



ระบบรักษาความปลอดภัยของหุ่นยนต์มิทซูบิชิแบบใหม่

โดยการมองเห็นอย่างชาญฉลาดแบบสามมิติ

A New Security System of Mitsubishi Robot using Intelligent 3D Vision

พานุพงศ์ ตันติประเสริฐศรี

Panupong Tantiprasertsri

อธิภัทร พันธ์นาชี

Atipat Panatnachi

สุเมธ เพ็ญผ่อง

Sumeth penphong

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยศรีนครินทรวิโรฒ ปีการศึกษา พ.ศ. 2563



คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

ชื่อหัวข้อโครงการ

ระบบรักษาความปลอดภัยของหุ่นยนต์มิซูบิชิแบบใหม่

โดยการมองเห็นอย่างชาญฉลาดแบบสามมิติ

A New Security System of Mitsubishi Robot using
Intelligent 3D Vision

นิสิต

นายภานุพงศ์ ตันติประเสริฐศรี 60102010112

นายอิทธิพร พันธ์นาชี 60102010349

นายสุเมธ เพ็ญผ่อง 60102010585

ปริญญา

วิทยาศาสตร์บัณฑิต (วท.บ.)

ภาควิชา

วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาโครงการ

อ.ดร.วีระ สอึ้ง

ลงชื่อ.....

(อ.ดร.วีระ สอึ้ง)

อาจารย์ที่ปรึกษาโครงการ

บทคัดย่อ

เทคโนโลยีหุ่นยนต์ของปัจจุบันและในอนาคตจะถูกพัฒนาขึ้นเพื่อให้ มนุษย์มีความสะดวกสบาย และมีความปลอดภัยในการดำรงชีวิตมากขึ้น มากไปกว่านั้นเทคโนโลยีหุ่นยนต์ยังมีความสามารถในการทำงานได้อย่างอัตโนมัติอีกด้วย แต่การที่เทคโนโลยีหุ่นยนต์ที่ทำงานอย่างอัตโนมัตินี้จะส่งผลให้เกิดอันตรายกับมนุษย์ได้เช่นกัน ถ้ามนุษย์มีการลุล้ำเข้าไปในพื้นที่หรือบริเวณที่หุ่นยนต์กำลังทำงานอยู่

ดังนั้นโครงการนี้จึงได้พัฒนาระบบที่จะช่วยป้องกันและแจ้งเตือน เมื่อเกิดการลุล้ำของมนุษย์ ที่กำลังเข้ามาในพื้นที่การทำงานของหุ่นยนต์อย่างอัตโนมัติโดยการนำเทคโนโลยีของกล้องสามมิติ (3D Camera) ที่มีความสามารถตรวจวัดระยะทางของสิ่งของต่างๆระหว่างตำแหน่งที่ติดตั้งกล้องและวัตถุที่กล้องกำลังมองอยู่ได้ ระบบที่จะพัฒนาขึ้นมานี้จะตรวจจับสิ่งแปลกปลอมที่เป็นมนุษย์และยังรวมไปถึงวัตถุแปลกปลอมอื่นๆที่เข้ามาใกล้หุ่นยนต์ในพื้นที่ที่กำหนด โดยจะแจ้งเตือนวัตถุที่เข้าใกล้หุ่นยนต์ให้รับรู้ถึงการลุล้ำเข้ามาในพื้นที่หวงห้าม ซึ่งจะแบ่งการแจ้งเตือนออกเป็น 3 ระยะ คือ ระยะเริ่มต้นที่จะแจ้งเตือนเป็นสัญญาณไฟ ระยะที่สองจะทำการแจ้งเตือนเป็นสัญญาณไฟและเสียง และจะสั่งให้หุ่นยนต์ทำงานช้าลง และในระยะสุดท้ายจะแจ้งเตือนด้วยสัญญาณไฟและเสียง และจะส่งสัญญาณไปยังห้องควบคุมหุ่นยนต์เพื่อให้เจ้าหน้าที่เข้าทำการตรวจสอบ มากไปกว่านั้นระบบจะสั่งให้ หุ่นยนต์หยุดการทำงานทันที เพื่อป้องกันการเกิดอันตรายที่จะเกิดขึ้นได้ในอนาคต

จากการทดลองนำระบบที่พัฒนาขึ้นใช้ร่วมกับหุ่นยนต์ พบว่าระบบสามารถทำการป้องกัน อันตรายที่จะเกิดขึ้นได้อย่างมีประสิทธิภาพ และสามารถทำการแจ้งเตือนการลุล้ำเข้าพื้นที่ทำงานของหุ่นยนต์ได้อย่างอัตโนมัติอีกด้วย

Abstract

The current and future of automated robotic technology will be developed to make human beings more comfortable and safer to use. However, this technology can be harmful to humans when intruding on the working area of the robot. So, this project proposed a system to automatically prevent and alert when a human intrudes on the working robotics area. The proposed system uses a threedimensional camera technology that can measure the distance between the position of the camera and the object to detect the intrusion in the working robot area. For an intrusion, this system will alert the approaching objects and recognize the intrusion types into the restricted area that will be divided into 3 stages. The first stage is the far intrusion that will alert as a light signal. The second stage is the middle intrusion that will provide a light signal and sound alert and then command the robot to slow down. The last stage is the near intrusion that will alert by lights and sounds. In addition, the proposed system will send a signal to the robot control room for officers to investigate and then will immediately stop the robot to prevent the occurrence of harm that may occur in the future. From the experimental results, the developed system was integrated with the robot that has been found the system can effectively prevent any harm caused and automatically alert the intrusion into the robotic workspace.

สารบัญ

บทคัดย่อ.....	3
Abstarct.....	4
สารบัญ.....	5
สารบัญรูปภาพ.....	7
สารบัญตาราง.....	8
บทที่ 1.....	9
บทนำ.....	9
1.1 ที่มาและความสำคัญ.....	9
1.2 วัตถุประสงค์.....	9
1.3 ขอบเขตงานวิจัย.....	9
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	10
บทที่ 2.....	11
องค์ความรู้ที่เกี่ยวข้อง.....	11
2.1 อุปกรณ์เก็บข้อมูล.....	11
2.2 การเขียนคำสั่งควบคุมการทำงานกล้อง.....	12
2.3 การเขียนคำสั่งควบคุมการทำงานหุ่นยนต์.....	13
บทที่ 3.....	14
วิธีการดำเนินงาน.....	14
3.1 ขั้นตอนการดำเนินงาน.....	14
3.2 แผนการดำเนินงานตลอดโครงการ.....	15
3.3 อุปกรณ์และเครื่องมือที่ใช้ในการทำวิจัย.....	16
3.4 การออกแบบระบบ.....	17
3.5 พื้นที่การทำงานของชุดทดลอง ในมหาวิทยาลัยศรีนครินทรวิโรฒ.....	19

3.6 ปัญหาและอุปสรรค	20
บทที่ 4	21
ผลการดำเนินงาน	21
4.1 อบรมและปรับพื้นฐานจากหลักสูตรต่างๆ	21
บทที่ 5	28
สรุปผล อภิปรายผล และข้อเสนอแนะ	28
5.1 สรุปผลการศึกษาค้นคว้า	28
5.2 อภิปรายผล	28
5.3 ข้อเสนอแนะ	28
บรรณานุกรม	32
ภาคผนวก	34

สารบัญรูปภาพ

ภาพประกอบที่ 1 Intel Realsense D435.....	11
ภาพประกอบที่ 2 Intel Realsense D435.....	11
ภาพประกอบที่ 3 RGB module	12
ภาพประกอบที่ 4 Face Detection	12
ภาพประกอบที่ 5 GX Work3	13
ภาพประกอบที่ 6 ตำแหน่งกล้อง	18
ภาพประกอบที่ 7 การเชื่อมต่ออุปกรณ์ต่างๆ เข้าด้วยกัน.....	19
ภาพประกอบที่ 8 อบรมหลักสูตร	22
ภาพประกอบที่ 9 อบรมหลักสูตร	22
ภาพประกอบที่ 10 ผลการดำเนินงาน 1	23
ภาพประกอบที่ 11 ผลการดำเนินงาน 2	24
ภาพประกอบที่ 12 ผลการดำเนินงาน 3	25
ภาพประกอบที่ 13 ผลการดำเนินงาน 4	26
ภาพประกอบที่ 14 ผลการดำเนินงาน 5	27
ภาพประกอบที่ 15 กล้องควบคุมการทำงาน.....	29
ภาพประกอบที่ 16 interface ภายในโปรแกรม GX Work 3.....	30
ภาพประกอบที่ 17 interface ภายในโปรแกรม GT Designer 3	31

สารบัญตาราง

ตารางที่ 1 แผนการดำเนินงาน.....	15
ตารางที่ 2 อบรมและปรับพื้นฐานหลักสูตร.....	21



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

เนื่องจากภาคเรียนที่ผ่านมากลุ่มของนิสิตได้ทำโครงการเกี่ยวกับกล้องที่สามารถเก็บข้อมูลพื้นที่และแสดงผลเป็น 3 มิติ ซึ่งเบื้องต้นสามารถทำงานได้ในระดับปานกลาง คือ สามารถแสดงผลให้พอเข้าใจได้ว่าพื้นที่นั้น ๆ คือพื้นที่ไหน มีสัดส่วนของห้องค่อนข้างชัดเจน โดยต่อมาในภาคเรียนนี้ กลุ่มนิสิตมีความสนใจที่จะนำโครงการในภาคเรียนที่ผ่านมาไปประสานเข้ากับแขนกลเพื่อการทำงานร่วมกันอย่างมีประสิทธิภาพมากขึ้น ทั้งในด้านคุณภาพและในด้านความปลอดภัย โดยจะนำเอาความสามารถของกล้องที่มีการวัดระยะเชิงลึกมาใช้ประโยชน์ในการวัดระยะห่างจากแขนกลสู่บุคลากรผู้ทำงานกับแขนกลนั้น เพื่อความปลอดภัยของผู้เข้าใกล้พื้นที่ทำงานและความปลอดภัยของทรัพย์สิน ทั้งยัง สามารถจดจำใบหน้าผู้ทำงานในระบบด้วยได้ และอาจทำให้แขนกลหลบหลีกผู้ที่เข้าใกล้หรือผ่านพื้นที่การทำงานของแขนกลได้

