



การสำรวจความเปลี่ยนแปลงภูมิทัศน์โดยอากาศยานไร้คนขับ

Surveys the Changing Landscape by UAV

นาย ณัฐกฤทธิ์ ฤทธิข้าบ 5607063

ปริญญาอนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ วิทยาลัยเทคโนโลยีสารสนเทศและการสื่อสาร

มหาวิทยาลัยรังสิต

ปีการศึกษา 2558

การสำรวจความเปลี่ยนแปลงภูมิทัศน์โดยอากาศยานไร้คนขับ

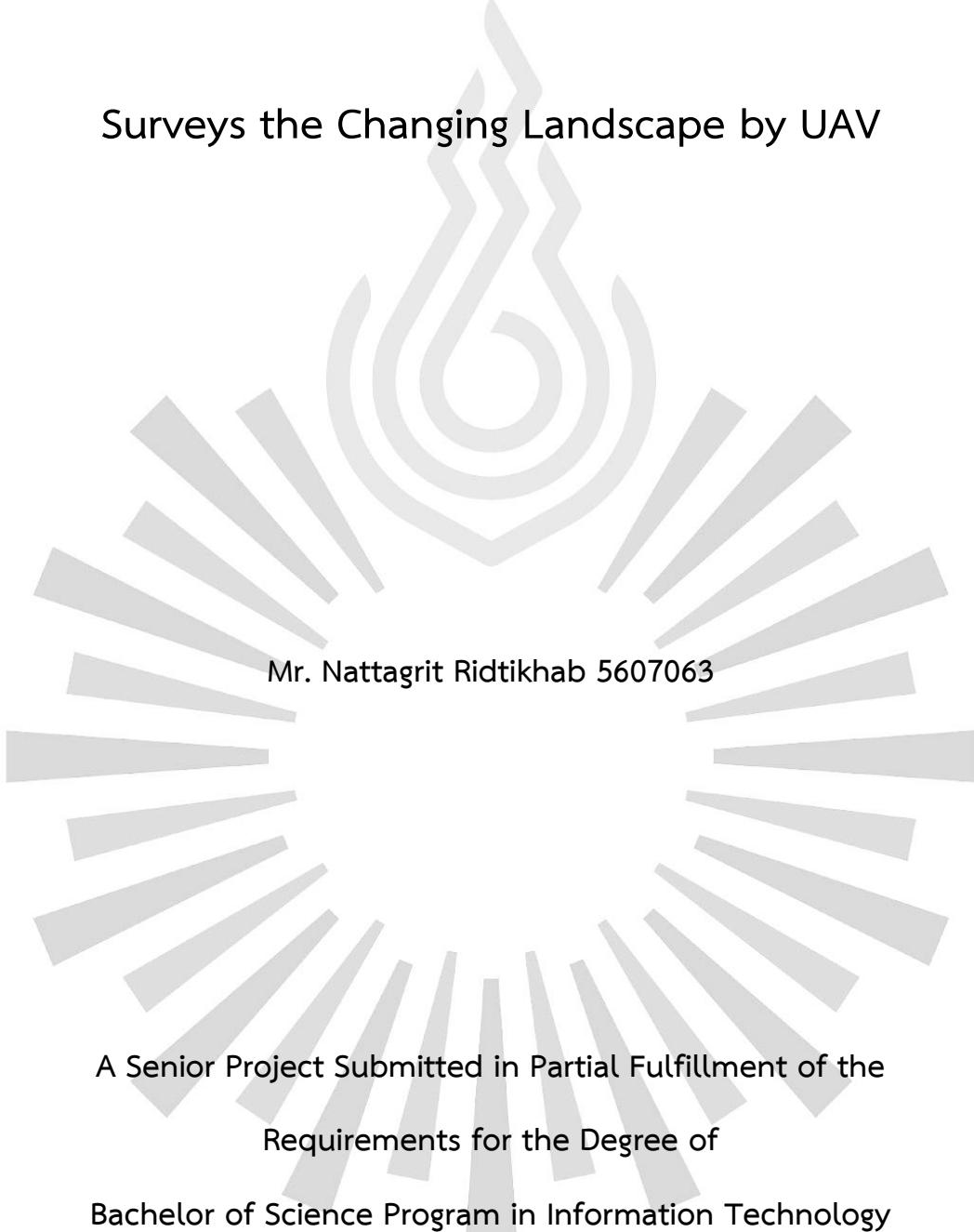
นาย ณัฐกฤทธิ์ ฤทธิข้าบ 5607063

ปริญญาอนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ วิทยาลัยเทคโนโลยีสารสนเทศและการสื่อสาร

มหาวิทยาลัยรังสิต

ปีการศึกษา 2558



Surveys the Changing Landscape by UAV

Mr. Nattagrit Ridtikhab 5607063

A Senior Project Submitted in Partial Fulfillment of the

Requirements for the Degree of

Bachelor of Science Program in Information Technology

Faculty of Information Technology

Rangsit University

Academic Year 2015



ใบรับรองปริญญานิพนธ์

หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
วิทยาลัยเทคโนโลยีสารสนเทศและการสื่อสาร มหาวิทยาลัยรังสิต

หัวข้อปริญญานิพนธ์

การสำรวจความเปลี่ยนแปลงภูมิทัศน์โดยอากาศยานไร้คนขับ

Surveys the Changing Landscape by UAV

โดย

นาย ณัฐกฤช ฤทธิข้า 5607063

อาจารย์ที่ปรึกษา

อาจารย์ ภาครุ่งษ์ ชัยศิริประเสริฐ

ปริญญานิพนธ์นี้ได้รับอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

หัวหน้าสาขาวิชา

(อาจารย์ สุรี ศิริสุทธิเดชา)

ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.วุฒิพงษ์ ชินศรี)

กรรมการ

(อาจารย์ สุรี ศิริสุทธิเดชา)

กรรมการและอาจารย์ที่ปรึกษา

(อาจารย์ ภาครุ่งษ์ ชัยศิริประเสริฐ)

ณัฐกฤษฐ์ ฤทธิขاب. 2558. การสำรวจความเปลี่ยนแปลงภูมิทัศน์โดยอากาศยานไร้คนขับ. ปริญญา
นิพนธ์สาขาวิชาเทคโนโลยีสารสนเทศ วิทยาลัยเทคโนโลยีสารสนเทศและการสื่อสาร
มหาวิทยาลัยรังสิต. อาจารย์ที่ปรึกษา: อาจารย์ภาควิชัยศิริประเสริฐ. 64 หน้า.

บทคัดย่อ

ปริญญานิพนธ์นี้มีวัตถุประสงค์เพื่อ วิเคราะห์ ออกแบบ และพัฒนาระบบ การสำรวจความเปลี่ยนแปลงภูมิทัศน์โดยอากาศยานไร้คนขับ เพื่อช่วยในการศึกษา ตรวจสอบ เฝ้าระวัง และติดตามความเปลี่ยนแปลงของภูมิทัศน์ ในการพัฒนาระบบนี้ เครื่องมือที่ใช้ คือ อากาศยานไร้คนขับ รุ่น Parrot AR. Drone 2.0 ส่วนโปรแกรมที่ใช้ในการพัฒนาระบบ ประกอบด้วย Robot Operating System (ROS), Open Source Computer Vision (OpenCV) ภาษาที่ใช้เขียนโปรแกรม คือ C++ โปรแกรมทั้งหมดนี้ทำงานบนระบบปฏิบัติการ Linux Ubuntu ซึ่งเป็น Open Source ทั้งหมด การทำงานคือ ใช้ AR. Drone ถ่ายภาพภูมิทัศน์ ก่อน และหลังเกิดความเปลี่ยนแปลง เข้าสู่โปรแกรมการประมวลผลภาพ เพื่อดึงความแตกต่างของภาพออกจากปรับค่าสี และแสดงของภาพ จากนั้นนำภาพที่ประมวลผลแล้ว มาวิเคราะห์ ช้อนกับภาพภูมิทัศน์ก่อนเกิดความเปลี่ยนแปลง ได้ภาพใหม่ที่มีผลของการเปลี่ยนแปลงอยู่ในภาพ พิริมราจานเปอร์เซ็นของปริมาณความเปลี่ยนแปลงที่เกิดขึ้น

ณัฐกฤษฐ์ ฤทธิขاب
(นาย ณัฐกฤษฐ์ ฤทธิขاب)

ภาควิชัยศิริประเสริฐ
(อาจารย์ ภาควิชัยศิริประเสริฐ)

Nattagrit Ridtikhab. 2015. Survey the Changing Landscape by UAV. Undergrad Senior Project, Department of Information Technology, Faculty of Information Technology, Rangsit University. Thesis Advisors: Pakpoom Chaisiriprasert. 64 pp.

Abstract

The objective of the project was to analyze, design and Development Survey the Changing Landscape by UAV. To assist in the monitoring and tracking of changes in the landscape at different times. UAV Parrot AR. Drone 2.0 was used for data analysis and program consists the Robot Operating System (ROS), Open Source Computer Vision (OpenCV). Using the C++ programming. All this runs on the Linux Ubuntu operating system, which is an Open Source all. The function is in operation. Take pictures taken before and after the change into an image processing program. To pull out the difference image adjust color and exposure, then the image is processed. The landscape overlaid with unrealized changes. A new visual effect of a change in the image. And report the percentage of volume change occurred



Nattagrit Ridtikhab

(Mr. Nattagrit Ridtikhab)

JmJp

(Advisors. Parkpoom Chaisiriprasert)

กิจกรรมประจำ

ปริญญาอินเทลลิเจนซ์สำเร็จลุล่วงไปได้ด้วยความช่วยเหลือและความเมตตากรุณาจากอาจารย์ภาควิชานิพัทธ์ ชัยศิริประเสริฐ อาจารย์ที่ปรึกษาปริญญาอินเทลลิเจนซ์ ผู้เป็นหลักสำคัญในการให้คำปรึกษาและแนะนำในสิ่งที่เป็นประโยชน์ รวมทั้งอุทิศเวลาอย่างมากในการตรวจแก้ไขข้อบกพร่องต่างๆ ของปริญญาอินเทลลิเจนซ์เล่มนี้ จนทำให้ปริญญาอินเทลลิเจนซ์สำเร็จลุล่วงไปได้ด้วยดี ผู้พัฒนาจึงขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี่

ผู้พัฒนาขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. วุฒิพงษ์ ชินศรี ประธานกรรมการการสอบ และ อาจารย์ สุรี ศิริสุทธิเดชา กรรมการสอบปริญญาอินเทลลิเจนซ์ ที่กรุณาชี้แนะแนวทางในการปรับปรุงปริญญาอินเทลลิเจนซ์ฉบับนี้ให้มีความสมบูรณ์ยิ่งขึ้น

ขอขอบพระคุณบิดามารดาที่ให้โอกาสในการศึกษาเล่าเรียน ตลอดจนช่วยเหลือและให้กำลังใจตลอดการพัฒนาระบบจนกระทั่งสำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ผู้พัฒนาขอขอบคุณเพื่อนๆ ทุกคน ที่ให้คำปรึกษา ช่วยแก้ไขปัญหา รวมถึงช่วยให้กำลังใจตลอดระยะเวลาในการทำปริญญาอินเทลลิเจนซ์ จนทำให้ปริญญาอินเทลลิเจนซ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

ณัฐกฤญ ฤทธิข้าบ

2558

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	๑
บทคัดย่อภาษาอังกฤษ.....	๒
กิตติกรรมประกาศ	๓
สารบัญ	๓
สารบัญรูปภาพ.....	๔
สารบัญตาราง	๕
บทที่ 1 บทนำ.....	๑
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตการพัฒนาระบบ.....	2
1.4 ขั้นตอนการพัฒนาระบบ	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.6 นิยามศัพท์เฉพาะ	3
1.7 แผนการดำเนินงานปริญญาอนิพนธ์	4
บทที่ 2 วรรณกรรมที่เกี่ยวข้อง	5
2.1 การสำรวจความเปลี่ยนแปลงภูมิทัศน์	6
2.2 การเคลื่อนที่อัตโนมัติของอากาศยานไร้คนขับ.....	8
2.3 คอมพิวเตอร์วิทัศน์ (Computer Vision).....	10
2.4 การประมวลผลภาพ (Image Processing)	12
2.5 ความรู้เบื้องต้นเกี่ยวกับ OpenCV	16
2.6 ระบบปฏิบัติการหุ่นยนต์ (ROS)	17
2.7 อากาศยานไร้คนขับ AR. Drone 2.0	18
บทที่ 3 ขั้นตอนวิธีการออกแบบและพัฒนาระบบ	19
3.1 ออกแบบสถาปัตยกรรมระบบ.....	20
3.2 วิธีการการถ่ายภาพภูมิทัศน์	21
3.3 วิธีการหาความเปลี่ยนแปลงภูมิทัศน์	22

สารบัญ (ต่อ)

	หน้า
3.4 ข้อกำหนดเกี่ยวกับเทคโนโลยีที่ใช้พัฒนาระบบ	24
บทที่ 4 การสำรวจความเปลี่ยนแปลงภูมิทัศน์โดยอากาศยานไร้คนขับ	25
4.1 การพัฒนาระบบ	26
4.2 การทำงานของระบบ	36
4.3 การทดสอบระบบ	41
บทที่ 5 บทสรุปและข้อเสนอแนะ.....	46
5.1 สรุปผลการพัฒนาระบบ.....	47
5.2 ปัญหาและอุปสรรคในการพัฒนาระบบ.....	48
5.3 แนวทางการแก้ไขปัญหาและข้อเสนอแนะ	48
บรรณานุกรม.....	49
ประวัติผู้จัดทำปริญญาในพิพิธ	51

สารบัญรูปภาพ

รูปที่	หน้า
2.1 การสำรวจจากระยะไกล ด้วยเครื่องบันทึกข้อมูลแบบต่างๆ	6
2.2 ภาพถ่ายจากดาวเทียม ติดตามการเปลี่ยนแปลงบริเวณพื้นที่เขื่อนป่าสักชลสิทธิ์.....	7
2.3 การเคลื่อนที่อัตโนมัติของ AR. Drone : โดยการจดจำเส้นทาง	8
2.4 ตำแหน่งอุปกรณ์ Sensors ของ AR. Drone 2.0.....	8
2.5 การเคลื่อนที่อัตโนมัติของ AR. Drone : โดยการอ่านคำสั่งจากภาพ	9
2.6 ตัวอย่างคำสั่งภาพในการเคลื่อนที่ของ AR. Drone เมื่อเข้าสู่โปรแกรม	9
2.7 การมองเห็นของคอมพิวเตอร์	10
2.8 สิ่งที่มนุษย์มองเห็นและเข้าใจได้	11
2.9 สิ่งที่คอมพิวเตอร์มองเห็นและวิธีการเข้าใจ	11
2.10 ขั้นตอนการประมวลผลภาพ (Image Processing)	12
2.11 ตำแหน่งของ Pixel.....	13
2.12 ตัวอย่างการปรับค่า Threshold	14
2.13 แบบจำลองสี (RGB Color)	15
2.14 แบบจำลองสี (Grayscale)	15
2.15 แบบจำลองสี (Alpha)	15
2.16 Logo OpenCV	16
2.17 Logo ROS industrial.....	17
2.18 การทำงานของ ROS	17
2.19 Parrot AR. Drone 2.0.....	18
3.1 System Architecture design	20
3.2 การตัดไม้ทำลายป่า (ซ้าย) การตัดไม้ทำลายป่าที่เพิ่มขึ้น (ขวา)	21
3.3 ภาพจำลองการใช้อากาศยานไร้คนขับ ถ่ายภูมิทัศน์จากรูปภาพ	22
3.4 ภาพตัวอย่าง เกมจับผิดภาพ ก่อนประมวลผล : นำเข้า.....	23
3.5 ภาพตัวอย่าง เกมจับผิดภาพ หลังประมวลผล : นำออก.....	23
4.1 ภาพภูมิทัศน์จำลองสถานการณ์ สิงปลูกสร้าง : ก่อน	26
4.2 ภาพภูมิทัศน์จำลองสถานการณ์ สิงปลูกสร้าง : หลัง	27

สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
4.3 ภาพภูมิทัศน์จำลองสถานการณ์ ตัดไม้ทำลายป่า : ก่อน	27
4.4 ภาพภูมิทัศน์จำลองสถานการณ์ ตัดไม้ทำลายป่า : หลัง	28
4.5 ภาพภูมิทัศน์จำลองสถานการณ์ การกัดเคาะของแม่น้ำ : ก่อน	28
4.6 ภาพภูมิทัศน์จำลองสถานการณ์ การกัดเคาะของแม่น้ำ : หลัง	29
4.7 Folder Package ROS BAYWORK	29
4.8 manifest.xml	30
4.9 CMakeList.txt	30
4.10 Folder bin Executable file	31
4.11 Folder src Source Code Program	31
4.12 Source Code Capture 1	32
4.13 Source Code Capture 2	32
4.14 Source Code Difference	33
4.15 Source Code Transparent 1	33
4.16 Source Code Transparent 2	33
4.17 Source Code Overlay 1	34
4.18 Source Code Overlay 2	34
4.19 Source Code Blend 1	35
4.20 Source Code Blend 2	35
4.21 Window Program Keycap	36
4.22 Window Program Difference	36
4.23 Window Program NOBLACK : Gray Color (ขวา) Transparent (ซ้าย)	37
4.24 Window Program OVERLAY	38
4.25 Window Program Blend	39
4.26 Bash Script File.sh	40
4.27 Test From Webcam Camera	41
4.28 Test From AR. Drone 2.0 Camera	42

สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
4.29 Test From Original Image.....	43
4.30 Test ความคลาดเคลื่อนของภาพ : ระดับน้อย	44
4.31 Test ความคลาดเคลื่อนของภาพ : ระดับปานกลาง	44
4.32 Test ความคลาดเคลื่อนของภาพ : ระดับมาก.....	45



สารบัญตาราง

ตารางที่

หน้า

1.1 แผนการดำเนินงานปริญญา妮พนธ์	4
--------------------------------------	---



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

จอมยุทธ บ้านจอมยุทธ (2543) ได้ศึกษาเบรียบเทียบ ผลการจัดลำดับความสำคัญของปัญหา ทรัพยากรธรรมชาติ และสิ่งแวดล้อม ด้วยมูลค่าความเสียหาย และการสำรวจทัศนคติของประชาชน พบว่า ปัญหาสำคัญ 5 ลำดับแรก มีดังนี้ ลำดับที่ 1 การสูญเสียทรัพยากรป่าไม้ ลำดับที่ 2 อุทกภัย และภัยแล้ง ลำดับที่ 3 ความเสื่อมโทรมของทรัพยากรดินและการใช้ที่ดิน ลำดับที่ 4 มลพิษจากขยายและลำดับที่ 5 มลพิษทางอากาศ

จากการศึกษาผลงานทางวิชาการ พบร่วมกับ “มนุษย์” เป็นผู้ทำลายสิ่งแวดล้อมมากที่สุด สาเหตุที่มนุษย์เป็นผู้ทำลายสิ่งแวดล้อมเกิดจากความเห็นแก่ตัวของมนุษย์เอง โดยใช้ความชัญฉลาด ของตนเอง มาทำลายล้าง เพื่อแก่งแย่งปัจจัยพื้นฐานในการดำรงชีวิตของมนุษย์ด้วยกันเอง จึงเกิดการทำลายทรัพยากรธรรมชาติ และสิ่งแวดล้อม เพื่อให้ตนเองอยู่รอดในสังคม เมื่อสิ่งแวดล้อมถูกทำลายมากขึ้น ผลกระทบก็ย้อนกลับมาทำลายตัวมนุษย์เอง เช่น เกิดการเปลี่ยนแปลงของบรรยากาศโลก เกิดสภาพเรือนกระเจา ภาวะโลกร้อน ตลอดจนเกิดภัยธรรมชาติต่างๆ เช่น น้ำท่วม แผ่นดินถล่ม ควันพิษ น้ำเน่าเสีย ขยะมูลฝอย และสิ่งปฏิกูล ซึ่งสิ่งเหล่านี้มีผลโดยตรงและทางอ้อม และไม่สามารถหลีกเลี่ยงได้

