

การออกแบบการประมวลผลภาพวิดีโอด้วยภาษาระดับสูงสำหรับระบบนับปริมาณรถแบบฝังตัว

A System-Level Design of Video Processing for an Embedded Vehicle Counting System

ชลธิศา เวทโอสถ¹ ณิชฐา จินดาพีชรี¹ และ นิคม สุวรรณวร²
¹ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์
ต.คอหงส์ อ.หาดใหญ่ จ.สงขลา 90112 โทรศัพท์: 0-7428-7045
²ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์
ต.คอหงส์ อ.หาดใหญ่ จ.สงขลา 90112 โทรศัพท์: 0-7428-7076
Email : i_juju10@hotmail.com snattha@eng.psu.ac.th kom@coe.psu.ac.th

บทคัดย่อ

บทความนี้นำเสนอการออกแบบระบบการนับปริมาณรถอย่างง่ายโดยใช้กระบวนการอิมเมจโปรเซสซิง ด้วยภาษาระดับสูง ImpulseC เพื่อช่วยในการออกแบบและพัฒนาาร่วมกันระหว่างซอฟต์แวร์และฮาร์ดแวร์ให้ช่วยลดระยะเวลาและความยุ่งยากในการสร้างวงจรจริงที่ได้สามารถพัฒนาบนเทคโนโลยีเอฟพีจีเอเพื่อเพิ่มประสิทธิภาพในการประมวลผลสัญญาณภาพให้ดียิ่งขึ้น โดยบทความนี้ใช้เทคนิคการลบภาพพื้นหลังด้วยวิธีหาผลต่างเฟรม การแก้ไขมุมมองภาพแบบ Perspective เพื่อให้รถในบริเวณที่ภาพอยู่ไกลจากกล้องซึ่งรถมีขนาดลดลง และทำการนับด้วยการหาพื้นที่จุดสีของรถ ซึ่งสามารถจำลองการทำงานได้จริงบนเทคโนโลยีเอฟพีจีเอ ก่อนนำไปใช้งานบนบอร์ด Virtex II Pro

คำสำคัญ: ImpulseC, การนับปริมาณรถ, การลบภาพพื้นหลัง, เอฟพีจีเอ

Abstract

This paper presents a design methodology of a simple vehicle counting circuit on a Field Programmable Gate Array (FPGA) using system-level language named ImpulseC. By using this, it could accelerate image processing system performance and also reduce the design development time. We use the frame different based background subtraction to separate the vehicles from the road, apply perspective view corrected for vehicles far from the camera. The vehicle count is calculated from the area of normalized pixels. This design flow allows completely simulating on FPGA technology before applying on the Virtex II Pro board.

Keywords: ImpulseC, Vehicle Counting, Background Subtraction, FPGA

1. บทนำ

ปัญหาการจราจรที่ติดขัดเป็นปัญหาสำคัญ โดยเฉพาะในช่วงเวลาเร่งด่วนที่มีปริมาณรถยนต์ในการจราจรมากกว่าช่วงปกติ ทำให้เกิดปัญหามากมาย เช่น ไม่มีความคล่องตัวในการจราจร สิ้นเปลืองพลังงาน เกิดอุบัติเหตุได้ง่าย เกิดปัญหามลภาวะ เป็นต้น หากมีระบบที่สามารถจัดการกับการปล่อยรถออกจากแยกได้อย่างชาญฉลาด ปัญหาเหล่านี้ก็จะได้รับการแก้ไขไปในทางที่ดีขึ้น เป็นเหตุให้เกิดแนวความคิดในการจัดการการจราจรอัจฉริยะ

หลักการการทำงานของระบบนี้สามารถคำนวณความหนาแน่นของปริมาณรถในสี่แยกจราจร เพื่อทำการปล่อยรถยนต์ให้มีประสิทธิภาพมากกว่าในปัจจุบัน ที่ติดกล้องเอาไว้ตามมุมสูงในแยกต่างๆ จะทำหน้าที่จับภาพรถบนท้องถนนแล้วส่งสัญญาณภาพไปประมวลผลภาพบนเซิร์ฟเวอร์ที่ศูนย์กลาง [2]-[4] โดยหากแยกใดมีปริมาณรถหนาแน่น ระบบจะทำการปล่อยสัญญาณไฟเขียวเพื่อให้รถออกไปมากกว่าแยกที่มีปริมาณรถน้อย แต่พบว่าระบบดังกล่าว เซิร์ฟเวอร์ต้องรับภาระงานหนักมากจึงไม่เหมาะกับถนนที่มีทางแยกจำนวนมาก

