

การตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอที่ใช้ระบบฝังตัว



ว่าที่ ร.ต.ชัยพร ชัชวาลกิจกุล

ศูนย์วิทยทรัพยากร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2550

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

จุฬาลงกรณ์มหาวิทยาลัย

DETECTION OF MOVING OBJECTS IN VIDEO SEQUENCE USING AN EMBEDDED SYSTEM



Acting 2Lt. Chaiyaphorn Chatchawarnkitkul

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Electrical Engineering

Department of Electrical Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2007

Copyright of Chulalongkorn University

501040

หัวข้อวิทยานิพนธ์ การตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอที่ใช้ระบบฝังตัว
โดย ว่าที่ ร.ต. ชัยพร ชัชวาลกิจกุล
สาขาวิชา วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษา อาจารย์ ดร. สุรีย์ พุ่มรินทร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่ง
ของการศึกษาตามหลักสูตรปริญญาโท

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร. บุญสม เลิศนირ์ญวงค์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร. เอกชัย ลีลาธรรม)

..... อาจารย์ที่ปรึกษา
(อาจารย์ ดร. สุรีย์ พุ่มรินทร์)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. สุภาวดี อร่ามวิทย์)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ชัยพร ชีवालกิจกุล: การตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอที่ใช้ระบบฝังตัว (DETECTION OF MOVING OBJECTS IN VIDEO SEQUENCE USING AN EMBEDDED SYSTEM), อ. ที่ปรึกษา: ดร. สุรีย์ พุ่มรินทร์, 61 หน้า

วิทยานิพนธ์นี้เสนอเทคนิคการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอที่ใช้ระบบฝังตัว และเป็นเทคนิคที่ถูกออกแบบด้วยภาษา C/C++ ให้มีการทำงานร่วมกับการส่งข้อมูลผ่านเครือข่ายในระบบฝังตัว ขั้นตอนวิธีเริ่มด้วยการแบ่งภาพในเฟรมหนึ่ง ๆ เป็นแผนภาพย่อยรูปสี่เหลี่ยมจัตุรัส ซึ่งแผนภาพเหล่านั้นจะถูกระบุเป็นแผนภาพเคลื่อนที่ หรือ แผนภาพหยุดนิ่ง จากค่าขีดเริ่มเปลี่ยนที่คำนวณได้ด้วยฮิสโทแกรมของการรวมผลต่างระหว่างเฟรมกำลังสองของแผนภาพทั้งหมดในเฟรมหนึ่งโดยอัตโนมัติ นอกจากนั้นเรลดความซ้ำซ้อนในการรวมผลต่าง ด้วยการคำนวณผลต่างระหว่างเฟรมกำลังสองของแผนภาพเพียงครั้งเดียว และเราสร้างแผนภาพที่สอดคล้องกับวัตถุเคลื่อนที่ในภาพ ณ เฟรมหนึ่ง จากแผนภาพเคลื่อนที่ในอนาคดกับแผนภาพเคลื่อนที่ในอดีตที่มีตำแหน่งร่วมกัน ทำให้เราสามารถกำจัดแผนภาพที่ไม่สอดคล้องกับวัตถุเคลื่อนที่ได้ พร้อมกันนี้เรลดข้อต่อของความต่างระหว่างเฟรมที่มีต่อวิธีการลบภาพพื้นหลัง โดยการจดจำข้อมูลของวัตถุเคลื่อนที่ในเฟรม โดยสรุปการตรวจหาการเคลื่อนที่ของวัตถุในลำดับภาพวิดีโอที่เรานำเสนอสามารถประมวลผลในระบบฝังตัว สำหรับภาพที่มีขนาด 320×240 จุดภาพ พร้อมกับรายงานผลลัพธ์ผ่านเครือข่ายด้วยการแสดงบนเว็บเบราว์เซอร์

ศูนย์วิทยทรัพยากร

จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมไฟฟ้า
สาขาวิชา วิศวกรรมไฟฟ้า
ปีการศึกษา 2550

ลายมือชื่อนิสิต ชัยพร ชีवालกิจกุล
ลายมือชื่ออาจารย์ที่ปรึกษา AS AS

##4870654821: MAJOR ELECTRICAL ENGINEERING

KEY WORD: FRAME DIFFERENCE / DETECTION OF MOVING OBJECT / MOVING PATCH / IMAGE PATCH / HISTOGRAM / EMBEDDED SYSTEM / MEMORY OBJECTS

CHAIYAPHORN CHATCHAWARNKITKUL: DETECTION OF MOVING OBJECTS IN VIDEO SEQUENCE USING AN EMBEDDED SYSTEM, THESIS ADVISOR: SUREE PUMRIN, Ph.D., 61 pp.

This thesis proposes a technique to detect moving objects in video sequence using an embedded system. The program is written in C/C++ language cooperated with transfer data via network program in embedded system. We divide an image frame into square-size blocks of image which are called image patches. There are two types of image patch: a moving image patch and an unmoving image patch. Each type of image patch is defined by a threshold which is automatically calculated from histogram of summation of square of frame difference. We can reduce a redundancy by computing the summation of square of frame difference of each patch only one time. We get a moving object by considering only the moving image patches of the previous frame that correspond to the moving image patches of the next frame. As a result, the ghost moving image is removed. Next, the location of the moving object is stored to reduce a noise degradation comparing to a background subtraction method. Finally, this technique can process a 320×240 pixel image and show the results in web browser.

ศูนย์วิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย

Department Electrical Engineering
Field of study Electrical Engineering
Academic year 2007

Student's signature ... ชัยพร ชีทวณิชกุล
Advisor's signature ... Suree P.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ ด้วยความช่วยเหลือของ ดร. สุรีย์ พุ่มรินทร์ อาจารย์ที่ปรึกษา วิทยานิพนธ์ ซึ่งได้สละเวลาให้คำแนะนำและข้อคิดเห็นต่างๆ ที่ทำให้นิสิตมีแนวความคิดในการทำวิทยานิพนธ์ ผู้วิจัยจึงใคร่ขอขอบพระคุณไว้ ณ ที่นี้

ขอขอบพระคุณรองศาสตราจารย์ ดร. เอกชัย ลีลารัมย์ ประธานกรรมการสอบวิทยานิพนธ์ ที่แนะนำแนวทางและความคิดอันบรรเจิดเสมอมา ผู้ช่วยศาสตราจารย์ ดร. สุภาวดี อร่ามวิทย์ กรรมการสอบวิทยานิพนธ์ ที่ได้สละเวลาตรวจสอบและให้คำแนะนำเพื่อให้วิทยานิพนธ์ฉบับนี้สมบูรณ์ยิ่งขึ้น และขอขอบพระคุณคณาจารย์ทุกท่านในห้องปฏิบัติการออกแบบและประยุกต์วงจรรวม ภาควิชาวิศวกรรมไฟฟ้า ที่ได้ประสิทธิ์ประสาทความรู้พื้นฐาน อันเป็นพื้นฐานในการศึกษาและทำวิทยานิพนธ์นี้

ขอกราบขอบพระคุณบิดา มารดา พี่ชาย พี่สาว ที่เป็นกำลังใจและกำลังทรัพย์ตลอดเวลา รวมทั้งให้โอกาสผู้วิจัยได้ศึกษาต่อในระดับปริญญาโทมาบัดนี้

ขอขอบคุณเพื่อนๆ รุ่นพี่ รุ่นน้องในห้องปฏิบัติการวิจัยออกแบบและประยุกต์วงจรรวม ทั้งในภาควิชาวิศวกรรมไฟฟ้า ที่ได้ให้กำลังใจและคำปรึกษา จนผู้วิจัยได้ทำวิทยานิพนธ์นี้ได้สำเร็จสมบูรณ์

สุดท้ายนี้ ขอขอบคุณห้องปฏิบัติการวิจัยออกแบบและประยุกต์วงจรรวม ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย สำหรับทรัพยากรต่างๆ ในการศึกษา ค้นคว้าและวิจัย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ	ฉ
สารบัญ	ช
สารบัญตาราง.....	ฅ
สารบัญภาพ	ญ
คำอธิบายสัญลักษณ์.....	ฎ
บทที่ 1 บทนำ.....	1
1.1 แนวเหตุผลและความเป็นมา	1
1.2 งานวิจัยที่เกี่ยวข้อง	1
1.3 เทคนิคที่นำเสนอ	3
1.4 วัตถุประสงค์	4
1.5 ขอบเขตของวิทยานิพนธ์	4
1.6 ขั้นตอนการดำเนินงาน	4
1.7 ประโยชน์ที่คาดว่าจะได้รับ	5
บทที่ 2 หลักการที่ใช้ในวิทยานิพนธ์	6
2.1 การระบุคุณลักษณะแฉงภาพ	6
2.1.1 การคำนวณค่าขีดเริ่มเปลี่ยน	8
2.2 การจดจำกลุ่มแฉงภาพเคลื่อนที่	16
2.3 การรวมผลต่างระหว่างเฟรมกำลังสอง	17
บทที่ 3 การพัฒนางานวิจัย.....	26
3.1 โครงสร้างโปรแกรมตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอ	26
3.2 การส่งภาพผ่านเครือข่าย	29
บทที่ 4 การทดสอบโปรแกรมตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอ.....	31
4.1 การทดสอบการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอ	31
4.1.1 การทดสอบที่ลานจอดรถหน้าตึกไฟฟ้าแรงสูง	32
4.1.2 การทดสอบบนถนนพหลโยธิน	33
4.1.3 การทดสอบบนถนนพญาไท	33

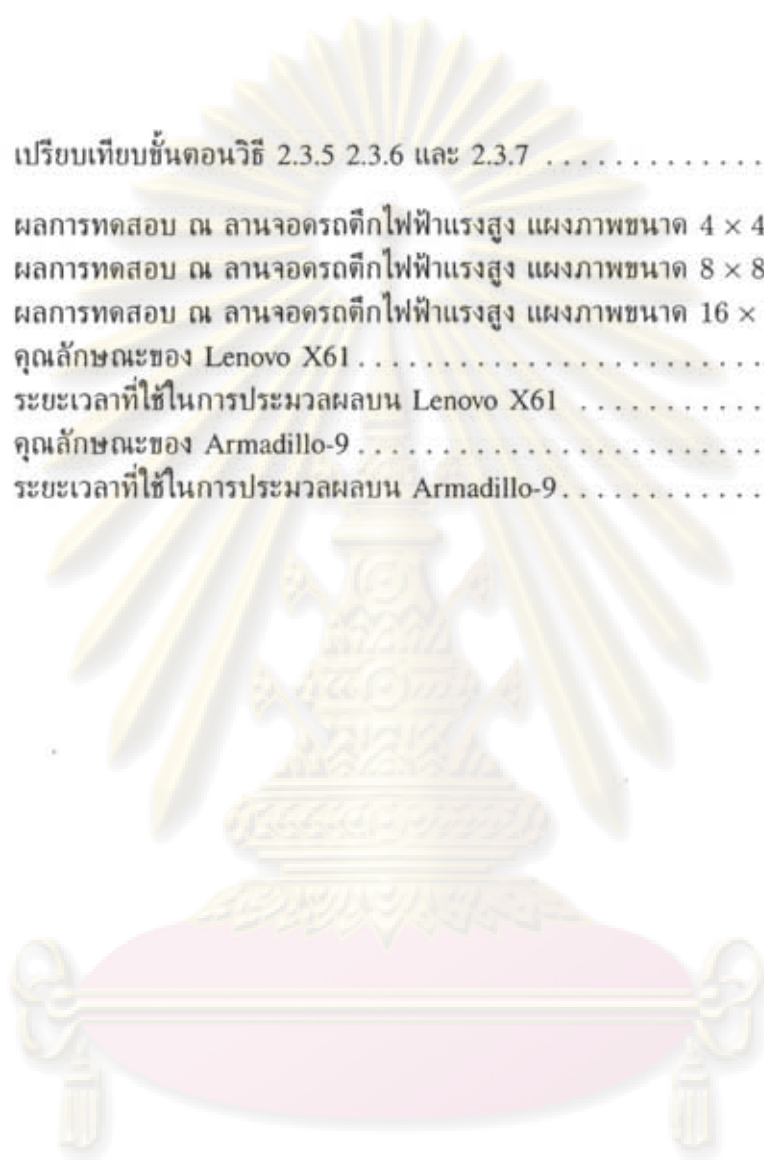
4.1.4	การทดสอบบริเวณคณะวิศวกรรมศาสตร์	43
4.2	การทดสอบเพื่อลดขนาดไฟล์ผลลัพธ์	46
4.3	การทดสอบด้านเวลาที่ใช้ในการประมวลผล	48
4.3.1	การทดสอบบนเครื่องคอมพิวเตอร์	48
4.3.2	การทดสอบบนระบบฝังตัว	49
บทที่ 5	ข้อสรุปและข้อเสนอแนะ	52
5.1	ข้อสรุป	52
5.2	ข้อเสนอแนะ	53
รายการอ้างอิง	54
ภาคผนวก	55
ก	บทความที่ได้รับการตีพิมพ์ในการประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 30	56
ประวัติผู้เขียนวิทยานิพนธ์	61



ศูนย์วิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

	หน้า
ตารางที่ 2.1	เปรียบเทียบขั้นตอนวิธี 2.3.5 2.3.6 และ 2.3.7 25
ตารางที่ 4.1	ผลการทดสอบ ณ ลานจอดรถตึกไฟฟ้าแรงสูง แผงภาพขนาด 4 × 4 จุดภาพ 32
ตารางที่ 4.2	ผลการทดสอบ ณ ลานจอดรถตึกไฟฟ้าแรงสูง แผงภาพขนาด 8 × 8 จุดภาพ 32
ตารางที่ 4.3	ผลการทดสอบ ณ ลานจอดรถตึกไฟฟ้าแรงสูง แผงภาพขนาด 16 × 16 จุดภาพ 33
ตารางที่ 4.4	คุณลักษณะของ Lenovo X61 48
ตารางที่ 4.5	ระยะเวลาที่ใช้ในการประมวลผลบน Lenovo X61 48
ตารางที่ 4.6	คุณลักษณะของ Armadillo-9 50
ตารางที่ 4.7	ระยะเวลาที่ใช้ในการประมวลผลบน Armadillo-9 51



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

	หน้า
รูปที่ 1.1 การเชื่อมต่อระหว่างระบบฝังตัวกับเครือข่ายเฝ้าระวังภัยอัตโนมัติ	2
รูปที่ 2.1 ปัญหาค่าขีดเริ่มเปลี่ยนของความต่างระหว่างเฟรม	7
รูปที่ 2.2 ปัญหาขอบของความต่างระหว่างเฟรม	7
รูปที่ 2.3 ตัวอย่างการแบ่งภาพในเฟรมหนึ่งออกเป็นแผนภาพย่อย	8
รูปที่ 2.4 ฮิสโทแกรมการรวมผลต่างกำลังสองของเฟรมที่มีวัตถุเคลื่อนที่	9
รูปที่ 2.5 ฮิสโทแกรมการรวมผลต่างกำลังสองของเฟรมที่ไม่มีวัตถุเคลื่อนที่แบบที่ 1	10
รูปที่ 2.6 ฮิสโทแกรมการรวมผลต่างกำลังสองของเฟรมที่ไม่มีวัตถุเคลื่อนที่แบบที่ 2	10
รูปที่ 2.7 ตั้งอย่างวัตถุเคลื่อนที่ ณ เวลาต่างกัน	11
รูปที่ 2.8 ตั้งอย่างผลต่างระหว่างเฟรมกำลังสอง	12
รูปที่ 2.9 ผลการอินเตอร์เซกชันระหว่างภาพในรูปที่ 2.8(ก) กับภาพในรูปที่ 2.8(ข)	12
รูปที่ 2.10 ปัญหาความไม่สอดคล้องกันระหว่างแผนภาพกับภาพ	13
รูปที่ 2.11 ภาพที่ได้จากขั้นตอนวิธี (2.1.1) ร่วมกับข้อมูลจากเฟรมที่ 17, 18 และ 19	14
รูปที่ 2.12 ปัญหาข้อมูลขาดหายเมื่อ $D_{t,p} > \delta_{min}$ แต่ $D_{t,p} < \delta_{max}$	15
รูปที่ 2.13 การดึงข้อข้อมูลที่ $D_{t,p} > \delta_{min}$ แต่ $D_{t,p} < \delta_{max}$	15
รูปที่ 2.14 ตัวอย่างการแบ่งแผนภาพ เพื่อการคำนวณ $D_{t,p}$	18
รูปที่ 2.15 การคำนวณ $D_{t,p}$ โดยใช้ขั้นตอนวิธี 2.3.5	19
รูปที่ 2.16 การคำนวณ $D_{t,p}$ โดยใช้ขั้นตอนวิธี 2.3.6	21
รูปที่ 2.17 การคำนวณ $D_{t,p}$ โดยใช้ขั้นตอนวิธี 2.3.7	24
รูปที่ 3.1 โครงสร้างโปรแกรมตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดิทัศน์โดยใช้ระบบฝังตัว	26
รูปที่ 3.2 การไหลข้อมูลการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดิทัศน์โดยใช้ระบบฝังตัว	27
รูปที่ 3.3 การรายงานการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดิทัศน์โดยใช้ระบบฝังตัว	30
รูปที่ 4.1 ทดสอบด้วยแผนภาพขนาด 4×4 จุดภาพ ณ ลานจอดรถหน้าตึกไฟฟ้าแรงสูง	34
รูปที่ 4.2 ทดสอบด้วยแผนภาพขนาด 8×8 จุดภาพ ณ ลานจอดรถหน้าตึกไฟฟ้าแรงสูง	35
รูปที่ 4.3 ทดสอบด้วยแผนภาพขนาด 16×16 จุดภาพ ณ ลานจอดรถหน้าตึกไฟฟ้าแรงสูง	36
รูปที่ 4.4 ทดสอบด้วยแผนภาพขนาด 4×4 จุดภาพ บนถนนพหลโยธิน	37
รูปที่ 4.5 ทดสอบด้วยแผนภาพขนาด 8×8 จุดภาพ บนถนนพหลโยธิน	38
รูปที่ 4.6 ทดสอบด้วยแผนภาพขนาด 16×16 จุดภาพ บนถนนพหลโยธิน	39
รูปที่ 4.7 ทดสอบด้วยแผนภาพขนาด 4×4 จุดภาพ บนถนนพญาไท	40
รูปที่ 4.8 ทดสอบด้วยแผนภาพขนาด 8×8 จุดภาพ บริเวณคณะวิศวกรรมศาสตร์	41
รูปที่ 4.9 ทดสอบด้วยแผนภาพขนาด 16×16 จุดภาพ บริเวณคณะวิศวกรรมศาสตร์	42
รูปที่ 4.10 ทดสอบด้วยแผนภาพขนาด 4×4 จุดภาพ บริเวณคณะวิศวกรรมศาสตร์	43
รูปที่ 4.11 ทดสอบด้วยแผนภาพขนาด 8×8 จุดภาพ บริเวณคณะวิศวกรรมศาสตร์	44
รูปที่ 4.12 ทดสอบด้วยแผนภาพขนาด 16×16 จุดภาพ บริเวณคณะวิศวกรรมศาสตร์	45

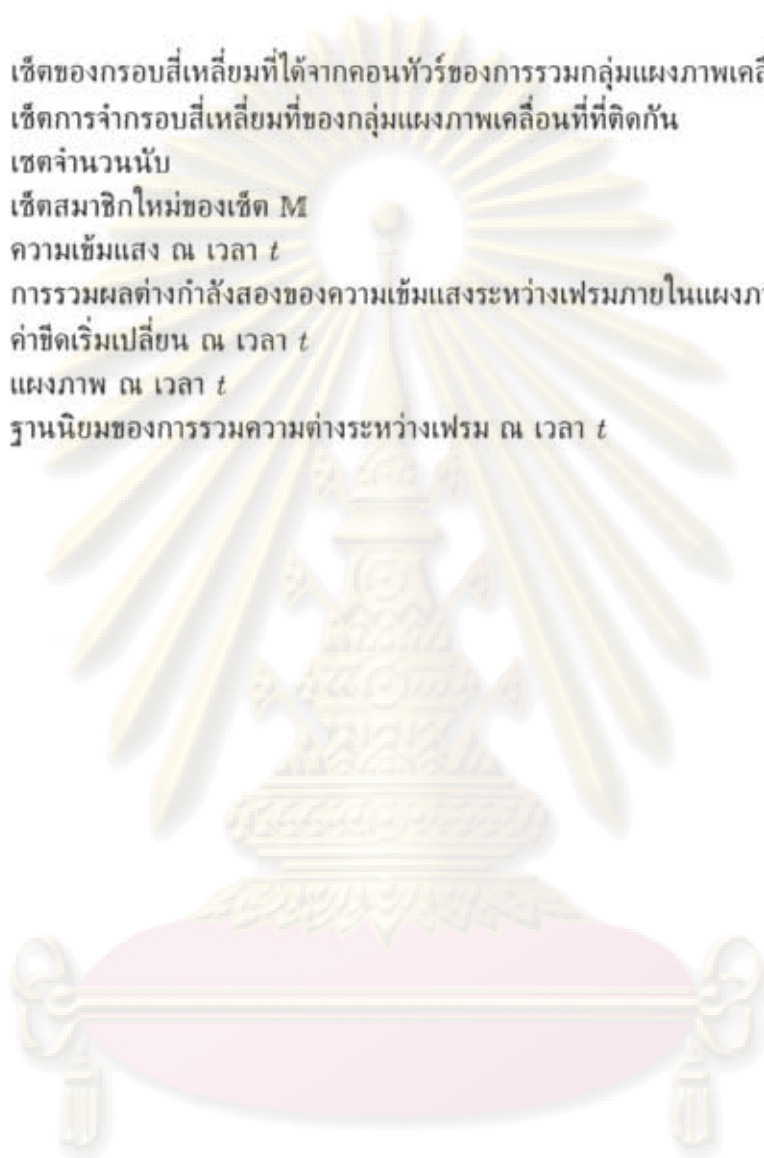
รูปที่ 4.13	ผลของการลดขนาดไฟล์ด้วยเทคนิคที่นำเสนอ ในเหตุการณ์ที่ 1	46
รูปที่ 4.14	ผลของการลดขนาดไฟล์ด้วยเทคนิคที่นำเสนอ ในเหตุการณ์ที่ 2	47
รูปที่ 4.15	บอร์ด Armadillo-9 ที่ใช้เป็นระบบฝังตัว	49
รูปที่ 4.16	บอร์ด Armadillo-9 ที่ใช้เป็นระบบฝังตัว	50



ศูนย์วิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย

คำอธิบายสัญลักษณ์

L	เซตของกรอบสี่เหลี่ยมที่ได้จากคอนทัวร์ของการรวมกลุ่มแผนภาพเคลื่อนที่ที่ติดกัน
M	เซตการจำกรอบสี่เหลี่ยมที่ของกลุ่มแผนภาพเคลื่อนที่ที่ติดกัน
N	เซตจำนวนนับ
Q	เซตสมาชิกใหม่ของเซต M
I_t	ความเข้มแสง ณ เวลา t
$D_{t,p}$	การรวมผลต่างกำลังสองของความเข้มแสงระหว่างเฟรมภายในแผนภาพ p ณ เวลา t
δ_t	ค่าขีดเริ่มเปลี่ยน ณ เวลา t
p_t	แผนภาพ ณ เวลา t
$mode_{D_t}$	ฐานนิยมของการรวมความต่างระหว่างเฟรม ณ เวลา t



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

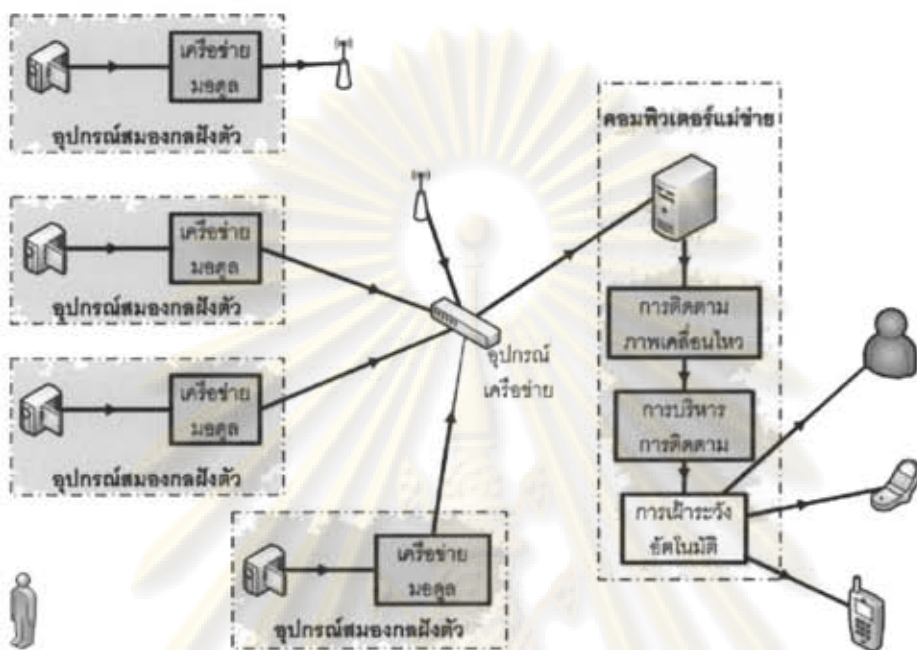
บทนำ

1.1 แนวเหตุผลและความเป็นมา

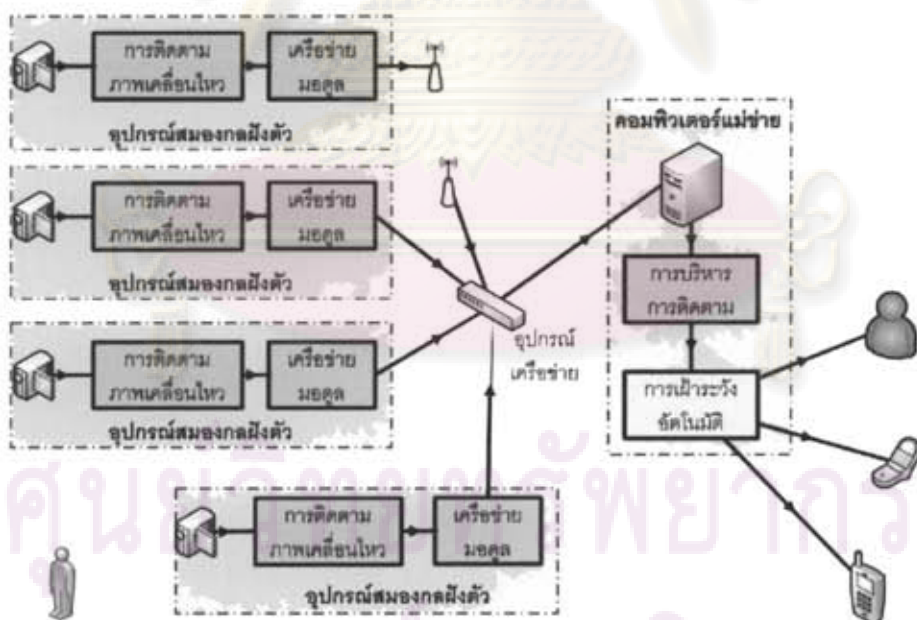
การติดตั้งกล้องวิดีโอในสถานที่ซึ่งยากต่อการติดตั้งสายส่ง สามารถทำได้ด้วยการส่งข้อมูลผ่านระบบไร้สาย และลดจำนวนข้อมูลที่ต้องส่งด้วยการเข้ารหัสก่อนส่งผ่านมอดูมเครือข่าย ดังรูปที่ 1.1(ก) ด้วยวิธีนี้เราจะได้รับข้อมูลตลอดเวลาโดยไม่สนใจความสำคัญของเหตุการณ์ แม้ว่าข้อมูลส่วนใหญ่อาจจะเป็นภาพฉากหลังมากกว่ากิจกรรมที่เกิดขึ้นในบริเวณนั้น แต่บางสถานการณ์ เราสนใจเฉพาะข้อมูลของวัตถุเคลื่อนที่ในลำดับภาพวิดีโอ ซึ่งเป็นข้อมูลที่จะถูกส่งผ่านทางเครือข่ายของระบบเฝ้าระวัง ดังรูปที่ 1.1(ข) การตรวจจับการเคลื่อนไหวในลำดับภาพวิดีโอโดยใช้ระบบฝังตัว เพื่อให้ระบบฝังตัวสามารถคัดกรองเหตุการณ์สำคัญได้ จึงเป็นขั้นตอนที่เข้ามามีบทบาท โดยอาจให้ข้อมูลของวัตถุเคลื่อนที่ในลำดับภาพวิดีโอไปเข้ารหัสก่อนส่งข้อมูล เพื่อลดขนาดของข้อมูลก่อนส่งไปยังเครือข่ายของระบบเฝ้าระวัง และลดเวลาที่ใช้ในการส่งข้อมูลจากระบบฝังตัว ทำให้สามารถยืดอายุพลังงานที่อุปกรณ์ของระบบฝังตัว ทั้งนี้ ขั้นตอนวิธีในการตรวจจับการเคลื่อนไหวในลำดับภาพวิดีโอจะต้องใช้เวลาประมวลผลน้อย เพื่อไม่สูญเสียพลังงานในส่วนนี้มากเกินไป

