

# ระบบสแกนวัตถุสามมิติด้วยอุปกรณ์คิเนค

## 3D Scanner System using a Kinect

จักรพันธ์ ยกเขื่น ปารเมศ ลิ้มตระกูล และฐิติวรรณ ศรีนาค

ห้องปฏิบัติการวิจัยนวัตกรรมวิทยาศาสตร์ (GIVE – Graphics Innovation and Vision Engineering Laboratory)

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

b5105553@ku.ac.th, b5105272@ku.ac.th, thitiwan.s@ku.ac.th

### บทคัดย่อ

การสร้างแบบจำลองจากวัตถุจริงโดยอาศัยเครื่องสแกนเนอร์สามมิติระดับมืออาชีพรยังคงเป็นเรื่องที่ทำได้ยากและไม่แพร่หลายนักเพราะเครื่องสแกนเนอร์ระดับนี้มีราคาสูงมาก แต่ทว่าในปัจจุบันการสร้างแบบจำลองสามมิติจากวัตถุจริงมีความเป็นไปได้โดยอาศัยเทคโนโลยีราคาไม่แพงจากอุปกรณ์กล้องคิเนค (Kinect) ในงานวิจัยนี้เรานำเสนอการพัฒนาาระบบสแกนเนอร์สามมิติโดยใช้กล้องคิเนค จากผลการทดลองแสดงให้เห็นว่า แบบจำลองมีความใกล้เคียงกับวัตถุ แต่ยังคงมีความผิดพลาดอยู่บ้างเนื่องจากความละเอียดของกล้องคิเนคที่ค่อนข้างต่ำ

### Abstract

*It is not practical and not popular to generate 3D models by using a professional 3D scanner because this kind of scanner is highly expensive. But now the creation of 3D models from real objects can come true due to the low price technology of Kinect. In this research, we propose a development of a 3D scanner by using Kinect. From the experiments, it can be seen that resulting models are similar to the real objects, but there are still some mistakes because of the low resolution images produced from Kinect.*

### คำสำคัญ

Kinect, Point Cloud Data, 3D scanner, Reconstruction, 3D models

### 1. บทนำ

อุปกรณ์หลายประเภทที่ใช้ในการสแกนวัตถุสามมิติมีหลักการคือ ใช้ลำแสงกราดไปที่วัตถุ และไล่สแกนเพื่อให้ได้ข้อมูลพื้นผิวของวัตถุเช่น CTscanner ซึ่งเป็นเครื่องสแกนที่ใช้รังสีในการเก็บข้อมูลเป็นสองมิติ จากนั้นนำข้อมูลผ่านการประมวลผลให้อยู่ในรูปแบบของข้อมูลสามมิติ หรืออุปกรณ์สแกนวัตถุด้วยเลเซอร์โดยกล้องที่ติดตั้งเซนเซอร์เพื่อเก็บข้อมูลจากการสแกนเข้ามาสำหรับสร้างแบบจำลองสามมิติ อย่างไรก็ตามอุปกรณ์เหล่านี้เหมาะสมกับการใช้งานเฉพาะทางมากกว่า เนื่องจากมีราคาที่สูงมากเกินกว่าที่คนทั่วไปหรือหน่วยงานขนาดเล็กจะซื้อหามาใช้ได้ ปัจจุบันบริษัทไมโครซอฟต์ได้พัฒนาอุปกรณ์สแกนเนอร์ที่เรียกว่า “คิเนค” สำหรับใช้ในการเล่นเกมกับเครื่อง XBOX360 ตัวกล้องคิเนคเองสามารถรับข้อมูลภาพ RGB และความลึกของพื้นผิววัตถุได้ในอุปกรณ์เพียงตัวเดียว ซึ่งข้อมูลดังกล่าวสามารถนำมาวิเคราะห์และประมวลผลเพื่อสร้างแบบจำลองสามมิติได้