ในงานวิจัยนี้เป็นส่วนหนึ่งของระบบสัญญาณไฟจราจรอัจฉริยะ คือ ในส่วนของการตรวจนับรถยนต์เพื่อนำไปคำนวณหาความเหมาะสมในการจัดการระบบการจราจร ขั้นตอนการทำงานของระบบเริ่มจาก สัญญาณภาพแต่ละแยกที่ได้มาจะถูกประมวลผลโดยใช้กระบวนการอิมเมจโปรเซสซิง บนบอร์ดเอฟพีจีเอ ที่ช่วยประมวลผลแทนเซิร์ฟเวอร์ จากข้อมูลปริมาณรถยนต์ในแต่ละแยกนี้ เซิร์ฟเวอร์จะสามารถประมวลผลในภาพรวมทั้งระบบเพื่อจัดการจราจรที่มีประสิทธิภาพต่อไป

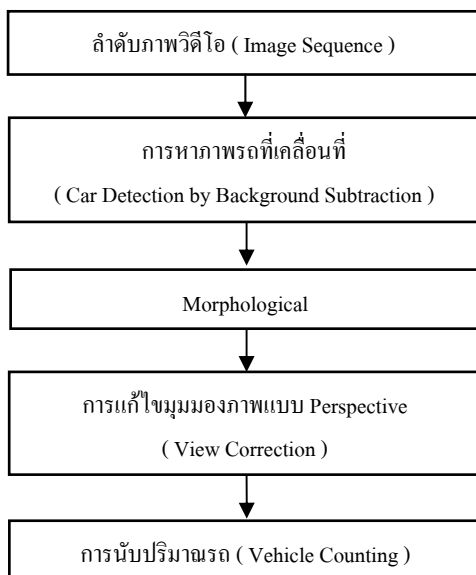
2. งานและทฤษฎีที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้องกับการนับปริมาณรถด้วยกระบวนการอิมเมจโปรเซสซิง ดังในงานวิจัย[2] ใช้เทคนิคการลบภาพพื้นหลังโดยใช้ผลต่างของเฟรม เนื่องจากภาพพื้นหลังค่อนข้างคงที่ นอกจากนี้

งานวิจัยดังกล่าวได้มีการวิเคราะห์หับล้อ และทำการนับรถเมื่อรถผ่านเส้นอ้างอิง แต่มีการนับผิดพลาดประมาณ 10% เนื่องจากการที่สีของภาพพื้นหลังใกล้เคียงกัน งานวิจัย [3] เป็นการตรวจหารถเคลื่อนที่โดยใช้โมเดลของภาพพื้นหลัง (Gaussian Mixture Modeling) รวมถึงการใช้เทคนิคการวิเคราะห์หับล้อ และคาลมาฟิลเตอร์ ในการติดตามรถที่เคลื่อนที่ ซึ่งเป็นกระบวนการที่ค่อนข้างซับซ้อน งานวิจัย [4] เป็นการตรวจหารถเคลื่อนที่โดยเปรียบเทียบจุดสีของรถกับถนน และใช้บริเวณของวัตถุที่เคลื่อนไหวมานับจำนวนจุดสีเพื่อหาพื้นที่ของรถ พบว่ามีความผิดพลาดในการนับรถประมาณ 2% ซึ่งน้อยมาก และมีการหาความกว้างและความยาวของรถเพื่อจำแนกประเภทรถ งานวิจัย [5] เป็นการออกแบบระบบร่วมฮาร์ดแวร์/ซอฟต์แวร์ ในการนับจำนวนรถ โดยใช้วิธีการ 2D FIR ฟิลเตอร์ และฮิสโตแกรมเทรซโซลด์ โดยทำการนับรถเมื่อรถมีการเคลื่อนที่ผ่านบริเวณสี่เหลี่ยมที่กำหนดขึ้น ซึ่งออกแบบในแพลตฟอร์มของ DSP บนเอฟพีจีเอที่เพิ่มประสิทธิภาพในการประมวลผล ซึ่งผู้เขียนได้เลือกใช้เทคนิคการลบภาพพื้นหลัง ด้วยวิธีหาผลต่างของเฟรม [2] เลือกเทคนิคการนับปริมาณรถด้วยวิธีการนับจำนวนจุดสีเพื่อหาพื้นที่ของรถ [4] และพัฒนาอัลกอริทึมการนับปริมาณรถบนเอฟพีจีเอ ที่เพิ่มประสิทธิภาพในการประมวลผล [5]

