สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การวิเคราะห์การเคลื่อนไหวของร่างกาย

BODY MOVEMENT ANALYSIS

T103022

ธนพล ภูวรัตนกุล
พงษ์ หังสวนัส
สุภาพ หวังพิทักษ์วงศ์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน 103022
วัน,เดือน,ปี 24 ส.ค. 2552

b 12043111
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2551

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การวิเคราะห์การเคลื่อนไหวของร่างกาย

BODY MOVEMENT ANALYSIS

ผู้จัดทำ

1. นายธนพล ภูวรัตนกุล รหัสนักศึกษา 48010345
2. นายพจน์ หังสวนัส รหัสนักศึกษา 48010568
3. นายสุภาพ หวังพิทักษ์วงศ์ รหัสนักศึกษา 48011007

อาจารย์ที่ปรึกษา

(รศ.ดร. บุญวัฒน์ อัตชู)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวิเคราะห์การเคลื่อนไหวของร่างกาย

นายธนพล ภูวรัตนกุล 48010345
นายพจน์ หังสวนัส 48010568
นายสุภาพ หวังพิทักษ์วงศ์ 48011007
รศ.ดร. บุญวัฒน์ อัตชู อาจารย์ที่ปรึกษา
ปีการศึกษา 2551

บทคัดย่อ

ปัจจุบันมีการประยุกต์ใช้คอมพิวเตอร์ ในการประมวลผลภาพเป็นอย่างมาก ไม่ว่าจะเป็นในโรงงานอุตสาหกรรม ตามหน่วยงานต่างๆ เช่น การค้นหาวัตถุในภาพ การตรวจจับวัตถุที่สนใจในภาพ การรู้จำใบหน้า นอกจากนี้ยังมีการนำคอมพิวเตอร์มาประมวลผลภาพการเคลื่อนไหวมากขึ้น เช่นการตรวจจับหาวัตถุที่เคลื่อนไหวในภาพ วิดีทัศน์ การตรวจจับหาผู้บุกรุก

โครงงานนี้ จะนำเทคโนโลยีในการประมวลผลภาพมาประยุกต์ใช้ในการศึกษาถึงการจับการเคลื่อนไหวที่เกิดขึ้นในภาพวิดีทัศน์หรือภาพจากกล้องวงจรปิดเพื่อเพิ่มความสามารถในการบันทึกเหตุการณ์ต่างๆที่เกิดขึ้น โดยกล้องวงจรปิดในปัจจุบัน ไม่สามารถบอกได้ว่าเฟรมที่จับได้นั้นมีการเคลื่อนไหวหรือไม่ทำให้จำเป็นต้องบันทึกภาพต่างๆที่เกิดขึ้นอยู่ตลอดเวลาซึ่งเป็นการสิ้นเปลืองพื้นที่บนหน่วยความจำไปโดยเปล่าประโยชน์ ดังนั้นถ้ากล้องวงจรปิดสามารถระบุได้ว่าเวลาไหนมีการเคลื่อนไหวเกิดขึ้นและบันทึกเฟรมนั้นๆลงไฟล์ก็จะเป็นการประหยัดพื้นที่บนหน่วยความจำและไม่เสียเวลาในเรียกดูไฟล์ในภายหลัง และยังเพิ่มความสามารถในการนับวัตถุที่ผ่านเข้าออกเฟรมและจำแนกว่าวัตถุเหล่านั้นมีความน่าจะเป็นที่จะเป็นการเคลื่อนไหวของมนุษย์หรือไม่โดยใช้เทคนิค Edge Detection และ Blob Analysis

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Body Movement Analysis

Mr. Tanapon Puvaratanagul 48010345

Mr. Pod Hunsavanus 48010568

Mr. Supharp Wangpitakwong 48011007

Assoc.Prof.Dr. Boonwat Attachoo Adivisor

Academic Year 2008

ABSTRACT

Image processing has found many applications in industries and organization. They include identification of interested object in a picture, monitoring of the object in a digital picture, facial recognition. Additionally, processing of moving object in a video and instruction detection system has been receiving a lot of attention.

This project has utilized the image processing technology to study the motions recorded by a video camera or a video surveillance system. Currently, a video surveillance system is unable to conclude whether a captured frame contains any motions in it; therefore, it wastes the memory as a video must always record. If a closed surveillance system is able to tell whether there is a motion or not, it can be programmed to capture only motions; thus reducing the memory required. The system is also enhanced by Edge Detection and Blob Analysis techniques to count the net throughput of objects and to indicate whether the motions were human-related.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานและปริญญานิพนธ์นี้คงมิอาจเกิดขึ้นได้ หากปราศจากการสนับสนุนจากอาจารย์ที่ปรึกษาโครงงาน ท่านอาจารย์ รศ.ดร. บุญวัฒน์ อัตชู ที่ได้ให้ทั้งคำปรึกษา คำแนะนำ และความช่วยเหลือต่างๆ ในการทำโครงงานนี้ จึงต้องขอขอบพระคุณท่านอาจารย์เป็นอย่างมากที่มีความกรุณาช่วยเหลือพวกข้าพเจ้าจนทำให้สามารถดำเนินโครงงานจนสำเร็จเสร็จสิ้นไปด้วยดี

รวมทั้งต้องขอขอบพระคุณท่านคณาจารย์ทุกท่านในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ประสิทธิ์ประสาทวิชาความรู้ทางด้านต่างๆ แก่พวกข้าพเจ้า จนสามารถนำความรู้เหล่านั้นมาบูรณาการและประยุกต์ใช้ในโครงงานนี้ได้อย่างถูกต้องและสมบูรณ์

นอกจากนี้ยังมีเพื่อนๆ พี่ๆ น้องๆ ทุกคนในภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ได้ให้ความคิดเห็น คำปรึกษา รวมทั้งกำลังใจในหลายๆโอกาส จึงต้องขอขอบคุณจากใจมา ณ ที่นี้

สุดท้ายนี้ต้องขอกราบขอบพระคุณคุณพ่อคุณแม่ และครอบครัวของพวกข้าพเจ้าทุกคน ซึ่งคอยชุบเลี้ยงพวกข้าพเจ้าจนเติบโตขึ้นมา ดูแลให้ได้รับการศึกษา รวมถึงคอยให้แรงใจ คอยช่วยเหลือสนับสนุน จนทำให้พวกเราได้เดินทางมาจนถึงจุดหมายที่พวกเราวาดหวังเอาไว้

ขอให้คุณประโยชน์อันใดที่จะสามารถพึงเกิดขึ้นได้ของปริญญานิพนธ์ฉบับนี้ ได้ส่งผลกลับไปถึงผู้มีพระคุณทุกๆท่าน

ธนพล ภูวรัตนกุล

พจน์ หังสวนัส

สุภาพ หวังพิทักษ์วงศ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญรูป VII

บทที่ 1 บทนำ 1

1.1 หลักการและเหตุผล 1

1.2 วัตถุประสงค์ 1

1.3 ขอบเขตของโครงงาน 2

1.4 ประโยชน์ที่คาดว่าจะได้รับ 2

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 3

2.1 การประมวลผลภาพ 3

2.1.1 การรับภาพและแปลงภาพ 3

2.1.2 การปรับปรุงคุณภาพของภาพ 3

2.1.3 การบีบอัดภาพ 4

2.1.4 การวิเคราะห์ข้อมูลภาพ 4

2.2 การประมวลผลภาพสี 5

2.2.1 มาตรฐานของสี 5

2.2.1.1 ระบบสี RGB 5

2.2.1.2 ระบบสี CMYK 6

2.2.1.3 ระบบสี HSV 6

2.2.1.4 ระบบสี YIQ 7

2.2.2 การแปลงค่าสีระหว่างระบบสีต่างๆ 7

2.2.2.1 การแปลงภาพสี RGB เป็นภาพระดับสีเทา 7

2.2.2.2 การแปลงภาพสี RGB เป็น HSV 8

2.2.2.3 การแปลงภาพสี RGB เป็น CMY 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.3 การสร้างภาพไบนารี 8
2.3.1 การหาค่าเทรชโฮลโดยการกำหนดค่าล่วงหน้า 9
2.3.2 การหาค่าเทรชโฮลจากค่ากลาง 10
2.3.3 การหาค่าเทรชโฮลโดยอัตโนมัติ 10
2.4 การกำจัดสิ่งรบกวนในภาพ 10
2.4.1 รูปแบบและคุณลักษณะของสิ่งรบกวน 10
2.4.2 Convolution Technique 11
2.4.3 Spatial Domain Filtering 15
2.5 Edge Detection 17
2.5.1 Gradient Method 18
2.5.2 Laplacian Method 18
2.5.3 Canny Edge Detection Algorithm 20
2.5.3.1 Smoothing 21
2.5.3.2 Gradient Calculation 21
2.5.3.3 Nonmaxima Suppression 22
2.5.3.4 Thresholding 22
2.6 Blob Analysis 22
2.7 Morphological Image Processing 24
2.7.1 Logic Operation 25
2.7.2 Dilation และ Erosion 26
2.7.3 Opening และ Closing 27
2.8 การบีบอัดข้อมูลวิดีโอ 28
2.8.1 MPEG-4 28
2.8.2 Xvid 29
2.9 OpenCV Library 30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 การออกแบบและพัฒนา 31

3.1 Use Case Diagram 31

3.2 การออกแบบ GUI 32

3.3 การเขียนโปรแกรมเพื่อใช้งาน OpenCV Library 34

3.4 เทคนิคต่างๆ ที่ใช้ในการประมวลผลภาพ 36

3.4.1 การกำหนดจุดอ้างอิง 38

3.4.2 การตรวจจับการเคลื่อนไหว 39

3.4.3 Update Background 41

3.4.4 การทำ Blob Analysis 43

บทที่ 4 การทดลองและผลการทดลอง 46

4.1 แบบจำลองที่ใช้ในการจำลองระบบ 46

4.2 การทดลองและผลการทดลอง 47

4.3 สรุปผลการทดลอง 54

บทที่ 5 บทสรุป 55

5.1 สรุปผลการดำเนินงาน 55

5.2 ข้อเสนอแนะ 55

5.3 แนวทางการพัฒนาต่อ 56

บรรณานุกรม 57

ภาคผนวก ก คู่มือการใช้งาน 59

ภาคผนวก ข เกี่ยวกับ OpenCV 66

ภาคผนวก ค โปรแกรม Eagle Eye 77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 การคำนวนทางด้านตรรกศาสตร์ 25

4.1 ค่าต่างๆ ที่เป็นข้อกำหนดในการทดลอง 46

4.2 สรุปผลการทดลอง 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 ระบบสี RGB 5

2.2 ระบบสีแบบลบ 6

2.3 ระบบสี HSV 7

2.4 ตัวอย่างรูปที่ผ่านการกำจัด noise โดย Median Filter 17

2.5 กระบวนการของ Edge Detection 17

2.6 กราฟการทำ Edge Detection 18

2.7 ตัวอย่างการหาขอบภาพโดยใช้ Edge Detector แบบต่างๆ 19

2.8 ขั้นตอนของ canny edge detection 20

2.9 การทำงานของ Blob Analysis โดยใช้วิธี Floodfill 23

2.10 การคำนวณพื้นฐานของ Set 24

2.11 การทำ Logic Operation ของภาพ A และ B 25

2.12 การ Dilation และ Erosion 26

2.13 การ Opening และ Closing 27

2.14 สถาปัตยกรรมของ OpenCV Library 30

3.1 use case diagram 31

3.2 หน้าต่าง GUI ของโปรแกรม 32

3.3 ตัวอย่างสถานที่ในการทดลอง 33

3.4 แผนผังเทคนิคในการนับวัตถุที่เข้าหรือออกจากเฟรม 36

3.5 แผนผังเทคนิคในการนับคนที่อยู่ในเฟรม 37

3.6 การกำจัดสิ่งรบกวนด้วย window ขนาดต่างๆ 39

3.7 Flow Chart การทำงานของระบบนับวัตถุที่เข้าหรือออกจากเฟรม 40

3.8 Flow Chart การทำงานของระบบโดยใช้เทคนิค Update Background 42

3.9 ก่อน – หลัง การทำ Blob 43

3.10 การทำ dilate 44

3.11 ผลการทำงานเมื่อวัตถุเริ่มเดินเข้ามา 44

3.12 การทำงานตรวจจับคนและวัตถุ 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

4.1 แบบจำลองสถานที่ที่ใช้ในการทดลอง 46

4.2 หน้าจอแสดงผลการทดลอง (4.2.1) 47

4.3 หน้าจอแสดงผลการทดลอง (4.2.2) 48

4.4 หน้าจอแสดงผลการทดลอง (4.2.3) 49

4.5 หน้าจอแสดงผลการทดลอง (4.2.4-a) 50

4.6 หน้าจอแสดงผลการทดลอง (4.2.4-b) 51

4.7 หน้าจอแสดงผลการทดลอง (4.2.5-a) 52

4.8 หน้าจอแสดงผลการทดลอง (4.2.5-b) 52

4.9 หน้าจอแสดงผลการทดลอง (4.2.5-c) 53

ก.1 การเปิดไฟล์ 59

ก.2 การเลือกทิศทางการตรวจจับ 60

ก.3 การประมวลผล และ แสดงผลลัพธ์ 60

ก.4 เส้นตรวจจับและปุ่มที่ใช้ในการซ่อน 61

ก.5 การหยุดการประมวลผล 61

ก.6 การจัดเก็บไฟล์ผลลัพธ์ 62

ก.7 การรับภาพจาก Webcam 63

ก.8 การแจ้งเมื่อพร้อมใช้งาน Webcam 63

ก.9 ปุ่มที่ใช้เลือกโหมดการใช้งาน Webcam 64

ก.10 เวลาที่ปรากฏในไฟล์ 64

ก.11 ปุ่มที่ใช้กำหนดขนาดเส้นตรวจจับ 65

ข.1 อธิบายขั้นตอนการติดตั้ง OpenCV ขั้นที่ 1 66

ข.2 อธิบายขั้นตอนการติดตั้ง OpenCV ขั้นที่ 2 67

ข.3 อธิบายขั้นตอนการติดตั้ง OpenCV ขั้นที่ 3 67

ข.4 อธิบายขั้นตอนการติดตั้ง OpenCV ขั้นที่ 4 68

ข.5 อธิบายขั้นตอนการติดตั้ง OpenCV ขั้นที่ 5 68

ข.6 อธิบายขั้นตอนการติดตั้ง OpenCV ขั้นที่ 6 69

ข.7 อธิบายขั้นตอนการติดตั้ง OpenCV ขั้นที่ 7 69

ข.8 การตั้งค่า Visual Studio C++ ให้ใช้งาน OpenCV ขั้นที่ 1 70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

ข.9 การตั้งค่า Visual Studio C++ ให้ใช้งาน OpenCV ขั้นที่ 2 71
ข.10 การตั้งค่า Visual Studio C++ ให้ใช้งาน OpenCV ขั้นที่ 3 72
ข.11 การตั้งค่า Visual Studio C++ ให้ใช้งาน OpenCV ขั้นที่ 4 73
ข.12 ตัวอย่างการ include ไฟล์ต่างๆของ OpenCV 74
ค.1 อธิบายขั้นตอนการติดตั้ง Eagle Eye ขั้นที่ 1 75
ค.2 อธิบายขั้นตอนการติดตั้ง Eagle Eye ขั้นที่ 2 76
ค.3 อธิบายขั้นตอนการติดตั้ง Eagle Eye ขั้นที่ 3 77
ค.4 อธิบายขั้นตอนการติดตั้ง Eagle Eye ขั้นที่ 4 78
ค.5 อธิบายขั้นตอนการติดตั้ง Eagle Eye ขั้นที่ 5 78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 หลักการและเหตุผล

ปัจจุบันมีการประยุกต์ใช้คอมพิวเตอร์ ในการประมวลผลภาพเป็นอย่างมาก ไม่ว่าจะเป็นในโรงงานอุตสาหกรรม ตามหน่วยงานต่างๆ เช่น การค้นหาวัตถุในภาพ การตรวจจับวัตถุที่สนใจในภาพ การรู้จำใบหน้า นอกจากนี้ยังมีการนำคอมพิวเตอร์มาประมวลผลภาพการเคลื่อนไหวมากขึ้นเช่นการตรวจจับหาวัตถุที่เคลื่อนไหวในภาพวีดีทัศน์การตรวจจับหาผู้บุกรุก

โครงงานนี้จะนำเทคโนโลยีในการประมวลผลภาพมาประยุกต์ใช้ในการศึกษาถึงการจับการเคลื่อนไหวที่เกิดขึ้นในภาพวีดีทัศน์หรือภาพจากกล้องวงจรปิดเพื่อเพิ่มความสามารถในการบันทึกเหตุการณ์ต่างๆที่เกิดขึ้น โดยกล้องวงจรปิดในปัจจุบันไม่สามารถบอกได้ว่าเฟรมที่จับได้นั้นมีการเคลื่อนไหวหรือไม่ทำให้จำเป็นต้องบันทึกภาพต่างๆที่เกิดขึ้นอยู่ตลอดเวลาซึ่งเป็นการสิ้นเปลืองพื้นที่บนหน่วยความจำไปโดยเปล่าประโยชน์ ดังนั้นถ้ากล้องวงจรปิดสามารถระบุได้ว่าเวลาไหนมีการเคลื่อนไหวเกิดขึ้นและบันทึกเฟรมนั้นๆลงไฟล์ก็จะเป็นการประหยัดพื้นที่บนหน่วยความจำและไม่เสียเวลาในเรียกดูไฟล์ในภายหลัง และยังเพิ่มความสามารถในการนับวัตถุที่ผ่านเข้าออกเฟรม และจำแนกว่าวัตถุเหล่านั้นมีความน่าจะเป็นที่จะเป็นการเคลื่อนไหวของมนุษย์หรือไม่

1.2 วัตถุประสงค์

- ศึกษาการประมวลผลภาพวีดีทัศน์ด้วยคอมพิวเตอร์
- พัฒนาความสามารถในการจับวัตถุที่เคลื่อนไหวในภาพวีดีทัศน์
- พัฒนาระบบในการนับวัตถุในภาพวีดีทัศน์
- พัฒนาให้กล้องเวปแคมมีความสามารถในการบันทึกไฟล์วิดีโอได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงงาน

- สามารถวิเคราะห์ได้ว่ามีการเคลื่อนไหวในภาพวีดีทัศน์
- สามารถแยกแยะวัตถุที่เคลื่อนไหวในวีดีทัศน์ออกมาได้
- สามารถวิเคราะห์การเคลื่อนไหวเหล่านั้นว่าเป็นมนุษย์หรือไม่
- สามารถนับวัตถุที่ผ่านเข้าหรือออกจากเฟรมได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- มีความรู้ความเข้าใจเกี่ยวกับการประมวลผลภาพ
- ได้รับความรู้เกี่ยวกับการเขียนโปรแกรมเพื่อนำมาใช้ในงานด้านการประมวลผลภาพ
- สามารถนำทฤษฎีด้านการประมวลผลภาพมาใช้ในชีวิตจริงได้
- เป็นแนวทางในการวิจัย การรู้จำการเคลื่อนไหวของร่างกาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 การประมวลผลภาพ

ในปัจจุบันวิวัฒนาการทางเทคโนโลยีก้าวหน้ามาก ทำให้การสร้างหรือการได้มาซึ่งรูปภาพนั้นเป็นไปได้ง่าย ทำให้ข้อมูลกลายมาเป็นส่วนหนึ่งในชีวิตประจำวัน ไม่ว่าจะเป็นการถ่ายภาพเพื่อเก็บเป็นที่ระลึกหรือการถ่ายภาพเพื่อใช้งาน ทำให้การประมวลผลรูปภาพมีความสำคัญเพิ่มขึ้นในการช่วยดำเนินการกับรูปภาพเพื่อให้ข้อมูลภาพนั้นอยู่ในรูปแบบที่เหมาะสมและมีคุณภาพดีเพียงพอที่จะนำไปใช้รูปแบบต่างๆ การประมวลผลรูปภาพที่เราจะกล่าวในที่นี้ไม่ใช่เป็นการจัดการกับรูปภาพโดยใช้อุปกรณ์ทางเคมีหรือ อุปกรณ์เกี่ยวกับแสง แต่เป็นการจัดการหรือประมวลผลภาพถ่ายดิจิตอล(Digital)เท่านั้น ซึ่งจะดำเนินการโดยใช้การคำนวณและประมวลผลทางคอมพิวเตอร์ โดยกระบวนการต่างๆเกี่ยวกับรูปภาพ มีดังนี้

2.1.1 การรับภาพและแปลงภาพ

เป็นการรับภาพจากตัวตรวจจับและแปลงสัญญาณภาพให้เป็นภาพถ่ายดิจิตอล โดยขั้นตอนการแปลงสัญญาณภาพนี้จะมีการทำ Sampling สัญญาณภาพที่ได้รับ จากนั้นนำค่าที่ได้ในแต่ละพิกเซล (Pixel) มาทำควอนไซ เพื่อแปลงค่าจำนวนจริงที่เป็น Discrete ที่ได้จากการ Sampling ให้เป็นค่าจำนวนเต็ม Discrete จากนั้นจำนวนเต็มนี้จะนำมาแปลงเป็นเลขฐานสองด้วยขั้นตอน Binary Representation ขั้นตอนการ Sampling และ Quantization นี้จะต้องมีการดำเนินการในลักษณะที่ลดความผิดเพี้ยนที่เกิดจากตัว Sensor เพื่อให้ได้ภาพที่มีคุณภาพ

2.1.2 การปรับปรุงคุณภาพของภาพ

เป็นการปรับปรุงหรือซ่อมแซมให้ข้อมูลภาพที่มีอยู่นั้นมีคุณภาพดีขึ้น เช่น ภาพที่ได้มาอาจมีContrast น้อยหรือเบลอ ไม่คมชัด เราสามารถปรับภาพให้คมชัดได้ด้วยเทคนิค เช่น Contrast Enhancement หรือ Edge Enhancement หรือในกรณีที่ภาพที่มีอยู่มีความไม่สมบูรณ์ เช่น มีสิ่งรบกวน(Noise) เราสามารถใช้เทคนิคการกรองสัญญาณภาพ (Image Filtering) เพื่อกำจัดสิ่งรบกวนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 การบีบอัดภาพ

เทคนิคการประมวลผลรูปภาพแบบนี้เป็นหัวใจสำคัญในการจัดการข้อมูลภาพเพื่อการสื่อสารรับส่งข้อมูลภาพและการจัดเก็บข้อมูลภาพด้วย เนื่องจากข้อมูลภาพนั้นมีขนาดใหญ่มาก โดยเฉพาะภาพสี การจัดเก็บข้อมูลหรือรับส่งไฟล์ข้อมูลขนาดใหญ่นั้นจะต้องเสียเวลามาก ดังนั้น เพื่อทำการจัดการเก็บและการรับส่งข้อมูลภาพเป็นไปได้รวดเร็วขึ้น จึงได้มีการทำการบีบอัดข้อมูลภาพก่อนเพื่อให้ขนาดของข้อมูลเล็กลง ตัวอย่างการบีบอัดข้อมูลภาพ เช่น JPEG หรือ GIF หรือการบีบอัดข้อมูลวิดีโอ เช่น MPEG หรือ AVI เป็นต้น

2.1.4 การวิเคราะห์ข้อมูลภาพ

เป็นการวัดหรือวิเคราะห์ข้อมูลภาพเชิงปริมาณ เพื่อสร้างคำจำกัดความให้กับภาพ แล้วนำไปวิเคราะห์ใช้งาน ตัวอย่างเช่น การนำภาพถ่ายของเซลล์เม็ดเลือดมาวิเคราะห์หาขนาดและมุมเอียงแล้วนำข้อมูลนี้ไปตรวจสอบความผิดปกติของเม็ดเลือดในผู้ป่วยหรือการนำภาพถ่ายมาวิเคราะห์เพื่อนำไปควบคุมการทำงานของหุ่นยนต์ เช่น ควบคุมให้หุ่นยนต์เคลื่อนย้ายวัตถุที่มีลักษณะตามกำหนดในการวิเคราะห์ข้อมูลภาพจำเป็นต้องมีการแยกแยะคุณลักษณะของวัตถุต่างๆ ในภาพเพื่อช่วยในการระบุชนิดของวัตถุนั้น นอกจากนี้ยังต้องมีการทำ Segmentation (การคัดแยก) วัตถุที่สนใจออกมาจากรูปภาพหรือฉากนั้นๆ เพื่อให้สามารถหาคุณลักษณะของวัตถุที่ต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

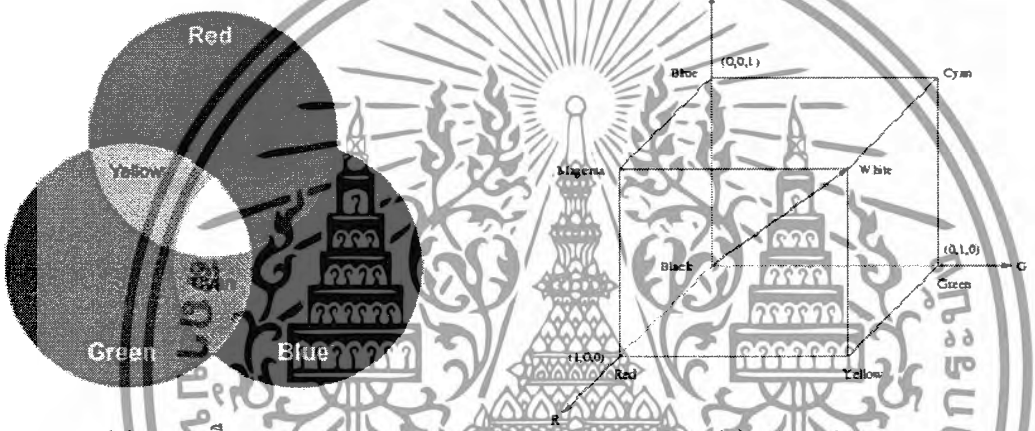
2.2 การประมวลผลภาพสี

2.2.1 มาตรฐานของสี

มาตรฐานของสีแบ่งออกเป็น 4 ระบบดังต่อไปนี้

2.2.1.1 ระบบสี RGB

ระบบสี RGB เป็นสีที่ใช้สำหรับการใช้งานกับจอภาพคอมพิวเตอร์ และเป็นระบบสีที่เกิดจากการผสมสีของแสงที่ประกอบด้วยแม่สีหลักของแสง 3 สี คือ สีแดง (Red) สีเขียว (Green) และน้ำเงิน (Blue) ในสัดส่วนความเข้มของแม่สีแต่ละตัวที่แตกต่างกัน เป็นแบบจำลองสี RGB เมื่อสีเหล่านี้มีการผสมกันแล้วเกิดเป็นสีสันต่างๆ จึงเรียกว่า "สีแบบบวก" (Additative Colors)



(ก) ระบบสีแบบบวก (ข) RGB cube

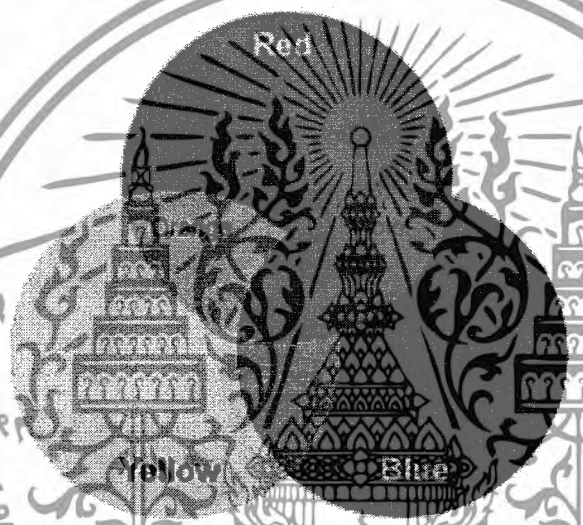
รูปที่ 2.1 ระบบสี RGB

แบบจำลองนี้สามารถแทนได้โดยใช้รูปลูกบาศก์ขนาด 1 หน่วย ดังรูปที่ 2.1 (ข) โดยที่สี RGB เป็นแกนทั้ง 3 ของรูปลูกบาศก์ โดยแต่ละสีมีค่าอยู่ในช่วง [0, 1] 0 แสดงถึงความมืด และ 1 แสดงถึงความสว่าง ภาพที่ได้จากแบบจำลอง RGB จะประกอบด้วยระนาบภาพสี RGB 3 ระนาบ ภาพที่เป็นอิสระต่อกัน ดังนั้นถ้าต้องการสร้างภาพจำลองบนจอภาพจะต้องอาศัยการผสมกันของระนาบภาพทั้งสาม จึงจะทำให้เกิดสีต่างๆ บนจอภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.2 ระบบสี CMYK

ระบบสี CMYK เป็นสีที่ใช้สำหรับการใช้งานกับเครื่องพิมพ์ และเป็นการถูกดูดกลืนแสงสะท้อนจากวัตถุของสี เมื่อแสงสีขาวตกกระทบกับวัตถุสีต่างๆ วัตถุนั้นจะดูดกลืนแสงที่มีความยาวคลื่นบางระดับไว้แล้วสะท้อนออกมาแสดงให้เห็นปริมาณที่ลดลง ประกอบด้วยสีม่วงแดง(Magenta) สีน้ำเงินแกมเขียว(Cyan) สีเหลือง (Yellow) และสีดำ (Black) เมื่อถูกผสมเข้าด้วยกันจะเห็นเป็นสีดำ จึงเรียกระบบสีนี้ว่า "ระบบสีแบบลบ" (Subtractive Colors System) แสดงดังรูปที่ 2.2 ซึ่งสามารถแทนแบบจำลองนี้ในรูปลูกบาศก์ขนาด 1 หน่วยได้การกำหนดมุม 3 มุมที่เหลือในแบบจำลอง RGB