### 2. ที่มาและแรงจูงใจของปัญหา

ปัจจุบันความต้องการใช้งานแบบจำลองสามมิติมีสูงขึ้นในหลายวงการได้แก่ ด้านการแพทย์เพื่อสร้างแบบจำลองของอวัยวะผู้ป่วยสำหรับการวินิจฉัย ด้านอุตสาหกรรมบันเทิงเพื่อใช้แบบจำลองสามมิติสำหรับตัวละคร วัตถุ หรือฉากภายในภาพยนตร์ หรือเกม เป็นต้น ดังนั้นการสร้างอุปกรณ์สแกนเนอร์ราคาถูกเพื่อช่วยในการสร้างแบบจำลองจึงเป็นสิ่งที่มีความต้องการสูง เพราะสามารถประหยัดงบประมาณในการจัดหาอุปกรณ์ราคาแพง และแบบจำลองที่ได้ยังเป็นประโยชน์แทนการขึ้นรูปสำหรับนักออกแบบงานกราฟิกส์อีกด้วย

### 3. งานและทฤษฎีที่เกี่ยวข้อง

#### อุปกรณ์คิเนค (Kinect)

อุปกรณ์คิเนค [1] เป็นอุปกรณ์ประเภทกล้องจากบริษัท ไมโครซอฟต์ มีลักษณะเป็นรูปแท่งวางตัวในแนวนอน ซึ่งเชื่อมต่อกับฐานขนาดเล็ก ที่แกนมีมอเตอร์ในตัว ซึ่งทำหน้าที่ ขยับมุมกล้อง ส่วนรับข้อมูลประกอบด้วยกล้อง RGB มีความละเอียดอยู่ที่ 640x480 พิกเซล กล้องคิเนคสามารถจับภาพเคลื่อนไหวได้ 30 เฟรมต่อวินาที นอกจากนี้ยังประกอบไปด้วยเซนเซอร์ตรวจจับความลึก (Infrared laser) และ ไมโครโฟนจำนวนสี่แถว การเชื่อมต่อกับคอมพิวเตอร์สามารถทำได้โดยใช้พอร์ต USB 2.0 รูปที่ 1 แสดงภาพกล้องคิเนค



รูปที่ 1 กล้องคิเนค (Kinect) [1]

#### การสร้างแบบจำลองสามมิติ (3D Modeling)

การสร้างแบบจำลองสามมิติ [2] คือกระบวนการในการคำนวณทางคณิตศาสตร์ของพื้นผิวในระบบสามมิติของวัตถุใด ๆ โดยผลที่ได้จะเรียกว่า “แบบจำลองสามมิติ” (3D Model) ซึ่งเราสามารถมองเห็นภาพแบบจำลองนี้ผ่านหน้าต่างแสดงผลได้โดยผ่านกระบวนการที่เรียกว่า “เรนเดอร์สามมิติ” (3D Rendering) แบบจำลองสามมิติเป็นตัวแทนของวัตถุสามมิติ โดยเก็บข้อมูลเป็นจุดในระบบพิกัดสามมิติ ซึ่งเชื่อมต่อกันด้วยรูปแบบทางเรขาคณิต เช่น สามเหลี่ยม เส้นโค้ง หรือพื้นผิวแบบจำลองสามมิติสามารถสร้างได้เองจากโปรแกรมประยุกต์เพื่อการออกแบบด้านกราฟิกส์สามมิติ โดยใช้ขั้นตอนวิธีการคำนวณหรือจากการสแกน

#### เมฆจุด (Point Cloud)

เมฆจุด [3] นำเสนอเซตของจุดในระบบพิกัดออร์โธโกนัลสามมิติ โดยปกติความสัมพันธ์ในพิกัดของ XYZ มักถูกใช้เป็นตัวแทนของพื้นผิวภายนอกของวัตถุ โดยส่วนมากเมฆจุด มักถูกสร้างมาจากเครื่องสแกนเนอร์สามมิติ ผลที่ได้จากการสแกน

