สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การนับแยกรถยนต์โดยการประมวลผลภาพ

VECHICLE CLASSIFICATION USING DIGITAL IMAGE PROCESSING

โดย

นาย ภาคภูมิ อรรถวิทกุล รหัส 46010569

นาย ภีมะ จำรัสประเสริฐ รหัส 46010583

อาจารย์ที่ปรึกษา

รศ.ดร. ชูชาติ ปิณฑวิรุจน์

เลขหมู่.....
เลขทะเบียน 72682
วัน,เดือน,ปี 21 มิ.ย. 2550

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

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา2549

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การนับแยกรถยนต์โดยใช้การประมวลผลภาพ

ผู้จัดทำ

1.นาย ภาคภูมิ อรรถวิทกุล รหัส 46010569

2.นาย ภีมะ จำรัสประเสริฐ รหัส 46010583

.....อาจารย์ที่ปรึกษา

(รศ.ดร. ชูชาติ ปิณฑวิรุจน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การนับแยกรถยนต์โดยใช้การประมวลผลภาพ

นาย ภาคภูมิ อรรถวิทกุล รหัส 46010569
นาย ภีมะ จำรัสประเสริฐ รหัส 46010583
รศ.ดร.ชูชาติ ปิณฑวิรุจน์(อาจารย์ที่ปรึกษา)
ปีการศึกษา 2549

บทคัดย่อ

โครงการนี้นำเสนอการออกแบบและเขียนโปรแกรมการนับจำนวนและแยกชนิดของยานพาหนะเพื่อใช้ในการตรวจสอบและวัดปริมาณของยานพาหนะ ที่ผ่านเส้นทางหนึ่ง โดยการใช้การประมวลผลภาพผ่านไฟล์วิดีโอ หรือ แบบเรียลไทม์โดยผ่านกล้องวีดิโอ ซึ่งสามารถกำหนดบริเวณตรวจสอบ หรือActive Area ให้มีรูปร่างแปรเปลี่ยนไปตามรูปทรงของช่องทางการจราจรได้ และเมื่อทำการประมวลผลเสร็จจะแสดงผลลัพธ์ เป็นจำนวนยานพาหนะ ค่าความเร็วเฉลี่ย และรูปกราฟแสดงค่าเฉลี่ยความเข้มแสงที่ได้จากการลบกันของภาพอ้างอิงกับภาพที่ต้องการประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VEHICLE CLASSIFICATION USING DIGITAL IMAGE PROCESSING

Mr.Pakpoom Attawitkul ID.46010569

Mr.Peema Chamratprasert ID.46010583

Assoc.Prof.Dr.Chuchart Pintavirooj(Advisor)

Educational Year 2006

Abstract

This project describes a design and programming of a vehicle classification program for monitoring and classification of the vehicle on the road, using digital image processing to process through a video file or process real time through a real time camera which can be assigned the multi-shape active area to process. After processing, program will show a result which consisted of number of vehicle, type of vehicle, average speed and reference graph.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำโครงงานการนับแยกจำนวนรถยนต์โดยการประมวลผลภาพนี้ คณะผู้จัดทำขอขอบพระคุณอาจารย์ที่ปรึกษาเป็นอย่างสูงที่ได้ให้คำปรึกษา และคำแนะนำในการทำโครงงานชิ้นนี้ขอบพระคุณเหล่าคณาจารย์และเจ้าหน้าที่ภาควิชาวิศวกรรมศาสตร์อิเล็กทรอนิกส์ ในความช่วยเหลือทุกๆด้าน ขอขอบคุณพ่อแม่ผู้ปกครองที่ให้การสนับสนุนทั้งด้านการเงินและคอยเป็นกำลังใจในการทำงานเสมอมา และขอบคุณเพื่อน ๆ และพี่ๆที่ ที่ช่วยให้การทำโครงงานงานนี้สำเร็จลุล่วงไปได้

นาย ภาคภูมิ อรรถวิทกุล
นาย ภีมะ จำรัสประเสริฐ
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	3
Abstract	4
กิตติกรรมประกาศ	5
สารบัญ	6
บทที่ 1 บทนำ	8
1.1 ความเป็นมาของปริญญานิพนธ์	8
1.2 จุดมุ่งหมายของปริญญานิพนธ์	8
1.3 ขอบเขตของโครงงาน	8
1.4 ประโยชน์และผลที่คาดว่าจะได้รับจากโครงงาน	9
บทที่ 2 ทฤษฎีและความรู้พื้นฐาน	10
2.1 ขั้นตอนพื้นฐานของการประมวลผลภาพ	10
2.2 ภารปรับปรุงภาพ(Image Enhancement)	11
2.2.1 การบีบย่าน(Range Compression)	12
2.2.2 การหักลบภาพและตรวจการเปลี่ยนแปลง	12
2.3 ภารวิเคราะห์รูปร่าง (Shape Analysis)	12
2.4 การแยกวัตถุจากภาพ(Segmentation)	13
2.4.1 การแยกภาพด้วยการพิจารณาการต่อเนื่องของข้อมูล	13
2.4.2 การแยกภาพด้วยวิธี Region Labeling	13
2.5 ภาพดิจิตอล(digital image)	14
2.6 ไฟล์ข้อมูลภาพชนิด bmp	15
2.7 ชนิดของภาพ (Image Type)	17
2.8 ลักษณะการจัดเก็บข้อมูลภาพแบบดิจิตอล	18
2.9 การแปลงระดับเกรย์	19
2.10 การสร้างภาพไบนารี	20
2.11 ฮิสโตแกรม (Histogram)	21
2.12 กลุ่มสีที่ใช้แสดงผล	22
บทที่ 3 หลักการทำงานและการออกแบบ	25
3.1 หลักการทำงานของโปรแกรม	25
3.2 การออกแบบ	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 ผลการทดลอง 34
4.1 ผลการทดลองการนับจำนวนยานพาหนะโดยประมวลผลจากไฟล์วิดีโอ 34
4.2 ผลการทดลองการนับจำนวนยานพาหนะโดยประมวลผลแบบเรียลไทม์ผ่านกล้องวิดีโอ 40
4.3 ผลการทดลองเปรียบเทียบการใช้ Active Area แบบกรอบสี่เหลี่ยมและแบบกำหนดรูปร่างได้ 44
4.4 ผลการทดลองที่เกิดข้อผิดพลาดขึ้น 44
บทที่ 5 สรุปและวิจารณ์ผลการทดลอง 45
5.1 สรุปและวิจารณ์ผลการทดลอง 45
5.2 ข้อดีและข้อเสีย 45
5.3 การนำไปประยุกต์ใช้งาน 45
5.4 แนวทางการพัฒนา 46
บรรณานุกรม 47
ภาคผนวก 48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของปริญญานิพนธ์

ในปัจจุบันปัญหาการจราจรติดขัดนับเป็นปัญหาอันดับต้นๆของเมืองที่มีประชากรอาศัยอยู่อย่างหนาแน่น ซึ่งการแก้ปัญหาการจราจรนี้สร้างความวิตกกังวลสำหรับบุคคลที่มีหน้าที่เกี่ยวข้องเป็นอย่างมาก กลุ่มผู้ดำเนินโครงงานได้เล็งเห็นถึงปัญหานี้ที่คาดว่าจะส่งผลกระทบต่อการจราจรของผู้คนเพิ่มมากขึ้น จึงได้กำเนิดเป็นโครงงานนี้ขึ้นมา ซึ่งเป็นโครงงานเกี่ยวกับการใช้คอมพิวเตอร์ในการประมวลผลจำนวนรถยนต์ ความเร็วรถยนต์ หรือแม้แต่ประเภทของรถยนต์ เพื่อนำไปวิเคราะห์สภาพการจราจรที่เกิดขึ้นในขณะนั้นๆ และสามารถนำไปใช้ร่วมกับสัญญาณไฟจราจรต่างๆ โดยจะเปลี่ยนแปลงตามสภาพการจราจรที่เป็นอยู่ในขณะนั้นเลยโดยไม่จำเป็นต้องใช้เจ้าหน้าที่ในการกดเปิดปิดสัญญาณไฟ อีกทั้งประโยชน์ทางอ้อมที่ได้รับจากโครงงานนี้คือสามารถตรวจสอบและแก้ปัญหาการขับรถผิดกฎจราจรเช่น ปัญหาการใช้ความเร็วเกินกว่าความเร็วที่กำหนดไว้ได้ กลุ่มผู้ดำเนินโครงงานจึงคิดว่าโครงงานนี้จะเป็นส่วนหนึ่งในการแก้ปัญหาการจราจรได้อีกทางหนึ่งที่มีการสูญเสียงบประมาณเพียงเล็กน้อยเท่านั้น

1.2 จุดมุ่งหมายของปริญญานิพนธ์

- ศึกษาการประมวลผลภาพด้วยคอมพิวเตอร์ โดยเฉพาะเทคนิคการลบภาพและการทำเทรชโฮล
- สามารถประมวลผลภาพการจราจรในรูปแบบไฟล์วิดีโอ แล้วแสดงผลออกมาในลักษณะข้อความและกราฟทางจอแสดงผล
- สามารถรับการข้อมูลประมวลได้ทั้งแบบเรียลไทม์และข้อมูลแบบไม่เรียลไทม์ได้
- สามารถปรับแก้จุดบกพร่องที่เกิดจากการที่รถแล่นคร่อมช่องทางการจรได้

1.3ขอบเขตของโครงงาน

โครงงานนี้จะเป็นการประมวลผลภาพวิดีโอในรูปแบบของไฟล์ .avi โดยจะเป็นการดึงภาพแต่ละเฟรมจากไฟล์วิดีโอมาทำการลบกับภาพอ้างอิงจากเฟรมแรกด้วยวิธีหาผลต่างสัมบูรณ์ (Absolute Difference)หลังจากนั้นจะนำภาพผลลัพธ์ที่ได้มาหาค่าเฉลี่ยความเข้มแสงเพื่อใช้ในการนับจำนวนและแยกประเภทของยานพาหนะต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ประโยชน์และผลที่คาดว่าจะได้รับจากโครงงาน

- เป็นการประยุกต์ใช้ทฤษฎีการประมวลผลภาพด้วยคอมพิวเตอร์ เพื่อใช้เป็นแนวทางในการนำไปใช้รายงานสภาพการจราจร ซึ่งทำให้เกิดความสะดวกรวดเร็วในการรายงาน และลดจำนวนของเจ้าหน้าที่ซึ่งในปัจจุบันต้องมาคอยควบคุมสภาพการจราจรแล้วรายงานผลด้วยตนเอง
- เป็นการศึกษาการใช้ภาษา Visual C++ .Net ในการเขียนโปรแกรมทางด้านการประมวลผลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2
ทฤษฎีและความรู้พื้นฐาน

การประมวลผลภาพดิจิตอล หมายถึง การประมวลผลภาพ 2 มิติโดยใช้เครื่องคอมพิวเตอร์ในทางดิจิตอล ภาพดิจิตอลหมายถึงอาร์เรย์ของจำนวนจริงหรือจำนวนเชิงซ้อนที่ถูกแทนด้วยบิทที่จำกัด โดยภาพจะถูกทำให้อยู่ในรูปของดิจิตอลและถูกเก็บอยู่ในรูปของเมทริกซ์ของเลขฐานสองในหน่วยความจำของคอมพิวเตอร์ การประมวลผลภาพดิจิตอลถูกนำมาใช้ในการประยุกต์ใช้งานงานต่างๆอย่างกว้างขวาง เช่นการวิเคราะห์ภาพถ่ายทางอากาศ การส่งภาพและเก็บภาพสำหรับงานธุรกิจ การประมวลผลทางการแพทย์ การประมวลผลภาพถ่ายด้วยเรดาร์ โซนาร์และสัญญาณความถี่ย่านเสียง งานด้านหุ่นยนต์ และการตรวจสอบอัตโนมัติของชิ้นส่วนทางอุตสาหกรรม ในที่นี้ผู้ดำเนินโครงงานได้นำมาใช้เพื่อประมวลผลจำนวนรถยนต์ ความเร็วรถยนต์ หรือแม้แต่ประเภทของรถยนต์ เพื่อนำไปวิเคราะห์สภาพการจราจรที่เกิดขึ้น

2.1 ขั้นตอนพื้นฐานของการประมวลผลภาพ

ขั้นตอนแรกของการประมวลผลภาพคือการรับข้อมูลภาพ(Image Acquisition) หน่วยรับข้อมูลภาพแบ่งออกเป็น หน่วยพื้นฐาน ส่วนแรกเป็นอุปกรณ์ที่ไวต่อสเปกตรัมของคลื่นแม่เหล็กไฟฟ้าได้แก่ เอ็กซ์เรย์ อุลตร้าไวโอเลต แสงที่ตามองไม่เห็น หรือย่านอินฟราเรด และให้สัญญาณไฟฟ้าเป็นสัดส่วนกับระดับพลังงานคลื่นแม่เหล็กไฟฟ้าที่ตรวจวัด หน่วยที่ 2 เป็น หน่วยที่ทำการเปลี่ยนเอาท์พุทที่เป็นสัญญาณไฟฟ้าให้อยู่ในรูปดิจิตอลซึ่งสามารถนำประมวลผลในเครื่องคอมพิวเตอร์ได้

หลังจากได้รับภาพที่เป็นสัญญาณดิจิตอล ขั้นตอนต่อไปเกี่ยวข้องกับขบวนการก่อนการประมวลผล หรือ Preprocessing วัตถุประสงค์ที่สำคัญของขั้นตอนนี้คือการปรับปรุงภาพ เพื่อเพิ่มโอกาสที่จะประสบผลสำเร็จของขั้นตอนการประมวลผล ตัวอย่างขั้นตอนการปรับปรุงภาพในขบวนการประมวลผลภาพ เพื่อการอ่านที่อยู่ที่ระบุบนจดหมายได้แก่การเพิ่ม คอนทราส การกำจัดน๊อยส์ การแยกบริเวณที่เป็นอักษรเป็นต้น

ขั้นตอนต่อไปเกี่ยวข้องกับขบวนการแยกส่วนภาพ(Segmentation) ความหมายกว้างๆของการแยกส่วนภาพ คือการแบ่งส่วนภาพออกตามองค์ประกอบหรือตามวัตถุที่ประกอบเป็นภาพ โดยทั่วไปแล้วการแยกส่วนภาพโดยอัตโนมัติเป็นงานที่ยากลำบากในการประมวลผลภาพดิจิตอล และ ความสำเร็จของการประมวลผลขึ้นกับประสิทธิภาพของการแยกส่วนภาพในตัวอย่างของการอ่านที่อยู่ที่ระบุบนจดหมาย เป้าหมายของการแยกส่วนภาพคือการแยกตัวอักษรเดี่ยวและคำออกจากพื้นหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอาท์พุทที่ได้จากขบวนการแบ่งส่วนภาพเป็นข้อมูลพิกเซลดิบๆซึ่งรวมส่วนที่เป็นขอบภาพ (Boundary) และจุดทุกจุดภายในบริเวณที่แยกได้ ไม่ว่าจะเป็นกรณีใดการเปลี่ยนแปลงข้อมูลให้อยู่ในรูปที่เหมาะสมสำหรับการประมวลผลด้วยคอมพิวเตอร์เป็นสิ่งจำเป็น เราต้องตัดสินใจว่าเราจะแทนข้อมูลด้วยบริเวณของขอบหรือบริเวณทั้งหมดของข้อมูล การแทน (Representation) ข้อมูลด้วยขอบภาพมีความเหมาะสมเมื่อเราต้องการใช้คุณลักษณะของรูปร่างภายนอก เช่นมุม หรือจุดความโค้งเป็นศูนย์ (Inflection Point) การแทนข้อมูลด้วยบริเวณมีความเหมาะสมเมื่อเราต้องการใช้คุณลักษณะภายในเช่น (Texture)
หรือรูปร่างของโครงร่าง อย่างไรก็ตามในการประยุกต์ใช้งานบางอย่าง การใช้การแทนข้อมูลทั้งสองแบบร่วมกันมีความเป็นไปได้ ซึ่งต้องการข้อมูลของขอบภายนอก โครงร่างภายใน

การแทนข้อมูลเป็นเพียงขั้นตอนหนึ่งที่ใช้แปลงข้อมูลดิบให้อยู่ในรูปแบบที่เหมาะสมแก่การประมวลผลด้วยคอมพิวเตอร์ต่อไป ขั้นต่อๆไปคือการระบุวิธีที่ใช้ในการอธิบายข้อมูลเพื่อให้ลักษณะที่เราสนใจ เด่นชัดขึ้น การอธิบายหรือที่เรียกว่าการเลือกลักษณะพิเศษเกี่ยวข้องกับการแยกลักษณะเพื่อให้ได้เป็นข้อมูลเชิงปริมาณ

ขั้นตอนสุดท้ายเกี่ยวข้องกับการจดจำและการตีความ การจดจำเป็นขบวนการทำฉลากให้กับวัตถุโดยใช้ลักษณะพิเศษที่ได้จากขั้นตอนการอธิบาย การตีความเกี่ยวกับการให้ความหมายกับกลุ่มของวัตถุที่จำได้

2.2 การปรับปรุงภาพ(Image Enhancement)

การปรับปรุงภาพ หมายถึงการเน้น (หรือการทำให้คม) ลักษณะของภาพ (Features)เช่น ขอบ (Edge) เขต (Boundaries) หรือ ความแตกต่างระหว่างบริเวณขาวกับดำหรือคอนทราส (Contrast)เพื่อทำให้การแสดงกราฟฟิกมีประโยชน์มากขึ้นสำหรับการแสดงและการวิเคราะห์ ขบวนการปรับปรุงภาพไม่ได้เพิ่มองค์ประกอบข้อมูลในภาพ แต่จะไปเพิ่มช่วงการทำงาน (Dynamic Range) ของลักษณะของภาพที่เลือกเพื่อให้ง่ายต่อการตรวจจับ การปรับปรุงภาพประกอบด้วยการปรับปรุงระดับเทาและ คอนทราส การลดสัญญาณรบกวน การเน้นขอบ การกรอง (Filtering) การอินเทอโพเลชั่น(Interpolation)
การขยาย (Magnification) การทำสีเทียม (Pseudocoloring) และอื่นๆ ข้อที่ยากที่สุดสำหรับการปรับปรุงภาพคือการตั้งเกณฑ์(Criterion) ที่เหมาะสำหรับการปรับปรุงภาพ ดังนั้นเทคนิคการปรับปรุงภาพส่วนใหญ่จะมีลักษณะเป็นการทดลอง หรือต้องการการโต้ตอบของผู้ใช้เพื่อให้ได้ผลที่น่าพอใจ อย่างไรก็ตามการปรับปรุงภาพยังคงเป็นหนึ่งในหัวข้อที่สำคัญที่สุดเพราะพบการประยุกต์ใช้งานอย่างมากในการประมวลผลภาพทางดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 การบีบย่าน(Range Compression)

ในบางครั้งย่านที่ใช้งาน(Dynamic Range) ของภาพมีขนาดใหญ่มาก ตัวอย่างในย่านการใช้งานของภาพที่ถูกแปลงแบบยูนิทารีมักจะมีขนาดใหญ่จนกระทั่งมีพิกเซลจำนวนน้อยเท่านั้นที่สามารถเห็นได้ ย่านการใช้งานสามารถถูกบีบโดยใช้การแปลงลอกการิทึม(Logarithm)

$V=C\log_{10}(1+|U|)$

โดยที่ C คือค่าคงที่การสเกล การแปลงนี้จะช่วยเพิ่มพิกเซลที่มีขนาด(Magnitude) เล็ก เมื่อเปรียบเทียบกับพวกขนาดใหญ่ดังรูป

2.2.2 การหักลบภาพและตรวจการเปลี่ยนแปลง(Image Subtraction and Change Detection)

ในการประยุกต์การใช้งานในการประมวลผลภาพหลายอย่าง เราอาจต้องการเปรียบเทียบภาพที่มีความสลับซับซ้อนสองภาพ วิธีที่ง่ายแต่มีประสิทธิภาพคือการเรียง(Align)ทั้งสองเข้าด้วยกันแล้วทำการหาผลต่าง จากนั้นก็ทำการปรับคุณภาพภาพผลต่าง ตัวอย่างเช่นอุปกรณ์ที่หายไปจากบอร์ดวงจรสามารถตรวจหาได้โดยการลบภาพชิงบอร์ดดังกล่าวกับบอร์ดที่มีอุปกรณ์วางถูกต้อง การประยุกต์ใช้งานอีกอย่างพบใช้ในกรณีที่เราต้องการทำภาพของหลอดเลือดและเส้นเลือดภายในร่างกาย ซึ่งทำได้โดยการฉีดสารทึบแสงเข้าไปในหลอดเลือดแล้วทำการถ่ายภาพเอ็กซ์เรย์จากนั้นทำการลบภาพออกจากภาพเอ็กซ์เรย์ของหลอดเลือดก่อนฉีดสารทึบแสง ผลต่างของภาพทั้งสองจะแสดงทางเดินของเลือดอย่างชัดเจน การประยุกต์ใช้งานอื่นๆคือการตรวจจับความแตกต่างในระบบรักษาความปลอดภัย หรือการตรวจแผ่นปรินท์วงจรโดยอัตโนมัติ และอื่นๆ

