การสร้างแผนที่เสมือนแบบเวลาจริงสำหรับนำทางยานพาหนะภาคพื้นดินอัตโนมัติโดยใช้ตัวตรวจรู้หลายชนิดร่วมกับเทคนิคปัญญาประดิษฐ์

นายศิริพงษ์ ปะวะโก

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมเมคคาทรอนิกส์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2568

REAL-TIME VIRTUAL MAPPING FOR GUIDED AGV USING MULTIPLE SENSORS WITH ARTIFICIAL INTELLIGENCE TECHNIQUES

SIRIPONG PAWAKO

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy in Mechatronics Engineering
Suranaree University of Technology
Academic Year 2025

การสร้างแผนที่เสมือนแบบเวลาจริงสำหรับนำทางยานพาหนะภาคพื้นดินอัตโนมัติโดยใช้ตัวตรวจรู้หลายชนิดร่วมกับเทคนิคปัญญาประดิษฐ์

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาดุษฎีบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

.....

(รศ. ดร.บัณฑิต กฤตาคม)
ประธานกรรมการ

.....

(รศ. ดร.จิระพล ศรีเสริฐผล)
กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

.....

(ผศ. ดร.สุรเดช ตัญตรัยรัตน์)
กรรมการ

.....

(ผศ. ดร.อุเทน ลีตน)
กรรมการ

.....

(ผศ. ดร.โศรฎา แข็งการ)
กรรมการ

.....

(รศ. ดร.ยุพาพร รักสกุลพิวัฒน์)
รักษาการแทนรองอธิการบดีฝ่ายวิชาการและประกันคุณภาพ

.....

(รศ. ดร.พรศิริ จงกล)
คณบดีสำนักวิชาวิศวกรรมศาสตร์

ศิริพงษ์ ปะวะโก : การสร้างแผนที่เสมือนแบบเวลาจริงสำหรับนำทางยานพาหนะภาคพื้นดินอัตโนมัติโดยใช้ตัวตรวจรู้หลายชนิดร่วมกับเทคนิคปัญญาประดิษฐ์ (REAL-TIME VIRTUAL MAPPING FOR GUIDED AGV USING MULTIPLE SENSORS WITH ARTIFICIAL INTELLIGENCE TECHNIQUES)
อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร.จิระพล ศรีเสริฐผล, 121 หน้า.

คำสำคัญ : รถขนส่งอัตโนมัติ/หุ่นยนต์เคลื่อนที่อัตโนมัติ/เซ็นเซอร์/การระบุตำแหน่ง/กล้องวงจรปิด/ปัญญาประดิษฐ์/ระบบควบคุมแบบรวมศูนย์

งานวิจัยนี้มุ่งเน้นการพัฒนาระบบระบุตำแหน่งและนำทางยานพาหนะภาคพื้นดินอัตโนมัติ (Autonomous Ground Vehicle, AGV) ภายในอาคาร โดยใช้กล้องวงจรปิดเป็นเซนเซอร์หลักร่วมกับการประมวลผลภาพและปัญญาประดิษฐ์ ภายใต้แนวคิดการควบคุมแบบรวมศูนย์ (Centralized Control) แทนการใช้เซนเซอร์ราคาแพงหรือเส้นนำทางแบบตายตัวซึ่งมีข้อจำกัดด้านความยืดหยุ่น ระบบที่นำเสนอเริ่มจากการตรวจจับ AGV และสิ่งกีดขวางด้วยโมเดล YOLOv11 ที่ได้รับการฝึกด้วยชุดข้อมูลเฉพาะของพื้นที่ทดสอบ จากนั้นใช้เทคนิค Perspective-n-Point (PnP) เพื่อประมาณค่าพิกัดในระบบโลก และเพิ่มความแม่นยำด้วย Gaussian Process Regression (GPR) เพื่อลดข้อผิดพลาดเชิงเรขาคณิตที่เกิดจากมุมมองกล้องและการบิดเบือนเลนส์ ข้อมูลตำแหน่งที่ได้ถูกนำมาสร้างเป็นแผนที่เสมือนแบบสองมิติ (2D Virtual Map) ซึ่งอัปเดตแบบเรียลไทม์ที่อัตรา 10 Hz และใช้สำหรับการวางเส้นทางอัตโนมัติด้วยอัลกอริทึม A* โดยเส้นทางดังกล่าวถูกส่งต่อไปยัง AGV ผ่านโปรโตคอล Message Queue Telemetry Transport (MQTT) และควบคุมการเคลื่อนที่ด้วย Pure Pursuit Algorithm ที่อาศัยข้อมูลทิศทางจาก Inertial Measurement Unit (IMU) บนตัวรถ ผลการทดลองในพื้นที่ขนาด 4×8 ตารางเมตร แสดงให้เห็นว่าโมเดล YOLOv11 สามารถตรวจจับวัตถุได้ด้วยความแม่นยำเฉลี่ย (Mean Average Precision, mAP@0.5-0.95) 95.60% ที่ความเร็วเฉลี่ย 20 Frames Per Second (FPS) ขณะที่กระบวนการ Localization สามารถลดค่าความคลาดเคลื่อนเฉลี่ยจาก 141 มิลลิเมตรลงเหลือ 22 มิลลิเมตรเมื่อปรับปรุงด้วย GPR และระบบควบคุมสามารถติดตามเส้นทางได้อย่างแม่นยำด้วยค่าความคลาดเคลื่อนเฉลี่ยต่ำกว่า 30 มิลลิเมตรในทุกเส้นทางที่ทดสอบ ผลลัพธ์ดังกล่าวยืนยันว่าระบบที่พัฒนาขึ้นมีศักยภาพสูงในการใช้งานจริง โดยสามารถลดต้นทุนด้านฮาร์ดแวร์ เพิ่มความแม่นยำ และปรับใช้ได้อย่างยืดหยุ่นในสภาพแวดล้อมภายในอาคาร เช่น โรงพยาบาล คลังสินค้า หรืออาคารสำนักงาน

สาขาวิชาวิศวกรรมเมคคาทรอนิกส์ ลายมือชื่อนักศึกษา .....

ปีการศึกษา 2568 ลายมือชื่ออาจารย์ที่ปรึกษา.....

SIRIPONG PAWAKO : REAL-TIME VIRTUAL MAPPING FOR GUIDED AGV USING MULTIPLE SENSORS WITH ARTIFICIAL INTELLIGENCE TECHNIQUES.
THESIS ADVISOR : ASSOC. PROF. JIRAPHON SRISERTPOL, Ph.D., 121 PP.

Keywords: AGV/AMR/Sensor/Localization/Surveillance Camera/Artificial Intelligence/Centralized Control System

This research focuses on the development of a real-time indoor localization and navigation system for Autonomous Ground Vehicles (AGVs) using surveillance cameras as the primary sensors in combination with computer vision and artificial intelligence, under a centralized control architecture. Unlike conventional AGV systems that rely on expensive onboard sensors or fixed guide paths with limited flexibility, the proposed system leverages infrastructure-based cameras to provide cost-effective and adaptable navigation. The system begins with object detection of AGVs and obstacles using a custom-trained YOLOv11 model, followed by position estimation via the Perspective-n-Point (PnP) method, and further refinement using Gaussian Process Regression (GPR) to reduce geometric errors caused by camera perspective and lens distortion. The resulting position data are integrated into a two-dimensional virtual map (2D Virtual Map), which is updated in real time at 10 Hz, and used for automatic path planning with the A* algorithm. The planned trajectory is transmitted to the AGV through the Message Queue Telemetry Transport (MQTT) protocol, while its motion is controlled using the Pure Pursuit algorithm assisted by onboard Inertial Measurement Unit (IMU) measurements. Experimental results in a 4×8 square meters indoor test field demonstrated that YOLOv11 achieved high detection accuracy with a Mean Average Precision (mAP@0.5-0.95) of 95.60% at approximately 20 Frames Per Second (FPS), while the localization error was significantly reduced from an average of 141 mm to only 22 mm after applying GPR. Furthermore, the AGV successfully tracked planned paths with an average tracking error of less than 30 mm across all tested routes. These results confirm that the proposed system offers high accuracy, real-time performance, and cost efficiency,

making it a practical and scalable solution for indoor environments such as hospitals, warehouses, and office buildings.

School of Mechatronics Engineering
Academic Year 2025

Student's Signature S. Pawako
Advisor's Signature

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงได้ ด้วยความกรุณาจากท่านรองศาสตราจารย์ ดร.จิระพล ศรีเสริฐผล อาจารย์ที่ปรึกษาวิทยานิพนธ์และอาจารย์ที่ปรึกษาโครงการต่างๆ ที่ได้เปิดโอกาสให้แสวงหาความรู้ ความสามารถทางวิชาการและชี้แนวทางในการวิจัย คอยสั่งและสอนความรู้และประสบการณ์จนสามารถแตกฉานและก้าวผ่านอุปสรรค์ไปได้ด้วยดี ตลอดทั้งเป็นแบบอย่างที่ดีสำหรับนักวิจัยและพัฒนารุ่นใหม่ และคอยชี้นำในหนทางที่ควรจะเป็น ซึ่งผู้วิจัยขอขอบพระคุณท่านเป็นอย่างสูง และจะนำคำสอนต่าง ๆ ไปเป็นแบบอย่างและถ่ายทอดต่อไป

กราบขอบพระคุณ รองศาสตราจารย์ เรืออากาศเอก ดร.กนต์ธร ชำนิประศาสน์ และผู้ช่วยศาสตราจารย์ ดร. สุรเดช ตัญตรัยรัตน์ ที่ประสิทธิ์ประสาทความรู้ พร้อมทั้งสร้างโอกาสสำหรับช่วงชีวิตนักศึกษาในหลาย ๆ มุมมอง ซึ่งผู้วิจัยขอระลึกไว้ตลอดไป

ขอขอบพระคุณ สำนักงานการวิจัยแห่งชาติ (วช.) ที่ให้ทุนร่วมกับบริษัทสุรนารีแพทย์ภัณฑ์ จำกัด ในโครงการพัฒนานักวิจัยและงานวิจัยเพื่ออุตสาหกรรม (พวอ.) ปี 2565 เลขที่สัญญา N41A650405 ในการศึกษา ณ มหาวิทยาลัยเทคโนโลยีสุรนารี

ขอขอบคุณ คุณอัษฎายุธ รอดพ่าย ที่คอยอำนวยความสะดวกในเรื่องอุปกรณ์ในการวิจัยต่าง ๆ ในการวิจัย

ขอขอบคุณ คุณวิภากานต์ เคล้าเคลีย และเจ้าหน้าที่ประจำสำนักวิชาวิศวกรรมศาสตร์ ที่อำนวยความสะดวกในเรื่องเอกสารต่าง ๆ

ขอขอบคุณกลุ่มวิจัย System and Control Engineering Laboratory (SCE) ที่คอยให้กำลังใจและช่วยเหลือในทุกด้านเกี่ยวกับงานวิจัย

ขอขอบคุณ คุณนรีรัตน์ พงษ์ศักดิ์ ที่คอยให้คำปรึกษาและอยู่เคียงข้างสำหรับทุกช่วงในการทำงานวิจัย

สุดท้ายนี้ ขอกราบขอบพระคุณ คุณพ่อสายรุ้ง และคุณแม่ทองถวิล ปะวะโก ที่ให้กำเนิดรวมไปถึงคุณเอกพงษ์ ปะวะโกที่คอยช่วยกันอบรมเลี้ยงดูให้ความรู้ ความรักและความเอาใจใส่ รวมไปถึงโอกาสทางการศึกษาจนทำให้ผู้วิจัยประสบความสำเร็จในชีวิตเรื่อยมา

ศิริพงษ์ ปะวะโก

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญรูป

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

คำอธิบายสัญลักษณ์และคำย่อ

d_l	=	ระยะทางจากคลื่นแสง (เมตร)
c	=	ความเร็วแสง
t_l	=	เวลาที่แสงเดินทางไปกลับของแสง
f	=	ระยะโฟกัสของกล้อง
b	=	Baseline
d_c	=	Disparity
d_u	=	ระยะทางจากคลื่นเสียง (เมตร)
v	=	ความเร็วเสียงในอากาศ
t_u	=	เวลาที่คลื่นเสียงเดินทางไปกลับของเสียง
$f(n)$	=	ต้นทุนรวมของเส้นทางผ่านโหนด n
$g(n)$	=	ต้นทุนจริงจากจุดเริ่มต้นถึง n
$h(n)$	=	ค่าประมาณระยะทางจาก n ไปยังจุดหมาย
X	=	พิกัดจุดบนโลกแกน X
Y	=	พิกัดจุดบนโลกแกน Y
Z	=	พิกัดจุดบนโลกแกน Z
u	=	ตำแหน่งจุดภาพแนว Wide
v	=	ตำแหน่งจุดภาพแนว Height
s	=	สเกลแฟกเตอร์
A	=	Intrinsic Matrix
R	=	Rotation Vector
T	=	Translation Vector
f_x	=	ความยาวโฟกัสในหน่วยพิกเซลในแนว x
f_y	=	ความยาวโฟกัสในหน่วยพิกเซลในแนว y
c_x	=	จุดตัดของแกนภาพในแนว x
c_y	=	จุดตัดของแกนภาพในแนว y

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

P_w	=	จุดในพิกัดโลก
P_c	=	จุดเดียวกันในระบบกล้อง
a_n	=	ค่าสัมประสิทธิ์ลำดับต่างๆ จาก Polynomial Regression Fitting
∂	=	ชุดข้อมูลฝึกฝนสำหรับ GPR
k	=	Kernel Function ใน GPR
l	=	Length-scale ควบคุม Smoothness ของฟังก์ชัน
σ_f^2	=	Signal Variance ควบคุมความแปรปรวนของฟังก์ชัน
σ_n^2	=	ความแปรปรวนของ Noise
x_*	=	ค่าพยากรณ์
μ	=	ค่ากลางการพยากรณ์
σ^2	=	ความไม่แน่นอนของการพยากรณ์
S	=	Feature Map ที่ได้หลังคอนโวลูชัน
K	=	Kernel Filter
P	=	Pooling Layer Function
C	=	Confidence Score
P_o	=	ความน่าจะเป็นที่มีวัตถุอยู่ในกริดเซลล์ต่างๆ
IoU	=	ค่าความซ้อนทับระหว่าง Predicted Box และ Ground Truth Box
x_c	=	พิกัดศูนย์กลางแกน X
y_c	=	พิกัดศูนย์กลางแกน Y
κ	=	ความโค้งของเส้นทาง
L	=	ระยะฐานล้อของรถ (Wheelbase)
L_d	=	ระยะทาง Lookahead
α	=	มุมระหว่าง Heading กับเส้นตรงไปยัง Lookahead Point
Pre	=	Precision จากการทำนาย
Re	=	Recall
TP	=	True Positives
FP	=	False Positives

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

FN	=	False Negatives
$F1$	=	F1 Score
mAP	=	Mean Average Precision
AP_i	=	Area under Precision–Recall Curve
MAE	=	Mean Absolute Error
$RMSE$	=	Root Mean Square Error
N	=	Number of Sample
y_i	=	ค่าเฉลี่ยของความคลาดเคลื่อนระหว่างค่าจริง
$\hat{y}_i$	=	ค่าที่ระบบอ่านหรือประมาณได้
TE	=	Tracking Error
x_{ref}	=	ตำแหน่งจริงในระนาบแกน x
y_{ref}	=	ตำแหน่งจริงในระนาบแกน y
L_a	=	Control Latency
t_{cmd}	=	เวลาที่หน่วยประมวลผลกลางส่งคำสั่ง
t_{act}	=	เวลาที่ AGV ตอบสนองจริง

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ในช่วงทศวรรษที่ผ่านมา ระบบหุ่นยนต์เคลื่อนที่สำหรับใช้งานภายในอาคารมีบทบาทเพิ่มขึ้นอย่างต่อเนื่องในหลากหลายภาคส่วน โดยเฉพาะในงานที่มีลักษณะทำซ้ำ มีความเสี่ยงต่อความปลอดภัยของมนุษย์ หรือมีความต้องการด้านความแม่นยำและประสิทธิภาพสูง เช่น งานลำเลียงในโรงพยาบาล โรงงานอุตสาหกรรม และพื้นที่ควบคุมเฉพาะด้านต่าง ๆ ในทางปฏิบัติ ระบบดังกล่าวสามารถแบ่งออกได้เป็นสองประเภทหลัก ได้แก่ ระบบที่พึ่งพาสิ่งชี้นำภายนอก ซึ่งเรียกว่า Automated Guided Vehicle (AGV) และระบบที่มีความสามารถในการนำทางด้วยตนเองโดยอาศัยการรับรู้ของหุ่นยนต์เอง ซึ่งเรียกว่า Autonomous Mobile Robot (AMR) อย่างไรก็ตาม ระบบที่ใช้งานภายในอาคารแบบดั้งเดิมจำนวนมากยังคงพึ่งพาแนวทางแรก ซึ่งอาศัยเส้นนำทางแบบตายตัว (fixed guide paths) หรือแนวทางระบบใหม่จำเป็นต้องใช้เซนเซอร์ที่มีต้นทุนสูง เช่น Light Detection and Ranging (LiDAR) หรือกล้องความลึกเฉพาะทาง (Depth Camera) ตลอดจนหน่วยประมวลผลประสิทธิภาพสูง เช่น Central Processing Unit (CPU) หรือ Graphics Processing Unit (GPU) โดยเฉพาะในงานที่เกี่ยวข้องกับปัญญาประดิษฐ์ (Artificial Intelligence: AI) ปัจจัยเหล่านี้ล้วนส่งผลให้ต้นทุนรวมในการติดตั้งและปรับใช้งานของระบบเพิ่มสูงขึ้น และยังจำกัดความยืดหยุ่นในการปรับเปลี่ยนหรือขยายสภาพแวดล้อมการทำงานตามความต้องการในอนาคต

ในเชิงสถาปัตยกรรม ระบบนำทางสำหรับ AGV ภายในอาคารสามารถจำแนกได้เป็นสองแนวทางหลัก ได้แก่ Infrastructure-Free (IF) และ Infrastructure-Based (IB) โดยแนวทาง IF อาศัยเซนเซอร์ที่ติดตั้งบนตัวรถเป็นหลักเพื่อประเมินสถานะและสร้างแผนที่ด้วยตนเอง เช่น เทคโนโลยี Simultaneous Localization and Mapping (SLAM) ซึ่งช่วยลดความจำเป็นในการติดตั้งโครงสร้างพื้นฐานภายนอก อย่างไรก็ตาม แนวทางนี้มักมาพร้อมกับต้นทุนด้านฮาร์ดแวร์และการประมวลผลที่สูงขึ้น รวมถึงความท้าทายที่เกิดจากความคลาดเคลื่อนสะสมของตำแหน่ง (positional drift) ตลอดจนข้อจำกัดที่เกี่ยวข้องกับสภาพแสงหรือสภาพแวดล้อมทางกายภาพของพื้นที่ ในทางตรงกันข้าม แนวทาง IB ใช้โครงสร้างพื้นฐานภายนอกช่วยในการรับรู้และระบุตำแหน่ง เช่น การใช้กล้องวงจรปิดที่มีอยู่เดิมภายในอาคาร ร่วมกับการประมวลผลแบบรวมศูนย์ (centralized processing) ซึ่งช่วยลดภาระด้านการตรวจจับและการคำนวณบนตัวรถ ส่งผลให้ต้นทุนต่อหน่วยลดลง และเอื้อต่อการบริหารจัดการหุ่นยนต์หลายคันจากศูนย์กลาง ภายใต้ข้อจำกัดด้านต้นทุนและ

ความต้องการความยืดหยุ่นในสภาพแวดล้อมที่ควบคุมได้ แนวทาง IB จึงถือเป็นทางเลือกที่เหมาะสมมากกว่า

เพื่อลดข้อจำกัดด้านต้นทุนของระบบและเพิ่มความยืดหยุ่นในการปรับใช้ งานวิจัยนี้นำเสนอการใช้กล้องวงจรปิด (surveillance cameras) เป็นระบบรับรู้ภายนอก (exteroceptive perception) ทำงานร่วมกับสถาปัตยกรรมการประมวลผลแบบรวมศูนย์สำหรับการตรวจจับและติดตามตำแหน่งของ AGV ในเวลาจริง โดยในส่วนการรับรู้ภาพได้ประยุกต์เทคนิคปัญญาประดิษฐ์สำหรับการตรวจจับวัตถุด้วย YOLO (You Only Look Once) เพื่อให้ได้กรอบวัตถุและพิกัดเชิงภาพที่เสถียรและมีอัตราเฟรมเพียงพอต่อการควบคุม แนวทางดังกล่าวไม่เพียงช่วยลดต้นทุนด้านฮาร์ดแวร์ภายในตัวรถ แต่ยังเพิ่มความยืดหยุ่นของระบบในการขยายพื้นที่หรือปรับเปลี่ยนผังอาคาร โดยสามารถอัปเดตหรือติดตั้งกล้องเพิ่มเติม รวมถึงปรับแก้ตรรกะที่ศูนย์กลางได้โดยไม่ส่งผลกระทบต่อระบบควบคุมบนตัวรถโดยตรง

เพื่อยกระดับความแม่นยำและความทนทานในการนำทาง งานวิจัยนี้ผสานข้อมูลจากกล้องวงจรปิดเข้ากับข้อมูลจากตัวตรวจรู้ภายในของ AGV ได้แก่ Inertial Measurement Unit (IMU) และ Encoders ผ่านการสื่อสารด้วย Message Queuing Telemetry Transport (MQTT) ไปยังหน่วยประมวลผลกลาง จากนั้นทำการประมวลผลแบบ Sensor Fusion และส่งคำสั่งควบคุมกลับสู่ AGV ในเวลาจริง กลไกดังกล่าวช่วยให้สามารถสร้างแผนที่เสมือนและเส้นทางนำทางออนไลน์ได้ผ่านซอฟต์แวร์ส่วนกลางที่พัฒนาขึ้นในรูปแบบ Graphical User Interface (GUI) ด้วยภาษา Python ส่งผลให้ระบบมีความยืดหยุ่น ประหยัดต้นทุน และสามารถขยายขอบเขตการใช้งานได้ภายในสภาพแวดล้อมในอาคารที่ใกล้เคียงกับการใช้งานจริง โดยไม่ต้องพึ่งพาเส้นนำทางแบบตายตัวหรือการควบคุมโดยมนุษย์

โดยสรุป แนวทาง IB ที่นำเสนอในงานนี้มีความสำคัญทั้งเชิงวิชาการและเชิงปฏิบัติ กล่าวคือ

1) ลดต้นทุนรวมของระบบต่อคัน ผ่านการย้ายภาระการรับรู้และการคำนวณไปยังโครงสร้างพื้นฐานที่มีอยู่แล้วในอาคาร

2) เพิ่มความยืดหยุ่นและการขยายตัวของระบบ AGV ภายในอาคาร ขณะยังคงความแม่นยำและความเสถียรของการติดตามและควบคุมในระดับที่เพียงพอต่อการใช้งานจริง

1.2 วัตถุประสงค์ของการวิจัย

1) เพื่อพัฒนาอัลกอริทึมการระบุตำแหน่งและนำทางแบบเรียลไทม์สำหรับยานพาหนะภาคพื้นดินอัตโนมัติโดยใช้ข้อมูลจากกล้องวงจรปิดและตัวตรวจรู้ภายใน

2) เพื่อสร้างแผนที่เสมือนจริงจากการประมวลผลภาพผ่านกล้องวงจรปิดและเทคนิค YOLO Object Detection

3) เพื่อออกแบบระบบควบคุมแบบรวมศูนย์ (Centralized Control) บนฐานภาษา Python ที่สามารถเชื่อมต่อกับ AGV ผ่าน MQTT และควบคุมการทำงานได้แบบเรียลไทม์

4) เพื่อศึกษาแนวทางการประยุกต์ใช้เทคนิค Optimization ในการเพิ่มความแม่นยำของการประมาณตำแหน่งสำหรับการประมวลผล

5) เพื่อเสนอแนวทางการออกแบบระบบนำทางภายในอาคารที่สามารถใช้งานร่วมกับกล้องวงจรปิดทั่วไปได้

1.3 สมมุติฐานของการวิจัย

1) กล้องวงจรปิดที่ติดตั้งในตำแหน่งคงที่ภายในพื้นที่ในร่ม สามารถใช้ตรวจจับตำแหน่งของยานพาหนะอัตโนมัติได้อย่างแม่นยำ เมื่อประยุกต์ใช้เทคนิค AI ด้วย YOLO ร่วมกับการวิเคราะห์เชิงเรขาคณิต

2) การหลอมรวมข้อมูลจากกล้องวงจรปิดภายนอก ร่วมกับข้อมูลการเคลื่อนที่จาก IMU และ Encoder ภายในตัวรถ จะเพิ่มความแม่นยำในการประมาณตำแหน่งของ AGV ได้อย่างมีนัยสำคัญ

3) ระบบควบคุมแบบรวมศูนย์ที่ประมวลผลผ่านคอมพิวเตอร์ส่วนกลางสามารถควบคุม AGV ผ่าน MQTT ได้แบบเรียลไทม์ โดยมีความหน่วงของระบบไม่เกินระดับที่ยอมรับได้

4) อัลกอริทึมที่พัฒนาสามารถทำงานได้อย่างมีประสิทธิภาพภายใต้สภาพแวดล้อมจำลองในอาคาร โดยไม่ต้องอาศัยโครงสร้างพิเศษหรือเซนเซอร์ความละเอียดสูงเช่น LIDAR หรือกล้อง RGB-D

1.4 ข้อตกลงเบื้องต้นของการวิจัย

1) สภาพแวดล้อมที่ใช้ในการทดสอบเป็นพื้นที่ในร่มซึ่งมีการควบคุมแสงและไม่มีการเปลี่ยนแปลงโครงสร้างหรือมุมกล้องระหว่างการทดลอง

2) กล้องวงจรปิดที่ใช้มีคุณสมบัติเพียงพอสำหรับการตรวจจับภาพเคลื่อนไหว โดยมีความละเอียดขั้นต่ำ 720p และอัตราเฟรมไม่ต่ำกว่า 10 FPS

3) ตัวแปรภายนอก เช่น การรบกวนของเครือข่าย การสูญเสียข้อมูล หรือการบิดเบือนของภาพจากกล้อง ไม่ส่งผลอย่างมีนัยสำคัญต่อการทำงานของระบบ

4) การสอบเทียบตำแหน่งกล้องและ AGV ดำเนินการล่วงหน้าอย่างถูกต้อง เพื่อให้การแปลงค่าระหว่างระบบพิกัดภาพและพิกัดเชิงพื้นที่มีความถูกต้องเพียงพอ

5) อัลกอริทึม YOLO จะต้องได้รับการฝึกด้วยชุดข้อมูลที่สอดคล้องกับสภาพแวดล้อมและประเภทของวัตถุที่ใช้งานจริง

1.5 ขอบเขตของการวิจัย

1) ระบบจะประยุกต์ใช้กล้องวงจรปิด (Surveillance Camera) ภายนอกอุปกรณ์ AGV และเซนเซอร์ภายใน AGV ได้แก่ IMU และ Encoder เท่านั้น โดยไม่ใช้ LIDAR หรือกล้องเชิงลึก RGB-D

2) ระบบประมวลผลหลักพัฒนาโดยใช้ภาษา Python และทำงานบนคอมพิวเตอร์โน้ตบุ๊กที่เชื่อมต่อกับ AGV ผ่านโปรโตคอล MQTT

3) ระบบบน AGV จะใช้ Jetson Nano ในการรับ-ส่งข้อมูลจากกล้องและเซนเซอร์ภายในพร้อมส่งต่อไปยัง Arduino Mega 2560 เพื่อควบคุมระบบการขับเคลื่อน

4) ความถี่ในการอัปเดตข้อมูลของระบบต้องไม่ต่ำกว่า 10 Hz

5) ความเร็วสูงสุดของ AGV ที่ใช้ในการทดสอบคือ 0.6 เมตรต่อวินาที

6) สภาพแวดล้อมที่ใช้ในการทดสอบจะเป็นพื้นที่จำลองภายในอาคาร ซึ่งมีลักษณะใกล้เคียงกับสถานการณ์ใช้งานจริง เช่น การหลบสิ่งกีดขวาง วัตถุทั่วไป และเส้นทางแคบ

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้ระบบระบุตำแหน่งและนำทาง AGV ที่สามารถใช้ร่วมกับกล้องวงจรปิดภายในอาคารที่มีอยู่เดิม โดยไม่ต้องติดตั้งอุปกรณ์เพิ่มเติมในพื้นที่

2) ได้อัลกอริทึมการสร้างแผนที่และการควบคุม AGV แบบเรียลไทม์ที่มีต้นทุนต่ำและสามารถขยายระบบได้ง่าย

3) ได้ระบบควบคุมอัตโนมัติแบบรวมศูนย์ที่สามารถรับข้อมูลจากหลายแหล่ง และตัดสินใจควบคุม AGV ได้แบบชาญฉลาด

4) เป็นแนวทางต้นแบบในการพัฒนา AGV สำหรับงานภายในอาคารในหน่วยงานที่มีข้อจำกัดด้านงบประมาณ

1.7 โครงสร้างของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ประกอบด้วย 5 บทหลัก ดังนี้:

1) **บทที่ 1** บทนำ: กล่าวถึงที่มา ความสำคัญ วัตถุประสงค์ ขอบเขต และประโยชน์ของงานวิจัย

2) **บทที่ 2** ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง: ศึกษาแนวคิดที่เกี่ยวข้องกับ AGV, การระบุตำแหน่ง, การประมวลผลภาพ, YOLO, ระบบควบคุม และเทคโนโลยีที่เกี่ยวข้อง

3) **บทที่ 3** วิธีดำเนินงานวิจัย: อธิบายขั้นตอนการออกแบบระบบ, วิธีการรวบรวมข้อมูล, อัลกอริธึมที่ใช้, และเกณฑ์การประเมินผล

4) **บทที่ 4** ผลการวิเคราะห์ข้อมูลและอภิปรายผล: นำเสนอผลลัพธ์ที่ได้จากการทดลองระบบในสถานการณ์ต่าง ๆ พร้อมการวิเคราะห์ผล

5) **บทที่ 5** สรุปและข้อเสนอแนะ: สรุปภาพรวมของงานวิจัย วิเคราะห์ข้อดีข้อจำกัด และเสนอแนวทางการพัฒนาในอนาคต

บทที่ 2
ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

ในงานวิจัยนี้ ได้พัฒนาระบบการนำทางยานพาหนะภาคพื้นดินอัตโนมัติ (Automated Guided Vehicle: AGV) ภายในอาคาร โดยใช้กล้องวงจรปิดร่วมกับเทคโนโลยีปัญญาประดิษฐ์เป็นระบบรับรู้ภายนอก (external perception) ตามแนวทาง Infrastructure-Based (IB) ซึ่งอาศัยโครงสร้างพื้นฐานภายนอกช่วยในการรับรู้และระบุตำแหน่งของ AGV แทนการใช้เซนเซอร์ที่มีราคาสูง เช่น Light Detection and Ranging (LIDAR) หรือ RGB-D ที่ติดตั้งบนตัวรถโดยตรง ภาพจากกล้องถูกประมวลผลด้วยสถาปัตยกรรม You Only Look Once (YOLO) เพื่อระบุตำแหน่ง AGV และสิ่งกีดขวาง จากนั้นจึงประมาณค่าตำแหน่งในพิกัดจริงผ่านเทคนิค Perspective-n-Point (PnP) และปรับความแม่นยำด้วย Gaussian Process Regression (GPR) โดยข้อมูลจากกล้องแต่ละตัวจะถูกเชื่อมโยงเพื่อสร้างแผนที่เสมือนของพื้นที่ (virtual map) สำหรับการนำทาง จากตำแหน่งเริ่มต้นไปยังเป้าหมาย ระบบจะสร้างเส้นทางอัตโนมัติด้วย A* algorithm พร้อมตรวจจับสิ่งกีดขวางแบบเวลาจริงด้วย YOLOv11 เพื่อนำมาปรับแผนที่และเส้นทางใหม่เมื่อจำเป็น เส้นทางที่ได้จะถูกติดตามด้วยอัลกอริทึมควบคุม Pure Pursuit ซึ่งใช้พิกัดจากกล้องวงจรปิดและทิศทางการมุ่งหน้า (heading) จาก Inertial Measurement Unit (IMU) เพื่อควบคุมการเคลื่อนที่แบบต่อเนื่อง ข้อมูลทั้งหมดจะถูกส่งและรับคำสั่งระหว่างศูนย์ควบคุม (Laptop) และตัวรถ AGV ผ่านโปรโตคอล Message Queuing Telemetry Transport (MQTT) โดยใช้ Nvidia Jetson Nano เป็น node ฝั่ง AGV และ Arduino Mega 2560 สำหรับควบคุมมอเตอร์ ทั้งระบบจึงเป็นการรวมหลายเทคนิคแบบบูรณาการ ครอบคลุมตั้งแต่ perception, localization, mapping, path planning, control และ communication ภายใต้แนวคิดการควบคุมแบบรวมศูนย์ เพื่อสร้างระบบที่แม่นยำ ต้นทุนต่ำ และเหมาะกับการใช้งานในสภาพแวดล้อมจริง การรวมการประมวลผลทั้งหมดไว้ที่ส่วนกลาง ทำให้สามารถตัดสินใจและควบคุม AGV แบบ real-time โดยไม่ต้องพึ่งความสามารถในการประมวลผลจากอุปกรณ์ onboard มากเกินไป

2.1 ยานพาหนะอัตโนมัติภาคพื้นดิน (Autonomous Ground Vehicles)

ยานพาหนะอัตโนมัติภาคพื้นดินเป็นเทคโนโลยีที่ถูกพัฒนาขึ้นเพื่อลดการพึ่งพาแรงงานมนุษย์ในภารกิจด้านการขนส่ง การลำเลียง และการปฏิบัติงานที่มีลักษณะทำซ้ำ (repetitive tasks) โดยมีจุดเด่นคือสามารถเคลื่อนที่ได้โดยไม่ต้องมีผู้ควบคุมโดยตรง ปัจจุบัน ยานพาหนะประเภทนี้ถูกนำไปใช้

อย่างแพร่หลายทั้งในอุตสาหกรรมการผลิต คลังสินค้า สถานพยาบาล และสภาพแวดล้อมภายในอาคารอื่น ๆ โดยทั่วไป ยานพาหนะอัตโนมัติภาคพื้นดินสามารถจำแนกออกเป็นสองกลุ่มหลักตามระดับความสามารถในการตัดสินใจและความเป็นอัตโนมัติของระบบ ได้แก่ ยานพาหนะที่ควบคุมด้วยเส้นนำ (AGV) และหุ่นยนต์เคลื่อนที่อัตโนมัติแบบอัจฉริยะ (Autonomous Mobile Robot: AMR) ซึ่งทั้งสองกลุ่มมีลักษณะการใช้งานและเทคโนโลยีที่แตกต่างกัน โดยจะอธิบายรายละเอียดเพิ่มเติมในหัวข้อย่อยถัดไป

2.1.1 ยานพาหนะที่ควบคุมด้วยเส้นนำ (AGV)

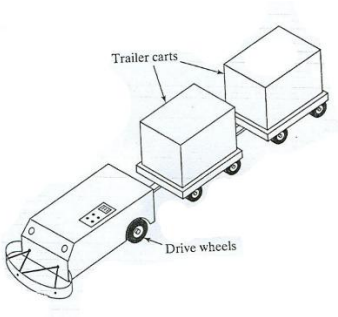
AGV คือ ยานพาหนะภาคพื้นดินอัตโนมัติแบบไร้คนขับที่สามารถเคลื่อนที่โดยอัตโนมัติภายในพื้นที่ที่กำหนด โดยไม่จำเป็นต้องอาศัยการควบคุมจากมนุษย์ขณะปฏิบัติงาน ยานพาหนะประเภทนี้อาศัยระบบการนำทางภายนอกที่ถูกกำหนดไว้ล่วงหน้า เช่น แถบแม่เหล็ก (magnetic tape), เส้นแสง (line following), รหัส Quick Response (QR) หรือป้าย Radio Frequency Identification (RFID) เพื่อระบุตำแหน่งและทิศทางการเคลื่อนที่ AGV เป็นเทคโนโลยีที่ได้รับการพัฒนาและใช้งานมาอย่างยาวนาน โดยถูกนำไปใช้แพร่หลายในหลายภาคส่วนอุตสาหกรรม เช่น สายการผลิตในโรงงานอุตสาหกรรม คลังสินค้าอัตโนมัติ ระบบขนส่งภายในโรงพยาบาล และระบบโลจิสติกส์ในสนามบิน จุดเด่นของ AGV คือความแม่นยำ ความเสถียร และความสามารถในการทำงานซ้ำได้อย่างต่อเนื่องภายใต้สภาพแวดล้อมที่คงที่

AGV สามารถจำแนกได้ตามโครงสร้างทางกล (mechanical configuration) และลักษณะของงานขนส่ง (load handling) ออกเป็น 3 ประเภทหลัก ได้แก่:

2.1.1.1 Drive Train (Towing AGV)

AGV ประเภทนี้ทำหน้าที่เป็นหัวลาก สำหรับพ่วงพาหนะที่ไม่มีระบบขับเคลื่อนในตัวดังรูปที่ 2.1 โดยมีจุดประสงค์เพื่อขนส่งโหลดขนาดใหญ่จำนวนมากในระยะทางไกลภายในพื้นที่โรงงานหรือคลังสินค้า พาหนะพ่วงอาจถูกต่อเป็นขบวน ทำให้สามารถบรรทุกชิ้นงานหลายชิ้นในเที่ยวเดียว ทั้งนี้ ตลอดเส้นทางอาจมีการโหลดหรือปลดสินค้าได้จากรถพ่วงในระหว่างการเดินทาง

1) ข้อดี: รองรับโหลดขนาดใหญ่, ขนส่งได้ครั้งละหลายชิ้น
2) ข้อจำกัด: ต้องมีพื้นที่เลี้ยวที่เพียงพอ, ความยืดหยุ่นต่ำในเส้นทางแคบ



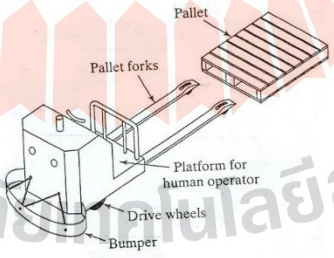
รูปที่ 2.1 รถขนส่งอัตโนมัติแบบ Drive Train (David and Per, 2006)

2.1.1.2 Pallet Truck (Forklift AGV)

AGV ประเภทนี้ดังรูปที่ 2.2 มีลักษณะการทำงานคล้ายกับรถยกพาเลทที่ควบคุมด้วยแรงงานคน โดยทำหน้าที่ขนย้ายสิ่งของหรือวัสดุที่วางบนพาเลทจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่ง AGV สามารถรับพาเลทได้โดยอัตโนมัติ (หรือโดยอาศัยการช่วยเหลือจากมนุษย์ในขั้นต้น) แล้วเคลื่อนที่ไปตามเส้นทางที่กำหนด โดยสามารถตั้งโปรแกรมให้ทำการขนส่งในรูปแบบอัตโนมัติเต็มระบบได้ ปัจจุบัน AGV ประเภทนี้ยังมีความสามารถในการยกสินค้าขึ้นในแนวดิ่ง รวมถึงการจัดเรียงสินค้าบนชั้นวางสูง (High-bay racking)

1) ข้อดี: เหมาะกับคลังสินค้าที่ใช้ระบบพาเลท, ยืดหยุ่นกว่าแบบลาก

2) ข้อจำกัด: ต้องใช้กลไกซับซ้อน เช่น ฟังก์ชันยกและโหลดพาเลท, มีต้นทุนสูงขึ้น



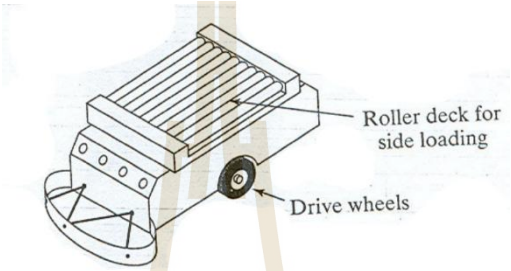
รูปที่ 2.2 รถขนส่งอัตโนมัติแบบ Pallet Truck (David and Per, 2006)

2.1.1.3 แบบ Unit Load Carrier

AGV ประเภทนี้ดังรูปที่ 2.3 มีแพลตฟอร์มที่สามารถบรรทุกชิ้นงานหรือโหลดขนาดกลางบนตัวรถโดยตรง โดยไม่ต้องอาศัยพาเลทหรือพาหนะพ่วง ชิ้นงานมักถูกโหลดและ

ปลดอัตโนมัติโดยระบบ conveyor, roller, lift table หรือกลไกพิเศษเฉพาะทาง AGV ประเภทนี้มักใช้ในสายการผลิตแบบแบ่งสถานี (workstation-based manufacturing)

1) ข้อดี: ทำงานได้อย่างเป็นระบบในสายการผลิตอัตโนมัติ
2) ข้อจำกัด: ต้องออกแบบให้เข้ากันกับอุปกรณ์โหลดอัตโนมัติของแต่ละสถานี



รูปที่ 2.3 รถขนส่งอัตโนมัติแบบ Unit Load Carier (David and Per, 2006)

ตามมาตรฐาน ISO 3691-4:2020 AGV ถูกกำหนดให้เป็นยานพาหนะที่เคลื่อนที่โดยอัตโนมัติบนเส้นทางที่กำหนด ภายใต้การควบคุมของระบบนำทางและการควบคุมแบบลูปปิด โดยไม่ต้องอาศัยการแทรกแซงจากผู้ใช้งานระหว่างปฏิบัติการหลัก อีกทั้งยังต้องมีระบบความปลอดภัยขั้นพื้นฐานเพื่อหยุดหรือหลีกเลี่ยงสิ่งกีดขวาง เช่น bumper switch หรือ infrared proximity sensor ซึ่งสะท้อนให้เห็นว่า AGV มีความเหมาะสมต่อการใช้งานในสภาพแวดล้อมอุตสาหกรรมที่ต้องการความปลอดภัย ความแม่นยำ และความเสถียรของระบบ (International Organization for Standardization, 2020)

เทคโนโลยีที่ใช้ใน AGV แบ่งออกเป็นหลายรูปแบบ ได้แก่:

1) Magnetic Tape Guidance: ใช้แถบแม่เหล็กฝังในพื้นหรือพื้นผิวใกล้พื้นเป็นแนวเส้นนำ
2) Inductive Wire Guidance: ใช้สายไฟฝังในพื้นปล่อยสัญญาณแม่เหล็ก
3) Vision-based Guidance: ใช้กล้องในตัวรถติดตามเส้นสีหรือ QR Code
4) RFID-based Routing: ใช้แท็ก RFID ที่ฝังไว้ตามจุดต่าง ๆ ของเส้นทาง

จากงานสำรวจของ Shaoping นั้น AGV ที่ใช้สายไฟหรือแถบแม่เหล็กยังคงเป็นระบบที่นิยมในสายการผลิต เนื่องจากมีความเสถียรแม้ในสภาพแวดล้อมที่มีการสั่นสะเทือนหรือการสะท้อนสัญญาณ (Shaoping, Chen, Ray and Lihui, 2018) อย่างไรก็ตาม ข้อจำกัดหลักคือความยืดหยุ่นของเส้นทาง หากมีการเปลี่ยนแปลงตำแหน่งของคลังสินค้า ผู้ใช้งานต้องทำการติดตั้งแถบ

แม่เหล็กหรือสายไฟใหม่ ซึ่งมีต้นทุนสูงและสูญเสียด้านเวลาในเชิงการใช้งานจริง AGV ถูกประยุกต์ใช้ในระบบดังต่อไปนี้

1) ระบบสายพานการผลิตอัตโนมัติ (Ata, Jinsong and Liefa, 2019)
2) ระบบขนส่งยาและเวชภัณฑ์ภายในโรงพยาบาล (Petter, 2019)
3) คลังสินค้าและศูนย์กระจายสินค้า (Zhen, Juan and Qing, 2023)
4) อุตสาหกรรมยานยนต์ โดยเฉพาะในกระบวนการประกอบชิ้นส่วน

