

การจับการเคลื่อนไหวสามมิติบนหน่วยประมวลผลกราฟิกโดยใช้คูดา 3D Motion Capture on Graphic Processing Unit using CUDA

ธนพนธ์ จันทรวงศ์สาลิ^{1*}, ไอรวิณ มงคลการ², ศิริชัย จูฑิม³, อภิสิทธิ์ รัตนตรานุรักษ์⁴

^{1*,2,3,4} สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะเทคโนโลยีอุตสาหกรรม
มหาวิทยาลัยราชภัฏสวนสุนันทา Email : apisit.ra@ssru.ac.th

บทคัดย่อ

ในบทความนี้กล่าวถึงการจับการเคลื่อนไหวสามมิติบนหน่วยประมวลผลกราฟิกโดยใช้คูดาซึ่งเป็นเฟรมเวิร์คในการพัฒนาโปรแกรมบนหน่วยประมวลผลกราฟิกของ NVIDIA บทความนี้เริ่มต้นโดยการวิเคราะห์เวลาในการสร้างหุ่นในแต่ละส่วน ได้แก่ แขนและมือขวา แขนและมือซ้าย หัวและลำตัว ขาและเท้าขวา ขาและเท้าซ้าย จากนั้นได้มีการนำคูดาเข้าไปประยุกต์ในอัลกอริธึมเพื่อใช้ในการสร้างหุ่นโดยใช้หน่วยประมวลผลกราฟิก ผลลัพธ์ที่ได้พบว่าค่าเฉลี่ยของความเร็วที่เพิ่มขึ้น คือ 73.74 เท่า บน 96 แกนคูดา

คำสำคัญ : หน่วยประมวลผลกราฟิก, การจับเคลื่อนไหวสามมิติ, คูดา

Abstract

This paper presents 3D Motion Capture on Graphic Processing Unit using CUDA which is the framework for developing program on NVIDIA Graphic Processing Unit. The paper starts with analysis run time to construct each part of model such as right arm and right hand, left arm and left hand, head and body, right leg and right foot, left leg and left foot. We then implemented CUDA to this algorithm to execute the model construction algorithm on Graphic Processing Unit. The results show that average Speed up of Graphic Processing Unit is 73.74x on 96 CUDA cores.

Keywords : Graphic Processing Unit, 3D Motion Capture, CUDA.

บทนำ

การจับการเคลื่อนไหว เป็นการสังเกตความเคลื่อนไหว โดยสามารถนำไปใช้ประโยชน์ไปได้หลายประเภทไม่ว่าจะเป็นสาขาวิทยาศาสตร์การกีฬาการสร้างภาพยนตร์ การสร้างการ์ตูนสามมิติ โดยการจับการเคลื่อนไหวนั้นจะนำการเคลื่อนไหวที่บันทึกได้ไปประมวลผลเป็นหุ่นสามมิติที่สามารถเคลื่อนไหวได้ ตามต้นแบบที่ต้องการ อย่างไรก็ตามการจับเคลื่อนไหวสามมิตินั้นใช้ทรัพยากรในการประมวลผลสูงมาก เนื่องจากต้องมีการประมวลผลลงในพิกัดสามมิติ จากต้นฉบับ นอกจากนั้นยังต้องมีการสร้างหุ่นสามมิติขึ้น โดยทรัพยากรที่ใช้นั้นจะขึ้นอยู่กับความละเอียดของหุ่นสามมิติอีกด้วย

ในการจับการเคลื่อนไหวสามมิติ (Moeslund, T. B., & Granum, E., 2001) มีหลายวิธีการ ซึ่งวิธีการที่น่าสนใจ คือ การใช้กล้องในการจับการเคลื่อนไหวสามมิติ เนื่องจากเป็นวิธีที่ไม่ต้องใช้เครื่องมือมากในการจับภาพ ทำให้สะดวกในการพัฒนา และการทำงาน โดยปัจจุบันนั้นยังมีเครื่องมือในการบันทึกภาพมากขึ้น และมีความละเอียดในการบันทึกผล ซึ่งทำให้สะดวกมากยิ่งขึ้นในการพัฒนาเครื่องมือในการจับการเคลื่อนไหวสามมิติ

ในปัจจุบันนั้นหน่วยประมวลผลกลาง (Central Processing Unit: CPU) ได้ถูกพัฒนาให้มีจำนวนแกนในการประมวลผลมากขึ้น ส่งผลให้สามารถทำงานได้หลายงานพร้อมกัน นอกจากนี้หน่วยประมวลผล