2.3 Shape Analysis การวิเคราะห์รูปร่าง

เทคนิคการวัดปริมาณและคุณภาพได้ถูกพัฒนาขึ้นเพื่อหาคุณลักษณะของรูปร่างของวัตถุในภาพ เทคนิคเหล่านี้มีประโยชน์ในการแยกแยะวัตถุ (Classifying) ที่ใช้ในระบบจดจำรูปแบบ (Pattern Recognition System)รวมถึงการอธิบายวัตถุในระบบเรียนรู้ภาพ (Image Understanding System) บางเทคนิคสามารถใช้กับภาพขาว-ดำ (Binary) ในขณะที่บางเทคนิคสามารถใช้กับภาพระดับเทา (Gray-level) ในที่นี้จะนำมาเฉพาะเรื่องที่เกี่ยวพันธ์กับโครงงาน

คุณสมบัติด้าน Topological (Topological Attributes)คุณสมบัติด้าน Topological ของรูปร่างเป็นคุณสมบัติที่ไม่แปรผันกับการเปลี่ยนแปลงแบบแผ่นยาง (Rubber-Sheet Transformation)การแปลงดังกล่าวมีลักษณะคล้ายกับการยืดแผ่นยางที่มีภาพของวัตถุอยู่ทำให้เกิดการผิดรูป(Spatial Deformed)ขึ้น การแปลงนี้ไม่รวมถึงการตัดแผ่นยางเพื่อย้ายวัตถุหนึ่งไปติดกับอีกวัตถุหนึ่ง เห็นได้ชัดว่าระยะทางแปรผันตามการเปลี่ยนแปลงแผ่นยางเนื่องจากระยะทางมีการเปลี่ยนแปลงตามการยืดหดของแผ่นยาง นอกจากนี้การขนานกัน(Parallelism)และการตั้งฉากกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Perpendicularity) ไม่มีคุณสมบัติด้าน Topological เช่นกัน การเชื่อมต่อกัน (Connectivity) มีคุณสมบัติด้าน Topological

2.4 การแยกวัตถุจากภาพ(Segmentation)

กระบวนการสำคัญอีกขั้นตอนหนึ่งในการประมวลผลเบื้องต้นก่อนจะนำไปสู่ขั้นตอนการจดจำ รูปแบบ ก็คือกระบวนการแยกวัตถุออกจากพื้นหลัง ซึ่งในที่นี้จะเป็นการแยกข้อมูลภาพที่เป็นยานพาหนะออกจากข้อมูลภาพทั้งหมดโดยจะนำรูปภาพนั้นมาลบกัน เพื่อนำไปสู่กระบวนการนับจำนวนวัตถุซึ่งสามารถประมวลผลได้ดีเมื่อมีวัตถุผ่านเข้ามาทีละหนึ่ง

2.4.1 การแยกภาพด้วยการพิจารณาการต่อเนื่องของข้อมูล

เมื่อรับข้อมูลภาพที่ได้จากการเปลี่ยนข้อมูลเป็นรูปแบบไบนารีที่มีค่า 0 กับ 1 แล้ว ซึ่ง *ข้อมูล 0 จะแทนส่วนที่เป็นพื้นหลัง และ ข้อมูล 1 จะแทนส่วนที่เป็นวัตถุหลักการเบื้องต้นคือ* หาค่าพิกเซลที่ เป็น 0 ที่ต่อเนื่องกันตลอดทั้งแนวตั้งและแนวนอนทำให้ได้ขนาดของกรอบ(Block) ข้อมูลภาพวัตถุที่มี ขนาดต่างๆกัน จากนั้นจะพิจารณาเลือกขนาดของกรอบที่ต้องการจากความแตกต่างของจำนวนพิกเซล ความสูงความกว้าง และตำแหน่ง เป็นต้น

2.4.2 การแยกภาพด้วยวิธี Region Labeling

ในการจำแนกภาพโดยวิธีนี้ได้ถือว่าบริเวณที่อยู่ข้างเคียงเป็นบริเวณที่สำคัญมาก จุดภาพที่อยู่ ข้างเคียงกันมักจะมีคุณสมบัติทางสถิติที่คล้ายกันหรือใกล้เคียงกันสำหรับจุดรอบข้างที่มาเชื่อมต่อกัน ในวิธีนี้จะทำการพิจารณาภาพบริเวณย่อยๆจำนวนมากจากนั้นพื้นที่ติดกันจะถูกนำมาพิจารณาถึงความ เป็นเนื้อเดียวกันร่วมกัน การรวมตัวกันจะสิ้นสุดเมื่อพื้นที่ข้างเคียงไม่สามารถพิจารณาถึงความเป็นเนื้อ เดียวกันได้ แต่ถ้าจุดของภาพที่อยู่ใกล้เคียงกันนั้นตรวจสอบแล้วไม่อยู่ในเกณฑ์การรวม จุดภาพนั้นจะ ไม่ถูกรวมเข้าไปในส่วนนั้นของภาพแต่จะถูกเลือกให้เป็นจุดเริ่มต้นของส่วนถัดๆต่อไป และหลังจากที่ จุดภาพทุกจุดได้รวมตัวกันเป็นกลุ่มเรียบร้อยแล้ว

ในกรณีนี้จะกล่าวถึงภาพที่มีวัตถุในภาพมาก วิธีที่จะแยกวัตถุออกจากกันจะทำได้โดยพิจารณา จากการติดกันของพิกเซลที่เป็น 1 โดยสามารถพิจารณาได้ดังนี้

- การติดกันแบบ 4 จุด จะพิจารณา 4 พิกเซลรอบข้างทางด้านแนวนอน และแนวตั้ง
- การติดกันแบบ 8 จุด จะพิจารณา 8 พิกเซลรอบทางด้านแนวนอนและแนวตั้ง

วิธีการแยกวัตถุแบบ Region Labeling นี้ จะพิจารณาแบบไบนารีเฉพาะพิกเซลที่มีค่า 1 ทีละแถวจากซ้ายไปขวาและจากบนลงล่าง ซึ่งเมื่อพิจารณาที่พิกเซลใดพิกเซลหนึ่ง ถ้าพิกเซลแถวบนและทางซ้ายซึ่งผ่านการกำหนด Label แล้วมีค่าไม่เป็น 1 ก็จะกำหนด Label เหมือนกับพิกเซลข้างเคียงที่เป็น 1 แต่ถ้าในกรณีที่พิกเซลข้างเคียงมีค่า 1 มากกว่า 1 พิกเซล และแต่ละพิกเซลมี Labelต่างกัน ก็จะกำหนด Label ให้กับพิกเซลที่พิจารณาอยู่ให้เหมือนกับ Label ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมมูลกันให้เหมือนกันซึ่งจะสามารถ ทราบถึงความแตกต่างของแต่ละวัตถุในภาพ โดยดูจาก Label ที่แตกต่างกัน

การทำการประมวลผลรูปภาพต้องทำการศึกษาพื้นฐานของรูปภาพก่อนว่ามีการสร้างอย่างไร ซึ่งในรูปภาพจะมีอยู่องค์ประกอบหนึ่งที่สามารถนำมาประมวลผลได้ คือ "พิกเซล (Pixel)" พิกเซลจะมี ลักษณะที่เป็นจุดสี่เหลี่ยมเล็กๆ ที่มีอยู่มากมายอยู่บนรูปภาพ ดังนั้นการประมวลผลภาพต้องนำพิกเซล แต่ละรูปภาพมาทำการประมวลผล โดยแต่ละพิกเซลของรูปภาพจะประกอบด้วยรายละเอียดสีต่างๆ มารวมกันเป็นพิกเซล ซึ่งเรียกกลุ่มของสีที่ใช้ว่า "Color Space" ซึ่งรูปแบบการบอกสีนั้นมีอยู่หลายรูปแบบ

2.5 ภาพดิจิตอล(digital image)

คำว่าภาพขาวดำ (Monochrome) หรือที่เรียกง่ายๆว่าภาพ หมายถึงฟังก์ชั่นความเข้มแสงสองมิติ $f(x,y)$ โดยที่ x และy แทนพิกัดเกี่ยวกับระยะทาง(Spatial Coordinate)และค่าของ f ที่จุดใดๆ(x,y)เป็นสัดส่วนกับความสว่าง(หรือระดับเทา)ของภาพที่จุดนั้นๆ ดังรูป แสดงแกนของภาพที่ใช้สังเกตความแตกต่าง

ระหว่างแกนของภาพและแกนของระบบพิกัดปกติ บางครั้งเราอาจมองภาพในลักษณะเพอเสปคทีฟ(Perspective) โดยมีค่าความเข้มแสงแทนแกนที่ 3 การมองในลักษณะนี้ทำให้ภาพมีลักษณะเป็นยอดหลายยอดในบริเวณที่ค่าความสว่างมีความเปลี่ยนแปลงเพียงเล็กน้อย

รูปที่2.1 แสดงแกนของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพดิจิตอลคือภาพf(x,y)ซึ่งถูกทำให้เป็นค่าดิสครีตทั้งค่าพิกัดและค่าความสว่าง การแทนข้อมูลความสว่างด้วยเลขขนาด 8 บิทค่าระดับเทาที่ "0" แทนความสว่างน้อยหรือสีดำ ค่าระดับเทาที่ "255" แทนความสว่างมากหรือสีขาว ภาพดิจิตอลขนาด M x N อาจถูกพิจารณาว่าเป็นเมทริกซ์ขนาด M x N ซึ่งค่าอินเด็กซ์ของเมทริกซ์(แถวและคอลัมน์) ระบุในภาพและค่าของเมทริกซ์ตำแหน่งดังกล่าวแทนค่าระดับเทาที่จุดนั้น หน่วยที่เล็กที่สุดของภาพดิจิตอลมีชื่อว่า"พิกเซล"

สำหรับกรณีภาพดิจิตอลสีขนาด M x N เราจะใช้เมทริกซ์ขนาด M x N จำนวน 3 เมทริกซ์ สำหรับ R G และ B ในการแทนค่าสีที่จุดใดๆ โดยที่ค่าสีที่จุดใดๆซึ่งเกิดจากการผสมสีจะได้มาจากค่าในเมทริกซ์ทั้งสาม ตัวอย่างเช่นค่าในเมทริกซ์ R G และ B สำหรับสีแดงคือ (255,0,0) เป็นต้น

2.6 ไฟล์ข้อมูลภาพชนิด bmp

รูปแบบของไฟล์ข้อมูล bmp เป็นฟอร์แมทของวินโดว์ Bitmapped ซึ่งเป็นมาตรฐานสำหรับไฟล์กราฟฟิกบนวินโดว์ ซึ่งใช้ในการตัดต่อหรือสำเนาต่างๆลงบนโปรแกรม Clipboard โครงสร้างของไฟล์ bmpจะประกอบด้วย 3 ส่วนคือ

- ข้อมูล header
- ข้อมูล palette
- ข้อมูลภาพ

ข้อมูลheader คือ ข้อมูลที่อยู่บริเวณส่วนหัวของไฟล์ซึ่งประกอบไปด้วยข้อมูลทั่วไปรายละเอียดของภาพ เช่น ความกว้าง ความสูง จำนวนสี จำนวนบิท ความละเอียด

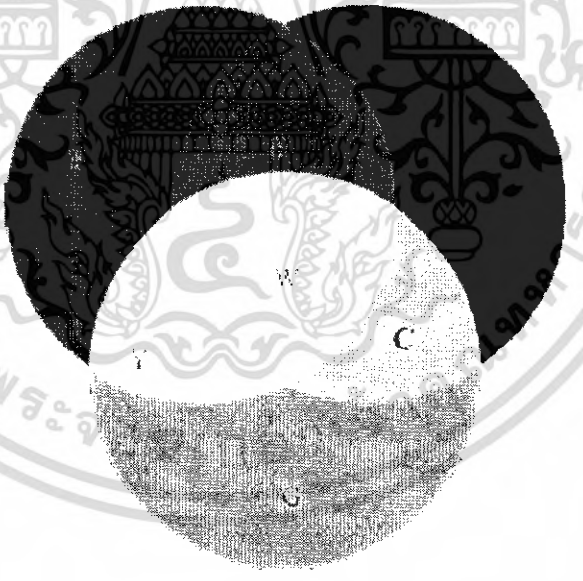
ข้อมูล palette คือ ข้อมูลที่บอกถึงชุดของจานสีที่เกิดจากการผสมแม่สีทั้ง3 คือ แดง เขียว น้ำเงิน มาผสมกันได้สีต่างๆตามจำนวนสีของภาพ เช่น รูปขนาด 4 บิต จะมี 16 สี รูปขนาด 8 บิต จะมี 256 สี ถ้ามีจำนวนสีน้อยๆ ก็จะมีการเก็บค่าจานสีนี้ลงไฟล์ด้วย แต่ถ้าเป็นรูปประเภท 24 บิท จะไม่มีค่าจานสี แต่จะใช้วิธีเก็บแม่สีทั้งสามลงไปเป็นข้อมูลแทน เพราะเก็บค่าจานสีที่มีค่าถึง 16.7 ล้านสี จะเปลืองพื้นที่มาก ไฟล์บิตแมปจะเก็บค่าจานสีชุดละ 4 ไบต์ แต่ใช้เพียง 3 ไบต์

ข้อมูลภาพ คือ ข้อมูลสีแต่ละจุดที่มาประกอบเป็นภาพ ค่าที่เก็บจะเป็นค่าที่ใช้ชี้บอกว่าเมื่อมีค่านั้นๆแล้วไปเปิดดูตารางจานสีว่าที่ค่านั้นๆ นั้นมีค่าสีเป็นเท่าไร เช่น แดง = 100, เขียว = 0, น้ำเงิน = 0 ก็จะได้จุดนี้เป็นสีแดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนบิตที่ใช้เก็บสีต่อหนึ่งจุด	Color Mode Name	จำนวนสีที่แสดงได้
1	Black and White	2
4	16 Color(EGA)	16
8	Pseudo Color	256
16	Hi-Color	65,536
24	True-Color	16,777,216

ตารางที่2.1 แสดงการกำหนดระดับสี



รูปที่2.2 แสดงการผสมสีในรูปแบบ RGB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.7 ชนิดของภาพ (Image Type)

- ภาพความเข้มแสง(Intensity images)
- ภาพไบนารี (Binary images)
- ภาพอินเด็กซ์ (Indexed iamges)
- ภาพสี (RGB images)

ภาพความเข้มแสง(Intensity Images)

ภาพความเข้มแสงคือเมทริกซ์ของข้อมูลซึ่งค่าที่ถูกสเกลลงให้เป็นค่าความเข้มแสง เมื่อสมาชิกของค่าในภาพความเข้มแสงเป็นชนิด unit8 หรือ unit16 ค่าของภาพจะอยู่ในช่วง[0,255] และ[0,65535]ตามลำดับ

ภาพไบนารี(Binary Images)

ภาพไบนารีคือ อาร์เรย์ของข้อมูลแบบตรรกะ(Logical Array)ที่มีค่า 0 หรือ 1 ภาพของข้อมูลชนิด unit8 ที่มีแต่ค่า 0 หรือ 1 ไม่ถือว่าเป็นภาพไบนารี

ภาพอินเด็กซ์ (Index Images)

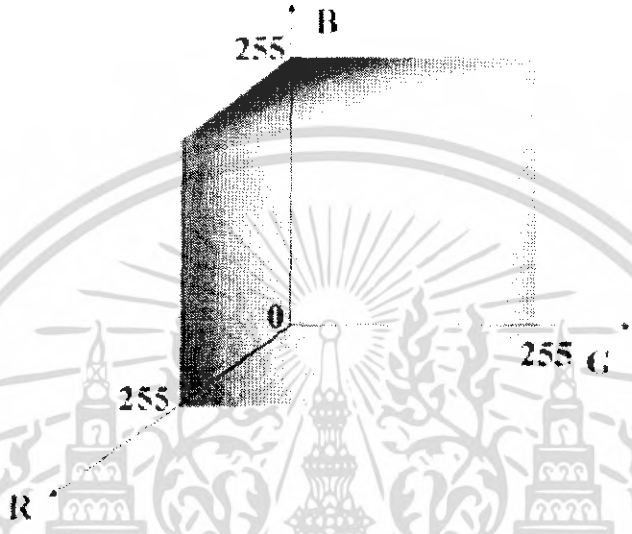
ภาพอินเด็กซ์มีสององค์ประกอบได้แก่เมทริกซ์ข้อมูลของจำนวนเต็ม x และเมทริกซ์ของคัลเลอร์แม็พ(Colormap) เมทริกซ์คัลเลอร์แม็พเป็นอาร์เรย์ขนาด m x 3 ที่ตัวเลขเป็นแบบ floating-point อยู่ในช่วง [0,1] ขนาด m เท่ากับจำนวนของสีที่ใช้กำหนดแต่ละแถวหมายถึงองค์ประกอบสีแดง เขียวและน้ำเงินของสีที่กำหนด ภาพอินเด็กซ์ใช้การแม็พโดยตรงของค่าความเข้มแสงของพิกเซลไปยังค่าของคัลเลอร์แม็พ นั่นคือใช้ค่าของ x ของพิกเซลใดๆเป็นตัวชี้ไปยังเมทริกซ์คัลเลอร์แม็พ ในกรณีที่จำนวนสี(ค่า m) มีค่าน้อยกว่าค่าของ x การแม็พจะไม่เป็นแบบหนึ่งต่อหนึ่ง ในกรณีต้องทำการแบ่ง x ออกเป็นช่วงๆ จำนวน m ช่วง ช่วงที่1 จะชี้ไปที่แถวแรกของเมทริกซ์คัลเลอร์แม็พ ช่วงที่สองจะชี้ไปที่แถวสองของเมทริกซ์คัลเลอร์แม็พ เป็นเช่นนี้เรื่อยไป

ภาพสี (RGB Images)

ภาพสี RGB คืออาร์เรย์ขนาด M x N x 3 ของพิกเซลสีโดยที่แต่ละพิกเซลสีคือค่า Triplet สอดคล้องกับองค์ประกอบสีแดง เขียว และน้ำเงินของภาพ RGB ที่ตำแหน่งที่ระบุ ภาพสีอาจมองว่าเป็นชั้น (Stack) ของภาพระดับเทาสามภาพ ที่ถูกป้อนให้กับอินพุทของจอมอนิเตอร์สี เพื่อทำให้เกิดสีบนจอภาพ โดยมาตรฐานแล้วภาพทั้งสามที่เป็นส่วนของภาพสี หมายถึงภาพองค์ประกอบสีแดงเขียวและน้ำเงิน คลาสของข้อมูลของภาพองค์ประกอบเป็นตัวกำหนดช่วงของค่า จำนวนบิทที่ใช้แทนค่าองพิกเซลของภาพองค์ประกอบเป็นตัวกำหนด Bit Depth ของภาพสี ตัวอย่างเช่น ถ้าแต่ละภาพองค์ประกอบคือภาพขนาด 8 บิท ภาพสีดังกล่าวถูกรียกว่ามีค่า Bit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
72682

Depth ที่ 24 โดยทั่วไปแล้ว จำนวนบิทในทุกองค์ประกอบของภาพมีค่าเดียวกันในกรณีนี้จำนวนของสีในภาพสีคือ $(2^b)^3$ โดยที่ b คือจำนวนบิทในแต่ละภาพองค์ประกอบ สำหรับกรณี 8 บิท จำนวนสีทั้งหมดที่มีได้คือ 16,777,216 สี



รูปที่2.3 แสดงความสัมพันธ์ของสีในกลุ่มสีแบบ R G และ B ใน 1 พิกเซล

2.8 ลักษณะการจัดเก็บข้อมูลภาพแบบดิจิตอล

โดยทั่วไปแล้ว ข้อมูลภาพจะมีค่าความเข้มตั้งแต่ 2 ระดับขึ้นไป แต่ที่ใช้กันมากจะใช้กันที่ค่าระดับความเข้มของจุดภาพเท่ากับ 256 ระดับ ซึ่งจะทำให้ค่าของจุดภาพอยู่ในช่วง(0-255)โดยใช้เนื้อที่การเก็บข้อมูลขนาด 1 ไบต์ หรือ 8 บิต สำหรับข้อมูล 1 จุดภาพ ในกรณีที่ต้องการภาพที่มีความละเอียดของระดับความเข้มสูงๆ อาจจะต้องการจำนวนบิตสำหรับการเก็บข้อมูลมากกว่า 8 บิต คืออาจจะเป็น 16 หรือ 24 บิต โดยค่าความเข้มของจุดภาพจะเท่ากับ 2^{16} และ 2^{24} โดยจะแยกให้เห็นอย่างชัดเจนดังนี้