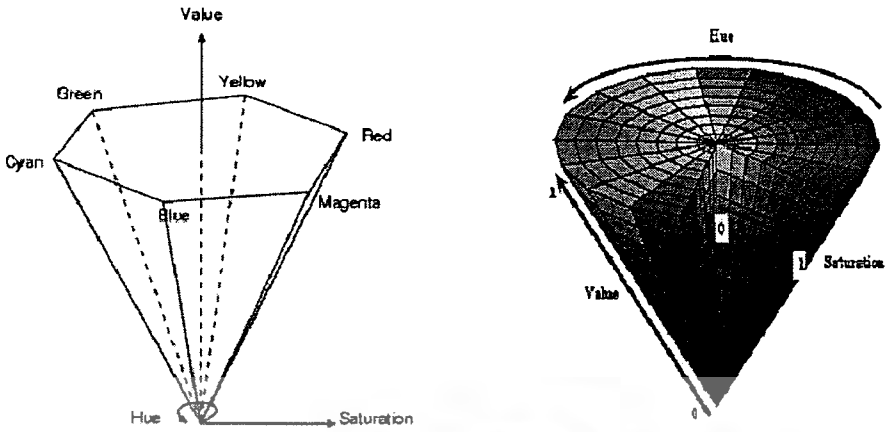


รูปที่ 2.2 ระบบสีแบบลบ

2.2.1.3 ระบบสี HSV

ระบบสี HSV เป็นสีพื้นฐานสำหรับการมองเห็นสีในสายตามนุษย์ และเป็นแบบจำลองสีอีกแบบหนึ่งที่พิจารณาสีโดยใช้ Hue Saturation และ Value แสดงดังรูปที่ 2.3 ซึ่ง H (Hue) คือค่าสีของสีหลัก (แดง เขียว และน้ำเงิน) สีเหล่านี้จะถูกจัดเป็นระเบียบด้วยการสร้างเป็นวงล้อสีที่มีค่าอยู่ระหว่าง 0 ถึง 255 ซึ่งถ้าค่า Hue มีค่าเท่ากับ 0 จะแทนสีแดง และเมื่อ Hue มีค่าเพิ่มขึ้นเรื่อยๆ สีก็จะเปลี่ยนแปลงไปตาม Spectrum ของสีจนถึง 256 จึงจะกลับมาเป็นแดงอีกครั้ง ส่วน S (Saturation) คือความเข้ม-ความอ่อนของสีที่วัดเป็นระยะห่างจากจุดศูนย์กลางของวงล้อสี เมื่อสีที่มี S=0 จะได้เป็นสีขาว เมื่อสีที่มีค่า S=1 จะได้สีที่มีความเข้มมากของสีนั้น และ V (Value) คือปริมาณแสงหรือปริมาณความสว่างของสี ถ้าค่าความสว่างสูงจะได้สีที่มีความสว่างมาก เมื่อ V=0 จะได้เป็นสีดำ V=1 จะได้สีที่สว่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 ระบบสี HSV

2.2.1.4 ระบบสี YIQ

ระบบสี YIQ เป็นระบบที่ใช้ใน TV Broadcasting เพื่อใช้งานกับ โทรทัศน์แบบขาวดำ โดยที่ค่า y คือความเข้มแสง ส่วน I และ Q เป็นสัญญาณที่เข้ารหัสสีของภาพ

2.2.2 การแปลงค่าสีระหว่างระบบสีต่างๆ

การแปลงค่าสีในส่วนนี้จะอธิบายถึงการแปลงค่าภาพสี RGB เป็นระบบสีต่างๆ

2.2.2.1 การแปลงภาพสี RGB เป็นภาพระดับสีเทา

เนื่องจากการประมวลผลภาพสีค่อนข้างยุ่งยาก ดังนั้นจึงนิยมที่จะแปลงภาพสีเป็นภาพระดับเทาก่อนการประมวลผลภาพ โมเดลสีที่พยายามแยกส่วนประกอบของสีกับส่วนประกอบของความเข้มแสงออกจากกัน และส่วนประกอบของความเข้มแสงนี้เองที่เป็นส่วนแสดงค่าระดับเทาของสีแต่ละภาพ โมเดลสีที่เป็นภาพระดับเทาที่ใช้กัน ได้แก่ โมเดลสี YIQ ซึ่งมีค่า Y เป็นค่าแสดงความเข้มแสง คำนวณได้จากสมการ

$$Y = 0.212671R + 0.715160G + 0.072169B \quad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2.2 การแปลงภาพสี RGB เป็น HSV

สามารถคำนวณแต่ละสีได้ดังนี้

Hue $$H = \cos^{-1}\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)^2+(G-B)^2}}\right\} \quad (2.2)$$

Saturation $$S = 1 - \frac{3}{R+G+B}(\min(R+G+B)) \quad (2.3)$$

Value $$V = \frac{R+G+B}{3} \quad (2.4)$$

2.2.2.3 การแปลงภาพสี RGB เป็น CMY

สามารถคำนวณแต่ละสีได้ดังนี้

$$C = 255 - R \quad (2.5)$$

$$M = 255 - G \quad (2.6)$$

$$Y = 255 - B \quad (2.7)$$

2.3 การสร้างภาพไบนารี่

ปัจจุบันอุปกรณ์ที่มีความสามารถในการแสดงผลได้แค่ 2 ระดับ หรือ 2 สี คือ สีขาวกับสีดำ ยังมีการใช้กันอย่างแพร่หลาย เช่น เครื่องพิมพ์ (Printer), เครื่องโทรสาร (Fax), จอภาพแสดงผลแบบโมโนโครม (Monochrome Monitor) เป็นต้น เนื่องจากอุปกรณ์เหล่านี้เป็นอุปกรณ์ที่มีราคาถูก ดังนั้นจึงเกิดปัญหาในการแสดงผลขึ้น การที่จะแสดงผลหรือพิมพ์รูปภาพที่มีระดับความเข้มของภาพหลายระดับซึ่งมีมากกว่าความสามารถในการแสดงผลของอุปกรณ์เหล่านั้นที่มีเพียงแค่ 2 ระดับ เราจะต้องทำการแปลงข้อมูลภาพให้เป็นภาพไบนารี่ (Binary Image) เสียก่อน ซึ่งการสร้างภาพไบนารี่นั่นก็หมายถึงการแปลงข้อมูลภาพที่มีระดับความเข้มหลายระดับ (Multilevel Image) ให้เป็นภาพที่มีระดับความเข้มเพียง 2 ระดับ นั่นคือ 1 จุดภาพ มีได้ 2 ค่าเท่านั้น คือ 0 กับ 1 โดยจุดภาพที่แทนด้วย 0 หมายถึงจุดภาพที่มีสีดำ ส่วนจุดที่แทนด้วย 1 จะหมายถึงจุดภาพที่มีสีขาว เมื่อทำการแปลงเป็นภาพไบนารี่แล้ว จึงสามารถนำภาพนั้นไปแสดงผลบนอุปกรณ์เหล่านั้น จะเห็นได้ว่าการแปลงข้อมูลภาพหลายระดับเป็นภาพไบนารี่จึงมีความจำเป็นและมีประโยชน์มากในการแสดงผลภาพที่มีความเข้มของภาพหลายระดับบนอุปกรณ์ที่มีความสามารถในการแสดงผลได้ 2 ระดับ สำหรับประโยชน์อีกประการหนึ่งในการแปลงข้อมูลภาพนั้นเป็นภาพไบนารี่ คือการลดเนื้อที่การเก็บข้อมูลภาพ ภาพไบนารี่นี้จะสามารถลดเนื้อที่ลงได้ถึง 8 เท่า นั่นคือถ้าเดิม 1 จุดภาพใช้เนื้อที่ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเก็บ 8 บิต เมื่อทำการแปลงเป็นภาพไบนารี่แล้วจะใช้เนื้อที่ในการเก็บ 1 จุดภาพนั้นเพียง 1 บิต เท่านั้นอีกทั้งยังสามารถนำไปประยุกต์ใช้งานได้อย่างแพร่หลาย เช่น นำไปประยุกต์ใช้ในการวิเคราะห์เอกสารในขั้นตอนที่เรียกว่าการประมวลผลขั้นต้น (Preprocessing) เป็นต้น

ในการสร้างภาพไบนารี่ สามารถทำได้โดยใช้เทคนิคการทำเทรชโฮล (Thresholding Technique) โดยพิจารณาว่าจุดภาพใดควรจะเป็นจุดขาวหรือจุดดำ จะกระทำโดยการเปรียบเทียบระหว่างจุดภาพเริ่มต้นกับค่าคงที่ค่าหนึ่งที่เรียกว่า ค่าเทรชโฮล (Threshold Value) เทคนิคนี้ใช้กันมากในกรณีที่ข้อมูลภาพมีลักษณะแตกต่างกันระหว่างวัตถุ (Object) และพื้นหลัง (Background) โดยค่าของจุดภาพใด ๆ ที่มีค่าน้อยกว่าค่าเทรชโฮลจะถูกกำหนดให้เป็น 0 (จุดดำ) และถ้าค่าของจุดภาพใด ๆ ที่มีค่ามากกว่าหรือเท่ากับค่าเทรชโฮล จะถูกเปลี่ยนให้เป็น 1 (จุดขาว)ในการสร้างภาพไบนารี่โดยใช้เทคนิคเทรชโฮลเพื่อให้ได้ผลลัพธ์ที่เหมาะสมและคมชัด สิ่งที่สำคัญที่สุดคือ ค่าเทรชโฮล เนื่องจากถ้าเลือกค่าเทรชโฮลที่ไม่เหมาะสม (ค่าเทรชโฮลที่มีค่าน้อยเกินไปหรือมากเกินไป) ภาพที่ได้อาจจะไม่เหมาะสม ขาดความคมชัด และอาจทำให้รายละเอียดบางส่วนขาดหายไป กล่าวคือภาพที่ได้อาจจะมืดเกินไปหรือสว่างเกินไป หรือภาพที่ได้มีสิ่งรบกวน(Noise)เกิดขึ้น อันเป็นผลทำให้ภาพผลลัพธ์ที่ได้ไม่สวยงามเท่าที่ควร ดังนั้นปัญหาของการสร้างภาพไบนารี่โดยวิธีเทรชโฮลนี้คือ ทำอย่างไรจึงจะสามารถคำนวณหาค่าเทรชโฮลที่เหมาะสมสำหรับแต่ละภาพที่จะนำมาทำการสร้างภาพไบนารี่ ซึ่งมีวิธีคำนวณหาค่าเทรชโฮลหลายวิธี โดยแต่ละวิธีเหมาะสมกับลักษณะการทำงานที่แตกต่างกันไป เช่น การหาค่าเทรชโฮลโดยการกำหนดค่าล่วงหน้า(Pre-assigned Threshold Value) การหาค่าเทรชโฮลจากค่ากลาง (Mid-range Threshold Value) ซึ่งแต่ละวิธีอธิบายได้ดังนี้

2.3.1 การหาค่าเทรชโฮลโดยการกำหนดค่าล่วงหน้า

การหาค่าเทรชโฮลโดยการกำหนดค่าล่วงหน้านี้เป็นวิธีที่ง่ายที่สุด โดยจะการคำนวณค่าเทรชโฮลโดยการกำหนดเองจากผู้ใช้ ซึ่งการกำหนดนี้จะขึ้นอยู่กับประสบการณ์ของผู้ใช้นั้น ๆ ทำได้โดยการเลือกค่าคงที่ค่าหนึ่ง ซึ่งเรียกค่านั้นว่า "ค่าเทรชโฮล" โดยค่าที่เลือกมานี้จะเป็นค่าที่อยู่ระหว่างค่าต่ำสุดและค่าสูงสุดของระดับความเข้มของข้อมูลภาพอินพุท เช่น ภาพข้อมูลอินพุทมีเกรย์สเกล 256 ระดับ จะมีค่าเกรย์สเกลได้ตั้งแต่ 0-255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 การหาค่าเทรชโฮลจากค่ากลาง (Mid-Range Threshold Value)

การหาค่าเทรชโฮลโดยพิจารณาจากค่ากลาง เป็นการหาค่าเทรชโฮลที่แตกต่างจากการหาค่าเทรชโฮลวิธีแรก สำหรับวิธีนี้จะเป็นการคำนวณหาค่าเทรชโฮลโดยอัตโนมัติโดยผู้ใช้ไม่ต้องเป็นผู้กำหนดเอง การคำนวณหาค่าได้อาศัยการคำนวณพื้นฐานทางสถิติในเรื่องของการหาค่ากลาง หรือค่าเฉลี่ย (Mean) มาประยุกต์ใช้ ซึ่งค่าเทรชโฮลที่คำนวณได้จะเป็นค่าที่เหมาะสมสำหรับภาพนั้น ๆ

2.3.3 การหาค่าเทรชโฮลโดยอัตโนมัติ

เป็นการทำ threshold ที่อ้างอิงจากทฤษฎี Otsu's method ซึ่งเป็นไปตามสมการดังนี้

$$\sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t) \tag{2.8}$$

ω_i คือน้ำหนักของความน่าจะเป็นของพื้นหลังพื้นหน้าซึ่งแบ่งแยกโดยค่า threshold t และตัวแปร σ_i^2 ของ class เหล่านั้น

2.4 การกำจัดสิ่งรบกวนในภาพ

การปรับปรุงคุณภาพของภาพเราสามารถทำได้หลายวิธี โดยวิธีที่ค่อนข้างสำคัญวิธีหนึ่งก็คือ การกำจัดสิ่งรบกวนในภาพ สิ่งรบกวนนี้อาจเกิดจากความผิดปกติของตัว Image Sensor ซึ่งเป็นผลมาจากสภาพแวดล้อมขณะรับภาพ หรือคุณภาพที่ผิดเพี้ยนไปของตัว Sensor เอง

2.4.1 รูปแบบและคุณลักษณะของสิ่งรบกวน

สำหรับภาพที่ถูกรบกวนด้วยสิ่งรบกวนหรือ Noise สามารถแสดงได้ดังนี้

$$g(x, y) = f(x, y) + \eta(x, y) \tag{2.9}$$

โดยที่ $\eta(x, y)$ เป็น Noise ที่เกิดขึ้นกับค่าความเข้มแสงของภาพ ณ ตำแหน่งพิกเซล (x,y) ใดๆ ในภาพ

รูปแบบของ Noise นี้สามารถแบ่งได้ตามลักษณะการกระจายของความถี่ของความเข้มแสงของNoise หรือค่า Probability Density Function (PDF) ของ Noise รูปแบบของ Noise ที่นิยมนำมาใช้เป็นแบบจำลองในการวิเคราะห์ผลของ Noise ในการประมวลผลภาพมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Gaussian Noise มักเกิดกับภาพถ่ายทั่วไปเกิดจากสัญญาณรบกวนจากวงจรทางอิเล็กทรอนิกส์หรืออุณหภูมิสูงมารบกวนกับ Sensor ของตัวกล้อง
- Rayleigh Noise หรือ Exponential Noise มักเกิดกับภาพจาก Laser
- Uniform Noise เป็นรูปแบบที่ใช้กันน้อยมากในทางปฏิบัติ แต่เป็นรูปแบบการกระจายที่เป็นประโยชน์ที่ใช้เป็นพื้นฐานในการสร้างตัวแปร Random ให้มีการกระจายในรูปแบบอื่นๆ เพื่อใช้ในการทำ Simulation ผลกระทบของ Noise ได้
- Impulse (Salt and Pepper) Noise เกิดได้จากการถูกรบกวนโดยฉับพลัน สามารถเกิดได้กับภาพทั่วไป

การเรียนรู้ถึงลักษณะรูปแบบการกระจายของ Noise หรือรูปร่างของ PDF ของ Noise นั้นเป็นประโยชน์ในการวิเคราะห์และแก้ไขผลกระทบของ Noise ชนิดต่างๆ ได้ Noise แต่ละชนิดมักจะเกิดในภาพถ่ายต่างๆ ชนิดกัน

2.4.2 Convolution Technique

เทคนิค Convolution นี้เป็นเทคนิคพื้นฐานในการประมวลผลภาพ ใช้เป็นพื้นฐานในการทำการกรอง Noise เทคนิคนี้ จะใช้ภาพย่อย (Subimage, Filter, Mask, Kernel, Template หรือ Window) เคลื่อนที่ไปตามพิกเซลต่างๆ ในภาพ โดยค่าในภาพย่อยหรือ Filter นี้เราเรียกว่า Filter Coefficients แทนที่จะเป็นค่าความเข้มแสงของพิกเซล ณ แต่ละตำแหน่งของแต่ละพิกเซลตำแหน่ง (x, y) ผลตอบสนองต่อตัว Filter ที่จุดนั้นจะเป็นค่าผลบวกของผลคูณระหว่าง Filter Coefficients และค่าความเข้มแสง ณ ตำแหน่งพิกเซลที่สัมพันธ์กันทั้งหมดในพื้นที่ของ Filter Mask เทคนิคการคำนวณเช่นนี้เราเรียกว่า Convolution ดังตัวอย่าง ถ้าเรามี Filter Mask(W) ซึ่งมีขนาด 3x3 พิกเซล

$$\begin{bmatrix} w(-1,-1) & w(-1,0) & w(-1,1) \\ w(0,-1) & w(0,0) & w(0,1) \\ w(1,1) & w(1,0) & w(1,1) \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลตอบสนอง (R)ที่ได้สำหรับตำแหน่ง (x, y)ในภาพจะมีค่าดังนี้

$$\left\{\begin{array}{l} w(-1,-1)f(x-1,y-1)+w(-1,0)f(x-1,y)+ \\ w(-1,1)f(x-1,y+1)+w(0,-1)f(x,y-1)+ \\ w(0,0)f(x,y)+w(0,1)f(x,y+1)+ \\ w(1,-1)f(x+1,y-1)+w(1,0)f(x+1,y)+ \\ w(1,1)f(x+1,y+1) \end{array}\right\}$$

Filter Coefficient w(0, 0) เป็นค่าที่ตรงกับตำแหน่ง (x,y) ที่ต้องการทำ Filtering ซึ่งเป็นตำแหน่งจุดกึ่งกลางของ Mask ปกติ Mask จะมีขนาด M x N โดยที่ค่า M = 2a + 1 และ N = 2b + 1 ซึ่ง a และ b เป็นค่าจำนวนเต็มไม่เป็นลบ เราสามารถเขียนการทำ Filtering ของภาพ f ขนาด M x N ด้วย Filter Mask ได้ดังนี้

$$g(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b}w(s,t)f(x+s,y+t) \qquad (2.10)$$

โดยค่า a = (M – 1) / 2 และ b = (N - 1) / 2 ทั้งนี้เราสามารถเขียน Pseudo Code สำหรับการทำ Convolution ได้ดังนี้

1. สร้าง Array สำหรับ Filter Mask (w) สำหรับเก็บค่า Filter Coefficients
2. a = (M – 1) / 2 และ b = (N - 1) / 2

```
Sum = 0
    For i=-a to a
        For j= -b to b
            Sum = Sum + w (i+a, j+b) f (x-i, y-j)
        End For
    End For
```

3. g(x,y) = Sum

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อหลีกเลี่ยงการเขียนสมการยาวๆ เหล่านี้สำหรับการทำ Convolution เราสามารถเขียนสมการ Convolution สำหรับภาพทั้งภาพโดยย่อได้ดังนี้

$$g = h*f \tag{2.11}$$

ในการทำ Convolution นี้เราต้องพบกับปัญหาใหญ่ๆ ในการคำนวณ 2 ปัญหาด้วยกัน ปัญหาแรกคือ ปัญหาการเก็บค่าผลลัพธ์ (g) ซึ่งจากการทำ Convolution ในแต่ละตำแหน่งอาจให้ผลลัพธ์ที่เป็นลบหรือมีค่าเกินช่วงของข้อมูลที่เรากำหนดไว้ ดังนั้นการสร้าง Array เพื่อเก็บค่าผลลัพธ์ (g)ควรกำหนดให้ข้อมูลเป็นแบบที่สามารถรองรับค่าของผลลัพธ์ที่ต้องการได้ เช่น ถ้าค่าผลลัพธ์มีโอกาสติดลบก็ให้กำหนดชนิดของข้อมูลเป็นแบบมีเครื่องหมาย (Sign Data) หรือแปลงค่าที่ต่ำกว่า 0 ให้ขยับขึ้นเป็น 16 บิตหรือ 32 บิต หรือทำการ Normalize ผลลัพธ์เพื่อให้อยู่ในช่วงเดียวกับชนิดของข้อมูลภาพที่เป็น Input ได้

ปัญหาที่สองเป็นปัญหาที่เกิดขึ้นที่ตำแหน่งของขอบภาพ ซึ่งไม่สามารถทำ Convolution ได้เนื่องจากบางส่วนของ Kernel หรือ Mask อยู่นอกภาพ ตัวอย่างเช่น filter mask ขนาด 3 x 3 พิกเซล 1 พิกเซลที่ขอบไม่สามารถทำ Convolution ได้ เพื่อแก้ปัญหานี้สามารถทำได้หลายวิธีเช่น

- ไม่ต้องทำ Convolution ที่ขอบ

วิธีนี้เป็นวิธีที่ง่ายที่สุด คือไม่สนใจพิกเซลที่ขอบที่ไม่สามารถทำ Convolution ได้ไปเลย และเนื่องจากก่อนจะทำ Convolution เราต้องทำการกำหนดค่าเริ่มต้นให้ Array เก็บผลลัพธ์ ให้เป็น 0 ทุกๆ ตำแหน่งใน Array ซึ่งการไม่ทำ Convolution ที่ขอบของภาพนี้ จะทำให้ตำแหน่งเหล่านี้มีค่าเป็น 0 หรือทำให้ขอบของภาพผลลัพธ์เป็นสีดำ ซึ่งอาจทำให้เกิดผลกระทบที่สำคัญในการคำนวณสถิติของความเข้มแสงของภาพและมีผลต่อการประมวลผลภาพจากภาพผลลัพธ์ได้

- ลอกค่าของภาพ Input ในตำแหน่งขอบไปใส่ไว้ที่ขอบของภาพผลลัพธ์

วิธีนี้ทำได้โดยตำแหน่งที่ทำที่ขอบของภาพ ที่ไม่สามารถทำ Convolution ได้ ให้นำค่าตำแหน่งนั้นที่ภาพ Input มาใส่แทน ดังนั้นค่าความเข้มแสงที่ตำแหน่งขอบภาพ จะเป็นค่าที่ไม่ถูกประมวลผลของภาพ Input นั่นเอง ซึ่งวิธีนี้จะมีผลน้อยกว่าข้อ1 ซึ่งขอบเป็นสีดำ

- ทิ้งตำแหน่งขอบภาพไปเลย

วิธีนี้จะทำการตัดผลลัพธ์ที่ตำแหน่งที่ไม่สามารถหาค่า Convolution ทิ้งไปเลยทำให้ผลลัพธ์ มีขนาดเล็กลงกว่าภาพ Input ซึ่งวิธีนี้อาจทำให้เกิดปัญหาตามมา ในกรณีที่ต้องการทำการประมวลผลทางคณิตศาสตร์หรือเปรียบเทียบภาพ Input และภาพผลลัพธ์ตำแหน่งต่อตำแหน่ง เนื่องจากขนาดของภาพทั้งสองไม่เท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ใช้ Filter mask ขนาดที่เหมาะสมกับบริเวณขอบภาพ

วิธีนี้จะใช้ Filter ขนาดพิเศษสำหรับตำแหน่งขอบเท่านั้น ตัวอย่างเช่น ถ้าต้องการทำ Convolution ด้วย Filter Mask ขนาด 3x3 ซึ่งมีค่า

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

เราสามารถตัดตัว Filter mask ให้ใช้กับมุมของภาพ ขอบด้านซ้ายและขวา และขอบบนและล่างดังนี้

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

ซึ่งผลจากการใช้ Filter Mask ที่ถูกตัดนี้จะให้ผลเช่นเดียวกับใช้ Filter Mask ขนาดเต็มหรือไม่ขึ้นกับค่าของ Filter Coefficient ทำให้การใช้เทคนิคนี้เพิ่มความซับซ้อนในการหาค่า Filter Coefficient ที่เหมาะสมและให้ผลเช่นเดียวกับ Filter Mask ขนาดเต็ม

- การกำหนดดัชนีแบบสะท้อน (Reflected Indexing)

ปัญหาการทำ Convolution ที่ขอบของภาพนั้นเกิดจากดัชนี (Index) ของภาพ ณ ตำแหน่งนั้น f(x - i, y - j) ค่าของ x - i และ y - j เป็นค่าติดลบหรือออกไปนอกภาพ ดังนั้นเราสามารถตรวจสอบค่าดัชนีเพื่อดูว่าค่าตำแหน่งพิกเซล x ใช้ได้หรือไม่ ถ้าใช้ไม่ได้ เราสามารถสะท้อนหรือย้อนกลับเข้ามาในภาพในตำแหน่งที่มีค่าได้ การตรวจสอบนี้ทำเช่นเดียวกับตำแหน่งพิกเซล y ด้วย

- การใช้ดัชนีภาพแบบวนรอบ (Circular Indexing)

วิธีนี้ใช้ค่าจริงของภาพมาทำ Convolution ในบริเวณขอบภาพ แต่แทนที่จะทำการสะท้อนเอาค่าที่อยู่ในตำแหน่งรอบๆ บริเวณที่ต้องการคำนวณค่ามาใช้ วิธีนี้ให้วนไปใช้ค่าอีกฝั่งของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 Spatial Domain Filtering

เมื่อมี Noise รบกวนภาพ ทำให้ภาพนั้นสูญเสียรายละเอียดไป การที่จะกำจัด Noise ให้หมดไปจากภาพเลยนั้นทำได้ยาก เนื่องจากตำแหน่งและขนาดของ Noise ที่เกิดการรบกวนนั้นไม่แน่นอน ที่เราทำได้ก็เพียงแต่เป็นการลดผลกระทบของ Noise เพื่อช่วยให้ได้ภาพที่มีรายละเอียดดีขึ้นและมีคุณภาพดีพอที่จะนำไปใช้งานได้ เทคนิคในการลดผลกระทบของ Noise ที่เราเรียกว่า Noise Filtering มีหลายเทคนิคด้วยกัน เช่น

- **Order - Statistics Filter**

Order-Statistics Filter เป็นเทคนิคแบบ Nonlinear ซึ่งผลที่ได้ขึ้นอยู่กับการจัดเรียงค่าของพิกเซลที่อยู่ในพื้นที่ที่ต้องการทำ Filter หรือ Filter Mask จากนั้นก็จะแทนค่าพิกเซลนั้นด้วยค่าผลลัพธ์ที่ ได้จากการจัดเรียงค่าของพิกเซล ตัวอย่างของเทคนิค Order - Statistics Filter ได้แก่ Median Filter, Max - Min Filter, Midpoint Filter และ Alpha - Trimmed Mean Filter แต่จะขออธิบายเฉพาะที่นำมาใช้ในโครงงานคือ Median Filter

Median Filter เป็นเทคนิคแบบ Order - Statistics Filter ที่เป็นที่รู้จักกันดีที่สุด เทคนิคนี้จะทำการแทนที่ค่าของพิกเซลนั้นด้วยค่า Median ของค่าความเข้มแสงของพิกเซลที่อยู่ในพื้นที่ Filter นั้น การหาค่า Median ทำได้ดังนี้ ในกรณีที่ Filter Mask ขนาด N x M และ N เป็นเลขคี่ ซึ่งหลังจาก การจัดเรียงค่าของความเข้มแสงของพิกเซล ที่อยู่ใน Filter Mask หรือ window (Sxy) จากค่าความเข้มแสงน้อยไปหามาก แล้วแทนที่ค่าพิกเซลนั้น g(x, y) ด้วยค่ากลางที่ตำแหน่ง จากค่าความเข้มแสงที่ถูกจัดเรียงไว้แล้ว

$$g(x,y) = \underset{(s,t)\in S_{xy}}{median}\{f(s,t)\} \tag{2.12}$$

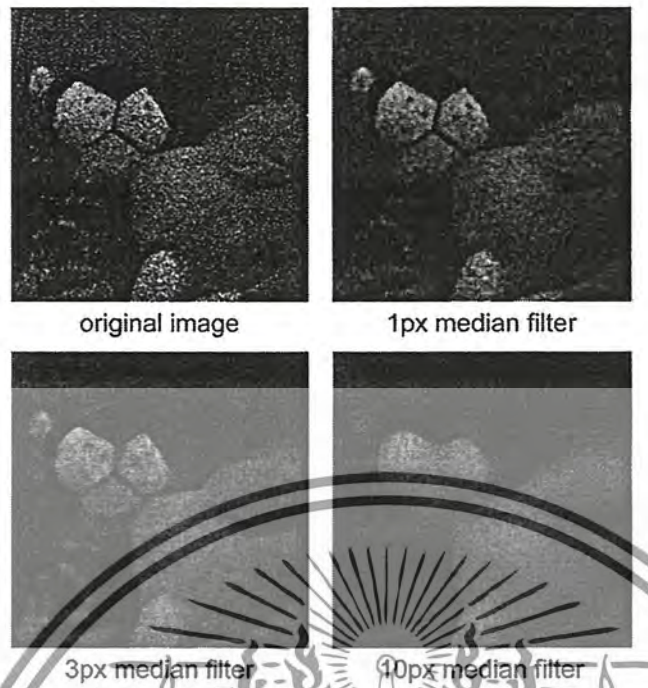
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้เทคนิคนี้มีความสามารถในการลดผลกระทบของ Noise โดยที่ทำให้ภาพที่ได้น้อยกว่า เทคนิคการทำ Mean (Linear) Filtering ที่มีขนาดของ Filter Mask ขนาดเดียวกัน โดยเฉพาะในกรณีของผลกระทบจาก Impulse Noise หรือ Salt and Pepper Noiseความคมชัดของภาพลดลง