กัญจน์เบจร ชูชีพ (2552) พบร่วมกับ ภาพถ่ายทางอากาศ สามารถนำไปใช้ประโยชน์เพื่อการสำรวจ และการศึกษาในสาขาวิชาต่างๆ เช่น การทำแผนที่สภาพภูมิประเทศ ธรณีวิทยา การป้องกันน้ำท่วม การสร้างถนนและเส้นทางคมนาคม การสร้างเขื่อนและอ่างเก็บน้ำ การจัดการทรัพยากรธรรมชาติ การวางแผนเมือง การประมง สิ่งแวดล้อม โบราณคดี และทางด้านยุทธศาสตร์ทางการทหาร อีกด้านหนึ่งคือ ในด้านการเกษตรภาพถ่ายทางอากาศได้อำนวยประโยชน์ในการดำเนินกิจกรรมหลายอย่าง เช่น การสำรวจและทำแผนที่ดิน การใช้ประโยชน์ที่ดิน การรังวัดที่ดิน การจัดการป่าไม้ การประเมินผลผลิตทางการเกษตร และการชลประทาน เป็นต้น

ภาพถ่ายทางอากาศทั่วไปมาจาก 2 แหล่ง ได้แก่ ดาวเทียม และเครื่องบิน ภาพถ่ายดาวเทียม ครอบคลุมพื้นที่ขนาดใหญ่ มีความละเอียดเพียง 50 ซม. ในขณะครื่นบินถ่ายภาพทางอากาศ จะมีความละเอียดถึง 10 ซม. แต่มีข้อจำกัดที่การขึ้นบินถ่ายภาพทางอากาศทุกครั้งจะต้องได้รับอนุญาต และการประสานงานของนักบินกับช่างภาพ ที่สำคัญที่สุดวิธีการทั้งสอง เป็นกระบวนการที่มีราคาแพง ใช้ต้นทุนในการสำรวจสูง

อากาศยานไร้คนขับ หรือ เครื่องบินบังคับวิทยุ ที่เรียกในอีกชื่อว่า โดรน ช่วยลดข้อจำกัด ดังกล่าว ในการถ่ายภาพทางอากาศได้ Parrot AR Drone รุ่น 2.0 เป็นเครื่องบินบังคับที่ถูกพัฒนาให้มีความสามารถมากกว่าเครื่องบินบังคับทั่วไป โดยจุดเด่นของ AR Drone 2.0 คือมีกล้องติดตั้งมา 2 ตัวที่บริเวณด้านหน้า และด้านล่างของตัวเครื่องบิน ซึ่งกล้องมีคุณภาพระดับ HD โดยกล้องสามารถบันทึกภาพแบบ Real-Time ส่งตรงมาอย่างคอมพิวเตอร์ได้ผ่านสัญญาณ Wi-Fi ซึ่งทำให้สามารถบังคับ เครื่องบินเข้าไปในพื้นที่ ที่มองไม่เห็นได้ โดยใช้กล้องในการคอยดูเส้นทาง

ข้อดีอีกอย่างของอากาศยานไร้คนขับ AR Drone 2.0 คือ สามารถเชื่อมต่อกับคอมพิวเตอร์ และเขียนโปรแกรมเพื่อเพิ่มความสามารถในการบินให้กับอากาศยานไร้คนขับได้

ดังนั้นผู้พัฒนาระบบจึงมีความสนใจในการนำอากาศยานไร้คนขับมาช่วยลดข้อจำกัดในการถ่ายภาพ และช่วยการบันทึกภาพของภูมิทัศน์ในเวลาแตกต่างกัน จากนั้นทำการประมวลผลภาพ วิเคราะห์ความเปลี่ยนแปลง เพื่อให้ถึงเห็นภาพรวมของการใช้พื้นที่ และการเปลี่ยนแปลงต่างๆ ตามที่ปรากฏบนพื้นผิวโลก เช่น การตัดไม้ทำลายป่า การเกิดอุทกภัย น้ำป่าไหลหลาก ไฟป่า การใช้ที่ดิน และการก่อสร้างสถานที่ เป็นต้น

1.2 วัตถุประสงค์

ปริญญาอนิพนธ์นี้มีวัตถุประสงค์เพื่อศึกษา วิเคราะห์ ออกแบบ และพัฒนาระบบ การสำรวจ ความเปลี่ยนแปลงภูมิทัศน์โดยอากาศยานไร้คนขับ

1.3 ขอบเขตการพัฒนาระบบ

- 1.3.1 ภาษาคอมพิวเตอร์ที่ใช้พัฒนาระบบ คือภาษา C++
- 1.3.2 ระบบปฏิบัติการที่นิยนต์ที่ใช้พัฒนาระบบ คือ ROS
- 1.3.3 การประมวลผลภาพที่ใช้พัฒนาระบบ คือ Library OpenCV
- 1.3.4 อากาศยานไร้คนขับที่ใช้พัฒนาระบบ คือ AR. Drone 2.0

1.4 ขั้นตอนการพัฒนาระบบ

- 1.4.1 ศึกษาความเป็นไปได้ในการพัฒนาระบบ
- 1.4.2 ศึกษาหาข้อกำหนดที่ต้องการพัฒนาระบบ
- 1.4.3 วิเคราะห์ และออกแบบระบบ
- 1.4.4 พัฒนาระบบทามแผนที่ออกแบบไว้
- 1.4.5 ทดสอบระบบ
- 1.4.6 ประเมินผลการใช้งานระบบ
- 1.4.7 จัดทำปริญญาพนธนัชบัปสมบูรณ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 ต้นแบบโปรแกรมเบื้องต้น ในการประมวลผลภาพความเปลี่ยนแปลงภูมิทัศน์
- 1.5.2 ช่วยในระบบงานด้านการถ่ายภาพทางอากาศ หรือภาพถ่ายดาวเทียม
- 1.5.3 เป็นแนวคิดเบื้องต้น และเป็นความรู้ สำหรับการพัฒนาต่อ

1.6 นิยามคัพท์เฉพาะ

- 1.6.1 AR Drone 2.0 หมายถึง อากาศยานไร้คนขับ
- 1.6.2 Image processing หมายถึง การประมวลผลภาพ
- 1.6.3 OpenCV หมายถึง คลังข้อมูลที่รวมรวมโปรแกรมการประมวลผลภาพ
- 1.6.4 C++ หมายถึง ภาษาคอมพิวเตอร์ที่ใช้ในการพัฒนาระบบ
- 1.6.5 ROS หมายถึง ระบบปฏิบัติการหุ่นยนต์

1.7 แผนการดำเนินงานปริญญาบัณฑิต

ตารางที่ 1.1 แผนการดำเนินงานปริญญาบัณฑิต

ลำดับ	ลักษณะการดำเนินงาน	ระยะเวลาการดำเนินงาน						
		ม.ย 58	ก.ค 58	ส.ค 58	ก.ย 58	ต.ค 58	พ.ย 58	ธ.ค 58
1	ศึกษาและปรึกษาความเป็นไปได้ในการพัฒนาระบบกับอาจารย์ที่ปรึกษาปริญญาบัณฑิต	↔						
2	ศึกษาหาเครื่องมือในการพัฒนาระบบ ในด้าน Hardware และ Software ที่ใช้ในการพัฒนา	↔						
3	วิเคราะห์และออกแบบระบบ		↔	↔				
4	เริ่มต้นพัฒนาระบบ			↔	↔			
5	ทดสอบและแก้ไขระบบให้ตรงกับความต้องการ				↔	↔		
6	ทดลองใช้ระบบที่ได้พัฒนา					↔	↔	
7	ประเมินผลการทำงานของระบบ						↔	↔
8	รวมเล่มปริญญาบัณฑิต							↔

บทที่ 2

วรรณกรรมที่เกี่ยวข้อง

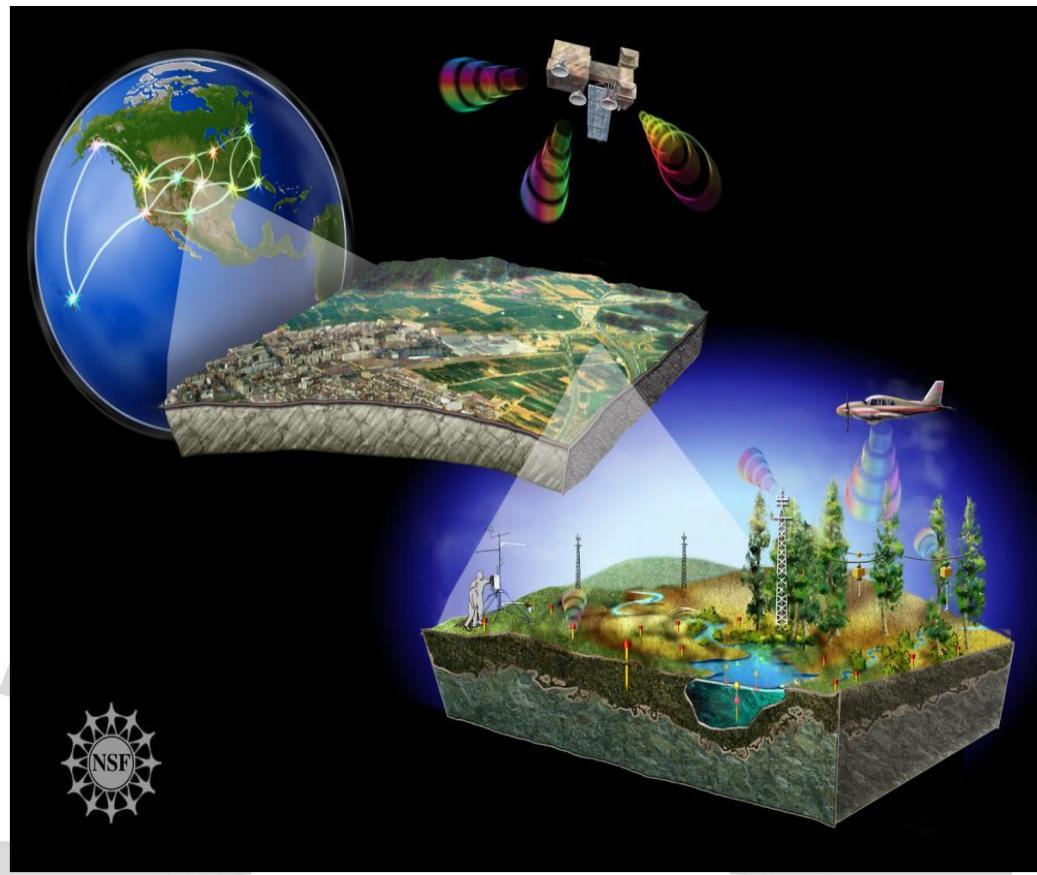
ในบทนี้ผู้พัฒนาระบบได้ปริศนาวรรณกรรมที่เกี่ยวข้อง โดยจำแนกออกเป็น 7 ตอน คือ

- 2.1 การสำรวจความเปลี่ยนแปลงภูมิทัศน์
- 2.2 การเคลื่อนที่อัตโนมัติของอากาศยานไร้คนขับ
- 2.3 คอมพิวเตอร์วิทัศน์ (Computer Vision)
- 2.4 การประมวลผลภาพ (Image Processing)
- 2.5 ความรู้เบื้องต้นเกี่ยวกับ OpenCV
- 2.6 ระบบปฏิบัติการหุ่นยนต์ (ROS)
- 2.7 อากาศยานไร้คนขับ AR. Drone 2.0

ทั้ง 7 ตอน มีรายละเอียดดังนี้

2.1 การสำรวจความเปลี่ยนแปลงภูมิทัศน์

2.1.1 เทคโนโลยีการสำรวจจากระยะไกล หรือ รีโมตเซนซิ่ง (Remote Sensing)

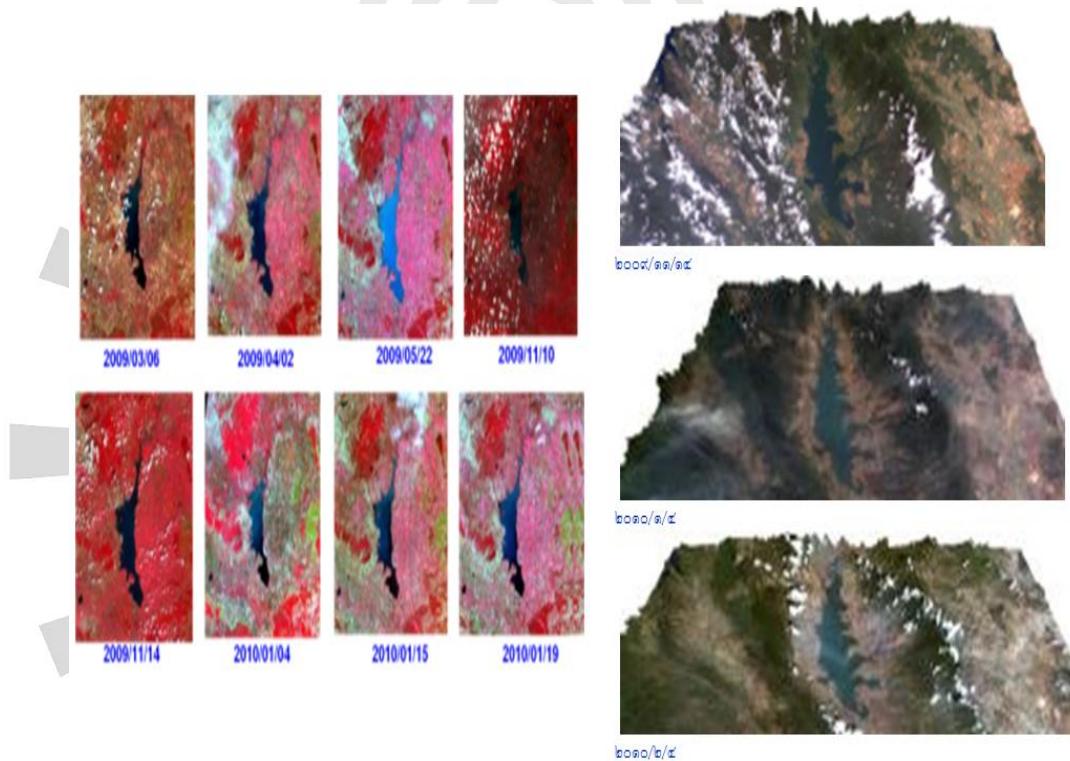


รูปที่ 2.1 การสำรวจจากระยะไกล ด้วยเครื่องบันทึกข้อมูลแบบต่างๆ

แก้ว นวลดีวีม (2535) ได้ให้ความหมายไว้ว่า คำนิยามของ "การสำรวจจากระยะไกล" หรือ "รีโมตเซนซิ่ง" นั้น หมายถึง กระบวนการที่เป็นทั้งศิลป์และศาสตร์ ในการทำให้ได้มาซึ่ง ข้อมูล และข่าวสารเกี่ยวกับวัตถุ หรือเหตุการณ์ที่อยู่ในระยะไกลออกไป จากเครื่องมือบันทึก ข้อมูล โดยปราศจากการสัมผัสโดยตรง ระหว่างเครื่องมือบันทึกข้อมูลกับวัตถุ หรือเหตุการณ์ นั้นๆ การที่จะให้ได้มาซึ่งข้อมูล และข่าวสารดังกล่าวแล้ว จะต้องใช้เครื่องมือ หรืออุปกรณ์ หลายอย่างประกอบกันเข้า เพื่อให้ทำงานอย่างมีประสิทธิภาพ ซึ่งเครื่องมือ หรืออุปกรณ์ เหล่านั้น ได้แก่ เครื่องบันทึกข้อมูล (Sensors) กรรมวิธีข้อมูล (Data Processing) วิธีการ และกรรมวิธีสนเทศ การสื่อสาร อากาศยาน และยานอวกาศ ที่เหมาะสม รวมไปถึงระบบ การทำงานอย่างมีประสิทธิภาพจากอากาศยาน และยานอวกาศ เป็นอาทิ ซึ่งทั้งหมดที่กล่าวมานี้รวมกันเข้าเป็นวิธีการของ เทคโนโลยีการสำรวจทรัพยากรจากระยะไกล

2.1.2 การตรวจจับความเปลี่ยนแปลงภูมิทัศน์ สถานีดาวเทียมจุฬาภรณ์

สถานีดาวเทียม CSRS (2553) พบว่าการติดตามสถานการณ์ของทรัพยากรธรรมชาติ ที่ส่งผลกระทบบุนแรงในพื้นที่ของประเทศไทย ไม่ว่าจะเป็นสถานการณ์น้ำของประเทศไทย การวิเคราะห์พื้นที่น้ำท่วมข้าzaาก การบุกรุกป่าไม้ และพื้นที่อุทยานแห่งชาติ ปัญหาเรื่องมลพิษ ทางอากาศ และการกัดเซาะชายฝั่ง เป็นต้น รวมถึงการติดตาม และประเมินความเสียหาย ของพื้นที่จากภัยพิบัติ การออกแบบระบบสื่อสารสำรองในภาวะภัยพิบัติ ด้วยการบูรณาการ ภาพถ่ายดาวเทียมร่วมกับปัจจัยต่างๆ ที่เกี่ยวข้อง ซึ่งการดำเนินการตามภารกิจตั้งแต่ล่าสุด และ วิเคราะห์ความเปลี่ยนแปลงที่เกิดขึ้นโดยอ้างอิงจากฐานข้อมูลที่มีรวมถึงสร้างคลังข้อมูล สำหรับใช้ในการตรวจสอบในด้านทรัพยากรธรรมชาติ



รูปที่ 2.2 ภาพถ่ายจากดาวเทียม ติดตามการเปลี่ยนแปลงบริเวณพื้นที่เขื่อนป่าสักชลสิทธิ์

การติดตามสถานการณ์น้ำ (ซ้าย) บริเวณพื้นที่เขื่อนป่าสักชลสิทธิ์ ตั้งแต่วันที่ 6 มีนาคม 2552 จนถึง 19 มกราคม 2553 และ (ขวา) บริเวณพื้นที่เขื่อนศรีนคินทร์แบบ 3 มิติ ในช่วง 14 พฤษภาคม 2552 จนถึง 4 กุมภาพันธ์ 2553 ด้วยข้อมูลภาพ CCD จากดาวเทียม SMMS โดยติดตามการเปลี่ยนแปลงของพื้นที่ และการใช้ประโยชน์พื้นที่

