การปรับปรุง RGBDSLAM สำหรับสิ่งแวดล้อมแบบพลวัตร

ศุภมร ศรีบุญแก้ว[1*] และ พจน์ ตั้งงามจิตต์[2]
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี บางมด ทุ่งครุ กรุงเทพมหานคร 10140

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อปรับปรุงการระบุตำแหน่งและสร้างแผนที่ 3 มิติด้วยอัลกอริทึม RGBDSLAM ให้รองรับกับสภาพแวดล้อมที่มีวัตถุเคลื่อนที่ โดยมีแนวคิดที่จะกรองภาพที่มีวัตถุเคลื่อนที่ออกไปก่อนที่ภาพนั้นจะถูกนำเข้าสู่กระบวนการระบุตำแหน่งและสร้างแผนที่ โดยการกรองภาพนั้นจะใช้วิธีคำนวณการติดตามการเคลื่อนที่ของ Lucas และ Kanade ในการวิเคราะห์คุณลักษณะของการเคลื่อนที่ ซึ่งได้มีการแบ่งภาพที่จะนำมาสร้างแผนที่ออกเป็น 2 ชนิด คือ ภาพที่มีวัตถุเคลื่อนที่และภาพที่ไม่มีวัตถุเคลื่อนที่ ในวิธีการที่นำเสนอนี้ ภาพที่มีวัตถุเคลื่อนที่จะถูกกรองออกไปก่อนเข้าสู่กระบวนการ RGBDSLAM ทำให้ผลการทดลองที่ประเมินความถูกต้องและแม่นยำของการระบุตำแหน่งและสร้างแผนที่ 3 มิติจากการเปรียบเทียบค่าความผิดพลาดเฉลี่ย (RMSE) ดีขึ้นเมื่อเทียบกับวิธีการเดิมโดยผลที่ได้จากวิธีการเดิมมีค่าความผิดพลาดเฉลี่ย 0.1 ส่วนวิธีที่ได้พัฒนาขึ้นมีค่าความผิดพลาดเฉลี่ย 0.03

คำสำคัญ : การสร้างแผนที่และระบุตำแหน่ง / RGBDSLAM

* *Corresponding Author : c-club-zaa@hotmail.com*

1 *นักศึกษาปริญญาโท สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์*

2 *รองศาสตราจารย์ ภาควิชาวิศวกรรมระบบควบคุมและเครื่องมือวัด คณะวิศวกรรมศาสตร์*

Improved RGBDSLAM Algorithm for Handing Dynamic Environment

Supamon Sriboonkaew[1*] and Poj Tangamchit[2]

King Mongkut's University of Technology Thonburi, Bang Mod, Thung Khru, Bangkok 10140

Abstract

The objective of this work was to improve the RGBDSLAM algorithm, which is a 3D localization and mapping method, to be able to handle dynamic environments. The idea is to exclude scenes that contain dynamic moving objects from entering the map building process. Lucas-Kanade technique was used to calculate optical flows, which were used to differentiate the scenes with moving objects from the scenes without. By analyzing the optical flows, it was possible to separate scenes into two types: moving-object scenes and stationary scenes. The moving-object scenes would then be filtered out from entering the regular RGBDSLAM process. The experiments were performed by comparing 3D map building accuracy of the conventional RGBDSLAM with the improved one under the same environment. The results showed that the conventional method had an average root mean square error of 0.1, whereas that of the improved method was 0.03.

Keywords : Self Localization and Mapping / RGBDSLAM

* *Corresponding Author : c-club-zaa@hotmail.com*

1 *Master's Student, Electrical Engineering Program, Faculty of Engineering.*

2 *Associate Professor, Department of Control System and Instrumentation Engineering, Faculty of Engineering.*

1. บทนำ

การระบุตำแหน่งพร้อมกับการสร้างแผนที่ (Self Localization and Mapping - SLAM) คือกระบวนการที่หุ่นยนต์ทำการสร้างแผนที่ของสภาพแวดล้อมในขณะที่กำลังเคลื่อนที่และระบุตำแหน่งของตัวเองไปพร้อมๆ กัน การสร้างแผนที่มีความสำคัญสำหรับหุ่นยนต์เคลื่อนที่เป็นอย่างมาก เนื่องจากข้อมูลแผนที่จำเป็นต้องใช้ เพื่อให้หุ่นยนต์ทำภารกิจต่างๆ ตามสภาพแวดล้อมนั้น อัลกอริทึม SLAM ถูกคิดค้นแล้วพัฒนาตั้งแต่ปี พ.ศ. 2529 ในการสร้างแผนที่จากการหาค่าความแปรปรวนร่วมระหว่างพิกัดของจุดสังเกตด้วยวิธีการทำนายความน่าจะเป็นโดย Smith [1] ต่อมา Moutarlier และ Chatila [2] ได้นำเอา Extended Kalman Filter (EKF) มาใช้ในการแก้ปัญหาของ SLAM ในช่วงแรกซึ่งเป็นวิธีการประมาณการเคลื่อนที่ของตำแหน่งหุ่นยนต์เป็นสถานะของระบบซึ่งจะอธิบายอยู่ในรูปความแปรปรวนที่ได้ข้อมูลจากอุปกรณ์วัดค่าแล้วนำมาปรับแก้เป็นสถานะของระบบ EKF-SLAM ซึ่งมีประสิทธิภาพในการประมาณสถานะเป็น $O(n^3)$ โดยที่ n เป็นขนาดของแผนที่และมีความเร็วในการคำนวณสูง แต่อย่างไรก็ตาม ถ้าสิ่งแวดล้อมที่มีขนาดใหญ่อาจจะเกิดปัญหาด้านเวลาในการคำนวณได้ ต่อมา Walter ได้แก้ปัญหา SLAM โดยใช้วิธี Information-Based [3] ซึ่งอธิบายสถานะของระบบด้วยข้อมูลเวกเตอร์ และข้อมูลเมตริก ขั้นตอนการประมาณจะมีลักษณะคล้ายกับวิธี EKF-SLAM แต่วิธีการเจาะจงข้อมูล (Information-Based) มีข้อได้เปรียบที่ใช้เวลาในการทำนายและปรับปรุงสถานะของ SLAM ดีกว่าเป็น $O(1)$ และ $O(m)$ ตามลำดับ เมื่อ m เป็นขนาดของข้อมูลการวัด อย่างไรก็ตามสถานะของระบบในรูปแบบข้อมูล (Information form) อยู่ในรูปที่หุ่นยนต์ยังนำไปใช้ไม่ได้จึงต้องทำการกู้คืนสถานะ (State Recovery) เพื่อแปลงสถานะให้อยู่ในรูปความแปรปรวน โดยเวลารวมที่ใช้ยังคงเท่ากับ $O(n^3)$ ซึ่งใกล้เคียงกับ EKF-SLAM ต่อมา Thrun พบว่าค่าคุณลักษณะพื้นฐาน (Feature-based) ส่วนใหญ่ในรูปแบบข้อมูลเมตริกมีค่าเข้าใกล้ศูนย์จึงได้เสนอการแก้ปัญหา SLAM ด้วยวิธี Sparse Extended Information Filter (SEIF) [4] เพื่อลดเวลาคำนวณลงไปได้อีก หลังจากนั้นได้เริ่มหันมาวิจัยที่ใช้กล้องวีดีโอเพียงอย่างเดียว และใช้อัลกอริทึม MonoSLAM ซึ่งถูกพัฒนาขึ้นโดย Davison [5] เป็นแนวทางใหม่สำหรับการระบุตำแหน่งและแผนที่ด้วยกล้องวีดีโอ (Vision-based SLAM) ซึ่งได้เสนอการหาเส้นทางการเคลื่อนที่ในแกนสามมิติ ที่อาศัยข้อมูลภาพโดยใช้กล้องเพียงตัวเดียว เป็นงานที่ประสบความสำเร็จมากสำหรับ SLAM ในตอนนั้นที่ไม่อาศัยข้อมูลการเคลื่อนที่จากเซนเซอร์อื่น และทำงานได้ดีกับสภาพแวดล้อมที่ไม่แน่นอน แต่ทำงานได้ไม่ดีนักกับการเคลื่อนที่เปลี่ยนมุมมองเร็วจนเกินไปจะไม่สามารถสร้างแผนที่ได้ทัน เนื่องจากมีมุมมองของกล้องที่แคบ หลังจากนั้นไม่นานก็มีงานคล้ายกับ MonoSLAM อีกมากมาย เช่น FastSLAM [6] ได้นำเสนอขึ้นโดย Montemerlo หรือเครื่องบินไร้คนขับที่ใช้วิธีการกรองความไม่แน่นอน (Unscented Kalman Filter) [7] ในการแก้ปัญหา SLAM โดย Sunderhauf ในระยะหลังในปี พ.ศ. 2551 Klein ได้เสนอวิธี Parallel Tracking and Mapping (PTAM) [8] ซึ่งเป็นวิธีในการติดตามตำแหน่งของกล้อง (Tracking) และสร้างแผนที่ (Mapping) โดยการนำเอาอุปกรณ์วัดระยะ (Visual Odometry) มาใช้ร่วมกับ SLAM วิธีการนี้ถึงแม้จะสิ้นเปลืองประสิทธิภาพในการคำนวณ แต่ก็ได้คำตอบที่มีความแม่นยำสูง ให้ประสิทธิภาพแบบทันทีทันใด (Real-Time) ได้ หลังจากนั้นไม่นาน Newcombe ได้เสนอวิธีการสร้างแผนที่แบบหนาแน่น (Dense Tracking and Mapping (DTAM)) [9] จากจุดสังเกตจำนวนมากในการติดตามตำแหน่งของกล้องและสร้างแผนที่แบบหนาแน่น ซึ่งมีข้อดีก็คือการใช้ข้อมูลของภาพทั้งหมดแทนที่จะใช้ข้อมูลแค่บางจุดสังเกต ทำให้มีความแม่นยำสูง แต่ข้อเสียการสร้างแผนที่แบบหนาแน่นจะใช้เวลาในการคำนวณมาก ไม่สามารถสร้างแผนที่สิ่งแวดล้อมขนาดใหญ่ได้ ในปี พ.ศ. 2555 Filix Endres ได้ประยุกต์ใช้เซนเซอร์ RGB-D ของไมโครซอฟท์ Kinect ให้ข้อมูลภาพสี RGB และข้อมูลภาพความลึก (Depth) ได้พัฒนาให้ใช้ร่วมกับ SLAM ในอัลกอริทึม RGBDSLAM [10] ที่มีข้อดี คือ ใช้เวลาในการคำนวณน้อยให้ข้อมูลรายละเอียดของภาพสูง สามารถทำงานได้ดีในสิ่งแวดล้อมที่ไม่แน่นอน ข้อเสีย มีความแม่นยำน้อยในสิ่งแวดล้อมที่เคลื่อนที่ หลังจากนั้น RGBDSLAM เริ่มมีชื่อเสียงก็ถูกพัฒนาต่อไปเรื่อยๆ ปี พ.ศ. 2556 Donghwa Lee [11] ได้แก้ปัญหาการ

