



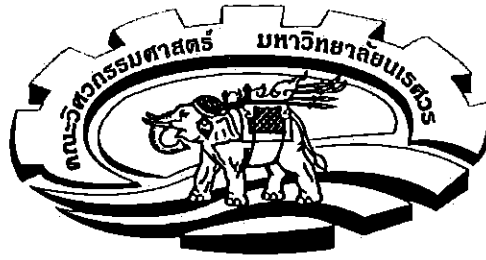
การควบคุมอากาศยานปีกหมุนแบบสี่ใบพัดด้วยท่าทางการเคลื่อนไหวของร่างกาย
GESTURE-BASED QUADCOPTER CONTROL



นายศราวุธ เทียนคำ รหัส 54360292
นายสิทธิศักดิ์ ทีโน รหัส 54360322

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครสวรรค์
ปีการศึกษา 2557

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ..... 20.0.0..... 2558.....
เลขทะเบียน..... 1689.21.4.....
เลขเรียกหนังสือ..... 255.....
มหาวิทยาลัยนครสวรรค์ ๗/๒๙ ๑ 2557



ใบรับรองปริญญาโท

หัวข้อโครงการ การควบคุมอากาศยานปีกหมุนแบบสี่ใบพัดด้วยท่าทางการเคลื่อนไหวของร่างกาย

ผู้ดำเนินโครงการ นายศรราช เทียนคำ รหัส 54360292
 นายสิทธิศักดิ์ ทีโน รหัส 54360322

อาจารย์ที่ปรึกษา อาจารย์รัฐภูมิ วรานุสาสน์

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2557

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยราชภัฏสุรินทร์ อนุมัติให้ปริญญาโทฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

.....ที่ปรึกษาโครงการ
(อาจารย์รัฐภูมิ วรานุสาสน์)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ริยะมงคล)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.พนัส นัถฤทธิ)

หัวข้อโครงการ	การควบคุมอากาศยานปีกหมุนแบบสี่ใบพัดด้วยท่าทางการเคลื่อนไหวของร่างกาย		
ผู้ดำเนินโครงการ	นายศราวุธ	เทียนคำ	รหัส 54360292
	นายสิทธิศักดิ์	ทีโน	รหัส 54360322
อาจารย์ที่ปรึกษา	อาจารย์รัฐภูมิ วรานุสาสน์		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2557		

บทคัดย่อ

โครงการนี้ได้นำเทคโนโลยีการเชื่อมต่อกับผู้ใช้งานแบบเป็นธรรมชาติมาประยุกต์ใช้เพื่อพัฒนาเป็นต้นแบบการควบคุมอากาศยานปีกหมุนแบบสี่ใบพัด ในหัวข้อ "การควบคุมอากาศยานปีกหมุนแบบสี่ใบพัดด้วยท่าทางการเคลื่อนไหวของร่างกาย" โดยกล้องที่ติดอยู่กับอากาศยานปีกหมุนจะทำหน้าที่รับภาพท่าทางการเคลื่อนไหวของร่างกาย และส่งภาพให้กับคอมพิวเตอร์ จากนั้นคอมพิวเตอร์จะทำการประมวลผลภาพโดยใช้เทคนิคฮาร์แคสแคส เพื่อตรวจจับร่างกายท่อนบน จากนั้นทำการกำจัดส่วนเกินในภาพที่เราไม่ต้องการออก เพราะการประมวลผลภาพต้องการเฉพาะรูปใบหน้า และรูปมือทั้ง 2 ข้างเท่านั้น จากนั้นทำการแปลงภาพเป็นภาพไบนารี แล้วหาพื้นที่ที่ติดกันจากกระบวนการข้างต้น เพื่อจะใช้ในการคำนวณหาตำแหน่งของใบหน้าและมือ จากนั้นก็จะกำหนดคำสั่งการเคลื่อนไหวจากตำแหน่งของใบหน้าและมือ แล้วส่งคำสั่งควบคุมไปยังอากาศยานปีกหมุนเพื่อให้เคลื่อนที่ตามคำสั่งต่างๆ ผ่านระบบเครือข่ายไร้สาย โดยระบบสามารถควบคุมอากาศยานปีกหมุนได้ 7 ลักษณะ คือ บินอยู่กับที่ บินขึ้น บินลง บินซ้าย บินขวา บินไปด้านหน้าและบินไปด้านหลัง ซึ่งผลการทดลองแสดงให้เห็นว่า เราสามารถนำหลักการประมวลผลภาพ มาประยุกต์ใช้ในการพัฒนาต้นแบบการควบคุมอากาศยานปีกหมุนแบบสี่ใบพัด ซึ่งทำให้เกิดความเป็นธรรมชาติในการควบคุมมากขึ้น

Project Title	Gesture-Based Quadcopter Control		
Name	Mr.Sarawut	Thiankham	ID. 54360292
	Mr.Sitthisak	Thino	ID. 54360322
Project Advisor	Mr.Rattapoom Waranusast		
Major	Computer Engineering.		
Department	Electrical and Computer Engineering.		
Academic year	2014		

Abstract

The aim of this project is to develop a prototype for controlling a Quadcopter using Natural User Interface (NUI) technology focusing on human gestures.

Firstly, a Quadcopter is connected to a computer via an ad-hoc wireless network. Secondly, the camera in front of the Quadcopter takes a video stream of human movements and feeds to the computer. Thirdly, The upper body of a person in the image is detected by Haar cascade technique. Next, unwanted noises are detected by Morphological operations. The image is then transformed into a binary image using Thresholding operation. Connected regions are then detected and used to determine the body pose. Lastly, the commands are determined from the body pose and sent to control the Quadcopter. And The computer generates seven commands to control the Quadcopter movement, i.e. maintain position, fly up, fly down, fly left, fly right, fly forward, and fly backward.

This study found that that the combination of the NUI technology and image processing technology can be used to develop an gesture-based user interface for a Quadcopter control.

กิตติกรรมประกาศ

โครงการวิศวกรรมฉบับนี้ สำเร็จลุล่วงได้ด้วยดีเนื่องจากความอนุเคราะห์ของอาจารย์ที่ปรึกษาโครงการคือ อาจารย์รัฐภูมิ วรรณสาสน์ ซึ่งสละเวลาในการช่วยให้คำปรึกษา ให้คำแนะนำได้ งามความก้าวหน้าของวิทยานิพนธ์มาโดยตลอด รวมทั้งยังแนะนำขั้นตอนต่างๆ ในการทำการทดลอง เพื่อให้การทดลองเป็นไปอย่างรอบคอบและถูกต้อง

ขอขอบพระคุณ คณะกรรมการโครงการทั้ง 2 ท่าน คือ ผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ริยะมงคล และผู้ช่วยศาสตราจารย์ ดร.พนัส นัถฤทธิ์ ที่ให้คำแนะนำในการแก้ไขและปรับปรุงโครงการ ให้มีประสิทธิภาพมากขึ้น

ขอขอบพระคุณ คณะอาจารย์ทุกท่าน บิดา มารดา และเพื่อนวิศวกรรมคอมพิวเตอร์ที่ได้ให้การช่วยเหลือและสนับสนุนในการทดลอง และเป็นกำลังใจแก่คณะผู้จัดทำ ขอขอบพระคุณทุกๆ ท่าน ที่ได้เฝ้าเฝ้าจนถึง ที่มีส่วนร่วมช่วยให้โครงการนี้สำเร็จลุล่วงไปด้วยดี

คณะผู้ดำเนินโครงการวิศวกรรม

ศราวุธ เทียนคำ

สิทธิศักดิ์ ทีโน



สารบัญ

	หน้า
ใบรับรองปริญญาโท.....	ก
บทคัดย่อภาษาไทย.....	ข
บทคัดย่อภาษาอังกฤษ.....	ค
กิตติกรรมประกาศ.....	ง
สารบัญ	จ
สารบัญตาราง.....	ช
สารบัญรูป.....	ฉ
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของโครงการ.....	2
1.4 ขั้นตอนการดำเนินงานของโครงการ.....	2
1.5 แผนการดำเนินงานของโครงการ.....	3
1.6 ผลที่คาดว่าจะได้รับ.....	3
1.7 งบประมาณของโครงการ.....	4
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง.....	5
2.1 ความรู้เบื้องต้นเกี่ยวกับการเชื่อมต่อกับผู้ใช้อย่างเป็นธรรมชาติ.....	5
2.2 ความรู้เบื้องต้นเกี่ยวกับไลบรารีของโอเพนซีวี.....	5
2.3 ความรู้เบื้องต้นเกี่ยวกับการประมวลผลภาพ.....	6
2.4 เกี่ยวกับ QuadCopter.....	6
2.4.1 ข้อมูลทั่วไปของ Quadcopter รุ่น AR.Drone 2.0.....	6
2.4.2 หลักการการบินของ Quadcopter.....	7
2.5 การประมวลผลภาพ.....	9
2.5.1 ภาพสี.....	9
2.5.2 ระบบสี.....	9
2.5.3 การแปลงภาพสีให้เป็นภาพในระดับเทา.....	10
2.5.4 การตรวจจับใบหน้าด้วยโอเพนซีวี.....	11
2.5.5 การประมวลผลภาพกับรูปร่างและโครงร่างของภาพ.....	13
2.5.6 การแปลงภาพเป็นขาวดำ.....	14
2.5.7 คอนทัวร์.....	14
2.5.8 การหาขอบภาพ.....	15
2.5.9 การแบ่งส่วนภาพและแยกภาพ.....	15

สารบัญ (ต่อ)

	หน้า
2.5.10 การติดตามวัตถุ.....	15
2.5.11 ประมาณท่าทางของมนุษย์.....	15
2.5.12 ผสานวัตถุเสมือน.....	16
บทที่ 3 วิธีการดำเนินโครงการ.....	19
3.1 ขั้นตอนการรับภาพจากอากาศยานปีกหมุน.....	20
3.1.1 การเชื่อมต่อคอมพิวเตอร์กับอากาศยานปีกหมุน.....	20
3.1.2 การรับภาพจากกล้องของอากาศยานปีกหมุน.....	20
3.2 ขั้นตอนการประมวลผลภาพขั้นต้น.....	21
3.2.1 การตรวจจ็บบร่างกายท่อนบน.....	21
3.2.2 การแปลงภาพเป็นภาพสี HSV และทำการเลือกระดับสีที่ต้องการ.....	22
3.2.3 การปรับปรุงภาพไบนารี.....	23
3.2.4 การหาจำนวนวัตถุในภาพ.....	24
3.3 การวิเคราะห์ภาพเพื่อให้ได้คำสั่งควบคุมอากาศยานปีกหมุน.....	24
3.3.1 วาดกรอบพื้นที่.....	24
3.3.2 การหาตำแหน่งของมือ.....	25
3.4 ขั้นตอนการส่งคำสั่งควบคุมให้กับอากาศยานปีกหมุน.....	27
บทที่ 4 ผลการทดลอง.....	29
4.1 ผลการทดลองการเชื่อมต่อระหว่างคอมพิวเตอร์กับอากาศยานปีกหมุน.....	29
4.2 ผลการทดลองการควบคุมอากาศยานปีกหมุนด้วยการเคลื่อนไหวของร่างกาย... 4.2.1 วิธีการควบคุมอากาศยานปีกหมุนด้วยการเคลื่อนไหวของร่างกาย.....	31 32
4.2.2 ผลของค่าสถานะ เมื่อเริ่มต้นรันโปรแกรม.....	39
4.3 ปัจจัยที่มีผลต่อการควบคุมอากาศยานปีกหมุนด้วยการเคลื่อนไหวของร่างกาย... 4.3.1 การแต่งกาย.....	41 41
4.3.2 สภาพลม.....	42
4.3.3 สภาพแสง.....	42
4.3.4 พื้นหลัง.....	42
4.3.5 ระยะห่างระหว่างผู้ควบคุมกับอากาศยานปีกหมุน.....	43
4.4 ผลการทดลองเมื่อควบคุมอากาศยานปีกหมุนด้วยระยะห่างที่แตกต่างกัน.....	43
4.5 สรุปผลการทดลองในส่วนของปัจจัยที่มีผลต่อการควบคุมอากาศยานปีกหมุน...	44
บทที่ 5 สรุปผลการดำเนินงานโครงการ.....	45
5.1 สรุปผลการทดลองของโครงการ.....	45
5.2 ปัญหาและอุปสรรคของโครงการ.....	46

สารบัญ (ต่อ)

	หน้า
5.3 ข้อเสนอแนะและแนวทางการพัฒนา.....	47
เอกสารอ้างอิง.....	48
ภาคผนวก ก. คู่มือการติดตั้งโปรแกรมประยุกต์.....	50
ภาคผนวก ข. อากาศยานปีกหมุนที่ใช้ในการทดสอบ.....	60
ภาคผนวก ค. คู่มือการใช้งานระบบ.....	61
ประวัติผู้ดำเนินโครงการ.....	63



สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินงาน.....	3
4.1 วิธีการควบคุมอากาศยานปีกหมุนด้วยคีย์บอร์ด.....	30
4.2 แสดงผลของการควบคุมอากาศยานปีกหมุนด้วยระยะห่างที่แตกต่างกัน.....	43
5.1 ปัญหาและอุปสรรคและแนวทางการแก้ไข.....	45



สารบัญรูป

รูปที่	หน้า
2.1 สัญลักษณ์ OpenCV.....	5
2.2 หลักการเบื้องต้นของการประมวลผลภาพ.....	6
2.3 ภาพ AR.Drone 2.0.....	6
2.4 แสดงการบินลอยตัวอยู่กับที่ (Hovering).....	7
2.5 แสดงการบินขึ้น-ลง (Throttle).....	7
2.6 แสดงการบินเอียงตัวซ้าย - ขวา (Roll).....	7
2.7 แสดงการบินเอียงหน้า - หลัง (Pitch).....	8
2.8 แสดงการบินแบบหมุนตัว (Yaw).....	8
2.9 ภาพสี และค่าในแต่ละพิกเซล.....	9
2.10 โครงสร้างระบบสี RGB.....	9
2.11 โครงสร้างระบบสี HSV.....	10
2.12 การแปลงภาพสี RGB ไปเป็นภาพระดับสีเทา.....	11
2.13 แสดงตัวอย่างลักษณะเด่นของ Haar ที่ถูกใช้ใน OpenCV.....	11
2.14 แสดงเทคนิคการรวมภาพ.....	12
2.15 แสดงสายโซ่ของตัวกรอง.....	12
2.16 แสดงตัวอย่างของตัวกรองสองตัวแรกใน Viola-Jones Cascade.....	13
2.17 เส้นคอนทัวร์ของวัตถุ.....	14
2.18 แสดงผลงานวิจัยเกี่ยวกับการประมาณท่าทางของมนุษย์.....	16
2.19 แสดงการรวมภาพคอมพิวเตอร์กราฟฟิคลงบนภาพวิดีโอคน.....	17
2.20 แสดงภาพแบบจำลองท่าทางของร่างกาย.....	17
2.21 แสดงการหาตำแหน่งของมือคนโดยใช้ภาพไบนารี.....	18
2.22 แสดงภาพการระบุตำแหน่งอวัยวะของร่างกายส่วนบนแบบเวลาจริง (Real-time).....	18
3.1 Flow การทำงานประมวลผลคำสั่งควบคุมอากาศยานปีกหมุน.....	19
3.2 ภาพแสดงการรับภาพจากอากาศยานปีกหมุน.....	20
3.3 ภาพแสดงการตรวจจับร่างกายท่อนบน.....	21
3.4 ภาพที่ได้จากการแปลงเป็นภาพระบบสี HSV.....	22
3.5 ภาพที่ได้จากการเลือกช่วงของสีที่ต้องการ.....	23
3.6 ภาพที่ได้หลังจากปรับปรุงภาพไบนารี.....	23
3.7 ภาพที่ได้จากการติกรอบสีเหลี่ยมรอบวัตถุที่เจอและแสดงวัตถุที่เจอ.....	24
3.8 ภาพการติกรอบพื้นที่.....	25
3.9 ภาพประมวลผลคำสั่ง ให้ไปทางซ้าย.....	26
3.10 ภาพประมวลผลคำสั่ง ให้บินขึ้นข้างบน.....	26
3.11 ภาพการควบคุมท่าอากาศยานปีกหมุนสีไบพัตโดยใช้ร่างกาย.....	28

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.1 ผลของการรันโปรแกรมเมื่อไม่ได้เชื่อมต่อคอมพิวเตอร์กับอากาศยานปีกหมุน.....	29
4.2 ผลของการรันโปรแกรมเมื่อเชื่อมต่อคอมพิวเตอร์กับอากาศยานปีกหมุนแล้ว.....	30
4.3 แสดงภาพที่ได้จากกล้องของอากาศยานปีกหมุน.....	31
4.4 แสดงวิธีการควบคุมอากาศยานปีกหมุนให้บินอยู่กับที่.....	32
4.5 แสดงวิธีการควบคุมอากาศยานปีกหมุนให้บินขึ้น.....	33
4.6 แสดงวิธีการควบคุมอากาศยานปีกหมุนให้บินลง.....	34
4.7 แสดงวิธีการควบคุมอากาศยานปีกหมุนให้บินไปด้านซ้าย.....	35
4.8 แสดงวิธีการควบคุมอากาศยานปีกหมุนให้บินไปด้านซ้าย.....	36
4.9 แสดงวิธีการควบคุมอากาศยานปีกหมุนให้บินไปด้านหน้า.....	37
4.10 แสดงวิธีการควบคุมอากาศยานปีกหมุนให้บินไปด้านหน้า.....	38
4.11 แสดงภาพที่ได้จากกล้องของอากาศยานปีกหมุนเมื่อเริ่มรันโปรแกรม.....	39
4.12 แสดงค่าสถานะเมื่อกล้องจากอากาศยานปีกหมุนยังไม่พบร่างกายท่อนบน.....	39
4.13 แสดงรูปเมื่อกล้องจากอากาศยานปีกหมุนพบร่างกายท่อนบน.....	40
4.14 แสดงรูปภาพต้นฉบับที่ผ่านกระบวนการเทอร์โซลด์ และคอนทัวร์ จากภาพต้นฉบับ.....	40
4.15 แสดงค่าสถานะเมื่อกล้องจากอากาศยานปีกหมุนยังพบร่างกายท่อนบน.....	41
4.16 แสดงวิธีการแต่งตัวที่เหมาะสมต่อการควบคุมอากาศยานปีกหมุน.....	42
4.17 แสดงระยะห่างระหว่างอากาศยานปีกหมุนกับผู้ควบคุมที่เหมาะสม.....	43

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

ปัจจุบันอากาศยานปีกหมุนแบบสี่ใบพัด (Quadcopter) ได้รับความนิยมและถูกนำมาใช้ประโยชน์ด้านต่างๆ ในชีวิตประจำวันของมนุษย์เป็นอย่างมาก ตัวอย่างเช่น การเล่นเพื่อความสนุกสนาน การบินสำรวจพื้นที่ต่างๆ ที่เข้าถึงได้ยาก หรือการบินเพื่อการถ่ายภาพในมุมสูง เป็นต้น โดยวิธีการควบคุมอากาศยานปีกหมุนแบบสี่ใบพัดในปัจจุบันนั้นจะเป็นการควบคุมโดยใช้ สมาร์ทโฟน หรือรีโมทคอนโทรล ซึ่งขาดความเป็นธรรมชาติในการควบคุม เนื่องจากการควบคุมด้วยวิธีที่กล่าวมานั้น ผู้ควบคุมจำเป็นต้องมีการฝึกฝนและต้องมีทักษะในการควบคุม

เทคโนโลยีการเชื่อมต่อระหว่างมนุษย์กับคอมพิวเตอร์ในปัจจุบัน มีระบบการเชื่อมต่อกับผู้ใช้งานแบบเป็นธรรมชาติ (Natural user interface) เพื่อเพิ่มความสะดวกสบายของการใช้เทคโนโลยีต่างๆ ในชีวิตประจำวันของมนุษย์ เช่น ระบบหน้าจอสัมผัส (Touch screen) ระบบการควบคุมด้วยเสียง (Voice control) ที่มีใช้อย่างแพร่หลายในปัจจุบัน เป็นต้น ซึ่งเทคโนโลยีที่กล่าวมานี้ก็สามารถนำมาประยุกต์ใช้กับการควบคุมอากาศยานปีกหมุนแบบสี่ใบพัด เพื่อลดปัญหาหรือข้อจำกัดต่างๆ ที่ใช้ในการควบคุมอากาศยานปีกหมุนแบบสี่ใบพัดแบบเก่าได้

ดังนั้นผู้จัดทำโครงการจึงมีแนวคิดที่จะพัฒนาต้นแบบการควบคุมอากาศยานปีกหมุนแบบสี่ใบพัด ให้สามารถเชื่อมต่อกับผู้ใช้งานแบบเป็นธรรมชาติ ในหัวข้อ "การควบคุมอากาศยานปีกหมุนแบบสี่ใบพัดด้วยท่าทางการเคลื่อนไหวของร่างกาย (Gesture-base Quadcopter control)" เป็นโครงการที่มีจุดประสงค์เพื่อเป็นต้นแบบสำหรับระบบการควบคุมการเคลื่อนที่ของอากาศยานปีกหมุนแบบสี่ใบพัด โดยที่กล้องที่ติดอยู่กับอากาศยานปีกหมุนจะทำหน้าที่รับภาพท่าทางการเคลื่อนไหวของร่างกาย จากนั้นภาพที่ได้จะถูกส่งไปที่เครื่องคอมพิวเตอร์ส่วนบุคคล (PC) เพื่อทำการประมวลผลภาพ (Image processing) เพื่อรู้จำท่าทาง เมื่อประมวลผลภาพเสร็จ จะได้คำสั่งซึ่งเป็นคำสั่งที่ใช้ในการควบคุมการเคลื่อนที่ของอากาศยานปีกหมุน โดยจะแตกต่างกันไปตามลักษณะการเคลื่อนไหวของร่างกาย เนื่องจากโครงการนี้เป็นโครงการต้นแบบ ซึ่งในอนาคตอาจจะนำแนวคิดของโครงการนี้ไปต่อยอดสร้างเป็นอุปกรณ์ที่ใช้เพิ่มความสะดวกสบายในชีวิตประจำวันของมนุษย์ เช่น หุ่นยนต์เอนกประสงค์ภายในบ้านที่สามารถสั่งงานด้วยเสียงหรือท่าทางการเคลื่อนไหวของร่างกาย เป็นต้น

1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อศึกษาระบบการเชื่อมต่อระหว่างมนุษย์และคอมพิวเตอร์ส่วนบุคคลแบบเน้นการเชื่อมต่อกับผู้ใช้งานแบบเป็นธรรมชาติ

1.2.2 เพื่อศึกษาวิธีการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ส่วนบุคคลกับอากาศยานปีกหมุน

1.2.3 เพื่อศึกษาวิธีการตรวจจับท่าทางการเคลื่อนไหวของร่างกาย โดยใช้หลักการประมวลผลภาพ

1.2.4 เพื่อเพิ่มทางเลือกในการควบคุมอากาศยานปีกหมุนแบบสี่ใบพัดให้กับผู้ใช้

1.2.5 เพื่อสร้างโปรแกรมต้นแบบที่ใช้ในการควบคุมอากาศยานปีกหมุนแบบสี่ใบพัด โดยใช้ท่าทางการเคลื่อนไหวของร่างกาย

