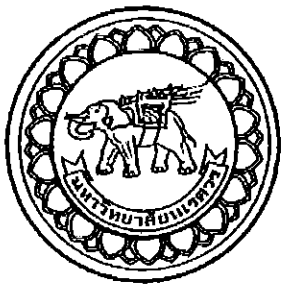


การตรวจสอบผู้บุกรุกโดยใช้การประมวลผลภาพ

Intruder Inspection By Image Processing

นางสาวนิรัชชา สุขอยู่ รหัส 48364784
นางสาวธรชญา อินต๊ะ รหัส 48364982

i509320x e.2
ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ...../...../.....
เลขทะเบียน 5200649
เลขเรียกหนังสือ.....
มหาวิทยาลัยนเรศวร
2551

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2551

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	การตรวจสอบผู้บุกรุกโดยใช้การประมวลผลภาพ
ผู้ดำเนินโครงงาน	นางสาวนิรัชชา สุขอยู่ รหัส 48364784
	นางสาวธรชญา อินต๊ะ รหัส 48364982
อาจารย์ที่ปรึกษา	ดร.พนมขวัญ ริยะมงคล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2551

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

..... ประธานกรรมการ
(ดร.พนมขวัญ ริยะมงคล)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น)

..... กรรมการ
(ดร.ไพศาล มุณีสว่าง)

หัวข้อโครงงาน	การตรวจสอบผู้บุกรุกโดยใช้การประมวลผลภาพ
ผู้ดำเนินโครงงาน	นางสาวนิรัชชา สุขอยู่ รหัส 48364784
	นางสาวธรรชญา อินต๊ะ รหัส 48364982
อาจารย์ที่ปรึกษา	ดร.พนมขวัญ ริยะมงคล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2551

.....

บทคัดย่อ

โครงงานนี้ได้พัฒนาโปรแกรมระบบรักษาความปลอดภัยภายในอาคาร ซึ่งอาศัยหลักการและทฤษฎีการประมวลผลภาพเพื่อแยกรูปภาพของคนจากวัตถุอื่น โดยโปรแกรมนี้ได้นำเอาสัญญาณภาพจากกล้องวิดีโอที่เชื่อมต่อกับเครื่องคอมพิวเตอร์นำมาผ่านกระบวนการแยกแยะวัตถุภายในรูปภาพและใช้ตัวคัดกรองในหลายลักษณะ ซึ่งเมื่อพบวัตถุที่ปรากฏอยู่ในรูปภาพแล้ว โปรแกรมจะทำการค้นหาลักษณะท่าทางของคนโดยหลักการจับคู่กับภาพแม่แบบ และทำการแจ้งเตือนให้ผู้ใช้งานได้ทราบโดยส่งเสียงเตือนออกทางลำโพง

Project Title	Intruder Inspection By Image Processing
Name	Miss. Nirachcha Sukyu ID. 48364784
	Miss. Thorrachaya Inta ID. 48364982
Project Advisor	Panomkhawn Riyamongkol, Ph.D
Major	Computer Engineering.
Department	Electrical and Computer Engineering.
Academic Year	2008

ABSTRACT

This project develops a program to detect a human inside the building. This project applyied the image processing theory to analyse human shape for other object in the building. The developed program gets a video stream from video camera that connect to the computer an analyse , then filter each object in images from the video stream in many aspects. This program can search and detect human form by using Template Matching theory. Once the program detected human, it will alert the user by sound.

กิตติกรรมประกาศ

โครงงานนี้ประสบความสำเร็จได้ด้วยดีด้วยความกรุณาจากอาจารย์ที่ปรึกษาโครงงานและคณะกรรมการทุกท่าน ขอขอบพระคุณ ดร.พนมขวัญ ริยะมงคล ซึ่งเป็นที่ปรึกษาโครงงานนี้ที่คอยให้คำแนะนำและคำปรึกษามาโดยตลอด อีกทั้ง ผู้ช่วยศาตราจารย์ ดร.สุชาติ แย้มแม่นและ ดร.ไพศาล มุณีสว่าง ที่ให้คำแนะนำในการปรับปรุงโครงงานให้ดียิ่งขึ้น จนทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

ในโอกาสนี้ทางคณะผู้จัดทำโครงงานจึงขอขอบพระคุณทุกๆ ท่านที่มีส่วนร่วมในการทำโครงงานนี้ ตลอดจนผู้คิดค้นทฤษฎีต่างๆ ที่โครงงานฉบับนี้ได้นำความรู้ที่ได้มาพัฒนาระบบจนประสบความสำเร็จ คณะผู้จัดทำขอบกราบขอบพระคุณไว้ ณ ที่นี้

นางสาวนิรัชชา สุขอยู่
นางสาวธรชญา อินต๊ะ

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ก

บทคัดย่อภาษาอังกฤษ..... ข

กิตติกรรมประกาศ..... ค

สารบัญ..... ง

สารบัญตาราง..... ฉ

สารบัญรูป..... ช

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของโครงงาน..... 1

1.2 วัตถุประสงค์ของโครงงาน..... 1

1.3 ขอบข่ายของโครงงาน..... 2

1.4 ขั้นตอนการดำเนินงาน..... 2

1.5 แผนการดำเนินงาน..... 2

1.6 ผลที่คาดว่าจะได้รับ..... 3

1.7 งบประมาณที่ใช้..... 3

บทที่ 2 หลักการและทฤษฎีที่ใช้

2.1 รูปร่างของภาพ (Image Shape) 4

2.2 มาตรฐานของสี..... 5

2.3 การค้นหาวัตถุจากรูปภาพโดยหลักการผลต่างของเมทริกซ์..... 6

2.4 ทฤษฎีการประมวลผลภาพ..... 9

2.5 ทฤษฎีการตรวจจับการเคลื่อนไหว..... 10

2.6 การตัดแบ่งส่วนรูปภาพ (Image Segmentation) 11

2.7 การจับคู่แม่แบบ (Template Matching) 14

สารบัญ (ต่อ)

หน้า

บทที่ 3 ขั้นตอนการดำเนินงาน

3.1 ภาพโดยรวมของระบบ..... 17

3.2 การออกแบบโปรแกรม..... 18

3.3 การทำงานของโปรแกรม (Flow Chart) 22

3.4 ออกแบบหน้าต่างการใช้งานของโปรแกรม..... 23

บทที่ 4 ผลการทดลอง

4.1 ขั้นตอนการทำงานของโปรแกรม..... 24

4.2 ผลการทดลองการตั้งค่าเทรชโฮลในการค้นหาผู้บุกรุก..... 25

4.3 ผลการทดลองการทำงานของโปรแกรมในส่วนของการจับคู่แม่แบบ..... 28

4.4 ผลการทดลองการใช้โปรแกรมตรวจสอบผู้บุกรุก..... 32

4.5 วิเคราะห์ผลการทดลอง..... 51

บทที่ 5 บทสรุป

5.1 สรุปการทดลอง..... 53

5.2 ปัญหาและแนวทางแก้ไข..... 54

เอกสารอ้างอิง..... 55

ภาคผนวก

ภาคผนวก ก การใช้งาน OpenCV เบื้องต้น..... 56

ภาคผนวก ข ฟังก์ชัน OpenCV ที่ใช้ในโปรเจค..... 67

ประวัติผู้เขียนโครงงาน..... 71

สารบัญตาราง

ตารางที่ หน้า

1.1 ขั้นตอนการทำงาน..... 2

3.1 แสดงเทมเพลตต้นแบบที่ใช้ในการเปรียบเทียบที่มีอยู่ในโปรแกรม..... 21

4.1 แสดงผลค่า R(x,y) ในแต่ละเทมเพลตของท่าทางที่ 1..... 28

4.2 แสดงผลค่า R(x,y) ในแต่ละเทมเพลตของท่าทางที่ 2..... 29

4.3 แสดงผลค่า R(x,y) ในแต่ละเทมเพลตของท่าทางที่ 3..... 30

4.4 แสดงผลค่า R(x,y) ในแต่ละเทมเพลตของท่าทางที่ 4..... 31

4.5 แสดงผลการทดลองโปรแกรมในสถานการณ์ต่างๆ..... 50

สารบัญรูป

รูปที่ หน้า

2.1 ภาพแสดงการแปลงภาพอ้างอิงมาเป็นเมทริกซ์อ้างอิง..... 7

2.2 ภาพแสดงการแปลงภาพเปรียบเทียบมาเป็นเมทริกซ์เปรียบเทียบ..... 7

2.3 ภาพแสดงการลบกันของเมทริกซ์อ้างอิงกับเมทริกซ์เปรียบเทียบ..... 7

2.4 ภาพแสดงผลต่างของ 2 เมทริกซ์..... 8

2.5 ภาพการแสดงภาพที่สร้างจากผลต่างของเมทริกซ์..... 8

2.6 ภาพปกติ..... 10

2.7 ภาพ Grayscale..... 10

2.8 ภาพการทำ Threshold..... 10

2.9 ทำการเทรชโฮลกับภาพระดับเทา..... 14

2.10 Histogram ของภาพแบคทีเรียด้วยค่าเทรชโฮลเท่ากับ 102..... 14

2.11 แสดงการเปรียบเทียบเทมแพลตแม่แบบและเทมเพลตที่นำมาเปรียบเทียบ..... 15

3.1 แสดงภาพโดยรวมของระบบ..... 17

3.2 บล็อกไดอะแกรมแสดงระบบการทำงานโดยรวม..... 18

3.3 บล็อกไดอะแกรมแสดงส่วนขยายของการรับภาพจากกล้อง..... 18

3.4 บล็อกไดอะแกรมแสดงการทำ Image segmentation..... 19

3.5 แสดงการลบเมทริกซ์ของ 2 เฟรม..... 19

3.6 แสดงเมริกที่ได้จากการลบกันของเมทริกซ์ A และ B..... 20

3.7 บล็อกไดอะแกรมแสดงการค้นหาวัตถุและวิเคราะห์วัตถุ 20

3.8 แสดงแผนภาพการทำงานของโปรแกรม..... 22

3.9 แสดงหน้าต่างการใช้งาน..... 23

4.1 แสดงรูปภาพเริ่มต้นที่ใช้ในการเปรียบเทียบและสถานะเริ่มต้น..... 24

4.2 แสดงภาพคนหรือวัตถุที่แยกออกจากพื้นหลังและกราฟแสดงค่าเทรชโฮลของคนหรือวัตถุ..... 25

4.3 แสดงโปรแกรมแจ้งเตือนมีผู้บุกรุก..... 25

4.4 แสดงภาพที่ได้จากการกำหนดค่าเทรชโฮลเท่ากับ 20 ทั้ง R, G และ B 26

4.5 แสดงภาพที่ได้จากการกำหนดค่าเทรชโฮลเท่ากับ 30 ทั้ง R, G และ B 26

4.6 แสดงภาพที่ได้จากการกำหนดค่าเทรชโฮลเท่ากับ 50 ทั้ง R , G และ B 27

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.7	แสดงภาพที่ได้จากการกำหนดค่าเทรชโฮลเท่ากับ 100 ทั้ง R , G และ B	27
4.8	ภาพที่รับจากกล้องในลักษณะที่ 1.....	28
4.9	ภาพที่รับจากกล้องในลักษณะที่ 2.....	29
4.10	ภาพที่รับจากกล้องในลักษณะที่ 3.....	30
4.11	ภาพที่รับจากกล้องในลักษณะที่ 4.....	31
4.12	แสดงหน้าต่างของโปรแกรมที่ไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้องในช่วงเวลากลางวันระยะตรวจสอบ 5 เมตร.	32
4.13	แสดงหน้าต่างของโปรแกรมที่ไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้องในช่วงเวลากลางวันระยะตรวจสอบ 7 เมตร.....	33
4.14	แสดงหน้าต่างของโปรแกรมที่ไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้องในช่วงเวลากลางวันระยะตรวจสอบ 10 เมตร.....	34
4.15	แสดงหน้าต่างของโปรแกรมที่คนเคลื่อนที่ผ่านกล้องในช่วงเวลากลางวันระยะตรวจสอบ 5 เมตร.....	34
4.16	แสดงหน้าต่างของโปรแกรมที่คนเคลื่อนที่ผ่านกล้องในช่วงเวลากลางวันระยะตรวจสอบ 7 เมตร.....	35
4.17	แสดงหน้าต่างของโปรแกรมที่คนเคลื่อนที่ผ่านกล้องในช่วงเวลากลางวันระยะตรวจสอบ 10 เมตร.....	36
4.18	แสดงหน้าต่างของโปรแกรมที่มีวัตถุเคลื่อนที่ผ่านกล้องในช่วงเวลากลางวันระยะตรวจสอบ 5 เมตร.....	36
4.19	แสดงหน้าต่างของโปรแกรมที่มีวัตถุเคลื่อนที่ผ่านกล้องในช่วงเวลากลางวันระยะตรวจสอบ 7 เมตร.....	37
4.20	แสดงหน้าต่างของโปรแกรมที่มีวัตถุเคลื่อนที่ผ่านกล้องในช่วงเวลากลางวันระยะตรวจสอบ 10 เมตร.....	38
4.21	แสดงหน้าต่างของโปรแกรมที่ไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้องในช่วงเวลากลางคืน ระยะตรวจสอบ 5 เมตร.....	38
4.22	แสดงหน้าต่างของโปรแกรมที่ไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้องในช่วงเวลากลางคืน ระยะตรวจสอบ 7 เมตร.....	39

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.23 แสดงหน้าต่างของโปรแกรมที่ไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้องในช่วงเวลากลางคืน ระยะตรวจสอบ 7 เมตร..... 40

4.24 แสดงหน้าต่างของโปรแกรมที่มีคนเคลื่อนที่ผ่านกล้องในช่วงเวลากลางคืน ระยะตรวจสอบ 5 เมตร..... 40

4.25 แสดงหน้าต่างของโปรแกรมที่ไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้องในช่วงเวลากลางคืน ระยะตรวจสอบ 7 เมตร..... 41

4.26 แสดงหน้าต่างของโปรแกรมที่ไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้องในช่วงเวลากลางคืน ระยะตรวจสอบ 10 เมตร..... 42

4.27 แสดงหน้าต่างของโปรแกรมที่มีวัตถุเคลื่อนที่ผ่านกล้องในช่วงเวลากลางคืน ระยะตรวจสอบ 5 เมตร..... 42

4.28 แสดงหน้าต่างของโปรแกรมที่มีวัตถุเคลื่อนที่ผ่านกล้องในช่วงเวลากลางคืน ระยะตรวจสอบ 7 เมตร..... 43

4.29 แสดงหน้าต่างของโปรแกรมที่มีวัตถุเคลื่อนที่ผ่านกล้องในช่วงเวลากลางคืน ระยะตรวจสอบ 10 เมตร..... 44

4.30 แสดงหน้าต่างของโปรแกรมที่มีคนเคลื่อนที่ผ่านกล้องในช่วงเวลากลางคืน แต่ภาพต้นแบบเป็นช่วงเวลากลางวันและระยะตรวจสอบ 5 เมตร..... 44

4.31 แสดงหน้าต่างของโปรแกรมที่มีคนเคลื่อนที่ผ่านกล้องในช่วงเวลากลางคืน แต่ภาพต้นแบบเป็นช่วงเวลากลางวันและระยะตรวจสอบ 7 เมตร..... 45

4.32 แสดงหน้าต่างของโปรแกรมที่มีคนเคลื่อนที่ผ่านกล้องในช่วงเวลากลางคืน แต่ภาพต้นแบบเป็นช่วงเวลากลางวันและระยะตรวจสอบ 10 เมตร..... 46

4.33 แสดงหน้าต่างของโปรแกรมที่มีคนเคลื่อนที่ผ่านกล้องในช่วงเวลากลางวัน แต่ภาพต้นแบบเป็นช่วงเวลากลางคืนและระยะตรวจสอบ 5 เมตร..... 47

4.34 แสดงหน้าต่างของโปรแกรมที่มีคนเคลื่อนที่ผ่านกล้องในช่วงเวลากลางวัน แต่ภาพต้นแบบเป็นช่วงเวลากลางคืนและระยะตรวจสอบ 7 เมตร..... 47

4.35 แสดงหน้าต่างของโปรแกรมที่มีคนเคลื่อนที่ผ่านกล้องในช่วงเวลากลางวัน แต่ภาพต้นแบบเป็นช่วงเวลากลางคืนและระยะตรวจสอบ 10 เมตร..... 48

4.36 แสดงหน้าต่างของโปรแกรมที่มีคนเดินผ่านกล้องแบบไม่เต็มตัว ในช่วงเวลากลางวัน..... 49

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.37 แสดงหน้าต่างของโปรแกรมที่มีคนเดินผ่านกล้องแบบไม่เต็มตัว ในช่วงเวลากลางคืน..... 49

ก.1 แสดงไฟล์ของโปรแกรมที่ใช้ในการติดตั้ง..... 57

ก.2 แสดงหน้าต่างการติดตั้งของโปรแกรม..... 57

ก.3 แสดงหน้าต่างการติดตั้งของโปรแกรมที่ให้ทำการตกลง..... 57

ก.4 แสดงหน้าต่างให้เลือกไดร์ที่จะทำการติดตั้ง..... 58

ก.5 แสดงหน้าต่างให้เลือกโฟล์เดอร์..... 58

ก.6 แสดงหน้าต่างให้ทำการเพิ่ม path..... 58

ก.7 แสดงหน้าต่างในการเริ่มการติดตั้ง..... 59

ก.8 แสดงหน้าต่างการติดตั้งสำเร็จ..... 59

ก.9 แสดงหน้าต่างไดเรกทอรี่ที่ได้กำหนด..... 60

ก.10 แสดงหน้าต่างในการเปิดโปรแกรม..... 60

ก.11 แสดงหน้าต่างไดเรกทอรี่ที่ได้กำหนด..... 60

ก.12 แสดงหน้าต่างการ Build OpenCV..... 61

ก.13 แสดงหน้าต่าง Status bar..... 61

ก.14 แสดงหน้าต่างในการตั้งค่าโปรแกรม..... 61

ก.15 แสดงหน้าต่างในการตั้งค่าโปรแกรมโดยเลือก Project and Solution..... 62

ก.16 แสดงหน้าต่างในการตั้งค่า Path ของโปรแกรม..... 62

ก.17 แสดงหน้าต่างในการตั้งค่า Include files..... 62

ก.18 แสดงหน้าต่างในการตั้งค่า Library files..... 63

ก.19 แสดงหน้าต่างในการทดสอบฟังก์ชัน..... 63

ก.20 แสดงหน้าต่างในการเริ่ม Debugging..... 63

ก.21 แสดงหน้าต่างทดสอบการทำงานของฟังก์ชัน..... 64

ก.22 แสดงหน้าต่างการตรวจสอบ System Property..... 65

ก.23 แสดงหน้าต่างการเพิ่ม Path..... 65

ก.24 แสดงหน้าต่างการตั้งค่า Project Property..... 66

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันปัญหาด้านอาชญากรรมและการโจรกรรมยังเป็นที่พบเห็นและเกิดขึ้นบ่อยครั้ง ซึ่งจากเหตุการณ์ดังกล่าวทำให้เกิดความเสียหายแก่ผู้ประสบเหตุไม่ว่าจะเป็นทางด้านทรัพย์สินหรือชีวิต ดังนั้น การพัฒนาระบบรักษาความปลอดภัยจึงเป็นเรื่องที่ควรได้รับการพิจารณาและถูกพัฒนาเพื่อแก้ปัญหาเหล่านี้เป็นอย่างยิ่ง ในปัจจุบันระบบรักษาความปลอดภัยมีหลากหลายระบบซึ่งโดยส่วนใหญ่นั้นยังใช้รูปแบบของระบบกล้องวงจรปิดซึ่งมีเจ้าหน้าที่รักษาความปลอดภัยนั่งเฝ้าเหตุการณ์ผ่านทางมอนิเตอร์รับภาพจากกล้องในห้องควบคุม โดยในทางปฏิบัติทั่วไปนั้นภายในห้องควบคุมจะมีมอนิเตอร์รับภาพจากกล้องจำนวนหลายตัวแต่อาจจะมีผู้ควบคุมและเฝ้าสังเกตทางมอนิเตอร์เพียง 1-2 คนเท่านั้น อาจจะมีเหตุการณ์ที่ผู้ควบคุมเหล่านั้นไม่สามารถเห็นเหตุการณ์ได้โดยทั้งหมดหรืออาจจะเกิดจากความประมาทและเลินเล่อซึ่งทำให้เกิดความเสียหายตามมาในที่สุด

จากเหตุผลดังกล่าวทำให้เกิดแนวคิดที่จะพัฒนาระบบการมองเห็นสำหรับการรักษาความปลอดภัยโดยสามารถทำงานได้อย่างอัตโนมัติไร้ผู้ควบคุม โดยใช้ระบบคอมพิวเตอร์ซึ่งเป็นเทคโนโลยีสมัยใหม่ที่มีการพัฒนาอย่างต่อเนื่องในทุกวันนี้และใช้ความรู้เชิงวิศวกรรมและการประยุกต์จากการประมวลผลภาพ ซึ่งสามารถนำมาประยุกต์ใช้กับการรักษาความปลอดภัยประเภทนี้ได้ เนื่องจากการเลือกใช้ข้อมูลประเภทสัญญาณภาพนั้นมีข้อดี คือ สามารถเห็นรายละเอียดของวัตถุและบุคคลได้ชัดเจนกว่าระบบการตรวจจับประเภทอื่น (อินฟราเรด, เทอร์มอลเซนเซอร์ ฯลฯ) และสามารถสร้างให้ระบบดังกล่าวตัดสินใจได้ว่าเป็นวัตถุหรือบุคคลแปลกปลอมได้จากรายละเอียดทั้งสีและรูปร่างได้ โดยทำการตัดสินจากรูปร่างที่กำหนดไว้ล่วงหน้าจากข้อมูลที่เก็บไว้ก่อนหน้านั้น จึงทำให้สามารถใช้ระบบรักษาความปลอดภัยนี้มาช่วยเหลือผู้ควบคุมในห้องมอนิเตอร์โดยจะสามารถทำงานได้ตลอดทั้ง 24 ชั่วโมง ซึ่งต่างกับมนุษย์ที่ต้องการการพักผ่อนแต่ระบบนี้สามารถทำงานได้อย่างต่อเนื่องและมีความถูกต้อง

1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาและพัฒนาระบบรักษาความปลอดภัยอัตโนมัติจากกล้องวีดีโอ

1.2.2 เพื่อศึกษาพัฒนาระบบการตัดสินใจและการวิเคราะห์ข้อมูลจากภาพ

1.2.3 เพื่อออกแบบอัลกอริทึมที่ใช้ในการตรวจสอบผู้บุกรุกโดยการประมวลผลภาพ

1.2.4 เพื่อพัฒนาระบบการแจ้งเตือนเมื่อมีผู้บุกรุกโดยการตรวจจับภาพจากกล้องวีดีโอ

1.3 ขอบข่ายของโครงงาน

1.3.1 ศึกษาทฤษฎีพื้นฐานของการประมวลผลภาพ

1.3.2 เพื่อให้ได้โปรแกรมพัฒนาระบบการมองเห็นจากการประมวลผลภาพ

1.3.3 เพื่อให้ได้โปรแกรมพัฒนาระบบการจดจำและการตัดสินใจ

1.3.4 เพื่อให้ได้โปรแกรมพัฒนาระบบการแจ้งเตือนจากกล้องวีดีโอ

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาและเก็บรวบรวมข้อมูลที่เกี่ยวข้องกับหัวข้อโครงงาน

1.4.2 ออกแบบอัลกอริทึมที่ใช้ในระบบการตรวจสอบผู้บุกรุก

1.4.3 พัฒนาระบบที่ใช้ในการตรวจสอบผู้บุกรุกโดยการประมวลผลภาพ

1.4.4 ทดสอบและปรับปรุงโปรแกรมที่ใช้ในการตรวจสอบผู้บุกรุก

1.4.5 ตรวจสอบและปรับปรุงแก้ไขโปรแกรมที่ใช้ในการตรวจสอบผู้บุกรุก

1.4.6 จัดทำคู่มือโครงงาน

1.5 แผนการดำเนินงาน

ตารางที่ 1.1 ขั้นตอนการดำเนินงาน

กิจกรรม	ปี 2551						ปี 2552	
	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.
1. ศึกษาและเก็บรวบรวมข้อมูลที่เกี่ยวข้องกับหัวข้อโครงงาน								
2. ออกแบบอัลกอริทึมที่ใช้ในระบบการตรวจสอบผู้บุกรุก								
3. พัฒนาระบบที่ใช้ในการตรวจสอบผู้บุกรุก								
4. ทดสอบและปรับปรุงโปรแกรมที่ใช้ในการตรวจสอบผู้บุกรุก								
5. ตรวจสอบและปรับปรุงแก้ไขโปรแกรมที่ใช้ในการตรวจสอบผู้บุกรุก								
6. จัดทำคู่มือโครงงาน								

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 ได้รับความรู้เกี่ยวกับ

- ทฤษฎีพื้นฐานการประมวลผลภาพและทฤษฎีอื่นๆ ที่เกี่ยวข้องกับหัวข้อโครงงาน
- อัลกอริทึมที่เกี่ยวข้องในการตรวจสอบผู้บุกรุกโดยการประมวลผลภาพ

1.6.2 สามารถที่จะเอาโปรแกรมที่ได้ไปใช้ร่วมกับระบบรักษาความปลอดภัยทั่วไปได้

1.6.3 สามารถนำความรู้ที่ได้ไปใช้งานในอนาคตได้

1.6.4 ได้ระบบที่เพิ่มประสิทธิภาพการป้องกันภัยจากกล้องวีดีโอ ซึ่งอาจจะช่วยลดปัญหาด้านอาชญากรรมและการโจรกรรม

1.7 งบประมาณที่ใช้

1.7.1 ค่าใช้จ่ายในการซื้อหนังสือ	1,000	บาท
1.7.2 ค่าใช้จ่ายในการทำรายงาน	500	บาท
1.7.3 ค่าใช้จ่ายเบ็ดเตล็ด	500	บาท
รวมทั้งสิ้น	2,000	บาท

(สองพันบาทถ้วน)

หมายเหตุ ขออนุมัติถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎีที่ใช้

2.1 รูปร่างของภาพ (Image Shape) [1]

วัตถุที่มีอยู่ตามธรรมชาติและที่มนุษย์สร้างขึ้น มีรูปร่างที่แตกต่างกันไปทั้งที่เป็นรูปทรงเรขาคณิตและไม่เป็นรูปทรงเรขาคณิต ในศาสตร์ของการประมวลผลภาพนั้นการกำหนดขอบเขตของภาพทุกภาพให้อยู่ในรูปสี่เหลี่ยม (Rectangular image model) เป็นวิธีที่นิยมใช้กันมากที่สุด เนื่องจากทำให้การอ่านภาพ การจัดเก็บข้อมูลภาพในหน่วยความจำ และการแสดงภาพออกทางอุปกรณ์ต่างๆ เป็นไปได้อย่างมีประสิทธิภาพ โดยการเก็บข้อมูลภาพลงหน่วยความจำของคอมพิวเตอร์สามารถทำได้โดยการจองหน่วยความจำของเครื่องไว้ในรูปของตัวแปรอะเรย์ (array) โดยค่าในแต่ละช่องของอะเรย์แสดงถึงคุณสมบัติของจุดภาพ (pixel) และตำแหน่งของช่องอะเรย์ก็เป็นตัวกำหนดตำแหน่งของจุดภาพ

สมมุติให้ Image เป็นตัวแปรแบบอะเรย์ขนาด M x N (M แถว และ N คอลัมน์) ที่ใช้เก็บภาพขนาด M x N จุด (M จุดในแนวนอน และ N จุดในแนวตั้ง) ค่าสี (หรือความสว่าง ในกรณีที่เป็นภาพ grey level) ของจุดภาพในแถวที่ 5 คอลัมน์ที่ 4 จะตรงกับค่าของ Image (5 , 4) จะเห็นว่าเราใช้ตำแหน่งของจุดภาพทั้งสองแกนเป็นตัวชี้ค่าข้อมูลในอะเรย์

จากการใช้หน่วยความจำเพื่อการเก็บภาพในลักษณะที่กล่าวมา เนื้อที่ในการเก็บภาพสามารถคำนวณได้จาก M x N x g เมื่อ g เป็นจำนวนเต็มที่แทนจำนวนบิตของข้อมูลในแต่ละจุดภาพ ตัวอย่างถ้า g มีค่าเท่ากับ 8 บิต จะสามารถเก็บความแตกต่างของระดับสีที่เป็นไปสูงสุด 256 ระดับ ค่า M และ N จะเป็นตัวบอกถึงความละเอียดของภาพ สำหรับคอมพิวเตอร์ทั่วไปในระบบ VGA (Video Graphic Array) จะมีขนาด 640 x 480, 800 x 600 และ 1024 x 768 จุด เป็นต้น การกำหนดความละเอียดจะขึ้นอยู่กับงานที่จะใช้ ซึ่งในงานบางอย่างใช้ความละเอียดของสีเพียง 30 x 50 จุด ก็พอแล้ว แต่ในงานบางชนิดใช้ความละเอียดของสีถึง 1000 x 1000 จุด ก็ยังไม่พอ

ปกติแล้วในการเก็บข้อมูลภาพโดยเครื่องมือต่างๆ จะเก็บตามมาตรฐานของโทรทัศน์ซึ่งมีอัตราส่วน x ต่อ y เท่ากับ 4 : 3 สำหรับเครื่องมือเก็บข้อมูลภาพที่ไม่เป็นไปตามอัตราส่วน 4 : 3 เมื่อนำภาพนี้ไปแสดงในจอภาพมาตรฐานจะทำให้ภาพที่แสดงนั้นมีขนาดของจุดภาพไม่เป็นสี่เหลี่ยมจัตุรัส เช่น ในบางระบบอาจจะใช้ความละเอียดในการแสดงเท่ากับ 640 x 512 ซึ่งจะทำให้ขนาดของจุดภาพที่ได้มีขนาดของด้านกว้างมีความยาวมากกว่าด้านสูง

