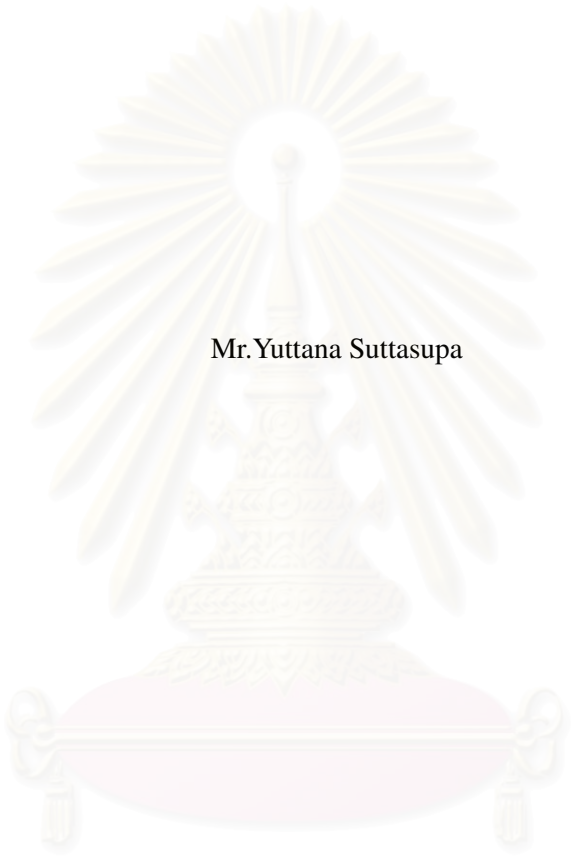
การระบุตำแหน่งและสร้างแผนที่ในสามมิติด้วยกล้องวิดีโอแบบออมนิ

นายยุทธนา สุทธสุภา

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2551

3D SLAM FOR OMNI-DIRECTIONAL CAMERA

Mr. Yuttana Suttasupa



A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering Program in Computer Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2008

หัวข้อวิทยานิพนธ์	การระบุตำแหน่งและสร้างแผนที่ในสามมิติด้วยกล้องวิดีโอแบบออมนิ
โดย	นายยุทธนา สุทธสุภา
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.อรรถวิทย์ สุดแสง

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.อรรถวิทย์ สุดแสง)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.วิทยา วัณณสุโภประสิทธิ์)

..... กรรมการภายนอกมหาวิทยาลัย
(อาจารย์ ดร.วรเศรษฐ สุวรรณิก)

ยุทธนา สุทธสุภา: การระบุตำแหน่งและสร้างแผนที่ในสามมิติด้วยกล้องวิดีโอแบบออมนิ. (3D SLAM FOR OMNI-DIRECTIONAL CAMERA) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ. ดร.อรรถวิทย์ สุดแสง, 105 หน้า.

วิธีการระบุตำแหน่งพร้อมกับการสร้างแผนที่ เป็นกระบวนการที่หุ่นยนต์สามารถระบุตำแหน่งของตนว่าอยู่ ณ จุดใดในแผนที่ ในขณะเดียวกันหุ่นยนต์ก็จะทำการสร้างแผนที่ของสิ่งแวดล้อมในบริเวณที่หุ่นยนต์เคลื่อนที่ผ่านไปพร้อม ๆ กัน โดยที่หุ่นยนต์นั้นไม่มีข้อมูลของสิ่งแวดล้อมมาก่อน ในงานวิจัยนี้ได้นำเสนอวิธีการระบุตำแหน่งพร้อมกับการสร้างแผนที่ในสามมิติด้วยกล้องวิดีโอแบบออมนิ ซึ่งกล้องวิดีโอแบบออมนินั้นจะสามารถเคลื่อนที่ได้อย่างอิสระในสามมิติ โดยที่ไม่จำเป็นต้องรู้ข้อมูลควบคุมการเคลื่อนที่ของกล้อง และไม่จำเป็นต้องรู้โมเดลการเคลื่อนที่ของกล้อง ผลลัพธ์ที่ได้จากการทำงานของอัลกอริทึม จะเป็นแผนที่ของสิ่งแวดล้อมในบริเวณที่กล้องเคลื่อนที่ผ่าน และสถานะของกล้องในสามมิติ โดยแผนที่จะถูกอธิบายด้วยตำแหน่งของจุดสังเกตในสามมิติจำนวนมาก ส่วนสถานะของกล้องจะประกอบด้วยตำแหน่งของกล้องและทิศทางการวางตัวของกล้องในสามมิติ ซึ่งงานวิจัยนี้ก็มีจุดมุ่งหมายว่าหากนำกล้องวิดีโอแบบออมนิไปติดตั้งบนหุ่นยนต์ใด ๆ อัลกอริทึมที่ได้นำเสนอจะสามารถระบุตำแหน่งของหุ่นยนต์ และสร้างแผนที่ในบริเวณที่หุ่นยนต์เคลื่อนที่ได้ในที่สุด และในตอนท้ายของงานวิจัยได้แสดงการทดลองการทำงานของการระบุตำแหน่งพร้อมกับการสร้างแผนที่ด้วยกล้องวิดีโอแบบออมนิในสิ่งแวดล้อมจริง โดยอาศัยมนุษย์ในการถือกล้องและเคลื่อนที่กล้องไปมาในสามมิติ ซึ่งอัลกอริทึมที่ได้นำเสนอนั้นก็สามารถระบุตำแหน่งของกล้อง และสร้างแผนที่ของสิ่งแวดล้อมได้เป็นอย่างดี

ภาควิชา ...วิศวกรรมคอมพิวเตอร์... ลายมือชื่อนิสิต ...ยุทธนา...สุทธสุภา.....
สาขาวิชา...วิศวกรรมคอมพิวเตอร์... ลายมือชื่ออ.ที่ปรึกษาวิทยานิพนธ์หลัก .....
ปีการศึกษา .....2551.....

5070413421: MAJOR COMPUTER ENGINEERING

KEYWORDS: MOBILE ROBOT/ LOCALIZATION/ MAPPING/ OMNI-DIRECTIONAL CAMERA/ SLAM

YUTTANA SUTTASUPA : 3D SLAM FOR OMNI-DIRECTIONAL CAMERA. ADVISOR : ASST.PROF. ATTAWITH SUDSANG, Ph.D., 105 pp.

Simultaneous localization and mapping (SLAM) is a technique that allows a mobile robot to localize itself and build up an environment map simultaneously while traversing in an unknown environment. This work proposes a SLAM method for a hand-held omni-directional camera which is able to move freely in a 3D environment without assuming any control input or a camera motion model. The algorithm presents its resulting environment map in the form of a group of 3D landmarks and presents the camera state containing the camera's position and direction. The thesis also presents the results from experiments showing the localization and mapping outcomes from the proposed method. The experiments are conducted in a real environment using a human carrying an omni-directional camera while traversing in a 3D environment.

Department : ...Computer Engineering... Student's Signature ...
Field of Study : ..Computer Engineering.. Advisor's Signature ...
Academic Year .....2008.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากการสนับสนุนและส่งเสริมเป็นอย่างดีจาก ผศ. ดร.อรรถวิทย์ สุดแสง อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งเป็นที่ปรึกษาทั้งในส่วนของแนวทางในการทำงานวิจัยชิ้นนี้ รวมถึงข้อแนะนำต่าง ๆ ของงานวิจัยและเรื่องอื่น ๆ นอกเหนือจากงานวิจัยข้าพเจ้าจึงขอขอบคุณอาจารย์เป็นอย่างสูงมา ณ ที่นี้

ขอขอบคุณประธานกรรมการสอบวิทยานิพนธ์ ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา ตลอดจนกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.วิทยา วัณณสุโภประสิทธิ์ และ อาจารย์ ดร.วรเศรษฐ สุวรรณิก ที่ได้กรุณาสละเวลา ตรวจสอบและให้คำแนะนำเกี่ยวกับวิทยานิพนธ์ฉบับนี้จนเสร็จสมบูรณ์

ที่จะขาดเสียมิได้ คือขอขอบคุณเหล่าพี่ ๆ เพื่อน ๆ และ น้อง ๆ แห่งห้องปฏิบัติการวิจัยระบบอัจฉริยะ ISL2 ทุก ๆ คนที่ช่วยให้คำแนะนำต่าง ๆ ทั้งในเรื่องของงานวิจัยและเรื่องอื่น ๆ รวมถึงแนทางในการแก้ปัญหาและอุปสรรคต่าง ๆ

สุดท้ายนี้ ขอขอบคุณ บิดามารดา ผู้ให้กำเนิด รวมไปถึงญาติพี่น้องทุกคน ที่ให้กำลังใจและสนับสนุนผู้วิจัยตลอดมา และขอขอบคุณอีกหลาย ๆ ท่านที่ไม่สามารถเอ่ยนามได้ทั้งหมด ณ ที่นี้ด้วยใจจริง

สารบัญ

หน้า

บทที่ หน้า

บทที่ หน้า

สารบัญตาราง

สารบัญภาพ

รูปที่ หน้า

รูปที่ หน้า

บทที่ 1

บทนำ

1.1 ปัญหา

เป็นเวลานานมาแล้วที่มนุษย์พยายามสร้างสรรค์เครื่องจักรที่มีความสามารถและมีความเฉลียวฉลาดมากพอที่จะทำงานที่มีความซับซ้อนแทนมนุษย์ได้ ตั้งแต่ครั้งโบราณกาลมนุษย์ได้พยายามประดิษฐ์คิดค้นเครื่องมือเครื่องใช้ในชีวิตประจำวัน เพื่อการทุ่นแรงอาทิเช่น มีด หอก จอบ เสียม หรือเพื่อใช้เป็นเครื่องมืออำนวยความสะดวก เช่น ถ้วยชาม หม้อ กระทะ ต่อมาในยุคล่าอาณานิคม มนุษย์ได้พัฒนาเครื่องจักรที่มีความสามารถมากขึ้น ไม่เพียงแค่ทุนแรงมนุษย์แต่เครื่องจักรได้เข้ามาช่วยงานของมนุษย์ หรือแม้แต่เข้ามาทำงานแทนมนุษย์เลยทีเดียว ไม่ว่าจะเป็น เครื่องทอผ้า, เครื่องจักรในโรงงาน, รถยนต์, รถไฟ, เรือหรือเครื่องบิน ถึงแม้เครื่องจักรเหล่านั้นจะทำหน้าที่ของมันได้เป็นอย่างดี แต่ว่าเครื่องจักรทั้งหลายเหล่านั้นทำงานได้เฉพาะงานที่มีความซับซ้อนน้อย ๆ ต่อมาเมื่อถึงยุคที่อุปกรณ์คอมพิวเตอร์เริ่มมีความแพร่หลาย มนุษย์ก็พยายามที่จะสร้างเครื่องจักรที่ความเฉลียวฉลาดมากขึ้น โดยมุ่งหวังว่าเครื่องจักรจะสามารถทำงานที่ซับซ้อนได้มากขึ้น จนสุดท้ายเครื่องจักรนั้นจะสามารถทำงานทุกอย่างได้แทนมนุษย์เลยทีเดียว ซึ่งเราได้เรียกเครื่องจักรเหล่านั้นว่า "หุ่นยนต์"

คำว่าหุ่นยนต์นั้นไม่มีคำจำกัดความที่แน่นอน แต่ก็เป็นที่เข้าใจในหมู่คนส่วนใหญ่ว่า เครื่องจักรที่สามารถเป็นหุ่นยนต์ได้ควรจะมีลักษณะดังต่อไปนี้

1. สามารถเคลื่อนที่ไปมาได้
2. มีระยาง หรือ อุปกรณ์ที่สามารถจัดการกับสิ่งแวดล้อมได้
3. สามารถตรวจวัดและมีปฏิสัมพันธ์กับสิ่งแวดล้อมได้
4. มีความฉลาดในเชิงความคิดการทำงาน โดยเฉพาะความสามารถที่คล้ายคลึงกับมนุษย์

เมื่อกล่าวถึงหุ่นยนต์ โดยทั่วไปแล้วคนส่วนใหญ่มักจะนึกถึงเครื่องจักรที่มีลักษณะคล้ายคลึงกับมนุษย์ ทั้งนี้เป็นเพราะอิทธิพลของภาพยนตร์จำนวนมาก ที่พยายามแสดงภาพลักษณ์ของหุ่นยนต์ไว้อย่างเลิศเลอ แท้จริงแล้วหุ่นยนต์อาจมีรูปร่างลักษณะได้หลายรูปแบบขึ้นอยู่กับวัตถุประสงค์ในการใช้งาน หรือหุ่นยนต์อาจไม่มีรูปกาย แต่เป็นเพียงซอฟต์แวร์ที่สามารถทำงานได้ด้วยตัวมันเองก็ได้ ซึ่งหุ่นยนต์ที่เป็นเพียงซอฟต์แวร์มักถูกเรียกว่า "บอต" (bots มาจากคำว่า robots)

ถึงแม้ในปัจจุบันนี้ หุ่นยนต์ยังไม่อาจจะทำงานแทนมนุษย์ได้โดยสมบูรณ์แบบ แต่หุ่นยนต์ก็ถูกสร้างขึ้นเพื่อทำงานแทนมนุษย์ในงานเฉพาะหลาย ๆ งานที่มนุษย์ไม่อยากทำหรือทำไม่ได้ ยกตัวอย่างเช่น งานที่ต้องทำซ้ำ ๆ (เช่น สายงานการผลิตในโรงงาน), งานที่ต้องเสี่ยงอันตราย (เช่น งานกู้ภัย หรืองานที่เกี่ยวกับสารเคมีอันตราย), งานที่ต้องทำระยะไกล (เช่น งานสำรวจใต้น้ำ สำรวจนอกโลก หรือในบริเวณที่มนุษย์ไปไม่ถึง) และงานประเภทอื่น ๆ อีกมากมาย ซึ่งหุ่นยนต์ที่ถูกใช้งานในปัจจุบันนี้ก็มีได้หลายประเภทเช่น หุ่นยนต์รับใช้ในบ้าน, หุ่นยนต์โรงงาน, หุ่นยนต์สำรวจ (หุ่นยนต์นำร่อง หุ่นยนต์สำรวจอวกาศ หุ่นยนต์ใต้น้ำ), หุ่นยนต์กู้ภัย, หุ่นยนต์สงคราม (เครื่องบินสอดแนมไร้คนขับ, หุ่นยนต์รบ), รถอัตโนมัติ, หุ่นยนต์ทางการแพทย์, หุ่นยนต์การเกษตร หรือแม้แต่ หุ่นยนต์เพื่อความบันเทิง (หุ่นยนต์เล่นเปียโน, หุ่นยนต์สุนัข, หุ่นยนต์แมวน้ำ)

สำหรับงานที่มีความท้าท้ายที่สุดอย่างหนึ่งสำหรับหุ่นยนต์ในขณะนี้ก็คือ ระบบอัจฉริยะ ซึ่งเป็นส่วนที่มีหน้าที่ประมวลผลเชิงความคิดการทำงานของหุ่นยนต์ทั้งหมด ซึ่งความท้าทายในส่วนระบบอัจฉริยะก็คือการที่จะทำให้หุ่นยนต์สามารถโต้ตอบกับสิ่งแวดล้อมที่เปลี่ยนแปลงอยู่ตลอดเวลาได้อย่างชาญฉลาด การทำงานของระบบอัจฉริยะจะประกอบด้วย 3 ส่วน คือ

1. ส่วนรับข้อมูลและวิเคราะห์ข้อมูลจากเซนเซอร์ ให้อยู่ในรูปแบบที่หุ่นยนต์สามารถเข้าใจได้
2. ส่วนปัญญาประดิษฐ์ หรือ Artificial Intelligence (AI) มีหน้าที่วิเคราะห์ข้อมูลที่ได้จากเซนเซอร์ และประมวลผลว่าจะทำงานอย่างไรเพื่อที่จะได้ตอบสนองต่อสิ่งแวดล้อมขณะนั้น ๆ ตามลักษณะงานที่ต้องการจะทำ
3. ส่วนควบคุมการทำงานของหุ่นยนต์ มีหน้าที่ควบคุมกลไกของหุ่นยนต์ ไม่ว่าจะเป็นแขนขา หรือล้อของหุ่นยนต์ เพื่อให้ตอบสนองต่อ พฤติกรรมของหุ่นยนต์ ตามที่ส่วนปัญญาประดิษฐ์ ต้องการ

ในส่วนรับข้อมูลและวิเคราะห์ข้อมูลจากเซนเซอร์นั้นมีก็เปรียบเสมือนประสาทสัมผัสทั้ง 5 ของมนุษย์ ซึ่งถ้าหากมนุษย์สูญเสียประสาทสัมผัสทั้งหมดแล้ว มนุษย์จะไม่สามารถรับข้อมูลอะไรจากสิ่งแวดล้อมได้เลยซึ่งจะส่งผลให้ไม่สามารถมีปฏิสัมพันธ์โต้ตอบกับสิ่งแวดล้อมได้ ซึ่งสำหรับหุ่นยนต์ก็เช่นกัน สิ่งที่ขาดไม่ได้เลยสำหรับหุ่นยนต์ก็คือ เซนเซอร์ (Sensor) และส่วนวิเคราะห์ข้อมูลจากเซนเซอร์ ยกตัวอย่างเช่น แขนกลในโรงงานก็จะต้องมี encoder ซึ่งมีหน้าที่วัดตำแหน่งการวางตัวของแขน เพื่อให้แขนวางตัวได้ในท่าที่ถูกต้อง และอาจจะมีเซนเซอร์ประเภทอื่นได้อีกอย่างเช่นอาจจะมีกล้อง เพื่อใช้วัดรูปร่างวัตถุที่จะให้แขนกลจับ หรือในหุ่นยนต์ประเภทอื่น เช่น หุ่นยนต์ตุ๊กตาสุนัข อาจจะมีเซนเซอร์ เช่น อุปกรณ์อินฟาเรดใช้วัดระยะ, force sensors ใช้วัดว่าหัวสุนัขถูกลูบหรือเปล่า หรือ มีกล้องวิดีโอใช้ตรวจหาวัตถุที่มีสีส้มเพื่อให้หุ่นยนต์สุนัขวิ่งตาม

สำหรับ Mobile robots หรือหุ่นยนต์ที่มีความสามารถในการเคลื่อนที่ไปมาได้อย่างอิสระ หุ่นยนต์จะต้องมีความสามารถที่สำคัญอย่างหนึ่งก็คือ ความสามารถในการระบุตำแหน่งของหุ่นยนต์ (Localization) ว่า ณ ขณะนั้นหุ่นยนต์อยู่ในตำแหน่งไหนในแผนที่ที่จำลองสิ่งแวดล้อมรอบ ๆ ตัวหุ่นยนต์ และถ้าหากว่าหุ่นยนต์ต้องทำงานในสภาพแวดล้อมที่ไม่คุ้นเคย หมายความว่าหุ่นยนต์จะไม่มีแผนที่สิ่งแวดล้อม ณ บริเวณที่หุ่นยนต์สนใจ หุ่นยนต์จำเป็นที่จะต้องมีความสามารถในการสร้างแผนที่ของสภาพแวดล้อมขณะปฏิบัติงานได้ ซึ่งงานทั้งสองงาน คือการระบุตำแหน่งของหุ่นยนต์และการสร้างแผนที่ของหุ่นยนต์ ก็ถือเป็นส่วนรับข้อมูลและวิเคราะห์ข้อมูลจากเซนเซอร์ ให้อยู่ในรูปแบบที่หุ่นยนต์สามารถเข้าใจได้ ซึ่งถือเป็นงานหลักที่มีความจำเป็นสำหรับ Mobile robots เป็นอย่างยิ่ง

การระบุตำแหน่งของหุ่นยนต์ (Localization) ก็คือการที่หุ่นยนต์สามารถระบุตนเองได้ว่าหุ่นยนต์อยู่ ณ ตำแหน่งหรือบริเวณไหนในแผนที่ โดยการระบุตำแหน่งของหุ่นยนต์ก็มีจุดประสงค์เพื่อให้หุ่นยนต์รู้ถึงการมีอยู่ของตนเองในแผนที่ ซึ่งจะทำให้หุ่นยนต์สามารถวิเคราะห์และวางแผนการเคลื่อนที่ได้ เมื่อเปรียบเทียบกับมนุษย์แล้วก็เหมือนกับการที่มนุษย์เดินเข้าไปในห้องห้องหนึ่งและมนุษย์สามารถบอกได้ว่าตนเองอยู่ในบริเวณใดของห้อง อยู่ห่างจากกำแพงประมาณกี่เมตร ซึ่งถ้าหากมนุษย์ต้องการหยิบของสักชิ้นมนุษย์ก็จะใช้ข้อมูลตำแหน่งของตนเองในการวัดเทียบว่าของในแผนที่นั้นอยู่ในทิศทางใดเทียบกับตนเองและมนุษย์ก็จะเดินไปหยิบของได้อย่างถูกต้อง

การระบุตำแหน่งของหุ่นยนต์นั้น มีการใช้ในแทบทุกงานของหุ่นยนต์ที่จะต้องมีการเคลื่อนที่ ทั้งนี้เป็นเพราะการระบุตำแหน่งของหุ่นยนต์ถือเป็นส่วนสำคัญที่สุดในการทำงานของ Mobile Robots ยกตัวอย่างเช่น หุ่นยนต์กวาดบ้าน จำเป็นจะต้องระบุตำแหน่งของตนเองเพื่อที่จะรู้ได้ว่าบริเวณห้องที่ต้องการจะกวาดมีขอบเขตตรงไหนบ้าง และหุ่นยนต์กำลังอยู่ตรงไหนในบริเวณห้อง เพื่อที่หุ่นยนต์จะได้นำข้อมูล

ไปวิเคราะห์ว่าบริเวณใดที่หุ่นยนต์กวาดไปแล้ว และบริเวณใดที่หุ่นยนต์ยังไม่ได้กวาด สำหรับหุ่นสำรวจก็ต้องการระบุตำแหน่งของตนเองเพื่อหุ่นยนต์จะได้รู้ว่าในบริเวณใดที่ทำการสำรวจไปแล้วบริเวณใดยังไม่ได้สำรวจ เพื่อหุ่นยนต์จะได้สำรวจได้อย่างทั่วถึง สำหรับหุ่นยนต์การเกษตรก็ต้องการระบุตำแหน่งว่าหุ่นยนต์นั้นอยู่ในตำแหน่งใดในสวนและ หุ่นยนต์จะได้รู้ว่ามีต้นไม้อยู่ในตำแหน่งไหนเทียบกับตัวหุ่นยนต์บ้าง เพื่อที่หุ่นยนต์จะได้รถน้ำใส่ปุ๋ยให้กับต้นไม้หรือกระทำกิจกรรมต่าง ๆ ได้อย่างถูกต้อง หรืออย่างเช่น รถอัตโนมัติไร้คนขับ ก็ต้องการระบุตำแหน่งของรถเพื่อจะได้วิเคราะห์ได้ว่าขณะนี้รถกำลังอยู่ในบริเวณใดในแผนที่ และการที่รถจะไปถึงเป้าหมายได้นั้นจะต้องเลี้ยวหรือขับไปทางใดบ้าง และการระบุตำแหน่งอย่างละเอียด เพื่อวิเคราะห์การจราจรว่าขณะนั้นรถอยู่บริเวณไหนในถนน มีรถคันอื่นอยู่ตรงไหนบ้างเทียบกับตัวเอง เพื่อจะได้วางแผนการขับได้อย่างถูกต้อง

การสร้างแผนที่ (Mapping) คือการที่หุ่นยนต์สามารถจำลองสภาพแวดล้อมบริเวณรอบ ๆ ตัวหุ่น หรือในบริเวณที่หุ่นยนต์สนใจได้ ข้อมูลของแผนที่ที่หุ่นยนต์สร้างได้นั้นจะต้องอยู่ในรูปแบบที่หุ่นยนต์สามารถเข้าใจได้ ซึ่งหุ่นยนต์จะนำข้อมูลแผนที่นี้ไปใช้ร่วมกับข้อมูลการระบุตำแหน่งเพื่อการวางแผนการทำงานต่อไป ยกตัวอย่างเช่น สำหรับห้องห้องหนึ่ง แผนที่สำหรับห้องห้องนั้นอาจจะเป็นตำแหน่งของสิ่งของต่าง ๆ ที่หุ่นยนต์สนใจเช่นในแผนที่จะเก็บข้อมูลว่าโต๊ะ ตู้ เตียงอยู่ ณ ตำแหน่งใด มีลักษณะทางกายภาพเป็นอย่างไร มีหนังสือ สมุด ดินสอ ปากกา แก้วน้ำ แจกัน อยู่ในตำแหน่งใดบ้างของแผนที่ เป็นต้น ซึ่งเมื่อนำมาประกอบกับข้อมูลการระบุตำแหน่งของหุ่นยนต์แล้ว หุ่นยนต์ก็จะรู้ได้ว่าตำแหน่งของตนเองอยู่ในตำแหน่งไหน เทียบกับตำแหน่งของสิ่งของต่าง ๆ และถ้าหุ่นยนต์จะต้องทำงานสักงานยกตัวอย่างเห็น หุ่นยนต์จะต้องเดินไปหยิบแก้วน้ำ หุ่นยนต์ก็จะรู้ได้ว่าหุ่นยนต์จะต้องเดินไปในทิศทางใดเพื่อที่จะได้ไปถึงแก้วน้ำ และหยิบได้อย่างถูกต้อง

ในการทำงานของหุ่นยนต์ การสร้างแผนที่อาจไม่จำเป็นก็ได้ถ้าหากว่าหุ่นยนต์รู้แน่นอนแล้วว่าจะต้องทำงานในสภาพแวดล้อมลักษณะเป็นอย่างไร หุ่นยนต์ก็สามารถเก็บข้อมูลของแผนที่ล่วงหน้าไว้ก่อน ซึ่งแผนที่ที่ถูกสร้างล่วงหน้านั้นอาจถูกสร้างโดยมนุษย์เอง หรืออาจถูกสร้างโดยวิธีอื่น แต่ถ้าหากว่าหุ่นยนต์จะต้องไปทำงานในสิ่งแวดล้อมที่มีลักษณะไม่แน่นอน สามารถเปลี่ยนแปลงได้ตลอดเวลา หรือหุ่นยนต์ต้องไปทำงานในสิ่งแวดล้อมที่ไม่คุ้นเคยมาก่อน ยกตัวอย่างเช่นหุ่นยนต์กู้ภัยที่ต้องทำงานในซากตึกปรักหักพัง หรือหุ่นยนต์สำรวจใต้น้ำ หุ่นยนต์สำรวจดาวอังคาร เป็นต้น หุ่นยนต์เหล่านี้จำเป็นจะต้องมีความสามารถในการสร้างแผนที่ได้เอง

ตัวอย่างการสร้างแผนที่ของหุ่นยนต์ อย่างเช่น หุ่นยนต์สำรวจดาวอังคาร ต้องการสร้างแผนในที่บริเวณที่หุ่นยนต์เคลื่อนที่ผ่าน นอกจากจะใช้วางแผนการสำรวจแล้ว ยังใช้เป็นข้อมูลให้มนุษย์ศึกษาอีกด้วย การสร้างแผนที่ของรถยนต์อัตโนมัติก็เพื่อวิเคราะห์สภาพแวดล้อมว่ามีถนนอยู่บริเวณใดบ้าง ถนนมีลักษณะเป็นอย่างไร มีรถคันอื่นอยู่ตรงตำแหน่งใดบ้าง เพื่อที่รถยนต์อัตโนมัติจะได้เคลื่อนที่หลบหลีกได้อย่างถูกต้อง หรือการสร้างแผนที่อาจไม่เกี่ยวข้องกับการเคลื่อนที่ของหุ่นยนต์ก็ได้ได้ เช่น การสร้างแผนที่พื้นผิวใต้ทะเลเพื่อการวิเคราะห์ร่องน้ำการเดินเรือ และอื่น ๆ

สำหรับหุ่นยนต์ที่ต้องทำงานในสภาพแวดล้อมที่ไม่คุ้นเคย หุ่นยนต์จำเป็นที่จะต้องมีความสามารถในการสร้างแผนที่และการระบุตำแหน่งแบบทันกาล ซึ่งหมายความว่าหุ่นยนต์จะต้องมีความสามารถในการสร้างแผนที่และระบุตำแหน่งให้ได้ ณ ขณะนั้น เพื่อที่จะได้ใช้ข้อมูลแผนที่และตำแหน่งในการวางแผนการเคลื่อนที่ต่อไป โดยการระบุตำแหน่งของหุ่นยนต์นั้นก็จำเป็นต้องใช้ข้อมูลแผนที่เพื่อเป็นกรอบอ้างอิงในการระบุพิกัดของหุ่นยนต์ ส่วนในการสร้างแผนที่นั้น จะสร้างขึ้นได้จากการนำข้อมูลจากเซนเซอร์หลาย ๆ ข้อมูลจากบริเวณหลาย ๆ บริเวณรอบสิ่งแวดล้อมที่หุ่นยนต์สนใจ มาประติดประต่อกันเป็นรูปร่างของแผนที่ ซึ่งในกระบวนการประติดประต่อข้อมูลนั้น หุ่นยนต์ก็จำเป็นจะต้องอาศัยตำแหน่งของหุ่นยนต์ ณ บริเวณที่วัดข้อมูลจากเซนเซอร์เพื่อการคำนวนแผนที่แบบสัมบูรณ์ ดังนั้นปัญหาการระบุตำแหน่งพร้อมกับการสร้างแผนที่ จึงเป็นปัญหาที่เกี่ยวพันกันเป็นอย่างมาก จึงทำให้มีวิธีการเพื่อแก้ปัญหาการระบุตำแหน่ง

พร้อมกับการสร้างแผนที่โดยเฉพาะ

วิธีการระบุตำแหน่งพร้อมกับการสร้างแผนที่ (Simultaneous Localization and Mapping: SLAM) [Durrant06] เป็นกระบวนการที่หุ่นยนต์สามารถสร้างแผนที่ของสภาพแวดล้อมในขณะที่กำลังเคลื่อนที่และระบุตำแหน่งของตนเองไปพร้อม ๆ กัน โดยที่หุ่นยนต์นั้นไม่มีข้อมูลของสิ่งแวดล้อมมาก่อน ซึ่งปัญหาของ SLAM นั้นมีความสำคัญเป็นอย่างมากสำหรับหุ่นยนต์ที่ต้องการการโต้ตอบแบบทันการณ์ โดย SLAM นั้นจะอธิบาย ตำแหน่งของหุ่นยนต์ และ แผนที่ ในรูปของความน่าจะเป็น ที่ขึ้นอยู่กับข้อมูลการวัดจากเซนเซอร์

ทั้งการสร้างแผนที่และการระบุตำแหน่งนั้นอาจจะมีวิธีการหรือขั้นตอนหลายรูปแบบ ขึ้นอยู่กับสภาพแวดล้อมที่จะสร้างแผนที่ และอุปกรณ์เซนเซอร์ที่ใช้ สำหรับลักษณะการติดตั้งอุปกรณ์เซนเซอร์ก็มีส่วนสำคัญต่อวิธีการ ซึ่งการติดตั้งอุปกรณ์เซนเซอร์สำหรับงานระบุตำแหน่งและการสร้างแผนที่ก็มีทั้งแบบ Global Sensor และแบบ Local Sensor การติดตั้งอุปกรณ์เซนเซอร์แบบ Global Sensor ก็คือการติดตั้งอุปกรณ์เซนเซอร์ส่วนหนึ่งหรือทั้งหมดบนวัตถุหรือสถานที่ใด ๆ ที่รู้พิกัดเป็นค่าตำแหน่งที่แน่นอน แล้วจึงใช้เซนเซอร์นั้น ๆ ตรวจวัดตำแหน่งของหุ่นยนต์ที่เคลื่อนที่ไปมาอย่างอิสระ ตัวอย่างเซนเซอร์ประเภทนี้ยกตัวอย่างเช่น อุปกรณ์ GPS ซึ่งนอกจากตัวรับสัญญาณ GPS ที่ติดตั้งที่ตัวหุ่นยนต์แล้ว ยังมีดาวเทียมโคจรอยู่นอกโลกซึ่งรู้วิถีโคจรที่แน่นอนในการช่วยประมาณตำแหน่งของอุปกรณ์ หรืออย่างเช่นการติดตั้งกล้องวิดีโอในมุมสูงเพื่อตรวจวัดตำแหน่งของหุ่นยนต์ที่อยู่ด้านล่าง เช่นนี้ก็ถือเป็น Global Sensor ซึ่งข้อดีของ Global Sensor ก็คือสามารถระบุตำแหน่งโดยรับประกันความคลาดเคลื่อนของการระบุตำแหน่งได้ ซึ่งแตกต่างกับ Local Sensor ซึ่งเป็นการติดตั้งอุปกรณ์เซนเซอร์บนตัวหุ่นยนต์เท่านั้นจึงทำให้ความคลาดเคลื่อนของการระบุตำแหน่งด้วย Local Sensor นั้นสามารถเพิ่มสูงขึ้นได้เรื่อย ๆ แต่ข้อดีของการติดตั้งอุปกรณ์แบบ Local Sensor ก็คือมีความคล่องตัวในการใช้งานสูงเพราะไม่ต้องเสียเวลาติดตั้งอุปกรณ์ที่ฐานซึ่งเคลื่อนย้ายลำบาก เป็นต้นว่าหากต้องการส่งหุ่นยนต์ไปสำรวจดาวอังคาร ก็จำเป็นที่จะต้องใช้ Local Sensor เพราะบนดาวอังคารนั้นใช้ GPS ไม่ได้

สำหรับวิทยานิพนธ์ฉบับนี้จะสนใจเฉพาะการระบุตำแหน่งและการสร้างแผนที่โดยใช้ Local Sensor เท่านั้น ซึ่งเซนเซอร์ที่ใช้ในงานการระบุตำแหน่งและการสร้างแผนที่ก็มีได้หลายชนิด อาทิเช่น กล้องวิดีโอ, โซนาร์ (Sonar) หรือ อุปกรณ์วัดระยะด้วยเลเซอร์ (Laser Range Finder) โดยเซนเซอร์แต่ละชนิดก็จะมีวิธีการใช้งานและข้อดีข้อเสียที่แตกต่างกัน เช่น

โซนาร์เป็นเซนเซอร์ซึ่งวัดระยะทางโดยอาศัยการจับเวลาของเสียงสะท้อน มีจุดเด่นในด้านราคาที่ไม่สูงมาก และความสะดวกในการใช้งาน แต่มีปัญหาในเรื่องความแม่นยำ และถูกรบกวนได้ง่ายจากปัจจัยภายนอก เช่นผิวสะท้อนที่แตกต่างกัน หรือ อุณหภูมิของอากาศ

อุปกรณ์วัดระยะด้วยเลเซอร์ เป็นอุปกรณ์วัดระยะที่ใช้หลักการสะท้อนของแสงจึงถูกรบกวนได้ยากกว่าเสียง ทำให้ให้เลเซอร์มีความแม่นยำ และความเร็วในการวัดค่าสูงกว่าโซนาร์ แต่ ก็จัดว่าอุปกรณ์วัดระยะด้วยเลเซอร์นั้นเป็นอุปกรณ์ที่มีราคาสูงชนิดหนึ่ง

กล้องวิดีโอนั้น จะให้ข้อมูลของสิ่งแวดล้อมเป็นปริมาณมาก (ข้อมูลเป็นภาพ) มีความแม่นยำของข้อมูลพอประมาณ ขึ้นกับความละเอียดของกล้อง มีการตอบสนองต่อข้อมูลสิ่งแวดล้อมแบบทันการณ์ (real-time) ราคาไม่สูงนัก พกพาสะดวก แต่การนำข้อมูลมาใช้นั้นมีความยุ่งยาก ใช้เวลาในการประมวลผลนาน และยังต้องอาศัยแสงจากภายนอกเพื่อช่วยในการทำงานในขณะรับภาพ จึงทำให้การเปลี่ยนแปลงของแสงจากสภาพแวดล้อม ส่งผลกระทบต่อการทำงานของกล้องได้ง่าย และ กล้องยังไม่อาจทำงานได้ในสภาพแวดล้อมที่มีแสงน้อยจนเกินไป

เนื่องในช่วงระยะหลังนี้กล้องวิดีโอ มีราคาลดต่ำลงเป็นอย่างมาก อีกทั้งคอมพิวเตอร์ก็มีประสิทธิภาพ

สูงขึ้นด้วย ดังนั้นกล้องวิดีโอจึงเป็นที่นิยมในการใช้เป็นเซนเซอร์สำหรับงานระบุตำแหน่งพร้อมกับการสร้างแผนที่ (SLAM) มากขึ้น ทั้งนี้เพราะข้อดีทั้งหลายของกล้องวิดีโอ อีกทั้งกล้องวิดีโอยังเป็นอุปกรณ์ที่มีการทำงานคล้ายคลึงกับการทำงานของดวงตาของมนุษย์อีกด้วย ข้อดีของการใช้กล้องวิดีโอในงานระบุตำแหน่งพร้อมกับการสร้างแผนที่ก็คือ กล้องวิดีโอให้ข้อมูลของสิ่งแวดล้อมเป็นภาพแบบ bitmap ปริมาณมาก ทำให้วัดข้อมูลสิ่งแวดล้อมได้ทีละหลาย ๆ ค่า แตกต่างกับเซนเซอร์ประเภทอื่นเช่น โซนาร์ ซึ่งวัดข้อมูลระยะทางได้เพียงค่าเดียว และภาพจากกล้องยังมีความละเอียดที่ค่อนข้างสูง ทำให้สามารถวัดข้อมูลของสิ่งแวดล้อมได้อย่างแม่นยำ นอกจากนี้กล้องวิดีโอยังมีความเร็วในการรับข้อมูลที่ค่อนข้างสูง ส่งผลให้หุ่นยนต์สามารถตอบสนองต่อสิ่งแวดล้อมได้เร็ว อีกทั้งกล้องวิดีโอยังให้ข้อมูลของสิ่งแวดล้อมเป็นค่าความเข้มของแสงประกอบด้วยสีหลายสี ทำให้สามารถแยกแยะสิ่งแวดล้อมที่แตกต่างกันได้ง่าย แต่ข้อเสียของกล้องวิดีโอก็คือข้อมูลจากกล้องวิดีโอซึ่งประกอบด้วยจุดสีจำนวนมากนั้น เป็นข้อมูลที่คอมพิวเตอร์ไม่สามารถเข้าใจได้โดยง่าย เพราะจะต้องนำข้อมูลของจุดสีหลาย ๆ จุดจากบริเวณข้างเคียงมาประกอบกันจึงจะสามารถตีความเป็นลักษณะของสิ่งแวดล้อมได้ ซึ่งงานตีความจากข้อมูลภาพด้วยคอมพิวเตอร์ (Computer Vision) นั้นเป็นงานที่ต้องใช้เวลาในการประมวลผลนาน อีกทั้งความผิดพลาดของการตีความยังเกิดขึ้นได้ง่าย เนื่องจากอัลกอริทึมของ Computer Vision ณ ปัจจุบันนั้นยังไม่ดีพอ ดังนั้นปัญหาการตีความจากข้อมูลภาพจึงเป็นปัญหาหลักของกล้องวิดีโอ นอกจากนี้กล้องวิดีโอยังมีปัญหาเรื่องการวัดค่า เนื่องจากข้อมูลภาพจากกล้องเป็นภาพในสองมิติ ซึ่งเกิดจากการ project ข้อมูลสิ่งแวดล้อมทั้งหลายที่เห็นสามมิติ ลงมาบนภาพดังนั้นจึงทำให้ภาพขาดข้อมูลที่สำคัญไปอย่างหนึ่งก็คือข้อมูลความลึกของวัตถุในภาพ ซึ่งในงานสร้างแผนที่และการระบุตำแหน่งของหุ่นยนต์นั้นข้อมูลความลึกของวัตถุในภาพย่อมมีความสำคัญ และเมื่อภาพจากกล้องนั้นขาดข้อมูลความลึกไปจึงทำให้กระบวนการสร้างแผนที่และระบุตำแหน่งมีความยุ่งยากมากขึ้น ปัญหาอีกประการหนึ่งของกล้องวิดีโอก็คือ กล้องวิดีโอแบบทั่วไปจะมีมุมมองของภาพที่ค่อนข้างแคบ ซึ่งหมายความว่าภาพที่ได้จากกล้องวิดีโอนั้น เป็นเพียงภาพในมุมมองเล็ก ๆ ของสิ่งแวดล้อมทั้งหมดเท่านั้น ทำให้การสร้างแผนที่ของสิ่งแวดล้อมทั้งหมดมีความยุ่งยากมากขึ้นซึ่งหุ่นยนต์อาจจะต้องหมุนตัวไปมาเพื่อให้กล้องสามารถมองเห็นได้โดยรอบ สำหรับปัญหาเรื่องมุมมองที่แคบของกล้องวิดีโอนั้น เป็นปัญหาที่แก้ไขได้ไม่ยาก ซึ่งการจะเพิ่มมุมมองของกล้องนั้นอาจจะใช้กล้องหลายตัว เพื่อเพิ่มมุมมองของกล้องให้มองเห็นครอบคลุมมากขึ้น หรืออาจจะใช้กล้องลักษณะพิเศษซึ่งเรียกว่า "กล้องวิดีโอแบบออมนิ" ที่มีมุมมองรอบทิศทางมาใช้แทนก็ได้

(ก) (ข)

รูปที่ 1.1: (ก) ภาพกล้องวิดีโอแบบออมนิที่ใช้ในงานวิจัย (ข) ภาพถ่ายจากกล้องวิดีโอแบบออมนิ

กล้องวิดีโอแบบออมนิ เป็นกล้องที่มีลักษณะพิเศษคือ กล้องจะมีความสามารถในการมองเห็นได้รอบทิศทาง ลักษณะของตัวกล้องจะประกอบด้วยกล้องวิดีโอแบบธรรมดาและกระจกดังรูปที่ 1.1ก ซึ่งมีหน้าที่รับแสงจากรอบทิศทางและสะท้อนเข้าไปในตัวกล้อง จึงทำให้กล้องสามารถมองเห็นสิ่งแวดล้อมรอบทิศทางได้ ภาพที่ได้จากกล้องออมนิแสดงได้ดังรูปที่ 1.1ข ซึ่งจะเห็นได้ว่าภาพที่ได้จากกล้องเกิดจากการสะท้อนของกระจกทรงกลมทำให้มองเห็นภาพได้รอบทิศทาง ข้อดีของกล้องวิดีโอแบบออมนิ ก็คือตัวกล้องจะมีมุมมองรอบทิศจึงทำให้รับข้อมูลของสิ่งแวดล้อมได้มากกว่ากล้องวิดีโอแบบทั่วไป แต่ข้อเสียของกล้องวิดี

โอแบบออมนิ ก็คือ เนื่องจากตัวกล้องสามารถรับภาพที่มีมุมมองกว้าง ดังนั้นภาพข้อมูลสิ่งแวดล้อมที่วัดได้จากกล้องจึงมีรายละเอียดลดลงเมื่อเทียบกับกล้องวิดีโอแบบทั่วไปที่มีความละเอียดของภาพเท่ากัน และข้อเสียอีกประการหนึ่งก็คือภาพที่ได้จากกล้องออมนิจะนำไปคำนวนเพื่อการตีความได้ยากขึ้นเพราะรูปแบบการรับภาพของกล้องแตกต่างจากกล้องวิดีโอแบบทั่วไป และแตกต่างจากมุมมองของมนุษย์อีกด้วย

สำหรับวิทยานิพนธ์ฉบับนี้ มุ่งเน้นที่จะเสนอวิธีการในการแก้ปัญหาการระบุตำแหน่งพร้อมกับการสร้างแผนที่ด้วยกล้องวิดีโอแบบออมนิ โดยในงานวิจัยนี้สนใจที่แก้ปัญหาการระบุตำแหน่งของกล้องวิดีโอแบบออมนิในสามมิติ อีกทั้งจะสร้างแผนที่สิ่งแวดล้อม ที่กล้องวิดีโอแบบออมนิตรวจวัดได้ ในลักษณะสามมิติด้วย โดยที่กล้องวิดีโอแบบออมนิจะเคลื่อนที่ได้อย่างอิสระในสามมิติโดยไม่จำเป็นต้องรู้ข้อมูลควบคุมการเคลื่อนที่ของกล้อง และสภาพแวดล้อมที่งานวิจัยนี้สนใจ จะเป็นสภาพแวดล้อมแบบภายในอาคารโดยในการการระบุตำแหน่งพร้อมกับการสร้างแผนที่นั้น ระบบไม่จำเป็นต้องรู้ลักษณะของสิ่งแวดล้อมมาก่อนและระบบไม่จำเป็นต้องอาศัยข้อมูลล่วงหน้าใด ๆ ในการเริ่มต้นระบบ ซึ่งในงานวิจัยนี้ก็มีจุดมุ่งหมายว่าเมื่อนำระบบ "การระบุตำแหน่งพร้อมกับการสร้างแผนที่ด้วยกล้องวิดีโอแบบออมนิ" นี้ไปติดตั้งบนหุ่นยนต์หรืออุปกรณ์ใด ๆ ระบบนี้จะสามาถช่วยในการระบุตำแหน่งและสร้างแผนที่ให้กับหุ่นยนต์หรืออุปกรณ์นั้น ๆ

1.2 งานวิจัยที่เกี่ยวข้อง

วิธีการระบุตำแหน่งพร้อมกับการสร้างแผนที่ (Simultaneous Localization and Mapping: SLAM) ถูกกล่าวถึงเป็นครั้งแรกเมื่อปี ค.ศ. 1986 โดย Smith และ Cheeseman (Smith and Cheeseman, 1986) ซึ่งได้นำเสนอวิธีในการประมาณความสัมพันธ์ และค่าความแปรปรวนร่วม (error covariance) ระหว่างโคออดิเนตเฟรมทั้งหลาย ซึ่งเป็นตัวแทนของวัตถุต่าง ๆ ด้วยวิธีการทางความน่าจะเป็น (probabilistic methods) และหลังจากนั้นไม่นาน Moutarlier และ Chatila (Moutarlier and Chatila, 1989; Chatila and Moutarlier, 1989) ก็ได้พัฒนาวิธีแก้ปัญหาสำหรับ SLAM ขึ้นครั้งแรกโดยใช้ Extend Kalman Filter (EKF) ในการประมาณค่าตำแหน่งของหุ่นยนต์และแผนที่ ซึ่งภายหลัง Extended Kalman Filter ก็ถูกใช้ในการแก้ปัญหา SLAM อย่างกว้างขวาง อย่างไรก็ตาม ก็ยังได้มีการพัฒนาอัลกอริทึมอื่น ๆ อีกมากมายสำหรับ SLAM เช่น FastSLAM (Hahnel et al., 2003; Montemerlo et al., 2003), UKF-SLAM (Wang and Zhang, 2007), SEIF-SLAM (Walter et al., 2007) ฯลฯ

SLAM นั้นถูกนำไปใช้อย่างแพร่หลาย ในงานนำร่องของหุ่นยนต์ ในสภาพแวดล้อมหลายรูปแบบ เช่น สภาพแวดล้อมในอาคาร(Bosse et al., 2002; Fox et al., 1998), สภาพแวดล้อมนอกอาคาร (Bailey, 2002), ใต้น้ำ (Williams et al., 2001), ใต้ดิน (Thrun et al., 2003; Scheding et al., 1997) หรือแม้กระทั่ง การสำรวจดาวเคราะห์ (LI et al., 2000; Uhlmann et al., 1999) นอกจากนี้ยังมีการใช้งานกับอุปกรณ์เซนเซอร์ที่แตกต่างกันไป ตัวอย่างเช่น กล้องวิดีโอแบบทั่วไป (Davison et al., 2007), กล้องแบบออมนิ (Murillo et al., 2006), อุปกรณ์วัดระยะด้วยเลเซอร์ (Laser Range Finder) (Diosi and Kleeman, 2005), เรดาร์ (Radar), โซน่าร์ (Sonar), GPS และอื่น ๆ

สำหรับเซนเซอร์ต่าง ๆ สำหรับงานระบุตำแหน่งพร้อมกับการสร้างแผนที่ (SLAM) กล้องวิดีโอ ถือเป็นอุปกรณ์หนึ่งที่มีการใช้ในงานวิจัยต่าง ๆ เป็นจำนวนมาก (Jeong and Lee, 2006; Gemeiner et al., 2007) ทั้งนี้เนื่องมาจากข้อดีของกล้องวิดีโอ คือ มีราคาถูก ใช้งานง่าย ให้ข้อมูลเป็นจำนวนมาก อีกทั้งมีความเร็วในการรับข้อมูลสูงและทันการณ์ (Real-time) แต่ปัญหาสำหรับกล้องวิดีโอ คือ ข้อมูลที่ได้มาจากกล้องนั้นเป็นภาพในสองมิติ ซึ่งเกิดจากการ projection สิ่งแวดล้อมในสามมิติลงมาบนภาพ จึงทำให้มีข้อมูลที่สูญหายไป ซึ่งก็คือข้อมูลระยะทางจากกล้องถึงวัตถุในนั้น ดังนั้นการสร้างแผนที่สำหรับสิ่งแวดล้อมซึ่งต้องอาศัยข้อมูลระยะจึงมีความยุ่งยากขึ้น ในงานวิจัยหลาย ๆ งานจึงมีเทคนิคในการช่วยหาข้อมูลระยะ เช่น การใช้กล้องสเตอริโอ (stereo camera) ช่วยในการหาระยะ (Han et al., 2007), การใช้กล้องวิดีโอ ร่วมกับเซนเซอร์วัดระยะอื่น (Fu et al., 2007), การประมาณข้อมูลระยะทางของสิ่งแวดล้อมจากการเคลื่อนที่ของกล้อง โดยตรวจวัดข้อมูลการเคลื่อนที่จาก อุปกรณ์ตรวจจับการเคลื่อนที่จากความเฉื่อย (iner-

tia sensors) หรือ อุปกรณ์ตรวจวัดความเร็วจากล้อ (wheel encoders) หรือ GPS หรืออื่น ๆ ประกอบ แต่ถึงกระนั้นก็ยังมีงานวิจัยที่อาศัยข้อมูลรูปภาพจากกล้องเพียงอย่างเดียวในการระบุตำแหน่งพร้อมกับการสร้างแผนที่ (SLAM)