1.3 ขอบเขตของโครงการ

1.3.1 ใช้อากาศยานปีกหมุนแบบสี่ใบพัด รุ่น AR.Drone 2.0 ในการทดสอบระบบควบคุม

1.3.2 สร้างระบบต้นแบบที่ใช้ควบคุมการเคลื่อนที่ของอากาศยานปีกหมุนแบบสี่ใบพัด ด้วยท่าทางการเคลื่อนไหวของร่างกาย

1.3.3 ระบบสามารถควบคุมการเคลื่อนที่ของอากาศยานปีกหมุน ได้ 7 ลักษณะดังนี้

1.3.3.1 บินอยู่กับที่

1.3.3.2 บินขึ้น

1.3.3.3 บินลง

1.3.3.4 บินไปด้านซ้าย

1.3.3.5 บินไปด้านขวา

1.3.3.6 บินไปด้านหน้า

1.3.3.7 บินไปด้านหลัง

1.3.4 สภาพแวดล้อมที่ใช้ในการทดสอบ ต้องเป็นภายในอาคารที่มีแสงเพียงพอเท่านั้น

1.3.5 ในขั้นตอนของการทดสอบระบบการควบคุม ต้องใช้ผู้ทดสอบเพียงคนเดียวเท่านั้น

1.4 ขั้นตอนการดำเนินงานของโครงการ

1.4.1 ศึกษาวิธีการเชื่อมต่อระหว่างอากาศยานปีกหมุนกับคอมพิวเตอร์

1.4.2 ศึกษาและค้นคว้าข้อมูลเกี่ยวกับไลบรารีโอเพนซีวี (Opencv Library)

1.4.3 ศึกษารายละเอียดของการรู้จำภาพและข้อมูลต่างๆ ที่เกี่ยวกับการประมวลผลภาพ

1.4.4 ออกแบบโปรแกรมการควบคุมการเคลื่อนที่ของอากาศยานปีกหมุน โดยใช้หลักการประมวลผลภาพ

1.4.5 เขียนโปรแกรมการควบคุมอากาศยานปีกหมุนและพัฒนาโปรแกรม

1.4.6 ทดสอบการควบคุมอากาศยานปีกหมุนและแก้ไขข้อบกพร่อง

1.4.7 สรุปผลการทำโครงการและจัดทำรูปเล่มรายงาน

1.5 แผนการดำเนินงานของโครงการ

ตารางที่ 1.1 แผนการดำเนินงาน

กิจกรรม	ปี 2557					ปี 2558				
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1. ศึกษาวิธีการเชื่อมต่อระหว่างอากาศยานปีกหมุนกับความพิวเตอร์	■									
2. ศึกษาและค้นคว้าข้อมูลเกี่ยวกับไลบรารีโอเพนซีวี	■	■								
3. ศึกษารายละเอียดของความรู้จำภาพและข้อมูลต่างๆ ที่เกี่ยวกับการประมวลผลภาพ	■	■	■	■						
4. ออกแบบโปรแกรมการควบคุมการเคลื่อนที่ของอากาศยานปีกหมุน โดยใช้หลักการประมวลผลภาพ					■	■	■			
5. เขียนโปรแกรมการควบคุมอากาศยานปีกหมุนและพัฒนาโปรแกรม					■	■	■			
6. ทดสอบการควบคุมอากาศยานปีกหมุนและแก้ไขข้อบกพร่อง									■	■
7. สรุปผลการทำโครงการและจัดทำรูปเล่มรายงาน										■

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 ได้ค้นแบบโปรแกรมการควบคุมอากาศยานปีกหมุนแบบสี่ใบพัด ที่ใช้ทำทางการเคลื่อนไหวของร่างกายเป็นตัวควบคุม

1.6.2 ช่วยลดความยุ่งยากในการควบคุมอากาศยานปีกหมุน สำหรับผู้ที่กำลังเริ่มเล่น

1.6.3 ช่วยลดต้นทุนในการเล่นอากาศยานปีกหมุน เนื่องจากเราสามารถควบคุมได้ด้วยร่างกาย จึงไม่จำเป็นต้องมีสมาร์ตโฟนหรืออุปกรณ์ควบคุมอื่นๆ

1.6.4 สร้างความสนุกสนานในการเล่นอากาศยานปีกหมุน เนื่องจากเราสามารถควบคุมโดยทำทางการเคลื่อนไหวของร่างกาย จึงทำให้เราควบคุมได้อย่างเป็นธรรมชาติ

1.6.5 สามารถนำต้นแบบของระบบที่พัฒนา ไปประยุกต์ใช้งานต่อไปได้ในอนาคต

1.7 งบประมาณของโครงการ

1.7.1 ค่าอุปกรณ์ในการดำเนินโครงการ	1,000	บาท
1.7.2 ค่าเอกสารที่ใช้ในการดำเนินโครงการ	150	บาท
1.7.3 ค่าหมึกพิมพ์	250	บาท
1.7.4 ค่าเช่าเล่มโครงการ	600	บาท
รวมเป็นเงินทั้งสิ้น	2,000	บาท

หมายเหตุ ขออนุมัติด้วยเฉลี่ยทุกรายการ



บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 ความรู้เบื้องต้นเกี่ยวกับการเชื่อมต่อกับผู้ใช้อย่างเป็นธรรมชาติ (Natural User Interface)

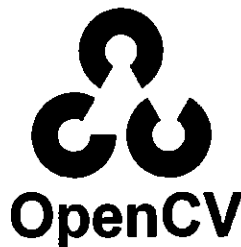
เทคโนโลยีการเชื่อมต่อระหว่างมนุษย์กับคอมพิวเตอร์ (Human Computer Interaction) ในปัจจุบัน มีระบบการเชื่อมต่อกับผู้ใช้งานแบบเป็นธรรมชาติ (Natural user interface) เพื่อเพิ่มความสะดวกสบายของการใช้เทคโนโลยีต่างๆ ในชีวิตประจำวันของมนุษย์ เช่น ระบบหน้าจอสัมผัส (Touch screen) ระบบการควบคุมด้วยเสียง (Voice control) ที่มีใช้อย่างแพร่หลายในปัจจุบัน เป็นต้น ซึ่งเทคโนโลยีที่กล่าวมานี้ ก็สามารถนำมาประยุกต์ใช้กับการควบคุมอากาศยานปีกหมุนแบบสไลโบพัด เพื่อลดปัญหาหรือขีดจำกัดต่างๆ ที่ใช้ในการควบคุมอากาศยานปีกหมุนแบบสไลโบพัดแบบเก่าได้

Human Computer Interaction (HCI) คือ การปฏิสัมพันธ์ระหว่างมนุษย์กับคอมพิวเตอร์ ซึ่งมีปัจจัยที่ทำให้คอมพิวเตอร์สามารถตอบโต้กับมนุษย์ได้อย่างประสบความสำเร็จ โดยจะประกอบไปด้วย 3 ส่วน คือ ผู้ใช้ คอมพิวเตอร์ และวิธีการทำงานร่วมกันระหว่างมนุษย์และคอมพิวเตอร์ โดยผู้ใช้งานหมายถึง บุคคลหรือกลุ่มคน ซึ่งมีความหลากหลาย และมีความสามารถในการปฏิสัมพันธ์กับคอมพิวเตอร์แตกต่างกันไป ส่วนคอมพิวเตอร์ หมายถึงตั้งแต่คอมพิวเตอร์ตั้งโต๊ะ ไปจนถึงระบบคอมพิวเตอร์ขนาดใหญ่ หรือเป็นอุปกรณ์ต่างๆ เช่น โทรศัพท์มือถือ แท็บเล็ต ก็เรียกว่าเป็นคอมพิวเตอร์ด้วยเหมือนกัน

2.2 ความรู้เบื้องต้นเกี่ยวกับไลบรารีของโอเพนซีวี

OpenCV ย่อมาจาก Open source Computer Vision เป็น Library ที่ถูกเขียนขึ้นจากภาษา C และ C++ ซึ่งไลบรารีโอเพนซีวี ได้มีการรวบรวมฟังก์ชันต่างๆ สำหรับใช้ในการประมวลผลภาพ (Image Processing) และคอมพิวเตอร์วิทัศน์ศาสตร์ (Computer Vision) เข้าไว้ด้วยกัน ซึ่งมีจุดเด่นที่สามารถรันได้บนทั้ง Linux, Mac OSX และ Windows และนอกจากนั้นยังมี Interface ที่ไว้เชื่อมต่อกับภาษาหรือ Tool อื่นๆ ด้วย อาทิ เช่น Python, Ruby, Matlab เป็นต้น

นอกจากนี้จุดเด่นอีกอย่างของ OpenCV คือสามารถประมวลผลภาพดิจิทัลได้ทั้งภาพนิ่ง และภาพเคลื่อนไหว เช่น ภาพจากกล้องวิดีโอ หรือไฟล์วิดีโอ เป็นต้น โดยไม่ยึดติดทางด้านฮาร์ดแวร์ และเป็นไลบรารีที่สร้างขึ้นเพื่อให้ผู้ใช้หรือนักพัฒนาสามารถใช้ฟังก์ชันในประยุกต์ใช้ในการพัฒนาชิ้นงานที่มีความซับซ้อนขึ้นมาได้ ตัวอย่าง เช่น การจดจำใบหน้า (Face Recognition), การจดจำม่านตา (Iris Recognition) เป็นต้น รูปที่ 2.1 แสดงตราสัญลักษณ์ของ OpenCV



รูปที่ 2.1 สัญลักษณ์ OpenCV

2.3 ความรู้เบื้องต้นเกี่ยวกับการประมวลผลภาพ

การประมวลผลภาพ (Image processing) หมายถึง เป็นหลักการนำภาพมาประมวลผลหรือคิดคำนวณด้วยคอมพิวเตอร์ เพื่อให้ได้ข้อมูลที่เรากำลังต้องการทั้งในเชิงคุณภาพและปริมาณ โดยมีขั้นตอนต่างๆ ที่สำคัญ คือ การทำให้ภาพมีความคมชัดมากขึ้น การกำจัดสัญญาณรบกวนออกจากภาพ การแบ่งส่วนของวัตถุที่เราสนใจออกมาจากภาพ เพื่อให้ได้ภาพในอีกลักษณะหนึ่ง แล้วนำภาพวัตถุที่ได้ไปวิเคราะห์หาข้อมูลเชิงปริมาณ ซึ่งสามารถเข้าใจได้โดยทั่วไป เช่น ขนาด รูปร่าง หรือทิศทางการเคลื่อนของวัตถุในภาพ จากนั้นเราสามารถนำข้อมูลเชิงปริมาณเหล่านี้ไปวิเคราะห์ และสร้างเป็นระบบ เพื่อใช้ประโยชน์ในงานด้านต่างๆ



รูปที่ 2.2 หลักการเบื้องต้นของการประมวลผลภาพ

เริ่มต้นเมื่อรับอินพุทภาพเข้ามาจะต้องทำการเปลี่ยนข้อมูลภาพให้อยู่ในรูปของไบนารีเสียก่อน นั่นคือ Digitize = Convert Analog to Digital เพื่อให้คอมพิวเตอร์สามารถเข้าใจข้อมูลภาพได้ จากนั้นทำการจัดการกับภาพ เช่น ปรับสี ความคมชัด เป็นต้น แล้วจึงนำภาพไปประมวลผลต่อไป

2.4 เกี่ยวกับ QuadCopter

2.4.1 ข้อมูลทั่วไปของ Quadcopter รุ่น AR.Drone 2.0 [1]

AR.Drone (รูปที่ 2.3) คือเฮลิคอปเตอร์บังคับ 4 ใบพัด ที่สามารถควบคุมด้วย ไอโฟน ,ไอแพด,ไอพอด และ สมาร์ทโฟน อื่นๆเป็น Remote controller simulator ผ่านช่องสัญญาณวิทยุ ภาย กล้องที่ติดมากับตัวเครื่องถูกปรับปรุงให้มีประสิทธิภาพด้วยความละเอียดระดับ HD 1280x720 สามารถถ่ายภาพวีดีโอระดับ 720p HD และส่งตรงมายัง ไอโฟน,ไอแพด,ไอพอด หรือ สมาร์ทโฟน ในแบบเวลาจริง (Real time) ใบพัดทั้ง 4 มุมทำจากพลาสติกโดยข้างใต้ใบพัดนอกจากจะเป็นมอเตอร์ของใบพัดแล้วยังถูกออกแบบให้เป็นขาตั้งเครื่องเวลาเรานำเครื่องบินขึ้นบินและลงจอด สำหรับตรงกลางตัวเครื่องเมื่อถอดกรอบออกมาแล้วจะเป็นที่ใส่แบตเตอรี่ โดยแบตเตอรี่มีขนาดความจุ 1,100 มิลลิแอมป์ สำหรับกล้อง นอกจากจะมีไว้ขนย้ายเครื่องบินแล้วยังเปรียบเสมือนเป็นฐานไว้ให้เครื่องบินขึ้นบินหรือจะมีไว้เพื่อฝึกลงจอดเครื่องได้อย่างดี สำหรับแอปพลิเคชันที่ใช้ในการควบคุมก็คือ AR.FreeFlight ที่มีทั้งในระบบปฏิบัติการแอนดรอยด์ ไอโอเอส และ windows 8

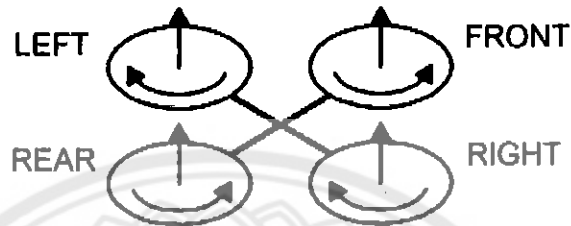


รูปที่ 2.3 ภาพ AR.Drone 2.0

2.4.2 หลักการการบินของ Quadcopter [2]

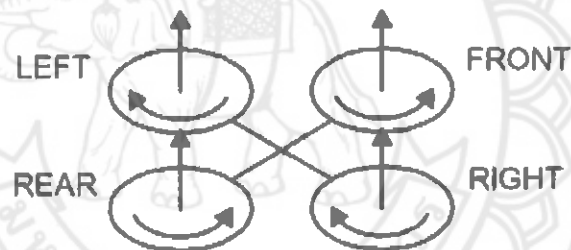
การบินของ Quadrotor จะเคลื่อนที่ 4 ทิศทาง หรือ เอาตามภาษาคนเล่น เฮลิคอปเตอร์ จะเรียกว่า 4 channel คือประกอบด้วย ขึ้น-ลง ,เดินหน้า-ถอยหลัง ,เอียงซ้าย-เอียงขวา และ หมุนซ้าย-หมุนขวา

1. Hovering (รูปที่ 2.4) หรือ การลอยตัวเฉยๆ ทำได้โดยควบคุมให้ความเร็ว ใบพัดทั้งสี่ตัว มีความเร็วที่เท่ากันครับ เพื่อสร้างแรงบิด (torque) และหักล้างแรงบิด ดูจากรูปจะเห็นว่า ใบพัด จะหมุนกันคนละทิศทาง ใบพัดหน้าและหลัง จะหมุน ตามเข็มนาฬิกา ใบพัด ซ้ายและขวาจะหมุนทวนเข็มนาฬิกา ทำให้เครื่องบินไม่หมุนตัว



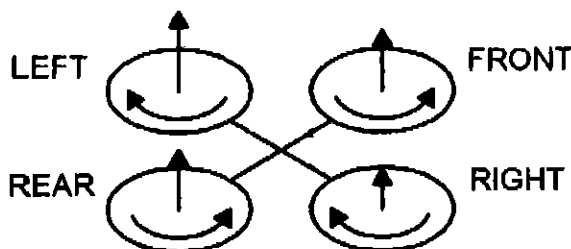
รูปที่ 2.4 แสดงการบินลอยตัวอยู่กับที่ (Hovering)

2. Throttle (รูปที่ 2.5) คำนวณ ความเร็ว ให้เครื่องบิน บิน ขึ้นลง ดูจากรูป ใบพัดทั้งสี่ใบจะต้องเพิ่มความเร็ว ทุกใบพัด ที่เท่ากัน ทำให้เครื่องบิน ลอยตัวขึ้นได้



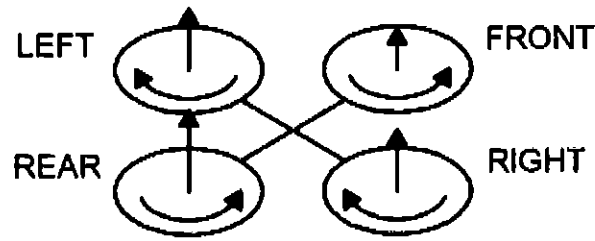
รูปที่ 2.5 แสดงการบินขึ้น-ลง (Throttle)

3. Roll (รูปที่ 2.6) เอียงตัวซ้าย-ขวา จากรูป ใบพัด หน้า (FRONT) หลัง (REAR) จะ ความเร็วเท่าเดิม แต่ความเร็วใบพัดซ้าย (LEFT) จะหมุนเร็วขึ้น ทิศทางนี้จะยกตัว ใบพัดขวา (Right) จะช้าลงทิศทางนี้จะตกลง ทำให้เกิดการ เอียงตัวไปทางขวาได้ ส่วนเอียงตัวซ้าย ก็คล้ายกัน



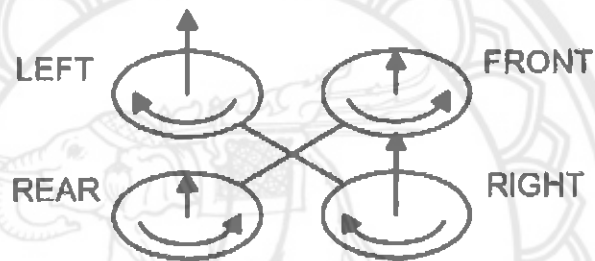
รูปที่ 2.6 แสดงการบินเอียงตัวซ้าย - ขวา (Roll)

4. Pitch (รูปที่ 2.7) เอียงหน้าและหลัง อันนี้คล้ายๆกับการ Roll แต่เปลี่ยนเป็น ใบพัด ซ้าย (LEFT) ขวา (RIGHT) จะความเร็วคงที่ แต่ความเร็วใบพัดหลัง (REAR) จะหมุนเร็วขึ้น ทางหลัง จะยก ใบพัดหน้า (FRONT) จะหมุนช้ากว่า ทางหน้าจะตก จึงทำให้เครื่องบินเอียงไปข้างหน้า



รูปที่ 2.7 แสดงการบินเอียงหน้า - หลัง (Pitch)

5. Yaw (รูปที่ 2.8) หรือการหมุนตัว อันนี้ง่ายๆ ให้ความเร็วใบพัด หน้า (FRONT) - หลัง (REAR) มากกว่า ความเร็วใบพัด ซ้าย (LEFT) - ขวา (RIGHT) เพื่อให้แรงบิด ด้านซ้าย หรือ ขวา มากกว่า จึงทำให้เครื่องบินหมุนตัวได้

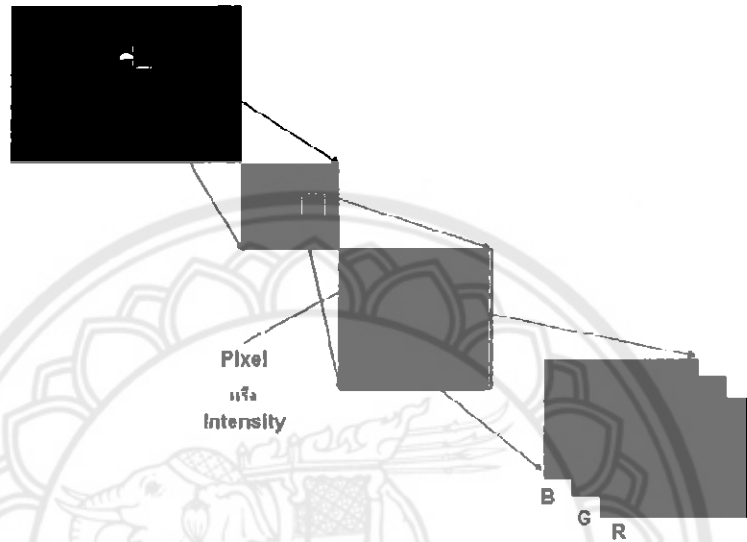


รูปที่ 2.8 แสดงการบินแบบหมุนตัว (Yaw)

2.5 การประมวลผลภาพ (Image Processing)

2.5.1 ภาพสี (Color Image) [4]

ค่าในแต่ละพิกเซลของภาพสี จะประกอบไปด้วยเวกเตอร์ที่แสดงถึงค่าของสีแดง สีเขียว และสีน้ำเงิน อย่างละ 8 บิต (สีแต่ละสีมีความเข้มแสง 0-255 ค่า) ดังนั้น ภาพสี 1 พิกเซล จะประกอบไปด้วยจำนวนบิตทั้งหมด 24 บิต ทำให้ภาพสีมีจำนวนสีที่เป็นไปได้ทั้งหมด 2^{24} สี ซึ่งแสดงดังรูปที่ 2.9

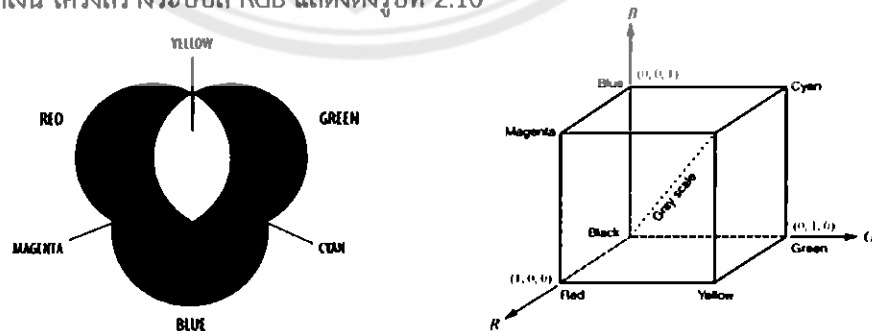


รูปที่ 2.9 ภาพสี และค่าในแต่ละพิกเซล

2.5.2 ระบบสี [3]

2.5.2.1 ระบบสี RGB

โครงสร้างระบบสี RGB เกิดจากการรวมกันของสเปกตรัมของแสงสีแดง เขียว และน้ำเงิน ในสัดส่วนความเข้มที่ต่างกัน จุดที่แสงทั้งสามสีรวมกัน คือ สีขาว บางครั้งเราเรียกสีที่มองเห็นในโครงสร้างระบบสี RGB ว่าเป็น สารเติมแต่งสี ลักษณะการรวมกันเช่นนี้ ถูกใช้สำหรับการส่องแสง ทั้งบนจอภาพทีวี และจอคอมพิวเตอร์ ซึ่งสร้าง จากสารที่ให้กำเนิดแสงสีแดง สีเขียว และสีน้ำเงิน โครงสร้างระบบสี RGB แสดงดังรูปที่ 2.10



