โปรแกรมวัดความเร็ววัตถุ

SPEED MEASURING PROGRAM

วิชสุดา จรูญกุล

สุนิสา แถวโสภา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

โปรแกรมวัดความเร็ววัตถุ

SPEED MEASURING PROGRAM

วิชสุดา จรูญกุล

สุนิสา แถวโสภา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2557

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมวัดความเร็ววัตถุ

SPEED MEASURING PROGRAM

ผู้จัดทำ

1. นางสาววิชสุดา จรูญกุล รหัสนักศึกษา 54011192
2. นางสาวสุนิสา แถวโสภา รหัสนักศึกษา 54011402

อาจารย์ที่ปรึกษา

(รศ. ดร. บุญวัฒน์ อัตชู)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมวัดความเร็ววัตถุ

นางสาววิชสุดา จรูญกุล 54011192
นางสาวสุนิสา แถวโสภา 54011402
รศ. ดร. บุญวัฒน์ อัตชู อาจารย์ที่ปรึกษา
ปีการศึกษา 2557

บทคัดย่อ

โครงงานนี้เป็นการนำทฤษฎีการประมวลผลภาพมาประยุกต์ใช้กับเครื่องคอมพิวเตอร์ เพื่อศึกษาการตรวจจับการเคลื่อนที่ของวัตถุ ในส่วนของการประมวลผลภาพได้มีการนำเทคนิคการหาค่าความต่างของเฟรมแต่ละเฟรมในวิดีโอ และคำนวณหาความเร็วของวัตถุที่ผ่านเข้ามา ได้มีการนำไลบรารี่ OpenCV เข้ามาช่วยในการพัฒนาแอพลิเคชั่นร่วมกับภาษา C++ ในโปรแกรม Visual Studio จากการทดลองพบว่าสามารถคำนวณหาความเร็วของวัตถุที่เคลื่อนที่เข้ามาได้ มีความคลาดเคลื่อน 1.46 เปอร์เซ็นต์ และมีตัวแปรที่สำคัญคือ สภาพแวดล้อมต้องอยู่ในสถานที่ที่มีแสงเพียงพอ วัตถุที่วิ่งผ่านเลนส์กล้องต้องมีเพียงวัตถุเดียว กำหนดระยะทางระหว่างจุดเริ่มต้นและจุดสิ้นสุดเฟรมเทียบกับระยะทางจริง กำหนดระยะทางระหว่างเลนส์กล้องกับระยะวัตถุ และเลนส์กล้องอยู่ในระดับความสูงเดียวกันกับวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPEED MEASURING PROGRAM

Ms.Witsuda	Charoonkul	54011192
Ms. Sunisa	Thaewsopha	54011402
Assoc.Prof.Dr.	Boonwat Attachoo	Advisor
Academic year	2014	

Abstract

Speed Measuring Program project aims to study image processing technology and apply for developing a program to measure average speed of a moving object. Video frame differencing technique is used to work in image processing part. The project is developed on Visual Studio using C++ language. OpenCV library is used for real time image processing and computer vision related. As the tests, the program can measure average object speed and has 5.6 percent of discrepancy. The tests must be under areas with enough lighting. There must be only one object in camera frames each measuring. Distance between beginning and ending of camera frame is fixed comparing to distance of the moving object, distance between a camera lens and object is fixed, and the lens is set at the same level as the object.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่อง โปรแกรมวัดความเร็ววัตถุ สามารถดำเนินงานได้ เนื่องจากได้รับความสนับสนุนและช่วยเหลือในด้านต่างๆจากบุคคลต่อไปนี้

ขอขอบพระคุณ รศ. ดร. บุญวัฒน์ อัตชู อาจารย์ที่ปรึกษาโครงงาน ที่ได้ให้คำแนะนำและข้อคิดเห็นต่างๆ รวมถึงให้ความช่วยเหลือ แก้ไขข้อบกพร่องต่างๆของโครงงาน

ขอขอบพระคุณ คณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่านที่ได้ให้ความรู้ ให้คำแนะนำและกำลังใจตลอดการศึกษาที่ผ่านมา

ขอขอบพระคุณคุณพ่อ คุณแม่ และผู้ปกครอง สำหรับความรัก ความห่วงใย ความดูแลเอาใจใส่และให้การสนับสนุนเสมอมา

ขอขอบคุณเพื่อนๆทุกคนสำหรับความห่วงใย ความช่วยเหลือ กำลังใจ รวมทั้งคำปรึกษาต่างๆที่ให้กันเสมอมา

คุณค่าและประโยชน์อันพึงมีจากโครงงานนี้ ผู้พัฒนาโครงงานขอมอบแด่ผู้มีพระคุณทุกท่าน

วิชสุดา จรูญกุล

สุนิสา แถวโสภา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญรูป VII

บทที่ 1 บทนำ 1

1.1 ความสำคัญของโครงงาน 1

1.2 ลักษณะของโครงงาน 1

1.3 วัตถุประสงค์ของโครงงงาน 1

1.4 ขอบเขตของโครงงาน 2

1.5 วิธีการเดินเนินงาน 2

1.6 เครื่องมือที่ใช้ในการพัฒนา 4

บทที่ 2 หลักการและทฤษฎี 5

2.1 หลักการเขียนโปรแกรม 5

2.2 OpenCV (Open Source Computer Vision Library) 7

2.3 หลักการของ Image processing 8

2.4 หลักการติดตามวัตถุ(Object Tracking) 18

2.5 Motion Detection 20

2.6 หลักการของความเร็ว 21

2.7 หลักการของการเคลื่อนที่ของวัตถุ 23

บทที่ 3 การออกแบบและการทำงาน 29

แนวคิดการออกแบบโปรแกรม 29

ลักษณะการทำงานของโปรแกรม 29

เอกสารนี้เป็นเอกสาร[illegible]เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลอง.....31

บทที่ 4 ผลการทดลอง

4.1 สภาพแวดล้อมที่ใช้ในการทดลอง 45

4.2 การทดลองและผลการทดลอง 46

4.3 สรุปผลการทดลอง 48

บทที่ 5 บทสรุป

5.1 สรุปผลการดำเนินงาน.....50

5.2 ข้อเสนอแนะ.....51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

1.1 แผนการดำเนินงาน.....3

2.1 ค่า Hue.....17

4.1 ระยะทางที่กำหนดในการถ่ายวีดีโอ.....45

4.2 ผลการทดลอง 49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่ หน้า

2.1 การสร้างโปรเจคใหม่.....6
2.2 การ Add New Item.....6
2.3 ปุ่ม Debug.....7
2.4 ลำดับขั้นตอนของ Digital image processing.....9
2.5 Image acquisition process.....9
2.6 ภาพระดับความเข้มเทาและแสดงการเก็บค่าระดับความเข้มเทา.....10
2.7 ภาพสีและแสดงการเก็บค่าระดับความเข้มสี.....11
2.8 ภาพขาวดำและแสดงการเก็บค่า.....11
2.9 ภาพดัชนีและแสดงการเก็บค่า.....12
2.10 ค่าระดับความเข้มเทาของภาพที่มีการเปลี่ยนแปลงแบบทันทีทันใด.....13
2.11 แบบจำลองสี RGB.....14
2.12 แบบจำลองสี CMYK.....15
2.13แบบจำลองสี HIS.....16
2.14แบบจำลองสี HSV.....17
2.15 ตัวอย่าง Optical flow.....21
2.15 Flow chart แสดงขั้นตอนการทำงานของโปรแกรม.....21
2.16การเคลื่อนที่ของรถไฟบนราง.....23
2.17การเคลื่อนที่ของรถบนถนนที่เป็นแนวเส้นตรง.....24
2.18 การเคลื่อนที่แบบโปรเจกไทล์.....24
2.19 การเคลื่อนที่แบบวงกลม.....26
3.1 Flow chart แสดงขั้นตอนการทำงานของโปรแกรม.....30
3.2 ผลลัพธ์เมื่อเปิดใช้งาน application ติดต่อกับกล้องจากโทรศัพท์มือถือ.....31
3.3 ตัวอย่างภาพถ่ายที่ได้จาก application ถ่ายรูปจากด้วยโทรศัพท์มือถือ.....33
3.4 ภาพตัวอย่างการใช้งานโปรแกรมตรวจจับใบหน้า.....33
3.5 ส่วนแสดงผล Background.....37
3.6 ส่วนแสดงผลของวัตถุที่ถูก Detect.....38
3.7 ผลการทดลองจากการแยกวัตถุออกจากพื้นหลังโดยการทำ background subtraction.....39
3.8 ผลการทดลองจากการทำ Image Threshold.....40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 ผลการทดลองจากการทำ Image Threshold.....40
3.9 ผลการทดลองการรันโปรแกรมโดยเมื่อไม่มีวัตถุผ่านเข้ามาในกล้อง.....41
3.10 ผลการทดลองเมื่อมีวัตถุผ่านเข้ามาในกล้อง จะเกิดการตรวจจับวัตถุที่เคลื่อนไหวได้ 41
3.11 วีดีโอเมื่อเริ่มใช้งานโปรแกรม พร้อมทั้งบอกจำนวนเฟรม และ fps.....42
3.12 ภาพรูปภาพที่บันทึกจากเฟรมวิดีโอแต่ละเฟรม.....43
3.13 เมื่อมีวัตถุผ่านเลนส์กล้อง จะทำการติดตามวัตถุ.....44
3.14 แสดง frame rate, จำนวนเฟรมในวิดีโอ และความเร็วของวัตถุ.....44
4.1 ทดลองตรวจจับความเร็วของรถจักรยานยนตร์.....46
4.2 ทดลองตรวจจับความเร็วของรถยนต์.....47
บรรณานุกรม.....52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญของโครงงาน

เนื่องจากปัจจุบัน มีการนำเครื่องคอมพิวเตอร์มาประยุกต์ใช้เข้ากับการทำงานในหลายๆด้าน โดยเฉพาะการทำงานทางด้านการประมวลผลภาพ (Image Processing) ซึ่งเมื่อนำมาประยุกต์ใช้กับคอมพิวเตอร์แล้ว จะทำให้การประมวลผลภาพมีประสิทธิภาพมากขึ้น

โครงงานชิ้นนี้ได้นำการประยุกต์ใช้เทคโนโลยีการประมวลผลภาพ Image Processing เพื่อใช้ในการศึกษาการตรวจจับการเคลื่อนที่ของวัตถุที่อยู่ในวิดีโอ ซึ่งอาจจะนำไปประยุกต์ใช้เพื่อตรวจจับความเร็วกับวัตถุได้หลายชนิด เช่น รถยนต์ที่วิ่งบนถนน สิ่งของที่ไหลตามสายพานในโรงงานอุตสาหกรรม เป็นต้น ทั้งนี้อาจเป็นการตรวจจับเพื่อควบคุมความเร็วของรถยนต์บนถนนเพื่อป้องกันอุบัติเหตุ หรือตรวจสอบความเร็วของสินค้าในโรงงานอุตสาหกรรมเพื่อควบคุมคุณภาพการผลิต

1.2 ลักษณะของโครงงาน

โครงงงานนี้ จะมีลักษณะเป็นการนำเอาเทคโนโลยีการประมวลผลภาพดิจิตอล (Image Processing) มาประยุกต์ใช้ เพื่อโปรแกรมที่สามารถวัดความเร็วของวัตถุที่เคลื่อนที่ได้ โดยทำการบันทึกภาพวีดีโอจากกล้องบนโทรศัพท์มือถือระบบปฏิบัติการ Android และนำภาพมาผ่านการประมวลผลภาพ Image Processing เพื่อหาความเร็วของวัตุถุที่วิ่งผ่านเลนส์กล้อง ตั้งแต่จุดเริ่มต้นจนถึงจุดสิ้นสุด ทั้งนี้จะต้องมีการจำกัดตัวแปรต่างๆ โดยกำหนดระยะห่างระหว่างเลนส์กล้องและวัตถุให้คงที่ เพื่อให้สามารถคำนวณระยะของการเคลื่อนที่ผ่านเลนส์ตั้งแต่จุดเริ่มต้นจนถึงจุดสิ้นสุดได้

1.3 วัตถุประสงค์ของโครงงาน

1) เพื่อศึกษาทฤษฎีการประมวลผลภาพดิจิตอล (Image Processing)
2) เพื่อนำทฤษฎีการประมวลผลภาพดิจิตอล (Image Processing) ไปประยุกต์ใช้กับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) เพื่อพัฒนาโปรแกรมบนคอมพิวเตอร์ให้สามารถวัดความเร็วของวัตถุที่เคลื่อนที่ได้และนำไปใช้ประโยชน์ในด้านต่างๆ เช่น การวัดความเร็วของรถยนต์ที่วิ่งอยู่บนถนน หรือการวัดความเร็วของวัตถุที่วิ่งอยู่บนสายพานในโรงงานอุตสาหกรรม เป็นต้น

1.4 ขอบเขตของโครงงาน

1) ศึกษาหลักการการเขียน C++ ร่วมกับไลบรารี่ OpenCV
2) ศึกษาหลักการวิธีหา Motion Detection
3) ออกแบบแนวคิดและอัลกอริทึม
4) แยกเฟรมจากไฟล์วิดีโอออกได้เป็นเฟรมๆและสามารถเปรียบเทียบเฟรมสองเฟรมได้
5) นำเฟรมภาพที่ได้จากไฟล์วีดีโอมาประมวลผลด้วยเทคโนโลยีการประมวลผลภาพดิจิตอล (Image Processing)

1.5 วิธีการดำเนินงาน

โครงงานที่ 1

1) ศึกษาปัญหาและแนวคิดของโครงงาน
2) ศึกษาการเขียนแอปพลิเคชัน
3) ศึกษาหลักการของการแยกวัตถุออกจากพื้นหลังด้วยวิธีการทำ Background Subtraction
4) ศึกษาอัลกอริทึมที่จะนำมาประยุกต์ใช้ในการพัฒนาโปรแกรม
5) ศึกษา Library ของ OpenCV เพื่อทดสอบอัลกอริทึม

โครงงานที่ 2

1) ศึกษาค้นคว้าหาข้อมูลและทฤษฎีต่างทางด้าน Image processing เพิ่มเติม
2) เขียนโปรแกรมโดยการนำอัลกอริทึมมาประยุกต์ใช้
3) ทดสอบความสามารถของโปรแกรมในการตรวจจับวัตถุ
4) ทดสอบความสามารถของโปรแกรมในการตรวจจับความเร็วของวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1.1 แผนการดำเนินงาน

รายละเอียดการดำเนินการ	ระยะเวลาในการดำเนินการ								
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
1. ศึกษาปัญหาและแนวคิดของโครงงาน									
2. ศึกษาการเขียนแอปพลิเคชัน									
3. ศึกษาหลักการของการแยกวัตถุออกจากพื้นหลังด้วยวิธีการทำ Background Subtraction									
4. ศึกษาอัลกอริทึมที่จะนำมาประยุกต์ใช้ในการพัฒนาโปรแกรม									
5. ศึกษา Library ของ OpenCV เพื่อทดสอบอัลกอริทึม									
6. ทำรายงานสรุปผลการดำเนินการของโครงงานที่ 1									
7. ศึกษาค้นคว้าหาข้อมูลและทฤษฎีต่างทางด้าน Image processing เพิ่มเติม									
8. เขียนโปรแกรมโดยการนำอัลกอริทึมมาประยุกต์ใช้									
9. ทดสอบความสามารถของโปรแกรมในการตรวจจับวัตถุ									
10. ทดสอบความสามารถของโปรแกรมในการตรวจจับความเร็วของวัตถุ									
11. ทำรายงานสรุปผลการดำเนินการของโครงงานที่ 2									

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 เครื่องมือที่ใช้ในการพัฒนา

1.6.1 เครื่องมือทางด้าน Hardware

- เครื่องคอมพิวเตอร์ส่วนบุคคล
- โทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์

1.6.2 เครื่องมือทางด้าน Software

- ระบบปฏิบัติการ Microsoft Window
- โปรแกรม Microsoft visual studio 2010
- Library ของ OpenCV สำหรับการเรียกใช้ฟังก์ชั่นต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

สำหรับหลักการและทฤษฎีที่เกี่ยวข้องในการทำโครงงานนี้มีอยู่ 7 ส่วน ได้แก่

2.1 หลักการเขียนโปรแกรม

2.2 OpenCV (Open Source Computer Vision Library)

2.3 หลักการของ Image processing

2.4 หลักการติดตามวัตถุ(Object Tracking)

2.5 Motion Detection

2.6 หลักการของความเร็ว

2.7 หลักการของการเคลื่อนที่วัตถุ

2.1 หลักการเขียนโปรแกรม

2.1.1 โปรแกรมและเครื่องมือต่างๆ

โปรแกรม Microsoft Visual Studio 2010 เป็นเครื่องมือที่ช่วยนักพัฒนาซอฟต์แวร์พัฒนาโปรแกรมคอมพิวเตอร์ เว็บไซต์ เว็บแอปพลิเคชัน และ เว็บเซอร์วิซ ระบบที่รองรับการทำงานนั้นมีไมโครซอฟท์ วินโดวส์ พ็อคเกตพีซี Smartphone และ เว็บเบราว์เซอร์ ในปัจจุบัน วิชวลสตูดิโอนั้นสามารถใช้ภาษาโปรแกรมที่เป็น .NET ในโปรแกรมเดียวกันเช่น VB.NET C++ C# เป็นต้น