ตารางที่ 2.1 ข้อเปรียบเทียบผลดีและผลเสียของ AGV

ข้อดีของ AGV	ข้อจำกัดของ AGV
1) ความเสถียรและแม่นยำในสภาพแวดล้อมควบคุม	2) ไม่สามารถเปลี่ยนเส้นทางได้โดยอัตโนมัติ
3) ค่าใช้จ่ายในการซ่อมบำรุงต่ำ	4) ไม่สามารถตัดสินใจเองตามสภาพแวดล้อมใหม่ได้
5) ง่ายต่อการบริหารจัดการและอบรมผู้ใช้งาน	6) ต้องอาศัยโครงสร้างพื้นฐานที่ตายตัวและมีค่าใช้จ่ายในการติดตั้ง

เมื่อเทียบกับ AMR ซึ่งมีความสามารถในการปรับตัวสูงกว่า AGV จึงเหมาะกับงานที่มีลักษณะการทำซ้ำสูงในสภาพแวดล้อมที่เปลี่ยนแปลงน้อย

2.1.2 หุ่นยนต์เคลื่อนที่อัตโนมัติแบบอัจฉริยะ (AMR)

AMR คือ หุ่นยนต์ภาคพื้นดินที่มีความสามารถในการนำทางโดยอัตโนมัติ โดยไม่จำเป็นต้องพึ่งพาเส้นทางตายตัว (fixed-path) หรือโครงสร้างนำทางภายนอก เช่น แถบแม่เหล็กหรือเส้นแสง หุ่นยนต์ประเภทนี้สามารถรับรู้และตีความสภาพแวดล้อม ปรับตัวตามการเปลี่ยนแปลง และตัดสินใจเคลื่อนที่ด้วยตนเอง โดยอาศัยเทคโนโลยีปัญญาประดิษฐ์และการหลอมรวมข้อมูลจากเซนเซอร์หลายประเภท (multi-sensor fusion)

2.1.2.1 การรับรู้ (Perception)

การรับรู้เป็นองค์ประกอบพื้นฐานของระบบหุ่นยนต์เคลื่อนที่แบบอัจฉริยะ AMR ซึ่งทำหน้าที่แปลสัญญาณจากสภาพแวดล้อมภายนอกให้กลายเป็นข้อมูลที่สามารถใช้ในการตัดสินใจและวางแผนการเคลื่อนที่ของหุ่นยนต์ได้ โดยเซนเซอร์ที่เกี่ยวข้องกับระบบการรับรู้ของ AMR มักประกอบด้วยหลายประเภท เพื่อให้ครอบคลุมลักษณะของสิ่งแวดล้อมที่หลากหลาย ซึ่งระบบเหล่านี้มักเรียกว่า Sensor Fusion โดยเซนเซอร์ที่ใช้งานหลักได้แก่:

1) Light Detection and Ranging (LIDAR)

LIDAR เป็นเซนเซอร์ที่ใช้เลเซอร์ในการวัดระยะห่างระหว่างตัวหุ่นยนต์กับวัตถุโดยรอบ โดยการยิงคลื่นเลเซอร์ออกไปและวัดเวลาที่แสงสะท้อนกลับมา (Time-of-Flight) ดังรูปที่ 2.4 ซึ่งสามารถสร้างแผนที่ 2D หรือ 3D ได้อย่างแม่นยำในสภาพแวดล้อมภายในและภายนอกอาคาร โดยสมการที่ (1) เป็นพื้นฐานในการคำนวณระยะทาง:

$$d_l = \frac{ct_l}{2} \tag{2.1}$$

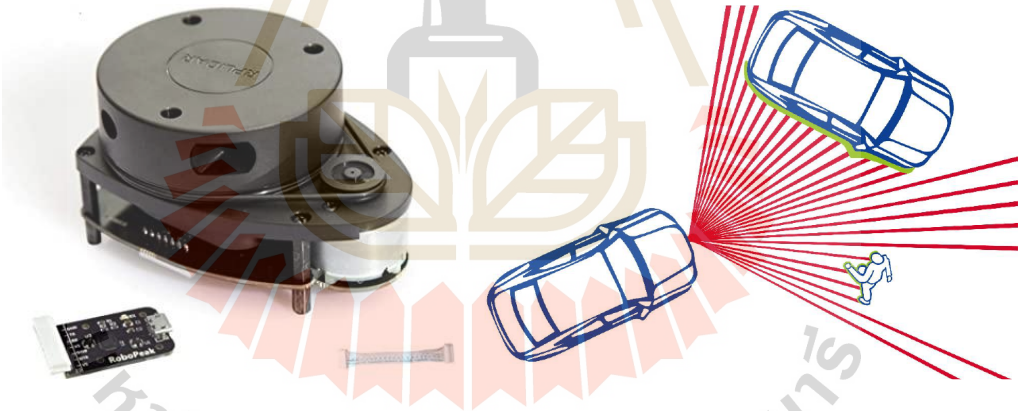
โดยที่:

d_l คือ ระยะทางจากคลื่นแสง (เมตร)

c คือ ความเร็วแสงประมาณ $3x10^8$ m/s

t_l คือ เวลาที่แสงเดินทางไปกลับของแสง

LIDAR มีจุดเด่นในด้าน ความละเอียดสูงและการตรวจจับแบบรอบทิศทาง (360 องศา) แต่มีราคาสูงและไวต่อฝุ่นละอองหรือพื้นผิวโปร่งแสง

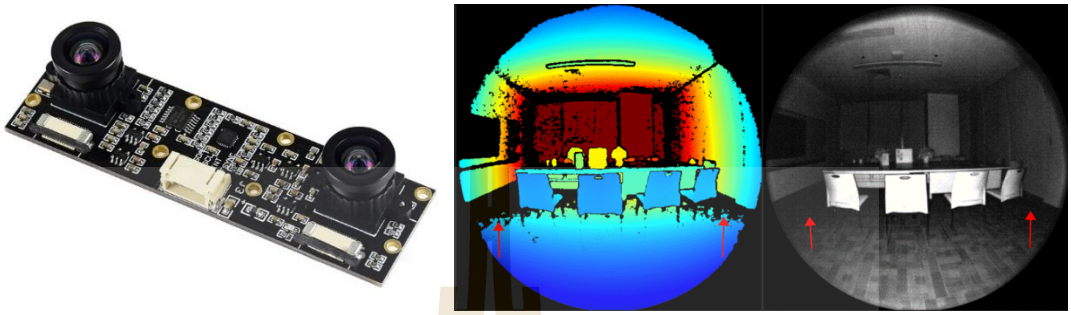


รูปที่ 2.4 Lidar และการรับรู้สภาพแวดล้อม (First Sensor AG., 2021)

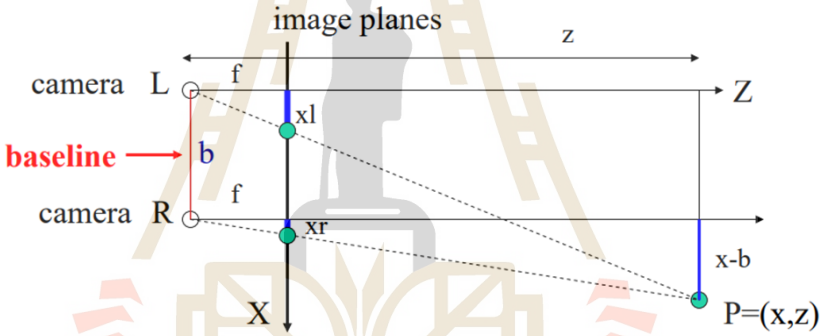
2) กล้อง RGB-D (RGB + Depth Camera)

กล้อง RGB-D เป็นการรวมกล้องสี (RGB Camera) เข้ากับเซนเซอร์สำหรับการวัดระยะเชิงลึก (Depth Sensing) ตัวอย่างเช่น Intel RealSense หรือ Microsoft Kinect โดยอาศัยหลักการ Structured Light หรือ Time-of-Flight (ToF) ในการคำนวณระยะของวัตถุในแต่ละพิกเซล ผลลัพธ์คือ Depth Map ซึ่งให้ค่าระยะลึกของพิกเซลทุกตำแหน่งในภาพ เมื่อรวมกับข้อมูลสี (RGB) จะได้

ข้อมูลเชิงปริมาตร (3D representation) ของสิ่งแวดล้อม สำหรับกรณีของกล้องสเตอริโอ (Stereo RGB-D Camera) ดังรูปที่ 2.5



รูปที่ 2.5 กล้อง RGB-D และการรับรู้สภาพแวดล้อม (Orbbec, 2025)



รูปที่ 2.6 การหาความลึกจากภาพ (Zhangyi et al., 2022)

จากรูปที่ 2.6 การคำนวณความลึกอาศัยระยะห่างระหว่างกล้อง (baseline, b) และตำแหน่งพิกเซลจากกล้องซ้าย (x_l) และขวา (x_r) โดย disparity $d_c = x_l - x_r$ และสามารถคำนวณระยะลึก Z ได้จากสมการที่ (2.2)

$$Z = \frac{fb}{d_c} \tag{2.2}$$

โดยที่

f คือ ระยะโฟกัสของกล้อง

วัตถุใกล้จะให้ค่า disparity มาก (จึงมีค่า Z น้อย) ขณะที่วัตถุไกลจะให้ค่า disparity น้อย (ทำให้ Z มีค่ามาก) นอกจากนี้ หากทราบค่าพิกัดภาพและพารามิเตอร์ภายในกล้อง (intrinsic

parameters) จะสามารถคำนวณเป็นพิกัดสามมิติ (X, Y, Z) ของวัตถุได้ ซึ่งทำให้หุ่นยนต์สามารถระบุรูปร่างและตำแหน่งของวัตถุในเชิงพื้นที่ได้อย่างแม่นยำ โดยเฉพาะอย่างยิ่งในสภาพแวดล้อมภายในอาคาร (Indoor Environment)

3) Inertial Measurement Unit (IMU)

IMU ดังรูปที่ 2.7 เป็นเซนเซอร์ที่ทำหน้าที่ตรวจวัดการเคลื่อนที่และการหมุนในสามมิติ โดยทั่วไปประกอบด้วย:

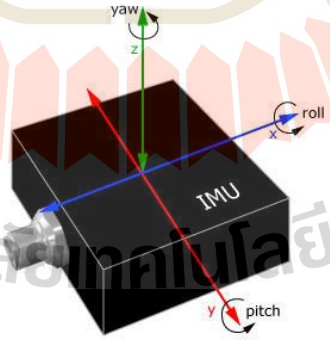
1) Accelerometer สำหรับวัดความเร่งเชิงเส้น (Linear Acceleration) บนแกน (x, y, z)

2) Gyroscope สำหรับวัดความเร็วเชิงมุม (Angular Velocity)

3) Magnetometer สำหรับวัดทิศทางสัมพันธ์กับสนามแม่เหล็กโลก เพื่อใช้ในการปรับแก้ค่าการหมุน (Heading)

ค่าที่ได้จากเซนเซอร์เหล่านี้สามารถนำมาคำนวณ Orientation ของหุ่นยนต์ ซึ่งสามารถแทนได้ทั้งในรูปของ Euler Angles (Roll, Pitch, Yaw) หรือ Quaternion โดย Quaternion มีข้อดีคือหลีกเลี่ยงปัญหา Gimbal Lock ที่อาจเกิดขึ้นใน Euler Angles

ในเชิงการประยุกต์ ข้อมูลจาก IMU มักถูกนำมาใช้ร่วมกับการระบุตำแหน่ง (Localization) เพื่อช่วยลดความคลาดเคลื่อนของระบบ ตัวอย่างเช่น Dead Reckoning ซึ่งอาศัยการบูรณาการความเร่งและความเร็วเชิงมุมเพื่อประมาณตำแหน่งปัจจุบันของหุ่นยนต์เมื่อไม่มีข้อมูลจากเซนเซอร์ภายนอก หรือการใช้ IMU เป็น Feedback Sensor ในระบบควบคุมการเคลื่อนที่แบบเรียลไทม์ เพื่อปรับปรุงความเสถียรและความแม่นยำ



รูปที่ 2.7 IMU และพฤติกรรมการรับรู้ (U.S. Geological Survey, 2025)

4) Ultrasonic Sensor

เซนเซอร์อัลตราโซนิกทำงานโดยส่งคลื่นเสียงความถี่สูง (Ultrasound) และวัดเวลาที่คลื่นสะท้อนกลับมาดังรูปที่ 2.8 โดยหลักการคล้ายกับ LIDAR แต่ใช้เสียงแทนแสง มีระยะทำงานสั้นอยู่

ในช่วง 10–400 ซม. และต้นทุนต่ำ เหมาะสำหรับหลีกเลี่ยงสิ่งกีดขวางในระยะใกล้ การคำนวณระยะจะคำนวณดังสมการคำนวณระยะทาง:

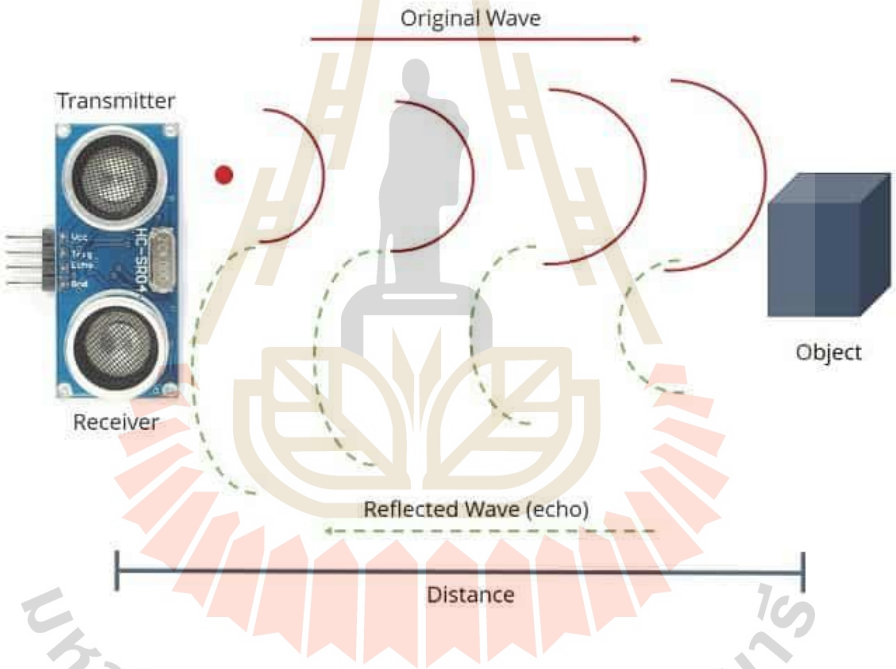
$$d_u = \frac{vt_u}{2} \tag{2.3}$$

โดยที่:

d_u คือ ระยะทางจากคลื่นเสียง (เมตร)

v คือ ความเร็วเสียงในอากาศประมาณ 343 m/s

t_u คือ เวลาที่คลื่นเสียงเดินทางไปกลับของเสียง



รูปที่ 2.8 Ultrasonic และพฤติกรรมการรับรู้ (Panagorko, 2020)

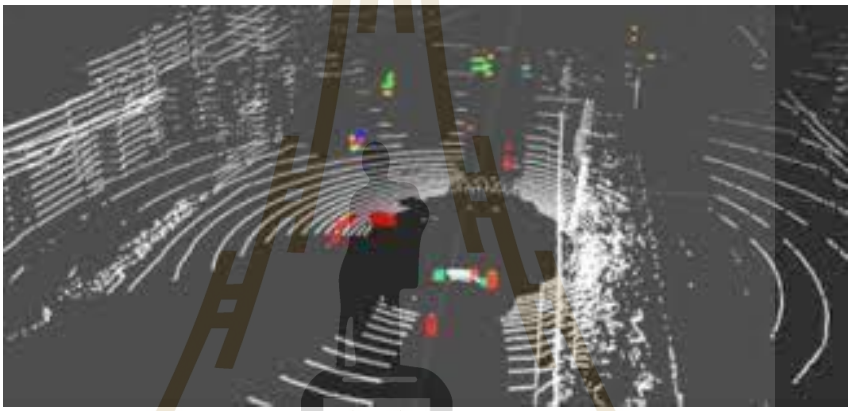
2.1.2.2 การระบุตำแหน่งและการสร้างแผนที่ (Localization and Mapping)

Localization and Mapping หรือการระบุตำแหน่งและการสร้างแผนที่ คือกระบวนการสำคัญในระบบ AMR ที่ช่วยให้หุ่นยนต์สามารถระบุว่าตนเองอยู่ที่ใดในพื้นที่ (Localize) และสามารถสร้างแบบจำลองของสภาพแวดล้อม (Map) เพื่อใช้ในการนำทาง หลีกเลี่ยงสิ่งกีดขวาง และตัดสินใจเคลื่อนที่อย่างมีประสิทธิภาพ

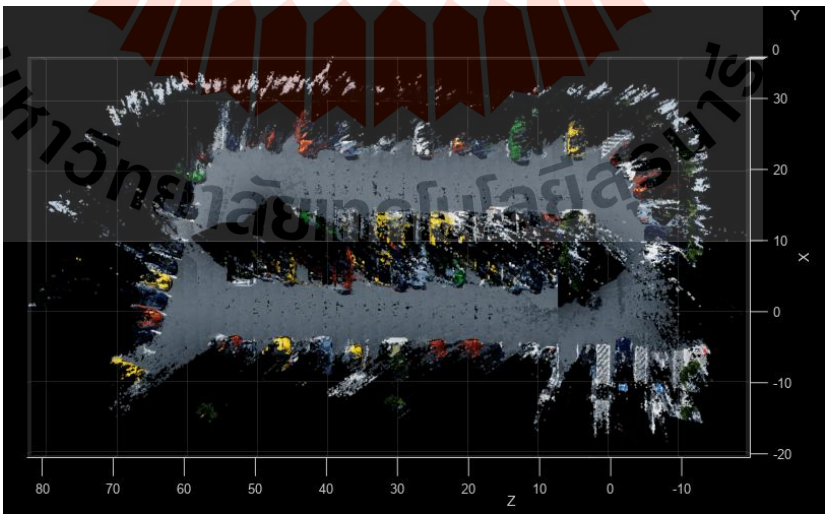
ระบบ Localization and Mapping สามารถแบ่งออกเป็นหลายแนวทาง ตามลักษณะของแหล่งข้อมูลและโครงสร้างพื้นฐาน ดังนี้:

1) Simultaneous Localization and Mapping (SLAM)

SLAM เป็นแนวทางดั้งเดิมที่หุ่นยนต์ใช้เซนเซอร์เช่น LIDAR, RGB-D, หรือ Visual Camera เพื่อสร้างแผนที่ไปพร้อมกับการระบุตำแหน่งของตนเองแบบเรียลไทม์ มีหลายเทคนิค เช่น LIDAR-based SLAM โดยจะใช้ LIDAR เป็นหลักสร้างแผนที่ 2D/3D แบบแม่นยำดังรูป 2.9 หรือ Visual SLAM (V-SLA
M) โดยใช้กล้อง RGB หรือ RGB-D เป็นหลักเช่น ORB-SLAM, DSO, RTAB-Map ดังรูปที่ 2.10



รูปที่ 2.9 Lidar-based SLAM Surrounding (Feng, Donghui, Weisong, Jiachen and Li-Ta, 2021)

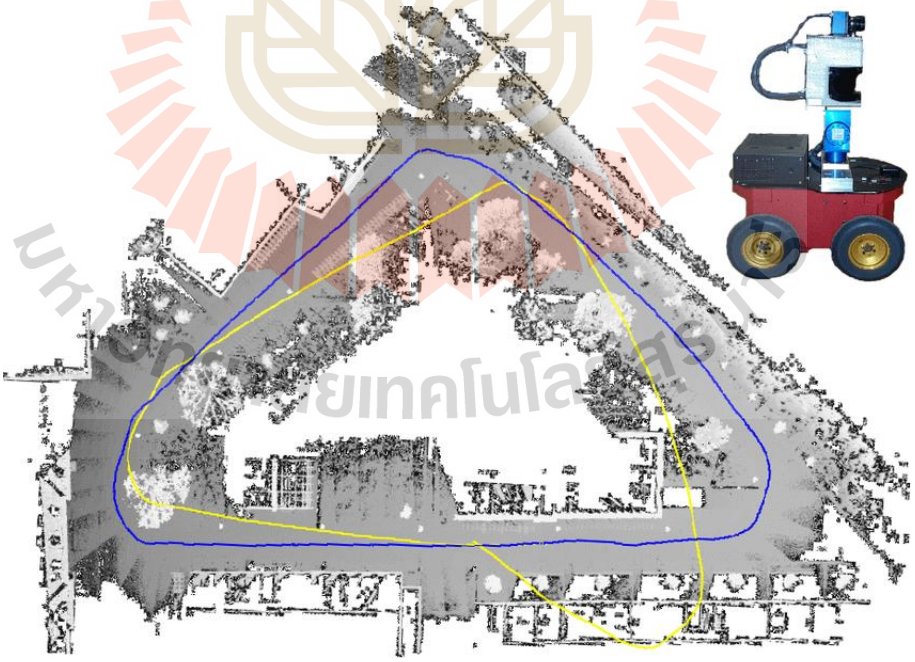


รูปที่ 2.10 Visual-based SLAM Surrounding (MathWorks, 2025)

อย่างไรก็ตาม SLAM ต้องการการประมวลผลสูง เนื่องจากต้องทำการคำนวณหลายขั้นตอนพร้อมกัน ได้แก่ การตรวจจับและจับคู่คุณลักษณะ (Feature Extraction & Matching) การคำนวณการเคลื่อนที่สัมพัทธ์ (Pose Estimation) และการอัปเดตแผนที่ (Map Update) แบบต่อเนื่องในเวลาจริง ซึ่งกระบวนการเหล่านี้มักมีความซับซ้อนเชิงคณิตศาสตร์และเชิงคอมพิวเตอร์สูง โดยเฉพาะอย่างยิ่งเมื่อใช้วิธีการที่อิงกับการประมวลผลภาพหรือการวิเคราะห์ข้อมูลสามมิติ นอกจากนี้ หากต้องการความแม่นยำระดับสูง ระบบ SLAM มักต้องพึ่งพาเซนเซอร์ที่มีคุณภาพสูงดังกล่าวข้างต้น ซึ่งมีราคาสูงและต้องการหน่วยประมวลผลที่มีประสิทธิภาพ ส่งผลให้ต้นทุนรวมของระบบเพิ่มขึ้น และเป็นข้อจำกัดสำคัญในการประยุกต์ใช้งานในเชิงพาณิชย์หรือในสภาพแวดล้อมที่มีข้อจำกัดด้านงบประมาณ

2) การระบุตำแหน่งโดยไม่พึ่งโครงสร้างพื้นฐาน (Infrastructure-Free Localization)

Infrastructure-Free (IF) Localization คือแนวทางการระบุตำแหน่งของหุ่นยนต์โดยไม่พึ่งพาองค์ประกอบภายนอกที่ต้องติดตั้งล่วงหน้า เช่น เส้นแม่เหล็ก, QR Code, RFID หรือป้ายกำกับทางกายภาพ แนวทางนี้อาศัยความสามารถของเซนเซอร์ที่ติดตั้งบนตัวหุ่นยนต์เอง ได้แก่ LIDAR, กล้อง RGB-D, IMU หรือ Ultrasonic Sensor เพื่อรับรู้สภาพแวดล้อม และนำข้อมูลที่ได้ไปประมวลผลร่วมกับอัลกอริทึมต่างๆ ตัวอย่างดังรูปที่ 2.11 ที่ใช้แนวทางการระบุตำแหน่งด้วย IF

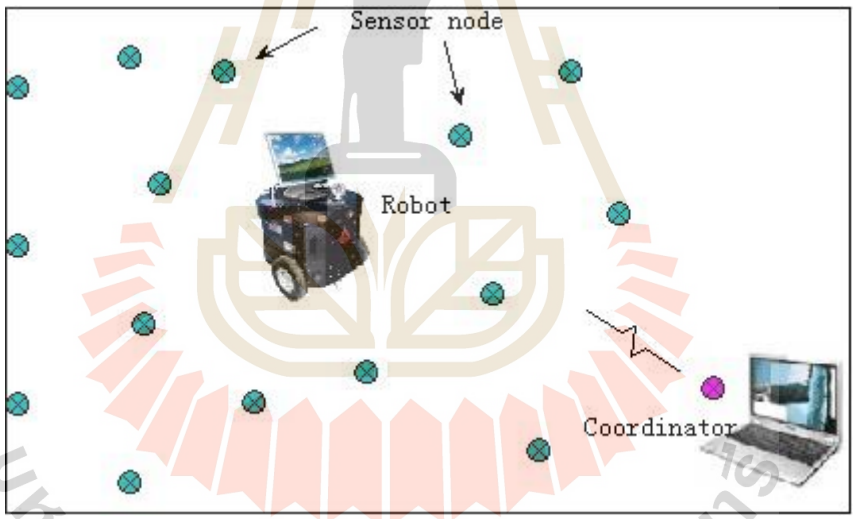


รูปที่ 2.11 Robot และ Infrastructure-Free Localization (Rainer, Patrick, Rudolph and Wolfram, 2007)

3) การระบุตำแหน่งโดยใช้โครงสร้างพื้นฐานภายนอก (Infrastructure-Based Localization)

Infrastructure-Based (IB) Localization คือ แนวทางการระบุตำแหน่งของหุ่นยนต์หรือยานพาหนะอัตโนมัติ โดยอาศัยโครงสร้างพื้นฐานที่ถูกติดตั้งไว้ล่วงหน้าในพื้นที่ปฏิบัติงาน เช่น เส้นแม่เหล็ก (Magnetic Tape), รหัส QR หรือ RFID Tag, Reflective Marker, Ultra-Wideband (UWB) Beacon, กล้องวงจรปิด (Surveillance Camera) หรือแม้แต่สัญญาณ Wi-Fi ที่ติดตั้งภายในอาคาร จุดประสงค์ของโครงสร้างพื้นฐานเหล่านี้คือเพื่อให้ระบบสามารถระบุตำแหน่งของตนเองได้อย่างแม่นยำและเสถียร โดยไม่จำเป็นต้องพึ่งการประมวลผลที่ซับซ้อนบนตัวหุ่นยนต์

แนวทางนี้ถูกใช้อย่างแพร่หลายใน AGV แบบดั้งเดิม โดยเฉพาะในโรงงานอุตสาหกรรม คลังสินค้าอัตโนมัติ และโรงพยาบาล ซึ่งต้องการระบบที่เสถียร คาดการณ์ได้ และมีความน่าเชื่อถือสูง เนื่องจากตำแหน่งและเส้นทางถูกกำหนดจากโครงสร้างพื้นฐานที่ติดตั้งไว้ล่วงหน้าดังรูปที่ 2.12 ทำให้การเคลื่อนที่ของหุ่นยนต์มีความแน่นอนและควบคุมได้ง่าย



รูปที่ 2.12 Robot และ Infrastructure-Base Localization (Lingfei et al., 2009)

อย่างไรก็ตาม ข้อจำกัดสำคัญของ Infrastructure-Based Localization คือ ความยืดหยุ่นต่ำ หากมีการปรับเปลี่ยน layout ของพื้นที่ เช่น การย้ายสายการผลิตหรือปรับผังคลังสินค้า จำเป็นต้องติดตั้งหรือตั้งค่าโครงสร้างนำทางใหม่ทั้งหมด ซึ่งก่อให้เกิด ต้นทุนสูงและใช้เวลาในการติดตั้ง เมื่อเปรียบเทียบกับ Infrastructure-Free Localization ที่สามารถปรับเปลี่ยนสภาพแวดล้อมได้ง่ายกว่า ข้อเปรียบเทียบอื่น ๆ จะแสดงดังตารางที่ 2.2

ตารางที่ 2.2 แสดงข้อเปรียบเทียบระหว่าง IF และ IB

เกณฑ์เปรียบเทียบ	Infrastructure-Free (IF)	Infrastructure-Based (IB)
หลักการทำงาน	ใช้เซนเซอร์บนหุ่นยนต์ (LIDAR, RGB-D, IMU)	ใช้โครงสร้างนำทางภายนอก (QR, RFID, Magnetic Tape, UWB, Camera)
ความยืดหยุ่น	สูง – ปรับเปลี่ยนพื้นที่ได้ทันที	ต่ำ – ต้องติดตั้ง/ปรับโครงสร้างใหม่
ต้นทุนเริ่มต้น	สูง (เพราะใช้เซนเซอร์ขั้นสูง + หน่วยประมวลผล)	ปานกลางถึงสูง (ขึ้นกับจำนวนโครงสร้างที่ติดตั้ง)
ต้นทุนระยะยาว	ต่ำ – ไม่ต้องติดตั้งเพิ่มเมื่อเปลี่ยน layout	สูง – ต้องซ่อมบำรุงและติดตั้งใหม่หาก layout เปลี่ยน
ความแม่นยำ/เสถียร	ขึ้นกับคุณภาพเซนเซอร์และอัลกอริทึม อาจแกว่งได้	เสถียรและแม่นยำสูงในสภาพแวดล้อมคงที่
การประมวลผล	สูง – ต้องใช้ SLAM/VO และ Sensor Fusion	ต่ำ – หุ่นยนต์เพียงอ่านค่าโครงสร้างนำทาง
การใช้งานที่เหมาะสม	พื้นที่ Dynamic, Layout เปลี่ยนบ่อย เช่น คลังสินค้าใหม่	พื้นที่ Fixed, Layout คงที่ เช่น สายการผลิต, โรงพยาบาล

2.1.2.3 Path Planning (การวางแผนเส้นทาง)

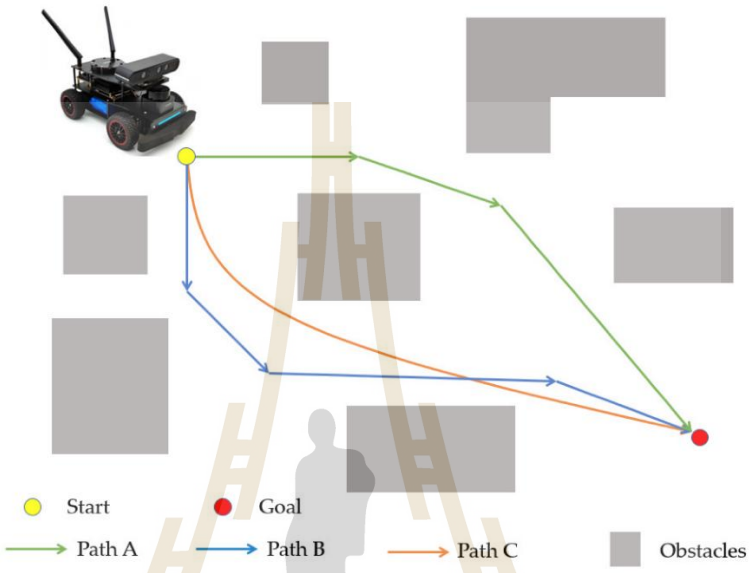
การวางแผนเส้นทาง (Path Planning) หมายถึงกระบวนการที่หุ่นยนต์คำนวณเส้นทางที่เหมาะสมจากตำแหน่งเริ่มต้นไปยังตำแหน่งเป้าหมาย โดยคำนึงถึงสิ่งกีดขวางในสภาพแวดล้อม เพื่อให้สามารถเคลื่อนที่ไปยังจุดหมายได้อย่างปลอดภัย รวดเร็ว และมีประสิทธิภาพ ดังรูปที่ 2.13 การวางแผนเส้นทางแบ่งออกเป็นสองระดับหลัก ได้แก่ Global Path Planning ซึ่งใช้ข้อมูลแผนที่ทั้งหมดในการหาทางที่เหมาะสมที่สุด และ Local Path Planning ซึ่งเน้นการปรับเส้นทางแบบเรียลไทม์เพื่อหลีกเลี่ยงสิ่งกีดขวางที่ตรวจพบระหว่างการเคลื่อนที่

แนวทาง Path Planning สามารถจำแนกได้หลายวิธี เช่น

1) Graph-based algorithms: ใช้โครงสร้างกราฟ เช่น Grid Map หรือ Voronoi Diagram เพื่อหาเส้นทางที่เหมาะสม (เช่น Dijkstra, A*)

2) Sampling-based algorithms: ใช้วิธีการสุ่มตัวอย่างในพื้นที่เพื่อสร้างเส้นทางที่เป็นไปได้ เช่น Rapidly-Exploring Random Tree (RRT), Probabilistic Roadmap (PRM)

3) Optimization-based algorithms: ใช้การหาคำตอบที่เหมาะสมที่สุด (Optimal Solution) ภายใต้ข้อจำกัด เช่น Genetic Algorithm (GA), Ant Colony Optimization (ACO)



รูปที่ 2.13 การทำงานของ Path Planning (Yi, Yunchuan, Jiakai and Zhiqiang, 2022)

A* Algorithm

อัลกอริทึม A* เป็นหนึ่งในวิธีที่นิยมใช้มากที่สุดสำหรับ Global Path Planning เนื่องจากมีคุณสมบัติทั้ง Optimality (หาทางที่สั้นที่สุด) และ Completeness (หาทางได้เสมอถ้ามีอยู่) โดยใช้ฟังก์ชันค่าใช้จ่าย (cost function) ดังสมการที่ (2.4):

$$f(n) = g(n) + h(n) \tag{2.4}$$

โดยที่

$f(n)$ คือ ต้นทุนรวมของเส้นทางผ่านโหนด n

$g(n)$ คือ ต้นทุนจริงจากจุดเริ่มต้นถึง n

$h(n)$ คือ ค่าประมาณระยะทางจาก n ไปยังจุดหมาย (เช่น Manhattan distance, Euclidean distance)

A* มีข้อได้เปรียบตรงที่หากใช้ Heuristic ที่เหมาะสม เช่น Euclidean Distance หรือ Manhattan Distance จะทำให้การค้นหาเร็วและแม่นยำมากยิ่งขึ้นและ A* จะเลือกขยายโหนด

ที่มีค่า $f(n)$ ต่ำที่สุดในแต่ละขั้นตอน ทำให้สามารถหาทางที่เหมาะสมและมีประสิทธิภาพสูงกว่าวิธีการพื้นฐานเช่น Dijkstra Algorithm

ตารางที่ 2.3 เปรียบเทียบอัลกอริทึม Path Planning

อัลกอริทึม	หลักการทำงาน	จุดเด่น	ข้อจำกัด	การใช้งานทั่วไป
Dijkstra	คำนวณระยะทางที่สั้นที่สุดจากจุดเริ่มต้นถึงทุกโหนด โดยไม่ใช้ heuristic	รับประกันได้เส้นทางที่สั้นที่สุด	ประสิทธิภาพต่ำเมื่อกราฟมีขนาดใหญ่ (ช้า)	การหาทางบนแผนที่ขนาดเล็กที่ไม่ซับซ้อน
A*	ใช้สมการ $f(n)$ เพื่อเลือกเส้นทางที่ cost ต่ำที่สุด	เร็วกว่า Dijkstra, ได้เส้นทาง optimal ถ้า heuristic ถูกต้อง	ประสิทธิภาพขึ้นกับ heuristic; อาจใช้เวลานานในแผนที่ซับซ้อนมาก	การวางแผนเส้นทางใน AMR, หุ่นยนต์ในอาคาร
RRT (Rapidly-Exploring Random Tree)	สุ่มจุดในพื้นที่เพื่อสร้าง tree เชื่อมโยงไปยังเป้าหมาย	ทำงานได้ดีในพื้นที่ที่ซับซ้อนและไม่ทราบแผนที่ล่วงหน้า	เส้นทางที่ได้ไม่ optimal และอาจต้อง post-processing	การนำทางในพื้นที่ dynamic หรือ outdoor

ในระบบ AMR ภายในอาคาร มักนิยมใช้อัลกอริทึม A* สำหรับการวางแผนเส้นทางในระดับ Global บนแผนที่กริด เนื่องจากให้ความแม่นยำและคาดการณ์ได้ ส่วน Local Planning มักใช้อัลกอริทึมเสริม เช่น Dynamic Window Approach (DWA) หรือ Velocity Obstacle (VO) เพื่อหลีกเลี่ยงสิ่งกีดขวางแบบเวลาจริง

2.1.2.4 การตัดสินใจอัตโนมัติ (Autonomous Decision-Making)

Autonomous Decision-Making (ADM) คือความสามารถของหุ่นยนต์เคลื่อนที่อัตโนมัติ AMR ในการวิเคราะห์สถานการณ์ในสภาพแวดล้อม และเลือกการกระทำที่เหมาะสมโดยไม่ต้องอาศัยคำสั่งจากมนุษย์แบบเรียลไทม์ การตัดสินใจดังกล่าวอาจอยู่ในรูปแบบกฎ

เชิงตรรกะ (Rule-based), พฤติกรรมที่กำหนดไว้ล่วงหน้า (Behavior-based) หรือการเรียนรู้จากข้อมูลด้วยวิธีการทางปัญญาประดิษฐ์ (AI-based) เช่น Reinforcement Learning

การตัดสินใจอัตโนมัติ (ADM) ของ AMR สามารถแบ่งได้เป็น 3 กลุ่มหลัก ได้แก่ Rule-based, Behavior-based และ AI-based แต่ละกลุ่มมีจุดเด่นและข้อจำกัดต่างกันดังตารางที่ 2.4 โดย Rule-based เหมาะกับระบบที่มีสภาพแวดล้อมคงที่และต้องการความน่าเชื่อถือสูง Behavior-based ให้ความยืดหยุ่นและตอบสนองต่อสิ่งแวดล้อมแบบเรียลไทม์ ส่วน AI-based มีศักยภาพสูงสุดในการปรับตัวและเรียนรู้พฤติกรรมที่ซับซ้อน แต่มีความต้องการทรัพยากรการประมวลผลและข้อมูลสูง

ตารางที่ 2.4 ประเภทของ Autonomous Decision-Making (ADM)

ประเภท	หลักการทำงาน	จุดเด่น	ข้อจำกัด	ตัวอย่างการใช้งาน
Rule-based (เช่น FSM – Finite State Machine)	ใช้กฎเชิงตรรกะ (if-else) หรือสถานะที่กำหนดไว้ล่วงหน้าในการเลือกการกระทำ	เข้าใจง่าย, พัฒนาได้เร็ว, มีความน่าเชื่อถือในสภาพแวดล้อมที่คงที่	ยืดหยุ่นต่ำ, ไม่เหมาะกับสภาพแวดล้อมที่เปลี่ยนแปลง	หยุดเมื่อเจอสิ่งกีดขวาง, เปลี่ยนเส้นทางเมื่อ sensor แจ้งทางตัน
Behavior-based (เช่น Subsumption Architecture)	หุ่นยนต์เลือกการกระทำจากชุดพฤติกรรมที่กำหนดโดยมีการจัดลำดับความสำคัญ (priority)	ตอบสนองต่อสิ่งแวดล้อมแบบเรียลไทม์, ยืดหยุ่นกว่าระบบกฎ	ซับซ้อนขึ้นเมื่อจำนวนพฤติกรรมมาก, อาจขัดแย้งกันระหว่างพฤติกรรม	หลีกเลี่ยงสิ่งกีดขวางโดยพฤติกรรม “avoid”, เคลื่อนที่ไปเป้าหมายโดยพฤติกรรม “go-to-goal”
AI-based (เช่น Reinforcement Learning, MDP, Neural Networks)	ใช้การเรียนรู้จากข้อมูลหรือการทดลองซ้ำเพื่อหานโยบาย (policy) ที่เหมาะสมในการตัดสินใจ	ยืดหยุ่นสูง, สามารถปรับตัวตามสภาพแวดล้อมที่เปลี่ยนไป, เรียนรู้พฤติกรรมซับซ้อนได้	ต้องการข้อมูลจำนวนมาก, ใช้การประมวลผลสูง, ยากต่อการอธิบายการตัดสินใจ (black-box)	AMR ใช้ Deep RL ในการเลือกเส้นทางเลี่ยงสิ่งกีดขวางในสภาพแวดล้อม dynamic

2.1.3 ตัวอย่างเทคโนโลยีและอัลกอริทึมที่ใช้ใน AMR

ตามมาตรฐาน ISO 8373:2012 หุ่นยนต์ที่สามารถปรับเปลี่ยนเส้นทางได้ตามสภาพแวดล้อมโดยไม่ต้องพึ่งพาโครงสร้างนำทางภายนอก จัดอยู่ในกลุ่ม Autonomous Service Robots ซึ่งต้องอาศัยอัลกอริทึมที่หลากหลายเพื่อให้ระบบสามารถรับรู้ (Perception), ระบุตำแหน่ง (Localization), สร้างแผนที่ (Mapping), วางแผนเส้นทาง (Path Planning), ตัดสินใจ (Decision-making) และควบคุมการเคลื่อนที่ (Control) ได้อย่างครบวงจร ตัวอย่างเทคโนโลยีและอัลกอริทึมที่ถูกใช้ใน AMR สามารถสรุปได้ดังตารางที่ 2.5

ตารางที่ 2.5 สรุปเทคโนโลยีที่ถูกใช้ใน AMR

หมวด	เทคโนโลยี / อัลกอริทึม	หลักการทำงาน	จุดเด่น	ข้อจำกัด	ตัวอย่างการใช้งาน
Perception / Object Detection	YOLO (You Only Look Once)	ใช้ CNN ตรวจจับวัตถุจากภาพในขั้นตอนเดียว (single-shot)	ตรวจจับวัตถุแบบเรียลไทม์, ความเร็วสูง	ความแม่นยำลดลงเมื่อวัตถุเล็กหรือซับซ้อน	ตรวจจับ AGV, สิ่งกีดขวางในโรงงาน
	Faster R-CNN	Region Proposal + CNN	ความแม่นยำสูง, ตรวจจับวัตถุซับซ้อนได้ดี	ใช้เวลาประมวลผลมาก, ไม่เหมาะกับ real-time	งานวิจัยด้าน Robot Vision
	SSD (Single Shot Detector)	Multi-scale feature maps	สมดุลระหว่างความเร็วและความแม่นยำ	ช้ากว่า YOLO เล็กน้อยในงาน real-time	ตรวจจับวัตถุใน AMR logistics
Localization & Mapping	ORB-SLAM / ORB-SLAM2	ใช้ ORB feature ใน Visual SLAM	ฟรี, ใช้งานได้จริง, ครอบคลุม monocular/stereo/RGB-D	มี drift ถ้าไม่มี loop closure	การสร้างแผนที่ใน indoor AMR
	Cartographer (Google)	LIDAR-based SLAM + scan matching	แม่นยำสูง, รองรับ multi-floor	ต้องใช้ LIDAR, ใช้ compute สูง	AMR ในคลังสินค้า

ตารางที่ 2.5 สรุปเทคโนโลยีที่ถูกใช้ใน AMR (ต่อ)

หมวด	เทคโนโลยี / อัลกอริทึม	หลักการทำงาน	จุดเด่น	ข้อจำกัด	ตัวอย่างการใช้งาน
	Visual Odometry (VO)	ติดตามการเปลี่ยนแปลงของภาพ	ไม่ต้องใช้ marker ภายนอก	มี drift สะสม	ใช้ใน AMR indoor/outdoor
Path Planning	Dijkstra	หาเส้นทางที่สั้นที่สุดจากทุกโหนด	หาคำตอบ optimal	ประสิทธิภาพต่ำ, ไม่เหมาะกับกราฟใหญ่	การวางแผนพื้นฐาน
	A*	ใช้ cost function (f(n)=g(n)+h(n))	เร็วกว่า Dijkstra, ได้ผล optimal ถ้า heuristic ดี	ประสิทธิภาพขึ้นกับ heuristic	Global planning ใน AMR
	RRT (Rapidly-Exploring Random Tree)	สุ่มจุดใน space สร้าง tree ไปยังเป้าหมาย	ใช้ได้ในพื้นที่ซับซ้อน	ไม่ optimal, ต้องปรับเส้นทางเพิ่ม	AMR ในพื้นที่ dynamic
	DWA (Dynamic Window Approach)	คำนวณความเร็วและทิศทางในกรอบเวลาสั้น	เหมาะกับ local planning	อาจติด local minima	หลีกเลี่ยงสิ่งกีดขวาง real-time
Control / Tracking	Pure Pursuit	เลือก look-ahead point แล้วปรับโค้งรถ	ติดตามเส้นทางราบรื่น	ต้องการข้อมูลตำแหน่งแม่นยำ	ติดตามเส้นทาง indoor AMR
	PID Control	ควบคุมค่าความผิดพลาด (error)	ใช้งานง่าย, เสถียร	ไม่เหมาะกับระบบ nonlinear	ควบคุมมอเตอร์ล้อ AGV
Sensor Fusion / State Estimation	EKF (Extended Kalman Filter)	Linearized state estimation	ใช้แพร่หลาย, แม่นยำ	ต้องการแบบจำลองแม่นยำ	รวม IMU+Encoder+Camera
	Particle Filter / MCL	ใช้การสุ่มตัวอย่างเพื่อประมาณสถานะ	เหมาะกับ non-linear, multi-modal	ใช้ compute มาก	Monte Carlo Localization

