ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

การใช้กล้องนับรถโดยวิธีการลบพื้นหลังภาพทิ้ง

Vehicle Counter Using Background Subtraction

นางสาวจิตราภรณ์ สุปิน รหัส 45360104
นายนุกานต์ อุดมบัว รหัส 45360203
นายปรัชญา เทียมลม รหัส 45360260

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 25 / มี.ค. 2553
เลขทะเบียน 1500512-7
เลขเรียกหนังสือ ป
จ458ป
มหาวิทยาลัยนเรศวร

2548

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2548

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	การใช้กล้องนับรถโดยวิธีการลบพื้นหลังภาพทิ้ง		
ผู้ดำเนินโครงงาน	นางสาวจิตราภรณ์	สุปิน	รหัส 45360104
	นายนุกานต์	อุดมบัว	รหัส 45360203
	นายปรัชญา	เทียมลม	รหัส 45360260
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. สุชาติ		แย้มเม่น
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2548		

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. สุชาติ แย้มเม่น)

.....กรรมการ

(อาจารย์ศิริพร เดชะศิลารักษ์)

.....กรรมการ

(ดร.พนมขวัญ ริยะมงคล)

หัวข้อโครงงาน	การใช้กล้องนับรถโดยวิธีการลบพื้นหลังภาพทิ้ง
ผู้ดำเนินโครงงาน	นางสาวจิตราภรณ์ สุปิน รหัส 45360104
	นายนุกานต์ อุดมบัว รหัส 45360203
	นายปรัชญา เทียมลม รหัส 45360260
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. สุชาติ แย้มเม่น
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2548

.....

บทคัดย่อ

โครงงานนี้ มีจุดมุ่งหมายเพื่อพัฒนาโปรแกรมภาษา C# ติดต่อกับกล้องวิดีโอสำหรับการนับรถโดยใช้วิธีการลบพื้นหลังภาพทิ้ง เพื่อนำไปใช้ประโยชน์ในระบบการรายงานผลการจราจรซึ่งเป็นระบบที่ใช้สำหรับเมืองที่มีปัญหาการจราจรคับคั่ง โดยโปรแกรมจะรับภาพจากกล้องวิดีโอทีละเฟรม นำมาแปลงให้เป็นภาพระดับสีเทา ทำพิกเซลเลท ลบพื้นหลังของภาพทิ้ง และทำการนับจำนวนรถ

จากผลการทดลองใช้โปรแกรมสำหรับการนับรถนี้พบว่าเปอร์เซ็นต์ความผิดพลาดในการนับรถจำนวน 20 คันเกิดความผิดพลาด 5 % ซึ่งความผิดพลาดนี้เกิดจากหลายปัจจัยอาทิเช่น ความละเอียดและเฟรมเรตของตัวกล้องวีดิโอ การปรับค่าของโปรแกรมซึ่งได้แก่ ค่าขนาดวัตถุที่ต้องการนับ ค่าขนาดของช่องที่ใช้ตรวจนับวัตถุและค่าความแกว่งของวัตถุให้เหมาะสมกับการตั้งกล้อง นอกจากนี้ โปรแกรม ยังสามารถนำไปพัฒนาต่อเนื่องด้วยการสร้างส่วนติดต่อกับผู้ใช้งานด้านการแก้ปัญหาจราจรในเมืองใหญ่เพิ่มเติมต่อไปได้

Project Title	Vehicle Counter Using Background Subtraction		
Name	Miss. Jittraporn	Supin	ID 45360104
	Mr. Nukarn	Udombua	ID 45360203
	Mr. Pratchaya	Tiamlom	ID 45360260
Project Advisor	Assistant Professor Suchart Yammen Ph.d		
Major	Computer Engineering		
Department	Electrical and Computer Engineering		
Academic Year	2005		

ABSTRACT

The purpose of this project is to develop a program that connect to a video camera for counting car amounts by background subtraction algorithm. This program is useful especially to traffic monitoring system which is used in a big city. The program receives frame from the camera frame by frame then convert them into gray scale. Next, make a pixellet on it, do background subtraction between two frames and finally count the number of cars.

According to the program test, we found that there is 5 % error on 20 cars. This error depends on many factors such as camera're solution and frame rate, program adjustment which is object size, region channel and threshold value. Furthermore, this program can be developed for users who involve in a big city's traffic monitoring system.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สำเร็จได้ด้วยดีเนื่องจากความอนุเคราะห์จากอาจารย์ที่ปรึกษาโครงงาน คือ ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น ที่ให้ความช่วยเหลือ เอาใจใส่ และแนะนำในทุกๆ ด้าน รวมถึงดร.พนมขวัญ ริยะมงคล และอาจารย์ศิริพร เดชะศิลารักษ์ ที่ปรึกษาร่วม ที่ได้เสียสละเวลาทำการตรวจสอบการทำรายงานและชี้แนวทางในการแก้ปัญหาในโครงงานนี้ และท่านอื่นๆ ที่มิได้กล่าวถึง ที่ได้คอยแนะนำและให้คำปรึกษาจนคลายความข้องใจ ซึ่งต้องขอบพระคุณทุกๆ ท่านเป็นอย่างมาก ที่มีส่วนช่วยให้โครงงานนี้ประสบความสำเร็จลุล่วงไปได้ด้วยดี

และต้องขอขอบพระคุณบุคคลที่สำคัญที่สุดที่ทำให้พวกข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูพวกข้าพเจ้ามาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่อย่างเต็มที่ ในทุกๆ ด้านอันหาที่เปรียบมิได้ พวกข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณและขอกราบขอบคุณมา ณ ที่นี้

จิตราภรณ์ สุปีน
ปรัชญา เทียมลม
นุกานต์ อุดมบัว

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....ก

บทคัดย่อภาษาอังกฤษ.....ข

กิตติกรรมประกาศ.....ค

สารบัญ.....ง

สารบัญตาราง.....ฉ

สารบัญรูป.....ช

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของโครงงาน.....1

1.2 วัตถุประสงค์ของโครงงาน.....2

1.3 ขอบข่ายของโครงงาน.....2

1.4 ขั้นตอนการดำเนินงาน.....2

1.5 แผนผังการดำเนินโครงงาน.....3

1.6 ผลที่คาดว่าจะได้รับ.....4

1.7 งบประมาณของโครงงาน.....4

บทที่ 2 ทฤษฎีพื้นฐาน

2.1 การตรวจจับภาพการเคลื่อนไหวโดยใช้ผลต่างของเฟรม.....3

2.2 การแปลงภาพสีไปเป็นภาพระดับสีเทา.....9

2.3 การทำพิกเซลเลส.....10

2.4 ไดเรคโชว์ เอพีไอ (DirectShow API).....10

2.4.1 สถาปัตยกรรมของไดเรคโชว์.....11

2.4.2 ขั้นตอนการสร้างแอพพลิเคชันด้วยไดเรคโชว์.....12

2.4.3 ไดเรคโชว์ฟิลเตอร์.....13

บทที่ 3 การออกแบบฮาร์ดแวร์และซอฟท์แวร์

3.1 ฮาร์ดแวร์.....14

3.2 ซอฟท์แวร์.....15

สารบัญ(ต่อ)

หน้า

บทที่ 4 ผลการทดลอง

4.1 ผลการทดลอง.....27

4.2 ผลการวิเคราะห์.....29

บทที่ 5 สรุปผล

5.1 สรุปผล.....32

5.2 แนวทางในการแก้ไขปัญหาและข้อเสนอแนะ.....33

เอกสารอ้างอิง.....34

ภาคผนวก.....35

ประวัติผู้เขียนโครงงาน.....44

สารบัญตาราง

ตารางที่ หน้า

4.1 แสดงผลการทดลอง.....27

สารบัญรูป

รูปที่ หน้า

2.1 แสดงการลำดับเปรียบเทียบและการเก็บค่าไว้ใน A ของแต่ละเฟรม....5

2.2 แสดงเฟรมที่ 1 และค่าระดับสีเทา....6

2.3 แสดงเฟรมที่ 2 และค่าระดับสีเทา....7

2.4 แสดงการลบเมตริกซ์ของ 2 เฟรม....8

2.5 แสดงค่าที่ได้จากการลบค่าพิกเซลของ 2 เฟรม....8

2.6 แสดงการตีกรอบล้อมรอบวัตถุและสัญลักษณ์....9

2.7 การแบ่งภาพระดับสีเทาเป็นช่องสี่เหลี่ยม....10

2.8 ฟิลเตอร์กราฟสำหรับเล่นกลับไฟล์เอวีไอ....11

2.9 ขั้นตอนการสร้างแอพพลิเคชันด้วยไดเรคโชว์....12

3.1 ตัวกล้องวีดีโอ....14

3.2 ตัวรับสัญญาณจากกล้อง....14

3.3 การ์ดรับสัญญาณ....15

3.4 แสดงเมดธอดในไลบารี่และคลาสต่างๆของไดเรคโชว์....16

3.5 แสดงขั้นตอนการนำภาพมาหาค่าความแตกต่าง....17

3.6 แสดงไลบารี่ของการประมวลผลของภาพและคลาสต่างๆ....17

3.7 แผนภาพแสดงขั้นตอนการนับรถ....18

3.8 แสดงการจับวัตถุที่เคลื่อนที่....19

3.9 แสดงภาพที่ถูกแปลงเป็นภาพระดับขาว-ดำ....19

3.10 ภาพที่ผ่านการทำพิกเซลเลท....19

3.11 ภาพที่ได้จากการตีกรอบ....20

3.12 แสดงการตีกรอบวัตถุและกำหนดสัญลักษณ์....20

3.13 แสดงการนับรถเมื่อรถเคลื่อนที่ผ่านความกว้างของพื้นที่ที่กำหนด(เส้นสีเขียว)....21

3.14 หน้าแรกของโปรแกรม....22

3.15 แสดงการเลือก Open Local Device....23

3.16 แสดงหน้าต่าง Open Local Capture Device....23

3.17 แสดงภาพที่ปรากฏบนฉาก....24

3.18 แสดงการตั้งค่าต่างๆ ของโปรแกรม....25

3.19 การ Reset ค่าใหม่....25

สารบัญรูป(ต่อ)

รูปที่ หน้า

3.20 แสดงจำนวนรถที่นับได้.....26

4.1 แสดงการกำหนดขนาดวัตถุที่เล็กเกินไป.....29

4.2 แสดงการนับรถเมื่อมีการกำหนดขนาดของวัตถุใหญ่เกินไป..... 30

4.3 แสดงการนับรถเมื่อมีการกำหนดขนาดของวัตถุใหญ่เกินไป.....30

4.4 แสดงการนับรถเมื่อกำหนดค่าแบบพอเหมาะ.....31

บทที่1
บทนำ

ปัจจุบันความต้องการในการใช้รถและถนนเพิ่มมากขึ้น ดังนั้นการจัดการระบบจราจรที่ดีจะช่วยให้การใช้รถบนท้องถนนเป็นไปได้อย่างสะดวก ปลอดภัย และลดสภาพการจราจรติดขัดลงได้ โดยโครงงานนี้จะเข้ามาเป็นส่วนหนึ่งของระบบรายงานผลการจราจร โดยใช้กล้องวิดีโอที่มีราคาถูกและวิธีการประมวลผลของภาพ(Image Processing) ซึ่งจะช่วยลดต้นทุนในการสั่งซื้อและติดตั้งระบบ เนื่องจากการนำระบบนี้เข้ามาจากต่างประเทศ จะมีราคาสูงมาก โดยโครงงานนี้จะช่วยในส่วนของการนับจำนวนยานพาหนะบนท้องถนน ซึ่งเมื่อนำไปประยุกต์ใช้กับระบบการจัดการจราจรแล้วก็จะช่วยในการลดปัญหาทางด้านการจราจรลงได้อย่างมีประสิทธิภาพ

1.1 ที่มาและความสำคัญของโครงงาน

เนื่องจากในปัจจุบันนี้ตามเมืองใหญ่ ๆ ของประเทศไทยมีประชากรอาศัยอยู่มาก ทำให้ความต้องการในการใช้รถและถนนจึงมีมากตามไปด้วย ปัญหาการจราจรจึงตามมาอย่างหลีกเลี่ยงไม่ได้ ถึงแม้ว่าจะมีการวางระบบการจราจรไว้ดีเพียงใดก็ตามปัญหาเหล่านี้ก็ยังเกิดขึ้นอยู่เสมอ ซึ่งปัญหาเหล่านี้ทำต้องเสียเวลาเป็นอย่างมากกับการที่ต้องติดอยู่บนถนนที่มีการจราจรติดขัด การที่จะแก้ปัญหาในเรื่องนี้ทำได้ยากมากเพราะปัจจุบันสังคมเมืองมีการขยายตัวตลอดเวลา ยิ่งไปกว่านั้นระบบการดำเนินการจราจรที่ดีและมีประสิทธิ์ภาพนั้นมีราคาค่อนข้างสูง เมื่อเทียบกับผลที่จะได้รับแล้วอาจไม่คุ้มค่ากับการลงทุน

ระบบการรายงานผลการจราจรจึงเป็นแนวทางอีกแนวทางหนึ่งที่อาจจะช่วยแก้ปัญหาในเรื่องเหล่านี้ได้บ้าง เป็นระบบที่สามารถดูความเคลื่อนไหวของรถ นับจำนวนรถได้ และสามารถรายงานผลกับผู้ใช้ด้วยวิธีการต่างๆ หลายวิธีโดยใช้อุปกรณ์ เช่น กล้องวีดีโอ ซึ่งกล้องวีดิโอที่มีประสิทธิภาพดีจะมีราคาที่สูงมาก

ทางผู้จัดทำได้นำปัญหาตรงจุดนี้มาเป็นแนวคิดในการทำโครงงาน โดยการใช้กล้องวีดิโอที่มีราคาถูกเพื่อเป็นการลดต้นทุน และอาศัยวิธีการประมวลผลของภาพเพื่อจะทำการนับจำนวนรถที่ถ่ายภาพได้จากกล้อง และสามารถนำมารายงานผลต่อผู้ที่ต้องการหลีกเลี่ยงเส้นทางการจราจรแออัดเพื่อประหยัดเวลาในการเดินทาง และจะดำเนินงานโดยใช้ต้นทุนที่ต่ำที่สุดเพื่อที่จะสามารถนำไปใช้กับพื้นที่ต่างๆ ได้มากที่สุดตามต้องการ

1.2 วัตถุประสงค์ของโครงงาน

พัฒนาโปรแกรม C# ติดต่อกับกล้องเพื่อใช้นับรถโดยวิธีการลบพื้นหลังภาพทิ้ง

1.3 ขอบข่ายของโครงงาน

1.3.1 สามารถตรวจจับยานพาหนะในลักษณะภาพเคลื่อนไหวที่ได้มาจากกล้องวิดีโอได้

1.3.2 สามารถใช้กล้องราคาถูกนับรถเพื่อประหยัดค่าใช้จ่าย

1.3.3 สร้างส่วนติดต่อกับผู้ใช้ที่เข้าใจได้ง่าย

1.4 ขั้นตอนของการดำเนินงาน

1.4.1 ดำเนินการหาวิธีการประมวลผลของภาพที่เหมาะสมและเป็นไปได้

1.4.2 ศึกษาและทำความเข้าใจเกี่ยวกับการเนื้อหาของวิธีการเลือกอย่างละเอียด

1.4.3 ศึกษาลักษณะของภาพที่ได้จากการถ่ายด้วยกล้องวิดีโอ

1.4.4 เขียนโปรแกรมที่จะใช้ดำเนินการกับภาพ

1.4.5 ทดลองนำโปรแกรมที่เขียนขึ้นไปทำการตรวจนับจำนวนรถ จากภาพที่ได้จากกล้องวีดีโอ

1.4.6 แก้ไขข้อบกพร่องของโปรแกรม

1.4.7 เขียนโปรแกรมส่วนติดต่อกับผู้ใช้

1.4.8 ทดลองกับสภาพการใช้งานจริง

1.4.9 แก้ไขข้อบกพร่อง

1.4.10 สรุปและจัดทำเอกสาร

1.5 แผนผังการดำเนินโครงงาน

เวลา / การดำเนินงาน	พ.ย. 2548	ธ.ค. 2548	ม.ค. 2549	ก.พ. 2549	มี.ค. 2549	เม.ย 2549	พ.ค. 2549	มิ.ย. 2549
หาวิธีการประมวลผลของภาพที่เหมาะสม	←	→						
ศึกษาทำความเข้าใจเนื้อหาที่เลือกมา		←	→					
ศึกษาลักษณะของภาพที่ได้จากการถ่ายด้วยกล้องวีดีโอ				←	→			
เขียนโปรแกรมที่จะใช้ดำเนินการกับภาพ				←	→			
ทดลองนำโปรแกรมที่เขียนได้ไปใช้งาน					←	→		
แก้ไขข้อบกพร่องของโปรแกรม					←	→	→	
เขียนโปรแกรมในส่วนติดต่อกับผู้ใช้					←	→	→	
ทดลองกับสภาพการใช้งานจริง					←	→	→	
แก้ไขข้อบกพร่อง						←	→	→
สรุปและจัดทำเอกสาร						←	→	→

