

การสร้างเส้นทางเดินในปริภูมิรูปภาพสำหรับหุ่นยนต์เคลื่อนที่ด้วยล้อ



นายวาริน ภูมรินทร์

จุฬาลงกรณ์มหาวิทยาลัย

CHULALONGKORN UNIVERSITY

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)  
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)  
are the thesis authors' files submitted through the University Graduate School.

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล

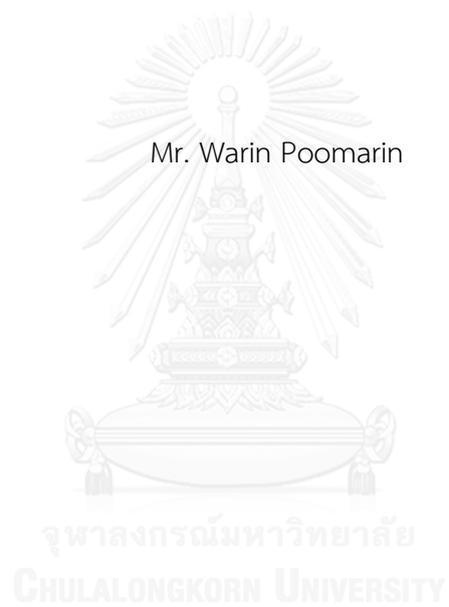
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2558

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

TRAJECTORY GENERATION IN IMAGE SPACE FOR A MOBILE ROBOT

Mr. Warin Poomarin



A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Master of Engineering Program in Mechanical Engineering  
Department of Mechanical Engineering  
Faculty of Engineering  
Chulalongkorn University  
Academic Year 2015  
Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การสร้างเส้นทางเดินในปริภูมิรูปภาพสำหรับหุ่นยนต์เคลื่อนที่ด้วยล้อ
โดย	นายวาริน ภูมรินทร์
สาขาวิชา	วิศวกรรมเครื่องกล
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.รัชทิน จันทร์เจริญ
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	ศาสตราจารย์ ดร.วิบูลย์ แสงวีระพันธุ์ศิริ

---

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์  
(ศาสตราจารย์ ดร.บัณฑิต เอื้ออาภรณ์)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร.สัมพันธ์ จันทรานุกวัฒน์)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก  
(รองศาสตราจารย์ ดร.รัชทิน จันทร์เจริญ)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม  
(ศาสตราจารย์ ดร.วิบูลย์ แสงวีระพันธุ์ศิริ)

.....กรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร.วิทยา วัฒนสุโขประสิทธิ์)

.....กรรมการภายนอกมหาวิทยาลัย  
(ดร.กฤษณันท์ มะลิทอง)

วาริน ภูมิรินทร์ : การสร้างเส้นทางเดินในปริภูมิรูปภาพสำหรับหุ่นยนต์เคลื่อนที่ด้วยล้อ (TRAJECTORY GENERATION IN IMAGE SPACE FOR A MOBILE ROBOT) อ.ที่ปรึกษา  
วิทยานิพนธ์หลัก: รศ. ดร.รัชทิน จันท์เจริญดร., อ.ที่ปรึกษาวิทยานิพนธ์ร่วม: ศ. ดร.วิบูลย์  
แสงวีระพันธุ์ศิริตร., 89 หน้า.

วิทยานิพนธ์นี้นำเสนอวิธีการสำหรับหุ่นยนต์เคลื่อนที่ด้วยล้อในการเข้าจอดที่สถานีโดยการใช้สัญญาณบ่อนกลับจากกล้องแบบ RGBD ด้วยกล้องแบบ RGBD สามารถตรวจจับได้ทั้งสีและรูปทรงของวัตถุ ซึ่งรวมถึงสิ่งกีดขวาง กำแพง และพื้น ซึ่งสามารถนำรูปภาพที่ได้มาประมวลผลเพื่อหาตำแหน่งบนพิกัดรูปภาพและคาร์ทีเซียน จากนั้น จะทำการวางตำแหน่งสิ่งกีดขวาง กำแพง และพื้นที่ตรวจพบบนแผนที่สองมิติแบบกริด วิทยานิพนธ์นี้ยังได้นำเสนอวิธีการสร้างเส้นทางเดินสำหรับหุ่นยนต์เพื่อมุ่งหน้าเข้าสู่สถานีจอด โดยจะเป็นเส้นทางเดินแบบเรียบที่สามารถหลบสิ่งกีดขวางและมุ่งหน้าเข้าจอดที่สถานีในทิศทางที่ถูกต้องได้ วิธีการที่นำเสนอมีพื้นฐานอยู่บนเทคนิคสนามศักย์ โดยได้ใช้การทดลองจริงในการแสดงศักยภาพของวิธีการ



ภาควิชา วิศวกรรมเครื่องกล

สาขาวิชา วิศวกรรมเครื่องกล

ปีการศึกษา 2558

ลายมือชื่อนิสิต .....

ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ลายมือชื่อ อ.ที่ปรึกษาร่วม .....

# # 5570370521 : MAJOR MECHANICAL ENGINEERING

KEYWORDS: RGBD CAMERA / MOBILE ROBOT / TRAJECTORY / DOCKING

WARIN POOMARIN: TRAJECTORY GENERATION IN IMAGE SPACE FOR A MOBILE ROBOT. ADVISOR: ASSOC. PROF.RATCHATIN CHANCHAROEN, Ph.D., CO-ADVISOR: PROF.VIBOON SANGVERAPHUNSIRI, Ph.D., 89 pp.

This thesis proposed a technique to smoothly dock a differential drive mobile robot based on a RGBD camera. The RGBD camera is capable of capture both the color and geometry of obstacles, walls and floor that are processed to determine their image and Cartesian positions. Obstacles, walls and floor are then placed on a 2D grid map which is used to navigate the robot. The technique to generate the trajectory towards the docking position, i.e. a smooth trajectory that stay away from obstacles but head to the docking position with a correct orientation, is also proposed. The trajectory is generated based on the potential field technique. The experimentation is used to demonstrate the applicable of the proposed technique.



Department:	Mechanical Engineering	Student's Signature .....
Field of Study:	Mechanical Engineering	Advisor's Signature .....
Academic Year:	2015	Co-Advisor's Signature .....

## กิตติกรรมประกาศ

ข้าพเจ้าขอขอบพระคุณ ท่าน รองศาสตราจารย์ ดร.รัชทิน จันทรเจริญ อาจารย์ที่  
ปรึกษาวิทยานิพนธ์ และ ท่าน ศาสตราจารย์ ดร. วิบูลย์ แสงวีระพันธุ์ศิริ อาจารย์ที่ปรึกษาร่วม  
วิทยานิพนธ์ ที่ให้คำแนะนำ คอยชี้แนะ และให้ความช่วยเหลือสิ่งต่างๆ ทั้งทางด้านอุปกรณ์  
สถานที่ และเงินทุน จนวิทยานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี และขอขอบคุณ เพื่อน ๆ รุ่นพี่ และ  
รุ่นน้องทุกท่าน ที่ได้ให้ความช่วยเหลือ ข้อเสนอแนะ และกำลังใจในการทววิจัยตลอดมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ คุณพ่อคุณแม่ และครอบครัวที่ได้คอยเป็น  
กำลังใจ และให้การสนับสนุนในทุก ๆ ด้านแก่ข้าพเจ้าโดนเสมอจนสำเร็จการศึกษา



## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญรูป .....	ญ
สารบัญตาราง.....	1
หน้า.....	1
บทที่ 1 บทนำ .....	1
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์ .....	2
1.3 ขอบเขตของโครงการ.....	3
1. 4 ขั้นตอนการดำเนินงานวิทยานิพนธ์.....	3
บทที่ 2 ปรัชญาธรรม.....	4
2.1 การทำงานร่วมกันระหว่างหุ่นยนต์เคลื่อนที่ด้วยล้ออัตโนมัติกับอุปกรณ์ตรวจรู้.....	4
2.2 ความเป็นมาของการสอบเทียบกล้อง .....	5
2.3 การสร้างเส้นทางเดินของหุ่นยนต์.....	6
บทที่ 3 ทฤษฎีที่เกี่ยวข้อง.....	8
3.1 แบบจำลองของกล้อง, เลนส์และการประมวลผลรูป.....	8
3.1.1 แบบจำลองกล้อง .....	8
3.1.2 การบิดเบี้ยวของเลนส์.....	12
3.1.3 การสอบเทียบกล้อง.....	13
3.1.4 การแปลงระนาบโฮโมกราฟฟี .....	15

3.1.5 การหาตัวแปรภายในและภายนอกของกล้อง .....	18
3.1.6 ปริภูมิสี.....	21
3.1.7 การคัดกรองรายละเอียดของภาพ .....	23
3.1.8 การคำนวณหาตำแหน่งของวัตถุในพิกัดสามมิติ .....	23
3.2 การแทนที่ตำแหน่งด้วยตารางแผนที่ Grid Occupation .....	25
3.3 การหาเส้นตั้งฉากของกราฟเชิงเส้นด้วยวิธีการถดถอยแบบเชิงเส้น.....	25
3.4 การสร้างเส้นทางเดินด้วยสนามศักย์.....	28
3.4.1 การสร้างสนามศักย์ (Potential field).....	29
3.4.2 ประเภทของสนามศักย์.....	31
3.4.3 การรวมสนามศักย์.....	33
3.5 การโปรแกรมหุ่นยนต์เคลื่อนที่ด้วยล้อ แบบ Differential Drive.....	34
3.5.1 การระบุตำแหน่งของหุ่นยนต์เคลื่อนที่ด้วยล้อ.....	35
3.5.2 การสร้างคำสั่งที่ควบคุมหุ่นยนต์ iRobot Create .....	36
บทที่ 4 ภาพรวมของระบบ.....	38
4.1 อุปกรณ์ตรวจรู้ Kinect.....	38
4.1.1 การตรวจจับความลึก.....	40
4.2 หุ่นยนต์เคลื่อนที่ด้วยล้อ iRobot Create .....	43
บทที่ 5 ผลการทดลอง.....	44
5.1 ทดสอบคำสั่งหุ่นยนต์เคลื่อนที่ด้วยล้อ iRobot Create .....	44
5.2 การติดตั้งระบบ.....	46
5.3 การทดสอบความแม่นยำของอุปกรณ์ตรวจรู้ Kinect.....	49
5.4 การประมวลผลผลลัพธ์จากอุปกรณ์ตรวจรู้ Kinect.....	51
5.4.1 การแยกแยะองค์ประกอบของสิ่งแวดล้อม .....	55

5.4.2 การสร้างแผนที่ และการปรับปรุงแผนที่ให้สามารถใช้ประโยชน์ได้.....	56
5.4.2 การหาเส้นตั้งฉากกับกำแพง .....	60
5.4.3 การสร้างเส้นทางเดินหุ่นยนต์ด้วยสนามศักย์ .....	61
5.5 แปลงเส้นทางการเคลื่อนที่ให้เป็นคำสั่งควบคุมหุ่นยนต์ .....	67
5.6 การเทียบเส้นทางเดินกลับไปยังปริภูมิภาพ.....	68
5.7 การใช้กล้อง RGB เพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์ให้เคลื่อนที่เข้าหาเป้าหมาย.....	70
5.8 การหาความสัมพันธ์ระหว่างตำแหน่งของวัตถุเป้าหมายและภาพที่ได้จากกล้องวิดีโอสี.....	72
5.9 ระบบกล้องตรวจจับตำแหน่งสเตอริโอออปติแตรีก (Opti track).....	76
5.10 ผลลัพธ์ที่ได้จากกล้องสเตอริโอออปติแตรีก.....	78
บทที่ 6 สรุปผลการวิจัย.....	80
6.1 สรุปผลการวิจัย.....	80
6.2 ข้อเสนอแนะเพื่องานวิจัยในอนาคต .....	82
รายการอ้างอิง .....	83
ประวัติผู้เขียนวิทยานิพนธ์ .....	89

## สารบัญรูป

หน้า

รูปที่ 1 แบบจำลองกล้องรูเข็มและแกนอ้างอิงที่เกี่ยวข้อง .....	10
รูปที่ 2 ภาพการบิดเบี้ยวของเลนส์ .....	11
รูปที่ 3 การฉายของตำแหน่งจุด P ในปริภูมิ 3 มิติไปยังจุด p บนปริภูมิภาพ.....	14
รูปที่ 4 วัตถุประสงค์เทียบ .....	15
รูปที่ 5 การฉายภาพมุมตีบไปยังระนาบโดยใช้วิธี โฮโมกราฟฟี .....	17
รูปที่ 6 การเทียบตำแหน่งจุดที่มุมของตารางหมากรุก จากในปริภูมิ 3 มิติไปยังปริภูมิรูปภาพ.....	17
รูปที่ 7 ปริภูมิสี่แบบต่างๆ.....	22
รูปที่ 8 ขั้นตอนการคัดกรองรายละเอียดของภาพ .....	24
รูปที่ 9 ตัวอย่างของการแทนตำแหน่งของตารางแผนที่ของสิ่งแวดล้อม .....	24
รูปที่ 10 ตัวอย่างการถดถอยแบบเชิงเส้นโดยการสร้างฟังก์ชันเส้นตรงจากชุดข้อมูลที่เป็นจุด ระนาบ 2 มิติ.....	25
รูปที่ 11 แผนภาพอธิบายพฤติกรรมของหุ่นยนต์ ตามนิยามของ Murphy .....	28
รูปที่ 12 การสร้างเวกเตอร์แรงจากแหล่งกำเนิดสนามศักดิ์.....	31
รูปที่ 13 สนามศักดิ์แบบต่างๆ.....	32
รูปที่ 14 การสร้างเส้นทางทางเดินจากสนามศักดิ์รวม .....	33
รูปที่ 15 กระบวนการโปรแกรมรหัสเอ็นซี .....	34
รูปที่ 16 ความสัมพันธ์ของเวกเตอร์ระหว่างจุดแต่ละจุด .....	37
รูปที่ 17 อุปกรณ์ตรวจจับ Kinect.....	39
รูปที่ 18 ลายจุดอินฟราเรดที่เกิดจากการฉายลำแสงอินฟราเรด ซึ่งเป็นลายที่มาจาก หน่วยความจำของอุปกรณ์ฉาย [31].....	41
รูปที่ 19 ตำแหน่งที่ความแตกต่างกันของจุดในภาพของกล้องรับความลึก (XL,YL) กับภาพจาก กล้องฉายลำแสงอินฟราเรด (XR,YR) ซึ่งค่านี้เป็นค่าที่อยู่ในหน่วยความทรงจำ.....	41

รูปที่ 20 จุดอินฟราเรดที่ตรวจรู้จากกล้องตรวจจับความลึก [31].....	42
รูปที่ 21 ตัวอย่าง ภาพแสดงความลึก โดยค่าความลึกของตำแหน่งภาพแต่ละจุดเป็นค่าสีเทาขาว [31].....	42
รูปที่ 22 หุ่นยนต์เคลื่อนที่ด้วยล้อ iRobot Create .....	43
รูปที่ 23 เส้นที่แดงคือเส้นทางเดินอ้างอิง เส้นสีเขียวคือเส้นทางเดินของหุ่นยนต์ .....	45
รูปที่ 24 อุปกรณ์ที่ใช้ในการทดลอง .....	46
รูปที่ 25 การติดตั้ง Kinect บนตัวหุ่นยนต์ iRobot Create .....	47
รูปที่ 26 แผนภาพรวมของระบบเข้าจออัตโนมัติ .....	48
รูปที่ 27 ตำแหน่งของเป้าหมายที่ใช้ในการทดสอบอุปกรณ์ตรวจรู้ Kinect .....	49
รูปที่ 28 ความความคลาดเคลื่อนเฉลี่ยเทียบกับระยะห่างของเป้าหมาย .....	50
รูปที่ 29 ภาพที่ได้จากการอุปกรณ์ตรวจรู้ Kinect .....	52
รูปที่ 30 จุดสีของภาพในสามมิติ.....	53
รูปที่ 31 รูปที่ได้จากการแยกแยะองค์ประกอบ.....	53
รูปที่ 32 การหาตำแหน่งของวัตถุเป้าหมายจากภาพของกล้อง Kinect.....	54
รูปที่ 33 แผนที่ที่ได้จากการแยกแยะส่วนประกอบ .....	57
รูปที่ 34 แผนที่ที่ได้จากกระบวนการ grid occupation .....	58
รูปที่ 35 แผนที่หลังจากทำ grid occupation และ กำหนดให้เป้าเป็นจุด (0,0).....	59
รูปที่ 36 เส้นตรงที่ได้จากจุดที่เป็นกำแพง.....	60
รูปที่ 37 สนามสนามศักดิ์จากมุมมองทั้ง 3 มิติ.....	64
รูปที่ 38 สนามสนามศักดิ์ตามแกน x-y .....	65
รูปที่ 39 เส้นทางเดินของหุ่นยนต์ที่ได้.....	66
รูปที่ 40 ตัวอย่างภาพที่ถ่ายภาพตารางหมากรุกจากหลาย ๆ มุม .....	68
รูปที่ 41 ภาพจำลองการถ่านภาพตารางหมากรุกจากหลาย ๆ มุม.....	68
รูปที่ 42 แสดงผลลัพธ์ที่สามารถหาความสัมพันธ์ของตำแหน่งวัตถุในภาพได้ .....	69

รูปที่ 43 เส้นทางเดินของหุ่นยนต์ในปริภูมิภาพ .....	70
รูปที่ 44 ภาพที่ได้จากการอุปกรณ์ตรวจรู้ Kinect หลังสั่งหุ่นยนต์เคลื่อนที่ .....	71
รูปที่ 45 ฟังก์ชันความสัมพันธ์ระหว่างระยะห่างของเป้าหมายกับขนาดเส้นผ่านศูนย์กลางของ เป้าหมายในภาพจากกล้อง RGB.....	73
รูปที่ 46 ความสัมพันธ์ของตำแหน่งเป้าหมายในปริภูมิภาพกับมุมหัน .....	75
รูปที่ 47 ภาพจากโปรแกรมของ OptiTrack Tool ของกล้องแต่ละตัว .....	77
รูปที่ 48 บริเวณพื้นที่ทำงานที่สามารถตรวจจับได้ด้วยระบบกล้อง OptiTrack.....	77
รูปที่ 49 การบันทึกตำแหน่งด้วยกล้องออฟติแทร็ก .....	78
รูปที่ 50 ภาพจากการอุปกรณ์ตรวจรู้ Kinect หลังสั่งหุ่นยนต์เคลื่อนที่.....	79



## สารบัญตาราง

	หน้า
ตารางที่ 1 ค่าตำแหน่งในปริภูมิ $x,y$ จากกล้องความลึก.....	50
ตารางที่ 2 ค่าตัวแปรที่ใช้สั่งการหุ่นยนต์.....	67
ตารางที่ 3 ความยาวเส้นผ่านศูนย์กลางของวัตถุเป้าหมายที่ได้จากกล้องวิดีโอ RGB ที่ตำแหน่ง ต่าง.....	72
ตารางที่ 4 ตำแหน่งจุดศูนย์กลางของเป้าหมายที่ปรากฏในภาพจากกล้องวิดีโอ RGB ณ ตำแหน่ง ต่างๆ.....	74
ตารางที่ 5 ค่ามุมหันของวัตถุเมื่อเทียบกับเส้นแกนमुखสำคัญของกล้อง ตำแหน่งจุดศูนย์กลางของ ...	74



# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญ

เทคโนโลยีในปัจจุบันนั้นได้มีการพัฒนาไปอย่างการกระโดดทำให้หุ่นยนต์เคลื่อนที่ด้วยล้อได้รับการพัฒนาอย่างรวดเร็ว ส่งผลให้หุ่นยนต์เคลื่อนที่ด้วยล้อได้รับความนิยมมากขึ้นเรื่อยๆ ไม่ใช่เพียงแต่ในโรงงานอุตสาหกรรมเท่านั้น แต่รวมถึงในออฟฟิศ [1] และในครัวเรือนด้วยเช่นกัน [2] เพื่อทำหน้าที่เฝ้าระวัง [3] หรือการสื่อสารด้วยภาพทางไกล [4] เนื่องจากการพัฒนาทำให้หุ่นยนต์มีความฉลาดมากขึ้นและราคาถูกลงอย่างมาก หุ่นยนต์ iRobot's Roomba เป็นตัวอย่างที่ดีขึ้นหุ่นยนต์ที่ถูกเลือกนำมาใช้งานกับผู้คน

หุ่นยนต์เคลื่อนที่ด้วยล้อในปัจจุบันนี้มีแนวโน้มที่จะทำงานใกล้ชิดกับมนุษย์มากขึ้น ด้วยจุดเด่นที่ว่าหุ่นยนต์ชนิดนี้ที่มีพื้นที่การทำงานกว้างมาก แต่อย่างไรก็ตามความปลอดภัยต่อมนุษย์ขณะทำงานในพื้นที่การทำงานของหุ่นยนต์นั้นคือปัจจัยหลักที่ถูกใช้ตัดสินว่าควรใช้หุ่นยนต์ทำงานนั้นหรือไม่ ด้วยเหตุนี้การเพิ่มความสามารถในการหลีกเลี่ยงอันตรายหรืออุบัติเหตุต่อผู้ใช้งานจึงเป็นประเด็นสำคัญที่ควรได้รับการพัฒนา

ระบบอัตโนมัตินั้นหมายความว่าหุ่นยนต์นั้นควรจะสามารถซ่อมบำรุงตัวเองและสามารถตัดสินใจเลือกเส้นทางการเคลื่อนที่ของตนเองได้ [5] อีกนัยหนึ่งก็คือหุ่นยนต์ควรสามารถเคลื่อนที่ไปยังตำแหน่งที่ได้รับมอบหมายได้ถูกต้องและมีความปลอดภัยมากขึ้น การพัฒนาระบบสร้างเส้นทางเดินและการหลบหลีกสิ่งกีดขวางจึงเป็นเรื่องที่ได้รับความสนใจเป็นอย่างมาก เพราะการหลบหลีกสิ่งกีดขวางเป็นคุณสมบัติขั้นพื้นฐานของหุ่นยนต์เคลื่อนที่ด้วยล้อ นั้นหมายความว่าฟังก์ชันอื่นๆที่ใช้ควบคุมการทำงานของหุ่นยนต์ต้องสอดคล้องกับคุณสมบัติพื้นฐานนี้ [6] ถ้าหากหุ่นยนต์สามารถหลีกเลี่ยงการสัมผัสกับวัตถุอื่นในสิ่งแวดล้อมได้นั้นหมายความว่า คุณสมบัติระดับสูงอื่นๆ ก็สามารถทำงานได้อย่างปลอดภัย

การรับรู้ถึงสิ่งที่อยู่ในสภาพแวดล้อมโดยรอบของหุ่นยนต์ หุ่นยนต์จะต้องมีอุปกรณ์ตรวจจับบางอย่างเพื่อรับข้อมูลสภาพและลักษณะสิ่งแวดล้อม อุปกรณ์ตรวจจับที่มักถูกเลือกเพื่อใช้หาวัตถุสิ่งกีด

ขวางได้แก่ อุปกรณ์วัดระยะห่างด้วยเลเซอร์ โซนาร์ และกล้องวีดีโอ [7] อุปกรณ์วัดระยะห่างด้วยเลเซอร์จะหาความลึกของวัตถุในสิ่งแวดล้อมได้มีความแม่นยำสูงแต่ต้องพลังงานมากและมีราคาแพง วงแหวนอินตราโซนิกกับอุปกรณ์ตรวจจับโซนาร์มีราคาถูกกว่าแต่ค่าที่ได้ก็นั้นหยابกว่าและแม่นยำน้อยกว่า อุปกรณ์ตรวจจับทั้งสอบบนนี้จะวัดความลึกได้เฉพาะแนวขนานกับพื้นเท่านั้น กล้องวีดีโอนั้นแตกต่างออกไปคือ ไม่สามารถวัดตำแหน่งความลึกได้โดยตรง ได้เพียงแคข้อมูลภาพ 2 มิติเป็นข้อมูลดิบ ต้องมีระบบอื่นเข้ามาช่วยเสริมเช่น การใช้กล้องหลายตัว (เช่น stereo camera) [8] เพื่อคำนวณหาความลึกของวัตถุที่พบเห็น ที่น่าสนใจก็คือ Xbox Kinect เป็นกล้อง RGB-D เป็นอุปกรณ์ตรวจจับที่มีความลึกได้แม่นยำ ข้อมูลที่ได้มีลักษณะเหมือนข้อมูลภาพซึ่งได้ทั้ง 2 แนวแกนคือ ในแนวตั้งฉากกับพื้นและแนวขนานกับพื้น

งานวิจัยนี้เสนอการพัฒนาเทคนิคการสร้างเส้นทางเดินเพื่อเข้าจอดอัตโนมัติของหุ่นยนต์เคลื่อนที่ด้วยล้อโดยหลบหลีกสิ่งกีดขวางไปพร้อมกัน ตรวจจับสิ่งกีดขวางและสร้างเส้นทางเพื่อบังคับหุ่นยนต์ให้เคลื่อนที่ไปยังเป้าหมายได้ใช้กล้อง RGB-D ซึ่งเป็นกล้องที่ให้ข้อมูลความลึกด้วย มีประโยชน์เพื่อเพิ่มข้อมูลช่วยให้หุ่นยนต์สามารถเคลื่อนกลับไปยังฐานเพื่อเติมแบตเตอรี่ [9] หรือรับการซ่อมบำรุง [10] งานวิจัยนี้เลือกใช้เฉพาะอุปกรณ์เชิงพาณิชย์ ได้การหุ่นยนต์ iRobot's Create และอุปกรณ์ตรวจจับ Xbox's Kinect ซึ่งเชื่อมต่อกับคอมพิวเตอร์โน้ตบุ๊กทางสายไฟ USB ติดตั้งบนหุ่นยนต์เคลื่อนที่ด้วยล้อเพื่อตรวจจับความลึกของสิ่งแวดล้อมแล้วใช้ระดับความสูงตัดแยกที่เป็นสิ่งกีดขวางออกจากพื้น ระบบจะสร้างเส้นทางเพื่อสั่งการเคลื่อนที่ให้กับหุ่นยนต์ลงบนส่วนที่เป็นพื้น จากนั้นจะรับคำสั่งเพื่อเคลื่อนที่ตามเส้นทางที่ถูกสร้างขึ้น ถึงกระนั้นอุปกรณ์ตรวจจับความลึกก็มีจุดอ่อนอยู่คือไม่สามารถตรวจจับวัตถุที่อยู่ไกลเกินไปได้ เพื่อที่แก้จุดอ่อนเมื่อวัตถุเป้าหมายเริ่มเข้าใกล้จนไม่สามารถตรวจจับได้จะใช้ความสามารถของกล้องวีดีโอสีมาช่วยในการสร้างเส้นทางควบคุมการเคลื่อนที่ของหุ่นยนต์แทน

## 1.2 วัตถุประสงค์

พัฒนาเทคนิคการควบคุมการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่ด้วยล้อเพื่อไปยังเป้าหมายและหลบหลีกสิ่งกีดขวางในระหว่างทาง โดยใช้สัญญาณจากกล้อง RGBD

### 1.3 ขอบเขตของโครงการ

- 1) พัฒนารูปแบบการสั่งการการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่ด้วยล้อแบบ Differential drive และการวัดเส้นทางของหุ่นยนต์
- 2) พัฒนาการจัดการสัญญาณจากกล้อง RGBD และจัดการระบบประมวลรูปภาพ เพื่อค้นหาเป้าหมายและสิ่งกีดขวาง
- 3) พัฒนาการสร้างเส้นทางเดินในปริภูมิรูปภาพ

### 1.4 ขั้นตอนการดำเนินงานวิทยานิพนธ์

- 1) ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง
- 2) พัฒนาระบบจัดการภาพที่จากกล้อง RGBD เพื่อหาตำแหน่งเป้าหมายและวัตถุสิ่งกีดขวาง
- 3) พัฒนาระบบจัดการข้อมูลเพื่อสร้างแผนที่ และระบุตำแหน่งวัตถุลงในแผนที่
- 4) พัฒนาระบบสร้างเส้นทางเดินจากปริภูมิภาพ
- 5) พัฒนาระบบสั่งการหุ่นยนต์เคลื่อนที่ด้วยล้อให้รับคำสั่งจากเส้นทางที่สร้างขึ้นมาได้
- 6) จัดสร้างสนามทดสอบ และประกอบกล้อง RGBD เข้ากับหุ่นยนต์เคลื่อนที่ด้วยล้อ
- 7) ทดสอบการทำงานของระบบหุ่นยนต์

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) เทคนิคการสั่งการหุ่นยนต์เคลื่อนที่ด้วยล้อ และการวัดค่าการเคลื่อนที่
- 2) เทคนิคการประมวลผลภาพที่ได้จากอุปกรณ์กล้อง RGBD เพื่อให้ได้ข้อมูลของวัตถุเป้าหมาย และวัตถุสิ่งกีดขวาง
- 3) เทคนิคการควบคุมหุ่นยนต์ โดยสร้างเส้นทางเดินจากปริภูมิรูปภาพ

## บทที่ 2

### ปริทัศน์วรรณกรรม

#### 2.1 การทำงานร่วมกันระหว่างหุ่นยนต์เคลื่อนที่ด้วยล้ออัตโนมัติกับอุปกรณ์ตรวจรู้

หุ่นยนต์เคลื่อนที่ด้วยล้อจะสามารถค้นหาและเคลื่อนไปยังตำแหน่งจอดอย่างอัตโนมัติ จำเป็นจะต้องมีอุปกรณ์ตรวจรู้และอุปกรณ์ส่งสัญญาณพิเศษช่วยในการหารับรู้สิ่งแวดล้อมเสมอ [10] ยกตัวอย่างเช่น ไฟกระพริบ อุปกรณ์ส่งสัญญาณอินฟราเรด [7] อุปกรณ์ตรวจรู้สัญญาณอินฟราเรด [11] เป็นต้น จนกระทั่งได้พัฒนาใช้แผนที่เพื่อช่วยในการหาเส้นทาง และอุปกรณ์ตรวจรู้การสัมผัสที่ออกแบบพิเศษเพื่อหาที่จอด และบางระบบก็เริ่มใช้การตรวจจับภาพเพื่อหาจุดจอด [12, 13]

ปัจจุบันอุปกรณ์ตรวจรู้ที่ถูกเลือกใช้สำหรับการรับรู้สภาพสิ่งแวดล้อมนั้นมีมากมาย ประกอบกับเทคนิคที่ถูกเลือกใช้ในการช่วยให้หุ่นยนต์ทำให้ภารกิจสำเร็จ ที่น่าสนใจมีดังนี้ [14] ระบบโซนาร์และระบบแสงเลเซอร์ [15] ได้ถูกใช้เป็นอุปกรณ์ตรวจรู้เพื่อช่วยในการควบคุมหุ่นยนต์เคลื่อนที่ด้วยล้อให้สามารถหลบหลีกสิ่งกีดขวางเป็นเวลาหลายปี [16] ในปี 1994 หุ่นยนต์ Dervish การใช้อุปกรณ์วงแหวนโซนาร์เพื่อส่งคลื่นโซนาร์ออกไปทุกทิศทาง และอุปกรณ์ตรวจรู้วัดมุมเพื่อช่วยให้หุ่นยนต์สามารถสร้างเส้นทางการเคลื่อนที่ งานวิจัยนี้เน้นผลักดันการทำงานของหุ่นยนต์เคลื่อนที่ด้วยล้อในออฟฟิต (การทำงานสิ่งแวดล้อมจริง) พบว่า หุ่นยนต์ Dervish สามารถสร้างเส้นทางเคลื่อนที่จากห้องทำงานหนึ่งผ่านห้องโถงแล้วไปยังห้องทำงานเป้าหมายได้ [17] นอกจากนั้นได้มีการศึกษาเทคนิคการตรวจจับและหลบหลีกสิ่งกีดขวาง ด้วยอุปกรณ์ส่งเลเซอร์เป็นแถบแสงแนวตั้ง แล้วใช้กล้องวิดีโอเป็นอุปกรณ์ตรวจรู้เพื่อบอกสภาพสิ่งแวดล้อม เนื่องจากกล้องวิดีโอไม่ได้ถูกออกแบบมาเพื่อวัดตำแหน่งในปริภูมิคาร์ทีเซียนของวัตถุในสิ่งแวดล้อมโดยตรง จึงมีการเสนอแนวทางหาตำแหน่งในปริภูมิคาร์ทีเซียนหลายวิธี หนึ่งในนั้นคือ การใช้กล้องสองตาตรวจจับวัตถุช่วยมนการสร้างเส้นทางเดินให้หุ่นยนต์ [3, 8] ถ้าหากไม่ใช้กล้องหลายตัว ก็ต้องใช้กล้อง monocular ช่วยในการหาตำแหน่งของวัตถุซึ่งผลลัพธ์คล้ายกับการใช้อุปกรณ์ตรวจรู้เลเซอร์วัดระยะ [18]

Microsoft Kinect เป็นอุปกรณ์ตรวจรู้ที่ประกอบด้วยกล้องดิจิตอลและกล้องตรวจจับความลึก (RGB-D) ซึ่งได้ผลลัพธ์เป็นข้อมูลภาพ 2 ชนิด คือภาพสี RGB และภาพแสดงความลึก [19] การใช้กล้องตรวจจับความลึกเริ่มมามีบทบาทช่วยในการหาตำแหน่งของหุ่นยนต์และสิ่งแวดล้อม เนื่องจาก

เป็นการหาตำแหน่งของสิ่งแวดล้อมในขอบเขตที่กว้างและได้ข้อมูลผลลัพธ์ปริมาณมากเป็นลักษณะข้อมูลรูปภาพ 2 มิติ การใช้กล้องตรวจจับความลึกจึงได้เป็นประเด็นการศึกษามากขึ้นเนื่องจากราคาที่ถูกลงและสะดวกต่อการติดตั้ง เมื่อเทียบกับอุปกรณ์ตรวจรู้ชนิดอื่นๆ เช่น อุปกรณ์ตรวจรู้เลเซอร์ สำหรับหุ่นยนต์เคลื่อนที่ด้วยล้อการรู้ตำแหน่งของตนเองในแผนที่ถือได้ว่าเป็นคุณสมบัติพื้นฐานที่สำคัญต่อการทำงานของหุ่นยนต์ จึงได้มีการเสนอเทคนิคในหาตำแหน่งของตัวเอง ทั้งการใช้กล้องติดตั้งแล้วตรวจจับหุ่นยนต์จากภายนอก [20] เป็นการใช้อุปกรณ์ตรวจรู้ Kinect ตรวจจับหาตำแหน่งของหุ่นยนต์และขอบเขตของสนามจากภายนอกสนาม หรือ การติดตั้งบนตัวหุ่นยนต์เลยด้วยเหตุนี้ทำให้มีการพัฒนาฟังก์ชันการทำงานของให้สามารถตรวจจับและจำแนกสิ่งแวดล้อม [21] ส่วนที่เป็นพื้น ฉาก กำแพง สิ่งกีดขวาง ออกจากกันได้ กล้องตรวจจับความลึกไม่ได้ถูกนำมาใช้เพียงกับหุ่นยนต์เคลื่อนที่ด้วยล้อเท่านั้น แต่ยังรวมถึงหุ่นยนต์ประเภทอื่นด้วย ตัวอย่างเช่น หุ่นยนต์ 7-Dof KUKA Light-Weight-Robot IV ซึ่งใช้กล้องตรวจจับความลึกในการหาตำแหน่งของแขนหุ่นยนต์ ผู้ใช้งานและสิ่งแวดล้อม ช่วยให้หุ่นยนต์สามารถหลีกเลี่ยงการปะทะกับผู้ใช้งานและสิ่งแวดล้อมโดยรอบได้ [22]

## 2.2 ความเป็นมาของการสอบเทียบกล้อง

โดนปกติแล้วภาพสีที่ได้จากกล้องมักจะมีคามบิดเบี้ยวเนื่องจากความนูนของเลนส์ ทำให้ข้อมูลภาพที่ได้นั้นมีความผิดพลาด จำเป็นที่จะต้องทำการสอบเทียบกล้องเพื่อให้ข้อมูลที่แม่นยำขึ้นเสมอ โดยทั่วไปการสอบเทียบกล้องที่ใช้มี 2 วิธี คือ 1) Photogrammetric calibration [23] คือการถ่ายภาพวัตถุที่ออกแบบพิเศษทำให้รู้ขนาดแน่นอนอยู่แล้วในทั้ง 3 มิติ ซึ่งวิธีการนี้จะได้ผลลัพธ์ที่มีความแม่นยำ แต่ก็ต้องแลกมาด้วยราคาอุปกรณ์ที่แพงและการติดตั้งที่ยุ่งยาก 2) Self-calibration [24] คือการถ่ายภาพฉากที่หลายๆมุม ซึ่งถ้าเราใช้กล้องตัวเดียวกันในการถ่ายจะรู้ได้ว่าภาพแต่ละภาพเกิดจากตัวแปลภายนอกกล้องมีค่าเท่ากัน ทำให้การถ่ายภาพอย่างน้อย 3 ภาพนั้นมากพอที่จะหาค่าตัวแปลทั้งภายในและภายนอกของกล้องได้ วิธีนี้มีความยืดหยุ่นสูงมาก แต่ผลที่ได้นั้นไม่ค่อยแม่นยำนัก วิธีการของ Zhengyou Zhang [25] นั้นเป็นวิธีการใหม่ ทำได้โดยการถ่ายภาพจากหลายๆมุม (อย่างน้อย 2 ภาพ) ของวัตถุสอบเทียบเป็นเพียงแค่ตารางหมากรุก โดยทั้งกล้องและวัตถุสอบเทียบสามารถกำหนดตำแหน่งได้อย่างอิสระ โดยไม่จำเป็นต้องเก็บบันทึกเส้นทางการเคลื่อนที่ของกล้อง ซึ่งภาพที่ถ่ายไว้นั้นโดยทั่วไปแล้วจะมีความบิดเบี้ยวอยู่ จากการทดสอบทั้งในระบบจำลองและใน

สิ่งแวดล้อมจริง ได้ผลลัพธ์ค่อนข้างแม่นยำเมื่อเทียบกับกระบวนการแบบดั้งเดิมซึ่งจะต้องใช้อุปกรณ์อย่างเช่น ระบายพิกัดฉากซึ่งมีราคาแพง วิธีที่นำเสนอนี้เป็นวิธีที่ใช้ง่ายและมีความยืดหยุ่นสูงกว่า