พิกเซลที่สนใจ(ก่อนหน้านี้เป็นค่า 97) ถูกแทนที่ด้วยค่ากลางที่หาได้จาก 9 ค่าคือ (4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.4 ตัวอย่างที่ผ่านการกำจัด noise โดย Median Filter

2.5 Edge Detection

การทำ Edge Detection หมายถึงการหาขอบของวัตถุใดๆ ในภาพ โดยสามารถ เขียนกระบวนการได้ดังรูปด้านล่างนี้

รูปที่ 2.5 กระบวนการของ Edge Detection

Edge Detection นั้นมีด้วยกันหลายวิธี แต่อย่างไรก็ตามสามารถแบ่งได้เป็น 2 กลุ่มหลัก คือ Gradient method และ Laplacian method โดยในแต่ละวิธีมีรายละเอียดดังต่อไปนี้

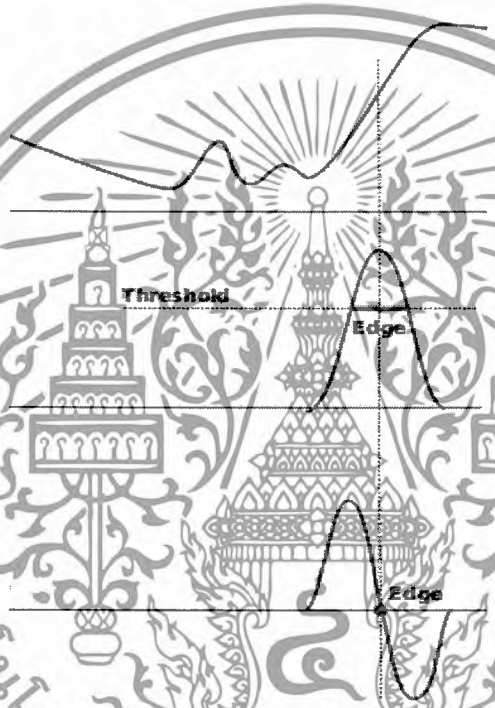
103022
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 Gradient Method

วิธีนี้จะหาขอบโดยการหาจุดต่ำสุดและจุดสูงสุดในรูปของอนุพันธ์อันดับหนึ่งของภาพ โดยจุดที่เป็นขอบจะอยู่ในส่วนที่เหนือค่า threshold (รูปที่ 2.6 b) จึงอาจทำให้เส้นขอบที่ได้มีลักษณะหนา ตัวอย่างวิธีการหาขอบของกลุ่มนี้ เช่น Roberts, Prewitt, Sobel และ Canny เป็นต้น

2.5.2 Laplacian Method

วิธีนี้จะหาขอบโดยใช้อนุพันธ์อันดับ 2 โดยใช้จุดที่ค่า y เป็น 0 (Zerocrossing) (รูปที่ 2.6 c) ซึ่งวิธีนี้จะใช้เวลาในการคำนวณมากกว่า Gradient method ตัวอย่างวิธีการหาขอบของกลุ่มนี้ เช่น Laplacian of Gaussian และ Marrs-Hildreth เป็นต้น



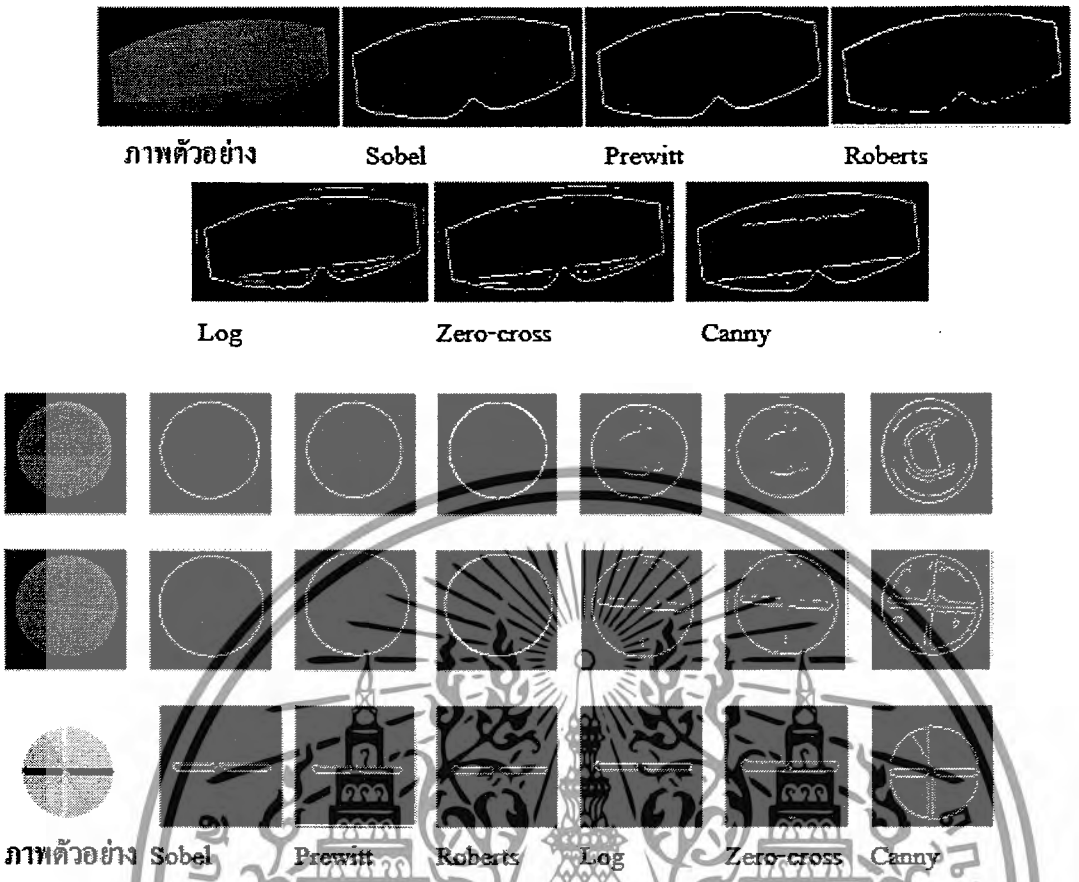
รูปที่ 2.6 กราฟการทำ Edge Detection

(a) ความแตกต่างของระดับความเข้มของสี

(b) กราฟการหาขอบด้วยวิธี Gradient method

(c) กราฟการหาขอบด้วยวิธี Laplacian method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



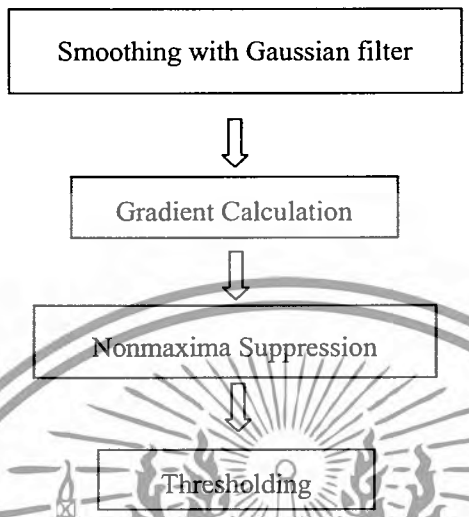
รูปที่ 2.7 ตัวอย่างการหาขอบภาพโดยใช้ Edge Detector แบบต่างๆ

โดยในโครงงานนี้เราได้เลือกใช้วิธี Canny ในการหาขอบภาพ เนื่องจากวิธีดังกล่าวมีการใช้ Gaussian filter ก่อนการหาขอบจึงสามารถควบคุมระดับความละเอียดของขอบที่ต้องการและสามารถลดสัญญาณรบกวนได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 Canny Edge Detection Algorithm

ขั้นตอนการหาขอบโดยวิธีของ Canny ประกอบด้วย 4 ขั้นตอน ดังรูปที่ 2.8



รูปที่ 2.8 ขั้นตอนของ Canny edge detection

การทำงานของ Canny edge detection นั้นเริ่มต้นจากการปรับภาพให้เรียบ (Smoothing) ด้วยตัวกรองเกาเซียน (Gaussian filter) เพื่อกำจัดสัญญาณรบกวน หลังจากนั้น คำนวณค่าขนาด (magnitude) และทิศทาง (orientation) ของ gradient โดยใช้การหาอนุพันธ์อันดับหนึ่ง ในลำดับมาจึงใช้ nonmaxima suppression กับ gradient magnitude เพื่อทำให้ได้ขอบที่บางลง และในขั้นตอนสุดท้ายใช้ double thresholding algorithm เพื่อระบุพิกเซลที่เป็นขอบและช่วยเชื่อมต่อขอบ โดยในแต่ขั้นตอนมีรายละเอียดดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3.1 Smoothing

ในขั้นตอนแรกของการหาขอบโดยอัลกอริทึมนี้จะต้องกำจัดสัญญาณรบกวนออกก่อนโดยใช้ Gaussian filter ซึ่งสามารถคำนวณได้จากการใช้กรอบ (mask) ขนาดเล็ก ขนาดของ Gaussian mask นี้หากมีขนาดกว้างจะมีผลทำให้ลดสัญญาณรบกวนได้มาก แต่ถ้ากว้างมากเกินไปจะมีผลทำให้ขอบย่อยๆ ที่เป็นส่วนรายละเอียดนั้นหายไป สำหรับการคำนวณหาภาพที่ได้จากการใช้ Gaussian filter เป็นดังสมการที่ 2.13

$$S[i, j] = G[i, j, \sigma] * I[i,j] \qquad (2.13)$$

กำหนดให้

I[i, j] เป็นภาพที่ต้องการหาขอบ

G[i, j, σ] เป็น Gaussian smoothing filter

σ เป็น spread of the Gaussian (ควบคุมระดับของการ smoothing)

2.5.3.2 Gradient Calculation

ในขั้นแรกนำ smoothing image S[i, j] มาสร้าง x, y partial derivatives P[i, j] และ Q[i, j] ตามลำดับ ดังสมการที่ 2.14 และ 2.15

$$P[i, j] \approx (S[i, j+1] - S[i, j] + S[i+1, j+1] - S[i+1, j]) / 2 \qquad (2.14)$$

$$Q[i, j] \approx (S[i, j] - S[i+1, j] + S[i, j+1] - S[i+1, j+1]) / 2 \qquad (2.15)$$

หลังจากนั้นนำค่า x , y partial derivatives มาคำนวณด้วยสูตรมาตรฐานสำหรับการแปลงรูปแบบจาก rectangular ไปเป็น polar (rectangular-to-polar conversion) เพื่อหาขนาดและทิศทางของ gradient ตามสมการที่ 2.16

$$M[i, j] = \sqrt{P[i, j]^2 + Q[i, j]^2}$$

$$\theta[i, j] = \arctan(Q[i, j], P[i, j]) \qquad (2.16)$$

จากสมการข้างต้นจะสามารถหาค่ามุม θ ออกมาได้เมื่อแทนค่าตัวแปรในฟังก์ชัน arctan (x , y)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3.3 Nonmaxima Suppression

สำหรับการหาขอบโดย Canny method จุดที่ถือเป็นเส้นขอบได้นั้นต้องเป็นจุดที่ให้ค่าสูงสุดเฉพาะที่และเป็นทิศทางเดียวกับ gradient ด้วย ซึ่งด้วยวิธีดังกล่าวนี้ทำให้ได้ขอบที่บางเพียง 1 พิกเซล ภาพที่ได้หลังการทำ Nonmaxima Suppression จะให้ค่าเป็นศูนย์ในทุกจุดยกเว้นจุดที่เป็น local maxima points ซึ่งจะยังคงค่าเดิมไว้

2.5.3.4 Thresholding

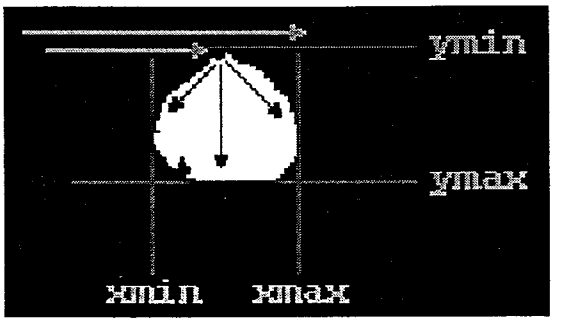
แม้ว่าภาพจะผ่านการ smoothing ในขั้นตอนแรกแล้วก็ตาม ภาพที่ได้อาจยังมีเส้นขอบที่ไม่ใช่ขอบที่แท้จริงปรากฏอยู่อันเนื่องจากสัญญาณรบกวนหรือลักษณะของวัตถุในภาพเป็นพื้นผิวที่มีลวดลายหรือมีรายละเอียดภายในมาก ดังนั้นเพื่อลดปัญหาดังกล่าวจึงได้มีการกำหนดค่า threshold ขึ้นมา 2 ค่า คือ high threshold (T1) และ low threshold (T2) โดยพิกเซลที่มีค่ามากกว่า T1 จะถูกปรับเป็น 1 (เป็นพิกเซลที่เป็นขอบ) แต่ถ้าน้อยกว่า T2 จะถูกปรับเป็น 0 ส่วนค่าที่อยู่ระหว่างค่า threshold ทั้งสอง การปรับเป็นค่า 0 หรือ 1 นั้นขึ้นอยู่กับพิกเซลที่อยู่รอบข้าง หากพบว่าพิกเซลที่อยู่รอบข้างของพิกเซลที่เป็นขอบ (ค่า >T1) มีค่ามากกว่า T2 แล้ว จะปรับค่าพิกเซลดังกล่าวให้มีค่าเป็น 1 และถือเป็นสมาชิกหนึ่งในภาพขอบด้วยเช่นกัน

2.6 Blob Analysis

Blob Analysis เป็นการประมวลผลหาตำแหน่งของ Blob (กลุ่มของพิกเซล) และข้อมูลอื่นๆ โดยใช้วิธีการ Floodfill ซึ่งจะทำการไล่ทีละจุดสีจนกว่าจะเจอจุดสีขาว โดยจะทำไปเรื่อยๆจนครบทุกจุด ในระหว่างที่วนนั้นก็จะทำการนับจำนวนจุดและหาขอบเขตบนและล่างเพื่อให้สามารถหาจุดกึ่งกลางได้ในภายหลัง

เนื่องจากในกระบวนการทำงานทางด้าน image processing ในบางครั้งจะมีการทำ Threshold ด้วย ซึ่งมักจะมีจุดที่เป็นจุดรบกวนเข้ามา โดยขึ้นกับช่วงสีที่ระบุว่ามีความเหมาะสมมากน้อยเพียงใด ทำให้การ Floodfill หาตำแหน่ง ยังไม่สามารถจำแนกจุดที่ต้องการออกมาได้โดยตรง จึงต้องมีการใช้ข้อมูลอื่นเข้ามาประกอบ เช่น ขนาดของ Blob โดยอาจพิจารณาจุดที่มีขนาดเล็กว่าเป็น Noise และจุดที่มีขนาดใหญ่เป็นจุดที่เราสนใจ ก็จะทำให้สามารถหาตำแหน่งของวัตถุที่เราต้องการออกมาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 การทำงานของ Blob Analysis โดยใช้วิธี Floodfill

ฟังก์ชันการทำงานของ Blob Analysis มีขั้นตอนดังต่อไปนี้

1. ตรวจสอบพิกเซลไปเรื่อยๆ เมื่อเจอพิกเซลสีขาวที่ยังไม่ได้ตรวจสอบมาก่อน ให้ไปทำข้อ 2
2. แสดงว่าพิกเซลนี้เป็นพิกเซลสีขาวของจุดสี (Blob) ใหม่
3. ตรวจสอบพิกเซล
4. ตรวจสอบพิกเซลข้างเคียงทั้ง 8 หากพิกเซลใดเป็นพิกเซลสีขาวที่ยังไม่ได้ผ่านมาก่อน ให้วนไปตรวจสอบทุกพิกเซลจนไม่มีพิกเซลสีขาวอีกเป็นอันหมดจุดสีนั้น
5. กลับไปทำข้อ 1 จนครบทุกพิกเซลบนภาพ

หลังจากได้ทำ Floodfill จนครบแล้วเราก็จะได้ขอบเขตของ Blob นั้นออกมาด้วย ซึ่งทำให้สามารถหาตำแหน่งกลางได้จากสมการ 2.17

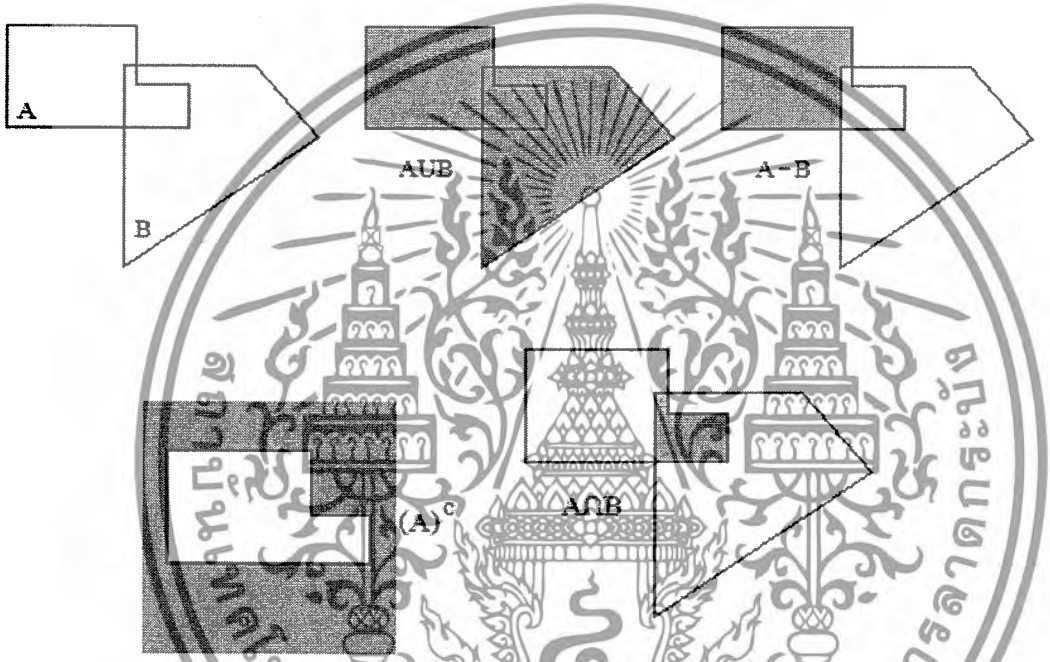
$$x = \frac{x_{max} + x_{min}}{2} \qquad y = \frac{y_{max} + y_{min}}{2} \tag{2.17}$$

โดย x_{max}, x_{min}, y_{max}, y_{min} คือ ค่าขอบเขตทั้ง 4 ด้านของ Blob

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 Morphological Image Processing

การประมวลผลภาพเกี่ยวกับ Morphology เป็นทฤษฎีที่เกี่ยวข้องกับคณิตศาสตร์ในเรื่องของ Set ดังนั้นการคำนวณพื้นฐานส่วนใหญ่จึงเป็นเรื่องของ Set ซึ่งใช้ช่วยในการอ้างอิงส่วนประกอบต่างๆ ของวัตถุที่กำลังสนใจอยู่ในภาพ ในส่วนนี้จะกระทำกับภาพที่เป็นระดับสีเทา ผลที่ได้จะเป็นการปรับปรุงวัตถุที่กำลังสนใจอยู่ให้มีลักษณะต่างไปจากเดิม โดยมีพื้นฐานการคำนวณจากเรื่องของ Set ดังนี้



รูปที่ 2.10 การคำนวณพื้นฐานของ Set

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

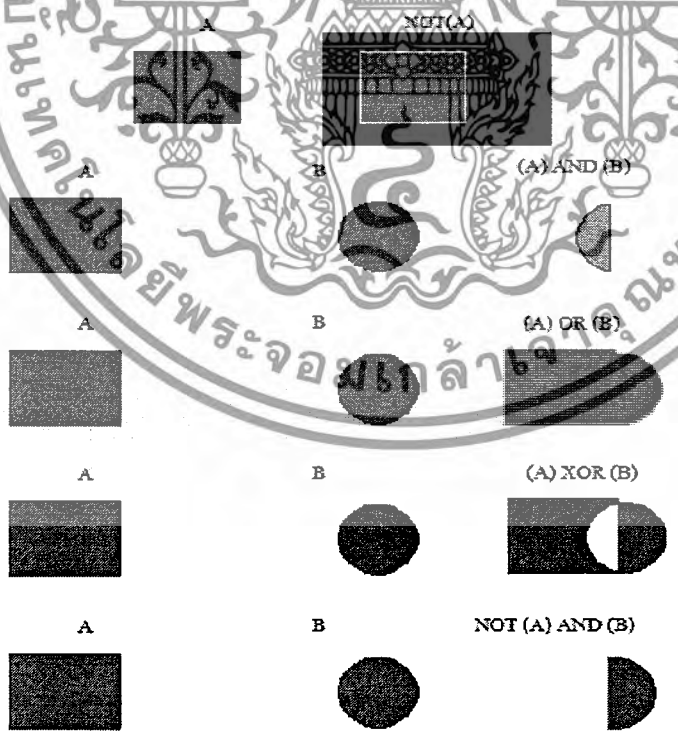
จากการคำนวณพื้นฐานของ Set สามารถนำมาประยุกต์กับการคำนวณอื่นๆ ทำให้ได้ฟังก์ชันการทำงานที่สามารถนำมาให้กับรูปภาพได้ ดังนี้

2.7.1. Logic Operation

การคำนวณทางด้านตรรกศาสตร์มีลักษณะการทำงานคล้ายๆ กับเรื่องของ Set ดังนั้นผลของการคำนวณจึงมีลักษณะเหมือนกันกับการทำงานของ Set แต่การทำงานแบบตรรกศาสตร์นี้จะเป็นการทำงานกับภาพไบนารีเท่านั้น เนื่องจากเป็นการทำงานทางด้านตรรกศาสตร์ผลลัพธ์ที่ได้จึงมีแค่สองคำตอบคือ 0 และ 1 ตามตารางที่ 2.1

ตารางที่ 2.1 การคำนวณทางด้านตรรกศาสตร์

p	q	p AND q	p OR q	NOT(p)
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0



รูปที่ 2.11 การทำ Logic Operation ของภาพ A และ B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2. Dilation และ Erosion

การทำ Dilation และ Erosion เป็นฟังก์ชันพื้นฐานของการประมวลผลภาพ Morphological กระบวนการ Dilation เป็นการขยายขนาดของวัตถุที่สนใจให้มีขนาดใหญ่ขึ้น โดยการนำเอาวัตถุที่มีขนาดเล็กกว่ามาช่วย โดยนำเอาจุดศูนย์กลางของวัตถุอันที่เล็กกว่ากระทำตามขอบของภาพโดยส่วนที่เกินออกมาจะถูกนำไปรวมกับเนื้อที่ของวัตถุที่สนใจ ทำให้วัตถุที่กำลังสนใจมีขนาดใหญ่ขึ้น และ Erosion ก็เป็นกระบวนการที่ทำย้อนกับ Dilation เป็นการทำให้วัตถุมีขนาดเล็กลง สามารถเขียนเป็นฟังก์ชันคณิตศาสตร์ในรูปของ Set ได้ดังนี้

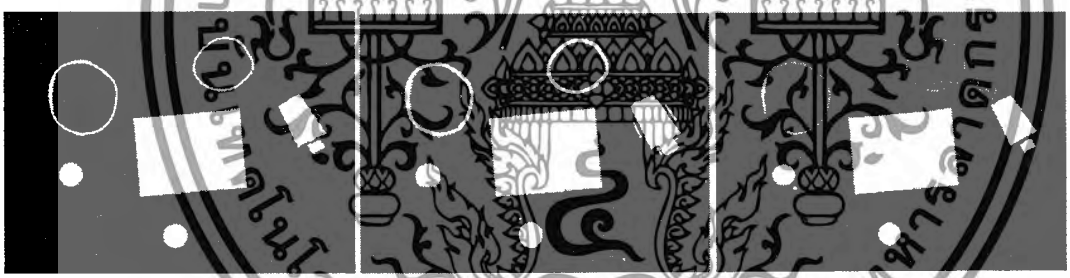
Dilation

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \varnothing\} \tag{2.18}$$

Erosion

$$A \ominus B = \{z \mid (B)_z \subseteq A\} \tag{2.19}$$

โดยที่ A,B เป็น Set ที่อยู่ใน Z^2



(a) (b) (c)

รูปที่ 2.12 การ Dilation และ Erosion (a) ภาพต้นฉบับ (b) ภาพ Dilation (c) ภาพ Erosion

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.3 Opening และ Closing

การกระทำกับภาพด้วยวิธี Opening และ Closing เป็นอีกวิธีการหนึ่งที่นำเอา Dilation และ Erosion เข้ามาประยุกต์ใช้งานมากขึ้น ซึ่งวิธีการ Opening จะเป็นการทำให้ขอบของวัตถุไม่มีมุมแหลม หรือเป็นการลดความคมของขอบของภาพ โดยการนำวัตถุที่มีขนาดเล็กกว่า และเป็นวงกลมเข้าไปกลิ้งตามขอบภายในวัตถุที่สนใจ และเมื่อขอบของวงกลมชนกับเส้นสองเส้น พื้นที่บริเวณที่ถูกปิดล้อม จะถูกตัดทิ้งไป ทำให้มุมของภาพกลายเป็นส่วนโค้งของวงกลม และการ Closing เป็นอีกวิธีซึ่งช่วยทำให้ บางส่วนของขอบภาพไม่มีมุมแหลม โดยการนำวัตถุที่มีขนาดเล็กกว่า และเป็นวงกลม กลิ้งตามขอบภายนอกของภาพ และเมื่อขอบของวงกลมชนกับเส้นตรงสองเส้น พื้นที่ที่ถูกปิดล้อม จะนำมารวมเข้าด้วยกัน สามารถเขียนเป็นฟังก์ชันคณิตศาสตร์ในรูปของเซ็ต ได้ดังนี้

Opening

$$A \circ B = (A \ominus B) \oplus B \tag{2.20}$$

Closing

$$A \bullet B = (A \oplus B) \ominus B \tag{2.21}$$

รูปที่ 2.13 การ Opening และ Closing (a) ภาพต้นฉบับ (b) ภาพ Opening (c) ภาพ Closing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 การบีบอัดข้อมูลวิดีโอ

ไฟล์วิดีโอมีลักษณะการส่งสัญญาณเป็นภาพเคลื่อนไหวและเสียง โดยสัญญาณภาพที่ส่งมีลักษณะเป็น เฟรม (หนึ่งเฟรมเท่ากับหนึ่งภาพ) ซึ่งถ้าจะให้เห็นภาพต่อเนื่อง ในหนึ่งวินาที จะต้องใช้ภาพไม่ต่ำกว่า 17 เฟรม ดังนั้นถ้าไฟล์วิดีโอที่เราต้องการจัดเก็บมีความยาวมากๆ ก็จะต้องใช้พื้นที่ในการจัดเก็บมากขึ้นไปด้วย ดังนั้นจึงจำเป็นต้องมีเทคนิคในการบีบอัดข้อมูลเพื่อให้ไฟล์มีขนาดเล็กลงโดยยังสามารถรักษาความถูกต้องของไฟล์ไว้ได้ ในที่นี้จะขอกล่าวเพียงแค่บางกรรมวิธีที่นิยมใช้ในปัจจุบัน

2.8.1 MPEG-4

MPEG-4 เป็นมาตรฐานที่ใช้ในการบีบอัดข้อมูลดิจิตอลของภาพและเสียง ถูกพัฒนาขึ้นและมุ่งหวังให้เป็นรูปแบบมาตรฐานไฟล์วิดีโอในอนาคตของ ISO/IEC Moving Pictures Expert Group (MPEG) ภายใต้มาตรฐาน ISO/IEC 14496 โดยผู้พัฒนา MPEG-4 2 รายแรกๆคือ Microsoft (.asf) และ DivX (.avi) สำหรับ DivX ในยุคแรกๆจะเป็นเวอร์ชั่นที่แฮกค์ .asf ให้เก็บเป็น .avi แต่ในภายหลัง DivX ได้แก้ไขให้เป็นอัลกอริทึมของตนเอง ในปัจจุบัน DivX ได้รับความนิยมสูงมาก โดยเฉพาะกลุ่มคนที่นิยมนำภาพยนตร์ดีวีดีไปเข้ารหัสใหม่(rip) เพื่อสำรองข้อมูลไว้เพราะได้คุณภาพเท่าดีวีดีในขนาดเท่าแผ่นซีดี