MonoSLAM (Davison et al., 2007) ซึ่งถูกนำเสนอโดย Davison เป็นงานหนึ่งที่มีความโดดเด่นสำหรับงานระบุตำแหน่งพร้อมกับการสร้างแผนที่ด้วยกล้องวิดีโอ (Vision-based SLAM) โดย Davison นั้นได้เสนออัลกอริทึมแบบ Real-time ที่ใช้การหาเส้นทางการเคลื่อนที่ในสามมิติของกล้องตัวเดี่ยว ที่เคลื่อนที่ในสิ่งแวดล้อมที่ไม่คุ้นเคยมาก่อน สำหรับ MonoSLAM นั้น ถือได้ว่าเป็นงานแรก ๆ ที่ประสบความสำเร็จสำหรับ Vision- based SLAM ที่อาศัยข้อมูลรูปภาพจากกล้องเพียงอย่างเดียว โดยไม่อาศัยข้อมูลการเคลื่อนที่ของกล้องจากเซนเซอร์อื่น หลังจากนั้นก็มีงานที่คล้ายคลึงกับ MonoSLAM อีกมากมายเช่น Eade และ Drummond (Eade and Drummond, 2006) ได้เสนอ SLAM โดยใช้กล้องตัวเดียวร่วมกับ FastSLAM หรือ Sunderhauf (Sunderhauf et al., 2007) ก็ได้เสนอ SLAM สำหรับ เครื่องบินไร้คนขับ (UAV) โดยใช้กล้องตัวเดียวและใช้ Unscented Kalman Filter (UKF) ในการแก้ปัญหา SLAM

ถึงแม้ว่า MonoSLAM จะทำงานได้ดีในสิ่งแวดล้อมที่ไม่มีโครงสร้างแน่นอน แต่เนื่องจากมุมมองของกล้อง (field of view) ที่แคบจึงทำให้ MonoSLAM ทำงานได้ไม่ดีนักกับการเคลื่อนที่ ที่เปลี่ยนมุมมองเร็วจนเกินไปจนไม่สามารถสร้างแผนที่ได้ทัน และ MonoSLAM ยังไม่สามารถทำงานได้กับการเคลื่อนที่ของกล้องบางรูปแบบอีกด้วย

ในปีค.ศ. 2003 Kim และ Chung (Kim and Chung, 2003) ได้เสนอวิธีการใช้งานกล้องออมนิ กับงานระบุตำแหน่งพร้อมกับการสร้างแผนที่ (SLAM) โดยเป็นการทำงานร่วมกันระหว่าง กล้องออมนิแบบสเตอริโอ (Omni-directional stereo vision) กับ การหาโครงสร้างสามมิติจากการเคลื่อนที่ (structure from motion: SFM) และเนื่องจาก กล้องออมนิ เป็นเซนเซอร์ที่มีมุมมองของภาพที่กว้าง จึงทำให้เซนเซอร์ตรวจวัดสิ่งแวดล้อมได้ทั่วถึง ส่งผลให้อัลกอริทึมของ SLAM มีความเสถียร และทนทานต่อการเคลื่อนที่ของกล้อง ภายหลังการใช้งานกล้องออมนิ สำหรับงานระบุตำแหน่งพร้อมกับการสร้างแผนที่ จึงเริ่มมีการแพร่หลายมากขึ้น Murillo, Guerrero และ กลุ่ม (Murillo et al., 2006) ได้เสนออัลกอริทึมสำหรับกล้องออมนิ ในการระบุตำแหน่งของหุ่นยนต์ในสองมิติ โดยใช้แนวเส้นตั้งกับกล้องเป็นจุดสังเกต และใช้เส้นตรงในสามมุมมองเพื่อคำนวณ Trifocal Tensor จากนั้นจึงใช้ tensor เพื่อการระบุตำแหน่งของหุ่นยนต์ Valgren, Lilienthal และ กลุ่ม (Valgren et al., 2006) ได้เสนออัลกอริทึมสำหรับสร้างแผนที่แบบโครงสร้าง (topological maps) โดยใช้ข้อมูลจากกล้องกล้องออมนิ เพียงอย่างเดียว Saedan, Lim และ กลุ่ม (Saedan et al., 2007) ได้เสนอวิธีในการระบุตำแหน่งโดยใช้ Appearance-based ซึ่งมีความสามารถในการสร้างแผนที่ขนาดใหญ่ และสามารถตรวจจับ สถานที่เดิมที่เคยเข้าถึงมาก่อน นอกจากนี้ยังมี งานวิจัยที่เกี่ยวข้องกับการระบุตำแหน่งพร้อมกับการสร้างแผนที่ (SLAM) โดยใช้กล้องออมนิ อีกมากมาย (Tamimi et al., 2005; Kim et al., 2006; Andreasson et al., 2005)

จากงานวิจัยที่กล่าวมาข้างต้น จะเห็นได้ว่า มีการใช้กล้องวิดีโอในงานระบุตำแหน่งของหุ่นยนต์พร้อมกับการสร้างแผนที่ (SLAM) อย่างแพร่หลายมาก แต่เนื่องจากปัญหามุมมองของกล้องที่แคบ ทำให้อัลกอริทึมของกล้องวิดีโอ ไม่ทนทานต่อการเคลื่อนที่มากนัก ดังนั้นการใช้กล้องมุมมองกว้าง อย่างเช่น กล้องออมนิ จึงเป็นทางออกที่ดีสำหรับ SLAM อย่างไรก็ตาม การพัฒนาอัลกอริทึมสำหรับกล้องออมนิ ยังยึดติดกับการระบุตำแหน่งของหุ่นยนต์ในสองมิติอยู่มาก อีกทั้งเนื่องจากกล้องออมนิมีความละเอียดของภาพที่ต่ำ จึงทำให้การทำงานของอัลกอริทึมทั้งหลายที่ถูกนำเสนอไม่ดีเท่าที่ควร วิทยานิพนธ์นี้จึงมีเป้าหมายที่จะพัฒนาอัลกอริทึมสำหรับระบุตำแหน่งและหาเส้นทางการเคลื่อนที่ของกล้อง พร้อมไปกับการสร้างแผนที่ของสิ่งแวดล้อม โดยทั้งเส้นทางการเคลื่อนที่ของกล้อง และแผนที่ของสิ่งแวดล้อมที่สร้างได้นั้นจะแสดงในสามมิติ โดยกล้องออมนิสามารถเคลื่อนที่ไปมาได้อย่างอิสระ โดยที่กล้องออมนิ อาจถูกติดตั้งไปกับหุ่นยนต์หรืออาจถูกจับเคลื่อนที่ไปมาโดยมนุษย์ ซึ่งการเคลื่อนที่ของกล้องออมนิจะเป็นแบบไม่สามารถประมาณการเคลื่อนที่ได้ และสำหรับอัลกอริทึมการระบุตำแหน่งพร้อมกับการสร้างแผนที่นั้น จะอาศัยข้อมูลเป็นภาพจาก

กล้องออมนิเพียงอย่างเดียว ไม่อาศัยข้อมูลจากเซนเซอร์อื่น ๆ ร่วมอีก โดยมีจุดมุ่งหมายว่า อัลกอริทึมจะสามารถทำงานได้ในสภาพแวดล้อมภายในห้อง และสามารถนำไปใช้งานร่วมกับงานอื่น ๆ ต่อไป

1.3 การนำเสนอและลำดับเนื้อหาวิทยานิพนธ์

การนำเสนอเนื้อหาวิทยานิพนธ์จะแบ่งออกเป็นส่วนย่อย ๆ 3 ส่วนหลักคือ ส่วนทฤษฎีที่เกี่ยวข้อง, ส่วนขั้นตอนการแก้ปัญหาของงานวิจัย และ ส่วนทดสอบการทำงานของอัลกอริทึมและสรุปผล

1. ส่วนทฤษฎีที่เกี่ยวข้อง จะอธิบายถึงทฤษฎีและหลักการทั่วไปที่ใช้ในการแก้ปัญหา การระบุตำแหน่งพร้อมกับการสร้างแผนที่ด้วยกล้องวิดีโอแบบออมนิ ซึ่งในส่วนเนื้อหานั้นจะประกอบด้วย

   บทที่ 2 จะอธิบายวิธีการในการประมาณค่าแบบต่าง ๆ เช่น Least Squares, Kalman Filter ส่วนสำคัญที่ใช้ในการวิเคราะห์ข้อมูลที่ได้จาก เซนเซอร์เพื่อให้หุ่นยนต์สามารถเข้าใจลักษณะของสภาพแวดล้อมได้

   บทที่ 3 จะกล่าวถึงหลักการของการระบุตำแหน่งพร้อมกับการสร้างแผนที่ ซึ่งวิธีการในการแก้ปัญหาการระบุตำแหน่งพร้อมกับการสร้างแผนที่แบบทั่วไปนั้น ได้มีคนเสนอวิธีการแก้ปัญหาไว้หลายรูปแบบ ไม่ว่าจะเป็น EKF SLAM, UKF SLAM, Fast SLAM ซึ่งในงานวิจัยนี้จะใช้ EKF SLAM ในการแก้ปัญหาเป็นหลัก ในบทนี้จึงจะอธิบายวิธีการของ EKF SLAM เป็นพิเศษ

   บทที่ 4 จะกล่าวถึงทฤษฎีที่เกี่ยวกับกล้องวิดีโอแบบออมนิ ซึ่งเป็นอุปกรณ์สำคัญในการรับภาพเพื่อใช้เป็นข้อมูลในการระบุตำแหน่งและสร้างแผนที่ สำหรับงานวิจัยนี้ โดยทฤษฎีที่สำคัญสำหรับกล้องออมนินั้น ก็คือการหาโมเดลของกล้องออมนิ เพื่อที่จะได้สามารถโปรเจคตำแหน่งของวัตถุในสามมิติลงบนภาพได้อย่างถูกต้อง

2. ส่วนขั้นตอนการแก้ปัญหาของงานวิจัย จะอธิบายในบทที่ 5 ซึ่งจะประกอบด้วยวิธีในการวิเคราะห์ภาพที่ได้จากกล้องวิดีโอแบบออมนิ และ การนำข้อมูลการวัดที่ได้จากล้องวิดีโอแบบออมนิไปใช้ใน EKF SLAM

3. ส่วนทดสอบการทำงานของอัลกอริทึมและสรุปผล จะแสดงผลการทดลองการระบุตำแหน่งพร้อมกับการสร้างแผนที่ในสามมิติ ในบทที่ 6 โดยจะนำเอากล้องวิดีโอแบบออมนิไปเก็บข้อมูลในสภาพแวดล้อมจริง และแสดงผลลัพธ์ที่ได้จากการทำงานของอัลกอริทึมที่ได้นำเสนอ นอกจากนี้ยังได้เปรียบเทียบผลลัพธ์การระบุตำแหน่งและสร้างแผนที่กับข้อมูล ground truth เพื่อหาความคาดเคลื่อนของอัลกอริทึมที่ได้นำเสนอ และสุดท้ายในบทที่ 7 ได้สรุปผลการวิจัยชิ้นนี้ และแนวทางในการวิจัยขั้นถัดไป

บทที่ 2

การประมาณค่า

ในโลกแห่งความเป็นจริงนั้น มนุษย์ไม่เคยรู้ค่าความจริงใด ๆ เลยเป็นต้นว่า มนุษย์ไม่เคยรู้เลยว่าแท้จริงแล้ว ตนเองนั้นมีความสูงเท่าไหร่ หรือ วัตถุหนึ่ง ๆ มีน้ำหนักเท่าไหร่ สิ่งที่พอจะรู้ได้ ก็คือค่าที่ได้จากการวัดด้วยอุปกรณ์ที่ใช้วัดค่าที่มนุษย์สร้างขึ้นมาเอง เช่น มนุษย์วัดความสูงด้วยตลับเมตร หรือวัดน้ำหนักด้วยตาชั่ง แต่ค่าที่ได้จากการวัดนั้นก็ย่อมมีความคลาดเคลื่อน ซึ่งเกิดขึ้นได้จากอุปกรณ์การวัดที่ไม่แม่นยำ และ ความคลาดเคลื่อนจากตัวผู้วัดเอง ดังนั้นค่าที่ได้จากการวัดนั้นย่อมไม่ใช่ค่าจริง สิ่งที่มนุษย์พอจะทำได้ก็คือประมาณค่าที่คิดว่าใกล้เคียงกับค่าจริงมากที่สุดโดยอาศัยข้อมูลจากการวัด ซึ่งวิธีการประมาณค่าความจริงให้ใกล้เคียงที่สุดนั้น วิธีง่ายที่สุดก็คือ การวัดหลาย ๆ ครั้งแล้วหาค่าเฉลี่ย ซึ่งถ้าหากว่าความคลาดเคลื่อนของการวัดมีการกระจายแบบ Uniform แล้ว ค่าเฉลี่ยก็จะเป็นค่าที่มีความใกล้เคียงค่าจริงมากที่สุด ซึ่งในหลาย ๆ กรณีแล้วค่าเฉลี่ยอาจจะไม่ใกล้เคียงกับค่าจริงก็ได้ เช่น การประมาณตำแหน่งของรถซึ่งติดตั้งอุปกรณ์ GPS คันหนึ่ง ที่วิ่งอยู่บนถนนด้วยความเร็วคงที่ ในความเป็นจริงแล้วมนุษย์ไม่มีทางรู้ตำแหน่งที่แท้จริง (ระบบพิกัดภูมิศาสตร์ แบบละติจูด (Latitude) และ ลองติจูด (Latitude)) ของรถคันนั้นได้เลย แต่มนุษย์พอจะวัดตำแหน่งของรถได้ด้วยอุปกรณ์ GPS ที่ติดตั้งอยู่บนรถ ซึ่งข้อมูลการวัดที่ได้จากอุปกรณ์ GPS นั้นก็มีความคลาดเคลื่อนตั้งแต่ 1 - 10 เมตร ดังนั้นหากจะประมาณตำแหน่งของรถด้วยข้อมูลการวัดที่ได้จากอุปกรณ์ GPS โดยตรงนั้นคงจะไม่ถูกต้องนัก และถ้าจะประมาณตำแหน่งของรถด้วยค่าเฉลี่ยข้อมูลการวัดจากอุปกรณ์ GPS ย่อมเป็นไปได้ยาก เพราะรถที่ต้องการจะประมาณตำแหน่งนั้นวิ่งอยู่ตลอดเวลา ดังนั้นจึงต้องใช้อัลกอริทึมวิธีที่เหมาะสมในการประมาณค่า

2.1 การประมาณด้วยวิธี Least Squares

Least Squares เป็นวิธีที่ใช้ในการแก้ปัญหาระบบสมการ ที่มีจำนวนสมการมากกว่าจำนวนตัวแปรไม่ทราบค่า (over-determined systems) (Bjorck, 1996) ซึ่งผลจากการที่มีจำนวนสมการมากกว่าจำนวนตัวแปรไม่ทราบค่า อาจทำให้ไม่สามารถหาค่าที่แท้จริงของตัวแปรไม่ทราบค่าได้ เช่น สำหรับสมการ $y_i = f_i(x_i, \beta)$ เมื่อ x_i เป็นตัวแปรต้น , y_i เป็รตัวแปรตาม และ β เป็นตัวแปรไม่ทราบค่า เมื่อมีคู่ข้อมูล $(x_i, y_i) i = 1, ..., n$ หลายคู่ อาจจะไม่สามารถหาค่า β รวมกันที่เหมาะสมสำหรับทุกคู่ข้อมูลได้ สาเหตุเนื่องมาจาก การประมาณฟังก์ชัน $f_i(x_i, \beta)$ อาจไม่ดีพอหรือการวัดค่า y_i มีความคลาดเคลื่อน ดังนั้นสมการที่เหมาะสมกว่าควรจะเป็น

$$y_i = f_i(x_i, \beta) + \epsilon_i \tag{2.1}$$

เมื่อ ϵ_i เป็นความคลาดเคลื่อนที่อาจจะเกิดขึ้นได้ ซึ่งวิธีการของ Least Squares นั้นก็คือการประมาณค่า β ที่ทำให้ผลรวมของความคลาดเคลื่อนกำลังสอง (ϵ_i^2) มีค่าน้อยที่สุด จะเขียนได้ว่า

$$\hat{\beta} = arg\min_{\beta}(\sum_{i=1}^{n} \epsilon_i^2), \qquad \epsilon_i = y_i - f(x_i, \beta) \tag{2.2}$$

และเมื่อเขียนรวมสมการทุกคู่ข้อมูลจะได้ว่า

$$\hat{\beta} = arg\min_{\beta}(\|\varepsilon\|^2), \qquad \varepsilon = Y - F(X, \beta) \tag{2.3}$$

โดยที่ $\varepsilon = [\epsilon_1, \epsilon_2, ..., \epsilon_n]^T$, $Y = [y_1, y_2, ..., y_n]^T$ และ $F(X, \beta) = [f(x_1, \beta), ..., f(x_n, \beta)]^T$

ผลลัพธ์ของ ผลรวมผลต่างกำลังสองน้อยสุด สามารถหาได้จากการแก้สมการ gradient ของ ผลรวมผลต่างกำลังสอง เทียบ β เป็น 0 จะได้ว่า

$$\frac{\partial \|\varepsilon\|^2}{\partial \beta} = \frac{\partial \|Y - F(X, \beta)\|^2}{\partial \beta} = 0 \tag{2.4}$$

ซึ่งเมื่อแก้สมการนี้ก็จะสามารถหาค่า β ได้

สำหรับการประมาณค่าสถานะของระบบด้วยวิธี Least Squares นั้น β ก็คือค่าสถานะที่ต้องการประมาณ, y_i เป็นค่าที่ได้จากการวัด, $f_i(x_i, \beta) + \epsilon_i$ เป็น โมเดลการวัด (Measurement model) ส่วน x_i เป็นพารามิเตอร์ของการวัด ซึ่งจะมีหรือไม่ก็ได้ ยกตัวอย่างเช่น

ต้องการประมาณตำแหน่งของรถซึ่งติดตั้งอุปกรณ์ GPS คันหนึ่ง ที่วิ่งอยู่บนถนนด้วยความเร็วคงที่ โมเดลการวัด (Measurement model) สามารถเขียนได้เป็น

$$s_i = s_0 + vt_i + \epsilon_i$$

เมื่อ s_i เป็นตำแหน่งของรถที่วัดได้โดย GPS ที่เวลา t_i, s_0 เป็นตำแหน่งของรถที่เวลา t_0, v เป็นความเร็วของรถซึ่งในที่นี้ได้กำหนดให้รถมีความเร็วคงที่ ส่วน ϵ_i เป็นค่าความคลาดเคลื่อนที่เกิดขึ้นจากการวัด

ดังนั้นจะได้ว่า t_i เป็นพารามิเตอร์ของการวัด (x_i) ส่วน $\hat{s}_0$ และ $\hat{v}$ ก็คือ ค่าสถานะที่ต้องการประมาณ ($\hat{\beta}$) ซึ่งตำแหน่งของรถที่ต้องการประมาณ ($\hat{s}_i$) นั้นจะหาได้จาก โมเดลการวัด และ ค่าประมาณของสถานะของระบบ ($\hat{s}_0$ และ$\hat{v}$) อีกที

2.1.1 Linear Least Squares

Linear Least Squares ก็คือการใช้ Least Squares มาเพื่อแก้ระบบสมการที่เป็นแบบ linear ซึ่งก็คือสมการสามารถเขียนได้ในรูป

$$f_i(x_i, \beta) = \sum_{j=1}^{m} \beta_j \phi_j(x_i), \qquad j = 1, ..., m \tag{2.5}$$

เมื่อ m เป็นจำนวนพารามิเตอร์ของ β และ$\phi_j(x_i)$ เป็นฟังก์ชันของ x_i

เมื่อเขียนรวมสมการทุกคู่ข้อมูล และกำหนดให้ $X_{ij} = \phi_j(x_i)$ เป็นพารามิเตอร์ แถวที่ i คอลัมน์ที่ j ของ Matrix X จะได้ว่า

$$F(X, \beta) = X\beta \tag{2.6}$$

เมื่อแทนค่าในสมการ 2.4 จะได้ว่า

$$\frac{\partial \|Y - X\beta\|^2}{\partial \beta} = 0 \tag{2.7}$$

$$\frac{\partial ((Y - X\beta)^T (Y - X\beta))}{\partial \beta} = 0 \tag{2.8}$$

$$\frac{\partial(Y^TY - Y^TX\beta - \beta^TX^TY + \beta^TX^TX\beta)}{\partial\beta} = 0 \quad (2.9)$$

$$-2X^TY + 2X^TX\beta = 0 \quad (2.10)$$

$$\beta = (X^TX)^{-1}X^TY \quad (2.11)$$

สำหรับตัวอย่างที่แล้วที่ต้องการประมาณตำแหน่งของรถซึ่งติดตั้งอุปกรณ์ GPS คันหนึ่ง ที่วิ่งอยู่บนถนนด้วยความเร็วคงที่ ซึ่งระบบที่ใช้ในการประมาณนั้นเป็นแบบ Linear ดังนั้นจะเขียนฟังก์ชัน $F(X,\beta)$ ได้เป็น

$$F(X,\beta) = X\beta = \begin{bmatrix} 1 & t_1 \\ \vdots & \vdots \\ 1 & t_n \end{bmatrix} \begin{bmatrix} s_0 \\ v \end{bmatrix}$$

และ $Y = \begin{bmatrix} s_1 & \dots & s_n \end{bmatrix}^T$

ดังนั้นเมื่อแก้สมการ $\beta = (X^TX)^{-1}X^TY$ แล้ว ก็จะสามารถคำนวนค่า $\hat{s}_0$ และ $\hat{v}$ ได้

2.1.2 Non-Linear Least Squares

Non-Linear Least Squares คือการหา Least Squares ของระบบสมการที่ไม่ได้เป็น Linear ซึ่งคำตอบของการประมาณนั้นจะไม่มี closed-form solution แต่ในการหาคำตอบนั้นจะใช้วิธีการทาง Numerical มาใช้ในการหาคำตอบแทนซึ่งพารามิเตอร์ที่ต้องการประมาณจะถูกปรับแก้ค่าแบบวนซ้ำไปเรื่อย ๆ จนกว่าจะมี error ของพารามิเตอร์เป็นที่น่าพอใจ จะได้ว่า

$$\beta(k+1) = \beta^k + \Delta\beta \quad (2.12)$$

เมื่อ k เป็นจำนวนรอบการ iterate และ model function สามารถประมาณให้เป็น linear ได้จากการกระจาย first order Taylor series เป็น

$$F(X,\beta^{k+1}) = F(X,\beta^k) + \frac{\partial F(X,\beta^k)}{\partial\beta^k}(\beta^{k+1} - \beta^k) = F(X,\beta^k) + J_F(X,\beta^k)\Delta\beta \quad (2.13)$$

เมื่อ $J_F(\beta^k)$ เป็น Jacobian Matrix ของ $F(X,\beta^k)$ เทียบกับ β^k

$$J_F(X,\beta^k) = \begin{bmatrix} \frac{\partial F}{\partial\beta_1^k} & \dots & \frac{\partial F}{\partial\beta_m^k} \end{bmatrix} \quad (2.14)$$

เมื่อแทนค่าในสมการ 2.4 จะได้ว่า

$$\frac{\partial \left\| Y - (F(X,\beta^k) + J_F\Delta\beta) \right\|^2}{\partial\beta} = 0 \quad (2.15)$$

$$\frac{\partial \left\| \Delta Y - J_F\Delta\beta \right\|^2}{\partial\beta} = 0 \quad (2.16)$$

$$\frac{\partial((\Delta Y - J_F\Delta\beta)^T(\Delta Y - J_F\Delta\beta))}{\partial\beta} = 0 \quad (2.17)$$

$$\frac{\partial(\Delta Y^T\Delta Y - \Delta Y^T J_F\Delta\beta - \Delta\beta^T J_F^T\Delta Y + \Delta\beta^T J_F^T J_F\Delta\beta)}{\partial\beta} = 0 \quad (2.18)$$

$$-2J_F^T\Delta Y + 2J_F^T J_F\Delta\beta = 0 \quad (2.19)$$

$$\Delta\beta = (J_F^T J_F)^{-1}J_F^T\Delta Y \quad (2.20)$$

จากนั้นจะทำงานวนซ้ำเพื่อปรับแก้ค่า β ไปเรื่อย ๆ ตามสมการ 2.12 จนกว่า error จะยอมรับได้

2.1.3 Weighted Least Squares

จากการประมาณ Least Square ข้างต้น อยู่บนสมมุติฐานที่ว่าข้อมูลทุก ๆ คู่ข้อมูล (x_i, y_i) มีความน่าเชื่อถือเท่า ๆ กัน แต่ในความเป็นจริงแล้วข้อมูลในแต่ละคู่ อาจจะมีความคลาดเคลื่อนไม่เท่ากันก็ได้ ดังนั้นเมื่อกำหนดให้ ความแปรปรวนร่วมของข้อมูลทุกค่า (Covariance) เป็น R การประมาณค่า β ที่ทำให้ผลรวมของความคลาดเคลื่อนกำลังสอง (ϵ_i^2) มีค่าน้อยที่สุด จะเขียนใหม่เป็น

$$\hat{\beta} = arg\min_{\beta} \left\| R^{-1}(Y - F(X,\beta)) \right\|^2 \quad (2.21)$$

ดังนั้นค่าประมาณของ β สำหรับ Linear Least Squares จะเขียนได้ใหม่เป็น

$$\beta = (X^T R^{-1} X)^{-1} X^T R^{-1} Y \quad (2.22)$$

ส่วนค่าประมาณของ β สำหรับ Non-Linear Least Squares จะเขียนได้ใหม่เป็น

$$\beta = (J_F^T R^{-1} J_F)^{-1} J_F^T R^{-1}\Delta Y \quad (2.23)$$

ซึ่งถ้าหากว่าความคลาดเคลื่อนข้อมูลในแต่ละคู่ข้อมูลเป็นอิสระต่อกัน R จะเขียน ได้เป็น

$$\begin{bmatrix} \sigma_{y1}^2 & 0 & \cdots \\ 0 & \sigma_{y2}^2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \quad (2.24)$$

Least Squares นั้น ถือเป็นวิธีการในการประมาณค่าที่ดีที่สุดอย่างหนึ่ง เนื่องจาก Least Squares สามารถรับประกันได้ว่า ผลรวมความคลาดเคลื่อนที่เกิดจากการประมาณยกกำลังสองนั้นจะน้อยที่สุด แต่ข้อเสียของ Least Squares ก็คือ เมื่อนำ Least Square มาใช้กับระบบแบบ dynamic (ระบบที่มีการเปลี่ยนแปลงขึ้นอยู่กับเวลา) ยกตัวอย่างเช่น การประมาณตำแหน่งของรถซึ่งติดตั้งอุปกรณ์ GPS คันหนึ่งที่วิ่งอยู่บนถนนด้วยความเร็วคงที่ ดังตัวอย่างก่อนหน้านี้ จะเห็นได้ว่า เมื่อระบบดำเนินไปเป็นเวลานาน ระบบสมการที่ใช้ในการประมาณ จะมีจำนวนสมการมากขึ้นเรื่อย ๆ ขึ้นอยู่กับเวลาที่ใช้ ส่งผลให้เสียเวลาในการประมวลผลมากขึ้น พูดได้ว่าประสิทธิภาพในการคำนวนของ Least Squares เป็น $O(n^2)$ เมื่อ n เป็นปริมาณเวลาที่ดำเนินไป จะเห็นได้ว่าอัลกอริทึมของ Least Squares นั้นไม่เหมาะสมเป็นอย่างมากสำหรับนำมาใช้ประมาณระบบที่ดำเนินต่อไปได้เรื่อย ๆ ไม่สิ้นสุด ซึ่งอัลกอริทึมในการประมาณสถานะของระบบแบบ dynamic ที่เหมาะสมกว่าก็คือ Kalman filter

2.2 Kalman filter

kalman filter เป็นอัลกอริทึมที่ดีวิธีหนึ่ง สำหรับการประมาณสถานะของระบบแบบ dynamic (ระบบที่มีการเปลี่ยนแปลงขึ้นอยู่กับเวลา) จากข้อมูลการวัดที่มีความคลาดเคลื่อน (Kalman, 1960) โดย kalman filter นั้นมีจุดมุ่งหมายเพื่อที่จะลดค่าความแปรปรวนรวมของความผิดพลาดในการประมาณค่า (Error Covariance) ของระบบให้ได้มากที่สุด

kalman filter นั้น แท้จริงแล้วก็คือ Least Squares ประเภทหนึ่งที่มีการกำหนดข้อบังคับเพิ่มเติมเพื่อให้เหมาะสมกับระบบแบบ dynamic โดย kalman filter นั้นจะมีสมมุติฐานอยู่ว่า สถานะของระบบที่ต้องการประมาณค่านั้น จะมีการเปลี่ยนแปลงไปเรื่อย ๆ ตามเวลา โดยสถานะของระบบ ณ เวลาปัจจุบันนั้น จะขึ้นอยู่กับสถานะของระบบ ณ เวลาก่อนหน้าเท่านั้น ซึ่ง kalman filter จะมีรูปแบบที่คล้ายคลึงกับ hidden Markov model มาก แตกต่างกันตรงที่ สถานะของระบบของ hidden Markov model จะอยู่ใน discrete space เท่านั้น ส่วนสถานะของระบบของ kalman filter จะอยู่ใน continuous space ความต่างอีกประการหนึ่งก็คือ hidden Markov model นั้นสามารถกำหนดการกระจายความน่าจะเป็นของสถานะถัดไปได้ตามใจชอบ ส่วน kalman filter นั้นการกระจายน่าจะเป็นของสถานะถัดไปจะขึ้นกับ Gaussian noise model

การทำงานของ kalman filter

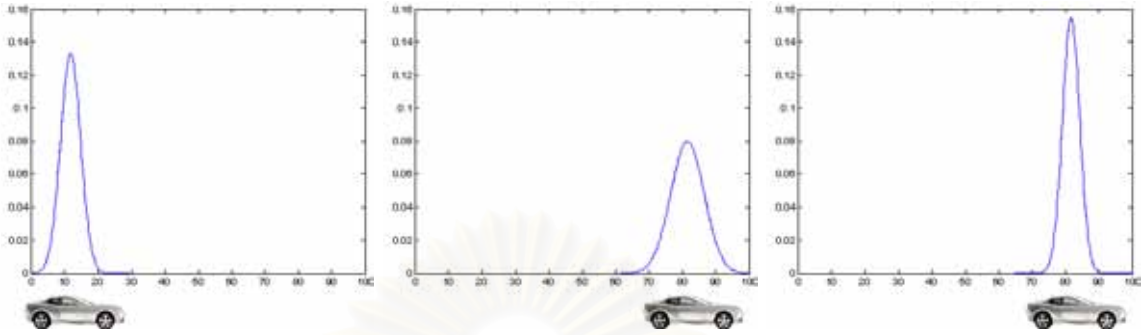
สำหรับระบบแบบ dynamic ระบบหนึ่งที่ต้องการประมาณค่าสถานะของระบบ ด้วย kalman filter ในที่นี้จะกำหนดให้ x_k เป็นสถานะของระบบที่ต้องการประมาณที่เวลา k และนอกจากนี้ kalman filter ยังต้องประมาณ การกระจายความน่าจะเป็นของสถานะของระบบ P_k ซึ่งในที่นี้ kalman filter ได้ประมาณการกระจายความน่าจะเป็น ว่ามีการกระจายแบบ Gaussian ซึ่งมี mean เป็น x_k ดังนั้น P_k จึงจะแทน ความแปรปรวนร่วมของสถานะของระบบ (Covariance)

การทำงานของ Kalman Filter นั้นจะประกอบไปด้วยสองขั้นตอนทำงานวนซ้ำกันไปเรื่อย ๆ คือ Prediction และ Update measurement ซึ่งจะประมาณและการปรับแก้ค่าสถานะของระบบ ($\hat{x}_k$) และความแปรปรวนร่วมของสถานะของระบบ (P_k) ไปเรื่อย ๆ โดยมุ่งเน้นให้ความแปรปรวนร่วม (P_k) มีค่าต่ำสุด ยกตัวอย่างเช่น

สำหรับการประมาณตำแหน่งของรถซึ่งติดตั้งอุปกรณ์ GPS คันหนึ่ง ที่วิ่งอยู่บนถนนด้วยความเร็วคงที่

ในตอนเริ่มต้นไม่รู้ตำแหน่งเริ่มต้นของรถที่แน่นอนจึงประมาณตำแหน่งเริ่มต้นของรถ ($\hat{x}_0$) ด้วย ค่าจาก GPS ณ ขณะนั้น และกำหนดให้ ความแปรปรวนร่วมของตำแหน่งเริ่มต้น (P_0) เท่ากับ error ของ GPS ซึ่งจะได้การกระจายความน่าจะเป็นของตำแหน่งของรถ ดังรูปที่ 2.1ก ต่อมาเมื่อรถมีการเคลื่อนที่และ GPS ตรวจวัดตำแหน่งของรถที่เคลื่อนที่ไปได้ ก็จะนำค่า GPS ที่ได้มาช่วยปรับแก้ตำแหน่งของรถ ที่ประมาณไว้ตอนเริ่มต้น แต่ก่อนที่จะนำค่าการวัดจาก GPS มาปรับแก้ จะต้องพึงระลึกไว้ว่าตำแหน่งของรถที่ประมาณไว้ตอนเริ่มต้น นั้นกลายเป็นอดีตไปเสียแล้ว เพราะรถมีการเคลื่อนที่ไปเรื่อย ๆ ดังนั้นก่อนที่จะนำค่าการวัดมาปรับแก้ จะต้องทำนายการเคลื่อนที่ของรถเสียก่อน (Prediction) ด้วยโมเดลการเคลื่อนที่ของรถ (ในที่นี้สมมุติว่ารู้โมเดลการเคลื่อนที่ของรถอยู่แล้ว) ซึ่งในการทำนายการเคลื่อนที่นั้น โมเดลการเคลื่อนที่ที่ใช้ในการประมาณย่อมมีความคลาดเคลื่อนเกิดขึ้นได้ อาจเป็นเพราะรถวิ่งแล้วล้อไถล หรือเครื่องยนต์ทำงานได้ไม่แม่นยำพอ ดังนั้นความแปรปรวนร่วมของตำแหน่งรถหลังจากการทำนายการเคลื่อนที่ย่อมมีการเปลี่ยนแปลง โดยการกระจายความน่าจะเป็นของตำแหน่งของรถจะมีการกระจายมากขึ้นดังรูปที่ 2.1ข หลังจากการทำนายการเคลื่อนที่แล้วก็จะได้ค่าประมาณตำแหน่งของหุ่นยนต์ ณ เวลาปัจจุบัน จากนั้นจึงนำค่าที่ได้จากการวัดจาก GPS มาปรับแก้ตำแหน่ง และ ความแปรปรวนร่วมของตำแหน่งรถ โดยใช้โมเดลการวัดของ GPS (สมมุติว่ารู้โมเดลการวัดของ GPS อยู่แล้ว) ซึ่งจะส่งผลให้ตำแหน่งของรถมีความแม่นยำมาก

ขึ้น ซึ่งก็คือ การกระจายความน่าจะเป็นของตำแหน่งของรถมีการกระจายน้อยลงดังรูปที่ 2.1ค และเมื่อวนซ้ำการทำงานสองขั้นตอน Prediction และ Update measurement ไปเรื่อย ๆ ก็จะสามารถประมาณตำแหน่งของรถที่แม่นยำได้



(ก) (ข) (ค)

รูปที่ 2.1: การทำงานของ Kalman Filter

ขั้นตอนการทำงานของ Kalman Filter แสดงได้โดยละเอียดได้ดังนี้

2.2.1 Prediction

สำหรับขั้นตอน Prediction นั้น เมื่อระบบมีการเปลี่ยนแปลงทางเวลา (เมื่อเวลาดำเนินไป) สถานะของระบบย่อมต้องมีความเปลี่ยนแปลงเกิดขึ้น ในส่วน Prediction นั้นจะเป็นการประมาณสถานะของระบบใหม่เมื่อเวลาเปลี่ยนแปลงไป โดยอาศัย Process Model ซึ่ง Process Model นั้นจะอธิบายการเปลี่ยนแปลงของสถานะของระบบเมื่อเวลาเปลี่ยนแปลงไป ดังรูป

$$x_k = f(x_{k-1}, u_k) + w_k \tag{2.25}$$

ในที่นี้ $f(x_{k-1}, u_k)$ เป็น state transition model, x_{k-1} เป็นสถานะของระบบเมื่อเวลาก่อนหน้า, x_k เป็นสถานะของระบบที่เวลาปัจจุบัน, u_k เป็น คำสั่งควบคุมของระบบ (ค่าคำสั่งที่ผู้ใช้กำหนดให้แก่ระบบ) ส่วน w_k เป็น process noise มีการกระจายแบบ normal มีค่าเฉลี่ย (mean) เป็น 0 และ ค่าความแปรปรวนร่วม(Covariance) เป็น Q_k ($w_k = N(0, Q_k)$)

สาเหตุที่ต้องมี process noise (w_k) ร่วมอยู่ในระบบนั้นเป็นเพราะว่าไม่มีแบบจำลองทางคณิตศาสตร์ใด ๆ ที่สามารถจำลองการเปลี่ยนแปลงของระบบได้แม่นยำ 100 เปอร์เซนต์ ดังนั้น w_k ก็คือความคลาดเคลื่อนของการประมาณ

สำหรับ Kalman Filter แบบทั่วไปนั้นจะใช้ได้เฉพาะระบบที่เป็นแบบ Linear dynamical systems เท่านั้น (สำหรับระบบที่ไม่เป็น Linear จะต้องใช้ Extended kalman filter ซึ่งจะกล่าวต่อไป) ดังนั้น state transition model จะเขียนได้เป็น

$$f(x_{k-1}, u_k) = F_k x_{k-1} + B_k u_k \tag{2.26}$$

เมื่อ F_k เป็น State transition Matrix และ B_k เป็น Control Input Matrix

ซึ่งอันที่จริงแล้ว x_k เป็นค่าจริงของสถานะของระบบซึ่งไม่สามารถทราบค่าได้ สิ่งที่พอทำได้คือการประมาณค่า $\hat{x}_k$ ให้ได้ค่าใกล้เคียงค่าจริงมากที่สุดซึ่ง $\hat{x}_k$ หาได้จาก

$$\hat{x}_k^- = E[x_k] = E[F_k x_{k-1} + B_k u_k + w_k] \quad (2.27)$$
$$\hat{x}_k^- = F_k E[x_{k-1}] + B_k E[u_k] + E[w_k] \quad (2.28)$$
$$\hat{x}_k^- = F_k \hat{x}_{k-1} + B_k u_k \quad (2.29)$$

ในที่นี้ $E[x_k]$ คือ ค่าความคาดหวัง(Expected Value) ของ x_k เขียนแทนด้วย $\hat{x}_k$ ส่วน $\hat{x}_k^-$ คือสถานะประมาณของระบบที่เกิดจากการทำนายล่วงหน้า และ $\hat{x}_{k-1}$ คือสถานะประมาณของระบบที่เวลาก่อนหน้า

ซึ่งจากสมการข้างต้น $E[w_k]$ เท่ากับ 0 เพราะได้กำหนดไว้แล้วว่า ค่าเฉลี่ยของ w_k เป็น 0 และ $E[u_k]$ เท่ากับ u_k เพราะคำสั่งควบคุม เป็นคำสั่งที่แน่นอนค่าเดียว

จากนั้นจะประมาณการกระจายความน่าจะเป็นของสถานะของระบบหลังจากการทำนายล่วงหน้า ได้ด้วยค่าสองค่า คือ ค่าเฉลี่ยของความคลาดเคลื่อน (Mean) ($E[\hat{e}_k^-]$) และ ค่าความแปรปรวนของความคลาดเคลื่อน (Covariance) ($cov[\hat{e}_k^-]$)

$$E[\hat{e}_k^-] = E[x_k - \hat{x}_k^-] = E[F_k x_{k-1} + B_k u_k + w_k - F_k \hat{x}_{k-1} - B_k u_k] \quad (2.30)$$
$$E[\hat{e}_k^-] = E[F_k(x_{k-1} - \hat{x}_{k-1}) + w_k] \quad (2.31)$$
$$E[\hat{e}_k^-] = E[F_k \hat{e}_{k-1} + w_k] = F_k E[\hat{e}_{k-1}] + E[w_k] = 0 \quad (2.32)$$

เพราะ $E[w_k]$ มีค่าเป็น 0 และ ค่าเฉลี่ยของความคลาดเคลื่อนเมื่อเวลาก่อนหน้า ($E[\hat{e}_{k-1}^-]$) ก็เป็น 0 เช่นกัน ดังนั้น กระจายความน่าจะเป็นจะอธิบายได้ด้วยค่าความแปรปรวน (Covariance) เพียงอย่างเดียว

$$P_k^- = cov[\hat{e}_k^-] = cov[F_k \hat{e}_{k-1} + w_k] \quad (2.33)$$
$$P_k^- = E[(F_k \hat{e}_{k-1} + w_k)(F_k \hat{e}_{k-1} + w_k)^T] \quad (2.34)$$
$$P_k^- = F_k E[\hat{e}_{k-1}\hat{e}_{k-1}^T] F_k^T + F_k E[\hat{e}_{k-1} w_k^T] + E[w_k \hat{e}_{k-1}^T] F_k^T + E[w_k w_k^T] \quad (2.35)$$
$$P_k^- = F_k P_{k-1} F_k^T + Q_k \quad (2.36)$$

สรุปแล้วในกระบวนการ Prediction คือการประมาณสถานะของระบบหลังจากมีการเปลี่ยนแปลงทางเวลา ซึ่งสถานะของระบบจะประมาณได้จากสมการ 2.29 ส่วน ค่าความแปรปรวนร่วมของสถานะของระบบประมาณได้จากสมการ 2.36 โดยที่ค่าเฉลี่ยของความคลาดเคลื่อนเป็น 0

2.2.2 Update measurement

Update measurement เป็นกระบวนการที่นำข้อมูลที่ได้จากการวัดค่าของระบบ มาใช้ในการปรับแก้ค่าประมาณของสถานะของระบบ เพื่อให้สถานะประมาณของระบบมีความแม่นยำมากขึ้น ซึ่งกระบวนการปรับแก้ค่านั้น ค่าการวัดไม่จำเป็นต้องวัดได้เป็นค่าสถานะของระบบโดยตรงก็ได้ ขอเพียงให้ค่าการวัดมีความสัมพันธ์กับสถานะของระบบก็พอ เช่น ในการประมาณตำแหน่งของหุ่นยนต์ในสามมิติด้วยกล้อง ที่รู้การวางตัวของกล้องอยู่ก่อนหน้าแล้ว ข้อมูลที่ได้จากกล้องนั้นย่อมไม่สามารถระบุตำแหน่งของหุ่นยนต์เป็นค่าพิกัดได้โดยตรง ถึงแม้ว่าข้อมูลภาพจากกล้องจะไม่สามารถคำนวนตำแหน่งหุ่นยนต์ในสามมิติได้ เพราะว่าข้อมูลการวัดที่ได้จากกล้องนั้นมีพารามิเตอร์น้อยเกินไป แต่จากตำแหน่งของหุ่นยนต์ในสามมิติ

สามารถนำไปคำนวนเป็นพิกัดในภาพของกล้องได้ เช่นนี้แล้วก็ถือว่าค่าการวัดมีความสัมพันธ์กับสถานะของระบบ หรือ อย่างเช่นในกระบวนการทำงานของอุปกรณ์ GPS อันที่จริงแล้วอุปกรณ์ GPS นั้นไม่สามารถวัดตำแหน่งบนโลกได้โดยตรง แต่อุปกรณ์ GPS จะใช้หลักการวัดเวลาในการเดินทางของคลื่นแม่เหล็กไฟฟ้าระหว่างอุปกรณ์ GPS กับดาวเทียมหลาย ๆ ดวง ซึ่งเวลาที่ใช้ในการเดินทางของคลื่นวิทยุ จะสามารถนำมาคำนวนเป็นระยะทางระหว่างอุปกรณ์ GPS กับดาวเทียมได้ จากนั้นจึงใช้ข้อมูลระยะทางจากดาวเทียมอย่างน้อย 4 ดวง และตำแหน่งของดาวเทียมที่ทราบค่าอยู่แล้วเพื่อคำนวนหาตำแหน่งบนโลก เวลาที่วัดได้ในการเดินทางของคลื่นวิทยุระหว่างอุปกรณ์ GPS กับดาวเทียมแต่ละดวงนั้น ก็ถือเป็นค่าการวัดได้เช่นกัน

การจะนำค่าการวัดมาใช้ปรับแก้ได้นั้นก่อนอื่นจะต้องทราบโมเดลการวัด หรือ Measurement model เสียก่อน โดย Measurement model เขียนได้เป็น

$$z_k = h(x_k) + v_k \tag{2.37}$$

ในที่นี้ $h(x_k)$ เป็น observation model, x_k เป็นสถานะของระบบที่เวลาปัจจุบัน, z_k เป็น ค่าที่ได้จากการวัดจริง ส่วน v_k เป็น observation noise มีการกระจายแบบ normal มี ค่าเฉลี่ย (mean) เป็น 0 และค่าความแปรปรวนร่วม(Covariance) เป็น R_k ($v_k = N(0, R_k)$)

สาเหตุที่ต้องมี observation noise (v_k) ร่วมอยู่ในระบบนั้นเป็นเพราะว่า observation model ไม่สามารถจำลองรูปแบบการวัดค่าได้แม่นยำ 100 เปอร์เซนต์ ดังนั้น v_k ก็คือความคลาดเคลื่อนของการประมาณ

และเมื่อเขียน observation model ให้อยู่ในรูปแบบ Linear แล้วจะได้ว่า

$$z_k = H_k x_k + v_k \tag{2.38}$$

ค่าประมาณการวัด ($\hat{z}_k$) หาได้จาก

$$\hat{z}_k = E[z_k] = E[H_k x_k + v_k] \tag{2.39}$$
$$\hat{z}_k = H_k E[x_k] + E[v_k] \tag{2.40}$$
$$\hat{z}_k = H_k \hat{x}_k^- \tag{2.41}$$

ค่า $\hat{z}_k$ นั้นเป็นค่าประมาณค่าการวัดโดยประมาณจากสถานะของระบบที่เกิดจากการทำนายล่วงหน้า ส่วน z_k เป็นค่าจริงที่วัดได้ ซึ่ง kalman filter จะปรับแก้ค่าสถานะของระบบโดยมีสมมุติฐานว่า สถานะของระบบหลังจากการปรับแก้ จะเท่ากับ สถานะของระบบจากการทำนายล่วงหน้า รวมกับค่า kalman gain คูณด้วยความต่างระหว่าง ค่าการวัดจริง และค่าประมาณค่าการวัด ดังสมการ

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{z}_k) \tag{2.42}$$
$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k \hat{x}_k^-) \tag{2.43}$$

เมื่อ $\hat{x}_k$ เป็นค่าประมาณสถานะของระบบที่ได้จากการปรับแก้ค่าแล้ว และ K_k เป็น kalman gain ซึ่งจะเป็น matrix ที่มีค่าเหมาะสมที่ทำจะให้ การกระจายความน่าจะเป็นของสถานะของระบบหลังการปรับแก้

มีค่าต่ำสุด

การกระจายความน่าจะเป็นของสถานะของระบบหลังการปรับแก้ จะประมาณได้ด้วยค่าสองค่า คือ ค่าเฉลี่ยของความคลาดเคลื่อน (Mean) ($E[\hat{e}_k]$) และ ค่าความแปรปรวนของความคลาดเคลื่อน (Covariance) ($cov[\hat{e}_k]$)

$$E[\hat{e}_k] = E[x_k - \hat{x}_k] = E[x_k - \hat{x}_k^- - K_k(z_k - H_k\hat{x}_k^-)] \quad (2.44)$$
$$E[\hat{e}_k] = E[\hat{e}_k^- - K_k(H_k x_k + v_k - H_k\hat{x}_k^-)] \quad (2.45)$$
$$E[\hat{e}_k] = E[\hat{e}_k^- - K_k(H_k\hat{e}_k^- + v_k)] \quad (2.46)$$
$$E[\hat{e}_k] = E[(I - K_kH_k)\hat{e}_k^- + K_kv_k] \quad (2.47)$$
$$E[\hat{e}_k] = (I - K_kH_k)E[\hat{e}_k^-] + K_kE[v_k] = 0 \quad (2.48)$$

เพราะ $E[v_k]$ มีค่าเป็น 0 และ ค่าเฉลี่ยความคลาดเคลื่อนเมื่อก่อนปรับแก้ ($E[\hat{e}_k^-]$) ก็เป็น 0 เช่นกัน ดังนั้น กระจายความน่าจะเป็นจะอธิบายได้ด้วยค่าความแปรปรวน (Covariance) เพียงอย่างเดียว

$$P_k = cov[\hat{e}_k] = cov[(I - K_kH_k)\hat{e}_k^- + K_kv_k] \quad (2.49)$$

เนื่องจาก v_k ไม่ขึ้นกับ term อื่น ๆ ดังนั้นจะได้

$$P_k = cov[(I - K_kH_k)\hat{e}_k^-] + cov[K_kv_k] \quad (2.50)$$
$$P_k = (I - K_kH_k)cov[\hat{e}_k^-](I - K_kH_k)^T + K_kcov[v_k]K_k^T \quad (2.51)$$
$$P_k = (I - K_kH_k)P_k^-(I - K_kH_k)^T + K_kR_kK_k^T \quad (2.52)$$

จะได้ว่ากระบวนการ Update measurement ก็คือการประมาณสถานะของระบบหลังจากที่มีข้อมูลการวัด ซึ่งการประมาณด้วยข้อมูลการวัดนั้นมีจุดประสงค์เพื่อการปรับแก้ค่าประมาณของสถานะของระบบให้มีความแม่นยำมากขึ้น ซึ่งสถานะของระบบจะประมาณได้จากสมการ 2.41 ส่วน ค่าความแปรปรวนร่วมของสถานะของระบบประมาณได้จากสมการ 2.52 โดยที่ค่าเฉลี่ยของความคลาดเคลื่อนเป็น 0 ซึ่งค่าสถานะของระบบก่อนการปรับแก้นั้นควรจะเป็นค่าสถานะที่เวลาเดียวกับเวลาที่วัดค่าได้

จากสมการ 2.41 และ 2.52 จะเห็นได้ว่าค่า kalman gain (K_k) จะส่งผลกระทบต่อการประมาณการกระจายความน่าจะเป็นของสถานะของระบบ ซึ่งค่า kalman filter ที่ดีนั้นจะต้องทำให้กระจายความน่าจะเป็นมีการกระจายต่ำสุด ดังนั้นค่า kalman gain (K_k) จะสามารถหาได้จากการแก้สมการ gradient ของ P_k เท่ากับ 0 จะได้ว่า

$$P_k = (I - K_kH_k)P_k^-(I - K_kH_k)^T + K_kR_kK_k^T \quad (2.53)$$
$$P_k = (P_k^- - K_kH_kP_k^- - P_k^-H_k^TK_k^T + K_kH_kP_k^-H_k^TK_k^T) + K_kR_kK_k^T \quad (2.54)$$
$$P_k = P_k^- - K_kH_kP_k^- - P_k^-H_k^TK_k^T + K_k(H_kP_k^-H_k^T + R_k)K_k^T \quad (2.55)$$
$$P_k = P_k^- - K_kH_kP_k^- - P_k^-H_k^TK_k^T + K_kS_kK_k^T \quad when \quad S_k = H_kP_k^-H_k^T + R_k \quad (2.56)$$
$$\frac{\partial P_k}{\partial K_k} = -2(H_kP_k^-)^T + 2K_kS_k = 0 \quad (2.57)$$