โดยทั่วไปแล้วกล้องดิจิทัลจะให้ผลลัพธ์เป็นภาพสี RGB เพื่อใช้ข้อมูลภาพสี RGB ในการค้นหาตำแหน่งเป้าหมายในปริภูมิภาพจำเป็นจะต้องแยกองค์ประกอบภาพที่เป็นวัตถุเป้าหมายออกจากส่วนอื่นของภาพเสียก่อน [26] โดยการกรองพื้นหลังออกด้วยปริภูมิสี RGB ซึ่งเป็นภาพขาว-ดำ 3 ชั้น ประกอบด้วยภาพความเข้มของแต่ละแม่สีแดง เขียวและน้ำเงิน การแยกแยะรายละเอียดด้วยปริภูมิสี RGB (RGB Color Space) นั้นมีข้อดีคือเมื่อความสว่างมีการเปลี่ยนแปลงอาจทำให้การตรวจจับนั้นผิดพลาดไป เมื่อเปรียบเทียบกระบวนการตรวจจับสี ด้วยปริภูมิสี HSV (HSV Color Space) แล้วปริภูมิสี HSV มีความเหมาะสมต่อสิ่งแวดล้อมจริง มากกว่าในกรณีที่มีความเข้มแสงไม่แน่นอนจากผลการทดลองใช้ในการตรวจจับป้ายสัญญาณบนถนน ณ ช่วงเวลาต่างๆ [27] ซึ่งเกิดขึ้นได้ง่ายสำหรับกล้องดิจิทัลเมื่อมีการเปลี่ยนแปลงมุมกล้องปริภูมิสี HSV ใช้หลักการแยกความสว่าง (Luma) และเนื้อสี (Chroma) ออกจากภาพออกจากจุดภาพ พบว่าปริภูมิ HSV ให้ผลลัพธ์ที่ชัดเจนกว่า เนื่องจากปริภูมิ HSV มีความถูกต้องของสีมากกว่าเมื่อแสงมีการเปลี่ยนแปลง ผลลัพธ์ที่จะเป็นภาพเทาขาว 1 มิติ ซึ่งบริเวณสีที่ต้องการกรองจะเป็นสีขาวกว่า การแบ่งขอบเขตสี [28] จะช่วยให้ภาพที่ได้เป็นเพียงสีขาวและดำ (ค่า 0 หรือ 1) [29] ใช้เทคนิคการวิเคราะห์แบบ Blob ในการหาตำแหน่งและขนาดของวัตถุเป้าหมาย [30]

### 2.3 การสร้างเส้นทางเดินของหุ่นยนต์

งานวิจัยนี้เป็นการพัฒนาเทคนิคการประมวลผลภาพที่ได้จากกล้อง RGB-D เพื่อสร้างเส้นทางเดินจากของหุ่นยนต์เคลื่อนที่ด้วยล้อจากปริภูมิรูปภาพ โดยประสานข้อมูลที่เป็นภาพสี RGB และภาพแสดงความลึก เพื่อแยกแยะองค์ประกอบของสิ่งแวดล้อม ค้นหาเป้าหมายและสิ่งกีดขวาง [31] โดยใช้ความสูงจากระดับพื้นที่ได้ของข้อมูลภาพแสดงความลึกในการแยกแยะประเภทองค์ประกอบภาพ แล้วใช้ภาพที่ได้จากกล้องสีในการช่วยหาตำแหน่งของเป้าหมาย โดยการหาตำแหน่งในปริภูมิภาพสีก่อนเพื่อเทียบไปยังตำแหน่งปริภูมิคาร์ทีเซียนในภาพความลึก

หุ่นยนต์เคลื่อนที่ด้วยล้อเป็นหุ่นยนต์ที่มีขอบเขตการทำงานไม่จำกัด ทำให้สิ่งแวดล้อมมีผลต่อการทำงานของหุ่นยนต์โดยตรง ดังนั้นการค้นหาตำแหน่งของหุ่นยนต์เคลื่อนที่ด้วยล้อจึงเป็นประเด็นที่สำคัญ หุ่นยนต์จำเป็นต้องรู้ตำแหน่งของตนเองและสิ่งต่างๆในสิ่งแวดล้อมที่ให้แน่นอน [32] การสร้างแผนที่เป็นวิธีหนึ่งที่ใช้ในการระบุตำแหน่ง โดยการนำค่าตำแหน่งที่เป็น 3 มิติ มาบันทึกลงในพิกัด 2 มิติ (x-y) เหตุที่ทำเช่นนี้ได้เป็นเพราะว่าหุ่นยนต์เคลื่อนที่ด้วยล้อนั้นเคลื่อนที่ในปริภูมิ 2 มิติอยู่แล้ว นั่นคือ หน้าหลัง-ซ้ายขวา

Rodney Brooks ได้นำเสนอแนวทางในการกำหนดพฤติกรรมการตอบสนองของหุ่นยนต์ นั่นคือการสร้างสนามศักย์ (Potential field) [33] การสนามศักย์ในการสร้างเส้นทางเดินถือเป็นแนวทางที่ใกล้เคียงกับพฤติกรรมตามธรรมชาติวิธีหนึ่ง [34] ข้อดีของสนามศักย์ก็คือ เป็นวิธีที่ตรงไปตรงมาทำให้ง่ายต่อการพัฒนาโปรแกรมและใช้เวลาในการคำนวณน้อย [35] โดยนิยามให้เป้าหมายเป็นขั้วสนามศักย์ส่งผลดึงดูดต่อหุ่นยนต์ ส่วนสิ่งกีดขวางให้มีขั้วสนามศักย์ส่งผลผลักดันต่อหุ่นยนต์ [36] เปรียบเสมือนเป็นตำแหน่งกำเนิดแรงออกมาในลักษณะเวกเตอร์ โดยเวกเตอร์นี้จะมีผลต่อการเคลื่อนที่ของหุ่นยนต์

## บทที่ 3

### ทฤษฎีที่เกี่ยวข้อง

#### 3.1 แบบจำลองของกล้อง, เลนส์และการประมวลผลรูป

หลักการการทำงานของกล้องวิดีโอ คือ การบันทึกข้อมูลสิ่งแวดล้อมที่ลักษณะเป็น 3 มิติ ให้เป็นภาพ 2 มิติลงบนฉากรับภาพ ซึ่งเป็นความสัมพันธ์ระหว่างภาพของวัตถุที่ฉายบนระนาบกล้องกับตำแหน่งของวัตถุที่แกนวัดจริง หรือเรียกอีกอย่างหนึ่งว่า ภาพฉายแบบมุมตึบ (Perspective projection) จะทำให้สูญเสียมิติความลึกไปหนึ่งมิติ ทุกๆจุดในภาพ 2 มิตินั้นเกิดจากรังสีแสงที่เป็นเส้นตรงออกมาจากตำแหน่งจริงใน 3 มิติ ผ่านรูเข็มของกล้องมายังฉากรับภาพ (Projective plane) [37] มีความสำคัญต่อการประมวลผลภาพอย่างมาก (Image processing) [24, 38-40]

##### 3.1.1 แบบจำลองกล้อง

จากหลักการของกล้องรูเข็ม ซึ่งภาพที่ปรากฏจะเป็นภาพจริงหัวกลับ ภาพจริงที่ปรากฏขึ้นนั้นแสดงดังรูปที่ 1 จากรูปแสดงความสัมพันธ์ระหว่างวัตถุจริงและภาพที่เกิดจากการฉายลำแสงฉายลงบนฉากรับภาพ จากสมบัติความสัมพันธ์สามเหลี่ยมคล้ายจะให้ความสัมพันธ์ดังนี้

กำหนดให้จุด  ${}^cP = [x \ y \ z]^T$  เป็นตำแหน่งใดๆของวัตถุจริงเทียบกับแกนมุขสำคัญ (Principal axis) และกำหนดให้ระยะจากจุดตัดแกนมุขสำคัญที่  $z_c = f$

เมื่อพิจารณาความสัมพันธ์สามเหลี่ยมคล้ายดัง ลำแสงที่ฉายจากจุด  $(x_c, y_c, z_c)$  ไปยังระนาบภาพ เมื่อนำจุดของภาพทั้งสองมาสร้างเป็นสามเหลี่ยมสามารถพิจารณาความสัมพันธ์สามเหลี่ยมคล้ายได้ดังนี้

$$\frac{f}{Z} = \frac{u}{X} = \frac{v}{Y}$$

$$u = \frac{fX}{Z}$$

$$v = \frac{fY}{Z}$$

นำสมการทั้งหมดมาระบบพิกัดเอกพันธ์ (Homogeneous coordinate) ของจุด  ${}^cP$  สามารถเขียนใหม่ได้เป็น

$$\begin{bmatrix} u \\ v \\ W \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

แต่ในความเป็นจริงเป็นไปได้ที่แกน  $Z$  นั้นไม่ได้อยู่ตรงกับเส้นแกนमुखสำคัญเสมอ จะต้องมีการระยะห่างออกจากเส้นแกนमुखสำคัญเสมอ ทำให้ต้องเพิ่มตัวแปลเข้าไปอีกสองตัวนั่นคือ  $c_x$  และ  $c_y$  เป็นระยะห่างแกน 3 มิติจากเส้นแกนमुखสำคัญ [30] จะทำให้เราได้ความสัมพันธ์ระหว่างจุด  $P$  กับจุด  $p$  คือ

$$u = f_x \frac{X}{Z} + c_x, \quad v = f_y \frac{Y}{Z} + c_y$$

ให้รู้ไว้ว่าตอนนี้เรามี ค่าความยาวโฟกัส 2 ค่า ที่เป็นเช่นนี้เป็นเพราะว่า ภาพโดยทั่วไปพิกเซล จะไม่เป็นสี่เหลี่ยมจัตุรัส ความยาวโฟกัสทั้งสองนี้เกิดจากการผลคูณของความยาวโฟกัสของเลน  $f_x$  กับขนาดพิกเซลของภาพ  $m_x$  และ  $f_y$  ก็มี  $m_y$  เช่นกัน ซึ่งค่า  $m_x$  และ  $m_y$  นั้นไม่สามารถวัดได้โดยตรง นำมารวมเป็นสมการ  $f_x = f m_x$  และ  $f_y = f m_y$

ความสัมพันธ์ระหว่างจุด  $P (X,Y,Z)$  ในโลกจริงกับจุด  $p (x,y)$  ในภาพเรียกว่า “projective transform” ซึ่งเป็น homogeneous coordinate จึงเพิ่มมิติจาก จาก  $x,y$  เป็น  $x,y,w$  จะได้เมทริกซ์ความสัมพันธ์ ระหว่างจุด  $P$  และจุด  $p$  ดังนี้

$$p = KP$$

$$\begin{bmatrix} u \\ v \\ W \end{bmatrix} = \begin{bmatrix} m_x f & s & c_x \\ 0 & m_y f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

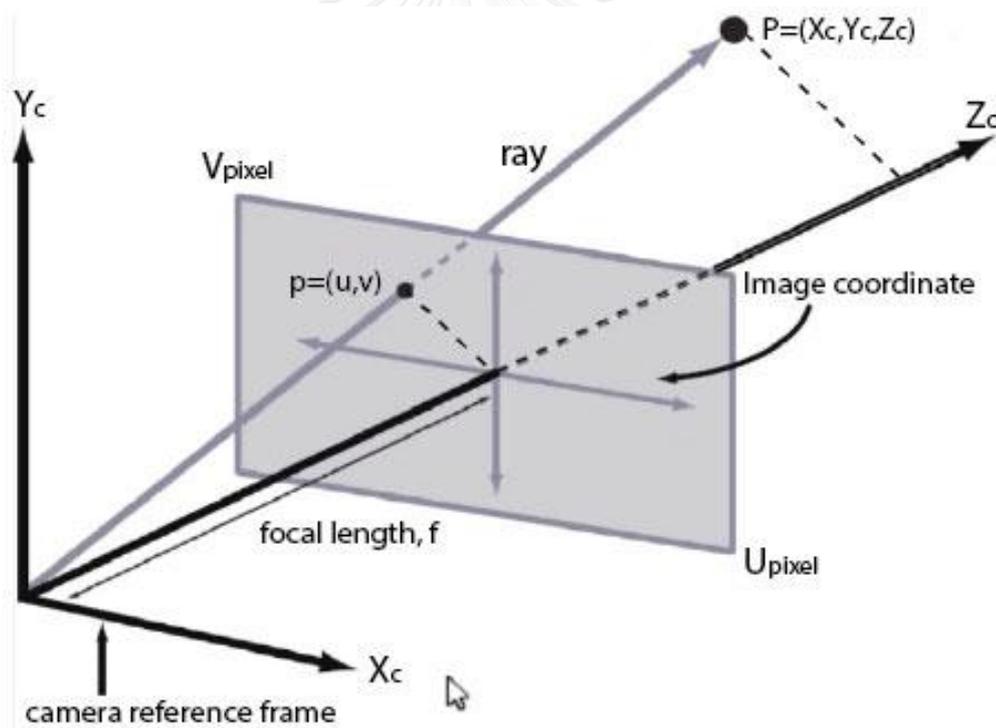
$K$  เรียกว่า เมทริกซ์การสอบเทียบกล้อง (camera calibration matrix) เป็นเมทริกซ์ขนาด  $3 \times 3$  ที่มีความสัมพันธ์โดยตรงกับพารามิเตอร์ภายในทั้ง 5 ตัวของกล้อง โดยที่

$$K = \begin{bmatrix} m_x f & s & c_x \\ 0 & m_y f & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

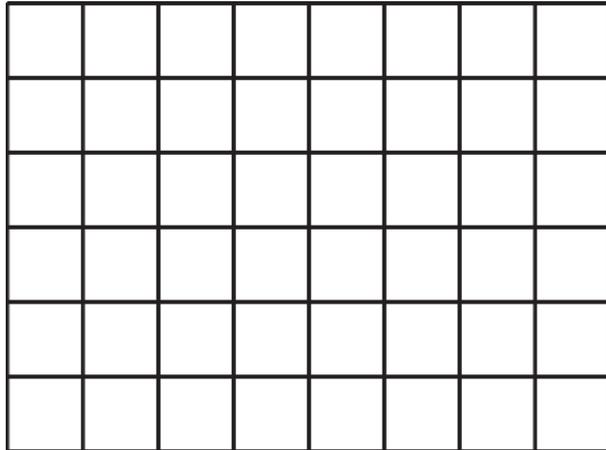
$m_x, m_y$  คือ ค่าคงที่ของเลนส์และความละเอียดของอุปกรณ์รับรู้ภาพ

$c_x, c_y$  คือ ตำแหน่งจุดศูนย์กลางของภาพซึ่งคือจุดกำเนิดของแกนอ้างอิงภาพ และเป็นจุดตัดของแกนमुखสำคัญของเลนส์ (optical axis) กับระนาบรับภาพ

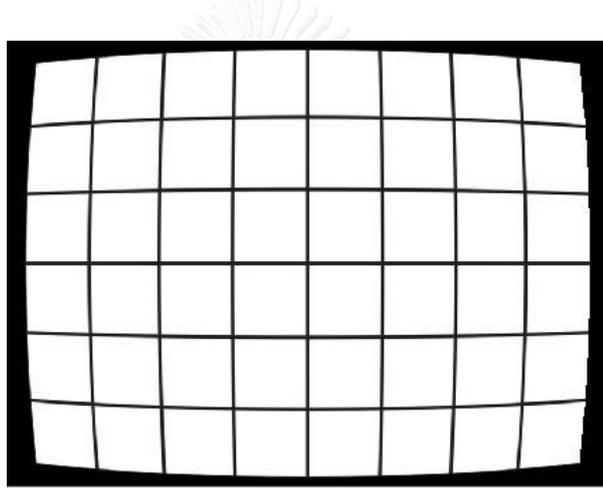
$s$  คือ ค่าความเข้ของกล้อง (Skew parameter) กล้องถ่ายภาพโดยทั่วไปแล้ว จะมีค่าความเข้มีค่าเท่ากับศูนย์ ถ้าหากว่าค่าตัวแปรนี้ไม่เท่ากับศูนย์นั้นหมายความว่า ตำแหน่งเลนส์และเซนเซอร์รับภาพไม่ตั้งฉากกัน ในกรณีนี้มักจะเกิดจากความผิดพลาดในการผลิต



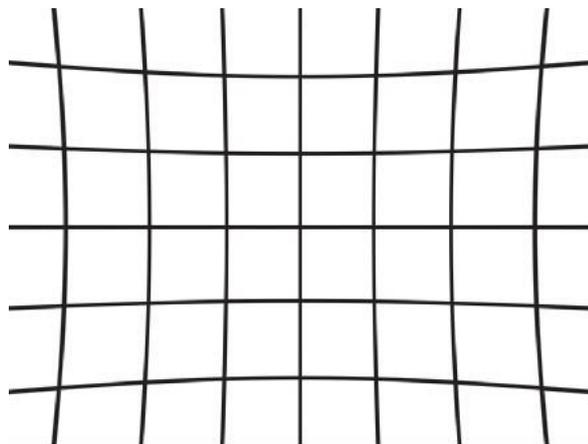
รูปที่ 1 แบบจำลองกล้องรูเข็มและแกนอ้างอิงที่เกี่ยวข้อง



ก. ภาพปกติ



ข. ภาพโป่งกลาง



ค. ภาพตึบกลาง

รูปที่ 2 ภาพการบิดเบี้ยวของเลนส์

### 3.1.2 การบิดเบี้ยวของเลนส์

ปกติแล้วภาวะที่ได้จากกล้องจะมีการบิดเบี้ยวของเลนส์เสมอเนื่องจากการความโค้งของตัวเลนส์เอง โดยทั่วไปแล้วค่าการบิดเบี้ยวของเลนส์จะมีค่าผกผันกับตัวเลนส์ หมายความว่าถ้าความยาวโฟกัสมากจะมีค่าการบิดเบี้ยวของเลนส์น้อย และถ้าความยาวโฟกัสน้อยจะมีค่าการบิดเบี้ยวของเลนส์มาก

การบิดเบี้ยวของเลนส์ที่มักจะมีสองประเภทได้แก่ การบิดเบี้ยวเชิงรัศมี ซึ่งเกิดจากรูปทรงที่ไม่สมบูรณ์แบบของเลนส์ และการบิดเบี้ยวเชิงสัมผัส เกิดความผิดพลาดจากการวางตำแหน่งของเลนส์และฉากรับภาพ จากงานวิจัย [30, 31] พบว่าเฉพาะการบิดเบี้ยวเชิงแนวรัศมีที่จำเป็นต้องแก้ไข ส่วนบิดเบี้ยวเชิงเส้นสัมผัสมีผลน้อยมากจึงไม่มีผลต่อการคำนวณ

การบิดเบี้ยวเชิงรัศมี จะทำให้เกิดปรากฏการณ์ดังรูปที่ 2 ความบิดเบี้ยวเชิงรัศมีที่เส้นแกนมุขสำคัญนั้นจะเท่ากับ 0 ( $r = 0$ ) และจะมากขึ้นเมื่อรัศมีมีค่ามากขึ้นตามอนุกรมเทลเลอร์ ทำให้เราสามารถสร้างสมการได้ดังนี้

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$(x, y)$  คือตำแหน่งของภาพก่อนการแก้ไขการบิดเบี้ยว  $(x_{corrected}, y_{corrected})$  คือตำแหน่งของภาพหลังการแก้ไขการบิดเบี้ยว

การบิดเบี้ยวเชิงสัมผัส มีสาเหตุจากการประกอบกล้องที่ทำให้ระนาบของเลนส์ไม่ขนานกับระนาบของฉากรับภาพพอดี ดังรูปที่ 2 ซึ่งมีแบบจำลองทางคณิตศาสตร์ดังนี้

$$x_{corrected} = x + [2p_1 y + p_2 (r^2 + 2x^2)]$$

$$y_{corrected} = y + [p_1 (r^2 + 2y^2) + 2p_2 x]$$

จากการบิดเบี้ยวทั้ง 2 แบบทำให้เราได้ตัวแปรที่ไม่ทราบทั้งหมด 5 ตัวแปร สามารถสร้างเป็นเวกเตอร์การบิดเบี้ยว ที่มีสมาชิก คือ  $k_1, p_1, k_2, p_2, k_3$

### 3.1.3 การสอบเทียบกล้อง

การสอบเทียบกล้องเป็นกระบวนการที่สำคัญสำหรับระบบวิสัยทัศน์ของคอมพิวเตอร์ (Computer Vision) เพื่อที่หาค่าตัวแปรในสามมิติ จากภาพถ่ายซึ่งมีแค่สองมิติ ซึ่งมีการค้นคว้าวิจัยมากมาย สามารถแยกกระบวนการสอบเทียบกล้องได้ออกเป็น 2 ประเภทหลัก ดังนี้ [28]

ประเภทที่ 1 Photogrammetric calibration วิธีนี้การสอบเทียบกล้องทำได้โดยการ ใช้กล้องถ่ายวัตถุสอบเทียบซึ่งเป็นวัตถุทรงเรขาคณิตที่มีขนาดที่แน่นอนทั้งสามมิติ การสอบเทียบด้วยวิธีนี้มีความแม่นยำสูงมาก [28] โดยวัตถุที่ใช้มักจะมีฉากที่ตั้งฉากซึ่งกันและกันในแต่ละแกนมีพิกัดสามมิติ แต่อย่างไรก็ตามวิธีนี้ต้องใช้เครื่องมือที่มีราคาแพงและมีการติดตั้งที่ยุ่งยาก

ประเภทที่ 2 Self-calibration. เทคนิคนี้ไม่จำเป็นต้องใช้วัตถุสอบเทียบ แต่ต้องใช้การถ่ายภาพในสิ่งแวดล้อมที่คงที่ โดยความคงที่นี้จะทำให้ทำให้ตัวแปรหลักๆของกล้องนั้นคงที่ [39] ตัวแปรภายในของกล้องจะถูกแทนที่ด้วยข้อมูลที่ได้จากภาพ หมายความว่าถ้าภาพทั้งหมดที่ถูกถ่ายด้วยกล้องตัวเดียวกันจะทำให้ตัวแปรภายในนั้นเท่ากัน วิธีนี้ต้องการภาพถ่ายอย่างน้อยสามภาพเพื่อทดแทนตัวแปรทั้งภายในและภายนอก ซึ่งทำให้สามารถสร้างโครงสร้างสิ่งแวดล้อมแล้วเทียบเป็นสามมิติได้ [38] กระบวนการนี้ถึงแม้ว่าจะเป็นวิธีที่ยืดหยุ่น และผลที่ได้นั้นไม่ครบถ้วน [41] เพราะมีตัวแปรหลายค่ามาจากการประมาณค่าทำให้ผลอาจจะไม่ค่อยน่าเชื่อถือ

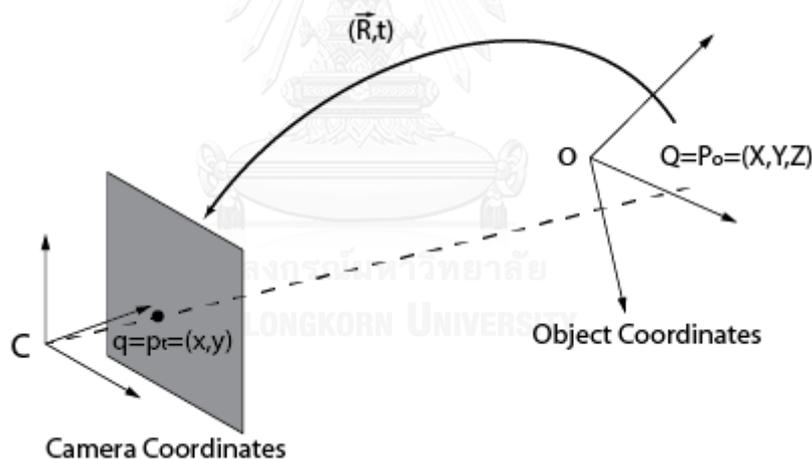
ในงานวิจัยนี้ได้ใช้เทคนิคที่ถูกเสนอโดย Zhang [25] เทคนิคนี้จำเป็นต้องใช้กล้องถ่ายรูปในมุมมองต่างๆ ถ่ายไปยังวัตถุสอบเทียบนั้นคือตารางหมากรุก ซึ่งเป็นวัตถุสอบเทียบที่เรียบง่าย การจัดตำแหน่งกล้องนั้นสามารถจัดได้ด้วยมือไม่จำเป็นต้องมีเทคนิคพิเศษและไม่จำเป็นต้องบันทึกตำแหน่งของกล้อง เทคนิคนี้ได้ใช้กระบวนการของทั้งสองเทคนิคที่กล่าวมาข้างต้นนั้นคือ photogrammetric calibration และ self-calibration เพราะใช้วัตถุสอบเทียบที่มีขนาดที่แน่นอนในสองมิติแทนสามมิติ ซึ่งผลลัพธ์ที่ได้นั้นเป็นที่น่าพอใจ, เรียบง่ายและราคาถูกกว่ามากเมื่อเทียบกับเทคนิค Photogrammetric calibration นอกจากนั้นเมื่อเทียบกับ self-calibration นับเป็นวิธีที่มีความแม่นยำสูงกว่า

ตามรูปที่ 3 เทคนิคการสอบเทียบกล้องทำได้โดยการถ่ายภาพจากกล้องหลายๆมุม และคำนวณหาตำแหน่งและมุมของกล้องในการถ่ายภาพแต่ละครั้ง โดยทั่วไปแล้วการอธิบายว่ากล้องนั้นหมุนไปอย่างไรบ้างนั้น สามารถเขียนได้หลายรูปแบบ ซึ่งการหมุนที่ถูกเลือกใช้ในที่นี่คือ หมุนตามแกน  $x$ ,  $y$  และ  $z$  ตามลำดับ

$$R = R_z(\theta)R_y(\varphi)R_x(\psi)$$

ทรานสเลท เวกเตอร์ (Translation vector) คือเวกเตอร์ที่บ่งบอกถึงการเลื่อนของแกนพิกัดจากจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่ง สามารถหาได้ดังนี้  $T = origin_{object} - origin_{camera}$  ดังนั้นเมื่อประกอบเข้ากับเมทริกการหมุนจะหาความสัมพันธ์ของจุดบนแกนของวัตถุ เทียบกับแกนของกล้องจะได้ดังนี้

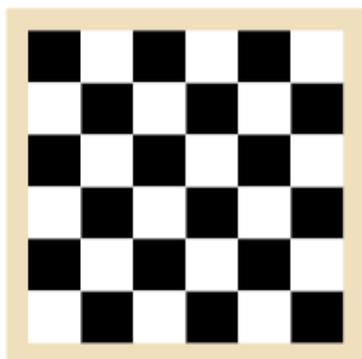
$$P_c = R(P_o - T)$$



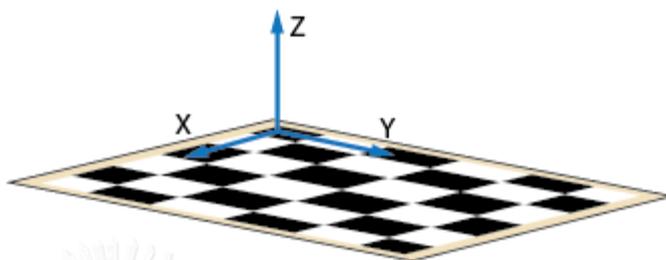
รูปที่ 3 การฉายของตำแหน่งจุด  $P$  ในปริภูมิ 3 มิติไปยังจุด  $p$  บนปริภูมิภาพ

จะเห็นได้ว่าการหา ทรานฟอเมชันเมทริก (Transformation Matrix) นี้มีตัวแปรที่ไม่ทราบอยู่ 6 ตัวแปร ประกอบกับตัวแปรของการสอบเทียบกล้องอีก 4 ตัว แต่ตัวแปรภายในเหล่านี้มีค่าเท่ากันในทุกภาพ จึงจำเป็นต้องถ่ายภาพอย่างน้อย 2 ภาพ เพื่อชดเชยข้อมูลรูปภาพให้เพียงพอ จึงต้องใช้วัตถุสอบเทียบที่มีความสำคัญและเหมาะสมกับการสอบเทียบกล้อง วัตถุสอบเทียบที่ใช้คือ ตารางหมากรุก ดังรูปที่ 4 เหตุที่ใช้ตารางหมากรุกเป็นเพราะว่า ประการที่หนึ่ง ตารางหมากรุกเป็น

วัตถุ 2 มิติ มีขนาดเพียงแค่ 2 แกนในระนาบ  $x$  และ  $y$  (แกน  $z = 0$ ) ทำให้ง่ายต่อการคำนวณ ประการที่สอง สามารถกำหนดขนาดของช่องที่แน่นอนได้ ประการที่สาม ตารางหมากรุกเป็นสี่เหลี่ยม ให้เราได้ข้อมูลจากมุมถึง 8 ค่า (ค่าในแกน  $x$  และแกน  $y$  อย่างละ 4 ค่า) จากถ่ายภาพแต่ละครั้ง



ก. กระดานสอเทียบ



ข. แกนอ้างอิงวัตถุสอเทียบ

รูปที่ 4 วัตถุสอเทียบ

### 3.1.4 การแปลงระนาบโฮโมกราฟฟี

ในการแปลงพิกัดบนระนาบภาพจากระนาบหนึ่งไปยังอีกระนาบหนึ่ง จำเป็นต้องทำการแปลงระนาบโฮโมกราฟฟี (Homography transformation) [42, 43] จะมีโฮโมกราฟฟีเมทริกซ์ เป็นเมทริกซ์เก็บตัวแปลงระหว่างจุดบนระนาบใดๆและระบบพิกัดที่กำลังหาจะได้ความสัมพันธ์ของระนาบเขียนเป็นสมการได้ดังนี้

$$p' = H \cdot p$$

โดยที่  $P$  คือ จุดที่อยู่บนระนาบจริง

$P'$  คือ จุดที่อยู่บนระนาบภาพ

$H$  คือ โฮโมกราฟฟีเมทริกซ์

โดยสามารถเขียนเป็นสมการได้ดังนี้

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

จากรูปที่ 5 จุดใดๆที่อยู่ในภาพหนึ่งจะต้องสามารถฉายไปยังจุดเดียวกันระนาบของอีกภาพหนึ่งได้ ปกติแล้วแล้ว จุดสี่จุดที่ปรากฏในภาพบนระนาบจะสามารถหาความสัมพันธ์เพื่อฉายไปหาอีกระนาบได้ ซึ่งความสัมพันธ์ของเส้นตรงที่อยู่บนระนาบจะได้สมการ

$$X = \frac{ax + by + c}{gx + hy + 1} \quad (2)$$

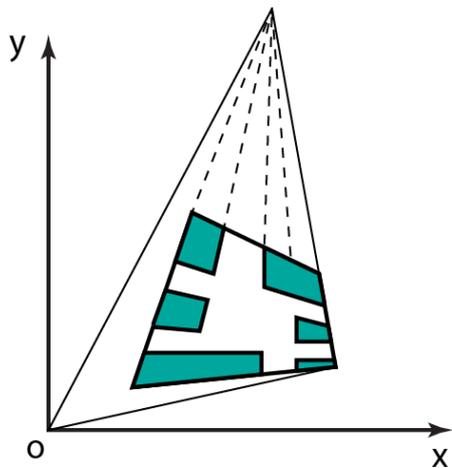
$$Y = \frac{dx + ey + f}{gx + hy + 1} \quad (3)$$

จากสมการที่ 2 และ 3 จะเห็นได้ว่ามีตัวแปรไม่ทราบค่าอยู่ 8 คำนั่นคือ  $a, b, c, d, e, f, g, h$  เพื่อแก้สมการหาค่าของตัวแปรจำเป็นจะต้องใช้จุดอย่างน้อย 4 จุด จากสมการขั้นต้น สามารถเขียนเมทริกซ์ความสัมพันธ์ได้ดังนี้

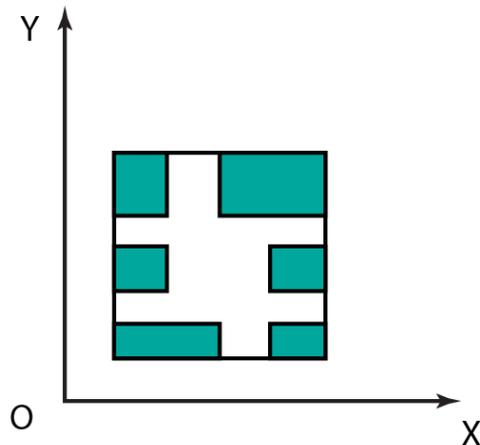
$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & x_1 X_1 & y_1 X_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & x_2 X_2 & y_2 X_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & x_3 X_3 & y_3 X_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & x_4 X_4 & y_4 X_4 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & x_1 Y_1 & y_1 Y_1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & x_2 Y_2 & y_2 Y_2 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & x_3 Y_3 & y_3 Y_3 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & x_4 Y_4 & y_4 Y_4 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix}$$

โดยที่

- a คือ ค่าสัดส่วนตัวคูณคงที่ในแกน X เมื่อสัดส่วนในแกน Y ไม่เปลี่ยนแปลง
- b คือ ค่าสัดส่วนตัวคูณในทิศทาง X โดยที่เป็นสัดส่วนกับระยะห่างจากจุด (0,0) ในทิศทาง X
- c คือ ค่าระยะห่างจากจุด (0,0) ไปในทิศทาง X
- d คือ ค่าสัดส่วนตัวคูณในทิศทาง Y โดยที่เป็นสัดส่วนกับระยะห่างจากจุด (0,0) ในทิศทาง X
- e คือ ค่าสัดส่วนตัวคูณคงที่ในแกน Y เมื่อสัดส่วนในแกน X ไม่เปลี่ยนแปลง
- f คือ ค่าระยะห่างจากจุด (0,0) ไปในทิศทาง Y
- g คือ ค่าสัดส่วนตัวคูณ X และ Y ที่เป็นฟังก์ชันของ X
- h คือ ค่าสัดส่วนตัวคูณ X และ Y ที่เป็นฟังก์ชันของ Y

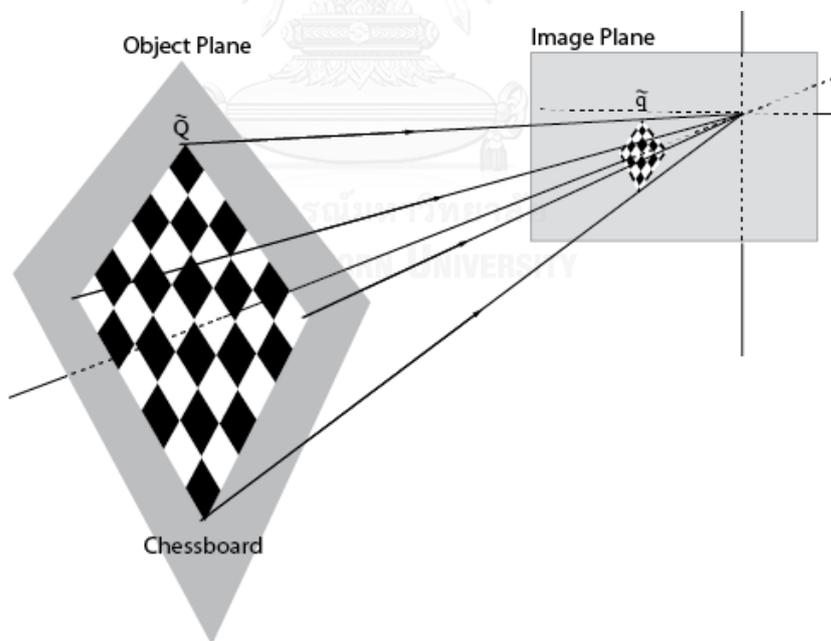


ก. ภาพที่เกิดจากการฉาย



ข. ลักษณะของภาพวัตถุจริง

รูปที่ 5 การฉายภาพมุมตีบไปยังระนาบโดยใช้วิธี โฮโมกราฟฟี



รูปที่ 6 การเทียบตำแหน่งจุดที่มุมของตารางหมากรุก จากในปริภูมิ 3 มิติไปยังปริภูมิรูปภาพ

### 3.1.5 การหาตัวแปรภายในและภายนอกของกล้อง

เทียบจุดต่างๆในระนาบรูปภาพกลับไปเพื่อหาตำแหน่งในระนาบวัตถุจากรูปดังรูป 6 จะได้เขียนเป็นสมการความสัมพันธ์ของระนาบได้ดังนี้  $\tilde{Q} = [X \ Y \ Z \ 1]^T$  และ  $\tilde{q} = [x \ y \ 1]^T$  ทำ

ให้ได้ความสัมพันธ์ดังสมการ 
$$\tilde{q} = sH\tilde{Q}$$

$s$  คือ ค่าสัดส่วนของขนาด มีไว้เทียบอัตราส่วนระหว่างหน่วยพิกเซลกับหน่วยเมตร

$H$  เป็นเมทริกซ์ ที่ประกอบด้วยตัวแปร 2 ส่วนได้แก่ เมทริกซ์การแปลงค่าตำแหน่ง (Transformation Matrix) หรือเมทริกซ์ภายนอก (Extrinsic Matrix) และ เมทริกซ์เก็บตัวแปรภายใน (Intrinsic Matrix) ทำให้เราสามารถเขียนใหม่ได้ดังนี้

การหาเมทริกซ์เก็บตัวแปรภายใน (intrinsic matrix)

$$\tilde{q} = sMW\tilde{Q}$$

$$M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad \text{และ} \quad W = [r_1 \ r_2 \ r_3 \ t]$$

$$\tilde{q} = s \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \times [r_1 \ r_2 \ r_3 \ t] \times \tilde{Q}$$

เนื่องจาก เราใช้แผ่นกระดาษตารางหมากรุก เป็นวัตถุสอบเทียบ ทำให้มีค่าในแกน  $Z = 0$  ทำให้สามารถเขียน สมการความสัมพันธ์ใหม่ได้ดังนี้

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = sM \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = sM \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

