การพัฒนาวิธีการตรวจจับช่องทางเดินรถที่เหมาะสมสำหรับการออกแบบบน FPGA

A Development of an Optimized Lane Detection for FPGA-Based Design

วรวิทย์ เผือกจีน

Worawit Phueakjeen

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา

วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering

Prince of Songkla University

2556

ชื่อวิทยานิพนธ์ การพัฒนาวิธีการตรวจจับช่องทางเดินรถที่เหมาะสมสำหรับการออกแบบบน FPGA

ผู้เขียน นายวรวิทย์ เผือกจีน

สาขาวิชา วิศวกรรมไฟฟ้า

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	**คณะกรรมการสอบ**
.....	.....ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ณัฏฐา จินดาเพ็ชร์**)**	(ดร.วฤทธิ์ วิชกูล)
	.....กรรมการ
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	(ดร. เดือนเพ็ญ กชกรจารุพงศ์)
.....	.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.นิคม สุวรรณวร)	(ผู้ช่วยศาสตราจารย์ ดร.ณัฏฐา จินดาเพ็ชร์)
	.....กรรมการ
	(ผู้ช่วยศาสตราจารย์ ดร.นิคม สุวรรณวร)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....

(รองศาสตราจารย์ ดร.ธีระพล ศรีชนะ)

คณบดีบัณฑิตวิทยาลัย

ขอรับรองว่า ผลงานวิจัยนี้เป็นผลมาจาการศึกษาวิจัยของนักศึกษาเอง และได้แสดงความขอบคุณบุคคลที่มีส่วนช่วยเหลือแล้ว

ลงชื่อ.....
(ผู้ช่วยศาสตราจารย์ ดร.ณัฏฐา จินดาเพ็ชร์)
อาจารย์ที่ปรึกษาวิทยานิพนธ์

ลงชื่อ.....
(นายวรวิทย์ เผือกจีน)
นักศึกษา

ข้าพเจ้าขอรับรองว่า ผลงานวิจัยนี้ไม่เคยเป็นส่วนหนึ่งในการอนุมัติปริญญาในระดับใดมาก่อน และไม่ได้ถูกใช้ในการยื่นขออนุมัติปริญญาในขณะนี้

ลงชื่อ.....

(นายวรวิทย์ เผือกจีน)

นักศึกษา

ชื่อวิทยานิพนธ์ การพัฒนาวิธีการตรวจจับช่องทางเดินรถที่เหมาะสมสำหรับการออกแบบบน FPGA

ผู้เขียน นายวรวิทย์ เผือกจีน

สาขาวิชา วิศวกรรมไฟฟ้า

ปีการศึกษา 2555

บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอการคัดเลือกอัลกอริทึมหาขอบภาพที่เหมาะสมที่สุด เพื่อนำไปพัฒนากระบวนการหาขอบของถนน หรือช่องทางเดินรถ ซึ่งมีส่วนของความเร็ว ความถูกต้อง และทรัพยากรการประมวลผลอันจำกัด เป็นประเด็นหลักที่ต้องนำมาพิจารณา เพื่อนำไปสู่การพัฒนาต่อให้สามารถทำงานได้จริงบนเทคโนโลยี FPGA (Field Programable Gate Array) โดยอัลกอริทึมการหาเส้นขอบภาพที่นำมาวิเคราะห์นั้น คือ Canny, Prewitt, Sobel และ Roberts โดยใช้โปรแกรม MATLAB เป็นเครื่องมือช่วยในการวิเคราะห์ นอกจากนี้ในงานนี้ยังมีการเพิ่มกระบวนการกรองเพื่อช่วยคัดแยกแสงสีแดง เขียว น้ำเงินออกจากภาพ ทำให้กระบวนการหาขอบภาพมีประสิทธิภาพสูงยิ่งขึ้น ภาพที่ใช้ในการวิเคราะห์เป็นภาพขนาด 640x480 พิกเซล ได้จากกล้องถ่ายวิดีโอที่ถ่ายภาพถนนด้วยอัตรา 30 เฟรมต่อวินาที ที่ความเร็วรถ 90 กิโลเมตรต่อชั่วโมง จากการทดลองพบว่าวิธีของ Canny ใช้เวลาในการคำนวณมากที่สุดแต่ให้คำตอบจำนวนเส้นถนนมากเกินไปที่สุด ในบรรดาวิธีการทั้งสี่ วิธีการของ Roberts นอกจากมีความซับซ้อนน้อยแล้ว ยังทำงานเร็วที่สุด (เร็วกว่าของ Canny 3.14 เท่า) และให้คำตอบที่เป็นเส้นถนนจริงถูกต้องมากที่สุด ผลจากการนำวิธีการของ Roberts มาพัฒนาระบบวิเคราะห์ช่องทางเดินรถบน FPGA รุ่น Xilinx Vertex-II Pro พบว่าสามารถประมวลผลภาพ 640x480 พิกเซล ได้ภายใน 2.929 มิลลิวินาที ด้วยความเร็วสัญญาณนาฬิกา 104.874 เมกกะเฮิร์ต

คำสำคัญ: การหาเส้นขอบภาพ, MATLAB, FPGA

Thesis Title	A Development of an Optimized Lane Detection for FPGA-Based Design
Author	Mr. Worawit Phueakjeen
Major Program	Electrical Engineering
Academic Year	2012

ABSTRACT

This thesis presents an investigation to obtain an optimal edge detection algorithm for using in the road lane detection process. The main issues, including the speed, the accuracy, and the limited processing resources, were taken to consider for the realization on the FPGA (Field Programable Gate Array) technology. The edge detection algorithms of Canny, Prewitt, Sobel and Roberts were compared using MATLAB. In addition, a mask filter was applied to remove red, green, and blue values to help the edge detection process be more efficient. Sample road images were captured by a video camera with the image size of 640x480 pixels and the frame rate of 30 frames per second, and at the car speed of 90 kilometers per hour. From the experimental results, the Canny algorithm was the most time consuming process, and gave too many lines outside the road lane. Among these, the Roberts algorithm is not only the smallest size, but also gained the fastest speed (3.14 times faster than the Canny algorithm) and the most accurate one to detect the lines of the actual road lanes. The results of the Roberts algorithm based road lane detection system implementation on a Xilinx Vertex-II PRO FPGA show that our system can process a 640x480-pixel image within 2.929 milliseconds with clock speed of 104.874 megahertz.

Keywords: Edge Detection, MATLAB, FPGA.

กิตติกรรมประกาศ

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ณัฏฐา จินดาเพ็ชร์ อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก ที่ได้อุทิศเวลา ให้คำปรึกษา ชี้แนะแนวทาง ให้กำลังใจในการทำงาน และให้ความรู้ในด้านต่างๆ รวมถึงให้การสนับสนุนในเรื่องอุปกรณ์ในการทำวิจัย ตลอดจนช่วยตรวจสอบและแก้ไขวิทยานิพนธ์ให้เป็นไปอย่างสมบูรณ์

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.นิคม สุวรรณวร อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ที่ได้กรุณาให้คำปรึกษา คำแนะนำ และให้ความช่วยเหลือในงานวิจัย ตลอดจนช่วยตรวจทานแก้ไขวิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอขอบพระคุณ ดร.วฤทธิ์ วิชกูล ที่ได้กรุณาอุทิศเวลามาเป็นประธานกรรมการสอบวิทยานิพนธ์ และตรวจทานแก้ไขวิทยานิพนธ์ให้มีความสมบูรณ์

ขอขอบพระคุณ ดร.เดือนเพ็ญ กชกรจารุพงศ์ ที่กรุณาอุทิศเวลาเดินทางมาเป็นกรรมการผู้ทรงคุณวุฒิในการสอบวิทยานิพนธ์ อีกทั้งตรวจทานและแก้ไขวิทยานิพนธ์ให้มีความสมบูรณ์ยิ่งขึ้น

ขอขอบพระคุณ คณาจารย์ และเจ้าหน้าที่ในภาควิชาวิศวกรรมไฟฟ้าทุกท่าน ที่ได้ให้คำปรึกษา และให้การช่วยเหลือในด้านต่างๆ มาโดยตลอด จนกระทั่งงานสำเร็จลุล่วง

ขอขอบพระคุณ บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ ที่ให้การสนับสนุนทุนในการทำวิจัยและให้ความช่วยเหลือด้านการประสานงานต่างๆ

ขอขอบคุณเพื่อนๆ นักศึกษา พี่น้องร่วมภาควิชา ที่ได้เป็นกำลังใจ และคอยชี้แนะกระตุ้นเตือนจนข้าพเจ้าได้ทำงานสำเร็จลุล่วง

และสุดท้าย ขอน้อมรำลึกถึงพระคุณบิดา มารดา ที่ส่งเสริมให้กำลังใจ และให้การสนับสนุนในเรื่องต่างๆ จนกระทั่งข้าพเจ้าประสบความสำเร็จในการศึกษา

วรวิทย์ เผือกจีน

สารบัญ

หน้า

รายการตาราง

หน้า

รายการภาพประกอบ

หน้า

สัญลักษณ์คำย่อและตัวย่อ

FPGA	Field Programmable Gate Array
BRAMs	Block RAMs
LUTs	Look Up Tables
DCM	Digital Clock Manager
fps	Frame per second
EDIF	Electronic Design Intermediate Form
IP	Intellectual Property
HDL	Hardware Description Language

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของงานวิจัย

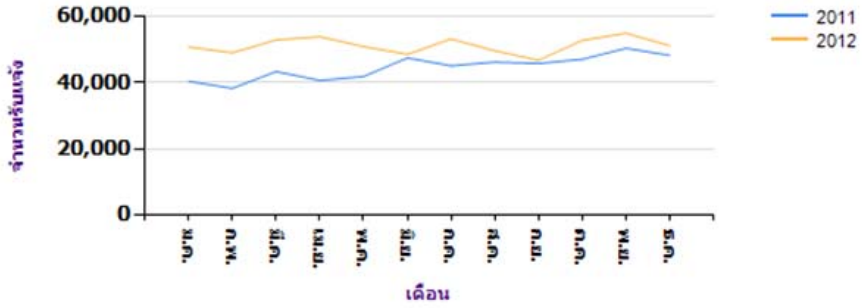
จากสถิติของกรมทางหลวง [1] พบว่า สาเหตุของการเกิดอุบัติเหตุในการจราจรทางบก มี 3 ประการ คือ

1. ตัวบุคคล ผู้ขับขี่รถมีร่างกายอ่อนเพลีย ไม่มีสติพร้อมสำหรับการขับรถ

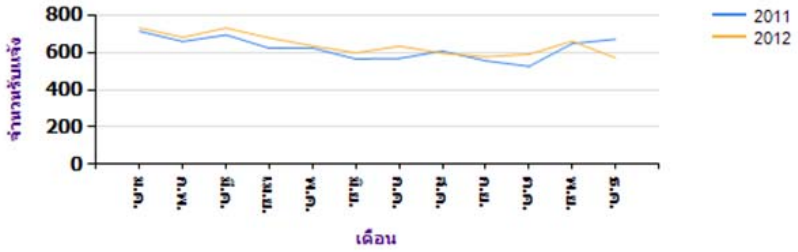
2. สิ่งแวดล้อม สภาพของแสงสว่าง ทัศนวิสัย ไม่ดีพอ

3. กฎหมาย มีบทลงโทษที่ไม่สามารถทำให้ผู้ขับขี่เกรงกลัว หรือ ความเข้มงวดยังไม่เพียงพอในการจับกุมเพื่อรักษากฎหมาย จึงอาจเป็นช่องทางทำให้เกิดการละเมิดกฎจราจร ซึ่งมักทำให้เกิดอุบัติเหตุ

จากภาพประกอบ 1-1 พบว่าอัตราการเกิดอุบัติเหตุจากท้องถนน มีปริมาณเพิ่มขึ้น และอัตราการเสียชีวิตก็มีแนวโน้มเพิ่มขึ้นตามไปด้วยดังภาพประกอบ 1-2 ปัญหาเหล่านี้ต้องได้รับการแก้ไขอย่างเร่งด่วน เป็นผลให้หน่วยงานที่เกี่ยวข้อง ต้องกำหนดเป็นนโยบายต่างๆ เพื่อควบคุมไม่ให้อัตราการสูญเสียเพิ่มมากขึ้น

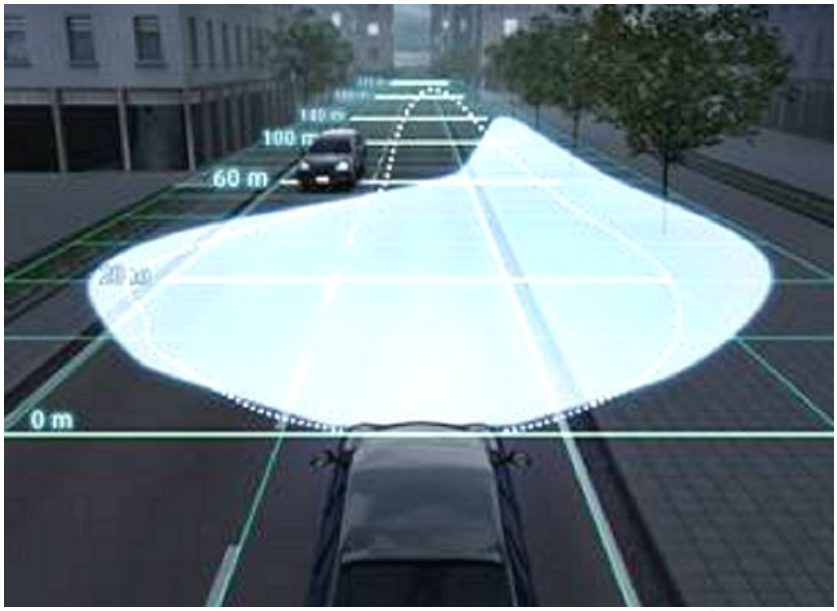


ภาพประกอบ 1-1 กราฟเปรียบเทียบการรับแจ้งอุบัติเหตุการจราจรทางบก ตามเดือน ปี 2011-2012 [1]

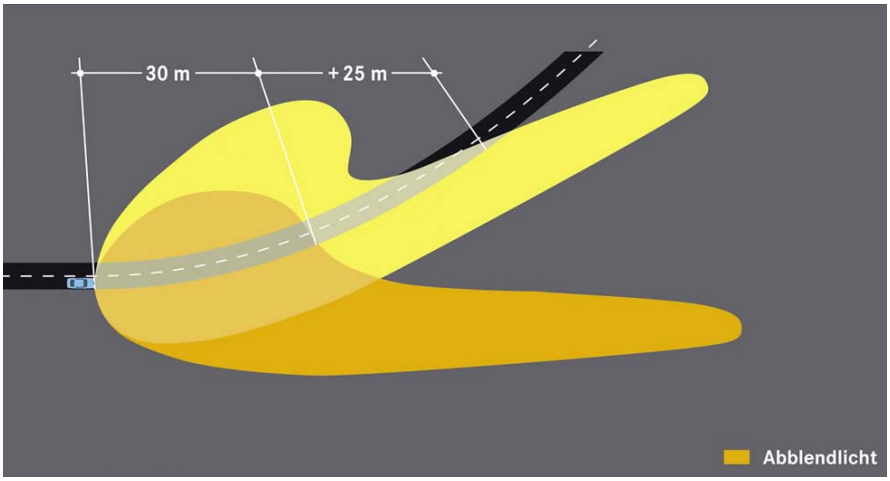


ภาพประกอบ 1-2 กราฟเปรียบเทียบจำนวนผู้เสียชีวิตจากอุบัติเหตุจากการจราจรทางบก ตามเดือน ปี 2011-2012 [1]

ปัญหาอุบัติเหตุบนท้องถนนที่มีสาเหตุมาจากทัศนวิสัยไม่เอื้อต่อการขับขี่ เป็นสิ่งที่สามารถป้องกันได้โดยการมีสติ และตั้งอยู่ในความไม่ประมาท อีกส่วนหนึ่งคือสามารถนำเทคโนโลยีเข้ามาช่วยเสริมศักยภาพการมองเห็น เพื่อเพิ่มความปลอดภัย และช่วยลดปัญหาที่อาจเกิดขึ้นได้อีกทางหนึ่ง ซึ่งปัจจุบันผู้ที่มีส่วนเกี่ยวข้องทางตรงอย่างเช่นบริษัทผู้ผลิตรถก็เป็นอีกหน่วยงานหนึ่งที่ได้เล็งเห็นถึงปัญหาที่เกิดขึ้นนี้ กระบวนการทางเทคโนโลยีต่างๆ จึงถูกคิดค้น วิจัย และพัฒนาขึ้น เพื่อเสริมสร้างความปลอดภัยให้ทั้งผู้ใช้รถและใช้ถนน เช่น



ภาพประกอบ 1-3 ระบบแสงไฟอัจฉริยะ ป้องกันแสงรบกวนรถที่วิ่งสวนเลน



ภาพประกอบ 1-4 ระบบแสงไฟอัจฉริยะปรับการส่องสว่างตามความโค้งถนน

ระบบแสงไฟอัจฉริยะ (Intelligent light system) เป็นระบบการจัดการไฟส่องสว่างที่เพิ่มทัศนวิสัยให้กับผู้ขับขี่ และยังป้องกันการไปทำลายทัศนวิสัยของรถที่สวนเลน ดังแสดงในภาพประกอบ 1-3 ระบบนี้ยังช่วยเพิ่มทัศนวิสัยขณะรถเข้าทางโค้ง ดังภาพประกอบ 1-4

ระบบมองภาพกลางคืน (Night vision system) เป็นอีกระบบหนึ่งที่ช่วยเพิ่มทัศนวิสัยในการใช้รถใช้ถนนตอนกลางคือ หรือในขณะที่แสงสว่างไม่เพียงพอ ช่วยให้สามารถมองเห็นสิ่งกีดขวางได้อย่างชัดเจนมายิ่งขึ้น โดยใช้เทคโนโลยีของแสงอินฟราเรด (Infrared, IR) สามารถจำแนกได้ 2 แบบ คือ Passive และแบบ Active โดยระบบแบบ Passive ใช้หลักการของแสงอินฟราเรดช่วงคลื่น 25-350 μm (Far-infrared, FIR) เพื่อตรวจจับการแผ่รังสีความร้อนจากวัตถุด้านหน้าของรถยนต์ ส่วนระบบแบบ Active ใช้หลักการสะท้อนของแสงอินฟราเรดช่วงคลื่น 0.7-5 μm (Near infrared, NIR) โดยแสงจะสะท้อนวัตถุที่อยู่เบื้องหน้าทำให้สามารถมองเห็นได้อย่างชัดเจนมายิ่งขึ้น ทั้งสองเทคโนโลยีที่กล่าวมานี้มีจุดประสงค์หลักคือ เพิ่มทัศนวิสัยของผู้ขับรถยนต์ในสภาพแวดล้อมที่แสงน้อยกว่าปกติ หรือ กลางคืน สาเหตุเนื่องมาจาก สถิติการเกิดอุบัติเหตุเกินครึ่งหนึ่ง เกิดจากสภาพทัศนวิสัยไม่เอื้อต่อการขับขี่

นอกจากนี้ความนิยมของการติดกล้องดิจิตอลหน้ารถ มีความแพร่หลายมากยิ่งขึ้น ซึ่งเหตุผลที่นิยมติดตั้งกล้องบันทึกเหตุการณ์ไว้หน้ารถเพื่อจะบันทึกสิ่งต่างๆ ที่เกิดขึ้น อาจใช้เป็นหลักฐาน หรือเป็นข้อมูลบันทึกการเดินทางเพื่อใช้ในธุรกิจเฉพาะด้าน แต่มีสิ่งหนึ่งที่สามารถใช้ประโยชน์จากกล้องหน้ารถได้คือ เป็นอุปกรณ์เตือนสติ หรือ เป็นอุปกรณ์ เพิ่มทัศนวิสัย ในกรณีที่สภาพแวดล้อมไม่เอื้ออำนวยต่อการมองเห็น

การแก้ปัญหาต่างๆ เหล่านี้โดยการพัฒนาระบบตรวจจับขอบถนน หรือการค้นหาช่องทางเดินรถโดยวิธีที่ผู้วิจัยได้นำเสนอนี้ จะมีส่วนช่วยแก้ปัญหาต่างๆ ที่กล่าวมา ทำให้ผู้ใช้รถใช้ถนนสามารถมองเห็นช่องทางเดินรถของตัวเองได้ชัดเจนมากยิ่งขึ้น เสมือนมีดวงตาที่ 3 คอยช่วยเหลือ หรือ มีการพัฒนาเป็นระบบเตือนสติของผู้ขับขี่เมื่อมีการขับรถที่ไม่ตรงช่องทางเดินรถของตัวเองซึ่งอาจจะก่อให้เกิดอุบัติเหตุได้ เทคโนโลยีที่สามารถประมวลผลได้อย่างทันทีนี้ถือเป็นหัวใจสำคัญ เนื่องจากรถที่วิ่งด้วยความเร็ว การทำงานและการตัดสินใจต่างๆ มีเวลาจำกัด ความถูกต้องของข้อมูลอาจจะมีสาระสำคัญน้อยกว่าความเร็วในการประมวลผล เนื่องจากเทคโนโลยีที่ได้กล่าวมานี้เป็นส่วนที่ช่วยในการอำนวยความสะดวกเท่านั้น ฉะนั้นผู้ที่ใช้เทคโนโลยีเองยังต้องมีการวิเคราะห์สิ่งที่เกิดขึ้นเองเป็นหลักควบคู่กันด้วย

ภาพประกอบ 1-5 เส้นขาวข้างถนน และเส้นเหลืองกลางถนน เป็นส่วนที่นำมาวิเคราะห์หาขอบของทางเดินรถ

หลักการที่ใช้ในการหาเส้นขอบถนนที่ผู้วิจัยได้นำมาพัฒนาคือ ใช้กระบวนการค้นหาเส้นตรงในภาพ เพื่อหาตำแหน่งของขอบถนน หรือ ช่องทางเดินรถ โดยมีวิธีการคือ ยึดถือ เส้นขาวของด้านข้างถนน และเส้นเหลืองกลางถนน ดังภาพประกอบ 1-5 เป็นตัวอ้างอิงขอบของช่องทางเดินรถ ซึ่งวิธีที่ทำให้เห็น เส้นขาวและเส้นเหลืองที่ชัดเจนนี้ ผู้ทำวิจัย ได้พัฒนาระบบ Mask filter ขึ้น ซึ่งเป็นส่วนสำคัญที่จะช่วยวิเคราะห์ขอบ และหาเส้นตรงต่อไป ซึ่งการพัฒนาส่วนสำคัญต่างๆ เหล่านี้ ผู้วิจัยได้พัฒนาลงใน FPGA เพื่อที่จะนำศักยภาพ ในด้านความเร็วของการประมวลผลมาใช้ ทำให้ระบบสามารถทำงานได้ทันเวลา ในขณะที่รถวิ่งด้วยความเร็ว

องค์ประกอบของการประมวลผลภาพดิจิตอลในปัจจุบันประกอบด้วย 2 ส่วนหลักๆ คือ ส่วนของการรับภาพ และส่วนของการประมวลผลภาพ ซึ่งปัญหาที่เกิดเมื่อนำระบบไปใช้งานจริง คือ การประมวลผลภาพต้องใช้เวลาในการวิเคราะห์นานกว่าส่วนของการรับภาพ ผลคือไม่สามารถทำงานได้ทันเวลาเมื่อให้ระบบทำงานในขณะที่รถกำลังเคลื่อนที่ด้วยความเร็ว จึงมีการนำเทคโนโลยีตัวประมวลผลสัญญาณดิจิตอล (Digital signal processor) ซึ่งเป็นอุปกรณ์ทางด้านสมองกลฝังตัว (Embedded system) เข้ามาใช้เพื่อช่วยให้การประมวลผลภาพสามารถทำงานได้เร็วมากยิ่งขึ้น แต่ความยืดหยุ่นในการทำงานน้อยกว่าและประมวลผลช้ากว่า เทคโนโลยี FPGA (Field Programmable Gate Array) ซึ่งมีความยืดหยุ่นในการเปลี่ยนโครงสร้างวงจรได้มากกว่าและสามารถประมวลผลได้เร็วกว่าด้วยสถาปัตยกรรมแบบขนาน

สำหรับในงานวิจัยฉบับนี้ได้นำเสนอการประมวลผลภาพเพื่อช่วยลดปัจจัยที่ทำให้เกิดอุบัติเหตุบนถนนโดยการค้นหาขอบช่องทางจราจรเพื่อเพิ่มทัศนวิสัยการมองเห็นให้ผู้ขับขี่ ซึ่งเป็นส่วนหนึ่งของการพัฒนาระบบเตือนสติผู้ขับขี่ เมื่อรถอยู่ในตำแหน่งที่อาจเกิดอันตราย โดยผู้วิจัยได้

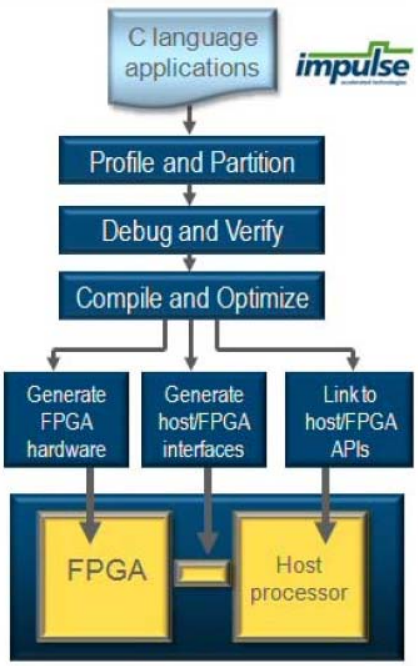
นำเทคโนโลยี FPGA เข้ามาใช้เพื่อปรับปรุงจุดด้อยของการประมวลผลภาพ ทำให้ระบบสามารถทำงานได้มีประสิทธิภาพสูงในขณะที่รถยนต์เคลื่อนที่ด้วยความเร็ว