ดังนั้นจะหาค่า kalman gain (K_k) ได้เป็น

$$K_k S_k = (H_k P_k^-)^T \quad (2.58)$$

$$K_k = P_k^- H_k^T S_k^{-1} \quad (2.59)$$

สรุปขั้นตอนของ kalman Filter ประกอบด้วยสองขั้นตอนคือ Prediction และ Update measurement

Prediction

Predicted State $\hat{x}_k^- = F_k \hat{x}_{k-1} + B_k u_k$

Predicted estimate covariance $P_k^- = F_k P_{k-1} F_k^T + Q_k$

Update measurement

Innovation or measurement residual $\hat{y}_k = z_k - H_k \hat{x}_k^-$

Innovation (or residual) covariance $S_k = H_k P_k^- H_k^T + R_k$

Optimal Kalman gain $K_k = P_k^- H_k^T S_k^{-1}$

Updated state estimate $\hat{x}_k = \hat{x}_k^- + K_k \hat{y}_k$

Updated estimate covariance $P_k = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T$

เมื่อ

$\hat{x}_{k-1}$ คือ สถานะประมาณของระบบ ที่เวลา $k-1$

$\hat{x}_k^-$ คือ สถานะประมาณของระบบ ภายหลังการทำนายล่วงหน้า ที่เวลา k

$\hat{x}_k$ คือ สถานะประมาณของระบบ ภายหลังการปรับแก้ค่าจากการวัด ที่เวลา k

P_{k-1} คือความแปรปรวนร่วมของสถานะของระบบ ที่เวลา $k-1$

P_k^- คือ ความแปรปรวนร่วมของสถานะของระบบ ภายหลังการทำนายล่วงหน้า ที่เวลา k

P_k คือ ความแปรปรวนร่วมของสถานะของระบบ ภายหลังการปรับแก้ค่าจากการวัด ที่เวลา k

z_k คือ ค่าการวัดสถานะของระบบ

u_k คือ คำสั่งควบคุมการเปลี่ยนแปลง สถานะของระบบ

F_k คือ state transition model

B_k คือ control model

H_k คือ observation model

Q_k คือ ความแปรปรวนร่วมของ Process noise

R_k คือ ความแปรปรวนร่วมของ observation noise

ตัวอย่างการใช้งาน kalman filter

เช่น การประมาณตำแหน่งของรถซึ่งติดตั้งอุปกรณ์ GPS คันหนึ่ง ที่วิ่งอยู่บนถนนด้วยความเร็วคงที่ ในกรณีนี้ สถานะของระบบนี้จะประกอบด้วย ตำแหน่งของรถที่เวลา k (p_k) และความเร็วของรถ (c_k)

$$x_k = \begin{bmatrix} p_k & c_k \end{bmatrix}^T \quad (2.60)$$

ในที่นี้ state transition model ซึ่งเป็นโมเดลการเปลี่ยนแปลงของระบบ จะเป็นสมการการเคลื่อนที่ของรถนั่นเอง

$$f(x_{k-1}, u_k) = \begin{bmatrix} p_k \\ c_k \end{bmatrix} = \begin{bmatrix} p_{k-1} + c_{k-1}\Delta t \\ c_{k-1} \end{bmatrix} \tag{2.61}$$

ทั้งนี้ได้กำหนดให้รถมีความเร็วคงที่ ซึ่งหมายความว่าระบบไม่มีคำสั่งควบคุมจากผู้ใช้ (ไม่มี u_k) ส่วน observation model ซึ่งเป็นโมเดลการวัด สามารถเขียนได้เป็น

$$h(x_k) = [p_k] \tag{2.62}$$

จาก observation model ข้างต้น หมายความว่า ค่าประมาณการวัดจาก GPS หาได้จากตำแหน่งของรถโดยตรง (ซึ่งในที่นี้สมมุติว่า ค่าที่วัดได้จากอุปกรณ์ GPS เป็นค่าในพิกัดเดียวกันกับตำแหน่งของรถที่ต้องการประมาณ) และเนื่องจากระบบนี้เป็นระบบแบบ linear ดังนั้นจึงสามารถเขียน process model และ measurement model แบบ linear ได้เป็น

$$x_k = F_k x_{k-1} + w_k \tag{2.63}$$
$$z_k = H_k x_k + v_k \tag{2.64}$$

โดยที่ $F_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$, $H_k = \begin{bmatrix} 1 & 0 \end{bmatrix}$

สำหรับในตอนเริ่มแรกของการประมาณค่าด้วย kalman filter นั้นจะต้องกำหนด สถานะเริ่มต้น ($\hat{x}_0$) ให้กับระบบเสียก่อน ซึ่งในกรณีนี้จะกำหนดให้ ตำแหน่งเริ่มต้นเป็นตำแหน่งที่วัดได้โดยอุปกรณ์ GPS ณ ขณะนั้นส่วนความเร็วของรถ ก็ไม่ทราบค่า ดังนั้นจึงกำหนดให้ความเร็วเริ่มต้นเป็น 0 เช่นกัน

$$\hat{x}_0 = \begin{bmatrix} z_0 & 0 \end{bmatrix}^T \tag{2.65}$$

จะเห็นได้ว่า สถานะเริ่มต้นนั้นเป็นค่าที่ไม่ถูกต้องนัก เนื่องจากว่า อุปกรณ์ GPS นั้นเป็นอุปกรณ์ที่มีความคลาดเคลื่อนสูง และความเร็วของรถนั้นย่อมไม่น่าจะเท่ากับ 0 แน่นอน แต่จะเห็นได้ว่าการกำหนดสถานะเริ่มต้นนั้นไม่มีทางเลือกมากนัก เนื่องจากว่าในตอนเริ่มต้นอุปกรณ์ GPS นั้นเพิ่งจะเริ่มต้นวัดค่าเป็นครั้งแรกได้เพียงค่าเดียว และความเร็วของรถก็อาจจะเป็นค่าติดลบก็ได้ (ในกรณีรถวิ่งถอยหลัง) สิ่งที่พอจะทำได้ก็คือ กำหนดให้ค่าความแปรปรวนเริ่มต้นมีค่ามากพอกับความคลาดเคลื่อนของสถานะเริ่มต้น จะได้ว่า

$$P_0 = \begin{bmatrix} \sigma_p^2 & 0 \\ 0 & \sigma_c^2 \end{bmatrix} \tag{2.66}$$

ซึ่ง σ_p เป็นค่าเบี่ยงเบนมาตราฐาน (standard deviation) ของตำแหน่งของรถ และ σ_c เป็นค่าเบี่ยงเบนมาตราฐานของความเร็วของรถ ในที่นี้อาจจะกำหนดให้ σ_p มีค่าประมาณ 10 เมตร เท่ากับความคลาดเคลื่อนของ GPS และ อาจจะกำหนดให้ σ_c มีค่าประมาณ 25 เมตร/วินาที เท่ากับความเร็วของรถทั่วไป

สำหรับสถานะเริ่มต้นของรถนั้นหากไม่ต้องการให้มีการกระจายความน่าจะเป็นมากจนเกินไปในตอนเริ่มต้น อาจจะประมาณสถานะเริ่มต้นด้วยวิธีการอื่นก่อน เช่นการใช้ Least Squares ช่วยประมาณความเร็วและจุดเริ่มต้นไปเป็นเวลาสักระยะหนึ่งจนกว่าสถานะเริ่มต้นมีค่าที่ยอมรับได้จึงนำมาใช้เป็น

สถานะเริ่มต้นของ kalman filter ต่อไป

นอกจากสถานะเริ่มต้นแล้ว อีกสิ่งหนึ่งที่จะต้องกำหนดให้กับระบบคือ process noise และ observation noise ซึ่ง noise ทั้งสองค่านั้นเป็นแบบ normal และมีค่าเฉลี่ย (mean) เป็น 0 ดังนั้น process noise จึงแสดงได้ด้วยความแปรปรวนร่วมของการเปลี่ยนแปลงของระบบ (Q_k) และ observation noise แสดงได้ด้วยความแปรปรวนร่วมของการวัดค่า (R_k)

โดยทั่วไปแล้ว ความแปรปรวนร่วมของการวัดค่า (R_k) นั้นมักจะกำหนดให้มีค่าเท่ากับความคลาดเคลื่อนของการวัดค่าจากอุปกรณ์ ส่วน ความแปรปรวนร่วมของการเปลี่ยนแปลงของระบบ (Q_k) มักจะคำนวนจากความคลาดเคลื่อนที่อาจจะเกิดขึ้นได้จากการประมาณการเปลี่ยนแปลงของระบบ เช่นในตัวอย่างนี้

$$R_k = [\sigma_z^2] \tag{2.67}$$

$$Q_k = \begin{bmatrix} \sigma_p^2 & 0 \\ 0 & \sigma_c^2 \end{bmatrix} \tag{2.68}$$

โดยที่ σ_z เป็นความคลาดเคลื่อนของอุปกรณ์ GPS มีค่าประมาณ 10 เมตร ส่วน σ_p และ σ_c กำหนดให้มีค่าน้อยมาก

ดังนั้นเมื่อรถมีการเคลื่อนที่ไปและ GPS สามารถวัดค่าก็จะสามารถประมาณตำแหน่งของรถที่เวลาถัดไปได้ด้วย สถานะเริ่มต้น($\hat{x}_0, P_0$), state transition model (F_k), observation model (H_k), process covariance (Q_k) และ observation covariance (R_k) ที่ได้หาไว้แล้วก่อนหน้านี้ ซึ่งเมื่อเวลาผ่านไปสักระยะหนึ่งแล้ว ตำแหน่งของรถที่ประมาณได้จะก็มีความแม่นยำขึ้นเอง (มี error covariance น้อยลง) อันเป็นผลมาจากการใช้ ค่าจาก GPS ปรับแก้ และจะสามารถประมาณความเร็วของรถได้จากสถานะประมาณของระบบ

ในการประมาณความเร็วของรถในตัวอย่างนี้ process model ที่กำหนดให้กับ kalman นั้นเป็น model ที่ค่อนข้างมีความแม่นยำเพราะเป็น model การเคลื่อนที่ของรถ จึงสามารถกำหนดให้ ความแปรปรวนร่วมของการเปลี่ยนแปลงของระบบ (Q_k) มีค่าน้อยได้ แต่ถ้าหากว่า process model นั้นสามารถประมาณ ได้เพียงคราว ๆ ค่า Q_k ก็จะมีค่ามากขึ้นซึ่ง Q_k ก็มักจะคำนวนได้จากการเปลี่ยนแปลงที่ไม่สามารถคาดเดาได้ ยกตัวอย่างเช่น

จากตัวอย่างนี้ ถ้ารถที่ต้องการประมาณตำแหน่ง ไม่ได้วิ่งด้วยความเร็วคงที่ อีกทั้งไม่สามารถวัดว่าคำสั่งที่ใช้ควบคุมรถนั้นเป็นเท่าไร่ process model ที่ประมาณได้นั้นก็จะมีความไม่แม่นยำเกิดขึ้นได้ และสมมุติว่าหากต้องการยืนกรานที่จะใช้ state transition model เดิมเพื่อใช้ในการประมาณ ซึ่งก็คือ

$$f(x_{k-1}, u_k) = \begin{bmatrix} p_k \\ c_k \end{bmatrix} = \begin{bmatrix} p_{k-1} + c_{k-1}\Delta t \\ c_{k-1} \end{bmatrix} \tag{2.69}$$

process model ที่ใช้ในการประมาณนั้น อาจมีความคลาดเคลื่อนได้ เนื่องจากการเปลี่ยนแปลงของความเร็ว ดังนั้นวิธีการหาความแปรปรวนร่วมของการเปลี่ยนแปลงของระบบ (Q_k) ก็คือ จะกำหนดให้ process model อาจมีการเปลี่ยนแปลงเกิดขึ้นได้ เนื่องจากความเร่งไม่ทราบค่า process model ใหม่ที่ใช้ในการประมาณจะได้เป็น

$$x_k = \begin{bmatrix} p_k \\ c_k \end{bmatrix} = \begin{bmatrix} p_{k-1} + c_{k-1}\Delta t + 0.5a_k\Delta t^2 \\ c_{k-1} + a_k\Delta t \end{bmatrix} \tag{2.70}$$

โดย a_k เป็นความเร่งไม่ทราบค่า และเมื่อเขียน process model ให้อยู่ในรูปแบบ linear ได้เป็น

$$x_k = F_k x_{k-1} + w_k \quad (2.71)$$
$$w_k = G_k a_k \quad (2.72)$$

โดยที่ $F_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$, $H_k = \begin{bmatrix} 0.5\Delta t^2 \\ \Delta t \end{bmatrix}$

ค่าความเร่ง a_k นั้นจะกำหนดให้มีค่าเฉลี่ยเป็น 0 และ มีค่าความแปรปรวนเป็น A_k ซึ่งเมื่อ $\hat{a}_k$ เป็น 0 ย่อมจะไม่ส่งผลต่อค่าประมาณสถานะของระบบ และ ค่าความแปรปรวนร่วมของการเปลี่ยนแปลงของระบบ (Q_k) จะหาได้จาก

$$Q_k = E[w_k w_k^T] = E[(G_k a_k)(G_k a_k)^T] \quad (2.73)$$
$$Q_k = G_k E[a_k a_k^T] G_k^T \quad (2.74)$$
$$Q_k = G_k A_k G_k^T \quad (2.75)$$

2.3 Extended kalman filter

Extended kalman filter (Julier and Uhlmann, 1997) คือการนำเอา kalman filter มาใช้ในการประมาณสถานะของระบบที่เป็นแบบ Non-linear ซึ่งในความเป็นจริงแล้ว kalman filter ไม่สามารถประมาณระบบสมการที่เป็น non-linear ได้โดยตรง แต่ Extended kalman filter จะทำการ linearize ฟังก์ชันที่เป็น non-linear ให้เป็นแบบ linear เสียก่อนโดยใช้ Jacobian

2.3.1 Prediction

Process model สามารถเขียนให้อยู่ในรูปสมการ non-linear ได้เป็น

$$x_k = f(x_{k-1}, u_k) + w_k \quad (2.76)$$

ซึ่ง state transition model สามารถประมาณให้เป็นฟังก์ชันแบบ Linear โดยใช้ Jacobian ได้เป็น

$$f(x_{k-1}, u_k) = f(\hat{x}_{k-1}, u_k) + F_k(x_{k-1} - \hat{x}_{k-1}) + \dots \quad (2.77)$$

$$F_k = \frac{\partial f}{\partial x}|_{\hat{x}_{k-1}, u_k} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_m} \end{bmatrix} \quad (2.78)$$

โดยที่ F_k เป็น Jacobian Matrix ของ f เทียบกับ $\hat{x}_{k-1}$ และ u_k

ค่าประมาณสถานะของระบบหลังการทำนายการเปลี่ยนแปลงของระบบ หาได้เป็น

$$\hat{x}_k^- = E[x_k] = E[f(x_{k-1}, u_k) + w_k] \quad (2.79)$$
$$\hat{x}_k^- = E[f(\hat{x}_{k-1}, u_k) + F_k(x_{k-1} - \hat{x}_{k-1}) + \cdots + w_k] \quad (2.80)$$
$$\hat{x}_k^- = E[f(\hat{x}_{k-1}, u_k)] + E[F_k(x_{k-1} - \hat{x}_{k-1})] + \cdots + E[w_k] \quad (2.81)$$
$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k) + F_k(\hat{x}_{k-1} - \hat{x}_{k-1}) + \cdots \quad (2.82)$$
$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k) \quad (2.83)$$

เมื่อความคลาดเคลื่อนการประมาณเป็น

$$\hat{e}_k^- = x_k - \hat{x}_k^- \quad (2.84)$$
$$\hat{e}_k^- = f(x_{k-1}, u_k) + w_k - f(\hat{x}_{k-1}, u_k) \quad (2.85)$$
$$\hat{e}_k^- \approx f(\hat{x}_{k-1}, u_k) + F_k(x_{k-1} - \hat{x}_{k-1}) + w_k - f(\hat{x}_{k-1}, u_k) \quad (2.86)$$
$$\hat{e}_k^- \approx F_k(x_{k-1} - \hat{x}_{k-1}) + w_k = F_k\hat{e}_{k-1} + w_k \quad (2.87)$$

ค่าเฉลี่ยของความคลาดเคลื่อน (Mean) ($E[\hat{e}_k^-]$) หาได้เป็น

$$E[\hat{e}_k^-] = E[F_k\hat{e}_{k-1}^- + w_(k)] = F_kE[\hat{e}_{k-1}^-] + E[w_k] = 0 \quad (2.88)$$

และค่าความแปรปรวนของความคลาดเคลื่อน (Covariance) ($cov[\hat{e}_k^-]$) หาได้เป็น

$$P_k^- = cov[\hat{e}_k^-] = cov[F_k\hat{e}_{k-1}^- + w_(k)] \quad (2.89)$$
$$P_k^- = E[(F_k\hat{e}_{k-1}^- + w_(k))(F_k\hat{e}_{k-1}^- + w_k)^T] \quad (2.90)$$
$$P_k^- = F_kE[e_{k-1}e_{k-1}^T]F_k^T + F_kE[e_{k-1}w_k^T] + E[w_ke_{k-1}^T]F_k^T + E[w_kw_k^T] \quad (2.91)$$
$$P_k^- = F_kP_{k-1}F_k^T + Q_k \quad (2.92)$$

สรุปแล้วในกระบวนการ Prediction คือการประมาณสถานะของระบบหลังจากมีการเปลี่ยนแปลงทางเวลา ซึ่งสถานะของระบบจะประมาณได้จากสมการ 2.83 ส่วน ค่าความแปรปรวนร่วมของสถานะของระบบประมาณได้จากสมการ 2.92 โดยที่ค่าเฉลี่ยของความคลาดเคลื่อนเป็น 0

2.3.2 Update measurement

Measurement model สามารถเขียนให้อยู่ในรูปสมการ non-linear ได้เป็น

$$z_k = h(x_k) + v_k \quad (2.93)$$

ซึ่ง observation model สามารถประมาณให้เป็นฟังก์ชันแบบ Linear โดยใช้ Jacobian ได้เป็น

$$h(x_k) = h(\hat{x}_k^-) + H_k(x_k - \hat{x}_k^-) + \ldots \quad (2.94)$$

$$H_k = \frac{\partial h}{\partial x}|_{\hat{x}_k^-} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_n}{\partial x_1} & \cdots & \frac{\partial h_n}{\partial x_m} \end{bmatrix} \tag{2.95}$$

โดยที่ H_k เป็น Jacobian Matrix ของ h เทียบกับ $\hat{x}_k^-$

ค่าประมาณค่าการวัดจากสถานะของระบบ หาได้เป็น

$$\begin{aligned} \hat{z}_k &= E[z_k] = E[h(x_k) + v_k] && (2.96) \\ \hat{z}_k &= E[h(\hat{x}_k^-) + H_k(x_k - \hat{x}_k^-) + \cdots + v_k] && (2.97) \\ \hat{z}_k &= E[h(\hat{x}_k^-)] + E[H_k(x_k - \hat{x}_k^-)] + \cdots + E[v_k] && (2.98) \\ \hat{z}_k &= h(\hat{x}_k^-) + H_k(\hat{x}_k^- - \hat{x}_k^-) + \cdots && (2.99) \\ \hat{z}_k &= h(\hat{x}_k^-) && (2.100) \end{aligned}$$

ค่าสถานะของระบบหลังการปรับแก้จะหาได้เป็น

$$\begin{aligned} \hat{x}_k &= \hat{x}_k^- + K_k(z_k - \hat{z}_k) && (2.101) \\ \hat{x}_k &= \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-)) && (2.102) \end{aligned}$$

เมื่อความคลาดเคลื่อนการประมาณเป็น

$$\begin{aligned} \hat{e}_k &= x_k - \hat{x}_k && (2.103) \\ \hat{e}_k &= x_k - \hat{x}_k^- - K_k(z_k - h(\hat{x}_k^-)) && (2.104) \\ \hat{e}_k &= \hat{e}_k^- - K_k(h(\hat{x}_k^-) + H_k(x_k - \hat{x}_k^-) + \cdots + v_k - h(\hat{x}_k^-)) && (2.105) \\ \hat{e}_k &\approx \hat{e}_k^- - K_k(H_k\hat{e}_k^- + v_k) && (2.106) \\ \hat{e}_k &\approx (I - K_kH_k)\hat{e}_k^- + K_kv_k && (2.107) \end{aligned}$$

ค่าเฉลี่ยของความคลาดเคลื่อน (Mean) ($E[\hat{e}_k]$) หาได้เป็น

$$\begin{aligned} E[\hat{e}_k] &= E[(I - K_kH_k)\hat{e}_k^- + K_kv_k] && (2.108) \\ E[\hat{e}_k] &= (I - K_kH_k)E[\hat{e}_k^-] + K_kE[v_k] = 0 && (2.109) \end{aligned}$$

ค่าความแปรปรวนของความคลาดเคลื่อน (Covariance) ($cov[\hat{e}_k]$) หาได้เป็น

$$P_k = cov[\hat{e}_k] = cov[(I - K_kH_k)\hat{e}_k^- + K_kv_k] \tag{2.110}$$

เนื่องจาก v_k ไม่ขึ้นกับ term อื่น ๆ ดังนั้นจะได้

$$P_k = cov[(I - K_k H_k)\hat{e}_k^-] + cov[K_k v_k] \quad (2.111)$$
$$P_k = (I - K_k H_k) cov[\hat{e}_k^-](I - K_k H_k)^T + K_k cov[v_k] K_k^T \quad (2.112)$$
$$P_k = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \quad (2.113)$$

จะเห็นได้ว่าทั้งสมการ 2.92 และ 2.113 ซึ่งเป็นสมการหาค่าประมาณ ค่าความแปรปรวนของสถานะของระบบ ทั้งก่อนปรับแก้ค่าการวัด และ หลังปรับแก้ค่าการวัด ซึ่งมีความคล้ายคลึงกับรูปแบบการประมาณระบบที่เป็น Linear ด้วย Kalman filter มากดังนั้น ค่า kalman gain (K_k) ย่อมต้องคล้ายคลึงกันด้วย

$$K_k = P_k^- H_k^T S_k^{-1} \quad (2.114)$$
$$S_k = H_k P_k^- H_k^T + R_k \quad (2.115)$$

สรุปขั้นตอนของ Extended kalman Filter ประกอบด้วยสองขั้นตอนคือ Prediction และ Update measurement

Prediction

Predicted State	$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k)$
Predicted estimate covariance	$P_k^- = F_k P_{k-1} F_k^T + Q_k$

Update measurement

Innovation or measurement residual	$\hat{y}_k = z_k - h(\hat{x}_k^-)$
Innovation (or residual) covariance	$S_k = H_k P_k^- H_k^T + R_k$
Optimal Kalman gain	$K_k = P_k^- H_k^T S_k^{-1}$
Updated state estimate	$\hat{x}_k = \hat{x}_k^- + K_k \hat{y}_k$
Updated estimate covariance	$P_k = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T$

โดยที่ F_k เป็น Jacobian Matrix ของ f เทียบกับ$\hat{x}_{k-1}$ และ u_k
โดยที่ H_k เป็น Jacobian Matrix ของ h เทียบกับ $\hat{x}_k^-$

บทที่ 3

การระบุตำแหน่งพร้อมกับการสร้างแผนที่

การระบุตำแหน่งและการสร้างแผนที่คือรูปแบบของผลลัพธ์ ของกระบวนการที่หุ่นยนต์รับข้อมูลจากเซนเซอร์และวิเคราะห์ข้อมูลให้อยู่ในรูปแบบที่หุ่นยนต์เข้าใจได้ ซึ่งการระบุตำแหน่งและการสร้างแผนที่นั้นถือเป็นวิธีการประมาณค่ารูปแบบหนึ่ง โดยหุ่นยนต์นั้นจะวัดข้อมูลสิ่งแวดล้อมได้จากเซนเซอร์ ชนิดต่าง ๆ ที่ติดตั้งอยู่บนตัวหุ่นยนต์ ไม่ว่าจะเป็น กล้องวิดีโอ, อุปกรณ์วัดระยะด้วยเลเซอร์, โซนาร์ (sonar) , accelerometers, gyro meters จากนั้นจึงใช้ข้อมูลจากเซนเซอร์เพื่อการประมาณลักษณะของแผนที่ และตำแหน่งของหุ่นยนต์ ซึ่งข้อมูลจากเซนเซอร์นั้นก็ย่อมมีความไม่แน่นอนในการวัดค่าดังนั้นการประมาณแผนที่ และตำแหน่งของหุ่นยนต์ ที่ดีจะต้องมุ่งเน้นเพื่อลดความไม่แน่นอนของแผนที่และตำแหน่งให้ได้มากที่สุด

วิธีการระบุตำแหน่งพร้อมกับการสร้างแผนที่ (Simultaneous Localization and Mapping: S-LAM) (Durrant-Whyte and Bailey, 2006) เป็นกระบวนการที่หุ่นยนต์จะสร้างแผนที่ของสภาพแวดล้อมในขณะที่กำลังเคลื่อนที่ และระบุตำแหน่งของตัวเองในเวลาพร้อม ๆ กัน โดยที่หุ่นยนต์นั้นไม่มีข้อมูลของสิ่งแวดล้อมมาก่อน ปัญหาการระบุตำแหน่งของหุ่นยนต์ในสภาพแวดล้อมที่มีไม่คุ้นเคย มีความแตกต่างกับปัญหาการระบุตำแหน่งของหุ่นยนต์ ในสภาพแวดล้อมที่รู้แผนที่อยู่แล้วเป็นอย่างมาก ทั้งนี้เนื่องจากว่าการระบุตำแหน่งของหุ่นยนต์จำเป็นจะต้องใช้ข้อมูลแผนที่เพื่อเป็นกรอบอ้างอิงในการระบุพิกัดของหุ่นยนต์ ส่วนในการสร้างแผนที่นั้น จะสร้างขึ้นได้จากการนำข้อมูลจากเซนเซอร์หลาย ๆ ข้อมูล จากบริเวณหลาย ๆ บริเวณทั่วสิ่งแวดล้อมที่หุ่นยนต์สนใจ มาประติดประต่อกันเป็นรูปร่างของแผนที่ ซึ่งในกระบวนการประติดประต่อข้อมูลนั้น หุ่นยนต์จำเป็นจะต้องอาศัยตำแหน่งของหุ่นยนต์ ณ บริเวณที่วัดข้อมูลจากเซนเซอร์เพื่อการคำนวนแผนที่แบบสัมบูรณ์ ดังนั้นปัญหาการระบุตำแหน่งพร้อมกับการสร้างแผนที่ จึงเป็นปัญหาที่เกี่ยวพันกันเป็นอย่างมาก การทำงานของ SLAM นั้นจะเป็นการใช้ข้อมูลจากเซนเซอร์มาประมาณแผนที่และตำแหน่งของหุ่นยนต์โดยพร้อม ๆ กันซึ่งจะใช้ข้อมูลแผนที่เดิมและข้อมูลตำแหน่งเดิม มาประมาณแผนที่ใหม่และตำแหน่งใหม่ ดังนั้นจึงกล่าวได้ว่า ความไม่แน่นอนของตำแหน่งของหุ่นยนต์ จะขึ้นกับความไม่แน่นอนของแผนที่ และ ความไม่แน่นอนของแผนที่ ก็ขึ้นกับความไม่แน่นอนของตำแหน่งของหุ่นยนต์ด้วย ซึ่งวิธีการของ SLAM นี้จะใช้ วิธีการทางความน่าจะเป็น (probabilistic methods) มาจัดการความไม่แน่นอนเหล่านี้ให้มีค่าต่ำสุด

3.1 การใช้งานของ SLAM

SLAM ถูกนำมาใช้งานทางด้านหุ่นยนต์อย่างแพร่หลาย อาทิเช่น

- **หุ่นยนต์สำรวจ** : มีการใช้ SLAM เพื่อการสร้างแผนที่ของสิ่งแวดล้อมที่ต้องการสำรวจและเพื่อการวิเคราะห์การเคลื่อนที่ของหุ่นยนต์ให้สำรวจได้เป็นบริเวณกว้างและทั่วถึง (Williams et al., 2001; LI et al., 2000)
- **หุ่นยนต์กู้ภัย** : มีจุดประสงค์การใช้งานคล้ายคลึงกับหุ่นยนต์สำรวจ แต่หุ่นยนต์กู้ภัยไม่ได้ต้องการสำรวจสภาพแวดล้อม แต่ต้องการสำรวจผู้ประสบภัยและให้ความช่วยเหลือ (Pfingsthorn et al., 2008)
- **รถอัตโนมัติ** : มีการใช้ SLAM เพื่อการระบุตำแหน่งของรถว่าอยู่จุดไหนในแผนที่เพื่อการขับเคลื่อนรถไปสู่จุดหมายได้อย่างถูกต้อง และจะต้องสร้างแผนที่สิ่งแวดล้อมบริเวณรอบ ๆ ตัวรถเพื่อที่จะระบุ

ได้ว่าบริเวณไหนคือถนน บริเวณไหนไม่ใช่ถนน และเพื่อให้สามารถหลบหลีกรถยนต์ขับอื่นได้อย่างปลอดภัย (Leonard, 2008)

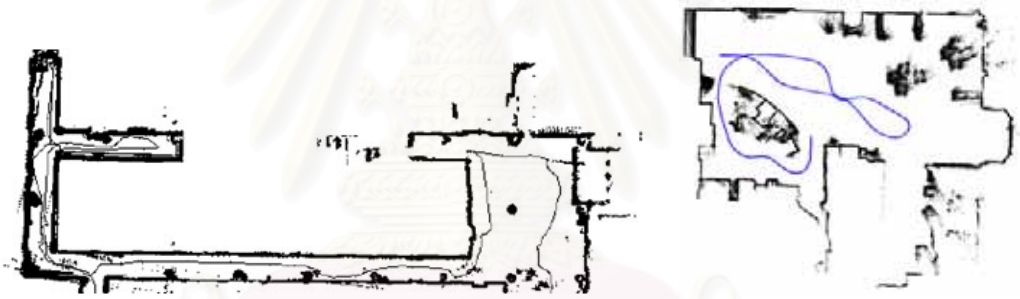
- **หุ่นยนต์การเกษตร** : มีการใช้งาน SLAM เพื่อสร้างแผนที่ของต้นไม้ว่ามีต้นไม้อยู่บริเวณไหนบ้างและเส้นทางเดินต่าง ๆ เพื่อให้หุ่นยนต์สามารถทำงานต่าง ๆ เกี่ยวกับต้นไม้ได้

3.2 ประเภทของ SLAM

รูปแบบผลลัพธ์ของ SLAM นั้นมีได้หลายรูปแบบขึ้นอยู่กับการใช้งานและอุปกรณ์เซนเซอร์ที่ใช้ ซึ่งจะถูกแบ่งออกเป็นประเภทใหญ่ ๆ ได้ 4 ประเภทคือ

Grid maps & Scans

SLAM ประเภท Grid maps และ Scans (Hahnel et al., 2003; Diosi and Kleeman, 2005) จะอธิบายแผนที่ในรูปเซตของจุดตำแหน่งปริมาณมากจนสามารถมองภาพรวมเห็นเป็นรูปร่างของสิ่งแวดล้อมได้ สำหรับ SLAM แบบ Grid maps นั้นตำแหน่งของจุดจะเรียงตัวในรูปแบบตาราง (สำหรับในกรณีแผนที่ 2 มิติ) ซึ่งแต่ละช่องของตารางก็จะเป็นความน่าจะเป็นที่จะปรากฏแผนที่ ส่วน SLAM แบบ Scans จะเป็นเซตของจุดจำนวนมากวางตัวกระจายกัน ซึ่งตำแหน่งจะไม่แน่นอนดังรูปที่ 3.1

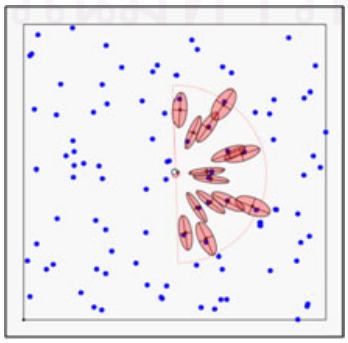


(ก) (ข)

รูปที่ 3.1: แผนที่แบบ Grid maps & Scans

Landmark-based

SLAM ประเภท Landmark-based (Asmar et al., 2006; Jeong and Lee, 2006) จะอธิบายแผนที่ในรูปแบบพารามิเตอร์ของ ลักษณะสำคัญ (features) ของสิ่งแวดล้อม ดังที่แสดงในรูป 3.2 เช่น ถ้า

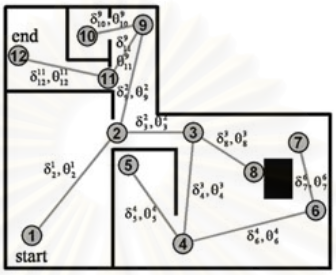


รูปที่ 3.2: แผนที่แบบ Landmark-based

สิ่งแวดล้อมมีลักษณะเป็น ป่า มีต้นไม้จำนวนมาก แผนที่ที่จะแสดงอาจจะในรูปแบบตำแหน่งของต้นไม้ต้นต่าง ๆ หรือหากสิ่งแวดล้อมที่สนในเป็นสิ่งแวดล้อมภายในอาคาร แผนที่ก็อาจจะเป็นเส้นตรงแทนด้านแต่ละด้านของผนัง

Topology

SLAM ประเภท Topology (Valgren et al., 2006; Motard et al., 2007) จะอธิบายแผนที่ในรูปแบบความสัมพันธ์ของสิ่งแวดล้อมบริเวณต่าง ๆ เป็นต้นว่าในบ้านหลังหนึ่งซึ่งมีห้องต่าง ๆ SLAM แบบ Topology นั้นจะอธิบายได้ว่าห้องต่าง ๆ นั้นมีความเชื่อมโยงกันอย่างไรบ้างดังที่แสดงในรูป 3.3



รูปที่ 3.3: แผนที่แบบ Topology

Appearance

SLAM ประเภท Appearance (Cummins and Newman, 2007, 2008) จะอธิบายแผนที่ในรูปแบบความสัมพันธ์ของสิ่งแวดล้อมบริเวณต่าง ๆ ซึ่งจะคล้ายเคียงกับ SLAM ประเภท Topology ดังที่แสดงในรูป 3.4 แต่ความต่างก็คือ SLAM แบบ Appearance จะใช้สำหรับกล้องวิดีโอเท่านั้น และความสัมพันธ์ของแผนที่จะเป็นความสัมพันธ์ของสิ่งแวดล้อมในเชิงความสัมพันธ์ของภาพมากกว่า เป็นต้นว่า ในการระบุตำแหน่งของหุ่นยนต์จะระบุได้ว่าตำแหน่งที่หุ่นยนต์อยู่ ณ ปัจจุบันมีความน่าจะเป็นตำแหน่งที่หุ่นยนต์เคยผ่านมาแล้วตอนไหนบ้าง และหุ่นยนต์จะสามารถคาดเดาได้ว่าตำแหน่งถัดไปที่หุ่นยนต์กำลังจะเคลื่อนที่ไปถึงจะมีลักษณะอย่างไร ในการหาความน่าจะเป็นของตำแหน่งหุ่นยนต์นั้นจะเป็นการหาความน่าจะเป็นของภาพว่าภาพปัจจุบันมีความคล้ายคลึงกับภาพในอดีตภาพไหนบ้าง ซึ่ง SLAM ประเภทนี้จะไม่สามารถบอกตำแหน่งที่แน่นอนเป็นพิกัดของหุ่นยนต์ได้

รูปที่ 3.4: แผนที่แบบ Appearance

3.3 การระบุตำแหน่งพร้อมกับการสร้างแผนที่ในเชิงความน่าจะเป็น

วิธีการระบุตำแหน่งพร้อมกับการสร้างแผนที่ (Simultaneous Localization and Mapping: SLAM) ถูกกล่าวถึงเป็นครั้งแรกเมื่อปี ค.ศ. 1986 โดย Smith และ Cheeseman (Smith and Cheeseman, 1986) ซึ่งได้นำเสนอวิธีในการประมาณการกระจายความน่าจะเป็นของตำแหน่งของหุ่นยนต์ และแผนที่ ด้วยวิธีการทางความน่าจะเป็น (probabilistic methods)

เพื่อความสะดวกในการอธิบาย จะขอกำหนดตัวแปรต่อไปนี้
เมื่อหุ่นยนต์เคลื่อนที่ผ่านสิ่งแวดล้อมพบจุดสังเกต (Landmarks) ณ เวลา k กำหนดให้

x_k เป็นสถานะของหุ่นยนต์ ประกอบด้วย ข้อมูลตำแหน่งและทิศทางของหุ่นยนต์ที่เวลา k

u_k เป็นคำสั่งของหุ่นยนต์ที่สั่งให้หุ่นเคลื่อนที่จากเวลา $k-1$ ไปสถานะ x_k ที่เวลา k

m_i เป็นเวกเตอร์บอกถึงตำแหน่งของจุดสังเกต (Landmarks) จุดที่ i โดยสมมุติว่าตำแหน่งที่แท้จริงไม่เปลี่ยนแปลงตามเวลา

$z_{i,k}$ เป็นตำแหน่งของจุดสังเกตที่ i ที่วัดได้จากหุ่นยนต์ที่เวลา k ซึ่งสามารถเขียนรวมทุกจุดสังเกตได้เป็น z_k

นอกจากนี้ยังขอกำหนดให้

$X_{0:k} = \{x_0, \cdots, x_k\} = \{X_{0:k-1}, x_k\}$: เป็นเซตของสถานะหุ่นยนต์ทั้งหมดตั้งแต่เวลา 0 ถึง k

$U_{0:k} = \{u_0, \cdots, u_k\} = \{U_{0:k-1}, u_k\}$: เป็นเซตของคำสั่งของหุ่นยนต์ทั้งหมดตั้งแต่เวลา 0 ถึง k

$m = m_0, \cdots, m_n$: เป็นเซตของจุดสังเกตทั้งหมด

$Z_{0:k} = \{z_0, \cdots, z_k\} = \{Z_{0:k-1}, z_k\}$: เป็นเซตของข้อมูลการวัดจุดสังเกตทั้งหมดตั้งแต่เวลา 0 ถึง k

รูปแบบของความน่าจะเป็นของปัญหาการระบุตำแหน่งพร้อมกับการสร้างแผนที่ (SLAM) สามารถเขียนในรูปแบบ การกระจายของความน่าจะเป็น (probability distribution) ได้ดังนี้

$$P(x_k, m | Z_{0:k}, U_{0:k}, x_0) \tag{3.1}$$

จากสมการแสดง การกระจายความน่าจะเป็นร่วมในภายหลัง (joint posterior density) ของสถานะหุ่นยนต์ (x_k) และตำแหน่งของจุดสังเกต (m) ที่เวลา k เมื่อกำหนด เซตของข้อมูลการวัดจุดสังเกตทั้งหมด ($Z_{0:k}$), คำสั่งทั้งหมดของหุ่นยนต์ ($U_{0:k}$) ตั้งแต่เวลา 0 ถึง k และ สถานะเริ่มต้นหุ่นยนต์ (x_0)

การคำนวนผลลัพธ์สำหรับปัญหาของ SLAM จะเป็นการคำนวนแบบเวียนบังเกิด (recursive solution) ซึ่งจะเริ่มต้นจากการคำนวนการกระจายความน่าจะเป็นร่วม $P(x_k, m | Z_{0:k}, U_{0:k}, x_0)$ ที่เวลา k เท่ากับ 0 จากนั้นเมื่อเวลาผ่านไป จะใช้คำสั่งควบคุมของหุ่นยนต์ (u_k) และ ข้อมูลการวัดจุดสังเกตที่วัดได้จากหุ่นยนต์ (z_k) มาใช้ในการประมาณกระจายความน่าจะเป็นร่วมที่เวลาใหม่โดยอาศัย Bayes theorem เวียนบังเกิดเช่นนี้ไปเรื่อย ๆ ซึ่งการคำนวนนั้นจำเป็นจะต้องอาศัยโมเดลการเปลี่ยนแปลงของสถานะหุ่นยนต์ (state transition model) และ โมเดลการวัดจุดสังเกต (observation model) ที่แปรผันตามคำสั่งของหุ่นยนต์ และจุดสังเกต ตามลำดับ มาใช้ในการประมาณการกระจายความน่าจะเป็นใหม่

โมเดลการวัดจุดสังเกต (observation model) จะบรรยายความน่าจะเป็น ของข้อมูลการวัดจุดสังเกตที่วัดได้จากหุ่นยนต์ (z_k) เมื่อรู้สถานะหุ่นยนต์ (x_k) และ ตำแหน่งของจุดสังเกต (m) ที่แน่นอน

$$P(z_k | x_k, m) \tag{3.2}$$

ในที่นี้จะสมสุติว่าข้อมูลการวัดจุดสังเกตที่วัดได้จากหุ่นยนต์ (z_k) จะขึ้นอยู่กับ สถานะหุ่นยนต์ (x_k) และ ตำแหน่งของจุดสังเกต (m) ปัจจุบันเท่านั้น ส่วนโมเดลการเปลี่ยนแปลงของสถานะหุ่นยนต์ (state transition model) จะบรรยายการกระจายความน่าจะเป็นของสถานะหุ่นยนต์ (x_k) ที่เวลา เมื่อรู้ สถานะหุ่นยนต์ที่เวลา และคำสั่งควบคุมของหุ่นยนต์ (u_k) ซึ่งแสดงได้ดังนี้

$$P(x_k | x_{k-1}, u_k) \tag{3.3}$$

สำหรับโมเดลการเปลี่ยนแปลงของสถานะหุ่นยนต์ (state transition model) จะใช้ในการทำนายการเปลี่ยนแปลงการกระจายความน่าจะเป็นของสถานะหุ่นยนต์ซึ่งเคลื่อนที่ไปเมื่อเวลาเปลี่ยน ส่วนโมเดลการวัดจุดสังเกต (observation model) จะใช้ในการประมาณสถานะใหม่ เมื่อได้รับข้อมูลการวัดจุดสังเกตุจากหุ่นยนต์ โดย state transition ทั้งหมดสามารถสมมุติให้เป็น Markov process ซึ่งสถานะถัดไป (x_k) จะขึ้นอยู่กับสถานะก่อนหน้า (x_{k-1}) และคำสั่งควบคุมของหุ่นยนต์ (u_k) เท่านั้น

วิธีการในการประมาณสถานะเมื่อมีการเปลี่ยนแปลงทางเวลาและเมื่อได้รับข้อมูลการวัดจุดสังเกตุจากหุ่นยนต์ จะสามารถอธิบายได้ด้วย อัลกอริทึมแบบเวียนบังเกิดสองขั้นตอนคือ การประมาณค่า (prediction, time-update) และการปรับแก้ค่า (correction, measurement-update) ซึ่งอยู่ในรูป

Time-update

คือการทำนายการเปลี่ยนแปลงการกระจายความน่าจะเป็นของสถานะหุ่นยนต์เมื่อเวลาเปลี่ยนแปลงไป การกระจายความน่าจะเป็นของสถานะใหม่จะประมาณได้จาก โมเดลการเปลี่ยนแปลงของสถานะหุ่นยนต์ (state transition model)

$$P(x_k, m|Z_{0:k-1}, U_{0:k}, x_0) = \int P(x_k|x_{k-1}, u_k) \cdot P(x_{k-1}, m|Z_{0:k-1}, U_{0:k-1}, x_0) \cdot dx_{k-1} \qquad (3.4)$$

Measurement Update

คือการประมาณสถานะใหม่ เมื่อได้รับข้อมูลการวัดจุดสังเกตุจากหุ่นยนต์ ซึ่งการกระจายความน่าจะเป็นของสถานะใหม่จะประมาณได้จาก โมเดลการวัดจุดสังเกต (observation model)

$$P(x_k, m|Z_{0:k}, U_{0:k}, x_0) = \frac{P(z_k|x_k, m) \cdot P(x_k, m|Z_{0:k-1}, U_{0:k}, x_0)}{P(z_k|Z_{0:k-1}, U_{0:k})} \qquad (3.5)$$

ในสมการที่ 3.4 และ 3.5 เป็นการคำนวนแบบเวียนบังเกิด สำหรับคำนวนหา ความน่าจะเป็นร่วมภายหลัง (joint posterior) $P(x_k, m|Z_{0:k}, U_{0:k}, x_0)$ สำหรับสถานะหุ่นยนต์ (x_k) และ เซตของจุดสังเกตทั้งหมดในแผนที่ m ที่เวลา k โดยขึ้นกับ เซตของจุดสังเกตทั้งหมดที่วัดได้ ($Z_{0:k}$) และ คำสั่งทั้งหมดของหุ่นยนต์ ($U_{0:k}$)

สำหรับปัญหาการสร้างแผนที่เพียงอย่างเดียวสามารถเขียน การกระจายความน่าจะเป็น ได้เป็น $P(m|X_{0:k}, Z_{0:k}, U_{0:k})$ โดยกำหนดว่ารู้สถานะหุ่นยนต์ (x_k) ที่แน่นอนตลอดเวลา ดังนั้นแผนที่จะเป็นการรวมกันระหว่างจุดสังเกตที่ตำแหน่งของหุ่นยนต์ต่าง ๆ กัน ในทางกลับกัน สำหรับปัญหาที่ต้องการการระบุตำแหน่งเพียงอย่างเดียวสามารถการกระจายความน่าจะเป็น ได้เป็น $P(x_k|Z_{0:k}, U_{0:k}, m)$ โดยที่สมมุติว่ารู้ตำแหน่งของจุดสังเกตุที่แน่นอนตลอดเวลา ดังนั้นจุดมุ่งหมายจะกลายเป็นการประมาณสถานะหุ่นยนต์

3.4 การแก้ปัญหาของ SLAM

สำหรับ Probabilistic SLAM ที่ได้กล่าวถึงข้างต้นนั้นได้กล่าวถึงเพียงวิธีการในการประมาณตำแหน่งของหุ่นยนต์และแผนที่ด้วยวิธีการทางความน่าจะเป็น ซึ่งยังไม่อาจจะนำไปใช้ในการแก้ปัญหาจริงได้ การจะนำเอา SLAM ไปใช้ในการแก้ปัญหาจริงนั้น จะต้องหารูปแบบการอธิบายโมเดลการเปลี่ยนแปลงของสถานะหุ่นยนต์ (state transition model) (3.3) และ โมเดลการวัดจุดสังเกต (observation model) (3.2) ร่วมถึงวิธีการอธิบายการกระจายความน่าจะเป็นของสถานะของหุ่นยนต์ที่มีประสิทธิภาพเพียงพอที่

ในการคำนวนการประมาณค่า (3.4) และการปรับแก้ค่า (3.5) ซึ่งโดยทั่วไปแล้วโมเดลการเปลี่ยนแปลงของสถานะหุ่นยนต์ และโมเดลการวัดจุดสังเกต จะถูกเสนออยู่ในรูปโมเดลของ state-space ที่มีการกระจายแบบ Gaussian noise โดยวิธีแก้ปัญหาของ SLAM นั้นก็มีได้หลายวิธี ยกตัวอย่างเช่น

EKF SLAM : เป็นการแก้ปัญหา SLAM ด้วย Extended Kalman filter (EKF) (Durrant-Whyte and Bailey, 2006) ซึ่งสถานะของหุ่นยนต์และแผนที่จะถูกกำหนดให้เป็นสถานะของระบบของ kalman filter ซึ่งมีการกระจายความน่าจะเป็นแบบ normal และใช้การ predict และ update ของ kalman filter เพื่อการประมาณค่า และการปรับแก้ค่าสถานะของหุ่นยนต์และแผนที่

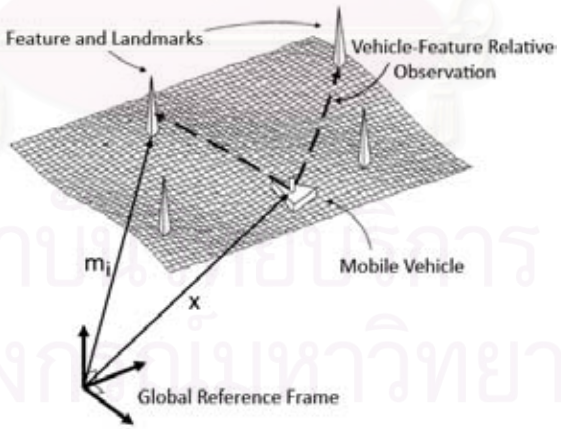
UKF SLAM : เป็นการแก้ปัญหา SLAM ด้วย Unscented Kalman filter (UKF) ซึ่งจะมีการทำงานคล้ายกับ Extended Kalman filter แต่วิธีการในการประมาณความน่าจะเป็นจะแตกต่างออกไป (Sunderhauf et al., 2007)

Fast SLAM : เป็นการแก้ปัญหา SLAM ด้วย Rao-Blackwellized particle filter ถูกนำเสนอโดย Montemerlo ในปี 2002 (Montemerlo et al., 2003) ซึ่งจะประมาณการกระจายความน่าจะเป็นด้วย particle จำนวนมาก

ซึ่งในงานวิจัยนี้จะใช้ Extended Kalman filter (EKF) ในการแก้ปัญหาเป็นหลัก ดังนั้นจึงจะอธิบายการทำงานของ Extended Kalman filter เป็นพิเศษ

3.5 EKF-SLAM

อัลกอริทึมของ SLAM นั้นอยู่บนสมมุติฐานที่ว่าเรารู้ Kinematic Model ของระบบ ซึ่งเป้าหมายของ SLAM คือการระบุตำแหน่งของหุ่นยนต์ ในแผนที่ที่สร้างขึ้นโดยอาศัยข้อมูลจากการวัดจุดสังเกต โดยที่หุ่นยนต์นั้นจะติดเซนเซอร์บนตัวหุ่น ซึ่งเซนเซอร์จะมีความสามารถในการวัดค่าจุดสังเกต สัมพัทธ์กับตัวหุ่นยนต์ ดังรูปที่ 3.5



รูปที่ 3.5: เซนเซอร์วัดค่าจุดสังเกตในสิ่งแวดล้อม สัมพัทธ์กับตัวหุ่นยนต์

Process Model

สถานะของระบบ จะประกอบด้วย ตำแหน่งและทิศทางของหุ่นยนต์ นอกจากนี้จะมี ตำแหน่งของจุดสังเกต (Landmark) โดยที่สถานะของหุ่นยนต์เขียนแทนด้วย c_k เมื่อเคลื่อนที่ผ่านสิ่งแวดล้อม สามารถสร้างเขียนโมเดลการเคลื่อนที่ได้ดังนี้

$$c_k = f_c(c_{k-1}, u_k) + w_k \tag{3.6}$$

โดยที่ $f_c(c_k)$ เป็น ฟังก์ชันการเคลื่อนที่ของหุ่นยนต์, u_k เป็น คำสั่งการเคลื่อนที่ของหุ่นยนต์, w_k เป็น noise ที่ไม่ขึ้นกับระบบหุ่นยนต์ มี (mean) เป็น 0 และ ความแปรปรวนร่วม (covariance) เป็น Q_k

