

## การประมวลผลวิดีโอแบบขนานโดยใช้เว็บซีแอล Parallel Video Processing Using WebCL

อัษฎาวุธ อัครมหาพงษ์<sup>1\*</sup>, สิทธิพนธ์ โทน้อย<sup>2</sup>, สุรศักดิ์ สุทธิ<sup>3</sup>, อภิลิธิ รัตนตรานูรักษ์<sup>4</sup>

<sup>1\*,2,3,4</sup> สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏสวนสุนันทา  
กรุงเทพฯ

Email : apisit.ra@ssru.ac.th<sup>4</sup>

### บทคัดย่อ

ในบทความนี้กล่าวถึงการประมวลผลภาพวิดีโอแบบขนานโดยใช้เว็บซีแอล ซึ่งเว็บซีแอลเป็นเฟรมเวิร์คในการพัฒนาโปรแกรมให้สามารถส่งโปรแกรมให้ทำงานบนระบบที่แตกต่างกันบนเว็บ บทความนี้เริ่มต้นจากการวิเคราะห์อัลกอริธึมต้นแบบ เช่น brightness contrast negative posterize threshold grayscale sepai flip sharpen blur laplacian และ bluescreen จากนั้นจึงมีการนำเว็บซีแอลไปประยุกต์กับ อัลกอริธึมเหล่านี้ เพื่อให้สามารถทำงานบนหน่วยประมวลผลกลางหลายแกน หน่วยประมวลผลกราฟิกรวม และหน่วยประมวลผลกราฟิกของ NVIDIA ผลลัพธ์ที่ได้จะเห็นว่า Speedup ของ หน่วยประมวลผลกราฟิก NVIDIA สูงถึง 1.31 เท่าเมื่อเปรียบเทียบกับอัลกอริธึมต้นฉบับที่ทำงานแบบตามลำดับ

**คำสำคัญ :** การโฆษณา, จาวาสคริปต์, อินเทอร์เน็ต เว็บฟอร์ม, ทวิตเตอร์

### Abstract

This paper presents parallel video processing algorithm using WebCL which is the framework for developing program that can execute on heterogeneous systems on web. This paper starts with analysis the original algorithms such as brightness, contrast, negative, posterize, threshold, grayscale sepia, flip, sharpen, blur, laplacian, and bluescreen. We have implemented WebCL to these algorithms to execute on Multi-core CPU, Integrated GPU and NVIDIA GPU. The results show that Speedup of NVIDIA GPU can be achieved to 1.31X compared with the original sequential version.

**Keywords:** Advertisement, JavaScript, Internet, Web forms, Twitter

## บทนำ

การถ่ายภาพ และวิดีโอในปัจจุบันนั้นมีประสิทธิภาพมากยิ่งขึ้น ส่งผลให้ความละเอียดของภาพ และวิดีโอสูงมากยิ่งขึ้น ซึ่งปัจจุบันการถ่ายวิดีโอนั้นสามารถถ่ายได้หลากหลายอุปกรณ์มากยิ่งขึ้น ไม่ว่าจะเป็นกล้องถ่ายวิดีโอ กล้องถ่ายภาพทีเอสแอลอาร์ แม้กระทั่งกล้องโทรศัพท์มือถือก็สามารถนำมาใช้ในการถ่ายวิดีโอได้อีกด้วย ซึ่งปัจจุบันกล้องโทรศัพท์มือถือนั้นสามารถถ่ายวิดีโอได้ระดับ 4K ทำให้ความละเอียดของวิดีโอมีค่าสูงมาก หากจะนำวิดีโอเหล่านี้ไปประมวลผลย่อมใช้เวลาในการประมวลผลมากขึ้นตามความละเอียดที่เพิ่มมากขึ้นด้วย

ปัจจุบันเทคโนโลยีเว็บเข้ามามีบทบาทในการใช้ชีวิตของมนุษย์มากยิ่งขึ้น ทำให้การพัฒนาเว็บแอปพลิเคชัน (Web Application) ซึ่งสามารถทำงานได้บนระบบปฏิบัติการที่หลากหลาย นั้นเกิดความสะดวก และได้รับความนิยมมากขึ้น เนื่องจากแอปพลิเคชันลักษณะนี้สามารถพัฒนาเพียงครั้งเดียว และเปิดใช้ผ่านเว็บเบราว์เซอร์ (Web browser) ทำให้เกิดความสะดวกในการพัฒนา และสามารถเข้าถึงได้บนทุกอุปกรณ์

ในการพัฒนาเว็บแอปพลิเคชัน โดยทั่วไปนั้นจะใช้ภาษาจาวาสคริปต์ (JavaScript) ในการพัฒนาซึ่งภาษานี้หากพัฒนาตามปกติจะสามารถทำงานได้บนซีพียูเพียง 1 แกน นั่นคือสามารถทำงานได้ในรูปแบบตามลำดับ ซึ่งปัจจุบันนั้นหน่วย