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 สามารถเข้าใจหลักการประมวลผลของภาพและนำไปประยุกต์ใช้งานได้จริง

1.6.2 สามารถนำหลักการประมวลผลของภาพไปทำงานร่วมกับกล้องวิดีโอได้

1.6.3 สามารถนำโครงงานไปเป็นแนวทางในการพัฒนาเป็นส่วนหนึ่งของระบบรายงานผลการจราจรได้

1.7 งบประมาณของโครงงาน

1.7.1 ค่าวัสดุสำนักงาน	เป็นเงิน	1,000	บาท
1.7.2 ค่าวัสดุคอมพิวเตอร์	เป็นเงิน	1,000	บาท
1.7.3 ค่าถ่ายเอกสารพร้อมเข้าเล่ม	เป็นเงิน	1,000	บาท
รวมเป็นเงินทั้งสิ้น	3,000	บาท (สามพันบาทถ้วน)	

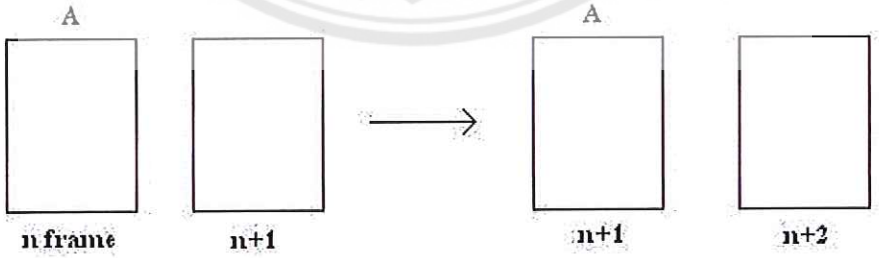
หมายเหตุ (ถ้วเฉลี่ยทุกรายการ)

บทที่ 2
ทฤษฎีพื้นฐาน

การตรวจจับวัตถุเคลื่อนที่แบบพื้นหลังเคลื่อนไหว (Dynamic background) จะคล้ายกับวิธีการตรวจจับวัตถุแบบพื้นหลังหยุดนิ่ง (Static background) โดยวิธีการแบบพื้นหลังเคลื่อนไหวจะมีประโยชน์ในกรณีที่ไม่สามารถคาดเดาพื้นหลัง (Background) ได้เนื่องจากเป็นภาพที่ได้มาจากกล้องวิดีโอ ซึ่งทำให้ไม่สามารถรู้ได้ว่าจะมีวัตถุอะไรเคลื่อนที่ผ่านหน้ากล้องไปบ้าง จึงต้องมีการเปลี่ยนแปลงพื้นหลังที่จะใช้ในการเปรียบเทียบตลอดเวลา จะสังเกตว่าภาพเคลื่อนไหวที่เกิดขึ้นใหม่นั้น พื้นหลังที่เปลี่ยนไปเกิดจากการเคลื่อนไหวของวัตถุในภาพหรือว่าเป็นการเคลื่อนไหวของกล้องที่ใช้ จึงทำการเปรียบเทียบเป็นแบบเฟรมก่อนหน้ามาเทียบกับเฟรมปัจจุบัน ซึ่งเป็นวิธีที่ง่ายกว่าการนำพื้นหลังตัวแรกไปเปรียบเทียบกับพื้นหลังตัวถัดไปตลอด เรียกวิธีการนี้ว่า การตรวจจับภาพการเคลื่อนไหวโดยใช้ผลต่างของเฟรม (Frame differentiating)

2.1 การตรวจจับภาพการเคลื่อนไหวโดยใช้ผลต่างของเฟรม(Frame differentiating)

การเปรียบเทียบจะเริ่มจากการรับภาพสี (RGB) จากกล้องวิดีโอเข้ามาทีละ 1 เฟรม(Frame) นำมาเก็บไว้ในรูปของบิตแมพ A (Bitmap A) เมื่อเฟรมแรกที่เข้ามา คือ เฟรมที่ n จะนำมาเก็บไว้ใน A ก่อนเพื่อรอภาพในเฟรมถัดไปมาเปรียบเทียบค่าพิกเซล (Pixel) เมื่อภาพในเฟรมถัดไป คือ เฟรมที่ n+1 เข้ามาจะนำมาเปรียบเทียบค่าพิกเซลกับเฟรมที่ n เมื่อทำการเปรียบเทียบแล้วจะนำเฟรมที่ n+1 มาเก็บไว้ใน A เพื่อรอภาพในเฟรมถัดไปเข้ามาเปรียบเทียบ ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 แสดงลำดับการเปรียบเทียบและการเก็บค่าไว้ใน A ของแต่ละเฟรม

รูปที่ 2.1 เป็นการเปรียบเทียบค่าระหว่างเฟรมที่ n ซึ่งถูกเก็บไว้ใน A กับเฟรมที่ n+1 ซึ่งเป็นเฟรมที่เพิ่งเข้ามาใหม่ เมื่อทำการเปรียบเทียบแล้วจะเก็บเฟรมที่ n+1 ไว้ใน A เมื่อรับภาพเฟรมที่ n+2 เข้ามา จะนำมาเปรียบเทียบกับเฟรมที่ n+1 โดยการเลือกค่า n จะต้องเป็นค่าที่พอเหมาะ จะต้องไม่มากเกินไปหรือ

น้อยเกินไป ซึ่งถ้าหากค่า n มากเกินไปจะไม่สามารถตรวจจับวัตถุเคลื่อนที่ได้ในบางกรณี และถ้าน้อยเกินไปจะไม่สามารถตรวจจับวัตถุที่เคลื่อนที่ผ่านหน้ากล้องอย่างช้าๆ ได้

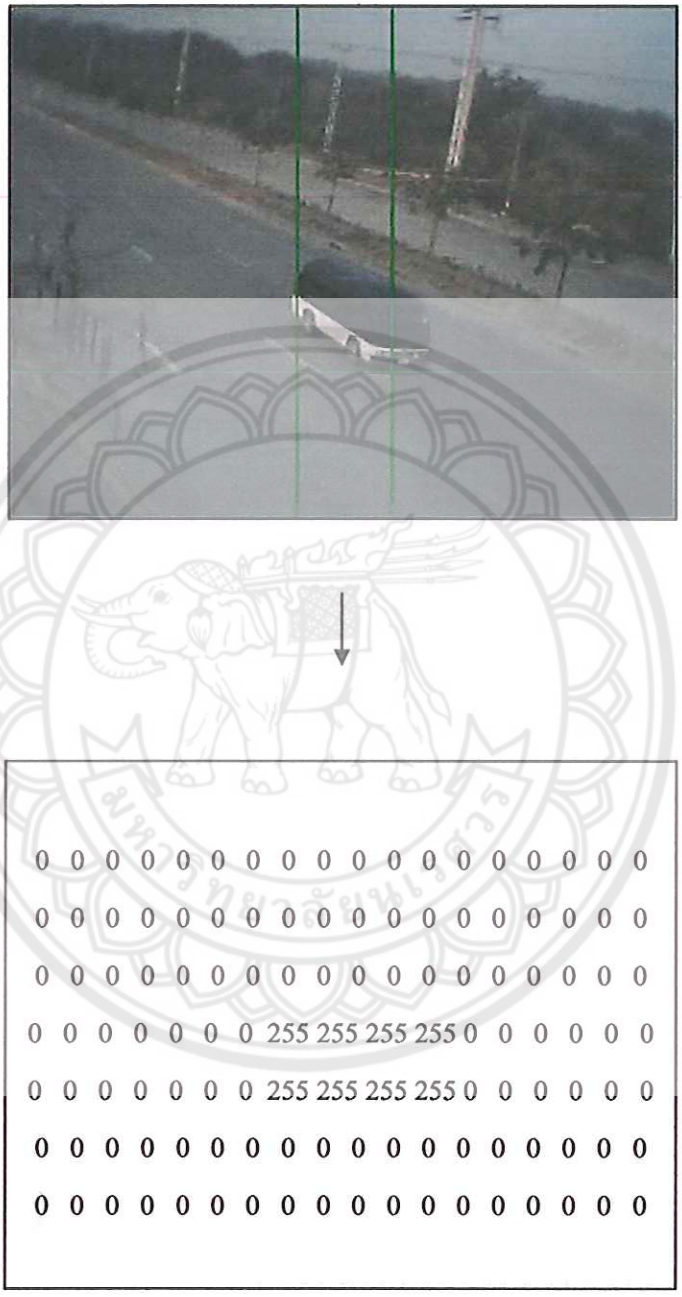
การเปรียบเทียบค่าพิกเซลระหว่างเฟรม 2 เฟรม จะทำโดยนำค่าของพิกเซลของทั้ง 2 เฟรมมาลบกัน ซึ่งถ้าหากเฟรม 2 เฟรมเป็นภาพที่แตกต่างกัน จะได้ผลลัพธ์ของค่าความต่างออกมาค่าหนึ่ง ซึ่งถ้าจำนวนพิกเซลที่มีความแตกต่างกันมีจำนวนมากเกินกว่าค่าเทรสโฮลแวลู(Threshold Value) จะทำการตีกรอบล้อมรอบส่วนที่เกิดความต่างนั้นไว้ ในส่วนที่ไม่มีความแตกต่างไม่ต้องกระทำการใดๆ เนื่องจากจะสนใจเฉพาะพิกเซลที่มีความแตกต่างกัน นำวิธีการเปรียบเทียบผลต่างของเฟรมมาใช้ในการตรวจจับวัตถุเคลื่อนที่ตามขั้นตอนดังนี้

1. รับภาพจากกล้องวิดีโอซึ่งเป็นภาพสีเข้ามาทีละ 1 เฟรม ดังแสดงในรูปที่ 2.2 ภาพบน นำแต่ละเฟรมมาแปลงเป็นภาพระดับสีเทาซึ่งจะกล่าวในหัวข้อ 2.2 แล้วนำภาพระดับสีเทามาทำพิกเซลเลท (Pixelate) ซึ่งถูกอธิบายในหัวข้อ 2.3 เมื่อได้ภาพที่ทำพิกเซลเลทแล้วจะนำมาเก็บค่าในลักษณะเมทริกซ์ (Matrix) 2 มิติ โดยแถวของเมทริกซ์จะสัมพันธ์กับความสูงของภาพ และหลักของเมทริกซ์จะสัมพันธ์กับความกว้างของภาพ ซึ่งแต่ละสมาชิกในเมทริกซ์จะเก็บค่าระดับสีเทาโดยจะเก็บไว้ในเมทริกซ์อ้างอิง ดังรูปที่ 2.2 ภาพล่างเป็นค่าระดับสีเทาที่แสดงในลักษณะเมทริกซ์ ซึ่งเป็นค่าที่สมมติขึ้นมา

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

รูปที่ 2.2 แสดงเฟรมที่ 1 และค่าระดับสีเทา

2. นำภาพสีในเฟรมถัดไป ดังรูปที่ 2.3 ภาพบน มาแปลงเป็นภาพระดับสีเทา และทำพิกเซลเลท จากนั้นนำมาเก็บค่าในลักษณะเมทริกซ์ ดังรูปที่ 2.3 ภาพล่างซึ่งค่าที่แสดงเป็นค่าสมมติ



รูปที่ 2.3 แสดงเฟรมที่ 2 และค่าระดับสีเทา

3. นำเมตริกซ์ที่เก็บค่าของเฟรมที่1 ลบกับเมตริกซ์ที่เก็บค่าของเฟรมถัดไป ดังรูปที่ 2.4 ตามหลักการลบเมตริกซ์ซึ่งมีบทนิยามและเงื่อนไขดังนี้

<u>บทนิยาม</u> ให้ $A = [a_{ij}]_{m\times n}$ และ $B = [b_{ij}]_{m\times n}$ จะได้ว่า $A - B = [C_{ij}]_{m\times n}$ โดยที่ $C_{ij} = a_{ij} - b_{ij}$

การนำเมทริกซ์ 2 เมทริกซ์มาลบกัน มีเงื่อนไข 2 ประการ กล่าวคือ

1. เมทริกซ์ที่จะนำมาลบกันต้องมีมิติเท่ากัน
2. นำสมาชิกที่อยู่ในตำแหน่งเดียวกันมาลบกัน

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	255	255	255	255	0	0	0	0	0			
0	0	0	255	255	255	255	0	0	0	0	0			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

—

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

รูปที่ 2.4 แสดงการลบเมตริกซ์ของ 2 เฟรม

จากรูปที่ 2.4 เมทริกซ์แรกเป็นเมทริกซ์ของภาพในเฟรมที่ 1 เมทริกซ์หลังเป็นเมทริกซ์ของภาพในเฟรมถัดไป

4. นำผลลัพธ์ที่ได้จากการลบเมทริกซ์มาพิจารณาการเคลื่อนที่ของวัตถุ

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	255	255	255	255	0	0	0	0	0	0	
0	0	0	0	0	0	0	255	255	255	255	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

รูปที่ 2.5 แสดงค่าที่ได้จากการลบค่าพิกเซลของ 2 เฟรม

จากรูปที่ 2.5 คือผลลัพธ์ที่ได้จากการลบเมทริกซ์ในข้อ 3 จะเห็นว่าค่าที่ลบกันแล้วได้ 0 คือค่าพิกเซลที่ไม่มีการเปลี่ยนแปลงใดๆ แต่หากลบกันแล้วได้ค่า 255 แสดงว่า ณ จุดพิกเซลของ 2 ภาพมีการเปลี่ยนแปลงเกิดขึ้น

5. บริเวณที่มีการเปลี่ยนแปลง ถ้าหากมีขนาดใหญ่กว่าค่าเทรสโฮลแวลูที่กำหนดไว้จะตีกรอบล้อมรอบบริเวณนั้น ดังรูปที่ 2.6