1. การใช้งานเบื้องต้นของโปรแกรม Microsoft Visual Studio 2010

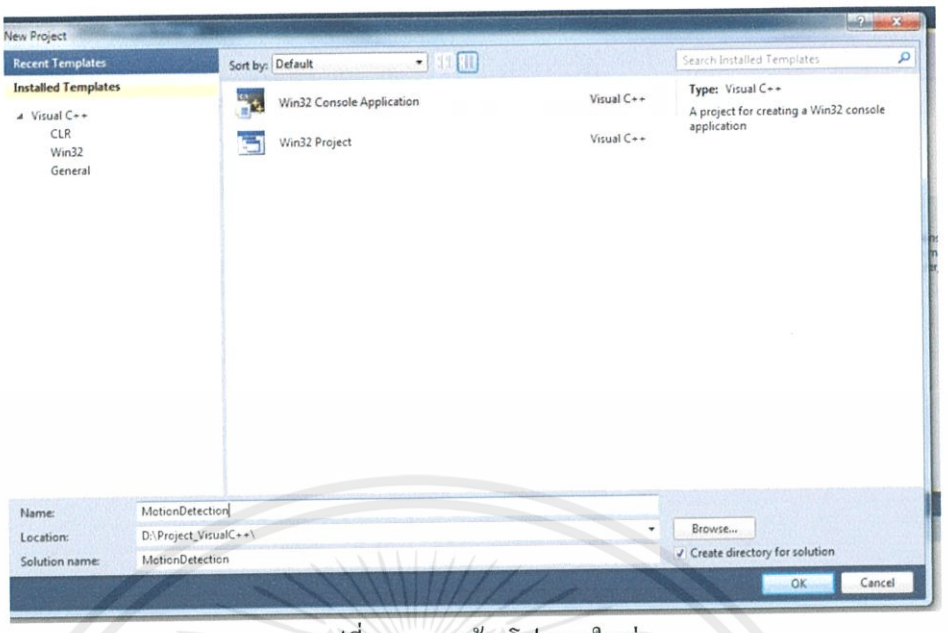
ก่อนที่เราจะเริ่มพัฒนาโปรแกรม จะต้องเข้าสู่สภาพแวดล้อมของการพัฒนาโปรแกรมเสียก่อน โดยเราต้องสร้างโปรเจ็กต์ใหม่ขึ้นมา โดยมีวิธีการสร้างดังนี้

วิธีที่ 1 ที่มุมมอง Start Page คลิกที่ลิงค์ New Project…

วิธีที่ 2 เลือกเมนู File -> New Project…

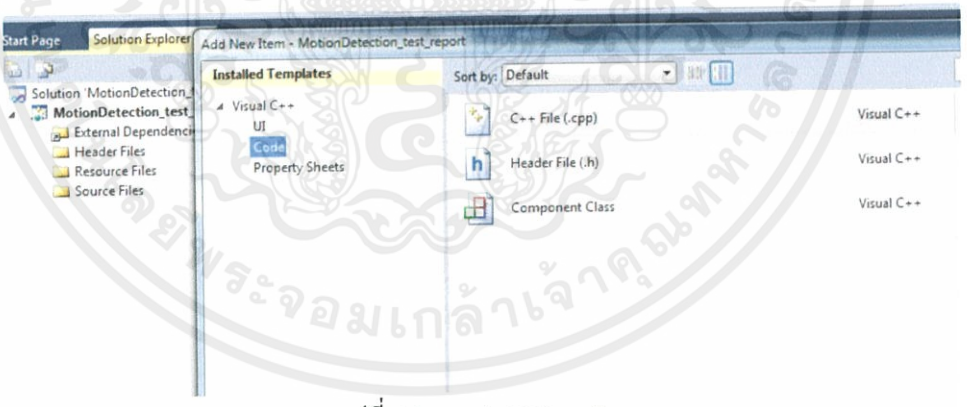
จากนั้นจะปรากฏหน้าต่าง New Project ขึ้นมา ในช่องของ Installed Templates ให้เลือก Win32 ในส่วนของ Location ให้ทำการ Browse… เพื่อกำหนดสถานที่ของ Project จากนั้นในช่องของ Name ให้ตั้งชื่อของ Project แล้วทำการเลือก Ok จากนั้นจะมีหน้าต่าง Win32 Application Wizard ขึ้นมาให้ทำการกด Next แล้วเลือก Empty project แล้วกดปุ่ม Finish ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 การสร้างโปรเจคใหม่

หลังจากขั้นตอนเบื้องต้นแล้ว ที่ช่อง Solution Explorer ให้คลิกขวาที่ Source Files -> Add -> New Item… จากนั้นจะปรากฏหน้าต่าง Add New Item ขึ้นมาทำการเลือก Code-> C++ File(.cpp) แล้วทำการตั้งชื่อที่ช่อง Name จากนั้นคลิก Add แล้วก็ทำการเขียนโค้ดโปรแกรมลงบน Item ที่เราสร้างขึ้น

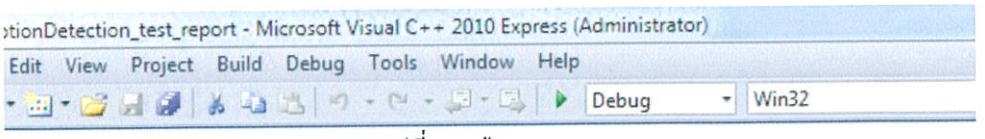


รูปที่ 2.2 การ Add New Item

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. **การรันโปรแกรมหรือทดสอบโปรแกรม**

วิธีที่ 1. กดปุ่ม



รูปที่ 2.3 ปุ่ม Debug

วิธีที่ 2. กดปุ่ม F5

วิธีที่ 3. เลือกเมนู Debug -> Start Debugging

2.2 OpenCV (Open Source Computer Vision Library)

OpenCV เป็นไลบรารี่สำหรับใช้งานเรื่องการประมวลผลภาพ (Image Processing) ซึ่งบางความสามารถของ OpenCV สามารถทำภาพเบลอ หา threshold หา Histogram ของภาพได้ แต่ความสามารถโดยส่วนใหญ่แล้วจะทำการค้นหาขอบของภาพ การตรวจสอบการเคลื่อนไหว และการทำ Image segmentation

นอกจากนี้ OpenCV สามารถจัดการกับข้อมูลแบบวิดีโอได้ด้วย เนื่องจาก OpenCV เป็นชุดคำสั่งที่ไม่ได้เป็นตัวโปรแกรม เมื่อต้องการเรียกใช้งานจึงต้องเขียนโปรแกรมเพื่อเรียกชุดคำสั่งเหล่านั้น ซึ่งภาษาที่นิยมเขียนคือภาษา C, C++ และภาษา Python ซึ่ง OpenCV จะประกอบด้วยสองส่วน คือ data structure ที่ใช้ในการเก็บข้อมูลต่างๆ อาทิ เช่น รูปภาพ เมตริกซ์และพิกัด สำหรับอีกส่วนคือ algorithm ซึ่งจะใช้ในการประมวลผลต่างๆโดยเฉพาะการประมวลทางรูปภาพ สำหรับใน OpenCV จะประกอบด้วยไลบรารี่อยู่ 4 ส่วน ได้แก่

1. CXCORE

เป็นฟังก์ชันเบื้องต้นที่ใช้จัดการเกี่ยวกับจุด ขนาด อาเรย์ หน่วยความจำ คำสั่งในการวาดภาพ การประกาศตัวแปรภาพ เป็นต้น ตัวอย่างคำสั่งในการประกาศรูปภาพ คือ IplImage, CvMat, CvMatND

2. CV

ใช้ในการประมวลผลและการวิเคราะห์รูปภาพ ฟังก์ชันส่วนใหญ่จะทำงานกับพิกเซลที่เป็นอาเรย์ 2 มิติ หรือที่เราเรียกว่าภาพนั่นเอง เช่น การหาขอบหรือมุม การทำฮิสโตแกรม (Histogram) และการทำออพติคอลโฟลว (Optical Flow) เป็นต้น

3. Machine Learning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นไลบรารีที่รวมคลาสและฟังก์ชันทางสถิติ การแยกคลาส และการแบ่งกลุ่มข้อมูล อัลกอริทึม (Algorithm) ที่จะใช้เขียนด้วยภาษา C++ แต่ละอัลกอริทึมจะมีคุณลักษณะเด่นแตกต่างกันไป แต่ทั้งหมด จะใช้คลาส CvStatModel ร่วมกัน

4. HighGUI

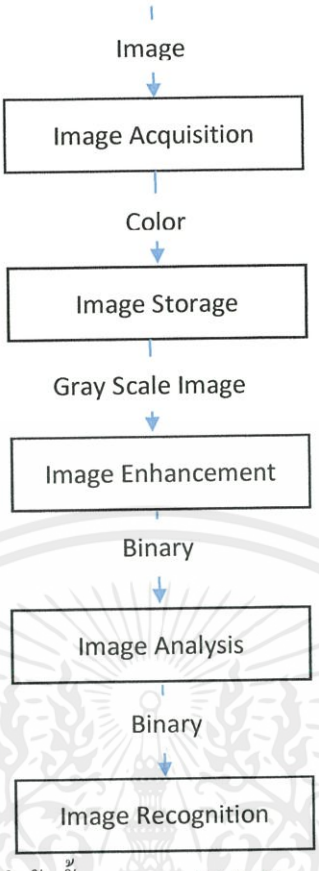
เป็นไลบรารีที่ใช้ในการโหลด (Load) และบันทึกภาพ ติดต่อกับกล้องวีดีโอ การสร้างหน้าต่างเพื่อแสดงภาพและทำลายภาพ การเปลี่ยนขนาดและเคลื่อนย้ายหน้าต่าง รวมไปถึงการตรวจสอบเม้าส์ (Mouse) และแป้นพิมพ์

2.3 หลักการของ Image processing

2.3.1 การประมวลผลภาพ

การประมวลผลภาพ คือ การแปลงข้อมูลภาพให้อยู่ในรูปแบบข้อมูลดิจิตอล ซึ่งสามารถที่จะนำเอาข้อมูลนี้มาวิเคราะห์และจัดการตามกระบวนการต่างๆด้วยคอมพิวเตอร์ได้ ซึ่งลำดับของ Digital Image Processing สามารถแสดงได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพ 2.4 ลำดับขั้นตอนของ Digital image processing

3. Image Acquisition เป็นกระบวนการให้ได้มาซึ่งข้อมูลภาพ โดยการแปลงสัญญาณอนาล็อกไปเป็นสัญญาณดิจิตอล ซึ่งสามารถประมวลผลได้ด้วยคอมพิวเตอร์

รูปที่ 2.5 Image acquisition process

4. Image Storage เป็นกระบวนการที่เก็บข้อมูลภาพ ซึ่งจะเก็บในรูปแบบสัญญาณดิจิตอลไว้ในหน่วยความจำ
5. Image Enhancement เป็นกระบวนการที่ประกอบด้วยเทคนิคที่หลากหลาย ในการจะนำมาซึ่งภาพที่มีความชัดเจนและคมชัดยิ่งขึ้นหรือเปนการแปลงภาพให้มีความเหมาะสมสำหรับการวิเคราะห์ภาพโดยมนุษย์หรือคอมพิวเตอร์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

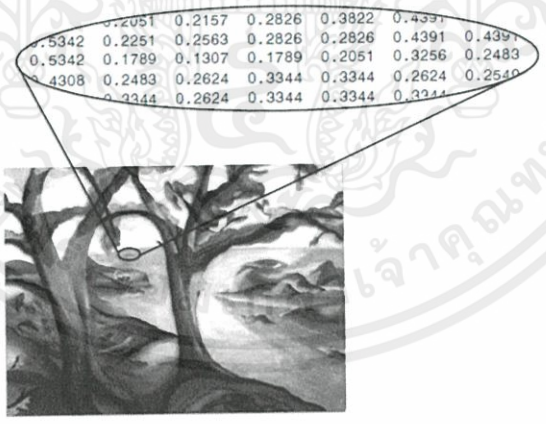
6. Image Analysis เกี่ยวกับวิธีการอธิบายและการจดจำข้อมูลภาพดิจิตอล ซึ่งอินพุตของระบบข้อมูลภาพดิจิตอลและเอาท์พุตจะเป็นเครื่องหมายที่ใช้แทนข้อมูลภาพดิจิตอลเหล่านั้น ในการวิเคราะห์ภาพมีอยู่หลายวิธีด้วยกันที่ได้นำมาจากการทำงานของภาพ
7. Image Recognition เป็นกระบวนการจดจำภาพ

การเก็บข้อมูลภาพลงหน่วยความจำของคอมพิวเตอร์สามารถทำได้โดยการจองหน่วยความจำของเครื่องไว้ในรูปของตัวแปรอาเรย์ (array) โดยค่าในแต่ละช่องของอาเรย์แสดงถึงคุณสมบัติของจุดภาพ (pixel) และตำแหน่งของช่องอาเรย์เป็นตัวกำหนดตำแหน่งของจุดภาพ

2.3.2 ชนิดของภาพ

2.3.2.1 ภาพระดับความเข้มเทา (Gray Scale Image)

ลักษณะของภาพชนิดนี้ ในแต่ละพิกเซลจะมีค่าความเข้มของแสงในแต่ละระดับที่แตกต่างกันไป ตั้งแต่ระดับเทาดำไปยังระดับสีขาว เราสามารถกำหนดระดับความเข้มของแสงนั้นได้โดยใช้ค่าระดับความเข้มเทา(Gray Scale หรือ Gray level) โดยปกติทั่วๆ ไป ภาพระดับสีเทาจะมีค่าระดับความเข้มเทาเท่ากับ 8 บิต ดังนั้นค่าความเข้มแสงจะถูกแบ่งออกเป็น 256 ระดับ เมื่อค่าระดับความเข้มเทาเป็น 0 จะหมายถึงจุดภาพนั้นมีค่าความเข้มของแสงต่ำ จะทำให้จุดภาพเป็นสีดำ ในทางกลับกัน หากค่าระดับความเข้มเทาเป็น 255 จะหมายถึงจุดภาพนั้นมีค่าความเข้มแสงมาก จะทำให้จุดภาพเป็นสีขาว ซึ่งสีขาวจะถูกแทนด้วยค่าความเข้มเทาเท่ากับ 255 (11111111) และสีดำจะถูกแทนด้วยค่าระดับความเข้มเทาเท่ากับ 0 (00000000) ส่วนค่าระหว่าง 0-255 ก็จะมีค่าไล่เฉดสีจากสีดำไปหาสีขาวนั่นเอง



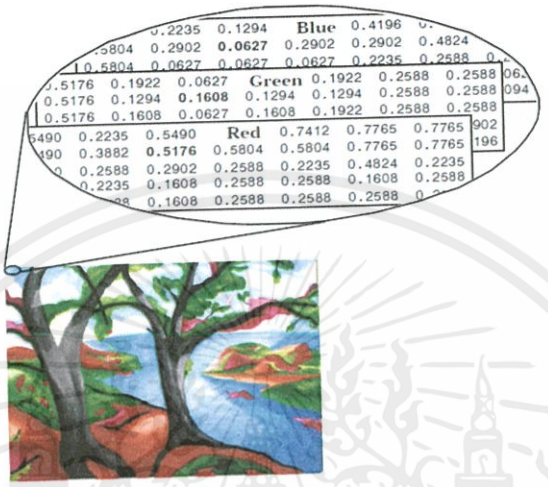
รูปที่ 2.6 ภาพระดับความเข้มเทาและแสดงการเก็บค่าระดับความเข้มเทา

ที่มา :http://www.mathworks.com/access/helpdesk/help/toolbox/images/[Access 15 พ.ค. 2015]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.2 ภาพสี (Color Image)

ภาพชนิดนี้ แต่ละจุดภาพหรือพิกเซลของภาพจะเก็บค่าระดับความเข้มเทาของแต่ละแถบแสงของแม่สีหลัก 3 สีที่ซ้อนกันอยู่คือ สีแดง (Red) สีเขียว (Green) สีน้ำเงิน (Blue) ซึ่งในแต่ละพิกเซลนั้นๆ ก็จะแสดงผลของค่าสีของแต่ละพิกเซลตามระดับความเข้มในแต่ละแถบแสดงสีนั้น ดังรูป

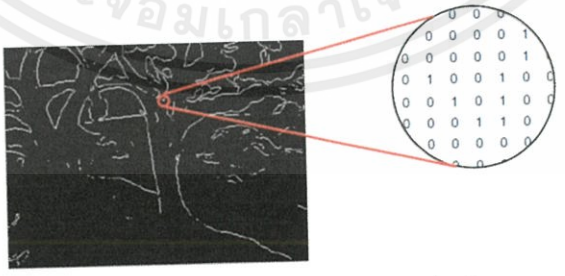


รูปที่ 2.7 ภาพสีและแสดงการเก็บค่าระดับความเข้มสี

ที่มา: http://www.mathworks.com/access/helpdesk/help/toolbox/images/[สืบค้นเมื่อ 15 พ.ค. 2015]