เคลื่อนที่ของสิ่งแวดล้อมเพียงเล็กน้อย (Low dynamic environment) แต่ยังมีปัญหาในเรื่องความเร็วในการประมวลผลที่ช้า

ดังนั้นผู้วิจัยจึงมีแนวคิดในการปรับปรุงอัลกอริทึม RGBDSLAM ในการแก้ปัญหาความแม่นยำในการระบุตำแหน่งที่มีวัตถุเคลื่อนที่ในสิ่งแวดล้อมที่มีอิทธิพลต่อการประมาณตำแหน่งการเคลื่อนที่ของกล้อง เพื่อเปรียบเทียบประสิทธิภาพการระบุตำแหน่งของหุ่นยนต์ โดยใช้วิธีการของ Lucas และ Kanade [12] ในการวิเคราะห์และคัดแยกเวกเตอร์ Optical flow โดยการศึกษาการเคลื่อนที่ของกล้องในทิศทางต่างๆ ในสภาพแวดล้อม 2 แบบคือ มีวัตถุเคลื่อนที่ (Dynamic) ดังตารางที่ 1 และไม่มีวัตถุเคลื่อนที่ (Static) ดังตารางที่ 2 โดยใช้ค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของเวกเตอร์ Optical flow เป็นตัวกำหนดค่าเริ่มต้น (Threshold) ว่าภาพดังกล่าวมีวัตถุเคลื่อนที่หรือไม่ ดังตารางที่ 3 แล้วเก็บผลความแม่นยำการระบุตำแหน่งและสร้างแผนที่ 3 มิติ ทั้ง 2 วิธี โดยเปรียบเทียบกับแผนที่มาตรฐาน (Ground truth) ในระยะทางการทดสอบ 180 เซนติเมตร ดังรูปที่ 5 ประเมินผลความแม่นยำการระบุตำแหน่งด้วยการคำนวณค่าผิดพลาดเฉลี่ย RMSE

2. วัตถุประสงค์งานวิจัย

เพื่อปรับปรุงประสิทธิภาพอัลกอริทึมการระบุตำแหน่งและสร้างแผนที่ 3 มิติ ด้วยกล้อง Kinect ซึ่งจะมุ่งเน้นให้รองรับสภาพแวดล้อมแบบพลวัตร โดยการศึกษาการเคลื่อนที่ของกล้องและวัตถุที่มีผลต่อประสิทธิภาพในการระบุตำแหน่งและการสร้างแผนที่จากการคำนวณทางคณิตศาสตร์ เพื่อคัดแยกภาพออกเป็น 2 ลักษณะ ระหว่างมีวัตถุเคลื่อนที่กับไม่มีวัตถุเคลื่อนที่ ผลที่ได้จากงานวิจัยสามารถลดค่าความผิดพลาดในการระบุตำแหน่งและการสร้างแผนที่ให้มีความถูกต้องมากขึ้น

3. ทฤษฎีที่เกี่ยวข้อง

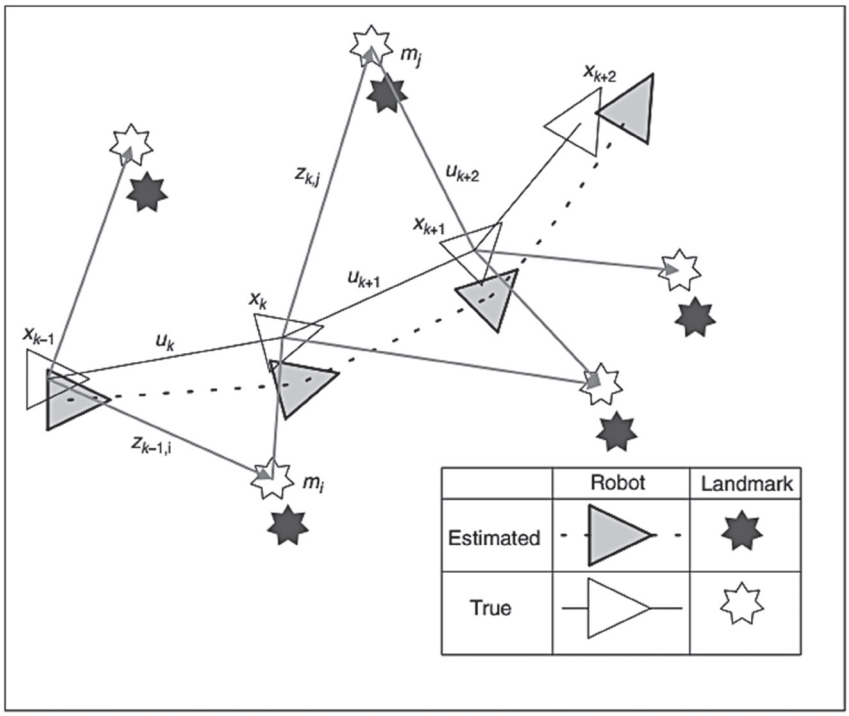
การระบุตำแหน่งและสร้างแผนที่ Simultaneous Localization And Mapping (SLAM) [13] เป็นวิธีการที่หุ่นยนต์สร้างแผนที่ 3 มิติ ของสิ่งแวดล้อมขณะเคลื่อนที่และระบุตำแหน่งของตัวมันเองซึ่งต้องอาศัยอุปกรณ์วัดค่าที่ติดตั้งอยู่บนตัวหุ่นยนต์ ในงานวิจัยนี้เลือกใช้กล้อง Kinect ข้อดีของกล้อง คือ สามารถให้ภาพสีที่มีข้อมูลความลึก RGB-D ส่วนประกอบหลักของกล้องประกอบด้วยกล้องสีและกล้องอินฟาเรดที่ใช้บอกระยะทางของวัตถุ ซึ่งมีความแม่นยำและสะดวกในการสร้างแผนที่ RGBDSLAM มากกว่าการใช้กล้องวีดีโอเพียงอย่างเดียวที่ต้องมีข้อมูลภาพมากทำให้ประมวลผลนาน เนื่องจากต้องใช้ภาพสิ่งแวดล้อมในมุมมองต่างกันเพื่อหาระยะของสิ่งแวดล้อมทำให้การเคลื่อนที่ไม่อิสระแต่ RGBDSLAM จะทำงานได้ไม่ดีในสภาพแวดล้อมที่มีการเคลื่อนที่ของวัตถุ การเคลื่อนที่เร็วเกินไปและสิ่งแวดล้อมมีขนาดใหญ่เกินไป เพื่อเพิ่มประสิทธิภาพให้กับ RGBDSLAM งานวิจัยนี้จึงตรวจจับการเคลื่อนที่ของวัตถุก่อนที่จะเข้าสู่กระบวนการของ SLAM หลักการแยกการเคลื่อนที่ โดยใช้ขนาดของมุมและความยาวของเวกเตอร์ Optical flow เพื่อคำนวณหาค่าความคลาดเคลื่อน (Root Mean Square Error : RMSE) อธิบายได้ด้วยค่าเฉลี่ย (Mean) และค่าเบี่ยงเบนมาตรฐาน (SD) ขั้นตอนของ SLAM แบ่งออกเป็น 3 ส่วน การตรวจหาจุดสังเกต (3.1), การประมาณการเคลื่อนที่ (3.2) การปรับแก้ตำแหน่งแผนที่ (3.3) ในส่วน (3.4) คือการคัดแยกวัตถุเคลื่อนที่โดยใช้เวกเตอร์ Optical flow

3.1 การตรวจหาจุดสังเกตของภาพ (Feature-Matching)