1.2 วัตถุประสงค์

- 1.2.1 ตรวจจับสิ่งที่เข้ามาใกล้หุ่นยนต์ได้
- 1.2.2 วัดระยะห่างระหว่างหุ่นยนต์กับคนได้
- 1.2.3 เก็บข้อมูลสิ่งที่เข้ามาใกล้หุ่นยนต์ได้
- 1.2.4 ป้องกันการเกิดความอันตรายหรือความเสียหายที่จะเกิดขึ้นในอนาคตได้
- 1.2.5 เพื่อพัฒนาระบบใหม่ในการป้องกันหุ่นยนต์ของบริษัทมิติซูบิซี อิเล็กทรอนิกส์ แพลทอรี ออโตเมชันได้

1.3 ขอบเขตงานวิจัย

- 1.3.1 การใช้กล้อง Intel® Realsense™ Depth Camera D435 ในการเก็บข้อมูล
- 1.3.2 การเขียนโปรแกรมเพื่อควบคุมกล้อง เพื่อให้ทำงานร่วมกับหุ่นยนต์และควบคุมหุ่นยนต์ได้
- 1.3.3 การเขียนโปรแกรมเพื่อทำงานและควบคุมหุ่นยนต์ของบริษัทมิติซูบิซี อิเล็กทรอนิกส์ แพลทอรี ออโตเมชัน

เมชั่น

- 1.3.4 การเชื่อมต่อระบบหุ่นยนต์กับระบบภายนอก

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 สามารถสร้างแผนที่ 3 มิติ ที่เข้าใจง่ายและมีความแม่นยำ
- 1.4.2 ลดการใช้แรงงานมนุษย์ ด้วยระบบอัตโนมัติและหุ่นยนต์
- 1.4.3 ป้องกันการเกิดความอันตรายหรือความเสียหายที่จะเกิดขึ้นกับคนและหุ่นยนต์ได้



บทที่ 2

องค์ความรู้ที่เกี่ยวข้อง

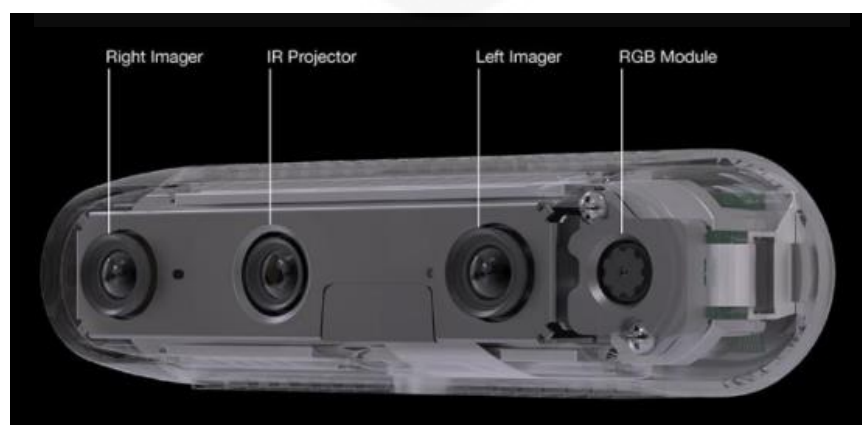
2.1 อุปกรณ์เก็บข้อมูล

2.1.1 Intel® RealSense™ Depth Camera D435

การเก็บข้อมูลสภาพแวดล้อมเพื่อนำมาทำเป็นแผนที่ 3 มิติ นั้นมีคุณภาพหากใช้อุปกรณ์ที่มีการเก็บข้อมูลความลึก โดยงานวิจัยนี้เลือกใช้ Intel RealSense รุ่น D435 ซึ่งเป็นกล้องที่มีระบบวัดความลึก มาในรูปแบบกล้องคู่ พร้อม RGB module [1] [2]



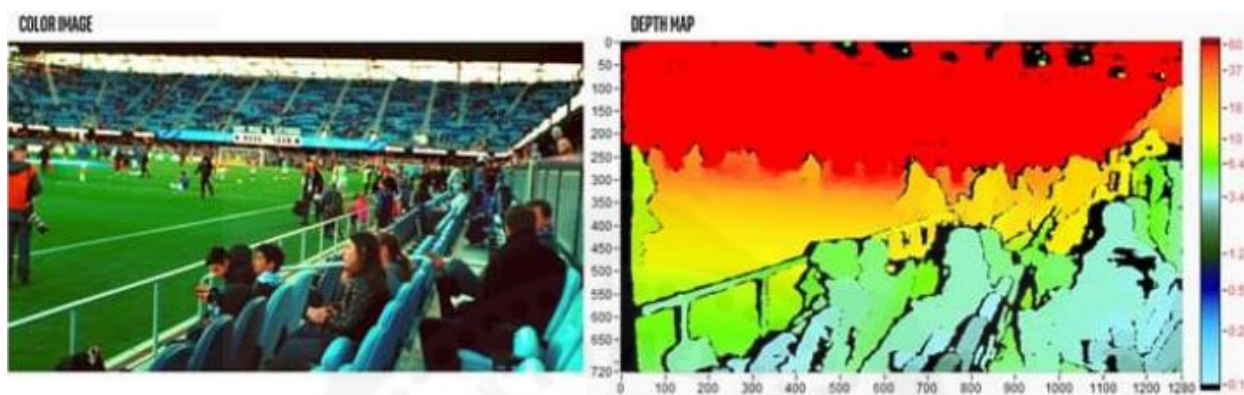
ภาพประกอบที่ 1 Intel RealSense D435



ภาพประกอบที่ 2 Intel RealSense D435

2.1.2 RGB module

กล้องที่แสดงผลด้วยภาพ 2 มิติ ในแต่ละ pixel จะเป็นค่าสี RGB (Red , Green , Blue) ตั้งแต่ 0 – 255 ซึ่ง pixel จะมีค่าที่ต่างกันตามระยะห่างจากกล้องหรือก็คือ ความลึกของวัตถุ แสดงให้เห็นด้วยค่าสี ในตัวอย่างนี้ สีฟ้าจะเป็นส่วนที่อยู่ใกล้กล้องมากที่สุด และสีแดงจะเป็นส่วนที่ไกลที่สุด



ภาพประกอบที่ 3 RGB module

2.2 การเขียนคำสั่งควบคุมการทำงานกล้อง

2.2.1 การตรวจจับใบหน้า (Face Detection)

คือกระบวนการค้นหาใบหน้าของบุคคลจากภาพหรือวิดีโอ หลังจากนั้นก็จะทำการประมวลผลภาพ ใบหน้าที่ได้สำหรับนำไปใช้ในขั้นตอนถัดไป อัลกอริทึมที่ใช้ในการตรวจจับใบหน้าในปัจจุบันก็มีอยู่ด้วยกันหลายวิธี ซึ่งอัลกอริทึมในการตรวจจับใบหน้าที่ดีนั้นมีส่วนช่วยในการจำแนก



ภาพประกอบที่ 4 Face Detection

บทที่ 3

วิธีการดำเนินงาน

3.1 ขั้นตอนการดำเนินงาน

- 3.1.1 ศึกษาค้นคว้าเนื้อหาและงานวิจัยที่เกี่ยวข้องกับหัวข้อที่จะทำ
- 3.1.2 เลือก จัดหาอุปกรณ์และโปรแกรมที่จะใช้ตามคำแนะนำต่าง ๆ ของงานวิจัย
- 3.1.3 อบรมและปรับพื้นฐานจากหลักสูตรต่างๆ
 - 3.1.3.1 Basic Robot (iQ-Platform)
 - 3.1.3.2 Basic PLC(GXW3)
 - 3.1.3.3 Basic GOT
- 3.1.4 พัฒนาระบบ Intelligent 3D Vision
- 3.1.5 เชื่อมต่อระบบกับระบบของหุ่นยนต์
- 3.1.6 ทดสอบการทำงานทั้งหมด
- 3.1.7 ตรวจสอบคุณภาพและความถูกต้อง
- 3.1.8 ปรับปรุงแก้ไขระบบ
- 3.1.9 ทดสอบอีกครั้ง
- 3.1.10 สรุปผลวิจัย

3.2 แผนการดำเนินงานตลอดโครงการ

แผนการดำเนินงานวิจัย	ปีการศึกษา 2563			
	ม.ค. 64	ก.พ. 64	มี.ค. 64	เม.ย. 64
1) วางแผนโครงการ				
เลือกหัวข้อโครงการที่จะพัฒนา				
ศึกษาแนวคิดและเทคโนโลยีที่จะนำมาใช้				
2) อบรมและปรับพื้นฐานหลักสูตร				
Basic Robot (iQ-Platform)				
Basic PLC(GXW3)				
Basic GOT				
3) ขั้นตอนการพัฒนา				
พัฒนาระบบ Intelligent 3D Vision				
เชื่อมต่อระบบกับระบบของหุ่นยนต์				
4) ทดสอบประสิทธิภาพ				
ทดสอบประสิทธิภาพโดยรวม				
ปรับปรุงแก้ไขระบบ				
สรุปและประเมินผล				
9. สรุปผลงาน เพื่อนำเสนอ				