การทำงานของ MPEG-4 เป็นการบีบอัดแบบ Lossy Compression ใช้หลักการเข้ารหัสกราฟฟิกและวีดีโอในแบบอัลกอริทึมที่ได้รับการพัฒนามาจาก MPEG-1, MPEG-2 และเทคโนโลยีของ Apple QuickTime โดยไฟล์ที่ได้รับการบีบอัดในรูปแบบ Wavelet – based MPEG-4 จะมีขนาดเล็ก ซึ่งเป็นผลมาจากการบีบอัดเป็นเฟรมๆแทนที่จะบันทึกภาพในทุกเฟรมซึ่งต้องใช้เนื้อที่ในการเก็บมาก การลดขนาดช่วงกว้างของแบนวิธ และรวมเอาไฟล์วีดีโอกับข้อความกราฟฟิกเข้าไว้ด้วยกัน นอกจากนี้ยังรวมเอาแอนิเมชัน 2และ 3 มิติไว้ด้วย เนื่องจากการบีบอัดแบบ Lossy Compression ทำให้ไม่สามารถแปลงไฟล์กลับไปเป็นไฟล์ต้นฉบับได้เหมือนเดิมทั้งหมด และอาจมีข้อมูลสูญเสียบางส่วนจากการบีบอัด แต่เป็นส่วนน้อยมากโดยที่ตาเรามองไม่เห็นและไม่สามารถสังเกตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การบีบอัดข้อมูลแบบ MPEG-4 ถูกนำไปใช้งานในหลายด้าน เช่น โทรศัพท์มือถือหรือเว็บไซต์ ในการนำเสนอภาพวีดีโอบนอินเทอร์เน็ต(Video Streaming) เพื่อให้ผู้ใช้สามารถเก็บภาพต่างๆ ซึ่งเมื่อต้องการความละเอียดมากขึ้นขนาดไฟล์จะใหญ่ขึ้นตาม โดยปัจจุบันขนาดภาพมีอยู่หลายขนาดขึ้นอยู่กับมาตรฐานหน้าจอโทรศัพท์มือถือ โดยถูกพัฒนาขึ้นเพื่อใช้เป็นมาตรฐานในการส่งภาพเคลื่อนไหวและเสียงในระบบดิจิตอลผ่านเครือข่ายความเร็วต่ำ ไฟล์ที่บีบอัดด้วยมาตรฐานนี้รองรับการใช้งานผ่านเครือข่ายที่มีความเร็วหลายระดับตั้งแต่ Modem 56 K,GPRS และ Hi-Speed Internet

2.8.2 Xvid

Xvid คือ Codec (COder/DECoder) ประเภทหนึ่งที่เกิดจากกลุ่มนักพัฒนาอิสระ(GNU GPL) ที่พัฒนารูปแบบการบีบอัดบนพื้นฐานของ MPEG-4 โดยเฉพาะ MPEG-4 Part 2 (MPEG-4 Advanced Simple Profile) ซึ่งเหมือนกับ DivX แต่ Xvid เป็น Open Source ทำให้ปัจจุบัน Xvid เป็น Codec ในกลุ่ม MPEG4 Part 2 (MPEG-4 Advanced Simple Profile) ที่ถูกนิยมใช้อย่างมาก

การทำงานของ Xvid คล้ายกับการ ZIP ไฟล์วีดีโอ แต่ต่างกันที่เมื่อแปลงไฟล์กลับแล้วจะไม่เหมือนไฟล์ต้นฉบับทั้งหมด เพราะเป็นการบีบอัดแบบ Lossy Compression โดยจะตัดข้อมูลบางส่วนที่มนุษย์ไม่สามารถรับรู้ได้ออกไป ทำให้สามารถบีบอัดไฟล์ได้ในอัตราที่สูงถึง 200 เท่า (200 : 1) เช่น ไฟล์วีดีโอความยาว 1 ชั่วโมงมีความละเอียดเท่าระบบ PAL หากไม่ได้ถูกบีบอัด (Uncompress) จะมีขนาดถึง 100 GB แต่เมื่อผ่านการบีบอัดด้วย Xvid แล้วจะมีขนาดเพียง 500 MB เท่านั้น และเนื่องจากว่ามาตรฐานการบีบอัดของ Xvid ใช้เป็นแบบ ASP (MPEG-4 Advanced Simple Profile) Xvid จึงสามารถเล่นบนโปรแกรมหรือเครื่องเล่น DVD ที่สามารถเล่นไฟล์ MP4 หรือ DivX ได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 OpenCV Library

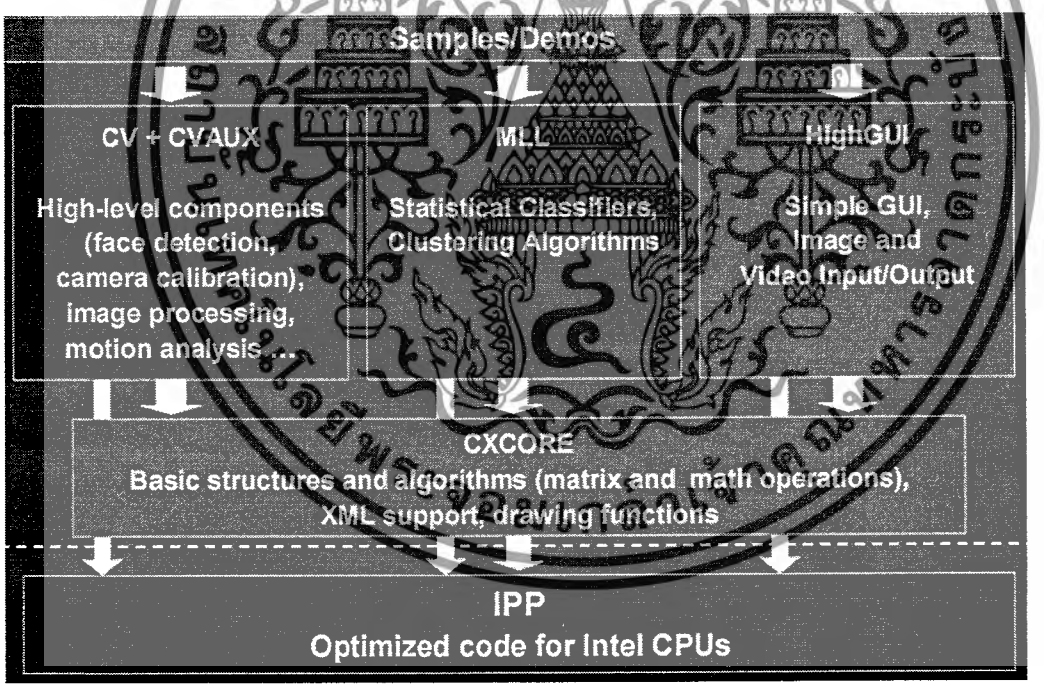
OpenCV เป็น Library ที่สร้างขึ้นเพื่อให้การพัฒนาโปรแกรมทางด้าน การมองเห็นของคอมพิวเตอร์ (Computer Vision) เป็นไปได้อย่างสะดวก OpenCV นี้พัฒนาขึ้นโดยได้รับการสนับสนุนจาก Intel Corporation ให้เป็นซอฟต์แวร์แบบเปิดเผยรหัส (Open Source) เพื่อให้สามารถนำไปต่อยอดพัฒนาโปรแกรมต่าง ๆ ได้ง่าย สามารถใช้ได้บนระบบปฏิบัติการทั้งที่เป็น Linux และ Microsoft Windows

OpenCV ประกอบด้วย Data Structure และ Algorithm

1.) Data Structure ใช้เก็บข้อมูลต่าง ๆ อาทิ เช่น รูปภาพ เมทริกซ์ พิกัด

2.) Algorithm เพื่อการประมวลผลต่าง โดยเฉพาะการประมวลผลทางรูปภาพ

ข้อจำกัดของ OpenCV คือ สามารถใช้งานได้เฉพาะเครื่องคอมพิวเตอร์ที่ใช้หน่วยประมวลผล (CPU) จาก Intel Corporation แต่ข้อจำกัดนี้ทำให้เกิดจุดเด่นเช่นกัน กล่าวคือ การประมวลผลต่าง ๆ จะใช้ความสามารถของหน่วยประมวลผลอย่างเต็มที่ ทำให้โปรแกรมที่พัฒนาโดยการใช้ OpenCV นี้มีประสิทธิภาพในการประมวลผลที่สูงมาก

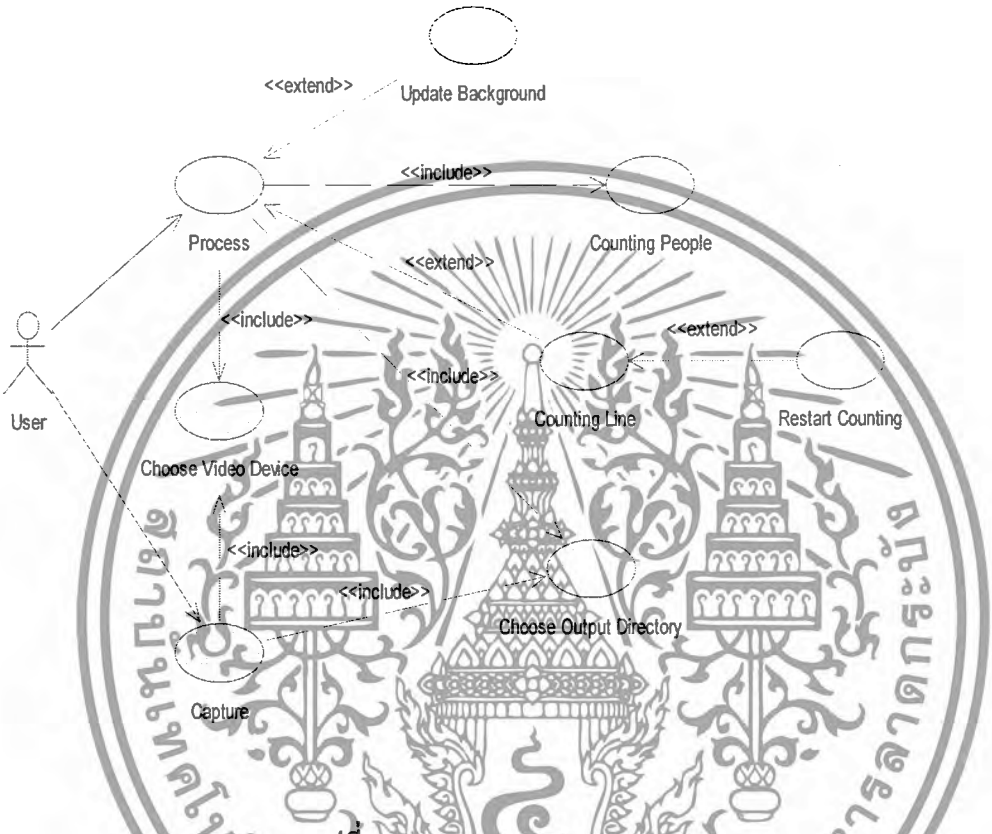


รูปที่ 2.14 สถาปัตยกรรมของ OpenCV Library

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและพัฒนา

3.1 Use Case Diagram

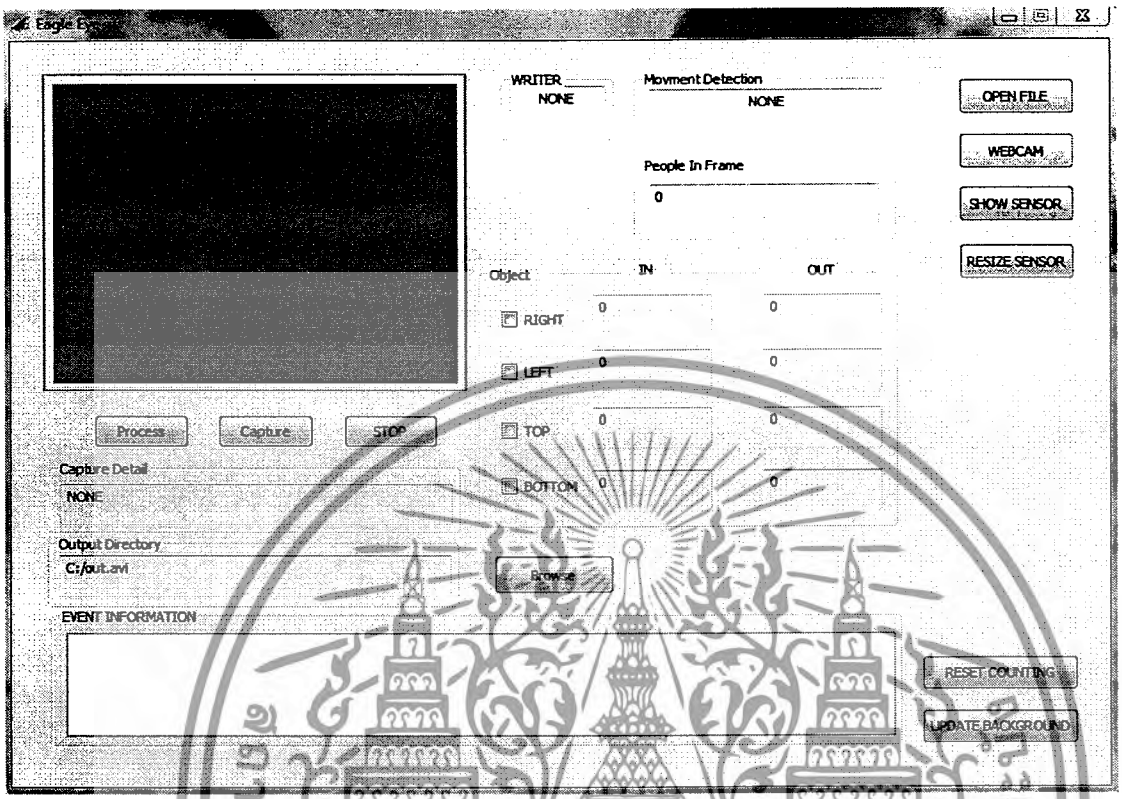


รูปที่ 3.1 use case diagram

จากรูปที่ 3.1 แสดงฟังก์ชันการทำงานในรูปแบบ use case diagram จากภาพจะเห็นว่าผู้ใช้สามารถเลือกการทำงานของโปรแกรมหลักๆได้สองส่วนคือส่วนการ process และ capture ซึ่งถ้าผู้ใช้เลือกส่วน process ผู้ใช้จะสามารถเลือกต่อไปได้อีกว่าต้องการอินพุธประเภทไหน จะเก็บผลลัพธ์ไว้ที่ใด รวมถึงเลือกว่าให้มีการนับวัตถุบนขอบเฟรมหรือไม่ สั่งเริ่มการนับใหม่ สั่งการอัพเดตพื้นหลังด้วยตัวเอง แต่ถ้าผู้ใช้เลือกส่วน capture ผู้ใช้จะสามารถเลือกได้แค่จะใช้อินพุทประเภทไหน และจะเก็บไฟล์ผลลัพธ์ที่ใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบ GUI



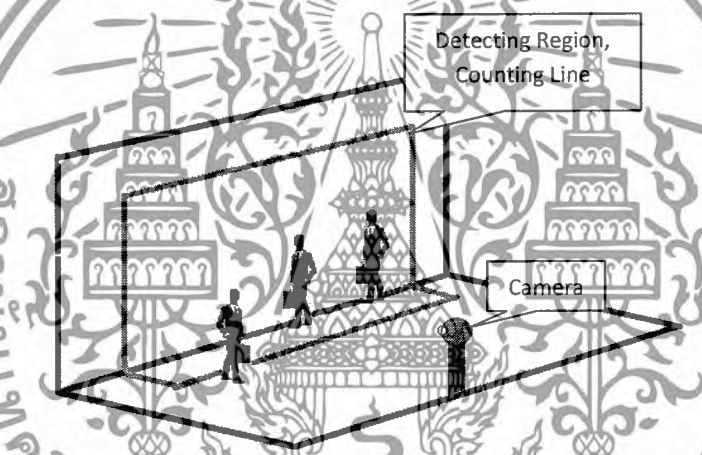
รูปที่ 3.2 หน้าต่าง GUI ของโปรแกรม

จากรูปภาพประกอบเป็น GUI ของโปรแกรม โดยส่วนประกอบของ GUI เป็นดังนี้

1. Detecting Area คือ ส่วนหนึ่งของเฟรมที่ใช้สำหรับการ track object โดยจากรูป Detecting Area คือพื้นที่ทั้งหมดบนเฟรมที่กล้องจับได้
2. Counting Line คือ เส้นสำหรับนับวัตถุที่ผ่านเข้าหรือออกจากเฟรม จากรูปกำหนดให้ขอบของเฟรมทั้งสี่ด้านคือ Counting Line
3. Direction Line คือ เส้นสำหรับพิจารณาว่าวัตถุที่เข้ามาในเฟรมมีทิศทางเข้าหรือออกจากเฟรม
4. Writer คือ ส่วนแสดงสถานะของการบันทึกไฟล์วิดีโอ
5. Movement Detection คือ ส่วนแสดงสถานะการตรวจจับการเคลื่อนไหว
6. People In Frame คือ ส่วนแสดงจำนวนบุคคลที่เฟรมปัจจุบัน
7. Object คือ ส่วนแสดงจำนวนวัตถุที่ผ่านเข้า – ออก
8. Capture Detail คือ ส่วนแสดงรายละเอียดของไฟล์ผลลัพธ์
9. Output Directory คือ ส่วนแสดงไดเรกทอรีและชื่อไฟล์ในการบันทึกไฟล์ผลลัพธ์
10. Even Information คือ ส่วนแสดงสถานการณ์ทำงานต่างๆของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. ปุ่ม Capture คือ ปุ่มสั่งอัดวิดีโอโดยไม่มีการ process
12. ปุ่ม Process คือ ปุ่มสั่งอัดวิดีโอแบบมีการ process
13. ปุ่ม Stop คือ ปุ่มสั่งหยุดการ process และ capture
14. ปุ่ม Open file คือ ปุ่มสำหรับเลือกไฟล์วิดีโอสำหรับการประมวลผล
15. ปุ่ม Webcam คือ ปุ่มสำหรับเลือกการประมวลผลผ่าน Webcam
16. ปุ่ม Show/Hide Sensor คือ ปุ่มสำหรับเลือกแสดงหรือไม่แสดงเส้นตรวจจับ
17. ปุ่ม Resize Sensor คือ ปุ่มสำหรับปรับขนาดเส้นตรวจจับ
18. ปุ่ม Browse คือ ปุ่มสำหรับเลือกไดเรกทอรีในการบันทึกไฟล์ผลลัพธ์
19. ปุ่ม Reset Counting คือ ปุ่มสำหรับเริ่มนับค่าต่างๆใหม่
20. ปุ่ม Update Background คือ ปุ่มสำหรับการ Update Background ด้วยตนเอง



รูปที่ 3.3 ตัวอย่างสถานที่ในการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การเขียนโปรแกรมเพื่อใช้งาน OpenCV Library

1. การ Include library OpenCV เพื่อให้โปรแกรมสามารถเรียกใช้งานได้

```
#include <highgui.h>
#include <cv.h>
#include <cxcore.h>
```

2. การเขียนโปรแกรมติดต่อกับกล้อง

Libraly ของ OpenCV อำนวยความสะดวกในการเขียนโปรแกรมติดต่อกับอุปกรณ์รับภาพที่สามารถติดต่อกับคอมพิวเตอร์ได้เช่น กล้องวิดีโอหรือ กล้องเวปแคม

```
CvCapture* capture;
capture = cvCaptureFromCAM(0);
```

3. การเขียนโปรแกรมติดต่อกับไฟล์วิดีโอ AVI

Libraly ของ OpenCV อำนวยความสะดวกในการเขียนโปรแกรมติดต่อกับ ไฟล์เพื่อประมวลผลไฟล์ดังกล่าวได้โดยไฟล์ที่ OpenCV สนับสนุนการทำงานได้แก่ ไฟล์รูปภาพทุกฟอร์แมต และไฟล์วิดีโอประเภท AVI

```
CvCapture* capture;
capture = cvCaptureFromAVI(cc);
```

4. การเขียนโปรแกรมเพื่อสร้าง writer เพื่อบันทึกเฟรมที่จับได้ลงไฟล์วิดีโอ

```
CvVideoWriter *writer = 0;
writer=cvCreateVideoWriter(zz,CV_FOURCC('X', 'V', 'I', 'D') ,
                fps,cvSize(frameW,frameH),1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. การเขียนโปรแกรมเพื่อประมวลผลไฟล์วิดีโอ

หลักการคือการดึงเฟรมของวิดีโอที่ละเฟรมออกมาประมวลผลตามลำดับโดยสามารถกำหนดความเร็วในการดึงเฟรมออกมาได้

```
IplImage* img = 0;
int nFrames = 50;
for(i=0;i<nFrames;i++){
  cvGrabFrame(capture);          // capture a frame
  img=cvRetrieveFrame(capture);  // retrieve the captured frame
  key=cvWaitKey(20);             // wait 20 ms
    }
```

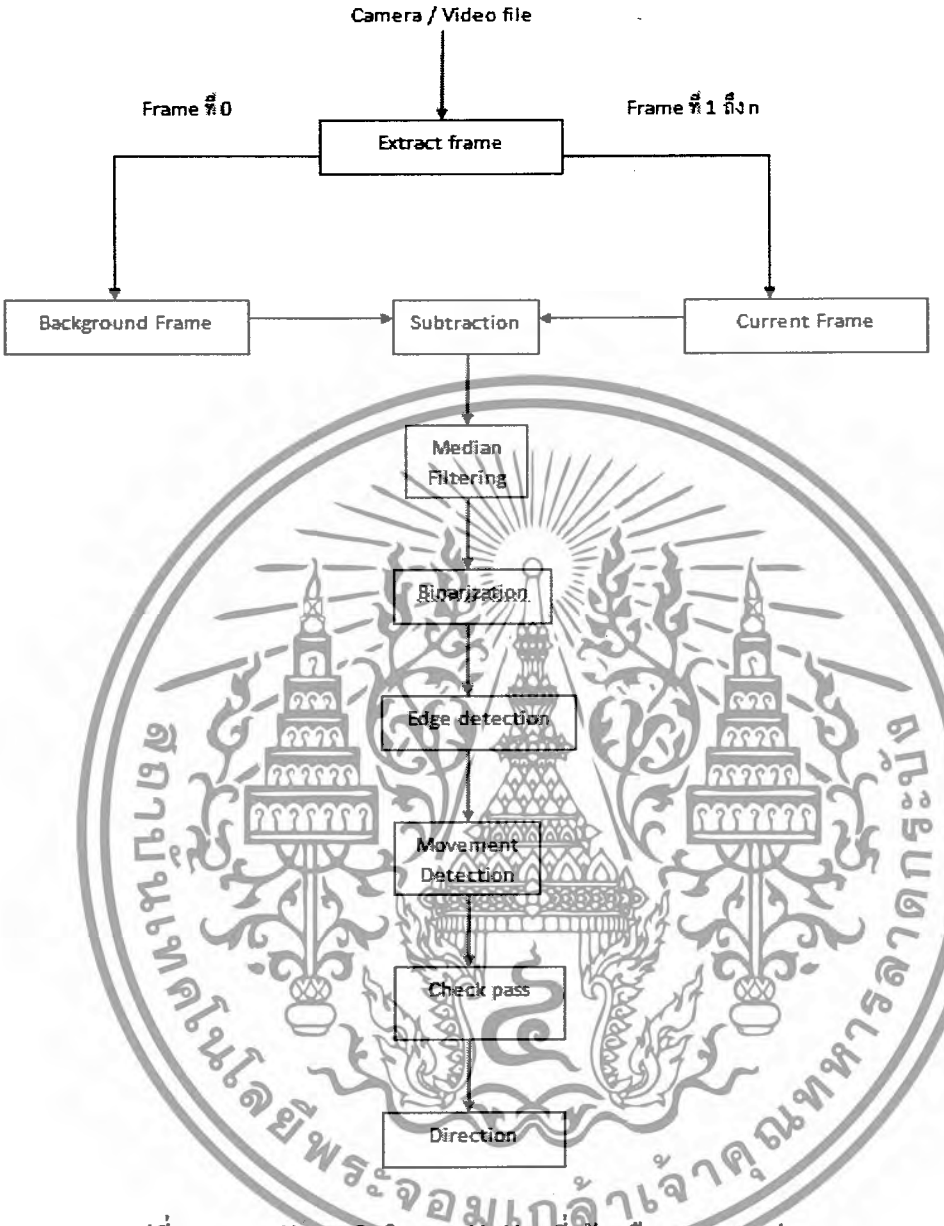
6. การยกเลิกการติดต่อกับกล้องและ writer

```
cvReleaseVideoWriter(&writer);