การตรวจจับจุดที่มีคุณลักษณะเด่นของภาพบริเวณที่มีการเปลี่ยนแปลงของค่าความเข้มของสีสูงโดยวิธีของ Harris Detector [14] แล้วนำมาเปรียบเทียบหาความคล้ายคลึงกันระหว่างภาพสองภาพของจุดสังเกตโดยวิธี SIFT [15], SURF [16] ที่มีคุณลักษณะใกล้เคียงกันระหว่างภาพปัจจุบันกับภาพก่อน ซึ่งแม้ว่าภาพจะมีขนาดต่างกันหรือมีการหมุนไปก็สามารถตรวจหาคุณลักษณะเดิมได้ ต่อมา Angeli [17] ได้นำเสนอวิธีตรวจจับการวนซ้ำ (Detect loop closure) คือ การนำเอาคุณลักษณะที่เก็บไว้ในแผนที่มาหาความสอดคล้องในภาพ โดยมีทั้งหมดสามแบบ คือ ภาพต่อภาพ ซึ่งเป็นการหาความสัมพันธ์ระหว่างภาพโดยตรง แผนที่ต่อแผนที่ ซึ่งเป็นการหาความสัมพันธ์โดยพิจารณาสร้างแผนที่เฉพาะส่วน และภาพต่อแผนที่ ซึ่งเป็นการฉายภาพลงบนข้อมูลของแผนที่เพื่อหาความสัมพันธ์

3.2 การประมาณการเคลื่อนที่ของหุ่นยนต์ (Pose Estimate)

การประมาณการเปลี่ยนแปลงสถานะของหุ่นยนต์โดยใช้ข้อมูลจากภาพในอดีตเปรียบเทียบกับภาพปัจจุบันแล้วนำไปประมาณความน่าจะเป็นข้อมูลการเคลื่อนที่ จะใช้วิธีการกระจายความน่าจะเป็นของตำแหน่งหุ่นยนต์และแผนที่ กำหนดค่าการวัด (Measurement) จากอุปกรณ์วัดค่า (Sensor)



รูปที่ 1 การประมาณการเคลื่อนที่โดยการประมาณความน่าจะเป็นของตำแหน่งหุ่นยนต์เทียบกับแผนที่จริง จากอุปกรณ์วัดค่า (Sensor)

การประมาณการเคลื่อนที่ คือ การอธิบายแผนที่ของ SLAM ด้วยจุดสังเกต Feature-based ซึ่งจุดสังเกตแทนด้วยตำแหน่งของวัตถุ จะอธิบายความน่าจะเป็นได้ด้วย ค่าเฉลี่ย และค่าความแปรปรวนซึ่งแบ่งได้เป็น 3 ขั้นตอน

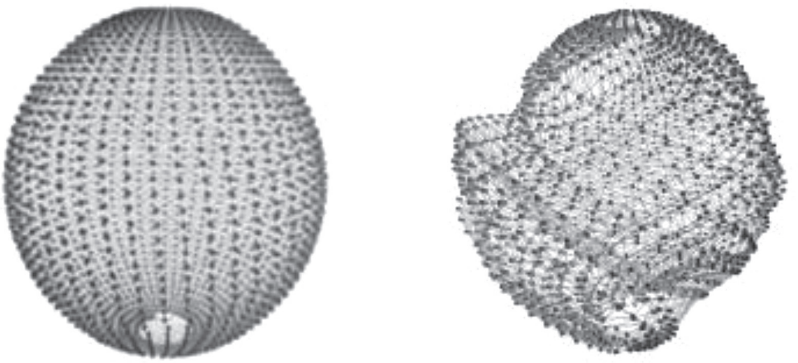
1. การเพิ่มสถานะเข้าไปในระบบ (State Augmentation) ซึ่งทำงานเมื่อหุ่นยนต์เคลื่อนที่ไปข้างหน้าแล้วเพิ่มสถานะเข้าไปในระบบกระจายความน่าจะเป็นใหม่

2. การประมาณค่า (Prediction) การประมาณค่าสถานะของการเคลื่อนที่ของระบบ ทำนายจุดสังเกตและความแปรปรวนที่เวลา t

3. ปรับปรุงสถานะ (Time-update) หลังจากทำนายสถานะแล้วได้ค่าการวัดจุดสังเกต ที่ได้จากการคำนวณค่าความแปรปรวน เพื่อที่จะปรับปรุงสถานะของกล้อง

3.3 การปรับแก้ตำแหน่งและแผนที่ (Graph Optimization)

การประมาณสถานะของหุ่นยนต์และแผนที่ทั้งหมดตั้งแต่เวลาเริ่มต้นถึงเวลาปัจจุบัน คือ การเทียบเคียงส่วนของแผนที่ให้ตรงกัน (Registration) รวบรวมชุดข้อมูล (Dataset) หลายๆ ชิ้นให้กลายเป็นชิ้นเดียว (Global model) โดยใช้การหาค่าความเหมาะสมของกราฟด้วยวิธี Network optimizer [18]



รูปที่ 2 การหาค่าความเหมาะสมที่สุดของกราฟ

เนื่องจากตำแหน่งของหุ่นยนต์จะเปลี่ยนแปลงไปหลังจากที่หุ่นยนต์เคลื่อนที่แผนที่จะเพิ่มจุดสังเกตใหม่ จึงต้องปรับแก้ตำแหน่งเพื่อให้ค่าความคลาดเคลื่อน (Alignment error) น้อยที่สุดโดยใช้วิธี จุดถึงจุด (Iterative Closet Point : ICP) [19] จะลดระยะทางจุดที่สัมพันธ์กันระหว่างจุด p_i กับ p_j ของชุดข้อมูล M_i ได้มีการแปลงในสามมิติและหมุนในแนวตั้ง แนวนอน และแนวทแยง $(x,y,z,roll.pitch.yaw)^T$

$$E_{j,i}(T*) = \sum_{k=1}^{Mi} \left[(T*)(p_i^k) - p_j^{coor(k)}.n_j^{coor(k)} \right]^2 \quad \textbf{(1)}$$

ซึ่งปัญหานี้จะสามารถแก้ไขได้ด้วยการคำนวณระยะห่างความผิดพลาดจากจุดใน p_i ถึงระนาบสัมผัส (Tangent plane) ของจุดที่สอดคล้องกัน (Correspondence) ใน p_j แบบจุดต่อระนาบ โดยที่ $corr(k)$ คือ ดัชนีของจุดใน p_j ที่มีความสอดคล้องกับจุดดัชนีที่ ใน p_i ส่วน $n_j^{corr(k)}$ คือ เวกเตอร์ปกติของระนาบสัมผัส ณ จุด $p_j^{corr(k)}$ จากนั้นจึงทำการหาค่าการแปลง $T*$ ที่ทำให้ฟังก์ชันความผิดพลาดนี้มีค่าน้อยที่สุด หรือใช้วิธีการสุ่มความเห็นพร้อม Random Sample Consensus (RANSAC) [20] ซึ่งเป็นวิธีการวนซ้ำเพื่อประมาณค่าโมเดลทางคณิตศาสตร์ที่ประกอบด้วยค่าปกติ (Inliers) และค่าผิดปกติ (Outliers) โดยจะสุ่มเลือกข้อมูลจำนวนหนึ่งแล้วสมมติให้เป็นค่าปกติจากกลุ่มจุดเหล่านี้จะได้โมเดลมาหนึ่งอันแล้วจะสุ่มทำไปเรื่อยๆ จนกว่าจะได้การจำลองที่ยอมรับได้ (Refined model) เช่น บริเวณพื้นห้องและกำแพง ในแต่ละครั้งก็จะได้โมเดลที่ถูกทิ้งเนื่องจากมีจุดที่มีค่าปกติน้อยเกินไป ข้อดี คือ มีความแม่นยำสูงถึงแม้จะมีค่าผิดปกติมาก ข้อเสีย การกำหนดจำนวนครั้งที่จะวนอาจจะไม่ได้คำตอบที่เหมาะสม

3.4 การคัดแยกวัตถุเคลื่อนที่โดยใช้ Optical flow

งานวิจัยนี้ใช้วิธีการคำนวณ Optical flow ของ Lucas และ Kanade ซึ่งเป็นวิธีติดตามการเคลื่อนไหวทั้งภาพโดยใช้อัลกอริทึมพีระมิด การติดตามนั้นจะเริ่มต้นจากชั้นที่สูงสุดของพีระมิดของภาพ และทำงานลงมาเรื่อยๆ จนถึงชั้นที่อยู่ต่ำสุด กำหนดให้ $I(x,y,t)$ คือ ทิศทางระดับความเข้มของสีระหว่างรูปภาพที่เปลี่ยนไปตามเวลา t ที่จุด x และ y สามารถคำนวณหาขนาดของความเร็วจากสมการความชัน $\Delta x,\Delta y$ คือระยะเวลาที่ล่วงเลยไปจะได้ค่า I คืออนุพันธ์ระหว่างรูปภาพที่เปลี่ยนไปตามเวลา ตามสมการที่ (2)

$$I(x,y,t) = I(x+\Delta x,y+\Delta y,t+\Delta t) \quad \textbf{(2)}$$

ถ้าพิกเซล (x,y) ที่เวลา t จะมีความเข้มเป็น $I(x,y,t)$ และพิกเซลเดียวกันนี้ในภาพถัดไปจากนั้นการเคลื่อนที่ด้วยเวลา dt ไปเป็นระยะทาง (dx,dy) จะมีความเข้มเท่าเดิมดังสมการที่ (3)