3.3 อุปกรณ์และเครื่องมือที่ใช้ในการทำวิจัย

3.3.1 อุปกรณ์

3.3.1.1 Computer

3.3.1.2 Intel Realsense D435

3.3.1.3 บอร์ด NVIDIA® Jetson Nano™

3.3.2 ซอฟต์แวร์

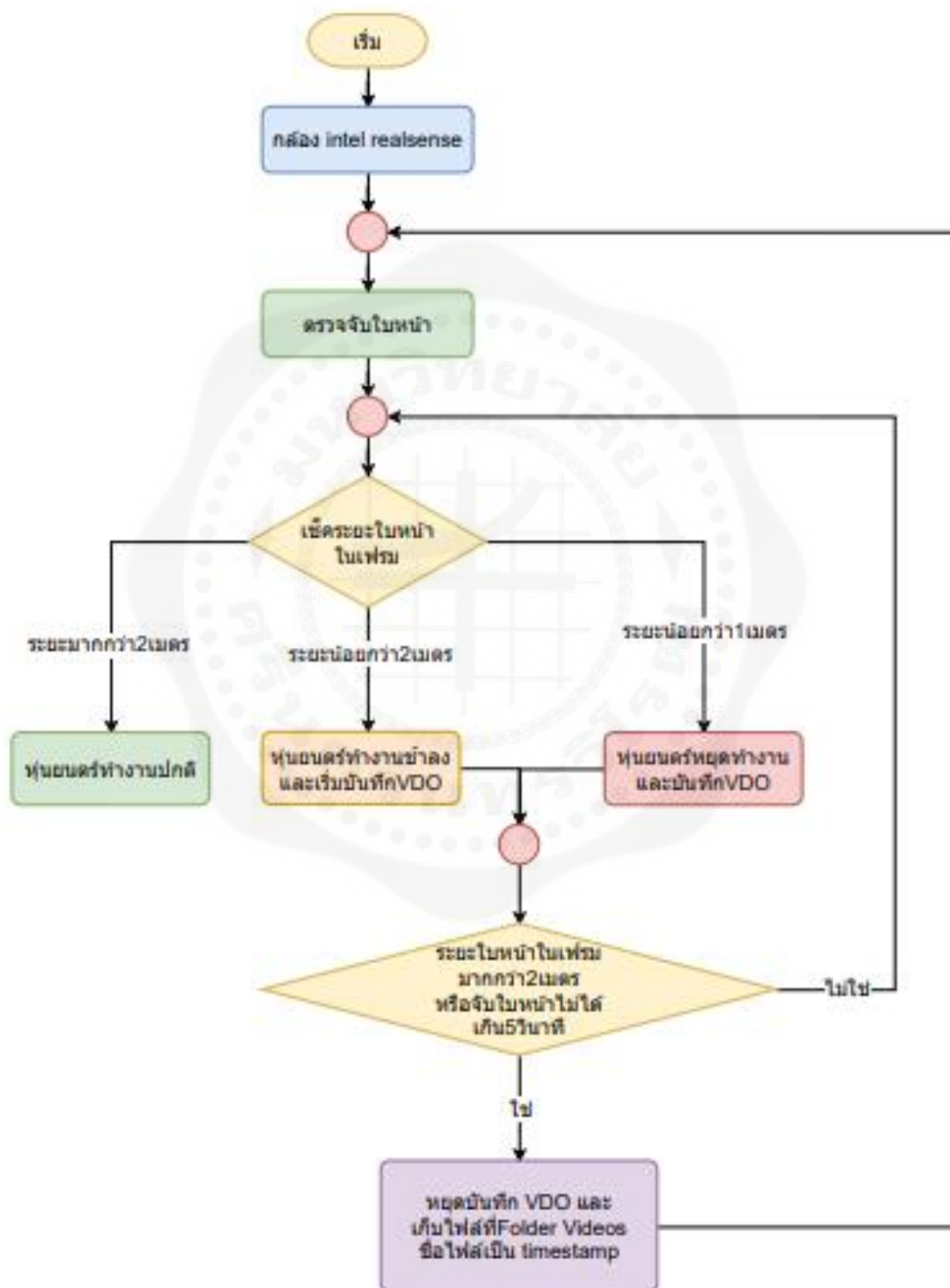
3.3.2.1 IntelRealsense Viewer

3.3.2.2 Pycharm

3.3.3 ภาษาที่ใช้

3.3.3.1 Python

3.4 การออกแบบระบบ



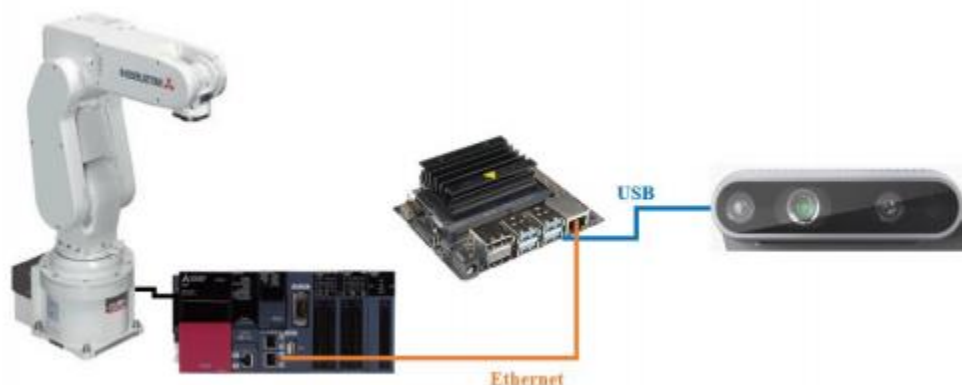
การทำงานของระบบ

โครงการที่กำลังจะพัฒนาขึ้นมาใหม่นี้ จะมีจุดประสงค์เพื่อการตรวจสอบการล่วงล้ำเข้าพื้นที่ ในการทำงานของหุ่นยนต์ โดยภาพรวมทั้งหมดจะแสดงในภาพต่อไปนี้



ภาพประกอบที่ 6 ตำแหน่งกล้อง

โครงการนี้จะพัฒนาระบบ Intelligent 3D Vision (I3V) ด้วยการใช้กล้องสามมิติรุ่น Intel Realsense D435 ที่มีความสามารถมองเห็นวัตถุในรูปแบบสามมิติ (Depth Vision) ได้ มากกว่านั้นยังสามารถตรวจสอบระยะทาง (Measurement Distance) ของวัตถุต่างๆที่เข้าใกล้กล้องได้อีกด้วย จึงนำอุปกรณ์นี้มาเพื่อพัฒนาระบบรักษาความปลอดภัย (Security System) ของหุ่นยนต์มิติซูบิ จี เพื่อป้องกันอันตรายที่จะเกิดขึ้นกับมนุษย์และความเสียหายที่อาจจะเกิดขึ้นกับอุปกรณ์ได้อีกด้วย โดยระบบที่จะพัฒนาขึ้นจะมีการเชื่อมต่อกับหุ่นยนต์ดังภาพต่อไปนี้



ภาพประกอบที่ 7 การเชื่อมต่ออุปกรณ์ต่างๆ เข้าด้วยกัน

ระบบที่พัฒนาขึ้นใหม่จะนำคอมพิวเตอร์ขนาดเล็กที่มีชื่อว่า NVIDIA Jetson Nano ในการเชื่อมต่อกับ กล้องสามมิติ และทำการสร้างระบบความฉลาดที่อุปกรณ์นี้ หลังจากนั้นจึงส่งข้อมูลผ่านทางระบบ Ethernet หรือระบบการเชื่อมต่ออื่น เพื่อติดต่อสื่อสารกับหุ่นยนต์ในการบอกสถานะของสิ่งต่างๆ ที่ เข้าพื้นที่รักษาความปลอดภัย (Security Area) หรือมากกว่านั้นเข้าถึงพื้นที่ทำงานของหุ่นยนต์ (Working Area) หลังจากนั้นจะส่งสัญญาณให้กับหุ่นยนต์ให้ทำงานช้าลงหรือหยุดทำงาน เพื่อป้องกัน อันตรายหรือความเสียหายที่จะเกิดขึ้นในอนาคตได้

3.5 พื้นที่การทำงานของชุดทดลอง ในมหาวิทยาลัยศรีนครินทรวิโรฒ



3.6 ปัญหาและอุปสรรค

3.5.1. ขาดความรู้ในเรื่องหุ่นยนต์จึงต้องเข้าคอร์สอบรมหลักสูตรและระยะเวลาในการอบรมก็ไม่สามารถกำหนดได้ด้วยตัวนิสิตเอง จึงต้องรอเวลาจากฝั่งผู้อบรมเพียงอย่างเดียว

3.5.2. เนื่องด้วยการแพร่ระบาดของ COVID-19 ทำให้นิสิตไม่สามารถใช้แขนกลที่มีภายในตึกคณะวิทยาศาสตร์ได้ จึงต้องเปลี่ยนรูปแบบโครงงานเป็นแบบจำลองการทำงานแทน

3.5.3. มาตรการของทางมหาวิทยาลัย ทำให้เจ้าหน้าที่ที่กลุ่มนิสิตได้ประสานงานไว้ในเรื่องการเชื่อมต่อแขนกลกับกล้อง ไม่สามารถเข้าไปทำงานได้

3.5.4. ความคลาดเคลื่อนของระยะจริง และระยะจากกล้องถึงบุคคลที่กล้องตรวจจับได้

3.5.5. ความแม่นยำของการตรวจจับมีคลาดเคลื่อนเล็กน้อย หากเห็นหน้าไม่ชัดเจน หรือมีสิ่งอื่นสิ่งใดปิดบังหน้าเล็กน้อย



บทที่ 4

ผลการดำเนินงาน

4.1 อบรมและปรับพื้นฐานจากหลักสูตรต่างๆ

Course	วันที่	เวลา
Basic Robot (iQ-Platform)	3-4 มีนาคม 2564	9.00 - 16.30 น.
Basic PLC(GXW3)	8-9 มีนาคม 2564	9.00 - 16.30 น.
Basic GOT	11-12 มีนาคม 2564	9.00 - 16.30 น.