กราฟิก (Graphic Processing Unit: GPU) ยังได้รับการพัฒนาให้สามารถนำมาใช้ในการประมวลผลทั่วไปได้ โดยหน่วยประมวลผลดังกล่าวมีจำนวนแกนอยู่มากกว่าหน่วยประมวลผลกลาง ซึ่งหากสามารถดึงทรัพยากรเหล่านี้นมาใช้ร่วมกับการจับการเคลื่อนไหวสามมิติได้ จะส่งผลให้การประมวลผลนั้นใช้เวลาอันน้อยลง และมีประสิทธิภาพมากยิ่งขึ้น

ในการประมวลผลด้วยหน่วยประมวลผลโดยนำเอาหน่วยประมวลผลกราฟิกเข้ามาใช้นั้น สามารถนำไลบรารี Compute Unified Device Architecture : CUDA (NVIDIA, C. U. D. A., 2007) เข้ามาพร้อมด้วย และเรียกใช้หน่วยประมวลผลกราฟิกเพื่อใช้ในการประมวลผลได้ ซึ่งไลบรารีนี้สามารถพัฒนาร่วมกันกับภาษา C/C++ ทำให้สะดวกต่อการพัฒนาให้ การจับการเคลื่อนไหวสามมิติ ให้สามารถดึงทรัพยากรหน่วยประมวลผลกราฟิกมาใช้ให้มีประสิทธิภาพสูงสุด อย่างไรก็ตามไลบรารีลักษณะเดียวกันนี้ที่ยังมีไลบรารี Open Computing Language : OpenCL (Khronos, 2010) (NVIDIA, 2009) (AMD, 2011) ซึ่งสามารถดึงหน่วยประมวลผลต่าง ๆ เช่น หน่วยประมวลผลกราฟิก มาใช้ในการประมวลผลได้เช่นกัน แต่อย่างไรก็ตามหากมีการพัฒนาแอปพลิเคชันด้วยหน่วยประมวลผลกราฟิกของ NVIDIA การเลือกใช้ CUDA ซึ่งพัฒนาโดย NVIDIA เอง จะมีประสิทธิภาพในการทำงานสูงกว่า

จากปัญหาข้างต้นนั้น ผู้วิจัยมีความเห็นว่า หากมีการนำหน่วยประมวลผลกราฟิก เข้ามาพัฒนาให้มีการประมวลผลแบบขนานโดยใช้ CUDA มาช่วยในการประมวลผลการจับการเคลื่อนไหวสามมิติ จะส่งผลให้เวลาที่ใช้ในการประมวลผลลดลง

งานวิจัย DreamWorld ของ (Shujun, Z., Cong, W., Xuqiang, S., & Wei, W., 2009) ได้มีการนำเอา CUDA มาช่วยในการสร้างโมเดลสามมิติ ซึ่งข้อมูลที่ได้มีความเที่ยงตรง และสามารถทำงานแบบ Real time ได้

งานวิจัยของ Kwolek et al. (Kwolek, B., Krzeszowski, T., & Wojciechowski, K., 2011) ได้มีการประยุกต์ Particle Swarm Optimization เข้ามาช่วยในการจับการเคลื่อนไหวสามมิติของมนุษย์ จากกล้องหลายตัว ซึ่งใช้จำนวน particle น้อยกว่าปกติ เพื่อจับการเคลื่อนไหวให้มีประสิทธิภาพมากยิ่งขึ้น

งานวิจัยของ Zheng Zhang และ Hock Soon Seah (Zhang, Z., & Seah, H. S., 2012) ได้มีการนำเอาหน่วยประมวลผลกราฟิกมาช่วยในการทำหุ่น สามมิติ โดยมีการตรวจสอบฉาก และการเคลื่อนที่ของแบบ และมีการนำเอา Particle Swarm Optimization เข้ามาช่วยในการประมาณการทำของแบบบนหน่วยประมวลผลกราฟิก

งานวิจัยของ Xiaolin Wei et al. (Wei, X., Zhang, P., & Chai, J., 2012)

ได้มีการจับการเคลื่อนไหวสามมิติ โดยใช้กล้องเดียว ซึ่งมีการแก้ปัญหานี้โดยใช้วิธี Maximum A Posteriori (MAP) รวมทั้งการเอารูปแบบของมนุษย์ใส่เข้าไปซ้ำ ๆ เพื่อให้ระบบสามารถตรวจสอบความลึกได้งานชิ้นนี้สามารถแก้ปัญหาการตรวจจับการเคลื่อนไหวที่ผิดพลาด และสามารถนำไปพัฒนาแบบขนาน โดยใช้หน่วยประมวลผลกราฟิกได้