จากตารางจะเห็นได้ว่า AMR อาศัยการผสมผสานอัลกอริทึมที่หลากหลายเพื่อครอบคลุมทุกมิติของการทำงาน ตั้งแต่การรับรู้สิ่งแวดล้อมไปจนถึงการควบคุมการเคลื่อนที่ งานวิจัยนี้เลือกใช้

YOLOv11 สำหรับการตรวจจับวัตถุ, เทคนิค PnP และ GPR สำหรับการระบุตำแหน่ง และใช้ Pure Pursuit ในการติดตามเส้นทาง ทั้งหมดถูกผสานภายใต้สถาปัตยกรรมการควบคุมแบบรวมศูนย์ ซึ่งช่วยให้ได้ทั้งความแม่นยำ ความยืดหยุ่น และต้นทุนต่ำเมื่อเทียบกับระบบที่ใช้เซนเซอร์ราคาแพงเช่น LIDAR

2.2 การระบุตำแหน่งในพื้นที่ในร่ม (Indoor Localization)

การระบุตำแหน่งในพื้นที่ในร่มหมายถึงกระบวนการหาตำแหน่งและทิศทาง (position and orientation) ของวัตถุหรือระบบที่เคลื่อนที่ได้ ภายในสภาพแวดล้อมที่สัญญาณดาวเทียม (GPS/GNSS) ไม่สามารถใช้งานได้หรือมีความไม่เสถียร เช่น ภายในอาคาร คลังสินค้า โรงพยาบาล หรือพื้นที่ใต้ดิน เทคโนโลยีนี้ถือเป็นหัวใจสำคัญของระบบหุ่นยนต์อัตโนมัติ (Autonomous Systems) โดยเฉพาะยานพาหนะภาคพื้นดินอัตโนมัติ (AGV, AMR) ที่ต้องอาศัยข้อมูลตำแหน่งแบบเรียลไทม์สำหรับการควบคุม นำทาง และหลีกเลี่ยงสิ่งกีดขวาง

แนวทางการระบุตำแหน่งในร่มสามารถแบ่งได้เป็น 3 กลุ่มหลัก ได้แก่:

1) Signal-based Localization

ใช้การวัดคุณสมบัติของสัญญาณไร้สายที่มีอยู่ในอาคาร เช่น Wi-Fi, Bluetooth Low Energy (BLE), RFID, Ultra-Wideband (UWB) หรือแม้กระทั่งสัญญาณจากเสาสื่อสาร วิธีการทั่วไปคือการวัดความแรงของสัญญาณ (Received Signal Strength Indicator: RSSI), เวลาเดินทาง (Time of Flight: ToF) หรือมุมของสัญญาณ (Angle of Arrival: AoA) เพื่อประมาณตำแหน่ง

จุดเด่น: ใช้อุปกรณ์พื้นฐานที่ติดตั้งอยู่แล้ว เช่น Wi-Fi access point, RFID reader

ข้อจำกัด: ความแม่นยำต่ำเมื่อมีการสะท้อนสัญญาณ (multipath) และแปรผันตามสภาพแวดล้อม

2) Vision-based Localization

ใช้กล้องเป็นตัวรับรู้ (sensor) หลักในการระบุตำแหน่ง โดยอาศัยการตรวจจับคุณลักษณะจากภาพ (feature detection), การคำนวณเรขาคณิตของกล้อง (camera geometry) หรือการใช้เทคนิค Computer Vision และ AI เช่น การตรวจจับวัตถุ (object detection) และ Visual SLAM (Simultaneous Localization and Mapping)

จุดเด่น: ต้นทุนต่ำ (โดยเฉพาะหากใช้กล้องที่มีอยู่แล้ว เช่น CCTV), สามารถใช้กับ indoor environment ที่ซับซ้อนได้

ข้อจำกัด: ขึ้นอยู่กับสภาพแสง, อาจมีปัญหาเมื่อเกิดการบัง (occlusion)

3) Hybrid / Sensor Fusion Approaches

ผสานข้อมูลจากหลายเซนเซอร์ เช่น การรวม IMU + Encoder + กล้อง (Visual-Inertial Odometry: VIO) หรือการรวม UWB + Camera เพื่อเพิ่มความแม่นยำและลดข้อจำกัดของวิธีการเดี่ยว

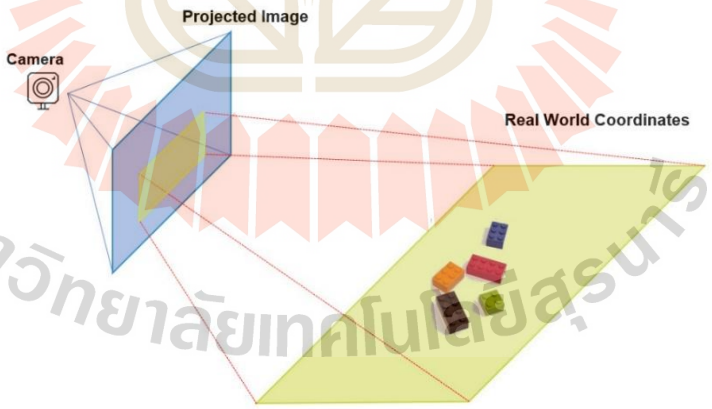
จุดเด่น: ให้ความแม่นยำและเสถียรภาพสูงกว่า single-sensor

ข้อจำกัด: ซับซ้อนกว่า ต้องการการคาลิเบรตและประมวลผลมากขึ้น

งานวิจัยนี้มุ่งเน้นไปที่การระบุตำแหน่งแบบ Vision-based โดยใช้กล้องวงจรปิดร่วมกับอัลกอริทึม Computer Vision และการเรียนรู้ของเครื่อง เนื่องจากมีต้นทุนต่ำ สามารถใช้โครงสร้างพื้นฐานที่มีอยู่แล้ว และเหมาะสมกับการประยุกต์ในสภาพแวดล้อมภายในอาคาร เช่น โรงพยาบาลและคลังสินค้า

2.2.1 พื้นฐานสำหรับการระบุตำแหน่งจากภาพ

การประมาณตำแหน่งของวัตถุจากภาพสองมิติถือเป็นเทคนิคพื้นฐานที่สำคัญในงานด้านการระบุตำแหน่งและการนำทางด้วยวิสัยทัศน์ โดยเฉพาะในกรณีที่กล้องไม่สามารถวัดระยะเชิงลึกได้โดยตรง เช่น กล้อง Surveillance หรือกล้อง RGB ทั่วไป หลักการสำคัญของวิธีการนี้อยู่ที่แบบจำลองเรขาคณิตของกล้อง (camera geometry) และการแปลงพิกัด (coordinate transformation) ระหว่างระบบพิกัดโลก (world frame) และระบบพิกัดกล้อง (camera frame) ดังรูปที่ 2.14



รูปที่ 2.14 Image Transform Projection (Siripong, Nopparut and Jiraphon, 2025)

ในทางคณิตศาสตร์ ความสัมพันธ์ระหว่างจุดในโลกสามมิติ (X, Y, Z) และจุดในภาพสองมิติ (u, v) สามารถอธิบายได้ด้วย pinhole camera model ดังสมการที่ (5):

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \{A\}\{R|T\}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.5)$$

โดยที่

(X, Y, Z) คือ พิกัดของจุดในโลก (world coordinates)

(u, v) คือ พิกัดของจุดบนภาพ (image coordinates)

s คือ สเกลแฟกเตอร์ (scale factor)

A คือ Intrinsic Parameter ของกล้อง

$R|T$ คือ extrinsic parameters ซึ่งประกอบด้วยเมทริกซ์การหมุน R และเวกเตอร์การเลื่อน T ที่ใช้ในการแปลงพิกัดจากระบบโลกมายังระบบกล้อง

แบบจำลองนี้ถือเป็นหัวใจสำคัญของการคำนวณตำแหน่งจากภาพ และเป็นพื้นฐานของเทคนิค Perspective-n-Point (PnP) รวมถึงการคำนวณอื่น ๆ ในงาน Computer Vision ที่เกี่ยวข้องกับการระบุตำแหน่งสามมิติ

2.2.1.1 Perspective-n-Point (PnP)

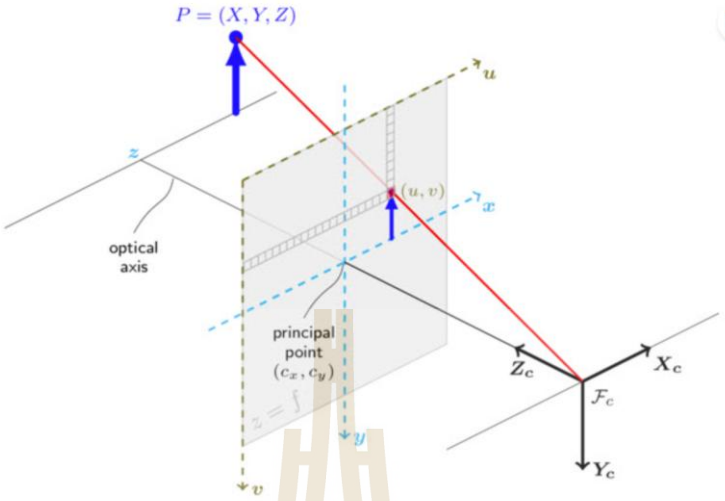
ปัญหา PnP เป็นการประยุกต์ใช้แบบจำลองกล้องรูเข็ม (pinhole camera model) เพื่อคำนวณตำแหน่งและท่าทาง (pose: R, T) ของกล้องหรือของวัตถุ เมื่อทราบการจับคู่ระหว่างจุดในโลก (world points) และจุดที่ปรากฏในภาพ (image points) ดังรูปที่ 2.15 โดยโจทย์สมการของ PnP สามารถกำหนดได้ดังสมการที่ (6):

มีจุดในโลก $P_i = (X_i, Y_i, Z_i)$ โดย i =1, 2, ..., n

มีจุดที่สอดคล้องกันบนภาพ $p_i = (u_i, v_i)$

ต้องการหาค่า การหมุน (R) และ การเลื่อน (T) ที่ทำให้ความสัมพันธ์ต่อไปนี้เป็นจริง:

$$s\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \{A\}\{R|T\}\begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}, \; i = 1, 2, ..., n \quad (2.6)$$



รูปที่ 2.15 ความสัมพันธ์ระหว่างจุดในภาพกับเทียบกับจุด World Coordinate (De Jong, Gustavo, John and Joseph, 2021)

ในการประมาณตำแหน่งสามมิติจากภาพถ่าย จำเป็นต้องอาศัยพารามิเตอร์ของกล้องสองกลุ่มหลัก ได้แก่ Intrinsic Parameters และ Extrinsic Parameters

1) พารามิเตอร์ภายในกล้อง (Intrinsic Parameters)

Intrinsic คือค่าที่บ่งบอกคุณสมบัติทางเรขาคณิตของกล้องเอง เช่น ความยาวโฟกัส (focal length), ตำแหน่งจุดศูนย์กลางภาพ (principal point), อัตราส่วนพิกเซล (pixel scaling factors) และการบิดเบือนของเลนส์ (lens distortion) โดยสามารถแทนในรูปเมทริกซ์ A ขนาด 3×3 ดังสมการที่ (2.7):

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

โดยที่

f_x, f_y คือ ความยาวโฟกัสในหน่วยพิกเซลในแนว x และ y

c_x, c_y คือ จุดตัดของแกนภาพ (image center หรือ principal point)
ซึ่งค่าเหล่านี้ได้มาจากการ Calibrate กล้อง ด้วยวิธีการ เช่น Chessboard Calibration

2) พารามิเตอร์ภายนอกกล้อง (Extrinsic Parameters)

Extrinsic คือค่าที่บอกตำแหน่งและทิศทางของกล้องในระบบพิกัดโลก โดยนิยามดังสมการที่ (2.8):

$$P_c = RP_w + T \tag{2.8}$$

โดยที่:

$P_w = [X, Y, Z]^T$ คือ จุดในพิกัดโลก

$P_c = [x, y, z]^T$ คือ จุดเดียวกันในระบบกล้อง

R คือ เมทริกซ์การหมุน 3x3

T คือ เวกเตอร์การเลื่อน 3x1

ตำแหน่งกล้องในพิกัดโลกสามารถหาได้จากสมการที่ (2.9):

$$C_w = -R^T T \tag{2.9}$$

การรวมกันของพารามิเตอร์ทั้งสองสามารถเขียนในรูปเมทริกซ์ $[R|T]$ ขนาด 3x4 ซึ่งใช้ในการแปลงพิกัดของจุด 3D ในโลกเข้าสู่ระบบของกล้อง

3) Scale Factor

ค่าของ scale factor (s) ในสมการ pinhole camera model ไม่ได้เป็นค่าคงที่ แต่ขึ้นอยู่กับระยะลึก (depth) ของแต่ละจุดในโลกจริง ทำให้ในการแก้ปัญหา PnP แต่ละจุดจะมีค่า s_i ที่แตกต่างกัน การมีจำนวนจุดมากขึ้นจึงช่วยให้สามารถแก้สมการเชิงเส้นที่เกิดจากจุดหลายคู่ (correspondences) ได้อย่างแม่นยำยิ่งขึ้น และลดความไม่แน่นอนที่เกิดจาก noise ของจุดใดจุดหนึ่ง

2.2.1.2 การประมาณค่าด้วยพหุนามผิวสามมิติ (3D 2nd Order Polynomial Surface Regression)

แม้ว่าการใช้ PnP จะสามารถแก้หาตำแหน่งและท่าทางของกล้องหรือวัตถุได้จากความสัมพันธ์ระหว่างจุดในโลกและจุดในภาพ แต่ในการใช้งานจริงยังคงเกิดความคลาดเคลื่อนจากปัจจัยหลายประการ เช่น ความไม่แม่นยำของการจับคู่จุด (feature matching), ความผิดพลาดในการคาลิเบรตกล้อง และโดยเฉพาะอย่างยิ่งความไม่สม่ำเสมอของ scale factor (s_i) ซึ่งมีค่าต่างกันไปในแต่ละจุดที่มีระยะลึก (depth) แตกต่างกัน

เพื่อแก้ปัญหานี้ งานวิจัยได้ใช้ การประมาณค่าด้วยพหุนามผิวสามมิติ (3D Polynomial Surface Regression) ลำดับสอง (2nd Order) เพื่อสร้างแบบจำลองฟังก์ชันที่อธิบายความสัมพันธ์เชิงไม่เชิงเส้นระหว่างพิกัดภาพ (u, v) และพิกัดโลก (X, Y, Z) โดยจะมีสมการที่ (2.10) ดังนี้

$$s(u, v) = a_0 + a_1 u + a_2 v + a_3 v^2 + a_4 vu + a_5 u^2 \qquad (2.10)$$

โดยที่:

$s(u, v)$ คือ Scale factor ของจุด

$a_0, \ldots, a_5$ คือ ค่าสัมประสิทธิ์ที่ได้จากการประมาณค่า (regression fitting)

การเลือกใช้ Polynomial Surface Regression ลำดับสอง (2nd Order) มีเหตุผลเชิงทฤษฎีและการปฏิบัติที่สอดคล้องกัน กล่าวคือ กล้องจริงไม่ได้เป็นกล้องรูเข็มที่สมบูรณ์แบบ แต่จะมี ความบิดเบือนของเลนส์ (lens distortion) ทั้งในเชิงรัศมี (radial distortion) และเชิงเส้น (tangential distortion) ดังรูปที่ 2.16 ซึ่งส่วนใหญ่สามารถอธิบายได้ในเชิง ควอดราติก (quadratic function) ของพิกัดภาพ ดังนั้น การใช้ Polynomial Regression ลำดับสองจึงถือว่า เหมาะสมที่สุดสำหรับการแก้ไขความไม่เชิงเส้นของ scale factor และการประมาณตำแหน่ง เนื่องจาก:

1) รองรับ Distortion เชิง Quadratic

ความผิดเพี้ยนหลักจากกล้อง เช่น barrel distortion และ pincushion distortion มักมีลักษณะเป็นฟังก์ชันกำลังสองของรัศมี (r^2 term)

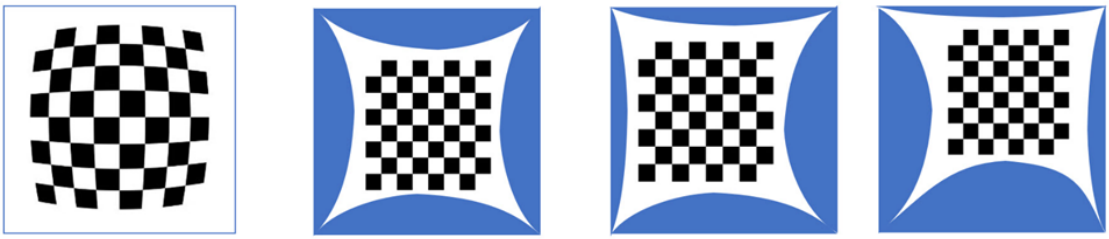
Polynomial ลำดับสองสามารถอธิบายปรากฏการณ์นี้ได้ดี โดยไม่จำเป็นต้องใช้ลำดับสูงกว่า

2) ลด Overfitting และ Noise Sensitivity

หากใช้ Polynomial ลำดับสูง อาจเพิ่มความซับซ้อนของโมเดลโดยไม่จำเป็น และทำให้เกิด overfitting ต่อ noise การใช้ลำดับสองจึงให้สมดุลระหว่างความแม่นยำและความเสถียร

3) ประสิทธิภาพเชิงคำนวณ

ลำดับสองมีจำนวนพารามิเตอร์ที่ไม่มาก ทำให้สามารถคำนวณได้รวดเร็ว เหมาะสำหรับงานที่ต้องการ real-time เช่น AGV และ AMR



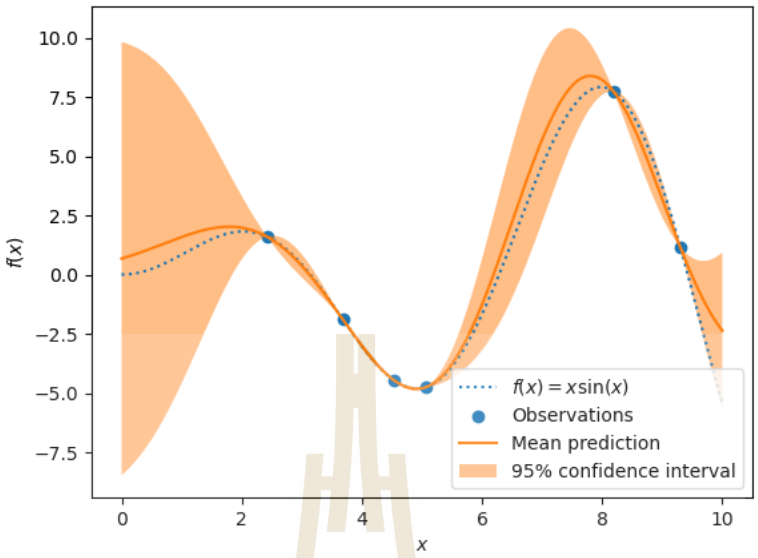
รูปที่ 2.16 ภาพที่เกิดขึ้นจากความบิดเบือนของเลนส์ (Dimanov, 2019)

กล่าวโดยสรุป การใช้ Polynomial Surface Regression ลำดับสอง สอดคล้องกับ ลักษณะความโค้งเชิงเรขาคณิต (quadratic distortion) ของระบบกล้อง และยังให้ประสิทธิภาพทั้งด้านความแม่นยำและความเร็วโดยไม่ก่อให้เกิดความซับซ้อนเกินจำเป็น

2.2.1.3 การปรับปรุงค่าด้วย Gaussian Process Regression (GPR)

แม้ว่าการประมาณค่าตำแหน่งด้วย Polynomial Surface Regression ลำดับสองจะสามารถอธิบายความไม่เชิงเส้นหลักที่เกิดจากการบิดเบือนของกล้องและความแตกต่างของ scale factor ได้ในระดับหนึ่ง แต่ในสภาพแวดล้อมจริง ข้อมูลจากกล้องยังคงมีความไม่แน่นอน (uncertainty) และมี noise จากหลายสาเหตุ เช่น แสงที่เปลี่ยนแปลง การบัง (occlusion) และคุณภาพของ feature detection

เพื่อเพิ่มความแม่นยำและความทนทานต่อ noise งานวิจัยนี้ได้นำ GPR มาใช้สำหรับการปรับปรุงค่าการประมาณตำแหน่ง โดย GPR เป็นวิธีการเชิงสถิติที่สามารถประมาณค่าฟังก์ชันจากข้อมูลตัวอย่างได้ โดยไม่จำเป็นต้องกำหนดรูปฟังก์ชันตายตัวล่วงหน้า แต่จะอาศัยการคาดคะเนในเชิง distribution ดังรูปที่ 2.17 ฟังก์ชันทุก ๆ จุดข้อมูลที่สังเกตได้สามารถอธิบายด้วยการแจกแจงร่วมแบบ Gaussian (multivariate Gaussian distribution)



รูปที่ 2.17 แสดงการประมาณค่าและช่วงด้วย GPR (scikit-learn developers, 2025)

การให้ชุดข้อมูลฝึกเป็นจะเป็นไปดังสมการที่ (2.11)

$$\partial = \{(x_i, y_i)\}_{i=1}^{n} \tag{2.11}$$

โดยที่:

x_i คือ input พิกัดภาพ (u, v)

y_i คือ output ตำแหน่ง (X, Y, Z) ของ AGV

สมมติว่า $y(x)$ มาจากฟังก์ชันที่แจกแจงแบบ Gaussian Process (GP) ดังสมการที่ (2.12):

$$f(x) \sim GP(m(x), k(x, x')) \tag{2.12}$$

โดยที่

$x \in R^d$ คือ ค่าพิกัดอินพุต

$f(x)$ คือ ฟังก์ชันเป้าหมาย

$m(x)$ คือ ฟังก์ชันค่าเฉลี่ย มักกำหนดให้เป็นศูนย์ (Zero-mean)

Kernel Function คือหัวใจของ GPR ซึ่งกำหนดรูปแบบความสัมพันธ์ระหว่างจุดข้อมูล โดยทั่วไปเลือกใช้ Radial Basis Function (RBF) / Squared Exponential Kernel ซึ่งจะมีฟังก์ชันดังสมการที่ (2.13)

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2l^2}\|x - x'\|^2\right) \quad (2.13)$$

โดยที่

l คือ Length-scale ควบคุม smoothness ของฟังก์ชัน

σ_f^2 คือ Signal variance ควบคุมความแปรปรวนของฟังก์ชัน

สมการสำหรับ Gaussian Process Regression เมื่อมีชุดข้อมูลฝึก ∂ และต้องการพยากรณ์ค่าที่จุดใหม่ x_* จะได้การแจกแจงร่วมดังสมการที่ (2.14)

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \eta\left(0, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X, x_*) \\ K(x_*, X) & K(x_*, x_*) \end{bmatrix}\right) \quad (2.14)$$

โดยที่

$K(X,X)$ คือ เมทริกซ์ Kernel ระหว่างข้อมูลเทรนทั้งหมด

σ_n^2 คือ ความแปรปรวนของ noise

x_* คือ ค่าที่ต้องการพยากรณ์

จากนั้น ค่าทำนาย ที่จุดใหม่ x_* จะมีการแจกแจงปกติดังสมการที่ (2.15), (2.16) และ (2.17)

$$y_* \sim \eta(\mu(x_*), \sigma^2(x_*)) \quad (2.15)$$

ซึ่ง

$$\mu(x_*) = K(x_*, X)(K(X,X) + \sigma_n^2 I)^{-1} y \quad (2.16)$$

$$\sigma^2(x_*) = K(x_*, x_*) - K(x_*, X)(K(X,X) + \sigma_n^2 I)^{-1} K(X, x_*) \quad (2.17)$$

โดยที่

$\mu(x_*)$ คือ ค่าที่ทำนายได้ (*mean prediction)*

$\sigma^2(x_*)$ คือ ค่าความไม่แน่นอน (*variance)* ของการทำนาย

2.2.2 การระบุตำแหน่งภายในอาคารด้วยกล้อง (Indoor Localization using Camera-based System)

การระบุตำแหน่งภายในอาคารโดยใช้กล้อง (Camera-based Indoor Localization) เป็นแนวทางที่ผสมผสานระหว่างวิธีการทางวิศวกรรมการมองเห็นของเครื่อง (Computer Vision) กับการคำนวณเรขาคณิตของกล้อง (Camera Geometry) ซึ่งมีข้อเปรียบเทียบดังตารางที่ 2.6 เพื่อประมาณค่าตำแหน่งของวัตถุหรือยานพาหนะในระบบพิกัดโลก โดยเฉพาะในบริบทของ AGV หรือ AMR ที่ต้องเคลื่อนที่อย่างแม่นยำในพื้นที่จำกัด

ตารางที่ 2.6 ข้อเปรียบเทียบประเภทกล้องกับการใช้ในงาน Localization

ประเภทกล้อง	ลักษณะ	ความสามารถ	ข้อจำกัด
Monocular RGB	กล้องทั่วไป	ราคาถูก, ใช้กับ *Deep Learning* ได้ดี	ไม่มีข้อมูลความลึก
Stereo Camera	มีกล้อง 2 ตัว	มีข้อมูลเชิงลึก (*Depth)*	ต้องการการ *Calibrate* ที่ซับซ้อน
RGB-D Camera	มีเซนเซอร์ความลึก	ใช้งานง่ายใน *SLAM*	ระยะตรวจจับจำกัด (*~3* เมตร)
CCTV / Surveillance	กล้องติดตั้งบนเพดาน/ผนัง	ครอบคลุมพื้นที่กว้าง, ใช้ตรวจจับ *AGV* จากภายนอก	ต้องแปลงภาพกลับไปยังพิกัดโลก

โดยสรุป การระบุตำแหน่งภายในอาคารด้วยกล้อง (Camera-based Indoor Localization) เป็นแนวทางที่มีศักยภาพสูง เนื่องจากสามารถใช้โครงสร้างพื้นฐานที่มีอยู่แล้ว เช่น กล้องวงจรปิด (Surveillance Camera) มาประยุกต์ร่วมกับเทคนิค Computer Vision และ AI เพื่อลดต้นทุนของระบบหุ่นยนต์เคลื่อนที่อัตโนมัติ ในงานวิจัยนี้ ผู้วิจัยได้เลือกใช้กล้องวงจรปิดร่วมกับอัลกอริทึมการตรวจจับวัตถุ YOLO, เทคนิค Perspective-n-Point (PnP), การปรับค่าความลึกด้วย Polynomial Regression, และการแก้ไขความคลาดเคลื่อนด้วย Gaussian Process Regression (GPR) พร้อมทั้งผสานข้อมูลจากหลายกล้อง (Multi-Camera Fusion) เพื่อเพิ่มความแม่นยำและความทนทานต่อสภาพแวดล้อมจริง แนวทางนี้จึงช่วยให้สามารถสร้างระบบนำทางที่มีความยืดหยุ่น ต้นทุนต่ำ และสอดคล้องกับเงื่อนไขการใช้งานจริงในอาคาร

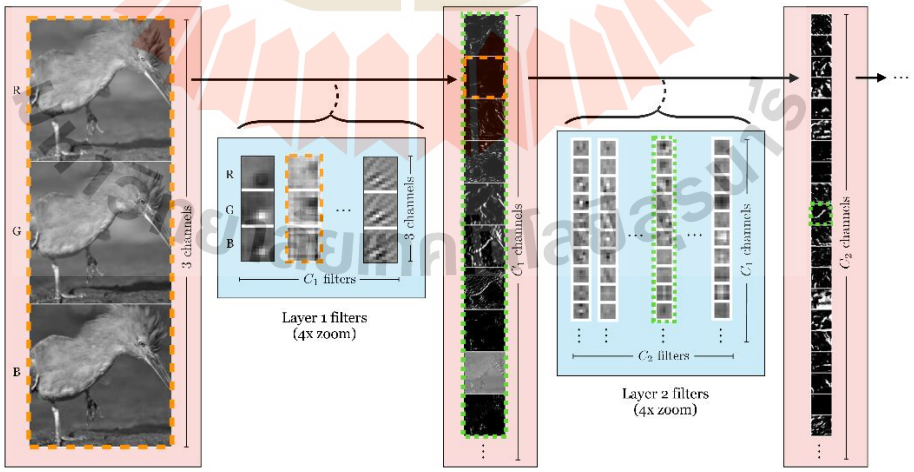
2.3 การประมวลผลภาพและการตรวจจับวัตถุโดยใช้ปัญญาประดิษฐ์ (AI for Visual Perception and Object Detection)

ในระบบยานพาหนะอัตโนมัติภายในอาคาร (Indoor Autonomous Ground Vehicles) การรับรู้สภาพแวดล้อม (perception) ถือเป็นพื้นฐานสำคัญสำหรับการนำทาง (navigation), การหลีกเลี่ยงสิ่งกีดขวาง (obstacle avoidance) และการโต้ตอบกับสิ่งแวดล้อม (environment interaction) อย่างมีประสิทธิภาพ

ในหัวข้อนี้จะอธิบายทั้งพื้นฐานและการพัฒนาล่าสุดของ AI สำหรับการประมวลผลภาพ โดยเริ่มจากพื้นฐานการทำงานของโครงข่ายประสาทเทียม (CNN) ไปจนถึงสถาปัตยกรรมที่เหมาะสมสำหรับงานตรวจจับวัตถุแบบเวลาจริง (YOLO) และการประยุกต์ใช้ในระบบกล้องวงจรปิด (Surveillance Camera) เพื่อสร้างโมดูลการรับรู้สภาพแวดล้อมสำหรับ AGV.

2.3.1 พื้นฐานการประมวลผลภาพด้วยปัญญาประดิษฐ์

การประมวลผลภาพด้วยปัญญาประดิษฐ์ (AI-based Computer Vision) เป็นการพัฒนาต่อยอดจากวิธีการดั้งเดิมที่อาศัยการ ดึงคุณลักษณะ (feature extraction) แบบกำหนดเอง เช่น SIFT (Scale-Invariant Feature Transform), SURF (Speeded-Up Robust Features), HOG (Histogram of Oriented Gradients) และ ORB ซึ่งต้องอาศัยการออกแบบ feature descriptor โดยผู้เชี่ยวชาญ อย่างไรก็ตาม วิธีการเหล่านี้มักไม่สามารถรับมือกับความซับซ้อนของสภาพแวดล้อมจริง เช่น การเปลี่ยนแปลงของแสง เงา หรือการบังวัตถุ (occlusion) ได้อย่างมีประสิทธิภาพ



รูปที่ 2.18 Feature Learning จาก CNN (Alex, Ilya and Geoffrey, 2012)

ด้วยการเกิดขึ้นของ ปัญญาประดิษฐ์เชิงลึก (Deep Learning) โดยเฉพาะโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Networks: CNN) ทำให้ระบบสามารถเรียนรู้คุณลักษณะโดยอัตโนมัติ (automatic feature learning) จากข้อมูลภาพจำนวนมาก โดยไม่ต้องพึ่งการออกแบบคุณลักษณะแบบตายตัวเหมือนในอดีต ดังรูปที่ 2.18 ส่งผลให้ AI-based Computer Vision กลายเป็นเทคโนโลยีหลักในงานนำทางหุ่นยนต์และการระบุตำแหน่งในปัจจุบัน

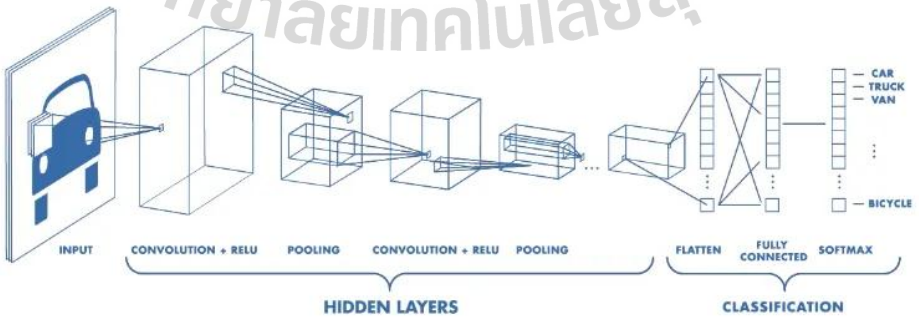
ในเชิงการทำงาน โครงข่าย CNN ใช้การคอนโวลูชัน (convolution) เพื่อดึงลักษณะเชิงพื้นที่ของภาพ (spatial features) จากระดับต่ำ (low-level features เช่น ขอบและเส้น) ไปจนถึงระดับสูง (high-level features เช่น รูปร่างและวัตถุ) สำหรับระบบยานพาหนะอัตโนมัติภายในอาคาร (Indoor AGV/AMR) AI-based Computer Vision ทำหน้าที่เสมือน “ดวงตา” ของหุ่นยนต์ โดยทำให้หุ่นยนต์สามารถ:

1) ตรวจจับวัตถุและสิ่งกีดขวาง (object detection)
2) ระบุตำแหน่งและการเคลื่อนไหวของวัตถุ (object tracking and localization)
3) ทำความเข้าใจโครงสร้างของสภาพแวดล้อม (scene understanding)
4) ส่งข้อมูลที่ประมวลผลแล้วไปยังโมดูลควบคุมการตัดสินใจ (decision-making)

ดังนั้น การประมวลผลภาพด้วยปัญญาประดิษฐ์จึงเป็นรากฐานของงานวิจัยที่เกี่ยวข้องกับการนำทางอัตโนมัติ เนื่องจากช่วยให้ AGV/AMR มีความสามารถในการรับรู้และปรับตัว (perception and adaptation) ได้อย่างอัจฉริยะและสอดคล้องกับสภาพแวดล้อมจริง

2.3.2 โครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN)

โครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) ดังรูปที่ 2.19 เป็นสถาปัตยกรรมสำคัญในงานด้านการประมวลผลภาพ โดยมีโครงสร้างเฉพาะที่ช่วยดึงคุณลักษณะเชิงพื้นที่ (spatial features) ของภาพออกมาอย่างเป็นลำดับชั้น ตั้งแต่ระดับต่ำ (low-level features) เช่น เส้นขอบและสี ไปจนถึงระดับสูง (high-level features) เช่น รูปร่างและวัตถุเป้าหมาย



รูปที่ 2.19 สถาปัตยกรรมแบบ Convolution (M. S., 2021)

โครงสร้างพื้นฐานของ CNN ประกอบด้วย:

1) Convolutional Layer
2) Activation Function
3) Pooling Layer
4) Fully Connected Layer

1) Convolutional Layer:

ใช้ Kernel/Filter เลื่อนผ่านภาพเพื่อดึงคุณลักษณะเฉพาะดังรูปที่ 2.20 เช่น ขอบ เส้น รูปร่างพื้นฐาน โดย สมการที่ (18) ของการคอนโวลูชันแบบ 2 มิติจะเป็น

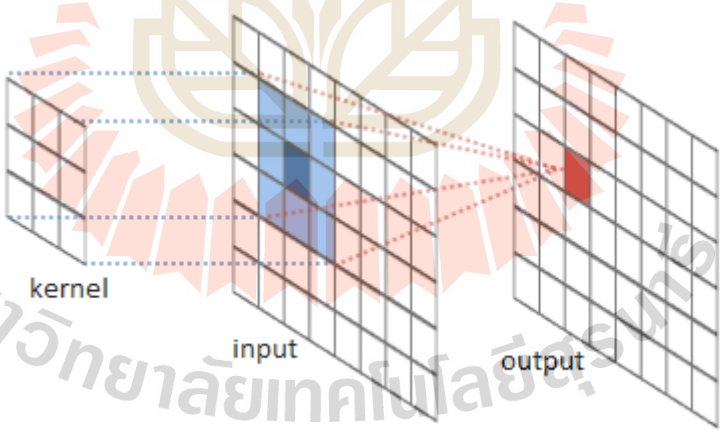
$$S(i,j) = (X * K)(i,j) = \sum_m \sum_n X(i+m, j+n)K(m,n) \qquad (2.18)$$

โดยที่

X คือ input image

K คือ kernel (ฟิลเตอร์)

S คือ feature map ที่ได้หลังคอนโวลูชัน



รูปที่ 2.20 Convolutional Layer (Qi et al., 2025)

2) Activation Function (ReLU)

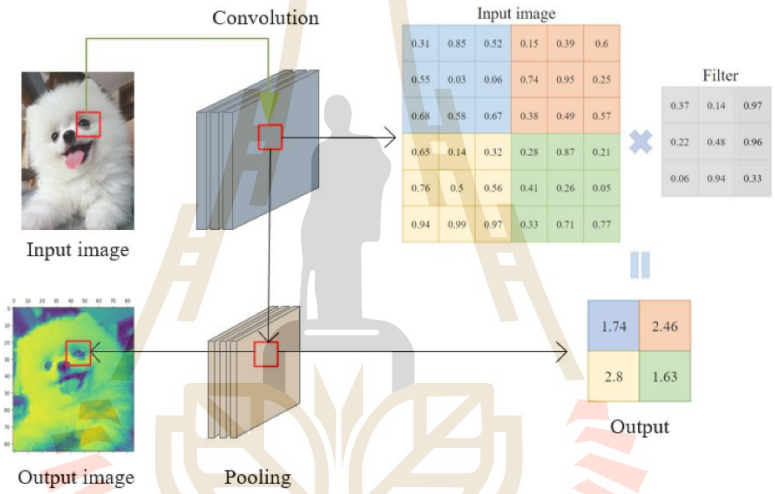
สมการที่ (2.19) จะเป็นฟังก์ชันที่ช่วยให้โมเดลสามารถเรียนรู้แบบไม่เชิงเส้น (non-linearity)

$$f(x) = \max(0, x) \tag{2.19}$$

3) Pooling Layer

Pooling Layer จากสมการที่ (20) จะอยู่ต่อจาก Convolutional Layer ใช้ลดขนาดของ feature map เพื่อควบคุมจำนวนพารามิเตอร์และลด overfitting โดย Max Pooling ดังรูปที่ 2.21:

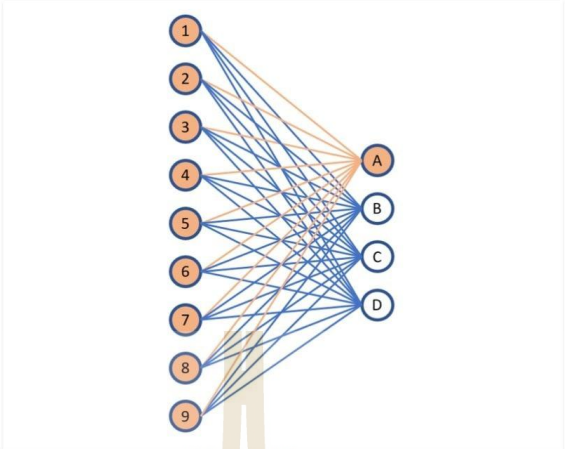
$$P(i,j) = \max_{(m,n)\in window} \{S_{(2i+m,2j+n)}\} \tag{2.20}$$



รูปที่ 2.21 Pooling Layer หลังกระบวนการ Convolutional Layer (Lei and Zhonglin, 2024)

4) Fully Connected Layer

เชื่อมโยงคุณลักษณะที่เรียนรู้แล้วไปยังหน่วย output สำหรับการจำแนกประเภทดังรูปที่ 2.22



รูปที่ 2.22 Fully Connected Layer สำหรับจำแนกประเภท (Adib, Jalal and Adil, 2024)

5) Output Layer

ใช้ฟังก์ชันเช่น Softmax หรือ Sigmoid เพื่อแปลงค่าเป็นความน่าจะเป็นสำหรับการจำแนกประเภท (classification) หรือการตรวจจับวัตถุ (object detection) ดังสมการที่ (2.21)

$$P(y_i) = \frac{e^{z_i}}{\sum_j e^{z_i}} \tag{2.21}$$

CNN ถูกนำมาใช้เป็นแกนหลักของสถาปัตยกรรม YOLOv11 สำหรับการตรวจจับวัตถุ (Object Detection) ซึ่งโมเดล CNN จะรับข้อมูลภาพจากกล้องวงจรปิด แล้วประมวลผลผ่าน convolution, pooling และ classification layers เพื่อระบุและแยกประเภทวัตถุที่อยู่ในภาพ

2.3.3 สถาปัตยกรรม YOLO และการประยุกต์ใช้งาน

YOLO (You Only Look Once) เป็นสถาปัตยกรรมสำหรับการตรวจจับวัตถุ (Object Detection) ที่พัฒนาขึ้นโดย Joseph, Santosh, Ross and Ali, (2016) มีแนวคิดสำคัญคือการตรวจจับวัตถุในขั้นตอนเดียว (single-stage detector) โดยมองปัญหาการตรวจจับเป็นงาน การถดถอยเชิงลึก (deep regression problem) จากภาพอินพุตไปยัง bounding box และ class ของวัตถุ โดยไม่ต้องแยกขั้นตอนการสร้าง Region Proposal เหมือนสถาปัตยกรรมแบบดั้งเดิม เช่น R-CNN หรือ Faster R-CNN โดยจะมีหลักการทำงานดังนี้

1) การแบ่งภาพ (Grid Division)

ภาพอินพุตถูกแบ่งออกเป็นกริด SxS (เช่น 7x7) แต่ละกริดจะรับผิดชอบการตรวจจับวัตถุที่ศูนย์กลางอยู่ภายในกริดนั้นดังรูปที่ 2.23

2) การทำนาย *Bounding Boxes* และ *Confidence*

แต่ละกริดจะทำนาย bounding box B จำนวนหนึ่ง โดยประกอบด้วย:

2.1) พิกัด (x, y, w, h)

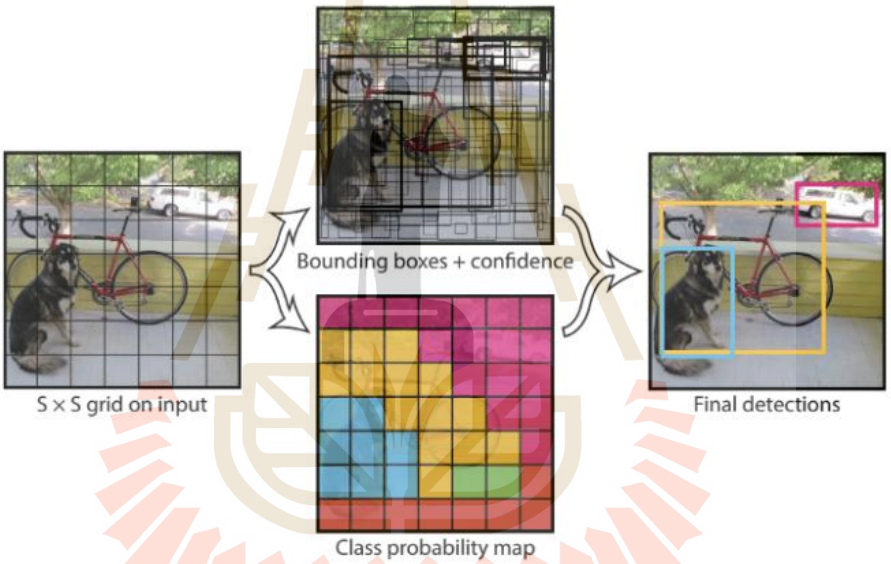
2.2) ค่าความมั่นใจ (confidence score) ว่ามีวัตถุอยู่ในกริด

3) การทำนาย *Class Probabilities*

แต่ละกริดยังทำนาย class probability ของวัตถุที่ตรวจจับได้

4) การรวมผลลัพธ์

Output สุดท้ายได้จากการรวม bounding boxes และ class prediction ผ่านขั้นตอน Non-Maximum Suppression (NMS) เพื่อตัดกล่องซ้อนทับและเหลือกล่องที่มั่นใจที่สุด