รูปที่ 2.10 โครงสร้างระบบสี RGB

2.5.2.2 ระบบสี HSV

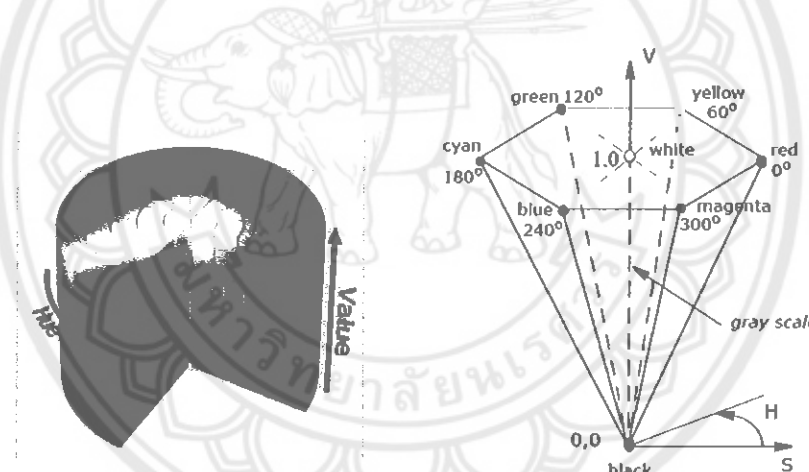
ระบบสี HSV ย่อมาจาก Hue Saturation Value เป็นการพิจารณาสีโดยใช้ Hue Saturation และ Value ซึ่ง Hue คือ ค่าสีของสีหลัก (แดง เขียวและน้ำเงิน) ในทางปฏิบัติจะอยู่ระหว่าง 0 และ 255 ซึ่งถ้า Hue มีค่าเท่ากับ 0 จะแทนสีแดงและเมื่อ Hue มีค่าเพิ่มขึ้นเรื่อย ๆ สีก็จะเปลี่ยนแปลงไปตามสเปกตรัมของสีจนถึง 256 จึงจะกลับมาเป็นสีแดงอีกครั้ง ซึ่งสามารถแทนให้อยู่ในรูปขององศาได้ดังนี้ คือ สีแดงเท่ากับ 0 องศา สีเขียวเท่ากับ 120 องศา และสีน้ำเงินเท่ากับ 240 องศา Hue สามารถคำนวณได้จากระบบสี RGB ได้ตามสมการที่ (2-1), (2-2) และ (2-3)

$$\text{red}_h = \text{red} - \min(\text{red}, \text{green}, \text{blue}) \quad (2-1)$$

$$\text{green}_h = \text{green} - \min(\text{red}, \text{green}, \text{blue}) \quad (2-2)$$

$$\text{blue}_h = \text{blue} - \min(\text{red}, \text{green}, \text{blue}) \quad (2-3)$$

แบบจำลอง HSV (Hue-Saturation-Value) จะมีการแยกความสว่างของภาพ (Luminance) ออกจากข้อมูลสี (Chromaticity) ทำให้สามารถเปรียบเทียบค่าสีได้ง่าย โดยผลลัพธ์ที่ได้ H คือค่าสีบริสุทธิ์ S คือค่าแสงผสมกับค่าสีบริสุทธิ์ H และ V คือค่าความสว่างของภาพ แสดงดังรูปที่ 2.11



รูปที่ 2.11 โครงสร้างระบบสี HSV

2.5.3 การแปลงภาพสีให้เป็นภาพในระดับเทา (Grayscale) [4]

รูปที่ 2.12 เป็นการแปลงภาพสีในระบบสี RGB ให้เป็นภาพระดับเทา (Grayscale) โดยใช้การคำนวณค่าสีตามสูตรทางคณิตศาสตร์ดังสมการที่ (2-4)

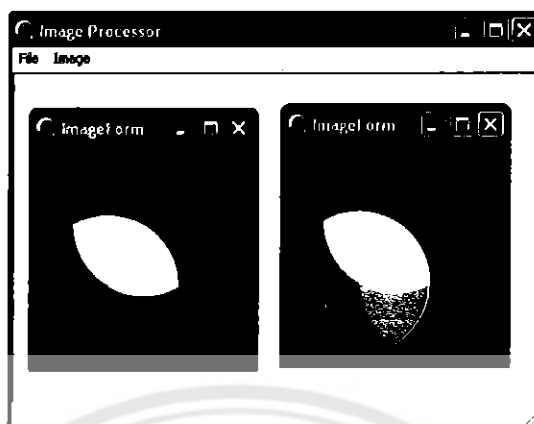
$$\text{Gray} = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad (2-4)$$

เมื่อ

Gray = ค่าความเข้มของสีเทาโดยจะมีค่าระหว่าง 0 - 255

R = ค่าความเข้มของสีแดงโดยจะมีค่าระหว่าง 0 - 255

- G = ค่าความเข้มของสีเขียวโดยจะมีค่าระหว่าง 0 - 255
 B = ค่าความเข้มของสีน้ำเงินโดยจะมีค่าระหว่าง 0 - 255



รูปที่ 2.12 การแปลงภาพสี RGB ไปเป็นภาพระดับสีเทา

ที่มา : <http://imageprocessingindelpi.blogspot.com/2008/08/rgb-to-gray-scale-conversion-using.html>

2.5.4 การตรวจจับใบหน้าด้วยโอเพนซีวี (Face Detection Using OpenCV) [5]

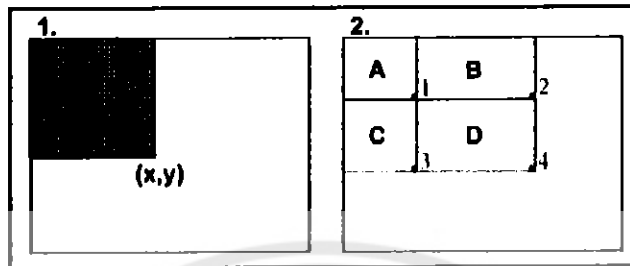
การตรวจจับใบหน้าของ OpenCV ใช้วิธีของ Paul Viola และ Michael Jones ซึ่งถูกตีพิมพ์ในปี 2001 ถูกเรียกง่าย ๆ ว่า “Viola-Jones method” โดยใช้การตรวจจับวัตถุในภาพร่วมกับแนวคิดหลัก 4 แนวคิดดังนี้

- ใช้ Haar features
 - ใช้การรวมภาพ (Integral Image) เพื่อตรวจจับลักษณะเด่นอย่างรวดเร็ว (Rapid Feature Detection)
 - ใช้ The Adaboost machine-learning method
 - ใช้ Cascaded classifier เพื่อรวมลักษณะเด่นต่างๆเข้าด้วยกันอย่างมีประสิทธิภาพ
- การมีอยู่ของลักษณะเด่นของ Haar (รูปที่ 2.13) ถูกกำหนดโดยการลบค่าเฉลี่ยบริเวณจุดภาพมืดออกจากค่าเฉลี่ยบริเวณจุดภาพสว่าง ถ้าผลลัพธ์ที่ได้มีค่ามากกว่าค่าขีดแบ่ง (ถูกตั้งค่าในระหว่างการเรียนรู้) แสดงว่ามีลักษณะเด่นของ Haar อยู่



รูปที่ 2.13 แสดงตัวอย่างลักษณะเด่นของ Haar ที่ถูกใช้ใน OpenCV

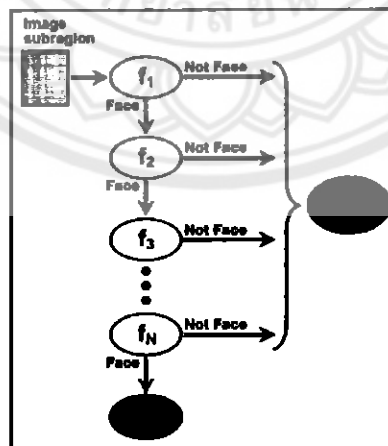
การตรวจสอบหาลักษณะเด่นของ Haar ในแต่ละส่วนของภาพอย่างมีประสิทธิภาพนั้น Viola และ Jones ใช้เทคนิคที่เรียกว่าการรวมภาพ (Integral Image) โดยการรวมจุดภาพเล็กๆเข้าด้วยกัน จากรูปที่ 2.14 การจะหาค่าของรูปสี่เหลี่ยม D คือการนำ A+B+C+D (ตำแหน่งที่ 4) ลบออกด้วย A+B (ตำแหน่งที่ 2) และ A+C (ตำแหน่งที่ 3) และบวกเพิ่มเข้าไปด้วย A (ตำแหน่งที่ 1) นั่นคือ $D = (A+B+C+D) - (A+B) - (A+C) + (A)$ ซึ่งมีค่าเท่ากับ $(x_4,y_4) - (x_2,y_2) - (x_3,y_3) + (x_1,y_1)$



รูปที่ 2.14 แสดงเทคนิคการรวมภาพ

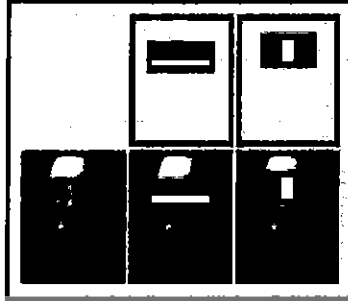
- (1) หลังจากที่เรารวมภาพแล้ว จุดภาพที่ตำแหน่ง (x,y) จะรวมค่าของทุกจุดภาพในสี่เหลี่ยมสีดำ
- (2) ผลรวมค่าของทุกจุดภาพในสี่เหลี่ยม D คือ $(x_4,y_4) - (x_2,y_2) - (x_3,y_3) + (x_1,y_1)$

ในการเลือกลักษณะเด่นของ Haar และการตั้งระดับค่าขีดแบ่ง Viola และ Jones ใช้ machine-learning method ที่เรียกว่า “Adaboost” ซึ่งรวม weak classifier (ได้รับคำตอบที่ถูกต้องมากกว่าการเดาสุ่มเพียงเล็กน้อย) จำนวนมากเข้าด้วยกันเพื่อสร้าง strong classifier โดยใช้หลักที่ว่า weak classifier แต่ละตัวค่อยๆ ผลักดันคำตอบสุดท้ายไปในทิศทางที่ต้องการ Adaboost จะทำการเลือกชุดของ weak classifier ในการรวมและกำหนดน้ำหนักของแต่ละตัว การรวมกันของน้ำหนักก็คือ strong classifier นั่นเอง Viola และ Jones รวมชุดของ Adaboost classifiers เป็นเหมือนโครงข่ายแสดงในรูปที่ 2.15 ซึ่งมีประสิทธิภาพสูงในการแบ่งพื้นที่ภาพ



รูปที่ 2.15 แสดงสายโซ่ของตัวกรอง บริเวณเล็กๆของภาพซึ่งสามารถผ่านตัวกรองทั้งหมดจะถูกจัดว่าเป็นใบหน้า ส่วนที่เหลือถูกจัดว่าไม่ใช่ใบหน้า

ตำแหน่งของตัวกรองในลำดับอยู่บนพื้นฐานของค่าน้ำหนักความสำคัญที่ Adaboost กำหนดไว้ ตัวกรองที่มีค่าน้ำหนักสูงสุดจะอยู่ในลำดับแรกเพื่อกำจัดส่วนของภาพที่ไม่ใช่ใบหน้าออกไป



รูปที่ 2.16 แสดงตัวอย่างของตัวกรองสองตัวแรกใน Viola-Jones Cascade

จากรูปที่ 2.16 จะพบว่าตัวกรองแรกใช้ความจริงที่ว่าบริเวณแก้มสว่างกว่าบริเวณดวงตา และตัวกรองตัวที่สองใช้ความจริงที่ว่าบริเวณสันจมูกจะสว่างกว่าบริเวณดวงตา

2.5.5 การประมวลผลภาพกับรูปร่างและโครงร่างของภาพ (Morphological Image Processing) [6]

การประมวลผลภาพกับรูปร่างและโครงร่างของภาพ (Morphological Image Processing) เป็นการประมวลผลภาพโดยการเปลี่ยนแปลงลักษณะรูปร่างหรือโครงร่างของภาพโดยใช้การดำเนินการพื้นฐานโดยทั่วไป

Structure Element คือ เมตริกที่ถูกนิยามให้เป็นรูปร่างและขนาดที่เป็น Neighborhood สำหรับการทำ Morphological Operation โดยในเมตริกจะประกอบด้วยค่า Binary 2 ค่าคือ 0 และ 1 ซึ่งสามารถมีรูปร่างตามที่เรากำหนด โดยที่ 1 จะกำหนดเป็น Neighborhood

2.5.5.1 การขยายภาพ (Dilation)

การ Dilation คือเทคนิคที่ใช้ขยายขอบของ Foreground หรือ Background ของ Image ตามที่กำหนด โดยทั่วไปมักใช้ในการเติมเต็มหลุม (Hole Filling) ซึ่งสามารถอธิบายด้วยสมการดังต่อไปนี้

$$A \cdot B = (AB) \oplus B \quad (2-5)$$

2.5.5.2 การย่อภาพ (Erosion)

การ Erosion คือเทคนิคที่ใช้ในการลบ (Remove) ดิ่งหรือจะหยอยของขอบ (Spike of edges) ของ Region ซึ่งสามารถอธิบายด้วยสมการดังต่อไปนี้

$$A \ominus B = \{p \mid B_p \subseteq A\} \quad (2-6)$$

2.5.5.3 การเปิดพื้นที่ว่างภายในภาพ (Opening)

การเปิดพื้นที่ว่างภายในภาพ (Opening) คือเอารูปภาพประมวลผลโดยวิธี Erosion จากนั้นนำภาพที่ได้ไปผ่านการประมวลผล Dilatation เพราะเนื่องการ Noise จะหายไปตอนทำ Erosion แต่ขนาดของวัตถุจะเล็กลงจึงใช้วิธีการทำ Dilatation เพื่อขยายภาพกลับมา ซึ่งสามารถอธิบายด้วยสมการดังต่อไปนี้

$$A \circ B = (A \ominus B) \oplus B \quad (2-7)$$

2.5.5.4 การปิดช่องว่างภายในภาพ (Closing)

การปิดพื้นที่ว่างภายในภาพ (Closing) คือกระบวนการที่ตรงข้ามกับ Opening โดยเป็นการเอารูปภาพประมวลผลโดยวิธี Dilatation ก่อนจากนั้นนำภาพที่ได้ไปผ่านการประมวลผลด้วยวิธี Erosion

$$A \cdot B = (A \oplus B) \ominus B \quad (2-8)$$

2.5.6 การแปลงภาพเป็นขาวดำ (Threshold) [7]

การสร้างภาพไบนารีด้วยวิธีเทรชโฮลด์ มีหลักการในการเลือกจุดว่าจุดนั้นจะเป็นสีขาวหรือสีดำ โดยจะเปรียบเทียบกับค่าคงที่ค่าหนึ่งซึ่งคือค่าเทรชโฮลด์ โดยถ้ากำหนด threshold = 100 แล้ว pixel ในรูปที่มีค่าน้อยกว่า 100 จะถูกเปลี่ยนเป็น 0 และ pixel ที่มีค่ามากกว่า 100 จะถูกเปลี่ยนเป็น 255 ดังสมการ

$$g(x,y) = 0 \quad \text{if } f(x,y) < \text{threshold value} \quad (2-9)$$

$$g(x,y) = 255 \quad \text{if } f(x,y) \geq \text{threshold value} \quad (2-10)$$

เมื่อ $f(x,y)$ คือ ตำแหน่งพิกเซลของภาพต้นฉบับ

$g(x,y)$ คือ ตำแหน่งพิกเซลของภาพผลลัพธ์

2.5.7 คอนทัวร์ (Contour)

คอนทัวร์ คือเส้นขอบที่แสดงถึงรูปร่างและขอบเขตพื้นที่ของวัตถุที่อยู่ในภาพ เมื่อมีวัตถุในภาพหลายวัตถุ หรือมีการแบ่งวัตถุในภาพออกเป็นหลายส่วน ซึ่งคอนทัวร์จะบอกให้ทราบว่าวัตถุมีขอบเขตและมีพื้นที่เท่าไร และยังสามารถบอกได้ว่า วัตถุในภาพนั้น เป็นวัตถุเดียวกันหรือไม่ ดังรูปที่ 2.17



รูปที่ 2.17 เส้นคอนทัวร์ของวัตถุ

<http://fivedots.coe.psu.ac.th/~somchai/Projects/ShapeDetector/ScreenShot/0.0.15/>

2.5.8 การหาขอบภาพ (Edge detection) [8]

การหาขอบภาพ คือ การตรวจสอบว่าเส้นขอบลากผ่านหรือใกล้เคียงกับจุดใด โดยวัดจากการเปลี่ยนแปลงของความเข้มในตำแหน่งที่ใกล้เคียงกับจุดดังกล่าว ซึ่งวิธีการหาขอบสามารถแบ่งได้เป็น 2 กลุ่มหลัก คือ วิธี Gradient และวิธี Laplacian โดยในแต่ละวิธีมีรายละเอียดดังต่อไปนี้

1. วิธี Gradient วิธีนี้จะหาขอบโดยการหาจุดต่ำสุดและจุดสูงสุดในรูปของอนุพันธ์อันดับหนึ่งของภาพ โดยจุดที่เป็นขอบจะอยู่ในส่วนที่เหนือค่าสมมุติที่ตั้งไว้เหมือนเป็นจุดตรวจสอบ (threshold) จึงอาจทำให้เส้นขอบที่ได้มีลักษณะหนา ตัวอย่างวิธีการหาขอบของกลุ่มนี้ เช่น Roberts, Prewitt, Sobel และ Canny เป็นต้น

2. วิธี Laplacian นี้จะหาขอบโดยใช้อนุพันธ์อันดับ 2 โดยใช้จุดที่ค่า y เป็น 0 (Zerocrossing) ซึ่งวิธีนี้จะใช้เวลาในการคำนวณมากกว่าวิธี Gradient

3. วิธี Canny จากการปรับภาพให้เรียบ (Smoothing) ด้วยตัวกรองเกาส์เซียน (Gaussian filter) เพื่อกำจัดสัญญาณรบกวน หลังจากนั้น คำนวณค่าขนาด (magnitude) และทิศทาง (orientation) ของ gradient โดยใช้การหาอนุพันธ์อันดับหนึ่ง ในขั้นตอนถัดมาทำให้ได้ขอบที่บาง (โดยใช้ nonmaxima suppression กับ gradient magnitude) และในขั้นตอนสุดท้าย ระบุพิกเซลที่เป็นขอบและช่วยเชื่อมต่อขอบของภาพ (โดยใช้ double thresholding algorithm)

2.5.9 การแบ่งส่วนภาพและแยกภาพ (Image segmentation and extraction)

เป็นการดึงส่วนสำคัญหรือส่วนที่ต้องการ ออกจากภาพที่ปรากฏตามลักษณะต่างๆ ที่สนใจ เช่น สนใจเฉพาะส่วนที่เป็นรูปสี่เหลี่ยม หรือ สนใจส่วนที่เป็นสีที่ต้องการ ก็ใช้เทคนิคนี้ทำการตรวจจับลักษณะต่างๆภายในภาพว่ามีส่วนที่ต้องการอยู่ในภาพนั้นหรือไม่

2.5.10 การติดตามวัตถุ (Object Tracking)

การตรวจจับวัตถุ เป็นเทคนิคที่ใช้ในการตรวจจับตำแหน่งและการเคลื่อนไหวของวัตถุ ซึ่งเป็นเทคนิคที่มีความสำคัญมากในการเพิ่มความจริงเสริม (Augmented Reality) โดยอาศัยศาสตร์ด้านการประมวลผลภาพ ภาพหนึ่งภาพมีข้อมูลอยู่มากมาย การที่จะดึงข้อมูลที่ต้องการออกมา (Image segmentation and extraction) โดยแยกจากข้อมูลด้านอื่นๆ รวมไปถึงสิ่งรบกวนต่างๆ (Noises) ที่ไม่ต้องการ ต้องอาศัยขั้นตอนวิธีที่เหมาะสม ในกรณีนี้เป็นการนำเอาหลักการด้านการประมวลผลภาพ

โดยในการพัฒนาโครงการนี้จะอาศัยเทคนิคการตรวจจับวัตถุในการ รายการอวัยวะของคนด้วยการตรวจจับสีผิวของคน เพื่อนำข้อมูลตำแหน่งที่ได้ไปใช้ในการประมาณท่าทางของมนุษย์ต่อไป

2.5.11 ประมาณท่าทางของมนุษย์ (Human pose estimation)

ในการประมาณท่าทางของมนุษย์ นี้จะใช้ข้อมูลจากภาพทั้ง ภาพเงาทึบ (Silhouette) รูปเค้าโครง (Contours) เส้นขอบ (Edge) ท่าทาง (Motions) และสี ในการประมาณตำแหน่งของคนในท่าทางต่างๆ เพื่อสร้างแบบจำลองของคน รวมไปถึงการประมาณตำแหน่งของภาพคนที่เกิดการซ้อนทับขึ้นด้วย

ซึ่งในการพัฒนานี้จะสนใจเฉพาะร่างกายส่วนบน โดยใช้ข้อมูลจากการ ระบุร่างกายส่วนบน (Upper body tracking) ได้แก่ มือ แขน และ ศีรษะ เป็นข้อมูลในการประมาณตำแหน่งท่าทางของคน

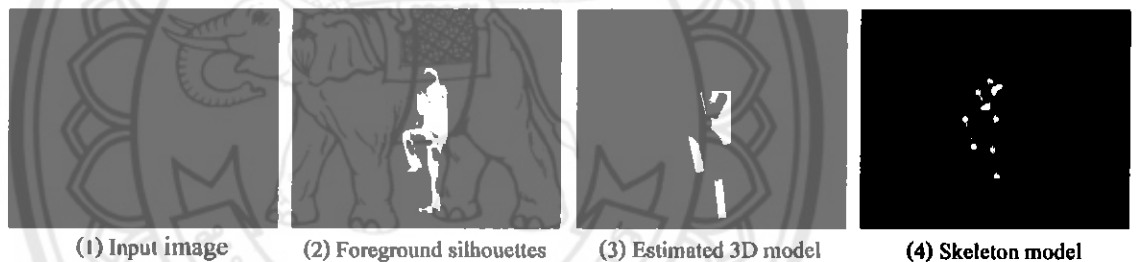
2.5.12 ผสานวัตถุเสมือน (Merging virtual object)

เป็นการใช้หลักการความจริงเสริม (Augmented reality) โดยมีตำแหน่งของคนเป็นเครื่องหมาย เทคนิคนี้จะใช้ค่าตำแหน่งจากการประมาณที่ได้ นำมาปรับตำแหน่งของวัตถุเสมือน (virtual object) ให้อยู่ในตำแหน่งตรงกับภาพ ซึ่งจะใช้ภาพที่เกิดการซ้อนทับ กับภาพที่ไม่เกิดการซ้อนทับ มารวมเข้ากับภาพต้นฉบับ เพื่อให้ภาพที่ได้มีความถูกต้องในเรื่องลำดับความลึกและการซ้อนทับของวัตถุ

2.6 งานวิจัยที่เกี่ยวข้อง

2.6.1 Human Pose Estimation From Monocular Image Captures [9]

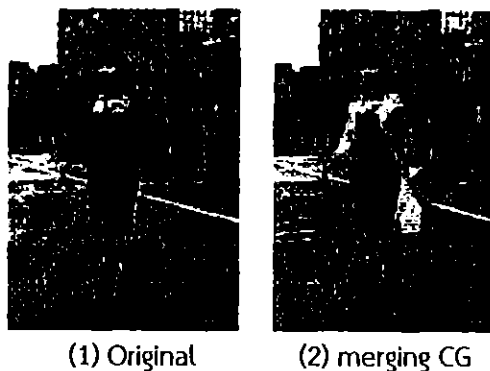
เสนอโดย Huei-Yung Lin, Ting-Wen Chen, Chih-Chang Chen, Chia-Hao Hsieh และ Wen-Nung Lie เป็นงานวิจัยเกี่ยวกับการประมาณท่าทางของมนุษย์ โดยใช้กล้องเพียงตัวเดียว โดยใช้หลักการปรับขนาดและขยับตำแหน่งของโมเดล 3 มิติให้เข้ากับภาพ ภาพเงาที่ทับ (silhouettes) โดยรองรับปัญหาการซ้อนทับกันของภาพด้วย ดังรูปที่ 2.18