ตำแหน่งของจุดสังเกต (Landmark) จุดที่ i จะแทนด้วย m_i โดยจะสมมุติว่าตำแหน่งที่แท้จริงของจุดสังเกต ไม่เปลี่ยนแปลงตามเวลา จะได้ state transition model สำหรับ landmark ตัวที่ i เป็น

$$m_{i,k} = m_{i,k-1} = m_i \tag{3.7}$$

เมื่อนำสถานะของหุ่นยนต์ และ ตำแหน่งของจุดสังเกตมาเขียนรวมกันจะได้

$$x_k = \begin{bmatrix} c_k \\ m_0 \\ \vdots \\ m_n \end{bmatrix} = \begin{bmatrix} f_c(c_{k-1}, u_k) \\ m_0 \\ \vdots \\ m_n \end{bmatrix} + \begin{bmatrix} w_k \\ 0 \\ 0 \\ 0 \end{bmatrix} = f(x_{k-1}, u_k) + W_k \tag{3.8}$$

โดยที่ $f(x_k)$ เป็น State transition model ของระบบทั้งหมด

Observation Model

เมื่อหุ่นยนต์ติดเซนเซอร์บนตัวหุ่น ซึ่งเซนเซอร์มีความสามารถในการวัดค่าจุดสังเกต สัมพัทธ์กับตัวหุ่นยนต์ดังนั้นจะได้ว่า observation model ของ landmark ตัวที่ i จะเขียนได้เป็น

$$z_{i,k} = h(x_k, m_i) + v_k \tag{3.9}$$

โดยที่ $h(x_k, m_i)$ เป็น ฟังก์ชันการวัดของเซนเซอร์, v_k เป็น noise ที่ไม่ขึ้นกับ model การวัด มี (mean) เป็น 0 และ ความแปรปรวนร่วม (covariance) เป็น R_k สาเหตุที่ฟังก์ชัน $h(x_k, m_i)$ รับสถานะของหุ่นยนต์เป็นพารามิเตอร์ด้วย นั่นเป็นเพราะการวัดค่าจุดสังเกตจะขึ้นอยู่กับตำแหน่งของหุ่นยนต์ด้วย

The Estimation Process

Extended Kalman Filter จะประมาณสถานะของระบบ x_k ด้วยวิธี recursive ซึ่งโมเดลการเปลี่ยนแปลงของระบบจะประมาณได้ตามสมการ 3.8 และ โมเดลการวัดค่าจุดสังเกตจะเป็นไปตามสมการ 3.9 สถานะของ SLAM ที่ต้องการจะประมาณ และความแปรปรวนร่วมของสถานะ จะเขียนได้เป็น

$$\hat{x}_k = E[x_k | Z_{0:k}] \tag{3.10}$$

$$\hat{e}_k = \hat{x}_k - x_k \tag{3.11}$$

$$P_k = E[\hat{e}_k \hat{e}_k^T | Z_{0:k}] \tag{3.12}$$

เมื่อ $\hat{x}_k$ เป็นค่าประมาณสถานะหุ่นยนต์และแผนที่ โดย $E[x_k | Z_{0:k}]$ คือ ค่าความคาดหวัง(Expected Value) ของ x_k เมื่อรู้ข้อมูลการวัดจุดสังเกตทั้งหมดตั้งแต่เวลา 0 ถึง k ส่วน $\hat{e}_k$ เป็นความคลาดเคลื่อน

การประมาณ และ P_k เป็นความแปรปรวนร่วมของสถานะ ซึ่งเมื่อสามารถประมาณสถานะหุ่นยนต์และความแปรปรวนร่วมได้ ก็หมายความว่าสามารถประมาณการกระจายความน่าจะเป็นของตำแหน่งของหุ่นยนต์และแผนที่ได้

อัลกอริทึมของ Extended kalman filter มีการทำงานอยู่ 3 ขั้นคือ Prediction, Observation และ Update

Prediction : เมื่อกำหนดให้ว่าโมเดลตามสมการ 3.8, 3.9, ค่าประมาณสถานะของระบบ $\hat{x}_{k-1}$ และ ค่าประมาณความแปรปรวนร่วมของระบบ P_{k-1} จะทำนายสถานะของระบบ, ค่าประมาณการวัดจุดสังเกตจุดที่ i และ ความแปรปรวนร่วมที่เวลา k หาได้เป็น

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k) \tag{3.13}$$
$$\hat{z}_{i,k} = h_i(\hat{x}_k^-) \tag{3.14}$$
$$P_k^- = F_k P_{k-1} F_k^T + Q_k \tag{3.15}$$

ตามลำดับ โดยที่ F_k คือ Jacobian ของ $f(\hat{x}_{k-1}, u_k)$ เทียบกับการเปลี่ยนแปลงของ $\hat{x}_{k-1}$ และ u_k

Observation : หลังจากการทำนายสถานะแล้ว เมื่อทราบค่าการวัดของจุดสังเกต $z_{i,k}$ จุดที่ i จะคำนวนค่าคลาดเคลื่อนการวัด (innovation) และ ความแปรปรวนร่วมค่าควาดเคลื่อน (innovation covariance matrix) ได้จาก

$$y_{i,k} = z_{i,k} - \hat{z}_{i,k} \tag{3.16}$$
$$S_k = H_k P_k^- H_k^T + R_k \tag{3.17}$$

ตามลำดับ โดยที่ H_k คือ Jacobian ของ $h(\hat{x}_k^-)$ เทียบกับการเปลี่ยนแปลงของ $\hat{x}_k^-$

Update : สถานะประมาณของระบบ หลังจากทราบค่าการวัดจุดสังเกต และ ความแปรปรวนร่วมที่เวลา k ได้เป็น

$$\hat{x}_k = \hat{x}_k^- + K_k y_k \tag{3.18}$$
$$P_k = P_k^- - K_k S_k K_k^T \tag{3.19}$$

โดยที่ตัวคูณ K_k หาได้จาก

$$K_k = P_k^- H_k^T S_k^{-1} \tag{3.20}$$

3.6 ปัญหาของ SLAM

ในการแก้ปัญหาของ SLAM นั้นมีปัญหาหลาย ๆ ที่ต้องพิจารณาเช่น

ปัญหาต้นทุนการคำนวน

ในการคำนวนการแก้ปัญหาของ SLAM นั้นจาก EKF SLAM จะเห็นได้ว่าถ้าหากปริมาณจุดสังเกต (Landmark) ในแผนที่มีจำนวนมากขึ้น เวลาที่ใช้ในการคำนวนจะเพิ่มเป็น $O(n^2)$ อีกทั้ง memory ที่ใช้ในการเก็บค่าสถานะยังเพิ่มเป็น $O(n^2)$ อีกด้วย ($O(n)$ สำหรับค่าสถานะ และ $O(n^2)$ สำหรับความแปรปรวนร่วม) โดยที่ n เป็นจำนวนจุดสังเกต (Landmark) ในแผนที่ ดังนั้นเมื่อแผนที่สิ่งแวดล้อมมีขนาดใหญ่มากอาจจะเกิดปัญหาสำหรับการคำนวนและ Storage ได้

ปัญหาการวัดค่าจุดสังเกต

ปัญหาของการวัดค่าจุดสังเกต ก็คือการต้องตีความข้อมูลที่วัดได้จากเซนเซอร์ เช่นถ้าใช้เซนเซอร์เป็นกล้องวิดีโอซึ่งรับข้อมูลเป็นภาพสองมิตินั้น การวัดค่าจุดสังเกตจะต้องใช้ข้อมูลภาพเพื่อการตีความว่าในบริเวณไหนตำแหน่งไหนในภาพคือ จุดสังเกตสำคัญที่ SLAM สนใจ นอกจากนี้ปัญหาของการวัดค่าจุดสังเกตยังมีปัญหาเกี่ยวกับการหาความสัมพันธ์ระหว่างค่าการวัดกับจุดสังเกตเดิมในแผนที่ (Data Association) เป็นต้นว่าในแผนที่มีจุดสังเกตอยู่จำนวนหนึ่ง และเมื่อเซนเซอร์วัดค่าจุดสังเกตได้ ปัญหาก็คือค่าการวัดที่วัดได้นั้นจะคู่กับจุดสังเกตเดิมจุดใดบ้าง และค่าการวัดใดเป็นของจุดสังเกตใหม่

ปัญหาการบรรยาย สิ่งแวดล้อมในแผนที่

วิธีการบรรยายสิ่งแวดล้อมในแผนที่นั้นมีได้หลายรูปแบบ ตามที่ได้นำเสนอไปในหัวข้อประเภทของ SLAM ไม่ว่าจะเป็น Feature-base, Grid maps, Scan matching ซึ่งในแต่ละวิธีการบรรยายก็จะมีความเหมาะสมสำหรับสิ่งแวดล้อมลักษณะต่าง ๆ กันไปซึ่งการเลือกวิธีบรรยายสิ่งแวดล้อมที่เหมาะสม ก็จะส่งถึงประสิทธิภาพของ SLAM ด้วย

ปัญหา Loop Closing

Loop Closing ก็คือการที่หุ่นยนต์เคลื่อนที่เป็นระยะทางไกลและวนกลับมาพบจุดเริ่มต้นจุดเดิมแล้วหุ่นยนต์สามารถรู้ได้ว่าหุ่นยนต์ได้กลับมายังจุดเริ่มต้นแล้ว ปัญหาของการ Close Loop ก็คือสำหรับหุ่นยนต์ที่มี sensor แบบ local แล้วเมื่อหุ่นยนต์เคลื่อนที่ห่างจากจุดเดิมเป็นระยะทางไกล ค่าความคลาดเคลื่อนของตำแหน่งหุ่นยนต์ย่อมที่มีค่าเพิ่มสูงด้วยเนื่องจาก error สะสม ดังนั้นเมื่อหุ่นยนต์วิ่งกลับไปที่ตำแหน่งเดิมจึงเป็นเรื่องยากที่ตำแหน่งของหุ่นยนต์ที่ประมาณได้จะทับกับจุดเดิมพอดี ซึ่งในการแก้ปัญหา Loop Closing นั้น ส่วนใหญ่มักใช้การรู้จำสิ่งแวดล้อมเข้ามาช่วยร่วมด้วยเพื่อหาความเป็นไปได้ที่หุ่นยนต์จะวิ่งกลับมาที่ตำแหน่งเดิมแล้ว

ปัญหาความสามารถในการวัด

ข้อมูลการวัดจากเซนเซอร์นั้น สำหรับเซนเซอร์ในแต่ละแบบจะให้ข้อมูล ที่มีลักษณะ ปริมาณ และจำนวนมิติที่แตกต่างกัน เช่นข้อมูลจากกล้องวิดีโอจะให้ข้อมูลสองมิติเป็นทิศทางของจุดสังเกต สำหรับข้อมูลจากอุปกรณ์วัดระยะด้วยเลเซอร์ จะให้ข้อมูลเป็นสองมิติเช่นกัน แต่จะเป็นข้อมูลทิศทางและข้อมูลความลึกของวัตถุในแนวระนาบ ซึ่งลักษณะข้อมูลการวัดทั้งหลายเหล่านี้ก็คือข้อจำกัดการวัดของอุปกรณ์เซนเซอร์ ซึ่งถ้าหาก SLAM ต้องการประมาณสถานะของระบบที่มีขนาดมิติสูง เช่นต้องการประมาณตำแหน่งหุ่นยนต์ในสามมิติ สถานะของหุ่นยนต์ก็จะประกอบด้วย การเคลื่อนตำแหน่ง (translation) 3 มิติ การหมุน (rotation) อีก 3 มิติ รวมกันเป็น 6 มิติ นอกจากนี้อาจจะมี ความเร็วหุ่นยนต์และความเร็วเชิงมุมอีก 6 มิติ รวมกันแล้วเป็น 12 มิติ ซึ่งถ้าหากข้อมูลจากเซนเซอร์มีมิติน้อยเกินไปก็จะเกิดความลำบากในการสร้างแผนที่และระบุตำแหน่งด้วย SLAM

บทที่ 4

อุปกรณ์สำหรับวัดค่า

อุปกรณ์วัดสำหรับค่าสิ่งแวดล้อม หรือเซนเซอร์ (Sensor) คืออุปกรณ์ที่มีหน้านี้รับข้อมูลจากสิ่งแวดล้อม ไม่ว่าจะเป็นข้อมูลภาพ ข้อมูลเสียง หรืออื่น ๆ แล้วจึงแปลงข้อมูลให้อยู่ในรูปสัญญาณไฟฟ้าเพื่อส่งต่อให้คอมพิวเตอร์สามารถรับรู้ถึงสิ่งแวดล้อมภายนอกได้ เซนเซอร์นั้นถือเป็นอุปกรณ์ที่หุ่นยนต์จำเป็นอย่างยิ่งสำหรับหุ่นยนต์ เพื่อที่จะได้รับรู้ถึงข้อมูลสิ่งแวดล้อมและส่งผลต่อการปฏิบัติต่อสิ่งแวดล้อมของหุ่นยนต์ได้อย่างถูกต้อง

สำหรับการระบุตำแหน่งและการสร้างแผนที่ของหุ่นยนต์นั้น หุ่นยนต์จะต้องใช้ข้อมูลจากอุปกรณ์เซนเซอร์เพื่อการวัดค่าสิ่งแวดล้อม สำหรับนำมาสร้างเป็นแผนที่ของสิ่งแวดล้อมและระบุตำแหน่ง อุปกรณ์เซนเซอร์สำหรับงานระบุตำแหน่งและการสร้างแผนที่มีอยู่หลายชนิด ซึ่งในวิทยานิพนธ์ฉบับนี้ จะจำเพาะเจาะจงใช้อุปกรณ์กล้องวิดีโอแบบออมนิ ในการระบุตำแหน่งและการสร้างแผนที่ ดังนั้นจึงจะขอเอ่ยถึงแต่อุปกรณ์กล้องวิดีโอแบบออมนิ เท่านั้น

4.1 กล้องวิดีโอแบบออมนิ

กล้องวิดีโอแบบออมนิ (Omni-directional Camera) แสดงได้ดังรูปที่ 4.1ก เป็นกล้องที่สามารถรับภาพได้รอบทิศทาง โดยอาศัยการสะท้อนภาพจากกระจกที่มีลักษณะต่าง ๆ เช่น กรวย พาราโบลา ไฮเปอร์โบลา และ อานม้า ซึ่งโดยทั่วไปแล้วจะให้มุมมองในแนวตั้งรอบตัวกล้อง 360 องศา และ มุมมองในมุมก้มเงยประมาณ 90-150 องศา ภาพจากกล้องออมนอนั้นแสดงได้ดังรูปที่ 4.1ข

(ก) (ข)

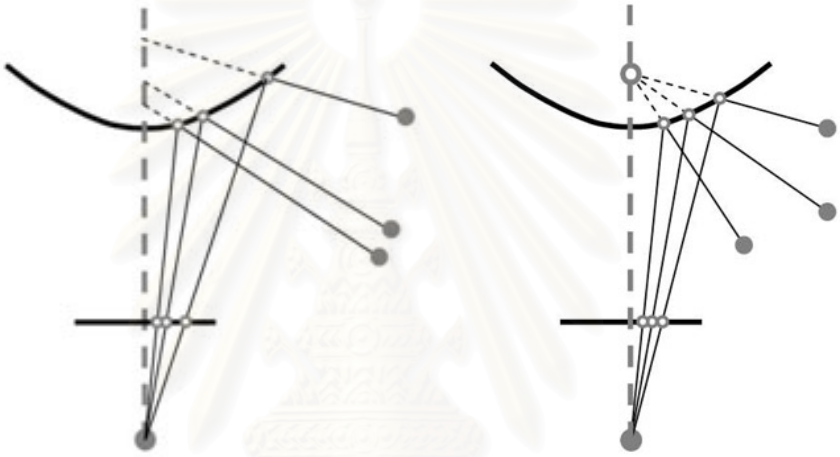
รูปที่ 4.1: (ก) ภาพกล้องวิดีโอแบบออมนิที่ใช้ในงานวิจัย (ข) ภาพถ่ายจากกล้องวิดีโอแบบออมนิ

กล้องออมนินั้นถูกใช้ในงานที่หลากหลาย ทั้งในงานของหุ่นยนต์อัตโนมัติ หรือการถ่ายภาพทั่วไป ทั้งนี้เนื่องจากกล้องออมนินั้นมีข้อดีเหนือกล้องวิดีโอทั่วไปคือ

- วัตถุหรือจุดสังเกตต่าง ๆ จะอยู่ในมุมมองของกล้องแทบทั้งสิ้น ยกเว้นกรณีที่มีการบดบังกัน ซึ่งการที่สามารถสังเกตเห็นจุดสังเกตได้เป็นเวลานานทำให้สามารถพัฒนาอัลกอริทึมที่ใช้ในการระบุตำแหน่งและสร้างแผนที่ที่แม่นยำได้

- มุมมองรอบทิศทางที่ได้จากกล้องออมนินั้นเกิดการการถ่ายภาพเพียงครั้งเดียว จึงทำให้ไม่มีปัญหาเรื่องการ synchronization ของภาพ
- กล้องออมนิไม่มีกลไกที่มีการเคลื่อนที่ จึงทำให้มีความทนทาน และมีความน่าเชื่อถือของภาพสูง เมื่อเทียบกับกล้อง pan-tilt ที่ใช้ระบบหันกล้องไปมาเพื่อเพิ่ม field of view

กล้องวิดีโอแบบออมนิที่ใช้กระจกในการขยายมุมมองภาพสามารถแบ่งออกได้เป็นสองประเภทใหญ่ ๆ คือ แบบ Central camers และ แบบ Non-central cameras สำหรับกล้องแบบ Non-Central นั้นรังสีของแสงที่เข้ามายังกล้องและตกกระทบบนกระจกจะไม่ตัดกันที่จุด ๆ เดียวกัน ดังในรูปที่ 4.2ก ส่วนกล้องแบบ Central ของแสงที่เข้ามายังกล้องและตกกระทบบนกระจกทุก ๆ เส้นจะตัดกันที่จุดจุดหนึ่ง ดังในรูปที่ 4.2ข ซึ่งคุณสมบัตินี้จะเรียกว่าจะเรียกว่า single effective viewpoint



(ก) Non-Central Cameras (ข) Central Cameras

รูปที่ 4.2: ลักษณะการตัดกันของลำแสงในกล้องออมนิแบบ Non-Central Cameras และ แบบ Central Cameras

สำหรับ ทฤษฎีของภาพแบบ Single viewpoint ได้อธิบายไว้อย่างละเอียดใน (Baker and Nayar, 1999) ซึ่งสรุปโดยย่อได้ว่า กล้องวิดีโอแบบออมนิแบบ central นั้นสามารถสร้างได้โดยการใช้กล้องรูเข็ม (pinhole camera หรือ เรียกอีกอย่างว่า perspective camera) ร่วมกับกระจกแบบ hyperbolic, parabolic, หรือ elliptical และเมื่อไม่นานมานี้ ยังมีเลนส์ตาปลา (fisheye lenses) ที่สามารถรับภาพได้ใกล้เคียงกับกล้องแบบ single effective viewpoint อีกด้วย ซึ่งระบบรับภาพดังกล่าวจะอาศัยเพียงแค่ เลนส์ตาปลา เพื่อขยายมุมมองภาพ (field of view) ของกล้อง โดยไม่ใช้กระจกในการสะท้อนภาพ กล้องในแบบแรก (ที่อาศัยกระจกในการรับภาพ) เรียกว่ากล้องออมนิ แบบ Catadioptric (Bruckstein and Richardson, 2000) ส่วนกล้องในแบบหลัง (ที่อาศัยเลนส์ตาปลาในการรับภาพ) จะเรียกว่า กล้องออมนิ แบบ Dioptric

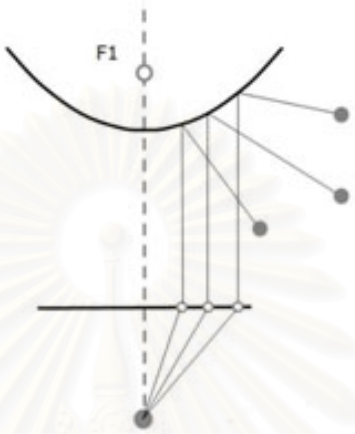
กล้องวิดีโอจะเป็นแบบ Central เมื่อใช้ร่วมกับอุปกรณ์ดังต่อไปนี้

กล้องวิดีโอร่วมกับ กระจกแบบ hyperbolic

จุดศูนย์กลางของกล้อง (ซึ่งก็คือจุดศูนย์กลางของเลนส์) จะต้องตรงกับจุดโฟกัสของไฮเปอร์โบลา ซึ่งจะรับประกันได้ว่า รังสีของแสงที่สะท้อนกระจกจะไปตัดกันที่ จุดเดียวกัน (ซึ่งจะเป็นจุด โฟกัส อันที่ใกล้กับกล้อง) ดังรูปที่ 4.2ข

กล้องวิดีโอร่วมกับ กระจกแบบ parabolic และ เลนส์แบบ orthographic

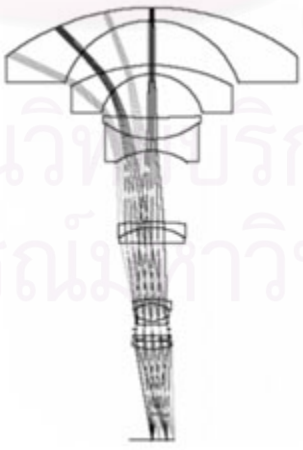
เมื่อใช้กระจก Parabolic รังสีสะท้อนที่มาจากสิ่งแวดล้อมเข้าสู่กล้อง จะขนานกับแกนของกระจก ซึ่งจะทำให้ กล้องแบบรูเข็ม (pin hole camera) ไม่สามารถใช้ได้ในกรณีนี้ อย่างไรก็ตาม ถ้าต้องการให้รังสีสะท้อน มาโฟกัส บน CCD ของกล้อง จำเป็นจะต้องใช้ เลนส์แบบ orthographic วางขั้นกลางระหว่างกล้องและกระจก parabolic ดังรูปที่ 4.3



รูปที่ 4.3: แสดงลำแสงที่เกิดจากการใช้กล้องร่วมกับ กระจกแบบ parabolic และ เลนส์แบบ orthographic

กล้องร่วมกับ เลนส์ตาปลา

ทางเลือกนี้จะเป็นวิธีที่จะขยายมุมมองภาพ (field of view) ของกล้อง โดยไม่ใช้กระจกในการสะท้อน ซึ่งจะเพิ่มเลนส์ตาปลาเข้าไปหน้ากล้อง ซึ่ง เลนส์ตาปลา จะสามารถขยายมุมมองภาพ (field of view) ของกล้องได้ถึง 190 องศา ดังรูปที่ 4.4 โดยทั่วไปแล้วจะไม่นิยมใช้เลนส์ตาปลาในระบบกล้องแบบ central แต่ถึงอย่างไรก็ตาม เลนส์ตาปลา ก็มีความใกล้เคียงกับกล้องแบบ single view point



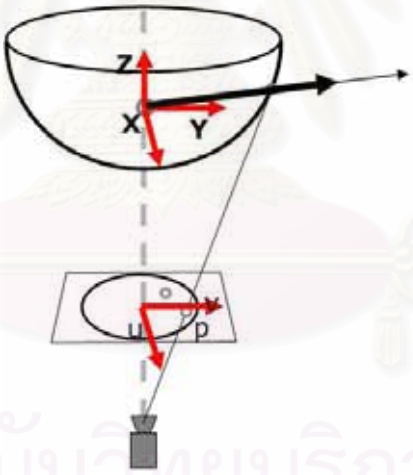
รูปที่ 4.4: แสงการหักเหของแสงในเลนส์ตาปลา

4.1.1 การหาพารามิเตอร์การเรียงตัวชิ้นส่วนของกล้อง

สำหรับกล้องวิดีโอทั่วไปแล้ว หลักการทำงานของกล้องก็คือการโปรเจควัตถุในพิกัดสามมิติให้มาอยู่ในพิกัด สองมิติของรูป ซึ่งการเรียงตัวของชิ้นส่วนของกล้องในแต่ละกล้องก็จะแตกต่างกันไปตามความโค้งของเลนส์ซึ่งไม่เท่ากันในแต่ละกล้อง

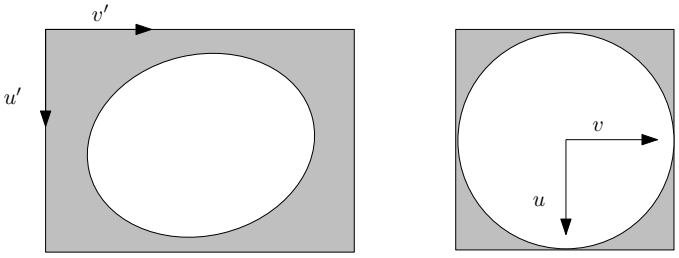
ข้อมูล Vision หรือข้อมูลที่ได้จากกล้องวิดีโอนั้นมีการใช้งานทางด้าน Robotics เป็นอย่างมากซึ่งในการใช้งานนั้น มีความจำเป็นอย่างยิ่งที่จะต้องทราบค่าพารามิเตอร์ และฟังก์ชัน Mapping ระหว่างตำแหน่งของวัตถุในสามสิติ และ ตำแหน่งจุดสีในภาพ เพื่อความแม่นยำในการคำนวณ ดังนั้นสิ่งที่ขาดไปมิได้ สำหรับงานทางด้าน Vision ก็คือการ Calibrate เพื่อหาค่าพารามิเตอร์ และ Mapping Function ของกล้อง

ในงานวิจัย (Scaramuzza et al., 2006a,b) ได้นำเสนอหลักการในการหาพารามิเตอร์ของกล้องออมนิ (Omni Camera Calibration) ซึ่งจะเป็นการหาความสัมพันธ์ระหว่าง จุดสองมิติในภาพ (p) กับตำแหน่งในโลกสามมิติ (P) ที่สะท้อนมาจากกระจกดังรูปที่ 4.5 ซึ่งในงานวิจัย (Scaramuzza et al., 2006b) มีสมมุติฐานว่า กล้องจะเป็นแบบ central ดังนั้นจะต้องมีจุดจุดหนึ่งในกระจกซึ่งเป็นจุดตัดของ รังสีของแสงทั้งหมด ซึ่งจะกำหนดให้จุดจุดนั้นเป็นจุดกำเนิดของระบบโคออดิเนต XYZ ของกล้อง, กล้องและกระจกได้วางอยู่ในแนวเดียวกัน, กระจกมีความสมมาตรในการหมุนในแกนของมัน และ การ Distortion ของเลนส์จะไม่ถูกพิจารณา ซึ่งเหตุผลที่ไม่สนใจการ Distortion ของเลนส์เป็นเพราะ กล้องออมนิแบบที่ใช้กระจกนั้นโดยปรกติแล้วจะต้องการความยาวโฟกัสที่มากเพื่อที่จะได้ โฟกัสภาพได้บนกระจก ดังนั้นการ Distortion ของเลนส์สามารถละเลยได้



รูปที่ 4.5: แสดงการโปรเจคของแสงที่สะท้อนเข้ามายังกล้องออมนิ

โมเดลของกล้องออมนิ



รูปที่ 4.6: แสดงโคออดิเนตของพิกเซล (u', v') และโคออดิเนตที่แก้ไขการ distortion แล้ว (u, v)

สมมุติให้กล้องและกระจกวางอยู่ในแนวเดียวกัน จากรูปที่ 4.6 กำหนดให้ (u', v') เป็นโคออดิเนตของพิกเซล ส่วน (u, v) เป็นโคออดิเนตที่แก้ไขการ distortion แล้ว จะเขียนการแปลง u' และ v' ได้ดังนี้

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} c & d \\ e & 1 \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} xc' \\ yc' \end{bmatrix} \tag{4.1}$$

จะได้ว่า

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1/det & -d/det \\ -e/det & c/det \end{bmatrix} \cdot \begin{bmatrix} u' - xc' \\ v' - yc' \end{bmatrix}, det = c - e \cdot d \tag{4.2}$$

ซึ่ง c, d, e, xc' และ yc' เป็น parameter ที่ได้มาจากการ calibrate โดย xc' และ yc' จะเป็นจุดศูนย์กลางภาพ ส่วน c, d และ e จะเป็น affine transform parameter

จากภาพที่ 4.5 จะหา vector จากกล้องชี้ไปยังจุดใน 3 มิติได้เป็น

$$r = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix} \tag{4.3}$$

$$\rho = \sqrt{u^2 + v^2} \quad , f(\rho) = a_0 + a_0\rho + a_0\rho^2 + a_0\rho^3 + a_0\rho^4 + \cdots \tag{4.4}$$

จากสมการข้างต้นฟังก์ชัน $f(\rho)$ จะถูกประมาณด้วย Polynomial function โดยค่า ρ จะมีค่าเป็น $\sqrt{u^2 + v^2}$ ซึ่งจะเป็นค่าระยะทางจากจุดศูนย์กลางกล้องโดยจากภาพที่ 4.5 แสดงให้เห็นว่าระยะทางจากจุดศูนย์กลางจะสอดคล้องกับความเอียงของกระจก ซึ่งจะส่งผลกับค่าในแกน z ด้วย ส่วนสัมประสิทธิ์ของการประมาณจะหาได้จากการ calibrate โดยจะใช้วิธี Least Square ในการประมาณซึ่งในที่นี้จะประมาณด้วย polynomial degree 4

บทที่ 5

การระบุตำแหน่งพร้อมกับการสร้างแผนที่ในสามมิติด้วยกล้องวิดีโอแบบออมนิ

วิธีการระบุตำแหน่งพร้อมกับการสร้างแผนที่ในสามมิติด้วยกล้องวิดีโอแบบออมนิประกอบไปด้วยขั้นตอนสองส่วนได้แก่ ส่วนการวิเคราะห์ข้อมูลภาพจากกล้องวิดีโอแบบออมนิ ซึ่งจะทำหน้าที่ตีความข้อมูลภาพที่ได้รับจากกล้องออมนิว่ามีลักษณะของสิ่งแวดล้อมรอบตัวเป็นอย่างไร โดยสิ่งแวดล้อมจะถูกบรรยายได้โดยจุดสังเกตในภาพ และส่วนที่สองก็คือส่วนการระบุตำแหน่งพร้อมกับการสร้างแผนที่ในสามมิติ ด้วยข้อมูลการวัดของกล้องวิดีโอแบบออมนิ ซึ่งได้มาจากการวิเคราะห์ข้อมูลภาพในส่วนแรก

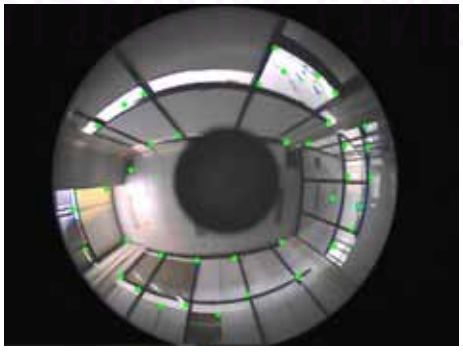
5.1 การวิเคราะห์ข้อมูลภาพจากกล้องวิดีโอแบบออมนิ

ข้อมูลภาพที่วัดได้จากกล้องวิดีโอนั้นประกอบด้วยข้อมูลจุดสีในสองมิติจำนวนมากโดยข้อมูลจุดสีในแต่ละจุดนั้นแสดงถึงความเข้มของแสงที่องศาต่าง ๆ เทียบกับหน้ากล้อง ซึ่งข้อมูลทั้งหลายเหล่านี้ถือเป็นข้อมูลฟุ่มเฟือย เพราะการจะอธิบายสภาพแวดล้อม ณ ขณะนั้นใช้การอธิบายเพียงลักษณะเฉพาะของสิ่งแวดล้อมที่สนใจก็เพียงพอแล้ว เช่น ในภาพของห้องห้องหนึ่งซึ่งประกอบด้วยจุดสีจำนวนมาก อาจจะอธิบายสั้น ๆ ได้ว่ามีวัตถุประเภทไหนอยู่ในตำแหน่งใดของภาพบ้าง ซึ่งหน้าที่ของส่วนวิเคราะห์ข้อมูลภาพก็คือการตีความภาพเพื่อลดข้อมูลฟุ่มเฟือยให้เหลือเฉพาะข้อมูลที่ SLAM สนใจ

ลักษณะเฉพาะของสิ่งแวดล้อมในที่นี้ก็คือจุดสังเกตในแผนที่ของ SLAM นั่นเอง โดยเมื่อจุดสังเกตซึ่งมีลักษณะเฉพาะเหล่านี้มาปรากฏบนภาพ ก็น่าจะเป็นตำแหน่งที่มีลักษณะเฉพาะของภาพ ในหัวข้อนี้จะขอบรรยายวิธีการตรวจหาจุดที่มีลักษณะเฉพาะดังกล่าว ซึ่งเมื่อหาจุดสังเกตเหล่านั้นได้แล้ว ก็จะแปลงข้อมูลให้อยู่ในรูปข้อมูลการวัดเพื่อนำไปใช้ใน SLAM ต่อไป

5.1.1 การตรวจหาจุดสังเกต (Features Detection)

จุดสังเกตในงานวิจัยนี้จะถือว่าเป็นจุดสำคัญที่ปรากฏในภาพ ซึ่งลักษณะของจุดสำคัญเหล่านั้นอาจเป็นตำแหน่งของมุม ขอบ หรือจุดตัดในสิ่งแวดล้อม ซึ่งเมื่อจุดเหล่านั้นมาปรากฏบนภาพแล้วก็จะเป็นจุดที่มีลักษณะสำคัญที่สามารถแยกแยะได้ ดังแสดงในรูปที่ 5.1 จะเห็นได้ว่าจุดที่จะถือเป็นจุดสังเกตในภาพ



รูปที่ 5.1: feature ในภาพจากกล้องออมนิ

จะเป็นบริเวณมุมในสิ่งแวดล้อม ซึ่งมีลักษณะพิเศษเฉพาะแตกต่างจากจุดอื่นโดยสามารถตรวจจับได้ง่าย และจะปรากฏเสมอไม่ว่าจะสังเกตจางมุมมองใด ซึ่งในงานวิจัยนี้จะใช้จุดดังกล่าวเป็นจุดสังเกตุสำคัญในสิ่งแวดล้อม โดยข้อมูลที่บรรยายความแตกต่างของจุดสังเกตแต่ละจุดก็คือข้อมูลรอบ ๆ จุดสังเกตนั้น ๆ และแผนที่ที่สร้างได้นั้นก็จะเป็นแผนที่แสดงตำแหน่งของจุดสังเกตเหล่านี้ใน 3 มิติ

วิธีการในการตรวจหาจุดสังเกตจะใช้ Harris corner detector (Harris and Stephens, 1988) ในการตรวจวัดซึ่งขั้นตอนในการตรวจหาจะอธิบายได้ดังนี้

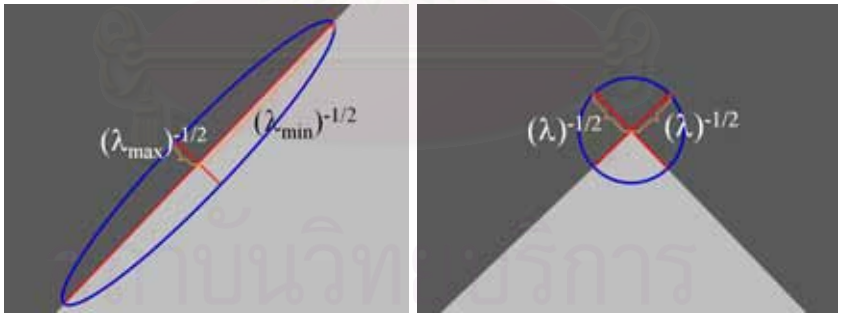
1. หาค่าความเป็นมุม (corner response) ของแต่ละพิกเซลในภาพโดยใช้ Harris Operator (ในที่นี้ได้อธิบายโดยละเอียดในภาคผนวก ก) โดยค่าความเป็นมุม (corner response) (R) จะคำนวนหาได้จาก

$$R = det - k \cdot trace^2 \quad (5.1)$$

$$det = \lambda_1 \lambda_2 \quad (5.2)$$

$$trace = \lambda_1 + \lambda_2 \quad (5.3)$$

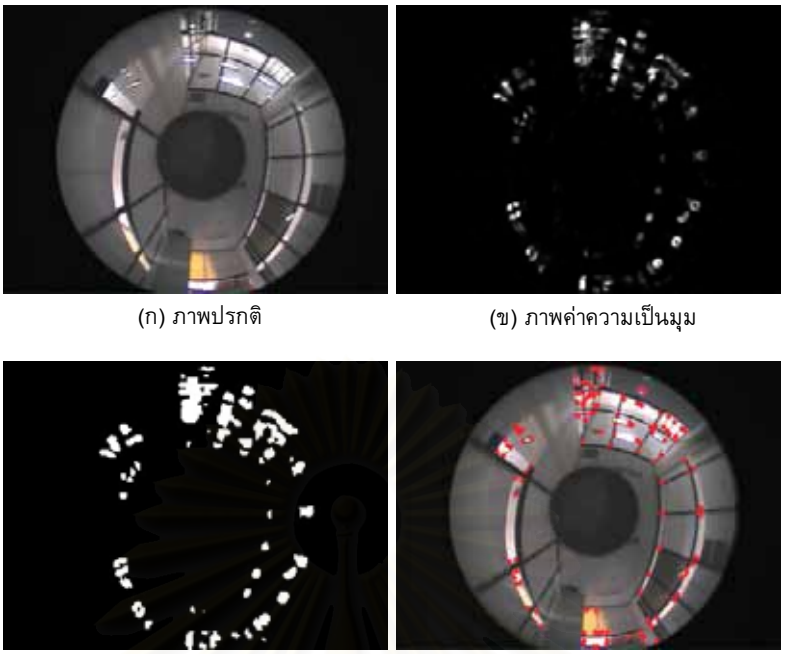
ซึ่ง k ในที่นี้จะเป็นค่าที่ผู้ใช้กำหนดเอง (empirical constant) จะมีค่าประมาณ 0.04-0.06 ส่วน λ_1 และ λ_2 จะเป็นค่า eigenvalue ณ พิกเซลนั้น ๆ ซึ่งค่า eigenvalue จะบ่งชี้ค่าความเปลี่ยนแปลงของสีในทิศทางสองทิศทางที่ตั้งฉากกัน ยกตัวอย่างเช่นในรูปที่ 5.2ก eigenvalue มีค่าสูงมากหนึ่งค่าแสดงว่าบริเวณนั้นมีการเปลี่ยนแปลงของค่าสีมากในหนึ่งทิศทางแสดงว่าบริเวณนั้นเป็น edge ในสิ่งแวดล้อม ส่วนในรูปที่ 5.2ข มีค่า eigenvalue สูงทั้งสองค่า แสดงว่าบริเวณนั้นมีการเปลี่ยนแปลงของค่าสีมากทั้งสองทิศทาง ชี้ให้เห็นว่าที่จุดจุดนั้นเป็น corner ในสิ่งแวดล้อม ซึ่งค่าความเป็นมุม (corner response) จะแสดงได้ดังรูปที่ 5.3ข



(ก) (ข)

รูปที่ 5.2: ค่า eigenvalue ของ edge และ corner

2. หาจุดในภาพที่มีค่าความเป็นมุม (corner response) เป็น local maxima (มีค่าสูงสุดในบริเวณใกล้เคียง) และมีค่าความเป็นมุมสูงกว่าค่า threshold ค่าหนึ่ง ดังแสดงได้ในรูป 5.3ค และ 5.3ง
3. เลือกคัดจุดที่มีค่าความเป็นมุม (corner response) ต่ำและอยู่ใกล้กับจุดอื่นมากเกินไปทิ้ง ก็จะได้จุดที่มีลักษณะเฉพาะที่จะใช้บรรยายสิ่งแวดล้อม



(ก) ภาพปรกติ (ข) ภาพค่าความเป็นมุม

(ค) รูป threshold (ง) feature ที่ตรวจพบได้

รูปที่ 5.3: ขั้นตอนของ Harris Detector

5.1.2 การหาค่าการวัดของจุดสังเกต (Features Measurement)

ค่าการวัดของจุดสังเกตคือค่าการวัดที่ได้จากภาพจากกล้องออมนิ ซึ่งจะถูกนำไปใช้เป็น measurement สำหรับ SLAM ต่อไป ส่วนการหาพารามิเตอร์การเรียงตัวชิ้นส่วนของกล้อง (Camera Calibration) นั้นแสดงให้เห็นว่า การหา vector ที่ชี้ไปยังจุดใน 3 มิติจากพิกเซลใด ๆ ในภาพนั้นหาได้โดยง่ายจากสมการ

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1/det & -d/det \\ -e/det & c/det \end{bmatrix} \cdot \begin{bmatrix} u' - xc' \\ v' - yc' \end{bmatrix}, det = c - e \cdot d \tag{5.4}$$

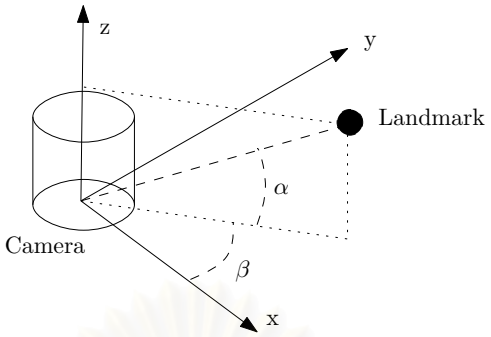
$$r = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix} \tag{5.5}$$

$$\rho = \sqrt{u^2 + v^2}, f(\rho) = a_0 + a_0\rho + a_0\rho^2 + a_0\rho^3 + a_0\rho^4 \tag{5.6}$$

ในที่นี้ $r = \begin{bmatrix} r_x & r_y & r_z \end{bmatrix}^T$ เป็น vector ที่ชี้ไปยังจุดใน 3 มิติ, (u', v') เป็นโคออดิเนตของพิกเซลที่วัดได้ในภาพจากกล้องออมนิ, (u, v) เป็นโคออดิเนตที่พิจารณา distortion แล้ว ส่วน c, d, e, xc' และ yc' เป็นพารามิเตอร์ที่ได้มาจากการ calibrate โดย xc' และ yc' จะเป็นจุดศูนย์กลางภาพ ส่วน c, d และ e จะเป็น affine transform matrix

อันที่จริงแล้วค่าของจุดในพิกัดของภาพที่หาได้ในหัวข้อ "การตรวจหาจุดสังเกต" นั้น ก็สามารถนำไปใช้เป็นค่าการวัดได้ แต่การจะคำนวนหาจุดในพิกัดของภาพเมื่อทราบจุดใน 3 มิติจะคำนวนหาได้ยาก เนื่องจากจะต้องหารากของสมการ polynomial degree 4 ทำให้โมเดลการวัดของจุดในพิกัดของภาพจากกล้องวิดีโอแบบออมนินั้นจึงความยุ่งยาก เกินกว่าจะใช้เป็นโมเดลสำหรับ SLAM ได้ดังนั้นจึงจะต้องแปลงค่าการวัดของจุดสังเกตให้อยู่ในรูปแบบอื่นเสียก่อน ซึ่งค่าการวัดของจุดสังเกตที่จะใช้อธิบายการวัดภาพ

จากกล้องในงานวิจัยนี้จะใช้เป็น ค่าการวัดเชิงองศาในรูปแบบ yaw, pitch angles (α, β) ดังแสดงในรูปที่ 5.4



รูปที่ 5.4: ค่าการวัดเชิงองศารูปแบบ yaw, pitch angles

โดยการคำนวนนั้นสามารถหาได้จาก

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \arctan(r_z/\sqrt{r_x^2 + r_y^2}) \\ \arctan(r_y/r_x) \end{bmatrix} \tag{5.7}$$

ซึ่งค่า yaw, pitch angles (α, β) นั้นก็จะนำไปใช้เป็นค่าการวัดสำหรับ SLAM ต่อไป

5.1.3 การหาความสัมพันธ์ของจุดสังเกต (Features Association)

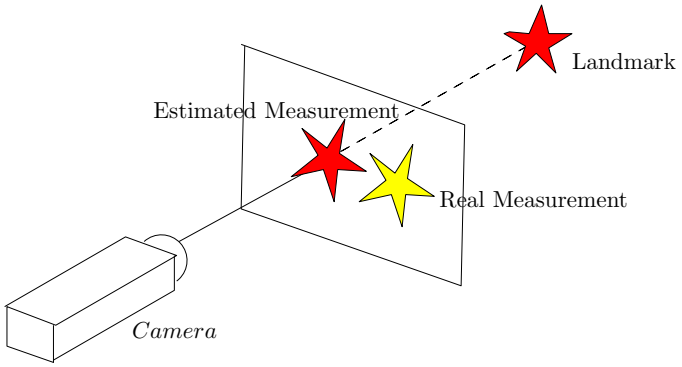
การหาความสัมพันธ์ของจุดสังเกตก็คือการหาความสัมพันธ์ระหว่าง feature ที่ตรวจวัดได้ในภาพปัจจุบัน กับจุดสังเกต (landmark) ที่ตรวจวัดได้ในอดีตที่ประมาณได้โดย SLAM ซึ่งจะเป็นตำแหน่งที่ใช้อธิบายแผนที่ของสิ่งแวดล้อม การหาความสัมพันธ์ของจุดสังเกตนั้น ก็เพื่อการพิจารณาว่า feature ในภาพปัจจุบันนั้นสอดคล้องกับจุดสังเกตเก่า (landmark) จุดไหนในอดีต หรือว่า feature ในภาพปัจจุบันเป็นการวัดของจุดสังเกตใหม่ ซึ่งเมื่อทราบความสัมพันธ์ของจุดสังเกตแล้ว ก็จะสามารถนำข้อมูลการวัดจากภาพไปปรับแก้ค่าประมาณของสถานะของกล้องและแกนที่ได้อย่างถูกต้อง

การหาความสัมพันธ์ของจุดสังเกตโดยทั่วไปสามารถแบ่งออกได้เป็นสองประเภท คือการหาความสัมพันธ์จากค่าประมาณการวัดจุดสังเกต (Features association from measurement) และ การหาความสัมพันธ์จากภาพโดยตรง (Features association from Image)

5.1.3.1 การหาความสัมพันธ์จากค่าประมาณการวัดจุดสังเกต

การหาความสัมพันธ์จากค่าประมาณการวัดจุดสังเกต คือการวัดเทียบระหว่างค่าการวัดจริง กับค่าการวัดที่ประมาณได้จากตำแหน่งของจุดสังเกตในแผนที่ ยกตัวอย่างเช่นในภาพที่ 5.5 จะเห็นได้ว่า รูปดาวสีเทาเข้มในภาพ เป็นข้อมูลการวัดที่ประมาณได้จาก ตำแหน่งของกล้องและ ตำแหน่งของจุดสังเกตในแผนที่ ส่วนรูปดาวสีเทาอ่อนในภาพนั้นเป็นค่าการวัดจริงได้ได้จากกล้อง สำหรับวิธีการวัดเทียบนั้นจะใช้การวัดความต่างด้วย mahalanobis distance

Mahalanobis distance นั้นเป็นวิธีการในการวัดระยะซึ่งถูกนำเสนอโดย P. C. Mahalanobis ในปี 1936 (Mahalanobis, 1936) ซึ่งวิธีการในการวัดระยะจะแตกต่างไปจาก Euclidean distance เพราะนอกจากจะพิจารณาระยะทางแบบ spatial แล้วยังพิจารณาความคลาดเคลื่อนของตำแหน่งอีกด้วย แนวคิดของ Mahalanobis distance ก็คือ สำหรับค่าประมาณการวัดค่าหนึ่ง ถ้าหากว่าค่าประมาณการวัดค่านี้มี



รูปที่ 5.5: ค่าการวัดจริง และค่าการวัดที่ประมาณได้จากตำแหน่งของจุดสังเกตในแผนที่

การกระจายความน่าจะเป็นที่มากแล้ว ความน่าจะเป็นที่ค่าการวัดจากภาพจริง ค่าหนึ่งจะสอดคล้องกับค่าประมาณค่านี้ย่อมมีความน่าจะเป็นที่สูงขึ้นด้วย (หมายความว่าวัด distance แล้วได้ค่าน้อย) ตรงกันข้าม ถ้าหากว่าค่าประมาณการวัดนั้นมีการกระจายความน่าจะเป็นที่น้อย หมายความว่าค่าประมาณนั้นสามารถประมาณได้แม่นยำมาก ๆ ค่าการวัดจากภาพจริงที่จะสอดคล้องกับค่าประมาณนั้น จะต้องมี Euclidean distance ที่ใกล้กับค่าประมาณมาก ๆ จนแทบจะเป็นศูนย์ จึงจะถือได้ว่าค่าการวัดนั้นสอดคล้องกัน ซึ่ง Mahalanobis distance นั้นจะหาได้จากสมการ

$$D_M(x) = \sqrt{(x-\mu)^T S^{-1}(x-\mu)} \tag{5.8}$$

โดย $D_M(x)$ เป็นค่า Mahalanobis distance, x เป็นค่าการวัดจริง, μ เป็นค่าที่ได้จากการประมาณ ส่วน S เป็นค่าความแปรปรวนร่วม (covariance) ของ μ

วิธีการหาความสัมพันธ์ของจุดสังเกตด้วย mahalanobis distance นั้นเป็นวิธีการหาความสัมพันธ์ที่เป็นที่นิยมมาก แต่ในงานวิจัยนี้จะ**ไม่ใช้**การหาความสัมพันธ์จากค่าประมาณการวัดจุดสังเกต เนื่องจากว่ามีวิธีการหาความสัมพันธ์ที่ดีกว่า ซึ่งก็คือ การหาความสัมพันธ์จากภาพโดยตรง

5.1.3.2 การหาความสัมพันธ์จากภาพโดยตรง

การหาความสัมพันธ์ของจุดสังเกตจากภาพโดยตรง ก็คือการหาความสัมพันธ์ของ feature ในภาพปัจจุบันกับ feature จากภาพในอดีต ซึ่งการหาความสัมพันธ์จากภาพนั้นเป็นข้อดีของการใช้กล้องวิดีโอเป็นเซนเซอร์ เนื่องจาก feature ที่หาได้จากภาพนั้นจะสามารถระบุความต่างระหว่าง Feature ได้ จึงสามารถตรวจหาความสอดคล้องของ Feature ได้ง่าย