ประมวลผลกลาง (Central Processing Unit: CPU) มีหลายแกนประมวลผล ซึ่งสามารถทำงานได้พร้อมกันหลายงาน ซึ่งการพัฒนาโปรแกรมโดยปกตินั้นจะไม่สามารถดึงทรัพยากรที่มีอยู่มาใช้ได้ นอกจากนี้ ปัจจุบันหน่วยประมวลผลกราฟิก (Graphics Processing Unit: GPU) สามารถนำมาใช้ในการประมวลผลแบบปกติได้ ซึ่งหน่วยประมวลผลนี้มีจำนวนแกนประมวลผลจำนวนมากกว่าซีพียูหลายเท่า ทำให้การดึงทรัพยากรเหล่านี้มาใช้ในการประมวลผลจะสามารถช่วยลดเวลาในการประมวลผลได้

ในการประมวลผลด้วย Multi-core CPU และ GPU ในงานทั่วไปบนเว็บนั้น สามารถใช้ไลบรารีเว็บซีแอล (Web Computing Language: WebCL) (Aarnio, T. and M. Bourges-Sevenier, 2012) ในการดึงหน่วยประมวลผลต่าง ๆ มาใช้ในการประมวลผลแบบขนานได้ โดยเว็บซีแอลนั้นจะมีลักษณะการทำงานเช่นเดียวกับโอเพนซีแอล (Open Computing Language: OpenCL) (Khronos, 2010)(NVIDIA, 2009)(AMD, 2011) ซึ่งมีความแตกต่างกันที่เว็บซีแอลใช้ในการประมวลผลบนเว็บ

จากปัญหาที่กล่าวมาข้างต้น ผู้วิจัยเห็นว่าหากมีการนำการประมวลผลวิดีโอบนเว็บมาพัฒนาให้มีการประมวลผลแบบขนานด้วยเว็บซีแอล โดยการนำทรัพยากรอื่น ไม่ว่าจะเป็น Multi-core CPU และ

GPU มาประมวลผลจะทำให้เวลาที่ใช้ในการประมวลผลลดลง

ในการพัฒนาเว็บแอปพลิเคชันนั้น โดยปกติแล้วจะทำงานอยู่เพียง 1 อุปกรณ์ นั่นคือบนซีพียู เพียงอย่างเดียว ซึ่งมีความวิจัยที่มีการนำเว็บซีแอล และโอเพนซีแอล ขึ้นมาพัฒนา ซึ่งงานวิจัยของ Xianglan Piao, et al. (Xianglan Piao et al, 2014) ได้มีการทดสอบการทำงาน โดยใช้ เว็บ ซี แอล บน N-body Simulation, Edge Filter, Matrix Multiplication และ Mandelbrot ในขนาดที่แตกต่างกัน บน GPU และบน Multi-core CPU โดยหากมีการประมวลผลเฉพาะบน GPU นั้นจะมีความเร็วเพิ่มขึ้น 33% และถ้าหากมีการพัฒนาบนทั้ง 2 อุปกรณ์จะสามารถเพิ่มประสิทธิภาพได้ถึง 65%

งานวิจัยของ Tommy Mac William, et al. (Tommy MacWilliam, and Cris Cecka., 2013) ได้มีการสร้างเฟรมเวิร์คที่ใช้ในการพัฒนาเว็บแอปพลิเคชันโดยใช้อุปกรณ์อื่น เช่น GPU แทนการใช้ CPU ปกติ ซึ่งเมื่อเปรียบเทียบการทำงานแล้วสามารถทำงานใช้เวลาน้อยกว่าการทำงานด้วย CPU ปกติ ซึ่งทดสอบบน Thomson problem โดยการตอบสนองต่อวินาทีสูงกว่าการใช้จาวาสคริปต์ปกติ สูงถึง 4.5 เท่า

งานวิจัยของ Stephen Herhut et al. (Stephen Herhut et al., 2012) มีการทำคอมไพเลอร์ JIT เพื่อเพิ่ม

ประสิทธิภาพของเว็บแอปพลิเคชัน ซึ่งสามารถแปลงจาวาสคริปต์ให้เป็นโอเพนซีแอลได้ โดยมีการทดสอบ Frame rate ของการคำนวณ 3D Particle flocking simulation ได้ถึง 50 Frames/second เมื่อทำงานบน 8 เธรด และการคำนวณ Dense Matrix Multiplication ซึ่ง Speedup ขึ้นไปถึง 10.99 เมื่อคำนวณ 1k x 1k เมทริกซ์

งานวิจัยของ Won Jeon et al. (Won Jeon et al., 2012) ได้มีการนำภาษาเอชทีเอ็มแอล 5 มาทำงานร่วมกับภาษาจาวาสคริปต์ และนำเอาเว็บแอปพลิเคชันมาพัฒนาร่วมกับ GPU และทดสอบบน Sobel filter, N-Body Simulation และ Deformable body simulation ซึ่ง 3 อัลกอริธึมใช้พลังงานในการประมวลผลสูงมาก ซึ่งผลลัพธ์ที่ได้ Sobel Filter มี Speedup เป็น 13 เท่าที่ภาพขนาด 256x256 พิกเซล N-Body Simulation มี Speedup 23 เท่าที่ 1024 พิกเซล และ Deformable body simulation มี Speedup สูงถึง 116 เท่า