จำนวนสีสูงสุดที่เป็นไปได้ของแต่ละจุดภาพขึ้นอยู่กับจำนวนบิตที่ใช้เมื่อมีการกำหนดให้ขนาดของบิตต่อจุดมากขึ้น ซึ่งจะทำให้จำนวนของสีมากขึ้นด้วย ตัวอย่างเช่น

1 บิต = 2^1 = 2 สี

2 บิต = 2^2 = 4 สี

4 บิต = 2^4 = 16 สี

8 บิต = 2^8 = 256 สี

16 บิต = 2^{16} = 65536 สี เป็นต้น

สำหรับการแสดงข้อมูลภาพที่มีขนาด 1 บิต และ 8 บิตนั้นจะมีการทำงานที่จะใกล้เคียงกัน เนื่องจากหน่วยประมวลผลจะไม่สามารถจัดการกับข้อมูลที่เป็นบิตเดี่ยวๆ ได้ ดังนั้นในการแสดงข้อมูลออกทางจอภาพตัวโปรเซสเซอร์จะทำการก็อปปี้ข้อมูลทั้ง 8 บิต (1 Byte) ส่งให้กับจอภาพซึ่งในกรณีที่จุดภาพมีขนาด 1 บิต เมื่อโปรเซสเซอร์จะทำงานกับบิตแรกที่ต้องการแล้วก็จะทำการคัดลอกข้อมูลชุดใหม่ทันทีโดยที่ไม่เกี่ยวกับข้อมูลอีก 7 บิตที่เหลือส่วนในกรณีจุดภาพมีขนาด 8 บิต โปรเซสเซอร์จะทำการก็อปปี้ข้อมูลจุดใหม่ก็ต่อเมื่อโปรเซสเซอร์ทำงานกับทุกบิตแล้ว

ตัวอย่างสำหรับระบบที่มีความละเอียดเท่ากับ 800 x 600 และมีขนาด 16 บิตต่อพิกเซล จะสามารถแสดงสีได้ทั้งหมด 65536 ระดับ และต้องใช้เนื้อที่ในการเก็บเท่ากับ 800 x 600 x 16 บิต

2.2 มาตรฐานของสี [2]

มาตรฐานของสีที่ใช้อยู่ในปัจจุบันมีอยู่หลายระบบด้วยกัน โดยจะขึ้นอยู่กับการนำไปใช้งาน แต่โดยทั่วไปแล้วทุกมาตรฐานจะมีแนวคิดเดียวกัน คือ การแทนจุดสีด้วยจุดที่อยู่ภายในสเปส 3 มิติ โดยจะมีแกนอ้างอิงสำหรับจุดสีนั้นในสเปส ซึ่งแต่ละแกนจะมีความเป็นอิสระต่อกัน ตัวอย่างเช่น ในระบบ RGB จะมีแกนสี คือ แกนสีแดง แกนสีเขียว และแกนสีน้ำเงิน ในระบบ HLS จะมีแกนเป็นค่าสี (hue) ความสว่าง (lightness) และความบริสุทธิ์ของสี (saturation)

ตัวอย่างระบบสีที่นิยมใช้กันได้แก่ ระบบ RGB , HSV (Hue Saturation Value) และ HLS (Hue Lightness Saturation)

2.2.1 ระบบสี RGB

ระบบสี RGB เป็นระบบสีที่เกิดจากการรวมกันของแสงสีแดง แสงสีเขียว และแสงสีน้ำเงิน โดยมีการรวมกันในลักษณะที่เพิ่มหรือบวก ซึ่งโดยปกติจะนำไปใช้ในจอภาพแบบ CRT (Cathode ray tube) ในการใช้งานระบบสี RGB ยังมีการสร้างมาตรฐานที่แตกต่างกัน

2.2.1.1 ระบบสีแบบ RGB ของ CIE

เป็นระบบสีที่พัฒนาขึ้นโดย CIE (Commission International l 'Eclairage) ซึ่งอ้างอิงสีด้วยสีแดงที่ 700 nm สีเขียวเท่ากับ 546.1 nm และสีน้ำเงิน 435.8 nm

2.2.1.2 ระบบสีแบบ RGB ของ NTSC

เป็นระบบที่พัฒนาโดย NTSC (National Television System Committee) เพื่อใช้สำหรับการแสดงภาพของจอภาพแบบ CRT เป็นมาตรฐานสำหรับผู้ผลิตแบบ CRT ให้มีลักษณะเดียวกัน

2.3 การค้นหาวัตถุจากรูปภาพโดยหลักการผลต่างพื้นหลังแบบคงตัว

(Object Detection by Background Substation Static Method) [3]

หลักการโดยทั่วไปของการนำรูปภาพมาวิเคราะห์ในการค้นหาวัตถุ ทำการอธิบายด้วยระบบเมทริกซ์ซึ่งจะทำให้เข้าใจและสามารถวิเคราะห์ผลการทดสอบได้ง่ายและชัดเจน โดยใช้วิธีการทางตัวเลขและคุณสมบัติทางเมทริกซ์มาแก้ปัญหา ซึ่งสามารถนำภาพมาแปลงให้อยู่ในรูปของเมทริกซ์โดยจับคู่ในลักษณะ 1 ต่อ 1 ระหว่างค่าสีในพิกเซลของภาพที่อยู่ในตำแหน่งหลัก (column) ให้เป็นค่าสีในระบบเมทริกซ์ที่ตำแหน่งหลักเช่นกัน และค่าสีในพิกเซลของภาพที่อยู่ในตำแหน่งแถว (row) ให้เป็นค่าสีในระบบเมทริกซ์ที่ตำแหน่งแถวเช่นกัน

การค้นหาวัตถุจากภาพนั้นจะอาศัยหลักการลบกันของเมทริกซ์ โดยในการลบกันจำเป็นต้องมี 2 เมทริกซ์ของรูปภาพ คือ เมทริกซ์อ้างอิงและเมทริกซ์เปรียบเทียบ นำทั้ง 2 เมทริกซ์มาลบกันในลักษณะของเมทริกซ์ ถ้าไม่มีวัตถุในภาพเปรียบเทียบจะทำให้ผลต่างของ 2 เมทริกซ์นั้นเป็นศูนย์ทั้งหมด (ในอุดมคติ) หรืออาจจะเป็นค่าเพียงเล็กน้อยเท่านั้นในแต่ละตำแหน่งในเมทริกซ์ แต่ถ้ามีวัตถุในภาพเปรียบเทียบจะทำให้ผลต่างของ 2 เมทริกซ์ มีค่ามากซึ่งสังเกตจากความเปลี่ยนแปลงเทียบจากศูนย์ได้ง่าย ทำให้ผู้วิเคราะห์สามารถบอกได้ว่าภาพทั้ง 2 ภาพ มีการเปลี่ยนแปลงเกิดขึ้น และในที่นี้ถ้าภาพที่ได้จากกล้องที่ตั้งอยู่ในลักษณะคงที่ (คือไม่มีการขยับของกล้อง) ทำให้เมื่อมีวัตถุผ่านเข้ามาจะได้ค่าจากผลต่างของ 2 เมทริกซ์ ที่มีค่ามากเมื่อเทียบกับศูนย์

2.3.1 ขั้นตอนการค้นหาวัตถุจากวิธีการผลต่างพื้นหลังแบบคงตัว

2.3.1.1 นำรูปภาพหรือเฟรมของวีดีโอที่ทำหน้าที่เป็นภาพหรือเฟรมอ้างอิง โดยที่ยังไม่มีวัตถุในภาพหรือเฟรมนั้นมาแปลงให้อยู่ในเมทริกซ์ ถ้าเป็นภาพสีให้สร้างในลักษณะเมทริกซ์ 3 มิติ คือ แถว หลัก และแม่สี โดยแถวและหลักของเมทริกซ์จะสัมพันธ์กับความสูงและความกว้างของภาพ ส่วนแม่สีนั้นคือ สีแดง สีเขียว และสีฟ้า ซึ่งมีระดับค่าสีเป็นตัวกำหนดค่าในแต่ละสมาชิกของเมทริกซ์ แต่ถ้าเป็นภาพระดับสีเทา (Gray Scale) ให้สร้างเมทริกซ์ 2 มิติ คือ แถวและหลัก โดยแถวของเมทริกซ์จะสัมพันธ์กับความสูงของภาพ และหลักของเมทริกซ์จะสัมพันธ์กับความกว้างของภาพซึ่งแต่ละสมาชิกในเมทริกซ์จะเก็บค่าของระดับสีเทาโดยให้เก็บไว้ในเมทริกซ์อ้างอิง ตามรูปที่ 2.1

$$\begin{bmatrix} 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \end{bmatrix}$$

รูปที่ 2.1 ภาพแสดงการแปลงภาพอ้างอิงมาเป็นเมทริกซ์อ้างอิง

2.3.1.2 นำรูปภาพหรือเฟรมของวีดีโอที่นำมาเปรียบเทียบทำในลักษณะเดียวกันกับในข้อที่ 1 โดยใช้อีกเมทริกซ์ คือ เมทริกซ์เปรียบเทียบ ตามรูปที่ 2.2

$$\begin{bmatrix} 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & x & 128 & 128 & 128 \\ 128 & 128 & 128 & x & x & 128 & 128 & 128 \\ 128 & 128 & 128 & x & x & 128 & 128 & 128 \\ 128 & 128 & 128 & x & x & 128 & 128 & 128 \\ 128 & 128 & 128 & x & x & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & x & 128 & 128 & 128 \\ 128 & 128 & 128 & x & x & 128 & 128 & 128 \\ 128 & 128 & 128 & x & x & 128 & 128 & 128 \\ 128 & 128 & 128 & x & x & x & 128 & 128 \\ 128 & 128 & 128 & x & x & 128 & 128 & 128 \\ 128 & 128 & 128 & x & x & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \end{bmatrix}$$

รูปที่ 2.2 ภาพแสดงการแปลงภาพเปรียบเทียบมาเป็นเมทริกซ์เปรียบเทียบ

2.3.1.3 นำเมทริกซ์อ้างอิงและเมทริกซ์เปรียบเทียบมาลบกันตามหลักการของการลบเมทริกซ์ตามรูป 2.3

$$\begin{bmatrix} 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \end{bmatrix} - \begin{bmatrix} 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & x & 128 & 128 & 128 \\ 128 & 128 & 128 & x & x & 128 & 128 & 128 \\ 128 & 128 & 128 & x & x & 128 & 128 & 128 \\ 128 & 128 & 128 & x & x & 128 & 128 & 128 \\ 128 & 128 & 128 & x & x & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & x & 128 & 128 & 128 \\ 128 & 128 & 128 & x & x & 128 & 128 & 128 \\ 128 & 128 & 128 & x & x & 128 & 128 & 128 \\ 128 & 128 & 128 & x & x & x & 128 & 128 \\ 128 & 128 & 128 & x & x & 128 & 128 & 128 \\ 128 & 128 & 128 & x & x & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \end{bmatrix}$$

รูปที่ 2.3 ภาพแสดงการลบกันของเมทริกซ์อ้างอิงกับเมทริกซ์เปรียบเทียบ

2.3.1.4 นำผลต่างที่ได้จากการลบกันมาพิจารณาเมื่อค้นหาวัตถุตามรูป 2.4

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & x & 0 & 0 \\ 0 & 0 & 0 & x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

รูปที่ 2.4 ภาพแสดงผลต่างของ 2 เมทริกซ์

- หาผลรวมของผลต่างของทุกสมาชิกในเมทริกซ์ผลต่าง ถ้าผลรวมที่หาได้มีค่ามากกว่าระดับ (Threshold) สามารถสรุปได้ว่า ภาพที่นำมาเปรียบเทียบนั้นต่างจากภาพอ้างอิง อาจจะเกิดจากการเคลื่อนไหวหรือการมีวัตถุอื่นเข้ามาในภาพ

- หาผลรวมของผลต่างของทุกสมาชิกในเมทริกซ์ผลต่าง ถ้าผลรวมที่หาได้มีค่าน้อยกว่าระดับ (Threshold) สามารถสรุปได้ว่า ภาพที่นำมาเปรียบเทียบนั้นคล้ายหรือใกล้เคียงกับภาพอ้างอิงหรืออาจจะมีวัตถุอื่นที่มีขนาดเล็กกว่าสิ่งที่สนใจเข้ามาในภาพ

2.3.1.5 สามารถนำเมทริกซ์ผลต่างที่ได้นั้นมาสร้างเป็นรูปภาพ เพื่อแสดงให้เห็นความแตกต่างได้ดังรูป 2.5

รูปที่ 2.5 ภาพการแสดงภาพที่สร้างจากผลต่างของเมทริกซ์

2.3.2 การบวกและการลบเมทริกซ์

การบวกเมทริกซ์

บทนิยาม ให้ $A = [a_{ij}]_{m\times n}$ และ $B = [b_{ij}]_{m\times n}$ จะได้ว่า $A + B = [c_{ij}]_{m\times n}$ โดยที่ $ci_j = a_{ij} + b_{ij}$

เมทริกซ์ที่เกิดจากเมทริกซ์ A บวกกับเมทริกซ์ B นั้นจะเกิดขึ้นได้ต้องมีเงื่อนไข 2 ประการ คือ

1. เมทริกซ์ A และ B ต้องมีมิติเท่ากัน (มิติ m x n)

2. สมาชิกที่เป็นผลลัพธ์นั้นเกิดจาการนำสมาชิกในเมทริกซ์ A และสมาชิกในเมทริกซ์ B มาบวกกัน แต่ต้องเป็นสมาชิกที่อยู่ในตำแหน่งเดียวกันทั้งหมด นั่นคือ

$$c_{11} = a_{11} + b_{11}$$

$$c_{12} = a_{12} + b_{12}$$

$$c_{13} = a_{13} + b_{13}$$

เป็นเช่นนี้ไปเรื่อยๆ จนครบสมาชิกทุกตำแหน่ง

การลบเมทริกซ์

บทนิยาม ให้ $A = [a_{ij}]_{m\times n}$ และ $B = [b_{ij}]_{m\times n}$ จะได้ว่า $A - B = [c_{ij}]_{m\times n}$ โดยที่ $ci_j = a_{ij} - b_{ij}$

การนำเมทริกซ์ 2 เมทริกซ์มาลบกัน มีเงื่อนไข 2 ประการเช่นเดียวกับการบวก กล่าวคือ

1. เมทริกซ์ที่จะนำมาลบกันต้องมีมิติเท่ากัน
2. นำสมาชิกที่อยู่ในตำแหน่งเดียวกันมาลบกัน

จะเห็นได้ว่าการประยุกต์หลักการของเมทริกซ์นั้น สามารถนำมาใช้ได้กับปัญหาในลักษณะที่รู้มิติที่แน่นอนของข้อมูล เช่น การวิเคราะห์ภาพที่มีจำนวนหลักและแถวที่แน่นอน โดยใช้แค่การคำนวณเพียงการลบกันเท่านั้น

2.4 ทฤษฎีการประมวลผลภาพ [4]

2.4.1 Grayscale Image

คือ ภาพที่มีการผสมผสานของสีสองสี คือ ขาวและดำ ดังรูปที่ 2.10 ซึ่งเป็นภาพ Grayscale ของรูปที่ 2.4

2.4.2 Threshold

คือ วิธีการแบ่งส่วนภาพขาวดำ โดยมีวิธีการคือ กำหนดค่า Threshold ขึ้นมา 1 ค่า ระหว่าง 0 ถึง 255 ถ้าจุดพิกเซลใดมีค่าสีมากกว่าค่าที่กำหนดจะถูกแทนที่ด้วยสีดำ ถ้าจุดพิกเซลใดมีค่าสีน้อยกว่าค่าที่กำหนดจะถูกแทนที่ด้วยสีขาว ดังรูปที่ 2.6

รูปที่ 2.6 ภาพปกติ

รูปที่ 2.7 ภาพ Grayscale

รูปที่ 2.8 ภาพการทำ Threshold

2.4.3 Spatial Domain Methods

คือ ค่าของจุดพิกเซล (pixel) ที่ประกอบอยู่ในภาพ โดยสามารถนำมาเขียนเป็นสมการได้ ดังนี้

$$g(x, y) = T[f(x, y)] \qquad (2.1)$$

ซึ่ง $f(x, y)$ คือ ภาพต้นแบบ

$g(x, y)$ คือ ค่าของพิกเซลในภาพที่ทำการประมวลผลแล้ว

T คือ โอเปอร์เรเตอร์ (operator) ที่กระทำกับ $f(x, y)$

2.5 ทฤษฎีการตรวจจับการเคลื่อนไหว

การตรวจจับการเคลื่อนไหวในข้อมูลภาพต่อเนื่องนั้นสามารถทำได้หลายวิธี โดยใช้หลักการที่สำคัญ คือ การนำเฟรมภาพปัจจุบันมาเปรียบเทียบกับเฟรมก่อนหน้าหรือเฟรมแรก

การตรวจจับโดยการเปรียบเทียบเฟรมปัจจุบันกับเฟรมก่อนหน้ากระทำได้ โดยขั้นแรกจะนำเฟรมปัจจุบันและเฟรมก่อนหน้ามาหาความแตกต่างและให้พิกเซลที่ต่างกันเป็นสีดำ จนกระทั่งจำนวนจุดสีดำมีมากพอสมควรจึงระบุว่าบริเวณนี้มีการเคลื่อนไหวเกิดขึ้น แต่กล้องนั้นสามารถเกิดสัญญาณรบกวนได้และมีการสร้างจุดสีดำขึ้นทั้งๆ ที่จุดนั้นไม่ได้มีการเปลี่ยนแปลงเกิดขึ้นจึงต้องทำการลดสัญญาณรบกวนก่อน ข้อเสียของการนำเฟรมปัจจุบันเทียบเฟรมก่อนหน้า คือ เมื่อวัตถุมีเคลื่อนไหวอย่างช้าๆ จะตรวจจับการเคลื่อนไหวได้เพียงเล็กน้อยเท่านั้นในเฟรมต่อเฟรม ดังนั้น จึงเป็นการยากที่จะจับหาวัตถุทั้งชิ้นที่เคลื่อนไหวอยู่และถ้าวัตถุเคลื่อนไหวช้ามากๆ ก็จะไม่สามารถจับอะไรได้เลย การตรวจจับโดยการเปรียบเทียบเฟรมปัจจุบันกับเฟรมแรก ซึ่งจะถือว่าเฟรมนี้เป็นเฟรมฉากหลังของกล้องเมื่อมีวัตถุเคลื่อนไหวผ่านมาเราจะสามารถตรวจจับได้ทันที แต่ข้อเสียคือไม่สามารถแน่ใจได้ว่าเฟรมแรกที่นำมาเปรียบเทียบนั้นเป็นฉากหลังจริงๆ เช่น เฟรมแรกอาจจะเป็นเฟรมที่รถกำลังวิ่ง เมื่อรถวิ่งพ้นกล้องไปแล้วบริเวณที่มีรถอยู่ในเฟรมแรกนั้นจะถูกตรวจจับว่ามีสิ่งเคลื่อนไหวทันที แต่ถ้าภาพแรกเป็นภาพฉากหลัง เช่น ภาพห้องว่างเปล่าก็จะสามารถตรวจจับวัตถุที่

เคลื่อนไหวผ่านเข้ามาในห้องได้ และจะทำแก้ปัญหาที่เกิดขึ้นโดยทำการกำหนดภาพฉากหลังขึ้นมาใหม่ทุกๆ ช่วงเวลาหนึ่งโดยการเปลี่ยนภาพฉากหลังเป็นภาพปัจจุบัน ณ เวลาใดๆ

2.6 การตัดแบ่งส่วนรูปภาพ (Image Segmentation) [4,5]

Image Segmentation คือ กระบวนการแบ่งภาพเป็นพื้นที่หรือวัตถุซึ่งมีความสำคัญมาก เพราะการวิเคราะห์เราจะไม่สามารถกระทำกับภาพทั้งภาพเพื่อให้ได้ผลลัพธ์ที่ดีได้ แต่เราจะวิเคราะห์ภาพเป็นส่วนๆ ไป เช่น การหาวัตถุที่ควรจะเป็นมนุษย์อยู่ตำแหน่งใดในภาพ เป็นต้น ซึ่งการตัดแบ่งส่วนรูปภาพนี้ เป็นขั้นตอนสำคัญของการวิเคราะห์ภาพโดยมีประโยชน์ดังนี้

1. ลดจำนวนข้อมูลในรูปภาพที่ไม่จำเป็นในการวิเคราะห์เนื่องจากการตัดแบ่งส่วนรูปภาพเป็นการแยกแยะระหว่างส่วนที่สนใจ เช่น วัตถุในภาพกับส่วนที่ไม่ต้องการ นั่นคือ ฉากหลัง เมื่อตัดข้อมูลในส่วนที่ไม่ต้องการออกไป จำนวนข้อมูลที่เหลือที่จำเป็นในการวิเคราะห์จริงจะลดลงอย่างมาก

2. จัดระเบียบข้อมูลในรูปภาพให้เป็นกลุ่มได้ดีขึ้น ข้อมูลภาพที่ผ่านการแบ่งแยกแล้วจะมีโครงสร้างที่ชัดเจนขึ้นและนำไปใช้งานได้สะดวกขึ้น

การทำการตัดแบ่งส่วนรูปภาพจะทำให้สามารถแยกข้อมูลภาพในส่วนที่ต้องการออกมาได้ โดยวิธีการพื้นฐานสำหรับการตัดแบ่งส่วนรูปภาพ ได้แก่ การพิจารณาความสว่างของภาพสำหรับภาพระดับเทา และความแตกต่างของสีสำหรับภาพสี นอกจากนี้ลักษณะของภาพและลักษณะของพื้นผิวก็เป็นองค์ประกอบหนึ่งที่จะทำให้สามารถทำการตัดแบ่งส่วนรูปภาพได้สะดวกยิ่งขึ้น โดยการตัดแบ่งส่วนรูปภาพมีหลายวิธี เช่น

- การสร้างภาพแบบไบนารีด้วยการทำเทรชโฮล (Thresholding)
- กระบวนการแบ่งกลุ่มของพิกเซล (Region Labeling)
- กระบวนการขยายกลุ่มของพิกเซล (Region Growing)
- กระบวนการแบ่งและรวมกลุ่มของพิกเซล (Region Splitting and Merging)
- กระบวนการหาขอบของวัตถุ (Boundary detection)
- ในโครงงานฉบับนี้ได้เลือกใช้วิธีการเทรชโฮล (Thresholding) โดยจะอธิบายในหัวข้อ 2.6.1 ต่อไป

2.6.1 การสร้างภาพแบบไบนารีด้วยการทำเทรชโฮล

ภาพที่มนุษย์มองเห็นโดยทั่วไปจัดเป็นภาพสี ซึ่งคุณสมบัติภาพสีนี้จะมี 3 ระนาบ ทำให้การวิเคราะห์ภาพมีความซับซ้อนและยากต่อการพิจารณา การลดทอนความเป็นสีสามารถทำได้โดยการแปลงให้เป็นภาพระดับสีเทา ซึ่งขั้นตอนดังกล่าวนั้นมีหลายวิธี เช่น เลือกระนาบใดระนาบหนึ่งจากภาพสีหรือนำทั้ง 3 ระนาบมาทำการหาค่าเฉลี่ยเพื่อให้เหลือเพียงระนาบเดียวเป็นต้น ซึ่งภาพระดับเทาที่ได้จะซับซ้อนน้อยกว่าภาพสีเมื่อนำมาวิเคราะห์วัตถุต่างๆ ภายในภาพ

ภาพแบบไบนารี คือ ภาพที่มีระดับความเข้มเพียง 1 บิต หรือ 2 ระดับ คือ ในหนึ่งพิกเซลมีค่าเพียงสองค่าคือ 0 หรือ 1 เท่านั้น โดยพิกเซลที่มีค่าเท่ากับ 1 จะมีสีขาวและพิกเซลที่มีค่าเท่ากับ 0 จะมีสีดำ ในการสร้างภาพแบบไบนารีนั้นจะใช้วิธีการทำเทรชโฮลโดยมีหลักการพิจารณาคือ จะพิจารณาจุดของภาพว่าจุดใดควรจะเป็นจุดดำหรือจุดขาว โดยเปรียบเทียบระหว่างค่าในพิกเซลเริ่มต้นกับค่าคงที่ค่าหนึ่งหรือค่าเทรชโฮลนั่นเอง ซึ่งวิธีเทรชโฮลนิยมใช้มากในกรณีที่วัตถุและพื้นหลังของภาพมีความแตกต่างกันอย่างชัดเจน

บางครั้งการเลือกค่าเทรชโฮลที่ไม่เหมาะสม คือ เลือกค่าเทรชโฮลที่มากหรือน้อยจนเกินไป จะทำให้ผลลัพธ์ที่ได้ขาดความชัดคมและรายละเอียดบางส่วนขาดหายไป กล่าวคือภาพที่ได้อาจจะมืดเกินไปหรือสว่างเกินไปหรือบางทีอาจมีสิ่งรบกวนเกิดขึ้น ดังนั้น จะแก้ปัญหาเหล่านี้ด้วยการเลือกเทรชโฮลที่เหมาะสม ซึ่งการเลือกเทรชโฮลนั้นมีอยู่หลายวิธีโดยแต่ละวิธีเหมาะสมกับลักษณะการทำงานที่แตกต่างกัน เช่น การหาค่าเทรชโฮลโดยการกำหนดค่าเองและการหาค่าเทรชโฮลแบบอัตโนมัติ ซึ่งแต่ละวิธีอธิบายได้ดังนี้

2.6.1.1 การหาค่าเทรชโฮลโดยการกำหนดค่าเอง

วิธีนี้เป็นที่นิยมเพราะว่าเป็นวิธีที่ง่ายที่สุด โดยการสุ่มค่าเทรชโฮล ซึ่งการสุ่มค่าเทรชโฮลนั้นขึ้นอยู่กับประสบการณ์ของผู้ใช้เอง โดยทำการเลือกค่าคงที่ที่เหมาะสมค่าหนึ่ง เป็นค่าคงที่ที่อยู่ระหว่างค่าต่ำสุดและสูงสุดของระดับความเข้มของข้อมูลภาพรับเข้า ถ้าเป็นภาพระดับเทาที่มีความลึก 8 บิต ค่าเทรชโฮลที่ได้จะอยู่ระหว่าง 0 ถึง 255 เช่น ถ้าเลือกค่าเทรชโฮลเท่ากับ 128 จะทำให้พิกเซลที่มีความเข้มมากกว่าหรือเท่ากับ 128 จะกลายเป็นสีขาว และพิกเซลที่มีความเข้มน้อยกว่า 128 จะกลายเป็นสีดำ ดังสมการ 2.2

$$g(x,y) = \begin{cases} 1 & f(x,y) > T \\ 0 & f(x,y) < T \end{cases} \qquad (2.2)$$

โดยที่ f(x,y) คือ ข้อมูลภาพรับเข้า ที่มีระดับความเข้ม 0 ถึง L

g(x,y) คือ ข้อมูลภาพผลลัพธ์เป็นภาพแบบไบนารี

T คือ ค่าเทรชโฮล

1 คือ จุดสีขาว

0 คือ จุดสีดำ

2.6.1.2 การหาค่าเทรชโฮลแบบอัตโนมัติ

วิธีการนี้เหมาะสำหรับภาพที่มีความเข้มของวัตถุและพื้นหลังแยกกันอย่างชัดเจน วิธีนี้เป็นการหาค่าเทรชโฮลโดยไม่ต้องให้ผู้ใช้กำหนด โดยจะพิจารณาจากค่ากลางใช้การคำนวณพื้นฐานทางสถิติเข้ามาช่วยซึ่งจะได้ค่าเทรชโฮลที่มีความเหมาะสม โดยจะมีความสัมพันธ์กับความเข้มของภาพ เช่น ภาพที่มืดค่าเทรชโฮลจะมีค่าน้อย หรือ ภาพที่มีความสว่างสูงค่าเทรชโฮลก็จะสูงไปด้วย เป็นต้น

วิธีหาค่าเทรชโฮลนี้ จะเริ่มต้นจากนำค่ากลางระหว่างค่าระดับความเข้มสูงสุดและค่าระดับความเข้มต่ำสุดภายในภาพเป็นค่าเทรชโฮลเริ่มต้น (T0) จากนั้นใช้ค่า T0 แบ่งพิกเซลเป็นสองส่วน คือ ส่วนที่มีความเข้มน้อยกว่า T0 และส่วนที่มีความเข้มมากว่าหรือเท่ากับ T0 จากนั้นหาค่าเฉลี่ยของกลุ่มพิกเซลทั้งสองได้เป็น $\mu 1$ และ $\mu 2$ ตามลำดับ แล้วทำการหาค่าเฉลี่ยระหว่าง $\mu 1$ และ $\mu 2$ จะได้ค่าเทรชโฮลใหม่ (T) ถึงขั้นตอนนี้จะได้ค่าเทรชโฮล 2 ค่า คือ T0 และ T ต่อมาจะทำการหาความแตกต่างระหว่าง T0 กับ T ถ้ามีค่าน้อยกว่าที่กำหนดไว้ (D) ซึ่งส่วนใหญ่จะกำหนดไว้น้อยกว่า 5 ก็ให้ใช้ T เป็นค่าเทรชโฮล ถ้าค่าความแตกต่างยังมากอยู่ แสดงว่าค่าเทรชโฮลที่ได้ยังไม่เหมาะสมจะต้องกลับไปหาค่าเทรชโฮลต่อด้วยขั้นตอนเดิมอีกครั้ง โดยให้ T เท่ากับ T0 ได้แสดงวิธีการคำนวณไว้ดังนี้