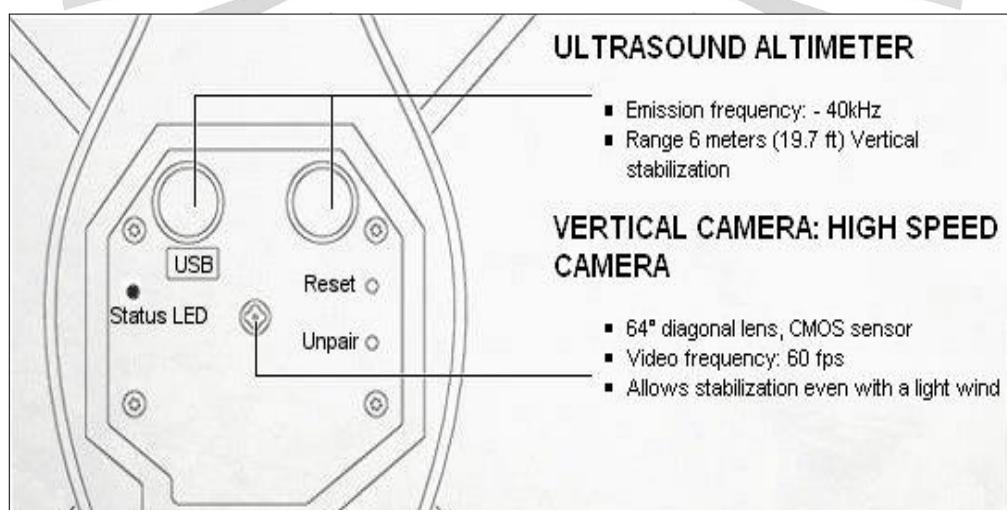
2.2 การเคลื่อนที่อัตโนมัติของอากาศยานไร้คนขับ

2.2.1 Visual Navigation System for Autonomous Figure Flying

Jakob Engel (2012) ได้พัฒนาระบบให้ AR. Drone สามารถระบุตำแหน่งที่อยู่ของตัวเองได้ โดยใช้ Sensor และกล้องหน้าของ AR. Drone ในการระบุตำแหน่งที่อยู่ ถ้าหาก AR. Drone ขยับไปอยู่ในตำแหน่งที่ผิดจากที่ตั้งไว้ AR. Drone จะทำการบินกลับมายังตำแหน่งที่ตั้งไว้อัตโนมัติ โดยไม่จำเป็นต้องใช้ GPRS แต่จะใช้การจดจำเส้นทางจาก 3D Map



รูปที่ 2.3 การเคลื่อนที่อัตโนมัติของ AR. Drone : โดยการจดจำเส้นทาง



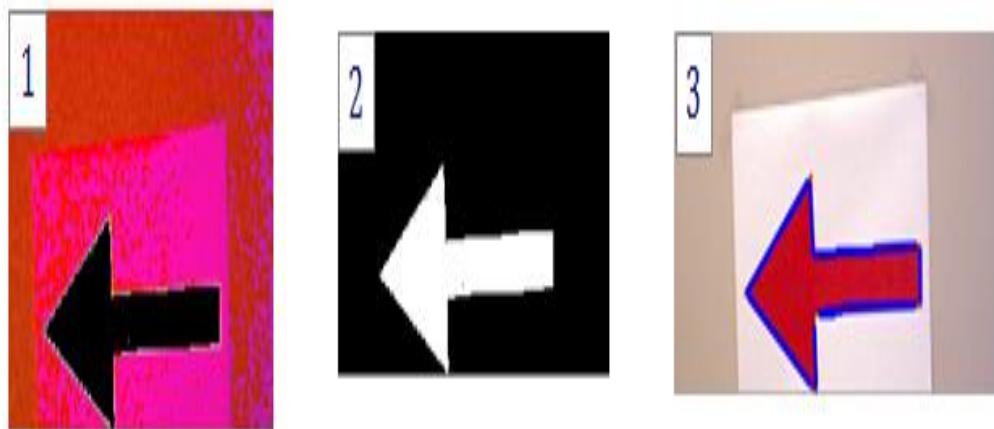
รูปที่ 2.4 ตำแหน่งอุปกรณ์ Sensors ของ AR. Drone 2.0 (Dcview.com, 2010)

2.2.2 AR Drone 2 Object Recognition with EmguCV

Mauro Ursino, Dalla Longa and Borghello (2014) ได้พัฒนาระบบ การบินอัตโนมัติ โดย AR. Drone 2.0 โปรแกรมภาษาภาษา C# ทำงานบนคอมพิวเตอร์ Laptop เป้าหมายจะถูกตรวจสอบโดยใช้หลักการณ์ EmguCV เป็น Library ของ OpenCV ในการประมวลผลภาพ การทำงานคือเมื่อ AR. Drone มองเห็นภาพ จะออกคำสั่งให้บินขึ้น จากนั้นคำสั่งถัดมาจะอยู่ในภาพต่อไป สั่งให้บินไปข้างหน้า เลี้ยว หรือหยุด เมื่อเจอคำสั่งภาพต่อไป

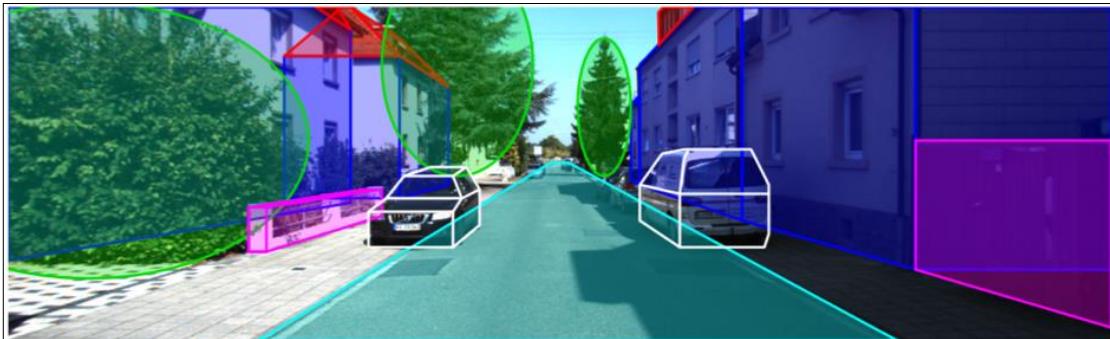


รูปที่ 2.5 การเคลื่อนที่อัตโนมัติของ AR. Drone : โดยการอ่านคำสั่งจากภาพ



รูปที่ 2.6 ตัวอย่างคำสั่งภาพในการเคลื่อนที่ของ AR. Drone เมื่อเข้าสู่โปรแกรม

2.3 คอมพิวเตอร์วิทัศน์ (Computer Vision)



รูปที่ 2.7 การมองเห็นของคอมพิวเตอร์ (Visual Scene Understanding, 2013)

ชุติสันต์ เกิดวิบูลย์เวช (2554) ได้ให้ความหมายว่า Computer หมายถึง คอมพิวเตอร์ทั่วไป Vision เป็นคำนาม แปลว่า ความสามารถในการมองเห็นภาพหรือเข้าใจภาพ เมื่อเอาสองคำมาบูรณาการกัน ในภาษาไทยเรียกว่าย่างๆ ว่า คอมพิวเตอร์วิทัศน์ ซึ่งก็คือเทคโนโลยีการทำให้คอมพิวเตอร์มีความสามารถในการมองเห็นภาพหรือเข้าใจภาพ หมายความว่าอย่างไร อธิบายเป็นหลักการง่ายๆ ก็คือเวลามนุษย์เราใช้ตามองไปที่จากๆ หนึ่ง เช่น เรามองไปที่วิวในห้อง เราสามารถรับรู้ได้เลยว่า นี่คืออะไร เก้าอี้ ตู้เย็น หรือล้าเรามองไปที่ถนน เราจะรับรู้ได้ว่ามีรถวิ่งอยู่ มีคนเดินอยู่ที่ทาง เดิน ที่เรา_rับรู้ได้ เพราะว่าเรามีตาในการมอง และมีสมองในการประมวลผลว่า นี่คือวัตถุ ต่างๆ แต่สำหรับคอมพิวเตอร์ มันไม่ง่ายอย่างนั้น เพราะฉะนั้นจึงเกิดศาสตร์ด้านวิศวกรรมศาสตร์คอมพิวเตอร์ขึ้นมาใหม่ ที่เราเรียกว่า Computer Vision นั่นก็คือการทำให้คอมพิวเตอร์ได้รับรู้ภาพหรือมองเห็นภาพได้เหมือนมนุษย์ หรือกล่าวอีกนัยหนึ่ง ก็คือการทำให้คอมพิวเตอร์สามารถทำการดึงข้อมูลสารสนเทศจากรูปภาพอุปกรณ์นั้นเอง

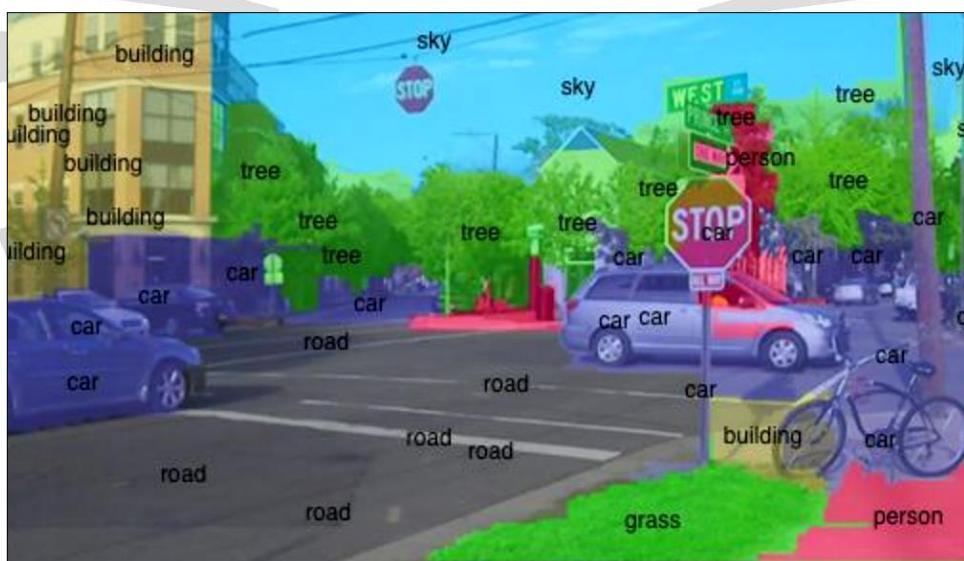
หลักการง่ายๆ ก็คือ ก่อนอื่นต้องติดตามให้กับคอมพิวเตอร์ ซึ่งตากของคอม พิวเตอร์จะเป็นอะไรไม่ได้นอกจากกล้องนั้นเอง กล้องนั้นจะเหลาเปรียบได้เหมือนตาของมนุษย์เรา และสิ่งถัดมา ก็คือเราจะต้อง สอนให้คอมพิวเตอร์รู้จักคำจำกัดความของวัตถุต่างๆ ซึ่งขั้นตอนนี้ก็เปรียบได้เหมือนกับสอนของมนุษย์

ยกตัวอย่างเช่น ที่เราเข้าใจและตี ความภาพต่างๆ ได้ว่า อันนี้คือโต๊ะ ตู้เย็น รถยนต์ ก็ เพราะเรามีสมองคอยประมวลผล ว่าโต๊ะ ควรจะมีขาโต๊ะ มีที่วางของ ส่วนตู้เย็นจะเป็นตู้สี่เหลี่ยม มีที่เปิด รถยนต์ควรจะมีล้อรถสี่ล้อ มีประตู มีตัวถัง มีฝากระโปรงรถ ซึ่งสมองของเราระบุ แล้วเรา ก็รับรู้เรื่องพวกรึม่าตั้งแต่เด็กๆ ว่านี่แหล่ะ ก็คือ วัตถุต่างๆ พวกรุ่นนี้ ซึ่งความยากของคอมพิวเตอร์วิทัศน์ก็คือทำอย่างไรให้คอมพิวเตอร์เข้าใจได้เหมือนหรือตีกว่าสมองมนุษย์ที่สามารถเข้าใจภาพต่างๆ ได้

เครื่องมือที่ใช้ใน Computer Vision หรือสมองที่เราจะใส่ให้คอมพิวเตอร์ หลักๆ เลยจะหนีไม่พ้นองค์ความรู้ทางคณิตศาสตร์ ออาทิ เรขาคณิต พิชคณิตเชิงเส้น สถิติ การวิเคราะห์เชิงฟังก์ชัน (Functional analysis) และการหาค่าเหมาะสมที่สุด (Optimization) โดยเครื่องมือเหล่านี้ใช้ในการสร้างขั้นตอน วิธีต่างๆ ในการแยกองค์ประกอบของภาพ แยกส่วนของภาพ และการจัดกลุ่มภาพเพื่อให้คอมพิวเตอร์สามารถเข้าใจภาพนั้นๆ ได้



รูปที่ 2.8 สิ่งที่มนุษย์มองเห็นและเข้าใจได้ (Phys.org, 2014)



รูปที่ 2.9 สิ่งที่คอมพิวเตอร์มองเห็นและวิธีการเข้าใจ (Phys.org, 2014)

2.4 การประมวลผลภาพ (Image Processing)

ชนกัทร สังขรัตน์ (2013) กระบวนการประมวลผลภาพเป็นอีกเทคโนโลยีที่ปัจจุบันได้มีการประยุกต์ใช้งานในกระบวนการผลิตแบบอัตโนมัติ แรกเริ่มกระบวนการประมวลผลภาพถูกใช้ในห้องปฏิบัติการ เพื่ออำนวยความสะดวกในการวิเคราะห์สิ่งที่มีขนาดเล็ก เช่น การวิเคราะห์ขนาด และรูปร่างของเกรนในเนื้อโลหะ ต่อมาระบบการประมวลผลภาพได้ถูกพัฒนาอย่างต่อเนื่องจนในปัจจุบันมีการนำกระบวนการประมวลผลภาพไปใช้อย่างแพร่หลาย เช่น การจำแนกบุคคลโดยการจดจำใบหน้า, การตรวจจับความเร็วของรถ, การตรวจจับผู้บุกรุก, การประยุกต์ใช้งานทางการแพทย์และการประยุกต์ใช้งานในภาคอุตสาหกรรม



รูปที่ 2.10 ขั้นตอนการประมวลผลภาพ (Image Processing)

Image processing ทำงานโดยการนำภาพที่ได้จากการถ่ายภาพมาใส่ฟังก์ชัน และสมการทางคณิตศาสตร์ที่ต้องการเข้าไป เพื่อให้ได้ผลตามที่เราต้องการ โดยแบ่งการทำงานออกเป็น 3 ส่วน คือ

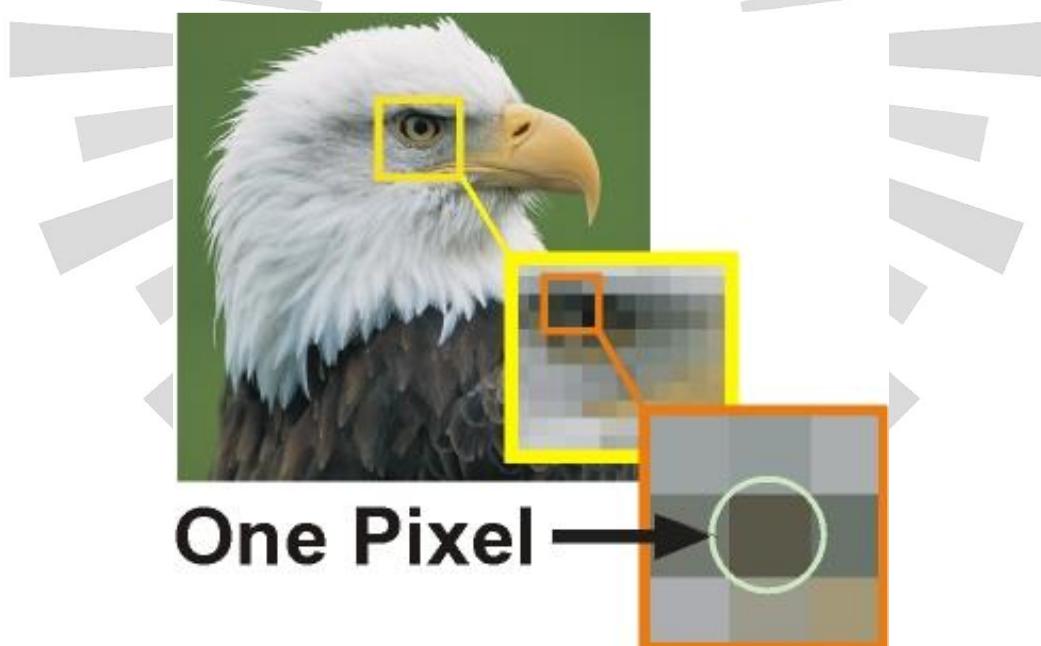
1. อุปกรณ์รับภาพ ทำหน้าที่จับภาพที่ต้องการเข้าสู่คอมพิวเตอร์ เช่น กล้อง Video กล้อง CCD เครื่อง Scanner และรวมไปถึง Capture card สำหรับแปลงสัญญาณจากกล้อง Analog เข้าสู่คอมพิวเตอร์
2. อุปกรณ์ในการประมวลผล ทำหน้าที่ประมวลผลภาพที่รับมาจากอุปกรณ์รับภาพโดยการคำนวณทางคณิตศาสตร์ เพื่อให้ได้ผลลัพธ์ตามที่เราต้องการ อุปกรณ์ที่ใช้ในการประมวลผลอาจอยู่ในรูปคอมพิวเตอร์ ไมโครคอนโทรลเลอร์ หรือเป็นคอนโทรลเลอร์สำหรับแปลงสัญญาณ โดยจัดเป็นชุดพร้อมใช้งานประกอบด้วย กล้อง Capture Card คอนโทรลเลอร์ และอุปกรณ์เพื่อต่อไปยัง Output
3. โปรแกรมประมวลผลภาพ ถือเป็นหัวใจหลักในกระบวนการประมวลผลภาพ เป็นตัวจัดการวิธีการทางกับภาพที่รับมา มีการปรับ-ปรุงคุณภาพของภาพก่อนการประมวลผล การใช้อัลกอริทึม และฟังก์ชันทางคณิตศาสตร์เพื่อให้ได้ Output ตามที่เราต้องการ

โปรแกรมที่ใช้ในการประมวลผลภาพมีหลายประเภททั้งแบบที่สามารถ Download มาใช้กันแบบฟรี ๆ เช่น OpenCV, OpenVIDIA ระบบกระบวนการประมวลผลภาพที่ดีจะต้องประกอบด้วย อุปกรณ์รับภาพที่เหมาะสม มีความละเอียด และความเร็วในการจับภาพเพียงพอกับงานที่ใช้ อุปกรณ์ที่ใช้ในการประมวลผล มีความเร็ว และความสามารถในการคำนวณ โปรแกรมจะต้องจัดการภาพอย่างอย่างเหมาะสม และเป็นระเบียบ รวมถึงสภาพแวดล้อมต่าง ๆ ในการทำงาน เช่น แสงจากภายนอกต้องไม่ไปรบกวนการทำงานของระบบ ขนาด ความยาวโฟกัสของเลนส์กล้อง และระบบแสงสว่างที่เหมาะสม

ทฤษฎีการประมวลผลภาพที่เกี่ยวข้อง

2.4.1 ความหมายของพิกเซล (Pixel)

พิกเซล (Pixel) คือ ความเข้มแสงที่รวมกันทำให้เกิดเป็นภาพ ภาพหนึ่ง ๆ จะประกอบด้วยพิกเซลมากมาย ซึ่งภาพแต่ละภาพที่สร้างขึ้นจะมีความความหนาแน่นของพิกเซลเหล่านี้แตกต่างกันไป ความหนาแน่นนี้เป็นตัวบอกรถึงความละเอียด (Resolution) ของภาพซึ่งมีหน่วยเป็น ppi (Pixel Per Inch) คือจำนวนพิกเซลต่อนิ้ว ซึ่งโดยทั่วไปลือว่า ภาพที่มีความละเอียดสูงหรือคุณภาพดีจะมีความละเอียด 300 x 300 ppi ขึ้นไป ค่า ppi ยิ่งสูงขึ้น ภาพก็จะมีความละเอียดและคมชัดมากขึ้น