- ภาพ 2 ระดับ คือ มีเพียงแค่จุดขาวกับดำเท่านั้น โดยแต่ละจุดภาพเป็นข้อมูลขนาด 1 บิต
- ภาพ 16 ระดับ คือ ในแต่ละจุดภาพจะมีขนาดของข้อมูล 8 บิต ซึ่งทำให้สามารถแสดงภาพได้ 256 ระดับสี หรือ 256 เกรย์สเกล ขึ้นอยู่กับภาพนั้นเป็นภาพสีหรือภาพขาวดำ
- ภาพ 256 ระดับ คือ ในแต่ละจุดภาพจะมีขนาดของข้อมูล 8 บิต ซึ่งทำให้สามารถแสดงภาพได้ 256 ระดับสี ขึ้นอยู่กับว่าภาพนั้นเป็นภาพสีหรือภาพขาวดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ภาพ ทรูคัลเลอร์ คือ ในแต่ละจุดภาพจะมีขนาดของข้อมูล 24 บิต ทำให้สามารถแสดงผลภาพได้เหมือนภาพจริงที่สุด เพราะสามารถแสดงสีได้ถึง 16,777,216 สี

2.9 การแปลงระดับเกรย์

ประโยชน์ของการแปลงระดับเกรย์

- การเปลี่ยนแปลงความสว่าง(Global alterations in brightness)

  ทำได้โดยการใช้ค่าคงที่บวกหรือลบออกจากทุกจุดของพิกเซลของภาพ เพื่อเพิ่มหรือลด ความสว่างของภาพ

- การทำเทรชโฮล (Threshold)

  เป็นการเปลี่ยนแปลงหรือหาแนวโน้มของค่าระดับเกรย์ในภาพเพื่อให้ค่าระดับเกรย์เกิด ความไม่ต่อเนื่องมากขึ้น โดยจะนำภาพมาทำฮิสโตแกรม แล้วกำหนดค่าระดับเกรย์ที่แน่นอนขึ้นมาเพื่อที่จะเป็นจุดตัดในการทำเทรชโฮล จากนั้นจะทำการตัดหรือปิดส่วนของระดับเกรย์ที่เราไม่ต้องการออกไป

- บันซ์ซิ่ง(Bunching)และการควอนไตซ์(Quantize)

  ใช้เพื่อลดระดับเกรย์ที่ไม่ต้องการในภาพลง ในบางครั้งบันซ์ซิ่งมีความหมายรวมถึงการควอนไตซ์ด้วย

- สปลิตติ้ง(Splitting)

  เป็นการเพิ่มความแตกต่างระหว่าง 2 กลุ่มของระดับเกรย์ ยกตัวอย่างเช่น หากเรามีตัวอักษรเขียนอยู่บนฉาก โดยตัวอักษรมีระดับเกรย์ที่ 98 และฉากมีระดับเกรย์ 99 ซึ่งตาของมนุษย์ไม่สารถแยกได้ ดังนั้นจึงแก้ปัญหาโดยการทำการสปลิต ฮิสโตแกรม โดยการดึงค่า 99 เป็น 120 และดึงค่า 98 ลงมาเป็น 80 ซึ่งก็จะทำให้ระหว่างตัวเลขและตัวอักษรมีความแตกต่างกันมากขึ้นจนสามารถสังเกตและแยกแยะได้

ระดับสีเทา (Grey Scale) หากเราต้องการค่าข้อมูลที่ละเอียดมากขึ้น ก็จำเป็นที่จะต้องเพิ่มจำนวนบิตการแสดงค่าของแต่ ละพิกเซล ยกตัวอย่างเช่น หากแบ่งความเข้มของการส่องสว่างให้มี 4 ระดับ ก็ต้องใช้เลขฐานสอง จำนวน 2 บิต และจำนวน 4 บิตสำหรับ 16 ระดับ, 8 บิตสำหรับ 256 ระดับ ซึ่งจำนวนระดับที่ใช้ใน ระดับสีเทาหรือระดับเกรย์นี้มักเป็นเลขยกกำลังของ 2, ค่าที่ต่ำที่สุดคือ 0 กำหนดให้เป็นสีดำ และ 1 หรือ ตัวเลขที่น้อยกว่าค่าสูงสุดของระดับเกรย์อยู่ 1 (เช่น 15 สำหรับระดับเกรย์ 16 ระดับ) แทนสีขาว ค่าที่ กำหนดให้ในแต่ละพิกเซลนี้นิยมใช้เป็นจำนวนเต็ม ในยุคแรก ๆ ของระบบการมองเห็นภาพจะใช้ระบบเลขฐานสอง แต่ในปัจจุบันเทคโนโลยีไมโครโปรเซสเซอร์เข้ามามีบทบาทมากขึ้น การแบ่งระดับเป็น 16, 64 หรือ 256 เป็นเรื่องธรรมดาแต่ทั้งนี้ ในการมองเห็นของมนุษย์จะสามารถแยกแยะความแตกต่างได้เพียง 10-15 ระดับเท่านั้น การแบ่งโดย ละเอียดเป็น 64 หรือ 256 ระดับอาจนำไปประยุกต์ใช้กับงานการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประมวลผลภาพหรือกระบวนการทาง อุตสาหกรรมอื่น ๆ

ซึ่งจะเห็นว่าจำนวนระดับเกรย์จะเป็นตัวจำกัดรายละเอียดของภาพ โดยทั่วไปแล้วยิ่งแบ่งระดับ เกรย์เป็นหลายระดับก็จะเป็นการเพิ่มคุณภาพของภาพด้วย และการเพิ่มจำนวนพิกเซล เช่น จาก 35 * 35 เป็น 250 * 250 ก็จะเป็นการเพิ่มความละเอียด (Resolution) และรายละเอียด (Detail) ของภาพเช่นกัน จะเห็นว่าจะแตกต่างกับการขยายภาพ (Zoom) โดยการขยายภาพนี้เป็นการขยายแต่ละพิกเซลให้ใหญ่ขึ้น ไม่ได้เป็นการเพิ่มจำนวนพิกเซล

2.10 การสร้างภาพไบนารี

อุปกรณ์ที่มีความสามารถในการแสดงผลได้แค่ 2 ระดับ หรือ 2 สี คือ สีขาวกับสีดำยังมีการใช้ กันอย่างแพร่หลาย เช่น เครื่องพิมพ์ (Printer), เครื่องโทรสาร (Fax), จอภาพแสดงผลแบบโมโนโครม (Monochrome Monitor) เป็นต้น เนื่องจากอุปกรณ์เหล่านี้เป็นอุปกรณ์ที่มีราคาถูก ดังนั้นจึงเกิดปัญหาใน การแสดงผลขึ้น การที่จะแสดงผลหรือพิมพ์รูปภาพที่มีระดับความเข้มของภาพหลายระดับ ซึ่งมีมากกว่าความสามารถในการแสดงผลของอุปกรณ์เหล่านั้นที่มีเพียงแค่ 2 ระดับ เราจะต้องทำการแปลง ข้อมูลภาพให้เป็นภาพไบนารี (Binary Image)เสียก่อน ซึ่งการสร้างภาพไบนารีนั้นก็หมายถึงการ แปลงข้อมูลภาพที่มีระดับความเข้มหลายระดับ (Multilevel Image) ให้เป็นภาพที่มีระดับความเข้มเพียง 2 ระดับ นั่นคือ 1 จุดภาพ มีได้ 2 ค่าเท่านั้น คือ 0 กับ 1 โดยจุดภาพที่แทนด้วย 0 หมายถึงจุดภาพที่มีสีดำ ส่วนจุดที่แทนด้วย 1 จะหมายถึงจุดภาพที่มีสีขาว เมื่อทำการแปลงเป็นภาพไบนารีแล้ว จึงสามารถนำ ภาพนั้นไปแสดงผลบนอุปกรณ์เหล่านั้น จะเห็นได้ว่าการแปลงข้อมูลภาพหลายระดับเป็นภาพไบนารีจึง มีความจำเป็นและมีประโยชน์มากในการแสดงผลภาพที่มีความเข้มของภาพหลายระดับบนอุปกรณ์ที่มี ความสามารถในการแสดงผลได้ 2 ระดับ สำหรับประโยชน์ อีกประการหนึ่งในการแปลงข้อมูลภาพนั้น เป็นภาพไบนารี คือการลดเนื้อที่การเก็บข้อมูลภาพ ภาพไบนารีนี้จะสามารถลดเนื้อที่ลงได้ถึง 8 เท่า นั่นคือถ้าเดิม 1 จุดภาพใช้เนื้อที่ในการเก็บ 8 บิต เมื่อทำการแปลงเป็นภาพไบนารีแล้วจะใช้เนื้อที่ในการเก็บ 1 จุดภาพนั้นเพียง 1 บิตเท่านั้น อีกทั้งยังสามารถนำไปประยุกต์ใช้งานได้อย่างแพร่หลาย เช่น นำไป ประยุกต์ใช้ในการวิเคราะห์เอกสารในขั้นตอนที่เรียกว่าการประมวลผลขั้นต้น (Preprocessing) เป็นต้น

ในการสร้างภาพไบนารี สามารถทำได้โดยใช้เทคนิคการทำเทรชโฮล (Thresholding Technique) โดยพิจารณาว่าจุดภาพใดควรจะเป็นจุดขาวหรือจุดดำ จะกระทำโดยการเปรียบเทียบระหว่างจุดภาพเริ่มต้นกับค่าคงที่ค่าหนึ่งที่เรียกว่า ค่าเทรชโฮล (Threshold Value) เทคนิคนี้ใช้กันมาก ในกรณีที่ข้อมูลภาพมีลักษณะแตกต่างกันระหว่างวัตถุ (Object) และพื้นหลัง (Background)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยค่าของ จุดภาพใด ๆ ที่มีค่าน้อยกว่าค่าเทรชโฮลจะถูกกำหนดให้เป็น 0 (จุดดำ) และถ้าค่าของจุดภาพใด ๆ ที่มีค่า มากกว่าหรือเท่ากับค่าเทรชโฮล จะถูกเปลี่ยนให้เป็น 1 (จุดขาว)

ในการสร้างภาพไบนารีโดยใช้เทคนิคเทรชโฮลเพื่อให้ได้ผลลัพธ์ที่เหมาะสมและคมชัด สิ่งที่ สำคัญที่สุดคือ ค่าเทรชโฮล เนื่องจากถ้าเลือกค่าเทรชโฮลที่ไม่เหมาะสม (ค่าเทรชโฮลที่มีค่าน้อยเกินไป หรือมากเกินไป) ภาพที่ได้อาจจะไม่เหมาะสม ขาดความคมชัด และอาจทำให้รายละเอียดบางส่วนขาด หายไป กล่าวคือภาพที่ได้อาจจะมืดเกินไปหรือสว่างเกินไป หรือภาพที่ได้มีสิ่งรบกวน (Noise) เกิดขึ้น อันเป็นผลทำให้ภาพผลลัพธ์ที่ได้ไม่สวยงามเท่าที่ควร ดังนั้นปัญหาของการสร้างภาพไบนารีโดยวิธีเทรชโฮลนี้คือ ทำอย่างไรจึงจะสามารถคำนวณหาค่าเทรชโฮลที่เหมาะสมสำหรับแต่ละภาพที่จะนำมาทำการสร้างภาพไบนารี ซึ่งมีวิธีคำนวณหาค่าเทรชโฮลหลายวิธี โดยแต่ละวิธีเหมาะสมกับลักษณะการ ทำงานที่แตกต่างกันไป เช่น การหาค่าเทรชโฮลโดยการกำหนดค่าล่วงหน้า(Pre-assigned Threshold Value) การหาค่าเทรชโฮลจากค่ากลาง (Mid-range Threshold Value) ซึ่งแต่ละวิธีอธิบายได้ดังนี้

การหาค่าเทรชโฮลโดยการกำหนดค่าล่วงหน้า (Pre-assigned Threshold Value)

การหาค่าเทรชโฮลโดยการกำหนดค่าล่วงหน้านี้เป็นวิธีที่ง่ายที่สุด เป็นการคำนวณค่าเทรชโฮลโดยการกำหนดเองจากผู้ใช้ ซึ่งการกำหนดนี้จะขึ้นอยู่กับประสบการณ์ของผู้ใช้นั้นๆ ทำได้โดยการเลือกค่าลงที่ค่าหนึ่ง ซึ่งเรียกค่านั้นว่า "ค่าเทรชโฮล" โดยค่าที่เลือกมานี้จะเป็นค่าที่อยู่ระหว่างค่าต่ำสุด และค่าสูงสุดของระดับความเข้มของข้อมูลภาพอินพุท เช่น ภาพข้อมูลอินพุทมีเกรย์สเกล256 ระดับ จะ มีค่าเกรย์สเกลได้ตั้งแต่ 0-255

การหาค่าเทรชโฮลจากค่ากลาง (Mid-Range Threshold Value)

การหาค่าเทรชโฮลโดยพิจารณาจากค่ากลาง เป็นการหาค่าเทรชโฮลที่แตกต่างจากการหาค่าเทรชโฮลวิธีแรก สำหรับวิธีนี้จะเป็นการคำนวณหาค่าเทรชโฮลโดยอัตโนมัติโดยผู้ใช้ไม่ต้องเป็นผู้กำหนดเอง การคำนวณหาค่าได้อาศัยการคำนวณพื้นฐานทางสถิติในเรื่องของการหาค่ากลาง หรือ ค่าเฉลี่ย (Mean) มาประยุกต์ใช้ ซึ่งค่าเทรชโฮลที่คำนวณได้จะเป็นค่าที่เหมาะสมสำหรับภาพนั้นๆ

2.11 ฮิสโตแกรม (Histogram)

ฮิสโตแกรมเป็นการแสดงให้เห็นถึงความถี่ของการนับจำนวนพิกเซล ที่มีค่าความเข้มแต่ละค่า หนึ่ง ๆ ในภาพระดับเกรย์ โดยแกน x ในกราฟแสดงค่าระดับเกรย์ และแกน y แสดงค่าจำนวนพิกเซล

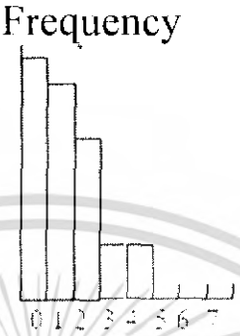
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการสร้างฮิสโตแกรม มีดังนี้

1. ทำการดิจิไตซ์ (Digitizing) ภาพ

2. นับจุดพิกเซลในแต่ละระดับเกรย์

3. พลอตกราฟระหว่างจำนวนจุดกับระดับเกรย์



รูปที่2.4 ภาพฮิสโตแกรมของ 8 ระดับเกรย์

รูปร่างของฮิสโตแกรมสามารถบอกลักษณะบางประการของภาพได้ เช่น หากฮิสโตแกรมมี ลักษณะที่แคบ จะหมายถึงภาพนั้นขาดการแยกแยะความผิดแผกของสี (Contrast) เป็นต้น ฮิสโตแกรมนี้ มีประโยชน์ในการกำหนดค่าเทรชโฮล เพื่อสำหรับทำการเปลี่ยนภาพระดับเกรย์ให้เป็นภาพไบนารี่ ฮิสโตแกรมของภาพแทนความถี่สัมพัทธ์ของการเกิดของแต่ละระดับเทาในภาพ ภาพที่มีคุณสมบัติไม่เหมือนกันมีลักษณะที่ไม่เหมือนกัน

ฮิสโตแกรมของภาพที่มีคุณสมบัติต่างกัน ถ้าภาพที่มีลักษณะโทนสีออกมืด ฮิสโตแกรมจะมีเส้นชิดขอบไปทางด้านซ้าย ถ้าภาพที่มีโทนสีสว่างเส้นกราฟจะอยู่ชิดไปทางด้านขวา ภาพที่มีคอนทราสต่ำโทนของภาพจะมีสีเทา ฮิสโตแกรมจะมีเส้นกราฟอยู่ติดกันในช่วงกลาง ภาพที่มีคอนทราสสูง ฮิสโตแกรมจะกระจายสม่ำเสมอตลอดช่วงของระดับเทา เทคนิคการสร้างแบบฮิสโตแกรมจะทำการเปลี่ยนแปลงภาพเพื่อให้ได้ฮิสโตแกรมตามรูปร่างที่ต้องการ เทคนิคนี้มีประโยชน์ในการยืดบริเวณของภาพที่มีคอนทราสต่ำที่มีฮิสโตแกรมแคบ การสร้างแบบฮิสโตแกรมเป็นเทคนิคที่มีประสิทธิภาพมากในการปรับปรุงภาพ(Image Enhancement)

2.12 กลุ่มสีที่ใช้แสดงผล

การเกิดภาพต่างๆบนจอคอมพิวเตอร์ จะเกิดจากการเรียงตัวของจุดสี่เหลี่ยม สีต่างๆ ที่เราเรียกว่า "พิกเซล" ซึ่ง "พิกเซล" เป็นตัวกำหนดความคมชัดของภาพในจอคอมพิวเตอร์ ยิ่งเรากำหนดให้ภาพแต่ละภาพมีจำนวนพิกเซลมาก ภาพก็จะมีความละเอียดมากขึ้นตามไปด้วย แต่จะทำ ให้ไฟล์ของภาพมีขนาดใหญ่ขึ้นเนื่องจากภาพนั้นมีความละเอียดที่เพิ่มขึ้น พิกเซล (Pixel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาจากคำว่า "Picture Element") เป็นหน่วยพื้นฐานของสีที่สามารถโปรแกรมได้ บนจอแสดงผล หรือบนภาพ ขนาดทางกายภาพของพิกเซลจะขึ้นอยู่กับการกำหนดความละเอียดของจอภาพ ถ้าทำการแสดงผลให้มีความละเอียดสูงสุด ขนาดทางกายภาพของพิกเซลก็จะเท่ากับขนาดทางกายภาพของจุด (Dot Pitch หรือ Dot Size) บนจอภาพ แต่ถ้าตั้งความละเอียดไว้ต่ำกว่า ค่าสูงสุด พิกเซลก็จะมีขนาดใหญ่กว่าขนาดทางกายภาพของจุดบนจอภาพ แสดงว่าแต่ละพิกเซลจะ ใช้มากกว่า 1 จุดในการแสดงผล

ในปัจจุบันกลุ่มสีที่แสดงผลมีมากมายหลายแบบ เช่น กลุ่มสีที่แสดงผล RGB เป็นกลุ่มสีที่ แสดงผลที่แสดงลักษณะของสี 3 สี คือ สีแดง สีเขียว และสีน้ำเงิน ส่วนกลุ่มสีที่แสดงผล HSV เป็น กลุ่มสีที่แสดงผลที่บอกลักษณะของภาพโดยใช้ ระดับสี ความเข้มของสี ความสว่างของสี เป็นต้น

2.12.1 คุณสมบัติของกลุ่มสีที่แสดงผล

โดยทั่วไปแล้วภาพจะอยู่ในรูปแบบของกลุ่มสีที่แสดงผล RGB ซึ่งแทนสีทั้งหมดในกลุ่มสีที่ แสดงผลด้วย เวกเตอร์ 3 มิติ เราจึงใช้กลุ่มสีที่แสดงผลนี้เป็นจุดเริ่มต้นของลักษณะเฉพาะของภาพได้ โดยใช้ลักษณะของสี (Color Feature) จากนั้นค่อยทำการลดระดับของสีให้เหมาะสมต่อการทำ ฮีสโตแกรม คุณสมบัติดังกล่าวคือ

2.12.1.1. Uniformity

Uniformity เป็นคุณสมบัติที่ใช้ในการเปรียบเทียบความใกล้เคียงของเมตริกสีโดยกลุ่มสีที่แสดง จะมีคุณสมบัตินี้จะต้องคำนวณได้ง่าย อย่างเช่น การหาความใกล้เคียงของเมตริกสีโดยใช้ฟังก์ชั่นที่ไม่ ขึ้นกับตำแหน่งในกลุ่มสีนั้น การทำการแปลงสีจะเป็นสีที่ทำให้กลุ่มสีมีคุณสมบัตินี้

2.12.1.2 Completeness

Completeness เป็นคุณสมบัติสำคัญที่กลุ่มสีที่แสดงผลจะต้องมีสีที่เราสามารถแยกแยะได้อย่าง ชัดเจน สามารถทำได้โดยการทำการลดระดับสี

2.12.1.3 Compactness

Compactness เป็นคุณสมบัติที่สีในกลุ่มสีที่แสดงผลจะต้องแตกต่างจากสีอื่นๆ ในกลุ่มสีที่ แสดงผลเดียวกัน คุณสมบัตินี้จะเป็นตัวกำหนดมิติของภาพเพื่อให้แน่ใจว่า เมื่อผ่านการเปรียบเทียบจาก กลุ่มสีที่แสดงผล RGB แล้วจะไม่มี สี 2 สีที่เหมือนกัน การลดระดับสีจะทำให้เกิดคุณสมบัตินี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12.1.4 Naturalness