รูปที่ 2.23 หลักการตรวจสอบภาพของ Yolo (Yasutoshi, Masaya and Hitoshi, 2022)

หนึ่งในหัวใจหลักของ YOLO คือ การคำนวณค่าคะแนนความมั่นใจ (confidence score) เพื่อประเมินว่า Bounding Box ที่ทำนายนั้นมี ความน่าเชื่อถือ และ ตรงกับวัตถุจริง มากน้อยเพียงใดดังสมการที่ (2.22)

$$C = P_o x IoU \tag{2.22}$$

โดยที่:

C คือ คะแนนความมั่นใจ (confidence score)

P_o คือ ความน่าจะเป็นที่มีวัตถุอยู่ในกริดเซลล์นั้น ๆ

IoU คือ ค่าความซ้อนทับระหว่างกล่องที่ทำนาย (Predicted Box) และกล่องจริง (Ground Truth Box) เรียกว่า Intersection over Union

YOLO จึงเป็นสถาปัตยกรรมที่เหมาะสมที่สุดในงานนี้ เพราะสามารถตรวจจับ AGV และสิ่งกีดขวางได้แบบ real-time แม้ในสภาพแวดล้อมจริงที่ซับซ้อน และเมื่อผสานกับ PnP, Polynomial Regression และ GPR ทำให้ได้ระบบ Localization ที่มีความแม่นยำและทนทาน

2.3.4 ข้อเปรียบเทียบของอัลกอริทึม Object Detection

สถาปัตยกรรมสำหรับการตรวจจับวัตถุ (Object Detection Architectures) สามารถแบ่งออกได้เป็นสองกลุ่มหลัก ได้แก่ กลุ่ม Two-Stage Detectors ซึ่งมีตัวแทนสำคัญ คือ R-CNN Family (R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN) และกลุ่ม Single-Stage Detectors ซึ่งมีตัวแทนสำคัญคือ YOLO Family ความแตกต่างระหว่างสองแนวทางนี้อยู่ที่จำนวนขั้นตอนการประมวลผล ซึ่งส่งผลโดยตรงต่อความเร็ว ความแม่นยำ และความเหมาะสมต่อการใช้งานจริงดังข้อมูลตารางที่ 2.7

ตารางที่ 2.7 ข้อเปรียบเทียบอัลกอริทึม Object Detection

คุณสมบัติ	ตระกูล R-CNN	ตระกูล YOLO
สถาปัตยกรรม (Architecture)	Two-Stage Detector ขั้นแรกสร้าง Region Proposal ขั้นสองทำการ Classification และ Bounding Box Regression	Single-Stage Detector ใช้ CNN เดียวแบ่งภาพเป็นกริด แล้วทำนาย Bounding Box และ Class พร้อมกัน
ความเร็ว (Speed)	ค่อนข้างช้า (R-CNN เดิม ~ 0.05 FPS, Faster R-CNN ~ 5–7 FPS) ไม่เหมาะกับงาน Real-time	เร็วมาก (YOLO สามารถทำได้ > 30 FPS) รองรับงาน Real-time
ความแม่นยำ (Accuracy)	แม่นยำสูง โดยเฉพาะในงานที่ต้องการ precision เช่น การตรวจจับในภาพที่ซับซ้อน	ความแม่นยำดี แต่โดยทั่วไปต่ำกว่า R-CNN เล็กน้อย โดยรุ่นใหม่ (YOLOv7–YOLOv11) พัฒนามากขึ้นใกล้เคียงหรือดีกว่า

ตารางที่ 2.7 ข้อเปรียบเทียบอัลกอริทึม Object Detection (ต่อ)

คุณสมบัติ	ตระกูล R-CNN	ตระกูล YOLO
การจัดการวัตถุขนาดเล็ก (Small Object Detection)	ดีกว่า YOLO โดย R-CNN สามารถโฟกัสได้ด้วย Region Proposal Network (RPN)	รุ่นแรก ๆ ของ YOLO ตรวจจับวัตถุเล็กได้ไม่ดี แต่ รุ่นใหม่ (YOLOv5–YOLOv11) ปรับ backbone + anchor-free ให้ดีขึ้น
การฝึกสอน (Training Complexity)	ซับซ้อน ต้องฝึกหลายโมดูล (R-CNN ดั้งเดิมใช้ SVM + CNN + Bounding Box Regressor) และต้องใช้เวลามาก	ฝึก end-to-end ได้ง่ายกว่า ใช้เพียง CNN ตัวเดียว
การใช้งาน (Use Cases)	งานที่เน้นความแม่นยำสูง เช่น การแพทย์ (Medical Imaging), Autonomous Driving (ระดับ Research)	งานที่เน้น real-time เช่น หุ่นยนต์ (Robotics), การเฝ้าระวัง (Surveillance), AGV/AMR Navigation
จุดเด่น (Strengths)	ความแม่นยำสูง, จัดการกับวัตถุซับซ้อนได้ดี	ความเร็วสูง, เรียบง่าย, เหมาะกับ real-time
จุดด้อย (Weaknesses)	ช้า, ใช้ทรัพยากรสูง	แม่นยำต่ำกว่านิดหน่อย (โดยเฉพาะวัตถุเล็กในรุ่นเก่า)

2.3.5 การใช้ AI กับกล้องวงจรปิด

ในระบบการนำทางของยานพาหนะอัตโนมัติภายในอาคาร (Indoor Autonomous Navigation) กล้องวงจรปิด (Surveillance) ถูกนำมาประยุกต์ใช้อย่างแพร่หลายเพื่อทำหน้าที่เป็น “ดวงตาภายนอก” ที่ช่วยรับรู้และตรวจสอบตำแหน่งของ AGV หรือวัตถุต่าง ๆ ในพื้นที่ โดยไม่ต้องพึ่งพาเซนเซอร์ภายในตัวรถโดยตรง เมื่อผนวกกับ ปัญญาประดิษฐ์ด้านการประมวลผลภาพ (AI-based Computer Vision) ด้วย Yolo กล้องวงจรปิดสามารถตรวจจับวัตถุแบบเรียลไทม์ และประมวลผลเพื่อหาตำแหน่งเชิงพิกัดในระบบพิกัดโลก (Global Coordinate Frame) ได้ การใช้กล้องวงจรปิดร่วมกับ AI เพื่อการตรวจจับและระบุตำแหน่งในงานวิจัยนี้มีขั้นตอนหลักดังนี้:

1) การจับภาพ: กล้องวงจรปิดจะทำหน้าที่จับภาพของพื้นที่ที่ติดตั้ง โดยมีการปรับมุมมอง (FOV) ให้ครอบคลุมเส้นทางของ AGV

2) การประมวลผลด้วย AI (YOLO): YOLOv11 ทำหน้าที่ตรวจจับวัตถุ (เช่น AGV และสิ่งกีดขวาง) โดยสร้าง bounding box รอบวัตถุ

3) การแปลงตำแหน่งภาพเป็นตำแหน่งโลก: ตำแหน่งพิกัดของ bounding box จะถูกประมวลผลต่อด้วย Perspective-n-Point (PnP) เพื่อหาค่าพิกัดในโลกจริง โดยใช้ Intrinsic & Extrinsic Parameters ของกล้องแต่ละตัว

4) การรวมข้อมูลจากหลายกล้อง: หากมีหลายกล้องครอบคลุมวัตถุเดียวกัน ระบบสามารถใช้ Multi-view Triangulation หรือ Sensor Fusion เพื่อเพิ่มความแม่นยำ

5) การส่งข้อมูลเข้าสู่ระบบควบคุม (Control Loop): พิกัดที่ได้จะถูกส่งผ่านเครือข่าย MQTT ไปยังตัวควบคุมกลาง เพื่อนำไปใช้วางแผนเส้นทาง (Path Planning) และควบคุมการเคลื่อนที่

การหาจุดศูนย์กลางจาก Bounding Box เพื่อใช้ระบุตำแหน่งในแผนที่ดังรูป 2.24 โดยที่ Bounding Box ที่ได้จาก YOLO มีพิกัดดังนี้:

x = จุดพิกัดมุมซ้ายบนตามแนวแกน x (ในหน่วยพิกเซล)

y = จุดพิกัดมุมซ้ายบนตามแนวแกน y (ในหน่วยพิกเซล)

w = ความกว้างของกรอบตรวจจับ (width)

h = ความสูงของกรอบตรวจจับ (height)

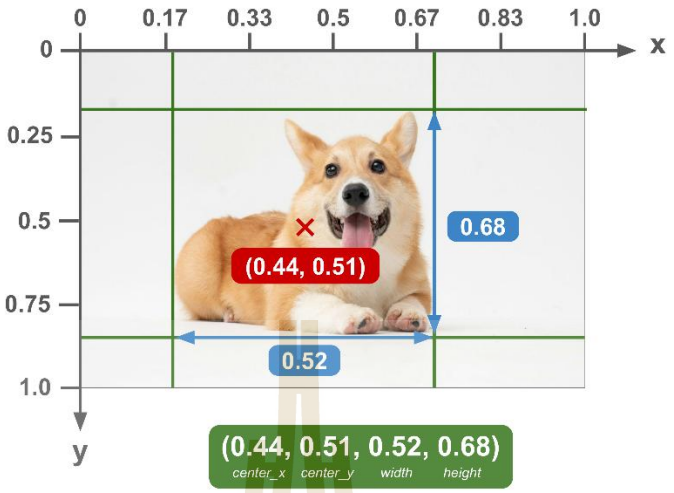
ค่าที่ได้ดังสมการที่ (2.23) คือ พิกัดจุดกึ่งกลางของ Bounding Box ในหน่วยพิกเซลในภาพ

$$(x_c, y_c) = \left(x + \frac{w}{2}, y + \frac{h}{2}\right) \tag{2.23}$$

โดยที่

x_c คือ พิกัดศูนย์กลางแกน x

y_c คือ พิกัดศูนย์กลางแกน y



รูปที่ 2.24 การหาตำแหน่ง Center จาก Bounding Box (DragonEye AI, 2025)

2.4 การหลอมรวมข้อมูลจากเซนเซอร์ (Sensor Fusion)

ในระบบการนำทางของยานพาหนะอัตโนมัติภายในอาคาร ความแม่นยำในการระบุตำแหน่ง (localization) จำเป็นต้องอาศัยข้อมูลจากหลายแหล่งเซนเซอร์เพื่อรักษาเสถียรภาพ เนื่องจากแต่ละเซนเซอร์มีข้อจำกัด เช่น

1) กล้องวงจรปิด (CCTV) ให้ข้อมูลตำแหน่งเชิงภาพ (image coordinates) แต่ไวต่อแสงและการบังวัตถุ (occlusion)

2) IMU ให้ข้อมูลความเร่งและการหมุน แต่มี drift สะสมเมื่อใช้เป็นเวลานาน

3) Encoder ให้ข้อมูลระยะทางจากการหมุนของล้อ แต่ไวต่อการลื่นไถล (wheel slip)

Sensor Fusion จึงเป็นการนำข้อมูลเหล่านี้มาผสาน (fuse) เพื่อให้ได้การประมาณค่าตำแหน่งที่มีความต่อเนื่องและเชื่อถือได้มากขึ้น โดยวิธีการ Sensor Fusion มีดังนี้

1) การหลอมรวมเชิงลำดับเวลา (Temporal Fusion)

ใช้ข้อมูลที่มาจากเซนเซอร์หลายตัวในช่วงเวลาเดียวกัน แล้วนำมาประมวลผลร่วมกัน เช่น การผสานข้อมูลระหว่างตำแหน่งจากกล้อง (global) และความเร่งจาก IMU (local)

2) การหลอมรวมเชิงคณิตศาสตร์ (Mathematical Fusion) อาศัยโมเดลทางคณิตศาสตร์ เช่น

1) Kalman Filter (KF) และ Extended Kalman Filter (EKF) ใช้สำหรับข้อมูลเชิงเส้นและไม่เชิงเส้น

2) Particle Filter (PF) ใช้สำหรับปัญหาที่ซับซ้อนและการแจกแจงแบบ non-Gaussian

3) Gaussian Process Regression (GPR) ใช้ปรับแก้ค่าที่มีความไม่แน่นอนสูง เช่น scale factor จากภาพ

3) การหลอมรวมเชิงสถาปัตยกรรม (System-level Fusion)

ในงานวิจัยนี้ ใช้ Centralized Fusion เซนเซอร์ทุกตัวส่งข้อมูลไปยังหน่วยประมวลผลกลาง (Central Processing Unit) เพื่อทำการประมวลผลและตัดสินใจเพียงจุดเดียว

1) ข้อดี: ลดภาระการประมวลผลบนหุ่นยนต์, บริหารจัดการหลายคันพร้อมกันได้

2) ข้อเสีย: พึ่งพาความเสถียรของระบบสื่อสาร

ดังนั้น Sensor Fusion เป็นหัวใจสำคัญในการทำให้ระบบระบุตำแหน่งของ AGV ต่อเนื่อง แม่นยำ และ robust ต่อข้อจำกัดของเซนเซอร์แต่ละตัว โดยงานวิจัยนี้ใช้การแบบผสม (Hybrid) โดยจะเป็นการผสานข้อมูลจาก CCTV + IMU + Encoder ผ่านการประมวลผลรวมศูนย์ (Centralized Fusion) ร่วมกับเทคนิค AI และ GPR เพื่อเพิ่มความทนทานต่อ noise และ occlusion

2.5 ระบบควบคุมแบบรวมศูนย์ (Centralized Control System)

ระบบควบคุมแบบรวมศูนย์ (Centralized Control System) คือ สถาปัตยกรรมที่ข้อมูลจากเซนเซอร์ทุกตัว เช่น กล้อง, IMU, Encoder จะถูกส่งไปยัง ศูนย์ควบคุมกลาง (Central Processing Unit) เพื่อประมวลผลและตัดสินใจ จากนั้นคำสั่งควบคุมจะถูกส่งกลับไปยังหุ่นยนต์หรือยานพาหนะ จุดเด่น คือ การที่ระบบสามารถ มองเห็นภาพรวม (global view) ของสภาพแวดล้อมทั้งหมด ทำให้การควบคุมยานพาหนะหลายคัน (multi-robot system) สามารถจัดการจากศูนย์กลางได้ง่ายขึ้น แต่ก็จะมีจุดด้อยบางประการดังตารางเปรียบเทียบที่ 2.8

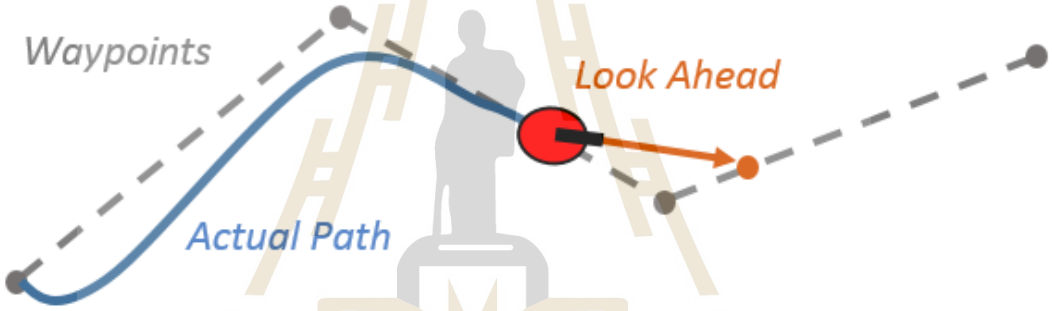
ตารางที่ 2.8 การเปรียบเทียบระหว่างระบบศูนย์รวมกับระบบแยก

คุณสมบัติ	Centralized Control	Decentralized Control
การประมวลผล	ทำที่ศูนย์กลางเพียงจุดเดียว	แต่ละหุ่นยนต์ตัดสินใจเอง
การสื่อสาร	ต้องส่งข้อมูลทั้งหมดไปยังศูนย์กลาง	ใช้การแลกเปลี่ยนข้อมูลเฉพาะที่ (local)
ข้อดี	เห็นภาพรวมทั้งระบบ ง่ายต่อการจัดการหลายคันและ ลดต้นทุนฮาร์ดแวร์บนหุ่นยนต์	ลดการพึ่งพาศูนย์กลางและ ทนทานต่อความล้มเหลวบางส่วน
ข้อเสีย	พึ่งพาศูนย์กลางสูง (single point of failure) และมี latency จากการสื่อสาร	ต้องใช้เซนเซอร์และประมวลผลบนหุ่นยนต์มากขึ้นและ ซับซ้อนต่อการ synchronize

หนึ่งในองค์ประกอบสำคัญของระบบควบคุมแบบรวมศูนย์ คือ การส่งคำสั่งเคลื่อนที่จากหน่วยประมวลผลกลางไปยัง AGV เพื่อให้สามารถติดตามเส้นทางที่วางไว้ได้อย่างราบรื่นและแม่นยำ โดยเทคนิคที่ใช้ในงานวิจัยนี้คือ Pure Pursuit Algorithm ซึ่งเป็นวิธีการควบคุมหุ่นยนต์ให้ติดตามเส้นทางโดยการเล็งเป้าหมายระยะใกล้ (lookahead point) ในเส้นทางที่กำหนด

Pure Pursuit Controller

Pure Pursuit เป็นหนึ่งในวิธีการนำทางแบบ kinematic ที่เน้น การคำนวณมุมเลี้ยว (curvature) จากตำแหน่งปัจจุบันของยานพาหนะไปยังจุดที่อยู่ข้างหน้า (lookahead point) ซึ่งอยู่บนเส้นทางที่ต้องการให้หุ่นยนต์เคลื่อนไป โดยใช้หลักการเรขาคณิตในการสร้างเส้นวงกลมที่เชื่อมระหว่างตำแหน่งปัจจุบันกับเป้าหมายในระยะใกล้ แล้วคำนวณรัศมีของวงกลมดังกล่าวเพื่อนำไปกำหนดมุมพวงมาลัยหรืออัตราความเร็วเชิงมุม ดังรูปที่ 2.25



รูปที่ 2.25 หลักการ Pure Pursuit Control (VineSM, 2020)

จากรูปทรงเรขาคณิต 2 มิติ สามารถคำนวณความโค้ง κ ที่ใช้ควบคุมได้จากสมการที่ (2.24)

$$\kappa = \frac{2 \sin \alpha}{L_d} \tag{2.24}$$

หรือหากใช้สำหรับคำนวณมุมพวงมาลัยสำหรับยานพาหนะจะแสดงดังสมการที่ (2.25)

$$\delta = \tan^{-1}\left(\frac{2L \sin \alpha}{L_d}\right) \tag{2.25}$$

โดยที่

L คือ ระยะฐานล้อของรถ (wheelbase)

L_d คือ ระยะทาง lookahead (ค่าคงที่ หรือปรับตามความเร็ว)

α คือ มุมระหว่าง *heading* กับเส้นตรงไปยัง *lookahead point*

κ คือ ความโค้งของเส้นทาง

ระบบควบคุมจะทำการวางแผนเส้นทางด้วย A* และส่งชุดของตำแหน่ง (waypoints) ให้กับ Pure Pursuit ที่ฝั่ง AGV ซึ่งจะใช้ heading จาก IMU และตำแหน่งจากกล้องวงจรปิดในการคำนวณทิศทางมุ่งหน้า (steering direction) พร้อมกำหนดความเร็วและทิศการเลี้ยวให้เหมาะสมกับระยะ lookahead และความโค้งของเส้นทางที่วางไว้ ดังนั้น ระบบควบคุมแบบรวมศูนย์ในงานวิจัยนี้ไม่เพียงช่วยลดต้นทุนและภาระการประมวลผลบน AGV แต่ยังช่วยให้การบริหารจัดการหลายคันจากศูนย์กลางมีประสิทธิภาพ โดยมี Pure Pursuit Algorithm เป็นกลไกหลักที่ใช้ในการติดตามเส้นทาง (path tracking) อย่างแม่นยำและมีความเสถียร

2.6 โปรโตคอลการสื่อสาร (MQTT)

Message Queuing Telemetry Transport (MQTT) เป็นโปรโตคอลการสื่อสารแบบ publish–subscribe ที่ถูกออกแบบมาสำหรับเครือข่ายที่มีแบนด์วิดท์จำกัดและความหน่วงต่ำ เช่น IoT และระบบอัตโนมัติ

โครงสร้างของ MQTT ประกอบด้วย:

อุปกรณ์ที่ส่งข้อมูล (เช่น Jetson Nano ที่ส่งข้อมูลเซนเซอร์)

Subscriber: อุปกรณ์ที่รับข้อมูล (เช่น Laptop ที่ทำหน้าที่ประมวลผล)

Broker: ตัวกลางที่ทำหน้าที่จัดการการส่งต่อข้อความ โดยใช้ topic เป็นตัวกำหนดช่องทาง

1) คุณสมบัติเด่น

Lightweight Protocol ซึ่งใช้ header ขนาดเล็ก ทำให้เหมาะกับ embedded devices

1) Low Latency โดยรองรับ real-time communication

2) Topic-based Routing แบ่งข้อมูลตามหัวข้อ เช่น AGV1/position, AGV2/control ทำให้รองรับ multi-robot ได้ง่าย

3) QoS (Quality of Service) กำหนดระดับความน่าเชื่อถือของการส่งข้อความได้ 3 ระดับ

QoS 0: at most once

QoS 1: at least once

QoS 2: exactly once

4) Scalability รองรับตั้งแต่ระบบภายในอาคารจนถึงการเชื่อมต่อกับ Cloud

2) การประยุกต์ใช้งานในงานวิจัย

ในงานนี้ MQTT ถูกนำมาใช้เป็นกลไกหลักในการสื่อสารระหว่างอุปกรณ์ ดังนี้:

1) ส่งข้อมูลจาก Jetson Nano ไปยัง Laptop (Central Processing Unit) เช่น ตำแหน่งจาก IMU, Encoder, และผลการตรวจจับจากกล้อง

2) ส่งคำสั่งควบคุมจาก Laptop ไปยัง Arduino Mega เพื่อสั่งการมอเตอร์

3) ทำงานผ่าน เครือข่าย Wi-Fi/LAN ที่มีอยู่ในอาคาร ซึ่งทำให้ติดตั้งง่าย ไม่ต้องเพิ่มโครงสร้างพื้นฐานพิเศษ

ตารางที่ 2.9 ข้อเปรียบเทียบของโปรโตคอลสื่อสาร

โปรโตคอล	ลักษณะเด่น	ข้อจำกัด
MQTT	Lightweight, real-time, ใช้ bandwidth ต่ำ, รองรับ pub-sub	ต้องมี broker กลาง
HTTP/REST	ใช้งานง่าย, เข้ากับ Web services ได้ดี	Latency สูง, ไม่เหมาะกับ real-time
ROS (Robot Operating System) Topics	เหมาะกับ robotics ecosystem	ต้องติดตั้ง ROS environment, ซับซ้อนกว่า
AMQP (Advanced Message Queuing Protocol)	รองรับ enterprise system, มีความปลอดภัยสูง	ใช้ bandwidth และพลังงานมากกว่า MQTT

2.7 ปริทัศน์วรรณกรรมที่เกี่ยวข้อง

เพื่อสร้างรากฐานทางวิชาการและระบุแนวทางการพัฒนาของงานวิจัยปัจจุบัน จำเป็นต้องทบทวนงานวิจัยที่เกี่ยวข้องกับการระบุตำแหน่งภายในอาคาร การตรวจจับวัตถุด้วยปัญญาประดิษฐ์ การหลอมรวมข้อมูลจากเซนเซอร์ การควบคุมแบบรวมศูนย์ และการใช้วิธีการเชิงสถิติขั้นสูงเพื่อเพิ่มความแม่นยำ การปริทัศน์วรรณกรรมในหัวข้อนี้จะช่วยชี้ให้เห็นถึงพัฒนาการของเทคโนโลยี จุดแข็งและข้อจำกัดของแนวทางที่มีอยู่ ตลอดจนแนวทางใหม่ที่งานวิจัยนี้จะนำเสนอเพื่อเติมเต็มช่องว่างดังกล่าว

2.7.1 การระบุตำแหน่งโดยใช้กล้องวงจรปิดและเทคนิค Image-Based

งานวิจัย: A CNN Regression Approach to Mobile Robot Localization Using Omnidirectional Images

คณะผู้วิจัย/ปีที่วิจัย: Mónica, Luis, Sergio, Oscar and Francisco, (2021)

จุดประสงค์การศึกษา: พัฒนาโมเดลระบุตำแหน่งของหุ่นยนต์เคลื่อนที่ในอาคาร โดยใช้ภาพจากกล้องมุมมองรอบทิศ (omnidirectional camera) และ Convolutional Neural Networks (CNNs) เพื่อให้สามารถทราบตำแหน่งของตนเองได้อย่างแม่นยำ

วิธีการและเทคนิค:

1) ใช้เทคนิค Transfer Learning กับ CNN (AlexNet) เพื่อจำแนกว่าหุ่นยนต์อยู่ในห้องใด (classification)

2) สร้าง CNN แบบ Regression สำหรับแต่ละห้อง เพื่อคำนวณพิกัด (X, Y)

3) ปรับปรุงความแม่นยำด้วยการ แบ่งพื้นที่ห้องใหญ่ ออกเป็นหลายโซน และฝึกโมเดลแยกกันในแต่ละโซน

งานวิจัย: A Review of Solutions for Perspective-n-Point Problem in Camera Pose Estimation

คณะผู้วิจัย/ปีที่วิจัย: Xiao, (2018)

จุดประสงค์การศึกษา: ทบทวนปัญหา Perspective-n-Point (PnP) ซึ่งใช้ประมาณตำแหน่งและมุมมองของกล้อง จากข้อมูล 3D-2D correspondence และอธิบายวิธีการแก้ไขที่นิยม ได้แก่ P3P กับ EPnP

วิธีการและเทคนิค:

1) P3P (Perspective-3-Point): ใช้จุดควบคุม 3 จุด คำนวณตำแหน่งกล้องด้วยระบบสมการพีชคณิต และเทคนิค Wu-Ritt's Zero Decomposition

2) EPnP (Efficient PnP): ใช้จุดควบคุมเสมือน 4 จุด เพื่อลดความซับซ้อนในการประมวลผลสำหรับกรณีที่มีจุดจำนวนมาก

3) เน้นความเหมาะสมของแต่ละวิธีในสถานการณ์ต่าง ๆ และนำไปใช้ในการประมาณท่าทางของกล้องในระบบนำทางของหุ่นยนต์

งานวิจัย: Real-Time Simultaneous Localisation and Mapping with a Single Camera

คณะผู้วิจัย/ปีที่วิจัย: Andrew, (2003)

จุดประสงค์การศึกษา: เพื่อพัฒนาเทคนิค SLAM แบบ real-time โดยใช้เพียง กล้องตัวเดียว (monocular) สำหรับการระบุตำแหน่งและสร้างแผนที่ของสภาพแวดล้อมในห้องที่ไม่รู้จักมาก่อน โดยเน้นการใช้งานที่รวดเร็วและแม่นยำ

วิธีการและเทคนิค:

1) ใช้ Bayesian filtering (Extended Kalman Filter) เพื่อประมาณตำแหน่งของกล้องและ landmark features ในฉาก

2) ออกแบบระบบให้สามารถ ติดตามตำแหน่งได้แบบ real-time ด้วยการเลือกใช้ feature ที่มีความโดดเด่นสูง และการจัดการกับความไม่แน่นอนอย่างมีประสิทธิภาพ

3) นำเสนอเทคนิค feature initialization แบบ Bayesian particle filtering เพื่อให้สามารถประมาณความลึกของจุดได้จากกล้องตัวเดียวโดยไม่ต้องใช้ stereo vision

งานวิจัย: Vision-based simultaneous localization and mapping with two cameras

คณะผู้วิจัย/ปีที่วิจัย: Gab-Hoe, Jong-Sung and Ki-Sang, (2005)

จุดประสงค์การศึกษา: เพื่อนำเสนอวิธีการทำ SLAM ด้วยกล้องสองตัว (ที่เคลื่อนที่แยกจากกัน) เพื่อแก้ไขข้อจำกัดของกล้องเดี่ยวในการประมาณตำแหน่ง และลดความซับซ้อนของการคำนวณโดยใช้การประมาณร่วมผ่าน Extended Kalman Filter

วิธีการและเทคนิค:

1) ใช้ Extended Kalman Filter สำหรับประมาณตำแหน่งกล้องและฟีเจอร์ในแผนที่ร่วมกันจากกล้องสองตัว

2) ออกแบบโมเดลการเคลื่อนที่และการวัด (motion & measurement model) สำหรับกล้องที่เคลื่อนที่อิสระ

3) ใช้ particle filtering และ triangulation เพื่อประมาณความลึกของฟีเจอร์ได้แม่นยำขึ้นและเร่งการ convergence ของ covariance เพื่อเพิ่มความเสถียรและลดภาระการคำนวณ

งานวิจัย: A Review of Recurrent Neural Network Based Camera Localization for Indoor Environments

คณะผู้วิจัย/ปีที่วิจัย: Muhammad, Farhan, Ali and AKM, (2023)

จุดประสงค์การศึกษา: เพื่อนำเสนอภาพรวมของการใช้ Recurrent Neural Networks (RNN) สำหรับประมาณ ตำแหน่งกล้อง (Camera Pose Estimation) ในสภาพแวดล้อมภายในอาคาร โดยอาศัยลำดับภาพต่อเนื่องเพื่อให้ได้ผลที่แม่นยำกว่าการใช้ภาพเดี่ยว

วิธีการและเทคนิค:

1) วิเคราะห์งานวิจัยที่ใช้โครงข่ายเช่น LSTM และ GRU สำหรับ localization จากลำดับภาพ

2) เปรียบเทียบการใช้ PoseNet (CNN) กับ RNN-based models ที่สามารถพิจารณาข้อมูลลำดับเวลา

3) เน้นจุดแข็งของ RNN ในด้านการรับรู้ context เชิงเวลา ทำให้ระบุตำแหน่งได้แม่นยำขึ้นในสภาพแวดล้อมที่เปลี่ยนแปลง

งานวิจัย: Indoor Positioning Using PnP Problem on Mobile Phone Images
คณะผู้วิจัย/ปีที่วิจัย: Hana, Karel, Radek and Daniel, (2020)
จุดประสงค์การศึกษา: เสนอวิธีการระบุตำแหน่งภายในอาคารโดยใช้ภาพจากกล้องของสมาร์ทโฟนและการแก้ปัญหา Perspective-n-Point (PnP) เพื่อคำนวณตำแหน่งกล้องจากจุดควบคุมที่รู้พิกัด
วิธีการและเทคนิค:

1) ใช้ การติดตาม feature (SURF, ORB) เพื่อจับคู่จุดบนภาพกับจุด 3 มิติในโมเดล

2) ใช้การแก้ปัญหา PnP (เช่น EPnP) เพื่อคำนวณตำแหน่งและทิศทางของกล้อง

3) ประเมินวิธีบนสมาร์ทโฟน Android และพบว่ามีศักยภาพสำหรับระบบนำทางภายในอาคารที่ไม่ต้องใช้ GPS

งานวิจัย: Smart Building Surveillance System as Shared Sensory System for Localization of AGVs
คณะผู้วิจัย/ปีที่วิจัย: Petr, Daniel, Jan, Václav and Zdenko, (2020)
จุดประสงค์การศึกษา: พัฒนาแนวคิด Shared Sensory System โดยใช้กล้องวงจรปิดที่มีอยู่แล้วในอาคารอัจฉริยะ เพื่อระบุตำแหน่ง AGVs โดยไม่จำเป็นต้องติดตั้งเซนเซอร์บนตัวหุ่นยนต์
วิธีการและเทคนิค:

1) ใช้กล้องวงจรปิดร่วมกับ Aruco marker และการประมวลผลภาพผ่าน OpenCV

2) ใช้ 3D gridboard ติดกับ AGV เพื่อการตรวจจับตำแหน่งที่แม่นยำ

3) ทดสอบผ่านการจำลองและการทดลองจริง พร้อมวิเคราะห์ผลความแม่นยำของตำแหน่งจากจำนวนกล้องและความละเอียดของภาพ

งานวิจัย: Visual indoor positioning with a single camera using PnP
คณะผู้วิจัย/ปีที่วิจัย: Edith, Mirza, Joshua and Michael, (2015)
จุดประสงค์การศึกษา: เพื่อพัฒนาวิธีการระบุตำแหน่งของกล้องในอาคารแบบ 6 องศาอิสระ (6 DOF) โดยใช้ภาพจากกล้องตัวเดียว (monocular) และเทคนิค PnP โดยไม่ต้องพึ่งพา IMU หรือ GPS

วิธีการและเทคนิค:

1) ใช้ ฐานข้อมูลของ feature descriptors ที่ได้จากภาพและตำแหน่งที่ทราบล่วงหน้า

2) ค้นหาภาพที่ใกล้เคียงกับภาพ query ด้วย feature matching

2.1) คำนวณตำแหน่งของกล้องด้วย PnP (Perspective-n-Point) โดยใช้ EPnP + RANSAC

2.2) ระบบสามารถทำงานแบบ real-time ด้วยความแม่นยำระดับ mm ในสภาพ indoor environment

2.7.2 การตรวจจับวัตถุด้วยปัญญาประดิษฐ์ (Object Detection for Mobile Robot)

งานวิจัย: AGV Detection in Industrial Environments through Computer Vision

คณะผู้วิจัย/ปีที่วิจัย: Wilson et al., (2022)

จุดประสงค์การศึกษา: เพื่อพัฒนาโมเดลตรวจจับ AGV ในโรงงานอุตสาหกรรมโดยใช้ภาพจากกล้องวงจรปิดร่วมกับเทคนิค Deep Learning แทนการใช้ระบบควบคุมราคาแพง และลดเวลาในการค้นหาเครื่องจักรที่ออกจากเส้นทาง

วิธีการและเทคนิค:

1) ใช้เทคนิค YOLOv5 สำหรับ Object Detection

2) ใช้ Transfer Learning เพื่อฝึกโมเดลจากชุดข้อมูลที่มีจำนวนจำกัด (1,067 ภาพจาก 4 รุ่น AGV)

3) ประเมินผลด้วย F1-score ซึ่งได้ผลลัพธ์แม่นยำกว่า 80% แม้ในสภาพแวดล้อมจริงที่มีแสงและพื้นหลังแตกต่างกัน

งานวิจัย: Building Occupancy Detection and Localization Using CCTV Camera and Deep Learning

คณะผู้วิจัย/ปีที่วิจัย: Shushan, Peng, Cathal and James, (2022)

จุดประสงค์การศึกษา: เพื่อพัฒนาระบบอัจฉริยะที่สามารถตรวจจับและระบุตำแหน่งของผู้อยู่ภายในอาคารแบบเรียลไทม์ โดยใช้ข้อมูลจากกล้องวงจรปิด (CCTV) ร่วมกับโมเดล Deep Learning

วิธีการและเทคนิค:

1) ใช้ YOLOv4 สำหรับการตรวจจับบุคคลในภาพจากกล้อง

2) ใช้ homography transformation และ camera calibration เพื่อแปลงตำแหน่งของบุคคลจากภาพไปยังแผนที่ของอาคาร

3) ระบบสามารถตรวจจับได้ทั้ง “การมีอยู่ (occupancy)” และ “ตำแหน่ง (localization)” ของบุคคลในแบบ real-time ได้อย่างแม่นยำ

งานวิจัย: Target Detection and Position Measurement Based on Machine Vision for AGV
คณะผู้วิจัย/ปีที่วิจัย: Guiyang, Lingyu, Siyu and Xu, (2023)
จุดประสงค์การศึกษา: เพื่อปรับปรุงความแม่นยำและความเร็วในการตรวจจับตำแหน่งปลายทางของ AGV โดยใช้ระบบมองเห็นด้วยกล้องร่วมกับเทคนิค machine vision และ deep learning
วิธีการและเทคนิค:

1) พัฒนา YOLOv3 โดยใช้ cross-layer connection และ deconvolution เพื่อเพิ่มความแม่นยำในการตรวจจับวัตถุขนาดเล็ก

2) ใช้กล้อง stereo vision สำหรับ 3D reconstruction ตำแหน่ง marker บนถาดเป้าหมาย

3) ดำเนินการทดสอบความแม่นยำของระบบ โดยได้ค่าความผิดพลาดในการวัดต่ำกว่า 1.5 มม. ในระยะไม่เกิน 5 เมตร

4) ระบบสามารถทำงานได้แบบ real-time และมีความทนทานแม้มี marker บางตัวสูญหายจากการตรวจจับ

2.7.3 การหลอมรวมข้อมูลเซนเซอร์สำหรับการระบุตำแหน่งภายในอาคาร

งานวิจัย: Artificial Neural Networks Aided Image Localization for Pedestrian Dead Reckoning for Indoor Navigation Applications
คณะผู้วิจัย/ปีที่วิจัย: Jhen-Kai, Kai-Wei, Hsiu-Wen and Yu-Hua, (2018)
จุดประสงค์การศึกษา: เพื่อเพิ่มความแม่นยำของระบบระบุตำแหน่งในอาคารโดยใช้ภาพถ่าย marker และการเดินคาดคะเน (Pedestrian Dead Reckoning - PDR) พร้อมการช่วยเหลือด้วยโครงข่ายประสาทเทียม
วิธีการและเทคนิค:

1) ใช้ Artificial Neural Networks (ANNs) เพื่อช่วยระบุตำแหน่งจากภาพถ่ายของ marker ที่ติดตั้งไว้ในอาคาร

2) ผสานข้อมูลจาก IMU sensor และ การประมวลผลภาพ (space resection) เพื่อคำนวณตำแหน่งของผู้ใช้งาน

3) ทดสอบในสมาร์ทโฟน Android และพิสูจน์ว่า ANN ช่วยเพิ่มความแม่นยำของ PDR ได้ชัดเจน

งานวิจัย: Camera/LiDAR Sensor Fusion-based Autonomous Navigation

คณะผู้วิจัย/ปีที่วิจัย: Abdullah, Akif and Ibrahim, (2024)

จุดประสงค์การศึกษา: นำเสนอวิธีการใหม่ในการหลอมรวมข้อมูลจากกล้องและเซนเซอร์ LiDAR สำหรับ การนำทางอัตโนมัติของหุ่นยนต์ภาคพื้นดิน (UGV) โดยเน้นการตรวจจับสิ่งกีดขวางและประมาณระยะทางด้วยความแม่นยำสูง

วิธีการและเทคนิค:

1) ใช้กล้อง RGB (ZED2) และ LiDAR (Velodyne VLP-16) พร้อมเทคนิค YOLOv7 สำหรับตรวจจับวัตถุ

2) ประมวลผลจุดข้อมูลจาก LiDAR โดยจับคู่กับ Bounding Boxes ที่ตรวจจับจากกล้องเพื่อประมาณ ระยะทางของวัตถุ

3) ดำเนินการทดลองจริงบนรถทดสอบพร้อมการวางแผนเส้นทางหลบหลีกสิ่งกีดขวางในสภาพแวดล้อมจริง

งานวิจัย: Enhancing Localization Accuracy and Reducing Processing Time in Indoor Positioning Systems: A Comparative Analysis of AI Models

คณะผู้วิจัย/ปีที่วิจัย: Salwa et al., (2025)

จุดประสงค์การศึกษา: ปรับปรุงความแม่นยำของระบบระบุตำแหน่งในอาคาร และลดเวลาในการประมวลผล โดยเปรียบเทียบประสิทธิภาพของโมเดล AI ต่าง ๆ

วิธีการและเทคนิค:

1) วิเคราะห์เปรียบเทียบโมเดล ANN, CNN, KNN, SVM, Decision Tree, XGBoost และ LightGBM

2) ใช้ข้อมูลจาก Wi-Fi, BLE, และ Magnetic field ในชุดข้อมูล UJIIndoorLoc

3) พบว่า LightGBM มีความแม่นยำสูงสุด (98.1%) และใช้เวลาประมวลผลน้อยที่สุด

งานวิจัย: Multiple Objects Localization With Camera-LIDAR Sensor Fusion

คณะผู้วิจัย/ปีที่วิจัย: Gökçe and Emrah (2025)

จุดประสงค์การศึกษา: พัฒนาวิธีระบุตำแหน่งของวัตถุหลายชิ้น (เช่น คนและรถ) โดยใช้การหลอมรวมข้อมูลจากกล้องและ LiDAR สำหรับการตรวจจับวัตถุอย่างแม่นยำในบริบทของระบบขับเคลื่อนอัตโนมัติ

วิธีการและเทคนิค:

1) ใช้กล้อง RGB สำหรับการตรวจจับวัตถุด้วยโมเดล YOLOv5

2) ผสานข้อมูลจาก LiDAR เพื่อคำนวณระยะห่างของวัตถุแต่ละตัวในภาพที่ตรวจจับได้

3) แปลงจุดตรวจจับบนภาพให้สอดคล้องกับจุดใน cloud ของ LiDAR ด้วย calibration matrix

4) ทดสอบในสภาพแวดล้อมจริง เช่น ท้องถนน โดยสามารถตรวจจับและระบุตำแหน่งของวัตถุได้หลายรายการพร้อมกันด้วยความแม่นยำสูง

2.7.4 การควบคุมยานพาหนะอัตโนมัติแบบรวมศูนย์ (Centralized AGV Control)

งานวิจัย: Global Localization and Position Tracking of an Automated Guided Vehicle

คณะผู้วิจัย/ปีที่วิจัย: Christopher and Christof, (2011)

จุดประสงค์การศึกษา: พัฒนาวิธีการระบุตำแหน่งแบบทั่วโลก (global localization) และการติดตามตำแหน่ง (position tracking) สำหรับ AGV ที่เคลื่อนที่แบบรอบทิศทาง โดยใช้ sensor fusion จากเซนเซอร์ระยะทางของ WSN (IEEE 802.15.4a) และ เลเซอร์ range finders เพื่อเพิ่มความแม่นยำ

วิธีการและเทคนิค:

1) ใช้ Monte Carlo Particle Filter (MCP) แทน Kalman Filter เพื่อจัดการกับข้อมูลที่ไม่เป็น Gaussian และรองรับความไม่แน่นอนหลายรูปแบบ

2) ข้อมูลจาก WSN (nanoLOC) ใช้สำหรับการระบุตำแหน่งแบบไม่รู้ล่วงหน้า (no a priori)

3) ข้อมูลจาก เลเซอร์เซนเซอร์ ใช้ตรวจจับ landmark ที่ติดแถบสะท้อนแสง เพื่อเพิ่มความแม่นยำใน docking

4) มีการทดลองจริงในห้องทดลองที่จำลองสภาพคลังสินค้า โดยวัดผลการนำทางและการเข้า docking

2.7.5 การประยุกต์ใช้ Gaussian Process Regression เพื่อเพิ่มความแม่นยำ

งานวิจัย: A Review of SLAM Techniques and Security in Autonomous Driving

คณะผู้วิจัย/ปีที่วิจัย: Ashutosh and Hung, (2019)

จุดประสงค์การศึกษา: นำเสนอภาพรวมของเทคนิค SLAM ที่ใช้ในรถยนต์ไร้คนขับ โดยเน้นเทคนิคที่ประเมินบนชุดข้อมูล KITTI และอธิบายข้อดี ข้อจำกัด รวมถึงปัญหาด้านความปลอดภัยของระบบรถอัตโนมัติ

วิธีการและเทคนิค:

1) วิเคราะห์เปรียบเทียบเทคนิค SLAM ต่าง ๆ เช่น LIDAR-based, stereo-based, และ monocular SLAM

2) เน้นเทคนิค Gaussian Process Regression ในการประมาณเส้นทางอย่างแม่นยำในวิธี STEAM และการแก้ drift error

3) กล่าวถึง การใช้ CNN-SLAM และ semantic SLAM เพื่อเพิ่มความสามารถในการเข้าใจสิ่งแวดล้อมแบบเชิงความหมาย

งานวิจัย: Gaussian Process Models for Indoor and Outdoor Sensor-Centric Robot Localization

คณะผู้วิจัย/ปีที่วิจัย: Alex, Alexei and Ben, (2008)

จุดประสงค์การศึกษา: พัฒนาโมเดล localization ของหุ่นยนต์ทั้งในและนอกอาคาร โดยไม่พึ่งพาแบบจำลองเรขาคณิตของเซนเซอร์ ด้วยการใช้ Gaussian Process (GP) ที่เรียนรู้จากข้อมูลภาพที่ได้จากกล้อง

วิธีการและเทคนิค:

