

ระบบควบคุมรถอัจฉริยะไร้คนขับโดยการประมวลผลภาพ

Smart vehicle control system via computer vision

นายณรงค์ รามสุข รหัส 49364455

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ..... 19 ส.ค. 2555
เลขหนังสือ..... 1575528/
เลขเรียกหนังสือ..... นร.
..... 213 8

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2552

หัวข้อโครงการ ระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการประมวลผลภาพ
ผู้ดำเนินโครงการ นายณรงค์ รวมสุข รหัส 49364455
อาจารย์ที่ปรึกษา อาจารย์เศรษฐา ตั้งคำวานิช
สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา 2552

บทคัดย่อ

โครงการนี้เป็นการพัฒนาระบบควบคุมรถอัจฉริยะไร้คนขับโดยการประมวลผลภาพ โดยทั่วไปรถยนต์จะต้องอาศัยมนุษย์ในการควบคุม ทั้งการขับเคลื่อนรถ และการตัดสินใจเมื่อเจอกับเหตุการณ์ต่าง ๆ อย่างไรก็ตาม ในปัจจุบันความต้องการในการเพิ่มศักยภาพของระบบยานยนต์ขนส่งอัจฉริยะมีมากขึ้น และหนึ่งในการพัฒนาระบบดังกล่าว คือ การพัฒนาระบบควบคุมรถอัจฉริยะไร้คนขับ โดยระบบจะทำการหาตำแหน่งรถ และการตัดสินใจควบคุมรถ โดยอาศัยสัญญาณภาพจะมีความสำคัญมาก โดยเฉพาะการตัดสินใจระยะใกล้

ผลที่ได้จากโครงการนี้ คือ การนำภาพที่ได้จากกล้องมาหาขอบของถนน แล้วทำการลากเส้นตัดขอบของถนน และทำการวิเคราะห์ว่าส่วนไหนของภาพคือถนน แล้วทำการหาจุดกึ่งกลางของถนน และสั่งให้รถขับเคลื่อนไปตามจุดกึ่งกลางของถนน และถ้ามีสิ่งกีดขวางสามารถสั่งให้รถขับเคลื่อนหลบหลีกได้

Project Title Smart vehicle control system via computer vision

Name Mr. Narong Roumsuk ID. 49364455

Project Advisor Mr. Settha Thangkawanit

Major Computer Engineering

Department Electrical and Computer Engineering

Academic Year 2009

.....

ABSTRACT

This project is the development of smart vehicle control system via computer vision. Generally, vehicle requires manpower to control, drive, and decide when facing with various events. However, the requirement to increase potential of intelligent transportation systems is rising nowadays, and one of these developing systems is smart vehicle control system via computer vision. For this system, it will find vehicle's direction and decide to control vehicle by using image signal, which is very important for short range decision.

The result of this project is bringing images from the VDO camera to find the edge of the road, and then trace cut it. After that, this system will analyze which the road image is, and find the middle point of the road. Then, the vehicle will be ordered to drive along the middle of the road. If there is roadblock, this system can order the vehicle to get around it.

กิตติกรรมประกาศ

ปริญญาโทนี้สำเร็จได้ด้วยดีเนื่องจาก อาจารย์เสรษฐา ตั้งคำวานิช ที่ได้ให้คำปรึกษาในการดำเนิน โครงการ และให้คำแนะนำในส่วนวิธีการต่างๆ ที่นำมาประยุกต์ใช้ใน โครงการ รวมทั้งช่วยตรวจสอบข้อผิดพลาดของโครงการ จึงขอขอบพระคุณมา ณ โอกาสนี้

ขอขอบพระคุณ คร.มูชิตา สงฆ์จันทร์ และ คร.อักรพันธ์ วงศ์กั้งแห ที่ได้คำแนะนำและคำปรึกษาในการปรับปรุงโครงการให้ดียิ่งขึ้น จนทำโครงการนี้สำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายขอขอบคุณเพื่อนๆ ทุกคนที่ให้คำแนะนำและคำปรึกษา รวมถึงให้กำลังใจ ผู้จัดทำโครงการจึงขอขอบพระคุณทุกท่านไว้ ณ โอกาสนี้



ผู้จัดทำ
ณรงค์ รวมสุข

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญรูป.....	ช

บทที่ 1 บทนำ

1.1 ที่มา และความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ขอบข่ายงาน.....	2
1.4 ขั้นตอนการดำเนินการ.....	3
1.5 แผนการดำเนินงาน.....	4
1.6 ผลที่คาดว่าจะได้รับ.....	5
1.7 งบประมาณที่ใช้.....	5

บทที่ 2 ทฤษฎีเบื้องต้น และหลักการควบคุม

2.1 การแยกคุณลักษณะ (Feature extraction).....	6
2.2 ระบบสี RGB.....	7
2.3 ระบบสี HSV.....	9
2.4 การหาขอบภาพ (Edge Detection).....	10
2.5 Canny Edge Detection Algorithm.....	12
2.6 การกรองภาพ (Image Filtering).....	13
2.7 Hough Transform.....	15
2.7 OpenCV (Open Source Computer Vision Library).....	17

สารบัญ(ต่อ)

หน้า

บทที่ 3 การวิเคราะห์ ออกแบบ และพัฒนาระบบ

3.1 วิเคราะห์ระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการประมวลผลภาพ.....	19
3.2 การออกแบบระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการประมวลผลภาพ.....	19
3.2.1 การออกแบบระบบภายนอก.....	20
3.2.2 การออกแบบระบบภายใน.....	21

บทที่ 4 ผลการทดลอง

4.1 ผลการทดสอบการติดต่อกล้อ.....	29
4.2 ผลการประมวลภาพถนน โดยไม่มีสิ่งกีดขวาง.....	30
4.2.1 ผลของการทำ Image segmentation.....	30
4.2.2 ผลของการทำ Canny Edge Detection.....	33
4.2.3 ผลของการทำ Hough Transform.....	36
4.2.4 ผลของการเปรียบเทียบเส้นเสมือน(Visual line).....	39
4.3 ผลการประมวลภาพถนน โดยมีสิ่งกีดขวาง.....	42

บทที่ 5 สรุปผล

5.1 สรุปผลการทดลอง.....	44
5.2 ปัญหาและแนวทางแก้ไข.....	44
5.3 แนวทางการพัฒนาต่อในอนาคต.....	45

เอกสารอ้างอิง..... 46

ภาคผนวก ก. การใช้งาน OpenCV

ก.1 ขั้นตอนการติดตั้ง OpenCV.....	48
ก.2 การติดตั้ง OpenCV เสร็จแล้ว ต่อไปเป็นการ built.....	52

สารบัญ(ต่อ)

	หน้า
ภาคผนวก ข. การใช้งาน OpenCV	
ข.1 ผลของการทำ Image segmentation	55
ข.2 ผลของการทำ Canny Edge Detection	60
ข.3 ผลของการทำ Hough Transform.....	62
ข.4 ผลของการทำเส้นเสมือน.....	65
ข.5 ผลของการประมวลผลภาพโดยไม่มีสิ่งกีดขวาง.....	70
ประวัติผู้เขียน โครงการ	78



สารบัญรูป

รูปที่	หน้า
2.1 โครงสร้างระบบสี RGB.....	8
2.2 ระบบสี RGB บน Cartesian coordinate system	8
2.3 แสดงระบบสี HSV.....	9
2.4 การหาขอบภาพโดยวิธี Gradient.....	11
2.5 การหาขอบภาพโดยวิธี Laplacian.....	11
2.6 ขั้นตอนวิธีการหาขอบโดยวิธี Canny.....	12
2.8 ตัวอย่างของ Gaussian Filter พารามิเตอร์หนึ่ง.....	14
2.9 ตัวอย่างของ Low Pass Filter	15
2.10 ตัวอย่างของ Hough Transform.....	16
2.11 ตัวอย่างของ Hough Transform.....	16
3.1 แสดงระบบโดยรวมของรถอัจฉริยะไร้คนขับ.....	18
3.2 แสดงการออกแบบระบบภายนอก.....	20
3.3 แสดงการออกแบบระบบภายใน.....	21
3.4 แสดงการทำ Capture ภาพจากกล้องวิดีโอ.....	22
3.5 แสดงการทำ Convert ระบบสี RGB to HSV.....	22
3.6 แสดงการทำ Get page Hue of HSV.....	23
3.7 แสดงการทำ Threshold.....	23
3.8 แสดงการทำ Edge detection แบบ Flowchart.....	24
3.9 แสดงการทำ Edge detection.....	24
3.10 แสดงการทำ Hough transform.....	25
3.11 แสดงการแยกวัตถุ.....	25
3.12 แสดงการถอดความหมายจากภาพ.....	26
3.13 แสดงการสั่งการทำงาน.....	28
4.1 แสดงผลการติดต่อกับ.....	29

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.2 ก,ข,ค และ ง เป็นภาพแสดงการทำ Image segmentation ในกรณีที่ดินนอยู่ทางขวา.....	30
4.3 ก,ข,ค,ง และ จ เป็นภาพแสดงการทำ Image segmentation ในกรณีที่ดินนอยู่ตรงกลาง.....	31
4.4 ก,ข,ค,ง และ จ เป็นภาพแสดงการทำ Image segmentation ในกรณีที่ดินนอยู่ทางซ้าย.....	32
4.5 ก,ข,ค และ ง เป็นภาพแสดงการทำ Canny edge detection ในกรณีที่ดินนอยู่ทางขวา.....	33
4.6 ก,ข,ค และ ง เป็นภาพแสดงการทำ Canny edge detection ในกรณีที่ดินนอยู่ตรงกลาง.....	34
4.7 ก,ข,ค และ ง เป็นภาพแสดงการทำ Canny edge detection ในกรณีที่ดินนอยู่ทางซ้าย.....	35
4.8 ก,ข และ ค เป็นภาพแสดงการทำ Hough transform ในกรณีที่ดินนอยู่ทางขวา.....	36
4.9 ก,ข,ค และ ง เป็นภาพแสดงการทำ Hough transform ในกรณีที่ดินนอยู่ตรงกลาง.....	37
4.10 ก,ข,ค และ ง เป็นภาพแสดงการทำ Hough transform ในกรณีที่ดินนอยู่ทางซ้าย.....	38
4.11 ก,ข และ ค เป็นภาพแสดงการทำเส้นเสมือน ในกรณีที่ดินนอยู่ทางขวา	39
4.12 ก,ข และ ค เป็นภาพแสดงการทำเส้นเสมือน ในกรณีที่ดินนอยู่ตรงกลาง.....	40
4.13 ก,ข และ ค เป็นภาพแสดงการทำเส้นเสมือน ในกรณีที่ดินนอยู่ทางซ้าย.....	41
4.14 ก และ ข เป็นภาพแสดงการประมวลผลภาพในกรณีที่มีสิ่งกีดขวางอยู่ทางซ้าย.....	42
4.15 ก และ ข เป็นภาพแสดงการประมวลผลภาพในกรณีที่มีสิ่งกีดขวางอยู่ทางขวา.....	43
ก.1 แสดงตัวติดตั้ง OpenCV.....	48
ก.2 แสดงหน้าต่างเริ่มต้นการติดตั้ง OpenCV.....	48
ก.3 แสดงหน้าต่าง License Agreement.....	49
ก.4 แสดงหน้าต่างการ Select Destination Location.....	49
ก.5 แสดงหน้าต่างการ Select Start Menu Folder.....	50
ก.6 แสดงหน้าต่างการ Select Additional Tasks.....	50
ก.7 แสดงหน้าต่างการ Ready to Install.....	51
ก.8 แสดงหน้าต่างการ Installing.....	51
ก.9 แสดงหน้าต่างการ Install Complete.....	52
ก.10 แสดงเปิด OpenCV Workspace.NET 2005.....	52
ก.11 แสดงการทำ Build Solution.....	53

สารบัญรูป(ต่อ)

รูปที่	หน้า
ก.12 แสดงหน้าต่างโปรแกรมคำสั่งทำการ Build.....	53
ก.13 แสดงหน้าต่างการ Set Path ต่างๆ.....	54
ข.1-24 แสดงผลของการทำ Image segmentation ในกรณีต่าง ๆ เพิ่มเติม.....	55
ข.25-36 แสดงผลของการทำ Canny Edge Detection ในกรณีต่าง ๆ เพิ่มเติม.....	60
ข.37-48 แสดงผลของการทำ Hough Transform ในกรณีต่าง ๆ เพิ่มเติม.....	62
ข.49-71 แสดงผลของการทำเส้นเสมือน ในกรณีต่าง ๆ เพิ่มเติม.....	65
ข.72-92 แสดงผลของการประมวลผลภาพโดยไม่มีสิ่งกีดขวาง เพิ่มเติม.....	70



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

โดยทั่วไป รถยนต์จะต้องอาศัยมนุษย์ในการควบคุมรถ ทั้งการขับเคลื่อนรถ และการตัดสินใจเมื่อเจอกับเหตุการณ์ต่าง ๆ อย่างไรก็ตาม ในปัจจุบันความต้องการในการเพิ่มศักยภาพของระบบยานยนต์ขนส่งอัจฉริยะมีมากขึ้น และหนึ่งในการพัฒนาระบบดังกล่าว คือ การพัฒนาระบบควบคุมรถอัจฉริยะไร้คนขับ ซึ่งจะก่อให้เกิดประโยชน์ต่าง ๆ มากมาย อาทิเช่น การช่วยเหลือคนพิการที่ไม่สามารถควบคุมรถได้, การนำไปใช้งานกับสถานที่ที่มนุษย์ไม่สามารถทำงานได้ และการนำไปใช้กับพวกที่เมาแล้วขับเพื่อลดอุบัติเหตุ เป็นต้น ดังนั้นในปัจจุบันจึงได้มีการพัฒนาระบบควบคุมรถอัจฉริยะไร้คนขับขึ้น ซึ่งโดยทั่วไปแล้วระบบควบคุมรถอัจฉริยะไร้คนขับ อาจแบ่งออกได้เป็นสองส่วนหลัก คือ ส่วนควบคุมระดับบน และส่วนควบคุมระดับล่าง โดยส่วนควบคุมระดับบนจะทำหน้าที่ในการตัดสินใจภาพรวมของการขับเคลื่อนของรถ ซึ่งอาศัยข้อมูลภาพจากกล้องวีดีโอบริเวณหน้ารถ เพื่อใช้ในการคำนวณตำแหน่งของถนน และอุปสรรคสิ่งกีดขวาง และข้อมูลจากระบบบอกตำแหน่งพิกัดโลก หรือ GPS เพื่อระบุเส้นทางที่ต้องการให้รถเดินทางไปสู่จุดหมาย เป็นต้น ในส่วนของส่วนควบคุมระดับล่างจะทำหน้าที่บังคับรถกลไกต่าง ๆ ของรถ อาทิเช่น พวงมาลัยรถ ความเร็วรถ เบรก ฯลฯ โดยให้ทำงานตามคำสั่งที่มาจากส่วนควบคุมระดับบน

การพัฒนาระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการประมวลผลภาพ จะทำการหาตำแหน่งรถ และการตัดสินใจควบคุมรถโดยอาศัยสัญญาณภาพ จะมีความสำคัญมาก โดยเฉพาะการตัดสินใจระยะใกล้ ซึ่งหากพิจารณาถึงวัตถุที่เกี่ยวข้องในสนามแข่ง จะพบว่าวัตถุที่สนใจอยู่ทั้งหมด 2 ประเภท คือ ถนน และสิ่งกีดขวาง จากข้อมูลเหล่านี้แสดงให้เห็นว่า วัตถุที่จำเป็นในระบบการมองเห็นทั้ง 2 ชนิดนี้ จะต้องถูกคัดกรองออกมาจากภาพที่รับได้ และทำการแยกแยะประเภทวัตถุดังกล่าวก่อน จากนั้นจึงทำการหาความหมาย และการตัดสินใจเมื่อพบวัตถุนั้น ๆ ต่อไป

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อศึกษาทฤษฎีการประมวลผลภาพ
- 1.2.2 เพื่อศึกษาการถอดความหมายของภาพ
- 1.2.3 เพื่อพัฒนาระบบควบคุมรถจักรยะไร้คนขับ โดยการประมวลผลภาพ
- 1.2.4 เพื่อพัฒนาระบบการตัดสินใจ โดยการประมวลผลภาพ
- 1.2.5 เพื่อพัฒนาระบบการแยกแยะสิ่งกีดขวาง โดยการประมวลผลภาพ
- 1.2.6 เพื่อได้โปรแกรมระบบควบคุมรถจักรยะไร้คนขับ โดยการประมวลผลภาพ
- 1.2.7 เพื่อทดสอบโปรแกรมระบบควบคุมรถจักรยะไร้คนขับ โดยการประมวลผลภาพ
- 1.2.8 เพื่อทำการเก็บข้อมูล นำมาวิเคราะห์ และปรับปรุงแก้ไขระบบควบคุมรถจักรยะไร้

คนขับ โดยการประมวลผลภาพ

1.3 ขอบข่ายงาน

- 1.3.1 รวบรวมข้อมูลเกี่ยวกับทฤษฎีการประมวลผลภาพ
- 1.3.2 รวบรวมข้อมูลของภาพซึ่งเป็นภาพของประเภทของวัตถุที่สนใจ
- 1.3.3 พัฒนาระบบให้สามารถแยกแยะพื้นถนนได้
- 1.3.4 พัฒนาระบบให้สามารถแยกแยะสิ่งกีดขวางได้
- 1.3.5 พัฒนาระบบให้สามารถควบคุมรถเพื่อหลบสิ่งกีดขวางได้
- 1.3.6 พัฒนาระบบให้สามารถควบคุมรถให้วิ่งบนถนนทางเรียบได้
- 1.3.7 พัฒนาระบบโดยใช้คอมพิวเตอร์ส่วนบุคคล 1 เครื่องในการควบคุม
- 1.3.8 พัฒนาระบบโดยใช้กล้องวีดีโอ 1 ตัว ในการรับภาพ
- 1.3.9 ระบบไม่สามารถทำงานได้ในสภาวะที่ฝนตก
- 1.3.10 ระบบไม่สามารถทำงานได้ในสภาวะที่มีแสงน้อย

1.4 ขั้นตอนการดำเนินการ

1.4.1 ศึกษา ไลบรารี ที่เกี่ยวข้องกับ OpenCV เพื่อใช้ในการเขียน โปรแกรมซึ่งกระทำเกี่ยวกับภาพ

1.4.2 เก็บข้อมูลรูปแบบเหตุการณ์ของภาพที่จะเป็นไปได้ซึ่งมีประเภทของวัตถุที่สนใจ

1.4.3 ทำการออกแบบโปรแกรม และเลือกวิธีการในการจัดการภาพ

1.4.4 ทำการทดลองกับภาพนิ่ง และทำการวิเคราะห์หาความเหมาะสมของค่าพารามิเตอร์

1.4.5 ทำการทดลองกับภาพเคลื่อนไหวและทำการวิเคราะห์หาความเหมาะสมของค่าพารามิเตอร์

1.4.6 สรุปงาน และจัดทำรายงาน



1.5 แผนการดำเนินงาน

ตาราง แสดงขั้นตอนการดำเนินโครงการ

กิจกรรม	ปี 2552							ปี 2553		
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1.5.1 ศึกษา OpenCV เพื่อใช้ในการ เขียนโปรแกรม ซึ่งกระทำเกี่ยวกับ ภาพ	←→									
1.5.2 เก็บข้อมูล รูปแบบเหตุการณ์ ของภาพที่จะเป็นไป ได้ซึ่งมีประเภทของ วัตถุที่สนใจ	←→									
1.5.3 ทำการ ออกแบบโปรแกรม และเลือกวิธีการใน การจัดการภาพ			←→							
1.5.4 ทำการ ทดลองกับภาพนิ่ง และทำการวิเคราะห์ หาความเหมาะสม ของค่าพารามิเตอร์					←→					

บทที่ 2

ทฤษฎีเบื้องต้น และหลักการควบคุม

การพัฒนากระบวนการควบคุมอัจฉริยะไร้คนขับโดยการประมวลผลภาพสามารถเข้ามาช่วยในการพัฒนาระบบ โดยการตัดสินใจควบคุมรถ และการหาตำแหน่งของรถ โดยอาศัยสัญญาณภาพ ซึ่งสามารถใช้ทฤษฎีดังต่อไปนี้เข้ามาช่วยในการพัฒนาระบบควบคุมอัจฉริยะไร้คนขับ โดยการประมวลผลภาพ ได้

2.1 การแยกคุณลักษณะ (Feature extraction)

กระบวนการในการแยกคุณลักษณะ (Feature extraction) [3,4] ต่างๆ ออกจากภาพเป็นขั้นตอนที่สำคัญที่จะนำไปสู่การถอดความหมายออกมาจากภาพ ดังนั้นการพิจารณาในแต่ละลักษณะที่เป็นไปได้จึงมีความแตกต่างกันออกไปดังนี้

สี (Color) เป็นลักษณะที่ใช้กันอย่างแพร่หลายมากที่สุดเนื่องจากสามารถทำได้ง่ายและมีความแม่นยำในระดับหนึ่ง โดยการเขียนแผนภูมิสี (Color Histogram) เป็นเทคนิคที่ได้รับความนิยมในการแทนลักษณะของสี เทคนิคนี้ใช้หลักของสถิติในการอธิบายคุณสมบัติของสี

รูปร่าง (Shape) แบ่งออกเป็น 2 ประเภทคือ เส้นขอบ (Boundary-based) และบริเวณ (Region-based) โดยเส้นขอบจะพิจารณาเฉพาะเส้นล้อมรอบวัตถุ ส่วนบริเวณจะพิจารณาพื้นที่ทั้งหมดที่อยู่ในเส้นล้อมรอบวัตถุ คุณลักษณะของรูปร่างที่ต้องการจะต้องไม่ขึ้นกับการเปลี่ยนแปลงตำแหน่ง การหมุน และการเปลี่ยน

พื้นผิว (Texture) เป็นรูปแบบการเรียงตัวของสี เป็นวิธีที่ดีในการแยกประเภทของภาพ แต่มีข้อเสียคือ ไม่สามารถอธิบายเป็นคำพูดที่ใช้อธิบายภาพได้ ซึ่งแบ่งเทคนิคต่างๆ ได้ 3 ประเภท คือ ความน่าจะเป็นกับสถิติ (Probabilistic and Statistical), แลปสี (Spectral) และ โครงสร้าง (Structural)

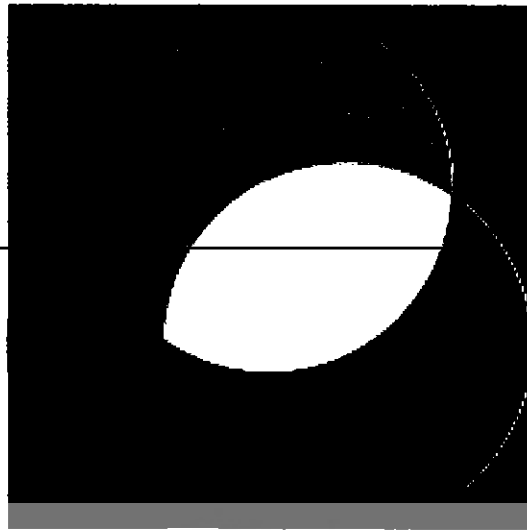
การทำดัชนีพื้นที่ของสี (Spatial Color Indexing) ในแต่ละวัตถุในภาพจะมีบริเวณเป็นของตัวเอง วิธีนี้จะกำหนดดัชนีให้กับบริเวณของวัตถุต่าง ๆ ในภาพ

วิธีแยกลักษณะของภาพได้อย่างถูกต้อง หรืออัตโนมัติเป็นเรื่องที่ต้องทำการค้นคว้าอีกมาก เนื่องจากภาพแต่ละภาพมีความแตกต่างกันมากในหลายเรื่อง เช่น มุมมอง สถานการณ์ และการแปรภาพ เป็นต้น

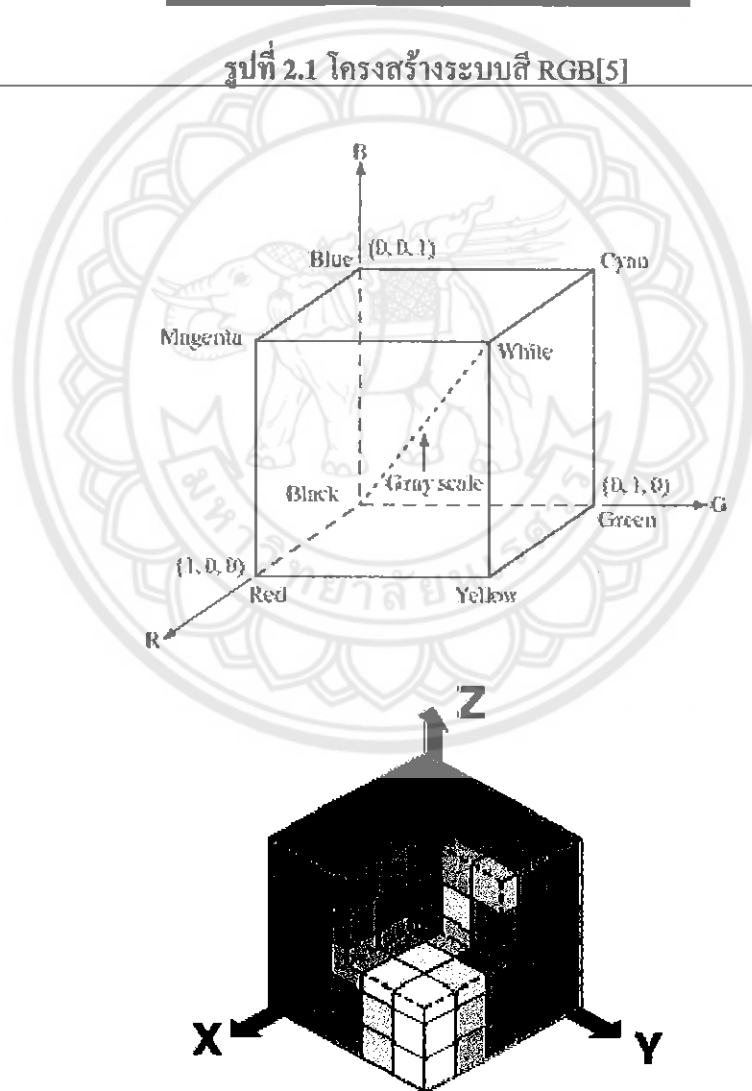
2.2 ระบบสี RGB

ระบบสี RGB[2,5] เป็นระบบสีของแสง ซึ่งเกิดจากการหักเหของแสงผ่านแท่งแก้วปริซึม ซึ่งจะเกิดแถบสีที่เรียกว่า สเปกตรัม (Spectrum) ซึ่งจะสามารถแยกสีตามที่สายตาเรามองเห็นได้ 7 สี คือ ม่วง คราม น้ำเงิน เขียว เหลือง แสด แดง ซึ่งเป็นพลังงานที่อยู่ในรูปของรังสี ที่มีช่วงคลื่นที่สายตาสามารถมองเห็นได้ แสงสีม่วงมีความถี่คลื่นสูงที่สุด คลื่นแสงที่มีความถี่สูงกว่าแสงสีม่วง เรียกว่า อัลตราไวโอเล็ต (Ultra Violet) และคลื่นแสงสีแดง มีความถี่คลื่นต่ำที่สุด ส่วนคลื่นแสงที่ต่ำกว่าแสงสีแดงเรียกว่า อินฟราเรด (Infrared) และคลื่นแสงที่มีความถี่สูงกว่าสีม่วง และต่ำกว่าสีแดงนั้น สายตาของมนุษย์ไม่สามารถรับได้

เมื่อศึกษาดูแล้วแสงสีทั้งหมดเกิดจากแสงสี 3 สี คือ สีแดง (Red) สีน้ำเงิน (Blue) และสีเขียว (Green) ทั้งสามสีถือเป็นแม่สีของแสง เมื่อนำมาฉายรวมกันจะทำให้เกิดสีใหม่ อีก 3 สี คือ สีแดงมาเจนต้า (Magenta) สีฟ้าไซแอน (Cyan) และสีเหลือง (Yellow) และถ้าฉายแสงสีทั้งหมดรวมกันจะได้แสงสีขาว จากคุณสมบัติของแสงนี้เรา ได้นำมาใช้ประโยชน์ทั่วไป ในการฉายภาพยนตร์ การบันทึกภาพวิดีโอ ภาพโทรทัศน์ การสร้างภาพเพื่อการนำเสนอทางจอคอมพิวเตอร์ และการจัดแสงสีในการแสดง เป็นต้น การรวมกันของแสงสีแดง เขียว และน้ำเงินนี้ มีการรวมกันแบบ Additive ซึ่งโดยปกติจะนำไปใช้ในจอภาพแบบ CRT และที่มีการเรียกระบบสีว่า RGB ก็มาจาก Red Green Blue นั่นเอง (Cathode ray tube) ในการใช้งานระบบสี RGB ยังมีการสร้างมาตรฐานที่แตกต่างกันออกไปที่นิยมใช้งาน ได้แก่ RGB-CIE และ RGB-NTSC



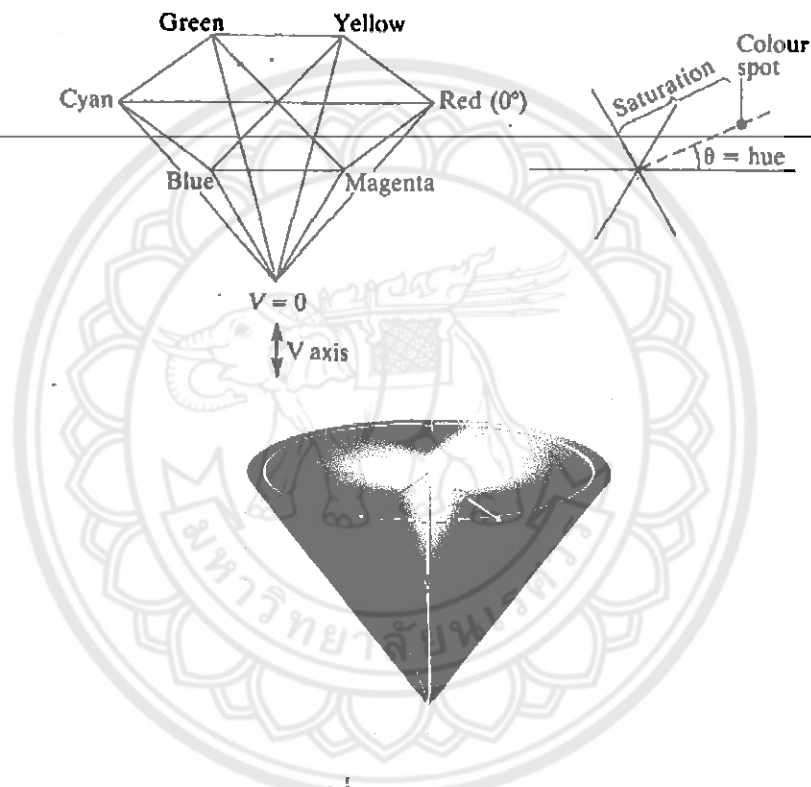
รูปที่ 2.1 โครงสร้างระบบสี RGB[5]