1.2 งานวิจัยที่เกี่ยวข้อง

งานวิจัยเกี่ยวกับการตรวจหาวัตถุเคลื่อนที่ในระบบฝังตัวที่น่าสนใจ มีสองงานคือ Fleck [1] กับ Lu [2] การตรวจหาวัตถุเคลื่อนที่ในระบบฝังตัวของ Fleck [1] พัฒนาระบบฝังตัวที่สามารถติดตามการเคลื่อนที่แบบเวลาจริง สำหรับระบบฝังตัว Fleck [1] ใช้ Xilinx Spartan-III FPGA กับ Motorola PowerPC 200MHz ในการประมวลผลเพื่อการทำงานได้ทันเวลาจริง ก่อนการติดตามด้วยตัวกรองอนุภาค Fleck [1] ใช้การลบภาพพื้นหลังแบบปรับตัวได้ในการค้นหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอ ซึ่งต้องมีการปรับพารามิเตอร์ให้เหมาะสมในแต่ละสถานการณ์ และต้องใช้กล้องที่สามารถปรับระบบปรับแสงขาวอัตโนมัติ ซึ่งเป็นข้อจำกัดในงานวิจัยของ Fleck [1] ที่จะต้องใช้อุปกรณ์ราคาแพง งานวิจัยที่ใช้อุปกรณ์ในราคาถูกกว่า Fleck [1] อย่างการตรวจหาวัตถุเคลื่อนที่ในระบบฝังตัวของ Lu [2] พัฒนาระบบฝังตัวที่สามารถตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอใช้วิธีการไหลเชิงแสง (optical flow) เพื่อความคงทนต่อการเปลี่ยนแปลงแสงเล็กน้อย และความคงทนต่อการขยับกล้องบางครั้ง Lu [2] คำนวณการไหลเชิงแสงโดยประมาณเพื่อความรวดเร็วในการคำนวณ ทำให้งานวิจัยนี้สามารถรองรับความเร็ว 15 ภาพต่อวินาที โดยใช้ตัวประมวลผล Xscale 400 MHz แต่การประมาณค่าของการไหลเชิงแสงของ Lu [2] ต้องกำหนดค่าขีดเริ่มเปลี่ยนโดยไม่มีหลักการในการกำหนดขนาดที่เหมาะสม และไม่รองรับวัตถุเคลื่อนที่เข้ามาใกล้หน้ากล้อง การตรวจหา



(ก) ที่ระบบฝังตัวไม่มีการตรวจจับการเคลื่อนที่



(ข) ที่ระบบฝังตัวมีการตรวจจับการเคลื่อนที่

รูปที่ 1.1 การเชื่อมต่อระหว่างระบบฝังตัวกับเครือข่ายเฝ้าระวังภัยอัตโนมัติ

ศูนย์วิจัยและพัฒนา
จุฬาลงกรณ์มหาวิทยาลัย

วัตถุเคลื่อนที่ในลำดับภาพวิดีโอของชัยพร [3] ปรับปรุงงานของ Lu [2] โดยแสดงให้เห็นช่วงของพารามิเตอร์ที่กำหนดได้กว้าง และแก้ปัญหาวัตถุเคลื่อนที่ใกล้บริเวณหน้ากล้องได้ ถึงอย่างไรก็ตามวิธีการไหลเชิงแสงเป็นวิธีการคำนวณที่ใช้เวลาในการประมวลผลนาน เพราะในขั้นตอนการคำนวณต้องกรองภาพให้เรียบด้วยตัวกรองเกาส์เซียน 2 ครั้ง แล้วคำนวณหาขอบในแนวนอนกับแนวตั้งของภาพ และผลต่างระหว่างภาพของสองเฟรมที่ลำดับติดกัน ก่อนขั้นตอนวิธีการคำนวณโดยประมาณค่าลำดับชั้นเมทริกซ์ (rank of matrix) ตามขั้นตอนวิธีของ A. Benedetti [4] เพื่อนำค่าลำดับชั้นของเมทริกซ์มากำหนดคุณสมบัติของแผนภาพนั้น ซึ่งเป็นข้อจำกัดของ Lu [2] กับ ชัยพร [3]

1.3 เทคนิคที่นำเสนอ

การคำนวณในระบบฝังตัวมีข้อจำกัดด้านพลังงาน ความเร็วของหน่วยประมวลผล หน่วยความจำ และหน่วยประมวลผลในระบบฝังตัวส่วนมากสามารถคำนวณเลขคณิตจุดลอยตัวได้ (floating point arithmetic) แต่ใช้เวลานานมาก ต่างจากการคำนวณเลขคณิตจุดตรึง (fixed point arithmetic) ดังนั้น การตรวจหาวัตถุเคลื่อนที่ในวิทยานิพนธ์นี้จึงต้องคำนวณเลขคณิตจุดตรึง และวิธีการไม่ซับซ้อน เพื่อลดเวลาการประมวลผล

การลบภาพพื้นหลัง (background subtraction) เป็นการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพที่มีงานวิจัยมาก แต่เป็นการคำนวณทางสถิติต้องการคำนวณเลขคณิตจุดลอยตัว ไม่คงทนต่อการเปลี่ยนแปลงของแสงที่ละน้อย และไม่คงทนต่ออุบัติเหตุที่ทำให้กล้องวิดีโอทัศนชัดขยับเล็กน้อย แม้มีวิธีการชดเชยข้อเสียของการลบภาพพื้นหลัง แต่วิธีการชดเชยนั้นทำให้ขั้นตอนวิธีมีความซับซ้อนยิ่งขึ้น วิธีที่ใช้ในการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพ โดยสามารถคำนวณเลขคณิตจุดตรึงได้ คือ การไหลเชิงแสง (optical flow) แต่วิธีการนี้ต้องคำนวณหาผลต่างเชิงอนุพันธ์ทางเวลา ทางตั้ง และทางระนาบ ผลต่างเชิงอนุพันธ์ทางเวลาได้จากผลต่างระหว่างภาพ กับผลต่างเชิงอนุพันธ์ทางตั้ง กับทางระนาบได้จากการกรอง (filter) การคำนวณหาผลต่างเชิงอนุพันธ์และการคำนวณการไหลเชิงแสงเป็นขั้นตอนที่ใช้เวลาประมวลผลนานซึ่งไม่เหมาะกับระบบฝังตัว การตรวจหาข้อมูลของวัตถุเคลื่อนที่โดยใช้วิธีหาความต่างระหว่างเฟรม (frame difference) ซึ่งเป็นวิธีการที่เร็วกว่าการลบภาพพื้นหลัง และการไหลเชิงแสง นอกจากความเร็วของวิธีการนี้แล้ว ยังมีความน่าสนใจอยู่ที่ความคงทนต่อการเปลี่ยนแปลงแสงที่ละน้อย และความคงทนต่ออุบัติเหตุที่ทำให้กล้องวิดีโอทัศนชัดขยับเล็กน้อย แต่ผลลัพธ์ที่ได้จากวิธีการนี้จะป็นข้อมูลที่ป็นขอบที่เกิดจากผลต่างระหว่างสองเฟรม ปัญหาการกำหนดขนาดของค่าขีดเริ่มเปลี่ยนที่เหมาะสม และปัญหาวัตถุเคลื่อนที่หายไปถ้าวัตถุในลำดับภาพหยุดเคลื่อนที่

วิทยานิพนธ์นี้เสนอเทคนิคการแก้ปัญหาที่เกิดจากวิธีความต่างระหว่างเฟรม โดยเฉพาะปัญหาขอบของวัตถุเคลื่อนที่ไม่ต่อเนื่องกัน การปรับขนาดของค่าขีดเริ่มเปลี่ยนอัตโนมัติ และการจดจำวัตถุเคลื่อนที่เมื่อวัตถุหยุดนิ่ง การประมวลผลทั้งหมดเป็นการคำนวณแบบเลขคณิตจุดตรึง เป็นวิธีการเพื่อใช้ในระบบฝังตัวได้

1.4 วัตถุประสงค์

1. เพื่อพัฒนากล้องระบบฝังตัวที่มีความสามารถตรวจจับวัตถุเคลื่อนที่ในลำดับภาพวีดิทัศน์ได้
2. เพื่อพัฒนากล้องระบบฝังตัวที่ไม่ต้องเก็บภาพพื้นหลังเพื่อเริ่มตรวจจับวัตถุเคลื่อนที่ในลำดับภาพวีดิทัศน์
3. เพื่อพัฒนากล้องระบบฝังตัวให้ส่งภาพเฉพาะวัตถุเคลื่อนที่เท่านั้น ไปยังคอมพิวเตอร์แม่ข่าย
4. เพื่อลดจำนวนข้อมูลภาพที่ต้องส่งจากชุดถ่ายภาพไปยังคอมพิวเตอร์แม่ข่าย
5. เพื่อลดงานคำนวณที่คอมพิวเตอร์แม่ข่ายในส่วนของการทำงานวัตถุเคลื่อนที่

1.5 ขอบเขตของวิทยานิพนธ์

1. สามารถตรวจจับวัตถุเคลื่อนที่ในลำดับภาพจากกล้องวีดิทัศน์ได้
2. สามารถรายงานภาพวัตถุเคลื่อนที่จากอุปกรณ์ระบบฝังตัวไปยังคอมพิวเตอร์แม่ข่ายได้
3. สามารถทำงานร่วมกับการบริหารจัดการการติดตามบนคอมพิวเตอร์แม่ข่ายได้
4. สามารถปรับพารามิเตอร์ที่อุปกรณ์ระบบฝังตัว เพื่อความสะดวกต่อการติดตั้งได้
5. สามารถรายงานภาพที่อุปกรณ์ระบบฝังตัว ผ่านคอนเนคเตอร์ VGA out เพื่อการติดตั้งได้

1.6 ขั้นตอนการดำเนินงาน

1. ศึกษาวิธีการตรวจจับการเคลื่อนไหวที่มีผู้วิจัยแล้ว เพื่อศึกษาข้อดีข้อเสียของแต่ละขั้นตอนวิธี และนำมา กำหนดคุณสมบัติของขั้นตอนวิธีที่จะพัฒนา
2. ทดสอบขั้นตอนวิธี ด้วยการเขียนโปรแกรมทดสอบด้วยภาษา C/C++ กับไลบรารี OpenCV ในคอมพิวเตอร์ เพื่อหาข้อจำกัดของแต่ละขั้นตอนวิธีที่ทดสอบ โดยทดลองกับไฟล์วีดิทัศน์
3. ศึกษาอุปกรณ์ระบบฝังตัว เพื่อนำขั้นตอนวิธีที่จำเป็นมาลงในอุปกรณ์ระบบฝังตัวให้สามารถใช้งานได้
4. ปรับปรุงแก้ไขข้อผิดพลาดที่เกิดขึ้น
5. สรุปและเขียนวิทยานิพนธ์

1.7 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับความรู้การเขียนโปรแกรมภาษา C/C++ ควบคู่กับไลบรารี OpenCV ทางการประมวลผลสัญญาณภาพแบบดิจิทัล
2. ได้รับความรู้การทำให้อุปกรณ์ระบบฝังตัวสามารถประมวลผลสัญญาณภาพแบบดิจิทัล
3. สามารถนำไปประยุกต์ใช้กับงานรักษาความปลอดภัย สำหรับส่วนของระบบเฝ้าระวังภัยอัตโนมัติ
4. สามารถพัฒนาต่อเพื่อสร้างเครือข่ายกล้องระวางภัย



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

หลักการที่ใช้ในวิทยานิพนธ์

เราตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอจากกล้องวิดีโอที่ถ่ายภาพ ณ บริเวณหนึ่งให้เป็นฉากคงที่ด้วยวิธีความต่างระหว่างเฟรม เพราะวิธีนี้เป็นวิธีการที่ทนต่ออุบัติเหตุให้กล้องขยับ ทนต่อการเคลื่อนที่เล็กน้อยอย่างไร้ไม่ไหว และทนต่อการเปลี่ยนแปลงสภาพแวดล้อมของบริเวณนั้น ทนต่อการเปลี่ยนแปลงแสงทั้งแบบที่ละน้อยและแบบทันทีทันใด วัตถุเคลื่อนที่ทำให้เกิดความต่างระหว่างสองเฟรม และความต่างนั้นต้องมีขนาดมากกว่าค่าขีดเริ่มเปลี่ยน ดังสมการที่ (2.1) โดยที่ เฟรม_{*i*} แทนเฟรมปัจจุบัน และเฟรม_{*i-1*} แทนเฟรมก่อนหน้า

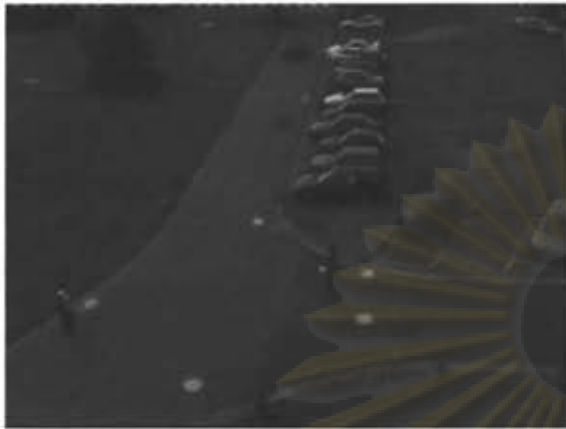
$$|\text{เฟรม}_i - \text{เฟรม}_{i-1}| > \text{ขีดเริ่มเปลี่ยน} \quad (2.1)$$

ค่าขีดเริ่มเปลี่ยนในสมการที่ (2.1) มีผลต่อการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพมาก การกำหนดค่าขีดเริ่มเปลี่ยนให้มีขนาดมาก หรือน้อยเกินไป มีผลต่อประสิทธิภาพของการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอ รูปที่ 2.1(ค) แสดงการกำหนดค่าขีดเริ่มเปลี่ยนที่มากเกินไป ทำให้ได้ข้อมูลไม่ครบถ้วน รูปที่ 2.1(ง) แสดงการกำหนดค่าขีดเริ่มเปลี่ยนที่น้อยเกินไป ทำให้มีตัวรบกวนเกิดขึ้น และ รูปที่ 2.1(ข) แสดงการกำหนดค่าขีดเริ่มเปลี่ยนที่เหมาะสม การกำหนดค่าขีดเริ่มเปลี่ยนคงที่ ไม่เหมาะสำหรับการใช้งานจริง เพราะระดับความเข้มแสงในสถานที่หนึ่ง ๆ มีการเปลี่ยนแปลงที่ละน้อยตลอดเวลา และความต่างระหว่างเฟรมมีผลลัพธ์เป็นขอบไม่ต่อเนื่องกันบนวัตถุเคลื่อนที่เดียวกัน เช่น ในรูปที่ 2.2(ข) เป็นภาพขอบที่ได้จากความต่างระหว่างเฟรมผกผัน ที่ได้จากผลต่างระหว่างเฟรมในรูปที่ 2.2(ก) กับ ภาพก่อนหน้าหนึ่งเฟรม

วิทยานิพนธ์นี้เสนอเทคนิคการแก้ปัญหาลักษณะ และค่าขีดเริ่มเปลี่ยนของการหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอด้วยวิธีความต่างระหว่างเฟรม โดยการแบ่งภาพในเฟรมหนึ่งออกเป็นแผงภาพย่อย ดังรูปที่ 2.3 และ ระบุแผงภาพย่อยนั้น เป็นแผงภาพเคลื่อนที่ หรือเป็นแผงภาพหยุดนิ่ง จากค่าขีดเริ่มเปลี่ยน โดยคำนวณค่าขีดเริ่มเปลี่ยนทุกครั้ง จากฮิสโทแกรมของการรวมผลต่างระหว่างเฟรมกำลังสอง และแก้ปัญหาวัดเคลื่อนที่หาย เมื่อวัตถุเคลื่อนที่ในลำดับภาพนั้นหยุดนิ่ง ด้วยการจำตำแหน่งและบริเวณของวัตถุนั้น

2.1 การระบุคุณลักษณะแผงภาพ

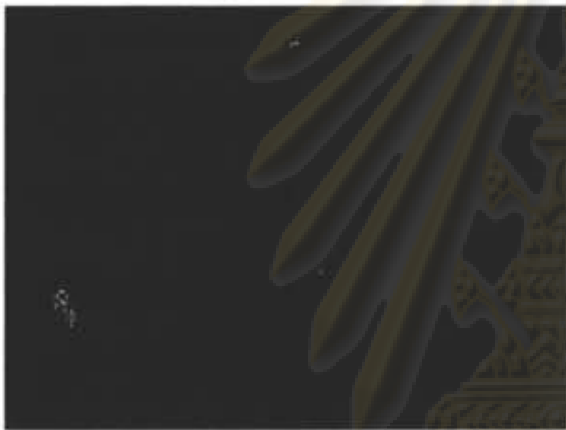
คุณลักษณะของแผงภาพในวิทยานิพนธ์นี้มีสองชนิด คือ แผงภาพเคลื่อนที่ กับแผงภาพหยุดนิ่ง เราระบุคุณลักษณะของแผงภาพ ด้วยวิธีความต่างระหว่างเฟรม โดยวิธีนี้เป็นการหาความเข้มแสงที่ต่างกัน ระหว่างสองเฟรม ในลำดับภาพวิดีโอที่ต่อเนื่องกัน เมื่อ I_t และ I_{t-1} แทนความเข้มแสง ณ เวลา t และ $t-1$



(ก) ภาพเริ่มต้น



(ข) ค่าขีดเริ่มเปลี่ยนที่เหมาะสม



(ค) ค่าขีดเริ่มเปลี่ยนมากเกินไป



(ง) ค่าขีดเริ่มเปลี่ยนน้อยเกินไป

รูปที่ 2.1 ปัญหาค่าขีดเริ่มเปลี่ยนของความต่างระหว่างเฟรม

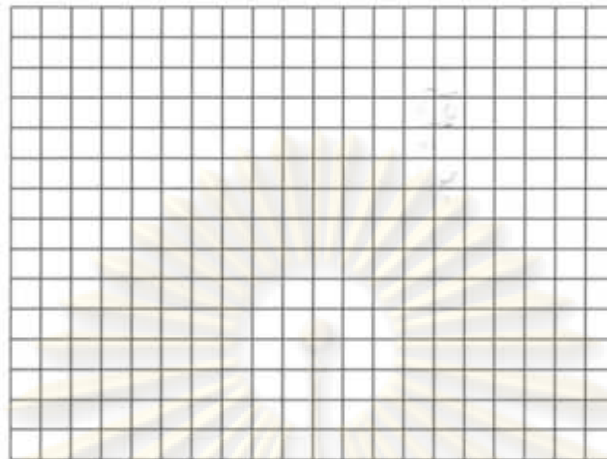


(ก) ภาพเริ่มต้น



(ข) ภาพขอบที่เกิดจากความต่างระหว่างเฟรมยกพื้น

รูปที่ 2.2 ปัญหาขอบของความต่างระหว่างเฟรม



รูปที่ 2.3 ตัวอย่างการแบ่งภาพในเฟรมหนึ่งออกเป็นแผงภาพย่อย

ตามลำดับ ผลต่างของความเข้มแสงในลำดับภาพวิดิทัศน์ ณ เวลา t กับ เวลา $t - 1$ เป็นผลต่างสืบเนื่องอันดับหนึ่ง (first difference) เรานิยามด้วยตัวดำเนินการ Δ ดังสมการ (2.2)

$$\Delta I_t = I_t - I_{t-1} \quad (2.2)$$

เราคำนวณหาคุณสมบัติของแผงภาพ p ที่มีขนาด $N \times N$ จุดภาพ จากการรวมผลต่างของความเข้มแสงระหว่างเฟรมกำลังสอง ภายในแผงภาพ p เป็นสมการที่ (2.3)

$$D_{t,p} = \sum_{i=1}^{N^2} ((\Delta I_{t,i})^2) \quad (2.3)$$

เรากำหนดคุณสมบัติของแผงภาพ p ณ เวลา t ได้จาก $D_{t,p}$ เปรียบเทียบกับขนาดของค่าขีดเริ่มเปลี่ยนวิธานิพนธ์นี้กำหนดให้ δ เป็นค่าขีดเริ่มเปลี่ยน สำหรับกำหนดคุณสมบัติของแผงภาพ โดยแผงภาพ p ณ เวลา t เป็นแผงภาพเคลื่อนที่ถ้า $D_{t,p}$ มีขนาดมากกว่า δ หรือ เป็นแผงภาพหยุดนิ่งถ้า $D_{t,p}$ มีขนาดน้อยกว่า δ สรุปเป็นสมการที่ (2.4)

$$D_{t,p} - \delta \begin{cases} > 0 & \text{แผงภาพเคลื่อนที่} \\ \leq 0 & \text{แผงภาพหยุดนิ่ง} \end{cases} \quad (2.4)$$

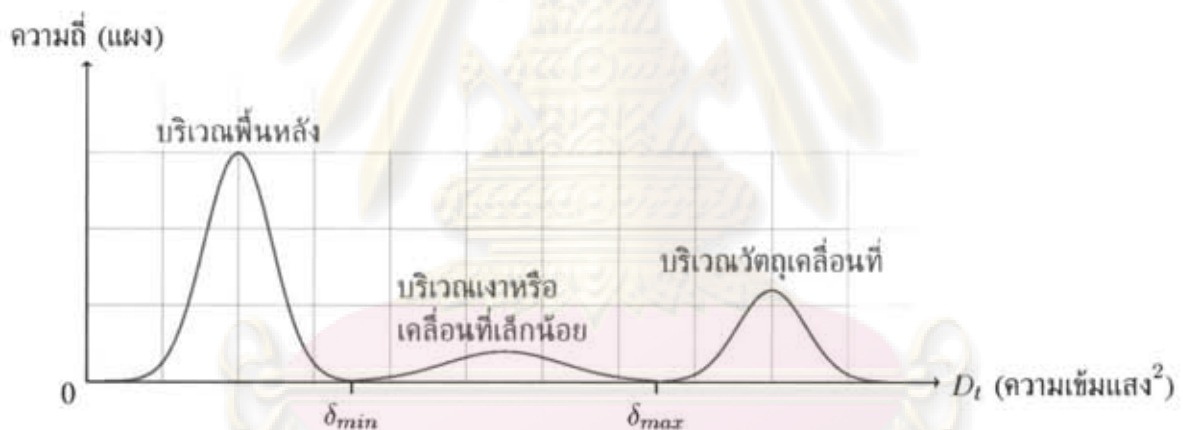
2.1.1 การคำนวณค่าขีดเริ่มเปลี่ยน

เนื่องจากผลรวมความต่างระหว่างเฟรมกำลังสองมีขนาดใหญ่ เมื่อนำมาทำฮิสโทแกรม เราจึงใช้สเกลลอการิทึมฐานสอง (base-two logarithm) สำหรับแบ่งช่วงความถี่ของฮิสโทแกรม เราเลือกค่าขีดเริ่มเปลี่ยนจากฮิสโทแกรมดังกล่าว โดยกำหนดค่าขีดเริ่มเปลี่ยนสองค่า คือ δ_{min} กับ δ_{max} ค่าขีดเริ่มเปลี่ยนเป็นค่าที่มีความถี่น้อยที่สุดในบริเวณหนึ่ง ซึ่งบริเวณนั้นมีค่ายอดของความถี่ในฮิสโทแกรมล้อมรอบ ดังรูปที่ 2.4

กับ รูปที่ 2.5 เห็นได้ชัดว่าค่าขีดเริ่มเปลี่ยนจะมีหรือไม่ขึ้นกับลักษณะของฮิสโทแกรม และลักษณะของฮิสโทแกรมสามารถบอกได้ว่ามีวัตถุเคลื่อนที่หรือไม่ ในที่นี้เราแบ่งแบบจำลองของฮิสโทแกรมเป็น 2 ชนิด คือ แบบจำลองของฮิสโทแกรมที่มีวัตถุเคลื่อนที่ กับแบบจำลองของฮิสโทแกรมที่ไม่มีวัตถุเคลื่อนที่ โดยแบบจำลองของฮิสโทแกรมที่มีวัตถุเคลื่อนที่ เป็นฮิสโทแกรมที่มีฐานนิยมสอดคล้องกับสมการที่ (2.5)