ส่วนมากมักจะถูกใช้ในงานออกแบบชิ้นส่วน เพื่อตรวจสอบคุณภาพ เมฆจุดในงานสามมิติมักถูกแปลงให้อยู่ในรูปแบบจำลองพื้นผิว (Surface Model) ในลักษณะแบบจำลองโครงข่าย (Mesh Model) เพื่อทำให้ง่ายต่อความเข้าใจ และสะดวกต่อการใช้ประโยชน์ในงานที่หลากหลาย

#### Iterative Closest Point algorithm (ICP)

ICP [4] เป็นขั้นตอนวิธีที่ใช้ในการหาความสัมพันธ์การเปลี่ยนแปลงเชิงพื้นที่ (Spatial Transformation) ของข้อมูลเมฆจุด (Point Cloud Data-PCD) สองชุดที่มีความคล้ายคลึงกัน หรือซ้อนทับกันในบางลักษณะ ในการใช้ ICP เราต้องทำให้ข้อมูลเมฆจุดมีระยะห่างที่ต่างกันน้อยที่สุด เพื่อให้เกิดความถูกต้องมากที่สุด อินพุตของ ICP คือ ข้อมูลเมฆจุดสองชุดที่ได้จากการสแกนสำหรับพิจารณาการเปลี่ยนแปลงระหว่างชุดข้อมูล และกำหนดเงื่อนไขการสิ้นสุดการคำนวณของ ICP ที่สำคัญคือ ระยะห่างสูงสุดที่เป็นไปได้ระหว่างชุดข้อมูลทั้งสอง เอาท์พุตของ ICP คือ เมฆจุดที่มีเปลี่ยนแปลงเรียบร้อยแล้ว ขั้นตอนการทำงานคือ (1) หาความสัมพันธ์ของเมฆจุดที่อยู่ในเกณฑ์ที่กำหนด (2) พิจารณาการเปลี่ยนแปลงของพารามิเตอร์ (3) เปลี่ยนแปลงเมฆจุดตามพารามิเตอร์ที่ได้คำนวณไว้ และ (4) พิจารณาความสัมพันธ์ของเมฆจุดอีกครั้งหนึ่ง สำหรับข้อเสียของ ICP คือมีความเสี่ยงที่เกิดข้อผิดพลาดสะสม (Accumulation Error) จากการทำ ICP ของหลายชุดข้อมูลต่อเนื่องกัน

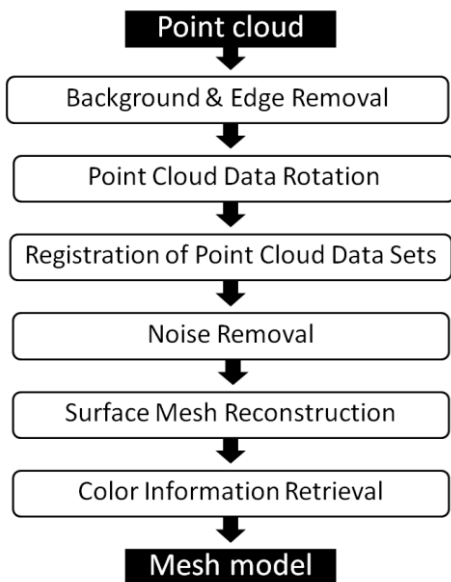
#### Poisson Surface Reconstruction Algorithm

Poisson surface reconstruction [5] ใช้สำหรับสร้างโครงสร้างพื้นผิวแบบโครงข่าย (Surface Mesh) จากข้อมูลเมฆจุด วิธีนี้เป็นที่นิยมในการสร้างพื้นผิวให้กับข้อมูลที่ได้มาจากเครื่องสแกนเนอร์ เพราะเนื่องจากมีความทนทานต่อสัญญาณรบกวนสูง แนวคิดโดยสังเขปคือ มุ่งหมายเพื่อแก้สมการในรูปแบบ Poisson โดยการประมาณ Indicator Function ของกลุ่มเมฆจุดจาก Indicator Gradient ของกลุ่มเมฆจุดเอง และมีการปรับค่า Scalar Function ซึ่งได้มาจากค่า Adaptive Octree เพื่อใช้ในการแก้สมการ Poisson สำหรับหาโครงสร้างพื้นผิวแบบโครงข่ายได้