2.3.2.3 ภาพไบนารี (Binary Image)

ภาพไบนารีจะแสดงลักษณะของข้อมูลภาพในรูปแบบขาวดำ กล่าวคือ ในแต่ละพิกเซลของภาพจะถูกแสดงด้วยค่าแบบบารี (Binary) คือมี 1 บิต ซึ่งประกอบด้วยค่า 1 กับ 0 โดยที่ 1 หมายถึงจุดภาพสีขาวและ 0 หมายถึงจุดภาพสีดำ ภาพประเภทนี้เหมาะสำหรับภาพที่เกี่ยวกับตัวอักษร (Text) ภาพลายนิ้วมือ (Finger Print) เป็นต้น ดังรูป



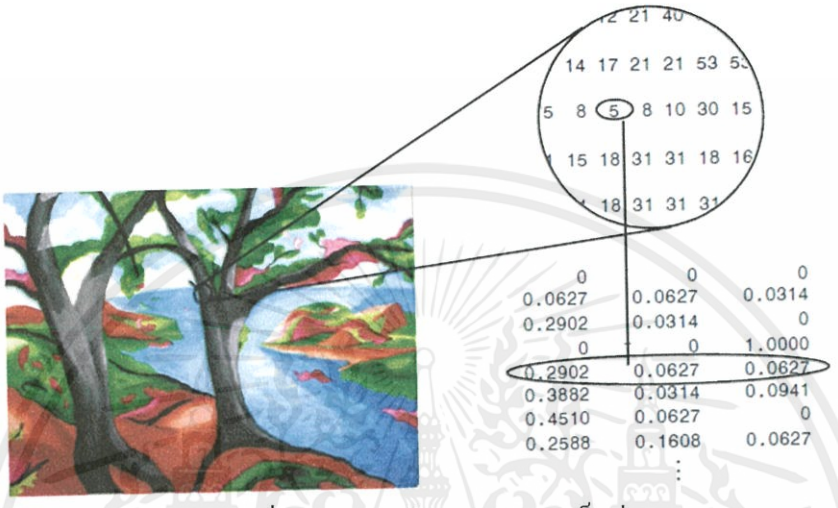
รูปที่ 2.8 ภาพขาวดำและแสดงการเก็บค่า

http://www.mathworks.com/access/helpdesk/help/toolbox/images/ [สืบค้นเมื่อ 15 พ.ค. 2015]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.4 ภาพแบบดัชนี (Index Image)

ภาพประเภทนี้ ในแต่ละพิกเซลของภาพจะเก็บค่าดัชนี (Index Number) ซึ่งเป็นตัวเลขจำนวนเต็มซึ่งจะถูกนำค่าดัชนีดังกล่าวไปเทียบกับตารางสี (Color Table) ซึ่งเป็นตารางแสดงค่าแสงสีแดง สีเขียว และสีน้ำเงิน ซึ่งค่าดัชนีนี้จะเป็นตัวบ่งชี้ให้เห็นว่าภาพในแต่ละตำแหน่งพิกเซลใดๆมีค่าอัตราส่วนของแม่แสง 3 สีในอัตราส่วนดังรูป



รูปที่ 2.9 ภาพดัชนีและแสดงการเก็บค่า

http://www.mathworks.com/access/helpdesk/help/toolbox/images/ [สืบค้นเมื่อ 15 พ.ค. 2015]

2.3.3 การแยกส่วนของภาพ (Image Segmentation)

การแยกส่วนของภาพ (Image Segmentation) เป็นเทคนิคหรือวิธีการประมวลผลภาพที่มีวัตถุประสงค์เพื่อแยกวัตถุ (Object) หรือลักษณะจุดเด่น (Feature) ในภาพออกจากพื้นหลัง (Background) ของภาพ วิธีการที่ใช้ในการแยกวัตถุหรือลักษณะเด่นออกจากพื้นหลังมีอยู่ 2 วิธีคือ

2.3.3.1 การหาขอบภาพ (Edge Detection)

ขอบของภาพ (Edge) คือส่วนของข้อมูลที่แสดงถึงโครงร่างของวัตถุภายในภาพ หรืออาจกล่าวได้ว่า ขอบต่างๆภายในภาพเกิดจากการเปลี่ยนแปลงของค่าระดับความเข้มเทาแบบทันทีทันใด จากค่าระดับต่ำๆไปเป็นค่าระดับความเข้มเทาสูงๆ หรือในทางตรงกันข้าม เปลี่ยนจากค่าระดับความเข้มเทาสูงๆไปเป็นค่าระดับความเข้มเทาของพิกเซลที่อยู่บริเวณติดกัน (Neighborhood Pixels) กล่าวคือ ค่าระดับความเข้มเทาของพิกเซลที่อยู่ติดกันมีค่าแตกต่างกันมาก ดังรูปที่ 2.3.1 จากรูปจะเห็นได้ว่าค่าระดับความเข้มเทาของข้อมูลภาพในหลักที่ 2 และ 3 มีความแตกต่างกันมาก ดังนั้นจึงสรุปได้ว่าบริเวณช่วงการเปลี่ยนแปลงแบบทันทีทันใดระหว่างค่าระดับความเข้มเทาของข้อมูลภาพในหลักที่ 2 และ 3 เป็นขอบของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50	50	160	160
50	50	160	160
50	50	160	160
50	50	160	160

รูปที่ 2.10 ค่าระดับความเข้มเทาของภาพที่มีการเปลี่ยนแปลงแบบทันทีทันใด

การหาขอบภาพมีอยู่ 2 วิธีคือ

- Gradient method วิธีนี้จะหาขอบโดยการหาจุดต่ำสุดและจุดสูงสุดในรูปของอนุพันธ์อันดับหนึ่งของภาพ โดยจุดที่เป็นขอบจะอยู่ในส่วนที่เหนือค่า Threshold จึงอาจทำให้เส้นขอบที่ได้มีลักษณะหนา ตัวอย่างวิธีการหาขอบของกลุ่มนี้ เช่น Roberts, Prewitt, Sobel และ Canny เป็นต้น

- Laplacian method จะหาขอบโดยใช้อนุพันธ์อันดับ 2 โดยใช้จุดที่ค่า y เป็น 0 (Zero crossing) ซึ่งวิธีนี้จะใช้เวลาในการค านวณมากกว่า Gradient method ตัวอย่างวิธีการหาขอบของกลุ่มนี้ เช่น Laplacian of Gaussian และ Marrs - Hildreth เป็นต้น

2.3.3.2 การทำเทรชโฮลด์ (Threshold)

การทำเทรชโฮลด์เป็นวิธีการหรือเทคนิคการประมวลผลภาพอย่างง่ายๆ เพื่อที่จะแบ่งแยกส่วนพื้นหน้า (Foreground) หรือวัตถุ (Object) ออกจากพื้นหลัง (Background) โดยใช้ค่าระดับเทรชโฮลด์หรือค่าระดับความเข้มเทาคงที่ค่าค่าหนึ่งเป็นตัวกำหนดในการแยกแยะส่วนของภาพ เพื่อทำให้ภาพผลลัพธ์ที่ได้เป็นภาพแบบไบนารี (Binary) ที่มีค่าระดับความเข้มเทาเพียง 2 ระดับเท่านั้นคือขาว และดำ ซึ่งในการกำหนดค่าเทรชโฮลด์นั้น ถ้าค่าเทรชโฮลด์ ที่เรากำหนดไว้มีค่าไม่เหมาะสม เช่น ค่าเทรชโฮลด์ที่มีค่าน้อยเกินไปหรือมากเกินไป หรือภาพอาจจะมีสิ่งไม่พึงประสงค์ปนมาด้วย เช่น สัญญาณรบกวน (Noise) ดังนั้นเราจะต้องมีการกำหนดค่าเทรชโฮลด์ที่เหมาะสม

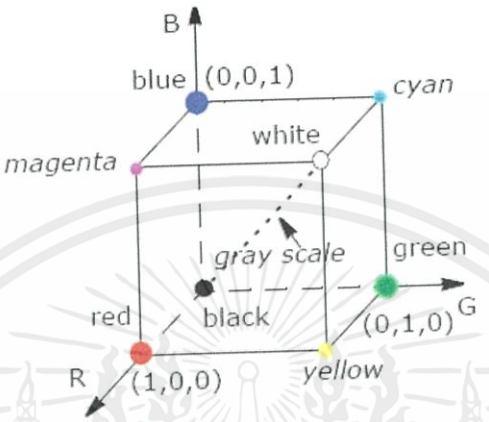
2.3.4 แบบจำลองสี (Color Model)

แบบจำลองสีเป็นแบบจำลองหรือโมเดลของสีที่ใช้กำหนดสีต่างๆ ให้เป็นแบบมาตรฐาน ซึ่งแบบจำลองสีในแต่ละแบบก็มีคุณสมบัติแตกต่างกันออกไป จึงเหมาะสำหรับการใช้งานที่แตกต่างกัน ในแบบจำลองของสีจะใช้แม่สีมาผสมผสานกลมกลืนกันเพื่อใช้ในการแสดงค่าสีอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4.1 แบบจำลองสี RGB

แบบจำลองนี้เป็นแบบจำลองของแม่สีหลัก 3 สี (Primary Color) ซึ่งเป็นสีที่เกิดจากการรวมตัวกันของแสง (Additive Color) ประกอบด้วยสีที่สำคัญ 3 สีด้วยกันได้แก่ สีแดง (Red) สีเขียว(Green) และสีน้ำเงิน(Blue) แบบจำลองสี RGB สามารถทำให้เกิดสีได้จำนวนมากที่สุดใกล้เคียงกับสีที่ตาของมนุษย์สามารถมองเห็นได้



รูปที่ 2.11 แบบจำลองสี RGB

ที่มา:Rafael C. Gonzalez and Richard E. Wood , 2002. Digital Image Processing 2nd Edition , หน้า 290.

2.3.4.2 แบบจำลองสี CMY

แบบจำลองสี CMY มีแม่สีหลักประกอบด้วยสีเขียวน้ำเงิน(Cyan) สีม่วงแดง(Magenta) และสีเหลือง (Yellow) คุณสมบัติของการแสดงสีจะใช้หลักการของการลบออก กล่าวคือ ถ้าแสงสีขาวตกกระทบหมึกที่มีสีเหลือง ซึ่งสีเหลืองเกิดจากสีแดงผสมกับสีเขียวในแบบจำลอง RGB ดังนั้น แสงที่สะท้อนออกมาจากสีเหลืองจะไม่มีสีน้ำเงิน เนื่องจากสีน้ำเงินถูกลบออกจากแสงสีขาว ทำให้เราเห็นแสงสะท้อนเป็นสีเหลือง แต่เนื่องจากแม่สีทั้งสามเมื่อนำมาผสมกันอย่างเท่าๆกันแล้วไม่สามารถทำให้เกิดเป็นสีดำจริงๆ ได้ จึงได้มีสีหลักเพิ่มมาคือสีดำ จึงเรียกว่า CMYK ซึ่งสี K หมายถึง Key Color

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (1.1)$$

สมการสำหรับแปลง RGB ให้อยู่ในรูป CMYK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

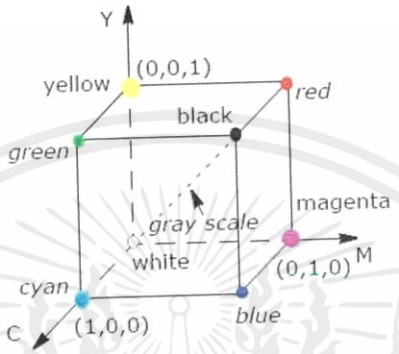
$$K = \min(C,M,Y)$$

$$C = C - K$$

$$M = M - K$$

$$Y = Y - K \qquad (1.2)$$

สมการของแบบจำลอง CMYK



รูปที่ 2.12 แบบจำลองสี CMYK

ที่มา: https://software.intel.com/en-us/node/503873 [สืบค้นเมื่อ 15 พ.ค. 2015]

2.3.4.3 แบบจำลองสี YUV

แบบจำลองสี YUV เป็นแบบจำลองที่ใช้ในสัญญาณวิดีโอแบบแอนะล็อก ในระบบPAL (Phase Alternating Line) และ SECAM (Sequential Color with Memory) โดย แบบจำลองสี YUV มีการใช้งานอยู่ในมาตรฐานของ CCIR 601

Y จะเก็บข้อมูลของความเข้มแสงหรือค่าความสว่าง (Luminance) ส่วน U และ V จะใช้เก็บข้อมูลของแสงสี (Chrominance)

2.3.4.4 แบบจำลอง YIQ

แบบจำลองนี้ถูกนำมาใช้กับระบบโทรทัศน์สีและวิดีโอในสหรัฐอเมริกา และประเทศอื่นๆที่ใช้ระบบ NTSC(The National Television Standards Commission) ใน แบบจำลองนี้ Y เป็นค่าความเข้มแสงหรือค่าความสว่าง (Luminance) ส่วน I และ Q จะใช้แทน ข้อมูลสี โดยที่ I จะเป็นแกนสีน้ำเงินและสีส้ม ส่วน Q จะเป็นแกนสีม่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

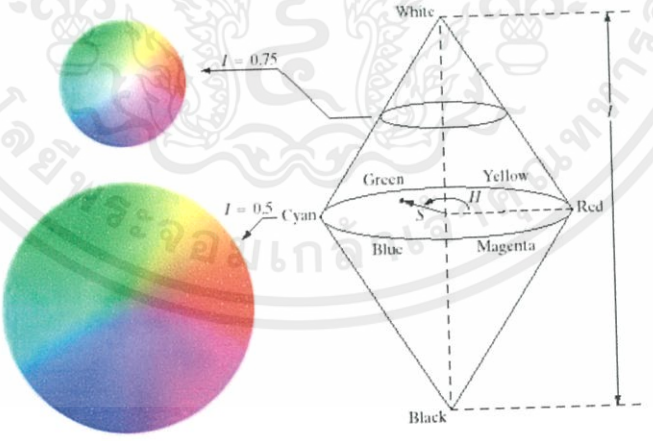
2.3.4.5 แบบจำลองสี YCbCr

แบบจำลองสี YCbCr เป็นแบบจำลองสีที่มีความคล้ายคลึงกับแบบจำลองสี YUV มาก เพียงแต่มีการเปลี่ยนแปลงสัญญาณซึ่งถูกปรับเปลี่ยนมาจากแบบจำลองสี YUV และแบบจำลองสี YCbCr นี้ยังมีการประยุกต์ใช้งานในกระบวนการบีบอัดข้อมูลของภาพแบบ JPEG และขนาดของข้อมูลวิดีโอ MPEG แบบจำลองสี YCbCr นี้ค่า Y จะใช้เก็บข้อมูลของความเข้มแสง (Luminance) ส่วน Cb และ Cr จะใช้เก็บข้อมูลของแสงสี

2.3.4.6 แบบจำลองสี HIS

แบบจำลองสีนี้เป็นแบบจำลองที่เหมาะสมและสอดคล้องกับการรับรู้และการมองเห็นค่าสีต่างๆ ของมนุษย์

- H (Hue) ใช้แทนข้อมูลสีที่สามารถกำหนดได้จากมุมที่หมุนไปในวงล้อสีมาตรฐาน หรือการไล่โทนสีเป็นวง 360 องศานั่นเอง ซึ่งวงจรที่เกิดขึ้นจากการพยายามแยกเฉดสีออกมาจากแสง ซึ่งเป็นอย่างเดียวกับปรากฏการณ์การเกิดสีของรุ้ง
- S(Saturation) คือค่าความบริสุทธิ์หรือความอิ่มตัวของสี ซึ่งเป็นการกำหนดว่าสีนั้นๆ จะมีสีเทามาเจือผสมมากน้อยเท่าไร ค่า saturation มีหน่วยวัดเป็นอัตราส่วนระหว่างสีจริงกับสีเทา หาก S เท่ากับ 1 ก็จะเป็นค่าสีอิ่มตัวมากสุด แต่ถ้าค่า S เท่ากับ 0 แสดงว่าไม่มีสีจริงอยู่เลยจึงกลายเป็นสีดำ
- I (Intensity) คือค่าความสว่างของสี โดยหากค่า I เท่ากับ 0 จะกลายเป็นสีดำ หาก I มีค่าเป็น 1 จะกลายเป็นสีขาว



รูปที่ 2.13 แบบจำลองสี HIS

ที่มา:Rafael C. Gonzalez and Richard E. Wood , 2002. Digital Image Processing 2nd Edition , หน้า 298.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

