

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

²สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีไทย-ญี่ปุ่น

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

โทร. 0-2763-2600 โทรสาร 0-2763-2700 E-Mail: warakorn@tni.ac.th

บทคัดย่อ

บทความนี้นำเสนอการพัฒนาซอฟต์แวร์เพื่อลดขนาดข้อมูลภาพโดยใช้เทคนิคการหาขอบภาพด้วยวิธี Canny และการแบ่งส่วนภาพเพื่อแยกและตัดข้อมูลส่วนที่ผู้ใช้ไม่ต้องการออก นำข้อมูลภาพส่วนที่สนใจมาใช้ พร้อมทั้งออกแบบรวมกับการบีบอัดข้อมูลภาพโดยใช้หลักการของ Zlib Deflate เพื่อบีบอัดข้อมูลให้มีขนาดเล็กลงมากขึ้น ซึ่งจะทำให้การส่งข้อมูลผ่านอินเทอร์เน็ตทำได้รวดเร็ว และมุ่งเน้นให้การสื่อสารผ่านอินเทอร์เน็ตมีความราบรื่นมากขึ้น 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 ซึ่งเป็นผลมาจากที่เทคโนโลยีดิจิทัลได้ถูกนำมาใช้เป็นจำนวนมาก ทั้งการส่งผ่านทางระบบคอมพิวเตอร์ และอุปกรณ์พกพา ปัญหาที่มีความสำคัญอย่างหนึ่งก็คือข้อจำกัดในเรื่องของอัตราการส่งผ่านข้อมูลบนอินเทอร์เน็ต ส่งผลทำให้การส่งข้อมูลภาพที่มีขนาดใหญ่มีปัญหาการแสดงผลภาพที่ไม่ต่อเนื่อง ซึ่งอาจทำให้เกิดปัญหาเรื่องความผิดพลาดและไม่ราบรื่นในการสื่อสารได้

ปัจจัยหลักที่มีผลกระทบต่ออัตราการส่งข้อมูลทางอินเทอร์เน็ตเช่นความเร็วของสัญญาณอินเทอร์เน็ตซึ่งในบางกรณีที่มีผู้ใช้จำนวนมากในเวลาเดียวกัน อาจทำให้สัญญาณอินเทอร์เน็ตมีความเร็วลดลง ปัจจัยหลักอีกประการหนึ่งคือปริมาณ หรือขนาดของข้อมูลที่จะถูกส่งผ่านอินเทอร์เน็ต ก็มีผลโดยตรงต่อความเร็วในส่งข้อมูล ซึ่งในปัจจุบันวิธีการต่าง ๆ ของการประมวลผลภาพก็ถูกนำมาใช้ในการแก้ปัญหา เช่นการพัฒนาความเร็วของอัตราการส่งสัญญาณ และการทำข้อมูลให้มีขนาดเล็กลง เพื่อให้เหมาะสมต่อการจัดเก็บข้อมูล และการส่งผ่านข้อมูลภายในระบบเครือข่ายอินเทอร์เน็ต

ในบทความนี้ได้นำเสนอวิธีการที่รวมการใช้เทคนิคการแบ่งส่วนภาพมาใช้ร่วมกับการบีบอัดข้อมูลภาพ เพื่อทำให้การลดขนาดของข้อมูลภาพทำได้มากขึ้น ส่งผลให้การส่งข้อมูล และการสื่อสารผ่านอินเทอร์เน็ตมีประสิทธิภาพและมีความราบรื่นมากขึ้น

2. ทฤษฎีที่เกี่ยวข้อง

2.1 ระบบภาพ

จุดภาพหรือพิกเซล (Pixel) เป็นหน่วยพื้นฐานที่มีขนาดเล็กที่สุดของภาพ ในรูปภาพจะมีจุดภาพหลาย ๆ จุด ที่ถูกรวมกันเป็นภาพหนึ่ง ๆ โดยภาพแต่ละภาพที่สร้างขึ้นจะมีความหนาแน่นของจุดภาพ หรือบางครั้ง