#### 4. รายละเอียดการพัฒนา

##### 4.1 ภาพรวมของระบบ

ระบบสแกนวัตถุสามมิติด้วยอุปกรณ์ Kinect ได้รับข้อมูลนำเข้าเป็นลักษณะข้อมูลเมฆจุด หลังจากที่ได้ข้อมูลเมฆจุดมาแล้วกระบวนการโดยย่อคือ เราย้ายข้อมูลเมฆจุดเข้าสู่กระบวนการ Registration และกำจัดส่วนที่ไม่ต้องการออกจากระบบในขั้นตอนสุดท้าย วัตถุผลลัพธ์ที่ได้จากการทำ Registration จะเข้าสู่ขั้นตอน Meshing เพื่อสร้างพื้นผิวให้กับวัตถุ และทำการถ่ายโอนสีให้กับวัตถุเป็นขั้นตอนสุดท้าย โดยผลลัพธ์ที่ได้คือแบบจำลองต้นแบบสามมิติของวัตถุจริง ภาพรวมการทำงานของระบบได้แสดงดังรูปที่ 2



รูปที่ 2 ขั้นตอนการทำงานทั้งหมดของระบบ

##### 4.2 การออกแบบและพัฒนาระบบ

รายละเอียดการทำงานของระบบสแกนวัตถุสามมิติด้วยอุปกรณ์ Kinect มีขั้นตอนในการทำงานดังต่อไปนี้ เริ่มจากการรับข้อมูลจากอุปกรณ์ Kinect มาในลักษณะของข้อมูลเมฆจุด ซึ่งเราได้เก็บข้อมูลมาทั้งหมด 8 มุมมอง ระยะความห่างของแต่ละมุมมองห่างกัน 45 องศา หลังจากที่ได้เก็บข้อมูลเมฆจุดแต่ละมุมมองมาครบแล้ว เรายังจะนำข้อมูลเมฆจุดมาประมวลผลในขั้นตอนต่อไป

##### Background & Edge Removal

เนื่องจากข้อมูลเมฆจุดที่ได้มานั้นมีข้อมูลที่ไม่เกี่ยวข้องกับวัตถุติดมาด้วย เช่น พื้นหลัง ซึ่งเราสามารถกำจัดด้วยการใช้ Pass-

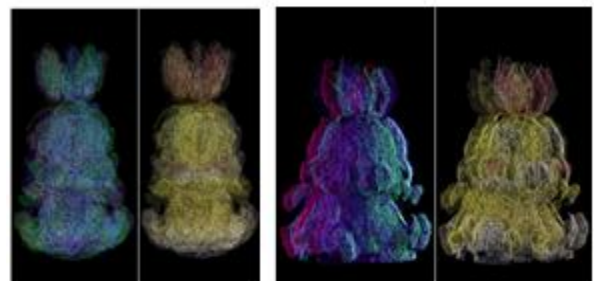
through filter ซึ่งใช้กำจัดข้อมูลที่ไม่เกี่ยวข้องในแนวแกน X แกน Y และแกน Z และใช้ Statically outlier removal filter เพื่อกำจัดข้อมูลบริเวณขอบของวัตถุซึ่งมีความไม่เสถียรอยู่ ตัวอย่างข้อมูลเมฆจุดที่ได้มาแสดงดังรูปที่ 3