1) ใช้ ภาพจากกล้อง omnidirectional ที่ผูกกับตำแหน่งที่รู้พิกัด

2) แปลงภาพด้วย PCA และ wavelet decomposition เป็น feature vectors

3) ฝึก Gaussian Process Regression เพื่อคาดคะเนพิกัดของหุ่นยนต์จากภาพ

4) ใช้ร่วมกับ Particle Filter ในการติดตามตำแหน่งแบบ real-time

งานวิจัย: Image-based localization using Gaussian processes

คณะผู้วิจัย/ปีที่วิจัย: Manuel, Nicolai and Javier, (2016)

จุดประสงค์การศึกษา: เสนอโมเดลการสังเกต (observation model) สำหรับการระบุตำแหน่งจากภาพ โดยใช้ Gaussian Processes ที่สามารถประมาณค่าความน่าจะเป็นของตำแหน่งของกล้องจากภาพถ่ายได้ในลักษณะต่อเนื่อง (ไม่จำกัดเฉพาะ keyframe)

วิธีการและเทคนิค:

1) ใช้ whole-image descriptors จาก CNN (เช่น AlexNet และ DCNN) เพื่อเป็นตัวแทนภาพ

2) สร้างโมเดล GP ที่เรียนรู้จากภาพและตำแหน่งของกล้อง เพื่อใช้ในการคำนวณความน่าจะเป็นของตำแหน่งใหม่

3) ประยุกต์ใช้ใน Particle Filter สำหรับ localization และเปรียบเทียบกับการใช้ข้อมูลจากเลเซอร์

4) ผลการทดลองในชุดข้อมูล TUMindoor แสดงให้เห็นว่าโมเดล GP นี้สามารถทำ localization ได้แม่นยำและใช้จำนวนน้อยกว่า particles ในขั้นเริ่มต้น

2.8 บทสรุปของการทบทวนวรรณกรรม

จากการศึกษาวรรณกรรมที่เกี่ยวข้อง พบว่าการประยุกต์ใช้ กล้องวงจรปิด (Surveillance Cameras) ร่วมกับ เทคนิคปัญญาประดิษฐ์ (Artificial Intelligence) โดยเฉพาะการตรวจจับวัตถุด้วย YOLO (You Only Look Once) สามารถนำมาใช้เพื่อสนับสนุนการระบุตำแหน่งและการนำทางของยานพาหนะภาคพื้นดินอัตโนมัติ (AGV) ภายในอาคารได้อย่างมีประสิทธิภาพ โดยไม่จำเป็นต้องพึ่งพาเซนเซอร์ 3 มิติที่มีต้นทุนสูง เช่น LIDAR หรือกล้อง RGB-D

นอกจากนี้ การออกแบบระบบควบคุมแบบ รวมศูนย์ (Centralized Control) ซึ่งใช้หน่วยประมวลผลกลางรับข้อมูลภาพจากกล้องและข้อมูลเซนเซอร์ภายใน AGV ผ่านโพรโทคอล MQTT ยังช่วยลดภาระของหน่วยประมวลผลภายในตัวรถ ส่งผลให้สามารถออกแบบฮาร์ดแวร์ AGV ให้มีต้นทุนต่ำลง โดยไม่ลดทอนประสิทธิภาพของระบบ

แนวทางในการหลอมรวมข้อมูลจากหลายแหล่ง (Sensor Fusion) อาทิ กล้อง, IMU, Encoder ร่วมกับเทคนิคการประมาณค่าทางสถิติ เช่น Gaussian Process Regression (GPR) และ Polynomial Fitting ยังสามารถช่วยปรับปรุงความแม่นยำของการระบุตำแหน่ง โดยเฉพาะในบริเวณที่มีจุดอับหรือสัญญาณภาพไม่สมบูรณ์

ในภาพรวม การบูรณาการองค์ความรู้จากหลากหลายสาขา ได้แก่ วิทยาการหุ่นยนต์ การประมวลผลภาพ ปัญญาประดิษฐ์ และการควบคุมอัตโนมัติ นำไปสู่แนวทางใหม่ในการพัฒนา ระบบนำทางอัตโนมัติที่มีต้นทุนต่ำ ประสิทธิภาพสูง และสามารถประยุกต์ใช้งานได้จริงในสภาพแวดล้อมภายในอาคาร ซึ่งถือเป็นแนวโน้มที่มีความสำคัญในงานวิจัยด้าน AMR/AGV ยุคปัจจุบัน

บทที่ 3
วิธีดำเนินงานวิจัย

ในบทนี้จะอธิบายถึงวิธีการออกแบบ พัฒนา และทดสอบระบบนำทางและขับเคลื่อนอัตโนมัติสำหรับยานพาหนะภาคพื้นดิน (Autonomous Ground Vehicle: AGV) ภายใต้บริบทของการใช้งานภายในอาคาร โดยใช้ข้อมูลจากกล้องวงจรปิด (Surveillance Cameras) ร่วมกับเทคนิคการประมวลผลภาพและปัญญาประดิษฐ์ สอดคล้องกับแนวคิดที่ได้กล่าวในบทที่ 1 และบทที่ 2 โดยจะนำองค์ความรู้ที่เกี่ยวข้อง ได้แก่ YOLOv11, PnP, GPR, A* และ Pure Pursuit มาประยุกต์ใช้จริง

ระบบที่นำเสนอมีลักษณะเป็น ระบบควบคุมแบบรวมศูนย์ (Centralized Control System) โดยหน่วยประมวลผลกลางจะรับข้อมูลภาพจากกล้องวงจรปิด และข้อมูลภายในจากตัว AGV ได้แก่ IMU และ Encoder ผ่านโปรโตคอล MQTT จากนั้นทำการตรวจจับวัตถุทั้ง AGV และสิ่งกีดขวางด้วย YOLOv11 และประเมินตำแหน่งด้วยเทคนิค PnP ซึ่งในงานวิจัยนี้เลือกใช้ จุดอ้างอิงจาก bounding box (4-point) ของ AGV ที่ได้จากการตรวจจับ ร่วมกับการปรับค่าความแม่นยำด้วย GPR เพื่อให้การประมาณตำแหน่งมีความถูกต้องมากขึ้น

จากข้อมูลตำแหน่งและแผนที่เสมือนที่สร้างขึ้น จะมีการวางเส้นทางอัตโนมัติด้วยอัลกอริทึม A* และควบคุมการเคลื่อนที่ด้วย Pure Pursuit เพื่อให้ AGV เดินทางไปยังเป้าหมายที่กำหนด โดยการประเมินผลจะใช้ตัวชี้วัดมาตรฐาน ได้แก่ Mean Average Precision (mAP), Precision, Recall สำหรับการตรวจจับวัตถุ และ Root Mean Square Error (RMSE) สำหรับการประมาณตำแหน่ง รวมถึง path tracking error สำหรับการควบคุมการเคลื่อนที่

เพื่อรองรับสภาพแวดล้อมจริง ระบบนี้ได้ถูกออกแบบให้สามารถใช้กับกล้องวงจรปิดทั่วไปที่มีอยู่แล้วในสถานที่ต่าง ๆ (Infrastructure-Based Sensors) โดยไม่จำเป็นต้องติดตั้งอุปกรณ์บนตัว AGV เพิ่มเติมมากนัก ส่งผลให้ต้นทุนโดยรวมของระบบต่ำลง และสามารถนำไปประยุกต์ใช้ในพื้นที่จริง เช่น โรงพยาบาล โกดังสินค้า หรืออาคารสำนักงาน

ภายในบทนี้จะประกอบด้วย:

1) แผนภาพระบบโดยรวม (System Overview Diagram)
2) รายละเอียดของฮาร์ดแวร์และซอฟต์แวร์ที่ใช้
3) ลำดับขั้นตอนการพัฒนาระบบ
4) การออกแบบการทดลองเพื่อทดสอบระบบในสภาวะต่าง ๆ
5) ตัวชี้วัดที่ใช้ในการประเมินผล

6) สมมุติฐานและข้อจำกัดของระบบ

เนื้อหาทั้งหมดจะเป็นพื้นฐานสำคัญสำหรับการดำเนินการทดลองในบทถัดไป และใช้เป็นกรอบในการวิเคราะห์ผลในบทที่ 4

3.1 ภาพรวมของระบบที่พัฒนา (System Overview)

ระบบที่พัฒนาในงานวิจัยนี้มีเป้าหมายในการควบคุมการเคลื่อนที่ของยานพาหนะภาคพื้นดินอัตโนมัติ (Autonomous Ground Vehicle: AGV) ภายในอาคาร โดยไม่พึ่งพาเซนเซอร์ราคาแพง เช่น LIDAR หรือ RGB-D กล้อง แต่ใช้กล้องวงจรปิด (Surveillance Camera) ที่ติดตั้งอยู่เดิมในพื้นที่ (Infrastructure-Based Sensors) ร่วมกับเทคนิคการประมวลผลภาพและปัญญาประดิษฐ์ (Artificial Intelligence: AI) เพื่อทำการตรวจจับวัตถุและระบุตำแหน่งของ AGV แทน โดยการควบคุมทั้งหมดจะถูกรวมศูนย์ไว้ที่หน่วยประมวลผลกลาง (Centralized Control System) ซึ่งสอดคล้องกับแนวคิดที่กล่าวไว้ในบทที่ 2 เกี่ยวกับการนำโครงสร้างพื้นฐานมาใช้ใน Localization

การทำงานของระบบสามารถอธิบายเป็นลำดับขั้นตอนดังนี้:

1) การรับภาพจากกล้องวงจรปิด

กล้องวงจรปิดทำหน้าที่จับภาพในพื้นที่การทดลอง โดยกล้องแต่ละตัวถูก calibrate ให้รู้ค่าพารามิเตอร์ Intrinsic และ Extrinsic เพื่อให้สามารถแปลงพิกัดจากภาพเข้าสู่ระบบพิกัดโลกได้

2) การตรวจจับวัตถุด้วย YOLOv11

ภาพจากกล้องถูกส่งเข้าสู่โมเดล YOLOv11 เพื่อทำการตรวจจับวัตถุ โดยมุ่งเน้นไปที่ AGV และสิ่งกีดขวาง โดย bounding box และ centroid ที่ได้จะถูกใช้เป็นจุดอ้างอิงสำหรับการ localization

3) การประมาณตำแหน่งด้วย Perspective-n-Point (PnP) และ Gaussian Process Regression (GPR)

จาก bounding box ของ AGV ระบบจะเลือก 4 จุดมุม (corners) เพื่อใช้แก้ปัญหา PnP สำหรับการแปลงพิกัดจาก image space ไปยัง world space จากนั้น GPR จะถูกใช้สร้าง Error Surface เพื่อลด systematic error และปรับปรุงความแม่นยำของการระบุตำแหน่ง

4) การสร้างแผนที่เสมือน (Virtual Map)

ข้อมูลจากกล้องทั้งหมดจะถูกนำมาสร้าง grid-based virtual map แสดงพื้นที่ที่สามารถเคลื่อนที่ได้ และตำแหน่งของสิ่งกีดขวางแบบ real-time

5) การวางแผนเส้นทางด้วย A* (A-star Algorithm)

ระบบจะคำนวณเส้นทางที่เหมาะสมที่สุดจากตำแหน่งเริ่มต้นไปยังจุดหมายด้วย A* และสามารถ re-plan ได้ทันทีหากตรวจพบสิ่งกีดขวางใหม่

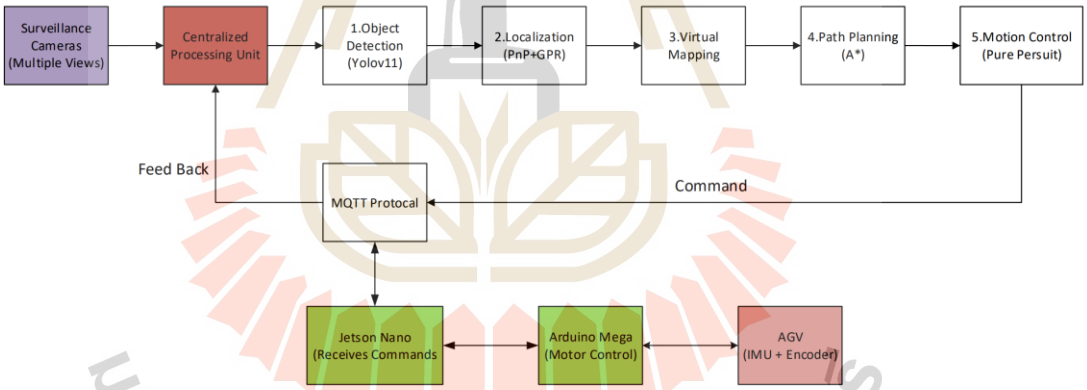
6) การควบคุมการเคลื่อนที่ด้วย Pure Pursuit

AGV จะติดตามเส้นทางด้วย Pure Pursuit ซึ่งอาศัยหลักการ geometric path tracking โดยคำนวณมุมเลี้ยว เพื่อปรับทิศทางไปยัง lookahead point ที่อยู่บนเส้นทาง ข้อมูล heading มาจาก IMU ภายใน AGV

7) การสื่อสารผ่าน MQTT และการควบคุมแบบรวมศูนย์

ระบบใช้โปรโตคอล MQTT ซึ่งมีความหน่วงต่ำและรองรับการสื่อสารแบบ publish–subscribe เพื่อเชื่อมต่อหน่วยประมวลผลกลาง (Laptop) กับ Jetson Nano และ Arduino Mega 2560 บน AGV ทำให้สามารถส่งข้อมูลเซนเซอร์และคำสั่งควบคุมแบบ real-time และรองรับการขยายสู่ multi-robot system

โดยสรุป ระบบที่พัฒนานี้เป็นการบูรณาการของ Computer Vision, Machine Learning, Path Planning, Motion Control และ Communication ภายใต้สถาปัตยกรรม Centralized Control ดังรูปที่ 3.1 ซึ่งมีจุดเด่นด้านความยืดหยุ่น ต้นทุนต่ำ และความเหมาะสมต่อการใช้งานจริงในสภาพแวดล้อมภายในอาคาร



รูปที่ 3.1 แผนภาพรวมของระบบ AGV กับ Centralized Control

3.2 อุปกรณ์และสภาพแวดล้อมในการทดลอง (Experimental Setup)

ในการดำเนินงานวิจัยนี้ ได้ออกแบบและติดตั้งระบบในสภาพแวดล้อมภายในอาคารจำลอง (indoor simulated environment) เพื่อทดสอบระบบการนำทางอัตโนมัติของยานพาหนะภาคพื้นดิน (AGV) โดยใช้อุปกรณ์และระบบสื่อสารแบบรวมศูนย์ (Centralized Control System) ที่ผสานเซนเซอร์ภายนอก (กล้องวงจรปิด) เข้ากับเซนเซอร์ภายใน AGV ผ่านระบบประมวลผลกลาง รายละเอียดของอุปกรณ์และการติดตั้งมีดังนี้:

3.2.1 กล้องวงจรปิด (Surveillance Cameras)

ระบบนำทางอัตโนมัติที่พัฒนาขึ้นในงานวิจัยนี้อาศัยการมองเห็นจากภายนอก (External Perception) เป็นหลัก โดยใช้กล้องวงจรปิดจำนวน 4 ตัวที่ติดตั้งในพื้นที่การทดลองเพื่อจับภาพของ AGV และสิ่งกีดขวางแบบเรียลไทม์ กล้องเหล่านี้จัดเป็นเซนเซอร์ประเภท Infrastructure-based ซึ่งมีข้อดีคือสามารถติดตั้งได้ง่ายในพื้นที่กว้าง และไม่สร้างภาระต่อการประมวลผลบนตัว AGV

กล้องที่เลือกใช้คือ TP-Link Tapo C200 ดังรูปที่ 3.2 ซึ่งเป็นกล้อง IP Camera แบบไร้สาย (Wireless IP Camera) รองรับการสตรีมผ่านโปรโตคอล RTSP และสามารถเชื่อมต่อกับเครือข่าย Wi-Fi ภายในอาคารเพื่อส่งข้อมูล ภาพแบบ real-time ไปยังหน่วยประมวลผลกลาง คุณสมบัติสำคัญของกล้องนี้จะแสดงดังตารางที่ 3.1

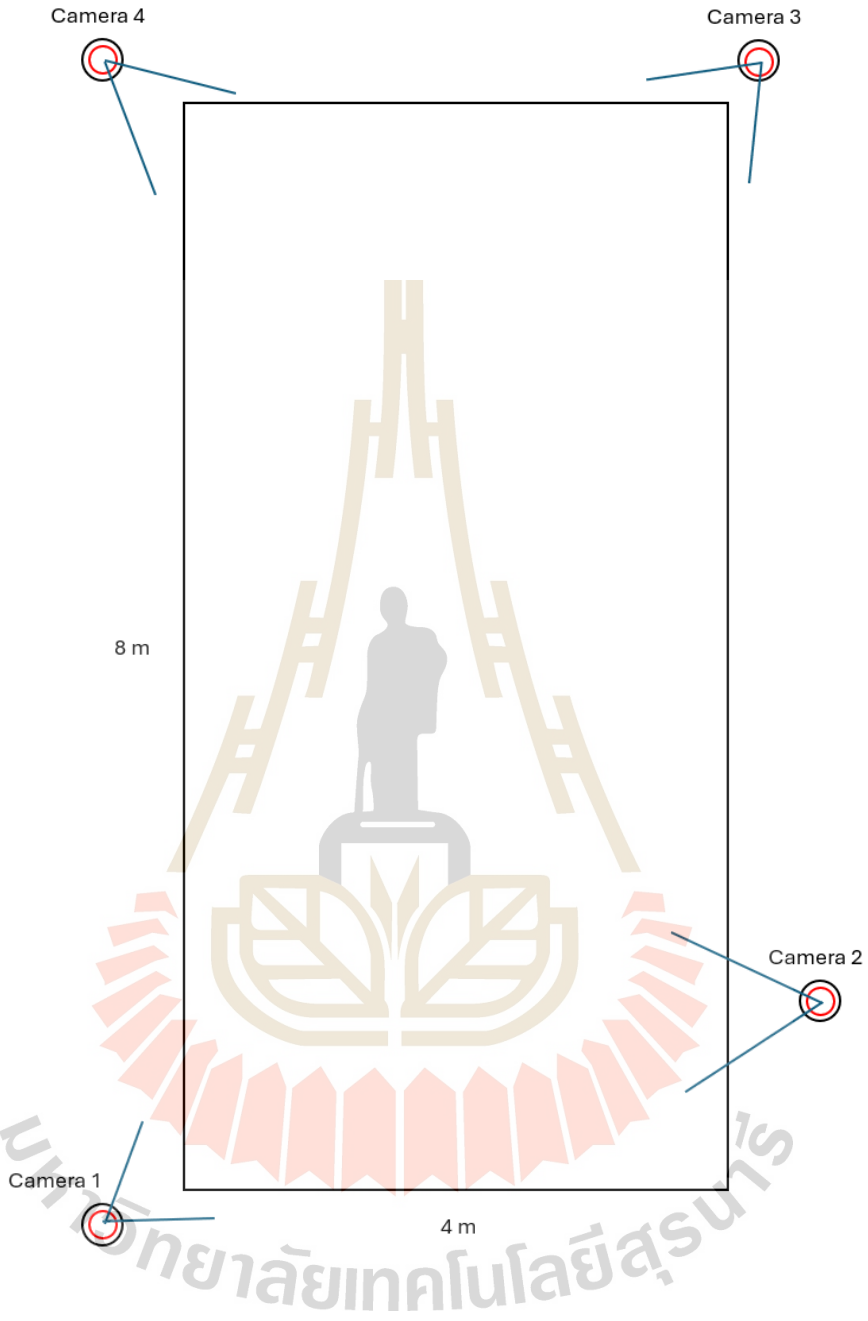
รูปที่ 3.2 กล้องวงจรปิด Tapo C200

ตารางที่ 3.1 รายละเอียดข้อมูลของกล้องวงจรปิด

รายการ	รายละเอียด
รุ่น	TP-Link Tapo C200
ประเภท	Wireless IP Camera (Pan/Tilt Home Security Wi-Fi Camera)
ความละเอียดวิดีโอ	1080p Full HD (1920 × 1080)
เลนส์	Fixed focal length 3.6mm, Aperture F2.4
อัตราเฟรม	30 เฟรมต่อวินาที
มุมมอง	Vertical tilt: 114°, Horizontal pan: 360°
การเชื่อมต่อ	IEEE 802.11 b/g/n 2.4GHz Wi-Fi
โปรโตคอลสตรีม	RTSP/ONVIF รองรับการสตรีมภาพภายนอก
การมองเห็นกลางคืน	มี IR Night Vision ระยะ ~9 เมตร
การบันทึกภาพ	รองรับ MicroSD สูงสุด 256GB
การควบคุม	ผ่านแอป Tapo บนสมาร์ตโฟน หรือเชื่อมต่อผ่าน API
การติดตั้ง	ยึดเพดานหรือวางบนพื้นราบ พร้อมไฟเลี้ยงแบบ micro-USB

เหตุผลที่เลือกใช้ Tapo C200 คือ มีต้นทุนต่ำ เข้าถึงง่าย สามารถพบเห็นได้ตามอาคารทั่วไป และสามารถใช้งานกับซอฟต์แวร์ open-source เช่น OpenCV และ YOLO ได้ทันที อีกทั้งความสามารถในการปรับมุม Pan-Tilt ยังช่วยให้ครอบคลุมพื้นที่การทดลองได้ดีขึ้น

การติดตั้งกล้องจำนวน 4 ตัวในพื้นที่ทดลองได้กำหนดไว้ที่ความสูง 2.94 เมตร จากพื้น ดังรูปที่ 3.3 ครอบคลุมพื้นที่การเคลื่อนที่ส่วนใหญ่ของ AGV โดยแต่ละตัวผ่านการ camera calibration (Intrinsic & Extrinsic parameters) เพื่อให้สามารถนำพิกัดภาพไปใช้ในการคำนวณ Perspective-n-Point (PnP) ได้อย่างถูกต้องและสอดคล้องกับระบบพิกัดโลก

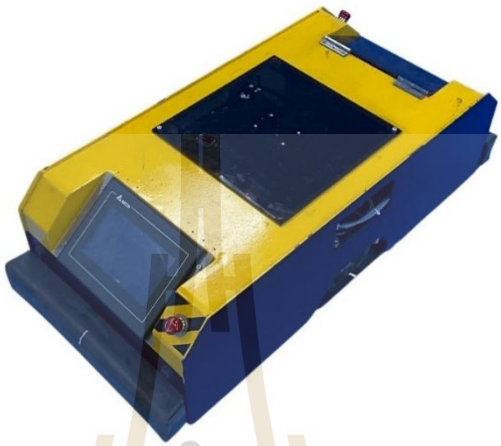


รูปที่ 3.3 พื้นที่ทดสอบที่ติดตั้งกล้องวงจรปิด

3.2.2 ยานพาหนะภาคพื้นดินอัตโนมัติ (AGV Platform)

AGV ที่ใช้ในการทดลองถูกออกแบบให้มีโครงสร้างแบบ Differential Drive โดยใช้ล้อขับเคลื่อนหลัก 2 ล้อ และล้ออิสระ (caster wheels) อีก 4 ล้อ (ด้านหน้าและหลัง) เพื่อเสริมความมั่นคง ดังรูปที่ 3.4 สามารถเคลื่อนที่ไปข้างหน้า ถอยหลัง และหมุนรอบตัวเองได้ในแนวราบอย่าง

อิสระ ตัวรถมีขนาดกว้าง 40 ซม., ยาว 89 ซม., สูง 25 ซม. น้ำหนักรวมประมาณ 20 kg. รองรับการบรรทุก (payload) ได้สูงสุด 150 kg.



รูปที่ 3.4 รถ AGV แบบอัตโนมัติ

ระบบขับเคลื่อนใช้ Brushless DC Motor (BLDC) รุ่น BLHM5100K-GFS ติดตั้งพร้อม Incremental Encoder ดังรูปที่ 3.5 โดยมีรายละเอียดดังตารางที่ 3.2 เพื่อวัดความเร็วเชิงมุมและระยะทางสะสมสำหรับการคำนวณ odometry และ sensor fusion.

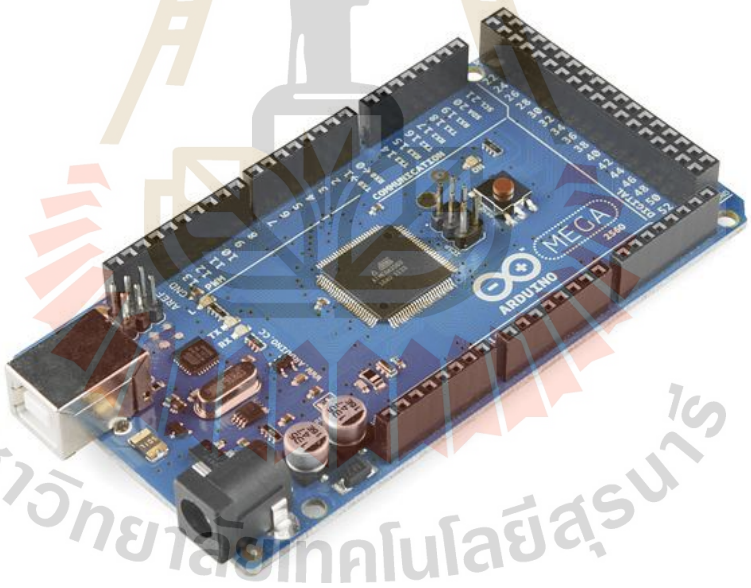
รูปที่ 3.5 BLDC Motor ที่นำมาใช้งาน

ตารางที่ 3.2 รายละเอียดข้อมูลของ Motor

รายการ	**รายละเอียด**
รุ่น	BLHM5100K-GFS
แรงดันไฟฟ้า	24V
กำลังไฟฟ้า	100W
ความละเอียด Encoder	900PPR

ระบบควบคุมถูกแบ่งออกเป็น 2 ระดับหลัก ได้แก่

1) หน่วยควบคุมระดับล่าง (Low-level Control): ใช้ Arduino Mega 2560 R3 ดังรูปที่ 3.6 ในการสร้างสัญญาณ PWM ควบคุมความเร็วรอบมอเตอร์และอ่านค่าจาก Encoder เพื่อควบคุมความเร็วอย่างแม่นยำ ซึ่งจะมีรายละเอียดดังตารางที่ 3.3



รูปที่ 3.6 บอร์ด Arduino Mega 2560 R3

ตารางที่ 3.3 รายละเอียดข้อมูลของ Arduino Mega 2560 R3

รายการ	**รายละเอียด**
ไมโครคอนโทรลเลอร์	ATmega2560
แรงดันทำงาน	5V
แรงดันอินพุต (แนะนำ)	7–12V
จำนวนขา I/O	54 Digital I/O (15 PWM), 16 Analog In
หน่วยความจำแฟลช	256 KB
SRAM	8 KB
EEPROM	4 KB
ความถี่นาฬิกา	16 MHz

2) หน่วยประมวลผลระดับกลาง (Mid-level Processing) ใช้ NVIDIA Jetson Nano ดังรูปที่ 3.7 ทำหน้าที่เชื่อมต่อกับ IMU และ Encoder บนตัวรถ และส่งข้อมูลผ่านโปรโตคอล MQTT ไปยังหน่วยประมวลผลกลาง (Laptop) เพื่อทำการควบคุมแบบรวมศูนย์ ซึ่งจะมีรายละเอียดดังตารางที่ 3.4

รูปที่ 3.7 บอร์ด Nvidia Jetson Nano

ตารางที่ 3.4 รายละเอียดข้อมูลบอร์ด Nvidia Jetson Nano

รายการ	รายละเอียด
หน่วยประมวลผล	Quad-core ARM Cortex-A57 CPU
GPU	128-core Maxwell GPU
หน่วยความจำ	4 GB LPDDR4
พอร์ตเชื่อมต่อ	GPIO, I2C, UART, USB 3.0, CSI Camera
ระบบปฏิบัติการ	Ubuntu 20.04
การสื่อสาร	รองรับการเชื่อมต่อ Wi-Fi ด้วย USB Wi-Fi Dongle
แหล่งจ่ายไฟ	5V (ผ่าน Micro-USB หรือ Barrel Jack)

สำหรับเซนเซอร์ภายใน (Onboard Sensors) ที่ติดตั้งบน AGV และแหล่งพลังงาน ประกอบด้วย

1) IMU รุ่น Hfi-A9: มีทั้ง accelerometer, gyroscope และ magnetometer ภายในโมดูลเดียวดังรูปที่ 3.8 รองรับการวัดทิศทางและการหมุนของตัวรถ (heading) เพื่อช่วยในการคำนวณ trajectory ซึ่งจะมีรายละเอียดดังตารางที่ 3.5

รูปที่ 3.8 IMU รุ่น HFI-A9

ตารางที่ 3.5 รายละเอียดข้อมูลของ IMU รุ่น HFI-A9

รายการ	รายละเอียด
ประเภทเซนเซอร์	9-DOF IMU (Accelerometer, Gyroscope, Magnetometer)
ความละเอียดของ Gyro	±2000°/s
ความละเอียดของ Accel	±8g
ความละเอียดของ Magnetometer	±4800μT
อัตราการส่งข้อมูล	สูงสุด ~100Hz
การเชื่อมต่อ	UART / I2C
ขนาด	25 × 16 mm

2) แบตเตอรี่ลิเธียม 12V 7Ah ดังรูปที่ 3.9 จำนวน 4 ก้อนซึ่งต่ออนุกรมเป็น 24V จำนวน 2 ชุด ซึ่งใช้เป็นแหล่งพลังงานสำหรับระบบมอเตอร์และอิเล็กทรอนิกส์

รูปที่ 3.9 Battery แหล่งพลังงาของ AGV

ระบบทั้งหมดนี้เชื่อมโยงเข้ากับสถาปัตยกรรม Centralized Control โดย Jetson Nano จะเป็นตัวกลางเชื่อมต่อระหว่างอุปกรณ์บน AGV และหน่วยประมวลผลกลาง (Laptop) ดังรูปที่ 3.10

ผ่านเครือข่าย Wi-Fi ทำให้สามารถส่งข้อมูลตำแหน่งและรับคำสั่งควบคุมการเคลื่อนที่ได้แบบ real-time ซึ่งรายละเอียดของ Laptop จะแสดงดังตารางที่ 3.6

รูปที่ 3.10 Laptop รุ่น MSI Raider GE68 HX 13V

ตารางที่ 3.6 รายละเอียดข้อมูลของ Laptop สำหรับระบบประมวลผลกลาง

รายการ	**รายละเอียด**
รุ่น	MSI Raider GE68 HX 13V
หน่วยประมวลผล	Intel Core i9 Gen 12 (24-core architecture)
หน่วยความจำหลัก (RAM)	32 GB DDR5
หน่วยประมวลผลกราฟิก (GPU)	NVIDIA GeForce RTX 4070
ระบบปฏิบัติการ	Windows 11
การเชื่อมต่อเครือข่าย	Wi-Fi 6 / Ethernet
โปรโตคอลการสื่อสาร	MQTT (Message Queuing Telemetry Transport)

3.2.3 หน่วยประมวลผลกลาง (Centralized Processing Unit)

ในการวิจัยนี้ได้มีการออกแบบระบบควบคุมยานพาหนะภาคพื้นดินอัตโนมัติ (AGV) ให้ทำงานภายใต้สถาปัตยกรรม ควบคุมแบบรวมศูนย์ (Centralized Control) หมายถึงการนำเอากระบวนการประมวลผลหลักของระบบ ไม่ว่าจะเป็นการตรวจจับวัตถุ การระบุตำแหน่ง การสร้างแผนที่ การวางแผนเส้นทาง และการควบคุมการเคลื่อนที่ มารวมไว้ที่อุปกรณ์คอมพิวเตอร์ภายนอกแทนที่จะกระจายการทำงานไปยังอุปกรณ์บนตัวรถ

หน่วยประมวลผลกลางที่เลือกใช้ คือ Laptop รุ่น MSI Raider GE68 HX 13V ซึ่งมีสมรรถนะเพียงพอสำหรับการรันโมเดลเชิงลึกอย่าง YOLOv11 และอัลกอริทึมคณิตศาสตร์ที่เกี่ยวข้องได้แบบเรียลไทม์ นอกจากนี้ ยังรองรับการเชื่อมต่อกับกล้องวงจรปิดหลายตัวผ่านเครือข่าย Wi-Fi และสามารถสื่อสารกับ AGV ได้อย่างต่อเนื่องผ่านโปรโตคอล MQTT

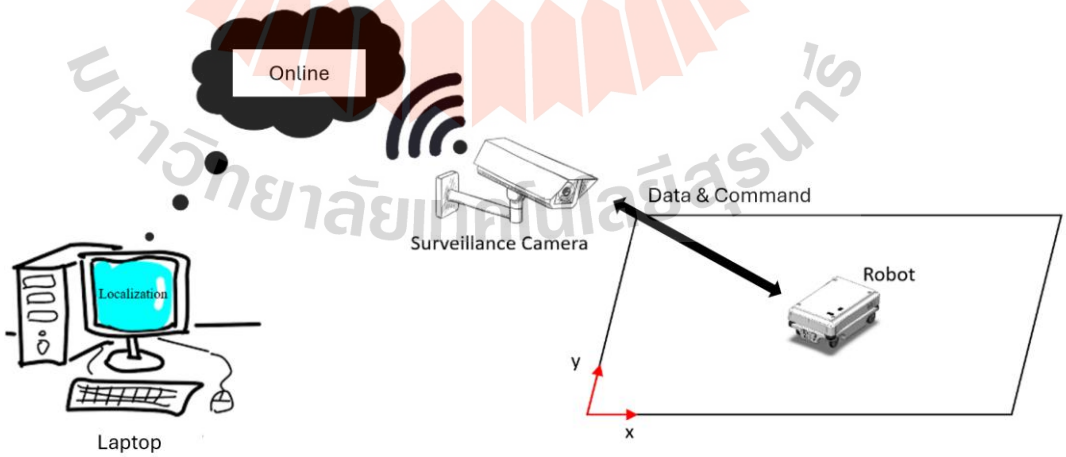
ในบริบทของสถาปัตยกรรมนี้ หน่วยประมวลผลกลางทำหน้าที่หลัก 3 ด้าน ได้แก่:

1) Perception (การรับรู้) รับสัญญาณภาพจากกล้องวงจรปิดหลายตัวผ่าน RTSP stream และประมวลผลด้วย YOLOv11 เพื่อตรวจจับ AGV และสิ่งกีดขวาง จากนั้นใช้ PnP และ GPR ในการประมาณตำแหน่งเชิงพิกัดจริงให้มีความแม่นยำสูงขึ้น

2) Planning (การวางแผน) รวมข้อมูลจากกล้องทั้งหมดเพื่อสร้าง Virtual Map ของพื้นที่ และใช้ A* Algorithm ในการหาทางเดินที่เหมาะสม พร้อมอัปเดตเส้นทางใหม่แบบ dynamic หากมีสิ่งกีดขวางเกิดขึ้น

3) Control (การควบคุม) ส่งคำสั่งควบคุมการเคลื่อนที่ไปยัง AGV ผ่าน MQTT โดยมี Jetson Nano เป็น node ตัวกลาง และ Arduino Mega ทำหน้าที่ขับเคลื่อนมอเตอร์ นอกจากนี้ยังรับข้อมูล feedback จาก IMU และ Encoder เพื่อนำมาใช้ใน Pure Pursuit Control และ Sensor Fusion

ด้วยสถาปัตยกรรมดังกล่าวดังรูปที่ 3.11 ระบบสามารถควบคุม AGV หลายคันพร้อมกัน ได้จากศูนย์กลาง และยังสามารถขยายไปยังพื้นที่ขนาดใหญ่โดยใช้กล้องวงจรปิดที่มีอยู่แล้วในโครงสร้างพื้นฐานของอาคาร (Infrastructure-based Sensor Network) โดยไม่จำเป็นต้องเพิ่มภาระการประมวลผลบนตัวรถเอง



รูปที่ 3.11 ระบบประมวลผลกลางของระบบ AGV

3.2.4 สภาพแวดล้อมจำลอง (Indoor Test Field)

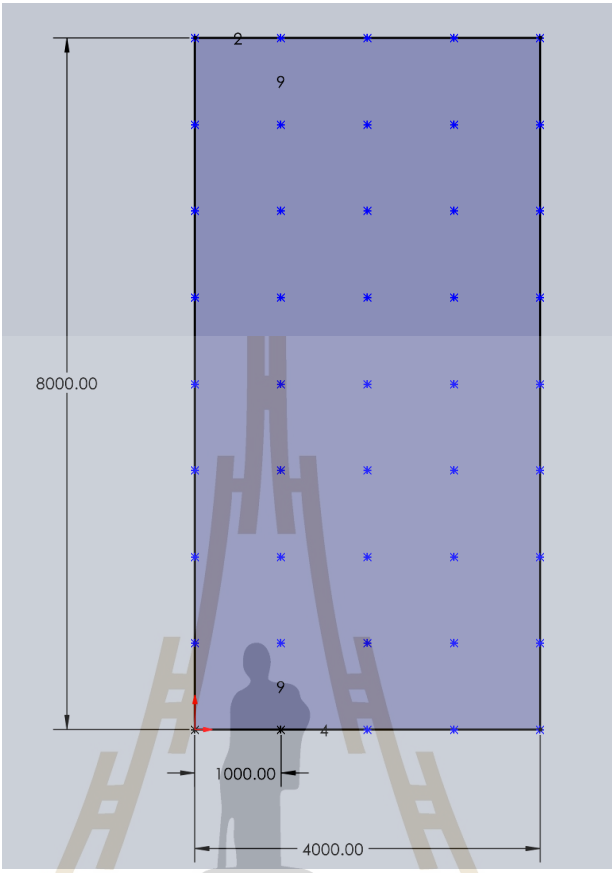
เพื่อทดสอบระบบนำทางอัตโนมัติที่พัฒนาขึ้น ได้มีการสร้างสภาพแวดล้อมจำลองภายในอาคาร โดยออกแบบให้ใกล้เคียงกับสภาพการใช้งานจริง เช่น คลังสินค้า ห้องโถงสำนักงาน หรือทางเดินในโรงพยาบาล สภาพแวดล้อมนี้ประกอบด้วย:

1) พื้นที่การทดสอบ (Test Area) ซึ่งกำหนดพื้นที่ขนาด 8×4 ตารางเมตร ภายในห้องทดลองดังรูปที่ 3.12 และแสดงเป็นแผนภาพขนาดดังรูป 3.13 โดยแบ่งพื้นที่เป็นเส้นทางหลักและอาจจะมีวัตถุสำหรับสิ่งกีดขวางตามสถานการณ์โซนวางสิ่งกีดขวาง (obstacles)

2) สิ่งกีดขวาง (Obstacles) จะใช้วัสดุจำลองภายในอาคารเพื่อสร้างสภาวะแวดล้อมแบบ dynamic ที่มีการเปลี่ยนแปลงตำแหน่งของสิ่งกีดขวางดังรูป 3.14 ได้แก่

1) เก้าอี้
2) ลังกระดาษ
3) พัดลม
4) ถังขยะ

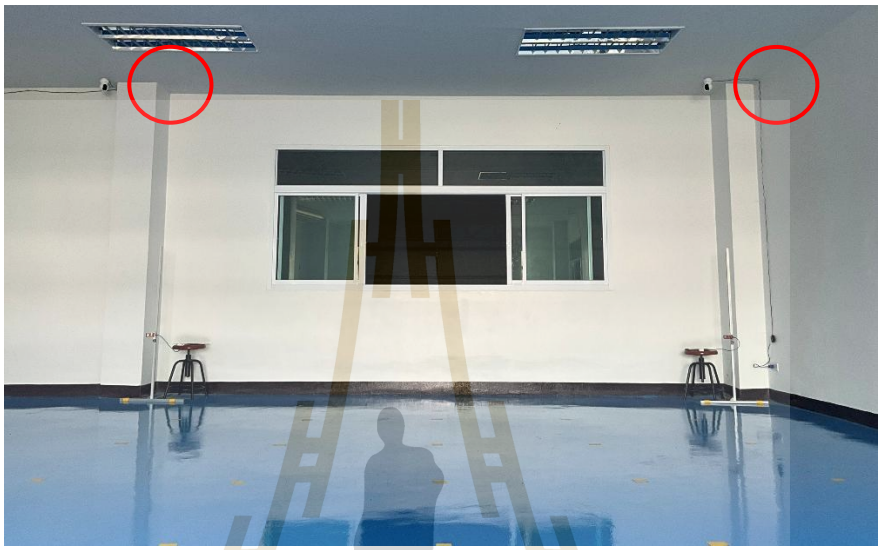
รูปที่ 3.12 พื้นที่ทดสอบ



รูปที่ 3.13 แผนภาพในพื้นที่ขนาด 8x4 ตารางเมตร

รูปที่ 3.14 อุปสรรคที่นำมาใช้ในพื้นที่

3) การติดตั้งกล้องวงจรปิดโดยกล้องถูกติดตั้งตามมุมห้องหรือบริเวณเพดานดังรูปที่ 3.15 ให้ครอบคลุมพื้นที่ทดสอบทั้งหมด โดยจัดตำแหน่งให้มุมมองของกล้องมีการ overlap กันบางส่วน เพื่อรองรับการทำ multi-camera fusion



รูปที่ 3.15 ตำแหน่งติดตั้งกล้องวงจรปิด

4) พื้นที่เป้าหมาย (Goal Points) มีการกำหนดจุดเริ่มต้น (start point) และจุดหมาย (goal points) หลายตำแหน่งตามผู้ทดลองต้องการเพื่อใช้ในการทดสอบระบบ path planning และการเปลี่ยนเส้นทางอัตโนมัติ

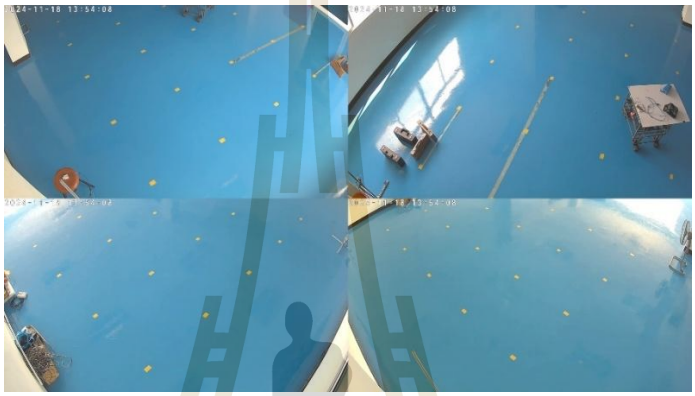
การสร้างสภาพแวดล้อมดังกล่าวช่วยให้สามารถประเมินประสิทธิภาพของระบบนำทางได้ทั้งในกรณี static (สิ่งกีดขวางอยู่กับที่) และ dynamic (สิ่งกีดขวางเคลื่อนที่หรือเปลี่ยนตำแหน่ง) รวมถึงการทดสอบความสามารถของระบบในการทำงานต่อเนื่องภายใต้สภาพแวดล้อมที่ซับซ้อนใกล้เคียงการใช้งานจริง

3.2.5 แผนที่เสมือน (Virtual Mapping)

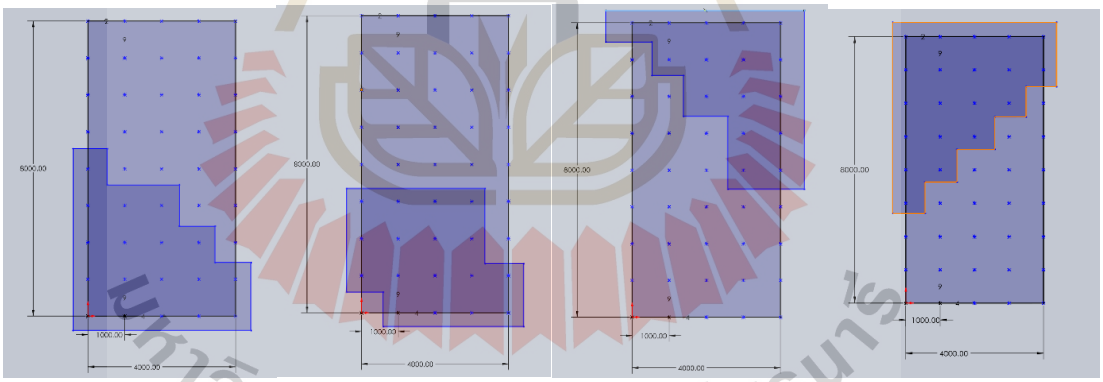
เมื่อกล้องวงจรปิดแต่ละตัวตรวจจับวัตถุหรือ AGV ในขอบเขต FOV ของตนเอง ระบบจะทำการระบุตำแหน่งเชิงพิกัดของวัตถุนั้นในโลกจริงด้วยเทคนิค PnP และนำผลลัพธ์ไปผ่านกระบวนการ GPR เพื่อปรับปรุงความแม่นยำ จากนั้นตำแหน่งจากกล้องทุกตัวจะถูกหลอมรวมเข้าด้วยกันเพื่อสร้าง แผนที่จำลองของพื้นที่ (Virtual Map) แบบเรียลไทม์ ซึ่งสามารถอัปเดตได้อย่างต่อเนื่องตามการเคลื่อนที่ของ AGV และการเปลี่ยนแปลงของสิ่งกีดขวาง ทำให้ระบบสามารถวางแผน