3. การออกแบบอัลกอริทึมของการนับปริมาณรถ

อัลกอริทึมการนับปริมาณรถ โดยภาพรวมแสดงดังรูปที่ 1 เริ่มต้นแยกรถที่เคลื่อนที่ด้วยการลบภาพพื้นหลังโดยการหาผลต่างระหว่างเฟรม ผลที่ได้นำมาทำกระบวนการ Morphological เพื่อปรับปรุงรูปร่างของวัตถุ จากนั้น การแก้ไขมุมมองภาพแบบ Perspective ของจุดสีในแต่ละแถว เพื่อขยายภาพรถในเลนถนนให้มีสัดส่วนเท่ากันทั่วทั้งภาพ และหาพื้นที่จำนวนจุดสีของรถทั้งหมดเพื่อทำการนับจำนวนรถ โดยเปรียบเทียบพื้นที่จำนวนจุดสีของรถทั้งหมดกับพื้นที่จำนวนจุดสีของรถ 1 คัน ซึ่งจะได้ปริมาณรถทั้งหมดใน 1 เฟรม



รูปที่ 1 แผนภาพอัลกอริทึมการนับปริมาณรถ

3.1 การหาภาพรถที่เคลื่อนที่

เพื่อแยกความแตกต่างของจุดสี ระหว่างรถที่เคลื่อนที่และ background วิธีนี้จะใช้ background ที่มีค่าได้เป็นเฟรมก่อนหน้า

เพื่อง่ายต่อการประมวลผล สำหรับขั้นตอนการประมวลผลจำเป็นจะต้องทำให้ภาพเป็นเกรสเกล เพื่อลดหน่วยความจำที่ใช้ ซึ่งใช้ 8 บิต ซึ่งมีสมการดังนี้