รูปที่ 3 ภาพทั้ง 8 มุมมองหลังจากกำจัดพื้นหลังแล้ว

##### Point Cloud Data Rotation

ก่อนการเรียกใช้งาน ICP เราจำเป็นต้องจัดเรียงข้อมูลให้อยู่ในตำแหน่งที่สอดคล้องกับตำแหน่งมุมการเก็บภาพจริงของวัตถุ ดังนั้นเราจึงต้องหมุนข้อมูลเมฆจุดให้อยู่ในตำแหน่งที่เป็นมุมมองข้อมูลเอง ซึ่งก่อนที่เราจะหมุนข้อมูลเมฆจุดได้นั้นข้อมูลเมฆจุดต้องถูกย้ายให้ไปอยู่ในตำแหน่งของจุดกำเนิด (0,0,0) ซึ่งในที่นี้คือตำแหน่งกล้อง ดังนั้นเราจึงทำการเลื่อนข้อมูลเมฆจุดเข้าสู่จุด (0,0,0) โดยระยะแกน Z จะเป็นระยะห่างของกล้องถึงกลางวัตถุ เมื่อเลื่อนมาอยู่ในตำแหน่งจุด (0,0,0) แล้วเราก็ทำการหมุนวัตถุให้อยู่ในตำแหน่งที่ถูกต้องของได้ ดังแสดงผลในรูปที่ 4



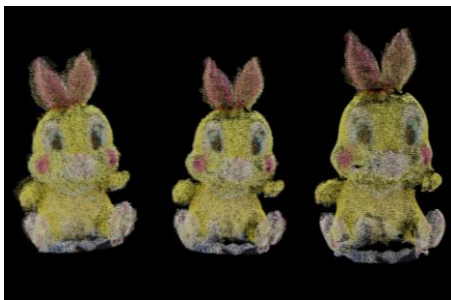
รูปที่ 4 ภาพทางซ้ายเป็นข้อมูลเมฆจุดก่อนหมุน ภาพทางขวาการ แสดงข้อมูลเมฆจุดหลังจากหมุนเรียบร้อยแล้ว

### Point Cloud Data Registration

หลังจากที่เราเตรียมข้อมูลเมฆจุดเสร็จเรียบร้อยแล้ว ข้อมูลที่ได้มามีลักษณะการเรียงที่ใกล้เคียงกับวัตถุจริง แต่ยังคงมีความคลาดเคลื่อนอยู่มาก ทำให้ต้องใช้ ICP เพื่อคำนวณหาความสัมพันธ์ของข้อมูลเมฆจุดแต่ละชุด เมื่อข้อมูลในแต่ละมุมมองเลื่อนไปสู่ตำแหน่งที่ถูกต้องมากขึ้น การหาความสัมพันธ์สามารถทำได้โดยใช้ข้อมูลเมฆจุดแต่ละคู่ที่อยู่ติดกันมาคำนวณ และทรานสฟอร์มข้อมูลแต่ละคู่เข้าด้วยกัน หลังจากนั้นเราจะทำการคำนวณและรวมข้อมูลเมฆจุดจากคู่ด้านหน้าวัตถุที่มุม 180 องศา รวมเข้ามาจนถึงด้านหน้าวัตถุ โดยการรวมเราจะแบ่งข้อมูลเป็นสองด้านคือ ซ้าย และขวาหลังจากที่ได้ชุดข้อมูลเมฆจุดด้านซ้ายและข้อมูลเมฆจุดด้านขวาของวัตถุแล้ว เราเรียกใช้ ICP อีกครั้งเพื่อรวมข้อมูลทั้งสองชุดเข้าด้วยกัน และสุดท้ายเราจะได้ข้อมูลเมฆจุดที่เกิดจาก 8 มุมมองรวมกัน

### Noise Removal

ข้อมูลเมฆจุดที่ได้จากการรวมแล้วจะมีข้อมูลจุดบางส่วนที่เป็นส่วนเกินออกมา เราจึงต้องทำการกำจัดจุดเหล่านี้โดยใช้ตัวกรอง Stochastic Outlier Removal Filter ก็จะทำให้ข้อมูลเมฆจุดที่ได้มีปริมาณจุดส่วนที่เกินออกมาลดน้อยลงไปในขั้นตอนสุดท้ายเราปรับพื้นผิวให้เรียบโดยการปรับเรียบ (Smoothing) ดังรูปที่ 5

