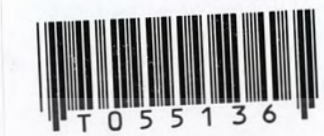
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบตรวจสอบสภาพการจราจร

TRAFFIC CHECKING SYSTEM



โดย

นายเสฐียรพงษ์ เบญจมานุกูล

นายทนงศักดิ์ มิ่งวัฒนชาติ

อาจารย์ที่ปรึกษา

อาจารย์สมเกียรติ วังศิริพิทักษ์

ดร.สมศักดิ์ วลัยรัชต์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขหมู่.....
เลขทะเบียน 55136
วัน,เดือน,ปี - 8 เม.ย. 2548

b.....
i.....

ปริญญานิพนธ์ ปีการศึกษา 2546

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบตรวจสอบการจราจร

TRAFFIC CHECKING SYSTEM

คณะผู้จัดทำ นายทนงศักดิ์ มิ่งวัฒนชาติ รหัส 44015326

นายเสฐียรพงษ์ เบญจมานุกูล รหัส 44015362

.....อาจารย์ที่ปรึกษา

(อ.สมเกียรติ วังศิริพิทักษ์)

.....อาจารย์ที่ปรึกษา

(ดร.สมศักดิ์ วลัยรัชต์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบตรวจสอบสภาพการจราจร

นายทนงศักดิ์ มิ่งวัฒนชาติ 44015326
นายเสฐียรพงษ์ เบญจมานุกูล 44015362
อ.สมเกียรติ วังศิริพิทักษ์ อาจารย์ที่ปรึกษา
ดร.สมศักดิ์ วลัยรัชต์ อาจารย์ที่ปรึกษาร่วม
ปีการศึกษา 2546

บทคัดย่อ

เนื่องจากที่ในปัจจุบันปัญหาการจราจรติดขัดมีแนวโน้มที่จะทวีความรุนแรงเพิ่มมากขึ้นเรื่อยๆ โดยเฉพาะในเมืองใหญ่ๆ ซึ่งเกิดจากการเพิ่มขึ้นของประชากร กลุ่มผู้ดำเนินโครงงานได้เล็งเห็นถึงปัญหาข้อนี้จึงได้พยายามสร้างระบบตรวจสอบสภาพการจราจร(Traffic Checking System) ให้สามารถนับปริมาณรถบนท้องถนน ตรวจสอบความเร็วบนท้องถนน และหาความหนาแน่นบนท้องถนนเพื่อที่จะได้นำข้อมูลเหล่านี้ไปใช้ในการแก้ไขปัญหาจราจรต่อไป การทำงานของระบบจะทำการรับภาพเข้ามาผ่านขบวนการประมวลผลภาพ และนำมาทำการวิเคราะห์หาจำนวนรถ ความเร็วของรถและความหนานแน่นของรถ โดยที่ระบบตรวจสอบสภาพการจราจรนี้สามารถทำงานได้ 2 แบบ คือทำงานในส่วนของเรียลไทม์ (Real-time) รับภาพจากกล้องวิดีโอเข้ามาประมวลในขณะนั้นเลยและการทำงานในส่วนของออฟไลน์ (Offline) ซึ่งจะนำภาพที่มีการบันทึกไว้ก่อนมาทำการประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRAFFIC CHECKING SYSTEM

Mr. Thanongsak Mingwattanachat

Mr. Sathianpong Benjamanukul

Mr. Somkiat Wangsiripitak Advisor

Dr. Somsak Walairacht Co-Advisor

Academic Year 2003

ABSTRACT

Since traffic jam problems, especially in the big city, have become more and more serious, we decided to develop the Traffic Checking System, which is capable of counting the number of cars on the road, measuring the speed of each car, and calculating the density of cars on that road.

Our system can function in 2 modes: off-line mode and real-time mode. In the former mode, the input of the system is an AVI file, whereas in the latter mode, the system gets the input images directly from a video camera connected via USB port. In both cases, our system can function pretty well.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยคำแนะนำ คำปรึกษาและคอยดูแลจากหลายๆฝ่ายด้วยกัน โดยเฉพาะอาจารย์ที่ปรึกษาทั้งสองท่านที่ให้โอกาสให้ข้าพเจ้าได้ทำปริญญานิพนธ์ฉบับนี้ คอยให้คำแนะนำและความช่วยเหลือเสมอมา คือ อาจารย์สมเกียรติ วังศิริพิทักษ์และอาจารย์สมศักดิ์ วลัยรัชต์ ซึ่งต้องขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ที่ได้จัดเตรียมสิ่งอำนวยความสะดวก เพื่อให้การวิจัยและพัฒนาโปรแกรมเป็นไปได้ด้วยความสะดวกและรวดเร็ว รวมทั้งยังมีห้องมัลติมิเดียให้บริการ สำหรับการค้นคว้าหาความรู้ต่างๆ ซึ่งท้ายที่สุดแล้วก็ประกอบกันเป็นส่วนหนึ่งของโครงงานนี้

และสุดท้ายต้องขอขอบคุณบุคคลที่สำคัญที่สุดในชีวิตที่ทำให้ข้าพเจ้ามีวันนี้ นั่นคือ บิดา มารดา และบุคคลในครอบครัว อันเป็นที่เคารพรัก ซึ่งได้เลี้ยงดู คอยสั่งสอนข้าพเจ้ามาเป็นอย่างดี พร้อมให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ ความรักเสมอมา ข้าพเจ้าขอกราบขอบพระคุณมา ณ ที่นี้ด้วย

นายทนงศักดิ์ มิ่งวัฒนชาติ

นายเสฐียรพงษ์ เบญจมานุกูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ขอบเขตของโครงงาน	1
1.4 วิธีการดำเนินงาน	2
บทที่ 2 ทฤษฎีการประมวลผลภาพ	3
2.1 Color model	3
2.1.1 The RGB color model	3
2.1.2 The YIQ color model	4
2.2 ทฤษฎีการหาขอบของวัตถุ (Edge Detection)	4
2.2.1 Gradient Method	5
2.2.2 Laplacian Method	6
2.3 ทฤษฎีการหาวัตถุที่กำลังเคลื่อนที่ (Object Detection)	7
2.3.1 Inter-frames difference	8
2.3.2 Sobel-Edge detection	10
2.3.3 Moving-Edge closure (MEC)	12
2.3.4 การทำ Filling	13
2.3.5 Labeling	14
2.3.6 การตีกรอบส่วนที่เป็นรถ (Boundary)	14
2.4 ทฤษฎีและหลักการที่ใช้ในการนับรถ	16
2.5 หฤษฎีและหลักการที่ใช้ในการวัดความเร็วรถ	17
บทที่ 3 รูปแบบของ ไฟล์ เอวีไอ (Format files AVI)	19
3.1 ส่วนหัวของไฟล์ เอวีไอ (AVI Header Format)	19
3.2 ส่วนหัวของเฟรม เอวีไอ (Frame AVI Header Format)	20
บทที่ 4 ทฤษฎีและหลักการคอมโพเน็นท์กับไดเร็กซ์เอ็กซ์	23
4.1 คอมโพเน็นท์ออบเจ็กท์โมเดล (Component Object Model – COM)	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้าที่
4.2 ไดเร็กซ์เอ็กซ์ออบเจ็กท์ กับ คอมโพเน็นท์ออบเจ็กท์	24
4.3 อินเตอร์เฟสของ COM ออบเจ็กท์	24
4.4 ไดเร็กซ์เอ็กซ์ 9 (DirectX 9)	25
4.4.1 ความรู้เบื้องต้นเกี่ยวกับไดเร็กซ์เอ็กซ์	25
4.4.2 ชุดพัฒนาซอฟต์แวร์สำหรับไดเร็กซ์เอ็กซ์	26
(DirectX software Development Kit: DirectX SDK)	26
4.4.3 คอมโพเน็นท์ในไดเร็กซ์เอ็กซ์	26
4.4.3.1 DirectX Graphics	26
4.4.3.2 DirectX Audio	26
4.4.3.3 DirectInput	26
4.4.3.4 DirectPlay	26
4.4.3.5 DirectShow	27
4.4.3.6 DirectSetup	27
บทที่ 5 โครงสร้างและการออกแบบ	28
5.1 ภาพรวมของระบบ	28
5.2 คุณสมบัติของโปรแกรมระบบตรวจสอบสภาพการจราจร	29
5.3 ระบบคอมพิวเตอร์	29
5.4 เครื่องมือที่ใช้ในการพัฒนา	30
5.4.1 ไมโครซอฟท์ วิชวล ซีพลัสพลัส ดอท เน็ท (Microsoft Visual C++.Net)	30
5.4.2 ไมโครซอฟท์ ไดเร็กซ์เอ็กซ์ 9 เอสดีเค (Microsoft DirectX9 SDK)	30
5.5 การออกแบบโปรแกรมระบบตรวจสอบสภาพการจราจร	31
5.5.1 ส่วนที่ทำการรับอินพุตระบบ	31
5.5.2 ส่วนของการควบคุมการทำงานทั้งหมดของระบบ	32
5.5.3 ส่วนของการ Recognize ภาพ	33
5.5.4 การออกแบบหน้าจอ	35
บทที่ 6 ผลการทดลอง/การทดสอบ	37
6.1 การหาจำนวนรถจากระบบตรวจสอบสภาพการจราจร	37
6.2 การหาความเร็วรถที่ได้จากระบบตรวจสอบสภาพการจราจร	38
6.3 สรุปผลการทดลอง	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้าที่
บทที่ 7 บทวิจารณ์และสรุป	41
7.1 บทวิจารณ์	41
7.2 ข้อสรุปและข้อเสนอแนะ	43
7.3 แนวทางในการพัฒนาโครงการ	43
ภาคผนวก คู่มือการติดตั้งและการใช้งานโปรแกรม	44
บรรณานุกรม	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้าที่
รูปที่ 2-1 แสดง RGB Color Model	3
รูปที่ 2-2 การหา Edge detection โดยการทำอนุพันธ์	4
รูปที่ 2-3 (a) Mask ที่ใช้ในการคำนวณ (b) Mask ที่ใช้ในการคำนวณ	5
รูปที่ 2-4 แสดงหลักการตรวจจับขอบโดยใช้ Gradient operator	6
รูปที่ 2-5 แสดงการทำงานของ Background algorithm	7
รูปที่ 2-6 แสดงลำดับการทำงานของ Inter-frames algorithm	8
รูปที่ 2-7 แสดงวิธีการทำ Double-difference operator	9
รูปที่ 2-8 (a) ภาพที่ลำดับ i-1	9
รูปที่ 2-8 (b) ภาพที่ลำดับ I	9
รูปที่ 2-8 (c) ภาพที่ลำดับ i+1	10
รูปที่ 2-8 (d) ภาพที่ได้จากการทำ Inter-frames difference	10
รูปที่ 2-9 (a) ภาพก่อนที่จะทำ Sobel-Edge Detection	11
รูปที่ 2-9 (b) ภาพที่ได้จากการทำ Sobel-Edge Detection	11
รูปที่ 2-10 Moving-Edge closure	12
รูปที่ 2-11 แสดงวิธีการทำ Filling	13
รูปที่ 2-12 หลังจากทำ Filling	13
รูปที่ 2-13 แสดงส่วนที่มีพิกเซลติดกัน	14
รูปที่ 2-14 แสดงส่วนที่มีพิกเซลติดกัน เป็นรถคันที่ 1 และ 2	14
รูปที่ 2-15 (a) ภาพที่ผ่านการทำ Boundary	15
รูปที่ 2-15 (b) ภาพต้นแบบ	15
รูปที่ 2-15 (c) นำภาพ (a) กับ (b) มารวมกัน จะได้ภาพที่มีการตีกรอบส่วนที่เป็นรถ	15
รูปที่ 2-16 วิธีการกำหนดเส้นตรวจนับรถ (Counter line)	16
รูปที่ 2-17 การใช้งานจริงของเส้นตรวจนับรถ	16
รูปที่ 2-18 วิธีการกำหนดเส้นตรวจจับความเร็ว	17
รูปที่ 2-19 การใช้งานจริงของเส้นตรวจจับความเร็ว	17
รูปที่ 3-1 แสดงรูปแบบของเฟรม เอวีไอ	20
รูปที่ 4-1 แสดงความสัมพันธ์ของคอมโพเน็นท์, อินเตอร์เฟสและฟังก์ชั่น	23
รูปที่ 4-2 แสดงการติดต่อกันระหว่าง COM ออบเจ็กท์ กับ ไดเร็กซ์เอ็กซ์ ออบเจ็กท์	24
รูปที่ 5-1 แสดงภาพรวมของระบบ	28
รูปที่ 5-2 แสดง Flow chart ส่วนที่ทำการรับอินพุตระบบ	31
รูปที่ 5-3 แสดง Flow chart ส่วนของการควบคุมการทำงานทั้งหมดของระบบ	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

	หน้าที่
รูปที่ 5-4 แสดง Flow chart ส่วนของการ Recognize ภาพ	33
รูปที่ 5-5 แสดงหน้าจอการใช้งาน	35
รูปที่ 6-1 มุมมองที่ทำการทดลอง และบริเวณที่กำหนดเส้นตรวจจับทั้ง 2 ตำแหน่ง	37
รูปที่ 6-2 แสดงจำนวนรถ จำนวนรถจริงกับจำนวนที่ได้จากโปรแกรม	38
รูปที่ 6-3 แสดงเปอร์เซ็นต์ความผิดพลาดการวัดความเร็วรถด้วย โปรแกรมโดยตั้งค่า frame rate เท่ากับ 3 frame/sec	39
รูปที่ 6-4 แสดงเปอร์เซ็นต์ความผิดพลาดการวัดความเร็วรถด้วย โปรแกรมโดยตั้งค่า frame rate เท่ากับ 5 frame/sec	39
รูปที่ 6-5 แสดงเปอร์เซ็นต์ความผิดพลาดการวัดความเร็วรถด้วย โปรแกรมโดยตั้งค่า frame rate เท่ากับ 8 frame/sec	40
รูปที่ 7-1 ความผิดพลาดที่เกิดจากการสะท้อนของแสง	41
รูปที่ 7-2 ความผิดพลาดที่เกิดจากรถที่มีขนาดใหญ่	42
รูปที่ 7-3 ความผิดพลาดที่เกิดจากช่วงเวลาในการแคปเจอร์	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้าที่
ตารางที่ 3- 1 รูปแบบส่วนหัวของไฟล์ เอวีไอ	19
ตารางที่ 3- 2 รูปแบบส่วนหัวของไฟล์ เอวีไอ	20
ตารางที่ 5-1 แสดงระบบคอมพิวเตอร์ที่โปรแกรมต้องการ	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันปัญหาใหญ่ที่พบตามมาจากปัญหาการเพิ่มขึ้นของประชากรในตัวเมืองใหญ่ๆ ก็คือ ปัญหาการจราจร อันเป็นผลเนื่องมาจากปริมาณของผู้ใช้รถบนท้องถนนที่เพิ่มมากขึ้นและมีแนวโน้มที่จะทวีความรุนแรงเพิ่มขึ้น ดังนั้นโครงงานระบบตรวจสอบการจราจรจึงจัดทำขึ้นเพื่อช่วยในการตรวจสอบปริมาณรถและความเร็วของรถบนท้องถนน เพื่อที่จะนำไปทำการวิเคราะห์หาความคล่องตัวของถนน เพื่อช่วยในการควบคุมการจราจรให้ง่ายขึ้น อีกทั้งยังสามารถช่วยลดปริมาณของบุคลากรในการควบคุมการจราจรให้ลดน้อยลงด้วย ระบบจะทำการรับภาพการจราจรบนท้องถนนจากกล้องวิดีโอแล้วนำภาพที่ได้ไปประมวลผลกับซอฟต์แวร์ ซึ่งซอฟต์แวร์จะทำการการนับรถและวัดความเร็วของรถด้วยเทคนิคประมวลผลภาพ (Image processing) ผลที่คาดว่าจะได้รับก็คือ สามารถนับรถและวัดความเร็วของรถได้ และนำข้อมูลที่ได้ไปทำการวิเคราะห์เพื่อนำไปใช้ในการลดปัญหาการจราจรต่อไป

จากการสำรวจวิธีการที่ใช้ในการตรวจสอบปริมาณและความเร็วของรถในปัจจุบันนี้จะใช้เทคโนโลยีของเซนเซอร์ประเภทต่างๆ เช่น อินฟาเรดในการตรวจสอบ ซึ่งในเซนเซอร์แต่ละตัวจะมีข้อจำกัดในการทำงาน เช่น เซนเซอร์ที่ใช้ในการตรวจสอบความเร็วของรถแต่ละตัวนั้นจะสามารถตรวจสอบรถได้เพียงช่องการจราจรเดียวเท่านั้น ฉะนั้นในการวิเคราะห์เพื่อที่จะควบคุมการจราจรบนท้องถนนให้มีประสิทธิภาพนั้นจะต้องอาศัยข้อมูลจากหลายๆ ด้านประกอบกัน การนำเทคโนโลยีของการประมวลผลภาพมาช่วยในการวิเคราะห์ จึงเป็นทางเลือกที่ช่วยให้ระบบมีความง่ายต่อการดูแลและติดตั้งมากขึ้น

1.2 วัตถุประสงค์ของโครงงาน

เพื่อสร้างระบบตรวจสอบการจราจร โดยใช้กล้องวิดีโอเป็นตัวรับภาพเข้ามาแล้วทำการประมวลผลภาพที่รับเข้ามา เพื่อตรวจสอบการเคลื่อนที่ของรถบนท้องถนน โดยมีผลลัพธ์ที่ได้ ดังนี้

1. สามารถนับปริมาณรถ ความเร็วในการเคลื่อนที่ของรถแต่ละคัน และค่าความเร็วเฉลี่ยของรถ ซึ่งจะนำไปเป็นข้อมูลในการวัดความหนาแน่นของรถ
2. เพื่อลดภาระหน้าที่ของเจ้าหน้าที่ในการตรวจสอบสภาพการจราจร
3. นำข้อมูลที่ได้ไปทำการวิเคราะห์เพื่อลดปัญหาการจราจร

1.3 ขอบเขตของโครงงาน

ระบบตรวจสอบสภาพการจราจรจะประกอบไปด้วย 2 ส่วนหลักๆ คือ ส่วนฮาร์ดแวร์ (Hardware) และส่วนซอฟต์แวร์ (Software)

1.ส่วนของฮาร์ดแวร์ ประกอบด้วยกล้องวิดีโอหนึ่งตัวที่ติดตั้งอยู่เหนือถนน และรับภาพเข้ามาผ่านทางพอร์ต USB ของคอมพิวเตอร์

2. ส่วนซอฟต์แวร์ เป็นส่วนของโปรแกรมที่ติดตั้งอยู่บนเครื่องคอมพิวเตอร์ จะนำภาพที่ได้จากกล้องวิดีโอมาทำการประมวลผลภาพเพื่อหาส่วนที่เป็นรถ เมื่อได้ส่วนที่เป็นรถแล้วจะทำการนับจำนวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และวัดความเร็วของรถ และแสดงผลลัพธ์ที่ได้ออกมาเป็นตัวเลข การทำงานจะแบ่งออกเป็น 2 โหมด ด้วยกัน คือส่วนการทำงานแบบเรียลไทม์ (Real time) มีการรับภาพจากกล้องวิดีโอแล้วนำไปทำการประมวลผลทันทีนั้น และส่วนของออฟไลน์ (Offline) จะมีการบันทึกภาพไว้ล่วงหน้าแล้วจึงนำภาพที่บันทึกมาทำการประมวลผล

สำหรับส่วนของซอฟต์แวร์นั้นจะทำการพัฒนาโดยใช้ ไมโครซอฟต์ วิชวล ซีพลัสพลัส ดอทเน็ท นอกจากนั้นยังใช้ไมโครซอฟต์ ไดเร็กซ์เอ็กซ์ 9.0 ในส่วนของการดึงภาพจากกล้องวิดีโอ

1.4 วิธีการดำเนินงาน

เริ่มจากการศึกษาทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้อง ได้แก่ ศึกษาและจัดหาข้อมูลในการจัดการกับไฟล์รูปภาพ ศึกษาวิธีการดึงภาพจากกล้องวิดีโอ ศึกษาแนวทางการตรวจสอบการเคลื่อนที่ของวัตถุเพื่อใช้ในการประมวลผลภาพ ศึกษาวิธีการในการนับรถและตรวจสอบความเร็วของรถ

หลังจากที่ได้ศึกษาทฤษฎีต่างๆ แล้วก็นำทฤษฎีที่ได้ศึกษามาใช้ ในส่วนของซอฟต์แวร์ จะทำการเขียนโปรแกรมดึงภาพจากกล้องวิดีโอและไฟล์ที่ได้ทำการบันทึกไว้ เขียนโปรแกรมในการจัดการกับไฟล์รูปภาพ เขียนโปรแกรมในการตรวจสอบการเคลื่อนที่ของวัตถุและเขียนโปรแกรมในการนับรถและเช็คความเร็วของรถ เขียนโปรแกรมในส่วนการติดต่อกับผู้ใช้งาน ในส่วนของฮาร์ดแวร์นั้นก็มีการนำกล้องวิดีโอไปทำการบันทึกภาพในมุมมองต่างๆ เพื่อนำมาใช้เป็นข้อมูลในการทดลอง

เมื่อลงมือปฏิบัติเรียบร้อยแล้ว ก็ทำการทดสอบระบบ แก้ไขจุดบกพร่องที่ได้จากทดสอบ เพื่อเพิ่มประสิทธิภาพการทำงานของระบบ ท้ายสุดก็นำผลการทดลองที่ได้มาทำการวิจารณ์และสรุปผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีการประมวลผลภาพ

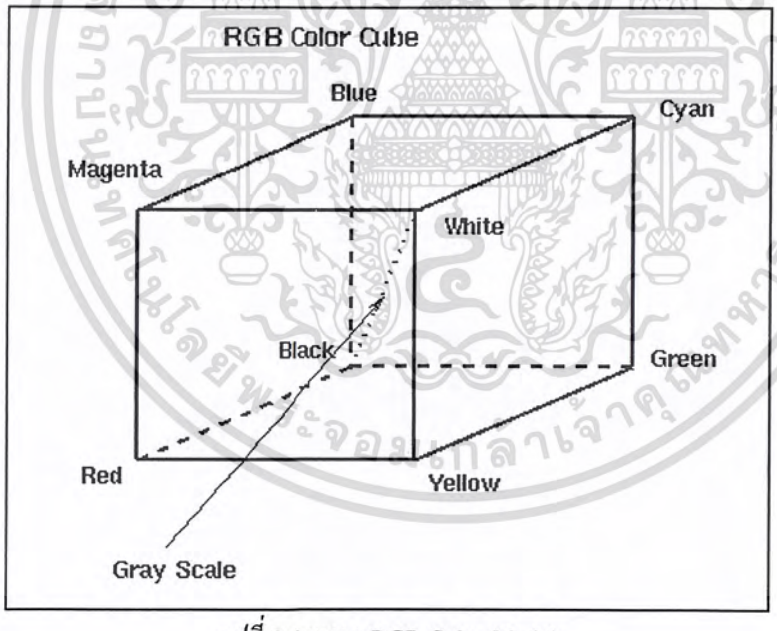
สำหรับหลักการและเทคนิคที่นำมาใช้กับโครงงานนี้ ส่วนใหญ่จะเป็นอัลกอริธึมที่เกี่ยวกับขบวนการประมวลผลภาพ (Image processing) เกือบทั้งสิ้น ซึ่งรายละเอียดของแต่ละขั้นตอนมีดังต่อไปนี้

2.1 Color model

ในการประมวลผลภาพจะมีโมเดลเกี่ยวกับสีที่มักนำมาใช้ 3 แบบคือ RGB โมเดล, YIQ โมเดล และ HSI โมเดล แต่โดยทั่วไปแล้วทุกมาตรฐานจะมีแนวคิดเดียวกันคือ การแทนจุดสีด้วยจุดที่อยู่ภายใน Space ซึ่งแต่ละแกนจะมีความอิสระต่อกัน สำหรับโมเดลที่โครงงานนี้นำมาใช้จะมีอยู่ 2 โมเดลคือ RGB โมเดลและ YIQ โมเดล ซึ่งมีรายละเอียดดังต่อไปนี้

2.1.1 The RGB color model

สำหรับ RGB โมเดลจะเป็นสีที่เกิดจากการรวมกันของสีแดง เขียวและน้ำเงิน รูปแบบของโมเดลจะอยู่ในรูปของคาทีเซี่ยน โดยสับเซตของสีจะเกิดจากสี่เหลี่ยมลูกบาศก์ดังรูปที่ 2-1 สีแดง เขียวและน้ำเงินจะอยู่ 3 มุม ส่วนมุมอื่นๆ ที่เหลือจะเป็นสี cyan, magenta และ yellow สีดำจะอยู่ที่จุด origin และสีขาวจะอยู่มุมตรงข้ามกับสีดำ ในโมเดลจะแทน Gray scale อยู่บนเส้นประที่ลากจากมุมสีดำไปยังมุมสีขาว ค่าของสีแดง เขียวและน้ำเงินจะถูกกำหนดให้อยู่ในช่วง [0, 1]



รูปที่ 2-1 แสดง RGB Color Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 The YIQ color model

เป็นระบบที่ใช้ในการกระจายสัญญาณโทรทัศน์ YIQ โมเดลเป็นการแปลงมาจาก RGB โมเดล เพื่อเพิ่มประสิทธิภาพและเพื่อให้เหมาะกับมาตรฐานของโทรทัศน์ขาวดำ ในความเป็นจริง Y ใน YIQ โมเดลนั้นจะเป็นข้อมูลภาพวิดีโอที่ต้องการของโทรทัศน์ขาวดำ การแปลงจาก RGB โมเดลเป็น YIQ โมเดลสามารถทำได้จากสมการต่อไปนี้

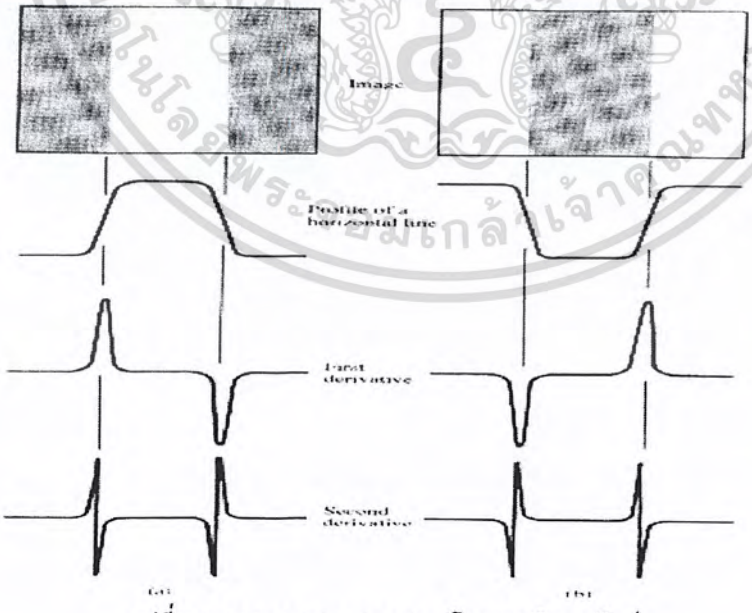
$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

ในส่วนของการแปลง YIQ โมเดลให้อยู่ในเซตของ RGB โมเดลเราสามารถทำได้โดยการทำ Inverse matrix เท่านั้น

ประโยชน์หลักๆ ของ YIQ โมเดลในการประมวลผลภาพก็คือค่า Y (Luminance) และส่วนที่เป็นข้อมูลของสี (I และ Q) ลูมิแนนซ์ในความคิดก็คือสัดส่วนของแสงที่ตาของเราได้รับ ความสำคัญของลูมิแนนซ์ในรูปภาพก็คือจะเป็นส่วนที่ไม่มีสีอยู่เลย

2.2 ทฤษฎีการหาขอบของวัตถุ (Edge Detection)

การพิจารณาขอบของวัตถุภายในภาพ ขอบของวัตถุนั้นจะเป็นบริเวณที่มีการเปลี่ยนแปลงของค่าความเข้ม (Intensity) อย่างรวดเร็ว (มีความแตกต่างของความเข้มมากเป็นพิเศษ) ซึ่งในการตรวจจับขอบนั้น เราสามารถให้เห็นบริเวณขอบของภาพมีความเปลี่ยนแปลงของความเข้มมากขึ้น โดยการหาอนุพันธ์ได้ดังรูปที่ 2-2



รูปที่ 2-2 การหา Edge detection โดยการทำอนุพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป (a) และ (b) เป็นรูปแสดงความเข้มของแสงและกราฟแสดงลำดับความเข้มของแสงเห็นได้ว่า เมื่อภาพมีการเปลี่ยนจากระดับความเข้มของแสงที่ต่ำไปยังระดับความเข้มของแสงที่สูง ดังรูป (a) กราฟความเข้มของแสงจะสูงขึ้น ในทางกลับกันเมื่อภาพมีการเปลี่ยนแปลงจากระดับสูงไปยังระดับความเข้มของแสงที่ต่ำกว่า ดังรูป (b) กราฟความเข้มของแสงจะต่ำลง

เมื่อทำอนุพันธ์อันดับที่ 1 กับภาพจะเห็นได้ว่า ส่วนที่กราฟความเข้มของแสงเปลี่ยนแปลงในขาขึ้น กราฟอนุพันธ์อันดับที่ 1 จะเป็นบวก ในส่วนที่กราฟความเข้มของแสงเปลี่ยนแปลงในขาลง กราฟอนุพันธ์อันดับที่ 1 จะเป็นลบ และช่วงใดที่ไม่มีการเปลี่ยนแปลงค่าความเข้มของแสง กราฟอนุพันธ์อันดับที่ 1จะมีค่าเท่ากับศูนย์จากหลักการเดียวกัน ถ้าเราทำอนุพันธ์อันดับที่ 2 จะได้กราฟสุดท้าย ดังรูป (a) และ (b) ซึ่งภาพที่ได้จากการทำอนุพันธ์อันดับที่ 2 จะมีขอบที่ชัดเจนขึ้น

2.2.1 Gradient Method

วิธีการ Gradient Method สำหรับการตรวจจับขอบของภาพมีหลักการคือ บริเวณขอบของวัตถุภายในภาพ มักจะมีค่า Gradient ที่สูง ดังนั้นจึงพิจารณาขนาดของ Gradient โดยจะทำการ Threshold ด้วยค่า T ถ้า $|\nabla P| > T$ จะเป็นขอบ

การทำ Gradient operator นี้ทำโดยการหาอนุพันธ์ตามแนวแกน X และแกน Y ของภาพแล้วนำมาคำนวณหาขนาดของ $|\nabla P|$ ดังสมการต่อไปนี้

$$\nabla P = \hat{i}\frac{\partial P}{\partial X} + \hat{j}\frac{\partial P}{\partial Y}$$

$$|\nabla P| = \sqrt{\left(\frac{\partial P}{\partial X}\right)^2 + \left(\frac{\partial P}{\partial Y}\right)^2}$$

ดังนั้น จะทำการตรวจจับขอบเมื่อ $|\nabla P| > T$; T คือค่าคงที่

การหาอนุพันธ์ตามแนวแกน X และแกน Y ของภาพนั้นสามารถทำได้โดยการใช้ mask ขนาด 3X3 คูณกับภาพ และ mask ที่ใช้ในการคำนวณจะใช้ mask ดังรูปที่ 2-3

-1	-2	-1
0	0	0
1	2	1

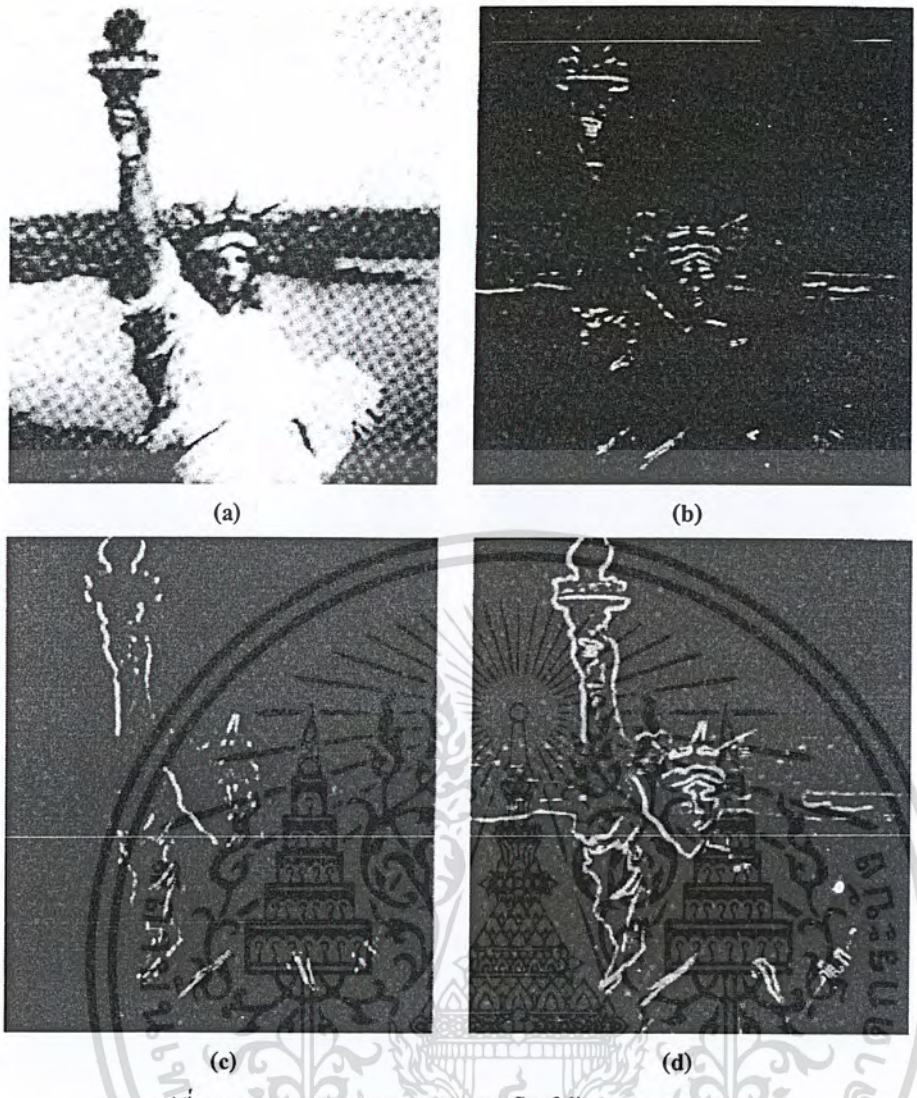
(a)

-1	0	1
-2	0	2
-1	0	1

(b)

รูปที่ 2-3 (a) Mask ที่ใช้ในการคำนวณ $\frac{\partial P}{\partial X}$ (b) Mask ที่ใช้ในการคำนวณ $\frac{\partial P}{\partial Y}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-4 แสดงหลักการตรวจจับขอบโดยใช้ Gradient operator

จากรูปที่ 2-4 แสดงหลักการตรวจจับขอบโดยใช้ Gradient operator รูป (a) เป็นภาพตั้งต้น รูป (b) แสดงภาพที่ได้จากการคำนวณ Partial Derivative กับแนวแกน X รูป (c) เป็นภาพที่ได้จากการคำนวณ Partial Derivative ในแนวแกน Y รูป (d) เป็นภาพสำเร็จหลังจากการทำ Gradient

2.2.2 Laplacian Method

Laplacian Method เปรียบได้กับการหาอนุพันธ์อันดับที่ 2 ของภาพเพื่อใช้ในการตรวจจับขอบของภาพ โดยภาพที่ผ่านการหาอนุพันธ์อันดับที่ 2 บริเวณที่เป็นส่วนของขอบจะเห็นได้ชัดเจนขึ้น

จากหลักการเดียวกันกับ Gradient Method เราจะทำการประมาณค่าของ $\nabla^2 P$ โดยใช้ mask รูปที่ 2-3 โดยตำแหน่งขอบของภาพ ก็คือค่า Zero Crossing ของ $\nabla^2 P$ ซึ่ง Laplacian operator นี้ทำโดยการหาอนุพันธ์อันดับที่ 2

ในการทำ Edge Detection โดยใช้ Laplacian operator นี้ที่บริเวณขอบจะมีตำแหน่งเดียวกับ Zero Crossing ของ $\nabla^2 P$ (ตำแหน่งที่พิกเซลของ $\nabla^2 P$ เปลี่ยนจากบวกเป็นลบหรือเปลี่ยนจากลบเป็นบวก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\nabla^2 P = \frac{\partial^2 P}{\partial X^2} + \frac{\partial^2 P}{\partial Y^2} \qquad ; \text{เมื่อ } \nabla^2 P \text{ คือ Laplacian Image}$$

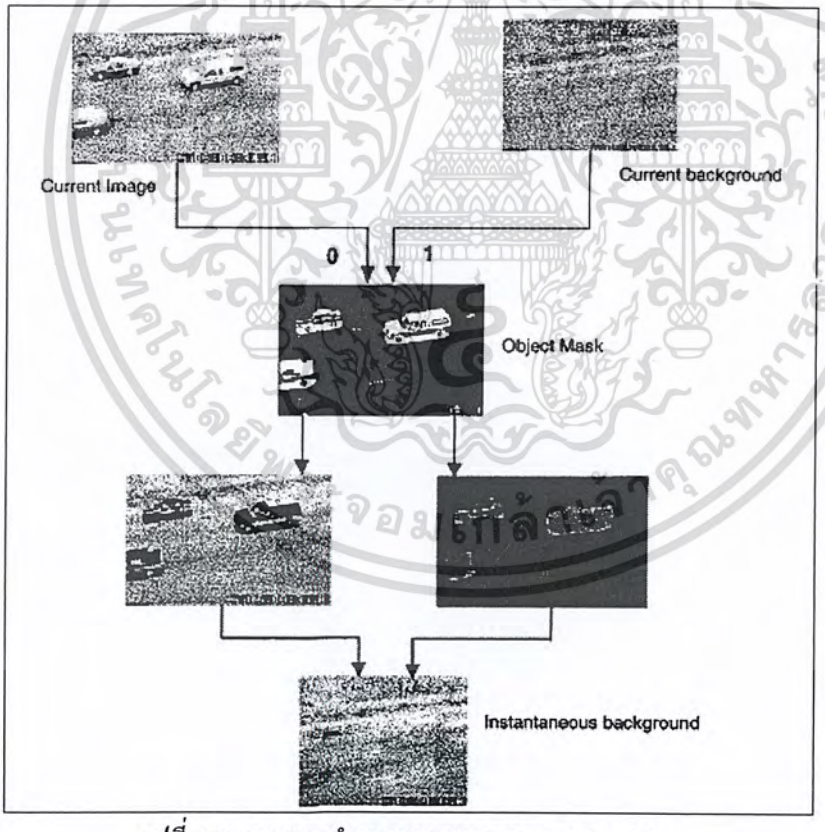
2.3 ทฤษฎีการหาวัตถุที่กำลังเคลื่อนที่ (Object Detection)

ในขบวนการประมวลผลภาพจะมีวิธีการในการตรวจสอบหาวัตถุที่มีการเคลื่อนที่ภายในรูปภาพอยู่หลายวิธีด้วยกัน สำหรับอัลกอริธึมแต่ละวิธีจะมีประสิทธิภาพและการทำงานที่ซับซ้อนต่างกัน การตรวจจับการเคลื่อนที่ของวัตถุแบ่งออกเป็น 2 อัลกอริธึมหลักๆ คือ Optical-flow algorithm และ Frame differencing algorithm

Optical-flow algorithm สามารถที่จะคำนวณขนาดและทิศทางการเคลื่อนที่ของวัตถุ ความถูกต้องของอัลกอริธึมสูง การทำงานต้องการความเร็วในการคำนวณสูง

ส่วน Frame differencing algorithm สามารถแบ่งออกเป็น 2 วิธีคือ Background และ Inter-frames difference

Background นั้นจะทำการแยกส่วนของวัตถุที่กำลังเคลื่อนที่ได้จากการคำนวณจากค่า Threshold ของเฟรมปัจจุบันกับเฟรมอ้างอิง ซึ่งเราจะเรียกเฟรมอ้างอิงนั้นว่า "Background image" วิธี Background จะมีข้อจำกัดอยู่ว่าจะต้องมีการคำนวณหา Background image เพื่อทำการอัพเดทภาพ background อยู่เสมอ



รูปที่ 2-5 แสดงการทำงานของ Background algorithm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการทำงานของ Inter-frames difference จะมีเทคนิคในการตรวจสอบหาวัตถุที่กำลังเคลื่อนที่ โดยการเปรียบเทียบตำแหน่งของวัตถุในเฟรมที่ต่อเนื่องกัน สำหรับโครงงานนี้ได้เลือกใช้อัลกอริธึมของ Inter-frame difference มาใช้ สำหรับอัลกอริธึมนี้จะได้มาจากการทำ Double-difference operator หรือเรียกอีกอย่างว่า Three- frames difference เหตุผลที่อัลกอริธึมนี้ได้ความนิยมก็คือ สามารถตรวจจับการเคลื่อนที่ของวัตถุในพื้นที่ต่างๆ ได้ดี ทำงานได้ดีกับสภาวะแวดล้อมที่มีสัญญาณรบกวน อีกทั้งกล้องสามารถสวิงมุมกล้องได้และสามารถที่จะกรองวัตถุเล็กๆ ที่เคลื่อนที่อยู่ภายในรูปภาพได้

รูปที่ 2-6 แสดงลำดับการทำงานของ Inter-frames algorithm

การทำงานในส่วนของการตรวจสอบหาการเคลื่อนที่ของวัตถุในส่วนของโครงงานนี้ จะมีอัลกอริธึมที่นำมาใช้อยู่ 5 ขั้นตอนดังต่อไปนี้

2.3.1 Inter-frames difference

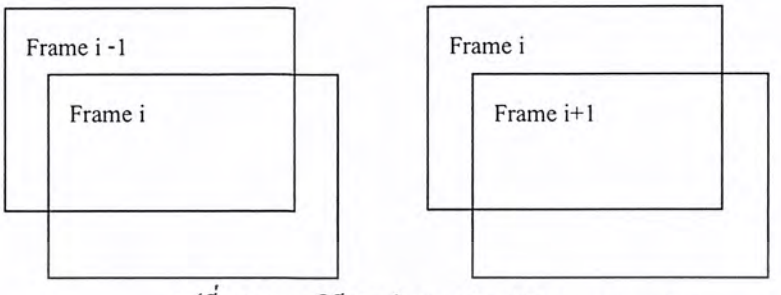
สำหรับเทคนิคนี้จะใช้สมการของ Double operator ในการทำงานในส่วนนี้ โดยการนำเฟรม 2 มาทำการเปรียบเทียบพิกเซลของทั้งสองภาพตามค่า Threshold (T) ที่เราได้กำหนดเอาไว้ซึ่งมีสมการดังนี้

$$D_n(i,j) = | I_n(i,j) - I_{n-1}(i,j) |$$

$$DD_n = \begin{cases} 1, if\ (D_{n+1}(i,j) > T) \wedge (D_n(i,j) > T) \\ 0, \qquad otherwise \end{cases}$$

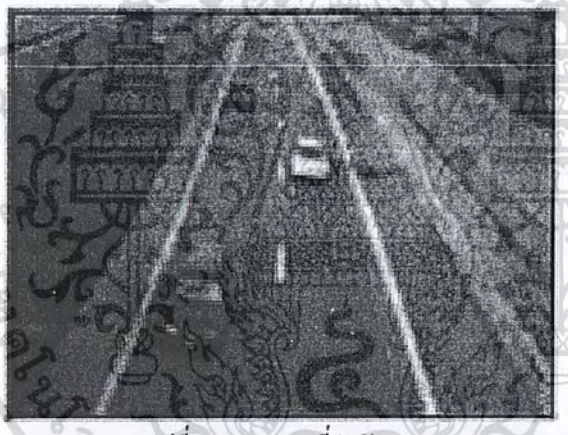
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการนำภาพมาเทียบพิกเซลกันแต่ละครั้งนั้นจะต้องใช้เฟรมที่นำมาเทียบกันทั้งหมด 3 เฟรม โดยจะเทียบกันระหว่างเฟรมปัจจุบันกับเฟรมก่อนหน้าและเฟรมปัจจุบันกับเฟรมถัดไปซึ่งจะมีการเปรียบเทียบกันทั้งหมด 2 ครั้ง