จากสมการข้างต้น สมมติให้  $H = [h_1 \ h_2 \ h_3] = sM [r_1 \ r_2 \ t]$  จะได้

$$h_1 = sMr_1 \text{ หรือ } r_1 = \lambda M^{-1}h_1$$

$$h_2 = sMr_2 \text{ หรือ } r_2 = \lambda M^{-1}h_2$$

$$h_3 = sMr_3 \text{ หรือ } r_3 = \lambda M^{-1}h_3$$

โดยที่  $\lambda = 1/s$  และ  $h$  เป็นเมทริก 3x1

เนื่องจากเวกเตอร์การหมุนมีสมบัติ orthogonal ต่อกันและกัน ทำให้ได้ความสัมพันธ์

$$r_1^T r_2 = 0$$

และจากสมบัติของเวกเตอร์  $(ab)^T = b^T a^T$  ใช้กับ  $r_1$  ส่วน  $r_2$  ให้เป็นค่าเท่าเดิม

$$h_1^T M^{-T} M^{-1} h_2 = 0$$

รู้ว่าขนาดของเวกเตอร์การหมุนนั้นเท่ากัน

$$\|r_1\| = \|r_2\| \text{ หรือ } r_1^T r_1 = r_2^T r_2$$

แทนค่า  $r_1, r_2$  ลงไปได้

$$h_1^T M^{-T} M^{-1} h_1 = h_2^T M^{-T} M^{-1} h_2$$

$$B = M^{-T} M^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix}$$

ให้

$$B = \begin{bmatrix} \frac{1}{f_x^2} & 0 & \frac{-c_x}{f_x^2} \\ 0 & \frac{1}{f_y^2} & \frac{-c_y}{f_y^2} \\ \frac{-c_x}{f_x^2} & \frac{-c_y}{f_y^2} & \frac{c_x}{f_x^2} + \frac{c_y}{f_y^2} + 1 \end{bmatrix}$$

จะเห็นว่าเมตริกซ์ B เป็นเมตริกซ์ที่แยงเมื่อเรานำมาแทนใน  $h_i^T B h_j$  แล้วเทียบกับสมาชิกในเวกเตอร์ที่ไม่ซ้ำกัน จะสามารถเขียนผลของการ dot product ของเวกเตอร์ขนาด  $1 \times 6$  ได้เวกเตอร์ใหม่ดังนี้

$$h_i^T B h_j = v_{ij}^T b = \begin{bmatrix} h_{i1}h_{j1} \\ h_{i1}h_{j2} + h_{i2}h_{j1} \\ h_{i2}h_{j2} \\ h_{i1}h_{j3} + h_{i3}h_{j1} \\ h_{i2}h_{j3} + h_{i3}h_{j2} \\ h_{i3}h_{j3} \end{bmatrix}^T \begin{bmatrix} B_{11} \\ B_{12} \\ B_{22} \\ B_{13} \\ B_{23} \\ B_{33} \end{bmatrix}^T$$

จากสมบัติของเวกเตอร์ที่กล่าวมาข้างต้นจะสามารถเขียน  $v_{ij}^T$  ให้ได้ดังนี้

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0$$

ถ้าเราเก็บภาพถ่ายของตารางหมากรุก จำนวนตั้งแต่ 2 ภาพขึ้นไปจะทำให้หาค่าตัวแปรต่างๆได้ดังนี้

$$\begin{aligned} f_x &= \sqrt{\lambda / B_{11}} \\ f_y &= \sqrt{\lambda B_{11} / (B_{11}B_{22} - B_{12}^2)} \\ c_x &= -B_{13}f_x^2 / \lambda \\ c_y &= (B_{12}B_{13} - B_{11}B_{23}) / (B_{11}B_{22} - B_{12}^2) \\ \text{และ } \lambda &= B_{33} - (B_{13}^2 + c_y(B_{12}B_{13} - B_{11}B_{23})) / B_{11} \end{aligned}$$

เมทริกซ์การแปลงค่าตำแหน่ง (Transformation Matrix) สามารถหาได้ดังนี้

$$\begin{aligned} r_1 &= \lambda M^{-1}h_1 \\ r_2 &= \lambda M^{-1}h_2 \\ r_3 &= r_1 \times r_2 \\ t &= \lambda M^{-1}h_3 \end{aligned}$$

### 3.1.6 ปริภูมิสี

ปริภูมิสี RGB เป็นปริภูมิสีที่ใช้กันอย่างแพร่หลายในระบบภาพดิจิทัล โดยแทนค่าของสีในแต่ละจุด เป็นสีในปฐมภูมิ คือ สีแดง สีเขียว และสีน้ำเงิน ดังรูปที่ 7 จากภาพแกนแต่ละแกนคือค่าของสีแดง สีเขียว และสีน้ำเงิน โดยมีช่วงของความเข้มของค่าสีระหว่าง 0-1 ซึ่งปริภูมิสี RGB จะมีลักษณะเป็นรูปลูกบาศก์

ปริภูมิสี HSV ใช้หลักการแยกความสว่าง(Luma)และ เนื้อสี(Chroma)ออกจากภาพออกจากจุดภาพ โดยปริภูมิสี HSV จะประกอบด้วย H แทนค่าเนื้อสี (Hue) [27] ซึ่งจะมีค่าบอกตำแหน่งของเนื้อสีอยู่ในช่วง0-1หรือบางกรณีก็จะใช้ช่วงองศา 0-360, S แทนค่าความสดของสี(Saturation) และ V แทนค่าความสว่างของแสง (Value) ซึ่งปริภูมิสี HSV จะมีลักษณะเป็นทรงกรวย ดังรูปที่ 7

โดยปกติแล้วกล้องถ่ายภาพดิจิทัลจะให้ผลลัพธ์ที่เป็นระบบปริภูมิสี RGB ซึ่งจะต้องทำการคัดเลือกช่วงที่ต้องการแยกโดยการปรับค่าตัวแปรสีถึง 3 ค่า เพราะภาพสี RBG เกิดจากการภาพของแม่สีทั้ง 3 ภาพ ซ้อนกันอยู่ และเป็นกระบวนการผสมสี เป็นเรื่องดีที่เปลี่ยนจากระบบปริภูมิสี RGB เป็น HSV เพราะทำให้ลดจำนวนภาพที่ซ้อนกันมารวมเป็นภาพเดียว ตัวแปรในการปรับช่วงของสีที่ต้องการจึงลดเหลือ 2 ค่าหลัก [44] ซึ่งมีข้อได้เปรียบเมื่อมีการเปลี่ยนแปลงของแสงในสิ่งแวดล้อม

การแปลงค่าของปริภูมิสี RGB มาเป็นปริภูมิสี HSV ทำได้ตามสมการนี้

$$H' = \begin{cases} \text{notdefined, if } C = 0 \\ 60 \left[ \frac{G - B}{\delta} \bmod 6 \right], \text{ if } M = R \\ 60 \left[ \frac{B - R}{\delta} + 2 \right], \text{ if } M = G \\ 60 \left[ \frac{R - G}{\delta} + 4 \right], \text{ if } M = B \end{cases}$$

$$H = 60^\circ \times H'$$

$$s = \begin{cases} 0, \text{ if } c = 0 \\ \frac{C}{V}, \text{ otherwise} \end{cases}$$

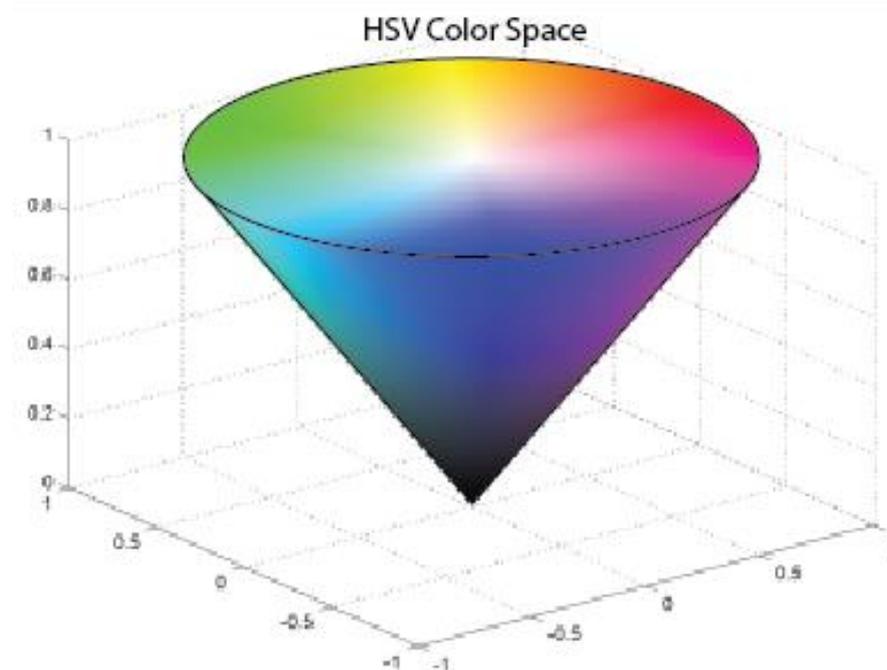
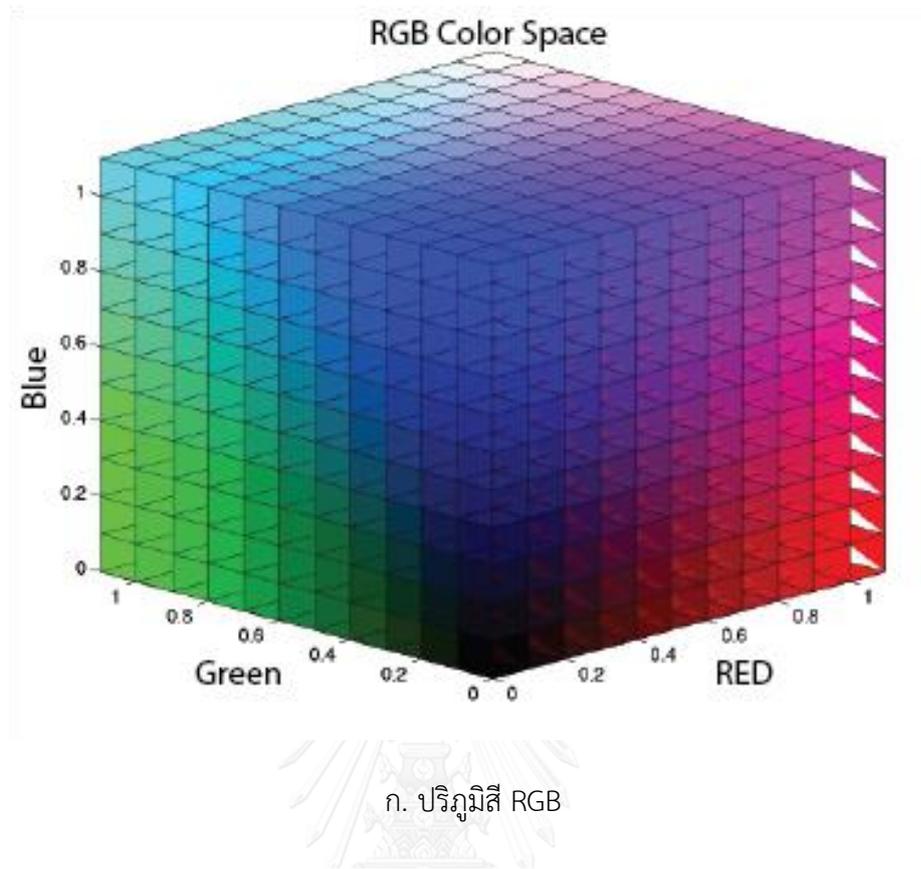
$$V = M$$

โดยที่

$$M = \max(R, G, B)$$

$$m = \min(R, G, B)$$

$$C = M - m$$



รูปที่ 7 ปริภูมิสีแบบต่างๆ

### 3.1.7 การคัดกรองรายละเอียดของภาพ

เทคนิคการตรวจจับโดยใช้สีของวัตถุแยกออกจากส่วนอื่นที่เป็นพื้นหลัง [26] ภาพที่ได้จากกล้องดิจิทัลเป็นระบบปริภูมิสี RGB ดังรูป 8ก แต่เมื่อแปลเป็นระบบปริภูมิสี HSV ทำให้สามารถเลือกช่วงโทนสีเป็นพิกัดองศาและความสดของสีในช่วง 0 ถึง 1 จากนั้นพิกเซลของภาพที่อยู่ในช่วงที่ต้องการมีค่า (เป็นสีเทาหรือสีขาว) และให้พิกเซลของภาพที่อยู่ในช่วงที่ไม่ต้องการไม่มีค่าหรือเท่ากับ 0 (เป็นสีดำ) ดังรูปที่ 8ข จากนั้นทำการปิดพิกเซลส่วนที่มีค่าน้อยมากๆ ให้เป็น 0 (เป็นสีดำ) และส่วนอื่นที่เหลือมีค่าเป็น 1 เป็นสีขาว [28] ดังแสดงในรูป 8ค

### 3.1.8 การคำนวณหาตำแหน่งของวัตถุในพิกัดสามมิติ

การตรวจจับของ Blob เป็นแบบจำลองที่ใช้ตรวจจับขอบเขตภาพในบริเวณที่เป็นสีดำหรือสีขาวมากกว่าบริเวณรอบข้าง ซึ่งจะทำให้ได้ข้อมูลที่สำคัญต่างๆ เช่น อาณาเขต พื้นที่ ตำแหน่ง ที่บริเวณที่สนใจ โดยภาพที่ได้รับมาจากกล้องดิจิทัลจะถูกแปลงทำการแยกสีแล้วแปลงเป็นภาพขาวดำ จากนั้น Blob detector นั้นจะทำงานได้ดีเมื่อเนื้อสีของวัตถุที่สนใจนั้นมีความแตกต่างมากพอเมื่อเทียบกับพื้นหลัง งานวิจัยนี้จึงใช้ในการหาขนาดและตำแหน่งของวัตถุเป้าหมายจากพิกัด 2 มิติ [29]

แบบจำลองของกล้องที่นำมาใช้ในงานวิจัยนี้คือ แบบจำลอง perspective projection คือการสมมติภาพที่ได้เกิดจากการฉายภาพจากมุมมองของกล้อง นั่นคือจุด  $cP = [x \ y \ z]^T$  เป็นจุดที่อยู่ในพิกัด 3 มิติในสิ่งแวดล้อมจริง ซึ่งจะถูกละทิ้งไปปรากฏในภาพเป็นจุด  $p = [u \ v]^T$  ตามความสัมพันธ์ดังนี้ [30]

$$\pi(x, y, z) = \begin{bmatrix} u \\ v \end{bmatrix} = \frac{f}{z} \begin{bmatrix} x \\ y \end{bmatrix}$$

เพื่อหาตำแหน่งในพิกัด 3 มิติ ถ้าเรากำหนดให้  $y$  คงที่ และสามารถหาตำแหน่งที่ได้จาก Blob detector จะทำให้เราสามารถ หาค่า  $x$  และค่า  $z$  จากสมการดังต่อไปนี้

$$z = \frac{f}{v} y \quad \text{และ} \quad x = \frac{u}{f} z$$

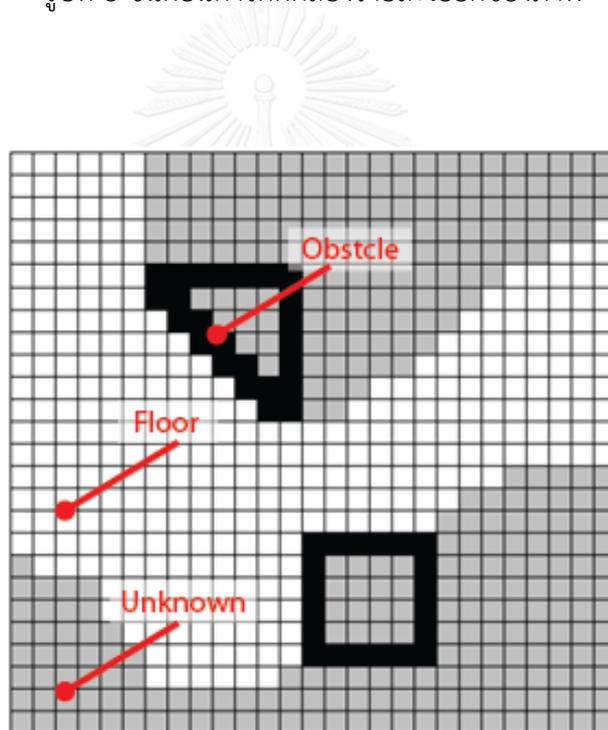


ก. ภาพข้อมูลดิบสี

ข. การคัดกรองโทนสี (Hue)

ค. การคัดกรองความสด (Saturation)

รูปที่ 8 ขั้นตอนการคัดกรองรายละเอียดของภาพ



รูปที่ 9 ตัวอย่างของการแทนตำแหน่งของตารางแผนที่ของสิ่งแวดล้อม

### 3.2 การแทนที่ตำแหน่งด้วยตารางแผนที่ Grid Occupation

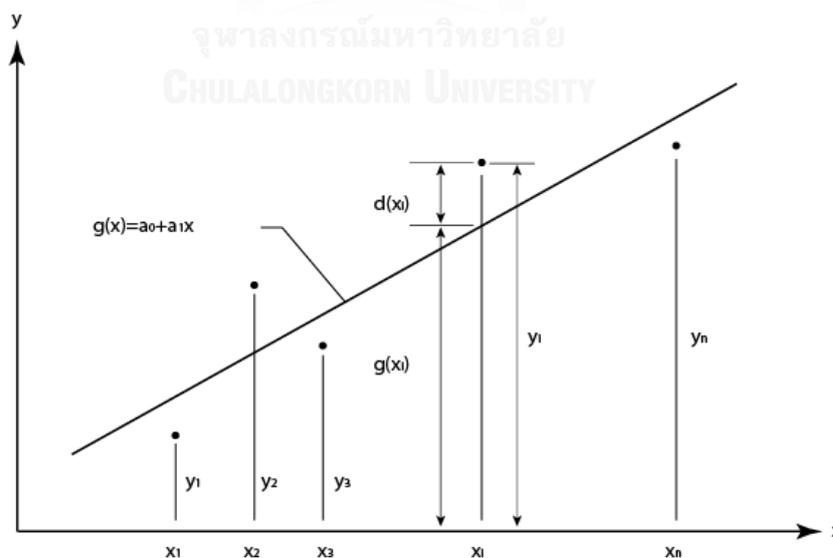
เนื่องจากข้อมูลที่ได้จากอุปกรณ์ตรวจรู้ Kinect นั้นมีปริมาณเยอะมากจนทำให้ไม่สามารถประมวลได้ จึงจำเป็นต้องมีวิธีการช่วยลดปริมาณของข้อมูลลงแต่ยังต้องรักษาคุณภาพของข้อมูลที่มีอยู่ไว้คงเดิม หนึ่งในวิธีที่เหมาะสมในการแก้ปัญหาคือการสร้างเซลล์ตารางที่มีขนาดจำกัดโดยแต่ละเซลล์นั้นจะกำหนดว่ามีข้อมูลเมื่อ ข้อมูลดิบมีข้อมูล ณ ตำแหน่งที่ตรงกับตารางว่าจะเป็นที่ว่างหรือเป็นสิ่งกีดขวาง และจะถูกกำหนดว่าไม่ทราบค่าเมื่อตำแหน่งเซลล์ในตารางนั้นไม่มีค่าปรากฏในข้อมูลดิบ ดังรูปที่ 9 ถึงแม้ว่าวิธีการแทนที่ตำแหน่งด้วยตารางแผนที่ช่วยลดปริมาณข้อมูลได้มากเพียงใด แต่ก็ต้องแลกกับความแม่นยำของข้อมูลเสมอจึงควรคำนึงถึงประเด็นนี้ไว้เสมอ [31]

### 3.3 การหาเส้นตั้งฉากของกราฟเชิงเส้นด้วยวิธีการถดถอยแบบเชิงเส้น

การถดถอยแบบเชิงเส้นอาจเป็นวิธีที่เรียบง่ายที่สุดในการสร้างฟังก์ชันเส้นตรง สำหรับชุดข้อมูลในแนวราบ 2 มิติ ดังรูปที่ 10 แสดงชุดของข้อมูลในระนาบ x-y อันประกอบด้วย  $x_i, y_i, i = 1, 2, \dots, n$  นั่นคือ มีจำนวนข้อมูลทั้งหมด  $n$  ข้อมูล ในที่นี้สมมติให้สมการเส้นตรงเป็นฟังก์ชันดังนี้

$$g(x) = a_0 + a_1x \quad (4)$$

ให้  $a_0$  และ  $a_1$  เป็นค่าคงที่ซึ่งไม่ทราบค่า และจำเป็นต้องหา เพื่อสร้างสมการเส้นตรงให้สมบูรณ์



รูปที่ 10 ตัวอย่างการถดถอยแบบเชิงเส้นโดยการสร้างฟังก์ชันเส้นตรงจากชุดข้อมูลที่เป็นจุด  
ระนาบ 2 มิติ

รูปที่ 10 จะเห็นได้ว่าตำแหน่งของ  $x_i$  ของข้อมูล  $i$  ใดๆ ค่าของฟังก์ชัน  $g(x)$  ที่ถูกสร้างขึ้นนั้นมีค่าแตกต่างไปจากค่าของ  $y_i$  เท่ากับ  $d(x_i)$  ที่ตำแหน่งนั้น นั่นหมายความว่า ความคลาดเคลื่อนสะสม  $E$  ที่เกิดจากข้อมูลทั้งหมด  $n$  ข้อมูล สามารถเขียนได้ดังนี้

$$E = \sum_{i=1}^n [d(x_i)]^2 \quad (5)$$

สาเหตุที่ต้องยกกำลังสองเพื่อกำจัดค่าความแตกต่างของ  $d(x_i)$  ที่อาจเป็นเครื่องหมายลบ ดังนั้นสมการ (5.2) จึงบ่งบอกถึงค่าความคลาดเคลื่อนโดยรวมของข้อมูลทั้งหมด สมการ (5) เขียนได้ว่า

$$E = \sum_{i=1}^n [y_i - g(x_i)]^2 \quad (6)$$

หากแทนสมการ (4) ที่  $x = x_i$  ลงในสมการ (6) จะได้

$$E = \sum_{i=1}^n [y_i - (a_0 + a_1 x_i)]^2 \quad (7)$$

จากสมการ (7) จะสามารถคำนวณหาค่าของ  $a_0$  และ  $a_1$  ได้ด้วยวิธีกำลังสองน้อยสุด (least-squares) ซึ่งใช้วิธีการหาค่าต่ำสุด (minimization) ของค่าความคลาดเคลื่อนโดยเกี่ยวข้องกับตัวแปรที่ไม่รู้ค่า นั่นคือ

$$\frac{\partial E}{\partial a_0} = 0 \quad (8ก) \quad \text{และ} \quad \frac{\partial E}{\partial a_1} = 0 \quad (8ข)$$

ตามเงื่อนไขสมการ (8ก) ให้ผลดังนี้

$$2 \sum_{i=1}^n [y_i - (a_0 + a_1 x_i)](-1) = 0$$

$$\sum_{i=1}^n y_i - \sum_{i=1}^n a_0 - \sum_{i=1}^n a_1 x_i = 0$$

$$n a_0 + \left( \sum_{i=1}^n x_i \right) = \sum_{i=1}^n y_i \quad (9ก)$$

และตามเงื่อนไขสมการ (8ข) ให้ผลดังนี้

$$2 \sum_{i=1}^n [y_i - (a_0 + a_1 x_i)](-1) = 0$$

$$\sum_{i=1}^n x_i y_i - \sum_{i=1}^n a_0 x_i - \sum_{i=1}^n a_1 x_i^2 = 0$$

$$\left( \sum_{i=1}^n x_i \right) a_0 + \left( \sum_{i=1}^n x_i^2 \right) a_1 = \sum_{i=1}^n x_i y_i \quad (9ข)$$

จากสมการ (9ก) และ (9ข) สามารถเขียนเป็นเมทริกซ์ได้คือ

$$\begin{bmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{Bmatrix} \quad (10)$$

ซึ่งสามารถใช้กฎของคราเมอร์ในการแก้ระบบสมการนี้ เพื่อหาค่าของ  $a_0$  และ  $a_1$  ได้ ดังนี้

$$a_0 = \frac{\left( \sum_{i=1}^n y_i \right) \left( \sum_{i=1}^n x_i^2 \right) - \left( \sum_{i=1}^n y_i x_i \right) \left( \sum_{i=1}^n x_i \right)}{n \left( \sum_{i=1}^n x_i^2 \right) - \left( \sum_{i=1}^n x_i \right)^2} \quad a_1 = \frac{n \left( \sum_{i=1}^n x_i y_i \right) - \left( \sum_{i=1}^n x_i \right) \left( \sum_{i=1}^n y_i \right)}{n \left( \sum_{i=1}^n x_i^2 \right) - \left( \sum_{i=1}^n x_i \right)^2}$$

และ

เท่านี้ก็สามารถหาค่า  $a_0$  และ  $a_1$  จากข้อมูลจุดที่เป็นก้ำกึ่งได้แล้ว โดย

$a_0$  คือ ค่าคงที่จุดตัดแกน y

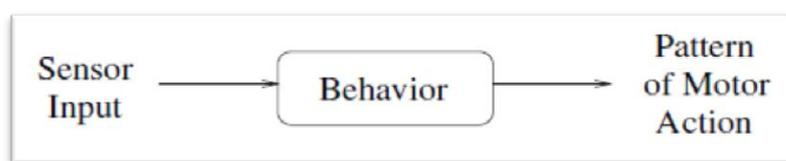
$a_1$  คือ ค่าคงที่ความชันของเส้นตรงที่สอดคล้องกับจุดที่เป็นก้ำกึ่ง

เพราะฉะนั้นจะหาเส้นตรงที่ตัดฉากกับก้ำกึ่งได้ ดังนี้

$$\text{slope} = -(a_1)^{-1}$$

### 3.4 การสร้างเส้นทางเดินด้วยสนามศักดิ์

แนวคิดพื้นฐานในการออกแบบพฤติกรรมของหุ่นยนต์ ควรเริ่มต้นจากพฤติกรรมกรรมการปรับตัวเข้ากับสิ่งแวดล้อมใดๆ ให้ได้เสียก่อน พฤติกรรมที่พึงประสงค์เหล่านี้จึงมักมีจุดเริ่มต้นของการกระทำมาจากการรับรู้ข้อมูลจากสิ่งแวดล้อมที่ส่งมาจากอุปกรณ์ตรวจรู้ ยกตัวอย่างเช่น เมื่ออุปกรณ์ตรวจรู้ตรวจจับแสงเมื่อรับแสงได้รับการกระตุ้นจากสิ่งแวดล้อมระบบจะออกคำสั่งเพื่อตอบโต้การกระตุ้นนั้น (อาจจะเป็นหันหน้าไปที่แสง) [33] กล่าวอีกนัยหนึ่ง Murphy กล่าวว่าพฤติกรรมของหุ่นยนต์ที่จากการถูกกระตุ้นของอุปกรณ์ตรวจรู้สามารถเขียนเป็นแผนภาพได้ดังรูปที่ 11



รูปที่ 11 แผนภาพอธิบายพฤติกรรมของหุ่นยนต์ ตามนิยามของ Murphy

วิธีการกระทำที่จะนำไปสู่พฤติกรรมอีกหลายอย่างของหุ่นยนต์มีทั้งหมด 2 วิธีการหลัก นั่นคือ พฤติกรรม Always-on และพฤติกรรม Sometime-on พฤติกรรม Always-on คือการที่หุ่นยนต์ตรวจรู้สิ่งต่างๆในสิ่งแวดล้อมเสมอก่อนออกคำสั่งพฤติกรรมครั้งต่อไป Sometime-on คือการที่หุ่นยนต์แสดงพฤติกรรมก็ต่อเมื่อมีสัญญาณบางอย่างจากสิ่งแวดล้อมมากระตุ้น

ภาพที่สามารถสื่อถึงสนามศักดิ์ได้ชัดเจนที่สุดคงจะเป็น ภาพของประจุไฟฟ้าเคลื่อนที่ในสนามไฟฟ้าบวก/ลบ และสนามแม่เหล็กเหนือ/ใต้ หรือภาพของลูกบอลเคลื่อนลงเนินตามสนามแรงโน้มถ่วง ซึ่งแนวคิดพื้นฐานของพฤติกรรมที่เกิดจากสนามนั้นได้มาจากพฤติกรรมของประจุไฟฟ้าและการคลึงลูกบอลในธรรมชาติทั้งสิ้น แต่ในที่นี้คือการรวมเอาพฤติกรรมหลายๆ อย่างเข้าด้วยกันเพื่อสร้างรูปทรงของสนาม แต่แตกต่างกันตรงที่สนามในสิ่งแวดล้อมจริงนั้นไม่สามารถเปลี่ยนแปลงได้แต่สำหรับการออกแบบพฤติกรรมของหุ่นยนต์ด้วยสนามศักดิ์ซึ่งเราสามารถเปลี่ยนแปลงได้ตามใจผู้ออกแบบ ซึ่งหมายความว่า ก) ผู้ออกแบบสามารถพฤติกรรมที่ซับซ้อนได้จากสนามที่สร้างขึ้น ข) สามารถให้หุ่นยนต์แสดงพฤติกรรมต่างๆ ต่อสนามศักดิ์ที่ละสนามได้ ค) สามารถนำสนามมารวมกันแล้วสร้างเส้นทางเดินของหุ่นยนต์จากสนามรวมได้ ในงานวิจัยนี้ได้เพียงแค่พฤติกรรมพื้นฐานนั้นคือสร้างเส้นเพื่อหลบหลีกสิ่งกีดขวางเพื่อนำหุ่นยนต์ไปยังเป้าหมายโดยการรวมของสนาม

### 3.4.1 การสร้างสนามศักย์ (Potential field)

สิ่งสำคัญที่จะได้จากการสร้างสนามคือ เวกเตอร์กระทำ (Action vector) ซึ่งสามารถบ่งบอกได้ถึงขนาดศักดิ์ของสนาม, ความเร็วและทิศทางการเคลื่อนที่หุ่นยนต์ โดยพฤติกรรมที่หุ่นยนต์แสดงออกมานั้นถูกขับเคลื่อนโดยเวกเตอร์กระทำโดยทั้งสิ้น ยกตัวอย่างเช่น การสร้างพฤติกรรมค้นหาเป้าหมาย (Seek Goal behavior) ผลลัพธ์ที่ได้คือเวกเตอร์ที่มีหัวลูกศรค้นหาเป้าหมาย ถ้าวัดพิจารณาเพียงแค่สองมิติจะเห็นได้ว่าจะมีเวกเตอร์หันหัวเข้าหาเป้าหมายรอบๆเป้าหมาย ดังรูปที่ 12 สนามเกิดจากการรวมกันของเวกเตอร์เหล่านี้เรียกว่า สนามศักย์ เพราะสามารถบ่งบอกได้ถึงความแรงของพฤติกรรมที่หุ่นยนต์จะแสดงออกมาโดยพิจารณาจากตำแหน่งของหุ่นยนต์ ในกรณีตัวอย่างนี้คือ ศักดิ์แรงดึงดูด (Attractive Potential) เพราะหุ่นยนต์จะแสดงพฤติกรรมเหมือนให้เคลื่อนที่เข้าหาเป้าหมายเหมือนถูกดูดด้วยแรงโน้มถ่วง

การสร้างสนามศักย์สามารถสร้างได้ดังนี้ สิ่งแรกคือต้องเข้าใจก่อนว่าสนามศักย์จะเป็นสนามของแรงและทิศทางของแรงกล่าวคือเวกเตอร์ที่กระทำกับของบริเวณรอบๆ แหล่งกำหนดนั้นหมายถึงเวกเตอร์ที่ประกอบด้วยทิศทางและขนาด เราสามารถหาทิศทางของเวกเตอร์ได้จากการเทียบตำแหน่งระหว่างจุดต่างๆ รอบๆ จุดกำเนิดสนามศักย์ จากรูปสองมิติ รูปที่ 12 ข้างต้น จะได้ความสัมพันธ์ดังนี้

ให้	$x$	คือ ตำแหน่งใดๆในแกน $x$
	$y$	คือ ตำแหน่งใดๆในแกน $y$
	$x_0$	คือ ตำแหน่งของแหล่งกำเนิดสนามศักย์ในแกน $x$
	$y_0$	คือ ตำแหน่งของแหล่งกำเนิดสนามศักย์ในแกน $y$
	$\vec{d}$	คือ เวกเตอร์จากแหล่งกำเนิดสนามศักย์ถึงตำแหน่งที่สนใจ

ขนาดระยะห่างจากแหล่งกำเนิดสนามศักย์ถึงตำแหน่งที่สนใจคือ  $d = \sqrt{(x_o - x)^2 + (y_o - y)^2}$   
สามารถแตก  $\vec{d}$  เป็นเวกเตอร์ในแนวแกน x และ y นั่นคือ  $\vec{d}_x$  และ  $\vec{d}_y$

โดยที่  $\vec{d}_x$  มีขนาด  $d_x = d \times \cos \theta$

$\vec{d}_y$  มีขนาด  $d_y = d \times \sin \theta$

เมื่อหาทิศทางของเวกเตอร์ได้แล้ว ต่อไปจะเป็นการหาขนาดของเวกเตอร์ ในที่นี้จะกำหนดให้ขนาดของเวกเตอร์นั้นผันตรงกับระยะทาง (d) หมายความว่าเมื่อระยะทางยิ่งห่าง (ค่า d มีค่ามาก) จะเกิดเป็นศักย์ขึ้นตามไปด้วย ซึ่งสามารถพิจารณาเวกเตอร์ที่ละแกนได้จากเวกเตอร์  $\vec{d}_x$  และ  $\vec{d}_y$  ที่หาไว้ก่อนหน้านี้ จะได้เวกเตอร์ของแรงศักย์ดังนี้

$\vec{F}_x = F \times d_x$  คือแรงในแกน x

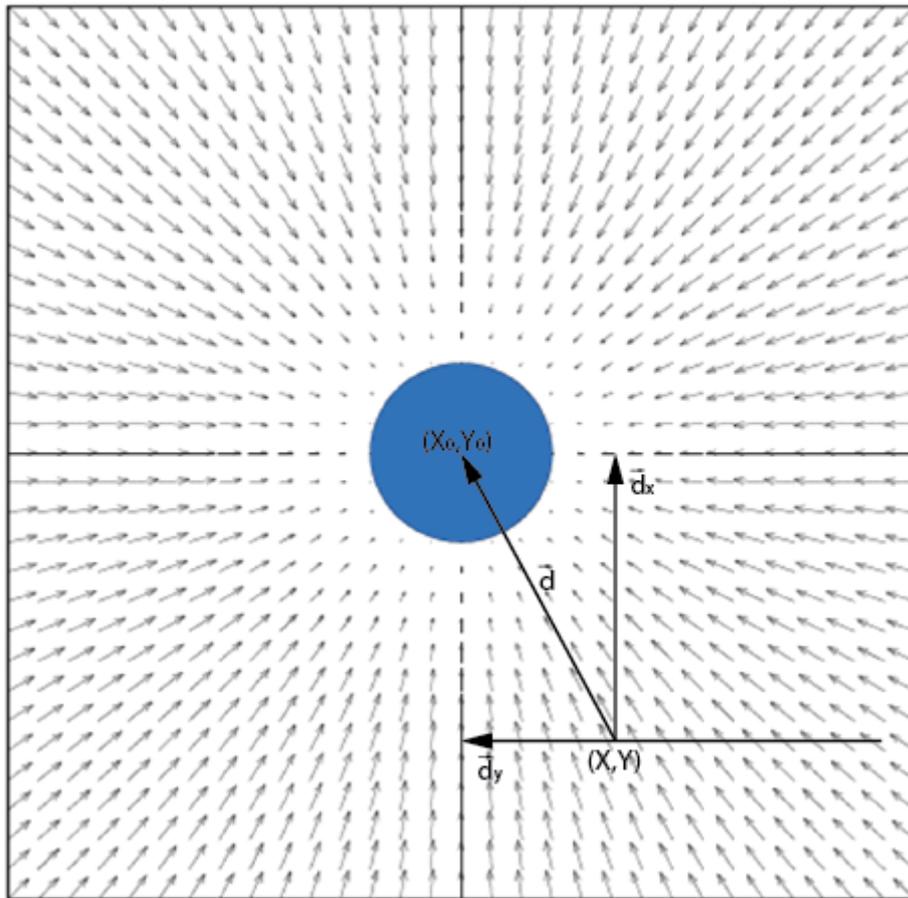
$\vec{F}_y = F \times d_y$  คือแรงในแกน y

ที่ค่า F เป็นค่าคงที่ใดๆ ก็ได้ที่ดูแล้วเหมาะสม สาเหตุที่เป็นเช่นนี้เป็นเพราะว่าการสร้างลักษณะของสนามศักย์นั้น ผู้สร้างต้องเป็นผู้กำหนดขนาดหรือแม้กระทั่งลักษณะทิศทางให้เอง จะอธิบายเพิ่มเติมในหัวข้อถัดไป เมื่อนำ  $\vec{F}_x$  รวมกับ  $\vec{F}_y$  จะได้เวกเตอร์แรงรวม  $\vec{F} = \vec{F}_x + \vec{F}_y$

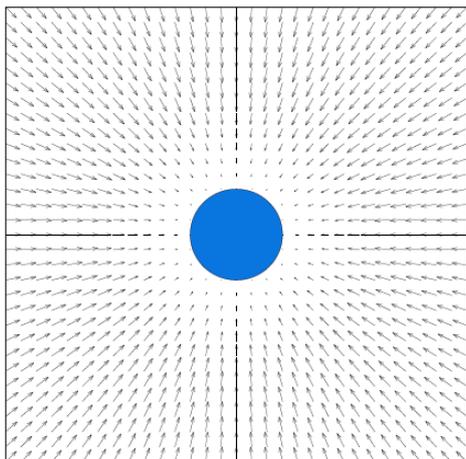
ถ้าต้องการเป็นสนามผลักตัน ก็สามารถทำได้เลยโดยการคูณด้วย -1

### 3.4.2 ประเภทของสนามศักย์

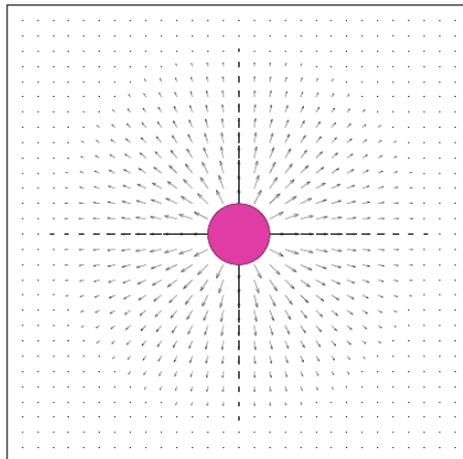
สนามศักย์ที่นิยมใช้มี 2 แบบ คือ สนามศักย์ดึงดูด ดังรูปที่ 13ก เพราะสร้างพฤติกรรมเคลื่อนที่เข้าหาเป้าหมาย และ สนามศักย์แรงผลัก เพราะสร้างพฤติกรรมหลบหลีกสิ่งกีดขวาง ดังรูปที่ 13ข นอกจากนี้แล้วยังมีสนามศักย์ที่น่าสนใจอีกหลายสนามศักย์ ยกตัวอย่างเช่น สนามศักย์แนวขนาน รูปที่ 13ค สนามศักย์แนวตั้งฉาก รูปที่ 13ง สนามศักย์หมุนวน รูปที่ 13จ หรือแม้กระทั่งสนามศักย์แบบสุ่ม รูปที่ 13ฉ เป็นต้น