$$|frame_i - frame_{i-1}| > T \quad (1)$$

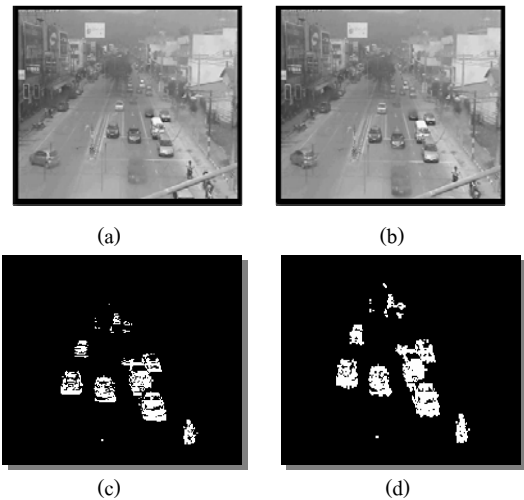
เมื่อ $frame_i$ คือ เฟรมปัจจุบัน

$frame_{i-1}$ คือ เฟรมก่อนหน้า และ

T คือ ค่า threshold

โดยรูปที่ 2 คือ กระบวนการหาผลต่างของเฟรมแล้ว

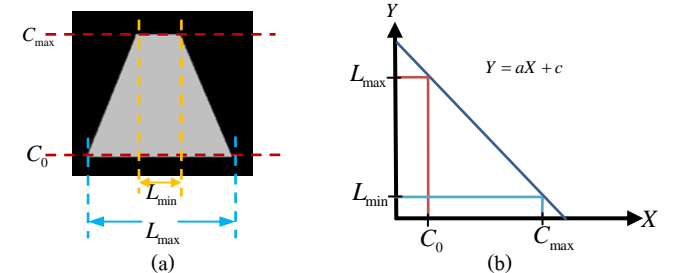
นำมาเทียบกับค่า Threshold ที่มีค่าเท่ากับ 25 ซึ่งได้ภาพรถที่เคลื่อนที่ดังรูปที่ 2 (c) ที่มีช่องว่างของรถจะต้องนำมาทำกระบวนการ Morphological โดยใช้โอเปอร์เรชัน Close ด้วยเทมเพลตเมตริกซ์ขนาด 2×2 เพื่อปิดช่องว่างรถดังรูป 2 (d)



รูปที่ 2 ผลจากการหาผลต่างของเฟรม (a) คือ เป็นภาพเฟรมก่อนหน้า (b) เป็นภาพของเฟรมปัจจุบัน (c) เป็นผลต่างของเฟรมปัจจุบันที่ลบด้วยเฟรมก่อนหน้าแล้วเปรียบเทียบกับค่า Threshold และ (d) การใช้โอเปอร์เรชัน Close เพื่อปิดช่องว่างในรถ

3.2 การแก้ไขมุมมองภาพแบบ Perspective

เนื่องจากภาพจากกล้องวิดีโอ ดังรูปที่ 3 รถที่อยู่บริเวณด้านหน้ากล้องจะมีขนาดใหญ่ และจะมีขนาดเล็กลงเรื่อยๆ เมื่อรถไกลจากกล้องวิดีโอ ซึ่งจำนวนพิกเซลของรถ 1 คัน จะมีขนาดลดลงเมื่อรถอยู่ไกลจากกล้องออกไป ซึ่งมีความสัมพันธ์เป็นไปในลักษณะเชิงเส้น ดังนั้นจึงต้องหาค่าพิกเซลของรถที่จำเป็นต้องนำมาคูณกับจำนวนพิกเซลในแต่ละแถว เพื่อเป็นการขยายภาพของรถในบริเวณที่อยู่ไกลจากกล้องออกไปให้เป็นสัดส่วนเดียวกันทั้งหมด



รูปที่ 3 บริเวณที่ต้องการนับปริมาณรถนำค่าจุดสีพล็อตกราฟเส้นตรง (a) กำหนดตัวแปรในบริเวณที่ต้องการนับ และ (b) นำค่าจุดสีมาพล็อตกราฟเส้นตรง

จากรูปที่ 3 สามารถหาความสัมพันธ์สมการเส้นตรง

$$Y = aX + c \quad (2)$$

เมื่อนำบริเวณของพื้นที่ที่เลือกมาพล็อตกราฟได้ดังรูปที่ 3 (b) ซึ่งค่าจำนวนจุดสีในแต่ละแถวนำมาพล็อตในแกน Y ส่วนจำนวนแถวของบริเวณที่เลือกในภาพจะนำมาพล็อตในแกน X จะได้เส้นตรงดังรูป ซึ่งสามารถนำมาเขียนสมการได้ดังสมการ (3)

$$Y = \left(\frac{L_{\min} - L_{\max}}{C_{\max}} \right) X + L_{\max} \quad (3)$$

ดังนั้น จะได้นอร์มอลไลซ์แฟคเตอร์ ดังสมการ (4)

$$F_x = \frac{L_{\max}}{\left(\frac{L_{\min} - L_{\max}}{C_{\max}} \right) X + L_{\max}} \quad (4)$$