เส้นทางและตัดสินใจการเคลื่อนที่ได้อย่างเหมาะสมแม้ในสภาพแวดล้อมที่มีการเปลี่ยนแปลงตลอดเวลา

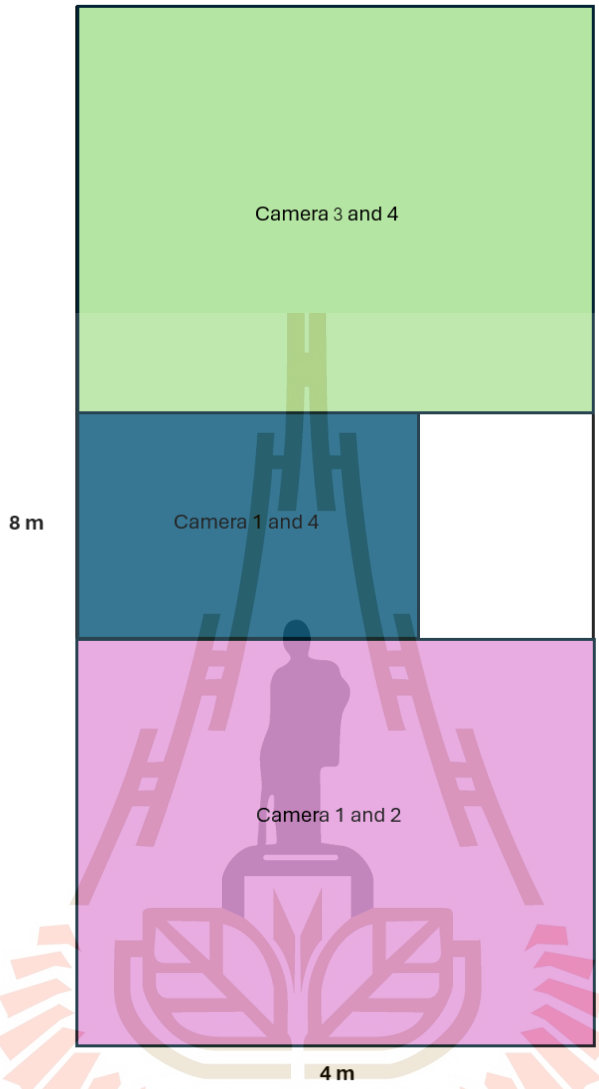
จากมุมมองของกล้องทั้ง 4 ดังรูปที่ 3.16 จะครอบคลุมพื้นที่ไม่เท่ากันในซึ่งจะแสดงในรูปที่ 3.17 ซึ่งจะมีจุดบอร์ดอยู่ 1 ช่วงที่ไม่มีกล้องใด ๆ สามารถมองเห็นได้ โดยจากข้อจำกัดดังกล่าวสามารถสร้างแผนที่เสมือนสำหรับกล้องวงจรปิดได้ตามพื้นที่ดังรูปที่ 3.18



รูปที่ 3.16 FOV ที่ได้จากกล้องทั้ง 4 ตัว



รูปที่ 3.17 พื้นที่คลอบคลุมจาก FOV ของกล้องแต่ละตัว



รูปที่ 3.18 พื้นที่จำลองแผนที่เสมือนจากทุกกล้อง

3.3 ขั้นตอนการพัฒนาระบบ (System Design and Development Process)

ระบบที่พัฒนาขึ้นในงานวิจัยนี้ประกอบด้วยองค์ประกอบหลายส่วนที่ทำงานร่วมกันแบบบูรณาการภายใต้แนวคิดการควบคุมแบบรวมศูนย์ (Centralized Control) โดยมีหน่วยประมวลผลกลางทำหน้าที่ควบคุมการทำงานทั้งหมด ตั้งแต่การรับข้อมูล การประมวลผล ไปจนถึงการตัดสินใจควบคุมการเคลื่อนที่ของยานพาหนะอัตโนมัติ (AGV) โดยกระบวนการทำงานของระบบมีลำดับดังนี้:

1) รับภาพจากกล้องวงจรปิด (Surveillance Feed)

1) กล้อง IP Camera (Tapo C200) ทำการสตรีมภาพแบบ Real-Time จาก 4 มุมมองในพื้นที่

2) ภาพถูกส่งไปยังหน่วยประมวลผลกลางผ่าน RTSP Protocol

2) ประมวลผลภาพด้วย YOLOv11

1) ใช้ YOLOv11 ตรวจจับ AGV และ Obstacle บนภาพ
2) ผลลัพธ์ คือ Bounding Box ของวัตถุ ซึ่งถูกนำไปใช้เป็นจุดอ้างอิงสำหรับการคำนวณตำแหน่ง
3) ในการทดลองนี้ใช้ centroid ของ Bounding Box เป็นจุดอินพุตสำหรับ PnP

3) ประมาณค่าตำแหน่งด้วย (PnP+GPR)

1) ใช้จุดที่ได้จาก Bounding Box เปรียบเทียบกับข้อมูล 3D ของ AGV (เช่น มุมของตัวรถหรือจุด marker) เพื่อตีความตำแหน่งในระบบพิกัดโลก (World Coordinate System)
2) การแก้ PnP ใช้ อย่างน้อย 4 จุดอ้างอิง ($n \geq 4$) สำหรับ Calibrate เพื่อคำนวณพิกัด AGV
3) ใช้ GPR ปรับปรุงค่าตำแหน่ง เพื่อลดความคลาดเคลื่อนจาก distortion และ noise ของกล้อง

4) สร้างแผนที่เสมือน (Virtual Mapping)

1) รวมตำแหน่งจากกล้องทุกตัวเข้าด้วยกันเพื่อสร้าง Grid-based Map
2) อัปเดตตำแหน่ง AGV และ Obstacle แบบ Real-Time เพื่อรองรับสภาพแวดล้อมที่มีการเปลี่ยนแปลง

5) วางแผนเส้นทางอัตโนมัติ (Path Planning)

1) ใช้ A* Algorithm ในการสร้างเส้นทางที่สั้นและปลอดภัยที่สุดจากจุดเริ่มต้นไปยังเป้าหมาย
2) ระบบสามารถ Re-plan เส้นทางใหม่เมื่อพบ Obstacle เพิ่มเติม

6) ส่งคำสั่งควบคุมไปยัง AGV

1) คำสั่งความเร็วและทิศทางถูกส่งจากหน่วยประมวลผลกลางผ่าน MQTT ไปยัง Jetson Nano
2) Jetson Nano ส่งต่อคำสั่งไปยัง Arduino Mega 2560 ผ่าน UART เพื่อควบคุมมอเตอร์

7) ติดตามเส้นทางด้วย Pure Pursuit

1) ระบบใช้ค่าตำแหน่งจากกล้องและทิศทางจาก IMU เพื่อปรับการเคลื่อนที่ของ AGV ให้ติดตามเส้นทางได้อย่างราบรื่น
2) Pure Pursuit รองรับการปรับ trajectory แบบ dynamic เมื่อมีการเปลี่ยนแปลงสิ่งกีดขวาง

กระบวนการทั้งหมดนี้ทำงานร่วมกันภายใต้สถาปัตยกรรมควบคุมแบบรวมศูนย์ โดยผลลัพธ์ของแต่ละโมดูล (การตรวจจับ การระบุตำแหน่ง การสร้างแผนที่ การวางเส้นทาง และการควบคุม) จะถูกประเมินในบทถัดไป ด้วยตัวชี้วัด ได้แก่ mAP สำหรับ Object Detection และ Localization Error สำหรับการระบุตำแหน่งของ AGV

3.4 การประเมินผล (Evaluation Metrics)

เพื่อวัดประสิทธิภาพของระบบนำทางอัตโนมัติที่พัฒนาขึ้น งานวิจัยนี้ได้กำหนดตัวชี้วัด (metrics) สำหรับแต่ละโมดูลหลักของระบบ ได้แก่ การตรวจจับวัตถุ (Object Detection), การระบุตำแหน่ง (Localization) และ การควบคุมแบบไดนามิก (Dynamic Control) ดังนี้:

3.4.1 การตรวจจับวัตถุ (Object Detection Metrics)

1) Precision (ความแม่นยำ)

แสดงสัดส่วนของวัตถุที่ระบบตรวจจับได้ถูกต้อง (True Positives) ต่อวัตถุที่ระบบตรวจจับทั้งหมด (รวม False Positives) จะแสดงดังสมการที่ (3.1)

$$Pre = \frac{TP}{TP+FP} \tag{3.1}$$

2) Recall (ความครอบคลุม)

แสดงความสามารถในการตรวจจับวัตถุทั้งหมดที่มีอยู่จริง จะแสดงดังสมการที่ (3.2)

$$Re = \frac{TP}{TP+FN} \tag{3.2}$$

3) F1 Score

เป็นค่ากลางเชิงฮาร์มอนิก (harmonic mean) ของ *Precision* และ *Recall* เพื่อสะท้อนสมดุลระหว่างสองค่า จะแสดงดังสมการที่ (3.3)

$$F1 = 2x\frac{(Precision)(Recall)}{Precision+Recall} \tag{3.3}$$

4) Mean Average Precision (mAP)

โดยที่ AP_i คือ ค่า Area under Precision–Recall Curve ของแต่ละคลาสวัตถุ และ N คือ จำนวนคลาสทั้งหมด จะแสดงดังสมการที่ (3.4)

$$mAP = \frac{1}{N}\sum_{i=1}^{n} AP_i \quad (3.4)$$

3.4.2 การระบุตำแหน่ง (Localization Metrics)

เพื่อประเมินความแม่นยำของการแปลงพิกัดจากภาพ (image space) สู่พิกัดโลก (world coordinate)

1) Mean Absolute Error (MAE)

เป็นค่าเฉลี่ยของความคลาดเคลื่อนเชิงตำแหน่งระหว่างค่าจริง (y_i) และค่าที่ระบบประมาณได้ ($\hat{y}_i$) จะแสดงดังสมการที่ (3.5)

$$MAE = \frac{1}{N}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (3.5)$$

2) Root Mean Square Error (RMSE)

เป็นตัวชี้วัดที่ให้น้ำหนักกับความคลาดเคลื่อนที่มีค่ามาก ทำให้สะท้อน error ที่รุนแรงได้ชัดเจนกว่าค่า MAE จะแสดงดังสมการที่ (3.6)

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (3.6)$$

3.4.3 การควบคุมแบบไดนามิก (Dynamic Control Metrics)

1) Tracking Error (TE)

เป็นระยะห่างระหว่างตำแหน่งเป้าหมายบนเส้นทางอ้างอิง (x_{ref}, y_{ref}) และตำแหน่งจริงของ AGV(x, y) ในเวลา t จะแสดงดังสมการที่ 3.7

$$TE(t) = \sqrt{(x_{ref}(t) - x(t))^2 + (y_{ref}(t) - y(t))^2} \quad (3.7)$$

2) Control Latency

เป็นค่าความหน่วงเวลาระหว่างที่หน่วยประมวลผลกลางส่งคำสั่ง (t_{cmd}) และเวลาที่ *AGV* ตอบสนองจริง (t_{act}) จะแสดงดังสมการที่ (3.8)

$$L_a = t_{act} - t_{cmd} \quad (3.8)$$

3.5 การออกแบบการทดลอง (Experimental Design)

งานวิจัยนี้มุ่งเน้นการพัฒนาระบบระบุตำแหน่งและควบคุมการเคลื่อนที่ของยานพาหนะภาคพื้นดินอัตโนมัติ (AGV) โดยอาศัยกล้องวงจรปิดร่วมกับปัญญาประดิษฐ์และการประมวลผลแบบรวมศูนย์ (Centralized Processing) โดยครอบคลุมกระบวนการตั้งแต่การสอบเทียบกล้อง การตรวจจับวัตถุ การระบุตำแหน่ง การวางเส้นทาง การควบคุมการเคลื่อนที่ ไปจนถึงการวัดและประเมินผลของระบบที่พัฒนาขึ้นในสภาพแวดล้อมจริง

1) วัตถุประสงค์ของการทดลอง

การทดลองถูกออกแบบเพื่อประเมินประสิทธิภาพของระบบในประเด็นหลักดังต่อไปนี้:

1) ความแม่นยำของระบบการระบุตำแหน่งของ AGV โดยใช้กล้องวงจรปิด + YOLOv11 + PnP + GPR

2) ความสามารถของระบบในการตรวจจับและหลีกเลี่ยงสิ่งกีดขวางโดยใช้ AI

3) ความเสถียรและความถูกต้องของระบบควบคุมการเคลื่อนที่ด้วย Pure Pursuit Controller

4) ประสิทธิภาพโดยรวมของระบบในการนำทาง AGV ไปยังตำแหน่งเป้าหมายภายในพื้นที่ทดสอบ

2) ตัวแปรในการทดลอง

ตารางที่ 3.7 ตัวแปรสำหรับการทดลอง

ประเภทตัวแปร	รายละเอียด
ตัวแปรต้น	ความเร็วของ *AGV*, จำนวนวัตถุในพื้นที่
ตัวแปรตาม	ความแม่นยำของตำแหน่ง, เวลาที่ใช้ในการเดินทาง
ตัวแปรควบคุม	ขนาดพื้นที่ทดลอง, จำนวนกล้อง, ความละเอียดของภาพ, อัตราอัพเดตภาพ (*10 Hz*)

3) การสอบเทียบระบบการมองเห็น (Vision Calibration)

เพื่อให้การระบุตำแหน่งจากภาพกล้องวงจรปิดมีความแม่นยำ จำเป็นต้องดำเนินการสอบเทียบทั้งในระดับกล้องเดี่ยวและระบบรวม ดังนี้:

1) Intrinsic Calibration: ใช้วิธี Chessboard Calibration เพื่อหาค่าพารามิเตอร์ภายในกล้อง ได้แก่ focal length, optical center และ lens distortion โดยใช้ OpenCV

2) Extrinsic Calibration: ใช้ Perspective Transformation Calibration โดยระบุจุดที่ทราบพิกัดจริงบนพื้นที่ทดสอบ เพื่อวางตำแหน่งกล้องลงใน World Coordinate System
3) Internal Scale Factor: คำนวณการแปลงจาก pixel space ไปยัง physical space ด้วย PnP โดยใช้จุดอ้างอิงจริงในพื้นที่
4) Polynomial Fitting: ใช้สมการ Polynomial Surface Order 2 เพื่อปรับ Scale Factor ให้สอดคล้องกับ Distortion ของเลนส์กล้อง
5) Gaussian Process Regression : ใช้เพื่อแก้ไขค่าความคลาดเคลื่อนที่เกิดจาก noise และ Occlusion
6) Multi-camera Alignment: รวมข้อมูลจากกล้องหลายตัวเข้าสู่แผนที่เสมือน (Virtual Map)

4) การฝึกและทดสอบโมเดล AI (YOLOv11 Custom Training)

1) Dataset: เก็บภาพจากกล้องวงจรปิดในพื้นที่ทดลอง โดยจัดวางวัตถุที่เกี่ยวข้อง ได้แก่ AGV, กล่องกระดาษ, เก้าอี้, ถังขยะ และพัดลม
2) Annotation: ใช้เครื่องมือ Roboflow ในการทำ bounding box
3) การแบ่งชุดข้อมูล: Training 80%, Validation 20%
4) การฝึกโมเดล: ใช้ PyTorch + Ultralytics YOLOv11 โดยกำหนด hyperparameters ได้แก่ learning rate, batch size, epoch
5) การประเมินผล: ใช้ Metrics ได้แก่ mAP, Precision, Recall, Confusion Matrix

5) การประมวลผลตำแหน่ง (Localization Pipeline)

1) YOLOv11 Detection: หาตำแหน่ง Bounding Box ของ AGV
2) PnP + GPR: คำนวณตำแหน่ง 3 มิติ และปรับปรุงความแม่นยำ
3) Multi-View Fusion: รวมผลจากหลายกล้อง โดยใช้ confidence weighting และการเทียบเชิงเรขาคณิต

6) การสร้างแผนที่และการวางเส้นทาง (Mapping and Planning)

1) ใช้ Grid-based Virtual Map ที่ได้จาก Fusion ของกล้องหลายตัว
2) ใช้ A* Algorithm ค้นหาเส้นทางที่ปลอดภัยและสั้นที่สุด
3) หากมี Obstacle แบบ dynamic เข้ามา ระบบจะ Replan เส้นทางแบบ Semi-Real-Time

7) การควบคุมการเคลื่อนที่ (Motion Control via MQTT)

1) Laptop ส่งคำสั่งผ่าน MQTT ไปยัง Jetson Nano

2) Jetson Nano ส่งผ่าน UART ไปยัง Arduino Mega
3) Arduino Mega ควบคุมมอเตอร์ BLDC + อ่านค่า Encoder
4) ใช้ Pure Pursuit Algorithm สำหรับ Trajectory Tracking โดยใช้ข้อมูล Heading จาก IMU

8) การวิเคราะห์ผลลัพธ์

ผลลัพธ์จากแต่ละการทดลองจะถูกเก็บเป็นเชิงปริมาณ (Quantitative Data) เช่น ค่า RMSE, F1 Score และ Tracking Error พร้อมทั้งทำการทดลองซ้ำ (3 รอบต่อ Scenario) เพื่อความน่าเชื่อถือ ผลที่ได้จะถูกวิเคราะห์หาค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐาน และนำไปเปรียบเทียบกับสมมติฐานของงานวิจัยเพื่อยืนยันความถูกต้อง

บทที่ 4
ผลการวิเคราะห์ข้อมูลและอภิปรายผล

ในบทนี้ ผู้วิจัยนำเสนอผลการทดลองและการวิเคราะห์ข้อมูลตามกรอบการออกแบบที่อธิบายไว้ในบทที่ 3 เพื่อพิสูจน์สมมติฐานของงานวิจัย ระบบที่พัฒนาขึ้นถูกทดสอบในหลายองค์ประกอบ ได้แก่ การสอบเทียบกล้อง (Vision Calibration), การตรวจจับวัตถุด้วย AI, การระบุตำแหน่ง (Localization), การสร้างแผนที่และวางแผนเส้นทาง (Mapping & Path Planning), และการควบคุมการเคลื่อนที่ (Motion Control) โดยผลลัพธ์ทั้งหมดจะถูกประเมินด้วย Metrics ที่กำหนดไว้

4.1 ผลการสอบเทียบกล้องและระบบการมองเห็น

เพื่อให้ระบบสามารถแปลงข้อมูลจากภาพกล้องวงจรปิดไปเป็นตำแหน่งพิกัดในโลกจริงได้อย่างแม่นยำ จำเป็นต้องดำเนินการ สอบเทียบกล้อง (Camera Calibration) สำหรับกล้องแต่ละตัวที่ใช้ในระบบ โดยการสอบเทียบนี้แบ่งออกเป็นสองขั้นตอนหลัก ได้แก่ การสอบเทียบค่าภายในกล้อง (Intrinsic Calibration) และ การสอบเทียบค่าภายนอกกล้อง (Extrinsic Calibration)

4.1.1 การสอบเทียบค่าภายในกล้อง (Intrinsic Calibration)

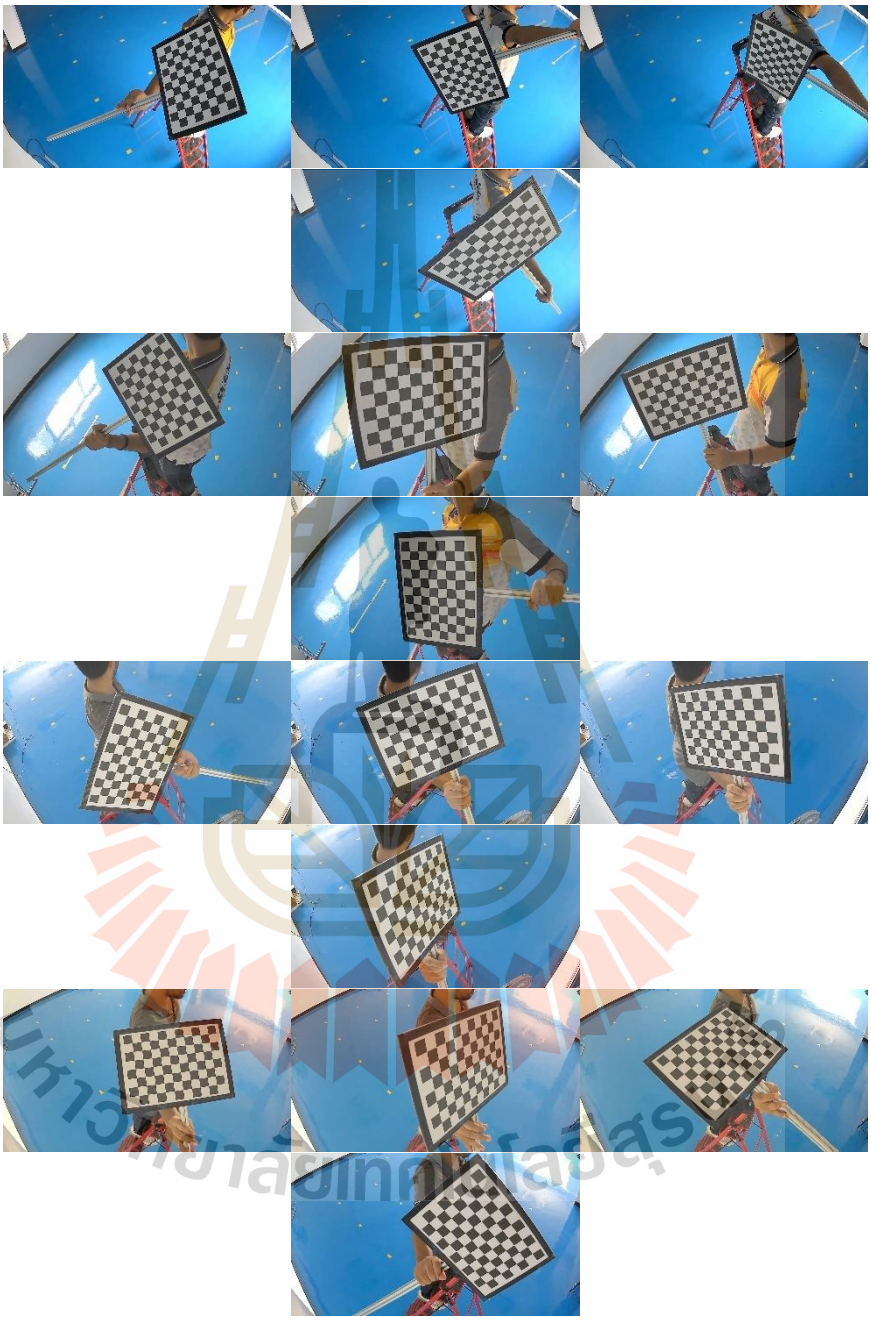
การสอบเทียบ Intrinsic Parameters ทำด้วยวิธี Chessboard Calibration โดยใช้บอร์ดลายตารางหมากรุกมาตรฐาน ซึ่งเป็นเทคนิคที่ได้รับการยอมรับอย่างแพร่หลายในการหาค่าพารามิเตอร์ภายในของกล้อง (Camera Intrinsics) ประกอบด้วย:

1) ค่าโฟกัส (Focal Length: f_x, f_y)
2) จุดศูนย์กลางของภาพ (Principal Point: c_x, c_y)
3) ค่าสัมประสิทธิ์การบิดเบือนของเลนส์ (Lens Distortion Coefficients)

ในการทดลองนี้ กล้องแต่ละตัวถูกสอบเทียบแยกกัน โดยใช้ภาพ Chessboard ที่ถ่ายจากมุมมองต่าง ๆ รวม 15 รูปต่อกล้องดังตัวอย่างรูปที่ 4.1 เพื่อนำไปประมวลผลด้วย OpenCV Calibration Library ผลลัพธ์จากการสอบเทียบจะได้เป็น Intrinsic Matrix ของแต่ละกล้องในรูป ดังสมการที่ (4.1)

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 0 \end{bmatrix} \tag{4.1}$$

พร้อมค่าของ Distortion Coefficients (k_1, k_2, p_1, p_2, k_1) ที่ใช้สำหรับการแก้ไขความโค้ง (Radial/Tangential Distortion) ดังตารางที่ 4.1



รูปที่ 4.1 ตัวอย่างรูป Calibration เพื่อหา Intrinsic Parameter

ตารางที่ 4.1 ผลการสอบเทียบ Intrinsic ของกล้อง (Tapo C200)

กล้อง	f_x	f_y	c_x	c_y	Distortion Coefficients
Camera 1	1621.77	1624.00	981.12	634.44	[-0.35, 0.05, 0, 0, 0.39]
Camera 2	1621.77	1624.00	981.12	634.44	[-0.35, 0.05, 0, 0, 0.39]
Camera 3	1610.83	1611.67	1041.85	660.69	[-0.37, 0.12, 0, 0, 0.01]
Camera 4	1582.78	1583.45	946.10	582.10	[-0.35, -0.01, 0, 0, 0.11]

4.1.2 การสอบเทียบค่าภายนอกกล้อง (Extrinsic Calibration)

หลังจากได้ค่าพารามิเตอร์ภายในกล้อง (Intrinsic Parameters) แล้ว ขั้นตอนถัดมา คือ การแปลงพิกัดของวัตถุในภาพให้สัมพันธ์กับระบบพิกัดโลก (World Coordinate System) ซึ่งจำเป็นต้องใช้ค่าพารามิเตอร์ภายนอก (Extrinsic Parameters) ได้แก่:

1) Rotation Vector (Rvec): กำหนดทิศทางการหมุนของกล้องเมื่อเทียบกับแกนพิกัดโลก

2) Translation Vector (Tvec): กำหนดตำแหน่งการเลื่อน (offset) ของกล้องในระบบพิกัดโลก

การสอบเทียบ Extrinsic Parameters ทำโดยการเลือก จุดอ้างอิง (Reference Points) ในพื้นที่จริง ซึ่งเป็นจุดที่มีพิกัดที่ทราบแน่นอนในระบบพิกัดโลกจะเป็นเส้น grid ดังรูปที่ 4.2 และมองเห็นได้จากกล้องแต่ละตัว จากนั้นใช้ OpenCV Calibration Tools ประเมินค่า Rvec และ Tvec เพื่อหาความสัมพันธ์เชิงเรขาคณิตระหว่างระบบพิกัดของกล้อง (Camera Coordinate) และระบบพิกัดโลก (Global Coordinate) ผลลัพธ์ของการสอบเทียบดังตารางที่ 4.2 ซึ่งได้ดังสมการที่ (4.2)

$$X_c = RX_w + T \tag{4.2}$$

โดยที่:

X_w คือ จุดในระบบพิกัดโลก (World Coordinate)

X_c คือ จุดในระบบพิกัดกล้อง (Camera Coordinate)

R คือ Rotation Matrix (Rvec)

T คือ Translation Vector (Tvec)

ตารางที่ 4.2 ผลการสอบเทียบ Extrinsic ของกล้อง (Tapo C200)

กล้อง	Rvec (rad)	Tvec (mm)
Camera 1	[0.924,0.023, -0.136]	[0.460, -0.563, 0.803]
Camera 2	[1.104, 1.800, -1.167]	[-0.146, 2.609, 7.601]
Camera 3	[0.904, 1.893, -1.588]	[-3.701, 2.098, 10.451]
Camera 4	[0.781, -1.772, 1.767]	[5.509, 4.215, 7.446]

รูปที่ 4.2 รูปจากกล้องทั้ง 4 ตัวที่ใช้ทำการ Calibration เพื่อหา Extrinsic Parameter

4.1.3 การหาค่า Scale Factor

หลังจากได้ค่าพารามิเตอร์ภายในและภายนอกของกล้องแล้ว ขั้นตอนถัดมา คือการหาค่า Scale Factor (s) ซึ่งใช้ในการแปลงค่าพิกัดจากภาพ (Pixel Coordinate) ไปยังพิกัดจริงบนพื้น (World Coordinate) โดยค่า Scale Factor จะไม่คงที่ แต่เปลี่ยนแปลงตามตำแหน่งของจุดบนภาพและความลึก (Depth) ของวัตถุที่กล้องตรวจจับ ซึ่งอ้างอิงจาก Pinhole Camera Model ดังสมการที่ (4.3)

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A[R|T]\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} \tag{4.3}$$

โดยที่:

(u, v) คือ พิกัดของจุดในภาพ (Pixel Coordinates)

s คือ Scale Factor (ขึ้นกับ Depth ในพื้นที่)

A คือ Intrinsic Matrix ของกล้อง

$[R|T]$ คือ Extrinsic Parameters

(X_w, Y_w, Z_w) คือ พิกัดจริงในโลก (World Coordinates)

ในการใช้งานจริง ค่า Scale Factor (s) ของแต่ละจุดจะมีความแตกต่างกันไปตามตำแหน่งบนภาพและความลึกของวัตถุ ดังนั้นจึงจำเป็นต้องทำการเก็บ จุดอ้างอิง (Calibration Points) ที่ทราบพิกัดจริงบนพื้น (Ground Truth) และสามารถมองเห็นได้จากกล้องแต่ละตัว เพื่อนำมาใช้ในการสร้างความสัมพันธ์ระหว่าง Pixel Space และ World Space สำหรับการหาค่า Scale Factor อย่างเป็นระบบ

ในกระบวนการนี้ ได้ทำการเก็บข้อมูลพิกัดจุดอ้างอิงในแต่ละกล้อง โดยจำนวนจุดที่เก็บได้จากกล้องแต่ละตัวไม่เท่ากัน ขึ้นอยู่กับ Field of View (FOV) และตำแหน่งการติดตั้งของกล้อง โดยข้อมูลดังกล่าวถูกนำเสนอในรูปของตาราง 4.3 – 4.6 ซึ่งสรุปผลการเก็บ Calibration Points สำหรับ กล้องที่ 1 ถึงกล้องที่ 4 ตามลำดับ เพื่อใช้ในการคำนวณและประมาณค่า Scale Factor ต่อไป

ตารางที่ 4.3 Calibration Points ของกล้องที่ 1 จำนวน 17 จุด

No.	$\mathbf{u}$ (pixel)	$\mathbf{v}$ (pixel)	$\mathbf{X_w}$ (m)	$\mathbf{Y_w}$ (m)	$\mathbf{Z_w}$ (m)	Scale Factor	Mean	Error
1	837	973	0	0	3.172	-2.356		1.552
2	1229	737	1	0	3.418	-2.606		1.303
3	1532	539	2	0	3.912	-3.102		0.807
4	1744	384	3	0	4.574	-3.765		0.143
5	1886	275	4	0	5.343	-4.535		-0.627
6	624	629	0	1	3.655	-2.926		0.982
7	981	451	1	1	3.871	-3.145		0.763
8	1285	300	2	1	4.313	-3.590		0.319
9	1515	187	3	1	4.922	-4.201	-3.909	-0.292
10	1685	110	4	1	5.643	-4.923		-1.014
11	478	377	0	2	4.32	-3.677		0.231
12	791	237	1	2	4.504	-3.865		0.044
13	1074	119	2	2	4.889	-4.252		-0.344
14	1305	30	3	2	5.433	-4.798		-0.890
15	383	195	0	3	5.095	-4.538		-0.629
16	653	80	1	3	5.252	-4.698		-0.790
17	315	69	0	4	5.938	-5.466		-1.558

ตารางที่ 4.4 Calibration Points ของกล้องที่ 2 จำนวน 17 จุด

No.	$\mathbf{u}$ (pixel)	$\mathbf{v}$ (pixel)	$\mathbf{X_w}$ (m)	$\mathbf{Y_w}$ (m)	$\mathbf{Z_w}$ (m)	Scale Factor	Mean	Error
1	949	103	1	0	5.134	4.984		1.010
2	805	298	2	0	4.449	4.332		0.358
3	621	556	3	0	3.903	3.633		-0.341
4	406	885	4	0	3.56	2.866		-1.108
5	1317	22	0	1	5.676	5.358		1.384
6	1244	177	1	1	4.864	4.750		0.776
7	1136	393	2	1	4.135	4.113	3.974	0.139
8	995	688	3	1	3.54	3.430		-0.544
9	803	1069	4	1	3.158	2.676		-1.298
10	1584	97	0	2	5.614	5.054		1.080
11	1552	265	1	2	4.791	4.449		0.475
12	1504	501	2	2	4.049	3.816		-0.158

ตารางที่ 4.4 Calibration Points ของกล้องที่ 2 จำนวน 17 จุด (ต่อ)

No.	**u** (pixel)	**v** (pixel)	$\mathbf{X_w}$ (m)	$\mathbf{Y_w}$ (m)	$\mathbf{Z_w}$ (m)	Scale Factor	Error
13	1417	825	3	2	3.44	3.139	-0.835
14	1833	184	0	3	5.728	4.690	0.716
15	1839	364	1	3	4.925	4.078	0.104
16	1839	612	2	3	4.207	3.438	-0.536
17	1805	942	3	3	3.624	2.751	-1.223

ตารางที่ 4.5 Calibration Points ของกล้องที่ 3 จำนวน 15 จุด

No.	**u** (pixel)	**v** (pixel)	$\mathbf{X_w}$ (m)	$\mathbf{Y_w}$ (m)	$\mathbf{Z_w}$ (m)	Scale Factor	Mean	Error
1	111	133	4	4	5.769	3.812		1.041
2	448	92	3	5	5.156	4.024		1.252
3	199	250	4	5	4.942	3.324		0.553
4	873	79	2	6	4.887	4.169		1.398
5	616	226	3	6	4.441	3.514		0.743
6	330	414	4	6	4.191	2.822		0.051
7	1309	91	1	7	5.018	4.239		1.468
8	1107	229	2	7	4.362	3.624	2.771	0.853
9	847	411	3	7	3.854	2.981		0.210
10	531	643	4	7	3.564	2.296		-0.475
11	1681	129	0	8	5.521	4.238		1.467
12	1567	248	1	8	4.725	3.650		0.879
13	1393	419	2	8	4.021	3.044		0.273
14	1148	651	3	8	3.465	2.410		-0.361
15	818	947	4	8	3.138	1.732		-1.039

ตารางที่ 4.6 Calibration Points ของกล้องที่ 4 จำนวน 22 จุด

No.	$\mathbf{u}$ (pixel)	$\mathbf{v}$ (pixel)	$\mathbf{X_w}$ (m)	$\mathbf{Y_w}$ (m)	$\mathbf{Z_w}$ (m)	Scale Factor	Mean	Error
1	1698	264	0	4	6.064	4.517		0.292
2	1492	147	1	4	6.329	5.223		0.998
3	1297	51	2	4	6.733	5.926		1.701
4	1591	369	0	5	5.236	3.830		-0.395
5	1358	229	1	5	5.54	4.535		0.310
6	1149	121	2	5	5.998	5.237		1.012
7	969	30	3	5	6.577	5.937		1.711
8	1441	511	0	6	4.478	3.142		-1.084
9	1182	345	1	6	4.831	3.846		-0.380
10	960	213	2	6	5.349	4.547		0.321
11	781	109	3	6	5.991	5.245	4.225	1.020
12	641	30	4	6	6.721	5.942		1.716
13	1221	701	0	7	3.833	2.451		-1.775
14	948	494	1	7	4.24	3.154		-1.072
15	729	333	2	7	4.822	3.854		-0.372
16	563	209	3	7	5.526	4.551		0.325
17	440	119	4	7	6.31	5.247		1.021
18	924	944	0	8	3.367	1.757		-2.469
19	650	683	1	8	3.823	2.459		-1.767
20	460	480	2	8	4.46	3.157		-1.068
21	324	333	3	8	5.123	3.856		-0.370
22	230	224	4	8	6.038	4.549		0.323

จากผลการทำ Perspective Calibration พบว่าค่า Scale Factor ($\mathbf{s}$) ที่ได้จากแต่ละจุดอ้างอิงภายในมุมมองของกล้องมีความแตกต่างกันอย่างชัดเจน ซึ่งสะท้อนให้เห็นถึงข้อจำกัดของการใช้วิธีการดังกล่าวเพียงอย่างเดียวในการประมาณค่าตำแหน่งในโลกจริง การนำค่าเฉลี่ย (Mean Scale Factor) มาใช้แทน แม้จะช่วยลดความซับซ้อนในการคำนวณ แต่ก็ยังคงก่อให้เกิด ความคลาดเคลื่อนเชิงตำแหน่ง (Localization Error) ในระดับที่สูง โดยเฉพาะเมื่อมีการใช้งานจริงในพื้นที่ที่มีการเปลี่ยนแปลงของระยะและมุมมองกล้อง ดังนั้น เพื่อแก้ไขปัญหาความไม่สม่ำเสมอดังกล่าว งานวิจัยนี้จึงได้นำเสนอการปรับปรุงเพิ่มเติมด้วย Polynomial Surface Regression และ GPR ซึ่งจะช่วยลด

ความคลาดเคลื่อนในการประมาณค่าตำแหน่ง และเพิ่มความแม่นยำของระบบโดยรวมได้อย่างมีนัยสำคัญ

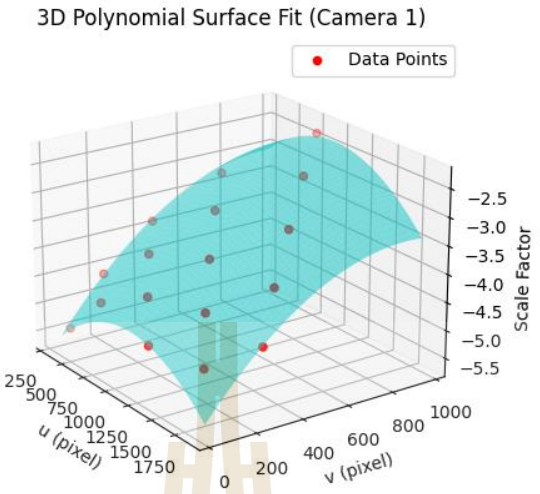
4.1.4 Polynomial Fitting สำหรับการประมาณค่า Scale Factor

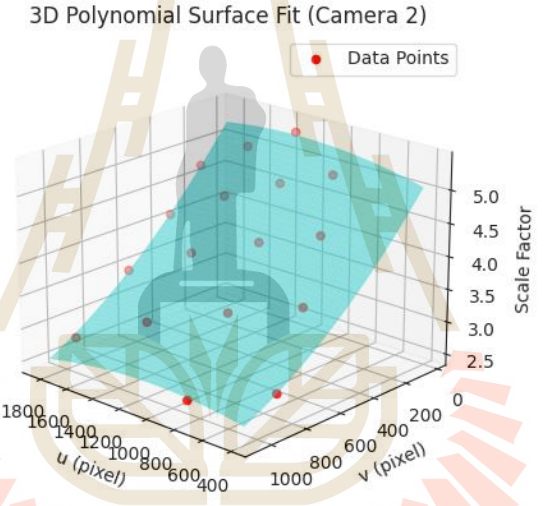
เนื่องจากความผิดเพี้ยนของกล้อง (Lens Distortion) มักมีลักษณะเป็น Quadratic (พาราโบลา) ดังนั้นการประมาณค่า Scale Factor มักใช้ สมการพหุนามลำดับที่ 2 (2nd Order Polynomial Regression) ซึ่งเพียงพอในการแก้ไขความโค้งงอของภาพ (Image Distortion) ดังสมการโดยผลที่ได้จะแสดงดังตารางที่ 4.7

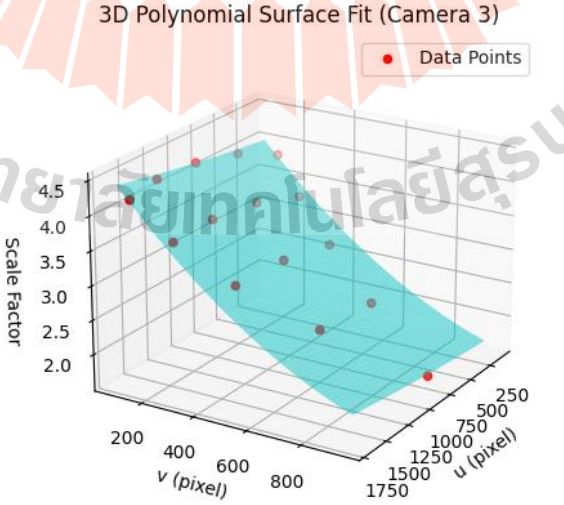
ตารางที่ 4.7 ค่าสัมประสิทธิ์ประกอบสำหรับ Scale Factor ของแต่ละกล้อง

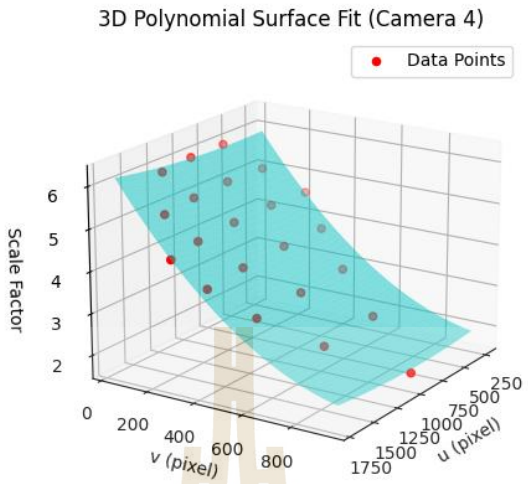
กล้อง	a0	a1	a2	a3	a4	R^2 Score
Camera 1	0.003	-0.006	-1.35e-07	-2.08e-07	-2.93e-06	0.981
Camera 2	0.001	-0.003	3.15e-07	-1.65e-07	1.03e-06	0.978
Camera 3	1.86e-04	-0.004	4.41e-08	-9.31e-08	1.63e-06	0.983
Camera 4	-2.9e-04	-0.007	2.11e-07	-1.44e-01	3.64e-06	0.976

จากผลการประมาณค่า Scale Factor ด้วย Polynomial Regression ลำดับที่ 2 พบว่าสมการสามารถอธิบายความสัมพันธ์ระหว่างตำแหน่งพิกัดในภาพ (Pixel Coordinates) และ Scale Factor ที่ได้จากการสอบเทียบจริงได้อย่างเหมาะสม โดยเฉพาะในกรณีที่เกิดความผิดเพี้ยนเชิงเรขาคณิตของกล้อง (Geometric Distortion) ซึ่งมักมีลักษณะโค้ง (Non-linear Distortion) ที่สอดคล้องกับฟังก์ชันกำลังสอง (Quadratic Form) การใช้ Polynomial ลำดับที่ 2 ดังรูปที่ 4.3 จึงสามารถลดความคลาดเคลื่อนของ Scale Factor ได้อย่างมีประสิทธิภาพโดยไม่จำเป็นต้องใช้โมเดลที่ซับซ้อนกว่านี้