รูปที่ 2.2 ระบบสี RGB บน Cartesian coordinate system[6,7]

2.3 ระบบสี HSV

ระบบสี HSV[2] เป็นระบบสีที่อาศัยหลักการใช้ Hue Saturation และ Value โดย Hue คือค่าสีหลักทั้งสามสี ได้แก่ ค่าสีแดง สีเขียว และสีน้ำเงิน ซึ่งมีค่าอยู่ระหว่าง 0 ถึง 255 ถ้าเกิดค่า Hue มีค่าเท่ากับ 0 จะแทนให้เป็นสีแดง และเมื่อ Hue มีค่าเพิ่มขึ้นเรื่อย ๆ สีก็จะเปลี่ยนไปตามความถี่สเปกตรัมของสีจนถึง 256 แล้วจะกลับมาเป็นสีแดงเช่นเดิมอีกครั้ง และสามารถแทนให้อยู่ในรูปองศาได้ คือ สีแดง มีค่าเท่ากับ 0 องศา สีเขียว มีค่าเท่ากับ 120 องศา และสีน้ำเงิน มีค่าเท่ากับ 240 องศา



รูปที่ 2.3 แสดงระบบสี HSV[7]

Hue สามารถคำนวณได้จากระบบสี RGB ได้ดังนี้

$$\begin{aligned} red_k &= red - \min(red, green, blue) \\ green_k &= green - \min(red, green, blue) \\ blue_k &= blue - \min(red, green, blue) \end{aligned} \quad (2.1)$$

จากสมการที่ (2.1) เป็นการหาค่าของสี red , green , blue ที่ตำแหน่ง k โดยลักษณะโมเดลของระบบ Hue พบว่าจะมีค่าอย่างน้อยหนึ่งค่าที่จะเท่ากับ 0 แต่ถ้ามีสองค่าเท่ากับ 0 แล้ว hue จะเป็น

มุมของสี(ค่าสี)มีค่าเป็นไปตามสีที่สาม และถ้าทั้งสามสีมีค่าเท่ากับ 0 แล้วจะทำให้ไม่มีค่าของ Hue หรือสีที่ได้จะมีค่าเท่ากับสีขาวนั่นเอง ตัวอย่างเช่น จอภาพขาว-ดำ ถ้าเกิดมีสีใดสีหนึ่งมีค่าเท่ากับ 0 จะทำให้ค่าสีที่ได้เป็นไปตามสีที่เหลือ การให้น้ำหนักในการพิจารณาเมื่อสีแดงมีค่าเท่ากับ 0 ด้วย ซึ่งการคำนวณหาค่า Hue เป็นดังสมการ ที่ (2.2) ซึ่งใช้ในการคำนวณหาค่า Hue ตำแหน่งที่ k ใด ๆ

$$Hue = \frac{(240 \times blue_k) + (120 \times green_k)}{blue_k + green_k} \quad (2.2)$$

Saturation คือความบริสุทธิ์ของสี ถ้า Saturation มีค่าเท่ากับ 0 หมายความว่า สีที่ได้จะเป็นสีขาว แต่ถ้า Saturation มีค่าเท่ากับ 255 หมายความว่า ไม่มีแสงสีขาวผสมอยู่ ซึ่งการคำนวณหาค่า Saturation เป็นดังสมการที่ (2.3)

$$Saturation = \frac{\max(red, green, blue) - \min(red, green, blue)}{\max(red, green, blue)} \quad (2.3)$$

Value คือ ความสว่างของสี ซึ่งสามารถวัดได้โดยค่าความเข้มของความสว่างของแต่ละสีที่ประกอบกันสามารถคำนวณได้จาก ซึ่งการคำนวณหาค่า Value เป็นดังสมการที่ (2.4)

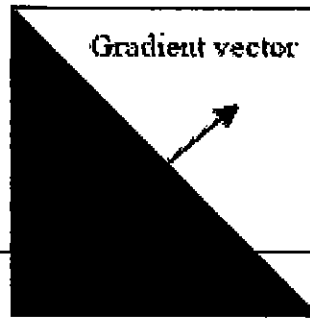
$$Value = \max(red, green, blue) \quad (2.4)$$

2.4 การหาขอบภาพ (Edge Detection)

การหาขอบภาพ (Edge Detection)[1,4] คือ การตรวจสอบว่าเส้นขอบลากผ่านหรือใกล้เคียงกับจุดใด โดยคำนวณจากการเปลี่ยนแปลงของความเข้มในตำแหน่งที่ใกล้เคียงกับจุดดังกล่าว ซึ่งวิธีการหาขอบนั้นมีด้วยกันหลายวิธี แต่อย่างไรก็ตามสามารถแบ่งได้เป็น 2 กลุ่มหลัก คือวิธี Gradient และวิธี Laplacian โดยในแต่ละวิธีมีรายละเอียดดังต่อไปนี้

2.4.1 วิธี Gradient

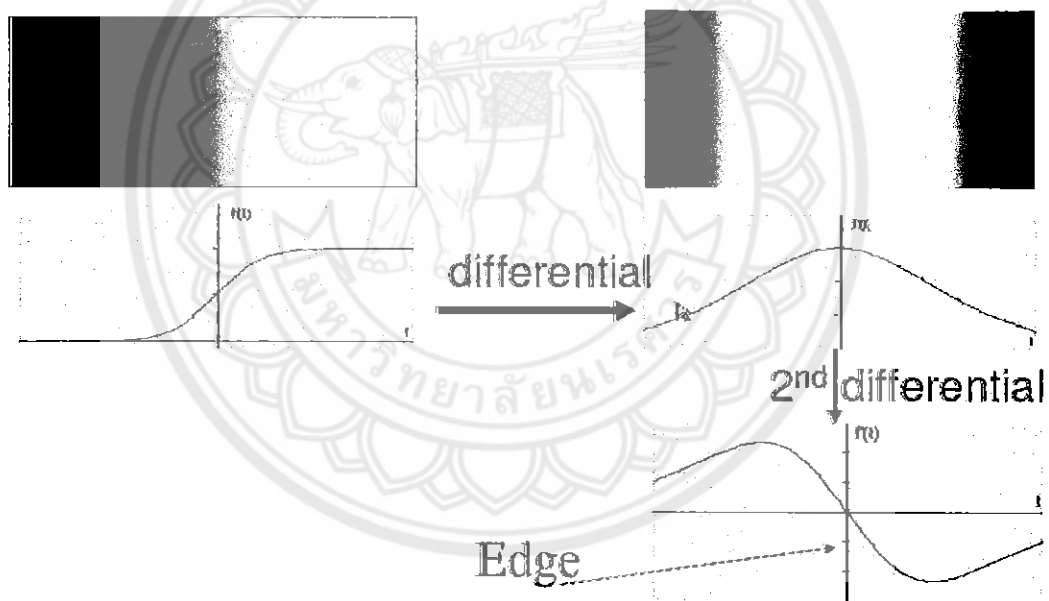
วิธี Gradient[1,4] เป็นความคิดแรกที่ใช้วิเคราะห์หาขอบภาพ โดยใช้วิธีการวัดความเปลี่ยนแปลงความเข้ม (Gray level) ของจุดภาพที่กำลังพิจารณากับจุดภาพที่อยู่ข้างเคียง เพื่อการตัดสินใจต่อไปว่าเป็นขอบภาพหรือไม่ โดยวิธีนี้จะหาขอบด้วยการหาจุดต่ำสุดและจุดสูงสุดในรูปของอนุพันธ์อันดับหนึ่งของภาพ โดยจุดที่เป็นขอบจะอยู่ในส่วนที่เหนือค่า Threshold จึงอาจทำให้เส้นขอบที่ได้มีลักษณะหนา ตัวอย่างวิธีการหาขอบของกลุ่มนี้ เช่น Roberts, Prewitt และ Canny



รูปที่ 2.4 การหาขอบภาพโดยวิธี Gradient[4]

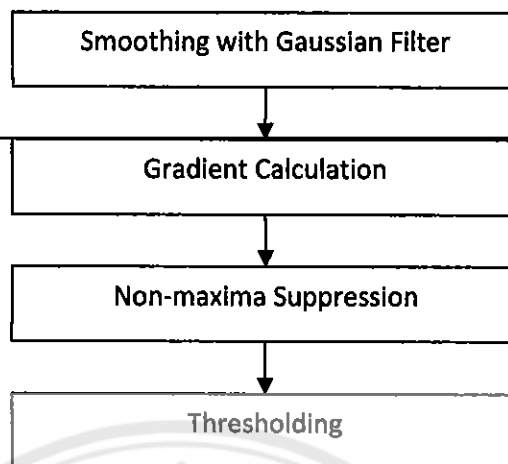
2.4.2 วิธี Laplacian

วิธี Laplacian[1,4] จะหาขอบโดยใช้อนุพันธ์อันดับ 2 โดยใช้จุดที่ค่า เป็น 0 ซึ่งวิธีนี้จะใช้เวลาในการคำนวณมากกว่าวิธี Gradient สามารถใช้กรองความถี่ต่ำได้ขอบโครงภาพ ตัวอย่างวิธีการหาขอบของกลุ่มนี้ เช่น Laplacian of Gaussian และ Marrs-Hildreth เป็นต้น



รูปที่ 2.5 การหาขอบภาพโดยวิธี Laplacian[7]

2.5 Canny Edge Detection Algorithm



รูปที่ 2.6 ขั้นตอนวิธีการหาขอบโดยวิธี Canny[6]

การทำงานของ Canny edge detection [3,6] นั้นเริ่มต้นจากการปรับภาพให้เรียบ (Smoothing) ด้วยตัวกรองเกาส์เซียน (Gaussian filter) เพื่อกำจัดสัญญาณรบกวน หลังจากนั้นจะคำนวณค่าขนาด (magnitude) และทิศทาง (orientation) ของ gradient โดยใช้การหาอนุพันธ์อันดับหนึ่ง ถัดมาจึงใช้ nonmaxima suppression กับขนาด (magnitude) ของ gradient เพื่อให้ได้ขอบที่บางลง และในขั้นตอนสุดท้ายจะใช้ double thresholding algorithm เพื่อหาพิกเซลที่เป็นขอบและทำการเชื่อมต่อขอบโดยในแต่ละขั้นตอนมีรายละเอียดดังต่อไปนี้

2.5.1 Smoothing

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

2.5.2 Nonmaxima Suppression

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

2.5.3 Thresholding

แม้ว่าภาพจะผ่านการ smoothing ในขั้นตอนแรกแล้วก็ตาม ภาพที่ได้อาจยังมีเส้นขอบที่ไม่ใช่ขอบที่แท้จริงปรากฏอยู่อันเนื่องจากสัญญาณรบกวนหรือลักษณะของวัตถุในภาพเป็นพื้นผิวที่มีลวดลายหรือมีรายละเอียดภายในมาก ดังนั้นเพื่อลดปัญหาดังกล่าวจึงได้มีการกำหนดค่า threshold ขึ้นมา 2 ค่า คือ high threshold (T_1) และ low threshold (T_2) โดยพิกเซลที่มีค่ามากกว่า T_1 จะถูกปรับเป็น 1 (เป็นพิกเซลที่เป็นขอบ) แต่ถ้าน้อยกว่า T_2 จะถูกปรับเป็น 0 ส่วนค่าที่อยู่ระหว่างค่า threshold ทั้งสอง การปรับเป็นค่า 0 หรือ 1 นั้นขึ้นอยู่กับพิกเซลที่อยู่รอบข้าง หากพบว่าพิกเซลที่อยู่รอบข้างของพิกเซลที่เป็นขอบ (ค่า $>T_1$) มีค่ามากกว่า T_2 แล้ว จะปรับค่าพิกเซลดังกล่าวให้มีความเป็น 1 และถือเป็นหนึ่งในขอบภาพด้วยเช่นกัน

2.6 การกรองภาพ (Image Filtering)

การกรองภาพ (Image Filtering) [1,3] คือ การนำภาพผ่านตัวกรองสัญญาณเพื่อให้ได้ผลลัพธ์ออกมา ผลลัพธ์ที่ได้ขึ้นอยู่กับตัวกรองที่นำมาใช้ จุดประสงค์ของการกรองข้อมูลภาพ คือ การเน้น (Enhance) หรือการลดทอน (Attenuate) คุณสมบัติบางประการของภาพ เพื่อให้ได้ภาพที่มีคุณสมบัติตามต้องการ Gonzalez, Woods และ Eddins กล่าวถึงการกรองภาพว่า เป็นการประมวลผลภาพอย่างหนึ่งที่จำเป็นมาก เนื่องจากในการใช้งานจริง ภาพส่วนมากมักจะมีสัญญาณรบกวน หรือสัญญาณไม่พึงประสงค์อื่นๆ ปะปนอยู่ด้วย การกรองสามารถปรับปรุงให้ภาพมีคุณสมบัติที่ดีขึ้นเหมาะสมกับการประมวลผลในขั้นต่อไป

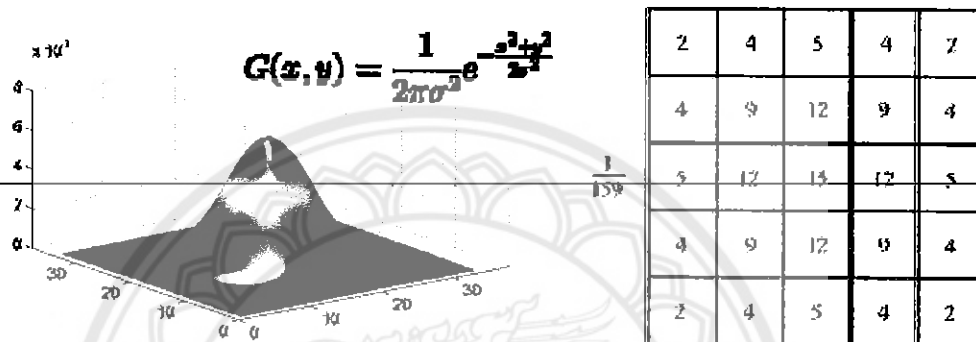
2.6.1 การกรองเฉลี่ย (Mean Moving Average Filtering)

เป็นการทำคอนโวลูชัน โดยใช้ตัวทาบที่มีค่าน้ำหนักเท่ากันทุกช่อง คือ S_{21} เมื่อ S คือ ขนาดของตัวทาบจัตุรัส และมีหน่วยเป็นจุดภาพ เป็นการกรองแบบเชิงเส้นและเป็นชนิดที่ขมให้

ความถี่ต่ำผ่าน (Low Pass Filter) ข้อดีของวิธีการนี้ คือ ใช้สำหรับการกำจัดสัญญาณรบกวนแบบ Gaussian ได้ดี แต่จะทำให้ภาพผลลัพธ์เบลอเส้นขอบภาพไม่ชัดเจน

2.6.2 การกรองแบบ Gaussian Filtering

เป็นการทำคอนโวลูชัน โดยใช้ตัวทาบที่มีค่าน้ำหนักลักษณะสมมาตรในเชิงวงกลม (Circularly Symmetric) ซึ่งมีค่าน้ำหนักแปรผันตามลักษณะการกระจายแบบ Gaussian ดังรูปที่ 2.8



รูปที่ 2.8 ตัวอย่างของ Gaussian Filter พารามิเตอร์หนึ่ง[7]

การกรองชนิดนี้เป็นแบบเชิงเส้นและเป็นชนิดที่ยอมให้ความถี่ต่ำผ่านข้อดี คือ ใช้สำหรับการกำจัดสัญญาณรบกวนแบบ Gaussian ได้ดี แต่จะทำให้ภาพเบลอ เส้นขอบภาพไม่ชัดเจน

2.6.3 การกรองแบบ Median Filtering (Order-Statistic Filtering) Distribution

โดยใช้ตัวทาบที่มีค่าน้ำหนักเท่ากันตลอด การกรองชนิดนี้เป็นแบบไม่เชิงเส้น (Nonlinear and Spatial Filtering) และเป็นชนิดที่ยอมให้ความถี่ต่ำผ่านได้ การกรองลักษณะนี้ แบ่งได้เป็น Running Median Smoothers และ Weighted Median Smoothers ข้อดีคือ ใช้สำหรับการกำจัดสัญญาณรบกวนแบบ Salt-and-pepper Noise และ Laplacian Noise ได้ดี อีกทั้งไม่ทำให้ภาพผลลัพธ์เบลอ และเส้นขอบยังคงชัดเจน

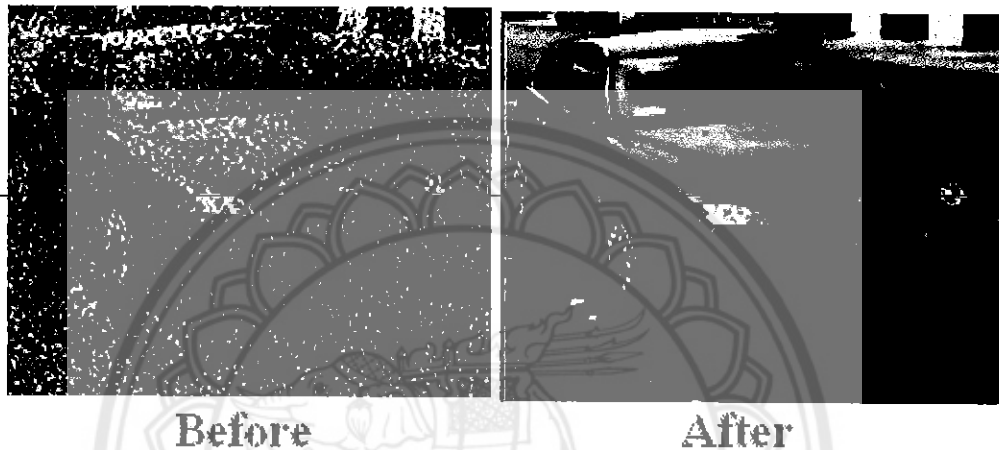
2.6.4 การกรองแบบ High-pass Filtering (Sharpening Edge Crispening)

เป็นการทำ Spatial Convolution ซึ่งใช้ตัวทาบที่มีค่าน้ำหนักตรงกลางเป็นค่าบวก และค่าน้ำหนักในช่องอื่นเป็นค่าลบ หรือศูนย์ เพื่อให้ผลรวมของค่าน้ำหนักทุกช่องบนตัวทาบรวมกัน

เป็นหนึ่งในมุ่งเน้นทำให้ภาพคมชัด กรองสัญญาณรบกวนออกจากภาพ ทำให้รายละเอียดของภาพปรากฏเด่นชัด

2.6.5 การกรองแบบ Low Pass Filter

ใช้สำหรับการสลายสัมพันธ์บริเวณรอยต่อของภาพ ทำให้ภาพบริเวณนั้นเบลอ ไม่ชัดเจน ซึ่งใช้ในการกำจัด noise



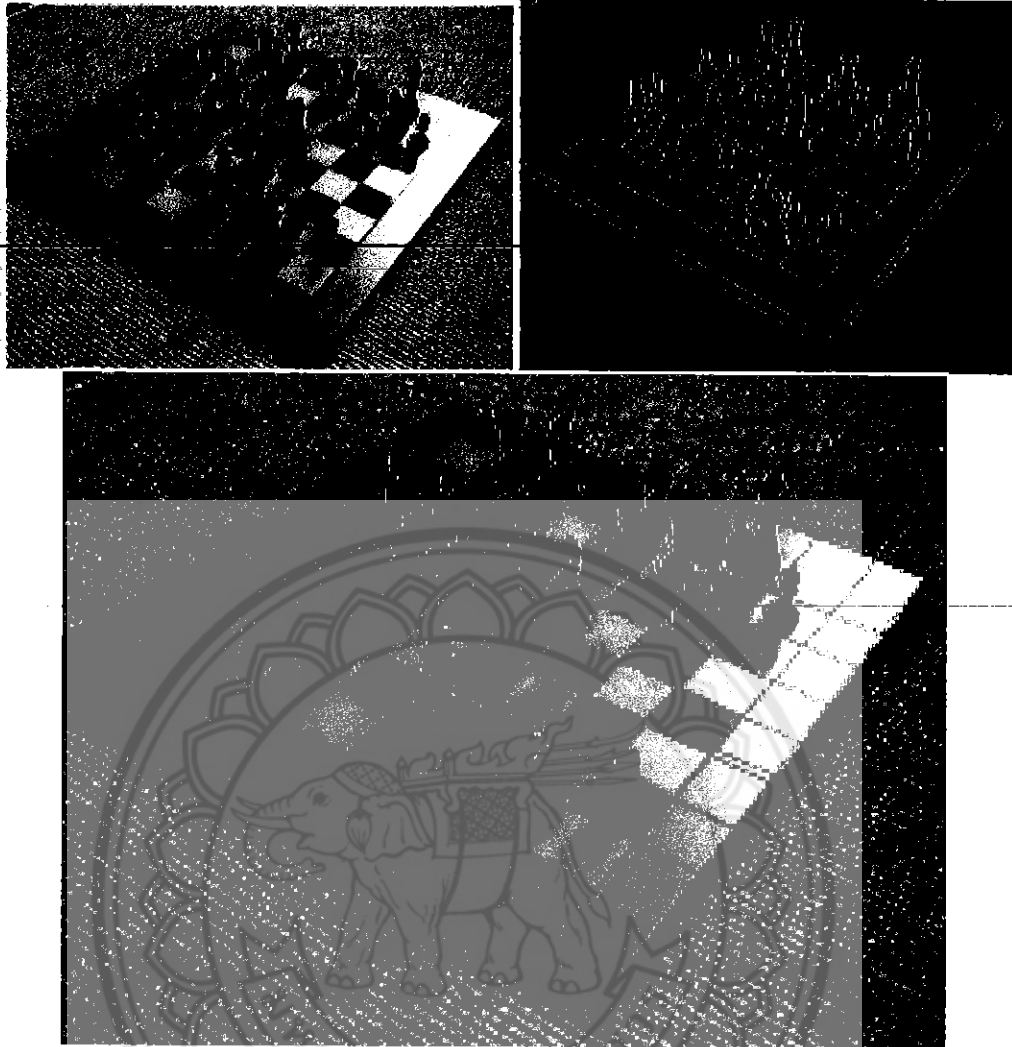
รูปที่ 2.9 ตัวอย่างของ Low Pass Filter[7]

2.7 Hough Transform

จุดประสงค์ของการใช้ Hough Transform [7,9] คือ จุดเริ่มต้นในการกำหนดจุดที่เป็นไปได้ ในจุดของขอบภาพของวัตถุ ที่มีประสิทธิภาพในการแสดงผลของขอบภาพที่ชัดเจน ซึ่งจากจุดที่ได้มานั้นนำมาใช้เป็นพารามิเตอร์ในสมการที่ (2.5) และ (2.6) ซึ่งสามารถนำมาสร้างเส้นตรง วงกลม หรือรูปวงรี ได้

$$y = \left(\frac{-\cos\theta}{\sin\theta} \right) x + \left(\frac{r}{\sin\theta} \right) \quad (2.5)$$

$$r = x \cos\theta + y \sin\theta \quad (2.6)$$



รูปที่ 2.10 ตัวอย่างของ Hough Transform[7]



รูปที่ 2.11 ตัวอย่างของ Hough Transform[9]

2.7 OpenCV (Open Source Computer Vision Library)

OpenCV[9] เป็นไลบรารีสำหรับใช้งานเรื่องการประมวลผลภาพ (Image Processing) นอกจากนี้ OpenCV ยังสามารถจัดการกับข้อมูลในรูปแบบที่เป็นวิดีโอได้อีกด้วย เนื่องจาก OpenCV เป็นชุดคำสั่งที่ไม่ได้เป็นตัวโปรแกรม เมื่อต้องการเรียกใช้งานจึงต้องเขียนโปรแกรมเพื่อเรียกชุดคำสั่งเหล่านั้น ซึ่งภาษาที่นิยมเขียนคือภาษา C, C++ และภาษา Python ซึ่ง OpenCV จะประกอบด้วยสอง ส่วน คือ data structure ที่ใช้ในการเก็บข้อมูลต่าง ๆ อาทิ เช่น รูปภาพ เมตริกซ์ และพิกัด สำหรับอีก ส่วนคือ algorithm ซึ่งจะใช้ในการประมวลผลต่าง ๆ โดยเฉพาะการประมวลผลทางรูปภาพ

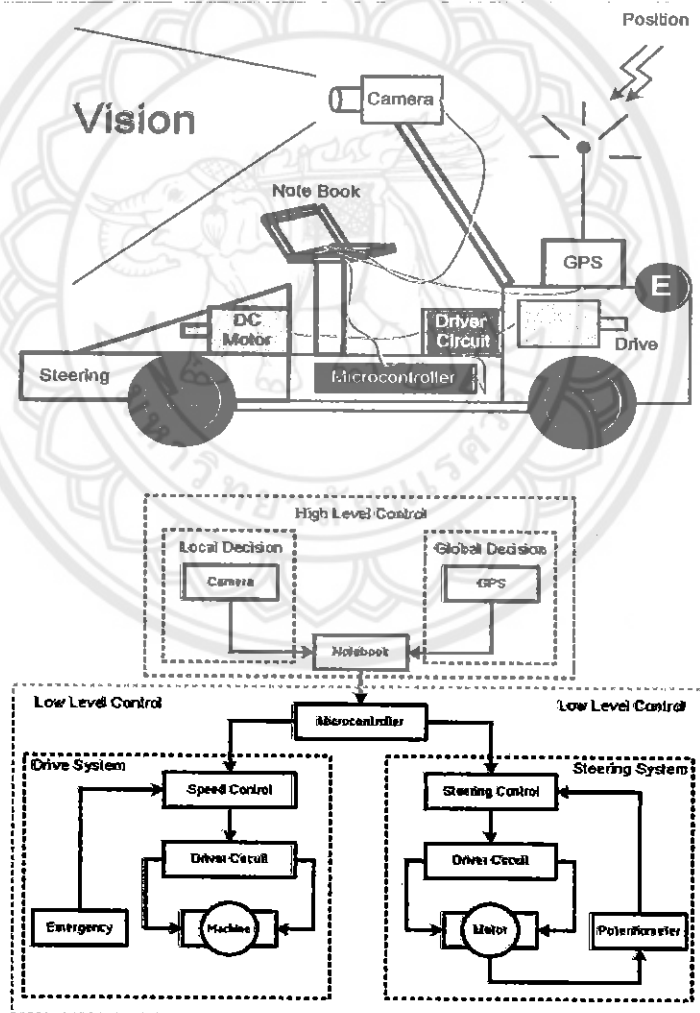
โดยทฤษฎีดังกล่าวข้างต้นจะสามารถเข้ามาช่วยในการพัฒนาระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการประมวลผลภาพ เพื่อให้ระบบควบคุมรถอัจฉริยะไร้คนขับมีประสิทธิภาพมากยิ่งขึ้น ซึ่งขั้นตอนการออกแบบ และวิธีทำจะอยู่ในบทถัดไป



บทที่ 3

การวิเคราะห์ ออกแบบ และพัฒนาระบบ

จากการศึกษาค้นคว้าข้อมูลและทฤษฎีข้างต้น ทำให้สามารถเข้าใจถึงหลักการและวิธีการในการดำเนินงาน ในบทนี้เราจะนำความรู้ที่ได้มาประยุกต์ใช้ในการพัฒนาระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการประมวลผลภาพ ซึ่งสามารถเข้ามาช่วยในการพัฒนาระบบ ในการตัดสินใจควบคุมรถ และการหาตำแหน่งของรถ โดยอาศัยสัญญาณภาพ ซึ่งการออกแบบ และพัฒนาระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการประมวลผลภาพ มีขั้นตอน และวิธีทำดังต่อไปนี้



รูปที่ 3.1 แสดงระบบ โดยรวมของรถอัจฉริยะไร้คนขับ

3.1 วิเคราะห์ระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการประมวลผลภาพ

โดยทั่วไปแล้วระบบควบคุมรถอัจฉริยะไร้คนขับ อาจแบ่งออกได้เป็นสองส่วนหลัก คือ ส่วนควบคุมระดับบน และส่วนควบคุมระดับล่าง โดยส่วนควบคุมระดับบนจะทำหน้าที่ในการตัดสินใจโดยรวมของการขับเคลื่อนของรถ ซึ่งอาศัยข้อมูลจากระบบบอกตำแหน่งพิกัดโลก หรือ GPS เพื่อระบุเส้นทางที่ต้องการให้รถเดินทางไปสู่จุดหมาย และข้อมูลภาพจากกล้องวีดีโอบริเวณหน้ารถ เพื่อใช้ในการคำนวณตำแหน่งของถนน และอุปสรรคสิ่งกีดขวาง เป็นต้น ในส่วนของส่วนควบคุมระดับล่างจะทำหน้าที่บังคับรถต่าง ๆ ของรถ

การพัฒนา ระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการประมวลผลภาพ จะทำการหาตำแหน่งรถ และการตัดสินใจควบคุมรถ โดยอาศัยสัญญาณภาพ จะมีความสำคัญมาก โดยเฉพาะการตัดสินใจระยะใกล้ ซึ่งหากพิจารณาถึงวัตถุที่เกี่ยวข้องในสนามแข่ง จะพบว่า มีวัตถุที่สนใจอยู่ทั้งหมด 2 ประเภท คือ ถนน และสิ่งกีดขวาง จากข้อมูลเหล่านี้แสดงให้เห็นว่า วัตถุที่จำเป็นในระบบการมองเห็นทั้ง 2 ชนิดนี้ จะต้องถูกคัดกรองออกมาจากภาพที่ได้รับได้ และทำการแยกแยะประเภทวัตถุดังกล่าวก่อน จากนั้นจึงทำการหาความหมาย และการตัดสินใจเมื่อพบวัตถุนั้น ๆ ต่อไป

การพัฒนา ระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการประมวลผลภาพ มีส่วนการออกแบอยู่ 2 ส่วนหลัก ๆ คือ ส่วนการออกแบบระบบภายนอก และส่วนการออกแบบระบบภายใน ซึ่งแต่ละส่วนมีความต้องการการใช้งานอุปกรณ์ ทฤษฎี และความสามารถของการทำงานในระบบนั้น ๆ โดยรายละเอียดของแต่ละส่วนจะอธิบายในหัวข้อต่อไป

3.2 การออกแบบระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการประมวลผลภาพ

การพัฒนา ระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการประมวลผลภาพ มีส่วนการออกแบอยู่ 2 ส่วนหลัก ๆ คือ ส่วนการออกแบบระบบภายนอก และส่วนการออกแบบระบบภายใน โดยมีรายละเอียดดังต่อไปนี้