ตารางที่ 2 อบรมและปรับพื้นฐานหลักสูตร

- Basic Robot (iQ-Platform)

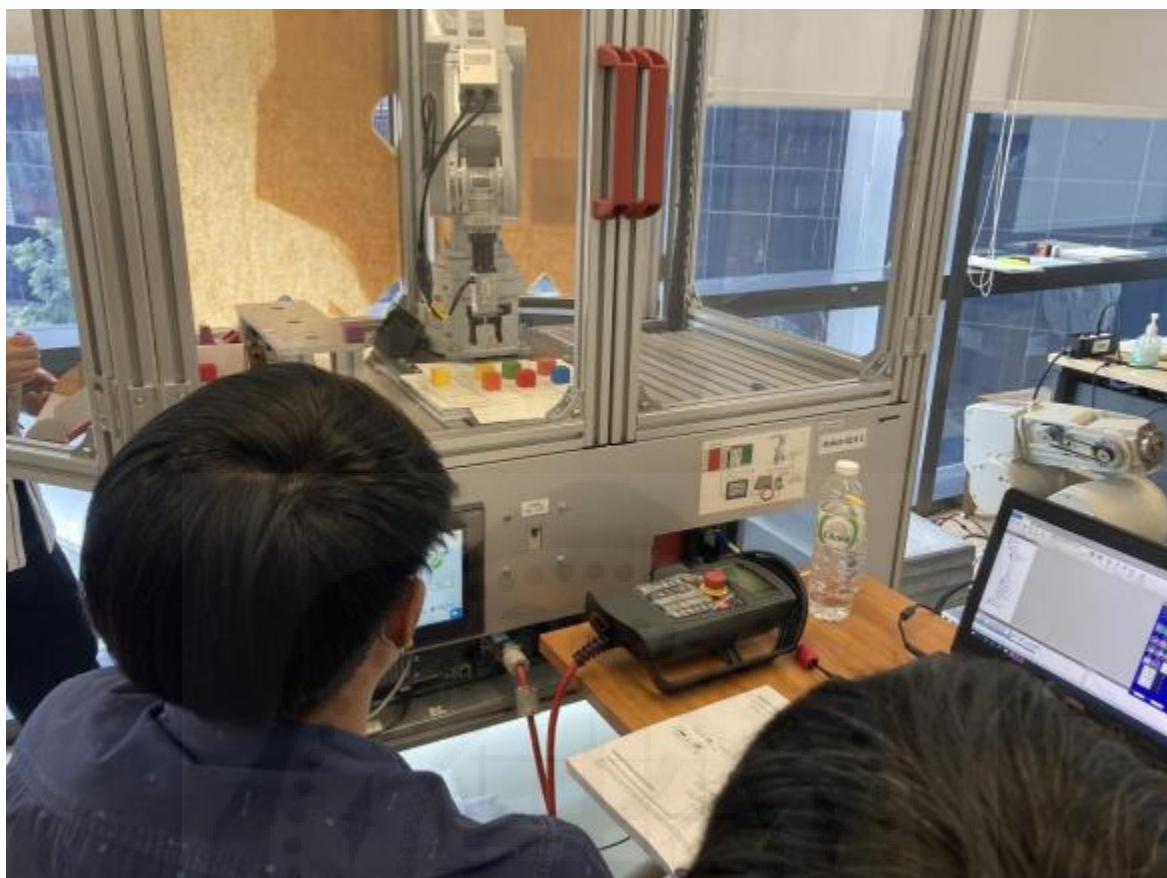
คอร์สนี้ นิสิตได้เรียนความรู้พื้นฐานในด้านการใช้งาน Robot iQ-Platform รวมทั้งสามารถตรวจสอบและแก้ไขปัญหาเบื้องต้นได้อีกด้วย นิสิตได้ทดลองใช้อุปกรณ์และชุดทดลองจริงด้วย ทำให้สามารถรู้วิธีควบคุมและแก้ปัญหาเบื้องต้นได้ เมื่อเกิดปัญหาระหว่างการดำเนินงานโครงการ

- Basic PLC (GXW3)

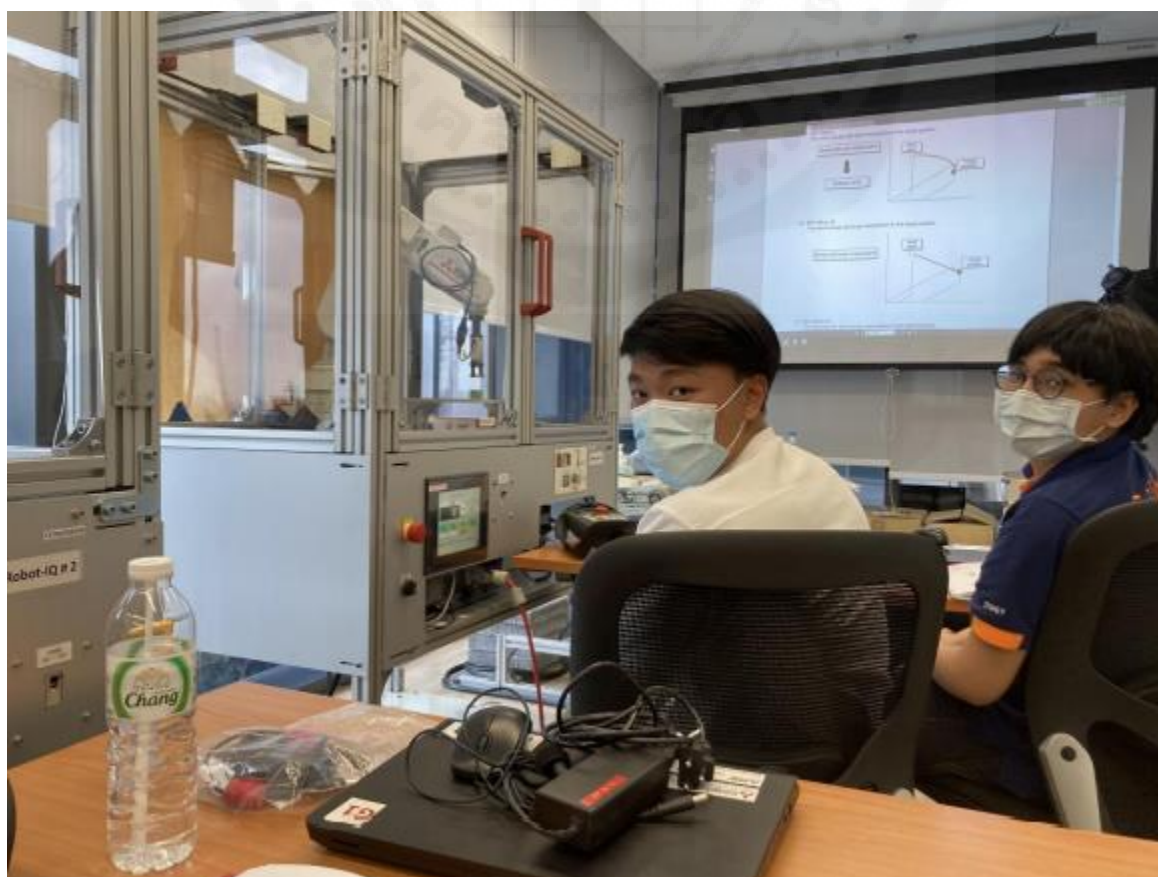
คอร์สนี้ นิสิตได้อบรมการใช้งาน PLC หลักสูตร Basic PLC (GX Works 3) โดยมีการฝึกปฏิบัติการออกแบบ และเขียนโปรแกรม PLC และฝึกปฏิบัติการ การใช้โปรแกรม GX Works3 และทราบถึงอุปกรณ์ PLC ขั้นพื้นฐาน ทำให้สามารถรู้วิธีการเขียนโปรแกรม PLC เพื่อใช้เชื่อมต่อและสั่งงานกับโครงการที่นิสิตได้ดำเนินการอยู่

- Basic GOT

คอร์สนี้ ทำให้นิสิตได้รู้ข้อมูลที่ทำหน้าที่เป็นสื่อกลางระหว่างเครื่องจักรและมนุษย์ ใช้เป็นตัวแสดงผลกราฟิก เช่น หลอดไฟ ตัวเลข ข้อความ รูปภาพ และนิสิตได้ใช้อุปกรณ์และชุดทดลองจริงในการฝึกออกแบบกราฟิก ทำให้นิสิตได้มีความรู้และสามารถออกแบบกราฟิกได้เอง เพื่อใช้งานร่วมกับโครงการที่นิสิตได้ดำเนินการอยู่