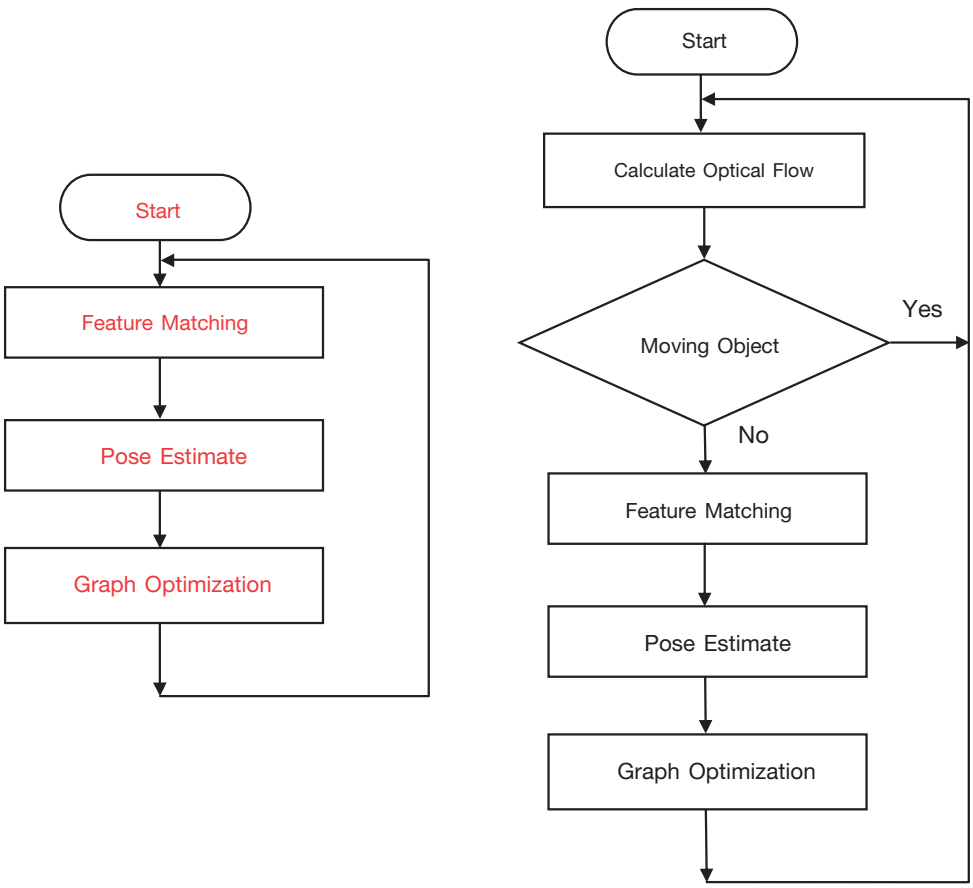
$$I(x+dx,y+dy,t+dt) = I(x,y,t) \quad \textbf{(3)}$$

4. วิธีการดำเนินการวิจัย

ออกแบบระบบโดยศึกษาการเคลื่อนที่ของกล้อง เว็บแคม Logitech Webcam Pro 9000 และการเคลื่อนของวัตถุในแนวแกน xyz เพื่อวิเคราะห์ความแตกต่างระหว่างการเคลื่อนที่ทั้ง 2 แบบ ด้วยวิธีของ Lucas และ Kanade ซึ่งคำนวณค่าเบี่ยงเบนกับค่าเฉลี่ยของเวกเตอร์ Optical flow ถ้าตรวจจับได้ว่ามีวัตถุเคลื่อนที่จะตัดภาพนั้นทิ้งไป หลังจากนั้นนำวิธีการคำนวณการคัดแยกเวกเตอร์ Optical flow เข้าไปรวมกับอัลกอริทึม RGBDSLAM ที่ใช้ระบบปฏิบัติการลีนุกซ์ Ubuntu เวอร์ชั่น Hydro 64 บิต ที่ติดตั้งระบบปฏิบัติการ ROS (Robot Operating System) เพื่อทำหน้าที่ควบคุมและติดต่อกับกล้อง Kinect ทดสอบโดยติดตั้งกล้องและคอมพิวเตอร์ไว้กับรถเข็นขนาด 4 ล้อ ใช้วิธีการประเมินความถูกต้องของแผนที่เทียบกับแผนที่มาตรฐาน (Ground truth) และสร้างแผนที่ภายในอาคารเรียน โดยผลที่ได้ทั้งหมดจะถูกเปรียบเทียบกันระหว่าง 2 วิธี (วิธีการเดิมและวิธีการที่นำเสนอ)

4.1 หลักการทำงานของส่วนคัดแยกวัตถุเคลื่อนที่

การคัดแยกการเคลื่อนที่ของวัตถุ (Moving Object) โดยใช้ขนาดของมุมและความยาวของเวกเตอร์ Optical flow เป็นตัวคัดแยกวัตถุเคลื่อนที่ เมื่อโปรแกรมเริ่มทำงาน โปรแกรมจะคำนวณมุมเฉลี่ยและความยาวเฉลี่ยของแต่ละภาพ หลังจากนั้นคำนวณค่าเบี่ยงเบนขนาดของมุมและขนาดของความยาว แล้วนำค่าที่คำนวณได้มาวิเคราะห์ว่าเฟรมดังกล่าวอยู่ในระยะค่าเริ่มต้น (Threshold) กำหนดไว้หรือไม่ ในกรณีที่ตรวจจับได้มีการเคลื่อนที่ของวัตถุจะตัดภาพเฟรมนั้นออกไปไม่เข้ากระบวนการ SLAM



รูปที่ 3 Flow chart ขั้นตอนการทำงานของระบบเดิม (ซ้าย) เพิ่มการคัดแยกวัตถุเคลื่อนที่ (ขวา)

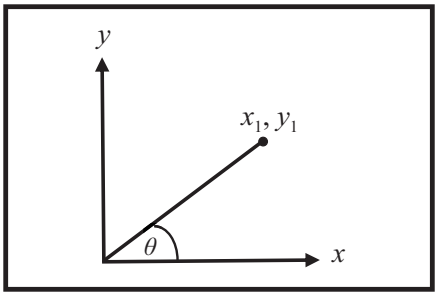
4.2 คำนวณขนาดมุมและความยาวเวกเตอร์ Optical flow

โดยคำนวณแบบเฟรมต่อเฟรมที่พิจารณาการเคลื่อนที่ใน 2 มิติ ในแต่ละเฟรมมุมของการเคลื่อนที่ในแต่ละจุด จะแตกต่างกันขึ้นอยู่กับขนาดและทิศทางของการเคลื่อนที่สังเกตได้จากเวกเตอร์เทียบกับแกน (x,y) โดยสามารถหามุมได้จาก

$$\theta = arctan^{-1}(\Delta x,\Delta y)^* \ 180/\pi \ \text{องศา} \qquad \textbf{(5)}$$

Δy = ค่าพิกัดแกน y

Δx = ค่าพิกัดแกน x



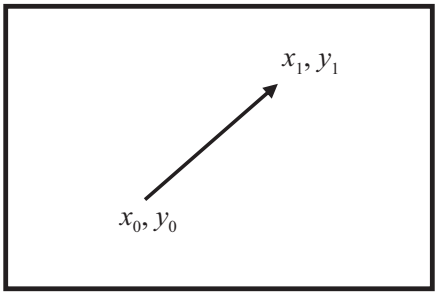
รูปที่ 4 ขนาดมุมเวกเตอร์ Optical flow

คำนวณขนาดความยาวเวกเตอร์ โดยใช้สมการเส้นตรงหาความยาวเวกเตอร์แต่ละจุดระหว่างเฟรมก่อนหน้าและเฟรมปัจจุบัน

$$\text{Length} = \Delta y^2 + \Delta x^2 \text{ หน่วย} \qquad (6)$$

Δy = ค่าพิกัดแกน y

Δx = ค่าพิกัดแกน x



รูปที่ 5 ขนาดความยาวเวกเตอร์ Optical flow

4.3 คำนวณพฤติกรรมการเคลื่อนที่เวกเตอร์ Optical flow

การระบุสถานะของการเคลื่อนที่จากค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน ที่ได้จากการคำนวณขนาดของมุมและขนาดของความยาวของเวกเตอร์ Optical flow เป็นตัวคัดแยกการเคลื่อนที่

$$s^2 = \frac{\sum(x-\bar{x})^2}{n-1} \qquad \textbf{(7)}$$

s^2 = ค่าความแปรปรวน

x = ข้อมูล

$\bar{x}$ = ค่าเฉลี่ยข้อมูล

n = จำนวนข้อมูลทั้งหมด

4.4 การคัดแยกเวกเตอร์ Optical flow

เราใช้ความละเอียดของเฟรมภาพที่ขนาด 320x420 พิกเซล เก็บผลการทดลองการเคลื่อนที่ของกล้องและวัตถุเพื่อนำไปเป็นข้อสรุปว่าเฟรมดังกล่าวนั้นมีการเคลื่อนที่ของวัตถุ (Dynamic) หรือไม่มีการเคลื่อนที่ (Static) ของวัตถุโดยใช้ค่าเบี่ยงเบนมาตรฐาน ค่าเฉลี่ยของมุมและความยาวเวกเตอร์ Optical flow สำหรับการเคลื่อนที่ของวัตถุได้กำหนดให้ใช้บุคคลเคลื่อนที่ในแกน (xyz) ผ่านทางหน้ากล้องวีดีโอไปในทิศทาง ซ้าย-ขวา และเข้า-ออก ตามลำดับ ในขณะเดียวกันนั้นให้กดหยุดโปรแกรมชั่วคราว (Pause) เพื่อบันทึกผลขนาดของมุมและความยาวของเวกเตอร์ ดังตารางที่ 1 ต่อมาเป็นการเคลื่อนที่ของกล้องหรือเฟรมที่ไม่มีวัตถุเคลื่อนที่ กำหนดให้ใช้บุคคลเข็นรถเข็นที่บรรทุกกล้องวีดีโอเคลื่อนที่ไปในแกน (xyz) ผ่านสิ่งแวดล้อมต่างๆ ในทิศทาง ซ้าย-ขวา และเข้า-ออก ตามลำดับ ในขณะเดียวกันนั้นให้กดหยุดโปรแกรมชั่วคราวเพื่อบันทึกผลลงใน ตารางที่ 2 การเก็บผลการทดลองทั้งหมด 7 ชุด แต่ละชุดทดลองประกอบไปด้วยการเคลื่อนที่ของกล้องและวัตถุทั้งหมด 3 ครั้ง แล้วเลือกข้อมูลมาแสดงผลจำนวน 1 ชุด ดังนี้