3.2.1 การออกแบบระบบภายนอก

การออกแบบระบบภายนอก จะทำการกำหนดรายละเอียดของอุปกรณ์ และการเชื่อมต่อของแต่ละอุปกรณ์ที่ต้องใช้ในการพัฒนาระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการประมวลผลภาพ โดยมีอุปกรณ์ดังต่อไปนี้

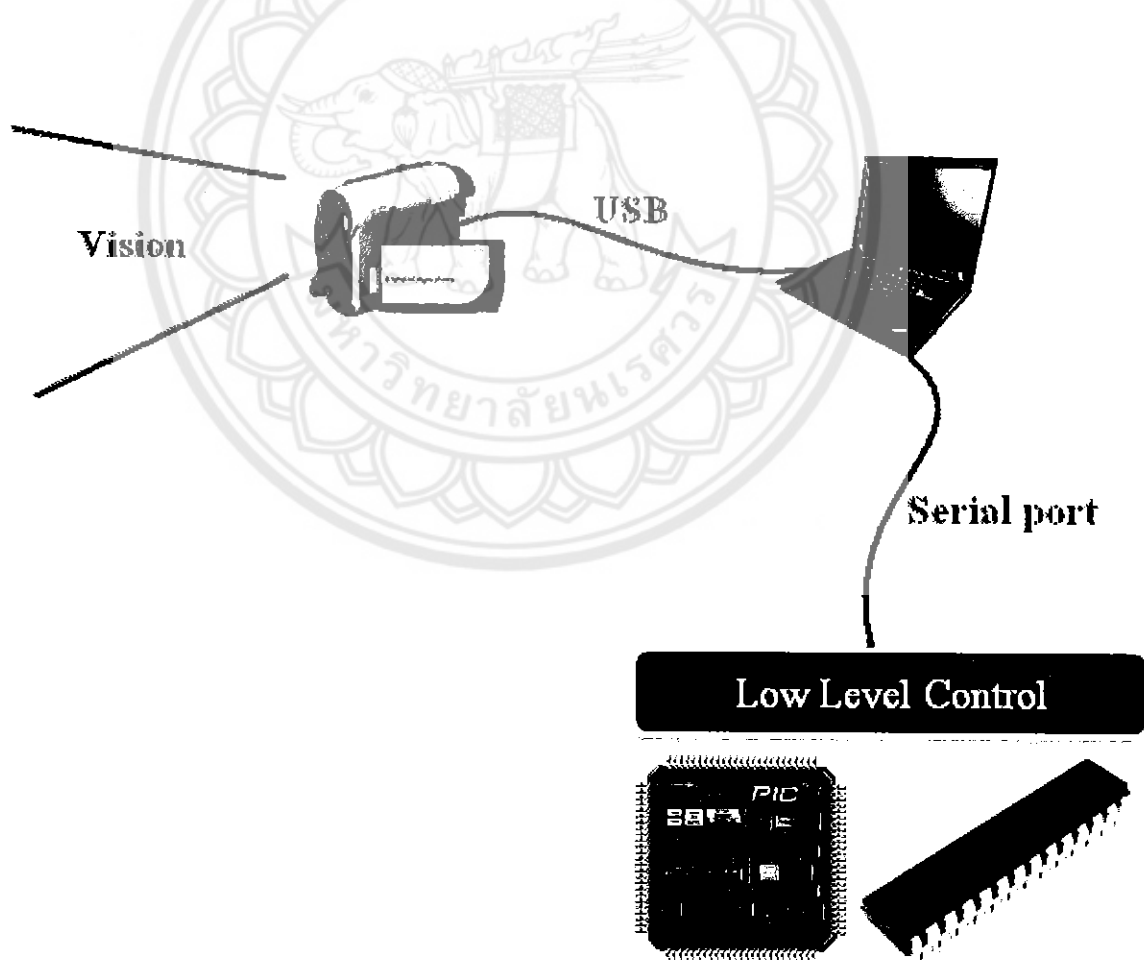
3.2.1.1 กล้องวิดีโอ 1 ตัว ใช้ในการรับภาพ

3.2.1.2 คอมพิวเตอร์ส่วนบุคคล 1 เครื่อง ใช้ในการควบคุมระบบ

3.2.1.3 ทำการเชื่อมต่อกล้องวิดีโอ กับคอมพิวเตอร์ส่วนบุคคล ผ่านสาย USB

3.2.1.4 ทำการเชื่อมต่อคอมพิวเตอร์ส่วนบุคคล กับส่วนควบคุมระดับล่าง ผ่านสายอนุกรม

โดยรายละเอียดของการออกแบบระบบภายนอก โดยมีการใช้อุปกรณ์ และการเชื่อมต่ออุปกรณ์ต่างๆ จะแสดงดังรูปที่ 3.2

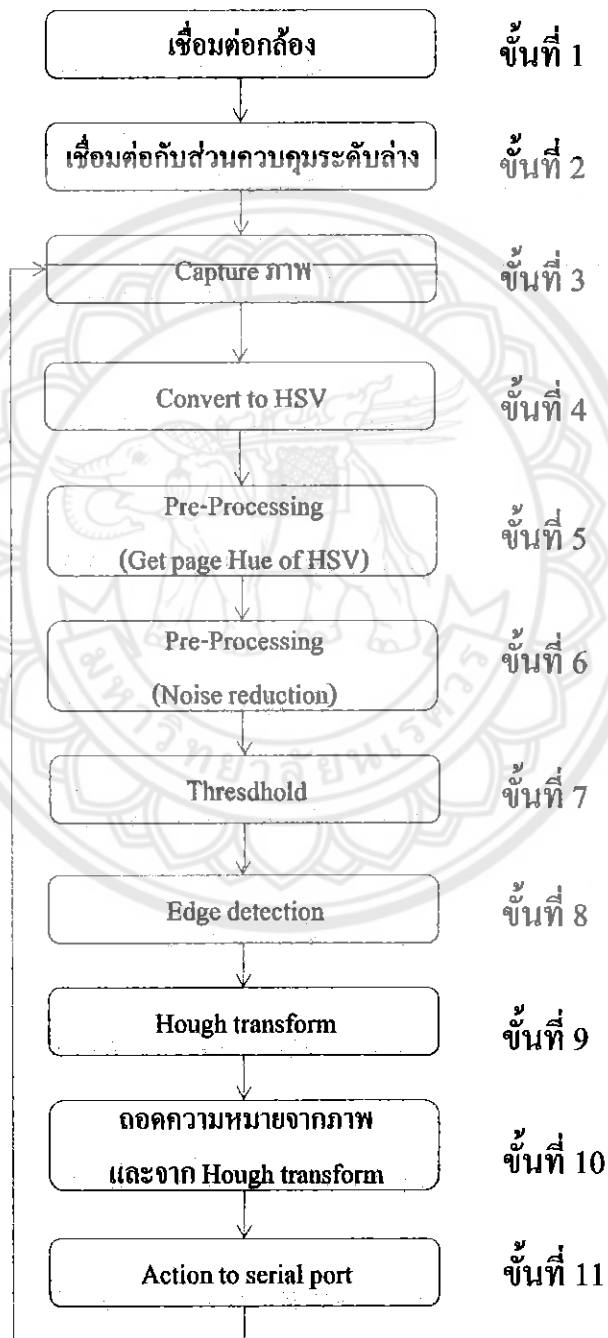


รูปที่ 3.2 แสดงการออกแบบระบบภายนอก[10-13]

3.2.2 การออกแบบระบบภายใน

การออกแบบระบบภายใน คือ การออกแบบส่วนของโปรแกรมที่ใช้ทำการควบคุมระบบควบคุมรถอัจฉริยะไร้คนขับ โดยทำการแสดงรายละเอียดส่วนการทำงานของโปรแกรม ซึ่งทำการอธิบายว่าแต่ละส่วนใช้ทฤษฎีเรื่องใดเข้ามาช่วยในการทำงานของส่วนการทำงานนั้น โดยแสดงดัง

รูปที่ 3.3

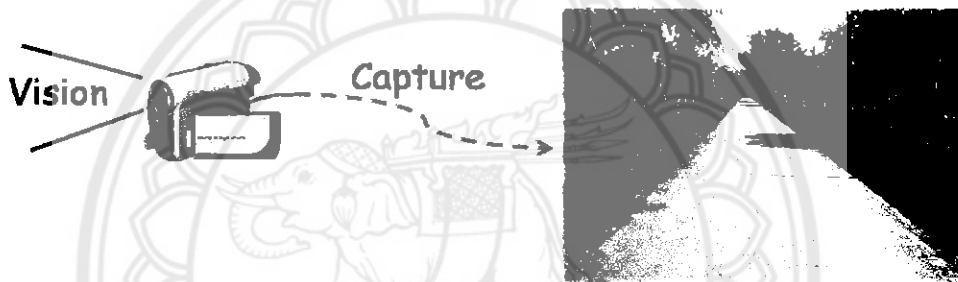


รูปที่ 3.3 แสดงการออกแบบระบบภายใน

3.2.1 ระบบภายในขั้นที่ 1 การเชื่อมต่อกล้องวิดีโอ กับคอมพิวเตอร์ส่วนบุคคล ผ่านสาย USB

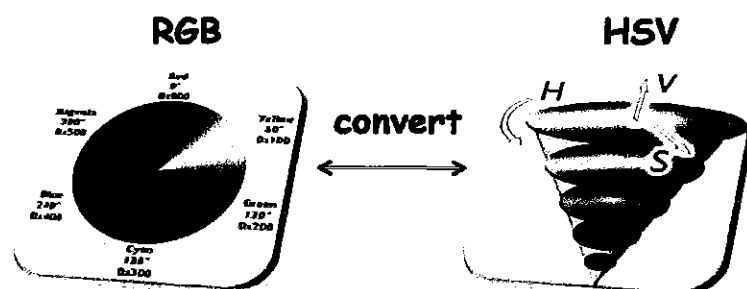
3.2.2 ระบบภายในขั้นที่ 2 การเชื่อมต่อคอมพิวเตอร์ส่วนบุคคล กับส่วนควบคุมระดับล่าง ผ่านสายอนุกรม(serial) โดยจะทำการกำหนดอัตราความเร็วในการรับส่งข้อมูล(Baud rate) และ เลขที่ของพอร์ตอนุกรม

3.2.3 ระบบภายในขั้นที่ 3 การ Capture ภาพ คือ การจับภาพเพียงภาพเดียวจาก ภาพเคลื่อนไหวที่ได้จากกล้องวิดีโอ โดยภาพเคลื่อนไหวที่ได้จากกล้องวิดีโอ เป็นภาพนิ่งหลาย ๆ ภาพเรียงต่อกันทำให้เกิดเป็นภาพเคลื่อนไหวขึ้นมาได้



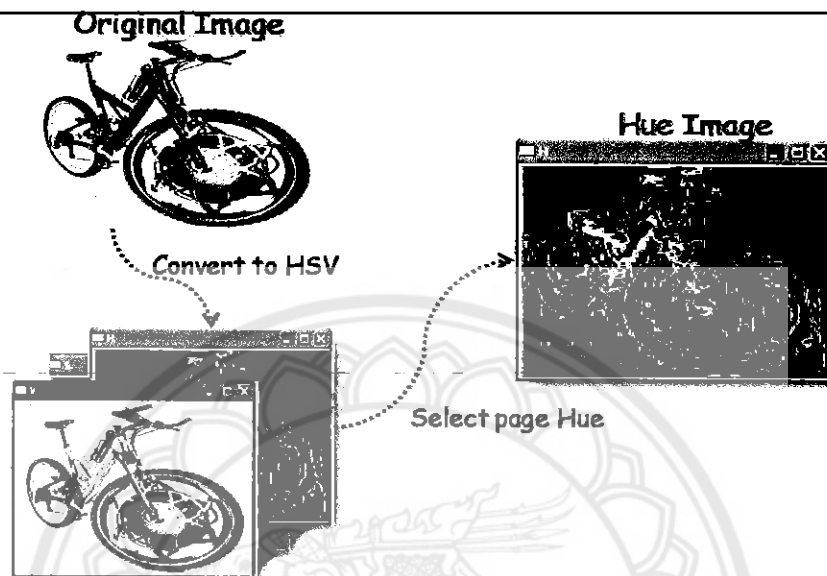
รูปที่ 3.4 แสดงการทำ Capture ภาพจากกล้องวิดีโอ[13]

3.2.4 ระบบภายในขั้นที่ 4 การ Convert to HSV ภาพ คือ การแปลงจากภาพสี RGB เป็น ภาพสี HSV โดยจะนำภาพที่ได้จากการ Capture ภาพ ซึ่งเป็นภาพสี RGB มาเข้าฟังก์ชันเพื่อแปลง ค่าของสีให้อยู่ในโซนของระบบสี HSV



รูปที่ 3.5 แสดงการทำ Convert ระบบสี RGB to HSV[14,15]

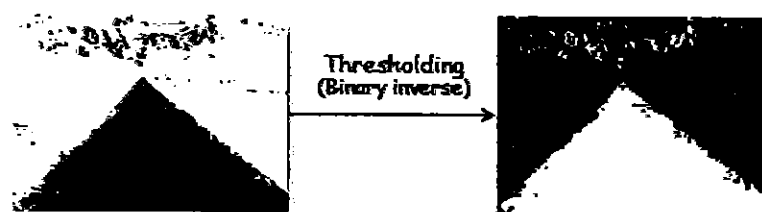
3.2.5 ระบบภายในขั้นที่ 5 การทำ Pre-Processing ในส่วนของการทำ Get page Hue of HSV คือ การเลือก page Hue จากรูปสี HSV ซึ่ง page Hue ที่ได้จะอยู่ในโซนของสีขาวดำ หรือ เทา ซึ่งมีค่าแต่ละพิกเซลตั้งแต่ 0-255



รูปที่ 3.6 แสดงการทำ Get page Hue of HSV

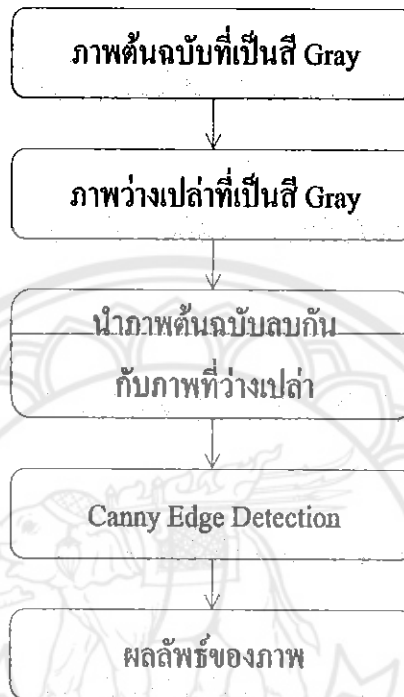
3.2.6 ระบบภายในขั้นที่ 6 การทำ Pre-Processing ในส่วนของการทำ Noise reduction คือ การกรองสัญญาณรบกวนออกจากภาพ ซึ่งจะใช้การกรองแบบ Gaussian Filtering โดยจะทำให้ขอบภาพเบลอ

3.2.7 ระบบภายในขั้นที่ 7 การทำ Threshold ในส่วนของการทำ Threshold คือ การกรองสัญญาณรบกวนออกจากภาพอีกชั้นหนึ่ง ซึ่งจะใช้วิธี Binary inverse โดยกระบวนการนี้จะทำให้เส้นขอบชัดเจนขึ้น

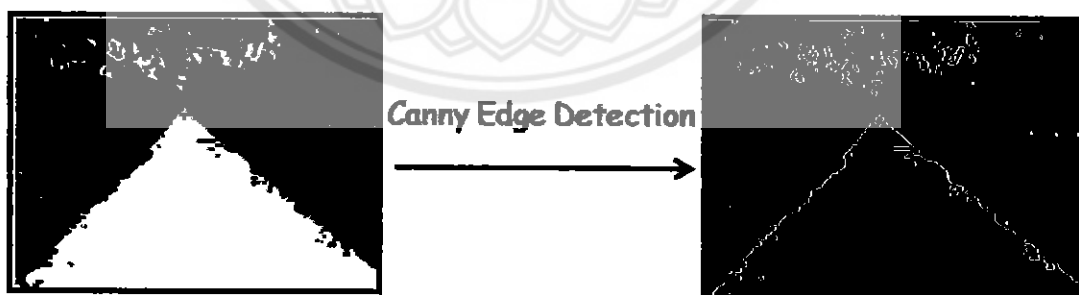


รูปที่ 3.7 แสดงการทำ Threshold

3.2.8 ระบบภายในขั้นที่ 8 การทำ Edge detection คือ การหาขอบของภาพ โดยจะทำการตรวจสอบว่าเส้นขอบลากผ่านหรือใกล้เคียงกับจุดใด โดยคำนวณจากการเปลี่ยนแปลงของความเข้มในตำแหน่งที่ใกล้เคียงกับจุดดังกล่าว โดยมีหลักการทำงานดังรูปที่ 3.8 และรูปที่ 3.9

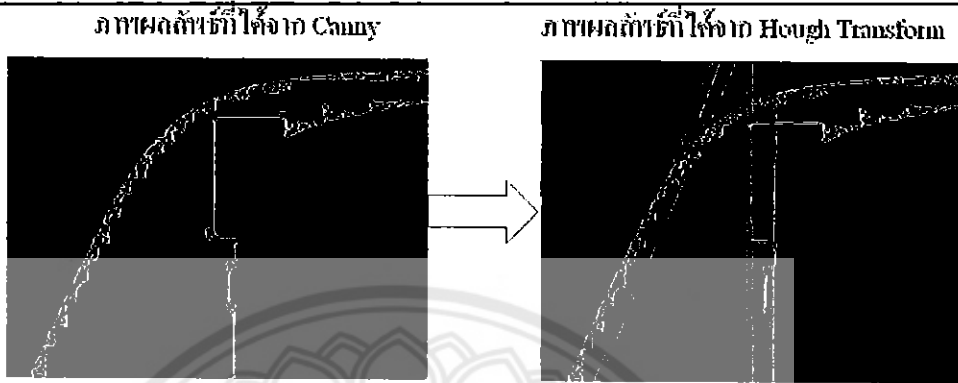


รูปที่ 3.8 แสดงการทำ Edge detection แบบ Flowchart



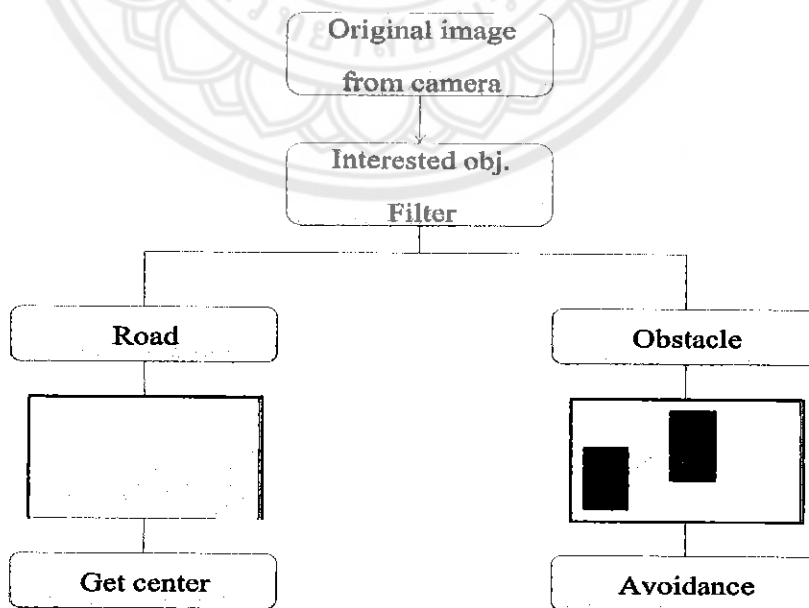
รูปที่ 3.9 แสดงการทำ Edge detection

3.2.9 ระบบภายในขั้นที่ 9 การทำ Hough transform คือ การจุด 2 จุดบนขอบภาพที่อยู่บนระนาบเดียวกันที่ได้จากการทำ Canny Edge Detection แล้วทำการลากเส้นตรงตัดผ่านจุด 2 จุดนั้น โดยมีการทำงานดังรูปที่ 3.10



รูปที่ 3.10 แสดงการทำ Hough transform

3.2.10 ระบบภายในขั้นที่ 10 การถอดความหมายจากภาพ และจาก Hough transform คือ การที่สามารถบอกได้ว่าส่วนใดคือ ถนน ส่วนใดคือ วัตถุที่เป็นสิ่งกีดขวาง เพื่อวิเคราะห์ได้ว่า จะกระทำการสิ่งใดกับวัตถุชิ้นนั้น ๆ ซึ่งสามารถแยกวัตถุ ได้ตามรูปที่ 3.11 และรูปที่ 3.12



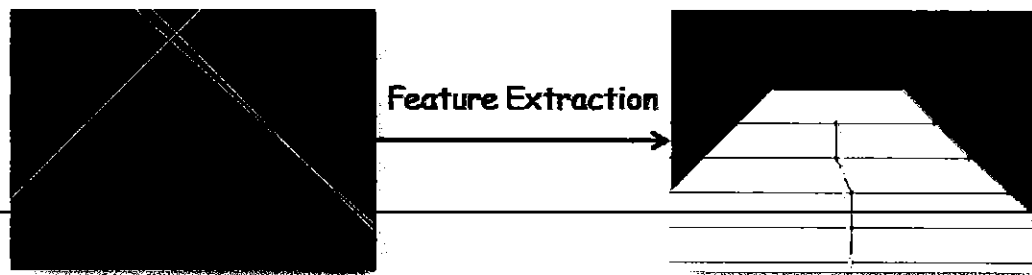
15755281

ร/ร.

คท 2/35

2552

รูปที่ 3.11 แสดงการแยกวัตถุ



รูปที่ 3.12 แสดงการถอดความหมายจากภาพ

จากรูปที่ 3.12 เป็นการถอดความหมายจากภาพที่ได้จากการทำ Hough transform ซึ่งผลลัพธ์ที่ได้ คือ พื้นถนนจะถูกถมด้วยสีขาว จุดกึ่งกลางของถนนคือเส้นสีน้ำเงิน เส้นสีแดงเป็นเส้นอ้างอิงที่ใช้ในการหาจุดกึ่งกลางของถนน และเส้นสีเหลืองจะสามารถบอกได้ว่าตอนนี้จะสั่งให้รถเลี้ยวที่มุมกี่องศา ซึ่งทำทำมุมกับแนวแกน x

เส้นสีแดง เกิดจากการกำหนดค่าเพื่อใช้อ้างอิงการหาจุดกึ่งกลางของถนน โดยเส้นสีแดง ทั้งหมด 5 เส้น จะกำหนดค่าแถวดังนี้ แถวของภาพที่ $y[0] = 230$, $y[1] = 200$, $y[2] = 170$, $y[3] = 140$ และ $y[4] = 110$

เส้นสีน้ำเงิน เป็นการหาจุดกึ่งกลางของถนน โดยจะหาจุดกึ่งกลางบนระนาบเส้นสีแดง ซึ่งคำนวณโดยการหาค่าคอดัชนีที่พิกเซลแรกที่เป็นสีขาวนับจากทางขวาคือค่า max และหาค่าคอดัชนีที่พิกเซลแรกที่เป็นสีขาวนับจากทางซ้ายคือค่า min ซึ่งจุดกึ่งกลางที่ได้เกิดจากการนำ $max - min$

เส้นสีเหลือง จะสามารถบอกได้ว่าจะสั่งให้รถเลี้ยวไปกี่มุมกี่องศา โดยคิดจากการหาค่าความน่าจะเป็นที่รถจะเลี้ยวไปซึ่งนำค่าจุดกึ่งกลางที่คำนวณได้แต่ละจุดมาทำการถ่วงน้ำหนัก จะทำให้ได้จุดอ้างอิงมาหนึ่งจุดเพื่อใช้ในการวาดเส้นสีเหลือง หาได้ดังนี้

ให้ $pointReference$ คือ จุดอ้างอิง และ med คือ จุดกึ่งกลางบนระนาบเส้นสีแดง

$$pointReference = \frac{(med(0) \times 0.8 + med(1) \times 0.6 + med(2) \times 0.4 + med(3) \times 0.2 + med(4) \times 0.1)}{(0.8 + 0.6 + 0.4 + 0.2 + 0.1)}$$

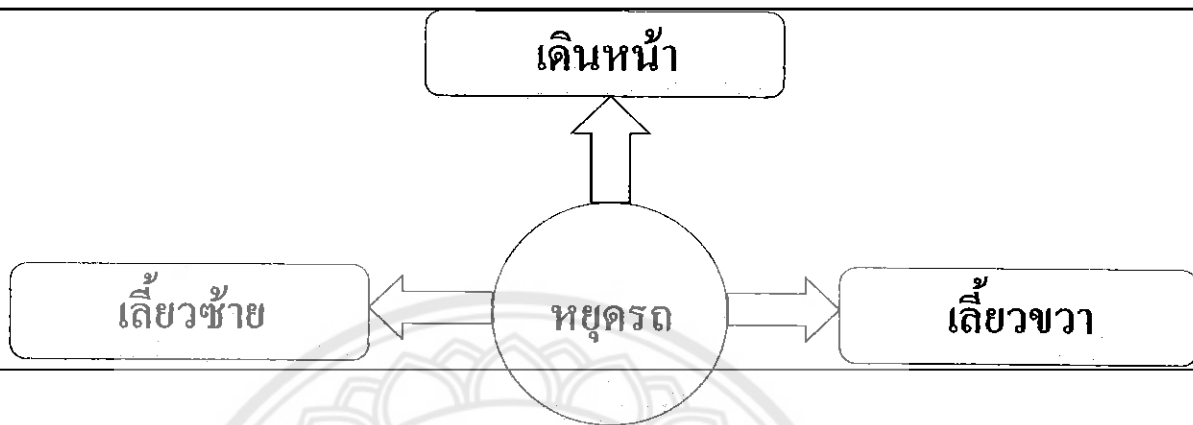
การหามุมของเส้นสี่เหลี่ยมที่ทำมุมกับแนวแกน x จะนำค่า pointReference ที่ได้จากการหาจุดอ้างอิงมาคิดด้วย โดยฟังก์ชันในการหามุมมีรายละเอียดดังนี้

ค่า x1 = ความกว้างของภาพหารด้วย 2 , y1 = ความสูงของภาพ และกำหนดค่าเป็น 0

ค่า x2 = pointReference , y1 = ความสูงของภาพ - y[4]

```
// @param Xpos1_      A position of x1
// @param Ypos1_      A position of y1
// @param Xpos2_      A position of x2
// @param Ypos2_      A position of y2
// @return angle between (x1,y1) and (x2,y2)
// @description get angle between two position
// @contract getAngleObj : int,int,int,int -> double
// @example getAngleObj(0,10,10,5) == 1
// @example getAngleObj(0,10,20,45) == 2
double getAngleObj(int Xpos1_,int Ypos1_,int Xpos2_,int Ypos2_)
{
    double arctan_T_ = 0.0;
    int delX_ = 0;
    int delY_ = 0;
    delX_ = Xpos2_ - Xpos1_;
    delY_ = Ypos2_ - Ypos1_;
    if( ( delY_ > 0 ) && ( delX_ > 0 ) ){
        arctan_T_ = ((atan(double(delY_)/double(delX_))*
180.0 * 7.0)/22.0); // atan = arctan
    }
    else if( ( delY_ > 0 ) && ( delX_ < 0 ) ){
        arctan_T_ = (180) +
((atan(double(delY_)/double(delX_))* 180.0 * 7.0)/22.0);
    }
    else if( ( delY_ < 0 ) && ( delX_ < 0 ) ){
        arctan_T_ = (180) +
((atan(double(delY_)/double(delX_))* 180.0 * 7.0)/22.0);
    }
    else if( ( delY_ < 0 ) && ( delX_ > 0 ) ){
        arctan_T_ = (360) +
((atan(double(delY_)/double(delX_))* 180.0 * 7.0)/22.0);
    }
    else if( (delX_ == 0) && ( delY_ > 0 ) ){
        arctan_T_ = (90.0);
    }
    else if( ( delX_ == 0 ) && ( delY_ < 0 ) ){
        arctan_T_ = (270.0);
    }
    else if( ( delY_ == 0 ) && ( delX_ > 0 ) ){
        arctan_T_ = 0;
    }
    else if( ( delY_ == 0 ) && ( delX_ < 0 ) ){
        arctan_T_ = (180.0);
    }
};
if(arctan_T_ >= 360){
    arctan_T_ = 360 - arctan_T_;
};
return arctan_T_;
}
```

3.2.11 ระบบภายในขั้นที่ 11 การสั่งการทำงานผ่านสายอนุกรม(serial) เพื่อควบคุม ส่วนควบคุมระดับล่าง โดยมีคำสั่งการทำงานคือ เดินหน้า , ถอยหลัง , เลี้ยวซ้าย , เลี้ยวขวา และหยุด รถ ซึ่งแสดงดังรูปที่ 3.13



รูปที่ 3.13 แสดงการสั่งการทำงาน

จากที่ได้ทำการวิเคราะห์ ออกแบบ และพัฒนาระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการ ประมวลผลภาพ โดยสามารถดูผลลัพธ์ของการพัฒนาระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการ ประมวลผลภาพ ได้ในบทถัดไป

บทที่ 4

ผลการทดลอง

จากที่ได้ทำการศึกษากระบวนการทำงานของระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการประมวลผลภาพ ซึ่งบทนี้จะกล่าวถึงผลการประมวลผลภาพของแต่ละขั้นตอนที่ได้ออกแบบไว้ในบทที่ 3 ซึ่งสามารถสรุปผลได้ดังต่อไปนี้

4.1 ผลการทดสอบการติดต่อกล้อ

การติดต่อกล้อจะทำการติดต่อผ่าน ไดรารี ของ OpenCV โดยใช้ภาษา C++ โดยภาพที่ได้จะเป็นรูปแบบของไฟล์ภาพวิดีโอ แล้วทำการ Capture ภาพออกมาเป็นภาพนิ่งทีละภาพเพื่อใช้ในการประมวลผล