แทนว่าความละเอียด หรือความคมชัด ที่แตกต่างกันไป คำว่า "พิกเซล" (Pixel) มาจากคำว่า "พิกเจอร์" (Picture) ที่แปลว่ารูปภาพ และ "เอลิเมนต์" (Element) ที่แปลว่าองค์ประกอบ จุดภาพจึงใช้ในการบอกคุณสมบัติของภาพ จอภาพ หรืออุปกรณ์แสดงผลภาพใด ๆ จอภาพที่สามารถแสดงจุดภาพได้มาก จะมีความละเอียดของภาพมาก โดยมากจำนวนจุดภาพจะถูกระบุในลักษณะ แนวนอน (x) คู่แนวตั้ง (y) เช่น 1366 x 768 จุดภาพ

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

	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 1 เมตริกซ์ของจุดภาพ

2.2 การแบ่งส่วนภาพ

การแบ่งย่อยส่วนประกอบภาพจะใช้ในการแยกองค์ประกอบต่าง ๆ ของรูปภาพออกจากกันตามลักษณะสำคัญที่เราพิจารณา ขั้นตอนนี้เป็นขั้นตอนสำคัญของการวิเคราะห์ภาพ ประโยชน์ของการแบ่งย่อยส่วนประกอบภาพคือ จะทำให้สามารถลดจำนวนข้อมูลในรูปภาพ โดยการตัดแยกส่วนข้อมูลภาพที่ไม่ต้องการ เนื่องจากการแยกแยะระหว่างส่วนที่เราสนใจ เช่นวัตถุในภาพ กับส่วนที่ไม่ต้องการ เช่นฉากหลัง และเมื่อตัดข้อมูลในส่วนที่ไม่ต้องการออกไปจำนวนข้อมูลที่เหลือที่จำเป็นในการวิเคราะห์จึงจะลดลงอย่างมาก 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 ของคุณสมบัติของจุดภาพ

บริเวณรอยต่อระหว่างวัตถุใด 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 และสามารถใช้ได้ในกรณีที่ความแตกต่างของสีมีน้อยเมื่อเทียบกับวิธีอื่น ๆ

การทำงานของ Canny Edge Detection นั้นเริ่มต้นจากการปรับภาพให้เรียบ (Smoothing) ด้วยตัวกรองเกาส์เซียน (Gaussian Filter) เพื่อกำจัดสัญญาณรบกวน หลังจากนั้น คำนวณค่าขนาด (Magnitude) และทิศทาง (Orientation) ของ Gradient โดยใช้การหาอนุพันธ์อันดับหนึ่ง ต่อมาจึงใช้ No Maxima Suppression กับ Gradient Magnitude เพื่อทำให้ได้ขอบที่บางลง และในขั้นตอนสุดท้ายใช้ Double Thresholding Algorithm เพื่อระบุพิกเซลที่เป็นขอบและช่วยเชื่อมต่อขอบ



รูปที่ 2 การหาขอบภาพด้วยวิธีของ Canny

2.3 การบีบอัดข้อมูล

การบีบอัดข้อมูลเป็นเทคนิคหนึ่งในการลดขนาดของข้อมูล ที่ว่าด้วยการศึกษาวิธีการในการจัดเก็บข้อมูล ที่ทำให้ข้อมูลมีขนาดเล็กลง การบีบอัดข้อมูล มีความสำคัญในระบบการสื่อสารเนื่องจากทำให้รับส่งข้อมูลได้มากขึ้น ในขณะที่มีความเร็วของสัญญาณอินเทอร์เน็ตเท่าเดิม

การบีบอัดข้อมูลมีประโยชน์ในการลดปริมาณการใช้ทรัพยากร เช่น ประหยัดพื้นที่จัดเก็บของฮาร์ดดิสก์ หรือใช้แบนด์วิดธ์ของระบบเครือข่ายน้อยลงเพื่อส่งข้อมูลที่ถูกบีบอัดให้มีขนาดลดลงแล้วเป็นต้น

