่องในการออกแบบ
- 1.2.7 สรุปผลงานวิจัย

1.3 ขอบเขตการวิจัย

1.3.1 พัฒนาซอฟต์แวร์ที่สามารถลดขนาดข้อมูลภาพโดยใช้เทคนิคการแบ่งส่วนภาพและบีบอัดข้อมูล

1.3.2 สร้างต้นแบบส่วนประสานกราฟิกกับผู้ใช้ เพื่อแสดงผลการส่งข้อมูลผ่านทางอินเทอร์เน็ตได้

1.4 วัตถุประสงค์ของโครงการวิจัย

1.4.1 เพื่อพัฒนาซอฟต์แวร์ซึ่งช่วยให้การส่งข้อมูลภาพทางอินเทอร์เน็ตมีประสิทธิภาพและมีความต่อเนื่องมากขึ้น

1.4.2 เพื่อศึกษาประสิทธิภาพของเทคนิคการแบ่งส่วนภาพและบีบอัดข้อมูล ที่นำมาใช้ในการลดขนาดข้อมูลภาพ

1.5 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

1.5.1 ได้ซอฟต์แวร์ที่สามารถลดขนาดข้อมูลภาพ เพื่อให้การแสดงผลภาพที่ส่งผ่านทางอินเทอร์เน็ตมีความต่อเนื่องมากขึ้น

1.5.2 ได้ต้นแบบซอฟต์แวร์ที่สามารถแสดงประสิทธิภาพของการแบ่งส่วนภาพและการบีบอัดข้อมูล



บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ระบบภาพ

2.1.1 ภาพดิจิทัล

ภาพเป็นกระบวนการทางแสง (Optical Process) ซึ่งเกิดจากพลังงานคลื่นแม่เหล็กไฟฟ้า (Electromagnetic Spectrum) หลาย ๆ ช่วงความถี่ เช่น แสงธรรมดา รังสีเอ็กซ์เรย์ (X-Ray) รังสีอินฟราเรด (Infrared) เป็นต้น และพลังงานเสียง เช่น อัลตราซาวนด์ (Ultrasound) ตกกระทบวัตถุแล้วสะท้อนกลับเข้าสู่ประสาทรับรู้ทางตาของมนุษย์ หรืออุปกรณ์ตรวจจับ เช่น เซนเซอร์ (Sensor) เป็นต้น

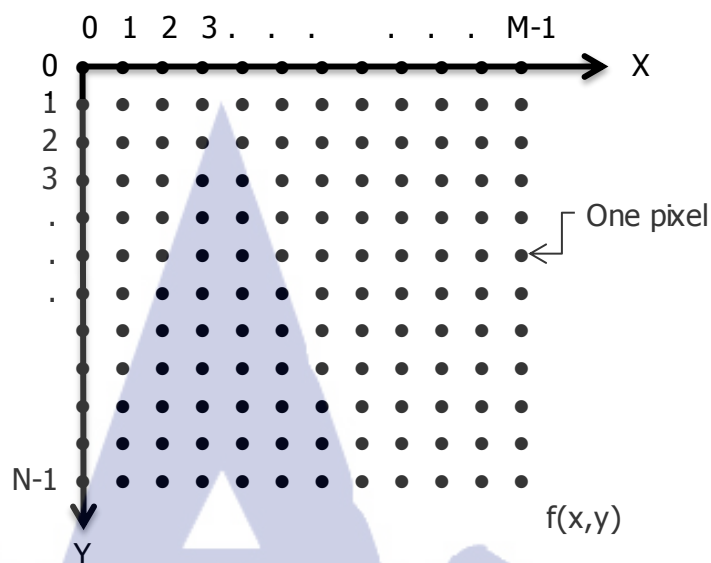
ในทางวิชาการ ภาพดิจิทัลนั้นก็คือ ฟังก์ชัน 2 มิติ หรือ $f(x,y)$ ของค่าความเข้มของแสงโดยที่ x และ y คือ ค่าที่บอกถึงตำแหน่งในระบบพิกัดฉาก และค่าของฟังก์ชัน ณ ตำแหน่งใด ๆ จะเป็นสัดส่วนกับความสว่างของแสง ณ ตำแหน่งนั้น กระบวนการแปลงภาพให้เป็นภาพในเชิงดิจิทัลเรียกว่า Image Digitization

ก่อนที่จะได้ภาพมา อันดับแรกจะต้องทำการถ่ายภาพ การถ่ายภาพเป็นการแปลงภาพเชิงต่อเนื่อง (Continuous Image) แบบ 3 มิติ ให้เป็นภาพเชิงต่อเนื่อง 2 มิติ โดยใช้อุปกรณ์เชิงแสง (Optical Device) เช่น กล้องถ่ายรูป เพื่อแปลงภาพให้มาเป็นภาพบนฟิล์ม รูปถ่ายบนกระดาษ หรือภาพบนจอคอมพิวเตอร์ โดยปกติแล้วภาพที่ตาเรามองเห็นกันอยู่นั้น มีทั้งความกว้าง ความสูง และความลึก ซึ่งเป็นแบบ 3 มิติ แต่การถ่ายภาพด้วยกล้องจะทำให้เราได้ภาพที่มีแต่ความกว้าง และความสูงเท่านั้น ซึ่งเป็นภาพแบบ 2 มิติ เพราะเราไม่สามารถถ่ายส่วนของความลึกมาด้วยได้

2.1.2 จุดภาพ (Pixel)

จุดภาพ (Pixel) เป็นหน่วยพื้นฐานของภาพ เป็นจุดภาพบนจอแสดงผล หรือจุดภาพในรูปภาพที่ถูกรวมกันเป็นภาพใด ๆ โดยภาพนั้น ๆ จะประกอบไปด้วยจุดภาพมากมาย และแต่ละภาพที่ถูกสร้างขึ้นจะมีความหนาแน่นของจุดภาพ หรือบางครั้งแทนด้วยความละเอียด (ความคมชัด) ที่แตกต่างกันไป

คำว่า "พิกเซล" (Pixel) มาจากคำว่า "พิกเจอร์" (Picture) ที่แปลว่ารูปภาพ และ "เอเลเมนต์" (Element) ที่แปลว่าองค์ประกอบ จึงใช้ในการบอกคุณสมบัติของภาพ จอภาพ หรืออุปกรณ์แสดงผลภาพใด ๆ จอภาพที่มีจำนวนจุดภาพมาก จะมีความละเอียดของภาพมาก โดยทั่วไปจะระบุจำนวนจุดภาพแนวนอน (x) คูณด้วยแนวตั้ง (y) เช่น 1366×768 จุดภาพ ดังรูปที่ 2.1



รูปที่ 2.1 ตำแหน่งของจุดภาพ

เมื่อ N และ M แทนจำนวนจุดภาพที่มากที่สุดในแกน Y และแกน X ตามลำดับ

ในภาพหนึ่ง ๆ เราสามารถอธิบายได้เป็นเมตริกซ์ของจุดภาพขนาด $N \times M$ โดยใช้คู่ลำดับ $p(i, j)$ แทนค่าของจุดแต่ละจุด ดังรูปที่ 2.2 โดยค่าสีของแต่ละจุดภาพจะแสดงหรือบ่งชี้ความเข้มแสงที่จุดภาพนั้น ๆ เป็นค่าตัวเลขใด ๆ

	0	j						M
0	(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)	(0,6)	(0,7)
	(1,0)	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)
i	(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)	(2,7)
	(3,0)	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)	(3,7)
	(4,0)	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)	(4,7)
N	(5,0)	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	(5,7)

รูปที่ 2.2 เมตริกซ์ของจุดภาพ

2.1.3 ฮิสโทแกรม (Histogram)

ข้อมูลที่เก็บรวบรวมมาได้และยังไม่ได้มีการจัดหมวดหมู่ข้อมูลชนิดนี้เรียกว่าข้อมูลดิบ (Raw Data) ถ้าข้อมูลดิบมีจำนวนน้อย ก็อาจใช้การจัดการโดยวิธีเรียงข้อมูล จากมากไปหาน้อย หรือจากน้อยไปหามาก ข้อมูลที่เรียงลำดับแบบนี้เรียกว่า Ungrouped Data แต่ถ้าข้อมูลดิบเหล่านี้มีจำนวนมาก จะทำให้สามารถมองเห็นลักษณะของข้อมูลได้ยาก จึงต้องมีการจัดเตรียมข้อมูลดิบให้เป็นหมวดหมู่ ซึ่งอาจใช้วิธีการแจกแจงความถี่ (Frequency Distribution)

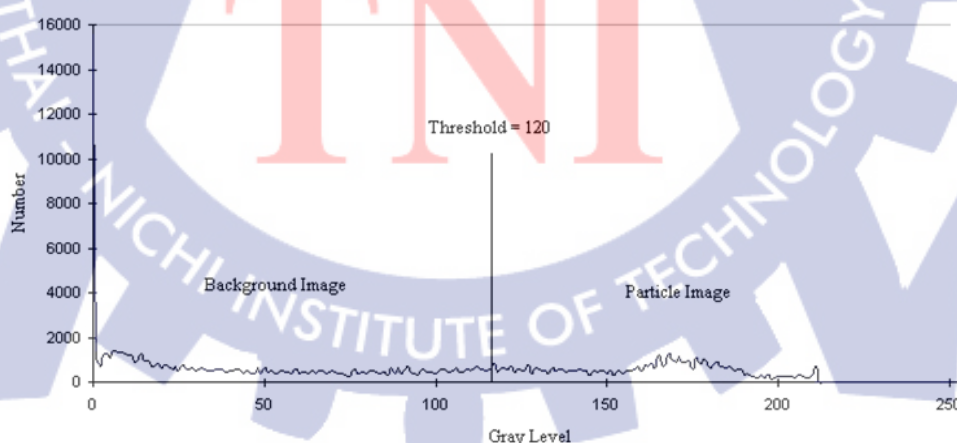
การแจกแจงความถี่ เป็นการจัดเรียงลำดับข้อมูลดิบที่เก็บรวบรวมมาได้ โดยจัดให้เป็นหมวดหมู่ แล้วหาจำนวนของข้อมูลในแต่ละหมวดหมู่ ข้อมูลที่หาได้โดยวิธีการนี้เรียกว่า ข้อมูลที่เป็นหมวดหมู่ (Grouped Data)

รูปแบบของการแจกแจงความถี่ สามารถทำได้ 3 รูปแบบคือ

- (1) ตารางแจกแจงความถี่ (Frequency Distribution)
- (2) ตารางแจกแจงความถี่สะสม (Cumulative Frequency Distribution)
- (3) แผนภูมิหรือกราฟ โดยจะกล่าวถึงการแจกแจงความถี่แบบแผนภูมิ ใน

รูปแบบ ฮิสโทแกรม

ฮิสโทแกรมในเรื่องระบบภาพ จะเป็นกราฟแสดงจำนวนจุดภาพที่ความสว่างต่างๆ ของภาพ สังเกตได้จากตัวอย่างด้านล่าง แกนนอนเป็นระดับความสว่างที่แบ่งระดับเป็น 256 ระดับ มักเรียกว่าระดับสีเทา หรือ Gray Level โดยมีค่าตั้งแต่ 0-255 เมื่อระดับสีเทามีค่าต่ำ (ด้านซ้ายมือ) หมายถึงมีความสว่างน้อย จะมองเห็นเป็นสีดำ ค่าระดับสีเทามาก (ด้านขวามือ) หมายถึงมีความสว่างมากจะมองเห็นเป็นสีขาว แกนตั้งของกราฟแสดงจำนวนจุดภาพในแต่ละระดับสีเทาซึ่งเป็นค่าสัมพัทธ์ การแสดงกราฟฮิสโทแกรมแสดงได้ดังรูปที่ 2.3



รูปที่ 2.3 กราฟฮิสโทแกรม

2.2 การแบ่งส่วนภาพ (Image Segmentation)

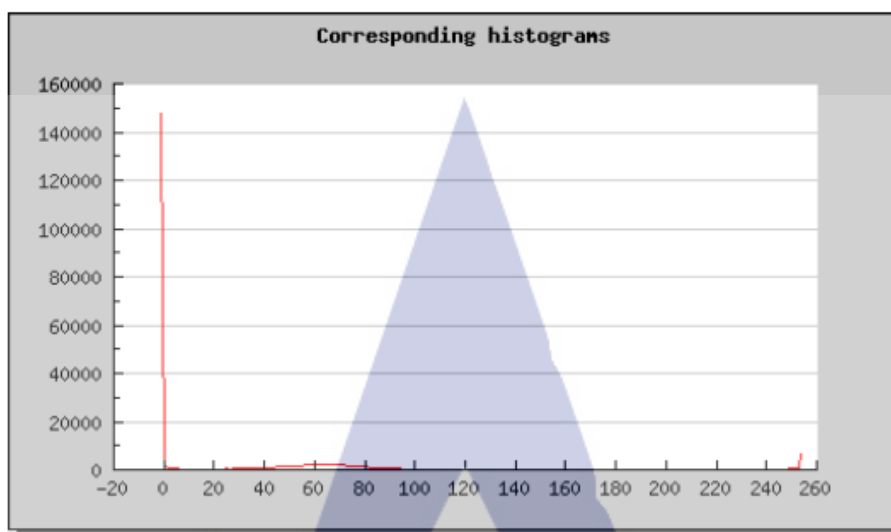
การแบ่งย่อยส่วนประกอบภาพ ใช้ในการแยกองค์ประกอบต่างๆ ของรูปภาพออกจากรูปภาพตามลักษณะสำคัญที่เราต้องการพิจารณา ขั้นตอนนี้เป็นขั้นตอนสำคัญของการวิเคราะห์ภาพ ดังนั้นประโยชน์ของการแบ่งย่อยส่วนประกอบภาพคือ สามารถลดจำนวนข้อมูลในรูปภาพที่ไม่จำเป็นในการวิเคราะห์ได้เนื่องจากการแบ่งย่อยส่วนประกอบของภาพเป็นการแยกแยะระหว่างส่วนที่เราสนใจ เช่น วัตถุในภาพ กับส่วนที่ไม่ต้องการ เช่น ฉากหลัง เมื่อตัดข้อมูลในส่วนที่ไม่ต้องการออกไปจำนวนข้อมูลที่เหลือที่จำเป็นในการวิเคราะห์จริงจะลดลงอย่างมาก และยังสามารถจัดระเบียบข้อมูลในรูปภาพให้เป็นกลุ่มได้ง่ายขึ้น ข้อมูลภาพที่ผ่านการแบ่งแยกแล้วจะมีโครงสร้างที่ชัดเจนขึ้นและนำไปใช้งานได้สะดวกขึ้น

2.2.1 ประเภทของการแบ่งส่วนภาพ

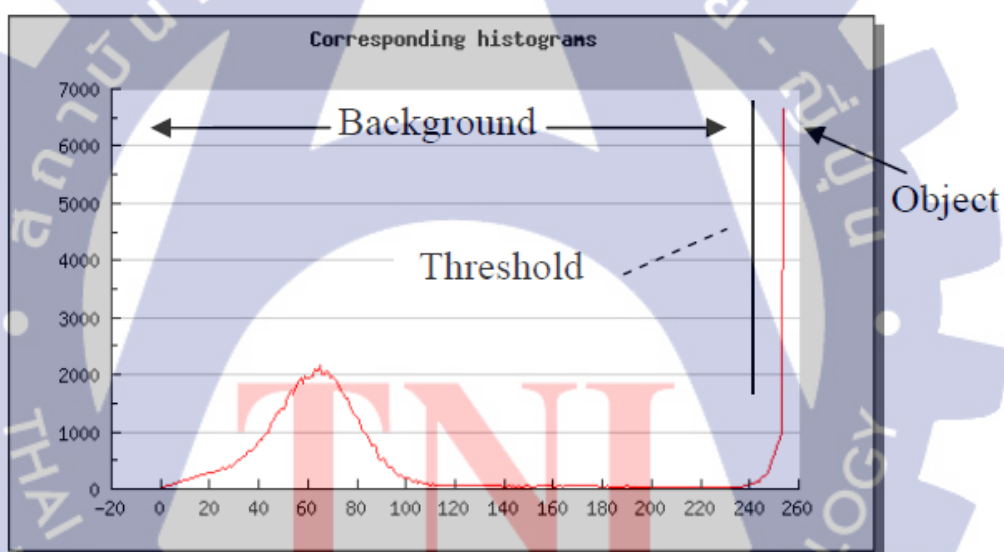
เราสามารถแบ่งประเภทของการแบ่งย่อยส่วนประกอบภาพได้ 3 แบบคือ

(1) Pixel Oriented Image Segmentation

Pixel Oriented Image Segmentation เป็นวิธีการแยกองค์ประกอบของรูปภาพโดยดูจากความเหมือนกันของคุณสมบัติของจุดภาพภายในพื้นที่เพียงอย่างเดียว เช่น สี ความเข้มแสง ค่าทางสถิติของจุดภาพ เช่น สีเหมือนกัน จะถูกจัดให้อยู่ภายในกลุ่มเดียวกัน ผลลัพธ์จะเป็นพื้นที่ เช่นวิธีการทำ Intensity Threshold วิธีนี้จะทำโดยการกำหนดค่า θ เรียกว่า Brightness threshold ซึ่งมาจากการสังเกตกราฟฮิสโทแกรม และในกรณีที่เมื่อนำภาพที่มีความเข้มสีใดสีหนึ่งที่เป็นส่วนประกอบเป็นส่วนมากในภาพ มาสร้างกราฟฮิสโทแกรม จะมีวิธีแยกความชัดเจน ดังตัวอย่างเช่น สร้างกราฟฮิสโทแกรมเป็น 2 กราฟ คือกราฟที่แสดงค่าความเข้มที่เป็นส่วนประกอบส่วนมาก และกราฟที่ไม่แสดงค่าความเข้มที่เป็นส่วนประกอบส่วนมาก เพื่อให้วิธีหลังสามารถสังเกตกราฟฮิสโทแกรมของวัตถุได้ชัดเจนขึ้น เมื่อสังเกตกราฟฮิสโทแกรมแล้วสังเกตลักษณะของกราฟมาตัดสินว่าช่วงความเข้มใดเป็นวัตถุ และช่วงความเข้มใดเป็นพื้นหลัง



ก) กราฟฮิสโทแกรมแสดงทุกค่าความเข้ม



ข) กราฟฮิสโทแกรมไม่แสดงค่าความเข้ม 0
ยกตัวอย่างโดยจุดภาพที่มีค่าความเข้มน้อยกว่า 240 เป็นพื้นหลัง
และจุดภาพที่ค่าความเข้มมากกว่า 240 เป็นภาพวัตถุที่สนใจ

รูปที่ 2.4 วิธีการแยกองค์ประกอบของรูปภาพโดยใช้กราฟฮิสโทแกรม (แกนนอนคือค่าความเข้ม และแกนตั้งคือจำนวนจุดภาพ)

หลักการนี้แบ่งได้เป็น 2 วิธีดังนี้คือ

วิธีที่ 1 คือกรณีที่วัตถุมีสีอ่อนกว่าพื้นหลัง ปรับแก้ $a[m,n]$ เพื่อแยกเป็นวัตถุหรือพื้นหลังจาก

$$a[m,n] \begin{cases} 1 & \text{if } a[m,n] \geq \theta \text{ (กรณีเป็นวัตถุ)} \\ 0 & \text{otherwise (กรณีเป็นพื้นหลัง)} \end{cases}$$

สำหรับวัตถุที่มีสีเข้มกว่าพื้นหลังจะใช้วิธีที่ 2 คือ

$$a[m,n] \begin{cases} 1 & \text{if } a[m,n] < \theta \text{ (กรณีเป็นวัตถุ)} \\ 0 & \text{otherwise (กรณีเป็นพื้นหลัง)} \end{cases}$$

เมื่อ $a[m,n]$ คือ ความสัมพันธ์ระหว่างจุดบนระนาบ ความเข้ม
 m แทนตำแหน่งจุดภาพบนแนวนอน
 n แทนตำแหน่งจุดภาพบนแนวตั้ง

(2) Region Oriented Image Segmentation

Region Oriented Image Segmentation เป็นวิธีการแยกองค์ประกอบของภาพ โดยดูจากตำแหน่งของจุดภาพ และความเหมือนกันของคุณสมบัติของจุดภาพ ภายในพื้นที่ โดยถ้าจุดภาพที่อยู่ติดกันและมีคุณสมบัติเหมือนกันจะถูกจัดให้เข้ากลุ่มเดียวกัน ข้อดีของการทำเช่นนี้จะได้พื้นที่ที่ต่อเนื่อง Region Oriented Image Segmentation แบ่งเป็น 2 วิธีใหญ่ๆ คือ Region Growing และ Region Splitting and Merging โดยวิธีการแรกเป็นการนำจุดภาพที่อยู่ติดกันและมีคุณสมบัติเหมือนกันเพื่อมารวบรวมเป็นพื้นที่เดียวกัน โดยการใช้การขยายตัวของพื้นที่ ส่วนวิธีการ Region Splitting and Merging เป็นแบ่งซอยรูปภาพออกเป็นพื้นที่ย่อย โดยให้พื้นที่ย่อยแต่ละพื้นที่มี Intensity เดียวกัน จากนั้นจึงทำการรวมพื้นที่ย่อยที่มี Intensity เหมือนกันเข้าด้วยกันเป็นพื้นที่ใหญ่