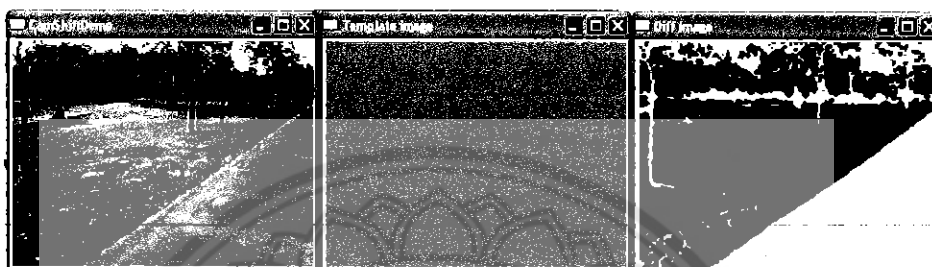


รูปที่ 4.1 แสดงผลการติดต่อกล้อ

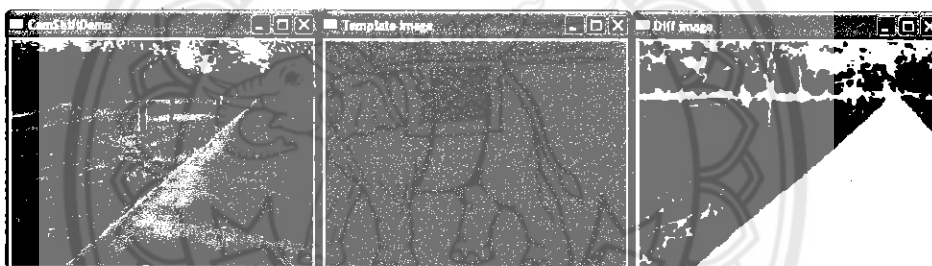
4.2 ผลการประมวลผลภาพถนนโดยไม่มีสิ่งกีดขวาง

4.2.1 ผลของการทำ Image segmentation

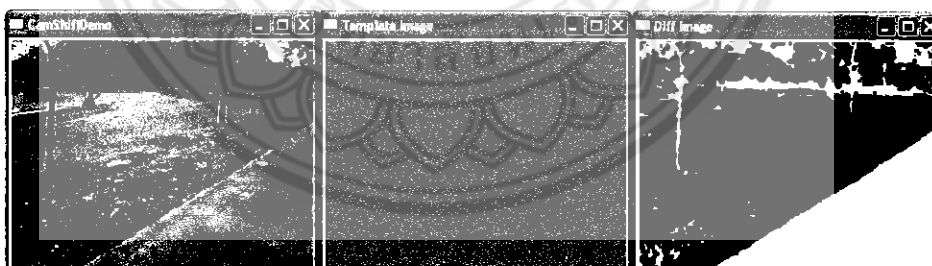
การทำ Image segmentation จะทำการนำภาพต้นฉบับ มาทำการลบด้วยภาพร่าง(ภาพสี - ถนน) แล้วนำมาทำกระบวนการ Threshold แบบ Binary inverse เพื่อทำให้เห็นขอบชัดเจนมากยิ่งขึ้น โดยผลลัพธ์ที่ได้เป็นไปตามรูปซึ่งมีหลายกรณี ดังต่อไปนี้



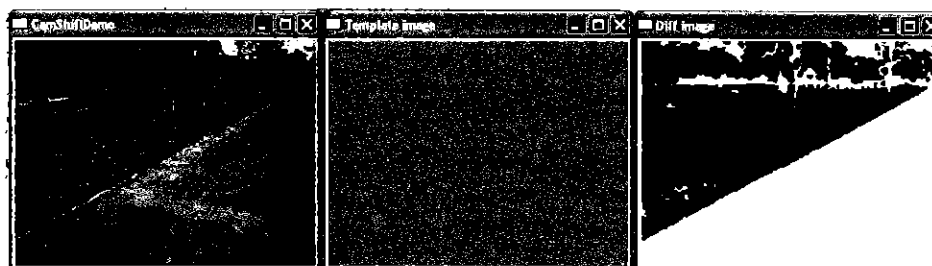
รูปที่ 4.2 ก



รูปที่ 4.2 ข

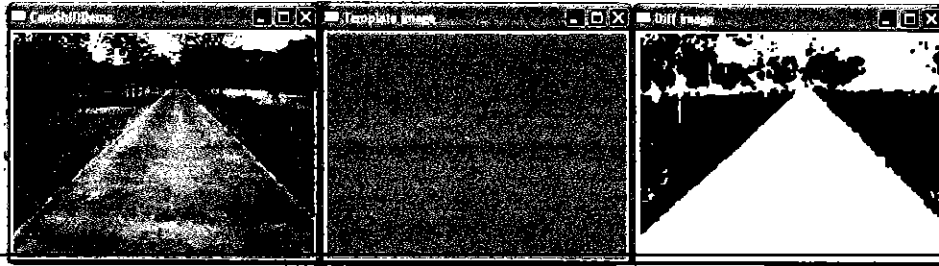


รูปที่ 4.2 ค

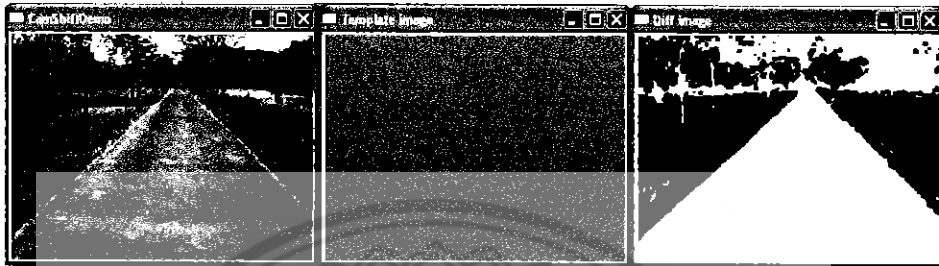


รูปที่ 4.2 ง

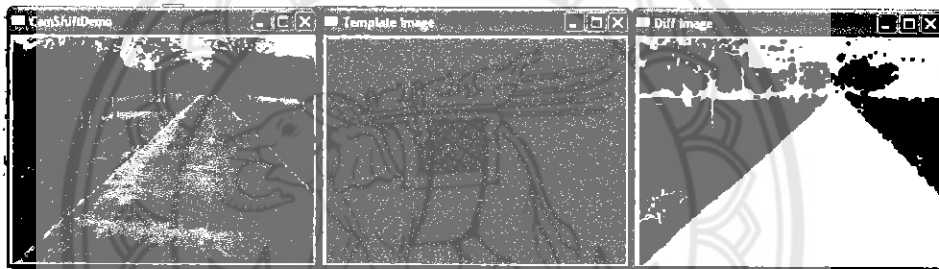
รูปที่ 4.2 ก,ข,ค และ ง เป็นภาพแสดงการทำ Image segmentation ในกรณีที่ถนนอยู่ทางขวา



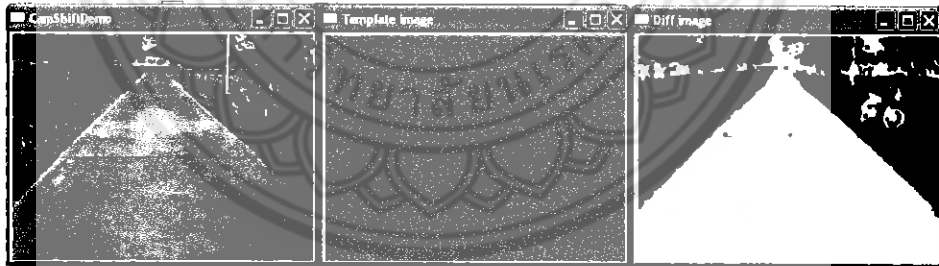
รูปที่ 4.3 ก



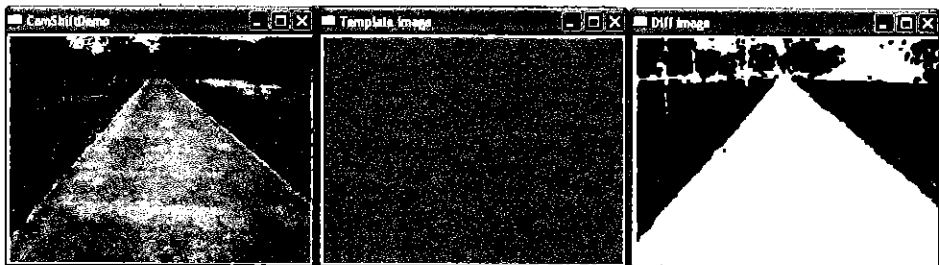
รูปที่ 4.3 ข



รูปที่ 4.3 ค

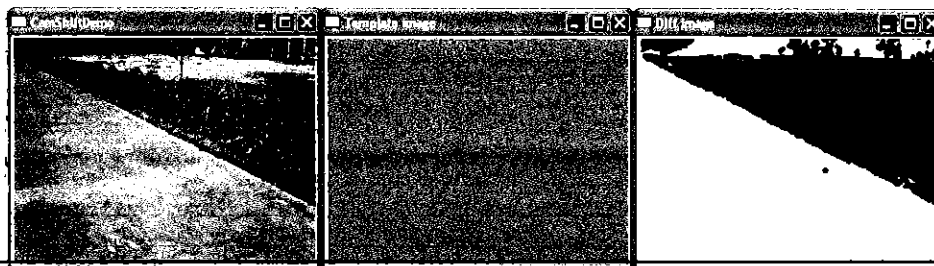


รูปที่ 4.3 ง

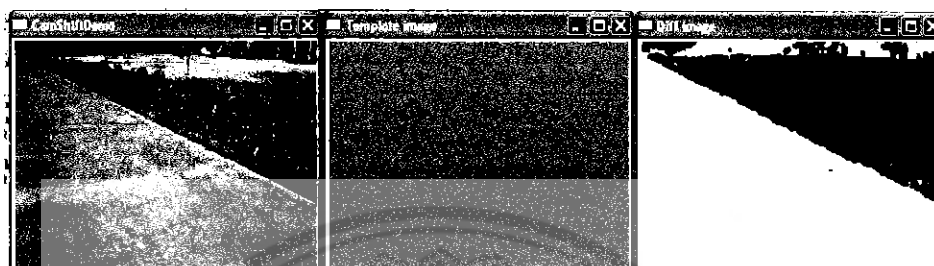


รูปที่ 4.3 จ

รูปที่ 4.3 ก,ข,ค,ง และ จ เป็นภาพแสดงการทำ Image segmentation ในกรณีที่ดินนอยู่ตรงกลาง



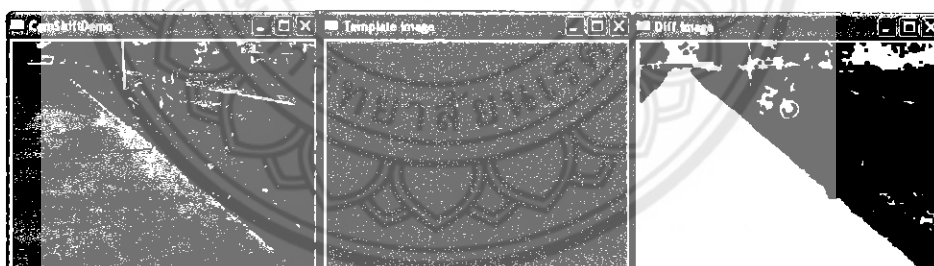
รูปที่ 4.4 ก



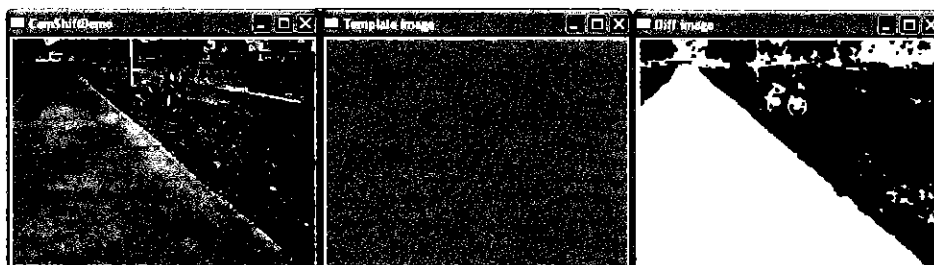
รูปที่ 4.4 ข



รูปที่ 4.4 ค



รูปที่ 4.4 ง



รูปที่ 4.4 จ

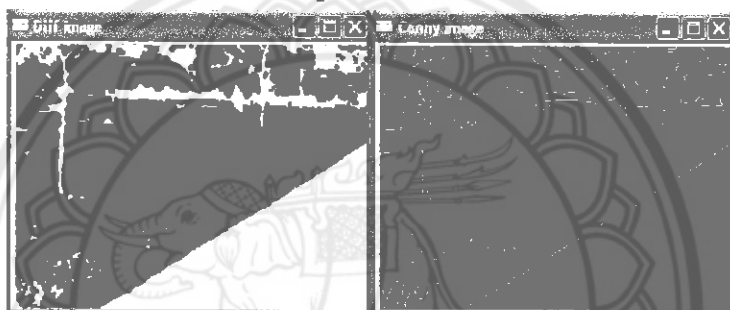
รูปที่ 4.4 ก,ข,ค,ง และ จ เป็นภาพแสดงการทำ Image segmentation ในกรณีที่ถนนอยู่ทางซ้าย

4.2.2 ผลของการทำ Canny Edge Detection

การทำ Canny edge detection จะทำการนำภาพที่ได้จากกระบวนการทำ Image segmentation มาหาขอบของภาพ โดยผลลัพธ์ที่ได้เป็นไปตามรูปซึ่งมีหลายกรณี ดังต่อไปนี้



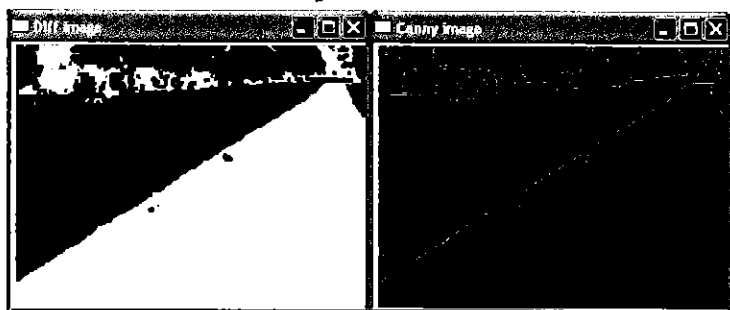
รูปที่ 4.5 ก



รูปที่ 4.5 ข

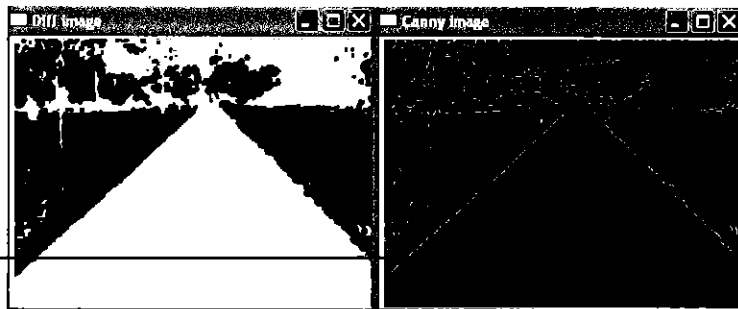


รูปที่ 4.5 ค

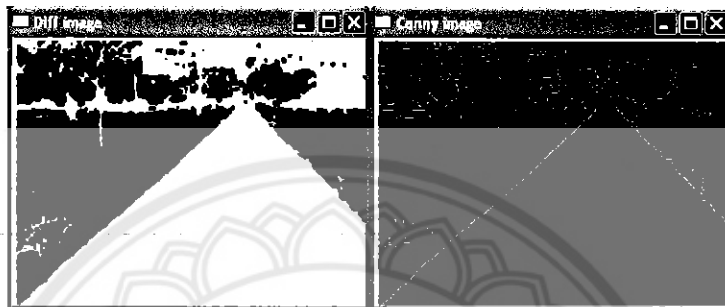


รูปที่ 4.5 ง

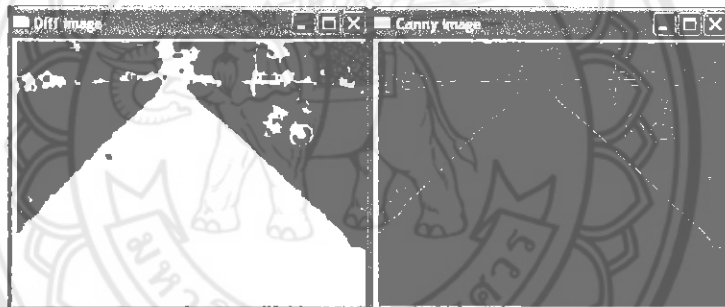
รูปที่ 4.5 ก, ข, ค และ ง เป็นภาพแสดงการทำ Canny edge detection ในกรณีที่ถนนอยู่ทางขวา



รูปที่ 4.6 ก



รูปที่ 4.6 ข



รูปที่ 4.6 ค



รูปที่ 4.6 ง

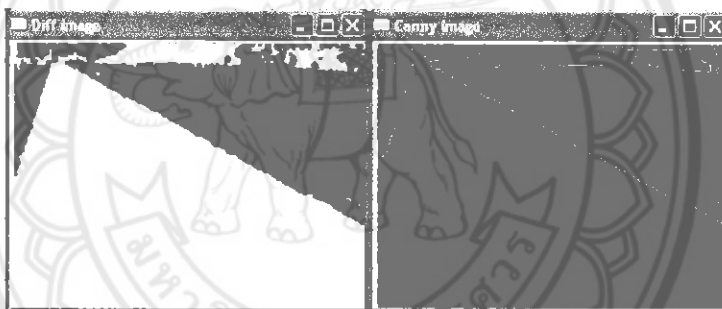
รูปที่ 4.6 ก, ข, ค และ ง เป็นภาพแสดงการทำ Canny edge detection ในกรณีที่ถนนอยู่ตรงกลาง



รูปที่ 4.7 ก



รูปที่ 4.7 ข



รูปที่ 4.7 ค

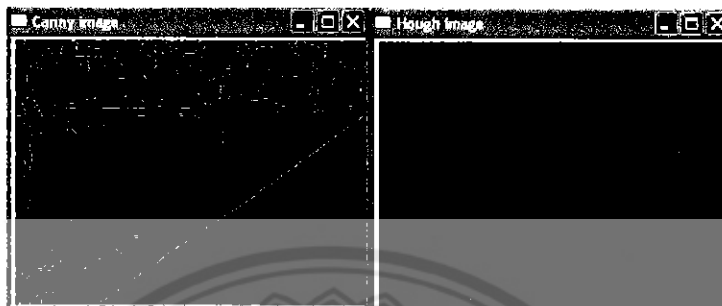


รูปที่ 4.7 ง

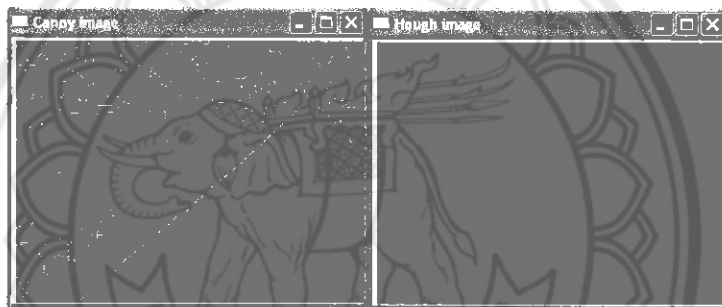
รูปที่ 4.7 ก, ข, ค และ ง เป็นภาพแสดงการทำ Canny edge detection ในกรณีที่ถนนอยู่ทางซ้าย

4.2.3 ผลของการทำ Hough Transform

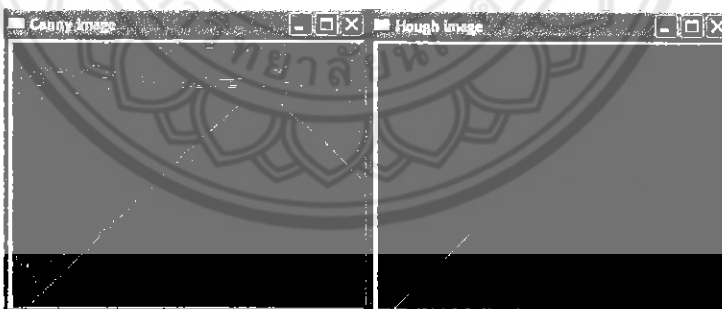
การทำ Hough transform จะทำการนำภาพที่ได้จากกระบวนการทำ Canny edge detection มาทำการลากเส้นตรง ซึ่งเส้นตรงจะได้มาจากจุดที่อยู่ระนาบเดียวกันบนขอบที่เกิดจากการทำ Canny edge detection โดยผลลัพธ์ที่ได้เป็นไปตามรูปซึ่งมีหลายกรณี ดังต่อไปนี้



รูปที่ 4.8 ก

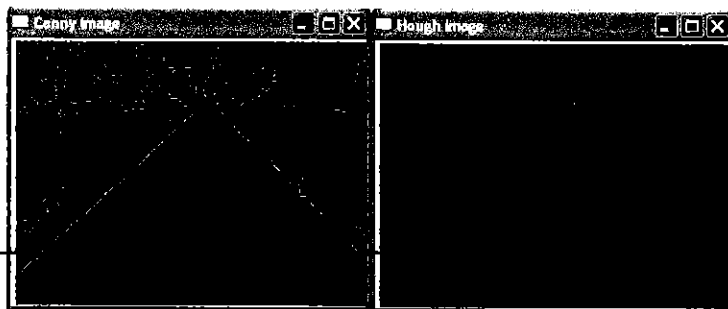


รูปที่ 4.8 ข

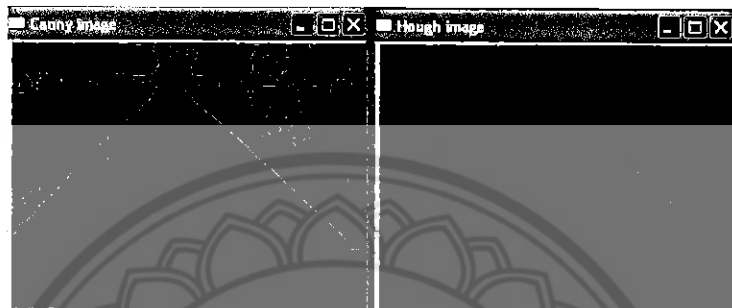


รูปที่ 4.8 ค

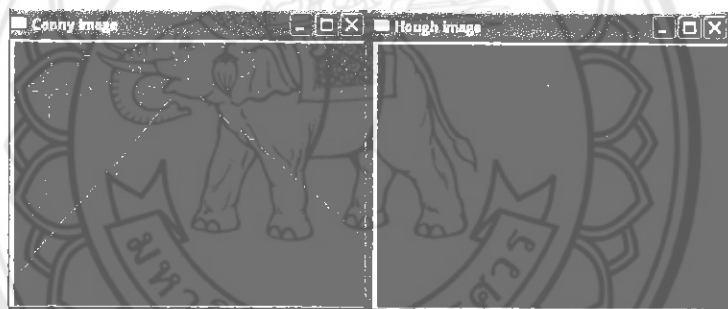
รูปที่ 4.8 ก, ข และ ค เป็นภาพแสดงการทำ Hough transform ในกรณีที่ดินนอยู่ทางขวา



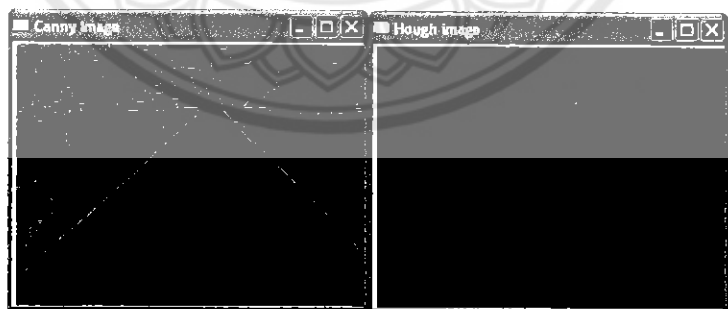
รูปที่ 4.9 ก



รูปที่ 4.9 ข

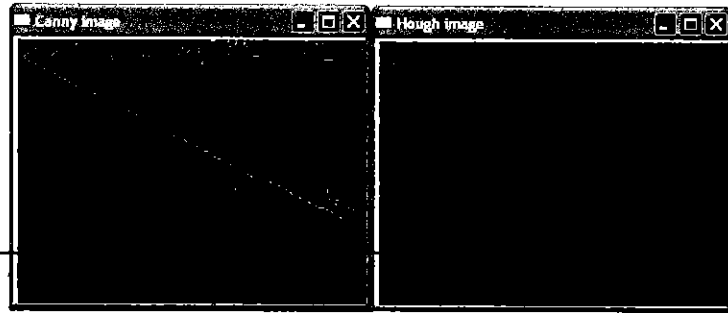


รูปที่ 4.9 ค



รูปที่ 4.9 ง

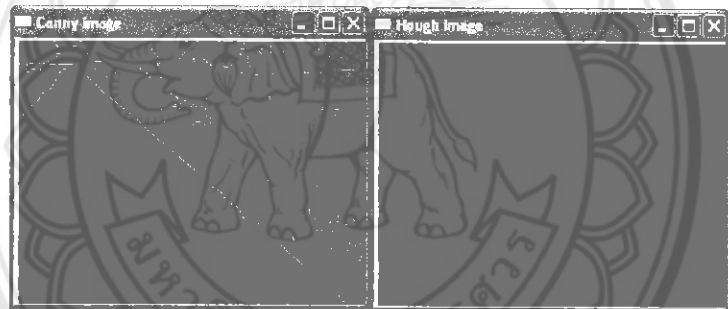
รูปที่ 4.9 ก, ข, ค และ ง เป็นภาพแสดงการทำ Hough transform ในกรณีที่ถนนอยู่ตรงกลาง



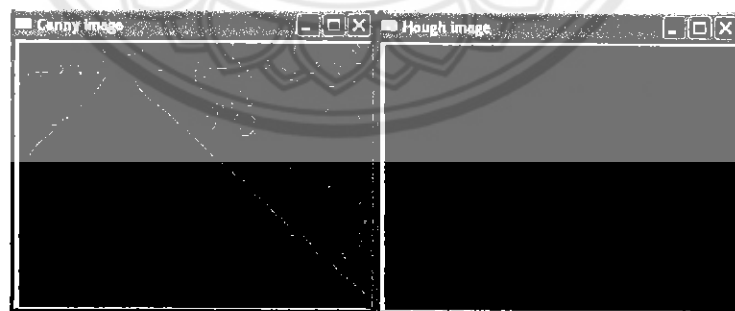
รูปที่ 4.10 ก



รูปที่ 4.10 ข



รูปที่ 4.10 ค

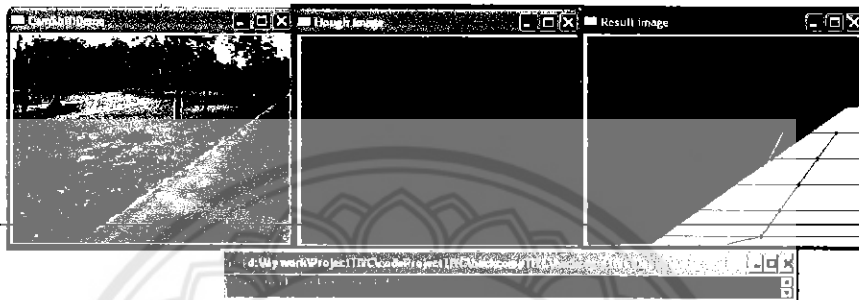


รูปที่ 4.10 ง

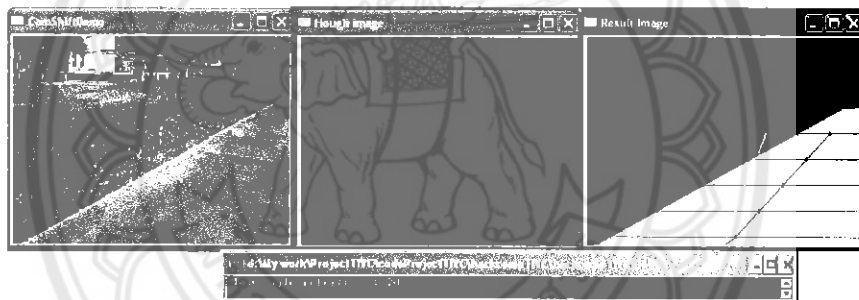
รูปที่ 4.10 ก, ข, ค และ ง เป็นภาพแสดงการทำ Hough transform ในกรณีที่ดินอยู่ทางซ้าย

4.2.4 ผลของการเปรียบเทียบเส้นเสมือน(Visual line) หรือการถอดความหมายจากภาพ

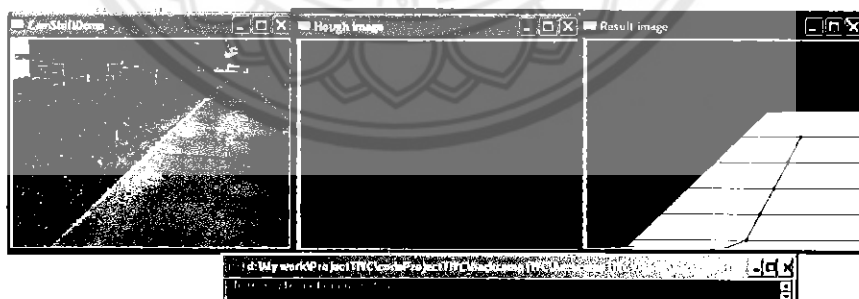
การทำเส้นเสมือน(Visual line) จะทำการนำภาพที่ได้จากกระบวนการทำ Hough transform มาทำการถมถนนเพื่อให้รู้ขอบเขตของถนน แล้วทำการหาจุดกึ่งกลางของถนน และมุมมองเสาที่รถจะต้องไป จะดูจากเส้นสีเหลืองโดยทำมุมกับแนวแกน x โดยผลลัพธ์ที่ได้เป็นไปตามรูปซึ่งมีหลายกรณี ดังต่อไปนี้



รูปที่ 4.11 ก

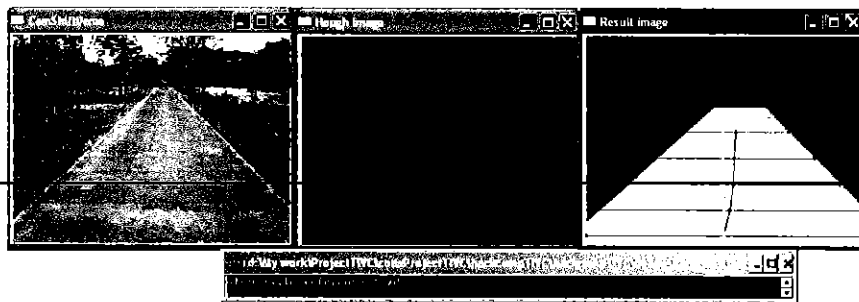


รูปที่ 4.11 ข



รูปที่ 4.11 ค

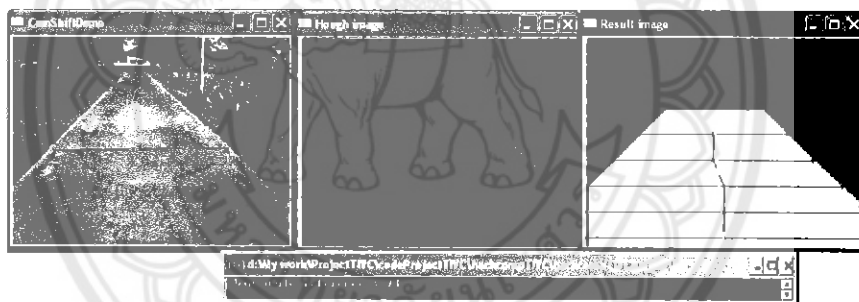
รูปที่ 4.11 ก, ข และ ค เป็นภาพแสดงการทำเส้นเสมือน ในกรณีที่ถนนอยู่ทางขวา โดยรูปที่ 4.11 ก จะแสดงมุม 63 องศา กับแนวแกน x , รูปที่ 4.11 ข จะแสดงมุม 70 องศา กับแนวแกน x , รูปที่ 4.11 ค จะแสดงมุม 72 องศา กับแนวแกน x