รูปที่ 2.18 แสดงผลงานวิจัยเกี่ยวกับการประมาณท่าทางของมนุษย์

2.6.2 Automatic Registration of Virtual Objects onto Human Image Sequences [10]

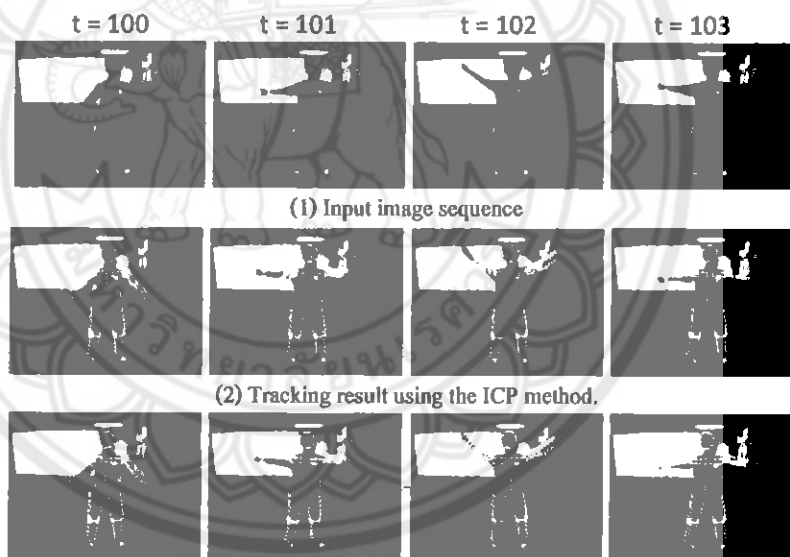
เสนอโดย Hoshino, J., Salto, H. และ Yamamoto, M. ได้กล่าวถึงการรวมภาพคอมพิวเตอร์กราฟฟิค (CG) ลงบนภาพวิดีโอคน เสมือนกับว่ามีวัตถุนั้นอยู่จริง โดยอาศัยการประมาณท่าทางของมนุษย์แบบ 3 มิติ (3D Human Pose Estimation) เพื่อนำข้อมูลมาใช้ในการปรับภาพคอมพิวเตอร์กราฟฟิค ให้อยู่ในตำแหน่งที่ถูกต้องแล้วจึงรวมเข้ากับภาพต้นฉบับ โดยใช้ทั้งภาพที่เกิดการซ้อนทับ และไม่เกิด มารวมกันให้เกิดความสมจริงในกรณีเกิดการซ้อนทับกัน ดังรูปที่ 2.19



รูปที่ 2.19 แสดงการรวมภาพคอมพิวเตอร์กราฟฟิกลงบนภาพวิดีโอคน

2.6.3 A novel fitting algorithm using the ICP and the particle filters for robust 3d human body motion tracking [11]

เสนอโดย Daehwan Kim และ Daijin Kim ได้กล่าวถึงการใช้วิธีการ ICP (Iterative Closet Point) ในการจับตำแหน่งของวัตถุ (ในที่นี้คือร่างกายคน) ซึ่งมีข้อดีคือ วิธีการนี้จะสามารถติดตามการเคลื่อนไหวของร่างกายได้อย่างรวดเร็วกว่าวิธีอื่นๆ ซึ่งมีการแสดงผลโดยการสร้างแบบจำลองร่างกาย นำมาวางทับลงบนภาพ ดังรูปที่ 2.20



รูปที่ 2.20 แสดงภาพแบบจำลองท่าทางของร่างกาย

2.6.4 Vision-Based Two Hand Detection And Tracking [12]

เสนอโดย Jiajun Wen และ Yinwei Zhan ได้กล่าวถึงวิธีการติดตามมือที่ปรากฏอยู่ในภาพ โดยการ ทำภาพให้เป็นภาพไบนารี (Binary image) กำหนดให้สีเนื้อของร่างกายเป็นสีขาว นอกจากนั้นให้เป็นสีดำ โดยใช้ลักษณะสีแบบ YCrCb หลังจากได้ภาพไบนารี แล้วก็ทำการจับตำแหน่งของปลายนิ้วมือแต่ละนิ้วว่าอยู่ที่ตำแหน่งใด ก็จะสามารถรู้ได้ว่ามือนั้นปรากฏอยู่ที่ตำแหน่งใดภายในภาพ ดังรูปที่ 2.21



รูปที่ 2.21 แสดงการทำตำแหน่งของมือคนโดยใช้ภาพไบนารี

2.6.5 Robust real-time upper body limb detection and tracking [13]

เสนอโดย Matheen Siddiqui และ Gerard Medioni ได้กล่าวถึงวิธีการติดตามแขนที่ปรากฏอยู่ในภาพ โดยมีวิธีการคือ ในขั้นแรกจะทำการตรวจจับว่าใบหน้าอยู่ส่วนใดของภาพ เมื่อได้ตำแหน่งของใบหน้าก็จะรู้ถึงสีร่างกายของผู้ใช้งาน จากนั้นจะทำการประมาณตำแหน่งของส่วนแขนตอนบน(ส่วนหัวไหล่ไล่ลงมาถึงข้อศอก) จากนั้นก็จะได้ว่าสีของร่างกายที่จับมาได้ตรงที่อยู่ใกล้กับส่วนแขนตอนบนที่ประมาณตำแหน่งขึ้นมา นั่นก็คือส่วนของแขนที่ต้องการ ดังรูปที่ 2.22

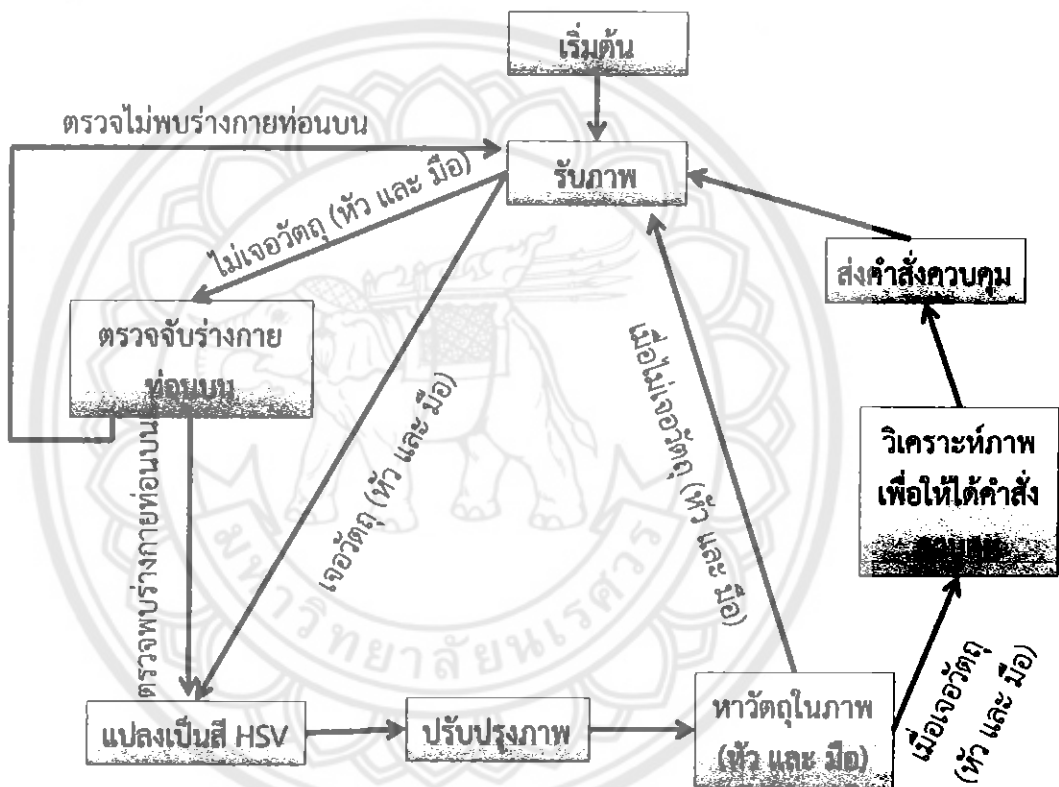


รูปที่ 2.22 แสดงภาพการระบุตำแหน่งอวัยวะของร่างกายส่วนบนแบบเวลาจริง (Real-time)

บทที่ 3 ขั้นตอนการดำเนินงาน

ขั้นตอนการทำงานการควบคุมอากาศยานปีกหมุนแบบสี่ใบพัด จะมีขั้นตอนการทำงานดังต่อไปนี้

- 3.1 ขั้นตอนการรับภาพจากอากาศยานปีกหมุน
- 3.2 ขั้นตอนการประมวลผลภาพขั้นต้น
- 3.3 ขั้นตอนการวิเคราะห์ภาพเพื่อให้ได้คำสั่งควบคุมอากาศยานปีกหมุน
- 3.4 ขั้นตอนการส่งคำสั่งควบคุมให้กับอากาศยานปีกหมุน



รูปที่ 3.1 Flow การทำงานประมวลผลคำสั่งควบคุมอากาศยานปีกหมุน

จากรูปที่ 3.1 เมื่อกำลังติดตามอยู่กับอากาศยานปีกหมุนรับภาพมาแล้ว ขั้นแรกจะทำการตรวจสอบว่าเจอวัตถุ(ศีรษะ และมือ) หรือยัง ถ้ายังไม่เจอก็จะให้วนลูปหาต่อไปจนเจอ หลังจากเจอแล้วจะทำการนำภาพมาแปลงให้อยู่ในระบบสี HSV เพื่อง่ายต่อการตรวจจับสีของผิว และเมื่อเจอสีของผิว(ศีรษะ และมือ) ก็จะทำการวิเคราะห์ภาพท่าทางการเคลื่อนไหวของมือ ว่ามีลักษณะตรงตามเงื่อนไขของการควบคุมแบบใด และเมื่อได้คำสั่งการควบคุมแล้ว คำสั่งนั้นก็จะถูกส่งไปที่อากาศยานปีกหมุน เพื่อให้อากาศยานปีกหมุนปฏิบัติตามคำสั่งนั้นๆ

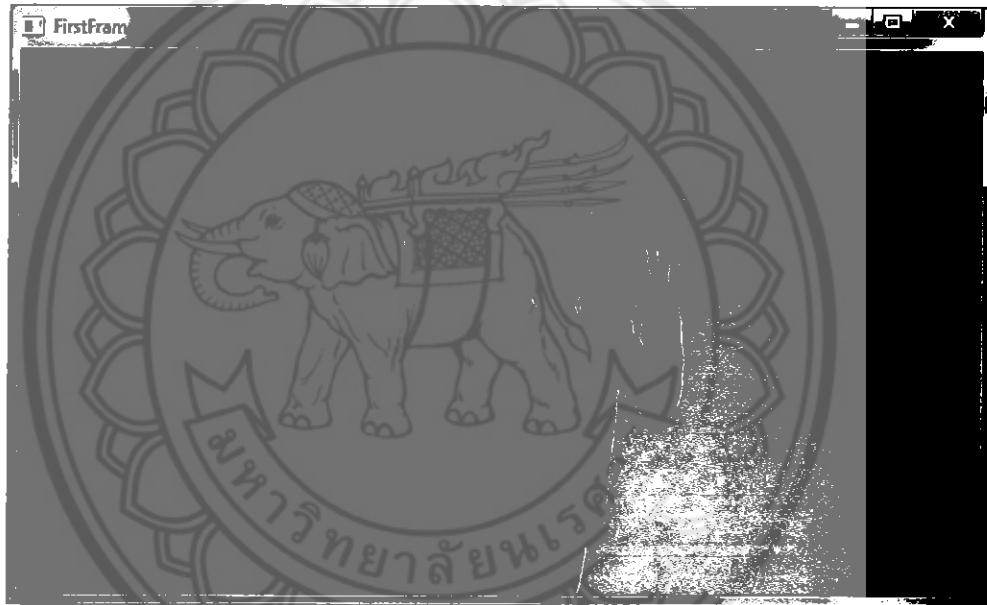
3.1 ขั้นตอนการรับภาพจากอากาศยานปีกหมุน

3.1.1 การเชื่อมต่อคอมพิวเตอร์กับอากาศยานปีกหมุน

ขั้นแรกให้เปิดเครื่องอากาศยานปีกหมุนให้เรียบร้อย จากนั้นเชื่อมต่อคอมพิวเตอร์กับอากาศยานปีกหมุนผ่านระบบ wifi ซึ่งจะได้ ip address 192.168.1.2

3.1.2 การรับภาพจากกล้องของอากาศยานปีกหมุน

หลังจากที่เชื่อมต่ออากาศยานปีกหมุนกับคอมพิวเตอร์เรียบร้อยแล้ว ให้ทำการรันโปรแกรมเพื่อเป็นการสั่งให้กล้องของอากาศยานปีกหมุนเปิดกล้องเพื่อรับภาพ และส่งให้กับคอมพิวเตอร์ ในการเขียนโปรแกรมจาก source code ที่เตรียมไว้สามารถใช้ class ARDrone (ในที่นี่ใช้ตัวแปรของ class ชื่อว่า ardrone) ได้เลย ซึ่งขั้นตอนแรกต้องเชื่อมต่อกับอากาศยานปีกหมุนกับโปรแกรมโดยใช้ "ardrone.open();" แล้วทำการรับภาพโดยใช้ "ardrone.getImage();" นำไปใส่ตัวแปรของ opencv ประเภท "cv::Mat image" จากนั้นแสดงภาพออกมาโดยคำสั่ง "cv::imshow("camera", image);"



รูปที่ 3.2 ภาพแสดงการรับภาพจากอากาศยานปีกหมุน

จากรูปที่ 3.2 เมื่อทำการรันโปรแกรม จะปรากฏหน้าต่าง ซึ่งเป็นภาพที่ได้จากกล้องของอากาศยานปีกหมุน

3.2 ขั้นตอนการประมวลผลภาพขั้นต้น

เมื่อรับภาพท่าทางการเคลื่อนไหวของร่างกายมาแล้ว ขั้นตอนนี้จะเป็นการนำภาพที่ได้มาปรับปรุงใหม่ เพื่อที่จะสามารถนำไปใช้งานในขั้นตอนถัดไป ซึ่งขั้นตอนนี้มีความสำคัญต่อประสิทธิภาพและความถูกต้องแม่นยำโดยรวมของโปรแกรม เพราะถ้าการประมวลผลภาพขั้นต้นผิด จะทำให้ผลลัพธ์ของโปรแกรมทั้งหมด เกิดความผิดพลาดตามไปด้วย ซึ่งขั้นตอนในการประมวลผลภาพ ซึ่งมีขั้นตอนโดยรวมดังนี้

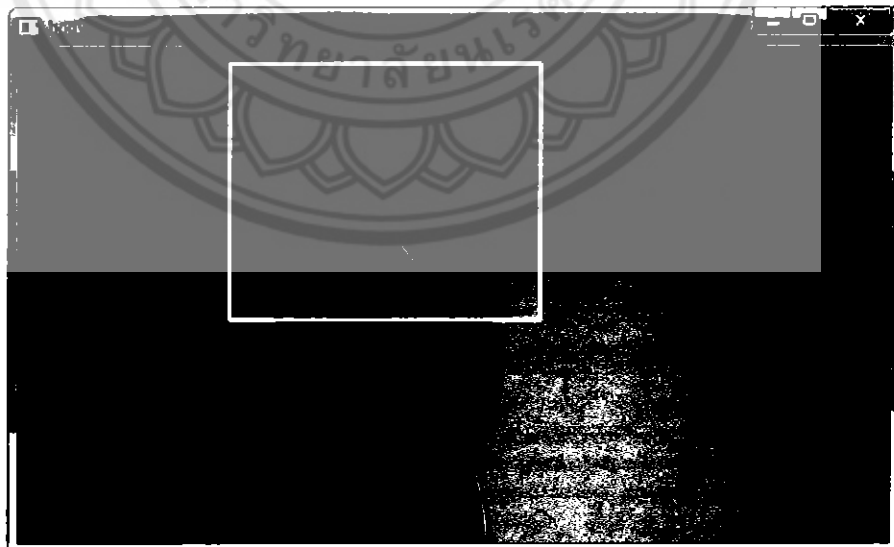
- ตรวจจบบรร่างกายท่อนบน
- แปลงภาพให้อยู่ในระบบสี HSV และเลือกระดับสีที่ต้องการ
- ปรับปรุงภาพไบนารี
- คำนวณหาวัตถุในภาพ

3.2.1 การตรวจจบบรร่างกายท่อนบน (Upper body Detection)

สิ่งแรกที่ต้องทำการตรวจจบบนหลังจากการรับภาพคนมา คือร่างกายท่อนบน เพื่อให้ได้รู้ว่ามีคนอยู่ในภาพแล้ว และเพื่อให้ทำการระบุการถัดไปได้ ถ้าเกิดไม่เจอร่างกายท่อนบนโปรแกรมก็จะไม่ประมวลผลภาพ จะมีแต่รับภาพออกมาแสดงทางหน้าจอคอมพิวเตอร์เท่านั้น

การตรวจจบบรร่างกายท่อนบนทำได้โดยสร้างตัวแปร `cv::CascadeClassifier` `upBody_cascade`; จากนั้นทำการโหลดไฟล์ `xml` (ไฟล์นี้ต้องทำการโหลดเข้ามาเพื่อใช้เป็นฐานในการเปรียบเทียบร่างกายท่อนบน)

`upBody_cascade.load("C:\\opencv\\sources\\data\\haarcascades\\haarcascade_upperbody.xml")` จากนั้นใช้ `upBody_cascade.detectMultiScale()` แล้วนำข้อมูล `output` ใส่ตัวแปร `bodysBuf` ถ้าเกิดว่าสามารถตรวจจบบรร่างกายท่อนบนได้ตัวแปร `bodysBuf` จะมีขนาดที่มากกว่า 0 ซึ่งจะทำให้เกิดการติกรอบที่ร่างกายท่อนบน ดังรูปที่ 3.3



รูปที่ 3.3 ภาพแสดงการตรวจจบบรร่างกายท่อนบน

3.2.2 การแปลงภาพเป็นภาพสี HSV และทำการเลือกระดับสีที่ต้องการ

หลังจากที่เราได้ภาพที่ถูกรวบรวมจากร่างกายท่อนบนเรียบร้อยแล้วนั้น ต่อไปเราจะนำภาพที่ได้มาทำการแปลงให้อยู่ในระบบสี HSV เพื่อนำไปแยกสี เอาเฉพาะสีที่เราต้องการเท่านั้น(สีผิว) เพราะระบบสี HSV สามารถแยกสีได้ง่ายกว่าระบบสี RGB เนื่องจากระบบสี HSV มีการไล่สีตามความเข้มแสง ทำให้ใช้ในสภาพที่แสงแตกต่างกันได้มากขึ้น

วิธีการแปลงภาพสีจาก RGB เป็น HSV ใน opencv จะใช้ คำสั่ง `cvtColor(OriginalFrame, hsvFrame, CV_BGR2HSV);`



รูปที่ 3.4 ภาพที่ได้จากการแปลงเป็นภาพระบบสี HSV

จากรูปที่ 3.4 จะได้ภาพที่อยู่ในระบบสี HSV ซึ่งสามารถปรับค่าต่างได้อย่างที่ต้องการ ซึ่งเราต้องการสีผิว ดังนั้นก็ทำการปรับค่าสีตามสีผิวโดยใช้คำสั่ง "inRange" ตัวอย่างเช่น `"inRange(hsv, cv::Scalar(0, 75, 55), cv::Scalar(60, 256, 256), mask);"` ซึ่งการเลือกสีผิวจากตัวอย่างการใช้คำสั่ง "inRange" นั้น มีปัจจัยที่มีผลต่อรูปผลลัพธ์คือแสง เพราะถ้าเราใช้ค่าเดียวกันในการตรวจจับเฉพาะสีผิว ในสถานที่ที่ต่างกัน รูปผลลัพธ์ที่ได้ก็จะต่างกัน เนื่องจากความเข้มของแสงในแต่ละสถานที่มีความต่างกัน

ผลลัพธ์ที่ได้จากการปรับค่าสี จะได้เป็นภาพไบนารี (ภาพขาวดำ) ดังรูปที่ 3.5 แต่จะเห็นว่าในรูปยังมีวัตถุอื่น ซึ่งไม่ใช่แค่มือ และใบหน้า เท่านั้น ซึ่งวัตถุเหล่านี้ไม่ควรอยู่ในรูปผลลัพธ์ เพราะจะทำให้เกิดความผิดพลาดในการออกคำสั่งในการควบคุมอากาศยานปีกหมุน ดังนั้น เราจึงควรกำจัดวัตถุส่วนเกินเหล่านี้ออกไป



รูปที่ 3.5 ภาพที่ได้จากการเลือกช่วงของสีที่ต้องการ

3.2.3 การปรับปรุงภาพไบนารี

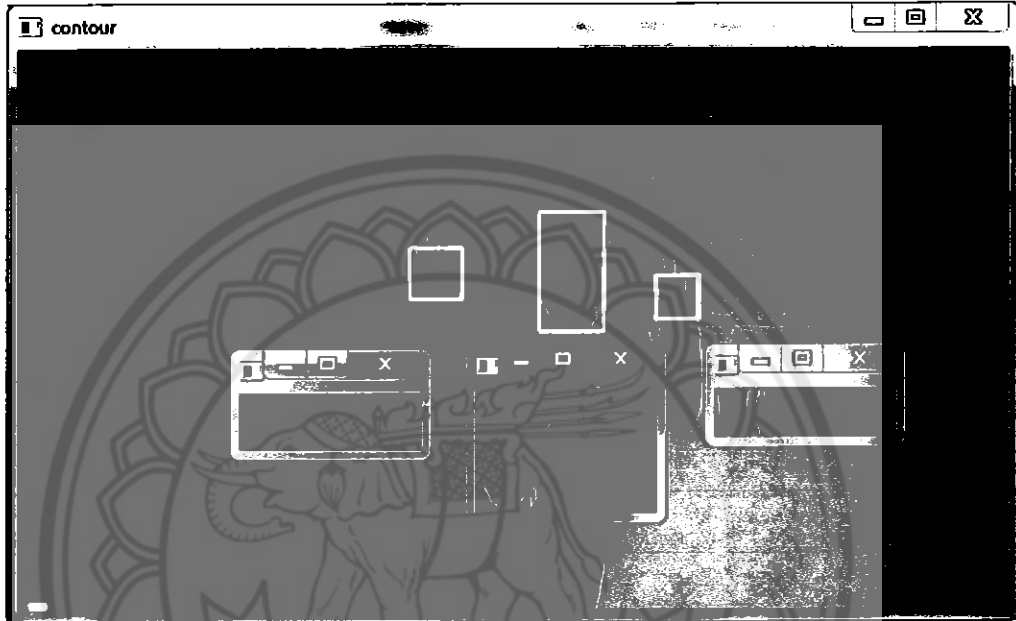
เมื่อได้รูปจากการปรับค่า HSV ซึ่งเป็นรูปที่ยังมีส่วนเกินอยู่นั้น ในขั้นตอนนี้จะทำการกำจัดส่วนเกินเหล่านั้นออก โดยใช้เทคนิค "Morphological" ขั้นแรกใช้ erosion ลบวัตถุที่มีขนาดเล็กออกไป จากนั้นใช้ dilation เพื่อลดช่องว่างของวัตถุ ที่เป็นวัตถุส่วนเดียวกัน และใช้ threshold เพื่อทำการแยกวัตถุออกจากพื้นหลังโดยชัดเจน (รูปที่ 3.6) เพราะในบางครั้งภาพที่ได้จากการทำ morphology วัตถุเดียวกันอาจจะขาดจากกันเพียงเล็กน้อย ซึ่งถ้านำไปประมวลผลในขั้นตอนถัดไป อาจจะทำให้โปรแกรมเข้าใจว่ามีวัตถุหลายชิ้น