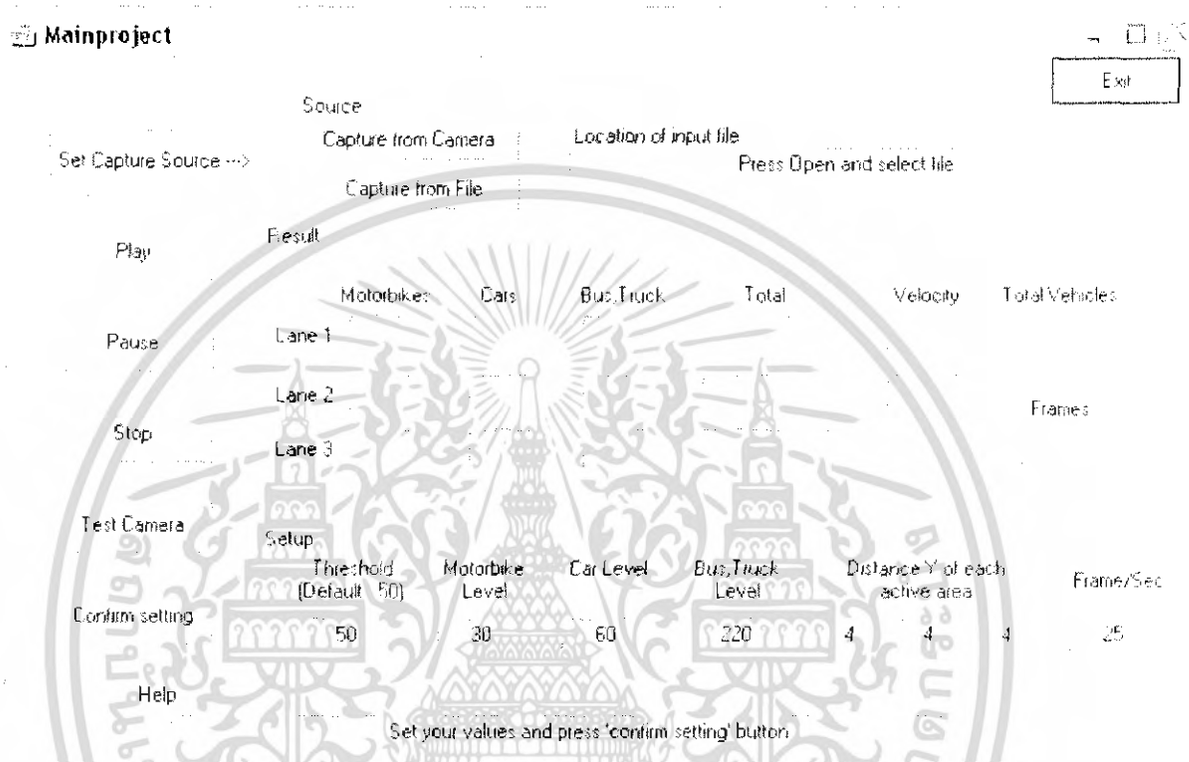
Naturalness เป็นคุณสมบัติของสีที่ผู้ใช้สามารถเข้าใจได้ง่าย ทำให้ผู้ใช้สามารถทำการดึงข้อมูล ในระบบแกน R จะแสดงปริมาณของสีแดงที่มีในพิกเซลนั้น แกน G จะแสดงปริมาณของสี เขียวที่มีในพิกเซลนั้น และแกน B จะแสดงปริมาณของสีน้ำเงินที่มีในพิกเซลนั้น เนื่องจากเก็บสีแยก เป็นคนละค่าของสีกันทำให้เมื่อมีการคำนวณต้องคำนวณแยกทีละค่าของสี การผสมสีบนคอมพิวเตอร์นั้นอาศัยระดับสีหลัก 3 สีคือ สีแดง สีเขียว และ น้ำเงิน หรือที่เราเรียกว่า "RGB Colors" (Red-Green-Blue) นั้นเอง ความเหมือนจริงของสีคอมพิวเตอร์นั้น ขึ้นอยู่กับใน หนึ่งจุดพิกเซลของการแสดงผลนั้นใช้ระดับของสี (Color depths) ที่มีค่าเท่าไหร่ เช่นถ้าสี RGB มีระดับ ของสีเป็น 8 planes นั่นคือ เราใช้ 8 บิตเก็บข้อมูลหนึ่ง สี หมายความว่า เฉพาะระดับสีหลัก เช่น สีแดง สีเดียวก็จะมีความเป็นแดง อยู่ถึง 2^8 = 256 ระดับ พูดง่ายๆ ก็คือว่า สมมติเราจะผสมสี หนึ่ง สี จาก แดง- เขียว-น้ำเงิน (RGB) เราจะใช้สีแดง กี่ส่วนจาก 0 ถึง 255 ส่วน ใช้เขียว กี่ส่วนจาก 0 ถึง 255 และเช่นเดียว กัน สีน้ำเงินกี่ส่วน จาก 0 ถึง 255 สีที่เกิดขึ้นก็จะเกิดจากการผสมของสีทั้งสาม ในอัตราส่วนต่างๆ กัน เช่น สีเหลืองธรรมดา เกิดจากการผสมสี แดง 255 ส่วน สีเขียว 255 ส่วน และสีน้ำเงิน 0 ส่วน จากข้างต้น ระดับของสีแดง มีถึง 256 ระดับ สีเขียว 256 ระดับ สีน้ำเงิน 256 ระดับ ดังนั้น RGB ทั้งหมดใช้ 24 บิต (8+8+8) ในการแสดงสี RGB ของหนึ่งจุดพิกเซลซึ่งสามารถแสดงสีได้มากถึง256 x 256 x 256 = 16.7 ล้านสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่3
หลักการทำงานและการออกแบบ

3.1 หลักการทำงานของโปรแกรม

หน้าตาของโปรแกรมหลัก



รูปที่3.1 หน้าต่างแสดงโปรแกรมหลัก

ปุ่ม Capture from Camera ใช้เพื่อเลือกประมวลผลผ่านกล้องวิดีโอแบบเรียลไทม์

ปุ่ม Capture from File ใช้เพื่อเลือกประมวลผลผ่านไฟล์วิดีโอที่บันทึกไว้

ปุ่ม Play ใช้เพื่อเริ่มการประมวลผลและแสดงภาพวิดีโอที่ทำการประมวลผล

ปุ่ม Pause ใช้เพื่อหยุดการประมวลผลชั่วคราว

ปุ่ม Stop ใช้เพื่อหยุดการประมวลผล

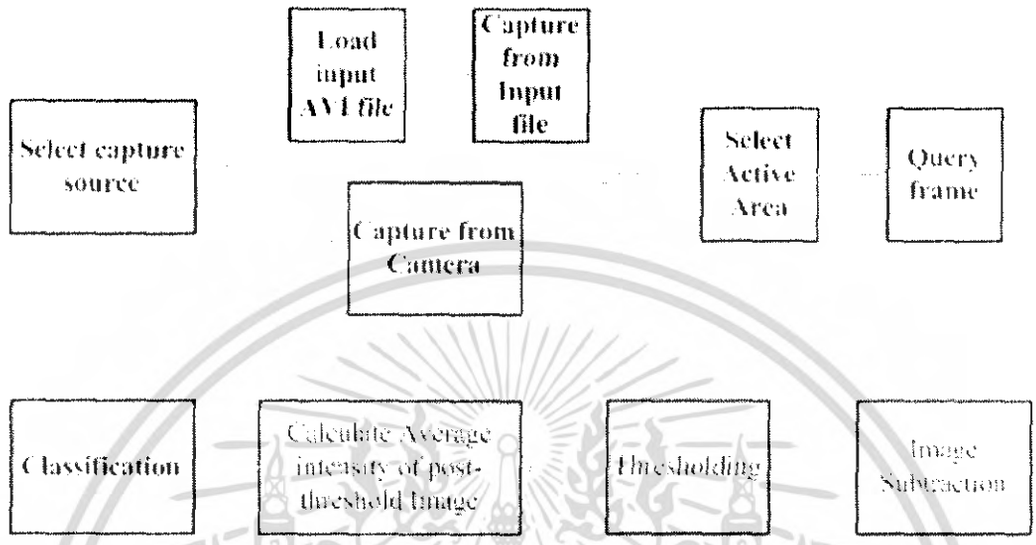
ปุ่ม Test Camera ใช้เพื่อทดสอบภาพจากกล้องวีดีโอ

ปุ่ม Confirm setting ใช้เพื่อยืนยันการตั้งค่าต่างๆในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบ

โครงสร้างโดยรวมของโปรแกรม



รูปที่ 3.2 บล็อคไดอะแกรมแสดงโครงสร้างของโปรแกรม

-Select capture source เลือกประมวลผลผ่านไฟล์วิดีโอหรือประมวลผลแบบเรียลไทม์ผ่านกล้องวิดีโอ

-Load input AVI file เลือกไฟล์อินพุท (.AVI) และกำหนดค่าให้พอยเตอร์ชี้ไปยังตำแหน่งของไฟล์อินพุทนั้น

-Capture from input file โหลดไฟล์วีดีโออินพุทและอ่านวีดีโสตรีมเก็บไว้ในหน่วยความจำ

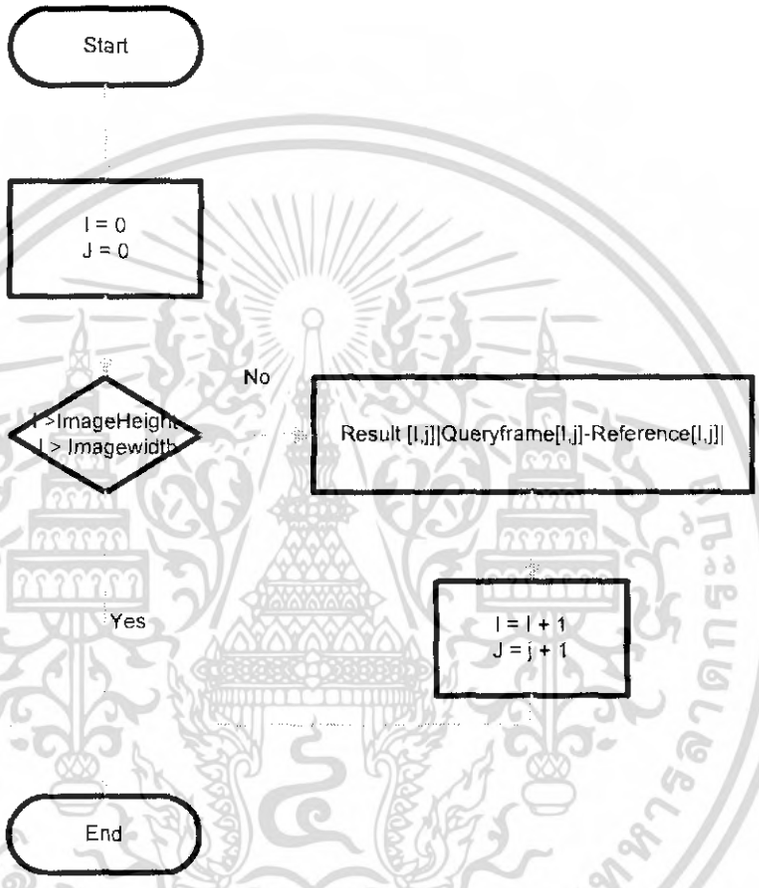
-Capture from Camera อ่านวีดีโสตรีมผ่านกล้องวิดีโอ

-Select Active Area กำหนดขอบเขตพื้นที่ในการประมวลผล

-Query frame ดึงเฟรมแรกจากไฟล์วีดีโอหรือกล้องวิดีโอมาเก็บ และบันทึกภาพในส่วนของActive Areaไว้เป็นภาพอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

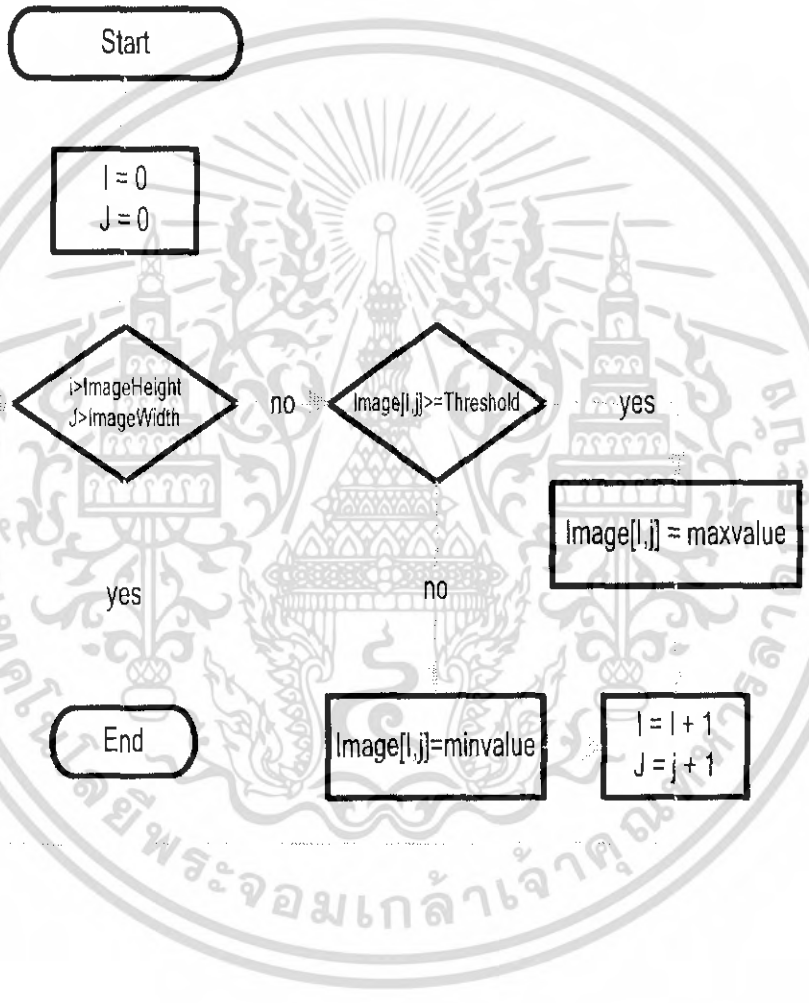
-Image subtraction นำภาพแต่ละเฟรมที่ถูกดึงออกมาจากส่วนของActive Areaของไฟล์วีดีโอหรือกล้องวิดีโอมาลบกับภาพอ้างอิงโดยเป็นการลบกันแบบหาผลต่างสัมบูรณ์(Absolute difference)



รูป ที่ 3.3 บล็อคไดอะแกรมของ Image Subtraction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

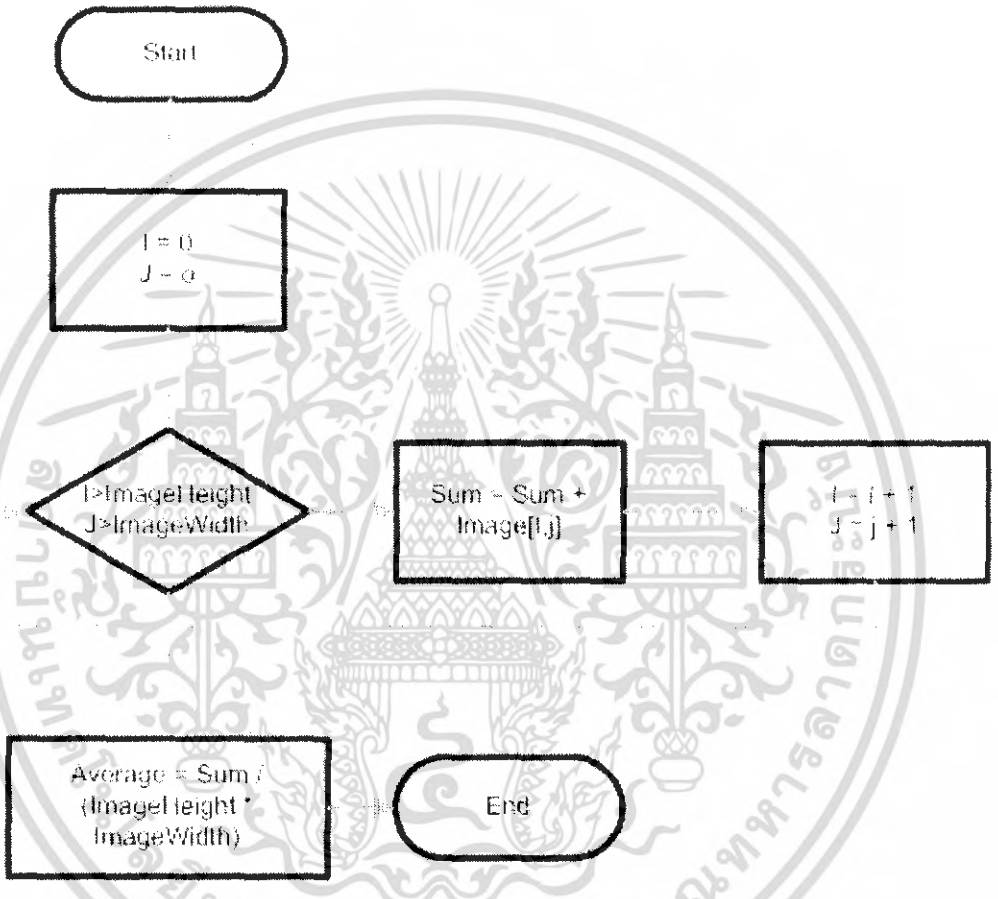
-**Threshold** เป็นการแยกระดับความเข้มแสงของภาพออกเป็นสองค่า คือ ค่าต่ำสุด และค่าสูงสุด โดยเปรียบเทียบกับค่าที่ตั้งไว้(Threshold) โดยเป็นการทำเพื่อกำจัดน๊อยส์จากการลบภาพ



รูปที่3.4 บล็อกไดอะแกรมของ Thresholding each Active Area

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

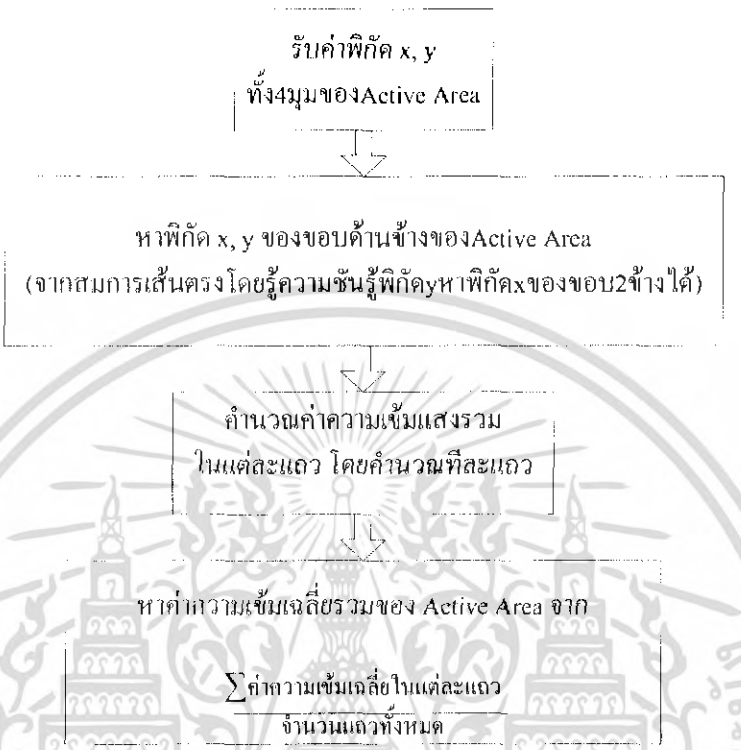
- Calculate Average Intensity of post-Threshold Image เป็นการหาค่าความเข้มแสงเฉลี่ยของภาพที่ผ่านการทำ Threshold แล้ว



รูปที่ 3.5 แสดง flow chart ของการหาความเข้มแสงเฉลี่ยของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