วิธีการหาความสัมพันธ์ของ feature ระหว่างภาพนั้นมีอยู่หลายวิธีด้วยกัน ซึ่งวิธีที่เป็นที่นิยมก็คือการใช้ SIFT (Lowe, 2004) เพื่อหาความสัมพันธ์ของ feature ระหว่างภาพ แต่ในงานวิจัยนี้จะใช้ Optical Flow (Bouguet, 2000) เพื่อหาความสัมพันธ์ของ feature ซึ่งวิธีการหาความสัมพันธ์ด้วย Optical Flow นั้นจะแตกต่างจากการหาความสัมพันธ์ด้วย SIFT ซึ่งสำหรับการหาความสัมพันธ์ด้วย SIFT นั้นจะใช้การเปรียบเทียบความต่างระหว่าง description ของ feature ของภาพสองภาพในทุก ๆ คู่ซึ่ง feature คู่ไหนที่มีความต่างน้อยสุดก็จะถือว่า feature คู่นั้นสอดคล้องกัน แต่ Optical Flow นั้นจะเป็นการประมาณการเลื่อนตำแหน่งของ feature ทำให้รู้ว่า feature ของภาพในอดีตนั้นได้เลื่อนมาเป็น feature ณ ภาพปัจจุบันที่ตำแหน่งไหน ซึ่งวิธีการของ optical Flow นั้นจะทำให้รู้ได้ว่า feature ของภาพปัจจุบันนั้นสอดคล้องกับ feature ของภาพในอดีต feature ไหนบ้าง โดยไม่ต้องทำการตรวจหา Feature ใหม่ ดังนั้นขั้นตอนการตรวจหาจุดสังเกตโดยใช้ Harris Corner Detector นั้นจะใช้สำหรับการตรวจหา Feature ใหม่เท่านั้น

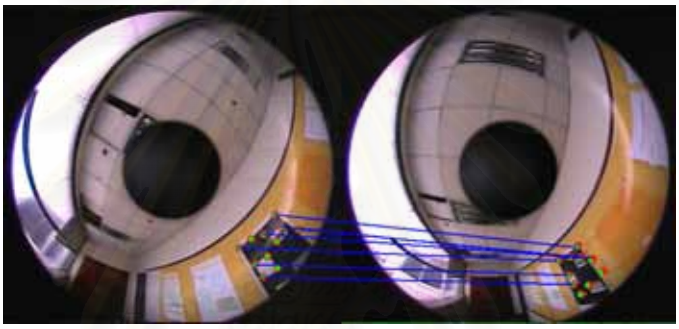
ขั้นตอนการหาความสัมพันธ์จากภาพในงานวิจัยนี้สามารถแบ่งได้เป็นสองขั้นตอน คือ การติดตามการเลื่อนตำแหน่งของ feature และ การปรับแก้ความถูกต้องของตำแหน่ง feature

การติดตามการเลื่อนตำแหน่งของ feature

การติดตามการเลื่อนตำแหน่งของ feature คือการหาว่าจุดสังเกตเดิม (landmark) ที่ปรากฏอยู่ในแผนที่นั้น น่าจะปรากฏที่ตำแหน่งใดในภาพจากกล้องวิดีโอแบบออมนิ ซึ่งวิธีการติดตามการเลื่อนตำแหน่งของ feature นั้นแบ่งออกเป็นสองวิธีด้วยจุดมุ่งหมายที่แตกต่างกันคือ

วิธีที่ 1 การติดตามการเคลื่อนที่ด้วย Optical Flow

วิธีนี้จะใช้ pyramids Lucas-Kanade optical flow (Bouguet, 2000) (ซึ่งได้อธิบายโดยละเอียดในภาคผนวก ข) เพื่อติดตามการเลื่อนตำแหน่งของ features ระหว่างภาพที่เวลาก่อนหน้า และภาพที่เวลาปัจจุบัน ดังแสดงได้ในภาพที่ 5.6



รูปที่ 5.6: การติดตามการเลื่อนตำแหน่งของ features ระหว่างภาพที่เวลาก่อนหน้า และภาพที่เวลาปัจจุบันด้วย optical flow

วิธีที่ 2 การติดตามการเคลื่อนที่ด้วยการประมาณค่าการวัดจากสถานะของกล้องและตำแหน่งของจุดสังเกต

วิธีนี้จะเป็นการประมาณตำแหน่งของจุดสังเกตที่จะปรากฏบนภาพในกรณีที่ไม่สามารถติดตามได้ด้วย Optical Flow ซึ่งอาจเป็นเพราะ จุดสังเกตนั้นได้หลุดออกไปจากภาพแล้ว หรือจุดสังเกตนั้นถูกบังด้วยสิ่งแวดล้อม นั้นหมายความว่าที่เวลาก่อนหน้าจะไม่มีข้อมูลการวัดสำหรับจุดสังเกตนั้น ๆ (เพราะตำแหน่งจุดสังเกตนั้นไม่ปรากฏในภาพ) สำหรับการประมาณตำแหน่งของจุดสังเกตที่จะปรากฏบนภาพด้วยโมเดลการวัดและพารามิเตอร์ของกล้องที่ได้จากการ Calibrate ซึ่งแน่นอนว่าตำแหน่งที่ประมาณได้นั้นย่อมไม่ถูกต้องและนำไปใช้ SLAM ไม่ได้ ดังนั้นหลังการประมาณแล้วจะต้องทำการปรับแก้ความถูกต้องของตำแหน่ง feature ซึ่งเป็นขั้นตอนต่อไป

การปรับแก้ความถูกต้องของตำแหน่ง feature

ในการติดตามการเคลื่อนที่ของ feature นั้นความคาดเคลื่อนของการประมาณตำแหน่งของ feature ย่อมเกิดขึ้นได้ซึ่งเป็นข้อเสียของการหาความสัมพันธ์ของจุดสังเกตด้วย Optical Flow ซึ่งการติดตามการเคลื่อนที่ของ feature ไปเรื่อย ๆ โดยไม่ปรับแก้นั้นความคาดเคลื่อนของตำแหน่งย่อมจะสะสม หรือการประมาณด้วยค่าการวัดจากสถานะของกล้องและตำแหน่งของจุดสังเกต ย่อมไม่สามารถให้ค่าที่ถูกต้องได้อยู่แล้ว ดังนั้นจุดมุ่งหมายของการปรับแก้ความถูกต้องของตำแหน่ง feature ก็คือการทำให้ความคาดเคลื่อนของตำแหน่ง feature นั้นคงที่และไม่ขึ้นกับเวลา นั่นหมายความว่าไม่ว่ากล้องจะเคลื่อนที่ไปนานเท่าใด feature ที่ตรวจวัดได้จะต้องเป็นตำแหน่งเดิมเสมอ

สำหรับการปรับแก้ตำแหน่ง feature นั้นจำเป็นจะต้องใช้ข้อมูลอ้างอิงของ feature เพื่อใช้ในการวัดเทียบการปรับแก้ตำแหน่ง ซึ่ง feature อ้างอิงในที่นี้จะใช้เป็น feature ของภาพในอดีต โดยจะถือว่า feature อดีตนั้นคือค่าการวัดของจุดสังเกตในโลกจริง นั่นหมายความว่าค่าการวัดจุดสังเกตของ feature ที่ใช้อ้างอิงนั้นจะมีความแม่นยำในการวัดมาก ซึ่งการการปรับแก้ตำแหน่ง feature เทียบกับ feature อ้างอิงอันเดิมนั้นจะช่วยทำให้ความคาดเคลื่อนของตำแหน่ง feature นั้นคงที่ได้

การปรับแก้ตำแหน่งนั้นจะใช้ Pyramids template matching ในการคำนวนบริเวณรอบ ๆ จุดที่ต้องการปรับแก้เพื่อหาจุดที่มีความสอดคล้องของภาพกับ template สูงสุด โดยจะแสดงขั้นตอนการทำงานได้ดังนี้

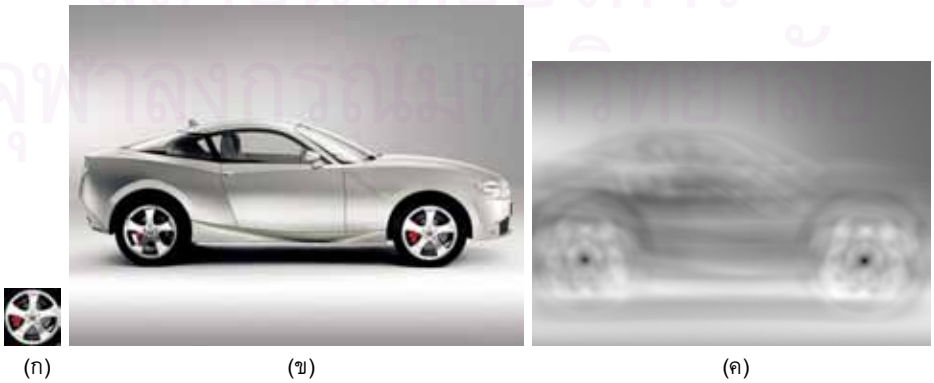
1. การเลือก template

Template ที่จะใช้ในการปรับแก้ตำแหน่ง feature ในที่นี้อาจมีได้หลาย template เนื่องจากว่าภาพถ่ายของจุดสังเกต (landmark) ที่มุมต่างกันย่อมต้องมีลักษณะต่างกันอีกทั้งภาพถ่ายจุดสังเกต (landmark) ที่ระยะแตกต่างกันย่อมต้องมีความละเอียดของภาพที่ต่างกันด้วย ซึ่งถ้าหากมีการเลือกใช้ template ที่มีความต่างของลักษณะ landmark แตกต่างกันเกินไปอาจทำให้การปรับแก้ตำแหน่ง feature ด้วย template matching ล้มเหลวได้

สำหรับวิธีการในการเลือก template นั้นเพียงแค่เลือก template ที่มุมถ่ายภาพใกล้เคียงกับมุมกล้องปัจจุบันและมีระยะห่างจากจุดสังเกต (landmark) ไม่ต่างกันมากนักก็ใช้ได้ ซึ่งตำแหน่งกล้องปัจจุบันนั้นสามารถประมาณได้จาก SLAM และถ้าหากว่าไม่มี template ที่เหมาะสมก็จะเลือกให้ภาพจากกล้องปัจจุบันเป็น template ตำแหน่งของกล้องที่ถ่ายภาพ template นั้นจะถูกเรียกว่า reference frame ซึ่ง reference frame นี้จะถูกใช้ในอีกหลาย ๆ เป้าหมายซึ่งจะได้อธิบายต่อไป

2. การประมาณ patch อ้างอิงที่จะใช้ในการค้นหา

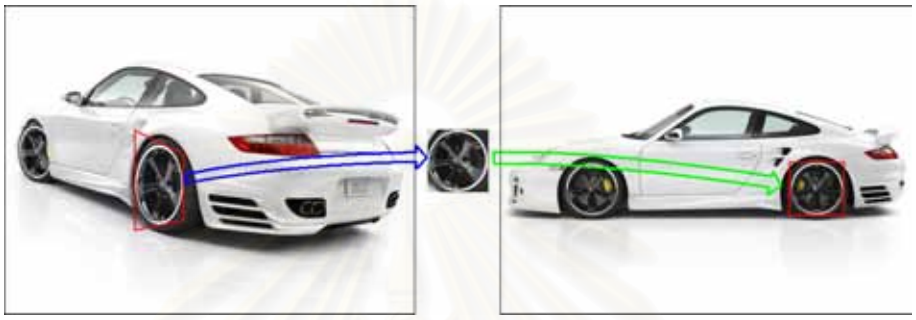
วิธีการของ Template matching นั้นจะเป็นการนำเอา template ที่มีลักษณะเป็น patch ดังแสดงในรูป 5.7ก มาทำการเปรียบเทียบทั่วบริเวณที่ต้องการทำการค้นหาโดยการเลื่อน patch ไปทั่วบริเวณแบบครบทุกพิกเซล และหาผลต่างระหว่าง patch กับภาพที่ต้องการค้นหา ในแต่ละพิกเซลนั้น ๆ สำหรับตำแหน่งไหนที่มีค่าความต่างน้อยสุดแสดงว่าจุดนั้นจะสอดคล้องกับ patch มากที่สุดดังแสดงในรูป 5.7ค ซึ่งจะเห็นได้ว่าบริเวณล้อหน้าและล้อหลังของรถ ค่าความต่างที่เป็นผลลัพธ์จาก Template matching นั้นมีค่าน้อยมาก (แสดงได้ด้วยสีดำ)



(ก) (ข) (ค)

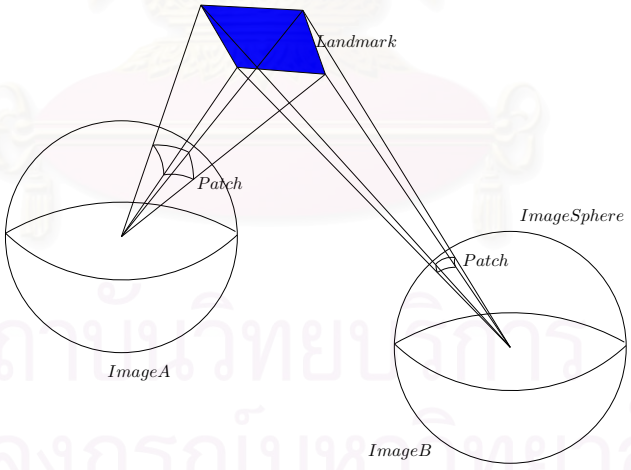
รูปที่ 5.7: วิธีการ Template matching (ก) ภาพ template (ข) ภาพต้นฉบับ (ค) ผลลัพธ์จาก Template matching

แต่เนื่องจากว่าวิธีการของ template matching นั้นไม่ Invariant ต่อ Scale และ Rotation ซึ่งหมายความว่า patch ที่จะนำมาใช้ในการค้นหาได้นั้น วัตถุที่ปรากฏใน patch จะต้องมีขนาดและทิศการวางตัว ตรงกับวัตถุในภาพที่ต้องการค้นหา แต่ในทางปฏิบัติแล้วมีความเป็นไปได้น้อยมากที่ patch และภาพที่ต้องการค้นหาจะมี Scale และ Rotation ตรงกัน ทั้งนี้เป็นเพราะว่าการถ่ายภาพที่ตำแหน่งแตกต่างกัน projection ของวัตถุบนภาพในแต่ละภาพย่อมต้องมีลักษณะต่างกันด้วย ดังแสดงในรูปที่ 5.8 จะเห็นได้ว่าโมเดลรถ ในภาพสองภาพจะมีขนาดและทิศการวางตัวที่แตกต่างกัน ดังนั้นจะไม่สามารถนำ patch ล้อรถในภาพที่ 1 ไปวัดเทียบกับภาพที่ 2 ได้โดยตรงเพราะล้อรถไม่สามารถซ้อนทับกันได้พอดี ดังนั้นในขั้นตอนนี้จะเป็นการประมาณ Scale และ Rotation ที่ถูกต้องของ patch แต่ละ feature เพื่อนำไปใช้กับ Template matching ต่อไป



รูปที่ 5.8: แสดงการแปลง patch จากภาพ template(ภาพซ้าย) ให้สอดคล้องกับภาพปัจจุบัน(ภาพขวา)

ผลลัพธ์ของการทำงานส่วนนี้คือการประมาณ patch จากภาพ template ที่มี Scale และ Rotation สอดคล้องกับภาพปัจจุบันเพื่อใช้ใน Template matching ดังนั้นจึงจะต้องหา Transform function ซึ่งในการแปลงดังแสดงในรูปที่ 5.8



รูปที่ 5.9: ภาพแสดงการ project ตำแหน่งของ patch ในสามมิติลงบนภาพจากกล้องออมนิเพื่อหาความสอดคล้องของ Patch ระหว่างภาพ A และภาพ B

แนวคิดของการหา patch จากภาพ template ที่สอดคล้องกับภาพปัจจุบันแสดงได้ดังรูปที่ 5.9 โดยรูปทรงกลมนั้นแสดงถึงฉากรับภาพของกล้องวิดีโอแบบออมนิ และทรงกลม A จะแทนภาพที่เวลาปัจจุบัน ส่วนทรงกลม B จะแทนภาพ template แนวคิดของการหา patch จากภาพ template นั้นจะเริ่มจากการฉายภาพ patch ของ feature จากภาพปัจจุบันให้อยู่ในพิกัดสามมิติ โดยอาศัยข้อมูลระยะทางจากตำแหน่งจุดสังเกตที่ประมาณได้จาก SLAM โดยภาพ feature นั้นจะมีทิศทางหันเข้ากล้องดังแสดงได้ดังรูปที่ 5.9 จากนั้นจึงโปรเจค patch ของ feature ในสามมิติกลับลงมาที่ภาพ template ทำให้ทราบได้ว่าตำแหน่งจุดใดใน patch ภาพ template จะเป็น patch ที่สอดคล้องกับภาพปัจจุบัน แต่การโปรเจค patch ของ feature

ในสามมิติกลับลงมาที่ภาพ templateนั้นจุดศูนย์กลางตำแหน่งโปรเจค อาจไม่ตรงกับตำแหน่ง feature ของภาพ template ก็เป็นได้ ดังนั้นวิธีแก้ก็เพียงแค่เลื่อนจุดโปรเจคทั้งหมดให้มีจุดศูนย์กลางที่ตรงกับตำแหน่ง feature ของภาพ template

การโปรเจคจุดทุกจุดใน patch ผลลัพธ์ที่ต้องการลงบนภาพ template เพื่อการหาฟังก์ชันการแปลง patch จากภาพ templateให้สอดคล้องกับภาพปัจจุบันอาจจะต้องใช้เวลามากในการคำนวนดังนั้น วิธีแก้ไขจะการโปรเจคจุดลงบนภาพ template เพียง 4 จุดแล้วจึงใช้ Perspective function ในการประมาณ Transform function โดยมีสมมุติฐานว่า patch ที่ใช้ในการค้นหานั้นมีขนาดเล็กมากเมื่อเทียบกับภาพจากกล้องออมนิทั้งภาพ ซึ่ง Perspective function สามารถเขียนได้ดังนี้

$$I_{patch}(x) = I_{template}(f(x)) \tag{5.9}$$

โดย $f(x)$ เป็น Perspective function

$$f(x) = \begin{bmatrix} u_x/u_z \\ u_y/u_z \end{bmatrix} \tag{5.10}$$

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = M \cdot \begin{bmatrix} x_x \\ x_y \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \cdot \begin{bmatrix} x_x \\ x_y \\ 1 \end{bmatrix} \tag{5.11}$$

เมื่อ M เป็น Perspective Matrix มีพารามิเตอร์ 8 ตัวคือ (a, b, c, d, e, f, g, h) และ $x = \begin{bmatrix} x_x & x_y \end{bmatrix}^T$ คือตำแหน่งพิกเซลบน patch ซึ่งสำหรับการหาพารามิเตอร์ของ Perspective function นั้นจะหาได้จากการแก้สมการแบบ linear ปรกติ ก็คือเมื่อกำหนดให้$f(x) = \begin{bmatrix} p_x & p_y \end{bmatrix}^T$ เป็นตำแหน่งบนพิกเซลภาพ template จะได้ว่า

$$\begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} = \begin{bmatrix} u_x/u_z \\ u_y/u_z \\ u_z/u_z \end{bmatrix} \tag{5.12}$$

$$\begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} = \frac{\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \cdot \begin{bmatrix} x_x \\ x_y \\ 1 \end{bmatrix}}{\begin{bmatrix} g & h & 1 \end{bmatrix} \cdot \begin{bmatrix} x_x \\ x_y \\ 1 \end{bmatrix}} \tag{5.13}$$

เมื่อแตกรูปสมการ Matrix จะได้ว่า

$$p_x = \frac{ax_x + bx_y + c}{gx_x + hx_y + 1} \tag{5.14}$$

$$p_y = \frac{dx_x + ex_y + f}{gx_x + hx_y + 1} \tag{5.15}$$

ซึ่งเมื่อจัดรูปสองสมการข้างต้นใหม่แล้วจะได้ว่า

$$p_x = ax_x + bx_y + c - gx_xp_x - hx_yp_x \quad (5.16)$$

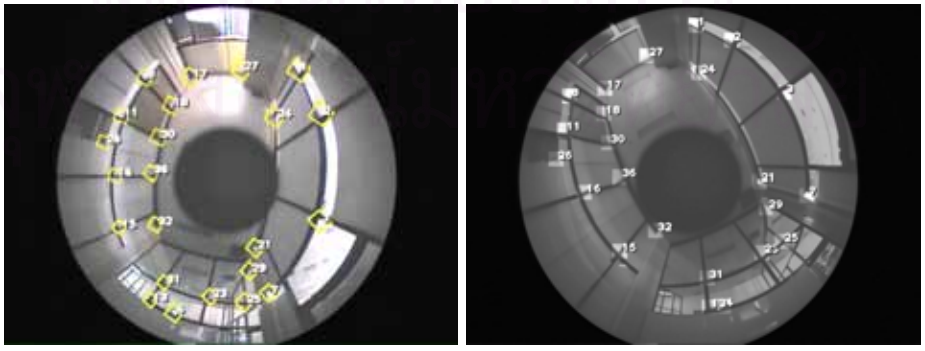
$$p_y = dx_x + ex_y + f - gx_xp_y - hx_yp_y \quad (5.17)$$

สำหรับกรณีที่มีคู่จุด $\begin{bmatrix} p_x & p_y \end{bmatrix}^T$ และ $\begin{bmatrix} x_x & x_y \end{bmatrix}^T$ หลายคู่จุดจะสามารถจัดให้อยู่ในรูปการคูณของ Matrix ได้ดังนี้

$$\begin{bmatrix} x_x & x_y & 1 & 0 & 0 & 0 & -x_xp_x & -x_yp_x \\ 0 & 0 & 0 & x_x & x_y & 1 & -x_xp_y & -x_yp_y \\ & & & & \vdots & & & \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ \vdots \end{bmatrix} \quad (5.18)$$

จากสมการข้างต้นสามารถเขียนได้ในรูป $Ay = B$ ซึ่งสำหรับการแก้สมการหาค่า y จะต้องใช้คู่จุด $\begin{bmatrix} p_x & p_y \end{bmatrix}^T$ และ $\begin{bmatrix} x_x & x_y \end{bmatrix}^T$ อย่างน้อย 4 คู่ซึ่งถ้าหากมีคู่จุด 4 คู่พอก็สามารถแก้สมการหาค่า y ได้โดยตรงจาก $y = A^{-1}B$ แต่ถ้าหากคู่จุดมีมากกว่า 4 คู่จะต้องแก้สมการโดยใช้ Least Squares โดยวิธีการแก้ Least Squares ที่ง่ายที่สุดแต่ไม่เสถียรทางเชิงตัวเลข (Numerically stable) คือการใช้ pseudo-inverse $y = (A^TA)^{-1}A^TB$

ในงานวิจัยนี้จะใช้คู่จุด 4 คู่ในการหา Perspective function สำหรับแปลง patch จากภาพ template ให้สอดคล้องกับภาพที่ต้องการค้นหา ซึ่งจุด $\begin{bmatrix} x_x & x_y \end{bmatrix}^T$ นั้นทราบค่าอยู่แล้วโดยกำหนดให้เป็นจุด 4 มุมของ patch ตามขนาดที่ต้องการใช้ ส่วนจุด $\begin{bmatrix} p_x & p_y \end{bmatrix}^T$ ซึ่งเป็นตำแหน่งบนภาพจาก template นั้นจะหาได้จากการ project ตำแหน่งของ patch ในสามมิติลงบนภาพ template โดยตำแหน่งของ patch สามมิตินั้นพอจะประมาณได้จากตำแหน่งประมาณของจุดสังเกต (landmark) ที่หาได้จาก SLAM สำหรับตัวอย่างผลลัพธ์การหา Perspective function แสดงได้ในรูปที่ 5.10



(ก) (ข)

รูปที่ 5.10: ผลลัพธ์การหา Perspective function (ก) ภาพ template และ patch (ข) ภาพปัจจุบันและ patch หลังจากการแปลง Perspective transform แล้ว

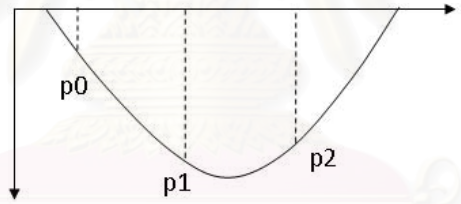
3. การค้นหาตำแหน่งที่สอดคล้องกับ patch ด้วย Pyramids templatematching

หลังจากที่หา patch ได้จากการคำนวนข้างต้นแล้วก็จะใช้ patch นั้นมาทำการค้นหาในภาพปัจจุบันในบริเวณรอบ ๆ จุดที่ได้จากการติดตามการเลื่อนตำแหน่งของ feature เพื่อเป็นการปรับแก้ค่าให้ feature สอดคล้องกับภาพ template ตลอดเวลา ซึ่งในการค้นหานั้นจะใช้การค้นหาแบบ Pyramids ซึ่งก็คือการหา template matching บนภาพหยาบก่อนแล้วจึงค่อยหา template matching บนภาพที่ละเอียดขึ้นไปเรื่อย ๆ สำหรับข้อดีของการหา template matching แบบ Pyramids ก็คือสามารถค้นหาได้เป็นบริเวณกว้างกว่าโดยไม่ต้องค้นหาแบบละเอียดทั้งภาพ ซึ่งในงานวิจัยนี้จะใช้ Pyramids เพียง 2 ระดับ เพราะถ้าหากจำนวนระดับมากกว่านี้จะทำให้ patch มีขนาดใหญ่เกินกว่าจะประมาณด้วย Perspective distortion ได้

4. การประมาณตำแหน่งระดับ sub-pixel

ในการปรับแก้ตำแหน่งของ Feature ด้วย template matching นั้นจะทำการเลื่อน patch ไปทีละพิกเซล และหาความต่างของภาพกับ patch ที่บริเวณนั้น จากนั้นจึงเลือกตำแหน่งที่มีผลต่างต่ำสุด เป็นตำแหน่งปรับแก้ของ feature จึงทำให้ตำแหน่งปรับแก้มีค่าเป็น discrete เท่านั้น สำหรับในขั้นตอนนี้จะเป็นการประมาณตำแหน่งปรับแก้ของ Feature ให้มีความละเอียดยิ่งขึ้นซึ่งในการประมาณนั้นเพื่อความสะดวกจะประมาณตำแหน่งในแกน x และแกน y แยกจากกัน

จากในรูปที่ 5.11 จุด p1 เป็นจุดต่ำสุดที่หาได้จาก template matching ส่วนจุด p0 และ p2 เป็นจุดข้างเคียงจุด p1 ซึ่งจากในรูปจะเห็นได้ว่าจุด p1 ไม่ใช่จุดต่ำสุดที่แท้จริง ซึ่งการประมาณจุดต่ำสุดที่แท้จริงในที่นี้จะใช้จุดสามจุดในการประมาณสมการพาราโบล่า จากนั้นจึงเลือกจุดต่ำสุดในสมการพาราโบล่าเพื่อเป็นตำแหน่งปรับแก้ของ Feature



รูปที่ 5.11: การประมาณตำแหน่งระดับ sub-pixel

5. การประเมินผลลัพธ์ของการปรับแก้ตำแหน่งของ Feature

ในการปรับแก้ตำแหน่งของ Feature นั้นอาจเกิดความล้มเหลวได้ในกรณีที่จุดสังเกต (landmark) ของ feature นั้น ๆ ได้ถูกบดบังโดยสิ่งแวดล้อมอื่น ๆ หรือได้หลุดไปจาก field of view ของกล้องไปแล้ว หมายความว่ากล้องไม่สามารถตรวจวัดจุดสังเกต (landmark) นั้นได้อีก ซึ่งวิธีการปรับแก้ความถูกต้องของตำแหน่ง feature นั้นนอกจากจะใช้ปรับแก้ตำแหน่งของ feature แล้วยังใช้ตรวจสอบได้อีกด้วยว่า feature นั้น ๆ ยังปรากฏอยู่บนภาพหรือไม่

สำหรับวิธีการในการตรวจสอบนั้นก็มีได้ 2 วิธีคือ วิธีแรกจะตรวจสอบค่าความต่างน้อยสุดระหว่าง feature ปัจจุบันกับ patch ของ feature template ซึ่งถ้าหากความแตกต่างมีค่าเกินเกณฑ์ ก็จะถือว่า landmark นั้นไม่ปรากฏบนภาพอีกต่อไป ส่วนวิธีที่สอง ก็คือการตรวจสอบว่าตำแหน่งของ feature หลังการปรับแก้นั้นเป็นตำแหน่งขอบของบริเวณค้นหาหรือไม่ ถ้าหากว่าตำแหน่งของ feature อยู่ตรงขอบการค้นหาแสดงว่าการค้นหาตำแหน่งล้มเหลว เพราะตำแหน่งของ feature ตำแหน่งนั้นอาจไม่ได้มีค่าความต่างน้อยสุดจริง ๆ ซึ่ง feature ที่ปรับแก้ตำแหน่งล้มเหลวเหล่านี้จะไม่ถูกพิจารณาเป็นข้อมูล Measurement

5.1.4 สรุปการวิเคราะห์ข้อมูลภาพจากกล้องวิดีโอแบบออมนิ

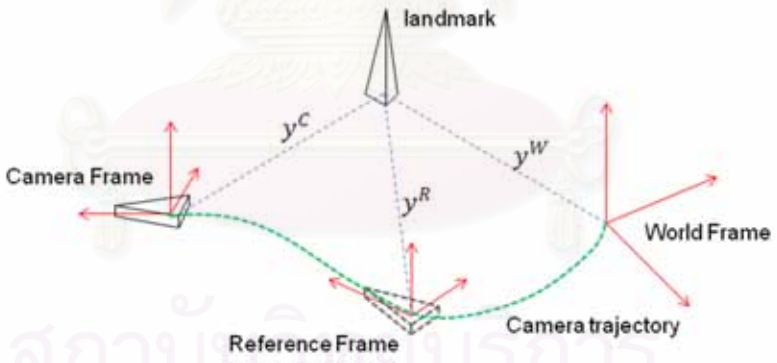
ผลลัพธ์ที่ได้จากการวิเคราะห์ข้อมูลภาพจากกล้องวิดีโอแบบออมนิ จะเป็นข้อมูลการวัดจุดสังเกต (landmark) ซึ่งแสดงเป็นค่าการวัดเชิงองศา โดยจะสามารถระบุได้ว่าข้อมูลการวัดจุดสังเกตเหล่านั้นสอดคล้องกับตำแหน่งจุดสังเกตที่ปรากฏบนแผนที่ของ SLAM ตำแหน่งไหนบ้าง หรือว่าข้อมูลการวัดจุดสังเกตนั้นเป็นข้อมูลจุดสังเกตจุดใหม่ ซึ่งยังไม่ปรากฏในแผนที่ โดยข้อมูลทั้งหลายเหล่านี้จะถูกนำไปใช้ในงานระบุตำแหน่งพร้อมกับสร้างแผนที่ (SLAM) ต่อไป

5.2 การระบุตำแหน่งพร้อมกับการสร้างแผนที่ในสามมิติ

การระบุตำแหน่งพร้อมกับการสร้างแผนที่ในสามมิติ เป็นการประมาณตำแหน่งของกล้องและสร้างแผนที่ของสิ่งแวดล้อมในรูปตำแหน่งของจุดสังเกต (Landmark) โดยใช้ข้อมูลการวัดของกล้องวิดีโอแบบออมนิที่ได้จากการวิเคราะห์ข้อมูลภาพ ในการประมาณค่าด้วย SLAM ในงานวิจัยนี้จะกำหนด กรอบอ้างอิงของระบบดังต่อไปนี้

5.2.1 กรอบอ้างอิงของระบบ

กรอบอ้างอิงที่จะใช้ในงานวิจัยนี้จะมีอยู่ 3 ประเภท คือ กรอบอ้างอิงหลัก (World Frame) แทนด้วย W, กรอบอ้างอิงย่อย (Reference Frame) แทนด้วย R และ กรอบอ้างอิงของกล้อง (Camera Frame) แทนด้วย C ซึ่งจากรูปที่ 5.12 World Frame จะใช้เป็น global Coordinate ส่วน Camera Frame จะใช้อ้างอิงตำแหน่งต่าง ๆ สัมพัทธ์กับกล้องซึ่งจะเปลี่ยนแปลงไปเมื่อกล้องมีการเคลื่อนที่ สำหรับ Reference Frame นั้นจะใช้เป็นจุดอ้างอิงพิเศษ ซึ่งในที่นี้จะเป็นกรอบอ้างอิงในอดีต ของกล้อง พูดอีกนัยหนึ่งก็คือ Reference Frame จะกระจายอยู่ตามเส้นทางที่กล้องมีการเคลื่อนที่ผ่าน



รูปที่ 5.12: กรอบอ้างอิงที่ใช้ในงานวิจัย

สำหรับในงานวิจัยนี้ Reference Frame จะเป็นกรอบอ้างอิงพิเศษที่ถูกกำหนดขึ้นมาเพื่อความสะดวกในการทำงานหลาย ๆ อย่างเช่น ใช้เก็บภาพถ่ายจากกล้องออมนิ ณ ตำแหน่งนั้นเพื่อใช้เป็น template ในการปรับแก้ตำแหน่ง feature ที่ได้กล่าวถึงไปในหัวข้อที่แล้ว, ใช้ในการประมาณการเคลื่อนที่ของกล้อง หรือ ใช้ในการช่วยปรับแก้ตำแหน่งจุดสังเกต ซึ่งจะได้กล่าวถึงต่อไป

5.2.2 สถานะของ SLAM

ในงานวิจัยนี้จะใช้ Extended Kalman filter (EKF) ในการแก้ปัญหา ดังนั้นจึงจะต้องกำหนดสถานะของ Kalman ที่ต้องการประมาณ ซึ่งแต่เดิมนั้น SLAM แบบทั่วไปสถานะของระบบจะประกอบด้วยตำแหน่งของกล้อง และตำแหน่งของจุดสังเกต แต่ในงานวิจัยนี้เมื่อมี Reference Frame เข้ามาเพิ่มจึงต้องเพิ่มสถานะของ Reference Frame เข้าไปในสถานะของระบบด้วย จึงได้ว่าการกระจายความน่าจะเป็นของ

สถานะของ kalman ณ เวลาหนึ่ง ๆ ประกอบด้วย เวกเตอร์สถานะ (state vector) ($\hat{x}$) และ ค่าความแปรปรวนร่วมของสถานะ (covariance matrix) (P) ซึ่งเวกเตอร์สถานะ จะมีข้อมูลสามส่วนคือ สถานะของกล้อง (camera state) ($\hat{x}_c$), สถานะของกรอบอ้างอิง (reference frame states) ($\hat{x}_r$) และ สถานะของจุดสังเกต (landmark states) ($\hat{x}_y$)

$$\hat{x} = \begin{bmatrix} \hat{x}_c \\ \hat{x}_r \\ \hat{x}_y \end{bmatrix} \quad P = \begin{bmatrix} P_{x_c x_c} & P_{x_c x_r} & P_{x_c x_y} \\ P_{x_r x_c} & P_{x_r x_r} & P_{x_r x_y} \\ P_{x_y x_c} & P_{x_y x_r} & P_{x_y x_y} \end{bmatrix} \tag{5.19}$$

$$\hat{x}_c = \begin{bmatrix} t^{WC} & q^{WC} \end{bmatrix}^T \tag{5.20}$$

$$\hat{x}_r = \begin{bmatrix} \hat{r}_1 & \hat{r}_2 & \cdots \end{bmatrix}^T \quad \hat{r}_i = \begin{bmatrix} t_i^{WR} & q_i^{WR} \end{bmatrix}^T \tag{5.21}$$

$$\hat{x}_y = \begin{bmatrix} \hat{y}_1 & \hat{y}_2 & \cdots \end{bmatrix}^T \tag{5.22}$$

สำหรับสถานะของกล้อง ($\hat{x}_c$) นั้นจะประกอบไปด้วย ตำแหน่งของกล้องสามมิติใน World Frame (t^{WC}) และ ทิศทางของกล้องสามมิติใน World Frame (q^{WC}) ซึ่งบรรยายการหมุนด้วย Z-Y-X Euler angles สถานะของจุดสังเกต ($\hat{x}_y$) จะประกอบไปด้วยพารามิเตอร์ของจุดสังเกต ($\hat{y}_j$) ในสามมิติ ซึ่งใช้ในการอธิบายสิ่งแวดล้อมทั้งหมดที่กล้องตรวจวัดได้ ในที่นี้จะเป็นตำแหน่งของจุดสังเกตในสามมิติ ส่วนสถานะของกรอบอ้างอิง ($\hat{x}_r$) ในที่นี้จะกำหนดให้เป็นสถานะของกล้องในอดีต ซึ่งสถานะของกรอบอ้างอิงจะประกอบด้วย กรอบอ้างอิง ($\hat{r}_i$) หลาย ๆ กรอบ ซึ่งแต่ละกรอบอ้างอิงจะประกอบไปด้วย ตำแหน่งของกรอบอ้างอิงเทียบกับ World Frame (t_i^{WR}) และ ทิศทางของกล้องเทียบกับ World Frame (q_i^{WR})

ในการประมาณค่าของ SLAM นั้นจะมีการใช้ Transformation Matrix เป็นจำนวนมากซึ่ง Transformation Matrix (M) จะหาได้จาก

$$M = \begin{bmatrix} cb \cdot cg & sa \cdot sb \cdot cg - ca \cdot sg & ca \cdot sb \cdot cg + sa \cdot sg & t_x \\ cb \cdot sg & sa \cdot sb \cdot sg + ca \cdot cg & ca \cdot sb \cdot sg - sa \cdot cg & t_y \\ -sb & sa \cdot cb & ca \cdot cb & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.23}$$

$$ca = cos(\alpha), \quad sa = sin(\alpha) \tag{5.24}$$

$$cb = cos(\beta), \quad sb = sin(\beta) \tag{5.25}$$

$$cg = cos(\gamma), \quad sg = sin(\gamma) \tag{5.26}$$

โดยที่ $q = \begin{bmatrix} \alpha & \beta & \gamma \end{bmatrix}^T$ เป็นเวกเตอร์แสดงทิศทาง และ $t = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T$ เป็นเวกเตอร์ตำแหน่งในสามมิติ

เวกเตอร์สถานะทั้งหลายเหล่านี้จะถูกประมาณด้วย Extended Kalman filter ซึ่งจะทำงานในสองกระบวนการวนซ้ำกันไปเรื่อย ๆ คือ prediction (time-update) และ correction (measurement-update) โดยอาศัยข้อมูลการวัดกล้องวิดีโอแบบออมนิในการประมาณค่า

5.2.3 การทำนายการเคลื่อนที่ของกล้อง (Prediction)

สำหรับงานระบบตำแหน่งของกล้องพร้อมกับการสร้างแผนที่ในที่นี้ มีข้อสมมุติอยู่ว่าสิ่งแวดล้อมที่ใช้ในการระบุตำแหน่งแหน่งและสร้างแผนที่ จะไม่มีการเปลี่ยนแปลงซึ่งหมายความว่าในระบบนี้ เมื่อเวลาผ่านไปจะมีเพียงตำแหน่งของกล้องเท่านั้นที่มีการเปลี่ยนแปลง ดังนั้นจึงต้องมีการประมาณการกระจายความน่าจะเป็นของตำแหน่งกล้องที่เวลาปัจจุบัน หรืออีกนัยหนึ่งคือการทำนายการเคลื่อนที่ของกล้อง

ในงานวิจัยนี้เนื่องจากว่ากล้องสามารถเคลื่อนที่ได้อย่างอิสระ โดยที่ไม่สามารถรู้โมเดลการเคลื่อนที่ของกล้อง และคำสั่งควบคุมการเคลื่อนที่ได้ ดังนั้นจึงยากที่จะประมาณการเคลื่อนที่ของกล้องด้วยโมเดลใดโมเดลหนึ่ง ดังนั้นจึงจะใช้วิธีการประมาณการกระจายความน่าจะเป็นใหม่ของตำแหน่งกล้อง โดยกำหนดให้ตำแหน่งของกล้องอยู่ที่ตำแหน่งเดิม และให้ค่าความแปรปรวนของตำแหน่งมีค่ามากเกินพอ ซึ่งหมายความว่ากล้องอาจจะเคลื่อนที่ไปในทิศใด ๆ ก็ได้ จะได้ว่า

$$\hat{x}^{-}_{c,k} = f(\hat{x}_{c,k-1}) = \begin{bmatrix} t^{WC}_{k-1} & q^{WC}_{k-1} \end{bmatrix}^T \tag{5.27}$$

$$P^{-}_{c,k} = P_{c,k-1} + Q_k \tag{5.28}$$

เมื่อ $\hat{x}^{-}_{c,k}$ เป็นสถานะของกล้องที่ทำนายได้ที่เวลาปัจจุบัน ซึ่งจะเท่ากับสถานะก่อนหน้า และ $P^{-}_{c,k}$ เป็นความแปรปรวนของสถานะของกล้องหลังการทำนาย ซึ่งหาได้จากความแปรปรวนเมื่อเวลาก่อนหน้าบวกด้วยค่าความแปรปรวนของระบบ Q_k จำนวนมากเกินพอ ซึ่งความไม่แน่นอนของสถานะของกล้องเหล่านี้จะถูกปรับแก้ให้มีความแม่นยำในภายหลัง

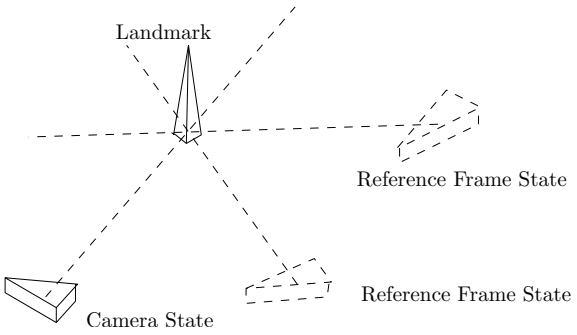
5.2.4 การปรับแก้ตำแหน่งของกล้องและจุดสังเกต (Correction)

การปรับแก้ตำแหน่งของกล้องและจุดสังเกตจะเป็นการนำเอาข้อมูลการวัดที่ได้จากกล้องวิดีโอแบบออมนิมาใช้ประมาณตำแหน่งของกล้องและตำแหน่งจุดสังเกต ซึ่งขั้นตอนในการปรับแก้จะแยกเป็นสองขั้นตอนหลัก ๆ ด้วยกันคือการปรับแก้ตำแหน่งของกล้อง และการปรับแก้ตำแหน่งจุดสังเกต ซึ่งทั้งสองขั้นตอนนี้ใช้ข้อมูลการวัดและโมเดลการวัดอย่างเดียวกัน

ในการปรับแก้ค่าสถานะของกล้องและจุดสังเกต จะใช้ข้อมูลการวัดจากกล้องออมนิ ณ เวลาปัจจุบันร่วมกับข้อมูลการวัดจากกล้องออมนิ ในอดีตที่ตำแหน่ง Reference frame โดยจะใช้ Reference frame ร่วมด้วยอย่างน้อย 1 frame การใช้ข้อมูลการวัดจากกล้องออมนิในอดีต จะมีส่วนช่วยลดความผิดพลาดที่เกิดจากการวัดที่ล้มเหลวของการวิเคราะห์ภาพจากกล้องออมนิ ซึ่งเป็นเหตุการณ์ที่เกิดขึ้นได้ปรกติอันเนื่องมาจากความไม่สมบรูณ์แบบของอัลกอริทึมทาง Computer Vision แต่ถ้าหากมีการใช้ข้อมูลการวัดจากกล้องออมนิในอดีตทุก Reference frame ที่มีความสัมพันธ์กับจุดสังเกต อาจจะส่งผลกระทบถึงประสิทธิภาพด้านการคำนวนได้ เนื่องจากว่าข้อมูลที่ใช้ประมวลผลมีมากเกินไป ดังนั้นจึงจะเลือกเฉพาะข้อมูลจาก Reference frame ที่มีความสอดคล้องกับตำแหน่งกล้องปัจจุบันสูงสุดเพียงสองถึงสาม frame เท่านั้นโดยวิธีการเลือกนั้นจะเลือก Reference frame ที่มีข้อมูลการวัดจุดสังเกตในอดีตตั้งฉากกับข้อมูลการวัดที่เวลาปัจจุบันมากที่สุด ดังแสดงได้ในรูป 5.13 เพื่อให้สามารถระบุตำแหน่งในสามมิติได้แม่นยำที่สุด

ข้อมูลการวัดจุดสังเกตจุดที่ i ของกล้องออมนิสามารถเขียนได้เป็น

$$\phi_i = \begin{bmatrix} z^C_i & z^{u(1)}_i & z^{u(2)}_i & \cdots \end{bmatrix}^T \tag{5.29}$$



รูปที่ 5.13: การประมาณด้วย Measurement หลายค่า

โดยฟังกชัน u สำหรับบรรยาย Reference frame ที่เป็นตำแหน่งของข้อมูลการวัดในอดีต ซึ่งโมเดลการวัดจุดสังเกตจุดที่ i สามารถเขียนได้เป็น

$$h_i(\hat{x}_c, \hat{x}_r, \hat{y}_i) = \begin{bmatrix} h_i^{'}(\hat{x}_c, \hat{y}_i) & h_i^{'}(\hat{r}_{u(1)}, \hat{y}_i) & h_i^{'}(\hat{r}_{u(2)}, \hat{y}_i) & \cdots \end{bmatrix}^T \tag{5.30}$$

โดยที่

$$h_i^{'}(x, y) = \begin{bmatrix} \arctan(y_z^{'}/\sqrt{y_x^{'} \cdot y_x^{'} + y_y^{'} \cdot y_y^{'}}) \\ \arctan(y_y^{'}/y_x^{'}) \end{bmatrix} \tag{5.31}$$

$$y^{'} = \begin{bmatrix} y_x^{'} & y_y^{'} & y_z^{'} \end{bmatrix}^T = T_x(y) \tag{5.32}$$

โดยที่ $T_x(y)$ เป็น Transformation function ที่แปลงตำแหน่งจุดสังเกต y จาก world coordinate มายังกรอบอ้างอิง x ซึ่งหาได้จาก $T_x(y) = M^{-1}(x) \times y$ เมื่อ $M(x)$ เป็น Transformation Matrix ของ x

5.2.4.1 การปรับแก้ตำแหน่งของกล้อง

การปรับแก้ตำแหน่งของกล้องจะใช้โมเดลการวัดของกล้องข้างต้นมาประมาณการกระจายความน่าจะเป็นของตำแหน่งกล้องใหม่โดยแต่เดิมนั้นกระจายความน่าจะเป็นของตำแหน่งกล้องหลังจากผ่านกระบวนการทำนายการเคลื่อนที่ (Prediction) ทำให้การกระจายมีค่าสูงมาก ซึ่งสำหรับการปรับแก้ตำแหน่งของกล้องนั้นก็มีกระบวนการเหมือน การ correction ของ Extended Kalman filter ทั่วไปคือ

หาการวัดส่วนต่าง (Measurement residual)	$\hat{v}_i = \phi_i - h_i(\hat{x}_c, \hat{x}_r, \hat{y}_i)$
หาความแปรปรวนส่วนต่าง (Residual covariance)	$S = H_c P^- H_c^T + H_y P^- H_y^T + R$
หา Kalman gain	$K = P^- H_c^T S^{-1}$
หาสถานะหลังการประมาณ	$\hat{x} = \hat{x}^- + K\hat{v}$
หาความแปรปรวนสถานะหลังการประมาณ	$P = (I - KH_c)P^-$

โดยที่ H_c เป็น Jacobian Matrix ของฟังก์ชัน h เทียบกับ $\hat{x}_c$
H_y เป็น Jacobian Matrix ของฟังก์ชัน h เทียบกับ $\hat{x}_y$

โดยในการประมาณตำแหน่งกล้องนี้จะสมมุติว่าตำแหน่งของจุดสังเกตมีความแม่นยำ ทำให้ค่าการวัดส่งผลโดยตรงต่อตำแหน่งของกล้องเท่านั้น ดังนั้นในการคำนวนค่าความแปรปรวนส่วนต่าง (Residual

covariance) จึงสามารถแยก Jacobian Matrix ของฟังก์ชันการวัดเทียบกับสถานะของกล้อง ($\hat{x}_c$) และ Jacobian Matrix เทียบกับสถานะของจุดสังเกต ($\hat{x}_y$) ออกจากกันได้ ในการคำนวนหา Kalman gain และการปรับแก้จะใช้เพียง Jacobian Matrix เทียบกับสถานะของกล้อง (H_c) เท่านั้น และเนื่องจากการกระจายตำแหน่งของกล้องก่อนการปรับแก้มีการกระจายมากเกินพอ ผลลัพธ์ที่ได้ก็คือตำแหน่งของกล้องจะถูกประมาณใหม่โดยความแปรปรวนของตำแหน่งของกล้องจะมีค่าน้อยสุด และตำแหน่งของจุดสังเกตจะถูกเปลี่ยนแปลงตำแหน่งน้อยมากจนถือได้ว่าไม่มีการเปลี่ยนแปลง

สำหรับการปรับแก้ตำแหน่งของกล้องที่มีความแม่นยำมากขึ้นสามารถทำได้โดยการประมาณตำแหน่งของกล้องด้วยข้อมูลการวัดทั้งหมด จากนั้นจึงคัดข้อมูลการวัดที่มีความคลาดเคลื่อนมากเกินไปออก โดยการพิจารณาค่าการวัดส่วนต่าง (Measurement residual) ($\hat{v}_i$) จากนั้นจึงย้อนกลับไปทำกระบวนการการปรับแก้ตำแหน่งของกล้องใหม่โดยใช้เฉพาะข้อมูลการวัดที่ถูกคัดเลือกแล้วเท่านั้น ก็จะทำให้ประมาณตำแหน่งของกล้องได้แม่นยำมากขึ้น

5.2.4.2 การปรับแก้ตำแหน่งของจุดสังเกต

การปรับแก้ตำแหน่งของจุดสังเกตนั้นก็มีกระบวนการดังนี้คือ

หาการวัดส่วนต่าง (Measurement residual)	$\hat{v}_i = \phi_i - h_i(\hat{x}_c, \hat{x}_r, \hat{y}_i)$
หาความแปรปรวนส่วนต่าง (Residual covariance)	$S = H_c P^- H_c^T + H_y P^- H_y^T + R$
หา Kalman gain	$K = P^- H_y^T S^{-1}$
หาสถานะหลังการประมาณ	$\hat{x} = \hat{x}^- + K\hat{v}$
หาความแปรปรวนสถานะหลังการประมาณ	$P = (I - KH_y)P^-$

โดยที่ H_c เป็น Jacobian Matrix ของฟังก์ชัน h เทียบกับ $\hat{x}_c$
H_y เป็น Jacobian Matrix ของฟังก์ชัน h เทียบกับ $\hat{x}_y$