1.2 การตรวจเอกสาร บทความ และงานวิจัยที่เกี่ยวข้อง

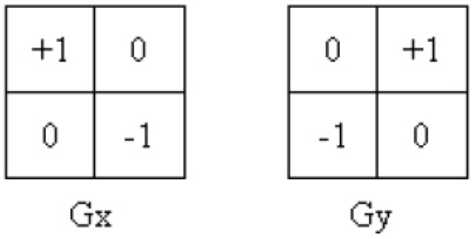
จากการศึกษางานวิจัยที่ผ่านมาทั้งในประเทศและต่างประเทศ ทางผู้วิจัยได้เก็บรวบรวมงานวิจัยที่เกี่ยวข้องดังนี้

1.2.1. การออกแบบและพัฒนาวงจรหาขอบภาพด้วยภาษาระดับสูง Impulse C [3]

บทความนี้ได้นำเสนอขั้นตอนการออกแบบวงจรสำหรับหาเส้นขอบภาพ โดยใช้ Impulse C ซึ่งมีโครงสร้างการทำงานดังแสดงในภาพประกอบ 1-6 สำหรับอัลกอริทึมที่นำมาใช้ในการหาขอบของภาพคือ อัลกอริทึมของ Roberts ด้วยเหตุผลที่เป็นอัลกอริทึมที่มีความซับซ้อนน้อยที่สุดและมีขนาดของ Kernel เล็กกว่าแบบอื่น ซึ่งแสดงใน ภาพประกอบ 1-7 ผลที่ได้จากการทดสอบโดยใช้ Impulse C Developer ปรากฏดังภาพประกอบ 1-7 ซึ่งสามารถหาขอบภาพได้อย่างชัดเจน



ภาพประกอบ 1-6 โครงสร้างการทำงานของ Impulse C Developer [3]

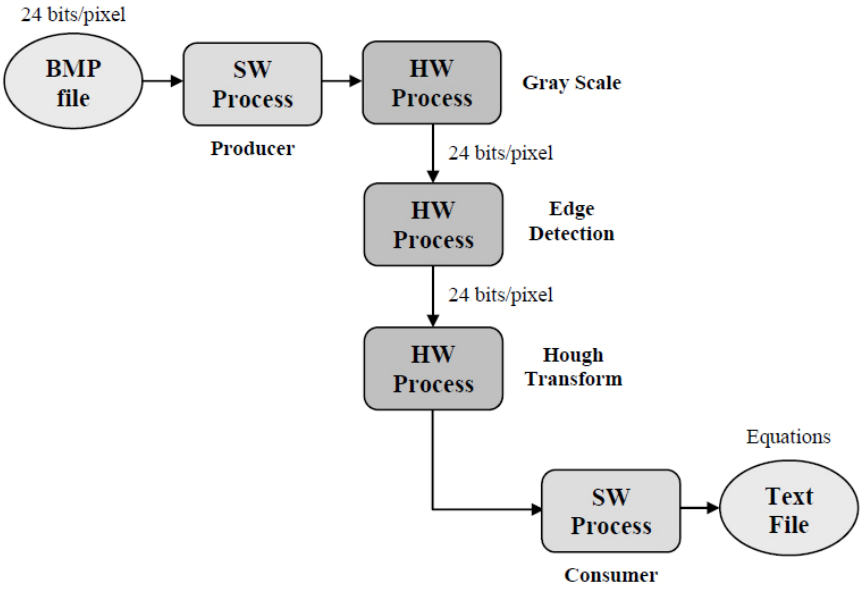


ภาพประกอบ 1-7 Kernel ของอัลกอริทึมหาเส้นขอบภาพ วิธีของ Roberts [3]

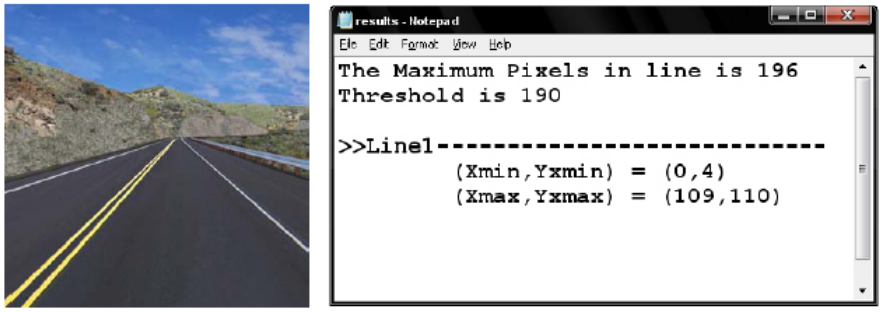
ภาพประกอบ 1-8 ผลลัพธ์จากการหาขอบภาพ โดยใช้ Impulse CoDeveloper [3]

1.2.2. Hardware/Software Co-design for Line Detection Algorithm on FPGA [6]

เป็นบทความที่ใช้ความสามารถของการออกแบบร่วมฮาร์ดแวร์และซอฟต์แวร์ ทำการหาเส้นตรงในภาพ เป็นการพัฒนาผ่านซอฟต์แวร์ Simulator ซึ่งช่วยวิเคราะห์การทำงานในด้านต่างๆ จากภาพประกอบ 1-9 เมื่อมีการนำรูปภาพเข้าสู่กระบวนการสังเคราะห์พบว่าผลลัพธ์สุดท้าย คือ สมการเส้นตรง ดังภาพประกอบ 1-10



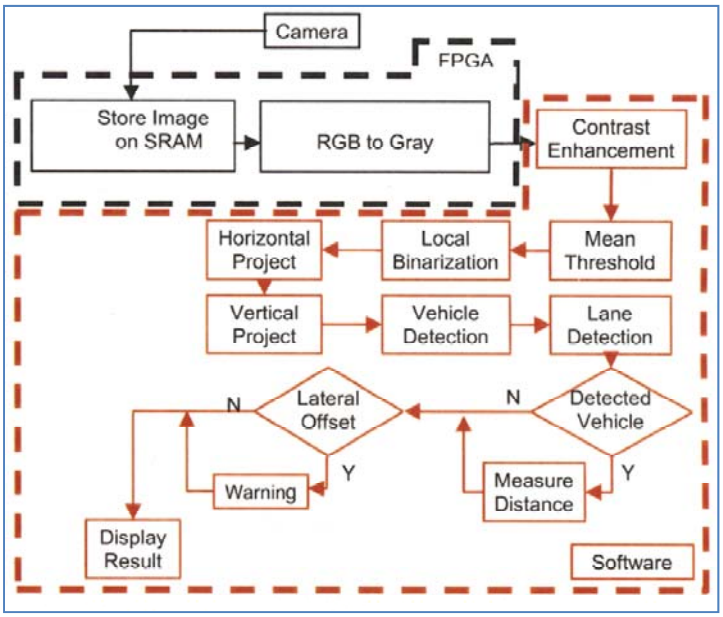
ภาพประกอบ 1-9 แผนภาพแสดงกระบวนการหาเส้นตรงในภาพ [6]



ภาพประกอบ 1-10 ผลลัพธ์ของกระบวนการหาเส้นตรงในภาพ [6]

1.2.3. Lane Detection System Based on Software and Hardware Co-design [4]

บทความนี้นำเสนอวิธีการหาช่องทางเดินรถ โดยการทำงานร่วมกันระหว่างซอฟต์แวร์และฮาร์ดแวร์ ซึ่งส่วนของฮาร์ดแวร์ใช้ FPGA เพียงสองส่วนคือ ส่วนของการรับภาพมาประมวลผลและส่วนของการแปลงสัญญาณจากภาพสี RGB เป็นภาพระดับขาวเทา (Gray scale) ส่วนอื่นนอกเหนือจากนี้จะถูกพัฒนาโดยใช้ซอฟต์แวร์ทั้งหมด ซึ่งผู้พัฒนาได้ใช้ บอร์ด Intel Xscale (PXA255, 400MHz)



ภาพประกอบ 1-11 ระบบการทำงานระหว่างฮาร์ดแวร์และซอฟต์แวร์ [4]

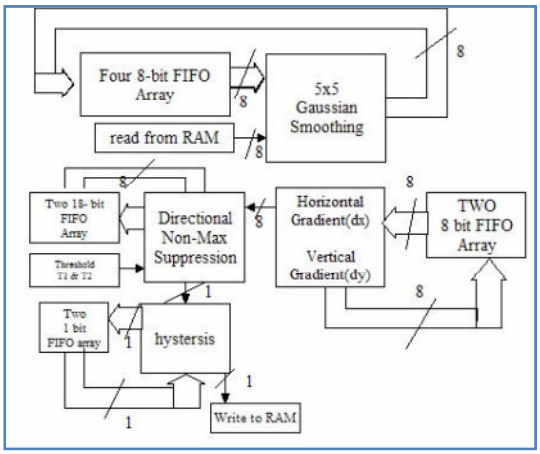
จากภาพประกอบ 1-11 การทำงานส่วนใหญ่จะอยู่ในฝั่งซอฟต์แวร์ซึ่งผู้พัฒนาได้ใช้ บอร์ด Intel เป็นตัว Run ซอฟต์แวร์

การทำงานในส่วนของซอฟต์แวร์ ผู้พัฒนาได้ใช้ภาษาซี ในการค้นหาช่องทางเดินรถ โดยทำการประมวลผลภาพที่ได้จากการแปลงเป็นภาพระดับขาวเทาจาก FPGA

1.2.4. Hardware Acceleration of Edge Detection Algorithm on FPGAs [5]

บทความนี้ได้เสนอแนวคิดที่จะเพิ่มความเร็วให้กับการประมวลผลภาพโดยใช้วิธีการสร้างวงจรในฮาร์ดแวร์ ซึ่งผู้วิจัยได้ทำการปรับปรุงวิธีการหาขอบภาพแบบ Canny ดังภาพประกอบ 1-11 โดยการพัฒนาขั้นตอนต่างๆในการหาขอบลงในฮาร์ดแวร์หลังจากที่ผลงานประสบความสำเร็จได้มีการสรุปเปรียบเทียบการทำงานให้เห็นว่าระหว่างพัฒนาลงฮาร์ดแวร์โดยตรงกับการใช้ซอฟต์แวร์ผลการทำงานใช้เวลาต่างกันมากน้อยเพียงไร

หลังจากพัฒนาระบบหาขอบภาพจนสมบูรณ์ ผู้พัฒนาได้ทดสอบความเร็วในการประมวลผลการทำงานระหว่างซอฟต์แวร์และฮาร์ดแวร์ที่ได้มีการพัฒนาขึ้น ผลปรากฏว่าเวลาในการทำงานโดยใช้ฮาร์ดแวร์เร็วกว่า ประมาณ 11 เท่า ในขณะที่ความถี่ของสัญญาณนาฬิกา (Clock) ที่ใช้ประมวลผล ช้ากว่าถึง 81 เท่า ซึ่งผลการทดลองแสดงในตาราง 1-1



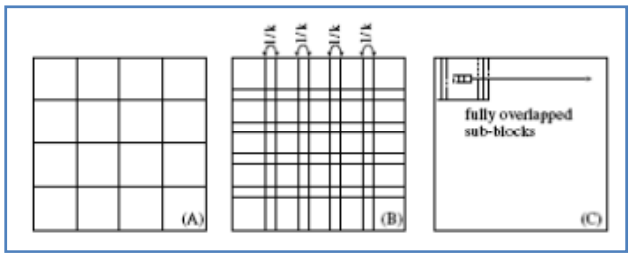
ภาพประกอบ 1-12 โครงสร้างการหาขอบภาพโดยวิธี Canny [5]

ตาราง 1-1 ผลการเปรียบเทียบการหาขอบภาพโดยวิธีของ Canny ระหว่างฮาร์ดแวร์บนบอร์ดทดลอง Xilinx Vertex-E กับซอฟต์แวร์ซึ่ง ทดลองบนบอร์ดทดลอง Pentium III [5]

บอร์ดทดลอง	Freq (MHz)	Time (ms)
Xilinx Vertex-E	16	4.2
Pentium III	1300	47

1.2.5. Real-Time Processing of Local Contrast Enhancement on FPGA [7]

บทความนี้ได้พัฒนาเทคนิค การปรับปรุงความคมชัด (Contrast enhancement) โดยการแบ่ง Histogram ออกเป็นส่วนย่อยๆ ดังภาพประกอบ 1-12 และทำการปรับปรุงความคมชัดของภาพ เป็นส่วนๆ ไป โดยการพัฒนาได้คำนึงถึงเรื่องความเร็วในการทำงาน จึงได้มีการออกแบบให้สามารถทำงานแบบขนานโดยใช้ FPGA เป็นเครื่องมือในการพัฒนา ซึ่งผลการทำงานได้แสดงในภาพประกอบ 1-14 ส่วนประสิทธิภาพการทำงาน และอัตราการใช้ทรัพยากรแสดงในตาราง 1-2



ภาพประกอบ 1-13 รูปแบบการแบ่งภาพออกเป็นส่วนๆ เพื่อปรับปรุงHistogram [7]

ภาพประกอบ 1-14 ผลลัพธ์จากการปรับปรุงแสงโดยการแบ่งปรับปรุงเป็นส่วนๆ [7]

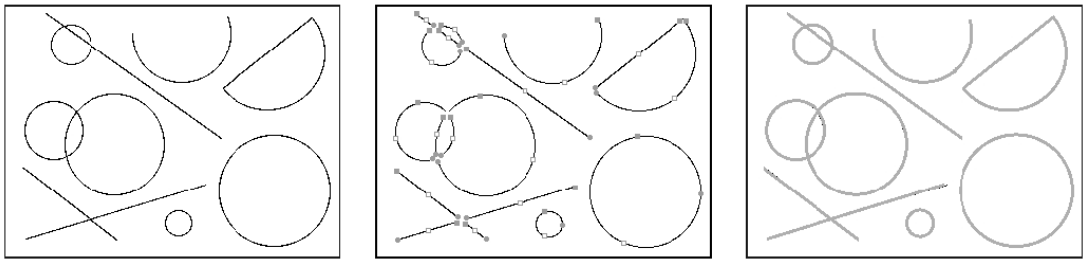
ตาราง 1-2 เปรียบเทียบการใช้ทรัพยากร และประสิทธิภาพในการทำงาน [7]

resource	non-unified	unified
LUTs	36123(26%)	38672(28%)
Block RAMs	192(67%)	128(44%)
Operational frequency	209.4MHz	197.4MHz
Performance	537.9 fps	253.6 fps

1.2.6. Hough Transform Modification [10]

เป็นบทความที่นำเสนอวิธีการปรับปรุง Hough transform ให้สามารถหาเส้นตรง เส้นโค้ง หรือวงกลมได้ง่ายขึ้น โดยการค้นหาจุดปลายหัวท้ายของแต่ละเส้น จากนั้นทำการวิเคราะห์ว่าเป็น

เส้นตรง เส้นโค้ง หรือวงกลม จุดมุ่งหมายของงานชิ้นนี้คือลดเวลาในการคำนวณ และลดพื้นที่ของหน่วยความจำ



ภาพประกอบ 1-15 ผลลัพธ์ของกระบวนการ Hough transform [10]

ภาพประกอบ 1-15 ภาพแรก คือภาพที่ใช้ทดสอบกระบวนการทำงาน ส่วน ภาพที่ 2 จะเป็นตำแหน่งที่กระบวนการทำการวิเคราะห์ เพื่อตรวจสอบว่าเป็นเส้นตรง หรือเส้นโค้ง ส่วนภาพสุดท้ายคือ ผลลัพธ์จากการค้นหา ปรากฏว่า ผลลัพธ์ที่ได้มีความแม่นยำค่อนข้างสูง เมื่อเทียบกับภาพที่นำมาวิเคราะห์

1.3 วัตถุประสงค์

1. คัดเลือกอัลกอริทึมการหาขอบที่เหมาะสำหรับพัฒนาใน FPGA เพื่อใช้ในกระบวนการค้นหาเส้นขอบช่องทางเดินรถ
2. เพื่อพัฒนาอัลกอริทึมค้นหาเส้น จากข้อมูลภาพที่ได้รับจากกล้องดิจิตอล ให้มีความเร็ว สามารถทำงานแบบทันเวลาได้

1.4 ขอบเขตของการวิจัย

1. ทำการทดลองเพื่อหาอัลกอริทึม หาขอบภาพที่เหมาะสมโดยพิจารณา 4 อัลกอริทึม คือ Canny, Roberts, Sobel และ Prewitt เท่านั้น เพราะงานวิจัยที่ผ่านมาได้พิสูจน์ให้เห็นแล้วว่า อัลกอริทึมทั้ง 4 มีโครงสร้างที่เหมาะจะพัฒนาใน FPGA เนื่องจากมีความซับซ้อนน้อย
2. การทดสอบภาพที่นำมาวิเคราะห์ จะเป็นภาพจริงที่ถ่ายมาจากกล้องดิจิตอล ซึ่งอัตราเฟรมประมาณ 30 fps ขนาดภาพ 640x480 พิกเซล ถ่ายภาพขณะรถมีความเร็ว 90 กิโลเมตร/ชั่วโมง
3. พัฒนาการทำงานลงในบอร์ด FPGA รุ่น Vertex-II Pro ซึ่งส่วนของการรับภาพ และจำลองภาพ เป็นหน่วยความจำที่จำลองขึ้นภายใน FPGA

1.5 ขั้นตอนและวิธีดำเนินงานวิจัย

1.5.1. ขั้นตอนการดำเนินงาน

a. รวบรวมข้อมูลเกี่ยวกับอัลกอริทึมหาขอบภาพแบบต่างๆ และกระบวนการ Hough transform
b. ทดลองเพื่อเลือกอัลกอริทึมหาขอบภาพ ที่เหมาะสำหรับพัฒนาบน FPGA โดยวิเคราะห์ผลผ่านโปรแกรม MATLAB
c. พัฒนา และทำการทดลอง Mask filter เพื่อช่วยเพิ่มประสิทธิภาพการทำงานในการหาขอบของภาพ วิเคราะห์ผล
d. ทดสอบกระบวนการหาเส้นตรงบนถนน เพื่อตรวจสอบความถูกต้องของการค้นหาเส้น
e. พัฒนาส่วนต่างๆ ลงบอร์ด FPGA รุ่น Xilinx Vertex-II Pro
f. ทดสอบการทำงาน ปรับปรุงแก้ไขระบบ
g. สรุป วิเคราะห์ผลการทำงาน และจัดทำรายงานฉบับสมบูรณ์

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้อัลกอริทึมที่สามารถพัฒนาลง FPGA และสามารถหาขอบของภาพได้
2. สามารถนำกระบวนการหาเส้นในภาพ ไปใช้ในการค้นหาช่องทางเดินรถได้
3. ได้ระบบการตรวจสอบช่องทางจราจรต้นแบบที่สามารถทำงานในขณะรถมีความเร็วสูงได้

1.7 ทรัพยากรที่ใช้ในการพัฒนาระบบ

ในการวิจัยนี้เป็นการพัฒนาระบบค้นหาช่องทางเดินรถ ซึ่งมีซอฟต์แวร์และอุปกรณ์ที่ใช้ในการออกแบบและทดสอบดังนี้

(1) คอมพิวเตอร์ส่วนบุคคล (Personal computer) ที่ใช้ในการทำงานวิจัยนี้เป็นคอมพิวเตอร์ที่มีหน่วยประมวลกลาง Core2 Duo อัตราเร็ว : 2.21 GHz และหน่วยความจำ : 3 GB

(2) กล้อง WEBCAM ICON IC-338 Resolution : 16 M pixels Frame rate : 30 fps

(3) บอร์ด FPGA รุ่น Vertex-II Pro

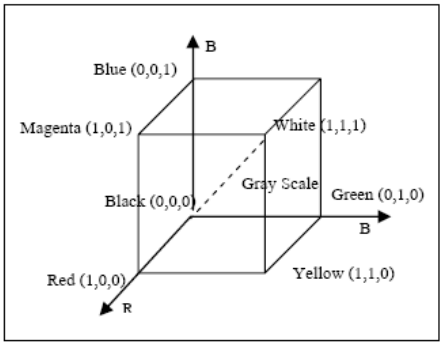
(4) ซอฟต์แวร์ Xilinx ISE Design Suite 10.1

(5) ซอฟต์แวร์ MATLAB

บทที่ 2
ทฤษฎีและหลักการ

2.1 การเปลี่ยนคุณสมบัติ จากภาพ RGB เป็นภาพระดับขาวเทา (Gray scale)

RGB ย่อมาจาก Red, Green และ Blue คือกระบวนการผสมสีจากแม่สี 3 สี คือ สีแดง สีเขียว และสีน้ำเงิน การใช้สัดส่วนของสี 3 สีต่างกัน จะทำให้เกิดสีต่างๆ ได้อีกมากมาย แสงสีต่างๆ เหล่านี้จะมีการรวมกันแบบ Additive ซึ่งโดยปกติจะนำไปใช้กับจอภาพแบบ CRT (Cathode ray tube)



ภาพประกอบ 2-1 Model Color Space [2]

ภาพประกอบ 2-2 การผสมแสง 3 สี [2]

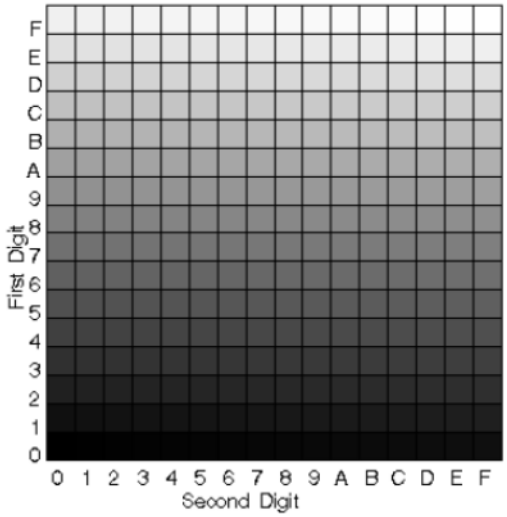
ในระบบพิกัด Color space ดังภาพประกอบ 2-1 แต่ละสีจะมีค่าตั้งแต่ 0 ถึง 1 โดย 0 หมายถึง สีนั้นมีความเข้มมากทำให้ดูมืด และ 1 หมายถึงสีนั้นมีความเข้มน้อย จึงดูสว่าง โดยทั่วไปบิตข้อมูลที่ใช้ในการแทนความเข้มของแม่สีแต่ละสี มี 256 ระดับ (0-255) จำนวน 8 บิต รวมแม่สีทั้งสามแล้วใช้ 8 x3 เท่ากับ 24 บิต ซึ่งสามารถใช้สร้างสีได้มากถึง 256 x 256 x 256 เท่ากับ 16,777,216 สี

การแปลงภาพสีเป็นขาวดำได้นั้นทำได้หลายวิธี เช่น การแปลงค่า RGB ให้เป็นค่าเฉลี่ยแล้วแทนลงไปในพิกเซลนั้นๆ ซึ่งจะได้ความลึกของภาพเท่ากับ 24 บิตเหมือนเดิม หรืออีกวิธีโดยการ

เปลี่ยนจากสี RGB เป็นระดับขาวเทา ซึ่งความลึกของภาพจะเหลือ 8 บิต โดยมีการคูณด้วยค่าคงที่ไปในแต่ละสีของ RGB โดยทั่วไปมักจะใช้ค่าดังสมการ ต่อไปนี้

$$Y = 0.299R + 0.587G + 0.114B \quad (2\text{-}1)$$

โดยที่ Y คือของความเข้มที่มีค่าอยู่ในช่วง 0-255 (สีดำ - สีขาว) ซึ่งผลที่ได้แสดงดังภาพประกอบ 2-3



ภาพประกอบ 2-3 ระดับขาวเทา (Gray scales) [2]

2.2 การหาขอบภาพ (Edge detection)

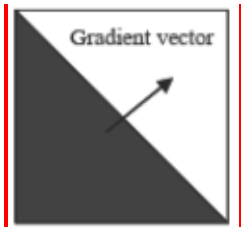
การหาขอบภาพ เป็นการหาเส้นรอบรูปที่อยู่ในภาพ ซึ่งขอบภาพเกิดจากความแตกต่างของความเข้มแสงจากจุดหนึ่งไปยังจุดหนึ่ง ซึ่งหากความแตกต่างมีมาก ขอบภาพจะยิ่งมีความชัดเจน ซึ่งวิธีการหาขอบภาพสามารถแบ่งได้เป็นสองกลุ่มหลักๆ คือ Gradient method และ Laplacian method โดยแต่ละวิธีมีรายละเอียดดังนี้

2.2.1 Laplacian method

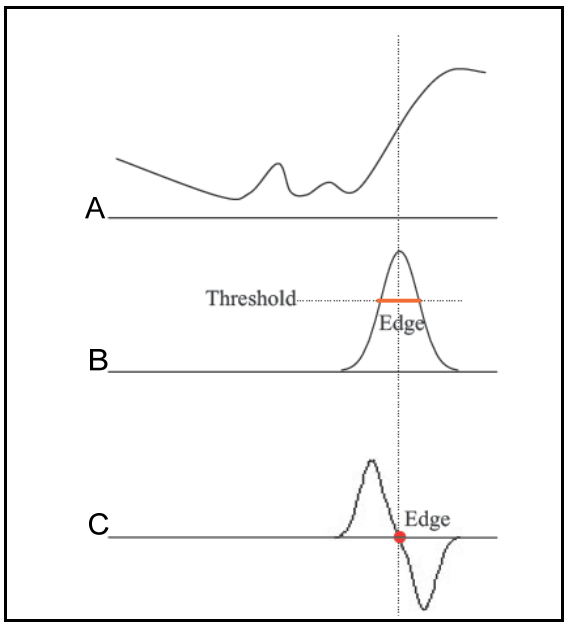
เป็นวิธีที่ใช้หลักการหาอนุพันธ์อันดับสอง โดยใช้จุดที่เป็น 0 เป็นขอบของภาพ อัลกอริทึมที่ใช้วิธีการนี้หาขอบได้แก่ Laplacian of Gaussian, Marrs Hildreth เป็นต้น

2.2.2 Gradient method

วิธีนี้จะหาขอบโดยการหาจุดต่ำสุดและจุดสูงสุดในรูปของอนุพันธ์อันดับหนึ่ง โดยจุดที่เป็นขอบ จะอยู่ในส่วนที่เหนือค่า Threshold อัลกอริทึมที่ใช้วิธีการนี้ในการหาขอบ ได้แก่ Prewitt, Sobel, Roberts, Canny เป็นต้น