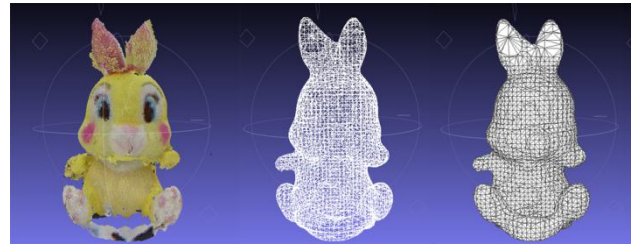


รูปที่ 5 รูปซ้ายเป็นภาพข้อมูลเมฆจุดหลังจากการใช้ ICP ภาพตรงกลางเป็นการใช้ Stochastic outlier removal filter ภาพขวาเป็นภาพหลังจากการทำการปรับเรียบ

### Surface Mesh Reconstruction

เมื่อเราได้ข้อมูลเมฆจุดที่ผ่านการ Registration แล้วเราจะนำข้อมูลเมฆจุดนั้นมาคำนวณหาพื้นผิวให้กับวัตถุ ในที่นี้เราใช้ขั้นตอนวิธี Poisson Surface Reconstruction แต่ก่อนอื่นเรา

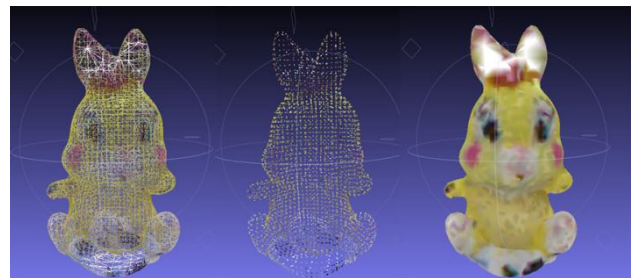
ต้องคำนวณหา Normal Vectors ของข้อมูลเมฆจุดก่อนโดยใช้ขั้นตอนวิธี K-Nearest Neighbor มาช่วยในการคำนวณหากรูปร่างที่อยู่ใกล้กันเพื่อนำไปใช้คำนวณหา Normal Vector ของแต่ละจุดได้ ผลจากการใช้ Poisson Surface Reconstruction คือ ข้อมูลจุดชุดใหม่ที่มีจำนวนน้อยลงจากเดิม และได้พื้นผิวที่สื่อความสัมพันธ์โครงข่ายระหว่างจุด ดังแสดงในรูปที่ 6



รูปที่ 6 ผลการปูพื้นผิวของวัตถุ

### Color Information Retrieval

หลังจากที่เราได้ปูพื้นผิวเสร็จเรียบร้อยแล้ว จะพบว่าพื้นผิวนั้นยังไม่มีข้อมูลสี ดังนั้นเราจะต้องนำค่าสีจากข้อมูลเมฆจุดเดิม ซึ่งมีค่า RGB ของแต่ละจุดอยู่ เข้าสู่พื้นผิวที่สร้างขึ้นใหม่ โดยขั้นตอนนี้เป็นขั้นตอนสุดท้ายในการทำงาน ในที่สุดเราจะได้แบบจำลองสามมิติของวัตถุที่มาจากกรสแกนเป็นที่เรียบร้อยแล้ว ดังแสดงตัวอย่างในรูปที่ 7



รูปที่ 7 โครงสร้างที่มีการโอนค่าสีเข้าสู่พื้นผิวที่สร้างขึ้นใหม่ และแสดงผลสุดท้ายของโมเดลที่ได้ในรูปแบบภาพสุดท้าย

### 4.3 เครื่องมือที่ใช้ในการพัฒนาระบบ

เครื่องมือด้านฮาร์ดแวร์ประกอบด้วยอุปกรณ์คิดค้น 1 เครื่อง เครื่องคอมพิวเตอร์ CPU Intel Core i5-2410M (2.30 GHz, 3 MB L3 Cache, up to 2.90 GHz) Graphic Chip AMD Radeon HD 6630M (1GB GDDR3) สำหรับการประมวลผล 1 เครื่อง ส่วนด้านซอฟต์แวร์ประกอบด้วยระบบปฏิบัติการ

Ubuntu เวอร์ชัน 11.10 PCL (Point Cloud Library) [6], ROS (Robot Operating System) [7] และ MeshLab [8]