งานวิจัยของ Bian et al. (Bian, Y., Zhao, X., Song, J., & Liu, Y., 2012) ได้มีการพัฒนาการจับการเคลื่อนไหวสามมิติโดยไม่ต้องใช้มาร์คเกอร์ และมีการตรวจจับการเคลื่อนไหว แบบขนาน โดยใช้ไลบรารีโอเพนซีแอล (OpenCL) ซึ่งผลลัพธ์ที่ได้สามารถทำงานได้แบบ Real time และมีความแม่นยำ อยู่ในอัตราส่วน 106 เมื่อเปรียบเทียบกับการทำงานแบบตามลำดับ ซึ่งขึ้นอยู่กับจำนวน particle ที่ตรวจจับ

งานวิจัยของ Muratov et al. (Muratov, O., Slynko, Y., Chernov, V., Lyubimtseva, M., Shamsuarov, A., & Bucha, V., 2016) ได้มีการนำเสนอวิธีการจับการเคลื่อนไหวสามมิติ โดยใช้สมาร์ตโฟน ซึ่งเป็นกล้องเดียว ซึ่งในงานวิจัยนี้ได้นำเสนอถึงวิธีการคำนวณหาความลึก จากนั้นจึงสร้างวัตถุสามมิติขึ้น รวมทั้งมีการนำไปประมวลผลโดยใช้ OpenCL บนเครื่องที่สามารถทำงานบน OpenCL ได้

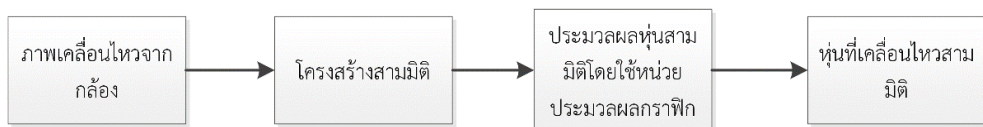
งานวิจัยของ Mokhov et al. (Mokhov, S. A., Song, M., Llewellyn,

J., Zhang, J., Charette, A., Wu, R., & Ge, S., 2016) ได้มีการนำเสนอวิธีการนำเอา Kinect 3 เครื่อง มาทำงานร่วมกันภายใน 1 แอปพลิเคชัน เพื่อแก้ปัญหาความผิดพลาดระหว่างการทำงานของการทำงานของเครื่องเคลื่อนไหวสามมิติ อย่างไรก็ตามงานวิจัยนี้ยังไม่สามารถทำงานได้บน CUDA เนื่องจากข้อจำกัดของเครื่องมือวิจัย

ในบทความนี้กล่าวถึงเนื้อหาตามลำดับดังนี้ วัตถุประสงค์ของการวิจัยซึ่งหัวข้อนี้จะแสดงถึงวัตถุประสงค์การวิจัย จากนั้นจะนำเสนอเกี่ยวกับวิธีการดำเนินงานวิจัย เพื่ออธิบายขั้นตอนการทำการวิจัย และวิธีการทดลอง จากนั้นจะมีการนำเสนอผลการทดลองที่ได้ และจะมีการสรุป อภิปรายผลเป็นส่วนสุดท้าย และแนวทางในการพัฒนาต่อ

วัตถุประสงค์ของการวิจัย

ในงานวิจัยชิ้นนี้มีวัตถุประสงค์เพื่อพัฒนาการจับภาพวิดีโอสามมิติจากกล้อง Kinect ที่มีการทำงานแบบตามลำดับ ให้มีการทำงานแบบขนาน โดยมีการดึงทรัพยากรจากหน่วยประมวลผลอื่นคือ หน่วยประมวลผลกราฟิก มาใช้ในการประมวลผลภาพสามมิติได้ เพื่อให้เวลาที่ใช้ในการประมวลผลลดลง โดยใช้ CUDA Toolkit ในงานนี้จะมีการทดสอบการประมวลผลวิดีโอหุ่นสามมิติของมนุษย์ และทดสอบหา ความเร็วที่เพิ่มขึ้นเพื่อเปรียบเทียบหาประสิทธิภาพ



ภาพที่ 1 รูปแบบการพัฒนาอัลกอริธึม

ที่มา : อภิสิตธี รัตนตรานุรักษ์. (2559) วาดภาพ

วิธีดำเนินการวิจัย

ในงานวิจัยนี้ จะมีการพัฒนาแอปพลิเคชันสำหรับจับภาพเคลื่อนไหวสามมิติ โดยนำเอาภาพเคลื่อนไหวที่ได้ไปประมวลผล และสร้างตำแหน่งสามมิติบนหน่วยประมวลผลกราฟิก เมื่อประมวลผลแล้วจะได้ผลลัพธ์เป็นหุ่นสามมิติของมนุษย์ที่ประมวลผลภาพแล้ว ซึ่งในงานวิจัยนี้