การบีบอัดข้อมูลภาพโดยทั่วไปแล้วจะแบ่งออกเป็น 2 ประเภทคือ

1. การบีบอัดแบบสูญเสียรายละเอียดข้อมูล (Lossy compression) วิธีการนี้ จะมีการเปลี่ยนแปลงค่าความสว่างของจุดภาพ จะส่วนมากจะทำให้สามารถลดขนาดของข้อมูลภาพได้มาก แต่วิธีการนี้ไม่เหมาะสมสำหรับข้อมูลภาพที่ต้องการมีการจำแนกข้อมูล (Classification) ตัวอย่างของการบีบอัดประเภทนี้ได้แก่ JPEG, MrSID หรือ ECW เป็นต้น

2. การบีบอัดแบบไม่มีการสูญเสียรายละเอียดข้อมูล (Lossless Compression) วิธีการนี้จะทำให้ค่าของแต่ละจุดภาพจะยังคงอยู่เหมือนเดิมหรือไม่มีการเปลี่ยนแปลงค่าของแต่ละจุดภาพ ซึ่งจะทำให้ข้อมูลภาพที่ถูกถอดรหัสแล้วจะยังคงเป็นเหมือนภาพต้นฉบับ 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

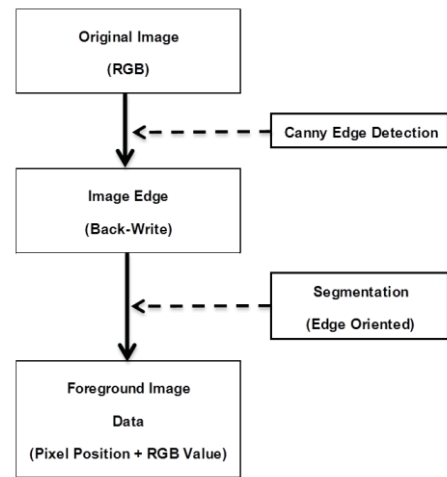
3. การออกแบบซอฟต์แวร์

3.1 การแบ่งส่วนภาพ (Image Segmentation)

การออกแบบจะเริ่มจากการประมวลผลภาพต้นแบบ (Input Image) ที่มาจากเฟรมภาพของเว็บแคม และทำการแยกส่วนภาพคนและภาพพื้นหลังออกจากกัน จากนั้นเขียนโปรแกรมเลือกเฉพาะตำแหน่งและค่าของจุดภาพที่ต้องการ เพื่อลดจำนวนข้อมูลในรูปภาพส่วนที่ไม่ต้องการลง

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

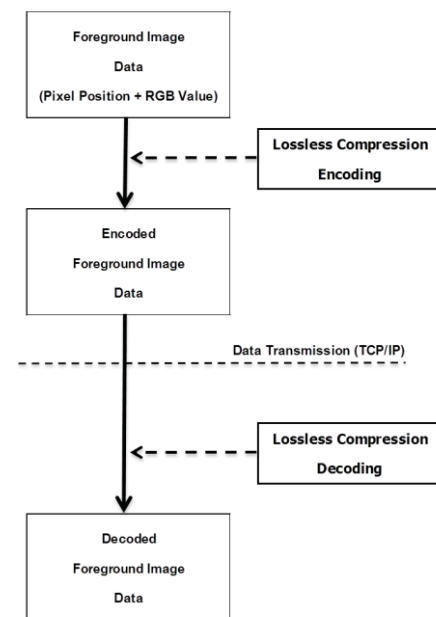
การออกแบบโปรแกรมจะเริ่มต้นจากการรับภาพสี (RGB) มาจากกล้องเว็บแคมด้วยขนาด 120x160x3 จุดภาพ จากนั้นหาขอบภาพโดยใช้วิธี Canny เพื่อที่จะสามารถหาตำแหน่งและเก็บข้อมูลของภาพส่วนที่ต้องการได้