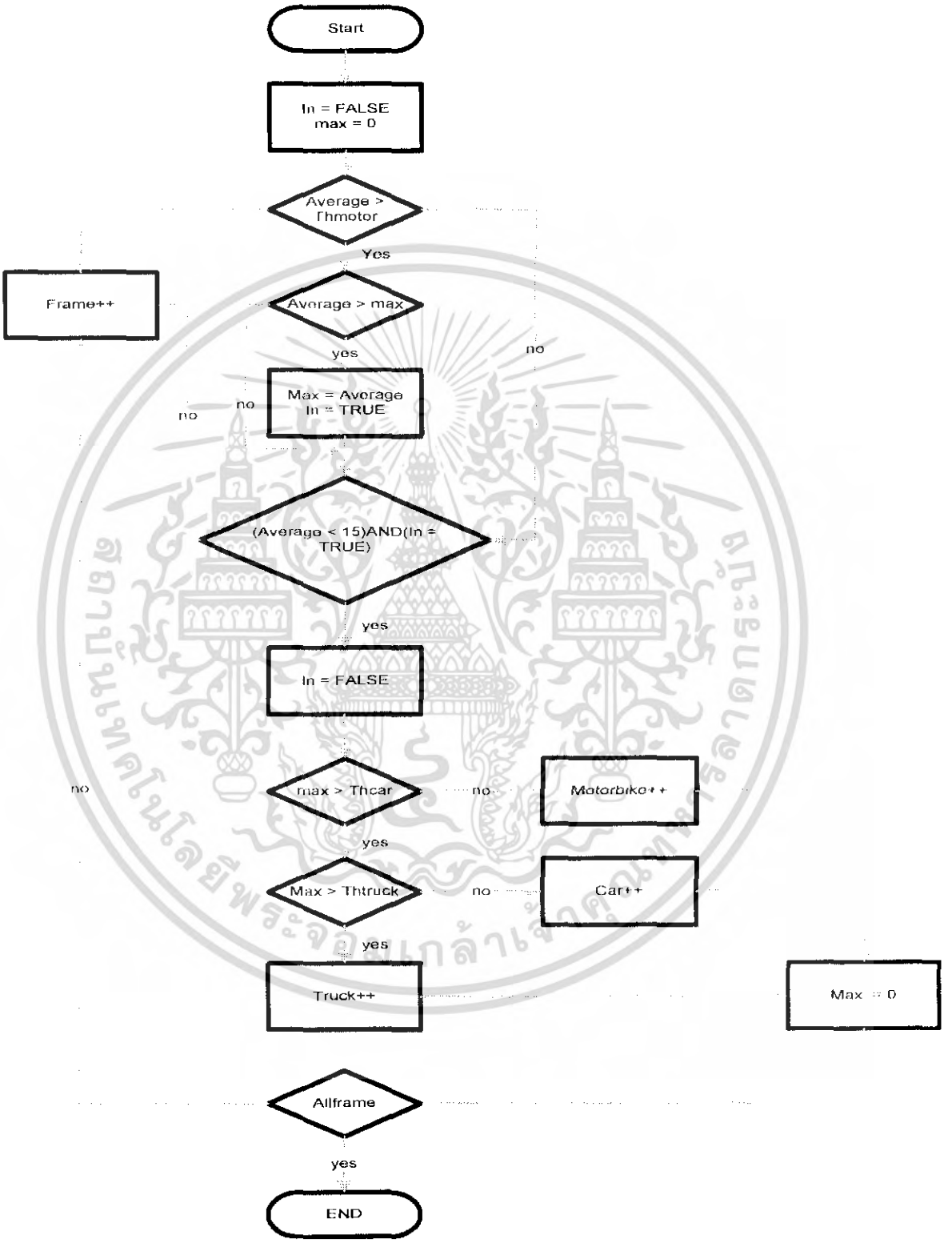
-การหาค่าความเข้มแสงเฉลี่ยของภาพในกรณีที่ใช้กรอบประมวลผลที่กำหนดรูปร่างได้



รูปที่ 3.6 แสดงขั้นตอนการการหาค่าความเข้มแสงเฉลี่ยของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Classification เป็นการนับและจำแนกชนิดของรถ โดยใช้ค่าเฉลี่ยความเข้มแสงที่ได้จากขั้นตอนก่อนหน้าและค่าระดับThreshold ที่ตั้งไว้ของรถประเภทต่างๆเป็นตัวตัดสินการนับประเภทของรถ



รูปที่ 3.7 แสดงขั้นตอนการนับแยกจำนวนรถยนต์

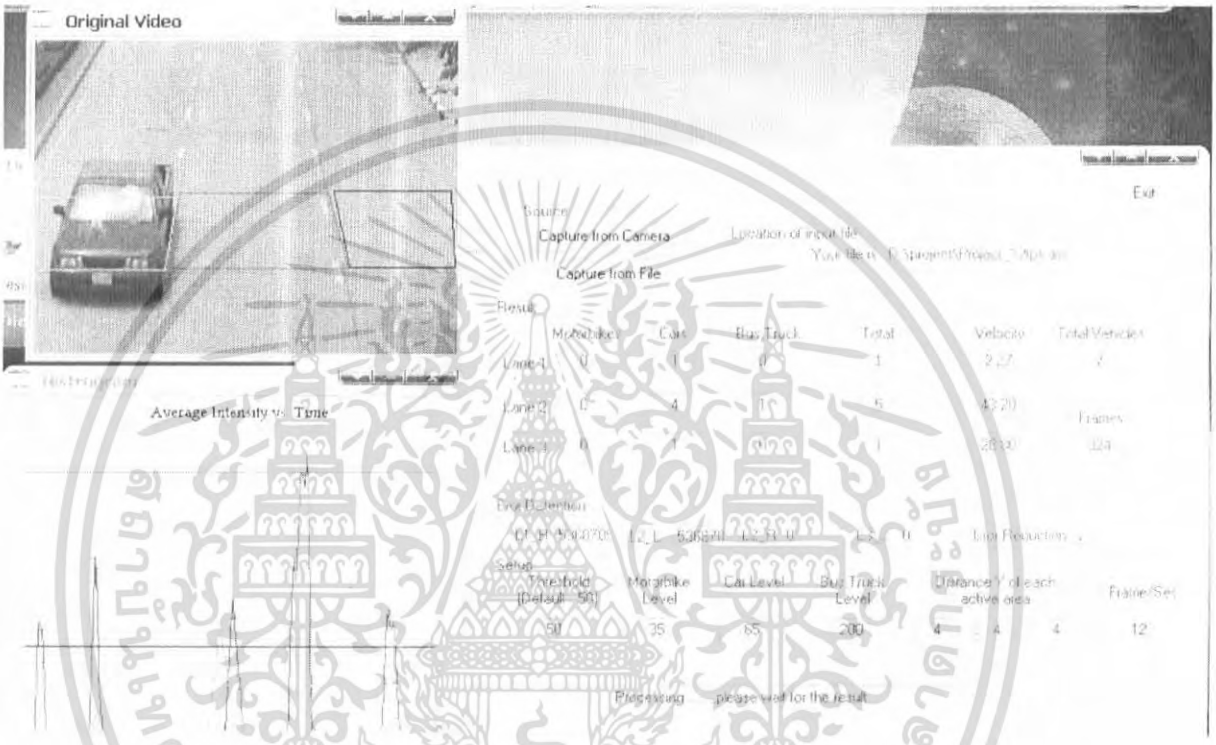
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-การแก้ไขข้อผิดพลาดที่เกิดจากการแล่นคร่อมช่องทางการจราจร

รูปที่ 3.8 แสดงขั้นตอนการแก้ไขข้อผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Display the result** เป็นการแสดงผลจากการประมวลผลของโปรแกรม โดยจะแยกแสดงเป็นตัวเลขบอกจำนวนของยานพาหนะแต่ละประเภท และ แสดงเป็นกราฟที่พลอต ระหว่างค่าความเข้มของแสงเฉลี่ยของภายในเฟรมที่ถูกประมวลผล เทียบกับเวลา



รูปที่ 3.9 แสดงGraphic user interface

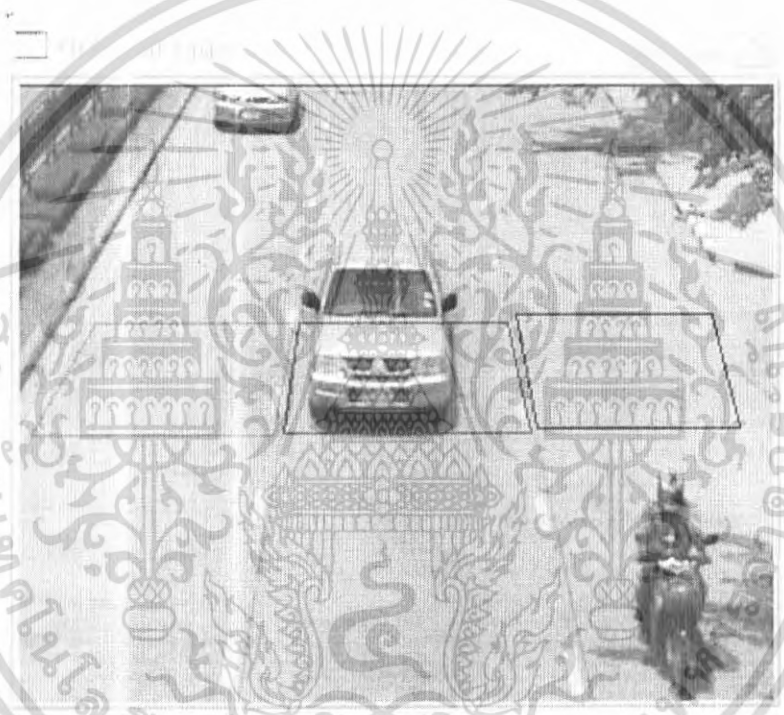
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่4
ผลการทดลอง

4. ผลการทดลองของโครงงาน

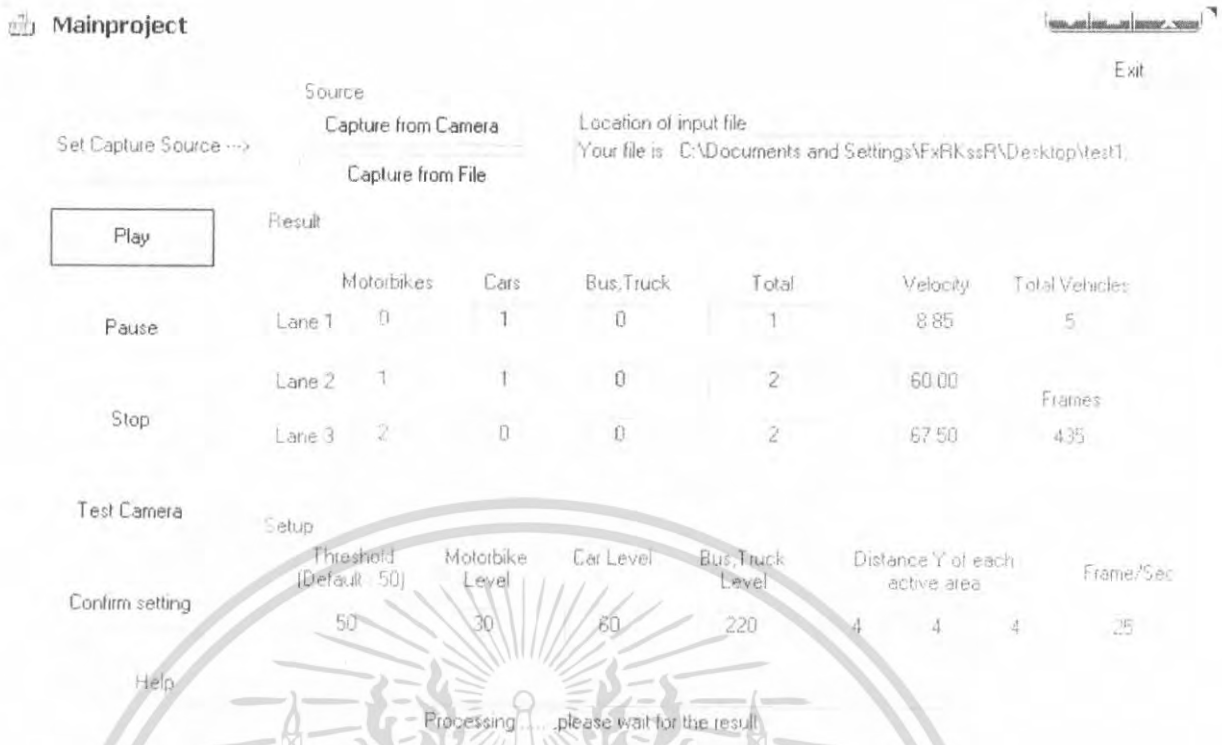
ในที่นี้จะทำการทดลองครั้งละหนึ่งช่องจราจร

4.1 ผลการทดลองการนับจำนวนยานพาหนะโดยประมวลผลจากไฟล์วิดีโอ



รูปที่4.1 ภาพยานพาหนะเคลื่อนที่เข้าสู่บริเวณประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



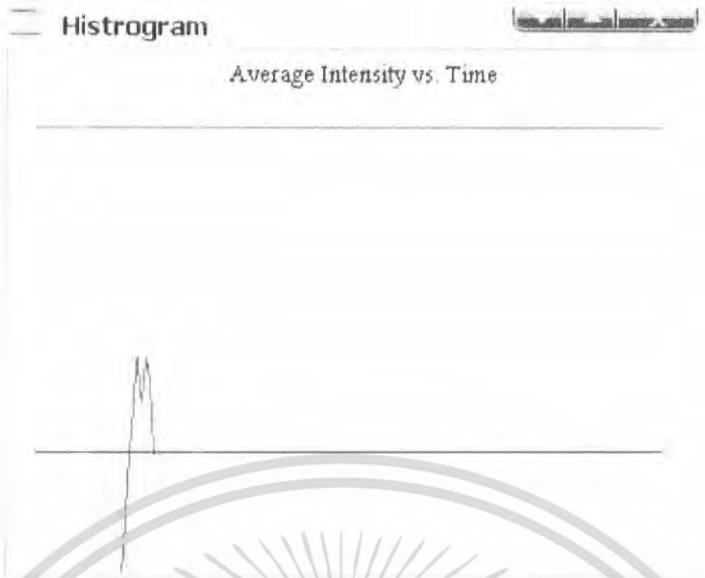
รูปที่4.2 ภาพโปรแกรมขณะประมวลผล

4.3(a) **4.3(b)**

รูปที่ 4.3(a) รูปภาพที่ผ่านการลบด้วยภาพอ้างอิงแล้ว

รูปที่ 4.3(b) รูปภาพหลังจากการลบด้วยภาพอ้างอิงและถูกทำเทรชโชลแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 กราฟแสดงผลขณะมียานพาหนะวิ่งเข้า สู่ ACTIVE AREA

4.1.1 ผลการทดลองการนับจำนวนยานพาหนะในช่องจราจรช่องกลาง

รูปที่ 4.5 รูปภาพแสดงการเลือก Active Area ในการประมวลผลช่องจราจรช่องกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่1.แสดงจำนวนยานพาหนะที่ได้จากการประมวลผลของโปรแกรมในช่องจราจรช่องกลาง

	Motorbikes	Cars	Bus,Truck	Total
ช่องจราจรกลาง	2	5	1	8

ตารางที่ 2. แสดงจำนวนยานพาหนะจริงที่ผ่านช่องจราจรช่องกลาง

	Motorbikes	Cars	Bus,Truck	Total
ช่องจราจรกลาง	2	5	1	8

ตารางที่ 3. แสดง %ความผิดพลาดของโปรแกรมในการนับจำนวนยานพาหนะในช่องจราจรกลาง

	Motorbikes	Cars	Bus,Truck	Total
ช่องจราจรกลาง	0%	0%	0%	0%

4.1.2 ผลการทดลองการนับจำนวนยานพาหนะในช่องจราจรด้านซ้าย

รูปที่ 4.6 รูปภาพแสดงการเลือก Active Area ในการประมวลผลช่องจราจรช่องซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4. แสดงจำนวนยานพาหนะที่ได้จากการประมวลผลของโปรแกรมในช่องจราจรช่องซ้าย

	Motorbikes	Cars	Bus,Truck	Total
ช่องจราจรซ้าย	1	4	0	5

ตารางที่ 5. แสดงจำนวนยานพาหนะจริงที่ผ่านช่องจราจรช่องซ้าย

	Motorbikes	Cars	Bus,Truck	Total
ช่องจราจรซ้าย	1	4	0	5

ตารางที่ 6. แสดง %ความผิดพลาดของโปรแกรมในการนับจำนวนยานพาหนะในช่องจราจรซ้าย

	Motorbikes	Cars	Bus,Truck	Total
ช่องจราจรซ้าย	0%	0%	0%	0%

4.1.2 ผลการทดลองการนับจำนวนยานพาหนะในช่องจราจรด้านขวา

รูปที่ 4.7 รูปภาพแสดงการเลือก Active Area ในการประมวลผลช่องจราจรช่องขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 7. แสดงจำนวนยานพาหนะที่ได้จากการประมวลผลของโปรแกรมในช่องจราจรช่องขวา

	Motorbikes	Cars	Bus,Truck	Total
ช่องจราจรขวา	3	0	0	3

ตารางที่ 8. แสดงจำนวนยานพาหนะจริงที่ผ่านช่องจราจรช่องขวา

	Motorbikes	Cars	Bus,Truck	Total
ช่องจราจรขวา	3	0	0	3

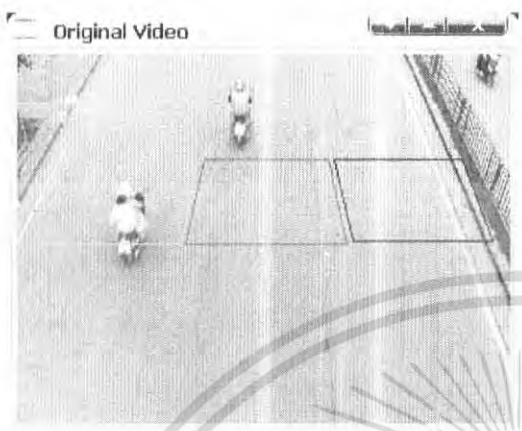
ตารางที่ 9. แสดง %ความผิดพลาดของโปรแกรมในการนับจำนวนยานพาหนะในช่องจราจรขวา

	Motorbikes	Cars	Bus,Truck	Total
ช่องจราจรขวา	0%	0%	0%	0%

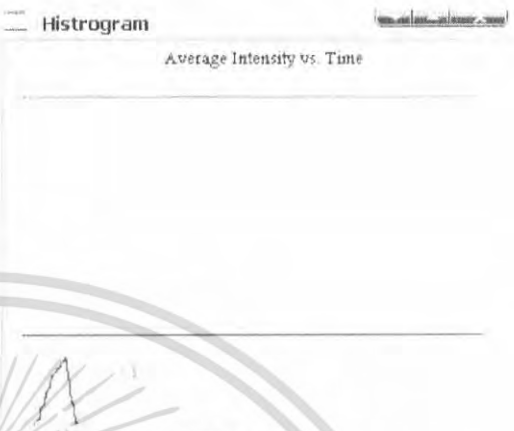
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดลองการนับจำนวนยานพาหนะโดยประมวลผลแบบเรียลไทม์ผ่านกล้องวิดีโอ

4.2.1 ผลการทดลองการนับจำนวนยานพาหนะในช่องจราจรด้านซ้าย



รูปที่ 4.8



รูป 4.9

รูปที่ 4.8 รูปภาพแสดงการเลือก Active Areaในการประมวลผลช่องจราจรช่องซ้าย

รูปที่ 4.9 รูปกราฟแสดงผลในการประมวลผลช่องจราจรช่องซ้าย

ตารางที่ 10. แสดงจำนวนยานพาหนะที่ได้จากการประมวลผลของโปรแกรมในช่องจราจรช่องซ้าย

	Motorbikes	Cars	Bus,Truck	Total
ช่องจราจรซ้าย	4	1	0	5

ตารางที่ 11. แสดงจำนวนยานพาหนะจริงที่ผ่านช่องจราจรช่องซ้าย

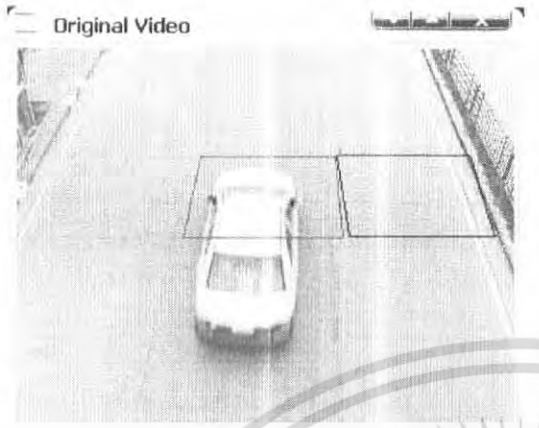
	Motorbikes	Cars	Bus,Truck	Total
ช่องจราจรซ้าย	3	2	0	5

ตารางที่ 12. แสดง %ความผิดพลาดของโปรแกรมในการนับจำนวนยานพาหนะในช่องจราจรซ้าย

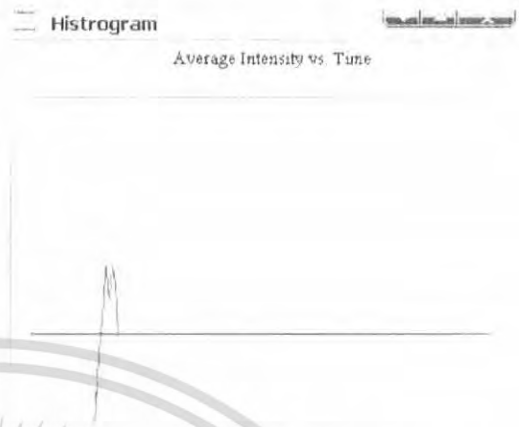
	Motorbikes	Cars	Bus,Truck	Total
ช่องจราจรซ้าย	33.33%	50%	0%	0%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ผลการทดลองการนับจำนวนยานพาหนะในช่องจราจรกลาง