ผู้วิจัยได้มีการเรียกใช้ไลบรารีจากกล้อง Kinect เพื่อเชื่อมต่อเข้ากับซอฟต์แวร์ที่พัฒนาขึ้น โดยงานนี้จะมีการปรับปรุงการประมวลผลหุ่นสามมิติของมนุษย์จากอัลกอริธึมต้นฉบับ ให้มีการทำงานแบบขนานโดยใช้ CUDA Toolkit และสร้างหุ่นสามมิติของมนุษย์ขึ้น โดยวิธีการดำเนินการวิจัยนี้มีกระบวนการดังภาพที่ 1

จากภาพที่ 1 ภาพเคลื่อนไหวสามมิติ จะได้รับการบันทึกผลจากกล้อง โดยในงานวิจัยชิ้นนี้ผู้วิจัยได้เลือกใช้กล้อง Kinect ในการรับภาพเคลื่อนไหว โดยภาพเคลื่อนไหวที่ได้นั้นจะได้รับการประมวลผลจากกล้อง Kinect และส่งข้อมูลโครงสร้างสามมิติมาให้แอปพลิเคชัน จากนั้นแอปพลิเคชันที่จะมีการประมวลผลหุ่นสามมิติของมนุษย์จากโครงสร้างที่ได้ โดยการส่งค่าตำแหน่งข้อต่อไปยังหน่วยประมวลผลกราฟิก เพื่อให้ประมวลผล และสร้างหุ่นสามมิติของมนุษย์ขึ้นมาในทุกเฟรม

ในการประมวลผลแบบตามลำดับนั้น ในการประมวลผลการจับภาพเคลื่อนไหวสามมิติเพื่อสร้างเป็นหุ่นสามมิติของมนุษย์นั้นจะต้องมีการประมวลผลในแต่ละเฟรม โดยในแต่ละเฟรมจะประกอบไปด้วย การประมวลผลโครงสร้างของหุ่นสามมิติของมนุษย์ในแต่ละส่วน โดยแต่ละส่วนนั้นจะต้องมีการต่อจุดของโครงสร้าง เพื่อสร้างหุ่นสามมิติของมนุษย์ ซึ่งโดยปกติแล้ว การสร้างหุ่นโดยทั่วไปจะมีการทำงานแบบตามลำดับ และวนซ้ำ สร้างหุ่นจากโครงสร้างไปเรื่อย ๆ จนกระทั่งครบทุกส่วนประกอบของหุ่น ดังภาพที่ 2

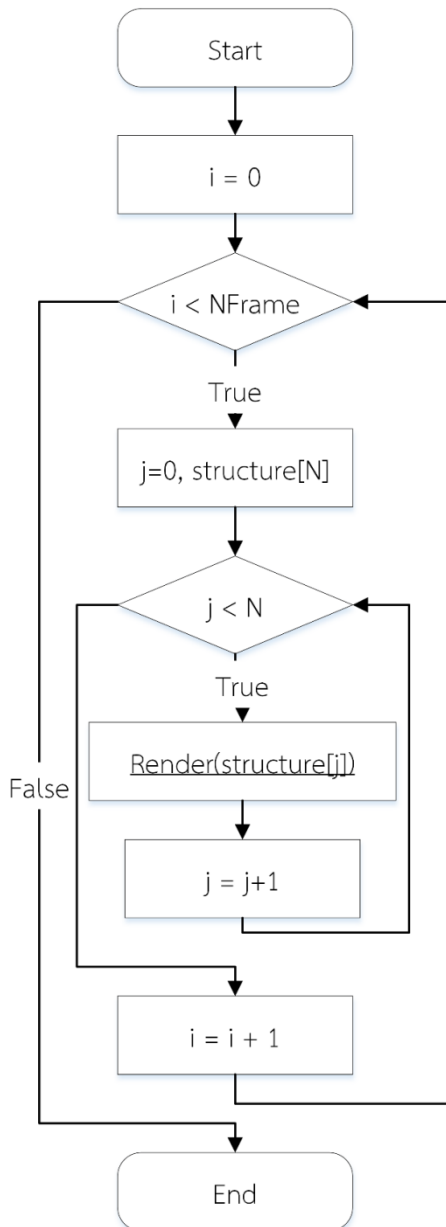
ในการพัฒนาให้การสร้างหุ่นสามมิติของมนุษย์นั้นมีการทำงานแบบขนานได้จะต้องมีการปรับปรุง 2 ส่วน คือส่วนของโฮสต์ (Host) หรือในที่นี้คือฝั่งหน่วยประมวลผลกลาง และส่วนของอุปกรณ์ (Devices) ในที่นี้คือหน่วยประมวลผลกราฟิก โดยในส่วนของโฮสต์นั้นจะมี