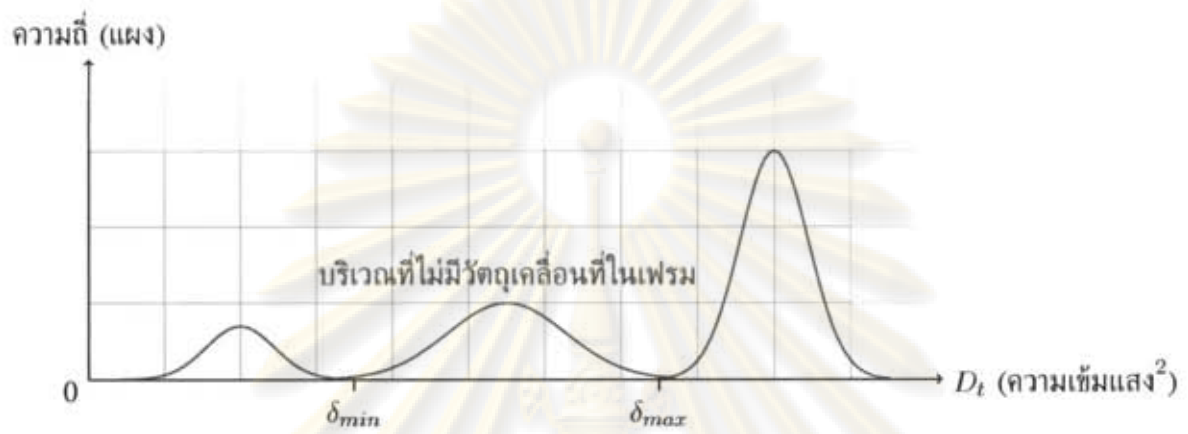
$$mode_{D_t < \delta_{min}} \geq mode_{D_t > \delta_{max}} \quad (2.5)$$

กำหนดให้ $mode_{D_t < \delta_{min}}$ เป็นฐานนิยมที่มี $D_t < \delta_{min}$ และ กำหนดให้ $mode_{D_t > \delta_{max}}$ เป็นฐานนิยมที่มี $D_t > \delta_{max}$ แบบจำลองฮิสโทแกรมในรูปที่ 2.4 เป็นฮิสโทแกรมที่สอดคล้องกับสมการที่ (2.5) และมีฐานนิยมสามส่วนคือ $mode_{D_t < \delta_{min}}$, $mode_{\delta_{min} < D_t < \delta_{max}}$ และ $mode_{D_t > \delta_{max}}$ โดย $mode_{D_t < \delta_{min}}$ เป็นฐานนิยมของแผนภาพหยุดนิ่ง ส่วน $mode_{D_t > \delta_{max}}$ เป็นฐานนิยมของแผนภาพเคลื่อนที่ และ $mode_{\delta_{min} < D_t < \delta_{max}}$ เป็นฐานนิยมของแผนภาพที่มีการเคลื่อนที่เล็กน้อย หรือเป็นบริเวณเงาของภาพ ส่วนฐานนิยมที่ไม่สอดคล้องกับสมการ (2.5) เป็นแบบจำลองของฮิสโทแกรมที่ไม่มีวัตถุเคลื่อนที่ ดังฮิสโทแกรมในรูปที่ 2.5 กับ ฮิสโทแกรมที่ไม่สามารถคำนวณหาค่าขีดเริ่มเปลี่ยนได้ ดังแบบจำลองในรูปที่ 2.6



รูปที่ 2.4 ฮิสโทแกรมการรวมผลต่างกำลังสองของเฟรมที่มีวัตถุเคลื่อนที่

แผนภาพเคลื่อนที่ ซึ่งระบุจากวิธีความต่างระหว่างเฟรม ไม่สอดคล้องกับภาพในเฟรมใดเฟรมหนึ่งเลย เราพิจารณาภาพในรูปที่ 2.7 รูปที่ 2.8 และรูปที่ 2.9 โดยรูปที่ 2.7(ก) ถึงรูปที่ 2.7(ค) เป็นลำดับภาพที่เวลา $(t - 1)$ (t) และ $(t + 1)$ ตามลำดับ เราใช้วิธีการตรวจหาวัตถุเคลื่อนที่ ด้วยวิธีความต่างระหว่างเฟรมกำลังสอง ได้ผลลัพธ์เป็นภาพในรูปที่ 2.8 โดยผลต่างระหว่างเฟรมกำลังสอง ในรูปที่ 2.8(ก) เป็นผลลัพธ์ระหว่างภาพในรูปที่ 2.7(ก) กับภาพในรูปที่ 2.7(ข) และผลต่างระหว่างเฟรมกำลังสองในรูปที่ 2.8(ข) เป็นผลลัพธ์ระหว่างภาพในรูปที่ 2.7(ข) กับภาพในรูปที่ 2.7(ค) ผลลัพธ์ในรูปที่ 2.8 ไม่สอดคล้องกับเฟรมใดเลยในรูปที่ 2.7 เราแก้ปัญหาด้วยการอินเตอร์เซกชันระหว่างภาพในรูปที่ 2.8(ก) กับภาพในรูปที่ 2.8(ข) จะได้ผลลัพธ์เป็นภาพในรูปที่ 2.9 ที่สอดคล้องกับภาพในรูปที่ 2.7(ข)



รูปที่ 2.5 ฮิสโทแกรมการรวมผลต่างกำลังสองของเฟรมที่ไม่มีวัตถุเคลื่อนที่แบบที่ 1



รูปที่ 2.6 ฮิสโทแกรมการรวมผลต่างกำลังสองของเฟรมที่ไม่มีวัตถุเคลื่อนที่แบบที่ 2

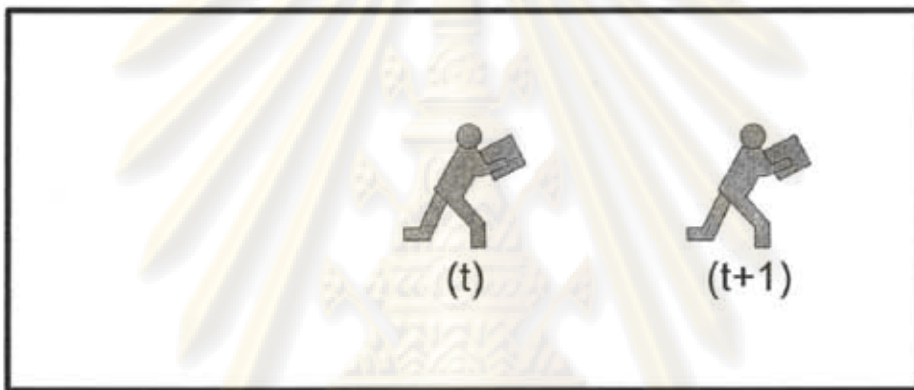
(ก) ภาพ ณ เวลา $t - 1$ (ข) ภาพ ณ เวลา t (ค) ภาพ ณ เวลา $t + 1$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 2.7 ตั้งอย่างวัตถุเคลื่อนที่ ณ เวลาต่างกัน



(ก) ผลต่างระหว่างเฟรมกำลังสอง ของภาพในรูปที่ 2.7(ก) กับภาพในรูปที่ 2.7(ข)



(ข) ผลต่างระหว่างเฟรมกำลังสอง ของภาพในรูปที่ 2.7(ข) กับภาพในรูปที่ 2.7(ค)

รูปที่ 2.8 ตัวอย่างผลต่างระหว่างเฟรมกำลังสอง



รูปที่ 2.9 ผลการอินเตอร์เซกชันระหว่างภาพในรูปที่ 2.8(ก) กับภาพในรูปที่ 2.8(ข)

เราใช้แนวทางดังกล่าวแก้ปัญหาในรูปที่ 2.10 เป็นปัญหาที่แผนภาพไม่สอดคล้องกับวัตถุเคลื่อนที่ ซึ่งเกิดจากความต่างระหว่างเฟรมกำลังสอง เราหาแผนภาพที่สอดคล้องกับวัตถุเคลื่อนที่ในภาพ ณ เวลา t โดยใช้ข้อมูล $D_{t,p}$ δ_t $D_{t+1,p}$ กับ δ_{t+1} กับขั้นตอนวิธี 2.1.1 เพื่อกำหนดแผนภาพเคลื่อนที่สำหรับแผนภาพ p ณ เวลา t หรือ p_t เราใช้ข้อมูลกับวิธีการดังกล่าว ได้ผลลัพธ์ใหม่เป็นรูปที่ 2.11 และผลลัพธ์ที่ได้มีความสอดคล้องกับภาพในเฟรมมากขึ้น เมื่อเปรียบเทียบระหว่างรูปที่ 2.11 กับรูปที่ 2.10



(ก) แผนภาพที่เกิดจากเฟรมที่ 17 กับ 18 กับภาพในเฟรมที่ 17 (ข) แผนภาพที่เกิดจากเฟรมที่ 17 กับ 18 กับภาพในเฟรมที่ 18

รูปที่ 2.10 ปัญหาความไม่สอดคล้องกันระหว่างแผนภาพกับภาพ

ขั้นตอนวิธี 2.1.1

- 1: **if** $(D_{t,p} > \delta_{t,max}) \wedge (D_{t+1,p} > \delta_{t+1,max})$ **then**
- 2: $p_t \leftarrow$ แผนภาพเคลื่อนที่
- 3: **else**
- 4: $p_t \leftarrow$ แผนภาพหยุดนิ่ง
- 5: **end if**

ขั้นตอนวิธี 2.1.2

- 1: **if** $\delta_{t+1,min} < D_{t+1,p} < \delta_{t+1,max}$ **then**
- 2: **if** $p_{t-1} =$ แผนภาพเคลื่อนที่ **then**
- 3: $p_t \leftarrow$ แผนภาพเคลื่อนที่
- 4: **else**
- 5: $p_t \leftarrow$ แผนภาพหยุดนิ่ง
- 6: **end if**
- 7: **end if**

การขาดหายของข้อมูลของแผนภาพที่มี D_t อยู่ในช่วงระหว่าง δ_{min} กับ δ_{max} ตัวอย่างในรูปที่ 2.12 โดยรูปที่ 2.12(ข) ไม่มีแผนภาพเคลื่อนที่ปรากฏที่บุคคลทางซ้ายมือ เพราะแผนภาพเคลื่อนที่ของผู้ชายด้านซ้ายมือ มี D_t อยู่ในช่วงระหว่าง δ_{min} กับ δ_{max} ดังรูปที่ 2.12(ค) เราเสนอขั้นตอนวิธี 2.1.2 เพื่อใช้ข้อมูล



รูปที่ 2.11 ภาพที่ได้จากขั้นตอนวิธี (2.1.1) ร่วมกับข้อมูลจากเฟรมที่ 17, 18 และ 19

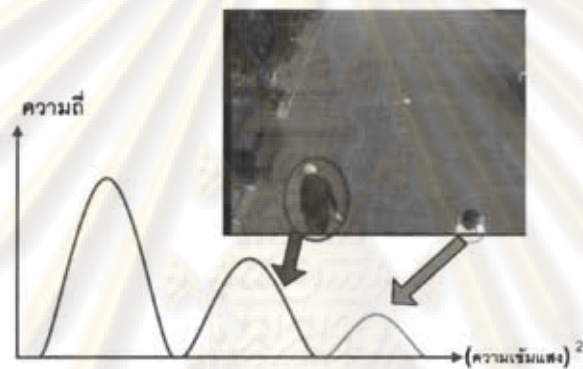
ที่ D_t ที่มีค่าในช่วงระหว่าง δ_{min} กับ δ_{max} รูปที่ 2.13(ข) เป็นผลลัพธ์ที่ได้จากการแก้ปัญหาดังกล่าว ด้วยขั้นตอนวิธี 2.1.2 โดย กากบาท (+) ในรูปแทนแผงภาพเคลื่อนที่

ขั้นตอนวิธี 2.1.3

Ensure: $N =$ เซตจำนวนนับ

- 1: คำนวณหา $D_{t,p}$ ของแต่ละแผงภาพในหนึ่งเฟรม
- 2: สร้างฮิสโทแกรมจาก D_t จากทุกแผงภาพในหนึ่งเฟรม
- 3: คำนวณหา $\delta_{t,min}$ กับ $\delta_{t,max}$
- 4: **if** $\delta_{t,min} \vee \delta_{t,max} \subset N$ **then**
- 5: **if** $(D_{t+1,p} > \delta_{t+1,max}) \wedge (D_{t,p} > \delta_{t,max})$ **then**
- 6: $p_t \leftarrow$ แผงภาพเคลื่อนที่
- 7: **else**
- 8: **if** $\delta_{t+1,min} < D_{t+1,p} < \delta_{t+1,max}$ **then**
- 9: **if** $p_{t-1} =$ แผงภาพเคลื่อนที่ **then**
- 10: $p_t \leftarrow$ แผงภาพเคลื่อนที่
- 11: **else**
- 12: $p_t \leftarrow$ แผงภาพหยุดนิ่ง
- 13: **end if**
- 14: **end if**
- 15: **end if**
- 16: **else**
- 17: ทุกแผงภาพ \leftarrow แผงภาพหยุดนิ่ง
- 18: **end if**

การระบุแผงภาพในวิทยานิพนธ์นี้ เป็นขั้นตอนวิธีที่ไม่ต้องการเลขคณิตจุดลอยตัว เหมาะสมกับระบบฝังตัว การทำงานแบบเวลาจริง และปรับค่าค่าขีดเริ่มเปลี่ยนอัตโนมัติจากฮิสโทแกรม โดยแบบจำลองของฮิสโทแกรม แบ่งเป็น 2 แบบจำลอง คือ แบบจำลองฮิสโทแกรมที่มีวัตถุเคลื่อนที่ กับ แบบจำลองฮิสโทแกรมที่ไม่มีวัตถุเคลื่อนที่ การระบุแผงภาพในขั้นตอนวิธี 2.1.1 กับ ขั้นตอนวิธี 2.1.2 สามารถสรุปเป็นขั้นตอนวิธี 2.1.3

(ก) แผงภาพที่มีขนาด $D_{t,p}$ มากกว่า δ_{max} ในเฟรมที่ 170(ข) แผงภาพที่มีขนาด $D_{t,p}$ มากกว่า δ_{max} ในเฟรมที่ 171

(ค) ความสัมพันธ์ของฮิสโทแกรมกับแผงภาพในเฟรมที่ 171

รูปที่ 2.12 ปัญหาข้อมูลขาดหายเมื่อ $D_{t,p} > \delta_{min}$ แต่ $D_{t,p} < \delta_{max}$

(ก) แผงภาพที่มีขนาด $D_{t,p}$ มากกว่า δ_{max} ในเฟรมที่ 170

(ข) แผงภาพที่แก้ปัญหาด้วยขั้นตอนวิธี (2.1.2) ในเฟรมที่ 171

รูปที่ 2.13 การดึงข้อมูลที่ $D_{t,p} > \delta_{min}$ แต่ $D_{t,p} < \delta_{max}$

2.2 การจดจำกลุ่มแผนภาพเคลื่อนที่

การรวมกลุ่มแผนภาพเคลื่อนที่ของวัตถุเดียวกัน เพื่อความสะดวกต่อการจัดการ เรารวมกลุ่มของแผนภาพเคลื่อนที่ เป็นกรอบสี่เหลี่ยมที่ครอบคลุมทุกแผนภาพเคลื่อนที่ของวัตถุเดียวกัน วิทยานิพนธ์นี้ใช้ประโยชน์ของการรวมกลุ่มแผนภาพเคลื่อนที่ เพื่อการจดจำข้อมูลวัตถุเคลื่อนที่ กรอบสี่เหลี่ยมที่ครอบคลุมทุกแผนภาพเคลื่อนที่บนวัตถุเดียวกัน คำนวณจากคอนทัวร์ (contour) ของแผนภาพเคลื่อนที่ การหาคอนทัวร์จากแผนภาพดีกว่าการหาคอนทัวร์ในระดับจุดภาพ เพราะจำนวนแผนภาพมีปริมาณน้อยกว่าจำนวนจุดภาพ ที่ความละเอียดของภาพในเฟรมหนึ่งมีความละเอียดเป็น 320×240 จุดภาพ จำนวนจุดภาพที่ต้องนำมาคำนวณหาคอนทัวร์คือ 76,800 จุดภาพ ถ้าใช้แผนภาพขนาด 8×8 จุดภาพ ในการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพ วิดิทัศน์ จะทำให้จำนวนแผนภาพที่ต้องนำมาคำนวณหาคอนทัวร์คือ 1,200 แผนภาพ การคำนวณหาคอนทัวร์ที่มีปริมาณข้อมูลที่น้อยกว่าต้องคำนวณเสร็จสิ้นเร็วกว่า การคำนวณหาคอนทัวร์ของแผนภาพขนาด $p \times p$ จะเร็วกว่าการคำนวณคอนทัวร์ทุกจุดภาพถึง $p \times p$ เท่า

กำหนดให้ L เป็นเซตของกรอบสี่เหลี่ยมที่มีตำแหน่งและขนาดที่ได้จากคอนทัวร์ของการรวมกลุ่มแผนภาพเคลื่อนที่ที่ติดกัน ณ เวลา t ดังสมการที่ (2.6) และ M เป็นเซตของความจำกรอบสี่เหลี่ยมที่มีตำแหน่ง และขนาดของกลุ่มแผนภาพเคลื่อนที่ที่ติดกัน หรือเป็นเซตของกรอบสี่เหลี่ยมที่มีขนาดและตำแหน่งครอบคลุมวัตถุที่ปรากฏในลำดับภาพวิดิทัศน์ ดังสมการที่ (2.7)

$$L = \{l_i : \text{กรอบสี่เหลี่ยมที่ครอบคลุมแผนภาพเคลื่อนที่ที่ติดกันกลุ่มที่ } i \text{ ณ เวลา } t\} \quad (2.6)$$

$$M = \{m_j : \text{กรอบสี่เหลี่ยมที่ครอบคลุมวัตถุ ที่ปรากฏลำดับที่ } j \text{ ณ เวลา } t-1\} \quad (2.7)$$

สมาชิกของเซต M ได้จากสมาชิกของเซต L ที่ไม่มีอินเตอร์เซกชันกับสมาชิกตัวใดเลยในเซต M เราใช้เซต Q แทนเซตสมาชิกใหม่ของเซต M และอธิบายเป็นสมการที่ (2.8)

$$Q = \{l_i | (\exists l_j \in L) \cap (\forall m_k \in M) = \phi\} \quad (2.8)$$

การปรับขนาดกับตำแหน่งของสมาชิกในเซต M ปรับจากสมาชิกบางตัวของเซต L ที่มีอินเตอร์เซกชันบางส่วนกับสมาชิกหนึ่ง ๆ ในเซต M การหยุดของวัตถุเคลื่อนที่ตอนเริ่มต้นจะมีการเปลี่ยนแปลงทางขนาดของกรอบสี่เหลี่ยมที่ครอบคลุมวัตถุเคลื่อนที่นั้น ซึ่งเป็นการลดขนาดพื้นที่ลงอย่างรวดเร็ว ดังนั้นการปรับจึงมีเงื่อนไข ในวิทยานิพนธ์นี้กำหนดระดับเล็กที่สุดที่จะไม่ปรับขนาดพื้นที่กับตำแหน่ง เมื่อมีการเปลี่ยนแปลงทางขนาดของพื้นที่จากใหญ่ไปเป็นขนาดพื้นที่เล็กลง โดยจะไม่ปรับขนาดพื้นที่กับตำแหน่งเมื่อขนาดของพื้นที่ใหม่เล็กกว่าร้อยละ 25 ของขนาดพื้นที่เดิม สรุปเป็นขั้นตอนที่ 2.2.4

ขั้นตอนวิธี 2.2.4

- 1: if $(L \ni l_i \cap m_j \in M) \neq \phi$ then
 - 2: if ขนาดพื้นที่ของ $l_i > \frac{1}{4}$ ขนาดพื้นที่ของ m_j then
 - 3: $m_j = l_i$
 - 4: end if
 - 5: end if
-

การกำจัดสมาชิกในเซต M โดย ถ้าขนาดของสมาชิกด้านกว้าง หรือด้านยาว มีขนาดเท่ากับด้านกว้าง หรือด้านยาว ของแผนภาพหนึ่งแผนดังสมการที่ (2.9)

$$M \neq \phi \forall m (m \in M \wedge m : \text{กรอบสี่เหลี่ยมที่มีด้านกว้าง กับด้านสูง ยาวกว่า 1 แผนภาพ}) \quad (2.9)$$

ภายในส่วนนี้แสดงการจดจำข้อมูลภายในเซต M เพื่อลดข้อบกพร่องของการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดิทัศน์ ด้วยวิธีการหาความต่างระหว่างเฟรม จุดสำคัญคือการสร้างสมาชิกใหม่ การปรับขนาดพื้นที่ที่ทับตำแหน่ง และการลบสมาชิกของเซต M โดยข้อมูลของสมาชิกในเซต M ณ เวลา t มีความสอดคล้องกับภาพในเฟรม ณ เวลา t ด้วยกัน

2.3 การรวมผลต่างระหว่างเฟรมกำลังสอง

ภาพเริ่มต้นในรูปที่ 2.14(ก) ถูกคำนวณหาความต่างระหว่างเฟรมกับภาพก่อนหน้า ได้ผลลัพธ์เป็นขอบในรูปที่ 2.14(ข) หลังจากนั้น แบ่งภาพผลลัพธ์ดังกล่าวเป็นแผนภาพย่อย ดังรูปที่ 2.14(ค) การคำนวณ $D_{t,p}$ ต้องคำนวณร่วมกับแผนภาพรอบข้าง เพื่อลดผลจากตัวรบกวนในภาพ เช่น รูปที่ 2.14(ง) แสดงแผนภาพที่ต้องนำมาคำนวณ $D_{t,p=x}$ ในแผนภาพ x ดังนั้นต้องการคำนวณ $D_{t,p=x}$ ต้องใช้แผนภาพ $a-h$ ซึ่งเป็นแผนภาพที่อยู่รอบ ๆ แผนภาพ x

เราใช้สมการที่ (2.10) คำนวณผลรวมความต่างระหว่างเฟรมกำลังสองในแผนภาพ $p(i, j)$ โดยคำนวณรวมกับข้อมูลจากแผนภาพรอบข้าง โดยแผนภาพขนาด $w \times w$ จุดภาพ บนภาพขนาด $width \times height$ จุดภาพ คู่อันดับ (i, j) เป็นตำแหน่งมุมซ้ายบนของแผนภาพที่เราสนใจ คู่อันดับ (u, v) เป็นตำแหน่งของจุดภาพ

$$D_{t,p(i,j)} = \sum_{\substack{v=j+w, \\ u=i+w, \\ v=j-w, \\ u=i-w}}^{v=j+w, \\ u=i+w} (\Delta I_t(u, v)) \quad (2.10)$$

ขั้นตอนวิธี 2.3.5 เป็นการคำนวณตามสมการที่ (2.3.5) รูปที่ 2.15 เป็นการคำนวณบางส่วนโดยใช้ขั้นตอนวิธี 2.3.5 บริเวณสี่เหลี่ยมในรูปที่ 2.15 คือบริเวณที่คำนวณระดับจุดภาพตามขั้นตอนวิธี 2.3.5 ในรูปที่ 2.15 ตั้งแต่ภาพ (ก) ถึงภาพ (ง) แสดงการวนซ้ำในบริเวณเดิมของขั้นตอนวิธีนี้



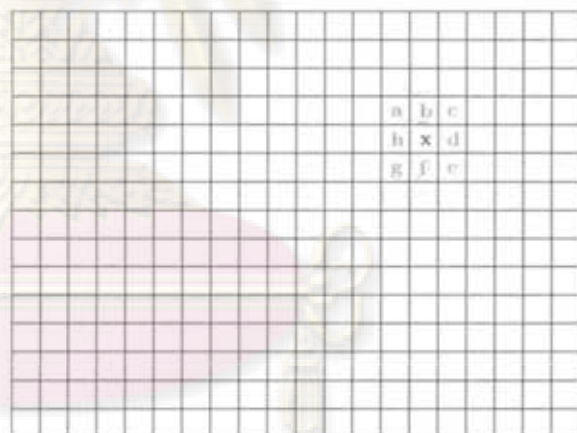
(ก) ภาพเริ่มต้น



(ข) ภาพผกผันของผลลัพธ์ความต่างระหว่างเฟรม



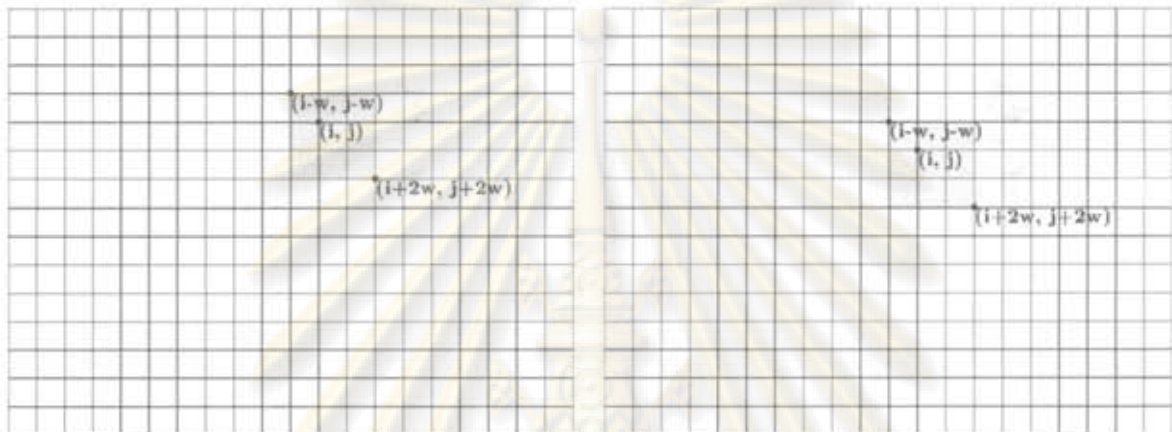
(ค) แบ่งภาพในหนึ่งเฟรมออกเป็นแผนภาพย่อย

(ง) แผนภาพ x ที่นำมาคำนวณ $D_{t,p-x}$ ต้องใช้การรวมผลต่างกำลังสองจากแผนภาพรอบข้าง $(-1, -1)$ มารวมคำนวณด้วย