ตารางที่ 1 ผลการคำนวณการเคลื่อนที่ของวัตถุ

หมุนแกน	มุมเฉลี่ย (องศา)			ค่าเบี่ยงเบนมาตรฐานมุม (องศา)			ความยาวเฉลี่ย (หน่วย)			ค่าเบี่ยงเบนมาตรฐานความยาว (หน่วย)		
ระยะทาง	**น้อยกว่า 1 เมตร**											
ครั้งที่	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
X (ล่าง - บน)	19.25	17.13	2.47	2.01	8.95	3.56	55.10	61.15	15.28	1.70	15.25	3.82
X (บน - ล่าง)	7.82	19.13	17.31	1.95	4.74	4.32	7.82	19.13	17.31	1.95	4.78	4.32
Y (ซ้าย - ขวา)	24.49	43.36	43.19	7.50	0.84	3.60	12.62	7.96	2.19	0.85	3.98	1.09
Y (ขวา - ซ้าย)	21.11	24.34	45.91	4.97	4.63	3.68	29.45	5.47	34.06	1.94	1.36	2.21
Z (เคลื่อนที่ไป)	5.93	9.42	4.15	0.48	0.64	0.30	42.26	35.75	41.41	0.23	0.20	0.21
Z (เคลื่อนที่กลับ)	9.78	5.93	8.90	0.64	0.44	0.60	45.11	49.05	48.18	0.25	0.16	0.21
ระยะทาง	**มากกว่า 5 เมตร**											
X (ล่าง - บน)	12.44	9.92	10.29	0.91	0.58	0.61	9.92	39.45	20.23	0.58	0.20	0.11
X (บน - ล่าง)	9.40	4.36	9.05	0.61	0.36	2.21	80.69	34.85	42.01	0.81	0.25	0.43
Y (ซ้าย - ขวา)	10.72	7.75	11.15	1.26	0.26	0.40	25.20	74.86	13.22	0.15	0.56	0.19
Y (ขวา - ซ้าย)	5.54	4.87	9.95	0.34	0.33	1.47	28.25	34.09	41.09	0.11	0.25	0.22
Z (เคลื่อนที่ไป)	1.03	2.46	5.38	0.04	0.11	0.25	34.29	44.94	30.61	0.19	0.17	0.10
Z (เคลื่อนที่กลับ)	9.61	1.32	3.89	0.35	0.06	0.18	44.80	35.75	50.05	0.16	0.16	0.17

ตารางที่ 2 ผลการคำนวณการเคลื่อนที่กล้อง

หมุนแกน / ครั้งที่	มุมเฉลี่ย (องศา)			ค่าเบี่ยงเบนมาตรฐาน มุม (องศา)			ความยาวเฉลี่ย (หน่วย)			ค่าเบี่ยงเบนมาตรฐาน ความยาว (หน่วย)		
	1	2	3	1	2	3	1	2	3	1	2	3
X (ล่าง - บน)	12.39	9.45	8.19	0.87	0.58	0.08	8.69	49.68	18.09	0.03	0.19	0.11
X (บน - ล่าง)	9.45	5.24	7.37	0.59	0.51	0.46	86.75	57.02	38.15	0.47	0.34	0.22
Y (ซ้าย - ขวา)	11.19	16.02	9.54	0.71	0.20	0.58	34.05	90.38	15.26	0.13	0.37	0.06
Y (ขวา - ซ้าย)	5.68	4.91	8.28	0.28	0.33	0.41	23.12	38.15	47.48	0.12	0.22	0.19
Z (เคลื่อนที่ไป)	0.13	0.15	0.16	0.01	0.01	0.41	33.94	25.96	23.80	0.03	0.16	0.07
Z (เคลื่อนที่กลับ)	0.16	0.15	0.25	0.01	0.00	0.02	35.34	15.66	10.49	0.13	0.16	0.08

4.5 สรุปผล

หลังจากนั้นเราเห็นถึงความแตกต่างของตัวเลขผลการทดลองระหว่างเฟรมที่มีวัตถุเคลื่อนที่กับไม่มีวัตถุเคลื่อนที่ด้วยค่าเฉลี่ยของการเก็บผลทั้ง 7 ชุด ผู้วิจัยพบว่าประมาณ 80% เฟรมที่มีการเคลื่อนที่ของวัตถุที่คำนวณค่าเบี่ยงเบนมาตรฐานขนาดของมุมเวกเตอร์ Optical flow มีค่ามากกว่า เฟรมที่ไม่มีการเคลื่อนที่ของวัตถุ จึงได้ใช้ค่าความแตกต่างของเวกเตอร์เฉลี่ยที่ได้มาเป็นตัวกำหนดค่าเริ่มต้นในการคัดแยกวัตถุ ตามตารางที่ 3

ตารางที่ 3 การกำหนดค่าเริ่มต้นเพื่อคัดกรองภาพ

การเคลื่อนที่	วัตถุเคลื่อนที่		กล้องเคลื่อนที่		ไม่มีการเคลื่อนที่	
ขนาดผลรวมเฉลี่ยเวกเตอร์	ค่าเฉลี่ย มุม (องศา)	ค่าเบี่ยงเบน มุม (องศา)	ค่าเฉลี่ย มุม (องศา)	ค่าเบี่ยงเบน มุม (องศา)	ค่าเฉลี่ย มุม (องศา)	ค่าเบี่ยงเบน มุม (องศา)
แกน X	> 14.75	>3.67	>10.25	<0.60	< 0.01	< 0.001
แกน Y	> 30.45	>4.42	>12.25	<0.50	< 0.01	< 0.001
แกน Z	> 6.5	>0.47	>0.15	<0.01	< 0.01	< 0.001

ดังนั้นจากตารางที่ 3.4 จึงกำหนดให้ เฟรมที่มีค่าเบี่ยงเบนมุมที่เป็นไปได้มากที่สุดว่าเป็นการเคลื่อนที่ของวัตถุมากกว่า 3.67 องศา คือ ภาพที่มีการเคลื่อนที่ของวัตถุจะถูกลบทิ้ง และกำหนดให้เฟรมที่มีค่าเบี่ยงเบนมุมที่เป็นไปได้มากที่สุดว่าไม่มีการเคลื่อนที่ของวัตถุน้อยกว่า 0.01 องศา นำภาพเฟรมดังกล่าวเข้าสู่กระบวนการ SLAM ต่อไป ดังรูปที่ 3 (ขวา)

5. ผลการทดลอง

ส่วนของการสร้างแผนที่ 3 มิติ จากการศึกษาการเคลื่อนที่ของกล้องและการเคลื่อนที่ของวัตถุ เพื่อแก้ไขปัญหาความผิดพลาดจากการระบุตำแหน่งและสร้างแผนที่โดยการตรวจจับการเคลื่อนที่ของวัตถุ

ส่วนนี้ผู้ทดลองได้ตั้งสมมติฐานขึ้นมาว่าในกรณีที่มีวัตถุเคลื่อนที่ให้ตัดภาพเฟรมนั้นออกไปเพื่อให้แผนที่นั้นออกมา

เหมือนจริงมากที่สุด ดังนั้นผู้ทดลองจึงได้ทำการศึกษาอัลกอริทึม RGBDSLAM ในส่วนที่แสดงผลของแผนที่เพื่อวิเคราะห์การจับภาพเพื่อมาสร้างแผนที่ซึ่งความแม่นยำของการทำแผนที่นั้นต้องอาศัยสิ่งแวดล้อมที่อยู่กับที่

ในกรณีที่สิ่งแวดล้อมมีวัตถุเคลื่อนที่ไปในทิศทางเดียวกันกับกล้องหรือสวนทางกันด้วยความเร็วต่างๆ ขณะทำการระบุตำแหน่งและสร้างแผนที่ให้กรองภาพที่มีความแปรปรวนสูง ซึ่งมีความเสี่ยงทำให้การประมาณการเคลื่อนที่ผิดพลาดไป โดยเรากำหนดให้การประเมินผลความแม่นยำของ RGBDSLAM ประกอบไปด้วย 3 ขั้นตอน คือ 1. การระบุตำแหน่งด้วยกล้องเปรียบเทียบกับการระบุตำแหน่งด้วยแผนที่มาตรฐาน 2. ทดสอบประสิทธิภาพการตรวจจับวัตถุเคลื่อนที่ 3. สร้างแผนที่สิ่งแวดล้อม 3 มิติ ภายในอาคาร