นอกจากนี้ยังมีงานวิจัยของ Jin Wang et al. (Jin Wang et al., 2014) ได้มีการพัฒนา ParalleUS ซึ่งเป็นการนำเสนอเฟรมเวิร์คที่ทำงานบน CPU, GPU โดยมีการพัฒนาภาษาระดับสูง โดย Speedup สูงถึง 26.8 เท่า บน simple reduction algorithm เมื่อเทียบกับจาวาสคริปต์ปกติ

ในงานวิจัยการประมวลผลแบบกระจายบนเบราว์เซอร์ของ Reginald Cushing (Reginald Cushing et al., 2013) ได้มีการนำภาษาจาวาสคริปต์มาพัฒนา เพื่อประมวลผลของเว็บเบราว์เซอร์ โดยใช้ WeevilScout มาทดสอบบนซอฟต์แวร์ชีวสารสนเทศ ซึ่งสามารถทำ Speedup ได้สูงมากขึ้นกว่าการประมวลผลแบบจาวาสคริปต์ตามปกติ

จากงานวิจัยที่ใช้หน่วยประมวลผลกราฟิกไปใช้ในการพัฒนาข้างต้น จะเห็นว่าการส่วนใหญ่จะมี Speedup ที่สูงมากขึ้น อย่างไรก็ตามงานวิจัยส่วนใหญ่จะเน้นไปทางด้านผลการประมวลผลพื้นฐาน หรือนำไปใช้ในงานเฉพาะทาง ดังนั้นจึงได้มีการนำมาแสดงไว้ในตารางที่ 1

ในบทความนี้กล่าวถึงเนื้อหาตามลำดับดังนี้ วัตถุประสงค์ของการวิจัย ซึ่งหัวข้อนี้จะแสดงถึงวัตถุประสงค์การทำวิจัย จากนั้นจะนำเสนอเกี่ยวกับวิธีการดำเนินงานวิจัย เพื่ออธิบายขั้นตอนการทำวิจัย และวิธีการทดลอง จากนั้นจะมีการ

นำเสนอผลการทดลองที่ได้ และจะมีการสรุป อภิปรายผลเป็นส่วนสุดท้าย และแนวทางในการพัฒนาต่อ

### วัตถุประสงค์ของการวิจัย

ในงานวิจัยชิ้นนี้มีวัตถุประสงค์เพื่อพัฒนาการประมวลผลภาพวิดีโอบนเว็บที่ทำงานแบบตามลำดับ ให้มีการทำงานแบบขนานโดยใช้เว็บซีแอล ซึ่งเว็บซีแอลนั้นจะสามารถดึงทรัพยากรการประมวลผลอื่น ไม่ว่าจะเป็น Multi-core CPU, GPU หรือหน่วยประมวลผลอื่น มาช่วยในการประมวลผลวิดีโอได้ เพื่อให้เวลาที่ใช้ในการประมวลผลลดลง ในฟังก์ชันการประมวลผลต่าง ๆ กัน ได้แก่ Brightness, Contrast, Negative, Posterize Threshold, Grayscale, Sepia, Vertical, Horizontal Flip, Sharpen, Blur, Laplacian, Bluescreen โดยจะทดสอบที่ความละเอียดวิดีโอที่แตกต่างกัน เพื่อหา Speedup เปรียบเทียบกับการประมวลผลแบบตามลำดับ

ตารางที่ 1 งานวิจัยที่เกี่ยวข้องที่นำหน่วยประมวลผลกราฟิกไปประมวลผลบนเว็บ

งานวิจัย	วิธีการทดสอบ	Maximum Speedup
Xianglan Piao et al, 2014	NBody, EdgeFilter, Matrix Multiplication, Mandelbrot (GPU) Thomson Problem	33x 4.5x
Tommy MacWilliam, and Cris Cecka., 2013	3D Particle flocking simulation 1k x 1k	10.99x
Stephen Herhut et al., 2012		13x
Won Jeon et al., 2012	Sobel Filter (256x256 images)	26.8x
Jin Wang et al., 2014	simple reduction	



ภาพที่ 1 รูปแบบการพัฒนาอัลกอริธึม  
ที่มา : อภิสิทธิ์ รัตนตรานูรักษ์. (2559) วาดภาพ