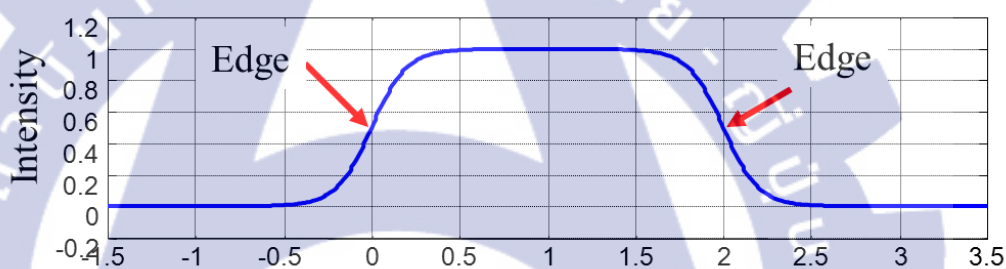
(3) Edge Oriented Image Segmentation

Edge Oriented Image Segmentation เป็นวิธีการแยกองค์ประกอบของภาพอาศัยความไม่ต่อเนื่อง (Discontinuity) ของคุณสมบัติของจุดภาพบริเวณรอยต่อระหว่างวัตถุกับฉากหลัง วิธีการนี้มุ่งที่ขอบของวัตถุเป็นหลักผลลัพธ์ที่ได้จากวิธีการนี้จะอยู่ในรูปเส้นพรมแดนระหว่างพื้นที่ต่าง ๆ ข้อได้เปรียบของวิธีการนี้ คือ ใช้ข้อมูลในการประมวลผลน้อยกว่าวิธี Region Oriented Image Segmentation เนื่องจากวิธีการนี้พิจารณาเฉพาะข้อมูลที่ขอบของวัตถุเท่านั้น

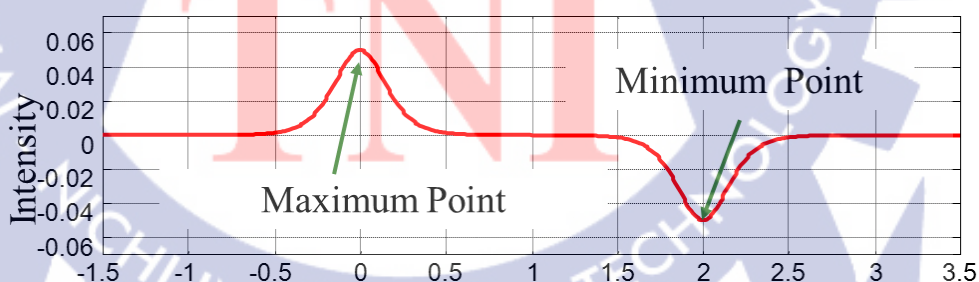
จากการศึกษาในเรื่องการแบ่งย่อยส่วนประกอบภาพ ผู้วิจัยพบว่าภาพถ่ายที่ถูกถ่ายเพื่อนำมาใช้ในการสื่อสารใดๆ ที่มีความแตกต่างของคุณสมบัติภาพ ระหว่างอวัยวะต่างๆ ของคนกับภาพพื้นหลัง จะสามารถนำมาแยกแยะภาพพื้นหลังกับภาพที่เราสนใจออกมาได้หลักการและวิธีการที่เหมาะสมในการแยกระหว่างภาพคนกับภาพพื้นหลังต่าง ๆ น่าจะเป็นวิธีการ ที่เกิดจากการวิเคราะห์คุณสมบัติของภาพ และใช้คุณสมบัติต่าง ๆ เหล่านี้มาประมวลผลและแยกแยะเพื่อนำมาใช้กับหลักการต่าง ๆ ได้

2.2.2 การหาขอบภาพ (Edge Detection)

ทฤษฎีการหาขอบของภาพ (Edge Detection Method) ที่ต้องการนำมาวิเคราะห์ผลหรือทฤษฎี Edge Detection ประกอบด้วยหลายวิธีด้วยกัน แต่อย่างไรก็ตามสามารถแบ่งได้เป็น 2 กลุ่มหลัก คือ Gradient Method และ Laplacian Method โดยในแต่ละวิธี ก็มีวิธีการหาขอบภาพที่แตกต่างกันดังนี้



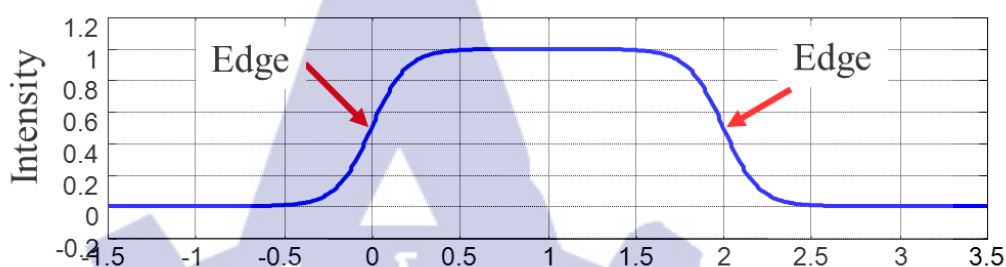
ก) สัญญาณของภาพตั้งต้น



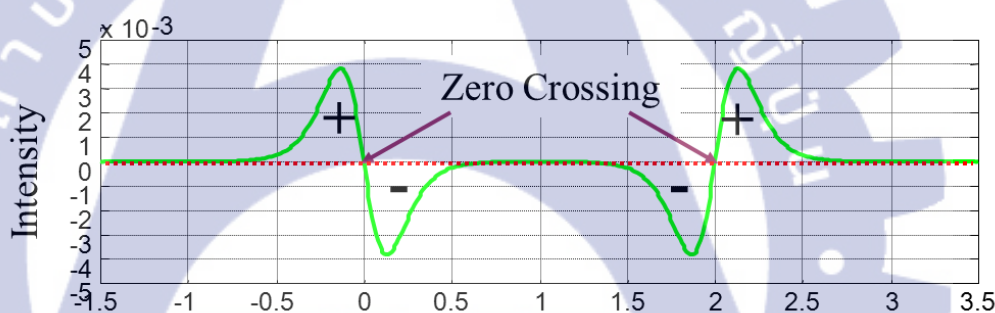
ข) อนุพันธ์อันดับหนึ่งของภาพตั้งต้น

รูปที่ 2.5 การหาขอบโดยใช้อนุพันธ์อันดับหนึ่ง

(1) วิธี Gradient method วิธีนี้เป็นการตรวจหาขอบภาพด้วยอนุพันธ์อันดับหนึ่ง (First Order Derivative) ซึ่งวิธีนี้จะหาขอบโดยการหาจุดต่ำสุดและจุดสูงสุดในรูปของอนุพันธ์อันดับหนึ่งของภาพ ซึ่งอาจทำให้เส้นขอบที่ได้มีลักษณะหนา โดยจุดที่เป็นขอบจะอยู่ในส่วนที่เหนือค่า Threshold ดังแสดงในรูปที่ 2.5 ตัวอย่างวิธีการหาขอบของกลุ่มนี้ได้แก่ Roberts, Prewitt, Sobel และ Canny เป็นต้น



ก) สัญญาณของภาพตั้งต้น

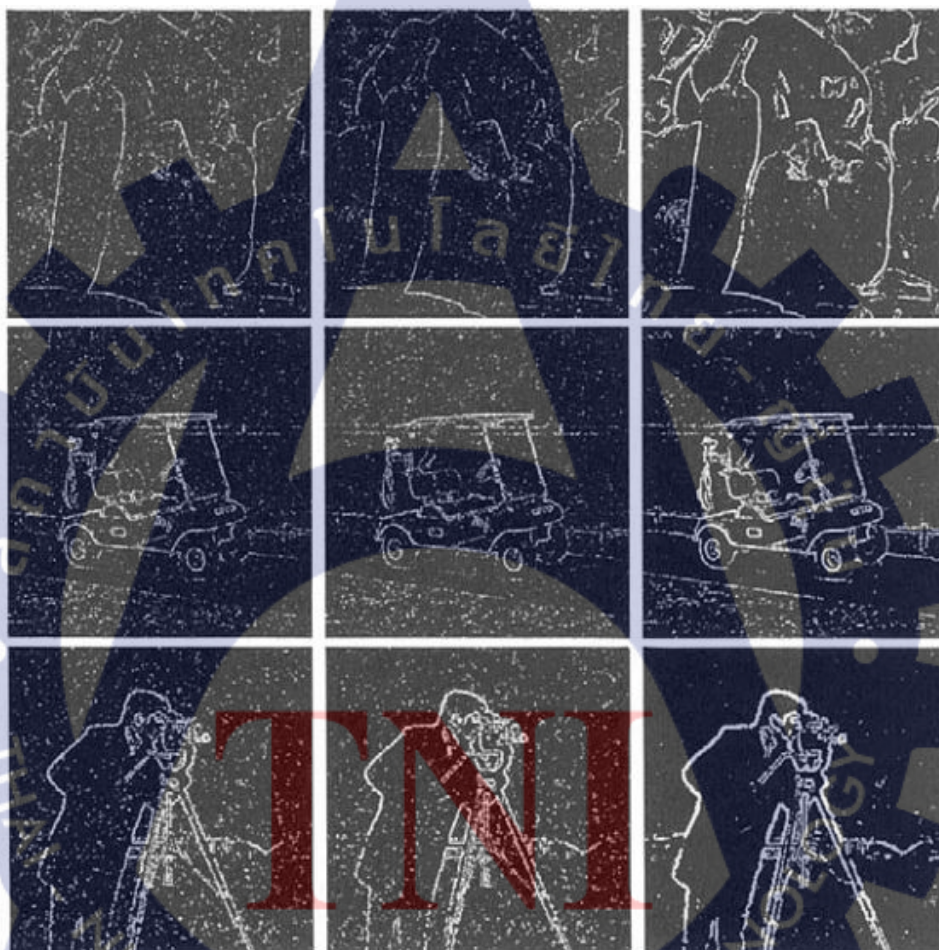


ข) อนุพันธ์อันดับสองของภาพตั้งต้น

รูปที่ 2.6 การหาขอบโดยใช้อนุพันธ์อันดับสอง

(2) วิธี Laplacian Method เป็นการหาขอบภาพ โดยการตรวจหาด้วยอนุพันธ์อันดับสอง (Second Order Derivative) จะใช้จุดที่ค่า y เป็น 0 (Zerocrossing) ดังแสดงในรูปที่ 2.6 ซึ่งวิธีนี้จะใช้เวลาในการคำนวณมากกว่า Gradient method ตัวอย่างวิธีการหาขอบของกลุ่มนี้ได้แก่ Laplacian of Gaussian และ Marrs-Hildreth เป็นต้น

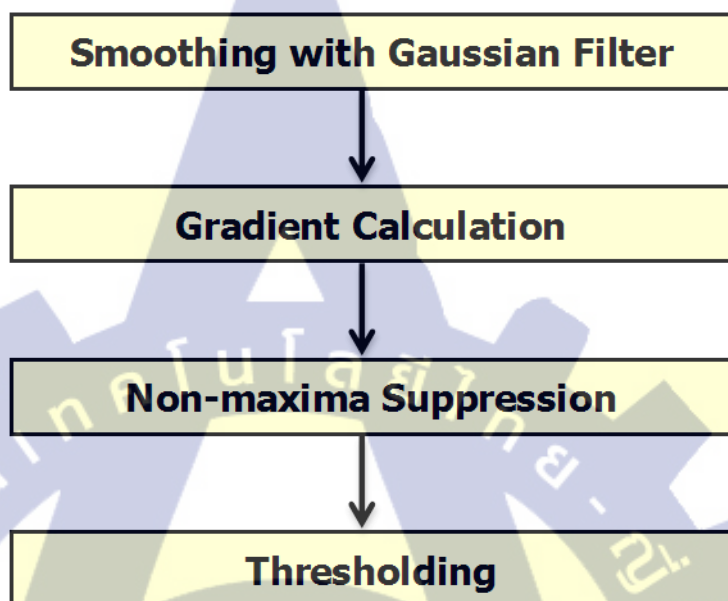
งานวิจัยนี้ได้เลือกวิธี Canny ในการหาขอบภาพ เนื่องจากวิธีดังกล่าวมีการใช้ Gaussian Filter ก่อนการหาขอบจึงสามารถควบคุมระดับความละเอียดของขอบที่ต้องการและสามารถลดสัญญาณรบกวนได้อีกด้วย อีกทั้งยังมีการประมวลผลที่เร็ว และด้วยวิธี Canny จะให้รายละเอียดภายในวัตถุได้ดีที่สุดเมื่อเปรียบเทียบกับวิธีอื่น ๆ เช่น Sobel และ Prewitt และวิธีของ Canny ยังสามารถใช้ได้ในกรณีที่ความแตกต่างของสีมีน้อยเมื่อเทียบกับวิธีอื่น ๆ



รูปที่ 2.7 ตัวอย่างการเปรียบเทียบผลของการหาขอบภาพด้วยวิธีต่าง ๆ (ซ้าย) Sobel, (กลาง) Prewitt, (ขวา) Canny

2.2.3 ขั้นตอนการหาขอบภาพโดยวิธีของ Canny

ขั้นตอนการหาขอบภาพโดยวิธีของ Canny ประกอบด้วย 4 ขั้นตอน ดังรูปที่ 2.8



รูปที่ 2.8 ขั้นตอนวิธีการหาขอบภาพโดยวิธี Canny Edge Detection

ขั้นตอนวิธีการหาขอบภาพของ Canny Edge Detection นั้นเริ่มต้นจากการปรับภาพให้เรียบ (Smoothing) ด้วยตัวกรองเกาส์เซียน (Gaussian filter) เพื่อกำจัดสัญญาณรบกวน หลังจากนั้นจะคำนวณค่าขนาด (Magnitude) และทิศทาง (Orientation) ของ Gradient โดยใช้การหาอนุพันธ์อันดับหนึ่ง จากนั้นใช้ Nonmaxima Suppression กับขนาด (Magnitude) ของ Gradient เพื่อให้ได้ขอบที่บางลง และในขั้นตอนสุดท้ายจะใช้ Double Thresholding เพื่อหาพิกเซลที่เป็นขอบและทำการเชื่อมต่อขอบโดยในแต่ละขั้นตอน มีรายละเอียดย่อยดังต่อไปนี้

(1) Smoothing

ในขั้นตอนแรกของการหาขอบโดยวิธีนี้จะต้องกำจัดสัญญาณรบกวนออกก่อนโดยใช้ Gaussian Filter ซึ่งสามารถคำนวณได้จากการใช้กรอบ (Mask) ขนาดเล็กขนาดของ Gaussian Mask นี้หากมีขนาดกว้างจะมีผลทำให้ลดสัญญาณรบกวนได้มาก แต่ถ้ากว้างมากเกินไปจะมีผลทำให้ขอบย่อย ๆ ที่เป็นส่วนรายละเอียดนั้นหายไปสำหรับการคำนวณหาภาพที่ได้จากการใช้ Gaussian Filter

(2) Gradient Calculation

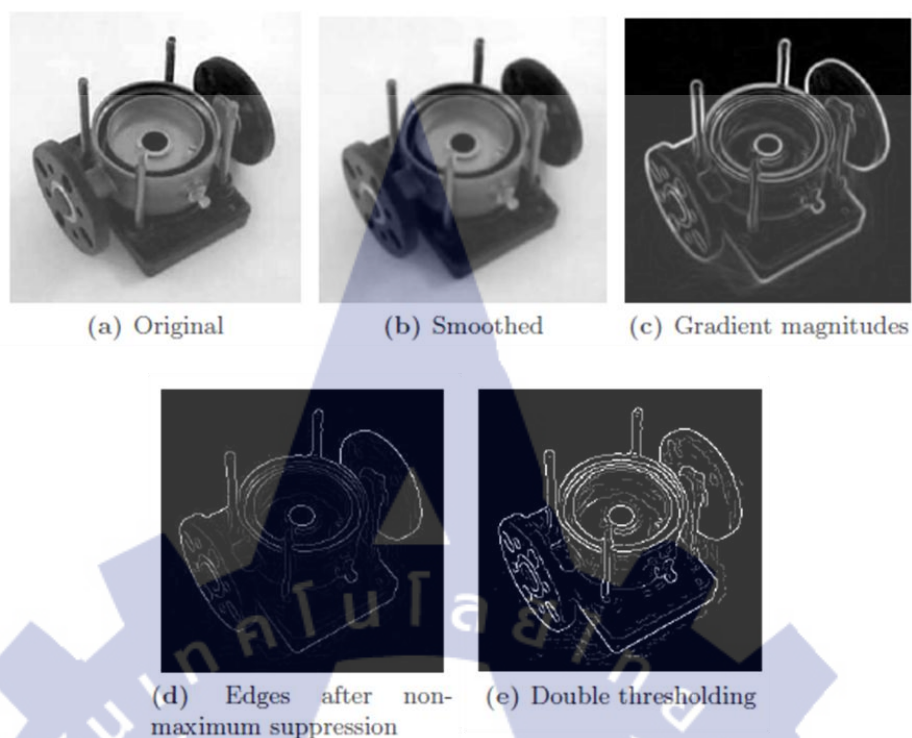
นำ Smoothing Image มาสร้าง x, y Partial Derivatives หลังจากนั้นนำค่า x, y Partial Derivatives มาคำนวณด้วยสูตรมาตรฐานสำหรับการแปลงรูปแบบจาก Rectangular ไปเป็น Polar (Rectangular-to-Polar Conversion) เพื่อหาขนาดและทิศทางของ Gradient

(3) Nonmaxima Suppression

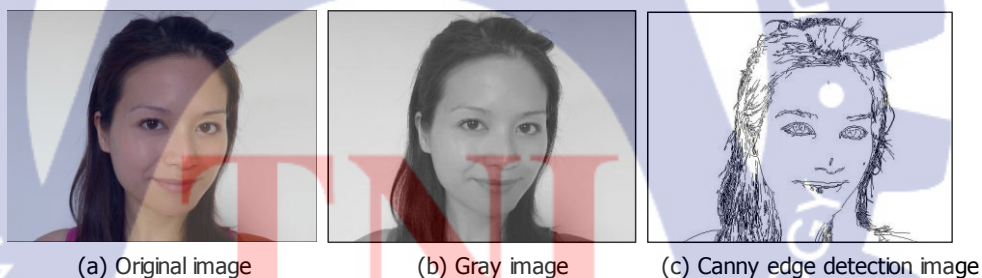
สำหรับการหาขอบโดย Canny method จุดที่ถือเป็นเส้นขอบได้นั้นต้องเป็นจุดที่ให้ค่าสูงสุดเฉพาะที่และเป็นทิศทางเดียวกับ Gradient ด้วยซึ่งด้วยวิธีดังกล่าวนี้ทำให้ได้ขอบที่บางเพียง 1 พิกเซล ภาพที่ได้หลังการทำ Nonmaxima Suppression จะให้ค่าเป็นศูนย์ในทุกจุด ยกเว้นจุดที่เป็น Local Maxima Points ซึ่งจะยังคงค่าเดิมไว้

(4) Thresholding

แม้ว่าภาพจะผ่านการ Smoothing ในขั้นตอนแรกแล้วก็ตาม ภาพที่ได้ก็ยังมีส่วนที่มีเส้นขอบที่ไม่ใช่ขอบที่แท้จริงปรากฏอยู่อันเนื่องจากสัญญาณรบกวนหรือลักษณะของวัตถุในภาพเป็นพื้นผิวที่มีลวดลายหรือมีรายละเอียดภายในมาก ดังนั้นเพื่อลดปัญหาดังกล่าวจึงได้มีการกำหนดค่า Threshold ขึ้นมา 2 ค่าคือ High Threshold (T_1) และ Low Threshold (T_2) โดยพิกเซลที่มีค่ามากกว่า T_1 จะถูกปรับเป็น 1 (เป็นพิกเซลที่เป็นขอบ) แต่ถ้าน้อยกว่า T_2 จะถูกปรับเป็น 0 ส่วนค่าที่อยู่ระหว่างค่า Threshold ทั้งสองการปรับเป็นค่า 0 หรือ 1 นั้นขึ้นอยู่กับพิกเซลที่อยู่รอบข้างหากพบว่าพิกเซลที่อยู่รอบข้างของพิกเซลที่เป็นขอบ (ค่า $> T_1$) มีค่ามากกว่า T_2 แล้วจะปรับค่าพิกเซลดังกล่าวให้มีค่าเป็น 1 และถือเป็นหนึ่งในขอบภาพด้วยเช่นกัน



รูปที่ 2.9 ตัวอย่างภาพของแต่ละลำดับขั้นตอนการหาขอบภาพด้วยวิธี Canny



รูปที่ 2.10 ตัวอย่างการหาขอบภาพด้วยวิธีของ Canny กับรูปใบหน้าคน

2.3 การบีบอัดข้อมูล (Data Compression)

การบีบอัดข้อมูล เป็นสาขาวิชาหนึ่งในวิทยาการคอมพิวเตอร์ หมายถึงการศึกษาวิธีการในการจัดเก็บข้อมูลที่ทำให้ใช้เนื้อที่ในการจัดเก็บน้อยลง