การปรับปรุงโดยการสร้าง อาร์เรย์ เก็บค่าโครงสร้างของหุ่นจากนั้นทุกเฟรม และทุกส่วนโครงสร้าง จากนั้นกำหนดตำแหน่งค่าใน block ของหน่วยประมวลผลกราฟิกด้วย blockIdx ตามด้วยแกนที่กำหนด ซึ่งในที่นี้สามารถประมวลผลได้พร้อมกัน จึงมีการกำหนดทั้ง x และ y ดังภาพที่ 3

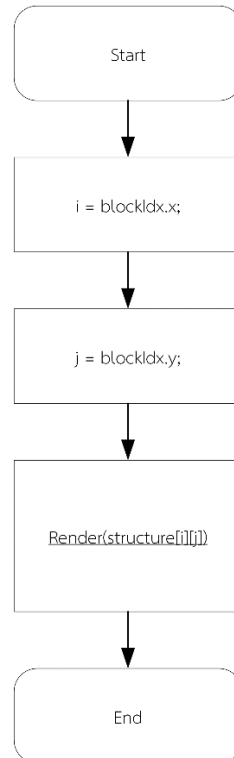
การประมวลผลลักษณะนี้มีข้อจำกัดด้วยจำนวนแกนประมวลผลของหน่วยประมวลผลกราฟิก เนื่องจากข้อมูลมีจำนวนมากกว่าแกนหน่วยประมวลผล แม้ว่าจำนวนแกนหน่วยประมวลผลจะมีมากก็ตาม หากแกนประมวลผลมีค่าเป็น c และข้อมูลเป็น x.y งานที่แบ่งไปจะได้เพียง x.y/c เท่านั้น

ผลการวิจัย

ในงานวิจัยนี้ได้มีการพัฒนาแอปพลิเคชันในการจับภาพสามมิติโดยใช้หน่วยประมวลผลกราฟิก บนตัวแปร และสภาพแวดล้อม ใช้เครื่องคอมพิวเตอร์ Intel Core i7 ซึ่งประกอบไปด้วยหน่วยประมวลผลกลาง 4 แกน และสามารถทำงานแบบ Hyper Thread ได้ โดยมีหน่วยความจำหลัก 8 GB หน่วยประมวลผลกราฟิก NVIDIA Geforce GT 630M ประกอบไปด้วย VRAM 2 GB บนระบบปฏิบัติการ Windows 8.1 64 bit นอกจากนี้ใช้ NVIDIA SDK version 1.8 ซึ่งประกอบไปด้วย CUDA version 6.5 และใช้ OpenGL เพื่อการสร้างหุ่นสามมิติของมนุษย์ version 4.5 ในส่วนของ Kinect ใช้ version 1.8 ดังตารางที่ 1



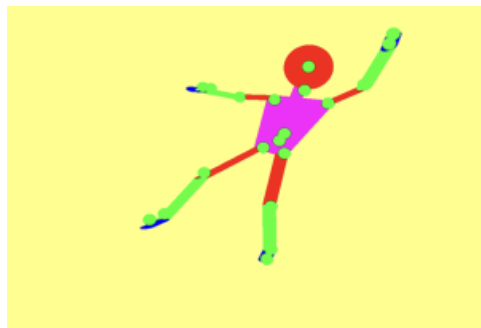
ภาพที่ 2 ผังงานของอัลกอริธึมที่ทำงานแบบตามลำดับ เมื่อ structure คือโครงสร้างของหุ่นแต่ละส่วน NFrame คือ จำนวนเฟรมทั้งหมด และ N คือ จำนวนโครงสร้างของหุ่น
ที่มา : อภิสิทธิ์ รัตนตรานุรักษ์. (2559) วาดภาพ



ภาพที่ 3 ผังงานของอัลกอริธึมที่ทำงานแบบขนาน เมื่อ structure คือโครงสร้างของหุ่นที่เฟรม i และ โครงสร้าง j
ที่มา : อภิสิทธิ์ รัตนตรานุรักษ์. (2559) วาดภาพ

ในการทดสอบจะมีการทดสอบหาเวลาส่วนที่ใช้ไปในการสร้างหุ่นสามมิติของมนุษย์ส่วนใหญ่ โดยจะทดสอบจับเวลาในการสร้างหุ่นแต่ละส่วน บนหน่วยประมวลผลกลาง ได้แก่ส่วนแขนซ้าย และมือซ้าย ส่วนแขนขวาและมือขวา ส่วนหัวและลำตัว ส่วนขาซ้ายและเท้าซ้าย ส่วนขาขวาและเท้าขวา ดังภาพที่ 4 เพื่อหาบริเวณที่ใช้เวลาในการสร้างหุ่นที่มากที่สุด เพื่อตัดสินใจในการเลือกพัฒนาให้มี