1. $T0 = (\text{Min}(f(x,y)) + \text{Max}(f(x,y))) / 2$

   โดย f(x,y) คือ ภาพระดับสีเทา

   Min(f(x,y)) คือ ระดับความเข้มที่น้อยที่สุดของภาพ

   Max(f(x,y)) คือ ระดับความเข้มที่มากที่สุดของภาพ

2. แบ่งพิกเซลเป็น 2 ส่วน

   g1(x,y) คือ f(x,y) ที่มีระดับความเข้ม $<$ T0

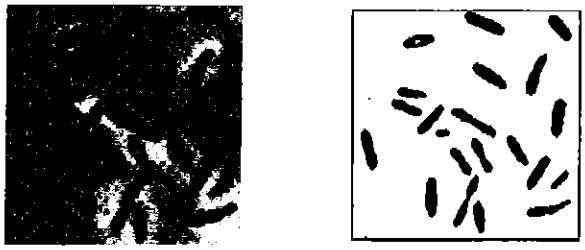
   g2(x,y) คือ f(x,y) ที่มีระดับความเข้ม $\geq$ T0

3. คำนวณค่าเฉลี่ยของ $g1(x,y) = \mu 1$ และ $g2(x,y) = \mu 2$

4. ตั้งค่า T ใหม่โดยให้ $T = (\mu 1 + \mu 2) / 2$

5. ย้อนกลับไปทำข้อ 2 อีกครั้ง จนกว่า T – T0 จะมีค่าน้อยกว่าค่าที่กำหนดไว้ (D)

เมื่อได้ทำการเทรชโฮลกับภาพระดับเทาซึ่งเป็นภาพของแบคทีเรีย โดยใช้ค่าเทรชโฮลเท่ากับ 102 แล้ว จะได้ภาพแบบไบนารี ดังรูป 2.9 (ก) และ (ข) ตามลำดับ



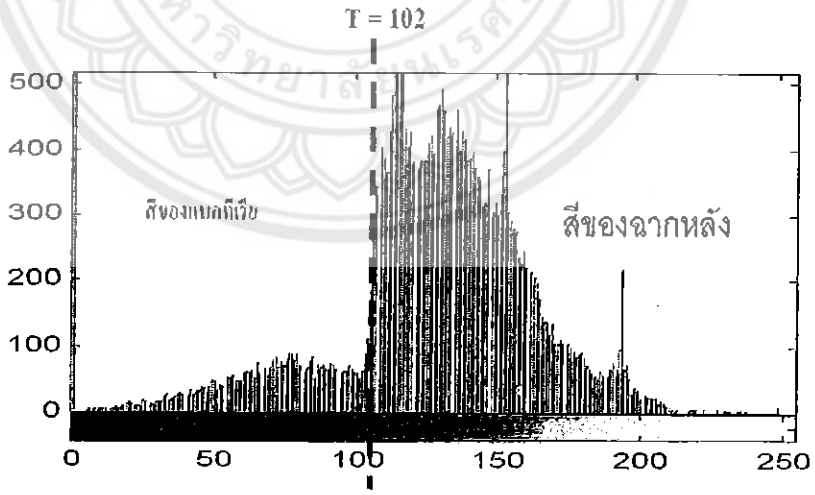
(ก) (ข)

รูปที่ 2.9 ทำการเทรชโฮลกับภาพระดับเทา
(ก) ภาพของแบคทีเรีย
(ข) ภาพของแบคทีเรียหลังการทำเทรชโฮล

เพื่อแสดงให้เห็นภาพของการเทรชโฮลได้ง่ายขึ้น การนำเสนอในรูปของ Histogram จะทำให้เข้าใจยิ่งขึ้น ซึ่ง Histogram คือ จำนวนพิกเซลที่มีในแต่ละระดับความเข้ม

ภาพแบคทีเรียด้วยค่าเทรชโฮลเท่ากับ 102 แสดงดังรูป 2.10 โดยแกน x คือ ค่าระดับความเข้มอยู่ในช่วง 0 ถึง 255 และแกน y คือ จำนวนพิกเซลของภาพแบคทีเรีย สัดส่วนภายในภาพมีดังนี้

- ส่วนที่เป็นแบคทีเรีย มีระดับความเข้มอยู่ในช่วง 0 ถึง 101
- ส่วนของฉากหลังจะมีจำนวนมากกว่าส่วนที่เป็นแบคทีเรียอยู่มาก มีระดับค่าความเข้มอยู่ในช่วง 102 ถึง 255

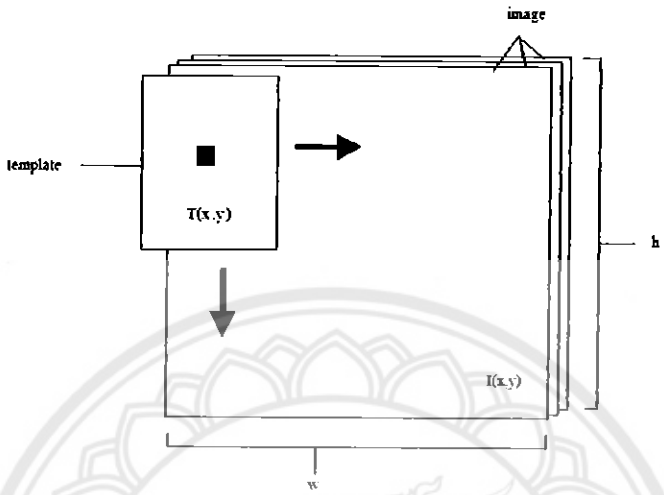


รูปที่ 2.10 Histogram ของภาพแบคทีเรียด้วยค่าเทรชโฮลเท่ากับ 102

2.7 การจับคู่แม่แบบ (Template Matching) [6,7]

วิธีการจับคู่แม่แบบ คือ กระบวนการในการหาความเหมือนหรือความแตกต่างของภาพ โดยการจับคู่แม่แบบสามารถวัดได้จากการคำนวณระหว่างเทมเพลต โดยเริ่มจากการนำเทมเพลต

แม่แบบทาบกับภาพที่เราต้องการเปรียบเทียบโดยใช้วิธีเปรียบเทียบแบบจุดต่อจุด โดยเริ่มจากเอาจุด T (1,1) ของเทมเพลตแม่แบบทาบกับจุด I (1,1) ของภาพที่นำมาเปรียบเทียบ แล้วเลื่อนจุดทาบไปทีละหลักและทีละแถว โดยให้เทมเพลตอยู่ใน w x h ตามรูปที่ 2.11



รูปที่ 2.11 แสดงการเปรียบเทียบเทมเพลตแม่แบบและภาพที่นำมาเปรียบเทียบ

จากนั้นหาค่าความคลาดเคลื่อน (error) ของการทาบในแต่ละครั้งโดยใช้วิธีผลต่างกำลังแบบปกติ (Square diff normal) ตามสมการที่ 2.3

$$R(x,y) = sum_{x',y'}[\ T(x',y') - I(x+x',y+y')\ ]^2 / sqrt\ [\ sum_{x',y'} T(x',y')^2 \cdot sum_{x',y'} I\ (x+x',y+y')^2\ ] \qquad (2.3)$$

โดย

I คือ รูปภาพ

T คือ เทมเพลต

R คือ ค่าคลาดเคลื่อน (error)

x ,y คือ ตำแหน่งพิกเซลในภาพ

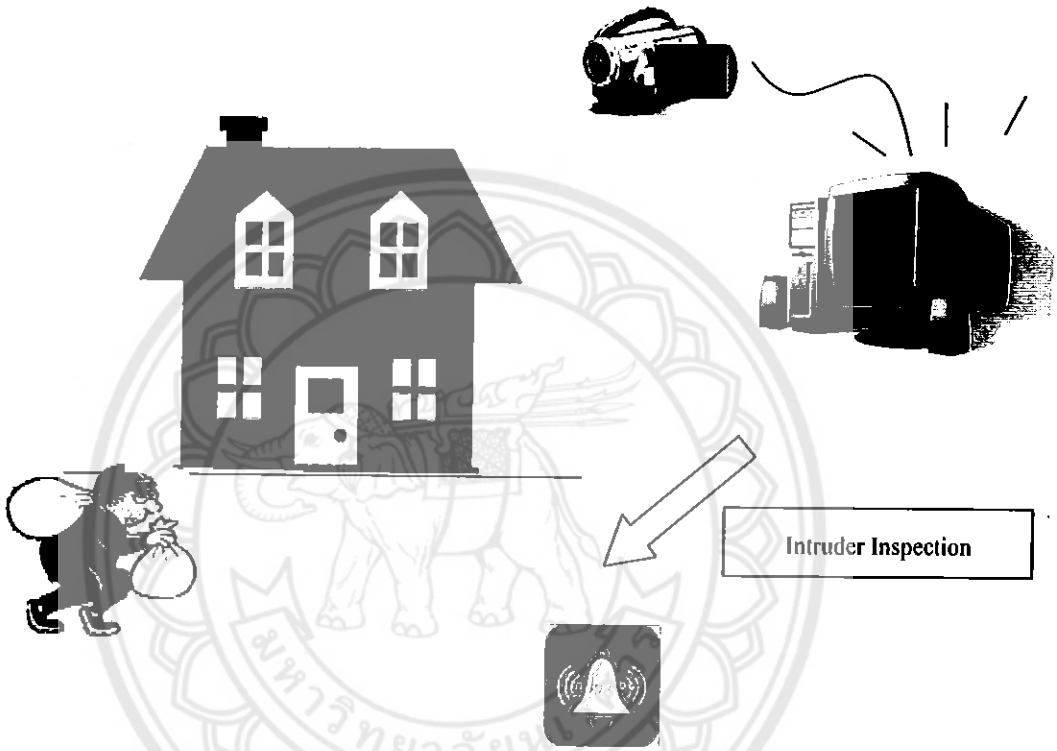
x',y' คือ ตำแหน่งพิกเซลในเทมเพลต

เมื่อได้ค่าความคลาดเคลื่อน (error) จากการคำนวณตามสมการจะนำค่าความคลาดเคลื่อน (error) ที่เกิดขึ้นน้อยสุดในการเปรียบเทียบเทมเพลตนำมาเปรียบเทียบกับค่าความคลาดเคลื่อน (error) ที่ได้กำหนดขึ้นถ้าค่าความคลาดเคลื่อน (error) ที่คำนวณได้มีค่าน้อยกว่าค่าที่ได้กำหนดจะพบว่าเป็นภาพหรือวัตถุชนิดเดียวกับแม่แบบ

เมื่อได้ศึกษาทฤษฎีและหลักการเกี่ยวกับการประมวลผลภาพดังที่ได้กล่าวมาในข้างต้นแล้ว จากนั้นจึงได้ทำการวิเคราะห์ปัญหาของโครงงานเพื่อทำการพัฒนาโปรแกรมเพื่อใช้ในการตรวจจับผู้บุกรุก โดยจะใช้มาตรฐานสีแบบ RGB ในการพิจารณาและนำวิธีผลต่างพื้นหลังแบบคงตัวมาเพื่อใช้ในการแยกแยะวัตถุออกจากภาพ จากนั้นใช้ตัวคัดกรองแบบแบ่งระดับเพื่อทำการคัดเลือกเอาแต่เฉพาะวัตถุที่สนใจโดยพิจารณาจากค่าสีที่ทำการคัดกรอง จากนั้นจึงเปลี่ยนภาพที่ได้ให้กลายเป็นภาพขาว-ดำ และสุดท้ายจึงทำการค้นหาลักษณะจับคู่แม่แบบเพื่อทำการตรวจสอบรูปร่างวัตถุกับเทมเพลตที่มีในระบบซึ่งขั้นตอนและรายละเอียดได้อธิบายในบทถัดไป

บทที่ 3
ขั้นตอนการดำเนินงาน

3.1 ภาพโดยรวมของระบบ



รูปที่ 3.1 แสดงภาพโดยรวมของระบบ

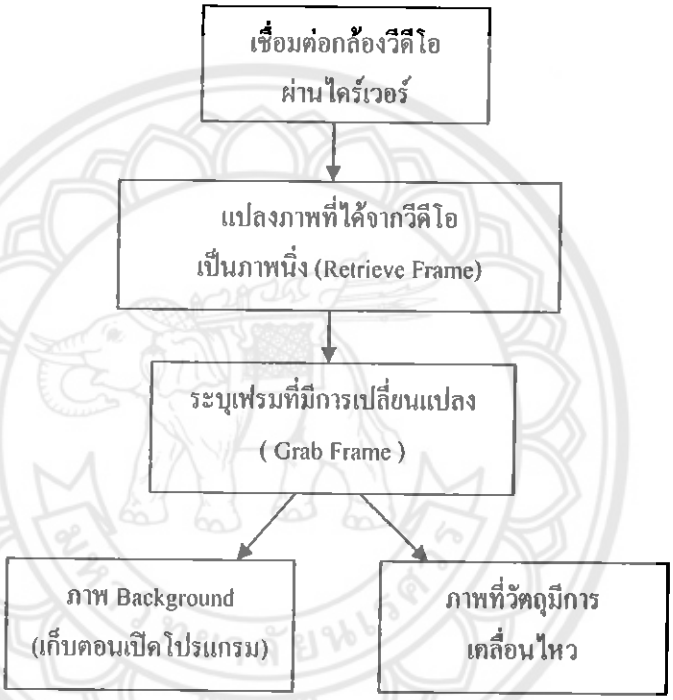
โปรแกรมตรวจสอบผู้บุกรุกเป็นระบบที่ทำการเชื่อมต่อกับกล้องวีดีโอ โดยรับสัญญาณภาพจากกล้องเข้ามาประมวลผลที่คอมพิวเตอร์ โดยใช้ Library OpenCV ในการเชื่อมต่อกล้องกับคอมพิวเตอร์ และใช้ฟังก์ชั่นใน OpenCV ในการประมวลผลตามกระบวนการที่จะกล่าวถึงในลำดับต่อไป

จากรูปเมื่อมีคนเดินผ่านเข้ามาในกล้อง คอมพิวเตอร์ก็จะนำภาพไปประมวลผล ซึ่งเมื่อพบว่าวัตถุในภาพเป็นรูปร่างของคน คอมพิวเตอร์จะทำการแจ้งเตือน

3.2 การออกแบบโปรแกรม

รูปที่ 3.2 บล็อกไดอะแกรมแสดงระบบการทำงานโดยรวม

3.2.1 ส่วนขยายของการรับภาพจากกล้อง (Video capture from camera)



รูปที่ 3.3 บล็อกไดอะแกรมแสดงส่วนขยายของการรับภาพจากกล้อง

อธิบายในส่วนขยายของการรับภาพจากกล้อง

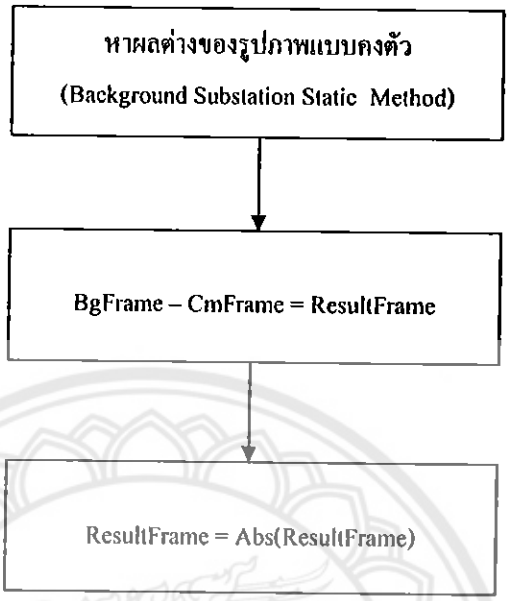
1. ติดต่อกล้องวีดีโอผ่านไดร์เวอร์
2. ทำการระบุเฟรมที่ต้องการ (Grab Frame) ของภาพเคลื่อนไหว
3. ทำการแปลงภาพเคลื่อนไหวเพื่อให้ได้ภาพนิ่ง (Retrieve Frame) ซึ่งจะได้เป็นสองภาพ คือ

   3.1 ภาพฉากหลัง

   3.2 ภาพที่วัตถุมีการเคลื่อนไหวเพื่อนำมาใช้เปรียบเทียบกับภาพฉากหลัก

3.2.2 ส่วนขยายของการแยกและวิเคราะห์วัตถุ (Human Inspection)

3.2.2.1 ทำ Image segmentation

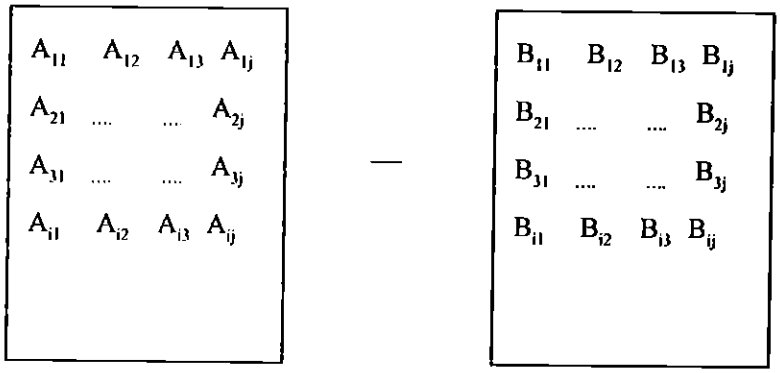


รูปที่ 3.4 บล็อกไดอะแกรมแสดงการทำ Image segmentation

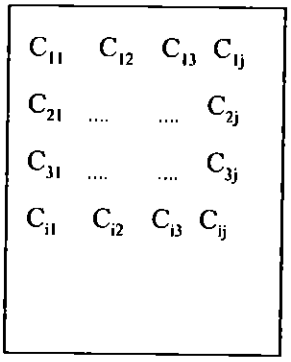
อธิบายในส่วนของการทำ Image segmentation

1. เมื่อได้ภาพฉากหลัง (Background Frame) และภาพที่นำมาทำการเปรียบเทียบ (Compared Frame) หลังจากนั้นนำเมทริกซ์ทั้ง 2 เมทริกซ์ มาลบกันตามลักษณะของเมทริกซ์โดยจะได้ตามนิยาม คือ ให้ภาพแรกคือ A และภาพที่สองคือ B โดย $A = [a_{ij}]_{mxn}$ และ $B = [b_{ij}]_{mxn}$ จะได้ว่า $A - B = [C_{ij}]_{mxn}$ ซึ่ง

$$C_{ij} = a_{ij} - b_{ij}$$



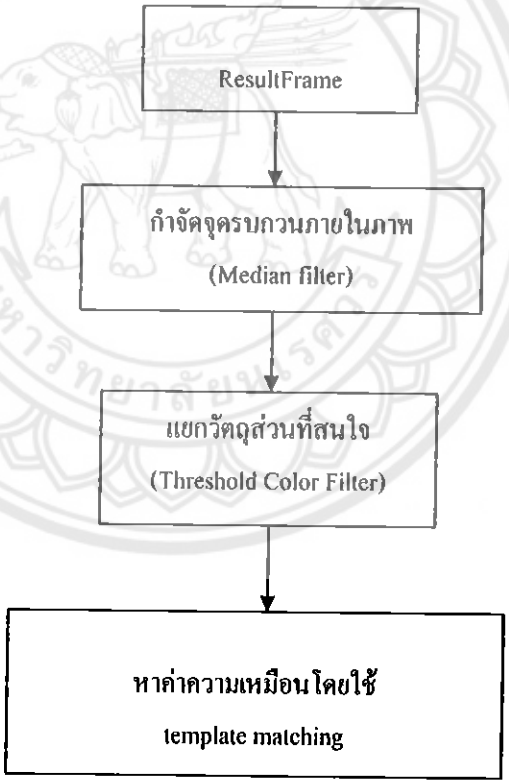
รูปที่ 3.5 แสดงการลบเมทริกซ์ของ 2 เฟรม



รูปที่ 3.6 แสดงเมริกที่ได้จากการลบกันของเมทริกซ์ A และ B

2. จากนั้นนำค่าในเมทริกซ์ C หรือภาพผลลัพธ์นำไปหาค่าสัมบูรณ์ในทุกๆ พิกเซลเพื่อสามารถพิจารณาค่าที่เปลี่ยนแปลงไปในทิศทางเดียว (พิจารณาค่าบวก)

3.2.2.2 ทำการค้นหาวัตถุและวิเคราะห์วัตถุ



รูปที่ 3.7 บล็อกไดอะแกรมแสดงการค้นหาวัตถุและวิเคราะห์วัตถุ

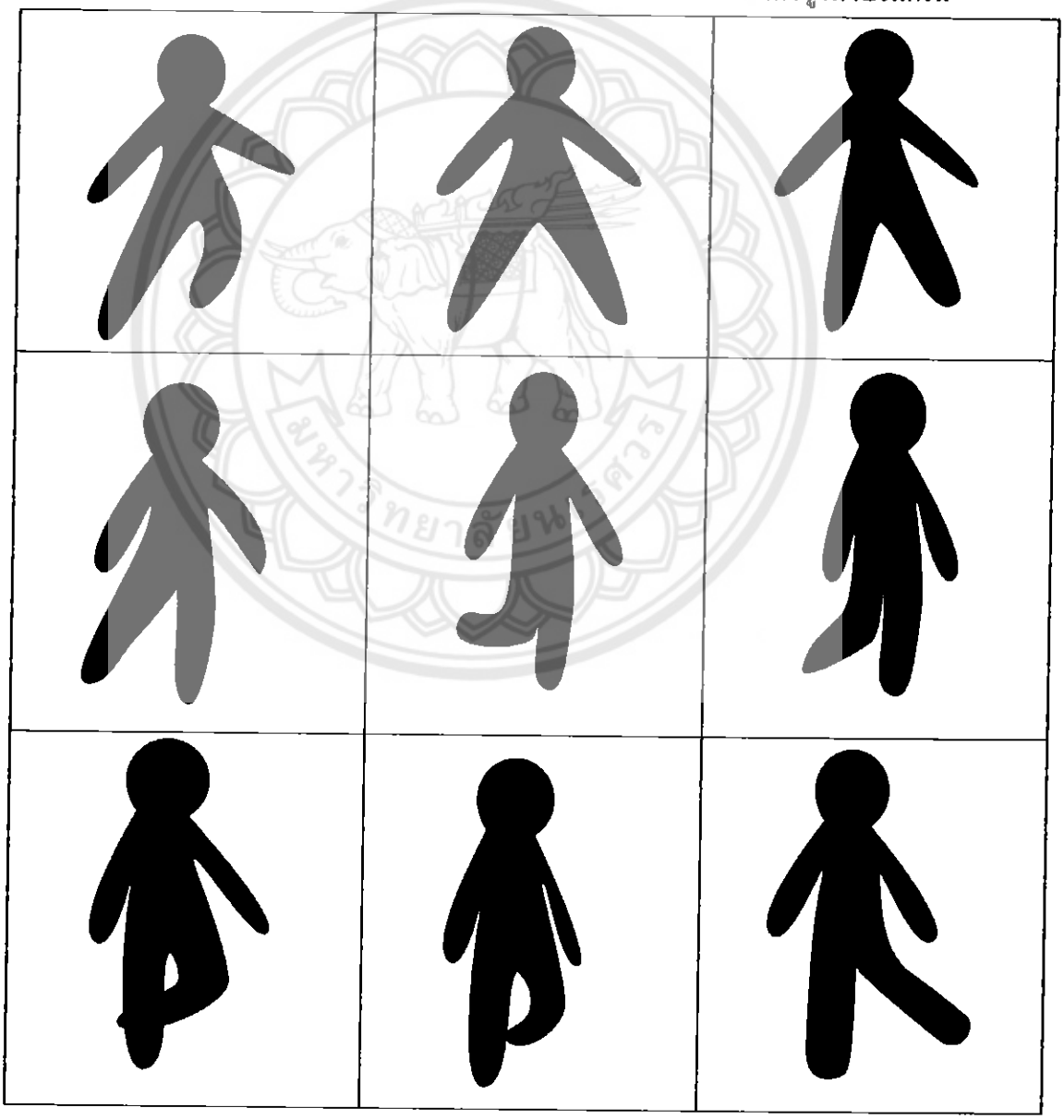
อธิบายในส่วนของการทำค้นหาวัตถุและวิเคราะห์วัตถุ

1. เมื่อได้รูปที่ผ่านการลบหรือภาพผลลัพธ์ นำภาพที่ได้มาทำการคัดกรองเพื่อกำจัดจุดรบกวน (noise) ในภาพ

2. จากนั้นนำมาผ่านตัวคัดกรองแบ่งระดับ (Threshold Color Filter) เพื่อทำการแยกเอาเฉพาะส่วนที่สนใจออกมาเท่านั้น โดยกำหนดให้ส่วนที่ผ่านค่าเทรชโฮลเป็นสีดำ และส่วนที่ไม่ผ่านค่าเทรชโฮลเป็นสีขาว ซึ่งคือการทำให้เป็นภาพไบนารีนั่นเอง

3. หาความเหมือนของวัตถุในภาพกับภาพต้นแบบโดยวิธีจับคู่แม่แบบ (template matching) ซึ่งจะกำหนดภาพต้นแบบไว้ทั้งหมด 9 แบบ ดังนี้

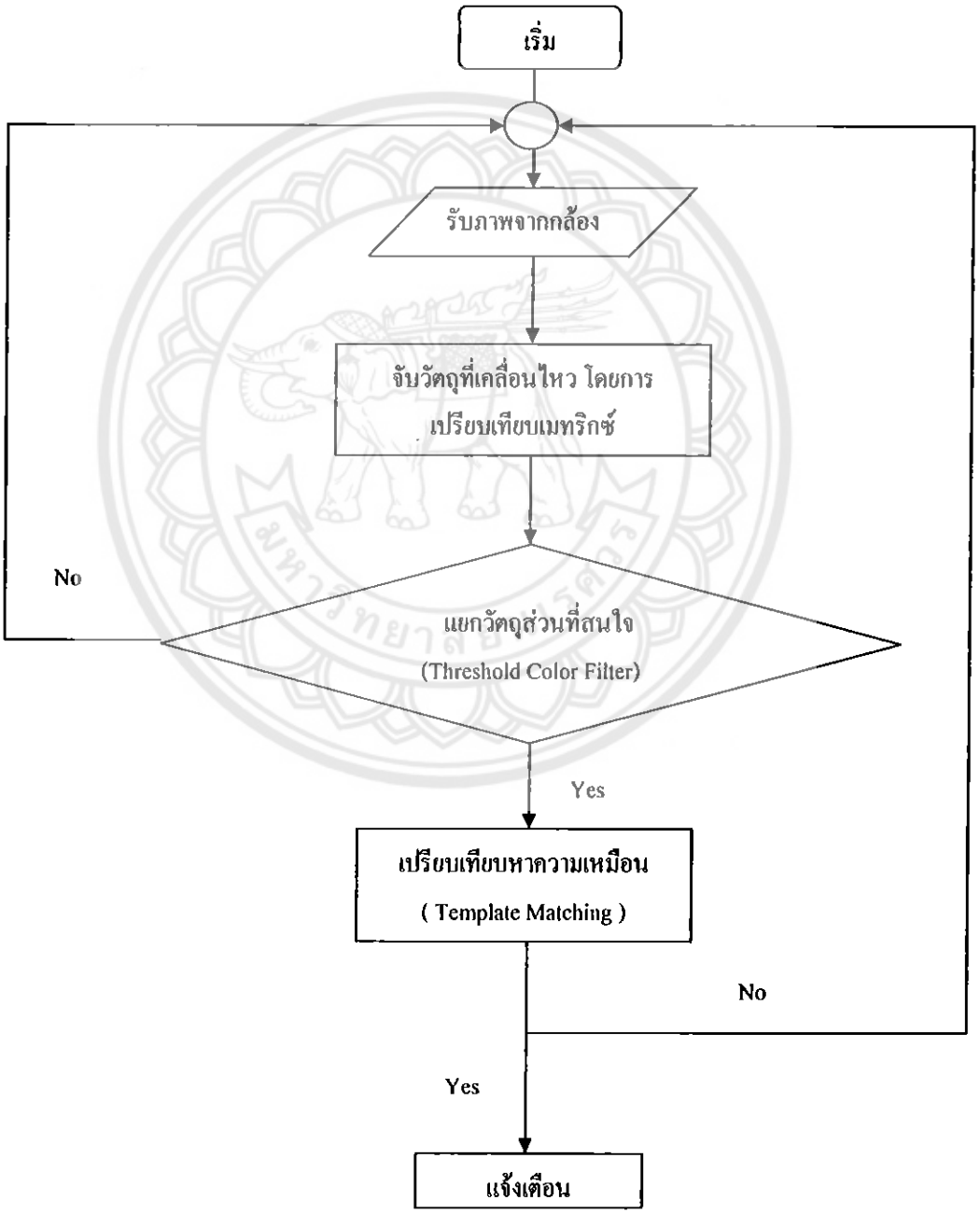
ตารางที่ 3.1 แสดงเทมเพลตต้นแบบที่ใช้ในการเปรียบเทียบที่มีอยู่ในโปรแกรม



3.2.3 ส่วนขยายของระบบแจ้งเตือน (Alarm System)

ในส่วนของการแจ้งเตือนโปรแกรมจะทำการแจ้งเตือนโดยใช้เสียงของคอมพิวเตอร์ในการแจ้งเตือน ซึ่งจะทำการแจ้งเตือนเมื่อโปรแกรมทำการประมวลผลแล้วพบวัตถุที่เป็นคนในกล้องวีดีโอ โดยประแกรมจะทำการเปลี่ยนเป็นสถานะจาก “NORMAL” เป็น “INTRUDER” และทำการแจ้งเตือน

3.3 การทำงานของโปรแกรม (Flow Chart)