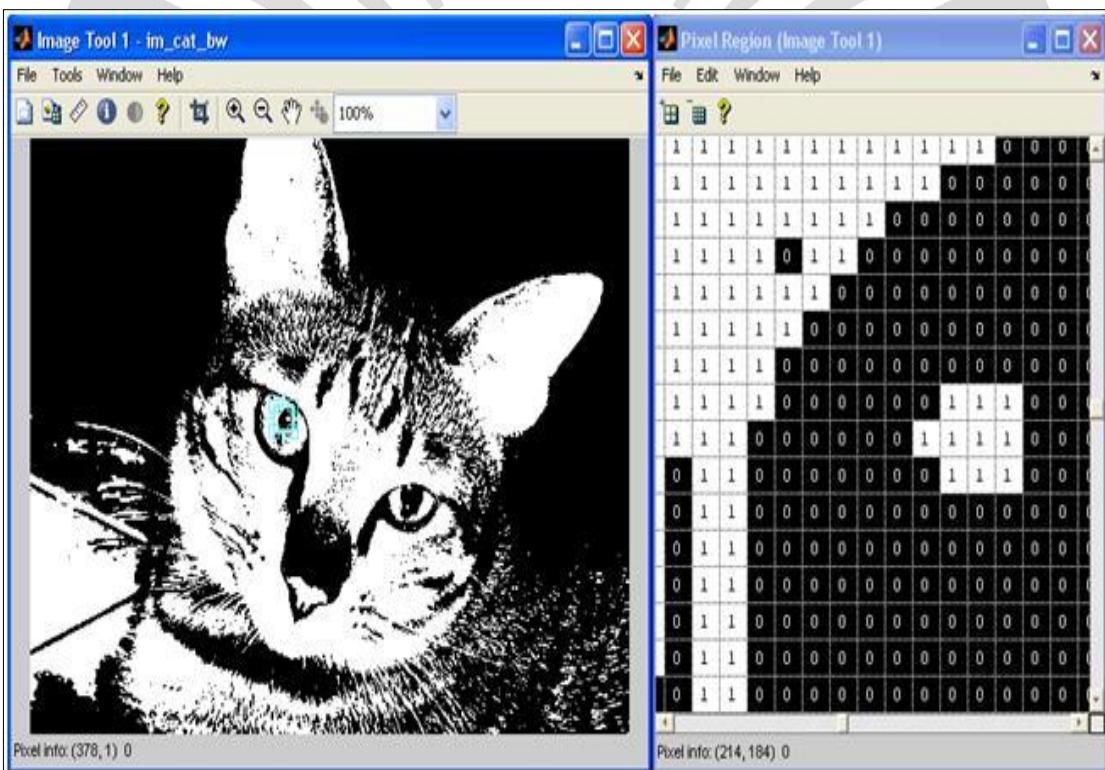


รูปที่ 2.11 ตำแหน่งของ Pixel (Digital Safari, 2015)

2.4.2 การสร้างภาพ Binary

Binary ในทางดิจิตอลหมายความว่า ภาพจะมีแค่ความเข้ม 2 ค่าเท่านั้น คือ 0 และ 1 หมายความว่า Pixel ใดมีค่าเป็น 0 Pixel เป็นสีดำ Pixel 1 สีขาว การสร้างภาพ Binary สามารถได้โดยใช้เทคนิคการทำ เทρชโซล (Thresholding Technique) โดยพิจารณาว่า พิกเซลสีขาวหรือสีดำ จะกระทำโดยการเปรียบเทียบระหว่างพิกเซลของภาพเริ่มต้นกับ ค่าคงที่หนึ่งเรียกว่า ค่าเทρชโซล (Threshold Value) เทคนิคนี้ใช้กันมากในกรณีที่ ข้อมูลภาพมีลักษณะที่ต่างกันระหว่างวัตถุ (Object) และพื้นหลัง (Background) โดยค่าของ พิกเซลของภาพใดๆ ที่มีค่ามากกว่าหรือเท่ากับค่าเทρชโซลจะถูกเปลี่ยนเป็น 0 (จุดดำ) และ พิกเซลของภาพที่มีน้อยมากกว่าหรือเท่ากับค่าเทρชโซลจะถูกเปลี่ยนเป็น 1 (จุดขาว) เป็นต้น

ในการสร้างภาพใบหน้าโดยใช้เทคนิคเทρชโซลเพื่อให้ได้ผลลัพธ์ที่เหมาะสมและ คมชัด สิ่งที่สำคัญที่สุด คือ ค่าเทρชโซล เนื่องจากถ้าเลือกค่าเทρชโซลที่ไม่เหมาะสม (ค่าเทρชโซลที่มีค่าน้อยเกินไปหรือมากเกินไป) ภาพที่ได้อาจจะสว่างน้อยเกินไป หรือสว่างมากเกินไป หรือภาพที่ได้มีสิ่งรบกวน (Noise) เกิดขึ้น อันเป็นผลทำให้ภาพที่ได้ไม่สวยงามเท่าที่ควร

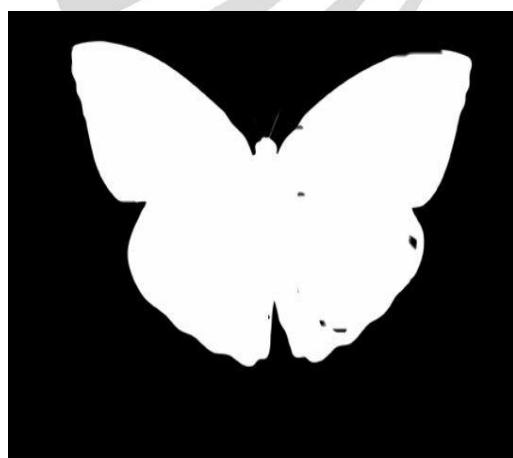
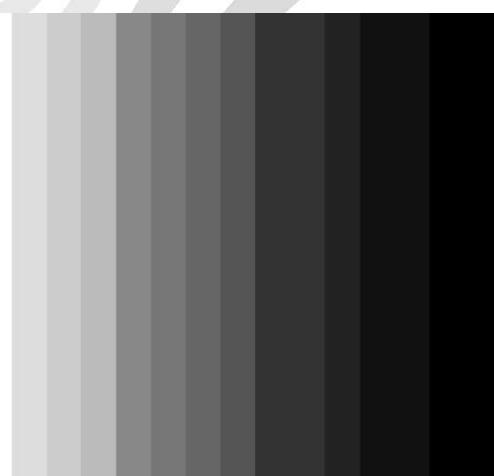
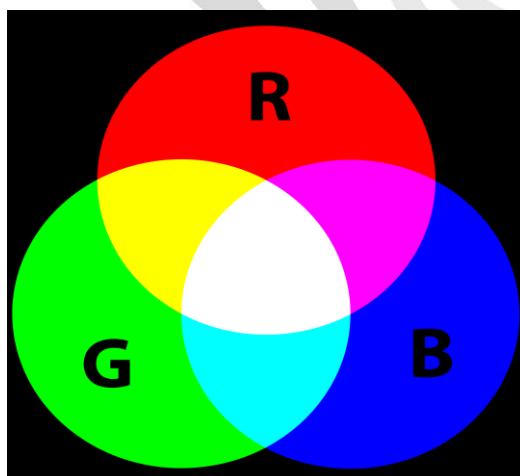


รูปที่ 2.12 ตัวอย่างการปรับค่า Threshold (Stanford.edu, 2000)

2.4.3 แบบจำลองสี RGB, Grayscale และ Alpha Channel

เป็นแบบจำลองที่เฉพาะเจาะจงกับภาพคอมพิวเตอร์ เนื่องจาก RGB Model ได้ทำการสร้างสีต่าง ๆ ขึ้นโดยการใช้เหล่งกำเนิดแสดงจำนวนสามสี ได้แก่ สีแดง (Red) สีเขียว (Green) และสีน้ำเงิน (Blue) ที่เกิดจากการเรืองแสงที่มีคุณสมบัติที่แตกต่างกันตามลำดับ ซึ่งแสงทั้งสามสีจะไม่เท่ากันในแต่ละอุปกรณ์ นอกเสียจากว่ามีคุณสมบัติของสารเรืองแสงและการตั้งค่าจօภาพ และสภาพแวดล้อมที่จօภาพคอมพิวเตอร์เหมือนกันทุกประการ ซึ่งโดยปกติแล้วจะมีค่าที่แตกต่างกันออกไป

ในการประมวลผลภาพ นอกจาก RGB แล้วยังมี Grayscale Color และ Alpha Channel ซึ่งมีลักษณะของการใช้สีแตกต่างกันออกไป Grayscale โดย dni จะมีเพียงสองสีคือ สีขาวและสีดำแต่จะมีระดับความเข้มของสี 255 ระดับ ส่วน Alpha Channel ใช้ในการปรับความโปร่งใส หรือความทึบแสงของภาพนั้น



รูปที่ 2.13 แบบจำลองสี (RGB Color)

รูปที่ 2.14 แบบจำลองสี (Grayscale)

รูปที่ 2.15 แบบจำลองสี (Alpha)

2.5 ความรู้เบื้องต้นเกี่ยวกับ OpenCV



รูปที่ 2.16 Logo OpenCV (Opencv.org, 2015)

OpenCV หรือ Open Source Computer Vision Library ใช้ในการประมวลผลภาพและงานทางด้าน การมองเห็นของคอมพิวเตอร์ Computer Vision Library นี้ถูกพัฒนาขึ้นด้วยภาษา C# และ C++ สามารถ Run ได้บนทั้ง Linux , Mac OSX และ Windows นอกจากนั้นยังมี Interface ที่ไว้เชื่อมต่อกับ Tool อื่นด้วย เช่น Python, Ruby, Mat lab เป็นต้น นอกจากนั้นแล้ว Library ยังถูกเขียนและมีการ Optimize Scheme ซึ่งทำให้ตัว Library ทำงานได้อย่างมีประสิทธิภาพ และยังรองรับ และสนับสนุนการทำงานแบบ Multi - Core Processors จุดเด่นอีกอย่างของ OpenCV คือเป็น Library ที่สร้างขึ้นเพื่อให้ผู้ใช้หรือนักพัฒนาสามารถใช้ฟังก์ชันใน Library ซึ่งมีมากกว่า 500 ใน การพัฒนาขึ้นงานที่มีความซับซ้อน โดยใช้เวลาในการพัฒนาไม่นาน

- Data Structure ใช้เก็บข้อมูลต่าง ๆ อาทิ เช่น รูปภาพ เมทริกซ์ พิกัด
- Algorithm เพื่อการประมวลผลต่าง โดยเฉพาะการประมวลผลทางรูปภาพ

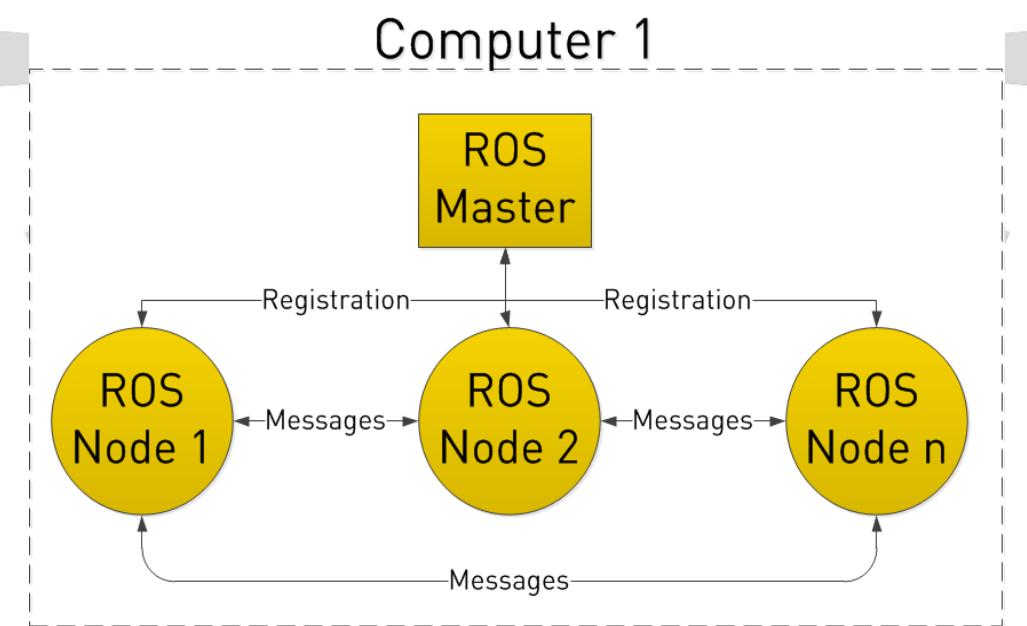
ภาณุพงษ์ เพชรเลิศ (2009) กล่าวว่า การใช้งานฟังก์ชันต่างๆของ OpenCV จะต้องมีการเขียนโปรแกรมที่เรียกว่าไฟล์ส่วนหัว (Header File) เพื่อเข้าถึง Library หรือ Function ต่างๆ ที่บรรจุอยู่ในไฟล์ DLL (Dynamic Link Library) โดยสร้างหลักประกอบด้วย 4 ส่วนคือ (1) CvCore ประกอบด้วยโครงสร้างข้อมูลต่างๆ ได้แก่ Tree List Queue Sequence Image และ Function สำหรับจัดการข้อมูลตั้งกล่าว (2) CvReference ประกอบด้วย Function สำหรับการวิเคราะห์และประมวลผลภาพ (3) CvAux ประกอบด้วย Function ที่อยู่ในระหว่างการพัฒนาหรือทดสอบ รวมถึง Function ที่เลิกใช้แล้ว (4) High-GUI สำหรับการแสดงผลภาพ การรับภาพ และการบันทึกไฟล์ภาพ

2.6 ระบบปฏิบัติการหุ่นยนต์ (ROS)



รูปที่ 2.17 Logo ROS industrial (Robotiq.com, 2014)

ROS (Robot Operating System) เป็นระบบปฏิบัติการคล้ายกับ Window หรือ Linux แต่เป็นระบบปฏิบัติการสำหรับหุ่นยนต์ เป็น Middleware ที่มี Library และ Tool สำหรับการทำหุ่นยนต์ไว้ค่อนข้างครบ มีทั้ง Driver ของอุปกรณ์ต่างๆ รวมไปถึง Algorithm และที่สำคัญเป็น Open Source ภาษาที่ใช้ในการเขียนโปรแกรมจะใช้ C++ และ Python โครงสร้างโปรแกรมจะเป็นลักษณะการทำงานแบบ Node ซึ่งแต่ละ Node จะทำงานตามหน้าที่ของตนเอง แต่ละ Node จะติดต่อสื่อสารกันผ่าน Topic และ Services



รูปที่ 2.18 การทำงานของ ROS (Student. Roboclub, 2014)

2.7 อากาศยานไร้คนขับ AR. Drone 2.0



รูปที่ 2.19 Parrot AR. Drone 2.0 (Parrot.com, 2014)

AR. Drone 2.0 คืออากาศยานไร้คนขับ วิทยุบังคับ มีสีใบพัด ผลิตโดยบริษัท Parrot เป็นอากาศยานไร้คนขับที่ถูกออกแบบมาให้ควบคุมผ่านสมาร์ทโฟนหรือแท็บเล็ต ซึ่งสนับสนุนระบบปฏิบัติการทั้ง Android และ IOS โดย รุ่น 2.0 ถูกพัฒนาให้กล้องในตัว มีประสิทธิภาพดีขึ้น ซึ่งอากาศยานไร้คนขับนี้ติดตั้งกล้องไว้ 2 ตัว บริเวณด้านหน้า 1 ตัว และบริเวณด้านล่างอีก 1 ตัว โดยกล้องหน้ามีความละเอียด 720p ส่วนกล้องล่างมีความละเอียด 360p อีกทั้งยังพัฒนาให้สามารถควบคุมได้ง่ายขึ้น สามารถสื่อสารระหว่างเครื่องได้โดยตรงผ่านเทคโนโลยี Wi-Fi

คุณสมบัติ AR. Drone 2.0 (Technical Specifications, 2013)

- 1 GHz 32 Bit ARM Cortex A8 Processor With 800MHz Video DSP
- Linux 2.6.32
- 1 GB bit DDR2 RAM At 200MHz
- USB 2.0 High Speed For Extensions
- Wi-Fi B.G.N
- HD Camera. 720p 30fps
- Can Fly As Far As 165 Feet Awa

บทที่ 3

ขั้นตอนวิธีการออกแบบและพัฒนาระบบ

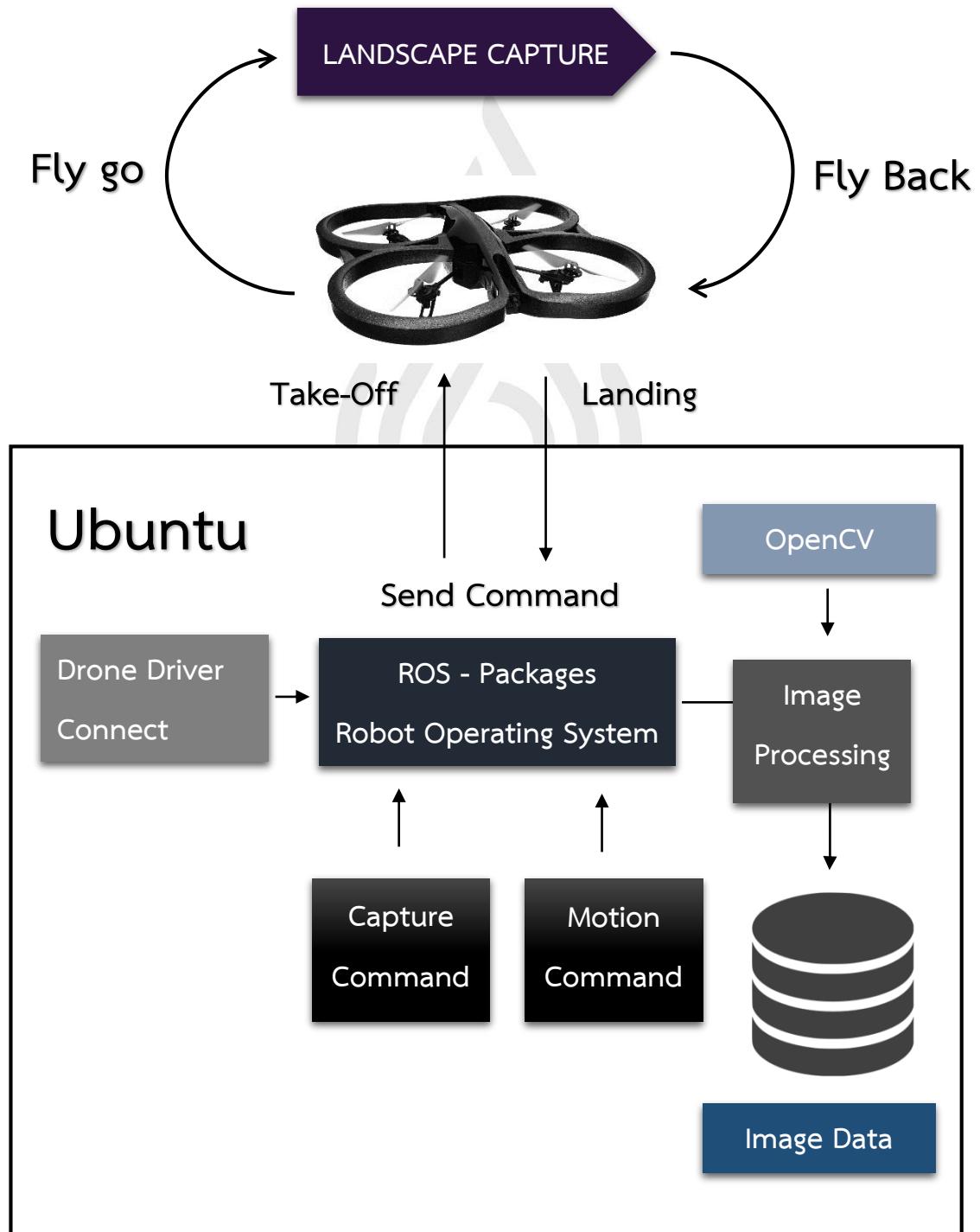
ในบทนี้จะกล่าวถึงการออกแบบระบบ การสำรวจความเปลี่ยนแปลงภูมิทัศน์โดยอากาศยาน
ไร้คนขับ เพื่อศึกษาความเป็นไปได้ในการพัฒนาระบบ