รูปที่ 2.6 แสดงการตีกรอบล้อมรอบวัตถุและสัญลักษณ์

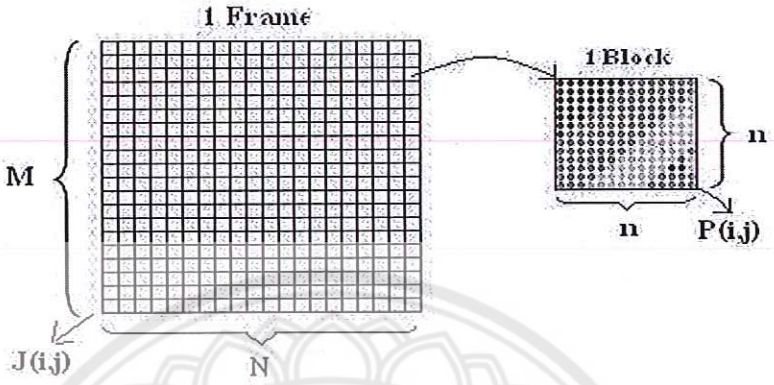
2.2 การแปลงภาพสีไปเป็นภาพระดับสีเทา

ภาพระดับสีเทาสร้างมาจากค่าเฉลี่ยของค่าสีแดง เขียว และน้ำเงินของภาพสี ซึ่งภาพสีสามารถเขียนเป็นภาพ I(i, j, k) โดยที่ I(i, j, k) คือ ค่าสีของภาพแต่ละพิกเซลบนตำแหน่ง (i, j, k) ซึ่ง i คือ ตำแหน่งของดัชนีแถว j คือ ตำแหน่งของดัชนีหลัก และ k คือ ตำแหน่งดัชนีแนวลึก และภาพระดับสีเทาสามารถเขียนเป็นภาพ J(i, j) โดยที่ J(i, j) คือ ค่าระดับสีเทาของภาพแต่ละพิกเซลบนตำแหน่ง (i, j) ซึ่ง i คือ ตำแหน่งของดัชนีแถว และ j คือ ตำแหน่งของดัชนีหลัก สำหรับทุกๆ พิกเซล I(i, j, k) บนภาพบิตแมพ การแปลงภาพสี I(i, j, k) เป็นภาพระดับสีเทา J(i, j) มีสูตรดังนี้

$$J(i,j) = \frac{\sum_{k=1}^{3} I(i,j,k)}{3} \qquad (2.1)$$

2.3 การทำพิกเซลเลท (Pixelate)

การทำพิกเซลเลทเป็นการนำภาพระดับสีเทา J(i, j) ขนาด M × N พิกเซล มาแบ่งเป็นช่องสี่เหลี่ยมจตุรัส(Block) P(i, j) ซึ่งแต่ละช่องมีขนาด n ×n พิกเซล ดังรูปที่ 2.7



รูปที่ 2.7 การแบ่งภาพระดับสีเทาเป็นช่องสี่เหลี่ยม

นำแต่ละพิกเซลในช่องสี่เหลี่ยมจตุรัสขนาด n ×n พิกเซล มาหาค่าเฉลี่ยตามสูตร 2.2 ดังนี้

$$P(i,j) = \frac{\sum_{j=1}^{N}\sum_{i=1}^{M} J(i,j)}{n * n} \qquad \textbf{(2.2)}$$

โดยที่ P(i, j) คือ ค่าระดับสีเทาของภาพพิกเซลเลทแต่ละพิกเซลบนตำแหน่ง (i, j)

J(i, j) คือ ค่าระดับสีเทาของภาพแต่ละพิกเซลบนตำแหน่ง (i, j)

n * n คือ ขนาดของช่องสี่เหลี่ยมจตุรัส 1 ช่อง

โดยในโครงงานนี้ใช้ค่า n = 20

2.4 ไดเรคโชว์ เอพีไอ (DirectShow API)

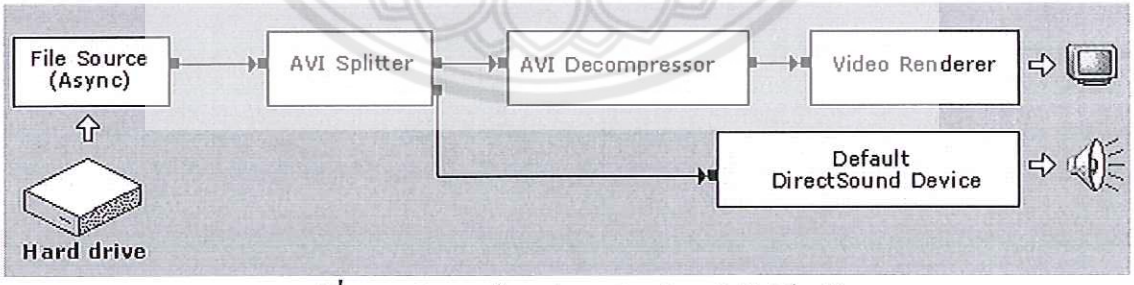
ไมโครซอฟท์ไดเรคโชว์เอพีไอ(Microsoft DirectShow API) เป็นสถาปัตยกรรมสำหรับการแสดงผลสื่อข้อมูลบนระบบปฏิบัติการไมโครซอฟท์วินโดว์(Microsoft Windows Platform) โดยเป็นส่วนหนึ่งของไมโครซอฟท์ไดเรคเอ็กซ์ เอพีไอ(Microsoft DirectX API) ซึ่งเราสามารถนำไดเรคโชว์มาใช้ในด้านการสร้างแอพพลิเคชั่นเกี่ยวกับมัลติมีเดียในรูปแบบต่างๆ ได้ เช่น โปรแกรมดำเนินกับภาพ ที่ได้มาจากกล้องและข้อมูลในลักษณะไฟล์วีดีโอ(VDO) โดยไดเรคโชว์จะเป็นเหมือนสะพานหรือล่ามเพื่อให้อุปกรณ์และซอฟท์แวร์สื่อสารกันได้รู้เรื่อง โดยสามารถดึงการทำงานของอุปกรณ์ให้ทำงานร่วมกับฮาร์ดแวร์ได้อย่างมีประสิทธิภาพ ไดเรคโชว์ใช้หลักการของCOM(Component Object Model) ในการเขียนไดเรคโชว์แอพพลิเคชั่น

2.4.1 สถาปัตยกรรมของไดเรคโชว์

ไดเรคโชว์เกิดจากเอาหน่วยย่อยของซอฟท์แวร์ที่ดำเนินการกับข้อมูล(Software Component) ที่เรียกว่า ฟิลเตอร์ (Filter) มาต่อกัน หลักการทำงานของไดเรคโชว์จะควบคุมเกี่ยวกับการไหล (Flow) ของข้อมูลพวกมัลติมีเดียทั้งหลาย โดยจะไหลผ่านบล็อก (ดังในรูป2.8) ที่ทำงานควบคุม และ กลั่นกรองการไหลของข้อมูล การทำงานหลาย ๆ บล็อก หรือโมดูล ซึ่งจะส่งข้อมูลผ่านกันเป็นทอด ๆ ซึ่งอาจจะเป็นไปได้ที่ตัวของบล็อกนั้นจะทำการแสดงผลด้วยตัวมันเองเมื่อเจอข้อมูลที่เหมาะสม หรือย้ายข้อมูลที่มันมีอยู่ไปยังบล็อกต่อไป ซึ่งเราเรียกกันว่ากระบวนการกรองข้อมูล จะมีขั้นตอนดังนี้

- อ่านข้อมูลดิบจากไฟล์ (File Source Filter) อาจจะเป็นฮาร์ดดิสก์, ซีดี, ดีวีดีรอมหรือ กล้องวีดีโอเป็นต้น
- ตรวจสอบส่วนหัวของ Header file AVI นั้นและทำการตรวจว่าจะวิเคราะห์แยกออกมาเป็นส่วนของ เสียง และ เฟรมภาพวีดีโอ ด้วยมาตรฐานใด (AVI Splitter)
- ถอดรหัสของเฟรมภาพวีดีโอที่ถูกบีบไว้ให้อยู่ในรูปที่สามารถแสดงผลได้ (สาเหตุที่มีการใช้ filter ที่หลากหลายในปัจจุบันก็เนื่องจากมีการเข้ารหัสไฟล์ หรือ การบีบอัดไฟล์ ตามรูปแบบมาตรฐานที่หลากหลายนั่นเอง)
- วาดภาพเฟรมของ Video (Video Renderer)
- ส่งรูปแบบเสียงที่วิเคราะห์ไปยัง Sound card (Default DirectSound Device)

สำหรับรูปที่ 2.8เป็นตัวอย่างของบล็อกหรือฟิลเตอร์ที่จะทำการแสดงไฟล์เอวีไอผ่านการทำงานของไดเรคโชว์



รูปที่ 2.8 ฟิลเตอร์กราฟสำหรับเล่นกลับไฟล์เอวีไอ

ประเภทของฟิลเตอร์สามารถแบ่งเป็นกลุ่ม ๆ ตามการทำงานได้ดังนี้

File Source Filter ทำหน้าที่ติดต่อกับข้อมูลให้กับ Filter graph อาจจะเป็นจากไฟล์ หรือ ระบบ Network หรือแม้กระทั่งในกล้องวีดีโอก็ตาม หรือนอกจากที่กล่าวมาข้างต้น ดังนั้นกลุ่มของฟิลเตอร์พวกนี้จะต่างกันไป ตามแต่ชนิดของ แหล่งข้อมูลที่เราจะทำการติดต่อ

AVI Splitter Filter จะทำหน้าที่แยกข้อมูลออกเป็น 2 หรือ มากกว่านี้ซึ่งขึ้นอยู่กับสายข้อมูลที่รับเข้ามาเช่น ข้อมูลวิดีโอที่ถูกบีบอัด และข้อมูลออดิโอ (Audio)

AVI Decompressor Filter จะทำการคลายข้อมูลวิดีโอที่ได้รับการบีบอัดได้ผลลัพธ์เป็นข้อมูลของวิดีโอแต่ละเฟรม

Video Renderer Filter เป็นฟิลเตอร์ที่จะทำหน้าที่เป็นตัวสุดท้ายของระบบ ซึ่งจะคอยแสดงผลที่ได้มาให้กับผู้ใช้ เช่น ตัว Video renders จะทำหน้าที่วาดเฟรมของวีดีโอไปยังหน้าจอ และตัว Audio render จะทำการ ส่งข้อมูลเสียงไปยัง Sound Card และ file-writer filter จะทำหน้าที่เขียนข้อมูลลงไฟล์

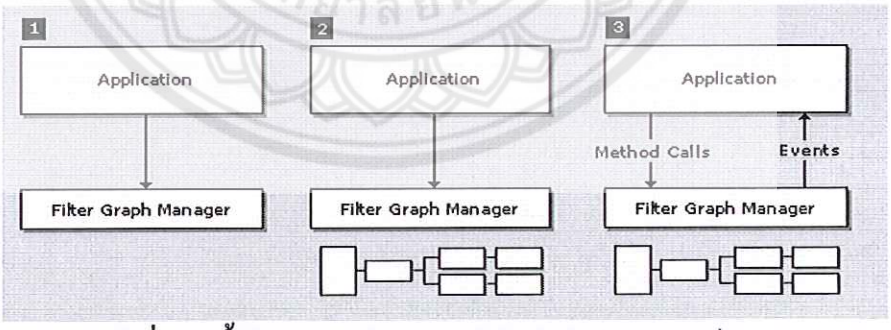
Default DirectSound Device Filter จะทำการแสดงเสียงออกทางลำโพง

แอพพลิเคชั่นจะควบคุมฟิลเตอร์เหล่านี้ได้โดยผ่านทางฟิลเตอร์กราฟเมเนเจอร์ (Filter Graph Manager) ซึ่งฟิลเตอร์กราฟเมเนเจอร์นี้จะคอยรับรู้การสั่งการผ่านแอพพลิเคชั่น เช่นการสั่งให้ฟิลเตอร์กราฟเริ่มต้นทำงานหรือหยุดทำงานได้โดยผ่านฟิลเตอร์กราฟเมเนเจอร์ แอพพลิเคชั่นยังสามารถรับรู้เหตุการณ์ที่เกิดภายในฟิลเตอร์กราฟโดยโดยการแจ้งเหตุการณ์ (Event) เป็นข้อความ (Message) ที่ส่งผ่านฟิลเตอร์กราฟเมเนเจอร์

นอกจากฟิลเตอร์กราฟเมเนเจอร์จะช่วยในการควบคุมการทำงานของฟิลเตอร์กราฟแล้วยังช่วยในการสร้างฟิลเตอร์กราฟอีกด้วย

2.4.2 ขั้นตอนการสร้างแอพพลิเคชันด้วยไดเรคโชว์

ก่อนที่จะทำการสร้างแอพพลิเคชันด้วยไดเรคโชว์ต้องทำการติดตั้ง Library dshow.dll เข้าไปในโฟลเดอร์อ้างอิง (Reference folder) ของC#ก่อน เพื่อให้C#สามารถเรียกใช้คำสั่งต่างๆ ในการสร้างแอพพลิเคชั่นได้ ขั้นตอนในการสร้างโปรแกรมที่ใช้ไดเรคโชว์มีอยู่ 3 ขั้น



รูปที่ 2.9 ขั้นตอนการสร้างแอพพลิเคชันด้วยไดเรคโชว์

1. สร้างอินสแตน (Instance) ของฟิลเตอร์กราฟเมเนเจอร์
2. ใช้ฟิลเตอร์กราฟเมเนเจอร์ ในการสร้างฟิลเตอร์กราฟ
3. เริ่มต้นการทำงานของฟิลเตอร์กราฟ

2.4.3 ไดเรคโชว์ฟิลเตอร์

ไดเรคโชว์ฟิลเตอร์คือส่วนที่ทำหน้าที่อย่างใดอย่างหนึ่งกับข้อมูลมัลติมีเดีย ไดเรคโชว์ประกอบด้วยฟิลเตอร์มากมายซึ่งเป็นฟิลเตอร์มาตรฐานอยู่แล้ว ผู้ใช้สามารถสร้างฟิลเตอร์ขึ้นใช้ได้เอง โดยการสร้างฟิลเตอร์สืบทอดจากฟิลเตอร์ที่มีอยู่แล้ว กล่าวคือสามารถเปลี่ยนแปลงขั้นตอนการกรองข้อมูลได้ด้วยตัวผู้ใช้เอง ทั้งนี้ขึ้นอยู่กับลักษณะงานที่ต้องการจะทำ

ฟิลเตอร์จะประกอบด้วยพิน (Pin) ที่ใช้สำหรับส่งผ่านข้อมูลจากฟิลเตอร์หนึ่งไปยังอีกฟิลเตอร์หนึ่ง พินเป็นซีโอเอ็มอ๊อบเจคต์สามารถแบ่งได้ 2 ชนิดคือ

1. อินพุทพิน (Input pin) จะทำหน้าที่รับข้อมูลเข้าสู่ฟิลเตอร์
2. เอาท์พุทพิน (Output pin) จะทำหน้าที่ส่งข้อมูลออกจากฟิลเตอร์

ฟิลเตอร์บางชนิดอาจจะประกอบด้วยอินพุทพินเพียงอย่างเดียว เช่น Renderer Filter หรือบางชนิดอาจจะมีเพียงเอาท์พุทพินเพียงอย่างเดียวเช่น Source Filter หรืออาจจะประกอบด้วยทั้งอินพุทและเอาท์พุทพิน เช่น Transform Filter ฟิลเตอร์อาจจะประกอบด้วยอินพุทหรือเอาท์พุทมากกว่า 1 ตัว เราสามารถแบ่งฟิลเตอร์ในไดเรคโชว์ออกเป็นกลุ่มได้ดังนี้

Source Filter

ทำหน้าที่ติดต่อกับข้อมูลให้กับฟิลเตอร์กราฟ อาจจะเป็นการติดต่อจากไฟล์ ระบบเน็ตเวิร์ค หรือกล้องวีดีโอ ดังนั้นกลุ่มของฟิลเตอร์นี้จะแตกต่างกันไปตามแหล่งชนิดของข้อมูลที่ทำการติดต่อ

Transform Filter

ทำหน้าที่นำข้อมูลเข้ามาเป็นสายข้อมูล(Stream) การส่งผ่านข้อมูลไปฟิลเตอร์ต่างๆ และนำข้อมูลออกเป็นสายข้อมูลเพื่อนำไปใช้งานต่อ

Renderer Filter

เป็นฟิลเตอร์ที่นำหน้าที่เป็นตัวสุดท้ายของระบบ ซึ่งจะทำการแสดงข้อมูลที่ได้มาให้กับผู้ใช้ เช่น วีดีโอเร็นเดอจะทำหน้าที่วาดเฟรมของวีดีโอออกทางหน้าจอ