ภาพประกอบที่ 8 อบรมหลักสูตร



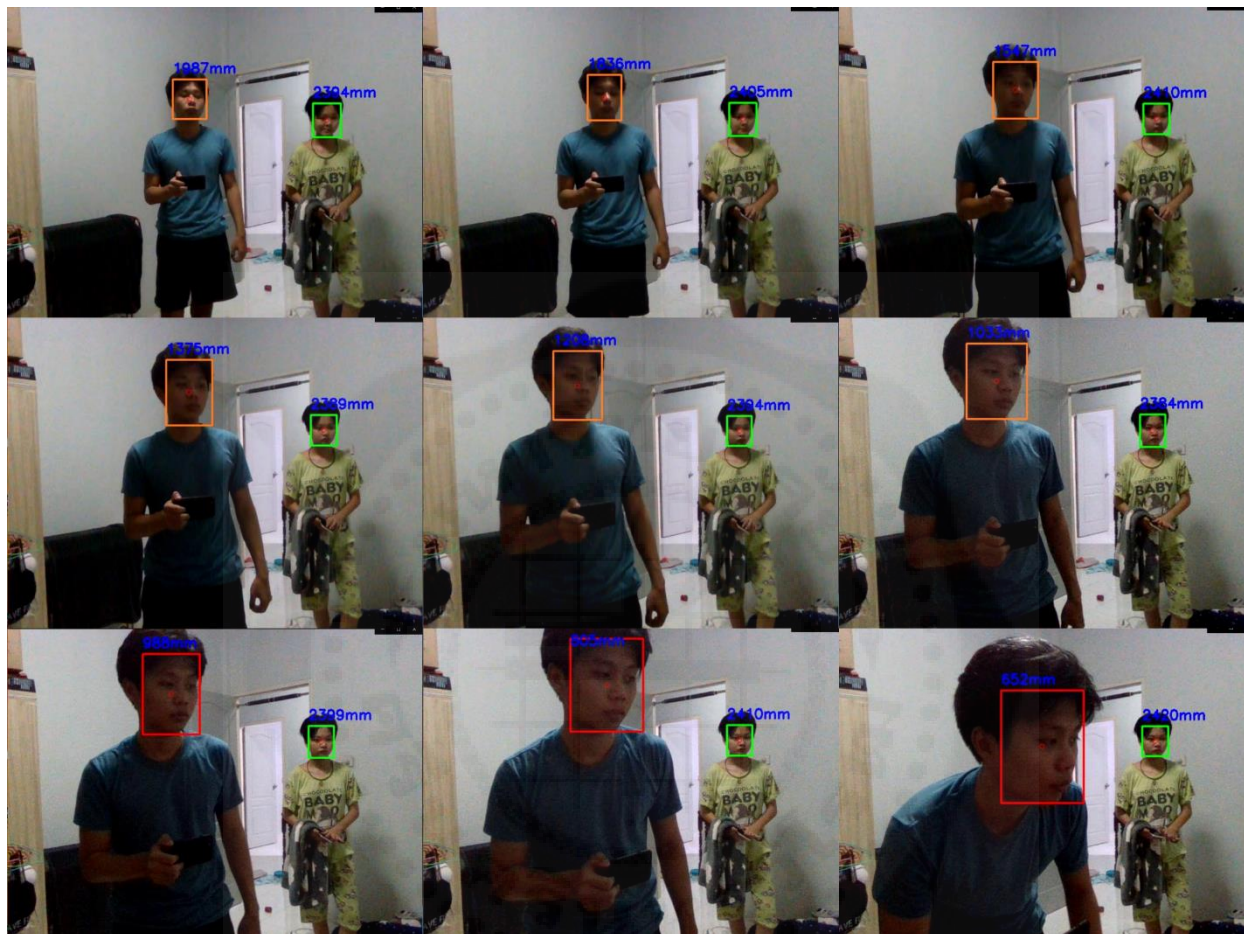
ภาพประกอบที่ 9 อบรมหลักสูตร

การดำเนินโครงการในครั้งนี้ผู้จัดทำได้เสนอผลการดำเนินโครงการ โดยแสดงผลดังต่อไปนี้



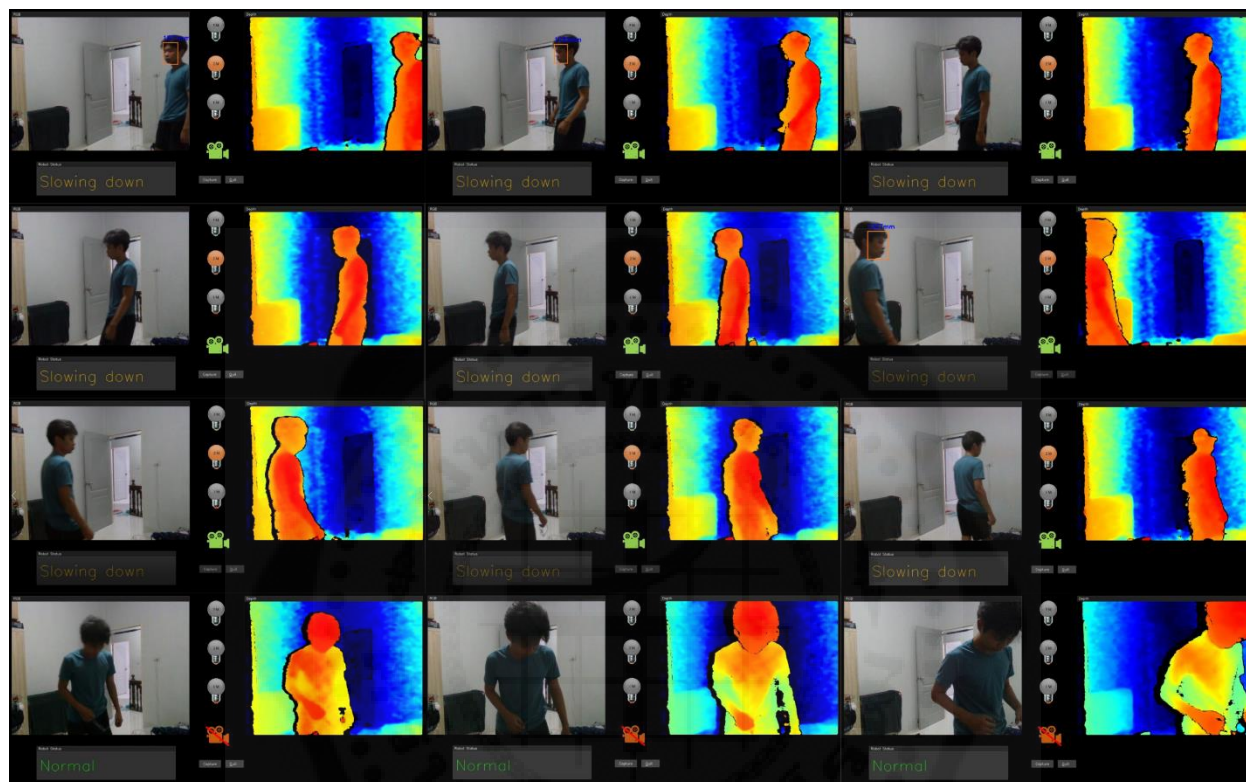
ภาพประกอบที่ 10 ผลการดำเนินงาน 1

ทำการทดสอบโดยการเดินเข้าหาพื้นที่แบบปกติ ซึ่งมีบุคคลให้กล้องตรวจจับใบหน้ามากกว่า 1 ใบหน้าในหนึ่งเฟรม เพื่อทดสอบว่าการควบคุมระบบด้วยกล้องสามารถทำงานได้ปกติตามระยะที่กำหนดไว้ และทดสอบว่ากล้องจะตรวจจับใบหน้าบุคคลที่ใกล้กว่าและนำค่าระยะห่างของบุคคลที่ใกล้กว่านั้นไปสั่งการควบคุมหุ่นยนต์



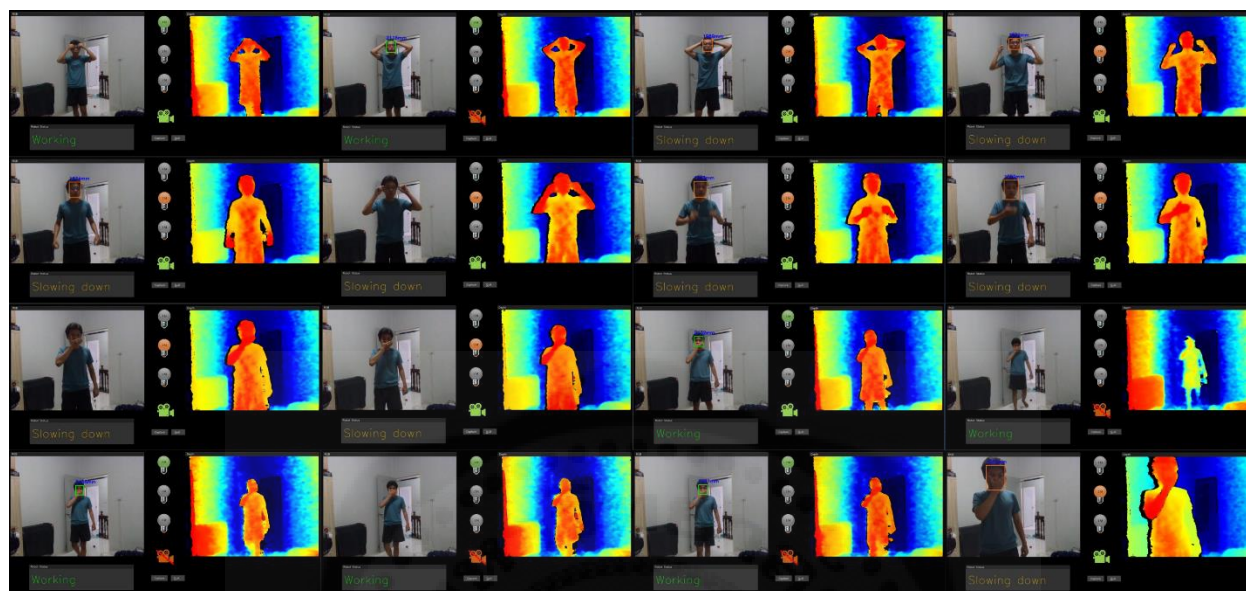
ภาพประกอบที่ 11 ผลการดำเนินงาน 2

เป็นวิดีโอสถานการณ์เดียวกันกับ ภาพประกอบที่10 แต่เป็นวิดีโอที่ได้จากการบันทึกของโปรแกรมควบคุมหุ่นยนต์ด้วยกล้องที่นิสิตพัฒนาไว้ ซึ่งจะจัดเก็บวิดีโอนี้ไว้ในโฟลเดอร์ Video ซึ่งชื่อของวิดีโอ นั้น ๆ จะเป็นเวลาที่ได้มีการบันทึกวิดีโอ