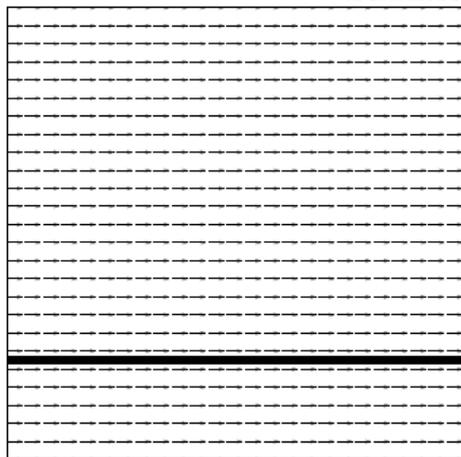
รูปที่ 12 การสร้างเวกเตอร์แรงจากแหล่งกำเนิดสนามศักย์



ก. สนามศักด์ิ่งจุด



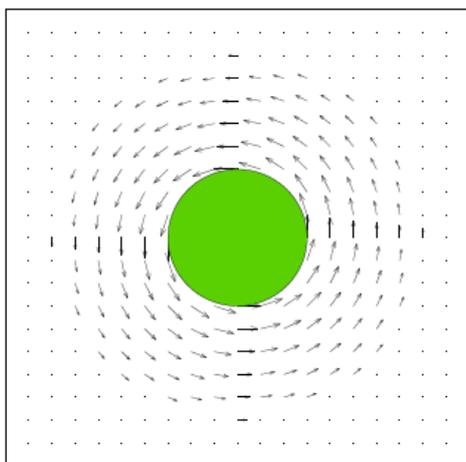
ข. สนามศักด์ิ่งล้กัตัน



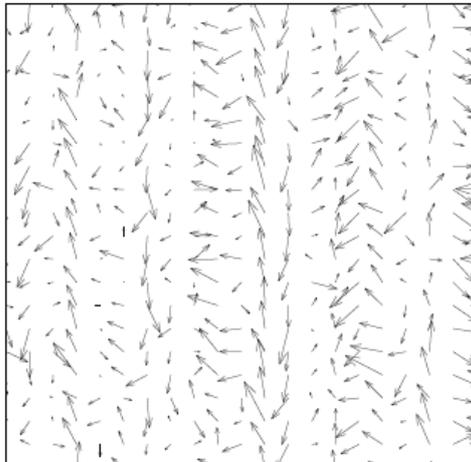
ค. สนามศักด์ิ่งแนวขนาน



ง. สนามศักด์ิ่งแนวตั้งฉาก



จ. สนามศักด์ิ่งหมุนวน

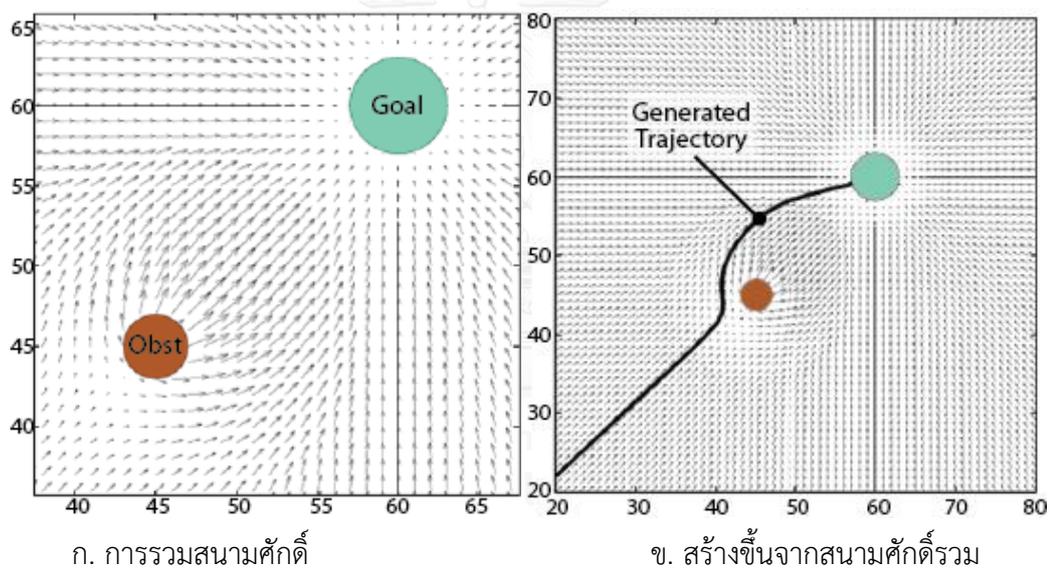


ฉ. สนามศักด์ิ่งแบบสุ่ม

รูปที่ 13 สนามศักด์ิ่งแบบต่างๆ

### 3.4.3 การรวมสนามศักย์

เพื่อที่จะสร้างเส้นทางหลบหลีกสิ่งกีดขวาง เมื่อหุ่นยนต์เคลื่อนที่เข้าใกล้สิ่งกีดขวางหุ่นยนต์ควรจะแสดงพฤติกรรมเคลื่อนที่หลบหลีกสิ่งกีดขวาง หรือมองสิ่งกีดขวางว่าเป็นสนามผลักดัน ในขณะเดียวกันก็ยังมีพฤติกรรมพยายามเคลื่อนที่เข้าหาเป้าหมายเช่นกัน เพื่อสร้างเส้นทางที่เหมาะสมจึงต้องทำการรวมสนามศักย์ขึ้น ซึ่งก็คือการนำเวกเตอร์ของแหล่งกำเนิดสนามศักย์แต่ละแหล่งที่มีผลต่อตำแหน่งเดียวกันมารวมกัน จะทำให้ได้เวกเตอร์ลัพธ์เป็นเวกเตอร์ควบคุมความประพฤติใหม่ในตำแหน่งนั้น เมื่อพิจารณาให้ครบทุกตำแหน่งในแผนที่จะได้สามารถสร้างสนามศักย์ รูปที่ 14ก เมื่อหุ่นยนต์เคลื่อนที่ไปยังตำแหน่งใหม่จะพิจารณาเวกเตอร์แรงลัพธ์ที่อยู่ ณ ตำแหน่งนั้นแล้วเคลื่อนที่ตามทิศทางที่เวกเตอร์ชี้ไปเป็นแบบนี้ไปเรื่อยจนกระทั่งเคลื่อนที่ถึงที่หมาย [34] ดังรูป 14ข



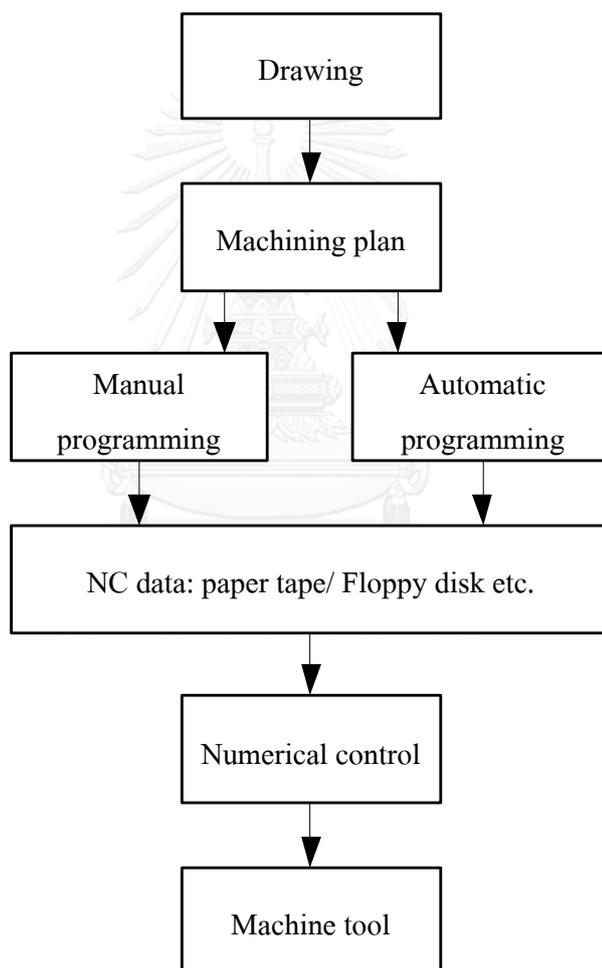
รูปที่ 14 การสร้างเส้นทางทางเดินจากสนามศักย์รวม

จะเห็นได้ว่าลักษณะของเวกเตอร์แรงลัพธ์ในบริเวณรอบๆ นั้นได้รับอิทธิพลจากแหล่งกำเนิดสนามศักย์หลายๆ แหล่งด้วยกัน โดยที่แต่ละแหล่งนั้นจะมีเวกเตอร์แรงเป็นของตัวเองตั้งนั้น ดังนั้นการหาเวกเตอร์แรงรวม ณ ตำแหน่งใดๆ สามารถทำได้โดยการรวมเวกเตอร์ธรรมดา สามารถเขียนเป็นสมการได้ดังนี้

$$\vec{F}_{sum} = \vec{F}_{o1} + \vec{F}_{o2} + \vec{F}_{o1} + \dots$$

### 3.5 การโปรแกรมหุ่นยนต์เคลื่อนที่ด้วยล้อ แบบ Differential Drive

NC ย่อมาจากคำว่า Numerical Control คือการนำข้อมูลระบบเชิงตัวเลขมาใช้สำหรับการควบคุม โดยเครื่องจักรจะทำตามชุดคำสั่งของมนุษย์ ผู้ใช้จะต้องเขียนชุดคำสั่ง ซึ่งเรียกการสั่งชุดคำสั่งนี้ว่า “Programming” จะเป็นภาษาเฉพาะ ที่เก็บรูปแบบคำสั่ง ควบคุมการเคลื่อนที่ของเครื่อง, ควบคุมความเร็วการหมุนของเครื่อง และคำสั่งเฉพาะต่างๆ แทนที่การเขียนโปรแกรมอันยุ่งยากโดยภาษาคอมพิวเตอร์อื่นๆ ในงานวิจัยนี้ทางผู้วิจัยจะใช้ชุดคำสั่งจากโปรแกรม MATLAB ที่ถูกพัฒนามาเพื่อควบคุมหุ่นยนต์เคลื่อนที่ด้วยล้อ



รูปที่ 15 กระบวนการโปรแกรมรหัสเอ็นซี

### 3.5.1 การระบุตำแหน่งของหุ่นยนต์เคลื่อนที่ด้วยล้อ

หุ่นยนต์เคลื่อนที่ด้วยล้อเป็นหุ่นยนต์ที่มีขอบเขตการทำงานไม่จำกัด ทำให้สิ่งแวดล้อมมีผลต่อการทำงานของหุ่นยนต์โดยตรง ดังนั้นการระบุค้นหาตำแหน่งของหุ่นยนต์เคลื่อนที่ด้วยล้อจึงเป็นปัญหาที่สำคัญมากต่อการทำงานของหุ่นยนต์ หุ่นยนต์ซึ่งจำเป็นต้องมีอุปกรณ์ตรวจสอบบางอย่างใช้ในการค้นหาตำแหน่งของวัตถุที่อยู่ในสิ่งแวดล้อม เช่น สร้างแผนที่ ตรวจสอบวัฏระยะทางหรือความเร็วการเคลื่อนที่ของหุ่นยนต์ เพื่อบอกว่าขณะนี้หุ่นยนต์กำลังอยู่ที่ตำแหน่งแผนที่ ซึ่งการระบุตำแหน่งของหุ่นยนต์นั้นได้แบ่งออกเป็น 2 กลุ่มด้วยกัน คือ local localization กับ global localization และ passive localization กับ active localization

Local localization คือ การให้หุ่นยนต์หาตำแหน่งโดยที่ไม่ให้ข้อมูลเริ่มต้นเกี่ยวกับตำแหน่งเลย หุ่นยนต์มีแผนที่แต่ไม่รู้ว่าตอนเริ่มต้นนั้นอยู่ตำแหน่งไหนในแผนที่

Global localization คือ คล้ายกับ Local localization ต่างกันตรงที่หุ่นยนต์มีข้อมูลเริ่มต้นเกี่ยวกับตำแหน่งที่อยู่

Passive localization คือ การเคลื่อนที่ของหุ่นยนต์ถูกควบคุมจากระบบภายนอก ระบุตำแหน่งของหุ่นยนต์จากการเคลื่อนที่โดยใช้อุปกรณ์ตรวจสอบเพื่อระบุตำแหน่ง

Active localization คือ หุ่นยนต์ควบคุมการเคลื่อนที่ตัวเอง เพื่อทำการระบุตำแหน่งของหุ่นยนต์ให้มีประสิทธิภาพมากขึ้นโดยการใช้ algorithm ในการพิจารณาข้อมูลที่ได้รับมา

### 3.5.2 การสร้างคำสั่งที่ควบคุมหุ่นยนต์ iRobot Create

หุ่นยนต์ iRobot Create ขับเคลื่อนแบบ Differential Drive ประกอบด้วยล้อคู่ 2 ล้อ และ ล้ออิสระอีก 1 ล้อ ทำให้หุ่นยนต์สามารถเคลื่อนที่เป็นเส้นตรง เป็นเส้นโค้งและหันซ้ายหันขวาได้จากการควบคุมความเร็วของล้อ แต่มีความจำกัดการเคลื่อนที่อยู่ที่คือหุ่นยนต์ไม่สามารถเคลื่อนที่แบบไถลด้านข้างได้ การกำหนดเป้าหมายการเคลื่อนที่ของหุ่นยนต์นั้นต้องการกำหนดจากการระบุเป้าหมายในพิกัดฉาก 2 มิติ นั่นคือสามารถกำหนดจุดบนพิกัดคาร์ทีเซียนได้เลย แต่การเขียนคำสั่งการเคลื่อนที่หุ่นยนต์จึงต้องเขียนในพิกัดวงกลม เพราะหุ่นยนต์ถูกออกแบบมาเช่นนั้น ดังนั้นเพื่อที่จะสั่งหุ่นยนต์ให้เคลื่อนที่ไปยังตำแหน่งบนพิกัดฉากให้ถูกต้องจำเป็นต้องแปลงตำแหน่งเป้าหมาย ให้เป็นองศาที่หุ่นยนต์ต้องเลี้ยวและระยะทางที่หุ่นยนต์ต้องเดินทาง [45] ดังรูปที่ 16

จากรูปที่ 16 จะเห็นได้ว่า  $(x_i, y_i)$  และ  $(x_{i+1}, y_{i+1})$  เป็นจุดที่กำหนดไว้ให้หุ่นยนต์ต้องเดินทาง ซึ่งสามารถสร้างเวกเตอร์ได้ดังนี้  $\vec{v}_i$  และ  $\vec{v}_{i+1}$  ตามลำดับ และ  ${}^i\vec{v}_{i+1}$  คือเวกเตอร์แสดงความสัมพันธ์ระหว่างจุดใหม่กับจุดเก่า เพื่อที่จะออกสั่งการหุ่นยนต์จำเป็นต้องหาระยะทางที่หุ่นยนต์ต้องเคลื่อนที่ โดยการหาขนาดของเวกเตอร์

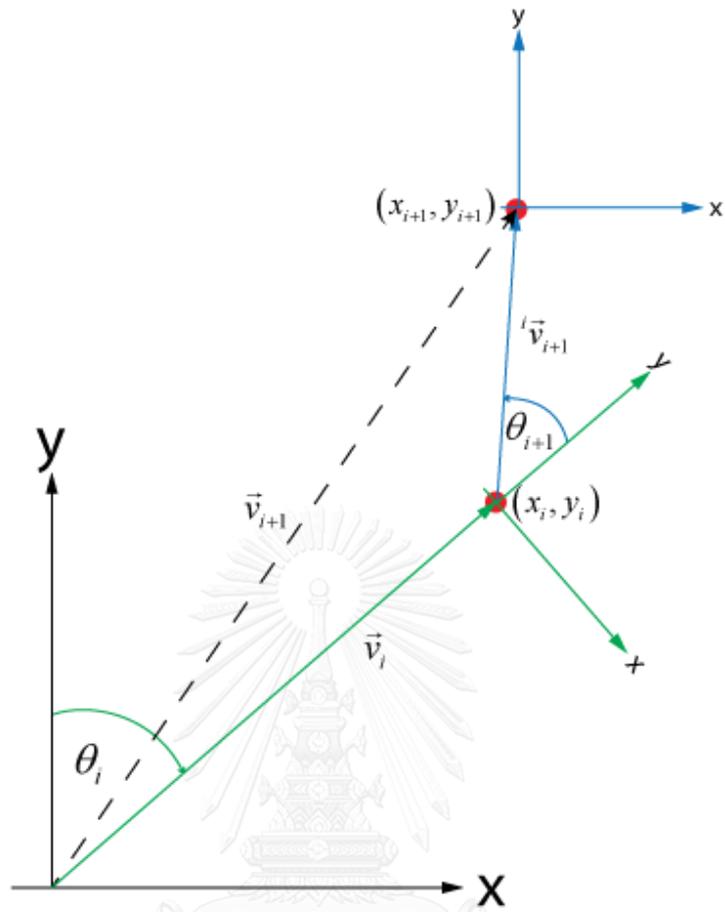
$${}^i\vec{v}_{i+1} = \vec{v}_{i+1} - \vec{v}_i$$

$$\text{ระยะที่ต้องเคลื่อนที่} \quad |{}^i\vec{v}_{i+1}| = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

และหามุมจากความสัมพันธ์ของเวกเตอร์ดังนี้

$${}^i\vec{v}_{i+1} \cdot \vec{v}_i = |{}^i\vec{v}_{i+1}| |\vec{v}_i| \cos \theta_{i+1}$$

$$\theta_{i+1} = \arccos \left( \frac{{}^i\vec{v}_{i+1} \cdot \vec{v}_i}{|{}^i\vec{v}_{i+1}| |\vec{v}_i|} \right)$$



รูปที่ 16 ความสัมพันธ์ของเวกเตอร์ระหว่างจุดแต่ละจุด

## บทที่ 4

### ภาพรวมของระบบ

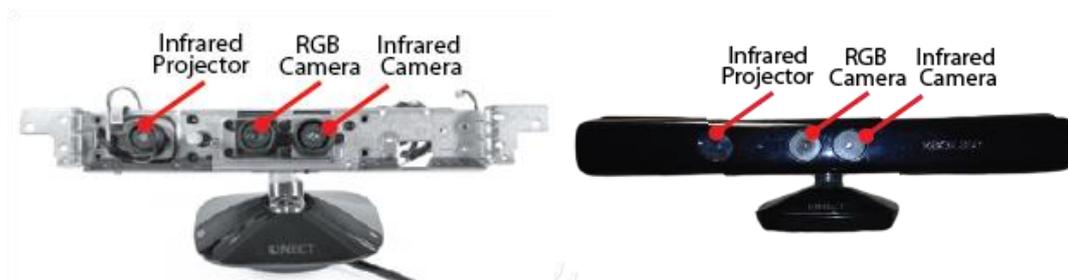
#### 4.1 อุปกรณ์ตรวจรู้ Kinect

ในเดือนมิถุนายน ปีพ.ศ. 2552 บริษัทไมโครซอฟ ได้เปิดตัวและสาธิตอุปกรณ์เสริมชนิดใหม่สำหรับเครื่องเล่นเกมในเครื่อง Xbox 360 ชื่อว่า Project Natal ซึ่งหลังจากนั้นถูกเรียกว่า Kinect sensor อุปกรณ์ตรวจรู้ชนิดนี้ได้รับความสนใจจากวงการหุ่นยนต์เป็นอย่างมาก เนื่องการศักยภาพและความเป็นไปได้ในการพัฒนา ปรับปรุง แก้ไข เพื่อใช้ประโยชน์ร่วมกับอุปกรณ์อื่นๆ นอกเหนือจากอุปกรณ์เล่นเกม อาทิเช่น ใช้เป็นอุปกรณ์ตรวจรู้ให้หุ่นยนต์ ใช้เป็นเครื่องวิเคราะห์วัตถุ สามมิติ Kinect ถูกเริ่มวางจำหน่ายตั้งแต่เดือน 2010 [31]

โปรแกรมพื้นฐานของ Kinect ถูกพัฒนาโดยบริษัทซอฟต์แวร์ชื่อว่า Rare เป็นบริษัทย่อยในเครือไมโครซอฟ ซึ่งเน้นพัฒนาเทคโนโลยีของกล้อง ส่วนเทคโนโลยีกล้องตรวจจับความลึกถูกพัฒนาโดยบริษัท PrimeSense [19, 46] ซึ่งสามารถแปลค่าจากสิ่งแวดล้อมจริงให้เป็นข้อมูลตำแหน่งในสามมิติได้ โดยใช้ประโยชน์จากการส่องแสงอินฟราเรด อุปกรณ์ตรวจรู้ความลึกนั้นประกอบไปด้วย อุปกรณ์ฉายแสงเลเซอร์อินฟราเรดทำงานร่วมกับอุปกรณ์ตรวจรู้ Monochrome CMOS ซึ่งตรวจจับและบันทึกข้อมูลในสามมิติของวัตถุภายใต้แสงธรรมชาติเป็นวิดีโอ อย่างไรก็ตามจากการทดลองที่ผ่านมายังไม่สามารถใช้ Kinect เพื่อตรวจจับหาตำแหน่งของวัตถุในบริเวณที่มีมืดหรือสว่างเกินไปได้ นอกจากนี้อุปกรณ์ Kinect ยังมีกล้องวิดีโอสี RB G ติดตั้งอยู่ในตำแหน่งระหว่างอุปกรณ์ฉายแสงอินฟราเรดและอุปกรณ์ตรวจรู้แสงอินฟราเรดแต่กล้องวิดีโอสีไม่ได้ถูกใช้ในการตรวจจับความลึก อย่างไรก็ตาม บริษัท PrimeSense ผู้พัฒนาอุปกรณ์ตรวจรู้ความลึกไม่ได้ตีพิมพ์ข้อมูลเทคโนโลยีของกล้องตรวจจับความลึกไว้แต่อย่างไร

Kinect มีระบบตรวจรู้หลายระบบประกอบเข้าด้วยกัน โดยเริ่มจากส่วนที่เด่นที่สุดคือ อุปกรณ์ตรวจจับความลึก, กล้องวิดีโอสี และไมโครโฟนสี่ตัว ทำให้สามารถตรวจรู้การเคลื่อนไหวร่างกาย จดจำใบหน้าและเสียงของผู้ใช้ได้ จากรูป 17 ประกอบด้วย อุปกรณ์ฉายแสงอินฟราเรดกล้องวิดีโอสี และ อุปกรณ์ตรวจรู้แสงอินฟราเรด ตามลำดับ ข้อมูลที่ได้จากกล้องวิดีโอ Kinect เป็นข้อมูลวิดีโอสี อัตราเฟรมที่ 30 โดยภาพวิดีโอสี RGB ที่ได้ถูกส่งผ่านด้วย VGA ขนาด 8 บิต ผ่านการกรองสี

ด้วย Bayer ความละเอียดที่ 640x480 pixels ส่วนภาพวิดีโอที่ได้จากอุปกรณ์ตรวจรู้ความลึก ได้ถูกส่งผ่านด้วย VGA ความละเอียดที่ 640x480 pixels ขนาดลึก 11 บิต ซึ่งได้การสัมผัสความลึกที่ 2,048 ระดับ อุปกรณ์ตรวจรู้มีขนาดมูรับสัมผัสที่ 57 องศาในแนวนอน และ 43 องศาในแนวตั้ง อุปกรณ์ตรวจรู้ Kinect มีระยะสัมผัสในช่วง 1.2 – 3.5 เมตรเมื่อใช้กับ Xbox ซอฟแวร์



ก. ภายในอุปกรณ์ตรวจรู้ Kinect

ข. ภายนอกอุปกรณ์ตรวจรู้ Kinect

รูปที่ 17 อุปกรณ์ตรวจรู้ Kinect

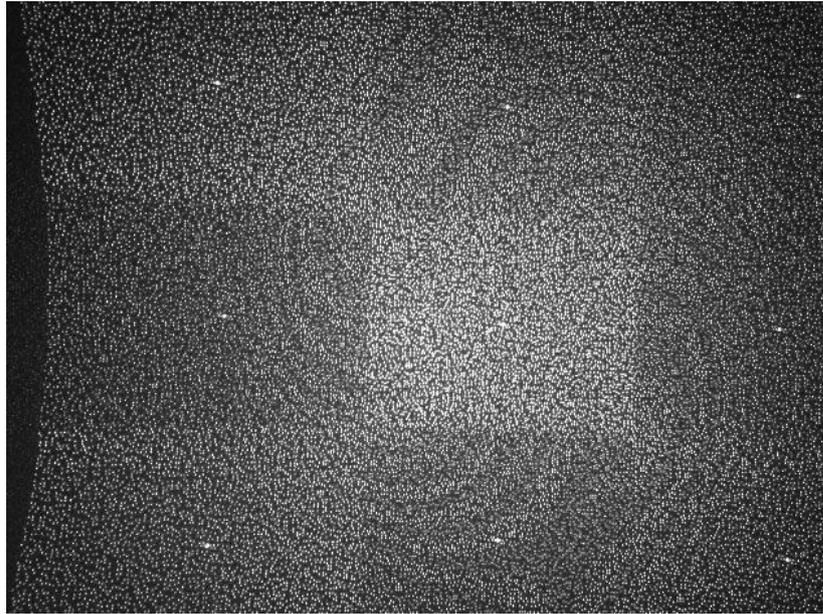
Kinect เป็นอุปกรณ์ตรวจรู้ที่พิเศษกว่ากล้องธรรมดาตรงที่สามารถตรวจจับความลึกได้ [19] ทำให้คอมพิวเตอร์สามารถแยกแยะออกได้ว่าส่วนไหนของภาพเป็นฉาก ส่วนไหนเป็นผู้ใช้ทำให้สามารถตีความคำสั่งจากการเคลื่อนไหวของผู้ใช้ นอกจากนั้นยังรับคำสั่งจากเสียงได้ด้วย Kinect เป็นอุปกรณ์ที่มีราคาถูกทำให้บุคคลทั่วไปเข้าถึงได้ง่าย เปิดทางให้กับวิธีใหม่ๆ ในการควบคุมหุ่นยนต์ [46] และเนื่องจากถูกพัฒนาขึ้นมาเพื่อใช้ควบคุมเล่นเกม Xbox 360 โดยผู้ใช้สามารถออกคำสั่งได้โดยการเคลื่อนไหวส่วนต่างๆ ของร่างกายได้อย่างเป็นธรรมชาติโดยไม่จำเป็นต้องกดปุ่มซึ่งไม่สามารถทำได้ง่ายด้วยกล้องวิดีโอธรรมดา ทำให้ได้รับความสนใจจากนักวิจัยจากหลายๆวงการ เพื่อที่จะตอบสนองให้ทันคำสั่งของมนุษย์ ระบบคอมพิวเตอร์จำเป็นจะต้องเข้าใจสิ่งที่ผู้ใช้กำลังทำก่อนที่แสดงผลต่อสนองผู้สั่ง และต้องรับมือกับปริมาณข้อมูลมหาศาลที่ได้จากกล้อง RGB-D ซึ่งปกติต้องใช้เวลาในการคำนวณนานมาก Microsoft จึงติดตั้งไมโครโพรเซสเซอร์ (Microprocessor) มาพร้อมในตัวของอุปกรณ์ตรวจรู้ Kinect เพื่อช่วยในการรับภาระในส่วนนี้

#### 4.1.1 การตรวจจับความลึก

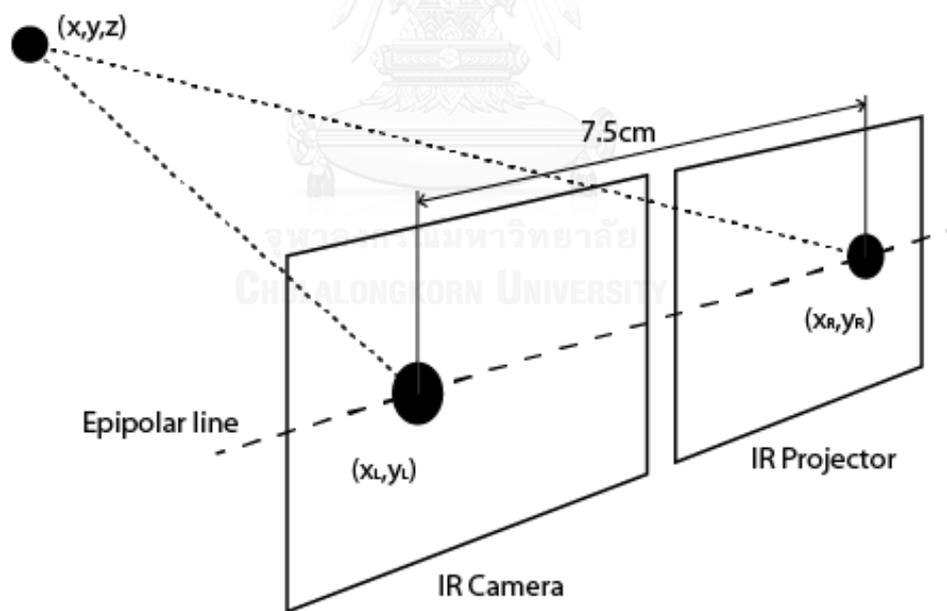
อันที่จริงแล้ว PrimeSense ได้บอกอย่างชัดเจนว่าระบบตรวจรู้ความลึกได้ใช้ระยะเวลาการเคลื่อนที่ในการคำนวณแต่ใช้บางสิ่งที่เรียกว่า “Light coding” ซึ่งไม่สามารถบอกเวลาการเดินทางของแสงได้ ซึ่งอุปกรณ์ฉายลำแสงอินฟราเรดสามารถฉายลำแสงแบบสุ่มที่มีลักษณะดังรูปที่ 18

การแก้สมการสามเหลี่ยม Stereo triangulation จำเป็นต้องใช้ภาพสองภาพเพื่อหาความลึกของแต่ละจุด ด้วยเทคนิคนี้จะรับข้อมูลภาพภาพแรกจากกล้องตรวจจับแสงอินฟราเรด และภาพที่สองมาจากหลอดฉายที่เครื่องฉายอินฟราเรดส่องออกไป ซึ่งแบบแผนการส่องแสงควรจะถูกบันทึกไว้ในชิปเก็บข้อมูลอยู่แล้ว ภาพทั้งสองนั้นย่อมมีความเหลื่อมล้ำกัน เนื่องจากอุปกรณ์ฉายและอุปกรณ์ตรวจรู้มีระยะห่างกันแน่นอน และตำแหน่งจุดที่ปรากฏบนภาพทั้งสองมีความแตกต่างกันให้สามารถแก้สมการสามเหลี่ยมเพื่อหาความลึกของจุดต่างๆ ดังรูปที่ 19 กล้องอินฟราเรดและอุปกรณ์ฉายลำแสงอินฟราเรดอยู่ห่างกันประมาณ 7.5 cm

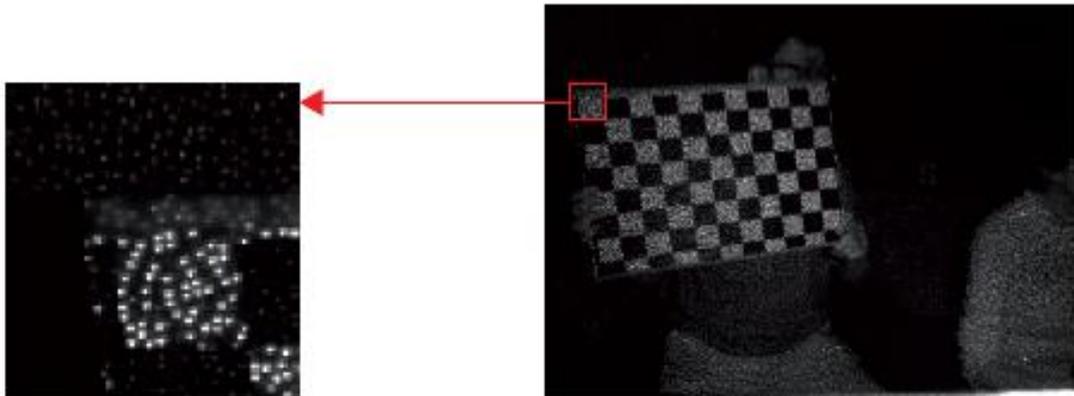
เนื่องจากลำแสงและอุปกรณ์ตรวจรู้ความลึกนั้นมีระยะห่างระหว่างกันที่แน่นอนที่เป็นประโยชน์ต่อการคำนวณต่อไป สิ่งที่เราเลือกก็คือการหาตำแหน่งจุดในภาพหลังจากทำการแก้การบิดเบี้ยวของเลนส์แล้วนำตำแหน่งดังกล่าวมาเทียบกับค่ารหัสของลำแสง นั้นหมายความว่าตราบไคที่สามารถตรวจจับจุดอินฟราเรด (IR dot) ได้ ดังรูปที่ 20 ก็สามารถเทียบตำแหน่งของภาพกลับมาเป็นข้อมูล 3 มิติได้เช่นกัน รูปที่ 21 แสดงตัวอย่าง ภาพแสดงความลึก โดยค่าความลึกของตำแหน่งภาพแต่ละจุดเป็นค่าโทนสีเทาขาว สีเข้มกว่าหมายถึงอยู่ใกล้กว่าเทียบจากปริภูมิของกล้อง สีดำหมายถึงไม่สามารถวัดความลึกได้ ซึ่งสามารถเกิดขึ้นได้ถ้าหากว่าสิ่งที่ตรวจจับนั้นอยู่ไกลเกินไป สาเหตุจากข้อมูลไม่มีความถูกต้องมากพอ และไกลเกินไป สาเหตุเนื่องมาจากระยะจำกัดของการมองเห็น ซึ่งค่าความลึกที่ได้จากบางครั้งนั้นจะมีความคลาดเคลื่อนอยู่บาง ทำให้ต้องทำการสอบเทียบก่อนใช้เสมอ



รูปที่ 18 ลายจุดอินฟราเรดที่เกิดจากการฉายลำแสงอินฟราเรด ซึ่งเป็นลายที่มาจากหน่วยความจำ  
ของอุปกรณ์ฉาย [31]



รูปที่ 19 ตำแหน่งที่ความแตกต่างกันของจุดในภาพของกล้องรับความลึก ( $X_L, Y_L$ ) กับภาพจากกล้อง  
ฉายลำแสงอินฟราเรด ( $X_R, Y_R$ ) ซึ่งค่านี้เป็นค่าที่อยู่ในหน่วยความทรงจำ



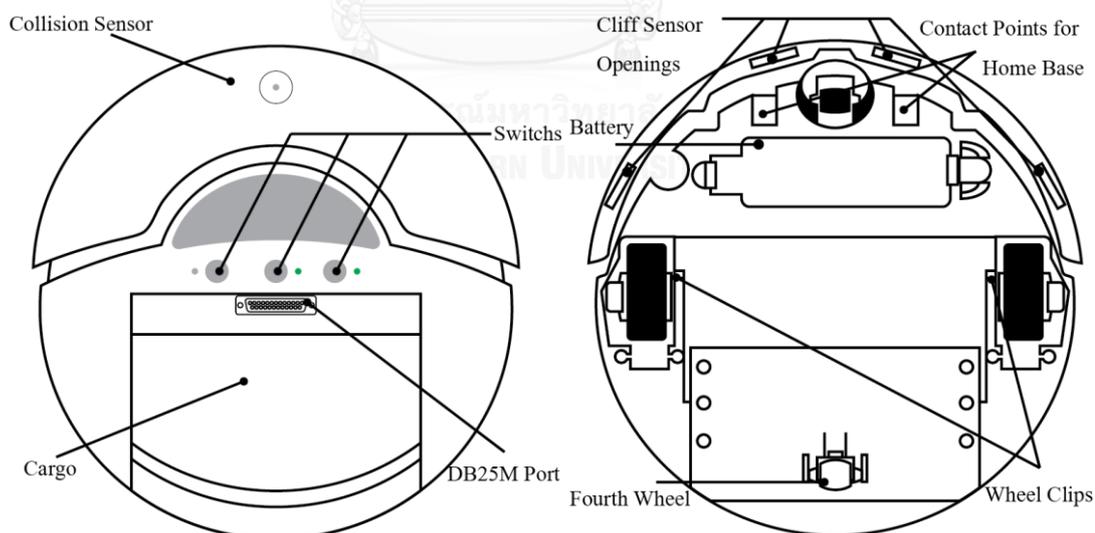
รูปที่ 20 จุดอินฟราเรดที่ตรวจรู้จากกล้องตรวจจับความลึก [31]



รูปที่ 21 ตัวอย่าง ภาพแสดงความลึก โดยค่าความลึกของแต่ละตำแหน่งภาพแต่ละจุดเป็นค่าสีเทาขาว [31]