รูปที่ 3 การออกแบบโปรแกรมการแบ่งส่วนภาพ

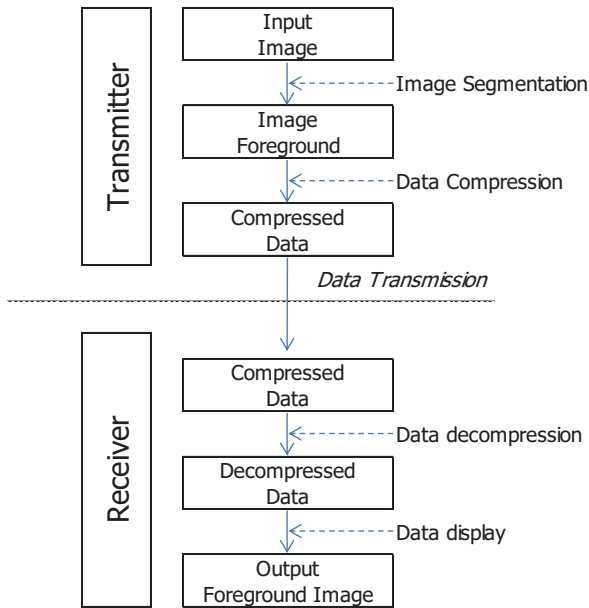
3.2 การบีบอัดข้อมูลภาพ (Data Compression)

ข้อมูลรูปภาพส่วนที่สนใจ (Foreground) จะถูกนำมาทำการบีบอัดข้อมูลแบบไม่มีการสูญเสียรายละเอียดข้อมูล (Lossless Compression) เพื่อทำการลดจำนวนของข้อมูล โดยที่จะไม่ส่งผลต่อคุณภาพของข้อมูล ซึ่งการออกแบบการบีบอัดข้อมูลภาพจะเป็นไปตามขั้นตอนดังรูปที่ 4 ในขณะเดียวกัน ส่วนของภาครับหลังจากการส่งผ่านข้อมูลทางอินเทอร์เน็ตแล้ว ก็จะต้องเขียนการถอดรหัสข้อมูลเพื่อเตรียมพร้อมที่จะแสดงผลข้อมูลภาพต่อไป



รูปที่ 4 การออกแบบการบีบอัดข้อมูลภาพ

ภาพรวมของการเขียนโปรแกรมการลดขนาดข้อมูลภาพโดยการใช้นิเทศของการแบ่งส่วนภาพรวมกับการบีบอัดข้อมูลภาพจะสามารถแสดงได้ดังรูปที่ 5

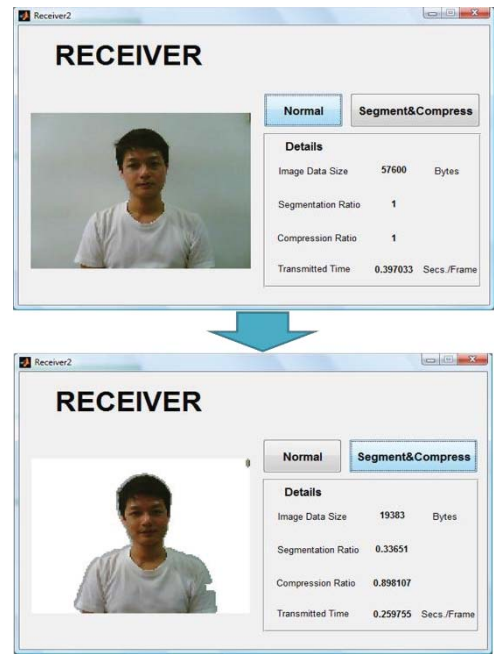


รูปที่ 5 ภาพรวมของโปรแกรมการลดขนาดข้อมูลภาพ

ภาพรวมการออกแบบโปรแกรมจะแบ่งการทำงานเป็นสองส่วนได้แก่ภาคส่ง (Transmitter) และภาครับ (Receiver) โดยในส่วนของภาคส่ง จะทำการเลือกค่าตำแหน่งจุดภาพและค่าความเข้มของแต่ละชั้นสี (RGB) ก่อนนำมาบีบอัดและส่งเป็นข้อมูลไปยังภาครับ ซึ่งภาครับก็จะทำการถอดรหัสข้อมูลเพื่อจะแสดงผลข้อมูลภาพเฉพาะส่วนภาพที่เราต้องการออกมาเป็นภาพสีในลักษณะเดียวกันกับภาพต้นแบบ แต่จะแสดงเฉพาะส่วนที่ต้องการ