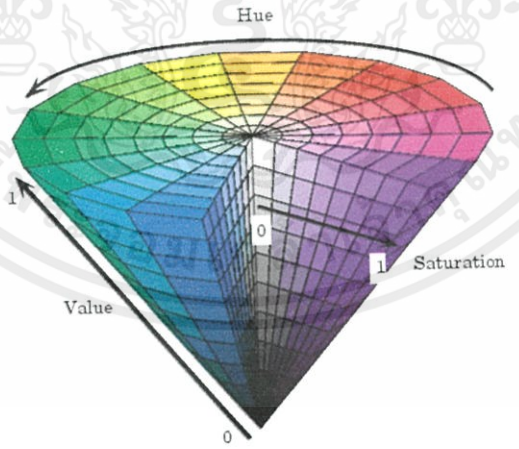
2.3.4.7 แบบจำลองสี HSV

H (Hue) คือค่าของแม่สีหลัก (สีแดง สีเขียว สีน้ำเงิน) ในทางปฏิบัติจะอยู่ระหว่าง 0-1 ซึ่งถ้า Hue มีค่าเท่ากับ 0 จะแทนด้วยสีแดง และเมื่อ Hue มีค่าเพิ่มขึ้นเรื่อยๆ สีก็จะเปลี่ยนแปลงไปตามสเปกตรัมของสีจนถึงค่า 1 จึงจะกลับมาเป็นสีแดงอีกครั้ง ดังตาราง

ตารางที่ 2.1 ค่า Hue

สี	ค่าระดับสี (Hue)
แดง	0
เหลือง	0.1667
เขียว	0.333
น้ำเงินเขียว	0.5
น้ำเงิน	0.6667
ม่วงแดง	0.8333

- S (Saturation) คือความบริสุทธิ์ของสี ซึ่งค่า Saturation มีค่าเท่ากับ 0 แล้วสีที่ได้จะไม่มีสี ซึ่งจะเป็นสีขาวล้วน แต่ถ้า Saturation มีค่าเท่ากับ 1 แสดงว่าจะไม่มีแสงสีขาวผสมอยู่เลย
- V (Value) คือค่าความสว่างของสี ซึ่งสามารถวัดได้จากค่าความเข้มของความสว่างของแต่ละสีที่ประกอบกัน



รูปที่ 2.14 แบบจำลองสี HSV

ที่มา: http://www.mathworks.in/help/images/converting- color-data-between-color-spaces.html [สืบค้นเมื่อ15 พ.ค. 2015]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 หลักการติดตามวัตถุ(Object Tracking)

การติดตามวัตถุ (Object Tracking) คือ การที่เราสามารถทราบว่าวัตถุอยู่ส่วนใดในบริเวณที่สนใจตลอดเวลา ยกตัวอย่างเช่น เมื่อเราได้รับลำดับภาพมาจำนวนหนึ่ง เราจะน้องสามารถทราบว่ามีวัตถุอยู่ ณ จุดตำแหน่งใดในทุกๆภาพ เมื่อวัตถุมีการเคลื่อนที่ ภาพที่ปรากฏบนกล้องจะมีการเปลี่ยนแปลงตามเวลา ซึ่งในการติดตามวัตถุนั้นมีการเคลื่อนที่อย่างไร ทิศทางใด ตำแหน่งปัจจุบันของวัตถุเป็นอย่างไร และตำแหน่งแตกต่างจากตำแหน่งของภาพก่อนหน้าอย่าไร ซึ่งวิธีการติดตามการเคลื่อนที่ของวัตถุนั้นมีหลายวิธีด้วยกัน ดังนี้

2.4.1 การหาความสมนัยกันของภาพ (Correlation based tracking)

การหาความสมนัยกันของภาพ (Correlation based tracking) เป็นวิธีการที่ใช้กันในยุคแรก โดยจะใช้การเปรียบเทียบความเหมือนกัน ของระดับสีเทาบนพื้นที่หนึ่ง ในภาพจากช่วงเวลาต่างๆ คือ โดยกำหนดแบ่งภาพออกเป็นช่องๆ จากนั้นจะเปรียบเทียบข้อมูลจากภาพในเวลาถัดไป ว่าเหมือนกับช่องไหนมากที่สุด โดยที่ช่องทั้งหมดจะต้องมีขนาดเท่ากัน ดังนั้นจะรู้ว่าช่องนั้นเคลื่อนที่ไปที่ตำแหน่งใด ในเฟรมถัดไป แต่ข้อเสียของวิธีนี้คือ วัตถุที่ปรากฏในแต่ละเฟรมจะต้องมีการเปลี่ยนแปลงน้อย ควรจะเป็นวัตถุแข็งแกร่ง (rigid body) หรือเป็นวัตถุที่มีการเปลี่ยนแปลงรูปร่างและมุมมองไม่มากนัก

2.4.2 การติดตามการไหลของแสง(Optical flow based tracking)

การติดตามการไหลของแสง(Optical flow based tracking) เป็นอีกวิธีหนึ่งซึ่งใกล้เคียงกับ correlation based tracking โดยจะหาการเคลื่อนที่ในภาพจากจุดภาพ ซึ่งจะวิเคราะห์ความเหมือนกัน ความสว่างของจุดภาพในแต่ละภาพ กล่าวคือ เมื่อมีวัตถุเคลื่อนที่ปรากฏในลำดับภาพ ในแต่ละภาพ จุดภาพที่วัตถุเคลื่อนที่ผ่านจะเปลี่ยนแปลงความสว่าง โดยจะสมมติว่าความสว่างของวัตถุจะเท่ากันตลอดทุกๆเฟรม ทำให้เป็นข้อเสียของการติดตามการไหลของแสง คือจะไวต่อการเปลี่ยนแปลงของแสง

2.4.3 การติดตามวัตถุจากลักษณะเฉพาะ(Feature based object tracking)

การติดตามวัตถุจากลักษณะเฉพาะ(Feature based object tracking) เป็นวิธีการหนึ่งที่แก้ปัญหาการติดตามที่ไวต่อการเปลี่ยนแปลงของแสง โดยมีหลักการทำงานคือ จะตรวจหาวัตถุและลักษณะของวัตถุจากลำดับภาพก่อน เพื่อนำไปใช้ในการเปรียบเทียบเพื่อระบุหาวัตถุในภาพโดยแบ่งการทำงานออกเป็น 2 ขั้นตอนย่อย ดังนี้ ขั้นตอนแรกคือ การติดตามหาลักษณะเฉพาะ (Feature extraction) และการติดตามลักษณะเฉพาะ(Feature tracking) โดยลักษณะเฉพาะที่นำมาใช้ในการติดตามจะต้องเป็นลักษณะที่มีความโดดเด่นและมีการเปลี่ยนแปลงไม่มากนัก ยกตัวอย่างเช่น มุม ขอบวัตถุ และสี เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 การติดตามโดยใช้ข้อมูลการเคลื่อนที่ของวัตถุ (Motion correspondence)

การติดตามโดยใช้ข้อมูลการเคลื่อนที่ของวัตถุ (Motion correspondence) เป็นวิธีการติดตามวัตถุโดยการวิเคราะห์วิถีการเคลื่อนที่ของวัถตุ(Trajectory) และตำแหน่งทั้งหมดของวัตถุที่ปรากฏในแต่ละภาพ และนำมาพิจารณาหาตำแหน่งวัตถุที่ควรจะเป็น เมื่อทราบข้อมูลการเคลื่อนที่เหล่านี้ เช่น ความเร็ว(velocity) ความเร่ง(acceleration) ทิศทางการเคลื่อนที่(direction) และความเร็วเชิงมุม(angle velocity) ซึ่งจะสามารถทำให้เราหาตำแหน่งในเฟรมถัดไปได้ ทำให้สามารถช่วยลดพื้นที่ในการค้นหาวัตถุ หรือการนำไปใช้ร่วมกับวิธีการอื่น เพื่อเพิ่มความแม่นยำ และลดเวลาในการติดตามวัตถุ

2.4.5 การใช้ลักษณะเด่นของวัถตุ (Explicit object representation)

การใช้ลักษณะเด่นของวัถตุ (Explicit object representation) กรณีที่เราทราบถึงลักษณะเด่นของวัตถุที่เราต้องการติดตาม เช่น รูปร่าง หรือลักษณะเด่นของวัตถุ เราสามารถใช้ลักษณะเหล่านี้ในการสร้างแบบจำลอง ลักษณะเด่นของวัตถุเพื่อใช้ในการติดตามต่อไปได้ ข้อดีของวิธีการนี้คือ ถ้าหากการติดตามผิดพลาดจากการไม่ตรงกันระหว่างแบบจำลองกับวัตถุ ระบบจะสามารถกู้คืนการติดตามได้โดยใช้แบบจำลองลักษณะเด่นที่ตรงกับวัตถุที่ติดตาม ซึ่งการสร้างแบบจำลองวัตถุนี้แบ่งออกเป็น 3 วิธีดังนี้

1. ใช้แบบจำลอง 3 มิติ(3-D Model representation) ตัวอย่าง ระบบงานประยุกต์ที่ประสบความสำเร็จในการตรวจจับคือ การติดตามรถยนต์บนถนนโดยใช้แบบจำลองโครงสร้างของรถยนต์แบบ 3 มิติ
2. การใช้แบบจำลองมุมมองต่างๆของวัตถุ (View based representation) ซึ่งเป็นแบบจำลองในลักษณะ 2 มิติ ในกรณีที่ไม่มีข้อมูล 3 มิติ เช่น การใช้เส้นรอบรูปความสว่างของวัตถุ รวมกับความสัมพันธ์แบบ 2 มิติ ระหว่างลักษณะดังกล่าวเช่น การติดตามวัตถุที่มีลักษณะเป็นข้อปล้องคือ การติดตามแขนขา หรือมือของมนุษย์
3. การใช้แบบจำลองลักษณะที่ปรากฏขึ้นของวัตถุ(Appearance based representation) จะเป็นการใช้วิธีการสร้างชุดข้อมูลรูปร่างลักษณะของวัตถุที่ปรากฏในหลายๆ มุมมอง และสรุปว่าสามารถที่จะพบลักษณะที่ตรงกับวัตถุได้ภายในชุดข้อมูลนี้ ตัวอย่างเช่น การติดตามวัตถุแข็งแกร่ง (rigid object) ซึ่งข้อเสียของวิธีนี้คือ ถ้ามีลักษณะที่ปรากฏขึ้นมา และไม่ตรงกับชุดข้อมูลจะทำให้การติดตามผิดพลาดได้ เช่น การที่วัตถุถูกบัง และการเปลี่ยนแปลงของแสงเป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 Motion Detection

Motion Detection คือการหาภาพเคลื่อนไหวจากภาพวิดีโอ โดยการหาภาพเคลื่อนไหวจากภาพวิดีโอนั้นสามารถหาได้ด้วยวิธี Background Subtraction และการคำนวณหา Optical Flow ของภาพ

2.5.1 Background Subtraction

Background Subtraction คือ การแยกวุ้ตถุที่มีการเคลื่อนที่ออกจากฉากหลัง โดยการลบฉากหลัง ด้วยการนำภาพสองเฟรม ณ บริเวณเดียวกันในเวลาที่ต่างกันมาทำการลบกัน หากมีการเคลื่อนที่เข้ามาของวัตถุในภาพเฟรมที่สอง เมื่อนำภาพทั้งสองเฟรมมาลบกันฉากหลังจะลบหักล้างกันระหว่างภาพสองเฟรม และมีส่วนผลต่างระหว่างภาพทั้งสองนั่นแสดงว่า มีการเคลื่อนไหวของวัตถุเข้ามายังภาพเฟรมที่สอง แต่ถ้าหากไม่มีผลต่างเกิดขึ้นแสดงว่าก็ไม่มีอะไรเกิดขึ้นบนภาพเฟรมที่สอง

การหาผลต่างของเป็นไปดังสมการต่อไปนี้

$$|\text{foreground}(x,y) - \text{background}(x,y)| \quad (1.3)$$

จากสมการข้างต้น เมื่อเรานำภาพทั้งสองเฟรมมาลบกันแล้ว ต่อไปก็ทำการเทรชโฮลด์ (Threshold) ต่อเพื่อเป็นการแยกวัตถุที่เคลื่อนไหวในภาพกับวัตถุที่ไม่ได้เคลื่อนไหว ซึ่งผลลัพธ์จะขึ้นอยู่กับค่าเทรชโฮลด์ ถ้าปรับให้มีค่ามากเกินไปจะทำให้มี noise เยอะ แต่ถ้าปรับไว้น้อยเกินไป จะทำให้ได้พื้นที่ของวัตถุที่กำลังเคลื่อนไหวมีน้อยเกินไป

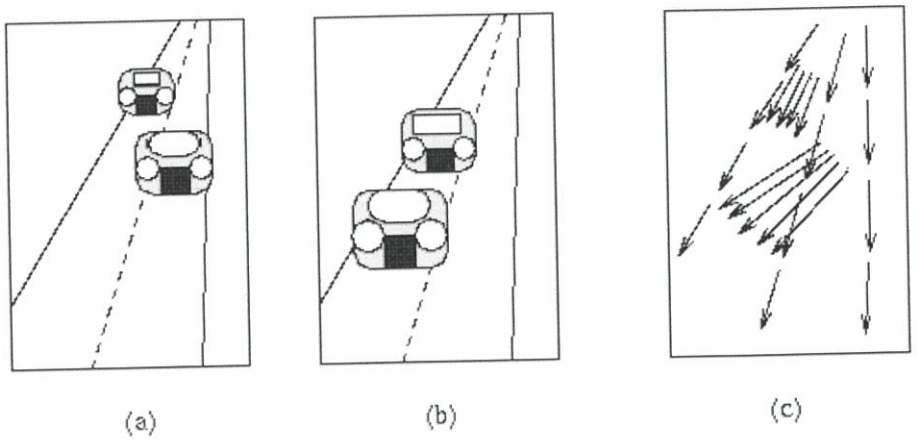
ข้อจำกัดของวิธีนี้คือ ต้องตั้งกล้องให้อยู่กับที่เท่านั้น เพื่อไม่ต้องการให้มีการเคลื่อนไหวของพื้นหลัง เพราะเราต้องใช้ภาพพื้นหลังในการแยกหาวัตถุที่เคลื่อนที่เข้ามาในภาพ

2.5.2 Optical Flow

Optical Flow คือ โมเดลการคำนวณหาเว็กเตอร์ลัพธ์ของภาพเคลื่อนไหว โดยจะวิเคราะห์ความเหมือนกันของแสงสว่างของจุดภาพในแต่ละภาพ กล่าวคือ เมื่อมีวัตถุเคลื่อนที่ปรากฏในลำดับภาพในแต่ละภาพ จุดภาพที่วัตถุเคลื่อนที่ผ่านจะเปลี่ยนแปลงความสว่าง โดยจะสมมุติว่า ความสว่างของวัตถุข้อจำกัดของการติดตามการไหลของแสง คือไวต่อการเปลี่ยนแปลงของแสงจะเท่ากันตลอดทุก ๆ เฟรม

ภาพเส้นทิศทางแสดงการเคลื่อนที่ของวัตถุต่าง ๆ ที่อยู่ในภาพ จาก Frame ที่ t1 กับ Frame ที่ t2 เป็นดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 ตัวอย่าง Optical flow

ที่มา: http://user.engineering.uiowa.edu/~dip/lecture/Motion2.html

[สืบค้นเมื่อ 15 พ.ค. 2015]

(a) รถสองคันกำลังวิ่งอยู่บนถนนโดยที่รถคันหน้าได้ทำการแซงคันข้างหลังแล้ว

(b) แสดงให้เห็นว่ารถคันหน้าที่แซงมานั้นกำลังเคลื่อนที่เข้าเลนส์ขวาของถนน

(c) แสดงทิศทางการไหลของวัตถุในภาพ นั่นคือ

- ทิศทางการไหลของรถคันหลังที่มีทิศทางวิ่งมาข้างหน้า
- ทิศทางการไหลของรถคันหน้าที่มีทิศทางวิ่งเข้าเลนส์ขวาของถนน

2.6 ความเร็ว

ความเร็ว คือ อัตราการเปลี่ยนแปลงของตำแหน่งต่อหน่วยเวลา มีหน่วยเป็นเมตรต่อวินาที (m/s) ในหน่วยเอสไอ ความเร็วเป็นปริมาณเวกเตอร์ซึ่งประกอบด้วยอัตราเร็วและทิศทาง ขนาดของความเร็วคืออัตราเร็วซึ่งเป็นปริมาณสเกลาร์ ตัวอย่างเช่น "5 เมตรต่อวินาที" เป็นอัตราเร็ว ในขณะที่ "5 เมตรต่อวินาทีไปทางทิศตะวันออก" เป็นความเร็ว ความเร็วเฉลี่ย v ของวัตถุที่เคลื่อนที่ไปด้วยการกระจัดขนาดหนึ่ง Δx ในช่วงเวลาหนึ่ง Δt สามารถอธิบายได้ด้วยสูตรนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\bar{v} = \frac{\Delta x}{\Delta t} \qquad (1.4)$$

โดยที่ $\bar{v}$ = อัตราเร็วของวัตถุ

Δx = ระยะทางการกระจัด

Δt = ช่วงเวลาที่ใช้ในการเปลี่ยนแปลงอัตราเร็ว

อัตราการเปลี่ยนแปลงของความเร็วคือ ความเร่ง คือการอธิบายว่าอัตราเร็วและทิศทางของวัตถุเปลี่ยนไปอย่างไรในช่วงเวลาหนึ่ง และเปลี่ยนไปอย่างไร ณ เวลาหนึ่ง