Splitter Filter

ทำหน้าที่แยกข้อมูลที่เข้ามทางอินพุตพินออกเป็นตั้งแต่ 2 สายขึ้นไปออกทางเอาท์พุทพิน เช่น AVI Spliter Filter ทำหน้าที่แยกสัญญาณภาพและเสียงของข้อมูลเอวีไอออกจากกัน

Mux Filter

ทำหน้าที่รวมข้อมูลตั้งแต่ 2 สายขึ้นไปให้ออกมาเป็นข้อมูลสายเดียว เช่น AVI Mux ทำหน้าที่รวมข้อมูลภาพและเสียงแล้วส่งออกไปพร้อมกัน

บทที่ 3
การออกแบบฮาร์ดแวร์และซอฟท์แวร์

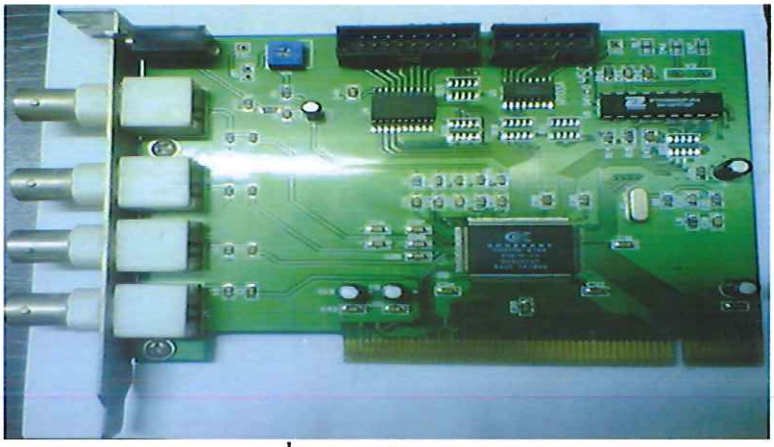
ในส่วนการออกแบบจะแบ่งออกเป็น 2 ส่วน เพื่อให้ง่ายแก่การเข้าใจ โดยส่วนแรกจะกล่าวถึงส่วนฮาร์ดแวร์ซึ่งเป็นอุปกรณ์ที่ใช้งาน และส่วนที่สองจะกล่าวถึงส่วนซอฟท์แวร์ซึ่งเป็นโปรแกรมที่ใช้ติดต่อกับทางฮาร์ดแวร์และวิธีการใช้งานโปรแกรม

3.1 ฮาร์ดแวร์

ในส่วนนี้จะใช้อุปกรณ์เพียงตัวเดียวคือ กล้องวิดีโอ ซึ่งกล้องวิดีโอประกอบด้วย ตัวกล้องวิดีโอ (รูปที่ 3.1) ตัวรับสัญญาณจากกล้อง (รูปที่ 3.2) การ์ดรับสัญญาณ (รูปที่ 3.3)

รูปที่ 3.1 ตัวกล้องวีดีโอ

รูปที่ 3.2 ตัวรับสัญญาณจากกล้อง



รูปที่ 3.3 การ์ดรับสัญญาณ

ซึ่งแต่ละอุปกรณ์จะมีหน้าที่ดังนี้

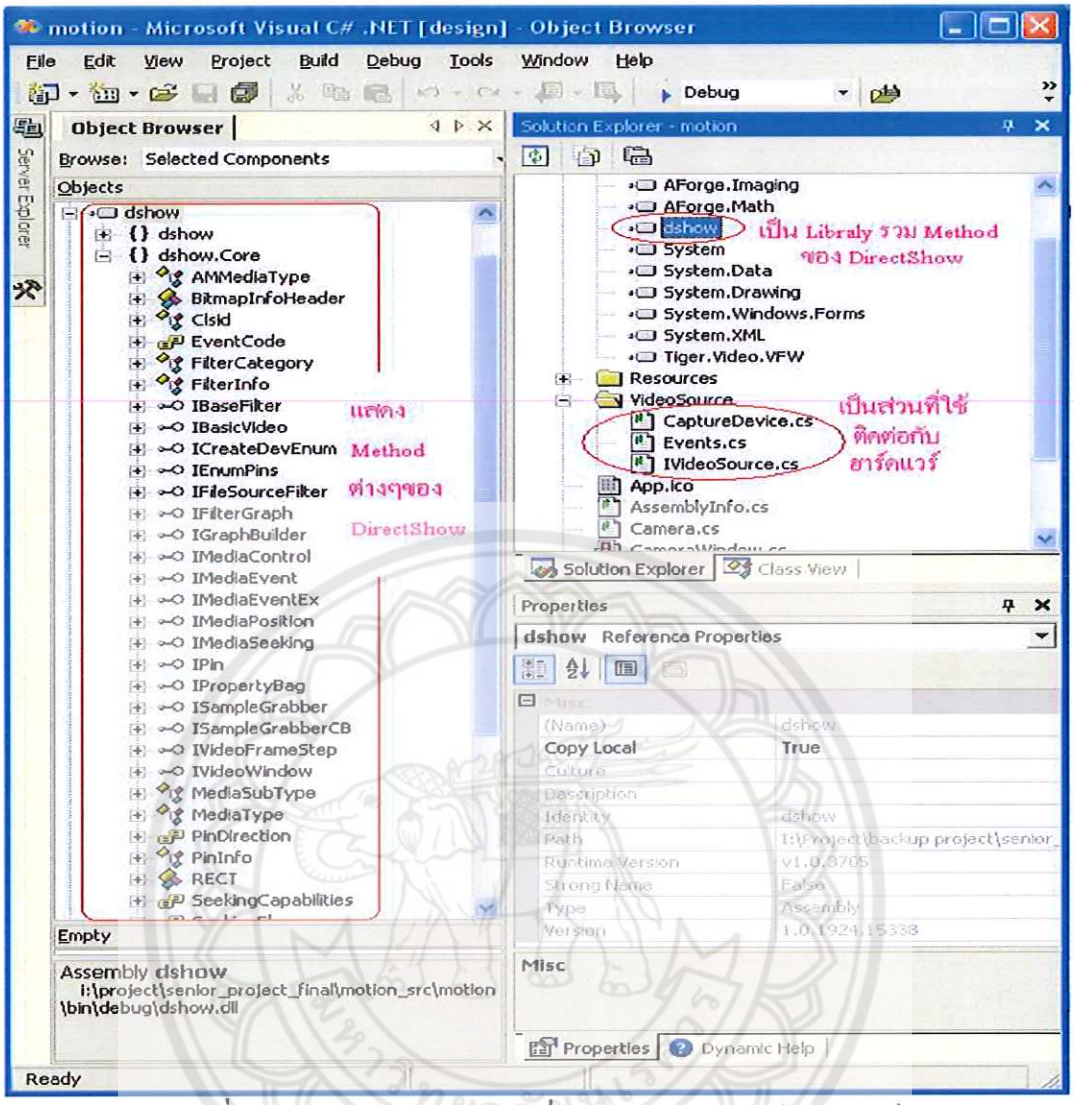
- ตัวกล้องวีดีโอ ทำหน้าที่จับภาพรถเข้ามาประมวลผล
- ตัวรับสัญญาณ เนื่องจากตัวกล้องที่ใช้เป็นกล้องไร้สาย จึงต้องมีตัวรับสัญญาณ
- การ์ดรับสัญญาณ ทำหน้าที่รับสัญญาณจากตัวรับสัญญาณโดยใช้โปรแกรมภาษา C# ในติดต่อกับการ์ด ซึ่งทำให้สามารถติดต่อกับกล้องผ่านไดรเวอร์ได้

การทำงานของกล้องวิดีโอจะเริ่มจากถ่ายภาพแล้วส่งสัญญาณภาพที่ได้ ผ่านเข้ามายังตัวรับสัญญาณ ซึ่งทำงานในลักษณะการส่งคลื่นเข้ามายังตัวรับคลื่น แล้วตัวรับคลื่นดังกล่าวจะส่งสัญญาณวีดีโอเข้าการ์ดรับสัญญาณ ซึ่งการ์ดรับสัญญาณนี้จะติดตั้งอยู่บนช่องพีซีไอ(PCI Slot) ในเครื่องคอมพิวเตอร์

3.2 ซอฟท์แวร์

โปรแกรมส่วนติดต่อกับฮาร์ดแวร์

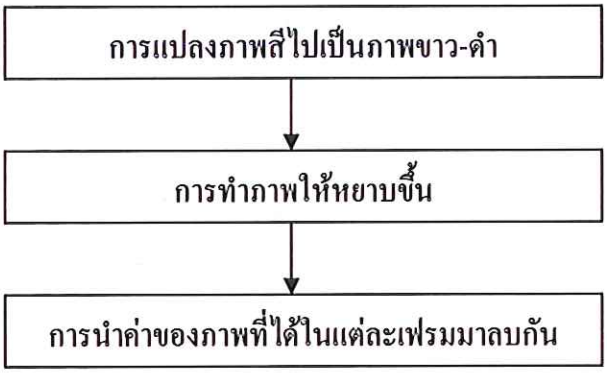
โปรแกรมส่วนติดต่อกับฮาร์ดแวร์ในโครงงานนี้จะใช้ไลบารี่(Library) ที่มีเมดธอต (Method) ของไดเรคโชว์ที่สามารถนำไปทำการพัฒนาโปรแกรมในภาษา C# ได้ และคลาสที่ทำการติดต่อกับอุปกรณ์รับภาพวีดีโอ รวมทั้งคลาสที่เขียนขึ้นเพื่อทำการรองรับเหตุการณ์ต่างๆ ของการดำเนินการเกี่ยวกับกล้อง ซึ่งคลาสและไลบารี่ต่างๆ ดังกล่าวมีสำเร็จรูปแล้ว จึงสามารถนำมาใช้เป็นเครื่องมือส่วนหนึ่งในการพัฒนาโครงงานได้โดยไม่ต้องเขียนโปรแกรมขึ้นมาใหม่ โดยลักษณะของคลาสและไลบารี่ต่าง ๆ จะแสดงดังรูปที่ 3.4



รูปที่ 3.4 แสดงเมดธอดในไลบารี่และคลาสต่างๆ ของไดเรคโชว์

โปรแกรมส่วนที่ใช้ตรวจจับวัตถุเคลื่อนที่และนับจำนวนรถ

ส่วนนี้จะนำภาพที่ได้จากการดำเนินงานในส่วนของฮาร์ดแวร์มาทำการตรวจหาวัตถุเคลื่อนที่ ทำการนับและนำมาแสดงผล โดยจะมีการเรียกใช้เมดธอดทางวิธีการประมวลผลของภาพต่างๆ หลายตัว ซึ่งเมดธอดเหล่านี้มาจากไลบารี่ทางวิธีการประมวลผลของภาพ ที่ชื่อว่า AForge.Imageing, AForge.Math ซึ่งจะรวมเอาเมดธอดทางวิธีการประมวลผลของภาพไว้มากมายและสามารถนำมาใช้ประโยชน์ในการพัฒนาแอบพลิเคชั่น (Application) บนภาษา C# ได้ง่ายและมีประสิทธิภาพตามแนวทางที่ได้นำเสนอไปในส่วนของบทที่ 2 เช่น การสร้างฟิลเตอร์ต่างๆ เพื่อที่จะนำมากระทำกับภาพ ซึ่งจะทำให้ได้ผลในรูปแบบต่าง ๆ ดังรูปที่ 3.5

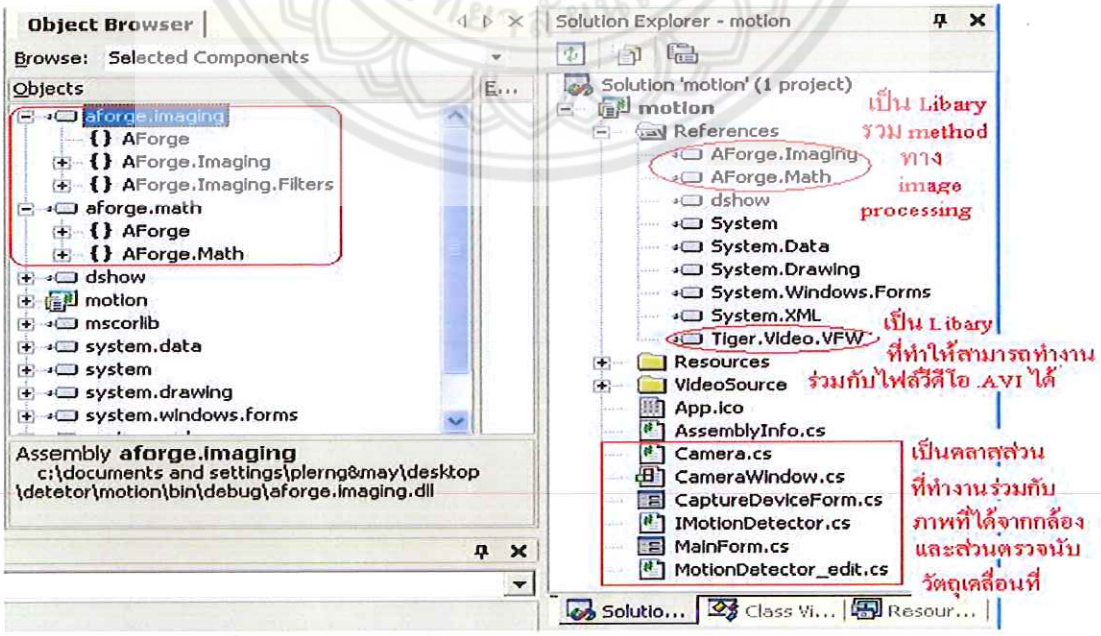


รูปที่ 3.5 แสดงขั้นตอนการนำภาพมาหาค่าความแตกต่าง

จากรูปที่ 3.5 ในขั้นตอนของการนำภาพมาลบกันเพื่อที่จะหาค่าความแตกต่างซึ่งจะนำไปสู่การหาวัตถุเคลื่อนที่ และในส่วนการนับวัตถุจะมีการสร้างเงื่อนไขต่างๆ เพื่อหลีกเลี่ยงความผิดพลาดในการนับ อันได้แก่