4. ผลการทดลอง

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



รูปที่ 6 ตัวอย่างผลของการส่งข้อมูลภาพด้วยฉากหลังสีอ่อน

จากรูปที่ 6 (บน) แสดงหน้าต่าง GUI ของผลที่ได้จากการทดลองส่งข้อมูลภาพปกติด้วยฉากหลังสีอ่อน ขนาดข้อมูลของภาพมีค่า 57,600 ไบต์ ใช้เวลาในการส่งข้อมูลภาพ 0.3970 วินาทีต่อเฟรม

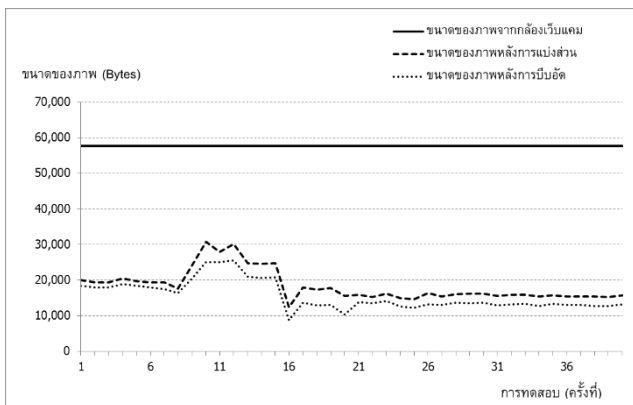
รูปที่ 6 (ล่าง) แสดงหน้าต่าง GUI ของผลที่ได้จากการทดลองส่งข้อมูลภาพที่ได้จากการแบ่งส่วนภาพและบีบอัดข้อมูลภาพ ขนาดข้อมูลของภาพมีค่า 19,383 ไบต์ สัดส่วนของข้อมูลภาพที่ถูกแบ่งส่วนต่อภาพดั้งเดิมมีค่า 0.3365 และสัดส่วนของข้อมูลที่ถูกบีบอัดแล้วต่อข้อมูลก่อนการบีบอัดมีค่า 0.8981 ใช้เวลาในการส่งข้อมูลภาพ 0.2598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

จากข้อมูลที่ได้จากการทดสอบ จะเห็นได้ว่าขนาดข้อมูลภาพที่ได้จากกล้องเว็บแคมมีค่าคงที่ที่ 57,600 ไบต์ และค่าข้อมูลภาพที่ได้จากการแบ่งส่วนจะมีค่าเฉลี่ยของอัตราการแบ่งส่วนข้อมูลภาพจากการทดสอบ

ทั้งหมด 40 ครั้งอยู่ที่ 0.3192 หรือสามารถลดขนาดข้อมูลภาพลดลงเหลือร้อยละ 31.92

การบีบอัดข้อมูลภาพหลังจากการแบ่งส่วนภาพ ทำให้ข้อมูลภาพมีค่าเฉลี่ยอยู่ที่ 0.2709 หรือสามารถลดขนาดข้อมูลภาพลดลงเหลือร้อยละ 27.09 เมื่อเทียบกับข้อมูลของภาพตั้งต้น

5. สรุป

งานวิจัยนี้ได้พัฒนาซอฟต์แวร์เพื่อเพิ่มประสิทธิภาพในการส่งผ่านข้อมูลทางอินเทอร์เน็ต ด้วยการลดขนาดข้อมูลภาพ โดยเทคนิคการแบ่งส่วนภาพ ที่อาศัยวิธีการหาขอบภาพของ Canny และเลือกส่วนข้อมูลที่น่าสนใจและบีบอัดข้อมูลให้มีขนาดเล็กลง ผลที่ได้คือสามารถส่งข้อมูลผ่านอินเทอร์เน็ตได้เร็วและมีความต่อเนื่องมากขึ้น

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

ซอฟต์แวร์ที่ได้พัฒนาขึ้นนี้ สามารถนำไปประยุกต์ใช้กับการสื่อสารที่มีการรับส่งข้อมูลภาพที่ต้องการลดภาระการเก็บและประมวลผลข้อมูลภาพที่มีขนาดใหญ่ได้เช่น การส่งข้อมูลภาพของคนไข้เพื่อการรักษาที่เร็วขึ้น การส่งภาพวีดิโอสาริตการทำงาน 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.