จะเห็นได้ว่าสมการสำหรับปรับแก้ค่าข้างต้นมีความแตกต่างจากการปรับแก้ตำแหน่งของกล้องเพียงเล็กน้อยเท่านั้นคือในการหา kalman gain และการหาความแปรปรวนสถานะหลังการประมาณ จะใช้ Jacobian Matrix เทียบกับสถานะของกล้อง (H_v) แทน Jacobian Matrix เทียบกับสถานะของกล้อง (H_c) ซึ่งหมายความว่าในการการปรับแก้ตำแหน่งของจุดสังเกตนั้นจะสมมุติว่าตำแหน่งของกล้องมีความแม่นยำดังนั้นค่าการวัดจะไม่ขึ้นอยู่กับตำแหน่งของกล้อง ผลลัพธ์ที่ได้คือ ตำแหน่งของจุดสังเกตจะถูกประมาณให้อยู่ในตำแหน่งที่เหมาะสมสุดโดยตำแหน่งของกล้องจะถูกเปลี่ยนแปลงเล็กน้อยตามความแปรปรวนร่วม แต่ในกระบวนหา kalman gain นั้นจะถูกพิจารณาว่าตำแหน่งของกล้องคงที่

สาเหตุที่จะต้องใช้การปรับแก้ตำแหน่งของกล้องและตำแหน่งของจุดสังเกตแยกกันเช่นนี้เพราะ ในการปรับแก้ค่าโดยมีข้อมูลการวัดเป็นปริมาณน้อย และมีการกระจายความน่าจะเป็นของสถานะก่อนการปรับแก้สูง หากทำการปรับแก้สถานะที่มีความขึ้นต่อกันสูงเช่น ตำแหน่งของกล้องและตำแหน่งของจุดสังเกต อาจทำให้ค่าสถานะหลังปรับแก้ลู่เข้าสู่ค่าศูนย์ได้ง่าย เป็นต้นว่าตำแหน่งของกล้องมีค่าเป็นศูนย์ และตำแหน่งจุดสังเกตมีค่าเป็นศูนย์ด้วย เมื่อคำนวนตามสมการแล้วทุกอย่างถูกต้อง แต่ค่าสถานะไม่ใช่สิ่งที่ต้องการ

สำหรับความคลาดเคลื่อนของการวัด (R) ที่ใช้ในการปรับแก้นั้น จะเป็นค่าที่กำหนดขึ้นเองโดยพิจารณาจากความคาดเคลื่อนที่เกิดขึ้นจากการประมาณการเลื่อนตำแหน่ง feature ส่วนความคลาดเคลื่อนของข้อมูลการวัดในอดีตนั้นจะถูกกำหนดให้มีค่าเป็นศูนย์ เนื่องจากในการประมาณการเลื่อนตำแหน่ง feature ได้ใช้ภาพ template ที่ Reference frame ใด ๆ เป็นภาพวัดเทียบดังนั้นค่าการวัดจุดสังเกตที่ Reference frame นั้น ๆ ย่อมมีความแม่นยำสูงสุด และการใช้ข้อมูลการวัดในอดีตซ้ำ ๆ ย่อมไม่ส่งผลให้ความแม่นยำ

ของตำแหน่งจุดสังเกตมีค่ามากเกินไปในทิศทางการวัด ทั้งนี้เป็นเพราะข้อมูลการวัดในอดีตซ้ำมีความแม่นยำสูงสุดอยู่แล้ว

5.2.5 การเพิ่มจุดสังเกตและ Reference frame ใหม่ (augmentation)

จุดสังเกต หรือว่า landmark นั้นเป็นจุดที่ในการอธิบายแผนที่ของสิ่งแวดล้อมว่ามีลักษณะอย่างไร ซึ่งในขณะที่กล้องมีการเคลื่อนที่ไปเรื่อย ๆ นั้น กล้องย่อมต้องมีโอกาสตรวจเจอสิ่งแวดล้อมใหม่อยู่เรื่อย ๆ ซึ่งจุดสังเกตใหม่ก็จะถูกเพิ่มเข้ามาในแผนที่ ส่วน reference frame นั้นจะถูกเพิ่มเข้ามาในระบบก็ต่อเมื่อ reference frame เก่านั้นไม่สามารถใช้อ้างอิงได้ดีพอจึงต้องมีการเพิ่ม reference frame ใหม่ ซึ่งการเพิ่มจุดสังเกตและ Reference frame เข้าเป็นสถานะของ SLAM จะมีวิธีการดังนี้

5.2.5.1 การเพิ่มจุดสังเกต

จุดสังเกตใหม่ที่ถูกตรวจวัดได้โดย Harris corner detector ตามที่ได้เคยกล่าวถึงในส่วนการตรวจหาจุดสังเกตนั้น จะถูกเพิ่มเข้ามาในแผนที่ก็ต่อเมื่อ feature ที่เป็นข้อมูลการวัดของจุดสังเกตใหม่นั้น มีระยะห่าง (ในภาพ) ห่างจาก feature เก่าพอสมควร และเนื่องจากว่าข้อมูลการวัดจุดสังเกตนั้นมีเพียงข้อมูลทิศทางเท่านั้นดังนั้นวิธีการในการเพิ่มจุดสังเกต จะมีการสมมุติระยะห่างระหว่างกล้องและสิ่งแวดล้อม ให้มีค่าคงที่ค่าหนึ่งและการกระจายความน่าจะเป็นของระยะทางมีค่าสูงมาก ซึ่งการสมมุติระยะทางที่แตกต่างกันจะมีผลเพียงทำให้ขนาดของแผนที่ที่สร้างได้มีขนาดต่างกันเท่านั้นแต่การทำงานอื่น ๆ ยังคงทำให้เหมือนเดิม และจะเขียนได้ว่าค่าการวัดจุดสังเกตเป็น

$$z_i = \begin{bmatrix} \theta_\alpha & \theta_\beta & n \end{bmatrix}^T \tag{5.33}$$

โดยที่ θ_α และ θ_β เป็นค่าการวัดเชิงองศาในรูปแบบ yaw, pitch angles ซึ่งสำหรับ โมเดลการเพิ่มจุดสังเกตนั้นเขียนได้เป็น

$$\hat{y}_i = f(\hat{x}_c, z_i) = T_{x_c}(y_i^C) \tag{5.34}$$

$$y_i^C = \begin{bmatrix} cos(\theta_\alpha) \cdot cos(\theta_\beta) \cdot n \\ cos(\theta_\alpha) \cdot sin(\theta_\beta) \cdot n \\ sin(\theta_\alpha) \cdot n \end{bmatrix} \tag{5.35}$$

เมื่อ $T_{x_c}(y_i^C)$ เป็น Transformation function ที่แปลงตำแหน่งจุดสังเกต y_i^C จาก camera coordinate มายัง world coordinate ซึ่งหาได้จาก $T_{x_c}(y_i^C) = M(x_c) \times y_i^C$ เมื่อ $M(x)$ เป็น Transformation Matrix ของ camera frame

ส่วนการประมาณค่าความแปรปรวนร่วม (covariance) นั้นจะหาได้จาก

$$P_{y_i y_i} = F_c P_{x_c} F_c^T + F_z R F_z^T \tag{5.36}$$

$$P_{y_i m} = P_{m y_i}^T = F_c P_{x_c m} \tag{5.37}$$

โดยที่ F_c เป็น Jacobian Matrix ของฟังก์ชัน f เทียบกับ $\hat{x}_c$

F_z เป็น Jacobian Matrix ของฟังก์ชัน f เทียบกับ z_i

ซึ่งสถานะของ SLAM ใหม่จะถูกเพิ่มดังนี้

$$\hat{x}_{new} = \begin{bmatrix} \hat{x}_{old} & \hat{y}_i \end{bmatrix}^T \quad (5.38)$$

$$P_{new} = \begin{bmatrix} P_{old} & P_{my_i} \\ P_{y_im} & P_{y_iy_i} \end{bmatrix} \quad (5.39)$$

5.2.5.2 การเพิ่ม Reference frame

Reference frame จะถูกเพิ่มเมื่อไม่สามารถหา template ที่เหมาะสมในการปรับแก้ตำแหน่งแหน่ง feature ได้ซึ่งโมเดลการเพิ่มจุด Reference frame นั้นเขียนได้เป็น

$$\hat{r}_i = f(\hat{x}_c) = \begin{bmatrix} t^{WC} \\ q^{WC} \end{bmatrix} \quad (5.40)$$

กล่าวได้ว่าการเพิ่ม Reference frame นั้นจะกำหนดให้มีค่าเท่ากับ camera frame ปัจจุบันโดยตรง และค่าความแปรปรวนร่วม (covariance) นั้นจะหาได้จาก

$$P_{r_ir_i} = P_{x_cx_c} \quad (5.41)$$

$$P_{r_im} = P^T_{mr_i} = P_{x_cx_c} \quad (5.42)$$

ซึ่งสถานะของ SLAM ใหม่จะถูกเพิ่มดังนี้

$$\hat{x}_{new} = \begin{bmatrix} \hat{x}_{old} & \hat{r}_i \end{bmatrix}^T \quad (5.43)$$

$$P_new = \begin{bmatrix} P_{old} & P_{mr_i} \\ P_{r_im} & P_{r_ir_i} \end{bmatrix} \quad (5.44)$$

5.2.6 การลดจุดสังเกตออกจากระบบ

จุดสังเกตจะถูกลบออกจากสถานะของ SLAM ก็ต่อเมื่อจุดสังเกตนั้นได้หายไปจากภาพจากกล้องออมนิ ซึ่งอาจเป็นไปได้ว่าจุดสังเกตโดนบังหรือหลุดไปจาก field of view ของกล้อง และค่าความแปรปรวนร่วม (covariance) ของจุดสังเกตมีค่าสูงเกินเกณฑ์ นั้นหมายความว่าจุดสังเกตนั้นไม่ดีพอสำหรับการบรรยายแผนที่สิ่งแวดล้อม

5.3 สรุปขั้นตอนการระบุตำแหน่งพร้อมกับการสร้างแผนที่ในสามมิติ

ผลลัพธ์ของการระบุตำแหน่งและการสร้างแผนที่นั้น จะเป็นค่าสถานะที่ประมาณได้จาก SLAM ซึ่งในงานนี้ได้กำหนดให้สถานะของระบบที่ต้องการประมาณประกอบด้วย สถานะของกล้อง, สถานะของจุดสังเกต และ สถานะของกรอบอ้างอิง ดังนั้นการระบุตำแหน่งจะมีผลลัพธ์เป็นตำแหน่งของกล้อง และการสร้างแผนจะมีผลลัพธ์เป็นสถานะของจุดสังเกต ส่วนสถานะของกรอบอ้างอิง นั้นเป็นผลลัพธ์เพิ่มเติมของอัลกอริทึมที่ได้นำเสนอ สำหรับขั้นตอนการทำงานการระบุตำแหน่งพร้อมกับการสร้างแผนที่ในสามมิติด้วยกล้องวิดีโอแบบออมนินั้น จะสามารถสรุปขั้นตอนการทำงานได้ดังนี้
ในการทำงานของอัลกอริทึม จะมีเซตของจุดสังเกตสองเซต ประกอบด้วย

- เซตของจุดสังเกตที่ยัง active อยู่ (หมายความว่า จุดสังเกตนั้นยังคงปรากฏอยู่ในภาพและถูกปรับแก้อยู่เรื่อย ๆ)
- เซตของจุดสังเกตเก่า หมายความว่าจุดสังเกตเหล่านั้นไม่ปรากฏในภาพแล้ว แต่อาจจะปรากฏในภาพได้ในภายหลังเมื่อกล้องเคลื่อนที่กลับไปที่จุดเดิม

เมื่อรับภาพได้จากกล้องวิดีโอแบบออมนิ การทำงานของอัลกอริทึมจะเป็นดังนี้

ขั้นตอนที่ 1 การหาความสัมพันธ์ของจุดสังเกต และค่าการวัดของจุดสังเกต

ขั้นตอนนี้จะเป็นการหาความสัมพันธ์ และค่าการวัดของจุดสังเกตที่ยัง active อยู่โดยพิจารณาจาก feature ของจุดสังเกตที่ปรากฏในภของจุดสังเกตที่ปรากฏในภาพก่อนหน้า ซึ่งมีขั้นตอนย่อยคือ

1. ติดตามการเคลื่อนที่ของ feature จากภาพที่แล้วด้วย Optical Flow
2. ปรับแก้ความถูกต้องของตำแหน่งของ feature ด้วย Template Matching ซึ่งภาพที่กรอบอ้างอิงที่จะถูกใช้เป็น template ในการปรับแก้นั้นได้ถูกหาไว้ล่วงหน้าแล้วในเวลาก่อนหน้านี้
3. ลบจุดสังเกตที่ไม่เหมาะสมออกจาก เซตของจุดสังเกตที่ยัง active โดยวิธีพิจารณาจุดสังเกตที่ไม่เหมาะสมนั้น ก็คือ สำหรับจุดสังเกตที่หลุดออกไปจากมุมมองของภาพแล้ว หรือ สำหรับ feature ของจุดสังเกตที่มีค่า eigenvalue ต่ำและอยู่ใกล้กับ feature อื่นมากเกินไปจะถูกลบออกจาก เซตของจุดสังเกตที่ยัง active
4. การเพิ่มจุดสังเกตเข้าไปในเซตของจุดสังเกตเก่า ซึ่งสำหรับจุดสังเกตที่ถูกลบออกจากเซตของจุดสังเกตที่ยัง active นั้นอาจจะถูกเพิ่มเข้าไปในเซตของจุดสังเกตเก่า ได้หากว่าค่า eigenvalue มากสุดของความแปรร่วนร่วมของตำแหน่งของจุดสังเกต มีค่าน้อยพอในเกณฑ์รับได้ ซึ่งหมายความว่าจุดสังเกตนั้นมีความแม่นยำเพียงพอ ก็จะถือว่าจุดสังเกตนั้นเป็นจุดสังเกตเก่าซึ่งจะมีโอกาสที่จะถูกตรวจพบได้โดยกล้องออมนิได้ในภายหลัง ส่วนจุดสังเกตที่ถูกลบออกจากเซตของจุดสังเกตที่ยัง active แล้วและมีค่า eigenvalue มากเกินไป หมายความว่าจุดสังเกตนั้นไม่มีความแม่นยำพอจะเป็นจุดสังเกตได้ ดังนั้นจุดสังเกตเหล่านั้นจะถูกลบออกจากระบบอย่างถาวร นั่นหมายความว่าจะถูกลบออกจากสถานะของ SLAM ด้วย
5. สำหรับ feature ของจุดสังเกตที่ยัง active อยู่ ถูกนำไปคำนวนหาค่าการวัดของจุดสังเกต เพื่อที่จะนำไปปรับแก้สถานะของ SLAM ต่อไป

ขั้นตอนที่ 2 การประมาณการเคลื่อนที่ของกล้องด้วย SLAM

การประมาณการเคลื่อนที่ของกล้องประกอบด้วยขั้นตอนย่อยดังต่อไปนี้

1. การทำนายการเคลื่อนที่ของกล้อง หรือขั้นตอน prediction ของ SLAM ซึ่งในขั้นตอนนี้ ค่าของสถานะของกล้องนั้นจะยังถูกประมาณให้มีค่าเท่าเดิม เพราะไม่รู้ทิศทางการเคลื่อนที่ของกล้องที่แน่นอน แค่ค่าความแปรปรวนร่วมของสถานะของกล้องจะมีค่าเพิ่มมากขึ้น ตามความไม่แน่นอนที่กล้องอาจจะเคลื่อนที่ไปได้
2. การปรับแก้การเคลื่อนที่ของกล้อง หรือขั้นตอน correction ของ SLAM ซึ่งในขั้นตอนนี้ ค่าของสถานะของกล้องจะถูกปรับแก้ให้มีความถูกต้องมากขึ้น และมีความแม่นยำมากขึ้น หมายความว่าค่าความแปรปรวนร่วมของสถานะของกล้องจะมีค่าลดลง ซึ่งในการปรับแก้นั้นจะใช้ค่าการวัดของจุดสังเกตที่เวลาปัจจุบัน ร่วมกับค่าการวัดของจุดสังเกตที่กรอบอ้างอิงย่อย ประมาณ 0 ถึง 3 กรอบอ้างอิง ซึ่งกรอบอ้างอิงที่เหมาะสมจะถูกเลือกไว้ล่วงหน้าแล้วที่เวลาก่อนหน้านี้

ขั้นตอนที่ 3 ปรับแก้ตำแหน่งของจุดสังเกตด้วย SLAM

หลังจากการปรับแก้ตำแหน่งของกล้องตามขั้นตอนที่แล้ว ก็จะปรับแก้ตำแหน่งของจุดสังเกตโดยอาศัยค่าการวัดของจุดสังเกตที่เวลาปัจจุบัน ร่วมกับค่าการวัดของจุดสังเกตที่กรอบอ้างอิงย่อย ซึ่งการเลือกกรอบอ้างอิงที่เหมาะสมเหล่านั้นจะถูกเลือกในขั้นตอนถัดไปของเวลาก่อนหน้านี้

ขั้นตอนที่ 4 การจัดการกรอบอ้างอิงย่อย

ในขั้นตอนนี้จะประกอบด้วยสองส่วนหลัก คือการหากรอบอ้างอิงที่เหมาะสมสำหรับการปรับแก้ตำแหน่ง feature ของจุดสังเกต และการหากรอบอ้างอิงที่เหมาะสมสำหรับการการปรับแก้สถานะของ S-LAM ซึ่งจะอธิบายการทำงานได้ดังนี้

1. การหากรอบอ้างอิงที่เหมาะสมสำหรับการปรับแก้ตำแหน่ง feature ของจุดสังเกตนั้น กรอบอ้างอิงที่เหมาะสมจะต้องมีมุมมองจุดสังเกตไม่แตกต่างกับมุมมองปัจจุบันมากนัก และระยะห่างจากกรอบอ้างอิงและจุดสังเกตก็ไม่ควรแตกต่างกับระยะห่างจากกล้องและจุดสังเกต มากจนเกินไป ดังนั้นการเลือกกรอบอ้างอิงที่เหมาะสม จะเลือกจากกรอบอ้างอิงที่มีมุมมองจุดสังเกตแตกต่างกับมุมมองปัจจุบันไม่เกินค่าหนึ่ง และความต่างระยะห่างจะแตกต่างกันได้ไม่เกินสองเท่า ซึ่งถ้าหากไม่มีกรอบอ้างอิงใดเลยเหมาะสมกับสำหรับการปรับแก้ตำแหน่ง feature ของจุดสังเกตปัจจุบัน ก็จะทำการเพิ่มกรอบอ้างอิง โดยถือเอาตำแหน่งของกล้องปัจจุบันเป็นกรอบอ้างอิงเลย
   สำหรับในกรณีที่สามารถหากรอบอ้างอิงที่เหมาะสมได้ กรอบอ้างอิงนั้นอาจไม่มี template สำหรับการปรับแก้ตำแหน่ง feature ก็เป็นได้ทั้งนี้อาจเป็นเพราะกรอบอ้างอิงถูกเพิ่มเข้ามาก่อนจุดสังเกต ซึ่งหมายความว่า ณ เวลาที่กล้องอยู่ที่ตำแหน่งกรอบอ้างอิงนั้นยังไม่ตรวจเจอจุดสังเกตที่สนใจได้ ซึ่งวิธีแก้ไขนั้นจะใช้การประมาณการวัดของจุดสังเกตที่กรอบอ้างอิงนั้น โดยการโปรเจคตำแหน่งจุดสังเกตลงบนภาพ ณ กรอบอ้างอิงนั้น และปรับแก้ความถูกต้องของตำแหน่ง feature ที่ได้จากการโปรเจคตำแหน่งจุดสังเกต ให้สอดคล้องกับตำแหน่ง feature ณ ภาพปัจจุบัน ซึ่งวิธีนี้ ก็ให้ผลไม่ต่างจากการเพิ่มกรอบอ้างอิงใหม่มากนัก เพราะ template สำหรับการปรับแก้ตำแหน่ง feature ก็มีการเคลื่อนตำแหน่งไปจาก template จริงแต่อย่างน้อย การมี template สำหรับอ้างอิงก็ทำให้ตำแหน่ง feature ในอนาคตมีความแม่นยำในระดับที่ยอมรับได้
2. การหากรอบอ้างอิงที่เหมาะสมสำหรับการการปรับแก้สถานะของ SLAM นั้น กรอบอ้างอิงที่เหมาะสมจะต้องมีข้อมูลการวัดของจุดสังเกตที่สนใจ และข้อมูลการวัดที่กรอบอ้างอิงนั้น ควรจะตั้งฉากกับ

ข้อมูลการวัดที่เวลาปัจจุบันให้ได้มากที่สุด ทั้งนี้เป็นเพราะว่าหากข้อมูลการวัดในสองตำแหน่งนั้นตั้งฉากกันมากเท่าไร่ การประมาณตำแหน่งจุดสังเกตย่อมมีความแม่นยำมากขึ้นเท่านั้น ดังนั้นการเลือกกรอบอ้างอิงที่เหมาะสมจะเลือกโดยพิจารณาค่าการวัด ณ กรอบอ้างอิงที่ตั้งฉากกับข้อมูลการวัดที่เวลาปัจจุบันมากสุด ไม่เกิน 3 กรอบอ้างอิง

ขั้นตอนที่ 5 การเพิ่มเซตของจุดสังเกตที่ active

การเพิ่มของจุดสังเกตที่ active นั้นสามารถเพิ่มได้จากสองกรณีคือ เพิ่มจากเซตของจุดสังเกตเก่า ที่ปรากฏขึ้นในภาพจากกล้องออมนิอีกครั้ง กับการเพิ่มจากจุดสังเกตใหม่ในภาพ ซึ่งขั้นตอนย่อยสำหรับการเพิ่มจุดสังเกตนั้นอธิบายได้ดังนี้

1. การเพิ่มจุดสังเกตจากเซตของจุดสังเกตเก่า สำหรับจุดสังเกตเก่าที่ตรวจวัดได้ว่าตำแหน่งของจุดสังเกตอยู่ในมุมมองของกล้อง จะมีการพิจารณาว่าจุดสังเกตนั้นมีความเหมาะสมจะเป็นจุดสังเกตที่ active หรือไม่โดยพิจาณาเงื่อนไขดังต่อไปนี้

    - จุดสังเกตนั้นมีความต่างของ ระยะห่างระหว่างตำแหน่งจุดสังเกตและตำแหน่งกล้อง กับ ระยะห่างระหว่างตำแหน่งจุดสังเกตและกรอบอ้างอิงใด ๆ ที่มีค่าการวัดของจุดสังเกตที่สนใจ ไม่เกินเกณฑ์ที่กำหนด ซึ่งก็หมายความว่าจุดสังเกตเก่า ไม่ได้อยู่ห่างจากตำแหน่งกล้องปัจจุบันมากจนเกินไป
    - มีกรอบอ้างอิงที่มุมมองจุดสังเกต ณ กรอบอ้างอิงนั้นไม่แตกต่างกับมุมมองปัจจุบันมากนักถ้าหากสถานะของจุดสังเกตเก่าสอดคล้องกับเงื่อนไขข้างต้นก็จะทำการประมาณตำแหน่ง feature ของจุดสังเกตเก่านั้นในภาพปัจจุบันโดยการ โปรเจคตำแหน่งจุดสังเกตในสามมิติลงบนภาพและปรับแก้ความถูกต้องของตำแหน่ง feature เทียบกับกรอบอ้างอิงที่มีมุมมองและระยะเหมาะสมสุด ซึ่งอันที่จริงแล้วขั้นตอนนี้ก็คือการหาความสัมพันธ์ของจุดสังเกต เพื่อเตรียมการสำหรับใช้เป็นค่าการวัดสำหรับ SLAM ต่อไป

    ในขั้นตอนการปรับแก้ความถูกต้องของตำแหน่ง feature นั้นก็คือการปรับตำแหน่ง feature ให้สอดคล้องกับ Template ซึ่งในการปรับแก้นั้นอาจจะล้มเหลวก็ได้ ซึ่งหมายความว่าไม่มีตำแหน่งที่เหมาะสมสำหรับ feature นั้น ๆ ทั้งนี้อาจเป็นเพราะจุดสังเกตในภาพนั้น อาจถูกบดบังด้วยสิ่งกีดขวางอื่นทำให้การปรับแก้ความถูกต้องของตำแหน่ง feature ล้มเหลว ในกรณีนี้ อัลกอริทึมก็จะยังไม่เพิ่มจุดสังเกตเก่าเข้าไปในเซตของจุดสังเกตที่ active

2. การเพิ่มจุดสังเกตใหม่ จะมีการเพิ่มในขั้นตอนเริ่มต้นของระบบ หรือเมื่อจุดสังเกตในแผนที่มีจำนวนน้อยเกินไป หรือในกรณีที่มีการเพิ่มกรอบอ้างอิง ก็จะถือโอกาสเพิ่มจุดสังเกตด้วย ซึ่งการเพิ่มจุดสังเกตใหม่นั้นจะหาได้จากการตรวจวัดจุดสังเกตในภาพโดยใช้ Harris Corner Detector ซึ่งการพิจารณาจุดสังเกตใหม่นั้นจะต้องพิจารณาด้วยว่าตำแหน่ง feature ในภาพมีระยะห่างจาก feature เก่า และตำแหน่งแหน่งประมาณของจุดสังเกตเก่า ไม่น้อยเกินไปด้วย

บทที่ 6

การทดสอบการระบุตำแหน่งพร้อมกับการสร้างแผนที่ และผลการทดสอบ

สำหรับการทดสอบการระบุตำแหน่งพร้อมกับการสร้างแผนที่ในสามมิติด้วยกล้องวิดีโอแบบออมนิ จะนำเอากล้องวิดีโอแบบออมนิเคลื่อนที่ไปมาในสิ่งแวดล้อมจริง และเก็บชุดข้อมูลภาพจากกล้องออมนิเพื่อนำมาประมวลผลการระบุตำแหน่งของกล้องและการสร้างแผนที่ของสิ่งแวดล้อมในรูปแบบสามมิติ โดยวิธีการเคลื่อนที่ของกล้องนั้นจะใช้สองวิธีคือ ใช้มนุษย์ถือกล้องออมนิเคลื่อนที่ไปมาในสามมิติ และใช้หุ่นยนต์ซึ่งติดตั้งกล้องออมนิเคลื่อนที่ไปมา ซึ่งการใช้มนุษย์เป็นผู้ถือกล้องจะใช้ทดสอบการเคลื่อนที่ของกล้องอย่างอิสระในสามมิติ ส่วนการใช้หุ่นยนต์ซึ่งติดตั้งกล้องออมนินั้นจะใช้เพื่อการวัดผลเป็นหลักสำหรับการเคลื่อนที่ของกล้องในแนวระนาบ โดยการทดลองนี้มีจุดมุ่งหมายเพื่อทดสอบวิธีการระบุตำแหน่งพร้อมกับการสร้างแผนที่ในสามมิติด้วยกล้องวิดีโอแบบออมนิ ว่าสามารถใช้งานได้ในสิ่งแวดล้อมจริง และในหัวข้อสุดท้ายจะเป็นการวัดผลความถูกต้องของผลลัพธ์ที่ได้จากอัลกอริทึมเทียบกับข้อมูลจริง

6.1 อุปกรณ์ที่ใช้ในการทดลอง

6.1.1 กล้องวิดีโอแบบออมนิ

กล้องวิดีโอแบบออมนิ สำหรับการทดลองนี้เป็นกล้องออมนิรุ่น OmniView360 ของ RemoteReality ซึ่งตัวกล้องออมนิจะประกอบไปด้วยกล้องวิดีโอทั่วไป และกระจกโค้งสองชิ้นรูปทรงพาลาโบลา มีลักษณะดังรูปที่ 6.1ก และภาพที่ได้จากกล้องออมมิดังกล่าวจะปรากฏดังรูปที่ 6.1ข ซึ่งตัวกล้องมีคุณสมบัติดังนี้

- ระบบภาพ NTSC ความถี่ 29.97 ภาพต่อวินาที
- ความละเอียด 640x480 พิกเซลแบบสลับเส้น (Interlace)
- ภาพที่ได้จากกล้องออมนิตามรูปที่ 6.1ข มีความละเอียดประมาณ 480x480 พิกเซล

(ก) (ข)

รูปที่ 6.1: (ก) ภาพกล้องวิดีโอแบบออมนิที่ใช้ในงานวิจัย (ข) ภาพถ่ายจากกล้องวิดีโอแบบออมนิ

6.1.2 หุ่นยนต์ทดลอง

หุ่นยนต์สำหรับการทดลองนี้เป็นหุ่นยนต์สำหรับงานวิจัย รุ่น Pioneer DX3 ของ MobileRobots ซึ่งใช้การขับเคลื่อนแบบสองล้ออิสระ ตัวหุ่นยนต์มีขนาด 44.5cm x 40.0cm ซึ่งเมื่อติดตั้งกล้องออมนิแล้ว จะมีลักษณะดังรูปที่ 6.2

รูปที่ 6.2: หุ่นยนต์สำหรับการทดลอง

6.1.3 อุปกรณ์ Wii Remote

อุปกรณ์ Wii Remote (แสดงในรูปที่ 6.3) เป็นอุปกรณ์ควบคุมสำหรับเครื่องเล่นเกม Nintendo's Wii ซึ่งสำหรับตัวอุปกรณ์ Wii Remote เองนั้นได้ติดตั้งเซนเซอร์หลายชนิดเช่น accelerometer, กล้องอินฟาเรด ซึ่งตัวกล้องอินฟาเรด ของ Wii Remote นั้นมีความสามารถในการตรวจจับจุดแสงอินฟาเรดได้ 4 จุด ที่ความละเอียด 1024x768 พิกเซล และมีความเร็วในการรับข้อมูลถึง 100 เฟรมต่อวินาที ดังนั้นในงานวิจัยนี้จึงเป็นการเหมาะที่จะใช้ Wii Remote มาใช้ในการจับภาพมุมสูงเพื่อระบุตำแหน่งของกล้องออมนิแบบ global เพื่อใช้ในการวัดผลความถูกต้องของอัลกอริทึม

รูปที่ 6.3: อุปกรณ์ Wii Remote

6.2 สถานที่สำหรับการทดสอบ

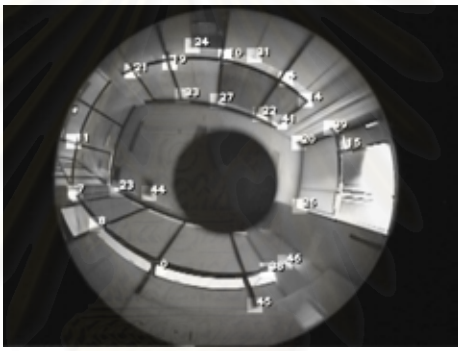
สำหรับสภานที่ในการทดสอบการระบุตำแหน่งพร้อมกับการสร้างแผนที่ในสามมิติด้วยกล้องวิดีโอแบบออมนิ จะเป็นสภาพแวดล้อมภายในอาคารซึ่งประกอบด้วยสถานที่หลายแห่งด้วยกันยกตัวอย่างเช่น ภายในห้องวิจัย, ห้องประชุม, โถงทางเดิน, ทางขึ้นลงบันได, ระเบียงทางเดิน, บริเวณห้องโถง ซึ่งในแต่ละสถานที่ก็จะมีลักษณะสิ่งแวดล้อมที่แตกต่างกันไป

6.3 ขั้นตอนการทดสอบ

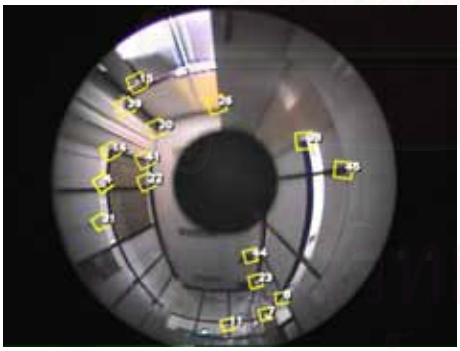
ในการทดสอบการระบุตำแหน่งพร้อมกับการสร้างแผนที่ด้วยกล้องวิดีโอแบบออมนิ จะใช้มนุษย์ถือกล้องออมนิเคลื่อนที่ไปมาในสิ่งแวดล้อมที่สนใจ หรือใช้หุ่นยนต์ติดตั้งกล้องออมนิและบังคับหุ่นยนต์ผ่านโปรแกรมควบคุมให้หุ่นยนต์เคลื่อนที่ไปในสิ่งแวดล้อมที่สนใจ จากนั้นจึงทำการบันทึกภาพที่กล้องออมนิตรวจจับได้ แล้วจึงนำภาพที่ได้มาประมวลผลสำหรับการระบุตำแหน่งและกับการสร้างแผนที่ด้วยโปรแกรมที่พัฒนาขึ้นด้วยภาษา C++ ร่วมกับ OpenCV และ OpenGL จากนั้นจึงแสดงผลลัพธ์การระบุตำแหน่งและการสร้างแผนที่ออกมาทางหน้าจอในรูปแบบสามมิติ

6.4 ผลการวิเคราะห์ข้อมูลภาพจากกล้องวิดีโอแบบออมนิ

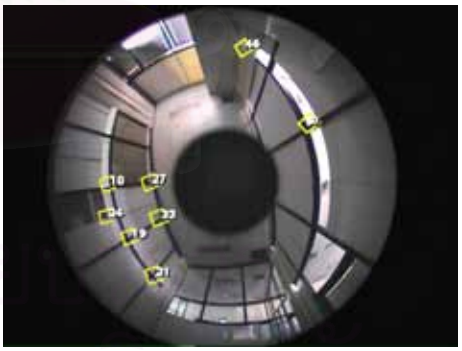
การวิเคราะห์ข้อมูลภาพจากกล้องวิดีโอแบบออมนินั้นก็เพื่อ ตรวจหาข้อมูลการวัดจุดสังเกต (landmark) และหาความสัมพันธ์ว่าข้อมูลการวัดจุดสังเกตเหล่านั้นสอดคล้องตำแหน่งจุดสังเกตที่ปรากฏบนแผนที่ของ SLAM ตำแหน่งไหนบ้าง หรือว่าข้อมูลการวัดจุดสังเกตนั้นเป็นข้อมูลจุดสังเกต จุดใหม่ซึ่งยังไม่ปรากฏในแผนที่ ซึ่งผลการวิเคราะห์ข้อมูลนั้นสามารถแสดงได้ดังรูปที่ 6.4



(ก) ผลการติดตามการเลื่อนตำแหน่งของ feature



(ข) ภาพ template



(ค) ภาพ template

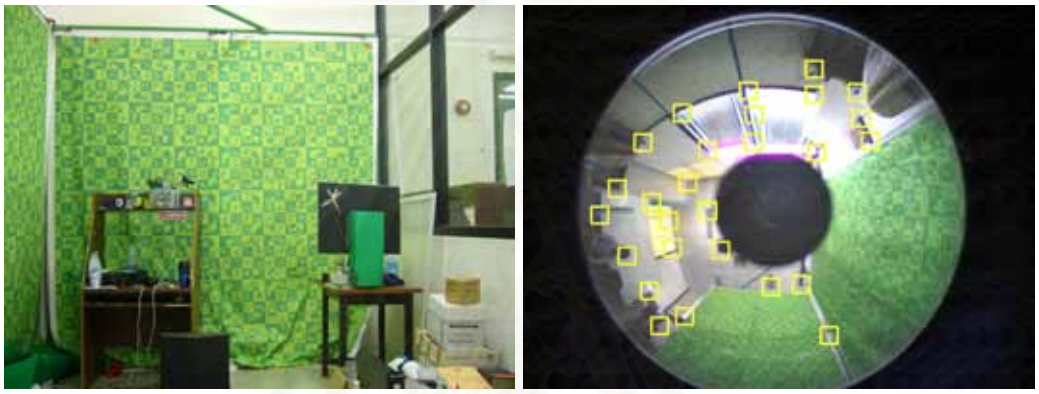
รูปที่ 6.4: ผลการวิเคราะห์ข้อมูลภาพจากกล้องวิดีโอแบบออมนิ

รูปที่ 6.4ก แสดงผลการติดตามการเลื่อนตำแหน่งของ feature และการปรับแก้ตำแหน่งของ feature เทียบกับภาพ template (รูปที่ 6.4ข, 6.4ค) จะเห็นได้ว่า patch ในกรอบสี่เหลี่ยมของภาพปัจจุบันกับภาพ template นั้นมีความสอดคล้องกัน และ patch ในภาพ template นั้นได้ถูกปรับแก้ด้วย Perspective distortion ให้สอดคล้องกับภาพปัจจุบันด้วย ซึ่งเมื่อกล้องออมนิเคลื่อนที่ไปมาเป็นเวลานาน ตำแหน่งของ feature จะไม่มีการเลื่อนออกไปจากจุดที่ควรจะเป็น ทั้งนี้เพราะผลจากการปรับแก้ตำแหน่ง feature ด้วย template matching

6.5 ผลการทดสอบการระบุตำแหน่งพร้อมกับการสร้างแผนที่

6.5.1 ผลการทดสอบภายในห้องวิจัย

ห้องวิจัยในที่นี้จะเป็นห้อง lab ISL2 ซึ่งลักษณะของห้องจะประกอบไปด้วยฉากลายสีเขียว (ซึ่งใช้ในงานวิจัยอื่น) ดังรูปที่ 6.5ก บริเวณห้องจะมีความซับซ้อนของสิ่งแวดล้อมสูง เนื่องจากห้อง lab ประกอบด้วยอุปกรณ์สำหรับการทดลองจำนวนมาก ทำให้ระบบสามารถตรวจหาจุดสังเกตได้เป็นจำนวนมาก แต่จะตรวจหาความสัมพันธ์ของจุดสังเกตได้ยาก อีกทั้งบริเวณฉากลายสีเขียวนั้นยังอาจจะทำให้เกิดความกำกวมของ feature ได้ง่าย ดังนั้นห้อง lab ISL2 จึงเหมาะในการใช้เพื่อการทดสอบประสิทธิภาพของอัลกอริทึมการตรวจวัดและหาความสัมพันธ์ของจุดสังเกตได้เป็นอย่างดี สำหรับการทดลองนี้จะใช้หุ่นยนต์ติดตั้งกล้องออมนิ เคลื่อนที่ไปในสิ่งแวดล้อมเพื่อบันทึกภาพ โดยตัวอย่างภาพ และ feature ที่ตรวจจับได้แสดงได้ในรูป 6.5ข

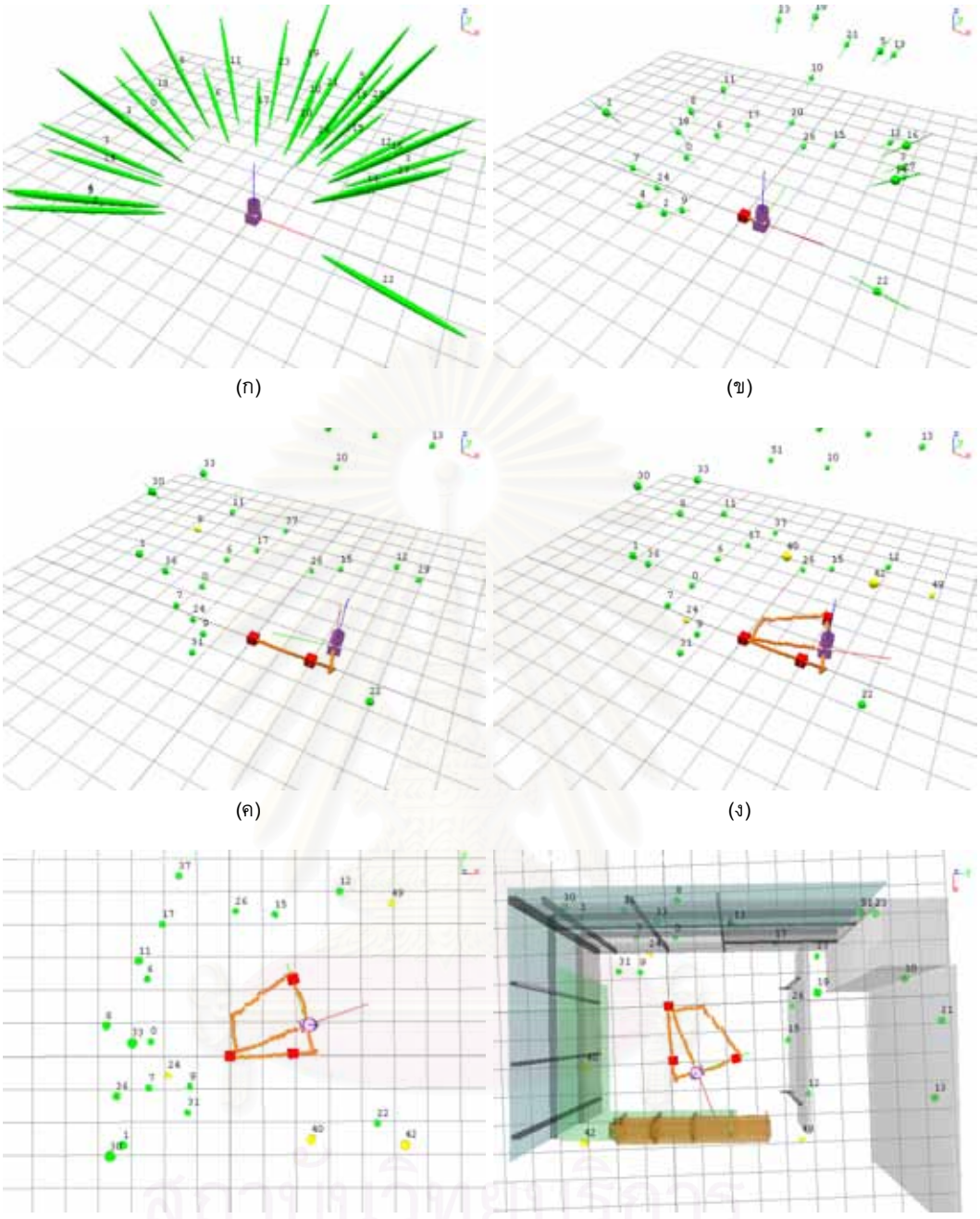


(ก) (ข)

รูปที่ 6.5: (ก) สภาพแวดล้อมห้องวิจัย ISL2 (ข) ภาพจากกล้องออมนิ และ feature ที่ตรวจจับได้

ผลการระบุตำแหน่งพร้อมกับการสร้างแผนที่แสดงได้ในรูปที่ 6.6 โดย รูปทรงรีสีเขียวนั้นจะแสดงถึงตำแหน่งของจุดสังเกตในแผนที่ และความแปรปรวนร่วม ซึ่งถ้าหากรูปทรงรีมีขนาดใหญ่ แสดงว่ามีความแปรปรวมมาก และสำหรับตำแหน่งกล้องนั้นแสดงเป็นรูปทรงกระบอกตรงกลางภาพ สำหรับกล่องลูกบาศก์นั้นจะเป็นตำแหน่งของ Reference frame ส่วนแนวเส้นเป็นเส้นแสดงการเคลื่อนที่ของกล้อง

จากรูปที่ 6.6ก แสดงขั้นตอนเริ่มต้นของระบบจะเห็นได้ว่าตำแหน่งของจุดสังเกตทุกตำแหน่งมีระยะห่างจากตัวกล้องเท่า ๆ กันหมด และมีการกระจายความน่าจะเป็นในด้านระยะทางจากตัวกล้องสูงมาก อันเป็นผลมาจากขั้นตอนการเพิ่มจุดสังเกตเข้าไปในแผนที่ ซึ่งภาพจากกล้องออมนิไม่มีข้อมูลความลึก จึงกำหนดให้ความน่าจะเป็นของความลึกมีการกระจายมาก จากนั้นเมื่อกล้องเคลื่อนที่ไปได้ระยะหนึ่งจะเห็นได้ว่าจุดสังเกตจะเริ่มเป็นรูปเป็นร่างของแผนที่มากขึ้น ขณะเดียวกันระบบก็สามารถประมาณการเคลื่อนที่ไปข้างหน้าของกล้องได้ดังรูปที่ 6.6ข เมื่อกล้องเคลื่อนที่ไประยะไกลมากขึ้น reference frame ก็จะถูกเพิ่มเข้ามาเป็นระยะรูปที่ 6.6ค และเมื่อกล้องเคลื่อนที่ไปเรื่อย ๆ SLAM ก็จะค่อย ๆ ประมาณตำแหน่งจุดสังเกตในแผนที่ และตำแหน่งให้มีความแม่นยำมากขึ้น ดังรูปที่ 6.6ง,6.6จ ส่วนรูปที่ 6.6ฉ จะแสดงแผนที่ที่สร้างได้ เทียบกับโครงร่างของสิ่งแวดล้อมจริง



(ก) (ข)

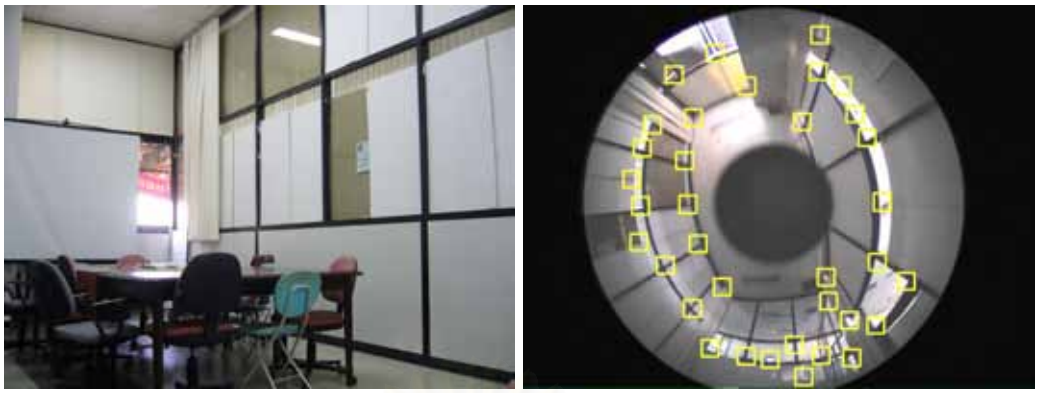
(ค) (ง)

(จ) (ฉ)

รูปที่ 6.6: ผลการระบุตำแหน่งพร้อมกับการสร้างแผนที่

6.5.2 ผลการทดสอบภายในห้องประชุม

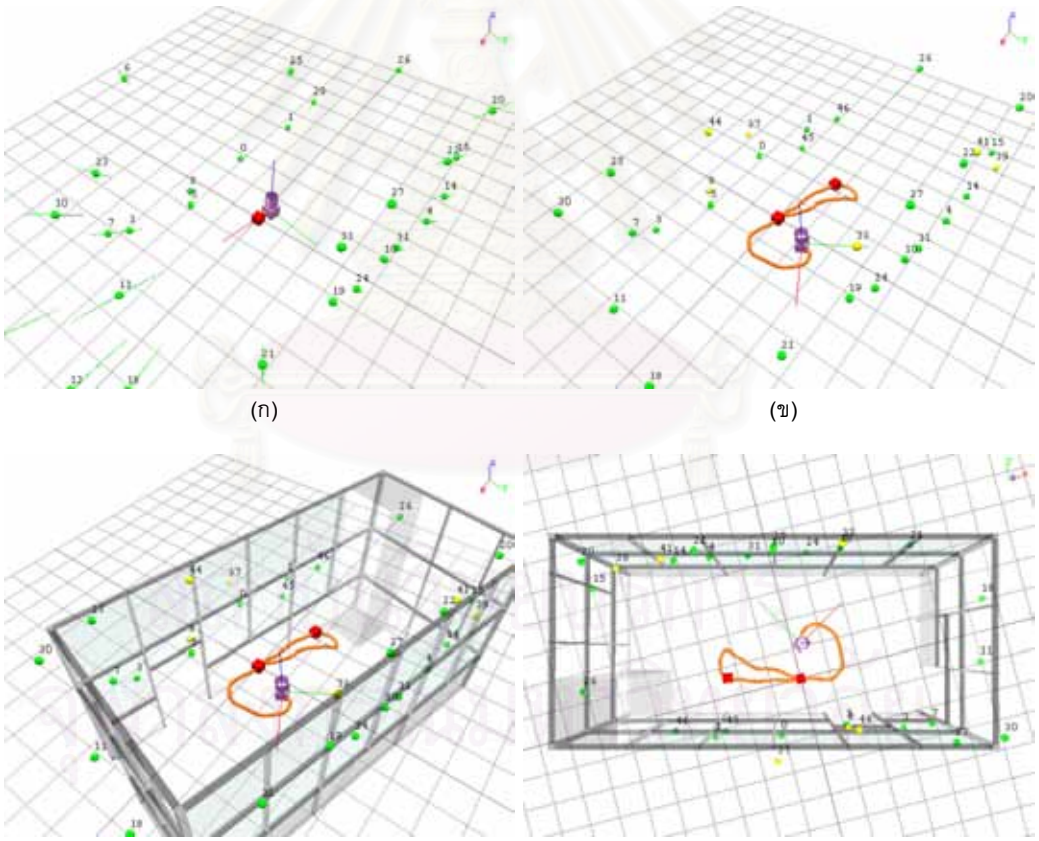
ห้องประชุมในที่นี้จะเป็นห้องประชุม 20-01 ชั้น 20 ตึกวิศวกรรมศาสตร์ 4 ซึ่งลักษณะห้องจะเป็นห้องขนาดเล็กสิ่งแวดล้อมมีความซับซ้อนน้อย ทำให้หาความสัมพันธ์ของจุดสังเกตได้ดี และเนื่องจากรูปแบบโครงสร้างห้องประชุมนี้ มีแบบแผนที่ค่อนข้างเป็นระเบียบ ทำให้เหมาะแก่การวัดผลการสร้างแผนที่เป็นอย่างมากซึ่ง ห้องประชุมนี้จะถูกใช้เป็นห้องเพื่อการวัดผลการสร้างแผนที่ต่อไป สำหรับลักษณะของห้องแสดงได้ดังรูปที่ 6.7ก ส่วนตัวอย่างภาพ และ feature ที่ตรวจจับได้จากภาพจากกล้องออมนิ แสดงได้ในรูปที่ 6.7ข



(ก) (ข)

รูปที่ 6.7: (ก) สภาพแวดล้อมห้องประชุม (ข) ภาพจากกล้องออมนิ และ feature ที่ตรวจจับได้

ในการทดลองนี้จะใช้มนุษย์ในการถือกล้องออมนิ และเคลื่อนที่วนไปมาในสิ่งแวดล้อม สำหรับผลการทดลองแสดงได้ดังรูปที่ 6.8



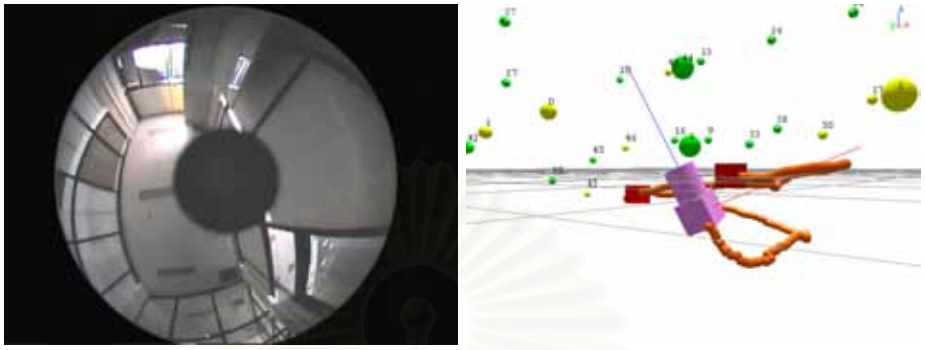
(ก) (ข)