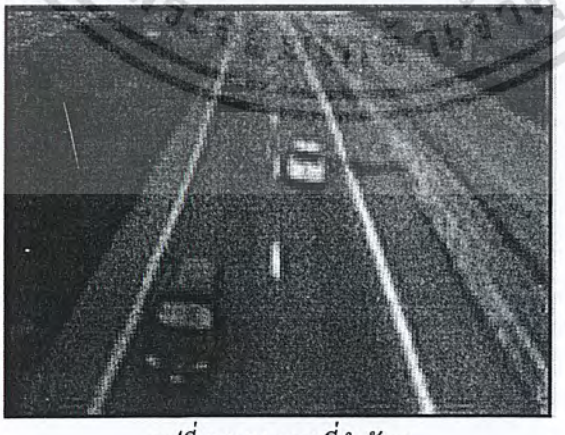


รูปที่ 2-7 แสดงวิธีการทำ Double-difference operator

เมื่อทำการเปรียบเทียบครบแล้วจะได้ส่วนที่มีความแตกต่างของพิกเซลในแต่ละเฟรมคือส่วนของวัตถุที่มีการเคลื่อนที่ ดังตัวอย่างจากรูปที่ จะเห็นได้ว่าจะมีส่วนที่แตกต่างกันของพิกเซล คือส่วนของรถที่มีการเคลื่อนที่ ดังนั้นเราจะทราบได้ว่าส่วนนี้เป็นรถที่แล่นอยู่บนท้องถนน

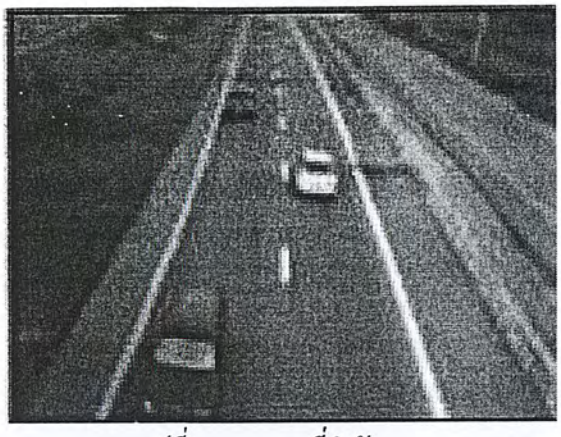


รูปที่ 2-8 (a) ภาพที่ลำดับ i-1

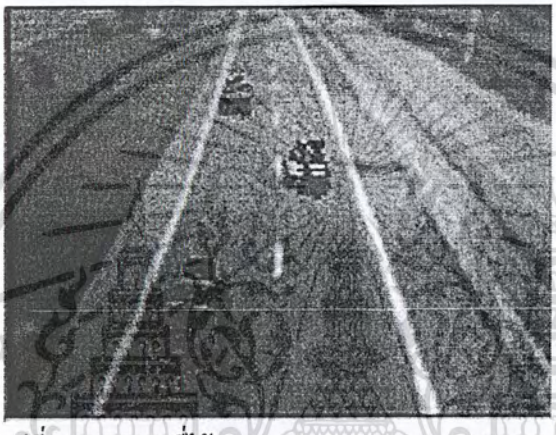


รูปที่ 2-8 (b) ภาพที่ลำดับ i

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-8 (c) ภาพที่ลำดับ i+1



รูปที่ 2-8 (d) ภาพที่ได้จากการทำ Inter-frames difference

2.3.2 Sobel-Edge detection

การทำ Sobel-Edge detection ทำเพื่อที่จะต้องการส่วนที่เป็นขอบของรูปภาพ ซึ่งการ Sobel-Edge detection เป็นการทำไฮพาสฟิลเตอร์ชนิดหนึ่ง โดยมีสมการดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Z1 **i-1, j-1**	**Z2** **i, j-1**	**Z3** **i+1, j-1**
Z4 **i-1, j**	**Z5** **i, j**	**Z6** **i+1, j**
Z7 **i-1, j+1**	**Z8** **i, j+1**	**Z9** **i+1, j+1**

$$G_x = (Z_7 + (2 * Z_8 + Z_9) - (Z_1 + (2 * Z_1) + Z_3)$$
$$G_y = (Z_3 + (2 * Z_6 + Z_9) - (Z_1 + (2 * Z_4) + Z_7)$$
$$G = |G_x| + |G_y|$$

หลังจากที่เราได้ทำ Sobel-Edge Detection ตามสมการข้างต้นแล้วจะได้ส่วนที่เป็นขอบของรูปภาพดังรูปที่ 2-9(b)

รูปที่ 2-9 (a) ภาพก่อนที่จะทำ Sobel-Edge Detection

รูปที่ 2-9 (b) ภาพที่ได้จากการทำ Sobel-Edge Detection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 Moving-Edge closure (MEC)

การทำ Moving-Edge closure เพื่อเป็นการเพิ่มรายละเอียดให้กับส่วนที่เป็นรถมากยิ่งขึ้น เปรียบเสมือนการเติมเต็มให้กับส่วนสีขาวที่อยู่ภายในรถ การทำ Moving-Edge closure เพื่อให้ง่ายต่อการทำ Labeling และ Boundary ที่ใช้ตรวจนับจำนวนของรถที่อยู่ภายในรูปภาพ ซึ่งการทำ Moving-Edge closure สามารถทำได้โดยการนำรูปที่ผ่านการทำ Inter-frames difference กับ Sobel-Edge detection มาทำสมการดังต่อไปนี้

$$MEC_0^n = \begin{cases} 1, \textit{ if } DD_n(i,j) = 1 \\ 0, \quad \textit{otherwise} \end{cases}$$

$$MEC_n^r = \begin{cases} 1, & \textit{if } MEC_n^{r-1}(i,j) = 1 \\ & \vee (\nabla I_n(i,j)) > T_g \\ & \wedge \exists (k,i) \in X \mid MEC_n^{r-1}(k,i) = 1) \\ 0, & \textit{otherwise} \end{cases}$$

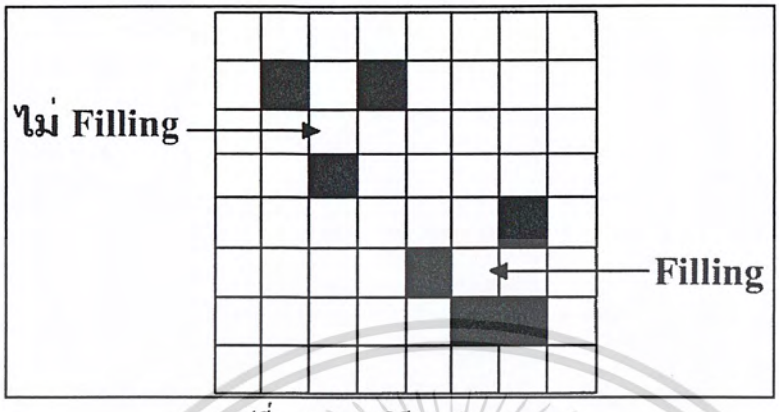
รูปที่ได้จากการทำ Moving-Edge closure จะมีความเข้มในส่วนที่เป็นรถมากยิ่งขึ้น จะเห็นความแตกต่างจากของส่วนรถได้อย่างชัดเจน ดังรูปที่ 2-10

รูปที่ 2-10 Moving-Edge closure

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 การทำ Filling

การทำ Filling เป็นการเติมเต็มในส่วนที่เป็นช่องว่างของวัตถุที่กำลังเคลื่อนที่ เพื่อให้ส่วนที่เป็นวัตถุกำลังเคลื่อนที่ชัดเจนยิ่งขึ้น และช่วยให้ขั้นตอนการทำ labeling ง่ายขึ้น



รูปที่ 2-11 แสดงวิธีการทำ Filling

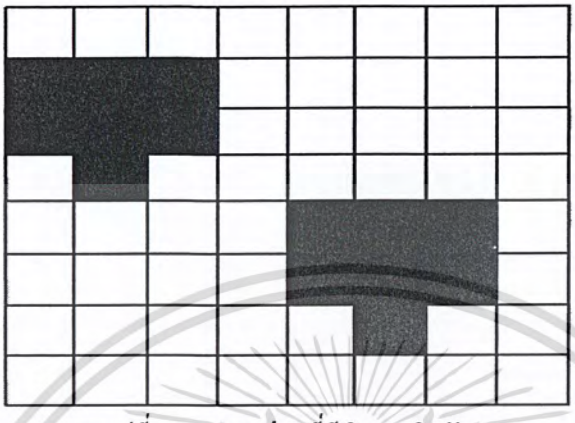
จากรูปที่ 2-11 การทำ Filling มีวิธีการคือ การ Filling จะพิจารณาพิกเซลรอบๆ พิกเซลที่สนใจซึ่งมีค่าไบนารี่บิตเป็น 0 (สีขาว) หากมีพิกเซลที่มีไบนารี่บิตเป็น 1 (สีดำ) มากกว่า 4 จุดขึ้นไปล้อมรอบพิกเซลที่สนใจก็จะเปลี่ยนไบนารี่บิตของพิกเซลที่สนใจนั้นให้เป็น 1 แต่ถ้ามี น้อยกว่า 4 จุดก็จะไม่ทำการเปลี่ยน หลังจากทำ filling แล้วจะเห็นได้ว่าส่วนที่เป็นวัตถุที่กำลังเคลื่อนที่มีความชัดเจนยิ่งขึ้น ดังรูปที่ 2-12

รูปที่ 2-12 หลังจากทำ Filling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

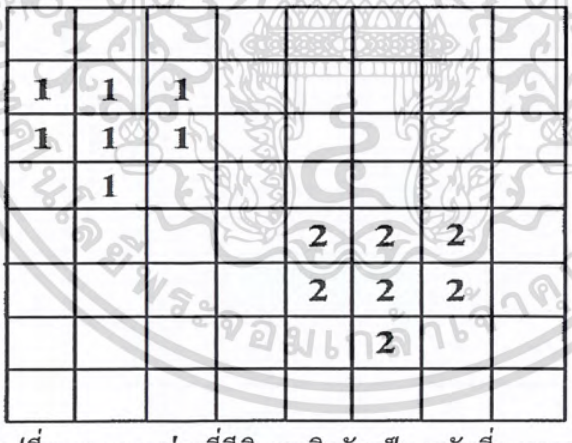
2.3.5 Labeling

การทำ Labeling เป็นการกำหนด label ให้กับพิกเซล ที่มีบิตไบนารี่เท่ากันและอยู่ติดกันทำเพื่อแสดงความสัมพันธ์ของพิกเซลที่เป็นของรถคันเดียวกัน การทำ Labeling ยังเป็นส่วนหนึ่งของการทำ Boundary อีกด้วย



รูปที่ 2-13 แสดงส่วนที่มีพิกเซลติดกัน

จากรูปที่ 2-13 แสดงให้เห็นว่ามีการเกาะกลุ่มกันของพิกเซลอยู่ 2 ส่วน ทำให้ทราบได้ว่าภายในรูปภาพมีส่วนที่เป็นวัตถุกำลังเคลื่อนที่อยู่ 2 ส่วน ดังนั้นจะทำการกำหนดให้ส่วนที่พิกเซลติดกันกลุ่มที่ 1 เป็นรถคันที่ 1 และส่วนที่พิกเซลติดกันกลุ่มที่ 2 เป็นรถคันที่ 2



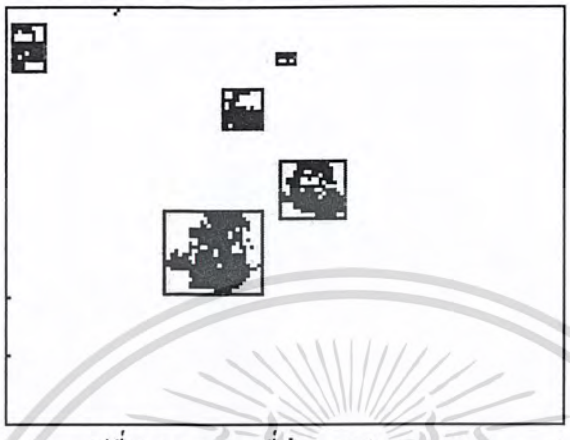
รูปที่ 2-14 แสดงส่วนที่มีพิกเซลติดกัน เป็นรถคันที่ 1 และ 2

2.3.6 การตีกรอบส่วนที่เป็นรถ (Boundary)

การทำ Boundary เป็นการตีกรอบครอบวัตถุที่เป็นรถ เพื่อที่จะนำไปใช้ในการนับรถ และตรวจสอบความเร็วของรถ โดยอัลกอริธึมที่ใช้ในการตีกรอบก็คือ จะทำการแสกนภาพเสียจากมุมล่างซ้าย ลำดับในการแสกน คือ ซ้ายไปขวา ล่างขึ้นบน เมื่อแสกนไปเจอไบนารี่บิตที่เป็นส่วนของตัวรถหรือพิกเซล

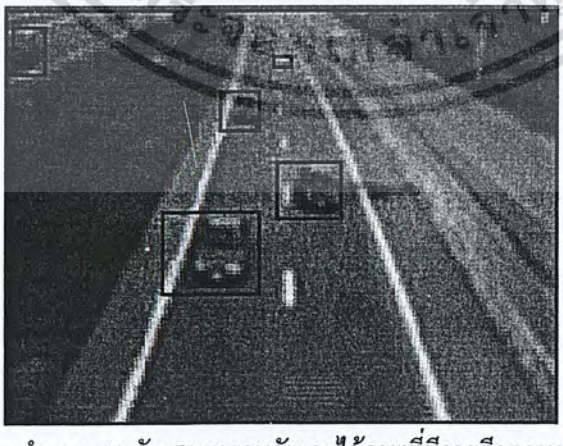
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่เป็นสีดำก็จะทำการกำหนดขอบตัวรถ และจะทำการแสกนจากมุมล่างซ้ายของอาณาเขตที่ได้กำหนดไว้อีกครั้งเพื่อหาอาณาเขตของรถทั้งหมดจากนั้นจึงทำการตีกรอบ หลังจากได้กรอบของตัวรถแล้ว จะทำการก๊อบปี้ส่วนกรอบของตัวรถนี้ไปเก็บไว้ในเมมโมรี่อีกส่วนหนึ่งเพื่อใช้ในการนับรถและตรวจสอบความเร็วของรถในขบวนการถัดไป ส่วนกรอบของตัวรถที่อยู่ในภาพปัจจุบัน จะถูกกำหนดให้เป็นสีขาว(สีพื้นหลัง) เพื่อที่จะทำการแสกนหาส่วนที่เป็นรถคันอื่นต่อไป



รูปที่ 2-15 (a) ภาพที่ผ่านการทำ Boundary

รูปที่ 2-15 (b) ภาพต้นแบบ

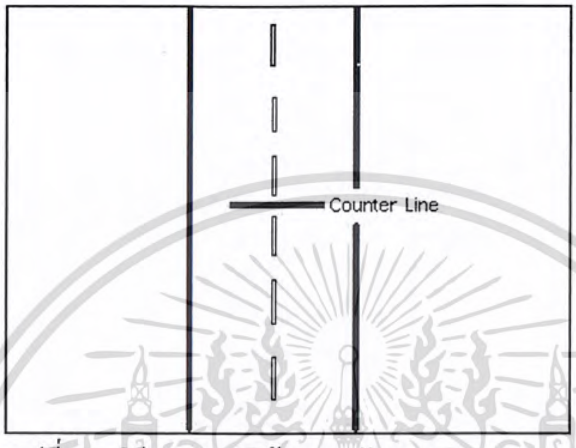


รูปที่ 2-15 (c) นำภาพ (a) กับ (b) มารวมกัน จะได้ภาพที่มีการตีกรอบส่วนที่เป็นรถ

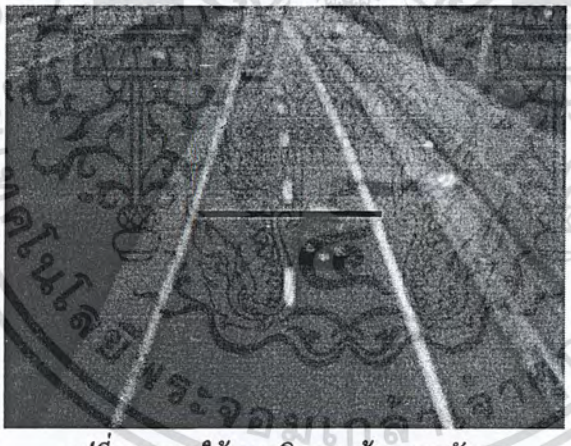
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ทฤษฎีและหลักการที่ใช้ในการนับรถ

สำหรับวิธีที่ใช้ในการนับรถมีอยู่หลายวิธีด้วยกัน เช่น การใช้อุปกรณ์อินฟาเรดในการนับรถ แต่ข้อเสียของการนับรถด้วยอุปกรณ์อินฟาเรดก็คือสามารถนับรถได้เพียงแค่ช่องทางเดียวเท่านั้น การใช้งานต้องมีตัวรับและตัวส่ง อายุการใช้งานของอุปกรณ์ไม่นาน ในขณะที่การประมวลผลทางภาพสามารถที่จะตรวจนับรถได้ทีละหลายๆ ช่องทาง ทำให้การนับรถทำได้อย่างสะดวก และอุปกรณ์ในการใช้งานสามารถหาได้ง่าย เพียงมีกล้องวิดีโอเพียงตัวเดียวติดตั้งอยู่เหนือถนนที่เราต้องการที่จะตรวจสอบ



รูปที่ 2-16 วิธีการกำหนดเส้นตรวจนับรถ (Counter line)



รูปที่ 2-17 การใช้งานจริงของเส้นตรวจนับรถ

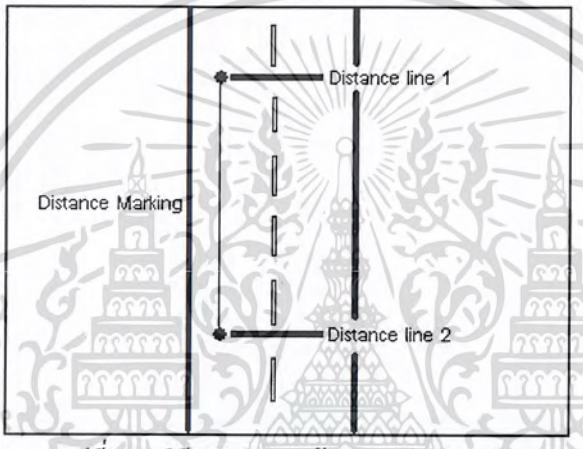
สำหรับเทคนิคที่นำมาใช้ในการนับรถสามารถหาจำนวนรถได้จากรูปที่ทำการ Boundary มาเรียบร้อยแล้ว เมื่อมีรถวิ่งเส้นตรวจนับจำนวนรถที่ได้กำหนดไว้เรียบร้อยแล้วดังรูปที่ 2-17 อ้างอิงของหน้าระบบจะทำการนับจำนวนรถ โดยสามารถนับรถที่วิ่งผ่านเส้นตรวจนับพร้อมกันทีละหลายๆ คันได้ เนื่องจากระบบจะทำการแสกนหารถที่ทับเส้นตรวจนับจากซ้ายไปขวา เพื่อทำการนับรถทุกคันที่ทับเส้นตรวจนับนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

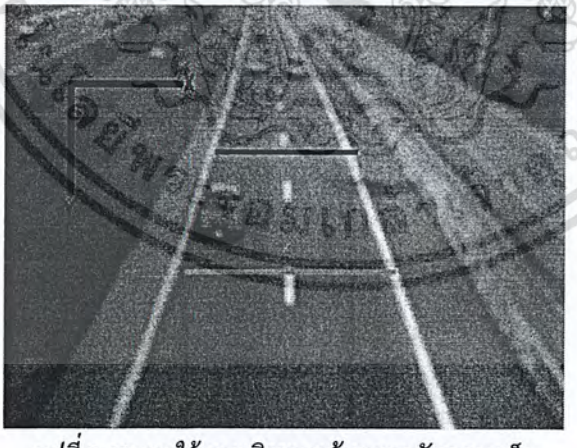
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.5 หฤษฎีและหลักการที่ใช้ในการวัดความเร็วรถ

วิธีการหาความเร็วสามารถทำได้หลายวิธี เช่น ใช้ทิวบ์ดีเทคเตอร์ (Tube detector) ลูพคอยล์ดีเทคเตอร์ (Loop coil detector) อุปกรณ์อินฟาเรด portable radar การนับด้วยมือ อุปกรณ์ Ultrasonic และการประมวลผลภาพ เป็นต้น การใช้อุปกรณ์ที่กล่าวมาข้างต้นยังมีอุปสรรค เช่น การใช้อุปกรณ์เรดาร์จะมีปัญหาเมื่อมีจำนวนรถมากกว่า 1 ช่องทาง การใช้ Loop coil หรือ Tube มีการ-เสื่อมสภาพง่าย อายุใช้งานไม่นาน การบำรุงรักษาหรือซ่อมแซมต้องปิดช่องทางจราจร ในขณะที่การประมวลผลภาพมีข้อดีเหนือกว่า เช่น ไม่ถูกกระทบกระเทือนจากการรบกวนบนผิวถนน สามารถย้ายไปตรวจจับความเร็วในบริเวณต่างๆ ได้โดยเปลี่ยนมุมกล้อง ด้วยข้อดีเหล่านี้จึงทำให้มีการใช้ภาพจากกล้องวิดีโอมาวิเคราะห์หาข้อมูลจราจรมากขึ้น อย่างไรก็ตามการใช้ภาพวิดีโอก็มีข้อจำกัดเกี่ยวกับความเร็วของกล้อง (จำนวนเฟรมต่อนาที) และมุมมองของภาพที่นำมาประมวลผล



รูปที่ 2-18 วิธีการกำหนดเส้นตรวจจับความเร็ว



รูปที่ 2-19 การใช้งานจริงของเส้นตรวจจับความเร็ว

สำหรับวิธีหาความเร็วรถจะคำนวณหาจากภาพที่ได้จากการทำ Boundary โดยรถจะวิ่งผ่านตามแนวแกน Y ของภาพดังรูปที่ 2-19 ในการหาความเร็วของรถต้องคำนวณตำแหน่งของขอบอ้างอิงหน้ารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
55136