รูปที่ 3.17 แสดงแผนภาพการทำงานของโปรแกรม

3.4 ออกแบบหน้าต่างการใช้งานของโปรแกรม

หน้าต่างผู้ใช่งาน (User) แบ่งเป็น 5 ส่วน คือ

ส่วนที่ 1 Original VDO เป็นส่วนที่แสดงภาพที่รับเข้ามาจากกล้อง วีดีโอ

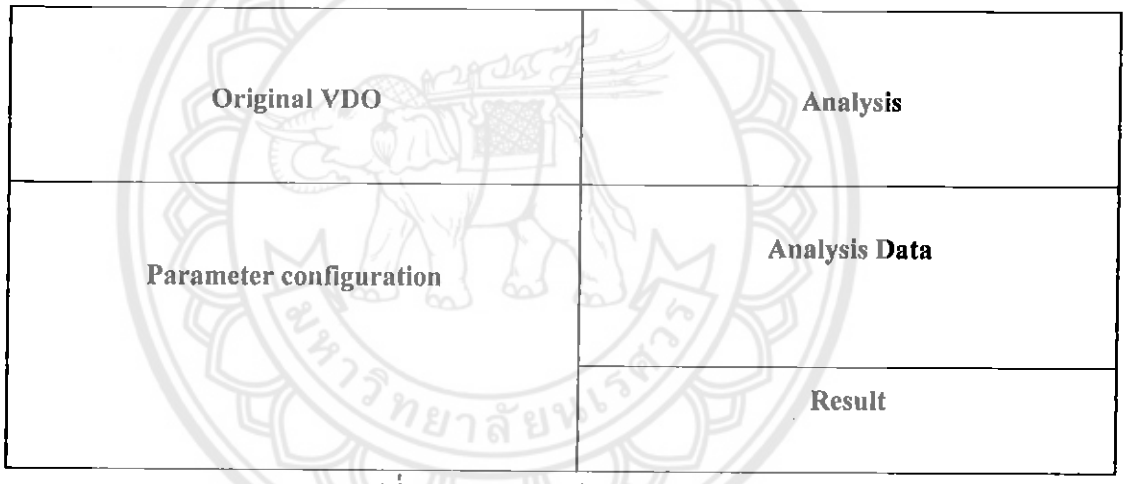
ส่วนที่ 2 Analysis เป็นส่วนที่ใช้พิจารณาวัตถุที่สนใจ ในที่นี้คือ รูปร่างของคน

ส่วนที่ 3 Analysis Data เป็นส่วนที่แสดงกราฟและข้อมูลที่ได้จากกระบวนการวิเคราะห์

ส่วนที่ 4 Parameter configuration เป็นส่วนที่ใช้ในการปรับแต่งภาพต่างๆ ซึ่งได้แก่ ค่า Threshold และค่าความเหมือนของภาพ

ส่วนที่ 5 Result เป็นส่วนที่แสดงผลลัพธ์โดยจะแสดง text ดังนี้

ถ้าพบคนเดินผ่านกล้อง	จะแสดง	"INTRUDER"
ถ้าพบวัตถุเคลื่อนที่ผ่านกล้อง	จะแสดง	"SUSPICIOUS OBJECT"
ถ้าเหตุการณ์เป็นปกติ	จะแสดง	"NORMAL "



รูปที่ 3.18 แสดงหน้าต่างการใช้งาน

บทที่ 4

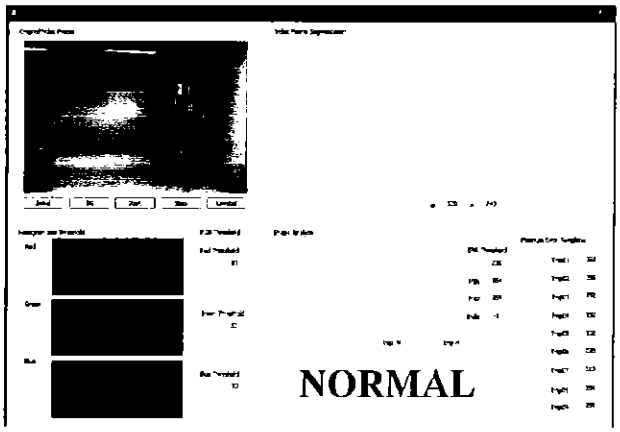
ผลการทดลอง

หลักการทำงานของโครงงานจะใช้วิธีการหาค่าความแตกต่างของรูปภาพจากการหาค่าผลต่างของเมทริกซ์โดยการนำภาพปัจจุบันเปรียบเทียบกับภาพที่ได้ทำการบันทึกไว้ก่อนหน้า เมื่อทั้งสองภาพมีความแตกต่างกันโปรแกรมจะนำภาพที่ได้ไปทำการจับคู่กับเทมเพลตแม่แบบที่โปรแกรมได้ทำการบันทึกไว้ เมื่อโปรแกรมทำการตรวจสอบและพบว่าเป็นผู้ที่บุกรุกเข้ามาโปรแกรมจะทำการแจ้งเตือน โดยเริ่มต้นโปรแกรมจะทำการจับภาพโดยใช้กล้องวีดีโอทำการจับภาพเริ่มต้น (ภาพที่ใช้ในการเปรียบเทียบ) เก็บไว้และทำการจับภาพต่อไป เมื่อโปรแกรมทำการประมวลผลถ้าไม่มีผู้บุกรุกหรือสิ่งที่แตกต่างจากภาพเริ่มต้นโปรแกรมจะย้อนกลับไปทำงานซ้ำที่ลูปเดิม แต่ถ้าหากมีผู้บุกรุกหรือมีสิ่งที่แตกต่างจากภาพเริ่มต้นที่ได้บันทึกไว้โปรแกรมจะทำการวิเคราะห์รูปร่างและทำการแจ้งเตือนต่อไป

ผลการทดลองเริ่มตั้งแต่การทดลองการทำงานของโปรแกรมเพื่อทดสอบฟังก์ชั่นการทำงานที่ได้ตั้งค่าไว้ทำงานได้อย่างถูกต้อง การตั้งค่าเทรชโฮลในการค้นหาผู้บุกรุก การทดลองการทำงานของโปรแกรมในส่วนของการจับคู่แม่แบบ (Template Matching) และการทดสอบการทำงานของโปรแกรมในสภาพแวดล้อมที่แตกต่างกัน เพื่อวิเคราะห์ว่าโปรแกรมตรวจสอบผู้บุกรุกโดยการใช้กล้องวีดีโอสามารถทำงานได้ดีในสภาพแวดล้อมแบบใดและเกิดข้อผิดพลาดในสภาพแวดล้อมแบบใด

4.1 ขั้นตอนการทำงานของโปรแกรม

1. เมื่อเปิดโปรแกรมการใช้งานให้เลือกกดปุ่ม Initial เพื่อให้โปรแกรมทำงานติดต่อกับกล้องหลังจากนั้นให้ทำการเลือกปุ่ม BG เพื่อทำการบันทึกภาพเริ่มต้นเพื่อใช้ในการเปรียบเทียบเมื่อทำการกดเลือกปุ่มจะได้ภาพเริ่มต้นขึ้นมาและจะแสดงสถานะเป็น "NORMAL" ดังรูปที่ 4.1

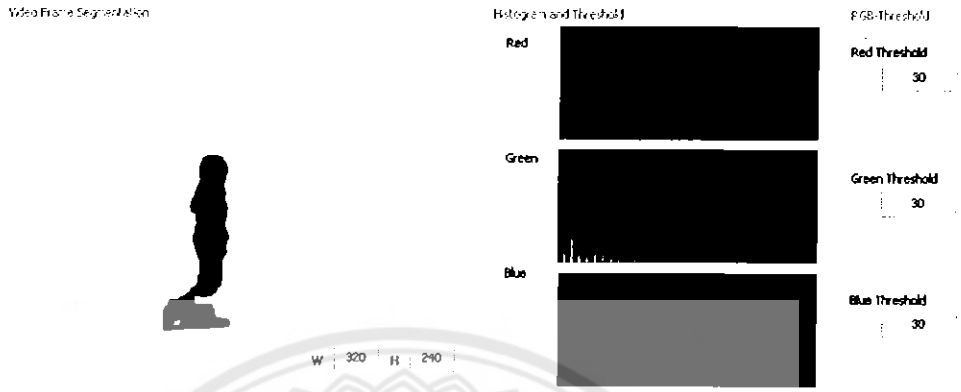


รูปที่ 4.1 แสดงรูปภาพเริ่มต้นที่ใช้ในการเปรียบเทียบและสถานะเริ่มต้น

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

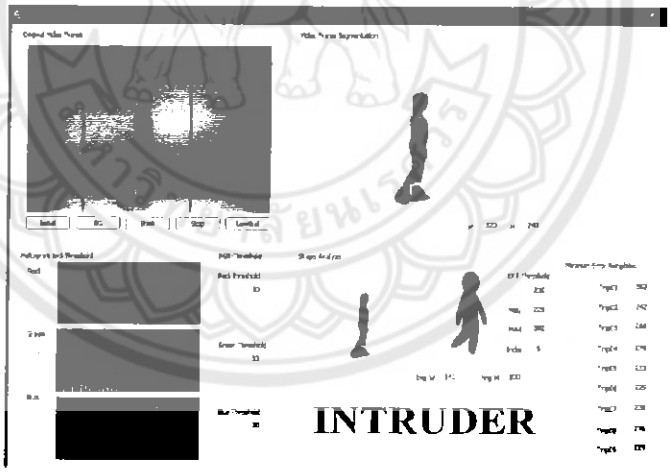
5200049

2. เลือก start เพื่อเริ่มการใช้งานโปรแกรม โดยกล้องจะทำการจับภาพ เมื่อมีผู้บุกรุกหรือวัตถุเคลื่อนที่ผ่านกล้องโปรแกรมจะทำการแยกคนหรือวัตถุออกจากพื้นหลัง และแสดงค่าของสีจากกราฟเทรชโฮลของสีแดง สีเขียว และสีน้ำเงิน ซึ่งสามารถกำหนดค่าเทรชโฮลได้ ดังรูปที่ 4.2



รูปที่ 4.2 แสดงภาพคนหรือวัตถุที่แยกออกจากพื้นหลังและกราฟแสดงค่าเทรชโฮลของคนหรือวัตถุ

3. หลังจากนั้นโปรแกรมจะทำการแปลงภาพคนหรือวัตถุที่โปรแกรมทำการแยกได้เป็นภาพแบบไบนารีและทำการเปรียบเทียบกับภาพที่ได้ทำการเก็บไว้ เมื่อมีลักษณะรูปร่างเหมือนกับเทมเพลตที่เก็บไว้โปรแกรมจะทำการแจ้งเตือนให้ทราบว่ามีผู้บุกรุก ดังรูปที่ 4.3



รูปที่ 4.3 แสดงโปรแกรมแจ้งเตือนมีผู้บุกรุก

4.2 ผลการทดลองการตั้งค่าเทรชโฮล ในการค้นหาผู้บุกรุก

เนื่องจากการกำหนดค่าเทรชโฮลนั้นมีความสำคัญกับการวิเคราะห์ภาพเพื่อแยกวัตถุออกจากภาพ ดังนั้น ในการทดลองที่ 4.2 นี้ จะทำการทดลองเพื่อค้นหาค่าเทรชโฮลที่เหมาะสมกับโปรแกรม เพื่อจะได้นำไปใช้ในการทดลองในขั้นต่อไป

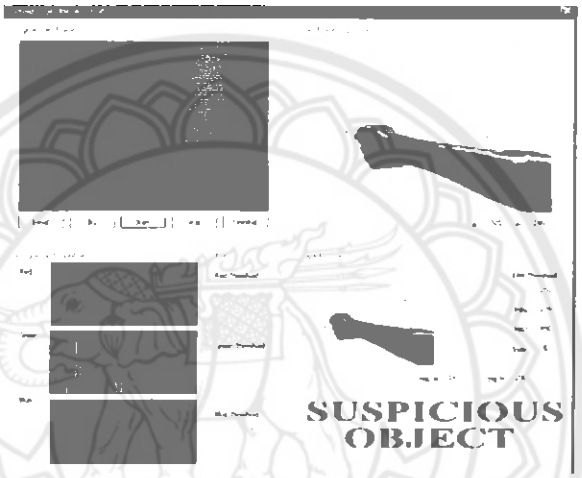
เทรชโฮลในที่นี้ คือ ค่าเทรชโฮลของค่าสี R , G และ B โดยใช้เทรชโฮลเป็นตัวกำหนดเพื่อตรวจสอบค่าผลต่างที่ได้จากการหาค่าโดยวิธีผลต่างแบบคงตัว ถ้าหากค่าสัมบูรณ์ของผลต่างที่

ได้ว่า ไม่มีวัตถุภายในรูปภาพหรือวัตถุมีขนาดเล็กพอที่จะสามารถละทิ้งไปได้ รวมทั้งอาจจะเป็นสัญญาณรบกวนที่เกิดขึ้นกับภาพ

ในทางกลับกันเมื่อค่าผลต่างที่ได้จากการแยกวัตถุออกจากภาพพื้นหลังมีค่ามากกว่าค่าเทรชโฮลที่กำหนดแสดงว่าจะต้องมีวัตถุปรากฏในรูปภาพ (แต่ยังไม่สามารถระบุได้ว่าเป็นวัตถุชนิดใด) ในการทดลองนี้จะทำการกำหนดค่าเทรชโฮลให้กับค่าสี R , G และ B ให้เท่ากันโดยเริ่มจาก

4.2.1 กำหนดค่าเทรชโฮลเท่ากับ 20

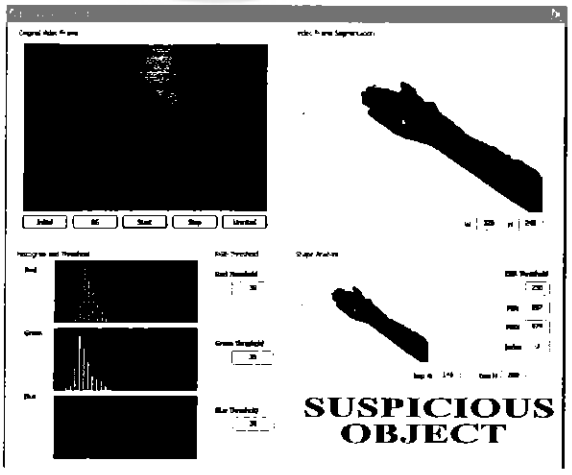
จากการกำหนดค่าเทรชโฮลในข้อนี้ พบว่าสามารถเห็นวัตถุที่ปรากฏอย่างชัดเจน แต่จะมีภาพพื้นหลังในหลายจุดที่ถูกแยกออกมาด้วยเช่นกัน (จุดหรือเม็ดสีที่ปรากฏบนภาพ) ดังรูปที่ 4.4



รูปที่ 4.4 แสดงภาพที่ได้จากการกำหนดค่าเทรชโฮลเท่ากับ 20 ทั้ง R, G และ B

4.2.2 กำหนดค่าเทรชโฮลเท่ากับ 30

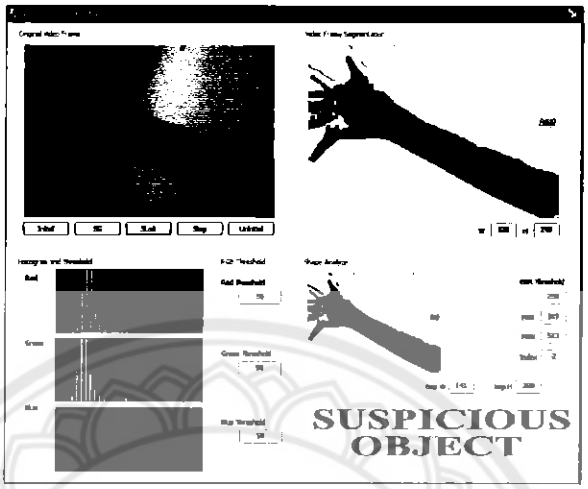
จากการกำหนดค่าเทรชโฮลในข้อนี้ พบว่าสามารถเห็นวัตถุที่ปรากฏอย่างชัดเจน แต่จะมีภาพพื้นหลังในบางจุดที่ถูกแยกออกมาด้วยเช่นกันแต่น้อยกว่าการทดลอง 4.2.1 ดังรูปที่ 4.5



รูปที่ 4.5 แสดงภาพที่ได้จากการกำหนดค่าเทรชโฮลเท่ากับ 30 ทั้ง R, G และ B

4.2.3 กำหนดค่าเทรชโฮลเท่ากับ 50

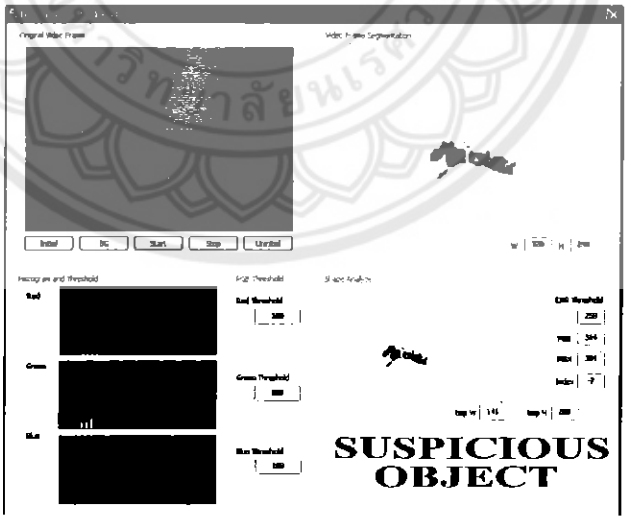
จากการกำหนดค่าเทรชโฮลในข้อนี้ พบว่าสามารถเห็นวัตถุที่ปรากฏมีลักษณะขาดหายไปในบางส่วนและมีภาพพื้นหลังในบางจุดถูกแยกออกมาด้วย ดังรูปที่ 4.6



รูปที่ 4.6 แสดงภาพที่ได้จากการกำหนดค่าเทรชโฮลเท่ากับ 50 ทั้ง R , G และ B

4.2.4 กำหนดค่าเทรชโฮลเท่ากับ 100

จากการกำหนดค่าเทรชโฮลในข้อนี้ พบว่าสามารถเห็นวัตถุที่ปรากฏมีลักษณะขาดหายไปในหลายส่วนๆ ดังรูปที่ 4.7

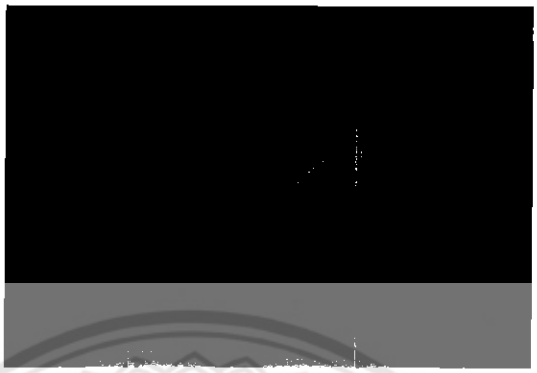


รูปที่ 4.7 แสดงภาพที่ได้จากการกำหนดค่าเทรชโฮลเท่ากับ 100 ทั้ง R , G และ B

จากการทดลอง ทั้ง 4 การทดลองในลักษณะเดียวกันในหลายครั้ง พบว่า การตั้งค่าเทรชโฮลเท่ากับ 30 นั้น สามารถเห็นวัตถุที่ถูกแบ่งแยกออกอย่างชัดเจน ดังนั้น ในการทดลองขั้นต่อๆ ไปของโครงงานนี้ จึงได้เลือกใช้ค่าเทรชโฮลของ R , G , และ B เท่ากับ 30

4.3 ผลการทดลองการทำงานของโปรแกรมในส่วนของการจับคู่แม่แบบ (Template Matching)

4.3.1 การทดลองการจับคู่แม่แบบในลักษณะที่ 1



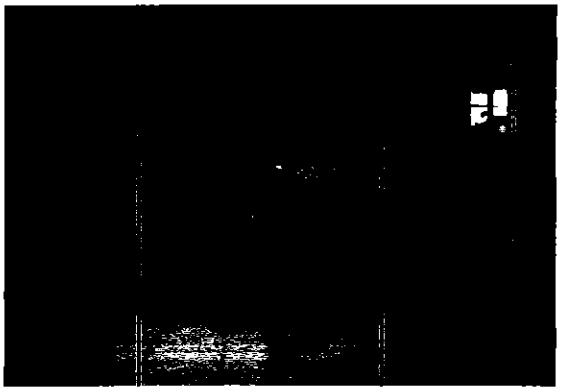
รูปที่ 4.8 ภาพที่รับจากกล้องในลักษณะที่ 1

จากการทดลองในลักษณะท่าทางที่ 1 เมื่อโปรแกรมทำการประมวลผลจะได้ค่าความแตกต่าง R(x,y) ของแต่ละเทมเพลตดังตารางที่ 4.1 ซึ่งค่าความแตกต่างในแต่ละเทมเพลตจะมีค่าไม่เท่ากันขึ้นอยู่กับลักษณะท่าทางของภาพที่นำมาทำการจับคู่แม่แบบ (template matching) โดยเทมเพลตที่มีค่าความแตกต่างน้อยที่สุดจะตรงกับลักษณะท่าทางที่ได้ทำการทดลองมากที่สุด โดยภาพในท่าทางที่ 1 จะตรงกับเทมเพลตที่ 6 เนื่องจากมีค่าความแตกต่างน้อยที่สุด

ตารางที่ 4.1 แสดงผลค่า R(x,y) ในแต่ละเทมเพลตของท่าทางที่ 1

1. R(x,y) = 302	2. R(x,y) = 342	3. R(x,y) = 266
4. R(x,y) = 275	5. R(x,y) = 233	**6. R(x,y) = 225**
7. R(x,y) = 236	8. R(x,y) = 276	9. R(x,y) = 259

4.3.2 การทดลองการจับคู่แม่แบบในลักษณะที่ 2



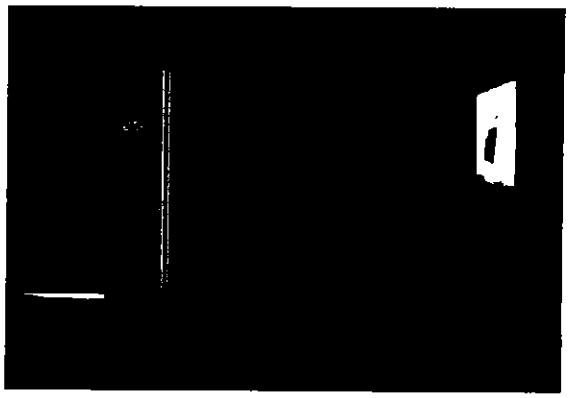
รูปที่ 4.9 ภาพที่รับจากกล้องในท่าทางที่ 2

จากการทดลองในลักษณะท่าทางที่ 2 เมื่อโปรแกรมทำการประมวลผลจะได้ค่าความแตกต่าง R(x,y) ของแต่ละเทมเพลตดังตารางที่ 4.2 ซึ่งค่าความแตกต่างในแต่ละเทมเพลตจะมีค่าไม่เท่ากันขึ้นอยู่กับลักษณะท่าทางของภาพที่นำมาทำการจับคู่แม่แบบ (template matching) โดยเทมเพลตที่มีค่าความแตกต่างน้อยที่สุดจะตรงกับลักษณะท่าทางที่ได้ทำการทดลองมากที่สุด โดยภาพในท่าทางที่ 2 จะตรงกับเทมเพลตที่ 9 เนื่องจากมีค่าความแตกต่างน้อยที่สุด

ตารางที่ 4.2 แสดงผลค่า R(x,y) ในแต่ละเทมเพลตของท่าทางที่ 2

1. R(x,y) = 300	2. R(x,y) = 328	3. R(x,y) = 278
4. R(x,y) = 281	5. R(x,y) = 237	6. R(x,y) = 296
7. R(x,y) = 243	8. R(x,y) = 284	**9. R(x,y) = 227**

4.3.3 การทดลองการจับคู่แม่แบบในลักษณะที่ 3



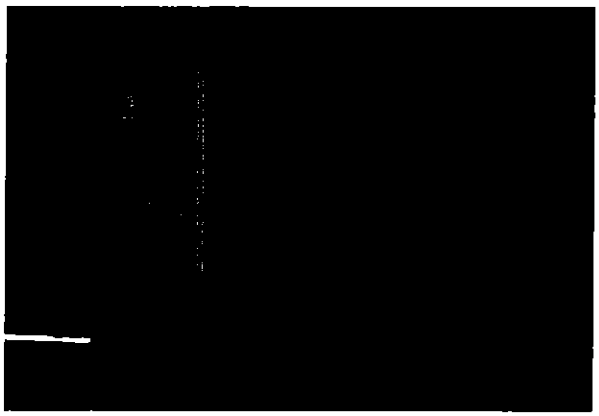
รูปที่ 4.10 ภาพที่รับจากกล้องในท่าทางที่ 3

จากการทดลองในลักษณะท่าทางที่ 3 เมื่อโปรแกรมทำการประมวลผลจะได้ค่าความแตกต่าง R(x,y) ของแต่ละเทมเพลตดังตารางที่ 4.3 ซึ่งค่าความแตกต่างในแต่ละเทมเพลตจะมีค่าไม่เท่ากันขึ้นอยู่กับลักษณะท่าทางของภาพที่นำมาทำการจับคู่แม่แบบ (template matching) โดยเทมเพลตที่มีค่าแตกต่างน้อยที่สุดจะตรงกับลักษณะท่าทางที่ได้ทำการทดลองมากที่สุด โดยภาพในท่าทางที่ 3 จะตรงกับเทมเพลตที่ 4 เนื่องจากมีค่าความแตกต่างน้อยที่สุด

ตารางที่ 4.3 แสดงผลค่า R(x,y) ในแต่ละเทมเพลตของท่าทางที่ 3

1. R(x,y) = 286	2. R(x,y) = 315	3. R(x,y) = 266
4. R(x,y) = 207	5. R(x,y) = 218	6. R(x,y) = 262
7. R(x,y) = 234	8. R(x,y) = 275	9. R(x,y) = 272

4.3.4 การทดลองการจับคู่แม่แบบในลักษณะที่ 4



รูปที่ 4.11 ภาพที่รับจากกล้องในท่าทางที่ 4

จากการทดลองในลักษณะท่าทางที่ 4 เมื่อโปรแกรมทำการประมวลผลจะได้ค่าความแตกต่าง R(x,y) ของแต่ละเทมเพลตดังตารางที่ 4.4 ซึ่งค่าความแตกต่างในแต่ละเทมเพลตจะมีค่าไม่เท่ากันขึ้นอยู่กับลักษณะท่าทางของภาพที่นำมาทำการจับคู่แม่แบบ (template matching) โดยเทมเพลตที่มีค่าความแตกต่างน้อยที่สุดจะตรงกับลักษณะท่าทางที่ได้ทำการทดลองมากที่สุด โดยภาพในท่าทางที่ 4 จะตรงกับเทมเพลตที่ 2 เนื่องจากมีค่าความแตกต่างน้อยที่สุด

ตารางที่ 4.4 แสดงผลค่า R(x,y) ในแต่ละเทมเพลตของท่าทางที่ 4

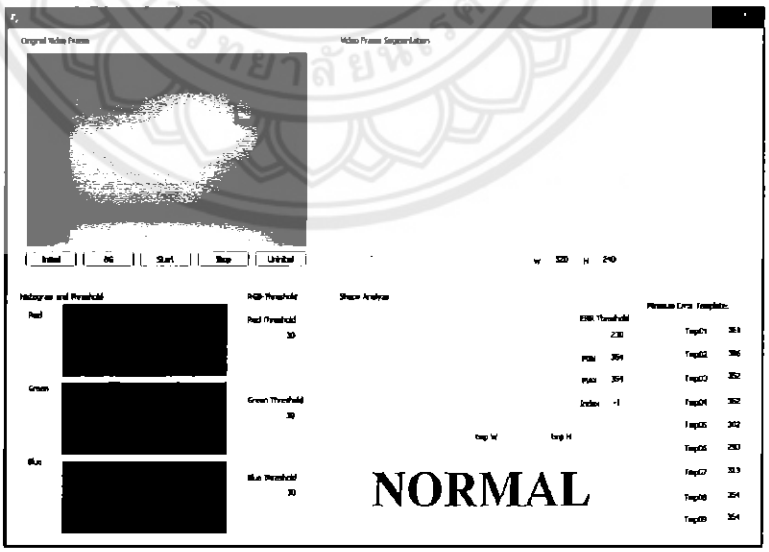
1. R(x,y) = 241	**2. R(x,y) = 165**	3. R(x,y) = 219
4. R(x,y) = 218	5. R(x,y) = 284	6. R(x,y) = 175
7. R(x,y) = 178	8. R(x,y) = 235	9. R(x,y) = 230

จากการทดลองการทำงานของโปรแกรมเพื่อทดสอบหาค่าความแตกต่างของแต่ละเทมเพลตเมื่อนำมาเปรียบเทียบกับรูปภาพในลักษณะต่างๆ ค่าความแตกต่างที่ได้จะมีค่ามากน้อยขึ้นอยู่กับท่าทางในขณะนั้นว่ามีความคล้ายคลึงกับเทมเพลตใดมากที่สุด โดยเมื่อทำการเปรียบเทียบถ้าค่า R(x,y) ที่ได้มีค่าน้อย แสดงว่าเทมเพลตนั้นมีความเหมือนกับรูปภาพที่นำมาเปรียบเทียบมากที่สุด ซึ่งจากการทดลองการทำงานของโปรแกรม จะเห็นได้ว่าโปรแกรมสามารถทำงานประมวลผลในการจับคู่แม่แบบ (template matching) ได้ประมาณ 80%