## 4.2 หุ่นยนต์เคลื่อนที่ด้วยล้อ iRobot Create

หุ่นยนต์เคลื่อนที่ด้วยล้อ IRobot Create เป็นหุ่นยนต์เชิงพาณิชย์ได้รับการเปิดตัวตั้งแต่ปี 2007 เป็นหุ่นยนต์ที่มีโครงสร้างเดียวกับหุ่นยนต์คู่แข่ง มีรูปทรงเป็นทรงกระบอกแบนเตี้ยคล้ายเหรียญ ดังรูปที่ 22ก มีความยาวเส้นผ่านศูนย์กลาง 33 เซนติเมตร มีความสูง 8.5 เซนติเมตร และมีน้ำหนักสุทธิไม่เกิน 2.5 กิโลกรัม ขณะที่ไม่บรรทุกสัมภาระ คำว่า 'Create' ในชื่อบ่งบอกถึงว่า ผู้ใช้สามารถพัฒนาฟังก์ชันเสริมให้กับหุ่นยนต์รุ่นนี้ได้ เพราะหุ่นยนต์รุ่นนี้มีกระเบบบรรทุกสัมภาระความจุ 1150 ลูกบาศก์เซนติเมตร และมีช่องเชื่อมต่อสัญญาณให้กับอุปกรณ์ภายนอก ซึ่งผู้วิจัยสามารถออกแบบกลไกเพื่อติดตั้งไว้ในบนกระเบบประกอบกับมีไฟเลี้ยงจากตัวหุ่นด้วย ตัวหุ่นยนต์เองมาพร้อมกับอุปกรณ์ตรวจรู้หลายชนิด ซึ่งมีประโยชน์ต่อการนำทาง ได้แก่ อุปกรณ์ตรวจรู้การชน, เอนโคเดอร์ (Encoder) ที่ความละเอียด 16 บิต, อุปกรณ์ชี้วัด LED, ปุ่มกด และ มอเตอร์กระแสตรง 2 มอเตอร์ ดังรูปที่ 22ข จากการทดลองพบว่าความเร็วสูงสุดของหุ่นยนต์อยู่ที่ 0.5 เมตรต่อวินาที ในทิศทางทั้งข้างหน้าและข้างหลัง [32] ขณะที่ยังไม่ได้บรรทุกสัมภาระสามารถเร่งไปยังความเร็วสูงสุดภายใน 2 วินาที [32, 45] คำสั่งควบคุมที่มาพร้อมกับตัวหุ่นยนต์สามารถควบคุมหุ่นยนต์ได้คำสั่งพื้นฐาน ได้แก่ ระยะทางการเคลื่อนที่ ความเร็วในการวิ่ง หรือ มุมที่ต้องการหัน [45]



ก. ภาพจากด้านบน

ข. ภาพจากด้านล่าง

รูปที่ 22 หุ่นยนต์เคลื่อนที่ด้วยล้อ iRobot Create

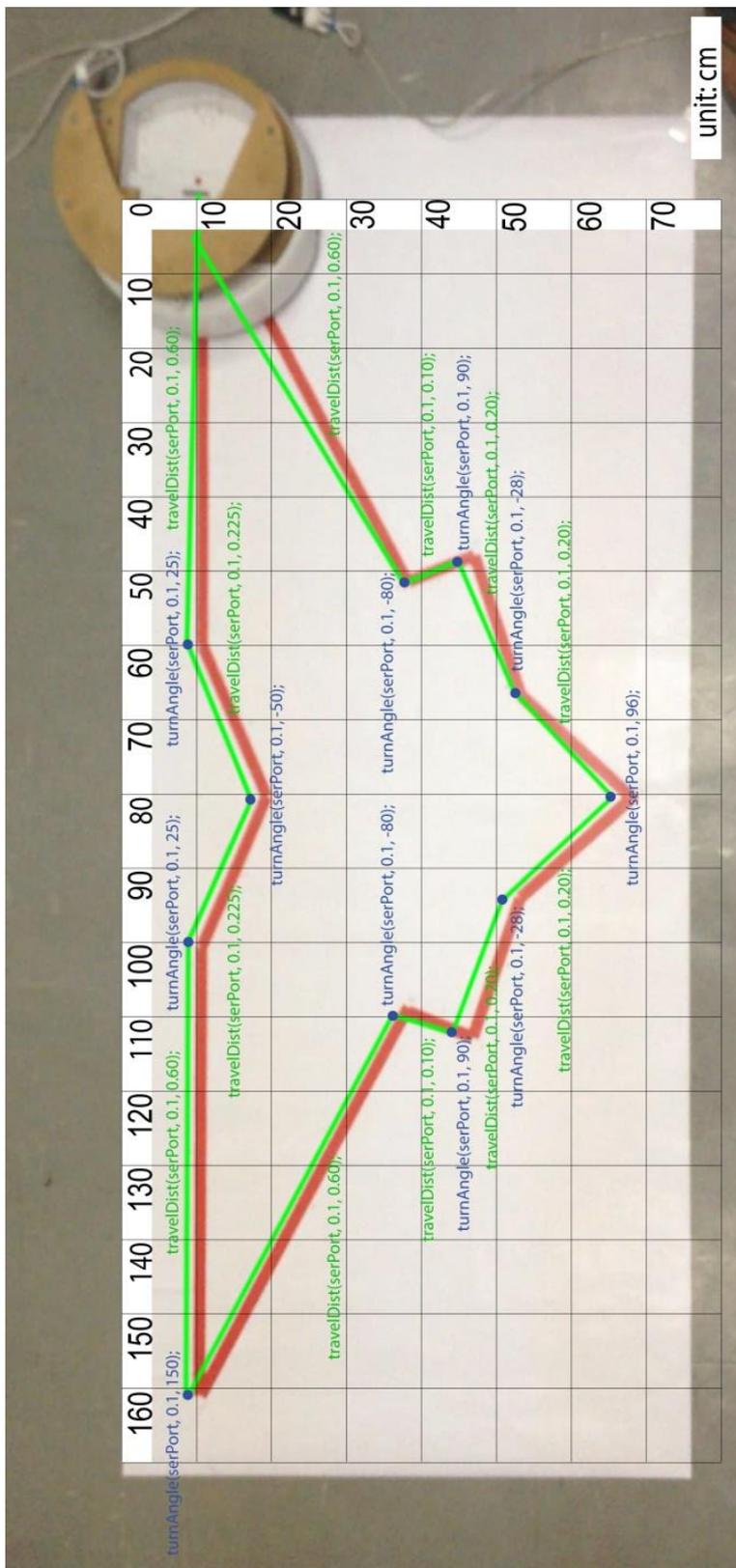
## บทที่ 5

### ผลการทดลอง

#### 5.1 ทดสอบคำสั่งหุ่นยนต์เคลื่อนที่ด้วยล้อ iRobot Create

ทดลองสั่งหุ่นยนต์ให้ไปยังตำแหน่งตามเส้นทางอ้างอิงที่ออกแบบขึ้นมาจากระบบพิกัดระนาบฉาก x-y โดยการแปลงพิกัดจากระนาบ x-y เป็นระนาบวงกลม ได้คำสั่งในโปรแกรมดังกล่าว จากนั้นสั่งขับหุ่นยนต์ตามคำสั่ง พร้อมบันทึกเส้นทางการเคลื่อนที่ของหุ่นยนต์ โดยการถ่ายวิดีโอจากมุมสูงได้เส้นทางดังรูปที่ 23 คำสั่งโปรแกรมหมีดังนี้

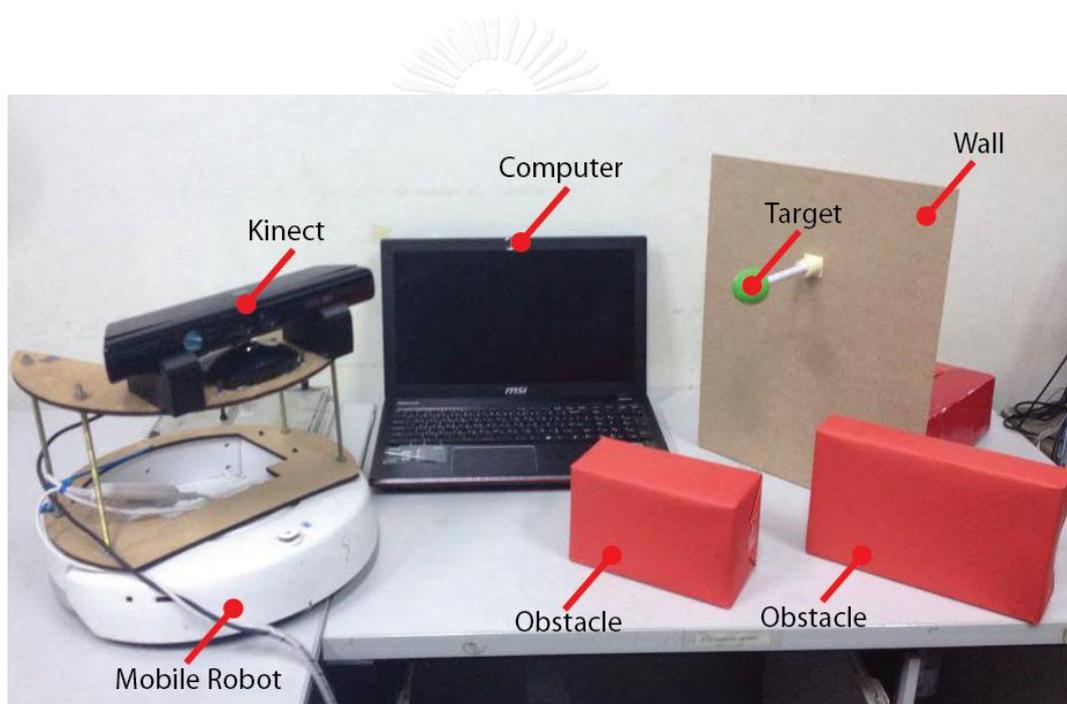
```
serPort = RoombaInit(3);
travelDist(serPort, 0.1, 0.6);
turnAngle(serPort, 0.1, 25); travelDist(serPort, 0.1, 0.225);
turnAngle(serPort, 0.1, -50); travelDist(serPort, 0.1, 0.225);
turnAngle(serPort, 0.1, 25); travelDist(serPort, 0.1, 0.6);
turnAngle(serPort, 0.1, 150); travelDist(serPort, 0.1, 0.6);
turnAngle(serPort, 0.1, -80); travelDist(serPort, 0.1, 0.1);
turnAngle(serPort, 0.1, 90); travelDist(serPort, 0.1, 0.2);
turnAngle(serPort, 0.1, -28); travelDist(serPort, 0.1, 0.20);
turnAngle(serPort, 0.1, 96); travelDist(serPort, 0.1, 0.20);
turnAngle(serPort, 0.1, -28); travelDist(serPort, 0.1, 0.20);
turnAngle(serPort, 0.1, 90); travelDist(serPort, 0.1, 0.10);
turnAngle(serPort, 0.1, -80); travelDist(serPort, 0.1, 0.60);
```



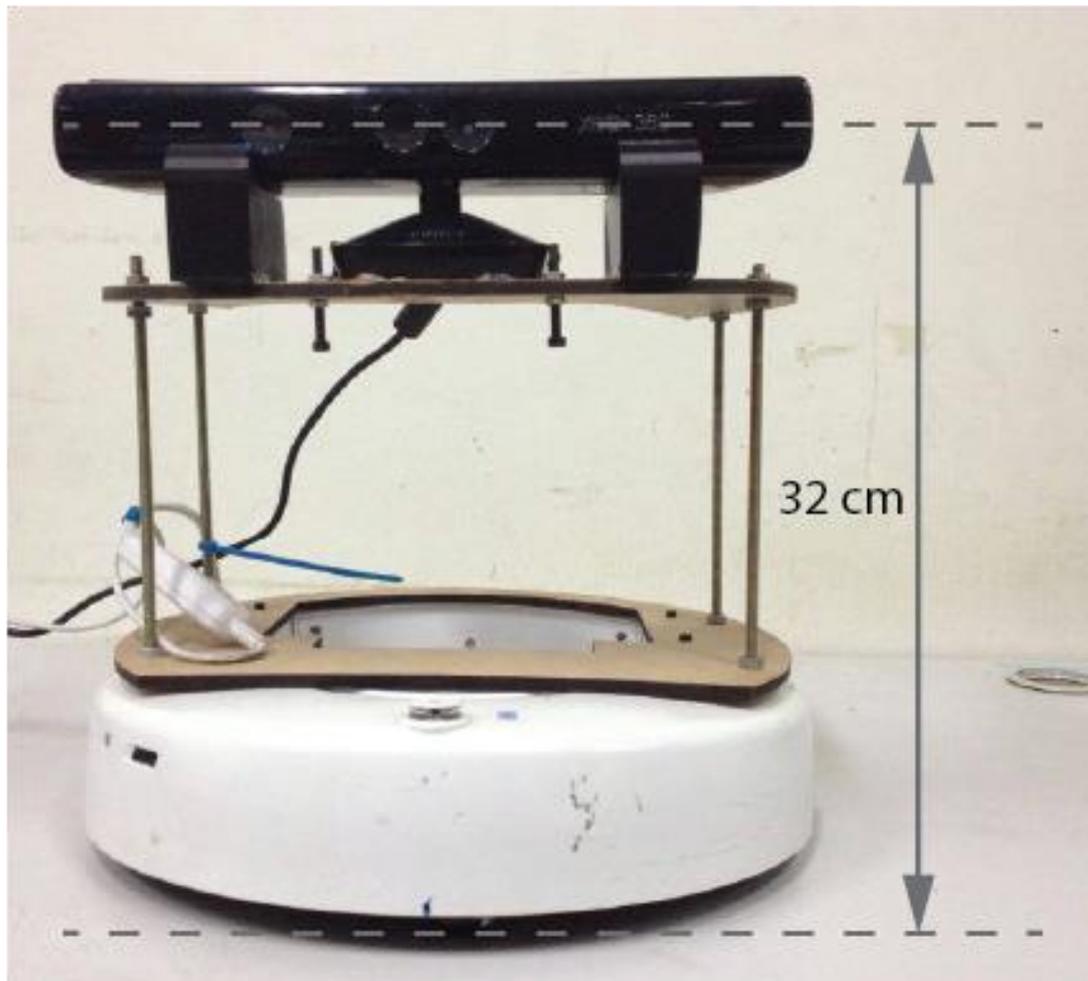
รูปที่ 23 เส้นที่แดงคือเส้นทางเดินอ้างอิง เส้นสีเขียวคือเส้นทางเดินของหุ่นยนต์

## 5.2 การติดตั้งระบบ

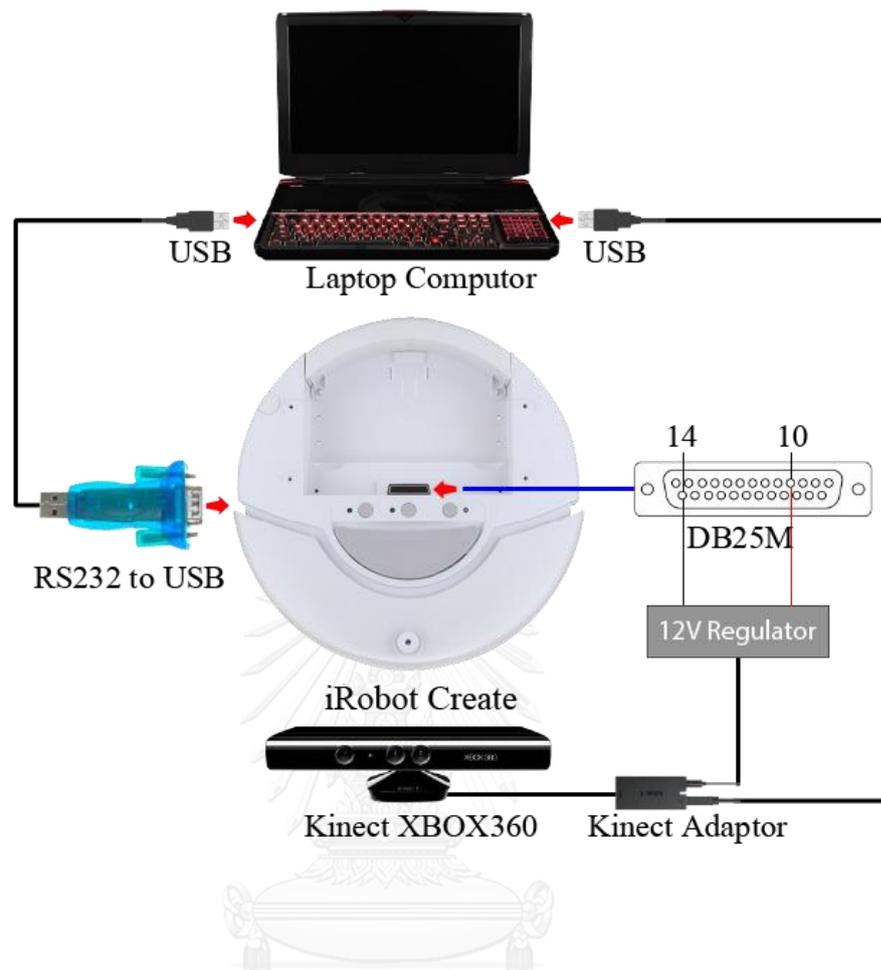
อุปกรณ์ที่ใช้ในงานวิจัยนี้ ได้แก่ 1) หุ่นยนต์เคลื่อนที่ด้วยล้อ iRobot create, 2) อุปกรณ์ตรวจจับ Kinect, 3) คอมพิวเตอร์ Laptop, 4) วัตถุสิ่งกีดขวางเป็นกล่องขนาดไม่เท่ากัน 2 กล่อง และ 5) วัตถุเป้าหมายเป็นวัตถุทรงกลมสีเขียวติดตั้งบนฉากผนังที่ ดังในรูปที่ 24 เริ่มจากการติดตั้งอุปกรณ์ตรวจจับ Kinect ไว้บนหุ่นยนต์ที่ความสูงจากพื้น 32 cm ตามในรูปที่ 25 ทำการเชื่อมต่ออุปกรณ์ตรวจจับ Kinect เข้ากับคอมพิวเตอร์ทางช่อง USB ตามรูปที่ 26



รูปที่ 24 อุปกรณ์ที่ใช้ในการทดลอง



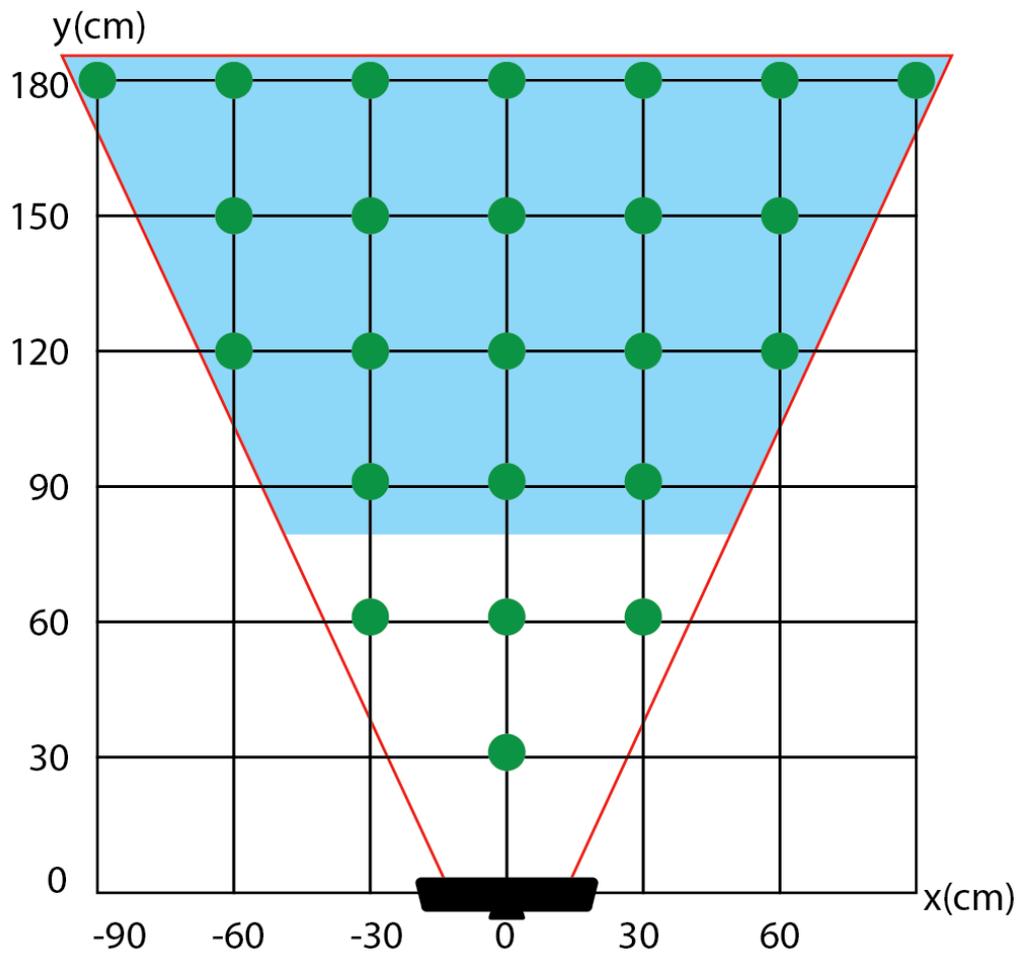
รูปที่ 25 การติดตั้ง Kinect บนตัวหุ่นยนต์ iRobot Create



รูปที่ 26 แผนภาพรวมของระบบเข้าจอดอัตโนมัติ

### 5.3 การทดสอบความแม่นยำของอุปกรณ์ตรวจรู้ Kinect

เช่นเดียวกับอุปกรณ์ตรวจรู้ทั่วไป อุปกรณ์ตรวจรู้ Kinect นั้นไม่สามารถวัดค่าได้อย่างแม่นยำเสมอไป โดยการใช้อุปกรณ์ตรวจรู้กล้อง Kinect ตรวจสอบตำแหน่งของวัตถุเป้าหมายที่ถูกจุดวางไว้ในตำแหน่งที่ทราบค่าแน่นอนหลายตำแหน่ง ตามรูปที่ 27 จากนั้นทำการวัดผลโดยให้อุปกรณ์ตรวจรู้อยู่กับที่ได้ผลลัพธ์ดังนี้

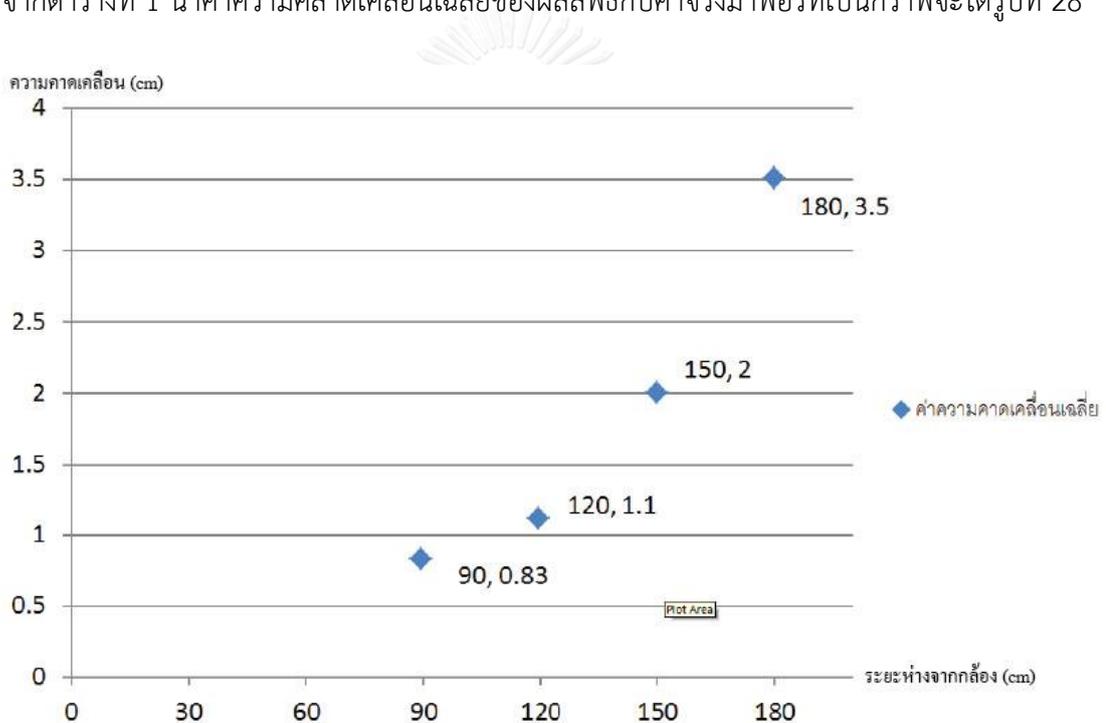


รูปที่ 27 ตำแหน่งของเป้าหมายที่ใช้ในการทดสอบอุปกรณ์ตรวจรู้ Kinect

	-90	-60	-30	0	30	60	90
180	(-85,175)	(-56,176)	(-27,177)	(4,178)	(29,177)	(58,176)	(86,175)
150	unseen	(-57,148)	(-28,149)	(3,148)	(30,147)	(58,148)	unseen
120	unseen	(-58,120)	(-29,120)	(2,119)	(28,121)	(58,120)	unseen
90	unseen	unseen	(-31,90)	(1,89)	(31,89)	unseen	unseen
60	unseen	unseen	-	-	-	unseen	unseen
30	unseen	unseen	unseen	-	unseen	unseen	unseen

ตารางที่ 1 ค่าตำแหน่งในปริภูมิ x,y จากกล้องความลึก

จากตารางที่ 1 นำค่าความคลาดเคลื่อนเฉลี่ยของผลลัพธ์กับค่าจริงมาพอร์ทเป็นกราฟจะได้รูปที่ 28

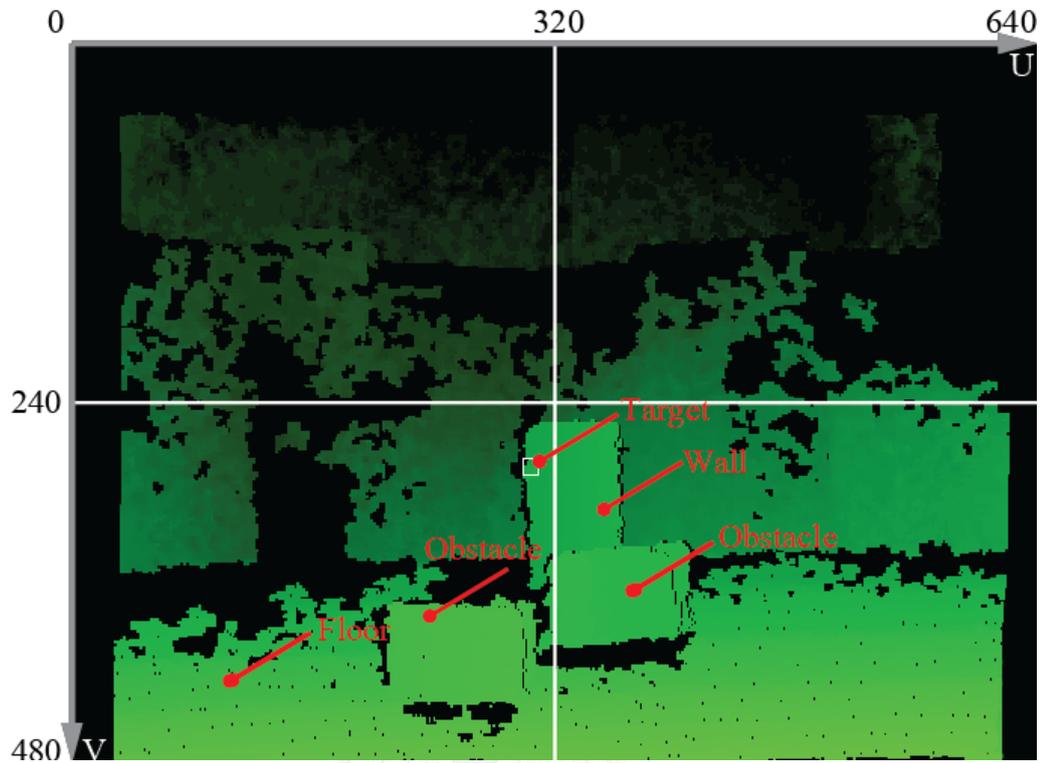


รูปที่ 28 ความความคลาดเคลื่อนเฉลี่ยเทียบกับระยะห่างของเป้าหมาย

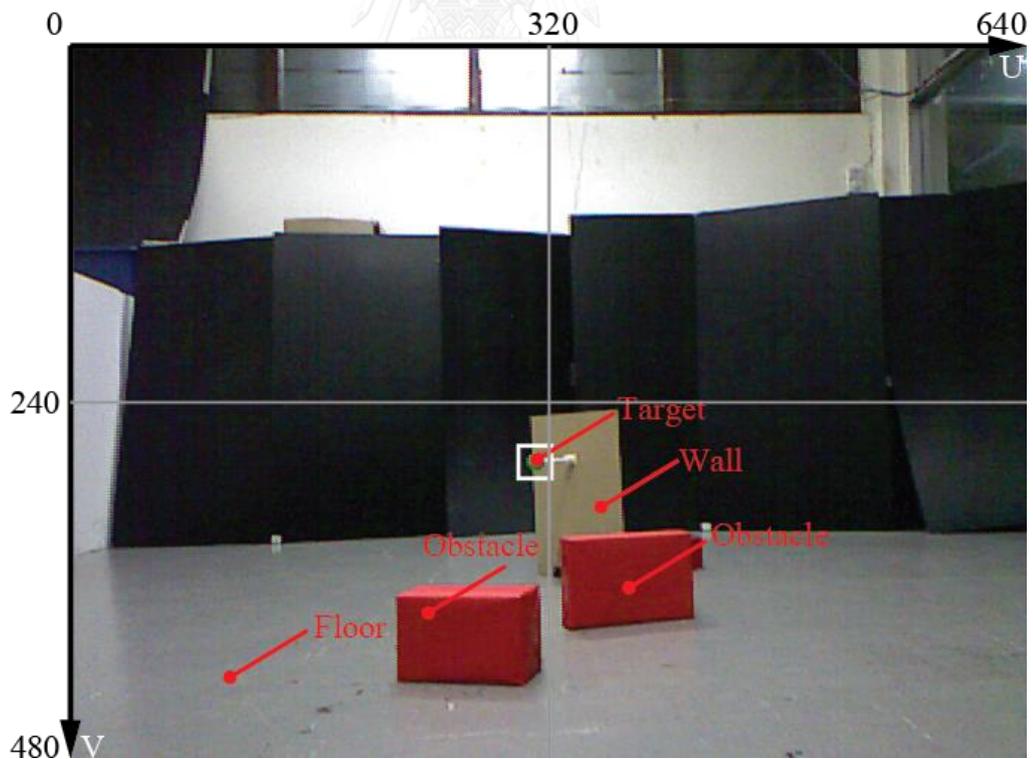
จากรูปที่ 28 เห็นได้ว่าค่าความคลาดเคลื่อนเฉลี่ยอยู่ในช่วง 0.5 ถึง 4 cm โดยค่าความคลาดเคลื่อนแปรผันตรงกับระยะห่างของเป้าหมาย นั่นคือ ยิ่งวัตถุอยู่ไกลจากอุปกรณ์ตรวจรู้เท่าไรค่าความคลาดเคลื่อนยิ่งมากตาม

#### 5.4 การประมวลผลผลลัพธ์จากอุปกรณ์ตรวจรู้ Kinect

เริ่มการทดลอง จัดการวางเป้าหมาย สิ่งกีดขวาง และหุ่นยนต์ลงในพื้นที่ทำการทดลอง โดยให้กล้อง Kinect สามารถมองเห็นวัตถุเป้าหมายในระยะห่างที่พอเพียงที่กล้องความลึก สามารถตรวจจับตำแหน่งเป้าหมายได้ ประมาณ 80 – 200 cm ข้อมูลภาพที่ได้จากกล้องวิดีโอ Kinect จะแบ่งเป็น 2 ส่วน ด้วยกันคือ 1) ภาพสี ในปริภูมิสี RGB ที่ความละเอียด 640x480 pixels ซึ่งประกอบด้วยภาพซ้อนกัน 3 ชั้น (แดง เขียว น้ำเงิน) ซ้อนกันโดยความเข้มของพิกเซลที่เป็นสีขาว แสดงถึงความเข้มของแม่สีนั้นๆ ดังรูปที่ 29 ก 2) ภาพแสดงความลึก ที่ความละเอียด 640x480 pixels โดยแต่ละพิกเซลจะเก็บตำแหน่งในปริภูมิคาร์ทีเซียน ดังรูปที่ 29 ข จุดสีเขียวดังรูปที่ปรากฏในภาพแสดงความลึกนั้นจะเก็บตำแหน่งในพิกัดคาร์ทีเซียนเอาไว้ จะเห็นได้ว่าทั้ง 2 ภาพนั้นมีความเกี่ยวเนื่องกันในพิกัดภาพ (UV) จึงสามารถเทียบตำแหน่งของสี RGB ไปสู่จุดต่างๆที่อยู่บนพิกัดคาร์ทีเซียนได้ (จุดสีเขียว) ทำให้สร้างเป็นภาพจำลองสามมิติสิ่งแวดล้อมที่ตรวจจับได้ ดังรูปที่ 30

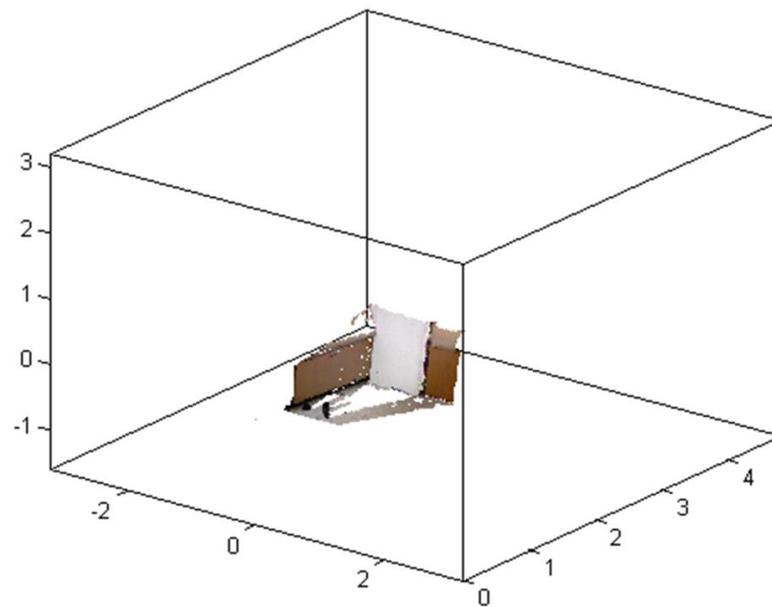


ก. ภาพความลึก จากกล้องความลึก

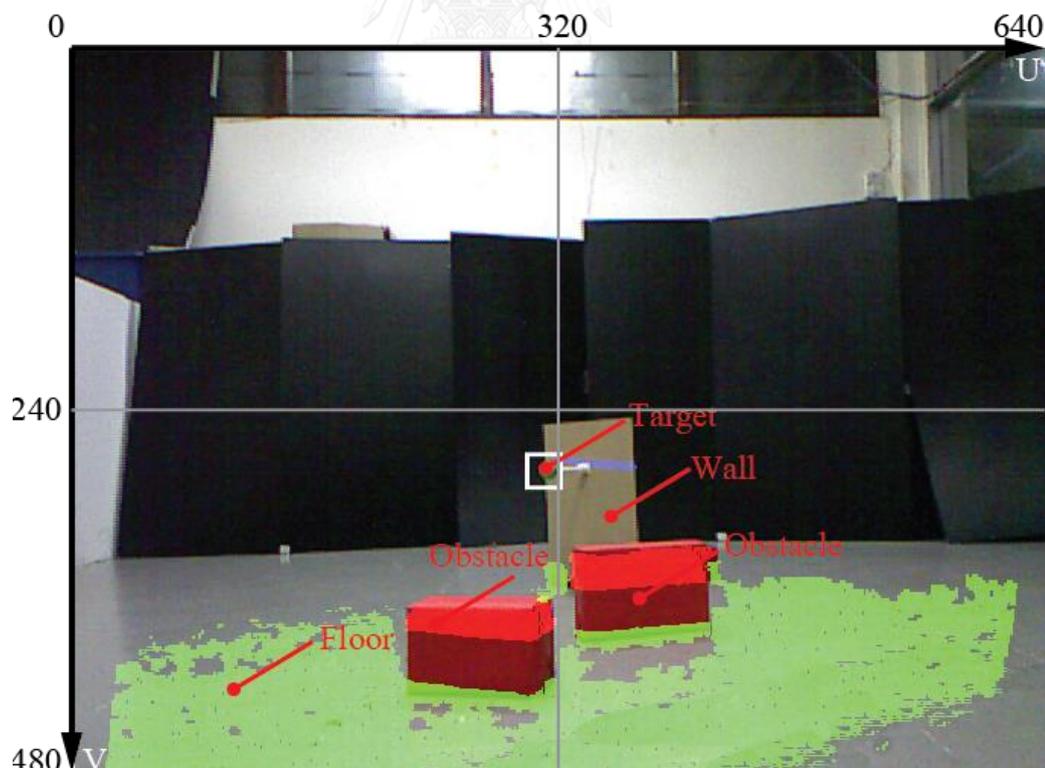


ข. ภาพสี จากกล้องวีดีโอ

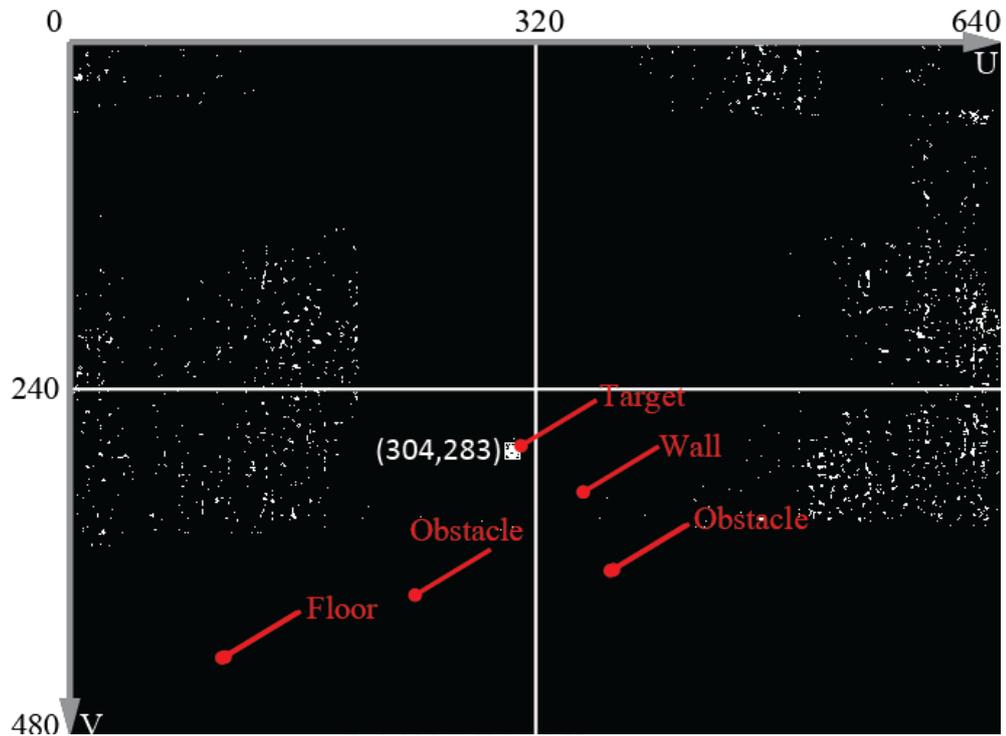
รูปที่ 29 ภาพที่ได้จากการอุปกรณ์ตรวจรู้ Kinect