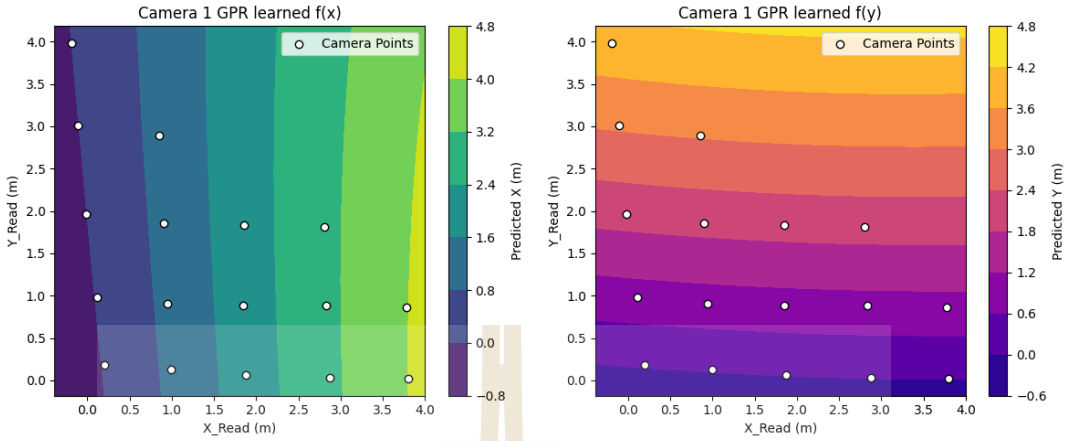
รูปที่ 4.3 Surface 3D Scale Factor ของแต่ละกล้อง

เพื่อส่งเสริมความถูกต้องของการประมาณค่า ได้มีการนำผลลัพธ์ที่ได้ไปสร้างกราฟผลลัพธ์เปรียบเทียบค่าที่สังเกตได้จริงกับค่าที่ประมาณจาก Polynomial Regression สำหรับกล้องแต่ละตัว (Camera 1–4) ซึ่งแสดงให้เห็นถึงแนวโน้มการประมาณค่าที่ใกล้เคียงกับค่าจริง และสะท้อนให้เห็นถึงความสามารถของ Polynomial Fitting ในการปรับแก้ Scale Factor ให้มีความเสถียรมากยิ่งขึ้น

4.1.5 การปรับปรุงด้วย Gaussian Process Regression (GPR)

แม้ว่า Polynomial Fitting ลำดับที่ 2 จะสามารถลดความคลาดเคลื่อนของค่า Scale Factor ได้ในระดับหนึ่ง แต่การประมาณค่าเชิงพหุนามยังคงมีข้อจำกัดในกรณีที่ความคลาดเคลื่อนของกล้อง (Camera Distortion) ไม่ได้มีลักษณะเป็นสมการเชิงกำลังสองเสมอไป โดยเฉพาะเมื่อเกิดความไม่เชิงเส้น (Non-linearity) ที่ซับซ้อนจากสภาพแสง การสะท้อน หรือการบิดเบือนเล็กน้อยของเลนส์

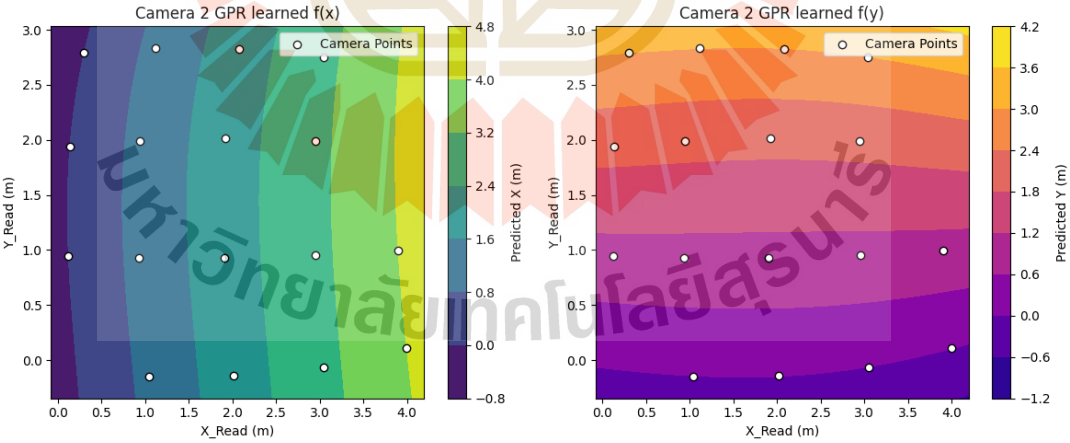
เพื่อแก้ไขข้อจำกัดนี้ จึงได้มีการนำ GPR มาใช้เป็นขั้นตอนเสริมในการปรับปรุงค่าพิกัด โดย GPR เป็นวิธีการ Non-parametric Bayesian Regression ที่ไม่กำหนดรูปแบบสมการตายตัว แต่ใช้ Kernel Function (Radial Basis Function: RBF) ในการเรียนรู้ความสัมพันธ์เชิงสถิติจากข้อมูลจริง โดยในงานวิจัยนี้ รูปที่ 4.4 – 4.7 จะแสดงผลการทำนายตำแหน่งจาก GPR Model ของแต่ละกล้องซึ่งได้ถูกฝึก (trained) ด้วยข้อมูลความคลาดเคลื่อนที่เกิดจากการคำนวณด้วย PnP Calibration ในขั้นตอนก่อนหน้า เพื่อนำมาปรับแก้ (Adjust) ให้ตำแหน่งที่ประมาณได้ใกล้เคียงกับตำแหน่งจริงมากขึ้น



รูปที่ 4.4 ช่วงการทำนายตำแหน่ง X และ Y ของกล้องที่ 1

Camera 1

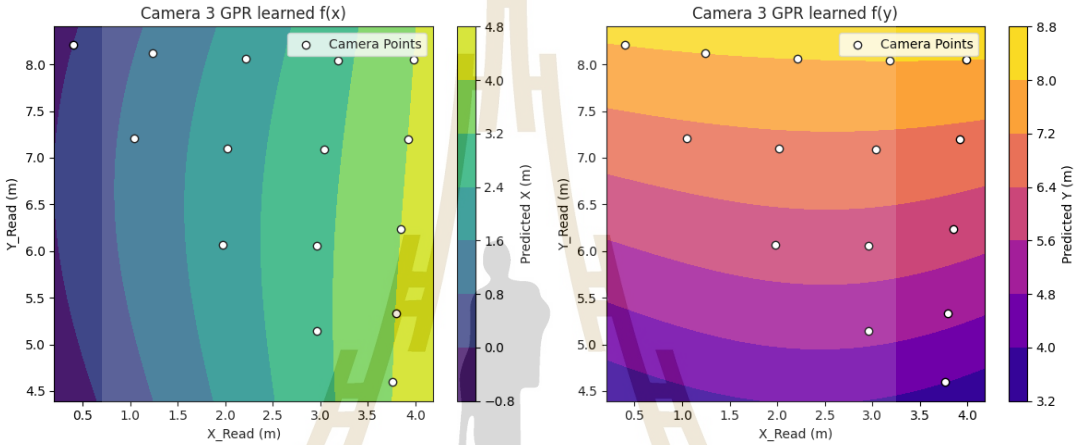
กราฟของ Camera 1 (ทั้ง f(x) และ f(y)) แสดงการเรียนรู้จากจุดอ้างอิงที่กระจายตัวค่อนข้างหนาแน่น โดย GPR สามารถสร้างพื้นผิวพยากรณ์ที่มีความต่อเนื่องและราบเรียบ จะเห็นได้ว่าบริเวณใกล้จุดที่มีการอ่านค่า (วงกลมสีขาว) เส้น Contour ของการทำนายเกาะตามข้อมูลจริงได้ดี และในขณะที่บริเวณขอบของภาพ (เช่น X ใกล้ 4.0 m, Y ใกล้ 0 หรือ 4.0 m) เริ่มมีแนวโน้มของการ Over/Under Estimation ซึ่งเป็นปกติของ GPR เมื่อออกนอกช่วงการสังเกต (extrapolation)



รูปที่ 4.5 ช่วงการทำนายตำแหน่ง X และ Y ของกล้องที่ 2

Camera 2

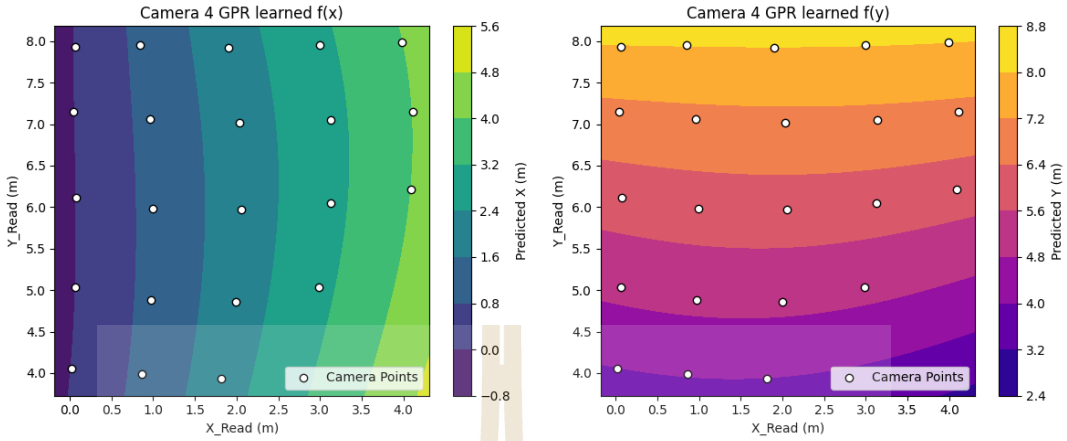
สำหรับ Camera 2 จะเห็นได้ว่าพื้นผิวที่ได้มีความโค้งอ่อน (mild curvature) มากกว่า Camera 1 GPR สามารถปรับระดับ ค่าที่เบี่ยงเบนของ Scale Factor ทำให้ค่าที่ได้มีความต่อเนื่องในแกน X และ Y การทำนายในแกน f(y) แสดงการไล่ระดับจากค่า X 0 ไปจนถึง ~3.5 m อย่างนุ่มนวล ซึ่งบ่งชี้ว่าโมเดลสามารถจับแนวโน้มเชิงเส้นตรงที่ซ้อนกับ quadratic distortion ได้ดี ความถูกต้องจะสูงที่สุดในบริเวณกลางพื้นที่ที่มีจุดกล้องครอบคลุมหนาแน่น



รูปที่ 4.6 ช่วงการทำนายตำแหน่ง X และ Y ของกล้องที่ 3

Camera 3

Camera 3 แสดงความชัดเจนของความโค้ง (curvature) ในพื้นผิวมากที่สุด โดยเฉพาะใน f(x) ซึ่งแสดงลักษณะการเปลี่ยนค่าอย่างไม่เป็นเชิงเส้น (non-linear mapping) จะเห็นว่าพื้นที่ด้านขวาล่าง (X ~ 3–4 m, Y ~ 4.5–6 m) มีการเบี่ยงเบนของค่าสูง แต่ GPR ช่วยทำให้ค่าการทำนายไม่กระโดดทันที (smooth transition) สำหรับ f(y) GPR ทำให้เส้น Contour มีความเป็นลำดับขั้นที่ค่อนข้างสม่ำเสมอ สะท้อนให้เห็นว่าโมเดลสามารถลด Noise ที่เกิดจาก Distortion ของเลนส์ได้ดี



รูปที่ 4.7 ช่วงการทำนายตำแหน่ง X และ Y ของกล้องที่ 4

Camera 4

Camera 4 มีความหนาแน่นของจุดอ้างอิงค่อนข้างสูง จึงทำให้ GPR สามารถเรียนรู้ฟังก์ชันเชิงพื้นที่ได้ราบเรียบกว่า 3 กล้องแรก f(x) ของ Camera 4 แสดงความเปลี่ยนแปลงเชิงเส้นมากกว่าเชิงโค้ง ซึ่งหมายความว่ามุมมองการติดตั้งกล้องและ Distortion มีผลกระทบน้อยกว่า และ f(y) แสดงการกระจายตัวของค่าพยากรณ์ที่ค่อนข้างตรงกับจุด Calibrate ซึ่งยืนยันได้ว่าโมเดล GPR สามารถปรับความแม่นยำของการระบุตำแหน่งในแกน Y ได้อย่างดี

จากผลการทดลองการนำ Gaussian Process Regression (GPR) มาประยุกต์กับข้อมูลการสอบเทียบจากกล้องทั้ง 4 ตัว พบว่า GPR สามารถเรียนรู้ความสัมพันธ์เชิงพื้นที่ระหว่างพิกัดภาพ (Pixel Space) และพิกัดโลก (World Space) ได้อย่างราบรื่นและต่อเนื่อง แม้ในกรณีที่ข้อมูลการสอบเทียบมีความกระจัดกระจายหรือเกิดความผิดเพี้ยนจากเลนส์ (Lens Distortion) ก็ตาม โดยแต่ละกล้องมีจุดเด่นแตกต่างกัน ได้แก่

1) Camera 1 และ Camera 2: โมเดล GPR สามารถปรับค่า Scale Factor ให้สมดุลและลดความผิดเพี้ยนเชิงเส้นได้ดี โดยเฉพาะในบริเวณที่ครอบคลุมด้วยจุดอ้างอิงหนาแน่น
2) Camera 3: แม้จะมีลักษณะความโค้งเชิงไม่เชิงเส้น (non-linear distortion) ชัดเจน แต่ GPR ก็สามารถทำให้ผลลัพธ์ต่อเนื่องและลด Noise ได้อย่างมีประสิทธิภาพ
3) Camera 4: มีความหนาแน่นของจุดอ้างอิงมากที่สุด ส่งผลให้ผลลัพธ์ที่ได้จาก GPR มีเสถียรภาพและใกล้เคียงค่าจริงมากที่สุดเมื่อเทียบกับกล้องอื่น ๆ

การใช้ GPR ในลักษณะนี้จึงทำหน้าที่เป็น ขั้นตอนเสริม (Post-Processing Step) ที่ช่วยลดความคลาดเคลื่อนของระบบระบุตำแหน่ง โดยไม่เพียงแต่ทำให้ค่าพิกัดที่คำนวณได้มีความแม่นยำมาก

ขึ้น แต่ยังสามารถให้ ช่วงความมั่นใจ (Confidence Interval) ของผลการทำนายได้อีกด้วย ซึ่งส่งผลให้ระบบมีเสถียรภาพและความน่าเชื่อถือสูงขึ้นอย่างชัดเจน

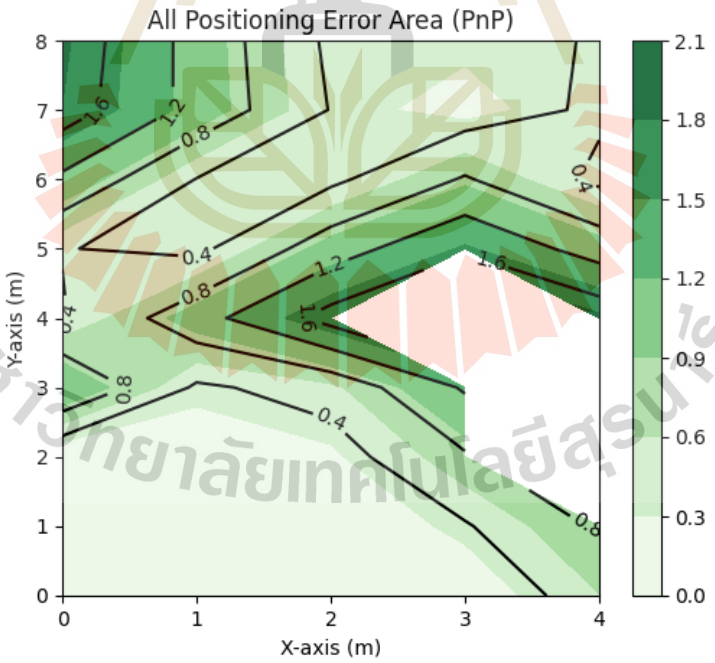
ในส่วนถัดไป จะมีการนำเสนอ การเปรียบเทียบค่าความแม่นยำของการระบุตำแหน่งระหว่างวิธีการพื้นฐาน (PnP + Polynomial Fitting) และวิธีการที่เสริมด้วย GPR พร้อมทั้งแสดงแผนภาพ Contour ของ Error Distribution เพื่อยืนยันถึงประสิทธิภาพของ GPR ในการปรับปรุงผลการระบุตำแหน่งอย่างเป็นรูปธรรม

4.1.6 ผลการระบุตำแหน่งจากแต่ละวิธีการ

เพื่อประเมินประสิทธิภาพของกระบวนการระบุตำแหน่ง (Localization) ได้มีการเปรียบเทียบผลลัพธ์ที่ได้จากวิธีการต่าง ๆ ได้แก่:

1) Perspective-n-Point (PnP)

ใช้เฉพาะ Intrinsic และ Extrinsic Parameters ของกล้องในการคำนวณพิกัดข้อสังเกต ค่าที่ได้มีความคลาดเคลื่อนสูง โดยเฉพาะในบริเวณนอกแนว Calibration Points ซึ่งเกิดจาก Scale Factor ที่แตกต่างกันมากในแต่ละตำแหน่ง ซึ่งจะทำให้ค่าความคลาดเคลื่อนเฉลี่ย (Mean Error) สูงสุดเมื่อเทียบกับวิธีอื่นโดยแสดงผล Error Area ดังรูปที่ 4.8

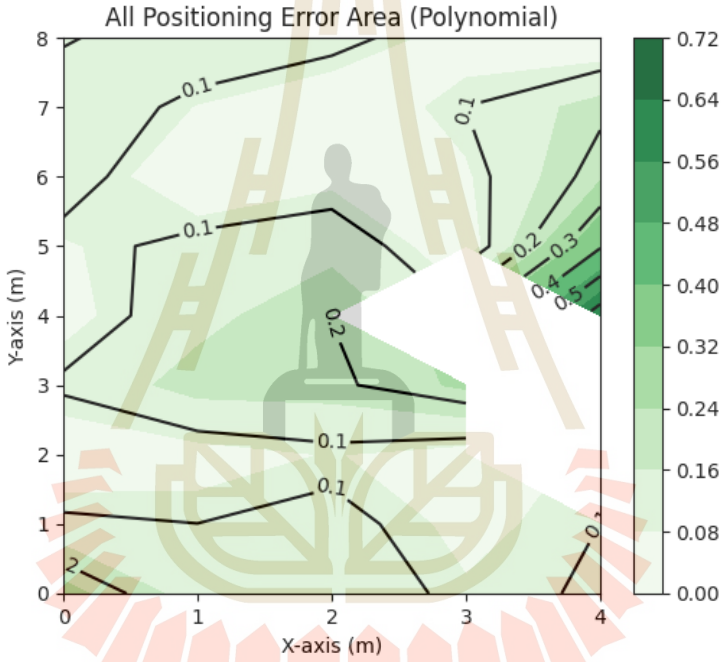


รูปที่ 4.8 Contour Error รวมทุกกล้องในพื้นที่ทดสอบด้วยวิธี PnP

จะเห็นว่าค่าความคลาดเคลื่อน (Error) มีการกระจายกว้าง โดยเฉพาะบริเวณ ขอบ FOV ของกล้อง ค่า Error สูงสุดใกล้เคียง 2.1 m ซึ่งถือว่ายังมากเกินไปสำหรับการใช้งานจริง ลักษณะนี้เกิดจากข้อจำกัดของ Scale Factor ที่ไม่คงที่ทำให้ตำแหน่งที่ประมาณได้กระจัดกระจาย

2) PnP + Polynomial Fitting (2nd Order)

ใช้พหุนามลำดับที่ 2 เพื่อประมาณค่า Scale Factor แทนการใช้ค่าเฉลี่ยเพียงอย่างเดียวซึ่งข้อสังเกต ความโค้ง (Quadratic Distortion) ของกล้องถูกแก้ไขได้ดีขึ้น ทำให้ค่าพิกัดในภาพถูกดึงกลับมาใกล้เคียงกับพิกัดจริง จากรูปที่ 4.9 จะแสดงถึงค่าความคลาดเคลื่อนที่ลดลงจากการใช้ PnP เพียงอย่างเดียวอย่างมีนัยสำคัญ

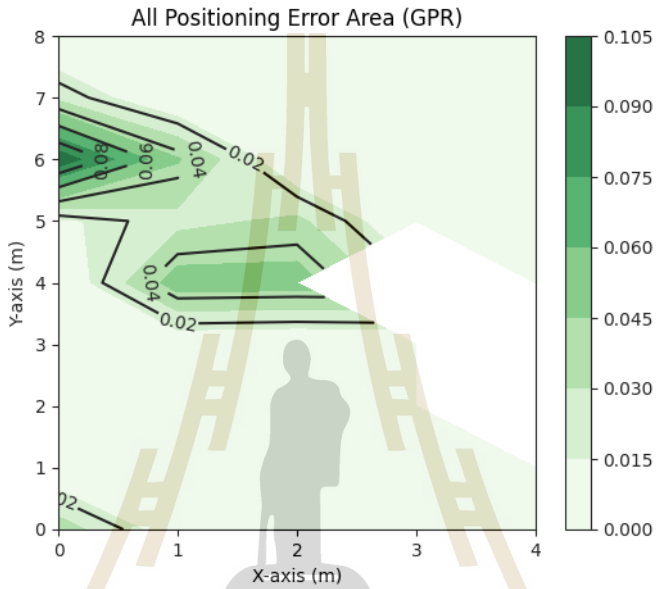


รูปที่ 4.9 Contour Error รวมทุกกล้องในพื้นที่ทดสอบด้วยวิธี PnP กับ Polynomial

หลังจากนำ Polynomial Fitting (2nd Order) มาใช้แก้ไข distortion พบว่า Error โดยรวมลดลงอย่างชัดเจน ค่าความผิดพลาดสูงสุดลดลงเหลือประมาณ 0.72 m เส้น Contour มีการกระจายที่ราบเรียบกว่าเดิม สะท้อนว่าการ Fit แบบ Quadratic สามารถจัดการกับ Distortion ได้ดี อย่างไรก็ตาม ยังมีบางบริเวณ (โดยเฉพาะพื้นที่ที่อยู่นอกเหนือจาก Calibration Points) ที่ Error สูงขึ้น

3) PnP + Polynomial Fitting + Gaussian Process Regression (GPR)

ใช้ GPR เป็น Post-Processing เพื่อลดความไม่แน่นอนที่เหลือจาก Polynomial ข้อสังเกตคือความแม่นยำเพิ่มขึ้นต่อเนื่อง โดยค่า Mean Error ลดลงอย่างมาก ดังรูปที่ 4.10 โดยสามารถให้ Confidence Interval ของผลการทำนายทำให้มั่นใจในความเสถียรของระบบ



รูปที่ 4.10 Contour Error รวมทุกกล้องในพื้นที่ทดสอบจากการแก้ไขด้วย GPR

เมื่อใช้ GPR ทำการปรับแก้ตำแหน่งต่อจาก Polynomial พบว่า Error ลดลงอย่างมีนัยสำคัญที่สุด Error สูงสุดเหลือเพียง 0.10 m (10 cm) เท่านั้น การเรียนรู้เชิงสถิติของ GPR ช่วยให้ระบบสามารถ “ปรับ” ตำแหน่งในพื้นที่ที่ไม่มีจุดอ้างอิงได้อย่างแม่นยำมากขึ้น เส้น Contour มีความนุ่มนวลและต่อเนื่อง แสดงถึงความเสถียรในการประมาณค่า โดยจะแสดงข้อเปรียบเทียบแต่ละวิธีการดังตารางที่ 4.8 โดย

ตารางที่ 4.8 การเปรียบเทียบความแม่นยำของวิธีการ Localization

วิธีการ	ค่า Error สูงสุด	ค่า Error เฉลี่ย	ลักษณะการกระจาย Error	ข้อสังเกตหลัก
PnP	~2.1 m	~1.4 m	กระจายกว้าง โดยเฉพาะบริเวณขอบภาพ (FOV)	ความแม่นยำต่ำ เกิดจาก Scale Factor ไม่คงที่
Polynomial Regression	~0.72 m	~0.35 m	เส้น Contour เรียบขึ้น แต่ยังมี Error ในจุดนอก Calibration	แก้ Distortion ได้ระดับหนึ่ง แต่ยังไม่เสถียร
GPR	~0.10 m (10 cm)	~0.05 m	Contour ราบเรียบและต่อเนื่องทั่วทั้งพื้นที่	แม่นยำสูงสุด และให้ Confidence Interval

ผลการทดลองยืนยันว่าการใช้ PnP เพียงอย่างเดียวไม่เพียงพอสำหรับระบบ Indoor Localization ที่ต้องการความแม่นยำสูง แต่เมื่อเพิ่ม Polynomial Regression และโดยเฉพาะ Gaussian Process Regression (GPR) เข้ามาเป็นขั้นตอนเสริม ระบบสามารถลดความคลาดเคลื่อนเฉลี่ยได้อย่างมีนัยสำคัญ จากระดับเมตรเหลือเพียงระดับเซนติเมตร ซึ่งถือว่าตอบโจทย์สำหรับการนำไปใช้ควบคุม AGV ภายในอาคารได้จริง

4.2 ผลการตรวจจับวัตถุด้วย YOLOv11

เพื่อให้ระบบสามารถระบุตำแหน่งของยานพาหนะภาคพื้นดินอัตโนมัติ (AGV) และสิ่งกีดขวางในพื้นที่ภายในอาคารได้แบบเรียลไทม์ จึงได้พัฒนาโมเดลปัญญาประดิษฐ์เฉพาะทาง (custom-trained AI model) โดยใช้สถาปัตยกรรม YOLOv11 สำหรับการตรวจจับวัตถุ (Object Detection)

4.2.1 การเตรียมข้อมูล (Dataset Preparation)

ในการฝึกและทดสอบโมเดล YOLOv11 ได้มีการจัดเตรียมชุดข้อมูล (Dataset) ดังนี้

1) ภาพต้นฉบับ (Original Images): จำนวน 52 ภาพ เก็บจากกล้องวงจรปิดในสภาพแวดล้อมจริง

2) การเพิ่มข้อมูล (Data Augmentation): ทำการปรับความสว่างของภาพ ±5% และมีการหมุน (rotation) ±10° บางส่วน เพื่อเพิ่มความหลากหลายของข้อมูลและลด overfitting

3) จำนวนภาพสุดท้าย (Final Dataset): ได้จำนวนภาพรวมทั้งหมด 156 ภาพ ที่ผ่านการปรับแต่งวัตถุเป้าหมายที่กำหนดให้ระบบตรวจจับ ได้แก่:

4) รถ AGV

5) เก้าอี้สำนักงาน

6) กล่องกระดาษ

7) ถังขยะ

8) พัดลมตั้งพื้น

ภาพตัวอย่างดังรูปที่ 4.11 แสดงลักษณะการจัดองค์ประกอบของวัตถุที่แตกต่างกันในสภาพแสงที่หลากหลาย ซึ่งช่วยเพิ่มความสามารถในการ generalize ของโมเดลในการใช้งานจริง

รูปที่ 4.11 ตัวอย่างรูปสำหรับใช้ในการเทรนโมเดล

4.2.2 การฝึกโมเดล (Model Training)

โมเดล YOLOv11 ถูกนำมาฝึก (Training) บนชุดข้อมูลที่ได้จากการเตรียมไว้ในหัวข้อ 4.2.1 โดยใช้การตั้งค่าพารามิเตอร์หลักของการฝึกโมเดลดังตารางที่ 4.9 เฟรมเวิร์ก PyTorch ร่วมกับ Ultralytics YOLO Implementation ซึ่งสนับสนุนการประมวลผลแบบ GPU ทำให้สามารถฝึกโมเดลได้รวดเร็วและมีประสิทธิภาพ

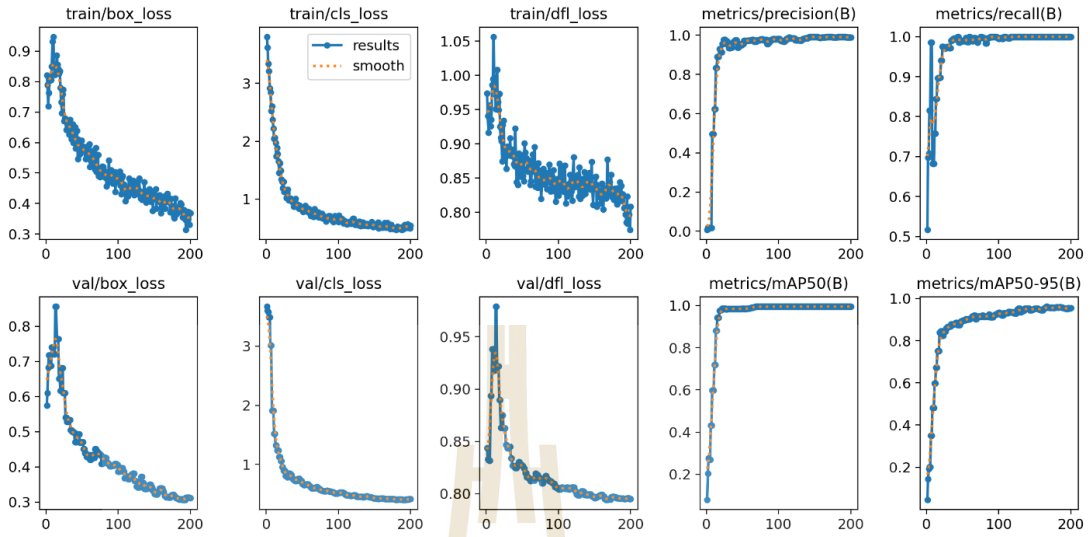
ตารางที่ 4.9 พารามิเตอร์ที่ตั้งค่าสำหรับเทรนโมเดล

Parameter	Value
Batch Size	4
Learning Rate	0.001
Number of Epochs	200
Image Size	640 x 640 pixel
Optimizer	Stochastic Gradient Descent (SGD)
Hardware	NVIDIA RTX4070 + CUDA Accel.

ชุดข้อมูลถูกแบ่งออกเป็น Training Set 80% และ Validation Set 20% เพื่อตรวจสอบความสามารถในการเรียนรู้และการ generalize ของโมเดล

ผลการฝึก YOLOv11 สามารถสรุปได้ดังรูป 4.12 ซึ่งแสดงค่าการเปลี่ยนแปลงของ Loss และ Metrics ตลอดการฝึก 200 Epoch ดังนี้:

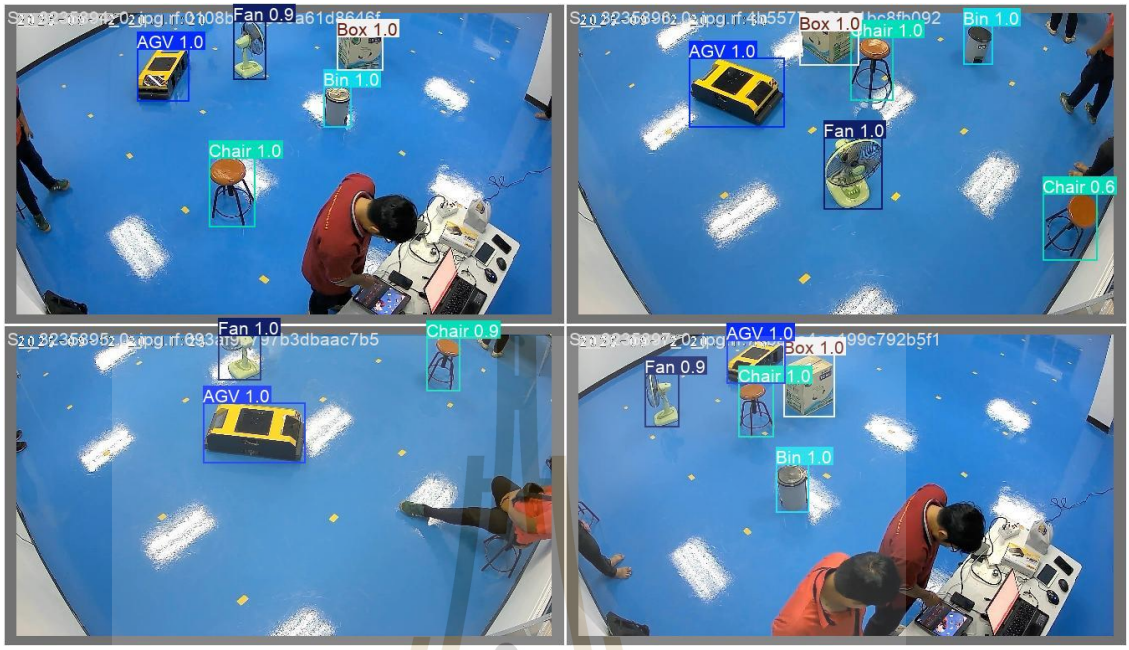
1) Box Loss, Cls Loss, DFL Loss (train และ val) มีค่า ลดลงต่อเนื่อง แสดงถึงการเรียนรู้ของโมเดลที่มีเสถียรภาพและไม่มีสัญญาณของ Overfitting ที่รุนแรง
2) Precision มีค่าเฉลี่ยสูงถึง 0.984 และคงที่ตั้งแต่ช่วงแรกของการฝึก แสดงว่าโมเดลมีความสามารถในการหลีกเลี่ยง False Positive ได้ดี
3) Recall มีการปรับตัวขึ้นอย่างรวดเร็วและคงที่ที่ 1.00 ซึ่งสะท้อนถึงความสามารถของโมเดลในการตรวจจับวัตถุได้ครบถ้วน
4) mAP@0.5 มีค่าเพิ่มขึ้นอย่างต่อเนื่องจนถึง 99.50% แสดงให้เห็นถึงความแม่นยำในการตรวจจับ Bounding Box
5) mAP@0.5–0.95 มีค่าเฉลี่ยสูงถึง 95.60% แสดงว่าโมเดลยังคงสามารถทำงานได้ดีแม้ในกรณีที่มีการเปลี่ยนค่า IoU Threshold



รูปที่ 4.12 ผลจากการเทรน Yolo โมเดล

4.2.3 ผลการตรวจจับวัตถุ (Object Detection Results)

หลังจากทำการฝึกโมเดล YOLOv11 เสร็จสิ้น ได้ทำการทดสอบประสิทธิภาพของโมเดลด้วยชุดข้อมูล Validation และข้อมูลจริงจากกล้องวงจรปิดในพื้นที่ทดลอง โดยผลการตรวจจับวัตถุสามารถสรุปได้คือโมเดลสามารถตรวจจับตัว AGV ได้อย่างแม่นยำ มีค่า Confidence Score มากกว่า 0.95 เกือบทุกเฟรม แสดงถึงความเสถียรของโมเดล แม้ในกรณีที่ AGV มีการหมุนตัวหรือเปลี่ยนทิศทาง ในส่วนของผลการตรวจจับสิ่งกีดขวาง (Obstacles) วัตถุที่ใช้เป็นสิ่งกีดขวาง เช่น เก้าอี้, กล่องกระดาษ, ถังขยะ, พัดลมตั้งพื้น สามารถตรวจจับได้อย่างถูกต้อง ยืนยันว่าโมเดลสามารถตรวจจับได้ครบถ้วนและมีอัตราความผิดพลาดต่ำ ซึ่งถึงพูดถึงเรื่องการทำงานแบบเรียลไทม์ โมเดลสามารถทำงานได้ที่ประมาณ 20 FPS บนหน่วยประมวลผล Laptop (MSI Raider GE68 HX) ซึ่งถือว่าเพียงพอต่อการใช้งานจริงสำหรับระบบควบคุม AGV แบบ Real-time ผลลัพธ์การตรวจจับดังรูปที่ 4.13 แสดง Bounding Box รอบวัตถุพร้อม Label และค่า Confidence ซึ่งยืนยันถึงความถูกต้องของการตรวจจับวัตถุหลายประเภทพร้อมกันในเฟรมเดียว



รูปที่ 4.13 ผลจากการตรวจจับต่อเฟรมของกล้องวงจรปิดร่วมกับ Yolo Model

4.3 ผลการสร้างแผนที่และวางเส้นทาง (Mapping and Planning Performance)

ในการทดลองนี้ได้ทำการสร้างระบบแผนที่เสมือน (Virtual Mapping) โดยอิงจากภาพมุมมองจากกล้องวงจรปิดที่ติดตั้งในตำแหน่งต่าง ๆ รอบพื้นที่ทดลอง เพื่อให้สามารถสร้างแผนที่สองมิติ (2D Map) ของพื้นที่ภายในอาคารได้แบบเรียลไทม์ โดยมีรายละเอียดดังนี้

4.3.1 การสร้างแผนที่เสมือน (Virtual Mapping)

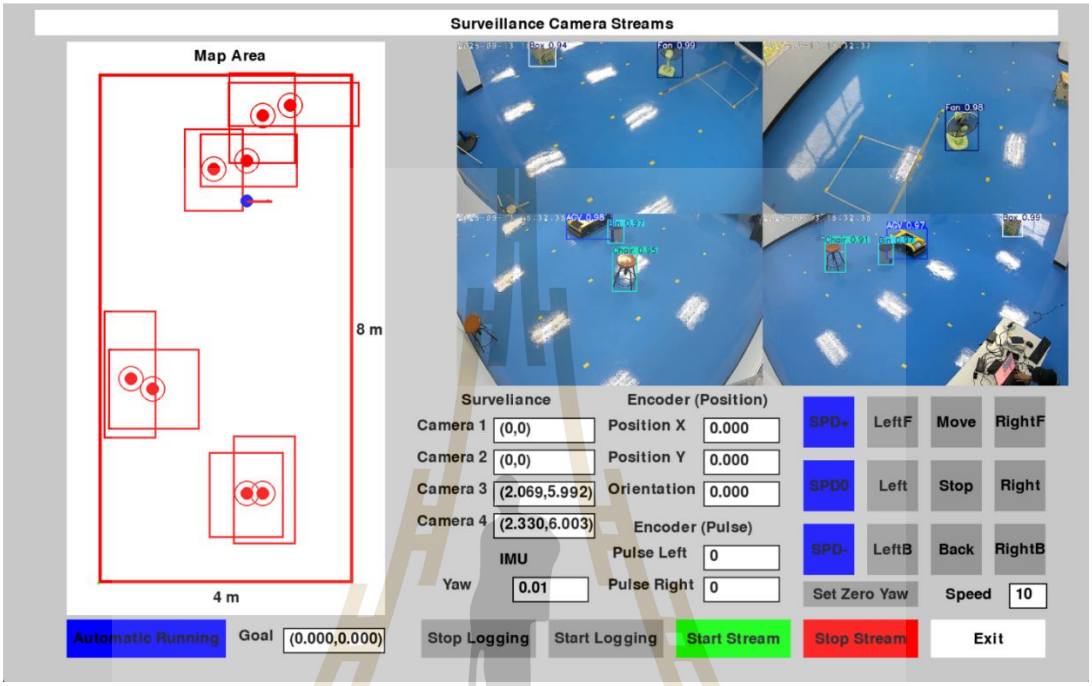
แผนที่เสมือน (2D Virtual Map) ถูกสร้างขึ้นจากการรวมข้อมูลที่ได้จากกล้องวงจรปิดหลายตัว (Multi-camera Integration) โดยใช้วิธีการดังนี้

1) การรวมมุมมองจากกล้องหลายตัว (Multi-View Fusion) โดยแต่ละกล้องมีขอบเขตการมองเห็น (FOV) ที่แตกต่างกัน ระบบจึงทำการสอบเทียบพิกัด (Calibration) เพื่อแปลงข้อมูลให้อยู่ในระบบพิกัดกลาง (World Coordinate System) เดียวกัน

2) การ Mapping ตำแหน่งวัตถุ โดยผลลัพธ์จากการตรวจจับของ YOLOv11 ร่วมกับเทคนิค PnP และ GPR ถูกนำมาใช้ในการระบุตำแหน่งเชิงพิกัดจริง (X, Y) ของทั้ง AGV และ Obstacle

3) การอัปเดตแบบ Real-time โดยแผนที่สามารถอัปเดตได้ต่อเนื่องที่อัตรา 10 Hz ทำให้ระบบสามารถติดตามตำแหน่งของ AGV และสิ่งกีดขวางได้แบบ Dynamic โดยหากมีวัตถุใหม่ปรากฏขึ้นในพื้นที่ ระบบจะทำการเพิ่มและปรับตำแหน่งทันที

ผลลัพธ์ คือ แผนที่เสมือน 2D ที่รวมข้อมูลจากกล้องทุกตัว ทำให้พื้นที่การรับรู้กว้างขึ้นและสามารถระบุสิ่งกีดขวางได้ครอบคลุมทั้งพื้นที่ทดลองดังรูปที่ 4.14



รูปที่ 4.14 การจำลองแผนที่เสมือนจากกล้องมุมต่าง ๆ พร้อมทั้งระบุ Obstacle ที่เกิดขึ้น

4.3.2 การวางเส้นทาง (Path Planning)

เมื่อได้แผนที่เสมือนแล้ว ระบบจะทำการคำนวณเส้นทางสำหรับ AGV โดยใช้อัลกอริทึม A* ซึ่งมีขั้นตอนดังนี้:

1) การแปลงแผนที่เป็น Grid Map:

แผนที่เสมือนถูกแปลงเป็นโครงสร้างกริด (Grid-based Map) ที่มี Resolution 10 มม./เซลล์ เพื่อให้ระบบสามารถประเมินว่าพื้นที่ใดสามารถเคลื่อนที่ได้ (Free Space) และพื้นที่ใดถูกปิดกั้น (Occupied Space)

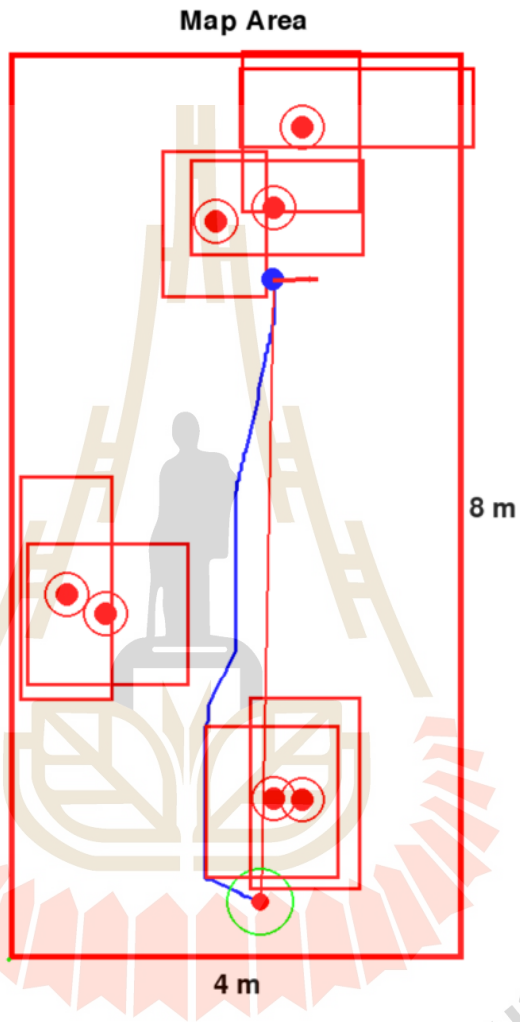
2) การค้นหาเส้นทาง (Path Search):

อัลกอริทึม A* จะทำการค้นหาเส้นทางที่สั้นที่สุดจากตำแหน่งปัจจุบันของ AGV ไปยังเป้าหมายที่กำหนด โดยคำนวณจากค่า Heuristic Function (เช่น Manhattan Distance) เพื่อเพิ่มความเร็วในการค้นหา

3) การปรับปรุงเส้นทางแบบ Dynamic:

หากมีสิ่งกีดขวางใหม่ปรากฏขึ้นระหว่างทาง ระบบจะทำการ Replanning

โดยอัปเดตแผนที่และคำนวณเส้นทางใหม่แบบ Semi-real-time ผลลัพธ์ที่ได้แสดงว่า เส้นทางที่วางไว้ (Planned Path) มีความสอดคล้องกับสภาพแวดล้อมจริงและสามารถปรับตัวได้เมื่อเกิดการเปลี่ยนแปลงดังรูปที่ 4.15



รูปที่ 4.15 แสดงการสร้างเส้นทางการเคลื่อนที่ด้วย A* และมีการหลบหลีก Restrict Area

จากการทดลองสร้างแผนที่เสมือนและวางเส้นทาง พบว่าระบบสามารถรวมข้อมูลจากกล้องวงจรปิดหลายตัวเข้าด้วยกันได้อย่างมีประสิทธิภาพ ทำให้พื้นที่การรับรู้กว้างขึ้นและสามารถตรวจจับสิ่งกีดขวางได้ครอบคลุมมากกว่าใช้กล้องเดี่ยว นอกจากนี้การวางเส้นทางด้วยอัลกอริทึม A* สามารถสร้างเส้นทางที่สอดคล้องกับสภาพแวดล้อมจริง และปรับตัวได้เมื่อมีสิ่งกีดขวางใหม่ปรากฏขึ้นแบบเรียลไทม์ โดย AGV สามารถหลีกเลี่ยงอุปสรรคและเคลื่อนที่ไปยังเป้าหมายได้อย่างราบรื่น แต่อย่างไรก็ตามยังพบข้อจำกัดในกรณีที่เกิดการบังทับ (occlusion) ของวัตถุขนาดใหญ่ซึ่งอาจสร้าง