4.4 ผลการทดลองการใช้โปรแกรมตรวจสอบผู้บุกรุก

การทดลองใช้โปรแกรมตรวจสอบผู้บุกรุกโดยการประมวลผลภาพ ได้ทำการทดลองโดยทำการติดตั้งกล้องในพื้นที่ที่ต้องการทำการตรวจสอบ พบว่าในขณะที่สถานการณ์ปกติไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้องโปรแกรมจะทำการตรวจสอบตลอดเวลาและไม่มีการแจ้งเตือน แต่เมื่อมีคนเคลื่อนที่ผ่านกล้องในพื้นที่ที่ทำการติดตั้งกล้องเมื่อโปรแกรมประมวลผลพบว่าเกิดความแตกต่างของภาพเริ่มต้นกับภาพปัจจุบันโปรแกรมจะทำการเปลี่ยนสถานะและทำการเปรียบเทียบกับเทมเพลตแม่แบบเมื่อพบว่าเป็นรูปร่างคนโปรแกรมจะทำการแจ้งเตือนด้วยเสียง หรือเมื่อมีวัตถุเคลื่อนที่ผ่านกล้องเมื่อโปรแกรมประมวลผลพบว่าเป็นวัตถุโปรแกรมจะทำการเปลี่ยนสถานะแต่ไม่มีเสียงแจ้งเตือน โดยการทดลองได้แบ่งสถานการณ์ที่ใช้ในการทดลองดังนี้

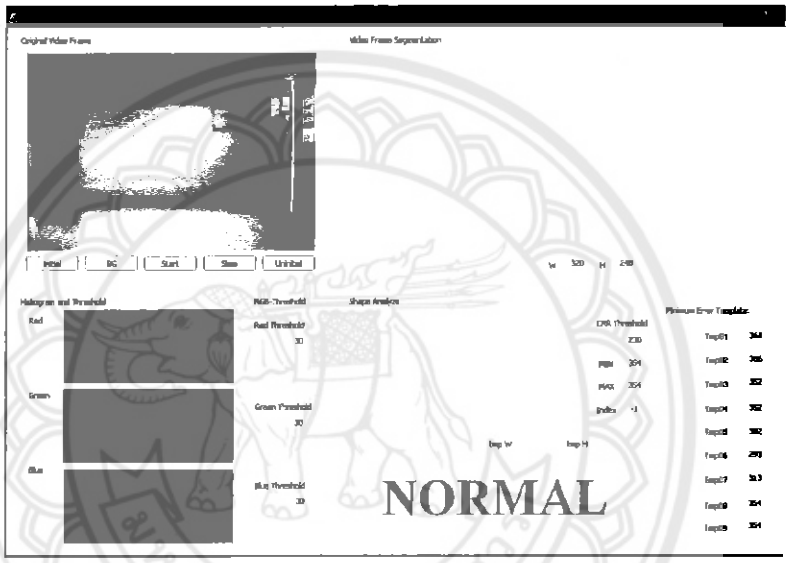
สถานการณ์ที่ 1 ใช้งานโปรแกรมในช่วงเวลากลางวันไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้องและระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 5 เมตร



รูปที่ 4.12 แสดงหน้าต่างของโปรแกรมที่ไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้องในช่วงเวลากลางวันระยะตรวจสอบ 5 เมตร

จากผลการทดลองเมื่อไม่มีการเคลื่อนที่ของคนหรือวัตถุเคลื่อนที่ผ่านกล้อง โปรแกรมจะแสดงสถานะเป็น "NORMAL" และไม่มีการแจ้งเตือนของโปรแกรม เนื่องจากโปรแกรมเมื่อทำการประมวลผลไม่พบความแตกต่างของภาพ เมื่อหาค่าผลต่างพื้นหลังแบบคงตัวของภาพพื้นหลังและภาพที่เปรียบเทียบได้ผลต่างมีค่าเป็นศูนย์หรือมีค่าน้อยมาก ในการแยกวิเคราะห์วัตถุจึงไม่ปรากฏวัตถุออกมาในส่วนของ Video Frame Segmentation และค่า index ที่แสดงเป็น -1 เนื่องจากโปรแกรมทำการประมวลผลแล้วไม่พบวัตถุ

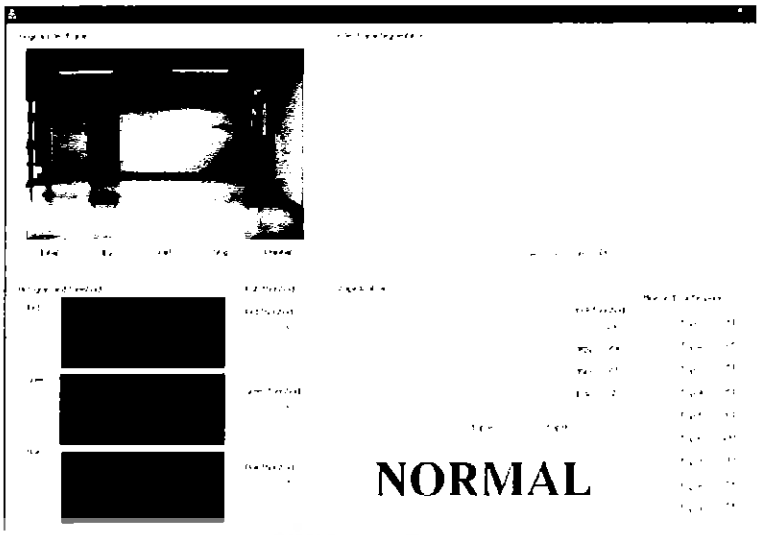
สถานการณ์ที่ 2 ใช้งานโปรแกรมในช่วงเวลากลางวันไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้องและระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 7 เมตร



รูปที่ 4.13 แสดงหน้าต่างของโปรแกรมที่ไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้องในช่วงเวลากลางวันระยะตรวจสอบ 7 เมตร

จากผลการทดลองเมื่อไม่มีการเคลื่อนที่ของคนหรือวัตถุเคลื่อนที่ผ่านกล้อง โปรแกรมจะแสดงสถานะเป็น "NORMAL" และไม่มีการแจ้งเตือนของโปรแกรม เนื่องจากโปรแกรมเมื่อทำการประมวลผลไม่พบความแตกต่างของภาพ เมื่อหาค่าผลต่างพื้นหลังแบบคงตัวของภาพพื้นหลังและภาพที่เปรียบเทียบได้ผลต่างมีค่าเป็นศูนย์หรือมีค่าน้อยมาก ในการแยกวิเคราะห์วัตถุจึงไม่ปรากฏวัตถุออกมาในส่วนของ Video Frame Segmentation และค่า index ที่แสดงเป็น -1 เนื่องจากโปรแกรมทำการประมวลผลแล้วไม่พบวัตถุ

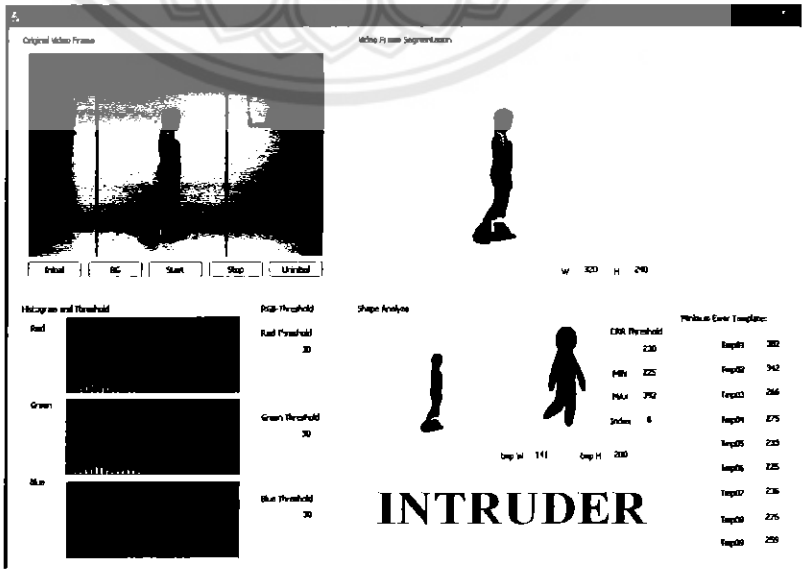
สถานการณ์ที่ 3 ใช้งานโปรแกรมในช่วงเวลากลางวันไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้องและระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 10 เมตร



รูปที่ 4.14 แสดงหน้าต่างของโปรแกรมที่ไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้อง
ในช่วงเวลากลางวันระยะตรวจสอบ 10 เมตร

จากผลการทดลองเมื่อไม่มีการเคลื่อนที่ของคนหรือวัตถุเคลื่อนที่ผ่านกล้อง โปรแกรมจะแสดงสถานะเป็น "NORMAL" และไม่มีการแจ้งเตือนของโปรแกรม เนื่องจากโปรแกรมเมื่อทำการประมวลผลไม่พบความแตกต่างของภาพ เมื่อหาค่าผลต่างพื้นหลังแบบคงตัวของภาพพื้นหลังและภาพที่เปรียบเทียบได้ผลต่างมีค่าเป็นศูนย์หรือมีค่าน้อยมาก ในการแยกวิเคราะห์วัตถุจึงไม่ปรากฏวัตถุออกมาในส่วนของ Video Frame Segmentation และค่า index ที่แสดงเป็น -1 เนื่องจากโปรแกรมทำการประมวลผลแล้วไม่พบวัตถุ

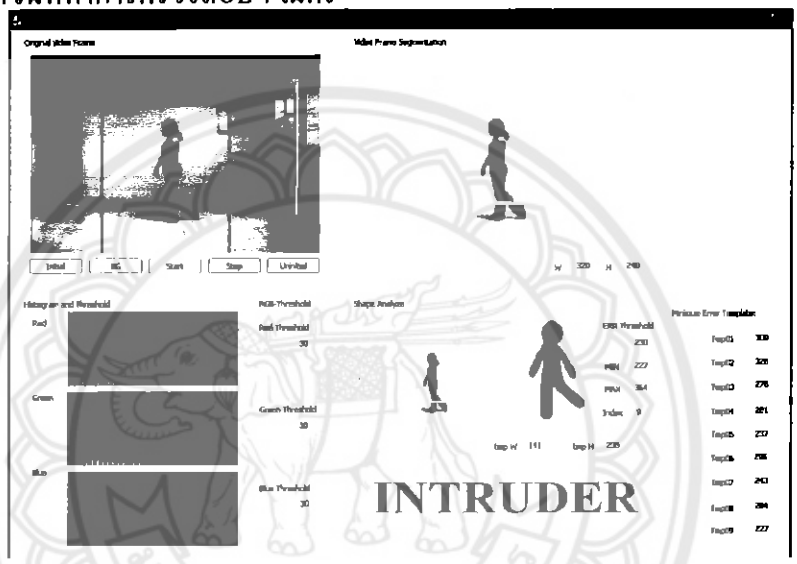
สถานการณ์ที่ 4 ใช้งานโปรแกรมในช่วงเวลากลางวันมีคนเดินผ่านกล้อง และระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 5 เมตร



รูปที่ 4.15 แสดงหน้าต่างของโปรแกรมที่คนเคลื่อนที่ผ่านกล้อง
ในช่วงเวลากลางวันระยะตรวจสอบ 5 เมตร

จากผลการทดลองเมื่อมีคนเดินผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "INTRUDER" และมีเสียงเตือนจากโปรแกรม โดยค่า min ที่แสดงในส่วนของ Shape Analyze ได้มาจากการหาค่าที่น้อยที่สุดของ Minimum Error template ทั้ง 9 template จากช่องด้านขวามือ ซึ่งในสถานการณ์นี้ตรงกับเทมเพลตที่ 6 โดยมีค่าของ Minimum Error ของ เทมเพลต เท่ากับ 226 และค่า index ที่แสดงเป็น 6 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าภาพที่เปรียบเทียบมีลักษณะตรงกับเทมเพลตที่ 6

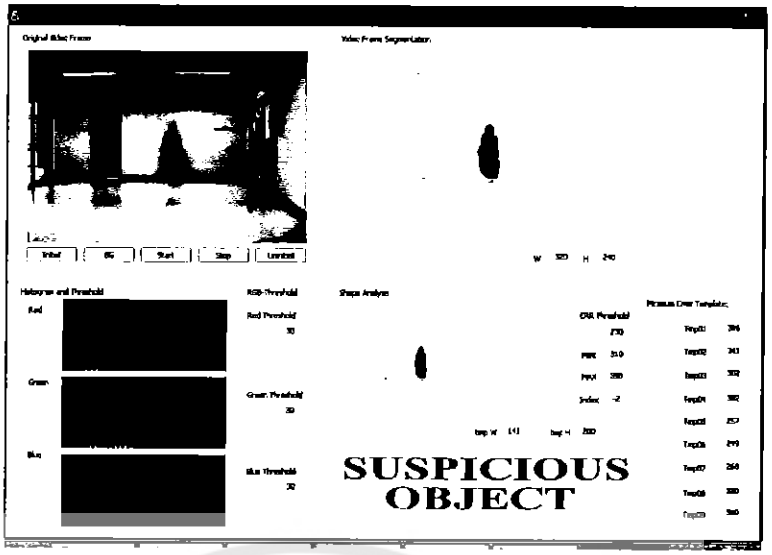
สถานการณ์ที่ 5 ใช้งานโปรแกรมในช่วงเวลากลางวันมีคนเดินผ่านกล้อง และระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 7 เมตร



รูปที่ 4.16 แสดงหน้าต่างของโปรแกรมที่คนเคลื่อนที่ผ่านกล้อง
ในช่วงเวลากลางวันระยะตรวจสอบ 7 เมตร

จากผลการทดลองเมื่อมีคนเดินผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "INTRUDER" และมีเสียงเตือนจากโปรแกรม โดยค่า min ที่แสดงในส่วนของ Shape Analyze ได้มาจากการหาค่าที่น้อยที่สุดของ Minimum Error template ทั้ง 9 template จากช่องด้านขวามือ ซึ่งในสถานการณ์นี้ตรงกับเทมเพลตที่ 9 โดยมีค่าของ Minimum Error ของ เทมเพลต เท่ากับ 227 และค่า index ที่แสดงเป็น 9 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าภาพที่เปรียบเทียบมีลักษณะตรงกับเทมเพลตที่ 9

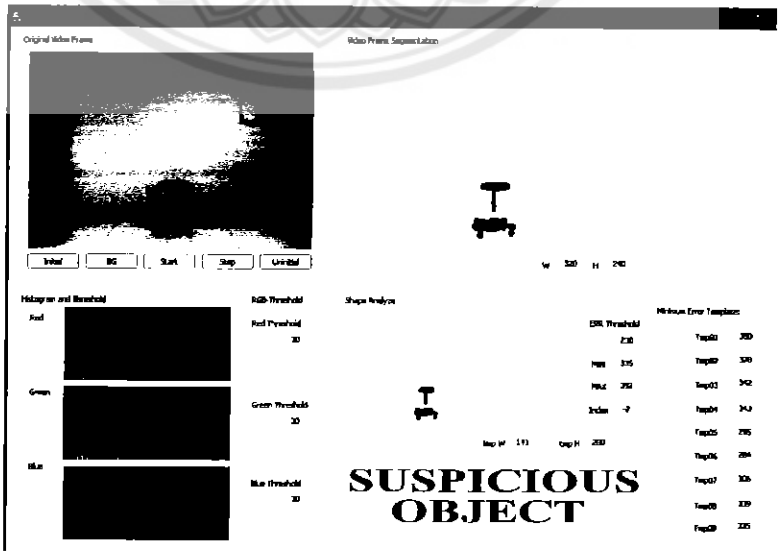
สถานการณ์ที่ 6 ใช้งานโปรแกรมในช่วงเวลากลางวันมีคนเดินผ่านกล้อง และระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 10 เมตร



รูปที่ 4.17 แสดงหน้าต่างของโปรแกรมที่คนเคลื่อนที่ผ่านกล้อง
ในช่วงเวลากลางวันระยะตรวจสอบ 10 เมตร

จากผลการทดลองเมื่อมีคนเดินผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "SUSPICIOUS OBJECT" และไม่มีเสียงเตือนจากโปรแกรม เนื่องจากโปรแกรมไม่สามารถค้นพบรูปร่างได้เพราะภาพที่ได้มีขนาดน้อยกว่าเทมเพลตที่ใช้ในการเปรียบเทียบ โดยค่า min ที่แสดงในส่วนของ Shape Analyze มีค่าเกินกว่าค่า Err Threshold ที่ได้ทำการตั้งค่าไว้ ซึ่งในสถานการณ์นี้ค่า min มีค่าเท่ากับ 310 และค่า index ที่แสดงเป็น -2 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าเป็นวัตถุ

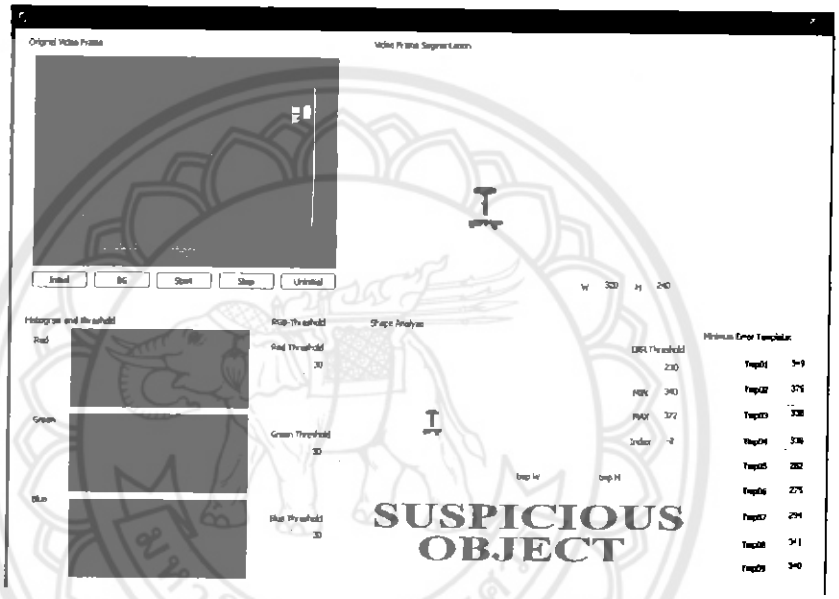
สถานการณ์ที่ 7 ใช้งานโปรแกรมในช่วงเวลากลางวันมีวัตถุเคลื่อนที่ผ่านกล้อง และระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 5 เมตร



รูปที่ 4.18 แสดงหน้าต่างของโปรแกรมที่มีวัตถุเคลื่อนที่ผ่านกล้อง
ในช่วงเวลากลางวันระยะตรวจสอบ 5 เมตร

จากผลการทดลองเมื่อการเคลื่อนที่ของวัตถุผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "SUSPICIOUS OBJECT" และไม่มีเสียงเตือนจากโปรแกรม เนื่องจากโปรแกรมไม่สามารถค้นพบรูปร่างได้ โดยค่า min ที่แสดงในส่วนของ Shape Analyze มีค่าเกินกว่าค่า Err Threshold ที่ได้ทำการตั้งค่าไว้ ซึ่งในสถานการณ์นี้ค่า min มีค่าเท่ากับ 335 และค่า index ที่แสดงเป็น -2 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าเป็นวัตถุ

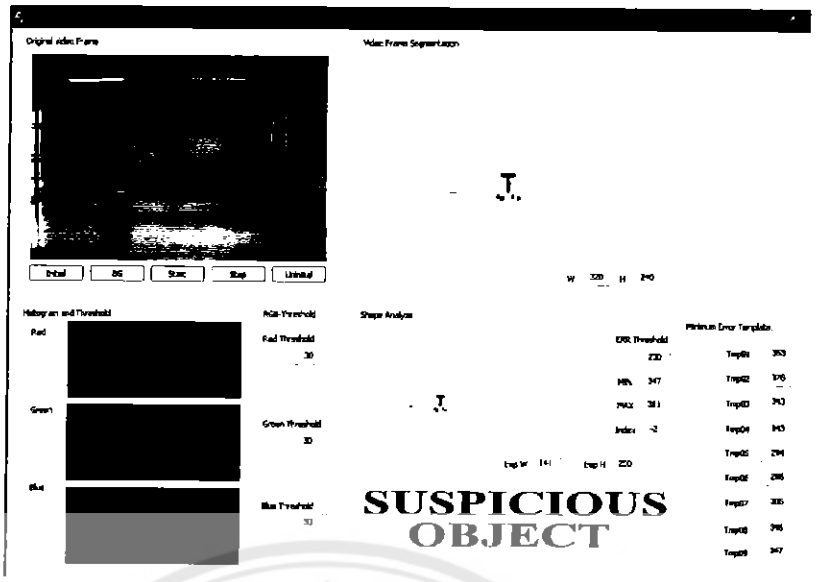
สถานการณ์ที่ 8 ใช้งานโปรแกรมในช่วงเวลากลางวันมีวัตถุเคลื่อนที่ผ่านกล้อง และระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 7 เมตร



รูปที่ 4.19 แสดงหน้าต่างของโปรแกรมที่มีวัตถุเคลื่อนที่ผ่านกล้อง ในช่วงเวลากลางวันระยะตรวจสอบ 7 เมตร

จากผลการทดลองเมื่อการเคลื่อนที่ของวัตถุผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "SUSPICIOUS OBJECT" และไม่มีเสียงเตือนจากโปรแกรม เนื่องจากโปรแกรมไม่สามารถค้นพบรูปร่างได้ โดยค่า min ที่แสดงในส่วนของ Shape Analyze มีค่าเกินกว่าค่า Err Threshold ที่ได้ทำการตั้งค่าไว้ ซึ่งในสถานการณ์นี้ค่า min มีค่าเท่ากับ 340 และค่า index ที่แสดงเป็น -2 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าเป็นวัตถุ

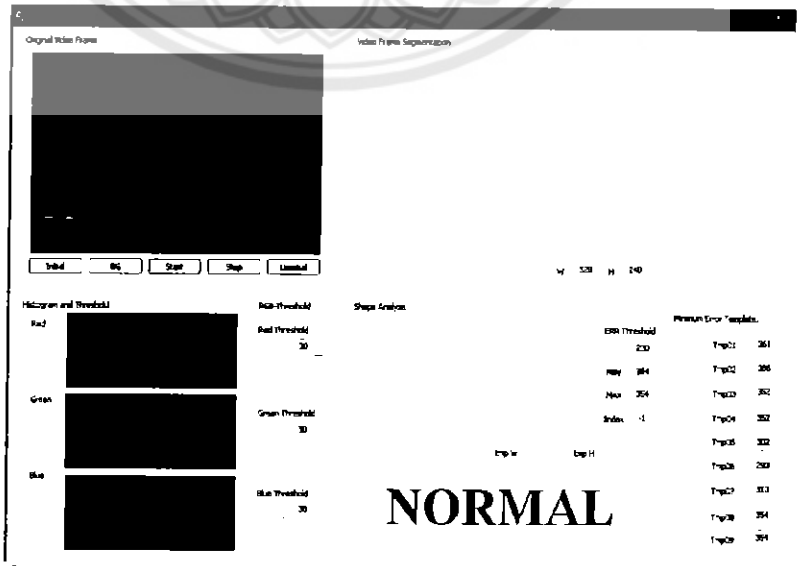
สถานการณ์ที่ 9 ใช้งานโปรแกรมในช่วงเวลากลางวันมีวัตถุเคลื่อนที่ผ่านกล้อง และระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 10 เมตร



รูปที่ 4.20 แสดงหน้าต่างของโปรแกรมที่มีวัตถุเคลื่อนที่ผ่านกล้อง
ในช่วงเวลากลางวันระยะตรวจสอบ 10 เมตร

จากผลการทดลองเมื่อการเคลื่อนที่ของวัตถุผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "SUSPICIOUS OBJECT" และไม่มีเสียงเตือนจากโปรแกรม เนื่องจากโปรแกรมไม่สามารถค้นพบรูปร่างได้ โดยค่า min ที่แสดงในส่วนของ Shape Analyze มีค่าเกินกว่าค่า Err Threshold ที่ได้ทำการตั้งค่าไว้ ซึ่งในสถานการณ์นี้ค่า min มีค่าเท่ากับ 347 และค่า index ที่แสดงเป็น -2 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าเป็นวัตถุ

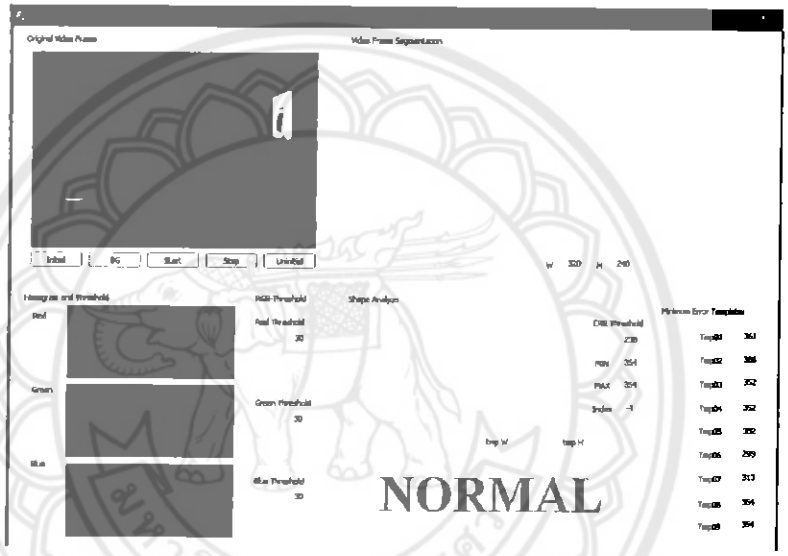
สถานการณ์ที่ 10 ใช้งานโปรแกรมในช่วงเวลากลางคืน ไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้อง และระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 5 เมตร



รูปที่ 4.21 แสดงหน้าต่างของโปรแกรมที่ไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้อง
ในช่วงเวลากลางคืนระยะตรวจสอบ 5 เมตร

จากผลการทดลองเมื่อไม่มีการเคลื่อนที่ของคนหรือวัตถุเคลื่อนที่ผ่านกล้อง โปรแกรมจะแสดงสถานะเป็น "NORMAL" และไม่มีการแจ้งเตือนของโปรแกรม เนื่องจากโปรแกรมเมื่อทำการประมวลผลไม่พบความแตกต่างของภาพ เมื่อหาค่าผลต่างพื้นหลังแบบคงตัวของภาพพื้นหลังและภาพที่เปรียบเทียบได้ผลต่างมีค่าเป็นศูนย์หรือมีค่าน้อยมาก ในการแยกวิเคราะห์วัตถุจึงไม่ปรากฏวัตถุออกมาในส่วนของ Video Frame Segmentation และค่า index ที่แสดงเป็น -1 เนื่องจากโปรแกรมทำการประมวลผลแล้วไม่พบวัตถุ

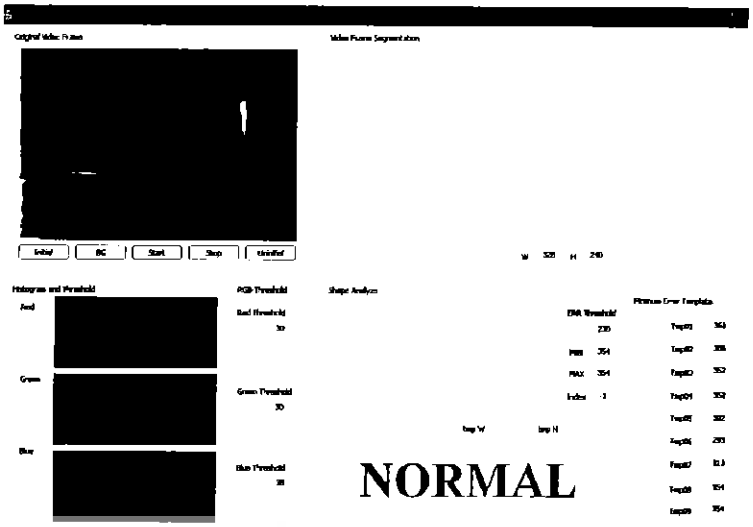
สถานการณ์ที่ 11 ใช้งานโปรแกรมในช่วงเวลากลางคืนไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้อง และระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 7 เมตร



รูปที่ 4.22 แสดงหน้าต่างของโปรแกรมที่ไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้องในช่วงเวลากลางคืนระยะตรวจสอบ 7 เมตร

จากผลการทดลองเมื่อไม่มีการเคลื่อนที่ของคนหรือวัตถุเคลื่อนที่ผ่านกล้อง โปรแกรมจะแสดงสถานะเป็น "NORMAL" และไม่มีการแจ้งเตือนของโปรแกรม เนื่องจากโปรแกรมเมื่อทำการประมวลผลไม่พบความแตกต่างของภาพ เมื่อหาค่าผลต่างพื้นหลังแบบคงตัวของภาพพื้นหลังและภาพที่เปรียบเทียบได้ผลต่างมีค่าเป็นศูนย์หรือมีค่าน้อยมาก ในการแยกวิเคราะห์วัตถุจึงไม่ปรากฏวัตถุออกมาในส่วนของ Video Frame Segmentation และค่า index ที่แสดงเป็น -1 เนื่องจากโปรแกรมทำการประมวลผลแล้วไม่พบวัตถุ