ศูนย์วิทยทรัพยากร

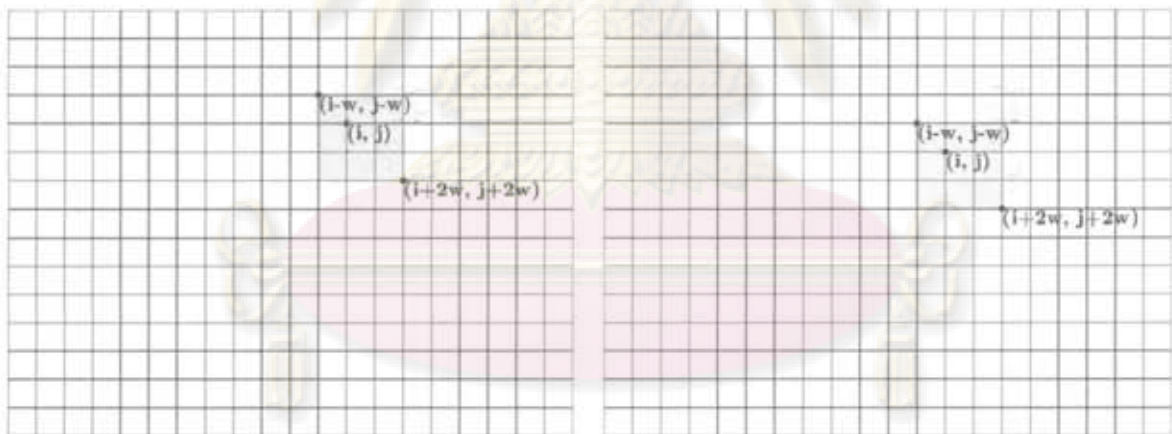
รูปที่ 2.14 ตัวอย่างการแบ่งแผนภาพ เพื่อการคำนวณ $D_{t,p}$

จุฬาลงกรณ์มหาวิทยาลัย



(ก) การคำนวณ $D_{t,p}$ ที่ $p(i = 11 \cdot w, j = 4 \cdot w)$ โดยบริเวณสี่เหลี่ยมแทนบริเวณที่มีการคำนวณระดับจุดภาพ

(ข) การคำนวณ $D_{t,p}$ ที่ $p(i = 11 \cdot w, j = 5 \cdot w)$ โดยบริเวณสี่เหลี่ยมแทนบริเวณที่มีการคำนวณระดับจุดภาพ



(ค) การคำนวณ $D_{t,p}$ ที่ $p(i = 12 \cdot w, j = 4 \cdot w)$ โดยบริเวณสี่เหลี่ยมแทนบริเวณที่มีการคำนวณระดับจุดภาพ

(ง) การคำนวณ $D_{t,p}$ ที่ $p(i = 12 \cdot w, j = 5 \cdot w)$ โดยบริเวณสี่เหลี่ยมแทนบริเวณที่มีการคำนวณระดับจุดภาพ

รูปที่ 2.15 การคำนวณ $D_{t,p}$ โดยใช้ขั้นตอนวิธี 2.3.5

ศูนย์วิจัยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ขั้นตอนวิธี 2.3.5

```

1: for j in {0, w, 2w, 3w, ..., height - 2w, height - w} do
2:   for i in {0, w, 2w, 3w, ..., width - 2w, width - w} do
3:      $D_{t,p(i,j)} \leftarrow 0$ 
4:     for v in {j - w, j + 1, j + 2, ..., j + 2w} do
5:       for u in {i - w, i + 1, i + 2, ..., i + 2w} do
6:          $D_{t,p(i,j)} \leftarrow D_{t,p(i,j)} + (\Delta I_t(u, v))^2$ 
7:       end for
8:     end for
9:   end for
10: end for

```

ขั้นตอนวิธี 2.3.6

Require: *buf_address*|*buf_address* = ตำแหน่งเก็บข้อมูลของที่פקข้อมูล

Require: *column_address*|*column_address* = ตำแหน่งของแผนภาพในแนวระนาบ

Require: *flag_computation*|*flag_computation* = true \longleftrightarrow มีข้อมูลเพียงพอที่จะคำนวณ $D_{t,p}$

```

1: flag_computation  $\leftarrow$  false
2: for j in {0, w, 2w, 3w, ..., height - 2w, height - w} do
3:   for i in {0, w, 2w, 3w, ..., width - 2w, width - w} do
4:      $d_{t,p(i,j)} \leftarrow 0$ 
5:     for v in {j, j + 1, j + 2, ..., j + w} do
6:       for u in {i, i + 1, i + 2, ..., i + w} do
7:          $d_{t,p(i,j)} \leftarrow d_{t,p(i,j)} + (\Delta I_t(u, v))^2$ 
8:       end for
9:     end for
10:    array[buf_address]  $\leftarrow$   $d_{t,p(i,j)}$ 
11:    buf_address  $\leftarrow$  buf_address + 1
12:    if buf_address =  $(2 \times \frac{\text{width}}{w} + 3)$  then
13:      buf_address  $\leftarrow$  0
14:      flag_computation  $\leftarrow$  true
15:    end if
16:    if flag_computation = true then
17:      if (column_address  $\neq$  0)  $\vee$  (column_address  $\neq$  1) then
18:         $D_{t,p(i-w,j-w)} \leftarrow d_{p(i,j)} + d_{p(i-1,j)} + d_{p(i-2,j)} + d_{p(i,j-1)} + d_{p(i-1,j-1)} + d_{p(i-2,j-1)} +$ 
 $d_{p(i,j-2)} + d_{p(i-1,j-2)} + d_{p(i-2,j-2)}$ 
19:      end if
20:    end if
21:    column_address  $\leftarrow$  column_address + 1
22:  end for
23: end for

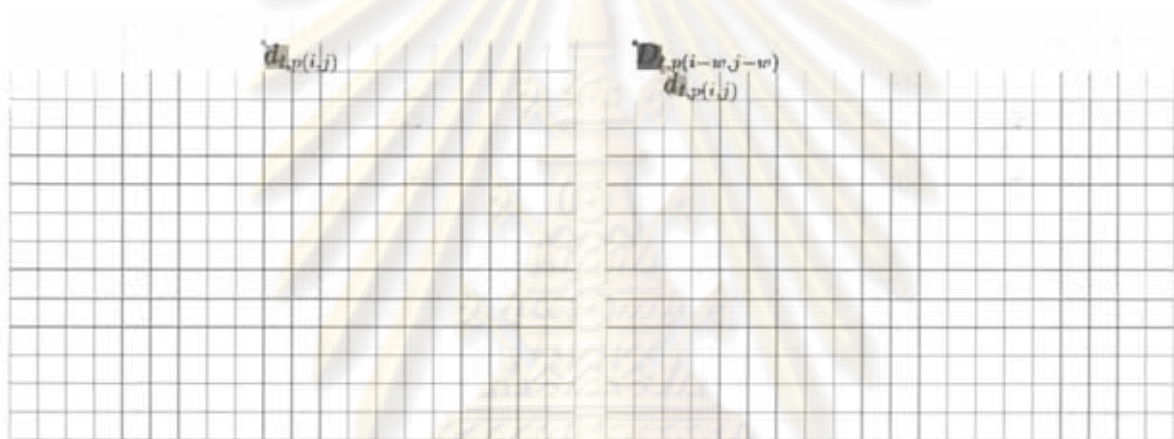
```

การคำนวณ $D_{t,p}$ ด้วยขั้นตอนวิธี 2.3.5 เกิดการคำนวณซ้ำตำแหน่งเดิมหลายรอบทำให้ใช้เวลาประมวลผลนาน เราออกแบบขั้นตอนวิธีใหม่เพื่อลดปัญหาดังกล่าว การรวมผลต่างระหว่างเฟรมในแผนภาพขนาด $w \times w$ โดยการเก็บผลของการรวมผลความต่างระหว่างเฟรมกำลังสองในแผนภาพขนาด $w \times w$ ในอาร์เรย์ (array)

เพื่อให้ในภายหลัง อาร์เรย์เป็นที่พักข้อมูลที่เก็บผลการรวมความต่างระหว่างเฟรมกำลังสองของแผนภาพย่อย d_p เป็นจำนวน $2 \times \frac{width}{w} + 3$ แผนภาพ และเขียนเป็นสมการที่ (2.11)

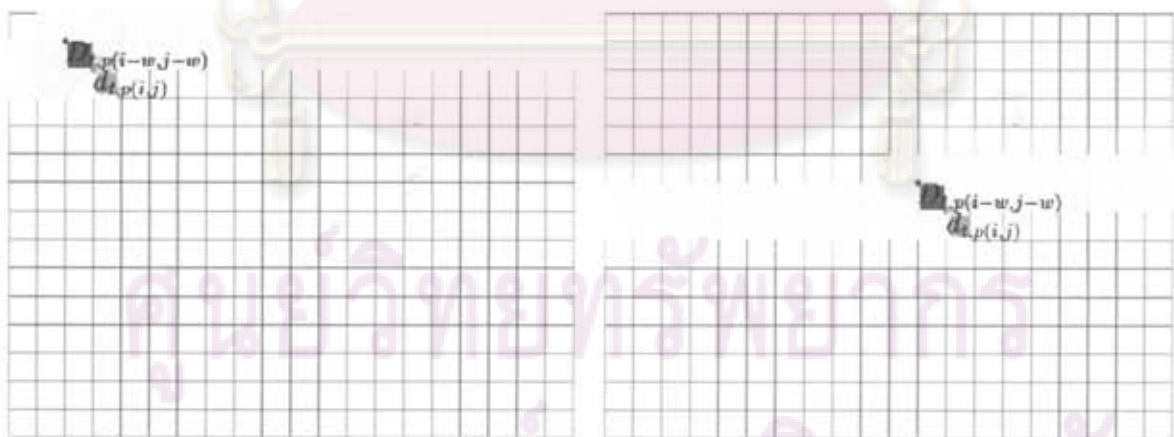
$$D_{t,p(i-1,j-1)} = d_{p(i,j)} + d_{p(i-1,j)} + d_{p(i-2,j)} + d_{p(i,j-1)} + d_{p(i-1,j-1)} + d_{p(i-2,j-1)} + d_{p(i,j-2)} + d_{p(i-1,j-2)} + d_{p(i-2,j-2)} \quad (2.11)$$

ขั้นตอนวิธี 2.3.6 ถูกสร้างจากสมการที่ (2.11) รูปที่ 2.16 แสดงบริเวณที่ต้องเก็บข้อมูลด้วยสี่เหลี่ยม และแสดงบริเวณที่มีการคำนวณระดับจุดภาพตามขั้นตอนวิธี 2.3.6 ด้วยสี่เหลี่ยม จากรูปที่ 2.16(ก) ถึงรูปที่ 2.16(ง) แสดงให้เห็นว่าขั้นตอนวิธีนี้จะไม่คำนวณระดับจุดภาพซ้ำที่จุดเดิมเลย



(ก) การเก็บข้อมูล $d_{t,p(i-w,j-w)}$ โดยบริเวณสี่เหลี่ยมแทนบริเวณที่มีการคำนวณระดับจุดภาพ บริเวณสี่เหลี่ยมแทนบริเวณแผนภาพที่ถูกเก็บในที่พักข้อมูล

(ข) การคำนวณ $D_{t,p(i-w,j-w)}$ เป็นการคำนวณครั้งแรก โดยบริเวณสี่เหลี่ยมแทนบริเวณที่มีการคำนวณระดับจุดภาพ บริเวณสี่เหลี่ยมแทนบริเวณแผนภาพที่ถูกเก็บในที่พักข้อมูล



(ค) การคำนวณ $D_{t,p(i-2w,j-w)}$ โดยบริเวณสี่เหลี่ยมแทนบริเวณที่มีการคำนวณระดับจุดภาพ บริเวณสี่เหลี่ยมแทนบริเวณแผนภาพที่ถูกเก็บในที่พักข้อมูล

(ง) การคำนวณ $D_{t,p(i-11w,j-6w)}$ โดยบริเวณสี่เหลี่ยมแทนบริเวณที่มีการคำนวณระดับจุดภาพ บริเวณสี่เหลี่ยมแทนบริเวณแผนภาพที่ถูกเก็บในที่พักข้อมูล

รูปที่ 2.16 การคำนวณ $D_{t,p}$ โดยใช้ขั้นตอนวิธี 2.3.6

ขั้นตอนวิธี 2.3.6 สามารถคำนวณระดับจุดภาพครั้งเดียว โดยการบวกกัน 8 ครั้ง หรือมีตัวดำเนินการ 8 ตัวดำเนินการ เพื่อรวมข้อมูลจากอาร์เรย์ของ 9 แฉงภาพ วิทยานิพนธ์นี้ต้องการลดตัวดำเนินการในการคำนวณ $D_{t,p}$ เพื่อเพิ่มความเร็วในการคำนวณ ด้วยการใช้วิธีอินทิกรัลอิมเมจ (integral image) ใน [5] และ สามารถเขียนเป็นสมการที่ (2.12)

$$D_{t,p(i-1,j-1)} = D_{t,p(i-2,j-1)} + D_{t,p(i-1,j-2)} - D_{t,p(i-2,j-2)} + d_{t,p(i,j)} - d_{t,p(i-3,j)} - d_{p(i,j-3)} + d_{(i-3,j-3)} \quad (2.12)$$

สมการที่ (2.12) ใช้ตัวดำเนินการน้อยกว่าขั้นตอนวิธี 2.3.6 และ ใช้ตัวดำเนินการเพียง 6 ครั้ง แต่ ขั้นตอนวิธี 2.3.7 ต้องการที่หักข้อมูลเพิ่มจากเดิมอีก $\frac{width}{w} + 1$ แฉงภาพ ดังนั้น อาร์เรย์ที่ใช้ในการหักข้อมูล ต้องสามารถเก็บผลการรวมความต่างระหว่างเฟรมกำลังสองของแฉงภาพเป็นจำนวน $3 \times \frac{width}{w} + 4$ แฉงภาพ สมการที่ (2.12) ต้องมีข้อมูล $D_{t,p(i-2,j-1)}$ $D_{t,p(i-1,j-2)}$ และ $D_{t,p(i-2,j-2)}$ ซึ่งการคำนวณครั้งแรก จะไม่มีข้อมูลเหล่านั้นเลย ดังนั้นการคำนวณครั้งแรกต้องใช้สมการที่ (2.11) การคำนวณแฉงภาพถัดไปจะ หักข้อมูล $D_{t,p(i-1,j-2)}$ กับ $D_{t,p(i-2,j-2)}$ จำเป็นต้องใช้สมการที่ (2.13)

$$D_{t,p(i-1,j=1)} = D_{t,p(i-2,j-1)} + d_{t,p(i,j)} + d_{t,p(i,j-1)} + d_{p(i,j-2)} - d_{(i-3,j)} - d_{(i-3,j-1)} - d_{(i-3,j-2)} \quad (2.13)$$

การคำนวณ $D_{t,p(i=1,j \geq 1)}$ จะหักข้อมูล $D_{t,p(i-2,j-1)}$ ดังนั้นการคำนวณ $D_{t,p(i=1,j \geq 1)}$ ต้องใช้ สมการที่ (2.14)

$$D_{t,p(i=1,j)} = D_{t,p(i-1,j-2)} + d_{t,p(i,j)} + d_{t,p(i-1,j)} + d_{p(i-2,j)} - d_{(i,j-3)} - d_{(i-1,j-3)} - d_{(i-2,j-3)} \quad (2.14)$$

สมการที่ (2.12) สมการที่ (2.13) และสมการที่ (2.14) ใช้ตัวดำเนินการน้อยกว่าสมการที่ (2.11) โดย ใช้ตัวดำเนินการเพียง 6 ตัวดำเนินการ ตามขั้นตอนวิธีนี้จะมีเพียงการคำนวณครั้งแรกเท่านั้นที่ต้องใช้ตัวเดิน การ 8 ตัวดำเนินการ เราแสดงรูปแบบที่ใช้ในการคำนวณ $D_{t,p}$ ในขั้นตอนวิธี 2.3.7 ด้วยภาพในรูปที่ 2.17

ขั้นตอนวิธีที่ใช้ในการคำนวณ $D_{t,p}$ สามารถคำนวณเร็วขึ้นได้ แต่ต้องการพื้นที่หน่วยความจำมากขึ้น มีเงื่อนไขมากขึ้น ซึ่งเห็นได้ชัดเจนจากการเปรียบเทียบขั้นตอนวิธี 2.3.5 ขั้นตอนวิธี 2.3.6 และ ขั้นตอนวิธี 2.3.7 ในตารางที่ 2.1 เป็นการเปรียบเทียบจำนวนตัวดำเนินการ และขนาดของที่หักข้อมูลที่ต้องใช้ สิ่งสามารถเปรียบเทียบ ความเร็วในการประมวลผลจากสมการคือ จำนวนของตัวดำเนินการ สมการที่ใช้ตัวดำเนินการที่น้อยกว่า แสดง ว่าใช้เวลาในการประมวลผลน้อยกว่า

ขั้นตอนวิธี 2.3.7

Require: *buf_address*|*buf_address* = ตำแหน่งเก็บข้อมูลของที่พักข้อมูล

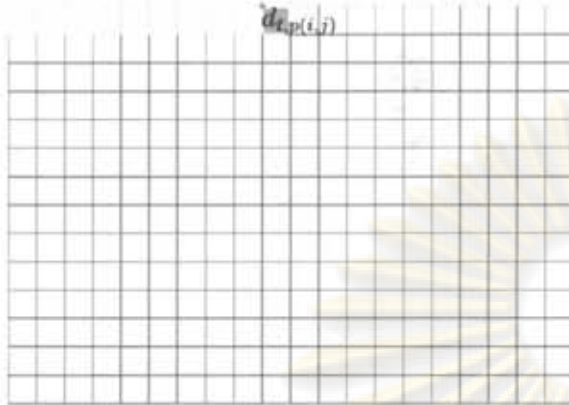
Require: *column_address*|*column_address* = ตำแหน่งของแผงภาพในแนวระนาบ

Require: *flag_computation*|*flag_computation* = true \longleftrightarrow มีข้อมูลเพียงพอที่จะคำนวณ $D_{t,p}$

```

1: flag_computation  $\leftarrow$  0
2: for j in {0, w, 2w, 3w, ..., height - 2w, height - w} do
3:   for i in {0, w, 2w, 3w, ..., width - 2w, width - w} do
4:      $d_{t,p(i,j)} \leftarrow 0$ 
5:     for v in {j, j + 1, j + 2, ..., j + w} do
6:       for u in {i, i + 1, i + 2, ..., i + w} do
7:          $d_{t,p(i,j)} \leftarrow d_{t,p(i,j)} + (\Delta I_t(u, v))^2$ 
8:       end for
9:     end for
10:    array[buf_address]  $\leftarrow d_{t,p(i,j)}$ 
11:    buf_address  $\leftarrow$  buf_address + 1
12:    if buf_address = ( $3 \times \frac{\text{width}}{w} + 4$ ) then
13:      buf_address  $\leftarrow$  0
14:    end if
15:    if flag_computation = 2 then
16:      if column_address  $\geq$  2 then
17:         $D_{t,p(i-w,j-w)} \leftarrow D_{t,p(i-2w,j-w)} + D_{t,p(i-w,j-2w)} - D_{t,p(i-2w,j-2w)} + d_{t,p(i,j)} + d_{t,p(i-3,j)} +$ 
           $d_{p(i,j-3)} - d_{(i-3,j-3)}$ 
18:      else if column_address = 1 then
19:         $D_{t,p(i-w,j-w)} \leftarrow D_{t,p(i-w,j-2w)} + d_{t,p(i,j)} + d_{t,p(i-1,j)} + d_{p(i-2,j)} - d_{p(i,j-3)} - d_{p(i-1,j-3)} -$ 
           $d_{p(i-2,j-3)}$ 
20:      end if
21:      else if flag_computation = 1 then
22:         $D_{t,p(i-w,j-w)} \leftarrow D_{t,p(i-2w,j-w)} + d_{t,p(i,j)} + d_{t,p(i,j-1)} + d_{p(i,j-2)} - d_{(i-3,j)} - d_{(i-3,j-1)} -$ 
           $d_{(i-3,j-2)}$ 
23:      else if flag_computation = 0 then
24:        if buf_address = ( $2 \times \frac{\text{width}}{w} + 4$ ) then
25:          flag_computation  $\leftarrow$  1
26:           $D_{t,p(i-w,j-w)} \leftarrow d_{p(i,j)} + d_{p(i-1,j)} + d_{p(i-2,j)} + d_{p(i,j-1)} + d_{p(i-1,j-1)} + d_{p(i-2,j-1)} +$ 
             $d_{p(i,j-2)} + d_{p(i-1,j-2)} + d_{p(i-2,j-2)}$ 
27:        end if
28:      end if
29:      column_address  $\leftarrow$  column_address + 1
30:    end for
31: end for

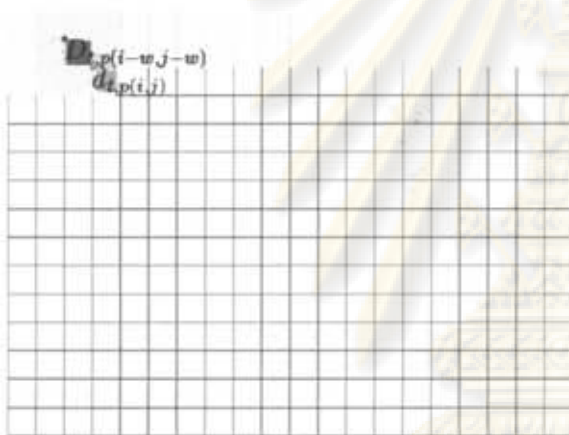
```



(ก) การเก็บข้อมูล $d_{t,p(i=9,j=9)}$ โดยบริเวณสีเทา แทนบริเวณที่มีการคำนวณระดับจุดภาพ บริเวณสีเหลืองแทนบริเวณแผนภาพที่ถูกเก็บในที่พักข้อมูล



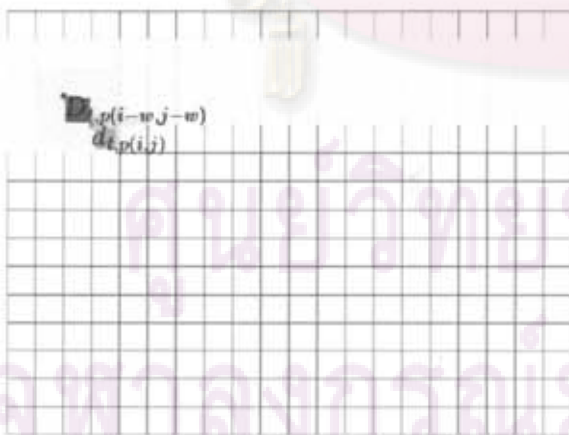
(ข) การคำนวณ $D_{t,p(i=9,j=9)}$ เป็นการคำนวณครั้งแรก โดยบริเวณสีเทาแทนบริเวณที่มีการคำนวณระดับจุดภาพ บริเวณสีเหลืองแทนบริเวณแผนภาพที่ถูกเก็บในที่พักข้อมูล



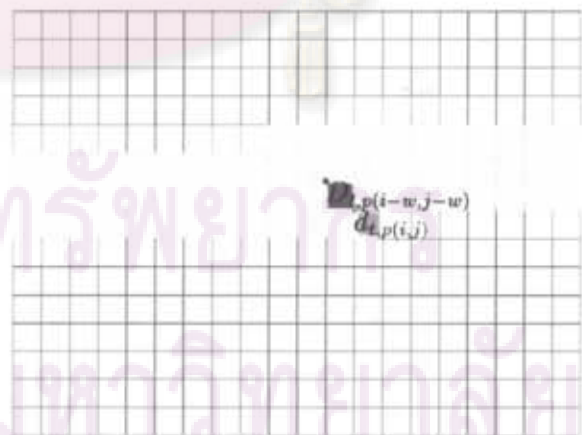
(ค) การคำนวณ $D_{t,p(i=2,j=2)}$ โดยบริเวณสีเทาแทนบริเวณที่มีการคำนวณระดับจุดภาพ บริเวณสีเหลืองแทนบริเวณแผนภาพที่ถูกเก็บในที่พักข้อมูล



(ง) การคำนวณ $D_{t,p(i=3,j=3)}$ โดยบริเวณสีเทาแทนบริเวณที่มีการคำนวณระดับจุดภาพ บริเวณสีเหลืองแทนบริเวณแผนภาพที่ถูกเก็บในที่พักข้อมูล



(จ) การคำนวณ $D_{t,p(i=2,w,j=3-w)}$ โดยบริเวณสีเทาแทนบริเวณที่มีการคำนวณระดับจุดภาพ บริเวณสีเหลืองแทนบริเวณแผนภาพที่ถูกเก็บในที่พักข้อมูล



(ฉ) การคำนวณ $D_{t,p(i=11-w,j=6-w)}$ โดยบริเวณสีเทาแทนบริเวณที่มีการคำนวณระดับจุดภาพ บริเวณสีเหลืองแทนบริเวณแผนภาพที่ถูกเก็บในที่พักข้อมูล

รูปที่ 2.17 การคำนวณ $D_{t,p}$ โดยใช้ขั้นตอนวิธี 2.3.7

ตารางที่ 2.1 เปรียบเทียบขั้นตอนวิธี 2.3.5 2.3.6 และ 2.3.7

รายการ	ขั้นตอนวิธี 2.3.5	ขั้นตอนวิธี 2.3.6	ขั้นตอนวิธี 2.3.7
ขนาดของที่พักข้อมูล (แผงภาพ)	0	$2 \times \frac{width}{w} + 3$	$3 \times \frac{width}{w} + 4$
การคำนวณ $d_{t,p}$ 1 แผงภาพ (ตัวดำเนินการ)	0	w^2	w^2
การคำนวณ $D_{t,p}$ 1 แผงภาพ (ตัวดำเนินการ)	$(3 \cdot w)^2$	8	6

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

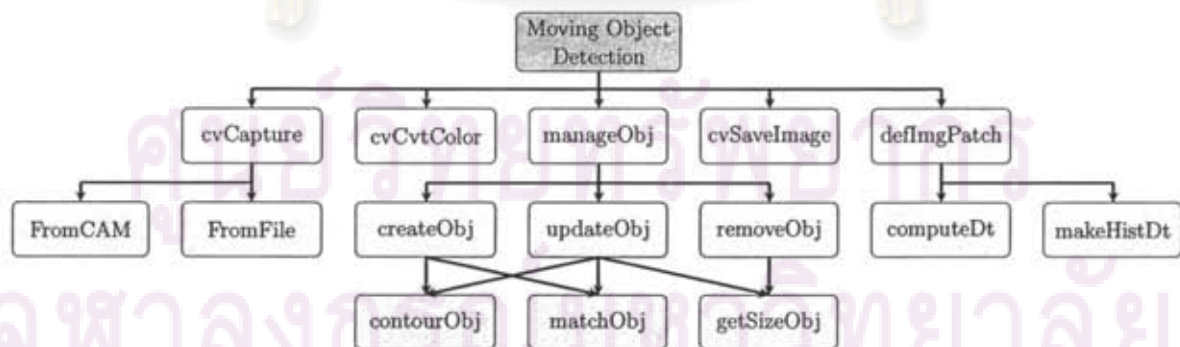
บทที่ 3

การพัฒนางานวิจัย

เดิมการพัฒนางานวิจัยในวิทยานิพนธ์นี้เป็นการติดตั้งการตรวจจับการเคลื่อนที่ใน บอร์ด DE2 ของ Altera โดยใช้ขั้นตอนวิธีของ Lu [2] ตามที่เสนอไป แต่ขั้นตอนวิธีของ Lu [2] ใช้ค่าขีดเริ่มเปลี่ยนคงที่ และต้องกรองภาพ 4 ครั้ง เราเห็นชัดว่าขั้นตอนวิธีของ Lu [2] ไม่คงทนต่อสถานะการด์ที่เปลี่ยนแปลงตลอด รวมทั้งการกรองภาพ 4 ครั้ง เป็นขั้นตอนที่ใช้เวลานานมาก ดังนั้น การใช้บอร์ด DE2 เพื่อลดเวลาที่ใช้ในการกรองภาพ เป็นการแก้ปัญหาที่ขึ้นกับอุปกรณ์ที่มีราคาสูง และไม่สามารถแก้ปัญหาค่าขีดเริ่มเปลี่ยนคงที่