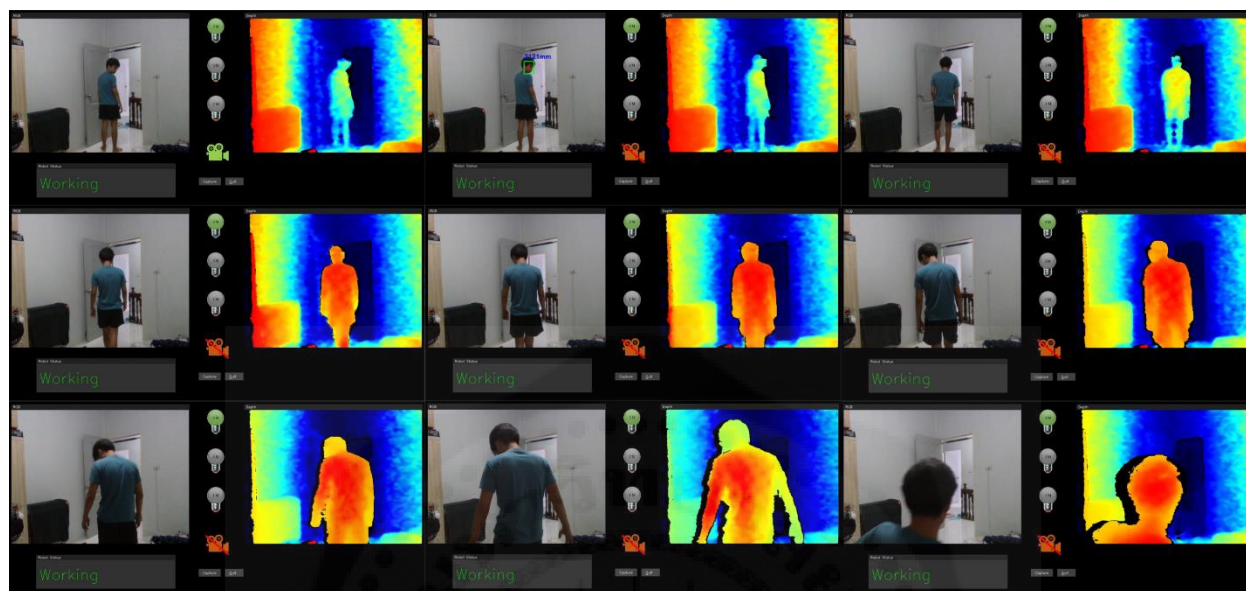
ภาพประกอบที่ 12 ผลการดำเนินงาน 3

ทดสอบโดยการเดินผ่านจากขวาไปซ้ายและจากซ้ายไปขวา โดยให้กล้องหันเพียงด้านข้างของใบหน้า จึงเห็นว่ากล้องมีความแม่นยำน้อยลงจากปกติ แต่ยังคงทำงานได้ในระดับหนึ่ง และได้ทำการทดสอบโดยการก้มหัวให้กล้องไม่สามารถเห็นใบหน้าเต็ม ๆ ได้ จึงทำให้กล้องไม่สามารถตรวจจับใบหน้าได้เลย



ภาพประกอบที่ 13 ผลการดำเนินงาน 4

ทดสอบโดยการที่นิสิตสวมแว่นตาปกติ ที่มีการปิดบังใบหน้าเล็กน้อย ทำให้กล้องมีความแม่นยำในการตรวจจับใบหน้าลดลง แต่ยังคงทำงานได้อยู่ และได้ทำการทดสอบโดยการใช้มือปิดบังใบหน้าบางส่วน ความแม่นยำในการตรวจจับใบหน้าลดลง แต่ยังคงทำงานได้อยู่เช่นเดียวกัน



ภาพประกอบที่ 14 ผลการดำเนินงาน 5

ทดสอบโดยการเดินถอยหลังเข้าหากลิ้ง จ้างสถานการณ์ที่คนงานอาจจะทำงานอยู่และต้องเคลื่อนที่ โดยการถอยหลัง ทำให้อาจไม่ทราบว่ามิหุ่นยนต์กำลังทำงานอยู่ด้านหลัง ซึ่งทำให้กลิ้งไม่สามารถตรวจจับใบหน้าได้เลย ส่งผลให้หุ่นยนต์นั้นไม่ถูกสั่งการให้เปลี่ยนรูปแบบการทำงาน ยังคงทำงานปกติ อาจทำให้เกิดความเสียหายได้ในเวลาต่อมา

บทที่ 5

สรุปผล อภิปรายผล และข้อเสนอแนะ

5.1 สรุปผลการศึกษาค้นคว้า

จากการดำเนินโครงการ A New Security System of Mitsubishi Robot using Intelligent 3D Vision สามารถควบคุมและสั่งการได้ระดับดี มีความคลาดเคลื่อนของระยะเพียงเล็กน้อย ($\pm 10 - 20$ cm) การรับระยะจากกล้องเพื่อสั่งการหุ่นด้วยการจำลอง ทำงานได้ดี ตรงตามวัตถุประสงค์ โดยสามารถตรวจจับใบหน้าได้มากกว่า 1 ใบหน้าและใช้ระยะของใบหน้าที่ใกล้กว่าเป็นค่าเพื่อสั่งการหุ่นให้ทำงานในระยะนั้น ๆ เพื่อความปลอดภัยในพื้นที่ทำงาน

5.2 อภิปรายผล

เนื่องจากตำแหน่งของกล้องอยู่สูงจากพื้น ทำให้มีความคลาดเคลื่อนระหว่างระยะจากกล้อง และการวัดระยะทางกายภาพของพื้นที่จริง

ระบบการตรวจจับใบหน้าที่ยังมีจุดอ่อนอยู่บ้าง เช่น การหันข้างเข้าหากล้อง จะทำให้ความแม่นยำในการจับใบหน้าลดลง อาจจับใบหน้าได้ และไม่ได้ในบางครั้ง หรือหากมีสิ่งของปิดบังใบหน้า เช่น แว่นตา หน้ากากอนามัย หมวกปิดหน้า หรือใช้มือบังหน้าไว้บางส่วนก็ทำให้การจับใบหน้าที่มีความแม่นยำลดลงเช่นกัน

5.3 ข้อเสนอแนะ

มีข้อจำกัดในเรื่องการตรวจจับใบหน้า เช่น หากพื้นที่ทำงานจริง มีพนักงานเดินถอยหลังเข้าสู่พื้นที่ทำงานของหุ่น จะทำให้กล้องไม่สามารถตรวจจับใบหน้า และหารระยะเพื่อสั่งการหุ่นให้ทำงานตามระบบได้ อาจทำให้เกิดอันตรายต่อพนักงานผู้นั้น

จากการประชุมกับทาง บ.มิตซู ได้รับข้อมูลว่า ด้วยตามมาตรฐาน safety ของหุ่นยนต์ ไม่สามารถสั่งการให้หุ่นยนต์เปลี่ยนแปลงการทำงานได้ ผ่านทางการเขียนโปรแกรม การสั่งการให้หุ่นยนต์เปลี่ยนแปลงการทำงาน ต้องทำการสั่งจากโปรแกรมผ่านกล่องที่แสดงในรูปภาพนี้