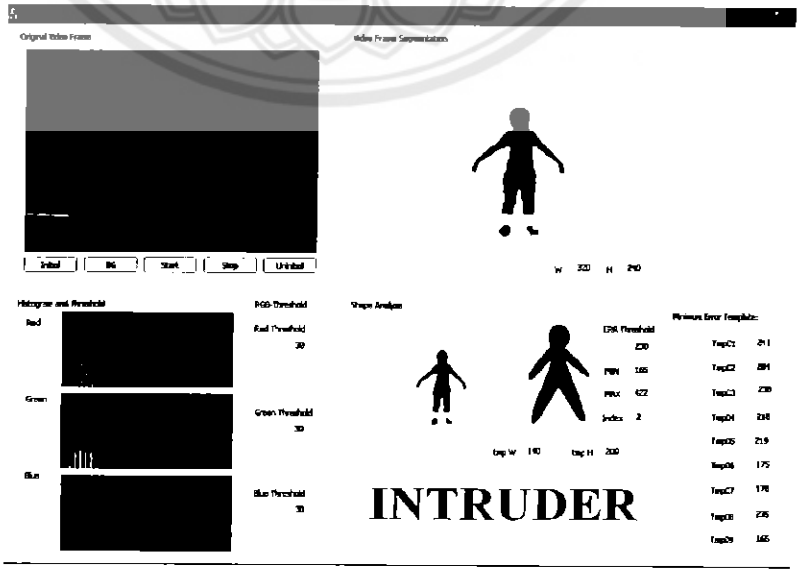
สถานการณ์ที่ 12 ใช้งานโปรแกรมในช่วงเวลากลางคืนไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้อง และระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 10 เมตร



รูปที่ 4.23 แสดงหน้าต่างของโปรแกรมที่ไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้อง
ในช่วงเวลากลางคืนระยะตรวจสอบ 10 เมตร

จากผลการทดลองเมื่อไม่มีการเคลื่อนที่ของคนหรือวัตถุเคลื่อนที่ผ่านกล้อง โปรแกรมจะแสดงสถานะเป็น "NORMAL" และไม่มีการแจ้งเตือนของโปรแกรม เนื่องจากโปรแกรมเมื่อทำการประมวลผลไม่พบความแตกต่างของภาพ เมื่อหาค่าผลต่างพื้นหลังแบบคงตัวของภาพพื้นหลังและภาพที่เปรียบเทียบได้ผลต่างมีค่าเป็นศูนย์หรือมีค่าน้อยมาก ในการแยกวิเคราะห์วัตถุจึงไม่ปรากฏวัตถุออกมาในส่วนของ Video Frame Segmentation และค่า index ที่แสดงเป็น -1 เนื่องจากโปรแกรมทำการประมวลผลแล้วไม่พบวัตถุ

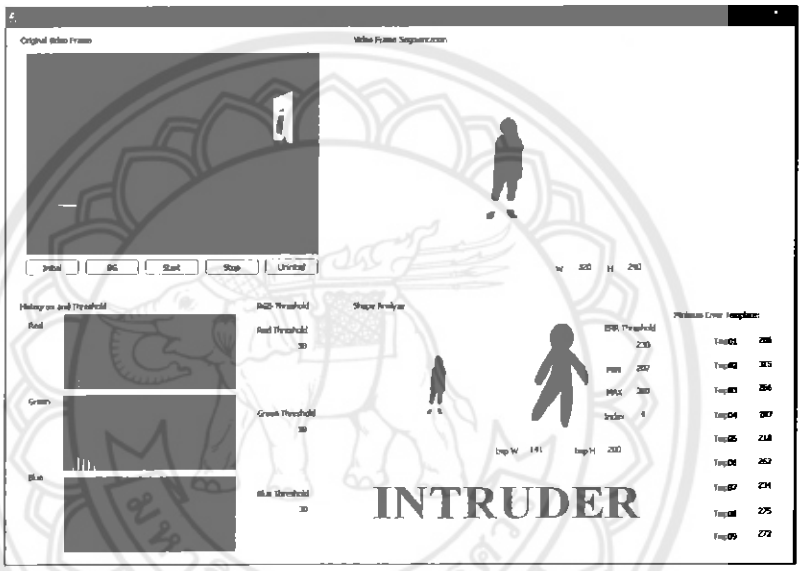
สถานการณ์ที่ 13 ใช้งานโปรแกรมในช่วงเวลากลางคืนมีคนเดินผ่านกล้อง และระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 5 เมตร



รูปที่ 4.24 แสดงหน้าต่างของโปรแกรมที่มีคนเคลื่อนที่ผ่านกล้อง
ในช่วงเวลากลางคืนระยะตรวจสอบ 5 เมตร

จากผลการทดลองเมื่อมีคนเดินผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "INTRUDER" และมีเสียงเตือนจากโปรแกรม โดยค่า min ที่แสดงในส่วนของ Shape Analyze ได้มาจากการหาค่าที่น้อยที่สุดของ Minimum Error template ทั้ง 9 template จากช่องด้านขวามือ ซึ่งในสถานการณ์นี้ตรงกับเทมเพลตที่ 2 โดยมีค่าของ Minimum Error ของ เทมเพลต เท่ากับ 165 และค่า index ที่แสดงเป็น 2 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าภาพที่เปรียบเทียบมีลักษณะตรงกับเทมเพลตที่ 2

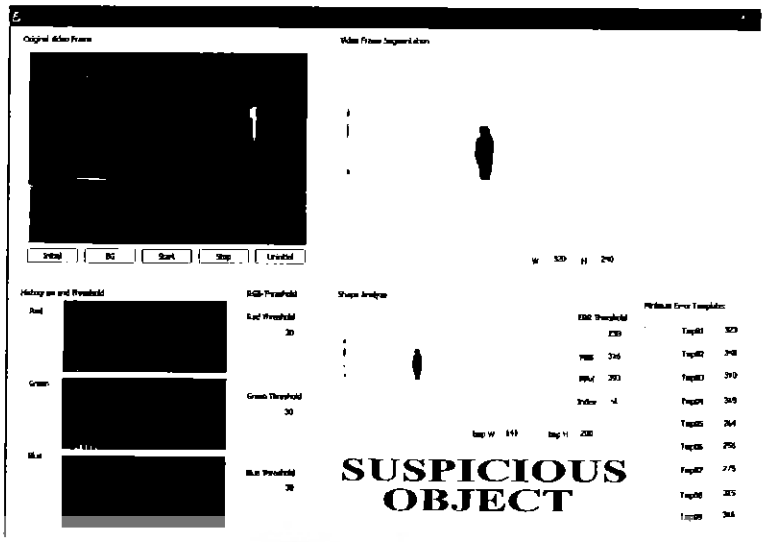
สถานการณ์ที่ 14 ใช้งานโปรแกรมในช่วงเวลากลางคืนมีคนเดินผ่านกล้อง และระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 7 เมตร



รูปที่ 4.25 แสดงหน้าต่างของโปรแกรมที่ไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้อง ในช่วงเวลากลางคืนระยะตรวจสอบ 7 เมตร

จากผลการทดลองเมื่อมีคนเดินผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "INTRUDER" และมีเสียงเตือนจากโปรแกรม โดยค่า min ที่แสดง ได้มาจากการหาค่าที่น้อยที่สุดของ Minimum Error template ทั้ง 9 template จากช่องด้านขวามือ ซึ่งในสถานการณ์นี้ตรงกับเทมเพลตที่ 4 โดยมีค่าของ Minimum Error ของเทมเพลต เท่ากับ 207 และค่า index ที่แสดงเป็น 4 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าภาพที่เปรียบเทียบมีลักษณะตรงกับเทมเพลตที่ 4

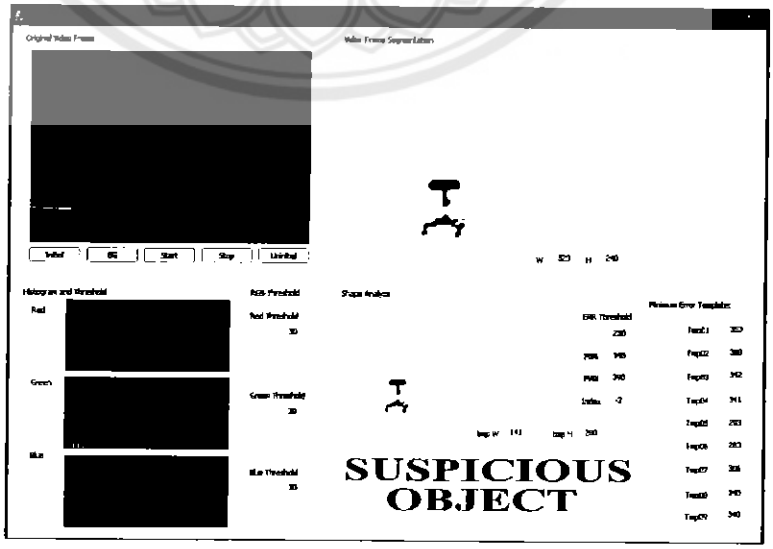
สถานการณ์ที่ 15 ใช้งานโปรแกรมในช่วงเวลากลางคืนมีคนเดินผ่านกล้อง และระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 10 เมตร



รูปที่ 4.26 แสดงหน้าต่างของโปรแกรมที่ไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้อง ในช่วงเวลากลางคืนระยะตรวจสอบ 10 เมตร

จากผลการทดลองเมื่อมีคนเดินผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "SUSPICIOUS OBJECT" และไม่มีเสียงเตือนจากโปรแกรม เนื่องจากโปรแกรมไม่สามารถค้นพบรูปร่างได้เพราะภาพที่ได้มีขนาดน้อยกว่าเทมเพลตที่ใช้ในการเปรียบเทียบ โดยค่า min ที่แสดงในส่วนของ Shape Analyze มีค่าเกินกว่าค่า Err Threshold ที่ได้ทำการตั้งค่าไว้ ซึ่งในสถานการณ์นี้ค่า min มีค่าเท่ากับ 316 และค่า index ที่แสดงเป็น -2 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าเป็นวัตถุ

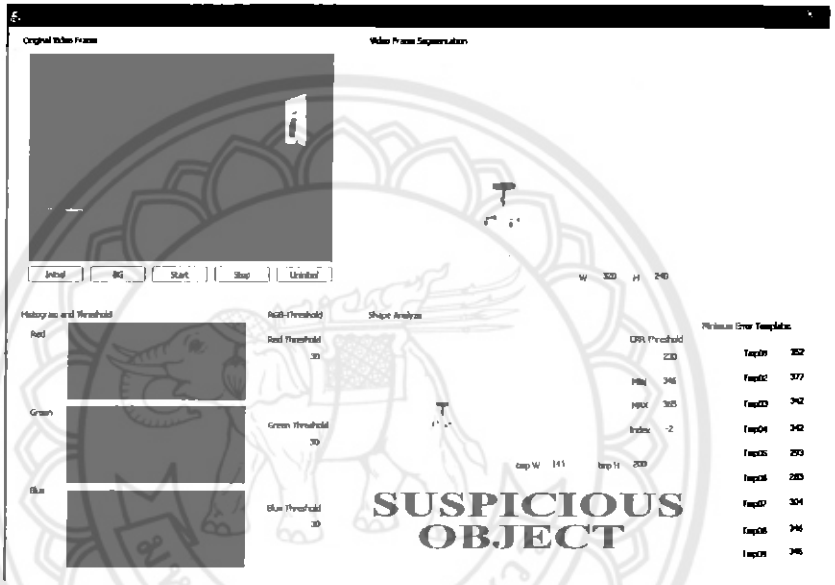
สถานการณ์ที่ 16 ใช้งานโปรแกรมในช่วงเวลากลางคืนมีวัตถุเคลื่อนที่ผ่านกล้อง และระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 5 เมตร



รูปที่ 4.27 แสดงหน้าต่างของโปรแกรมที่มีวัตถุเคลื่อนที่ผ่านกล้อง ในช่วงเวลากลางคืนระยะตรวจสอบ 5 เมตร

จากผลการทดลองเมื่อการเคลื่อนที่ของวัตถุผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "SUSPICIOUS OBJECT" และไม่มีเสียงเตือนจากโปรแกรม เนื่องจากโปรแกรมไม่สามารถค้นพบรูปร่างได้ โดยค่า min ที่แสดงในส่วนของ Shape Analyze มีค่าเกินกว่าค่า Err Threshold ที่ได้ทำการตั้งค่าไว้ ซึ่งในสถานการณ์นี้ค่า min มีค่าเท่ากับ 340 และค่า index ที่แสดงเป็น -2 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าเป็นวัตถุ

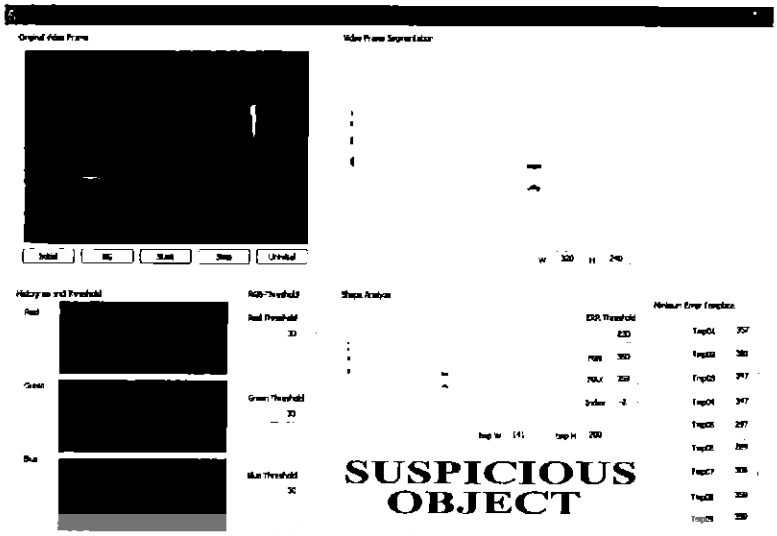
สถานการณ์ที่ 17 ใช้งานโปรแกรมในช่วงเวลากลางคืนมีวัตถุเคลื่อนที่ผ่านกล้อง และระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 7 เมตร



รูปที่ 4.28 แสดงหน้าต่างของโปรแกรมที่มีวัตถุเคลื่อนที่ผ่านกล้อง ในช่วงเวลากลางคืนระยะตรวจสอบ 7 เมตร

จากผลการทดลองเมื่อการเคลื่อนที่ของวัตถุผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "SUSPICIOUS OBJECT" และไม่มีเสียงเตือนจากโปรแกรม เนื่องจากโปรแกรมไม่สามารถค้นพบรูปร่างได้ โดยค่า min ที่แสดงในส่วนของ Shape Analyze มีค่าเกินกว่าค่า Err Threshold ที่ได้ทำการตั้งค่าไว้ ซึ่งในสถานการณ์นี้ค่า min มีค่าเท่ากับ 346 และค่า index ที่แสดงเป็น -2 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าเป็นวัตถุ

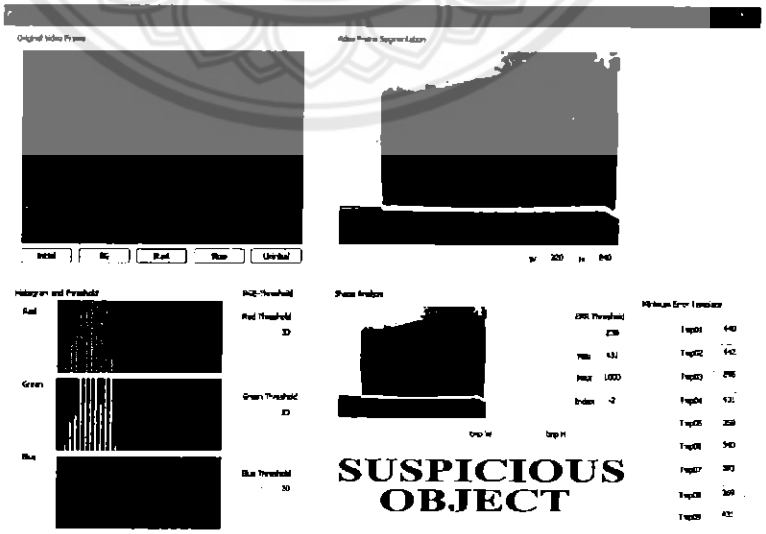
สถานการณ์ที่ 18 ใช้งานโปรแกรมในช่วงเวลากลางคืนมีวัตถุเคลื่อนที่ผ่านกล้อง และระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 10 เมตร



รูปที่ 4.29 แสดงหน้าต่างของโปรแกรมที่มีวัตถุเคลื่อนที่ผ่านกล้อง
ในช่วงเวลากลางคืนระยะตรวจสอบ 10 เมตร

จากผลการทดลองเมื่อการเคลื่อนที่ของวัตถุผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "SUSPICIOUS OBJECT" และไม่มีเสียงเตือนจากโปรแกรม เนื่องจากโปรแกรมไม่สามารถค้นพบรูปร่างได้ โดยค่า min ที่แสดงในส่วนของ Shape Analyze มีค่าเกินกว่าค่า Err Threshold ที่ได้ทำการตั้งค่าไว้ ซึ่งในสถานการณ์นี้ค่า min มีค่าเท่ากับ 350 และค่า index ที่แสดงเป็น -2 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าเป็นวัตถุ

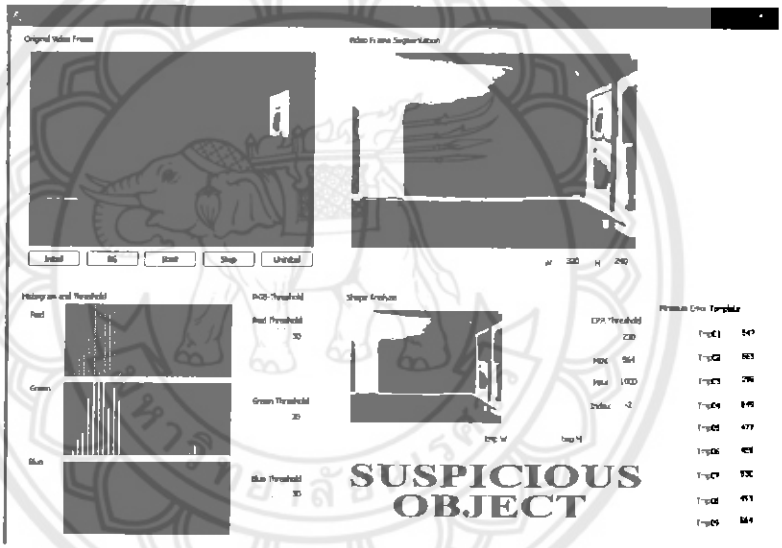
สถานการณ์ที่ 19 ใช้งานในช่วงเวลากลางคืนแต่ภาพต้นแบบเป็นช่วงเวลากลางวันและมีคนเดินผ่านกล้อง โดยระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 5 เมตร



ที่ 4.30 แสดงหน้าต่างของโปรแกรมที่มีคนเคลื่อนที่ผ่านกล้องในช่วงเวลากลางคืน
แต่ภาพต้นแบบเป็นช่วงเวลากลางวันและระยะตรวจสอบ 5 เมตร

จากผลการทดลองเมื่อมีคนเดินผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "SUSPICIOUS OBJECT" และไม่มีเสียงเตือนจากโปรแกรม เนื่องจากโปรแกรมทำการประมวลผลหาค่าผลต่างพื้นหลังแบบคงตัวของภาพที่เก็บไว้และภาพที่เปรียบเทียบจะได้ผลต่างมีค่าความคลาดเคลื่อน (error) มากในหลายจุดของภาพในการแยกวิเคราะห์วัตถุจึงได้ภาพพื้นหลังติดมากับวัตถุในบางส่วน ซึ่งแสดงให้เห็นในส่วนของ Video Frame Segmentation ทำให้โปรแกรมไม่สามารถค้นพบรูปร่างได้ โดยค่า min ที่แสดงในส่วนของ Shape Analyze มีค่าเกินกว่าค่า Err Threshold ที่ได้ทำการตั้งค่าไว้ ซึ่งในสถานการณ์นี้ค่า min มีค่าเท่ากับ 431 และค่า index ที่แสดงเป็น -2 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าเป็นวัตถุ

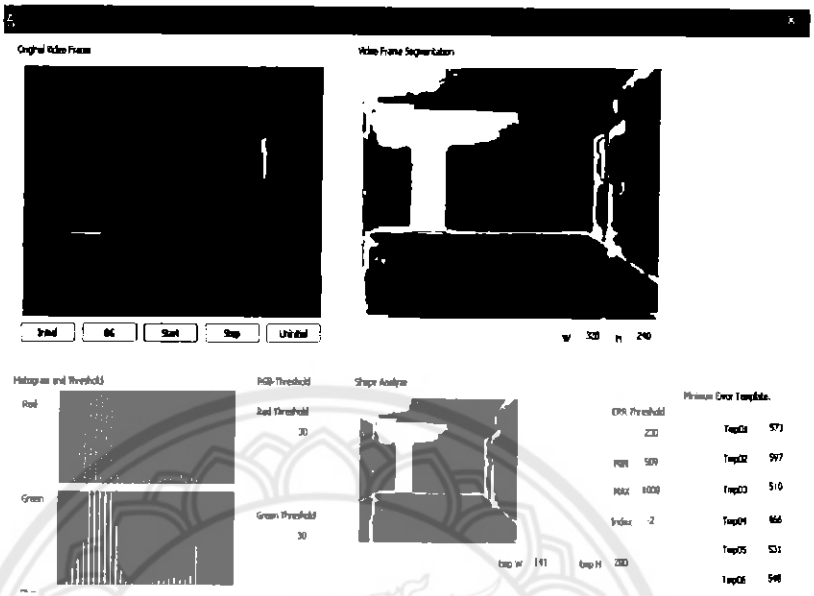
สถานการณ์ที่ 20 ใช้งานในช่วงเวลากลางคืนแต่ภาพต้นแบบเป็นช่วงเวลากลางวันและมีคนเดินผ่านกล้อง โดยระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 7 เมตร



ที่ 4.31 แสดงหน้าต่างของโปรแกรมที่มีคนเคลื่อนที่ผ่านกล้องในช่วงเวลากลางคืน แต่ภาพต้นแบบเป็นช่วงเวลากลางวันและระยะตรวจสอบ 7 เมตร

จากผลการทดลองเมื่อมีคนเดินผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "SUSPICIOUS OBJECT" และไม่มีเสียงเตือนจากโปรแกรม เนื่องจากโปรแกรมทำการประมวลผลหาค่าผลต่างพื้นหลังแบบคงตัวของภาพที่เก็บไว้และภาพที่เปรียบเทียบจะได้ผลต่างมีค่าความคลาดเคลื่อน (error) มากในหลายจุดของภาพในการแยกวิเคราะห์วัตถุจึงได้ภาพพื้นหลังติดมากับวัตถุในบางส่วน ซึ่งแสดงให้เห็นในส่วนของ Video Frame Segmentation ทำให้โปรแกรมไม่สามารถค้นพบรูปร่างได้ โดยค่า min ที่แสดงในส่วนของ Shape Analyze มีค่าเกินกว่าค่า Err Threshold ที่ได้ทำการตั้งค่าไว้ ซึ่งในสถานการณ์นี้ค่า min มีค่าเท่ากับ 564 และค่า index ที่แสดงเป็น -2 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าเป็นวัตถุ

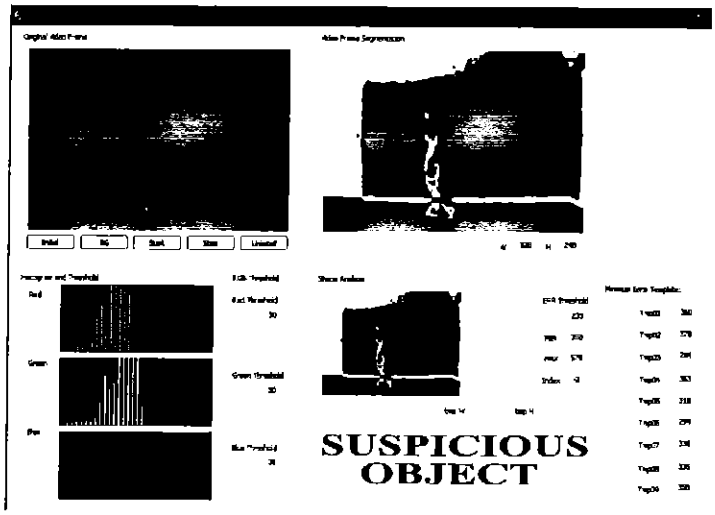
สถานการณ์ที่ 21 ใช้งานในช่วงเวลากลางคืนแต่ภาพต้นแบบเป็นช่วงเวลากลางวันและมีคนเดินผ่านกล้อง โดยระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 10 เมตร



ที่ 4.32 แสดงหน้าต่างของโปรแกรมที่มีคนเคลื่อนที่ผ่านกล้องในช่วงเวลากลางคืน แต่ภาพต้นแบบเป็นช่วงเวลากลางวันและระยะตรวจสอบ 10 เมตร

จากผลการทดลองเมื่อมีคนเดินผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "SUSPICIOUS OBJECT" และไม่มีเสียงเตือนจากโปรแกรม เนื่องจากโปรแกรมทำการประมวลผลหาค่าผลต่างพื้นหลังแบบคงตัวของภาพที่เก็บไว้และภาพที่เปรียบเทียบจะได้ผลต่างมีค่าความคลาดเคลื่อน (error) มากในหลายจุดของภาพในการแยกวิเคราะห์วัตถุจึงได้ภาพพื้นหลังติดมากับวัตถุในบางส่วน ซึ่งแสดงให้เห็นในส่วนของ Video Frame Segmentation ทำให้โปรแกรมไม่สามารถค้นพบรูปร่างได้ โดยค่า min ที่แสดงในส่วนของ Shape Analyze มีค่าเกินกว่าค่า Err Threshold ที่ได้ทำการตั้งค่าไว้ ซึ่งในสถานการณ์นี้ค่า min มีค่าเท่ากับ 509 และค่า index ที่แสดงเป็น -2 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าเป็นวัตถุ

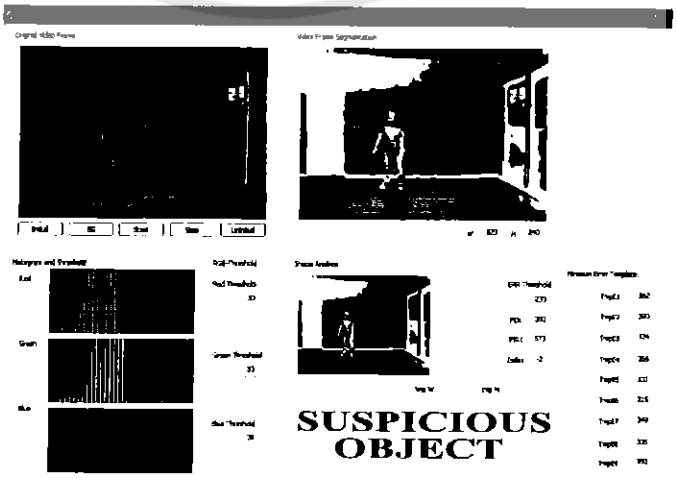
สถานการณ์ที่ 22 ใช้งานในช่วงเวลากลางวันแต่ภาพต้นแบบเป็นช่วงเวลากลางคืนและมีคนเดินผ่านกล้อง โดยระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบผู้บุกรุก 5 เมตร



รูปที่ 4.33 แสดงหน้าต่างของโปรแกรมที่มีคนเคลื่อนที่ผ่านกล้องในช่วงเวลากลางวัน แต่ภาพต้นแบบเป็นช่วงเวลากลางคืนและระยะตรวจสอบ 5 เมตร

จากผลการทดลองเมื่อมีคนเดินผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "SUSPICIOUS OBJECT" และไม่มีเสียงเตือนจากโปรแกรม เนื่องจากโปรแกรมทำการประมวลผลหาค่าผลต่างพื้นหลังแบบคงตัวของภาพที่เก็บไว้และภาพที่เปรียบเทียบจะได้ผลต่างมีค่าความคลาดเคลื่อน (error) มากในหลายจุดของภาพในการแยกวิเคราะห์วัตถุจึงได้ภาพพื้นหลังติดมากับวัตถุในบางส่วน ซึ่งแสดงให้เห็นในส่วนของ Video Frame Segmentation ทำให้โปรแกรมไม่สามารถค้นพบรูปร่างได้ โดยค่า min ที่แสดงในส่วนของ Shape Analyze มีค่าเกินกว่าค่า Err Threshold ที่ได้ทำการตั้งค่าไว้ ซึ่งในสถานการณ์นี้ค่า min มีค่าเท่ากับ 358 และค่า index ที่แสดงเป็น -2 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าเป็นวัตถุ

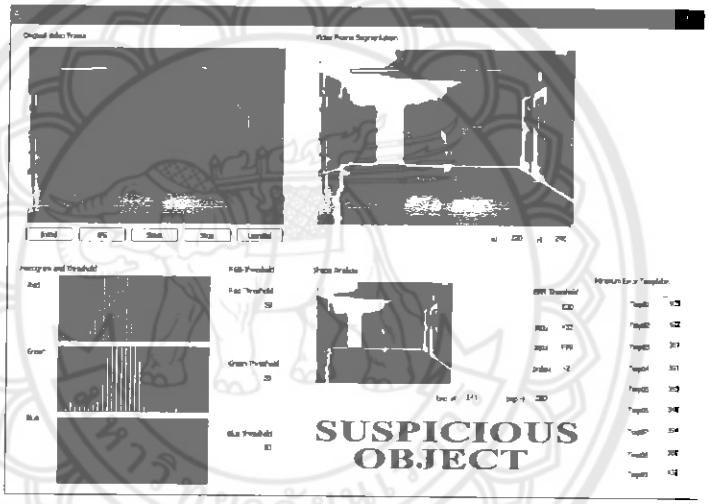
สถานการณ์ที่ 23 ใช้งานในช่วงเวลากลางวันแต่ภาพต้นแบบเป็นช่วงเวลากลางคืนและมีคนเดินผ่านกล้อง โดยระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 7 เมตร



ที่ 4.34 แสดงหน้าต่างของโปรแกรมที่มีคนเคลื่อนที่ผ่านกล้องในช่วงเวลากลางวัน แต่ภาพต้นแบบเป็นช่วงเวลากลางคืนและระยะตรวจสอบ 7 เมตร