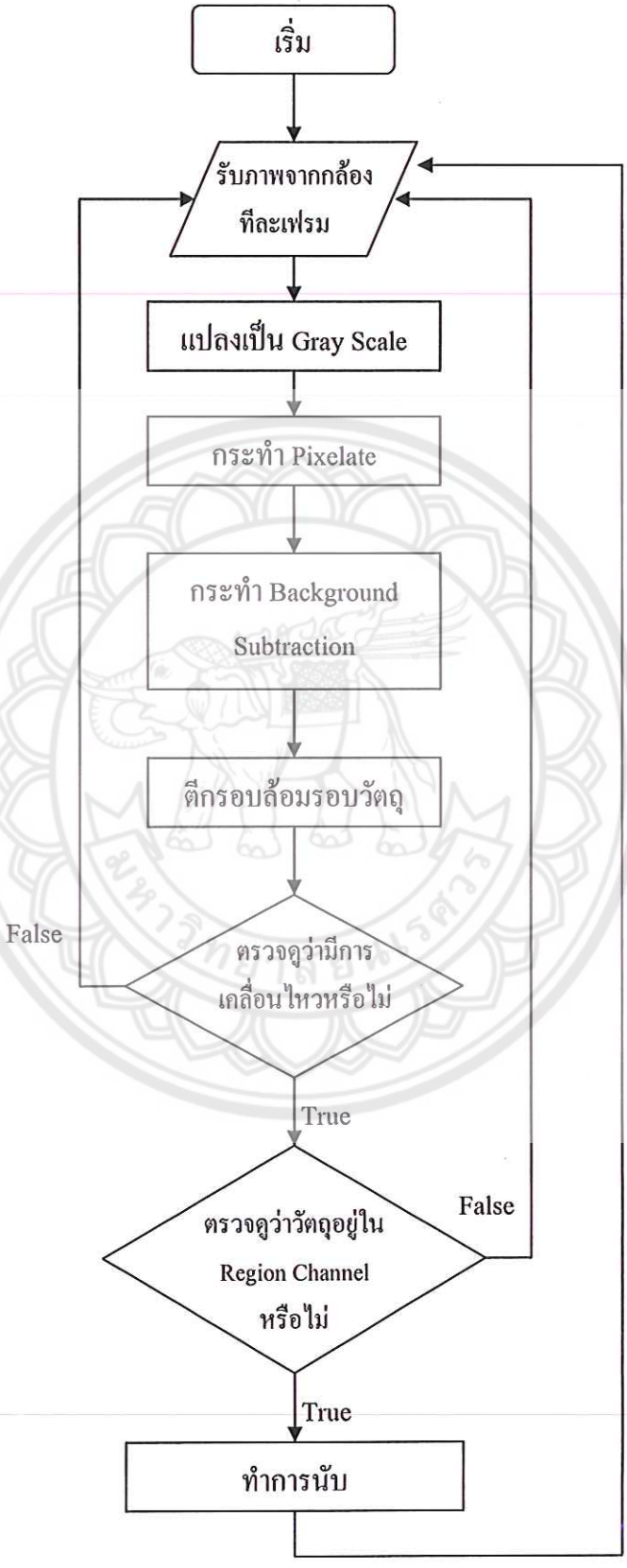
- การสร้างพื้นที่ในการนับ(region channel) เมื่อวัตถุเคลื่อนที่ผ่านจึงจะทำการนับ
- การสร้างเงื่อนไขกำหนดขนาดของวัตถุเพื่อที่จะหลีกเลี่ยงการนับวัตถุที่มีขนาดเล็กกว่ารถยนต์ ซึ่งเป็นสิ่งที่ไม่ต้องการนับ

นอกจากนี้ยังมีไลบารี่พิเศษอีกหนึ่งตัวที่ทำให้ C# สามารถทำงานร่วมกับข้อมูลวีดีโอในรูปแบบนามสกุล .AVI ได้คือ Tiger.Video.VFW โดยคลาส (Class) ไลบารี่และเมดธอดต่างๆ ในไลบารี่ดังแสดงในรูปที่3.6



รูปที่ 3.6 แสดงไลบารี่ของการประมวลผลของภาพและคลาสต่างๆ

ขั้นตอนการนับรถ



รูปที่ 3.7 แผนภาพแสดงขั้นตอนการนับรถ

จากรูปที่ 3.7 สามารถอธิบายแต่ละขั้นตอนพร้อมตัวอย่างภาพได้ดังนี้

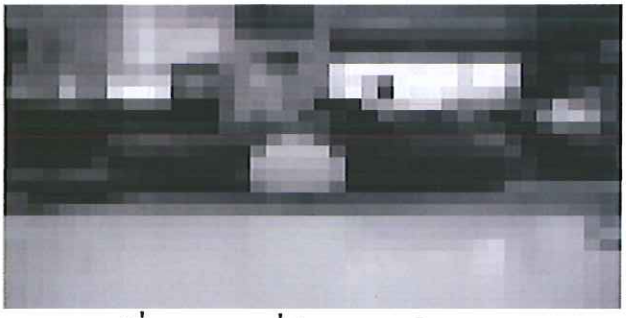
1. รับภาพจากกล้อง ซึ่งภาพที่รับมาจะเป็นภาพสี(RGB) ดังรูปที่ 3.8

รูปที่ 3.8 แสดงการจับวัตถุที่เคลื่อนที่

2. แปลงเป็นภาพระดับสีเทา ดังตัวอย่างในรูปที่ 3.9 เป็นการนำรูปที่ 3.8 ซึ่งเป็นภาพสีมาแปลงเป็นภาพระดับสีเทา

รูปที่ 3.9 แสดงภาพที่ถูกแปลงเป็นภาพระดับขาว-ดำ

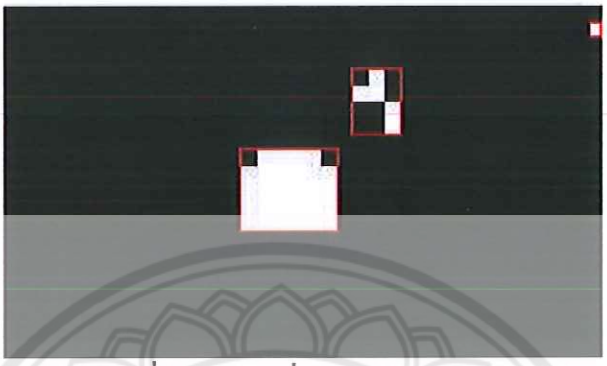
3. การทำพิกเซลเลท ซึ่งเป็นการทำภาพให้มีความหยาบเพื่อเก็บข้อมูลภาพได้น้อยลง ตัวอย่างดังแสดงในรูปที่ 3.10 เป็นการนำภาพในรูปที่ 3.9 มาทำพิกเซลเลท



รูปที่ 3.10 ภาพที่ผ่านการทำพิกเซลเลท

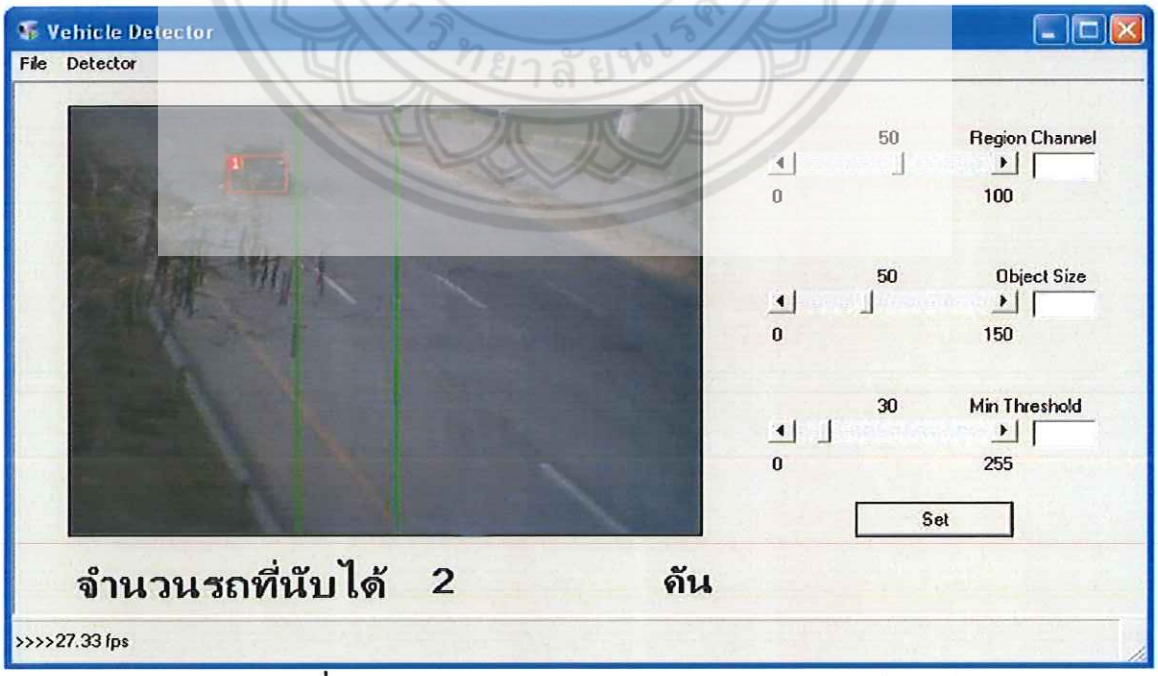
4. เมื่อนำภาพในเฟรมที่ 1 มาทำพิกเซลเลทแล้วภาพเฟรมที่ 1 จะถูกจัดเก็บให้เป็นพื้นหลัง เมื่อมีภาพในเฟรมที่ 2 เข้ามาจึงจะมีการลบพื้นหลังของภาพทิ้งตามทฤษฎีที่กล่าวไว้ในบทที่ 2

5. ตีกรอบล้อมรอบวัตถุ เมื่อทำการลบพื้นหลังของภาพทิ้งแล้วจะได้ค่าผลต่างออกมา ซึ่งค่าผลต่างนี้หมายถึงมีวัตถุเข้าในภาพ จะทำการตีกรอบล้อมรอบวัตถุ ดังแสดงตามรูปที่ 3.11



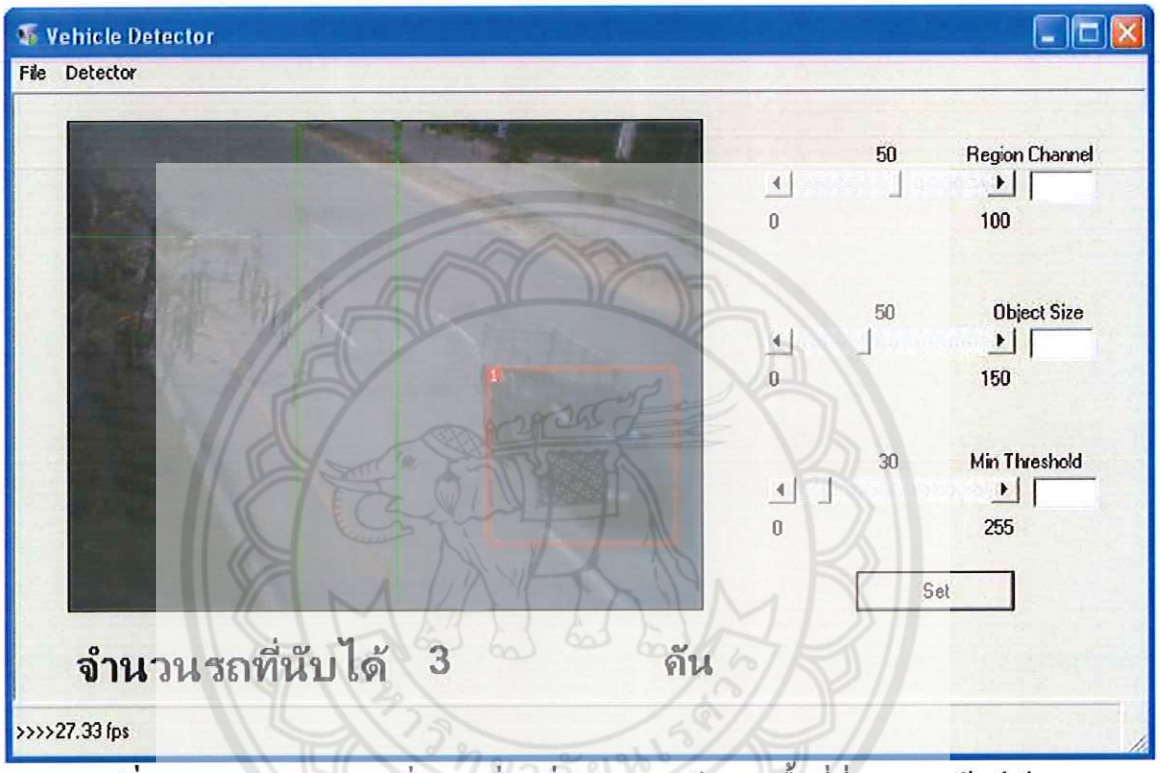
รูปที่ 3.11 ภาพที่ได้จากการตีกรอบ

6. ตรวจดูว่าวัตถุที่ได้ทำการตีกรอบไว้มีการเคลื่อนที่หรือไม่ ถ้าหากขนาดของวัตถุที่เคลื่อนที่มีขนาดใหญ่กว่าหรือเท่ากับขนาดที่กำหนดไว้จะมีการกำหนดสัญลักษณ์ตัวเลข(Label) ให้กับวัตถุนั้น ดังรูปที่ 3.12 ถ้าหากวัตถุนั้นมีขนาดเล็กกว่าที่กำหนดจะไม่มีการกำหนดสัญลักษณ์ตัวเลขให้ แต่ถ้าหากไม่มีการเคลื่อนที่จะกลับไปรับภาพในเฟรมต่อไป



รูปที่ 3.12 แสดงการตีกรอบวัตถุและกำหนดสัญลักษณ์

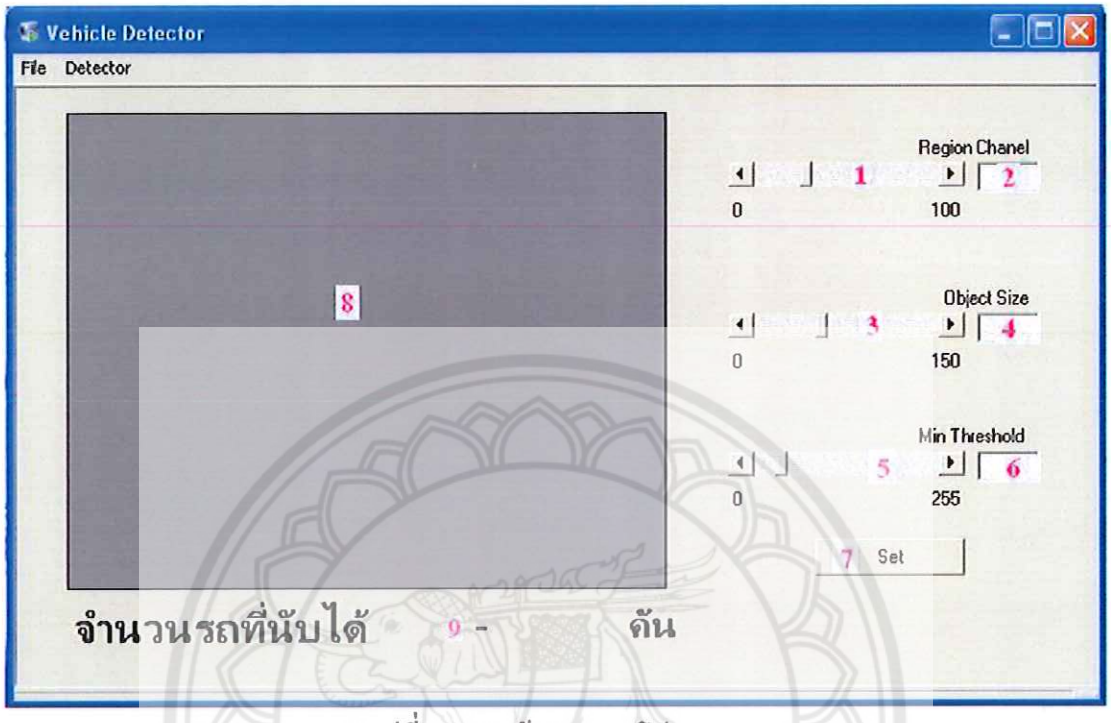
7. ถ้าหากวัตถุที่เคลื่อนที่ได้ถูกกำหนดสัญลักษณ์ตัวเลขให้แล้ว จะนำมาตรวจสอบว่าได้เคลื่อนที่ผ่านพื้นที่ที่กำหนดหรือไม่ ถ้าหากเคลื่อนที่ผ่านความกว้างของพื้นที่ที่กำหนด(เส้นสีเขียว 2 เส้นที่ปรากฏอยู่ดังรูปที่3.13) จึงจะทำการนับรถ ถ้าหากไม่ผ่านจะไม่ทำการนับ ดังรูปที่ 3.12 จำนวนรถที่นับได้เท่ากับ 2 คัน และเมื่อรถเคลื่อนที่ผ่านความกว้างของพื้นที่ที่กำหนดดังรูปที่ 3.13 จำนวนรถที่นับได้จึงเป็น 3 คัน



รูปที่ 3.13 แสดงการนับรถเมื่อรถเคลื่อนที่ผ่านความกว้างของพื้นที่ที่กำหนด(เส้นสีเขียว)

การใช้งานโปรแกรม

1. เริ่มต้นการทำงาน



รูปที่ 3.14 หน้าแรกของโปรแกรม

หมายเลข 1 แถบเลื่อนสำหรับเลือกขนาดของพื้นที่ในการนับ(Region Channel)