ตำแหน่งแรก เมื่อเวลาผ่านไปจึงคำนวณหาตำแหน่งของขอบอ้างอิงของรถที่ตำแหน่งที่สอง ซึ่งความเร็วที่ได้คือระยะทางที่รถเคลื่อนที่จากตำแหน่งแรกไปสู่ตำแหน่งที่สองหารด้วยเวลาที่ใช้ในการเคลื่อนที่ และในการหาระยะทางจากตำแหน่งขอบอ้างอิงตำแหน่งรถ ทำได้โดยประมาณจากตำแหน่งที่ทราบค่าแล้วบนพื้นถนนโดยกำหนดจุดอ้างอิง (Reference point) ซึ่งขั้นตอนในการหาความเร็วมีดังนี้

1. กำหนดเส้นตรวจจับ (Detecting line) 2 เส้นในช่องทางจราจรที่รถแล่นผ่านระยะห่างกันพอประมาณ เพื่อตรวจจับบริเวณหน้าของรถตำแหน่งที่ 1 และตำแหน่งที่ 2 ดังรูปที่ 2-18 และ 2-19

2. อ่านข้อมูลภาพจากกล้องวิดีโอ ซึ่งภาพนั้นจะต้องผ่านขบวนการ Boundary มาเรียบร้อยแล้ว

3. หาตำแหน่งขอบอ้างอิงของหน้ารถในบริเวณเส้นตรวจจับที่ 1 (Detecting line 1) ถ้ามีปรากฏอยู่จะได้ตำแหน่งหน้ารถมีระยะเป็น s1 เมตร และเวลาขณะนั้นเป็น t1 เก็บลงไว้ใน Circular Queue ทั้งนี้เพื่อเก็บข้อมูลทุกคันที่ผ่านเส้นตรวจจับที่ 1 และทุกครั้งที่มีการรับเฟรมใหม่เข้ามาจะมีการอัพเดทค่ามุมทั้ง 4 มุมที่เป็นกรอบของรถตลอด เพื่อป้องการวัดความเร็วของรถผิดคัน

4. หาตำแหน่งขอบอ้างอิงของหน้ารถในบริเวณเส้นตรวจจับที่ 2 ถ้ามีรถปรากฏอยู่ จะได้ตำแหน่งหน้ารถมีระยะทางเป็น s2 เมตร และเวลาขณะนั้นเป็น t2

5. ถ้าขั้นตอนที่ 5 มีรถปรากฏอยู่ให้นำข้อมูล s1 และ t1 ออกจากคิว (Queue) คำนวณความเร็วได้จากสูตรต่อไปนี้

$$V = (S2-S1)/(T2-T1)$$

โดยที่ V เป็นความเร็วที่ระบบคำนวณได้

S1 S2 เป็นระยะทางที่รถเคลื่อนที่ จากเส้นตรวจจับที่ 1 และเส้นตรวจจับที่ 2 ตามลำดับ

T1 T2 เป็นเวลา ณ เส้นตรวจจับที่ 1 และเส้นตรวจจับที่ 2 ตามลำดับ

7. อ่านข้อมูลภาพจากกล้องวิดีโอไปเรื่อยๆ ถ้ายังไม่จบให้เริ่มทำข้อ 4 ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

รูปแบบของ ไฟล์ เอวีไอ

(Format files AVI)

รูปแบบของไฟล์ เอวีไอ นั้นประกอบด้วย 2 ส่วน คือ ส่วนที่เป็น ส่วนหัวของไฟล์ และส่วนที่เป็น เฟรม ของไฟล์ เอวีไอ

3.1 ส่วนหัวของไฟล์ เอวีไอ (AVI Header Format)

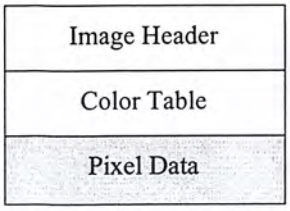
ไฟล์ เอวีไอ นั้นจะมีส่วนการอธิบายข้อมูลต่างๆของไฟล์อยู่ 56 ไบต์ ซึ่งจะมีการอธิบายส่วนต่างๆ ของไฟล์อยู่ทั้งหมด 14 ส่วนดังนี้

ตำแหน่ง	ขนาด	คำอธิบาย
0	4	ระยะเวลาในการแสดงภาพในแต่ละเฟรม มีหน่วยเป็น ไมโครเซกค์
4	4	อัตราข้อมูล ของข้อมูลแบบเอวีไอ
8	4	การเติมขนาดของข้อมูล ซึ่งอย่างน้อยมีค่าเท่ากับ 2048
12	4	ค่าพารามิเตอร์ แฟลกซ์ต่างๆ
16	4	จำนวนของ วีดิโอเฟรม
20	4	จำนวนของเฟรมที่นำมา แสดงเป็นตัวอย่าง
24	4	จำนวนของชุดข้อมูล (1 หรือ 2)
28	4	ข้อแนะนำว่าในเวลาแสดงผลควรใช้ขนาดบัฟเฟอร์ เท่าไร มีหน่วยเป็นไบต์
32	4	ความกว้างของภาพ วิดีโอ มีหน่วยเป็น พิกเซล
36	4	ความสูงของภาพ วีดิโอ มีหน่วยเป็น พิกเซล
40	4	ช่วงเวลา อย่างน้อย คือ 30
44	4	อัตราข้อมูล (อัตรา เฟรม = อัตราข้อมูล / ระยะเวลา)
48	4	เวลาเริ่มต้น , อย่างน้อย 0
52	4	ขนาดของข้อมูล เอวีไอ ในหนึ่งหน่วยเวลา

ตารางที่ 3- 1 รูปแบบส่วนหัวของไฟล์ เอวีไอ

ในที่นี้จะขอพูดถึง รูปแบบ ของเฟรม เอวีไอ เท่านั้น เฟรมของไฟล์เอวีไอ จะถูกเก็บอยู่ในรูปของ ดีไอบี (Device Independent Bitmap (DIB)) โดยโครงสร้างมาตรฐานของดีไอบีประกอบไปด้วย Image Header, optional color table of RGBQUAD's และ Data แสดงได้ดังรูปที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-1 แสดงรูปแบบของเฟรม เอวีไอ

3.2 ส่วนหัวของเฟรม เอวีไอ (Frame AVI Header Format)

เฟรม เอวีไอ นั้นจะมีส่วนการอธิบายข้อมูลต่างๆของแต่ละเฟรมอยู่ทั้งหมด 54 ไบต์ ซึ่งจะมีการอธิบายส่วนต่างๆ ของเฟรมอยู่ทั้งหมด 14 ส่วนดังแสดงในตารางที่ 2

ตำแหน่ง	ขนาด(ไบต์)	คำอธิบาย
0	2	เครื่องหมาย, ซึ่งต้องเป็น 4D42 hex
2	4	ขนาดของไฟล์รูปภาพ มีหน่วยเป็น ไบต์ (อาจเชื่อถือไม่ได้)
6	2	ถูกสงวนไว้, ต้องมีค่าเป็น 0
8	2	ถูกสงวนไว้, ต้องมีค่าเป็น 0
10	4	ตำแหน่งเริ่มต้นของข้อมูลภาพ มีหน่วยเป็นไบต์
14	4	ขนาดของข้อมูลในส่วนหัวของไฟล์ของรูปภาพ มีหน่วยเป็น ไบต์, ต้องมีขนาด 40
18	4	ความกว้างของภาพ มีหน่วยเป็น พิกเซล
22	4	ความสูงของภาพ มีหน่วยเป็น พิกเซล
26	2	จำนวนระนาบ ในภาพ, ต้องเท่ากับ 1
28	2	จำนวนของบิต ต่อ หนึ่งพิกเซล (1,4,8 หรือ 24)
30	4	ประเภทของการบีบอัด (0 = ไม่มีการบีบอัด, 1= อาร์แอลอี -4(RLE-8),2= อาร์แอลอี -4(RLE-4))
34	4	ขนาดของข้อมูล ภาพ มีหน่วยเป็น ไบต์(ทำการประกาศ เพื่อใช้ในการเติม)
38	4	การแสดงผลในแนวนอน มีหน่วยเป็น พิกเซล ต่อ เมตร(อาจไม่มีก็ได้)
42	4	การแสดงผลในแนวตั้ง มีหน่วยเป็น พิกเซล ต่อ เมตร(อาจไม่มีก็ได้)
46	4	จำนวนแถวของตารางสี, อาจมีค่าเป็น 0
50	4	จำนวนสีที่จะนำมาแสดงถ้าฟิลด์นี้มีค่าเป็น 0 แสดงว่าใช้สีทั้งหมด

ตารางที่ 3- 2 รูปแบบส่วนหัวของเฟรม เอวีไอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างคลาสส่วนหัวของดีไอบี:

```
DWORD       biSize;
LONG        biWidth;
LONG        biHeight;
WORD        biPlanes;
DWORD       biBitCount;
DWORD       biCompresstion;
DWORD       biSizeImage;
LONG        biXPelsPerMeter;
LONG        biYPelsPerMeter;
DWORD       biClrUsed;
DWORD       biClrImportant;
```

โดยแต่ละฟิลด์มีรายละเอียดดังนี้

biSize

เป็นฟิลด์ที่ใช้บอกขนาดของ BITMAPINFOHEADER มีหน่วยเป็นไบต์ ขนาด 40 ไบต์

biWidth

เป็นฟิลด์บอกขนาดความกว้างของภาพมีหน่วยเป็น พิกเซล

biHeight

เป็นฟิลด์บอกขนาดความสูงของภาพมีหน่วยเป็น พิกเซล

biPlanes

ฟิลด์นี้จะถูกเซ็ตเป็น 1 เสมอ

biBitCount

เป็นฟิลด์ที่ใช้บอกว่าภาพนี้ใช้กี่บิตต่อพิกเซล [1,4,8,16,24,32]

biCompression

- Uncompressed

  BI_RGB

- Compressed

  BI_RLE8, BI_RLE4, BI_BITFIELDS

BiSizeImage

บอกขนาดจำนวนไบต์ที่ใช้เก็บภาพ

BiXPelsPerMeter

ไม่ใช้ ให้ = 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

biYPelsPerMeter

ไม่ใช้ ให้ = 0

biClrUsed

ฟิลด์บอกจำนวนแถวของตารางสี

biClrImportant

ฟิลด์บอกจำนวนสีที่จะนำมาแสดงถ้าฟิลด์นี้เป็น 0 แสดงว่าใช้สีทั้งหมด

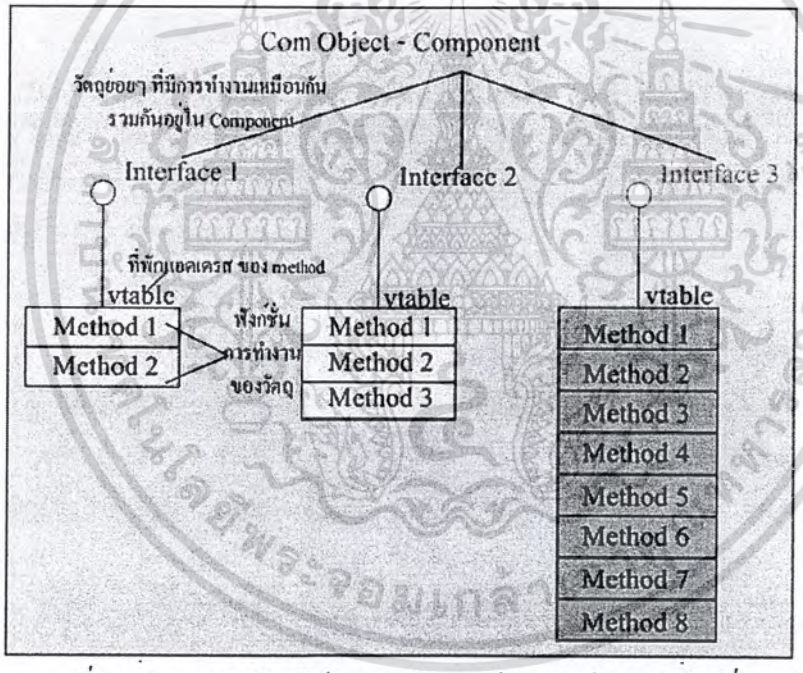
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ทฤษฎีและหลักการคอมโพเน็นท์กับไดเร็กซ์เอ็กซ์

4.1 คอมโพเน็นท์ออบเจ็กท์โมเดล (Component Object Model – COM)

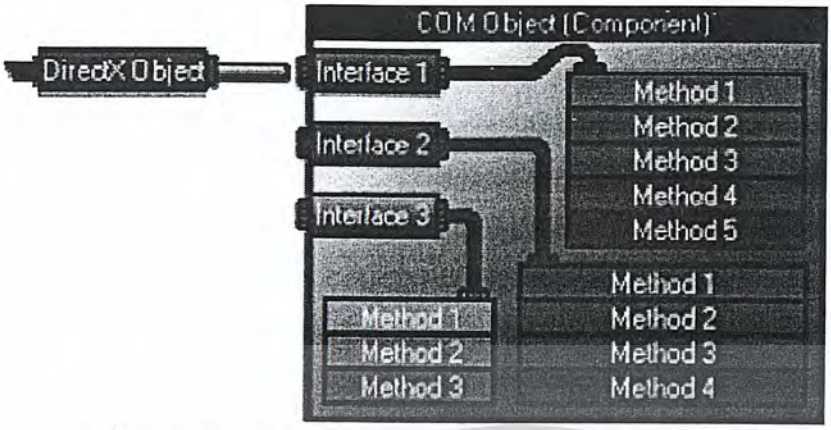
หลักการคอมโพเน็นท์ออบเจ็กท์โมเดลนี้ เป็นวิธีที่ทางไมโครซอฟท์คิดขึ้นมาเพื่อทำให้วัตถุต่างๆ ที่มีความเกี่ยวข้องหรือไม่เกี่ยวข้องกันก็ตาม สามารถติดต่อแลกเปลี่ยนข้อมูลกันได้ และสามารถนำส่วนการใช้งานมาใช้ใหม่ในโปรแกรมที่จะเขียนขึ้นใหม่ได้ โดยไม่ขึ้นกับภาษาที่ใช้เขียน ไม่ขึ้นกับระบบปฏิบัติการ แนวคิดวิธีที่จะทำให้ได้ผลดังกล่าว คือ ใช้หลักการออกแบบ วัตถุ (Object) ให้เป็นคอมโพเน็นท์ คือเป็นการรวมเอาวัตถุ (Object) ย่อยๆ ต่างๆ ที่มีหน้าที่การทำงานในแบบเดียวกันมารวมกันเป็นคอมโพเน็นท์ โดยเรียก วัตถุ (Object) ย่อยๆ เหล่านั้นใหม่ว่าอินเตอร์เฟส (Interface)คอมโพเน็นท์นั้น จะเป็นชุดที่ รวมเอาวัตถุย่อยๆ หรืออินเตอร์เฟส ที่มีฟังก์ชั่นการทำงานในแบบเดียวกัน และในแต่ละอินเตอร์เฟสนั้น จะมีดีไวซ์ตัวหนึ่งที่มีชื่อว่า วีเทเบิล (vtable) สำหรับพักแอดเดรสของฟังก์ชั่น (Method) ตามในรูปที่ 1



รูปที่ 4-1 แสดงความสัมพันธ์ของคอมโพเน็นท์, อินเตอร์เฟสและฟังก์ชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ไดเร็กซ์เอ็กซ์ออบเจ็กท์ กับ คอมโพเน็นท์ออบเจ็กท์



รูปที่ 4-2 แสดงการติดต่อกันระหว่าง COM ออบเจ็กท์ กับ ไดเร็กซ์เอ็กซ์ ออบเจ็กท์

คอมโพเน็นท์ เปรียบเสมือน กล่องที่มีรูหรือช่องไว้ให้เสียบ ซึ่งอาจจะมีหลายๆ รู แต่ละรูเปรียบเสมือนอินเตอร์เฟส โดยแต่ละอินเตอร์เฟส ก็จะมีฟังก์ชั่นการทำงาน (Method) สำหรับไดเร็กซ์เอ็กซ์ออบเจ็กท์ (DirectX Object) นั้น ก็เปรียบเสมือนแจ็คที่สามารถเสียบที่รูหรืออินเตอร์เฟสได้ หากวัตถุของไดเร็กเอ็กซ์ เสียบเข้าที่อินเตอร์เฟสใด ก็สามารถเรียกใช้งานฟังก์ชั่นการทำงานในอินเตอร์เฟสนั้นๆได้ หากต้องการใช้งานฟังก์ชั่นการทำงานอื่น ที่อยู่ในอินเตอร์เฟสอื่นๆ ก็ต้องสร้างไดเร็กซ์เอ็กซ์ออบเจ็กท์ขึ้นมาใหม่ เพื่อนำไปเสียบกับอินเตอร์เฟสที่จะใช้งานฟังก์ชั่นการ และถ้าเลิกใช้อินเตอร์เฟสใดก็ต้องถอดแจ็คหรือวัตถุนั้นออก

กล่าวได้ว่า อินเตอร์เฟสแต่ละอัน ก็จะมีฟังก์ชั่นการทำงานต่างๆ ของตัวเอง หากต้องการจะใช้งานฟังก์ชั่นการทำงาน จะต้องสร้างไดเร็กซ์เอ็กซ์ออบเจ็กท์ที่ชี้ไปยังอินเตอร์เฟสที่มีฟังก์ชั่นการทำงานที่ต้องการ โดยการสร้างไดเร็กซ์เอ็กซ์ออบเจ็กท์นี้ สามารถทำได้โดยการเรียกใช้ฟังก์ชั่นที่ไดเร็กซ์เอ็กซ์เตรียมไว้ให้

การปรับปรุงความสามารถของวัตถุประเภท COM จะใช้การสร้างอินเตอร์เฟสตัวใหม่ขึ้นมา เพื่อใช้รองรับความสามารถใหม่มากกว่าจะเปลี่ยนแปลงฟังก์ชั่นที่มีอยู่ในอินเตอร์เฟสปัจจุบัน

4.3 อินเตอร์เฟสของ COM ออบเจ็กท์

ทุก COM ออบเจ็กท์ จะสนับสนุนอินเตอร์เฟสที่ชื่อ IUnknown อินเตอร์เฟสนี้เป็นอินเตอร์เฟสมาตรฐาน ซึ่งคอยควบคุมความสามารถในการเข้าใช้งานของไดเร็กซ์เอ็กซ์ออบเจ็กท์ ที่เข้ามาใช้งานอินเตอร์เฟสต่างๆ โดยทุกอินเตอร์เฟสจะสืบทอดมาจากคลาส IUnknown ซึ่งเป็นคลาสที่ใช้สร้าง IUnknown อินเตอร์เฟส

สำหรับ IUnknown อินเตอร์เฟสนี้จะมีเมธอดอยู่ 3 ตัวด้วยกันคือ AddRef(), QueryInterface()และ Release () ดังนั้นอินเตอร์เฟสตัวอื่นๆ ที่สืบทอดไปจึงมีเมธอดทั้ง 3 นี้ด้วยเสมอ สังเกตได้จาก ไฟล์ Header สำหรับใช้งานอินเตอร์เฟส เช่น C:\DXSDK\include\dinput.h ในบรรทัดที่มีการใช้ #define INTERFACE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะมีเมธอดทั้ง 3 ที่สืบทอดมาจาก IUnknow อยู่บรรทัดแรกๆ ซึ่ง COM ออบเจ็กท์ จะใช้ IUnknown เป็นตัวจัดการไดเร็กซ์เอ็กซ์ออบเจ็กท์ที่เข้ามาใช้อินเตอร์เฟส ที่อยู่ใน COM ออบเจ็กท์นั้นๆ

• AddRef() จะทำการเพิ่มค่าการนับแก่ reference count ของออบเจ็กท์ ไปอีก 1 เมื่ออินเตอร์เฟสหรือโปรแกรมประยุกต์อื่นๆ เข้าใช้งานออบเจ็กท์นั้นๆ

• QueryInterface() ตรวจดูว่าออบเจ็กท์นั้นมีอินเตอร์เฟสที่ต้องการหรือไม่ และยังสามารถใช้สร้างไดเร็กซ์เอ็กซ์ออบเจ็กท์ตัวใหม่ จากไดเร็กซ์เอ็กซ์ออบเจ็กท์เดิมที่มีอยู่ได้ด้วย โดยจะไปสอบถาม COM ออบเจ็กท์ ซึ่งเป็นเจ้าของอินเตอร์เฟสที่ไดเร็กซ์เอ็กซ์ออบเจ็กท์ตัวเดิมกำลังใช้อยู่ว่า มีอินเตอร์เฟสที่ไดเร็กซ์เอ็กซ์ออบเจ็กท์ตัวใหม่ต้องการหรือไม่

• Release() ลดค่าการนับแก่ reference count ของออบเจ็กท์ ลงไป 1 เมื่อลดลงไปถึง 0 ออบเจ็กท์จะถูกทำลาย

ทุกครั้งที่สร้างไดเร็กซ์เอ็กซ์ออบเจ็กท์ ซึ่งก็คือ การต้องการจะติดต่อกับอินเตอร์เฟสตัวใดในคอมโพเน็นท์ของไดเร็กซ์เอ็กซ์ โดย COM ออบเจ็กท์ จะเรียก QueryInterface() เพื่อตรวจดูว่ามีอินเตอร์เฟสที่ต้องการใช้มีอยู่ในออบเจ็กท์หรือไม่ หากมี จะเตรียมพื้นที่ในหน่วยความจำแล้วส่งตำแหน่งแอดเดรสกลับมา พร้อมทั้งเรียกใช้ AddRef() เพื่อเพิ่มการนับแก่ Reference Count อีก 1 ถ้ามีไดเร็กซ์เอ็กซ์ออบเจ็กท์ตัวอื่นๆ มาใช้อินเตอร์เฟสอื่นๆ หรือใช้อินเตอร์เฟสซ้ำกันกับไดเร็กซ์เอ็กซ์ออบเจ็กท์ตัวอื่น ReferenceCount ก็จะถูกบวกค่าเพิ่มขึ้นไปเรื่อยๆ โดยหน่วยความจำที่ไดเร็กซ์เอ็กซ์ออบเจ็กท์ใช้นี้จะยังคงอยู่ตลอด หากไม่ทำการเรียกใช้Release() แม้จะปิดโปรแกรมไปแล้วก็ตาม ซึ่งทำให้เกิดปัญหาเรื่องทรัพยากรตอบสนองไม่เพียงพอ หรือLow resource ดังนั้น จะต้องเรียกใช้ Release() เมื่อไม่ต้องการใช้งานแล้วเสมอ เพื่อเป็นการคืนหน่วยความจำ และลดค่า Reference Count ทีละ 1 เมื่อลดจนเหลือ 0 ก็แสดงว่าไม่มีไดเร็กซ์เอ็กซ์ออบเจ็กท์ตัวใด ใช้ COM ออบเจ็กท์นั้นแล้ว กลไกภายใน COM ก็จะลบ COM ออบเจ็กท์ ตัวนั้นออกไปจากหน่วยความจำโดยอัตโนมัติ