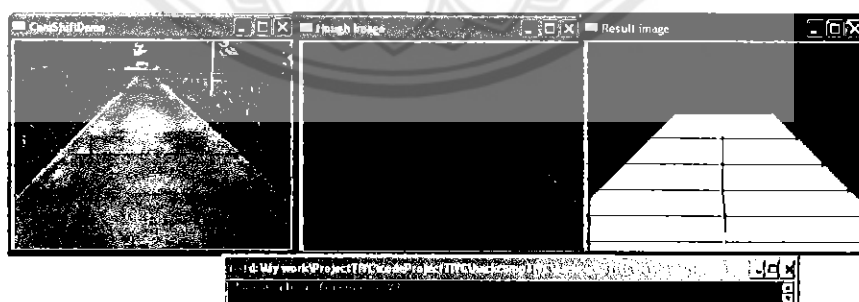
รูปที่ 4.12 ก



รูปที่ 4.12 ข

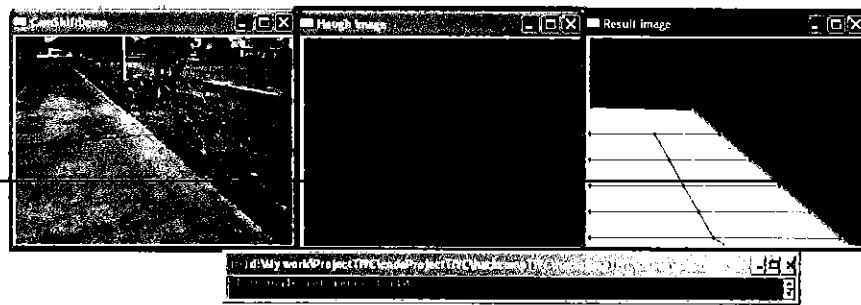


รูปที่ 4.12 ค



รูปที่ 4.12 ง

รูปที่ 4.12 ก, ข และ ค เป็นภาพแสดงการทำเส้นเสมือน ในกรณีที่ถนนอยู่ตรงกลาง โดยรูปที่ 4.12 ก จะแสดงมุม 90 องศา กับแนวแกน x , รูปที่ 4.12 ข จะแสดงมุม 90 องศา กับแนวแกน x , รูปที่ 4.12 ค จะแสดงมุม 94 องศา กับแนวแกน x , รูปที่ 4.12 ง จะแสดงมุม 93 องศา กับแนวแกน x



รูปที่ 4.13 ก



รูปที่ 4.13 ข



รูปที่ 4.13 ค

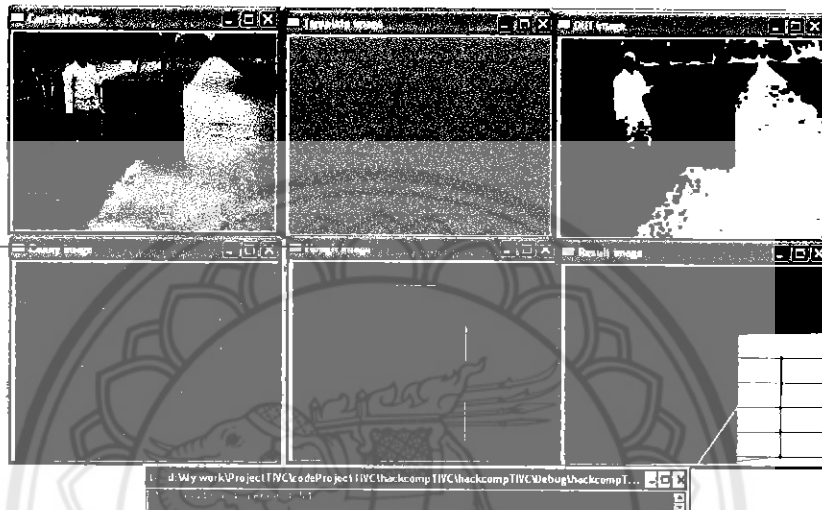


รูปที่ 4.13 ง

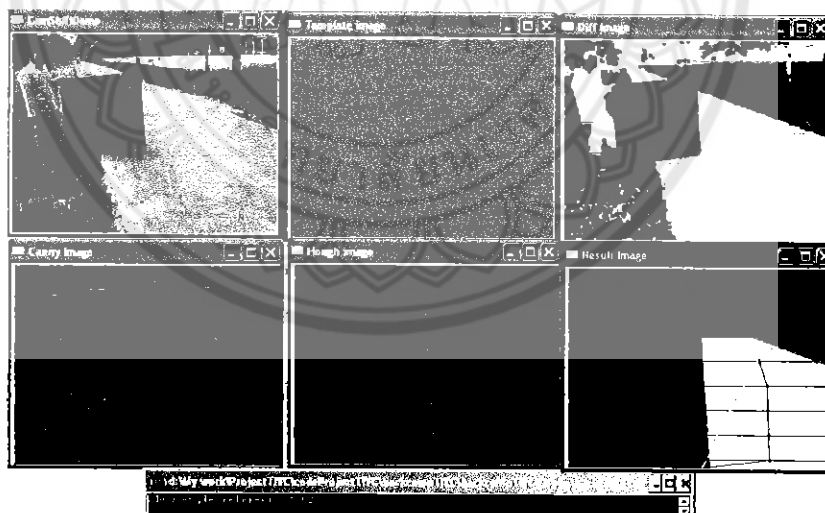
รูปที่ 4.13 ก, ข และ ค เป็นภาพแสดงการทำเส้นเสมือน ในกรณีที่ดินนอยู่ทางซ้าย โดยรูปที่ 4.13 ก จะแสดงมุม 110 องศา กับแนวแกน x , รูปที่ 4.13 ข จะแสดงมุม 105 องศา กับแนวแกน x , รูปที่ 4.12 ค จะแสดงมุม 100 องศา กับแนวแกน x , รูปที่ 4.13 ง จะแสดงมุม 99 องศา กับแนวแกน x

4.3 ผลการประมวลผลภาพถนนโดยมีสิ่งกีดขวาง

จากผลการประมวลผลภาพถนนโดยมีสิ่งกีดขวาง จะพบว่าระบบสามารถแยกแยะสิ่งกีดขวางได้ประมาณ 70 เปอร์เซ็นต์ ซึ่งถ้าดูตามผลการทดลองจะพบว่า ระบบจะพยายามจะสั่งให้เลี้ยวหนีสิ่งกีดขวาง โดยดูได้จากเส้นสีเหลือง และเส้นสีเหลืองจะทำมุมกับแนวแกน x ซึ่งผลลัพธ์ที่ได้เป็นไปตามรูปซึ่งมีหลายกรณี ดังต่อไปนี้

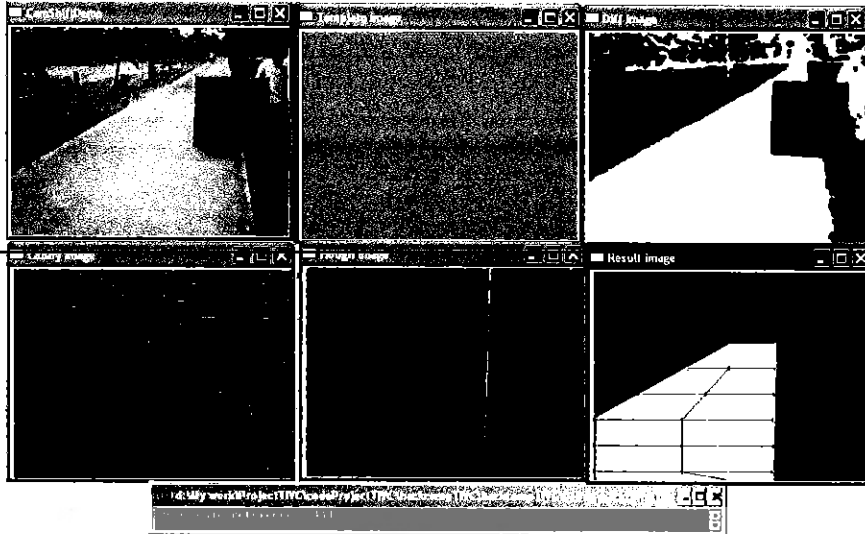


รูปที่ 4.14 ก

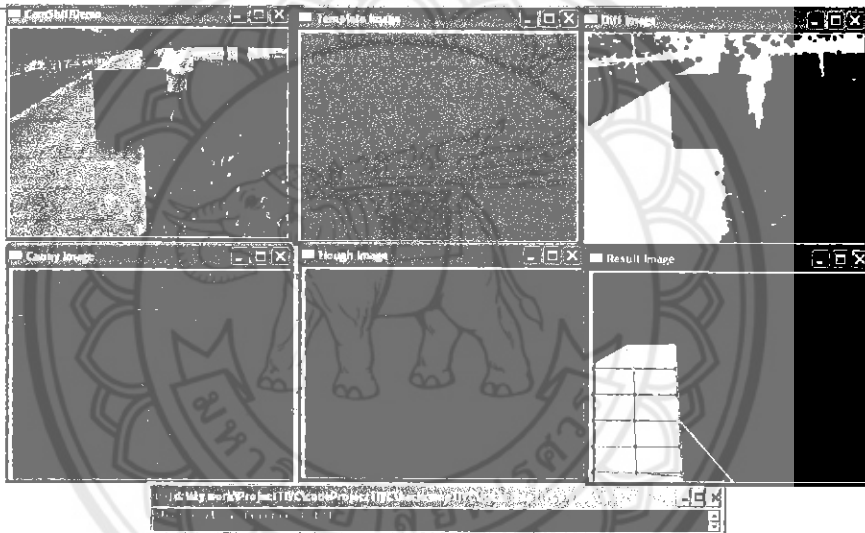


รูปที่ 4.14 ข

รูปที่ 4.14 ก และ ข เป็นภาพแสดงการประมวลผลภาพในกรณีที่มีสิ่งกีดขวางอยู่ทางซ้าย โดยรูปที่ 4.14 ก จะแสดงมุม 54 องศา กับแนวแกน x , รูปที่ 4.14 ข จะแสดงมุม 62 องศา กับแนวแกน x



รูปที่ 4.15 ข



รูปที่ 4.15 ก

รูปที่ 4.15 ก และ ข เป็นภาพแสดงการประมวลผลภาพในกรณีที่มีสิ่งกีดขวางอยู่ทางขวา โดยรูปที่ 4.15 ก จะแสดงมุม 111 องศา กับแนวแกน x , รูปที่ 4.15 ข จะแสดงมุม 131 องศา กับแนวแกน x

จากผลการทดลองที่ได้กล่าวไว้ข้างต้น ได้ยกตัวอย่างของผลการทดลองแต่ละกรณีไว้พอสังเขป ซึ่งสามารถศึกษาผลการทดลองเพิ่มเติมต่อได้ที่ภาคผนวก ข.

บทที่ 5

สรุปผล

5.1 สรุปผลการทดลอง

จากผลการทดลองระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการประมวลผลภาพ จะพบว่าระบบมีความไวต่อแสงมาก ดังนั้นจึงเปลี่ยนระบบสีจาก RGB มาใช้ระบบสี HSV ซึ่งจะสามารถช่วยตัดแสงออกไปได้ และก่อนที่จะทำกระบวนการ Canny edge detection จะต้องทำการกรองภาพก่อนเพื่อทำการลดสัญญาณรบกวนออกจากภาพ โดยในการทำกระบวนการ Canny edge detection จะต้องผ่านกระบวนการการทำ threshold เพื่อให้ขอบของภาพชัดขึ้น จึงทำให้สามารถสร้างเส้นตรงจากฟังก์ชัน Hough transform ได้แม่นยำขึ้น ซึ่งทำให้สามารถระบุถนนได้ชัดขึ้น และสามารถหามุมมองเสาที่รถจะไปได้

จากผลการประมวลผลภาพถนน โดยไม่มีสิ่งกีดขวาง ระบบสามารถระบุว่าส่วนไหนเป็นถนน และตัดสนใจในการสังเกตุ ในกรณีต่างๆ ได้

จากผลการประมวลผลภาพถนน โดยมีสิ่งกีดขวาง จะพบว่าระบบสามารถแยกแยะสิ่งกีดขวางได้ประมาณ 70 เปอร์เซ็นต์

5.2 ปัญหาและแนวทางแก้ไข

1. ปัญหาเรื่อง แสง ซึ่งระบบควบคุมรถอัจฉริยะไร้คนขับ โดยการประมวลผลภาพ จะต้องทำงานในที่โล่งแจ้ง โดยแสงจะสะท้อนเข้ากล้องมากเกินไป ทำให้ภาพที่ได้ไม่สามารถระบุสีที่แน่นอนได้ ซึ่งสามารถแก้ไขได้โดย เปลี่ยนไปใช้ระบบสี HSV โดยนำภาพส่วนของ Hue มาใช้ เพื่อตัดแสงออก ทำให้ภาพที่ได้สีคมชัดมากขึ้น

2. สภาพพื้นผิวของถนน โดยฤดูนี้เป็นฤดูร้อน ทำให้สภาพพื้นผิวถนน และบริเวณข้างถนนแห้งแล้ง ซึ่งทำให้สีระหว่างพื้นผิวถนน กับบริเวณข้างถนน มีสีที่ใกล้เคียงกัน ซึ่งสามารถแก้ไขได้ โดยทำการปรับตั้งค่าของกล้อง โดยเพิ่มความบริสุทธิ์ของสีให้มากขึ้น

5.3 แนวทางการพัฒนาต่อในอนาคต

1. ทำการปรับปรุงการประมวลผลภาพ ให้สามารถแยกสิ่งกีดขวางได้แม่นยำมากขึ้น
2. ทำการปรับปรุงการประมวลผลภาพ ให้สามารถจับสัญญาณไฟจราจรได้
3. ทำการปรับปรุงการประมวลผลภาพ ให้สามารถจับป้ายจราจรได้
4. ทำการปรับปรุงการประมวลผลภาพ ให้สามารถใช้กล้อง 2 ตัวได้ เพื่อเพิ่มมุมมองของถนนให้กว้างขึ้น



เอกสารอ้างอิง

[1] Linda G. Shapiro and George C. Stockman. **COMPUTER VISION**. New Jersey : Prentice-Hall, Inc. 2001.

[2] Rafael C. Gonzalez and Richard E. Woods. **Digital Image Processing**, Second Edition, New Jersey : Prentice-Hall, Inc.2002.

[3] William K. Pratt. **Digital Image Processing**. Third Edition, Reading : John Wiley & Sons, Inc.2001.

[4] มหาวิทยาลัยศิลปากร. “ทฤษฎีการประมวลผลภาพ.” [Online]. Available:

http://www.thapra.lib.su.ac.th/objects/thesis/fulltext/snamcn/Siriporn_Boonplianpol/Chapter2.pdf

[5] Parkzz. “ระบบสี RGB.” [Online]. Available:

<http://web.ofebia.com/Contents/view/50.htm>

[6] Computer Engineering Department, KMUTT. “ระบบสี RGB.” [Online]. Available:

<http://cpe.kmutt.ac.th/previousproject/2005/6/theory-color.html>

[7] ดร.สมยศ เกียรติวนิชวิไล. “การใช้กล้องและการจัดการภาพ.” [Slide]. KMITL:2009

http://tivc.ait.ac.th/info_arch?action=download&fileid=544

[8] บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร. “การหาขอบของเม็ดยา.” [Online]. Available:

http://www.thapra.lib.su.ac.th/objects/thesis/fulltext/snamcn/Jaravee_Chantasitiporn/Chapter3.pdf

[9] Intel. “OpenCV.” [Online]. Available:

file:///C:/Program%20Files/OpenCV/docs/ref/opencvref_cv.htm

[10] PIC. “Microcontroller.” [Online]. Available:

<http://www.insidedsp.com/Portals/0/articlepix/200712/MIPS%20PIC32%20Block%20Diagram.jp>

[11] ATMEL. "Microcontroller." [Online]. Available:

http://www.warf.com/imagesitem/original/2252_2509.jpg

[12] Lenovo Thinkpad. "Laptop." [Online]. Available:

http://www.yopi.co.th/image/prod_pics/1950/e/1950821.jpg

[13] Sony. "Digital Camera." [Online]. Available:

<http://lostbiro.com/blog/wp-content/uploads/2007/03/sony%20DCRHC38E%20.jpg>

[14] Byondhome. "RGB color picture." [Online]. Available:

<http://files.byondhome.com/DreamMakers/2008-12/LummoxJR-0002/colorwheel.png>

[15] Colorotate. "HSV color picture." [Online]. Available:

<http://learn.colorotate.org/color-models.html>



ภาคผนวก ก.

การใช้งาน OpenCV

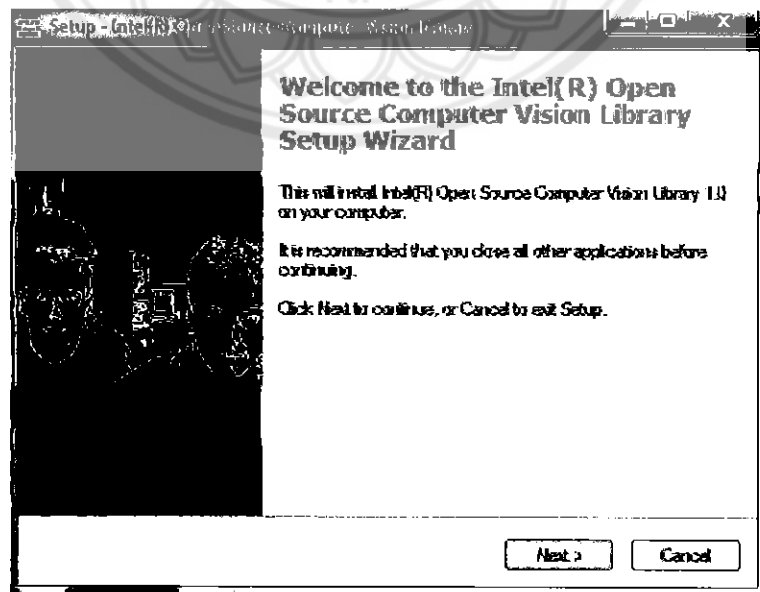
ก.1 ขั้นตอนการติดตั้ง OpenCV

1. เปิดไฟล์ OpenCV_1.0.exe เพื่อเริ่ม Setup



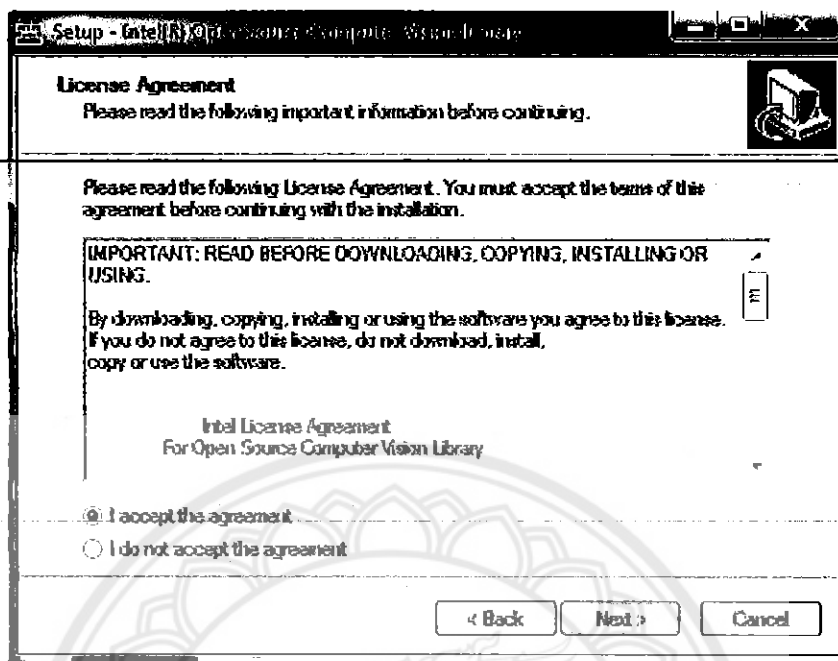
รูปที่ ก.1 แสดงตัวติดตั้ง OpenCV

2. คลิก Next



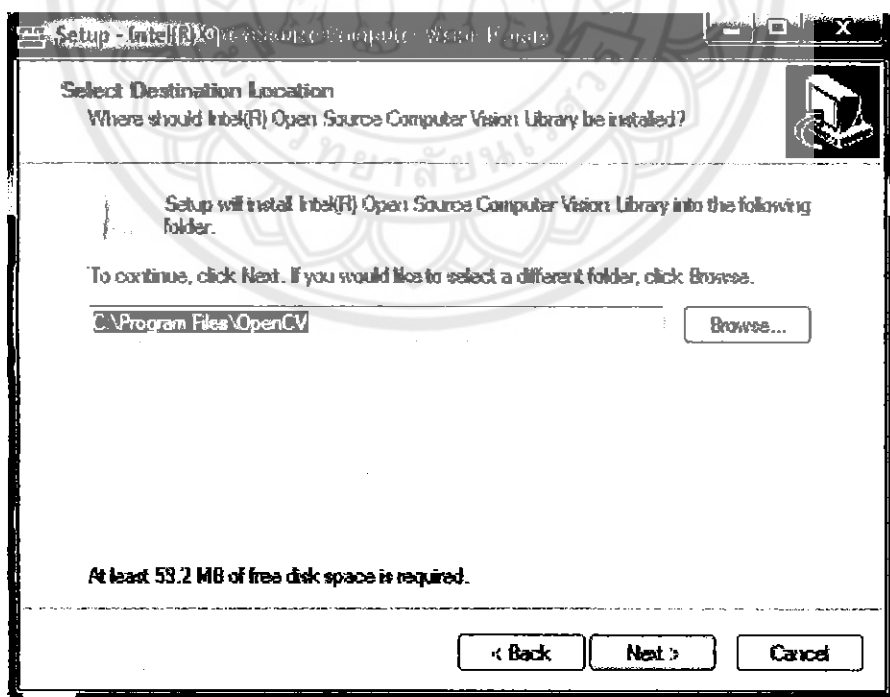
รูปที่ ก.2 แสดงหน้าต่างเริ่มต้นการติดตั้ง OpenCV

3. เลือก I accept the agreement แล้วกด Next



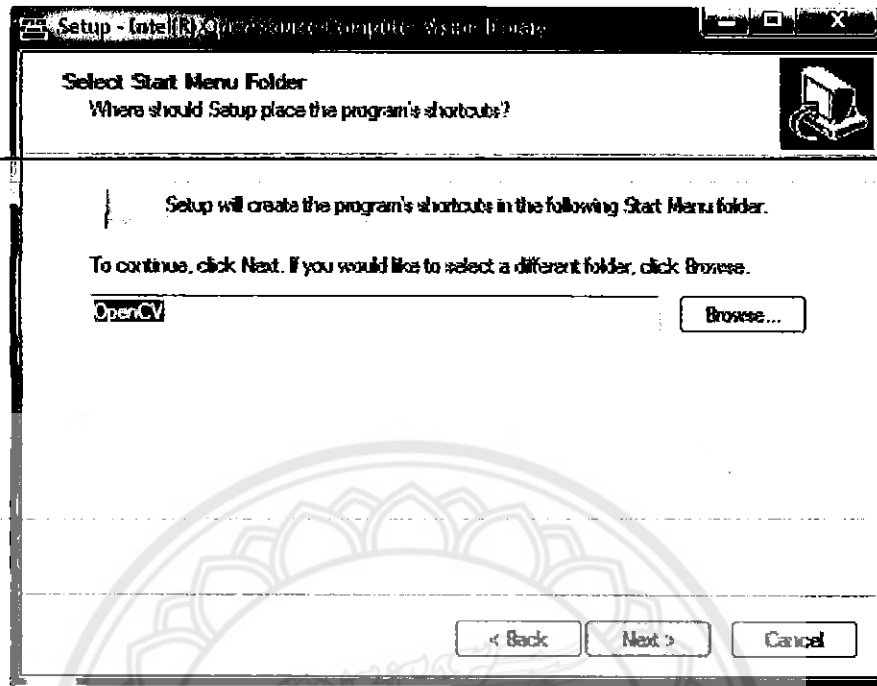
รูปที่ ก.3 แสดงหน้าต่าง License Agreement

4. กด Next



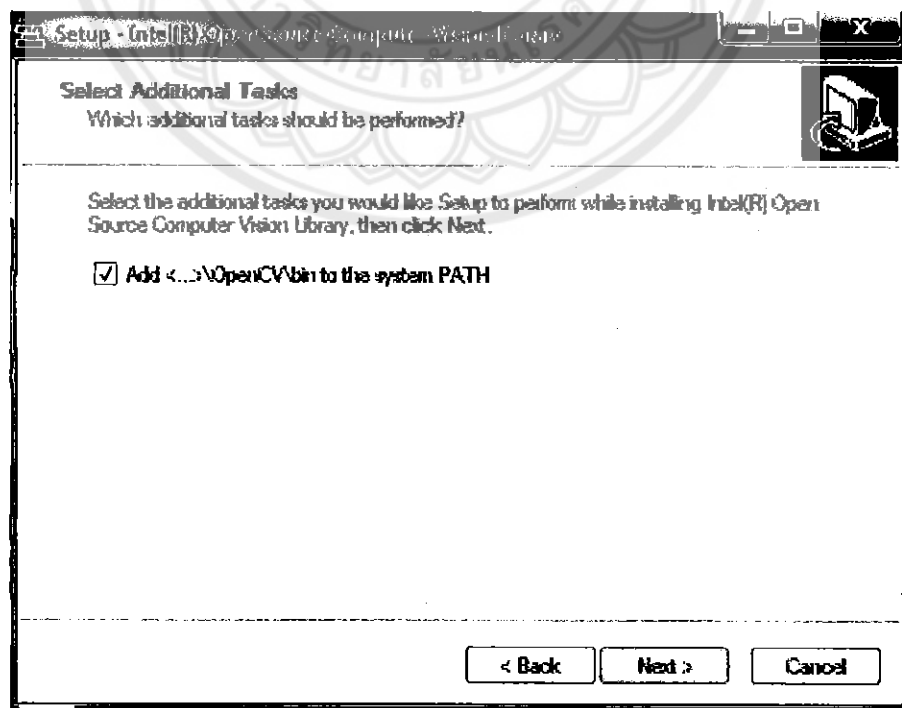
รูปที่ ก.4 แสดงหน้าต่างการ Select Destination Location

5. กด Next



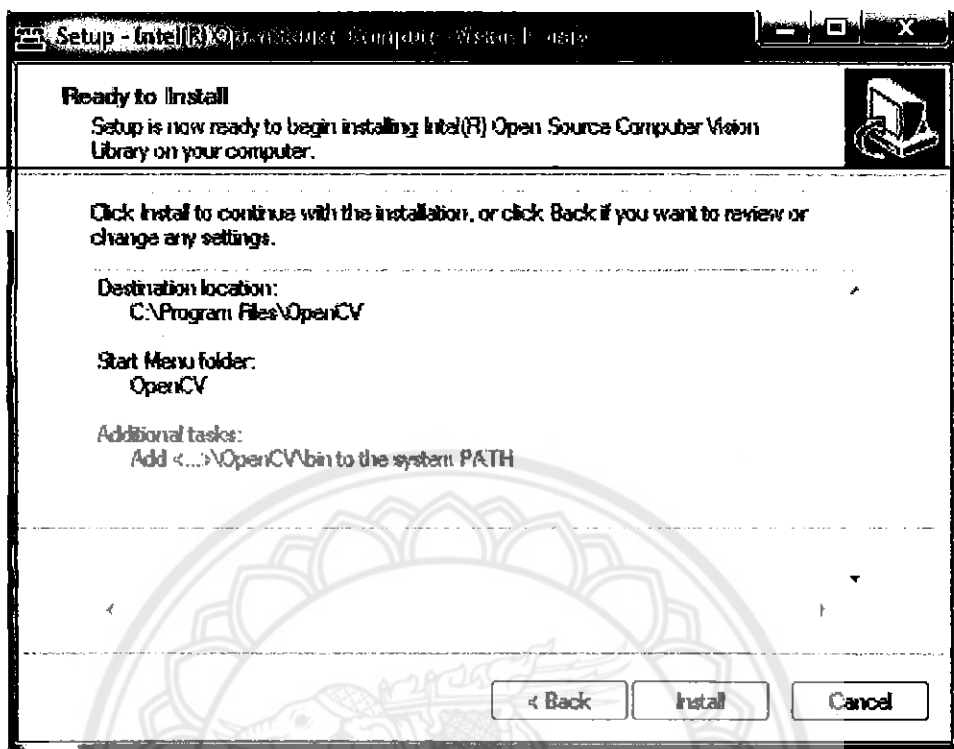
รูปที่ ก.5 แสดงหน้าต่างการ Select Start Menu Folder