(ค) (ง)

รูปที่ 6.8: ผลการระบุตำแหน่งพร้อมกับการสร้างแผนที่

เมื่อกล้องเคลื่อนที่วนซ้ำไปมา จะเห็นได้ว่า SLAM สามารถระบุตำแหน่งและสร้างแผนที่ได้ดี และเมื่อลองนำแผนที่ที่สร้างได้ มาเปรียบเทียบกับโมเดลของห้องประชุมดังรูปที่ 6.8ค และรูปที่ 6.8ง จะเห็นได้ว่าแผนที่ที่สร้างได้ก็ค่อนข้างสอดคล้องกับโมเดลของห้องประชุม

สำหรับภาพที่ 6.9ข นั้นแสดงผลของการระบุตำแหน่งเมื่อมีการเคลื่อนที่กล้องวิดีโออย่างอิสระในสามมิติ และภาพที่ได้จากกล้องแสดงได้ดังรูปที่ 6.9ก จะเห็นได้ว่ากล้องออมนิที่ใช้ในการทดลองนั้นสามารถเคลื่อนที่ขึ้นลงในสามมิติได้อย่างอิสระร่วมถึงสามารถเอียงกล้องไปมาได้ โดยที่อัลกอริทึมที่ได้นำเสนอนั้นยังสามารถระบุตำแหน่งและสร้างแผนที่ได้เป็นอย่างดี

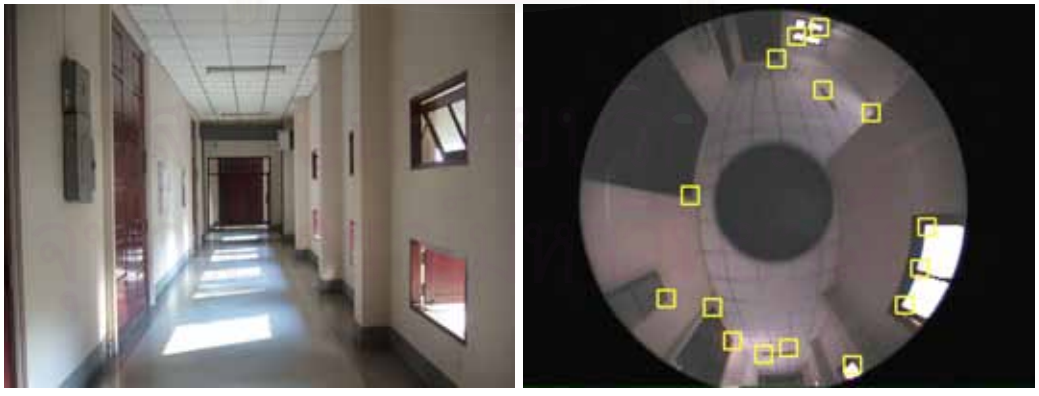


(ก) (ข)

รูปที่ 6.9: ผลการระบุตำแหน่งพร้อมกับการสร้างแผนที่

6.5.3 ผลการทดสอบบริเวณโถงทางเดิน

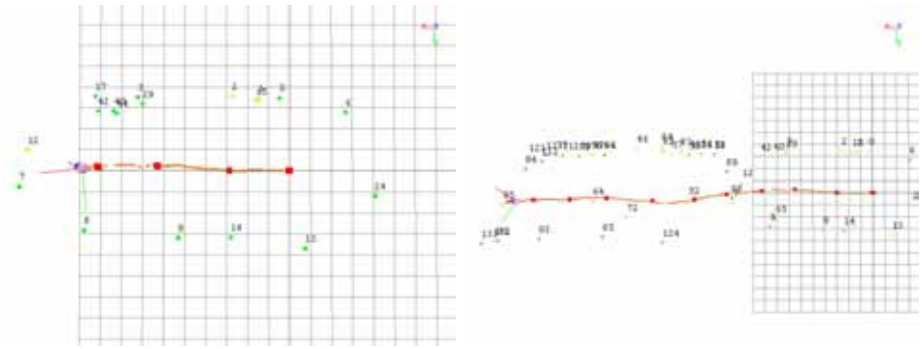
โถงทางเดินที่ใช้ในการทดลองจะเป็นโถงทางเดินบริเวณชั้น 4 ตึกวิศวกรรมศาสตร์ 3 ซึ่งลักษณะของโถงนั้นจะเป็นทางเดินยาว มีกำแพงรอบด้าน เหมาะสำหรับทดสอบการระบุตำแหน่งและการสร้างแผนที่ในระยะทางไกล และเหมาะสำหรับการทดสอบว่าเมื่อกล้องเคลื่อนที่ไปเป็นระยะทางไกลและย้อนกลับมาที่จุดเดิมแล้ว ระบบจะยังสามารถระบุตำแหน่งที่จุดเดิมได้หรือไม่ และเนื่องจากลักษณะสิ่งแวดล้อมของบริเวณโถงทางเดินมีลักษณะเป็นกำแพงเรียบเสียเป็นส่วนมากมี feature ปรากฏในภาพน้อย ดังนั้นบริเวณโถงทางเดินจึงเหมาะสำหรับการทดสอบการทำงานของระบบในสิ่งแวดล้อมที่มีจุดสังเกตน้อยอีกด้วย สำหรับลักษณะของโถงทางเดิน แสดงได้ดังรูปที่ 6.10ก ส่วนตัวอย่างภาพและ feature ที่ตรวจจับได้จากภาพจากกล้องออมนิ แสดงได้ในรูปที่ 6.10ข



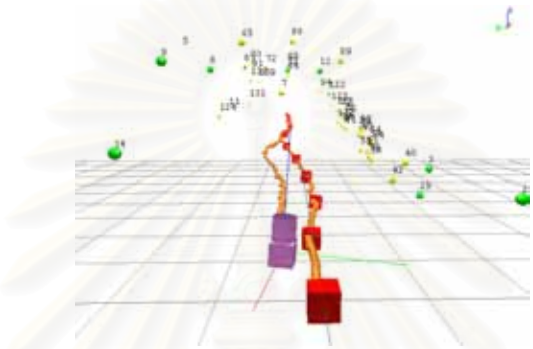
(ก) (ข)

รูปที่ 6.10: (ก) สภาพแวดล้อมบริเวณโถงทางเดิน (ข) ภาพจากกล้องออมนิ และ feature ที่ตรวจจับได้

ในการทดลองนี้จะใช้มนุษย์เดินถือกล้องออมนิ และเคลื่อนที่ไปและกลับในบริเวณบริเวณโถงทางเดิน สำหรับผลการทดลองแสดงได้ดังรูปที่ 6.11



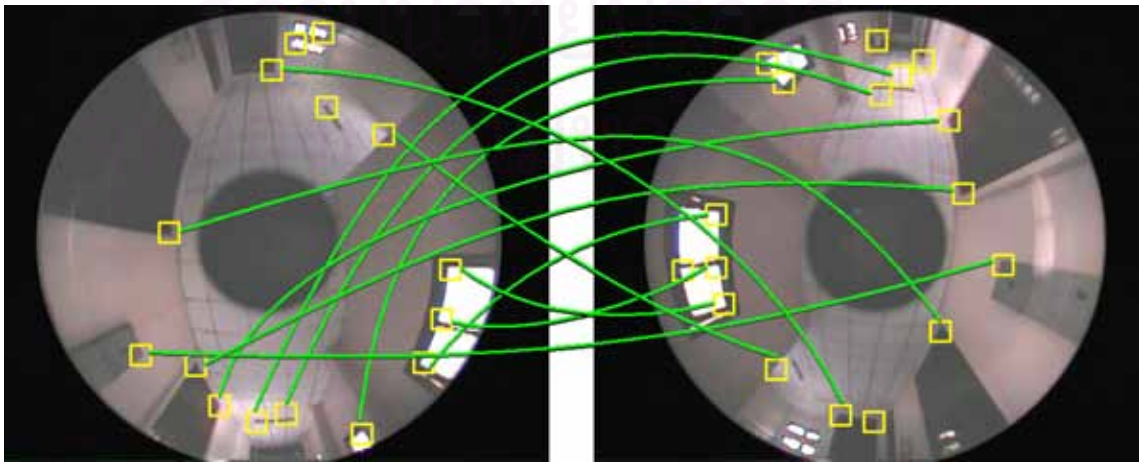
(ก) (ข)



(ค)

รูปที่ 6.11: ผลการระบุตำแหน่งพร้อมกับการสร้างแผนที่

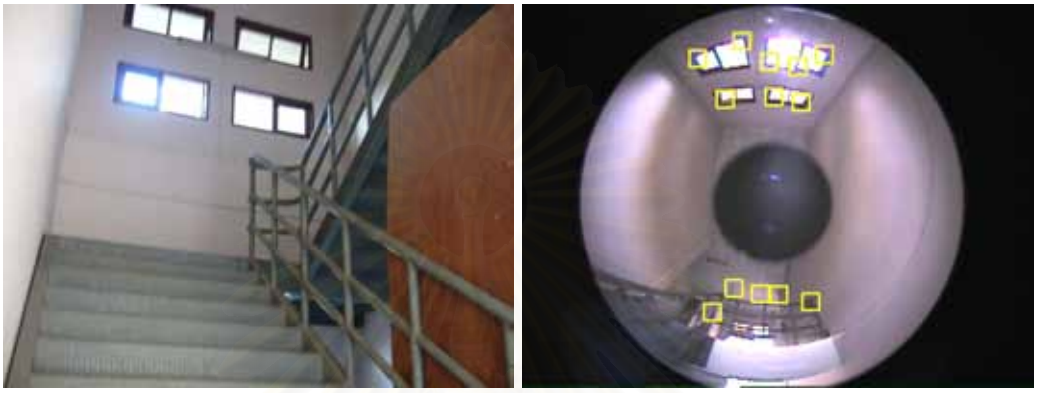
จากผลการทดลองจะเห็นได้ว่า การระบุตำแหน่งและสร้างแผนที่จะยังพอจะสามารถทำงานได้ แม้ว่าจะมีข้อมูลสิ่งแวดล้อมน้อย และรูปที่ 6.11ข แสดงให้เห็นการระบุตำแหน่งของกล้องและการสร้างแผนที่สำหรับการเคลื่อนที่ระยะไกล และรูปที่ 6.11ค แสดงให้เห็นว่ากล้องเคลื่อนที่ไปเป็นระยะทางไกลและย้อนกลับมาที่จุดเดิมแล้ว ระบะจะยังสามารถระบุตำแหน่งที่จุดเดิม โดยในรูปที่ 6.12 จะแสดงให้เห็นความสัมพันธ์ของ feature ของจุดสังเกตระหว่าง ภาพที่เวลาเริ่มต้น และภาพเมื่อกล้องเคลื่อนที่ไปและกลับมาที่จุดเดิม ซึ่งจะเห็นได้ว่าตำแหน่งของ feature หลังจากที่กล้องเคลื่อนที่เป็นเวลานานยังคงสอดคล้องกับตำแหน่ง feature ของภาพที่เวลาเริ่มต้น ทั้งนี้เป็นผลเนื่องมาจากการปรับแก้ตำแหน่ง feature (Feature Refinement) จึงทำให้สามารถระบุตำแหน่งของกล้องได้เมื่อกล้องย้อนกลับมาที่จุดเดิม



รูปที่ 6.12: ความสัมพันธ์ของ feature ของจุดสังเกตระหว่างภาพที่เวลาเริ่มต้นกับภาพ ณ เวลาปัจจุบัน

6.5.4 ผลการทดสอบบริเวณทางขึ้นลงบันได

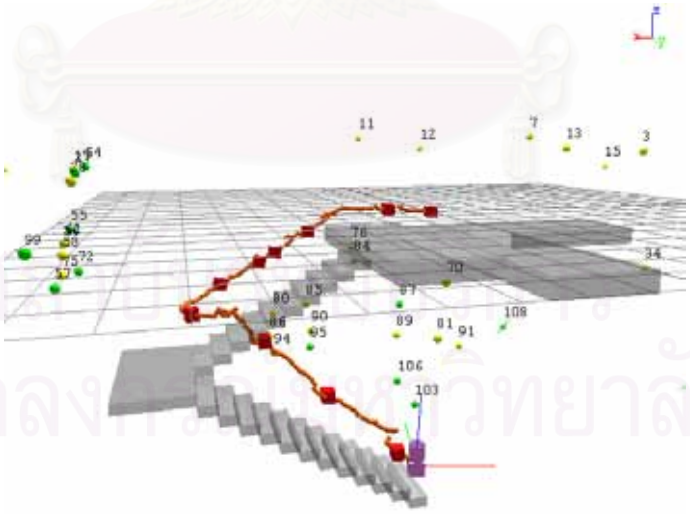
ทางขึ้นลงบันไดที่ใช้ในการทดลองเป็นบันไดขึ้นลงระหว่างชั้น 3 และชั้น 4 ตึกวิศวกรรมศาสตร์ 3 ลักษณะของบันไดจะเป็นบันไดสองช่วงมีจุดพักตรงกลางระหว่างชั้น ซึ่งบริเวณทางขึ้นลงบันไดนั้น เหมาะสำหรับการทดสอบการระบุตำแหน่งและการสร้างแผนที่ โดยที่บริเวณสิ่งแวดล้อมที่สนใจนั้นไม่ได้อยู่ในระนาบเดียวกัน ลักษณะของบันไดแสดงได้ดังรูปที่ 6.13ก ส่วนตัวอย่างภาพและ feature ที่ตรวจจับได้จากภาพจากกล้องออมนิ แสดงได้ในรูปที่ 6.13ข



(ก) (ข)

รูปที่ 6.13: (ก) สภาพแวดล้อมบริเวณบันได (ข) ภาพจากกล้องออมนิ และ feature ที่ตรวจจับได้

สำหรับการทดลองนี้จะใช้มนุษย์เดินถือกล้องออมนิ และเดินลงบันได ผลการทดลองแสดงได้ดังรูปที่ 6.14 ซึ่งจะเห็นได้ว่าอัลกกอริทึมที่นำเสนอ สามารถระบุตำแหน่งการเคลื่อนที่ลงบันไดของกล้องออมนิ และสร้างแผนที่ได้



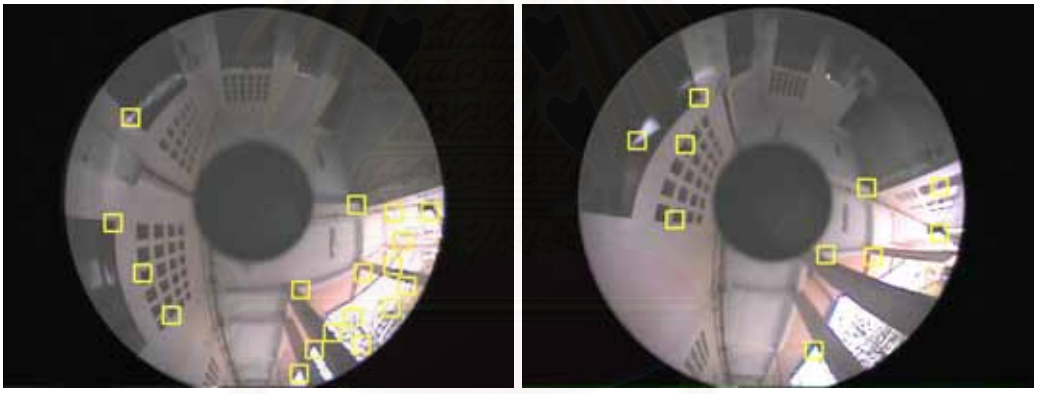
รูปที่ 6.14: ผลการระบุตำแหน่งพร้อมกับการสร้างแผนที่

6.5.5 ผลการทดสอบบริเวณระเบียงทางเดิน

ระเบียงทางเดินที่ใช้ในการทดสอบนั้นเป็นระเบียงทางเดินบริเวณชั้น 3 ตึกวิศวกรรมศาสตร์ 3 ซึ่งสิ่งแวดล้อมบริเวณระเบียงนั้นมีลักษณะกึ่งภายในอาคารกึ่งภายนอกอาคาร ทั้งนี้เนื่องจากลักษณะของระเบียงนั้นเป็นทางเดินยาวมีเพดานสูง ทางด้านหนึ่งของระเบียงนั้นเปิดโล่ง สามารถมองเห็นสภาพ

แวดล้อมภายนอกได้ทั่วและมีแสงสว่างจากภาพนอกอาคารเป็นแหล่งกำเนิดแสงหลัก บริเวณภายนอกนั้นมีลักษณะเป็นต้นไม้จำนวนมาก ดังนั้นบริเวณระเบียงทางเดิน จึงเหมาะแก่การทดสอบวิธีการระบุตำแหน่งและการสร้างแผนที่ในสภาพแวดล้อมแบบสุดขั่ว เนื่องจากลักษณะของต้นไม้ภายนอกอาคารนั้นนอกจากจะอยู่ไกลจากตัวอาคารแล้วจังมีความซับซ้อนของจุดสังเกตมากจนไม่สามารถตรวจหาความสัมพันธ์ของจุดสังเกตได้ดี นอกจากนี้ยังมีแสงจากภายนอกรบกวนการทำงานของกล้องอีกด้วย ลักษณะของห้องจะแสดงได้ดังรูปที่ 6.15 ส่วนตัวอย่างภาพและ feature ที่ตรวจจับได้จากภาพจากกล้องออมนิ แสดงได้ในรูปที่ 6.16

รูปที่ 6.15: สภาพแวดล้อมบริเวณระเบียงทางเดิน

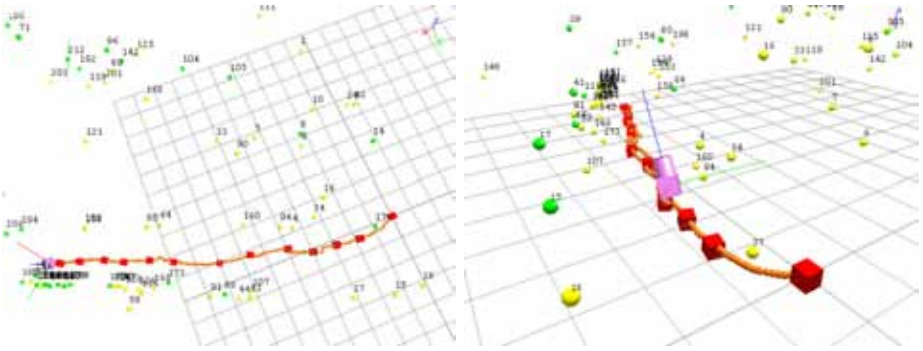


(ก) (ข)

รูปที่ 6.16: ตัวอย่างภาพจากกล้องออมนิ และ feature ที่ตรวจจับได้

จากรูปที่ 6.16ก แสดง feature ที่ตรวจวัดได้ จะเห็นได้ว่าในภาพนั้นจะประกอบด้วย feature ที่ใช้งานไม่ได้เป็นจำนวนมาก ซึ่งจะประกอบไปด้วย feature ของต้นไม้ หรือ feature ที่เกิดจากเสาและสภาพแวดล้อมด้านหลัง ซึ่ง feature ประเภทนี้ไม่ถือว่าเป็น feature สำหรับจุดสังเกตเพราะไม่สามารถระบุตำแหน่งของจุดสังเกตที่แน่นอนได้ และ feature เหล่านี้จะถูกลบทิ้งเมื่อเวลาผ่านไปดังแสดงได้ในรูปที่ 6.16ข และถึงแม้ว่า ภาพที่ได้จากกล้องออมนิ จะประกอบด้วย feature ที่ใช้ไม่ได้จำนวนมาก แต่ว่าอัลกอริทึมที่นำเสนอ ยังพอจะสามารถระบุตำแหน่งและสร้างแผนที่ได้ดังแสดงในรูปที่ 6.17

จากรูปที่ 6.17ก แสดงผลลัพธ์การระบุตำแหน่งและสร้างแผนที่ เมื่อใช้มนุษย์ถือกล้องออมนิและเดินไปมาในบริเวณระเบียง และรูปที่ 6.17ข แสดงผลลัพธ์เมื่อกล้องเคลื่อนที่กลับยังตำแหน่งเดิม จะเห็นได้ว่าอัลกกอริทึมที่นำเสนอ ยังสามารถระบุตำแหน่งได้ และถึงแม้ว่าผลลัพธ์การระบุตำแหน่งอาจจะดูไม่แม่นยำนัก แต่อัลกอริทึมยังสามารถรู้จำได้ว่าในบริเวณไหนเป็นบริเวณที่กล้องเคยเคลื่อนที่ผ่านมาแล้ว และพอกล้องเคลื่อนที่กลับมาที่จุดเดิมอีกครั้ง ก็สามารถระบุตำแหน่ง ณ จุดเดิมได้

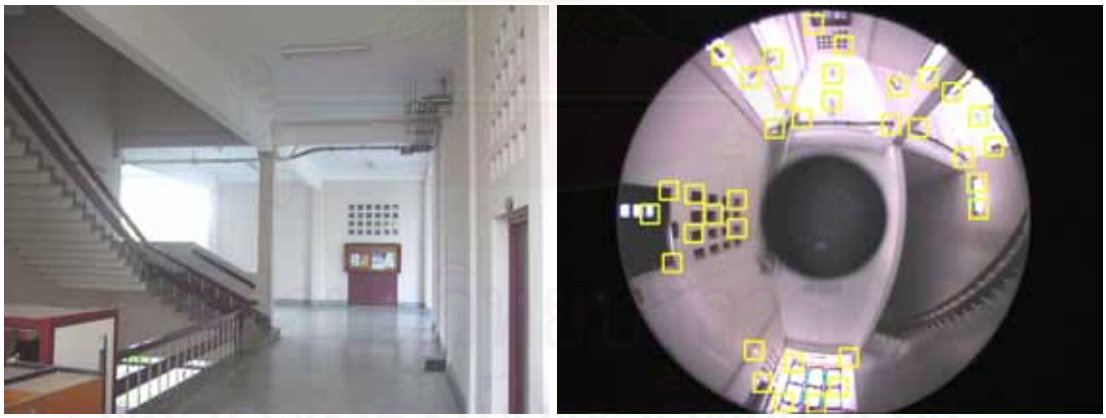


(ก) (ข)

รูปที่ 6.17: ผลการระบุตำแหน่งพร้อมกับการสร้างแผนที่

6.5.6 ผลการทดสอบบริเวณห้องโถง

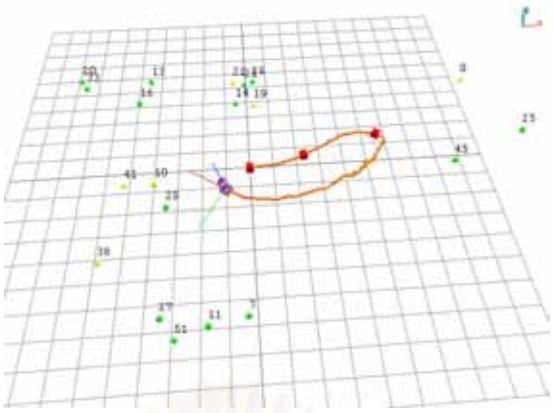
บริเวณห้องโถงที่ใช้ในการทดสอบนี้จะเป็นโถงขนาดใหญ่บริเวณหน้าห้องน้ำชั้น 3 ตึกวิศวกรรมศาสตร์ 3 ลักษณะสภาพแวดล้อมนั้นเป็นพื้นที่โล่งมีเพดานสูง มีบันไดอยู่ด้านข้างและผนังแต่ละด้านอยู่ห่างกันมาก ดังนั้นจึงเหมาะแก่การทดสอบวิธีการระบุตำแหน่งและการสร้างแผนที่ในบริเวณกว้าง ซึ่งความท้าทายของการระบุตำแหน่งและการสร้างแผนที่ในบริเวณกว้างนั้นก็คือเนื่องจากว่าสิ่งแวดล้อมในแต่ละด้าน อยู่ห่างจากตัวกล้องมาก ดังนั้นการเคลื่อนที่ของกล้องจึงส่งผลต่อการเปลี่ยนแปลงของสิ่งแวดล้อมในภาพจากกล้องน้อยมาก ดังนั้นจึงเป็นการยากในการสร้างแผนที่และระบุตำแหน่งของกล้องในบริเวณโลงกว้าง ลักษณะของบริเวณห้องโถงแสดงได้ดังรูปที่ 6.18ก ส่วนตัวอย่างภาพและ feature ที่ตรวจจับได้จากภาพจากกล้องออมนิ แสดงได้ในรูปที่ 6.18ข



(ก) (ข)

รูปที่ 6.18: (ก) สภาพแวดล้อมบริเวณห้องโถง (ข) ภาพจากกล้องออมนิ และ feature ที่ตรวจจับได้

สำหรับการทดลองนี้จะใช้มนุษย์เดินถือกล้องออมนิ และเดินไปมาบริเวณโถง ผลการทดลองแสดงได้ดังรูปที่ 6.19



รูปที่ 6.19: ผลการระบุตำแหน่งพร้อมกับการสร้างแผนที่

สำหรับผลการทดลองทั้งหมดทั้งหลายเหล่านี้เป็นผลการทดลองการระบุตำแหน่งของกล้องพร้อมกับสร้างแผนที่ในสามมิติด้วยกล้องวิดีโอแบบออมนิ ซึ่งจะเห็นได้ว่าอัลกอริทึมที่ได้นำเสนอนั้นสามารถระบุตำแหน่งและสร้างแผนที่ได้ในสิ่งแวดล้อมหลาย ๆ รูปแบบโดยลักษณะของการระบุตำแหน่งนั้น จะระบุเป็นพิกัดตำแหน่งของกล้องในสามมิติ และทิศทางของกล้องในรูปแบบ Euler angles ส่วนลักษณะของแผนที่ที่สร้างได้จะเป็นลักษณะของกลุ่มตำแหน่งของจุดสังเกตในสามมิติ ซึ่งรูปแบบของแผนที่นั้นจะเป็นรูปแบบที่เหมาะสมสำหรับการนำมาใช้เพื่อเป็นกรอบอ้างอิงในการระบุตำแหน่งของกล้อง แต่ข้อมูลแผนที่นี้อาจจะไม่เหมาะสมสำหรับงานอื่น ๆ เช่นงานหาเส้นทางการเคลื่อนที่ของหุ่นยนต์ (path planning) ทั้งนี้เป็นเพราะข้อมูลแผนที่นั้นไม่ได้ให้ข้อมูลว่าบริเวณไหนที่สามารถเคลื่อนที่ผ่านได้ ดังนั้นสำหรับในงานอื่น ๆ ของหุ่นยนต์นั้น อาจจะต้องสร้างแผนที่ในรูปแบบอื่นที่เหมาะสมสำหรับการใช้งานนั้นๆ ซึ่งการสร้างแผนที่รูปแบบอื่นนั้นสามารถทำได้ไม่ยากโดยอาศัยข้อมูลการระบุตำแหน่งของกล้องที่หาได้จาก SLAM ก็จะสามารถสร้างแอผนที่รูปแบบอื่นได้ การการสร้างแผนที่ที่เหมาะสมกับงานอื่น ๆ จะไม่อยู่ในขอบเขตของงานวิจัยนี้

6.6 การวัดผลความถูกต้องของการระบุตำแหน่ง

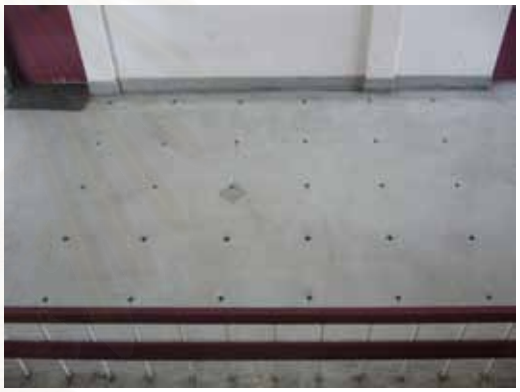
การวัดผลความถูกต้องของการระบุตำแหน่ง มีจุดประสงค์เพื่อหาความแม่นยำของวิธีการระบุตำแหน่งในสามมิติที่ได้นำเสนอในงานวิจัยนี้ ซึ่งในการวัดความถูกต้องของการระบุตำแหน่งนั้นจะใช้การวัดเทียบผลลัพธ์การระบุตำแหน่งในสามมิติที่ได้จากวิธีการที่ได้นำเสนอ และตำแหน่งที่ระบุได้จากอุปกรณ์ระบุตำแหน่งที่มีความน่าเชื่อถือ และมีความคลาดเคลื่อนคงที่ซึ่งการวัดผลความถูกต้องของการระบุตำแหน่งจะแบ่งเป็นสองวิธีด้วยกัน การวัดผลความถูกต้องของการระบุตำแหน่งในระนาบสองมิติ และ การวัดผลความถูกต้องของการระบุตำแหน่งในสามมิติ

6.6.1 การวัดผลความถูกต้องของการระบุตำแหน่งในระนาบสองมิติ

ในการวัดผลการระบุตำแหน่งในสองมิติในที่นี้ จะใช้หุ่นยนต์ที่ติดตั้งกล้องวิดีโอแบบออมนิเคลื่อนที่ไปมาในระนาบสองมิติโดนที่ปลายกล้องออมนินั้นจะติดตั้งแหล่งกำเนิดแสงอินฟาเรดไว้ 6.20ก ซึ่งแสงอินฟาเรดนั้นจะถูกตรวจจับโดยอุปกรณ์ Wii Remote ซึ่งติดตั้งเป็นกล้องมุมสูงดังรูปที่ 6.20ข ตัวอุปกรณ์ Wii Remote นั้นจะถูก Calibrate ด้วยจุดจำนวนมากที่รู้ตำแหน่งแน่นอนบนพื้นดังรูปที่ 6.20ค ทำให้สามารถทราบตำแหน่งที่แน่นอนของอุปกรณ์ Wii Remote จากนั้นจึงสามารถระบุตำแหน่งของกล้องออมนิในระนาบสองมิติได้ ซึ่งการระบุตำแหน่งนั้นก็เป็นเพียงแค่การแปลง perspective projection ของจุดอินฟาเรดที่อุปกรณ์ Wii Remote ตรวจจับได้ให้ไปอยู่ในพิกัดในระนาบสองมิติซึ่งตัวอย่างโปรแกรมแสดงได้ในรูปที่ 6.21

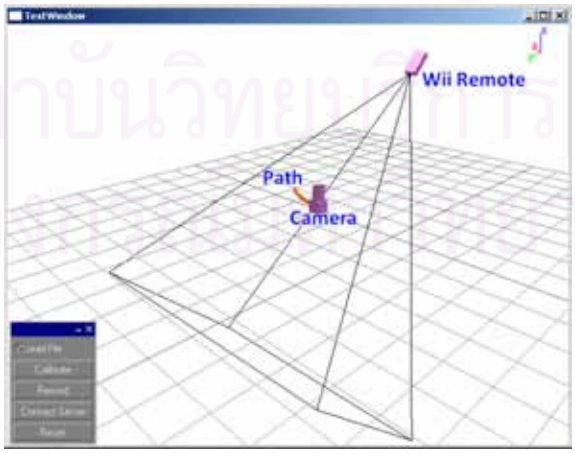
(ก)

(ข)



(ค)

รูปที่ 6.20: (ก) หุ่นยนต์ที่ติดตั้งกล้องวิดีโอแบบออมนิและแหล่งกำเนิดแสงอินฟาเรด (ข) อุปกรณ์ Wii Remote ซึ่งติดตั้งเป็นกล้องมุมสูง (ค) จุดบนพื้นสำหรับการ Calibrate



รูปที่ 6.21: โปรแกรมสำหรับการระบุตำแหน่งกล้องในระนาบสองมิติด้วยอุปกรณ์ Wii Remote

6.6.1.1 ความคลาดเคลื่อนของการระบุตำแหน่งด้วยอุปกรณ์ Wii Remote

กล้องอินฟาเรดในอุปกรณ์ Wii Remote นั้นรับภาพอินฟาเรดได้ที่ความละเอียด 1024x768 พิกเซล มี ความยาวโฟกัส (focal lengths) 1280 พิกเซลต่อยูนิต ไม่มี distortion ของเลนส์ ในที่นี้จะสมมุติว่าพื้นที่ใช้ในการ calibrate นั้นคือเป็นระนาบในการเคลื่อนที่ของกล้องและความสูงจากระนาบจะเป็นแกน z การหาตำแหน่งอินฟาเรดในระนาบสองมิตินั้นสามารถหาได้จาก

$$f(u,v) = \begin{bmatrix} x \\ y \\ h \end{bmatrix} = \begin{bmatrix} tx \\ ty \\ tz \end{bmatrix} + (\frac{h - tz}{rz}) \cdot \begin{bmatrix} rx \\ ry \\ rz \end{bmatrix} \tag{6.1}$$

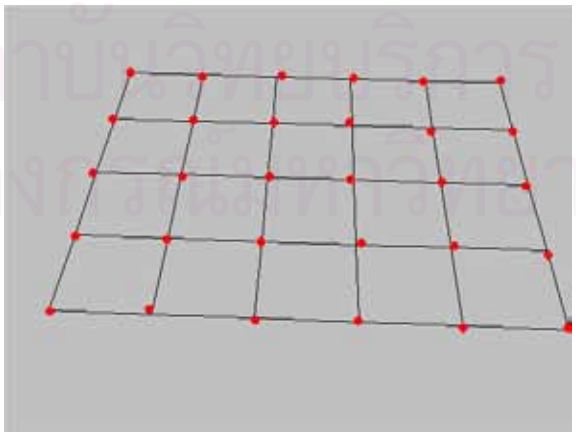
$$\begin{bmatrix} rx \\ ry \\ rz \end{bmatrix} = R \times \begin{bmatrix} (u - cx)/fx \\ (v - cy)/fy \\ -1 \end{bmatrix} \tag{6.2}$$

ซึ่ง (x, y) จะเป็นพิกัดบนระนาบ, h คือความสูงของจุดอินฟาเรดจากพื้น, $[\ tx \quad ty \quad tz\ ]^T$ เป็นตำแหน่งของอุปกรณ์ Wii Remoteในสามมิติ, R เป็น Matrix การหมุนของอุปกรณ์ Wii Remote, (cx, cy, fx, fy) เป็น intrinsic parameters ของกล้องอินฟาเรดในอุปกรณ์ Wii Remote, และ (u, v) เป็นตำแหน่งของจุดอินฟาเรดบนภาพอินฟาเรด

สำหรับ intrinsic parameters ของกล้องอินฟาเรดในอุปกรณ์ Wii Remote นั้นจะเป็นค่าคงที่โดยที่ cx เท่ากับ 512, cy เท่ากับ 384, fx และ fy เท่ากับ 1280 ส่วน translation ($[\ tx \quad ty \quad tz\ ]^T$) และ rotation ($R$) หาได้จากกการ calibrate ตำแหน่ง ดังนั้นความคลาดเคลื่อนของพิกัดบนระนาบสองมิติจะหาได้จาก

$$E = FWF^T \tag{6.3}$$

โดยที่ E เป็นความแปรปรวนร่วมของการวัดตำแหน่งบนระนาบ และ W เป็นความแปรปรวนร่วมของการวัดตำแหน่งบนภาพ ส่วน เป็น Jacobian Matrix ของ f เทียบกับตำแหน่งบนภาพ $[uv]^T$ ซึ่งความแปรปรวนร่วมของการวัดตำแหน่งบนภาพ หาได้จากค่าเฉลี่ยความคลาดเคลื่อนของการ calibrate



รูปที่ 6.22: จุดอินฟาเรดที่ตรวจจับได้จากอุปกรณ์ Wii Remote

การหาค่าเฉลี่ยความคลาดเคลื่อนของการ calibrate จะหาจากค่าความต่างระหว่าง ค่าโปรเจคตำแหน่งจุด calibrate ในสามมิติลงบนภาพอินฟาเรด กับตำแหน่งจุดอินฟาเรดที่วัดได้จริงบนภาพ ซึ่งจาก

รูปที่ 6.22 วงกลมสีแดงแสดง ตำแหน่งจุดอินฟาเรดที่วัดได้จริงส่วนตารางจะแสดงตำแหน่งจุด calibrate ที่ได้จากการประมาณด้วยการโปรเจคตำแหน่งในสามมิติลงบนภาพ และความแปรปรวนร่วมของการวัดตำแหน่งบนภาพ (W) จะหาได้เป็น

$$W = \begin{bmatrix} \sigma_u^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix} \tag{6.4}$$

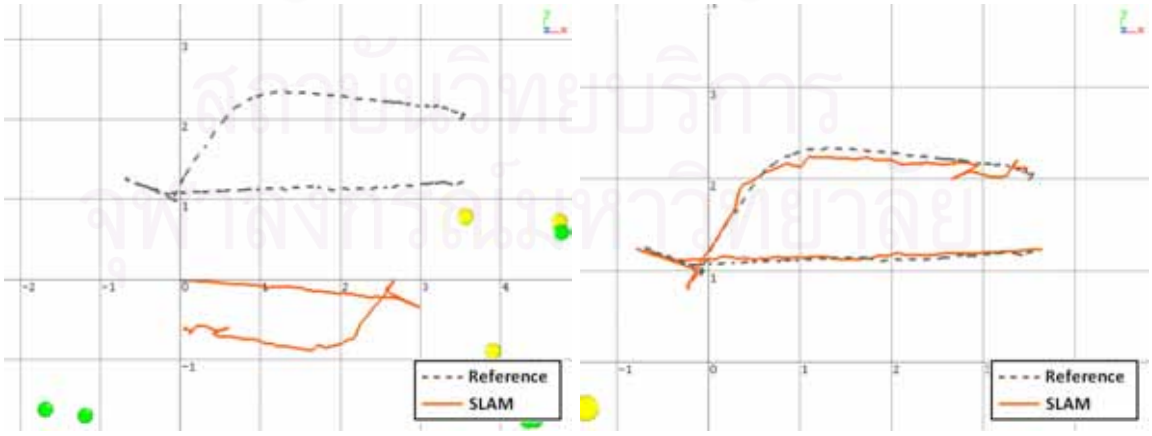
โดยที่ σ_u และ σ_v จะเป็นความคลาดเคลื่อนเฉลี่ยหรือค่าเบี่ยงเบนมาตรฐาน ในแกน x และ y ในภาพอินฟาเรด ซึ่งในการทดลองนี้จะหาค่าความแปรปรวนร่วมของการวัดตำแหน่งบนภาพ (W) ได้เป็น

$$W = \begin{bmatrix} 14.632 & 0 \\ 0 & 12.233 \end{bmatrix} \tag{6.5}$$

โดยที่ความแปรปรวนร่วมของการวัดตำแหน่งบนระนาบ (E) นั้นจะเปลี่ยนแปลงไปเรื่อย ๆ ตามตำแหน่งจุดอินฟาเรด แต่ค่าความแปรปรวนร่วมของตำแหน่งบนระนาบนั้น สามารถรับประกันได้ว่าจะมีความคลาดเคลื่อนคงที่ที่ตำแหน่งเดิม ไม่ว่ากล้องออมนิจะเคลื่อนที่ผ่านไปมานานเท่าใด

6.6.1.2 ผลลัพธ์การวัดความคลาดเคลื่อนของการระบุตำแหน่ง

ผลลัพธ์การระบุตำแหน่งที่ได้จากอัลกอริทึมที่ได้นำเสนอนั้น จะได้ออกมาเป็นเส้นทางการเคลื่อนที่ของกล้องในสามมิติ ซึ่ง scale ของเส้นทางการเคลื่อนที่ของกล้องนั้นอาจจะไม่ตรงกับ scale ของเส้นทางจริงในหน่วยเมตรก็เป็นได้ ทั้งนี้เป็นเพราะในการระบุตำแหน่งและการสร้างแผนที่นั้นจะสร้างจากข้อมูลภาพจากกล้องออมนิเท่านั้น ทำให้หน่วยวัดที่ใช้ในการระบุตำแหน่งและสร้างแผนที่นั้นเป็นหน่วยที่กำหนดขึ้นเอง ไม่อาจสอดคล้องกับหน่วยเมตรได้ ดังนั้นในการวัดเทียบความถูกต้องของการระบุตำแหน่งนั้นจะใช้การ ปรับ scale และโคออดิเนต ของเส้นทางการเคลื่อนที่ที่สร้างได้จาก SLAM ให้ตรงกับเส้นทางการเคลื่อนที่ที่หาได้จากอุปกรณ์ Wii Remote เสียก่อนแล้วจึงวัดเทียบความถูกต้องของการระบุตำแหน่ง ซึ่งในการปรับ scale และโคออดิเนต จะใช้การปรับแก้โดยมนุษย์เอง โดยจะเลือกตำแหน่งที่มีความสอดคล้องของเส้นทางการเคลื่อนที่สูงสุดดังแสดงได้ในรูปที่ 6.23ข



(ก) เส้นทางการเคลื่อนที่ก่อนการปรับ scale และโคออดิเนต (ข) เส้นทางการเคลื่อนที่หลังการปรับ scale และโคออดิเนต

รูปที่ 6.23: เส้นทางการเคลื่อนที่ที่ตรวจวัดได้จาก Wii Remote และ SLAM

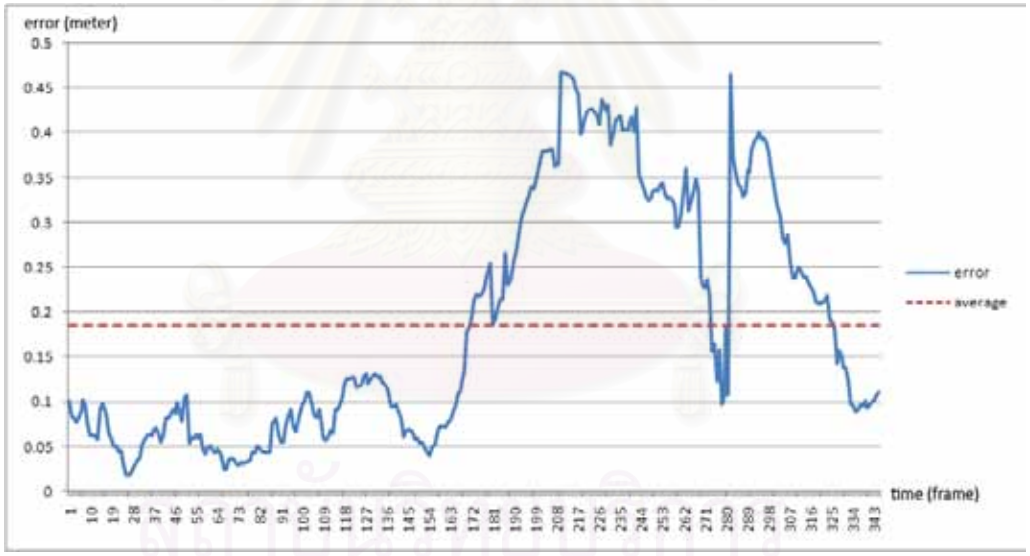
ความแปรปรวนร่วมความคลาดเคลื่อนการระบุตำแหน่ง (P) ณ แต่ละจุดบนเส้นทางการเคลื่อนที่หาได้จาก

$$P = Q + E \tag{6.6}$$

โดยที่ E เป็นความแปรปรวนร่วมของการวัดตำแหน่งบนระนาบด้วยอุปกรณ์ Wii Remote ซึ่งจะใช้เพียงความแปรปรวนร่วมในระนาบเท่านั้น (ไม่ใช้ความแปรปรวนในแกนความสูง) ส่วน Q เป็นความแปรปรวนร่วมค่าความต่างของตำแหน่งของกล้อง โดยจะหาได้จาก

$$Q = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \tag{6.7}$$

โดยที่ σ_x และ σ_y ความต่างของเส้นทางการเคลื่อนที่ที่สร้างได้จาก SLAM กับเส้นทางการเคลื่อนที่ที่หาได้จากอุปกรณ์ Wii Remote ในแกน x และ y โดยความผิดพลาดมากสุดของการระบุตำแหน่งจากอัลกอริทึมที่ได้นำเสนอ จะหาได้จาก รากที่สองของค่า Eigenvalues ที่มากที่สุดของ P ซึ่งค่าความผิดพลาดของการระบุตำแหน่งตลอดเส้นทางการเคลื่อนที่ของกล้องแสดงได้ดังรูปที่ 6.24 และค่าความผิดพลาดของการระบุตำแหน่งเฉลี่ย แสดงได้ในตารางที่ 6.1 โดยจากตารางที่ 6.1 สามารถพูดได้ว่าความคลาดเคลื่อนของการระบุตำแหน่งนั้นมีค่าประมาณ 0.1596482 เมตร



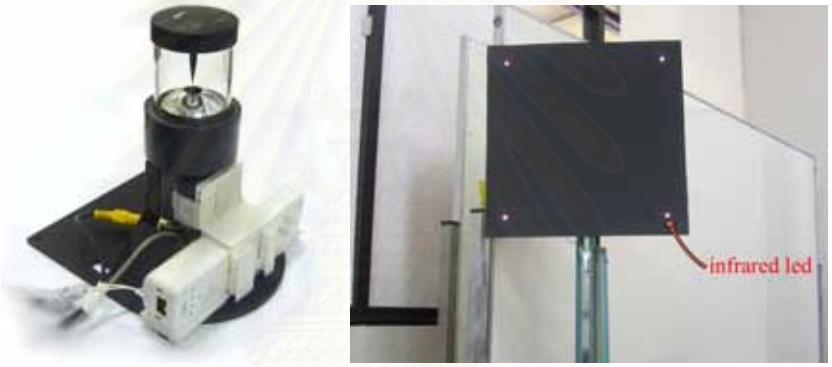
รูปที่ 6.24: กราฟแสดงความคลาดเคลื่อนการระบุตำแหน่งในสองมิติ

ตารางที่ 6.1: ความคลาดเคลื่อนของการระบุตำแหน่งด้วยอัลกอริทึมที่นำเสนอ

การทดลองที่	Error (เมตร)
1	0.188296
2	0.212112
3	0.0785367
เฉลี่ย	0.1596482

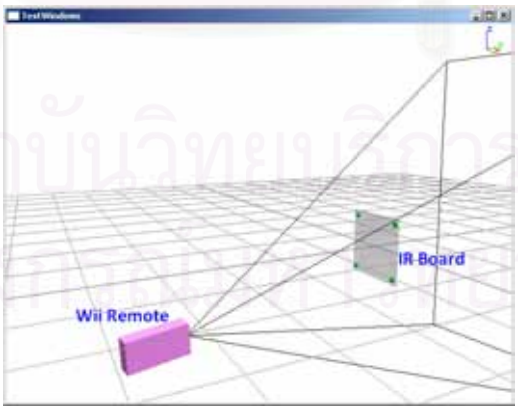
6.6.2 การวัดผลความถูกต้องของการระบุตำแหน่งในสามมิติ

การวัดผลการระบุตำแหน่งในสามมิตินั้น จะใช้อุปกรณ์ Wii Remote ในการวัดผลเช่นกัน แต่วิธีการในการระบุตำแหน่งด้วยอุปกรณ์ Wii Remote จะแตกต่างออกไป ในการระบุตำแหน่งของกล้องออมนิด้วยอุปกรณ์ Wii Remote นั้นในที่นี้จะติดตั้งอุปกรณ์ Wii Remote คู่ไปกับกล้องวิดีโอแบบออมนิดังรูปที่ 6.25ก โดยอุปกรณ์ Wii Remote นั้นจะใช้ตรวจจับแผ่นกระดานอ้างอิงแผ่นหนึ่งซึ่งติดตั้งแหล่งกำเนิดแสงอินฟาเรดไว้จำนวน 4 จุด โดยที่แต่ละจุดนั้นจะทราบระยะห่างระหว่างกันแน่นอนดังรูปที่ 6.25ข เมื่ออุปกรณ์ Wii Remote ตรวจจับจุดอินฟาเรดได้อย่างน้อย 3 จุดก็จะสามารถคำนวนหาตำแหน่งของอุปกรณ์ Wii Remote ในสามมิติเทียบกับตำแหน่งของแผ่นกระดานอ้างอิงได้ ซึ่งเนื่องจากว่าอุปกรณ์ Wii Remote ถูกติดตั้งคู่ไปกับกล้องวิดีโอแบบออมนิ ดังนั้นเมื่อทราบตำแหน่งของอุปกรณ์ Wii Remote ก็จะทราบตำแหน่งของกล้องออมนิด้วย ดังแสดงได้ในรูปที่ 6.26 ซึ่งการระบุตำแหน่งด้วยอุปกรณ์ Wii Remote วิธีนี้จะทำให้ทราบตำแหน่งของกล้องออมนิในสามมิติได้ แต่มีข้อแม้ว่า อุปกรณ์ Wii Remote จะต้องหันหน้าเข้าหาแผ่นกระดานอ้างอิงเสมอ และอุปกรณ์ Wii Remote จะเคลื่อนที่ห่างจากแผ่นกระดานอ้างอิง มากไม่ได้ เพราะจะทำให้ผลการระบุตำแหน่งด้วยอุปกรณ์ Wii Remote นั้นคลาดเคลื่อนมากขึ้น



(ก) (ข)

รูปที่ 6.25: (ก) อุปกรณ์ Wii Remote ติดตั้งกับกล้องวิดีโอแบบออมนิ (ข) แผ่นกระดานอ้างอิง ติดตั้งแหล่งกำเนิดแสงอินฟาเรด

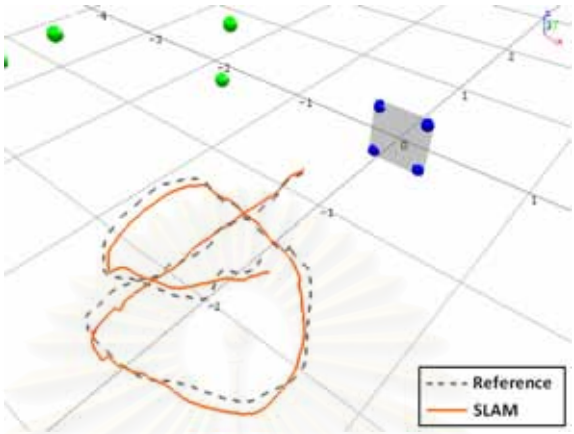


รูปที่ 6.26: โปรแกรมสำหรับการระบุตำแหน่งกล้องในระนาบสามมิติด้วยอุปกรณ์ Wii Remote