4.4 ไดเร็กซ์เอ็กซ์ 9 (DirectX 9)

4.4.1 ความรู้เบื้องต้นเกี่ยวกับไดเร็กซ์เอ็กซ์

ไมโครซอฟท์ ไดเร็กเอ็กซ์(Microsoft DirectX) เป็นชุดของแอพพลิเคชั่นโปรแกรมมิ่งอินเตอร์เฟส (Application Programming Interface – API) ได้รับการพัฒนาและออกแบบเพื่อใช้จัดเตรียมอินเตอร์เฟส สำหรับควบคุมฮาร์ดแวร์ทางด้านมัลติมีเดีย บนระบบปฏิบัติการ ไมโครซอฟท์ วินโดว์ (Microsoft Windows) ได้อย่างมีประสิทธิภาพ และเป็นเครื่องให้โปรแกรมเมอร์ใช้คำสั่งควบคุมฮาร์ดแวร์ได้อย่างใกล้ชิด โดยไม่ต้องยุ่งยากกับการสร้างส่วนโปรแกรมติดต่อในระดับล่าง ซึ่งเครื่องมือที่ใช้ติดต่อจะแตกต่างกันไปตามประเภทของอุปกรณ์ทำให้โปรแกรมเมอร์สร้างซอฟต์แวร์เพื่อทำงานทางด้านมัลติมีเดียได้เป็นอย่างดี เนื่องจากการติดต่อในแต่ละคอมโพเน็นท์จะเป็นของอุปกรณ์ประเภทหนึ่งๆ เป็นอิสระในการเขียนโปรแกรม

API ของ ไดเร็กซ์เอ็กซ์นั้น สร้างขึ้นมาบนรากฐานของ HAL (Hardware Abstraction Layer) สามารถซ่อนลักษณะของดีไวซ์ (Device) ที่เกี่ยวพันกับฮาร์ดแวร์และออกแบบมาเพื่อให้สามารถรองรับความขยายตัวของฮาร์ดแวร์ใหม่ๆ ที่จะออกมาในอนาคต ดังนั้นจึงสามารถรองรับความสามารถของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฮาร์ดแวร์ที่มีความสามารถเร่งความเร็วใหม่ๆ ที่ปัจจุบันยังไม่มี ด้วยความสามารถในการจำลองการทำงานผ่าน HEL (Hardware Emulation Layer) หรือเลือกที่จะหลีกเลี่ยงไม่ใช้ความสามารถนี้ ถ้า HEL ไม่สนับสนุน ทำให้โปรแกรมเมอร์พัฒนางานทางด้านมัลติมีเดียง่ายขึ้น ไม่จำเป็นต้องทดสอบกับอุปกรณ์ทุกตัวในท้องตลาด เพียงแต่ทดสอบกับไดเร็กซ์เอ็กซ์ก็เพียงพอแล้ว สำหรับผู้สร้างอุปกรณ์ฮาร์ดแวร์ ก็ต้องสร้างไดรเวอร์ของอุปกรณ์นั้นๆ เพื่อให้สามารถทำงานเข้ากับไดเร็กซ์เอ็กซ์ได้

4.4.2 ชุดพัฒนาซอฟต์แวร์สำหรับไดเร็กซ์เอ็กซ์ (DirectX software Development Kit: DirectX SDK)

ชุดพัฒนาซอฟต์แวร์สำหรับไดเร็กซ์เอ็กซ์ นี้ประกอบด้วยชุดของไลบรารี่ ไฟล์ DLL ไฟล์ header รวมไปถึงเอกสาร และตัวอย่างโปรแกรม ซึ่งช่วยในการพัฒนาซอฟต์แวร์ทางด้านมัลติมีเดีย ซึ่งการที่จะใช้ไดเร็กซ์เอ็กซ์ได้นั้น ก็ต้องทำการลงชุดพัฒนาซอฟต์แวร์สำหรับไดเร็กซ์เอ็กซ์นี้ก่อน

4.4.3 คอมโพเน็นท์ในไดเร็กซ์เอ็กซ์

ไดเร็กซ์เอ็กซ์ 9 (DirectX 9) จะประกอบด้วย Component 6 ตัว คือ

4.4.3.1 DirectX Graphics

DirectX Graphics เป็นคอมโพเน็นท์ใหม่ที่รวมคอมโพเน็นท์เดิมที่ชื่อ DirectDraw กับ Direct3D ไปเป็นอินเตอร์เฟสอันเดียวคือ Direct3D Interface ซึ่งอยู่ในคอมโพเน็นท์ DirectX Graphics นี้ ใช้จัดการทางด้านกราฟฟิกการแสดงรูปภาพ 2 มิติ และวัตถุ 3 มิติ ซึ่งช่วยจัดการการใช้อุปกรณ์แสดงผล 3 มิติ ในรูปแบบที่ไม่ขึ้นกับชนิดของอุปกรณ์ที่จะใช้ (Device-Independent)

4.4.3.2 DirectX Audio

DirectX Audio เป็นคอมโพเน็นท์ใหม่ที่รวมคอมโพเน็นท์เดิมที่ชื่อ DirectSound กับ DirectMusic มาเป็นอินเตอร์เฟสในคอมโพเน็นท์ DirectX Audio นี้ ใช้จัดการทางด้านเสียง สนับสนุนการเร่งความเร็วทางด้านฮาร์ดแวร์ (acceleration) ในการสร้างเสียงแบบไดนามิค (Dynamic soundtrack) และเอฟเฟ็คเสียง 3 มิติขั้นสูง (Advanced 3D position effect)

4.4.3.3 DirectInput

DirectInput ใช้จัดการทางด้านการรับข้อมูลจากอุปกรณ์อินพุต เช่นจาก คีย์บอร์ด, เมาส์, จอยสติก เป็นต้น เพื่อเข้าถึงข้อมูลการติดต่อโดยตรงกับฮาร์ดแวร์ด้วยฮาร์ดแวร์ไดรเวอร์ (Hardware Driver) ซึ่งจะทำงานได้เร็วกว่าการติดต่อด้วยเมสเสจของวินโดว์ นอกจากนี้ยังสนับสนุนการทำงานกับอุปกรณ์แบบแรงตอบสนอง (Force Feedback) อีกด้วย

4.4.3.4 DirectPlay

DirectPlay เป็นมิเดียที่เป็นอิสระบนเน็ตเวิร์ค ซึ่งหมายถึง สามารถรันได้บนเครือข่ายแบบTCP/IP เครือข่ายแบบ IPX หรือแม้แต่การต่อตรงจากโมเด็ม และการเป็นอิสระทางเน็ตเวิร์คนี้จะสามารถถูกพัฒนาต่อไปได้ในอนาคต นั่นคือ สามารถรองรับอุปกรณ์และโปรโตคอลใหม่ๆ เช่น IPV6, multicastโดยปกติ DirectPlay ใช้จัดการการทำงานด้านเน็ตเวิร์ค โดยไม่จำเป็นต้องไปยุ่งกับการทำงานในระดับล่าง มีหน้าที่การทำงานสำหรับการส่งข้อมูลทีเดียวให้หลายๆ เครื่องเป็นกรุ๊ปได้ สามารถเขียนโปรแกรมติดต่อทางด้านเน็ตเวิร์ค ทั้งในการติดต่อผ่านเน็ตเวิร์คระหว่างโปรแกรมแบบมัลติเพลเยอร์ (Multiplayer) แบบเพียร์ทูเพียร์ (Peer-to-Peer) และแบบไคลเอ็นท์/เซิร์ฟเวอร์ (Client/Server)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.3.5 DirectShow

DirectShow ใช้สำหรับการทำงานด้านมัลติมีเดีย เป็นสถาปัตยกรรมทางด้านมีเดียสตรีมมิ่ง (media-streaming) ของไมโครซอฟท์วินโดว์ ที่มีความสามารถในการแคปเจอร์ (Capture) สูง สามารถบรรจุข้อมูลที่เป็นวิดีโอ และการบีบอัดข้อมูลไฟล์ (Audio data compressed) ได้หลายรูปแบบ เช่น การเล่นไฟล์วีดีโอ ประเภท MPEG, Audio-Video (AVI), MPEG-1 Layer3 (MP3) และไฟล์นามสกุล .wav เป็นต้น

4.4.3.6 DirectSetup

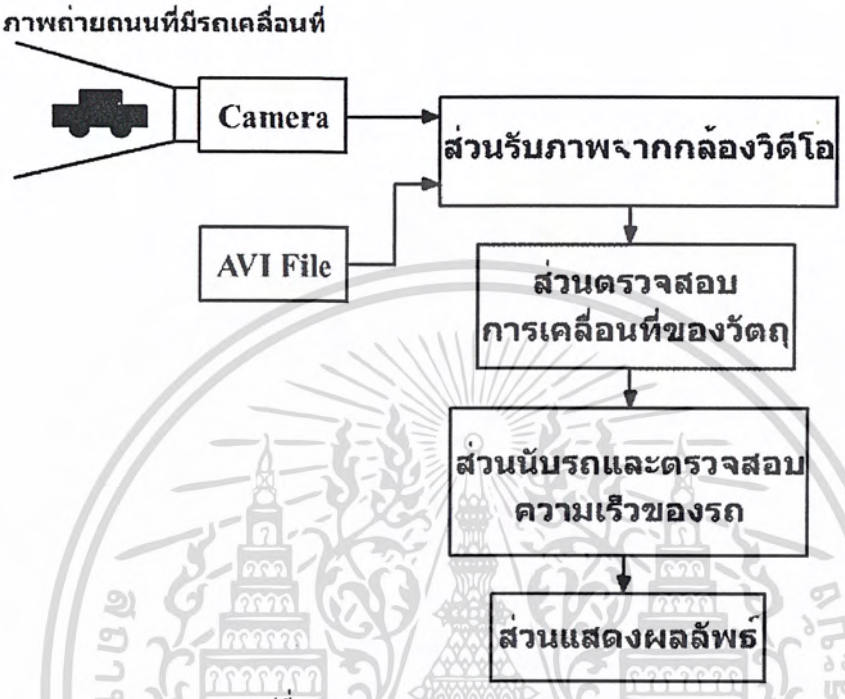
DirectSetup ใช้สำหรับการจัดการทางด้านการติดตั้งองค์ประกอบที่จะต้องใช้ในงานไดเร็กซ์เอ็กซ์บนเครื่องคอมพิวเตอร์ของผู้ใช้อย่างอัตโนมัติ หากเครื่องคอมพิวเตอร์นั้นยังไม่ได้ทำการติดตั้ง

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
โครงสร้างและการออกแบบ

5.1 ภาพรวมของระบบ



รูปที่ 5-1 แสดงภาพรวมของระบบ

จากรูปที่ 5-1 สามารถแบ่งภาพรวมของโครงงานระบบตรวจสอบสภาพการจราจรเป็นส่วนย่อยๆ ได้ดังนี้

1.ส่วนของกล้องวิดีโอและภาพที่ได้รับการบันทึกไว้แล้ว

2.ส่วนรับภาพจากล้องวิดีโอ เป็นตัวแคปเจอร์ (Capture) ภาพเพื่อนำภาพไปใช้ในส่วนถัดไป

3.ส่วนการตรวจสอบการเคลื่อนที่ของวัตถุ นำภาพที่ได้แคปเจอร์มาตรวจสอบหาส่วนที่เป็นรถภายในรูปภาพ

4.ส่วนนับรถและตรวจสอบความเร็วของรถ นำภาพที่ได้จากข้อที่ 3 มาทำการประมวลผลตามอัลกอริธึมที่ได้ศึกษามา

5.ส่วนแสดงผลลัพธ์ จะแสดงตัวเลขของจำนวนรถและแสดงความเร็วของรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 คุณสมบัติของโปรแกรมระบบตรวจสอบสภาพการจราจร

1.ระบบจะต้องมีกล้อง 1 ตัว

2.กล้องจะถูกติดตั้งอยู่ทางด้านบนของถนน เช่น บนสะพานลอยหรือบนทางต่างระดับ

3.กล้องจะทำมุมประมาณ 45 องศา กับพื้นถนนหรือปรับตามเหมาะสมขึ้นอยู่กับสภาพแวดล้อมที่ทำการตรวจสอบ

4.เมื่อทำการปรับตั้งค่าตัวแปรต่างๆเป็นที่เรียบร้อยแล้ว กล้องจะไม่สามารถเคลื่อนไหวได้

5.ก่อนการใช้งานจริง จะมีการปรับตั้งค่าตัวแปรต่างๆ เช่น กำหนดอาณาเขตของถนน

6.อาณาเขตของถนนที่กำหนดเพื่อทำการตรวจสอบสภาพการจราจรจะต้องตีกรอบเฉพาะบริเวณที่รถวิ่งไปในทิศทางเดียวกันและเป็นทิศทางที่เคลื่อนที่เข้าหากล้องเท่านั้น

7.สามารถทำงานได้ในเวลากลางวันที่มี สภาพอากาศที่ค่อนข้างดี

8.กล้องสามารถปรับความชัดแบบอัตโนมัติ

9.ข้อมูลที่รับเข้ามาจะต้องเป็นภาพที่ต่อเนื่อง ที่ได้มาจากกล้องวิดีโอหรือจากไฟล์ภาพวิดีโอ

10.การแสดงผลจะอยู่ในรูปแบบของตัวเลข โดยระบุจำนวนรถ ความเร็ว และความหนาแน่นของรถ

11.ไฟล์ภาพที่รับเข้ามาจะต้องมีนามสกุล AVI เท่านั้น

12.ขนาดของภาพที่รับเข้ามาจาก ไฟล์ AVI และกล้องวิดีโอจะต้องมีขนาด 320*240 พิกเซล เท่านั้น

13.การตรวจสอบทำได้เฉพาะรถที่มีการเคลื่อนที่เท่านั้น

5.3 ระบบคอมพิวเตอร์

ระบบคอมพิวเตอร์ที่สนับสนุนความต้องการขั้นต่ำของระบบ มีดังต่อไปนี้

CPU	Intel, AMD หรืออื่นๆ ความเร็วอย่างต่ำ 1.7 GHz
RAM	256 MB
HARDDISK	20 GB
OS	Windows XP
อื่นๆ	DirectX 9.0,การเชื่อมต่อผ่านพอร์ต USB, Driver ของ Web cam

ตารางที่ 5-1 แสดงระบบคอมพิวเตอร์ที่โปรแกรมต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 เครื่องมือที่ใช้ในการพัฒนา มีดังนี้

5.4.1 ไมโครซอฟท์ วิชวล ซีพลัสพลัส ดอทเน็ท (Microsoft Visual C++.Net)

ไมโครซอฟท์ วิชวล ซีพลัสพลัส ดอทเน็ท เป็นเครื่องมือที่ใช้ในการพัฒนาโปรแกรมบนระบบปฏิบัติการวินโดว์ เป็นเครื่องมือที่พัฒนาโดยบริษัทไมโครซอฟท์ และได้รับความนิยมจากนักพัฒนาเป็นจำนวนมาก โดยมีตัวคอมไพล์และดีบักที่ใช้งานง่ายและสะดวก มีเครื่องมือในการเรียกใช้ในลักษณะคอมโพเน็นท์ต่างๆ มากมายในการเขียนโปรแกรม ใช้พื้นฐานภาษา ซีพลัสพลัส (C++) ซึ่งเป็นพื้นฐานการเขียนโปรแกรม โดยใช้หลักการเชิงวัตถุ (Object-Oriented Programming: OOP) ทำให้ประหยัดเวลาในการพัฒนาลงไปได้มาก มีคอมโพเน็นท์ให้เรียกใช้ ซึ่งสามารถศึกษาและทำการเขียนโปรแกรมควบคุมการทำงานเพิ่มเพื่อพัฒนาต่อได้

5.4.2 ไมโครซอฟท์ ไดเร็กซ์เอ็กซ์ 9 เอสดีเค (Microsoft DirectX9 SDK)

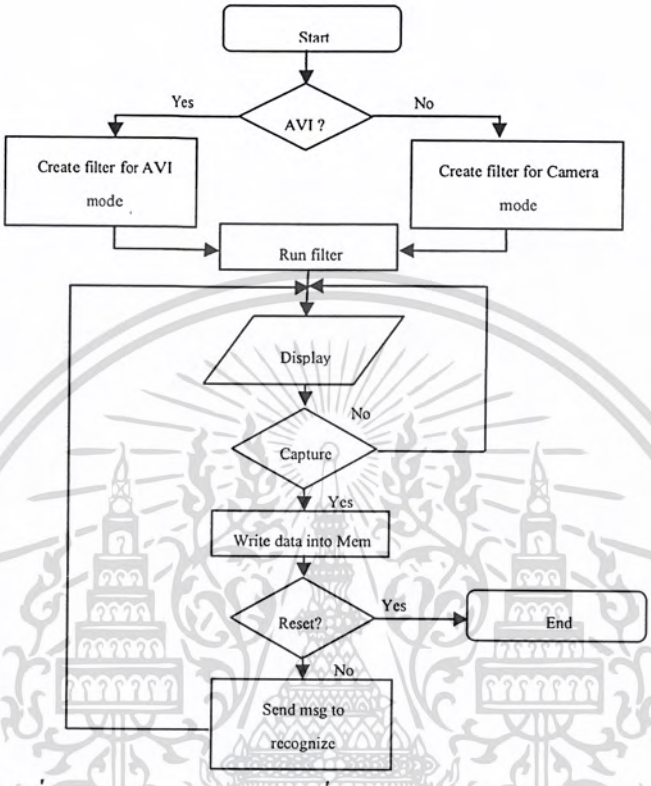
ไมโครซอฟท์ไดเร็กซ์เอ็กซ์ 9 เอสดีเค (Microsoft DirectX9 SDK (Software Development Kit)) เป็นเครื่องมือที่ใช้ในการพัฒนาโปรแกรมทางด้านมัลติมีเดีย เป็นเครื่องมือที่พัฒนาขึ้นโดยบริษัทไมโครซอฟท์ ซึ่งมีความสามารถในการพัฒนาของเอพีไอ (application programming interfaces: APIs) สำหรับการสร้างเกมหรือโปรแกรมอื่นๆ ที่ต้องการความสามารถทางมัลติมีเดียสูง การพัฒนาโปรแกรมทางด้านมัลติมีเดียจะทำได้โดยง่ายขึ้น โดยไมโครซอฟท์ไดเร็กซ์เอ็กซ์ 9 เอสดีเค สนับสนุนการสร้างโปรแกรมทางด้านกราฟิกทั้งสองมิติและสามมิติ เสียง ดนตรี การรับข้อมูลจากอุปกรณ์อินพุต รวมถึง การทำงานเป็นระบบผู้ใช้งานหลายๆ คน บนเครือข่ายเน็ตเวิร์คการพัฒนา ไมโครซอฟท์ไดเร็กซ์เอ็กซ์ 9 เอสดีเค ได้ถูกพัฒนาขึ้นจนมาถึงปัจจุบันเวอร์ชั่น 9 แต่ข้อมูลในการนำมาอ้างอิงยังไม่สมบูรณ์นัก เวอร์ชั่น 9 มีอินเตอร์เฟสใหม่ๆ เพิ่มขึ้นและเปลี่ยนแปลงโครงสร้างจากเวอร์ชั่นเก่ามาก เพื่อตอบสนองความต้องการในการพัฒนาให้ครบรูปแบบมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 การออกแบบโปรแกรมระบบตรวจสอบสภาพการจราจร

ในโปรแกรมมีการออกแบบให้แบ่งการทำงานออกเป็น 3 ส่วนหลักๆ ดังนี้

5.5.1 ส่วนที่ทำการรับอินพุตระบบ



รูปที่ 5-2 แสดง Flow chart ส่วนที่ทำการรับอินพุตระบบ

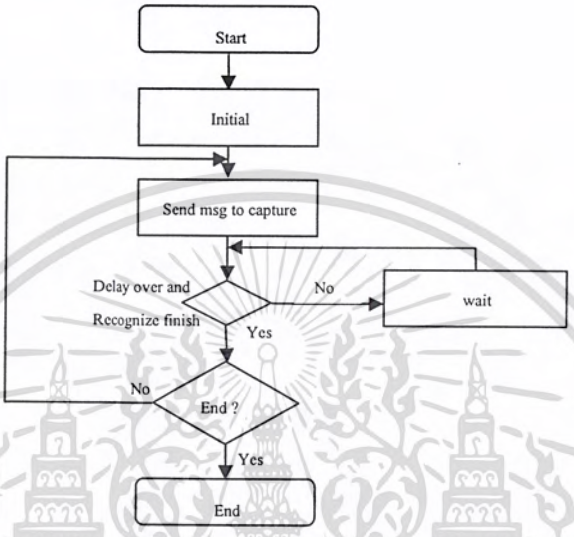
มีขบวนการทำงานดังนี้

1. ผู้ใช้ต้องทำการเลือกว่าจะให้ระบบรับอินพุตประเภทไหนโดยสามารถเลือกได้ 2 ประเภท
    - จากไฟล์นามสกุล AVI ในออฟไลน์โหมด (Offline mode)
    - จากกล้องวิดีโอในเรียลไทม์โหมด (Real time mode)
2. ระบบทำการสร้างการติดต่อ เพื่อรองรับกับ อินพุตของระบบที่ผู้ใช้ได้ทำการระบุในขั้นต้น
3. แสดงภาพอินพุตของระบบที่ไดอะล็อกของระบบ
4. ระบบทำการตรวจสอบว่ามีเมสเสจส่งมาเพื่อให้ทำการแคปเจอร์ภาพ ณ. เวลานั้นเพื่อนำไปสู่การประมวลผลหรือไม่
    - หากมีความต้องการให้มีการแคปเจอร์ภาพ
        - ทำการแคปเจอร์ภาพ
        - เขียนข้อมูลลงหน่วยความจำเพื่อรอการนำไปประมวลผลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการส่งเมสเสจไปให้ ส่วนของการประมวลผลภาพ ให้สามารถมาอ่านข้อมูลและทำการประมวลผลภาพตามลำดับ
- หากไม่มีความต้องการให้มีการแคปเจอร์ภาพ จะกลับไปทำดังข้อที่ 3