รูปที่ 3.6 ภาพที่ได้หลังจากปรับปรุงภาพไบนารี

3.2.4 การหาจำนวนวัตถุในภาพ

ขั้นตอนนี้จะเป็นการหาจำนวนของวัตถุจากภาพที่ได้ในขั้นตอนข้างต้น โดยใช้หลักการคอนทัวร์ (findContours) เพื่อนับและหาวัตถุทั้งหมดในภาพไบนารี และทำการคำนวณหามือและศีรษะ เพื่อให้โปรแกรมรู้ว่าอะไรคือมือ อะไรคือศีรษะ จากค่าการคอนทัวร์ โดยผู้พัฒนาโปรแกรมกำหนดให้ ค่าของการคอนทัวร์ที่ใหญ่ที่สุด คือศีรษะ และค่าที่ต่ำลงมาก็คือมือ เมื่อรู้ว่าอะไรคือมือ อะไรคือศีรษะแล้ว ก็ทำการตีกรอบให้กับมือและศีรษะที่ได้ ดังรูปที่ 3.7 เพื่อที่จะนำตำแหน่งของกรอบที่ได้ ไปใช้วิเคราะห์ภาพในขั้นตอนถัดไป



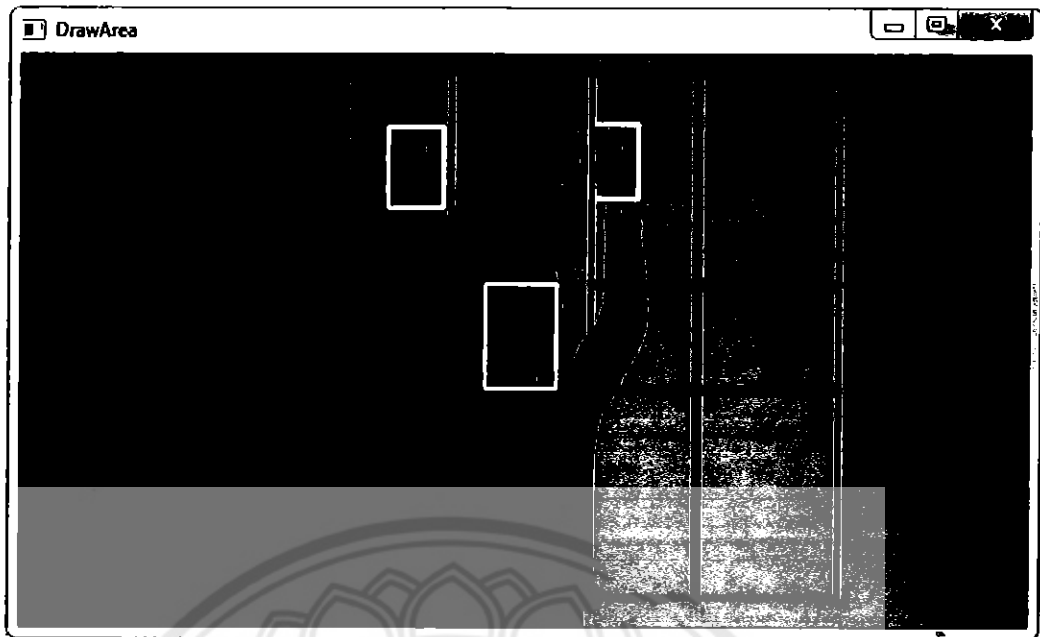
รูปที่ 3.7 ภาพที่ได้จากการตีกรอบสี่เหลี่ยมรอบวัตถุที่เจอและแสดงวัตถุที่เจอ

3.3 การวิเคราะห์ภาพเพื่อให้ได้คำสั่งควบคุมอากาศยานปีกหมุน

หลังจากขั้นตอนการประมวลผลขั้นต้น ขั้นตอนถัดมาจะเป็นขั้นตอนการวิเคราะห์ภาพ โดยจะวิเคราะห์ภาพจากลักษณะเฉพาะของภาพ ในที่นี้คือ ตำแหน่งของศีรษะ และตำแหน่งของมือทั้ง 2 ข้าง ซึ่งมีขั้นตอนการทำงานดังนี้

3.3.1 วาดกรอบพื้นที่

ในการวาดกรอบพื้นที่ทั้ง 2 ข้างนั้น จะวาดโดยอิงจากตำแหน่งและขนาดของศีรษะที่ถูกตีกรอบไว้ โดยขนาดของกรอบที่ได้นั้น จะมีความกว้างเป็นสองเท่าของความกว้างของกรอบศีรษะ และมีความสูงเป็นสามเท่าของขนาดความสูงของกรอบศีรษะ ซึ่งผลลัพธ์จะเป็นดังรูป 3.8 จะปรากฏกรอบสี่เหลี่ยม 2 กรอบ ซึ่งแต่ละกรอบจะถูกแบ่งย่อยเป็นกรอบละ 6 ช่อง โดยช่องต่างๆนี้ จะถูกนำมาใช้เป็นเงื่อนไขในการสร้างคำสั่งการควบคุมอากาศยานปีกหมุน ซึ่งจะกล่าวในขั้นตอนถัดไป



รูปที่ 3.8 ภาพการตีกรอบพื้นที่

3.3.2 การหาคำแหน่งของมือ

จะสังเกตเห็นว่ามือจะถูกตีกรอบไว้ จึงนำกรอบนั้นมาหาจุดกลาง เพื่อนำไปใช้ในการระบุตำแหน่งของมือในกรอบพื้นที่ ซึ่งกรอบพื้นที่จะถูกแบ่งเป็น บนซ้าย บนขวา กลางซ้าย กลางขวา ล่างซ้าย ล่างขวา ซึ่งตำแหน่งต่างๆ นี้ จะถูกนำไปใช้สร้างเป็นเงื่อนไขการควบคุมอากาศยานปีกหมุนต่อไป

3.3.2.1 การประมวลผลหาคำสั่งควบคุมจากตำแหน่งของมือ

คำสั่งควบคุมจะมี ไปทางซ้าย ไปทางขวา บนขึ้น บนลง เดินหน้า ถอยหลัง

- บินอยู่กับที่ ได้จาก มือข้างซ้าย(ของภาพ) อยู่ตำแหน่งกลางขวา และมือข้างขวา(ของภาพ) อยู่ตำแหน่งกลางซ้าย
- บินไปทางซ้าย ได้จาก มือข้างซ้าย(ของภาพ) อยู่ตำแหน่งกลางซ้าย และมือข้างขวา(ของภาพ) อยู่ตำแหน่งกลางซ้าย หรือมือข้างขวาของภาพไม่มีก็ได้
- บินไปทางขวา ได้จาก มือข้างขวา(ของภาพ) อยู่ตำแหน่งกลางขวา และมือข้างซ้าย(ของภาพ) อยู่ตำแหน่งกลางขวา หรือมือข้างซ้ายของภาพไม่มีก็ได้
- บินขึ้น ได้จาก มือข้างซ้าย(ของภาพ) อยู่ตำแหน่งบนขวา และมือข้างขวา(ของภาพ) อยู่ตำแหน่งบนซ้าย
- บินลง ได้จาก มือข้างซ้าย(ของภาพ) อยู่ตำแหน่งล่างขวา และมือข้างขวา(ของภาพ) อยู่ตำแหน่งล่างซ้าย
- บินไปทางหน้า ได้จาก มือข้างซ้าย(ของภาพ) อยู่ตำแหน่งบนซ้าย และมือข้างขวา(ของภาพ) อยู่ตำแหน่งบนขวา
- บินถอยหลัง ได้จาก มือข้างซ้าย(ของภาพ) อยู่ตำแหน่งล่างซ้าย และมือข้างขวา(ของภาพ) อยู่ตำแหน่งล่างขวา



รูปที่ 3.9 ภาพประมวลผลคำสั่ง ให้ไปทางซ้าย



รูปที่ 3.10 ภาพประมวลผลคำสั่ง ให้บินขึ้นข้างบน

จากรูปที่ 3.9 และ 3.10 คือตัวอย่างการควบคุมอากาศยานปีกหมุน เพื่อให้บินทางซ้าย และบินขึ้น ตามลำดับ

3.4 ขั้นตอนการส่งคำสั่งควบคุมให้กับอากาศยานปีกหมุน

คำสั่งที่ใช้ในโปรแกรมจะเป็นคำสั่ง `ardrone.move3D(vx, vy, vz, vr)`; โดยมีการใช้งานดังนี้

- $vx > 0$ คือ บินไปข้างหน้า
- $vx < 0$ คือ บินถอยหลัง
- $vy > 0$ คือ บินไปทางซ้าย
- $vy < 0$ คือ บินไปทางขวา
- $vz > 0$ คือ บินขึ้น
- $vz < 0$ คือ บินลง
- $vr > 0$ คือ หมุนไปทางซ้าย
- $vr < 0$ คือ หมุนไปทางขวา

ตัวอย่างเช่น `ardrone.move3D(1.0, 0.0, 0.0, 0.0)`; คือการส่งคำสั่งเดินทาง เป็นต้น ส่วนคำสั่ง `landing` และ `takeoff` คือ `ardrone.takeoff()`; และ `ardrone.landing()`;

จากขั้นตอนข้างต้น เมื่อเรานำมือไปไว้ที่กรอบพื้นที่แล้ว นั้นหมายความว่า ได้ทำการสร้างคำสั่งควบคุมขึ้นมาแล้ว เช่น อิงจากรูปที่ 3.9 เมื่อมือข้างซ้าย(ของภาพ) อยู่ตำแหน่งบนขวา และมือข้างขวา(ของภาพ) อยู่ตำแหน่งบนซ้าย โปรแกรมจะเข้าเงื่อนไข ดังนี้

```
if (Hleft == "up" && HRight == "up")
{
    cout << "Up" << endl;
    cv::putText(OriginalFrame, "Move Up", cv::Point(50,50), 1, 1,
    cv::Scalar(0, 255, 0), 2);
    vz = 1.0;
}
```

จากโค้ดข้างต้น โปรแกรมตรวจสอบพบเงื่อนไขว่า มีข้างซ้าย(ของภาพ) อยู่ตำแหน่งบนขวา และมือข้างขวา(ของภาพ) อยู่ตำแหน่งบนซ้าย จึงสั่งให้บินขึ้น จากโค้ด `vz = 1.0`;

Overview



รูปที่ 3.11 ภาพการควบคุมท่าอากาศยานปีกหมุนสี่ใบพัดโดยใช้ร่างกาย



บทที่ 4

ผลการทดลอง

4.1 ผลการทดลองการเชื่อมต่อระหว่างคอมพิวเตอร์กับอากาศยานปีกหมุน

ก่อนที่เราจะสามารถควบคุมอากาศยานปีกหมุนให้บินตามที่เรต้องการได้นั้น เราต้องทำการเชื่อมต่อคอมพิวเตอร์กับอากาศยานปีกหมุนให้มองเห็นกันได้ก่อน โดยเชื่อมต่อกันด้วยสัญญาณไวไฟ เพราะถ้าเราไม่เชื่อมต่อคอมพิวเตอร์กับอากาศยานปีกหมุน จะทำให้เราไม่สามารถรันโปรแกรมเพื่อควบคุมอากาศยานปีกหมุนได้

4.1.1 เมื่อไม่ได้เชื่อมต่อสัญญาณไฟระหว่างคอมพิวเตอร์กับอากาศยานปีกหมุน

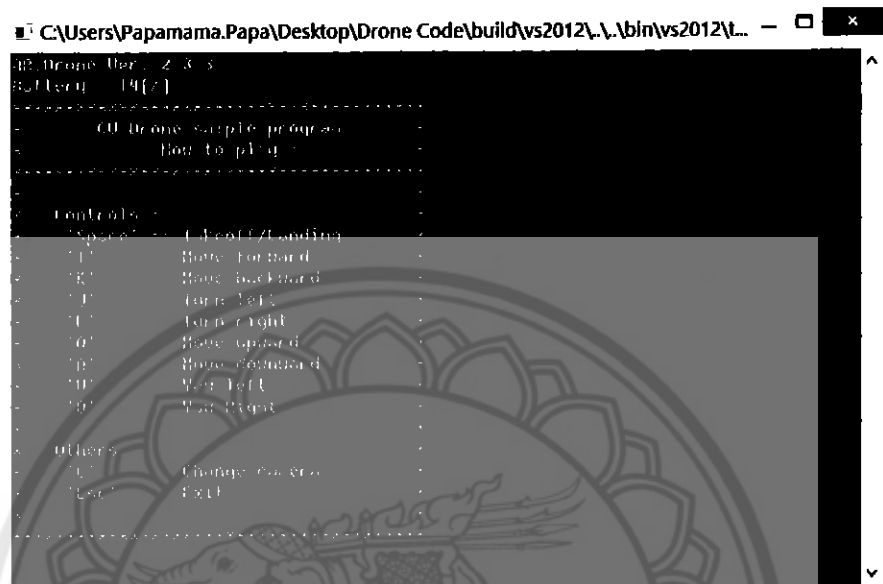
ถ้าเราไม่เชื่อมต่อสัญญาณไฟระหว่างคอมพิวเตอร์กับอากาศยานปีกหมุนก่อน จะทำให้ผลของการรันโปรแกรมแสดงออกมาเป็น “Failed to initialize” ดังรูป 4.1 ซึ่งจะไม่ได้สามารถดำเนินการในขั้นตอนต่อไปได้ ดังนั้น เราต้องทำการเชื่อมต่อสัญญาณไฟระหว่างคอมพิวเตอร์กับอากาศยานปีกหมุนให้เรียบร้อยก่อน



รูปที่ 4.1 ผลของการรันโปรแกรมเมื่อไม่ได้เชื่อมต่อคอมพิวเตอร์กับอากาศยานปีกหมุน

4.1.2 เมื่อเชื่อมต่อสัญญาณไฟโระหว่างคอมพิวเตอร์กับอากาศยานปีกหมุนแล้ว

เมื่อเราทำการเชื่อมต่อสัญญาณไฟโระหว่างคอมพิวเตอร์กับอากาศยานปีกหมุนก่อนรันโปรแกรม ผลที่ได้คือโปรแกรมจะรันเข้าสู่หน้าหลักซึ่งจะแสดงรายละเอียดต่างๆ ดังรูปที่ 4.2 และหน้าต่างซึ่งแสดงภาพ ที่ได้จากกล้องของอากาศยานปีกหมุน ดังรูปที่ 4.3

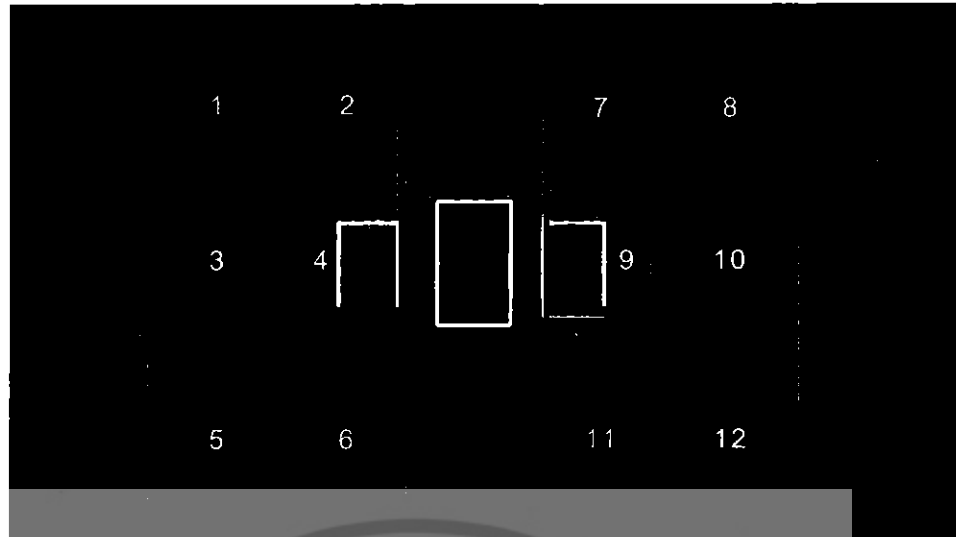


รูปที่ 4.2 ผลของการรันโปรแกรมเมื่อเชื่อมต่อคอมพิวเตอร์กับอากาศยานปีกหมุนแล้ว

จากรูปที่ 4.2 โปรแกรมจะรันเข้าสู่หน้าจอหลักของ ซึ่งหน้าต่างนี้จะแสดงเวอร์ชันของอากาศยานปีกหมุน ปริมาณของแบตเตอรี่ที่เหลืออยู่ และวิธีการควบคุมอากาศยานปีกหมุนด้วยคีย์บอร์ด ดังตารางที่ 4.1

ตารางที่ 4.1 แสดงวิธีการควบคุมอากาศยานปีกหมุนด้วยคีย์บอร์ด

ปุ่มที่กด	คำสั่ง
"Spacebar"	Take off/Landing
"I"	บินไปด้านหน้า
"K"	บินไปด้านหลัง
"J"	บินไปด้านซ้าย
"L"	บินไปด้านขวา
"Q"	บินขึ้น
"A"	บินลง
"U"	หมุนซ้าย
"O"	หมุนขวา



รูปที่ 4.3 แสดงภาพที่ได้จากกล้องของอากาศยานปีกหมุน

จากรูปที่ 4.3 เมื่ออากาศยานปีกหมุนเริ่มบินขึ้น แล้วกล้องจากอากาศยานปีกหมุนพบร่างกายท่อนบน จะปรากฏกรอบสี่เหลี่ยม(สีน้ำเงิน) 2 กรอบ ที่ถูกแบ่งย่อยออกเป็นกรอบละ 6 ช่อง ซึ่งช่องย่อยทั้ง 6 นี้ เราจะนำมาใช้เป็นเงื่อนไขในการสร้างคำสั่งการควบคุมอากาศยานปีกหมุนให้บินในลักษณะต่างๆ ซึ่งจะถูกกำหนดตามการเคลื่อนที่ของฝ่ามือ เช่น ถ้าเราขยับมือขวา (ที่ถูกกำหนดให้เป็น Left) ไปที่ช่องหมายเลข 3 และขยับมือซ้าย (ที่ถูกกำหนดให้เป็น Right) ไปที่ช่องหมายเลข 9 ผลลัพธ์ที่ได้คือ อากาศยานปีกหมุนจะบินไปด้านซ้าย เป็นต้น ซึ่งวิธีการควบคุมทั้งหมดนั้น จะอธิบายในหัวข้อถัดไป

4.2 ผลการทดลองการควบคุมอากาศยานปีกหมุนด้วยการเคลื่อนไหวของร่างกาย

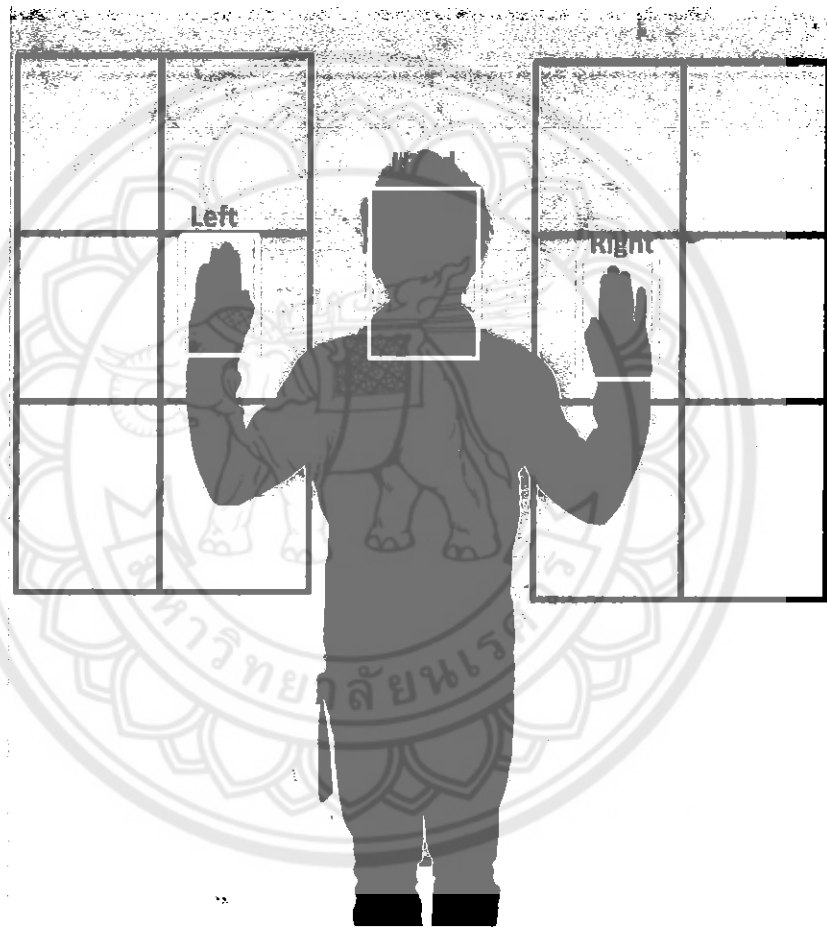
ในการควบคุมอากาศยานปีกหมุนด้วยท่าทางการเคลื่อนไหวของร่างกายนั้น ประกอบไปด้วยท่าทางทั้งหมด 7 ท่าทาง คือ บินอยู่กับที่ บินขึ้น บินลง บินไปด้านซ้าย บินไปด้านขวา บินไปด้านหน้า และบินไปด้านหลัง ซึ่งการที่ผู้ควบคุมจะสามารถควบคุมอากาศยานปีกหมุนให้ได้ผลดีที่สุด จำเป็นต้องอยู่ภายใต้ปัจจัยหลายๆอย่าง ซึ่งมีผลต่อการควบคุม เช่น การแต่งกายของผู้ควบคุม สภาพลม แสง ระยะห่างระหว่างอากาศยานปีกหมุนกับผู้ควบคุม พื้นหลัง เป็นต้น

4.2.1 วิธีการควบคุมอากาศยานปีกหมุนด้วยการเคลื่อนไหวของร่างกาย

เมื่ออากาศยานปีกหมุนเริ่มบินขึ้น กล้องที่ติดอยู่กับอากาศยานปีกหมุนจะเริ่มรับภาพผู้ควบคุมเพื่อตรวจจ็ับร่างกายที่นอนบน และเมื่อเจอร่างกายที่นอนบน หัวและมือทั้ง 2 ข้าง จะถูกตีกรอบ (สีขาว) แล้วนำไปสร้างกรอบสี่เหลี่ยมสีน้ำเงิน ดังรูปที่ 4.4 เพื่อใช้เป็นตัวกำหนดเงื่อนไขในการควบคุม ซึ่งเราสามารถควบคุมทิศทางการบินของอากาศยานปีกหมุนได้ที่หมด 7 ลักษณะ ดังนี้