หมายเลข 2 ช่องสำหรับเลือกใส่ขนาดของพื้นที่ในการนับ

หมายเลข 3 แถบเลื่อนสำหรับเลือกขนาดของวัตถุ(Object Size)

หมายเลข 4 ช่องสำหรับเลือกใส่ขนาดของวัตถุ

หมายเลข 5 แถบเลื่อนสำหรับเลือกขนาดของค่าความแกว่ง(Min Threshold)

หมายเลข 6 ช่องสำหรับเลือกใส่ขนาดของค่าความแกว่ง

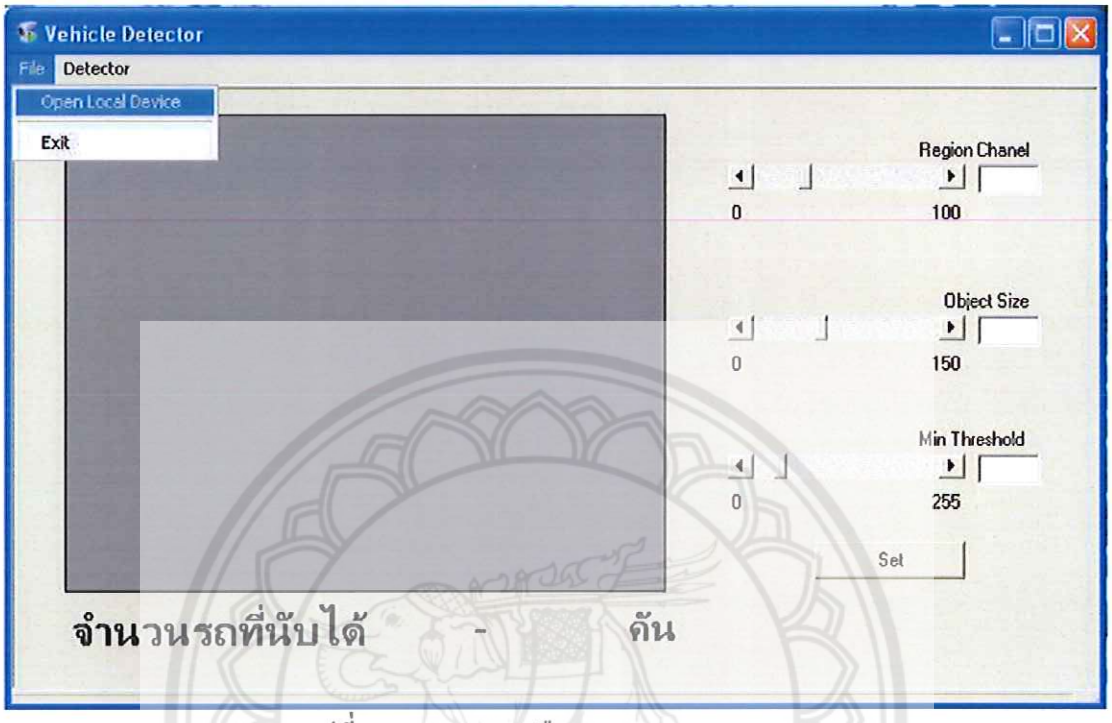
หมายเลข 7 เมื่อเลือกใส่ขนาดต่างๆ ตามหมายเลข 2, 4, 6 แล้วจะต้องทำการคลิกที่ปุ่ม Set ทุกครั้ง

หมายเลข 8 เป็นฉากสำหรับแสดงภาพการนับรถ

หมายเลข 9 เป็นช่องสำหรับแสดงจำนวนรถที่นับได้

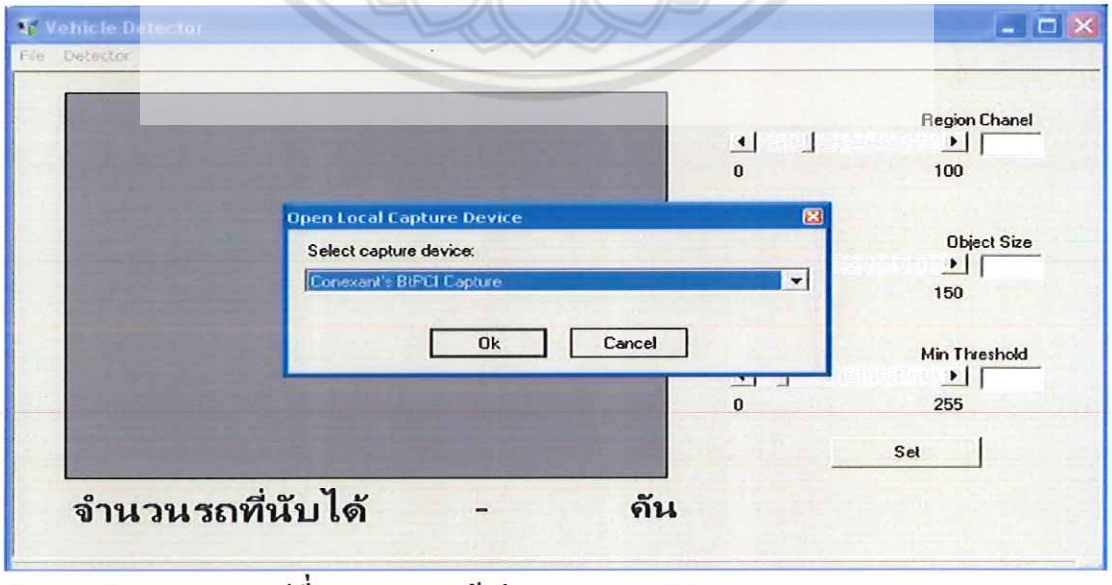
2. ขั้นตอนการใช้งาน

1. เมื่อเปิดโปรแกรมการใช้งานแล้ว ให้เลือกที่ File -> Open Local Device ดังรูปที่ 3.15



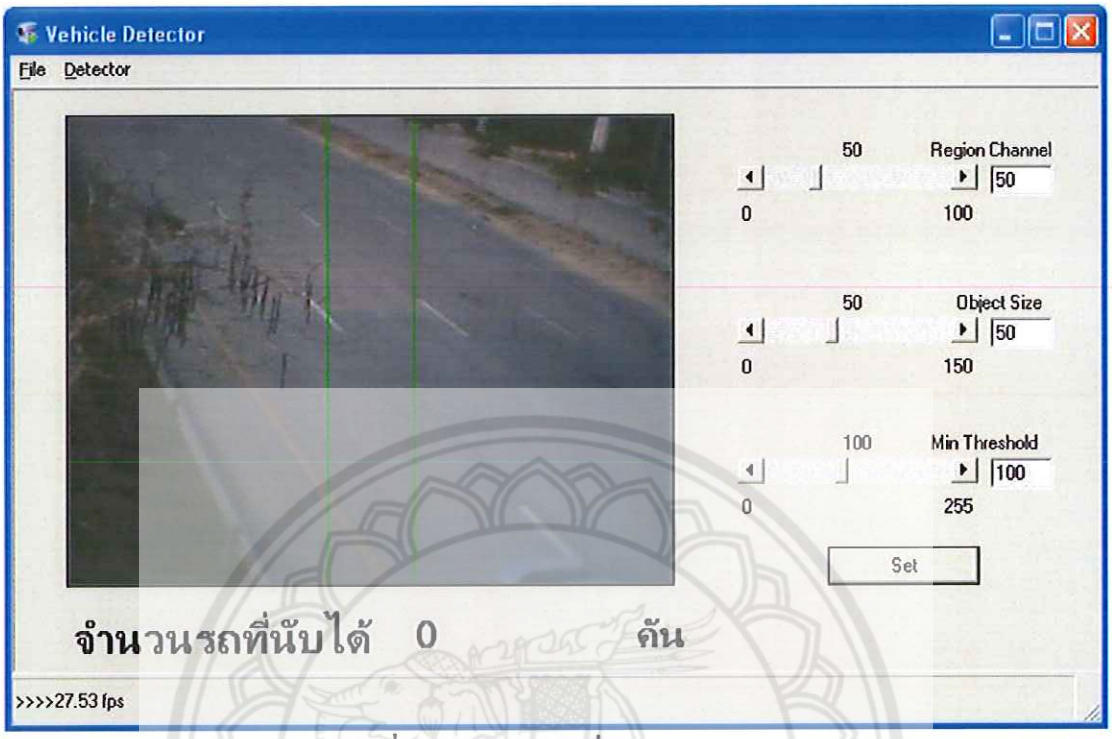
รูปที่ 3.15 แสดงการเลือก Open Local Device

2. เมื่อเลือก Open Local Device จะปรากฏหน้าต่าง Open Local Capture Device ดังรูปที่ 3.16 เพื่อให้เลือกกล้องที่ต้องการใช้งาน เมื่อเลือกกล้องที่จะใช้งานได้แล้วให้คลิกที่ปุ่ม OK



รูปที่ 3.16 แสดงหน้าต่าง Open Local Capture Device

3. เมื่อเลือกกล้องได้แล้ว จะมีภาพปรากฏขึ้นมาตรงบริเวณหมายเลข 8 ดังรูปที่ 3.17



รูปที่ 3.17 แสดงภาพที่ปรากฏบนฉาก

4. ปรับค่า พื้นที่ในการนับ, ขนาดของวัตถุ และค่าความแกว่ง โดยค่าต่างๆ มีความหมายดังนี้

พื้นที่ในการนับ – เป็นการกำหนดความกว้างของเส้นสีเขียว 2 เส้น คือเมื่อรถวิ่งผ่านเส้นสีเขียว 2 เส้นจึงจะมีการนับ แต่ถ้าหากมีรถอยู่บนจอภาพแต่ไม่ได้วิ่งผ่านเส้นสีเขียว 2 เส้นนี้ จะไม่มีการนับ ซึ่งค่าเริ่มต้นที่กำหนดไว้ในโปรแกรมคือ 60

ขนาดของวัตถุ – เป็นการกำหนดขนาดของวัตถุที่จะทำการตรวจจับว่าถ้าขนาดของวัตถุใหญ่กว่าขนาดที่กำหนดจึงจะทำการตรวจจับ โดยการทำกรอบสีแดงล้อมรอบและทำการนับ แต่ถ้าหากวัตถุเล็กกว่าที่ได้กำหนดไว้จะทำเพียงตีกรอบสีแดงล้อมรอบเท่านั้น แต่จะไม่ทำการนับซึ่งค่าเริ่มต้นที่กำหนดไว้ในโปรแกรมคือ 40

ค่าความแกว่ง – ถ้าหากวัตถุมีการเคลื่อนที่ที่ช้ามากๆ หรือมีค่าน้อยกว่าค่าความแกว่งที่กำหนดจะไม่ทำการนับวัตถุนั้น ซึ่งจะถือว่ารถคันนั้นไม่มีการเคลื่อนที่หรือหยุดนิ่ง ซึ่งค่าเริ่มต้นที่กำหนดไว้ในโปรแกรมคือ 30

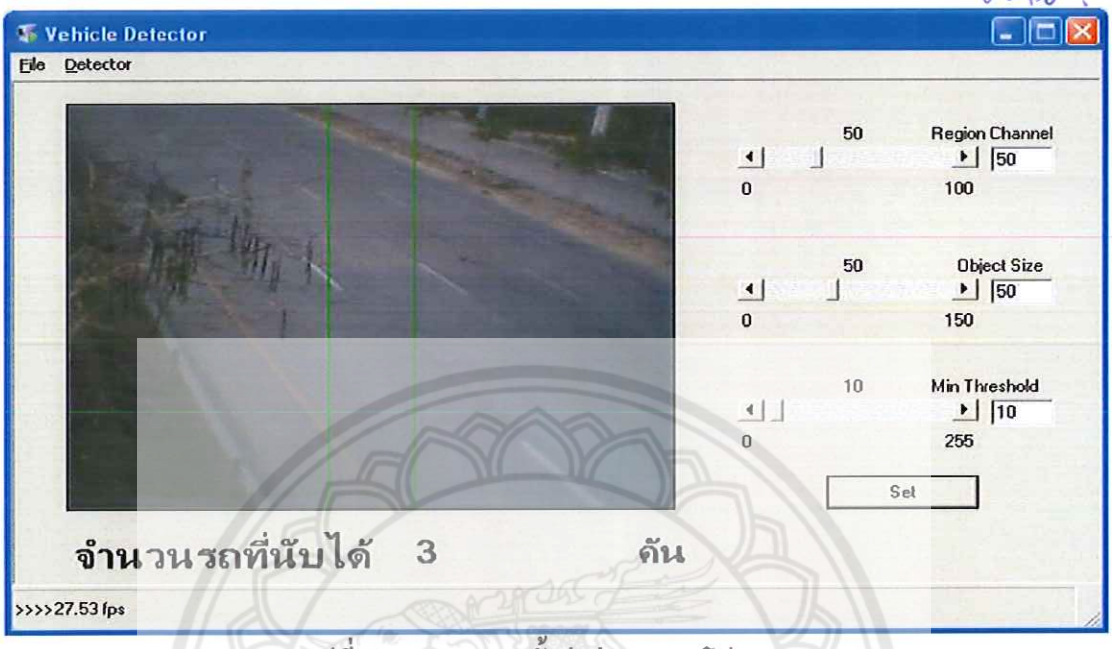
การตั้งค่าต่างๆ สามารถทำได้ 2 แบบ คือ

1. ตั้งค่าโดยการเลื่อนแถบตรงบริเวณหมายเลข 1, 3, 5
2. ตั้งค่าโดยการใส่ค่าตรงช่องบริเวณหมายเลข 2, 4, 6 แล้วทำการคลิกที่ปุ่ม Set

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

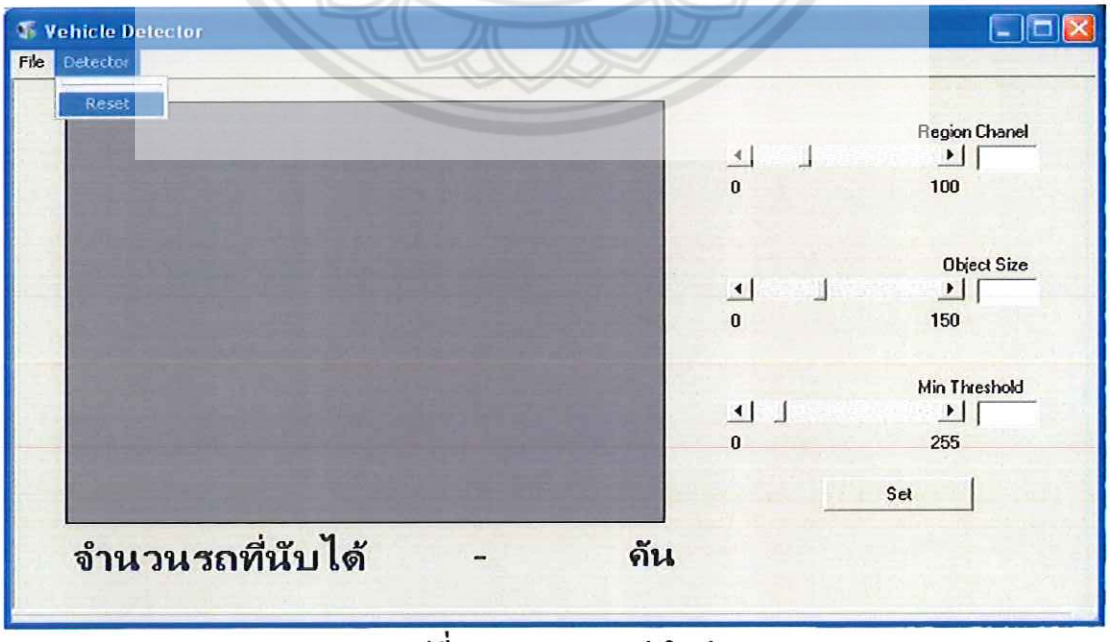
แต่ถ้าหากต้องการกลับไปยังค่าเริ่มต้นที่ตั้งไว้ในโปรแกรม ให้คลิกที่ปุ่ม Clear ดังรูปที่ 3.18 เป็นตัวอย่างการตั้งค่าต่างๆ ของโปรแกรม