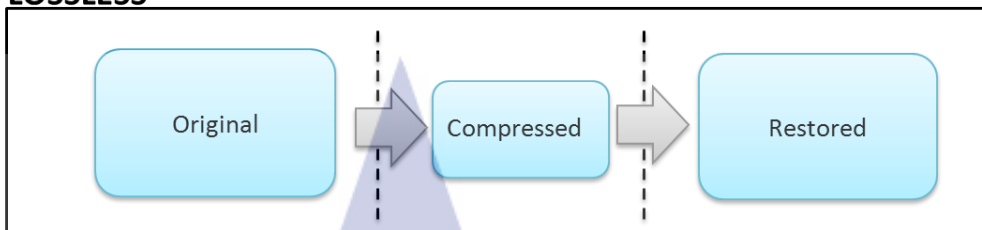
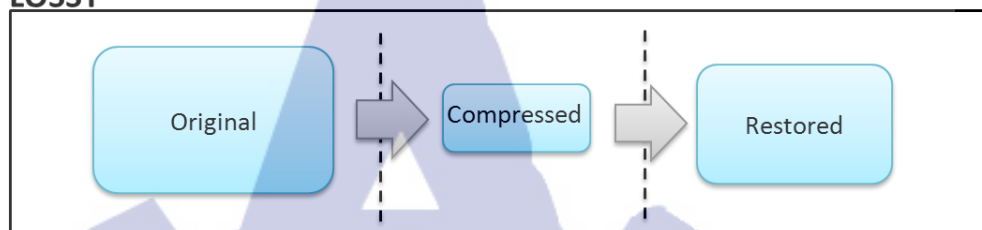
การบีบอัดข้อมูล มีความสำคัญในระบบการสื่อสารและจัดเก็บข้อมูล เนื่องจากทำให้เก็บหรือรับส่งข้อมูลได้มากขึ้น โดยใช้เนื้อที่เท่าเดิม การบีบอัดข้อมูลจะช่วยให้ข้อมูลที่จะถูกส่งออกไปมีขนาดเล็กลง และเป็นการเพิ่มอัตราในการส่งข้อมูลให้สูงขึ้น โดยการจัดการใน

ลักษณะนี้ถูกเรียกว่า Source Coding มีด้วยกัน 2 ลักษณะคือ Lossy Compression และ Lossless Compression โดยแบบแรกเป็นพวกบีบอัดแล้วข้อมูลเสียหายก็คือเมื่อทำการ Decompression แล้วไม่สามารถนำข้อมูลทั้งหมดกลับมาได้ เช่น JPEG, MPEG เป็นต้น และพวกที่สองเป็นพวกบีบอัดแล้วข้อมูลไม่หายไป หมายความว่าสามารถนำข้อมูลที่บีบอัดกลับมาเป็นก่อนไม่บีบอัดได้ การทำงานในลักษณะนี้นิยมใช้สำหรับข้อมูลที่มีความสำคัญ แต่ก็เปลืองทรัพยากรที่ใช้มากกว่าแบบแรก โดยเทคนิคพวกนี้ได้แก่ Run-length, Huffman, Delta และ LZW เป็นต้น

2.3.1 ประเภทของการบีบอัดข้อมูล

(1) การบีบอัดข้อมูลแบบ Lossless วิธีนี้จะอาศัยหลักการที่ว่าปกติข้อมูลที่ใช้อยู่ มักจะมีข้อมูลที่ซ้ำกัน เช่น คำในภาษาอังกฤษมักพบอักษร 'e' ได้บ่อยกว่า 'z' และความเป็นไปได้ที่อักษร 'q' จะตามด้วย 'z' มีน้อยมาก ๆ ยกตัวอย่างในภาษาอังกฤษในบทความหรือหนังสือเล่มหนึ่งจะปรากฏคำว่า the อยู่บ่อยมาก สมมติให้เก็บอักษร t แทน the หรืออาจใช้วิธีเก็บตำแหน่งที่ปรากฏคำนั้น ๆ แทน ก็สามารถลดความยาวข้อมูลที่จะเก็บได้ จะเห็นว่าการบีบอัดข้อมูลแบบนี้ ข้อมูลต้นฉบับกับข้อมูลที่บีบอัดแล้วคลายออกมาจะเหมือนกันไม่ผิดเพี้ยน

(2) การบีบอัดข้อมูลแบบ Lossy จะมีแนวคิดต่างกันไป โดยใช้หลักว่าความผิดเพี้ยนของข้อมูลเล็กน้อยเป็นสิ่งที่ยอมรับได้ เช่น ตาของมนุษย์ไม่สามารถแยกความแตกต่างของสีบางสีได้ทั้งหมด จึงไม่จำเป็นต้องเก็บข้อมูลทุกสีทุกตำแหน่ง เก็บเพียงบางสีที่แยกความแตกต่างได้ก็พอ ดังจะเห็นตัวอย่างจากไฟล์ประเภท .jpg ที่ใช้การบีบอัดข้อมูลแบบ Lossy ซึ่งจะทำให้ได้ขนาดไฟล์ภาพที่เล็กลงมาก แต่ก็สูญเสียรายละเอียดบางอย่างไป (โดยใช้หลักแนวคิดที่ว่า เป็นส่วนของสีที่มนุษย์ไม่สามารถแยกแยะความแตกต่างได้) เห็นว่าการบีบอัดข้อมูลแบบ Lossy มักจะใช้กับข้อมูลที่มนุษย์รับรู้ อีกตัวอย่างหนึ่งคือ ไฟล์เสียงประเภท .mp3 ซึ่งทำการตัดเสียงในย่านความถี่ที่มนุษย์ไม่สามารถได้ยินออกไป ยกตัวอย่างง่าย ๆ เพื่อเทียบการบีบอัดข้อมูลแบบ Lossless กับ Lossy เช่น 25.8888888888 อาจบีบอัดให้เหลือ 25.[9]8 (9 ในปีกกา หมายถึงมีเลข 8 จำนวน 9 ตัว) ซึ่งสามารถแปลงกลับเป็นข้อมูลเดิมได้ทั้งหมดแบบไม่ผิดเพี้ยน (Lossless) ในขณะที่ถ้าหากสามารถยอมรับได้ว่าข้อมูลผิดเพี้ยนไปบ้างก็ไม่เป็นไรก็อาจบีบอัดให้เหลือ 26 ซึ่งจะเห็นว่าประหยัดพื้นที่การเก็บข้อมูลได้เยอะกว่า แต่ขณะเดียวกันข้อมูลจะไม่เหมือนเดิม และไม่สามารถเรียกคืนกลับมาได้ (Lossy)

LOSSLESS**LOSSY**

รูปที่ 2.11 การบีบอัดข้อมูลแบบ Lossless และแบบ Lossy

การออกแบบงานวิจัยนี้ ผู้วิจัยได้เลือกที่จะใช้ Lossless Compression เนื่องจากข้อมูลที่จะถูกใช้แสดงผล จะต้องเป็นข้อมูลที่มีความสมบูรณ์ จึงต้องการการบีบอัดที่แบบไม่สูญเสียข้อมูลไป โดยในนี้การบีบอัดแบบ Lossless ก็มีอยู่หลายรูปแบบ ซึ่งในปัจจุบันรูปแบบที่ใช้แพร่หลายก็จะเป็น Run-Length, Huffman และ LZW ดังนั้นจึงได้ทำการเปรียบเทียบ 3 แบบดังกล่าวเป็นดังต่อไปนี้

(1) Run-Length เป็นการบีบอัดข้อมูลที่ง่ายที่สุด ซึ่งนับความยาวของข้อมูลที่เหมือนกัน และติดกันยกตัวอย่างดังนี้คือ ข้อมูล “000001111001101” จะเป็น “501420211011” โดยเทคนิคนี้เหมาะสมกับข้อมูลที่มีความซ้ำ ๆ กันมาก เช่นพวก Icon หรือ ลายเส้นแบบง่าย ๆ เป็นต้น แต่เทคนิคนี้บางครั้งเมื่อบีบอัดแล้ว อาจได้ข้อมูลที่มีขนาดใหญ่กว่าข้อมูลจริงก็ได้ ทำให้เกิดความไม่แน่นอนในการบีบอัดมากนัก

(2) Huffman Code เทคนิคนี้ได้รับความนิยมใช้อย่างแพร่หลายเนื่องจากมีประสิทธิภาพในการบีบอัดสูง ข้อมูลจะถูกแตกออกเป็นในลักษณะของกราฟต้นไม้โดยแต่ละกิ่งจะมี Code ที่เรียกว่า Pre-Fix Code อยู่ และเหมาะกับข้อมูลที่มีค่าน้ำหนักเช่น ข้อความหรือประโยค โดยจะมีการให้น้ำหนักกับตัวอักษร (สระในภาษาอังกฤษจะมีโอกาสอยู่ในประโยคได้มากกว่าตัวอักษรตัวอื่น ๆ ดังนั้นจะมีค่าน้ำหนักสูงกว่า) เป็นต้น และใช้วิธีการ Entropy Encoding มาใช้ในการคำนวณ สุดท้ายขนาดของข้อมูลจะมีขนาดที่เล็กลงตามค่าน้ำหนักที่จะคำนวณออกมา แต่ข้อเสียที่เกิดจากวิธีนี้คือการที่จะต้องรู้ค่าน้ำหนักเพื่อใช้ในการคำนวณ ซึ่งในทางปฏิบัติบางครั้งจะไม่มีทางรู้ค่าเหล่านี้ได้เลย

(3) LZW (Lempel-Ziv) Code เป็นเทคนิคที่ใช้กันแพร่หลายที่สุด โดยซอฟต์แวร์ตามท้องตลาดก็นำวิธีนี้ไปใช้เช่น WinZip เป็นต้น เนื่องด้วยเทคนิคนี้จะใช้วิธีสร้างดิกชันนารีขึ้นมาเก็บข้อมูล ถ้าเป็นข้อมูลใหม่ก็จะเก็บเอาไว้ในฐานข้อมูลไปเรื่อยๆ และแทนที่ด้วยรหัสแบบไบนารีซึ่งถ้าเจอข้อมูลที่ซ้ำก็จะแทนที่ได้เลย โดยวิธีนี้สิ่งที่ต้องคำนึงก็คือขนาดของดิกชันนารีว่าจะมีขนาดเท่าไร และไปสัมพันธ์กับวิธีการค้นหาข้อมูลในฐานข้อมูลนั้นก็คือความเร็วในการสืบค้นข้อมูลนั่นเอง โดยปกติขนาดของดิกชันนารีจะใช้ขนาด 12 บิตข้อมูลหรือสามารถสืบค้นข้อมูลได้ 4,096 ตัว แต่จะสามารถปรับเปลี่ยนขนาดได้ ไม่จำเป็นต้องยึดกับค่านี้นั้น ดังนั้นข้อเสียหลักก็คือขนาดของไฟล์ที่ใช้ในการบีบอัดจะต้องมีขนาดใหญ่พอสมควร ถ้าบีบอัดข้อมูลไม่มากก็จะทำให้ขนาดไฟล์ลดลงได้ไม่มาก ตามที่จะเห็นได้ในซอฟต์แวร์ WinZip เมื่อบีบอัดไฟล์ที่มีขนาดเล็กก็จะได้ไม่มากเท่าไร

ตารางที่ 2.1 การเปรียบเทียบผลการบีบอัดที่ได้จากแต่ละวิธี

	RLE	Huffman	LZW
1	-0.71	37.42	38.24
2	1.25	38.32	43.78
3	-0.13	32.6	30.7
4	11.39	41.7	47.98
5	11.79	41.94	69.03
6	3.91	41.07	44.35
7	-0.4	37.38	50.39
8	0.54	42.24	55.85
9	0.11	39.15	
10	0.53	37.71	49.31

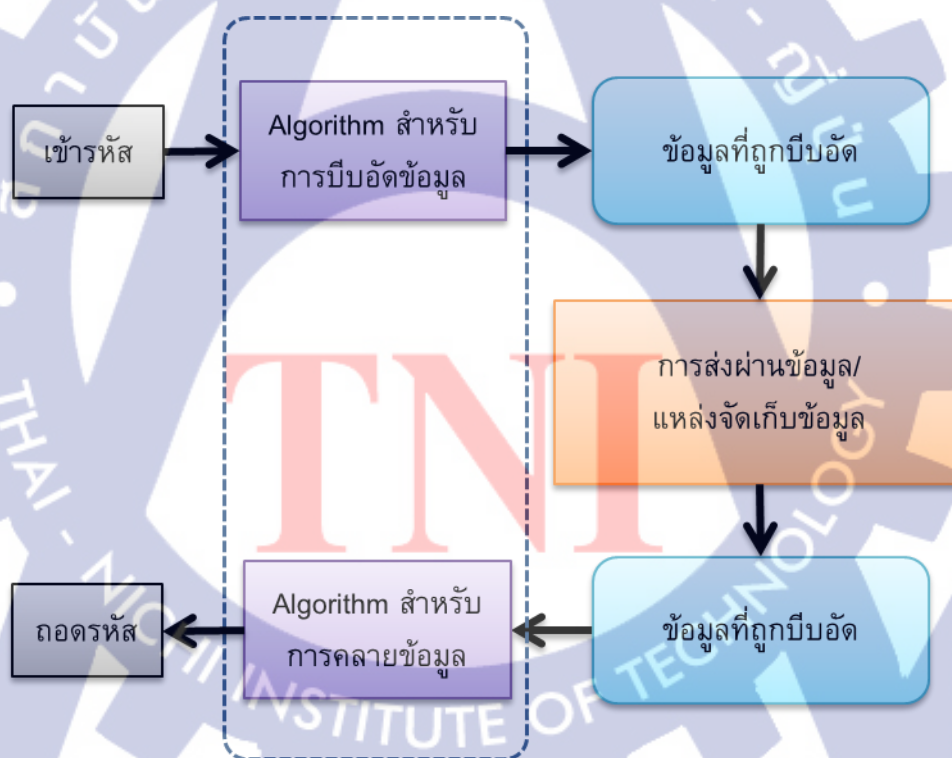
จากตารางที่ 2.1 แสดงการเปรียบเทียบผลร้อยละของข้อมูลที่ลดลง ของการบีบอัดที่ได้จากวิธี Run-Length, Huffman และ LZW (Lempel-Ziv) ที่ทดลองกับข้อมูลที่เป็นตัวอักษร พบว่าวิธี Huffman และ LZW (Lempel-Ziv) นั้น ให้ผลการบีบอัดที่ดีกว่าวิธี Run-Length

วิธีที่ได้กล่าวมาข้างต้น เป็นแค่ข้อดีและข้อเสียของการบีบอัดในแต่ละแบบที่นิยมใช้ในเชิงพาณิชย์ และยังมีการบีบอัดแบบอื่น ๆ ที่ใช้กันอยู่ไม่ว่าจะเป็นมาตรฐาน CCSDS (เป็นมาตรฐานที่ใช้กับระบบการสื่อสารดาวเทียม) หรืองานวิจัยอื่นๆ ที่พัฒนาขึ้นมาใช้เฉพาะงานเหล่านั้น แต่ที่ได้กล่าวมาทั้งหมดการเลือกใช้การบีบอัดข้อมูลในเบื้องต้น ขึ้นอยู่กับว่าระยะเวลา

ในการพัฒนาต้นแบบและงบประมาณ โดยจะมีอยู่ 2 แนวทาง แนวทางแรกคือ จะสามารถหาเทคนิคที่ไม่จำเป็นต้องเสียเงินสำหรับจ่ายค่าลิขสิทธิ์ และเป็นไลบรารีที่มีอยู่แล้ว เพื่อความสะดวกและรวดเร็วในการพัฒนาต้นแบบ หรืออีกแนวทางหนึ่งก็คือทำการพัฒนาเทคนิคการบีบอัดขึ้นมาเอง โดยอาจประยุกต์จากองค์ความรู้ที่มีอยู่ แต่ในกรณีนี้ก็จะเสียเวลาในการพัฒนาต้นแบบไปด้วย ดังนั้นในขั้นแรกของการออกแบบซอฟต์แวร์นี้ ผู้วิจัยจะดำเนินการไปในแนวทางโดยหาไลบรารีที่มีอยู่แล้ว ซึ่งจะไม่ทำให้ใช้เวลาในการพัฒนามากนัก

งานวิจัยนี้ใช้วิธีการบีบอัดข้อมูลแบบ Lossless Compression โดยใช้หลักการของ Zlib Deflate ซึ่งจะอาศัยการรวมกันของ Huffman Code และ Lempel–Ziv 1977 (LZ77) ที่ออกแบบมาแทนที่ Lempel–Ziv–Welch (LZW) จะมีข้อดีตรงที่สามารถบีบอัดได้อย่างอิสระ มีประสิทธิภาพสูง เก็บไฟล์หลายไฟล์ในตัวเองโดยไม่ต้องพึ่งโปรแกรมอื่น และยังสามารถใช้ข้อมูลที่ซ้ำซ้อนระหว่างไฟล์มาเป็นประโยชน์ในการบีบอัดได้อีกด้วย

การออกแบบโปรแกรมในส่วนการบีบอัดข้อมูลภาพนี้ใช้หลักการคร่าว ๆ ดังรูปที่ 2.12



รูปที่ 2.12 การเข้าและถอดรหัสการบีบอัดข้อมูล

2.4 การส่งข้อมูลผ่านระบบเครือข่ายคอมพิวเตอร์

ระบบเครือข่ายคอมพิวเตอร์ หมายถึง การเชื่อมต่อคอมพิวเตอร์ตั้งแต่ 2 เครื่องขึ้นไป เข้าด้วยกันด้วยสายเคเบิล หรือสื่ออื่น ๆ ทำให้คอมพิวเตอร์สามารถรับส่งข้อมูลแก่กันและกันได้

ในกรณีที่เป็นการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์หลายๆ เครื่องเข้ากับเครื่องคอมพิวเตอร์ขนาดใหญ่ที่เป็นศูนย์กลาง เราเรียกคอมพิวเตอร์ที่เป็นศูนย์กลางนี้ว่า โฮสต์ (Host) และเรียกคอมพิวเตอร์ขนาดเล็กที่เข้ามาเชื่อมต่อว่าไคลเอนต์ (Client)

ระบบเครือข่าย (Network) จะเชื่อมโยงคอมพิวเตอร์เข้าด้วยกันเพื่อการติดต่อสื่อสาร เราสามารถส่งข้อมูลภายในอาคาร หรือข้ามระหว่างเมืองไปจนถึงอีกจุดหนึ่งของโลก ซึ่งข้อมูลต่างๆ อาจเป็นทั้งข้อความ, รูปภาพ และเสียง ก่อให้เกิดความสะดวกรวดเร็วแก่ผู้ใช้ ซึ่งความสามารถเหล่านี้ ทำให้เครือข่ายคอมพิวเตอร์มีความสำคัญ และจำเป็นต่อการใช้งาน ในแวดวงต่าง ๆ

2.4.1 โพรโทคอล (Protocol)

โพรโทคอล (Protocol) คือระเบียบวิธีการในการติดต่อสื่อสาร เมื่อมาใช้กับเทคโนโลยีสื่อสารโทรคมนาคม จึงหมายถึงขั้นตอนการติดต่อสื่อสาร ซึ่งรวมถึง กฎ ระเบียบ และข้อกำหนดต่าง ๆ รวมถึงมาตรฐานที่ใช้ เพื่อให้ตัวรับและตัวส่งสามารถดำเนินกิจกรรมทางด้านสื่อสารได้สำเร็จ

การที่คอมพิวเตอร์เครื่องหนึ่งจะส่งข้อมูลไปยังคอมพิวเตอร์อีกเครื่องหนึ่งได้นั้น จะต้องอาศัยกลไกหลาย ๆ อย่างร่วมกันทำงานต่างหน้าที่กันและเชื่อมต่อเป็นเครือข่ายเข้าด้วยกัน ปัญหาที่เกิดขึ้นคือการเชื่อมต่อมีความแตกต่างระหว่างระบบและอุปกรณ์ หรือเป็นผู้ผลิตคนละรายกัน ซึ่งเป็นสิ่งที่ทำให้การสร้างเครือข่ายเป็นเรื่องยากมาก เนื่องจากขาดมาตรฐานกลางที่จำเป็นในการเชื่อมต่อ จึงได้เกิดหน่วยงานกำหนดมาตรฐานสากลขึ้นคือ International Standards Organization ขึ้นและทำการกำหนดโครงสร้างทั้งหมดที่จำเป็นต้องใช้ในการสื่อสารข้อมูลและเป็นระบบเปิด เพื่อให้ผู้ผลิตต่าง ๆ สามารถแยกผลิตในส่วนที่ตัวเองถนัด แต่สามารถนำไปใช้ร่วมกันได้ ระบบเครือข่ายคอมพิวเตอร์สมัยใหม่จะถูกออกแบบให้มีโครงสร้างที่แน่นอน และเพื่อเป็นการลดความซับซ้อน ระบบเครือข่ายส่วนมากจึงแยกการทำงานออกเป็นชั้นๆ (Layer) โดยกำหนดหน้าที่ในแต่ละชั้นไว้อย่างชัดเจน

2.4.2 การนำส่งข้อมูล

ชั้นสื่อสารนำส่งข้อมูล (Transport Layer) แบ่งโพรโทคอลเป็น 2 ชนิดตามลักษณะการทำงาน ได้แก่

(1) UDP (User Datagram Protocol) เป็นโพรโทคอลที่อยู่ใน Transport Layer โดยการส่งข้อมูลของ UDP นั้นจะเป็นการส่งครั้งละ 1 ชุดข้อมูล เรียกว่า UDP Datagram ซึ่งจะ