4.2.1.1 การบินอยู่กับที่ (Hovering)

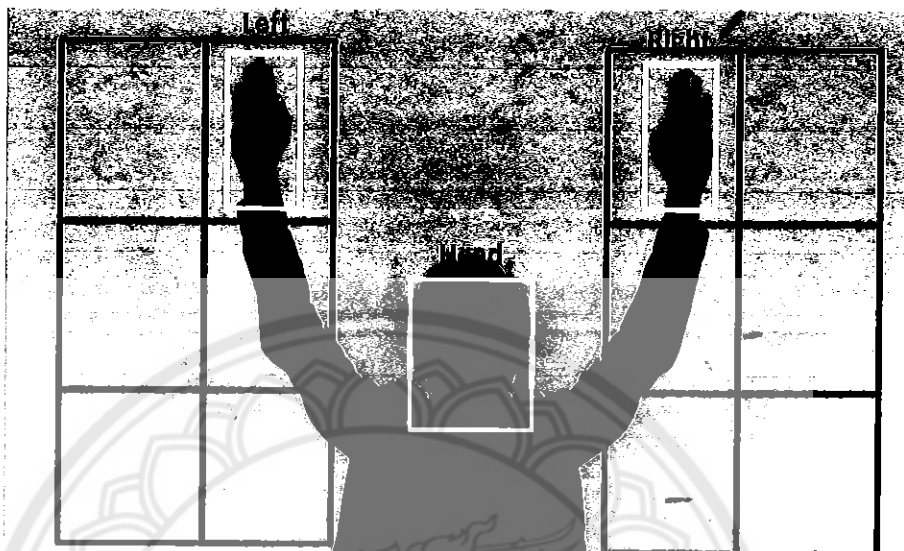
ผู้ควบคุมต้องนำมือทั้ง 2 ข้าง วางไว้ในช่อง ดังรูปที่ 4.4 เพื่อสั่งการให้อากาศยานปีกหมุนบินอยู่กับที่



รูปที่ 4.4 แสดงวิธีการควบคุมอากาศยานปีกหมุนให้บินอยู่กับที่

4.2.1.2 การบินขึ้น

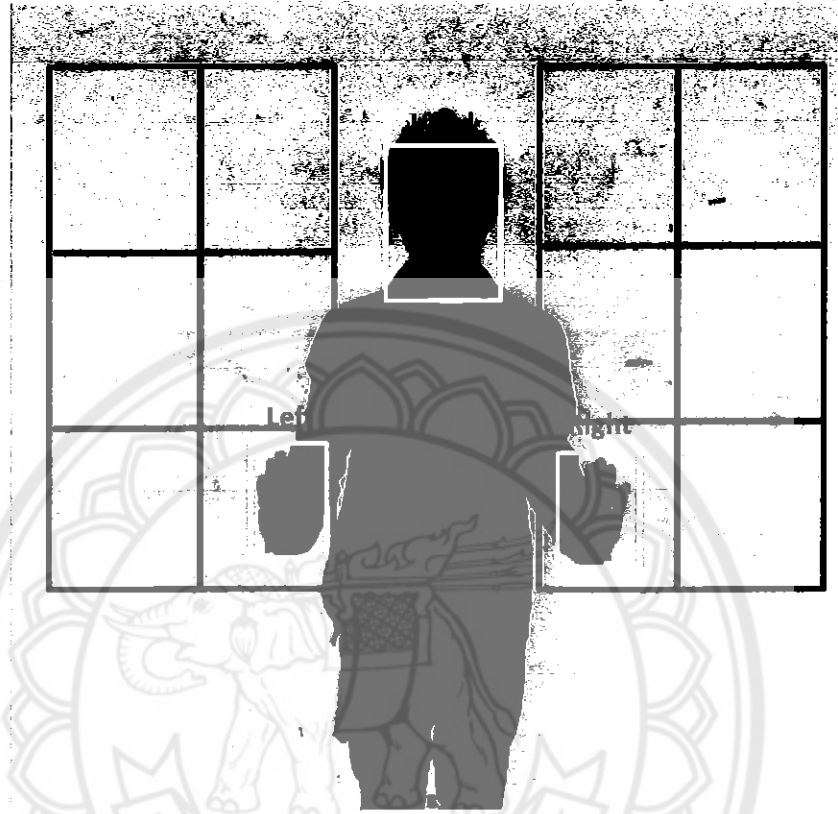
ผู้ควบคุมต้องนำมือทั้ง 2 ช้าง วางไว้ในช่อง ดังรูปที่ 4.5 เพื่อสั่งการให้อากาศยานปีกหมุนบินขึ้น



รูปที่ 4.5 แสดงวิธีการควบคุมอากาศยานปีกหมุนให้บินขึ้น

4.2.1.3 การบินลง

ผู้ควบคุมต้องนำมือทั้ง 2 ข้าง วางไว้ในช่อง ดังรูปที่ 4.6 เพื่อสั่งการให้อากาศยานปีกหมุนบินลง



รูปที่ 4.6 แสดงวิธีการควบคุมอากาศยานปีกหมุนให้บินลง

4.2.1.4 การบินไปด้านซ้าย

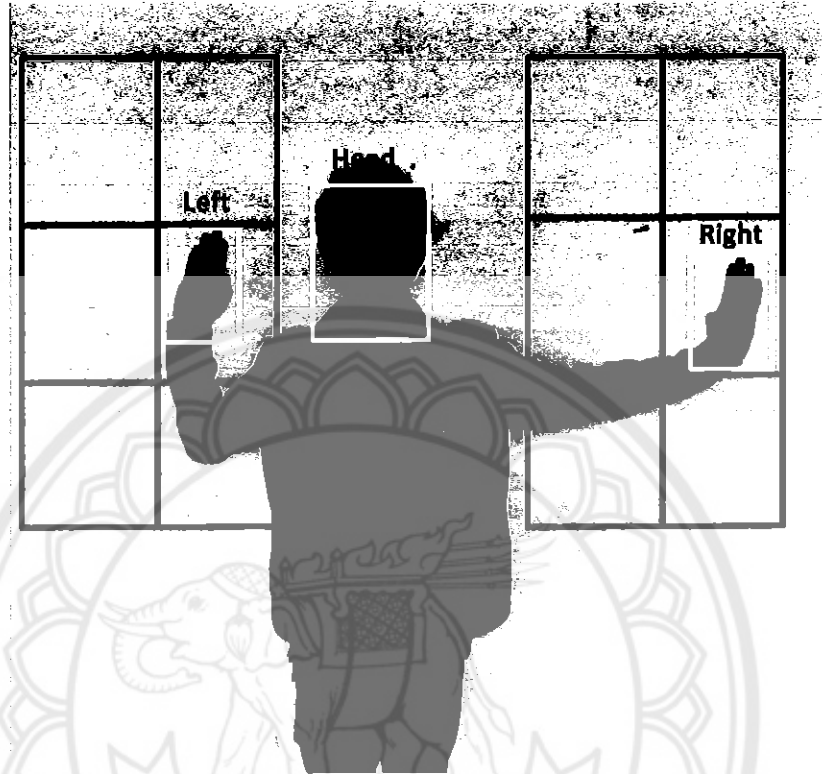
ผู้ควบคุมต้องนำมือทั้ง 2 ข้าง วางไว้ในช่อง ดังรูปที่ 4.7 เพื่อสั่งการให้อากาศยานปีกหมุนบินไปด้านซ้าย



รูปที่ 4.7 แสดงวิธีการควบคุมอากาศยานปีกหมุนให้บินไปด้านซ้าย

4.2.1.5 การบินไปด้านขวา

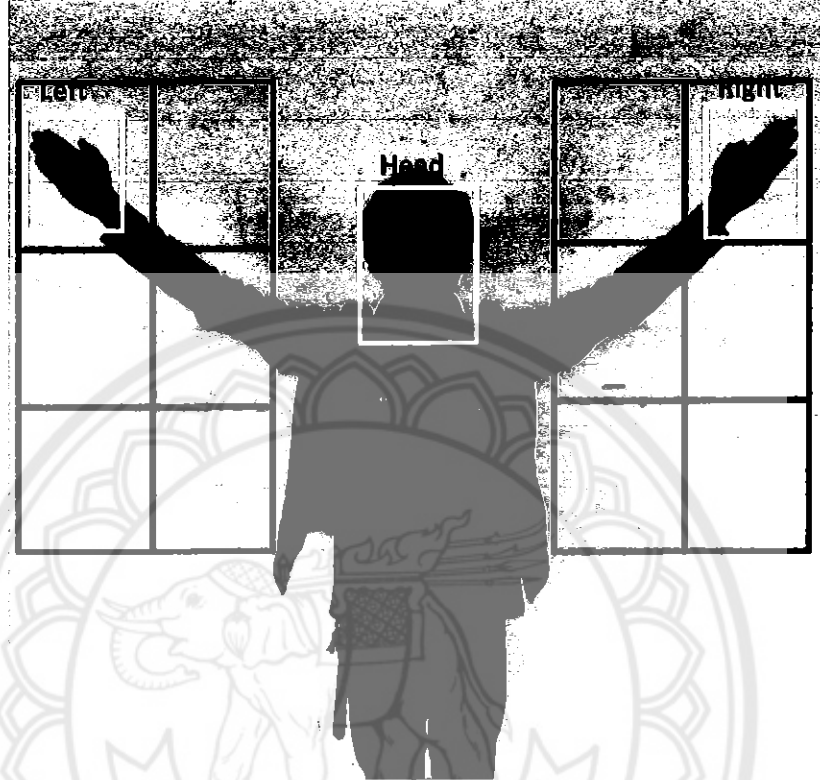
ผู้ควบคุมต้องนำมือทั้ง 2 ข้าง วางไว้ในช่อง ดังรูปที่ 4.8 เพื่อสั่งการให้อากาศยานปีกหมุนบินไปด้านขวา



รูปที่ 4.8 แสดงวิธีการควบคุมอากาศยานปีกหมุนให้บินไปด้านขวา

4.2.1.6 การบินไปด้านหน้า

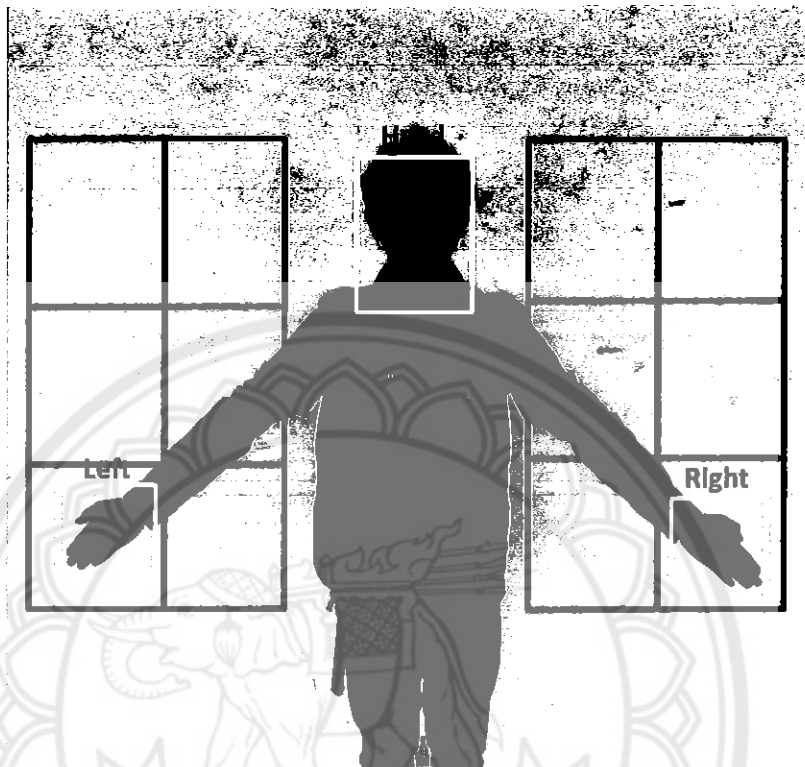
ผู้ควบคุมต้องนำมือทั้ง 2 ข้าง วางไว้ในช่อง ดังรูปที่ 4.9 เพื่อสั่งการให้อากาศยานปีกหมุนบินไปด้านหน้า



รูปที่ 4.9 แสดงวิธีการควบคุมอากาศยานปีกหมุนให้บินไปด้านหน้า

4.2.1.7 การบินไปด้านหลัง

ผู้ควบคุมต้องนำมือทั้ง 2 ข้าง วางไว้ในช่อง ดังรูปที่ 4.10 เพื่อสั่งการให้อากาศยานปีกหมุนบินไปด้านหลัง



รูปที่ 4.10 แสดงวิธีการควบคุมอากาศยานปีกหมุนให้บินไปด้านหลัง

4.2.2 ผลของค่าสถานะ เมื่อเริ่มต้นรันโปรแกรม

เมื่อเริ่มต้นรันโปรแกรม จะปรากฏหน้าต่างของภาพที่ได้จากอากาศยานปีกหมุนและหน้าต่างคอนโซลที่แสดงสถานะของการตรวจจับร่างกายท่อนบน ใบหน้าและฝ่ามือ ดังรูปที่ 4.11 และ 4.12 ซึ่งค่าการแสดงผลสถานะจะถูกแบ่งการแสดงผลออกเป็น 2 ลักษณะ คือ เมื่อกำลังจากอากาศยานปีกหมุนพบร่างกายท่อนบนและเมื่อกำลังจากอากาศยานปีกหมุนยังไม่พบร่างกายท่อนบน

4.2.2.1 เมื่อกำลังจากอากาศยานปีกหมุนยังไม่พบร่างกายท่อนบน

เมื่อกำลังจากอากาศยานปีกหมุนพบร่างกายท่อนบน จะปรากฏหน้าต่างเพียง 2 หน้าต่างเท่านั้น ดังรูปที่ 4.11 และ 4.12



รูปที่ 4.11 แสดงภาพที่ได้จากกล้องของอากาศยานปีกหมุนเมื่อเริ่มรันโปรแกรม

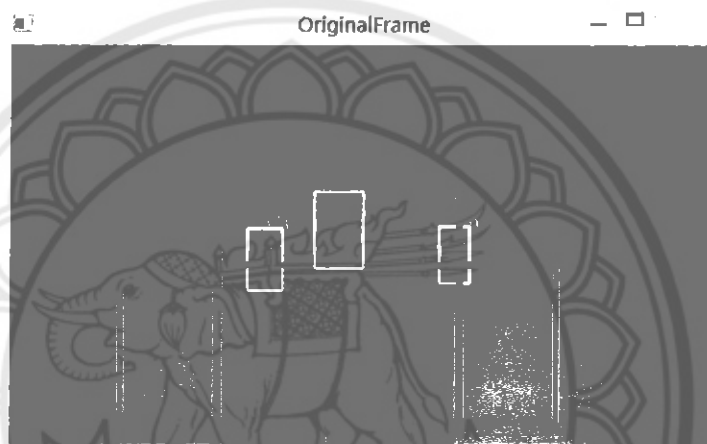


รูปที่ 4.12 แสดงค่าสถานะเมื่อกำลังจากอากาศยานปีกหมุนยังไม่พบร่างกายท่อนบน

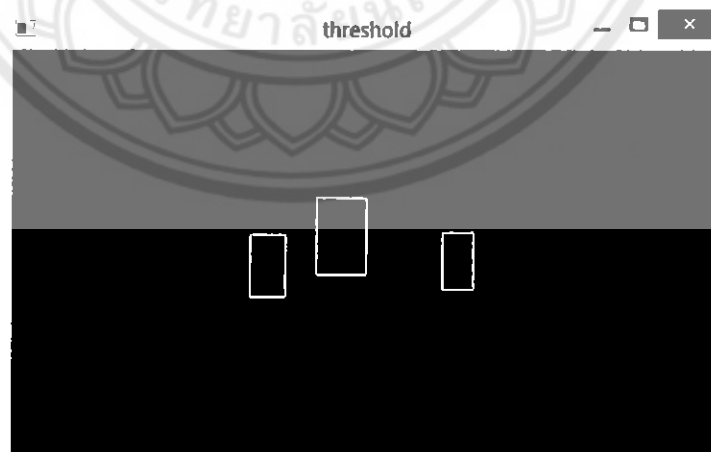
จากรูปที่ 4.12 ผลของค่าสถานะคือ “body 0” หมายความว่า กล้องจากอากาศยานปีกหมุนยังไม่พบร่างกายท่อนบน ซึ่งดูได้จากรูปที่ 4.11 จะสังเกตว่า เห็นเฉพาะร่างกายท่อนล่างของผู้ควบคุมเท่านั้น ซึ่ง ค่าสถานะ “body 0” จะถูกแสดงค่าวนลูไปเรื่อยๆ จนกว่าจะพบร่างกายท่อนบน

4.2.2.1 เมื่อกล้องจากอากาศยานปีกหมุนพบร่างกายท่อนบน

เมื่อกล้องจากอากาศยานปีกหมุนพบร่างกายท่อนบน จะมีหน้าต่างเพิ่มมาอีก 1 หน้าต่าง ดังรูปที่ 4.14 ซึ่งเป็นหน้าต่างของการแสดงภาพที่ผ่านกระบวนการเทรชโฮลด์ และคอนทัวร์ ซึ่งผลของกระบวนการดังกล่าวจะทำให้เราได้ภาพไบนารีที่ทำให้ระบบสามารถบอกได้ว่าอะไรคือใบหน้า อะไรคือฝ่ามือ จากค่าสถานะที่บอกในรูปที่ 4.15



รูปที่ 4.13 แสดงรูปเมื่อกล้องจากอากาศยานปีกหมุนพบร่างกายท่อนบน



รูปที่ 4.14 แสดงรูปภาพต้นฉบับที่ผ่านกระบวนการเทรชโฮลด์ และคอนทัวร์ จากภาพต้นฉบับ



รูปที่ 4.16 แสดงวิธีการแต่งตัวที่เหมาะสมต่อการควบคุมอากาศยานปีกหมุน

4.3.2 สภาพลม

เนื่องจากอากาศยานปีกหมุนมีความอ่อนไหวต่อลมเป็นอย่างมาก ดังนั้น ผู้ควบคุมควรทดสอบในห้องปิดที่ไม่มีลมไหลผ่าน เพราะถ้าเป็นสถานที่ที่มีลมแรง อากาศยานปีกหมุนถูกพัดไปตามลม จนผู้ควบคุมไม่สามารถควบคุมอากาศยานปีกหมุนได้

4.3.3 สภาพแสง

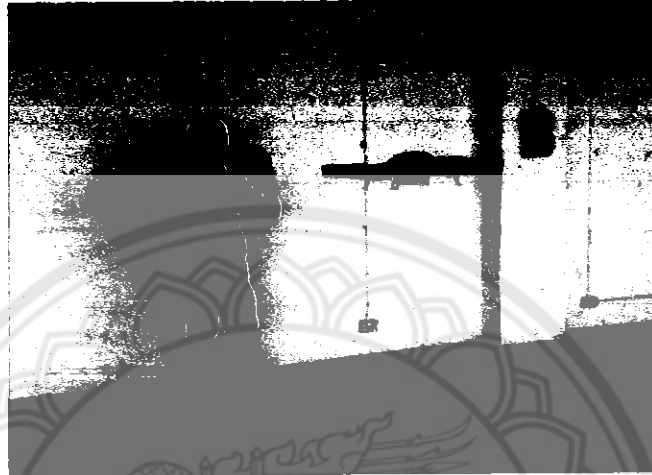
เนื่องจากอากาศยานปีกหมุนต้องรับภาพของฝ่าและใบหน้า เพื่อนำไปทำการตรวจจับสีผิว ดังนั้น ผู้ควบคุมต้องทดสอบในสถานที่ ที่มีปริมาณแสงที่เพียงพอ

4.3.4 พื้นหลัง

เนื่องจากอากาศยานปีกหมุนต้องตรวจจับสีผิว (ใบหน้าและฝ่ามือ) เพื่อไปทำการประมวลผลภาพ ดังนั้น เราต้องเลือกสถานที่ในการทดสอบที่มีพื้นหลังเป็นสีโทนเดียว ดังรูปที่ 4.4 (ที่ไม่ตรงกับสีผิว) เพราะถ้าเราทดสอบในสถานที่ที่มีพื้นหลังหลายสี เช่น กลางป่า หรือในห้องที่มีคนจำนวนมาก จะทำให้กล้องจากอากาศยานปีกหมุน ตรวจพบสิ่งนี้อาจมีสีคล้ายคลึงกับสีของผิวหนังของผู้ควบคุม ส่งผลให้อากาศยานปีกหมุนเข้าใจว่าสิ่งนั้นเป็นสีผิวของผู้ควบคุม ซึ่งจะก่อให้เกิดความผิดพลาดในการควบคุม

4.3.5 ระยะห่างระหว่างผู้ควบคุมกับอากาศยานปีกหมุน

เนื่องจากอากาศยานปีกหมุนต้องรับภาพจากร่างกายของผู้ควบคุม ซึ่งระยะห่างระหว่างผู้ควบคุมกับอากาศยานปีกหมุนควรอยู่ที่ประมาณ 2 เมตร ดังรูปที่ 4.18 เพราะถ้าเริ่มไกลมากขึ้น จะทำให้อากาศยานปีกไม่สามารถตรวจจับใบหน้า และฝ่ามือของผู้ควบคุมได้ และถ้าใกล้เกินไปก็จะเป็นอันตรายต่อผู้ควบคุมได้



รูปที่ 4.17 แสดงระยะห่างระหว่างอากาศยานปีกหมุนกับผู้ควบคุมที่เหมาะสม

4.4 ผลการทดลองเมื่อควบคุมอากาศยานปีกหมุนด้วยระยะห่างที่แตกต่างกัน

ขั้นตอนนี้เป็นผลการทดสอบระบบการควบคุมระหว่างอากาศยานปีกหมุนกับผู้ควบคุม โดยควบคุมทั้งหมด 6 ท่า ท่าละ 5 ครั้ง ด้วยระยะห่างตั้งแต่ 0.5 เมตร ถึง 3.5 เมตร โดยทดสอบระบบในห้องปิด ไม่มีลมไหลผ่าน ผนังห้องทดลองเป็นสีขาว โดยที่ผู้ควบคุมสวมเสื้อแขนยาวและกางเกงขาสั้น

ตารางที่ 4.2 แสดงผลของการควบคุมอากาศยานปีกหมุนด้วยระยะห่างที่แตกต่างกัน

ระยะ (เมตร)	ท่าทางการเคลื่อนไหวของอากาศยานปีกหมุน						รวม (30 ครั้ง)	เปอร์เซ็นต์ การควบคุมสำเร็จ
	ซ้าย (5 ครั้ง)	ขวา (5 ครั้ง)	บน (5 ครั้ง)	ล่าง (5 ครั้ง)	เดินหน้า (5 ครั้ง)	ถอยหลัง (5 ครั้ง)		
0.5	2	1	0	0	0	0	3	10%
1.0	5	4	3	3	3	4	22	73%
1.5	5	5	4	5	5	5	29	97%
2.0	4	5	5	5	5	4	28	93%
2.5	5	4	5	5	5	5	29	97%
3.0	4	5	5	5	4	5	28	93%
3.5	3	3	4	3	2	2	17	57%