5.1 ผลการทดสอบการระบุตำแหน่ง RGBDSLAM

การทดสอบความแม่นยำการระบุตำแหน่งและสร้างแผนที่ 3 มิติ ด้วยอัลกอริทึม RGBDSLAM โดยเปรียบเทียบกับแผนที่มาตรฐาน (Ground truth) ในระยะทางการทดสอบ 180 เซนติเมตร ได้แบ่งการทดสอบออกเป็น 6 ช่วง แต่ละช่วงเป็นระยะทาง 30 เซนติเมตร การเก็บผลการทดลองแบ่งออกเป็น 4 แบบ 1. การระบุตำแหน่งและสร้างแผนที่ด้วยวิธีการเดิมที่ไม่มีวัตถุเคลื่อนที่ 2. การระบุตำแหน่งและสร้างแผนที่ด้วยวิธีการเดิมที่มีวัตถุเคลื่อนที่ 3. การระบุตำแหน่งและสร้างแผนที่ด้วยวิธีการที่นำเสนอที่มีวัตถุเคลื่อนที่ 4. การระบุตำแหน่งและสร้างแผนที่ด้วยวิธีการที่นำเสนอที่ไม่มีวัตถุเคลื่อนที่ การทดลองทั้ง 4 แบบจะเข็นรถเป็นเส้นตรงตามระยะทางที่กำหนดไว้ วิธีการเข็นรถต้องใช้ความเร็วในการเคลื่อนที่อย่างสม่ำเสมอประมาณ 1 ฟุต/วินาที ในส่วนการจำลองสถานการณ์ที่มีวัตถุเคลื่อนที่ได้อธิบายไว้ในหัวข้อ 5.2 ผลการทดลองทั้ง 4 แบบ

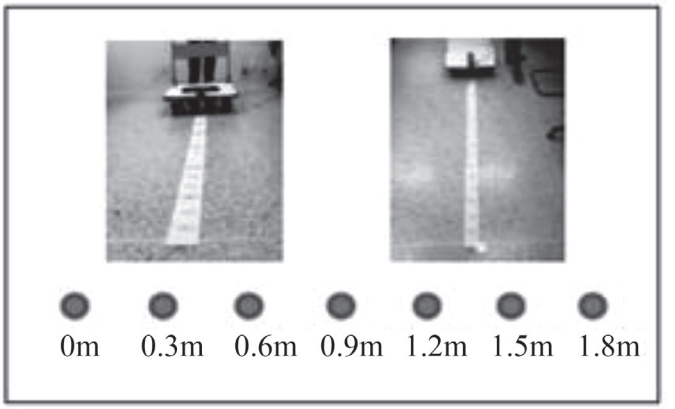
1. พบว่าเมื่อพิจารณาระยะทางการระบุตำแหน่งด้วยวิธีการเดิมที่ไม่มีวัตถุเคลื่อนที่มีค่าความผิดพลาดเฉลี่ย RMSE น้อยกว่าวิธีการเดิมที่มีวัตถุเคลื่อนที่ 0.079

2. พบว่าเมื่อพิจารณาระยะทางการระบุตำแหน่งด้วยวิธีการเดิมที่มีวัตถุเคลื่อนที่ อัลกอริทึม RGBDSLAM จะระบุตำแหน่งผิดพลาดไปจากระยะทางจริงมากขึ้น เนื่องจากเฟรมที่มีวัตถุเคลื่อนที่นั้นมีจุดสังเกตที่ซ้ำกันในเฟรมก่อนหน้าและเฟรมถัดไป เมื่อพิจารณาผลการสร้างแผนที่ ดังรูปที่ 9 (บน) ภาพที่มีการเคลื่อนที่จะซ้อนทับกัน

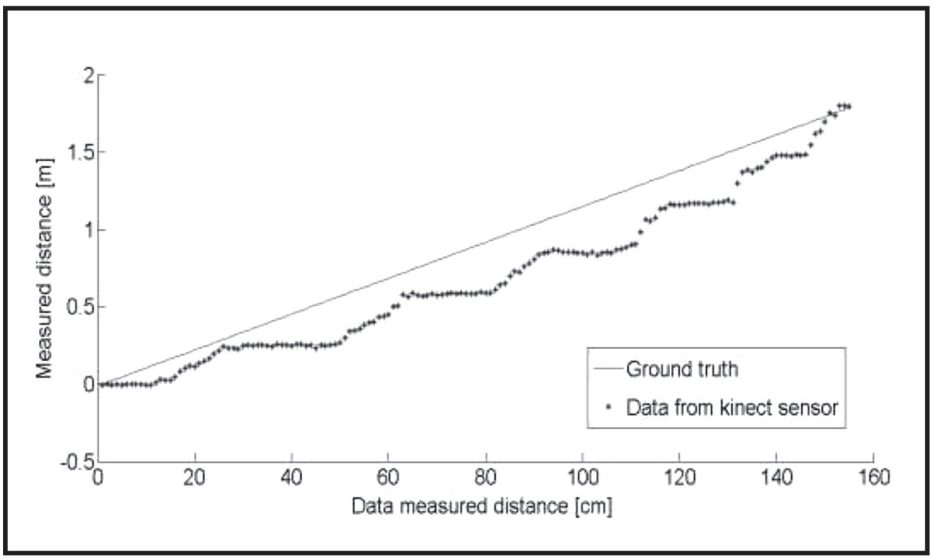
3. พบว่าเมื่อพิจารณาระยะทางการระบุตำแหน่งด้วยวิธีการที่นำเสนอที่มีวัตถุเคลื่อนที่ค่าความผิดพลาดเฉลี่ย RMSE น้อยกว่าวิธีการเดิมที่มีวัตถุเคลื่อนที่ 0.127 เมื่อพิจารณาผลการสร้างแผนที่ ดังรูปที่ 9 (ล่าง) ภาพที่วัตถุเคลื่อนที่จะถูกกรองออกไปประมาณ 95 %

4. พบว่าเมื่อพิจารณาระยะทางการระบุตำแหน่งด้วยวิธีการที่นำเสนอที่ไม่มีวัตถุเคลื่อนที่ค่าความผิดพลาดเฉลี่ย RMSE มีค่าน้อยที่สุด การประมาณระยะทางการเคลื่อนอัลกอริทึม RGBDSLAM ที่ใช้วิธีตรวจจับการเคลื่อนที่ของวัตถุ สามารถกรองเอาเฟรมที่มีค่าความผิดปกติ (Outliers) สูงออกไปก่อนนำภาพเข้าสู่กระบวนการ SLAM ได้

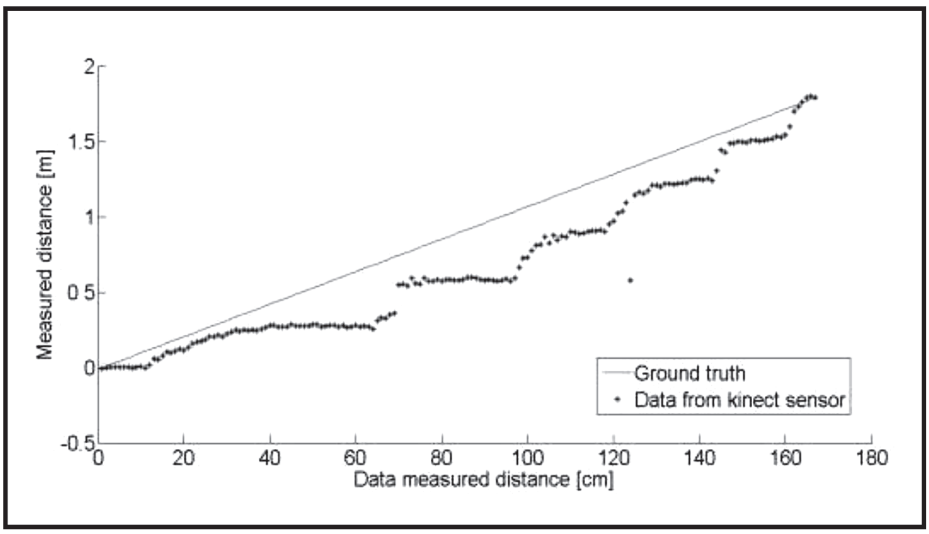
ดังนั้นวิธีการที่นำเสนอกับวิธีการเดิมทั้ง 2 วิธี เมื่อไปทดลองที่ไม่มีวัตถุเคลื่อนที่ให้ผลออกมาไม่เหมือนกัน ซึ่งวิธีการที่นำเสนอมีเปอร์เซ็นต์ความผิดพลาดในการระบุตำแหน่ง 2.35% ใกล้เคียงกับระยะทางจริงมากกว่าวิธีการเดิมที่มีเปอร์เซ็นต์ความผิดพลาดในการระบุตำแหน่ง 9.878 % ความแม่นยำในการระบุตำแหน่งที่มากขึ้นทำให้การระบุตำแหน่งและสร้างแผนที่ถูกต้องมากยิ่งขึ้น ซึ่งเปรียบเทียบได้จากรูปที่ 11



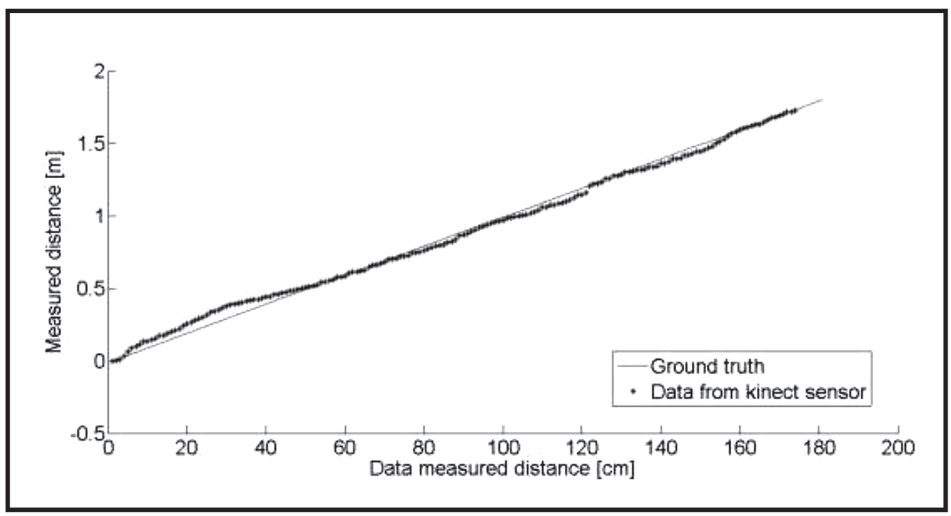
รูปที่ 4 ระยะทางการระบุตำแหน่งบนแผนที่มาตรฐาน (Ground truth) ที่ทดสอบ