5.5.2 ส่วนของการควบคุมการทำงานทั้งหมดของระบบ

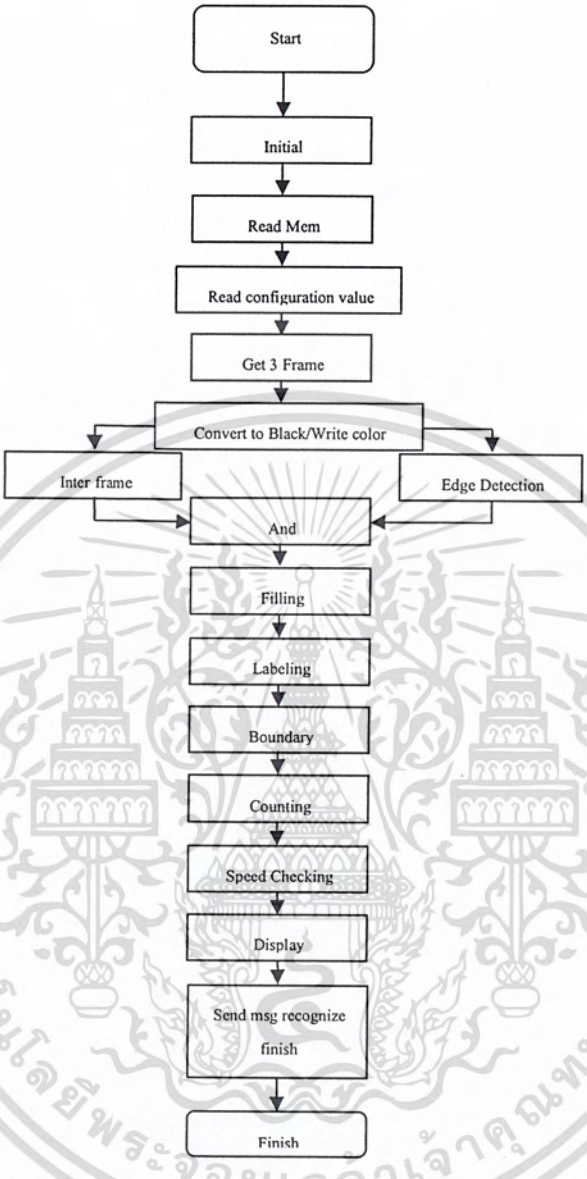


รูปที่ 5-3 แสดง Flow chart ส่วนของการควบคุมการทำงานทั้งหมดของระบบ

ในส่วนนี้จะคอยดูแลควบคุมในส่วนของ Frame Rate โดยจะมีหน้าที่หลักคือคอยดูแลเรื่องเวลาและส่งเมสเสจไปควบคุมให้ส่วนของการติดต่ออินพุตให้มีการแคปเจอร์ภาพ หากผู้ใช้มีการปรับค่า Frame Rate ที่สูงเกินไปจนระบบไม่สามารถประมวลผลได้ทัน ระบบจะมีการเพิ่มค่า ดีเลย์ให้เพื่อให้ระบบทำงานได้ทัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.3 ส่วนของการ Recognize ภาพ



รูปที่ 5-4 แสดง Flow chart **ส่วนของการ Recognize ภาพ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อส่วนของการติดต่อกับอินพุตของระบบได้ทำการแคปเจอร์เป็นที่เรียบร้อยแล้ว จะมีการส่งเมสเสจมาเรียกให้ส่วนนี้ทำงานโดยมีหลักการทำงานดังนี้

1.ระบบทำการอ่านข้อมูลภาพขึ้นมาจากหน่วยความจำ

2.แปลงภาพที่รับเข้ามาให้เป็นภาพขาวดำ โดยแปลงจาก RGB color model มาอยู่ใน YIQ color model แล้วนำค่า Y ที่เป็นค่าลูมิแนนซ์ไปใช้งาน

3.เมื่อได้ภาพที่เป็นขาวดำเรียบร้อยแล้ว จะนำไปสู่ขบวนการประมวลผลภาพ เพื่อจัดการหาวัตถุที่กำลังเคลื่อนที่ภายในภาพ สำหรับขั้นตอนการตรวจจับวัตถุที่กำลังเคลื่อนที่มีดังต่อไปนี้

3.1 การทำ Inter-frames differences

3.2 การทำ Sobel-Edge detection

3.3 การทำ Moving-Edge detection

3.4 การทำ filling

3.5 การทำ Labeling

3.6 การทำ Boundary

4.นำภาพที่สามารถตรวจจับวัตถุที่กำลังเคลื่อนที่เรียบร้อยแล้วไปตรวจนับจำนวนรถ

5.นำภาพที่สามารถตรวจจับวัตถุที่กำลังเคลื่อนที่เรียบร้อยแล้วไปตรวจสอบความเร็วของรถ

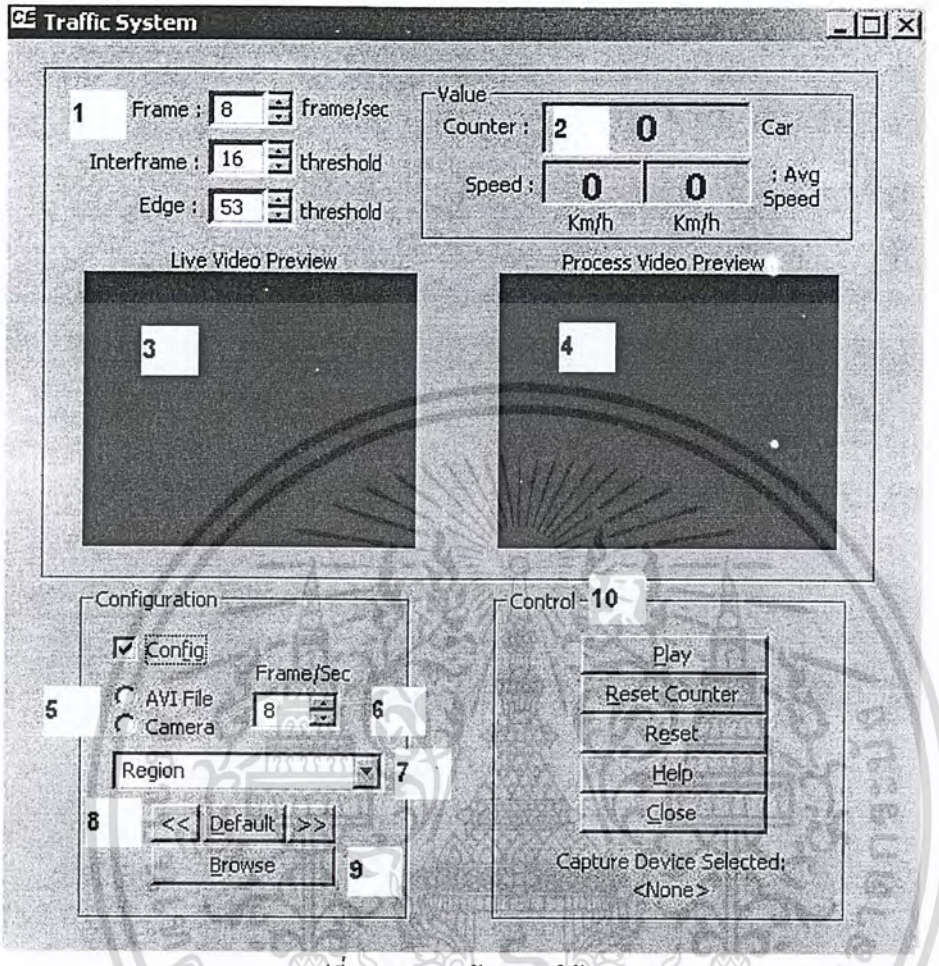
6. แสดงผลลัพธ์การทำงานของระบบตรวจสอบสภาพการจราจร

7. ระบบทำการส่งเมสเสจไปบอกกับส่วนควบคุมการทำงานของระบบว่าทำการประมวลผลภาพ ณ ช่วงเวลานั้นเป็นที่เรียบร้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.4 การออกแบบหน้าจอ

หน้าจอโปรแกรมของระบบแสดงดังรูปที่ 5-6



รูปที่ 5-5 แสดงหน้าจอการใช้งาน

รายละเอียดของหน้าจอมีดังนี้

1.ส่วนของการปรับแต่งค่า Frames rate ของการแคปเจอร์ภาพและ Thresholds ของอัลกอริธึม Inter-frame กับ Edge-detection ในโหมด Configuration

2.ส่วนของการแสดงผลลัพธ์ของโปรแกรม ประกอบไปด้วยจำนวนรถ ความเร็วของรถและความเร็วเฉลี่ยของรถ

3.ส่วนแสดงภาพวิดีโอสดที่ได้มาจากกล้องวิดีโอและไฟล์นามสกุล AVI ที่เราได้บันทึกไว้

4.ส่วนแสดงภาพสดที่ผ่านการประมวลผลของโปรแกรม และเป็นส่วนที่ใช้ในการปรับแต่งค่าเส้นตรวจนับรถและเส้นตรวจจับความเร็วของรถด้วย

5.เลือกการรับภาพอินพุทจากกล้องวิดีโอหรือไฟล์นามสกุล AVI ที่ได้บันทึกไว้

6.ใช้ปรับค่า Frames rate ของการแคปเจอร์มาประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.ใช้เลือกการปรับแต่งค่าเส้นตรวจนับรถ เส้นตรวจจับความเร็วรถ ปรับค่า Thresholds ของการประมวลผลของอัลกอริธึมต่างๆ และยังใช้เลือกดูค่าผลลัพธ์ของภาพที่ผ่านขบวนการประมวลผลภาพ

8.ใช้สำหรับปรับค่า Thresholds กลับไปเป็นค่ามาตรฐานที่ระบบได้กำหนดไว้

9.ใช้เลือกไฟล์ไดเร็คทอรี่ที่เก็บไฟล์นามสกุล AVI ไว้ ในกรณีที่รับอินพุทจากไฟล์นามสกุล AVI ที่ได้บันทึกไว้

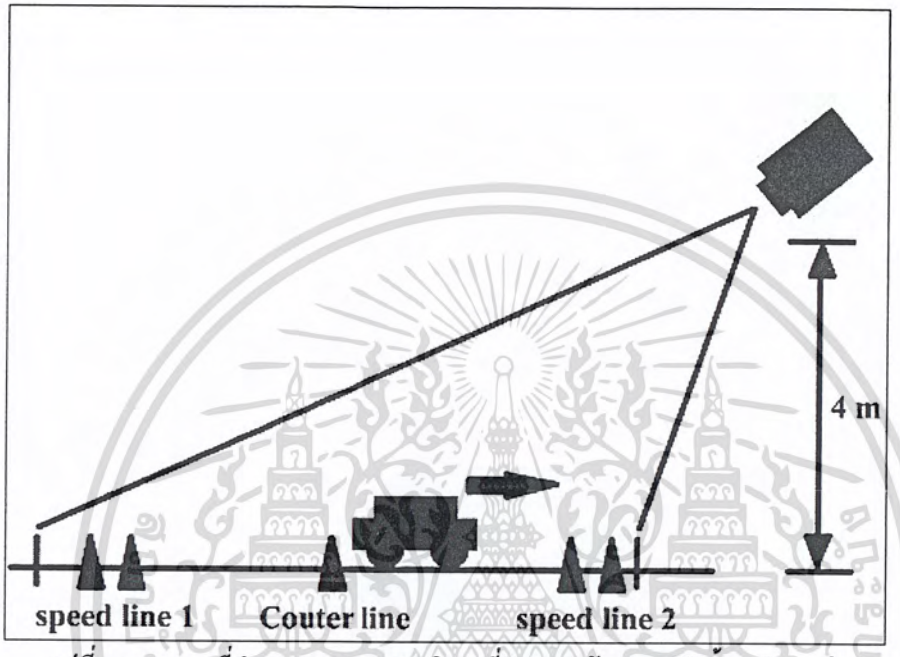
10.ใช้ควบคุมการทำงานของระบบประกอบไปด้วยปุ่ม Play ใช้สำหรับรันการทำงานโปรแกรม ปุ่ม Reset counter ให้เริ่มนับจำนวนรถใหม่ ปุ่ม Reset ใช้ยกเลิกการทำงานของระบบ ปุ่ม Help เพื่อเรียกตัวช่วยในการใช้งานโปรแกรมและปุ่ม Close ใช้ปิดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

ผลการทดลอง/การทดสอบ

ในการทดลองนี้ได้ใช้เครื่อง Pentium IV 2 GHz มีหน่วยความจำหลัก 256 MB ข้อมูลขาเข้า (Input) ได้จากการใช้กล้องบันทึกภาพรถที่มีความเร็วตั้งแต่ 30-150 กิโลเมตรต่อชั่วโมง โดยถ่ายภาพดังกล่าวบนสะพานลอยมีมุมกดดังรูปที่ 6-1

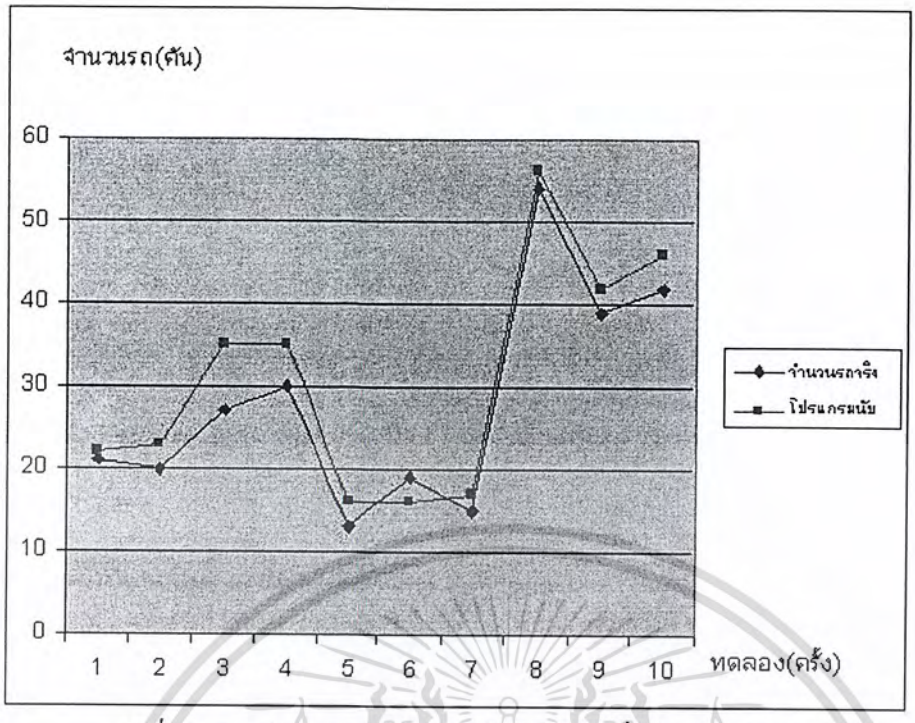


รูปที่ 6-1 มุมมองที่ทำการทดลอง และบริเวณที่กำหนดเส้นตรวจจับทั้ง 2 ตำแหน่ง

6.1 การหาจำนวนรถจากระบบตรวจสอบสภาพการจราจร

การทดสอบการนับรถด้วยระบบตรวจสอบสภาพการจราจรที่ได้สร้างขึ้นเปรียบเทียบกับจำนวนรถจริงๆ สามารถบ่งบอกถึงประสิทธิภาพของระบบได้ว่ามีความแม่นยำเพียงไร กราฟรูปที่ 6-2 ได้จากผลการทดลอง โดยมีการตั้งค่า Thresholds ของ Inter-frames เท่ากับ 30 ค่า Thresholds ของค่า Sobel-Edge detection เท่ากับ 140 ความเร็วของรถวิ่งไม่เกิน 150 กิโลเมตรต่อชั่วโมง สภาพแวดล้อมของถนนอยู่ในสภาพที่ดี จากผลการทดลองพบว่า มีค่าความผิดพลาดของการนับรถเพียงเล็กน้อยเท่านั้น นับว่าสามารถนับรถได้ถูกต้องอยู่ในระดับที่น่าพอใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



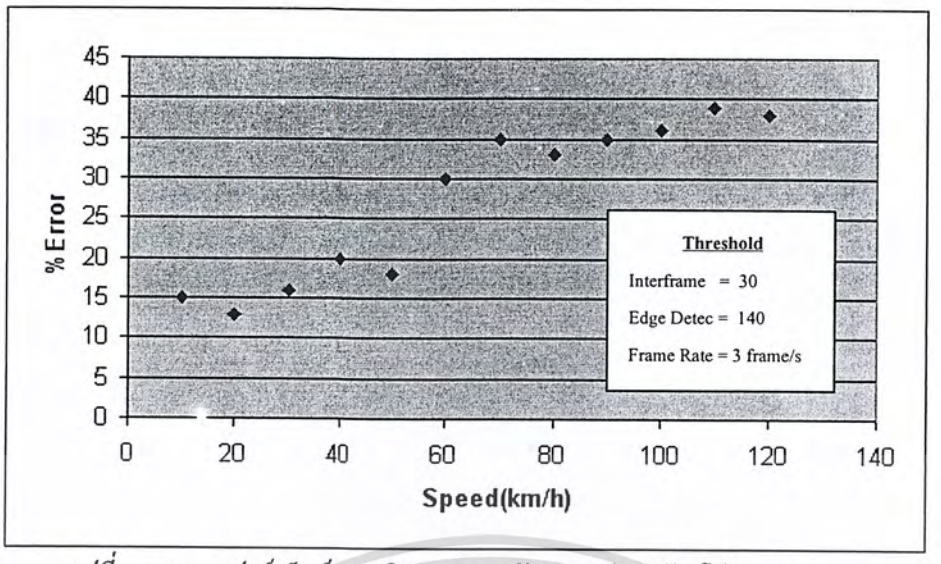
รูปที่ 6-2 แสดงจำนวนรถ จำนวนรถจริงกับจำนวนที่ได้จากโปรแกรม

6.2 การหาความเร็วรถที่ได้จากระบบตรวจสอบสภาพการจราจร

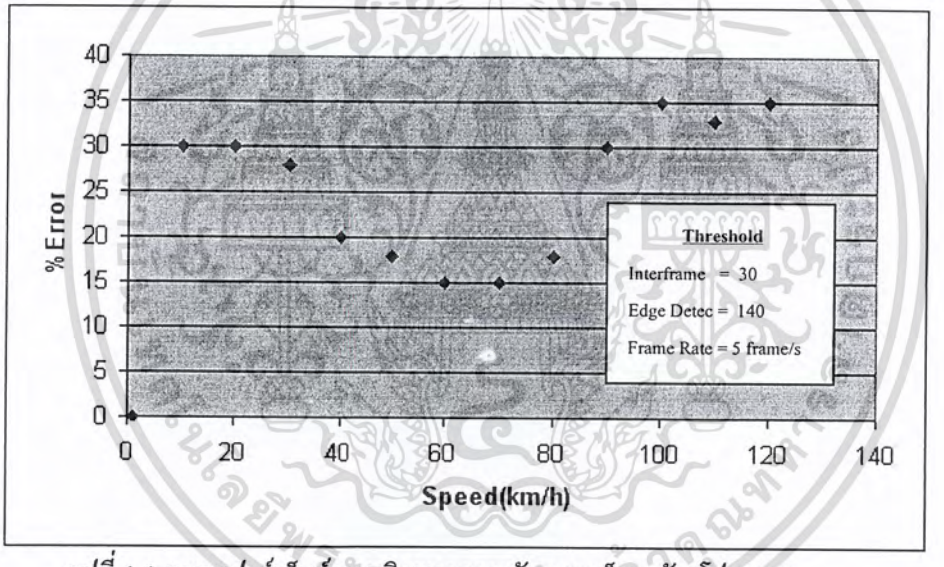
การทดลองเกี่ยวกับความเร็วนั้น เนื่องจากทางผู้ดำเนินโครงงานไม่สามารถหาอุปกรณ์ในการตรวจสอบความเร็วได้ ดังนั้นการทดลองจึงเป็นการประมาณความเร็วเฉลี่ยของรถที่วัดในความเร็วต่างๆ และประเมินผลที่ได้รับว่ามีความผิดพลาดมากน้อยเพียงใด

สำหรับการทดสอบวัดความเร็วรถนั้น ผู้ดำเนินโครงงานได้ทำการตั้งค่า Thresholds ของ Inter-frames ไว้ที่ 30 ค่า Thresholds ของ Sobel-Edge detection ที่ 140 ค่า Frame rate ของการจับหน้าจอได้แบ่งออกเป็น 3 ระดับด้วยกันคือ 3,5,8 frame/sec ผลการทดลองหาความเร็วของรถของโปรแกรมระบบตรวจสอบสภาพการจราจร แสดงเปอร์เซ็นต์ความผิดพลาดด้วยกราฟดังรูปที่ 6-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

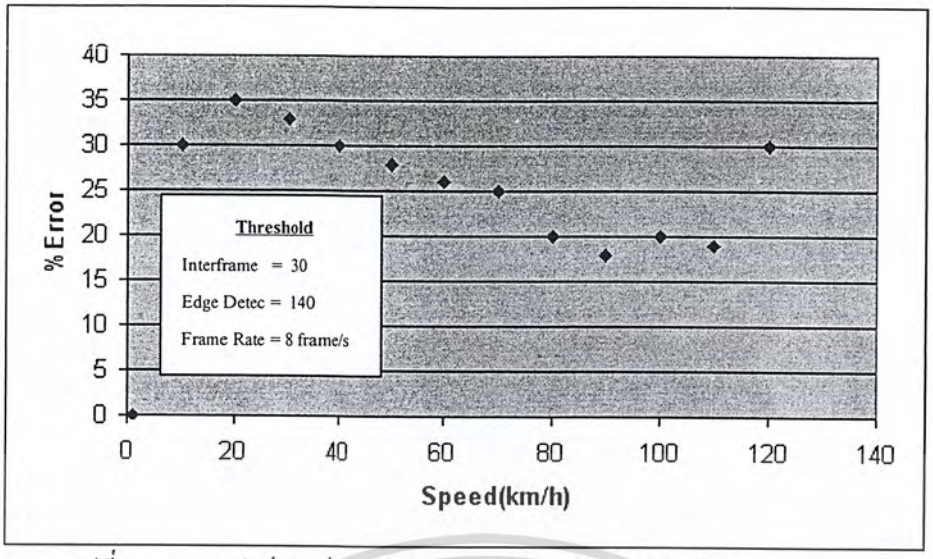


รูปที่ 6-3 แสดงเปอร์เซ็นต์ความผิดพลาดการวัดความเร็วรถด้วยโปรแกรม โดยตั้งค่า frame rate เท่ากับ 3 frame/sec