เราพัฒนาโปรแกรมบนระบบปฏิบัติการ GNU/Linux Debian เพราะเป็นระบบเปิด และสามารถใช้งานบนระบบปฏิบัติการ GNU/Linux Emdebian ซึ่งเป็นระบบปฏิบัติการที่ทำงานบนระบบฝังตัว การพัฒนาโปรแกรมบนระบบปฏิบัติการดังกล่าวทำให้เราสามารถใช้งานได้ทั้งเครื่องคอมพิวเตอร์ตั้งโต๊ะ และระบบฝังตัว และระบบปฏิบัติการทั้งสองมีโปรแกรมสำหรับส่งข้อมูลผ่านเครือข่ายอย่าง apache2 หรือ proftpd เราสามารถพัฒนาระบบฝังตัวเป็นระบบดั่งรูปที่ 1.1(ข) ได้ด้วยระบบปฏิบัติการ GNU/Linux Emdebiens การตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอในงานวิจัยนี้เลือกใช้โปรแกรม g++ สำหรับการโปรแกรมด้วยภาษา C/C++ ร่วมกับฟังก์ชันประมวลผลภาพในไลบรารีโอเพ่นซีวี (OpenCV, Open Computer Vision Library) ซึ่งเป็นไลบรารีที่ถูกออกแบบและพัฒนาโดยบริษัทอินเทลคอร์ปอเรชันสำหรับคอมพิวเตอร์วิทัศน์เวลาจริง

3.1 โครงสร้างโปรแกรมตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอ



รูปที่ 3.1 โครงสร้างโปรแกรมตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอโดยใช้ระบบฝังตัว

โครงสร้างโปรแกรมตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอโดยใช้ระบบฝังตัว ประกอบด้วยฟังก์ชัน



รูปที่ 3.2 การไหลข้อมูลการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอที่สนโดยใช้ระบบฝังตัว

จุฬาลงกรณ์มหาวิทยาลัย

ต่าง ๆ ดังรูปที่ 3.1 โดยเป็นฟังก์ชันในไลบรารีโอเพ่นซีวีซึ่งเป็นฟังก์ชันพื้นฐาน กับฟังก์ชันที่สร้างขึ้นเพื่อรองรับเทคนิคที่วิทยานิพนธ์นี้นำเสนอ แต่ละฟังก์ชันมีลำดับการทำงานตามการไหลข้อมูลดังรูปที่ 3.2 และมีหน้าที่การทำงานดังต่อไปนี้

1. cvCapture เป็นฟังก์ชันในไลบรารีโอเพ่นซีวี โดยมีคำสั่ง cvCaptureFromCAM ทำหน้าที่รับลำดับภาพจากกล้องวิดีโอ และ cvCaptureFromFile ทำหน้าที่รับลำดับภาพจากไฟล์วิดีโอ ขั้นตอนการทดสอบขั้นตอนวิธีเราใช้ cvCaptureFromFile เพื่อทดสอบตามเหตุการณ์ในไฟล์ที่ควรทำได้ เมื่อได้ขั้นตอนวิธีที่พอใจเราเปลี่ยนจาก cvCaptureFromFile เป็น cvCaptureFromCAM ภาพที่ได้จากฟังก์ชันนี้เป็นภาพสีแดงเขียวน้ำเงิน
2. cvCvtColor เป็นฟังก์ชันในไลบรารีโอเพ่นซีวี ฟังก์ชันนี้มีหน้าที่แปลงภาพสีแดงเขียวน้ำเงิน เป็นภาพความเข้มแสง หรือ ภาพสเกลเทา ฟังก์ชันนี้รับข้อมูลจากฟังก์ชัน cvCapture
3. computeDt เป็นฟังก์ชันคำนวณหา $D_{t,p}$ จากภาพความเข้มแสง และถูกสร้างขึ้นตามขั้นตอนวิธี 2.3.7
4. makeHistDt เป็นฟังก์ชันที่นำ $D_{t,p}$ มาแจกแจงความถี่ตั้งแต่ $2^0 - 2^1, 2^1 - 2^2, \dots, 2^{30} - 2^{31}$ เพื่อหาค่าขีดเริ่มเปลี่ยน δ_{min} กับค่าขีดเริ่มเปลี่ยน δ_{max}
5. defImgPatch ภายในส่วนนี้มีสองขั้นตอนคือการคำนวณ $D_{t,p}$ และสร้างฮิสโตแกรมของ $D_{t,p}$ ฟังก์ชันนี้ทำหน้าที่ตัดสินแผนภาพเป็นแผนภาพเคลื่อนที่หรือเป็นแผนภาพหยุดนิ่ง โดยใช้ผลลัพธ์จากฟังก์ชัน computeDt กับ makeHistDt การทำงานของฟังก์ชันนี้ใช้แบบจำลองตามที่เสนอในรูปที่ 2.4 เพื่อคำนวณหาวัตถุเคลื่อนที่ในเฟรม กับ รูปที่ 2.6 เพื่อทราบว่าไม่มีการเคลื่อนที่ที่เกิดขึ้นระหว่างเฟรมนั้น และการคำนวณแผนภาพที่สอดคล้องกับภาพที่เฟรมหนึ่งด้วยขั้นตอนวิธี 2.1.3
6. contourObj เป็นฟังก์ชันที่สร้างกรอบสี่เหลี่ยมที่ครอบคลุมวัตถุเคลื่อนที่จากคอนทัวร์ของแผนภาพเคลื่อนที่เริ่มจากการสร้างคอนทัวร์จากแผนภาพเคลื่อนที่จากฟังก์ชัน defImgPatch ด้วยฟังก์ชัน cvContour ซึ่งเป็นฟังก์ชันในไลบรารีโอเพ่นซีวี
7. matchObj ฟังก์ชันตรวจสอบการซ้อนทับกันระหว่างกรอบสี่เหลี่ยมสองรูป ระหว่างผลลัพธ์ของฟังก์ชัน contourObj ซึ่งเป็นกรอบสี่เหลี่ยมที่ครอบคลุมวัตถุเคลื่อนที่ กับกรอบสี่เหลี่ยมของวัตถุเคลื่อนที่ที่ต้องการจดจำ
8. getSizeObj ฟังก์ชันเปรียบเทียบเพื่อหาความยาวของด้านที่เล็กที่สุดระหว่างด้านกว้างกับด้านสูงของกรอบสี่เหลี่ยมที่ครอบคลุมวัตถุเคลื่อนที่
9. manageObj ฟังก์ชันนี้ถูกสร้างขึ้นเพื่อจดจำวัตถุเคลื่อนที่ซึ่งหยุดการเคลื่อนที่และยังไม่ได้ออกไปจากบริเวณที่เฝ้าระวังอยู่ ในส่วนนี้มีหน้าที่หลักคือการจดจำวัตถุใหม่ที่เข้ามา อัปเดตข้อมูลของวัตถุเคลื่อนที่ และกำจัดข้อมูลของวัตถุที่พ้นจากบริเวณเฝ้าระวังแล้ว โดยใช้ผลลัพธ์จาก createObj updateObj และ

removeObj ตามลำดับ เริ่มจากการสร้างคอนทัวร์ของแผนภาพเคลื่อนที่จากฟังก์ชัน contourObj เราสร้างกรอบสี่เหลี่ยมที่ครอบวัตถุเคลื่อนที่จากคอนทัวร์ของแต่ละวัตถุ เราตรวจสอบความใหม่ของวัตถุเคลื่อนที่ในแต่ละเฟรมด้วยฟังก์ชัน matchObj ถ้าเป็นวัตถุที่มีได้ซ้อนทับกับกรอบสี่เหลี่ยมของวัตถุที่มีอยู่เดิมแล้ว แสดงว่าวัตถุนั้นเป็นวัตถุเคลื่อนที่ตัวใหม่ เราจะสร้างข้อมูลเพื่อการจดจำวัตถุใหม่นี้ แต่ถ้ามีการซ้อนทับกับวัตถุเดิมจะต้องอัปเดตข้อมูลใหม่ตามเงื่อนไขในขั้นตอนวิธี 2.2.4 สุดท้ายเราตรวจสอบขนาดด้วยฟังก์ชัน getSizeObj เราต้องการเพียงด้านกว้าง หรือด้านยาว โดยเราจะกำจัดข้อมูลที่จะจดจำวัตถุเคลื่อนที่หนึ่ง ๆ ออกไปถ้าตรงตามเงื่อนไขในสมการที่ (2.9)

10. cvSaveImage บันทึกภาพผลลัพธ์ของขบวนการที่เสนอในวิทยานิพนธ์นี้ การบันทึกด้วยฟังก์ชันนี้ช่วยให้ไฟล์ที่ถูกบันทึกมีขนาดเล็กลงด้วย ทำให้เหมาะแก่การส่งข้อมูลในระยะไกล

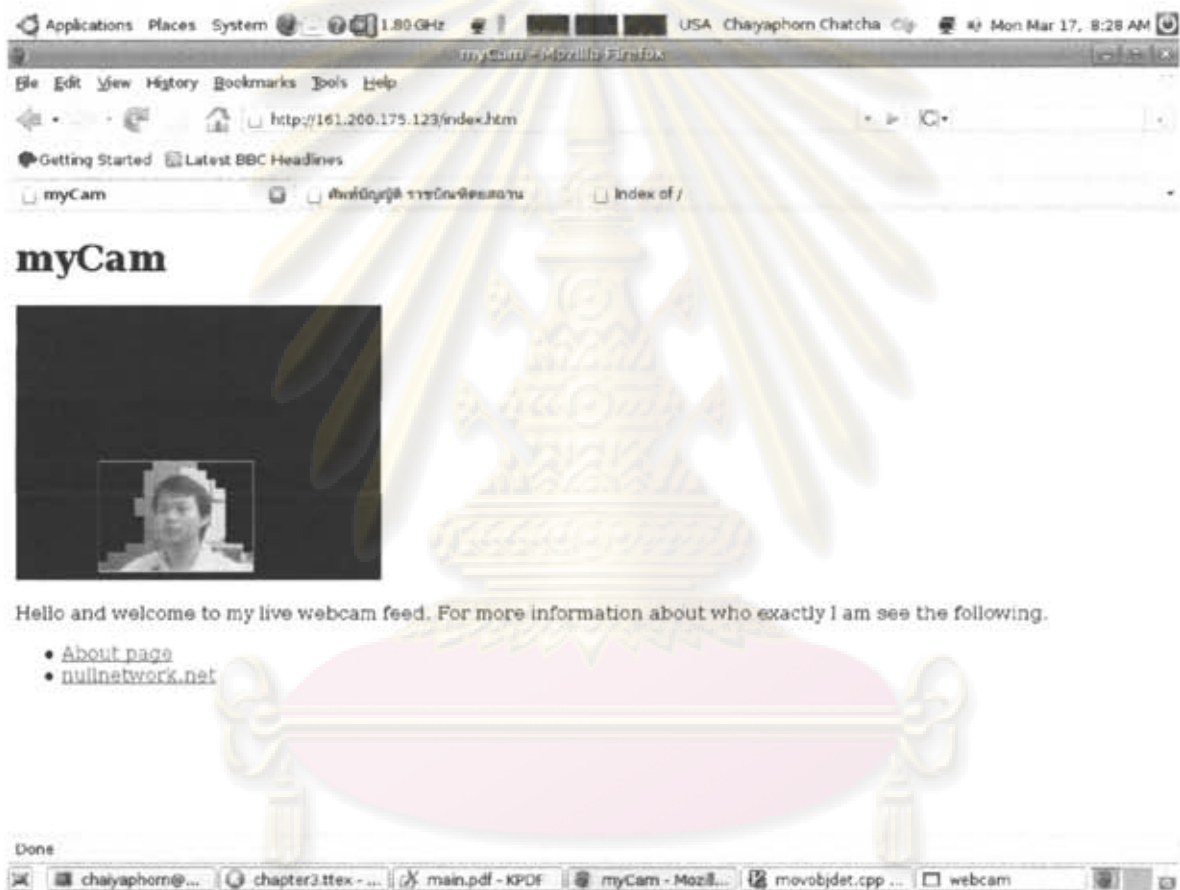
3.2 การส่งภาพผ่านเครือข่าย

ภายในงานวิจัยนี้ต้องการใช้ในระบบฝังตัว และส่งข้อมูลผ่านเครือข่ายโดยใช้ระบบฝังตัวด้วย เพื่อการทำงานดังรูปที่ 1.1(ข) เราใช้ระบบปฏิบัติการ GNU/Linux Emdebian ซึ่งมีโปรแกรม apache2 เป็นโปรแกรมสำหรับสร้างเว็บเซิร์ฟเวอร์ ช่วยรายงานผลการตรวจหาวัตถุในลำดับภาพวิดีโอที่ส่งจากระบบฝังตัวผ่านเว็บเบราว์เซอร์ได้ โดยยูอาร์แอลของระบบฝังตัวคือตำแหน่งไอพีของระบบฝังตัว เช่น ระบบฝังตัวมีตำแหน่งไอพีเป็น 161.200.175.123 เรารับภาพผลลัพธ์ผ่านเว็บเบราว์เซอร์โดยยูอาร์แอล <http://161.200.175.123/index.htm> ดังรูปที่ 3.3 ไฟล์ index.htm เป็นไฟล์ที่สามารถทำให้เว็บเบราว์เซอร์ปลายทางอัปเดตภาพอัตโนมัติ ภายในไฟล์ index.htm ต้องมีรายละเอียดที่หัวไฟล์ดังนี้

```
<meta http-equiv="refresh" content="2">
```

การเข้าถึงข้อมูลด้วยโปรแกรม apache2 เป็นวิธีที่สะดวกในการเข้าถึงรูปภาพผลลัพธ์จากกล้อง แต่เมื่อความปลอดภัยในการเข้าถึงข้อมูลมีความสำคัญ เราใช้โปรแกรมสำหรับส่งข้อมูลชนิดอื่นแทนได้เช่น vsftpd ซึ่งเป็นโปรแกรมส่งข้อมูลแบบ FTP ที่มีความปลอดภัยสูงกว่าการเข้าถึงด้วยเว็บเบราว์เซอร์

การพัฒนาในวิทยานิพนธ์นี้แบ่งการทำงานเป็นสองส่วนคือ การตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอที่ใช้ระบบฝังตัว กับการรายงานผลลัพธ์ของการตรวจหาวัตถุเคลื่อนที่โดยใช้ระบบฝังตัวผ่านทางเครือข่าย การตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอที่เราใช้เทคนิคที่นำเสนอเป็นสำคัญ โดยพัฒนาโปรแกรมด้วยภาษา C/C++ สำหรับการรายงานผลลัพธ์ผ่านเครือข่ายเราใช้โปรแกรมที่มีใน GNU/Linux Emdebian ช่วยให้การพัฒนามีประสิทธิภาพและรวดเร็ว



รูปที่ 3.3 การรายงานการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวีดิทัศน์โดยใช้ระบบฝังตัว

ศูนย์วิจัยทรัพย์สินทางปัญญา
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

การทดสอบโปรแกรมตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอ

การทดสอบโปรแกรมตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอ ในสถานะการณ์ที่มีการเปลี่ยนแปลงของแสงที่ละน้อยในบริเวณทดสอบ เช่นแสงอาทิตย์ โดยเฉพาะเวลากลางวัน เป็นช่วงเวลาที่แสงมีการรบกวนมากกว่าช่วงเวลาอื่น และการรบกวนอันเนื่องจากการเคลื่อนไหวเล็กน้อย เช่น การเคลื่อนไหวของกิ่งไม้หรือใบไม้ในบริเวณทดสอบ เพื่อทดสอบความสามารถในการใช้งานในที่สาธารณะ หรือภายนอกอาคาร ซึ่งการรบกวนอันเนื่องมาจากแสงอาทิตย์กับการรบกวนของกิ่งไม้ ใบไม้ในบริเวณทดสอบ เป็นปัญหาใหญ่สำหรับการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอด้วยวิธีการลบภาพพื้นหลัง การทดสอบนี้แสดงความสามารถที่มีมากกว่าวิธีการลบภาพพื้นหลังในแง่ของผลลัพธ์ที่ประมวลผลได้ กับเวลาที่ใช้ในการคำนวณที่น้อยกว่า เราทดสอบในสถานที่ต่าง ๆ ตามเวลาจริงในสถานะการณ์จริง พร้อมกับบันทึกลำดับภาพวิดีโอ เพื่อนับวัตถุเคลื่อนที่จริงในบริเวณทดสอบเปรียบเทียบกับผลที่นับได้จากโปรแกรม ที่ถูกสร้างด้วยวิธีการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอที่นำเสนอในวิทยานิพนธ์

ในแต่สถานะการณ์เราทดสอบด้วยแผนภาพสามขนาด คือ 4×4 , 8×8 และ 16×16 จุดภาพ เป็นการทดสอบความทนทานต่อการรบกวนในบริเวณเดียวกัน และผลลัพธ์ที่ได้ เมื่อใช้แผนภาพขนาดต่างกัน เราทดสอบการลดขนาดของไฟล์ภาพด้วยการแทนบริเวณของภาพพื้นหลังด้วยศูนย์ สุดท้ายเราวัดเวลาที่ใช้ในการประมวลผลบนเครื่องคอมพิวเตอร์ Lenovo X61 กับระบบฝังตัว Armadillo-9 เป็นการทดสอบความสามารถในการคำนวณ เพื่อตอบสนองการทำงานตามเวลาจริงของวิธีการที่นำเสนอในวิทยานิพนธ์นี้

4.1 การทดสอบการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอ

เราแบ่งการทดสอบออกเป็น 4 การทดสอบ ตามสถานที่ คือ ลานจอดรถหน้าตึกไฟฟ้าแรงสูง ถนนพหลโยธิน ถนนพญาไท และบริเวณคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย แต่ละสถานที่สามารถรับภาพในมุมมองที่ต่างกัน โดย การทดสอบที่ลานจอดรถหน้าตึกไฟฟ้าแรงสูง เป็นการติดตามบุคคลเคลื่อนที่ นอกนั้นเป็นการทดสอบตรวจจับยานพาหนะบนท้องถนน การทดสอบบนถนนพหลโยธิน เป็นการถ่ายภาพจากมุมสูง และสูงกว่าการทดสอบบนถนนพญาไท สำหรับการทดสอบในมุมมองที่แตกต่างจากการทดสอบทั้งสาม เราทดสอบโปรแกรมในมุมมองระดับสายตา บริเวณคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

4.1.1 การทดสอบที่ลานจอดรถหน้าตึกไฟฟ้าแรงสูง

การทดสอบติดตามบุคคลเคลื่อนที่ในลำดับภาพ ณ ลานจอดรถหน้าตึกไฟฟ้าแรงสูง ซึ่งเป็นสถานที่เปิดโล่ง มีต้นไม้ และสภาพแสงแดดที่ไม่คงที่ในบริเวณทดสอบ การทดสอบด้วยแผงภาพขนาด 4×4 , 8×8 และ 16×16 จุดภาพ ในตารางที่ 4.1, 4.2 และ 4.3 เห็นชัดว่าแผงภาพขนาดใหญ่ให้ผลลัพธ์ที่มีเสถียรภาพมากกว่าแผงภาพขนาดเล็ก และในรูปที่ 4.1, 4.2 และ 4.3 แสดงให้เห็นว่าการใช้แผงภาพขนาดใหญ่จะได้ผลลัพธ์เป็นกรอบวัตถุเคลื่อนที่มีขนาดต่างจากวัตถุเคลื่อนที่จริง มากกว่าการใช้แผงภาพขนาดเล็ก โดยรูปบนของทุกแผงภาพย่อย ของรูปที่ 4.1, 4.2 และ 4.3 แสดงผลลัพธ์ของการตรวจหาวัตถุเคลื่อนที่ และรูปล่างของทุกแผงภาพย่อย ของรูปที่ 4.1, 4.2 และ 4.3 แสดงสีสโทแกรมของการรวมผลต่างกำลังสอง แกน x แสดงความเข้มแสงกำลังสอง แกน y แสดง ความถี่ของแผงภาพ

ตารางที่ 4.1 ผลการทดสอบ ณ ลานจอดรถตึกไฟฟ้าแรงสูง แผงภาพขนาด 4×4 จุดภาพ

ภาพลำดับที่	วัตถุเคลื่อนที่จริง	โปรแกรมนับวัตถุเคลื่อนที่ได้	นับถูก (เฟรม)	นับผิด (เฟรม)
900 – 953	1	1	54	0
954 – 955	1	2	0	2
956 – 959	1	1	4	0
960 – 960	1	2	0	1
961 – 972	1	1	12	0
973 – 978	1	2	0	6
979 – 980	1	1	2	0
981 – 986	1	2	0	6
987 – 987	1	1	1	0
988 – 999	1	2	0	12
			รวม 73	รวม 27

ตารางที่ 4.2 ผลการทดสอบ ณ ลานจอดรถตึกไฟฟ้าแรงสูง แผงภาพขนาด 8×8 จุดภาพ

ภาพลำดับที่	วัตถุเคลื่อนที่จริง	โปรแกรมนับวัตถุเคลื่อนที่ได้	นับถูก (เฟรม)	นับผิด (เฟรม)
700 – 799	1	1	100	0
			รวม 100	รวม 0

ตารางที่ 4.3 ผลการทดสอบ ณ ลานจอตกรตึกไฟฟ้าแรงสูง แผงภาพขนาด 16×16 จุดภาพ

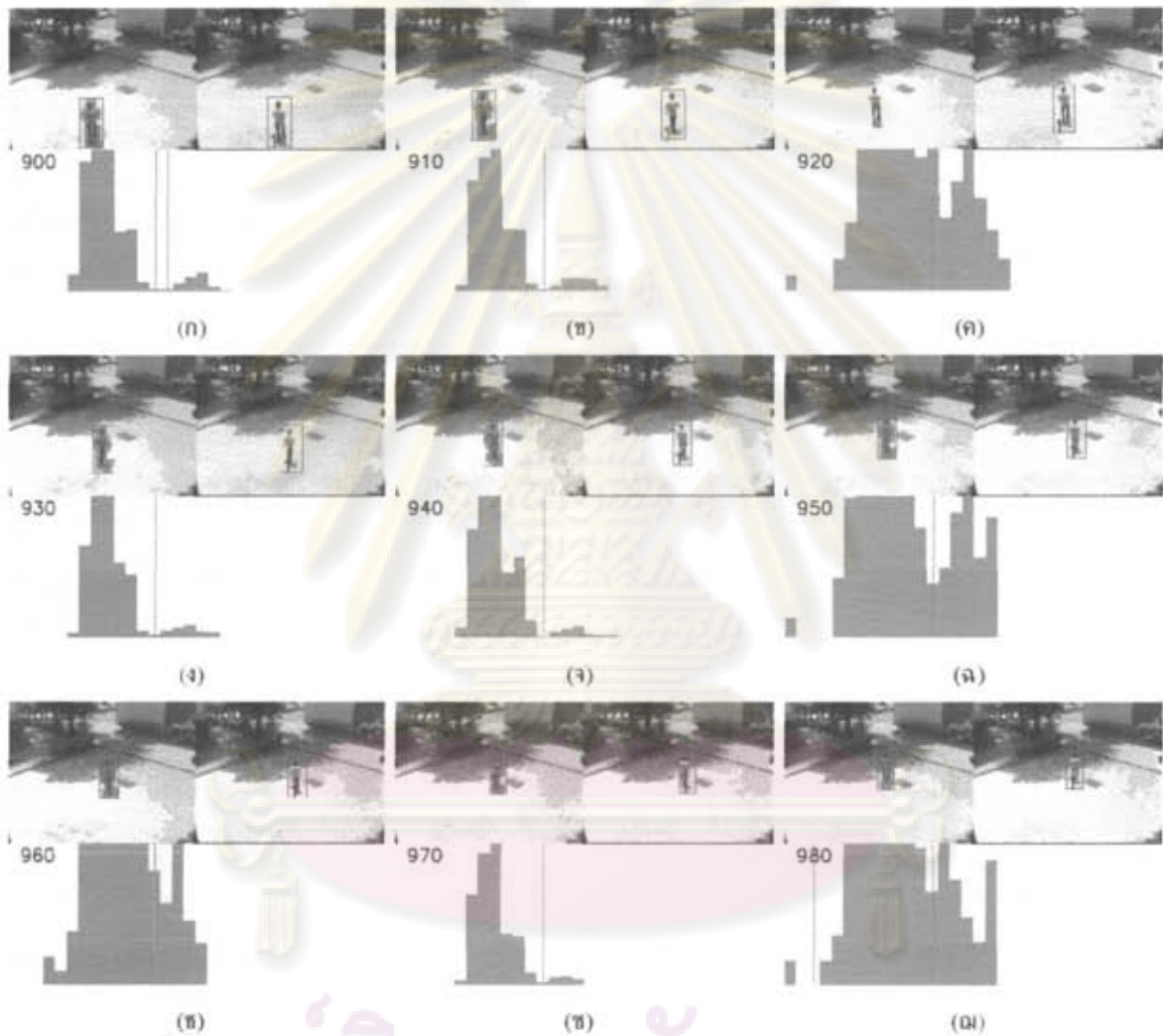
ภาพลำดับที่	วัตถุเคลื่อนที่จริง	โปรแกรมนับวัตถุเคลื่อนที่ได้	นับถูก (เฟรม)	นับผิด (เฟรม)
700 – 799	1	1	100	0
			รวม 100	รวม 0

4.1.2 การทดสอบบนถนนพหลโยธิน

การทดสอบนี้มีการรบกวนอันเนื่องจากแสงแดดที่เปลี่ยนแปลงที่ละน้อยตลอดการทดลอง รูปที่ 4.4, 4.5 และ 4.6 เป็นภาพตัวอย่างของผลการทดสอบบนถนนพหลโยธิน ด้วยแผงภาพขนาด 4×4 , 8×8 และ 16×16 จุดภาพ ตามลำดับ โดย รูปบนของทุกแผงภาพย่อย ของรูปที่ 4.4, 4.5 และ 4.6 แสดงผลลัพธ์ของการตรวจหาวัตถุเคลื่อนที่ และรูปล่างของทุกแผงภาพย่อย ของรูปที่ 4.4, 4.5 และ 4.6 แสดงฮิสโทแกรมของการรวมผลต่างกำลังสอง แกน x แสดงความเข้มแสงกำลังสอง แกน y แสดง ความถี่ของแผงภาพ เราแสดงการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอที่ถูกรบกวนจากแสงแดด โดยทดลองบนถนนพหลโยธินหน้ามหาวิทยาลัยกรุงเทพ ซึ่งมีสะพานลอยที่สูงเหมาะกับการทดลอง เราสามารถตรวจหาวัตถุเคลื่อนที่บนท้องถนนได้ตั้งภาพตัวอย่างผลการทดสอบ และผลการทดสอบเห็นชัดว่า การใช้แผงภาพขนาดใหญ่ จะได้ผลลัพธ์เป็นกรอบวัตถุเคลื่อนที่มีขนาดต่างจากวัตถุเคลื่อนที่จริงมากกว่าการใช้แผงภาพขนาดเล็ก