#### 4.4 ข้อจำกัดของระบบ

ระบบยังคงมีข้อจำกัดดังต่อไปนี้ (1) ข้อมูลเมฆจุดที่เก็บมาต้องมีการเรียงข้อมูลเป็นลำดับที่ถูกต้อง (2) ยังไม่สามารถสแกนวัตถุที่มีขนาดเล็กมากเกินไปเนื่องจากข้อมูลไม่เพียงพอต่อการคำนวณ (3) การวางวัตถุควรวางอยู่กลางกล้อง และ (4) ไม่สามารถสแกนวัตถุที่มีลักษณะโปร่งใสได้เนื่องจากกล้อง Kinect ไม่สามารถจับข้อมูลความลึกในส่วนนั้นได้

#### 5. การทดสอบการใช้งาน

การทดสอบผลการทำงานของระบบแบ่งออกเป็น การทดสอบประสิทธิภาพในด้านความเร็วในการทำงานของระบบ และด้านความถูกต้องของขนาดและลักษณะผลลัพธ์แบบจำลองสามมิติที่ได้

##### 5.1 สภาพแวดล้อมในการทดสอบ

อุปกรณ์ที่ใช้ในการทดสอบระบบประกอบด้วย อุปกรณ์ Kinect แทนหมุนสำหรับวางวัตถุ สายวัด และวัตถุ 5 ชนิดที่มีขนาดและรูปร่างแตกต่างกันดังแสดงในรูปที่ 8

##### 5.2 ผลการทดสอบและการวิจารณ์ผล

รูปที่ 9 แสดงผลลัพธ์ของแบบจำลองสามมิติที่ได้จากระบบ การทดสอบความเร็วในการคำนวณ เราวัดความเร็วในการประมวลผลผลการทดลองดังกล่าวแสดงในตารางที่ 1



รูปที่ 8 ภาพวัตถุ 5 ชนิดที่ใช้ทดสอบ



รูปที่ 9 ผลที่ได้จากการสแกนวัตถุ 5 ชนิด

ตารางที่ 1 เปรียบเทียบเวลาการทำงาน

วัตถุ	Point cloud pre-process (points)	Point cloud post-process (points)	Total time (sec)
1	103,821	8,929	253
2	91,922	6,698	229
3	273,461	4,895	705
4	103,585	9,065	288
5	269,374	3,482	597

จากการทดสอบเราพบว่า เวลาในการทำงานแปรผันตามจำนวนของข้อมูลเมฆจุด หรือขนาดของวัตถุ ถ้าวัตถุมีขนาดใหญ่ ระบบก็ใช้เวลาในการคำนวณมาก การทดสอบความถูกต้องของขนาดและลักษณะของแบบจำลองผลลัพธ์ที่ได้จากการสแกน เราทำการทดสอบโดยใช้วัตถุทั้งหมด 5 ชนิดในการทดสอบเราใช้สายวัดเพื่อวัดขนาดจริงซึ่งเป็นขนาดสูงตามแนวแกน X แกน Y และแกน Z ของวัตถุ เพื่อดูความแตกต่างของวัตถุจริงกับวัตถุสามมิติผลที่ได้เป็นตามตารางที่ 2



ตารางที่ 2 ความคลาดเคลื่อนของแบบจำลองกับวัตถุจริง

วัตถุ	ขนาดจริง (cm) (X/Y/Z)	ขนาด โมเดล(cm) (X/Y/Z)	%ความคลาด เคลื่อน (X/Y/Z)
1	17.25/	16.54/	-4.12/
	9.20/	9.54/	3.50/
	12.45	12.76	2.49
2	11.00/	10.61/	-3.54/
	15.00/	15.19/	1.26/
	8.00	8.57	7.13
3	20.10/	20.23/	0.64/
	33.50/	33.22/	-0.84/
	13.80	16.74	21.30
4	9.90/	10.47/	5.76/
	16.30/	14.60/	-16.56/
	9.90	10.34	4.44
5	38.10/	39.67/	4.12/
	15.05/	14.8/	-1.26/
	12.10	13.99	15.62