รูปที่ 6-4 แสดงเปอร์เซ็นต์ความผิดพลาดการวัดความเร็วรถด้วยโปรแกรม โดยตั้งค่า frame rate เท่ากับ 5 frame/sec

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-5 แสดงเปอร์เซ็นต์ความผิดพลาดการวัดความเร็วรถด้วยโปรแกรม โดยตั้งค่า frame rate เท่ากับ 8 frame/sec

6.3 สรุปผลการทดลอง

ระบบตรวจสอบสภาพการจราจรได้ใช้เทคนิคต่างๆ เพื่อลดขนาดของข้อมูลและเวลาที่ใช้ในการประมวลผลเพื่อที่ระบบนี้จะสามารถทำงานได้เรียลไทม์ จากการทดสอบพบว่าระบบมีความสามารถในการตรวจนับและตรวจจับความเร็วของรถในระดับที่น่าพอใจ เสถียรภาพของระบบเป็นที่น่าพอใจ และมีความผิดพลาดที่ยอมรับได้ในช่วงความเร็วที่ทดสอบโดยในส่วนของความเร็วนั้นจากผลที่ได้จากการบันทึกแสดงให้เห็นว่า ค่าความถูกต้องในการทำงานของระบบในส่วนของการตรวจสอบความเร็วนั้น จะขึ้นอยู่กับการระบุค่า Frame rate ให้เหมาะสมกับสภาพความคล่องตัวบนท้องถนนด้วย หากมีสภาพถนนที่รถเคลื่อนตัวได้ช้าก็ควรปรับค่า Frame rate ให้ต่ำลงในทางกลับกันหากสภาพถนนที่รถเคลื่อนตัวได้ค่อนข้างดี ก็ควรปรับค่าส่วนนี้ให้สูงขึ้นตามความเหมาะสม ในส่วนของความหนาแน่นของท้องถนนซึ่งยังไม่สามารถหาทฤษฎีมารองรับในส่วนนี้ได้จึงได้มีการสรุปออกมาเป็นความเร็วเฉลี่ยของรถซึ่งผ่านเข้ามาในระบบแทน

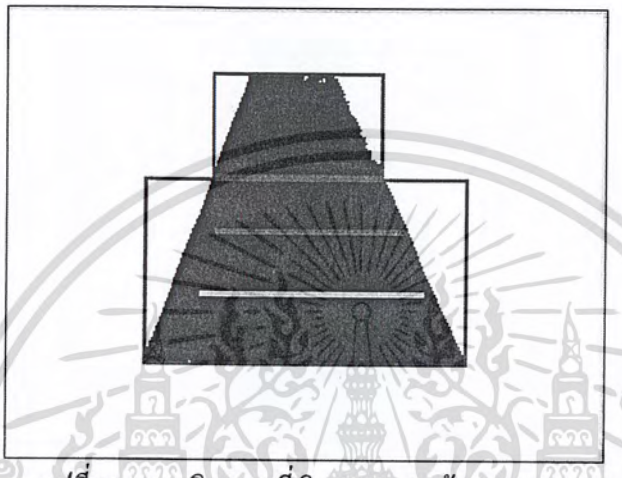
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
บทวิจารณ์และสรุป

7.1 บทวิจารณ์

จากการทดลองทำให้พบข้อจำกัดต่างๆ ของโครงงานระบบตรวจสอบสภาพการจราจรดังต่อไปนี้

1.ปัญหาที่เกิดจากความผิดพลาดในการคำนวณของโปรแกรมเมื่อมีแสงสะท้อนจากหลังคารถเข้าสู่กล้อง ดังรูปที่ 7-1



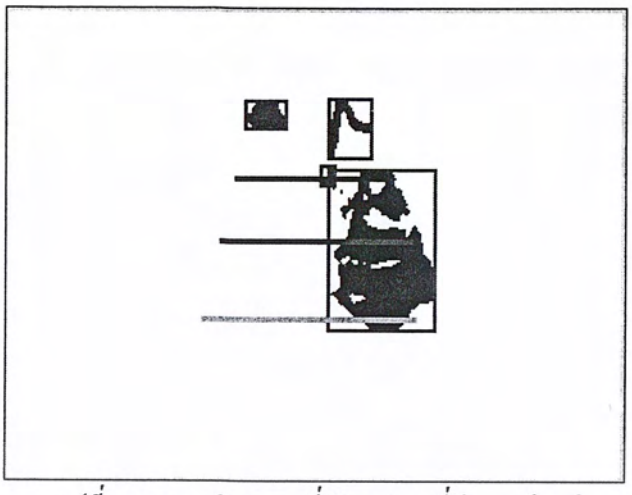
รูปที่ 7-1 ความผิดพลาดที่เกิดจากการสะท้อนของแสง

ในกรณีนี้ทำให้ขั้นตอนการทำ Interframe differences ตรวจพบว่าภาพทั้งภาพมีการเปลี่ยนแปลงของพิกเซลทำให้ไม่สามารถตรวจสอบรถในช่วงเวลานี้ได้ว่ามีหรือไม่ ทำให้การนับรถและความเร็วผิดพลาดไป แต่หลังจากที่เฟรมนี้ผ่านไปแล้วหรือไม่มีการสะท้อนของแสงแล้วโปรแกรมยังสามารถทำงานต่อไปได้

2.อุปสรรคที่เกิดจากกรณีที่มีรถใหญ่หรือรถที่มีความยาวมากๆ เช่น รถเมล์ รถบรรทุก ฯลฯ วิ่งผ่านในกรณีนี้การทำงานของโปรแกรมจะตรวจสอบจำนวนรถผิดพลาด จากที่ควรจะนับได้ 1 คัน เป็นนับได้หลายๆ คัน

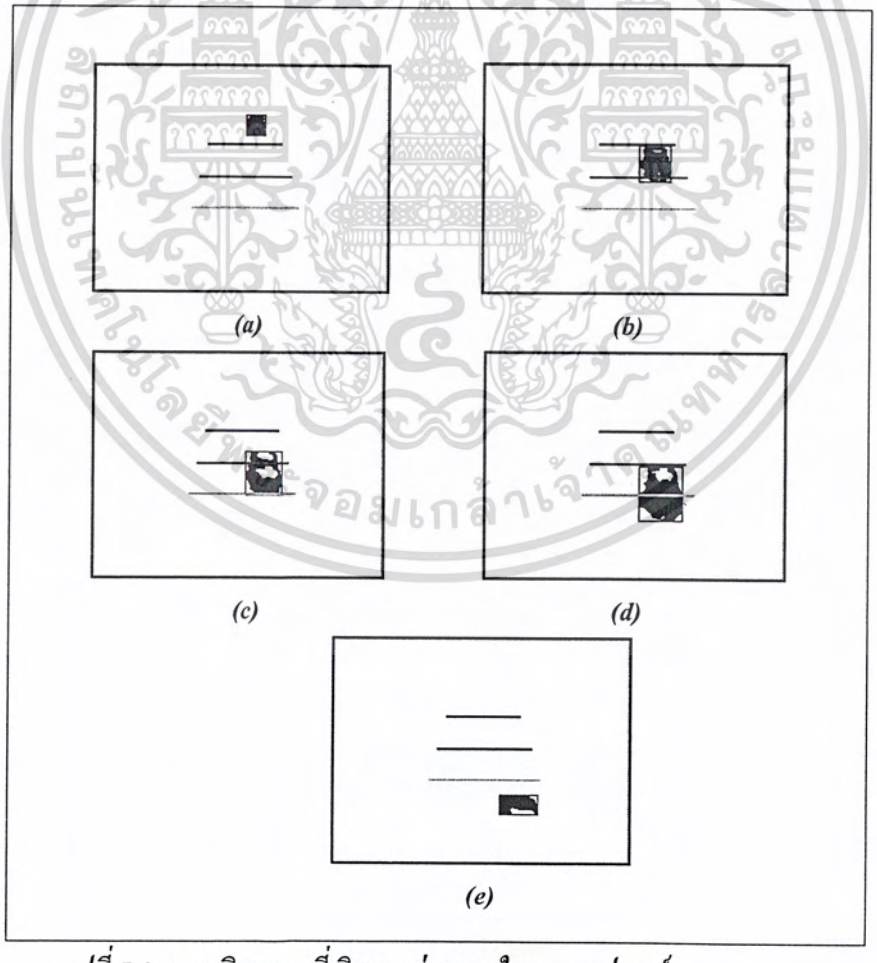
ความผิดพลาดนี้เกิดจากข้อจำกัดวิธีการที่ทางผู้จัดทำได้นำมาใช้ เนื่องจากรถที่มีขนาดยาวมากและมีช่วงลำตัวรถที่มีลักษณะคล้ายกัน เมื่อทำการเทียบหาความแตกต่างตามวิธีการ Inter-frame algorithm บริเวณช่วงลำตัวรถที่มีลักษณะคล้ายกัน จะถูกวิเคราะห์ว่าเป็นพื้นถนนทำให้เกิดการแบ่งส่วนหัว และท้ายรถ ออกเป็น 2 ส่วน โปรแกรมจึงวิเคราะห์ผิดพลาดว่าเป็นรถ 2 คัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-2 ความผิดพลาดที่เกิดจากรถที่มีขนาดใหญ่

3.ปัญหาที่เกิดจากช่วงเวลาในการแคปเจอร์รูปภาพ จากรูปที่ 7-3 จะสังเกตเห็นว่าถ้าวัตถุมีความเร็วในการเคลื่อนที่สูง จะทำให้รถวิ่งผ่านเส้นตรวจจับ โดยไม่ทับเส้นตรวจจับ ผลลัพธ์คือทำให้คำนวณความเร็วของวัตถุออกมาผิดพลาด



รูปที่ 7-3 ความผิดพลาดที่เกิดจากช่วงเวลาในการแคปเจอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 ข้อสรุปและข้อเสนอแนะ

ระบบตรวจสอบสภาพการจราจรได้ใช้เทคนิคต่างๆ ในการนับรถและวัดความเร็วของรถ ซึ่งจากการทดลองพบว่าระบบมีความสามารถในการนับรถและวัดความเร็วของรถ โดยมีความผิดพลาดอยู่ในระดับที่น่าพอใจ จากผลการทดลองนับจำนวนรถ และวัดความเร็วรถจากภาพถ่ายถนนที่มีรถวิ่งด้วยความเร็วเฉลี่ยไม่เกิน 90 กิโลเมตรต่อชั่วโมงพบว่าระบบสามารถนับและวัดความเร็วของรถได้ผิดพลาดไม่ถึง 20% ส่วนการวัดความหนาแน่นของรถนั้น เนื่องจากผู้ดำเนินโครงการยังไม่มีข้อมูลเรื่องความจุของถนนจึงไม่สามารถระบุได้ว่าความหนาแน่นของรถบนถนน ณ ช่วงเวลานั้นเป็นกี่เปอร์เซ็นต์ แต่จากการทดสอบสามารถคำนวณหาค่าความเร็วเฉลี่ยของรถได้ ซึ่งความเร็วเฉลี่ยของรถนี้สามารถใช้เป็นตัวชี้วัดความคล่องตัวบนถนนได้เช่นกัน

7.3 แนวทางในการพัฒนาโครงการ

แนวทางในการนำโครงการไปพัฒนาต่อไป มีดังต่อไปนี้

1.การนำไปพัฒนาต่อในส่วนของการกำจัดเงาของรถในช่วงเวลาที่มีเงาพาดผ่านส่วนของรถ ข้อผิดพลาดของระบบในปัจจุบันคือ ระบบวิเคราะห์ว่ารถที่ก่อให้เกิดเงาและรถที่ถูกเงาพาดผ่านเป็นรถคันเดียวกัน ทำให้ระบบคำนวณค่าต่างๆ ผิดไป

2.การพัฒนาในส่วนของการแยกแยะประเภทของรถ เช่น รถกระบะ รถเก๋ง รถตู้ รถบรรทุก ฯลฯ เพื่อที่สามารถนับจำนวนของรถแต่ละประเภท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

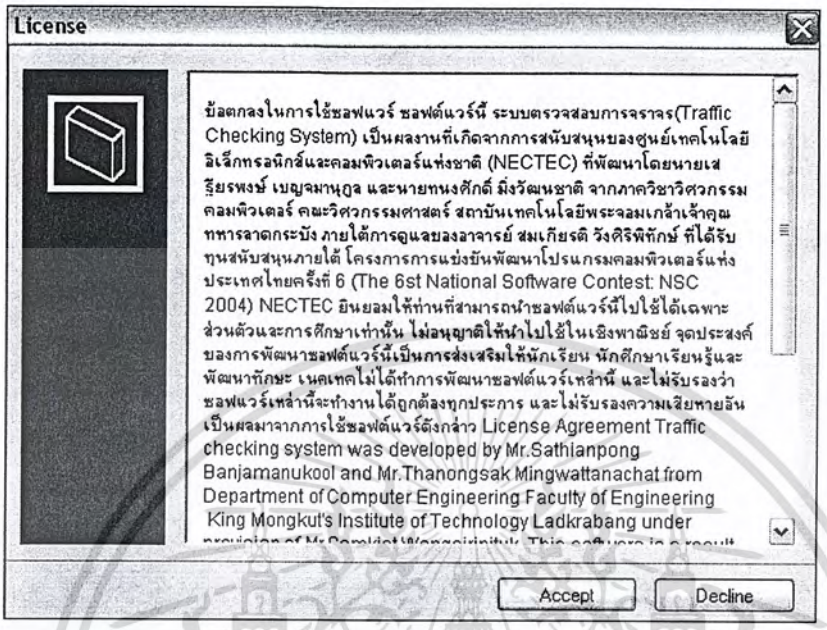
ภาคผนวก

คู่มือการติดตั้งและการใช้งานโปรแกรม

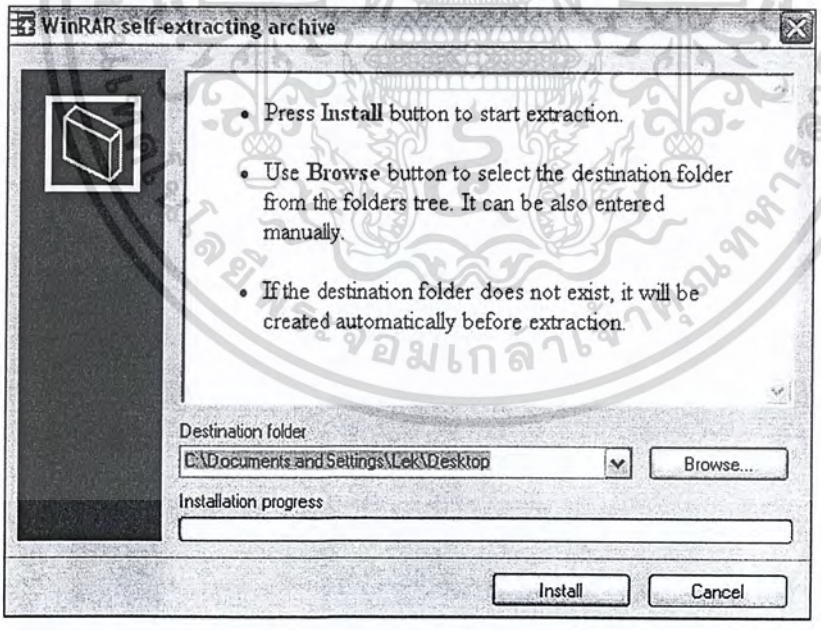
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการติดตั้งอย่างละเอียด

1. ทำการดับเบิ้ลคลิกที่ไฟล์ TrafficSystem.exe
2. กด Accept เพื่อยอมรับข้อตกลงในการใช้งานซอฟต์แวร์



3. ทำการ Browse หาโฟลเดอร์ที่ต้องการลงโปรแกรม หลังจากนั้นก็คลิกที่ปุ่ม Install



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งาน

1. จะต้องทำการเลือกว่าจะรับภาพอินพุทจากกล้องวิดีโอหรือไฟล์นามสกุล AVI ที่ได้ทำการบันทึกไว้ในเครื่องคอมพิวเตอร์ กรณีที่เลือกภาพจากกล้องวิดีโอจะเป็นการทำงานในส่วนของเรียลไทม์ (Real time) ดังนั้นจะต้องทำการติดตั้งกล้องก่อน โดยกล้องวิดีโอจะต่อผ่านพอร์ต USB ส่วนในกรณีที่ทำการเลือกจากไฟล์นามสกุล AVI ที่ทำการบันทึกไว้จะเป็นการทำงานในส่วนของออฟไลน์ (Offline) เมื่อทำการเลือกแล้วให้คลิกปุ่ม Browse เพื่อทำการเลือกไฟล์นามสกุล AVI ที่ต้องการ การปรับแต่งค่าต้องกระทำขณะที่รันโปรแกรมอยู่ และหลังจากจบโปรแกรมทุกครั้งจะต้องมีการรีเซตโปรแกรมด้วย

2. หลังจากทำการเลือกไฟล์เรียบร้อยแล้วต่อไปจะเป็นขั้นตอนในส่วนของการกำหนดค่าต่างๆ (Configuration) ซึ่งบางอย่างระบบจะทำการตั้งค่า Default ไว้เรียบร้อยแล้ว ค่า default นี้เป็นค่าที่ผู้ดำเนินโครงการได้เลือกใช้ระหว่างทำการทดลอง ดังนั้นการที่ผู้ใช้นำโปรแกรมไปใช้งาน ภาพอินพุทอาจจะมีความสว่างของภาพต่างกันจึงควรจะทำการปรับค่าต่างๆ เหล่านี้ให้เหมาะสม ซึ่งมีลำดับขั้นตอนการทำงานดังนี้

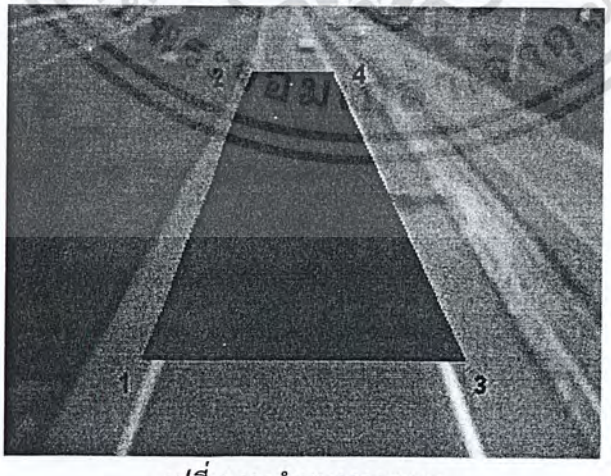
การกำหนดอาณาเขต

การกำหนดค่าอาณาเขตนั้นจะต้องทำทุกครั้งก่อนใช้งาน ผู้ใช้จะต้องเป็นคนกำหนดอาณาเขตของถนนที่ต้องการหาจำนวน ความเร็วและความหนาแน่นของรถ

การกำหนดอาณาเขตสามารถทำได้โดยการนำเมาส์ไปคลิกบนรูปภาพในลักษณะของกรอบสี่เหลี่ยม ซึ่งจะต้องทำการคลิกที่ตำแหน่งมุมของสี่เหลี่ยมตามลำดับที่กำหนดเท่านั้นโดยมีลำดับการคลิกดังต่อไปนี้

- ทำการคลิกจุดที่ตำแหน่ง 1
- ทำการคลิกจุดที่ตำแหน่ง 2
- ทำการคลิกจุดที่ตำแหน่ง 3
- ทำการคลิกจุดที่ตำแหน่ง 4

ตำแหน่งของลำดับจุดนั้นสามารถดูได้ ดังรูปที่ 1



รูปที่ 1 การกำหนดอาณาเขต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

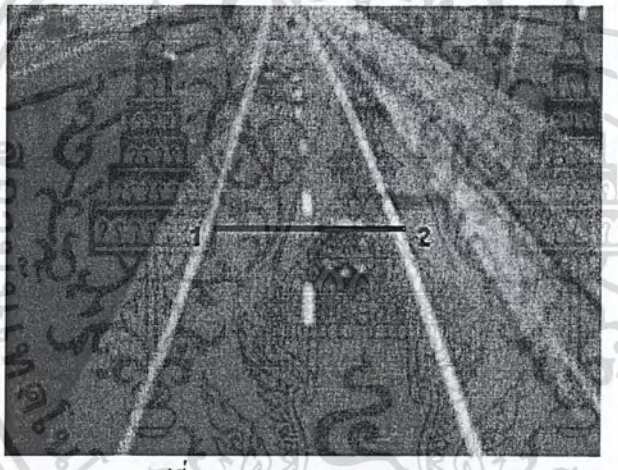
ในกรณีที่ทำการกำหนดอาณาเขตแล้วไม่พอใจ สามารถกำหนดอาณาเขตได้ใหม่โดยการคลิกจุดตามลำดับข้างต้นได้ทันที เปรียบเสมือนการกำหนดอาณาเขตใหม่

นอกจากนี้เราสามารถจะทำการปรับค่า Frame rate ได้โดยทำการคลิกปุ่มปรับค่าที่ช่อง Configuration หรือป้อนค่าที่เราต้องการลงไป หากเราป้อนค่า Frame rate มากเกินไปจะทำให้หน่วยประมวลผลกลางของคอมพิวเตอร์ทำงานหนัก แต่ถ้าป้อนค่าน้อยไปอาจทำให้ระบบตรวจสอบรถผิดพลาดได้

Counter line

เป็นการตีเส้นตรวจนับรถ เมื่อรถทำการวิ่งผ่านเส้นนี้ระบบจะทำการนับรถให้ทันที การตีเส้นจะต้องการทำการตีเส้นภายในอาณาเขตที่เรากำหนดมาก่อนหน้านี้ เส้นที่ตีควรจะอยู่ในแนวขวางของถนน โดยการตีเส้นจะต้องนำเมาส์ไปคลิกที่จุดเริ่มต้นของเส้นและปลายของเส้นตามลำดับหรือดูตัวเลขจากหน้าจอแสดงผลได้ ดังนี้