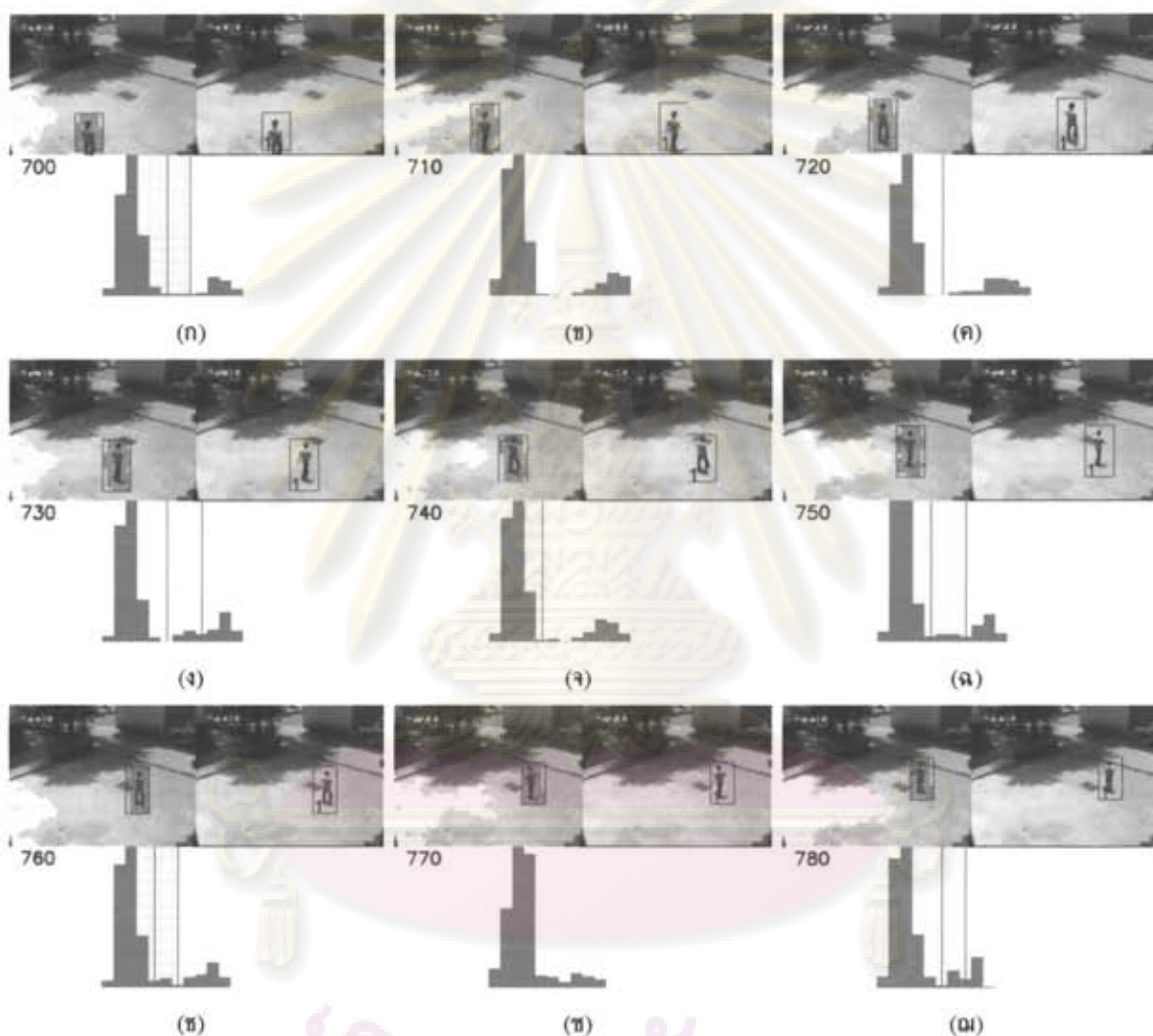
4.1.3 การทดสอบบนถนนพญาไท

การทดสอบนี้มีการรบกวนอันเนื่องจากแสงแดดที่เปลี่ยนแปลงที่ละน้อยตลอดการทดลอง รูปที่ 4.7, 4.8 และ 4.9 เป็นภาพตัวอย่างของผลการทดสอบบนถนนพญาไท ด้วยแผงภาพขนาด 4×4 , 8×8 และ 16×16 จุดภาพ ตามลำดับ โดย รูปบนของทุกแผงภาพย่อย ของรูปที่ 4.7, 4.8 และ 4.9 แสดงผลลัพธ์ของการตรวจหาวัตถุเคลื่อนที่ และรูปล่างของทุกแผงภาพย่อย ของรูปที่ 4.7, 4.8 และ 4.9 แสดงฮิสโทแกรมของการรวมผลต่างกำลังสอง แกน x แสดงความเข้มแสงกำลังสอง แกน y แสดง ความถี่ของแผงภาพ เราแสดงการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอที่ถูกรบกวนจากแสงแดด โดยทดลองบนถนนพญาไท ซึ่งมีสะพานลอยที่ต่ำกว่าการทดลองแรก ความสูงระดับนี้เราสามารถตรวจหาวัตถุเคลื่อนที่บนท้องถนนได้ ตั้งภาพตัวอย่างผลการทดสอบ และผลการทดสอบเห็นชัดว่า การใช้แผงภาพขนาดใหญ่ จะได้ผลลัพธ์เป็นกรอบวัตถุเคลื่อนที่มีขนาดต่างจากวัตถุเคลื่อนที่จริงมากกว่าการใช้แผงภาพขนาดเล็ก



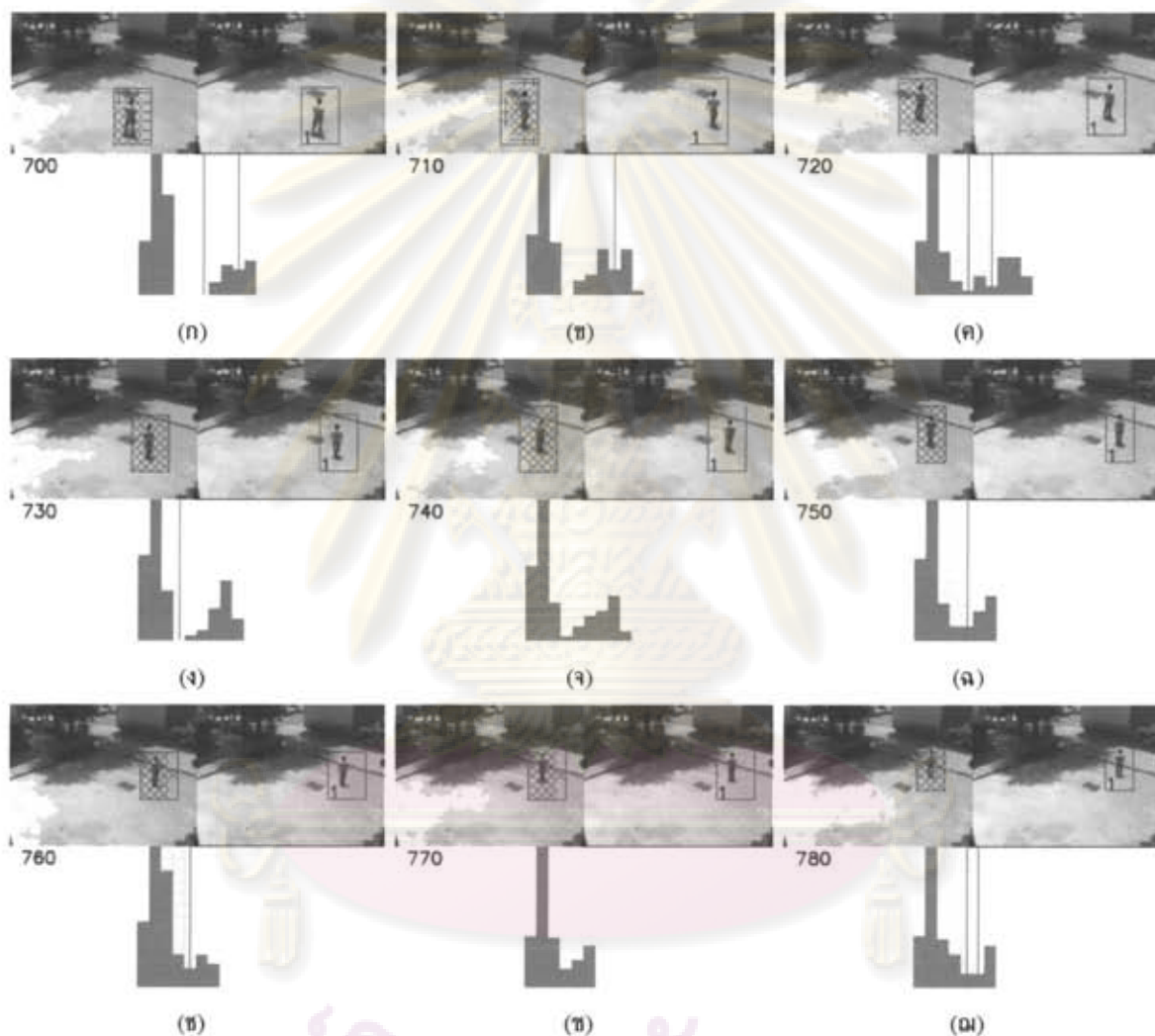
รูปที่ 4.1 ทดสอบด้วยแผงภาพขนาด 4×4 จุดภาพ ณ ลานจอดรถหน้าตึกไฟฟ้าแรงสูง โดยครอบแดงเป็นข้อมูลที่ถูกจดจำ กากบาทแดงเป็นแผงภาพเคลื่อนที่

ศูนย์วิทยพัชกร
จุฬาลงกรณ์มหาวิทยาลัย



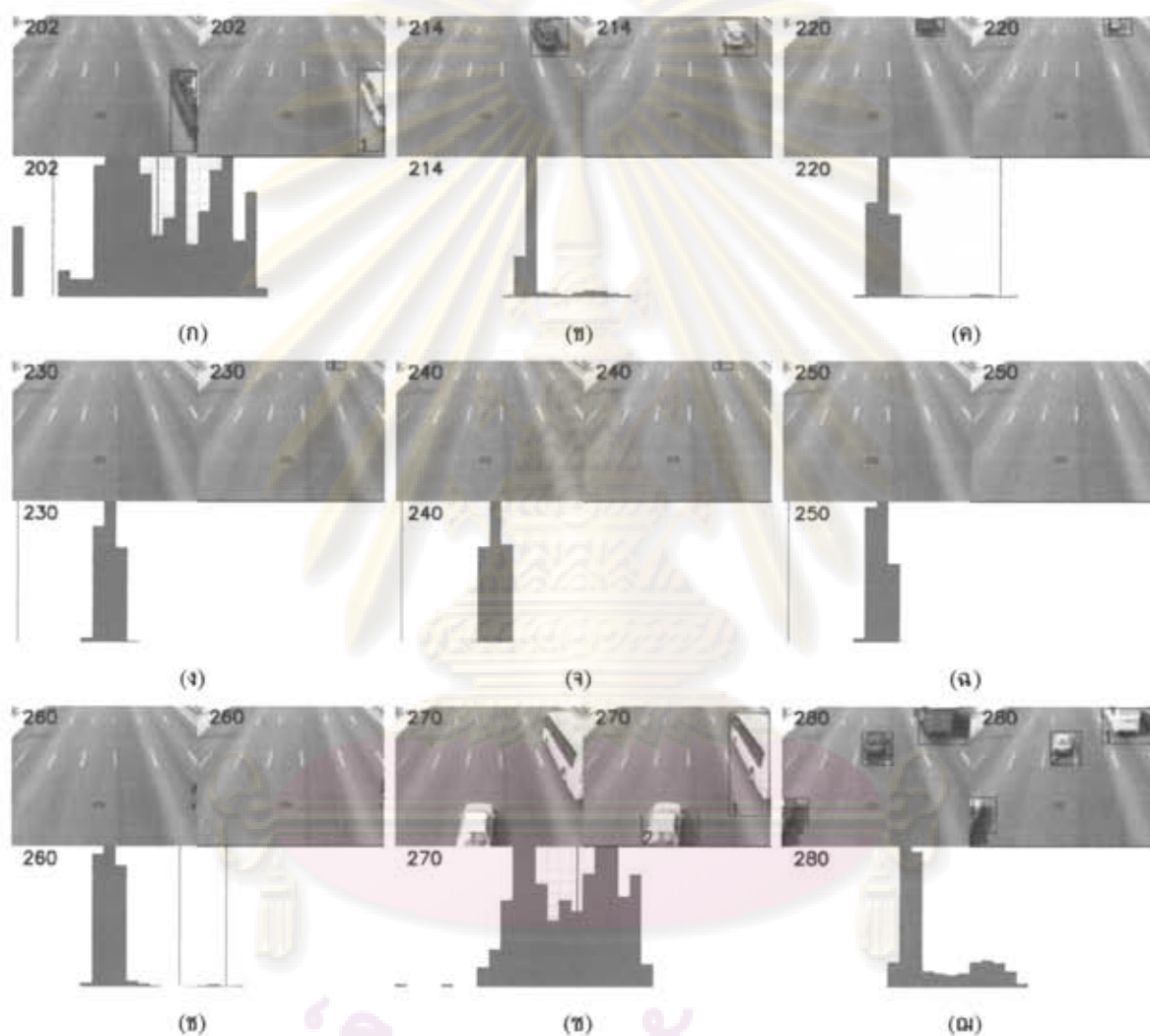
รูปที่ 4.2 ทดสอบด้วยแมงภาพขนาด 8×8 จุดภาพ ณ ลานจอดรถหน้าตึกไฟฟ้าแรงสูง โดยกรอบแดงเป็นข้อมูลที่ถูกรั่วจำ กากบาทแดงเป็นแมงภาพเคลื่อนที่

ศูนย์วิทยุตำรวจ
จุฬาลงกรณ์มหาวิทยาลัย



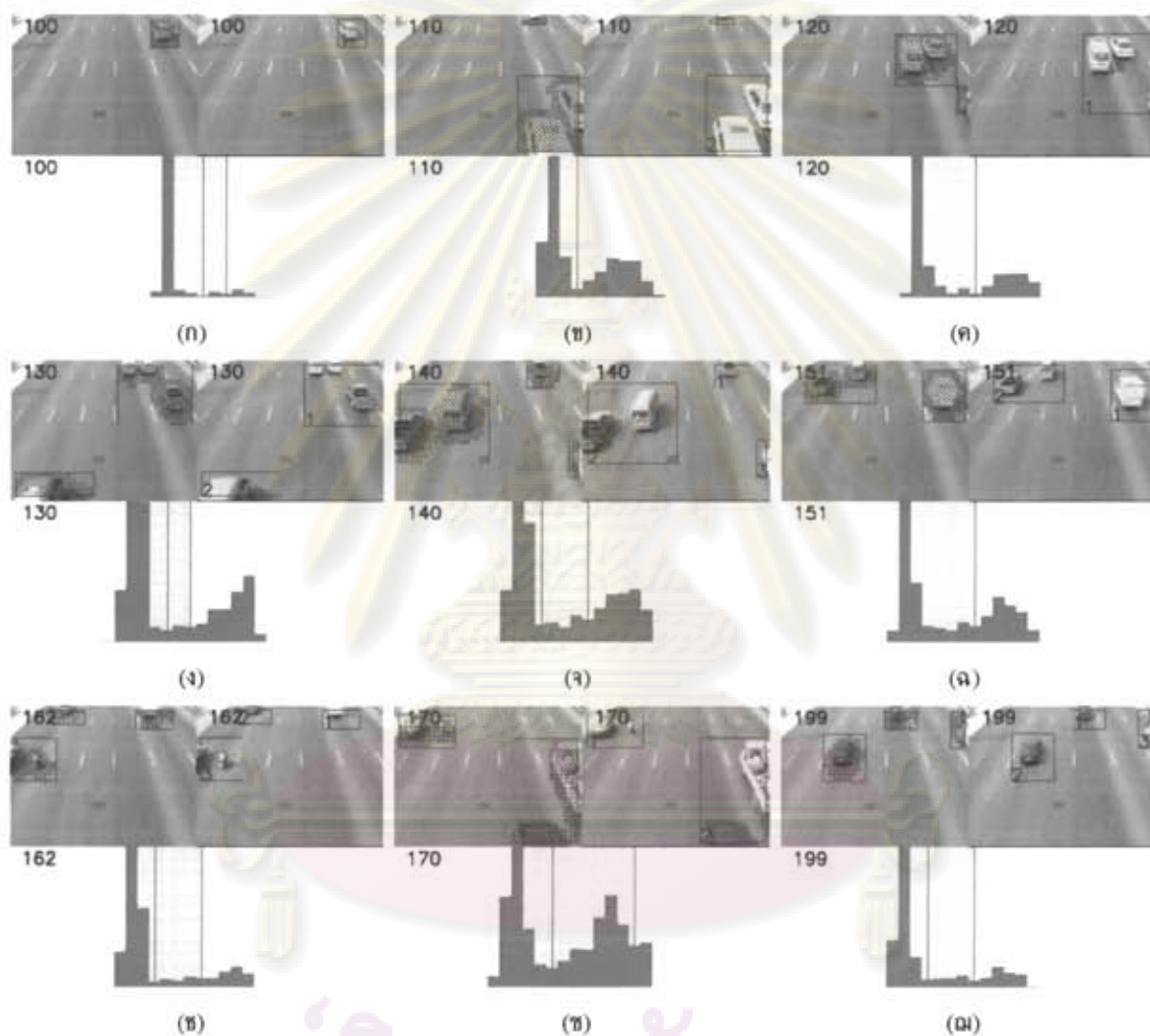
รูปที่ 4.3 ทดสอบด้วยแผนภาพขนาด 16×16 จุดภาพ ณ ลานจอดรถหน้าตึกไฟฟ้าแรงสูง โดยกรอบแดงเป็นข้อมูลที่ถูกรบกวน กากบาทแดงเป็นแผนภาพเคลื่อนที่

จุฬาลงกรณ์มหาวิทยาลัย



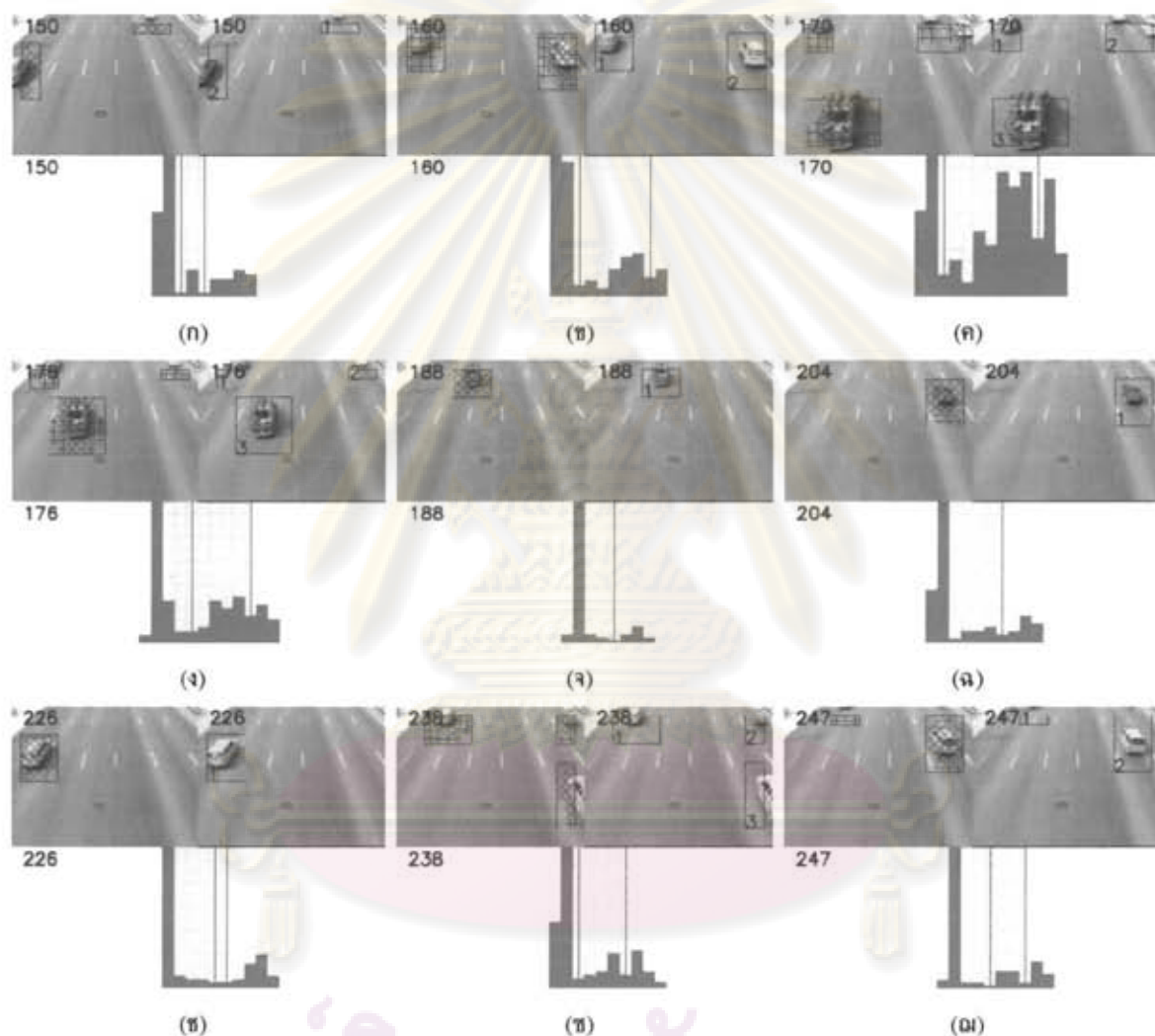
รูปที่ 4.4 ทดสอบด้วยแผงภาพขนาด 4×4 จุดภาพ บนถนนพหลโยธิน โดยกรอบแดงเป็นข้อมูลที่ถูกต้องจากภาพที่แดงเป็นแผงภาพเคลื่อนที่

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



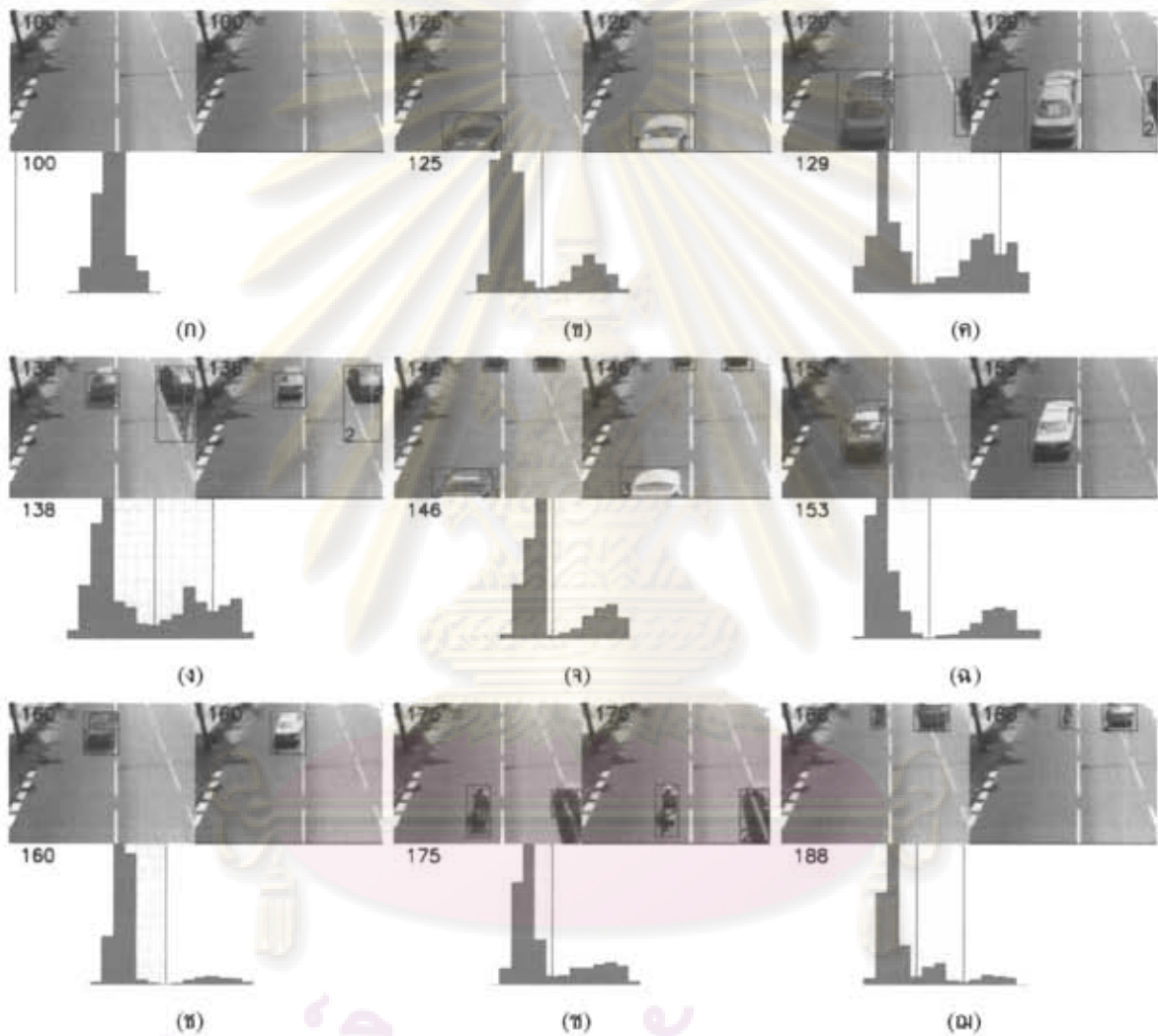
รูปที่ 4.5 ทดสอบด้วยแผงภาพขนาด 8×8 จุดภาพ บนถนนพหลโยธิน โดยกรอบแดงเป็นข้อมูลที่ถูกระบุจากภาพที่แสดงเป็นแผงภาพเคลื่อนที่