การทำงานแบบขนานด้วยหน่วยประมวลผลกราฟิก โดยเมื่อพัฒนาแบบขนานแล้วจะมีการเปรียบเทียบกันของเวลาระหว่างหน่วยประมวลผลกลาง และหน่วยประมวลผลกราฟิก และหาความเร็วที่เพิ่มขึ้น เปรียบเทียบกัน ซึ่งจะทดลองบน 300 เฟรม คิดค่าเฉลี่ย 10 ครั้ง โดยตัวแปรที่ใช้ในการทดลองได้แสดงไว้ดังตารางที่ 2



ภาพที่ 4 หุ่นสามมิติที่สร้างขึ้น
ที่มา : ธนพนธ์. (2559) วาดภาพ

ตารางที่ 1 สภาพแวดล้อมในการทดลอง

ชนิดของสภาพแวดล้อม	สภาพแวดล้อมที่ใช้ในการทดลอง
หน่วยประมวลผลกลาง	Intel Core i7-3610QM 2.3 GHz 4 cores with Hyper Threading Technology
หน่วยความจำหลัก	8 GB
หน่วยประมวลผลกราฟิก	NVIDIA Geforce GT 630M VRAM 2GB
ระบบปฏิบัติการ	Windows 8.1
CUDA version	6.5
OpenGL version	4.5
NVIDIA SDK version	1.8
Kinect version	1.8

ตารางที่ 2 ตัวแปรที่ใช้ในการทดลอง

ตัวแปร	ค่า
ส่วนของหุ่นสามมิติของมนุษย์	ส่วนแขนซ้ายและมือซ้าย ส่วนแขนขวาและมือขวา หัว และลำตัว ส่วนขาซ้ายและเท้าซ้าย ส่วนขาขวาและเท้าขวา
อุปกรณ์ที่ใช้ในการประมวลผล	หน่วยประมวลผลกลาง หน่วยประมวลผลกราฟิก
จำนวนเฟรม	300 เฟรม (ขนาด 640x480 พิกเซล ที่ 30 เฟรม/วินาที)

ตารางที่ 3 ค่าเฉลี่ยเวลา (ไมโครวินาที) ในการสร้างหุ่นสามมิติ ในแต่ละส่วน ต่อ 1 เฟรม บนหน่วยประมวลผลกลาง

	ส่วนแขนซ้าย และมือซ้าย	ส่วนแขนขวา และมือขวา	หัวและลำตัว	ส่วนขาซ้าย และเท้าซ้าย	ส่วนขาขวา และเท้าขวา
เวลา	435.19	398.31	347.48	412.81	467.95
ร้อยละ	21.11	19.32	16.85	20.02	22.70

ตารางที่ 4 ค่าเฉลี่ยเวลา (ไมโครวินาที) ในการสร้างหุ่นสามมิติ ในแต่ละส่วน ต่อ 1 เฟรม บนหน่วยประมวลผลกราฟิก

	ส่วนแขนซ้าย และมือซ้าย	ส่วนแขนขวา และมือขวา	หัวและลำตัว	ส่วนขาซ้าย และเท้าซ้าย	ส่วนขาขวา และเท้าขวา
เวลา	5.9589	5.4557	4.6735	5.6245	6.2642
ร้อยละ	21.30	19.50	16.70	20.10	22.40

ตารางที่ 5 การเปรียบเทียบความเร็วที่เพิ่มขึ้นกับงานวิจัยที่เกี่ยวข้อง

	งานวิจัยนี้	Shujun, Z., Cong, W., Xuqiang, S., & Wei, W., 2009	Bian, Y., Zhao, X., Song, J., & Liu, Y., 2012
ความละเอียด	640x480 (30 fps)	640x512 (30 fps)	704x576
ความเร็วที่เพิ่มขึ้น	73.74	46.82	105.07

จากตารางที่ 3 แสดงเวลาในการสร้างหุ่นในแต่ละเฟรม ได้แก่ ส่วนแขนซ้ายและมือซ้ายใช้เวลาร้อยละ 21.11 ส่วนแขนขวาและมือขวาใช้เวลาร้อยละ 19.32 หัวและลำตัวใช้เวลาร้อยละ 16.85 ส่วนขาซ้ายและเท้าซ้ายใช้เวลาร้อยละ 20.02 และส่วนขาขวาและเท้าขวา ใช้เวลาร้อยละ 22.70 โดยเฉลี่ย ซึ่งจะเห็นว่าแต่ละส่วนใช้เวลาใกล้เคียงกัน ประมาณ 20% ดังนั้น ความสำคัญในการพัฒนาให้มีการทำงานแบบขนานโดยใช้หน่วยประมวลผลกราฟิก