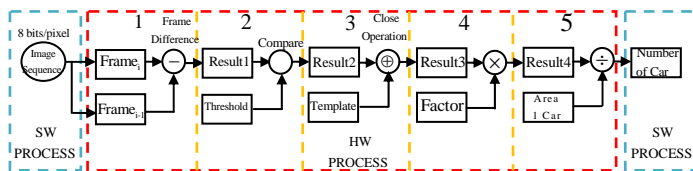
เมื่อ $X = 0$ ถึง C_{\max}

4. การพัฒนาอัลกอริทึมนับปริมาณรถบนเอฟพีจีเอ

การออกแบบและพัฒนากระบวนการรวมฮาร์ดแวร์/ซอฟต์แวร์ [1] ในงานวิจัยนี้ใช้ภาษา ImpulseC [6] โดยแบ่งออกเป็นสองส่วนหลักคือ ระบบซอฟต์แวร์ และระบบฮาร์ดแวร์ โดยมีพื้นฐานมาจากการเขียนด้วยภาษา C ซึ่งกระบวนการทำงานของ CoDeveloper [7] สามารถนำอัลกอริทึมในส่วนของระบบฮาร์ดแวร์มาสร้างในรูปของภาษาทางฮาร์ดแวร์ เป็นภาษา VHDL หรือ Verilog ได้ และสามารถทำการจำลองแบบการทำงานระดับระบบ (System – Level Simulation) ได้

4.1 ออกแบบและพัฒนาวงจรการนับจำนวนรถ

เมื่อพัฒนาการนับปริมาณรถด้วยภาษา ImpulseC แล้ว จากนั้นใช้โปรแกรม Impulse CoDeveloper แปลภาษาเพื่อให้ได้วงจรที่อธิบายด้วยภาษาทางฮาร์ดแวร์ ในที่นี้เลือกใช้เป็นภาษา Verilog จากนั้นใช้ซอฟต์แวร์ที่มีชื่อเรียกว่า Xilinx ISE [8] ช่วยในการสังเคราะห์วงจรและจำลองการทำงาน ซึ่งขั้นตอนการออกแบบวงจรบนเทคโนโลยีเอฟพีจีเอสามารถอธิบายได้ตามไดอะแกรม ดังรูปที่ 4 เป็นการออกแบบวงจรนับปริมาณรถแบบตามลำดับ ที่มี 5 โอเปอร์เรเตอร์ เพื่อให้สามารถรองรับการทำงานแบบไปป์ไลน์ ส่วนในกรณีที่ไม่มีไปป์ไลน์ ผลลัพธ์ Result1-5 ในรูปที่ 4 สามารถใช้หน่วยความจำตัวเดียวกันได้ เพื่อเป็นการประหยัดหน่วยความจำ



รูปที่ 4 ไดอะแกรมอัลกอริทึมการนับปริมาณรถบนเอฟพีจีเอแบบตามลำดับ

5. ผลการทดลอง

เมื่อทดสอบอัลกอริทึมการนับปริมาณรถกับเฟรมภาพวิดีโอขนาด 320×240 ซึ่งเป็นบริเวณแยกห้างสรรพสินค้าคาร์ฟูร์ หาดใหญ่ โดยนับจำนวนจุดสีในทุกแถวของภาพ ในบริเวณที่ต้องการนับปริมาณรถ ในตำแหน่งกล้องที่ ดังรูปที่ 3 (a) ซึ่งมีคุณสมบัติคือ บริเวณที่ต้องการนับ

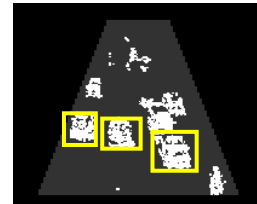
ปริมาณมี 145 แถว และจำนวนจุดสีมีจำนวนตั้งแต่ 53 – 165 จากนั้น นำค่าจำนวนแถวทั้งหมดของบริเวณที่ต้องการนับปริมาณรถมาพล็อตกราฟในแกน X และจำนวนของรถในแต่ละแถวมาพล็อตกราฟในแกน Y ดังรูปที่ 3 (b)

จะได้ว่า $L_{\max} = 165$, $L_{\min} = 53$ และ $C_{\max} = 144$

เมื่อแทนค่าในสมการ (4) จะได้

$$F_x = \frac{165}{\left(\frac{-112}{144} \right) X + 165} \quad \text{เมื่อ } X = 0 - 144$$