ภาพประกอบ 2-4 การหาขอบภาพโดยวิธีเกรเดียนต์ [8]



ภาพประกอบ 2-5 การหาขอบ Gradient method และ Laplacian method [8]

จากภาพประกอบ 2-5 A เป็นสัญญาณของภาพโดยทั่วๆ ไป ซึ่งหากทำการหาอนุพันธ์อันดับหนึ่งแล้ว พิจารณากับค่า Threshold ค่าที่อยู่เหนือเส้นแดงในรูป B จะมองเห็นเป็นขอบภาพ ความชัดเจนของขอบจะมากน้อยแค่ไหนขึ้นอยู่กับ ขนาดของส่วนที่เหนือค่า Threshold ไปเรียกวิธีการหาขอบแบบนี้ว่าการหาขอบแบบ Gradient method ส่วนภาพ C จะหาขอบโดยการหาอนุพันธ์อันดับ2 ค่าที่เป็น 0 จะเป็นตำแหน่งของขอบ เรียกวิธีการหาขอบแบบนี้ว่า การหาขอบแบบ Laplacian method

เนื่องจากในงานวิจัยฉบับนี้ได้ทำการเพิ่มประสิทธิภาพการหาขอบภาพโดยพัฒนาอัลกอริทึมเป็นฮาร์ดแวร์การหาขอบภาพแบบ Gradient method จึงมีความเหมาะสมกว่า การหาขอบแบบ Laplacian method เพราะการหาขอบแบบ Laplacian จะต้องทำการหาอนุพันธ์อันดับสอง ซึ่งอาจสร้างความซับซ้อนให้กับกระบวนการทำงานได้

หลักการทำงานของการหาขอบภาพแบบ Gradient method คือ บริเวณขอบของวัตถุในภาพจะมีเกรเดียนต์สูง การพิจารณาขนาดของเกรเดียนต์ เปรียบเทียบกับ ค่า Threshold ที่กำหนดขึ้นเมื่อค่าของ $|\nabla P|$ มีค่ามากกว่าค่าอ้างอิง แสดงว่าจุดดังกล่าวคือขอบของวัตถุที่ปรากฏในภาพที่จุด P(x,y) การค้นหาขอบของวัตถุโดยใช้อนุพันธ์อันดับหนึ่ง เป็นวิธีแยกส่วนประกอบของภาพ และเมื่อความไม่ต่อเนื่องของค่าพิกเซลบริเวณรอยต่อระหว่างวัตถุกับพื้นหลัง และค่าอนุพันธ์ย่อยที่ไม่ต่อเนื่องตามทิศทางของเกรเดียนต์ ของแนวแกน x และ y กำหนดค่าได้ตามสมการ (2-2) และ (2-3)

$$\nabla xP(x,y) = P(x,y) - P(x-1,y) \tag{2-2}$$

$$\nabla yP(x,y) = P(x,y) - P(x,y-1) \tag{2-3}$$

ขนาดของเกรเดียนต์ ของ P(x,y) กำหนดได้จากการหาขอบภาพ ลักษณะของ เกรเดียนต์จะแตกต่างออกไปขึ้นอยู่กับผู้คิดค้น ซึ่งสามารถแสดงได้ดังต่อไปนี้

Operator	Row gradient	Column gradient
Pixel difference	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Roberts	$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Prewitt	$1/3\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$	$1/3\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Sobel	$1/4\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	$1/4\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

ภาพประกอบ 2-6 เกรเดียนต์ โอเปอเรเตอร์ [3]

2.2.2.1 การหาขอบภาพ โดยวิธีของ Robert

ในกรณีศึกษานี้จะยกตัวอย่างการหาขอบโดยใช้วิธีของ Roberts เพราะจากงานวิจัยพบว่าวิธีของ Roberts ให้ผลการทำงานดีที่สุด เมื่อนำไปพัฒนาบน FPGA การหาขอบด้วยวิธีนี้ใช้เทคนิค

การปรับปรุงขอบที่ไม่ต่อเนื่อง ให้ $a \in R^x$ เป็นภาพต้นฉบับ และขอบที่ได้ เป็น $b \in R^x$ วิธีการของ Roberts คือ

$$b(i,j) = \sqrt{(a(i,j) - a(i+1,j+1))^2 + (a(i,j+1) - a(i+1,j))^2} \quad (2\text{-}4)$$

ให้ S คือ Mask ของแนวแกน X

T คือ Mask ของแนวแกน Y

$$S = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad T = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

เมื่อทำการคำนวณหาขอบทุกพิกเซลของทั้งภาพแล้ว จะได้ขอบเส้นสีขาวซึ่งจะเป็นขอบของภาพ

ภาพประกอบ 2-7 ภาพที่ได้จากการของขอบด้วยวิธีของ Roberts [3]

2.2.2.2 การหาขอบของภาพ โดยวิธีของ Prewitt

วิธีการนี้จะคำนวณขอบที่เป็นเกรเดียนต์เวคเตอร์ของทุกจุดบนภาพที่เป็นภาพต้นฉบับ ขอบที่ผ่านการปรับปรุงแล้วจะมาจากเกรเดียนต์เวคเตอร์ Mask ที่ใช้แทนอนุพันธ์จะเกี่ยวข้องกับ x และ y ให้ $a \in R^x$ เป็นภาพต้นฉบับ และ a1,a2,a3,….,a7 เป็นค่าของแต่ละพิกเซล 8 จุดที่ตำแหน่ง (i,j) ตามทิศทางทวนเข็มนาฬิกา ดังภาพประกอบ 2-8

a_3	a_2	a_1
a_4	I(i,j)	a_0
a_5	a_6	a_7

ภาพประกอบ 2-8 ตำแหน่งของตัวแปร ในอัลกอริทึมหาขอบโดย Prewitt [3]

ให้

$$u = (a_5 + a_6 + a_7) - (a_1 + a_2 + a_3) \quad (2\text{-}5)$$

$$v = (a_0 + a_1 + a_7) - (a_3 + a_4 + a_5) \quad (2\text{-}6)$$

ขอบของภาพ Prewitt เป็น $b \in R^x$

ให้

$$b(i, j) = \sqrt{u^2 + v^2} \quad (2\text{-}7)$$

และทิศทางของขอบภาพ $d \in R^x$ คือ

$$d(i, j) = \tan^{-1} \frac{v}{u} \quad (2\text{-}8)$$

ให้ S คือ Mask ของแนวแกน x

T คือ Mask ของแนวแกน y

$$S = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \qquad T = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

ภาพประกอบ 2-9 ภาพที่ได้จากการหาขอบด้วยวิธีของ Prewitt [3]

2.2.2.3 การหาขอบของภาพ โดยวิธีของ Sobel

วิธีนี้ เป็นการหาขอบที่ไม่เป็นเชิงเส้น สามารถเปลี่ยนแปลงค่าความไม่ต่อเนื่องได้ตามการปรับปรุงขอบให้ $a \in R^x$ เป็นภาพต้นฉบับ และ a0, a1, a2 ,…, a7 แสดงถึงตำแหน่งของแต่ละพิกเซลทั้ง 8 จุด ทวนเข็มนาฬิกา

ให้

$$u = (a_5 + 2a_6 + a_7) - (a_1 + 2a_2 + a_3) \quad (2\text{-}9)$$

$$v = (2a_0 + a_1 + a_7) - (a_3 + 2a_4 + a_5) \quad (2\text{-}10)$$

ขนาดขอบภาพ Soble เป็น $m \in R^x$

ให้

$$m(i, j) = \sqrt{(u^2 + v^2)} \quad (2\text{-}11)$$

และให้ทิศทางของเกรเดียนต์ของ d คือ

$$d(i, j) = \tan^{-1} \frac{u}{v} \quad (2\text{-}12)$$

ให้ S คือ Mask ของแนวแกน x

T คือ Mask ของแนวแกน y

$$S = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \qquad T = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

ภาพประกอบ 2-10 ภาพที่ได้จากการหาขอบด้วยวิธีของ Sobel [3]

2.2.2.4 การหาขอบภาพโดยวิธีของ Canny

การหาขอบภาพโดยวิธี Canny ประกอบด้วยขั้นตอน 4 ขั้นตอน

- **การปรับภาพให้เรียบ ด้วยการทำ Gussian smoothing**

โดยกำหนด Kernel เป็นเมตริกซ์ ขนาด 3x3 หรือมีขนาดเท่ากับ 9 พิกเซล ผลของการปรับภาพหาได้จากสมการ (2-13)

$$S_{(i,j)} = G_{(i,j,\sigma)} \bullet I_{(i,j)} \tag{2-13}$$

กำหนดให้ $I_{(i,j)}$ คือ ภาพที่ต้องการหาขอบ

$G_{(i,j,\sigma)}$ คือ Gaussian Smoothing Filter

σ คือ ตัวควบคุมระดับของการทำ Smoothing

$\bullet$ คือ โอเปอเรชันการคูณ

- **การคำนวนค่า Gradiant**

ขั้นแรกปรับภาพ $I_{(i,j)}$ ให้มีความเรียบ ผลที่ได้คือ ค่าของภาพในฟังก์ชัน $S_{(i,j)}$ จากนั้นทำการหาค่า Gradiant ในแกน x และ y และกำหนดขนาดของอนุพันธ์อันดับหนึ่งของ $Px_{(i,j)}$ และ $Qy_{(i,j)}$ ตามสมการ

$$Px_{(i,j)} \approx \frac{(S_{(i,j+1)} - S_{(i,j)} + S_{(i+1,j+1)} - S_{(i+1,j)})}{2} \tag{2-14}$$

$$Qy_{(i,j)} \approx \frac{(S_{(i,j)} - S_{(i+1,j)} + S_{(i,j+1)} - S_{(i+1,j+1)})}{2} \quad (2\text{-}15)$$

นำค่า $Px_{(i,j)}$ และ $Qy_{(i,j)}$ ที่ผ่านการหาอนุพันธ์อันดับหนึ่งแปลงรูปจากพิกัดฉาก เป็นพิกัดเชิงขั้ว เพื่อหาขนาด และทิศทาง Gradiant โดยใช้สมการ (2-14) และ (2-15) และจะได้ขนาดและทิศทางดังนี้คือ

ขนาด คือ $M_{(i,j)} = \sqrt{P_x^2(i,j) + Q_y^2(i,j)}$ ทิศทางคือ $\theta_{(i,j)} = \tan^{-1}(\frac{Q_y(i,j)}{P_x(i,j)})$

สามารถหาค่ามุมโดยแทนค่าตัวแปรในฟังก์ชัน $\theta = \tan^{-1}(x,y)$

- **Non-maxima suppression**

โดยจุดที่เป็นขอบภาพต้องเป็นจุดที่ให้ค่าสูงสุดเฉพาะที่และเป็นทิศทางเดียวกับ Gradiant การค้นหาขอบภาพโดยใช้อนุพันธ์อันดับที่ 1 ทำให้ได้ขอบที่บางเพียง 1 พิกเซล แต่ภาพที่ได้หลังจากทำ Non-maxima suppression จะให้ค่าเป็น 0 ในทุกจุด ยกเว้นจุดที่เป็น Local maxima point ซึ่งจะยังคงเดิมไว้

- **Thresholding**

ซึ่งค่า Threshold จะถูกกำหนดขึ้นมา 2 ค่า ได้แก่ ค่า High threshold (T1) และ Low threshold (T2) โดยพิกเซลที่มีค่ามากกว่า T1 จะถูกกำหนดให้มีค่าเป็น 1 ซึ่งเป็นพิกเซลที่เป็นขอบภาพ แต่ถ้ามีค่าน้อยกว่า T2 จะถูกกำหนดค่าเป็น 0 ส่วนค่าที่อยู่ระหว่าง Threshold ทั้งสอง จะถูกปรับค่า เป็น 1 หรือ 0 ขึ้นอยู่กับพิกเซลที่อยู่รอบข้าง การหาขอบโดยวิธีนี้จึงทำให้เกิดขอบหนาและขอบบาง ซึ่งทำให้การหาขอบภาพให้ผลลัพธ์ที่ดี

2.3 Hough transform

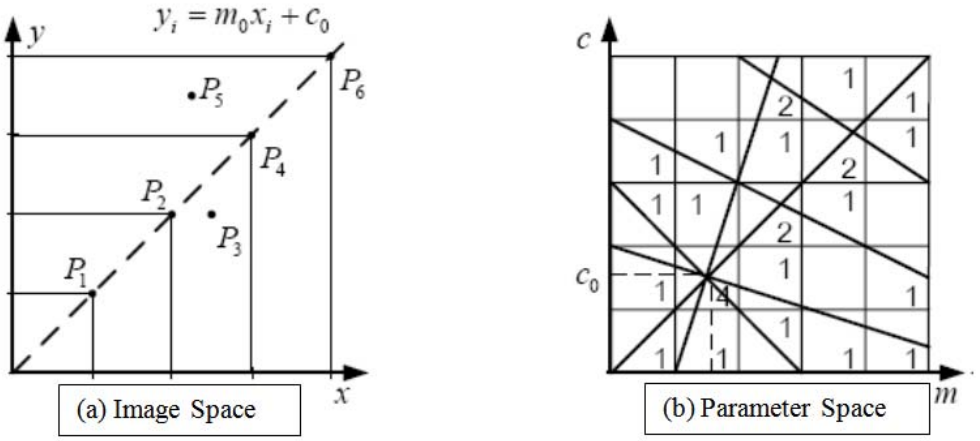
เป็นวิธีการหารูปทรงเรขาคณิตจากจุดที่นำมาพิจารณา โดยทั่วไป จะนำไปใช้ค้นหาเส้นตรง ซึ่งมีการทำงานคือจุดแต่ละจุดจะทำการหาเก็บค่าน้ำหนักว่าอยู่บนเส้นตรงใดบ้าง เมื่อทำการเก็บค่าน้ำหนักทุกจุดแล้ว เส้นตรงที่มีค่าน้ำหนักมากที่สุดจะถูกนำไปใช้ เช่น ต้องการหาสมการเส้นตรงที่ผ่านจุด *(x,y)* จะมีเส้นตรงจำนวนมากที่ผ่านจุดดังกล่าว เมื่อพิจารณาเส้นตรง

$$y_0 = mx_0 + c \quad (2\text{-}16)$$

เส้นตรงที่ผ่านจุด *(x,y)* มีค่าพารามิเตอร์คงที่ *คือ m และ c* ซึ่งค่า

$$c_0 = y - m_0 x \quad (2\text{-}17)$$

ดังนั้นจุดหนึ่งๆ จะโหวตให้กับสมการเส้นตรงที่มีพารามิเตอร์ต่างๆ กันได้หลายสมการ แต่จะมีเพียงสมการไม่กี่สมการเท่านั้นที่จะได้รับการโหวตมากที่สุด ซึ่งจะเป็นข้อบ่งชี้ได้ว่า จะมีเส้นตรงผ่านจุดดังกล่าวเหล่านั้น



ภาพประกอบ 2-11 การนับจำนวนเส้นตรงของ Hough transform [3]

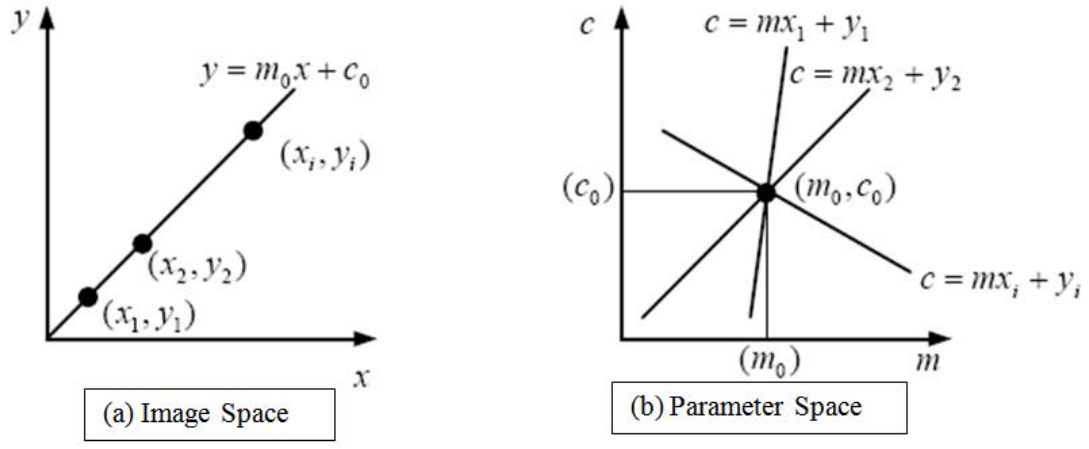
2.3.1 การแปลงรูปแบบ จาก Image space สู่ Parameter space

จากสมาการเส้นตรง

$$y_i = mx_i + c \tag{2-18}$$

เมื่อจุด (x_i, y_i) ที่ได้แสดงในภาพ (a) ค่าความชัน และจุดแกน y ของสมการเป็นค่าคงที่ (m_0, c_0) หรือเรียกว่าส่วนของ Parameter space

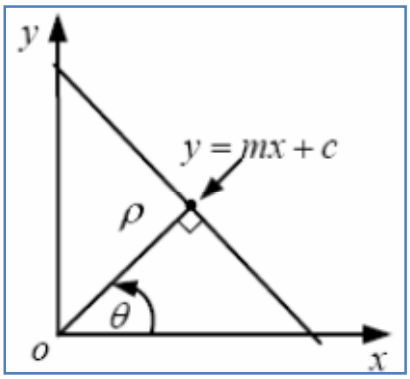
ดังนั้นการเปลี่ยนจาก Image space ไปสู่ Parameter space สมการของการเปลี่ยนแปลงคือที่จุด (m_0, c_0) จะเท่ากับ $c_i = y - mx_i$ ดังแสดงในภาพ (b)



ภาพประกอบ 2-12 การแปลงรูประหว่าง Image space กับ Parameter space [3]

จากภาพประกอบ 2-12 (a) มีจำนวนจุดทั้งสามจุดที่เส้นตรงที่มีค่าของความชันและจุดตัดแกน y ที่ตำแหน่ง (m_0, c_0) ดังนั้น เมื่อพิจารณาในรูปแบบของ Parameter space ที่จุด (m_0, c_0) ก็จะมีเส้นตรงที่เกิดขึ้นจากสมการเส้นตรงได้ทั้งหมดสามสมการ ที่ลากผ่านจุดดังกล่าว

ดังนั้นเมื่อกำหนดจุด *(x,y)* บนระนาบของ Image space และทำการเปลี่ยนเป็นระนาบ Parameter space หรือเรียกว่า Hough space ภาพประกอบ 2-13



ภาพประกอบ 2-13 ความสัมพันธ์ระหว่างค่าของเวกเตอร์ ρ กับจุดของเส้นตรง [3]

ความสัมพันธ์ระหว่างการเปลี่ยนรูปแบบคือที่จุด (ρ, θ) บน Parameter space จะเป็นจุดที่อยู่บนสมการเส้นตรงที่ลากผ่านจุด *(x,y)* เมื่อพิจารณาค่าของเวกเตอร์ ρ ที่ตั้งฉากกับจุดดังกล่าวและทำมุมกับแกน x เท่ากับมุม θ ดังนั้นการแปลงรูปจาก Image space ไปสู่ Parameter space สามารถหาค่าของเวกเตอร์ ρ ได้จากสูตร

$$\rho = x\cos\theta + y\sin\theta \tag{2-19}$$

เมื่อกำหนดให้ ρ คือ ระยะที่วัดจากจุดกำเนิดไปตั้งฉากกับเส้นตรง

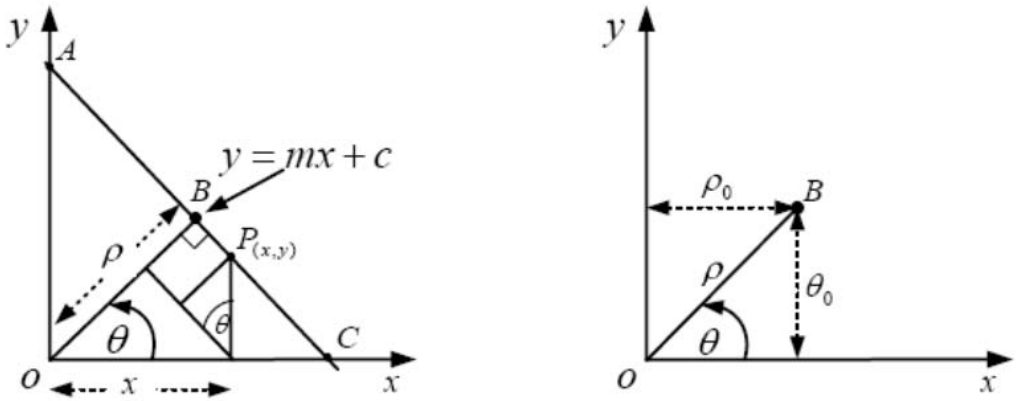
θ คือ ค่าของมุมระหว่าง เวกเตอร์ ρ กับแกน x

2.3.2 การกำหนดเส้นโค้งโดยใช้ Hough transform ใน 2 มิติ

หลักการของ Hough transform เพื่อการค้นหาองค์ประกอบของภาพสองมิติที่มีส่วนเส้นตรงปรากฏในภาพนั้น เป็นการหาเส้นตรงจากฟังก์ชันของสมการ (2-20)

$$f(x, y, \rho_0, \theta_0) = x\cos\theta_0 + y\sin\theta_0 \tag{2-20}$$

และกำหนดให้จุดของภาพสองมิติมีค่าเท่ากับ (ρ_0, θ_0) เมื่อ ρ_0 ระยะที่วัดจากจุดกำเนิดไปตั้งฉากกับเส้นตรง และจุด θ_0 เป็นค่าของมุมระหว่างเวกเตอร์ ρ กับแกน x แสดงในภาพประกอบ 2-14



ภาพประกอบ 2-14 การหาเส้นตรงของภาพสองมิติโดยใช้ Hough transform [3]

วิธีการของ Hough transform คือจากสมการ (2-19) มีการกำหนดจุด (x_0, y_0) ของ Image space ดังนั้นค่าของ ρ จะเท่ากับสมการ (2-21)

$$\rho = \sqrt{x_0{}^2 + y_0{}^2}\left(\frac{x_0}{\sqrt{x_0{}^2 + y_0{}^2}}\cos\theta + \frac{y_0}{\sqrt{x_0{}^2 + y_0{}^2}}\sin\theta\right) \tag{2-21}$$

$$\rho = r_0(\cos\alpha_0\cos\theta + \sin\alpha_0\sin\theta) \tag{2-22}$$

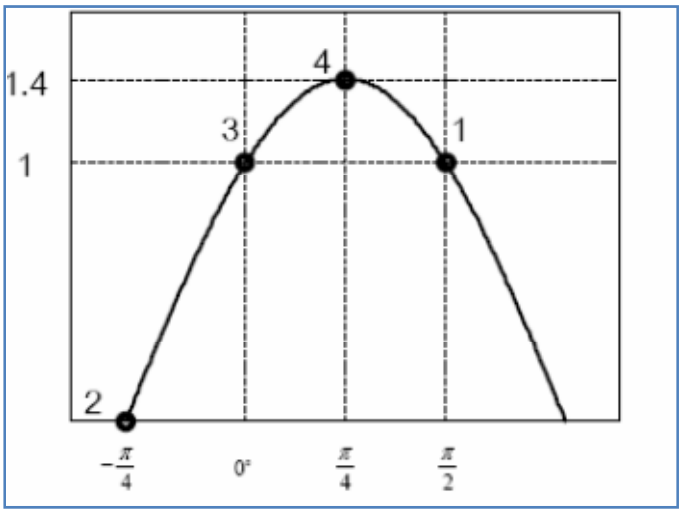
กำหนดให้ $r_0 \cong \sqrt{x_0^2 + y_0^2}$ และ $\alpha_0 \cong \tan^{-1}\left(\frac{y_0}{x_0}\right)$ และเมื่อแทนค่าของสมการ(2-20) จะได้ค่าของ ρ ตามสมการ (2-22)

การแปลงรูปของ Hough transform พบว่า $\rho = x\cos\theta + y\sin\theta$ ของจุด (x_0, y_0) ใน Image space เป็นการแปลงรูปแบบไปสู่เส้นโค้งรูป ซายด์ ใน Parameter space ทิศทางทวนเข็มนาฬิกา และจุด (ρ_0, θ_0) ของรูปเส้นโค้งรูปซายด์นี้ แสดงให้เห็นแทนเส้นตรงที่ลากผ่านจุด

(x_0, y_0) ใน Image space เมื่อทดลองแทนค่า ρ และ θ ตามตารางต่อไปนี้ ซึ่งผลที่ได้จะแสดงในภาพประกอบ 2-15

ตาราง 2-1 ค่าของ θ และ ρ

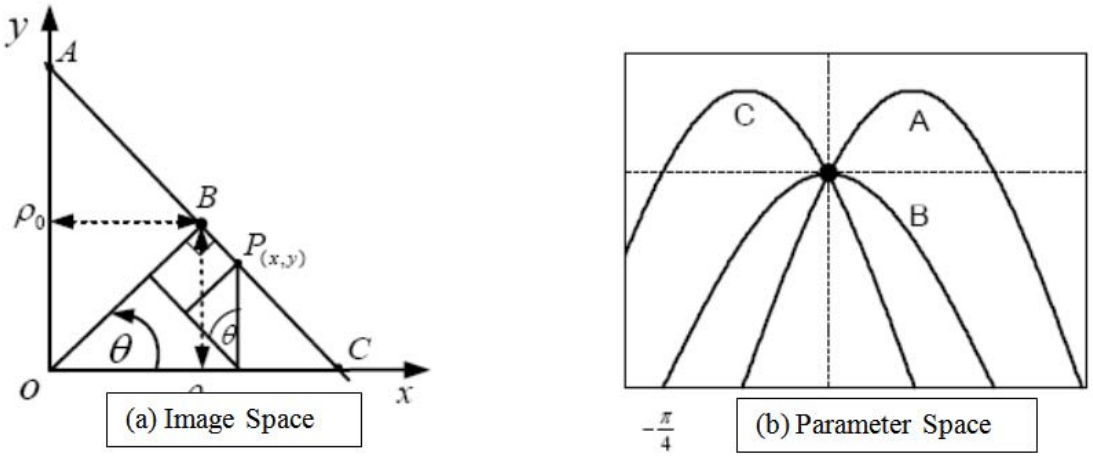
	จุดที่1	จุดที่2	จุดที่3	จุดที่4
θ	1.57	-0.785	0	0.785
ρ	1	0	1	1.414