6. คลิกเครื่องหมายถูกแล้วกด Next



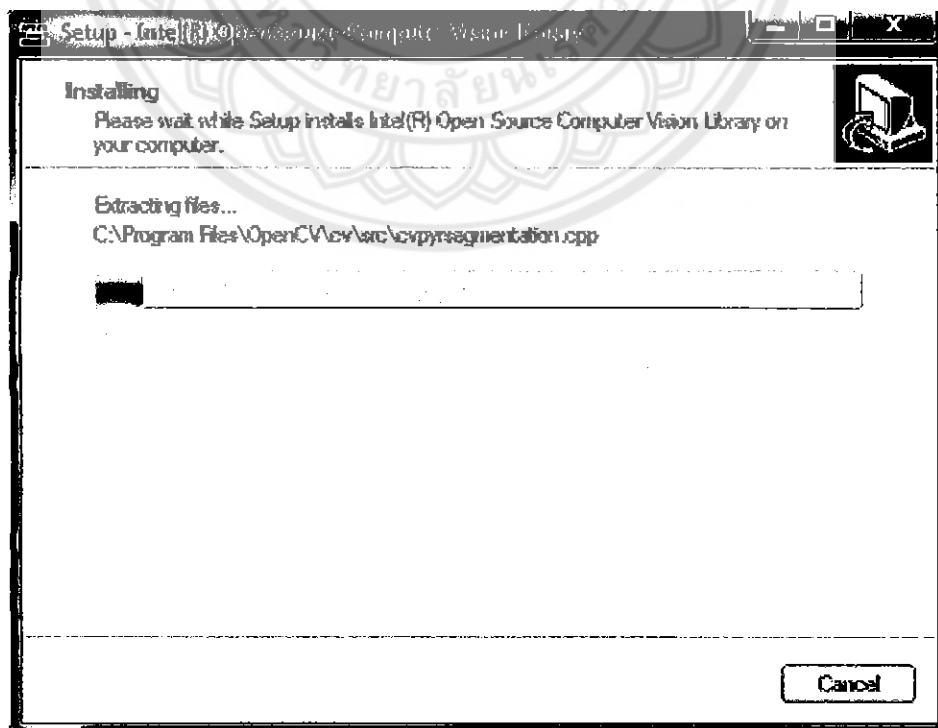
รูปที่ ก.6 แสดงหน้าต่างการ Select Additional Tasks

7. คลิก Install



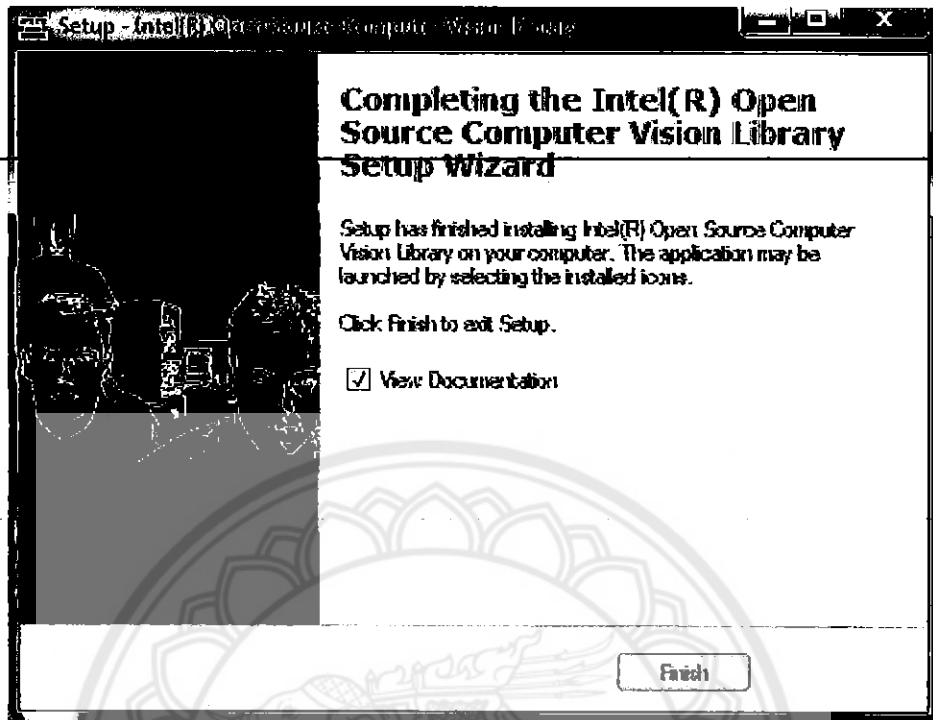
รูปที่ ก.7 แสดงหน้าต่างการ Ready to Install

8. รอนกว่าจะ Install เสร็จ



รูปที่ ก.8 แสดงหน้าต่างการ Installing

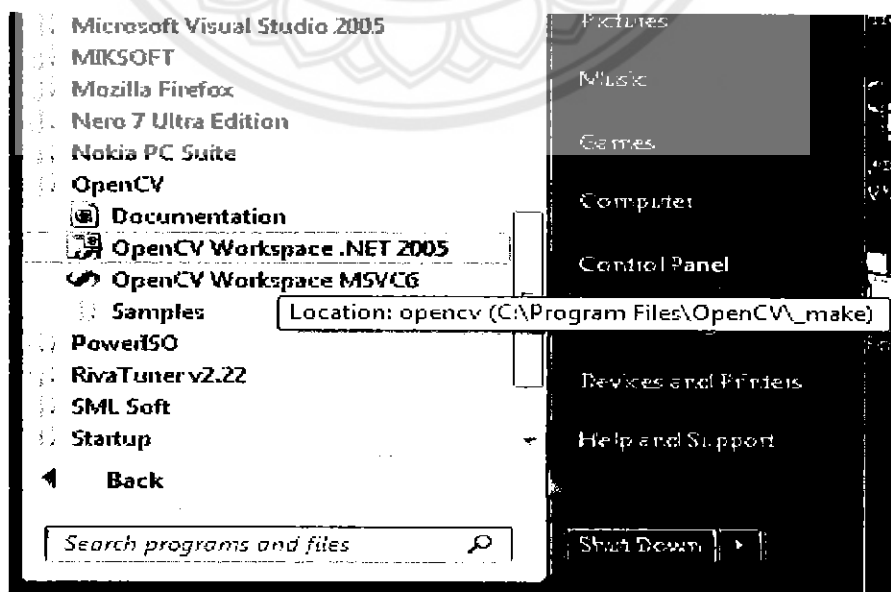
9. คลิก Finish



รูปที่ ก.9 แสดงหน้าต่างการ Install Complete

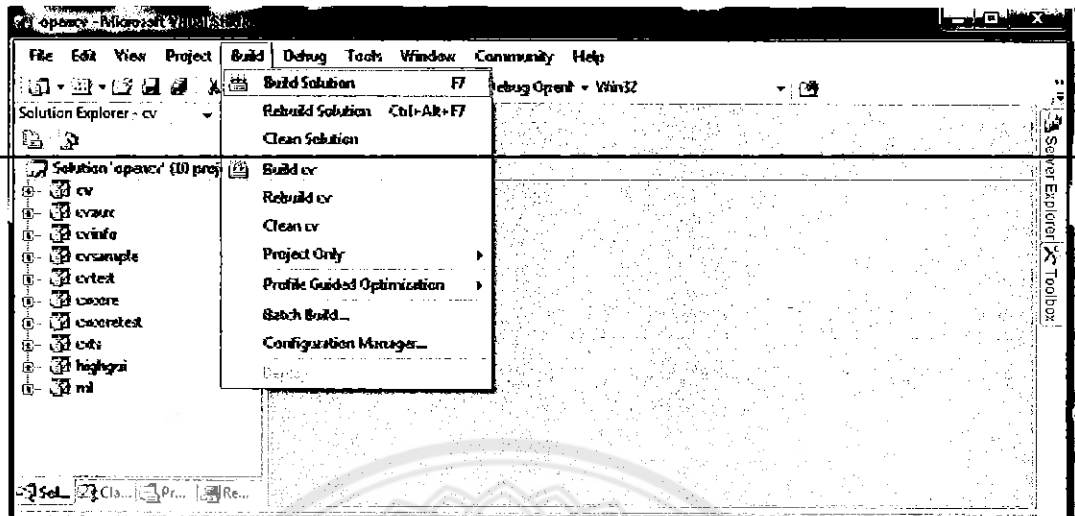
ก.2 การติดตั้ง OpenCV เสร็จแล้ว ต่อไปเป็นการ build

1. เปิด OpenCV Workspace.Net 2005



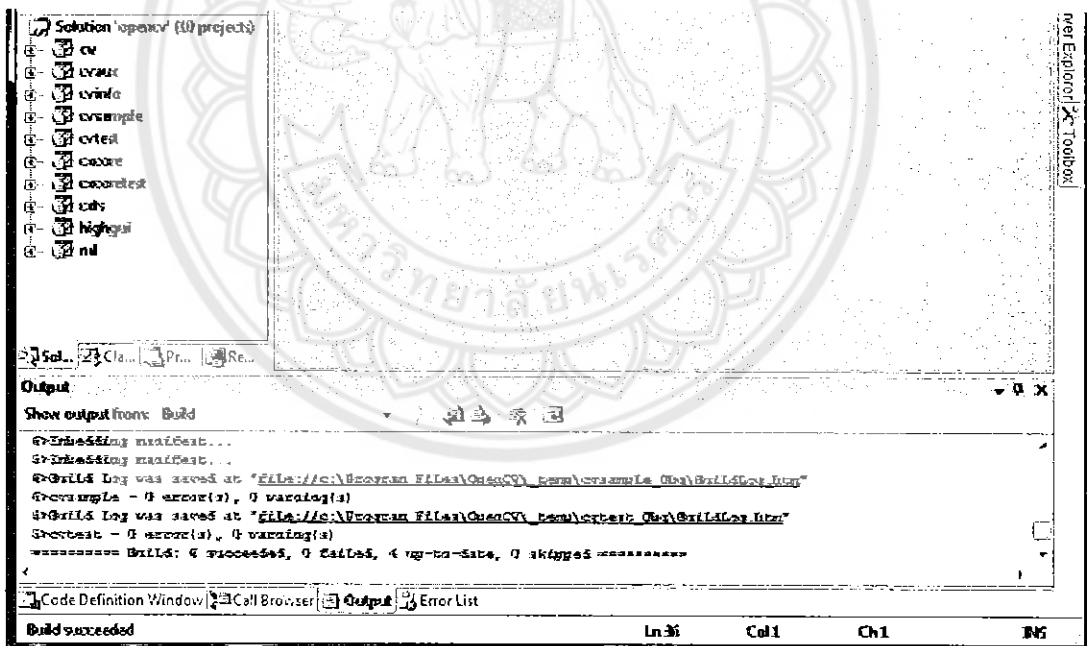
รูปที่ ก.10 แสดงเปิด OpenCV Workspace.NET 2005

2. กด Build Solution หรือ กด F7



รูปที่ ก.11 แสดงการทำ Build Solution

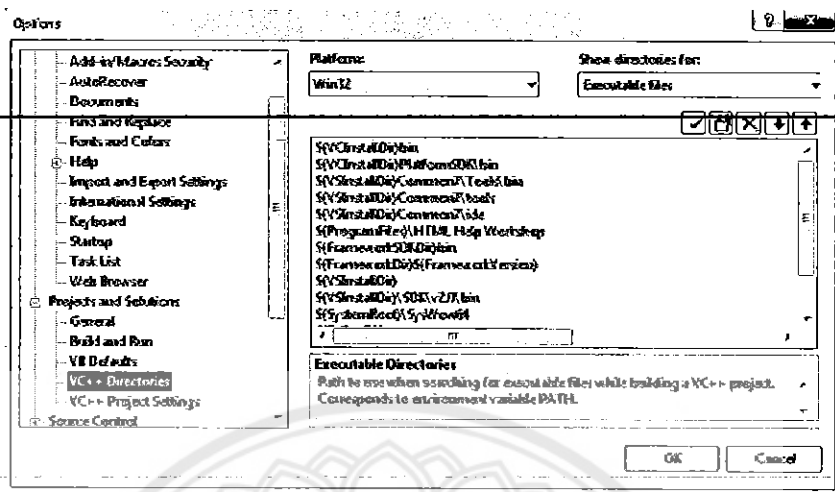
3. รอกจน build เสร็จ



รูปที่ ก.12 แสดงหน้าต่าง โปรแกรมกำลังทำการ Build

4. ต่อไปเป็นการกำหนด VC++ Directories

เข้าไปที่ Tool->Option->Project and Solutions -> VC++ Directories



รูปที่ ก.13 แสดงหน้าต่างการ Set Path ต่างๆ

add directories ตามนี้

Executable files

C:\Program Files\OpenCV\bin

Include files

C:\Program Files\OpenCV\cv\include

C:\Program Files\OpenCV\cvaux\include

C:\Program Files\OpenCV\otherlibs\cvcam\include

C:\Program Files\OpenCV\otherlibs\highgui

C:\Program Files\OpenCV\cxcore\include

Libraly file

C:\Program Files\OpenCV\lib

C:\Program Files\OpenCV\otherlibs

C:\Program Files\OpenCV\otherlibs\cvcam

C:\Program Files\OpenCV\otherlibs\highgui

C:\Program Files\OpenCV\cxcore\includ

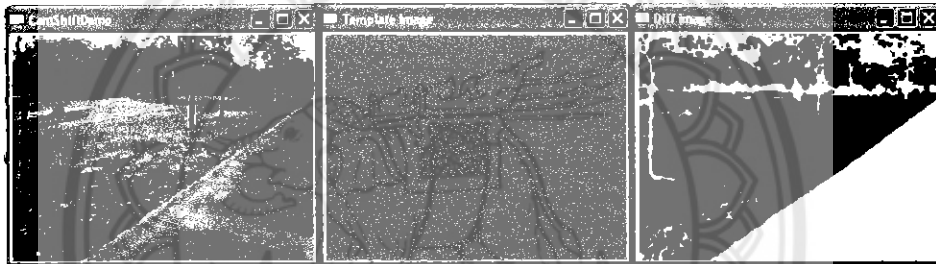
จบการ Install Open CV

ภาคผนวก ข.

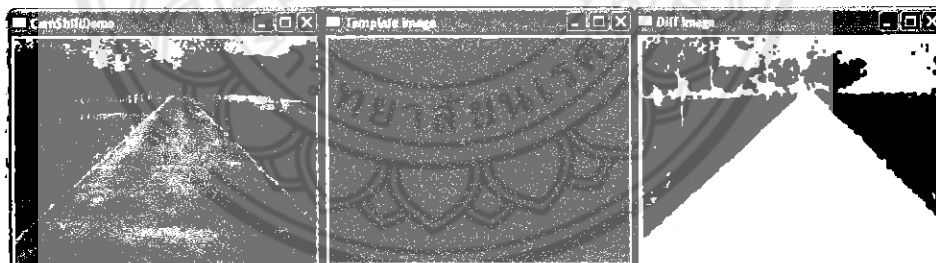
การใช้งาน OpenCV

ข.1 ผลของการทำ Image segmentation

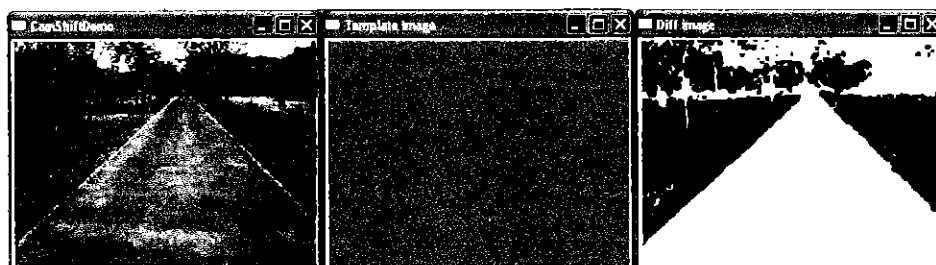
ผลของการทำ Image segmentation ในกรณีที่ถนนอยู่ทางซ้าย ตรงกลาง และทางขวา ซึ่งเป็นการทดลองที่ได้ทำไว้เพิ่มเติม เพื่อให้รู้ว่าการทำ Image segmentation สามารถทำได้จริง โดยผลการทดลองสามารถดูได้ดังรูปที่ ข.1-24



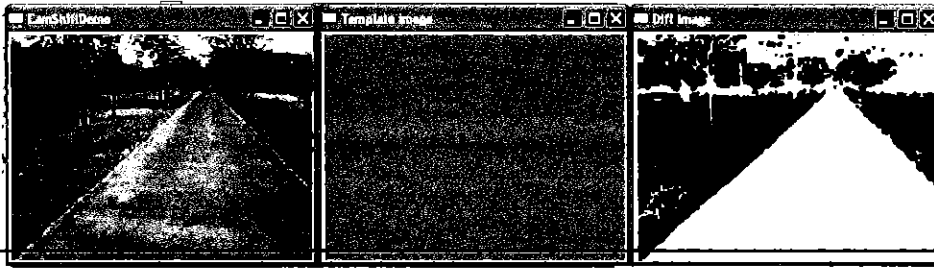
รูปที่ ข.1 แสดงการทำ Image segmentation



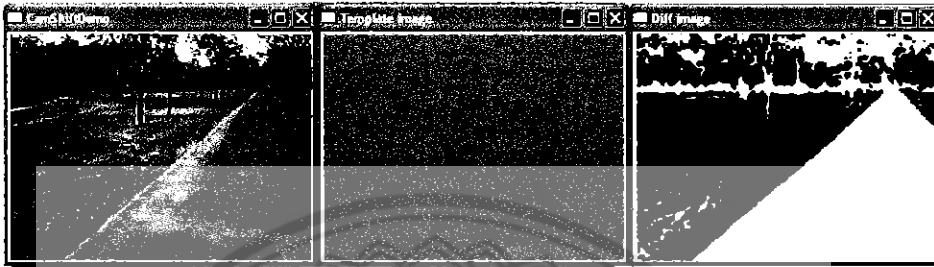
รูปที่ ข.2 แสดงการทำ Image segmentation(ต่อ)



รูปที่ ข.3 แสดงการทำ Image segmentation(ต่อ)



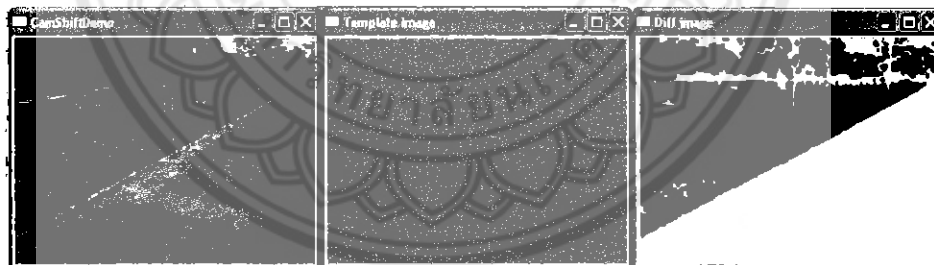
รูปที่ ข.4 แสดงการทำ Image segmentation(ต่อ)



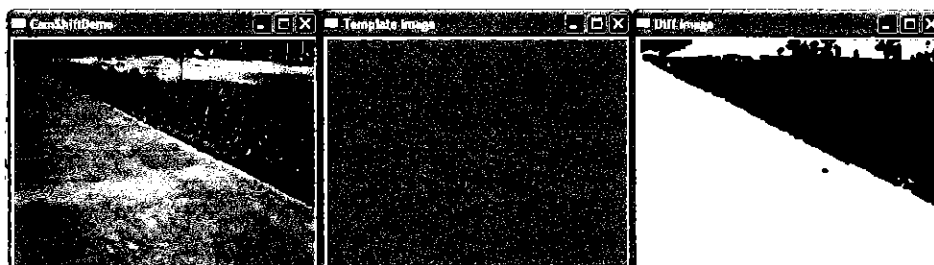
รูปที่ ข.5 แสดงการทำ Image segmentation(ต่อ)



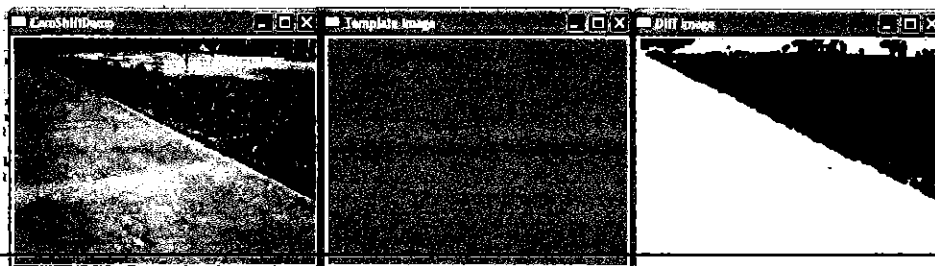
รูปที่ ข.6 แสดงการทำ Image segmentation(ต่อ)



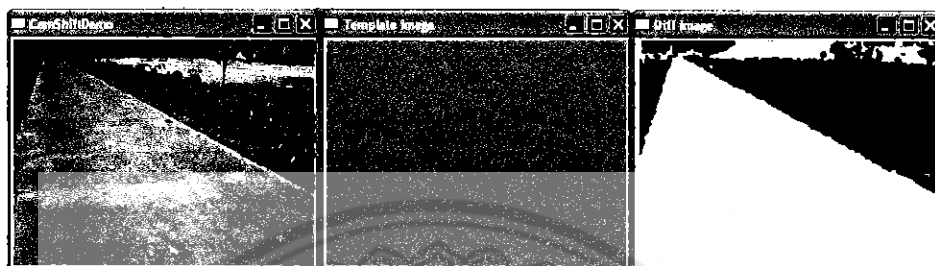
รูปที่ ข.7 แสดงการทำ Image segmentation(ต่อ)



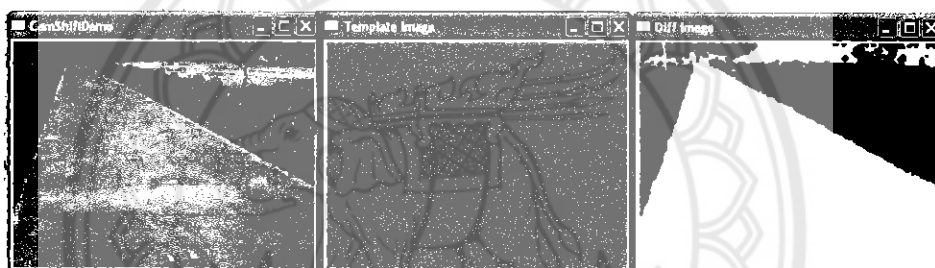
รูปที่ ข.8 แสดงการทำ Image segmentation(ต่อ)



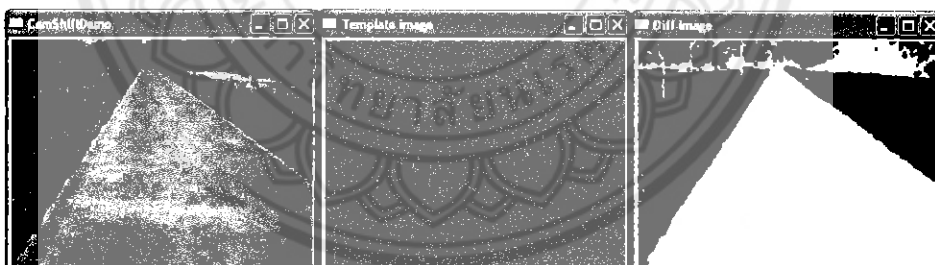
รูปที่ ข.9 แสดงการทำ Image segmentation(ต่อ)



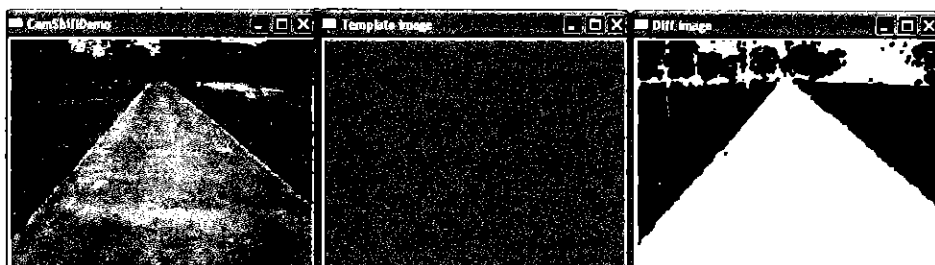
รูปที่ ข.10 แสดงการทำ Image segmentation(ต่อ)



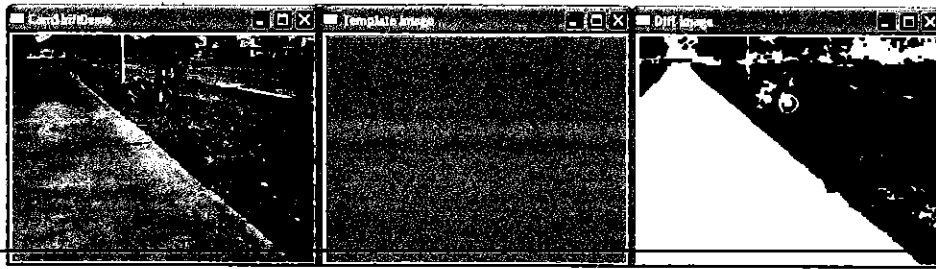
รูปที่ ข.11 แสดงการทำ Image segmentation(ต่อ)



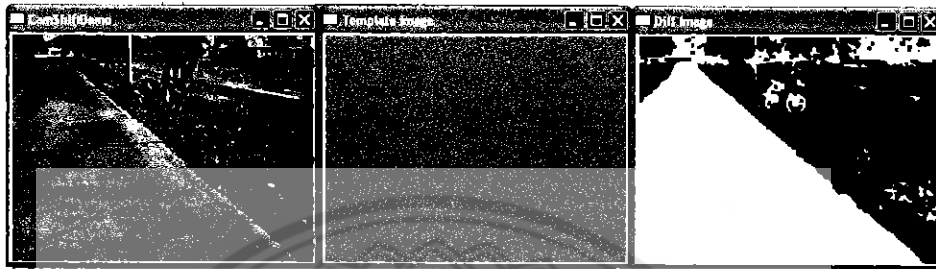
รูปที่ ข.12 แสดงการทำ Image segmentation(ต่อ)



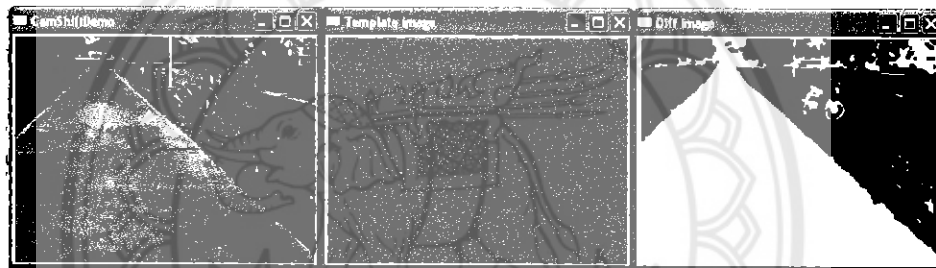
รูปที่ ข.13 แสดงการทำ Image segmentation(ต่อ)



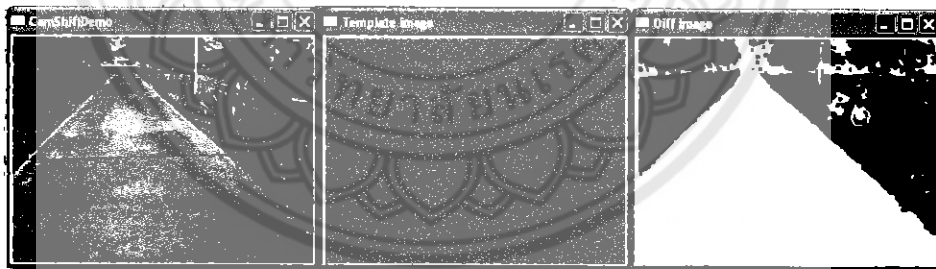
รูปที่ ข.14 แสดงการทำ Image segmentation(ต่อ)



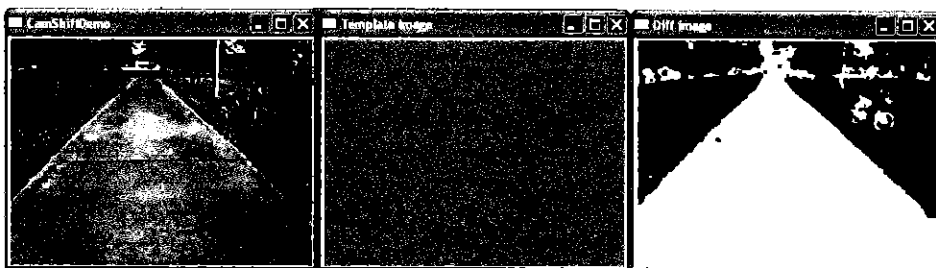
รูปที่ ข.15 แสดงการทำ Image segmentation(ต่อ)



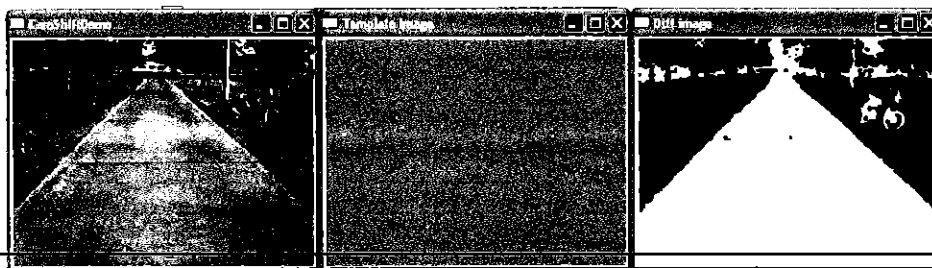
รูปที่ ข.16 แสดงการทำ Image segmentation(ต่อ)



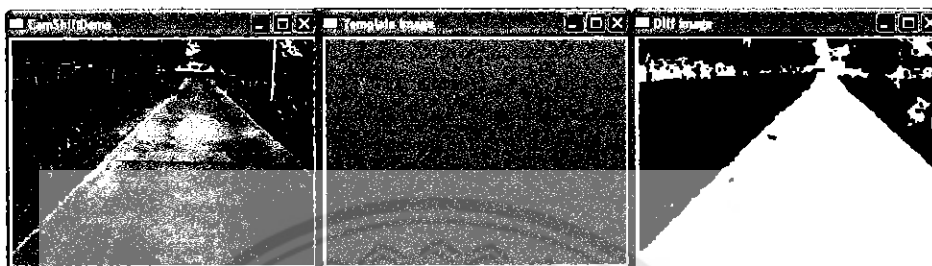
รูปที่ ข.17 แสดงการทำ Image segmentation(ต่อ)



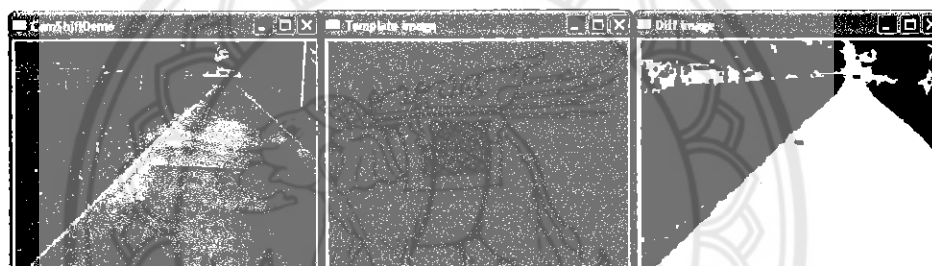
รูปที่ ข.18 แสดงการทำ Image segmentation(ต่อ)