แบ่งออกเป็น 4 ตอน คือ

- 3.1 ออกแบบสถาปัตยกรรมระบบ
- 3.2 วิธีการการถ่ายภาพภูมิทัศน์
- 3.3 วิธีการหาความเปลี่ยนแปลงภูมิทัศน์
- 3.4 ข้อกำหนดเกี่ยวกับเทคโนโลยีที่ใช้พัฒนาระบบ

ทั้ง 4 ตอน นี้รายละเอียดดังนี้

3.1 ออกแบบสถาปัตยกรรมระบบ



รูปที่ 3.1 System Architecture design

จากการออกแบบสถาปัตยกรรมระบบ แนวคิดการทำงานของระบบ เริ่มต้นจากการเชื่อมต่อ อากาศยานไร้คนขับกับคอมพิวเตอร์ ผ่าน WI-FI จากนั้นใช้ ROS เปิดระบบให้อากาศยานไร้คนขับ เริ่มทำงาน การทำงานของระบบเริ่มต้นจากส่งคำสั่ง ควบคุม และ คำสั่งถ่ายภาพ ผ่าน ROS ให้อากาศยาน บินขึ้น และหันทิศทางไปยังตำแหน่งที่เราต้องการ จากนั้นบินไปข้างหน้าจนถึงระยะที่ต้องการ หยุดและถ่ายภาพภูมิทัศน์ หนึ่งครั้ง จากนั้นหันหลัง และบินกลับในทิศทางเดิม ลงจอดที่เดิม ส่วนภาพถ่ายภูมิทัศน์จะถูกจัดเก็บลงในโฟลเดอร์เรียบร้อย หนึ่งภาพ พร้อมบันทึกวันที่และเวลา จากนั้น สั่งให้อากาศยานไร้คนขับ บินแบบเดิมอีกรอบ แต่แก้ไขภูมิทัศน์เดิมให้เกิดความเปลี่ยนแปลง ภาพล่าสุดจะถูกจัดเก็บลงโฟลเดอร์ รวมเป็น 2 ภาพ ที่เกิดความเปลี่ยนแปลงขึ้น จากนั้นโปรแกรมจะทำการประมวลผลภาพ เพื่อหาความเปลี่ยนแปลง เป็นอันเสร็จ ข้อมูลนี้เป็นเพียงแนวคิดเบื้องต้นในการออกแบบโปรแกรมเท่านั้น

3.2 วิธีการการถ่ายภาพภูมิทัศน์

การสำรวจความเปลี่ยนแปลงภูมิทัศน์ นั้นหมายถึง ภูมิทัศน์ที่ทำการสำรวจต้องเกิดความเปลี่ยนแปลงไปจากเดิม เป็นเหตุผลให้จึงต้องหาทางที่จะหาภาพภูมิทัศน์ ที่เกิดความเปลี่ยนแปลง

เนื่องมาจากการหัวข้อคือภูมิทัศน์ และปัญหาคือ สิ่งแวดล้อม จึงต้องการภาพที่เป็น ป่าไม้ แม่น้ำ ลำคลอง เชื่อม สวน การทำเกษตรกรรม หรืออุตสาหกรรม เป็นต้น จึงจะ Solve ปัญหาได้ดีที่สุด ผู้ผันนา จึงใช้รูปภาพของภูมิทัศน์จริงๆ ที่มีความชัดพอสมควร นำมาทำการปรับแต่งภาพให้เกิดความเปลี่ยนแปลง ยกตัวอย่างภาพ เช่น



รูปที่ 3.2 การตัดไม้ทำลายป่า (ซ้าย) การตัดไม้ทำลายป่าที่เพิ่มขึ้น (ขวา) (Amazon, 2009)



รูปที่ 3.3 ภาพจำลองการใช้อากาศยานไร้คนขับถ่ายภูมิทัศน์จากรูปภาพ

(จากรูปที่ 3.1) จะเห็นได้ว่าภูมิทัศน์เกิดความเปลี่ยนแปลงจริงๆ และดู Solve ปัญหาดีที่สุด เพื่องานหลักของหัวข้อ การถ่ายภาพจึงต้องดูเป็นงานที่มีคุณค่า สามารถนำข้อมูลที่ได้ไป ศึกษา วิเคราะห์ ติดตามผิด หรือแก้ปัญหาต่างๆ ได้อีกด้วย (รูปที่ 3.2) รูปตัวอย่างการนำเอารูปภาพขนาด A4 ให้อากาศยานไร้คนขับถ่าย โดยไม่จำเป็นที่จะต้องเอาอากาศยานไร้คนขับบินไปยังสถานที่จริงๆ การถ่ายภาพเป็นงานรอง งานหลักคือการหาความเปลี่ยนแปลงของภูมิทัศน์ได้

3.3 วิธีการหาความเปลี่ยนแปลงภูมิทัศน์

แนวคิด ในส่วนของวิธีการหาความเปลี่ยนแปลง จะใช้ Library ของ OpenCV ในการ ประมวลผลภาพ เพราะผู้พัฒนา เคยได้ศึกษาในรายวิชา Computer Vision และมีความสนใจใน Image Processing จึงมองเห็นแนวทางความเป็นไปได้ในการหาความแตกต่างของภาพ ได้ โดยมี หลักการต่างๆ มากมายที่ใกล้เคียงกับระบบบันทึก น่าจะเพียงพอที่จะ นำทฤษฎีหรือ พังก์ชั่น ต่างๆ มา ผสมผสานกันจนเกิดเป็นสิ่งใหม่ และเห็นแนวทางความสำเร็จ

เบื้องต้นการออกแบบวิธีการหาความเปลี่ยนแปลงภูมิทัศน์ ผู้พัฒนาได้ออกแบบ โปรแกรมว่า จะมีขั้นตอนการทำงานดังต่อไปนี้ (รูปที่ 3.4) กรรมวิธีนี้เป็นเพียงแนวคิดเบื้องต้นของการหาความเปลี่ยนแปลง ยังมิได้พัฒนาระบบท่อไปได้ ยังอยู่ในขั้นตอนการศึกษาความเป็นไปได้ต่อไป

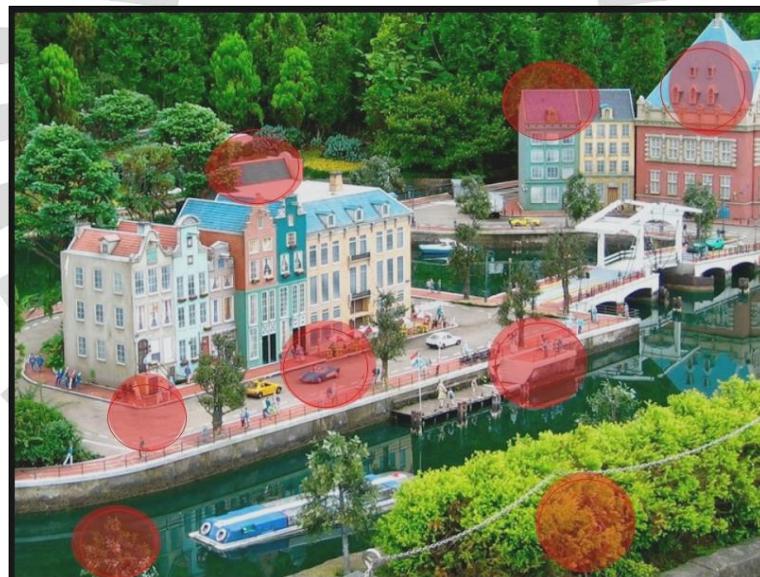
การหาความเปลี่ยนแปลง ก็เหมือนกับการหาความแตกต่าง ผู้พัฒนาระบบนิเกลิงเกมจับผิดภาพ เกมฝึกสมองที่ถ้าไม่สังเกตดี ก็จะไม่สามารถหาส่วนที่แตกต่างของภาพได้เลย เพราะ มีส่วนที่แตกต่างกันเต็มไปหมด ในที่นี้ ผู้พัฒนาขอยกตัวอย่างเกมจับผิดภาพ เป็นแนวคิดในการหาความเปลี่ยนแปลงภูมิทัศน์ หรือจากรูปภาพ มีขั้นตอนดังนี้

1. นำเข้ารูปภาพ 2 รูปที่มีความแตกต่างกัน เพื่อเข้าสู่การเปรียบเทียบภาพ



รูปที่ 3.4 ภาพตัวอย่าง เกมจับผิดภาพ ก่อนประมวลผล : นำเข้า (Krypton Games, 2558)

2. หลังจากโปรแกรมประมวลผล ต้องบอกได้ว่าจุดไหนบ้างคือความแตกต่าง ดังภาพ



รูปที่ 3.5 ภาพตัวอย่าง เกมจับผิดภาพ หลังประมวลผล : นำออก (Krypton Games, 2558)

3.4 ข้อกำหนดเกี่ยวกับเทคโนโลยีที่ใช้พัฒนาระบบ

ในการพัฒนาระบบในครั้งนี้ ผู้พัฒนาระบบได้กำหนดข้อตกลงพื้นฐาน หรือ Specification ของ Hardware ที่ใช้ หรือ ข้อกำหนดเกี่ยวกับ Software ในการออกแบบพัฒนาระบบไว้ ดังนี้

3.4.1 Hardware ที่ใช้ในการพัฒนาระบบ

- Parrot AR. Drone Version 2.0
- Logitech HD Webcam C270

3.4.2 Software ที่ใช้ในการพัฒนาระบบ

- OpenCV Library Version 2.4.9
- Ubuntu Operating System 12.04 LTS 32 Bit
- Robot Operating System (ROS) รุ่น Fuerte Turtle
- C++ Programming Language

เป็นเครื่องมือพื้นฐานในการเริ่มต้นพัฒนาระบบ

บทที่ 4

การสำรวจความเปลี่ยนแปลงภูมิทัศน์โดยอากาศยานไร้คนขับ

ในการพัฒนาระบบ ผู้พัฒนาระบบได้ใช้เครื่องมือดังอธิบายรายละเอียดไว้แล้วในหัวข้อ 3.4 เป็นเครื่องมือหลักในการพัฒนาระบบตามขั้นตอนวิธีการออกแบบที่กำหนดไว้ (รายละเอียดในบทที่ 3) แบ่งออกเป็น 3 ตอน คือ

4.1 การพัฒนาระบบ

- 4.1.1 ภาพถ่ายภูมิทัศน์ที่เปลี่ยนแปลง
- 4.1.2 อธิบาย Package ROS
- 4.1.3 อธิบาย Source Code Program

4.2 การทำงานของระบบ

- 4.2.1 ถ่ายภาพภูมิทัศน์
- 4.2.2 หาความเปลี่ยนแปลงของภาพถ่ายภูมิทัศน์
- 4.2.3 ปรับภาพ Transparent, Threshold & Alpha
- 4.2.4 วางช่องภาพ Overlay
- 4.2.5 ผสมผสานภาพ Blend
- 4.2.6 เรียกใช้ Bash Script

4.3 การทดสอบระบบ

- 4.3.1 ทดสอบความซัดของรูปภาพภูมิทัศน์
- 4.3.2 ทดสอบความคลาดเคลื่อนของภูมิทัศน์
- 4.3.3 สรุปผลการทดสอบระบบ

ทั้ง 3 ตอน มีรายละเอียดดังนี้

4.1 การพัฒนาระบบ

4.1.1 ภาพถ่ายภูมิทัศน์ที่เปลี่ยนแปลง

เนื่องจากการสำรวจความเปลี่ยนแปลงภูมิทัศน์ ต้องใช้ภาพที่เกิดความเปลี่ยนแปลงจริง ผู้พัฒนาจึงได้จำลอง ด้วยการนำภาพถ่ายทางอากาศมาปรับแต่ง ให้เกิดความเปลี่ยนแปลงในสถานการณ์ต่างๆ 3 แบบจำลองสถานการณ์ด้วยกัน ดังนี้

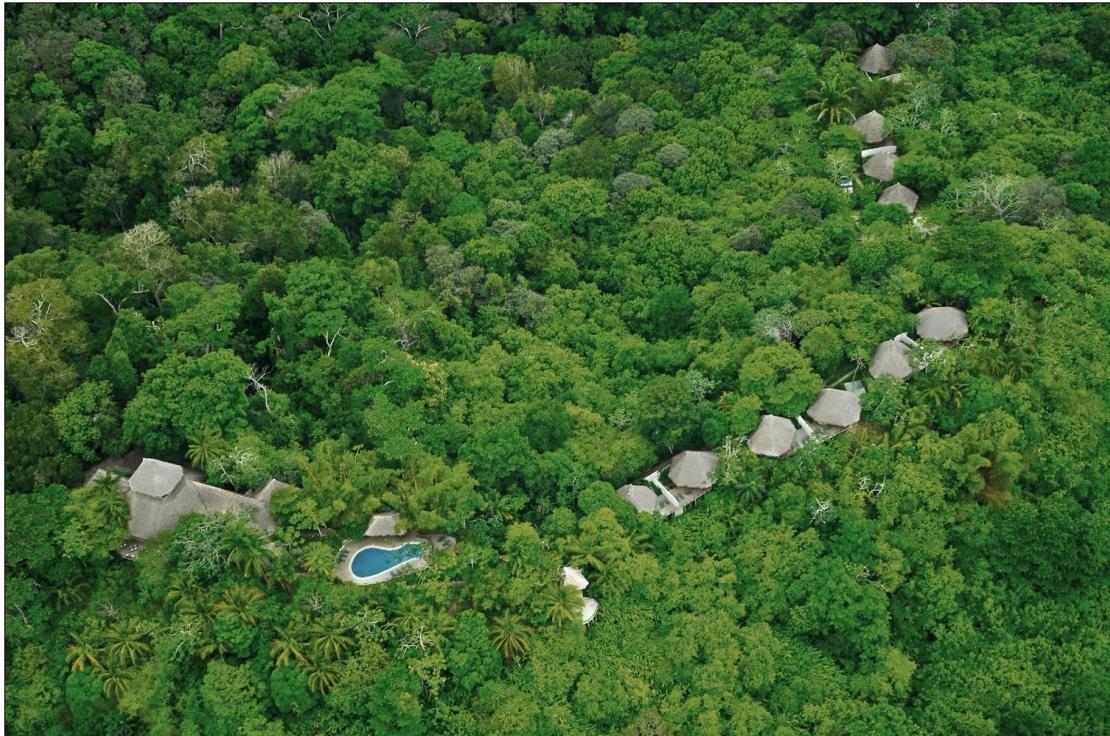
4.1.1.1 ภาพภูมิทัศน์จำลองสถานการณ์ สิ่งปลูกสร้าง

4.1.1.2 ภาพภูมิทัศน์จำลองสถานการณ์ ตัดไม้ทำลายป่า

4.1.1.3 ภาพภูมิทัศน์จำลองสถานการณ์ การกัดเซาะของแม่น้ำ



รูปที่ 4.1 ภาพภูมิทัศน์จำลองสถานการณ์ สิ่งปลูกสร้าง : ก่อน



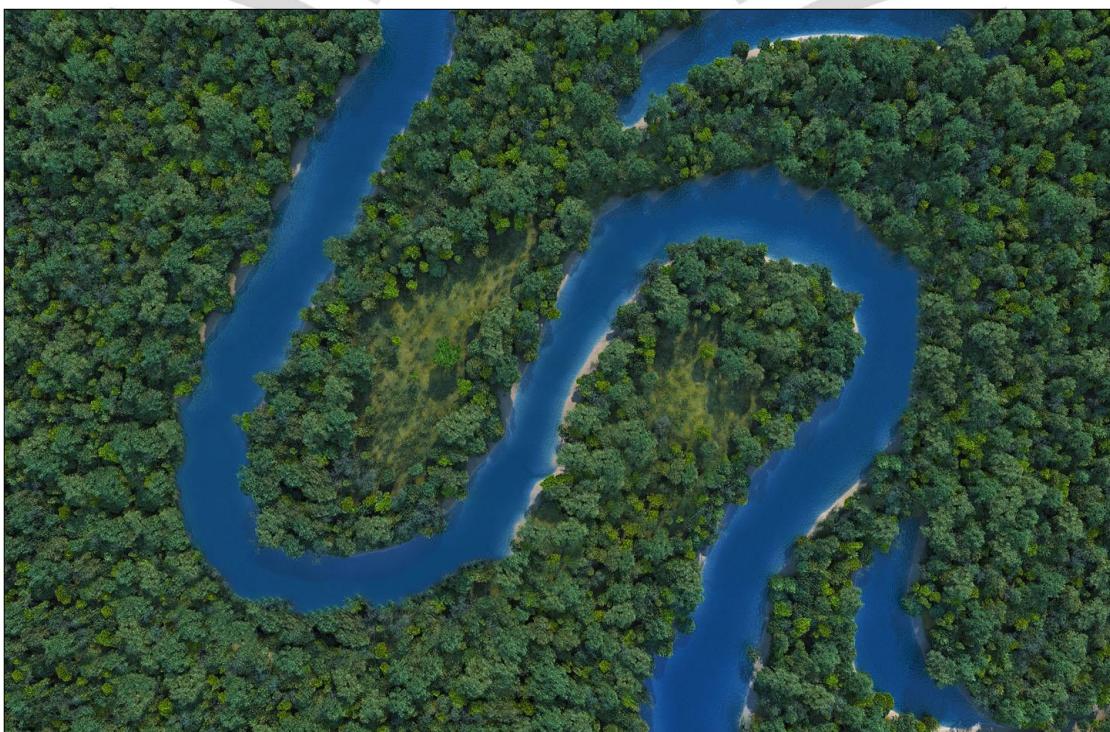
รูปที่ 4.2 ภาพภูมิทัศน์จำลองสถานการณ์ สิ่งปลูกสร้าง : หลัง