### วิธีดำเนินการวิจัย

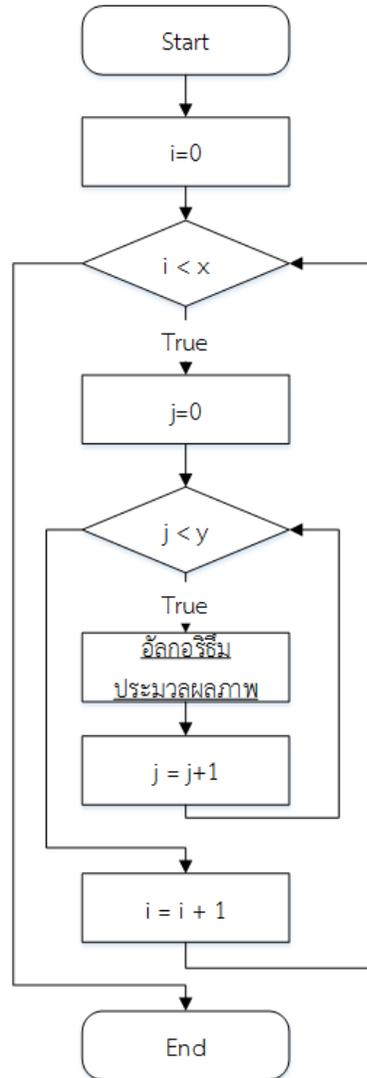
ในงานวิจัยนี้จะมีการพัฒนาเว็บที่สามารถนำเอาวิดีโอไปประมวลผลโดยใช้ทรัพยากรบนฝั่งผู้ใช้ โดยจะนำวิดีโอต้นแบบไปถอดรหัสให้เป็นชุดของภาพ จากนั้นจะทำการประมวลผลในแต่ละอัลกอริธึม ในทุก ๆ เฟรมภาพที่มีการถอดรหัสออกมา เมื่อมีการประมวลผลวิดีโอเรียบร้อยแล้ว จะมีการเข้ารหัสกลับ

ให้เป็นวิดีโอเช่นเดิม ซึ่งวิธีการดำเนินการวิจัยนั้นมีกระบวนการดังภาพที่ 1

ในการประมวลผลภาพโดยทั่วไปนั้นจะมีการประมวลผลแบบตามลำดับ ซึ่งในการประมวลผลภาพจะต้องมีการประมวลผลทั้งแกน x และแกน y ของภาพ เมื่อความกว้างของภาพเป็น x พิกเซล และความยาวของภาพเป็น y พิกเซล ซึ่งจะทำให้เกิดการวนซ้ำในแกน x และในแกน y

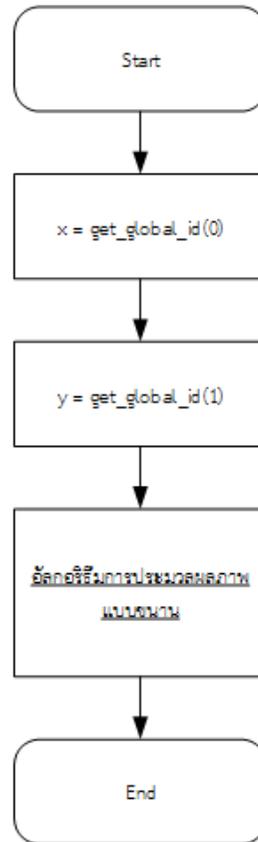
ซึ่งภายในรูปนั้นจะมีอัลกอริธึมการประมวลผลภาพที่ทำงานแบบตามลำดับอยู่ และจะมีการประมวลผลเช่นนี้ไปจนกระทั่งวนซ้ำค่า  $x$  และค่า  $y$  จนครบ ดังภาพที่ 2

เมื่อมีการวนซ้ำค่า  $x$  และ  $y$  ดังที่กล่าวมาข้างต้นจะส่งผลให้ความซับซ้อนของอัลกอริธึมสูงมากขึ้น ซึ่งความซับซ้อนจะอยู่ที่  $x.y$  เมื่อประมวลผลภาพ 1 ภาพนั้นแสดงให้เห็นว่าหากภาพมีความกว้างและความยาวมากขึ้นจะส่งผลให้ใช้เวลาในการประมวลผลมากขึ้น และหากจำนวนเฟรมของภาพมีค่า  $z$  จะส่งผลให้ความซับซ้อนเป็น  $x.y.z$  ซึ่งสิ่งเหล่านี้จะส่งผลต่อเวลาในการประมวลผลของอัลกอริธึมดังกล่าว



ภาพที่ 2 ผังงานของอัลกอริธึมที่ทำงานแบบตามลำดับ  
ที่มา : อภิสิตธิ์ รัตนาตราณูรักษ์. (2559)  
วาดภาพ

เมื่อมีการนำภาพไปถอดรหัสจะเกิดภาพหลายภาพ ทำให้ปริมาณงานในการประมวลผลสูงขึ้น และเหมาะสมในการประมวลผลแบบขนาน โดยอัลกอริธึมที่จะนำมาพัฒนาให้มีการทำงานแบบขนาน โดยในงานวิจัยนี้ จะมีการนำอัลกอริธึม Brightness, Contrast, Negative, Posterize Threshold, Grayscale, Sepia, Vertical, Horizontal Flip, Sharpen, Blur, Laplacian, Bluescreen มาทดสอบ โดยจะมีการปรับอัลกอริธึมที่ทำงานแบบตามลำดับ มาปรับเป็นการทำงานแบบขนาน โดยใช้เว็บไซต์เสียก่อน โดยอัลกอริธึมโดยภาพรวมเมื่อมีการพัฒนาให้มีการทำงานแบบขนานเป็นดังภาพที่ 3



ภาพที่ 3 ผังงานของอัลกอริธึมที่ทำงานแบบขนาน  
ที่มา : อภิสิตธี รัตนาตราณูรักษ์. (2559)  
วาดภาพ