ศูนย์วิทยุตำรวจ
จุฬาลงกรณ์มหาวิทยาลัย



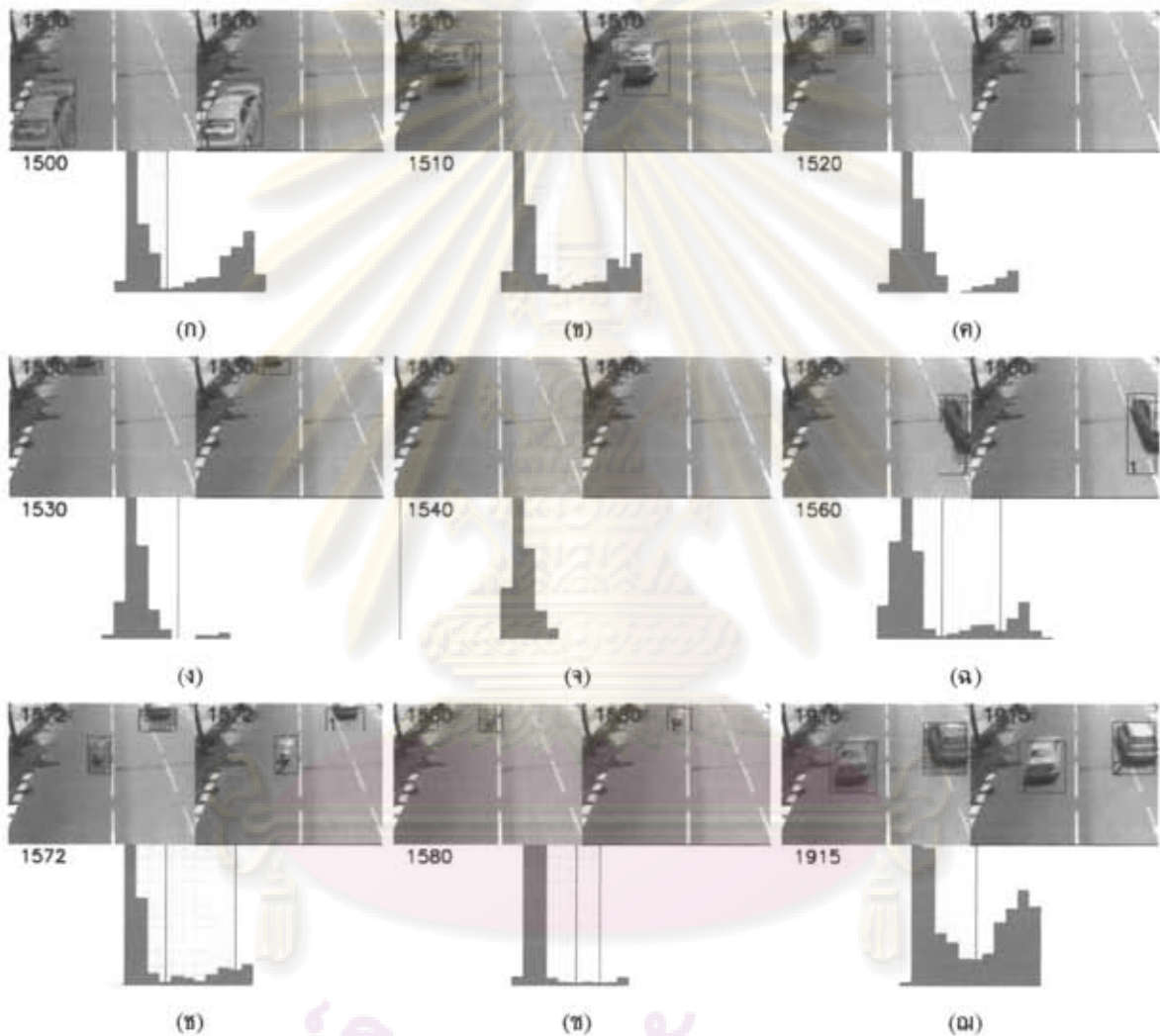
รูปที่ 4.6 ทดสอบด้วยแผงภาพขนาด 16×16 จุดภาพ บนถนนพหลโยธิน โดยกรอบแดงเป็นข้อมูลที่ถูกรัดจํา กากบาทแดงเป็นแผงภาพเคลื่อนที่

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



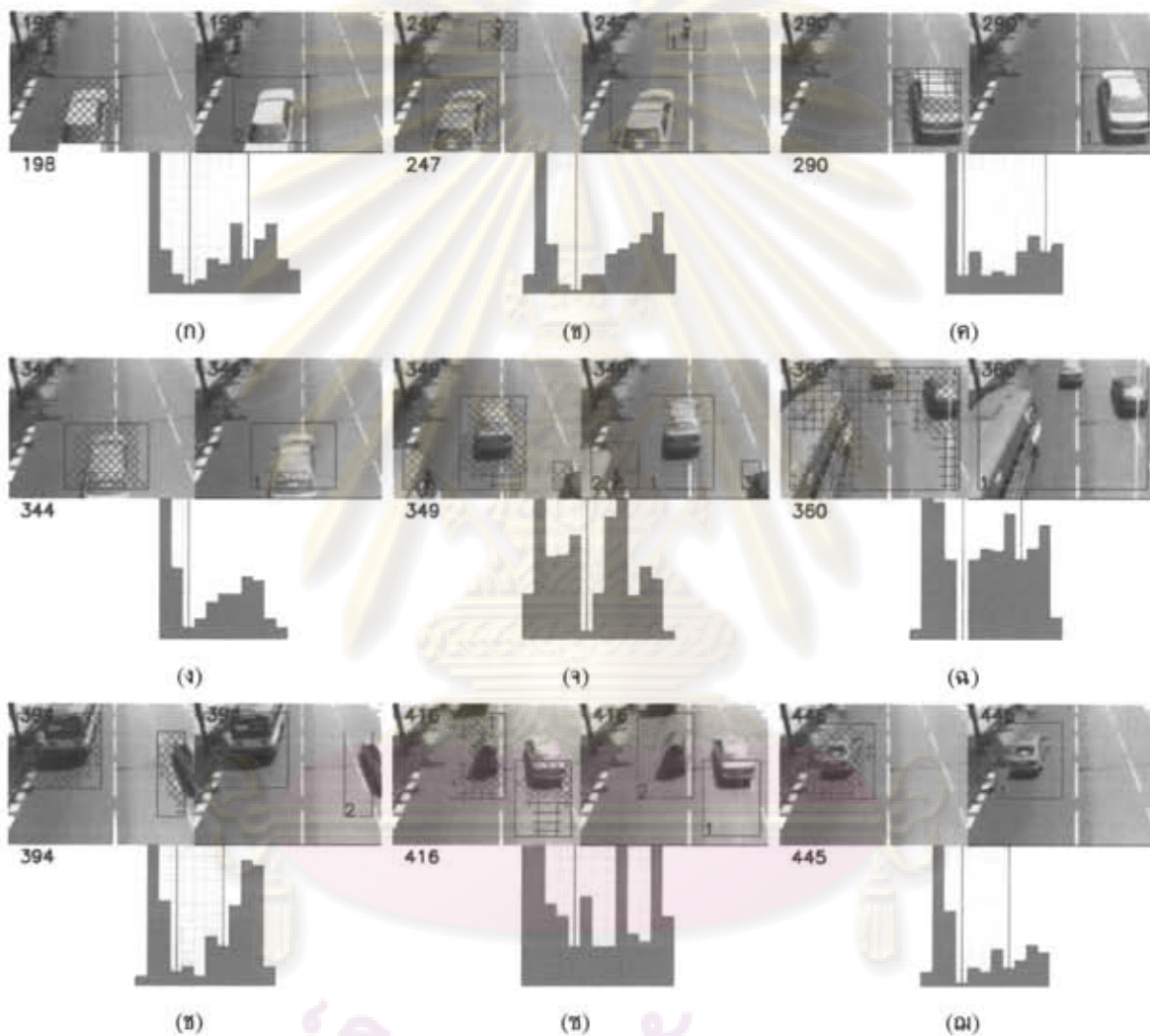
รูปที่ 4.7 ทดสอบด้วยแมงภาพขนาด 4×4 จุดภาพ บนถนนพญาไท โดยกรอบแดงเป็นข้อมูลที่ถูกรัดจํา กากบาทแดงเป็นแมงภาพเคลื่อนที่

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.8 ทดสอบด้วยแมงภาพขนาด 8×8 จุดภาพ บริเวณเคอะวิศกรรมศาสตร์ โดยกรอบแดงเป็นข้อมูลที่ ถูกจดจำ จากบาทแดงเป็นแมงภาพเคลื่อนที่

จุฬาลงกรณ์มหาวิทยาลัย

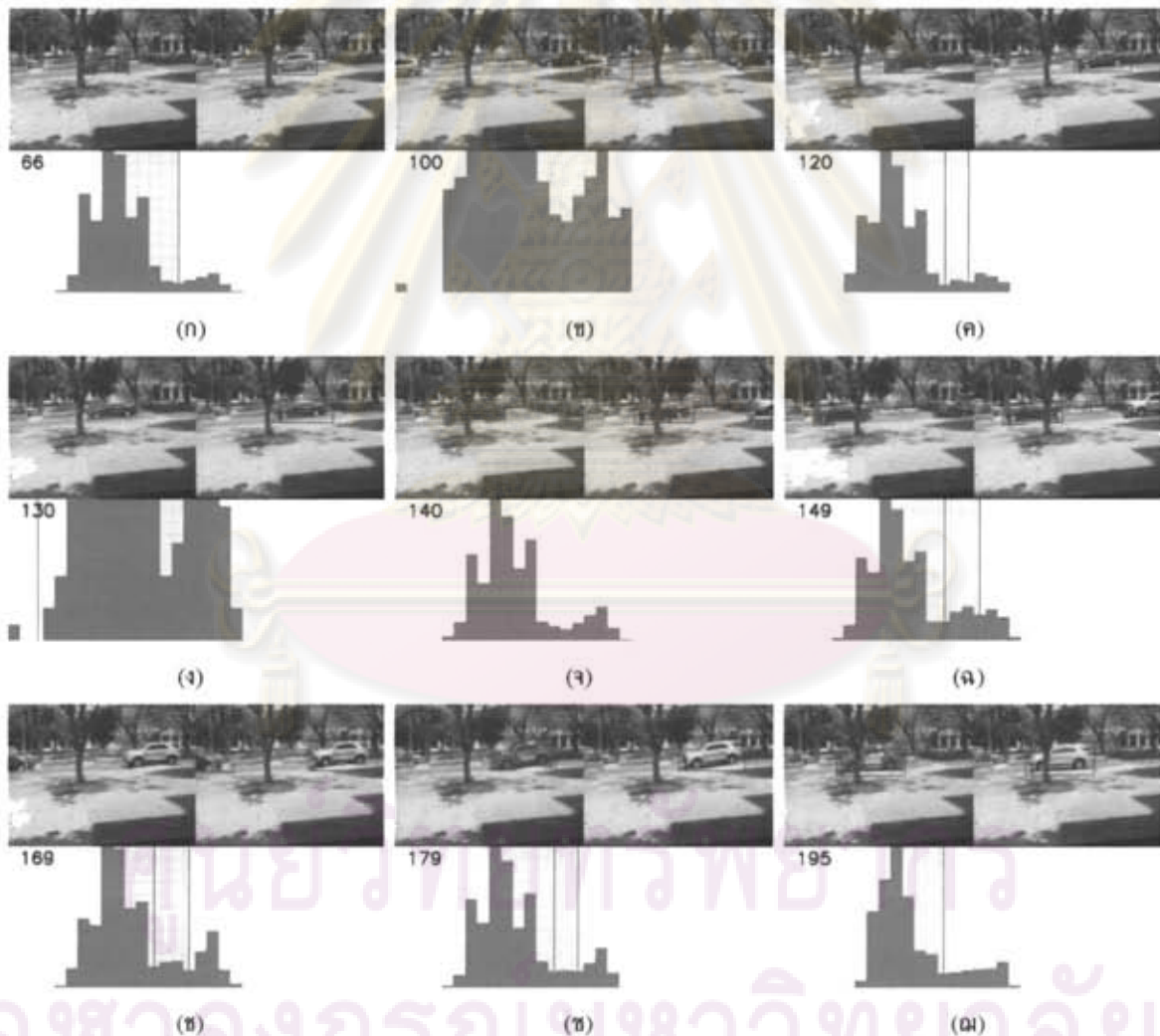


รูปที่ 4.9 ทดสอบด้วยแผงภาพขนาด 16×16 จุดภาพ บริเวณถนนวิศวกรรมศาสตร์ โดยกรอบแดงเป็นข้อมูลที่ถูกต้องดำ กากบาทแดงเป็นแผงภาพเคลื่อนที่

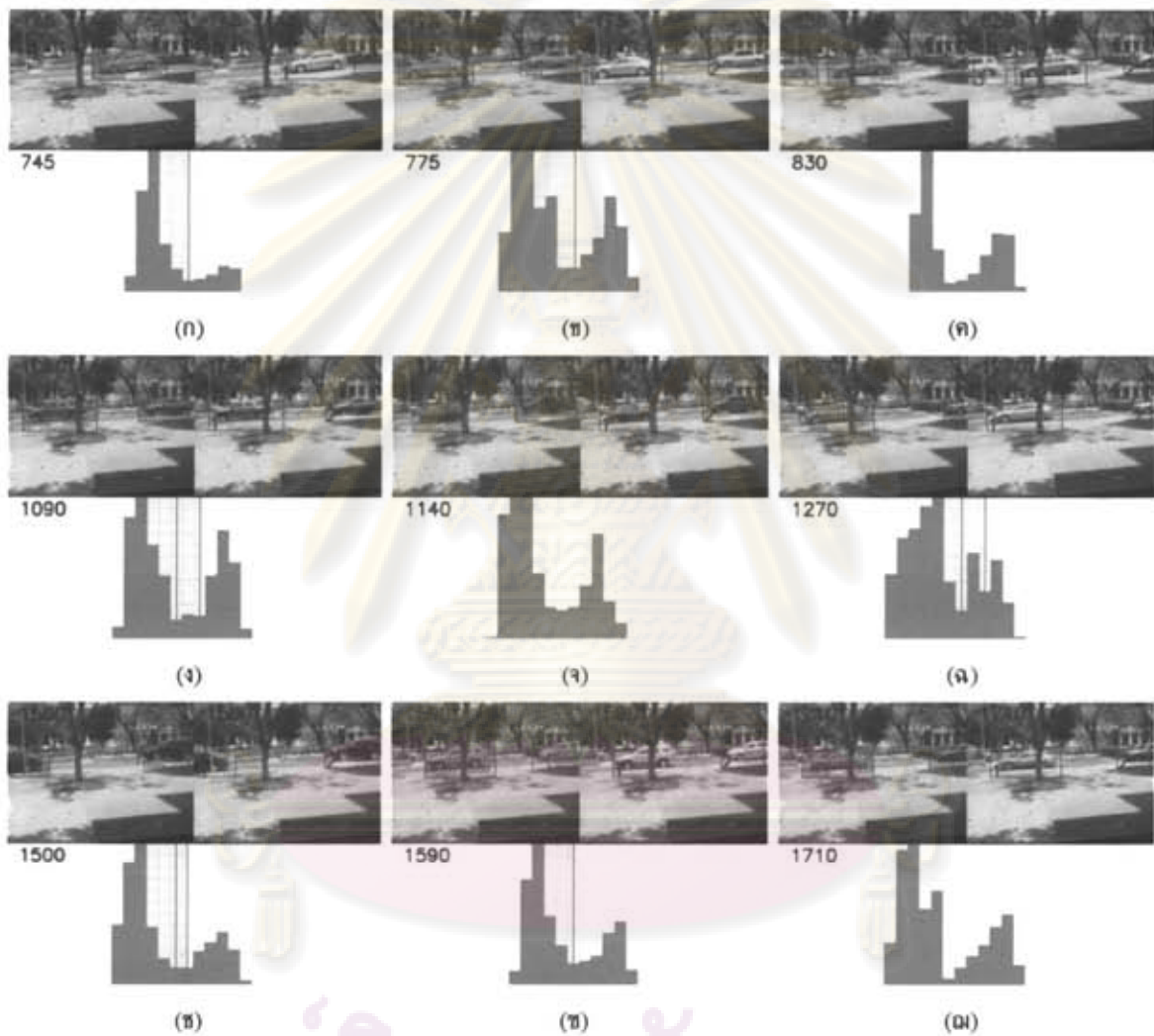
จุฬาลงกรณ์มหาวิทยาลัย

4.1.4 การทดสอบบริเวณคณะวิศวกรรมศาสตร์

การทดสอบนี้มีการรบกวนอันเนื่องมาจากแสงแดดที่เปลี่ยนแปลงที่ละน้อยตลอดการทดลอง กับการเคลื่อนไหวเล็กน้อยของกิ่งไม้ หรือใบไม้ในบริเวณทดสอบ รูปที่ 4.10, 4.11 และ 4.12 เป็นภาพตัวอย่างของผลการทดสอบบริเวณคณะวิศวกรรมศาสตร์ ด้วยแผงภาพขนาด 4×4 , 8×8 และ 16×16 จุดภาพ ตามลำดับ โดย รูปบนของทุกแผงภาพย่อย ของรูปที่ 4.10, 4.11 และ 4.12 แสดงผลลัพธ์ของการตรวจหาวัตถุเคลื่อนที่ และรูปล่างของทุกแผงภาพย่อย ของรูปที่ 4.10, 4.11 และ 4.12 แสดงฮิสโทแกรมของการรวมผลต่างกำลังสอง แกน x แสดงความเข้มแสงกำลังสอง แกน y แสดง ความถี่ของแผงภาพ เราแสดงการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอที่ถูกรบกวนจากแสงแดด กับการเคลื่อนไหวเล็กน้อยของกิ่งไม้ หรือใบไม้ ในมุมมองที่ต่างจากสามการทดสอบแรก

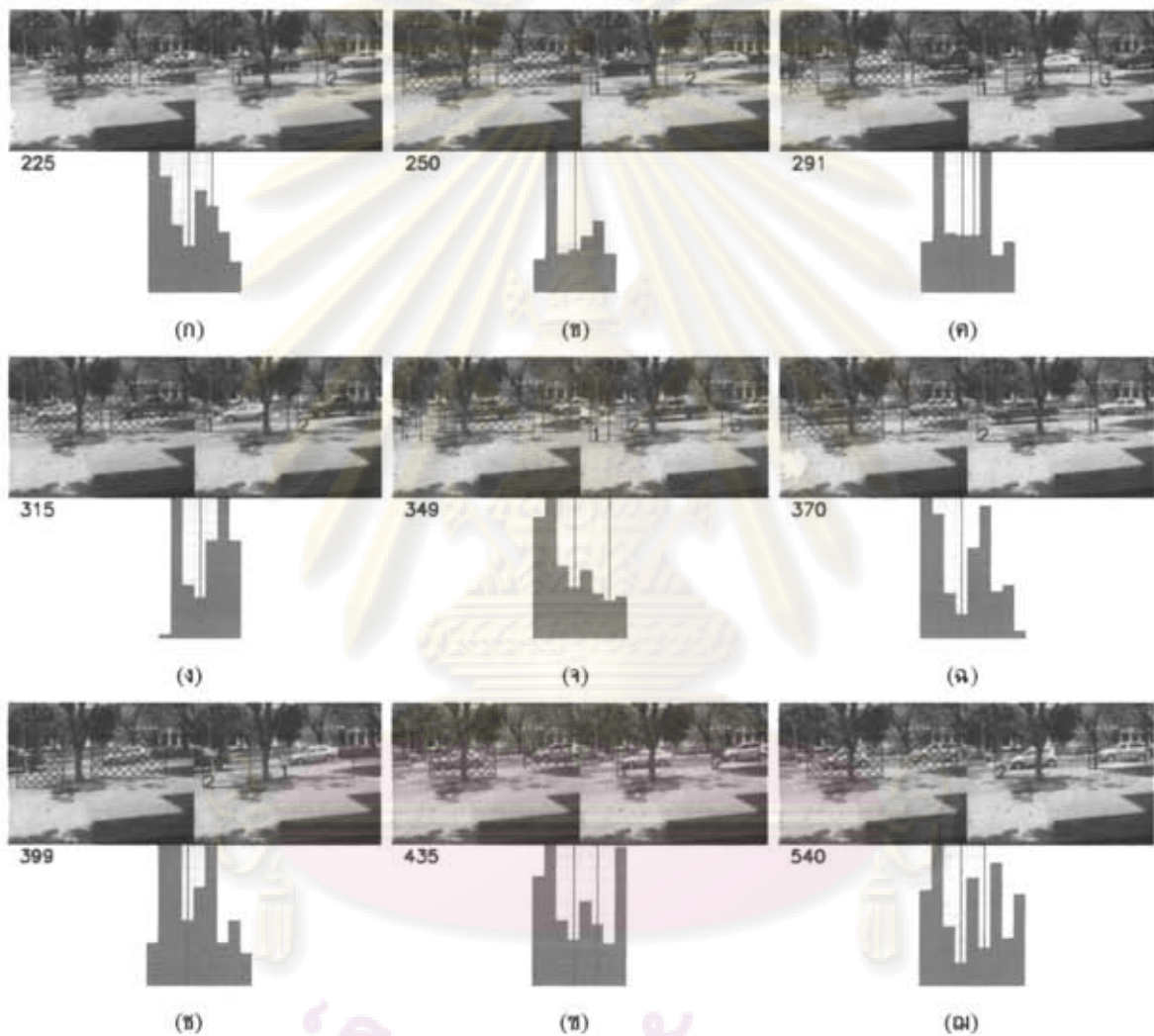


รูปที่ 4.10 ทดสอบด้วยแผงภาพขนาด 4×4 จุดภาพ บริเวณคณะวิศวกรรมศาสตร์ โดยกรอบแดงเป็นข้อมูลที่ถูกจดจำ กากบาทแดงเป็นแผงภาพเคลื่อนที่



รูปที่ 4.11 ทดสอบด้วยแผงภาพขนาด 8×8 จุดภาพ บริเวณคณะวิศวกรรมศาสตร์ โดยกรอบแดงเป็นข้อมูลที่ถูกต้องดำ กากบาทแดงเป็นแผงภาพเคลื่อนที่

จุฬาลงกรณ์มหาวิทยาลัย

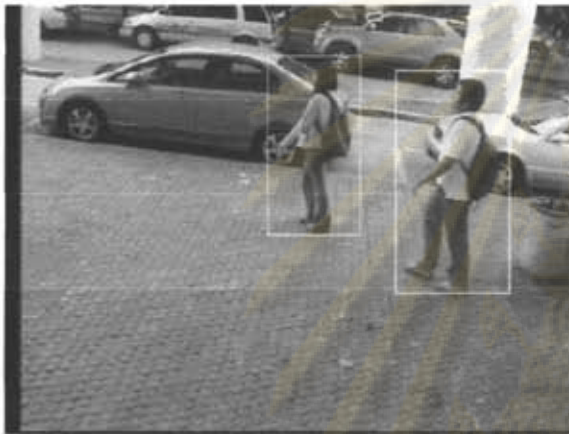


รูปที่ 4.12 ทดสอบด้วยแมงภาพขนาด 16×16 จุดภาพ บริเวณคณะวิศวกรรมศาสตร์ โดยกรอบแดงเป็นข้อมูลที่ถูกจดจำ กากบาทแดงเป็นแมงภาพเคลื่อนที่

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4.2 การทดสอบเพื่อลดขนาดไฟล์ผลลัพธ์

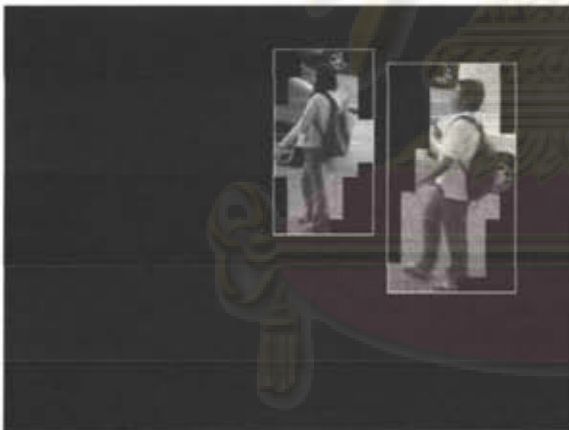
การเปรียบเทียบขนาดไฟล์ภาพที่ไม่ได้ตัดภาพพื้นหลังออกปกติกับ ขนาดไฟล์ภาพที่ตัดภาพพื้นหลังออกแล้วแทนด้วยศูนย์ โดยใช้การตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอที่นำเสนอ ในการแยกวัตถุเคลื่อนที่ออกจากภาพพื้นหลัง เราทดสอบกับภาพชนิด PNG กับ JPG ซึ่งเป็นการบีบอัดภาพทั่วไปที่ไลบรารีโอเพ่นซีวีรองรับ



(ก) ภาพเริ่มต้น, JPG 39.5 KB, PNG 144.0 KB



(ข) แผงภาพ 4 × 4, JPG 9.7 KB, PNG 18.7 KB



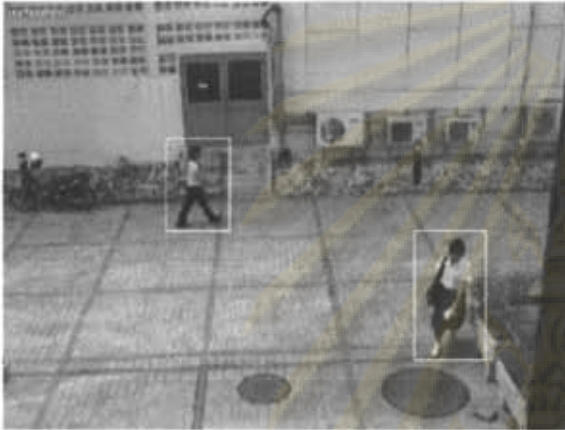
(ค) แผงภาพ 8 × 8, JPG 9.5 KB, PNG 22.8 KB



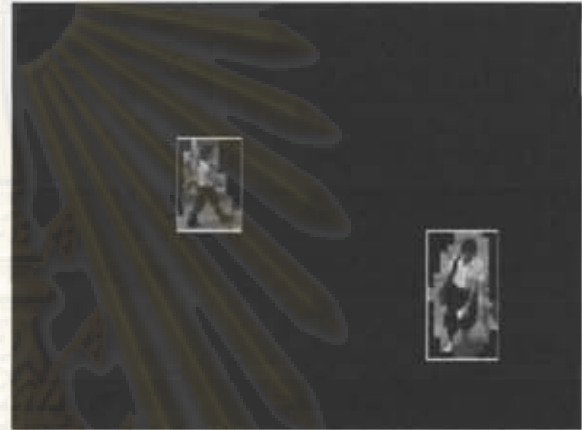
(ง) แผงภาพ 16 × 16, JPG 14.1 KB, PNG 43.0 KB