cvReleaseCapture(&capture);
```

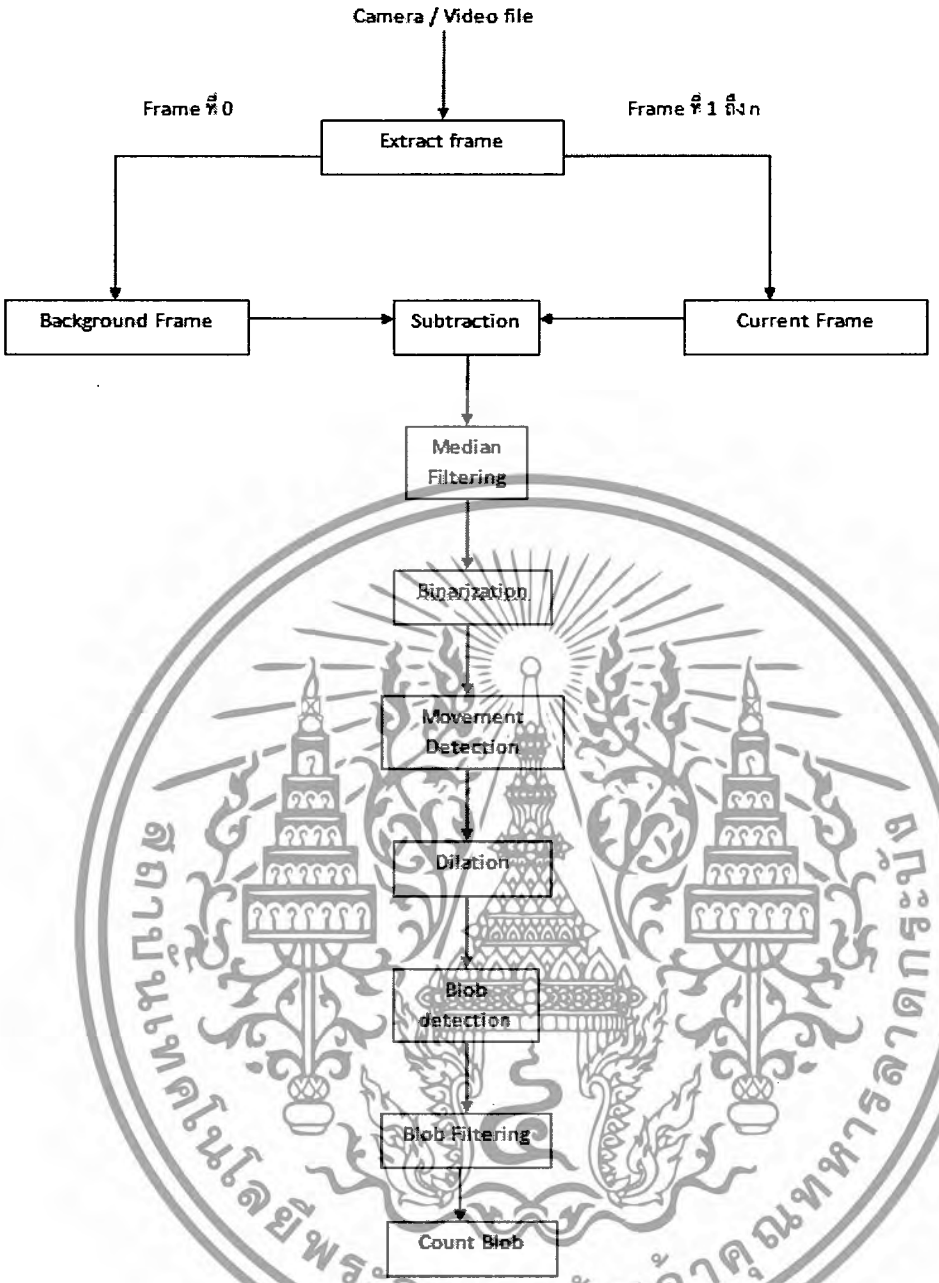
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 เทคนิคต่างๆ ที่ใช้ในการประมวลผลภาพ



รูปที่ 3.4 แผนผังเทคนิคในการนับวัตถุที่เข้าหรือออกจากเฟรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แผนผังเทคนิคในการนับคนที่อยู่ในเฟรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 การกำหนดจุดอ้างอิง

การที่จะสามารถระบุได้ว่ามีการเคลื่อนไหวเกิดขึ้นในเฟรมหรือไม่นั้น จำเป็นที่จะต้องมีจุดที่สามารถอ้างอิงหรือเปรียบเทียบ เพื่อให้สามารถแสดงผลของการเปรียบเทียบและบอกได้ว่ามีการเคลื่อนไหวเกิดขึ้นหรือไม่ ดังนั้นจากการศึกษาพบว่ามีสองวิธีที่น่าสนใจในการเลือกจุดที่อ้างอิงดังกล่าวคือ

3.4.1.1 การกำหนดให้พื้นหลังของเฟรมเป็นจุดอ้างอิง

กระบวนการการทำงานของวิธีนี้คือการเก็บเฟรมที่ระบุได้ว่าไม่มีการเคลื่อนไหวเป็นจุดอ้างอิง เพื่อนำมาเปรียบเทียบกับเฟรมปัจจุบันซึ่งผลของการเปรียบเทียบจะแสดงสิ่งที่แตกต่างจากเฟรมที่เป็นจุดอ้างอิงออกมาก็ต่อเมื่อมีการเคลื่อนไหวเกิดขึ้นบนเฟรมปัจจุบัน

3.4.1.2 การกำหนดให้เฟรมก่อนหน้าเป็นเฟรมอ้างอิง

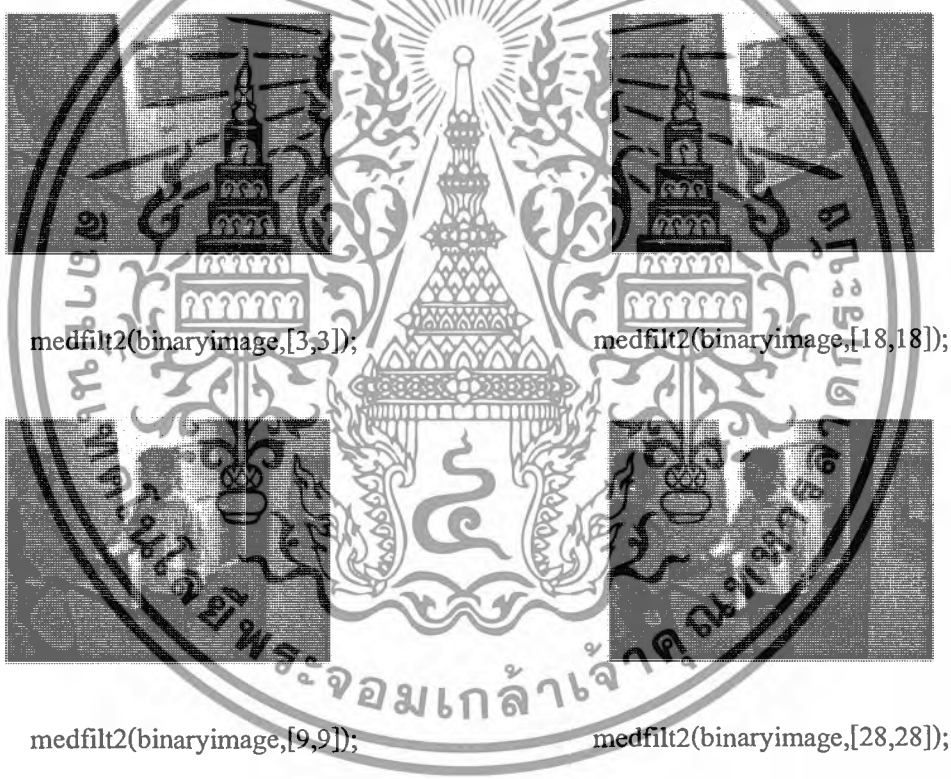
กระบวนการการทำงานของวิธีนี้คือการนำเฟรมปัจจุบันมาเปรียบเทียบกับเฟรมก่อนหน้าหนึ่งเฟรมเพื่อระบุถึงความแตกต่าง โดยการเปรียบเทียบนี้จะเปรียบเทียบไปเรื่อยๆจนถึงเฟรมสุดท้าย ซึ่งผลของการเปรียบเทียบจะแสดงถึงสิ่งแตกต่างที่เกิดขึ้นแบบเฟรมต่อเฟรม

จากการศึกษาถึงข้อดีข้อเสียเพื่อเป็นประโยชน์ในการประกอบการตัดสินใจในการเลือกวิธีข้างต้นมาประยุกต์ใช้กับโครงงานพบว่า เมื่อพิจารณาถึงวัตถุประสงค์ของโครงงานที่ต้องการระบุการเคลื่อนไหวที่เกิดขึ้นเพื่อประยุกต์ใช้กับกล้องวงจรปิด จากการทดลองพบว่าวิธีที่หนึ่งจะสามารถระบุความเคลื่อนไหวที่เกิดขึ้นได้ดีกว่าวิธีที่สองเนื่องจากวิธีที่สองนั้นไม่สามารถระบุการเคลื่อนไหวได้ถูกต้องในกรณีที่วัตถุมีการหยุดการเคลื่อนไหวภายในเฟรม ซึ่งตามวัตถุประสงค์ของโครงงานกำหนดว่ากรณีดังกล่าวนั้นถือว่าเป็นการเคลื่อนไหวเช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 การตรวจจับการเคลื่อนไหว (Movement Detection)

การที่จะสามารถระบุที่ได้ว่ามีการเคลื่อนไหวเกิดขึ้นในเฟรมปัจจุบันหรือไม่ จะใช้การเปรียบเทียบกับจุดอ้างอิงนั้นไม่เพียงพอด้วยเหตุผลที่ว่าในบางเหตุการณ์ที่ผลของการเปรียบเทียบแสดงถึงความแตกต่างที่เกิดขึ้น แต่เราไม่จัดว่าเป็นการเคลื่อนไหว เช่นในกรณีที่ เกิดการเปลี่ยนแปลงของระดับแสงภายในเฟรม ซึ่งเป็นเรื่องยากในการจัดการกับปัญหาดังกล่าวและสามารถจัดได้ว่ากรณีดังกล่าวนั้นคือสัญญาณรบกวน (noise) ดังนั้นจึงมีความจำเป็นที่จะต้องศึกษาถึงเทคนิคในการกำจัด noise เพื่อที่จะหาเทคนิคที่เหมาะสมกับ noise ที่เกิดขึ้น จากการศึกษาพบว่า median filtering เป็นเทคนิคที่นิยมใช้กัน โดยกระบวนการของเทคนิคดังกล่าวคือจะสร้าง window ที่มีขนาดตามที่ user ระบุมาครอบทุกๆ Pixel โดยเริ่มตั้งแต่บิตแรกจนถึงบิตสุดท้ายของภาพโดยจะอ่านค่า pixel ที่ทุกครอบมาเรียงจากน้อยไปมากและเลือกค่ากลางมาเป็น pixel ใหม่ให้กับภาพ



รูปที่ 3.6 การกำจัดสิ่งรบกวน ด้วย window ขนาดต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองจะเห็นว่าถ้าเลือกขนาดของ window ที่มีขนาดเล็กเกินไปจะไม่สามารถกำจัด noise ได้หมดแต่ถ้าเลือกขนาดของ window ที่มีขนาดใหญ่เกินไปจะทำให้ภาพที่ได้มีความเพี้ยนไปจากต้นฉบับทำให้เป็นอุปสรรคต่อการประมวลผลภาพ

หลังจากกำจัด noise ออกไปแล้วเราจึงนำเฟรมที่ได้มาทำการตรวจจับการเคลื่อนไหวตามขั้นตอนดังนี้

รูปที่ 3.7 Flow Chart การทำงานของระบบนับวัตถุที่เข้าหรือออกจากเฟรม

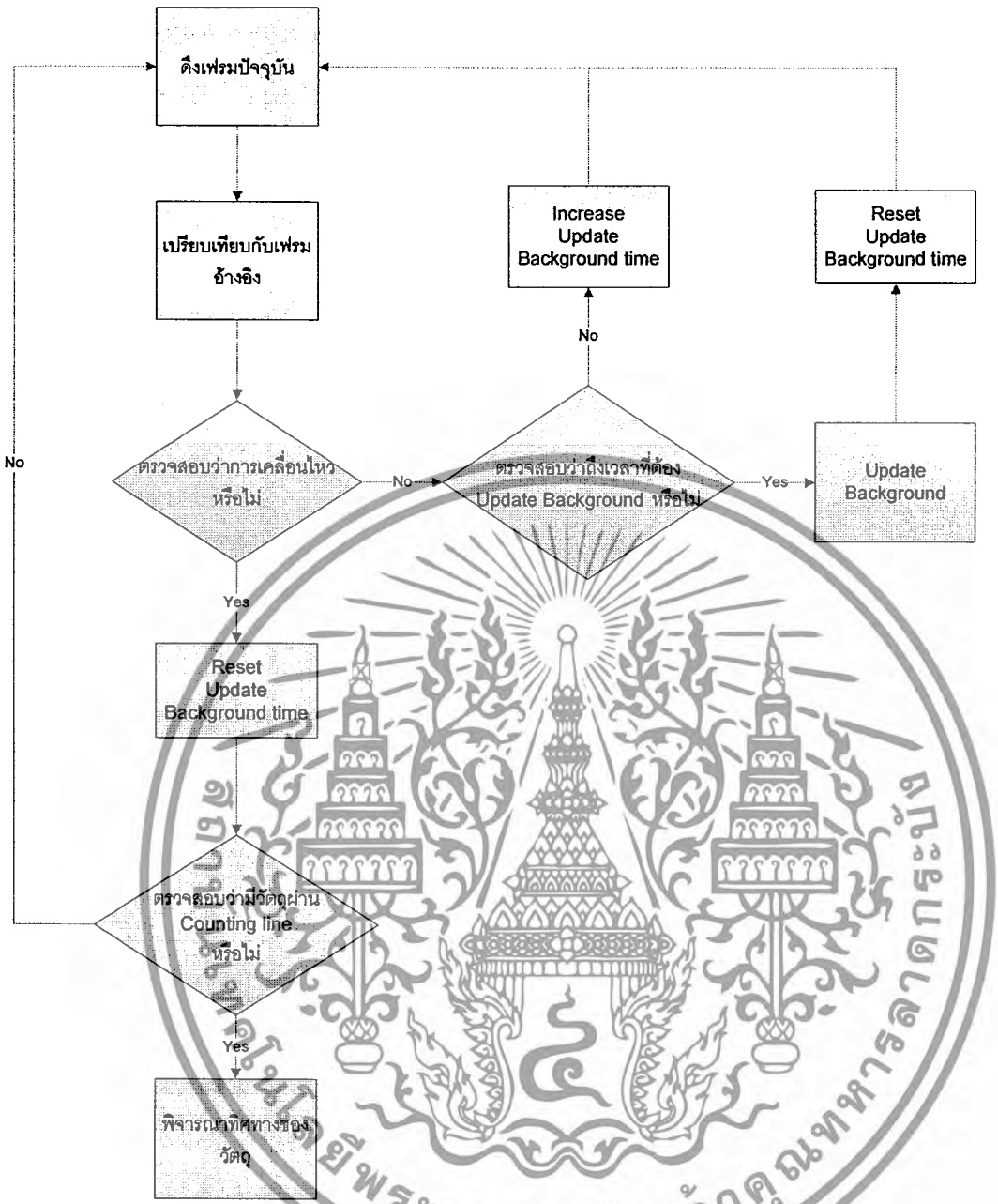
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพิจารณาหาทิศทางการเคลื่อนที่ของวัตถุที่ผ่าน counting line จะใช้ประโยชน์ของ direction line โดยพิจารณาว่าเมื่อวัตถุมีการเคลื่อนผ่าน counting line แล้ววัตถุมีการเคลื่อนที่ผ่าน direction line ในทิศทางเดียวกับ counting line หรือไม่ถ้ามีจะถือว่าวัตถุนั้นมี direction เป็น in ถ้าไม่มีจะถือว่าวัตถุนั้นมี direction เป็น out

3.4.3 Update Background

จากการทดลองพบว่า median filtering นั้นไม่สามารถแก้ปัญหาดังกล่าวได้อย่างถูกต้อง 100 % เนื่องจากในกรณีที่เกิดการเปลี่ยนแปลงของระดับแสงมาก จะทำให้ median filtering ไม่สามารถกำจัด noise ได้หมด ดังนั้นเทคนิคที่จะมารองรับกรณีดังกล่าวคือ การอัพเดตพื้นหลัง โดยมีกระบวนการการทำงานคือ การกำหนดให้มีการอัพเดตพื้นหลังเป็นช่วงเวลาในกรณีที่พบว่าไม่มีการเคลื่อนไหวที่เกิดขึ้นในเฟรม ข้อดีของเทคนิคดังกล่าวคือสามารถลดผลกระทบของการเปลี่ยนระดับความเข้มแสงในระดับที่ median filtering ไม่สามารถที่จะกำจัด noise ได้หมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 Flow Chat การทำงานของระบบโดยใช้เทคนิค Update Background

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.4 การทำ Blob Analysis

Blob คือการรวมกลุ่มของพิกเซลที่มีตำแหน่งติดกันและมีระดับสีใกล้เคียงกันให้เป็นกลุ่มใหญ่ โดยประโยชน์ของ blob คือสามารถดึงวัตถุที่สนใจภายในเฟรมมาพิจารณาได้ ดังนั้นจึงมีความจำเป็นที่ต้องมีการศึกษาเรื่อง blob เพื่อมาประยุกต์ในการทำโครงงาน เนื่องจากเมื่อเราสามารถจำแนกได้แล้วว่าเกิดเคลื่อนไหวขึ้นภายในเฟรมจากนั้นนำผลของการเปรียบเทียบซึ่งแสดงวัตถุที่มีการเคลื่อนไหวในเฟรมปัจจุบัน ซึ่งการพิจารณาวัตถุต่างๆที่จับได้ว่ามีความน่าจะเป็นที่จะเป็นมนุษย์หรือไม่นั้นต้องอาศัยเทคนิคดังกล่าว

(a) (b)

รูปที่ 3.9 ตัวอย่าง ก่อน-หลัง การทำ Blob

จากภาพ 3.9 (b) แสดง blob ทั้งหมดที่โปรแกรมจับได้ โดย blob ที่จับได้จะมีการระบายสีเพื่อแสดงให้เห็น blob แต่ละตัว จากภาพจะเห็นว่ามี blob ทั้งหมด 4 ตัว

แต่เนื่องจากกระบวนการการจับการเคลื่อนไหวของโปรแกรมที่ประกอบด้วย เทคนิค median filtering และ binarization นั้นอาจทำให้ขนาดที่แท้จริงของวัตถุมีความคลาดเคลื่อนได้ซึ่งจะส่งผลให้การหา Blob มีความผิดพลาดตามไปด้วยดังนั้นจึงต้องมีการเพิ่มเทคนิคในการขยายภาพ (Dilation) ซึ่งจะทำให้ลดความคลาดเคลื่อนดังกล่าวลงได้บ้างดังรูป 3.10

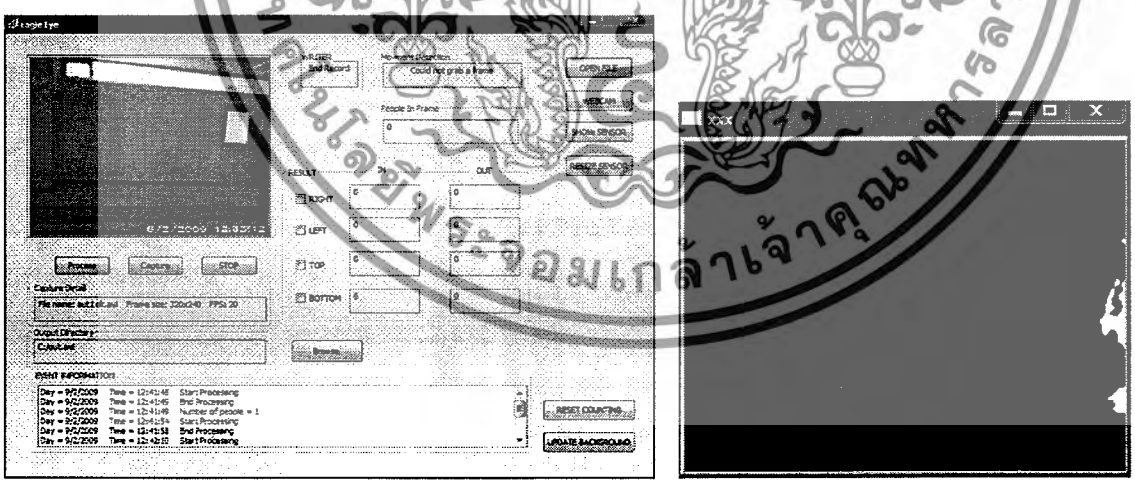
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(a) (b) (c)

รูปที่ 3.10 การทำ dilate (a) เฟรมปัจจุบัน (b) เฟรมที่ได้จากการเทรชโฮล (c) เฟรมที่ผ่านการ dilate แล้ว

เมื่อเราได้ blob ที่สมบูรณ์มาแล้วต่อไปคือการพิจารณาว่า blob แต่ละ blob นั้นมีความเป็นไปได้ไหมที่จะเป็นมนุษย์หรือไม่ โดยใช้ความสามารถของ blob ให้เป็นประโยชน์นั้นคือคุณสมบัติเฉพาะตัวของ blob ตัวอย่างคุณสมบัติของ blob เช่น จุดศูนย์กลาง ขนาด ความสูง เส้นรอบรูป และ พื้นที่ ซึ่งคุณสมบัติเหล่านี้เป็นปัจจัยสำคัญในการระบุความเป็นไปได้ที่ว่า blob นั้นๆ จะเป็นมนุษย์หรือไม่

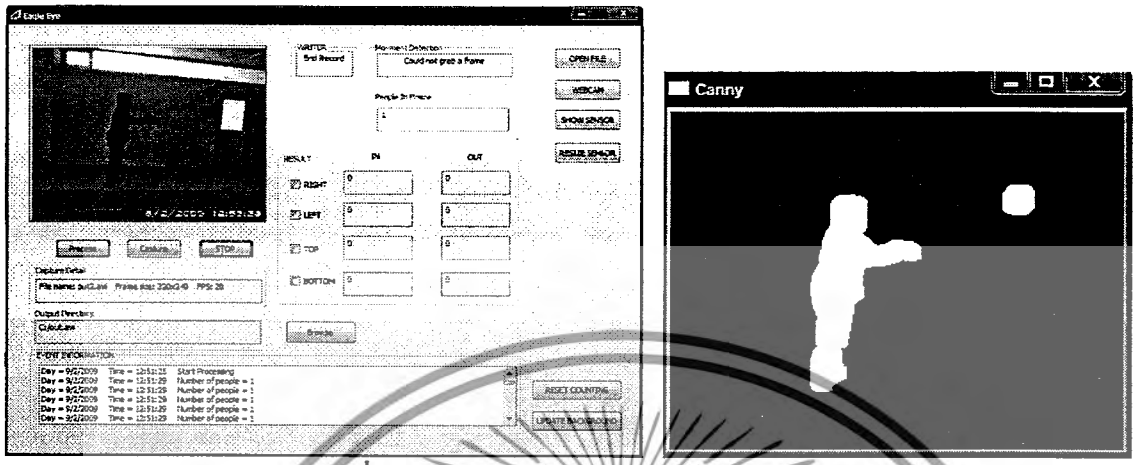
จากการศึกษาพบว่า พื้นที่ของ blob เป็นคุณสมบัติที่ดีในการจำแนก blob ที่ถูกต้องออกมาได้ โดยจากการทดลองหลายๆครั้งจะสามารถบอกได้ว่าขนาดของพื้นที่ของ blob ที่เป็นช่วงที่ถูกต้องนั้นเป็นเท่าไหร่ และจะทำให้การนับคนในเฟรมนั้นมีความถูกต้องแม่นยำมากขึ้น ดังรูป



รูปที่ 3.11 ผลการทำงานเมื่อวัตถุเริ่มเดินเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปเมื่อวัตถุเริ่มเดินเข้ามาผ่านที่ขอบสังเกตได้ว่าขนาดของ blob มีพื้นที่ ขนาดเล็กมาก เราจะไม่ถือว่าวัตถุนั้นเป็นมนุษย์



รูปที่ 3.12 การทำงานตรวจจับคนและวัตถุ

จากรูปสังเกตว่า โปรแกรมจะนับคนในเฟรมเพียงหนึ่งคนเท่านั้น โดยโปรแกรมจะไม่นับ blob ที่เป็นลูกบอลเพราะมีความน่าจะเป็นที่ blob ดังกล่าวจะ ไม่ใช่มนุษย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

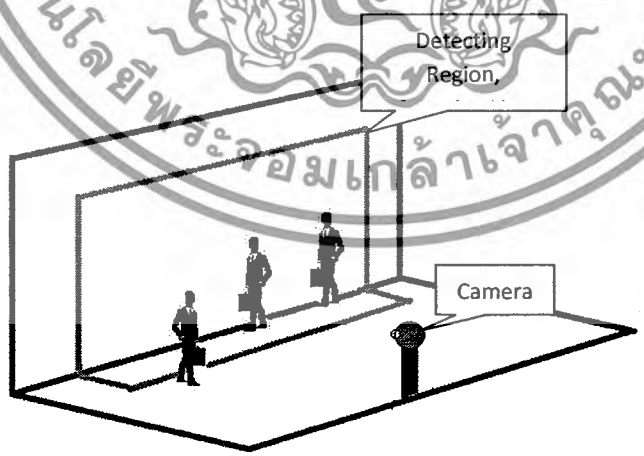
การทดลองและผลการทดลอง

4.1 แบบจำลองที่ใช้ในการจำลองระบบ

แบบจำลองเพื่อใช้ในการทดสอบผ่านเว็บแคมบนคอมพิวเตอร์ ได้มีการกำหนดค่าต่างๆ เพื่อให้ได้ผลการทดลองที่ถูกต้องดังต่อไปนี้

ตารางที่ 4.1 ค่าต่าง ๆ ที่เป็นข้อกำหนดในการทดลอง

ข้อกำหนด	ค่าที่ใช้
ความสูงของเว็บแคมจากพื้น	120 เซนติเมตร