จากภาพที่ 3 จะเห็นว่า เมื่อมีการปรับอัลกอริธึมให้มีการทำงานแบบขนานโดยใช้เว็บไซต์แล้ว จะมีการประมวลผลภาพไปพร้อมกันในพิกเซลแกน x และแกน y โดยจะมีการกำหนดค่าแกน x และ y ในแต่ละพิกเซล ซึ่งจะมีการแจกงานไปในแต่ละหน่วยประมวลผล และทุกประมวลผลจะมีการประมวลผลพร้อมกันทุกพิกเซล ซึ่ง

การประมวลผลในลักษณะนี้จะช่วยลดเวลาในการประมวลผล อย่างไรก็ตามในการประมวลผลวีดีโอนั้นจะต้องมีการประมวลผลภาพเฟรมอื่น หากจำนวนเฟรมมีค่าเป็น  $z$  ความซับซ้อนจะมีค่าเป็น  $z$  ในกรณีของแกนประมวลผลมากเกินพอ และหากแกนประมวลผลมีค่าเป็น  $c$  จะทำให้ความซับซ้อนมีค่าเป็น  $z/c$

### ผลการวิจัย

ในงานวิจัยนี้ได้มีการพัฒนาและทดลองการประมวลผลวีดีโอด้วยเว็บซีแอลบนสภาพแวดล้อมและตัวแปรต่าง ๆ โดยในแต่ละตัวแปรจะมีทั้งตัวแปรที่มีการทำงานแบบตามลำดับ และการทำงานแบบขนานโดยใช้เว็บซีแอล ซึ่งสภาพแวดล้อมในการทดลองได้มีการทดลองบนหน่วยประมวลผลกลาง Intel Core i7-4710HQ 2.5 GHz ซึ่งประกอบไปด้วยแกนประมวลผลจำนวน 4 แกน รองรับการทำงานแบบไฮเปอร์เธรด (Hyper Threading Technology) และมีแคช L3

จำนวน 6 MB ซึ่งมีหน่วยประมวลผลกราฟิกร่วมเป็น Intel HD Graphics 4600 โดยจะมีการแบ่ง VRAM 2 GB และมีหน่วยประมวลผลกราฟิกแยก NVIDIA GeForce GTX 860M หน่วยความจำหลักของระบบที่ใช้ในการทดลอง 8GB ทำงานบนระบบปฏิบัติการ Windows 7 Ultimate 64-bit เว็บเบราว์เซอร์ที่ใช้ Mozilla Firefox 34 และติดตั้งส่วนเสริม Nokia WebCL 1.0 ดังตารางที่ 2

งานวิจัยนี้ได้มีการใช้วีดีโอเข้ามาทดสอบโดยใช้วีดีโอ 5 วินาที ขนาด 300x150, 480x240 และ 640x320 มาทดลองด้วยอัลกอริธึม Brightness, Contrast, Negative, Posterize Threshold, Grayscale, Sepia, Vertical, Horizontal Flip, Sharpen, Blur, Laplacian, Bluescreen โดยทดลองบนซีพียู 1 แกน ซีพียูหลายแกน หน่วยประมวลผลกราฟิกร่วม และหน่วยประมวลผลกราฟิกแยก ดังตารางที่ 3

## ตารางที่ 2 สภาพแวดล้อมในการทดลอง

ชนิดของสภาพแวดล้อม	สภาพแวดล้อมที่ใช้ในการทดลอง
หน่วยประมวลผลกลาง	Intel Core i7-4710HQ 2.5 GHz
หน่วยความจำหลัก	8 GB
หน่วยประมวลผลกราฟิก	NVIDIA Geforce GTX 860M Intel HD Graphics 4600
ระบบปฏิบัติการ	Windows 7 Ultimate 64-bit
WebCL	Nokia WebCL 1.0
เบราว์เซอร์	Mozilla Firefox 34

## ตารางที่ 3 ตัวแปรที่ใช้ในการทดลอง

ตัวแปร	ค่า
วิดีโอ	5 วินาที ขนาด 300x150, 480x240, 640x320 พิกเซล bitrate 1948 kbps 23 fps
อัลกอริธึม	Brightness, Contrast, Negative, Posterize Threshold, Grayscale, Sepia, Vertical, Horizontal Flip, Sharpen, Blur, Laplacian, Bluescreen
อุปกรณ์ที่ใช้ในการประมวลผล	CPU (1 core), Multi-core CPU, Integrated GPU, Dedicated GPU

จากภาพที่ 4 แสดงผลการทดลองด้านเวลาของการประมวลผลวิดีโอด้วยจาวาสคริปต์ ซึ่งจะทำงานบนซีพียูเพียง 1 แกนเท่านั้น โดยจะเห็นว่า เมื่อขนาดของเฟรมในวิดีโอมีความละเอียดมากยิ่งขึ้น จะส่งผลให้เวลาที่ใช้ในการประมวลผลสูงขึ้น โดยฟังก์ชัน Sharpen Blur และ Laplacian ใช้เวลาในการประมวลผลของ

ซีพียู 1 แกนสูงกว่าฟังก์ชันอื่น เนื่องจากความซับซ้อนของอัลกอริธึมสูงกว่า

จากภาพที่ 5 แสดงผลการทดลองด้านเวลาของการประมวลผลวิดีโอด้วยซีพียูหลายแกนด้วยเว็บซีแอล โดยในที่นี้ประกอบไปด้วยซึ่งจะเห็นว่าเวลาที่ใช้ในการประมวลผลจะลดลงบางฟังก์ชัน เช่น Sharpen Blur Laplacian โดยจะเห็นชัดที่ความละเอียด 640x320 pixel