ไม่มีความสัมพันธ์กันระหว่าง Datagram และจะไม่มีกลไกการตรวจสอบความสำเร็จในการรับส่งข้อมูล กลไกการตรวจสอบโดย Checksum ของ UDP นั้นเพื่อเป็นการป้องกันข้อมูลที่อาจจะถูกแก้ไข หรือมีความผิดพลาดระหว่างการส่ง และหากเกิดเหตุการณ์ดังกล่าวปลายทางจะรู้ว่าข้อมูลผิดพลาดเกิดขึ้น แต่มันจะเป็นการตรวจสอบเพียงฝ่ายเดียวเท่านั้น โดยในข้อกำหนดของ UDP หากพบว่า Checksum Error ก็ให้ผู้รับปลายทางทำการทิ้งข้อมูลนั้น แต่จะไม่มีการแจ้งกลับไปยังผู้ส่งแต่อย่างใด การรับส่งข้อมูลแต่ละครั้งหากเกิดข้อผิดพลาดในระดับ IP เช่น ส่งไม่ถึงผู้รับ หรือหมดเวลา ผู้ส่งจะได้รับ Error Message จากระดับ IP เป็น ICMP (Internet Control Message Protocol) Error Message แต่เมื่อข้อมูลส่งถึงปลายทางถูกต้องแต่เกิดข้อผิดพลาดขึ้น ในส่วนของ UDP เอง จะไม่มีการยืนยันหรือแจ้งให้ผู้ส่งทราบแต่อย่างใด

(2) TCP (Transmission Control Protocol) อยู่ใน Transport Layer เช่นเดียวกับ UDP ซึ่ง TCP จะทำหน้าที่จัดการและควบคุมการรับส่งข้อมูล ที่มีความสามารถและรายละเอียดมากกว่า UDP โดย Datagram ของ TCP จะมีความสัมพันธ์ต่อเนื่องกัน และมีการกลไกควบคุมการรับส่งข้อมูลให้มีความถูกต้อง (Reliable) และมีการสื่อสารอย่างเป็นทางการ (Connection-Oriented)

กล่าวโดยสรุปคือ การนำส่งข้อมูลแบบ UDP (User Datagram Protocol) เป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless) มีการตรวจสอบความถูกต้องของข้อมูลแต่จะไม่มีการแจ้งกลับไปยังผู้ส่ง จึงอาจถือได้ว่าไม่มีการตรวจสอบความถูกต้องของความสำเร็จในการรับส่งข้อมูล อย่างไรก็ตาม วิธีการนี้มีข้อดีในด้านความเร็วในการส่งข้อมูล จึงนิยมใช้ในระบบผู้ให้บริการ (Client/Server System) ซึ่งมีการสื่อสารแบบ ถาม/ตอบ (Request/Reply) นอกจากนี้ยังใช้ในการส่งข้อมูลประเภทภาพเคลื่อนไหว หรือการส่งเสียง (Voice) ทางอินเทอร์เน็ต

แต่ TCP (Transmission Control Protocol) เป็นแบบที่มีการกำหนดช่วงการสื่อสารตลอดระยะเวลาการสื่อสาร (Connection-Oriented) ซึ่งจะยอมให้มีการส่งข้อมูลเป็นแบบ Byte Stream ที่ไว้วางใจได้โดยไม่มีข้อผิดพลาด ข้อมูลที่มีปริมาณมากจะถูกแบ่งออกเป็นส่วนเล็กๆ เรียกว่า Message ซึ่งจะถูกส่งไปยังผู้รับ ผ่านทางชั้นสื่อสารของอินเทอร์เน็ต ทางฝ่ายผู้รับจะนำ Message มาเรียงต่อกันตามลำดับเป็นข้อมูลตัวเดิม TCP ยังมีความสามารถในการควบคุมการไหลของข้อมูลเพื่อป้องกันไม่ให้ผู้ส่ง ส่งข้อมูลเร็วเกินกว่าที่ผู้รับจะทำงานได้ทันอีกด้วย

ในปัจจุบัน TCP/IP เป็นโพรโทคอลชนิดหนึ่งที่ได้รับการนิยมนำมาใช้ทั้งในเครือข่าย Internet LAN และ WAN ในการเชื่อมโยงระบบคอมพิวเตอร์ที่มีความแตกต่างกันไม่ว่าจะเป็นระบบ UNIX, OS/2 MS-DOS, มินิคอมพิวเตอร์ หรือเมนเฟรม และระบบปฏิบัติการหลายชนิดในระบบ LAN เช่น Netware 4.X และ 5.X , VINES และ LAN Manager ก็ออกแบบมาให้สามารถรองรับโพรโทคอล TCP/IP ได้ แม้ว่าส่วน ใหญ่จะถูกนำมาใช้กับเครือข่าย LAN แบบ

Ethernet แต่ TCP/IP ก็สามารถนำมาใช้กับ LAN แบบ Token-Ring และแบบ ARCnet ได้เช่นกัน ซึ่งแทบจะ เรียกได้ว่าโพรโทคอลตัวนี้เป็นตัวหลัก และสำคัญมากที่สุดตัวหนึ่งในปัจจุบัน

งานวิจัยนี้จึงได้ให้ความสนใจไปที่ TCP/IP เนื่องจากเป็นโพรโทคอลที่มีการใช้งานบนเครือข่ายอินเทอร์เน็ต และอินเทอร์เน็ตทั่วไปในปัจจุบัน และที่สำคัญในการส่งข้อมูลผ่านทาง TCP/IP นั้นจะสามารถป้องกันความผิดพลาดของการรับส่งได้ โดยจะทำการแบ่งข้อมูลนั้น ๆ ออกเป็นส่วนย่อย ๆ ซึ่งเรียกว่า Packet โดยแต่ละส่วนจะถูกเพิ่มข้อมูลบอกตำแหน่งต้นทางและปลายทางที่จะส่งไว้ให้ จากนั้น Packet เหล่านี้จะถูกส่งกระจายผ่านไปยังเส้นทางต่าง ๆ ที่เชื่อมโยงกันในระบบตามเส้นทางที่สามารถส่งถึงปลายทางได้ โดยแต่ละ Packet ไม่จำเป็นต้องเรียงลำดับหรือไปตามเส้นทางเดียวกัน ซึ่งในระบบจะมีอุปกรณ์ที่เรียกว่า Router จะเป็นตัวคอยจัดหาเส้นทางที่ดีที่สุดให้กับทุก Packet และเมื่อถูกส่งไปถึงผู้รับที่ปลายทางแล้วข้อมูลเหล่านี้จะถูกนำมารวมกันเป็นข้อความยาว ๆ เหมือนเดิม แต่ถ้า Packet ใดเกิดการขาดหายหรือตกหล่น คอมพิวเตอร์ก็จะตรวจสอบ และส่ง Packet มาใหม่ จนทำให้ข้อมูลครบเหมือนเดิม



ก) การบริการแบบ Connectionless ของโพรโทคอล UDP

ไม่มีสัญญาณตอบรับ (ACK)



ข) การบริการแบบ Connection-oriented ของโพรโทคอล TCP

รูปที่ 2.13 เปรียบเทียบระหว่างโพรโทคอล UDP และโพรโทคอล TCP

2.5 ประวัติและเทคโนโลยีที่เกี่ยวข้อง

2.5.1 ประวัติและเทคโนโลยีที่เกี่ยวข้องกับการแบ่งส่วนภาพ

งานวิจัยที่เกี่ยวข้องกับการตรวจจับความแตกต่างและความไม่ต่อเนื่องของความเข้มสีของภาพ ได้ถูกนำเสนอในหลายปีที่ผ่านมา ตั้งแต่ โรเบิร์ต (Roberts. 1963) และหลังจากนั้น ฟรีวิท (Prewitt. 1970), คริส (Kirsh. 1971), โรบินสัน (Robinson. 1976), เฟร; และเซน (Frei; and Chen. 1977), และแคนนี่ (Canny. 1986) ได้นำเสนอแนวคิดต่างๆ ที่ใช้หลักการของ Point, Line, and Edge Detection มาใช้ในการประมวลผลภาพในลักษณะต่างๆ

งานวิจัยของ เชียน; และ ฮวง (Qian; and Huang. 1996) และ ซอง หวัง; เฟิง เกอ; และ ตี๋ เจิง ลีว (Song Wang; Feng Ge; and Tiecheng Liu. 2006) นำเสนอวิธีการแบ่งส่วนภาพโดยสำหรับเทคนิคของ Edge Detection

งานวิจัยของ สเวน ลอนคาสิก; และ โควาซีวิก (Sven Loncaric; and Kovacevic. 1997) ได้แบ่งการขั้นตอนการแบ่งย่อยส่วนประกอบภาพแบบอัตโนมัติ ออกเป็น 4 ขั้นตอน คือ ขั้นตอนหาค่าความเข้มโดยใช้วิธี K-mean clustering algorithm ในขั้นตอนที่ 2 เป็นการตัดทอนส่วนลักษณะต่างๆ ในขั้นตอนที่ 3 เป็นการวิเคราะห์ จุดภาพ ด้วย Neural Network ในขั้นตอนสุดท้ายจะใช้ Rule-based expert system

งานวิจัยของ ริคคาโด บอสโคโล; แมทธิว บราวน์; และมิเชล แม็คนิตต์ เกรย์ (Riccardo Boscolo; Matthew S. Brown; and Michael F. McNitt - Gray. 2002) ได้มีการพัฒนาวิธีการแบ่งย่อยส่วนประกอบของภาพแบบใหม่โดยรวมกับระบบฐานความรู้ ร่วมกับวิธี Active contour-based ที่ควบคุมความสัมพันธ์กับตัวแปรเช่น ศูนย์กลาง ขนาด ค่าเฉลี่ยความเข้มของภาพ

งานวิจัยของ เชีย ชือ เถิง; ลินดา แชพริโอ; และไอรา คาเลต (Chia-Chi Teng; Linda G. Shapiro; and Ira Kalet. 2006) ได้มีมุ่งเน้นที่จะพัฒนาการแบ่งย่อยส่วนออกจากภาพ CT (Computed Tomography) ด้วยวิธีอัตโนมัติ โดยประยุกต์ใช้วิธี Dynamic Threshold จากการศึกษา Set ของภาพศีรษะและคอกว่า 30 Set งานวิจัยนี้มุ่งเน้นประโยชน์เพื่อใช้ในการแพทย์

2.5.2 ประวัติและเทคโนโลยีที่เกี่ยวข้องกับการบีบอัดข้อมูล

ในปี 1948 การพัฒนาศาสตร์ทางด้านการบีบอัดข้อมูลได้ทวีความสำคัญขึ้นอย่างมาก โดยเฉพาะหลังจากที่ระบบสื่อสารมีการพัฒนาเข้าสู่การใช้เทคโนโลยีสัญญาณดิจิทัล แทนระบบอะนาล็อกแบบเดิม ยกตัวอย่างเช่น โทรศัพท์เคลื่อนที่ ระบบสื่อสารผ่านดาวเทียม การสื่อสารผ่านโครงข่ายอินเทอร์เน็ต และระบบถ่ายทอดโทรทัศน์ ดังนั้นเทคนิคทางด้านนี้จึงได้รับความสนใจอย่างมาก ซึ่งผลงานที่สำคัญนั้นคือ ทฤษฎีเชิงคณิตศาสตร์สำหรับระบบสื่อสาร (The Mathematical Theory of Communication) โดย Claude Shannon ซึ่งได้รับการตีพิมพ์

เผยแพร่ในวารสาร The Bell System Technical Journal ในทฤษฎีของแชนนอนได้รับการพัฒนาขึ้นบนพื้นฐานและหลักการของความน่าจะเป็น และการค้นพบของ Claude Shannon นับว่ามีความสำคัญและเป็นประโยชน์อย่างยิ่งต่อการพัฒนาองค์ความรู้ของระบบสื่อสาร โดยเฉพาะในส่วนของทฤษฎีความจุของช่องสัญญาณและการวัดปริมาณข่าวสาร ซึ่งจัดว่าเป็นพื้นฐานหลักของการพัฒนากรรมวิธีการเข้ารหัสเพื่อบีบอัดข้อมูลและการสื่อสาร

ในปี 1952 ได้มีการคิดค้น รหัสฮัฟฟ์แมน (Huffman Code) โดย Huffman โดยได้ตีพิมพ์ในบทความวิชาการชื่อ Minimum-redundancy coding ทฤษฎีนี้ใช้หลักการอัลกอริทึมเพื่อใช้สำหรับการเข้ารหัสข้อมูล เนื่องจากอัลกอริทึมมีประสิทธิภาพสูงมาก สามารถให้ค่ารหัสที่มีขนาดโดยเฉลี่ยต่ำสุด ถึงแม้ว่ารหัสฮัฟฟ์แมนจะมีสมรรถนะในการบีบอัดที่สูงสุดเท่าที่เป็นไปได้ แต่อยู่ใต้เงื่อนไขที่กำหนดว่าการเข้ารหัสที่กำหนดว่าการเข้ารหัสจะพิจารณาสัญลักษณ์ที่อยู่ในข้อมูลต้นฉบับทีละตัว อย่างเป็นอิสระจากกัน และสัญลักษณ์แต่ละตัวได้รับการแปลงให้เป็นค่ารหัสเฉพาะตัวไม่ซ้ำกับสัญลักษณ์อื่น ผลการเข้ารหัสที่ได้จึงประกอบด้วยค่ารหัสของสัญลักษณ์แต่ละตัวที่นำมาต่อเรียงกัน จึงมีการพัฒนาระบบอัลกอริทึมรูปแบบใหม่ที่สามารถบีบอัดข้อมูลได้ดีกว่ารหัสฮัฟฟ์แมน หากแต่กระบวนการที่ใช้ต้องมีความแตกต่างไป กล่าวคือ ไม่ใช่ใช้ข้อบังคับเงื่อนไขเดียวกับ รหัสฮัฟฟ์แมน เช่น รหัสเลขคณิต (Arithmetic Code) โดยรหัสเลขคณิตจะแปลงสัญลักษณ์ในข้อมูลต้นฉบับทั้งหมดให้ได้เป็นตัวเลขจำนวนจริงจำนวนหนึ่งตัว และจึงแปลงต่อไปเป็นค่ารหัสเพียงหนึ่งค่า ทั้งนี้พบว่ารหัสเลขคณิตมีขีดความสามารถในการบีบอัดให้ได้ผลใกล้เคียงกับค่าเอนโทรปีของแหล่งกำเนิดข้อมูลมาก

งานวิจัยของ แครรี่ และ วิทเทน (J. G. Cleary; and I. H. Witten. 1984) ได้เสนอเทคนิคการบีบอัดโดยวิธี PPM (Prediction by Partial Matching) ซึ่งหลักการทำงานของวิธีนี้อยู่บนพื้นฐานของแบบจำลองมาร์คอฟ (Markov model) ที่มีกลไกการปรับเปลี่ยนค่าความน่าจะเป็นของแบบจำลองแหล่งกำเนิดข้อมูลอย่างต่อเนื่อง เพื่อให้กระบวนการเข้ารหัสสามารถใช้ประโยชน์จากการทราบบริบทของข้อมูลได้อย่างมีประสิทธิภาพสูง แต่ก็ต้องอาศัยการคำนวณในปริมาณที่สูงมากด้วยเช่นกัน แนวคิดของวิธีการเข้ารหัส PPM คือการสร้างแบบจำลองที่สามารถทำนายอักขระตัวปัจจุบันที่กำลังจะเข้ารหัสจากข้อมูลอักขระในอดีตที่ทราบอยู่ก่อนหน้านี้ ยกตัวอย่างเช่น สมมติชุดข้อมูลอักขระที่เพิ่งได้รับการเข้ารหัสคือ 'techni' ก็สามารถทำนายได้ระดับหนึ่งว่าอักขระตัวถัดไปมีความเป็นไปได้เพียง 2 แบบเท่านั้นคือ c และ q เพราะในภาษาอังกฤษคำที่ขึ้นต้นด้วย techni มีเพียง technical , technician และ technique เท่านั้น ตัวอย่างชี้ให้เห็นว่าบริบทของข้อมูลที่อยู่ก่อนหน้าสามารถนำมาใช้ให้เกิดประโยชน์ได้

งานวิจัยของ เบอร์โรวส์ และ วิลเลอร์ (M. Burrows; and D.J. Wheeler. 1994) ได้นำเสนอแนวทางการบีบอัดข้อมูลรูปแบบใหม่ที่น่าสนใจเป็นอย่างมาก เพราะเป็นวิธีที่เรียบง่าย ไม่ซับซ้อน และสามารถให้ผลการบีบอัดที่ใกล้เคียงค่าเอนโทรปี โดยเมื่อนำวิธีใหม่นี้ไปเปรียบเทียบกับวิธีการบีบอัดโดยใช้พจนานุกรม เช่น LZ77 และ LZ78 พบว่ามีประสิทธิภาพการ

บีบอัดที่ดีกว่า หรือหากนำไปเทียบกับวิธี PPM พบว่าให้ค่าที่ใกล้เคียงกันมาก แนวคิดที่เสนอนั้นแบ่งออกเป็นสองขั้นตอน ขั้นตอนแรกให้นำข้อมูลต้นฉบับไปผ่านกรรมวิธีการแปลงข้อมูลที่เรียกว่า BWT (Burrows-Wheeler Transform) ก่อนจากนั้นในขั้นตอนที่สองให้นำข้อมูลผ่านการแปลงแล้วไปบีบอัดด้วยวิธีการเข้ารหัสเอนโทรปี จะเห็นว่ากรรมวิธีการแปลงจัดเป็นหัวใจหรือคุณลักษณะเฉพาะตัวที่สำคัญของการบีบอัดแบบนี้ หลักการทำงานของ การแปลงนี้เป็นการสลับลำดับตำแหน่งการเรียงตัวของสัญลักษณ์ใหม่ โดยไม่ก่อให้เกิดการเปลี่ยนแปลงของรูปแบบหรือจำนวนสัญลักษณ์แต่อย่างใด ฉะนั้น การแปลง BWT เพียงอย่างเดียวจึงไม่จัดว่าเป็นการบีบอัดแต่อย่างใด คุณลักษณะสำคัญหรือข้อดีของการแปลง BWT คือข้อมูลที่ผ่านการแปลงมีแนวโน้มที่จะอยู่ในรูปแบบหรือบริบทที่เหมาะสมกับการบีบอัดได้อย่างมีประสิทธิภาพมากขึ้น เมื่อเทียบกับการบีบอัดโดยตรงจากภาพต้นฉบับ



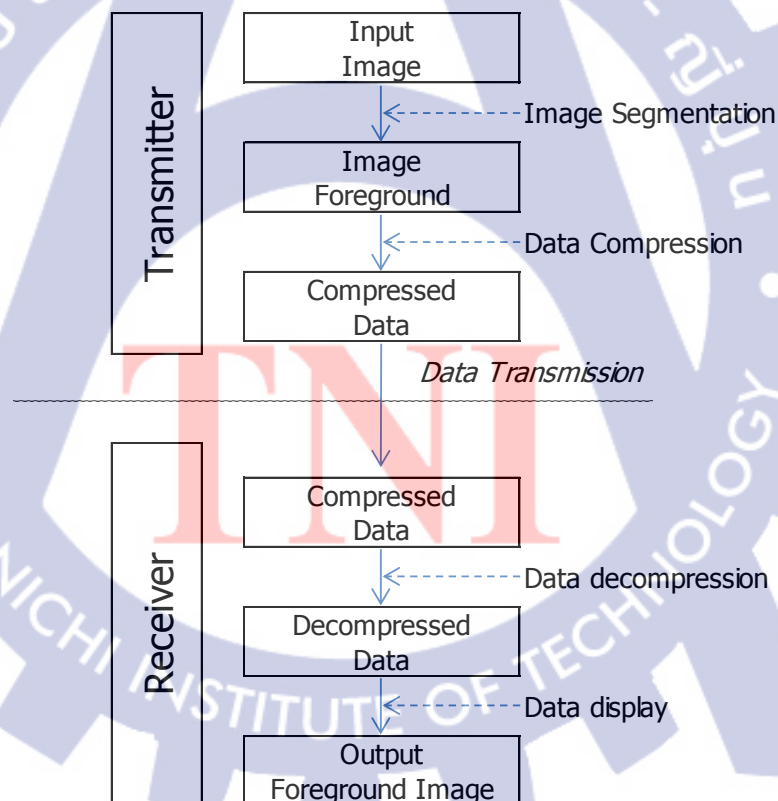
บทที่ 3

การออกแบบซอฟต์แวร์

3.1 การออกแบบซอฟต์แวร์

ซอฟต์แวร์ถูกออกแบบด้วยภาษาของ MATLAB รุ่น R2012b ภายใต้ระบบปฏิบัติการ Microsoft Window XP ผู้วิจัยใช้สมมติฐานของการตัดข้อมูลบางส่วนที่ไม่จำเป็นออกไป จะทำให้ข้อมูลที่ต้องการใช้เพื่องานใด ๆ มีขนาดลดลง อีกทั้งยังทำการจัดการข้อมูลนั้น ๆ ให้ง่ายขึ้น ซึ่งในงานออกแบบและวิจัยนี้จะออกแบบซอฟต์แวร์เพื่อแบ่งส่วนข้อมูล ที่รวมเข้ากับการบีบอัด มีจุดประสงค์เพื่อให้ได้ข้อมูลที่ต้องการ และทำให้มีขนาดเล็กลง เพื่อมุ่งเน้นให้การส่งข้อมูลผ่านทางอินเทอร์เน็ตทำได้เร็ว และการแสดงผลมีความต่อเนื่องมากขึ้น

ขั้นตอนของการเขียนโปรแกรมการลดขนาดข้อมูลภาพโดยใช้เทคนิคของการแบ่งส่วนภาพร่วมกับการบีบอัดข้อมูลภาพ แสดงดังรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนของโปรแกรมการลดขนาดข้อมูลภาพ

ขั้นตอนการออกแบบโปรแกรม ถูกแบ่งการทำงานเป็นสองส่วนได้แก่ ภาคส่ง (Transmitter) และภาครับ (Receiver) โดยในส่วนของภาคส่งจะทำการเลือกค่าตำแหน่งจุดภาพ และความเข้มของแต่ละชั้นสี (RGB) ก่อนนำมาบีบอัดและส่งเป็นข้อมูลไปยังภาครับ ซึ่งภาครับก็จะทำการถอดรหัสข้อมูลเพื่อจะแสดงผลข้อมูลภาพออกมาเป็นภาพสีในลักษณะเดียวกันกับภาพต้นแบบ แต่จะแสดงเฉพาะส่วนที่ต้องการ