รูปที่ 4.3 ภาพภูมิทัศน์จำลองสถานการณ์ ตัดไม้ทำลายป่า : ก่อน



รูปที่ 4.4 ภาพภูมิทัศน์จำลองสถานการณ์ ตัดไม้ทำลายป่า : หลัง

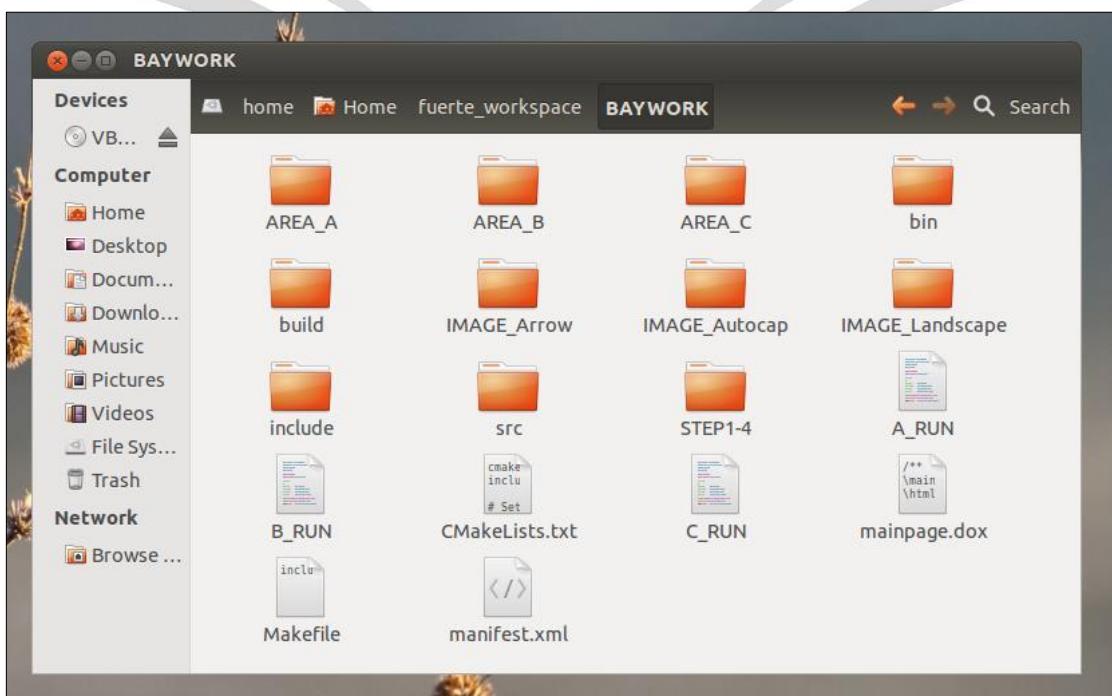


รูปที่ 4.5 ภาพภูมิทัศน์จำลองสถานการณ์ การกัดเซาะของแม่น้ำ : ก่อน



รูปที่ 4.6 ภาพภูมิทัศน์จำลองสถานการณ์ การกัดเซาะของแม่น้ำ : หลัง

4.1.2 อธิบาย Package ROS



รูปที่ 4.7 Folder Package ROS BAYWORK

ภายใน Workspace ของ ROS คือส่วนที่ใช้ในการ Build และ Run โปรแกรมซึ่งทั้งหมดทำงานภายใต้ระบบปฏิบัติการทุนยนต์ ส่วนสำคัญคือ (รูปที่ 4.11) Folder src ใช้เก็บ Source Code ของโปรแกรม , (รูปที่ 4.8) manifest.xml จะเป็นตัวกำหนด Library ที่ใช้ในการ Build โปรแกรม , (รูปที่ 4.9) CMakeList.txt ใช้ Add ชื่อไฟล์ Source Code ที่ต้องการ เพื่อ Build เป็น Executable ไฟล์ คำสั่งในการ Build คือ “rosmake” และจะถูกจัดเก็บไว้ใน (รูปที่ 4.10) Folder bin

```
-<package>
<description brief="BAYWORK"> BAYWORK </description>
<author>BAY</author>
<license>BSD</license>
<review status="unreviewed" notes="" />
<url>http://ros.org/wiki/BAYWORK</url>
<depend package="std_msgs"/>
<depend package="rospy"/>
<depend package="roscpp"/>
<depend package="cv_bridge"/>
<depend package="image_transport"/>
<depend package="opencv2"/>
</package>
```

รูปที่ 4.8 manifest.xml

```
rosbuild_add_executable(AUTOCAP src/AUTOCAP.cpp)
rosbuild_add_executable(A_KEYCAP src/A_KEYCAP.cpp)

rosbuild_add_executable(A_DIFF src/A_DIFF.cpp)
rosbuild_add_executable(A_NOBLACK src/A_NOBLACK.cpp)
rosbuild_add_executable(A_OVERLAY src/A_OVERLAY.cpp)

rosbuild_add_executable(SHOW_DIFF src/SHOW_DIFF.cpp)
rosbuild_add_executable(SHOW_NOBLA src/SHOW_NOBLA.cpp)
rosbuild_add_executable(SHOW_OVERLA src/SHOW_OVERLA.cpp)
rosbuild_add_executable(SHOW_BLEND src/SHOW_BLEND.cpp)
```

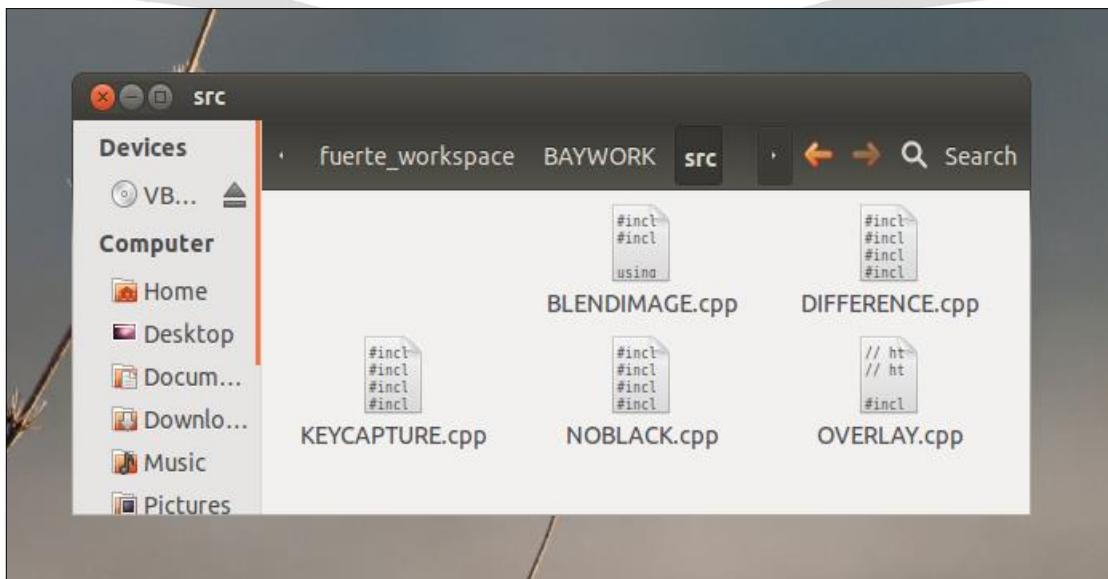
รูปที่ 4.9 CMakeList.txt



รูปที่ 4.10 Folder bin Executable file

4.1.3 อธิบาย Source Code Program

โปรแกรมจะใช้ OpenCV Library ในการประมวลผลภาพ ภาษาที่ใช้เขียนคือ C++ ในส่วนนี้จะอธิบายถึง Source Code หลักๆ ที่ใช้ในการพัฒนาโปรแกรม ว่ามีวิธีการทำงานอย่างไร ใช้ Algorithm อะไรบ้างในการพัฒนาโปรแกรม ซึ่ง Source Code ที่ใช้จะมี การถ่ายภาพ ปรับค่าสีและแสง การผสานภาพ และการซ่อนภาพ จะประกอบไปด้วย 5 ไฟล์ด้วยกัน ดังนี้



รูปที่ 4.11 Folder src Source Code Program

4.1.3.1 Code KEYCAPTURE.cpp (รูปที่ 4.12) เป็นคำสั่งเปิดกล้อง และสร้างตัวแปรเก็บค่า frame รูปภาพ (รูปที่ 4.13) ตั้งคិែិ៍ในการกดถ่ายภาพเอาไว้เป็น Key “S” ซึ่งในนี้จะสร้างตัวแปรสำคัญในการเรียกใช้เวลาปัจจุบัน และเมื่อกดถ่ายภาพจะบันทึกเป็นวันที่และเวลาตามที่เราได้กำหนดไว้ใน Buffer เป็นชื่อของรูปภาพ

```
cvNamedWindow("Camera", 1); //Create window
CvCapture* capture = cvCaptureFromCAM(CV_CAP_ANY); //Capture using any camera connected

while (1)
{ //Create infinite loop for live streaming

    IplImage* frame = cvQueryFrame(capture); //Create image frames from capture
    cvShowImage("Camera", frame); //Show image frames on created window
    key = cvWaitKey(10); //Capture Keyboard stroke
```

รูปที่ 4.12 Source Code Capture 1

```
else if (char(key) == 's')
{
    C++;

    // current date/time based on current system
    time_t now = time(0);
    tm * ltm = localtime(&now);

    // print various components of tm structure.
    cout << "Keycap Count " << c << endl;
    cout << "Date&Time : " << "";
    cout << ltm->tm_mday << "-";
    cout << 1 + ltm->tm_mon << "-";
    cout << 1900 + ltm->tm_year << "_";
    cout << ltm->tm_hour << ":";
    cout << ltm->tm_min << ":";
    cout << ltm->tm_sec << endl;
    cout << "Image Is Save." << endl;
    cout << "_____" << endl;

    ltm->tm_year += 1900;
    ltm->tm_mon += 1;

    sprintf(buffer, "/home/nattagrit/fuerte_workspace/BAYWORK/KEYCAP_%d_%d", ltm->tm_year, ltm->tm_mon, ltm->tm_hour, ltm->tm_min, ltm->tm_sec);
    cvSaveImage(buffer, frame);
```

รูปที่ 4.13 Source Code Capture 2

4.1.3.2 Code DIFFERENCE.cpp หากความแตกต่างของภาพ โปรแกรมจะจัดเก็บ
ภาพความแตกต่าง และวนลูปไปในแต่ละ Pixel ของทั้ง 2 ภาพเพื่อคำนวณหาความแตกต่าง
ในช่อง RGB Channel สามารถปรับค่าสี ที่เป็นผลลัพธ์ได้ตามต้องการ

```
// create a difference image storage
diff = cvCloneImage(img1);
sub = cvCloneImage(img1);

cvZero(diff);

// We go over all pixels of the image
for( int y=0; y<diff->height; y++ ) {
    uchar* ptr1 = (uchar*) (img1->imageData + y * img1->widthStep);
    uchar* ptr2 = (uchar*) (img2->imageData + y * img2->widthStep);
    uchar* ptr = (uchar*) (diff->imageData + y * diff->widthStep);
    for( int x=0; x<diff->width; x++ ) {
        // 3 channel:
        // B
        //ptr[3*x] = ptr1[3*x] + ptr2[3*x] - 2 * min(ptr1[3*x], ptr2[3*x]);
        // G
        //ptr[3*x+1] = ptr1[3*x+1] + ptr2[3*x+1] - 2 * min(ptr1[3*x+1], ptr2[3*x+1]);
        // R
        ptr[3*x+2] = ptr1[3*x+2] + ptr2[3*x+2] - 2 * min(ptr1[3*x+2], ptr2[3*x+2]);
    }
}
```

รูปที่ 4.14 Source Code Difference

4.1.3.3 Code NOBLACK.cpp (รูปที่ 4.16) สร้าง Track bar เพื่อสามารถปรับ
ค่าความสว่าง และลด Pixel ของภาพ และส่งค่ามาทำงานยัง (รูปที่ 4.15) Convert ภาพ
เป็นสีเทา เพื่อแยกสีกับภาพออกจากกันและปรับภาพเป็น Transparent ภาพแบบโปร่งแสง

```
cvtColor(src,tmp,CV_BGR2GRAY);
threshold(tmp, alpha, border, weight, THRESH_BINARY); // Alpha 60 %

Mat rgb[3];
split(src,rgb);

Mat rgba[4]={rgb[0],rgb[1],rgb[2],alpha};
merge(rgba,4,dst);
```

รูปที่ 4.15 Source Code Transparent 1

```
namedWindow( "Transparent");
createTrackbar("Threshold : ", "Transparent", &border, maxborder, binarization );
createTrackbar("Alpha : ", "Transparent", &weight, maxweight, binarization );

binarization(1, 0);
```

รูปที่ 4.16 Source Code Transparent 2

4.1.3.4 Code OVERLAY.cpp (รูปที่ 4.17) เป็น Algorithm ที่นำภาพที่ผ่านการประมวลผลภาพเป็น Transparent หรือ ภาพโปร่งแสงไม่มีพื้นหลัง ให้สามารถนำมาซ้อนทับกับภาพภูมิทัศน์ได้ โดยที่พื้นหลังยังโปร่งใสอยู่ (รูปที่ 4.18) เป็นการกำหนด ข้อความที่เราต้องการแสดงผลโดย บันหน้าจอภาพ ซึ่งสามารถกำหนดสี ตำแหน่ง พิกัดของข้อความได้

```
void overlayImage(Mat* src, Mat* overlay, const Point& location)
{
    for (int y = max(location.y, 0); y < src->rows; ++y)
    {
        int fY = y - location.y;

        if (fY >= overlay->rows)
            break;

        for (int x = max(location.x, 0); x < src->cols; ++x)
        {
            int fX = x - location.x;

            if (fX >= overlay->cols)
                break;

            double opacity = ((double)overlay->data[fY * overlay->step + fX * overlay->channels() + 3]) / 255;

            for (int c = 0; opacity > 0 && c < src->channels(); ++c)
            {
                unsigned char overlayPx = overlay->data[fY * overlay->step + fX * overlay->channels() + c];
                unsigned char srcPx = src->data[y * src->step + x * src->channels() + c];
                src->data[y * src->step + src->channels() * x + c] = srcPx * (1. - opacity) + overlayPx * opacity;
            }
        }
    }
}
```

รูปที่ 4.17 Source Code Overlay 1

```
cv::Scalar sums;
sums = cv::sum(underlay);
double totalSum = sums[0] + sums[1] + sums[2];
//cout << "R: " << sums[2]/totalSum * 100 << "% \n";
//cout << "G: " << sums[1]/totalSum * 100 << "% \n";
//cout << "B: " << sums[0]/totalSum * 100 << "% \n";

double sumred = sums[2]/totalSum * 100;
double sumwarn = 40;

sprintf(name,"% RedColor = %.2f",sumred);
putText(image,name, Point(360,440) , FONT_HERSHEY_SIMPLEX, .7, Scalar(255,255,255), 2,8, false );

if (sumred > sumwarn)
{
    sprintf(name,"WARNING !!");
    putText(image,name, Point(200,440) , FONT_HERSHEY_SIMPLEX, .7, Scalar(0,255,255), 2,8, false );
}
```

รูปที่ 4.18 Source Code Overlay 2

4.1.3.5 Code BLEND.cpp (รูปที่ 4.19) สร้าง Track bar สำหรับการผสมภาพ
ภาพด้วยการปรับ Alpha และ Beta สลับความโปร่งแสงระหว่าง 2 ภาพ เพื่อแสดงถึงความ
เปลี่ยนแปลงของภาพได้ (รูปที่ 4.20) สังค่า Track bar ไปยัง Void on_trackbar และ
แสดงผลการรวมภาพ ตัวแปร src1 และ src2

```
/// Create Trackbars
char TrackbarName[50];
sprintf( TrackbarName, "Alpha x %d", alpha_slider_max );
createTrackbar( TrackbarName, "Blend", &alpha_slider, alpha_slider_max, on_trackbar );

/// Show some stuff
on_trackbar( alpha_slider, 0 );
```

รูปที่ 4.19 Source Code Blend 1

```
void on_trackbar( int, void* )
{
    alpha = (double) alpha_slider/alpha_slider_max ;
    beta = ( 1.0 - alpha );

    addWeighted( src1, alpha, src2, beta, 0.0, dst);

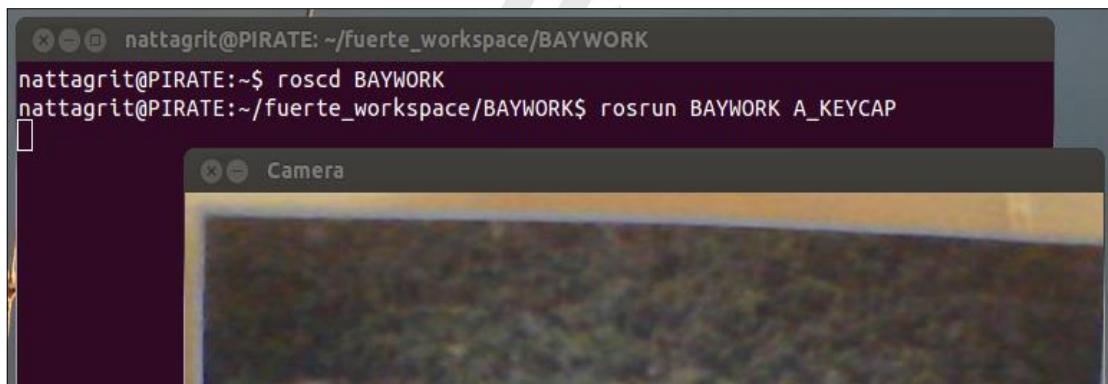
    imshow( "Blend", dst );
    imwrite( "/home/nattagrit/fuerte_workspace/BAYWORK/STEP1-4/Step4 Blend.png", dst );
}
```

รูปที่ 4.20 Source Code Blend 2

4.2 การทำงานของระบบ

4.2.1 ถ่ายภาพภูมิทัศน์

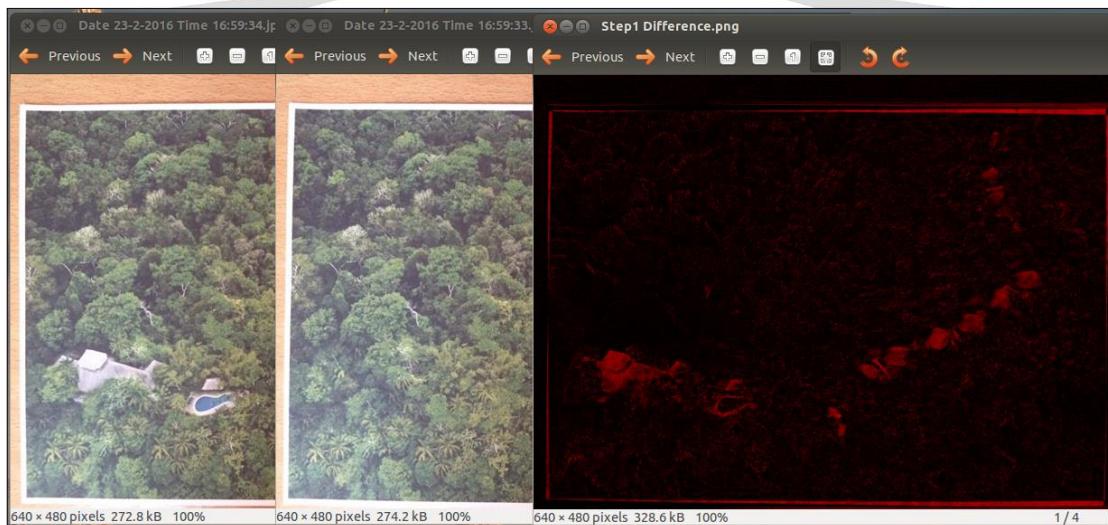
เปิด Terminal ขึ้นมา พิมพ์คำสั่ง roscl BAYWORK เข้าสู่ Packages ROS ตามด้วยคำสั่ง rosrun BAYWORK KEYCAP เป็นการรัน Executable File ชื่อ KEYCAP เป็นโปรแกรมการเปิดกล้อง กด keyboard “S” เพื่อถ่ายภาพ