สมการการเคลื่อนที่

สมการของความเร็วของวัตถุยังสามารถหาได้จากปริพันธ์ของสมการของความเร่งที่วัตถุเคลื่อนที่ตั้งแต่เวลา t_0 ไปยังเวลา t_n

วัตถุที่มีความเร็วเริ่มต้นเป็น u มีความเร็วสุดท้ายเป็น v และมีความเร่งคงตัว a ในช่วงเวลาหนึ่ง Δt ความเร็วสุดท้ายหาได้จาก

$$v = u + a\Delta t \qquad (1.5)$$

โดยที่ v = ความเร็วสุดท้าย

u = ความเร็วเริ่มต้น

a = อัตราเร่งคงตัวของวัตถุ

Δt = ช่วงเวลาที่ใช้ในการเปลี่ยนแปลงอัตราเร็ว

ความเร็วเฉลี่ยอันเกิดจากความความเร่งคงตัวจึงเป็น $\frac{(u+v)}{2}$ ตำแหน่ง x ที่เปลี่ยนไปของวัตถุดังกล่าวในช่วงเวลานั้นหาได้จาก

$$\Delta x = \frac{(u+v)}{2}\Delta t \qquad (1.6)$$

โดยที่ v = ความเร็วสุดท้าย

u = ความเร็วเริ่มต้น

Δx = ตำแหน่งที่เปลี่ยนไปของวัตถุ

Δt = ช่วงเวลาที่ใช้ในการเปลี่ยนแปลงอัตราเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ทราบเพียงความเร็วเริ่มต้นของวัตถุเพียงอย่างเดียว คำนวณได้ดังนี้

$$\Delta x = u\Delta t + \frac{1}{2}a\Delta t^2 \qquad (1.7)$$

โดยที่ Δx = ตำแหน่งที่เปลี่ยนไปของวัตถุ

u = ความเร็วเริ่มต้น

a = อัตราเร่งคงตัวของวัตถุ

Δt = ช่วงเวลาที่ใช้ในการเปลี่ยนแปลงอัตราเร็ว

2.7 การเคลื่อนที่ของวัตถุ

การเคลื่อนที่ คือการเปลี่ยนตำแหน่งของวัตถุในช่วงเวลาหนึ่ง ซึ่งวัดโดยผู้สังเกตที่เป็นส่วนหนึ่งของกรอบอ้างอิง เมื่อปลายคริสต์ศตวรรษที่ 19เซอร์ไอแซก นิวตัน ได้เสนอกฎการเคลื่อนที่ของนิวตันในหนังสือ *Principia* ของเขา ซึ่งต่อมาได้กลายเป็นกฎพื้นฐานของฟิสิกส์ดั้งเดิม การคำนวณการเคลื่อนที่ของวัตถุต่างๆ โดยใช้ฟิสิกส์ดั้งเดิมนั้นประสบความสำเร็จมาก จนกระทั่งนักฟิสิกส์เริ่มศึกษาเกี่ยวกับสิ่งที่เคลื่อนที่ด้วยความเร็วสูงมาก

2.7.1 ลักษณะการเคลื่อนที่ของวัตถุ

ลักษณะของการเคลื่อนที่แบ่งได้ 4 ลักษณะ คือ

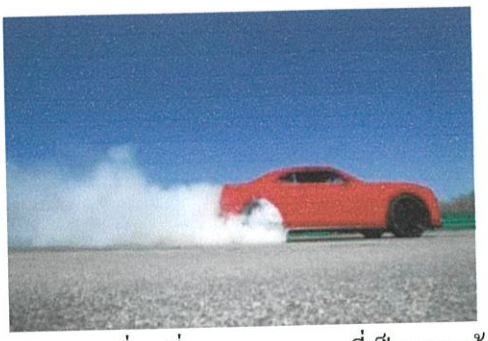
2.7.1.1 การเคลื่อนที่เป็นแนวเส้นตรง

ลักษณะของการเคลื่อนที่แบบนี้เป็นพื้นฐานของการเคลื่อนที่ เพราะทิศทางการเคลื่อนที่จะมีทิศทางเดียวแต่อาจจะเคลื่อนที่ไป-กลับได้ รูปแบบการเคลื่อนที่อาจจะแตกต่างกันออกไป ดังรูป

รูปที่ 2.16 การเคลื่อนที่ของรถไฟบนราง

ที่มา: http://wiki.stjohn.ac.th/groups/poly_ordinarycourse/wiki/a5bb1/ [สืบค้นเมื่อ 15 พ.ค. 2015]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

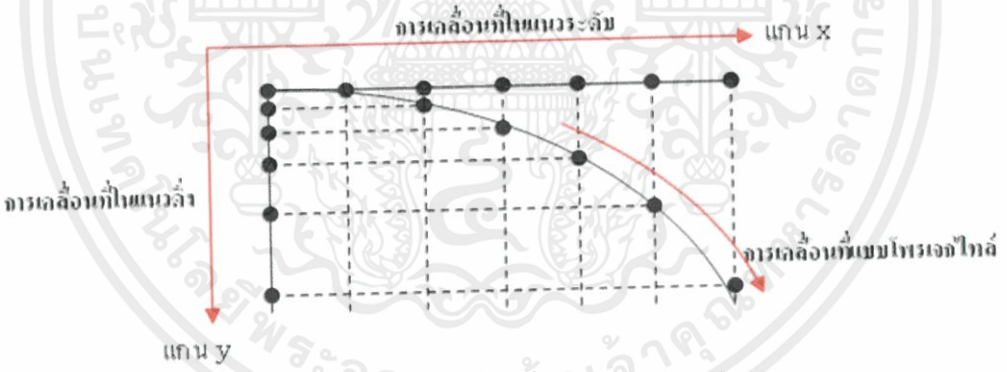


รูปที่ 2.17 การเคลื่อนที่ของรถบนถนนที่เป็นแนวเส้นตรง

ที่มา: http://wiki.stjohn.ac.th/groups/poly_ordinarycourse/wiki/a5bb1/ [สืบค้นเมื่อ 15 พ.ค. 2015]

2.7.1.2 การเคลื่อนที่แบบโปรเจกไทล์

การเคลื่อนที่แบบโปรเจกไทล์ เป็นการเคลื่อนที่ 2 แนวพร้อมกัน คือ แนวระดับ และแนวดิ่ง ซึ่งพบว่า ความเร็วในแนวระดับ ไม่มีผลต่อการเคลื่อนที่ในแนวดิ่ง โดยจากการทดลองปล่อยวัตถุให้ตกอย่างอิสระ พร้อมกับวัตถุที่ถูกดีดออกไปในแนวระดับ พบว่า เมื่อใช้แรงมากวัตถุที่ถูกดีดจะตกไกล แต่ตกถึงพื้นพร้อมกับวัตถุที่ตกในแนวดิ่ง แสดงว่า การเคลื่อนที่ในแนวระดับไม่มีผลต่อการเคลื่อนที่ในแนวดิ่ง ดังนั้นเราจึงแยกพิจารณาการเคลื่อนที่ออกเป็น 2 แนว คือ ในแนวดิ่ง และในแนวระดับ



รูปที่ 2.18 การเคลื่อนที่แบบโปรเจกไทล์

ที่มา: https://sites.google.com/site/supod45/sc30113/12557/room12/kar-kheluxnthi-baeb-pho-r-cek-thil-3 [สืบค้นเมื่อ 15 พ.ค. 2015]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **การเคลื่อนที่ในแนวระดับ**

พิจารณาการเคลื่อนที่ของวัตถุในอากาศ ขณะที่วัตถุลอยในอากาศ มีแรงกระทำต่อวัตถุเพียงแรงเดียว คือ

$$m\vec{g} \qquad (1.8)$$

โดยที่ m = มวลของวัตถุ

$\vec{g}$ = แรงโน้มถ่วงโลก มีค่า 9.807 เมตรต่อวินาทีกำลังสอง

(ไม่คิดแรงต้านอากาศ) ในทิศเข้าสู่ศูนย์กลางของโลก หรือในแนวดิ่ง ดังนั้น ในแนวระดับจึงไม่มีแรงกระทำต่อวัตถุ หรือแรงลัพธ์ที่กระทำต่อวัตถุในแนวระดับเป็นศูนย์

$$(\sum F_x = 0) \qquad (1.9)$$

โดยที่ F_x = แรงลัพธ์ที่กระทำต่อวัตถุในแนวระดับ

จากกฎการเคลื่อนที่ของนิวตัน เราพบว่า เมื่อไม่มีแรงลัพธ์มากระทำต่อวัตถุ วัตถุจะรักษาสภาพการเคลื่อนที่ให้คงที่ ผลคือ วัตถุเคลื่อนที่ด้วยความเร็วคงที่

$$u_x = v_x = \text{ความเร็วในแนวแกน x} \qquad (2.0)$$

ดังนั้น การเคลื่อนที่ในแนวระดับของการเคลื่อนที่แบบโปรเจกไทล์สามารถหาการกระจัดในแนวระดับได้ตามสมการ

$$S_x = u_x t \qquad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ S_x = การกระจัดในแนวระดับ

u_x = ความเร็วในแนวระดับ

t = ช่วงเวลาของการเคลื่อนที่

- **การเคลื่อนที่ในแนวดิ่ง**

เมื่อพิจารณาการเคลื่อนที่ในแนวดิ่ง พบว่า วัตถุมีแรงกระทำในแนวดิ่ง คือ แรงโน้มถ่วงของโลก

$$m\vec{g} \qquad (1.8)$$

โดยที่ m = มวลของวัตถุ

$\vec{g}$ = แรงโน้มถ่วงโลก มีค่า 9.807 เมตรต่อวินาทีกำลังสอง

2.7.1.3 การเคลื่อนที่แบบวงกลม

รูปที่ 2.19 การเคลื่อนที่แบบวงกลม

ที่มา: http://fieldtrip.ipst.ac.th/intro_sub_content.php?content_id=18&content_folder_id=188 [สืบค้นเมื่อ 15 พ.ค. 2015]

วัตถุที่กำลังเคลื่อนที่เป็นวงกลม จะต้องมีแรงกระทำต่อวัตถุในทิศเข้าหาศูนย์กลางของวงกลม เรียกว่า แรงสู่ศูนย์กลาง(centripetal force) แรงนี้จะทำให้วัตถุเปลี่ยนทิศของความเร็วซึ่งอยู่ในแนวเส้นสัมผัสของวงกลมทำให้วัตถุวิ่งเป็นวงกลมอยู่ได้ หากไม่มีแรงสู่ศูนย์กลางกระทำต่อวัตถุ วัตถุจะไม่เคลื่อนที่เป็นวงกลมแต่จะวิ่งไปในแนวเส้นสัมผัสแรงที่ทำหน้าที่เป็นแรงสู่ศูนย์กลางอาจเป็นแรงเสียดทานที่กระทำต่อวัตถุเมื่อวัตถุสัมผัสกับพื้น หรือเป็นแรงที่พื้นกระทำต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัตถุซึ่งแยกแรงมาอยู่ในแนวเข้าสู่ศูนย์กลางได้ กรณีผูกด้วยเชือกแล้วแกว่งเป็นวงกลมในแนวระดับ แรงดึงของเชือกก็จะทำหน้าที่เป็นแรงสู่ศูนย์กลาง เป็นต้น

การเคลื่อนที่แบบวงกลมมีลักษณะเฉพาะ โดยมีการเคลื่อนที่เป็นรอบ จำนวนรอบที่วัตถุเคลื่อนที่ได้ใน 1 หน่วยเวลา เรียกว่า ความถี่ (frequency) ใช้สัญลักษณ์ f มีหน่วยเป็นรอบต่อวินาทีหรือ เฮิรตซ์ (Hz) และถ้าวัตถุกำลังเคลื่อนที่เป็นวงกลมที่มีรัศมี r ด้วยอัตราสม่ำเสมอ อัตราเร็วของวัตถุ v จะหาได้จากสมการ

$$v = 2\pi rf \qquad (2.2)$$

โดยที่ v = อัตราเร็วของวัตถุ

π = ค่าคงที่ มีค่าเท่ากับ $\frac{22}{7}$

r = รัศมีของวงกลม

f = ความถี่

ขณะวัตถุกำลังเคลื่อนที่เป็นวงกลม แรงที่ทำหน้าที่เป็นแรงสู่ศูนย์กลาง (F_c) จะต้องมีขนาดเท่าใดนั้นจะสัมพันธ์กับมวลของวัตถุ (m) อัตราเร็วของวัตถุ (v) และรัศมีการเคลื่อนที่ (r) ตามสมการ

$$F_c = mv^2/r \qquad (2.3)$$

โดยที่ F_c = แรงหนีศูนย์กลาง

m = มวลของวัตถุ

v = อัตราเร็วของวัตถุ

r = รัศมีของวงกลม

2.7.1.4 การเคลื่อนที่แบบฮาร์มอนิกอย่างง่าย

ลักษณะของการเคลื่อนที่แบบฮาร์มอนิกอย่างง่าย จะเป็นการเคลื่อนที่ที่มีลักษณะพิเศษ คือ วัตถุจะเคลื่อนที่กลับไปกลับมาที่เราเรียกว่า แกว่ง หรือ สั่น การเคลื่อนที่แบบนี้จะเป็นการเคลื่อนที่อยู่ในช่วงสั้นๆ มีขอบเขตจำกัด เราเรียกว่า แอมพลิจูด (Amplitude) โดยนับจากตำแหน่งสมดุล ซึ่งอยู่ตรงจุดกลางวัดไปทางซ้ายหรือขวา เช่น การแกว่งของชิงช้า หรือยานไวกิงในสวนสนุก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2 การบอกตำแหน่งและการกระจัด

2.7.2 1. ตำแหน่งและการกระจัด

ในขณะที่เราเคลื่อนที่ เราจะเปลี่ยนตำแหน่งที่อยู่ตลอดแนว เช่น ขณะเราขับรถยนต์ไปตามท้องถนนจากโรงเรียนวังบ่อวิทยา อ.หนองบัว ไปยัง อ.เมืองนครสวรรค์ เราจะเคลื่อนที่ผ่านถนน ถนนอาจเป็นทางตรง ทางโค้ง หรือหักเป็นมุมฉาก ระยะทางที่รถเคลื่อนที่อาจเป็นระยะทางตามตัวเลขที่ราบของการเคลื่อนที่ แต่หากบางครั้งเราจะพบว่า จุดปลายทางที่เราเดินทางห่างจากจุดต้นทางในแนวเส้นตรง ไม่มากนัก

2.7.2.2 ระยะทาง (distance)

ระยะทาง คือ ความยาวตามเส้นทางที่วัตถุเคลื่อนที่ไปได้ทั้งหมด เป็นปริมาณสเกลาร์ คือ มีแต่ขนาดอย่างเดียว มีหน่วยเป็นเมตร (m) โดยทั่วไปเราใช้สัญลักษณ์ S

2.7.2.3 การกระจัด (displacement)

การกระจัด คือ เส้นตรงที่เชื่อมโยงระหว่างจุดเริ่มต้น และจุดสุดท้ายของการเคลื่อนที่เป็นปริมาณเวกเตอร์ คือ ต้องคำนึงถึงทิศทางด้วย มีหน่วยเป็นเมตร(m) โดยทั่วไปเขียนแบบเว็กเตอร์เป็น **S**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการทำงาน

3.1 แนวคิดการออกแบบโปรแกรม

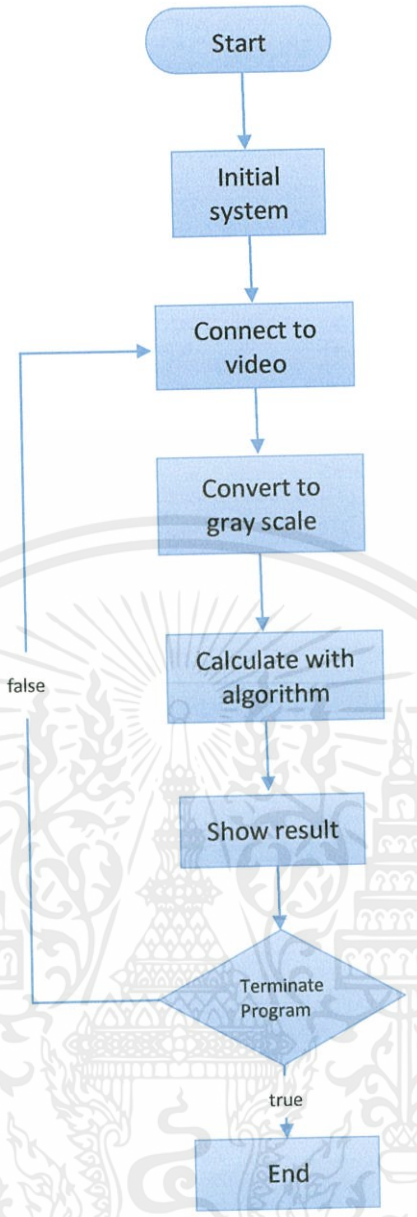
การทำงานของโปรแกรมคำนึงถึงเงื่อนไขของโปรแกรมที่จะต้องอยู่ภายในเงื่อนไขคือ