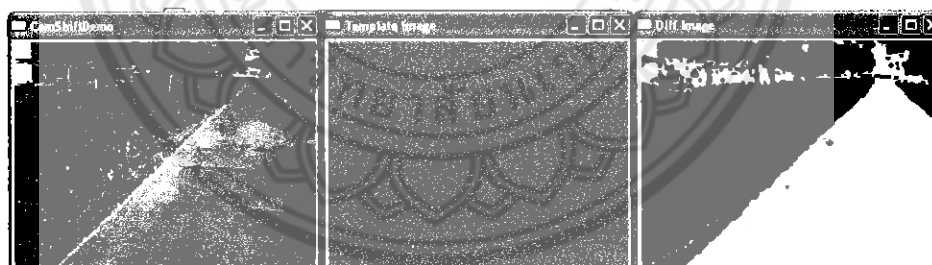
รูปที่ ข.19 แสดงการทำ Image segmentation(ต่อ)



รูปที่ ข.20 แสดงการทำ Image segmentation(ต่อ)



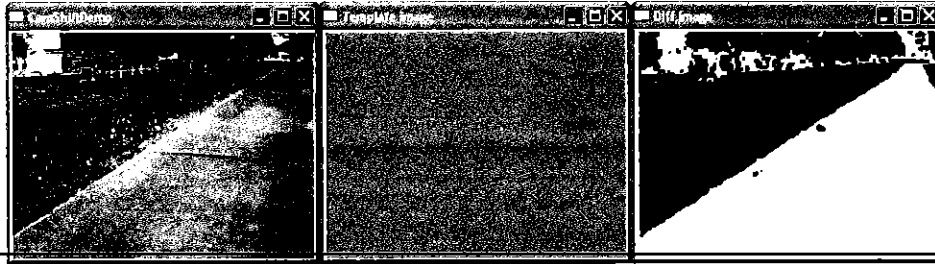
รูปที่ ข.21 แสดงการทำ Image segmentation(ต่อ)



รูปที่ ข.22 แสดงการทำ Image segmentation(ต่อ)



รูปที่ ข.23 แสดงการทำ Image segmentation(ต่อ)

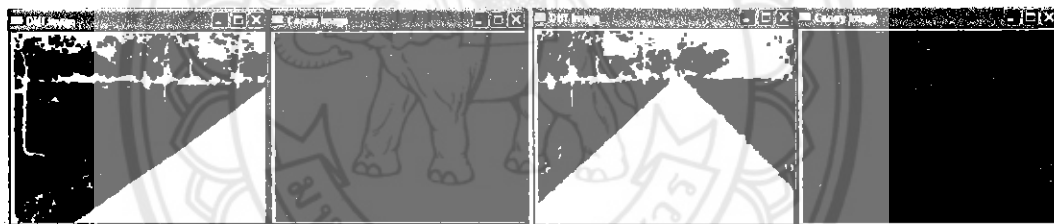


รูปที่ ข.24 แสดงการทำ Image segmentation(ต่อ)

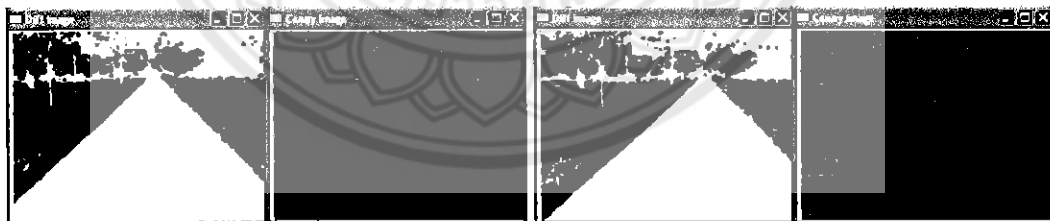
รูปที่ ข.1-24 แสดงผลของการทำ Image segmentation ในกรณีต่าง ๆ เพิ่มเติม

ข.2 ผลของการทำ Canny Edge Detection

ผลของการทำ Canny Edge Detection ในกรณีที่ถนนอยู่ทางซ้าย ตรงกลาง และทางขวา ซึ่งเป็นผลการทดลองที่ได้ทำไว้เพิ่มเติม เพื่อให้รู้ว่าการทำ Canny Edge Detection สามารถทำได้จริง โดยผลการทดลองสามารถดูได้ดังรูปที่ ข.25-36



รูปที่ ข.25 แสดงการทำ Canny Edge Detection



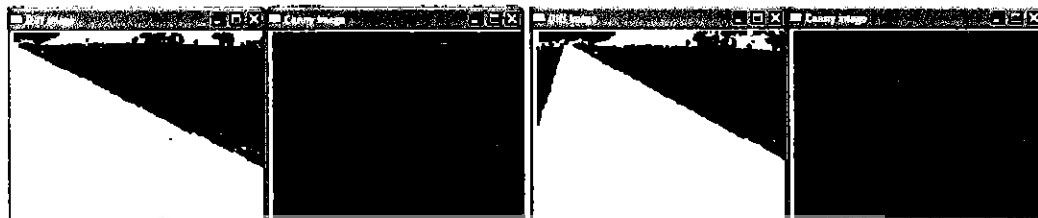
รูปที่ ข.26 แสดงการทำ Canny Edge Detection(ต่อ)



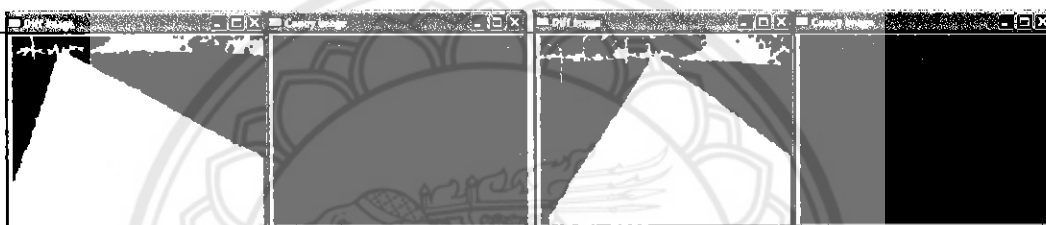
รูปที่ ข.27 แสดงการทำ Canny Edge Detection(ต่อ)



รูปที่ ข.28 แสดงการทำ Canny Edge Detection(ต่อ)



รูปที่ ข.29 แสดงการทำ Canny Edge Detection(ต่อ)



รูปที่ ข.30 แสดงการทำ Canny Edge Detection(ต่อ)



รูปที่ ข.31 แสดงการทำ Canny Edge Detection(ต่อ)



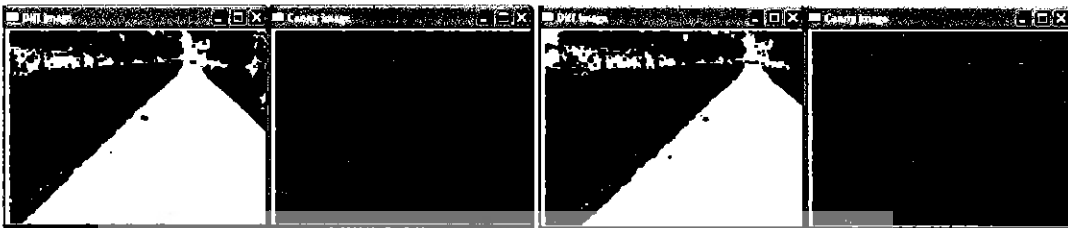
รูปที่ ข.32 แสดงการทำ Canny Edge Detection(ต่อ)



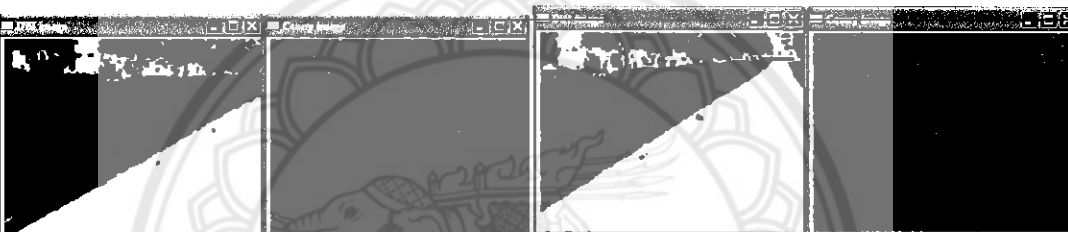
รูปที่ ข.33 แสดงการทำ Canny Edge Detection(ต่อ)



รูปที่ ข.34 แสดงการทำ Canny Edge Detection(ต่อ)



รูปที่ ข.35 แสดงการทำ Canny Edge Detection(ต่อ)

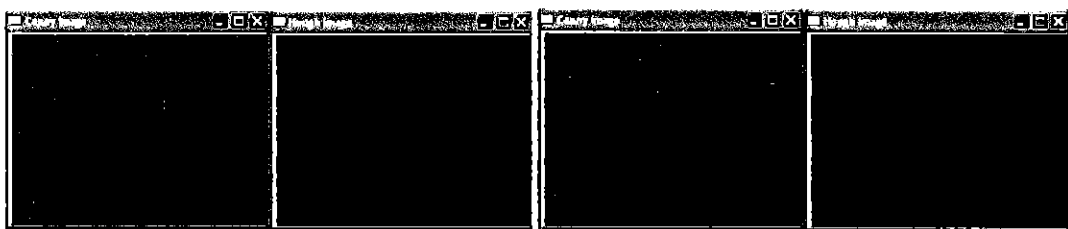


รูปที่ ข.36 แสดงการทำ Canny Edge Detection(ต่อ)

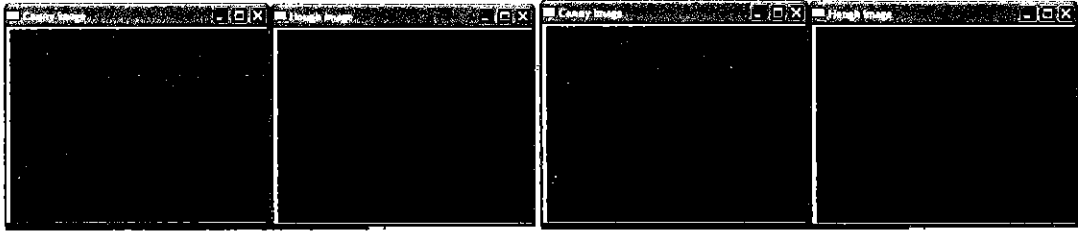
รูปที่ ข.25-36 แสดงผลของการทำ Canny Edge Detection ในกรณีต่าง ๆ เพิ่มเติม

ข.3 ผลของการทำ Hough Transform

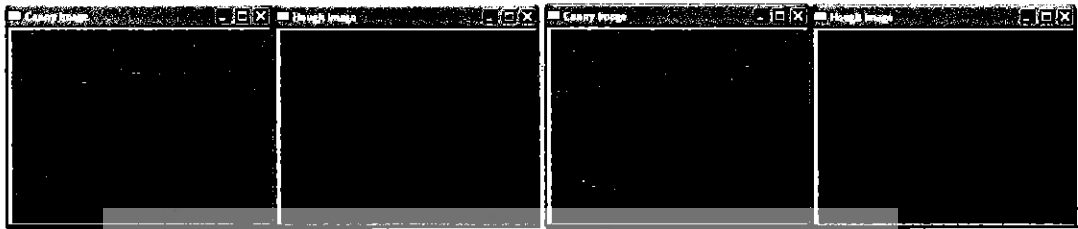
ผลของการทำ Hough Transform ในกรณีที่ถนนอยู่ทางซ้าย ตรงกลาง และทางขวา ซึ่งเป็นผลการทดลองที่ได้ทำไว้เพิ่มเติม เพื่อให้รู้ว่าการทำ Hough Transform สามารถทำได้จริง โดยผลการทดลองสามารถดูได้ดังรูปที่ ข.37-48



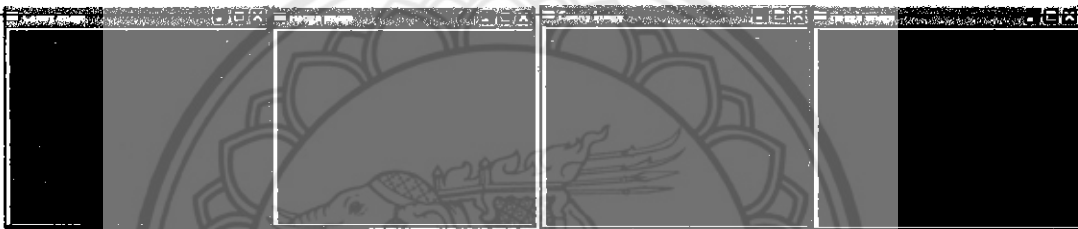
รูปที่ ข.37 แสดงการทำ Hough Transform



รูปที่ ข.38 แสดงการทำ Hough Transform(ต่อ)



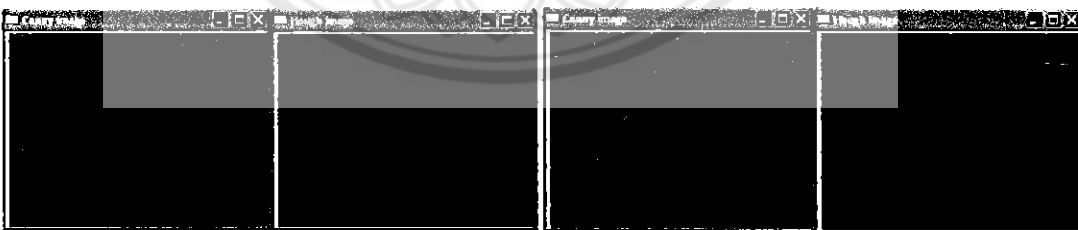
รูปที่ ข.39 แสดงการทำ Hough Transform(ต่อ)



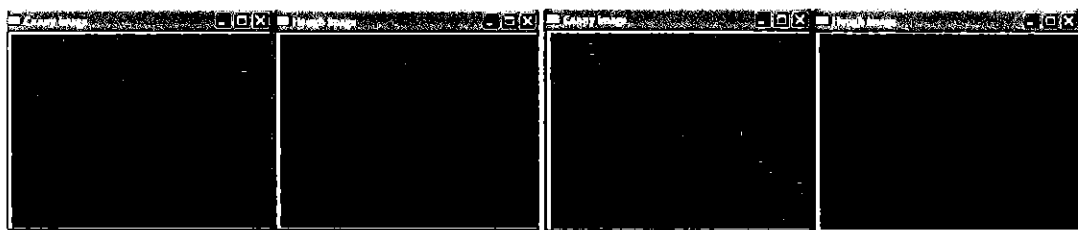
รูปที่ ข.40 แสดงการทำ Hough Transform(ต่อ)



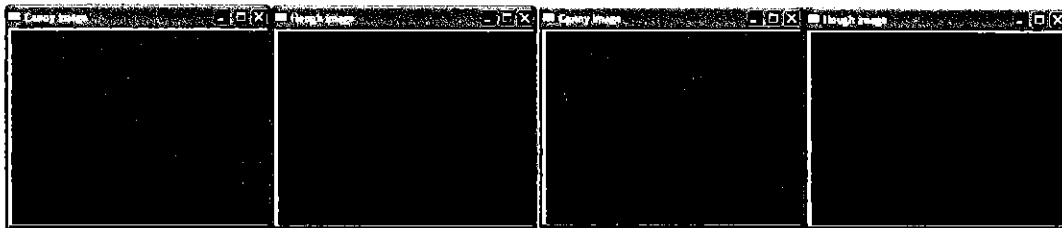
รูปที่ ข.41 แสดงการทำ Hough Transform(ต่อ)



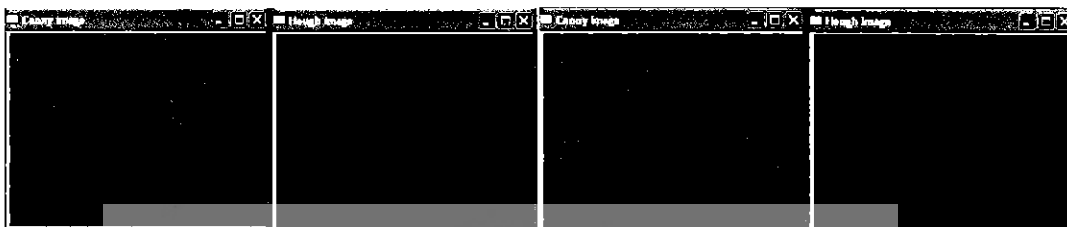
รูปที่ ข.42 แสดงการทำ Hough Transform(ต่อ)



รูปที่ ข.43 แสดงการทำ Hough Transform(ต่อ)



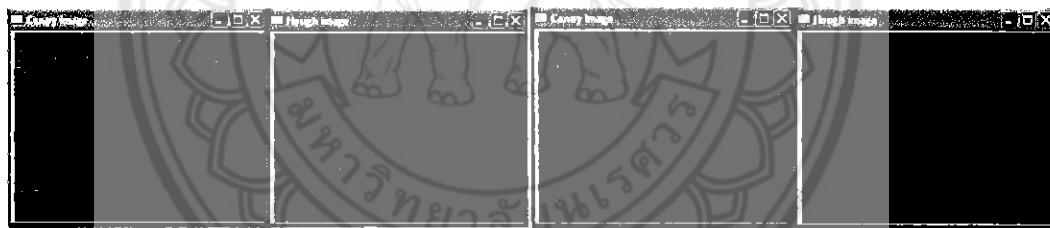
รูปที่ ข.44 แสดงการทำ Hough Transform(ต่อ)



รูปที่ ข.45 แสดงการทำ Hough Transform(ต่อ)



รูปที่ ข.46 แสดงการทำ Hough Transform(ต่อ)



รูปที่ ข.47 แสดงการทำ Hough Transform(ต่อ)



รูปที่ ข.48 แสดงการทำ Hough Transform(ต่อ)

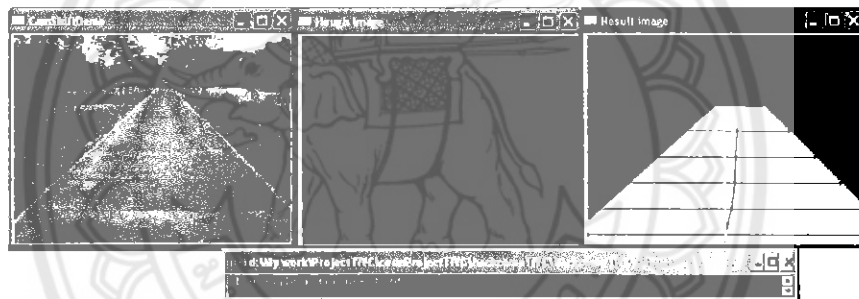
รูปที่ ข.37-48 แสดงผลของการทำ Hough Transform ในกรณีต่าง ๆ เพิ่มเติม

ข.4 ผลของการทำเส้นเสมือน

ผลของการทำเส้นเสมือน ในกรณีที่ดินนอูทางซ้าย ตรงกลาง และทางขวา ซึ่งเป็นผลการทดลองที่ได้ทำไว้เพิ่มเติม เพื่อให้รู้ว่าการทำเส้นเสมือนสามารถทำได้จริง โดยผลการทดลองสามารถดูได้ดังรูปที่ ข.49-71



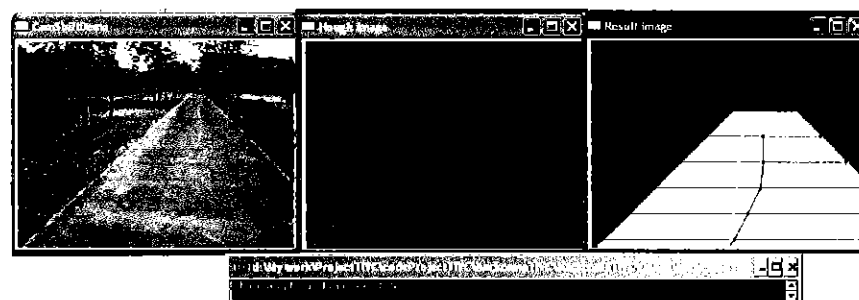
รูปที่ ข.49 แสดงการทำ เส้นเสมือน



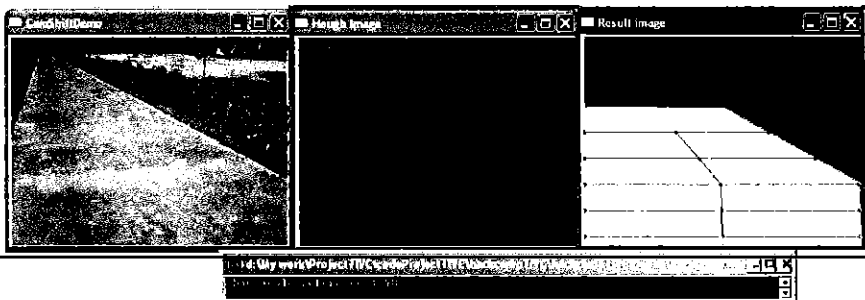
รูปที่ ข.50 แสดงการทำ เส้นเสมือน(ต่อ)



รูปที่ ข.51 แสดงการทำ เส้นเสมือน(ต่อ)



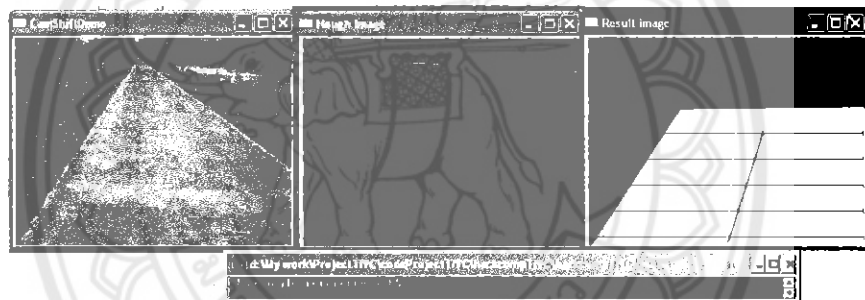
รูปที่ ข.52 แสดงการทำ เส้นเสมือน(ต่อ)



รูปที่ ข.58 แสดงการทำ เส้นเสมือน(ต่อ)



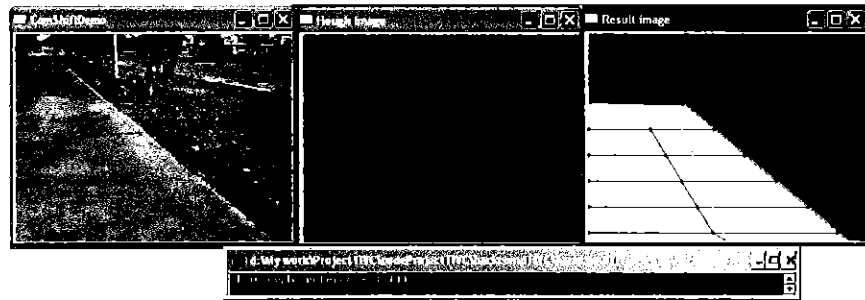
รูปที่ ข.59 แสดงการทำ เส้นเสมือน(ต่อ)



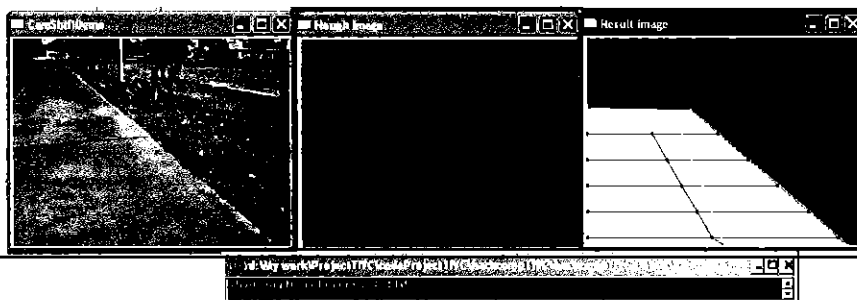
รูปที่ ข.60 แสดงการทำ เส้นเสมือน(ต่อ)



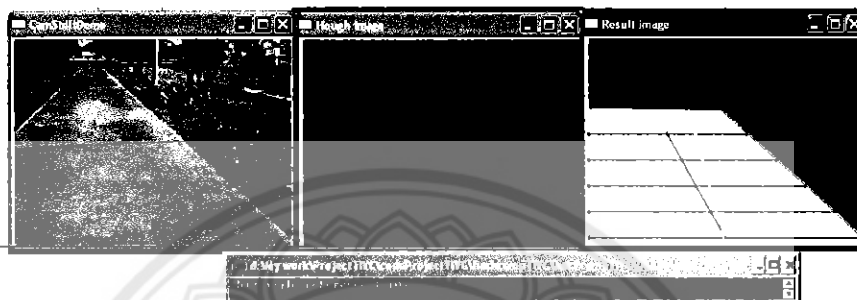
รูปที่ ข.61 แสดงการทำ เส้นเสมือน(ต่อ)



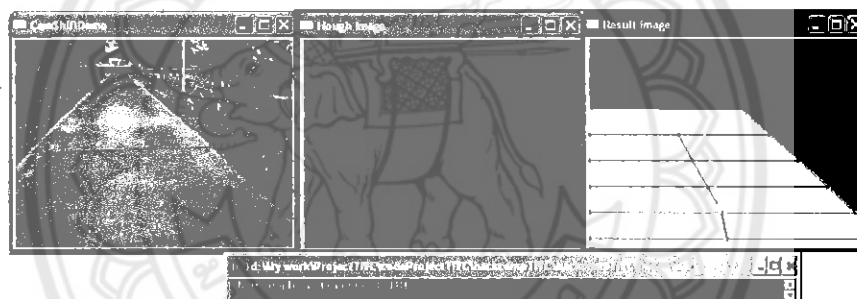
รูปที่ ข.62 แสดงการทำ เส้นเสมือน(ต่อ)



รูปที่ ข.63 แสดงการทำ เส้นเสมือน(ต่อ)



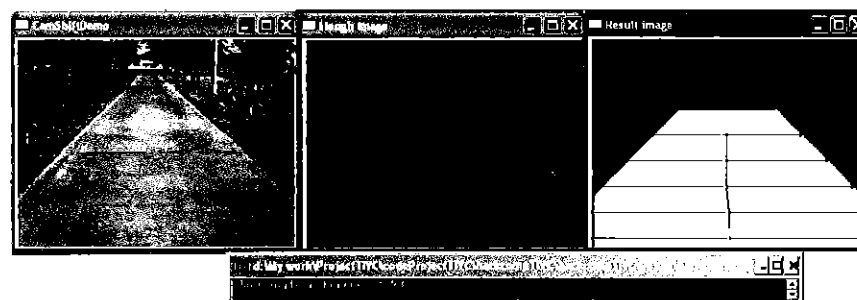
รูปที่ ข.64 แสดงการทำ เส้นเสมือน(ต่อ)



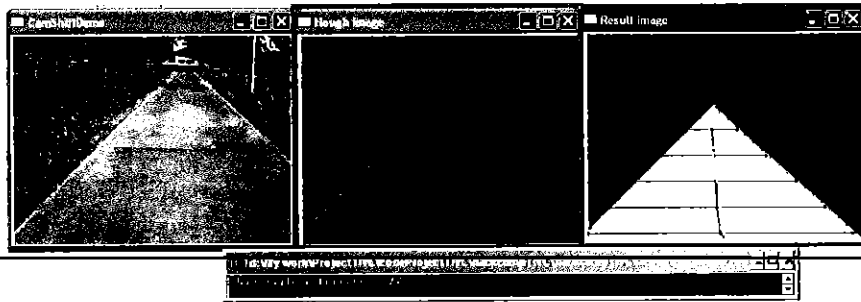
รูปที่ ข.65 แสดงการทำ เส้นเสมือน(ต่อ)



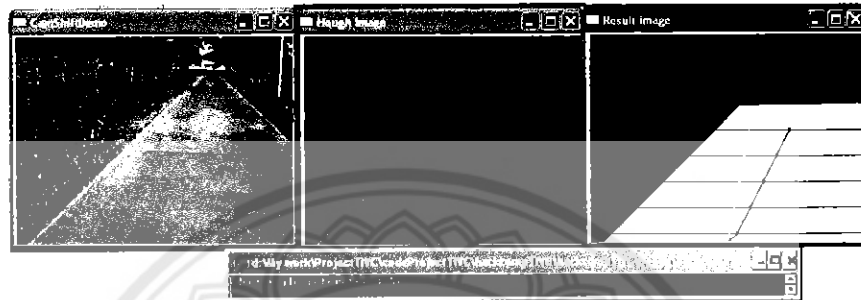
รูปที่ ข.66 แสดงการทำ เส้นเสมือน(ต่อ)



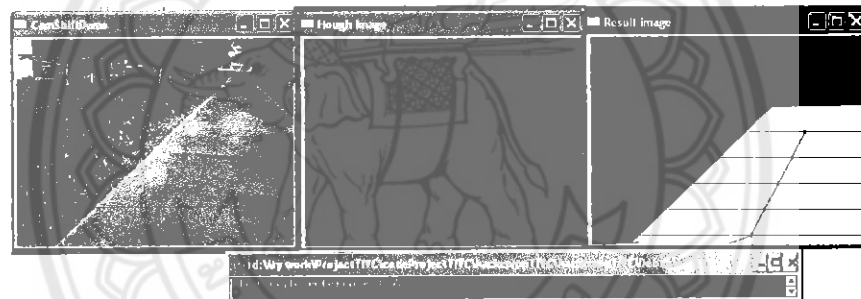
รูปที่ ข.67 แสดงการทำ เส้นเสมือน(ต่อ)



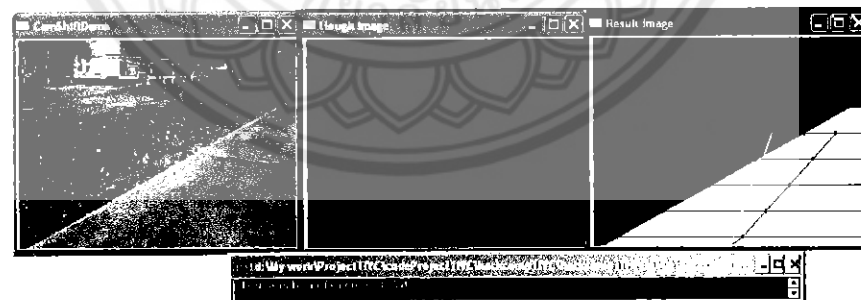
รูปที่ ข.68 แสดงการทำ เส้นเสมือน(ต่อ)



รูปที่ ข.69 แสดงการทำ เส้นเสมือน(ต่อ)



รูปที่ ข.70 แสดงการทำ เส้นเสมือน(ต่อ)



รูปที่ ข.71 แสดงการทำ เส้นเสมือน(ต่อ)