รูปที่ 4.10



รูป 4.11

รูปที่ 4.10 รูปภาพแสดงการเลือก Active Areaในการประมวลผลช่องจราจรช่องกลาง

รูปที่ 4.11 รูปกราฟแสดงผลในการประมวลผลช่องจราจรช่องกลาง

ตารางที่ 13. แสดงจำนวนยานพาหนะที่ได้จากการประมวลผลของโปรแกรมในช่องจราจรช่องกลาง

	Motorbikes	Cars	Bus,Truck	Total
ช่องจราจรกลาง	1	4	1	6

ตารางที่ 14. แสดงจำนวนยานพาหนะจริงที่ผ่านช่องจราจรช่องกลาง

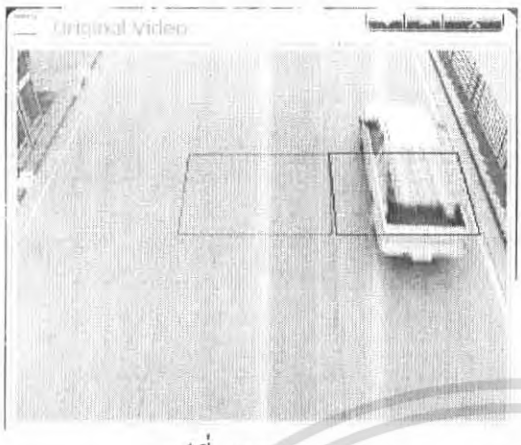
	Motorbikes	Cars	Bus,Truck	Total
ช่องจราจรกลาง	1	4	1	6

ตารางที่ 15. แสดง %ความผิดพลาดของโปรแกรมในการนับจำนวนยานพาหนะในช่องจราจรกลาง

	Motorbikes	Cars	Bus,Truck	Total
ช่องจราจรกลาง	0%	0%	0%	0%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 ผลการทดลองการนับจำนวนยานพาหนะในช่องจราจรขวา



รูปที่ 4.12

รูป 4.13

รูปที่ 4.12 รูปภาพแสดงการเลือก Active Area ในการประมวลผลช่องจราจรช่องขวา

รูปที่ 4.13 รูปกราฟแสดงผลในการประมวลผลช่องจราจรช่องขวา

ตารางที่ 7. แสดงจำนวนยานพาหนะที่ได้จากการประมวลผลของโปรแกรมในช่องจราจรช่องขวา

	Motorbikes	Cars	Bus,Truck	Total
ช่องจราจรขวา	1	4	0	5

ตารางที่ 8. แสดงจำนวนยานพาหนะจริงที่ผ่านช่องจราจรช่องขวา

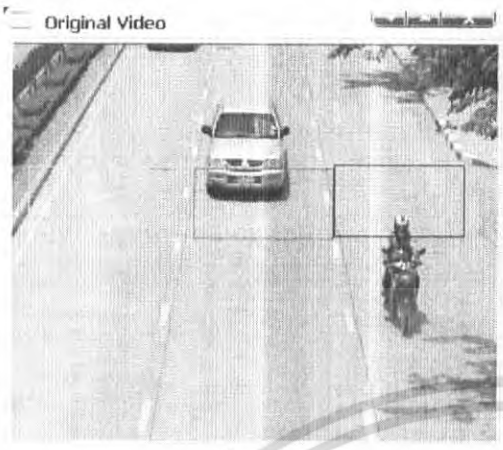
	Motorbikes	Cars	Bus,Truck	Total
ช่องจราจรขวา	1	4	0	5

ตารางที่ 9. แสดง %ความผิดพลาดของโปรแกรมในการนับจำนวนยานพาหนะในช่องจราจรขวา

	Motorbikes	Cars	Bus,Truck	Total
ช่องจราจรขวา	0%	0%	0%	0%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการทดลองเปรียบเทียบการใช้ Active Area แบบกรอบสี่เหลี่ยมและแบบกำหนดรูปร่างได้



รูปที่ 4.14 **รูป 4.15**

รูปที่ 4.14 รูปภาพแสดงActive Area แบบสี่เหลี่ยม

รูปที่ 4.15 รูปภาพแสดงActive Area แบบกำหนดรูปร่างได้

ตารางที่ 19. แสดงจำนวนยานพาหนะที่ได้จากการประมวลและจากการนับจริงด้วยสายตา

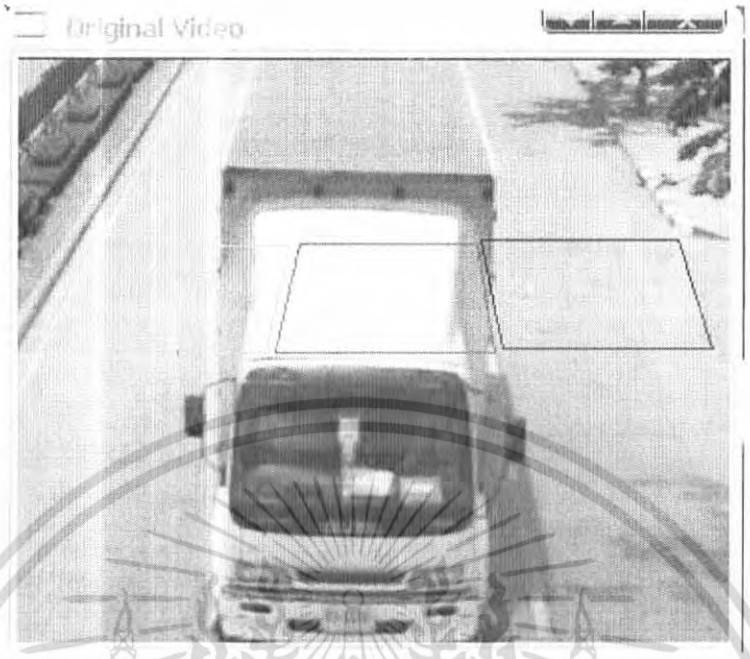
		Motorbikes	Cars	Bus,Trucks	Total
Active Area แบบสี่เหลี่ยม	ช่องซ้าย	1	6	0	7
	ช่องกลาง	1	6	1	8
	ช่องขวา	7	2	0	9
Active Area แบบกำหนดรูปแบบได้ (New)	ช่องซ้าย	0	6	0	6
	ช่องกลาง	1	6	1	8
	ช่องขวา	6	2	0	8
จำนวนยานพาหนะที่ได้จากการนับจริง	ช่องซ้าย	0	6	0	6
	ช่องกลาง	1	6	1	8
	ช่องขวา	6	2	0	8

ตารางที่ 20. แสดง %ความผิดพลาดของโปรแกรมในการนับจำนวนยานพาหนะโดยใช้active areaแบบต่างๆ

% ความผิดพลาด	Active area แบบสี่เหลี่ยม	Active area แบบกำหนดรูปแบบได้
ช่องซ้าย	16.67 %	0 %
ช่องกลาง	0 %	0 %
ช่องขวา	12.5 %	0 %

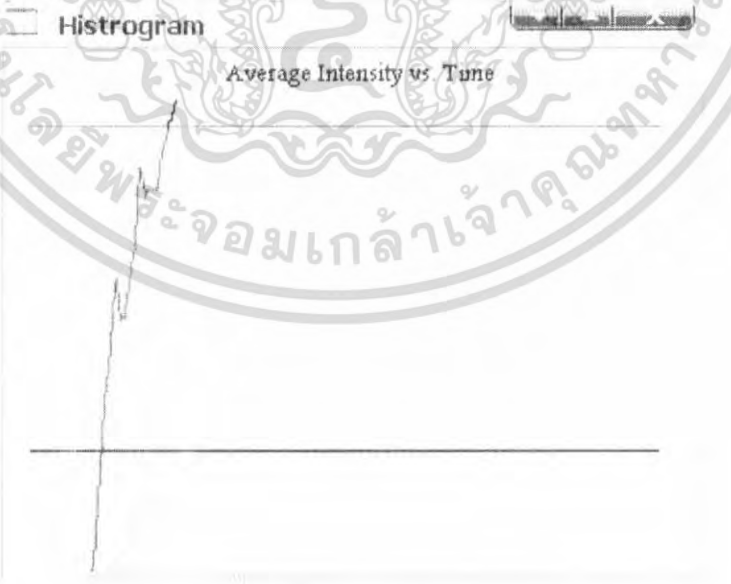
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ผลการทดลองที่เกิดข้อผิดพลาดขึ้น



รูปที่ 4.14 รูปภาพแสดงการเกิดข้อผิดพลาด

จากรูป 4.12 จะเห็นว่ากราฟที่เกิดขึ้นจะมีโอกาสเกิดได้ 2 แบบคือ ค่าจำนวนพาหนะขนาดเล็กจะเพิ่มขึ้น 1 หรือ ค่าจำนวนพาหนะขนาดกลางจะเพิ่มขึ้น 1



รูปที่ 4.15 รูปกราฟที่เกิดข้อผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่5
สรุปและวิจารณ์ผลการทดลอง

5.1 สรุปและวิจารณ์ผลการทดลอง

จากโปรแกรมที่ได้ออกแบบนั้นเป็นการนับจำนวนและแยกประเภทของยานพาหนะ ที่เกิดจากการใช้ทฤษฎีของ การลบกันของภาพ(Image Subtraction) และเทคนิค การทำเทรชโฮล (Thresholding Technique) โดยเมื่อทำการทดลองและพิจารณาผลการทดลองจะเห็นว่าผลการทดลองมีค่าความผิดพลาดน้อยลงจากเดิมอย่างมากเนื่องจากได้มีการปรับปรุงรูปทรงของactive area ให้มีรูปทรงคล้ายกับช่องทางการจราจรมากขึ้น แต่ข้อจำกัดที่มีเพิ่มขึ้นมาจากเดิมคือเวลาในการทดลองจะต้องมีค่าลดลงจากเดิมเนื่องจากการประมวลผลของโปรแกรมมีความซับซ้อนมากขึ้น

5.2 ข้อดีและข้อเสีย

ข้อดี คือสามารถประมวลผลนับจำนวนรถยนต์ ความเร็วรถยนต์ หรือแม้แต่ประเภทของรถยนต์ เพื่อนำไปวิเคราะห์สภาพการจราจรที่เกิดขึ้นในขณะนั้นๆ อีกทั้งยังสามารถตรวจสอบและแก้ปัญหาการขับรถผิดกฎจราจรเช่น ปัญหาการใช้ความเร็วเกินกว่าความเร็วที่กำหนดไว้ได้ ได้ทำการปรับปรุงและแก้ไขความผิดพลาดต่างๆแล้ว

ข้อเสีย คือเนื่องจากโปรแกรมมีขั้นตอนการประมวลผลที่ซับซ้อนมากขึ้น การทำงานจึงต้องใช้เครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูง

5.3 การนำไปประยุกต์ใช้งาน

สามารถนำไปประยุกต์ใช้งานได้จริงตามจุดบนถนนต่างๆ เพื่อง่ายต่อการวิเคราะห์ของตำรวจจราจรว่าควรกดสัญญาณไฟจราจรอย่างไรที่จะทำให้การจราจรไม่เกิดการติดขัด และสามารถนำไปประยุกต์ใช้งานเป็นเครื่องตรวจจับความเร็วรถยนต์ หรือ เครื่องคัดแยกรถเพื่อเข้าสู่ช่องทางที่เหมาะสมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 แนวทางการพัฒนา

-ปรับปรุงวิธีการหรือหาแนวความคิดที่ดีกว่ามาใช้ในการประมวลผลภาพของรถยนต์บนถนน โดยเพาะในเวลากลางคืน และเวลาฝนตก

-ปรับปรุงให้โปรแกรมสามารประมวลผลได้ในคอมพิวเตอร์ที่มีประสิทธิภาพไม่สูงมากนักได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. ยุทธนา ลีลาศวัฒนกุล, "คู่มือการเขียนโปรแกรมวินโดว์ขั้นสูงด้วย Visual C++.NET Episode one", 2546.

2. รศ.ดร. ชูชาติ ปิณฑวิรุจน์,"การประมวลผลภาพเชิงเลข"ภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

3. ยุทธนา ลีลาศวัฒนกุล, "คู่มือการเขียนโปรแกรมและใช้งาน Visual C++.NET ฉบับสมบูรณ์", 2546