1. ตัวกล้องที่จับภาพนั้นจะต้องอยู่นิ่ง ไม่มีการเคลื่อนไหวซึ่งจะทำให้ภาพพื้นหลังไม่มีการเปลี่ยนตำแหน่ง
2. วัตถุที่เคลื่อนไหวอยู่ภายในกล้องนั้นจะต้องมีเพียงวัตถุเดียว

3.2 ลักษณะการทำงานของโปรแกรม

เมื่อโปรแกรมเริ่มทำงานจะมีการกำหนดค่าเริ่มต้นให้กับตัวแปรที่ได้ประกาศไว้ หลังจากที่ผ่านกระบวนการดังกล่าวแล้ว โปรแกรมจะทำการดึงวิดีโอมาประมวลผลโดยภายในวิดีโอจะมีวัตถุเคลื่อนไหวเข้ามา จากนั้นทำการ Capture ภาพแล้วนำไปเข้าสู่กระบวนการทาง Image Processing โดยอาจจะการทำให้เป็นภาพขาวดำ เพื่อการประมวลผลที่เร็วแล้วนำภาพมาประมวลผลทีละเฟรมภาพ แล้วเข้าสู่กระบวนการแยกวัตถุที่เคลื่อนที่ออกจากภาพพื้นหลัง เฟรมแรกที่ได้จะเก็บไว้ใน Buffer และนำภาพถัดไปมาลบภาพแรก ซึ่งหากมีวัตถุเคลื่อนที่เข้ามายังเฟรมที่สอง มันจะมีค่าของผลต่าง จากนั้นก็ทำการคำนวณหาความเร็วของวัตถุที่ผ่านเข้ามาในเฟรม แล้วทำการประมวลผลคำนวณหาผลลัพธ์ตามอัลกอริทึมหลังจากนั้นโปรแกรมจะตรวจสอบเงื่อนไขว่าต้องการจะหยุดการประมวลผลโปรแกรมต่อหรือไม่ ถ้าต้องการโปรแกรมก็จะหยุดการทำงาน แต่ถ้าไม่ต้องการโปรแกรมก็จะทำงานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 Flow chart แสดงขั้นตอนการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การทดลอง

3.3.1 ส่วนการทดลองเขียนแอปพลิเคชัน Android

- **ทดลองเขียนแอปพลิเคชันติดต่อกับกล้องทางโทรศัพท์มือถือ**

ส่วนของ Surface View ที่ใช้ติดต่อกับจอภาพ โดยจะมี 3 สถานะ

```
public void surfaceChanged(SurfaceHolder arg0, int arg1, int arg2, int arg3) {
}

public void surfaceCreated(SurfaceHolder arg0) { }

public void surfaceDestroyed(SurfaceHolder arg0) { }
```

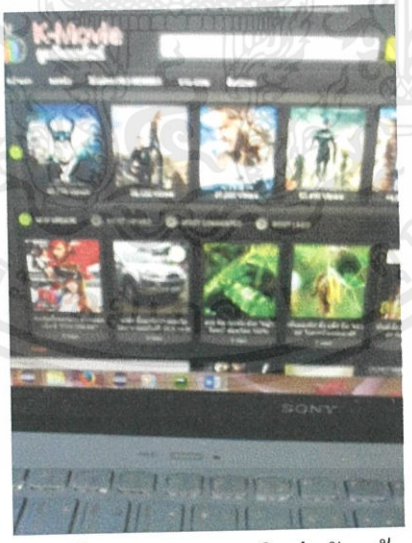
ใช้งาน onResume เพื่อเปิดหน้ากล้อง และ onPause เพื่อปิดหน้ากล้อง

```
public void onResume() {
    Log.d("System","onResume");
    super.onResume();
    mCamera = Camera.open();
}

public void onPause() {
    Log.d("System","onPause");
    super.onPause();
    mCamera.release();
}
```

ในส่วนของ AndroidManifest.xml เป็นการขอ Permission เพื่อใช้งานกล้อง

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera.autofocus" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```



รูปที่ 3.2 ผลลัพธ์เมื่อเปิดใช้งาน application ติดต่อกับกล้องจากโทรศัพท์มือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **ทดลองเขียนแอปพลิเคชันถ่ายภาพจากโทรศัพท์มือถือ**

ประกาศตัวแปรคลาส Camera

```
Camera mCamera;
```

ตัวแปร mCamera เรียกใช้คำสั่งถ่ายภาพ โดยมีการรับ parameter 4 ตัว ส่วนนี้จะเป็นส่วนของ ShutterCallback โดย shutter คือ call back จะทำงานเมื่อ shutter ทำงาน raw คือส่วนของข้อมูลดิบเมื่อทำการถ่ายภาพ postview คือภาพที่แสดงชั่วคราวเมื่อถ่าย และ jpeg คือชนิดของไฟล์ภาพที่ต้องการเก็บ

```
mCamera.takePicture(shutter, raw, postview, jpeg);
```

แสดงค่า parameter ต่างๆ

```
public void onClick(View arg0) {
    if(!saveState) {
        saveState = true;
        mCamera.takePicture(Main.this, null, null, Main.this);
    }
}
```

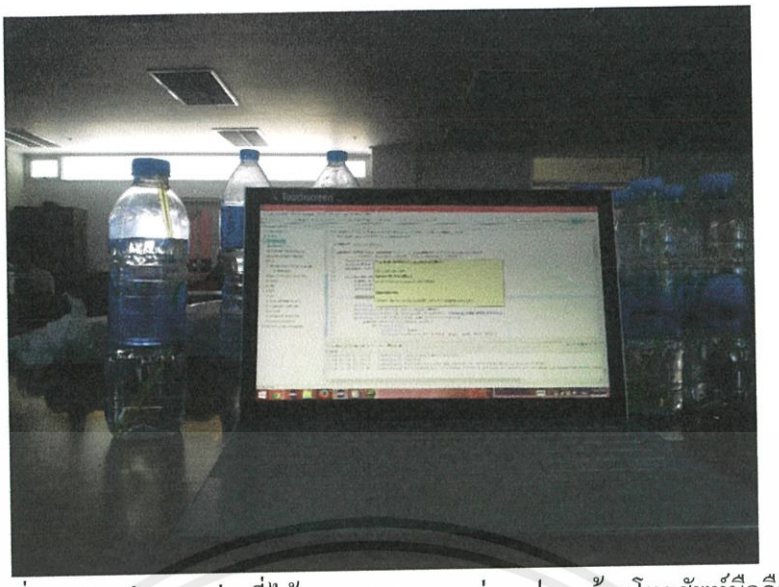
คำสั่งถ่ายภาพก็มีการเรียก Callback ของ shutter กับ jpegโดยให้ Callback ไปที่ Main.this จึงต้องประกาศ Implement ด้วย

```
public class Main extends Activity implements SurfaceHolder.Callback
        , Camera.ShutterCallback, Camera.PictureCallback {
```

กำหนด OnClickListener ให้กับ Surface View เพื่อให้สามารถถ่ายภาพได้เมื่อกดส่วนใดๆ บนหน้าจอ และกำหนด callback และ jpeg ไปที่ Main.this โดยหาก saveState มีค่า true จะเป็นการถ่ายภาพ

```
mPreview.setOnClickListener(new OnClickListener() {
    public void onClick(View arg0) {
        if(!saveState) {
            saveState = true;
            mCamera.takePicture(Main.this, null, null, Main.this);
        }
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 ตัวอย่างภาพถ่ายที่ได้จาก application ถ่ายรูปจากด้วยโทรศัพท์มือถือ

- **ทดลองทำ Face Detection โดยทำการศึกษาจากไลบรารี่ OpenCV**

รูปที่ 3.4 ตัวอย่างการใช้งานโปรแกรมตรวจจับใบหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนนี้จะ implement LoaderCallbackInterface เมื่อเริ่มโปรแกรม

```
private BaseLoaderCallback  mLoaderCallback = new BaseLoaderCallback(this) {
```

ส่วนที่ทำการโหลด Library หลักที่จะใช้งาน คือ detection_based_tracker

```
System.loadLibrary("detection_based_tracker");
```

โหลด cascade ไฟล์จากแอพพลิเคชั่น

```
InputStream is = getResources().openRawResource(R.raw.lbpcascade_frontalface);
File cascadeDir = getDir("cascade", Context.MODE_PRIVATE);
mCascadeFile = new File(cascadeDir, "lbpcascade_frontalface.xml");
FileOutputStream os = new FileOutputStream(mCascadeFile);
```

ส่วนต่างๆที่จะให้แสดงบนหน้าจอเมื่อใช้งานโปรแกรม โดยส่วนนี้จะเป็น Library ของ NDK

```
if (mDetectorType == JAVA_DETECTOR) {
    if (mJavaDetector != null)
        mJavaDetector.detectMultiScale(mGray, faces, 1.1, 2, 2, // TODO: objdetect.CV_HAAR_SCALE_IMAGE
                new Size(mAbsoluteFaceSize, mAbsoluteFaceSize), new Size());
}
else if (mDetectorType == NATIVE_DETECTOR) {
    if (mNativeDetector != null)
        mNativeDetector.detect(mGray, faces);
}
else {
    Log.e(TAG, "Detection method is not selected!");
}
```

- **ทดลองเขียนแอปพลิเคชันสำหรับตรงจับวัตถุเคลื่อนไหว**

ส่วน Permission สำหรับเปิดใช้งาน CAMERA

```
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.HARDWARE_CAMERA"/>
<uses-permission android:name="android.permission.HARDWARE_CAMERA_AUTOFOCUS"/>
```

Class สำหรับ camera preview

```
public class SurfacePreview extends SurfaceView implements SurfaceHolder.Callback {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนนี้จะถูกเรียกใช้ทันทีเมื่อ create Surface

```
//create Surface
@Override
public void surfaceCreated(SurfaceHolder holder) {
    // TODO Auto-generated method stub
    //to tell camera where to put a preview
    if (mCamera == null){
        mCamera = Camera.open();
    }
    //Exception
    try{
        mCamera.setPreviewDisplay(holder);
        mCamera.startPreview();
    } catch (IOException e){
        mCamera.release();
        mCamera = null;
        Log.d(TAG, "Error setting camera preview: " + e.getMessage());
    }

}
```

ส่วนนี้เป็นการตั้งค่าหากหน้าจอมีการเปลี่ยน Surface

```
@Override
public void surfaceChanged(SurfaceHolder holder, int format, int w, int h) {
```

หยุดการทำงานของ SurfacePreview

```
@Override
public void surfaceDestroyed(SurfaceHolder holder) {
    // TODO Auto-generated method stub
    if(mCamera != null){
        mCamera.stopPreview();
        mCamera.setPreviewCallback(null);
        mCamera.release();
        mCamera = null;
    }
```

Class หลัก

```
public class MainActivity extends Activity {

    private Boolean capturePictures = false;
    private Camera mCamera = null;
    private Button button;
    private SurfacePreview view;
    public byte[] data0 = null;
    public byte[] data1 = null;
    private Context context;
```

ฟังก์ชัน Callback ของ Camera

```
private PictureCallback mPictureCallback = new PictureCallback(){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อเรียกใช้งาน Camera และ SurfaceView

```
private void captureNextPicture(){
```

Method พื้นฐาน

```
protected void onCreate(Bundle savedInstanceState) {
            protected void onPause(){
             protected void onResume(){
```

ส่วนนี้แปลง bitmap ที่เป็น byte ไปเป็น Mat ซึ่งเป็นชนิดที่ใช้ได้กับ OpenCV

```
private Mat convertToMAT(byte[] data){
    Bitmap bitmap = BitmapFactory.decodeByteArray(data, 0, data.length);
    Mat tmp = new Mat(bitmap.getWidth(), bitmap.getHeight(), CvType.CV_16UC4);
    Utils.bitmapToMat(bitmap, tmp);
    return tmp;
}
```

ส่วนนี้จะเป็นส่วนเปรียบเทียบเฟรมภาพ โดยมี threshold เป็น 35

```
Core.absdiff(mat2, mat1, d1);
Core.absdiff(mat3, mat1, d2);
Core.bitwise_and(d1, d2, result);
Imgproc.threshold(result, result, 35, 255, Imgproc.THRESH_BINARY);
double value = Core.norm(result);
return value;
```

ผลการทดลองปรากฏว่า เมื่อรันโปรแกรมลงบนโทรศัพท์มือถือระบบปฏิบัติการ Android เกิดข้อผิดพลาดขึ้นดังนี้

1. ระบบปฏิบัติการของโทรศัพท์ที่ใช้อยู่อาจเก่าเกินไป
2. ปัญหา Exception ของโค้ดในโปรแกรม
3. ปัญหาในการ return ค่าต่างๆ ของโปรแกรม
4. ปัญหาในการเรียกใช้งาน method ต่างๆ ในโปรแกรมอาจไม่ตรงกับระบบปฏิบัติการที่ใช้

ด้วยการเกิดข้อผิดพลาดดังกล่าวทางผู้จัดทำเลยทำการทดลองบนระบบปฏิบัติการ Windows และทำการพัฒนาโปรแกรมด้วยภาษา C++ บน Visual Studio 2010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 ส่วนการทดลองพัฒนาโปรแกรมด้วย Visual Studio 2010

- **การทดลองเขียนโปรแกรมสำหรับการทำ Background Subtraction**

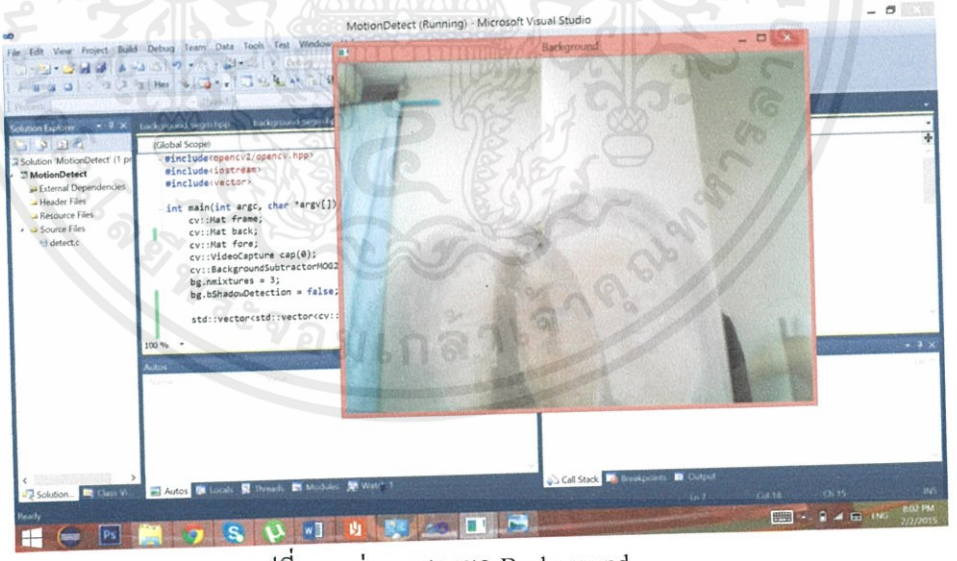
การทำ Background subtraction จะใช้วิธี Mixture of Gaussian ที่หาทั้งค่า background และ foreground เพื่อหาค่าความเปลี่ยนแปลง ส่วนการ Detect วัตถุและวาดขอบของวัตถุในภาพจะใช้วิธีการของ Vectors และมีการเรียกใช้คำสั่งใน library ของ OpenCV โดยทำการ add path ต่างๆ ในโปรเจค และทำการ include library ที่หัวของโปรเจค

ฟังก์ชั่นของ Mixture of Gaussian เพื่อทำ Background subtraction

```
cv::BackgroundSubtractorMOG2 bg;
bg.nmixtures = 3;
bg.bShadowDetection = false;
```

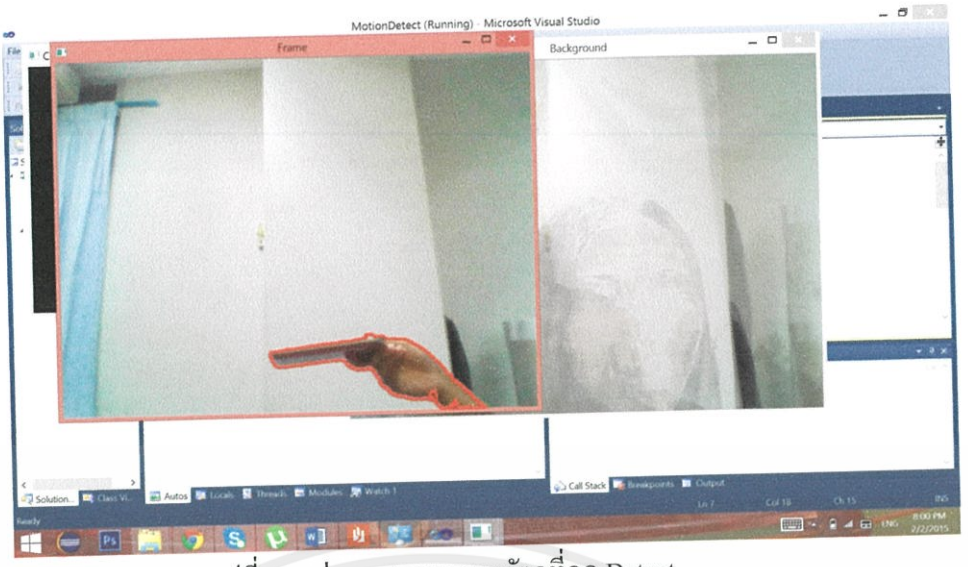
ฟังก์ชันใน Detect และวาดขอบของวัตถุ