ในแต่ละส่วนมีความสำคัญใกล้เคียงกัน ดังนั้นในงานวิจัยนี้จะมีการพัฒนาให้แอปพลิเคชันมีการทำงานแบบขนานในทุกส่วน จากภาพที่ 5 แสดงเวลาที่ใช้ในการประมวลผลเปรียบเทียบระหว่างหน่วยประมวลผลกลาง และหน่วยประมวลผลกราฟิก ซึ่งจะเห็นว่าจากการทดลองหน่วยประมวลผลกราฟิกใช้เวลาที่ใช้ในการประมวลผลน้อยกว่าหน่วยประมวลผลกลาง โดยค่าเฉลี่ยของหน่วยประมวลผลกลางอยู่ที่ 2188.85 ไมโครวินาที และค่าเฉลี่ย

ของหน่วยประมวลผลกราฟิกอยู่ที่ 29.70 ไมโครวินาที เมื่อทดสอบบน 300 เฟรม

เมื่อนำเวลามาวิเคราะห์ในแต่ละส่วนของหุ่นสามมิติของมนุษย์ จะได้ผลการทดลองดังตารางที่ 4 ซึ่งจะเห็นว่าเวลาที่ใช้ในการประมวลผลแต่ละส่วนมีค่าลดลงในร้อยละที่มีความใกล้เคียงกัน

จากภาพที่ 6 แสดงความเร็วที่เพิ่มขึ้นของหน่วยประมวลผลกราฟิก ซึ่งค่าความเร็วที่เพิ่มขึ้น เป็นสัดส่วนระหว่างเวลาที่ใช้ในการประมวลผลด้วยหน่วยประมวลผลกลาง และเวลาที่ใช้ในการประมวลผลด้วยหน่วยประมวลผลกราฟิก จะเห็นว่าค่าเฉลี่ยของ ความเร็วที่เพิ่มขึ้นอยู่ที่ 73.74 เมื่อทดสอบบน 300 เฟรม และได้มีการเปรียบเทียบความเร็วที่เพิ่มขึ้นกับงานวิจัยที่เกี่ยวข้องดังตารางที่ 5

สรุปและอภิปรายผล

ในงานวิจัยชิ้นนี้จะเห็นว่ามีการพัฒนาแอปพลิเคชันการจับการเคลื่อนไหวสามมิติที่ทำงานบนหน่วยประมวลผลกลาง ให้มีการประมวลผลบนหน่วยประมวลผลกราฟิก โดยใช้ CUDA ซึ่งเป็นไลบรารีของ NVIDIA ซึ่งความเร็วที่เพิ่มขึ้น ที่ได้อยู่ที่ 73.74 เมื่อทดสอบบนการจับเคลื่อนไหว 300 เฟรม

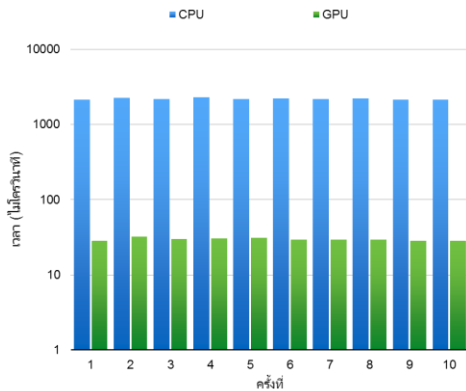
จากการทดสอบในการจับการเคลื่อนไหวสามมิติโดยนำมาสร้างเป็นหุ่น โดยมีการทดลองในแต่ละส่วน ซึ่งผลลัพธ์ที่ได้ปรากฏว่า ในการสร้างหุ่นแต่ละส่วนนั้นใช้เวลาในการประมวลผลใกล้เคียงกัน และใช้

เวลาในการประมวลผลสูง ประมาณ 20% ซึ่งด้วยความใกล้เคียงกันนี้ จึงสามารถนำไปพัฒนาโปรแกรมให้มีการประมวลผลบนหน่วยประมวลผลกราฟิกได้

นอกจากนี้การทดลองจะเห็นว่า แม้ว่าหน่วยประมวลผลกราฟิกจะมีความถี่สัญญาณนาฬิกาแต่ละแกนน้อยกว่าหน่วยประมวลผลกลาง แต่จำนวนแกนของหน่วยประมวลผลกราฟิกนั้นมีจำนวนมากกว่าหน่วยประมวลผลกลางมาก ซึ่งมีจำนวน CUDA cores ถึง 96 แกน ดังนั้น เวลาที่ใช้ในการสร้างหุ่นสามมิติของมนุษย์จึงใช้น้อยกว่า