ซึ่งต้องนำค่า F_x ไปคูณกับจำนวนจุดสีของรถในแต่ละแถว จะได้บริเวณพื้นที่ของรถทั้งหมด ที่มีการขยายพื้นที่ของรถในบริเวณที่ไกลจากกล้องให้มีขนาดเดียวกัน เมื่อได้พื้นที่โดยรวมของรถทั้งหมดในแต่ละเฟรมภาพ จะต้องนำมาเปรียบเทียบด้วยพื้นที่ของรถ 1 คัน ผลลัพธ์ที่ได้ก็จะเป็นจำนวนรถในแต่ละเฟรมภาพ โดยค่าเฉลี่ยของรถ 1 คัน มาจากพื้นที่เฉลี่ยของรถ 3 คัน ดังรูปที่ 5 (a) - (d)



(a)



(b)



(c)



(d)

รูปที่ 5 การเลือกบริเวณพื้นที่เฉลี่ยของรถ 1 คัน (a) คือ บริเวณที่ทำการเลือกพื้นที่รถแต่ละคัน (b) คือ รถที่เลือกคันที่ 1 (c) คือ รถที่เลือกคันที่ 2 (d) คือ รถที่เลือกคันที่ 3

เมื่อเลือกเฟรมที่มีจำนวนรถ ตั้งแต่ 3 – 10 คัน ที่แตกต่างกัน 10 กรณี ผลลัพธ์ที่ได้แสดงดังตารางที่ 1 และรูปที่ 6

ตารางที่ 1 ผลการนับปริมาณรถตามจำนวน 3 – 10 คัน ที่แตกต่างกัน 10 กรณี จาก CoDeveloper

จำนวนรถภายในเฟรมภาพ (คัน)	จำนวนรถที่นับได้โดยเฉลี่ย 10 กรณี (คัน)
3	3.46
4	4.05
5	5.35
6	5.21
7	6.19
8	6.11
9	7.37
10	8.41



รูปที่ 6 ตัวอย่างผลการนับปริมาณรถในเฟรมต่างๆ

ตารางที่ 2 ผลการสังเคราะห์วงจรบนเฟรมเวิร์ก Virtex II Pro XC2VP30

Latency (s)	Throughput (frame/s)	Slices/ Flip Flops	BRAMs	MULT 18×18s
0.81	1.23	4940/2332	41	8

จากตารางที่ 1 พบว่า การนับปริมาณรถมีการผิดพลาดไม่เกิน 2 คัน ทั้งนี้เนื่องมาจากสภาพกล้องที่ขุ่นมัวทำให้ภายในภาพเห็นรถไม่ชัดเจน รถที่มีสีใกล้เคียงกับพื้นถนนเมื่อหลายครั้งของเฟรมแล้วทำให้บางส่วนของรถหายไป และไม่ได้แยกประเภทรถจักรยานยนต์ รถบัส และรถบรรทุก

จากตารางที่ 2 แสดงจำนวนทรัพยากรที่จะใช้งานบนบอร์ด VirtexII Pro XC2VP30 โดยจำนวน Slices เป็นตัวบอกถึงพื้นที่โดยรวมที่จะใช้งานทรัพยากร ประกอบด้วย จำนวนการใช้ลอจิก Flip Flops , แสดงจำนวนการใช้บล็อกแรม (BRAMs) แต่ละบล็อกแรมมีขนาด 18 กิโลไบต์ , แสดงการใช้บล็อกตัวคูณ (MULT18×18) แต่ละบล็อกตัวคูณมีขนาด 18×18บิต และLatency(s)เป็นตัวบอกความเร็วของวงจรในการประมวลผลภาพต่อหนึ่งเฟรม

เมื่อตรวจสอบความถูกต้องของผลลัพธ์แล้ว จะเข้าสู่ขั้นตอนการพัฒนางานบนเทคโนโลยีเฟรมเวิร์กตามรูปที่ 4 บนโปรแกรม Xilinx ISE เพื่อทำการ Simulation ก่อนนำไปใช้งานจริงบนบอร์ดเฟรมเวิร์ก