จากภาพที่ 6 แสดงผลการทดลองด้านเวลาของการประมวลผลวิดีโอด้วย Integrated GPU ซึ่งเวลาในการประมวลผลจะลดลงในบางฟังก์ชัน เช่นกัน โดยฟังก์ชันที่ใช้เวลาในการประมวลผลลดลงได้แก่ Sharpen Blur Laplacian

จากภาพที่ 7 แสดงผลการทดลองด้านเวลาของการประมวลผลวิดีโอด้วย NVIDIA GPU โดยเวลาที่ใช้ในการประมวลผลนั้นลดลงมากที่สุดที่ฟังก์ชัน Laplacian และมีฟังก์ชันอื่น ๆ ที่ใช้เวลาในการประมวลผลลดลงเช่นกันได้แก่ Sharpen และ Blur

เมื่อนำผลการทดลองด้านเวลาไปหา Speedup ซึ่งเป็นสัดส่วนระหว่าง เวลาที่ใช้ในการทำงานแบบตามลำดับ ต่อ เวลาที่ใช้ในการทำงานแบบขนาน จะได้ผลลัพธ์ดังภาพที่ 8 – 10 โดยจะแสดง Speedup การทำงานด้วยเว็บซีแอล ของ Multi-core CPU, Integrated GPU และ NVIDIA GPU ตามลำดับ ซึ่งจะเห็นว่าเมื่อมีการพัฒนาโปรแกรมแบบขนานด้วยเว็บซีแอล ไม่ว่าจะเป็น Multi-core CPU, Integrated GPU หรือ NVIDIA GPU ฟังก์ชัน Sharpen, Blur, Laplacian จะมี Speedup ที่มีค่าสูงกว่า 1 นั่นคือใช้เวลาในการทำงานน้อยลง เมื่อขนาดของวิดีโอมีค่ามากขึ้น

อย่างไรก็ตามหากความละเอียดของวิดีโอมีค่าน้อย จะส่งผลให้ค่า Speedup ต่ำกว่า 1 นั่นคือใช้เวลาในการประมวลผลมากกว่าการประมวลผลแบบ

ตามลำดับ ดังนั้นจะเห็นได้ว่าสิ่งที่ผลต่อ Speedup นั้นนอกจากจะอยู่ที่อุปกรณ์ที่ใช้ในการประมวลผล ความละเอียดของไฟล์วิดีโอแล้ว ยังขึ้นอยู่กับอัลกอริธึมอีกด้วย ซึ่งในที่นี้จะวิเคราะห์ในส่วนของการสรุปและอภิปรายผล

### สรุปและอภิปรายผล

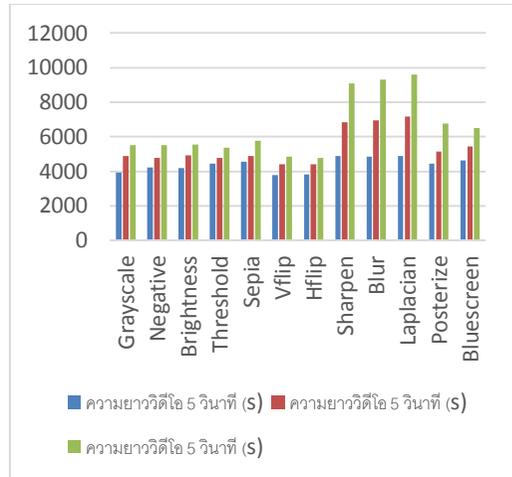
ในงานวิจัยชิ้นนี้จะเห็นว่ามีการพัฒนาอัลกอริธึมการประมวลผลวิดีโอที่ทำงานแบบตามลำดับโดยใช้จาวาสคริปต์ให้มีการทำงานแบบขนานโดยใช้เว็บซีแอลเข้ามาเรียกใช้อุปกรณ์ต่าง ๆ เพื่อช่วยในการประมวลผล ไม่ว่าจะเป็น Multi-core CPU, Integrated GPU, NVIDIA GPU ซึ่ง Speedup สูงสุดเมื่อเปรียบเทียบกับการทำงานแบบตามลำดับสูงถึง 1.31 เท่า บน NVIDIA GPU โดยใช้วิดีโอขนาดเพียง 640x320 pixel

จากผลการทดลองที่ได้จะพบว่า อุปกรณ์ที่เหมาะสมกับการประมวลผลวิดีโอ นั่นคือ NVIDIA GPU, Intel GPU และ Multi-core CPU ตามลำดับ เนื่องจากจำนวนหน่วยประมวลผลของ NVIDIA GPU นั้นจะมีมากกว่า Multi-core CPU ถึงแม้ว่าความถี่ของ Multi-core CPU จะมีค่ามากกว่าก็ตาม