ภาพประกอบ 2-15 ตัวอย่างการทำ Hough transform [3]

เมื่อทั้ง 4 จุดเกิดขึ้นใน Hough space ค่าของจุด 2 3 4 และ 1 ในภาพประกอบ 2-16 (b) จะมีค่าตรงกันในแต่ละเส้นของในภาพประกอบ 2-16 (a) Image space ลักษณะของภาพแบบการแปลงกลับของ Hough transform ค่าของจุด (ρ_0, θ_0) ใน Parameter space สามารถอธิบายได้ด้วยบทกลับของการแปลงรูปจาก Spatial domain ไปสู่การแทนค่าของเส้นตรงและอธิบายได้ด้วยสมการ (2-23)

$$p = x\cos\theta + y\sin\theta - \sigma_0 = 0 \qquad (2\text{-}23)$$



ภาพประกอบ 2-16 การแปลงค่าของ Hough space เมื่อมีการเปลี่ยนจุดในเส้นตรง [3]

จากภาพประกอบ 2-16 ตัวอย่างของ 3 จุดของตำแหน่ง *A(0,2) B(1,1)* และ *C(2,0)* ของทุกจุดบนเส้นตรง จุดดังกล่าวนี้เป็นจุดที่ตรงใน Hough space ในความเป็นจริงแล้วจุดเป็นค่าหนึ่งของสมการเส้นตรงที่สองและสามและมีค่าสอดคล้องกับเส้นโค้งที่ร่วมอยู่ในจุดของค่าเวกเตอร์ ρ เท่ากับ 1.414 และค่าของมุม θ เท่ากับ 0.79 เมื่อนำค่าของ (ρ,θ) แทนค่าในสมการ (2-23) ได้ดังนี้

$$1.414 = x\cos(0.79) + y\sin(0.79) \tag{2-24}$$

ดังนั้น ค่าของ

$$x + y = 2 \tag{2-25}$$

เมื่อเราทำ Hough transform แล้วจุดที่ *P(x,y)* ที่ปรากฏอยู่ในภาพจะมีเส้นตรงจำนวนมากมายที่ลากผ่านได้ ดังนั้นวิธีการของ Hough transform คือการพิจารณาว่าจุด *P(x,y)* ดังกล่าวมีจำนวนของเส้นตรงที่ลากผ่านจุดนี้จำนวนเท่าไหร่ และถ้าพิจารณาใน Hough space ก็จะมองเห็นเป็นจุดที่มีค่าของเส้นโค้งรูปซายด์ตัดผ่าน แสดงว่าจุดนั้นคือจุดเด่นของเส้นตรงที่ผ่านจุด *P(x,y)* มีค่ามากที่สุด จุดดังกล่าวมีค่าระยะห่างจากจุดกำเนิดมากที่สุดก็ต่อเมื่อเวกเตอร์ ρ ตั้งฉากกับเส้นตรงที่ผ่านจุด *P(x,y)* ดังกล่าวนั่นเอง

บทที่ 3
การออกแบบระบบตรวจจับช่องทางเดินรถ

จากแนวคิดในการหาขอบของถนน หรือการหาช่องทางจราจรเพื่อช่วยเพิ่มความปลอดภัยในการขับรถบนถนน ปัจจุบันนวัตกรรมนี้ยังไม่ได้ปรับใช้ให้เกิดผลเป็นรูปธรรม เนื่องจากระบบต้องการการตอบสนองที่รวดเร็ว ประมวลผลสัญญาณในขณะที่รถมีความเร็วสูง จากข้อจำกัดดังกล่าว เป็นเหตุผลที่สำคัญและจำเป็นในการพัฒนาระบบ อีกทั้งแนวโน้มการติดตั้งกล้องภายในรถยนต์ ได้รับความนิยมเพิ่มมากขึ้น แต่กล้องเหล่านั้นทำหน้าที่เพียงบันทึกการเดินทางเก็บลงสู่หน่วยความจำ ซึ่งไม่ได้นำมาใช้ให้เกิดประโยชน์ในด้านความปลอดภัยขณะขับรถ โดยจะให้ประโยชน์ในด้านหลักฐานประกอบหลังเกิดอุบัติเหตุเพียงเท่านั้น หากนำอุปกรณ์เหล่านี้มีพัฒนาให้มีส่วนช่วยเพิ่มความปลอดภัย จะทำให้อัตราการเกิดอุบัติเหตุบนถนน ลดลงอย่างแน่นอน

จากข้อจำกัดที่ได้กล่าวไว้เบื้องต้น ผู้วิจัยจึงนำเสนอการประมวลผลภาพเพื่อหาเส้นขอบถนนโดยใช้ FPGA เป็นส่วนช่วยในการเพิ่มความเร็ว ทำให้เวลาในการประมวลผลสัญญาณลดลง ระบบสามารถทำงานได้อย่างมีประสิทธิภาพแม้ขณะรถวิ่งด้วยความเร็วสูง ซึ่งการออกแบบอัลกอริทึมภายใน FPGA ต้องคำนึงถึงพื้นฐานของอัลกอริทึมค้นหาเส้นทั่วไปเป็นอันดับแรก ซึ่งผู้วิจัยได้ใช้โปรแกรม MATLAB เป็นเครื่องมือช่วยในการวิเคราะห์ จากนั้นจึงปรับปรุงอัลกอริทึมดังกล่าวให้สามารถทำงานภายใน FPGAได้

3.1 ภาพรวมของการออกแบบระบบ

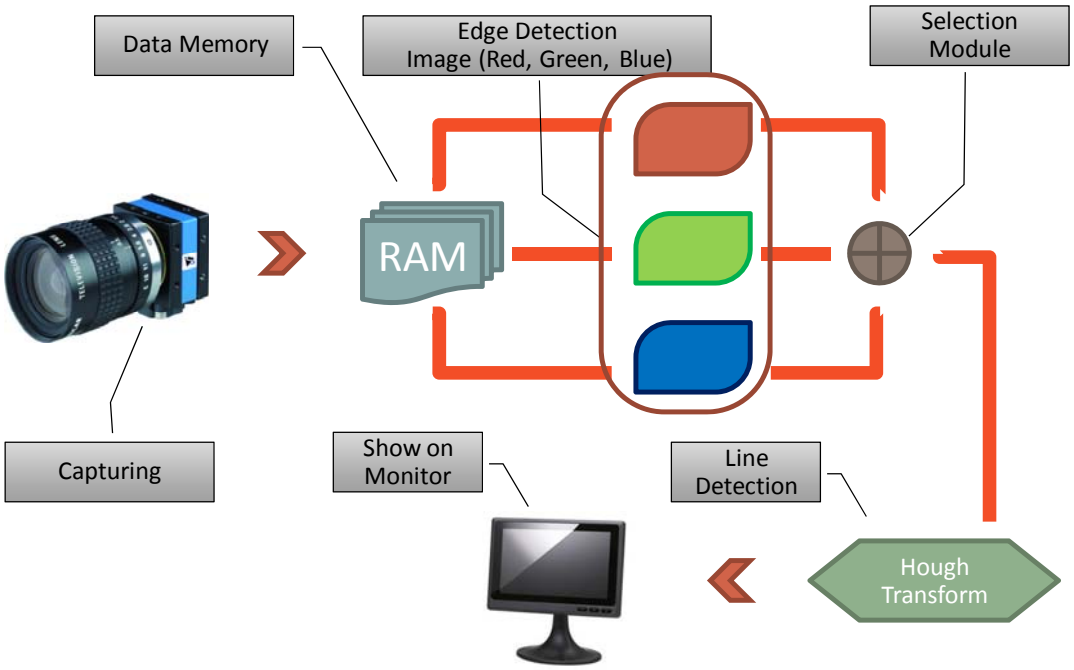
การหาเส้นตรงบนถนนเป็นหนึ่งในกระบวนการประมวลผลภาพ ระบบการทำงานมีองค์ประกอบ 3 ส่วน คือ กล้องดิจิตอลเป็นส่วนรับภาพ ส่วนประมวลผลภาพ และส่วนการแสดงผล

กล้องดิจิตอล (Digital camera) เป็นเพียงอุปกรณ์ที่รับสัญญาณภาพจากวัตถุหน้ารถ เพื่อส่งต่อไปยังส่วนของการประมวลผล ซึ่งส่วนสำคัญที่มีผลต่อระบบการหาเส้นเมื่อรถมีความเร็วสูงมากยิ่งขึ้น คือ อัตราเฟรม (Frame rate) เพราะหาก ค่าอัตราเฟรมน้อยเกินไป จะทำให้ระยะทางระหว่างเฟรม ไม่ได้ถูกหาตำแหน่งของเส้น ซึ่งอาจส่งผลต่อความต่อเนื่องของการหาเส้นขอบถนนได้

ส่วนของหน่วยประมวลผลภาพ (Image processor) ทำหน้าที่วิเคราะห์สัญญาณภาพที่ได้รับจากกล้อง เพื่อค้นหาเส้นขอบถนนตามอัลกอริทึมที่ได้กำหนดไว้ การทำงานในส่วนนี้จำเป็นต้องมีความเร็วในการประมวลผลเป็นอย่างมาก เนื่องจากภาพที่ได้รับจากกล้องจะถูกส่งมาอย่างรวดเร็วและต่อเนื่อง ยกตัวอย่างเช่น หากกล้องที่นำมาใช้มีอัตราเฟรมอยู่ที่ 30 fps แสดงว่า ใน 1วินาที ภาพจะถูกส่งมาประมวลผล 30 ภาพ ซึ่งเป็นหน้าที่ของหน่วยประมวลผล ที่ต้องดำเนินการให้เสร็จ โดย

อัตราการประมวลผลภาพต้องห้ามเกิน 30 มิลลิวินาที ต่อหนึ่งภาพ ระบบจึงสามารถทำงานได้ทันเวลา

ส่วนการแสดงผล ทำหน้าที่เพียงแสดงผลลัพธ์จากการประมวลผล ซึ่งมีได้หลายรูปแบบ เช่นแสดงผลรูปภาพผ่านทางหน้าจอ หรือหากมีการพัฒนาใช้กับระบบการแจ้งเตือนต่างๆ ก็สามารถทำได้โดยไม่จำเป็นต้องใช้หน้าจอในการแสดงผล เป็นต้น



ภาพประกอบ 3-1 ภาพรวมระบบค้นหาเส้นภายในภาพ

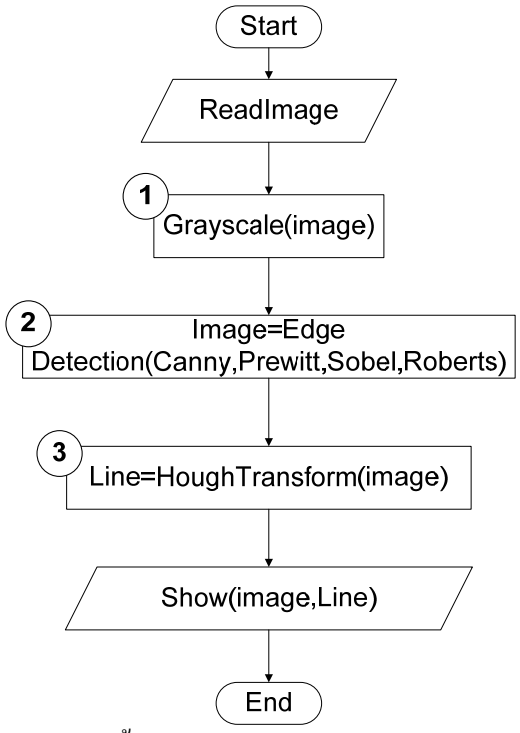
ภาพรวมของระบบจึงประกอบด้วยส่วนต่างๆ ดังภาพประกอบ 3-1 โดยโมดูลกล้องจะทำหน้าที่ในการจับภาพ และส่งสัญญาณไปพักไว้ใน RAM ซึ่งอยู่ภายใน FPGA จากนั้น ระบบจะทำการแยกสีภายในภาพออกเป็น 3 สี ได้แก่ สี แดง สีเขียว สีน้ำเงิน และทำการหาขอบภาพของแต่ละสี จากนั้นภาพที่ได้จะถูกส่งไปยังกระบวนการหาเส้นตรงด้วย Hough transform ซึ่งภาพที่นำมาวิเคราะห์จะต้องเลือกเอาขอบภาพใดภาพหนึ่ง จากทั้ง 3 สี โดยผู้วิจัยจะสามารถกำหนดผ่านทางโมดูล Selection และส่งสัญญาณภาพออกมาวิเคราะห์หาเส้นตรงก่อนแสดงผลทางหน้าจอเป็นลำดับสุดท้าย

ในงานวิจัยฉบับนี้ มุ่งเน้นพัฒนาระบบโดยใช้ FPGA เป็นส่วนประมวลผลสัญญาณภาพ ดังนั้นจุดมุ่งหมายของการพัฒนาระบบค้นหาเส้นจะเป็นการวิเคราะห์เพื่อคัดเลือกอัลกอริทึมที่เหมาะสมที่สุดสำหรับพัฒนาบน FPGA โดยระหว่างทำการวิเคราะห์ ผู้วิจัยจะทำการบรรจุภาพที่

ต้องการหาเส้นตรงไว้ใน RAM เพื่อความสะดวกในการวิเคราะห์ และผู้วิจัยจะทำการวิเคราะห์ผลการหาขอบภาพที่เหมาะสมเพื่อพัฒนาบน FPGA โดยคำนึงถึง ความเร็วในการประมวลผล ความซับซ้อนของอัลกอริทึม และความถูกต้องของการหาเส้นตรง ระบบจะแสดงผลการหาขอบภาพผ่านทางหน้าจอแสดงผล ส่วนขั้นตอนการหาเส้นตรง โดย Hough transform ผู้วิจัยไม่ได้มีการพัฒนาลง FPGA เนื่องจากอัลกอริทึมดังกล่าวมีการทำงานที่ซับซ้อน และเกินขอบเขตของงานวิจัยนี้

3.2 กระบวนการหาเส้นตรงในภาพ

หัวข้อนี้เป็นขั้นตอนการคัดเลือกอัลกอริทึมหาขอบภาพที่เหมาะสมสำหรับพัฒนาบน FPGA ผ่านโปรแกรม MATLAB โดยคัดเลือกจาก 4 อัลกอริทึม ได้แก่ Canny, Prewitt, Sobel และ Roberts



ภาพประกอบ 3-2 ขั้นตอนการหาเส้นโดยโปรแกรม MATLAB

ระบบการค้นหาเส้นตรงเป็นอัลกอริทึมในส่วนของ หน่วยประมวลผลภาพมีขั้นตอนการค้นหาเส้นแสดงดังภาพประกอบ 3-2

หมายเลข ① เป็นการแปลงภาพจากภาพสี เป็นภาพระดับขาวเทา (Gray scale) วัตถุประสงค์เพื่อเปลี่ยนจากภาพที่มีการผสมกันของ 3 สี มาอยู่ในโหมดสีเดียวที่ต่างกันเพียงความเข้มของสี

หมายเลข ② เป็นกระบวนการหาขอบของภาพ ซึ่งทำการทดสอบ 4 อัลกอริทึมที่ได้กล่าวมาแล้ว เพื่อคัดเลือกวิธีที่เหมาะสมที่สุด

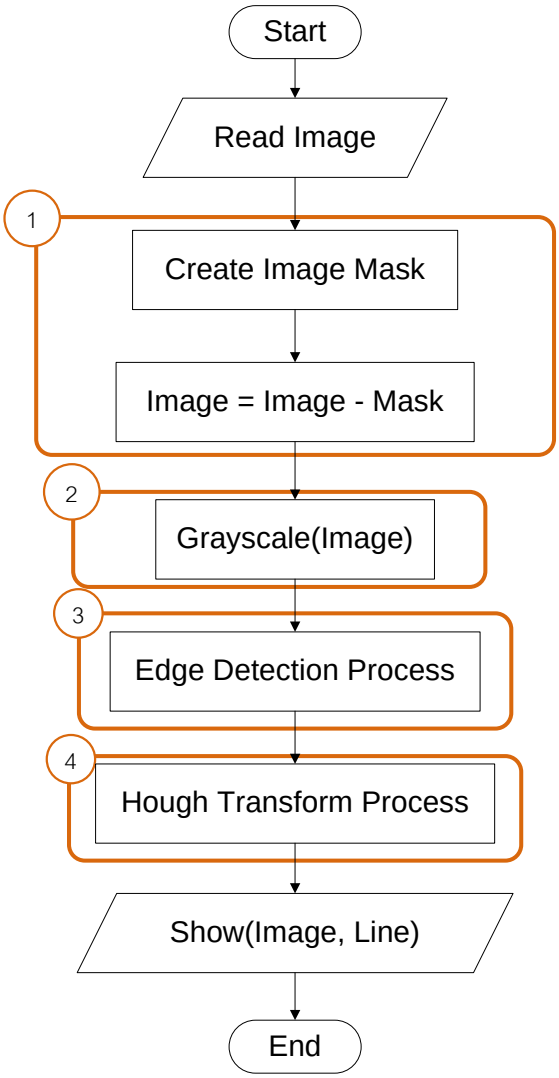
หมายเลข ③ เป็นกระบวนการหาเส้นตรงภายในภาพ ซึ่งในการทดสอบกับโปรแกรม MATLAB มีการเก็บข้อมูลสำหรับตรวจสอบความถูกต้องของเส้นตรง เพื่อเป็นข้อมูลใช้วิเคราะห์ต่อไป

3.3 การปรับปรุงความแม่นยำในการหาเส้นตรงด้วย Mask filter

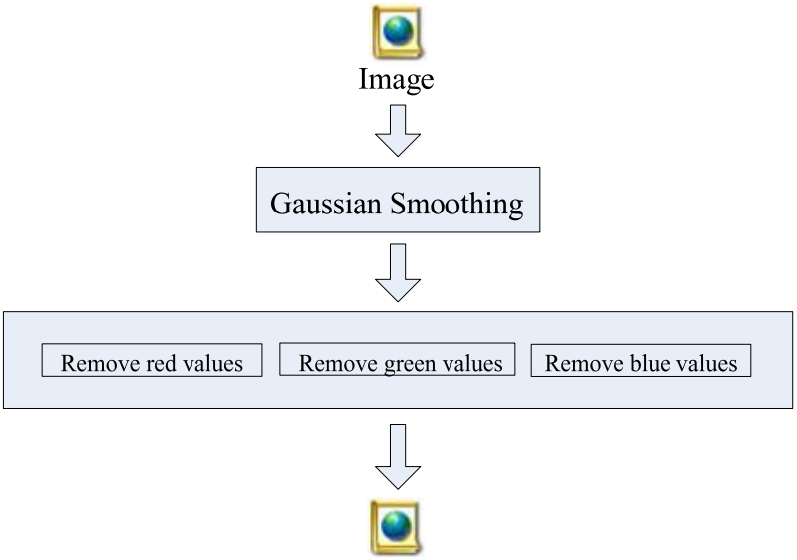
Mask filter เป็นกระบวนการหนึ่งที่ผู้วิจัยได้คิดค้นและพัฒนาขึ้น เพื่อเป็นเครื่องมือช่วยให้การหาขอบของภาพมีประสิทธิภาพมากยิ่งขึ้น เพราะสามารถลดเวลาประมวลผลในกระบวนการ Hough transform โดย Mask filter ประกอบด้วย Mask image ซึ่งเป็นเสมือนวงจรกรอง (Filter) ที่ทำหน้าที่เป็นตัวกรองสัญญาณภาพ โดยแนวคิดของวิธีนี้คือตัดส่วนที่ไม่จำเป็นต้องประมวลผลออก เป็นการลดภาระการทำงานของหน่วยประมวลผลอีกวิธีหนึ่ง

เมื่อผู้วิจัยได้ทำการรวมโครงสร้างของ Mask filter เข้ากับ ระบบการวิเคราะห์ที่ได้พัฒนาขึ้น โครงสร้างของโปรแกรมที่ใช้ทดสอบในโปรแกรม MATLAB จึงเป็นไปดังภาพประกอบ 3-3 ขั้นตอนหมายเลข ① คือขั้นตอนของ Mask filter โดยการทำงาน เริ่มต้นจากการสร้างตัว Mask image จากการคัดลอกภาพต้นฉบับ และลบช่วงสีที่ต้องการออก มีขั้นตอนการสร้างดังภาพประกอบ 3-4 เมื่อได้ Mask image ระบบจะทำการนำภาพที่ต้องการหาขอบ ลบออกจากภาพ Mask image ผลของภาพที่ถูกลบจะถูกนำเข้าสู่กระบวนการหาขอบเป็นลำดับถัดไป โดยภาพที่ได้จะถูกตัดข้อมูลบางส่วนที่ไม่จำเป็นออก ทำให้ภาระในการหาขอบและหาเส้นตรงลดลง

ในงานวิจัยชิ้นนี้ ระบบต้องการให้เส้นขาวของขอบถนน และเส้นสีเหลืองตรงกลางถนนเด่นชัดขึ้นดังแสดงในภาพประกอบ 3-5 ฉะนั้น Mask image จะเป็นภาพที่นำช่วงสีของเส้นสีขาวและช่วงสีของเส้นสีเหลืองออกจากภาพ ตัวอย่างดังภาพประกอบ 3-6 ได้แสดงผลการทำ Mask image



ภาพประกอบ 3-3 ขั้นตอนการหาเส้นโดยโปรแกรม MATLAB ที่ผ่านกระบวนการ Mask filter



ภาพประกอบ 3-4 อัลกอริทึมในการสร้าง Mask image

ภาพประกอบ 3-5 ตำแหน่งของเส้นขอบของถนน หรือช่องทางเดินรถที่ต้องการนำมาพิจารณา

ภาพประกอบ 3-6 ผลจากการสร้าง Mask image

กระบวนการ Mask filter จะทำการกรองสัญญาณภาพ โดยนำภาพต้นแบบลบกับ Mask image ผลสุดท้ายจะได้ภาพที่มีรายละเอียดของเส้นเหลือง และเส้นขาวที่มีความชัดเจนมากยิ่งขึ้น ซึ่งผลการทำงานจะแสดงในบทที่ 4

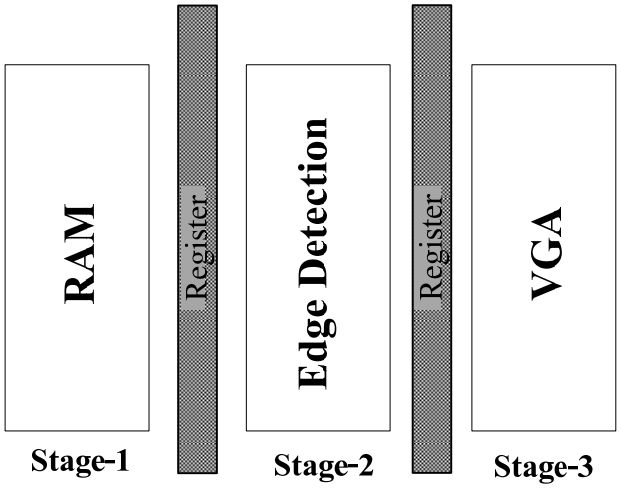
เมื่อระบบทำงานผ่านกระบวนการ Mask filter แล้ว ภาพที่ได้จะเข้าสู่กระบวนการหาขอบภาพ โดยกระบวนการนี้จะแสดงให้เห็นเส้นขอบเขตของกลุ่มสีที่อยู่ในภาพให้มีความเด่นชัดยิ่งขึ้น ส่งผลให้เห็นแนวโน้มของการหาเส้นตรงได้ การหาขอบของภาพมีอยู่หลายอัลกอริทึมแต่ผู้วิจัยได้พิจารณาคัดเลือก 4 อัลกอรึทึม ได้แก่ Canny, Prewitt, Soble และ Robert เนื่องจากอัลกอริทึมต่างๆ เหล่านี้เป็นที่นิยมในการนำมาพัฒนาในทางการประมวลผลภาพ และมีโครงสร้างที่สามารถพัฒนาใน FPGA ได้ แต่อย่างไรก็ตาม ในงานวิจัยชิ้นนี้จะนำมาเพียง หนึ่งอัลกอริทึมเท่านั้นที่จะพัฒนาใน FPGA ส่วนขั้นตอนการคัดเลือก และทดสอบผลนั้นจะกล่าวต่อในบทที่ 4 เช่นกัน

3.4 การพัฒนาระบบ บน FPGA

FPGA เป็นอุปกรณ์สารกึ่งตัวนำชนิดหนึ่งที่สามารถโปรแกรมการทำงานได้ มีโครงข่ายการเชื่อมต่อภายในแบบเมตริกซ์ การทำหน้าที่เปรียบเสมือนการต่อวงจรลอจิกโดยผู้ออกแบบสามารถโปรแกรม หรือ นิยามการทำงานของวงจรได้ โดยผ่านทางภาษา VHDL หรือ Verilog HDL นอกจากวงจรลอจิกแบบโปรแกรมได้แล้ว ยังมีบล็อกของหน่วยความจำ ที่เป็น Flip-Flop อย่างง่ายบรรจุอยู่ภายในอีกด้วย