3.1.1 การแบ่งส่วนภาพ (Image Segmentation)

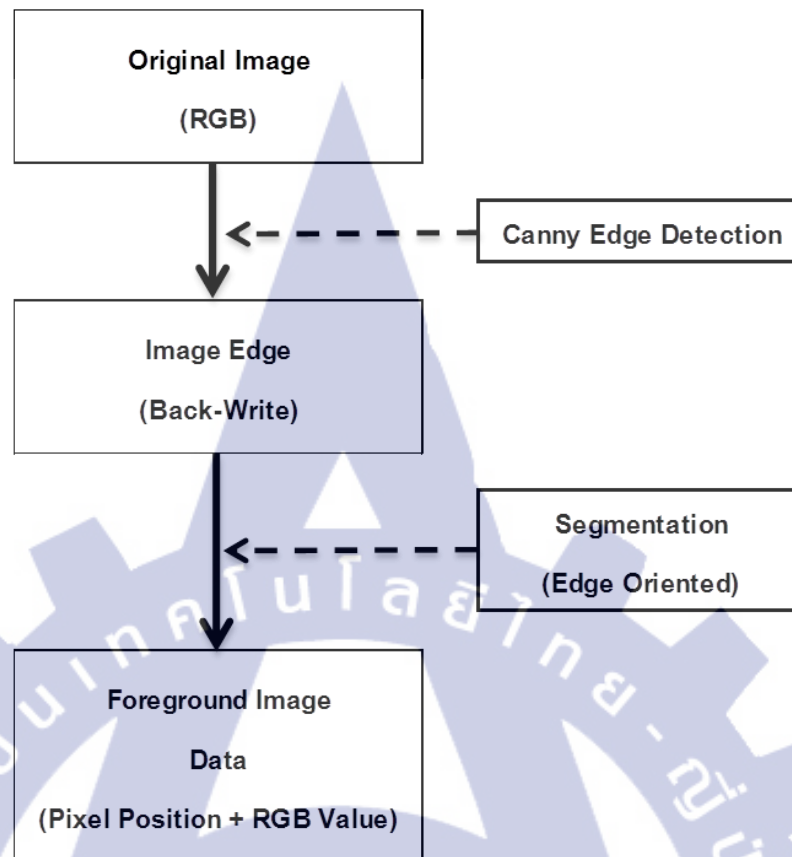
การออกแบบเริ่มจากการประมวลผลภาพต้นแบบ (Input Image) ที่มาจากเฟรมภาพของเว็บแคม และทำการแยกส่วนภาพคนและภาพพื้นหลังออกจากกัน จากนั้นเขียนโปรแกรมเลือกเฉพาะตำแหน่งและค่าของจุดภาพที่ต้องการ เพื่อลดจำนวนข้อมูลในรูปภาพส่วนที่ไม่ต้องการลง

การออกแบบโปรแกรมเริ่มต้นจากการรับภาพสี (RGB) มาจากกล้องเว็บแคมด้วยขนาด $120 \times 160 \times 3$ จุดภาพ จากนั้นหาขอบภาพโดยใช้วิธี Canny เพื่อที่จะสามารถหาตำแหน่งและเก็บข้อมูลตำแหน่งจุดภาพและค่าข้อมูลสี RGB ของแต่ละตำแหน่งของจุดภาพของภาพส่วนที่ต้องการได้



รูปที่ 3.2 การหาขอบภาพด้วยวิธี Canny

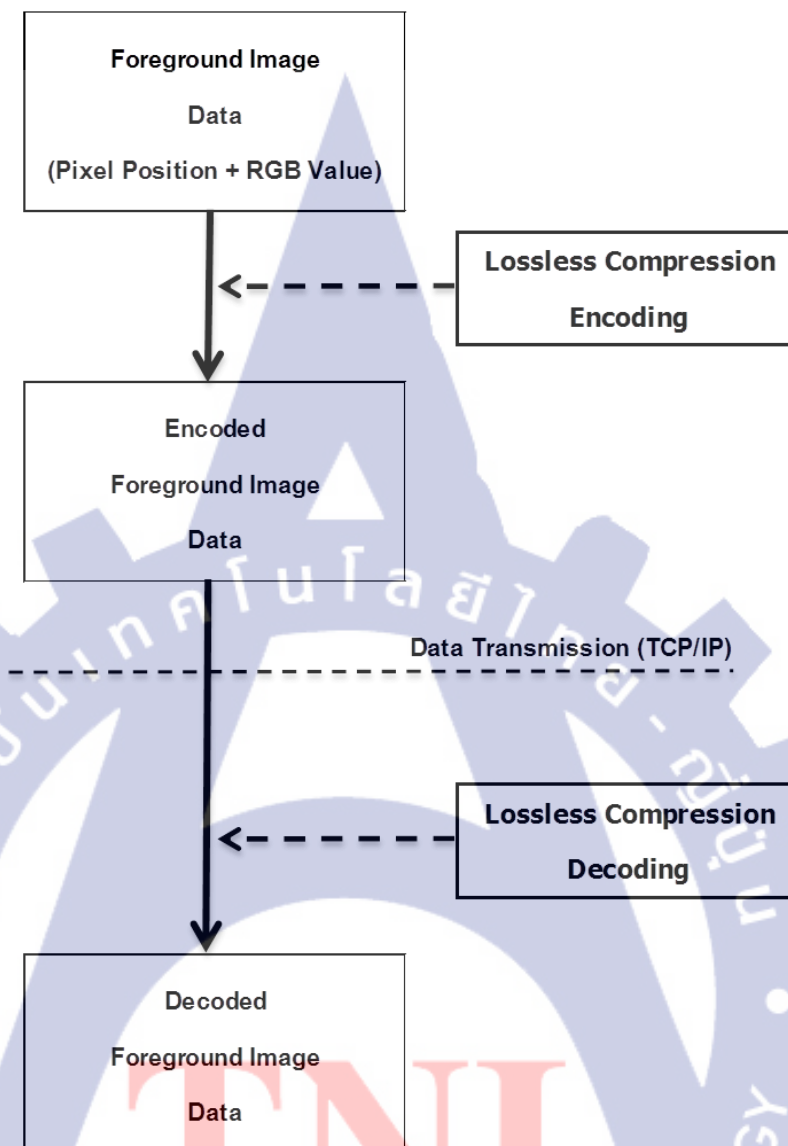
เนื่องจากการแบ่งย่อยส่วนประกอบของภาพเป็นการแยกแยะระหว่างส่วนที่เราสนใจ (Foreground) ซึ่งในกรณีนี้จะเป็นภาพของคน กับส่วนที่ไม่ต้องการ (Background) ซึ่งในกรณีนี้จะเป็นภาพของฉากหลัง ซึ่งการออกแบบการแบ่งส่วนภาพจะใช้ขั้นตอนวิธีดังรูปที่ 3.3



รูปที่ 3.3 การออกแบบการแบ่งส่วนภาพ

3.1.2 การบีบอัดข้อมูลภาพ (Data Compression)

ข้อมูลรูปภาพส่วนที่ต้องการ (Foreground) จะถูกนำมาทำการบีบอัดข้อมูลแบบไม่มีการสูญเสียรายละเอียดข้อมูล (Lossless Compression) เพื่อทำการลดจำนวนของข้อมูลเฉพาะส่วนที่เราสนใจ โดยที่ จะไม่ส่งผลต่อคุณภาพของข้อมูล ซึ่งการออกแบบการบีบอัดข้อมูลภาพจะเป็นไปตามขั้นตอนดังรูปที่ 3.4 ซึ่งในส่วนของภาครับ หลังจากการส่งผ่านข้อมูลทางอินเทอร์เน็ตแล้ว ก็จะต้องเขียนการถอดรหัสข้อมูลและการจัดลำดับข้อมูลที่ได้รับเพื่อเตรียมพร้อมที่จะแสดงผลข้อมูลภาพต่อไป



รูปที่ 3.4 การออกแบบการบีบอัดข้อมูลภาพ

3.2 การออกแบบระบบทดสอบและการแสดงผล

เพื่อให้ผู้ใช้ ใช้งานโปรแกรมได้สะดวก และเพื่อการแสดงผลทดสอบงานวิจัยที่ชัดเจน ผู้วิจัยทำการออกแบบ GUI (Graphic User Interface) หรือส่วนต่อประสานกราฟิกกับผู้ใช้ การออกแบบโปรแกรมในส่วนนี้ แบ่งการทำงานเป็นสองส่วนได้แก่ ภาคส่ง (Transmitter) และ ภาครับ (Receiver) โดยในส่วนของภาคส่งจะทำการเลือกค่าตำแหน่งจุดภาพและค่าความเข้มของแต่ละชั้นสี (RGB) ก่อนนำมาบีบอัดและส่งเป็นข้อมูลไปยังภาครับ ซึ่งภาครับก็จะทำการ

ถอดรหัสข้อมูลเพื่อการแสดงผลข้อมูลภาพเฉพาะส่วนภาพที่เราต้องการออกมาเป็นภาพสีในลักษณะเดียวกันกับภาพต้นแบบแต่จะแสดงเฉพาะส่วนที่ต้องการ

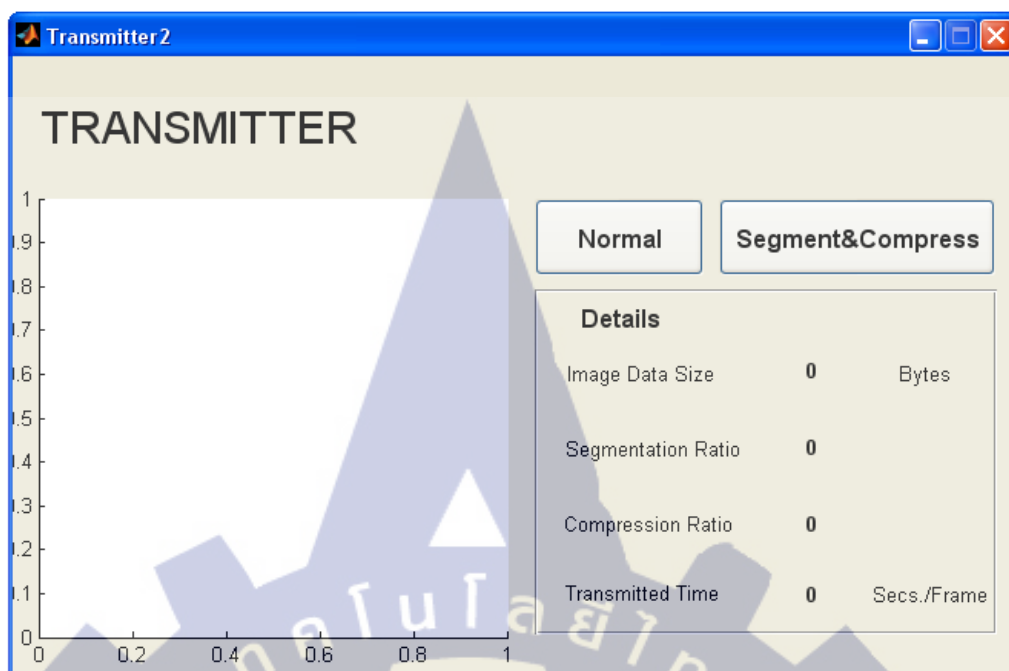
ข้อมูลที่จะถูกส่งและรับ รวมถึงการแสดงผล ถูกกำหนดเป็น 2 กรณีคือ

- (1) ข้อมูลทั้งที่ได้ถูกแบ่งส่วนและบีบอัดแล้ว
- (2) ข้อมูลที่ยังไม่ได้ถูกแบ่งส่วนและบีบอัด (ข้อมูลตั้งต้น)

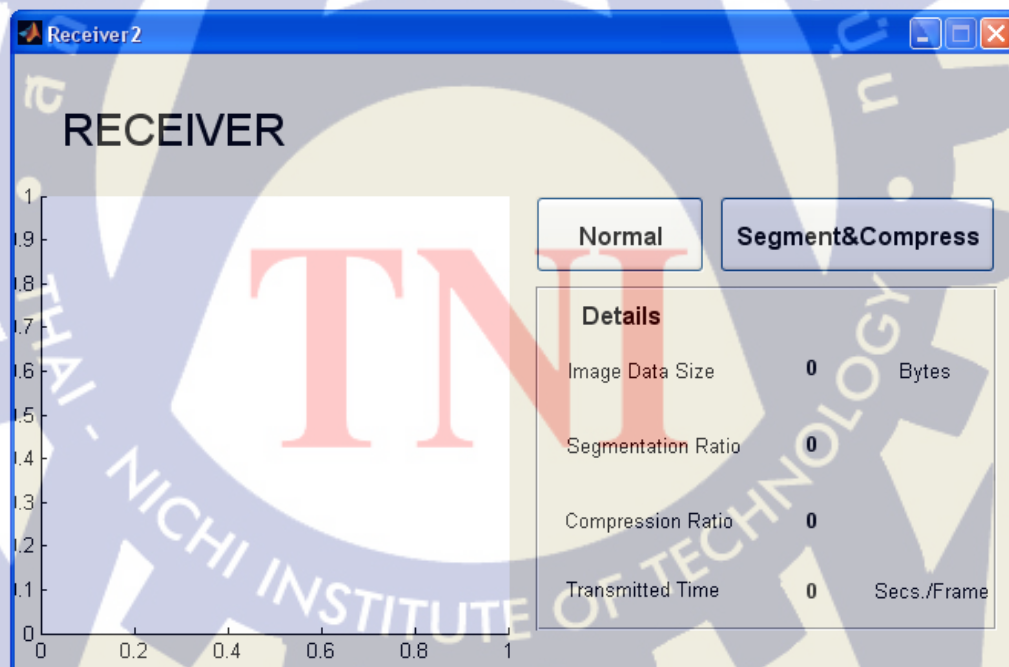
การออกแบบให้การส่งและรับข้อมูลเป็น 2 กรณีเพื่อให้ ผู้ใช้งานโปรแกรมสามารถเลือกการแสดงผลข้อมูลภาพได้ อีกทั้งยังสามารถเปรียบเทียบผลลัพธ์ที่ได้จากการส่งและรับข้อมูลของทั้ง 2 กรณีได้

การออกแบบโปรแกรมในส่วนนี้ จะออกแบบให้แสดงผลในลักษณะเดียวกับเว็บแคม และจะทำการทดสอบโดยการส่งข้อมูลผ่านระบบอินเทอร์เน็ต และวัดผลการประมวลผลของโปรแกรมที่ออกแบบ โดยจะออกแบบให้หน้าต่างแสดงขนาดของข้อมูล (Image Data Size), อัตราการแบ่งส่วนขนาดข้อมูลภาพ (Segmentation Ratio), อัตราการบีบอัดข้อมูลภาพ (Compression Ratio), และเวลาที่ใช้ประมวลผลและส่งข้อมูลภาพ (Transmitted Time) โดยหน้าต่างของการแสดงผลจะมีลักษณะดังรูปที่ 3.5

การทำงานของโปรแกรม เริ่มจากภาคส่ง (Transmitter) รับข้อมูลภาพจากกล้องเว็บแคม และประมวลผลภาพโดยใช้ทฤษฎีการหาขอบภาพของ Canny จากนั้นทำการกำหนดเลือกค่าตำแหน่งจุดภาพ และค่าความเข้มของแต่ละชั้นสี แดง เขียว น้ำเงิน (RGB) ของตำแหน่งจุดภาพที่ขอบ และส่วนภายในขอบซึ่งเป็นภาพคนที่เราสนใจ ก่อนนำข้อมูลภาพนั้นมาบีบอัดและส่งข้อมูลไปยังภาครับ ซึ่งภาครับก็จะทำหน้าที่รับข้อมูลที่ถูกส่ง จากนั้นทำการถอดรหัสข้อมูล ขั้นตอนต่อมาคือการจัดเรียงข้อมูลเพื่อแสดงผลข้อมูลภาพเฉพาะส่วนภาพที่เราต้องการในตำแหน่งของจุดภาพนั้น ๆ เป็นภาพสีในลักษณะเดียวกันกับภาพต้นแบบ ซึ่งภาพที่ออกมาจะแสดงในลักษณะของการตัดภาพพื้นหลังออก ดังรูปที่ 3.6

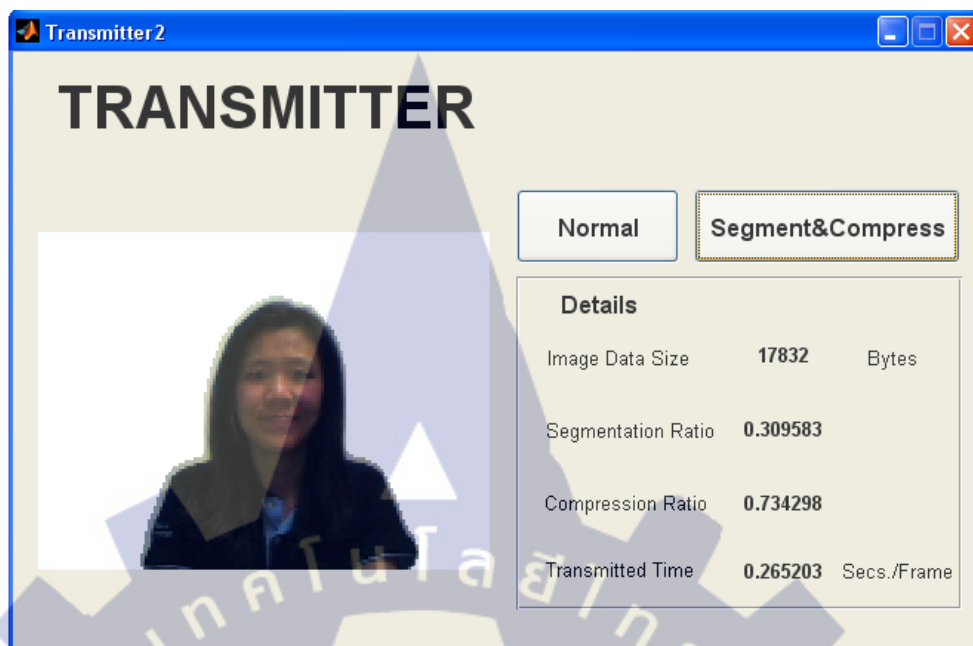


ก) หน้าต่างภาคส่ง (Transmitter)

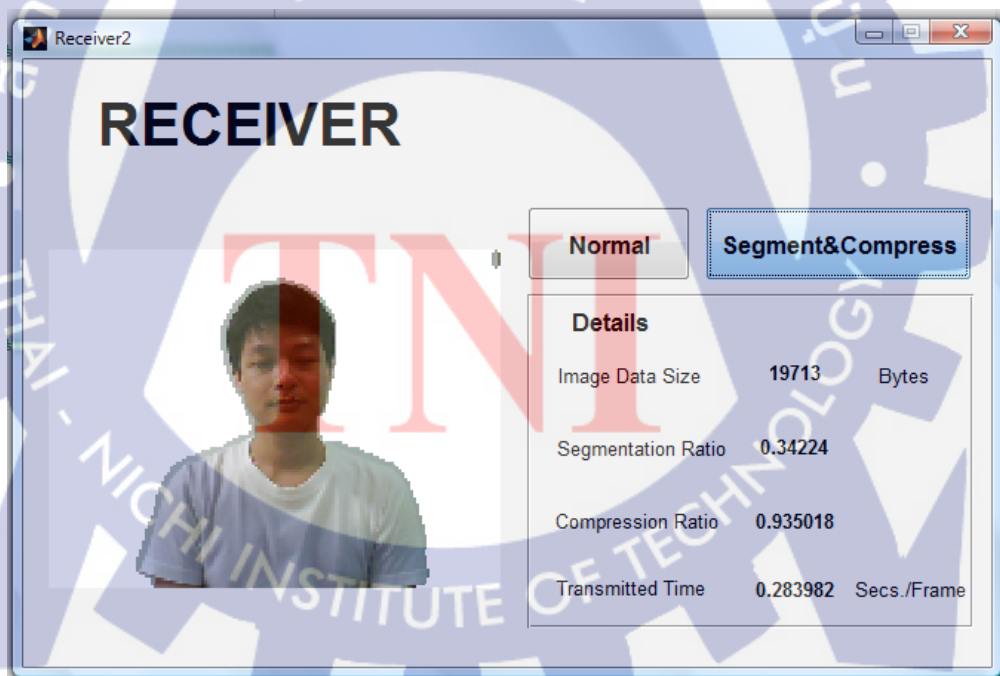


ข) หน้าต่างภาครับ (Receiver)

รูปที่ 3.5 หน้าต่างที่ออกแบบเพื่อการแสดงผล



ก) หน้าต่างภาคส่ง (Transmitter)



ข) หน้าต่างภาครับ (Receiver)

รูปที่ 3.6 การแสดงผลของการทดสอบ

3.3 วิธีการทดสอบประสิทธิภาพของโปรแกรม

วิธีการทดสอบประสิทธิภาพของโปรแกรมจะทำการทดสอบการทำงานจริงผ่านระบบอินเทอร์เน็ต Wi-Fi ของผู้ให้บริการเครือข่ายอินเทอร์เน็ต โดยการติดตั้งโปรแกรมบนเครื่องคอมพิวเตอร์สองเครื่องและทดสอบการใช้งานของโปรแกรมจำนวน 40 ครั้ง จดบันทึกค่าที่ได้ และนำมาวิเคราะห์ผลการทดสอบ