รูปที่ 4.21 Window Program Keycap

4.2.2 หาความเปลี่ยนแปลงของภาพถ่ายภูมิทัศน์

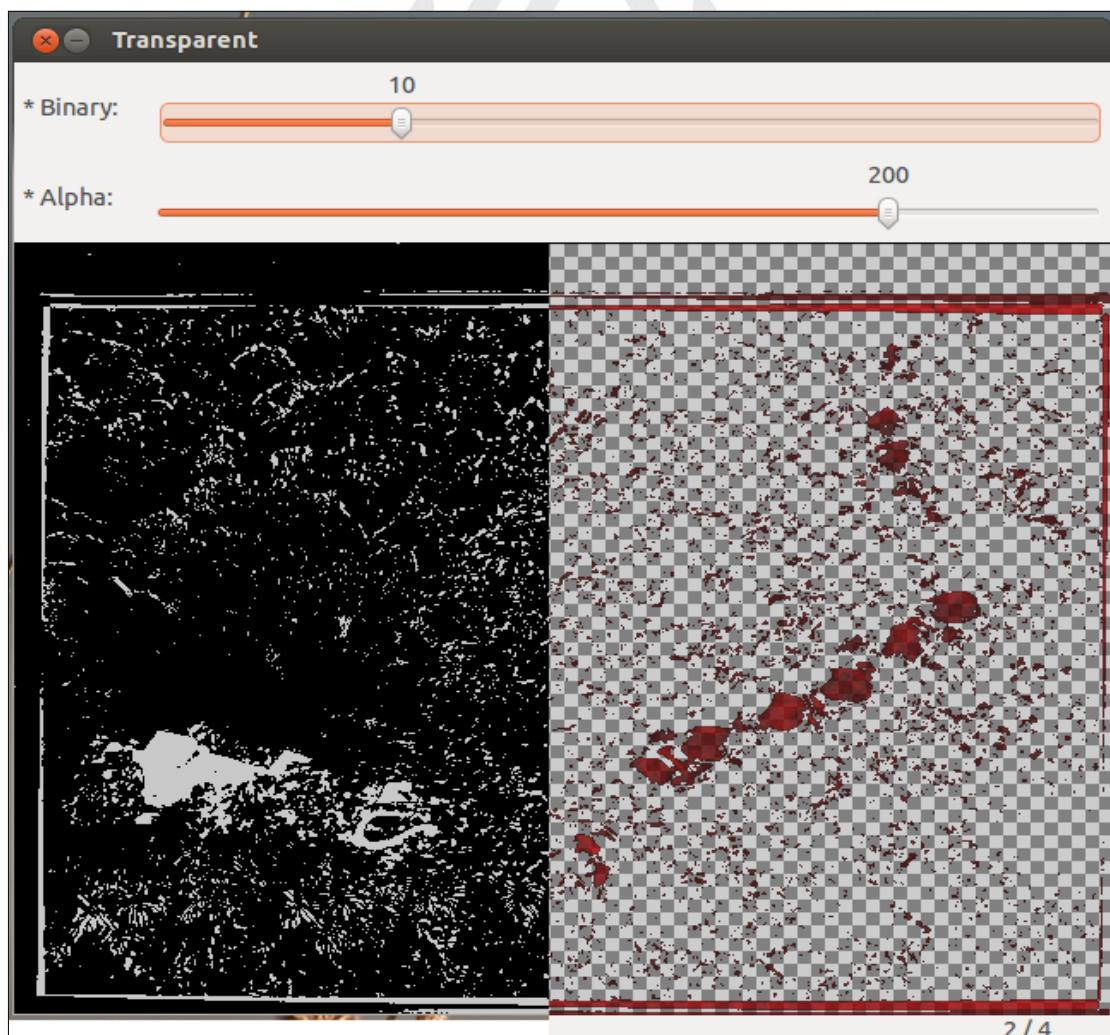
rosrun BAYWORK DIFFERENCE เพื่อแสดงผลของการเปรียบเทียบภูมิทัศน์ 2 ภาพ ที่ถ่ายโดยฟังก์ชันก่อนหน้านี้ และแสดงความแตกต่างออกมามีผู้พัฒนาได้ปรับเป็นสีแดง



รูปที่ 4.22 Window Program Difference

4.2.3 ปรับภาพ Transparent, Threshold & Alpha

rosrun BAYWORK NOBLACK ฟังก์ชันของโปรแกรมนี้จะนำเข้าภาพที่ได้จากโปรแกรมหา DIFFERENCE มาปรับภาพเป็น Gray Color เพื่อแยกองค์ประกอบของสีออกจากกัน จากนั้น Track bar Binary ส่วนบนของโปรแกรมจะทำหน้าที่ปรับ Threshold เพื่อลบ Pixel ที่กระจายออกจนเหลือเฉพาะส่วนสำคัญที่เป็นความเปลี่ยนแปลงของภาพไว้ ส่วน Track bar Alpha จะทำหน้าที่ปรับความทึบแสงเพื่อให้ภาพดูกลมกลืนขึ้น เมื่อเราปรับทุกอย่างเสร็จ เมื่อ Output ภาพออกมาพื้นหลังของภาพจะถูกแยกออกไป เหลือเพียงส่วนที่เป็นความต่างของภาพ จะได้ภาพที่ไม่มีพื้นหลัง หรือ Transparent



รูปที่ 4.23 Window Program NOBLACK : Gray Color (ขวา) Transparent (ซ้าย)

4.2.4 วางแผนภาพ Overlay

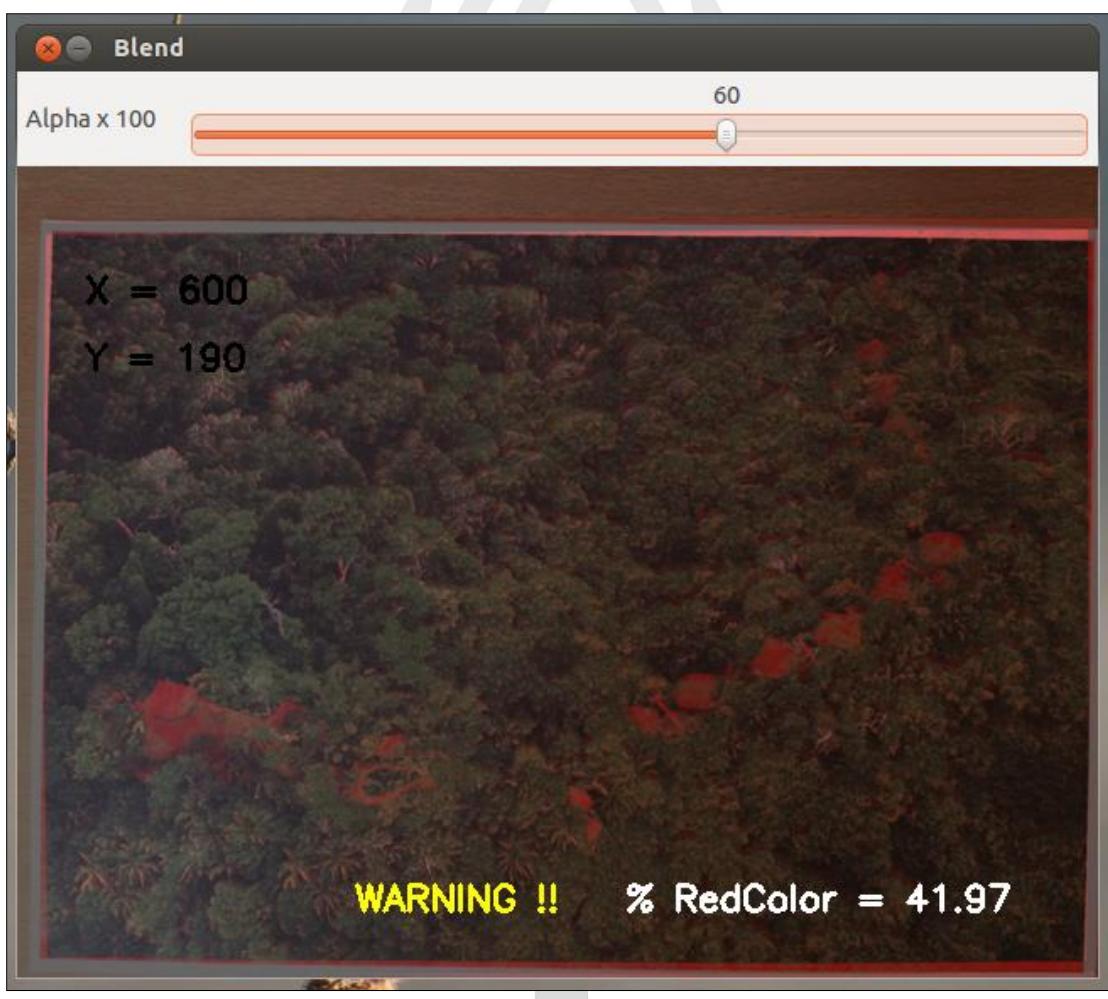
rosrun BAYWORK OVERLAY โปรแกรมนี้คือการซ้อนภาพ โดยการนำภาพที่ได้จากขั้นตอน NOBLACK มาซ้อนทับกับภาพถ่ายภูมิทัศน์ภาพแรก ที่ใช้ในการอ้างอิงเมื่อเกิดความเปลี่ยนแปลง โดยหน้าต่างของโปรแกรมจะแสดง แกน X , Y คือพิกัดของตำแหน่งใน Frame ส่วน Red Color = 38.47 หมายถึงปริมาณโทนสีแดง ซึ่งเป็นส่วนที่มีความเปลี่ยนแปลง คิดเป็น % ต่อสิ่หั้งหมดที่เกิดขึ้นบน Frame โดยคำนวนจากค่า RGB ของภาพขั้นตอนนี้เป็นขั้นตอนสุดท้ายของการเบรียบเทียบความเปลี่ยนแปลงของภูมิทัศน์ จะใช้ภาพนี้ในการ Update ความเปลี่ยนแปลงต่อรอบในการสำรวจ



รูปที่ 4.24 Window Program OVERLAY

4.2.5 ผสมผสานภาพ Blend

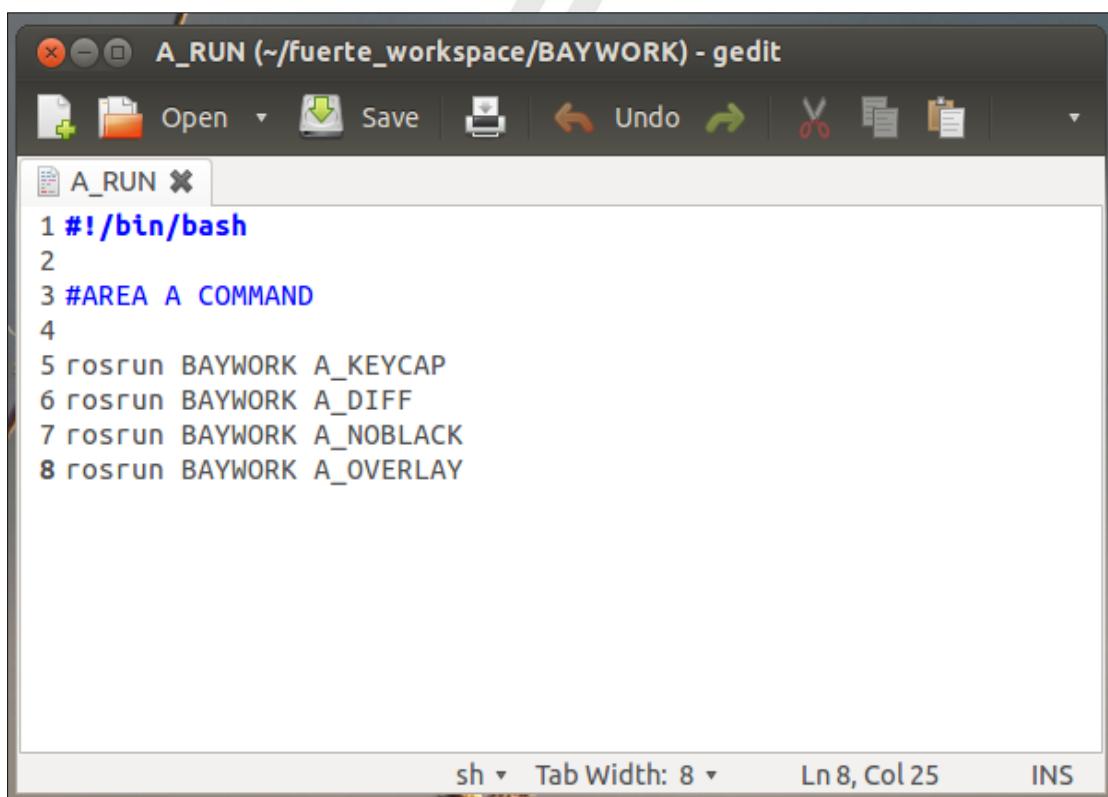
rosrun BAYWORK BLENDF โปรแกรมนี้คือการผสมผสานภาพ โดยการนำภาพที่ได้จากขั้นตอน DIFFERENCE มาผสมผสานกันโดย สลับความทึบความสว่างของภาพ ระหว่างทั้ง 2 ภาพ ให้เหลือมล้ำกัน และแสดงให้เห็นถึงความเปลี่ยนแปลงในภาพ แต่มีข้อเสียคือโหนสีของภาพจะทึบเกิดจากการนำภาพที่ยังมีพื้นหลังสีเข้มมาทำการผสมผสาน แต่ก็ได้ผลดีในการหาความเปลี่ยนแปลง พังก์ชั่นเพิ่มเติมในหน้าต่างจะเหมือนกับโปรแกรม OVERLAY แต่ตรง Red Color ถ้ามีปริมาณของโหนสีแดงมากกว่า 40 % จะกำหนดให้ แสดงข้อความว่า WARNNING !!! ซึ่งหมายความว่า เกิดความเปลี่ยนแปลงที่มากเกินกว่า 40 % จะแจ้งคำเตือน



รูปที่ 4.25 Window Program Blend

4.2.6 เรียกใช้ Bash Script

นำคำสั่งที่เราต้องการ Run ทุกคำสั่ง มาใส่ใน Bash Script เพื่อรับรู้ขั้นตอนในการประมวลภาพ ให้ได้ออกมาซึ่งผลลัพธ์เลย ซึ่งทั้งหมดนี้ต้องตั้งค่า Executable File ทั้งหมด ให้กดแสดงหน้าต่างโปรแกรม และข้ามไปยังการบันทึกข้อมูลเลย เราเก็บจะได้โปรแกรมที่ สั่งให้ทุกโปรแกรมทำงาน เพียงการ Run แค่ครั้งเดียว



```
1 #!/bin/bash
2
3 #AREA A COMMAND
4
5 rosrun BAYWORK A_KEYCAP
6 rosrun BAYWORK A_DIFF
7 rosrun BAYWORK A_NOBLACK
8 rosrun BAYWORK A_OVERLAY
```

รูปที่ 4.26 Bash Script File.sh

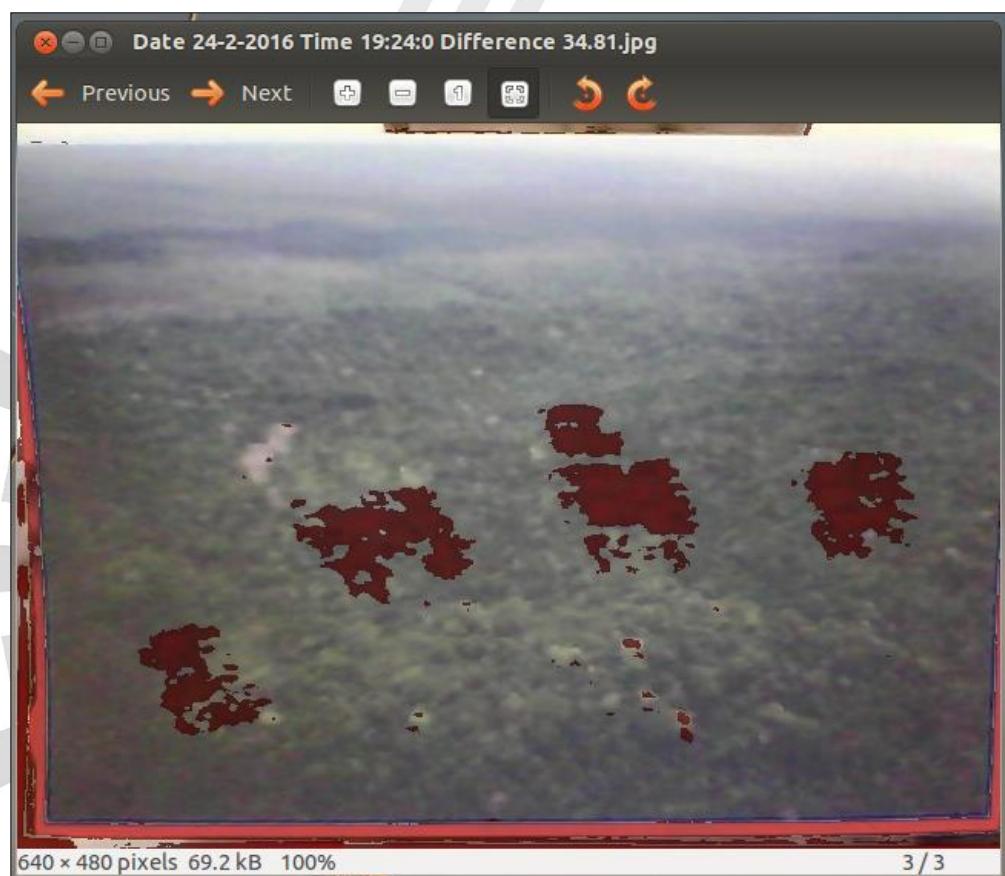
4.3 การทดสอบระบบ

เนื่องจากระบบเป็นการประมวลผลภาพ ผู้พัฒนาจึงได้ทำการออกแบบกรณีทดสอบ เพื่อหาความเหมาะสมในการเลือกใช้เครื่องมือ และความเหมาะสมของภาพ 2 กรณีตัวยึดกัน คือ

4.3.1 ทดสอบการประมวลผลภาพ ในด้านความชัดของรูปภาพภูมิทัศน์

จำนวน 3 รูปแบบการทดสอบ คือ

4.3.1.1 ภาพภูมิทัศน์ จากรูปภาพ ถ่ายโดยกล้อง Webcam Notebook



รูปที่ 4.27 Test From Webcam Camera

ผลการทดสอบพบว่า ภาพที่ถ่ายโดยกล้อง Webcam Notebook มีความชัดต่ำ มีความมัว สามารถตรวจสอบความเปลี่ยนแปลงในระดับที่น้อยมาก การทดสอบนี้จึงไม่ง่าย

4.3.1.2 ภาพภูมิทัศน์ จากรูปภาพ ถ่ายโดยกล้อง AR. Drone 2.0



รูปที่ 4.28 Test From AR. Drone 2.0 Camera

ผลการทดสอบพบว่า ภาพที่ถ่ายโดยกล้อง AR. Drone 2.0 มีความชัดมาก สามารถตรวจสอบความเปลี่ยนแปลงในระดับที่ต่ำมาก การทดสอบนี้จึงช่วยการันตีได้ว่าการประมวลผลภาพของระบบ จำเป็นต้องใช้กล้องที่มีคุณภาพเทียบเท่ากับกล้องของ AR. Drone 2.0 หรือดีกว่า จึงจะทำการประมวลผลได้ดีหรือพอเหมาะสม การทดสอบนี้จึงผ่าน