จากตารางที่ 4.2 พบว่า ที่ระยะห่าง (ระหว่างอากาศยานปีกหมุนกับผู้ควบคุม) ตั้งแต่ 1.5 เมตร ถึง 3 เมตร มีร้อยละของความถูกต้องในการควบคุมอากาศยานปีกหมุนดีที่สุดใน 95 % โดยเฉลี่ย และที่ระยะห่าง 0.5 เมตร ร้อยละของความถูกต้องในการควบคุมอากาศยานปีกหมุนต่ำที่สุดใน 10 %

4.5 สรุปผลการทดลองในส่วนของปัจจัยที่มีผลต่อการควบคุมอากาศยานปีกหมุน

จากการทดลอง ในหัวข้อ “การควบคุมอากาศยานปีกหมุนแบบสี่ใบพัดด้วยท่าทางการเคลื่อนไหวของร่างกาย” เพื่อให้ผู้ควบคุมสามารถควบคุมอากาศยานปีกหมุนได้อย่างมีประสิทธิภาพมากที่สุด ผู้ควบคุมควรแต่งกายด้วยเสื้อแขนยาว กางเกงขายาว ทดสอบในห้องที่ไม่มีลมพัดผ่าน มีแสงที่เพียงพอ และมีพื้นหลังเป็นสีโทนเดียวซึ่งต้องไม่เป็นโทนเดียวสีผิว และระยะห่างระหว่างระหว่างผู้ควบคุมกับอากาศยานปีกหมุนควรอยู่ที่ประมาณ 2 เมตร



บทที่ 5

สรุปผลการดำเนินงานโครงการ

5.1 สรุปผลการทดลองของโครงการ

โครงการนี้ได้นำเทคโนโลยีการเชื่อมต่อกับผู้ใช้อย่างเป็นธรรมชาติ (Natural User Interface) มาประยุกต์ใช้กับการประมวลผลภาพ (image processing) เพื่อสร้างโปรแกรมการควบคุมอากาศยานปีกหมุนโดยใช้ท่าทางการเคลื่อนไหวของร่างกาย รวมทั้งเพื่อเป็นการเพิ่มทางเลือกสำหรับผู้สนใจ ที่จะสามารถเปลี่ยนจากการควบคุมอากาศยานปีกหมุนแบบเดิมๆ มาเป็นการควบคุมโดยใช้ท่าทางการเคลื่อนไหวของร่างกาย

โดยขั้นตอนการดำเนินงานจะเริ่มจากกล้องที่ติดอยู่กับอากาศยานปีกหมุนจะทำหน้าที่รับภาพท่าทางการเคลื่อนไหวของร่างกาย และส่งภาพให้กับคอมพิวเตอร์ จากนั้นคอมพิวเตอร์จะทำการประมวลผลภาพ (Image processing) โดยใช้เทคนิค Haar Cascade เพื่อตรวจจับร่างกายท่อนบน (Upperbody) จากนั้นทำการกำจัดส่วนเกินในภาพที่เราไม่ต้องการออก (Morphology Fundamentals: Dilation and Erosion) เพราะเราต้องการเฉพาะรูปใบหน้า และรูปมือทั้ง 2 ข้างเท่านั้น และทำการแปลงภาพเป็นภาพไบนารี (Thresholding) แล้วหาขอบของภาพจากกระบวนการข้างต้น (Find contours) เพื่อจะใช้ในการคำนวณหาตำแหน่งของหน้าและมือ จากนั้นก็จะกำหนดคำสั่งการเคลื่อนไหวจากตำแหน่งของหน้าและมือ แล้วส่งคำสั่งควบคุมไปยังอากาศยานปีกหมุนเพื่อให้อากาศยานเคลื่อนที่ตามคำสั่งต่างๆ ผ่านระบบ WIFI โดยระบบสามารถควบคุมอากาศยานปีกหมุนได้ 7 ลักษณะ คือ บินอยู่กับที่ บินขึ้น บินลง บินซ้าย บินขวา บินไปด้านหน้า และบินไปด้านหลัง

จากผลการทดลองการบินควบคุมการบินของอากาศยานปีกหมุนด้วยท่าทางการเคลื่อนไหวของร่างกาย พบว่าเราสามารถนำเทคโนโลยีการเชื่อมต่อกับผู้ใช้อย่างเป็นธรรมชาติและหลักการประมวลผลภาพมาประยุกต์ใช้ในการพัฒนาต้นแบบของระบบการควบคุมอากาศยานปีกหมุนด้วยท่าทางการเคลื่อนไหวของร่างกาย ซึ่งทำให้เกิดความเป็นธรรมชาติในการควบคุมมากขึ้น อีกทั้งยังเป็นการสร้างประสบการณ์ใหม่ๆ ของการควบคุมอากาศยานปีกหมุนแก่ผู้สนใจ และเป็นอีกหนึ่งทางเลือกแก่ผู้ที่มีความบกพร่องทางการใช้นิ้ว ซึ่งไม่สามารถใช้นิ้วในการกดรีโมทคอนโทรลได้ และท้ายที่สุดในอนาคตอาจจะนำแนวคิดของโครงการนี้ไปต่อยอด เพื่อสร้างเป็นอุปกรณ์ที่ใช้อำนวยความสะดวกสบายในชีวิตประจำวันของมนุษย์ เช่น หุ่นยนต์อ่อนกประสงค์ภายในบ้าน ที่สามารถสั่งการด้วยเสียงหรือท่าทางการเคลื่อนไหวของร่างกาย เป็นต้น

5.2 ปัญหาและอุปสรรคของโครงการ

ในระหว่างการดำเนินโครงการมักมีปัจจัยที่ส่งผลให้เกิดปัญหาด้านต่างๆระหว่างการทำงาน ซึ่งปัญหาต่างๆเหล่านี้ส่งผลกระทบต่อการทำงานเป็นอย่างมาก จนบางครั้งทำให้โปรแกรมไม่สามารถทำการประมวลผลได้อย่างมีประสิทธิภาพและถูกต้องตามที่คุณดำเนินโครงการได้คาดหวังไว้ ดังนั้น ผู้ดำเนินโครงการจำเป็นต้องหาแนวทางการแก้ไขกับปัญหาต่างๆ เพื่อให้การทำโครงการสำเร็จลุล่วงไปได้ด้วยดี โดยปัญหาที่พบและแนวทางการแก้ไข มีดังต่อไปนี้

ตารางที่ 5.1 แสดงปัญหาและอุปสรรคและแนวทางการแก้ไข

ปัญหาและอุปสรรคของโครงการ	แนวทางการแก้ไข
1. การเชื่อมต่ออากาศยานปีกหมุนกับคอมพิวเตอร์ จำเป็นต้องใช้ไลบรารีเฉพาะ ซึ่งผู้ดำเนินโครงการยังไม่มีความรู้ในเรื่องนี้มากนัก	1. ศึกษาไลบรารีของอากาศยานปีกหมุนให้เข้าใจ
2. ขาดความรู้ในเรื่องของไลบรารีโอเพ่นซีวี เนื่องจากไม่มีในหลักสูตรการเรียน	2. ใช้เวลาศึกษาไลบรารีโอเพ่นซีวี ให้มากขึ้น
3. ในช่วงของการทดลองการควบคุมอากาศยานปีกหมุน ไม่สามารถควบคุมในที่แสงน้อยได้เลย เพราะจะทำให้สีผิวเปลี่ยนไปจากเดิม จนทำให้โปรแกรมไม่สามารถตรวจจับสีผิวได้	3. ทดลองในเวลากลางวันเท่านั้น และต้องเป็นห้องที่มีแสงเข้าอย่างพอเหมาะ
4. ห้องทดลองที่แรกไม่สามารถทดลองได้เนื่องจากมีลมแรงเกินไป และผนังห้องไม่เป็นสีโทนเดียว	4. หาห้องที่มีลักษณะดังกล่าว และผลสุดท้ายก็ได้ห้องที่อยู่ชั้น 7 ของตึก EE
5. การคิดรูปแบบของการควบคุมอากาศยานปีกหมุนให้ดูเป็นธรรมชาติและเหมาะสมที่สุด	6. ประชุมกับสมาชิกในกลุ่ม เพื่อหาวิธีที่เหมาะสมที่สุด

5.3 ข้อเสนอแนะและแนวทางการพัฒนา

1. พัฒนาความสามารถในการตรวจจับสีผิวให้ดียิ่งขึ้น เนื่องจากบางครั้งแสงเปลี่ยนเพียงเล็กน้อย โปรแกรมก็ไม่สามารถตรวจจับสีผิวได้เลย
2. เปลี่ยนการรับภาพท่าทางการเคลื่อนไหวของร่างกาย จากกล้องที่ติดอยู่กับอากาศยานปีกหมุนมาเป็นกล้องที่ตั้งอยู่กับที่ เนื่องจากการควบคุมเก้าอี้ควบคุมต้องเดินตามกล้องที่ติดอยู่กับอากาศยานปีกหมุนตลอด เพราะเมื่อเกิดการเคลื่อนที่ของอากาศยานปีกหมุน กล้องที่ติดอยู่กับอากาศยานปีกหมุนก็จะเคลื่อนที่ตาม จนทำให้ในบางครั้งผู้ควบคุมอาจจะหลุดออกจากเฟรมของกล้องจากอากาศยานปีกหมุนได้ แต่ถ้าเราใช้กล้องที่ตั้งอยู่กับที่ ผู้ควบคุมจะไม่หลุดออกจากเฟรมของกล้อง และสามารถควบคุมอากาศยานปีกหมุนจากที่ใดก็ได้ (ภายในระยะ 50 เมตร)
3. หาวิธีการประมวลผลภาพที่ดีกว่าที่ใช้อยู่ เพื่อให้ระบบทำงานได้เร็วขึ้น
4. เพิ่มการบังคับทางตั้งให้กับอากาศยานปีกหมุน
5. พัฒนาระบบการเคลื่อนที่ของอากาศยานปีกหมุนให้เคลื่อนที่ไปที่ละ 1 หน่วยเวลา ต่อการสั่งการในแต่ละครั้ง เพราะจะทำให้เกิดความปลอดภัย และควบคุมได้ง่ายยิ่งขึ้น
6. นำหลักการที่ใช้ในโครงการไปสร้างเป็นอุปกรณ์ที่สามารถสร้างความสะดวกสบายในชีวิตประจำวันของมนุษย์ เช่น หุ่นยนต์เอนกประสงค์ภายในบ้านที่สามารถสั่งงานด้วยเสียงหรือท่าทางการเคลื่อนไหวของร่างกาย เป็นต้น



เอกสารอ้างอิง

- [1] “ข้อมูลทั่วไปของ Quadcopter รุ่น AR.Drone 2.0” [online]. Available :
<http://on.fb.me/1brd0bm>
- [2] “หลักการการบินของ Quadcopter” [online]. Available :
<http://www.ayarafun.com/2010/04/what-is-quadrotor>
- [3] “ระบบสี่” [online]. Available :
<http://www.ecpe.nu.ac.th/panomkhawn/imagepro/pdf/ch06.pdf>
- [4] นายธนากร ทิอ้าย และ นางสาวกนิษฐา เบญญาไพจิตร. (2556). “การตรวจหาดอกไม้โดยใช้การประมวลผลภาพ” ปริญญาโท วศ.บ., มหาวิทยาลัยนเรศวร, พิษณุโลก
- [5] “การตรวจจับใบหน้าด้วย OpenCV” [online]. Available :
<http://bit.ly/1JL8Bio>
- [6] “การประมวลผลภาพกับรูปร่างและโครงสร้างของภาพ (Morphological Image Processing)” [online]. Available : <http://bit.ly/1GKwDEd>
- [7] “การแปลงภาพเป็นขาวดำ (Threshold)” [online]. Available :
http://web.eng.nu.ac.th/eng2012/work/file2-2014_03_21_11_05_57.pdf
- [8] “การหาขอบภาพ (Edge detection)” [online]. Available :
<http://www.cp.su.ac.th/~prawim/doc/Jaravee%20Proposal/chapter3.doc>. 2556
- [9] “Human Pose Estimation From Monocular Image Captures”
ที่มา : Huei-Yung Lin, Ting-Wen Chen, Chih-Chang Chen, Chia-Hao Hsieh and Wen-Nung Lie, Multimedia and Expo, IEEE International Conference on, 2009.
- [10] “Automatic Registration of Virtual Objects onto Human Image Sequences”
ที่มา : Hoshino, J., Saito, H. Yamamoto, M., “Automatic Registration of Virtual Objects into Human Image Sequences”, Pattern Recognition, Proceedings 15th International Conference on, 2000.
- [11] “A novel fitting algorithm using the ICP and the particle filters for robust 3d human body motion tracking”
ที่มา : Daehwan Kim and Daijin Kim, Vision networks for behavior analysis, Proceeding of the 1st ACM workshop on, 2008.
- [12] “Vision-Based Two Hand Detection And Tracking”
ที่มา : Jiajun Wen and Yinwei Zhan, Interaction Sciences : IT, Culture and Human, Proceeding of the 2nd International Conference on, 2009.
- [13] “Robust real-time upper body limb detection and tracking”
ที่มา : Matheen Siddiqui and Gerrard Medioni, Video surveillance and sensor networks, Proceeding of the 4th ACM international workshop on, 2006.

- [14] นางสาวจิราภา ทิพกรณ์ และ นายอริญชัย บั้งเงิน. (2555). "ต้นแบบสำหรับคอมพิวเตอร์
แบบเน้นการสัมผัสบนพื้นผิว : กรณีศึกษาคอมพิวเตอร์ช่วยสอนโน้ตดนตรีเบื้องต้น" ปรินญา
นิพนธ์ วศ.บ., มหาวิทยาลัยนเรศวร, พิษณุโลก



ภาคผนวก ก.

คู่มือการติดตั้งโปรแกรมประยุกต์

ก่อนที่จะสามารถรันโปรแกรมและประมวลผลภาพได้นั้น จำเป็นต้องติดตั้งโปรแกรม Microsoft Visual Studio 2012 และการตั้งค่าไลบรารีโอเพนซีวี เสียก่อน

1. การติดตั้งโปรแกรม Microsoft Visual C++ Express 2012 64-bit ในระบบปฏิบัติการ Windows 8 64-bit นั้นจะใช้เพื่อเป็นโปรแกรมในการคอมไพล์ (Compile) ขั้นตอนวิธีบางตัวที่เป็นภาษา C++ เพื่อนำไปใช้งานในโปรแกรมการควบคุมอากาศยานปีกหมุนด้วยท่าทางการเคลื่อนไหวของร่างกาย



รูปที่ ก-1 โปรแกรม Microsoft Visual Studio 2012

2. ดาวน์โหลดและติดตั้งไลบรารี OpenCV ได้ จาก “<http://opencv.willowgarage.com/wiki>” ในส่วน ของ Download ให้เลือก release version ดังรูป

Downloads

- You can download the **release version (2.4)** for **Unix**, **Windows** or **Android**.
- For the latest code, see the [Install Guide](#).

3. เลือกระบบปฏิบัติการของคอมพิวเตอร์ (ในที่นี้ผู้จัดทำเลือกใช้ opencv-win) ดังรูป

Looking for the latest version? [Download OpenCV-2.4.4-beta.exe \(334.8 MB\)](#)

Home

Name	Modified	Size	Downloads
opencv-android	2013-02-12		1
opencv-ios	2013-02-12		1
opencv-unix	2013-02-12		1
opencv-win	2013-02-12		1

Totals: 4 Items

4. เลือก version ของ OpenCV (ผู้จัดทำเลือกใช้ version 2.4.3) ดังรูป

Looking for the latest version? [Download OpenCV-2.4.4-beta.exe \(334.8 MB\)](#)

Home / opencv-win

Name	Modified	Size	Downloads
↑ Parent folder			
2.4.4	2013-07-12		1
2.4.3	2012-11-22		1
2.4.2	2012-07-04		1
2.4.1	2012-06-21		1
2.4.0	2012-05-01		1
2.3.1	2012-04-04		1

5. เลือก OpenCV-2.4.3.exe เพื่อดาวน์โหลดดังรูป

Looking for the latest version? [Download OpenCV-2.4.4-beta.exe \(334.8 MB\)](#)

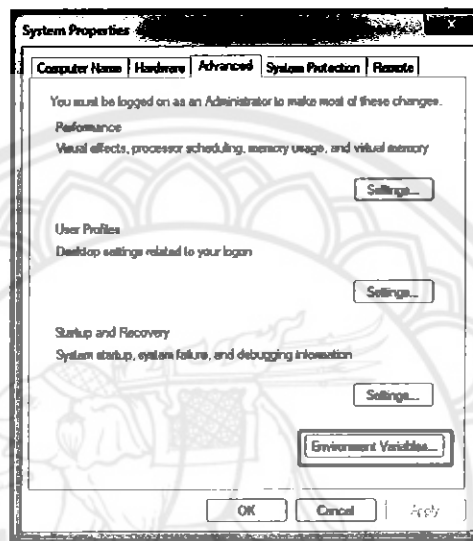
Home / opencv-win / 2.4.3

Name	Modified	Size	Downloads
↑ Parent folder			
OpenCV-2.4.3-GPU-demos-pack-x64.exe	2012-11-22	356.0 MB	164
OpenCV-2.4.3-GPU-demos-pack-x86.exe	2012-11-22	426.4 MB	148
checksum-2.4.3.bt	2012-11-02	103 Bytes	82
OpenCV-2.4.3.exe	2012-11-02	263.7 MB	5,369
Totals: 4 Items		1.0 GB	5,763

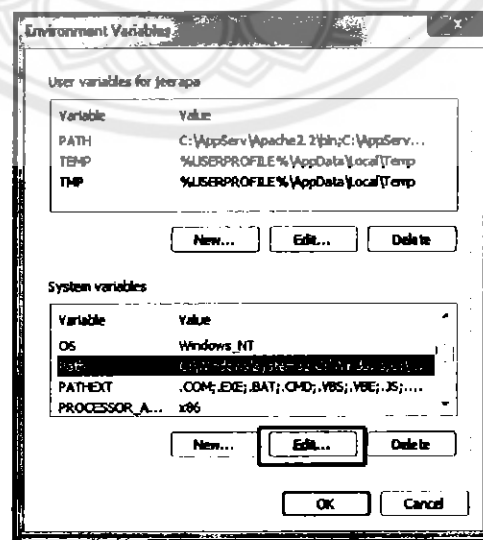
6. ทำการติดตั้งให้เรียบร้อย จะได้โฟลเดอร์ของ OpenCV ดังรูป

Modus	2/13/2013 11:00 PM	File folder
Mp3 To All Converter	8/15/2012 3:01 AM	File folder
OpenCV	11/13/2012 10:38 ...	File folder

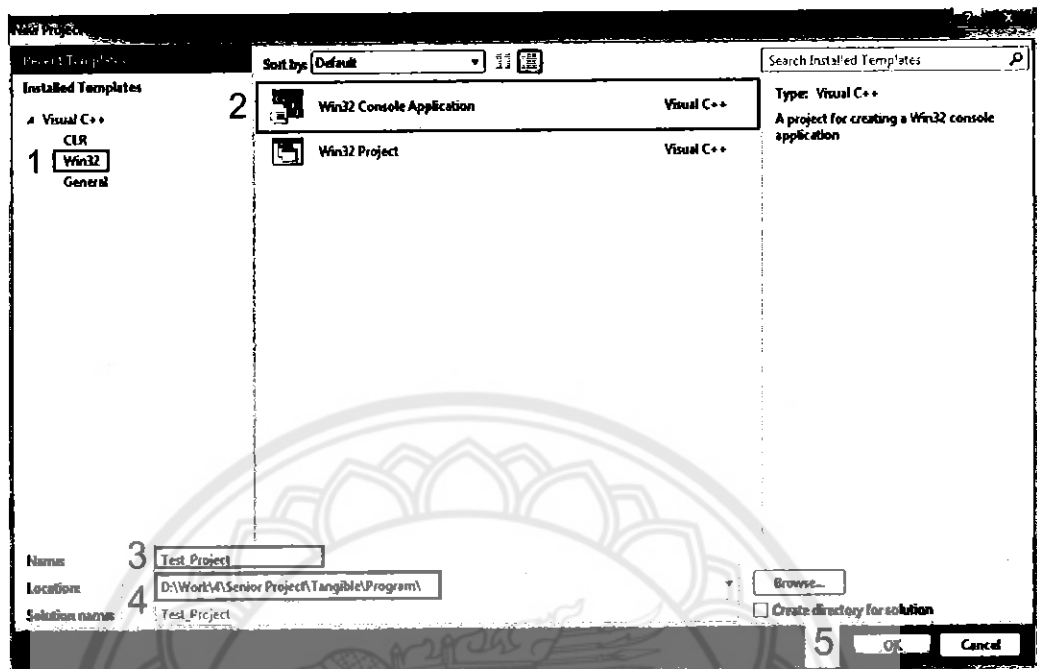
7. ทำการตั้งค่า path ดังนี้ ไปที่ “Control Panel > System and Security > System > Advance system settings” แล้วเลือก Environment Variable ดังรูป



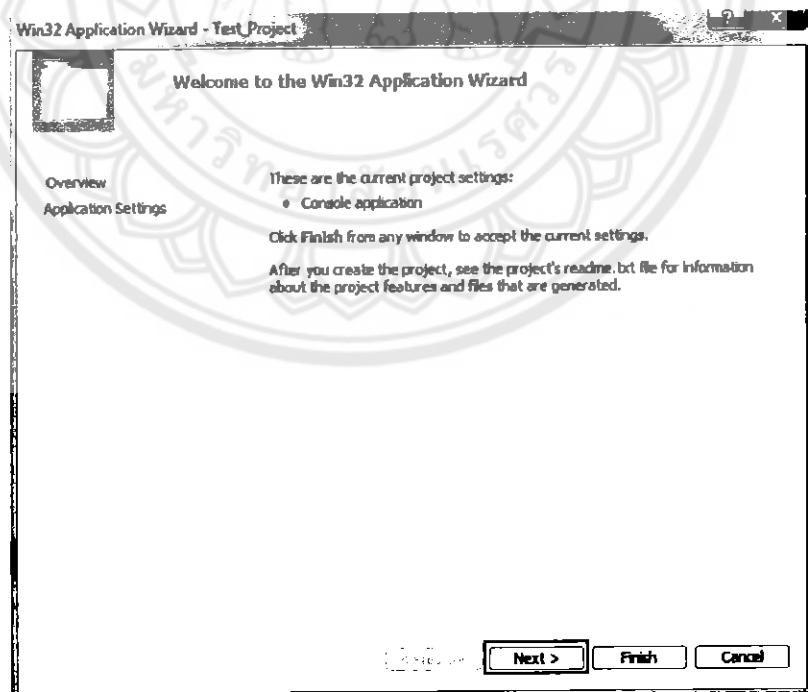
8. ที่ System variables ให้เลือกที่ Path แล้วคลิกปุ่ม edit เพื่อเพิ่ม path ของ OpenCV ดังนี้ “C:\OpenCV\opencv\build\x86\vc10\bin;” (ซึ่งชื่อของ Drive ที่ขีดเส้นใต้ จะขึ้นอยู่กับว่าผู้ใช้งานติดตั้ง OpenCV ไว้ที่ไหน) จากนั้นกดปุ่ม OK