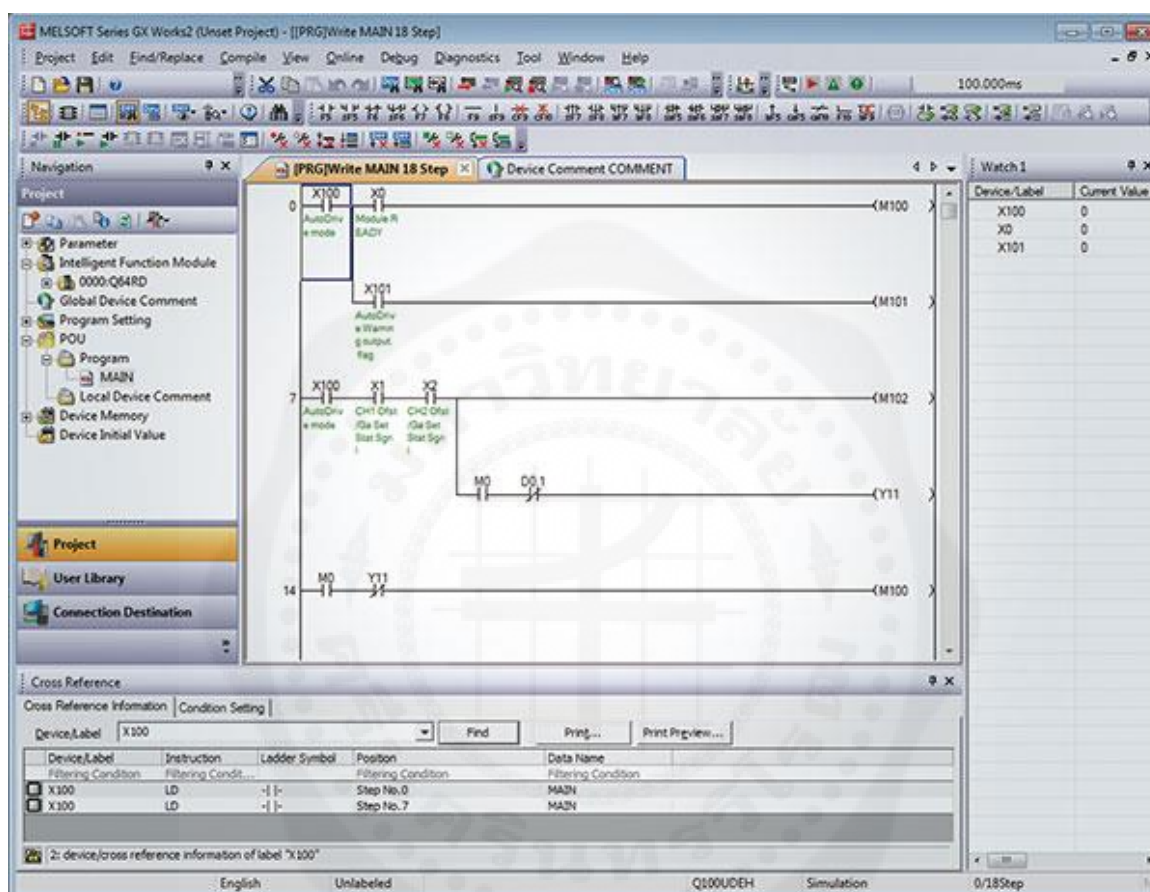


ภาพประกอบที่ 15 กล่องควบคุมการทำงาน

กล่องควบคุมนี้ จะส่งคำสั่งผ่านสายคำสั่ง ที่เชื่อมต่อกับหุ่นยนต์โดยตรง เพื่อเปลี่ยนแปลงการทำงานต่อไป ดังนั้น การพัฒนาระบบในอนาคตต้องทำตามมาตรฐานของหุ่นยนต์

GX work 3

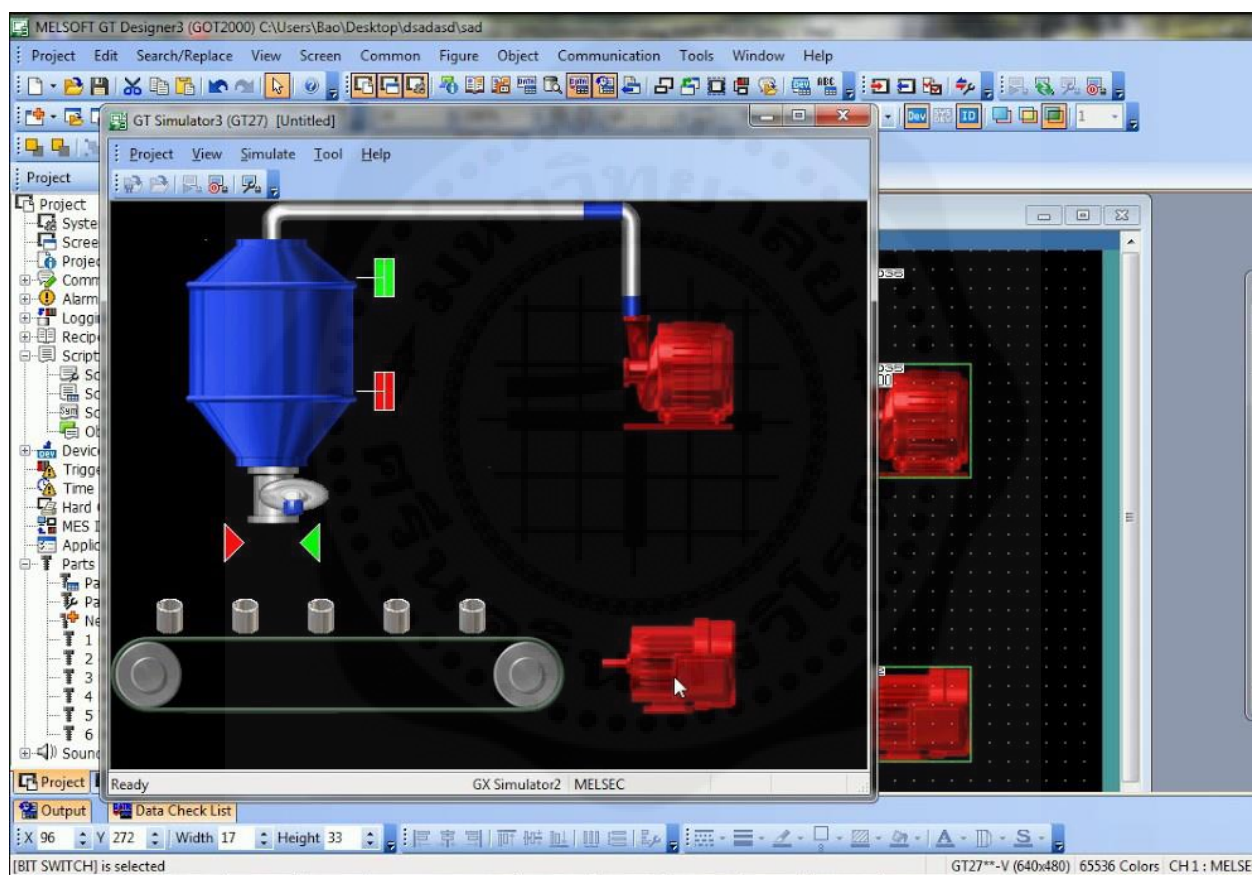
เป็นโปรแกรมสำหรับเขียนคำสั่ง สั่ง PLC ซึ่งเป็นโปรแกรมรุ่นใหม่ ที่ใช้กับ PLC series ปัจจุบัน เป็นภาษา Ladder มี license หน้าตาภายในคล้ายกับวงจรไฟฟ้า ใช้ความรู้ทางวงจรไฟฟ้าในการทำงานได้เช่นกัน



ภาพประกอบที่ 16 interface ภายในโปรแกรม GX Work 3

GT Designer3

เป็นโปรแกรมสำหรับการสร้าง HMI หรือ UI เพื่อเป็นสื่อกลางสำหรับผู้ใช้งานกับหุ่นยนต์ ซึ่งจ่อจะเป็นรุ่น GOT2000 ที่ใช้กับโปรแกรมนี้ เป็นรุ่นปัจจุบัน เชื่อมต่อด้วยสาย Ethernet จากจ่อ GOT2000 สู่ PLC ของหุ่น สามารถออกแบบ I/O เป็นการจำลองการทำงานได้ มีระบบสัมผัส การออกแบบใช้เป็นกราฟฟิกตามไลบรารีที่โปรแกรมมีมาให้ และทำงานร่วมกับโค้ด Ladder ของ GX work3 ซึ่งต้องเชื่อมกันระหว่างโปรแกรม เพื่อการสั่งการหุ่นด้วยจ่อ GOT2000



ภาพประกอบที่ 17 interface ภายในโปรแกรม GT Designer 3

บรรณานุกรม

ยุทธนา สุทธสุภา. (2012). การระบุตำแหน่งพร้อมกับการสร้างแผนที่ขนาดใหญ่ด้วยกล้องวิดีโอมุมกว้าง.

สืบค้น 10 ตุลาคม 2563, จาก <https://isl2.cp.eng.chula.ac.th/research-file/aun/doctor/aun-proposal.pdf>

Doronhi. (2018). **ROS Wrapper for Intel RealSense Devices**. Retrieved 10 October 2020,

From <https://github.com/IntelRealSense/realsense-ros>

Mirjana Maksimovic. (2014). **Raspberry Pi as Internet of Things hardware: Performances**

and Constraints. Retrieved 10 October 2020,

From <https://www.scribd.com/document/259010266/Raspberry-Pi-as-Internet-of-Things-Hardware-Performances-and-Constraints>

Makerbro. (2020). **Raspberry Pi 4 and Intel RealSense D435**. Retrieved 11 October 2020,

From https://github.com/acrobotic/Ai_Demos_RPi/wiki/Raspberry-Pi-4-and-Intel-RealSense-D435

Sergey Dorodnicov. (2019). **SLAM with D435i**. Retrieved 10 October 2020,

From <https://github.com/IntelRealSense/realsense-ros/wiki/SLAM-with-D435i>

TullyFoote. (2020). **Ubuntu install of ROS Melodic**. Retrieved 5 October 2020,

From <http://wiki.ros.org/melodic/Installation/Ubuntu>

Vladimir Tadic. (2019). **Application of Intel RealSense Cameras for Depth Image**

Generation in Robotics. WSEAS TRANSACTIONS on COMPUTERS, 2019(18), 107-112.

<https://www.wseas.org/multimedia/journals/computers/2019/a285105-086.php>

Wolfgang Hess. (2016). **Real-time loop closure in 2D LIDAR SLAM**.