เลขหมู่ 621.5127 ป5. ฉบับที่ 2548



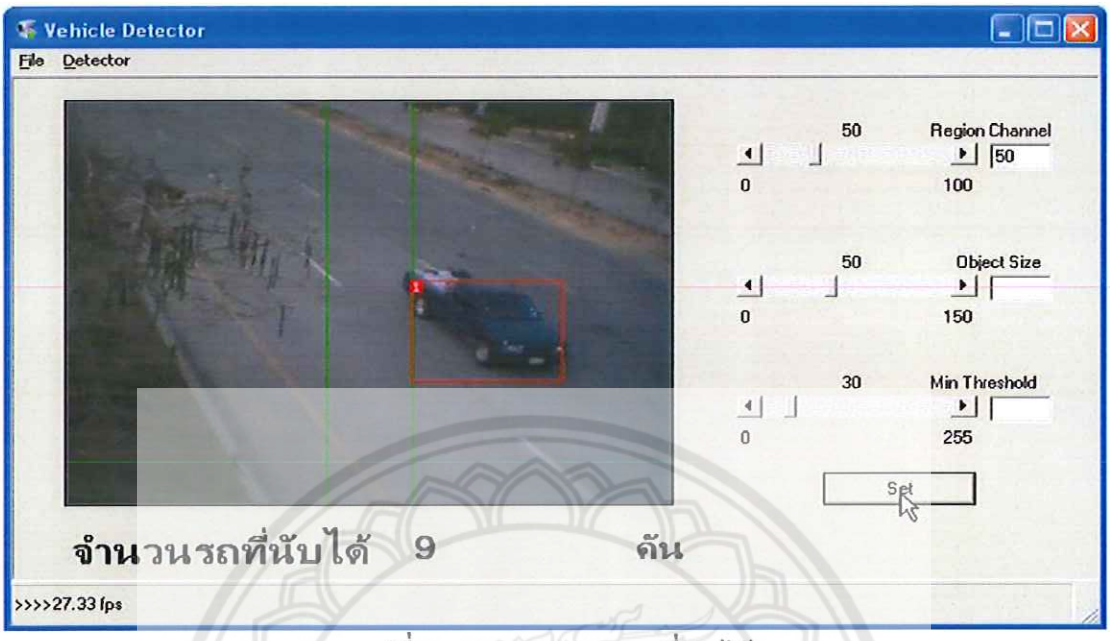
รูปที่ 3.18 แสดงการตั้งค่าต่างๆ ของโปรแกรม

5. เมื่อตั้งค่าต่าง ๆ เรียบร้อยแล้ว ให้เลือกที่ Detector -> Reset เพื่อเป็นการเริ่มนับใหม่ ดังรูปที่ 3.19 เนื่องจากก่อนการตั้งค่าอาจจะมีการนับวัตถุไปบ้างแล้วซึ่งอาจเป็นค่าที่ไม่ถูกต้อง ดังนั้นควรมีการ Reset ทุกครั้งที่ทำการตั้งค่าใหม่



รูปที่ 3.19 การ Reset ค่าใหม่

6. จำนวนรถที่นับได้จะแสดงบริเวณหมายเลข 10 ดังรูปที่ 3.20



รูปที่ 3.20 แสดงจำนวนรถที่นับได้

บทที่ 4
ผลการทดลอง

โปรแกรมสำหรับโครงงานนี้มีการตั้งค่าอยู่ 3 ตำแหน่งคือ

1. การตั้งค่าพื้นที่ในการนับ
2. การตั้งค่าขนาดของวัตถุ
3. การตั้งค่าความแกว่ง

ในการทดลองจะทำการเปรียบเทียบผลที่ได้จากการปรับค่าทั้ง 3 ตำแหน่ง

4.1 ผลการทดลอง

ในการทดลองจะทดลองการตั้งค่าหลายๆ ค่า เพื่อให้เห็นความแตกต่างของภาพที่ได้ และคำนวณออกมาเป็นเปอร์เซ็นต์ความผิดพลาดเพื่อทดลองว่าการนับรถ 20 คัน โปรแกรมจะนับพลาดกี่คันและคิดเป็นร้อยละเท่าใด ซึ่งมีสูตรการคำนวณแปอร์เซ็นต์ความผิดพลาดดังนี้คือ

$$\text{เปอร์เซ็นต์ความผิดพลาด} = \frac{\text{จำนวนรถที่นับผิดพลาด}}{\text{จำนวนรถทั้งหมด}} \times 100 \qquad (4.1)$$

เริ่มต้นด้วยการตั้งค่าให้พื้นที่ในการนับ, ขนาดของวัตถุ และค่าความแกว่ง มีค่าน้อยๆ แล้วเพิ่มค่ามากขึ้น

ตารางที่ 4.1 แสดงผลการทดลองจากการใช้กล้องเวบแคม

Region Channel	Object Size	Min Threshold	จำนวนรถที่นับได้ (คัน)	เปอร์เซ็นต์ความผิดพลาด (%)
10	20	10	28	40
10	20	30	26	30
10	20	100	27	35
10	50	10	12	40

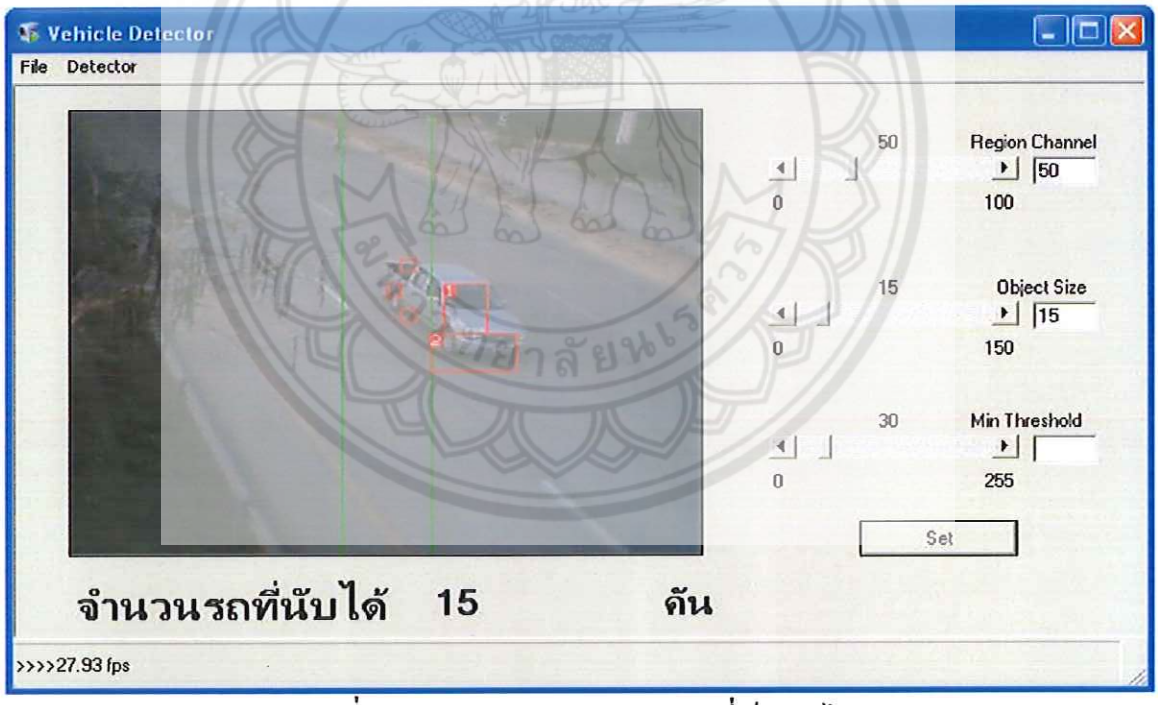
ตารางที่ 4.1 (ต่อ)

Region Channel	Object Size	Min Threshold	จำนวนรถที่นับได้ (คัน)	เปอร์เซ็นต์ความผิดพลาด (%)
10	50	30	14	30
10	50	100	10	50
10	90	10	6	70
10	90	30	4	80
10	90	100	3	85
50	20	10	27	35
50	20	30	26	30
50	20	100	30	50
50	50	10	19	5
50	50	30	21	5
50	50	100	18	10
50	90	10	5	75
50	90	30	4	80
50	90	100	7	65
100	20	10	40	100
100	20	30	34	70
100	20	100	36	80
100	50	10	32	60
100	50	30	29	45
100	50	100	27	35
100	90	10	9	55
100	90	30	8	60
100	90	100	6	70

4.2 ผลการวิเคราะห์

จากผลการทดลองข้างต้นจะเห็นความแตกต่างในการปรับค่าต่างๆ ดังนี้คือ

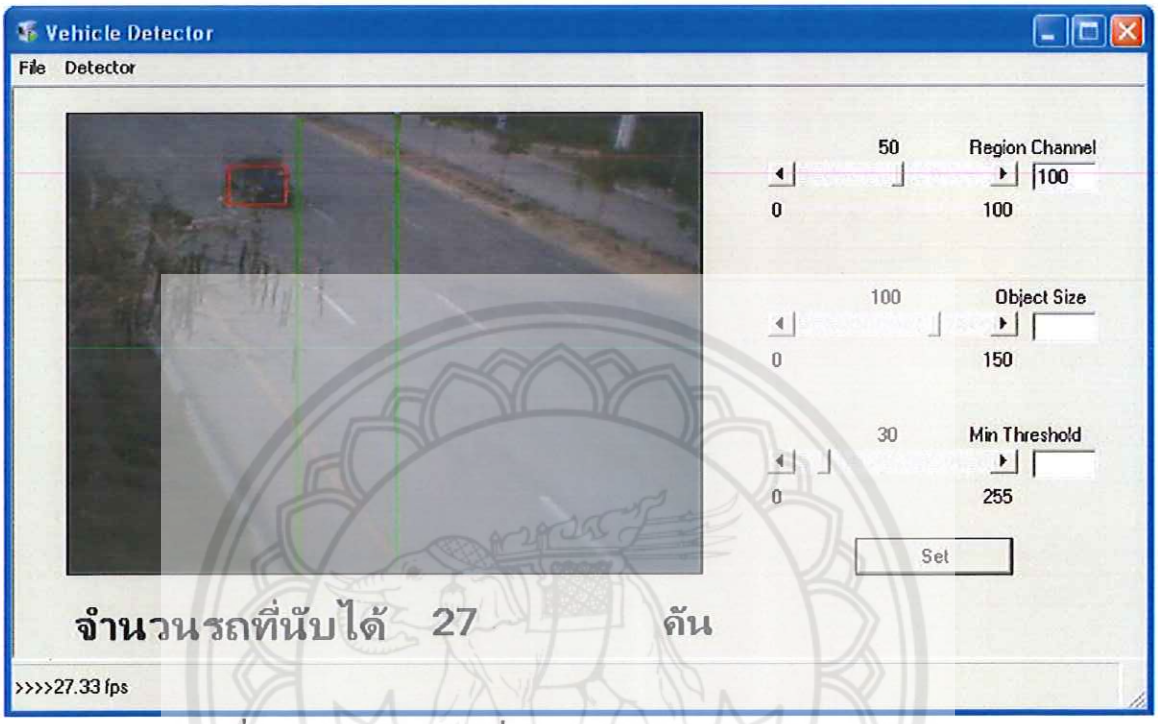
- ถ้าพื้นที่ในการนับมีค่าน้อยหรือกำหนดให้มีช่วงแคบมากเกินไป จะทำให้ไม่สามารถนับรถที่เคลื่อนที่ด้วยความเร็วสูงได้
- ถ้าพื้นที่ในการนับมีค่ามากหรือกำหนดให้มีช่วงกว้างมากเกินไป จะพบว่าเมื่อมีรถเคลื่อนที่เข้ามาอย่างช้าๆ จะทำให้เกิดการนับรถซ้ำ
- ถ้าขนาดของวัตถุมีค่าน้อยหรือการกำหนดขนาดวัตถุเล็กเกินไป จะทำให้นับรถจักรยานยนต์หรือรถจักรยานเข้าไปด้วย จึงทำให้จำนวนรถที่นับได้มีค่ามากกว่าจำนวนรถที่กำหนดไว้(20 คัน) และเมื่อมีรถยนต์ผ่านเข้ามาโปรแกรมจะตีกรอบได้หลายกรอบทำให้การนับรถยนต์ 1 คัน โปรแกรมจะนับได้เป็น 2-3 คัน เนื่องจากโปรแกรมสามารถกำหนดสัญลักษณ์ได้มากกว่า 1 สัญลักษณ์ ดังรูปที่ 4.1 กำหนดขนาดวัตถุเท่ากับ 15 ทำให้โปรแกรมตีกรอบรถยนต์ได้เป็นหลายๆ กรอบ และทำให้นับรถยนต์ 1 คัน เป็น 2 คัน



รูปที่ 4.1 แสดงการกำหนดขนาดวัตถุที่เล็กเกินไป

- ถ้าขนาดของวัตถุมีค่ามากหรือมีการกำหนดขนาดวัตถุใหญ่เกินไป จะพบว่าไม่สามารถนับรถที่มีขนาด 4 ล้อหรือเล็กกว่า 4 ล้อได้ จะนับได้เพียงแต่รถที่มีขนาด 6 ล้อขึ้นไป เช่น รถบรรทุก เป็นต้น ทำให้จำนวนรถที่นับได้มีปริมาณน้อยกว่าจำนวนรถที่กำหนดไว้ ดังรูปที่4.1 และ 4.2 เป็นตัวอย่างการกำหนดขนาดวัตถุที่ใหญ่เกินไป โดยรูปที่ 4.1 เป็นภาพขณะที่รถยังไม่เคลื่อนที่

ผ่านเส้นสีเขียวจึงยังไม่เกิดการนับและมีจำนวนรถที่นับได้ 27 คัน เมื่อรถเคลื่อนที่ผ่านเส้นสีเขียว ดังรูปที่ 4.2 จำนวนรถที่นับได้จะต้องเป็น 28 คัน แต่ในรูปที่ 4.2 ยังเป็น 27 คันเหมือนกับในรูปที่ 4.1แสดงว่ากำหนดขนาดวัตถุใหญ่เกินไปโปรแกรมจึงไม่ทำการนับรถ



รูปที่ 4.2 แสดงการนับรถเมื่อมีการกำหนดขนาดของวัตถุใหญ่เกินไป

รูปที่ 4.3 แสดงการนับรถเมื่อมีการกำหนดขนาดของวัตถุใหญ่เกินไป

- ถ้าค่าความแกว่งมีค่าน้อยเกินไป จะพบว่ามีการนับรถที่เคลื่อนที่ผ่านเข้ามาด้วยความเร็วที่ต่ำมากหรือเคลื่อนที่เข้ามาอย่างช้า ๆ จนเกือบจะหยุดรถเข้าไปด้วย หรืออีกความหมายหนึ่งก็คือจะมีการนับรถที่หยุดนิ่งเข้าไปด้วย
- ถ้าค่าความแกว่งมีค่ามากเกินไป จะพบว่าสามารถนับได้เฉพาะรถที่มีความเร็วสูงมากเท่านั้น ส่วนรถที่เคลื่อนที่ด้วยความเร็วธรรมดาหรือความเร็วต่ำ จะไม่มีการนับเนื่องจากมีความแกว่งไม่เพียงพอ

จากตารางผลการทดลองจะเห็นว่า ค่าที่เหมาะสมกับสภาพแวดล้อมและการตั้งกล้องคือ ค่าความกว้างของพื้นที่เท่ากับ 50, ค่าขนาดของวัตถุเท่ากับ 50 และค่าความแกว่งเท่ากับ 30 จากรูปที่ 4.4 แสดงการตั้งค่าที่เหมาะสมสำหรับสภาพแวดล้อมที่ทำการทดลอง