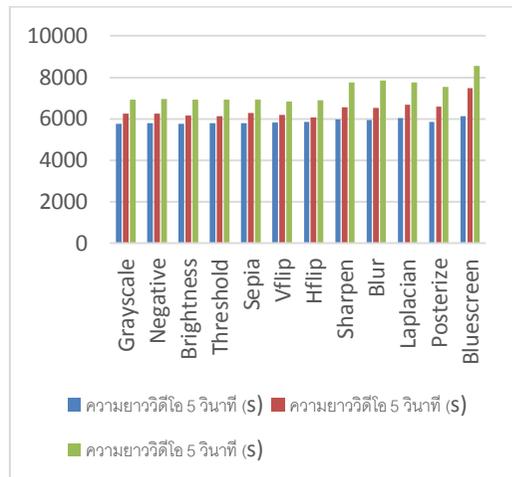
นอกจากนี้จากผลการทดลองจะเห็นว่า เมื่อเพิ่มความละเอียดของวิดีโอ ให้มีความละเอียดมากขึ้น จะส่งผลให้ Speedup สูงมากยิ่งขึ้น ในทุกอุปกรณ์ ดังนั้นหากผู้ใช้มีการประมวลผลวิดีโอที่

ความละเอียดสูงมากกว่าการทดลองนี้ จะทำให้ Speedup สูงขึ้นตามไปด้วย อย่างไรก็ตามจากผลการทดลองจะเห็นว่า ผลการทดลองส่วนใหญ่จะมีค่า Speedup ต่ำกว่า 1.0 เนื่องจากความละเอียดของวิดีโอยังมีค่าไม่สูงมากเพียงพอ ทำให้ Overhead ในการส่งข้อมูลวิดีโอไปยังอุปกรณ์ รวมถึงการแจกงาน ไปในแต่ละแกนประมวลผล สูงมากกว่าเวลาที่ใช้ในการประมวลผล อีกทั้งความซับซ้อนของอัลกอริธึมของแต่ละฟังก์ชันมีค่าไม่เท่ากัน เช่น Blur Laplacian และ Sharpen มีความซับซ้อนของอัลกอริธึมมากกว่า ฟังก์ชันอื่น จึงทำให้ปริมาณงานที่มีการประมวลผลจริงมีน้อยกว่า Overhead และส่งผลทำให้ Speedup ของฟังก์ชันอื่น ที่ไม่ใช่ 3 ฟังก์ชันนี้ มีค่าน้อยกว่า 1.0

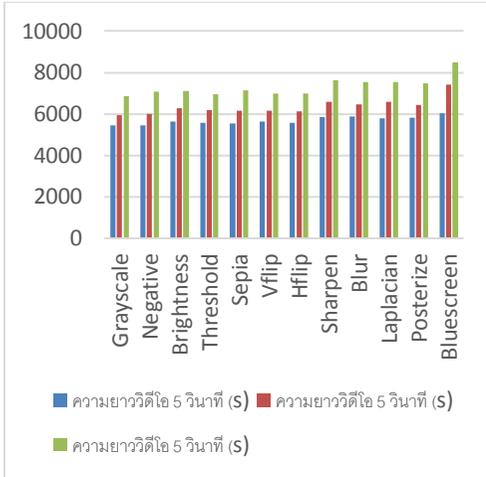
งานในอนาคตของงานวิจัยชิ้นนี้สามารถต่อยอดได้โดยการพัฒนาในส่วนของการเข้ารหัส และถอดรหัสไฟล์วิดีโอ ให้มีการทำงานแบบขนาน โดยใช้เว็บซีแอลได้ เพื่อให้ประสิทธิภาพในการทำงานโดยรวมสูงขึ้น



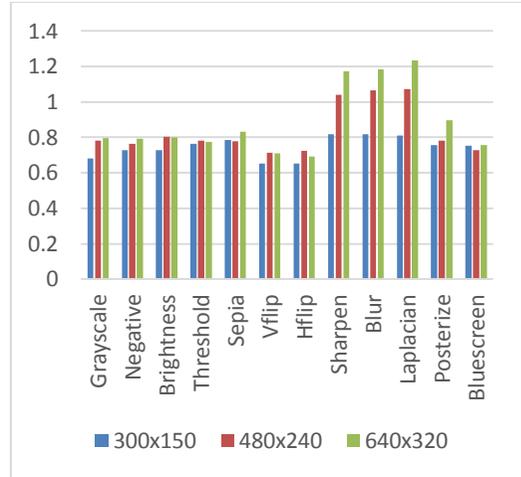
ภาพที่ 4 เวลาการประมวลผลวิดีโอโดยใช้ จาวาสคริปต์ (1 CPU Core)  
ที่มา : อภิลิทธิ์ รัตนตรานูรักษ์. (2559)  
วาดภาพ



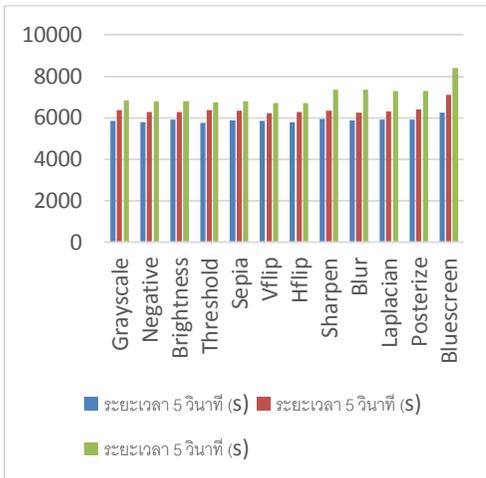
ภาพที่ 5 เวลาการประมวลผลวิดีโอโดยใช้ เว็บซีแอล (Multi-core CPU)  
ที่มา : อภิลิทธิ์ รัตนตรานูรักษ์. (2559)  
วาดภาพ