3.4 ผลการทดสอบที่คาดหวัง

ผลการทดสอบคาดหวังคือโปรแกรมที่ออกแบบจะสามารถทำการแยกส่วนภาพคนออกจากภาพพื้นหลังได้ และทำให้ข้อมูลภาพมีขนาดลดลง

ประสิทธิภาพในการส่งข้อมูลภาพผ่านอินเทอร์เน็ตเมื่อพิจารณาด้านความเร็วในการส่งต่อเฟรมภาพ จะมีความเร็วเพิ่มขึ้น เมื่อหักล้างเวลาการประมวลผลที่ลดลงเนื่องจากการลดลงของข้อมูลภาพ กับเวลาที่ใช้ในการประมวลผลของโปรแกรมที่ออกแบบ



บทที่ 4

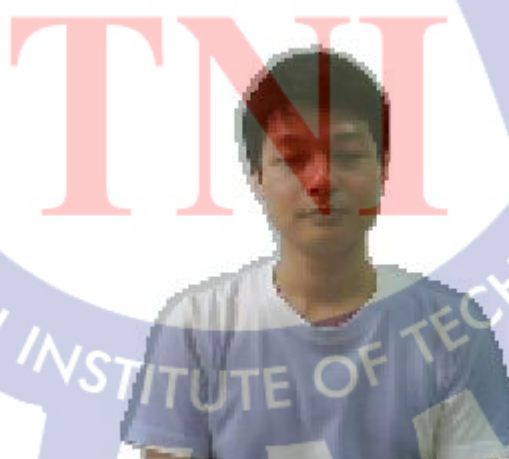
ผลการทดสอบและการวิเคราะห์ผล

4.1 ผลการทดสอบ

จากการออกแบบซอฟต์แวร์และทดสอบวัดค่าของผลที่ได้ พบว่าโปรแกรมที่ออกแบบสามารถแบ่งส่วนที่สนใจออกจากภาพพื้นหลังได้ ซึ่งในกรณีนี้เป็นภาพคนที่ถูกแสดงโดยการตัดภาพพื้นหลังออก ดังแสดงในรูปที่ 4.1



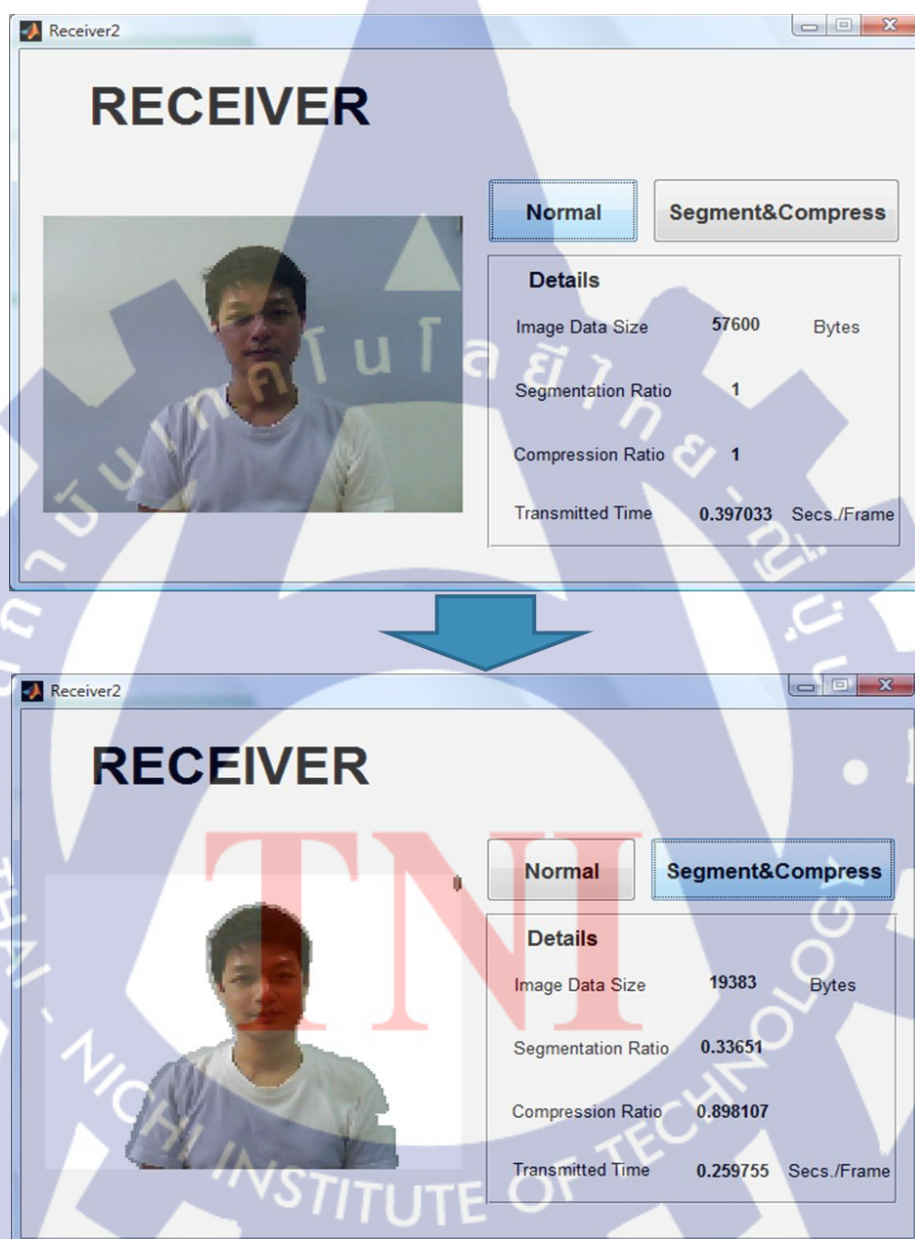
ก) ตัวอย่างภาพตั้งต้นที่ได้จากกล้องเว็บแคม



ข) ตัวอย่างภาพที่ได้จากการแบ่งส่วน

รูปที่ 4.1 ผลที่ได้จากการแบ่งส่วนภาพ

จากการออกแบบซอฟต์แวร์เพื่อการแสดงผลโดยหน้าต่าง GUI (Graphic User Interface) ทำให้ได้ผลการแสดงผล ขนาดข้อมูลภาพ (Image Data Size), อัตราการแบ่งส่วน ขนาดข้อมูลภาพ (Segmentation Ratio), อัตราการบีบอัดข้อมูลภาพ (Compression Ratio), และเวลาที่ใช้ประมวลผลและส่งข้อมูลภาพ (Transmitted Time) ดังแสดงในรูปที่ 4.2 และ 4.3



รูปที่ 4.2 ตัวอย่างภาพผลที่ได้จากการทดสอบกับพื้นหลังสีอ่อน



รูปที่ 4.3 ตัวอย่างภาพผลที่ได้จากการทดสอบกับพื้นหลังสีเข้ม

จากการแสดงผลของโปรแกรม จะเห็นได้ว่าภาพที่ถูกแสดงเป็นภาพคนที่ถูกตัดภาพพื้นหลังออกไป และขนาดของข้อมูลภาพจากภาพตั้งต้นที่มีขนาด 57,600 ไบต์ มีขนาดลดลงได้อีกทั้งเวลาที่ใช้ในส่วนของการประมวลผลภาพการส่งผ่านข้อมูลทางอินเทอร์เน็ตมีค่าตัวเลขที่ลดลงกว่าภาพตั้งต้นอีกด้วย ซึ่งในส่วนของการรายละเอียดจะถูกกล่าวในลำดับต่อไป

จากการทดสอบการลดขนาดของข้อมูลภาพจากการแบ่งส่วนภาพและบีบอัดข้อมูลจำนวน 40 ครั้ง พบว่าผลที่ได้จากการแบ่งส่วนทำให้ข้อมูลภาพมีขนาดลดลงเฉลี่ยร้อยละ 68 และผลที่ได้จากการบีบอัดทำให้ข้อมูลภาพมีขนาดลดลงเฉลี่ยอีกร้อยละ 15 ทำให้ผลการทดสอบโดยรวมทำให้ภาพที่มีขนาดข้อมูล 57,600 ไบต์ (Bytes) ลดลงเหลือขนาดเฉลี่ย 15,601 ไบต์ กล่าวคือข้อมูลภาพมีขนาดลดลงเฉลี่ย 41,999 ไบต์ หรือร้อยละ 72.91 ดังแสดงในตารางที่ 4.1

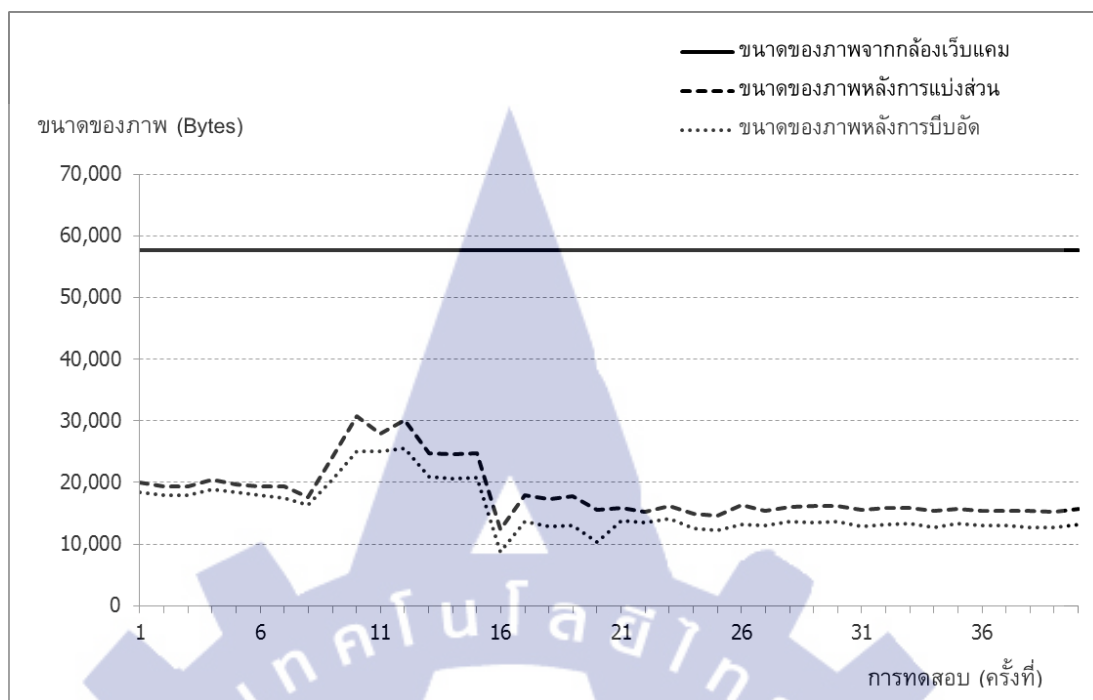
ตารางที่ 4.1 ขนาดของข้อมูลที่ได้จากการทดสอบ

การทดสอบ ครั้งที่	① ขนาดของภาพ จากกล้องเว็บแคม (ไบต์)	② สัดส่วน การแบ่งส่วน	③=①×② ขนาดของภาพ หลังการแบ่งส่วน (ไบต์)	④ สัดส่วน การบีบอัด	⑤=③×④ ขนาดของภาพ หลังการบีบอัด (ไบต์)	⑥=⑤-① ขนาดของภาพ ที่ลดลง (ไบต์)
1	57,600	0.3458	19,920	0.9253	18,432	-39,168
2	57,600	0.3366	19,389	0.9242	17,920	-39,680
3	57,600	0.3356	19,332	0.9270	17,920	-39,680
4	57,600	0.3544	20,412	0.9281	18,944	-38,656
5	57,600	0.3422	19,713	0.9350	18,432	-39,168
6	57,600	0.3365	19,383	0.9245	17,920	-39,680
7	57,600	0.3365	19,383	0.8981	17,408	-40,192
8	57,600	0.3054	17,589	0.9315	16,384	-41,216
9	57,600	0.4190	24,135	0.8486	20,480	-37,120
10	57,600	0.5343	30,777	0.8152	25,088	-32,512
11	57,600	0.4855	27,963	0.8972	25,088	-32,512
12	57,600	0.5231	30,129	0.8497	25,600	-32,000
13	57,600	0.4305	24,798	0.8447	20,948	-36,652
14	57,600	0.4254	24,501	0.8440	20,680	-36,920
15	57,600	0.4291	24,717	0.8383	20,720	-36,880
16	57,600	0.2149	12,381	0.7117	8,812	-48,788
17	57,600	0.3122	17,985	0.7614	13,693	-43,907
18	57,600	0.3001	17,283	0.7476	12,920	-44,680
19	57,600	0.3096	17,832	0.7343	13,094	-44,506
20	57,600	0.2703	15,570	0.6598	10,273	-47,327

ตารางที่ 4.1 ขนาดของข้อมูลที่ได้จากการทดสอบ (ต่อ)

การทดสอบ ครั้งที่	① ขนาดของภาพ จากกล้องเว็บแคม (ไบต์)	② สัดส่วน การแบ่งส่วน	③=①×② ขนาดของภาพ หลังการแบ่งส่วน (ไบต์)	④ สัดส่วน การบีบอัด	⑤=③×④ ขนาดของภาพ หลังการบีบอัด (ไบต์)	⑥=⑤-① ขนาดของภาพ ที่ลดลง (ไบต์)
21	57,600	0.2752	15,852	0.8673	13,749	-43,851
22	57,600	0.2642	15,219	0.8882	13,517	-44,083
23	57,600	0.2799	16,125	0.8791	14,176	-43,424
24	57,600	0.2604	14,997	0.8354	12,528	-45,072
25	57,600	0.2529	14,568	0.8363	12,183	-45,417
26	57,600	0.2840	16,356	0.8070	13,200	-44,400
27	57,600	0.2660	15,324	0.8497	13,021	-44,579
28	57,600	0.2792	16,080	0.8506	13,678	-43,922
29	57,600	0.2801	16,134	0.8401	13,554	-44,046
30	57,600	0.2805	16,155	0.8433	13,623	-43,977
31	57,600	0.2692	15,504	0.8329	12,914	-44,686
32	57,600	0.2756	15,873	0.8344	13,245	-44,355
33	57,600	0.2759	15,894	0.8400	13,351	-44,249
34	57,600	0.2668	15,366	0.8281	12,724	-44,876
35	57,600	0.2723	15,687	0.8488	13,315	-44,285
36	57,600	0.2669	15,375	0.8440	12,976	-44,624
37	57,600	0.2675	15,408	0.8400	12,943	-44,657
38	57,600	0.2667	15,360	0.8313	12,769	-44,831
39	57,600	0.2655	15,291	0.8316	12,716	-44,884
40	57,600	0.2722	15,681	0.8362	13,113	-44,487
ค่าเฉลี่ย	57,600	0.3192	18,386	0.8453	15,601	-41,999

จากข้อมูลที่ได้จากการทดสอบ จะเห็นได้ว่าขนาดข้อมูลภาพที่ได้จากกล้องเว็บแคมมีค่าคงที่ที่ 57,600 ไบต์ และค่าข้อมูลภาพที่ได้จากการแบ่งส่วนจะมีค่าสูงสุดที่ 30,777 ไบต์ ในการทดสอบครั้งที่ 10 และค่าต่ำสุดอยู่ที่ 12,381 ไบต์ ในการทดสอบครั้งที่ 16 โดยมีค่าเฉลี่ยของอัตราการแบ่งส่วนข้อมูลภาพจากการทดสอบทั้งหมด 40 ครั้งอยู่ที่ 0.3192 หรือสามารถลดขนาดข้อมูลภาพลดลงเหลือร้อยละ 31.92



รูปที่ 4.4 ผลการทดสอบขนาดข้อมูลที่ถูกแบ่งส่วนและบีบอัด

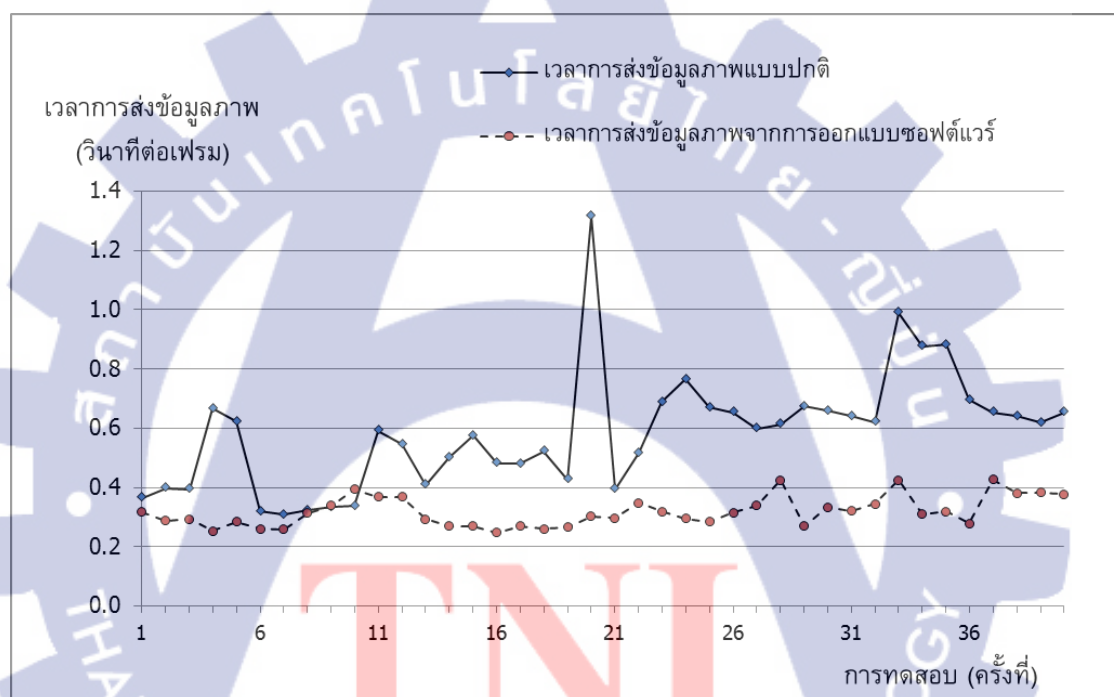
การบีบอัดข้อมูลภาพหลังจากการแบ่งส่วนภาพ ทำให้ข้อมูลภาพมีค่าสูงสุดที่ 25,600 ไบต์ ในการทดสอบครั้งที่ 12 และค่าต่ำสุดอยู่ที่ 8,812 ไบต์ ในการทดสอบครั้งที่ 16 โดยมีค่าเฉลี่ยของอัตราการแบ่งส่วนและบีบอัดข้อมูลภาพจากการทดสอบทั้งหมด 40 ครั้งอยู่ที่ 0.2709 หรือสามารถลดขนาดข้อมูลภาพลดลงเหลือร้อยละ 27.09 เมื่อเทียบกับข้อมูลของภาพตั้งต้น

ตารางที่ 4.2 เวลาในการส่งข้อมูลที่ได้จากการทดสอบ

การทดสอบ ครั้งที่	เวลาการส่งข้อมูลภาพ แบบปกติ (วินาทีต่อเฟรม)	เวลาการส่งข้อมูลภาพ จากการออกแบบซอฟต์แวร์ (วินาทีต่อเฟรม)	อัตราส่วนเวลา ที่ลดลง (ร้อยละ)
1	0.3649	0.3180	12.85
2	0.3970	0.2880	27.47
3	0.3934	0.2935	25.40
4	0.6654	0.2534	61.92
5	0.6217	0.2840	54.32
6	0.3198	0.2597	18.79
7	0.3084	0.2596	15.83
8	0.3231	0.3127	3.23
9	0.3333	0.3396	-1.90
10	0.3378	0.3932	-16.40
11	0.5919	0.3691	37.65
12	0.5447	0.3677	32.50
13	0.4088	0.2942	28.05
14	0.5023	0.2701	46.23
15	0.5750	0.2695	53.12
16	0.4808	0.2480	48.42
17	0.4800	0.2690	43.96
18	0.5214	0.2602	50.10
19	0.4266	0.2652	37.83
20	1.3145	0.3019	77.04
21	0.3948	0.2967	24.86
22	0.5159	0.3480	32.55
23	0.6879	0.3199	53.50
24	0.7653	0.2948	61.48
25	0.6698	0.2846	57.51
26	0.6529	0.3159	51.62
27	0.5989	0.3399	43.25
28	0.6129	0.4252	30.62
29	0.6744	0.2694	60.06
30	0.6591	0.3329	49.49
31	0.6399	0.3204	49.92
32	0.6212	0.3448	44.49
33	0.9899	0.4246	57.11
34	0.8768	0.3090	64.76
35	0.8812	0.3199	63.70
36	0.6954	0.2782	59.99
37	0.6533	0.4266	34.70
38	0.6399	0.3805	40.53
39	0.6178	0.3827	38.05
40	0.6533	0.3769	42.30
ค่าเฉลี่ย	0.5853	0.3177	45.72

ผลการทดสอบจากการจับเวลาในการส่งข้อมูลภาพผ่านระบบอินเทอร์เน็ต พบว่าเวลาที่ใช้ส่งข้อมูลภาพแบบปกติมีค่าสูงสุดอยู่ที่ 1.3145 วินาทีต่อเฟรม ในการทดสอบครั้งที่ 20 เวลาที่ใช้ส่งข้อมูลภาพแบบปกติมีค่าต่ำสุดอยู่ที่ 0.3084 วินาทีต่อเฟรม ในการทดสอบครั้งที่ 7 โดยมีเวลาเฉลี่ยเป็น 0.5853 วินาทีต่อเฟรม