จากการทดสอบ เราพบว่าวัตถุที่มีขนาดเล็กมีความคลาดเคลื่อนน้อยกว่าวัตถุที่มีขนาดใหญ่ ความคลาดเคลื่อนในแนวแกน Z มีความคลาดเคลื่อนสูงที่สุด ลักษณะโดยรวมของวัตถุที่สแกนมาได้ นั่นก็คือว่ามีรูปทรงสมบูรณ์ แต่รายละเอียดและส่วนของวัตถุที่มีความคมชัดหายไป เนื่องจากภาพตั้งต้นมีความละเอียดที่ต่ำและกระบวนการปรับผิวเรียบก็มีส่วนทำให้ข้อมูลขาดหายไป และขนาดของแบบจำลองยังคงมีความคลาดเคลื่อนอยู่บ้าง อย่างไรก็ตามโดยรวมแล้วสัดส่วนที่ได้ ยังคงมีความถูกต้องอยู่

## 6. บทสรุป

บทความนี้นำเสนอการพัฒนาระบบสแกนวัตถุสามมิติด้วยการใช้ Iterative Closest Point Algorithm (ICP) ในการสร้างแบบจำลองสามมิติของวัตถุ จากข้อมูลเมฆจุดที่ได้มาจากอุปกรณ์ Kinect จากผลการทดลองแบบผลลัพธ์จำลองสามมิติที่ได้ยังรักษารูปทรงโดยรวมของวัตถุได้ถูกต้อง แต่ว่ารายละเอียดขาดหายไป ส่วนความเร็วในการประมวลผลขึ้นอยู่กับขนาดข้อมูลเมฆจุดที่ได้รับมาจากอุปกรณ์ Kinect

## 6.1 แนวทางการพัฒนาต่อ

เราสามารถพัฒนาให้ระบบนี้สามารถสแกนวัตถุที่มีขนาดใหญ่ขึ้น และเราอาจเพิ่มจำนวนอุปกรณ์ Kinect เพื่อให้สามารถสแกนวัตถุในรอบเดียวซึ่งจะทำให้ความคลาดเคลื่อนลดลงได้นอกจากนี้ยังอาจเปลี่ยนจากการวางกล้องอยู่ที่ตำแหน่งคงที่มาเป็นขยับกล้องแทนการขยับวัตถุ แต่ทว่ารูปแบบการคำนวณจะแตกต่างออกไป และท้ายที่สุดคือการพัฒนาขั้นตอนวิธีให้การคำนวณมีประสิทธิภาพมากยิ่งขึ้น

## 7. กิตติกรรมประกาศ

ขอขอบคุณสถาบันวิจัยและพัฒนาแห่งมหาวิทยาลัยเกษตรศาสตร์ (ว-ท(ด) 91.53)

## 8. เอกสารอ้างอิง

- [1] Kinect [ออนไลน์], เข้าถึงได้  
<http://en.wikipedia.org/wiki/Kinect>.
- [2] Hearn, Donald and Baker, M. Pauline. Computer Graphics C version, Second Edition. 1996.
- [3] Point cloud [ออนไลน์], เข้าถึงได้  
[http://en.wikipedia.org/wiki/Point\\_cloud](http://en.wikipedia.org/wiki/Point_cloud)
- [4] A. Makadia, A. I. Petterson and K. Daniilidis. Fully Automatic Registration of 3D Point Clouds. Computer Vision and Pattern Recognition, 2006. CVPR '06. IEEE Conference on, Pages 1297 – 1304.
- [5] M.Kazhdan, M.Bolitho and H.Hoppe. Poisson Surface Reconstruction, The fourth Eurographics symposium on Geometry Processing, 2006. SGP '06. Pages 61 – 70.
- [6] R. Bogdan Rusu and S.Cousins. 3D is here: Point Cloud Library (PCL).International Conference on Robotics and Automation. 2011. ICRA '11.
- [7] ROS (Robot Operating System) [ออนไลน์], เข้าถึงได้  
<http://www.ros.org/wiki/>
- [8] MeshLab [ออนไลน์], เข้าถึงได้  
<http://meshlab.sourceforge.net/>