จากผลการทดลองเมื่อมีคนเดินผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "SUSPICIOUS OBJECT" และไม่มีเสียงเตือนจากโปรแกรม เนื่องจากโปรแกรมทำการประมวลผลหาค่าผลต่างพื้นหลังแบบคงตัวของภาพที่เก็บไว้และภาพที่เปรียบเทียบจะได้ผลต่างมีค่าความคลาดเคลื่อน (error) มากในหลายจุดของภาพในการแยกวิเคราะห์วัตถุจึงได้ภาพพื้นหลังติดมากับวัตถุในบางส่วน ซึ่งแสดงให้เห็นในส่วนของ Video Frame Segmentation ทำให้โปรแกรมไม่สามารถค้นพบรูปร่างได้ โดยค่า min ที่แสดงในส่วนของ Shape Analyze มีค่าเกินกว่าค่า Err Threshold ที่ได้ทำการตั้งค่าไว้ ซึ่งในสถานการณ์นี้ค่า min มีค่าเท่ากับ 383 และค่า index ที่แสดงเป็น -2 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าเป็นวัตถุ

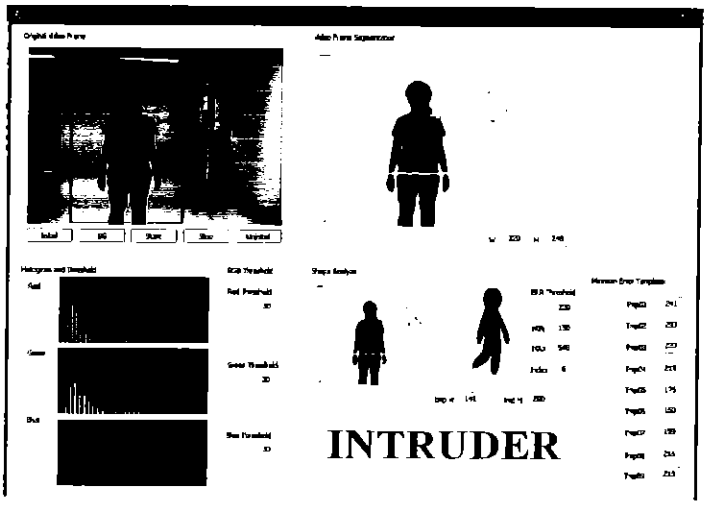
สถานการณ์ที่ 24 ใช้งานในช่วงเวลากลางวันแต่ภาพต้นแบบเป็นช่วงเวลากลางคืนและมีคนเดินผ่านกล้อง โดยระยะห่างของกล้องกับส่วนที่ทำการตรวจสอบ 10 เมตร



รูปที่ 4.35 แสดงหน้าต่างของโปรแกรมที่มีคนเคลื่อนที่ผ่านกล้องในช่วงเวลากลางวัน แต่ภาพต้นแบบเป็นช่วงเวลากลางคืนและระยะตรวจสอบ 10 เมตร

จากผลการทดลองเมื่อมีคนเดินผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "SUSPICIOUS OBJECT" และไม่มีเสียงเตือนจากโปรแกรม เนื่องจากโปรแกรมทำการประมวลผลหาค่าผลต่างพื้นหลังแบบคงตัวของภาพที่เก็บไว้และภาพที่เปรียบเทียบจะได้ผลต่างมีค่าความคลาดเคลื่อน (error) มากในหลายจุดของภาพในการแยกวิเคราะห์วัตถุจึงได้ภาพพื้นหลังติดมากับวัตถุในบางส่วน ซึ่งแสดงให้เห็นในส่วนของ Video Frame Segmentation ทำให้โปรแกรมไม่สามารถค้นพบรูปร่างได้ โดยค่า min ที่แสดงในส่วนของ Shape Analyze มีค่าเกินกว่าค่า Err Threshold ที่ได้ทำการตั้งค่าไว้ ซึ่งในสถานการณ์นี้ค่า min มีค่าเท่ากับ 432 และค่า index ที่แสดงเป็น -2 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าเป็นวัตถุ

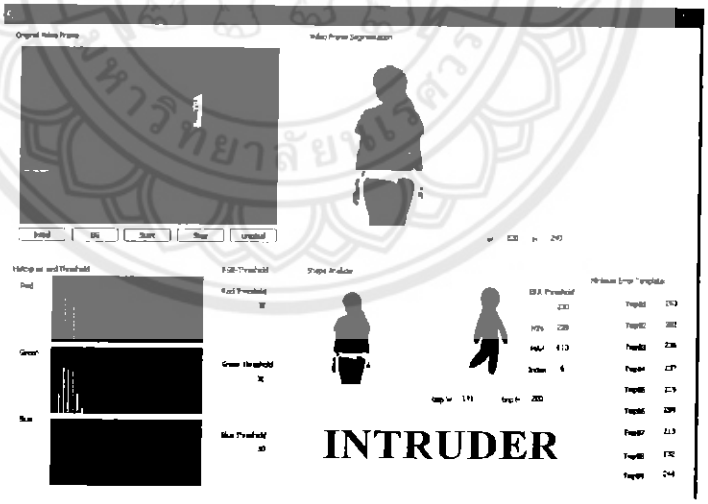
สถานการณ์ที่ 25 ใช้งานในช่วงเวลากลางวันโปรแกรมตรวจสอบรูปร่างแบบไม่เต็มตัว



รูปที่ 4.36 แสดงหน้าต่างของโปรแกรมที่มีคนเดินผ่านกล้องแบบไม่เต็มตัวในช่วงเวลากลางวัน

จากผลการทดลองเมื่อมีคนเดินผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "INTRUDER" และมีเสียงเตือนจากโปรแกรม โดยค่า min ที่แสดง ได้มาจากการหาค่าที่น้อยที่สุดของ Minimum Error template ทั้ง 9 template จากช่องด้านขวามือ ซึ่งในสถานการณ์นี้ตรงกับเทมเพลตที่ 6 โดยมีค่าของ Minimum Error ของเทมเพลต เท่ากับ 150 และค่า index ที่แสดงเป็น 6 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าภาพที่เปรียบเทียบมีลักษณะตรงกับเทมเพลตที่ 6

สถานการณ์ที่ 26 ใช้งานในช่วงเวลากลางคืนโปรแกรมตรวจสอบรูปร่างแบบไม่เต็มตัว



รูปที่ 4.37 แสดงหน้าต่างของโปรแกรมที่มีคนเดินผ่านกล้องแบบไม่เต็มตัวในช่วงเวลากลางคืน

จากผลการทดลองเมื่อมีคนเดินผ่านกล้อง โปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "INTRUDER" และมีเสียงเตือนจากโปรแกรม โดยค่า min ที่แสดง ได้มาจากการหาค่าที่น้อยที่สุดของ Minimum Error template ทั้ง 9 template จากช่องด้านขวามือ ซึ่งในสถานการณ์นี้ตรงกับเทมเพลตที่ 6 โดยมีค่าของ Minimum Error ของเทมเพลต เท่ากับ 209และค่า index ที่แสดงเป็น 6 เนื่องจากโปรแกรมทำการประมวลผลแล้วพบว่าภาพที่เปรียบเทียบมีลักษณะตรงกับเทมเพลตที่ 6

จากการทดลองในหลายๆ สถานการณ์ที่เกิดขึ้น สามารถแบ่งสถานการณ์ที่แตกต่างกันได้เป็น 26 สถานการณ์ ดังนี้

ตารางที่ 4.5 แสดงผลการทดลองโปรแกรมในสถานการณ์ต่างๆ

สถานการณ์	ระยะทาง (เมตร)	เหตุการณ์ที่เกิดขึ้น	การแจ้งเตือน	ความถูกต้อง
1	5	ช่วงเวลากลางวันไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้อง	ไม่แจ้ง	/
2	7	ช่วงเวลากลางวันไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้อง	ไม่แจ้ง	/
3	10	ช่วงเวลากลางวันไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้อง	ไม่แจ้ง	/
4	5	ช่วงเวลากลางวันมีคนเดินผ่านกล้อง	แจ้ง	/
5	7	ช่วงเวลากลางวันมีคนเดินผ่านกล้อง	ไม่แจ้ง	X
6	10	ช่วงเวลากลางวันมีคนเดินผ่านกล้อง	แจ้ง	/
7	5	ช่วงเวลากลางวันมีวัตถุเคลื่อนที่ผ่านกล้อง	ไม่แจ้ง	/
8	7	ช่วงเวลากลางวันมีวัตถุเคลื่อนที่ผ่านกล้อง	ไม่แจ้ง	/
9	10	ช่วงเวลากลางวันมีวัตถุเคลื่อนที่ผ่านกล้อง	ไม่แจ้ง	/
10	5	ช่วงเวลากลางคืนไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้อง	ไม่แจ้ง	/
11	7	ช่วงเวลากลางคืนไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้อง	ไม่แจ้ง	/
12	10	ช่วงเวลากลางคืนไม่มีคนหรือวัตถุเคลื่อนที่ผ่านกล้อง	ไม่แจ้ง	/
13	5	ช่วงเวลากลางคืนมีคนเดินผ่านกล้อง	แจ้ง	/
14	7	ช่วงเวลากลางคืนมีคนเดินผ่านกล้อง	แจ้ง	/
15	10	ช่วงเวลากลางคืนมีคนเดินผ่านกล้อง	ไม่แจ้ง	X
16	5	ช่วงเวลากลางคืนมีวัตถุเคลื่อนที่ผ่านกล้อง	ไม่แจ้ง	/
17	7	ช่วงเวลากลางคืนมีวัตถุเคลื่อนที่ผ่านกล้อง	ไม่แจ้ง	/
18	10	ช่วงเวลากลางคืนมีวัตถุเคลื่อนที่ผ่านกล้อง	ไม่แจ้ง	/

ตารางที่ 4.5 แสดงผลการทดลองโปรแกรมในสถานการณ์ต่างๆ (ต่อ)

สถานการณ์	ระยะทาง (เมตร)	เหตุการณ์ที่เกิดขึ้น	การแจ้งเตือน	ความถูกต้อง
19	5	ช่วงเวลากลางคืนแต่ภาพต้นแบบเป็นช่วงเวลาที่กลางวันมีคนเดินผ่านกล้อง	ไม่แจ้ง	X
20	7	ช่วงเวลากลางคืนแต่ภาพต้นแบบเป็นช่วงเวลาที่กลางวันมีคนเดินผ่านกล้อง	ไม่แจ้ง	X
21	10	ช่วงเวลากลางคืนแต่ภาพต้นแบบเป็นช่วงเวลาที่กลางวันมีคนเดินผ่านกล้อง	ไม่แจ้ง	X
22	5	ช่วงเวลากลางวันแต่ภาพต้นแบบเป็นช่วงเวลากลางคืนมีคนเดินผ่านกล้อง	ไม่แจ้ง	X
23	7	ช่วงเวลากลางวันแต่ภาพต้นแบบเป็นช่วงเวลากลางคืนมีคนเดินผ่านกล้อง	ไม่แจ้ง	X
24	10	ช่วงเวลากลางวันแต่ภาพต้นแบบเป็นช่วงเวลากลางคืนมีคนเดินผ่านกล้อง	ไม่แจ้ง	X
25	-	ช่วงเวลากลางวันและโปรแกรมตรวจสอบรูปร่างไม่เต็มตัว	แจ้ง	/
26	-	ช่วงเวลากลางคืนและโปรแกรมตรวจสอบรูปร่างไม่เต็มตัว	แจ้ง	/

4.5 วิเคราะห์ผลการทดลอง

จากผลการทดลองสามารถวิเคราะห์ได้ว่า ถ้ามีการจัดเก็บภาพเริ่มต้นในช่วงเวลาเดียวกับที่มีการใช้งานโปรแกรมในสถานการณ์ปกติโปรแกรมจะอยู่ในสถานะ "NORMAL" เมื่อมีคนเดินผ่านกล้องโปรแกรมจะทำการเปลี่ยนสถานะจาก "NORMAL" เป็น "INTURDER" และทำการแจ้งเตือนด้วยเสียง แต่ถ้าเป็นวัตถุที่มีการเคลื่อนที่ผ่านกล้องโปรแกรมจะทำการเปลี่ยนสถานะจาก "NORMAL" เป็น "SUSPICIOUS OBJECT" และจะไม่มีเสียงเตือนจากโปรแกรม และในบางกรณีเมื่อเพิ่มระยะห่างระหว่างกล้องเมื่อมีคนเดินผ่านกล้องจะพบว่าโปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "SUSPICIOUS OBJECT" และจะไม่มีเสียงเตือนจากโปรแกรม แต่เมื่อคนเดินเข้าใกล้กล้องมากขึ้นโปรแกรมจะทำการเปลี่ยนสถานะเป็น "INTURDER" และทำการแจ้งเตือนด้วยเสียง โปรแกรมจะทำการหยุดแจ้งเตือนด้วยเสียงจนกว่าผู้บุกรุกจะออกจากพื้นที่ที่ได้ติดตั้งกล้องไว้และจะกลับมาทำการตรวจสอบอีกครั้ง แต่ถ้ามีการจัดเก็บภาพเริ่มต้นในช่วงเวลาหนึ่งแต่ใช้

งานอีกช่วงเวลาหนึ่งโปรแกรมจะทำงานผิดพลาดเมื่อมีคนเดินผ่านกล้อง โดยโปรแกรมจะเปลี่ยนสถานะจาก "NORMAL" เป็น "SUSPICIOUS OBJECT" เนื่องจากแสงของภาพต้นแบบและภาพที่เปรียบเทียบไม่เท่ากันทำให้ภาพพื้นหลังติดมาด้วย โปรแกรมจึงไม่สามารถค้นพบรูปร่างของคนที่ได้ทำการจัดเก็บไว้ได้ จึงไม่มีการแจ้งเตือนของโปรแกรม ผลการทดลองทำให้ทราบว่าโปรแกรมสามารถบอกสถานะของสถานการณ์ในขณะนั้นได้อย่างถูกต้องและสามารถทำการแจ้งเตือน เมื่อมีผู้บุกรุกเมื่อโปรแกรมทำงานในช่วงเวลาที่แสงไม่สว่างมากจนเกินไปและระยะทางระหว่างส่วนที่ตั้งกล้องกับส่วนที่ตรวจสอบผู้บุกรุกมีระยะห่างไม่มากนัก

บทที่ 5
บทสรุป

โครงงานนี้ได้พัฒนาโปรแกรมตรวจจับผู้บุกรุกและแจ้งเตือนเมื่อพบผู้บุกรุก ด้วยภาษา C++ พัฒนาบน Microsoft Visual Studio.NET 2005 และเรียกใช้ฟังก์ชั่นการทำงานของการประมวลผลภาพจาก Library ที่ชื่อว่า OpenCV ซึ่งเป็น Library แบบ Free-Ware และนิยมใช้กันอย่างแพร่หลาย ซึ่งโครงงานนี้นำ OpenCV มาทำการรับภาพที่เชื่อมต่อจากกล้องวิดีโอเข้าสู่เครื่องคอมพิวเตอร์และใช้ตัวคัดกรองในลักษณะต่างๆ จาก Library นี้ รวมทั้งการใช้งานฟังก์ชั่น Template Matching ที่มีอยู่ใน OpenCV เพื่อทำการค้นหาลักษณะท่าทางของผู้บุกรุก เมื่อเปรียบเทียบกับเทมเพลตที่มีในระบบได้

เมื่อพัฒนาโปรแกรมตรวจจับผู้บุกรุกได้ถึงขั้นที่สามารถนำมาทดสอบ และได้แสดงผลการทดลองในบทที่ 4 ไปแล้วนั้น ในบทนี้จะทำการสรุปผลการทดลอง วิเคราะห์ผลการทดลอง ผลการทดสอบ และแสดงถึงปัญหาที่พบของโครงงานนี้ พร้อมทั้งเสนอแนวทางในการแก้ไขปัญหาในเบื้องต้น

5.1 สรุปการทดลอง

จากผลการทดลองในบทที่ 4 ซึ่งได้ทำการทดลองกับตัวแปรต่างๆคือ ระยะระหว่างกล้องกับวัตถุหรือคนที่เดินผ่านกล้อง คือ 5 เมตร 7 เมตร และ 10 เมตร ตัวแปรถัดไปคือ ประเภทของวัตถุหรือคน ซึ่งประกอบไปด้วย ไม่มีคนเดินผ่านกล้อง มีคนเดินผ่านกล้อง และมีวัตถุเคลื่อนที่ผ่านกล้อง และถัดมาคือตัวแปรที่เกี่ยวกับช่วงเวลาเพื่อใช้เป็นตัวชี้วัดความสว่างของสถานที่ที่ใช้ในการทดลอง และตัวแปรสุดท้ายคือช่วงเวลาที่ใช้ในการเก็บภาพพื้นหลัง ซึ่งทั้งหมดนั้นได้แสดงในตารางที่ 4.1 ในบทที่ 4 แล้ว

จากผลการทดลองที่ได้สรุปเป็นตารางที่ 4.1 สามารถสรุปได้ว่า โปรแกรมตรวจจับผู้บุรุกที่ได้พัฒนาขึ้นนี้สามารถทำการตรวจจับและแจ้งเตือนได้ โดยจากการทดลองทั้ง 26 สถานการณ์ พบว่า สามารถวิเคราะห์และทำการแจ้งเตือนเมื่อมีผู้บุกรุกได้ถูกต้อง 18 สถานการณ์ หรือคิดเป็นเปอร์เซ็นต์ของความถูกต้องได้เท่ากับ 69.23% ซึ่งอยู่ในเกณฑ์ที่ดี ซึ่งสิ่งที่ทำให้โปรแกรมเกิดความผิดพลาดไป 8 สถานการณ์ เนื่องจากการเก็บภาพพื้นหลังนั้นได้เก็บในช่วงเวลาที่แสงสว่างไม่เท่ากันกับตอนที่ทำการตรวจสอบ ทำให้เมื่อทำการแยกวัตถุออกจากรูปภาพจะไม่ได้วัตถุที่แท้จริง เนื่องจากจะมีพื้นหลังบางตำแหน่งติดมาด้วยเช่นกัน และระยะทางที่ใช้ในการตรวจสอบทำให้เกิดข้อผิดพลาดด้วยเช่นกัน

5.2 ปัญหาและแนวทางแก้ไข

จากการทดลองจะเห็นได้ว่า ปัญหาของการตรวจจับผู้บุกรุกด้วยโปรแกรมที่ได้พัฒนาขึ้น พบว่า ปริมาณความเข้มของแสงในสภาวะที่แตกต่างกันกับตอนที่ทำการเก็บภาพพื้นหลังนั้น มีผลให้การทำงานของโปรแกรมนี้วิเคราะห์และตรวจจับได้คลาดเคลื่อน เนื่องจากผลต่างของภาพที่เกิดขึ้นนั้นได้รับผลกระทบจากค่าแสงที่เปลี่ยนไป สำหรับการแก้ปัญหาของประเด็นนี้ อาจจะทำการเก็บภาพพื้นหลังใหม่เมื่อไม่มีผู้บุกรุกในทุกๆ 30 นาที หรือ 60 นาที เพื่อเป็นการลดปัญหาการอ้างอิงกับภาพพื้นหลังที่ห่างไกลกับเหตุการณ์ที่ตรวจสอบได้

ปัญหาที่ 2 คือ ระยะทางมีผลต่อการตรวจจับวัตถุ เนื่องจากกล้องวิดีโอมีความสามารถในการตรวจจับและมองเห็นในระยะที่กำหนด ในบางครั้งอาจจะไม่สามารถมองเห็นภาพที่อยู่ไกลในระยะ 10 เมตร ได้ดีนัก หรือในกรณีที่กล้องสามารถจับภาพได้แต่โปรแกรมที่พัฒนาขึ้นไม่สามารถระบุรูปร่างได้ ซึ่งอาจจะเนื่องมาจากภาพที่ทำการแยกออกจากพื้นหลังนั้นมีขนาดเล็กกว่าเมื่อนำไปเปรียบเทียบกับเทมเพลตที่เตรียมไว้ ทำให้ไม่สามารถค้นหารูปร่างได้ทั้งที่อาจจะมีรูปร่างคล้ายกันแต่ขนาดไม่เท่ากัน (Size's Problem) ซึ่งเป็นปัญหาปกติของวิธีเทมเพลตแมชชิ่ง ดังนั้น วิธีที่จะแก้ไขปัญหาในลักษณะเช่นนี้คือ การเพิ่มเทมเพลตที่มีขนาดของรูปร่างในหลายๆ ขนาดลงไปในโปรแกรมก็จะสามารถทำการค้นหาเทมเพลตที่ใกล้เคียงจนพบ แต่วิธีการนี้จะทำให้เพิ่มระยะเวลาในการคำนวณด้วยเช่นกัน ซึ่งจะทำให้โปรแกรมทำงานช้าลงตามปริมาณเทมเพลตที่เพิ่มเข้าไป

เอกสารอ้างอิง

[1] " Image Shape " [online]. Available :
http://www.irecog.com/km/imgprocessing/imageshape.php

[2] " มาตรฐานของสี (Standard of Color) " [online]. Available :
http://www.irecog.com/km/imgprocessing/color.php

[3] รองศาสตราจารย์สมัย เหล่าวานิชย์. คณิตศาสตร์ ม.5 เล่ม 3. กรุงเทพมหานคร : บริษัท ไฮเอ็ดพับบลิชชิ่ง จำกัด. 2535

[4] Baxes , Gregory A. Digital Image Processing. Canada : John Wiley & sons, Inc. 1994.

[5] " Image Segmentationt " [online]. Available :
http://fivedots. coe. psu. ac. th/~ montri/Teaching/image/segment. DOC2551

[6] " Template Matching " [online]. Available :
http://en.wikipedia.org/wiki/Template_matching

[7] " Template Matching " [online]. Available :
http://www.rtsd.mi.th/New%202008/12-51/image_processing/Image%20Processing6.pdf

ภาคผนวก ก

การใช้งาน OpenCV เบื้องต้น ✳

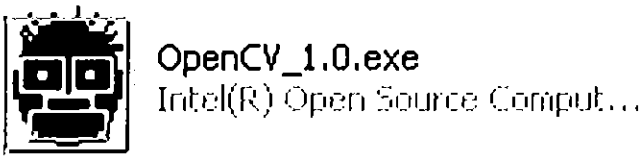
OpenCV เป็น Library สำหรับช่วยในการเขียนโปรแกรมประเภทการประมวลผลภาพ (Image Processing) และการประมวลการมองเห็น (Computer Vision) โดยพัฒนาด้วยภาษา C++ ซึ่ง OpenCV ได้ถูกพัฒนามาจาก Core ของ DirectX SDK ในส่วนของ DirectShow จนกระทั่งรุ่นปัจจุบันนั้นสามารถพัฒนาแยกออกมาจาก DirectX SDK อย่างเต็มตัว ทำให้ผู้ใช้งานไม่จำเป็นต้องติดตั้ง DirectX SDK ก่อน สามารถติดตั้ง OpenCV แล้วใช้งานได้ทันที ซึ่งสามารถพัฒนาโปรแกรมโดย IDE ของ Microsoft คือ Visual C++ (Microsoft Visual Studio 6.0 และ/หรือ Microsoft Visual Studio.NET) โดยภาษา C++

OpenCV เป็น Library ที่มีฟังก์ชั่นทางการประมวลผลภาพและโครงสร้างของข้อมูลทางการประมวลผลภาพเชิงดิจิตอล การวิเคราะห์ภาพถ่ายดิจิตอลที่รองรับรูปแบบมาตรฐาน เช่น นามสกุล .JPEG , .BMP และสามารถทำการติดต่อกับอุปกรณ์ตรวจจับภาพได้ เช่น กล้องวีดิโอ ซึ่งสามารถทำงานผ่านไดเวอร์ของกล้องนั้นๆ (USB, IEEE, AV) สามารถกำหนดและปรับแต่งค่าคุณสมบัติของกล้องนั้นผ่านไดเวอร์ได้เช่นกัน รองรับโมเดลสี (Color Model) ใน Spatial โดเมนหลายชนิด เช่น RGB , HSV , YUV เป็นต้น รวมทั้งมีฟังก์ชั่นในการแปลงสีในโมเดลต่างๆ นอกจากสามารถวิเคราะห์ใน Spatial โดเมนดังกล่าวแล้วยังสนับสนุน Frequency โดเมนด้วย โดยมีการแปลงและใช้งานฟังก์ชั่นในโดเมนความถี่ได้เช่นกัน มีฟังก์ชั่นประเภทตัวคัดกรอง (Filtering) ต่างๆ ให้เลือกใช้งานตามความเหมาะสม การถอดความหมายของรูปภาพ (Feature extraction) รวมถึงกระบวนการทางคณิตศาสตร์เชิงรูปภาพต่างๆ

โดยรายละเอียดปลีกย่อยนั้นจะถูกบรรจุอยู่ในเอกสารที่ถูกติดตั้งมาพร้อมกับการติดตั้ง OpenCV ในคอมพิวเตอร์โดยอยู่ในไดเรกทอรี่ "...\OpenCV\docs" หรือ start→All Program→ OpenCV→Documentation (html) โดยจะอธิบายฟังก์ชั่นต่างๆ พร้อมรูปแบบของอาร์กิวเมนท์และเอาต์พุตและแสดงตัวอย่างการใช้งานอย่างชัดเจน พร้อมทั้งมีตัวอย่าง Source Code ที่สามารถนำมาพัฒนาในการใช้งานในรูปแบบต่างๆ เช่น การนำภาพจากไฟล์เข้ามาประมวลผล การค้นหาขอบรูปภาพ การติดต่อกับกล้องวิดีโอ เป็นต้น

1.1 การติดตั้ง OpenCV

สามารถดาวน์โหลดไฟล์ติดตั้งได้จาก http://www.sourceforge.net/projects/opencvlibrary ซึ่งรุ่นที่มีความสมบูรณ์และแนะนำใช้ในปัจจุบันนี้ (2007) คือ OpenCV 1.0 โดยมีลักษณะดังรูป

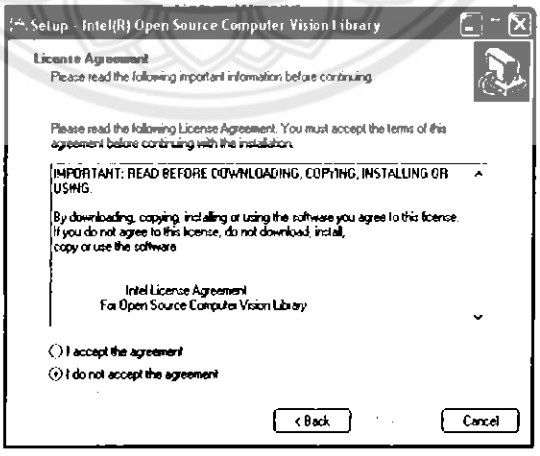


รูป ก.1 แสดงไฟล์ของโปรแกรมที่ใช้ในการติดตั้ง

เมื่อทำการติดตั้งโดย Double Click จะปรากฏหน้าต่างของการติดตั้งดังรูปข้างล่างนี้ จากนั้นกดที่ปุ่ม Next เพื่อทำการติดตั้งต่อไป

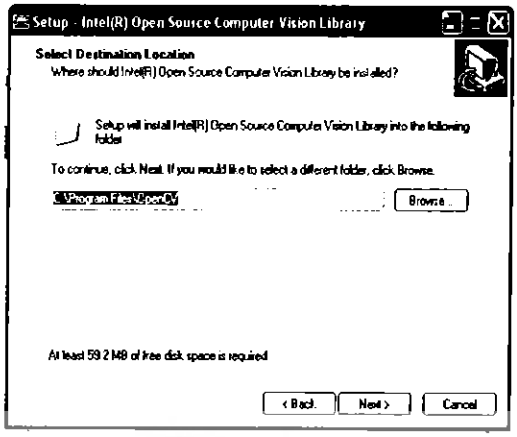
รูป ก.2 แสดงหน้าต่างการติดตั้งของโปรแกรม

จากนั้นจะเป็นการตกลงเรื่อง License เมื่ออ่านข้อความในกรอบข้างล่างจนครบแล้ว และยอมรับข้อตกลง จากนั้นเลือก "I accept the agreement" แล้วจึงกดปุ่ม Next ต่อ



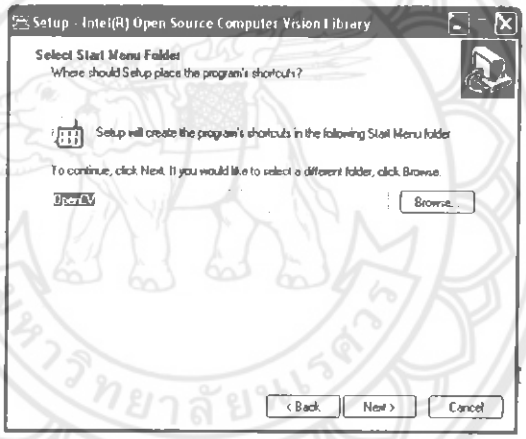
รูป ก.3 แสดงหน้าต่างการติดตั้งของโปรแกรมที่ให้ทำการตกลง

จากนั้นทำการเลือกไดเรกทอรี่ในการติดตั้ง และกดปุ่ม Next ต่อไป



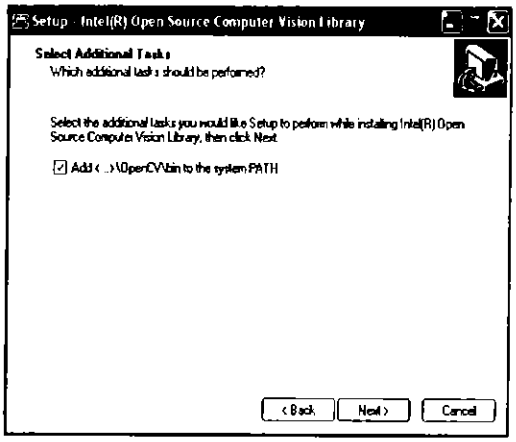
รูป ก.4 แสดงหน้าต่างให้เลือกไดร์ที่จะทำการติดตั้ง