ผลการทดสอบจากการจับเวลาในการส่งข้อมูลภาพผ่านระบบอินเทอร์เน็ต ของภาพที่ได้จากการประมวลผลของโปรแกรมซึ่งผ่านการแบ่งส่วนและบีบอัดแล้ว พบว่าเวลาที่ใช้ส่งข้อมูลภาพมีค่าสูงสุดอยู่ที่ 0.4266 วินาทีต่อเฟรม ในการทดสอบครั้งที่ 37 เวลาที่ใช้ส่งข้อมูลภาพมีค่าต่ำสุดอยู่ที่ 0.2480 วินาทีต่อเฟรม ในการทดสอบครั้งที่ 16 โดยมีเวลาเฉลี่ยเป็น 0.3177 วินาทีต่อเฟรม



รูปที่ 4.5 ผลการทดสอบเวลาในการส่งข้อมูล

จากรูปที่ 4.5 ผลของเวลาที่ได้จากการทดลองส่งข้อมูลภาพปกติ และส่งด้วยภาพที่ได้จากการแบ่งส่วนข้อมูลและบีบอัด จะมาจากปัจจัยที่ส่งผลหลัก ๆ คือขนาดของข้อมูลภาพและอัตราความเร็วของระบบเครือข่ายอินเทอร์เน็ต พบว่าในบางครั้งที่ทำทดสอบอาจมีค่าเวลาที่ไม่คงที่ดังจะเห็นได้ในการทดสอบครั้งที่ 20 ซึ่งอาจส่งผลมาจากตัวแปรนอกเหนือที่เราพิจารณา เช่น อัตราความเร็วของระบบเครือข่ายอินเทอร์เน็ต ก็เป็นไปได้เช่นกัน แต่ในการวิเคราะห์โดยค่าเฉลี่ยที่ได้นั้น พบว่าการส่งข้อมูลภาพแบบปกติที่ใช้เวลาเฉลี่ย 0.5853 วินาทีต่อเฟรม ส่วน

การทดลองส่งด้วยภาพที่ได้จากการแบ่งส่วนข้อมูลและบีบอัด ใช้เวลาเฉลี่ยลดลงเหลือ 0.3177 วินาทีต่อเฟรม มีอัตราการใช้เวลาในการส่งข้อมูลภาพลดลงเฉลี่ยร้อยละ 45.72

4.2 การวิเคราะห์ผล

จากการทดลองการออกแบบโปรแกรมการแบ่งส่วนภาพโดยอาศัยทฤษฎีการหาขอบภาพด้วยวิธีของ Canny และเก็บค่าตำแหน่งพร้อมทั้งความเข้มสีของจุดภาพ เพื่อส่งข้อมูลและแสดงผล ทำให้ได้การแบ่งส่วนภาพคนออกมาได้ค่อนข้างชัดเจน แต่อย่างไรก็ดีในกรณีที่ภาพส่วนที่เราสนใจมีลักษณะค่าสีใกล้เคียงกับพื้นหลังมาก ๆ ก็จะทำให้ภาพบางส่วนที่เราสนใจถูกจัดแบ่ง เป็นส่วนพื้นหลังแล้วถูกตัดออกไปได้

อัตราข้อมูลที่ลดลงของการแบ่งส่วนภาพมีผลขึ้นกับสัดส่วนภาพที่สนใจต่อภาพตั้งต้น กล่าวคือ ถ้าภาพส่วนที่สนใจมีขนาดเล็ก ก็จะทำให้มีอัตราการแบ่งส่วนมาก ซึ่งเป็นผลทำให้การส่งข้อมูลภาพทางอินเทอร์เน็ตเร็วขึ้นได้มาก แต่ถ้าภาพส่วนที่สนใจมีขนาดใหญ่ก็ทำให้มีอัตราการแบ่งส่วนได้น้อย ซึ่งอาจทำให้ส่งข้อมูลภาพได้ไม่เร็วขึ้นมากนัก ในกรณีนี้อาจเป็นไปได้ที่ทำให้ใช้เวลามากขึ้นเนื่องจากต้องใช้เวลาในส่วนของการประมวลผลข้อมูลภาพ ซึ่งถ้าเปรียบเทียบกับ การส่งแบบปกติแล้ว อาจต้องใช้เวลามากกว่าก็เป็นไปได้เช่นกัน



บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้ได้พัฒนาซอฟต์แวร์เพื่อเพิ่มประสิทธิภาพในการส่งผ่านข้อมูลผ่านอินเทอร์เน็ต โดยอาศัยการแบ่งส่วนภาพ ด้วยวิธีการหาขอบภาพของ Canny เพื่อเลือกส่วนข้อมูลที่น่าสนใจและบีบอัดข้อมูลให้มีขนาดเล็กลง ผลที่ได้คือสามารถส่งข้อมูลผ่านอินเทอร์เน็ตได้เร็วและมีความต่อเนื่องมากขึ้น หรือใช้เวลาในการส่งข้อมูลภาพลดน้อยลง

ผลการทดสอบกับภาพจากเว็บแคม โดยใช้เทคนิคการแบ่งส่วนภาพในการกำจัดภาพพื้นหลังออก รวมกับการบีบอัดข้อมูลภาพเป็นผลทำให้การส่งข้อมูลผ่านอินเทอร์เน็ตมีประสิทธิภาพมากขึ้น จากเวลาที่ใช้ในการส่งภาพแบบปกติที่ใช้เวลาเฉลี่ย 0.5853 วินาทีต่อเฟรม ลดลงเหลือ 0.3177 วินาทีต่อเฟรม หรือเทียบได้กับอัตราการส่งข้อมูลภาพที่เพิ่มขึ้นจาก 1.7085 เฟรมต่อวินาที เป็น 3.147 เฟรมต่อวินาที ทั้งนี้อัตราความเร็วก็ยังขึ้นกับอัตราความเร็วของระบบเครือข่ายอินเทอร์เน็ตด้วยเช่นกัน

ซอฟต์แวร์ที่ได้พัฒนาขึ้นนี้ สามารถนำไปประยุกต์ใช้กับการสื่อสารที่มีการรับส่งข้อมูลภาพที่ต้องการลดภาระของการเก็บและประมวลผลข้อมูลภาพที่มีขนาดใหญ่ได้เช่น การส่งข้อมูลภาพของคนไข้เพื่อการรักษาที่เร็วขึ้น การส่งภาพวิดีโอสาธิตการทำงาน อีกทั้งยังใช้ได้กับการสื่อสารในลักษณะของเว็บแคมได้อีกด้วย

5.2 ข้อเสนอแนะและแนวทางการปรับปรุงงานวิจัย

จากการทดสอบการทำงานของโปรแกรมที่พัฒนาขึ้น ทำให้ทราบว่าโปรแกรมยังคงมีข้อจำกัดอยู่บางประการเช่น การแสดงภาพที่มีบางส่วนในบริเวณที่สนใจขาดหายไป เนื่องจากความใกล้เคียงกันระหว่างบริเวณที่สนใจกับภาพพื้นหลัง สิ่งเหล่านี้อาจส่งผลกับประสิทธิภาพการแบ่งส่วน และความสมบูรณ์ของการแสดงผลภาพ

แนวทางการปรับปรุงงานวิจัยนี้อาจทำได้โดยการปรับใช้ทฤษฎีการหาขอบภาพอื่นที่มีการแบ่งส่วนที่มีประสิทธิภาพมากขึ้น หรือนำเอาทฤษฎีอื่น ๆ มาปรับใช้ร่วมกันในการแบ่งส่วนของข้อมูลภาพ ซึ่งอาจจะทำให้ภาพที่ได้มีความสมบูรณ์มากขึ้น

ในส่วนของการบีบอัดข้อมูลก็เช่นกัน งานวิจัยนี้ได้เสนอการบีบอัดข้อมูลแบบไม่สูญเสีย ซึ่งอาจทำให้การบีบอัดทำได้ไม่มากนัก แนวทางการปรับปรุงอาจทำได้โดยการประยุกต์ใช้กับวิธีการอื่นที่อาจทำให้มีอัตราการบีบอัดดีขึ้น แต่อย่างไรก็ตามในโปรแกรมทางด้านการแสดงผลภาพนั้น ยังคงต้องให้ความสำคัญกับภาพที่จะถูกแสดงเป็นหลักด้วยเช่นกัน



บรรณาธิการ

บรรณานุกรม

- กระทรวงวิทยาศาสตร์และเทคโนโลยี. (2553). ระบบวิเคราะห์ฝ่าเท้าเพื่อการวินิจฉัยโรค **Foot Analysis System for Diagnosis**. (โปรแกรมเพื่องานการพัฒนาด้าน วิทยาศาสตร์และเทคโนโลยี) กรุงเทพฯ : สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ กระทรวงฯ.
- พีรพงษ์ วงษ์กรรวเรช. (2549). การแบ่งย่อยส่วนประกอบของภาพจากระบบสร้าง ภาพตัดขวางจากรังสีเอ็กซด้วยเครื่องคอมพิวเตอร์เพื่อสร้างแบบจำลองสาม มิติของกะโหลกศีรษะมนุษย์. วิทยานิพนธ์ วศ.ม. (วิทยาการคอมพิวเตอร์). กรุงเทพฯ : บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ.
- วรุตม์ ขยันกิจ. (2552). การออกแบบร่วมฮาร์ดแวร์/ซอฟต์แวร์สำหรับการประมวลผลหา เส้นตรงบนภาพด้วยเอฟพีจีเอ. **Hardware/Software Co-design for Line Detection Algorithm on FPGA**. วิทยานิพนธ์ วศ.ม. (วิศวกรรมคอมพิวเตอร์). สงขลา : บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์.
- สัณห์ อุทัยรัตน์. (2549). การวิจัยและพัฒนาส่วนประมวลผลสัญญาณดิจิทัลของระบบสื่อสารในสถานีภาคพื้นดินสำหรับดาวเทียมอเนกประสงค์ขนาดเล็ก. วิทยานิพนธ์ วศ.ม. (วิศวกรรมไฟฟ้า). กรุงเทพฯ : บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์.
- A.S. Godbole. (2003). **Data Compression and Encryption**. New Delhi : Tata McGraw-Hill
- Canny, J. (1986, November). A Computational Approach to Edge Detection. **IEEE Transaction on Pattern Analysis and Machine Intelligence**. 8 (6) : 679-698.
- Chia-Chi Teng; Linda G. Shapiro; and Ira Kalet. (2006). **Automatic Segmentation of Neck CT Images**. Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems. PP. 442-445.
- Claude Shannon. (1948, July and October). A Mathematical Theory of Communication. **The Bell System Technical Journal**. 27 : 379-423, 623-656.
- Digital Systems Research Center. (1994). **A Block-sorting Lossless Data Compression Algorithm**. California : Digital Systems Research Center.

- Frei; and Chen (1977, October). Fast Boundary Detection: A Generalization and New Algorithm. **IEEE Transactions on Computers**. 26 (10) : 988-998.
- Gonzalez, R. C.; Woods, R. E.; and Eddins, S. L. (2010). **Digital Image Processing Using MATLAB. 3rd ed.** USA : Gatesmark Publishing.
- Huffman. (1952, September). A Method for the Construction of Minimum Redundancy Codes. **Proceedings of the Institute of Radio Engineer**. 40 (9) : 1,098-1,101.
- Jacob Ziv; and Abraham Lempel. (1978, September). Compression of Individual Sequences via Variable-Rate Coding. **IEEE Transactions on Information Theory**. 24 (5) : 530-536.
- John Canny. (1986, November). A computational approach to edge detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, PAMI-8 (6) : 679-698.
- J. G. Cleary; and I. H. Witten. (1984, April). Data Compression Using Adaptive Coding and Partial String Matching. **IEEE Transactions on Communications**. 32 (4) : 396-402.
- Kirsch, R. (1971, June). Computer Determination of the Constituent Structure of Biological Images. **Computers and Biomedical Research**. 4 (3) : 315–328.
- Prewitt. J. (1970). **Object Enhancemet And Extraction Picture Processing and Psychopictorics**. New Yoke : NY Academic.
- Qian & Huang. (1996, July). Optimal Edge Detection in Two-Dimensional Images. **IEEE Transactions on Image Processing**. 5 (7) : 1,215–1,220.
- Riccardo Boscolo; Matthew S. Brown; and Michael F. McNitt - Gray. (2002, March-April). Medical Image Segmentation with Knowledge-guided Robust Active Contours. **US National Library of Medicine Radiographics**. 22 (2) : 437-448.
- Roberts, L.G. (1963). **Machine Perception of Three-Dimensional Solids**. Thesis Ph.D. (Electrical Engineering). Massachusetts : Department of Electrical Engineering Massachusetts Institute of Technology.
- Robinson, G.S. (1976, August). Detection and Coding of Edges using Directional Masks. **SPIE Journals and Proceedings Optical Engineering (Advances in Image Transmission Techniques)**. 16 : 580-585.

Song Wang; Feng Ge; and Tiecheng Liu. (2006, January). Evaluating Edge Detection through Boundary Detection. **EURASIP Journal on Applied Signal Processing**. 2006 : 213-213.

Sven Loncaric; and Kovacevic. (1997, September). A Method for Segmentation of CT Head Images. **Image Analysis and Processing**. 2 : 388-395.

S.R. Kodituwakku. (2010). Comparison of Lossless Data Compression Algorithms for Text Data. **Indian Journal of Computer Science and Engineering**. 1 (4) : 416-425.

Z.J. Hou; and G.W. Wei. (2002). A New Approach to Edge Detection. **The Journal of the Pattern Recognition Society**. 35 : 1,559-1,570.





ภาคผนวก
ผลงานตีพิมพ์

การพัฒนาซอฟต์แวร์การแบ่งส่วนภาพและบีบอัดข้อมูลภาพเพื่อเพิ่มประสิทธิภาพการส่งข้อมูลผ่านอินเทอร์เน็ต

Software Development on Image Segmentation and Data Compression for Efficiency Enhancement in Data Transmission over Internet

ไพศาล ศรีพระราม¹ และ วรากร ศรีเชวงทรัพย์²

¹สาขาเทคโนโลยีวิศวกรรม บัณฑิตวิทยาลัย สถาบันเทคโนโลยีไทย-ญี่ปุ่น

1771/1 ซ.พัฒนาการ 37 ถนนพัฒนาการ แขวงสวนหลวง เขตสวนหลวง กรุงเทพฯ 10250

โทร. 0-2763-2600 โทรสาร 0-2763-2700 E-Mail: confi.safe@gmail.com

²สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีไทย-ญี่ปุ่น

1771/1 ซ.พัฒนาการ 37 ถนนพัฒนาการ แขวงสวนหลวง เขตสวนหลวง กรุงเทพฯ 10250

โทร. 0-2763-2600 โทรสาร 0-2763-2700 E-Mail: warakorn417@gmail.com

บทคัดย่อ

บทความนี้นำเสนอการพัฒนาซอฟต์แวร์เพื่อลดขนาดข้อมูลภาพ โดยใช้เทคนิคการหาขอบภาพด้วยวิธี Canny และการแบ่งส่วนภาพเพื่อแยกและตัดข้อมูลส่วนที่ผู้ใช้ไม่ต้องการออก นำข้อมูลภาพส่วนที่สนใจมาใช้ พร้อมทั้งออกแบบรวมกับการบีบอัดข้อมูลภาพโดยใช้หลักการของ Zlib Deflate เพื่อบีบอัดข้อมูลให้มีขนาดเล็กลงมากขึ้น ซึ่งจะทำให้การส่งข้อมูลผ่านอินเทอร์เน็ตทำได้รวดเร็วและมุ่งเน้นให้การสื่อสารผ่านอินเทอร์เน็ตมีความราบรื่นมากขึ้น ซึ่งในบทความนี้จะแสดงการออกแบบซอฟต์แวร์ที่พัฒนาขึ้นด้วยโปรแกรม MATLAB ผลที่ได้จากการแบ่งส่วนภาพจะทำให้ข้อมูลภาพมีขนาดลดลง และส่งผลให้การส่งข้อมูลผ่านอินเทอร์เน็ตมีความต่อเนื่องมากขึ้น

คำสำคัญ: การแบ่งส่วนภาพ, การบีบอัดข้อมูลภาพ

Abstract

This article describes a software development on image data reduction by using Canny edge detection and image segmentation in order to remove unnecessary data such as image background. Zlib Deflate compression algorithm is used to achieve an effective data reduction capability. This software is developed using MATLAB programming language and MATLAB GUI. Results show that the developed software can reduce the transmitted data size and increase the frame rate.

Keywords: Image Segmentation, Data Compression

1. คำนำ

ในปัจจุบันการส่งผ่านข้อมูลในระบบเครือข่ายอินเทอร์เน็ตได้เข้ามามีบทบาทต่อชีวิตประจำวันมากขึ้น ซึ่งเป็นผลมาจากที่เทคโนโลยีดิจิทัลที่ได้ถูกนำมาใช้เป็นจำนวนมาก ทั้งการส่งผ่านทางระบบคอมพิวเตอร์ และอุปกรณ์พกพา ปัญหาที่มีความสำคัญอย่างหนึ่งก็คือ ข้อจำกัดในเรื่องของอัตราการส่งผ่านข้อมูลบนอินเทอร์เน็ต ส่งผลทำให้การส่งข้อมูลภาพที่มีขนาดใหญ่มีปัญหาการแสดงผลภาพที่ไม่ต่อเนื่อง ซึ่งอาจทำให้เกิดปัญหาเรื่องความผิดพลาดและไม่ราบรื่นในการสื่อสารได้

ปัจจัยหลักที่มีผลกระทบสำคัญต่อการส่งข้อมูลทางอินเทอร์เน็ตเช่นความเร็วของสัญญาณอินเทอร์เน็ตซึ่งในบางกรณีที่มีผู้ใช้จำนวนมากในเวลาเดียวกัน อาจทำให้สัญญาณอินเทอร์เน็ตมีความเร็วลดลง ปัจจัยหลักอีกประการหนึ่งคือปริมาณ หรือขนาดของข้อมูลที่จะถูกส่งผ่านอินเทอร์เน็ต ก็มีผลโดยตรงต่อความเร็วในส่งข้อมูล ซึ่งในปัจจุบันวิธีการต่าง ๆ ของการประมวลผลภาพก็ถูกนำมาใช้ในการแก้ปัญหา เช่นการพัฒนาความเร็วของอัตราการส่งสัญญาณ และการทำข้อมูลให้มีขนาดเล็กลงเพื่อให้เหมาะสมต่อการจัดเก็บข้อมูล และการส่งผ่านข้อมูลภายในระบบเครือข่ายอินเทอร์เน็ต

ในบทความนี้ได้นำเสนอวิธีการที่รวมการใช้เทคนิคการแบ่งส่วนภาพมาใช้ร่วมกับการบีบอัดข้อมูลภาพ เพื่อทำให้การลดขนาดของข้อมูลภาพทำ

ได้มากขึ้น ส่งผลให้การส่งข้อมูล และการสื่อสารผ่านอินเทอร์เน็ตมีประสิทธิภาพและมีความราบรื่นมากขึ้น

2. ทฤษฎีที่เกี่ยวข้อง

2.1 ระบบภาพ

จุดภาพหรือพิกเซล (Pixel) เป็นหน่วยพื้นฐานที่มีขนาดเล็กที่สุดของภาพ ในรูปภาพจะมีจุดภาพหลาย ๆ จุด ที่ถูกรวมกันเป็นภาพหนึ่ง ๆ โดยภาพแต่ละภาพที่สร้างขึ้นจะมีความหนาแน่นของจุดภาพ หรือบางครั้งแทนว่าความละเอียด หรือความคมชัด ที่แตกต่างกันไป คำว่า "พิกเซล" (Pixel) มาจากคำว่า "พิกเจอร์" (Picture) ที่แปลว่ารูปภาพ และ "เอลิเมนต์" (Element) ที่แปลว่าองค์ประกอบ จุดภาพจึงใช้ในการบอกคุณสมบัติของภาพ จอภาพ หรืออุปกรณ์แสดงผลภาพใด ๆ จอภาพที่สามารถแสดงจุดภาพได้มาก จะมีความละเอียดของภาพมาก โดยมากจำนวนจุดภาพจะถูกระบุในลักษณะ แนวนอน (x) คูณแนวตั้ง (y) เช่น 1366 x 768 จุดภาพ

ในภาพหนึ่ง ๆ เราสามารถอธิบายได้เป็นเมตริกซ์ของจุดภาพขนาด $N \times M$ โดยใช้คู่ลำดับ (i, j) แทนค่าของจุดแต่ละจุด และบ่งชี้ความเข้มสีที่จุดภาพนั้น ๆ เมตริกซ์ของจุดภาพแสดงได้ดังรูปที่ 1

	0	j $\longrightarrow$ M						
0	(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)	(0,6)	(0,7)
	(1,0)	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)
i	(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)	(2,7)
	(3,0)	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)	(3,7)
	(4,0)	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)	(4,7)
N	(5,0)	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	(5,7)

รูปที่ 1 เมตริกซ์ของจุดภาพ

2.2 การแบ่งส่วนภาพ

การแบ่งย่อยส่วนประกอบภาพจะใช้ในการแยกองค์ประกอบต่าง ๆ ของรูปภาพออกจากกันตามลักษณะสำคัญที่เราพิจารณา ขั้นตอนนี้เป็นขั้นตอนสำคัญของการวิเคราะห์ภาพ ประโยชน์ของการแบ่งย่อยส่วนประกอบภาพคือ จะทำให้สามารถลดจำนวนข้อมูลในรูปภาพ โดยการคัดแยกส่วนข้อมูลภาพที่ไม่ต้องการ เนื่องจากการแยกแยะระหว่างส่วนที่เราสนใจ เช่นวัตถุในภาพ กับส่วนที่ไม่ต้องการ เช่นฉากหลัง และเมื่อตัดข้อมูลในส่วนที่ไม่ต้องการออกไปจำนวนข้อมูลที่เหลือที่จำเป็นในการ