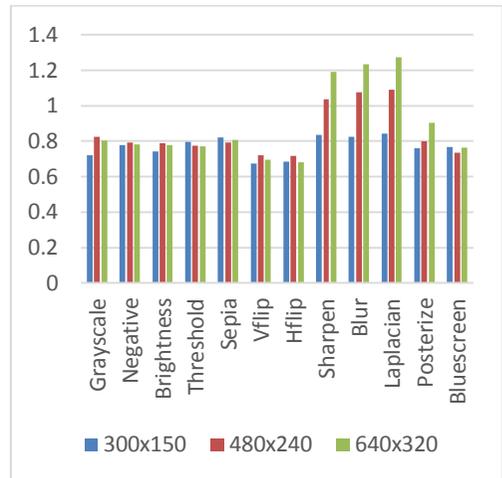
ภาพที่ 6 เวลาการประมวลผลวีดีโอโดย  
เว็บซีแอล (Integrated GPU)  
ที่มา : อภิสทิธิ รัตนาตรานุรักษ์. (2559)  
วาดภาพ



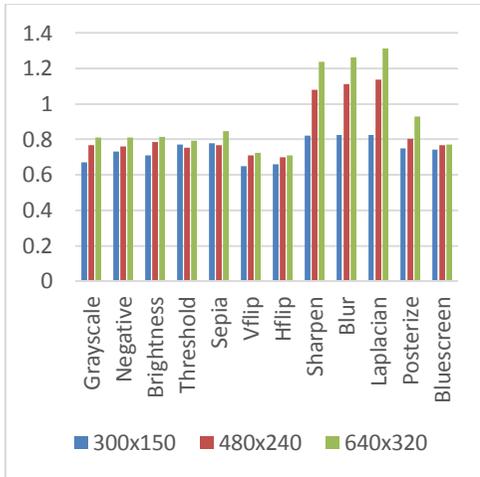
ภาพที่ 8 Speedup ของการประมวลผล  
วีดีโอโดยเว็บซีแอล (Multi-core CPU)  
ที่มา : อภิสทิธิ รัตนาตรานุรักษ์. (2559)  
วาดภาพ



ภาพที่ 7 เวลาการประมวลผลวีดีโอโดย  
เว็บซีแอล (NVIDIA GPU)  
ที่มา : อภิสทิธิ รัตนาตรานุรักษ์. (2559)  
วาดภาพ



ภาพที่ 9 Speedup ของการประมวลผล  
วีดีโอโดยเว็บซีแอล (Integrated GPU)  
ที่มา : อภิสทิธิ รัตนาตรานุรักษ์. (2559)  
วาดภาพ



ภาพที่ 10 Speedup ของการประมวลผล  
วิดีโอโดยเว็บซีแอล (NVIDIA GPU)  
ที่มา : อภิสสิทธิ์ รัตนาตราวุฑ์ (2559)  
วาดภาพ

### เอกสารอ้างอิง

Khronos OpenCL Working Group  
(2010). The OpenCL  
Specification (version  
1.1). NVIDIA (2009). NVIDIA  
OpenCL Jump start guide  
AMD (2011). OpenCL  
programming guide. URL  
<http://developer.amd.com/sdks/AMDAPPSDK/documentation>.

- Piao, X., Oh, Y., Kim, H., & Lee, J. W. (2014, April). Efficient CPU-GPU work sharing for data-parallel JavaScript workloads. In Proceedings of the 23rd International Conference on World Wide Web (pp. 357-358). ACM.
- MacWilliam, T., & Cecka, C. (2013, July). CrowdCL: Web-based volunteer computing with WebCL. In HPEC (pp. 1-6).
- Herhut, S., Hudson, R. L., Shpeisman, T., & Sreeram, J. (2012). Parallel programming for the web. In Presented as part of the 4th USENIX Workshop on Hot Topics in Parallelism.
- Jeon, W., Brutch, T., & Gibbs, S. (2012, April). WebCL for hardware-accelerated web applications. In TIZEN Developer Conference May (pp. 7-9).
- Wang, J., Rubin, N., & Yalamanchili, S. (2014, March). ParalleUS: An Execution Framework for JavaScript on Heterogeneous Systems. In Proceedings of Workshop on General Purpose Processing Using GPUs (p. 72). ACM.

Aarnio, T., & Bourges-Sevenier, M.  
(2012). WebCL working draft.  
Khronos WebCL Working  
Group. Cushing, R., Putra, G.  
H. H., Koulouzis, S.,  
Belloum, A., Bubak, M., & De  
Laat, C. (2013). Distributed  
computing on an ensemble  
of browsers. *IEEE Internet  
Computing*, 17(5), 54-61.