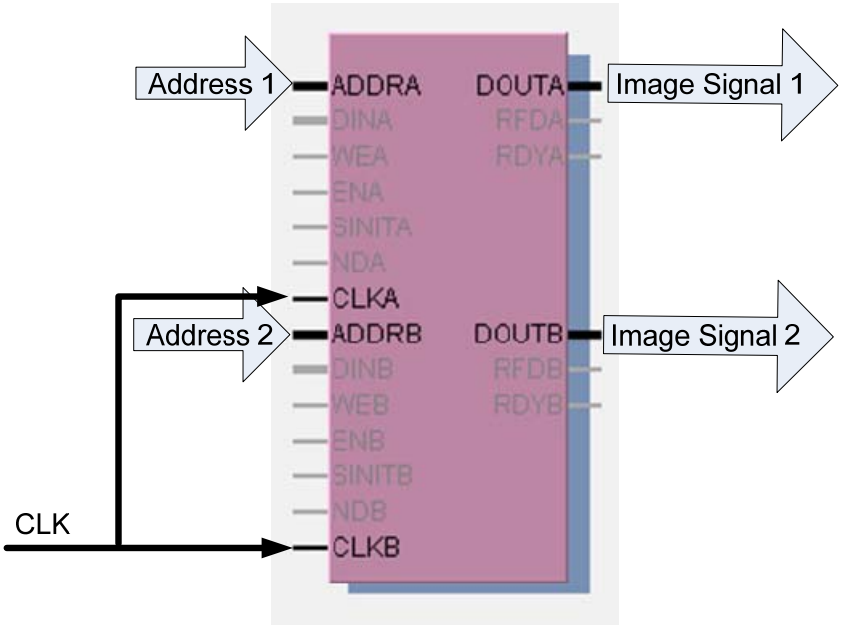
FPGA เป็นอุปกรณ์ที่ช่วยให้ผู้ออกแบบวงจรดิจิตอลสามารถทำงานได้ง่ายมากยิ่งขึ้น โดย FPGA จะช่วยลดขนาดของวงจรดิจิตอลให้มีขนาดเล็กลง เพราะผู้ออกแบบวงจรสามารถสร้างวงจรดิจิตอลภายในตัวอุปกรณ์ อีกทั้งยังช่วยลดเวลาในการพัฒนาวงจร และผู้ออกแบบสามารถแก้ไขวงจรได้ตลอดเวลา ซึ่งต่างจากแบบเดิมที่ผู้ออกแบบวงจรจะต้องออกแบบทั้งการเชื่อมต่อ ออกแบบลายวงจร และเมื่อสร้างวงจรเสร็จ ก็ไม่สามารถแก้ไขวงจรหากเกิดข้อผิดพลาดได้อีก

สำหรับวิจัยฉบับนี้ ผู้วิจัยได้พัฒนาระบบการหาขอบภาพบน FPGA โดยใช้โครงสร้างซึ่งแสดงในภาพประกอบ 3-3 เป็นเกณฑ์ โดยแบ่งไปป์ไลน์การทำงานออกเป็น 3 สเตจ ดังภาพประกอบ 3-7



ภาพประกอบ 3-7 โครงสร้างไปป์ไลน์ในการหาขอบภาพภายใน FPGA

สเตจที่ 1 เป็นการอ่านข้อมูลจาก RAM มาเก็บที่ รีจิสเตอร์ เพื่อเตรียมเข้าสู่กระบวนการหาขอบภาพ โมดูล RAM ที่ผู้วิจัยนำมาใช้พัฒนาบน FPGA เป็นโมดูลที่สามารถนำสัญญาณภาพออก 2 สัญญาณพร้อมกัน (Dual-port RAM) โดยระบบสามารถกำหนดแอดเดรสของพิกเซลได้พร้อมกัน 2 พิกเซล แต่ใช้สัญญาณนาฬิกาเพียง 1 สัญญาณ แสดงดังภาพประกอบ 3-8



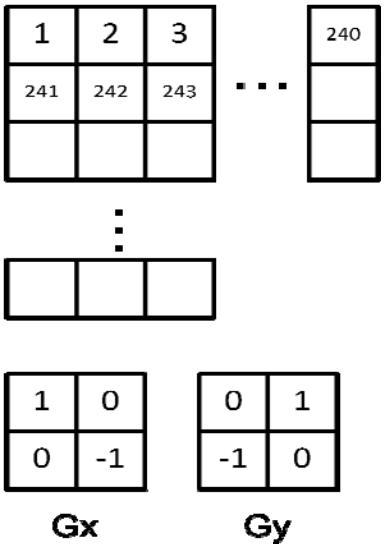
ภาพประกอบ 3-8 โมดูล Dual-port RAM ที่สร้างขึ้นบน FPGA

สเตจที่ 2 เป็นกระบวนการหาขอบภาพ โดยข้อมูลสัญญาณจากรีจิสเตอร์ในสเตจที่หนึ่งจะถูกส่งเข้ามาประมวลผล ซึ่งโครงสร้างภายในของเสตจนี้ จะเป็นการคำนวณทางคณิตศาสตร์โดยลักษณะการคำนวณจะขึ้นอยู่กับอัลกอริทึมหาขอบที่เลือกพัฒนาในส่วนนี้ เช่น หากเลือกการหาขอบภาพโดยวิธีของ Roberts ซึ่ง มีการใช้ Kernel G_x และ G_y เป็นเมทริกซ์ ขนาด 2x2 ดังสมการ (3-2) เป็นผลให้การทำ Convolution ทางฮาร์ดแวร์โดยใช้ Dual Port Ram สามารถทำให้เข้าถึงข้อมูลได้พร้อมๆ กัน

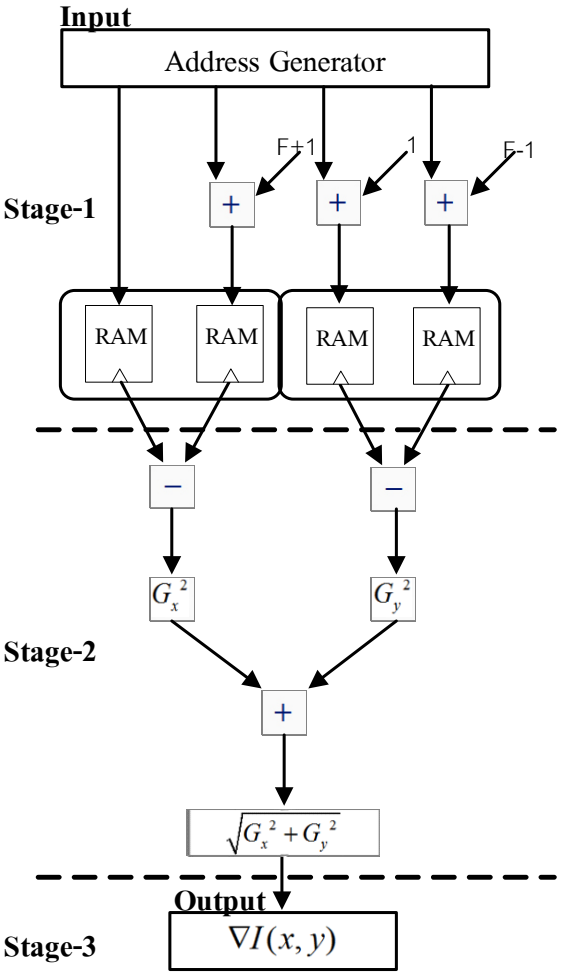
$$G_x(x,y)=\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} \quad G_y(x,y)=\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}$$

$$\nabla I(x,y)=G(x,y)=\sqrt{G_x^{\,2}+G_y^{\,2}} \tag{3-2}$$

จากภาพประกอบ 3-9 พบว่าในข้อมูลรูปภาพแอดเดรสของแต่ละพิกเซล จะถูกเรียงตามลำดับ และเมื่อนำ Gx มา Convolution จะต้องใช้ข้อมูลของแอดเดรสที่ 1 และแอดเดรสที่ 242 ซึ่งกระบวนการนี้สามารถอธิบายเป็น Data flow graph ได้ดังภาพประกอบ 3-10 โดยค่าพารามิเตอร์ F คือขนาดของความกว้างของ Frame ภาพ เช่น Frame ขนาดภาพ 640x480 พิกเซล ค่า F จึงเท่ากับ 640

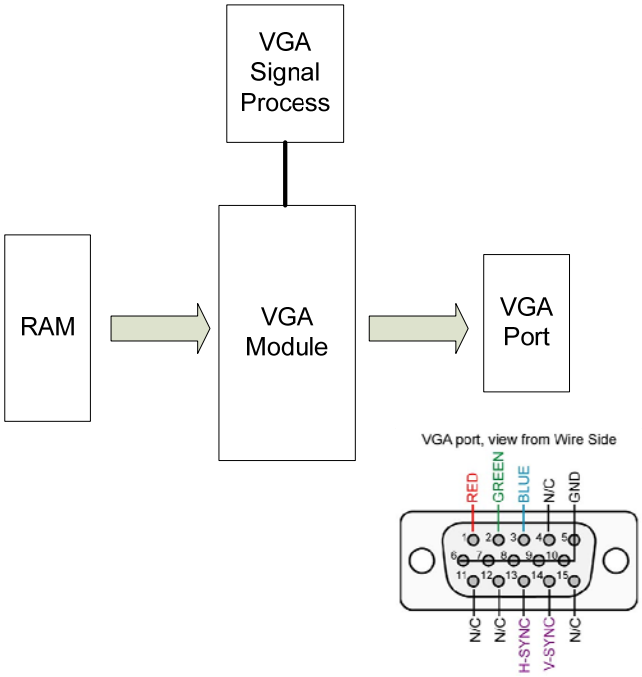


ภาพประกอบ 3-9 โครงสร้างของภาพ และการทำ Convolution



ภาพประกอบ 3-10 Data flow graph การหาขอบภาพ โดยวิธีของ Roberts

สเตจที่ 3 เป็นการแสดงผลออกทางหน้าจอ ในการพัฒนาส่วนของการหาขอบใน FPGA จำเป็นต้องมีการสร้างโมดูลแสดงผล เพราะกระบวนการทำงานภายในสามารถวิเคราะห์ได้จากโค้ดภาษา Verilog HDL แต่ผลลัพธ์ของการทำงาน ต้องพิจารณาทางจอแสดงผลเท่านั้น โดยโครงสร้างของโมดูล VGA จะแสดงภาพประกอบ 3-11



ภาพประกอบ 3-11 โครงสร้างของโมดูล VGA บน FPGA

เนื่องจากระบบการแสดงผลที่ผู้วิจัยได้พัฒนาขึ้น ใช้การส่งสัญญาณภาพผ่านจอ CRT โดยมีพอร์ต VGA ซึ่งติดมากับบอร์ด Vertex-II Pro เป็นตัวเชื่อม ดังนั้นสัญญาณภาพจะถูกควบคุมโดยส่วนควบคุมสัญญาณ VGA ซึ่งถูกพัฒนาขึ้นและอยู่บน FPGA ด้วยเช่นกัน

ในบทนี้ได้แสดงแนวคิดและขั้นตอนการทำงานในส่วนต่างๆ ซึ่งประกอบไปด้วย 2 ส่วนหลัก คือ ส่วนของการทดสอบการทำงานของอัลกอริทึม ซึ่งได้ทำการทดสอบบน MATLAB และได้มีการออกแบบอัลกอริทึมทั้งส่วนของ การหาเส้นตรงโดยไม่มีกระบวนการ Mask filter และอัลกอริทึม ที่มีกระบวนการ Mask filter โดยในการศึกษาในงานวิจัยฉบับนี้ได้เปรียบเทียบอัลกอริทึมหาขอบที่เหมาะสมที่จะพัฒนาใน FPGA ซึ่งมี 4 อัลกอริทึมที่ผู้ทำวิจัยให้ความสนใจ คือ Canny, Prewitt, Sobel และ Roberts อีกส่วนหนึ่งของบทนี้ จะกล่าวถึงขั้นตอนการพัฒนาการหาขอบภาพแบบ Roberts ลง FPGA ซึ่งเป็นการแปลงอัลกอริทึมทางคณิตศาสตร์ เป็นวงจรดิจิตอลทำ

ให้เกิดผลลัพธ์ขึ้นได้ อีกทั้งช่วยให้การทำงานเร็วมากยิ่งขึ้น และในส่วนถัดไปจะกล่าวถึงผลการทดสอบอัลกอริทึมตามที่ได้กล่าวมาแล้วในบทนี้

บทที่ 4
การทดลองและการวิเคราะห์ผล

ในบทนี้อธิบายการทดสอบเพื่อคัดเลือกกระบวนการหาขอบภาพที่เหมาะสม เพื่อนำไปพัฒนากระบวนการหาช่องทางเดินรถโดยพัฒนาบน FPGA หลังจากที่ได้กระบวนการที่เหมาะสมแล้ว ผลจากการใช้ Mask filter ที่ผู้วิจัยได้คิดค้นขึ้น จะแสดงให้เห็นประสิทธิภาพของการหาขอบและการหาเส้นตรงภายในภาพ โดยจะวัดจากความถูกต้องของเส้นที่พบในภาพนั้นๆ ส่วนสุดท้ายจะเป็นผลจากการสร้างวงจรลงบอร์ด FPGA

4.1 แนวทางการทดลองและการเก็บผลการทดลอง

ในงานวิจัยฉบับนี้ เป็นการทดลองเพื่อหาอัลกอริทึม สำหรับพัฒนาใน FPGA โดยขั้นตอนการทดลองจะใช้โปรแกรม MATLAB เป็นเครื่องมือในการค้นหาและพัฒนาอัลกอริทึม ประการสำคัญคือ อัลกอริทึมที่ค้นหาและพัฒนานั้น ต้องสามารถนำไปพัฒนาเป็นอัลกอริทึมภายใน FPGA ได้ ซึ่งการพัฒนาอัลกอริทึมให้สอดคล้องกับเทคโนโลยี FPGA มีข้อจำกัดในเรื่องการใช้ทรัพยากรภายใน และความซับซ้อนของการคำนวณ อีกทั้งความถูกต้องของผลลัพธ์ที่ได้จากอัลกอริทึมนั้นต้องสามารถยอมรับได้ ด้วยเหตุนี้ผู้วิจัยจึงได้ออกแบบการทดลองเพื่อค้นหาอัลกอริทึมโดยแบ่งออกเป็น 3 ส่วน ได้แก่

4.1.1 การทดสอบเพื่อหากระบวนการหาขอบภาพ

ในการทดลองดังกล่าว ผู้วิจัยได้นำภาพถ่ายถนนซึ่งเป็นสภาพแวดล้อมตามจริงเพื่อทำการทดลองหาเส้นตรงภายในรูป โดยใช้อัลกอริทึมหาขอบทั้ง 4 วิธี ได้ แก่ Canny, Prewitt, Soble และ Roberts ทำการทดสอบผ่านโปรแกรม MATLAB เพื่อ ให้ได้เส้นขอบของภาพ และทำการหาเส้นตรง จากนั้นทำการวัดผลการทำงานโดยดูเวลาในการประมวลผลพร้อมทั้งวิเคราะห์ผลการค้นหาเส้นตรงเปรียบเทียบกัน

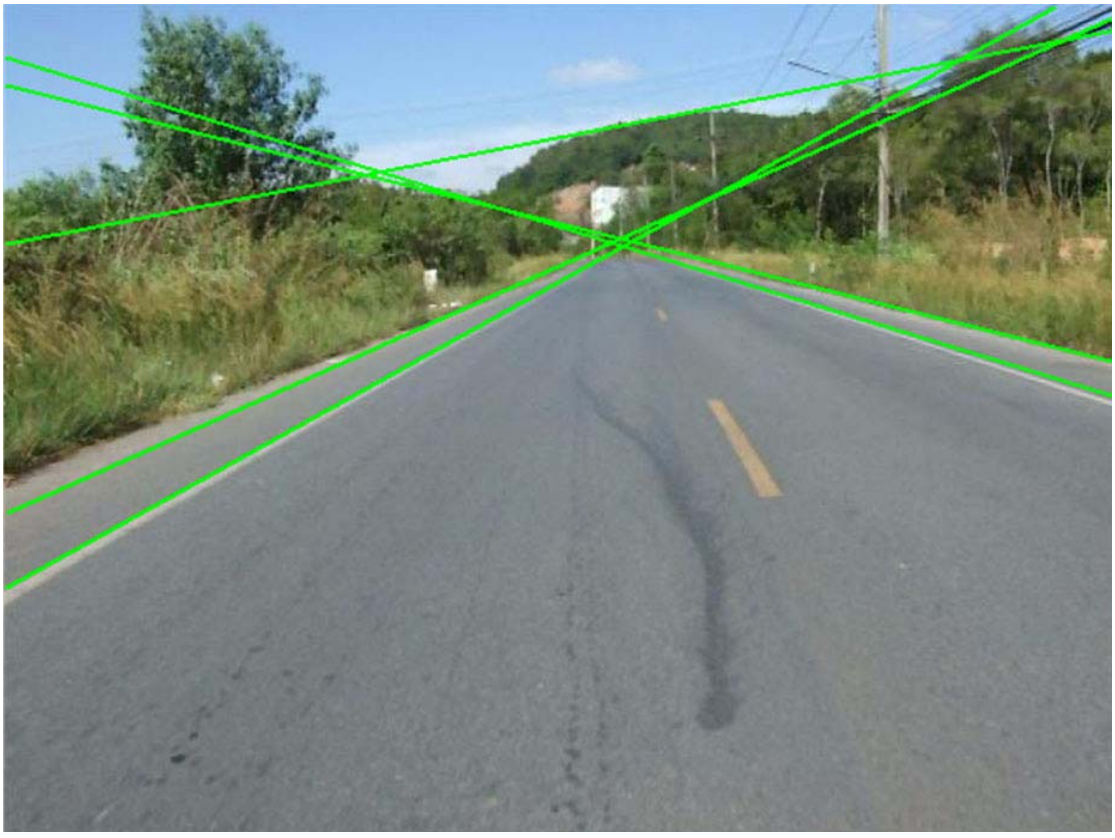
สำหรับการทดลองในโปรแกรม MATLAB ผู้วิจัยได้ทำการถ่ายภาพจากสถานที่จริงในช่วงเวลาประมาณ 10.00 น.ในวันที่ท้องฟ้าแจ่มใส โดยผู้วิจัยได้ทำการบันทึกเป็นไฟล์วิดีโอ และทำการแยกรูปออกเป็นเฟรมเพื่อให้โปรแกรมทดสอบพี่ผู้วิจัยพัฒนาขึ้น สามารถอ่านภาพที่กำหนดไว้ได้

4.1.2 การทดสอบเพื่อหากระบวนการหาเส้นตรงโดยใช้ Mask filter ในการปรับปรุงภาพ

เงื่อนไขในการทดลองคือ ผู้วิจัยจะทำการใช้ภาพเดิมที่ได้ใช้วิเคราะห์ในข้อ 4.1.1 เพื่อเป็นการเปรียบเทียบผลลัพธ์อย่างมีประสิทธิภาพ

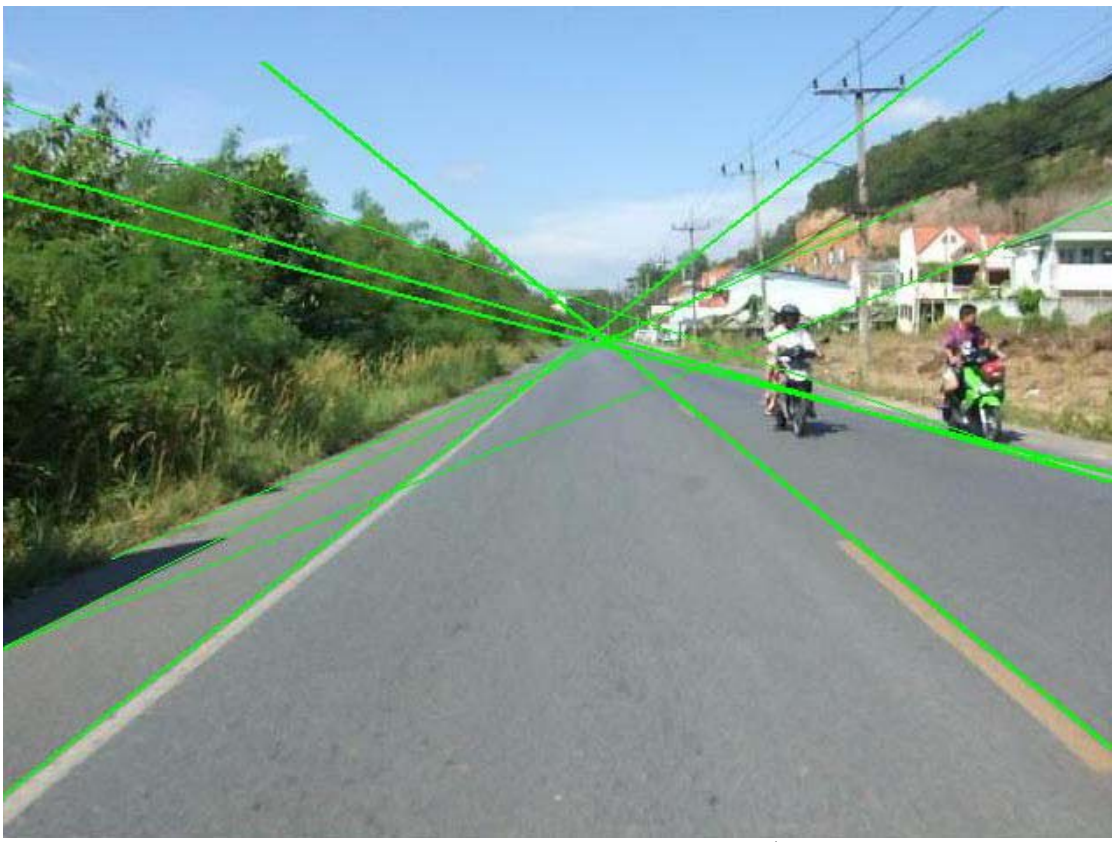
4.1.3 การคำนวณหาความถูกต้องของการหาเส้นตรง

หลังจากที่ระบบทำการค้นหาเส้นตรงแล้ว ต้องมีการวิเคราะห์ว่าเส้นที่พบ เป็นเส้นตรงที่งานวิจัยต้องการหรือไม่ ซึ่งเส้นที่เป็นประโยชน์ในการหาขอบถนน สามารถเกิดขึ้นได้หลายเส้น ในงานวิจัยนี้หากเส้นที่พบอยู่ในเส้นแนวที่ระบบต้องการ ผู้วิจัยจะอนุมานว่าเป็นเส้นที่ต้องการ



ภาพประกอบ 4-1 ผลการค้นหาเส้นตรงในภาพ

จากภาพประกอบ 4-1 พบว่ามีเส้นตรงที่ค้นพบทั้งสิ้นจำนวน 5 เส้น ซึ่งเป็นเส้นที่ถูกต้องและระบบต้องการจำนวนเพียง 4 เส้น และในภาพประกอบ 4-2 พบว่ามีเส้นตรงทั้งสิ้น 8 เส้น ซึ่งบางส่วนซ้อนทับกัน และบางส่วนตัดกัน ซึ่งผู้วิจัยพิจารณาให้พบเส้นตรงที่ต้องการ 5 เส้น



ภาพประกอบ 4-2 ผลการหาเส้นตรงกรณีพบเส้นตรงซ้ำกันหลายเส้น

การคำนวณหาเปอร์เซ็นต์ความถูกต้อง (%TP) สามารถคำรวณได้จากสมการ (4-1)

$$\%(\mathrm{TP}) = \left\{ \frac{TP}{(TP + FP)} \right\} \times 100 \qquad (4\text{-}1)$$

โดย TP คือ True Positive หมายถึง พบเส้น และเป็นเส้นในตำแหน่งที่ต้องการ

FP คือ False Positive หมายถึง พบเส้น และเป็นเส้นที่ในตำแหน่งไม่ต้องการ

4.2 ผลการทดสอบการหาขอบภาพและเส้นตรงโดย MATLAB

จากงานวิจัยที่ผ่านมา ได้กล่าวไว้ว่าการหาขอบภาพวิธีของ Canny ได้รับการยอมรับว่ามีประสิทธิภาพสูง ซึ่งหลังจากทำการทดลองด้วยตัวเองพบว่า การหาขอบโดยใช้วิธี Canny จะให้รายละเอียดที่มากกว่า การหาขอบโดยวิธีอื่น ซึ่งสามารถแสดงผลการหาขอบได้ดังภาพประกอบต่อไปนี้

ภาพประกอบ 4-3 ภาพตัวอย่าง ก่อนผ่านกระบวนการหาขอบ

ภาพประกอบ 4-4 ภาพที่ผ่านกระบวนการหาขอบภาพโดยวิธี Canny

ภาพประกอบ 4-5 ภาพที่ผ่านกระบวนการหาขอบภาพโดยวิธี Prewitt

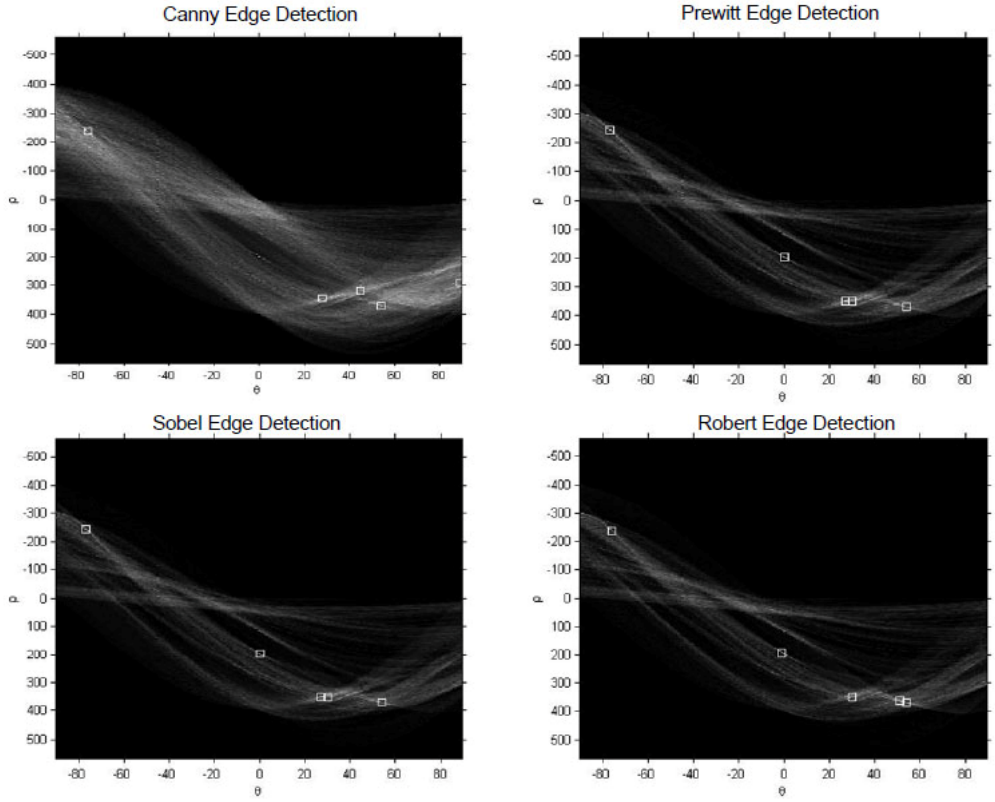
ภาพประกอบ 4-6 ภาพที่ผ่านกระบวนการหาขอบภาพโดยวิธี Sobel