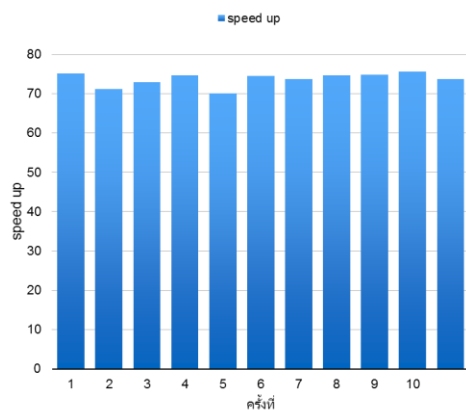
งานในอนาคตของงานวิจัยชิ้นนี้สามารถพัฒนาให้มีประสิทธิภาพมากขึ้น โดยการสร้างหุ่นให้มีความละเอียดสูงมากยิ่งขึ้น จะส่งผลให้ใช้เวลาในการสร้างหุ่นสูงมากขึ้นเช่นกันเมื่อใช้หน่วยประมวลผลกลาง หากส่งไปให้หน่วยประมวลผลกราฟิกไปประมวลผลจะส่งผลให้ใช้เวลาไม่สูงเท่าหน่วยประมวลผลกลาง

นอกจากนี้ยังสามารถพัฒนาการสร้างหุ่นสามมิติของมนุษย์ให้มีประสิทธิภาพมากขึ้นโดยการนำเอาหน่วยประมวลผลกลางประมวลผลพร้อมกันกับหน่วยประมวลผลกราฟิก ซึ่งอาจจะใช้ไลบรารี OpenCL ในการประมวลผลแทน CUDA ซึ่งจะส่งผลให้ประสิทธิภาพในการสร้างหุ่นสามมิติของมนุษย์เพิ่มมากยิ่งขึ้น



ภาพที่ 5 เวลาการประมวลผลระหว่างหน่วยประมวลผลกลาง และหน่วยประมวลผลกราฟิก 10 ครั้ง

ที่มา : อภิสิทธิ์ รัตนตรานุรักษ์. (2559) วาดภาพ



ภาพที่ 6 ความเร็วที่เพิ่มขึ้นของการประมวลผล โดยใช้ หน่วยประมวลผลกราฟิก 10 ครั้ง

ที่มา : อภิสิทธิ์ รัตนตรานุรักษ์. (2559) วาดภาพ

เอกสารอ้างอิง

AMD (2011). OpenCL programming guide. จาก <http://developer.amd.com/sdks/AMDAPPSDK/documentation>.

Bian, Y., Zhao, X., Song, J., & Liu, Y. (2012, November). Parallelized annealed particle filter for real-time marker-less motion tracking via heterogeneous computing. In Pattern Recognition (ICPR), 2012 21st International Conference on (pp. 2444-2447). IEEE.

Khronos OpenCL Working Group (2010). The OpenCL Specification (version 1.1).

Kwolek, B., Krzeszowski, T., & Wojciechowski, K. (2011, August). Swarm intelligence based searching schemes for articulated 3D body motion tracking. In International Conference on Advanced Concepts for Intelligent Vision Systems (pp. 115-126). Springer Berlin Heidelberg.

Moeslund, T. B., & Granum, E. (2001). A survey of computer vision-based human motion capture. Computer vision and image understanding, 81(3), 231-268.

Mokhov, S. A., Song, M., Llewellyn, J., Zhang, J., Charette, A., Wu, R., & Ge, S. (2016, July). Real-time collection and analysis of 3-Kinect v2 skeleton data in a single application. In ACM SIGGRAPH 2016 Posters (p. 53). ACM.

- Muratov, O., Slynko, Y., Chernov, V., Lyubimtseva, M., Shamsuarov, A., & Bucha, V. (2016). 3DCapture: 3D Reconstruction for a Smartphone. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 75-82).
- NVIDIA, C. U. D. A. (2007). Compute unified device architecture programming guide.
- NVIDIA. (2009). NVIDIA OpenCL Jump start guide
- Shujun, Z., Cong, W., Xuqiang, S., & Wei, W. (2009, May). DreamWorld: CUDA-accelerated real-time 3D modeling system. In 2009 IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurements Systems (pp. 168-173). IEEE.
- Wei, X., Zhang, P., & Chai, J. (2012). Accurate realtime full-body motion capture using a single depth camera. ACM Transactions on Graphics (TOG), 31(6), 188.
- Zhang, Z., & Seah, H. S. (2012, December). CUDA acceleration of 3D dynamic scene reconstruction and 3D motion estimation for motion capture. In Parallel and Distributed Systems (ICPADS), 2012 IEEE 18th International Conference on (pp. 284-291). IEEE.