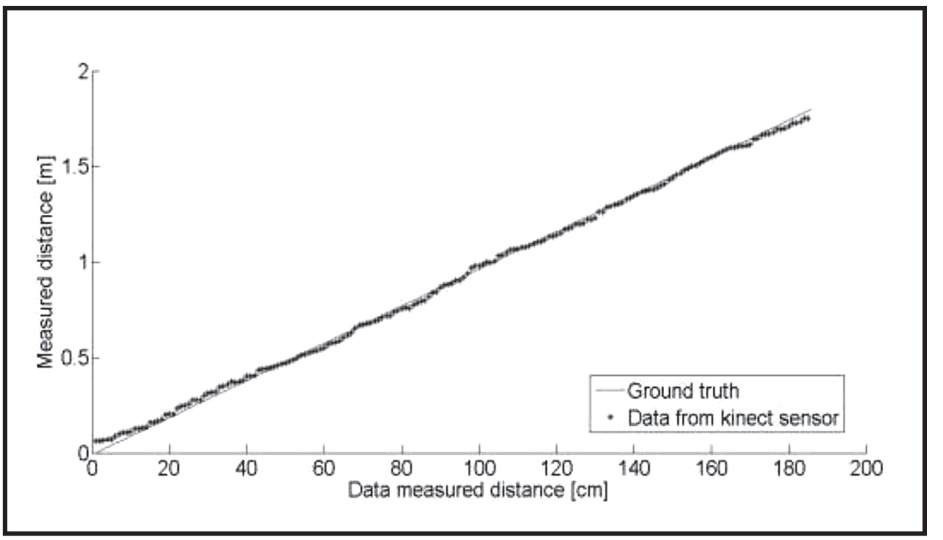
รูปที่ 5 กราฟแสดงความสัมพันธ์ระหว่าง ระยะทางการระบุตำแหน่งที่ได้จาก RGBDSLAM กับแผนที่มาตรฐานด้วยวิธีการเดิมที่ไม่มีวัตถุเคลื่อนที่



รูปที่ 6 กราฟแสดงความสัมพันธ์ระหว่างระยะทางการระบุตำแหน่งที่ได้จาก RGBDSLAM กับแผนที่มาตรฐานด้วยวิธีการเดิมที่มีวัตถุเคลื่อนที่



รูปที่ 7 กราฟแสดงความสัมพันธ์ระหว่างระยะทางการระบุตำแหน่งที่ได้จาก RGBDSLAM กับแผนที่มาตรฐานด้วยวิธีที่นำเสนอที่มีวัตถุเคลื่อนที่



รูปที่ 8 กราฟแสดงความสัมพันธ์ระหว่างระยะทางการระบุตำแหน่งที่ได้จาก RGBDSLAM กับแผนที่มาตรฐานด้วยวิธีที่นำเสนอที่ไม่มีวัตถุเคลื่อนที่

ตารางที่ 5 ระยะทางความผิดพลาดการระบุตำแหน่งที่มีวัตถุเคลื่อนที่

ระยะทาง [cm]	ความแตกต่างของข้อมูล[cm]		ค่าความผิดพลาด [cm]		% ความผิดพลาด	
	วิธีการเดิม	วิธีที่นำเสนอ	วิธีการเดิม	วิธีที่นำเสนอ	วิธีการเดิม	วิธีที่นำเสนอ
0-0.3	0.2264	0.3876	0.0737	0.0876	7.37	8.76
0.3-0.6	0.2779	0.6028	0.3221	0.0028	32.21	0.28
0.6-0.9	0.5779	0.8763	0.3211	0.024	32.11	2.4
0.9-1.2	0.9743	1.1601	0.2257	0.0399	22.57	3.99
1.2-1.5	1.3864	1.4501	0.1136	0.0499	11.36	4.99
1.5-1.8	1.6596	1.7365	0.1404	0.0635	14.04	6.35

เปอร์เซ็นต์ความผิดพลาดเฉลี่ยวิธีการเดิม = 19.943 %

RMSE วิธีการเดิม = 0.179 เซนติเมตร

เปอร์เซ็นต์ความผิดพลาดวิธีการที่นำเสนอ = 4.461 %

RMSE วิธีการที่นำเสนอ = 0.052 เซนติเมตร

ตารางที่ 6 ระยะทางความผิดพลาดการระบุตำแหน่งที่ไม่มีวัตถุเคลื่อนที่

ระยะทาง [cm]	ความแตกต่างของข้อมูล[cm]		ค่าความผิดพลาด [cm]		% ความผิดพลาด	
	วิธีการเดิม	วิธีที่นำเสนอ	วิธีการเดิม	วิธีที่นำเสนอ	วิธีการเดิม	วิธีที่นำเสนอ
0-0.3	0.2365	0.3496	0.0635	0.0496	5.02	4.96
0.3-0.6	0.5009	0.5895	0.0991	0.0105	9.91	1.05
0.6-0.9	0.8083	0.9024	0.0917	0.0024	9.17	0.24
0.9-1.2	1.1063	1.1801	0.0937	0.02	9.37	2
1.2-1.5	1.3726	1.4897	0.1274	0.0103	12.17	1.03
1.5-1.8	1.6858	1.7516	0.1142	0.0484	11.65	4.84

เปอร์เซ็นต์ความผิดพลาดวิธีการเดิม = 9.878 %

RMSE วิธีการเดิม = 0.100 เซนติเมตร

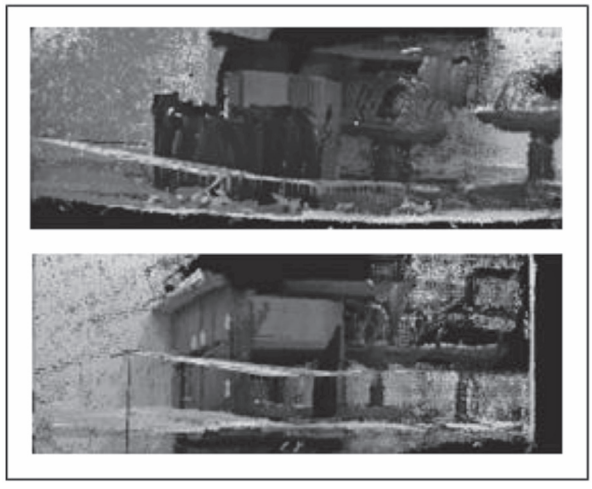
เปอร์เซ็นต์ความผิดพลาดวิธีการที่นำเสนอ = 2.35%

RMSE วิธีการที่นำเสนอ = 0.030 เซนติเมตร

5.2 ผลทดสอบประสิทธิภาพการตรวจจับวัตถุเคลื่อนที่

การจำลองสถานการณ์ขณะระบุตำแหน่งและสร้างแผนที่ที่มีวัตถุเคลื่อนที่ โดยกำหนดให้มีบุคคลใส่กางเกงขายาวสีดำรองเท้าสีส้มเดินผ่านหน้ากล้องไป - กลับ

จากผลการทดลอง พบว่าเมื่อมีวัตถุเคลื่อนที่กล้องจะถ่ายภาพ 30 ภาพต่อวินาทีแล้วทุกภาพถูกนำเข้าสู่กระบวนการระบุตำแหน่งและสร้างแผนที่จะเห็นบริเวณที่มีการเคลื่อนที่ของวัตถุแผนที่จะไม่ถูกต้องตามสิ่งแวดล้อมที่เป็นจริงแต่ถ้าตรวจจับได้ว่าเป็นการเคลื่อนที่ของวัตถุแล้วลบภาพนั้นทิ้งไปก็จะได้แผนที่ที่ถูกต้องตามสภาพแวดล้อมจริง แต่อย่างไรก็ตามภาพที่มีการเคลื่อนที่ของวัตถุยังคงมีเหลืออยู่บ้าง ดังรูปที่ 9 (ล่าง)



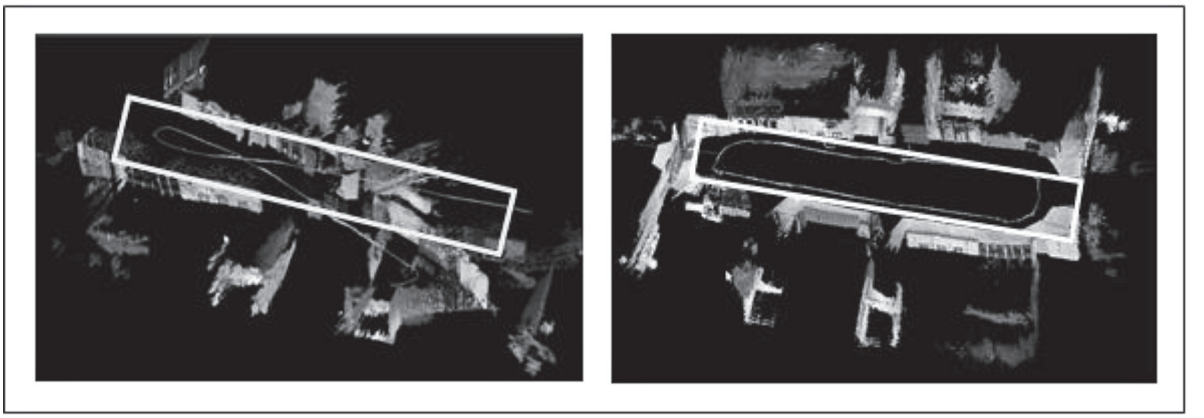
รูปที่ 9 เปรียบเทียบความถูกต้องของแผนที่ 3 มิติ วิธีการเดิม (บน)
วิธีที่นำเสนอตรวจจับวัตถุเคลื่อนที่ (ล่าง)