```
cv::findContours(fore,
                 contours,
                 CV_RETR_EXTERNAL,
                 CV_CHAIN_APPROX_NONE);
cv::drawContours(frame,
                 contours,
                 -1,
                 cv::Scalar(0,0,255),
                 2);
```



รูปที่ 3.5 ส่วนแสดงผล Background

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 ส่วนแสดงผลของวัตถุที่ถูก Detect

- **การทดลองเขียนโปรแกรมสำหรับการทำ Threshold ภาพ และทำ Background Subtraction**

การทำ Background Subtraction ก่อน โดยไม่มีการเรียกใช้ฟังก์ชันตรวจจับเงา ซึ่งทำให้แน่ใจว่าสิ่งที่เคลื่อนที่เข้ามาเป็นวัตถุแน่ๆ ไม่ใช่เพียงแค่เงาของวัตถุ โดยจากเดิมจะเป็นการตรวจจับทันทีเมื่อวัตถุวิ่งเข้ามาผ่านกล้อง Web Camera แต่ได้ทำการแก้ไขโดยการเรียกใช้งานไฟล์วีดีโอ เพื่อที่ต่อไปจะสามารถนำภาพที่ถ่ายจากกล้องจากโทรศัพท์ระบบปฏิบัติการ Android ได้

ใช้จุดกึ่งกลางของวัตถุเพื่อหาตำแหน่ง

```
Rect objectBoundingRectangle = Rect(0,0,0,0);
```

หากรอบให้วัตถุโดยการทำ contour

```
findContours(temp,contours,hierarchy,CV_RETR_EXTERNAL,CV_CHAIN_APPROX_SIMPLE );// external contours
```

กำหนดให้ contour ที่ใหญ่ที่สุดเป็นวัตถุ

```
vector< vector<Point> > largestContourVec;
largestContourVec.push_back(contours.at(contours.size()-1));
```

วาดกรอบสี่เหลี่ยมให้กับวัตถุที่ทำการ contour ได้

```
objectBoundingRectangle = boundingRect(largestContourVec.at(0));
int xpos = objectBoundingRectangle.x+objectBoundingRectangle.width/2;
int ypos = objectBoundingRectangle.y+objectBoundingRectangle.height/2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนค่าใน array หากวัตถุมีการเปลี่ยนตำแหน่ง

```
theObject[0] = xpos , theObject[1] = ypos;
```

การทำ subtraction

1. อ่านเฟรมภาพที่ 1
2. แปลงเป็น gray scale
3. อ่านเฟรมภาพที่ 2
4. แปลงเป็น gray scale

```
capture.read(frame1);
cv::cvtColor(frame1,grayImage1,COLOR_BGR2GRAY);
capture.read(frame2);
cv::cvtColor(frame2,grayImage2,COLOR_BGR2GRAY);
```

การแสดงภาพ difference และ threshold

```
cv::imshow("Difference Image",differenceImage);
cv::imshow("Threshold Image", thresholdImage);
```

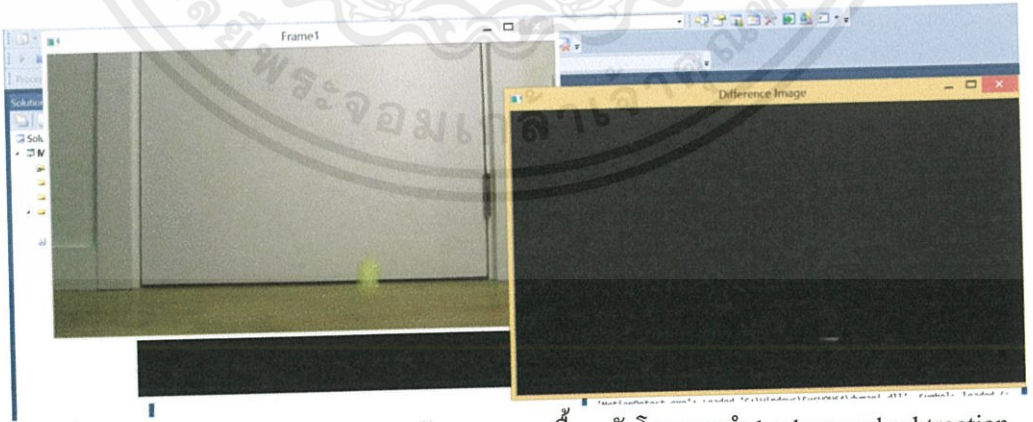
ทำการ blur ภาพเพื่อกำจัด noise แล้วแสดง threshold อีกครั้งเพื่อให้ได้วัตถุจริงๆ

```
cv::blur(thresholdImage,thresholdImage,cv::Size(BLUR_SIZE,BLUR_SIZE));

//threshold again
cv::threshold(thresholdImage,thresholdImage,SENSITIVITY_VALUE,255,THRESH_BINARY);
```

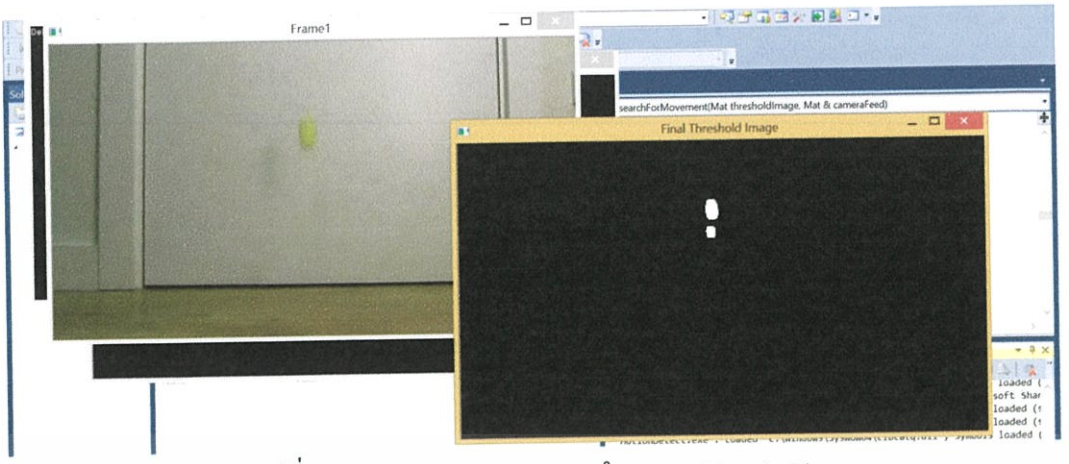
ทำการตรวจจับวัตถุโดยการเรียกใช้ภาพที่ได้ทำการ contour ไว้แล้ว

```
if(trackingEnabled){
    searchForMovement(thresholdImage,frame1);
}
```



รูปที่ 3.7 ผลการทดลองจากการแยกวัตถุออกจากพื้นหลังโดยการทำ background subtraction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 ผลการทดลองจากการทำ Image Threshold

- **การทดลองเขียนโปรแกรมเพื่อตรวจจับวัตถุที่วิ่งเข้ามาผ่านกล้อง Web Camera และทำการ Contour วัตถุ**

วาดกรอบสี่เหลี่ยมให้กับวัตถุที่ทำการ contour ได้

```
for(size_t i=0;i<contours.size();i++){
    for(size_t j=0;j<contours[i].size();j++){
        cv::Point p=contours[i][j];
        points.push_back(p);
    }
}

if(points.size()>0){
    cv::Rect brect=cv::boundingRect(cv::Mat(points).reshape(2));
    cv::rectangle(cameraFeed, brect.tl(), brect.br(), cv::Scalar(100,100,200),2,CV_AA);

}
```

ใน capture.open จะให้ค่าเป็นศูนย์เนื่องจากไม่ได้มีการเรียกเปิดไฟล์ video

```
VideoCapture capture;
capture.open(0);
VideoWriter oVideoWriter;
double dWidth = capture.get(CV_CAP_PROP_FRAME_WIDTH);
double dHeight = capture.get(CV_CAP_PROP_FRAME_HEIGHT);

Size frameSize(static_cast<int>(dWidth), static_cast<int>(dHeight));
```

ทำการตรวจจับวัตถุโดยการหาการเคลื่อนที่ของวัตถุใน video

```
if(trackingEnabled){
    motionDetected = detectMotion(thresholdImage,frame1);

}else{

    motionDetected = false;

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.9 ผลการทดลองการรันโปรแกรมโดยเมื่อไม่มีวัตถุผ่านเข้ามาในกล้อง

รูปที่ 3.10 ผลการทดลองเมื่อมีวัตถุผ่านเข้ามาในกล้อง จะเกิดการตรวจจับวัตถุที่เคลื่อนไหวได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **การทดลองเขียนโปรแกรมเพื่อแยกเฟรมวิดีโอออกเป็นรูปภาพ**

การแยกเฟรมวีดีโอออกเป็นรูปภาพ แล้วนับจำนวนเฟรม จากนั้นทำการนับค่าความเร็วของวีดีโอนั้นนั้นต่อวินาที หรือ fps แล้วทำการคำนวณค่าความยาวของวีดีโอจาก time=frames/fps ซึ่งค่านี้เป็นค่าความยาวของวีดีโอทั้งหมด

ตัวแปรสำหรับเก็บค่า โดย loop เป็นค่าจากจำนวนเฟรมโดยเริ่มจาก 0 frame เป็น array เพื่อเก็บค่าชื่อของเฟรมวีดีโอ

```
CvCapture* capture = cvCaptureFromFile("video1.avi");
IplImage* frame = NULL;
int loop=0;
char fname[20];
```

ใส่ค่าให้ชื่อเฟรม และจำนวน loop เพื่อบันทึกภาพ

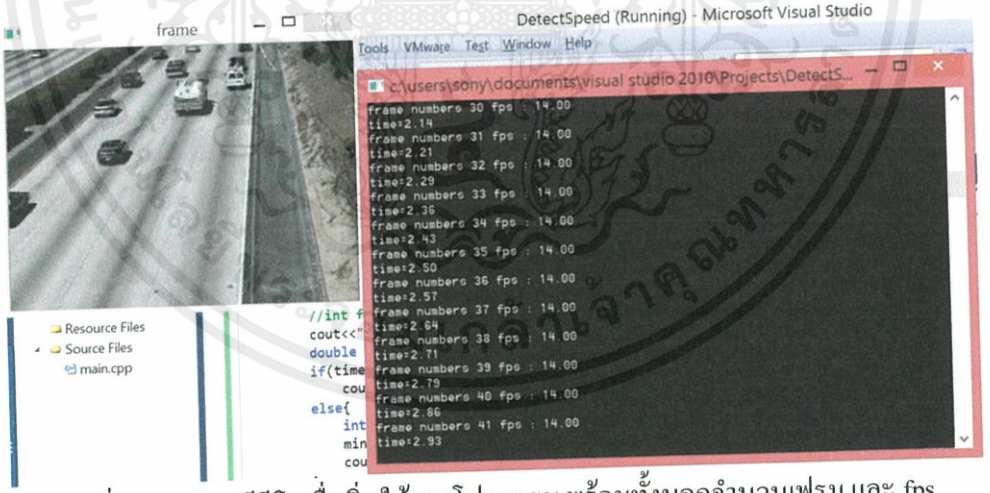
```
sprintf(fname,"fname%d.jpg",loop);
cvSaveImage(fname,frame);
```

นับค่า fps และคำนวณเวลาโดย time = loop/fps

```
double fps=(int) cvGetCaptureProperty(capture,CV_CAP_PROP_FPS);
cout<<"frame numbers "<<loop<<" fps : "<<fps<<endl;
double time = loop/fps;
```

loop เพิ่มขึ้นตามจำนวนเฟรม

```
loop++;
```



รูปที่ 3.11 แสดงวีดีโอเมื่อเริ่มใช้งานโปรแกรม พร้อมทั้งบอกจำนวนเฟรม และ fps

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.12 ภาพรูปภาพที่บันทึกจากเฟรมวิดีโอแต่ละเฟรม

- **การทดลองเขียนโปรแกรมเพื่อจับความเร็ววัตถุที่เคลื่อนที่ผ่านเข้ามาในวิดีโอ**

การเขียนโปรแกรมเพื่อนับจำนวนเฟรมที่มีวัตถุอยู่ในเฟรมเพื่อนำมาคำนวณหาความเร็วของวัตถุโดยการทดลอง ใช้กล้องถ่ายวีดีโอจากโทรศัพท์มือถือ Samsung Galaxy Note3 และมีอัตรา fps อยู่ที่ 30 frame/second

การทำงานคือคำนวนเฟรมที่วัตถุวิ่งผ่านเข้ามาในเลนส์กล้อง โดยที่นำจำนวนเฟรมทั้งหมดที่จับได้ตอนวัตถุผ่านกล้องมาคำนวณเทียบกับ fps เพื่อหาเวลา โดย

จำนวนเฟรม 30 เฟรม ใช้เวลา = 1 วินาที

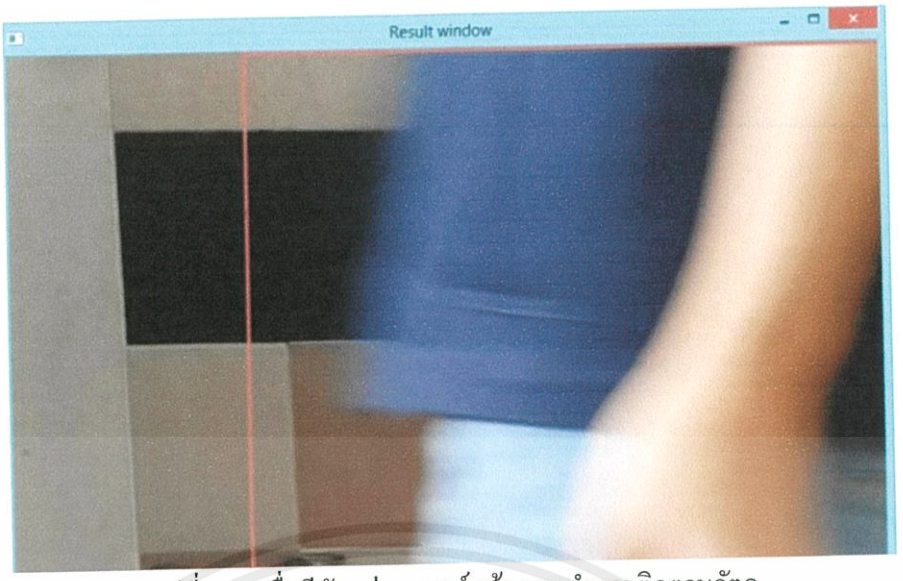
จำนวนเฟรมที่จับได้ ใช้เวลา = $\frac{\text{จำนวนเฟรมที่จับได้} \times 1 \text{ วินาที}}{30}$

การหาความเร็วทำได้โดย $v = \frac{s}{t}$

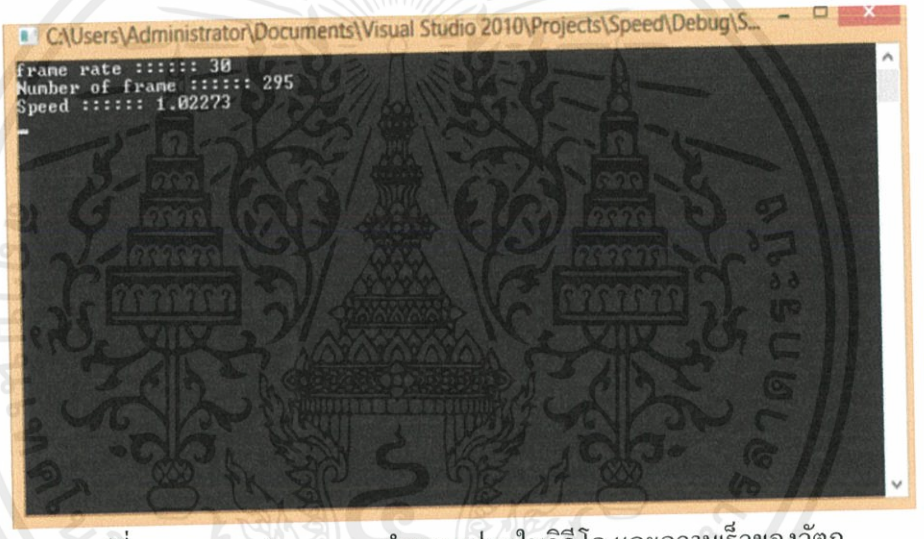
ฟังก์ชันในการหาจำนวน frame rate เพื่อหาเวลา