รูปที่ 4.4 แสดงการนับรถเมื่อกำหนดค่าแบบพอเหมาะ

บทที่5

สรุปผล

โครงงานนี้ทำการศึกษาและพัฒนาการใช้กล้องตรวจนับรถโดยวิธีการลบพื้นหลังภาพทิ้งแบบพื้นหลังเคลื่อนไหว ซึ่งเจาะจงลงไปในส่วนของการวิเคราะห์วัตถุเคลื่อนที่ เป็นการนำเอาทฤษฎีทางการประมวลผลภาพมาประยุกต์ใช้กับกล้องราคาถูกเพื่อนับจำนวนรถที่เคลื่อนที่ผ่านหน้ากล้อง เพื่อให้สามารถวิเคราะห์หรือรายงานผลสภาพการจราจร โดยทำการทดลองเพื่อเปรียบเทียบความแตกต่างในการปรับค่าต่างๆ คือ พื้นที่ในการนับ ขนาดของวัตถุและความแกว่ง

เมื่อได้ทำการทดลองทำให้ได้ทราบถึงข้อมูลและปัญหาบางประการ ทั้งในระหว่างการพัฒนาและการทดลองใช้งานหลายประการ โดยสามารถสรุปเป็นหัวข้อได้ดังนี้

5.1 สรุปผล

จากผลการทดลองซึ่งแสดงให้เห็นเปอร์เซ็นต์ความผิดพลาดที่เกิดจากการปรับค่าทั้ง 3 ค่า จะพบว่าควรปรับค่าพื้นที่ในการนับให้มีค่าอยู่ระหว่าง 40-60 ค่าขนาดของวัตถุควรปรับให้มีค่าอยู่ระหว่าง 40 – 70 และค่าความแกว่งควรปรับให้มีค่าอยู่ระหว่าง 30 – 50 จึงจะทำให้เกิดเปอร์เซ็นต์ความผิดพลาดน้อยลง แต่เปอร์เซ็นต์ความผิดพลาดที่เกิดขึ้นไม่ได้ขึ้นอยู่กับการปรับค่าต่าง ๆ เพียงอย่างเดียว แต่ขึ้นอยู่กับปริมาณของรถที่วิ่งผ่านหน้ากล้อง ถ้าหากจำนวนรถที่วิ่งผ่านหน้ากล้องมีปริมาณมากผลการนับรถที่ได้จะมีเปอร์เซ็นต์ความผิดพลาดมากกว่าสถานการณ์ที่มีรถวิ่งผ่านหน้ากล้องปริมาณน้อย และเมื่อมีวัตถุอื่น ๆ เช่น คน รถจักรยานยนต์ เป็นต้น ผ่านเข้ามาในกล้องจะไม่ทำการนับ และจากตารางที่ 4.1 พบว่า การนับรถเกิดความผิดพลาดร้อยละ 5

จึงสรุปได้ว่าโปรแกรมการตรวจนับรถด้วยวิธีการลบพื้นหลังของภาพ สามารถทำการตรวจนับรถได้ แต่ต้องทำการปรับแต่งค่าต่างๆ และปัจจัยหลายประการให้เหมาะสมเพื่อผลในการตรวจนับที่ถูกต้องแม่นยำที่สุด โดยปัจจัยที่เกี่ยวข้องกับความถูกต้องในการนับรถและแนวทางการแก้ไขปัญหาจะนำเสนอในหัวข้อ 5.2

5.2 แนวทางในการแก้ไขปัญหาและข้อเสนอแนะ

การทดลองกับสถานการณ์จริงมีปัจจัยที่เกี่ยวข้องกับความถูกต้องในการนับหลายประการ ซึ่งเกิดจากสภาพแวดล้อมของสถานการณ์นั้น ได้แก่

1. ความสว่างของสภาพแวดล้อมขณะจับภาพ ถ้าหากมีความสว่างไม่เพียงพอ เช่น ช่วงใกล้ ๆ ค่ำ จะทำให้การนับรถมีความผิดพลาด
2. ความเร็วของรถขณะเคลื่อนที่ผ่านหน้ากล้อง
3. ความละเอียดของกล้องที่ใช้ในการจับภาพ กล้องที่มีความละเอียดต่ำมีผลทำให้การนับรถมีความผิดพลาด
4. เฟรมเรตของกล้องวิดีโอซึ่งกล้องวิดีโอมีเฟรมเรตไม่คงที่จึงเป็นสาเหตุของความผิดพลาดในการนับจำนวนรถ
5. การปรับมุมกล้อง

ปัญหาที่เกี่ยวข้องกับแสงสว่าง เฟรมเรต และปัญหาความละเอียดของกล้องซึ่งทำให้การตรวจจับวัตถุเคลื่อนที่เกิดความผิดพลาดแทนที่จะตรวจจับรถยนต์ 1 คันได้เป็น 1 คัน อาจจะตรวจจับได้เป็น 2 คันหรือ 3 คัน ในส่วนนี้เป็นเพราะคุณภาพของกล้องที่ใช้ ถ้าใช้กล้องที่มีความละเอียดสูงและเฟรมเรตไม่ไวต่อการเปลี่ยนแปลงของแสงจะทำให้ผลการตรวจนับรถมีความผิดพลาดน้อยลงได้

การแก้ปัญหาเบื้องต้นทำได้โดยการปรับแต่งค่าต่างๆ ในโปรแกรม เช่น การปรับค่าขนาดของวัตถุที่ต้องการนับจะแก้ปัญหาในส่วนการนับเฉพาะส่วนที่เป็นรถยนต์ไม่นับคนหรือรถจักรยานยนต์ตลอดจนการจัดมุมกล้องให้เหมาะสมกับสภาพแวดล้อมของสถานที่ จะช่วยแก้ปัญหาในบางส่วนไปได้โดยใช้อุปกรณ์เดิม

เอกสารอ้างอิง

[1] David A.Forsyth and Jean Ponce. **Computer Vision A MODERN APPROACH.** New Jersey : Pearson Education International. 2003.

[2] Zhengyou Zhang. **"A Flexible New Technique for Camera Calibration"** [Online]. Available: http://research.microsoft.com/~zhang/Papers/TR98-71.pdf

[3] Chris Yang. **"Three-dimension Reconstruction from Stereo Extrinsic and Camera Intrinsic Properties"** [Online]. Available: http://member.rogers.com/yanger/project.html.

[4] Intel Corperation. **"OpenCV"** [Online]. Available: http://www.sourceforge.net/project/ - opencvlibrary/opencvman_old.pdf.

ภาคผนวก

คำอธิบายโปรแกรม

โปรแกรมในส่วนของการตรวจจับวัตถุเคลื่อนที่และตรวจนับ

```
{
        //เป็นการเรียกไลบารี่ต่าง ๆ ที่ต้องการใช้งาน
        using System;
        using System.Drawing;
        using System.Reflection;
        using AForge.Imaging;
        using AForge.Imaging.Filters;
        /// <summary>
        /// ส่วนของการกำหนดตัวแปรที่จะใช่ในคลาสต่าง ๆ
        /// </summary>
        public class MotionDetector : IMotionDetector
        {
                public static int regionCh_ = 60;
                public static int sizeOfObj_ = 40;
                public static byte minThresh_ = 30;
                public IFilter      grayscaleFilter = new GrayscaleBT709();
                public IFilter      pixellateFilter = new Pixellate();
                public Difference differenceFilter = new Difference();
                public IFilter thresholdFilter = new Threshold(minThresh_, 255);
                public MoveTowards moveTowardsFilter = new MoveTowards();
                public FiltersSequence    processingFilter1 = new FiltersSequence();
                public FiltersSequence    processingFilter2 = new FiltersSequence();
                public Bitmap    backgroundFrame;
                public int                    counter = 0;
```

```
//-----------------
private bool checkCount = false;
private bool check = false;
private int nPrevious = -1;
private int count = 0;

//-----------------
private Bitmap[] numbersBitmaps = new Bitmap[9];
// Constructor 1
public MotionDetector()
{
        processingFilter1.Add(grayscaleFilter);
        processingFilter1.Add(pixellateFilter);
        processingFilter2.Add(differenceFilter);
        processingFilter2.Add(thresholdFilter);

        Assembly assembly = this.GetType().Assembly;

        for (int i = 1; i <= 9; i++)
        {
                numbersBitmaps[i - 1] = new
                Bitmap(assembly.GetManifestResourceStream(
                string.Format("motion.Resources.{0}.gif", i)));
        }
}
```

```
// Constructor 2
public MotionDetector( int a_, int b_, byte c_)
{
        this.setRegion(a_);
        this.setObjSize(b_);
        this.setThresh(c_);
// สร้างฟิลเตอร์เพื่อจะนำเมดธอตเข้ามาในฟิลเตอร์
        processingFilter1.Add(grayscaleFilter);
        processingFilter1.Add(pixellateFilter);
        processingFilter2.Add(differenceFilter);
        processingFilter2.Add(thresholdFilter);

// จองพื้นที่ให้กับสัญลักษณ์ตัวเลขของกรอบวัตถุที่เคลื่อนที่จำนวน 9 ตัวที่จะนำไปทำ
ลาเบลให้กับวัตถุเคลื่อนที่ที่ตรวจจับได้
        Assembly assembly = this.GetType().Assembly;

        for (int i = 1; i <= 9; i++)
        {
                numbersBitmaps[i - 1] = new
        Bitmap(assembly.GetManifestResourceStream(
        string.Format("motion.Resources.{0}.gif", i)));
        }
}
```

```
//ทำการส่วนของการตรวจจับวัตถุไปจุดเริ่มต้นคือถ้าแบคกราวด์ไมเป็นศูนย์เราจะตั้งค่าให้มันเป็นศูนย์
public void Reset()
{
        if (backgroundFrame != null)
        {
                backgroundFrame.Dispose();
                backgroundFrame = null;
        }
        counter = 0;
}
//เป็นการสร้างฟังก์ชั่นในการนำค่าที่กำหนดผ่านหน้าโปรแกรมเข้ามาทำงาน
/// <summary>
// ---------------------------------
/// </summary>
/// <returns></returns>
public int getCount()
{
        return count;
}
//---------------------------------
/* Accesser and Mutator*/
public int getRegion()
{
        return regionCh_;
}
//---------------------------------
public int getObjSize()
{
        return sizeOfObj_;
}
```

```
//-----------------------------------
public byte getThresh()
{
        return minThresh_;
}
//-----------------------------------
public void setRegion(int regionCh)
{
        regionCh_ = regionCh;
}
//-----------------------------------
public void setObjSize(int sizeOfObj)
{

        sizeOfObj_ = sizeOfObj;
}
//-----------------------------------
public void setThresh(byte minThresh)
{
        minThresh_ = minThresh;
}
//-----------------------------------
//---------------
public void resetCount( )
{
        count = 0;
}
//--------------
```

```
// เริ่มต้นดำเนินการกับเฟรมแรกที่เข้ามา
public void ProcessFrame(ref Bitmap image)
{
        if (backgroundFrame == null)
        {
                // ถ้ายังไม่มีแบคกราวน์ให้นำเฟรมแรกที่เข้ามาเป็นแบคกราวน์
                backgroundFrame = processingFilter1.Apply(image);

                // จะมีกระบวนการนี้เกิดขึ้นเพียงรอบแรกเท่านั้น
                return;
        }
        Bitmap tmpImage;
        // นำเฟรมที่เข้ามานำเนินการกับฟิลเตอร์1ที่ได้สร้างเตรียมไว้
        tmpImage = processingFilter1.Apply(image);

        if (++counter == 2)
        {
                counter = 0;

        // นำภาพที่นำเข้ามาใหม่ไปเป็น background
                moveTowardsFilter.OverlayImage = tmpImage;
                Bitmap tmp = moveTowardsFilter.Apply(backgroundFrame);

        // เอาแบคกราวน์ตัวเก่าทิ้ง
                backgroundFrame.Dispose();
                backgroundFrame = tmp;
        }
```

```
// เตรียมแบคกราวน์สำหรับทำการลบ
differenceFilter.OverlayImage = backgroundFrame;


//นำฟิลเตอร์ตัวที่สองเข้ามาดำเนินการกับเฟรมเพื่อกระทำการลบหาค่าความแตกต่างของเฟรม
Bitmap tmpImage2 = processingFilter2.Apply(tmpImage);
tmpImage.Dispose();


// เป็นการค้นหาวัตถุส่วนต่างที่ได้จากการลบกัน
Rectangle[] rects = BlobCounter.GetObjectRectangles(tmpImage2);
tmpImage2.Dispose();


// สร้างเส้นรีเจียนที่จะใช้เป็นขอบเขตในการนับเมื่อวัตถุเคลื่อนที่ผ่าน
Graphics g = Graphics.FromImage(image);
Pen penRegion = new Pen(Color.Green, 1);
g.DrawLine(penRegion , ((image.Width/2) - (regionCh_/2))
,0,((image.Width/2) - (regionCh_/2)),(image.Height));
g.DrawLine(penRegion , ((image.Width/2) + (regionCh_/2))
,0,((image.Width/2) + (regionCh_/2)),(image.Height));


using (Pen pen = new Pen(Color.Red, 1))
{
        int n = 0;

        // วาดกรอบสีแดงล้อมรอบวัตถุเคลื่อนที่ที่ตรวจจับได้
        foreach (Rectangle rc in rects)
        {
                g.DrawRectangle(pen, rc);
```

```
// กำหนดขนาด ขอบเขต ของวัตถุ ถ้า มากกว่า จึงเอาตัวเลขที่เตรียมไว้มาใส่
เพื่อเป็นการกำหนดลาเบล
if ((n < 10) && (rc.Width > sizeOfObj_) && (rc.Height > sizeOfObj_))
                {
// ขนาดของตัวอักษรบอก วัตถุอันดับที่ ต่างๆ
        g.DrawImage(numbersBitmaps[n], rc.Left, rc.Top, 7, 9);
                        n++;
                        //----------
                        if((rc.X > (image.Width/2) - (regionCh_/2))
                        &&(rc.X<(image.Width/2) + (regionCh_/2))
                        &&(checkCount == false))
                        {
                                check = true;
                        }
                        else
                        {
                                checkCount = false;
                                nPrevious =-1;
                        }
                        if((check == true)&&(nPrevious != n))
                        {
                                count++;
                                check = false;
                                checkCount = true;
                                nPrevious = n;
                        }
                        //----------

                }
        }
```

```
                }

                g.Dispose();

            }
        }
    }
```

ประวัติผู้เขียนโครงงาน

ชื่อ นางสาวจิตราภรณ์ สุปิน

ภูมิลำเนา 7/1 ตรอกบ้านสันคือ ถ.ประสาทวิถี อ.แม่สอด จ.ตาก 63110

ประวัติการศึกษา

- จบมัธยมศึกษาจาก โรงเรียนสรรพวิทยาคม
- ปัจจุบัน กำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : janis_m@hotmail.com

ชื่อ นาย นุกานต์ อุดมบัว

ภูมิลำเนา 47 ม.2 ต.บ้านบัว อ.เกษตรสมบูรณ์ จ.ชัยภูมิ 36120

ประวัติการศึกษา

- จบมัธยมศึกษาจาก โรงเรียนเกษตรสมบูรณ์วิทยาคม
- ปัจจุบัน กำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : toto_udombua@hotmail.com

ชื่อ นายปรัชญา เทียมลม

ภูมิลำเนา 383 หมู่2 ต.โนนสะอาด อ.โนนสะอาด จ.อุดรธานี 41240

ประวัติการศึกษา

- จบมัธยมศึกษาจาก โรงเรียนขอนแก่นวิทยายน
- ปัจจุบัน กำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : pratchaya_ss@hotmail.com