5.3 การระบุตำแหน่งและสร้างแผนที่ 3 มิติ ภายในอาคาร

การทดสอบการระบุตำแหน่งและสร้างแผนที่ 3 มิติ ด้วยการติดตั้งกล้อง Kinect และคอมพิวเตอร์ไว้กับรถเข็น 4 ล้อ ใช้ระบุตำแหน่งและสร้างแผนที่สิ่งแวดล้อม 3 มิติที่ไม่รู้จักสภาพแวดล้อมมาก่อน ในงานวิจัยนี้ ได้สร้างแผนที่อาคารเรียนรวม 4 ชั้น 7 การทดลองเก็บผลทั้งหมด 2 ครั้ง เปรียบเทียบกันระหว่างทั้ง 2 วิธี โดยไม่มีวัตถุเคลื่อนที่ ประเมินความถูกต้องด้วยแบบแปลนของอาคารตามรูปที่ 10 ในกรอบสีขาว คือ บริเวณพื้นที่ทดลอง ใช้เวลาในการทดสอบ 17.13 นาที

ผลการทดสอบ พบว่าเมื่อพิจารณาเส้นทางการระบุตำแหน่งของหุ่นยนต์ที่ได้จากอัลกอริทึม RGBDSLAM วิธีการเดิม มีลักษณะแตกต่างไปจากพื้นที่จริง ดังรูปที่ 11 (ซ้าย) เป็นทางเดินรูปสี่เหลี่ยมผืนผ้า แต่ผลการระบุตำแหน่งนั้นซ้อนทับกับเส้นทางเดิม เปรียบเทียบกับวิธีที่ได้พัฒนาขึ้นให้ตรวจจับการเคลื่อนที่ของวัตถุแล้วกรองภาพที่วัตถุเคลื่อนที่ออกไปจะเห็นได้ว่าการระบุตำแหน่งการเคลื่อนที่ของหุ่นยนต์นั้นมีความใกล้เคียงกับตำแหน่งแผนที่จริงมากขึ้น ดังรูปที่ 11 (ขวา)

รูปที่ 10 แบบแปลงโครงสร้างอาคารเรียนรวม 4 ชั้น 7



รูปที่ 11 ผลลัพธ์การสร้างแผนที่สิ่งแวดล้อม 3 มิติ ด้วยวิธีการเดิม (ซ้าย) และวิธีการที่นำเสนอ (ขวา)

6. สรุปผลการวิจัย

จากการทดลองอัลกอริทึม RGBDSLAM ที่ใช้วิธีการวิเคราะห์เวกเตอร์ Optical flow เพื่อคัดแยกลักษณะการเคลื่อนที่เปรียบเทียบกับวิธีการเดิม ผลการระบุตำแหน่งและสร้างแผนที่ 3 มิติ ของหุ่นยนต์ ผลการจำลองสถานการณ์การระบุตำแหน่งและสร้างแผนที่ที่มีวัตถุเคลื่อนที่ ผลการสร้างแผนที่ 3 มิติ ภายในอาคาร จากวิธีการที่ได้พัฒนาขึ้นแสดงผลการทดลองในรูปแบบของความผิดพลาดในการระบุตำแหน่งที่ไม่มีวัตถุเคลื่อนที่เทียบกับข้อมูลแผนที่มาตรฐาน เมื่อพิจารณาเปอร์เซ็นต์ความผิดพลาด 2.35% ดังนั้นวิธีการที่พัฒนาขึ้นมีค่าความผิดพลาดของการระบุตำแหน่งน้อยลงเมื่อเทียบกับวิธีการเดิมที่ไม่มีวัตถุเคลื่อนที่มีเปอร์เซ็นต์ความผิดพลาดที่ 9.878% และภาพรวมของผลการสร้างแผนที่ 3 มิติใกล้เคียงกับสภาพแวดล้อมจริงมากยิ่งขึ้นเมื่อมีวัตถุเคลื่อนที่

โดยความแม่นยำที่มากขึ้นจะช่วยให้กระบวนการระบุตำแหน่งและสร้างแผนที่ 3 มิติ ในสภาพแวดล้อมที่มีการเปลี่ยนแปลงได้ ส่งผลให้หุ่นยนต์สามารถทำภารกิจต่างๆ ได้เพิ่มมากขึ้น

7. กิตติกรรมประกาศ

ผู้วิจัยขอขอบพระคุณ รศ.ดร.พจน์ ตั้งงามจิตต์ ภาควิชาวิศวกรรมระบบควบคุมและเครื่องมือวัด คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ที่ได้อำนวยความสะดวกในการทำวิจัยและห้องปฏิบัติการ

8. เอกสารอ้างอิง

1. Smith, R.C. and Cheeseman, P., 1986, "On the Representation and Estimation of Spatial-uncertainty," *The International Journal of Robotics Research*, 5, p. 56.

2. Moutarlier, P. and Chatila, R., 1989, "An Experimental System for Incremental Environment Modeling by an Autonomous Mobile Robot," 1st *International Symposium on Experimental Robotics*, Montreal.

3. Walter, M.R., Eustice, R.M. and Leonard, J.J., 2007, "Exactly Sparse Extended Information Filters for Feature-based SLAM," *The International Journal of Robotics Research*, 28, pp. 335-359.

4. Thrun, S., Liu, Y., Koller, D., Ng, A.Y., Ghahramani Z. and Durrant-Whyte, H.F., 2004, "Simultaneous Localization and Mapping with Sparse Extended Information Filters," *International Journal of Robotics Research*, 23, pp. 693-716.

5. Davison, A.J., Reid, I.D., Molton, N.D. and Stasse, O., 2007, "Monoslam : Real-time Single Camera Slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29, pp. 1052-1067.

6. Montemerlo, M., Thrun, S., Roller, D. and Wegbreit, B., 2003, "Fastslam 2.0 : An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges," *Proceeding of the International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, p. 1151.

7. Sunderhauf, N., Lange, S. and Protzel, P., 2007, "Using the Unscented Kalman Filter in Mono-slam with Inverse Depth Parameterization for Autonomous Airship Control," *Proceeding of IEEE International Workshop on Safety Security and Rescue Robotics*, Rome, Italia, pp. 1-6.

8. Klein, G. and Murray, D., 2007, "Parallel Tracking and Mapping for Small AR Workspaces," *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 1-10.

9. Newcombe, R., Lovegrove, S. and Davison, A., 2011, "DTAM : Dense Tracking and Mapping in Real-Time," *13th International Conference on Computer Vision (ICCV)*, Washington, USA, pp. 2320-2327.

10. Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D. and Burgard, W., 2012, "An Evaluation of the RGB-D SLAM System,"

IEEE international conference on Robotics and Automation, Saint Paul, Minnesota, pp. 1691-1695.

11. Walcott, A. and Bryant, A., 2012, "Dynamic Pose Graph SLAM Long-term Mapping in Low Dynamic Environments," *Intelligent Robot and Systems IEEE International Conference*, Vilamoura, Portugal, pp. 1871-1878.

12. Lucas, B.D. and Kanade, T., 1981, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proceedings of Imaging Understanding Workshop*, pp. 121-130.

13. Durant, H. and Bailey, T., 2006, "Simultaneous Localization and Mapping Part I, II," *IEEE Robotics and Automation Magazine*, pp. 99-110.

14. Harris, C. and Stephens, M., 1988, "A Combined Corner and Edge Detector," *Proceedings of The Fourth Alley Vision Conference*, Manchester, England, pp. 147–151.

15. Lowe, D.G., 2004, "Distinctive Image Features from Scale-Invariant Key Points," *International Journal of Computer Vision*, 60, pp. 91-110.

16. Bay, H., Tuytelaars, T. and Van Gool, L., 2008, "SURF : Speeded Up Robust Features," *Computer Vision and Image Understanding*, 110, pp. 346-359.

17. Angeli, A., Filliat, D. and Doncieux, S., 2008, "A Fast and Incremental Method for Loop-closure Detection using Bags of Visual Words," *IEEE Transactions on Robotics Special Issue on Visual SLAM*, 24, pp. 1027-1037.

18. Grisetti, G., Kummerle, R., Stachniss, C. and Burgard, W., 2010, "A Tutorial on Graph-Based SLAM," *IEEE Intelligent Transportation Systems Magazine*, 2, pp. 31-43.

19. Rusinkiewicz, S. and Levoy, M., 2001, "Efficient Variants of the ICP Algorithm," *Proceeding of the 3rd IEEE International Conference 3D Digital Image and Modeling*, Quebec, French, pp. 145-152.

20. Chum, O., Matas, J. and Kittler, J., 2003, "Locally Optimized RANSAC," *DAGM-Symposium on Pattern Recognition*, 25, pp. 236-243.