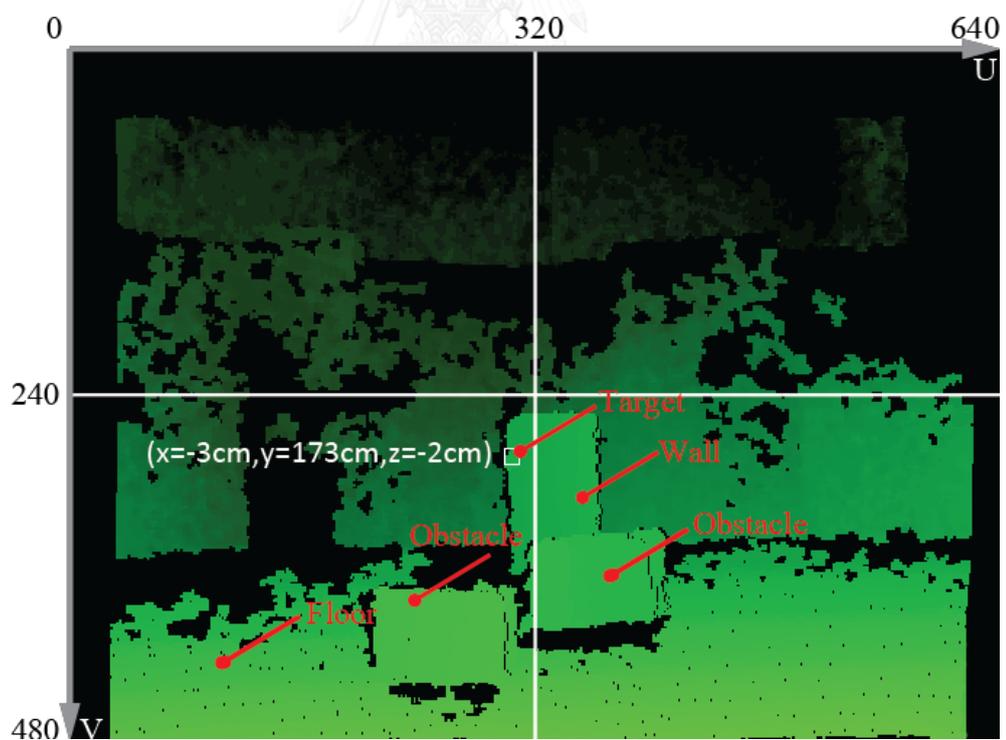
รูปที่ 30 จุดสีของภาพในสามมิติ



รูปที่ 31 รูปที่ได้จากการยึดแยกองค์ประกอบ



ก. ตำแหน่งของวัตถุเป้าหมายที่ได้จากการคัดกรองรายละเอียดภาพ



ข. ตำแหน่งของวัตถุเป้าหมายในภาพความลึก

รูปที่ 32 การหาตำแหน่งของวัตถุเป้าหมายจากภาพของกล้อง Kinect

#### 5.4.1 การแยกแยะองค์ประกอบของสิ่งแวดลอม

ขั้นที่หนึ่ง การคัดแยกองค์ประกอบสิ่งกีดขวางออกจากพื้น เนื่องจากอุปกรณ์ตรวจจู้ Kinect นั้นถูกติดตั้งไว้ที่ระดับความสูงจากพื้นที่ตั้งที่ 32 cm ทำให้สามารถใช้ความสูงของค่าคัดกรองเพื่อแยกแยะองค์ประกอบของสิ่งแวดลอม นั่นคือ จุดอินฟราเรดใดๆก็ตามที่ปรากฏในภาพความลึกที่อยู่ในระดับความสูงในแกน z ต่ำกว่า -30 cm จะให้เป็นพื้นทั้งหมด ส่วนตำแหน่งที่อยู่สูงกว่าในตอนนี้จะให้เป็นสิ่งกีดขวางไปก่อน ดังรูปที่ 31 จะเห็นได้ว่าในรูปนั้นได้ทำการเปลี่ยนของภาพสี RGB ในตำแหน่งพิกเซลในภาพความลึกที่มีตำแหน่งค่า z ต่ำกว่า -30cm เป็นสีเขียว

ขั้นที่สองการคัดแยกองค์ประกอบเป้าหมายและผนังกำแพงออกจากสิ่งกีดขวางตอนนี้จุดความลึก ที่เป็นเป้าหมายและกำแพงนั้นถูกเหมารวมเข้ากับจุดที่เป็นสิ่งกีดขวางไปด้วย โดยมีขั้นตอนดังนี้ ขั้นแรกนำข้อมูลภาพซึ่งต่อนี้อยู่ในปริภูมิ RGB มาแปลงเป็นปริภูมิ HSV เนื่องจากทำให้ง่ายต่อการการคัดกรองลายละเอียดของภาพ ขั้นต่อมา ทำการเลือกช่วงโทนสีในที่นี้คือช่วงสีเขียวตั้งแต่ 0.2-0.3 ของค่า Hue และความสดของสีที่ 0.1 เป็นต้นไป Saturation ส่วนนี้หมายความว่าพิกเซลใดๆก็ตามที่มีโทนสีอยู่ในช่วงของสีเขียวจะให้มีความเท่ากับ 1 (เป็นสีขาว) และพิกเซลอื่นๆมีค่าเป็น 0 (เป็นสีดำ) จากนั้นวิธีการตรวจจับของ Blob เพื่อหาตำแหน่งในพิกัดภาพของเป้าหมาย จะได้ผลลัพธ์ ดังรูปที่ 32 ก จะเห็นได้ว่าตำแหน่งบริเวณเป้าหมายจะเป็นสีขาวมากที่สุด ซึ่งในที่นี้วิธีการตรวจจับของ Blob จับได้ที่ตำแหน่ง (304,283) เมื่อได้ตำแหน่งของเป้าหมายในพิกัดภาพแล้ว จึงไม่ใช่เรื่องยากที่จะหาตำแหน่งในพิกัดคาร์ทีเซียนจากการเทียบกับข้อมูล ภาพความลึก ซึ่งแสดงอยู่ในรูปที่ 32ข แสดงให้เห็นตำแหน่งในพิกัดคาร์ทีเซียนที่ (-3,176,-2) ของ จุดความลึก ณ ตำแหน่งภาพ (304,283)

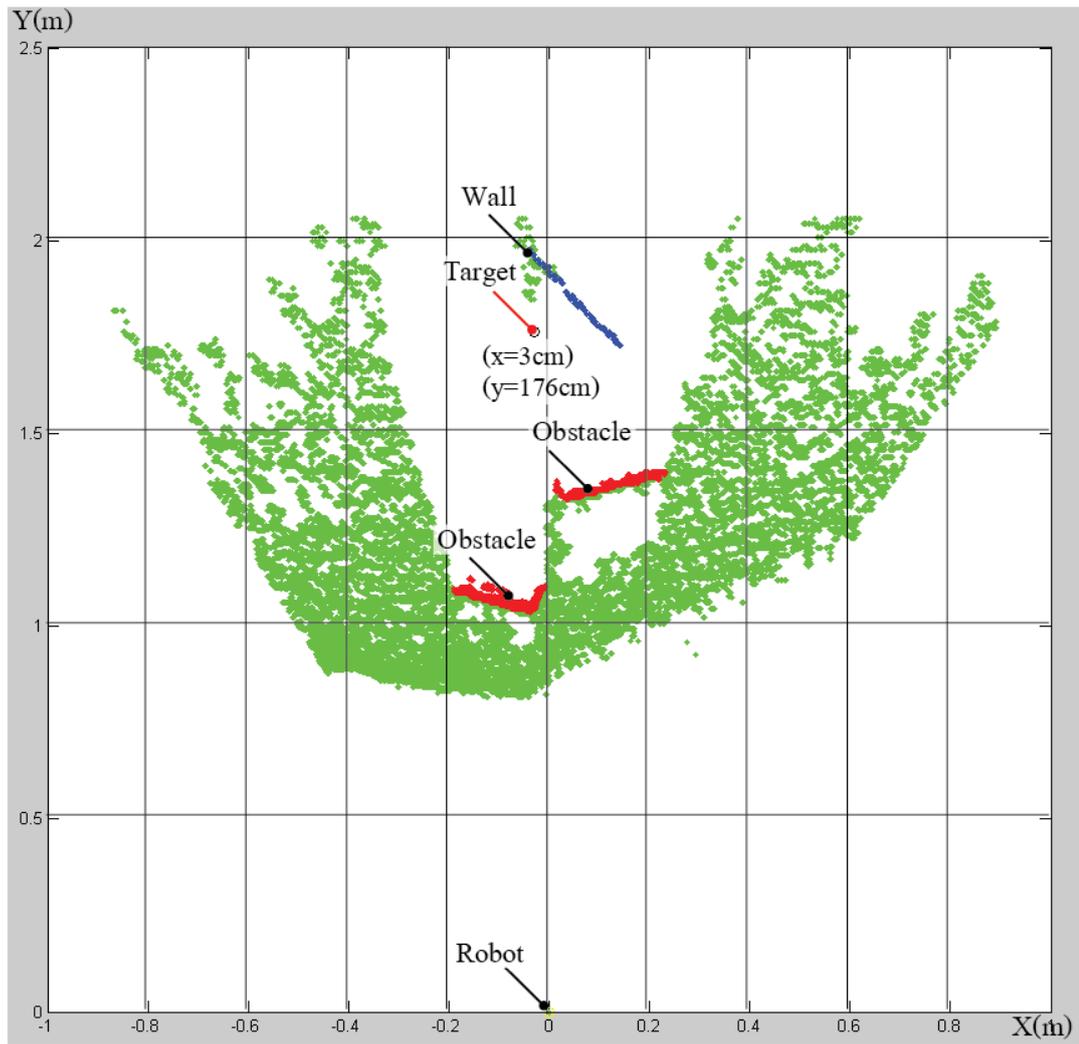
ขั้นตอนต่อมาคือการหาส่วนองค์ประกอบที่เป็นผนังกำแพงของวัตถุเป้าหมาย ซึ่งทำได้ไม่ได้อย่างเมื่อได้ทราบตำแหน่งของเป้าหมายพิกัดคาร์ทีเซียนแล้ว เพราะว่าตำแหน่งของวัตถุเป้าหมายและผนังกำแพงของเป้าหมายนั้นเป็นอุปกรณ์ชิ้นเดียวกันจึงมีความสัมพันธ์เชิงสถิติต่อกันและกัน ทำให้สามารถหาองค์ประกอบส่วนที่เป็นผนังกำแพงได้โดยการกำหนดจุดความลึก ใดๆก็ตามที่ไม่ใช่จุดที่เป็นพื้นและที่อยู่ในบริเวณรัศมี 10 cm รอบๆ วัตถุเป้าหมายเป็นกำแพงทั้งหมด เพียงเท่านี้ก็สามารถแยกแยะองค์ประกอบของสิ่งแวดลอมนั้นก็คือ พื้น สิ่งกีดขวาง เป้าหมาย และผนังกำแพงของเป้าหมายได้แล้ว

สิ่งสุดท้ายคือการเปลี่ยนสีของภาพเพื่อบอกว่าเป็นอะไรในพิกัดภาพ โดยการเทียบค่าจากพิกัดคาร์ทีเซียนกลับมาเป็นพิกัดภาพในภาพแล้วตั้งค่าพิกเซลในพิกัดภาพ RGB ให้แสดงความเด่นชัดขึ้นมา ในที่นี้คือ บริเวณที่เป็นพื้นจะให้ค่าพิกเซลของปริภูมิสีเขียวมีค่าเป็น 255, บริเวณที่เป็นสิ่งกีดขวางจะให้ค่าพิกเซลของปริภูมิสีแดงมีค่าเป็น 255 และ บริเวณที่เป็นผนังกำแพงของเป้าหมายจะให้ค่าพิกเซลของปริภูมิสีน้ำเงินมีค่าเป็น 255 จากนั้นรวมทั้งสามปริภูมิลบกลับจะเป็นภาพสี RGB ดังรูปที่ 31 เห็นได้ว่าพื้นคือส่วนที่เป็นสีเขียว, สิ่งกีดขวางคือส่วนที่เป็นสีแดง และผนังกำแพงของเป้าหมายคือส่วนที่เป็นสีน้ำเงิน ส่วนเป้าหมายนั้นได้เก็บค่าที่แน่นอนไว้แล้ว

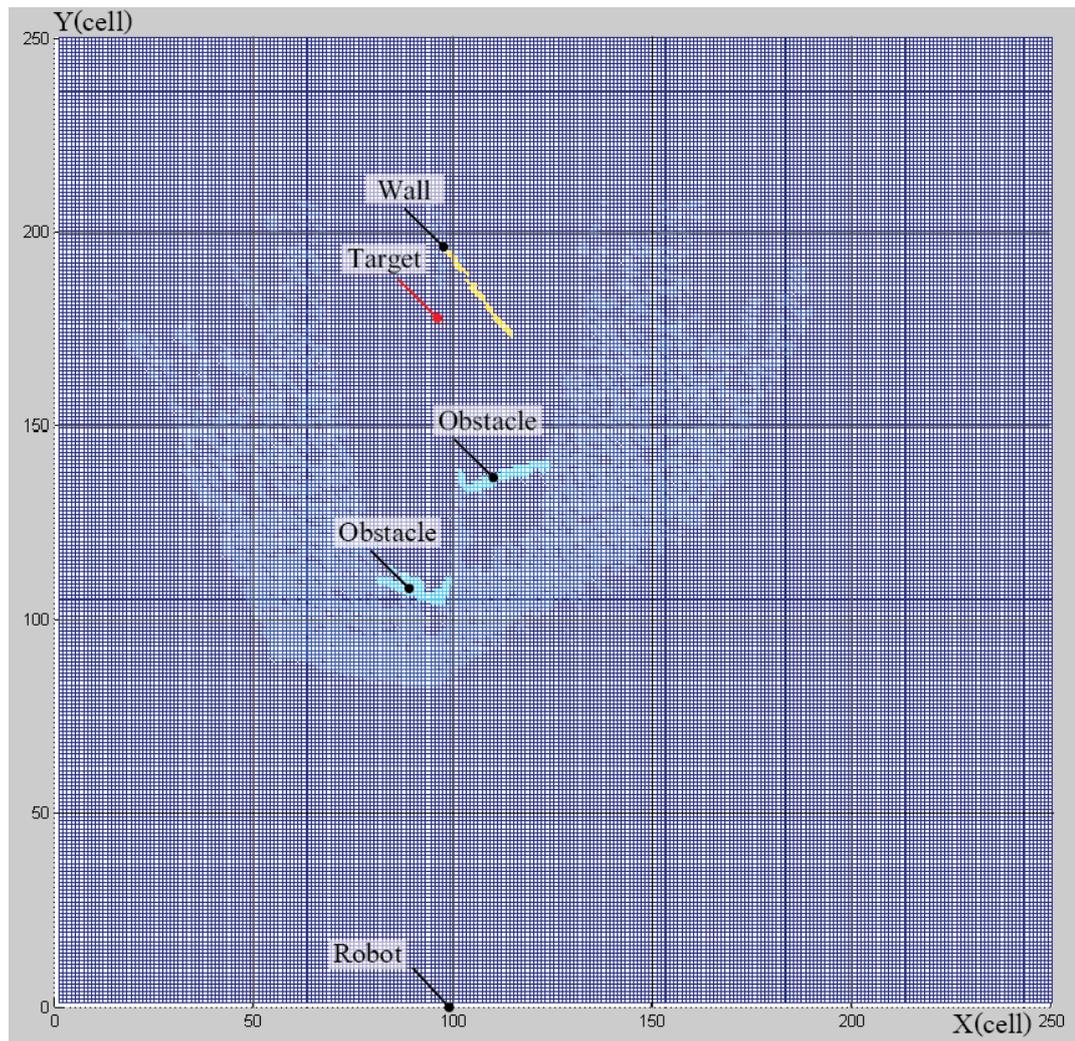
#### 5.4.2 การสร้างแผนที่ และการปรับปรุงแผนที่ให้สามารถใช้ประโยชน์ได้

จุดที่อยู่ในภาพความลึก ทุกๆจุดได้ถูกคัดแยกออกจากกันเป็นกลุ่มๆอย่างชัดเจนแล้ว นั่นคือกลุ่มจุดที่เป็นสิ่งกีดขวางสีแดง, กลุ่มจุดที่เป็นพื้นสีเขียว และกลุ่มจุดที่เป็นผนังกำแพงของวัตถุเป้าหมาย ส่วนตำแหน่งวัตถุเป้าหมายนั้นถูกเก็บไว้ตั้งแต่แรกแล้ว เนื่องจากหุ่นยนต์เคลื่อนที่ด้วยล้อนั้นเคลื่อนที่ได้เพียง 2 มิติ นั่นคือ ในแกน X และแกน Y ทำให้ค่าในแกน Z ของกลุ่มจุดความลึก ไม่จำเป็นต้องนำมาพิจารณา เมื่อนำจุดความลึกมาพล็อตในกราฟได้เป็นแผนที่ ดังรูปที่ 33 ถึงกระนั้นข้อมูลที่มีอยู่ขณะนี้ยังไม่สามารถนำไปใช้ประโยชน์ได้เนื่องจากปริมาณข้อมูลที่ได้จากกล้องความลึกนั้นมีปริมาณมากกว่า 300,000 ข้อมูล ที่เป็นเช่นนี้เป็นเพราะว่าทุกๆจุดในแกน Z ได้ถูกรวมด้วย ซึ่งมากเกินไปที่คอมพิวเตอร์ที่ใช้อยู่ในขณะนี้จะคำนวณได้

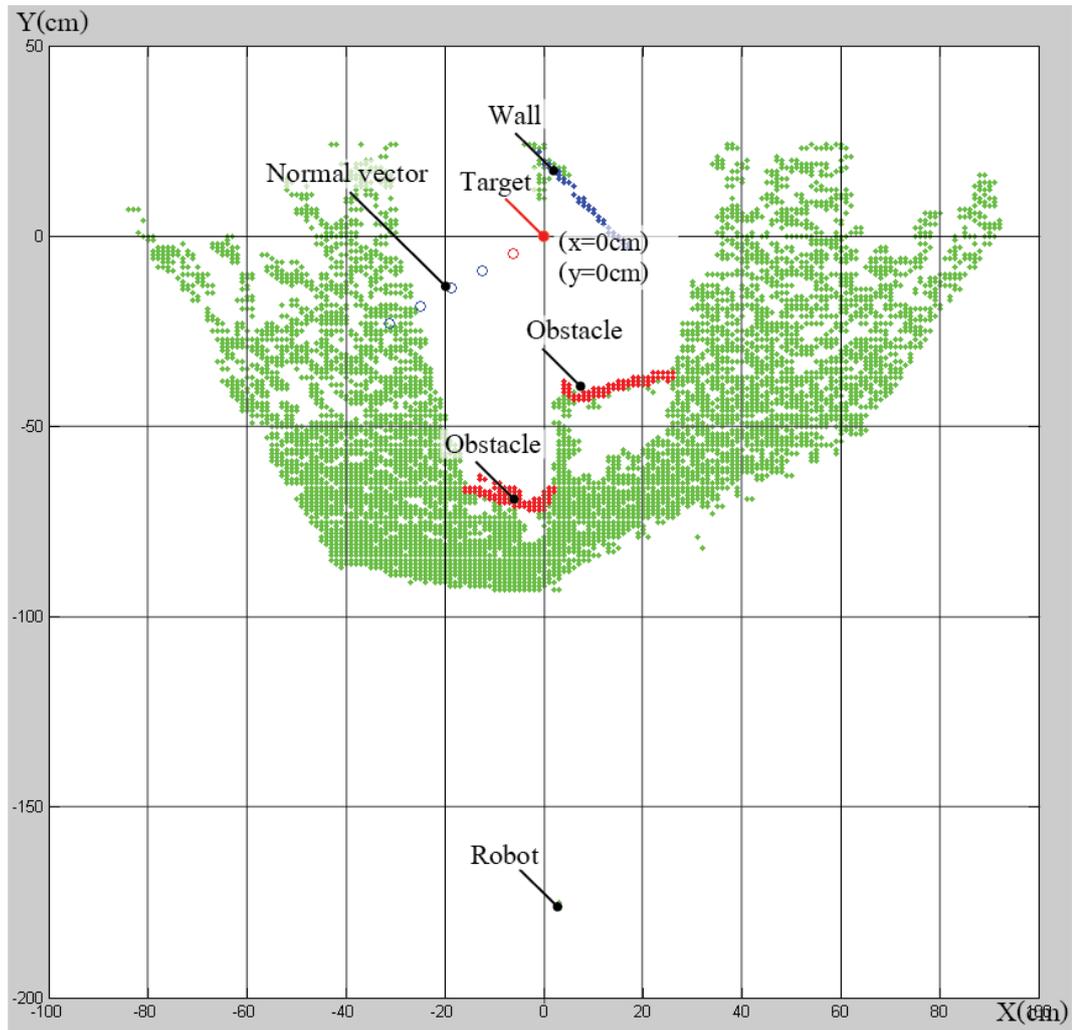
เพื่อลดปริมาณข้อมูลให้ลดลงแต่ยังคงคุณภาพความถูกต้องของข้อมูลไว้ด้วยวิธี การแทนที่ตำแหน่งด้วยตารางแผนที่ (Grid Occupation) ในที่นี้ให้ขนาดเซลล์ตารางแต่ละเซลล์มีความกว้าง 1 cm ดังรูปที่ 34 หมายความว่าทุกจุดที่อยู่ในพื้นที่เซลล์ตารางนั้นนับเป็น 1 จุด ผลที่ได้คือปริมาณข้อมูลที่ได้นั้นลดลงอย่างเห็นได้ชัดเจนจากหลักแสนเหลืออยู่ที่หลักพันข้อมูล จากนั้นทำการเทียบตำแหน่งของจุดในตารางกลับไปยังแผนที่ แต่ตอนนี้จะเลื่อนให้ตำแหน่งของจุดทุกๆจุด โดนให้จุดเป้าหมายเป็นจุด (0,0) จะได้ผลลัพธ์ดังรูปที่ 35 จะสังเกตเห็นได้จุดต่างๆในแผนที่ตอนนี้เรียงกันอย่างเป็นระเบียบไม่เหมือนแผนที่เก่า เนื่องจากผลของการทำ Grid Occupation และจุด (0,0) ถูกย้ายไปอยู่ที่ตำแหน่งของเป้าหมาย



รูปที่ 33 แผนที่ที่ได้จากการแยกแยะส่วนประกอบ



รูปที่ 34 แผนที่ที่ได้จากระบวนการ grid occupation



รูปที่ 35 แผนที่หลังจากทำ grid occupation และ กำหนดให้เป้าเป็นจุด (0,0)

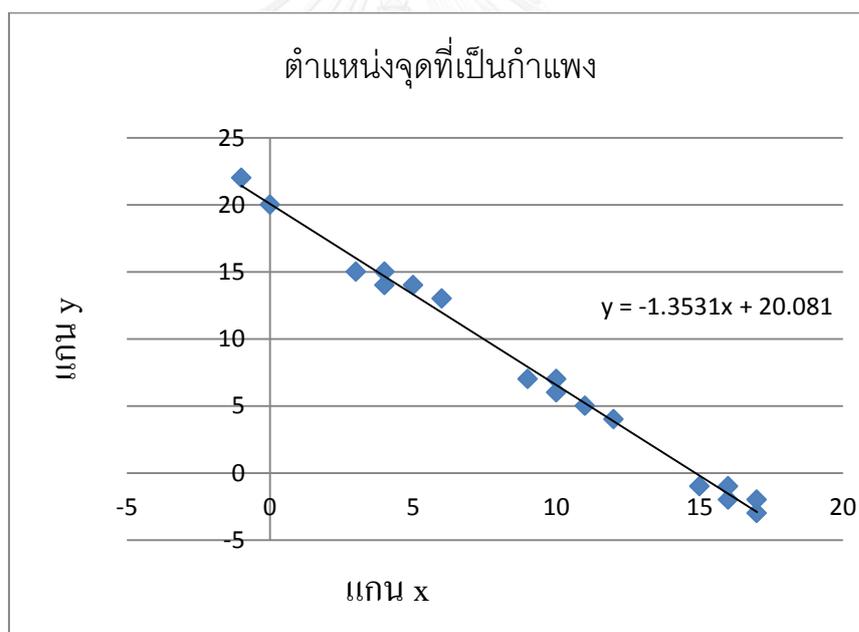
#### 5.4.2 การหาเส้นตั้งฉากกับกำลัง

เพื่อให้หุ่นยนต์เข้าจอดโดยหันหน้าตั้งฉากกับกำลังสามารถหาได้จากการหาสมการที่ตั้งฉากกับสมการเส้นตรงของจุดที่เป็นผนังกำลังของเป้าหมายบนแผนที่ หาเส้นตรงของกลุ่มจุดผนังกำลังของเป้าหมาย ด้วยสมการความสัมพันธ์การถดถอยแบบเชิงเส้น  $y(x) = a_0 + a_1x$  ดังรูปที่ 36

$$a_1 = \frac{n \left( \sum_{i=1}^n x_i y_i \right) - \left( \sum_{i=1}^n x_i \right) \left( \sum_{i=1}^n y_i \right)}{n \left( \sum_{i=1}^n x_i^2 \right) - \left( \sum_{i=1}^n x_i \right)^2}$$

$a_0$  คือ ค่าคงที่จุดตัดแกน y

$a_1$  คือ ค่าความชันของเส้นตรงที่เกิดจากกลุ่มจุดผนังกำลัง



รูปที่ 36 เส้นตรงที่ได้จากจุดที่เป็นกำแพง

จากการคำนวณได้ค่า  $a_1 = -1.3531$ ,  $a_0 = 20.081$

หาความชันของเส้นตรงตั้งฉากจากผนังกำลัง  $slope = -(a_1)^{-1}$

เส้นตรงที่ตั้งฉากกับผนังกำลังที่ผ่านจุด (0,0) จะมี สมการ  $y = 0.7390x$

### 5.4.3 การสร้างเส้นทางเดินหุ่นยนต์ด้วยสนามศักย์

เพื่อให้ง่ายต่อการคำนวณและความเข้าใจแยกพิจารณา ชนิดของสนามศักย์ออกเป็น 3 ส่วน คือ 1) ส่วนที่เป็นผลมาจากกลุ่มจุดที่เป็นสิ่งกีดขวาง 2) ส่วนที่เป็นผลมาจากเป้าหมาย 3) ส่วนที่เป็นผลมาจากกลุ่มจุดที่เป็นกำแพง โดยที่หาขนาดของสนามศักย์ในบริเวณรอบพื้นที่ที่ทำการทดลองเป็นของเขตสี่เหลี่ยมจัตุรัส  $2.5 \text{ m} \times 2.5 \text{ m}$  รอบๆ เป้าหมาย เพื่อให้สะดวกต่อการสร้างสมการจึงกำหนด  $d$  คือค่าระยะห่างจากตำแหน่งที่สนใจ ถึงจุดกำเนิดสนามศักย์ โดยที่

$$\vec{d}(x, y) = \sqrt{(x_i - x)^2 + (y_i - y)^2} \hat{r}$$

$$|\vec{d}|(x, y) = \sqrt{(x_i - x)^2 + (y_i - y)^2}$$

$x$  และ  $y$  คือค่าตำแหน่งที่กำลังสนใจ

$x_i$  และ  $y_i$  คือค่าตำแหน่งของจุดกำเนิดสนามศักย์

$i = 1, 2, 3, \dots, n$  โดยที่  $n$  คือจำนวนจุดกำเนิดสนามศักย์

$\hat{r}$  คือ เวกเตอร์ 1 หน่วยบอกทิศทางจากตำแหน่งที่สนใจไปยังตำแหน่งกำเนิดสนามศักย์

สร้างสนามศักย์ของสิ่งกีดขวาง

เป้าหมายหลักของการทดลองนี้คือการให้หุ่นยนต์เคลื่อนที่หลบหลีกสิ่งกีดขวางเพื่อไปจอดหน้าเป้าหมาย เพราะฉะนั้นหุ่นยนต์จะต้องหลีกเลี่ยงการปะทะกับสิ่งกีดขวาง จึงกำหนดให้สิ่งกีดขวางมีสนามศักย์ออกแรงผลัก หรือให้มีลักษณะเป็นเนินสูง แต่เนื่องจากจุดกำเนิดสนามศักย์มีหลายจุด จึงพิจารณาทีละจุดเป็นจุดๆ ไป แล้วนำมารวมกันเป็นเวกเตอร์รวม ณ ตำแหน่งที่กำลังสนใจนั้นๆ งานวิจัยนี้เลือกใช้ ฟังก์ชัน

$$\vec{F}_i(x, y) = -K / |\vec{d}|^2 \hat{r}$$

$$\vec{F}(x, y) = \sum_{i=1}^n \vec{F}_i(x, y)$$

โดยที่  $K$  คือค่าคงที่ความแรงของสนามศักย์

สำหรับการทดลองนี้ให้  $K$  เท่ากับ 30 จะได้สนามศักย์ ดังรูปที่ 37 ก เป็นกราฟแสดงลักษณะทางการภาพของสนามศักย์ ของสิ่งกีดขวางโดยแกนแนวตั้งคือระดับแรงผลึก เมื่อมองตามกราฟ X-Y ตามรูปที่ 38 ก จะแสดงระดับแรงผลึกนั้นดูแทนที่ด้วยสีจากแดงไปยังสีน้ำเงิน อีกนัยหนึ่งคือจากมากไปน้อย ตามลำดับ

สร้างสนามศักย์ของเป้าหมาย

ด้วยหลักการคล้ายๆกัน กับการสร้างสนามศักย์แรงผลึก ขณะเดียวกันต้องให้หุ่นยนต์เคลื่อนที่เข้าเป้าหมายด้วยจึงกำหนดให้เป้าหมายมีสนามศักย์ออกแรงดึงดูดหรือให้เป็นกรวยหงายโดยจุดต่ำสุดของกรวย อยู่ที่ตำแหน่ง (0,0)นั่นหมายความว่า ยิ่งตำแหน่งไกลออกไปจากจุด (0,0) เท่าไรแรงดึงดูดยิ่งมากและเนื่องจากเป้าหมายเป็นจุดเดี่ยว จึงกำหนดให้ฟังก์ชันที่เป็นดังนี้

$$\vec{F}(x, y) = K \frac{\vec{d}}{|\vec{d}|^2}$$

ในที่นี้ให้ค่า  $K$  เท่ากับ 2 จะได้ลักษณะสนามศักย์ ดังรูปที่ 37 ข เป็นกราฟแสดงลักษณะทางการภาพของสนามศักย์ ของวัตถุเป้าหมายโดยแกนแนวตั้งคือระดับแรงผลึก รูปที่ 38 ข แสดงลักษณะสนามศักย์ดึงดูดมองตามกราฟ X-Y

สร้างสนามศักย์ของกำแพง

เพื่อให้การเข้าจอดของหุ่นยนต์หันหน้าตั้งฉากเข้าหากำแพงของวัตถุเป้าหมาย จึงเพิ่มสนามศักย์ที่มีลักษณะสอดคล้องกับแนวตั้งฉากของกำแพงขึ้น นั่นคือเป็นสนามศักย์ที่มีทิศพุ่งเข้าหาเวกเตอร์ตั้งฉากนั่นเอง หรือกล่าวอีกนัยหนึ่งก็คือมีทิศทางแนวเดียวกับสมการฟังก์ชันเส้นตรงที่เกิดจากข้อมูลจุดที่เป็นกำแพงนั่นเอง เพราะฉะนั้นลักษณะสนามศักย์ก็ควรจะเป็นทรงรูปตัววี “V” เหมือนร่อง จากสมการเส้นตรงที่เกิดจากกลุ่มจุดกำแพง

$$y_{wall} = -1.3531x + 20.081$$

และสมการเส้นตรงตั้งฉากกำแพงคือ

$$y_{normal} = 0.7390x$$

จะเห็นได้ว่า  $-1.3531$  คือความชันของเส้นตรง จะได้สมการฟังก์ชันดังนี้

$$\vec{F}(x, y) = K(-1.3531\hat{x} + \hat{y}) \quad \text{เมื่อ } y_{normal} > y$$

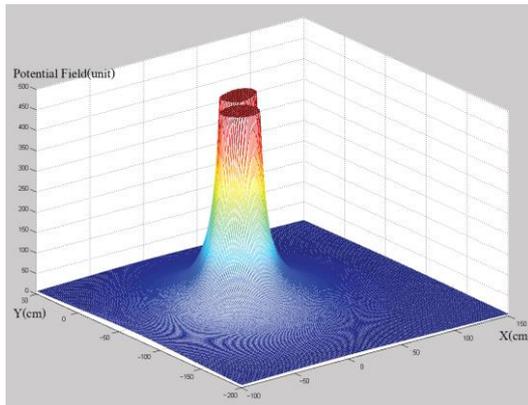
$$\vec{F}(x, y) = K(1.3531\hat{x} + \hat{y}) \quad \text{เมื่อ } y_{normal} < y$$

โดยที่  $\hat{x}$  และ  $\hat{y}$  คือเวกเตอร์ 1 หน่วยบนแกน  $x$  และ  $y$  ตามลำดับ

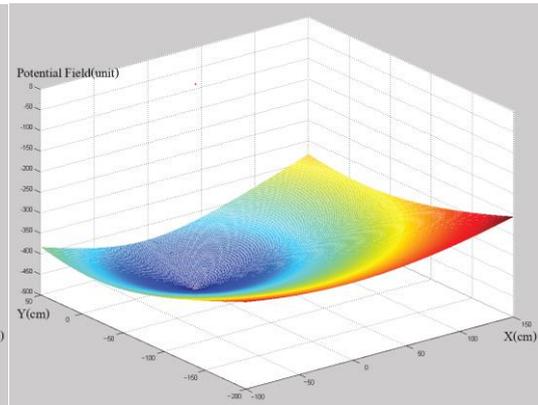
ในที่นี้ให้ค่า  $K$  เท่ากับ 1 จากสมการจะสร้างสนามศักย์ได้ ดังรูปที่ 37 ค เป็นกราฟแสดงลักษณะทางการภาพของสนามศักย์ ของวัตถุเป้าหมายโดยแกนแนวตั้งคือระดับแรงผลึก รูปที่ 38 ค แสดงลักษณะสนามศักย์ที่ตั้งดูมองตามกราฟ X-Y

การรวมสร้างสนามศักย์

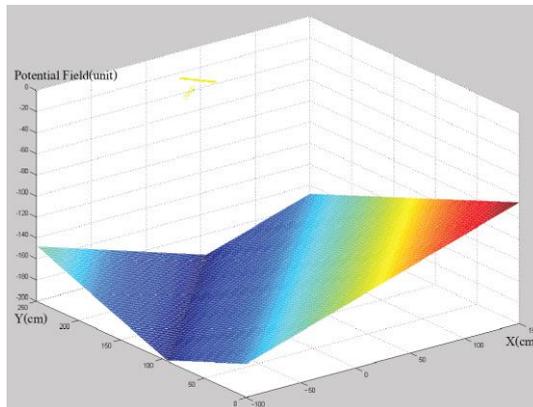
เมื่อได้สนามศักย์ครบแล้วทำการรวมสนามศักย์ทั้งหมดเข้าด้วยกัน จะได้เป็นสนามศักย์ใหม่ ดังรูปที่ 37 ง และรูปที่ 38 ง จากนั้นทำการจำลองการเคลื่อนที่ของหุ่นยนต์ โดยกำหนดให้ธรรมชาติ การเคลื่อนที่จากบริเวณที่ศักย์สูงไปยังบริเวณที่มีศักย์ต่ำ เหมือนกับลูกบอลที่ไหลลงตามเนิน ซึ่งจะได้เส้นทางการเคลื่อนที่ตามรูปที่ 39 แล้วทำการบันทึกตำแหน่งของเส้นทางของหุ่นยนต์ไว้



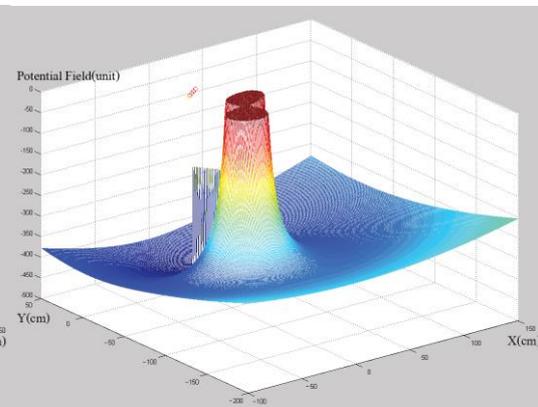
ก. สนามศักย์ไฟฟ้าหลักของสิ่งกีดขวาง



ข. สนามศักย์ไฟฟ้าดึงดูดของเป้าหมาย

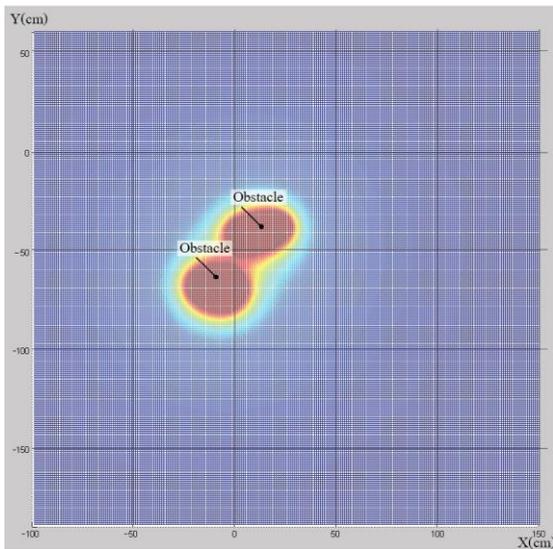


ค. สนามศักย์ไฟฟ้าดึงดูดแนวตั้งฉากของเส้นแนวนอน

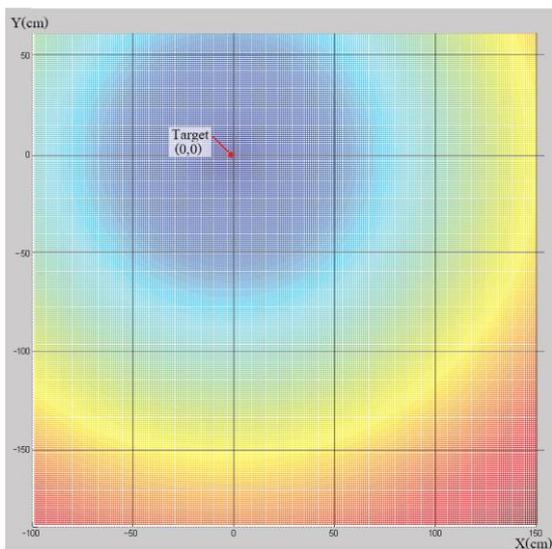


ง. สนามศักย์รวม

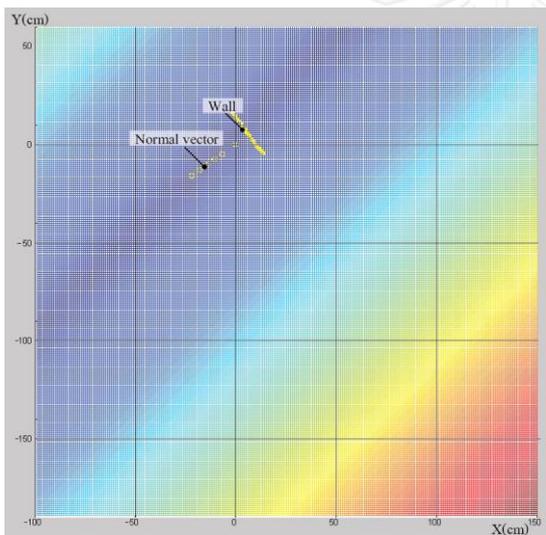
รูปที่ 37 สนามสนามศักย์จากมุมมองทั้ง 3 มิติ