ระยะห่างน้อยที่สุดจากเว็บแคมถึงวัตถุที่ผู้ใช้เลือก	3 เมตร
ระยะห่างมากที่สุดจากเว็บแคมถึงวัตถุที่ผู้ใช้เลือก	4 เมตร
จำนวนเว็บแคม	1 ตัว
ทิศทางการตรวจจับ	ขวา, ซ้าย
ค่า High Threshold	255
ค่า Low Threshold	128
จำนวนครั้งในการทดลองแต่ละกรณี	50



รูปที่ 4.1 แบบจำลองสถานที่ที่ใช้ในการทดลอง

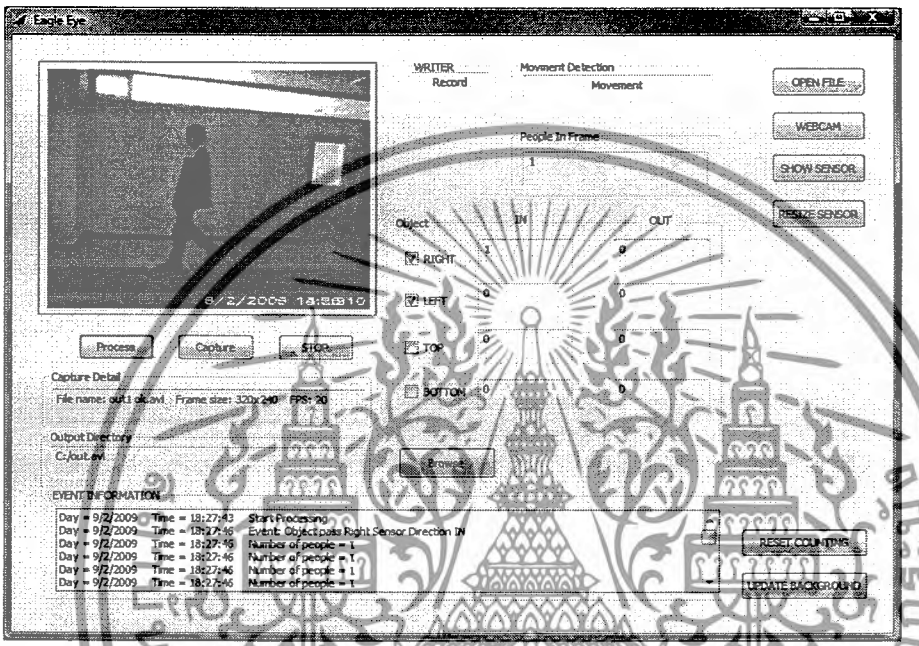
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองและผลการทดลอง

จากแบบจำลองที่กำหนดไว้ ทำให้ได้ผลการทดลองต่างๆ ดังต่อไปนี้

4.2.1 การทดลองโดยให้คนเดินผ่านกล้องทีละ 1 คน

ทำการทดลองโดยให้คนเดินผ่านหน้ากล้องเพียงคนเดียว พบว่าสามารถตรวจนับจำนวนคนที่เดินผ่านเข้า – ออก และจำนวนคนภายในเฟรมได้อย่างถูกต้อง

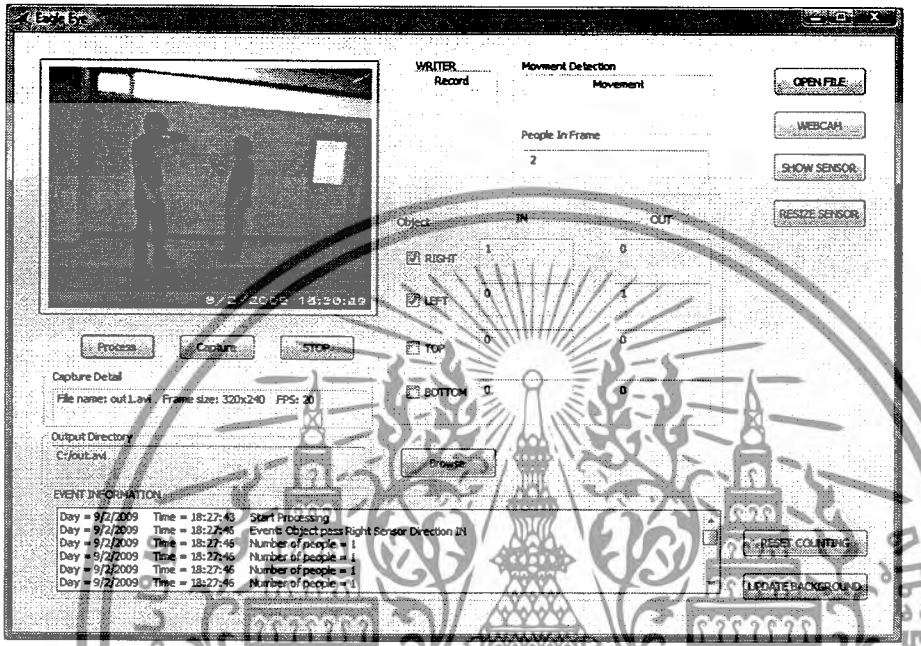


รูปที่ 4.2 หน้าจอแสดงผลการทดลอง (4.2.1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การทดลองโดยให้คนเดินผ่านกล้องทีละ 2 คน

ทำการทดลองโดยให้คนเดินผ่านหน้ากล้องพร้อมกัน 2 คน พบว่าการตรวจนับจำนวนคนที่เดินผ่านเข้า – ออก ยังมีความผิดพลาดอยู่บ้าง แต่การนับจำนวนคนภายในเฟรมยังมีความถูกต้องใกล้เคียงกับการทดลองโดยให้คนเดินผ่านกล้องทีละ 1 คน

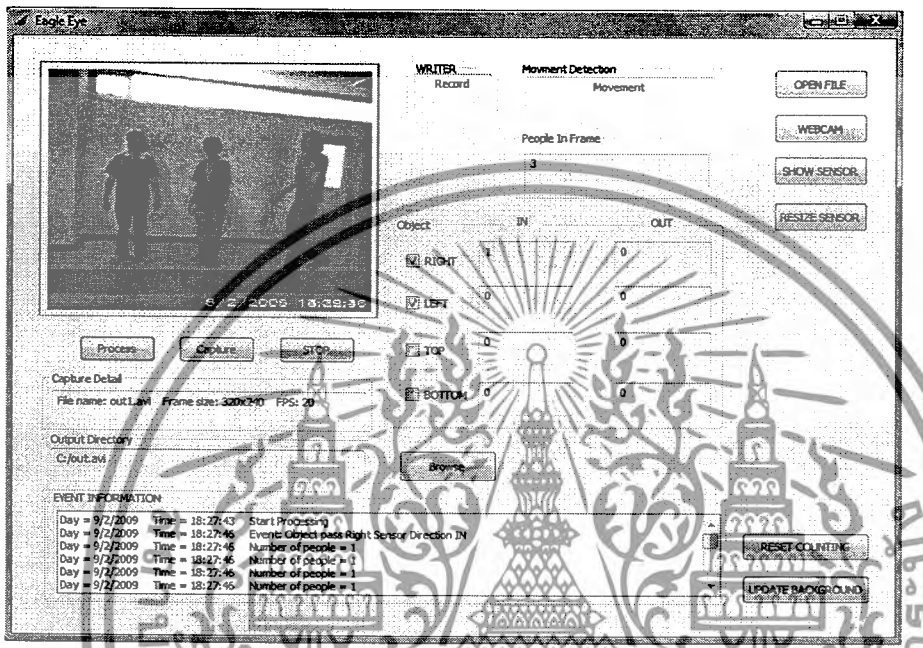


รูปที่ 4.3 หน้าจอแสดงผลการทดลอง (4.2.2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 การทดลองโดยให้คนเดินผ่านกล้องทีละ 3 คน

ทำการทดลองโดยให้คนเดินผ่านหน้ากล้องพร้อมกัน 3 คน พบว่าการตรวจนับจำนวนคนที่เดินผ่านเข้า – ออก มีความผิดพลาดมากขึ้น แต่การนับจำนวนคนภายในเฟรมยังมีความถูกต้องใกล้เคียงกับการทดลองโดยให้คนเดินผ่านกล้องทีละ 1 คน

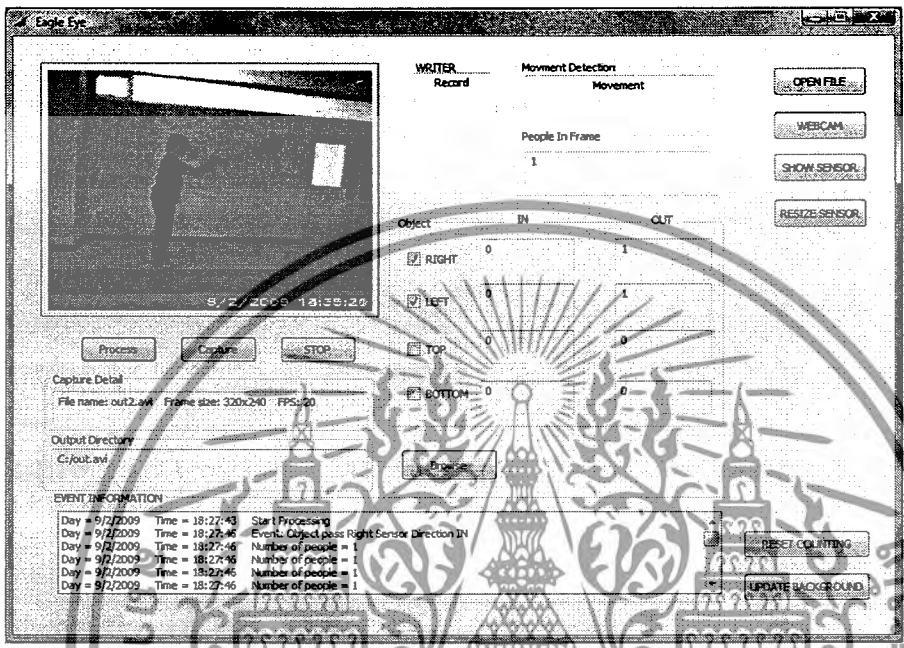


รูปที่ 4.4 หน้าจอแสดงผลการทดลอง (4.2.3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 การทดลองเพื่อทดสอบการจำแนกบุคคลและวัตถุ

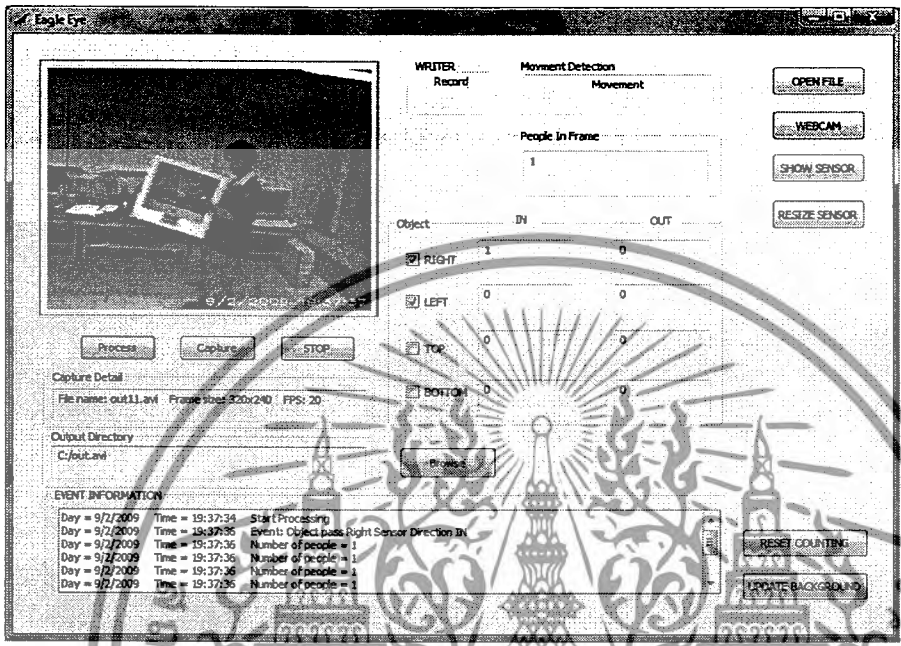
การทดลองนี้ทำการทดลอง 2 แบบ โดยแบบแรกให้คนเดินผ่านหน้ากล้อง 1 คน จากนั้นให้คนที่อยู่ภายนอกโยนลูกบาสเข้ามา พบว่ามีการตรวจนับลูกบาสเป็นคนที่เดินผ่านเข้า – ออก แต่จะไม่นับลูกบาสเป็นจำนวนคนภายในเฟรม



รูปที่ 4.5 หน้าจอแสดงผลการทดลอง (4.2.4 - a)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองแบบที่ 2 ทำการทดลองโดยให้คนถือกล่องขนาดใหญ่เดินผ่านหน้ากล้อง พบว่ามีความผิดพลาดเกิดขึ้นบางครั้งในการตรวจนับจำนวนคนที่เดินผ่านเข้า – ออก และจำนวนคนภายในเฟรม

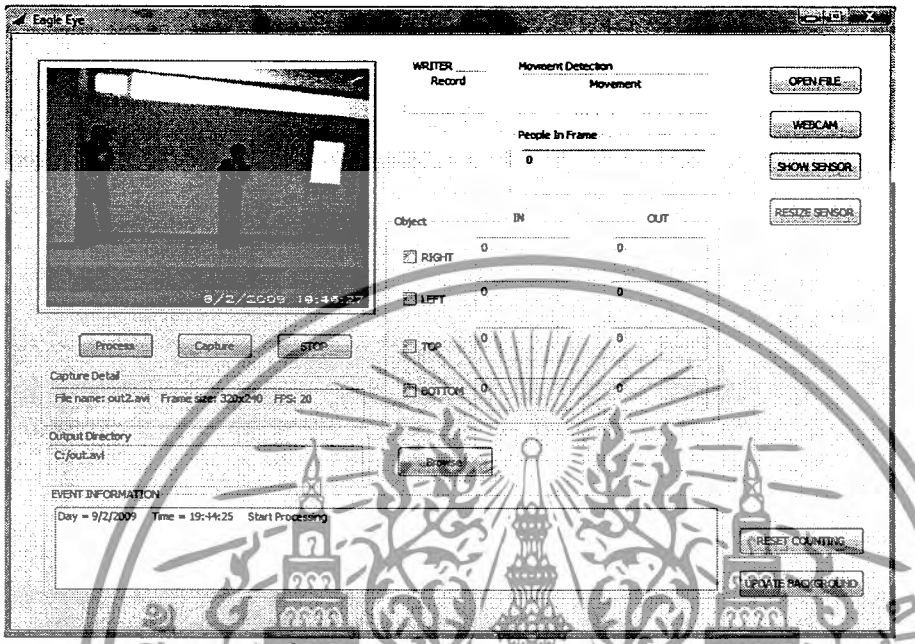


รูปที่ 4.6 หน้าจอแสดงผลการทดลอง (4.2.4 - b)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

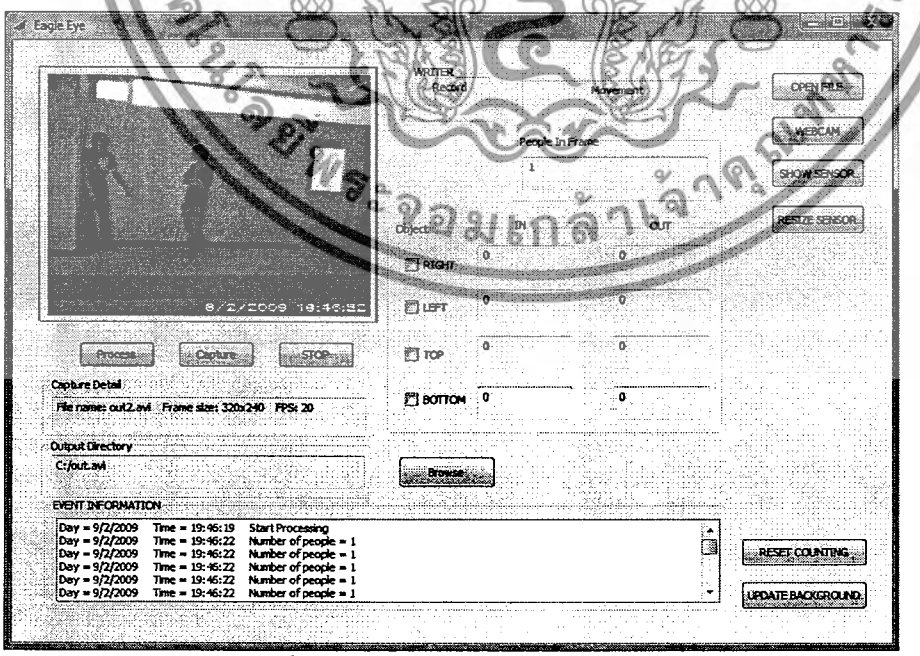
4.2.5 การทดลองเพื่อทดสอบการนับจำนวนคนที่เคลื่อนไหว

ทำการทดลองโดยให้คนอยู่ในเฟรมตั้งแต่เริ่มทำการทดลอง เมื่อให้โปรแกรมเริ่มทำงาน พบว่า หากคนไม่เคลื่อนไหวจะไม่นับจำนวนคนที่อยู่ภายในเฟรม



รูปที่ 4.7 หน้าจอแสดงผลการทดลอง (4.2.5 - a)

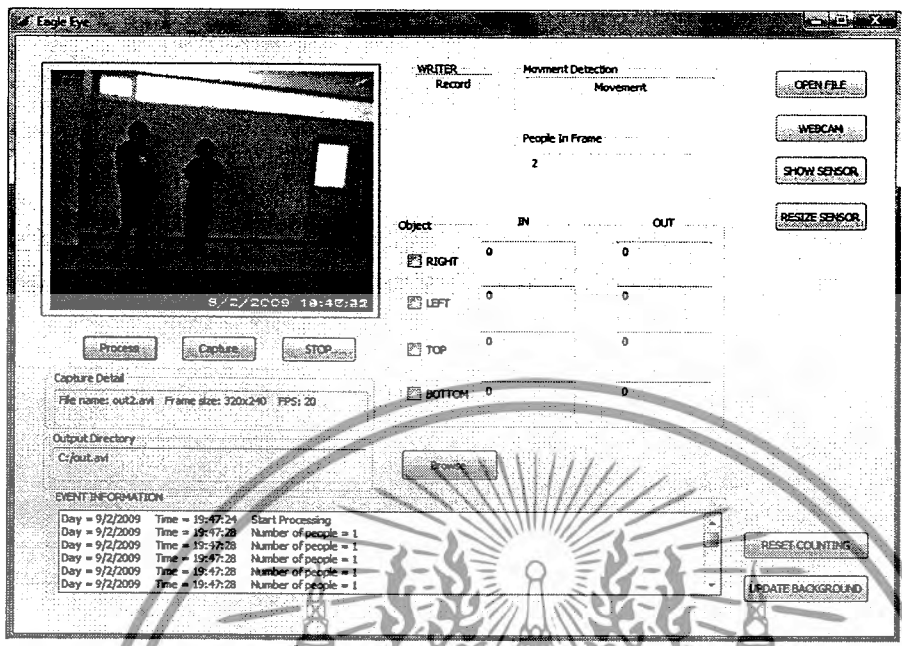
เมื่อให้คนหนึ่งเคลื่อนที่ โปรแกรมจะทำการนับจำนวนคนที่เคลื่อนไหวในเฟรม



รูปที่ 4.8 หน้าจอแสดงผลการทดลอง (4.2.5 - b)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อให้อีกคนหนึ่งเคลื่อนที่ โปรแกรมจะทำการนับจำนวนคนที่เคลื่อนไหวทั้ง 2 คน



รูปที่ 4.9 หน้าจอแสดงผลการทดลอง (4.2.5 – c)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 สรุปผลการทดลอง

จากการทดลองสามารถสรุปได้ว่าโปรแกรมสามารถทำงานได้อย่างมีประสิทธิภาพสูงสุด เมื่อมีคนผ่านหน้ากล้องเพียง 1 คนเท่านั้น และจะเริ่มเกิดความผิดพลาดขึ้นเมื่อมีจำนวนคนที่ผ่านหน้ากล้องเพิ่มมากขึ้น ซึ่งจากการทดลองสามารถคำนวณเปอร์เซ็นต์ความถูกต้องในการทำงานของโปรแกรมได้ดังนี้

ตารางที่ 4.2 สรุปผลการทดลอง

การทดลอง	การนับวัตถุผ่านเส้นตรวจจับ	การนับวัตถุผ่านเส้นตรวจจับ (เข้า)	การนับวัตถุผ่านเส้นตรวจจับ (ออก)	การนับจำนวนคนในเฟรม
คนเดินผ่าน 1 คน	94 %	84 %	92 %	94 %
คนเดินผ่าน 2 คน	88 %	68 %	82 %	85 %
คนเดินผ่าน 3 คน	88 %	54 %	74 %	75 %
การจำแนกคนและวัตถุ	-	-	-	92 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป

5.1 สรุปผลการดำเนินงาน

โครงงานนี้เป็นโครงงานที่ค่อนข้างจะมีความซับซ้อน ดังนั้นถ้าต้องการจะพัฒนาให้ได้ตามเป้าหมายที่แท้จริงคือวิเคราะห์การเคลื่อนไหวร่างกายเพื่อนำไปสู่การรู้จำการเคลื่อนไหวของร่างกายและจำแนกเอกลักษณ์ของแต่ละบุคคลเพื่อนำไปใช้ในการระบุตัวตนจากการวิเคราะห์การเคลื่อนไหวของร่างกาย ยังจำเป็นต้องศึกษาข้อมูลและทำการทดลองในเชิงลึกอีกมาก ยิ่งถ้าต้องการความถูกต้องแม่นยำด้วยแล้วนั้น ยิ่งต้องอาศัยความซับซ้อนยิ่งขึ้นไปอีก โครงงานนี้จึงเป็นจุดเริ่มต้นที่ดีในการศึกษาและพัฒนาต่อของรุ่นถัดไปหรือบุคคลที่สนในหัวข้อดังกล่าว

5.2 ข้อเสนอแนะ

โครงงานนี้เป็นการศึกษาในเรื่องของการประมวลผลภาพ (Image Processing) ซึ่งถ้าศึกษาอย่างละเอียดจะเห็นว่ามีปัจจัยหลายปัจจัยที่ควรพิจารณาและเป็นอุปสรรคต่อการประมวลผลภาพ อาทิเช่น การกำจัดสัญญาณรบกวน สภาพแวดล้อมในการประมวลผล รวมถึงการทำงานของโปรแกรมซึ่งโปรแกรมสำหรับการประมวลผลภาพส่วนใหญ่ล้วนใช้หน่วยความจำและหน่วยการประมวลผลที่มากดังนั้นถ้าผู้สนใจพัฒนาต่อควรคำนึงถึงอัลกอริทึมต่างๆที่จะพัฒนาขึ้นมาว่าสามารถรองรับกับปัญหาดังกล่าวได้หรือไม่ ข้อคำนึงถัดไปคือการทำงานแบบ Real time ของโปรแกรมซึ่งอาจมีข้อจำกัดทางฮาร์ดแวร์หรือเทคนิคทางการประมวลผลภาพที่ต้องมีการทำงานที่รวดเร็วรองรับกรณีดังกล่าวได้ และอีกข้อพิจารณาที่สำคัญอีกข้อหนึ่งคือสิ่งแวดล้อมในการประมวลผลภาพซึ่งจากการทดลองพบว่ามีผลเป็นอย่างมาก ยกตัวอย่าง เช่น สัญญาณรบกวน (noise) พบว่าในแต่ละสถานที่นั้นมีสัญญาณรบกวนที่แตกต่างกันดังนั้นเป็นการยากมากที่แอปพรีเคชันที่พัฒนาขึ้นมานั้นจะใช้งานได้ครอบคลุมในทุกสถานที่และทุกกรณี การแก้ปัญหาดังกล่าวคือการจำกัดขอบเขตของโครงงานให้รัดกุมและเหมาะสมที่สุดเพื่อใช้ในสถานที่ที่หนึ่งและในกรณีใดกรณีหนึ่งเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางการพัฒนาต่อ

แนวทางในการพัฒนาสำหรับผู้ที่สนใจที่ทางกลุ่มเห็นควรว่าเป็นเรื่องที่เหมาะสมและมีความสามารถที่จะพัฒนาต่อไปได้คือ Blob Analysis เนื่องจากเมื่อศึกษาจากเอกสารที่อธิบายรายละเอียดอย่างละเอียดแล้วจะเห็นว่า Blob มีความสามารถที่หลากหลายและมีคุณสมบัติที่มากมายที่จะนำมาพิจารณาว่า Blob เหล่านั้นมีความน่าจะเป็นที่จะเป็นวัตถุอะไรและอาจรวมถึงท่าทางของคนที่จับได้ด้วย แต่ทั้งนี้ทั้งนั้นการจะพัฒนาให้ไปได้ถึงขั้นนั้นยังจำเป็นอย่างมากที่จะต้องอาศัยการทดลองจำนวนมากเพื่อให้ได้มาซึ่งการระบุตัวตนที่ถูกต้องแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

กฤษฎา สวัสดี และภาณุพงศ์ ศิริพร ณ ราชสีมา. 2545. "โปรแกรมจับรูปแบบการเคลื่อนไหว." ปริญญานิพนธ์วิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

กฤษณ์ ลี้เจริญ. 2549. "โครงร่างเพื่อการประมวลผลภาพโดยใช้โปรแกรมแมทแลป." ปริญญานิพนธ์วิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

จิระศักดิ์ สิทธิกร. 2551. **Image Processing.** [Online]. Available : http://www.kmitl.ac.th/~ksjirasa/Lecture/ImageProcessing.html.

พิพัฒน์พล ลาภอมรภิญโญ และสถิตพร ยอดนาราศรี. 2550. "การวิเคราะห์สภาพจราจรด้วยการประมวลผลภาพ." ปริญญานิพนธ์วิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

สกล จารุเสรีนนท์ และสุรพันธ์ จักรมณี. 2549. "ระบบสร้างตัวละคร 3 มิติโดยการตรวจจับการเคลื่อนไหวของร่างกาย." ปริญญานิพนธ์วิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

Advanced Info View Co.,Ltd. 2006. **Mpeg 4 คืออะไร และทำงานอย่างไร.** [Online]. Available : http://www.aiv-thai.com/article.php?id=18389&lang=th.

D. M. Gavrila. 1999. **The Visual Analysis of Human Movement: A Survey.** [Online]. Available : http://www.gavrila.net/Computer_Vision/Research/3-D_Human_Body_Tracking/cviu99.pdf.

Intel. 2007. **Introduction to OpenCV.** [Online]. Available : http://downloads.sourceforge.net/opencvlibrary/opencv_introduction_2007June9.pdf?modtime=1181770251&big_mirror=1.