```
double rate = capture.get(CV_CAP_PROP_FPS);
int fr = capture.get(CV_CAP_PROP_FRAME_COUNT);
double time = fr/rate;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 เมื่อมีวัตถุผ่านเลนส์กล้อง จะทำการติดตามวัตถุ



รูปที่ 3.14 แสดง frame rate, จำนวนเฟรมในวิดีโอ และความเร็วของวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

4.1 สภาพแวดล้อมที่ใช้ในการทดลอง

การทดลองจะเป็นการทดลองบนถนนโดยมีการควบคุมตัวแปรต่างๆ ดังนี้

1. ถ่ายวีดีโอของวัตถุจากกล้องโทรศัพท์มือถือระบบปฏิบัติการ Android โดยในการทดลองนี้สามารถใช้กล้องที่มีความเร็วกี่ภาพต่อวินาทีก็ได้ เนื่องจากในโปรแกรมจะมีส่วนที่ใช้คำนวณค่าความเร็วของเฟรมต่อวินาที หรือ FPS (frame per second) อยู่แล้ว โดยในการทดลองนี้ได้ใช้กล้องวีดีโอที่มีค่า FPS เป็น 30
2. การถ่ายวีดีโอทำในสถานที่ที่มีแสงเพียงพอ
3. วัตถุต้องวิ่งผ่านเลนส์กล้องเพียงวัตถุเดียว สามารถวิ่งจากซ้ายไปขวา หรือขวาไปซ้ายก็ได้ และวัตถุต้องวิ่งเป็นเส้นตรงตั้งแต่จุดเริ่มต้นจนถึงจุดสิ้นสุดเฟรมวีดีโอ
4. วัดระยะทางระหว่างจุดเริ่มต้นและจุดสิ้นสุดเฟรมเทียบกับระยะทางจริง และระยะทางระหว่างเลนส์กล้องกับระยะที่วัตถุวิ่ง โดยวัดเป็นเส้นตั้งฉากกับเส้นที่วัตถุวิ่ง
5. เลนส์กล้องควรอยู่ในระดับความสูงเดียวกับวัตถุ โดยมีการกำหนดระยะทางดังนี้

ตารางที่ 4.1 แสดงระยะทางที่กำหนดในการถ่ายวีดีโอ

ระยะทางการวิ่งของวัตถุ	ระยะทางระหว่างวัตถุ-เลนส์กล้อง
7.7 เมตร	7 เมตร
20 เมตร	10 เมตร
30 เมตร	15 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองและผลการทดลอง

การทดลองที่ 1 ระยะทาง 7.7 เมตร พาหนะที่ใช้ในการทดลองคือรถจักรยานยนต์

รูปที่ 4.1 ทดลองตรวจจับความเร็วของรถจักรยานยนตร์

- ความเร็วจริง 10 กิโลเมตร/ชั่วโมง

ผลการทดลอง

```
frame rate :::::: 30
total frame ::::::171
Number of frame :::::: 60
Time VDO :::::: 5.7
Time Object :::::: 2
Speed :::::: 3.85 m/s
Speed :::::: 13.86 km/s
```

ความเร็วที่วัดได้ 13.86 กิโลเมตร/ชั่วโมง

- ความเร็วจริง 20 กิโลเมตร/ชั่วโมง

ผลการทดลอง

```
frame rate :::::: 30
total frame ::::::146
Number of frame :::::: 41
Time VDO :::::: 4.86667
Time Object :::::: 1.36667
Speed :::::: 5.63415 m/s
Speed :::::: 20.2829 km/s
```

ความเร็วที่วัดได้ 20.3 กิโลเมตร/ชั่วโมง

- ความเร็วจริง 25 กิโลเมตร/ชั่วโมง

ผลการทดลอง

```
frame rate :::::: 30
total frame ::::::191
Number of frame :::::: 33
Time VDO :::::: 6.36667
Time Object :::::: 1.1
Speed :::::: 7 m/s
Speed :::::: 25.2 km/s
```

ความเร็วที่วัดได้ 25.2 กิโลเมตร/ชั่วโมง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2 ระยะทาง 20 เมตร พาหนะที่ใช้ในการทดลองคือรถยนต์

รูปที่ 4.2 ทดลองตรวจจับความเร็วของรถยนต์

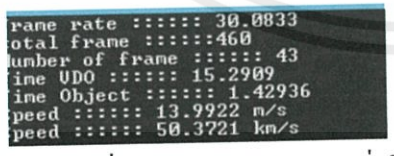
- ความเร็วจริง 40 กิโลเมตร/ชั่วโมง

ผลการทดลอง

ความเร็วที่วัดได้ 41.65 กิโลเมตร/ชั่วโมง

- ความเร็วจริง 50 กิโลเมตร/ชั่วโมง

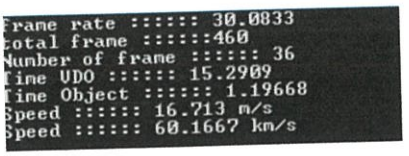
ผลการทดลอง



ความเร็วที่วัดได้ 50.37 กิโลเมตร/ชั่วโมง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความเร็วจริง 60 กิโลเมตร/ชั่วโมง



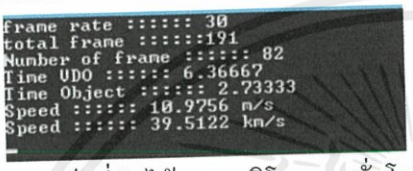
ผลการทดลอง

ความเร็วที่วัดได้ 60.17 กิโลเมตร/ชั่วโมง

การทดลองที่ 3 ระยะทาง 30 เมตร พาหนะที่ใช้ในการทดลองคือรถยนต์

- ความเร็วจริง 40 กิโลเมตร/ชั่วโมง

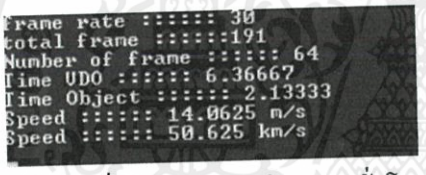
ผลการทดลอง



ความเร็วที่วัดได้ 39.51 กิโลเมตร/ชั่วโมง

- ความเร็วจริง 50 กิโลเมตร/ชั่วโมง

ผลการทดลอง



ความเร็วที่วัดได้ 50.63 กิโลเมตร/ชั่วโมง

- ความเร็วจริง 60 กิโลเมตร/ชั่วโมง

ความเร็วที่วัดได้ 61.13 กิโลเมตร/ชั่วโมง

4.3 สรุปผลการทดลอง

จากการทดลองสรุปได้ว่า โปรแกรมสามารถทำงานได้อย่างมีประสิทธิภาพและให้ผลการทดลองที่ถูกต้อง หรือใกล้เคียงกับความเป็นจริงเมื่อควบคุมสภาพแวดล้อมของระบบตามที่ได้กำหนดไว้ในหัวข้อ 4.1 โดยจากการทดลองทั้งหมด 9 ครั้ง ที่ระยะทาง 7.7 เมตร จำนวน 3 ครั้ง พบว่าความเร็วที่วัดได้มีความคาดเคลื่อนเฉลี่ย 13.63% ที่ระยะทาง 20 เมตร จำนวน 3 ครั้ง พบว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเร็วที่วัดได้มีความคาดเคลื่อนเฉลี่ย 1.72% ที่ระยะทาง 30 เมตร จำนวน 3 ครั้ง พบว่าความเร็วที่วัดได้มีความคาดเคลื่อนเฉลี่ย ทั้งนี้ใน โดยสรุปความคาดเคลื่อนเฉลี่ยอยู่ที่ 1.46% ซึ่งสาเหตุอาจมาจากการที่วัตถุไม่สามารถวิ่งเป็นเส้นตรงตลอดเวลาได้ ทำให้การคำนวณคาดเคลื่อน ซึ่งถือเป็นข้อจำกัดของโปรแกรม

ตารางที่ 4.2 แสดงผลการทดลอง

ระยะทาง	ความเร็วจริง	ความเร็วที่วัดได้	เปอร์เซ็นต์ความคาดเคลื่อน
7.7 เมตร	10.00 km/hr	13.86 km/hr	38.6%
7.7 เมตร	20.00 km/hr	20.30 km/hr	1.5%
7.7 เมตร	25.00 km/hr	25.20 km/hr	0.8%
20 เมตร	40.00 km/hr	41.65 km/hr	4.13%
20 เมตร	50.00 km/hr	50.37 km/hr	0.74%
20 เมตร	60.00 km/hr	60.17 km/hr	0.28%
30 เมตร	40.00 km/hr	39.51 km/hr	1.23%
30 เมตร	50.00 km/hr	50.63 km/hr	1.26%
30 เมตร	60.00 km/hr	61.13 km/hr	1.88%
ความคาดเคลื่อนเฉลี่ย			5.6%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป

5.1 สรุปผลการดำเนินงาน

โครงงานนี้เป็นโครงงานที่ต้องกำหนดปัจจัยในการควบคุมหลายอย่าง เนื่องจากข้อจำกัดของการใช้งานเทคนิคประมวลผลภาพ โดยจะเห็นได้ว่าโปรแกรมสามารถทำงานได้ผลที่แม่นยำในระยะทางที่มีการกำหนดไว้แล้ว และต้องกำหนดระยะทางระหว่างเลนส์กล้องกลับเส้นทางการเคลื่อนที่ของวัตถุ อีกทั้งต้องมีวัตถุวิ่งไม่เกิน 1 วัตถุ และวัตถุต้องวิ่งเป็นเส้นตรง ซึ่งจากการทำงานของโปรแกรมจะมีความคลาดเคลื่อนอยู่บ้าง ขึ้นอยู่กับการเคลื่อนที่ของวัตถุด้วย ซึ่งโดยปกติในความเป็นจริงนั้นวัตถุไม่สามารถวิ่งเป็นเส้นตรงได้โดยสมบูรณ์อยู่แล้ว แต่ก็ถือว่าสามารถวัดความเร็วเฉลี่ยของวัตถุที่เคลื่อนที่ได้ใกล้เคียงกับค่าจริง ทำให้สามารถรู้ค่าประมาณความเร็วของวัตถุที่เคลื่อนที่ได้ โดยข้อผิดพลาดของโปรแกรมมักจะเกิดขึ้นเนื่องจากเหตุการณ์ต่อไปนี้

1). เกิดเงาขึ้นในขณะถ่ายวีดีโอ แม้ว่าจะไม่มีวัตถุอีกชิ้นผ่านเข้ามา แต่หากเกิดเงาขึ้นโปรแกรมอาจจะคิดว่าเป็นวัตถุ เนื่องจากโปรแกรมใช้เทคนิคการเปรียบเทียบวีดีโอแบบเฟรมต่อเฟรม

2). วัตถุเคลื่อนที่ไม่เป็นเส้นตรง หรือหยุดระหว่างทาง เนื่องจากตัวโปรแกรมนั้นต้องมีการกำหนดระยะทางที่วัตถุเคลื่อนทีเป็นเส้นตรงจากจุดเริ่มต้นไปจนถึงจุดสิ้นสุดเฟรม หากวัตถุไม่เคลื่อนที่ไปจนจุดสิ้นสุดเฟรม โปรแกรมก็จะหยุดการติดตามวัตถุเนื่องจากไม่มีวัตถุเคลื่อนที่ และทำการคำนวณความเร็ว ซึ่งความเร็วที่ได้จะไม่ใช่ความเร็วจริง

3). กล้องมีการเคลื่อนไหว ในส่วนนี้จะมีความผิดพลาดเนื่องจากหากกล้องมีการเคลื่อนไหว โปรแกรมจะคิดว่าฉากหลังทั้งหมดเคลื่อนไหว ทำให้โปรแกรมติดตามวัตถุไม่แม่นยำ

จะเห็นว่าหากควบคุมสภาพแวดล้อมให้ดีก็จะสามารถวัดความเร็วได้อย่างแม่นยำ โดยสามารถนำไปประยุกต์ใช้ได้ทั้งในการตรวจจับความเร็วของรถยนต์ การตรวจจับความเร็วของวัตถุในโรงงานอุตสาหกรรม เป็นต้น ดังนั้นหากต้องการจะพัฒนาให้โปรแกรมมีประสิทธิภาพมากขึ้นจาเป็นต้องศึกษาข้อมูลและทาการทดลองเชิงลึกอีกมาก เพื่อค้นคว้าหาอัลกอริทึมที่สามารถช่วยให้โปรแกรมสามารถใช้งานได้ครอบคลุมในทุกสถานที่และทุกกรณี โครงงานนี้จึงเป็นจุดเริ่มต้นที่ดีในการศึกษาและพัฒนาต่อสาหรับบุคคลที่มีความสนใจในหัวข้อดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ข้อเสนอแนะ

โครงงานนี้เป็นการศึกษาเทคนิคการประมวลผลภาพเพื่อนำมาประยุกต์ใช้ในการคำนวณความเร็วของวัตถุ ซึ่งเป็นการทดลองจากวีดีโอของกล้องโทรศัพท์มือถือ โดยวัดระยะทางการเคลื่อนที่ของวัตถุเทียบกับความยาวของเฟรมที่กล้องสามารถจับได้ หากอยากได้ระยะทางที่ไกลขึ้นต้องให้เลนส์กล้องและระยะวัตถุมีระยะทางไกลขึ้น โดยมีปัจจัยที่เป็นอุปสรรคในการประมวลผลภาพ เช่น สัญญาณรบกวน (noise) สภาพแวดล้อมขณะทำการทดลอง เนื่องจากเทคนิคการประมวลผลภาพนั้นต้องใช้ทรัพยากรเครื่องคอมพิวเตอร์อย่างมาก ใช้หน่วยประมวลผลที่มีประสิทธิภาพ ดังนั้นประสิทธิภาพในการประมวลผลขึ้นอยู่กับทรัพยากรทางคอมพิวเตอร์ที่ใช้ในการประมวลผลด้วย ทั้งนี้หากต้องการประมวลผลแบบ Real Time สามารถใช้กล้องจากคอมพิวเตอร์ได้ แต่อาจจะให้ความคมชัดไม่เท่ากล้องจากโทรศัพท์มือถือ

5.3 แนวทางการพัฒนาต่อ

สำหรับแนวทางการพัฒนาโปรแกรมนี้ สามารถนำไปประยุกต์ใช้ได้หลากหลาย เช่น นำไปใช้ในกล้องตรวจจับความเร็วรถยนต์บนถนน นำไปใช้ในโรงงานอุตสาหกรรม เป็นต้น ทั้งนี้หากต้องการนำไปประยุกต์ใช้ในการประมวลผลแบบ real time ก็สามารถทำได้ เพียงเปลี่ยนค่า input จากวีดีโอเป็นกล้องเว็บแคมจากคอมพิวเตอร์ หากจะนำไปประยุกต์ใช้ในการตรวจจับความเร็วของวัตถุที่มีความเร็วสูง ประสิทธิภาพของกล้องและหน่วยประมวลผลควรสูงด้วย เนื่องจากเทคนิคการประมวลผลภาพเป็นการประมวลผลที่มีความละเอียดอ่อน และไวต่อสภาพแวดล้อม หากนำไปประยุกต์ใช้ควรมีเทคนิคที่สามารถตรวจจับภาพในขณะที่มีแสงสว่างไม่มาก เช่น ในตอนกลางคืนบนถนน เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ผศ.ดร.อรฉัตร จิตต์โสภักตร์. 2552. Digital Image Processing. พิมพ์ครั้งที่ 1. กรุงเทพฯ : บริษัท สงวนกิจ พริ้นท์ แอนด์ มีเดีย.

รศ.ดร. สมเกียรติ อุดมหรรษากุล. 2554. การประมวลผลภาพดิจิตอลเบื้องต้น. – กรุงเทพฯ : บริษัท สำนักพิมพ์ท้อป จำกัด.

รศ.ดร. กิตติ ไพฑูรย์วัฒนกิจ. 2549. การประมวลผลภาพดิจิตอล. พิมพ์ครั้งที่ 1. กรุงเทพฯ : ภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

Rafael C. Gonzalez and Richard E. Woods. 2002. Digital Image Processing. Prentice Hall. สืบค้นจาก : http://users.dcc.uchile.cl/~jsaavedr/libros/dip_gw.pdf

ฐานิศร์ เฉลิมบุญ. 2551. "Motion Detection by Optical Flow." วิทยานิพนธ์วิศวกรรมศาสตบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย , มหาวิทยาลัยสงขลานครินทร์ .

นางสาวภัทรา สุมโนจิตราภรณ์. 2551. "Component Classification of Occluded Objects using Path Analysis", วิทยานิพนธ์วิศวกรรมศาสตบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย, มหาวิทยาลัยเกษตรศาสตร์.

Wikipedia. 2015. **ความเร็ว.** [Online]. Available : http://th.wikipedia.org/wiki/ความเร็ว.

Digital Image Processing, 1997. Chapter 14, **Motion Analysis: Optical flow**.[Online]. Available : http://user.engineering.uiowa.edu/~dip/lecture/Motion2.html

LABPLAYS, 2013. Emgu Cv Lab8: ตรวจจับวัตถุที่กำลังเคลื่อนไหว Background subtraction.[Online] Available : http://kengzer.blogspot.com/2013/01/lab8.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้