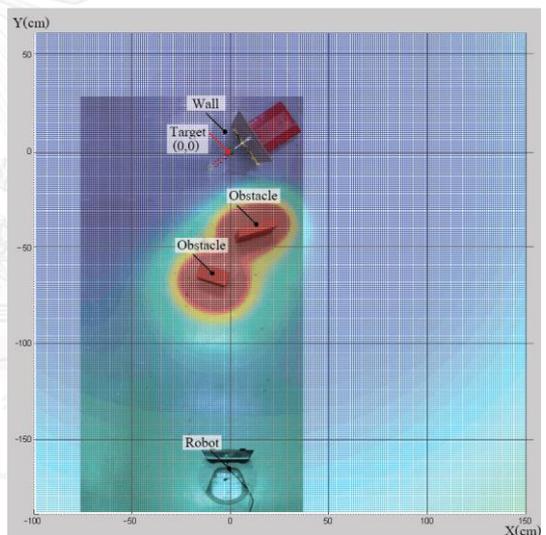
ก. สนามศักย์หลักของสิ่งกีดขวาง



ข. สนามศักย์ดึงดูดของเป้าหมาย

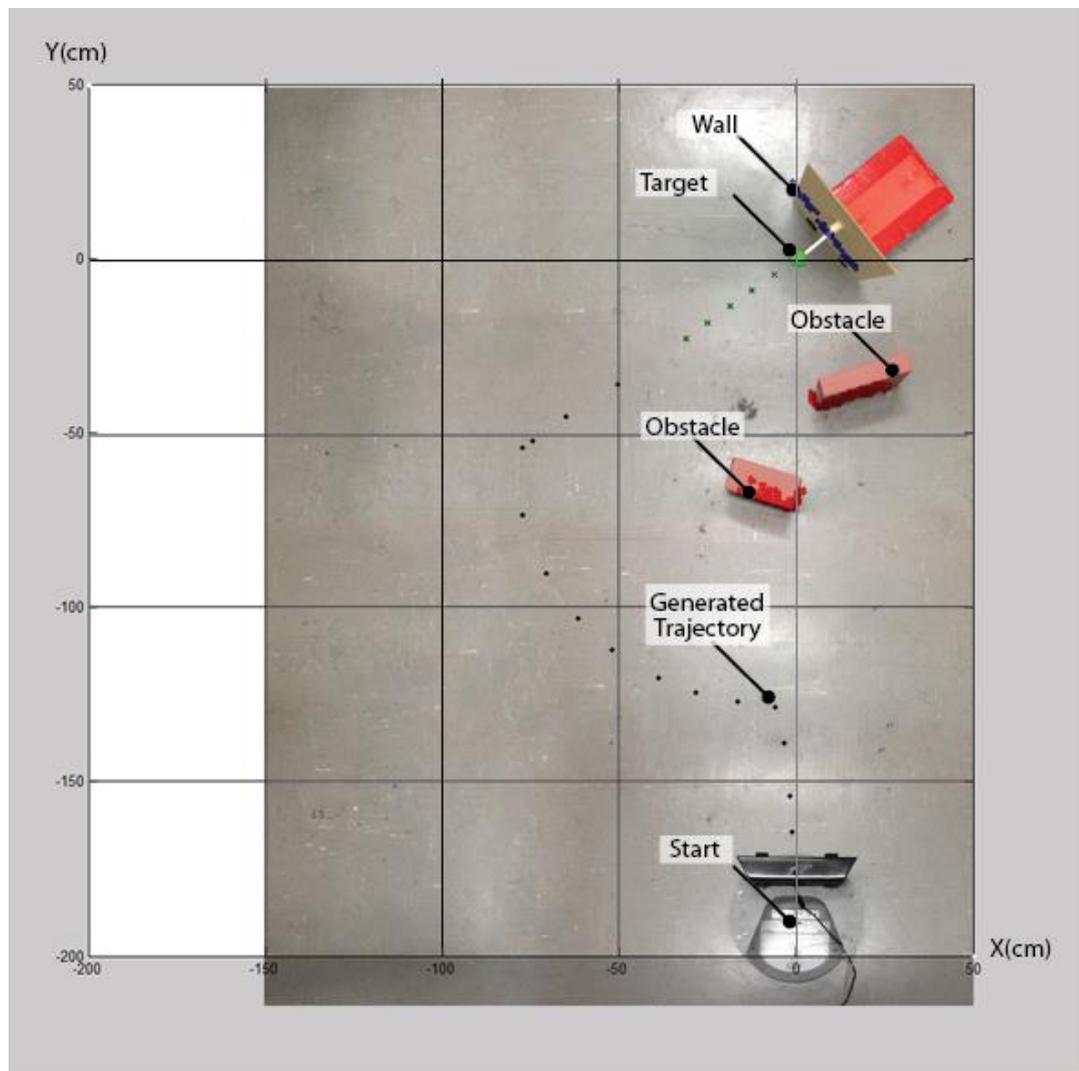


ค. สนามศักย์ดึงดูดแนวตั้งฉากของเส้นแนวนาน



ง. สนามศักย์รวม

รูปที่ 38 สนามสนามศักย์ดีตามแกน x-y



รูปที่ 39 เส้นทางเดินของหุ่นยนต์ที่ได้

### 5.5 แปลงเส้นทางการเคลื่อนที่ให้เป็นคำสั่งควบคุมหุ่นยนต์

จากรูปที่ 39 จะเห็นได้ว่าจุดสีดำเป็นจุดที่กำหนดไว้ให้หุ่นยนต์ต้องเดินทางผ่าน ซึ่งหุ่นยนต์จะเคลื่อนที่เป็นเส้นตรงไปที่ละจุด  $(x_i, y_i), (x_{i+1}, y_{i+1}), \dots$  เหมือนเวกเตอร์สามารถสร้างเวกเตอร์  $\vec{v}_i, \vec{v}_{i+1} \dots$  ตามลำดับ และ  $\vec{v}_{i+1}$  คือเวกเตอร์แสดงความสัมพันธ์ระหว่างจุดใหม่กับจุดเก่า เพื่อที่จะสั่งการหุ่นยนต์จำเป็นต้องหา

$$1) \text{ ระยะทางที่หุ่นยนต์ต้องเคลื่อนที่การขนาดของเวกเตอร์ } |\vec{v}_{i+1}| = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

$$2) \text{ หามุมจากความสัมพันธ์ของเวกเตอร์ดังนี้ } \theta_{i+1} = \arccos \left( \frac{\vec{v}_{i+1} \cdot \vec{v}_i}{|\vec{v}_{i+1}| |\vec{v}_i|} \right)$$

ได้ผลลัพธ์เป็นชุดตัวแปรสำหรับคำสั่งดังตารางที่ 2

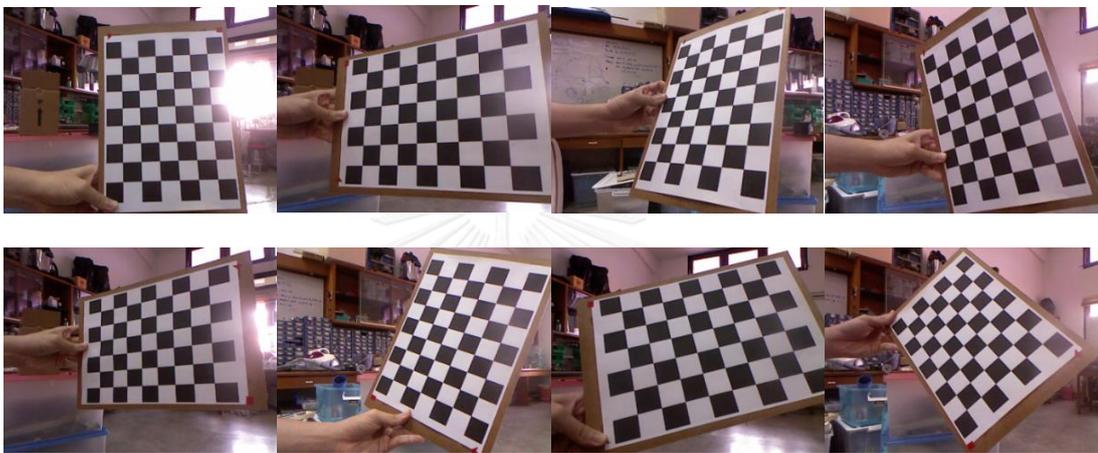
รอบคำสั่ง	1	2	3	4	5	6	7	8
ระยะทาง(cm)	11.7	10.3	15.2	10.6	10.7	12.2	11.25	15.4
มุมเลี้ยว (องศา)	-4.03	-0.32	-0.97	-8.74	-68.10	4.34	9.38	10.64
รอบคำสั่ง	9	10	11	12	13	14	15	16
ระยะทาง (cm)	13.1	15.7	18.2	19.0	3.7	10.2	10.2	10.2
มุมเลี้ยว (องศา)	11.3	11.3	13.0	22.15	53.74	0	0	0

ตารางที่ 2 ค่าตัวแปรที่ใช้สั่งการหุ่นยนต์

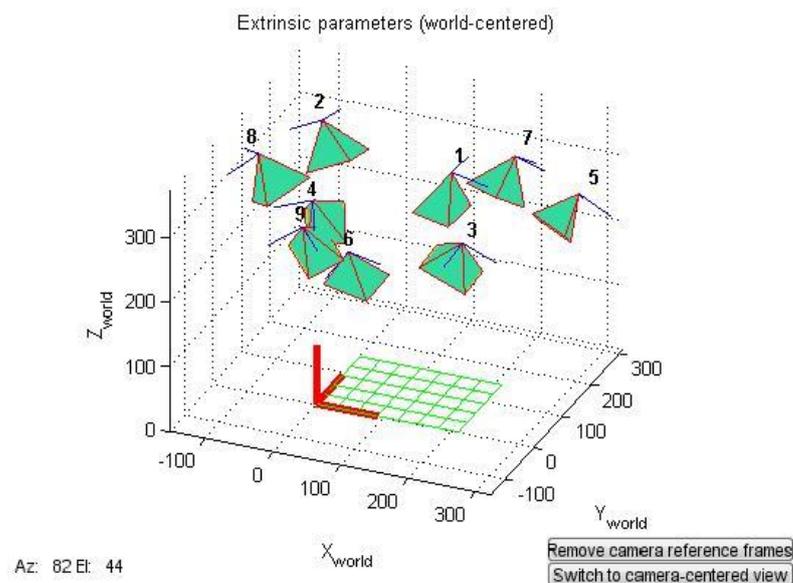
จะนำค่าเหล่านี้ไปแทนค่าตัวแปรในโปรแกรมคำสั่ง เหมือนในหัวข้อที่ 5.1 เพื่อรอการสั่งเริ่มทำงาน

## 5.6 การเทียบเส้นทางเดินกลับไปยังปริภูมิภาพ

เพื่อให้ตำแหน่งจุดเส้นทางเดินของหุ่นยนต์กลับไปปรากฏในภาพ RGB นั่นคือการเทียบค่าจากปริภูมิ  $(x,y,z)$  ให้ไปยังปริภูมิ  $(u,v)$  จำเป็นต้องทราบ ค่าตัวแปรภายในของกล้องเสียก่อน แต่เนื่องจากภาพที่ได้จากกล้องวิดีโอ นั้นจะเกิดการบิดเบี้ยวเสมอ จำเป็นต้องทำการสอบเทียบเพื่อให้ได้ข้อมูลภาพที่ถูกต้องก่อน โดยใช้วิธีของ Zhang ถ่ายภาพตารางหมากรุกจากหลายๆ มุม ดังรูปที่ 40 และรูปที่ 41 แสดงภาพจำลองที่ได้จากการถ่ายตารางหมากรุก



รูปที่ 40 ตัวอย่างภาพที่ถ่ายตารางหมากรุกจากหลาย ๆ มุม



รูปที่ 41 ภาพจำลองการถ่านภาพตารางหมากรุกจากหลาย ๆ มุม

จากการสอบเทียบกล้องได้ผลลัพธ์ดังนี้ ตัวแปรของกล้อง

Focal Length:  $f_c = [ 526.17422 \ 527.21573 ] \pm [ 6.37273 \ 6.26373 ]$

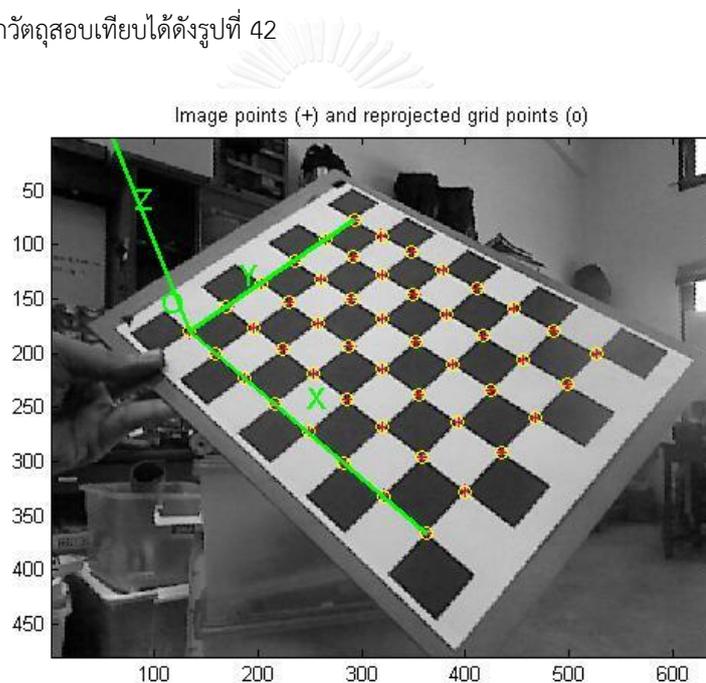
Principal point:  $cc = [ 333.43873 \ 261.29703 ] \pm [ 4.67185 \ 4.62454 ]$

Skew:  $\alpha_c = [ 0.00000 ] \pm [ 0.00000 ] \Rightarrow$  angle of pixel axes =  $90.00000 \pm 0.00000$  degrees

Distortion:  $k_c = [ 0.21848 \ -0.50260 \ -0.00086 \ 0.00139 \ 0.00000 ]$   
 $\pm [ 0.02580 \ 0.10493 \ 0.00356 \ 0.00396 \ 0.00000 ]$

Pixel error:  $err = [ 0.35062 \ 0.26880 ]$

ตัวแปรของภายนอกวัตถุสอบเทียบได้ดังรูปที่ 42



รูปที่ 42 แสดงผลลัพธ์ที่สามารถหาความสัมพันธ์ของตำแหน่งวัตถุในภาพได้

Translation vector:  $T_{c\_ext} = [ -136.960616 \ -56.131585 \ 367.061956 ]$

Rotation vector:  $omc\_ext = [ 2.362620 \ 0.916725 \ -0.404333 ]$

Rotation matrix:  $R_{c\_ext} = [ 0.719680 \ 0.690521 \ -0.07239$   
 $50.519073 \ -0.604359 \ -0.604412$   
 $-0.461112 \ 0.397404 \ -0.793376 ]$

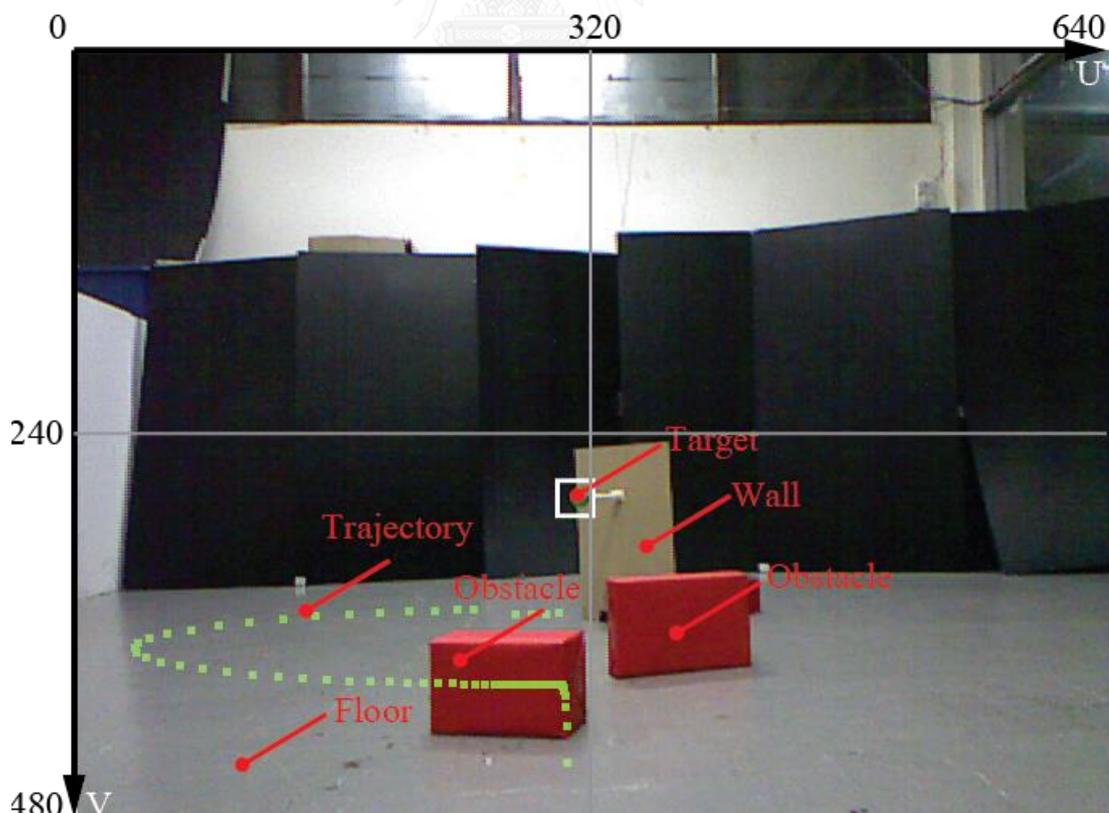
Pixel error:  $err = [ 0.21788 \ 0.19256 ]$

เมื่อได้ค่าความยาวโฟกัสแล้ว จะทำให้สมการสมบูรณ์เพียงต่อการเทียบจุดตำแหน่งของหุ่นยนต์ในปริภูมิคาร์ทีเซียน (x,y,z) ไปยังปริภูมิภาพ (u,v) จากนั้นกำหนดให้จุดบนพิกัดภาพเป็นทิวเขี้ยวเพื่อแสดงเส้นทางของหุ่นยนต์บนปริภูมิภาพ ดังรูปที่ 43

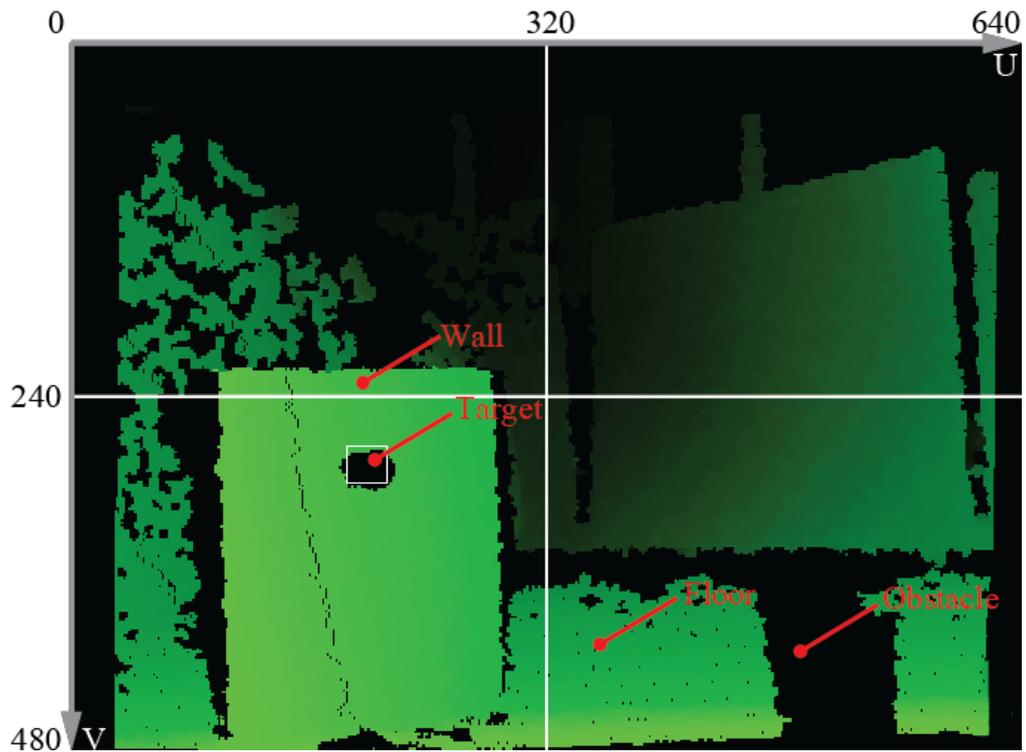
$$u = \frac{fX}{Z} \quad \text{และ} \quad v = \frac{fY}{Z}$$

### 5.7 การใช้กล้อง RGB เพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์ให้เคลื่อนที่เข้าหาเป้าหมาย

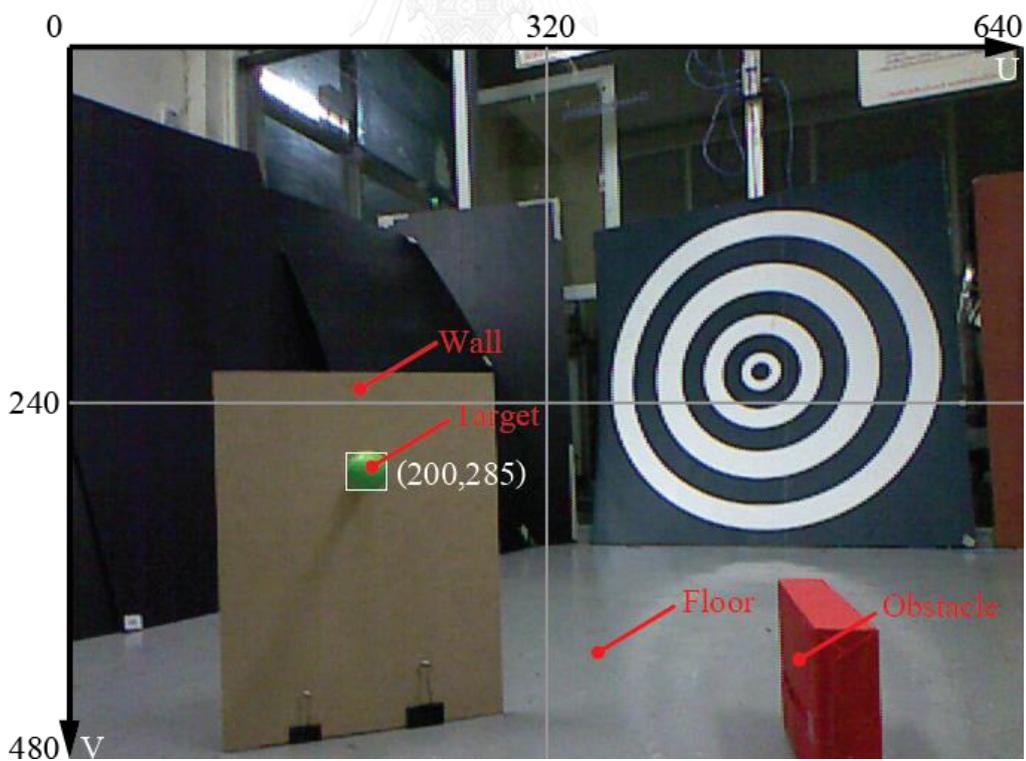
เมื่อสั่งให้หุ่นยนต์เคลื่อนที่ตามเส้นทางที่ถูกสร้างขึ้นมาจากขั้นตอนที่ผ่านมา เมื่อสิ้นสุดหุ่นยนต์มักจะไม่หยุดตรงที่เป้าหมายเสมอไป เนื่องจากความคาดเคลื่อนต่างๆ ที่ถูกสะสมมาจากขั้นตอนก่อนหน้านี้ แต่อย่างไรก็ตามหุ่นยนต์จะไปหยุดอยู่บริเวณใกล้ๆ กับเป้าหมาย แต่ปัญหาในขณะนี้คือ กล้องความลึกไม่สามารถตรวจจับวัตถุเป้าหมายได้เนื่องจากอยู่ไกลเกินไป ดังรูปที่ 44 ข แสดงให้เห็นว่ากล้องสี RGB ยังทำงานหาตำแหน่งเป้าหมายได้ปกติ ส่วนผลลัพธ์จากกล้องความลึกเป็นตามรูปที่ 44 ก เห็นได้ว่ากล้องความลึกไม่สามารถตรวจจับตำแหน่งของวัตถุเป้าหมายได้แล้ว



รูปที่ 43 เส้นทางเดินของหุ่นยนต์ในปริภูมิภาพ



ก. ภาพความลึก จากกล้องความลึก



ข. ภาพสี จากกล้องวีดีโอ

รูปที่ 44 ภาพที่ได้จากการอุปกรณ์ตรวจรู้ Kinect หลังสั่งหุ่นยนต์เคลื่อนที่

### 5.8 การหาความสัมพันธ์ระหว่างตำแหน่งของวัตถุเป้าหมายและภาพที่ได้จากกล้องวิดีโอ

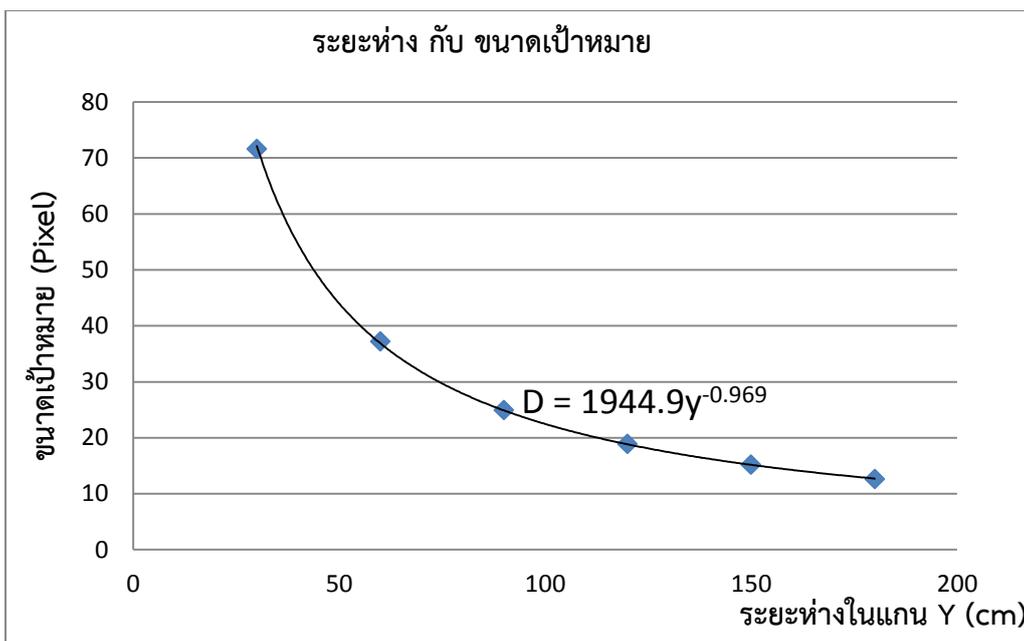
อย่างที่ทราบมาแล้วขณะนี้จำเป็นต้องใช้กล้องวิดีโอเพื่อใช้ในการควบคุมหุ่นยนต์เพียงอย่างเดียว และตัวแปรที่จำเป็นต่อการสั่งให้หุ่นยนต์เคลื่อนที่ทั้งหมด 2 ตัวแปรนั่นก็คือ ตัวแปรเก็บค่ามุมที่หุ่นยนต์ต้องหมุน และ ตัวแปรเก็บค่าระยะทางที่หุ่นยนต์ต้องเคลื่อนที่ จึงต้องทำการทดลองเก็บข้อมูลโดยใช้วิธีเดียวกับหัวข้อที่ 5.3 ตามรูปที่ 27 จะได้ข้อมูลที่น่าสนใจ พอว่าสามารถหาสมการความสัมพันธ์ของความยาวเส้นผ่านศูนย์กลางของวัตถุเป้าหมายที่ได้จากกล้องวิดีโอ RGB ที่มีผลต่อตำแหน่งระยะห่างวัตถุเป้าหมาย ได้ตามตารางที่ 3

	-90	-60	-30	0	30	60	90
180	12.7	12.8	12.6	12.6	12.6	12.5	12.5
150	unseen	15.5	15.4	15.2	15.3	15.4	unseen
120	unseen	19.4	19.2	18.9	19.1	19.6	unseen
90	unseen	unseen	25.3	24.9	25.5	unseen	unseen
60	unseen	unseen	39.6	37.2	39.8	unseen	Unseen
30	unseen	unseen	unseen	71.6	unseen	unseen	Unseen

ตารางที่ 3 ความยาวเส้นผ่านศูนย์กลางของวัตถุเป้าหมายที่ได้จากกล้องวิดีโอ RGB ที่ตำแหน่งต่าง

จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

จากข้อมูลใน ตารางที่ 3 จะสามารถหาความสัมพันธ์เชิงเส้นระหว่างระยะห่างของเป้าหมายและความยาวเส้นผ่านศูนย์กลางได้เป็นฟังก์ชันไฮเปอร์โบลิก ได้ดังรูปที่ 45



รูปที่ 45 พังค์ชันความสัมพันธ์ระหว่างระยะห่างของเป้าหมายกับขนาดเส้นผ่านศูนย์กลางของเป้าหมายในภาพจากกล้อง RGB

จากรูปที่ 45 ให้แกน  $y$  คือระยะห่างของเป้าหมาย และแกน  $D$  คือ ขนาดเส้นผ่านศูนย์กลางของเป้าหมาย พบความสัมพันธ์นั้นเป็นไปตามสมการ

$$D = 1944.9y^{-0.969}$$

ซึ่งหมายความว่า สามารถหาตำแหน่งของวัตถุเป้าหมายได้จากขนาดเส้นผ่านศูนย์กลางของเป้าหมายได้ตามสมการนี้

$$y = (1.944.9D)^{-1.032}$$

นอกจากนั้น ยังมีข้อมูลที่น่าสนใจเพื่อใช้ในการหามุมที่หุ่นยนต์ต้องหมุน นั่นคือข้อมูลของตำแหน่งจุดศูนย์กลางวงกลมที่ปรากฏในปริภูมิภาพ เนื่องจากเป้าหมายมีความสูงคงที่เพราะฉะนั้นจึงเลือกเก็บค่าเฉพาะพิกัดในแนวนอนของภาพ หรือ 'U' และเป็นที่ทราบว่าคุณภาพมีความละเอียดในแนว U จาก 1 ถึง 640 พิกเซล เพราะฉะนั้นจึงต้องเลื่อนแกนให้ (0,0) มาอยู่ตรงกลางภาพ แล้วเก็บค่าตำแหน่งจุดศูนย์กลางของเป้าหมายเท่ากับจุดกึ่งกลางภาพ และทำการคำนวณหามุมจากตำแหน่งจริงของเป้าหมายจากคุณสมบัติตรีโกณมิติ ได้ค่าตามตารางที่ 4

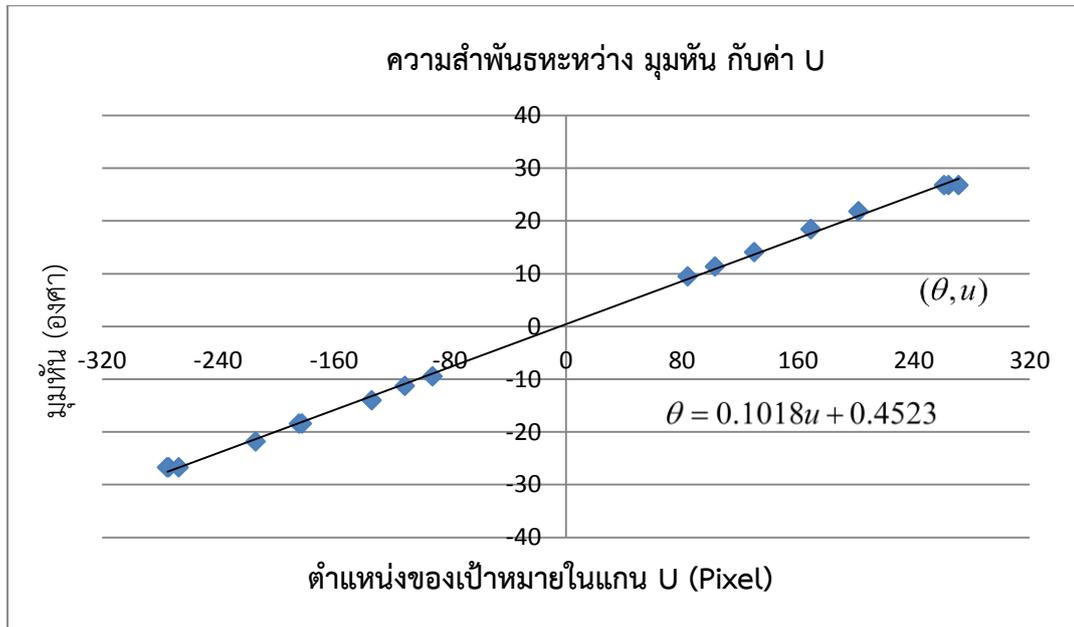
		-90	-60	-30	0	30	60	90
180	มุ่ม	-26.75	-18.43	-9.46	0	9.46	18.43	26.57
	U	-274	-184	-92	4	84	169	264
150	มุ่ม	unseen	-21.80	-11.31	0	11.31	21.80	unseen
	U	unseen	-214	-111	3	103	202	unseen
120	มุ่ม	unseen	-26.75	-14.04	0	14.04	26.75	unseen
	U	unseen	-275	-134	2	130	261	unseen
90	มุ่ม	unseen	unseen	-18.43	0	18.43	unseen	unseen
	U	unseen	unseen	-182	1	169	unseen	unseen
60	มุ่ม	unseen	unseen	-26.57	0	26.57	unseen	unseen
	U	unseen	unseen	-267	-	271	unseen	unseen
30	มุ่ม	unseen	unseen	unseen	0	unseen	unseen	unseen
	U	unseen	unseen	unseen	-	unseen	unseen	unseen

ตารางที่ 4 ตำแหน่งจุดศูนย์กลางของเป้าหมายที่ปรากฏในภาพจากกล้องวิดีโอ RGB ณ ตำแหน่งต่างๆ

จากข้อมูลที่ได้สามารถเขียนนำค่าตำแหน่ง U และมุ่มหันมาบันทึกลงในตารางที่ 5 ได้ดังนี้

มุ่ม (องศา)	-	-	-26.75	-21.80	-18.43	-18.43	-14.04	-11.31	-9.46
U(พิก เซล)	-274	-275	-267	-214	-184	-182	-134	-111	-92
มุ่ม (องศา)	9.46	11.31	14.04	18.43	18.43	21.80	26.75	26.75	26.75
U(พิก เซล)	84	103	130	169	169	202	271	261	264

ตารางที่ 5 ค่ามุ่มหันของวัตถุเมื่อเทียบกับเส้นแกนमुखสำคัญของกล้อง ตำแหน่งจุดศูนย์กลางของ