วิเคราะห์จริงจะลดลงอย่างมาก และจัดระเบียบข้อมูลในรูปภาพให้เป็นกลุ่มได้ดีขึ้น ข้อมูลภาพที่ผ่านการแบ่งแยกแล้ว จะมีโครงสร้างที่ชัดเจนขึ้นและนำไปใช้งานได้สะดวกขึ้น

จากการศึกษาในเรื่องการแบ่งย่อยส่วนประกอบภาพ ผู้วิจัยพบว่าภาพที่ถูกถ่ายเพื่อนำมาใช้ในการสื่อสารใด ๆ ที่มีความแตกต่างของคุณสมบัติภาพระหว่างอวัยวะต่าง ๆ ของมนุษย์กับภาพพื้นหลัง ซึ่งจะสามารถนำมาแยกแยะภาพพื้นหลังกับภาพที่เราสนใจออกมาได้ หลักการและวิธีการที่เหมาะสมในการแยกระหว่างภาพคนกับภาพพื้นหลังต่าง ๆ น่าจะเป็นวิธีการ ที่เกิดจากการวิเคราะห์คุณสมบัติของภาพ และใช้คุณสมบัติต่าง ๆ เหล่านี้มาประมวลผลและแยกแยะ แล้วนำมาใช้กับหลักการและวิธีการต่าง ๆ ได้

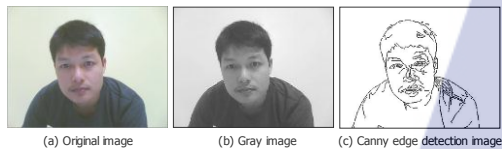
Edge Oriented Image Segmentation เป็นวิธีการแยกส่วนหรือองค์ประกอบของภาพโดยอาศัยความไม่ต่อเนื่อง ของคุณสมบัติของจุดภาพบริเวณรอยต่อระหว่างวัตถุใด ๆ วิธีการนี้มุ่งที่ขอบของวัตถุเป็นหลัก ผลลัพธ์ที่ได้จากวิธีการนี้จะอยู่ในรูปเส้นพรมแดนระหว่างพื้นที่ต่าง ๆ

การหาขอบภาพคือการตรวจสอบว่าเส้นขอบลากผ่านหรือใกล้เคียงกับจุดใด โดยวัดจากการเปลี่ยนแปลงของความเข้มในตำแหน่งที่ใกล้เคียงกับจุดดังกล่าว ซึ่งวิธีการหาขอบนั้นมีด้วยกันหลายวิธี แต่อย่างไรก็ตามสามารถแบ่งได้เป็น 2 กลุ่มหลัก คือการค้นหาขอบภาพโดยการใช้อนุพันธ์อันดับหนึ่ง (Gradient method) และการค้นหาขอบภาพโดยการใช้อนุพันธ์อันดับสอง (Laplacian method) [1]

โปรแกรมแมทแลป (MATLAB) สามารถช่วยในการหาขอบภาพ ซึ่งมีคำสั่งที่ใช้ในการหาขอบทั้งหมด 6 วิธี คือ Roberts, Sobel, Canny, Laplacian of Gaussian, Zero Cross และ Prewitt งานวิจัยตัวอย่างนี้ได้เลือกวิธี Canny [2] ในการหาขอบภาพ เนื่องจากวิธีดังกล่าวมีการใช้ Gaussian Filter ก่อนการหาขอบจึงสามารถควบคุมระดับความละเอียดของขอบที่ต้องการและสามารถลดสัญญาณรบกวนได้อีกด้วย ทำให้สามารถตัดขั้นตอนการประมวลผลภาพเบื้องต้น และด้วยวิธี Canny จะให้รายละเอียดของขอบวัตถุได้ดีที่สุด และสามารถใช้ได้ในการกรณีที่ความแตกต่างของสีมีน้อยเมื่อเทียบกับวิธีอื่น ๆ

การทำงานของ Canny Edge Detection นั้น เริ่มต้นจากการปรับภาพให้เรียบ (Smoothing) ด้วยตัวกรองเกาส์เซียน (Gaussian Filter) เพื่อกำจัดสัญญาณรบกวน หลังจากนั้น คำนวณค่าขนาด (Magnitude) และทิศทาง (Orientation) ของ Gradient โดยใช้การหา

อนุพันธ์อันดับหนึ่ง ต่อมาจึงใช้ No Maxima Suppression กับ Gradient Magnitude เพื่อให้ได้ขอบที่บางลง และในขั้นตอนสุดท้ายใช้ Double Thresholding Algorithm เพื่อระบุพิกเซลที่เป็นขอบและช่วยเชื่อมต่อขอบ



รูปที่ 2 การหาขอบภาพด้วยวิธีของ Canny

2.3 การบีบอัดข้อมูล

การบีบอัดข้อมูลเป็นเทคนิคหนึ่งในการลดขนาดของข้อมูล ที่ว่าด้วยการศึกษาวิธีการในการจัดเก็บข้อมูลที่ทำให้ข้อมูลมีขนาดเล็กลง การบีบอัดข้อมูล มีความสำคัญในระบบการสื่อสารเนื่องจากทำให้รับส่งข้อมูลได้มากขึ้น ในขณะที่มีความเร็วของสัญญาณอินเทอร์เน็ตเท่าเดิม

การบีบอัดข้อมูลมีประโยชน์ในการลดปริมาณการใช้ทรัพยากร เช่น ประหยัดพื้นที่จัดเก็บของฮาร์ดดิสก์ หรือใช้แบนด์วิดท์ของระบบเครือข่ายน้อยลงเพื่อส่งข้อมูลที่ถูกระบุบีบอัดให้มีขนาดลดลงแล้วเป็นต้น

การบีบอัดข้อมูลภาพโดยทั่วไปแล้วจะแบ่งออกเป็น 2 ประเภทคือ

1. การบีบอัดแบบสูญเสียรายละเอียดข้อมูล (Lossy compression) วิธีการนี้จะมีการเปลี่ยนแปลงค่าความสว่างของจุดภาพ จะส่วนมากจะทำให้สามารถลดขนาดของข้อมูลภาพได้มาก แต่วิธีการนี้ไม่เหมาะสมสำหรับข้อมูลภาพที่ต้องมีการจำแนกข้อมูล (Classification) ตัวอย่างของการบีบอัดประเภทนี้ได้แก่ JPEG, MrSID หรือ ECW เป็นต้น

2. การบีบอัดแบบไม่มีการสูญเสียรายละเอียดข้อมูล (Lossless Compression) วิธีการนี้จะทำให้ค่าของแต่ละจุดภาพจะยังคงอยู่เหมือนเดิม หรือไม่มีการเปลี่ยนแปลงค่าของแต่ละจุดภาพ ซึ่งจะทำให้ข้อมูลภาพที่ถูกถอดรหัสแล้วยังคงเป็นเหมือนภาพต้นฉบับ ซึ่งการบีบอัดวิธีนี้จะอาศัยเทคนิคการจัดเก็บข้อมูลเชิงตัวเลขในการลดขนาดของข้อมูล ตัวอย่างเช่น Huffman code, Lempel-Ziv 1977 (LZ77), Lempel-Ziv-Welch (LZW), Run-Length Encoding (RLE), PackBits และ Zlib Deflate เป็นต้น

ในงานวิจัยนี้จะใช้วิธีการบีบอัดข้อมูลแบบ Lossless Compression โดยใช้หลักการของ Zlib Deflate ซึ่งจะอาศัยการรวมกันของ Huffman code และ Lempel-

Ziv 1977 (LZ77) [3] [4] [5] ที่ออกแบบมาแทนที่ Lempel-Ziv-Welch (LZW) มีข้อดีตรงที่สามารถบีบอัดได้อย่างอิสระ และมีประสิทธิภาพสูง

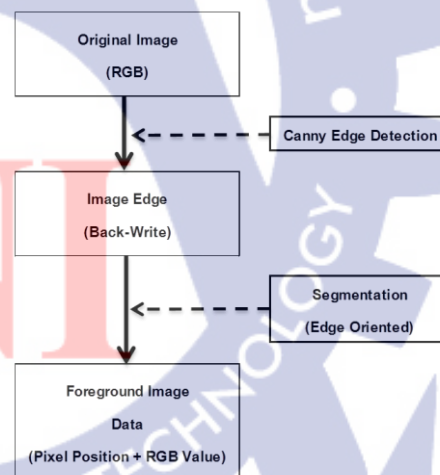
3. การออกแบบซอฟต์แวร์

3.1 การแบ่งส่วนภาพ (Image Segmentation)

การออกแบบจะเริ่มจากการประมวลผลภาพต้นแบบ (Input Image) ที่มาจากเฟรมภาพของเว็บแคม และทำการแยกส่วนภาพคนและภาพพื้นหลังออกจากกัน จากนั้นเขียนโปรแกรมเลือกเฉพาะตำแหน่งและค่าของจุดภาพที่ต้องการ เพื่อลดจำนวนข้อมูลในรูปภาพส่วนที่ไม่ต้องการลง

เนื่องจากการแบ่งย่อยส่วนประกอบของภาพเป็นการแยกแยะระหว่างส่วนที่เราสนใจ (Foreground) ซึ่งในกรณีนี้จะเป็นภาพของคน กับส่วนที่ไม่ต้องการ (Background) ซึ่งในกรณีนี้จะเป็นภาพของฉากหลัง ซึ่งการออกแบบการแบ่งส่วนภาพจะใช้ขั้นตอนวิธีดังรูปที่ 3

การออกแบบโปรแกรมจะเริ่มต้นจากการรับภาพสี (RGB) มาจากกล้องเว็บแคมด้วยขนาด 120x160x3 จุดภาพ จากนั้นหาขอบภาพโดยใช้วิธี Canny เพื่อที่จะสามารถหาตำแหน่งและเก็บข้อมูลของภาพส่วนที่ต้องการได้

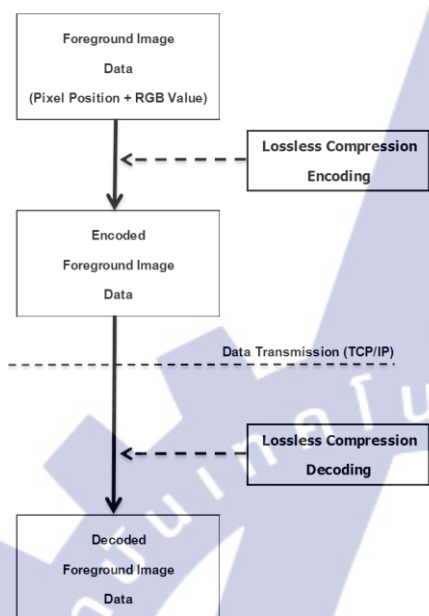


รูปที่ 3 การออกแบบโปรแกรมการแบ่งส่วนภาพ

3.2 การบีบอัดข้อมูลภาพ (Data Compression)

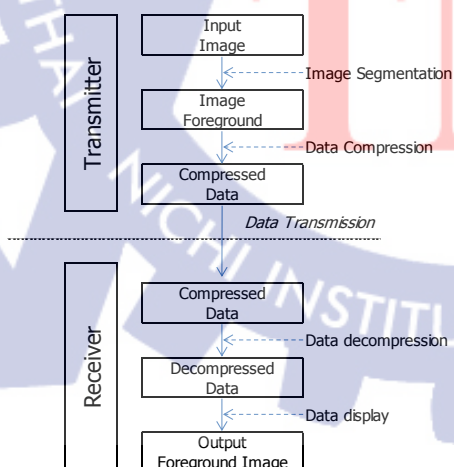
ข้อมูลรูปภาพส่วนที่สนใจ (Foreground) จะถูกนำมาทำการบีบอัดข้อมูลแบบไม่มีการสูญเสียรายละเอียดข้อมูล (Lossless Compression) เพื่อทำการลดจำนวนของข้อมูล โดยที่จะไม่ส่งผลกระทบต่อคุณภาพของข้อมูล ซึ่งการออกแบบ

การบีบอัดข้อมูลภาพจะเป็นไปตามขั้นตอนดังรูปที่ 4 ในขณะเดียวกัน ส่วนของภาครับหลังจากการส่งผ่านข้อมูลทางอินเทอร์เน็ตแล้ว ก็จะต้องเขียนการถอดรหัสข้อมูลเพื่อเตรียมพร้อมที่จะแสดงผลข้อมูลภาพต่อไป



รูปที่ 4 การออกแบบการบีบอัดข้อมูลภาพ

ภาพรวมของการเขียนโปรแกรมการลดขนาดข้อมูลภาพโดยใช้เทคนิคของการแบ่งส่วนภาพร่วมกับ การบีบอัดข้อมูลภาพจะสามารถแสดงได้ดังรูปที่ 5



รูปที่ 5 ภาพรวมของโปรแกรมการลดขนาดข้อมูลภาพ

ภาพรวมการออกแบบโปรแกรมจะแบ่งการทำงานเป็นสองส่วนได้แก่ภาคส่ง (Transmitter) และภาครับ (Receiver) โดยในส่วนของภาคส่งจะทำการเลือกค่าตำแหน่งจุดภาพและค่าความเข้มของแต่ละชั้นสี (RGB) ก่อนนำมาบีบอัดและส่งเป็นข้อมูลไปยังภาครับ ซึ่งภาครับก็จะทำการถอดรหัสข้อมูลเพื่อจะแสดงผลข้อมูลภาพเฉพาะส่วนภาพที่เราต้องการออกมาเป็นภาพสีในลักษณะเดียวกันกับภาพต้นแบบ แต่จะแสดงเฉพาะส่วนที่ต้องการ

4. ผลการทดลอง

จากการออกแบบซอฟต์แวร์และทดสอบวัดค่าของผลที่ได้ พบว่าโปรแกรมที่ออกแบบสามารถแสดงส่วนภาพที่สนใจและตัดส่วนภาพพื้นหลังออกได้ และจากการทดสอบ 40 ครั้ง พบว่าการแบ่งส่วนภาพเพื่อคัดแยกข้อมูลเฉพาะที่เราสนใจ สามารถลดขนาดของข้อมูลภาพลงได้เฉลี่ย 68% และจากการบีบอัดข้อมูลส่วนที่เราสนใจจะสามารถลดขนาดลงได้เฉลี่ย 15% ส่งผลให้เวลาที่ใช้ในการส่งข้อมูลภาพผ่านอินเทอร์เน็ตลดลงจาก 0.5853 วินาทีต่อเฟรม เป็น 0.3177 วินาทีต่อเฟรม หรือเทียบได้กับการส่งข้อมูลภาพที่เพิ่มขึ้นจาก 1.7085 เฟรมต่อวินาที เป็น 3.1470 เฟรมต่อวินาที



รูปที่ 6 ตัวอย่างผลของการส่งข้อมูลภาพด้วยฉากหลังสีอ่อน

จากรูปที่ 6 (บน) แสดงหน้าต่าง GUI ของผลที่ได้จากการทดลองส่งข้อมูลภาพปกติด้วยฉากหลังสีอ่อน

ขนาดข้อมูลของภาพมีค่า 57,600 ไบต์ ใช้เวลาในการส่งข้อมูลภาพ 0.3970 วินาทีต่อเฟรม

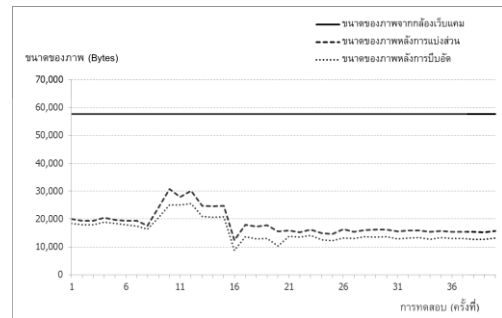
รูปที่ 6 (ล่าง) แสดงหน้าต่าง GUI ของผลที่ได้จากการทดลองส่งข้อมูลภาพที่ได้จากการแบ่งส่วนภาพและบีบอัดข้อมูลภาพ ขนาดข้อมูลของภาพมีค่า 19,383 ไบต์ สัดส่วนของข้อมูลภาพที่ถูกแบ่งส่วนต่อภาพตั้งต้นมีค่า 0.3365 และสัดส่วนของข้อมูลที่ถูกบีบอัดแล้วต่อข้อมูลก่อนการบีบอัดมีค่า 0.8981 ใช้เวลาในการส่งข้อมูลภาพ 0.2598 วินาทีต่อเฟรม



รูปที่ 7 ตัวอย่างผลของการส่งข้อมูลภาพด้วยฉากหลังสีเข้ม

จากรูปที่ 7 (บน) แสดงหน้าต่าง GUI ของผลที่ได้จากการทดลองส่งข้อมูลภาพปกติด้วยฉากหลังสีเข้ม ขนาดข้อมูลของภาพมีค่า 57,600 ไบต์ ใช้เวลาในการส่งข้อมูลภาพ 0.57497 วินาทีต่อเฟรม

รูปที่ 7 (ล่าง) แสดงหน้าต่าง GUI ของผลที่ได้จากการทดลองส่งข้อมูลภาพที่ได้จากการแบ่งส่วนภาพและบีบอัดข้อมูลภาพ ขนาดข้อมูลของภาพมีค่า 16,125 ไบต์ สัดส่วนของข้อมูลภาพที่ถูกแบ่งส่วนต่อภาพตั้งต้นมีค่า 0.2799 และสัดส่วนของข้อมูลที่ถูกบีบอัดแล้วต่อข้อมูลก่อนการบีบอัดมีค่า 0.8791 ใช้เวลาในการส่งข้อมูลภาพ 0.3199 วินาทีต่อเฟรม



รูปที่ 8 ผลการทดสอบขนาดข้อมูลที่ถูกแบ่งส่วนและบีบอัด

จากข้อมูลที่ได้จากการทดสอบ จะเห็นได้ว่าขนาดข้อมูลภาพที่ได้จากกล้องเว็บแคมมีค่าคงที่ ที่ 57,600 ไบต์ และค่าข้อมูลภาพที่ได้จากการแบ่งส่วนจะมีค่าเฉลี่ยของอัตราการแบ่งส่วนข้อมูลภาพจากการทดสอบทั้งหมด 40 ครั้งอยู่ที่ 0.3192 หรือสามารถลดขนาดข้อมูลภาพลดลงเหลือร้อยละ 31.92

การบีบอัดข้อมูลภาพหลังจากการแบ่งส่วนภาพทำให้ข้อมูลภาพมีค่าเฉลี่ยอยู่ที่ 0.2709 หรือสามารถลดขนาดข้อมูลภาพลดลงเหลือร้อยละ 27.09 เมื่อเทียบกับข้อมูลของภาพตั้งต้น

5. สรุป

งานวิจัยนี้ได้พัฒนาซอฟต์แวร์เพื่อเพิ่มประสิทธิภาพในการส่งผ่านข้อมูลทางอินเทอร์เน็ต ด้วยการลดขนาดข้อมูลภาพ โดยเทคนิคการแบ่งส่วนภาพ ที่อาศัยวิธีการหาขอบภาพของ Canny และเลือกส่วนข้อมูลที่สนใจและบีบอัดข้อมูลให้มีขนาดเล็กลง ผลที่ได้คือสามารถส่งข้อมูลผ่านอินเทอร์เน็ตได้เร็วและมีความต่อเนื่องมากขึ้น

ผลการทดสอบกับภาพจากกล้องเว็บแคม เวลาที่ใช้ในการส่งภาพแบบปกติที่ใช้เวลาเฉลี่ย 0.5853 วินาทีต่อเฟรม ลดลงเหลือ 0.3177 วินาทีต่อเฟรม หรือเทียบได้กับอัตราการส่งข้อมูลภาพที่เพิ่มขึ้นจาก 1.7085 เฟรมต่อวินาที เป็น 3.1470 เฟรมต่อวินาที ทั้งนี้อัตราความเร็วก็ยังขึ้นกับอัตราความเร็วของระบบเครือข่ายอินเทอร์เน็ตด้วยเช่นกัน

ซอฟต์แวร์ที่ได้พัฒนาขึ้นนี้ สามารถนำไปประยุกต์ใช้กับการสื่อสารที่มีการรับส่งข้อมูลภาพที่ต้องการลดภาระการเก็บและประมวลผลข้อมูลภาพที่มีขนาดใหญ่ได้เช่น การส่งข้อมูลภาพของคนไข้เพื่อการรักษาที่เร็วขึ้น การส่งภาพวีดิโอสาธิตการทำงาน อีกทั้งยังใช้ได้กับการสื่อสารในลักษณะของเว็บแคมได้อีกด้วย

เอกสารอ้างอิง

- [1] Gonzalez, R. C., Woods, R. E., and Eddins, S. L. [2010]. Digital Image Processing Using MATLAB, 3rd ed., Gatesmark Publishing, Knoxville, TN.
- [2] John Canny. A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-8(6):679–698, November 1986.
- [3] A.S. Godbole, Data Compression and Encryption, Data Communication and Networks, International Edition, 2003.
- [4] Ken Huffman. Profile: David A. Huffman, Scientific American, pp. 54–58, September 1991.
- [5] Jacob Ziv and Abraham Lempel; Compression of Individual Sequences via Variable-Rate Coding, IEEE Transactions on Information Theory, September 1978.