2016 IEEE International Conference on Robotics and Automation (ICRA),

2016(June, 2016), 1-8. DOI: 10.1109/ICRA.2016.748725

Arun Ponnusamy. (2018). **CNN based face detector from dlib**. Retrieved 1 April 2021,

From [https://towardsdatascience.com/cnn-based-face-detector-from-dlib-](https://towardsdatascience.com/cnn-based-face-detector-from-dlib-c3696195e01c)

[c3696195e01c](https://towardsdatascience.com/cnn-based-face-detector-from-dlib-c3696195e01c)

ภาคผนวก

```

1  import numpy as np
2  import cv2
3  import cvui
4  from realsense_depth import *
5  import time
6  import cvlib as cv
7  import os
8
9  WINDOW_NAME = 'App'
10 dc = DepthCamera()
11 def main():
12
13     def capture(img):
14         timestr = time.strftime("%Y%m%d-%H%M%S")
15         img_name = timestr + ".png"
16         path = 'Images'
17         cv2.imwrite(os.path.join(path, img_name), img)
18         print("Capture!")
19
20     def resize(img, size):
21         result = cv2.resize(img, (int(img.shape[1] * size), int(img.shape[0] * size)))
22         return result

```

```

23
24     def record_vid():
25         global video_writer
26         timestr = time.strftime("%Y%m%d-%H%M%S")
27         filename = timestr + ".avi"
28         path = 'Videos'
29         fw = 640
30         fh = 480
31         # video recorder
32         fourcc = cv2.VideoWriter_fourcc('M', 'J', 'P', 'G') # cv2.VideoWriter_fourcc() does not work
33         video_writer = cv2.VideoWriter(os.path.join(path, filename), fourcc, 25, (fw, fh))
34
35     def release_vid():
36         video_writer.release()
37
38     win = np.zeros((700, 1500, 3), np.uint8)
39
40     textstatus = "Normal"
41     colorstatus = 0x00ff00
42     # รูปกำลังอัด
43     img_record = cv2.imread('record.png', cv2.IMREAD_COLOR)
44     img_unrecord = cv2.imread('unrecord.png', cv2.IMREAD_COLOR)
45     img_unrecord2 = resize(img_unrecord, 0.291)

```

```

46 # หลอดไฟ
47 # 3m เขียว
48 ngreen = cv2.imread('3m.png', cv2.IMREAD_COLOR)
49 ngreen2 = resize(ngreen, 0.25)
50 green = cv2.imread('3mgreen.png', cv2.IMREAD_COLOR)
51 green2 = resize(green, 0.25)
52 # 2m ส้ม
53 norange = cv2.imread('2m.png', cv2.IMREAD_COLOR)
54 norange2 = resize(norange, 0.25)
55 orange = cv2.imread('2morange.png', cv2.IMREAD_COLOR)
56 orange2 = resize(orange, 0.25)
57 # 1m แดง
58 nred = cv2.imread('1m.png', cv2.IMREAD_COLOR)
59 nred2 = resize(nred, 0.25)
60 red = cv2.imread('1mred.png', cv2.IMREAD_COLOR)
61 red2 = resize(red, 0.25)
62
63 record = True
64 recording = False
65 min = 9999
66 start = 0
67 #ตัวแปรสำหรับแจ้งเตือน
68 clean = False
69 warning = False
70 danger = False
71 cvui.init(WINDOW_NAME)
72

```

```

73 while (True):
74
75     ret, depth_frame, color_frame, colorized_depth = dc.get_frame()
76
77     # apply face detection
78     face, confidence = cv.detect_face(color_frame)
79     distance = []
80
81     win[:] = (49, 52, 49)
82
83     #กระบวนการทำงาน
84     for idx, f in enumerate(face):
85         (startX, startY) = f[0], f[1]
86         (endX, endY) = f[2], f[3]
87         point = ((f[0] + f[2]) // 2, (f[1] + f[3]) // 2)
88         distance.insert(idx, depth_frame[point[1], point[0]])
89         cv2.circle(color_frame, point, 4, (0, 0, 255))
90
91         if distance[idx] > 2000:
92             color = (0, 255, 0)
93         elif distance[idx] <= 2000 and distance[idx] > 1000:
94             color = (31, 120, 255)
95         elif distance[idx] <= 1000:
96             color = (0, 0, 255)
97         # draw rectangle over face
98         cv2.rectangle(color_frame, (startX, startY), (endX, endY), color, 2)
99
100        text = "{}mm".format(distance[idx])

```

```

101
102     Y = startY - 10 if startY - 10 > 10 else startY + 10
103
104     # write confidence percentage on top of face rectangle
105     cv2.putText(color_frame, text, (startX, Y), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 0, 0), 2)
106
107     #window Show Camera
108     cvui.beginRow(win, 10, 10, 500, 1000, 200)
109     cvui.window(640, 480, 'RGB')
110     cvui.image(win, 10, 30, color_frame)
111     cvui.window(640, 480, 'Depth')
112     cvui.image(win, 850, 30, colorized_depth)
113     cvui.endRow()
114

```

```

115     #กระบวนการหา min
116     onetime = True
117     for dis in distance:
118         if onetime == True:
119             min = dis
120             onetime = False
121         if (dis <= 2000):
122             if (record):
123                 print("กำลังอัด")
124                 record = False
125                 recording = True
126                 start = time.time()
127                 record_vid()
128             elif (record == False):
129                 start = time.time()
130         if(min > dis):
131             min = dis
132
133

```

```

134 # เอา minimum เพื่อไปดูว่าควรแจ้งเตือนระดับไหน
135 if (min <= 1000):
136     clean = False
137     warning = False
138     danger = True
139     textstatus = "STOP"
140     colorstatus = 0xff0000
141 elif (min <= 2000 and min > 1000):
142     clean = False
143     warning = True
144     danger = False
145     textstatus = "Slow Down"
146     colorstatus = 0xffc000
147 elif (min > 2000 and min < 9000):
148     clean = True
149     warning = False
150     danger = False
151     textstatus = "Normal"
152     colorstatus = 0x00ff00
153
154 # สถานะเครื่องจักร
155 cvui.beginRow(win, 100, 550, 100, 100)
156 cvui.window(400, 120, 'Robot Status ')
157 cvui.text(win, 110, 600, textstatus, 1.8, colorstatus)
158 cvui.endRow()
159

```

```

160 #กระบวนการอัด
161 if (time.time() - start >= 5 and recording == True):
162     release_vid()
163     recording = False
164     record = True
165     clean = False
166     warning = False
167     danger = False
168     textstatus = "Normal"
169     colorstatus = 0x00ff00
170     min = 9999
171     print("Record Stop")
172
173 if recording == True:
174     video_writer.write(color_frame)
175
176 #ไม่ record
177 if(recording == False):
178     cvui.image(win, 705, 460, img_unrecord2)
179 else:
180     cvui.image(win, 705, 460, img_record)

```

```

181     #Green 20 550
182     #Orange 150 550
183     #Red 280 550
184     # 1 685 10
185     # 2 685 150
186     # 3 685 290
187     if clean == True:
188         cvui.image(win, 685, 10, green2)
189         cvui.image(win, 685, 150, norange2)
190         cvui.image(win, 685, 290, nred2)
191     elif warning == True:
192         cvui.image(win, 685, 10, ngreen2)
193         cvui.image(win, 685, 150, orange2)
194         cvui.image(win, 685, 290, nred2)
195     elif danger == True:
196         cvui.image(win, 685, 10, ngreen2)
197         cvui.image(win, 685, 150, norange2)
198         cvui.image(win, 685, 290, red2)
199     else:
200         cvui.image(win, 685, 10, ngreen2)
201         cvui.image(win, 685, 150, norange2)
202         cvui.image(win, 685, 290, nred2)
203
204     if (cvui.button(win, 685, 600, 'Capture')):
205         capture(color_frame)
206     if (cvui.button(win, 780, 600, '&Quit')):
207         break
208

```

```

209     cvui.update()
210     cv2.imshow(WINDOW_NAME, win)
211     key = cv2.waitKey(1)
212
213     if (cv2.waitKey(20) == 27 or cv2.getWindowProperty(WINDOW_NAME, cv2.WND_PROP_ASPECT_RATIO) < 0):
214         break
215
216     dc.release()
217     cv2.destroyAllWindows()
218     if __name__ == '__main__':
219         main()

```

```

1  import pyrealsense2 as rs
2  import numpy as np
3
4  class DepthCamera:
5      def __init__(self):
6          # Configure depth and color streams
7          self.pipeline = rs.pipeline()
8          config = rs.config()
9
10         # Get device product line for setting a supporting resolution
11         pipeline_wrapper = rs.pipeline_wrapper(self.pipeline)
12         pipeline_profile = config.resolve(pipeline_wrapper)
13         device = pipeline_profile.get_device()
14         device_product_line = str(device.get_info(rs.camera_info.product_line))
15
16         config.enable_stream(rs.stream.depth, 640, 480, rs.format.z16, 30)
17         config.enable_stream(rs.stream.color, 640, 480, rs.format.bgr8, 30)
18         # Start streaming
19         self.pipeline.start(config)

```

```

20
21  def get_frame(self):
22      frames = self.pipeline.wait_for_frames()
23      depth_frame = frames.get_depth_frame()
24      color_frame = frames.get_color_frame()
25
26      depth_image = np.asanyarray(depth_frame.get_data())
27      colorizer = rs.colorizer()
28      colorized_depth = np.asanyarray(colorizer.colorize(depth_frame).get_data())
29      color_image = np.asanyarray(color_frame.get_data())
30      if not depth_frame or not color_frame:
31          return False, None, None
32      return True, depth_image, color_image, colorized_depth
33
34  def release(self):
35      self.pipeline.stop()

```