ภาพประกอบ 4-7 ภาพที่ผ่านกระบวนการหาขอบภาพโดยวิธีของ Roberts

ภาพประกอบ 4-3 เป็นภาพตัวอย่างที่ได้นำมาวิเคราะห์เพื่อดูประสิทธิภาพการทำงานของอัลกอริทึม หลังจากทดสอบโดยใช้การหาขอบวิธี Canny พบว่ามีลายเส้นของขอบภาพเป็นจำนวนมากดังปรากฏในภาพประกอบ 4-4 แต่เมื่อทำการทดสอบโดยใช้อัลกอริทึมของ Prewitt, Sobel และ Roberts พบว่าจำนวนของเส้นขอบที่พบมีน้อยกว่าของ Canny แต่ทั้ง 3 อัลกอริทึมมีปริมาณเส้นขอบที่พบใกล้เคียงกัน ซึ่งผลการทำงานได้แสดงในภาพประกอบ 4-5 ถึง ภาพประกอบ 4-7 จากการทดสอบนี้ ผู้วิจัยมองว่า สาเหตุหนึ่งที่ทำให้ผลออกมาใกล้เคียงกัน เพราะโครงสร้างการทำงานของทั้ง 3 อัลกอริทึม มีความคล้ายคลึงกัน ซึ่งแตกต่างกับ อัลกอริทึมของ Canny

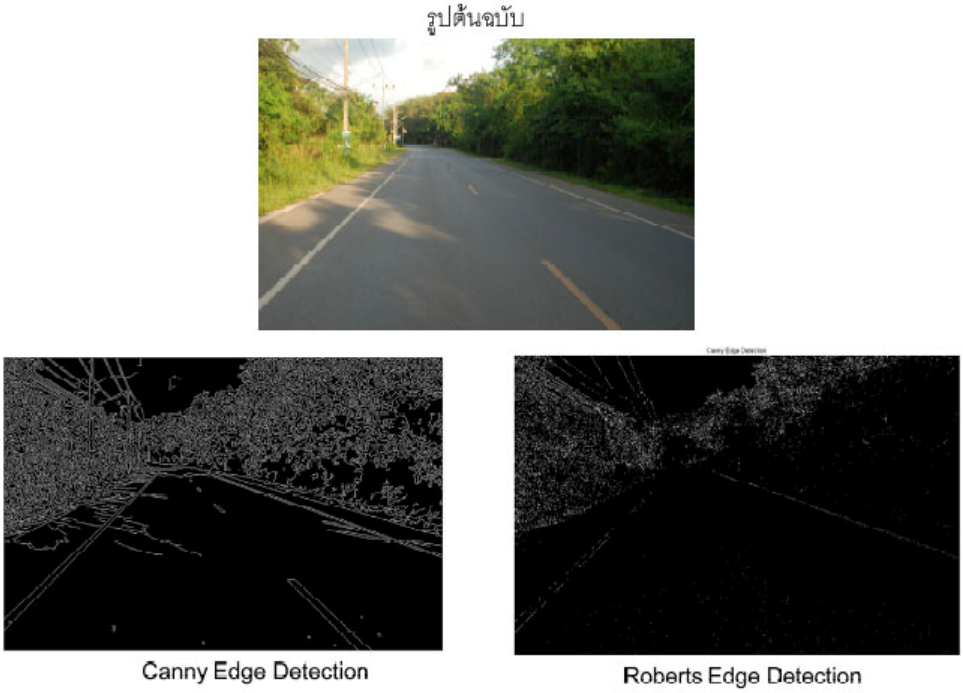
จากผลการทดลองสามารถทำการหา Hough space ได้ดังภาพประกอบ 4-8 จาก Hough space ดังกล่าวสามารถวิเคราห์ได้ว่าการหาขอบภาพแบบ Canny มีอัตราการทำ Hough transform

มากที่สุด ทั้งนี้เนื่องจากมีรายละเอียดของขอบภาพมากที่สุดนั่นเอง ส่วนการทำ Hough transform ของวิธีอื่นๆ มีการทำงานที่ใกล้เคียงกันมาก และหากพิจารณา การหาขอบภาพ ก็มีลักษณะขอบที่ใกล้เคียงกัน ทั้งนี้งานวิจัยที่ผ่านมา ได้นำเสนอวิธีการหาขอบของ Roberts เพื่อทำการพัฒนาลง FPGA เพราะเป็นวิธีที่ใช้ทรัพยากรณ์น้อยที่สุด และผลที่ได้มีความใกล้เคียงกับวิธีการหาขอบภาพแบบ Sobel และ Prewitt มาก

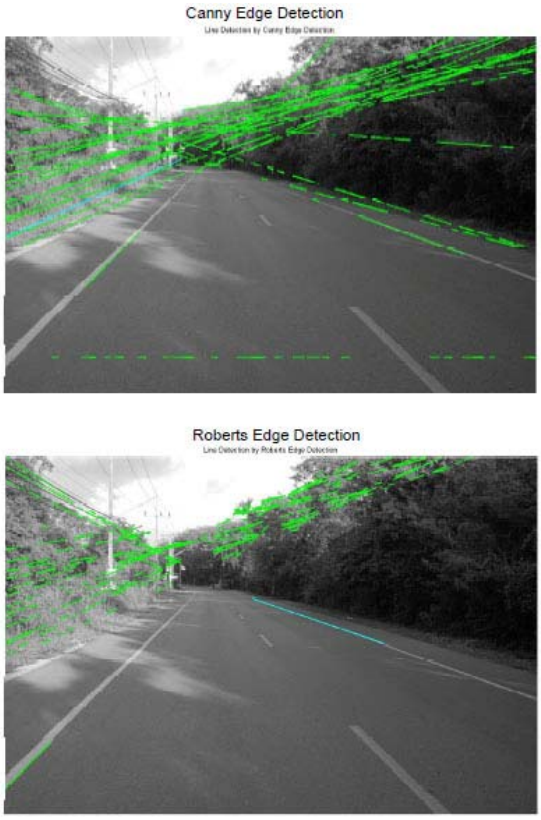


ภาพประกอบ 4-8 ผลการทำ Hough transform เพื่อหาเส้นตรงในแต่ละอัลกอริทึม

ดังนั้นเพื่อทำการวิเคราะห์ผล จะทำการทดลองโดยจะนำวิธีหาขอบ แบบ Canny เปรียบเทียบกับ Roberts ทำการทดลองโดยการไปถ่ายภาพจากสถานที่จริง หลังจากนั้นได้ลองหาขอบของภาพ โดยวิธีต่างๆ และสร้าง Hough space จากนั้นทำการ หาเส้นตรงในภาพถ่าย ผลที่ได้สามารถแสดงได้ดังภาพประกอบ 4-9

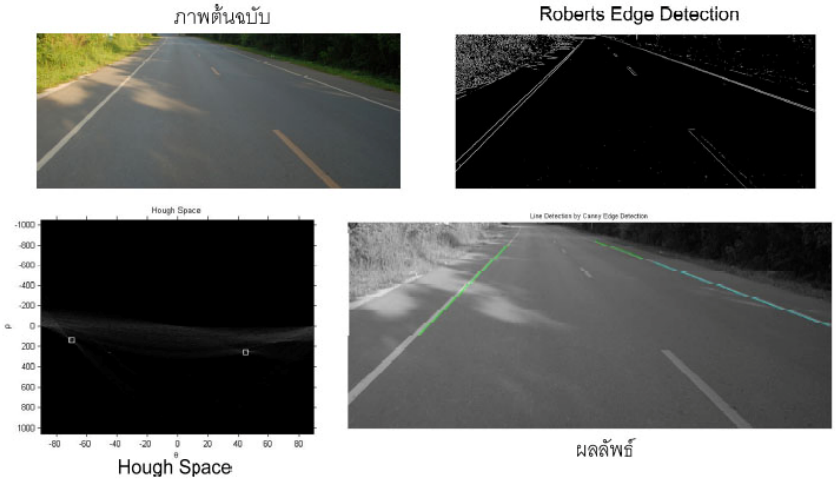


ภาพประกอบ 4-9 ผลการหาขอบภาพ โดยใช้วิธี Canny และ Roberts

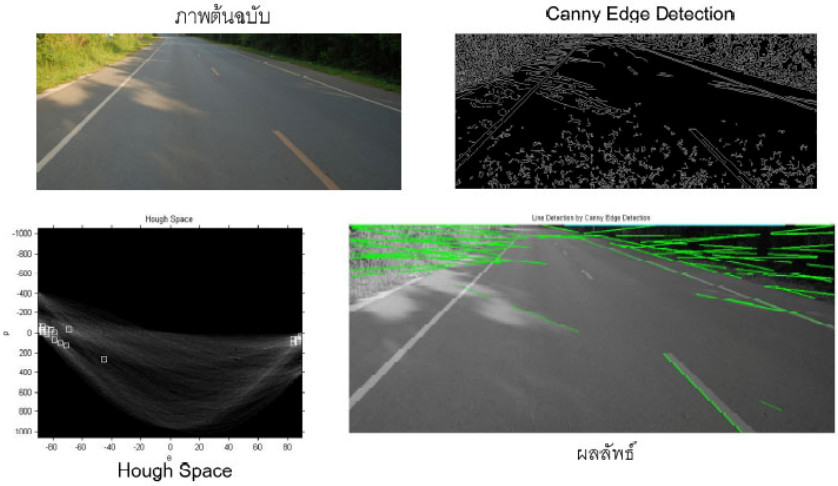


ภาพประกอบ 4-10 ผลการหาเส้นตรง โดยใช้การหาขอบภาพของ Canny และ Roberts

จากผลการทดลองพบว่า ในการหาเส้นตรงในภาพ ยังเกิดส่วนที่เป็นค่าผิดพลาด ซึ่งไม่ต้องการเป็นจำนวนมาก ทั้งนี้เนื่องจากว่า ตำแหน่ง พิกเซล ที่หาได้ในระบวนการหาขอบ อาจอยู่ในสมการเส้นตรงเดียวกัน จึงทำให้เกิดเป็นเส้นตรงที่เราไม่ต้องการได้ ซึ่งวิธีการแก้ปัญหาที่สามารถเป็นไปได้คือ ลดกรอบการพิจารณาลงให้เหลือขอบเขตในวงจำกัด ซึ่งผลลัพธ์ที่ได้ค่อนข้างน่าพอใจกว่าครั้งแรก คือ สามารถค้นหาเส้นตรงที่ต้องการได้ ซึ่งผลลัพธ์ที่กล่าวถึงแสดงในภาพประกอบ 4-11 และภาพประกอบ 4-12



ภาพประกอบ 4-11 ผลการค้นหาเส้นตรงโดยเลือกเฉพาะพื้นที่ และใช้วิธีหาขอบภาพของ Roberts



ภาพประกอบ 4-12 ผลการค้นหาเส้นตรงโดยเลือกเฉพาะพื้นที่ และใช้ การหาขอบภาพแบบ Canny

จากผลลัพธ์ทั้งสองวิธี จะเห็นได้ชัดว่า การหาขอบภาพแบบละเอียด ไม่ได้ส่งผลให้เกิดผลลัพธ์ที่ดีในการหาเส้นตรง เสมอไป เช่นในภาพประกอบ 4-12 ได้มีการใช้ อัลกอริทึม การหาขอบที่มีประสิทธิภาพสูงกว่า แต่ผลการหาเส้นตรง มีความผิดพลาดสูงมาก เปรียบเทียบกับภาพประกอบ 4-11 ซึ่งประสิทธิภาพในการหาขอบภาพด้อยกว่า แต่ผลที่ได้ กลับเป็นที่น่าพอใจกว่ามาก

เมื่อทำการวิเคราะห์เวลาในการประมวลผลการหาขอบภาพได้ผลดังตาราง 4-1 ซึ่งเมื่อทำการเปรียบเทียบโดยใช้การหาขอบภาพโดยวิธีของ Canny เป็นเกณฑ์ในการวัด ได้ผลคือ Prewitt, Sobel และ Roberts ใช้เวลาในการประมวลผลน้อยกว่า ถึง 3 เท่า (วิเคราะห์ โดยใช้ MATLAB)

จากผลการวัดการหาขอบภาพโดยวิธี Roberts ใช้เวลาในการทำงาน น้อยที่สุด และจากการหาความถูกต้องของการค้นหาเส้นวิธีการของ Roberts มี เปอร์เซ็นต์ความถูกต้องของเส้นที่พบ สูงเช่นกัน

ตาราง 4-1 เวลาในการประมวลผลในขณะหาขอบภาพของแต่ละอัลกอริทึม

Edge Detection	Processing Time (sec.)	Normalized speed (time)
Canny	1.652	1
Prewitt	0.538	3.071
Sobel	0.533	3.099
Roberts	0.526	3.141

4.3 ผลการทดสอบการปรับปรุงความแม่นยำในการหาเส้นตรง ด้วย Mask filter

ภายในส่วนของ Mask จะเป็นการเช็คสีในแต่ละพิกเซลซึ่งค่าช่วงสีที่เลือกมาอยู่ใน Mask นั้น จะเป็นช่วงสีที่ไม่ต้องการมาพิจารณา อย่างเช่น หากต้องการพิจารณาเส้นสีเหลือง สีที่อยู่ใน Mask จะเป็นสีอื่นๆ ที่ไม่ใช่สีเหลือง จากภาพประกอบ 4-13 เมื่อต้องการวิเคราะห์ภาพที่มีเส้นสีขาว อัลกอริทึมจะลบค่าสีที่มีส่วนผสมของสีขาวออก หลังจากนั้น จึงนำภาพที่รับเข้ามา ลบกับภาพ Mask จะได้ภาพที่จะนำไปวิเคราะห์หาขอบของภาพต่อไป ซึ่งภาพที่ได้ออกมานั้น จะเป็นภาพที่มีส่วนผสมของสีที่ต้องการอยู่เท่านั้น

Original Image

Mask

Result Image

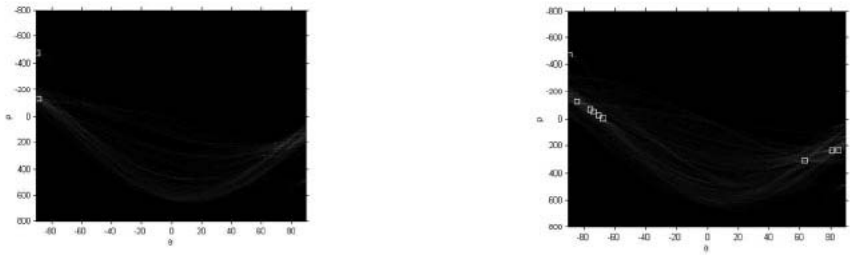
ภาพประกอบ 4-13 การทำ Mask image และผลลัพธ์จากการทำ Mask filter

จากภาพประกอบ 4-15 จะเห็นว่าประสิทธิภาพของ Hough transform ดีขึ้น ทำให้สามารถหาเส้นขอบถนนได้ชัดเจนกว่า การทดลองที่ไม่ผ่าน Mask filter ซึ่งการหาเส้นตรงได้อย่างแม่นยำหรือไม่นั้น ขึ้นอยู่กับขั้นตอนของการหาขอบขอบภาพ จากภาพประกอบ 4-14 จะเห็นความแตกต่างในการหาขอบภาพของทั้ง 2 อัลกอริทึมอย่างชัดเจน

ในภาพประกอบ 4-16 เป็นผลการใช้อัลกอริทึมที่ได้พัฒนาขึ้น มาทดสอบประมวลผลภาพที่ได้จากเฟรม ในแต่ละแฟรมของกล้องวีดีโอ ซึ่งผู้ทำวิจัยได้สุ่มยกตัวอย่างบางเฟรมมาแสดงผล

Original Image

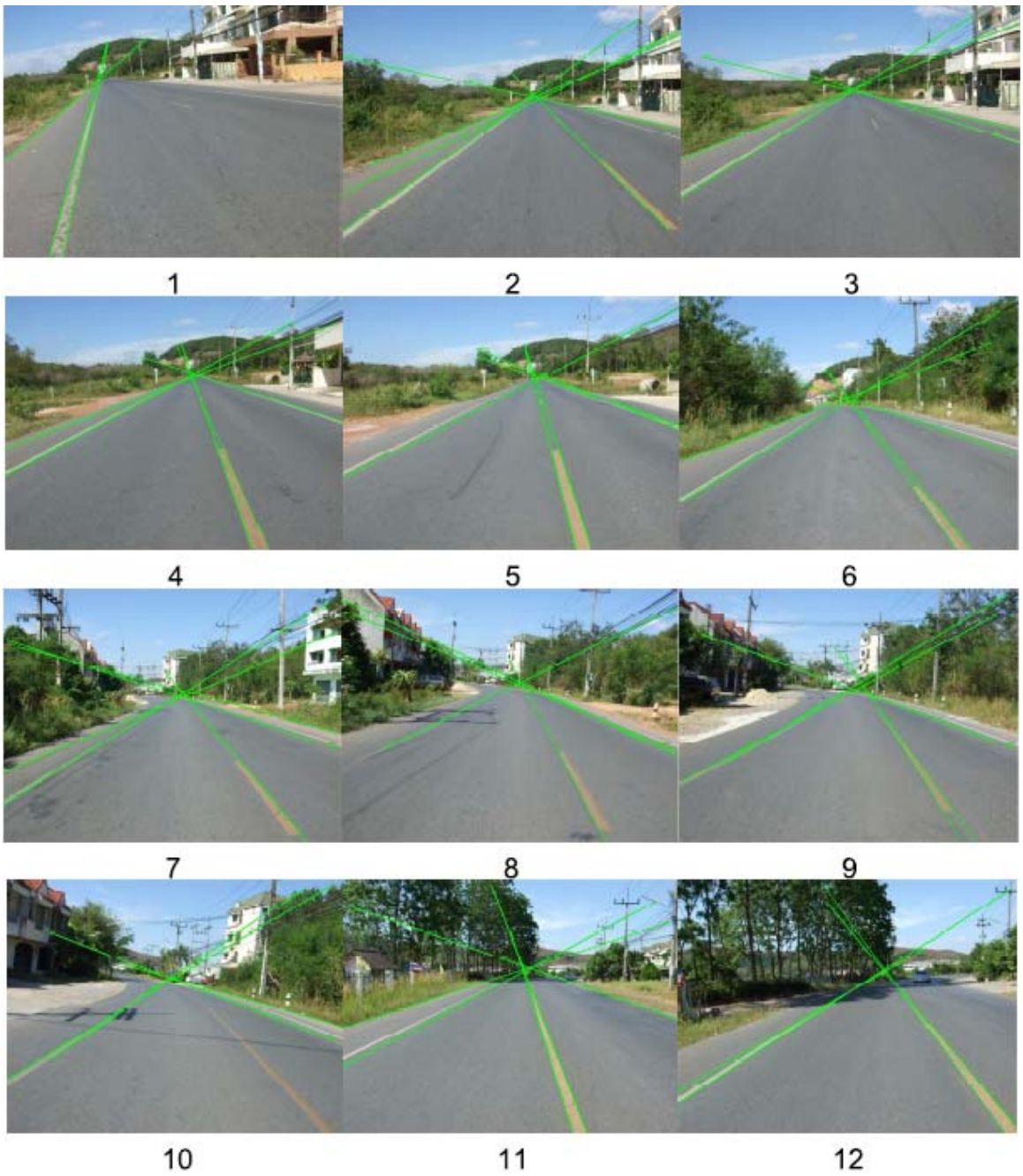
Roberts Edge Detection



Hough Transform

ภาพประกอบ 4-14 เปรียบเทียบการหาขอบภาพ โดยใช้ และไม่ใช้ Mask filter

ภาพประกอบ 4-15 เปรียบเทียบผลการหาเส้น โดยภาพที่ผ่าน และภาพที่ไม่ผ่าน Mask filter



ภาพประกอบ 4-16 ผลการใช้กระบวนการ Mask filter เพื่อช่วยในการหาเส้นตรงในภาพ

ตาราง 4-2 ผลการสุ่มหาเปอร์เซ็นต์ความถูกต้องของเส้นตรงที่พบ ตั้งแต่แฟรม 100-1500

Algorithm	Canny		Prewitt		Sobel		Roberts	
frame number	DL	AL	DL	AL	DL	AL	DL	AL
100	5	4	2	2	4	4	4	4
200	5	1	2	2	2	2	4	4
300	5	4	3	3	3	3	4	4
400	6	4	3	3	3	3	7	6
500	7	4	1	1	1	1	4	3
600	7	4	4	4	3	3	7	5
700	5	2	3	2	5	3	7	4
800	8	5	8	5	8	5	6	4
900	5	4	6	3	6	3	5	3
1000	7	4	3	2	6	2	2	1
1100	6	3	1	1	1	1	5	4
1200	8	4	0	0	0	0	0	0
1300	7	3	2	2	2	2	5	5
1400	4	3	1	1	1	1	3	3
1500	2	2	3	3	3	3	4	3
total	**87**	**51**	**42**	**34**	**48**	**36**	**67**	**53**

DL (Detected Line) หมายถึง เส้นที่พบในภาพ

AL (Actual Lanes) หมายถึง เส้นที่อยู่ในตำแหน่งช่องทางเดินรถที่ถูกต้อง

ตาราง 4-3 ผลของความถูกต้องของเส้นที่ระบบสามารถตรวจจับได้

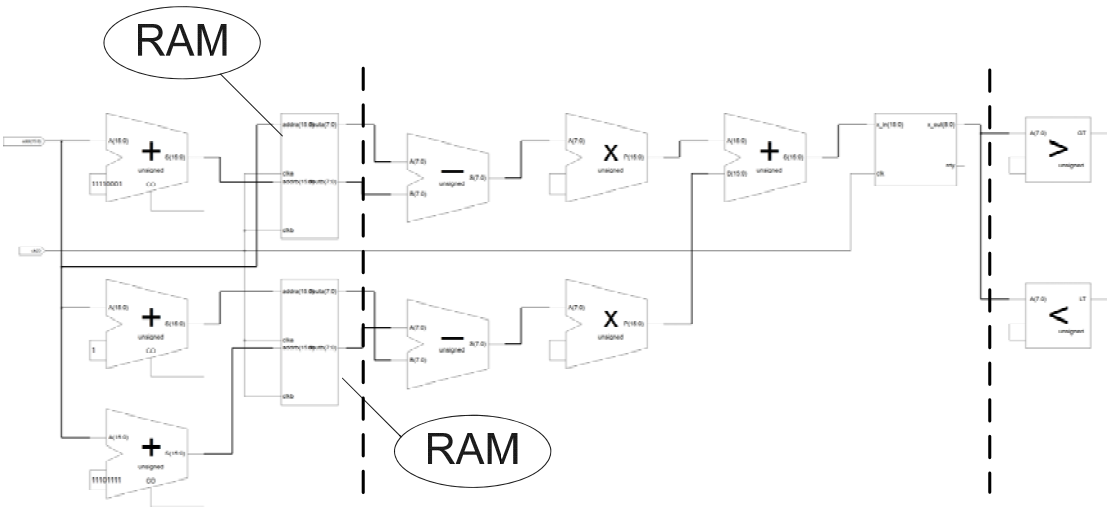
Edge Detection	Detected Line	Actual Lanes	True Positive (%)
Canny	87	51	76.67
Prewitt	42	34	73.33
Sobel	48	36	80
Roberts	67	53	83.33

หลังจากทำการจำลองการทำงานใน MATLAB พบว่าปัญหาของการหาเส้นตรงยังมีอยู่ คือ ระบบยังพบเส้นตรงที่ไม่เกี่ยวข้องกับสิ่งที่ต้องการ ซึ่งการจะแก้ปัญหาส่วนนี้ ควรจะอยู่ในส่วนของการสร้าง Mask ให้มีประสิทธิภาพมากกว่าเดิม และอาจจะต้องนำความชันของเส้นที่หาได้มาพิจารณาความชัน เพื่อเป็นตัวช่วยลดเส้นที่ไม่จำเป็นออกไปได้ส่วนหนึ่ง