4. Robert Laganie're,"Programming computer visionApplication",VIVA lab, University of Ottawa.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

```
// Mainprojectdlg.cpp : implementation file
//

#include "stdafx.h"
#include "Mainproject.h"
#include "MainprojectDlg.h"
#include ".\mainprojectdlg.h"
#include <stdlib.h>
#include <sstream>
#include <string>

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

bool Eexit = false;
bool draw = false;
bool CropRect = false;
bool rect[3] = {false};
bool CreateRect[3] = {false};
bool played = false;
bool processing = false;
bool Headercreated = false;
bool init,init2,init3 = true;
bool end,end2,end3 = false;
bool In1,In2,In3 = false;
bool In_1,In_2,In_3 = false;
bool calxy = false;

int AllRect = 0;
int press;
int allFrame = 0;
int firstFrame = 1;
int motorcycle1,car1,truck1 = 0;
int motorcycle2,car2,truck2 = 0;
int motorcycle3,car3,truck3 = 0;
int Dt1 = 4;
int Dt2 = 4;
int Dt3 = 4;
int detect = 0;
int ddr1,ddl2,dcr2,ddl3 = 0 ;
int framecount;
int FPS = 12;
int beginframe,beginframe2,beginframe3 = 0;
int endframe,endframe2,endframe3 = 0;
int axisx = 0;
int thmotor = 35;
int thcar = 65;
int thtruck = 200;
int minx[250];
int maxx[250];
int minx2[250];
int maxx2[250];
int minx3[250];
int maxx3[250];
int psum[250];
int psum2[250];
int psum3[250];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
float threshold=50;
float max_value=255;
float t1,t2,t3 =0;
float t_1,t_2,t_3 =0;
float dr1,dl2,dr2,dl3 =0;
float dr_1,dl_2,dr_2,dl_3 =0;
float sum = 0;
float dsuml = 0;
float dsumr = 0;
float max_1,max_2,max_3 = 0;
float max1,max2,max3 = 0;
float time1,speed = 0;
float time2,speed2 = 0;
float time3,speed3 = 0;

CvPoint startmouse = {0, 0};
CvPoint endmouse = {0, 0};

CvPoint mm1 = {100,60};
CvPoint mm2 = {100,150};

CvPoint secondpoint = {0, 0};
CvPoint lastpoint = {0, 0};

CvPoint startRect = {0, 0};
CvPoint endRect   {0, 0};
CvPoint startRect_1 = {0, 0};
CvPoint endRect_1 = {0, 0};
CvPoint spRect = {0, 0};
CvPoint lpRect = {0, 0};

CvPoint startRect_2 = {0, 0};
CvPoint endRect_2 = {0, 0};
CvPoint startRect2 = {0, 0};
CvPoint endRect2 = {0, 0};
CvPoint spRect2   {0, 0};
CvPoint lpRect2 = {0, 0};

CvPoint startRect_3 = {0, 0};
CvPoint endRect_3 = {0, 0};
CvPoint startRect3 = {0, 0};
CvPoint endRect3 = {0, 0};
CvPoint spRect3   {0, 0};
CvPoint lpRect3 = {0, 0};

CvCapture *cap = 0;
CvCapture *input_video = 0;


IplImage* ProcessArea = NULL;
IplImage* ProcessArea2 = NULL;
IplImage* ProcessArea3 = NULL;

IplImage* ProcessAreaRef = NULL;
IplImage* ProcessAreaRef2 = NULL;
IplImage* ProcessAreaRef3 = NULL;
IplImage* imcap   0;
IplImage* image   0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IplImage* saveImage1 = NULL;
IplImage* saveImage2 = NULL;
IplImage* saveImage3 = NULL;

IplImage* dst1 = NULL;
IplImage* dst2 = NULL;
IplImage* dst3 = NULL;

IplImage* thresh = NULL;
IplImage* thresh2 = NULL;
IplImage* thresh3 = NULL;

CvSize ProcessAreaSize = {0, 0};
CvSize ProcessAreaSize2 = {0, 0};
CvSize ProcessAreaSize3 = {0, 0};

CvRect SRect;
CvRect SRect2;
CvRect SRect3;


// ********** [illegible] **********
IplImage* graph   cvLoadImage("c:\\backgraph.bmp",1);
const char* filenamegraph1="c:\\backgraph1.bmp";

CvPoint pt1begin,pt2begin,pt3begin = {0,0};
CvPoint pt1stop,pt2stop,pt3stop = {0,0};
CvPoint ptMotorS,ptCarS,ptTruckS = {0,0};
CvPoint ptMotorE,ptCarE,ptTruckE = {0,0};
//**************************************************

// CMainprojectDlg dialog


CMainprojectDlg::CMainprojectDlg(CWnd* pParent /*=NULL*/)
	: CDialog(CMainprojectDlg::IDD, pParent)
	, m_show(_T("Press Open and select file"))
	, m_Total(_T(""))
	, m_Thresh(50)
	, m_ShowTH(0)
	, m_Motor(_T(""))
	, m_Car(_T(""))
	, m_Truck(_T(""))
	, m_Help(_T("Set your values and press 'confirm setting'
button"))
	, m_Lvmotor(35)
	, m_Lvcar(65)
	, m_Lvtruck(200)
	, m_velocity(_T(""))
	, m_iniframe(_T(""))
	, m_Motor2(_T(""))
	, m_Car2(_T(""))
	, m_Truck2(_T(""))
	, m_Total2(_T(""))
	, m_Motor3(_T(""))
	, m_Car3(_T(""))
	, m_Truck3(_T(""))
	, m_Total3(_T(""))
	, m_velocity2(_T(""))
	, m_velocity3(_T(""))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        , m_TTV(_T(""))
        , m_Dt1(4)
        , m_Dt2(4)
        , m_Dt3(4)
        , m_FPS(12)
        , m_dr1(_T(""))
        , m_dl2(_T(""))
        , m_dr2(_T(""))
        , m_dl3(_T(""))
        , m_detect(_T(""))
{
        m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CMainprojectDlg::DoDataExchange(CDataExchange* pDX)
{
        CDialog::DoDataExchange(pDX);
        DDX_Text(pDX, IDC_show, m_show);
        DDX_Text(pDX, IDC_Total, m_Total);
        DDX_Text(pDX, IDC_Thresh, m_Thresh);
        DDX_Text(pDX, IDC_ShowTH, m_ShowTH);
        DDX_Text(pDX, IDC_Motor, m_Motor);
        DDX_Text(pDX, IDC_Car, m_Car);
        DDX_Text(pDX, IDC_Truck, m_Truck);
        DDX_Text(pDX, IDC_Help, m_Help);
        DDX_Text(pDX, IDC_Lvmotor, m_Lvmotor);
        DDX_Text(pDX, IDC_Lvcar, m_Lvcar);
        DDX_Text(pDX, IDC_Lvtruck, m_Lvtruck);
        DDX_Text(pDX, IDC_velocity, m_velocity);
        DDX_Text(pDX, IDC_iniframe, m_iniframe);
        DDX_Text(pDX, IDC_Motor2, m_Motor2);
        DDX_Text(pDX, IDC_Car2, m_Car2);
        DDX_Text(pDX, IDC_Truck2, m_Truck2);
        DDX_Text(pDX, IDC_Total2, m_Total2);
        DDX_Text(pDX, IDC_Motor3, m_Motor3);
        DDX_Text(pDX, IDC_Car3, m_Car3);
        DDX_Text(pDX, IDC_Truck3, m_Truck3);
        DDX_Text(pDX, IDC_Total3, m_Total3);
        DDX_Text(pDX, IDC_velocity2, m_velocity2);
        DDX_Text(pDX, IDC_velocity3, m_velocity3);
        DDX_Text(pDX, IDC_TTV, m_TTV);
        DDX_Text(pDX, IDC_Dt1, m_Dt1);
        DDX_Text(pDX, IDC_Dt2, m_Dt2);
        DDX_Text(pDX, IDC_Dt3, m_Dt3);
        DDX_Text(pDX, IDC_FPS, m_FPS);
        DDX_Text(pDX, IDC_dr1, m_dr1);
        DDX_Text(pDX, IDC_dl2, m_dl2);
        DDX_Text(pDX, IDC_dr2, m_dr2);
        DDX_Text(pDX, IDC_dl3, m_dl3);
        DDX_Text(pDX, IDC_detect, m_detect);
}

BEGIN_MESSAGE_MAP(CMainprojectDlg, CDialog)
        ON_WM_PAINT()
        ON_WM_QUERYDRAGICON()
        //}}AFX_MSG_MAP
        ON_BN_CLICKED(IDOK, OnBnClickedOk)
        ON_BN_CLICKED(IDC_Open, OnBnClickedOpen)
        ON_BN_CLICKED(IDC_Play, OnBnClickedPlay)
        ON_BN_CLICKED(IDC_Stop, OnBnClickedStop)
        ON_BN_CLICKED(IDC_SetTH, OnBnClickedSetth)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	ON_BN_CLICKED(IDC_Camera, OnBnClickedCamera)
	ON_BN_CLICKED(IDC_file, OnBnClickedfile)
	ON_BN_CLICKED(IDC_ccamera, OnBnClickedccamera)
	ON_BN_CLICKED(IDC_Pause, OnBnClickedPause)
	ON_BN_CLICKED(IDC_B, OnBnClickedB)
END_MESSAGE_MAP()


// CMainprojectDlg message handlers

BOOL CMainprojectDlg::OnInitDialog()
{
	CDialog::OnInitDialog();

	// Set the icon for this dialog.  The framework does this
automatically
	//  when the application's main window is not a dialog
	SetIcon(m_hIcon, TRUE);			// Set big icon
	SetIcon(m_hIcon, FALSE);		// Set small icon

	// TODO: Add extra initialization here

	return TRUE;  // return TRUE  unless you set the focus to
a control
}

// If you add a minimize button to your dialog, you will need the
code below
//  to draw the icon.  For MFC applications using the document view
model,
//  this is automatically done for you by the framework.

void CMainprojectDlg::OnPaint()
{
	if (IsIconic())
	{
		CPaintDC dc(this); // device context for painting

		SendMessage(WM_ICONERASEBKGND,
reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

		// Center icon in client rectangle
		int cxIcon = GetSystemMetrics(SM_CXICON);
		int cyIcon = GetSystemMetrics(SM_CYICON);
		CRect rect;
		GetClientRect(&rect);
		int x = (rect.Width() - cxIcon + 1) / 2;
		int y = (rect.Height() - cyIcon + 1) / 2;

		// Draw the icon
		dc.DrawIcon(x, y, m_hIcon);
	}
	else
	{
		CDialog::OnPaint();
	}
}

// The system calls this function to obtain the cursor to display
while the user drags
//  the minimized window.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
HCURSOR CMainprojectDlg::OnQueryDragIcon()
{
	return static_cast<HCURSOR>(m_hIcon);
}


void on_mouse( int event, int x, int y, int flags )
{
    if( !image )
        return;
      if( image->origin )           //Convert Image Coordi nate
        y = image->height - y;

    switch( event )
    {
    case CV_EVENT_LBUTTONDOWN:

                  startmouse.x = x;
                startmouse.y = y;
                  CropRect = false ;
                  draw = false ;
            break;

      case CV_EVENT_LBUTTONUP:

                  secondpoint.x = x;
                  secondpoint.y = startmouse.y;
                  draw = true ;
            break;

      case CV_EVENT_RBUTTONDOWN:

                  lastpoint.x = x;
                lastpoint.y = y;
            break;

      case CV_EVENT_RBUTTONUP:

                  endmouse.x = x;
                  endmouse.y = lastpoint.y;
                  CropRect = true ;
            break;
      }
}
void cropImage(void)
{
	ProcessArea = cvCloneImage(image);
	ProcessArea2 = cvCloneImage(image);
	ProcessArea3 = cvCloneImage(image);

	cvSetImageROI(ProcessArea,SRect);
	cvSetImageROI(ProcessArea2,SRect2);
	cvSetImageROI(ProcessArea3,SRect3);

}

void initImage(void)
{
	ProcessAreaSize.height = abs(startRect.y - endRect.y);
	ProcessAreaSize.width = abs(startRect.x - endRect.x);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ProcessAreaSize2.height = abs(startRect2.y - endRect2.y);
        ProcessAreaSize2.width = abs(startRect2.x - endRect2.x);

        ProcessAreaSize3.height = abs(startRect3.y - endRect3.y);
        ProcessAreaSize3.width = abs(startRect3.x - endRect3.x);

        ProcessArea = cvCreateImageHeader(ProcessAreaSize,
IPL_DEPTH_8U, 3);
        ProcessArea2 = cvCreateImageHeader(ProcessAreaSize2,
IPL_DEPTH_8U, 3);
        ProcessArea3 = cvCreateImageHeader(ProcessAreaSize3,
IPL_DEPTH_8U, 3);

        ProcessAreaRef = cvCreateImageHeader(ProcessAreaSize,
IPL_DEPTH_8U, 3);
        ProcessAreaRef2 = cvCreateImageHeader(ProcessAreaSize2,
IPL_DEPTH_8U, 3);
        ProcessAreaRef3 = cvCreateImageHeader(ProcessAreaSize3,
IPL_DEPTH_8U, 3);

        dst1 = cvCreateImageHeader(ProcessAreaSize, IPL_DEPTH_8U, 3);
        dst2 = cvCreateImageHeader(ProcessAreaSize2, IPL_DEPTH_8U,
3);
        dst3 = cvCreateImageHeader(ProcessAreaSize3, IPL_DEPTH_8U,
3);

        thresh = cvCreateImageHeader(ProcessAreaSize, IPL_DEPTH_8U,
3);
        thresh2 = cvCreateImageHeader(ProcessAreaSize2, IPL_DEPTH_8U,
3);
        thresh3 = cvCreateImageHeader(ProcessAreaSize3, IPL_DEPTH_8U,
3);

        // create imageData
        cvCreateImageData(ProcessArea);
        cvCreateImageData(ProcessArea2);
        cvCreateImageData(ProcessArea3);

        cvCreateImageData(ProcessAreaRef);
        cvCreateImageData(ProcessAreaRef2);
        cvCreateImageData(ProcessAreaRef3);

        cvCreateImageData(dst1);
        cvCreateImageData(dst2);
        cvCreateImageData(dst3);

        cvCreateImageData(thresh);
        cvCreateImageData(thresh2);
        cvCreateImageData(thresh3);

        Headercreated = true;
}

// Access Pixel
void process(void* img)
{

  IplImage* image1 = reinterpret_cast<IplImage*>(img);

  int step= image1->widthStep; // because of a padding
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   sum=0;

   unsigned char *data= reinterpret_cast<unsigned char *>(image1-
>imageData);

   for (int i=lpRect.y ; i<startRect.y ; i++) {
     for (int j=minx[i]; j<maxx[i]; j+= image1->nChannels)
       {
             psum[i] = psum[i] + (int)data[i,j];

     }
      data+= step;  // next line
   }
   for (int z=lpRect.y ; z<startRect.y ; z++){
         sum = sum + psum[z] ;
   }


}

void process2(void* img)
{

   IplImage* image1 = reinterpret_cast<IplImage*>(img);

   int step= image1->widthStep; // because of alignment

   sum = 0;

   unsigned char *data= reinterpret_cast<unsigned char *>(image1-
>imageData);

   for (int i=lpRect2.y ; i<startRect2.y ; i++) {
     for (int j=minx2[i]; j<maxx2[i]; j+= image1->nChannels)
       {
             psum2[i] = psum2[i] + (int)data[i,j];

     }
      data+= step;  // next line
   }
   for (int z=lpRect2.y ; z<startRect2.y ; z++){
         sum = sum + psum2[z] ;
   }

}

void process3(void* img)
{

   IplImage* image1 = reinterpret_cast<IplImage*>(img);

   int step= image1->widthStep; // because of alignment

   sum = 0;
     unsigned char *data= reinterpret_cast<unsigned char *>(image1-
>imageData);

   for (int i=lpRect3.y ; i<startRect3.y ; i++) {
     for (int j=minx3[i]; j<maxx3[i]; j+= image1->nChannels)
       {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                psum3[i] = psum3[i] + (int)data[i,j];

        }
         data+= step;  // next line
    }
    for (int z=lpRect3.y ; z<startRect3.y ; z++){
          sum = sum + psum3[z] ;
    }

}


void coordinate (void)
{
      for (int y= lpRect.y ; y<startRect.y ; y++ ){
            minx[y] = (int) ((abs(y-lpRect.y)*(abs(startRect.x-
lpRect.x)/abs(startRect.y-lpRect.y)))+lpRect.x);
            maxx[y] = (int) ((abs(y-endRect.y)*(abs(spRect.x-
endRect.x)/abs(spRect.y-endRect.y)))+endRect.x);
      }

      for (int y= lpRect2.y ; y<startRect2.y ; y++ ){
minx2[y] = (int) ((abs(y-lpRect2.y)*(abs(startRect2.x-
lpRect2.x)/abs(startRect2.y-lpRect2.y)))+lpRect2.x);

maxx2[y] = (int) ((abs(y-endRect2.y)*(abs(spRect2.x-
endRect2.x)/abs(spRect2.y-endRect2.y)))+endRect2.x);
      }

      for (int y= lpRect3.y ; y<startRect3.y ; y++ ){
minx3[y] = (int) ((abs(y-lpRect3.y)*(abs(startRect3.x-
lpRect3.x)/abs(startRect3.y-lpRect3.y)))+lpRect3.x);

maxx3[y] = (int) ((abs(y-endRect3.y)*(abs(spRect3.x-
endRect3.x)/abs(spRect3.y-endRect3.y)))+endRect3.x);
      }
}


void detectright(void* img)
{

  IplImage* image2 = reinterpret_cast<IplImage*>(img);

  int nl= image2->height;
  int nc= image2->width * image2->nChannels;
  int step= image2->widthStep; // because of alignment

  dsumr=0;

  unsigned char *data= reinterpret_cast<unsigned char *>(image2-
>imageData);

  for (int i=0; i<nl; i++) {
    for (int j= nc - 20 ; j<nc; j+= image2->nChannels)
      {
            dsumr += (int)data[i,j];
    }
     data+= step;  // next line
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}


void detectleft(void* img)
{

  IplImage* image1 = reinterpret_cast<IplImage*>(img);

  int nl= image1->height;
  int nc= image1->width * image1->nChannels;
  int step= image1->widthStep; // because of alignment

  dsum1=0;

  unsigned char *data= reinterpret_cast<unsigned char *>(image1-
>imageData);

  for (int i=0; i<nl; i++) {
    for (int j= 0 ; j<20; j+= image1->nChannels)
      {
            dsum1 += (int)data[i,j];
    }
     data+= step;  // next line
   }

}


void CMainprojectDlg::OnBnClickedOk()
{
      // TODO: Add your control notification handler code here
      cvDestroyAllWindows();
      cvReleaseImage(&ProcessArea);
      cvReleaseImage(&ProcessArea2);
      cvReleaseImage(&ProcessArea3);
      cvReleaseImage(&ProcessAreaRef);
      cvReleaseImage(&ProcessAreaRef2);
      cvReleaseImage(&ProcessAreaRef3);
      Eexit = true;
      cvReleaseCapture(&input_video);
      cvReleaseCapture(&cap);
      OnOK();
}

void CMainprojectDlg::OnBnClickedOpen()
{
      // TODO: Add your control notification handler code here
      allFrame = 1 ;
}

void CMainprojectDlg::OnBnClickedPlay()
{
      // TODO: Add your control notification handler code here
      if(AllRect == 0)
                  {
                        MessageBox("Invalid Input file or forget to
sellect processing area","Cannot Play",MB_ICONINFORMATION);
                        return;
                  }
                  m_Help = "Processing ..... ,please wait for the
result" ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        UpdateData(FALSE);

                        played = true;
                        allFrame = 1;

                        if(!Headercreated)
                              initImage();

}

void CMainprojectDlg::OnBnClickedStop()
{
      // TODO: Add your control notification handler code here
            Exit = true ;

}



void CMainprojectDlg::OnBnClickedSetth()
{
      // TODO: Add your control notification handler code here
      UpdateData(TRUE);
      threshold = m_Thresh ;
      thmotor = m_Lvmotor ;
      thcar = m_Lvcar ;
      thtruck = m_Lvtruck ;
      m_ShowTH = m_Thresh ;
      Dt1 = m_Dt1 ;
      Dt2 = m_Dt2 ;
      Dt3 = m_Dt3 ;
      FPS = m_FPS ;
      m_Help = "Press Open button and select your input file(.avi)"
;
      UpdateData(FALSE);

}



void CMainprojectDlg::OnBnClickedCamera()
{
      // TODO: Add your control notification handler code here
      CvCapture *cap = cvCaptureFromCAM(0);
      cvNamedWindow("Video from Camera", CV_WINDOW_AUTOSIZE);
      for(;;)
      {
      imcap = cvQueryFrame(cap);
      cvShowImage("Video from Camera", imcap);
      press = cvWaitKey(1);
      if(press == '\x0D') // Enter
            {
                  break;
            }
      }

}

void CMainprojectDlg::OnBnClickedfile()
{
      // TODO: Add your control notification handler code here
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CFileDialog dlg(TRUE, _T("*.avi"),
"",OFN_FILEMUSTEXIST|OFN_PATHMUSTEXIST|OFN_HIDEREADONLY,"AVI files
(*.avi) |*.avi|All Files (*.*)|*.*||",NULL);

      char title[]= {"Open File"};
      CString path;
      dlg.m_ofn.lpstrTitle= title;

      if (dlg.DoModal() == IDOK)
      {

      CString path= dlg.GetPathName();  // Contain the selected
filename

      }
      path= dlg.GetPathName();
      m_show = "Your file is   " + path ;
      UpdateData(FALSE);
      CvCapture *input_video = cvCaptureFromFile(path);
      if (input_video == NULL)
            {
            MessageBox("Can not open your selected file","Cannot
open",MB_ICONINFORMATION);
            return;
            }

      //cvQueryFrame( input_video );
      CvSize frame_size;
      frame_size.height = (int) cvGetCaptureProperty( input_video,
CV_CAP_PROP_FRAME_HEIGHT );
      frame_size.width = (int) cvGetCaptureProperty( input_video,
CV_CAP_PROP_FRAME_WIDTH );

      cvNamedWindow("Original Video", CV_WINDOW_AUTOSIZE);
      cvMoveWindow("Original Video",25,40);
      cvNamedWindow("Histrogram", CV_WINDOW_AUTOSIZE);
      cvMoveWindow("Histrogram",600,380);
      cvSetMouseCallback( "Original Video", on_mouse );
      m_Help = "Select Active Area by drag and drop your mouse then
press Enter key" ;
      UpdateData(FALSE);
      for(;;)
    {
            if (Exit)
            {
                cvReleaseCapture(&input_video);
                  break;

            }
            if ( firstFrame ? 1 : allFrame)
            {
                  image = cvQueryFrame(input_video);
                  firstFrame = 0;
                  if(!image)
                        break;
            }
            press = cvWaitKey(1);

            if(draw){
                  if(!played)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        cvLine( image, startmouse, secondpoint,
CV_RGB(255, 255, 255), 1, 8, 0 );
                }

                if(CropRect){
                        if(!played)
                        {
                        cvLine( image, startmouse, secondpoint,
CV_RGB(255, 255, 255), 1, 8, 0 );
                        cvLine( image, startmouse, lastpoint,
CV_RGB(255, 255, 255), 1, 8, 0 );
                        cvLine( image, endmouse, secondpoint,
CV_RGB(255, 255, 255), 1, 8, 0 );
                        cvLine( image, endmouse, lastpoint,
CV_RGB(255, 255, 255), 1, 8, 0 );
                        }
                }
                if(press == '\x0D') // Enter
                {       AllRect = AllRect + 1 ;
                        if(AllRect == 1)
                        {
                                CreateRect[0] = true;
                                rect[0] = true;
                        }
                        else if(AllRect == 2)
                        {
                                CreateRect[1] = true;
                                rect[1] = true;
                        }
                        else if(AllRect == 3)
                        {
                                CreateRect[2] = true;
                                rect[2] = true;
                        }
                        else if(AllRect != 0)
                        {
                                AllRect = 0;
                                CropRect = false;
                                CreateRect[0] = false;
                                CreateRect[1] = false;
                                CreateRect[2] = false;
                                startRect.x = 0;
                                startRect.y = 0;
                                endRect.x = 0;
                                endRect.y = 0;

                                startRect2.x = 0;
                                startRect2.y = 0;
                                endRect2.x = 0;
                                endRect2.y = 0;

                                startRect3.x = 0;
                                startRect3.y = 0;
                                endRect3.x = 0;
                                endRect3.y = 0;
                        }
                }
                if(rect[0])
                {
                        startRect_1.x = startmouse.x;
                        startRect_1.y = startmouse.y;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            startRect.x = startmouse.x
            startRect.y = startmouse.y;

            spRect.x = secondpoint.x;
            spRect.y = secondpoint.y;

            lpRect.x = lastpoint.x;
            lpRect.y = lastpoint.y;

            endRect_1.x = endmouse.x;
            endRect_1.y = endmouse.y;
            endRect.x = endmouse.x;
            endRect.y = endmouse.y;

            rect[0] = false;
        }

        if(rect[1])
        {
            startRect_2.x = startmouse.x;
            startRect_2.y = startmouse.y;
            startRect2.x = startmouse.x
            startRect2.y = startmouse.y;

            spRect2.x = secondpoint.x;
            spRect2.y = secondpoint.y;

            lpRect2.x = lastpoint.x;
            lpRect2.y = lastpoint.y;

            endRect_2.x = endmouse.x;
            endRect_2.y = endmouse.y;
            endRect2.x = endmouse.x
            endRect2.y = endmouse.y;


            rect[1] = false;
        }

        if(rect[2])
        {
            startRect_3.x = startmouse.x;
            startRect_3.y = startmouse.y;
            startRect3.x = startmouse.x
                startRect3.y = startmouse.y;

            spRect3.x = secondpoint.x;
            spRect3.y = secondpoint.y;

            lpRect3.x = lastpoint.x;
            lpRect3.y = lastpoint.y;

            endRect3.x = endmouse.x;
            endRect3.y = endmouse.y;
            endRect_3.x = endmouse.x;
            endRect_3.y = endmouse.y;

            rect[2] = false;
            processing = true;
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(CreateRect[0])
                {
                        cvLine( image, startRect_1, spRect,
CV_RGB(0, 255, 0), 1, 8, 0 );
                        cvLine( image, startRect_1, lpRect,
CV_RGB(0, 255, 0), 1, 8, 0 );
                        cvLine( image, endRect_1, spRect, CV_RGB(0,
255, 0), 1, 8, 0 );
                        cvLine( image, endRect_1, lpRect, CV_RGB(0,
255, 0), 1, 8, 0 );
                }
            if(CreateRect[1])
                {
                        cvLine( image, startRect_2, spRect2,
CV_RGB(255, 0, 0), 1, 8, 0 );
                        cvLine( image, startRect_2, lpRect2,
CV_RGB(255, 0, 0), 1, 8, 0 );
                        cvLine( image, endRect_2, spRect2,
CV_RGB(255, 0, 0), 1, 8, 0 );
                        cvLine( image, endRect_2, lpRect2,
CV_RGB(255, 0, 0), 1, 8, 0 );
                }
                if(CreateRect[2])
                {
                        cvLine( image, startRect_3, spRect3,
CV_RGB(0, 0, 255), 1, 8, 0 );
                        cvLine( image, startRect_3, lpRect3,
CV_RGB(0, 0, 255), 1, 8, 0 );
                        cvLine( image, endRect_3, spRect3,
CV_RGB(0, 0, 255), 1, 8, 0 );
                        cvLine( image, endRect_3, lpRect3,
CV_RGB(0, 0, 255), 1, 8, 0 );
                }

            if(processing)
            {
                SRect.height = abs(startRect.y - endRect.y);
                SRect.width = abs(startRect.x - endRect.x);

                SRect.x = startRect.x;
                SRect.y = startRect.y;

                SRect2.height = abs(startRect2.y - endRect2.y);
                SRect2.width = abs(startRect2.x - endRect2.x);

                SRect2.x = startRect2.x;
                SRect2.y = startRect2.y;

                SRect3.height = abs(startRect3.y - endRect3.y);
                SRect3.width = abs(startRect3.x - endRect3.x);

                SRect3.x = startRect3.x;
                SRect3.y = startRect3.y;

                ProcessAreaRef = cvCloneImage(image);