รูปที่ 46 ความสัมพันธ์ของตำแหน่งเป้าหมายในปริภูมิภาพกับมุมหัน

จากรูปที่ 46 สามารถหาสมการความสัมพันธ์ระหว่างตำแหน่งของเป้าหมายในปริภูมิภาพ ทำให้ได้สมการดังนี้

$$\theta = 0.1018u + 0.4523$$

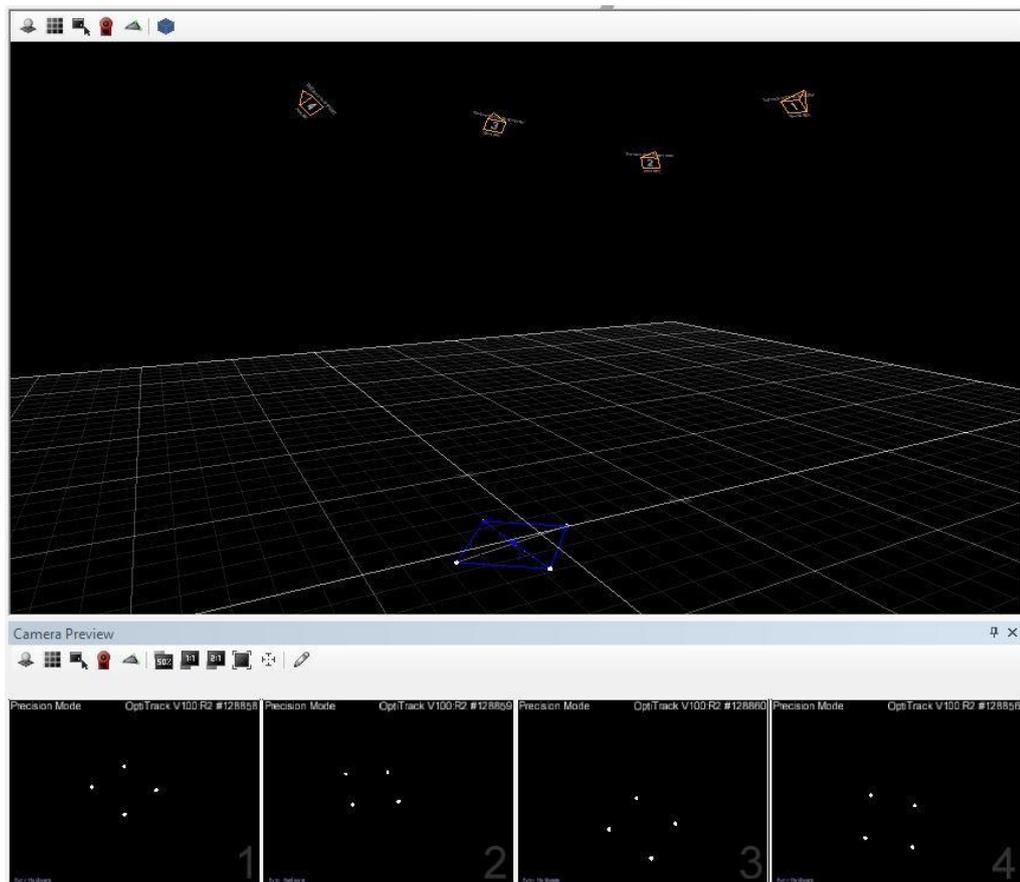
จากผลการทดลองข้างต้นทำให้สามารถหาตัวแปรจากข้อมูลที่ได้จากกล้องวิดีโอสี RGB ที่จะใช้ในการสั่งหุ่นยนต์ให้เครื่องที่ไปยังเป้าหมายได้ครบ

### 5.9 ระบบกล้องตรวจจับตำแหน่งสเตอริโออ็อปติแตรีก (Opti track)

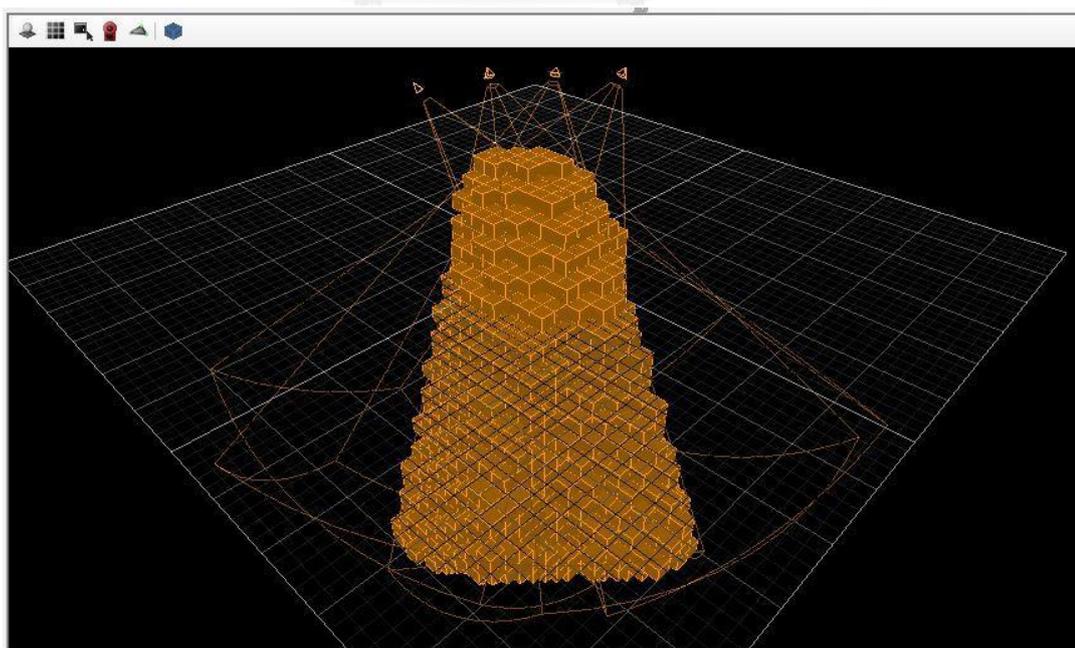
ในงานวิจัยนี้ได้ใช้ระบบกล้องสเตอริโออ็อปติแตรีกเป็นเครื่องมือสำหรับตรวจจับการเคลื่อนไหวของหุ่นยนต์ (Motion capture) เนื่องจากสามารถตรวจจับการเคลื่อนไหวได้ทั้งตำแหน่งและการหมุน ที่มีประสิทธิภาพทางด้านความละเอียดและอัตราเร็วภาพสูงถึง 100 Hz ซึ่งบริษัท NaturalPoint Inc เป็นผู้พัฒนาระบบกล้องสเตอริโออ็อปติแตรีก

ขั้นตอนแรกของการติดตั้งระบบกล้องสเตอริโออ็อปติแตรีกคือการกำหนดตำแหน่งศูนย์กลางของพื้นที่ที่ต้องการทำงาน โดยให้จุดศูนย์กลางพื้นที่นี้จะอยู่ที่กึ่งกลางภาพของกล้องแต่ละตัว ขั้นตอนต่อมาคือ การสอบเทียบระบบกล้องสเตอริโอ โดยระบบสเตอริโออ็อปติแตรีกใช้กับโปรแกรม OptiTrack Tool ได้ใช้วิธีการสอบเทียบของ T. Svoboda, A convenient Multi-Camera Self-Calibration for virtual environments จากนั้นขั้นตอนต่อไป จากโปรแกรม OptiTrack tool จะได้ตำแหน่งของกล้องแต่ละตัวในแบบจะลองตามแนวแกนอ้างอิงของระบบกล้องเป็นภาพ 3 มิติ ดังแสดงตามรูปที่ 47 โดยกรอบหน้าต่างด้านบนแสดงภาพรวมของตำแหน่งและมุมการหมุนของกล้องทุกตัวเทียบกับวัตถุที่มองเห็นในแบบจำลองตามพิกัด 3 มิติ ส่วนหน้าต่างข้างล่างทั้งภาพ 4 ภาพแสดงการมองเห็นของกล้องแต่ละตัว

ข้อควรระวังหลังจากที่เสร็จสิ้นกระบวนการสอบเทียบของระบบกล้องสเตอริโออ็อปติแตรีก ห้ามเคลื่อนย้ายหรือขยับกล้องเลยทั้งสิ้น ถ้ามีการขยับหรือเคลื่อนจากตำแหน่งแม้แต่เพียงนิดเดียว จะต้องทำการสอบเทียบใหม่ มิเช่นนั้นผลที่ได้จะมีความคลาดเคลื่อน นอกจากนี้ระบบกล้องสเตอริโออ็อปติแตรีกยังสามารถตรวจสอบขนาดพื้นที่ที่สามารถตรวจจับวัตถุในได้ ดังรูปที่ 48 ซึ่งกล้องสี่เหลี่ยมสี่มุมสี่ถึงบริเวณพื้นที่การทำงาน เพื่อความถูกต้องแม่นยำจะต้องมีกล้องอย่างน้อย 3 ตัวที่มองเห็นวัตถุเป้าหมายจากจำนวนกล้องทั้งหมด 4 ตัว



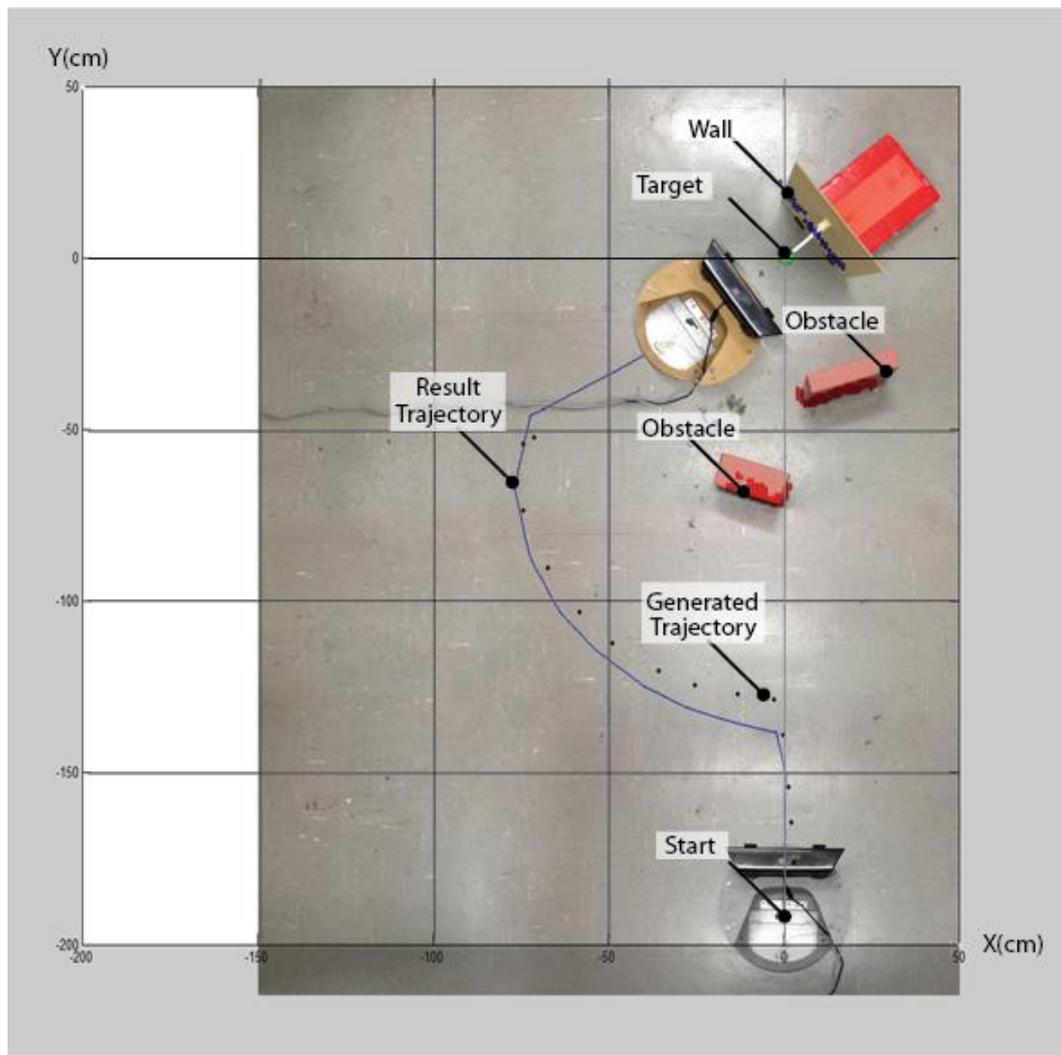
รูปที่ 47 ภาพจากโปรแกรมของ OptiTrack Tool ของกล้องแต่ละตัว



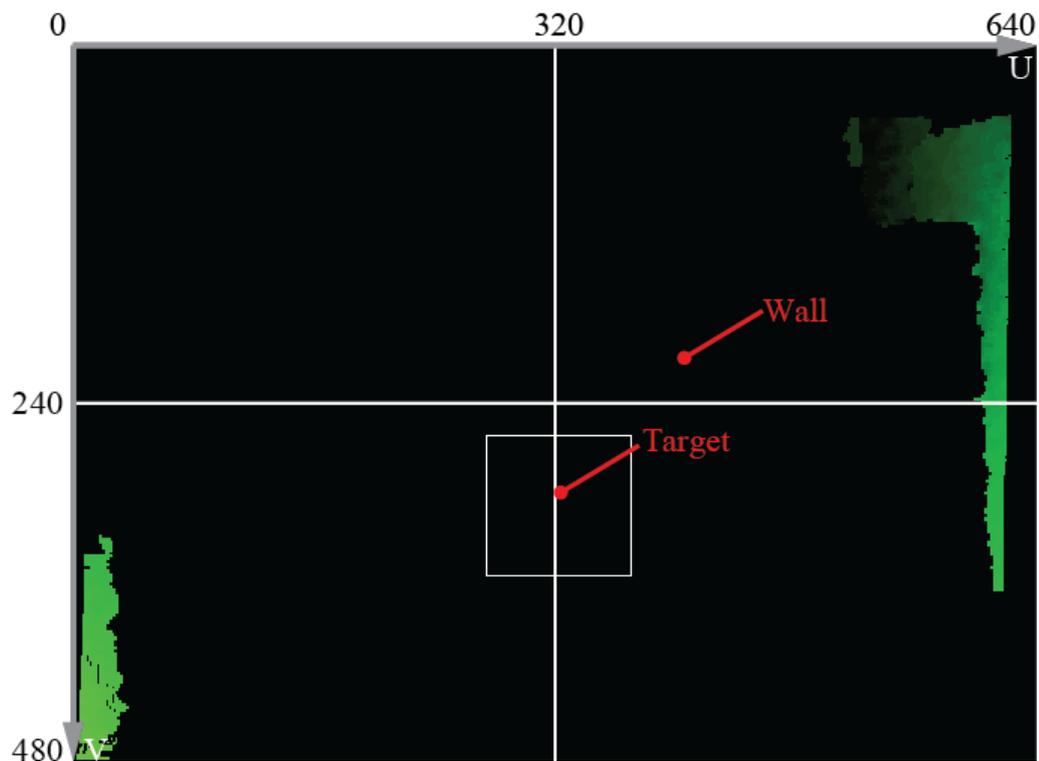
รูปที่ 48 บริเวณพื้นที่ทำงานที่สามารถตรวจจับได้ด้วยระบบกล้อง OptiTrack

### 5.10 ผลลัพธ์ที่ได้จากกล้องสเตอริโอออฟติคัล

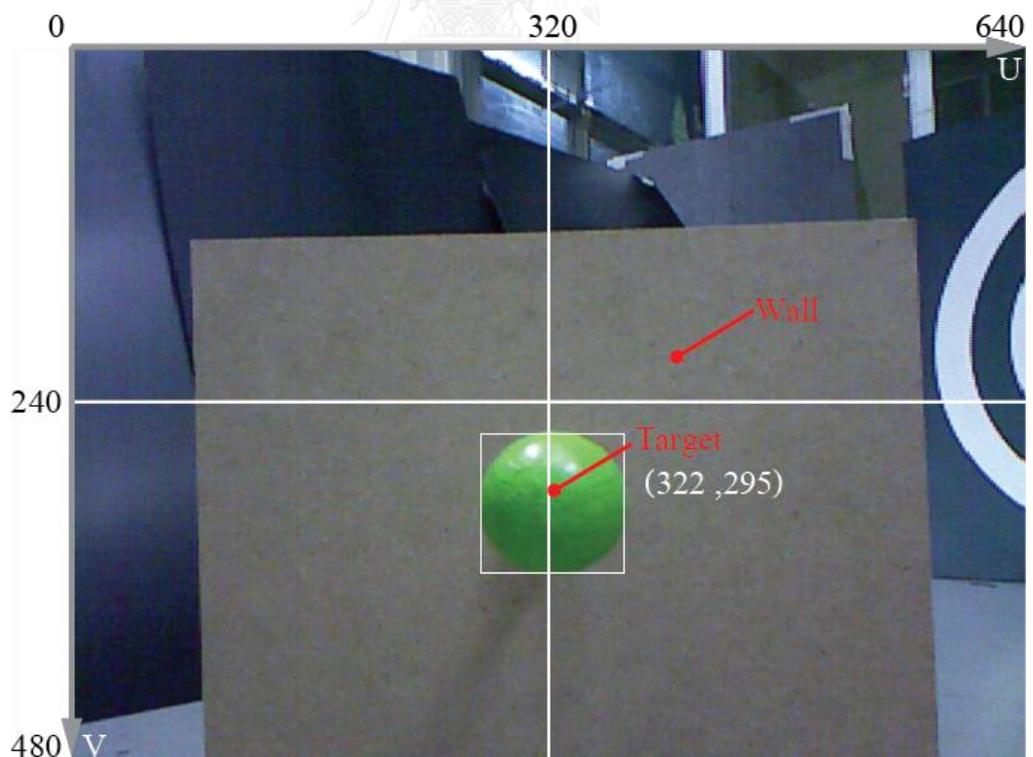
ดำเนินการสั่งให้หุ่นยนต์เคลื่อนที่ตามเส้นทางที่ถูกสร้างขึ้นด้วยการประมวลจากอุปกรณ์ตรวจรู้ RGB-D จนกระทั่งอยู่ในระยะที่กล้องความลึกไม่สามารถตรวจจับเป้าหมายได้แล้ว จากนั้นจึงใช้กล้องวิดีโอเพื่อสั่งให้หุ่นยนต์เคลื่อนที่เข้าหาเป้าหมาย พร้อมกับเก็บค่าตำแหน่งขณะที่หุ่นยนต์เคลื่อนที่ จะได้ผลลัพธ์ดังรูปที่ 49 จะเห็นได้ว่าการเคลื่อนที่ของหุ่นยนต์ (เส้นสีน้ำเงิน) สอดคล้องกับเส้นทางเดินที่สร้างขึ้นมา นั่นคือต้องหลบหลีกสิ่งกีดขวางและไปจอดหันหน้าตั้งฉากกับวัตถุเป้าหมายตามรูปที่ 50 เห็นได้ว่าวัตถุเป้าหมายนั้นอยู่ตรงหน้าพอดี



รูปที่ 49 การบันทึกตำแหน่งด้วยกล้องออฟติคัล



ก. ภาพความลึก จากกล้องความลึก



ข. ภาพสี จากกล้องวิดีโอ

รูปที่ 50 ภาพจากการอุปกรณ์ตรวจรู้ Kinect หลังสั่งหุ่นยนต์เคลื่อนที่

## บทที่ 6

### สรุปผลการวิจัย

#### 6.1 สรุปผลการวิจัย

วิทยานิพนธ์บรรลุวัตถุประสงค์ในการพัฒนาการสร้างเส้นทางเดินในปริภูมิรูปภาพสำหรับหุ่นยนต์เคลื่อนที่ด้วยล้อ โดยการประสานข้อมูลจากกล้อง RGB และกล้องความลึกเพื่อช่วยในการสร้างเส้นทางหุ่นยนต์เคลื่อนที่ด้วยล้อแบบอัตโนมัติให้ไปจอด ณ ตำแหน่งเป้าหมายได้จริง กล่าวคือ หุ่นยนต์สามารถเคลื่อนที่ตามเส้นทางที่ระบบสร้างขึ้นมานั้นโดยจะต้องหลบหลีกสิ่งกีดขวางแล้วไปหยุดที่ตำแหน่งหน้าตั้งฉากกับกำแพงของเป้าหมาย โดยสภาพแวดล้อมทดลองตำแหน่งของสิ่งกีดขวางและเป้าหมายเป็นแบบที่ไม่เจาะจงและนิ่งเฉย

จากการทดลองพบว่าวิทยานิพนธ์นี้ประสบความสำเร็จตามวัตถุประสงค์ของวิทยานิพนธ์

1) ด้านการจัดการภาพของกล้องวีดีโอสีของอุปกรณ์ตรวจตรวจรู้ Kinect มีความเชี่ยวชาญของเลนส์อยู่บ้างแต่ก็สามารถชดเชยได้ โดยการสอบเทียบกล้องทำให้ผลลัพธ์ดีขึ้น จนสามารถใช้ทำการคัดแยกละเอียดของภาพ จากการเปลี่ยนปริภูมิสี RGB เป็นปริภูมิสี HSV ทำให้สามารถตรวจจับตำแหน่งของได้ง่ายขึ้นจนได้ตำแหน่งของเป้าหมายในปริภูมิรูปภาพได้ ส่วนกล้องความลึกตรวจจับความลึกนั้น จากการทดสอบพบที่มีความคาดเคลื่อนเล็กน้อยอยู่ในช่วงไม่เกิน 5 เซนติเมตร ความคาดเคลื่อนแปรผันตรงกับระยะห่างของตำแหน่งวัตถุที่กำลังสนใจ แต่มีข้อแม้ยู่ที่ว่าวัตถุที่ต้องการตรวจจับนั้นต้องอยู่ไกลออกไปจากตัวกล้องอย่างต่ำ 60 เซนติเมตร มิฉะนั้นกล้องจะมองไม่เห็น

2) สามารถแยกแยะและค้นหาตำแหน่งของสิ่งแวดล้อมในปริภูมิคาร์ทีเซียนได้สำเร็จ ด้วยการประสานข้อมูลดิบที่ได้จากกล้องตรวจจับความลึกและกล้องสี RGB แยกแยะสิ่งกีดขวางออกจากพื้น โดยใช้ความระดับสูงของจุดอินฟราเรดช่วยในการคัดแยก แล้วใช้ค่าตำแหน่งของเป้าหมายที่ได้จากการจัดการภาพจากกล้องสี RGB เทียบไปยังตำแหน่งในปริภูมิคาร์ทีเซียนของภาพจากกล้องตรวจจับความลึกทำให้ได้ตำแหน่งในปริภูมิคาร์ทีเซียนของเป้าหมาย สุดท้ายกำหนดขอบเขตของจุดความลึกรอบๆเป้าหมายในการแยกแยะตำแหน่งของกำแพง เนื่องจากหุ่นยนต์เคลื่อนที่ด้วยล้อสามารถเคลื่อนที่ได้เพียง 2 มิติ ทำให้สามารถละลายค่าในแกนความสูงจากพื้นของปริภูมิคาร์ทีเซียนได้เลย จึง

ใช้เฉพาะค่าในแกน  $x$  และ  $y$  มาพอร์ตในแผนที่ 2 มิติ แต่ว่าข้อมูล ณ จุดนี้ยังไม่สามารถใช้ประโยชน์ ต้องทำการลดปริมาณข้อมูลที่ซ้ำกันออกไปด้วยวิธีการแทนที่จุดในตาราง (Grid Occupation) ถึงแม้ว่าวิธีนี้ลดปริมาณข้อมูลได้อย่างมากแต่ก็ต้องแลกมาด้วยการสูญเสียความแม่นยำของข้อมูลไป เพราะฉะนั้นจึงต้องพึงระวังถึงเรื่องนี้ไปเสมอ สุดท้ายแล้วจากกระบวนการข้างตอนที่กล่าวมา สามารถจัดการสัญญาณจากอุปกรณ์ตรวจจับ Kinect จนได้แผนที่ แสดงตำแหน่งของ เป้าหมาย ฉากกำแพง และสิ่งกีดขวาง

3) สามารถสร้างเส้นทางเดินของหุ่นยนต์ให้แสดงในปริภูมิรูปภาพได้สำเร็จโดยใช้เทคนิคการสร้างสนามศักย์ จากแผนที่ที่ได้จากการจัดการภาพจากอุปกรณ์ตรวจจับ Kinect กำหนดให้จุดที่เป็นเป้าหมายจะกำเนิดสนามศักย์ดึงดูดหุ่นยนต์ ส่วนจุดที่เป็นสิ่งกีดขวางจะกำเนิดสนามศักย์ผลักดันหุ่นยนต์ นอกจากนี้ยังใช้ข้อมูลภาพถ่ายช่วยในการหาเส้นแนวตั้งฉากกับกำแพง ทำให้สามารถสร้างสนามศักย์ดึงดูดแนวตั้งฉาก เส้นทางการเคลื่อนในส่วนท้ายจึงตั้งฉากกับกำแพงของเป้าหมายได้ จากนั้นจำลองการเคลื่อนที่เปรียบเทียบให้หุ่นยนต์เป็นประจุไฟฟ้าปล่อยให้เคลื่อนที่ในสนามไฟฟ้าตามธรรมชาติและบันทึกตำแหน่งที่ได้จากการจำลองทุกๆระยะ 10 เซนติเมตร ในขั้นตอนนี้จะเห็นได้ว่าลักษณะของสนามศักย์นั้นมีอิทธิพลโดยตรงต่อความประพติของหุ่นยนต์ เพราะฉะนั้นจึงมีความเป็นไปได้ที่จะเปลี่ยนลักษณะ ความแรง หรือแม้กระทั่งประเภทของสนามศักย์ตามความเหมาะสมกับวัตถุประสงค์ จากนั้นนำตำแหน่งที่หุ่นยนต์ต้องเคลื่อนที่เทียบกลับไปยังปริภูมิภาพ เพื่อแสดงเส้นทางการเคลื่อนที่ในภาพ จากกล่าวมาข้างต้นความสูงของจุดนั้นไม่มีนัยสำคัญต่อการเคลื่อนที่ของหุ่นยนต์ การใช้กล้องวีดีโอสีควบคุมการเคลื่อนที่ของหุ่นยนต์เมื่อหุ่นยนต์เคลื่อนที่เข้าใกล้วัตถุเป้าหมายจนกล้องตรวจจับนั้นไม่สามารถตรวจจับตำแหน่งเป้าหมายได้แล้ว พบว่าจากการความสัมพันธ์ของตำแหน่งเป้าหมายกับมุมหันของหุ่นยนต์ทำให้หุ่นยนต์สามารถเคลื่อนที่เข้าหาเป้าหมายได้โดยรักษาตำแหน่งให้วัตถุเป้าหมายอยู่ตรงกลางภาพก่อนออกคำสั่งเคลื่อนที่

4) สามารถแปลงค่าตำแหน่งที่ได้จากกระบวนการแปลงข้อมูลจากอุปกรณ์ตรวจจับ Kinect สร้างเส้นทางเดินให้เป็นชุดตัวแปรคำสั่งให้เหมาะสมสำหรับสั่งให้หุ่นยนต์เคลื่อนที่ด้วยล้อ irobot create ซึ่งมีลักษณะการขับเคลื่อนแบบ Differential Drive จากนั้นใช้ระบบกล้อง optitrack ในการเก็บเส้นทางการเคลื่อนที่ของหุ่นยนต์

## 6.2 ข้อเสนอแนะเพื่องานวิจัยในอนาคต

ปัญหาที่พบในงานวิจัย คือ การเชื่อมต่ออุปกรณ์ต่างๆของระบบใช้เส้นทางสายเคเบิลเสียบปลั๊กจึงมีโอกาสมากที่บริเวณเชื่อมต่อสายกับอุปกรณ์ต่างๆจะหลวมทำให้ระบบทั้งหมดล้มเหลวจนต้องเริ่มทำการทดลองใหม่ตั้งแต่ต้น จึงยังไม่เหมาะสมสำหรับการนำไปใช้งานจริง ควรปรับปรุงเรื่องการเชื่อมต่ออุปกรณ์จากการเสียบปลั๊กเป็นการเชื่อมสายไฟแทนเพราะจะเป็นการตัดปัญหาในส่วนนี้ไปเลย

สิ่งแวดล้อมทั้งหมดในการวิจัยนี้เป็นแบบคงที่ แต่ในสภาพความเป็นจริงนั้นสิ่งแวดล้อมมีทั้งเป็นแบบคงที่และแบบพลวัต เพื่อปรับปรุงให้ระบบเหมาะสมสำหรับการทำงานใช้ในสภาพแวดล้อมจริง เช่นบนท้องถนน หรือออฟฟิตในช่วงเวลากลางวัน เพื่อหลีกเลี่ยงอันตรายจากอุบัติเหตุในสภาพแวดล้อม จึงควรพัฒนาระบบข้อหุนยนต์ให้สามารถหลีกเลี่ยงการปะทะกับวัตถุที่กำลังเคลื่อนที่อยู่ และยังต้องปรับปรุงเรื่องความเร็วของระบบรวมเพราะ ผลจากงานวิจัยยังพบว่าความเร็วของอุปกรณ์ตรวจรู้ Kinect อาจจะไม่เร็วพอสำหรับการตอบสนองต่อวัตถุที่มีความเร็ว อาจจะพัฒนาระบบการคำนวณหรืออาจจะต้องถึงขั้นมุ่งเน้นพัฒนาในตัวอุปกรณ์ตรวจรู้โดยตรงด้วยเช่นกัน

กล้องตรวจจับความลึกของอุปกรณ์ตรวจรู้ Kinect มีข้อด้อยที่ชัดเจนอยู่เรื่องหนึ่งนั่นคือ ไม่สามารถตรวจจับวัตถุที่อยู่ภายในช่วง 60 เซนติเมตรได้ ส่วนกล้องวีดีโอสี RGB นั้นมองเห็นสิ่งที่อยู่ใกล้ก็จริงแต่การทำงานเพียงลำพังของกล้องวีดีโอสีไม่สามารถแยกแยะและหาตำแหน่งของสิ่งแวดล้อมได้อย่างมีประสิทธิภาพ เมื่อนำมาใช้กับหุ่นยนต์เคลื่อนที่ด้วยล้อ ระบบจะเสียคุณสมบัติในการตรวจจับสิ่งแวดล้อมที่กำลังจะชนไปซึ่งประเด็นนี้นั้นมีความสำคัญมากต่อความปลอดภัยของการทำงานของหุ่นยนต์ เพื่อให้ระบบรวมของหุ่นยนต์มีประสิทธิภาพมากขึ้นจะลดจุดด้อยนี้ลง ซึ่งอาจจะทำได้โดยการปรับปรุงการคำนวณของกล้องสี RGB ให้มีประสิทธิภาพในการตรวจหาและแยกแยะ ชนิดและตำแหน่งของวัตถุในสิ่งแวดล้อมได้ หรือเพิ่มอุปกรณ์ตรวจรู้ชนิดอื่นในช่วงให้กับหุ่นยนต์

## รายการอ้างอิง

1. Luo, R.C., C.T. Liao, and K.C. Lin, *Vision-based docking for automatic security robot power recharging*. Advanced Robotics and its Social Impacts, 2005. IEEE Workshop, 2005 p. 214 - 219.
2. Tribelhorn, B. and Z. Dodds, *Evaluating the Roomba: A low-cost, ubiquitous platform for robotics research and education*. Robotics and Automation, 2007 IEEE International Conference, 2007: p. 1393 - 1399.
3. Bai, M., Y. Zhuang, and W. Wang, *Stereovision based obstacle detection approach for mobile robot navigation*. Intelligent Control and Information Processing (ICICIP), 2010 International Conference, 2010: p. 328 - 333.
4. Henry, P., et al., *RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments*. Proceedings - 2013 International Conference on 3D Vision, 3DV 2013, 2013: p. 398 - 405.
5. Song, G., et al., *Automatic docking system for recharging home surveillance robots*. Consumer Electronics IEEE Transactions on, 2011. 57(2): p. 428 - 435.
6. Peasley, B. and S. Birchfield, *Real-time obstacle detection and avoidance in the presence of specular surfaces using an active 3D sensor*. Robot Vision (WORV), 2013 IEEE Workshop 2013: p. 197 - 202.
7. Borenstein, J., et al., *Mobile Robot Positioning -- Sensors and Techniques*. Invited paper for the Journal of Robotic Systems, Special Issue on Mobile Robots., 1997. 14(4): p. 231 - 249.
8. M. Kumano, A., Ohya, and S. Yuta, *Obstacle avoidance of autonomous mobile robot using stereo vision sensor*. In Intl. Symp. Robot. Automat., 2000: p. 497 - 502.
9. Silverman, M.C., et al., *Staying alive: a docking station for autonomous robot recharging*. Proceedings of IEEE International Conference on Robotics and Automation(ICRA'02), 2002. 1: p. 1050 - 1055.

10. Luo, R.C., et al., *Automatic Docking and Recharging System for Autonomous Security Robot*. Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on, 2005: p. 2953 - 2958.
11. Yuta, S. and Y. Hada, *Long term activity of the autonomous robot – Proposal of a bench-mark problem for the autonomy*. Proceedings of the 1998 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, 1998: p. 1871 - 1878.
12. Arkin, R.C. and R.R. Murphy, *Autonomous navigation in a manufacturing environment*. Robotics and Automation, IEEE Transactions 1990. 6(4): p. 445 - 454.
13. Minten, B.W., et al., *Low-order-complexity vision-based docking*. Robotics and Automation, IEEE Transactions, 2001. 17(6): p. 922 - 930.
14. Ulrich, I. and J. Borenstein, *VFH+: reliable obstacle avoidance for fast mobile robots*. Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on, 1998. 2: p. 1572 - 1577.
15. B., O. and u. K., *High-speed navigation using the global dynamic window approach*. Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference., 1999. 1: p. 341 - 346.
16. Nourbakhsh, I., R. Powers, and S. Birchfield, *Dervish: An officenavigating robot*. AI Magazine, 1995. 16(2): p. 53 - 60.
17. Soumare, S., A. Ohya, and S. Yuta, *Real-time obstacle avoidance by an autonomous mobile robot using an active vision sensor and a vertically emitted laser slit*. Intelligent Autonomous Systems, 2002: p. 301 - 308.
18. Michels, J., A. Saxena, and A.Y. Ng, *High-speed obstacle avoidance using monocular vision and reinforcement learning*. Proceedings of the Twenty-second International Conference on Machine Learning (ICML), 2005: p. 593 - 6000.
19. Zhang, Z., *Microsoft Kinect Sensor and Its Effect*. IEEE Computer Society, IEEE MultiMedia 2012. 19(2): p. 4 - 12.

20. Khandelwal, P. and P. Stone, *A low cost ground truth detection system for RoboCup using the Kinect*. Proceedings of the RoboCup International Symposium 2011 (RoboCup 2011), 2011.
21. Cunha, J., et al., *Using a depth camera for indoor robot localization and navigation*. 2011.
22. Flacco, F., et al., *A depth space approach to human-robot collision avoidance*. Robotics and Automation (ICRA), 2012 IEEE International Conference, 2012: p. 338 - 345.
23. Remondino, F. and C. Fraser, *DIGITAL CAMERA CALIBRATION METHODS: CONSIDERATIONS AND COMPARISONS*. ISPRS Commission V Symposium 'Image Engineering and Vision Metrology', 2014.
24. Fraser, C.S., *Invited Review Paper Digital camera self-calibration*. SPRS Journal of Photogrammetry & Remote Sensing, 1997. 52(4): p. 149 - 159.
25. Zhang, Z., *Flexible camera calibration by viewing a plane from unknown orientations*. Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference, 1999. 1: p. 666 - 673.
26. Weijer, J. and C. Schmid, *Coloring Local Feature Extraction*. Computer Vision – ECCV 2006, 2006: p. 334 - 348.
27. Ali, N.M., et al., *Performance Comparison between RGB and HSV Color Segmentations for Road Signs Detection*. Applied Mechanics and Materials 2013. 393: p. 550 - 555.
28. Sezgi, M. and B. Sankur, *Survey over image thresholding techniques and quantitative performance evaluation*. J. Electron. Imaging., 2004. 13(1): p. 146 - 168.
29. Ming, A. and H. Ma, *A blob detector in color images*. The 6th ACM international conference on Image and video retrieval, 2007: p. 364 - 370.
30. H., S. and G.D.H.P. C., *A Tutorial on Visual Servo Control*. IEEE Transactions on robotics and automation, 1996. 12(5).
31. Mojtahedzadeh, R., *Robot Obstacle Avoidance using the Kinect*, in *Computer Science*. 2011, Royal Institute of Technology. p. 61.

32. Ruiz, E., et al., *Development of a Control Platform for the Mobile Robot Roomba Using ROS and a Kinect Sensor*. Robotics Symposium and Competition (LARS/LARC), 2013 Latin American, 2013 p. 55 - 60.
33. A.G., M., *Potential Fields Tutorial*.
34. Koren, Y. and J. Borenstein, *Potential field methods and their inherent limitations for mobile robot navigation*. 1991 IEEE International Conference, 1991. 2: p. 1398 - 1404.
35. Gao, Y., et al., *Obstacle detection with 3D camera using U-V-Disparity*. Systems, Signal Processing and their Applications (WOSSPA), 2011 7th International Workshop, 2011: p. 239 - 242.
36. Khatib, O., *Real-time obstacle avoidance for manipulators and mobile robots*. Robotics and Automation. Proceedings. 1985 IEEE International Conference, 1985. 2: p. 500 - 505.
37. Bradski, G. and A. Kaehler, *Learning OpenCV*. 2008.
38. Hartley, R.I., *An algorithm for self-calibration from several views*. Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference, 1994: p. 908 - 912.
39. Heikkila, J. and O. Silven, *A four-step camera calibration procedure with implicit image correction*. Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference, 1997: p. 1106 - 1112.
40. Luong, Q.-T. and O.D. Faugeras, *Self-calibration of a moving camera from point correspondences and fundamental matrices*. International Journal of Computer Vision, 1997. 22(3): p. 261.
41. Bougnoux, S., *From projective to euclidean space under any practical situation, a criticism of self-calibration*. Proc. 6th International Conference on Computer Vision, 1998: p. 790 - 796.
42. Sturm, P.F. and S.J. Maybank, *On plane-based camera calibration: A general algorithm, singularities, applications*. Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference 1999. 1: p. 437.
43. Clark, P. and M. Mirmehdi, *Estimating the Orientation and Recovery of Text Planes in a Single Image*. 2001.

44. P., W., R. C., and V. S., *Automatic Docking with Obstacle Avoidance of a Differential Wheel Mobile Robot*. International Journal of Mechanical Engineering and Robotics Research, 2015. 5(1): p. 11 - 16.
45. *iRobot Create Open Interface (OI) Specification*. 2006; Available from: [http://www.irobot.com/filelibrary/pdfs/hrd/create/Create%20Open%20Interface\\_v2.pdf](http://www.irobot.com/filelibrary/pdfs/hrd/create/Create%20Open%20Interface_v2.pdf).
46. Boucher, S., *Obstacle Detection and Avoidance Using TurtleBot Platform and Xbox Kinect*, in *Computer Science*. 2012, Rochester Institute of Technology.





ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

## ประวัติผู้เขียนวิทยานิพนธ์

ชื่อ – ชื่อสกุล: นายวาริน ภูมรินทร์

วัน เดือน ปีเกิด: 14 มิถุนายน 2529

สถานที่เกิด: จังหวัดพิษณุโลก

สถานที่อยู่ปัจจุบัน: 27/19 ถนนธรรมบูชา ตำบลในเมือง อำเภอเมือง จังหวัดพิษณุโลก  
รหัสไปรษณีย์ 65000

ประวัติการศึกษา

พ.ศ. 2552: หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาฟิสิกส์ มหาวิทยาลัยเชียงใหม่  
จังหวัดเชียงใหม่