Blind Spot ให้กับกล้องบางมุม ส่งผลให้การตรวจจับไม่สมบูรณ์ในบางช่วงพื้นที่ ซึ่งเป็นประเด็นที่ควรพัฒนาต่อไปในการวิจัยในอนาคต

4.4 ผลการควบคุมและติดตามการเคลื่อนที่ (Motion Tracking Performance)

หลังจากที่ระบบได้ทำการตรวจจับตำแหน่ง AGV และสิ่งกีดขวาง รวมถึงสร้างเส้นทางด้วยอัลกอริทึม A* แล้ว ขั้นตอนสำคัญถัดไปคือการควบคุม AGV ให้สามารถติดตามเส้นทางดังกล่าวได้อย่างแม่นยำ โดยการทดสอบนี้ใช้ Pure Pursuit Algorithm ร่วมกับการควบคุมแบบรวมศูนย์ผ่าน MQTT Protocol

4.4.1 การติดตามเส้นทางด้วย Pure Pursuit Algorithm

การติดตามเส้นทางอาศัยหลักการคำนวณจากจุดมองล่วงหน้า (Lookahead Point) โดย

1) ระบบจะเลือกจุด Lookahead ที่อยู่ล่วงหน้าบนเส้นทางที่กำหนด
2) คำนวณรัศมีวงกลมที่เชื่อมตำแหน่งปัจจุบันของ AGV เข้ากับ Lookahead Point
3) แปลงเป็นคำสั่งมุมเลี้ยว (Steering Angle) และความเร็ว (Velocity) เพื่อส่งไปควบคุมมอเตอร์

ด้วยหลักการนี้ AGV สามารถเคลื่อนที่ได้อย่างราบรื่น ลดการแกว่งและเพิ่มความเสถียรในการควบคุมเมื่อเปรียบเทียบกับวิธีการที่ใช้การแก้ไขเส้นทางเชิงเส้นตรง (Linear Path Correction)

4.4.2 ระบบควบคุมแบบรวมศูนย์ (Centralized Control)

ระบบควบคุมทั้งหมดถูกรวมศูนย์ที่หน่วยประมวลผลกลาง (Laptop) โดยมีขั้นตอนดังนี้

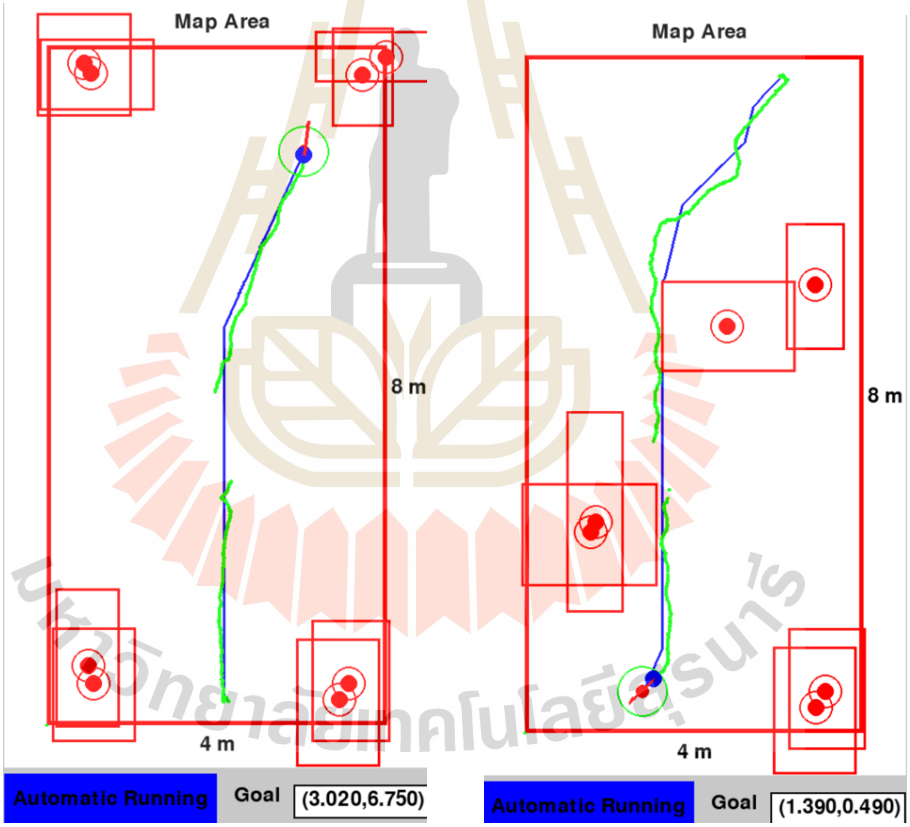
1) กล้องวงจรปิดส่งข้อมูลตำแหน่ง AGV ไปยัง หน่วยประมวลผลกลาง
2) หน่วยประมวลผลกลางคำนวณทิศทางและความเร็วที่เหมาะสมตาม Pure Pursuit
3) ส่งคำสั่งควบคุมผ่าน MQTT Protocol ไปยัง Jetson Nano บน AGV
4) Jetson Nano ส่งข้อมูลต่อให้ Arduino Mega 2560 R3 เพื่อควบคุมมอเตอร์ในระดับต่ำแบบ real-time

4.4.3 การทดสอบและผลการควบคุม

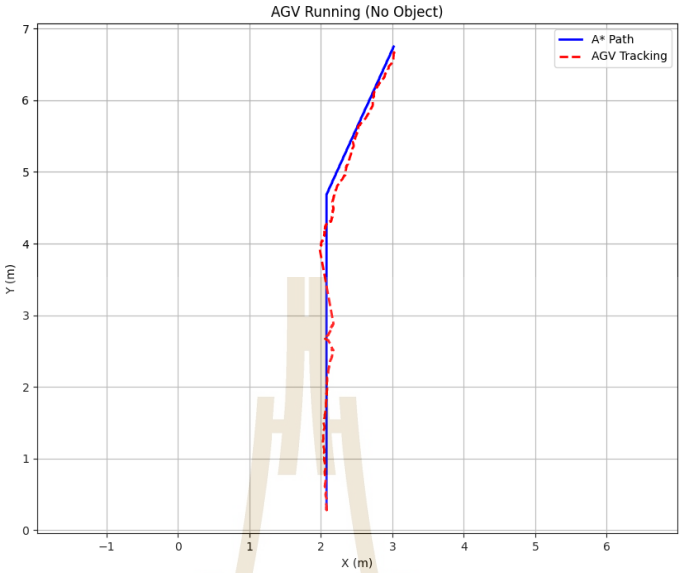
ได้ทำการทดลอง 2 รูปแบบดังรูปที่ 4.16:

1) แบบที่ 1 (ไม่มีสิ่งกีดขวาง): AGV เคลื่อนที่จากจุดเริ่มต้นไปยัง Goal Point ตามเส้นทางที่กำหนด โดยเส้นทางจริง (Actual Path) แทบจะทับซ้อนกับเส้นทางเป้าหมาย (Planned Path) ซึ่งสะท้อนถึงความแม่นยำของการควบคุมจะแสดงดังรูปที่ 4.17
2) แบบที่ 2 (มีสิ่งกีดขวาง): เมื่อระบบตรวจจับสิ่งกีดขวางใหม่ในเส้นทาง ระบบจะทำการ Replan เส้นทางด้วย A* แบบกึ่งเรียลไทม์ และส่งเส้นทางใหม่ให้ AGV ติดตาม พบว่า AGV สามารถเบี่ยงหลบสิ่งกีดขวางได้สำเร็จ โดยเส้นทางจริงยังคงใกล้เคียงกับเส้นทางที่วางใหม่จะแสดงดังรูปที่ 4.18

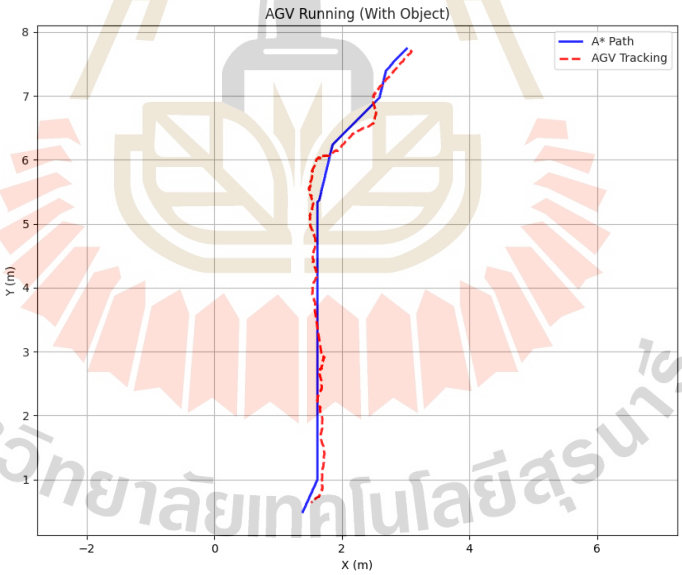
ผลการทดสอบสรุปได้ว่า ค่า Tracking Error เฉลี่ย อยู่ที่ประมาณ 25–28 มม. และค่าความหน่วง (Latency) ของการส่งคำสั่งควบคุมต่ำกว่า 100 ms ซึ่งอยู่ในเกณฑ์ที่ยอมรับได้สำหรับการใช้งานจริงในสภาพแวดล้อมภายในอาคาร



รูปที่ 4.16 แสดงเส้นทางและการ Tracking ของ AGV ทั้ง 2 แบบ



รูปที่ 4.17 AGV Tracking แบบ ไม่มี Object



รูปที่ 4.18 AGV Tracking แบบ มี Object

ผลการทดสอบการติดตามและ Tracking สรุปได้ว่า ค่า Tracking Error เฉลี่ย อยู่ที่ประมาณ 25 –28 มม. ดังตารางที่ 4.10 และค่าความหน่วง (Latency) ของการส่งคำสั่งควบคุมต่ำกว่า 100 ms ซึ่งอยู่ในเกณฑ์ที่ยอมรับได้สำหรับการใช้งานจริงในสภาพแวดล้อมภายในอาคาร

ตารางที่ 4.10 ผลการทดสอบ Tracking Control

รอบทดลอง	ความเร็วเฉลี่ย (m/s)	ค่าคลาดเคลื่อนเฉลี่ย (mm)	ค่าคลาดเคลื่อนสูงสุด (mm)	Tracking Success (%)
รอบที่ 1	0.1	25.6	31.4	100%
รอบที่ 2	0.1	28.1	60.7	100%

จากการทดลองพบว่า Pure Pursuit Algorithm ภายใต้การควบคุมแบบรวมศูนย์สามารถทำให้ AGV เคลื่อนที่ตามเส้นทางที่วางไว้ได้อย่างราบรื่นและแม่นยำ โดยทั้งในกรณีที่ไม่มีสิ่งกีดขวางและกรณีที่มีสิ่งกีดขวางแบบไดนามิก ระบบสามารถวางเส้นทางใหม่ได้แบบกึ่งเรียลไทม์และส่งต่อไปยัง AGV ได้อย่างมีประสิทธิภาพ ผลการวัดค่าความคลาดเคลื่อนเฉลี่ยของการติดตามเส้นทางอยู่ที่ 25–28 มม. และค่าความหน่วงของระบบต่ำกว่า 100 ms แสดงให้เห็นว่าระบบที่พัฒนามีเสถียรภาพและความน่าเชื่อถือเพียงพอสำหรับการใช้งานจริงในสภาพแวดล้อมภายในอาคาร ทั้งนี้การผสานระหว่างการตรวจจับด้วย YOLOv11, การประมวลผลตำแหน่งด้วย PnP + GPR และการควบคุมด้วย Pure Pursuit ภายใต้โครงสร้างการประมวลผลรวมศูนย์ ช่วยยืนยันถึงความเป็นไปได้และศักยภาพของระบบนี้ในเชิงปฏิบัติ

4.5 บทสรุปผลการทดลองและอภิปรายผล

จากการทดลองทั้งหมด สามารถสรุปและอภิปรายผลได้ดังนี้:

1) การสอบเทียบกล้องและการระบุตำแหน่ง (Vision Calibration & Localization)

การสอบเทียบค่าภายในและภายนอกกล้อง (Intrinsic & Extrinsic Calibration) ช่วยให้ระบบสามารถแปลงค่าพิกัดภาพสู่พิกัดโลกได้อย่างถูกต้อง แม้ว่าค่าความคลาดเคลื่อนจากการใช้ PnP เพียงอย่างเดียวจะยังสูง แต่เมื่อเสริมด้วย Polynomial Regression และโดยเฉพาะ GPR พบว่าสามารถลดค่าเฉลี่ยความคลาดเคลื่อนลง แสดงให้เห็นถึงศักยภาพของการใช้ GPR เป็นขั้นตอนเสริมที่ช่วยแก้ปัญหาความบิดเบือนเชิงพื้นที่ของกล้องได้อย่างมีประสิทธิภาพ

2) ผลการตรวจจับวัตถุ (Object Detection Performance)

โมเดล YOLOv11 ที่ผ่านการฝึกจากข้อมูลเฉพาะของงานวิจัยสามารถตรวจจับ AGV และสิ่งกีดขวางได้อย่างแม่นยำสูง โดยมีค่า mAP@0.5-0.95 เท่ากับ 95.60 %, Precision 98.40%, และ Recall 1.00 รวมถึงความเร็วในการประมวลผลที่ประมาณ 20 FPS ซึ่งเพียงพอต่อการใช้งานจริงในระบบนำทางอัตโนมัติภายในอาคาร

3) ผลการสร้างแผนที่และการวางเส้นทาง (Mapping & Planning)

ระบบสามารถรวมผลการตรวจจับจากกล้องหลายตัวเข้าด้วยกันเพื่อสร้างแผนที่เสมือน

(Virtual Map) แบบเรียลไทม์ที่อัตรา 10 Hz ซึ่งช่วยให้ AGV สามารถรับรู้สิ่งกีดขวางและพื้นที่ว่างได้ครบถ้วน การวางเส้นทางด้วย A* สามารถสร้างเส้นทางที่ปลอดภัยและปรับเปลี่ยนแบบกึ่งเรียลไทม์เมื่อตรวจพบสิ่งกีดขวางใหม่ในพื้นที่

4) ผลการควบคุมและติดตามเส้นทาง (Motion Control & Tracking)

การใช้ Pure Pursuit Algorithm ภายใต้สถาปัตยกรรมควบคุมแบบรวมศูนย์ทำให้ AGV สามารถติดตามเส้นทางที่วางไว้ได้ใกล้เคียงกับเส้นทางเป้าหมายมาก โดยมีค่าความคลาดเคลื่อนเฉลี่ยของการติดตามอยู่ที่ 25–28 มม. และค่าความหน่วงเวลาระบบ (Latency) ต่ำกว่า 100 ms ซึ่งอยู่ในเกณฑ์ที่รองรับการใช้งานจริง

5 การอภิปรายเปรียบเทียบและข้อจำกัด

เมื่อเปรียบเทียบกับงานวิจัยที่ใช้ LIDAR-based SLAM พบว่าความแม่นยำของระบบที่พัฒนานี้ (Mean Error 22 มม.) อยู่ในระดับใกล้เคียง แต่มีข้อได้เปรียบที่ ใช้กล้องวงจรปิดเป็น Infrastructure-based Sensors ซึ่งมีต้นทุนต่ำกว่าและสามารถนำไปประยุกต์ใช้กับพื้นที่ที่มีกล้องติดตั้งอยู่แล้ว เช่น โรงพยาบาลและคลังสินค้า อย่างไรก็ตาม ระบบยังมีข้อจำกัดในสภาพที่มีแสงรบกวนสูง เงา หรือการ Occlusion ที่ทำให้การตรวจจับวัตถุยากขึ้น และยังขึ้นอยู่กับความหน่วงของเครือข่ายในการสื่อสารข้อมูล

สรุปภาพรวม

ผลการทดลองแสดงให้เห็นว่างานวิจัยนี้สามารถบูรณาการเทคนิคจากหลายองค์ประกอบได้แก่ YOLOv11, PnP, Polynomial Regression, GPR, A*, และ Pure Pursuit เข้าด้วยกันภายใต้โครงสร้างควบคุมแบบรวมศูนย์ได้อย่างมีประสิทธิภาพ โดยระบบสามารถตรวจจับ ระบุตำแหน่ง วางเส้นทาง และควบคุม AGV ได้แบบเรียลไทม์ในสภาพแวดล้อมภายในอาคาร ทั้งยังสามารถขยายไปใช้กับ multi-robot system ได้ในอนาคต ถือเป็นแนวทางที่ช่วยเพิ่มความคุ้มค่าและศักยภาพของ AGV โดยไม่ต้องพึ่งพาเซนเซอร์ราคาแพง

บทที่ 5
สรุปและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาระบบระบุตำแหน่งและนำทาง ยานพาหนะภาคพื้นดินอัตโนมัติ (AGV) ภายในอาคาร โดยใช้ กล้องวงจรปิด (Surveillance Cameras) ร่วมกับเทคนิคปัญญาประดิษฐ์ (AI) แทนการใช้เซนเซอร์ที่มีต้นทุนสูง เช่น LIDAR หรือ SLAM โดยระบบถูกออกแบบให้มีประสิทธิภาพสูง ใช้งานได้จริงในอาคารทั่วไป เช่น สำนักงาน โรงพยาบาล และคลังสินค้า

สรุปผลลัพธ์หลักที่ได้

1) ระบบตรวจจับวัตถุด้วย YOLOv11 สามารถตรวจจับวัตถุเป้าหมายได้แม่นยำ โดยมี mAP@0.5-0.95 อยู่ที่ 95.60%, Precision อยู่ที่ 98.40% และ Recall อยู่ที่ 1.00 และทำงานแบบเรียลไทม์อยู่ที่ 20 FPS

2) การระบุตำแหน่งด้วย PnP เพียงอย่างเดียว มีค่า Error เฉลี่ย 141 มม. แต่เมื่อใช้ GPR ร่วมด้วย ลดลงเหลือ 22 มม.

3) ระบบสามารถสร้างแผนที่เสมือน 2 มิติ (2D Virtual Map) แบบ Real-Time (10 Hz) จากกล้องหลายตัว และวางเส้นทางด้วย A* ได้อย่างถูกต้อง

4) การติดตามเส้นทางด้วย Pure Pursuit Algorithm ให้ความคลาดเคลื่อนเฉลี่ยน้อยกว่า 30 มม. และมี Latency ต่ำ ที่ 100 ms

5) สถาปัตยกรรม Centralized Control ช่วยลดภาระการประมวลผลบนตัว AGV และสามารถขยายระบบเพื่อรองรับหลาย AGV ได้

5.2 การมีส่วนร่วมของงานวิจัย (Contributions)

งานวิจัยนี้ได้สร้างคุณูปการใหม่และคุณค่าทางวิชาการดังนี้

1) การบูรณาการเทคนิค AI และ Computer Vision แบบใหม่ – การใช้ YOLOv11, PnP, และ GPR ร่วมกันในระบบที่ใช้กล้องวงจรปิดเป็น Sensor หลัก เพื่อแก้ปัญหาความแม่นยำและต้นทุนของการนำทางอัตโนมัติ

2) การพิสูจน์ประสิทธิภาพของ GPR ใน Localization – งานนี้ชี้ชัดว่า GPR สามารถลด Error การระบุตำแหน่งได้มากกว่า 80–90% ซึ่งเป็นหลักฐานเชิงประจักษ์สำหรับงานวิจัยด้าน Vision-based Localization

3) การออกแบบสถาปัตยกรรม Centralized Control – เสนอระบบที่ใช้กล้องวงจรปิดเป็น Infrastructure-based Sensor Network ที่ลดต้นทุน ลดภาระบนตัว AGV และสามารถขยายระบบได้ง่าย

5.3 ข้อจำกัดของงานวิจัย

แม้ระบบที่พัฒนาจะประสบความสำเร็จ แต่ยังมีข้อจำกัดบางประการ ได้แก่

1) การทดลองยังอยู่ใน สภาพแวดล้อมจำลองที่ควบคุมได้ ผลลัพธ์อาจแตกต่างเมื่อใช้ในพื้นที่จริงที่มีแสงรบกวน เงา การสะท้อน และการ Occlusion ของวัตถุ

2) การสอบเทียบกล้อง (Calibration) ต้องใช้เวลาและความแม่นยำสูง หากเกิดการเปลี่ยนตำแหน่งกล้องต้องทำการสอบเทียบใหม่

3) ระบบยังพึ่งพาเครือข่าย Wi-Fi เป็นหลัก ซึ่งอาจเกิดปัญหาหน่วงเวลาหรือแพ็กเก็ตสูญหายในสภาพแวดล้อมจริง

5.4 ข้อเสนอแนะสำหรับงานวิจัยในอนาคต

เพื่อพัฒนาต่อยอดงานวิจัยนี้ สามารถดำเนินการดังนี้

1) Dynamic Re-calibration พัฒนาอัลกอริทึมสำหรับการสอบเทียบกล้องอัตโนมัติ/กึ่งอัตโนมัติ เพื่อลดภาระการตั้งค่าใหม่เมื่อสภาพแวดล้อมเปลี่ยน

2) การทดลองในสภาพแวดล้อมซับซ้อนขึ้น โดยทดสอบระบบในพื้นที่ใหญ่ขึ้น หรือในสภาพแวดล้อมจริง เช่น โรงพยาบาลหรือคลังสินค้า

3) Multi-AGV Coordination โดยศึกษาการทำงานร่วมกันของ AGV หลายตัวภายใต้การควบคุมรวมศูนย์ เพื่อดูความสามารถในการหลีกเลี่ยงการชนและการจัดเส้นทางร่วม

4) Advanced Sensor Fusion ซึ่งผสานข้อมูลจาก IMU, Encoder หรือแม้แต่ UWB/Visual-Inertial Odometry (VIO) เพื่อเพิ่มความทนทานของระบบเมื่อกล้องไม่สามารถตรวจจับ AGV ได้

รายการอ้างอิง

Alam, M. S., Mohamed, F. B., Selamat, A. & Hossain, A. B. (2023). A Review of Recurrent Neural Network Based Camera Localization for Indoor Environments. *IEEE Access, vol. 11, pp. 43985-44009, doi: 10.1109/ACCESS.2023.3272479.*

Ballesta, M., Payá, L., Cebollada, S., Reinoso, O. & Murcia, F. (2021). A CNN Regression Approach to Mobile Robot Localization Using Omnidirectional Images. *Applied Sciences*, vol. 11, no. 16, p. 7521, doi: 10.3390/app11167521.

Barioni, W. E., Latini, I. P., Lazzaretti, A., Teixeira, M., Neves F. & De Arruda, L. V. R. (2022). AGV Detection in Industrial Environments through Computer Vision. 2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE), São Bernardo do Campo, Brazil, 2022, pp. 1-6, doi: 10.1109/LARS/SBR/ WRE568 24.2022.9995994.

Brooks, A., Makarenko, A. & Upcroft, B. (2008). Gaussian Process Models for Indoor and Outdoor Sensor-Centric Robot Localization. *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1391–1403, doi: 10.1109/TRO.2008.2007122.

Choudhari, M. S. (2021). Breast Cancer Detection using Deep Learning Techniques. *International Journal for Research in Applied Science & Engineering Technology*, vol. 9, no. VI, pp. 3959–3963, doi: 10.22214/ijraset.2021.35757.

Chen, Z., Li, X., Wang, L., Shi, Y., Sun, Z. & Sun, W. (2022). An Object Detection and Localization Method Based on Improved YOLOv5 for the Teleoperated Robot, *Application of Artificial Intelligence in Mechatronics*, vol. 12, no. 22, p. 11441, doi: 10.3390/app122211441.

Davison A. J. (2003). Real-Time Simultaneous Localisation and Mapping with a Single Camera. *Proceedings Ninth IEEE International Conference on Computer Vision, Nice, France, 2003, pp. 1403-1410 vol.2, doi: 10.1109/ICCV.2003.1238654.*

Deretey, E., Ahmed, M. T., Marshall J. A. and Greenspan, M. (2015). Visual indoor positioning with a single camera using PnP. 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Banff, AB, Canada, 2015, pp. 1-9, doi: 10.1109/IPIN.2015.7346756.

Dimanov, M. (2019). *Computer Vision: Stereo Vision and Volume Measurement*. Retrieved from: https://medium.com/@softarex/computer-vision-stereo-vision-and-volume-measurement-910c381f0f75

DragonEye AI. (2025). *A guide to bounding box formats*. Retrieved from: https://dragoneye.ai/blog/a-guide-to-bounding-box-formats

First Sensor AG. (2021). *How to use LiDAR sensors for automotive and mobility systems*. Retrieved from: https://www.azosensors.com/ article. aspx?ArticleID=1907

Hocaoglu, G. S., & Benli, E. (2025). Multiple Objects Localization With Camera-LIDAR Sensor Fusion. IEEE Sensors Journal, vol. 25, no. 7, pp. 11892-11905, 1 April1, 2025, doi: 10.1109/JSEN.2025.3541431.

Huang, F., Shen, D., Wen, W., Zhang, J. & Hsu, L-T. (2021). A Coarse-to-Fine LiDAR-Based SLAM with Dynamic Object Removal in Dense Urban Areas. the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021), St. Louis, Missouri, pp. 3162–3172. doi: 10.33012/2021.18083.

Hu, S., Wang, P., Hoare, C. and O'Donnell, J. (2022). Building Occupancy Detection and Localization Using CCTV Camera and Deep Learning. IEEE Internet of Things Journal, vol. 10, no. 1, pp. 597-608, 1 Jan.1, 2023, doi: 10.1109/JIOT.2022.3201877.

International Organization for Standardization. (2020). ISO 3691-4:2020 – Industrial trucks -Safety requirements and verification - Part 4: Driverless industrial trucks and their systems. 1st ed. Geneva: ISO; Retrieved from: https://www.iso.org/standard/70660.html

Kim, G.-H., Kim, J.-S. & Hong, K.-S. (2005). Vision-based simultaneous localization and mapping with two cameras. *Proc. 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Edmonton, Canada, 2005, pp. 1671–1676, doi: 10.1109/IROS.2005.1545496.

Kirsch, C. & Röhrig, C. (2011). Global Localization and Position Tracking of an Automated Guided Vehicle. *Proc. 18th IFAC World Congress*, Milano, Italy, pp. 14036–14041. doi:10.3182/20110828-6-IT-1002.01245

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). Imagenet Classification with Deep Convolutional Neural Networks. Communications of the ACM, Vol. 60, Issue 6, pp. 84 – 90 DOI:10.1145/3065386

Kubíˇcková, H., Jedliˇcka, K., Fiala, R. & Beran, D. (2020). Indoor Positioning Using PnP Problem on Mobile Phone Images. ISPRS International Journal of Geo-Information, 9, 368; doi:10.3390/ijgi9060368

Kümmerle, R., Pfaff, P., Triebel, R. & Burgard, W. (2007). Active Monte Carlo Localization in Outdoor Terrains Using Multi-level Surface Maps. *Autonome Mobile Systeme 2007*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 29–35. doi: 10.1007/978-3-540-74764-2_5.

Kvalsvik, P. A. (2019). *Internal Logistics in Hospitals: Automated Guided Vehicles' Affect on Hospital Layout* (Master's thesis). Trondheim, Norway: Norwegian University of Science and Technology; 2019. Retrieved from: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2622524

Liao, J.-K., Chiang, K.-W., Chang H.-W. & Li, Y.-H. (2018). Artificial Neural Networks Aided Image Localization for Pedestrian Dead Reckoning for Indoor Navigation Applications. 2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS), Wuhan, China, 2018, pp. 1-10, doi: 10.1109/UPINLBS.2018.8559833.

Lu, S., Xu, C., Zhong, R. Y. & Wang, L. (2018). A passive RFID tag-based locating and navigating approach for automated guided vehicle. *Computers & Industrial Engineering.* 2018;125:628-636. doi:10.1016/j.cie.2017.12.026.

MathWorks. (2025). *Develop visual SLAM algorithm using Unreal Engine simulation*. MATLAB & Simulink. Retrieved from: https://au.mathworks.com/help/vision/ug/develop-visual-slam- algorithm-using-unreal-engine-simulation.html

Moshayedi, A. J., Jinsong, L., & Liao, L. (2019). *AGV (automated guided vehicle) robot: Mission and obstacles in design and performance*. *Journal of Simulation & Analysis of Novel Technologies in Mechanical Engineering 12,* Retrieved from https://journals.iau.ir/article_669256_3a335ebc4024b8096eb12808cae4fd0e.pdf?utm_source=chatgpt.com

Nomura, Y., Inoue, M. & Furuta, H. (2022). Evaluation of crack propagation in concrete bridges from vehicle-mounted camera images using deep learning and image processing. *Front. Built Environ.*, vol. 8, p. 972796, doi: 10.3389/ fbuil.2022.972796.

Orbbec. (2025). *Depth camera documentation*. Retrieved from:https:// www. orbbec. com/documentation/depth-camera/

Oščádal, P., Huczala, D., Bém, J., Krys, V. & Bobovský, Z. (2020). Smart Building Surveillance System as Shared Sensory System for Localization of AGVs," *Appl. Sci.*, vol. 10, no. 23, p. 8452, doi: 10.3390/app10238452.

Oubadriss, A., Laassiri, J. & Makrani, A. E. (2024). An Overview Comparison between Convolutional Neural Networks and Vision Transformers. *Proceedings of the 7th International Conference on Networking, Intelligent Systems and Security*, Meknes AA Morocco: ACM, Apr. 2024, pp. 1–9. doi: 10.1145/3659 677.3659719.

Panagorko. (2020). *Next level ultrasonic sensor*. Arduino Project Hub. Retrieved from: https://projecthub.arduino.cc/panagorko/next-level-ultrasonic-sensor-df5768

Pawako, S., Khaewnak, N. & Srisertpol, J. (2025). Implementation of perspective-n-point techniques and YOLOv5 algorithm based on surveillance camera for localization. *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 37, no. 3, p. 1744, Mar. 2025, doi: 10.11591/ ijeecs.v37.i3.pp1744-1757.

Lilliebladh, D., & Svensson, P. (2006). Studie av materialhantering på Proton Finishing i Forsheda (Bachelor's thesis). DiVA portal, Jönköping University. Retrieved from: https://ju.diva-portal.org/smash/record.jsf?pid=diva2:4157

Lopez-Antequera, M., Petkov, N. & Gonzalez-Jimenez, J. (2016). Image-based Localization Using Gaussian Processes. *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2016, pp. 3204–3208, doi: 10.1109/IPIN.2016.7743697

Lu, X. X. (2017). A Review of Solutions for Perspective-n-Point Problem in Camera Pose Estimation," *J. Phys.: Conf. Ser.*, vol. 1087, p. 052009, 2018, doi: 10.1088/1742- 6596/1087/5/052009.

Redmon, J., Divvala, S., Girshick R. & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

Sahnoun, S., Souissi, R., Chiboub, S., Chabchoub, A., Baazaoui, M.K., Fakhfakh, A. & Derbel, F. (2025). Enhancing Localization Accuracy and Reducing Processing Time in Indoor Positioning Systems: A Comparative Analysis of AI Models. Sensors 2025, 25, 475. https://doi.org/10.3390/s25020475

Scikit-learn developers. 1.7. (2025). Gaussian Processes. scikit-learn. Retrieved from: https://scikit- learn.org/stable/modules/gaussian_process.html

Singandhupe, A. & La, H. M. (2019). A Review of SLAM Techniques and Security in Autonomous Driving. *Proc. 2019 Third IEEE Int. Conf. on Robotic Computing (IRC)*, Naples, Italy, 2019, pp. 602–607, doi: 10.1109/IRC.2019.00122

U.S. Geological Survey. (2025). *Inertial measurement unit (IMU)*. U.S. Department of the Interior.Retrieved from: https://www.usgs.gov/ centers/pcmsc/ science/inertial-measurement-unit- imu

VineSM. (2020). *Introduction to pure pursuit tracking algorithm*. Retrieved from: https://vinesmsuic.github.io/robotics-purepersuit/

Wang, Q., Zhu, J., He, C., Wang, S., Wang, X., Ren, Y. & Ye, T. T. (2025). Karatsuba Algorithm Revisited for 2D Convolution Computation Optimization. Entropy 2025, 27, 506.https://doi.org/10.3390/e27050506

Wu, L., Meng, H.-Q. M., Lin, Z., He, W., Chao, P. & Liang, H. (2009). A practical evaluation of radio signal strength for mobile robot localization. *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Guilin: IEEE, pp. 516–522. doi: 10.1109/ROBIO.2009.5420700.

Yeong, D. J., Velasco-Hernandez, G., Barry, J. & Walsh, J. (2021). Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review. *Sensors*, vol. 21, no. 6, p. 2140, Mar. 2021, doi: 10.3390/s21062140.

Yusefi, A., Durdu A. & Toy, I. (2024). "Camera/LiDAR Sensor Fusion-based Autonomous Navigation," 2024 23rd International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, Bosnia and Herzegovina, pp. 1-6, doi: 10.1109/INFOTEH 60418.2024.10495974.

Zhao, L. & Zhang, Z. (2024). A improved pooling method for convolutional neural networks. *Sci Rep*, vol. 14, no. 1, p. 1589, Jan. 2024, doi: 10.1038/s41598-024-51258-6.

Zhang, Y., Hu, H., Lu, J. & Shi, Z. (2022). Research on Path Planning of Mobile Robot Based on Improved Theta* Algorithm. *Algorithms*, vol. 15, no. 12, p. 477, doi: 10.3390/a15120477.

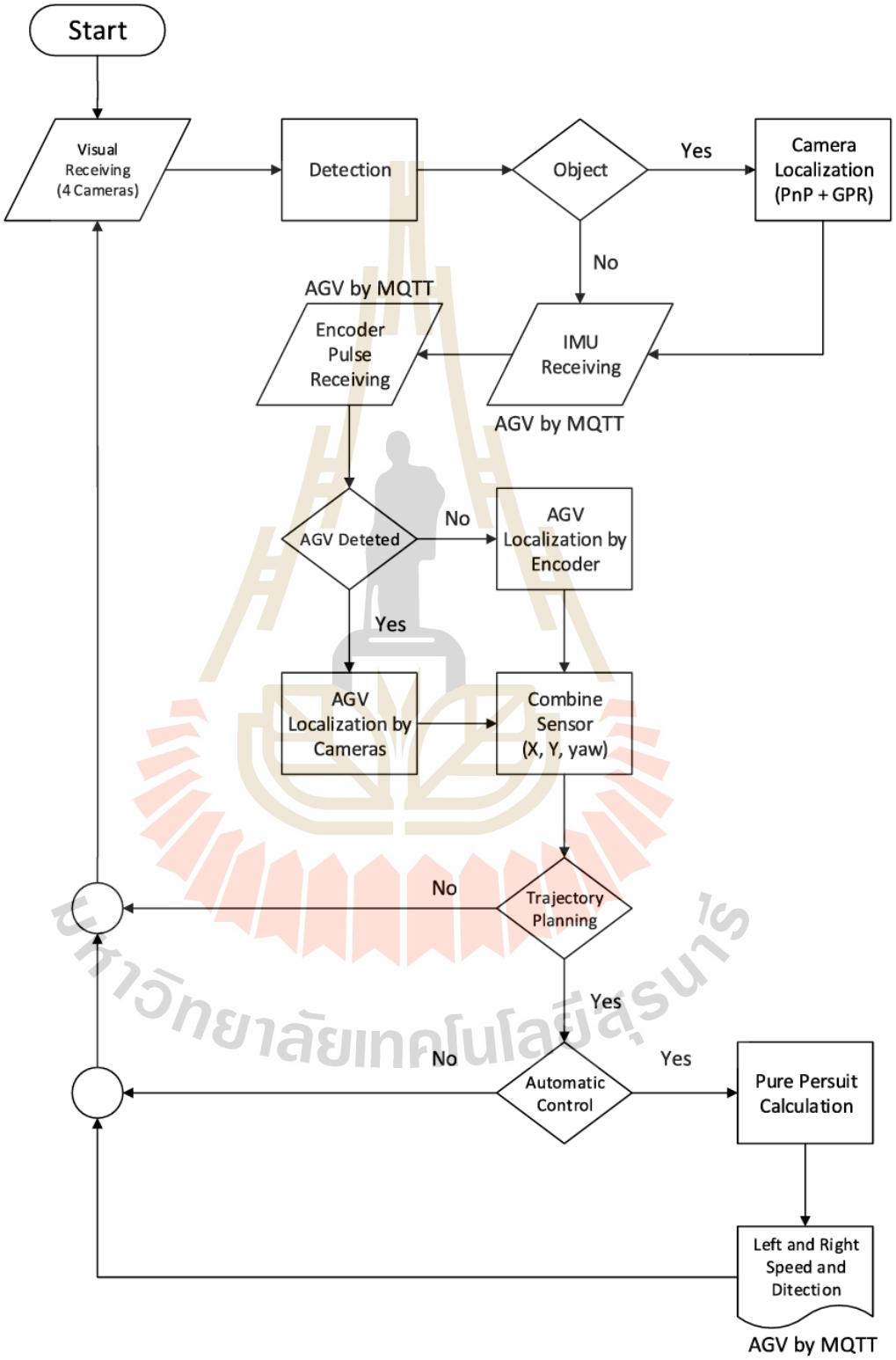
Zhang, Z., Chen, J., & Guo, Q. (2023). *Application of Automated Guided Vehicles in Smart Automated Warehouse Systems: A Survey, Computer Modeling in Engineering & Sciences, vol.134, no.3, pp 1529-1563, DOI:10.32604/cmes. 2022.021451*

Zhang, G., Zhu, L., Ji, S. and Wu, X. (2023). Target Detection and Position Measurement Based on Machine Vision for AGV. *Proc. 2023 8th Int. Conf. on Automation, Control and Robotics Engineering (CACRE)*, pp. 343–347, doi: 10.1109/ CACRE58689.2023.10208667

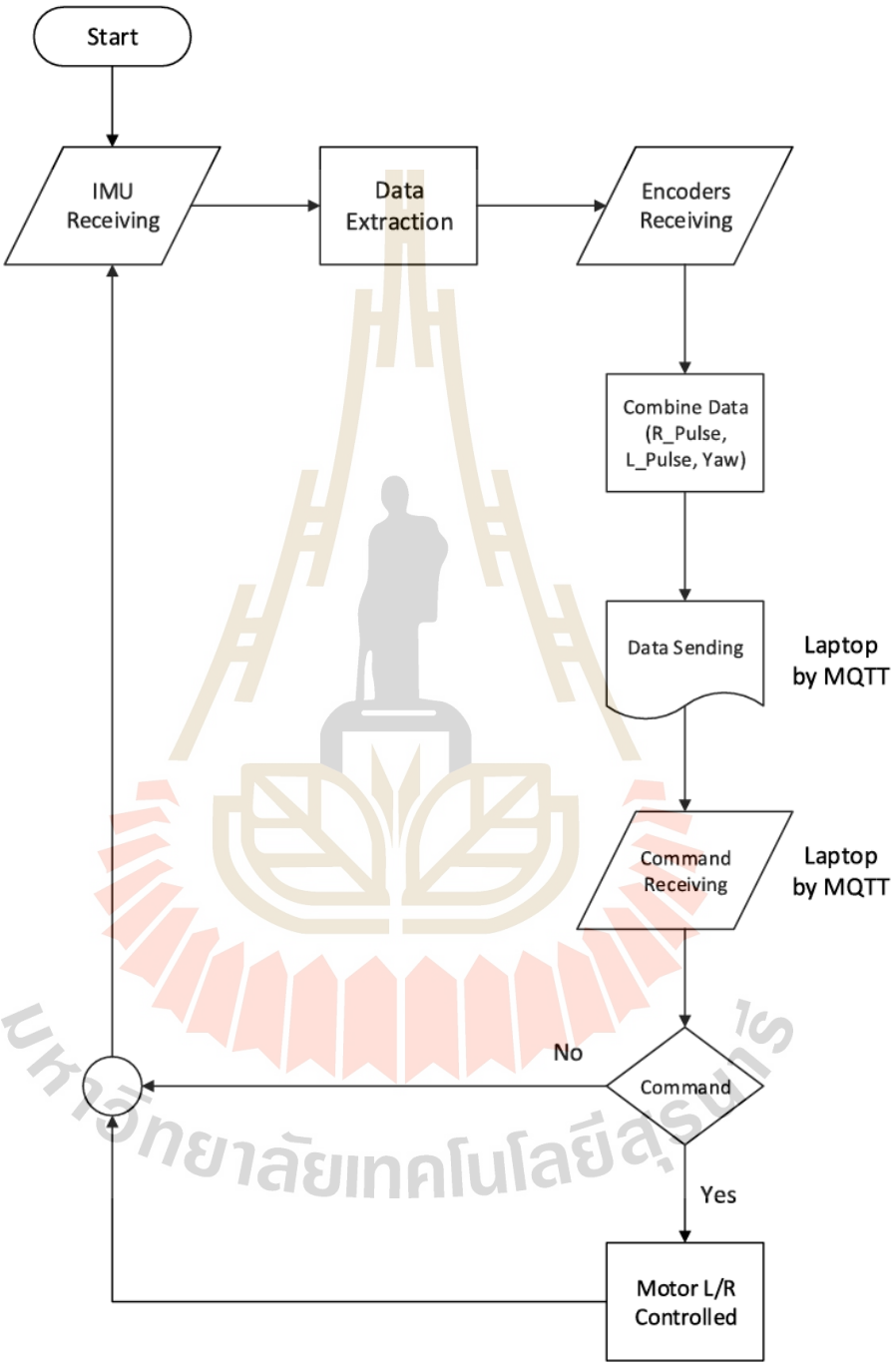
ภาคผนวก

แผนภาพการทำงานของโปรแกรม

แผนภาพการทำงานของโปรแกรม Centralized Control บน Laptop



แผนภาพการทำงานของโปรแกรมการควบคุมและสื่อสารบน AGV



ประวัติผู้เขียน

นายศิริพงษ์ ปะวะโก เกิดเมื่อวันที่ 8 กรกฎาคม พ.ศ. 2536 ที่จังหวัดกรุงเทพมหานคร จบการศึกษาระดับประถมศึกษาจากโรงเรียนวัดบุณยประดิษฐ์ จังหวัดกรุงเทพมหานคร จบการศึกษาระดับมัธยมศึกษาตอนต้นจากโรงเรียนวัดบุณยประดิษฐ์ จังหวัดกรุงเทพมหานคร จบการศึกษระดับมัธยมศึกษาตอนปลายจากโรงเรียนราชวินิตบางแคปานขำ จังหวัดกรุงเทพมหานคร และสำเร็จการศึกษาได้รับปริญญา วิศวกรรมศาสตรบัณฑิต (วิศวกรรมเครื่องกล) หลักสูตร 4 ปี ประจำปีการศึกษา 2558 มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา เมื่อปี พ.ศ. 2559 และต่อมาได้สำเร็จการศึกษาปริญญาวิศวกรรมศาสตร์มหาบัณฑิต (วิศวกรรมเมคคาทรอนิกส์) หลังจากนั้นได้เข้ารับการศึกษาต่อในระดับวิศวกรรมศาสตร์ดุษฎีบัณฑิต (วิศวกรรมเมคคาทรอนิกส์) ประจำปีการศึกษา 2561 ณ สถาบันเดิมโดยได้รับทุนจากสำนักงานวิจัยแห่งชาติ (วช.) ร่วมกับบริษัทสุรนารีแพทย์ภัณฑ์ จำกัด ในโครงการพัฒนานักวิจัย และงานวิจัยเพื่ออุตสาหกรรม (พวอ.) ปัจจุบันมีผลงานตีพิมพ์ในระดับนานาชาติจำนวน 3 ฉบับ และระดับชาติ 1 ฉบับ รวมถึงผลงานประชุมวิชาการทั้งในระดับชาติและนานาชาติกว่า 10 ฉบับ