                cvSetImageROI(ProcessAreaRef,SRect);

                ProcessAreaRef2 = cvCloneImage(image);
                cvSetImageROI(ProcessAreaRef2,SRect2);

                ProcessAreaRef3 = cvCloneImage(image);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    cvSetImageROI(ProcessAreaRef3,SRect3);

                    ProcessAreaRef->origin = 0 ;
                    ProcessAreaRef2->origin = 0 ;
                    ProcessAreaRef3->origin = 0 ;
                    const char* filename1="ReferenceImage.bmp";
                    cvSaveImage(filename1,ProcessAreaRef);
                    saveImage1 = cvLoadImage(filename1,1);

                    const char* filename2="ReferenceImage2.bmp";
                    cvSaveImage(filename2,ProcessAreaRef2);
                    saveImage2 = cvLoadImage(filename2,1);

                    const char* filename3="ReferenceImage3.bmp";
                    cvSaveImage(filename3,ProcessAreaRef3);
                    saveImage3 = cvLoadImage(filename3,1);
                    //**************

                    processing = false;
               }
               cropImage();
//*****************************************************************
               if(played)
               {
                    framecount = framecount + 1;
                    float max_value=255;

                    // Lane 1 *******
                    cvAbsDiff(ProcessArea,saveImage1,dst1);
                    cvThreshold(dst1,thresh, threshold,max_value,
CV_THRESH_BINARY );
                    process(thresh);
                    t_1 = sum;
                    detectright(thresh);
                    dr_1 = dsumr;
                    dr1 = dr_1/(20 * (thresh->height));

                    // Lane 2 *******
                    cvAbsDiff(ProcessArea2,saveImage2,dst2);
                    cvThreshold(dst2,thresh2, threshold,max_value,
CV_THRESH_BINARY );
                    process(thresh2);
                    t_2 = sum;
                    detectleft(thresh2);
                    dl_2 = dsuml;
                    dl2 = dl_2 /(20*thresh2->height);
                    detectright(thresh2);
                    dr_2 = dsumr;
                    dr2 = dr_2 /( 20* (thresh2->height));

                    // Lane 3 *******
                    cvAbsDiff(ProcessArea3,saveImage3,dst3);
                    cvThreshold(dst3,thresh3, threshold,max_value,
CV_THRESH_BINARY );
                    process(thresh3);
                    t_3 = sum;
                    detectleft(thresh3);
                    dl_3 = dsuml;
                    dl3 = dl_3 /(20 * (thresh3->height));

                    if((dr1>30) && (dl2>30)){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                In_1 = true;
        }
        if(((dr1<15) && (dl2<15)) && In_1){
                detect = detect + 1 ;
                In_1 = false ;
                if(t_1>t_2)
                        t_2 = t_2 - dl_2;
                else t_1 = t_1 - dr_1;
        }

        if((dr2>30) && (dl3>30)){
                In_2 = true;
        }
        if(((dr2<15) && (dl3<15)) && In_2){
                detect = detect + 1 ;
                In_2 = false ;
                if(t_2>t_3)
                        t_3 = t_3 - dl_3;
                else t_2 = t_2 - dr_2;
        }


        t1 = t_1/(thresh->width*thresh->height);
        t2 = t_2/(thresh2->width*thresh2->height);
        t3 = t_3/(thresh3->width*thresh3->height);

        if(t1 > thmotor)
        {
                if (t1 > max1)
                        max1 = t1;
                        In1 = true;
        }
        if((t1 < 15) && In1 )
          {
                if (max1 < thcar)
                        motorcycle1=motorcycle1-1;
                if ((max1 >= thcar) && (max1 <thtruck))
                        car1=car1+1;
                if (max1 >= thtruck)
                        truck1=truck1+1;

                In1 = false;
                max1 = 0;
                end = true;

          }

          // check speed

        if((t1>15) && (init))
        {
                beginframe = framecount;
                init = false;

        }
        if(end)
        {
                endframe = framecount;
                time1 = (float)(endframe - beginframe)/FPS;
                speed = Dt1/time1;
                init = true;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                end = false;
                speed = speed * (18/5);
                m_velocity.Format("%.2f",speed);
        }


        m_Total.Format("%d",motorcycle1+car1+truck1);
        m_Motor.Format("%d",motorcycle1);
        m_Car.Format("%d",car1);
        m_Truck.Format("%d",truck1);

        axisx++;
        ptlbegin.x=axisx;
        ptlbegin.y=(255-(int)t1);

        if(t2 > thmotor)
        {
                if (t2 > max2)
                        max2 = t2;
                        In2 = true;
        }
        if((t2 < 15) && In2 )
          {
                if (max2 < thcar)
                        motorcycle2=motorcycle2+1;
                if ((max2 >= thcar) && (max2 <thtruck))
                        car2=car2+1;
                if (max2 >= thtruck)
                        truck2=truck2+1;

                In2 = false;
                max2 = 0;
                end2 = true;

          }
        // Count speed

        if((t2>12) && (init2))
        {
                beginframe2 = framecount;
                init2 = false;

        }
        if(end2)
        {
                endframe2 = framecount;
                time2 = (float)(endframe2 -
beginframe2)/FPS;
                speed2 = Dt2 / time2;
                init2 = true;
                end2 = false;
                speed2 = speed2 * (18/5);
                m_velocity2.Format("%.2f",speed2);
        }


        m_Total2.Format("%d",motorcycle2+car2+truck2);
        m_Motor2.Format("%d",motorcycle2);
        m_Car2.Format("%d",car2);
        m_Truck2.Format("%d",truck2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    //axisx++;
                    pt2begin.x=axisx;
                    pt2begin.y=(255-(int)t2);



                    if(t3 > thmotor)
                    {
                          if (t3 > max3)
                                max3 = t3;
                                In3 = true;
                    }
                    if((t3 < 15) && In3 )
                      {
                          if (max3 < thcar)
                                motorcycle3=motorcycle3+1;
                          if ((max3 >= thcar) && (max3 <thtruck))
                                car3=car3+1;
                          if (max3 >= thtruck)
                                truck3=truck3+1;

                          In3 = false;
                          max3 = 0;
                          end3 = true;

                      }
                      // check speed

                    if((t3>12) && (init3))
                    {
                          beginframe3 = framecount;
                          init3 = false;

                    }
                    if(end3)
                    {
                          endframe3 = framecount;
                          time3 = (float)(endframe3 -
beginframe3)/FPS;
                          speed3 = Dt3/time3;
                          init3 = true;
                          end3 = false;
                          speed3 = speed3 * (18/5);
                          m_velocity3.Format("%.2f",speed3);
                    }


                    m_Total3.Format("%d",motorcycle3+car3-truck3);
                    m_Motor3.Format("%d",motorcycle3);
                    m_Car3.Format("%d",car3);
                    m_Truck3.Format("%d",truck3);

                    //axisx++;
                    pt3begin.x=axisx;
                    pt3begin.y=(255-(int)t3);

                    //////////////////////////////////
                    m_dr1.Format("%d",dr1);
                    m_dl2.Format("%d",dl2);
                    m_dr2.Format("%d",dr2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    m_dl3.Format("%d",dl3);
                    m_detect.Format("%d",detect);

                    UpdateData(FALSE);
                    m_initframe.Format("%d",framecount);

      m_TTV.Format("%d",motorcycle1+car1+truck1+motorcycle2+car2+tr
uck2+motorcycle3+car3+truck3);

                    ptMotorS.x=10;  //axisx;
                    ptMotorS.y=255-thmotor;
                    ptMotorE.x=320;
                    ptMotorE.y=255-thmotor;
                    ptCarS.x=10;
                    ptCarS.y=255-thcar;
                    ptCarE.x=320;
                    ptCarE.y=255-thcar;
                    ptTruckS.x=10;
                    ptTruckS.y=255-thtruck;
                    ptTruckE.x=320;
                    ptTruckE.y=255-thtruck;

                    if(axisx==1)
                          graph = cvLoadImage("c:\\backgraph.bmp",1);
                    else
                          graph = cvLoadImage( filenamegraph1, 1 );

                    cvLine( graph, pt1begin, pt1stop, CV_RGB(0, 255,
0),1, 8, 0 );
                    cvLine( graph, pt2begin, pt2stop, CV_RGB(255, 0,
0),1, 8, 0 );
                    cvLine( graph, pt3begin, pt3stop, CV_RGB(100,
150, 255),1, 8, 0 );

                    cvLine( graph, ptMotorS, ptMotorE,
CV_RGB(255,255,0),1, 8, 0 );
                    cvLine( graph, ptCarS, ptCarE, CV_RGB(0,0,255),1,
8, 0 );
                    cvLine( graph, ptTruckS, ptTruckE,
CV_RGB(255,0,150),1, 8, 0 );

                    pt1stop = pt1begin;
                    pt2stop = pt2begin;
                    pt3stop = pt3begin;

                    cvSaveImage("c:\\backgraph1.bmp",graph);

                    if (axisx > 350)
                    {
                          axisx = 0;
                          graph = cvLoadImage("c:\\backgraph.bmp",1);
                    }
                    dst1->origin = 1 ;
                    dst2->origin = 1 ;
                    dst3->origin = 1 ;
                    thresh->origin = 1 ;
                    thresh->origin = 1 ;
                    thresh2->origin = 1 ;
                    thresh3->origin = 1 ;
              }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            cvShowImage("Original Video", image);
            cvShowImage("Histrogram", graph);
            cvReleaseImage(&ProcessArea);
      }
      cvReleaseCapture(&input_video);
      cvDestroyWindow("Original Video");
}

void CMainprojectDlg::OnBnClickedccamera()
{
      // TODO: Add your control notification handler code here
      CvCapture *input_video = cvCaptureFromCAM(0);
      if (input_video == NULL)
            {
            MessageBox("Can not open your selected file","Cannot
open",MB_ICONINFORMATION);
            return;
            }

      //cvQueryFrame( input_video );
      CvSize frame_size;
      frame_size.height = (int) cvGetCaptureProperty( input_video,
CV_CAP_PROP_FRAME_HEIGHT );
      frame_size.width = (int) cvGetCaptureProperty( input_video,
CV_CAP_PROP_FRAME_WIDTH );

      cvNamedWindow("Original Video", CV_WINDOW_AUTOSIZE);
      cvMoveWindow("Original Video",25,40);
      cvNamedWindow("Histrogram", CV_WINDOW_AUTOSIZE);
      cvMoveWindow("Histrogram",600,380);
      cvSetMouseCallback( "Original Video", on_mouse );
      m_Help = "Select Active Area by drag and drop your mouse then
press Enter key" ;
      UpdateData(FALSE);
      for(;;)
    {
            if (Eexit)
            {
                cvReleaseCapture(&input_video);
                break;

            }
            if ( firstFrame ? 1 : allFrame)
            {
                  image = cvQueryFrame(input_video);
                  firstFrame = 0;
                  if(!image)
                        break;
            }
            press = cvWaitKey(1);
            if(CropRect){
                  if(!played)
                  {
                        cvRectangle( image, startmouse, endmouse,
CV_RGB(255, 255, 255), 1, 8, 0 );
                  }
            }
            if(press == '\xOD') // Enter
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{       AllRect = AllRect + 1 ;
            if(AllRect == 1)
            {
                  CreateRect[0] = true;
                  rect[0] = true;
            }
            else if(AllRect == 2)
            {
                  CreateRect[1] = true;
                  rect[1] = true;
            }
            else if(AllRect == 3)
            {
                  CreateRect[2] = true;
                  rect[2] = true;
            }
            else if(AllRect != 0)
            {
                  AllRect = 0;
                  CropRect = false;
                  CreateRect[0] = false;
                  CreateRect[1] = false;
                  CreateRect[2] = false;
                  startRect.x = 0;
                  startRect.y = 0;
                  endRect.x = 0;
                  endRect.y = 0;

                  startRect2.x = 0;
                  startRect2.y = 0;
                  endRect2.x = 0;
                  endRect2.y = 0;

                  startRect3.x = 0;
                  startRect3.y = 0;
                  endRect3.x = 0;
                  endRect3.y = 0;
            }
}
if(rect[0])
{
      startRect.x = startmouse.x;
      startRect.y = startmouse.y;

      endRect.x = endmouse.x;
      endRect.y = endmouse.y;

      rect[0] = false;
}

if(rect[1])
{
      startRect2.x = startmouse.x;
      startRect2.y = startmouse.y;

      endRect2.x = endmouse.x;
      endRect2.y = endmouse.y;

      rect[1] = false;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(rect[2])
            {
                  startRect3.x = startmouse.x;
                  startRect3.y = startmouse.y;

                  endRect3.x = endmouse.x;
                  endRect3.y = endmouse.y;

                  rect[2] = false;
                  processing = true;
            }

            if(CreateRect[0])
                  cvRectangle( image, startRect, endRect, CV_RGB(0,
255, 0), 1, 8, 0 );
            if(CreateRect[1])
                  cvRectangle( image, startRect2, endRect2,
CV_RGB(255, 0 , 0), 1, 8, 0 );
            if(CreateRect[2])
                  cvRectangle( image, startRect3, endRect3,
CV_RGB(0, 0, 255), 1, 8, 0 );

            if(processing)
            {
                  SRect.height = abs(startRect.y - endRect.y);
                  SRect.width = abs(startRect.x - endRect.x);

                  SRect.x = startRect.x;
                  SRect.y = startRect.y;

                  SRect2.height = abs(startRect2.y - endRect2.y);
                  SRect2.width = abs(startRect2.x - endRect2.x);

                  SRect2.x = startRect2.x;
                  SRect2.y = startRect2.y;

                  SRect3.height = abs(startRect3.y - endRect3.y);
                  SRect3.width = abs(startRect3.x - endRect3.x);

                  SRect3.x = startRect3.x;
                  SRect3.y = startRect3.y;

                  ProcessAreaRef = cvCloneImage(image);
                  cvSetImageROI(ProcessAreaRef,SRect);

                  ProcessAreaRef2 = cvCloneImage(image);
                  cvSetImageROI(ProcessAreaRef2,SRect2);

                  ProcessAreaRef3 = cvCloneImage(image);
                  cvSetImageROI(ProcessAreaRef3,SRect3);

                  ProcessAreaRef->origin = 0 ;
                  ProcessAreaRef2->origin = 0 ;
                  ProcessAreaRef3->origin = 0 ;
                  const char* filename1="ReferenceImage.bmp";
                  cvSaveImage(filename1,ProcessAreaRef);
                  saveImage1 = cvLoadImage(filename1,1);

                  const char* filename2="ReferenceImage2.bmp";
                  cvSaveImage(filename2,ProcessAreaRef2);
                  saveImage2 = cvLoadImage(filename2,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                const char* filename3="ReferenceImage3.bmp";
                cvSaveImage(filename3,ProcessAreaRef3);
                saveImage3 = cvLoadImage(filename3,1);
                //-----------------

                processing = false;
        }
        cropImage();
        //-----------------------------------------------------------------
        if(played)
        {
                framecount = framecount + 1;
                float max_value=255;

                // Part 1 -------
                cvAbsDiff(ProcessArea,saveImage1,dst1);
                cvThreshold(dst1,thresh, threshold,max_value,
CV_THRESH_BINARY );
                process(thresh);
                t1 = sum/(thresh->width*thresh->height);


                if(t1 > thmotor)
                {
                    if (t1 > max1)
                        max1 = t1;
                        in1 = true;
                }
                if((t1 < 15) && in1 )
                    {
                    if (max1 < thcar)
                        motorcycle1=motorcycle1-1;
                    if ((max1 >= thcar) && (max1 <thtruck))
                        car1=car1+1;
                    if (max1 >= thtruck)
                        truck1=truck1+1;

                    in1 = false;
                    max1 = 0;
                    end = true;

                    }
                // the t speed

                if((t1>15) && (init))
                {
                    beginframe = framecount;
                    init = false;

                }
                if(end)
                {
                    endframe = framecount;
                    time1 = (float)(endframe - beginframe)/FPS;
                    speed = Dt1/time1;
                    init = true;
                    end = false;
                    speed = 3*(speed * 18/5);
                    m_velocity.Format("%.2f",speed);
                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    m_Total.Format("%d",motorcycle1+car1+truck1);
                    m_Motor.Format("%d",motorcycle1);
                    m_Car.Format("%d",car1);
                    m_Truck.Format("%d",truck1);

                    axisx+1;
                    pt1begin.x=axisx;
                    pt1begin.y=(255-(int)t1);
                    //Update data(FALSE);
                    //*****************************

                    // Unit 2 *******
                    cvAbsDiff(ProcessArea2,saveImage2,dst2);
                    cvThreshold(dst2,thresh2, threshold,max_value,
CV_THRESH_BINARY );
                    process(thresh2);
                    t2 = sum/(thresh2->width*thresh2->height);


                    if(t2 > thmotor)
                    {
                          if (t2 > max2)
                              max2 = t2;
                              In2 = true;
                    }
                    if((t2 < 15) && In2 )
                      {
                          if (max2 < thcar)
                              motorcycle2=motorcycle2+1;
                          if ((max2 >= thcar) && (max2 <thtruck))
                              car2=car2+1;
                          if (max2 >= thtruck)
                              truck2=truck2+1;

                          In2 = false;
                          max2 = 0;
                          end2 = true;

                      }
                    // check speed

                    if((t2>12) && (init2))
                    {
                          beginframe2 = framecount;
                          init2 = false;

                    }
                    if(end2)
                    {
                          endframe2 = framecount;
                          time2 = (float)(endframe2 -
beginframe2)/FPS;
                          speed2 = Dt2 / time2;
                          init2 = true;
                          end2 = false;
                          speed2 =3*(speed2 * 18/5);
                          m_velocity2.Format("%.2f",speed2);
                    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    m_Total2.Format("%d",motorcycle2+car2+truck2);
                    m_Motor2.Format("%d",motorcycle2);
                    m_Car2.Format("%d",car2);
                    m_Truck2.Format("%d",truck2);

                    //axisx++;
                    pt2begin.x=axisx;
                    pt2begin.y=(255-(int)t2);
                    //UpdateData(TRUE);
                    //*************************************

                    // Image 3 ********
                    cvAbsDiff(ProcessArea3,saveImage3,dst3);
                    cvThreshold(dst3,thresh3, threshold,max_value,
CV_THRESH_BINARY );
                    process(thresh3);
                    t3 = sum/(thresh3->width*thresh3->height);


                    if(t3 > thmotor)
                    {
                          if (t3 > max3)
                                max3 = t3;
                                In3 = true;
                    }
                    if((t3 < 15) && In3 )
                          {
                          if (max3 < thcar)
                                motorcycle3=motorcycle3+1;
                          if ((max3 >= thcar) && (max3 <thtruck))
                                car3=car3+1;
                          if (max3 >= thtruck)
                                truck3=truck3+1;

                          In3 = false;
                          max3 = 0;
                          end3 = true;

                      }
                    // Car speed

                    if((t3>12) && (init3))
                    {
                          beginframe3 = framecount;
                          init3 = false;

                    }
                    if(end3)
                    {
                          endframe3 = framecount;
                          time3 = (float)(endframe3 -
beginframe3)/FPS;
                          speed3 = Dt3/time3;
                          init3 = true;
                          end3 = false;
                          speed3 =3*(speed3 * 18/5);
                          m_velocity3.Format("%.2f",speed3);
                    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    m_Total3.Format("%d",motorcycle3+car3+truck3);
                    m_Motor3.Format("%d",motorcycle3);
                    m_Car3.Format("%d",car3);
                    m_Truck3.Format("%d",truck3);

                    //axisx++;
                    pt3begin.x=axisx;
                    pt3begin.y=(255-(int)t3);

                    //----------------------

                    UpdateData(FALSE);
                    m_iniframe.Format("%d",framecount);

      m_TTV.Format("%d",motorcycle1+car1+truck1+motorcycle2+car2+tr
uck2+motorcycle3+car3+truck3);

                    ptMotorS.x=10;  //axisx;
                    ptMotorS.y=255-thmotor;
                    ptMotorE.x=320;
                    ptMotorE.y=255-thmotor;
                    ptCarS.x=10;
                    ptCarS.y=255-thcar;
                    ptCarE.x=320;
                    ptCarE.y=255-thcar;
                    ptTruckS.x=10;
                    ptTruckS.y=255-thtruck;
                    ptTruckE.x=320;
                    ptTruckE.y=255-thtruck;

                    if(axisx==1)
                          graph = cvLoadImage("c:\\backgraph.bmp",1);
                    else
                          graph = cvLoadImage( filenamegraph1, 1 );

                    cvLine( graph, pt1begin, pt1stop, CV_RGB(0, 255,
0),1, 8, 0 );
                    cvLine( graph, pt2begin, pt2stop, CV_RGB(255, 0,
0),1, 8, 0 );
                    cvLine( graph, pt3begin, pt3stop, CV_RGB(100,
150, 255),1, 8, 0 );

                    cvLine( graph, ptMotorS, ptMotorE,
CV_RGB(255,255,0),1, 8, 0 );
                    cvLine( graph, ptCarS, ptCarE, CV_RGB(0,0,255),1,
8, 0 );
                    cvLine( graph, ptTruckS, ptTruckE,
CV_RGB(255,0,150),1, 8, 0 );

                    pt1stop = pt1begin;
                    pt2stop = pt2begin;
                    pt3stop = pt3begin;

                    cvSaveImage("c:\\backgraph1.bmp",graph);

                    if (axisx > 350)
                    {
                          axisx = 0;
                          graph = cvLoadImage("c:\\backgraph.bmp",1);
                    }
                    dst1->origin = 1 ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                dst2->origin = 1 ;
                dst3->origin = 1 ;
                thresh->origin = 1 ;
                thresh->origin = 1 ;
                thresh2->origin = 1 ;
                thresh3->origin = 1 ;
            }


            cvShowImage("Original Video", image);
            cvShowImage("Histrogram", graph);
            cvReleaseImage(&ProcessArea);
        }
        cvReleaseCapture(&input_video);
        cvDestroyWindow("Original Video");

}

void CMainprojectDlg::OnBnClickedPause()
{
        // TODO: Add your control notification handler code here

        if    (allFrame = 1)
              allFrame = 0;

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้