4.3.1.3 ภาพภูมิทัศน์ โดยตรงจากคอมพิวเตอร์ (มีความชัดสูงที่สุด)



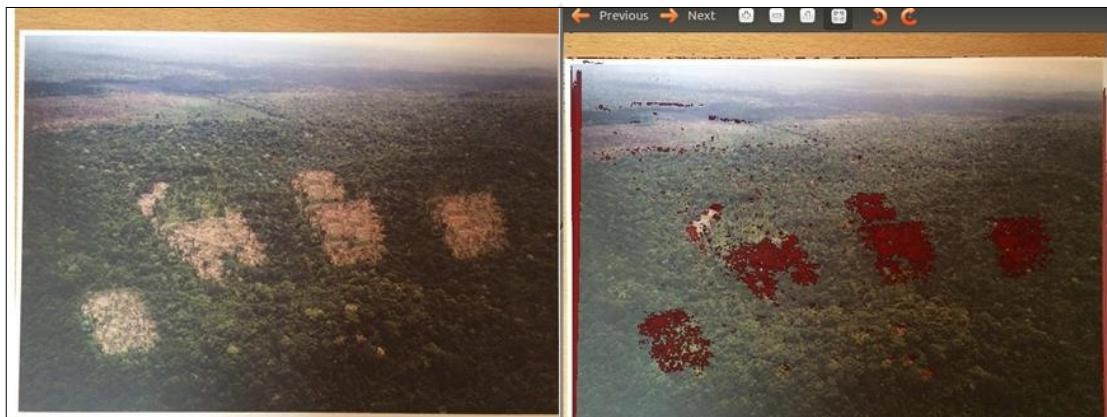
รูปที่ 4.29 Test From Original Image

ผลการทดสอบพบว่า ภาพที่นำเข้าจากคอมพิวเตอร์โดยตรง มีความชัดสูงสุด สามารถตรวจสอบความเปลี่ยนแปลงได้ ในระดับที่ละเอียดมาก แต่ในความเป็นจริงแล้วเราไม่สามารถทำแบบนี้ได้ เพราะภาพมีความเหมือนกันเกินไป จึงไม่มีความสมจริง การทดสอบนี้จึงไม่ผ่าน

4.3.2 ทดสอบการประมวลผลภาพ ในด้านระดับความคลาดเคลื่อนของภูมิทัศน์เดิม

ทดสอบโดยกล้อง AR. Drone 2.0 จำนวน 3 รูปแบบการทดสอบ

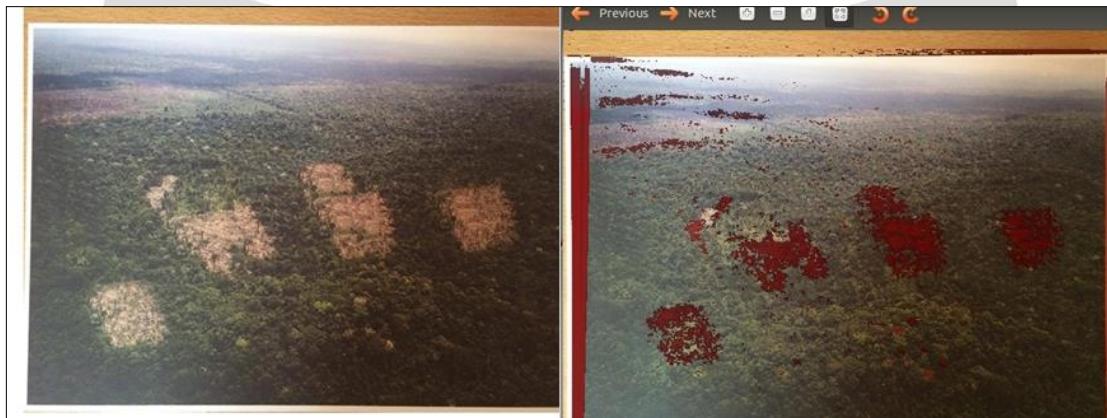
4.3.2.1 ภาพที่มีความคลาดเคลื่อนของภูมิทัศน์ ระดับเล็กน้อย



รูปที่ 4.30 Test ความคลาดเคลื่อนของภาพ ระดับน้อย

ผลการทดสอบพบว่า ความคลาดเคลื่อนในระดับน้อย ยังสามารถตรวจสอบความเปลี่ยนแปลงได้ สังเกตจากปริมาณสีแดงภายในภาพมีการกระจายเล็กน้อย

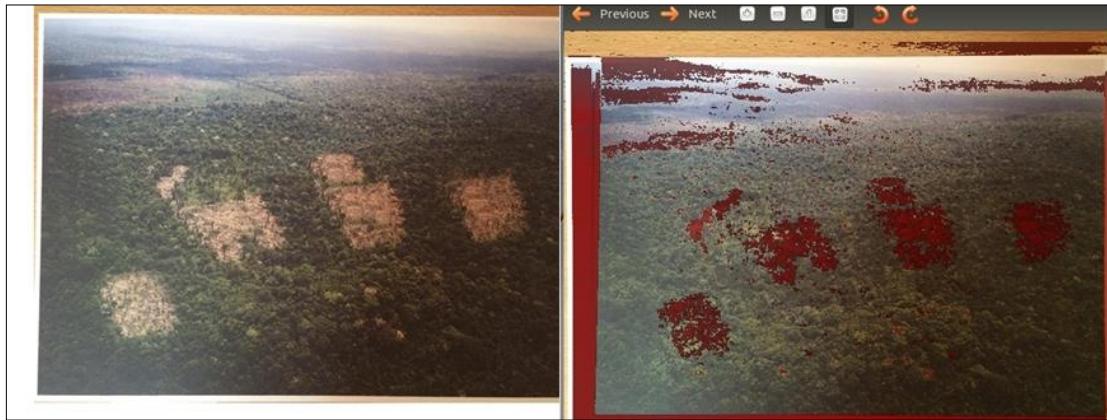
4.3.2.2 ภาพที่มีความคลาดเคลื่อนของภูมิทัศน์ ระดับปานกลาง



รูปที่ 4.31 Test ความคลาดเคลื่อนของภาพ ระดับปานกลาง

ผลการทดสอบพบว่า ความคลาดเคลื่อนในระดับปานกลาง ยังค่อนข้างสามารถตรวจสอบความเปลี่ยนแปลงได้ในระดับที่ไม่ได้มาก สังเกตจากปริมาณสีแดงภายในภาพมีการกระจายที่เพิ่มขึ้น เพราะว่าสถานที่คลาดเคลื่อนจากตำแหน่งเดิม

4.3.2.1 ภาพที่มีความคลาดเคลื่อนของภูมิทัศน์ ระดับมาก



รูปที่ 4.32 Test ความคลาดเคลื่อนของภาพ ระดับมาก

ผลการทดสอบพบว่า ความคลาดเคลื่อนในระดับมาก ไม่สามารถตรวจสอบความเปลี่ยนแปลงได้ เพราะว่าปริมาณของสีที่เกิดขึ้นจากความคลาดเคลื่อนของภาพมีมากเกินไป ความเปลี่ยนแปลงภูมิทัศน์จึงไม่ใช่ตำแหน่งเดิม ภาพความเปลี่ยนแปลงนี้จึงไม่สามารถนำไปใช้ได้

4.3.3 สรุปผลการทดสอบระบบ

การทดสอบระบบ ทั้ง 2 กรณีทดสอบ ได้ผลสรุปว่า ระบบการสำรวจความเปลี่ยนแปลงภูมิทัศน์ ต้องใช้กล้องที่มีคุณภาพในระดับ เดียวกับ AR. Drone 2.0 หรือ ในระดับ HD ที่ใกล้เคียงกันจะสามารถแสดงผลได้ดี ส่วนกรณีที่สอง ด้านความคลาดเคลื่อนของภาพ จะต้องอยู่ในระดับน้อยถึงปานกลาง ถ้าคิดเป็นเปอร์เซ็น ต้องมีความคลาดเคลื่อนไม่เกิน 10 % โดยประมาณ อ้างอิงจากภาพภูมิทัศน์แรก ถึงจะให้ผลลัพธ์ที่ถูกต้อง

บทที่ 5

บทสรุปและข้อเสนอแนะ

ในบทนี้เป็นการสรุปสาระสำคัญของผลการพัฒนาที่ทำมาทั้งหมด โดยมีความสอดคล้องกับ
บทคัดย่อ และบอกข้อเสนอแนะแนวทางในการพัฒนาต่อ เพื่อนำผลการพัฒนาไปใช้ให้เกิดประโยชน์
แบ่งออกเป็น 3 ตอน คือ

5.1 สรุปผลการพัฒนาระบบ

5.2 ปัญหาและอุปสรรคในการพัฒนาระบบ

5.3 แนวทางการแก้ไขปัญหาและข้อเสนอแนะ

ทั้ง 3 ตอน มีรายละเอียดดังนี้

5.1 สรุปผลการพัฒนาระบบ

จากการศึกษาปัญหา รวบรวมข้อมูล วิเคราะห์เทคโนโลยีที่สามารถนำมาใช้เพื่อพัฒนาระบบ ออกแบบความสามารถของระบบ วางแผนการดำเนินงาน ตลอดจนลงมือพัฒนา และทดสอบระบบ ในหัวข้อปริญญา呢พนร “การสำรวจความเปลี่ยนแปลงภูมิทัศน์โดยอากาศยานไร้คนขับ” ทำให้ได้ต้นแบบโปรแกรมเบื้องต้น ประกอบด้วยส่วนของซอฟต์แวร์ คือ โปรแกรมการประมวลผลภาพ เพื่อหาความเปลี่ยนแปลงภูมิทัศน์ และส่วนที่เป็นhardtware คือ เทคโนโลยีการถ่ายภาพทางอากาศ อากาศยานไร้คนขับ ซึ่งเป็นการนำโปรแกรมคอมพิวเตอร์มาประยุกต์ใช้กับกล้องของอากาศยานไร้คนขับ เพื่อการถ่ายภาพภูมิทัศน์ และส่งมายังโปรแกรมประมวลผลความเปลี่ยนแปลงแบบอัตโนมัติ พร้อมอัพเดตความเปลี่ยนแปลงภูมิทัศน์ได้เรื่อยๆ จนกว่าจะเปลี่ยนภูมิทัศน์ในการสำรวจใหม่ สามารถนำมาใช้กับระบบงานภาพถ่ายทางอากาศ ระบบงานภาพถ่ายดาวเทียม หรือระบบงานภาพถ่ายมุมสูง เพื่อหาความเปลี่ยนแปลง ได้ เช่น กัน

คุณสมบัติของระบบ

1. สามารถหาความเปลี่ยนแปลงภูมิทัศน์ล่าสุดได้ทันที เมื่อมีการถ่ายภาพภูมิทัศน์ล่าสุดเข้ามาในระบบ
2. สามารถบันทึก วันที่ - เวลา ของภาพภูมิทัศน์เก็บลงในแฟ้มข้อมูลภาพเพื่อใช้ในการอ้างอิง เรื่องเวลา กับความเปลี่ยนแปลง
3. สามารถคำนวนความเปลี่ยนแปลงภูมิทัศน์ และแสดงผลคิดเป็นเบอร์ เช่นว่า มีความเปลี่ยนแปลงเกิดขึ้นกี่เบอร์ เช่น
4. นอกจากหาความเปลี่ยนแปลงภูมิทัศน์แล้ว โปรแกรมสามารถหาความแตกต่างของภาพอื่นๆ ได้ทุกภาพที่อยู่ในมุมมองเดียวกัน

สรุปได้ว่าระบบสามารถทำงานได้ตามวัตถุประสงค์ และขอบเขตของการพัฒนาระบบที่กำหนดไว้ ในเรื่องของการตรวจสอบความเปลี่ยนแปลงภูมิทัศน์ ถือว่าทำงานหลักของหัวข้อได้สำเร็จ

5.2 ปัญหาและอุปสรรคในการพัฒนาระบบ

ในการพัฒนาระบบ แนวคิดแรกเริ่มคือ การให้อาชญาณไร้คนขับ bin ขึ้นไปถ่ายภาพภูมิทัศน์แบบอัตโนมัติ เพื่อนำภาพที่ได้มาประมวลผลแบบอัตโนมัติเข่นกัน แต่ด้วยข้อจำกัดของการประมวลผลภาพความเปลี่ยนแปลงภูมิทัศน์ จำเป็นต้องใช้ภาพที่มีความใกล้เคียงกับตำแหน่งเดิม เพื่อให้การหาความเปลี่ยนแปลงเป็นไปอย่างแม่นยำไม่คลาดเคลื่อน อาชญาณไร้คนขับซึ่งเป็นไปได้ยากมากที่จะถ่ายภาพตำแหน่งใกล้เคียงกับภูมิทัศน์เดิม เพราะด้วยแรงลม และการกระยะทิศทางเดิม ยังไม่เห็นแนวทางที่จะทำได้สำเร็จ จึงได้เปลี่ยนแนวคิดเป็น จำลองการถ่ายภาพภูมิทัศน์จากรูปภาพแทน ซึ่งสามารถใช้กล้องทั่วไปที่เชื่อมต่อกับคอมพิวเตอร์ จำลองว่าเป็นอาชญาณไร้คนขับ ถ่ายภาพจากรูปซึ่งจำลองว่าเป็นภูมิทัศน์จริงๆ แทน

ปัญหาในการพัฒนาระบบอีกอย่างหนึ่ง คือ ไม่มีหน้า Interface หมายถึงระบบนี้ ไม่มีหน้าตาของโปรแกรม ผู้พัฒนาได้พยายามสร้างหน้าตาของโปรแกรมด้วยการจำลองบน Localhost แต่ไม่สามารถทำได้สำเร็จเนื่องจากติดปัญหาบางประการ

5.3 แนวทางการแก้ไขปัญหาและข้อเสนอแนะ

1. ติดตั้งอุปกรณ์เสริมให้กับอาชญาณไร้คนขับเพื่อเพิ่มความสามารถในการบินไปยังตำแหน่งเดิม เช่น GPRS
2. เรียนรู้เกี่ยวกับ PHP และ Bash Script ในการสร้างหน้าตา Interface ของโปรแกรมบนระบบปฏิบัติการ Linux, Ubuntu
3. เรียนรู้เกี่ยวกับ การเคลื่อนที่ และการถ่ายภาพอัตโนมัติ ของอาชญาณไร้คนขับ
4. เลือกใช้โถนสีให้ตัดกับสีของภาพ ในการหาความเปลี่ยนแปลง

បរណានុករម

Asatisa. (2016, Jan 28). Introductions To Linux Ubuntu. Retrieved from startubuntu-asatisa: <http://startubuntu-asatisa.blogspot.com>

Engel, J. (2011). Autonomous Camera-Based Navigation of a Quadrocopter. Master's thesis: Technical University Munich.

Engel, J. S. (2012). Accurate Figure Flying with a Quadrocopter Using Onboard Visual and Inertial Sensing. In Proc. of the Workshop on Visual Control of Mobile Robots (ViCoMoR) at the IEEE/RJS International Conference on Intelligent Robot Systems (IROS).

Engel, J. S. (2012). Camera-Based Navigation of a Low-Cost Quadrocopter. Proc. of the International Conference on Intelligent Robot Systems (IROS).

Engel, J. S. (2014). Scale-Aware Navigation of a Low-Cost Quadrocopter with a Monocular Camera. In Robotics and Autonomous Systems (RAS): volume 62.

Gary, R. B., & Adrian, K. (2008). Learning OpenCV Computer Vision with the OpenCV Library. United States: O'Reilly Media, Inc.,

Itseez. (2016, Jan 15). OpenCV Tutorials. Retrieved from opecv: <docs.opencv.org>

Parrot. (2016, Jan 11). AR. Drone 2.0 Technical specifications State of the art technology. Retrieved from ardrone2.parrot: <ardrone2.parrot.com/ardrone-2/specifications/>

Robotics Foundation. (2016, Jan 15). ROS Tutorials & Learning. Retrieved from wiki.ros: <http://wiki.ros.org/fuerte>

Tutorials Point. (2016, Jan 19). C++ Tutorial & Learning. Retrieved from tutorialspoint: <http://www.tutorialspoint.com/cplusplus/>

Tutorials Point. (2016, Jan 17). Digital Image Processing Introduction. Retrieved from tutorialspoint:

http://www.tutorialspoint.com/dip/image_processing_introduction.htm

- Тематические.** (2016, Jan 18). OpenCV - Comparison of images and generation differences pictures & Overlay Between Images. Retrieved from robocraft.ru: <http://robocraft.ru/blog/computervision/506.html>
- แก้ว นวลชนิม. (2016, Feb 9). Remote sensing and geographic information technology in Thailand เทคโนโลยีการสำรวจระยะไกลด้วยรีโมตเซนซิ่ง. Retrieved from negistd: <http://negistda.kku.ac.th/jresgat/vol11/index.html>
- กาญจน์เชจร ชูชีพ. (2016, Feb 14). ภาพถ่ายทางอากาศ การสำรวจระยะไกล. Retrieved from Research: <https://research.rdi.ku.ac.th/forest/Person.aspx?id=420006>
- จอมยุทธ บ้านจอมยุทธ. (2016, Feb 4). ความสำคัญของปัญหาทรัพยากรธรรมชาติ. Retrieved from baanjomyut: <http://www.baanjomyut.com/>
- ชุติสันต์ เกิดวิบูลย์เวช. (2016, ธันวาคม). จากตามนุษย์สู่สมองคอมพิวเตอร์. Retrieved from นิตยสารผู้จัดการ: <http://info.gotomanager.com/news/printnews.aspx?id=93754>
- ธนภัทร สังขรัตน์. (2016, Feb 13). Image Processing ผู้ช่วยคนใหม่ในกระบวนการผลิต. Retrieved from Thailandindustry: <http://www.thailandindustry.com/guru/view.php?id=19424§ion=9>
- นายเอ็มซ่าส์. (2016, Jan 26). เรียนรู้เกี่ยวกับ : ระบบไฟล์ใน Linux. Retrieved from 9mza: <http://9mza.blogspot.com/2016/02/filesystem-linux.html>
- สถานีดาวเทียม CSRS. (2016, Feb 11). สถานีรับสัญญาณดาวเทียมมุฟารอน (CSRS). Retrieved from Csrs: csrs.ku.ac.th

ประวัติผู้จัดทำปริญญา呢พนธ์

ชื่อ-นามสกุล	นาย ณัฐกฤษ์ ฤทธิขاب
วัน/เดือน/ปีเกิด	26 ตุลาคม พ.ศ. 2535
สถานที่เกิด	กรุงปี
ที่อยู่ปัจจุบัน	6 หมู่ที่ 1 ซอยจันทสุวรรณ ตำบลควนลัง อำเภอหาดใหญ่ จังหวัดสงขลา 90110
ประวัติการศึกษา	สำเร็จการศึกษาในระดับ มัธยมศึกษาตอนต้น และมัธยมศึกษาตอนปลาย สายศิลป์ – ผู้รังสรรค์ จากโรงเรียนหาดใหญ่วิทยาลัย
เกียรติประวัติ	ฝึกงานด้าน Business Intelligence ที่บริษัท Innobiz