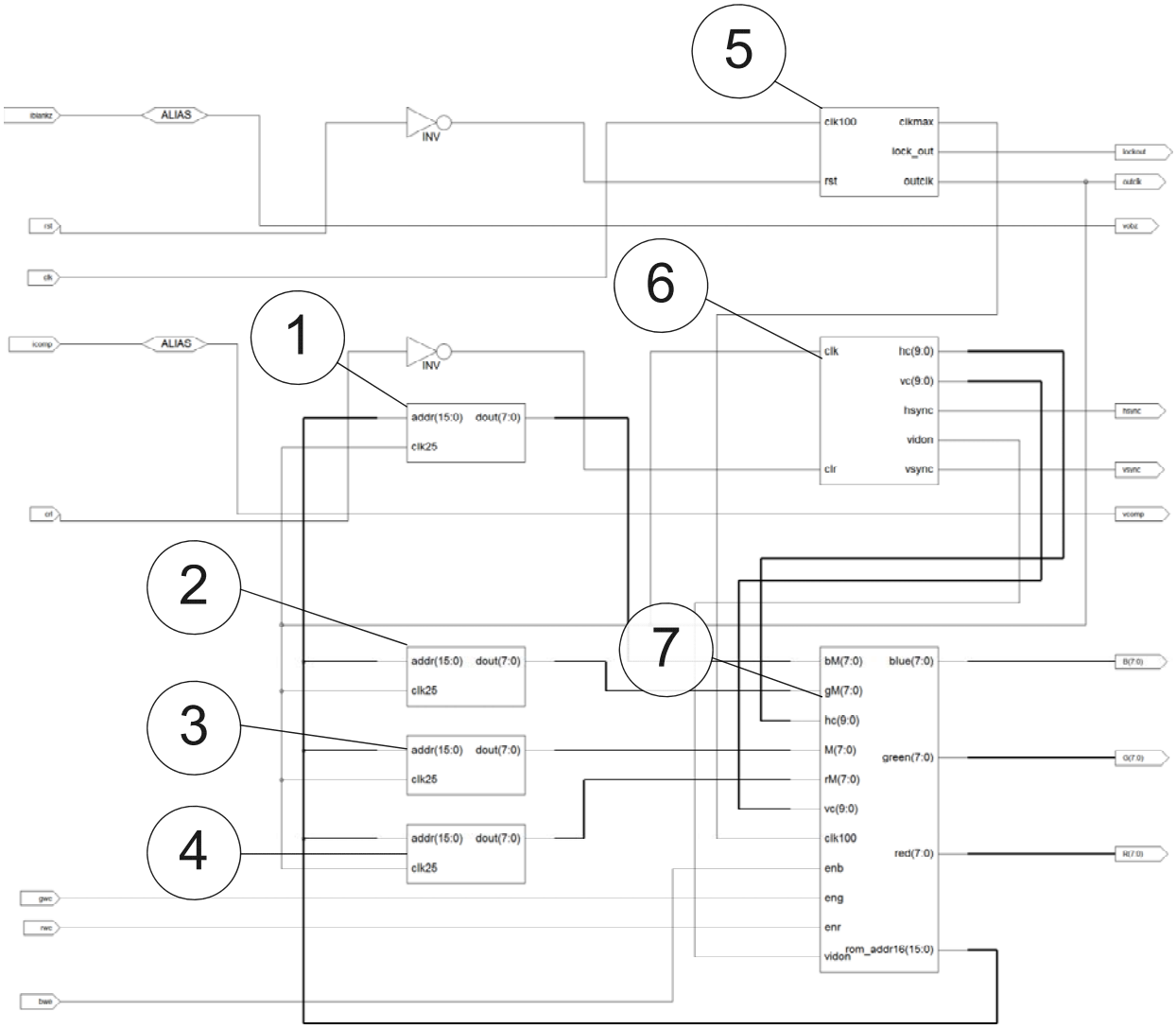
ตาราง 4-2 เป็นการตรวจสอบความถูกต้องของเส้นที่พบโดยทำการสุ่มแฟรมภาพ 15 ภาพ จาก 1500 ภาพ โดยใช้สมการที่ได้กล่าวไว้ในบทที่ 3 ซึ่งผลความถูกต้องจากการสุ่มตรวจได้แสดงในตาราง 4-3

4.4 ผลการทดสอบการทำงานหลังจากสร้างวงจรลง FPGA

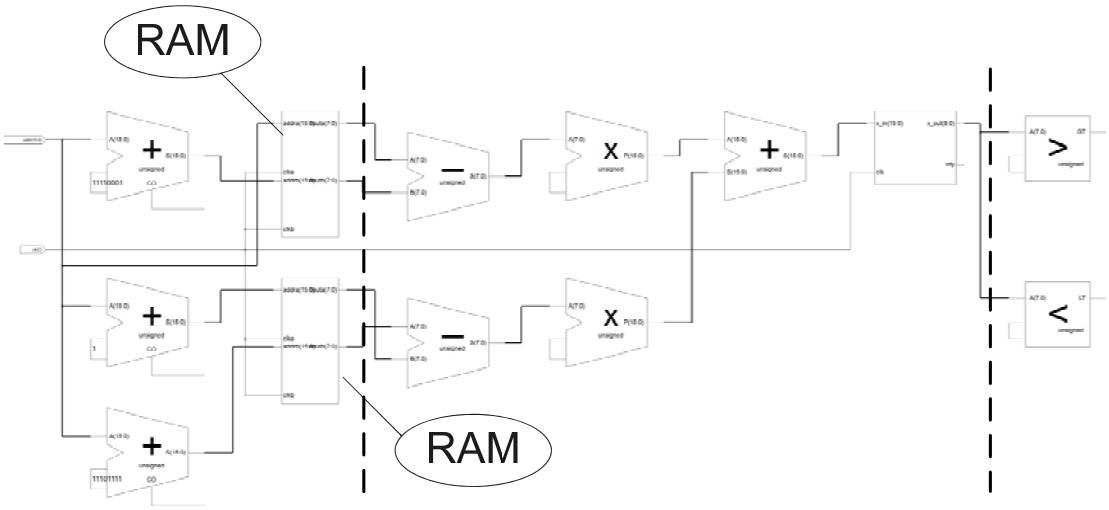
จากภาพประกอบ 4-17 ภาพรวมระบบค้นหาเส้นภายในภาพ และ



ภาพประกอบ 4-18 โครงสร้างไปป์ไลน์ในการหาขอบภาพภายใน FPGA ดังที่ได้นำเสนอในบทที่ 3 เมื่อนำมาสร้างวงจรลง FPGA ได้ผลดังภาพประกอบ 4-17 หมายเลข 1, 2, 3 และ 4 เป็นส่วนของแรม ที่ใช้เก็บภาพ และเป็นส่วนที่มีการหาขอบของภาพภายในตัวเอง สำหรับหมายเลข 5 เป็นที่ใช้กำหนดสัญญาณนาฬิกา ให้การทำงานสามารถดำเนินไปได้ โดย ความถี่ที่ใช้ในการทำงานอยู่ที่ 25 MHz ภาพหมายเลข 6 และ หมายเลข 7 เป็นส่วนของการเชื่อมต่อจอรับภาพโดยการแสดงผลต้องใช้จอ CRT ที่เชื่อมต่อกับบอร์ด FPGA ผ่าน พอร์ต VGA



ภาพประกอบ 4-17 RTL ภายใน FPGA แสดงองค์ประกอบโดยรวมของระบบ

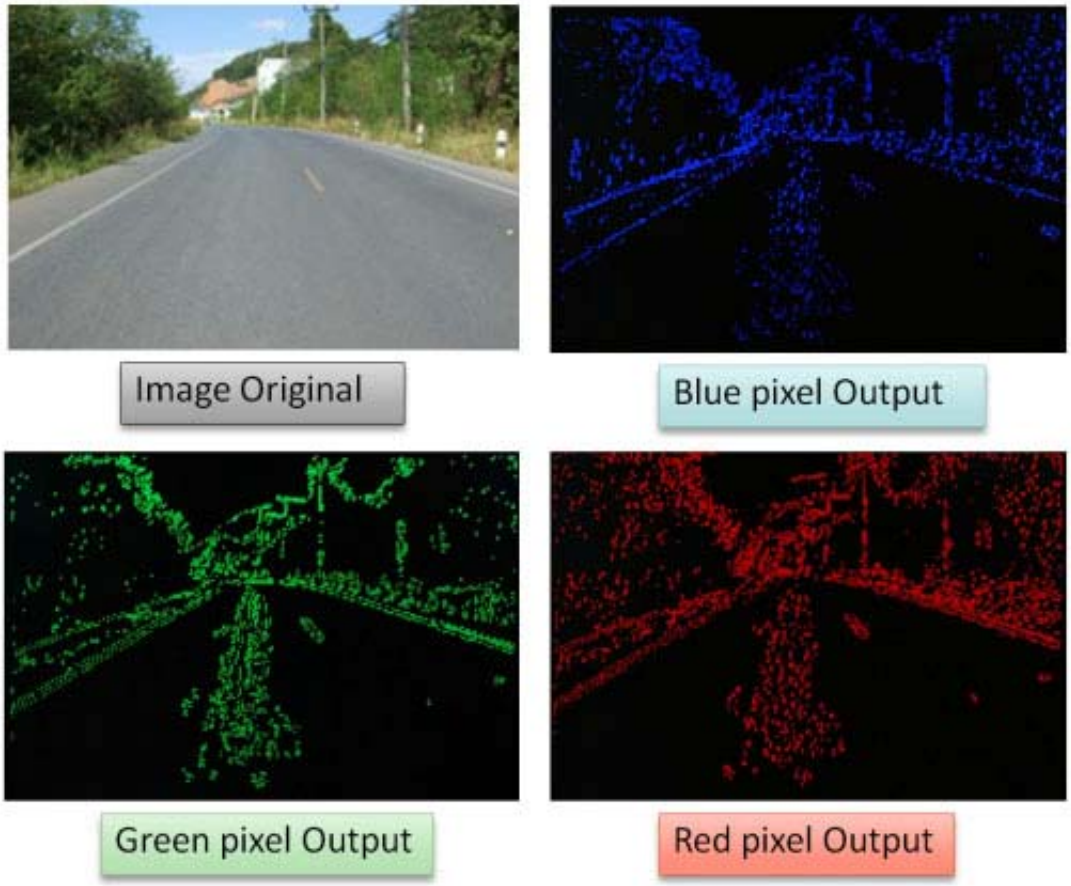


ภาพประกอบ 4-18 RTL โครงสร้างภายใน FPGA ของอัลกอริทึมหาขอบโดยวิธีของ Roberts

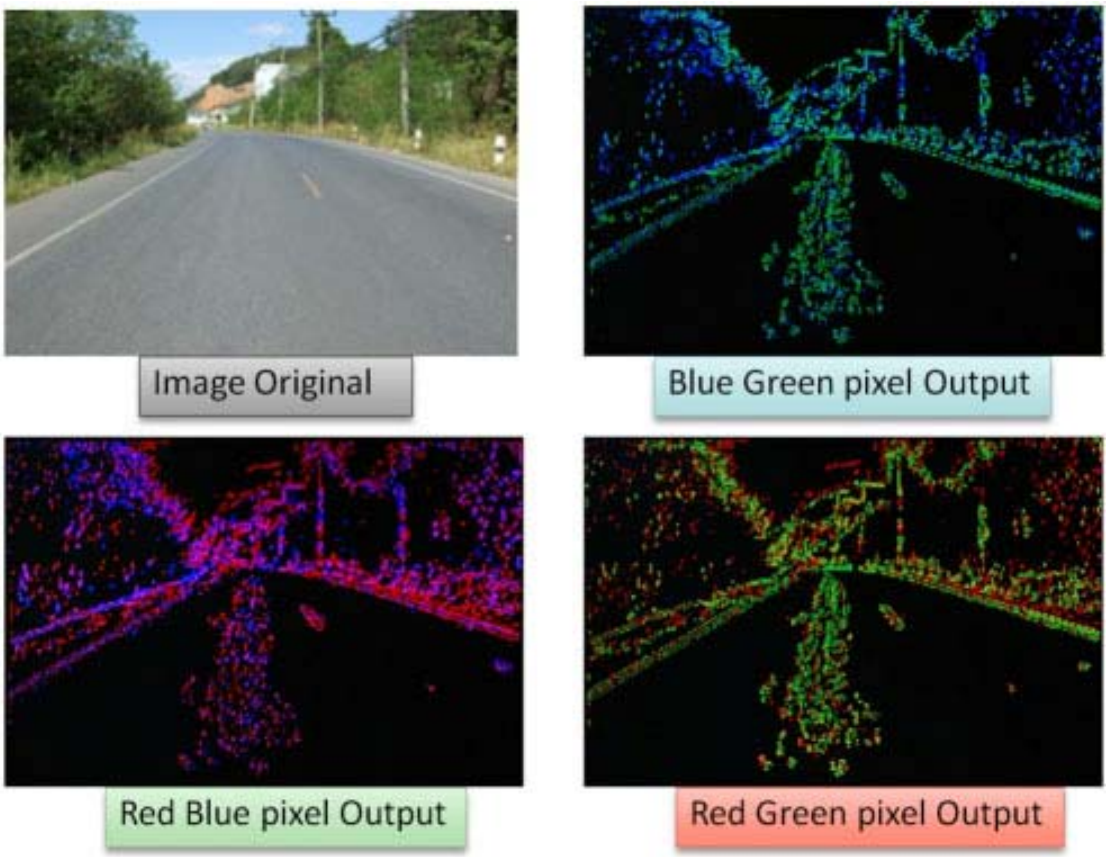
โครงสร้างที่ได้แสดงในภาพประกอบ 4-18 เป็นโมดูลหมายเลข 1-4 ในภาพประกอบ 4-17 จะเชื่อมต่อส่วนของแอดเดรสบัส (Address bus) เพื่อกำหนดตำแหน่งของ พิกเซลของภาพและนำตำแหน่งของค่าในพิกเซลเหล่านั้นไปทำการ Convolution ต่อไปตามอัลกอริทึมหาขอบ โดยผลลัพธ์ที่ออกมาจะเป็นขอบของภาพที่ส่งต่อไปยังโมดูลแสดงผล ผ่านพอร์ต VGA

หลังจากทำการสร้างวงจรลงบนบอร์ด Xilinx รุ่น Vertex-II Pro แล้ว ผลการหาขอบของภาพเป็นดังภาพประกอบ 4-19, ภาพประกอบ 4-20 โดย ภาพประกอบ 4-19 เป็นผลจากการหาขอบภาพโดยวิธี Roberts ในแต่ละสี และภาพประกอบ 4-20 เป็นการแสดงผลการหาขอบรวมกันสองสี

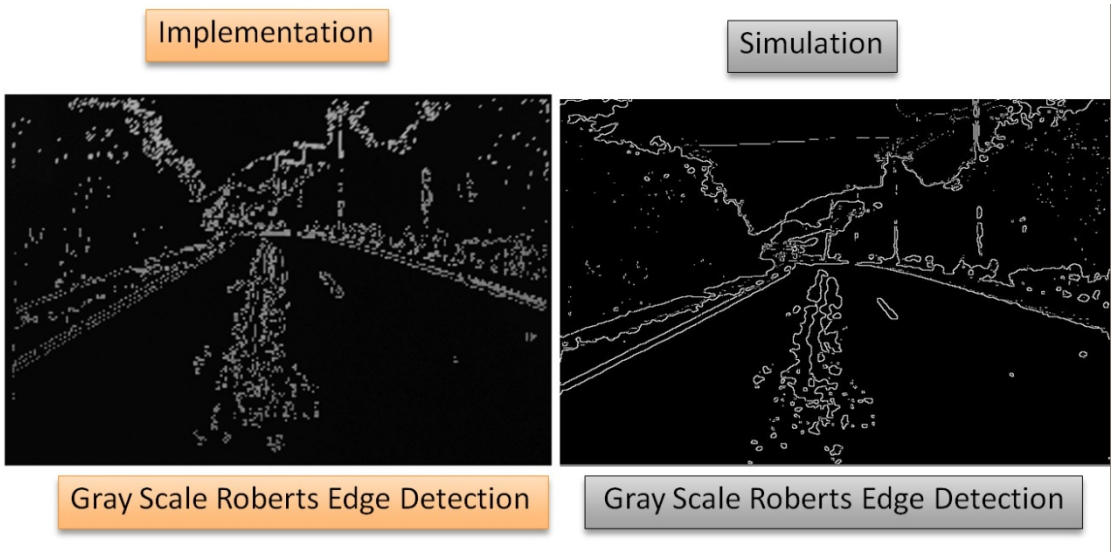
ภาพประกอบ 4-21 เปรียบเทียบให้เห็นผลการหาขอบภาพ โดยวิธีการสร้างวงจรบนฮาร์ดแวร์และ การจำลองแบบใน MATLAB ผลที่ออกมามีลักษณะใกล้เคียงกัน แตกต่างกันที่ การสร้างวงจรในฮาร์ดแวร์ได้มีการแยกกระบวนการทำงานออกเป็นสีต่างๆ ซึ่งมีขนาดของพิกเซลสีละ 8 บิต ส่วนในขั้นตอนการจำลองแบบ มีการแปลงเป็นภาพระดับขาวเทาขนาด 8 บิต ก่อนหาขอบของภาพ



ภาพประกอบ 4-19 ผลการหาขอบภาพแยกแต่ละสีจากการทำงานของฮาร์ดแวร์จริงโดยถ่ายภาพจากจอแสดงผล



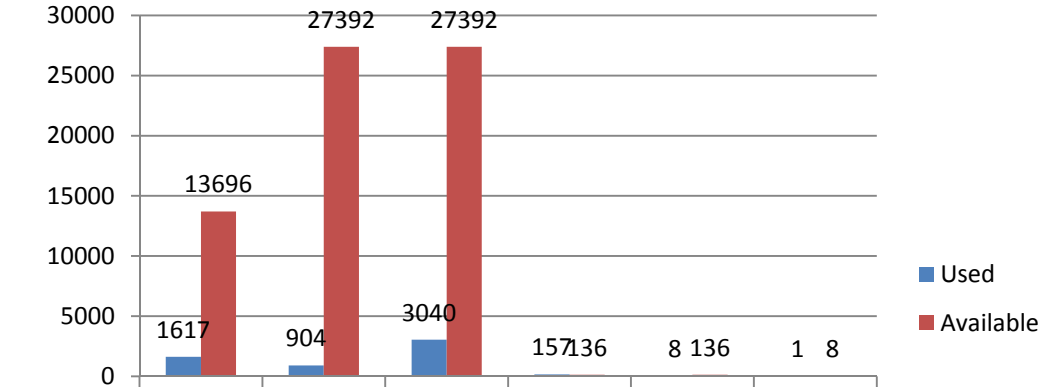
ภาพประกอบ 4-20 ผลการหาขอบภาพโดยพิจารณา 2 สีรวมกันจากการทำงานของฮาร์ดแวร์จริง โดยถ่ายภาพจากหน้าจอแสดงผล



ภาพประกอบ 4-21 เปรียบเทียบการหาขอบภาพระหว่างการทำงานของฮาร์ดแวร์จริง และการทำงานโดย MATLAB

ข้อมูลส่วนสำคัญอีกประการหนึ่งคือ ปริมาณการใช้ทรัพยากรภายใน FPGA เปรียบเทียบกับทรัพยากรทั้งหมดที่มี ดังแสดงในตาราง 4-4 พบว่าระบบที่พัฒนามาการใช้ทรัพยากรค่อนข้างต่ำกว่าที่ระบบมี ยกเว้นในส่วนของ BRAMs ซึ่งในระบบมีอยู่ 136 แต่ระบบใช้ไป 157 โดยส่วนที่เกินมาระบบจะทำการใช้ทรัพยากรส่วนของ Flip Flop และ LUTs เข้ามาประกอบเป็น BRAMs ทดแทน สาเหตุที่ทำให้มีการใช้ BRAMs เป็นจำนวนมากเพราะ ผู้พัฒนาออกแบบโครงสร้างให้มีการเก็บภาพที่ต้องการทดสอบไว้ในส่วนของ FPGA ซึ่งหากมีการพัฒนาถึงขั้นเป็นระบบที่สมบูรณ์ มีการเชื่อมต่อส่วนของกล้องหน้ารถ ส่วนเก็บภาพที่อยู่ภายใน FPGA นี้ก็ไม่มีความจำเป็นอีกต่อไป โดยระบบจะสามารถ Stream ภาพจากกล้องเข้าสู่ FPGA ได้เลย และโดยพื้นฐานกล้องดิจิตอลทั่วๆไประบบภายในกล้องเองก็มีส่วนของหน่วยความจำสำรองไว้สำหรับถ่ายโอนข้อมูลอยู่แล้ว และอีกประการหนึ่งคือ ในงานวิจัยฉบับนี้ผู้วิจัยได้ทำการสร้างโมดูลหาขอบภาพไว้ 4 โมดูล ซึ่งเป็นภาพหมายเลข 1-4 ในภาพประกอบ 4-17 แต่ในการนำไปใช้งานจริง จะมีโมดูลหาขอบภาพเพียงโมดูลเดียวเท่านั้น สาเหตุที่ผู้วิจัยพัฒนาไว้ถึง 4 โมดูล เพื่อต้องการแยกการหาขอบภาพ เป็นสีต่างๆ และภาพขาวเทา เพื่อเปรียบเทียบในขั้นการทดลองเท่านั้น

ตาราง 4-4 ผลการใช้ทรัพยากรเปรียบเทียบกับทรัพยากรที่มีอยู่ใน FPGA



	Number of Slices	Number of Slice Flip Flops	Number of 4 input LUTs	Number of BRAMs	Number of MULT18x 18s	Number of DCMs
Used	1617	904	3040	157	8	1
Available	13696	27392	27392	136	136	8

```
Timing Summary:
---------------
Speed Grade: -7

   Minimum period: 9.535ns (Maximum Frequency: 104.874MHz)
   Minimum input arrival time before clock: 3.703ns
   Maximum output required time after clock: 7.049ns
   Maximum combinational path delay: No path found
```

ภาพประกอบ 4-22 Timing result ของระบบ

จาก Timing Summary ที่ได้จากการวิเคราะห์ พบว่าระบบสามารถประมวลผลได้ 104.874 ล้านพิกเซลต่อวินาที เนื่องจากเป็นการทำงานแบบไปป์ไลน์ 3 สเตจ ซึ่งหากนำไปใช้งานกับกล้องที่มีขนาดของภาพ 640x480 พิกเซล ระบบจะใช้เวลาในการทำงานเพียง 2.929 มิลลิวินาที เท่านั้น ซึ่งหากคำนวณจากรถที่วิ่งด้วยความเร็ว 90 กิโลเมตรต่อชั่วโมง กับกล้องที่มีเฟรมเรต 30 เฟรมต่อวินาที พบว่าระบบต้องทำงานให้เสร็จภายในเวลา 30 มิลลิวินาที ซึ่งผลจากการพัฒนาบน FPGA สามารถทำงานได้ทันเวลาอย่างแน่นอน

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงการสรุปผลและข้อเสนอแนะที่ได้จากการดำเนินการทำงานวิจัย ตลอดจนปัญหาและอุปสรรคที่เกิดขึ้นขณะทำงานวิจัย และให้ข้อเสนอแนะแก่ผู้สนใจจะนำงานวิจัยชุดนี้ไปพัฒนาต่อ

5.1 สรุปผลการวิจัย

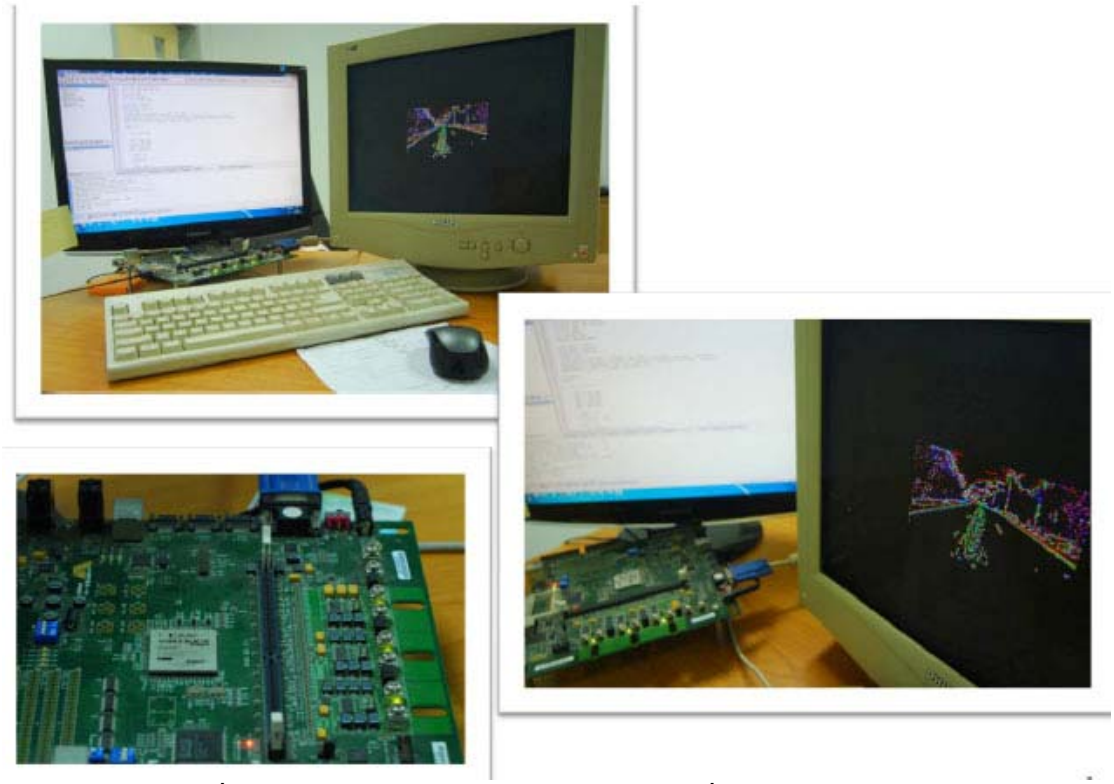
จากการนำเสนอการทดสอบเพื่อหาอัลกอริทึมหาขอบที่เหมาะสำหรับพัฒนาใน FPGA โดยเลือกจากอัลกอริทึมทั้ง 4 ได้แก่ Canny, Prewitt, Sobel และ Roberts สามารถแบ่งอัลกอริทึมออกเป็น 2 กลุ่ม ได้แก่

5.1.1 กลุ่มที่สามารถหาขอบภาพที่มีประสิทธิภาพสูง ซึ่งสามารถมาพิจารณาได้จากผลการทดลอง อัลกอริทึมดังกล่าว ได้แก่ Canny รวมทั้งได้มีงานวิจัย [5] ที่นำการหาขอบภาพโดยวิธี Canny ไปใช้ได้กล่าวไว้ว่าเป็นอัลกอริทึมที่มีประสิทธิภาพสูง แต่จุดด้อยของอัลกอริทึมกลุ่มนี้คือใช้เวลาในการประมวลผลค่อนข้างมาก เมื่อเทียบกับอัลกอรึทึมอื่นๆ และผลจากการหาเส้นตรง พบว่า ระบบได้พบเส้นตรงจำนวนมาก ซึ่งมากกว่าครึ่งของเส้นตรงที่พบ เป็นเส้นตรงที่งานวิจัยฉบับนี้ ไม่ต้องการ เช่น เส้นของรอยเบรค ของรถ เส้นของสายไฟฟ้า ลายต่างๆ บนพื้นถนน ฯลฯ และ อีกประการหนึ่งคือ เมื่อทำการพิจารณา ในเรื่องความถูกต้องของเส้นตรงที่ต้องการแล้ว ผลของการวัดประสิทธิภาพความถูกต้องของเส้นอยู่ที่ 76.67% ซึ่งไม่ได้อยู่ในตำแหน่งที่สูงที่สุด