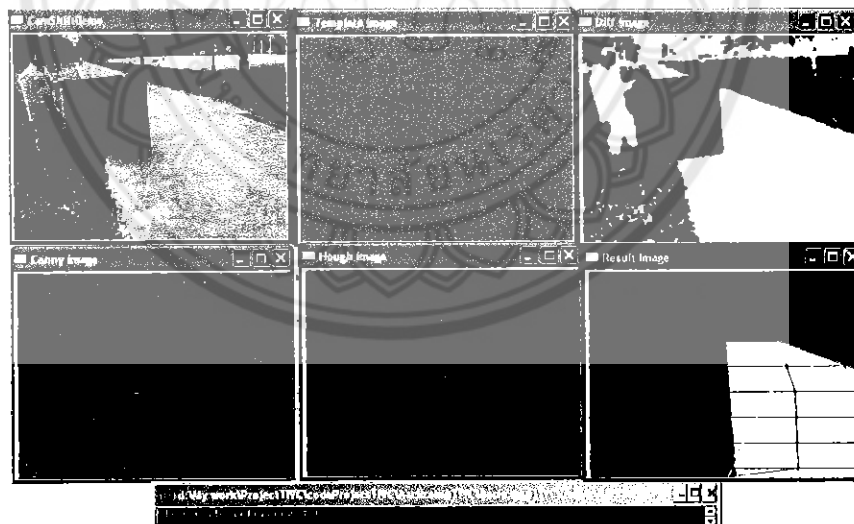
รูปที่ ข.49-71 แสดงผลของการทำ เส้นเสมือนในกรณีต่าง ๆ เพิ่มเติม

ข.5 ผลของการประมวลผลภาพโดยไม่มีสิ่งกีดขวาง

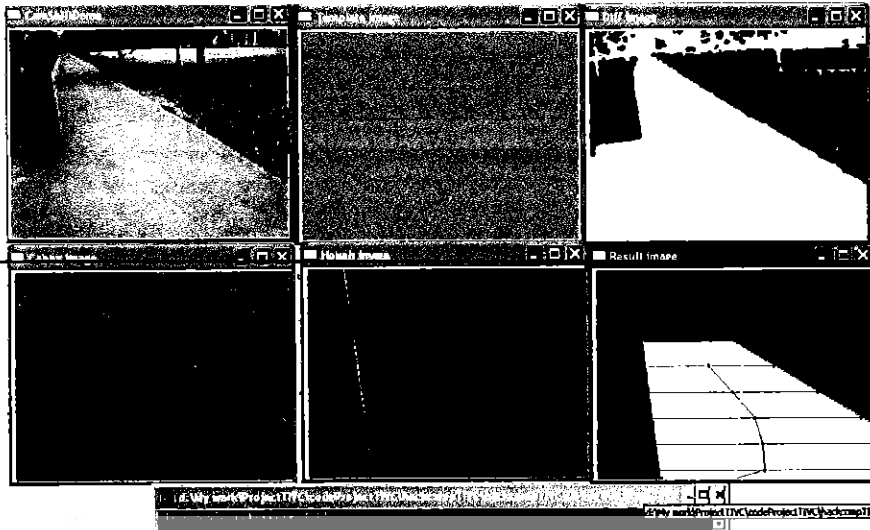
ผลของการประมวลผลภาพโดยไม่มีสิ่งกีดขวาง ในกรณีที่ดินนอู่ทางซ้าย ตรงกลาง และทางขวา ซึ่งเป็นผลการทดลองที่ได้ทำไว้เพิ่มเติม เพื่อให้รู้ว่าการประมวลผลภาพโดยไม่มีสิ่งกีดขวางสามารถทำได้ประมาณ 70 เปอร์เซ็นต์ โดยผลการทดลองสามารถดูได้ดังรูปที่ ข.72-92



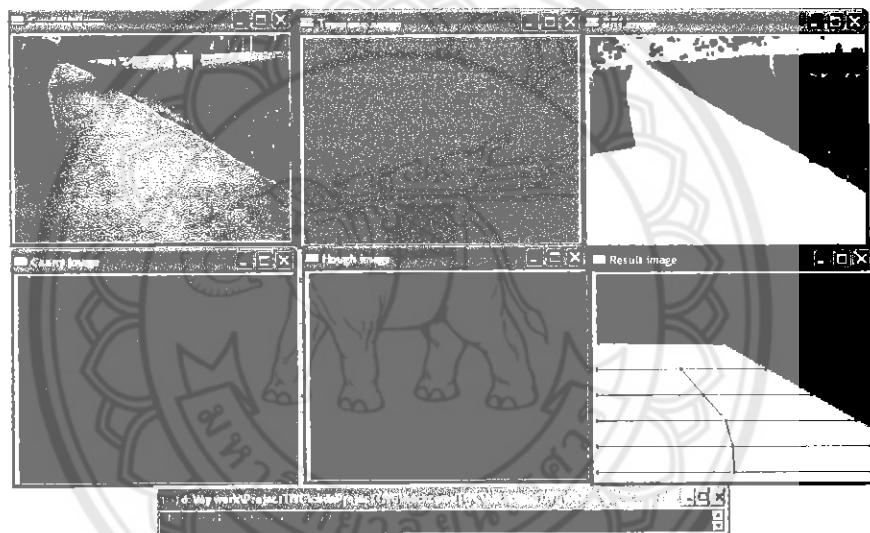
รูปที่ ข.72 แสดงการประมวลผลภาพโดยไม่มีสิ่งกีดขวาง



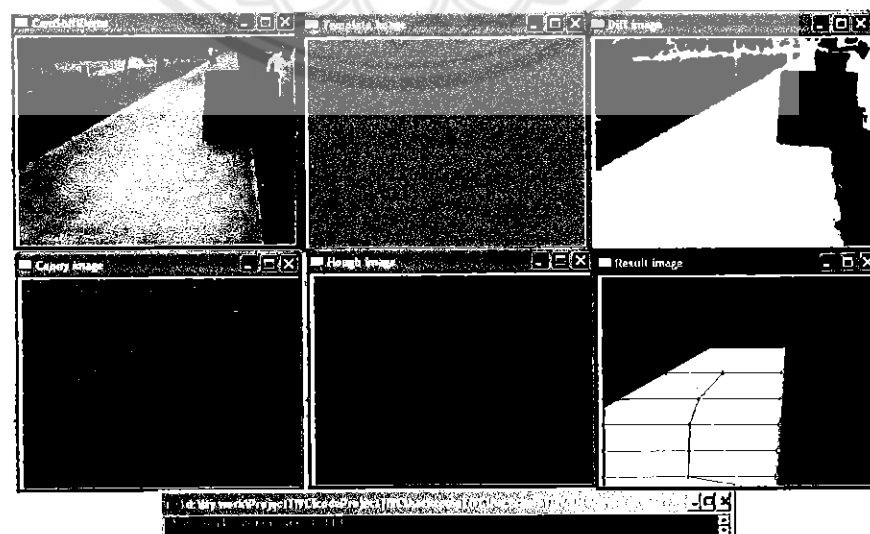
รูปที่ ข.73 แสดงการประมวลผลภาพโดยไม่มีสิ่งกีดขวาง(ต่อ)



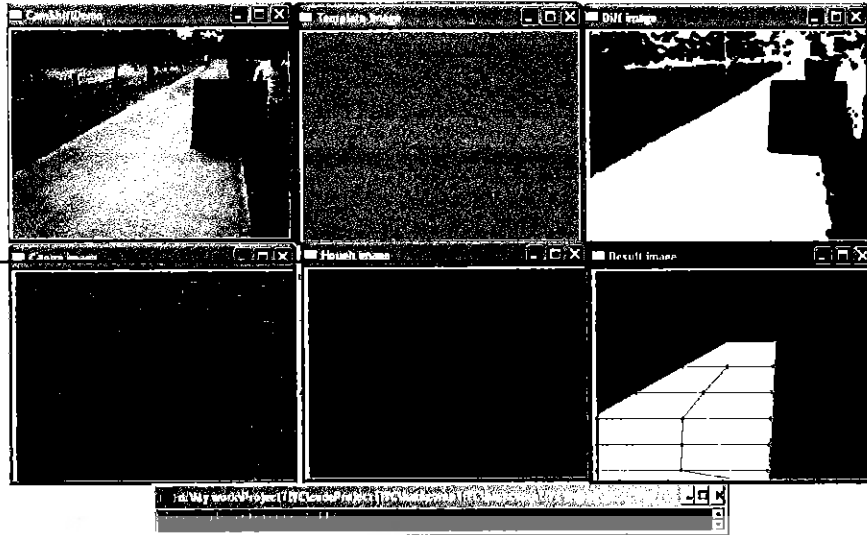
รูปที่ ข.74 แสดงการประมวลผลภาพ โดยไม่มีสิ่งกีดขวาง(ต่อ)



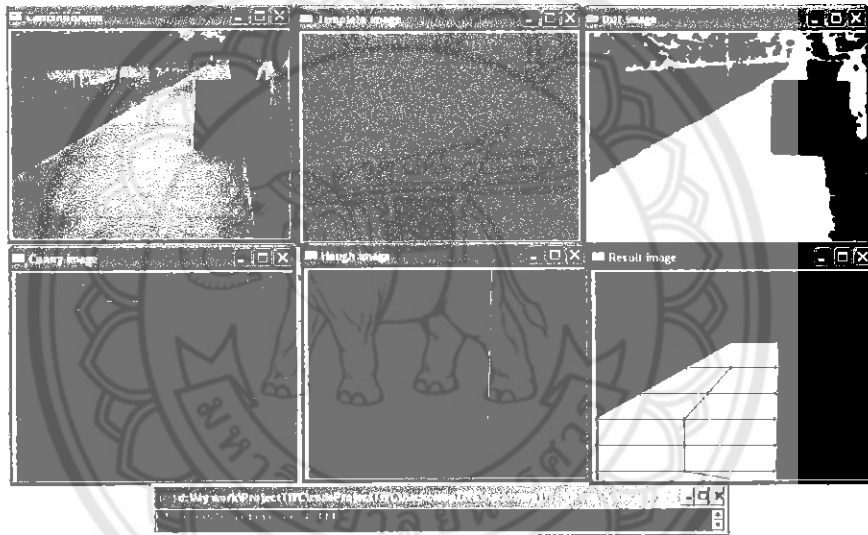
รูปที่ ข.75 แสดงการประมวลผลภาพ โดยไม่มีสิ่งกีดขวาง(ต่อ)



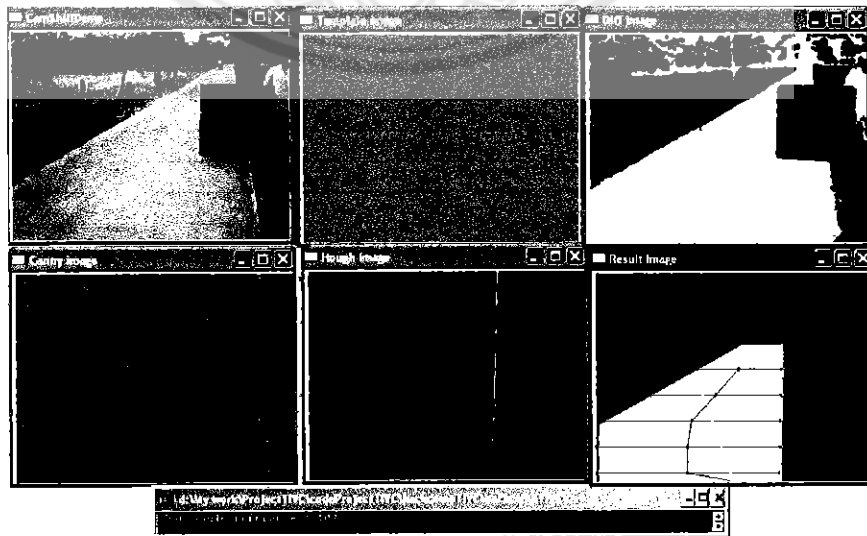
รูปที่ ข.76 แสดงการประมวลผลภาพ โดยไม่มีสิ่งกีดขวาง(ต่อ)



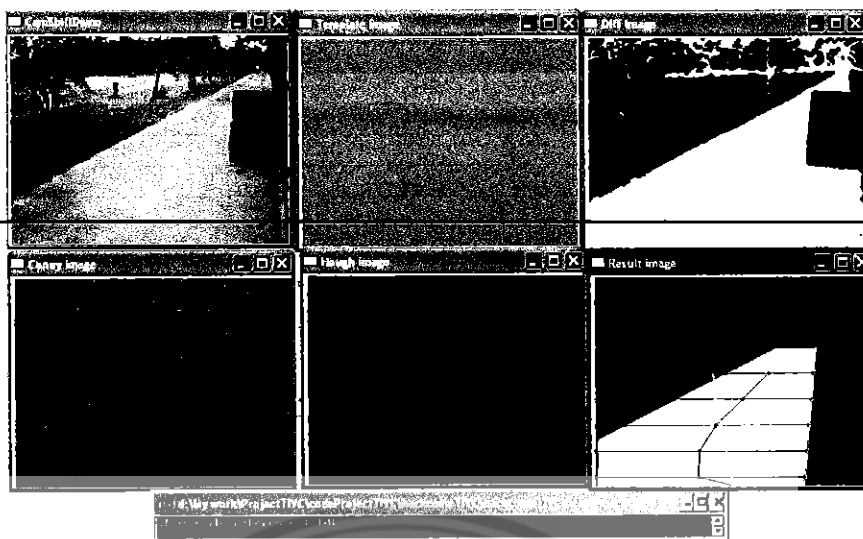
รูปที่ ข.77 แสดงการประมวลผลภาพโดยไม่มีสิ่งกีดขวาง(ต่อ)



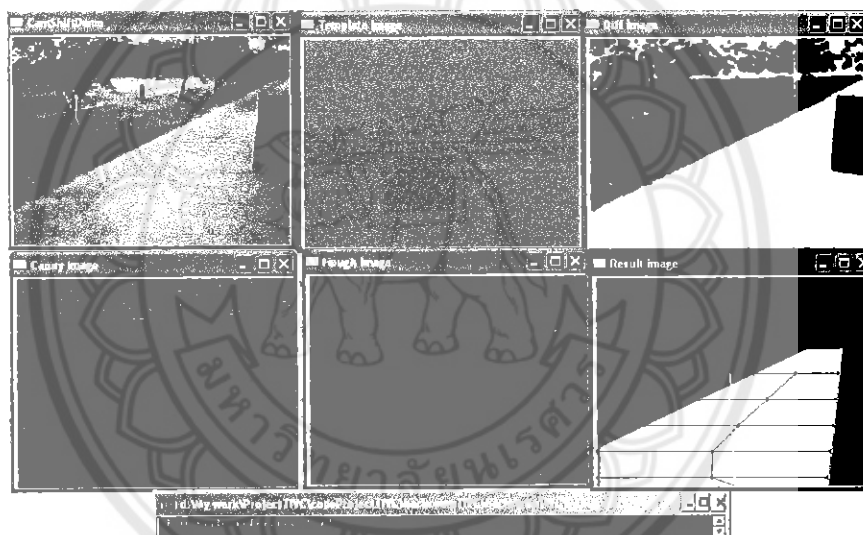
รูปที่ ข.77 แสดงการประมวลผลภาพโดยไม่มีสิ่งกีดขวาง(ต่อ)



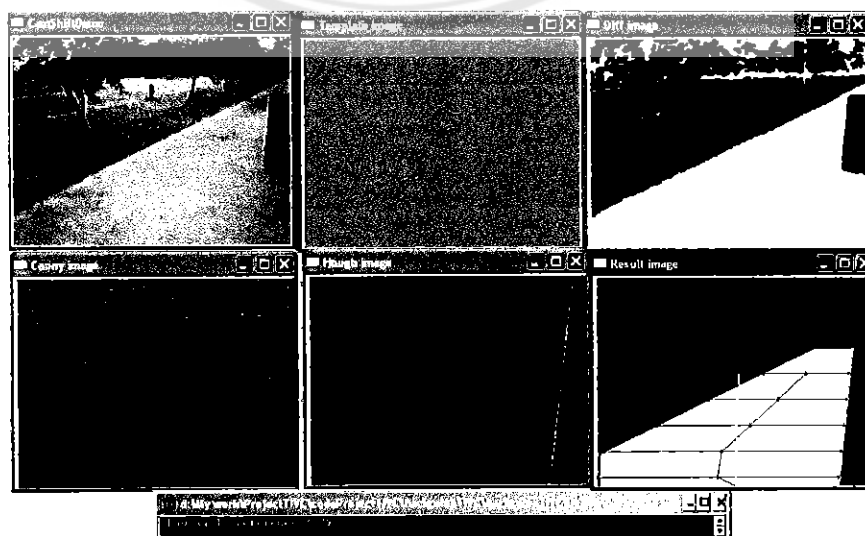
รูปที่ ข.78 แสดงการประมวลผลภาพโดยไม่มีสิ่งกีดขวาง(ต่อ)



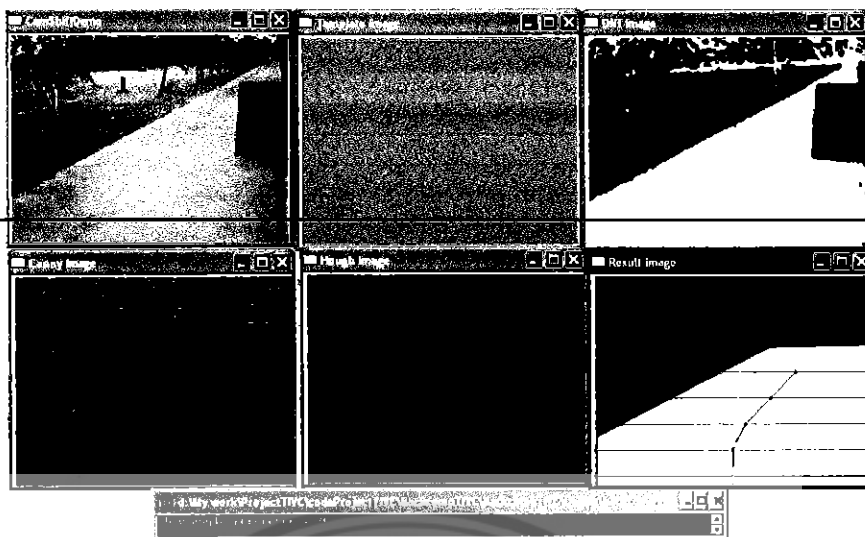
รูปที่ ข.79 แสดงการประมวลผลภาพโดยไม่มีสิ่งกีดขวาง(ต่อ)



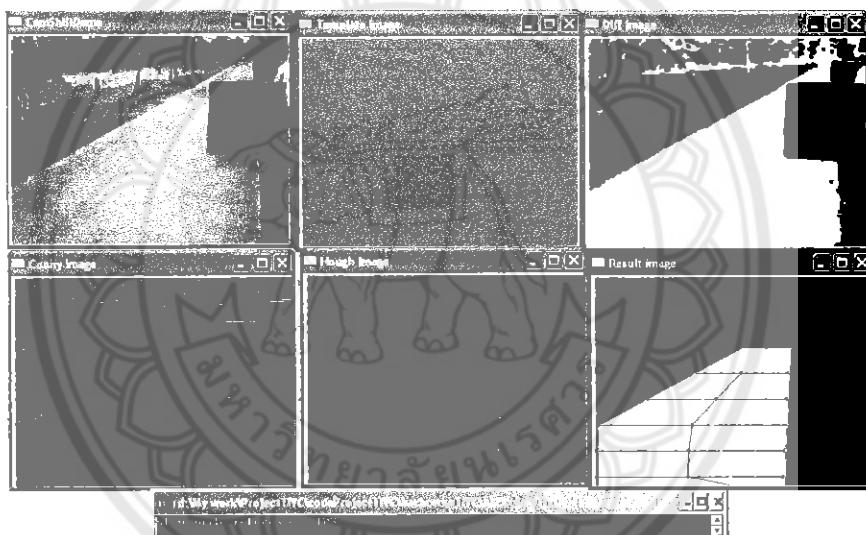
รูปที่ ข.80 แสดงการประมวลผลภาพโดยไม่มีสิ่งกีดขวาง(ต่อ)



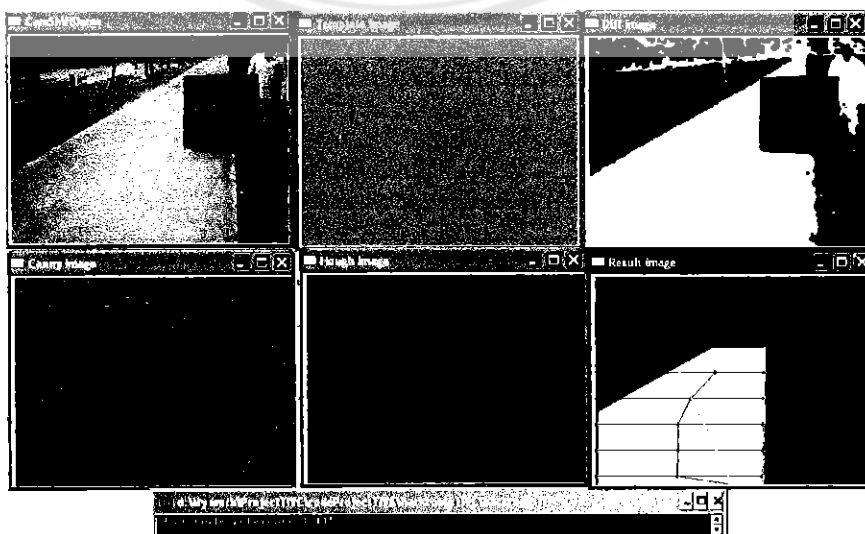
รูปที่ ข.81 แสดงการประมวลผลภาพโดยไม่มีสิ่งกีดขวาง(ต่อ)



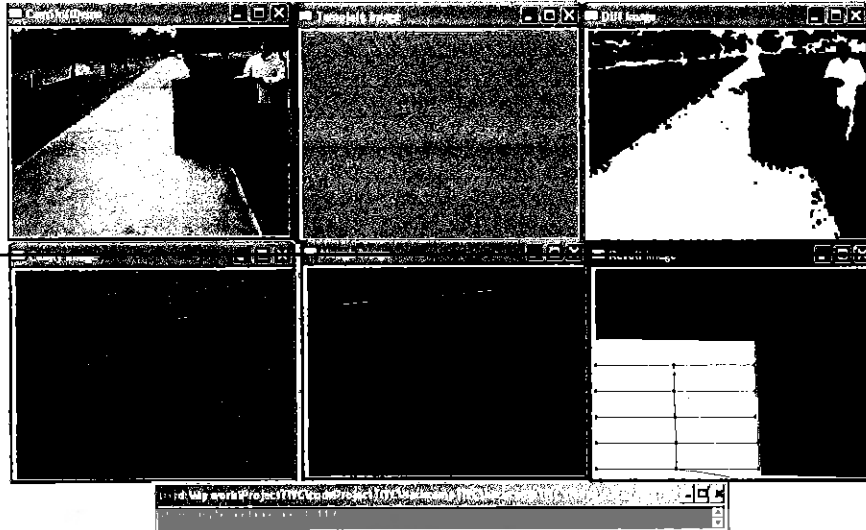
รูปที่ ข.82 แสดงการประมวลผลภาพ โดยไม่มีสิ่งกีดขวาง(ต่อ)



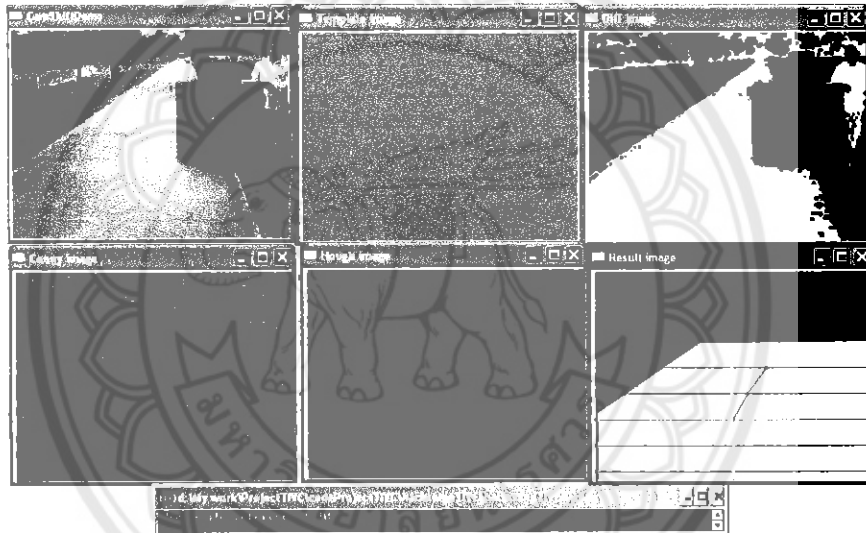
รูปที่ ข.83 แสดงการประมวลผลภาพ โดยไม่มีสิ่งกีดขวาง(ต่อ)



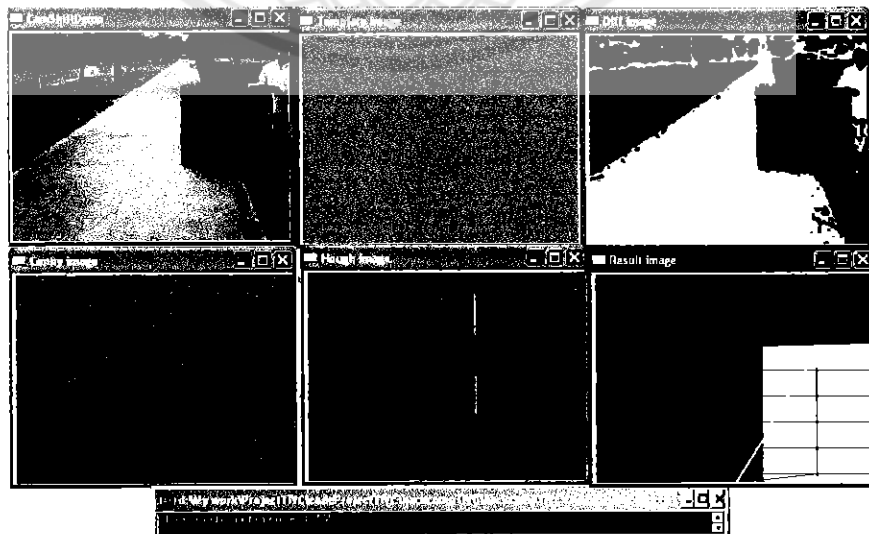
รูปที่ ข.84 แสดงการประมวลผลภาพ โดยไม่มีสิ่งกีดขวาง(ต่อ)



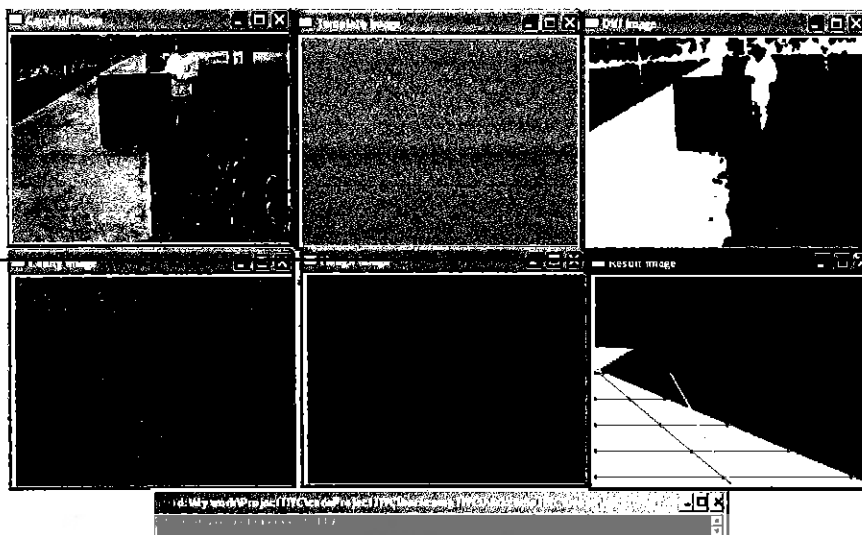
รูปที่ ข.85 แสดงการประมวลผลภาพ โดยไม่มีสิ่งกีดขวาง(ต่อ)



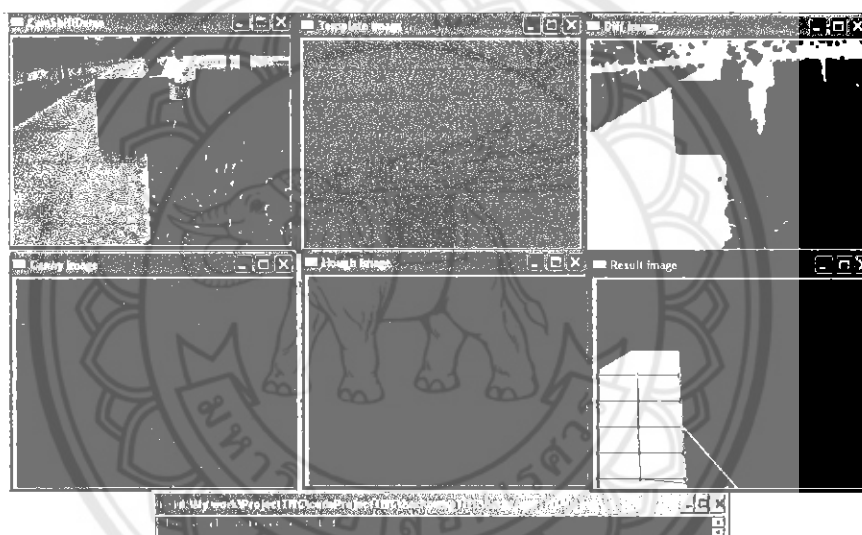
รูปที่ ข.86 แสดงการประมวลผลภาพ โดยไม่มีสิ่งกีดขวาง(ต่อ)



รูปที่ ข.87 แสดงการประมวลผลภาพ โดยไม่มีสิ่งกีดขวาง(ต่อ)



รูปที่ ข.91 แสดงการประมวลผลภาพ โดยไม่มีสิ่งกีดขวาง(ต่อ)



รูปที่ ข.92 แสดงการประมวลผลภาพ โดยไม่มีสิ่งกีดขวาง(ต่อ)

รูปที่ ข.72-92 แสดงผลของการประมวลผลภาพ โดยไม่มีสิ่งกีดขวาง ในกรณีต่าง ๆ เพิ่มเติม

ประวัติผู้เขียนโครงการ



ชื่อ นายณรงค์ รวมสุข
 ภูมิลำเนา 255 หมู่ 2 ตำบลบง อำเภอกุฉินารายณ์
 จังหวัดพะเยา 56110

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนกุฉินารายณ์วิทยาคม
- ปัจจุบันกำลังศึกษาอยู่ในระดับปริญญาตรีชั้นปีที่ 4
 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
 มหาวิทยาลัยนเรศวร

E-mail : hack_comp@hotmail.com