6. บทสรุป

บทความนี้แนะนำการออกแบบระบบการนับปริมาณรถอย่างง่ายโดยใช้กระบวนการอิมเมจโปรเซสซิง ด้วยภาษาระดับสูง ImpulseC เพื่อช่วยในการออกแบบและพัฒนาร่วมกันระหว่างซอฟต์แวร์และฮาร์ดแวร์ให้ช่วยลดระยะเวลาและความยุ่งยากในการสร้างวงจรจริงที่สามารถพัฒนาบนเทคโนโลยีเฟรมเวิร์กเพื่อเพิ่มประสิทธิภาพในการ

ประมวลผลสัญญาณภาพให้ดียิ่งขึ้น และผลที่ได้มีความผิดพลาดในการนับปริมาณรถเล็กน้อยไม่เกิน 2 คันในสภาพกล้องที่ขุ่นมัว

7. แนวทางการพัฒนาต่อ

งานวิจัยนี้แนะนำเสนอแนวความคิดเบื้องต้นของการนับปริมาณรถอย่างง่าย ผลที่ได้จากการนับปริมาณรถที่ระบุเป็นจำนวนคันที่กำหนดเป็นขนาดของรถยนต์ส่วนบุคคลเท่านั้น ที่มาจกค่าเฉลี่ยของขนาดของรถยนต์ 1 คัน ที่สุ่มเลือกเป็นจำนวน 3 คัน ไม่มีการแยกขนาดของรถชนิดอื่นๆ ดังนั้น แนวทางการพัฒนาต่อไปควรมีการแยกขนาดของรถชนิดอื่นๆ ที่มีขนาดแตกต่างกันเพิ่มเติม เช่น รถประจำทาง รถจักรยานยนต์ เป็นต้น และหาค่าเฉลี่ยของขนาดของรถ 1 คัน จากขนาดของรถที่สุ่มเลือกในจำนวนที่มากขึ้น เพื่อความถูกต้อง แม่นยำ และมีประสิทธิภาพของระบบการนับปริมาณรถที่จะพัฒนาต่อไป

8. กิตติกรรมประกาศ

ขอขอบคุณเทศบาลนครหาดใหญ่ในความเอื้อเฟื้อภาพวิดีโอสำหรับการทดสอบการออกแบบระบบนับรถบนเฟรมเวิร์ก

9. เอกสารอ้างอิง

- [1] ชีรัช เวียงทอง, “รู้จักกับการออกแบบร่วมกันระหว่างฮาร์ดแวร์-ซอฟต์แวร์เบื้องต้น”, ภาควิชาวิศวกรรมอิเล็กทรอนิกส์ มหาวิทยาลัยเทคโนโลยีมหานคร, 2547.
- [2] Thou-Ho Chen, Yu-Feng Lin, and Tsong-Yi Chen, “Intelligent Vehicle Counting Method Based on Blob Analysis in Traffic Surveillance”, Proceedings of the Second International Conference on Innovative Computing, Information and Control, Kaohsiung, Taiwan, 5-7 Sept. 2007, pp. 238 – 238.
- [3] Erhan Ba, A. Murat Tekalp, and F. Sibel Salman, “Automatic Vehicle Counting from Video for Traffic Flow Analysis”, Proceedings of the 2007 IEEE Intelligent Vehicles Symposium Istanbul, Turkey, June 13-15, 2007.
- [4] Tae-Seung Lee, Eung-Min Lee, Hyeong-Taek Park, Young-Kil Kwag, Sang-Seok Lim, Joong-Hwan Baek, and Byong-Won Hwang, “Implementation of Traffic Flow Measuring Algorithm Using Real-Time Dynamic Image Processing”, Springer-Verlag Berlin Heidelberg, Korea, 2003, pp. 78–87.
- [5] Manoel E.de Lima, Pablo Viana da Silva, Alejandra C Frery, Edna Barros, “A Codesign Approach for a High Performance Vehicle Detector”, High Performance Computing HPC2003-ASTC2003, San Diego, USA, January 2003.
- [6] David Pellerin and Scott Thibault, “Practical FPGA Programming in C”, Prentice Hall, 2005.
- [7] <http://www.impulsec.com>
- [8] <http://www.xilinx.com>