5.1.2 กลุ่มที่มีโครงสร้างของอัลกอริทึมใกล้เคียงกัน และใช้เวลาในการประมวลผลใกล้เคียงกัน ได้แก่ Prewitt, Soble และ Roberts ซึ่งการใช้เวลาในการประมวลผล Roberts ใช้เวลาน้อยที่สุด คือ 0.526 sec โดยการทดสอบผ่านโปรแกรม MATLAB โดยใช้ภาพต้นฉบับเดียวกัน ซึ่งเวลาในการประมวลผลดังกล่าว เร็วกว่า Canny 3.14 เท่า และประสิทธิภาพเรื่องความถูกต้อง ของ Roberts อยู่ที่ 83.33% ซึ่งอยู่ในอันดับที่สูงที่สุด

จากผลทั้งการทดสอบ และการคำนวณ ผนวกกับโครงสร้างพื้นฐานของอัลกอริทึมในการหาขอบภาพ จึงสรุปได้ว่าการหาขอบภาพวิธีของ Roberts เป็นอัลกอริทึมที่เหมาะสมในการพัฒนาลง FPGA ด้วยโครงสร้างของ Kernel ที่เล็กกว่าอัลกอริทึมแบบอื่น จึงทำให้สามารถพัฒนาใน FPGA ได้ง่ายยิ่งขึ้น

หลังจากที่ได้พัฒนาใน FPGA ระบบก็สามารถทำงานได้อย่างสมบูรณ์และสามารถทำงานได้อย่างมีประสิทธิภาพ จากภาพประกอบ 5-1 แสดงให้เห็นว่าระบบสามารถนำไปพัฒนาใช้ใน FPGA ได้จริงและสามารถแสดงผลภาพได้ใกล้เคียงกับการทดสอบใน MATLAB



ภาพประกอบ 5-1 การเชื่อมต่ออุปกรณ์แสดงผล กับ FPGA บอร์ด เพื่อ แสดงผลการทำงาน

5.2 ข้อเสนอแนะ

5.2.1 งานวิจัยฉบับนี้เป็นการริเริ่มให้มีการพัฒนาระบบค้นหาเส้นตรงในภาพโดยทำการทดสอบเพื่อหาอัลกอริทึมที่มีความเหมาะสมสำหรับพัฒนาใน FPGA ซึ่งในกระบวนการทดสอบผู้วิจัยได้ใช้ภาพจากถนนจริงมาทำการทดสอบ ฉะนั้นผลที่ได้เป็นเครื่องพิสูจน์ว่า ระบบสามารถทำงานได้จริง และสามารถนำไปใช้พัฒนาต่อได้

5.2.2 Mask filter ในงานวิจัยฉบับนี้ เป็นระบบที่ช่วยทำให้คัดกรองเส้นสีขาว สีเหลือง และเป็นอัลกอริทึมที่มีประสิทธิภาพสูง ซึ่งสามารถนำไปปรับใช้สำหรับคัดกรองวัตถุอื่น ที่มีสีชัดเจน

5.2.3 อัลกอริทึมภายในงานวิจัยฉบับนี้ สามารถนำไปพัฒนาใช้ใน FPGA รุ่นอื่นๆ ได้แต่สิ่งสำคัญคือ ควรมีจำนวนทรัพยากร ไม่ต่ำกว่าที่อัลกอริทึมต้องการ

5.2.4 หากต้องนำไปพัฒนาให้เป็นผลิตภัณฑ์ที่สามารถใช้งานได้จริง ต้องมีการพัฒนาส่วนของซอฟต์แวร์เพื่อทำงานในส่วนของ Hough transform เพราะอัลกอริทึมส่วนนี้มีการคำนวณที่ค่อนข้างซับซ้อน หากนำมาพัฒนาใน FPGA อาจทำให้ใช้ทรัพยากรภายใน FPGA ไม่คุ้มค่าได้

5.2.5 การทดสอบอัลกอริทึมที่ได้พัฒนาขึ้น ผู้วิจัยได้ทำการถ่ายภาพจากสถานที่จริงซึ่งสภาพแวดล้อมไม่เป็นอุปสรรค์ต่อทัศนวิสัย และเป็นช่วงกลางวันที่อากาศแจ่มใส และไม่มีอุปสรรคจากเงาไม้ริมถนน ซึ่งหากมีอุปสรรคในเรื่องเงาของต้นไม้ หรือวัตถุแปลกปลอมบนถนน อัลกอริทึมค้นหาเส้นอาจไม่สามารถทำงานได้อย่างมีประสิทธิภาพ

บรรณานุกรม

[1] "สาเหตุของอุบัติเหตุในการจราจรทางบก," [ออนไลน์], เข้าถึงได้จาก: http://www.ipesp.ac.th/learning/supitcha/html/E4-2-1.html. (วันที่สืบค้น 15 มกราคม 2556)

[2] มนตรี กาญจนะเดชะ, เอกสารประกอบการเรียนคอมพิวเตอร์กราฟิก ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่, http://fivedots.coe.psu.ac.th/~montri/Teaching/240-373/

[3] วรุตม์ ขยันกิจ, วรรณรัช สันติอมรทัต, " การออกแบบและพัฒนาวงจรหาขอบภาพด้วยภาษาระดับสูง Impulse C ", การประชุมวิชาการทางวิศวกรรมศาสตร์มหาวิทยาลัยสงขลานครินทร์ ครั้งที่ 7, 21-22 พฤษภาคม 2552

[4] M.J. Jeng, C. Guo, B. Shiau, and P. Hsiao, "Lane Detection System Based on Software and Hardware Codesign", Proceedings of the 4th International Conference on Autonomous Robots and Agents, Wellington, New Zealand, Feb 10-12, 2009.

[5] M. Venkatesan and D. Venkateshwar, " Hardware Acceleration of Edge Detection Algorithm on FPGAs", Celoxica Inc. research papers (2004), http://www.celoxica.com/techlib/files/CEL-W040414XRZ-282.pdf

[6] W. Kayankit and W. Suntiamomtut, "Hardware/Software Co-design for Line Detection Algorithm on FPGA", 6th International Conference on Electrical Engineer/Electronics, Computer, Telecommunications and Information Technology 2009 (ECTI-CON 2009), Vol. 01, 6-9 May 2009, Page(s):604-606.

[7] K. Kokufuta and T. Maruyama, "Real-Time processing of local contrast enhancement on FPGA", . International Conference on Field Programmable Logic and Applications, 2009 (FPL 2009), Aug. 31 2009- Sept.2 2009, Page(s) 288-293.

[8] เจริญชัย ฮวดอุปัต , "การวิเคราะห์ลักษณะผิดปกติบนผิวหนังของใบหู โดยใช้การประมวลผลภาพ An Analysis of Unusual appearances on Ear Skin Using Image Processing" ,2547, ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่, http://dc.oas.psu.ac.th/dcms/files/02700/262249_ch2.pdf

[9] ดุสิตา ล่องเซ่ง, โครงงาน "Plate Detection in Traffic Control Designed for Video Surveillance System", ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ ม.สงขลานครินทร์ วิทยาเขตหาดใหญ่, http://fivedots.coe.psu.ac.th/~kom/wp-content/uploads/2009/07/4810214.pdf

[10] ศิริพร บุญเปลี่ยนพล, " การประยุกต์วิธีโมดิฟายเจอเนอรัลไลฟ์ฮัฟทรานฟอร์ม สำหรับรู้จำภาพจิตรกรรมฝาผนัง : กรณีศึกษา: วัดชนะสงครามราชวรมหาวิหาร กรุงเทพมหานคร/

นครปฐม", สาขาวิชาวิทยาการคอมพิวเตอร์ บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร, 2548, http://www.thapra.lib.su.ac.th/objects/thesis/fulltext/snamcn/Siriporn_Boonplianpol/Chapter2.pdf

[11]จารวี ฉันทสิทธิพร, " การจำแนกชนิดยาเม็ดจากภาพถ่าย โดยใช้เทคนิคเครือข่ายใยประสาท", สาขาวิชาวิทยาการคอมพิวเตอร์ บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร, 2548, http://www.thapra.lib.su.ac.th/objects/thesis/fulltext/snamcn/Siriporn_Boonplianpol/Chapter2.pdfhttp://www.thapra.lib.su.ac.th/objects/thesis/fulltext/snamcn/Jaravee_Chantasitiporn/Chapter3.pdf

[12]K.Mayasandra, H.M. Ladak, and Wei Wang, "A distributed arithmetic hardware architecture for real-time Hough transform based segmentation", Canadian Conference on Electrical and Computer Engineering 2005, 1-4 May 2005, Page(s) 1469-1472.

[13] P. Pankiewicz, W. Powiertowski, and G. Roszak, "VHDL implementation of the lane detection algorithm", 15th International Conference on Mixed Design of Integrated Circuits and Systems, 2008 (MIXDES 2008), 19-21 June 2008, Page(s) 581-584.

ภาคผนวก

ภาคผนวก ก

บทความทางวิชาการที่นำเสนอใน

8th International Conference on Electrical Engineer/Electronics, Computer, Telecommunications and Information Technology, 2011 (ECTI-CON 2011)

17-19 พฤษภาคม 2554 จ. ขอนแก่น


ECTI-CON 2011
KHON KAEN UNIVERSITY
8th
Electrical Engineering/ Electronics, Computer, Telecommunications and Information Technology (ECTI) Association, Thailand - Conference 2011
Khon Kaen, Thailand
May 17-19, 2011
Pullman Khon Kaen Raja Orchid Hotel
ECTI Association
KHON KAEN UNIVERSITY
IEEE
THAILAND SESSION

Signal Processing
(Image Processing)
Paper ID 1440

A Study of the Edge Detection for Road Lane

Worawit Phueakjeen[#1], Nattha Jindapetch[#2], Leang Kuburat[#3], and Nikom Suvanvorn[*4]

[#] *Department of Electrical Engineering and* [*] *Department of Computer Engineering*

Faculty of Engineering, Prince of Songkla University.
Hat Yai, Songkhla 90112

[1]worawit_pergjeen@hotmail.com
[2]nattha.s@psu.ac.th
[3]leang.k@psu.ac.th
[4]kom@coe.psu.ac.th

***Abstract*— This article presents an investigation of an optimum algorithm for edge detection in order to use in the road lane detection process. The main issues, including the speed, the accuracy, and the limited resources, were taken to consider for the realization on the FPGA technology. The edge detection algorithms of Canny, Prewitt, Sobel and Roberts were compared using MATLAB. A number of road images were captured by a video camera with the image size of 640x480 pixels and the frame rate of 30 fps. In addition, a mask filter was applied to remove red, green, and blue values to help the edge detection process be more efficient. From the experimental results, the Canny algorithm was the most time consuming process, and gave too many lines outside the road lane. Among these, the Roberts algorithm is not only the smallest size, but also gained the fastest speed (3.14 times faster than the Canny algorithm) and the most accurate one to detect the lines of the actual road lanes.**

***Keywords*—Edge detection, road lane, FPGA**

I. Introduction

Nowadays the digital camera technology has been developing so quickly. The advanced researches on camera based monitoring and decision are used to benefit mankind. For example, the developments of image processing and computer vision are widely used in the area of security. The development for road lane detection is also very helpful to increase visibility. We can drive more safely. The algorithm for detecting the road lanes needs to be done by high-speed processors because it is constrained by the vehicle running with high-speed. The parallel architecture of FPGA-based implementations have potential to cope with this requirement.

The hardware and software co-designs in [1-5] have been implemented in the FPGA. Kayankit et al [1-2] proved that the Roberts algorithm achieved the smallest area cost compared to Sobel and Prewitt algorithms, but only few benchmark images was tested. Venkatesan et al [3] presented a hardware acceleration of edge detection algorithm on FPGAs, but only the Canny algorithm was considered. Kaszubiak et al [4] presented a real-time vehicle and lane detection with embedded hardware, but the lane detection is applied by using only Hough transform. Jeng et al [5] presented a real-time lane detection system with a processing time of less than 50ms and can detect the lane line correctly under all the conditions whether at day or night. However, there is no discussion of the optimum edge detection.

In this paper, we investigate an optimum algorithm for edge detection in order to use in the road lane detection process. The main issues, including the speed, the accuracy, and the limited resources, were taken to consider for the realization on the FPGA technology. The edge detection algorithms of Canny, Prewitt, Sobel and Roberts were compared using MATLAB. A number of images captured from the real roads are tested to achieve the correct results for the system realization.

II. Edge Detection

Edge detection is a process to determine the perimeter in an image. The edge is caused by the difference in brightness from one point to the others. If the brightness changes sharply, the edge is identified clearly. For road lane detection, the result of applying an edge detector to an image should lead to a set of connected lines that indicate the road lane.

The methods to find the edges can be divided into two categories: the gradient method and the Laplacian method. The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. As shown in Fig. 1(b), the edge is detected because the maximum of the first derivative of the original image in Fig.1(a) is greater than the threshold. The well-known gradient based edge detection algorithms are Sobel, Prewitt, Roberts and Canny. In the other way, the Laplacian method searches for zero-crossings in the second derivative of the image to find edges. The value of zero will be at the position of the edge as shown in Fig. 1(c). The algorithms that use this method are Laplacian of Gaussian, Marrs Hildreth, etc.

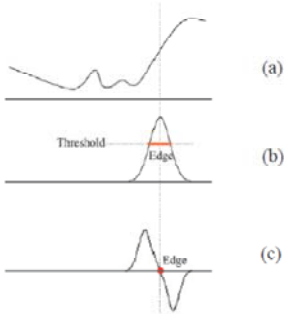


Fig. 1 Edge detection algorithms: (a) an image signal, (b)Gradient edge signal, and (c) Laplacian edge signal.

In this paper, the gradient based algorithms of Sobel, Prewitt, Roberts and Canny are discussed as follows.

A) The algorithms of Sobel, Prewitt and Roberts

Sobel, Prewitt and Roberts algorithms have a common process to perform a 2-dimention spatial gradient measurement on an image and so emphasizes regions of high spatial gradient that correspond to edges. In practice, this can be done by convolving the input image with a pair of convolution kernels. Each algorithm has the corresponding kernels below.

Roberts kernels:

$$Gx = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \qquad Gy = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$$

Prewitt kernels:

$$Gx = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \quad Gy = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Sobel kernels:

$$Gx = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad Gy = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid in the two perpendicular orientations (*Gx* and *Gy*). The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation. Then, an approximate gradient magnitude is computed using:

$$|G| = |Gx| + |Gy|. \qquad (1)$$

The Sobel algorithm is slower to compute than the Roberts algorithm, but its larger convolution kernels smooth the input image to a greater extent and so makes the operator less sensitive to noise. The Prewitt algorithm works in a very similar way to the Sobel algorithm but uses slightly different kernels.

B) The Canny algorithm

The Canny edge detection algorithm is known to many as the optimal edge detector. To find the edges by the Canny algorithm, a series of steps must be followed.

Step 1: Smoothing is the first step to eliminate interference by using the Gaussian filter which can be performed using standard convolution methods.

Step 2: The gradient magnitude is calculated to find the edge strength by taking the same kernels as in the Sobel algorithm. The regions with high spatial derivatives are highlighted.

Step 3: The direction of the edge is computed using the gradient in the x and y directions. The formula for finding the edge direction is just: $= tan^{-1}(Gy/Gx)$.

Step 4: The non-maxima suppression is performed by tracking along the highlighted step 2. Therefore, any pixel that is not at the regions from maximum is suppressed (set it equal to 0). This will give a thin line in the output image.

Step 5: Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, T1, it is set to zero (made a non-edge). If the magnitude is above the high threshold, T2, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above T2.

III. ROAD LANE DETECTION

For road lane detection, the result of applying an edge detector to an image should lead to a set of connected lines that indicate the road lane. Therefore, the process after the edge detection should be the straight line detection. Fig.2 shows a procedure to find the straight lines from an image. The image is firstly converted to the grayscale format. Then, the edge algorithm is applied. Finally, the Hough transform (basic function in MATLAB) is applied to obtain the straight lines.

From Fig.2, the edge detection is the most important part of finding lines on the road. Although there are a number of ways to perform edge detection, these ways must be investigated to obtain the optimum solution for real-time embedded systems. Some algorithms, such as Canny, algorithm provide good results in the amount of lines. Some algorithms use fewer resources. However, these way may not satisfy the application requirements.

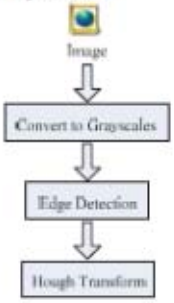


Fig. 2 Steps to find the straight lines in an image.

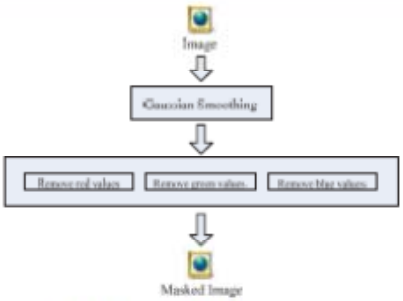


Fig. 3 Steps to create a masked image.

Fig. 4 The colour range.

The design constraints for road lane detection are speed, processing resources, and accuracy in lines on the road. In this paper, the gradient based algorithms of Sobel, Prewitt, Roberts and Canny are compared in terms of speed and accuracy. For the most effective resources, Roberts algorithm has been proved in [1] [3].

In addition to the steps in Fig. 2, a colour mask filter is be applied to the image to increase the system efficiency. The steps to create a masked image shown in Fig.3, include the Gaussian smoothing process and colour removing process. Red, green, and blue values are removed from the image by the reason that the road lanes are white and yellow as the colour range shown in Fig. 4.

IV. Test Results

To investigate the most suitable edge detection algorithm for road lane detection, a set of images taken from the real road by a video camera with the image size of 640x480 pixels and the frame-rate of 30fps were tested. The tests were performed in two cases: the process without mask and the process with colour mask. The results were measured in terms of speed and the number of desired straight lines as follows.

A. Straight line detection without the mask process

The first experiment was performed by using the steps illustrated in Fig. 2. An example of the resulted images is shown in Fig. 5 and 6, and the processing time to find the straight lines in an image is summarized in Table 1.

From an example of the resulted image shown in Fig. 5 and 6, the results may be divided into two groups: the Canny group and the Prewitt, Sobel, and Roberts group. The Canny group has more details of lines than the latter group.

The processing times in the 2nd column of Table 1 were measured in seconds by MATLAB while applying an image to each algorithm. The 3rd column shows the normalized speed calculated by converting the processing time to the corresponding frequency and then normalizing the Canny speed to 1. Obviously, the Prewitt, Sobel, and Roberts group gains up to three times faster than the Canny group. Among these algorithms, Roberts is the fastest algorithm.

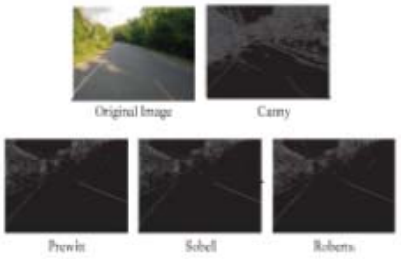


Fig. 5 The image resulted from various algorithms.

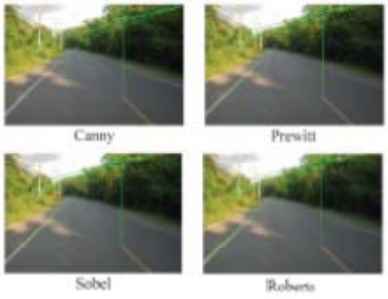


Fig. 6 The result of the straight line detection by different algorithms.

TABLE I
RESULTS OF THE STRAIGHT LINE DETECTION PROCESSING TIMES.

Edge Detection	Processing Time (sec)	Normalized speed
Canny	1.652	1
Prewitt	0.538	3.071
Sobel	0.533	3.099
Roberts	0.526	3.141

B. Straight line detection with the mask process

In this experiment, the steps to create a masked image shown in Fig.3 were applied to remove the red, green, blue colours from the image before finding the edge. Fig. 7 shows how an image is masked. The road lanes are more clearly seen and the other details outside the road are reduced. This makes the edge detection simpler and the straight detection line more accurate.

Fig. 8 shows the results of straight line detection of an image that was masked before finding the edges. We can see that the detected lines are on the road and outside the road. The lines on the road are considered the actual lanes. Although, the Canny algorithm gave more lines, many lines were not the actual lanes.

Fig. 7 A is the image master, B is the masked image, and C is the image for the next step to find the edges.

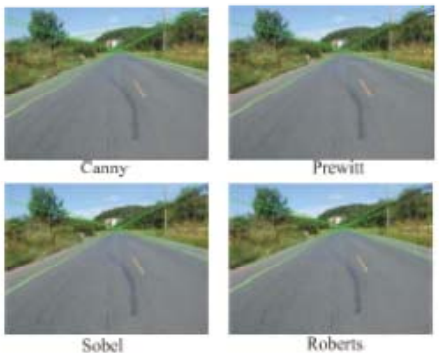


Fig 8 The result of finding the lines with a mask to filter before finding the image edges.

TABLE II

RESULT OF THE ACCURACY OF THE PROCESSING ALGORITHMS TO FIND LINES ON THE ROAD BY RANDOMLY SELECTING 15 FRAMES FROM THE VIDEO FRAME 100TH TO 1500TH.

Edge Detection	Detected Lines	Actual Lanes	True Positive (%)	False Positive (%)
Canny	87	51	76.67	23.33
Prewitt	42	34	73.33	26.67
Sobel	48	36	80.00	20.00
Roberts	67	53	83.33	16.67

Table 2 shows the results in the percentage of detection accuracy of the actual lines on the road per the total detected lines in each algorithm. 15 image frames were randomly selected from the video frames of 100th to 1500th for testing. The percentage of true positive and false positive represents the reliability of algorithms. The algorithms can detect the striped line at the center of road if there is enough information. We can see that the Canny algorithm detected more lines than the other algorithms, but provided the lowest accuracy. This is because some detected lines are outside the road as shown in Fig.8. In contrast, the Roberts algorithm, which is the simplest algorithm, was the most accurate one to detect the lines of the actual road lanes.

V. CONCLUSIONS

The aim of this paper is to find the most appropriate algorithm for the edge detection of the road lanes. Speed, accuracy and the limited of resources were taken into consideration. MATLAB program was used as a tool for analysis. The results from the experiments show that the suitable algorithm in the term of processing time was Prewitt, Sobel and Roberts. Among these, the Roberts algorithm is the most fastest, and 3.14 times faster than the Canny algorithm. In the term of accuracy, the Roberts also gained the most accurate one to find the actual lanes on the road. Although, the Canny algorithm has been known as the best edge algorithm and gave more lines in this test, many lines were not the actual lanes. In the term of resources, the Roberts algorithm has been known as the most simplest edge algorithm compared to the algorithm of Canny, Prewitt and Sobel. This is because the Roberts algorithm uses the 2x2 kernels while the others use the 3x3 for the gradient calculation. The Canny algorithm further performs more complicated processes. From these reasons, the Roberts algorithm is the most suitable edge detection in order to use in the embedded system based road lane detection process.

VI. ACKNOWLEDGEMENT

This work has been supported by High-Performance Embedded Systems research team, Department of Electrical Engineering, Prince of Songkla University.

VII. REFERENCES

[1] Warut Kayankit and Wannarat Santiamornrat, "The Design and Development Cycle to Find the Image Edge with a High Level Language Impulse C", 7th PSU Engineering Conference, 21-22 May 2009.

[2] W. Kayankit and W. Santiamornrat, "Hardware/Software Co-design for Line Detection Algorithm on FPGA", 6th International Conference on Electrical Engineer/Electronics, Computer, Telecommunications and Information Technology 2009 (ECTI-CON 2009), Vol. 01, 6-9 May 2009 ,Page(s):604-606.

[3] M. Venkatesan and D. Venkateshwar, "Hardware Acceleration of Edge Detection Algorithm on FPGAs", Celoxica Inc. research papers (2004), http://www.celoxica.com/techlib/files/CEL-W040414XRZ-282.pdf

[4] J. Kaszubiak, M. Tornow, R.W. Kuhn, B. Michaelis, and C. Knoeppple, "Real-Time Vehicle and Lane Detection with Embedded Hardware", Proceedings of IEEE Intelligent Vehicles Symposium, 2005.

[5] M.J. Jeng, C. Guo, B. Shiau, and P. Hsiao, "Lane Detection System Based on Software and Hardware Co-design", Proceedings of the 4th International Conference on Autonomous Robots and Agents, Wellington, New Zealand, Feb 10-12, 2009.

ECTI-CON 2011

Certificate of Participation

This certifies that

Worawit Phueakjeen

has participated and presented the paper

"A Study of the Edge Detection for Road Lane"

which was peer reviewed at

The 8th Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI) Association, Thailand – Conference 2011 (ECTI-CON 2011) held on May 17-19, 2011, Khon Kaen, Thailand

IEEE THAILAND SECTION

ECTI Association

Asst.Prof.Dr.Boonying Charoen
ECTI-CON 2011 Vice Chair

1110

ประวัติผู้เขียน

ชื่อ สกุล นายวรวิทย์ เผือกจีน

รหัสประจำตัวนักศึกษา 5110120090

วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วิศวกรรมศาสตรบัณฑิต (วิศวกรรมคอมพิวเตอร์)	มหาวิทยาลัยสงขลานครินทร์	2550

ทุนการศึกษา (ที่ได้รับในระหว่างการศึกษา)

ทุนอุดหนุนงานวิจัยมหาวิทยาลัยสงขลานครินทร์ประจำปีการศึกษา 2552

การตีพิมพ์เผยแพร่ผลงาน

W. Phueakjeen, N. Jindapetch, L. Kuburat and N. Suvanvorn, “A Study of Edge Detection for Road Lane”, 8th International Conference on Electrical Engineer/Electronics, Computer, Telecommunications and Information Technology, 2011 (ECTI-CON 2011), 17-19 May 2011 Page(s):995-998.