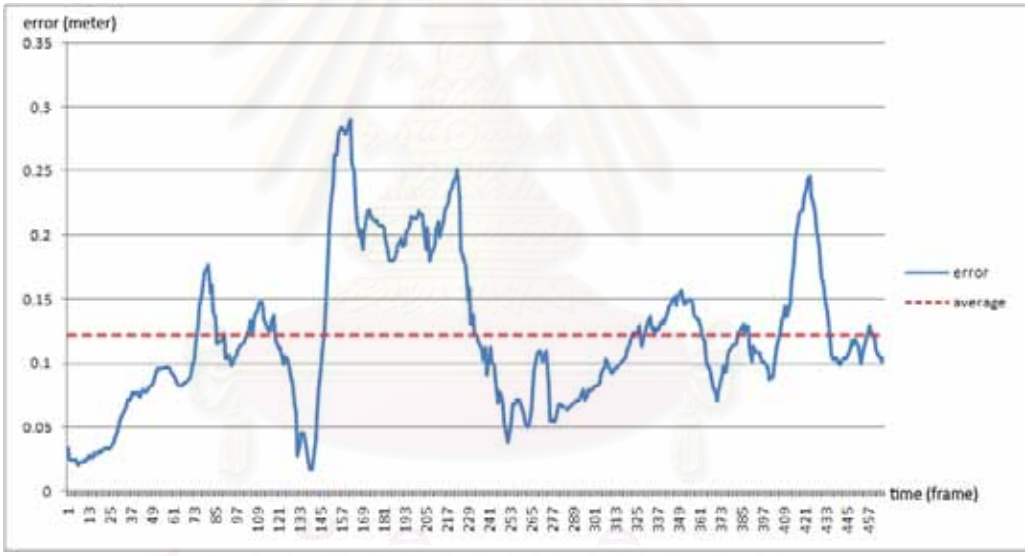
6.6.2.1 ผลลัพธ์การวัดความคลาดเคลื่อนของการระบุตำแหน่ง

สำหรับความคลาดเคลื่อนของการระบุตำแหน่งด้วยอุปกรณ์ Wii Remote นั้นมีวิธีการคำนวนค่อนข้างยุ่งยากดังนั้นจะสมมุติว่ามีความคลาดเคลื่อนน้อยจนไม่พิจารณา และการวัดความคลาดเคลื่อนของการระบุตำแหน่งในสามมิติ จะวัดเพียงความต่างของตำแหน่งที่ระบุได้ด้วยอุปกรณ์ Wii Remote และ

ตำแหน่งที่ระบุได้ด้วยอัลกอริทึมที่ได้นำเสนอเท่านั้น โดยจะใช้ Euclidean distance ในการวัดความต่าง และจะมีการปรับแก้ scale และโคออดิเนตโดยมนุษย์ เช่นเดียวกับการวัดความคลาดเคลื่อนของการระบุตำแหน่งในระนาบสองมิติ ซึ่งผลการวัดเทียบแสดงได้ดังรูปที่ 6.27 และความคลาดเคลื่อนตลอดเส้นทางแสดงได้ดังรูปที่ 6.28



รูปที่ 6.27: เส้นทางการเคลื่อนที่ในสามมิติที่ตรวจวัดได้จาก Wii Remote และ SLAM



รูปที่ 6.28: กราฟแสดงความคลาดเคลื่อนการระบุตำแหน่งในสามมิติ

ค่าความคลาดเคลื่อนการเคลื่อนที่เฉลี่ยตลอดเส้นทาง แสดงได้ในตารางที่ 6.2

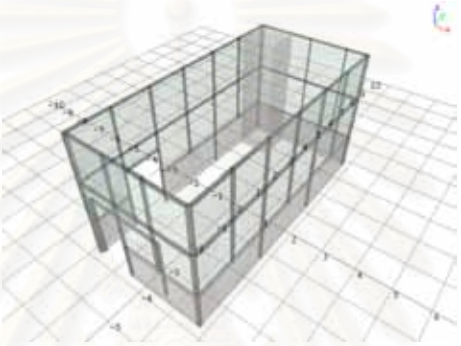
ตารางที่ 6.2: ความคลาดเคลื่อนของการระบุตำแหน่งด้วยอัลกอริทึมที่นำเสนอ

การทดลองที่	Error (เมตร)
1	0.121575
2	0.108459
3	0.103404
เฉลี่ย	0.111146

6.7 การวัดผลความถูกต้องของการสร้างแผนที่

การวัดผลความถูกต้องของการสร้างแผนที่ มีจุดประสงค์เพื่อการตรวจสอบความแม่นยำของแผนที่สิ่งแวดล้อมที่สร้างได้จากวิธีการสร้างแผนที่ด้วยกล้องออมนิที่ได้นำเสนอ ซึ่งวิธีการการวัดผลการสร้างแผนที่นั้นจะใช้การวัดเทียบระหว่างแผนที่ที่สร้างได้จากวิธีการที่ได้นำเสนอ เทียบกับสิ่งแวดล้อมจริงที่รู้รูปร่างลักษณะของสิ่งแวดล้อมที่แน่นอนอยู่แล้ว ซึ่งสิ่งแวดล้อมที่จะใช้ในการวัดเทียบในที่นี้จะเป็นสถานที่ภายในห้องประชุม 20-01 ชั้น 20 ตึกวิศวกรรมศาสตร์ 4 ซึ่งเนื่องจากรูปแบบโครงสร้างห้องประชุมนี้ มีแบบแผนที่ค่อนข้างเป็นระเบียบ ทำให้สามารถวัดผลได้ง่าย

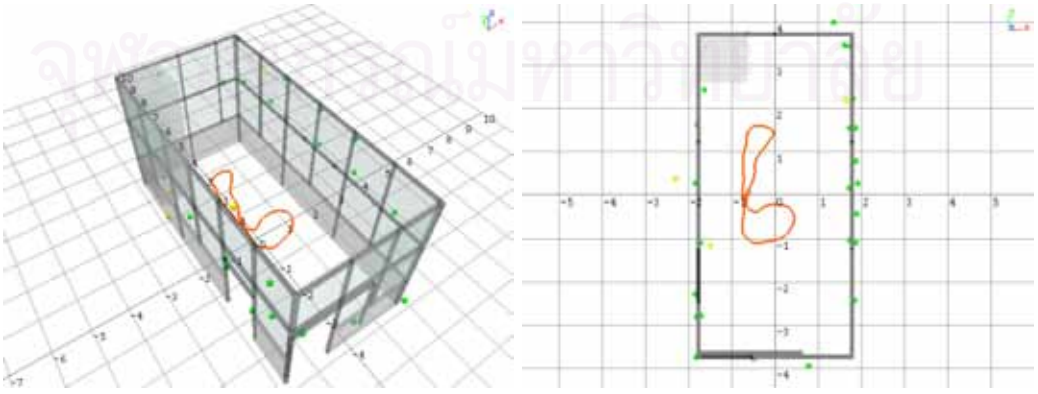
การวัดความถูกต้องของการสร้างแผนที่ จำเป็นจะต้องสร้างแผนที่ของสิ่งแวดล้อมที่ต้องการใช้วัดเทียบขึ้นมาตามขนาดที่ถูกต้องเสียก่อน ซึ่งโมเดลของห้องประชุม 20-01ได้ถูกสร้างขึ้นมาตามขนาดที่ถูกต้อง แสดงได้ดังรูปที่ 6.29



รูปที่ 6.29: โมเดลสามมิติของห้องประชุม 20-01

6.7.1 ผลลัพธ์การวัดความคลาดเคลื่อนของการสร้างแผนที่

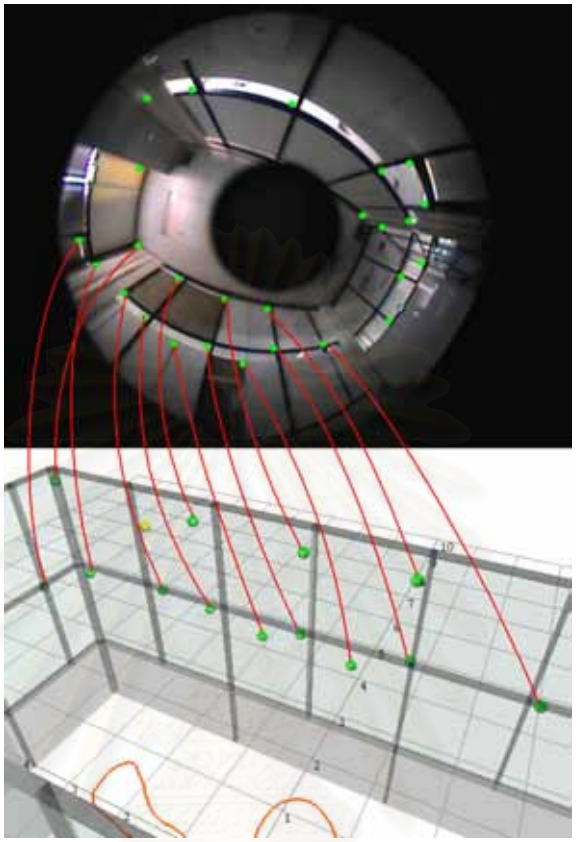
ผลลัพธ์การระบุสร้างแผนที่ที่ได้จากอัลกอริทึมที่ได้นำเสนอนั้น จะได้เป็นจุดสังเกตของสิ่งแวดล้อมในสามมิติ ซึ่ง scale ของแผนที่ที่สร้างได้นั้นนั้นอาจจะไม่ตรงกับ scale ของสิ่งแวดล้อมจริงในหน่วยเมตรเฉกเช่นเดียวกับผลการระบุตำแหน่ง ดังนั้นในการวัดเทียบความถูกต้องของการสร้างแผนที่นั้นจะใช้การปรับ scale และโคออดิเนต ของแผนที่ที่สร้างได้จาก SLAM ให้ตรงกับโมเดลจำลองสิ่งแวดล้อมเสียก่อนแล้วจึงวัดเทียบความถูกต้องของการสร้างแผนที่ ซึ่งในการปรับ scale และโคออดิเนต จะใช้การปรับแก้โดยมนุษย์เอง โดยจะเลือกตำแหน่งที่มีความสอดคล้องของแผนที่สูงสุดดังแสดงได้ในรูปที่ 6.30



(ก) (ข)

รูปที่ 6.30: ความคลาดเคลื่อนของแผนที่ที่สร้างได้เทียบกับโมเดลสิ่งแวดล้อมที่จำลองขึ้น

สำหรับรูปที่ 6.30ก และ 6.30ข ได้แสดงผลลัพธ์ของการสร้างแผนที่เทียบกับโมเดลสิ่งแวดล้อมที่รู้ขนาดแน่ชัด ซึ่งในรูปที่ 6.30ก จะแสดงผลลัพธ์การสร้างแผนที่มุมมองในสามมิติ ส่วนในรูปที่ 6.30ข นั้นจะเป็นมุมมองด้านบน ซึ่งสำหรับมุมมองระบะใกล้นั้นจะแสดงได้ดังรูปที่ 6.31



รูปที่ 6.31: เปรียบเทียบตำแหน่ง feature ที่ตรวจวัดได้จากภาพ กับตำแหน่งของจุดสังเกตที่ประมาณได้จากอัลกอริทึม

รูปที่ 6.31 นั้นแสดง feature ที่ตรวจวัดได้จากภาพเทียบกับตำแหน่งของจุดสังเกตที่ประมาณได้จากอัลกอริทึมที่ได้นำเสนอในโมเดลสามมิติของสิ่งแวดล้อมที่รู้ขนาดแน่ชัด และสำหรับการวัดความคลาดเคลื่อนนั้นจะใช้การหาระยะทางน้อยสุดระหว่างจุดสังเกตกับโมเดลสิ่งแวดล้อมที่จำลองขึ้นมาตามขนาดจริง ซึ่งค่าความคลาดเคลื่อนของจุดสังเกตเฉลี่ยทั้งแผนที่ แสดงได้ในตารางที่ 6.3

ตารางที่ 6.3: ความคลาดเคลื่อนของการสร้างแผนที่ด้วยอัลกอริทึมที่นำเสนอ

การทดลองที่	Error (เมตร)
1	0.124627
2	0.0939502
3	0.113272
เฉลี่ย	0.1106164

บทที่ 7

สรุปการวิจัยและแนวทางการวิจัยในขั้นถัดไป

7.1 สรุปการวิจัย

จากการผลทดลองข้างต้น ได้แสดงผลการระบุตำแหน่งพร้อมกับการสร้างแผนที่ในสามมิติด้วยกล้องวิดีโอแบบออมนิ โดยในการทดลองนั้นก็ได้ทำการทดลองในสิ่งแวดล้อมที่หลากหลายรูปแบบไม่ว่าจะเป็นภายในห้องวิจัย, ห้องประชุม, โถงทางเดิน, ทางขึ้นลงบันได, ระเบียงทางเดิน, บริเวณห้องโถง หลังจากนั้นก็ได้ทำการวัดผลการระบุตำแหน่งโดยใช้อุปกรณ์ Wii Remote และ วัดผลการสร้างแผนที่ด้วยการวัดเทียบแผนที่ที่สร้างได้กับสิ่งแวดล้อมจริง ซึ่งจากผลการทดลองที่ได้นั้นจะเห็นได้ว่าความแม่นยำของการระบุตำแหน่งและแผนที่สร้างได้จะขึ้นอยู่กับลักษณะของสิ่งแวดล้อมโดยตรง ซึ่งลักษณะของสิ่งแวดล้อมที่ส่งผลกระทบต่อความแม่นยำของการระบุตำแหน่งและสร้างแผนที่ จะมีดังนี้

1. ลักษณะของสิ่งแวดล้อมที่มีความฉูดฉาดมากหรือน้อยจนเกินไป

สำหรับสิ่งแวดล้อมที่มีความฉูดฉาด จะทำให้การปรักแก้ตำแหน่งของ feature เป็นไปได้ยากเพราะสภาพแวดล้อมที่มีความฉูดฉาดทำให้มี feature ที่คล้ายคลึงกันได้ง่าย ส่วนสำหรับสิ่งแวดล้อมที่มีความฉูดฉาดน้อยนั้น อาจทำให้หา feature สำหรับสิ่งแวดล้อมนั้นไม่ได้หรือหาได้น้อย ซึ่งถ้าหากมี feature ในภาพน้อยจนเกินไปย่อมส่งผลกระทบต่อการปรับแก้ตำแหน่งจุดสังเกตและแผนที่ใน SLAM

2. ลักษณะของสิ่งแวดล้อมที่มีความซ้ำซ้อนสูง

ลักษณะของสิ่งแวดล้อมที่มีความซ้ำซ้อนสูง ยกตัวอย่างเช่น ต้นไม้, ตะแกรงช่องระบายลม, บานเกล็ด เป็นต้น สิ่งแวดล้อมที่มีความซ้ำซ้อนเหล่านี้จะส่งผลกระทบต่อการติดตาม และ ปรับแก้ตำแหน่งของจุดสังเกต เพราะอัลกอริทึมอาจจะเกิดความไขว้เขวได้ง่ายว่าตำแหน่งในเป็นจุดสังเกตจริง ทางแก้ของการติดตาม และ ปรับแก้ตำแหน่งของจุดสังเกตในสิ่งแวดล้อมที่มีความซ้ำซ้อนสูง นั้นก็คือการใช้ Optical Flow และ Template Matching แบบ pyramids ซึ่งจะช่วยการมองภาพรวมได้มากขึ้น แต่ pyramids นั้นจะใช้ระดับชั้นสูงมากไม่ได้ เพราะทั้งอัลกอริทึม Optical Flow และ Template Matching ได้ประมาณว่าภาพที่ใช้ในอัลกอริทึมมีมุมมองคล้ายกล้องแบบ perspective ทั่วไป

3. สิ่งแวดล้อมที่มีความซับซ้อนมาก

สำหรับสิ่งแวดล้อมที่มีความซับซ้อนมาก ย่อมส่งผลกระทบต่อการตรวจหา การติดตาม และการปรับแก้ตำแหน่งของ feature โดยตรงเพราะว่าในสิ่งแวดล้อมที่มีความซับซ้อนสูง จะมีการบดบังกันของสิ่งแวดล้อมเป็นจำนวนมาก ซึ่งการบดบังกันของสิ่งแวดล้อมนอกจากจะทำให้ติดตามการเคลื่อนที่ของ feature ได้ไม่ต่อเนื่องแล้ว ยังทำให้เกิด feature ปลอมได้อีกด้วย ซึ่งสำหรับ feature ปลอมนั้นจะเป็นคำใช้เรียกบริเวณที่ไม่ใช่จุดสังเกตแต่ถูกตรวจพบว่าเป็น feature ยกตัวอย่างเช่นในรูปที่ 7.1 จุดที่เกิดจากการตัดกันระหว่างเสากับกันสาด ด้านหลังนั้นจริงแล้วไม่ใช่จุดสังเกตที่มีลักษณะเด่นอะไร แต่เมื่อมาปรากฏบนภาพกลับตรวจพบเป็น feature

สำหรับ feature ปลอมเหล่านั้นอัลกอริทึมที่งานวิจัยนี้ได้นำเสนอ สามารถตรวจหาและกำจัด feature ปลอมเหล่านี้ได้ ซึ่งวิธีในการตรวจหานั้น ถ้าหากกล้องมีการเคลื่อนที่ไปเรื่อย ๆ กันสาดกับเสา ก็จะเลิก

รูปที่ 7.1: ภาพ feature ปลอม

ตัดกันในที่สุดซึ่งจะส่งผลให้การปรับแก้ตำแหน่งของ feature ล้มเหลว อัลกอริทึมจึงจะสามารถรับรู้ได้ว่า feature นั้นไม่ได้แสดงถึงจุดสังเกตจริง ๆ ก็จะลบ feature นั้นทิ้งไป แต่ถึงแม้ว่าอัลกอริทึมที่ได้นำเสนอจะสามารถตรวจหาและกำจัด feature ปลอมได้ แต่ก่อนที่ feature เหล่านั้นจะถูกลบ feature เหล่านั้นก็ได้สร้างผลกระทบต่อการระบุตำแหน่งและสร้างแผนที่ไปพอสมควร และถ้าหากมี feature ประเภทนี้มากเกินไปก็อาจจะทำให้การระบุตำแหน่งและการสร้างแผนที่ล้มเหลวได้

feature ที่มีความคลาดเคลื่อนนั้น เมื่อนำไปใช้เป็นข้อมูลการวัดสำหรับการระบุตำแหน่งและสร้างแผนที่ ย่อมจะส่งผลต่อความแม่นยำ ซึ่งถ้าหาก feature ที่มีความคลาดเคลื่อนเหล่านั้นมีจำนวนน้อยเมื่อเทียบกับจำนวน feature ทั้งหมด การระบุตำแหน่งและสร้างแผนที่ย่อมไม่คลาดเคลื่อนไปมาก แต่ถ้าหากว่า feature ที่ผิดพลาดมีมากจนเกินไปก็อาจจะทำให้การระบุตำแหน่งและสร้างแผนที่ล้มเหลวได้

ความคลาดเคลื่อนของการระบุตำแหน่งและสร้างแผนที่นั้นแสดงได้ดังในส่วนการวัดผลความถูกต้องของการระบุตำแหน่งและสร้างแผนที่ ซึ่งผลลัพธ์ของการวัดผลนั้นแสดงให้เห็นว่า ถึงแม้เส้นทางการเคลื่อนที่ของกล้อง และแผนที่ที่สร้างได้นั้น จะไม่ซ้อนทับพอดีกับการเคลื่อนที่จริงและแผนที่จริง แต่ผลลัพธ์ที่ได้ก็แสดงให้เห็นถึงแนวเส้นทางการเคลื่อนที่ของกล้องและโครงสร้างของสิ่งแวดล้อมอย่างคร่าว ซึ่งเมื่อเปรียบเทียบกับการระบุตำแหน่งและสร้างแผนที่ของมนุษย์แล้ว ก็จะเห็นได้ว่ามนุษย์เองก็ไม่สามารถระบุตำแหน่งและสร้างแผนที่ได้ละเอียดมากนัก แต่มนุษย์สามารถรู้โครงสร้างของสิ่งแวดล้อม และระบุตำแหน่งของตนแบบประมาณได้ ซึ่งข้อมูลเท่านี้ก็เพียงพอสำหรับการประมวลผลสำหรับการทำกิจกรรมต่าง ๆ ของหุ่นยนต์แล้ว

7.2 แนวทางการวิจัยในขั้นถัดไป

สำหรับในงานวิจัยนี้ได้นำเสนอวิธีการในการระบุตำแหน่งและสร้างแผนที่ในสามมิติด้วยกล้องวิดีโอแบบออมนิซึ่ง กล้องออมนิที่ใช้ในงานวิจัยนี้สามารถเคลื่อนที่ได้อย่างอิสระโดยไม่ต้องรู้โมเดลการเคลื่อนที่ของกล้องก่อน จึงทำให้ งานวิจัยนี้สามารถนำไปใช้ในการระบุตำแหน่งและสร้างแผนที่สำหรับหุ่นยนต์ได้อย่างหลากหลายมาก แต่ปัญหาของงานวิจัยนี้ก็คล้ายคลึงกับงานวิจัยด้าน SLAM อื่น ๆ คืออัลกอริทึมสามารถทำงานได้ในบริเวณจำกัด เพราะการระบุตำแหน่งและสร้างแผนที่ในสิ่งแวดล้อมที่มีขนาดใหญ่มากย่อมต้องส่งผลกระทบต่อประสิทธิภาพการคำนวน ทำให้ SLAM ไม่สามารถเพิ่มขนาดแผนที่ให้ใหญ่ขึ้นเรื่อย ๆ ได้ แตกต่างจากมนุษย์ ซึ่งแท้จริงแล้วมนุษย์ก็ไม่สามารถสร้างแผนที่ที่มีขนาดใหญ่มากได้ แต่มนุษย์เราจะใช้การสร้างแผนที่เป็นแบบลำดับชั้นต้นไม้ หมายความว่ามนุษย์จะมองภาพรวมก่อนจากนั้นจึงลงรายละเอียดเฉพาะในแผนที่ที่สนใจทำให้มนุษย์สามารถสร้างความสัมพันธ์ของแผนที่และระบุตำแหน่งของตนเองได้ว่าอยู่ในตำแหน่งไหนในโลก ดังนั้นการวิจัยที่น่าสนใจในขั้นถัดไปก็คือการทำให้หุ่นยนต์สามารถมองแผนที่แบบภาพรวมได้เช่นเดียวกับมนุษย์

รายการอ้างอิง

Andreasson, H., Treptow, A., and Duckett, T. 2005. Localization for mobile robots using panoramic vision, local features and particle filter. In International Conference on Robotics and Automation.

Asmar, D. C., Zelek, J. S., and Abdallah, S. M. 2006. Tree trunks as landmarks for outdoor vision slam. In Conference on Computer Vision and Pattern Recognition Workshop.

Bailey, T. 2002. Mobile Robot Localisation and Mapping in Extensive Outdoor Environments. PhD thesis, Australian Centre for Field Robotics Department of Aerospace, Mechanical and Mechatronic Engineering The University of Sydney.

Baker, S., and Nayar, S. K. 1999. A theory of single-viewpoint catadioptric image formation. International Journal of Computer Vision 35:175-196.

Bjorck, A. 1996. Numerical Methods for Least Squares Problems. SIAM.

Bosse, M., Leonard, J., and Teller, S. 2002. Large-scale cml using a network of multiple local maps. In Workshop Notes of the ICRA Workshop on Concurrent Mapping and Localization for Autonomous Mobile Robots.

Bouguet, J. Y. 2000. Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. Technical report, Intel Corporation, Microprocessor Research Labs.

Bruckstein, A. M., and Richardson, T. J. 2000. Ominview cameras with curved surface mirrors. In Proceedings of the IEEE Workshop on Omnidirectional Vision.

Chatila, R., and Moutarlier, P. 1989. Stochastic multisensory data fusion for mobile robot location and environment modeling. In In 5th Int. Symposium on Robotics Research.

Cummins, M., and Newman, P. 2007. Probabilistic appearance based navigation and loop closing. In International Conference on Robotics and Automation.

Cummins, M., and Newman, P. 2008. Accelerated appearance-only slam. In International Conference on Robotics and Automation.

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.

Diosi, A., and Kleeman, L. 2005. Laser scan matching in polar coordinates with application to slam. In International Conference on Intelligent Robots and Systems.

Durrant-Whyte, H. and Bailey, T. 2006. Simultaneous localization and mapping:part i,ii. IEEE Robotics & Automation Magazine June.

Eade, E., and Drummond, T. 2006. Scalable monocular slam. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

Fox, D., Hexmoor, H., and Mataric, M. 1998. A probabilistic approach to concurrent mapping and localization for mobile robots. Machine Learning and Autonomous Robots p. 29.

Fu, S., Liu, H.-y., Gao, L.-f., and Gai, Y.-x. 2007. Slam for mobile robots using laser range finder and monocular vision. In International Conference on Mechatronics and Machine Vision in Practice.

Gemeiner, P., Ponweiser, W., Einramhof, P., and Vincze, M. 2007. Real-time slam with a high-speed cmos camera. In International Conference on Image Analysis and Processing.

Hahnel, D., Burgard, W., Fox, D., and Thrun, S. 2003. A highly efficient fastslam algorithm for generating cyclic maps of large-scale environments from raw laser range measurements. In International Conference on Intelligent Robots and Systems.

Han, C., Xiang, Z., Liu, J., and Wu, E. 2007. Stereo vision based slam in outdoor environments. In IEEE International Conference on Robotics and Biomimetics.

Harris, C., and Stephens, M. 1988. A combined corner and edge detector. In Proceedings of The Fourth Alvey Vision Conference.

Jeong, W. Y., and Lee, K. M. 2006. Visual slam with line and corner features. In International Conference on Intelligent Robots and Systems.

Julier, S. J., and Uhlmann, J. K. 1997. A new extension of the kalman filter to nonlinear systems. Int. Symp. Aerospace/Defense Sensing, Simul. and Controls 3.

Kalman, R. 1960. A new approach to linear filtering and prediction problems. Journal of Basic Engineering 82:35--45.

Kim, J.-H., and Chung, M. J. 2003. Slam with omni-directional stereo vision sensor. In International Conference on Intelligent Robots and Systems.

Kim, J., Yoon, K.-J., Kim, J.-S., and Kweon, I. 2006. Visual slam by single-camera catadioptric stereo. In SICE-ICASE International Joint Conference.

Leonard, J. 2008. A perception driven autonomous urban vehicle. Journal of Field Robotics .

LI, R., MA, F., XU, F., MATTHIES, L., OLSON, C., and XIONG, Y. 2000. Large scale mars mapping and rover localization. using descent and rover imagery. In Proceedings of ISPRS 19th Congress.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision .

Mahalanobis, P. C. 1936. On the generalised distance in statistics. In Proceedings of the National Institute of Sciences of India.

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. In Proc. of the Int. Conf. on Artificial Intelligence (IJCAI) p. 1151.

Motard, E., Raducanu, B., Cadenat, V., and Vitria, J. 2007. Incremental on-line topological map learning for a visual homing application. In International Conference on Robotics and Automation.

Moutarlier, P., and Chatila, R. 1989. An experimental system for incremental environment modeling by an autonomous mobile robot. In In 1st International Symposium on Experimental Robotics.

Murillo, A. C., Guerrero, J. J., and Sagues, C. 2006. Robot and landmark localization using scene planes and the 1d trifocal tensor. In International Conference on Intelligent Robots and Systems.

Pfingsthorn, M., Slamet, B., and Visser, A. 2008. A scalable hybrid multi-robot slam method for highly detailed maps. RoboCup 2007: Robot Soccer World Cup XI 5001:457--464.

Saedan, M., Lim, C. W., and Ang, M. H. 2007. Appearance-based slam with map loop closing using an omnidirectional camera. In Advanced Intelliget Mechatronics.

Scaramuzza, D., Martinelli, A., and Siegwart, R. 2006a. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In IEEE International Conference on Computer Vision Systems.

Scaramuzza, D., Siegwart, R., and Martinelli, A. 2006b. A toolbox for easily calibrating omnidirectional cameras. In International Conference on Intelligent Robots and Systems.

Scheding, S., Nebot, E. M., Stevens, M., and Durrant-Whyte, H. 1997. Experiments in autonomous underground guidance. In IEEE International Conference on Robotics and Automation.

Smith, R. C., and Cheeseman, P. 1986. On the representation and estimation of spatial uncertainty. The International Journal of Robotics Research 5:56.

Sunderhauf, N., Lange, S., and Protzel, P. 2007. Using the unscented kalman filter in mono-slam with inverse depth parametrization for autonomous airship control. In In Proc. of IEEE International Workshop on Safety Security and Rescue Robotics.

Tamimi, H., Andreasson, H., Treptow, A., Duckett, T., and Zell, A. 2005. Localization of mobile robots with omnidirectional vision using particle filter and iterative sift. In In Proceedings of the 2005 European Conference on Mobile Robots.

Thrun, S., et al. 2003. A system for volumetric robotic mapping of abandoned mines. In IEEE International Conference on Robotics and Automation.

Uhlmann, J., Lanzagorta, M., and Julier, S. 1999. The nasa mars rover: A testbed for evaluating applications of covariance intersection. In In Proceedings of the SPIE 13th Annual Symposium in Aerospace/Defence Sensing, Simulation and Controls.

Valgren, C., Lilienthal, A., and Duckett, T. 2006. Incremental topological mapping using omnidirectional vision. In International Conference on Intelligent Robots and Systems.

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 26:335.

Wang, X., and Zhang, H. 2007. A upf-ukf framework for slam. In International Conference on Robotics and Automation.

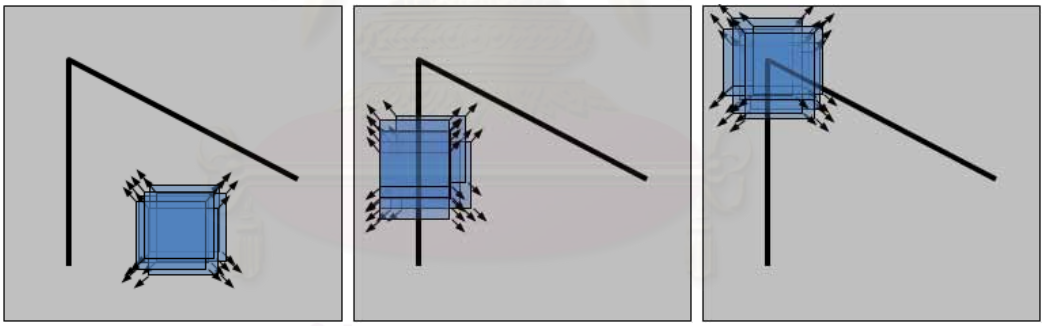
Williams, S., Dissanayake, G., and Durrant-Whyte, H. 2001. Towards terrain-aided navigation for underwater robotics. Advanced Robotics .

ภาคผนวก

ภาคผนวก ก

HARRIS CORNER DETECTOR

Harris Corner Detector เป็นวิธีในการตรวจหา feature แบบจุดที่เป็นที่นิยมวิธีหนึ่ง ซึ่ง Feature ที่ตรวจหาได้จาก Harris Corner Detector นั้นจะเป็น feature แบบจุดซึ่งบรรยายลักษณะสำคัญของวัตถุในสิ่งแวดล้อมจริง ข้อดีของ Harris Corner Detector คือ Harris Corner Detector สามารถตรวจหาตำแหน่ง feature ได้โดยไม่ขึ้นกับ การหมุน, การขยายขนาด, การเปลี่ยนแปลงสภาพแสง หรือ noise ของภาพ สำหรับแนวคิดของ Harris Corner Detector มีอยู่ว่าจุดที่จะเป็นตำแหน่งของ feature ที่ดีนั้น เมื่อทำการเลื่อนตำแหน่งไปเพียงเล็กน้อยจะมีความเปลี่ยนแปลงความเข้มของสีอย่างมหาศาล ดังแสดงดังรูปที่ ก.1 ซึ่งในรูปที่ ก.1ก กรอบสี่เหลี่ยมวางอยู่ในบริเวณราบเรียบ แสดงให้เห็นว่าเมื่อมีการเลื่อนกรอบสี่เหลี่ยมไปในทิศทางใด ๆ นั้นแทบไม่มีความเปลี่ยนแปลงของภาพในกรอบเลย ดังนั้น บริเวณในกรอบสี่เหลี่ยมย่อมไม่ใช่ feature ที่ดี สำหรับในรูปที่ ก.1ข กรอบสี่เหลี่ยมนั้นวางตัวอยู่ในแนวเส้นตรง เมื่อมีการเลื่อนกรอบสี่เหลี่ยมในแนวขึ้นลงตามเส้นจะเห็นได้ว่าภาพที่ปรากฏในกรอบไม่มีการเปลี่ยนแปลงมากนัก แต่เมื่อเลื่อนกรอบในแนวซ้ายขวาจะเกิดการเปลี่ยนแปลงของภาพในกรอบ แสดงว่าบริเวณในกรอบสี่เหลี่ยมพอจะเป็น feature ได้หากภาพมีการเปลี่ยนแปลงเฉพาะในแนวซ้ายขวาเท่านั้น แต่ก็ยังไม่ใช่ feature ที่ดีพอสำหรับทุกกรณี ส่วนในรูปที่ ก.1ค นั้นกรอบสี่เหลี่ยมวางตัวอยู่ในบริเวณมุม ดังนั้นเมื่อมีการเปลี่ยนแปลงตำแหน่งของกรอบสี่เหลี่ยมไปเพียงเล็กน้อย ย่อมต้องเกิดการเปลี่ยนแปลงของภาพในกรอบสี่เหลี่ยมเป็นอย่างมาก ดังนั้นจุดบริเวณตรงกลางกรอบสี่เหลี่ยมจึงถือว่าเป็น feature ที่ดี



(ก) flat region (ข) edge region (ค) corner region

รูปที่ ก.1: แนวคิดการทำงานของ Harris corner detector

การคำนวนหาการเปลี่ยนแปลงความเข้มของสีเมื่อมีการเลื่อนกรอบไปเพียงเล็กน้อย แสดงได้ดังนี้

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2 \qquad \text{(ก.1)}$$

เมื่อ $E(u,v)$ เป็นการเปลี่ยนแปลงความเข้มของสีเมื่อเลื่อนตำแหน่งของ feature ไปเล็กน้อยเป็นระยะ $[u,v]$, $I(x,y)$ เป็นความเข้มของสีที่ตำแหน่ง $[x,y]$, $I(x+u,y+v)$ เป็นความเข้มของสีเมื่อเลื่อนตำแหน่งไปเล็กน้อย ส่วน $w(x,y)$ จะเป็น window ฟังก์ชัน มีหน้าที่กำหนดขอบเขตของผลรวมการหาความเปลี่ยนแปลงความเข้มของสีบริเวณรอบ ๆ ตำแหน่ง feature ซึ่ง window ฟังก์ชันก็มีได้หลายรูปแบบดังแสดงในรูปที่ ก.2

(ก) มีค่าเป็น 1 เมื่ออยู่ในกรอบ เป็น 0 เมื่ออยู่นอกกรอบ (ข) มีการกระจายแบบ Gaussian

รูปที่ ก.2: window ฟังก์ชันสำหรับ Harris corner detector

เมื่อสมมุติให้การเลื่อนตำแหน่ง $[u, v]$ มีค่าน้อยมากจะประมาณ $I(x+u, y+v)$ ด้วย Taylor's expansion ได้ดังนี้

$$I(x+u, y+v) = I(x,y) + I_x u + I_y v + \varepsilon \quad \text{(ก.2)}$$

$$E(u,v) = \sum_{x,y} w(x,y)[I_x u + I_y v + \varepsilon]^2 \quad \text{(ก.3)}$$

$$E(u,v) \approx \sum_{(} x,y) w(x,y)[I_x^2 u^2 + I_x u I_y v + I_y v I_x u + I_y^2 v^2] \quad \text{(ก.4)}$$

$$E(u,v) \approx \sum_{(} x,y) w(x,y) \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad \text{(ก.5)}$$

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M(x,y) \begin{bmatrix} u \\ v \end{bmatrix} \quad \text{(ก.6)}$$

โดยที่ $M(x,y)$ เป็นโครงสร้างความเข้มสีบริเวณใกล้เคียงของตำแหน่ง featrue

$$M(x,y) = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \quad \text{(ก.7)}$$

เมื่อ I_x เป็นอนุพันธ์อันดับหนึ่งของความเข้มของสีในแนวแกน x ส่วน I_y เป็นอนุพันธ์อันดับหนึ่งของความเข้มของสีในแนวแกน y

ค่าการวัดความเป็นมุม (corner response) สามารถหาได้จาก

$$R = detM - k(traceM)^2 \quad \text{(ก.8)}$$

โดยถ้า R มีค่ามากแสดงว่ามีค่าความเป็นมุมสูงที่จุด (x,y) แต่ถ้า R มีค่าน้อยแสดงว่าที่จุดนั้นมีค่าความเป็นมุมต่ำ หมายความว่าจุดนั้นไม่ใช่ deature ที่ดี

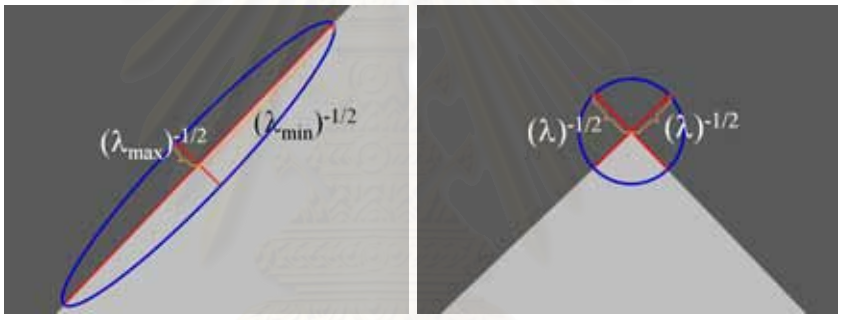
เมื่อวิเคราะห์การเปลี่ยนแปลงความเข้มของสี ด้วย eigenvalue จะได้ว่าเนื่องจาก $M(x,y)$ เป็น Symmetric matrix จึงสามารถหา Eigendecomposition ได้เป็น

$$M(x,y) = U \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} U^T \quad \text{(ก.9)}$$

โดยที่ λ_1, λ_2 เป็นค่า eigenvalue ของ $M(x,y)$ และ U เป็น matrix ซึ่งที่แต่ละแถวของ matrix ประกอบด้วยค่า eigenvector ของ $M(x,y)$

สำหรับค่า eigenvalue ของ $M(x,y)$ นั้นแสดงถึงค่าความเปลี่ยนแปลงค่าสีในทิศทางที่ระบุโดย eigenvector ยกตัวอย่างเช่นในภาพ ก.3ก eigenvalue มีค่าสูงมาก 1 ค่าแสดงว่าบริเวณนั้นมีการเปลี่ยนแปลงของค่าสีมากในหนึ่งทิศทาง ส่วนในภาพ ก.3ข มีค่า eigenvalue สูงทั้งสองค่า แสดงว่าบริเวณนั้นมีการเปลี่ยนแปลงของค่าสีมากทั้งสองทิศทาง ดังนั้นการพิจาณาลักษณะของ feature จึงสามารถดูได้จากค่า eigenvalue ซึ่งเป็นที่มาของสูตรการคำนวนค่าการวัดความเป็นมุม (corner response) (ก.8) โดยการพิจาณาลักษณะของ feature สามารถแบ่งเป็นเงื่อนไขได้เป็น

- ถ้า eigenvalue มีค่าต่ำมากทั้งสองค่า แสดงว่าภาพบริเวณนั้นราบเรียบ (flat)
- ถ้า eigenvalue ค่าหนึ่งมีค่าสูง ส่วนอีกค่ามีค่าต่ำ แสดงว่า feature ตำแหน่งนั้นเป็นขอบ (edge) ของวัตถุ
- ถ้า eigenvalue มีค่าสูงทั้งสองค่า แสดงว่า feature ตำแหน่งนั้นเป็นมุม (corner) ของวัตถุ



(ก) (ข)

รูปที่ ก.3: ค่า eigenvalue ของ edge และ corner

และเนื่องจาก U เป็น Orthogonal matrix ดังนั้น determinant ของ U จึงมีค่าเป็น 1 การค่า determinant ของ $M(x,y)$ จึงหาได้จาก

$$detM = \lambda_1 \lambda_2 \tag{ก.10}$$

ส่วน trace ของ $M(x,y)$ หาได้จาก

$$traceM = \lambda_1 + \lambda_2 \tag{ก.11}$$

ภาคผนวก ข

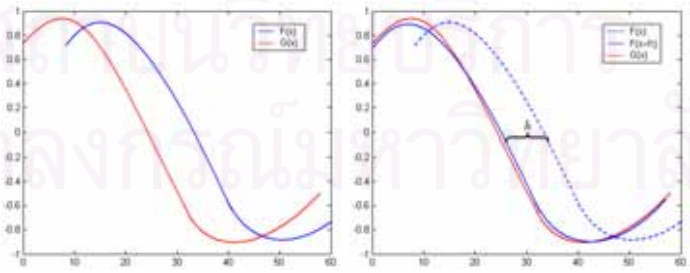
LUCAS & KANADE OPTICAL FLOW

Optical flow เป็นวิธีที่ใช้ในการติดตามการเคลื่อนที่ของ Feature ในภาพโดยพิจารณาจากการเปลี่ยนแปลงของลำดับภาพ ซึ่งการเคลื่อนที่ของ Feature นั้นอาจจะเกิดจากการเคลื่อนที่ของวัตถุ หรือเกิดจากการเคลื่อนที่ของกล้องก็ได้ แนวคิดของ Optical flow ก็คือ เมื่อต้องการติดตามการเคลื่อนที่ของ Feature ในภาพปัจจุบันโดยรู้ตำแหน่งของ Feature จากภาพในอดีต วิธีการของ Optical flow จะพยายามหาการวางตัวของ patch จากภาพในอดีตซึ่งแทนบริเวณรอบ ๆ feature บนภาพปัจจุบัน เพื่อหาตำแหน่งที่มีความสอดคล้องระหว่าง patch กับภาพปัจจุบันมากสุด ดังแสดงได้ดังรูปที่ 1 ซึ่งการทำงานของ Optical flow นั้นเมื่อได้รับภาพใหม่มา ก็จะทำการติดตามตำแหน่ง feature บนภาพใหม่จากตำแหน่งของ feature ในภาพก่อนหน้า เป็นเช่นนี้ไปเรื่อย ๆ

รูปที่ ข.1: รูปการติดตามการเลื่อนตำแหน่งของ Optical flow

วิธีการทำงานของ Optical flow

การทำงานของ Optical flow นั้นจะสมมุติว่าข้อมูลภาพมีความต่อเนื่องกัน สำหรับ Optical Flow ในหนึ่งมิตินั้น จากรูปที่ ข.2ก แสดงให้เห็นค่าสีของภาพสองภาพ ในที่นี้กำหนดให้ $F(x)$ เป็นค่าสีของภาพในอดีต ส่วน $G(x)$ เป็นค่าสีของภาพปัจจุบัน



(ก) (ข)

รูปที่ ข.2: แสดงค่าสีของภาพสองภาพในหนึ่งมิติ

เมื่อสมมุติให้มีการเลื่อนรูปในอดีต ไปเป็นระยะ h เพื่อให้รูปในอดีตซ้อนทับกับรูปปัจจุบันดังรูปที่ ข.2ข ความผิดพลาดเนื่องจากความไม่พอดีของการซ้อนทับหาได้จาก

$$E = \sum_{x \in R} [F(x+h) - G(x)]^2 \qquad \text{(ข.1)}$$

และเมื่อประมาณฟังก์ชัน $F(x+h)$ ด้วยการ Taylor's expansion จะได้ว่า

$$E = \sum_{x \in R} [F(x) + hF^{'}(x) + \varepsilon - G(x)]^2 \quad \text{(ข.2)}$$

$$E \approx \sum_{x \in R} [F(x) + hF^{'}(x) - G(x)]^2 \quad \text{(ข.3)}$$

โดยที่ $F^{'}(x)$ เป็นอนุพันธ์อันดับหนึ่งของฟังก์ชัน F ที่ตำแหน่ง x

สำหรับการติดตามการเคลื่อนที่ด้วย Optical Flow นั้นผลลัพธ์การติดตามการเคลื่อนที่จะดีสุดก็ต่อเมื่อความคลาดเคลื่อนของการซ้อนทับกันของภาพสองภาพ (E) มีค่าต่ำสุด ดังนั้นจะสามาถประมาณการเลื่อนตำแหน่งของ Feature (h) ได้โดยการแก้สมการอนุพันธ์ของความคลาดเคลื่อนเทียบกับ h มีค่าเท่ากับ 0

$$0 = \frac{\partial E}{\partial h} \approx \frac{\partial}{\partial h} \sum_{x} [F(x) + hF^{'}(x) - G(x)]^2 \quad \text{(ข.4)}$$

$$= \sum_{x} 2F^{'}(x)[F(x) + hF^{'}(x) - G(x)] \quad \text{(ข.5)}$$

$$= \sum_{x} 2F^{'}(x)[F(x) - G(x)] + \sum_{x} 2hF^{'}(x)^2 \quad \text{(ข.6)}$$

$$h \approx \frac{\sum_x F^{'}(x)[G(x) - F(x)]}{\sum_x F^{'}(x)^2} \quad \text{(ข.7)}$$

การประมาณค่า h ข้างต้นนั้นจะงานได้ดีในบริเวณภาพที่มีการเปลี่ยนแปลงแบบ linear (ค่าอนุพันธ์อันดับสองของ $F(x)$ มีค่าต่ำ ($F''(x) \approx 0$)) แต่ถ้าค่าอนุพันธ์อันดับสองของ $F(x)$ มีค่าสูงการประมาณการเลื่อนตำแหน่งของ Feature (h) จะทำได้ยากเพราะมีความคลาดเคลื่อนจากการประมาณฟังก์ชัน $F(x+h)$ ด้วยการ Taylor's expansion ดังนั้นจึงจะต้องคำนวนค่าน้ำหนักของความเชื่อมั่นในแต่ละจุด x ด้วย และเนื่องจากว่า

$$F^{''}(x) \approx (F^{'}(x) - G^{'}(x))/h$$

ดังนั้นค่าน้ำหนักจะกำหนดให้เป็น

$$w(x) = \frac{1}{|F^{'}(x) - G^{'}(x)|} \quad \text{(ข.8)}$$

ซึ่งจะหมายความว่าถ้าหาก(ค่าอนุพันธ์อันดับสองของ $F(x)$ มีค่ามากค่าน้ำหนักก็จะน้อยดังนั้นสมการการประมาณการเลื่อนตำแหน่งของ Feature จึงเขียนได้ใหม่เป็น

$$h \approx \frac{\sum_x w(x)F^{'}(x)[G(x) - F(x)]}{\sum_x w(x)F^{'}(x)^2} \quad \text{(ข.9)}$$

และเนื่องจากว่าค่า h ที่ได้นั้นเป็นค่าที่เกิดจากการประมาณฟังก์ชัน $F(x+h)$ ด้วยการ Taylor's expansion ดังนั้นการประมาณค่า h แบบวนซ้ำจะให้ผลลัพธ์ที่แม่นยำกว่า ดังนั้นสมการการประมาณการ

เลื่อนตำแหน่งของ Feature จะได้ว่า

$$h_{k+1} = h_k + \frac{\sum_x w(x)F'(x+h_k)[G(x) - F(x+h_k)]}{\sum_x w(x)F'(x+h_k)^2} \quad \text{(ข.10)}$$

การทำงานข้างต้นจะทำการประมาณโดยเริ่มจาก $h_0 = 0$ และวนซ้ำไปเรื่อย ๆ จนกว่าค่า h_k จะมีการเปลี่ยนแปลงน้อยมาก

สำหรับกรณี n มิติกำหนดให้ x และ h เป็นเวกเตอร์ขนาด n มิติ โดยที่ $x = [x_1 \quad x_2 \quad \cdots \quad x_n]$ และ$h = [h_1 \quad h_2 \quad \cdots \quad h_n]$ จะประมาณฟังก์ชัน $F(x+h)$ ด้วยการ Taylor's expansion จะได้เป็น

$$F(x+h) \approx F(x) + h\frac{\partial}{\partial x}F(x) \quad \text{(ข.11)}$$

โดยที่ $\partial/\partial x$ เป็น gradient operator เทียบ x

$$\frac{\partial}{\partial x} = \left[\begin{array}{cccc} \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_1} & \cdots & \frac{\partial}{\partial x_n} \end{array} \right]^T \quad \text{(ข.12)}$$

เมื่อคำนวนหาอนุพันธ์เท่ากับ 0 จะได้ว่า

$$0 = \frac{\partial E}{\partial h} \approx \frac{\partial}{\partial h}\sum_x \left[F(x) + h\frac{\partial F}{\partial x} - G(x)\right]^2 \quad \text{(ข.13)}$$

$$= \sum_x 2\frac{\partial F}{\partial x}\left[F(x) + h\frac{\partial F}{\partial x} - G(x)\right] \quad \text{(ข.14)}$$

$$h = \left[\sum_x 2\frac{\partial F}{\partial x}^T [G(x) - F(x)]\right]\left[\sum_x 2\frac{\partial F}{\partial x}^T \frac{\partial F}{\partial x}\right]^{-1} \quad \text{(ข.15)}$$

จะเห็นได้ว่าค่าประมาณการเลื่อนตำแหน่งของ Feature (h) ในหลายมิติมีความคล้ายคลึงกับการประมาณในหนึ่งมิติเป็นอย่างมาก ซึ่งเมื่อประมาณค่า h แบบวนซ้ำจะได้ว่า

$$h_{k+1} = h_k + \left[\sum_x 2\frac{\partial F}{\partial x}^T [G(x) - F(x+h_k)]\right]\left[\sum_x 2\frac{\partial F}{\partial x}^T \frac{\partial F}{\partial x}\right]^{-1} \quad \text{(ข.16)}$$

และเมื่อหาการเลื่อนตำแหน่งของ Feature (h) ได้แล้วก็จะประมาณตำแหน่ง feature บนภาพปัจจุบันได้

ประวัติผู้เขียนวิทยานิพนธ์

นายยุทธนา สุทธสุภา เกิดเมื่อวันที่ 8 พฤษภาคม 2528 ที่จังหวัดเชียงใหม่ สำเร็จการศึกษาระดับปริญญาวิศวกรรมศาสตรบัณฑิต (เกียรตินิยม อันดับหนึ่ง) สาขาวิชาวิศวกรรมคอมพิวเตอร์ จากคณะวิศวกรรมศาสตร์จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2549 และเข้าศึกษาในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ณ ภาควิชาวิศวกรรมคอมพิวเตอร์คณะวิศวกรรมศาสตร์จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2550

มีความสนใจในงานวิจัยที่เกี่ยวข้องกับหุ่นยนต์และระบบอัตโนมัติ โดยเฉพาะงานวิจัยที่เกี่ยวข้องกับการวิเคราะห์ข้อมูลสิ่งแวดล้อมจากเซนเซอร์ของหุ่นยนต์ เพื่อให้หุ่นยนต์สามารถเข้าใจสภาพแวดล้อมขณะปฏิบัติงานได้