J.W. Kim K.S. Choi W.S. Park J.Y. Lee and S.J. Ko, Ph.D. 2002. **ROBUST REAL-TIME PEOPLE TRACKING SYSTEM FOR SECURITY.** [Online]. Available : http://dali.korea.ac.kr/publication/int_pro/paper/12.pdf.

Takayuki Kanda. 2002. **Body Movement Analysis of Human-Robot Interaction.** [Online]. Available : http://www.irc.atr.jp/~kanda/pdf/kanda-ijcai03.pdf.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม (ต่อ)

Wikipedia. 2008. **Median filter.** [Online].Available: http://en.wikipedia.org/wiki/Median_filter.

Wikipedia. 2009. **MPEG-4.** [Online].Available : http://en.wikipedia.org/wiki/MPEG-4.

Wikipedia. 2009. **MPEG-4 Part 2.** [Online].Available : http://en.wikipedia.org/wiki /Advanced_Simple_Profile.

Wikipedia. 2009. **Xvid.** [Online]. Available : http://en.wikipedia.org/wiki/Xvid.

Xvid. 2009. **General Info.** [Online]. Available : http://www.xvid.org/General-Info.19.0.html.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

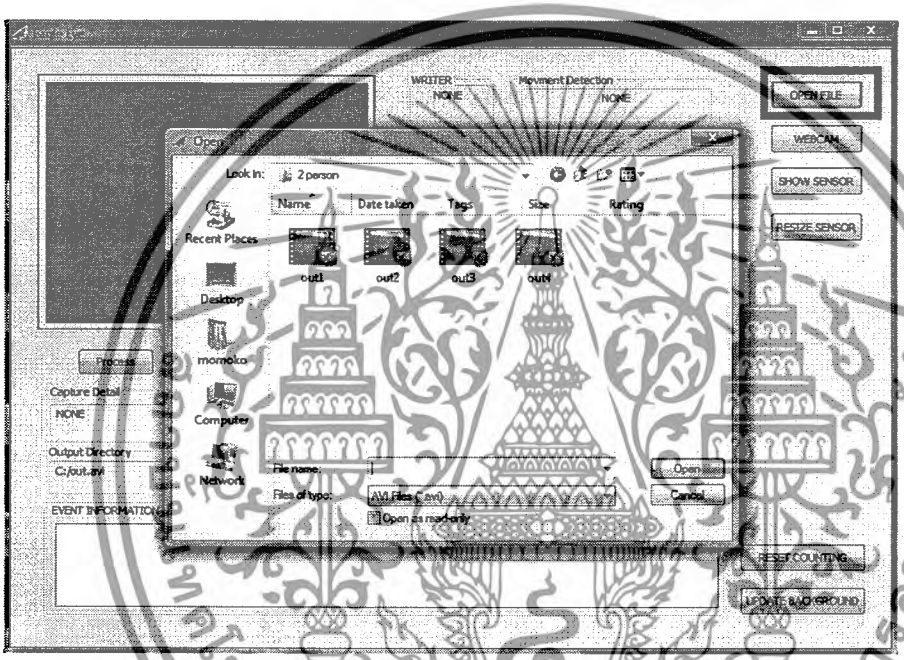
ภาคผนวก ก

คู่มือการใช้งาน

ก.1 การใช้งานโปรแกรม Eagle Eye

ก.1.1 การประมวลผลจากไฟล์วิดีโอ

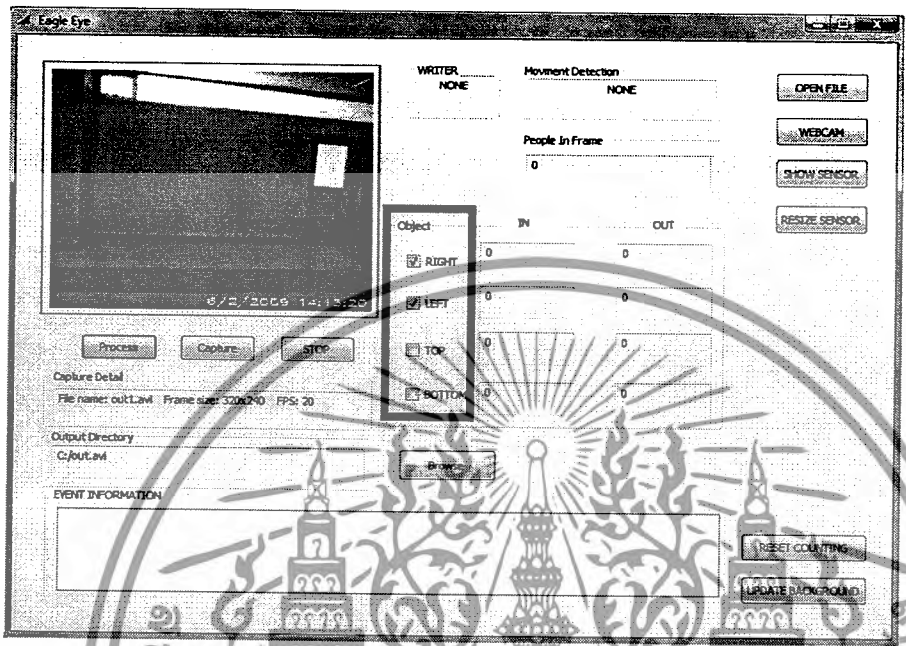
a. กดที่ปุ่ม OPEN FILE จะปรากฏหน้าต่างให้เลือกไฟล์ที่ต้องการประมวลผล



รูปที่ ก.1 การเปิดไฟล์

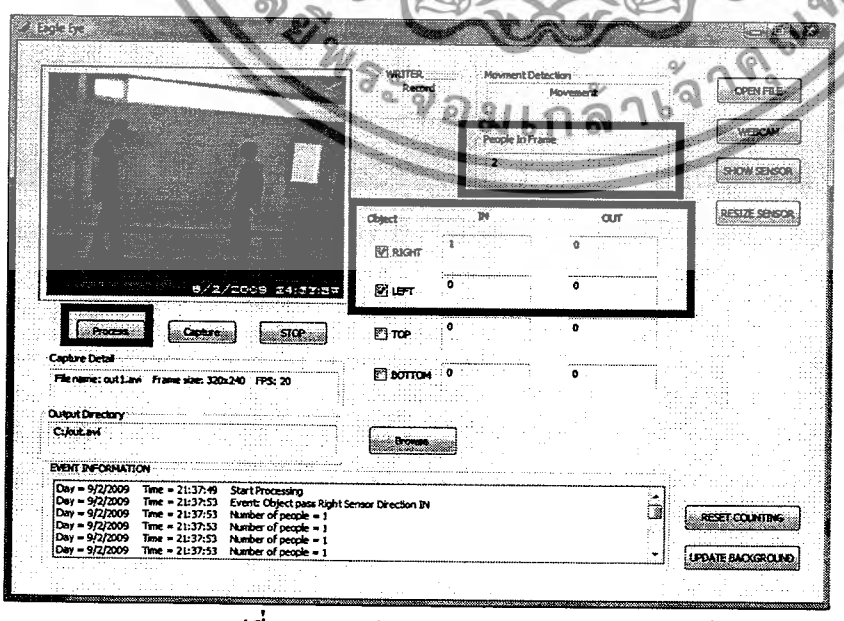
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b. เมื่อเลือกไฟล์แล้วโปรแกรมจะแสดงผลเฟรมแรกของไฟล์นั้น จากนั้นผู้ใช้สามารถเลือกทิศทางที่ต้องการตรวจจับการเข้าออกได้โดยการติ๊กเครื่องหมาย √ หน้าทิศทางที่ต้องการ



รูปที่ ก.2 การเลือกทิศทางการตรวจจับ

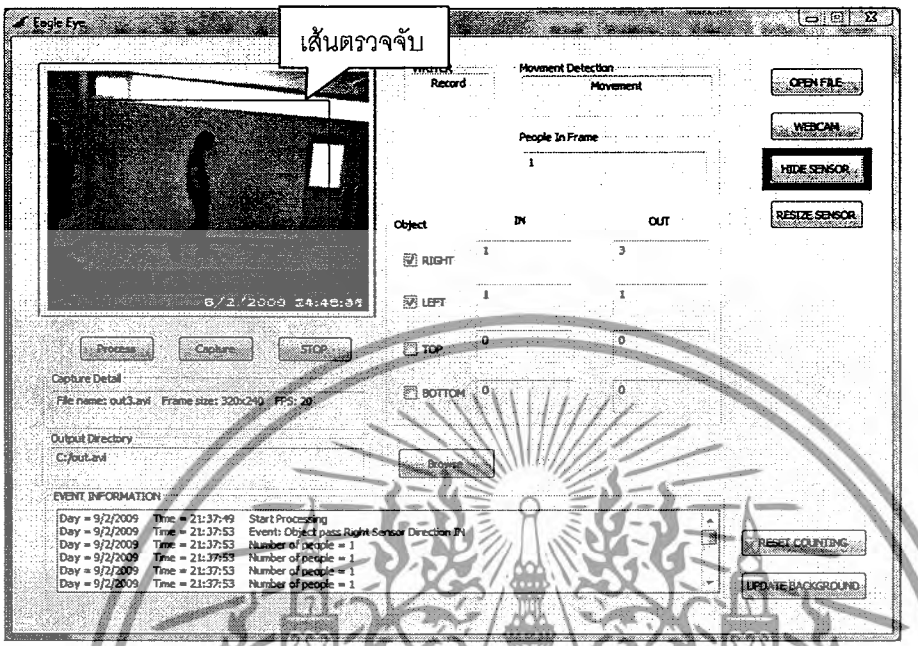
c. กดปุ่ม Process เพื่อเริ่มการประมวลผลไฟล์ โปรแกรมจะแสดงจำนวนคนที่เคลื่อนที่ภายในเฟรมในช่อง “People In Frame” และแสดงจำนวนคนเข้า – ออกในช่อง IN และ OUT



รูปที่ ก.3 การประมวลผล และ แสดงผลลัพธ์

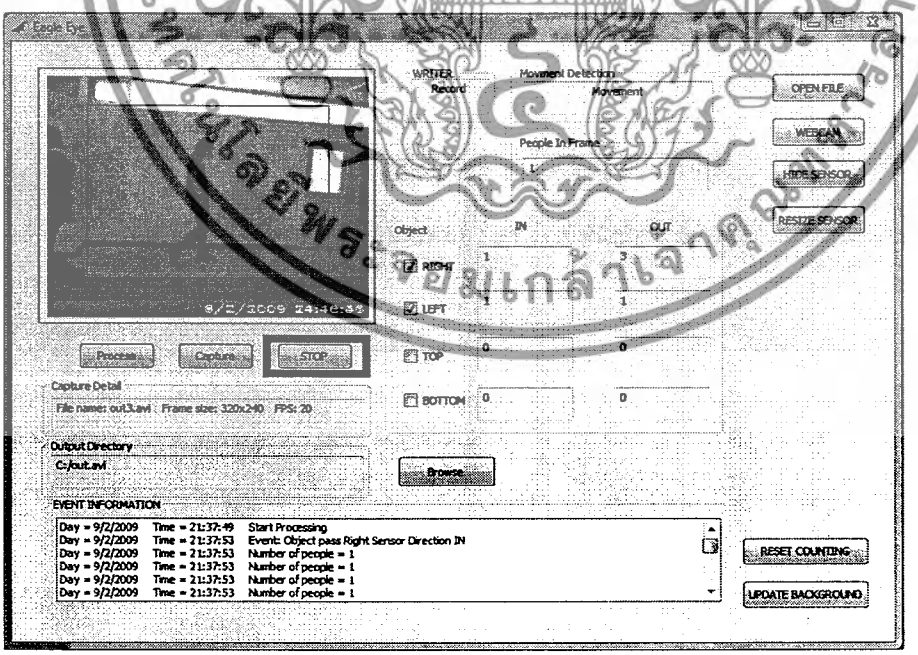
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

d. ผู้ใช้สามารถให้โปรแกรมแสดงเส้นตรวจจับได้โดยการกดปุ่ม SHOW SENSOR และสามารถปิดการแสดงผลเส้นตรวจจับได้โดยการกดปุ่ม HIDE SENSOR



รูปที่ ก.4 เส้นตรวจจับและปุ่มที่ใช้ในการซ่อน

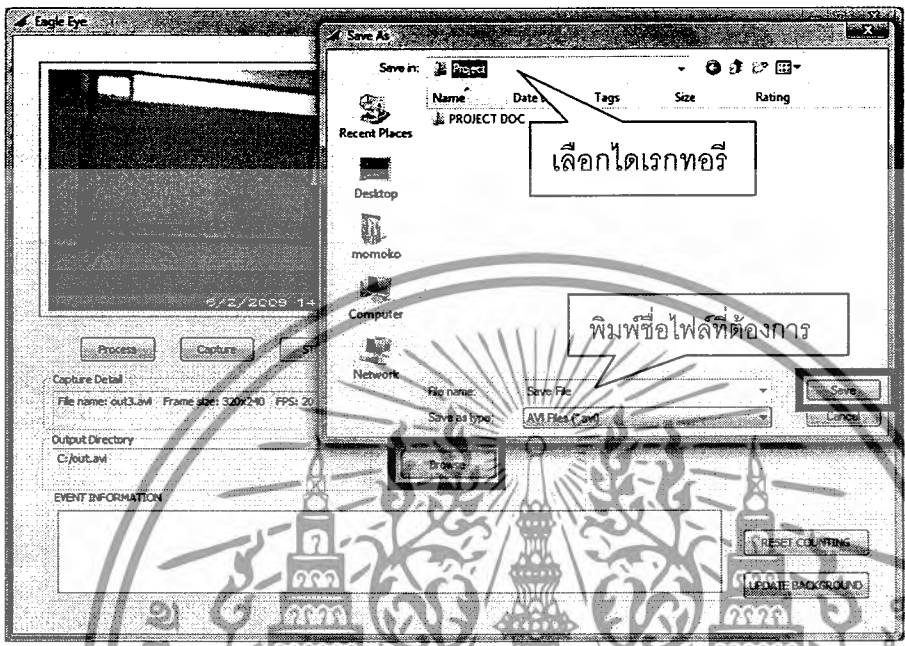
e. ผู้ใช้สามารถยกเลิกการประมวลผลไฟล์ด้วยการกดปุ่ม STOP



รูปที่ ก.5 การหยุดการประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

f. ไฟล์ที่ได้หลังการประมวลผลจะถูกบันทึกไว้ที่ C:/ โดยมีชื่อไฟล์ว่า out.avi แต่ผู้ใช้สามารถเปลี่ยนไดเรกทอรีที่บันทึกได้โดยการกดปุ่ม Browse จากนั้นเลือกไดเรกทอรีและพิมพ์ไฟล์ที่ต้องการบันทึกแล้วกดปุ่ม Save

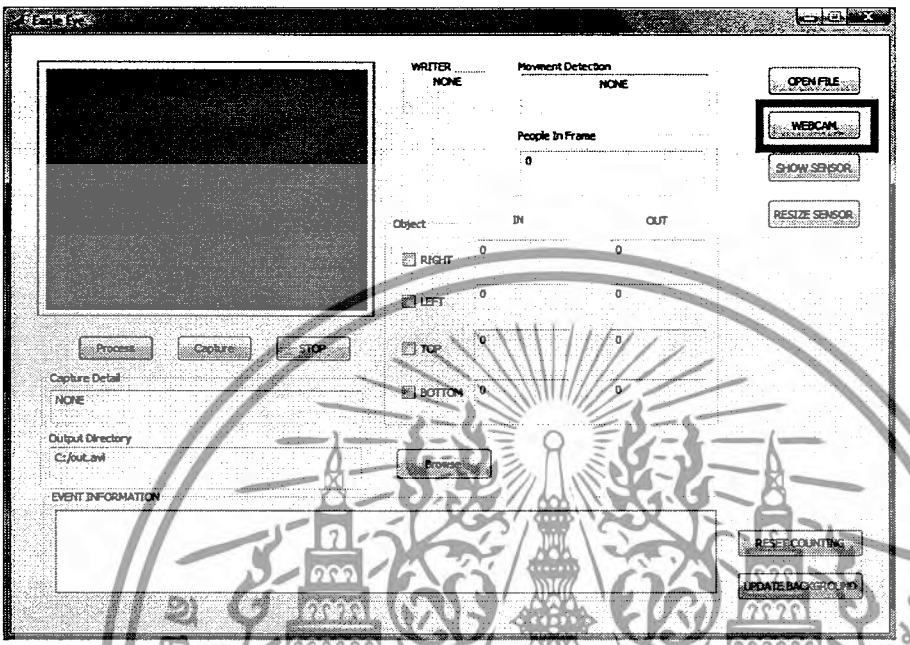


รูปที่ ก.6 การจัดเก็บไฟล์ผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

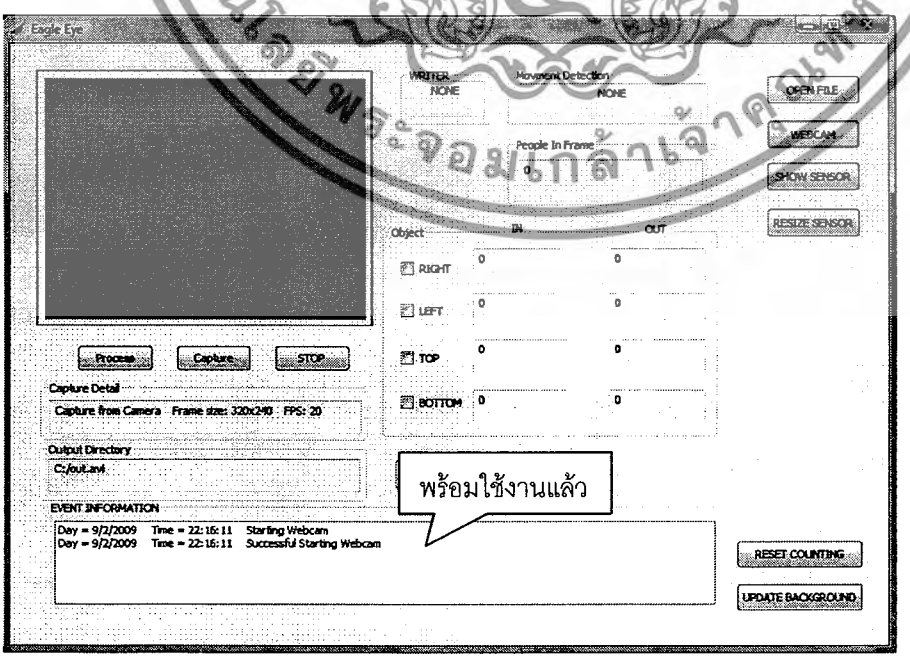
ก.1.2 การประมวลผลจาก WEBCAM

a. ผู้ใช้สามารถให้โปรแกรมประมวลผลจาก WEBCAM แบบ real-time ได้โดยกดปุ่ม “WEBCAM”



รูปที่ ก.7 การรับภาพจาก Webcam

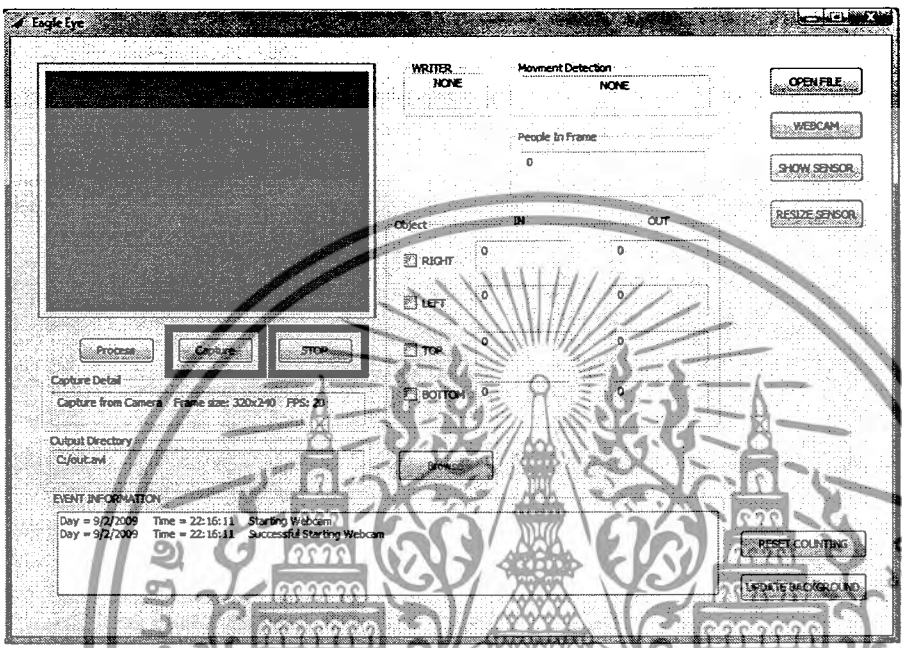
b. เมื่อ WEBCAM พร้อมใช้งานจะแสดงข้อความ “Successful Starting Webcam” ที่บริเวณ EVENT INFORMATION



รูปที่ ก.8 การแจ้งเมื่อพร้อมใช้งาน Webcam

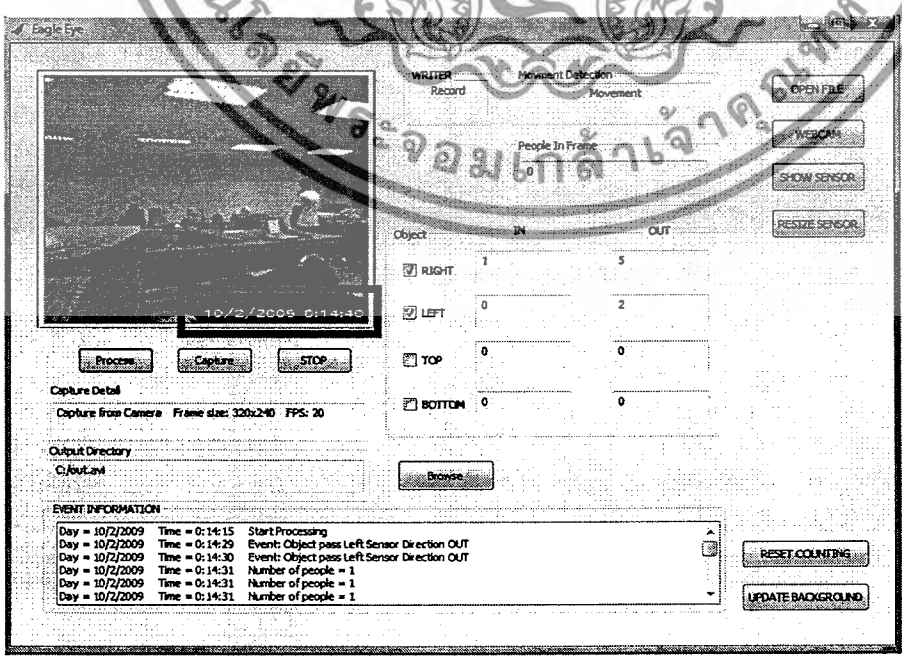
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

c. ในการใช้งาน WEBCAM นั้นจะมีอยู่ 2 โหมดคือการ PROCESS และ CAPTURE
   - PROCESS จะเป็นการประมวลผลภาพเช่นเดียวกับการประมวลผลไฟล์ แต่จะเป็นการประมวลผลแบบ real-time
   - CAPTURE จะเป็นการบันทึกไฟล์วิดิโอไว้ และสามารถประมวลผลได้ด้วย



รูปที่ ก.9 ปุ่มที่ใช้เลือกโหมดการใช้งาน Webcam

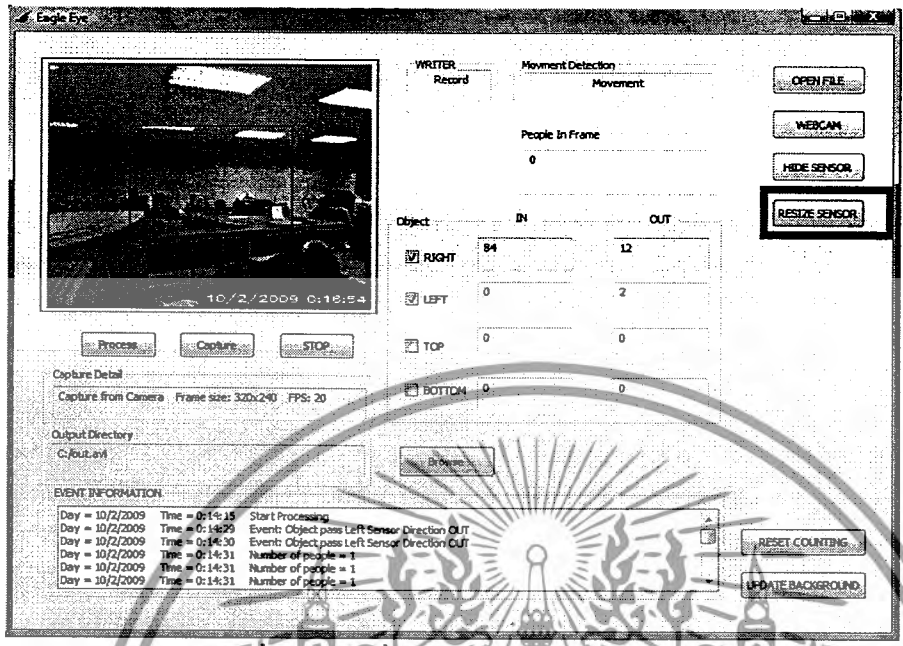
d. ทั้งโหมด PROCESS และ CAPTURE จะมีการบันทึกวันและเวลาลงไปในไฟล์



รูปที่ ก.10 เวลาที่ปรากฏในไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

e. ผู้ใช้สามารถเปลี่ยนขนาดเส้นตรวจจับได้โดยการกดปุ่ม RESIZE SENSOR



รูปที่ ก.ii ปุ่มที่ใช้กำหนดขนาดเส้นตรวจจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

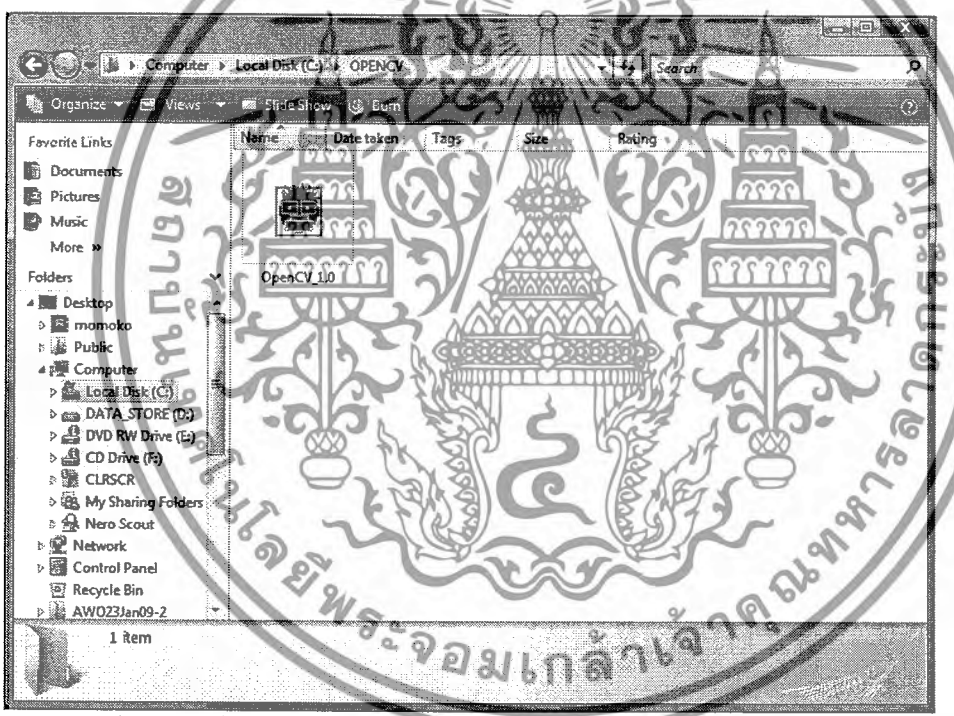
ภาคผนวก ข

เกี่ยวกับ OpenCV

ข.1 การติดตั้ง OpenCV

OpenCV สร้างขึ้นเพื่อให้การพัฒนาการมองเห็นของคอมพิวเตอร์เป็นเรื่องที่ง่ายขึ้น โดยสามารถดาวน์โหลด OpenCV ได้จาก http://sourceforge.net/projects/opencvlibrary/ โดยจะได้แฟ้มข้อมูลที่ชื่อว่า OpenCV_1.0.exe มา จากนั้นจึงทำการติดตั้งซอฟต์แวร์ OpenCV ตามขั้นตอนดังนี้

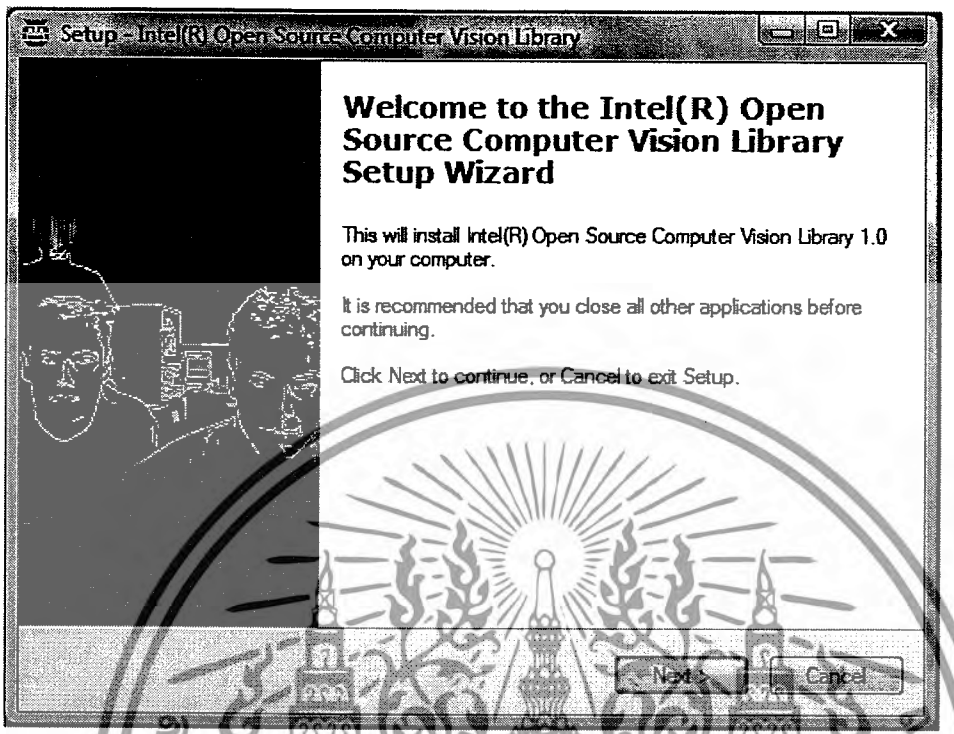
1. ดับเบิ้ลคลิกไฟล์ OpenCV_1.0.exe



รูปที่ ข.1 อธิบายขั้นตอนการติดตั้ง OpenCV ขั้นที่ 1

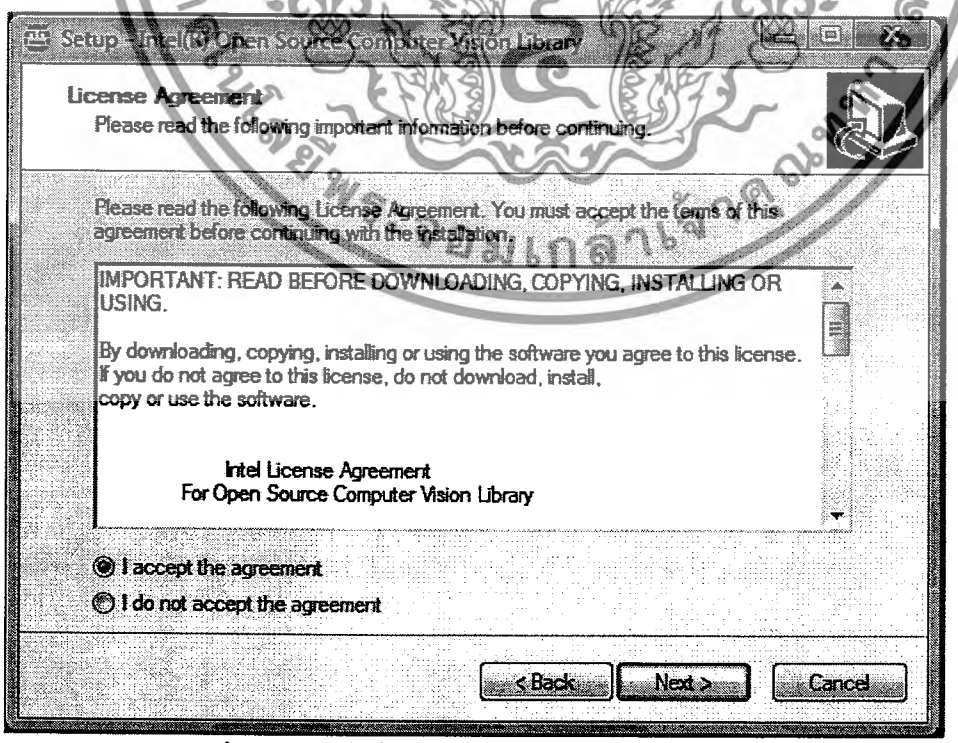
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กด Next >



รูปที่ ข.2 อธิบายขั้นตอนการติดตั้ง OpenCV ขั้นที่ 2

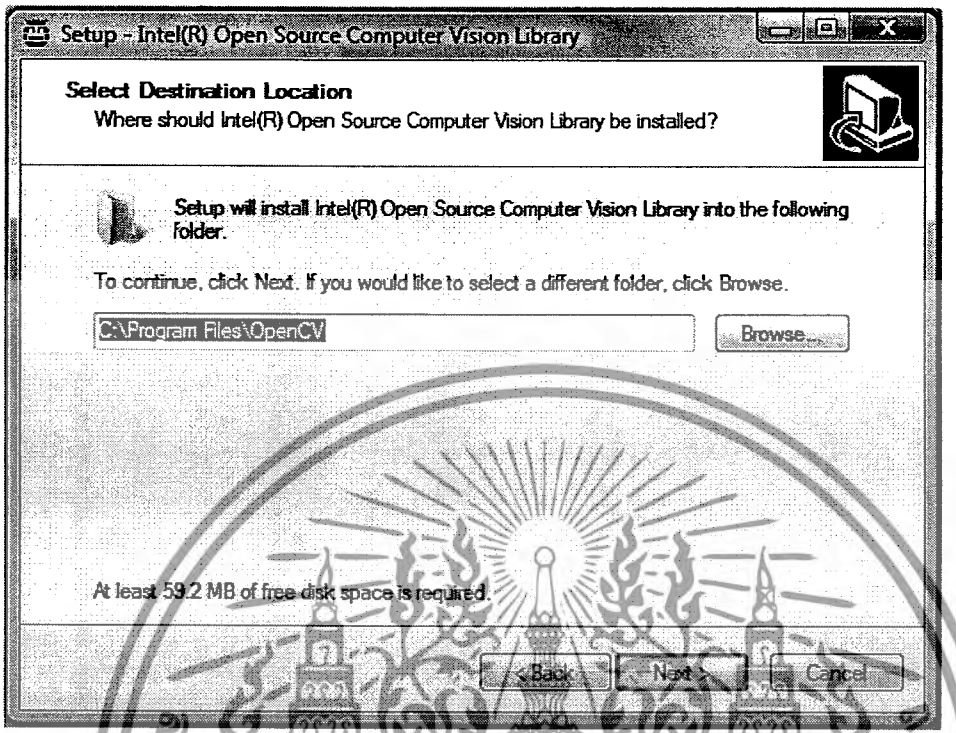
3. เลือก I accept the agreement จากนั้นกด Next >



รูปที่ ข.3 อธิบายขั้นตอนการติดตั้ง OpenCV ขั้นที่ 3

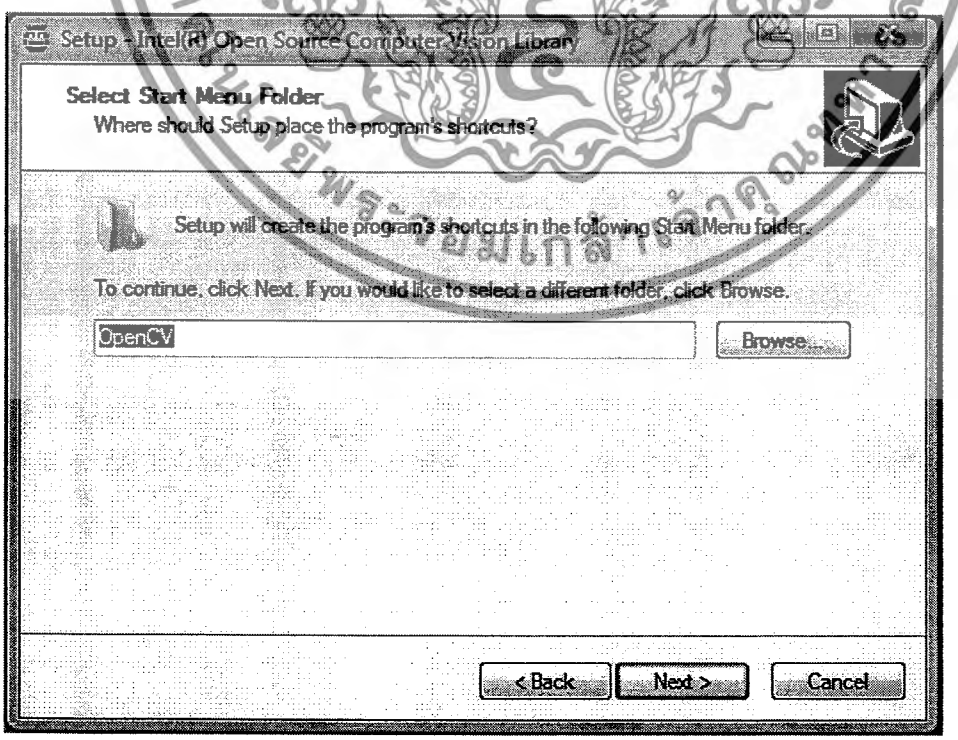
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เลือก Directory ที่ต้องการติดตั้งซอฟต์แวร์นี้ แล้วกด Next >



รูปที่ ข.4 อธิบายขั้นตอนการติดตั้ง OpenCV ขั้นที่ 4

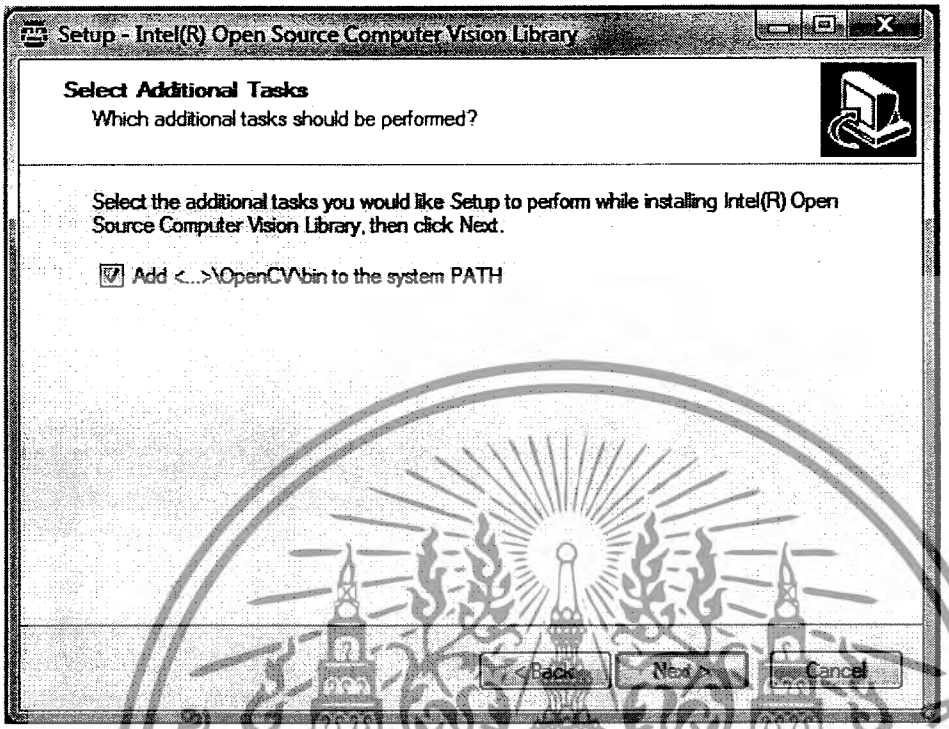
5. กด Next >



รูปที่ ข.5 อธิบายขั้นตอนการติดตั้ง OpenCV ขั้นที่ 5

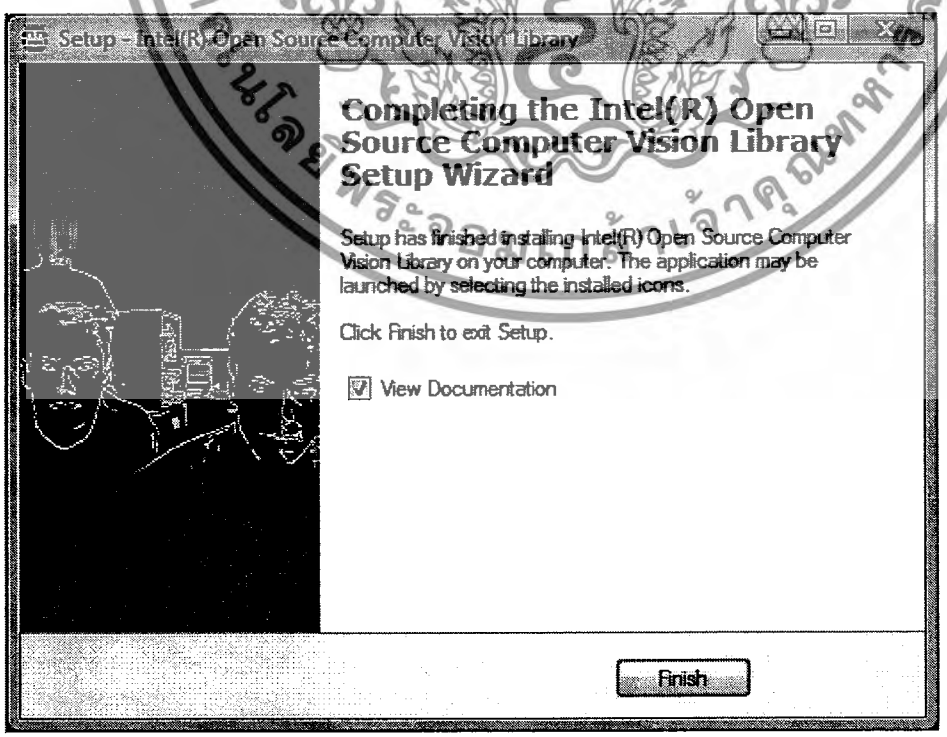
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. กด Next >



รูปที่ ข.6 อธิบายขั้นตอนการติดตั้ง OpenCV ขั้นที่ 6

7. รอโปรแกรมติดตั้งซอฟต์แวร์เสร็จ กด Finish



รูปที่ ข.7 อธิบายขั้นตอนการติดตั้ง OpenCV ขั้นที่ 7

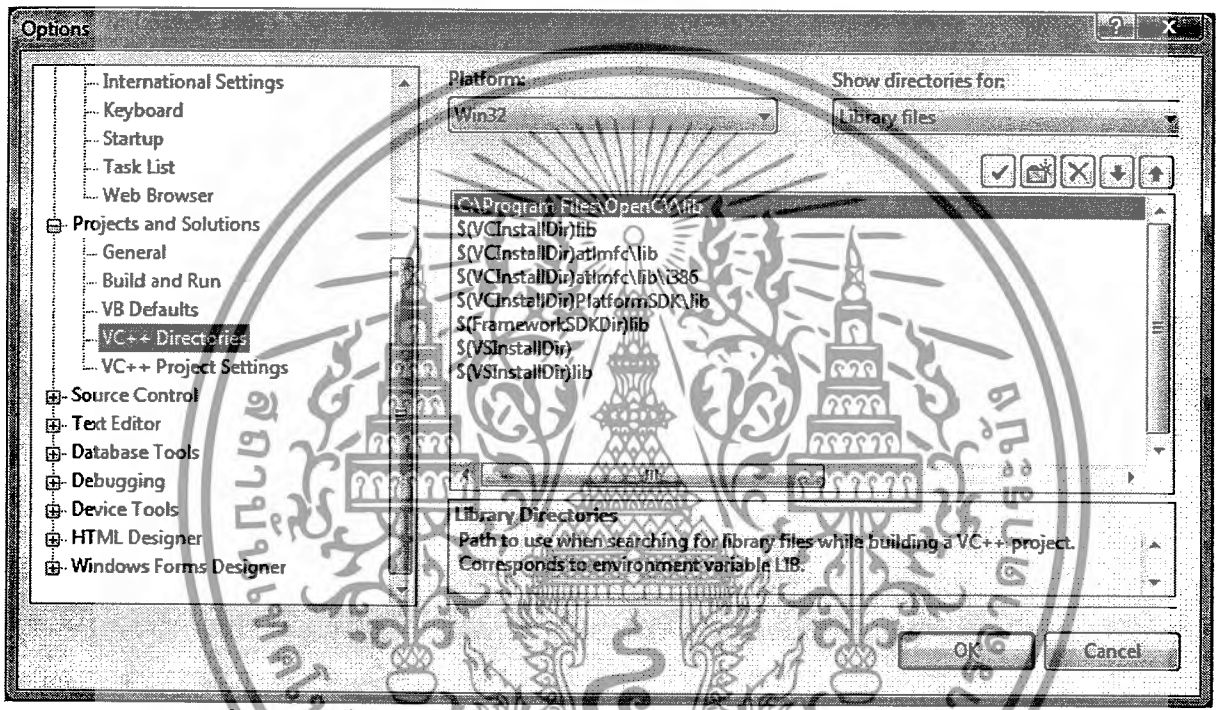
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.2 การตั้งค่า Visual Studio C++ ให้สามารถทำงานร่วมกับ OpenCV

เปิดโปรแกรม Visual Studio C++ ไปที่ Tool> options

- ที่ Project and solutions เลือก VC++ Directories
- ที่ Show Derectories for: เลือก **Library Files** >> add path เป็น

[root path]/OpenCV/lib โดยที่ [root path] คือ ที่ๆเราลง openCV ไว้ เช่น ถ้าอยู่ใน C:/Program Files จะได้ path เป็น C:/Program Files/OpenCV/lib



รูปที่ ข.8 การตั้งค่า Visual Studio C++ ให้ใช้งาน OpenCV ขั้นที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ที่ Show Derectories for: เลือก **Include Files** >> add path เป็น

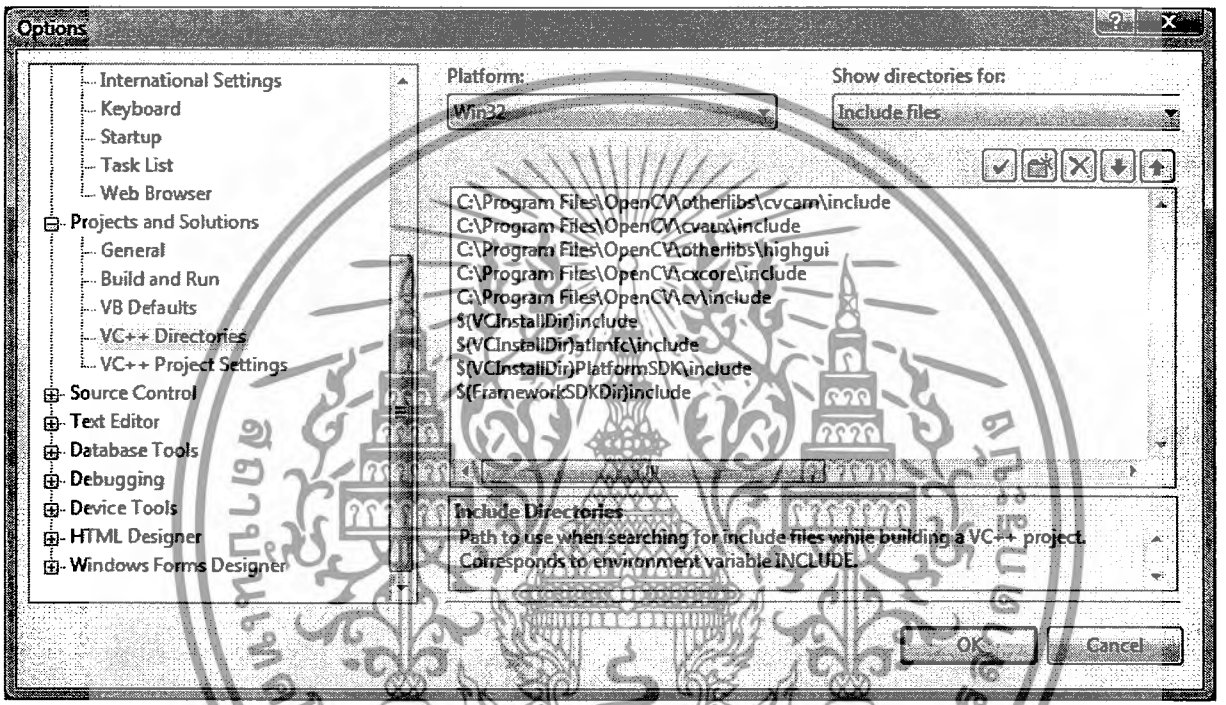
[root path]/OpenCV/cv/include

[root path]/OpenCV/cvavx/include

[root path]/OpenCV/cxcore/include

[root path]/OpenCV/otherlibs/cvcam/include

[root path]/OpenCV/otherlibs/highgui



รูปที่ ข.9 การตั้งค่า Visual Studio C++ ให้ใช้งาน OpenCV ขั้นที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ที่ Show Directories for: เลือก **Source Files** >> add path เป็น

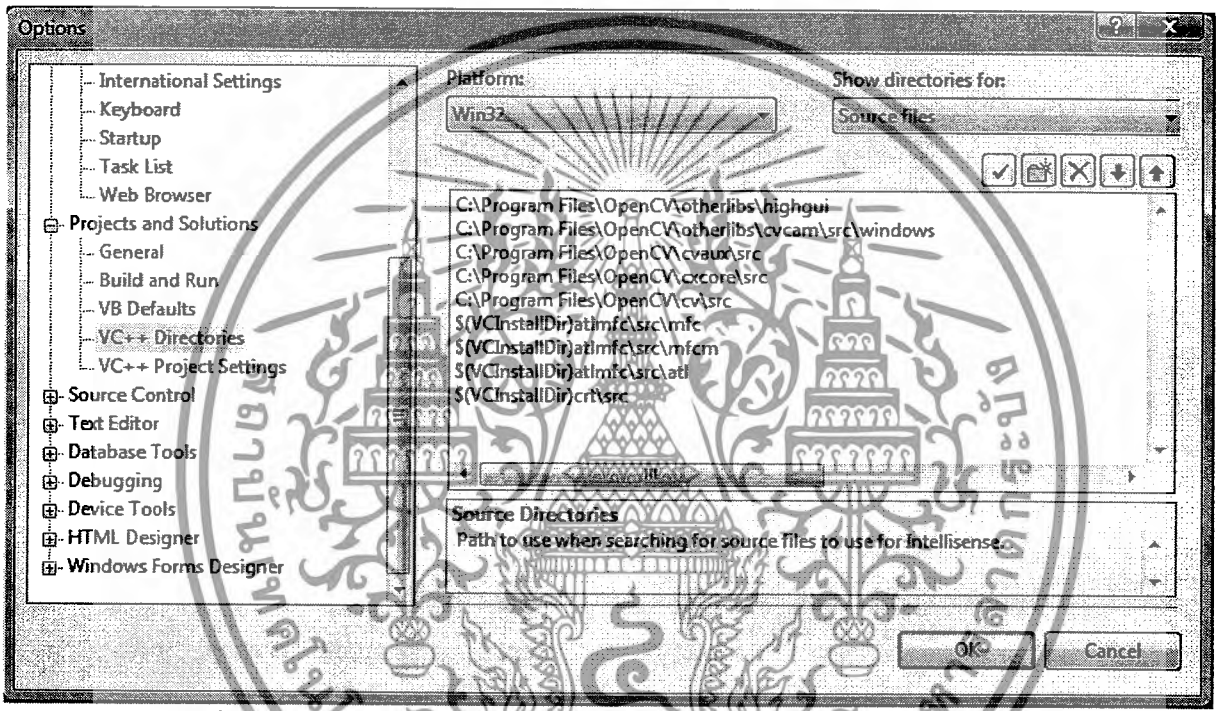
[root path]/OpenCV/cv/src

[root path]/OpenCV/cvavx/src

[root path]/OpenCV/cxcore/src

[root path]/OpenCV/otherlibs/cvcam/src/windows

[root path]/OpenCV/otherlibs/highgui



รูปที่ ข.10 การตั้งค่า Visual Studio C++ ให้ใช้งาน OpenCV ขั้นที่ 3

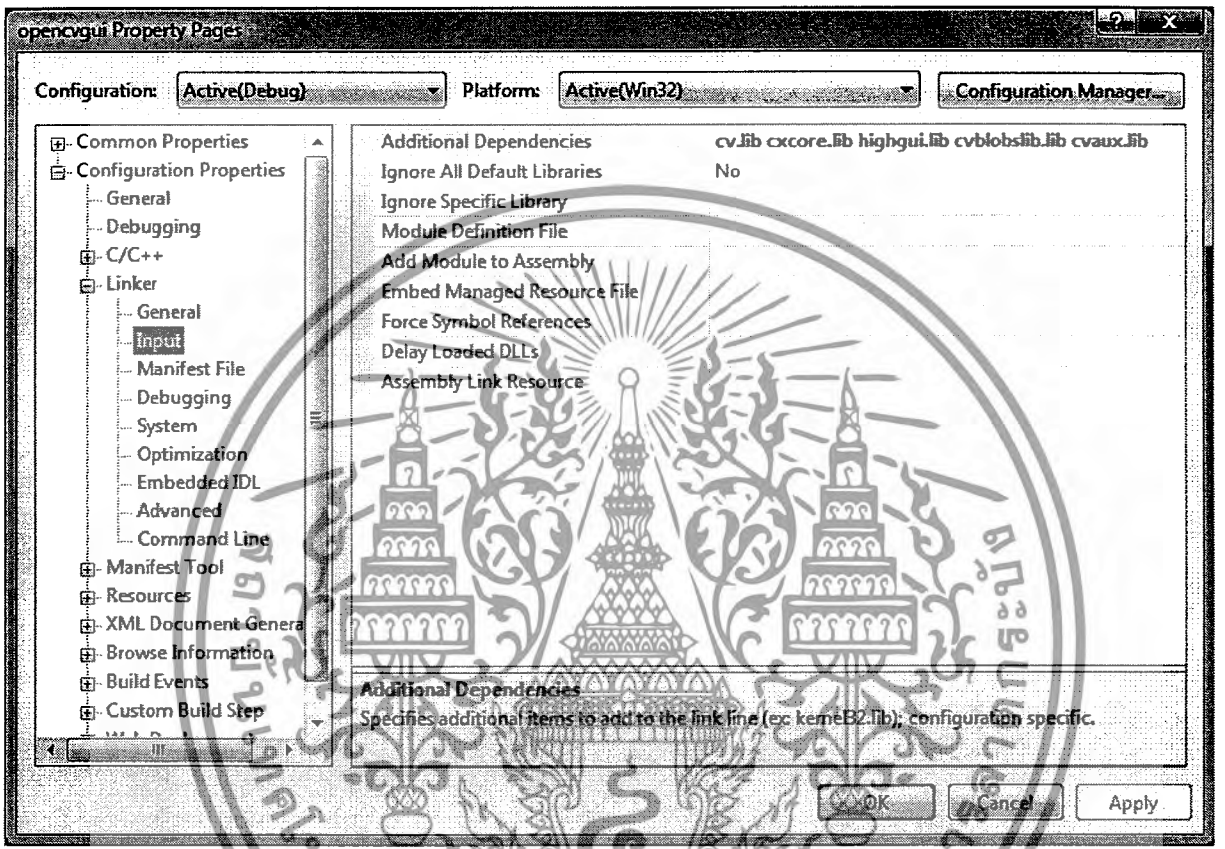
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สร้าง project ที่เป็น C++

- ที่ Project> Properties

ไปที่ Configuration Properties > Linker> Input

- ที่ Additional Dependencies ใส่ **cxcore.lib cv.lib highgui.lib cvaux.lib cvblobslib.lib**



รูปที่ ข.11 การตั้งค่า Visual Studio C++ ให้ใช้งาน OpenCV ขั้นที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการ include ไฟล์ต่างๆของ OpenCV ในโปรแกรมที่จะนำ OpenCV ไปใช้งาน

opencvguiDlg.cpp

(Global Scope)