- กำหนดจุดเริ่มต้นของเส้น คือจุดที่ 1
- กำหนดจุดเริ่มต้นของเส้น คือจุดที่ 2



รูปที่ 2 การกำหนด Counter Line

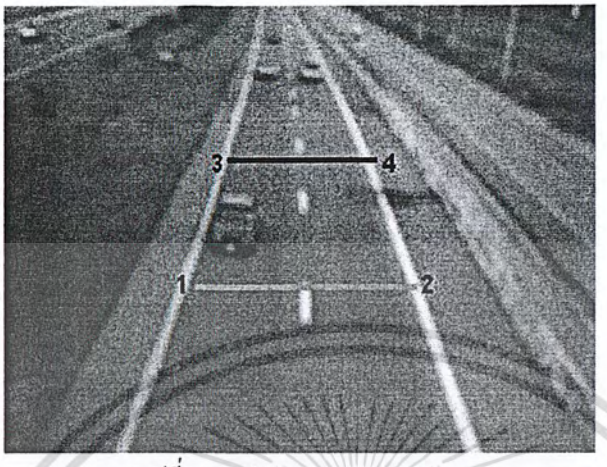
Detecting line

เป็นการตีเส้นตรวจจับความเร็วของรถบนถนน โดยจะต้องทำการตีเส้นทั้งหมด 2 เส้น ซึ่งจะต้องอยู่ภายในอาณาเขตเพื่อใช้ในการวัดความเร็วของรถ ทั้ง 2 เส้นจะต้องอยู่ในแนวขวางของถนน เส้นแรกจะเป็นเส้นที่วิ่งผ่านทีหลัง การตีเส้นสามารถทำได้โดยการนำเมาส์ไปคลิกบนภาพหรือดูตัวเลขจากหน้าจอแสดงผลได้ โดยซึ่งมีลำดับการคลิกจุดดังต่อไปนี้

- กำหนดจุดเริ่มต้นของเส้นแรก คือจุด 1
- กำหนดจุดปลายของเส้นแรก คือจุด 2
- กำหนดจุดเริ่มต้นของเส้นสอง คือจุด 3
- กำหนดจุดปลายของเส้นสอง คือจุด 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากการกำหนดเส้นในการวัดความเร็วรถแล้ว ผู้ใช้จะต้องระบุระยะห่างตามความเป็นจริงของถนนจากเส้นแรกไปยังเส้นที่สอง ซึ่งผู้ใช้อาจจะทำการวัดมาแล้วหรือใช้การประมาณจากเส้นประบนถนนก็ได้

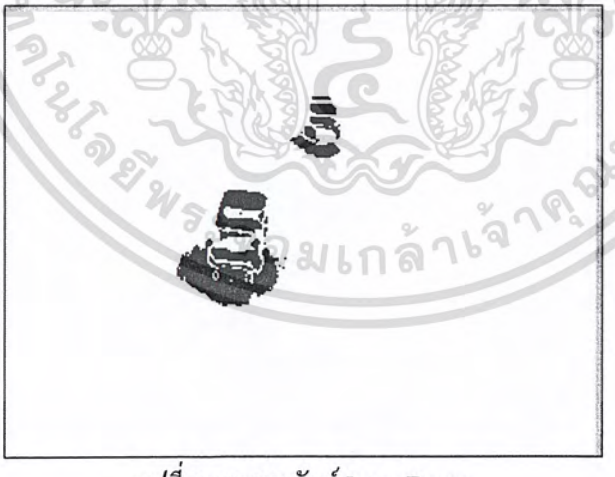


รูปที่ 3 การกำหนด Detecting line

3.เป็นขั้นตอนในการปรับค่า Threshold ต่างๆ และการดูผลลัพธ์ของภาพที่ผ่านการประมวลผลมาเรียบร้อยแล้ว ซึ่งมีขั้นตอนดังต่อไปนี้

Inter Frame

เป็นการแสดงผลลัพธ์ของการทำ Inter-frame และยังสามารถปรับค่า Threshold ของการทำ Inter-frame ได้อีกด้วย

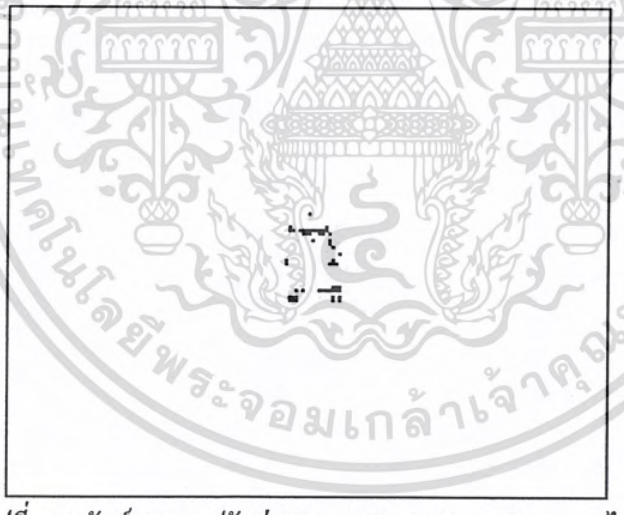


รูปที่ 4 แสดงผลลัพธ์ Inter-Frame

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับค่า Threshold นั้นหากเราทำการปรับค่าที่ต่ำไปจะทำให้ภาพเข้มจนไม่สามารถแยกส่วนที่เป็นรถยนต์กับพื้นถนนได้เลย แต่ถ้าเราทำการปรับค่า Threshold มากเกินไปจะทำให้มองเห็นส่วนที่เป็นรถยนต์ไม่ชัดหรือแทบมองไม่เห็นเลย ดังนั้นผู้ใช้ควรที่จะทำการปรับค่า Threshold ให้ได้ภาพที่มีรถยนต์วิ่งชัดเจนที่สุด หากไม่พอใจสามารถคลิกที่ปุ่ม Default เพื่อกลับไปเริ่มต้นที่ค่าเดิมที่ได้กำหนดไว้ได้

รูปที่ 5 ผลลัพธ์ของการปรับค่า Threshold ของ Inter-Frame ต่ำไป

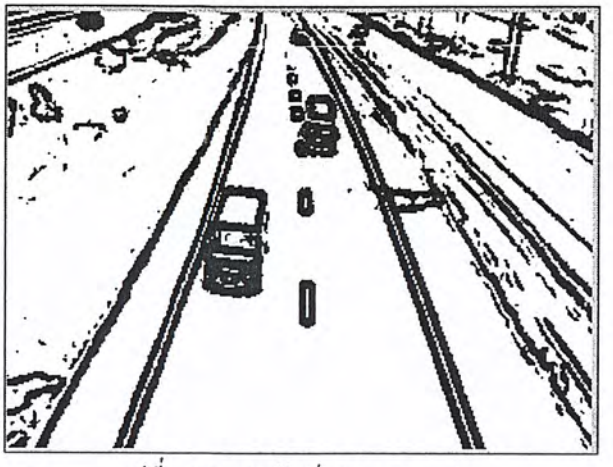


รูปที่ 6 ผลลัพธ์ของการปรับค่า Threshold ของ Inter-Frame สูงไป

Edge Detection

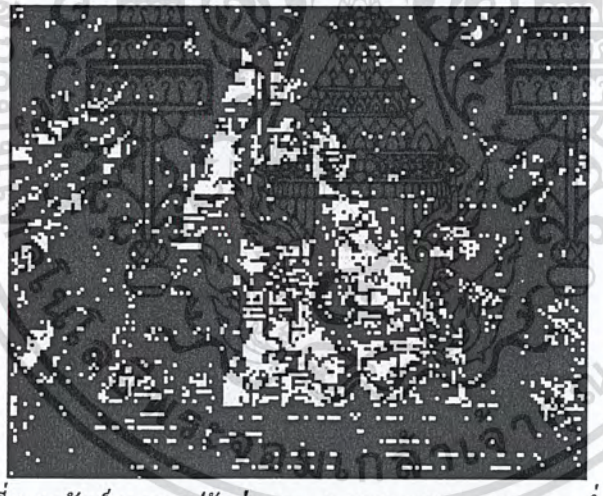
เป็นการแสดงภาพที่เป็นขอบของรถยนต์ที่ได้จากการทำ Edge Detection และใช้ปรับค่า Threshold ของการทำ Edge Detection ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



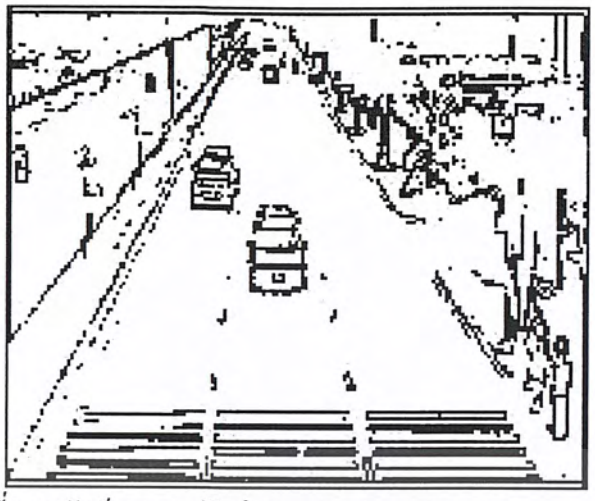
รูปที่ 7 แสดงผลลัพธ์ Edge Detection

การปรับค่า Threshold ของการทำ Edge Detection นั้นหากเราทำการปรับค่าต่ำเกินไปจะทำให้ภาพเข้มจนมองไม่เห็นขอบของรถยนต์ที่กำลังวิ่งอยู่ แต่ถ้าเราปรับค่าสูงเกินไปจะทำให้ภาพจางจนมองไม่เห็นขอบของรถยนต์เช่นเดียวกัน ดังนั้นวิธีการปรับค่า Threshold ก็คือต้องปรับค่าจนกระทั่งคิดว่ามองเห็นภาพขอบของรถยนต์ให้ชัดเจนที่สุด หากเราไม่พอใจสามารถคลิกที่ปุ่ม Default เพื่อกลับไปเริ่มต้นที่ค่าเดิมที่ได้กำหนดไว้ได้



รูปที่ 8 ผลลัพธ์ของการปรับค่า Threshold ของ Edge-detection ต่ำไป

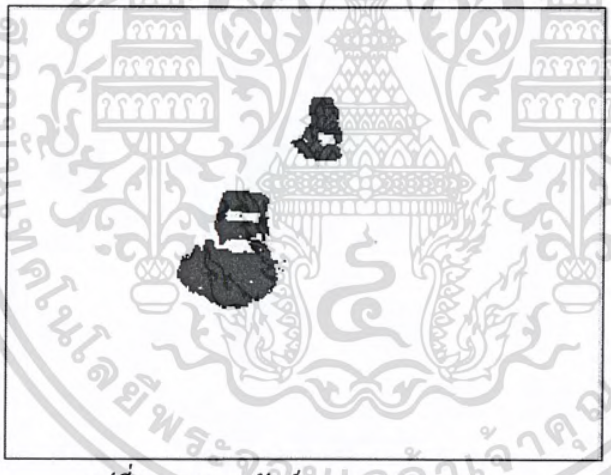
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9 ผลลัพธ์ของการปรับค่า Threshold ของ Edge-detection สูงไป

Moving-Edge closure

แสดงผลลัพธ์ของภาพที่มาจากการ Moving-Edge detection กันของภาพจาก Inter frame และ Edge Detection ในระหว่างที่ผู้ใช้งานดูผลลัพธ์ของภาพ ผู้ใช้งานสามารถปรับค่า Frame rate, threshold ของ Inter Frame และ Edge Detection ได้

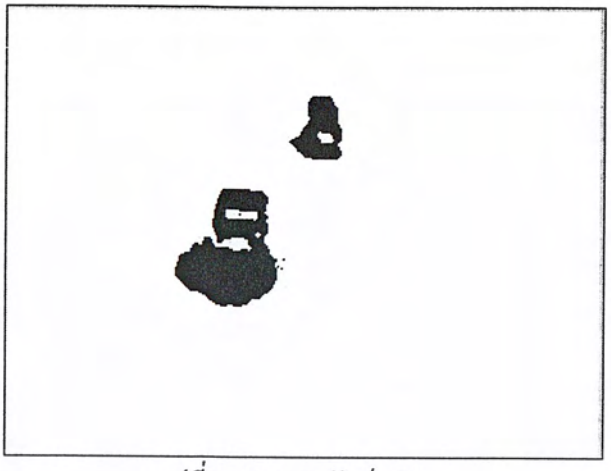


รูปที่ 10 แสดงผลลัพธ์ Moving-Edge closure

Filter

แสดงผลลัพธ์ที่เราได้ทำการ Filling ภาพที่ได้มาจากการ AND กันข้างต้น เพื่อเพิ่มรายละเอียดของภาพให้ได้ภาพที่มีส่วนของรถที่ทำการเคลื่อนที่มีความคมชัดมากยิ่งขึ้น ในระหว่างที่ดูผลลัพธ์ของภาพผู้ใช้งานสามารถปรับค่า Frame rate, threshold ของ Inter Frame และ Edge Detection ได้

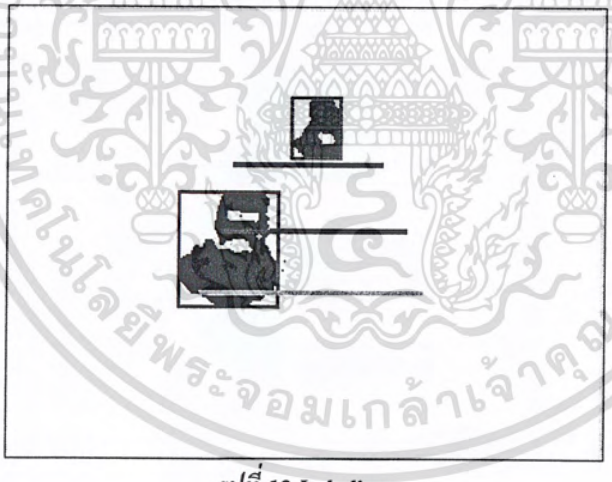
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 11 แสดงผลลัพธ์ Filling

Labeling

แสดงภาพที่มีการตีกรอบส่วนที่เป็นรถยนต์ และแสดงเส้นนับจำนวนรถและวัดความเร็วของรถ นอกจากนี้ยังแสดงผลลัพธ์ที่เป็นตัวเลขของจำนวนรถ ความเร็วรถและความหนาแน่นของรถ ในส่วนของการแสดงผลลัพธ์ของโปรแกรม ถ้าผู้ใช้งานคลิกที่เครื่องหมายถูกหน้า Config เพื่อลบเครื่องหมายถูกออก โปรแกรมจะทำการแสดงผลภาพในรูปแบบเดียวกันกับภาพสดพร้อมทั้งแสดงผลลัพธ์ที่เป็นตัวเลข ในระหว่างที่ดูผลลัพธ์ของภาพ ผู้ใช้งานสามารถปรับค่า Frame rate, threshold ของ Inter Frame และ Edge Detection ได้อีกด้วย



รูปที่ 12 Labeling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 13 หน้าจอแสดงผลให้กับผู้ใช้งาน

หน้าที่ของปุ่มต่างๆ ที่ใช้ในการควบคุมโปรแกรม (Control)

1. ปุ่ม Play ใช้สำหรับรันโปรแกรมเมื่อทำการเลือกอินพุทเสร็จเรียบร้อยแล้วหรืออาจรันโปรแกรมเพื่อทำการ Config ค่าต่างๆ

2. ปุ่ม Close ใช้สำหรับออกจากโปรแกรม

3. ปุ่ม Reset Counter ใช้รีเซตค่าของตัวเลขรถที่เราได้ทำการนับไว้

4. ปุ่ม Reset ใช้สำหรับรีเซตค่าต่างๆ ภายในระบบที่เราได้ทำการปรับแต่งค่าไว้ก่อนหน้านี้ จะต้องกดทุกครั้งหลังจากมีการใช้งานแล้วเมื่อต้องการจะเริ่มต้นการทำงานใหม่อีกครั้ง

5. ปุ่ม Help จะแสดงตัวช่วยเพื่อทำความเข้าใจในกรณีที่ผู้ใช้ไม่เข้าใจการทำงานของโปรแกรม

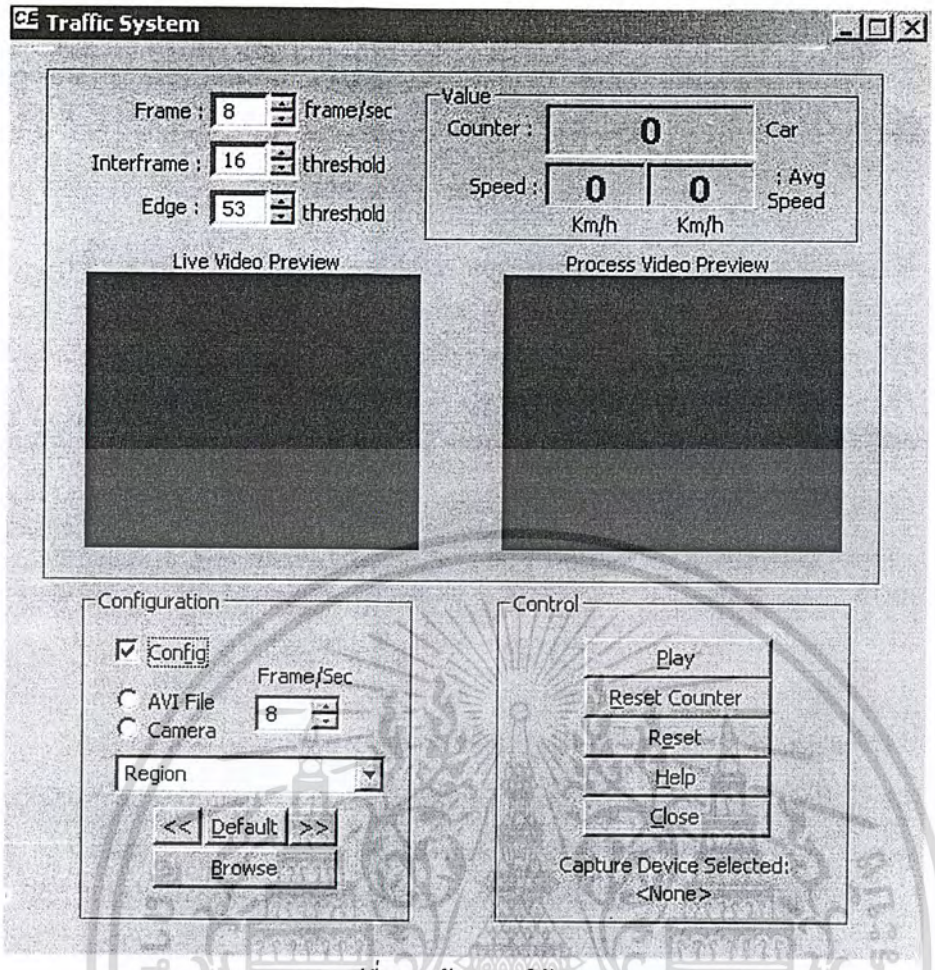
หน้าจอแสดงผลลัพธ์ (Value)

1. ส่วน Counter แสดงผลลัพธ์ของจำนวนรถ

2. ส่วน Speed แสดงผลลัพธ์ความเร็วของรถที่วิ่งผ่านเส้นตรวจจับความเร็วเส้นที่ 2

3. ส่วน Avg. speed แสดงผลลัพธ์ของความเร็วเฉลี่ยของรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 14 หน้าจอการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, Addison-Wesley Publishing Company, September 1993

[2] Richard M. Jones, Introduction MFC Programming with Visual C++, Prentice Hall, Inc., 2000

[3] นิรุธ อำนวยศิลป์, คู่มือการเขียนโปรแกรม Microsoft Visual C++ Version 6.0, พิมพ์ที่ บริษัท ส. เอเซียเพรส (1989) จำกัด

[4] Xiaokun Li and Fatih Porikli, Traffic Event Detection: A Survey, February 2003

[5] Rita Cucchiara, Massimo Piccardi and Paola Mello, Image Analysis and Rule-Based Reasoning for a Traffic Monitoring System, June 2000

[6] Richard Johnsonbaugh and Martin Kalin, Application Programming in C++, Prentice Hall Inc., 1999

[7] ยุทธนา ลีลาศวัฒนกุล, คู่มือการเขียนโปรแกรมและใช้งาน Visual C++.NET ฉบับสมบูรณ์, พิมพ์ที่ บริษัท ด่านสุทธาการพิมพ์ จำกัด, พิมพ์ครั้งที่ 1 มกราคม 2546

[8] กฤษดา โชคสินอนันต์, สืบสกุล พิภพมงคล, สรวิศ นฤปิติ, การวัดความเร็วของรถยนต์โดยวิธีหาขอบหน้ารถ (Measuring velocity of car by finding edge of car, ภาควิศวกรรมคอมพิวเตอร์และภาควิชาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัยกรุงเทพฯ

[9] Stanislaw Berka and Kent Lall, "News perspectives for ATMS: Advanced Technologies In Traffic Detection", Journal of Transportation, January/February 1998

[10] Cheng-Chen kou and Randy B.Machemehl, "Probabilistic Speed Estimation Measurement Errors through Video Images", Journal of Transportation engineering, March/April 1997

[11] Viroat Srisuapanon, "Analysis of Traffic Data using the Image Processing Approach", Graduate school, Asian Institute of Technology, Thailand 1993

[12] เกสร สุนสิน, "การพัฒนาโปรแกรมวิเคราะห์ภาพเพื่อนับยานพาหนะบนถนน", วิทยานิพจน์ปริญญามหาบัณฑิตภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ประเทศไทย, 2538

[13] Marie-Pierre Dubuisson, "Segmentation and Matching of Moving Vehicle in Complex Outdoor Scenes Doctoral Dissertation", Department of Computer Science, Graduate School, Michigan State University, 1995

[14] Irwin Miller and John E.Freund, "Probability and Statistics for Engineer", Prentice-Hall .Inc, 1977

[15] R.C. Gonzalex and R.E. Woods, "Digital Image processing", Addison-Wesley Publishing Company Inc., 1993

[16] ปราโมทย์ เดชะอำไพ, "ระเบียบวิธีเชิงตัวเลขในงานวิศวกรรม", สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้