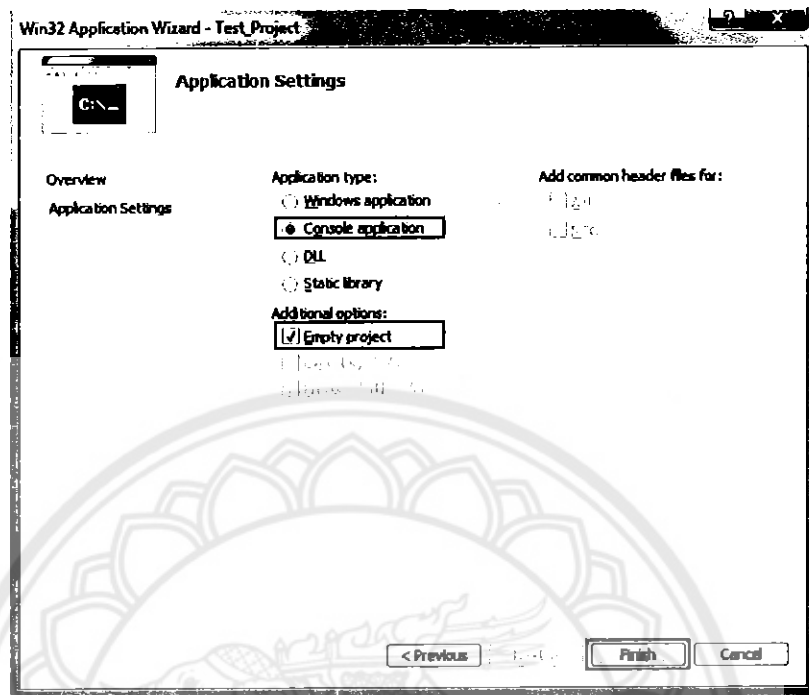
9. เปิดโปรแกรม Microsoft Visual Studio จากนั้น ทำการสร้าง New Project แล้วเลือก Win32 > Win32 Console Application > ตั้งชื่อ Project > เลือก Location ที่ต้องการ save แล้วกด OK



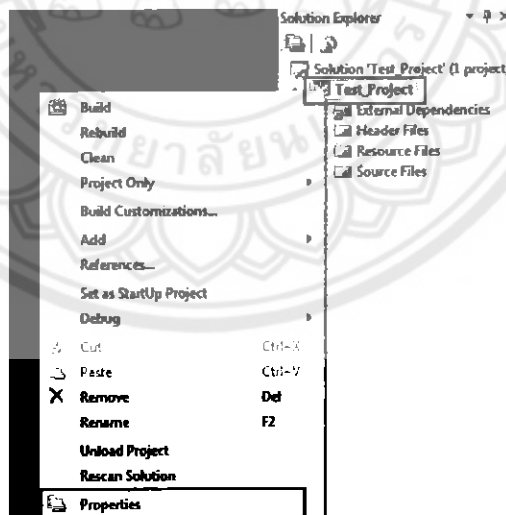
10. คลิก Next



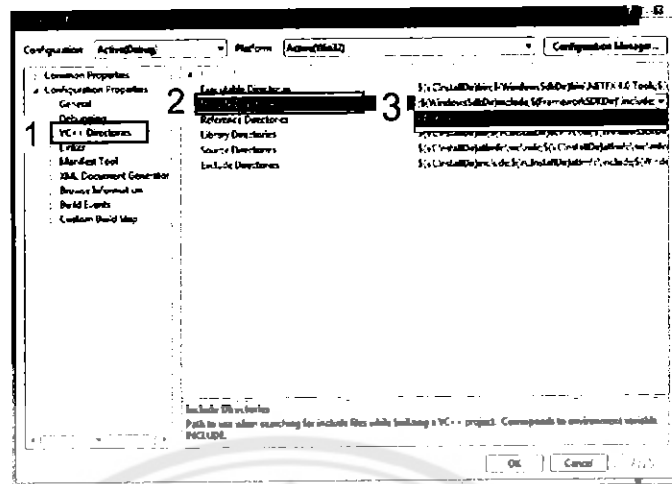
11. เลือก Console type เป็น “Console application” และ Additional options เป็น “Empty project” แล้วกดปุ่ม Finish



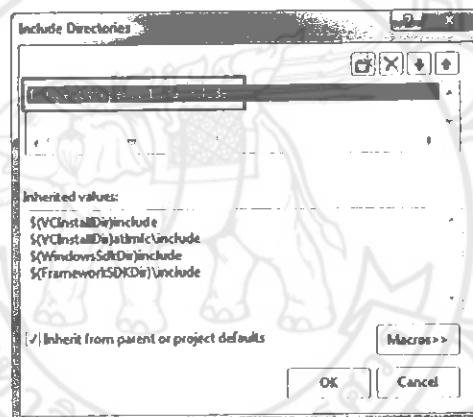
12. ที่หน้าต่าง Solution Explorer ให้คลิกขวาที่ ชื่อ project เลือก Properties



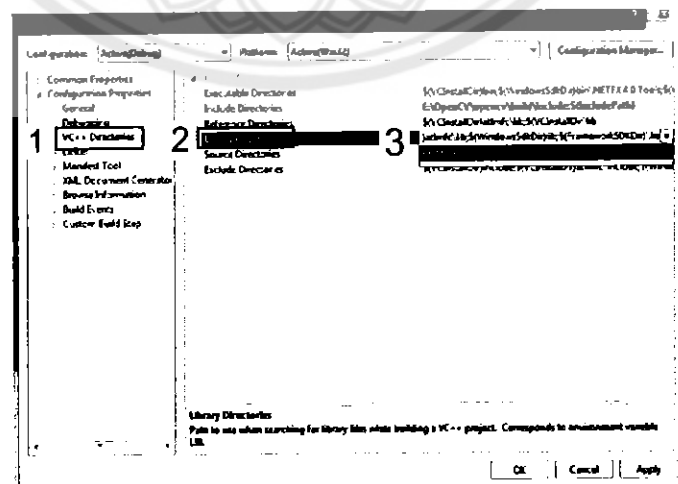
13. เลือก VC++ Directories > Include Directories > Edit



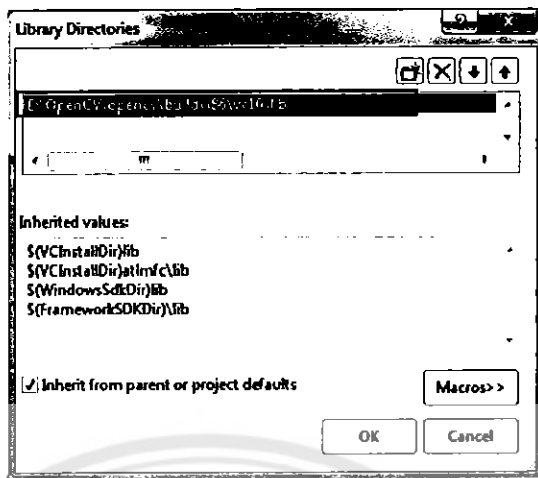
14. ให้เพิ่มโฟลเดอร์ C:\OpenC\opencv\build\include เข้ามาดังรูป แล้วกด OK



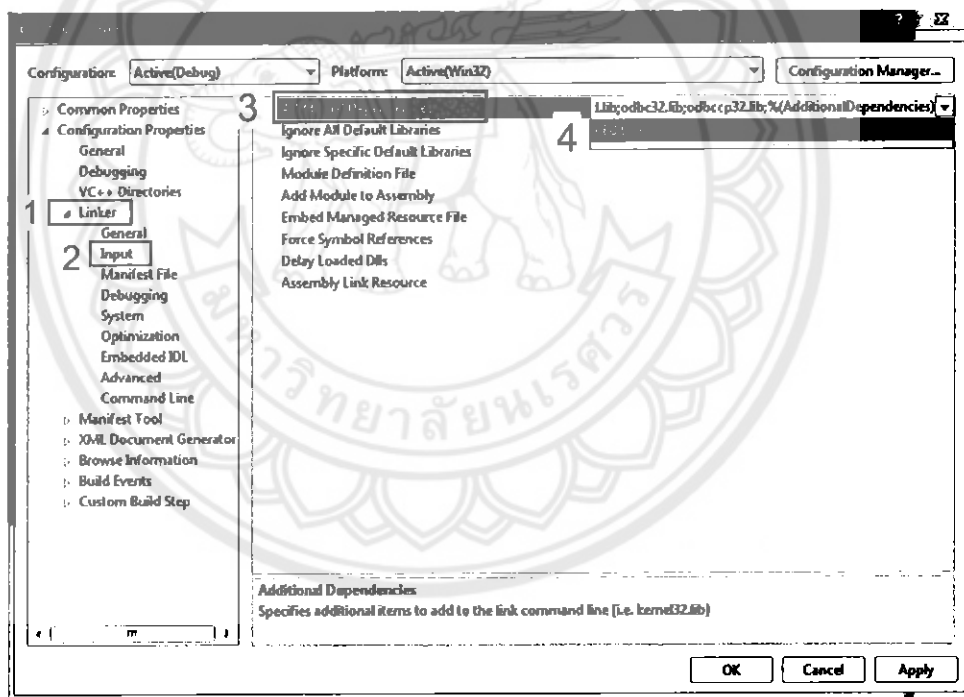
15. เลือก VC++ Directories > Library Directories > Edit



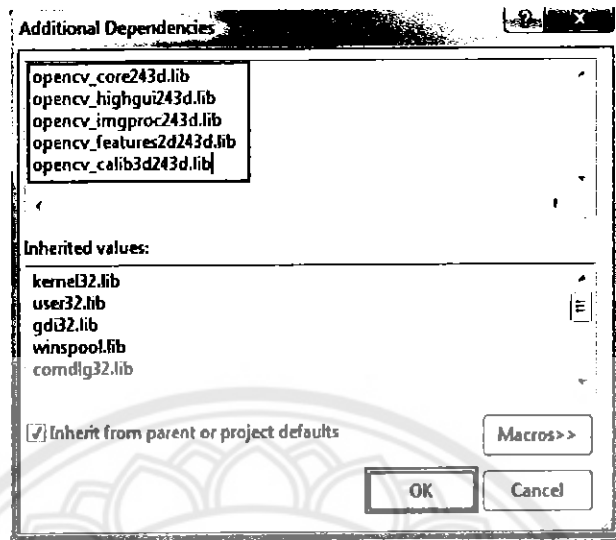
16. ให้เพิ่มโฟลเดอร์ E:\OpenCV\opencv\build\x86\vc10\lib เข้ามาดังรูป แล้วกด OK



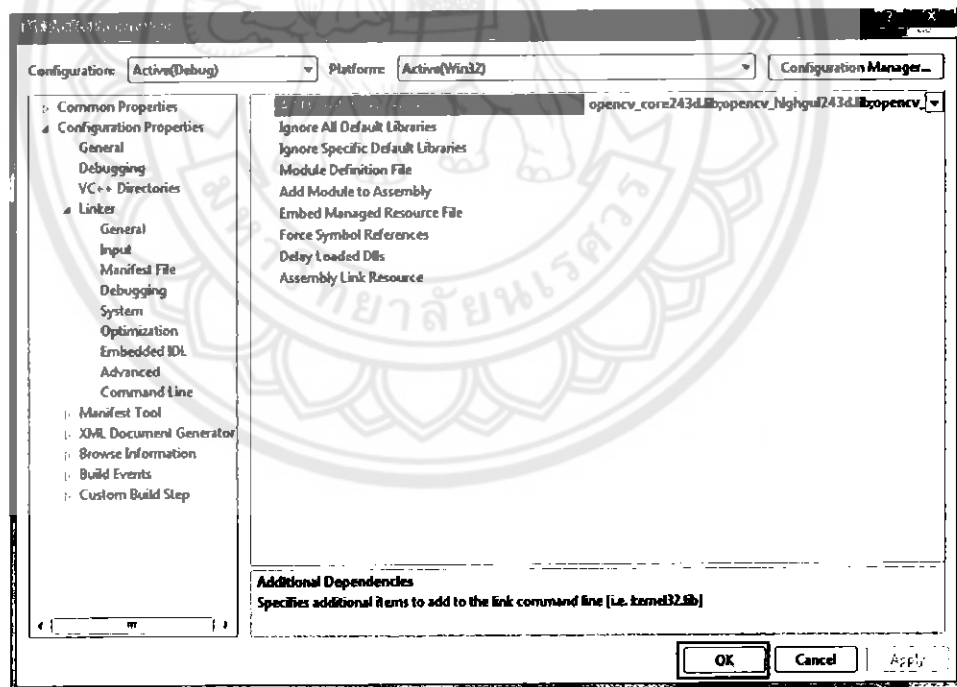
17. เลือก Linker > Input > Additional Dependencies > Edit



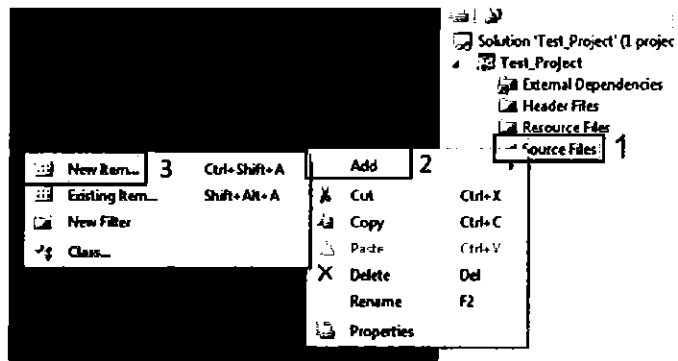
18. ทำการเพิ่ม .lib ดังรูป แล้วกด OK (opencv_core243d.lib ตัวเลข 243 คือ version ของ OpenCV ซึ่งในที่นี้เป็น version 2.4.3 ส่วน d คือบอกว่าเป็น debug mode)



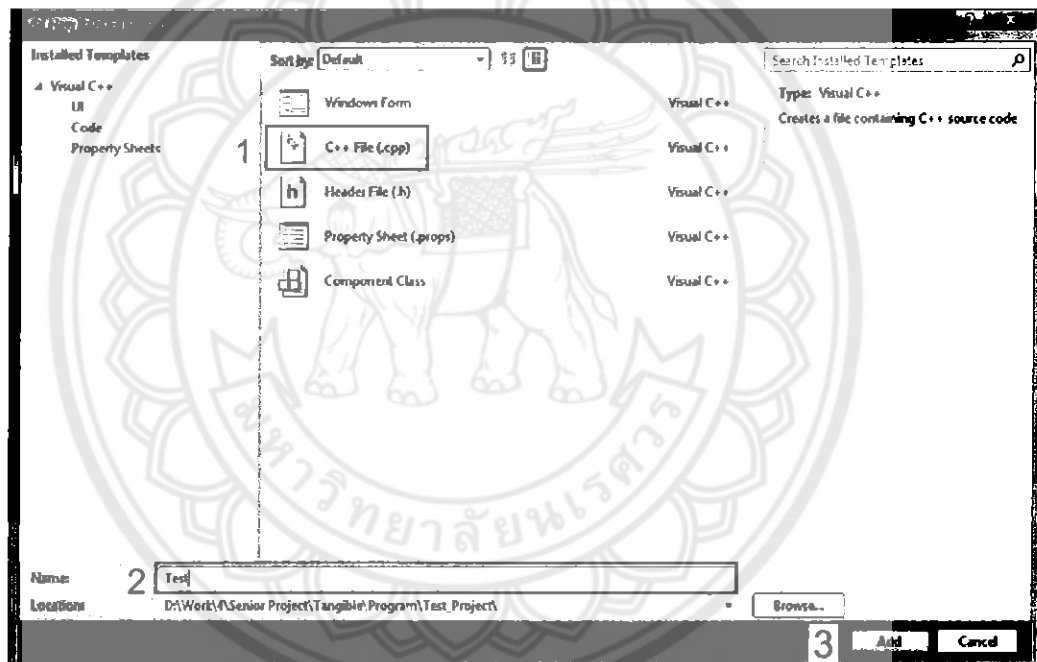
19. เมื่อตั้งค่าเสร็จ ให้คลิกปุ่ม OK



20. เมื่อตั้งค่าเสร็จแล้วให้คลิกขวาที่ Source File เลือก Add > New Item



21. เลือก C++ File (.cpp) > ตั้งชื่อ แล้วกด Add



การทดสอบการใช้งานโอเพนซีวีเมื่อติดตั้งเสร็จ

1. copy code ด้านล่างไปใส่ใน Microsoft Visual Studio

```
#include<opencv2\core\core.hpp>
#include<opencv2\highgui\highgui.hpp>
#define TEST_IMAGE "D:\\test.jpg" ← Path ของรูปภาพที่ใช้ทดสอบ
int main()
{
    cv::Mat img = cv::imread(TEST_IMAGE);

    cv::namedWindow("Image:", CV_WINDOW_AUTOSIZE);
    cv::imshow("Image:", img);

    cv::waitKey(0);

    cv::destroyWindow("Image:");
    return 0;
}
```

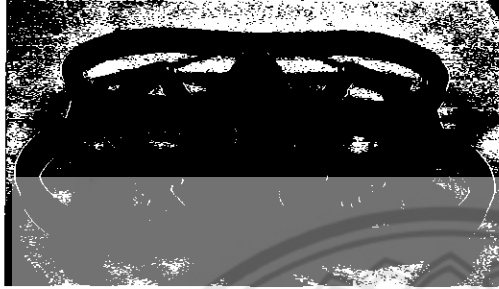
2. จะได้ผลการทดลองดัง ซึ่งหมายความว่า การติดตั้ง OpenCV สำเร็จ



ภาคผนวก ข.

อากาศยานปีกหมุนที่ใช้ในการทดสอบ

1. อากาศยานปีกหมุนแบบสี่ใบพัดรุ่น AR.Drone 2.0 ซึ่งมีลักษณะดังรูป



ด้านหน้า



ด้านหลัง

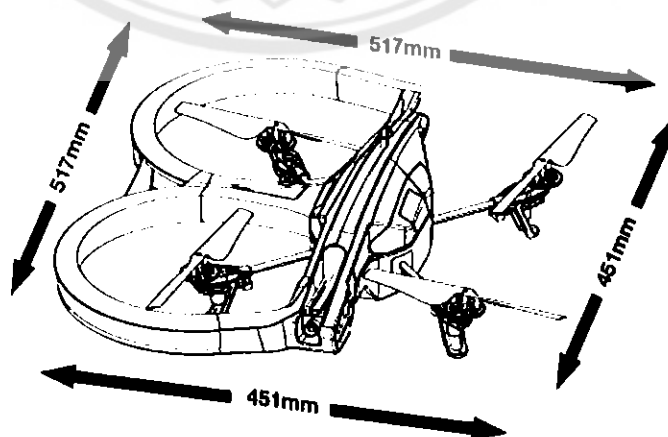


ด้านบน



ด้านล่าง

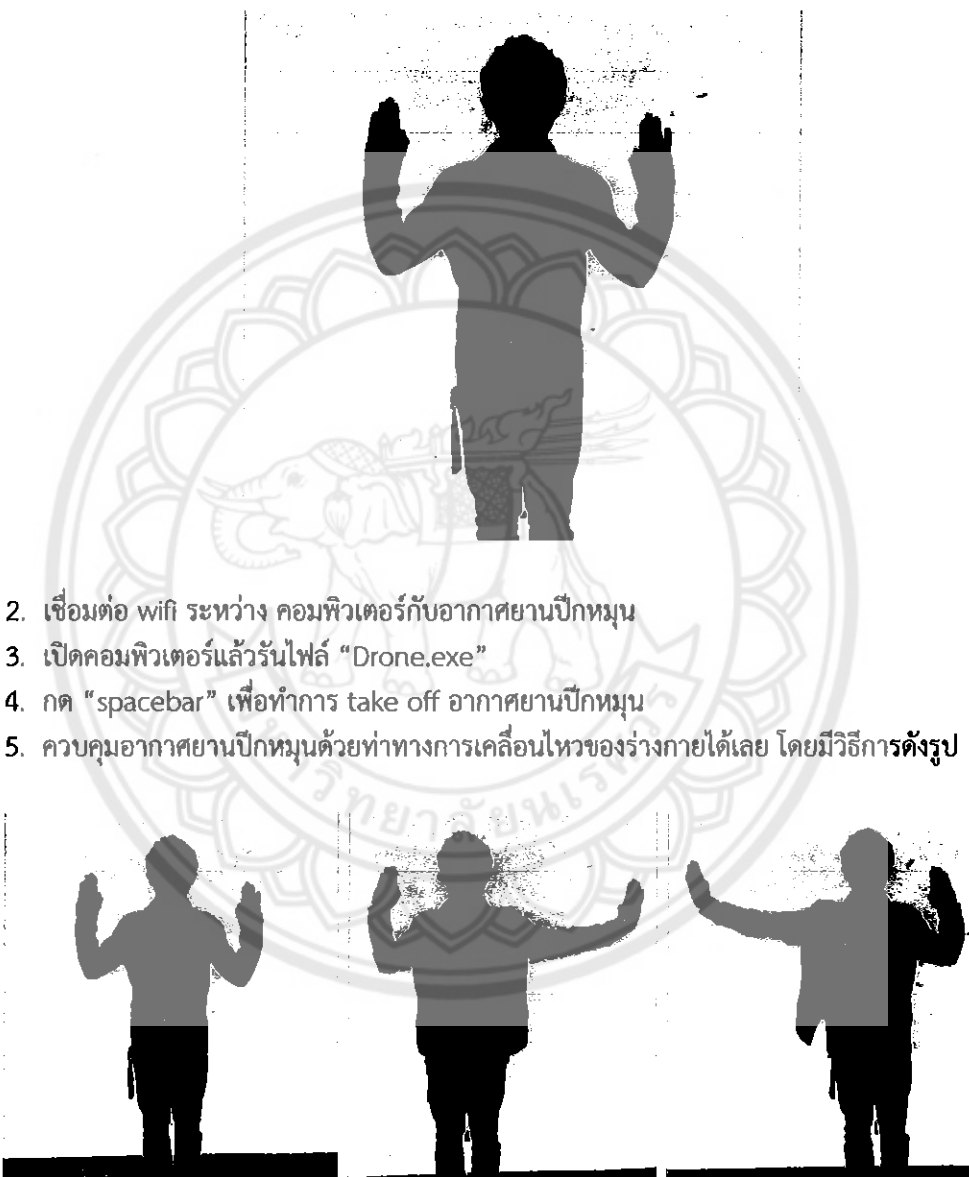
2. AR.Drone 2.0 มีน้ำหนักโดยรวมเท่ากับ 420 กรัม และมีขนาดความกว้าง-ยาว ดังรูป



ภาคผนวก ค.

คู่มือการใช้งานระบบ

1. เตรียมอุปกรณ์ (อากาศยานปีกหมุนและคอมพิวเตอร์) และสวมเสื้อผ้าให้เรียบร้อยดังรูป โดยสถานที่ที่ใช้ในการทดลอง ต้องไม่มีลม มีแสงที่เพียงพอ และมีพื้นหลังเป็นสีทึบเดียว



2. เชื่อมต่อ wifi ระหว่าง คอมพิวเตอร์กับอากาศยานปีกหมุน
3. เปิดคอมพิวเตอร์แล้วรันไฟล์ "Drone.exe"
4. กด "spacebar" เพื่อทำการ take off อากาศยานปีกหมุน
5. ควบคุมอากาศยานปีกหมุนด้วยท่าทางการเคลื่อนไหวของร่างกายได้เลย โดยมีวิธีการดังรูป

บินอยู่กับที่

บินไปทางซ้าย

บินไปทางขวา



บินขึ้น

บินลง



บินไปด้านหน้า



บินไปด้านหลัง

6. ถ้าต้องการ Landing ให้กด “spacebar” อีกครั้ง