จากนั้นทำการเลือกชื่อไดเรกทอรี่ใน Start Menu และกดปุ่ม Next ต่อไป



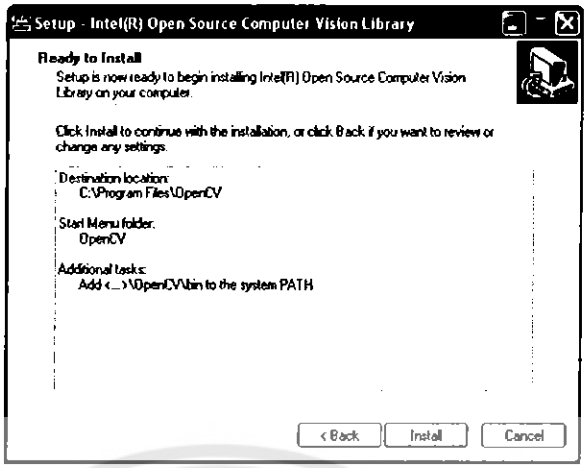
รูป ก.5 แสดงหน้าต่างให้เลือกโฟล์เดอร์

เลือก Add <...>\OpenCV\bin to the system Path เพื่อทำการสร้างพาทให้ตัวแปรกับระบบ



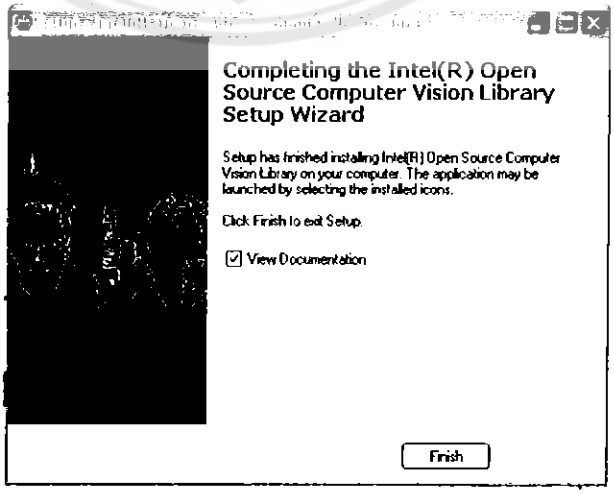
รูป ก.6 แสดงหน้าต่างให้ทำการเพิ่ม path

ในกรอบด้านล่างนั้นจะแสดงข้อมูลที่กำหนดในการติดตั้งดังกล่าว จากนั้นกดปุ่ม Install



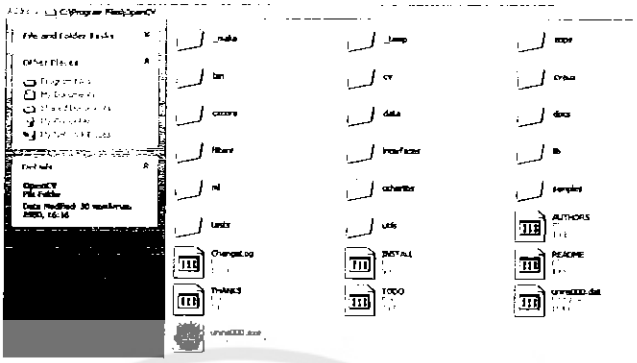
รูป ก.7 แสดงหน้าต่างในการเริ่มการติดตั้ง

รอการติดตั้งจนสำเร็จ



รูป ก.8 แสดงหน้าต่างการติดตั้งสำเร็จ

โปรแกรมจะถูกติดตั้งในไดเรกทอรี่ที่ได้กำหนดไว้ข้างต้น และจะมีไฟล์และไดเรทอรี่ดังนี้

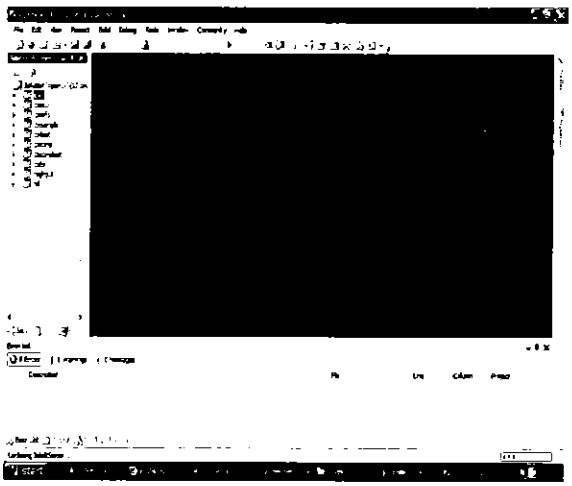


รูป ก.9 แสดงหน้าต่างไดเรกทอรี่ที่ได้กำหนด

ไปที่ Start Menu → All Program → OpenCV → OpenCV Workspace.NET 2005

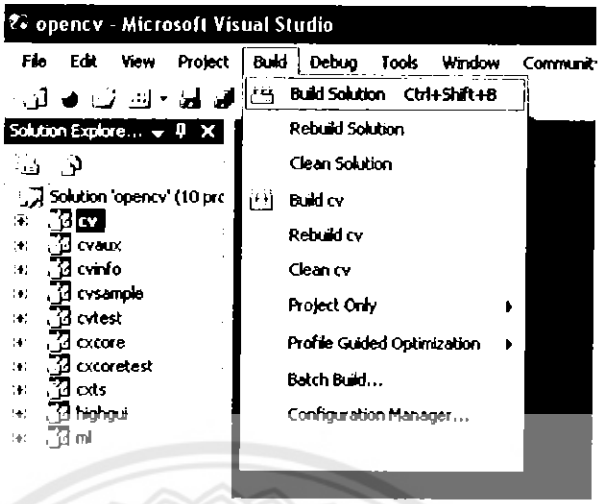
รูป ก.10 แสดงหน้าต่างในการเปิดโปรแกรม

จะเป็นการเรียก Workspace ของ OpenCV ขึ้นมา เพื่อที่จะทำการ Build เพื่อให้เกิด เอาพุตไฟล์ หรือสามารถ ไปที่ Microsoft Visual Studio .NET 2005 แล้วเลือก file→Open Project แล้ว Browse ไปหา opencv.sln ใน C:\Program Files\OpenCV_make แล้วเปิดมาก็ได้เช่นกัน



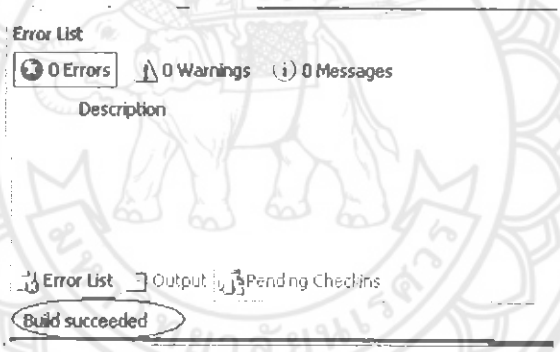
รูป ก.11 แสดงหน้าต่างไดเรกทอรี่ที่ได้กำหนด

ไปที่ Build → Build Solution เพื่อทำการ Build OpenCV



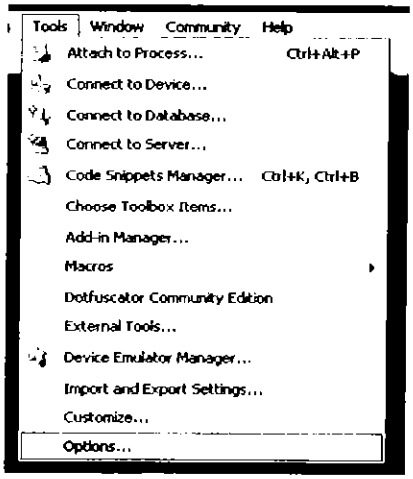
รูป ก.12 แสดงหน้าต่างการ Build OpenCV

รอจนกระทั่ง ที่ Status bar ด้านล่าง เป็น "Build Succeeded"



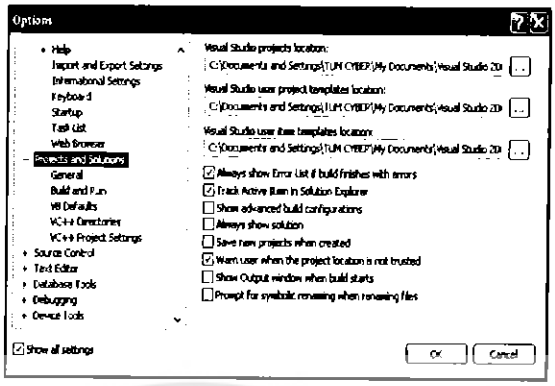
รูป ก.13 แสดงหน้าต่าง Status bar

จากนั้นทำการ Set Path ใช้กับ Visual Studio โดยไปที่ Tool→Option...



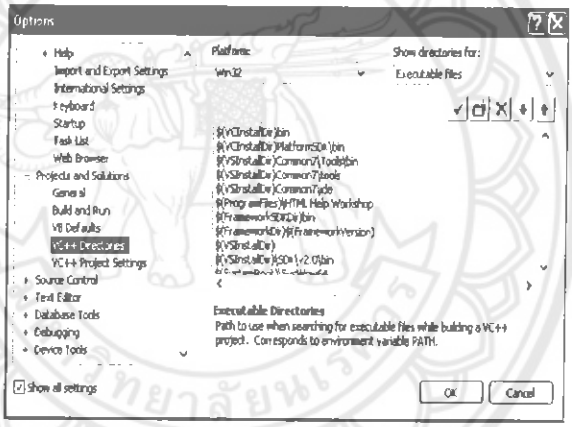
รูป ก.14 แสดงหน้าต่างในการตั้งค่าโปรแกรม

ไปที่แถบ Project and Solution คลิกเครื่องหมาย + ข้างหน้าแถบจะเกิดแถบรายการย่อย ดังรูปด้านล่าง



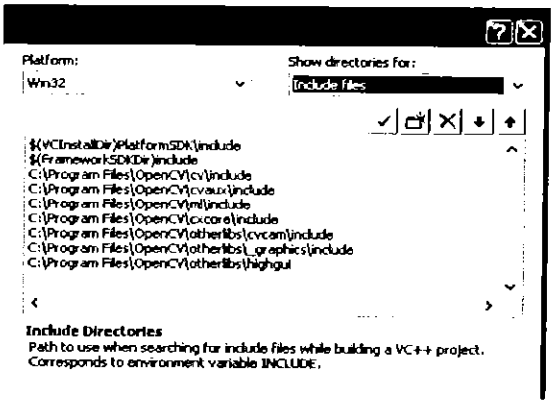
รูป ก.15 แสดงหน้าต่างในการตั้งค่าโปรแกรมโดยเลือก Project and Solution

เลือก VC++ Directories เพื่อที่จะทำการกำหนด Path ให้กับตัวแปรภายในต่างๆ



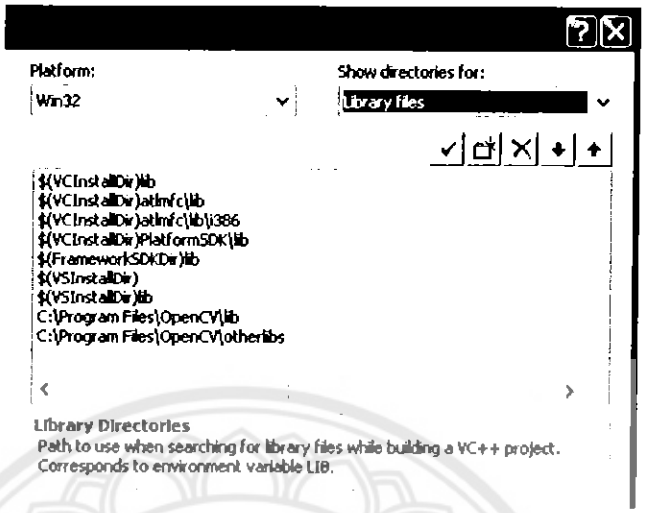
รูป ก.16 แสดงหน้าต่างในการตั้งค่า Path ของโปรแกรม

คลิกในแถบ Menu Show directories for: โดยเริ่มจาก Include files แล้วเพิ่ม path ดังรูปข้างล่างนี้ เพื่อให้การอ้างถึง ฟังก์ชั่นที่เรียกใช้ โดย .h ต่างๆ ได้



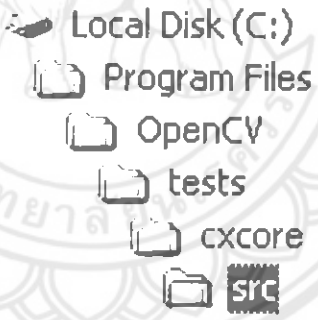
รูป ก.17 แสดงหน้าต่างในการตั้งค่า Include files

คลิกในแถบ Menu Show directories for: เลือก Library files แล้วเพิ่ม path ดังรูปข้างล่างนี้ เพื่อให้การอ้างถึงฟังก์ชั่นที่เรียกใช้ที่อยู่ใน Library files อีกที (.LIB)



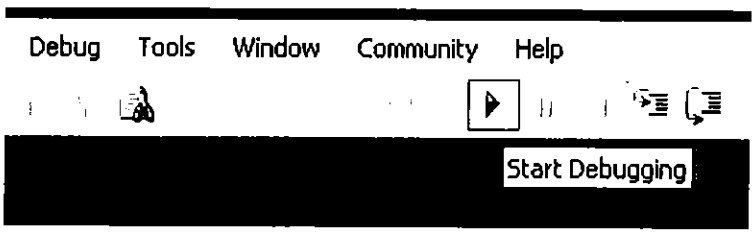
รูป ก.18 แสดงหน้าต่างในการตั้งค่า Library files

จากนั้น ลองทำการทดสอบฟังก์ชั่นพื้นฐานเพื่อตรวจสอบว่าการลง OpenCV นี้สามารถใช้งานได้หรือยังโดยไปที่ file→Open Project แล้วไปที่



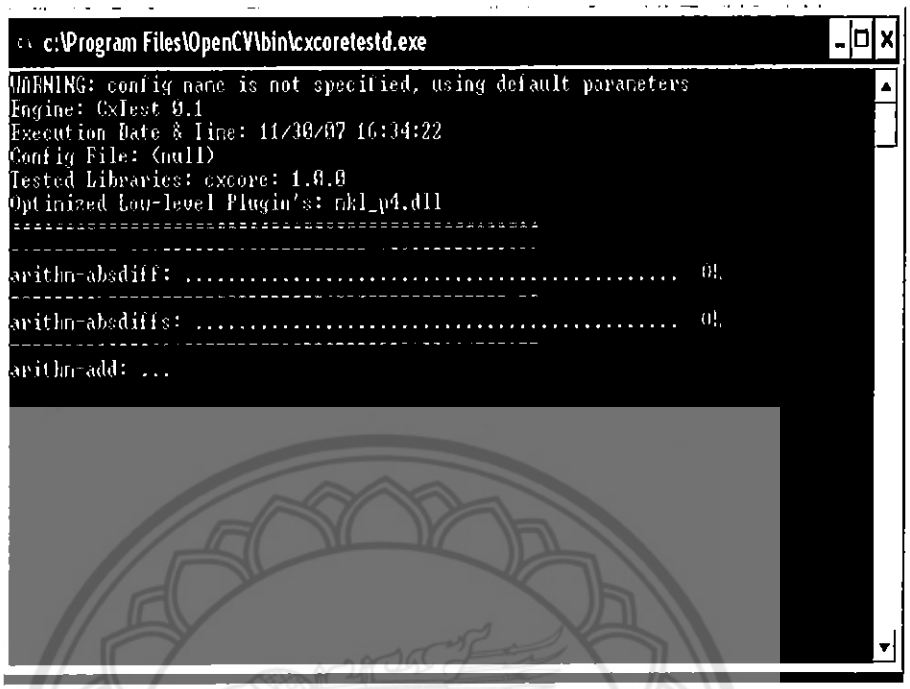
รูป ก.19 แสดงหน้าต่างในการทดสอบฟังก์ชัน

เลือก cxcoretest.vcproj เปิดขึ้นมา จากนั้นทำการ Start Debugging



รูป ก.20 แสดงหน้าต่างในการเริ่ม Debugging

ทดสอบการทำงานของฟังก์ชั่นต่างๆ



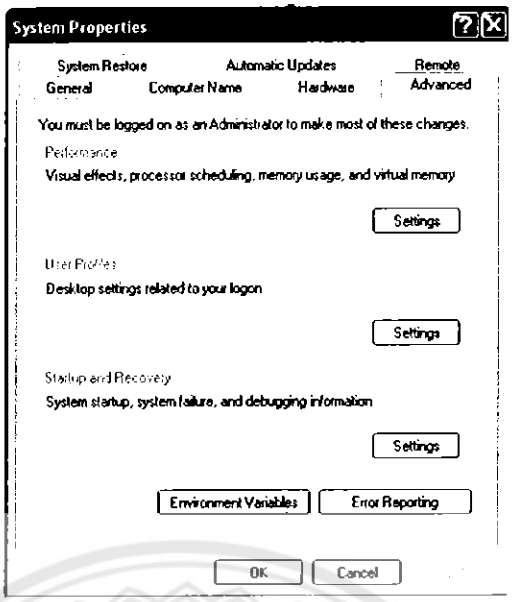
รูป ก.21 แสดงหน้าต่างทดสอบการทำงานของฟังก์ชัน

1.2 วิธีแก้ไขปัญหาที่อาจจะพบ

1. เมื่อรันหรือคอมไพล์แล้วมี error เกี่ยวกับ "cannot open XXX.h" แสดงถึงว่า มีปัญหาเกี่ยวกับ header file ของฟังก์ชั่น ให้ไปตรวจสอบที่การ Set Path ให้กับ Include files ที่ Tool→ Option...→ Project and Solution→ VC++ Directories → Include files แล้ว Set Path ดังที่กล่าวมาในข้างต้น

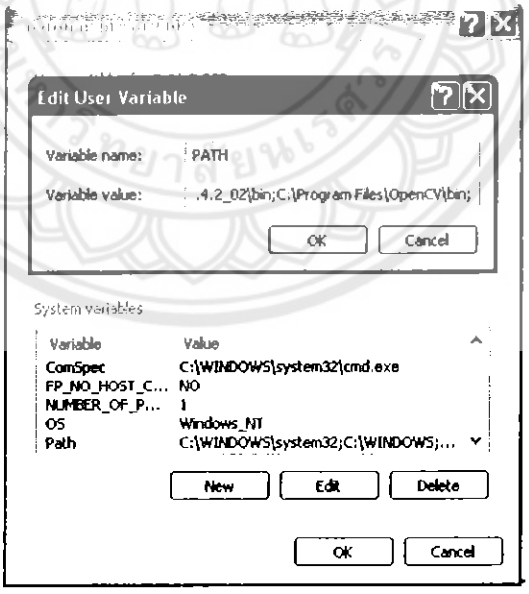
2. เมื่อรันหรือคอมไพล์แล้วมี error เกี่ยวกับ "cannot open XXX.LIB" แสดงถึงว่ามีปัญหาเกี่ยวกับ Library file ของฟังก์ชั่น ให้ไปตรวจสอบที่การ Set Path ให้กับ Include files ที่ Tool→ Option...→ Project and Solution→ VC++ Directories → Library files แล้ว Set Path ดังที่กล่าวมาในข้างต้น

3. เมื่อรันหรือคอมไพล์แล้ว มี error เกี่ยวกับ "XXX.DLL" แสดงถึงว่ามีปัญหาเกี่ยวกับ System file ให้ไปตรวจสอบที่ System Property ของคอมพิวเตอร์โดยคลิกขวาที่ My computer จากนั้นไปที่ แท็ป Advanced



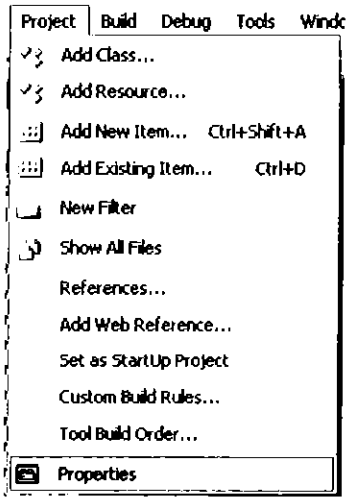
รูป ก.22 แสดงหน้าต่างการตรวจสอบ System Property

ไปที่ Environment Variables แล้วไปที่ User Variables หาตัวแปรชื่อว่า Path ถ้ามี ทำการ Edit โดยเพิ่มเครื่องหมาย ; เป็นการคั่น Path ที่มีอยู่ แล้ว ใส่ค่า C:\Program Files\OpenCV\bin; ลงไป ถ้าหากไม่มีตัวแปร Path ให้ New ขึ้นมาใหม่ แล้ว ใส่ค่า C:\Program Files\OpenCV\bin; ลงไปเช่นกัน



รูป ก.23 แสดงหน้าต่างการเพิ่ม Path

4. เมื่อรันหรือคอมไพล์ แล้ว มี error เกี่ยวกับ "..... LINKED Error" แสดงถึงว่า มีปัญหาเกี่ยวกับ Linker ให้ไปตรวจสอบที่ Project Property



รูป ก.24 แสดงหน้าต่างการตั้งค่า Project Property

ไปที่ แถบ Configuration → Linker→Command Line เพิ่มในส่วนที่ Link นั้นๆ มีปัญหาโดยทั่วไปก็เพิ่มดังนี้ cv.lib highgui.lib cvaux.lib cxcore.lib

ภาคผนวก ข

ฟังก์ชันของ OpenCV ที่ใช้ในโปรเจค

1. cvSmooth

```
void cvSmooth( const CvArr* src, CvArr* dst,
        int smoothtype=CV_GAUSSIAN,
        int param1=3, int param2=0, double param3=0, double param4=0 );
```

เป็นฟังก์ชันที่ใช้ในการทำให้รูปภาพเกิดความชัดเจนขึ้น โดยมีวิธี**ทำให้ภาพเกิดความชัดเจน**ขึ้นได้ 5 วิธี คือ

- CV_BLUR_NO_SCALE (simple blur with no scaling)
- CV_BLUR (simple blur)
- CV_GAUSSIAN (gaussian blur
- CV_MEDIAN (median blur)
- CV_BILATERAL (bilateral filter)

โดยในโครงงานนี้ได้ใช้ 3 วิธี คือ CV_BLUR , CV_GAUSSIAN และ CV_MEDIAN ซึ่งจะประกอบด้วยพารามิเตอร์ ดังนี้

- param1 : เป็นพารามิเตอร์แรกที่ใช้การทำภาพให้ชัดเจน
- param2 : เป็นพารามิเตอร์ที่สองที่ใช้การทำภาพให้ชัดเจน
- param3 : ในกรณีที่ใช้ Gaussian kernel พารามิเตอร์นี้จะต้องระบุ Gaussian sigma (standard deviation) ถ้ามีค่าเท่ากับ 0, จะถูกคำนวณได้จาก:

  sigma = (n/2 - 1)*0.3 + 0.8 เมื่อ n = param1 สำหรับ horizontal kernel

  n = param2 สำหรับ vertical kernel
- param4 : ในกรณีที่ใช้ non-square Gaussian kernel พารามิเตอร์นี้จะต้องระบุผลต่าง (จาก param3) sigma ในแกนตั้ง

2. CvtColor

```
void cvCvtColor( const CvArr* src, CvArr* dst, int code );
```

เป็นฟังก์ชั่นที่ใช้ในการเปลี่ยนแปลงสีของรูปภาพจากสีปกติของรูปภาพเป็นสีที่ต้องการเปลี่ยนแปลงโดย

- src : ภาพที่ต้องการทำการเปลี่ยนแปลงชนิดสี 8-bit (8u), 16-bit (16u) หรือเป็น single-precision floating-point (32f) image.
- dst : ภาพผลลัพธ์ที่ถูกเปลี่ยนแปลงโดยจะต้องมีขนาด และchannel ที่เท่ากันกับ src
- code : ชนิดของระบบสีที่ต้องการเปลี่ยน โดยหาได้จากค่าคงที่ CV_<src_color_space>2<dst_color_space>

3. cvMatchTemplate

```
void cvMatchTemplate( const CvArr* image, const CvArr* templ,
                      CvArr* result, int method );
```

เป็นฟังก์ชั่นที่ใช้ในการเปรียบเทียบกับรูปภาพก่อนหน้ากับรูปภาพที่ได้รับเข้ามา โดยมีพารามิเตอร์ดังนี้

- image : เป็นรูปภาพที่ใช้ในการหารูปแบบ
- templ : ใช้ในการหารูปแบบ; must be not greater than the source image and the same data type as the image.
- result : ผลลัพธ์ที่ได้จากการเปรียบเทียบ
- method : วิธีในการหาภาพต้นแบบและวิธีการเปรียบเทียบภาพ

4. cvLoadImage

```
IplImage* cvLoadImage( const char* filename, int
                       flags=CV_LOAD_IMAGE_COLOR);
```

เป็นฟังก์ชั่นที่ใช้ในการโหลดรูปภาพจากไฟล์เข้ามาในโปรแกรม ซึ่งประกอบด้วย

- filename : เป็นชื่อไฟล์ที่ต้องการโหลด
- flags : เป็นตัวกำหนดสีและลักษณะของรูปภาพที่จะทำการโหลด

5. cvCreateCameraCapture

```
CvCapture* cvCreateCameraCapture( int index );
```

เป็นฟังก์ชั่นที่ใช้ในการเริ่มต้นการจัดเก็บภาพเคลื่อนไหวจากกล้อง โดยพารามิเตอร์ index เป็นพารามิเตอร์ที่บอกสถานะเริ่มต้นของการใช้กล้อง

7. cvReleaseCapture

```
void cvReleaseCapture( CvCapture** capture );
```

เป็นฟังก์ชั่นที่เป็นโครงสร้างของฟังก์ชัน CvCapture โดยพารามิเตอร์ capture เป็นพารามิเตอร์ที่จะระบุว่าต้องการส่วนไหนของ VDO

8. cvGrabFrame

```
int cvGrabFrame( CvCapture* capture );
```

เป็นฟังก์ชั่นที่ใช้ในการระบุภาพจาก VDO โดยพารามิเตอร์ capture เป็นพารามิเตอร์ที่นำภาพนิ่งที่ได้จาก VDO เข้ามาใช้ในการประมวลผล

9. cvRetrieveFrame

```
IplImage* cvRetrieveFrame( CvCapture* capture );
```

เป็นฟังก์ชันที่นำรูปภาพจากการระบุภาพที่ได้จากฟังก์ชัน cvGrabFrame โดยพารามิเตอร์ capture เป็นพารามิเตอร์ที่นำภาพนิ่งที่ได้จาก VDO เข้ามาใช้ในการประมวลผล

10. cvFlip

```
void  cvFlip( const CvArr* src, CvArr* dst=NULL, int flip_mode=0);
```

เป็นฟังก์ชั่นที่ใช้ในการกลับภาพหรือพลิกภาพจากบนลงล่างและจากซ้ายไปขวา โดยมีพารามิเตอร์ดังนี้

- Src : เป็นอะเรย์เริ่มต้น
- dst : เป็นอะเรย์สุดท้าย
- flip_mode : ตัวกำหนดวิธีการคำนวณของอะเรย์

11. cvCloneImage

```
IplImage* cvCloneImage( const IplImage* image );
```

เป็นฟังก์ชั่นที่ใช้ในการคัดลอกรูปภาพ โดยพารามิเตอร์ image เป็นรูปภาพดั้งเดิม

12. cvCreateImage

```
IplImage* cvCreateImage( CvSize size, int depth, int channels );
```

เป็นฟังก์ชันที่ใช้ในการสร้างส่วนหัวและกำหนดรายละเอียดของข้อมูล โดยมีพารามิเตอร์ดังนี้

- size : เป็นตัวกำหนดความกว้างและความสูงของรูปภาพ
- depth : เป็นหน่วยที่กำหนดองค์ประกอบของรูปภาพ
- channels : เป็นตัวเลขที่บอกถึงจำนวนต่อองค์ประกอบของรูปภาพที่มีหน่วยเป็นพิกเซล

13. cvAbsDiff

```
void cvAbsDiff( const CvArr* src1, const CvArr* src2, CvArr* dst );
```

เป็นฟังก์ชั่นที่ใช้ในการหาค่าสัมบูรณ์ระหว่างสองอะเรย์ที่มีค่าแตกต่างกัน โดยมีพารามิเตอร์ ดังนี้

- src1 : เป็นค่าของอะเรย์เริ่มต้น
- src2 : เป็นค่าของอะเรย์ตัวที่สอง
- dst : เป็นค่าของอะเรย์ที่คำนวณออกมาได้

ห้องสมุดคณะวิศวกรรม มหาวิทยาลัย

ประวัติผู้เขียนโครงงาน

ชื่อ นางสาวนิรัชชา สุขอยู่

ภูมิลำเนา 87/1 ถ.นอกทางรถไฟ ต.ในเมือง อ.เมือง จ.พิจิตร 66000

ประวัติการศึกษา

- จบการศึกษาระดับมัธยมศึกษาจากโรงเรียนพิจิตรพิทยาคม
- ปัจจุบันกำลังในระดับปริญญาตรีชั้นปีที่ 4

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยนเรศวร

E – mail : moomoo.lady@hotmail.com

ชื่อ นางสาวธรชญา อินต๊ะ

ภูมิลำเนา 58 ม. 2 ต. หนองป่าครั่ง อ.เมือง จ.เชียงใหม่ 50000

ประวัติการศึกษา

- จบการศึกษาระดับมัธยมศึกษาจากโรงเรียนกาวิละวิทยาลัย
- ปัจจุบันกำลังในระดับปริญญาตรีชั้นปีที่ 4

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยนเรศวร

E – mail : ning_cpe@hotmail.com