```
// opencvguiDlg.cpp : implementation file
//

#include "stdafx.h"
#include "opencvgui.h"
#include "opencvguiDlg.h"

#include <highgui.h>
#include <cv.h>
#include <cxcore.h>
#include <cvaux.h>

#include "blob.h"
#include "BlobResult.h"
#include "BlobExtraction.h"
#include "BlobLibraryConfiguration.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

HWND hWnd,hWnd2;

  IplImage* img = 0;
  IplImage* bg = 0;
  IplImage* subimg = 0;
  IplImage* cimgflip = 0;
```

รูปที่ ข.12 ตัวอย่างการ include ไฟล์ต่างๆของ OpenCV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

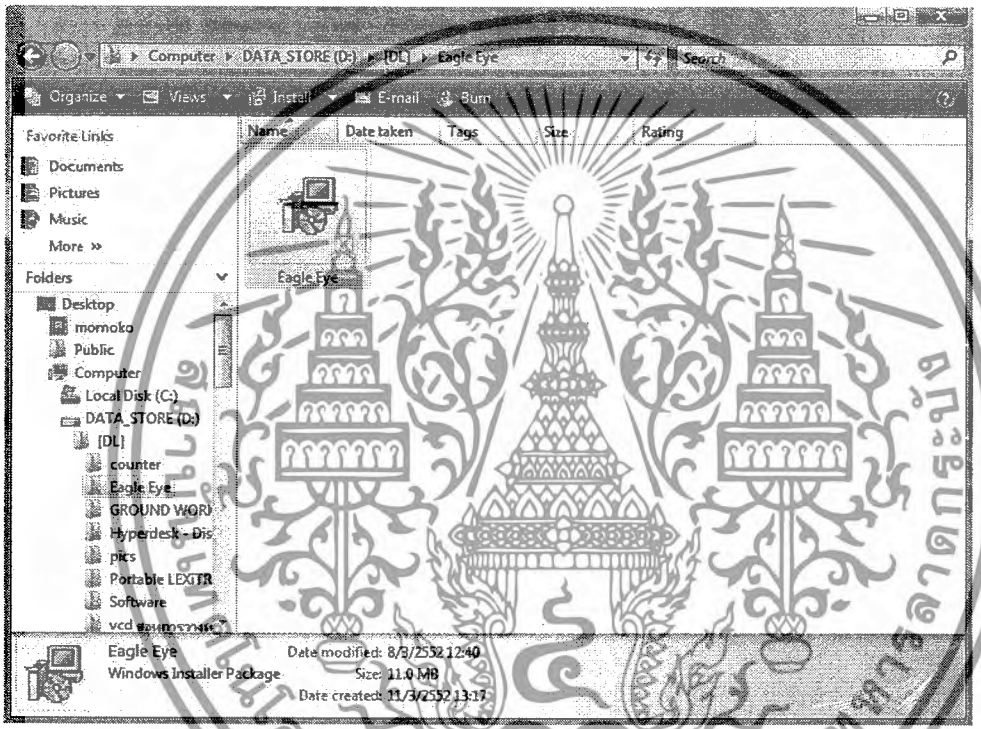
ภาคผนวก ค

โปรแกรม Eagle Eye

ค.1 การติดตั้ง Eagle Eye

ทำการติดตั้งซอฟต์แวร์ Eagle Eye ตามขั้นตอนดังนี้

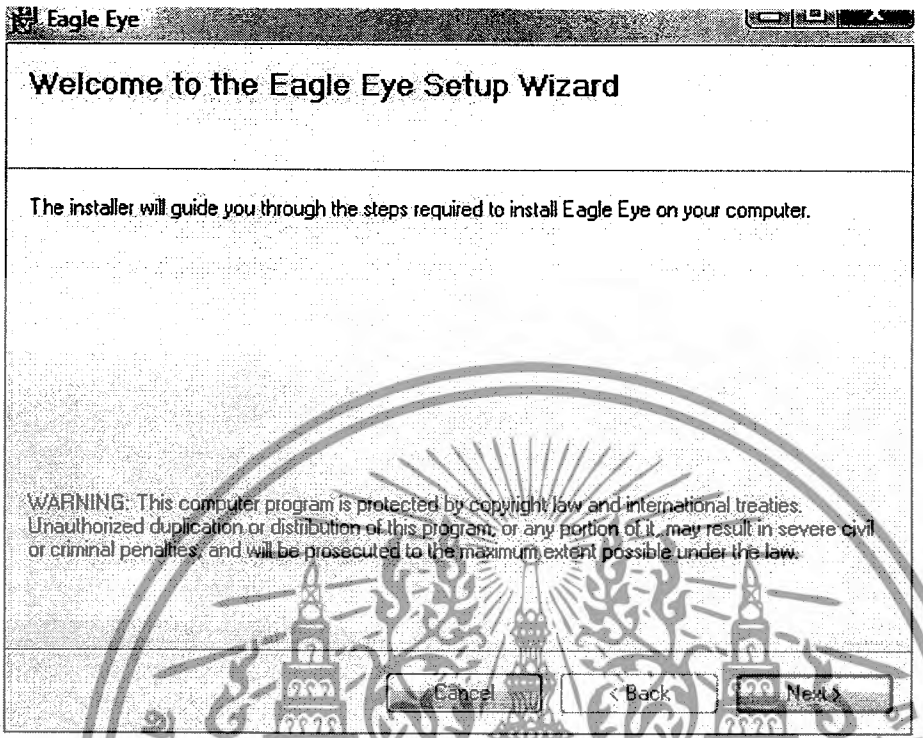
1. ดับเบิ้ลคลิ๊กไฟล์ Eagle Eye.msi



รูปที่ ค.1 อธิบายขั้นตอนการติดตั้ง Eagle Eye ขั้นที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

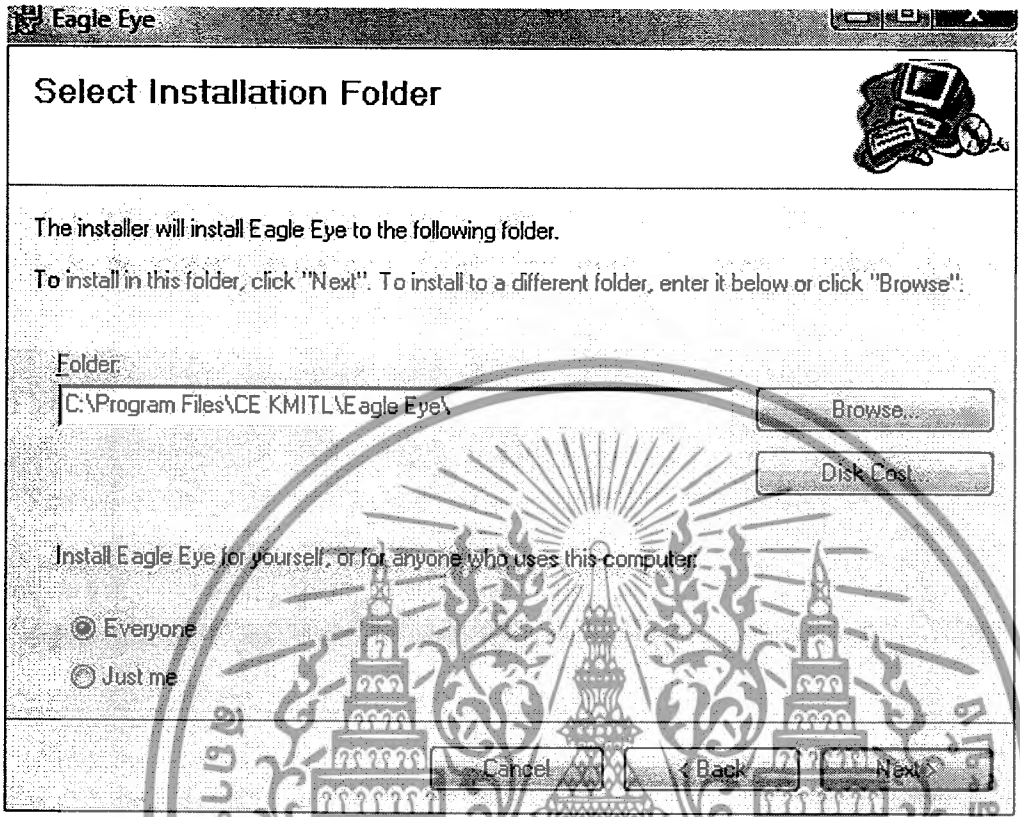
2. กด Next >



รูปที่ ก.2 อธิบายขั้นตอนการติดตั้ง Eagle Eye ขั้นที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

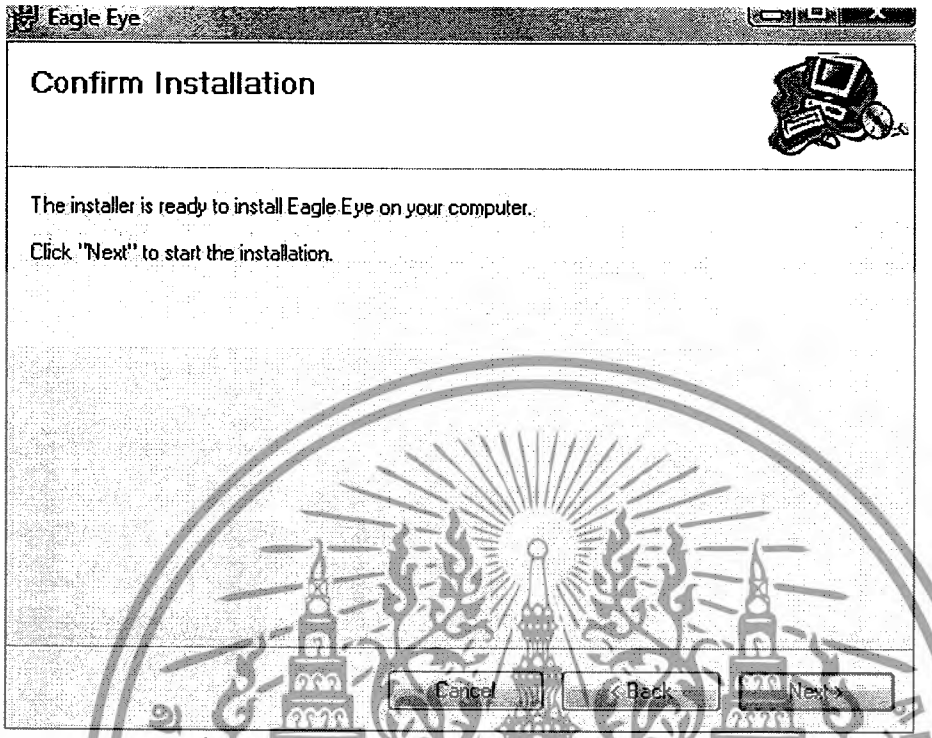
3. เลือก Directory ที่ต้องการติดตั้งซอฟต์แวร์นี้ แล้วกด Next >



รูปที่ ค.3 อธิบายขั้นตอนการติดตั้ง Eagle Eye ขั้นที่ 3

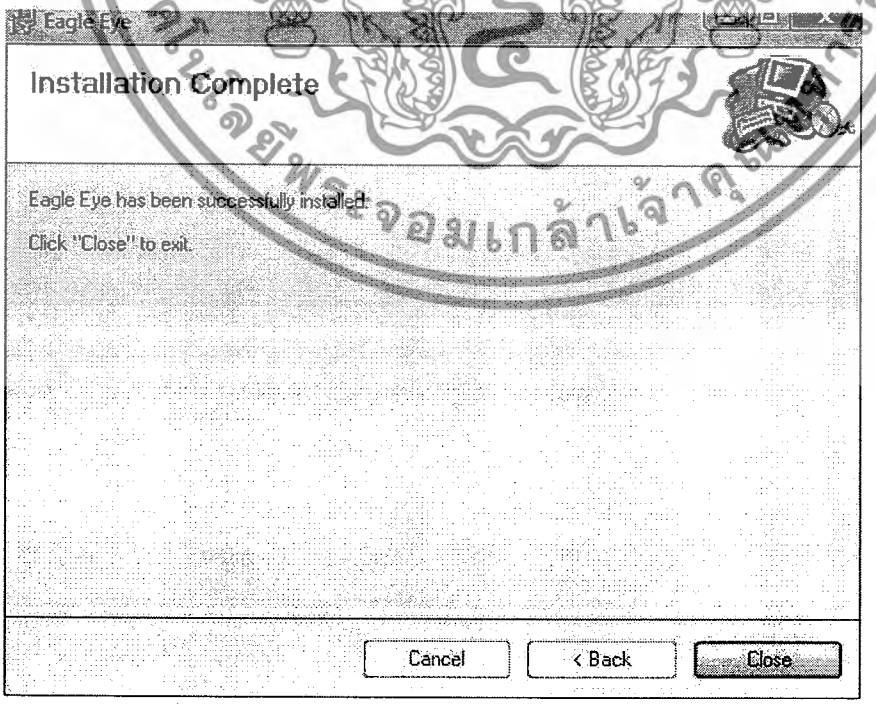
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. กด Next >



รูปที่ ค.4 อธิบายขั้นตอนการติดตั้ง Eagle Eye ขั้นที่ 4

5. รอโปรแกรมติดตั้งซอฟต์แวร์เสร็จ กด Close



รูปที่ ค.5 อธิบายขั้นตอนการติดตั้ง Eagle Eye ขั้นที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้