รูปที่ 4.13 ผลของการลดขนาดไฟล์ด้วยเทคนิคที่นำเสนอ ในเหตุการณ์ที่ 1

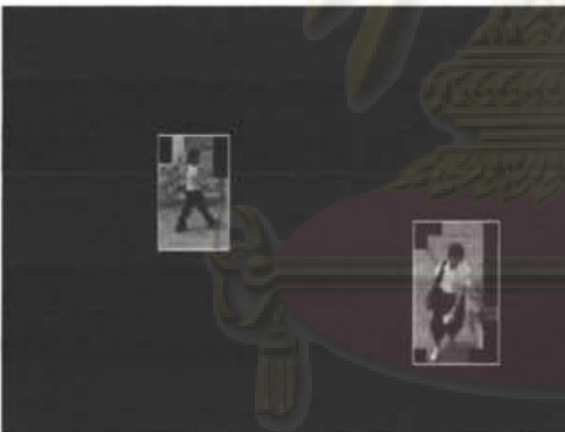
จากผลการทดลองในรูปที่ 4.13 กับผลการทดลองในรูปที่ 4.14 เราสามารถลดขนาดของไฟล์ภาพชนิด JPG และ PNG โดยลดขนาดไฟล์สำหรับส่งข้อมูลได้ประมาณ 5 และ 10 เท่า ตามลำดับ เพราะการแทนบริเวณที่ไม่เคลื่อนที่ด้วยศูนย์ทำให้บีบอัดข้อมูลได้มากขึ้น



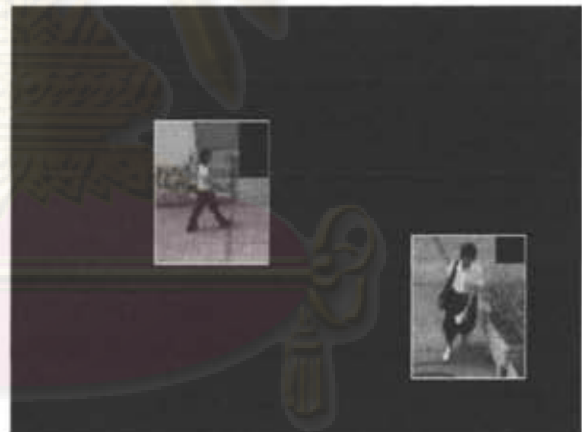
(ก) ภาพเริ่มต้น, JPG 31.9 KB, PNG 124.4 KB



(ข) แผงภาพ 4 × 4, JPG 5.8 KB, PNG 8.6 KB



(ค) แผงภาพ 8 × 8, JPG 5.8 KB, PNG 11.6 KB



(ง) แผงภาพ 16 × 16, JPG 7.5 KB, PNG 18.9 KB

ศูนย์วิทยทรัพยากร

รูปที่ 4.14 ผลของการลดขนาดไฟล์ด้วยเทคนิคที่นำเสนอ ในเหตุการณ์ที่ 2

จุฬาลงกรณ์มหาวิทยาลัย

4.3 การทดสอบด้านเวลาที่ใช้ในการประมวลผล

การทดสอบด้านเวลาที่ใช้ในการประมวลผล เร่วัดระยะเวลาเฉลี่ยในการประมวลผลบนเครื่องคอมพิวเตอร์กับระบบฝังตัว ดังนี้

4.3.1 การทดสอบบนเครื่องคอมพิวเตอร์

การทดสอบเวลาที่ใช้ในการประมวลผลบนเครื่องคอมพิวเตอร์ Lenovo X61 โดยมีคุณสมบัติดังตารางที่ 4.4 และมีผลการวัดระยะเวลาที่ใช้ในการประมวลผลดังตารางที่ 4.5

ตารางที่ 4.4 คุณสมบัติของ Lenovo X61

รายการ	รายละเอียด
หน่วยประมวลผล	Intel Core 2 Duo processor T7100 (1.8 GHz, 3 MB, 800 MHz)
หน่วยความจำหลัก	DDR2 SDRAM 1.5 GB/667 MB

ตารางที่ 4.5 ระยะเวลาที่ใช้ในการประมวลผลบน Lenovo X61

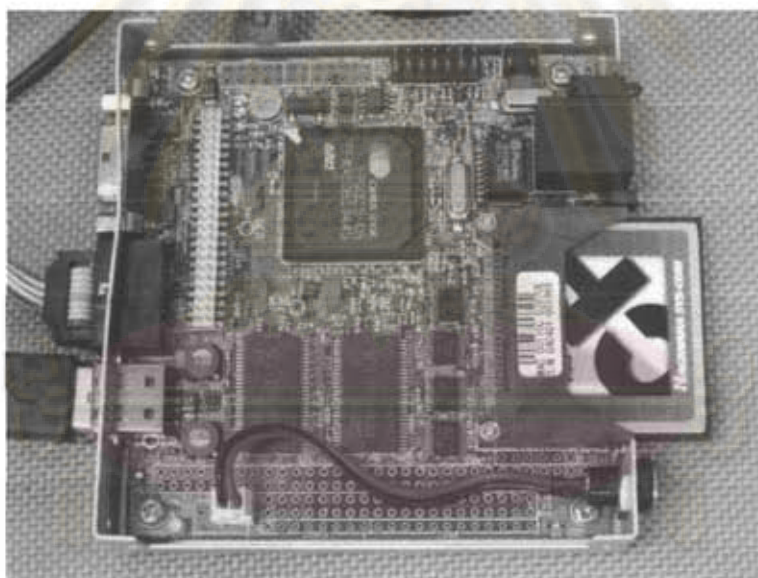
แผงภาพขนาด 4 × 4	แผงภาพขนาด 8 × 8	แผงภาพขนาด 16 × 16
5.194 ms	4.475 ms	4.297 ms
5.116 ms	4.505 ms	4.318 ms
5.152 ms	4.507 ms	4.296 ms
5.150 ms	4.705 ms	4.317 ms
5.131 ms	4.516 ms	4.322 ms
5.226 ms	4.924 ms	4.317 ms
5.151 ms	4.519 ms	4.320 ms
5.129 ms	4.497 ms	4.303 ms
5.197 ms	4.642 ms	4.314 ms
5.149 ms	4.621 ms	4.305 ms
เฉลี่ย 5.159ms	เฉลี่ย 4.591ms	เฉลี่ย 4.310ms

ในขั้นตอนการทดสอบเราเปิดโปรแกรม apache2 เพื่อจำลองการทำงานเสมือนการใช้งานจริงมากที่สุด ดังนั้นขณะทดสอบเราสามารถรับภาพจากผลลัพธ์ผ่านเว็บเบราว์เซอร์ ดังแสดงในรูปที่ 3.3 การทดสอบเรา

ทดสอบที่แผงภาพขนาดต่างกัน 3 ขนาด แผงภาพเล็กใช้เวลานานที่สุดเพราะการคำนวณคอนทัวร์ จำนวนข้อมูลของแผงภาพขนาดเล็กมีจำนวนข้อมูลมากกว่า จำนวนข้อมูลของแผงภาพขนาดใหญ่ จากค่าเฉลี่ยในผลการทดลองในตารางที่ 4.5 พบว่าขนาดแผงภาพ 4×4 8×8 และ 16×16 จุดภาพ สามารถรองรับอัตราข้อมูลเข้ามากที่สุด 193.83 217.81 และ 232.01 เฟรมต่อวินาที ตามลำดับ

4.3.2 การทดสอบบนระบบฝังตัว

การทดสอบเวลาที่ใช้ในการประมวลผลบนระบบฝังตัว ในงานวิจัยนี้เลือก Armadillo-9 ดังรูปที่ 4.15 เป็นระบบฝังตัว เราสามารถติดตั้งระบบปฏิบัติการ GNU/Linux Emdebian ในระบบฝังตัวนี้ได้ และมีพอร์ตต่อกล้องเว็บแคมผ่านทางพอร์ต USB ดังรูปที่ 4.16 และ Armadillo-9 มีพอร์ตต่อเครือข่ายสำหรับรับส่งข้อมูลผ่านเครือข่ายได้ Armadillo-9 มีคุณสมบัติดังตารางที่ 4.6 และมีผลการวัดระยะเวลาที่ใช้ในการประมวลผลตามเทคนิคที่นำเสนอดังตารางที่ 4.7



รูปที่ 4.15 บอร์ด Armadillo-9 ที่ใช้เป็นระบบฝังตัว

การทดสอบใช้ขั้นตอนเดียวกับในการทดสอบในเครื่องคอมพิวเตอร์โดย เราเปิดโปรแกรม apache2 เป็นการทดสอบเสมือนการใช้งานจริงมากที่สุด ขณะทดสอบสามารถรับภาพจากผลลัพธ์ผ่านเว็บเบราว์เซอร์ ดังแสดงในรูปที่ 3.3 การทดสอบเราทดสอบที่แผงภาพขนาดต่างกัน 3 ขนาด แผงภาพเล็กใช้เวลานานที่สุดเพราะการคำนวณคอนทัวร์ จำนวนข้อมูลของแผงภาพขนาดเล็กมีจำนวนข้อมูลมากกว่า จำนวนข้อมูลของแผงภาพขนาดใหญ่ จากค่าเฉลี่ยในผลการทดลองในตารางที่ 4.7 พบว่าขนาดแผงภาพ 4×4 , 8×8 และ 16×16 จุดภาพ สามารถรองรับอัตราข้อมูลเข้า 4.26, 6.00 และ 6.17 เฟรมต่อวินาที ตามลำดับ



รูปที่ 4.16 บอร์ด Armadillo-9 ที่ใช้เป็นระบบฝังตัว

ตารางที่ 4.6 คุณลักษณะของ Armadillo-9

รายการ	รายละเอียด
หน่วยประมวลผล	EP9315 (Cirrus Logic)
แกนหน่วยประมวลผล	ARM920T CPU core ARM9TDMI 16kByte instruction cache 16kByte data cache
ความถี่สัญญาณนาฬิกากระบบ	ความถี่สัญญาณนาฬิกา หน่วยประมวลผล 200 MHz ความถี่สัญญาณของบัส 100 MHz 16kByte instruction cache 16kByte data cache
หน่วยความจำ	SDRAM 64MB FLASH 8MB
USB	2.0 Full Speed (12Mbps) 1 channel. Type A connector

ตารางที่ 4.7 ระยะเวลาที่ใช้ในการประมวลผลบน Armadillo-9

แผงภาพขนาด 4 × 4	แผงภาพขนาด 8 × 8	แผงภาพขนาด 16 × 16
237.707 ms	166.841 ms	161.909 ms
234.263 ms	166.424 ms	161.788 ms
234.777 ms	166.061 ms	162.135 ms
235.787 ms	165.414 ms	161.760 ms
232.285 ms	166.262 ms	162.356 ms
235.296 ms	165.532 ms	162.222 ms
233.888 ms	166.281 ms	161.941 ms
233.101 ms	168.238 ms	161.922 ms
232.742 ms	167.512 ms	161.591 ms
232.905 ms	166.832 ms	161.732 ms
เฉลี่ย 234.275 ms	เฉลี่ย 166.540 ms	เฉลี่ย 161.936 ms

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

ข้อสรุปและข้อเสนอแนะ

5.1 ข้อสรุป

ระบบฝังตัวที่ถูกนำเสนอครั้งแรก คือ DE2 ซึ่งเป็นอุปกรณ์ที่มีความสามารถสูง และทำงานได้เร็วมาก แต่ DE2 มีราคาสูง พร้อมกับโปรแกรมพัฒนาที่มีราคาแพง จึงเป็นอุปสรรคต่อประโยชน์ในวงกว้าง ดังนั้น วิทยานิพนธ์นี้จึงเลือก Armadillo-9 เป็นระบบฝังตัว ซึ่งมีราคาถูกกว่า และเครื่องมือที่ใช้ในการพัฒนาเป็นระบบเปิดจึงไม่ต้องเสียค่าใช้จ่ายในส่วนนี้ แต่ Armadillo-9 ทำงานช้ากว่า DE2 ทำให้วิทยานิพนธ์นี้ต้องมีขั้นตอนวิธีที่ทำงานเร็ว มีประสิทธิภาพ และสามารถทำงานตามเวลาจริงได้ในระดับที่น่าพอใจ เนื่องจากเราสามารถนำผลลัพธ์ของ Armadillo-9 มาแสดงในเว็บเบราว์เซอร์ และตั้งค่าต่าง ๆ ของโปรแกรมตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอที่ส่งผ่านเครือข่ายได้ เป็นวิธีการที่สะดวกกว่าการแสดงผลผ่าน VGA out เมื่อเราลดอุปกรณ์ในส่วนนี้ ทำให้เราลดต้นทุนและสามารถนำไปใช้งานในวงกว้างได้

การตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอที่ส่งโดยใช้ระบบฝังตัว เป็นการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอที่ส่งที่ต้องคำนวณด้วยเลขคณิตจุดตรึง และต้องใช้เวลาประมวลผลน้อย โดยยังคงคุณภาพของผลลัพธ์ที่ได้ เทคนิคที่สำคัญคือการแบ่งภาพออกเป็นแผงภาพย่อย ทำให้ลดตัวรบกวนในภาพได้ แผงภาพขนาดใหญ่เหมาะกับวัตถุเคลื่อนที่ที่มีขนาดใหญ่ หรือ การเคลื่อนที่ของวัตถุใกล้หน้ากล้อง แผงภาพขนาดเล็กเหมาะกับวัตถุเคลื่อนที่ที่มีขนาดเล็ก หรือ การเคลื่อนที่ของวัตถุในระยะไกล ส่วนแผงภาพขนาดใหญ่จะตอบสนองต่อการเคลื่อนที่ช้ากว่าแผงภาพขนาดเล็ก แต่แผงภาพขนาดใหญ่ทนต่อการรบกวนได้ดีกว่า การปรับค่าขีดเริ่มเปลี่ยนอัตโนมัติ ช่วยรองรับทุกสถานะการณ์ต่าง ๆ ได้มากขึ้น และสามารถใช้งานได้กับกล้องที่ไม่สามารถตัดระบบปรับแสงขาวอัตโนมัติ หรือกล้องเว็บแคมสำหรับผู้บริโภค ซึ่งมีราคาถูก หาซื้อได้ง่าย เราสามารถกำจัดแผงภาพที่ไม่สอดคล้องกับการเคลื่อนที่จริง ณ เฟรมหนึ่ง ๆ ได้ และด้วยเทคนิคนี้เราสามารถตรวจจับการเคลื่อนที่ที่ชัดเจนเท่านั้น เราสามารถทำงานบนระบบฝังตัวอย่าง Armadillo-9 ที่เป็นเว็บเสิร์ฟเวอร์ได้ เราจึงส่งไฟล์ภาพผ่านเครือข่ายอินเทอร์เน็ตได้ และทำงานด้วยความเร็วถึง 7 เฟรมต่อวินาที เป็นอัตราเฟรมสูงสุดที่สามารถรองรับได้ โดยภาพมีขนาด 320×240 จุดภาพ ในวิทยานิพนธ์นี้สามารถลดขนาดข้อมูลที่ต้องส่งได้ด้วยการแทนบริเวณพื้นหลังในภาพด้วยศูนย์ ทำให้เราลดขนาดไฟล์ชนิด PNG ได้ประมาณ 10 เท่า และลดขนาดไฟล์ชนิด JPG ได้ประมาณ 5 เท่า ขึ้นกับขนาดของวัตถุในลำดับภาพที่ตรวจพบ เราเพิ่มการจดจำวัตถุเคลื่อนที่ในลำดับภาพวิดีโอที่ส่ง เพื่อรองรับวัตถุเคลื่อนที่ในลำดับภาพที่หยุดการเคลื่อนไหวในบริเวณที่ปรากฏในเฟรม ทำให้เราได้ผลที่ใกล้เคียงกับวิธีการลบภาพพื้นหลัง

5.2 ข้อเสนอแนะ

การทำให้โปรแกรมตรวจหาวัตถุเคลื่อนที่ในลำดับภาพโดยใช้ระบบฝังตัวทำงานได้มีประสิทธิภาพมากยิ่งขึ้น และสามารถนำไปใช้งานได้จริง จึงได้เสนอบทสรุปที่ควรปรับปรุงและข้อเสนอแนะสำหรับการพัฒนาโปรแกรมในอนาคต ดังต่อไปนี้

1. เราสามารถลดเวลาในการประมวลผลลงได้ ด้วยการลดขั้นตอนที่ทำงานซ้ำกัน เพราะในวิทยานิพนธ์นี้ใช้ฟังก์ชันที่มีอยู่แล้วในโอเพ่นซีวี จึงมีโอกาสมที่จะมีการวนรอบคำนวณซ้ำที่เดิมได้ เช่นการแปลงภาพ ณ เฟรมหนึ่ง ๆ กับการคำนวณการรวมผลต่างระหว่างภาพกำลังสอง ซึ่งเป็นฟังก์ชันของโอเพ่นซีวี กับฟังก์ชันที่สร้างขึ้นเอง ตามลำดับ ตามหลักการแล้วสองส่วนนี้สามารถวนรอบคำนวณพร้อมกันได้
2. ในระบบฝังตัวส่วนมากใช้ v4l เป็นโมดูลรับภาพ การโปรแกรมโดยใช้จาก v4l จะช่วยให้โปรแกรมสามารถใช้ในระบบฝังตัวที่มีขนาดเล็ก ซึ่งมีราคาถูกกว่าระบบฝังตัวที่สามารถติดตั้งไลบรารีโอเพ่นซีวีได้ และระบบฝังตัวที่ติดตั้งไลบรารีโอเพ่นซีวีได้จะเป็นระบบฝังตัวที่มีขนาดปานกลางถึงขนาดใหญ่ การใช้ไลบรารี v4l สามารถทำให้เราใช้กล้องเว็บแคมที่ส่งภาพเป็น YUV ทำให้เราลดขั้นตอนการแปลงภาพจาก RGB ของโอเพ่นซีวี เป็นความเข้มแสงได้
3. ควรเพิ่มประสิทธิภาพของผลลัพธ์ด้วยการติดตามวัตถุเคลื่อนที่ โดยการจำคุณลักษณะของวัตถุเคลื่อนที่หนึ่ง ๆ เพื่อเพิ่มความน่าเชื่อถือยิ่งขึ้น และสามารถแก้ปัญหาการซ้อนทับของภาพได้ในบางกรณี
4. สามารถนำไปประยุกต์กับการนับวัตถุเคลื่อนที่ เช่น การนับรถบนท้องถนน หรือ เครื่องรายงานสภาพจราจรเป็นอัตราการเคลื่อนที่ของรถยนต์ในบริเวณหนึ่งที่มีหน่วยเป็น คันต่อนาที หรือ คันต่อชั่วโมง
5. สามารถนำไปประยุกต์กับงานเฝ้าระวังอัตโนมัติได้ โดยสามารถแจ้งเหตุได้จากจุดที่ถ่ายภาพ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

- [1] Sven Fleck, Sven Lanwer and Wolfgang Straßer. A smart camera approach to real-time tracking. 13th European Signal Processing Conference (EUSIPCO 2005) (September 2005).
- [2] Xiaoye Lu and Roberto Manduchi. Fast image motion computation on an embedded computer. Computer Vision and Pattern Recognition (CVPR '06) (June 2006): 120.
- [3] ชัยพร ชัชวาลกิจกุล และ สุรีย์ พุ่มรินทร์. การตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวีดิทัศน์. การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 30 (EECON-30) เล่มที่ 2 (ตุลาคม 2550): 677-680.
- [4] Arrigo Benedetti and Pietro Perona. Real-time 2-d feature detection on a reconfigurable computer. Computer Vision and Pattern Recognition (CVPR '98) (June 1998): 586.
- [5] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. Computer Vision and Pattern Recognition (CVPR '01). (June 2001):747.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

บทความที่ได้รับการตีพิมพ์
ในการประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 30



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

การตรวจหาการเคลื่อนที่ของวัตถุในลำดับภาพวิดีโอ

Detection of moving objects in video sequence

ชัยพร ชัยवालกิจกุล และ สุรีย์ พุ่มรินทร์

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ถนนพญาไท ปทุมวัน กรุงเทพฯ 10330 โทร (02) 218-6487 โทรสาร (02) 251-8991

E-mail: chaiyaphorn@digital.ee.eng.chula.ac.th, Suree.P@chula.ac.th

บทคัดย่อ

บทความนี้ กล่าวถึงการตรวจหาการเคลื่อนที่ของวัตถุในลำดับภาพวิดีโอที่มีฉากหลังคงที่โดยไม่ต้องเก็บภาพฉากหลัง (background image) และเน้นการคำนวณชนิดจุดคงที่ (fixed-point) ทั้งหมด การคำนวณเริ่มจากภาวะอนุสถานะของแผงภาพ (image patch) จากค่าลำดับชั้น (rank) ของเมทริกซ์ในสมการคำนวณความเร็วของวิธีการไหลเชิงแสง (optical flow) ด้วยวิธีกำลังสองน้อยที่สุด (the least squares approach) กับวิธีถดถอยกำลังสองทั้งหมด (the total least squares regression) วิธีการคำนวณค่าลำดับชั้นที่รวดเร็วเป็นส่วนสำคัญของงานวิจัย ผลลัพธ์ของการระบุสถานะของแผงภาพต้องแบ่งกลุ่มภาพตามวัตถุเคลื่อนที่เพื่อความถูกต้อง ในส่วนนี้เราเสนอการปรับปรุงการขยายบริเวณ (region growing) โดยกำหนดให้บริเวณเคลื่อนที่ของวัตถุเป็นบล็อบบlob) ทรงสี่เหลี่ยมที่ครอบวัตถุได้พอดี ซึ่งบล็อบบริเวณจากแผงภาพที่มีการเคลื่อนที่ที่ใกล้เคียงกันมีระยะห่างกันน้อยกว่าค่าขีดเริ่มเปลี่ยน (threshold) ด้วยวิธีการขยายบริเวณในงานวิจัยนี้ทำให้การตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอมีความทนทานต่อสถานะการเคลื่อนที่ต่าง ๆ ได้มากขึ้น โดยผลลัพธ์ที่ได้เป็นบล็อบบริเวณพร้อมกับตำแหน่งของวัตถุเคลื่อนที่ในลำดับภาพของแต่ละเฟรม

คำสำคัญ: การติดตามภาพ, การไหลเชิงแสง, ค่าลำดับชั้น, การขยายบริเวณ

Abstract

This paper presents a detection of moving objects in video sequence that uses fixed-point calculation. We estimate the rank of matrix with noisy data and decide whether the image patch is moving from the rank of matrix. This rank is calculated from optical flow equations using the least squares approach and the total least squares regression. The fast calculation of this matrix's rank is considered to be one of the most critical parts. Results from image patch definition will be processed by region growing using threshold method. The region growing in this paper increases a robustness for rank estimation. Finally, we can identify the blobs and the moving objects.

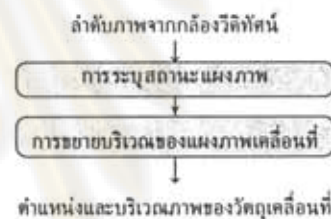
Keywords: image tracking, optical flow, rank, region growing

1 บทนำ

โดยทั่วไปการทำงานบางส่วนในระบบเฝ้าระวังอัตโนมัติ จะมีคอมพิวเตอร์แม่ข่าย (server computer) รับผิดชอบแล้วคำนวณหาวัตถุเคลื่อนที่ในภาพก่อนที่จะนำข้อมูลส่งไปวิเคราะห์ยังส่วนอื่น วิธีนี้คอมพิวเตอร์แม่ข่ายและระบบเครือข่ายต้องทำงานตลอดเวลา และเสียเวลาส่วนมากไปกับเฟรมที่ไม่มีวัตถุเคลื่อนที่ เราสามารถลดต้นทุนการคำนวณและลดปริมาณข้อมูลที่ต้องส่งจากจุดถ่ายภาพไปยังคอมพิวเตอร์แม่ข่ายได้ โดยการคำนวณหาวัตถุเคลื่อนที่ในลำดับภาพ แล้วส่งข้อมูลตำแหน่งบล็อบบริเวณของวัตถุกับบล็อบบริเวณของภาพวัตถุ

เคลื่อนที่ ซึ่งเป็นข้อมูลสำหรับการวิเคราะห์ในระบบเฝ้าระวังอัตโนมัติได้

การไหลเชิงแสงเป็นการหาความต่างระหว่างสองภาพที่เหมือนกันโดยมีลำดับติดกัน ทำให้การเปลี่ยนแปลงของแสงที่ละน้อยไม่มีผลกระทบต่อหารวัตถุเคลื่อนที่ในลำดับภาพวิดีโอ และมีข้อมูลขาออกเมื่อมีการเคลื่อนที่เกิดขึ้น ณ จุดถ่ายภาพวิดีโอ การตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอเริ่มจากการแบ่งภาพออกเป็นแผงขนาด 4×4 จุดภาพ แล้วระบุสถานะของแผงภาพแต่ละแผง ข้อมูลแผงภาพที่ถูกระบุสถานะจะถูกนำไปวิเคราะห์ยังส่วนการขยายบริเวณได้ผลลัพธ์เป็นบล็อบบริเวณที่สอดคล้องกับการเคลื่อนที่จริงของวัตถุในลำดับภาพวิดีโอ โดยบล็อบบริเวณเป็นข้อมูลตำแหน่งกับบริเวณของภาพวัตถุเคลื่อนที่ ขั้นตอนวิธีที่กล่าวในบทความนี้แสดงดังรูปที่ 1



รูปที่ 1: ขั้นตอนวิธีการระบุสถานะของแผงภาพจากค่าลำดับชั้นของเมทริกซ์

2 การระบุสถานะแผงภาพ

การคำนวณหาวัตถุเคลื่อนที่ในลำดับภาพ เป็นการวิเคราะห์ความเข้มแสงของภาพที่มีความต่อเนื่องกันและภายในลำดับภาพที่มีฉากหลังเดียวกัน โดยวัตถุเคลื่อนที่แต่ละลำดับจะมีความแตกต่างกันเพียงเล็กน้อย เป็นการตรวจหาการเคลื่อนที่โดยวิธีการไหลเชิงแสง [1] ดังสมการที่ (1) เมื่อ $f(x, y, t)$ แทนฟังก์ชันความเข้มแสงที่แปรตามตำแหน่งและเวลา

$$\frac{\partial f}{\partial x} \frac{dx}{dt} + \frac{\partial f}{\partial y} \frac{dy}{dt} + \frac{\partial f}{\partial t} = 0 \quad (1)$$

สมการที่ (1) จัดให้อยู่ในรูปของเมทริกซ์ เพื่อความสะดวกต่อการพิจารณา ดังสมการที่ (2)

$$\nabla I^T v + I_t = 0 \quad (2)$$

เมื่อ $\nabla I = (I_x, I_y)^T$; $I_x = \frac{\partial f}{\partial x}$; $I_y = \frac{\partial f}{\partial y}$; $I_t = \frac{\partial f}{\partial t}$ และ $v = \left(\frac{dx}{dt}, \frac{dy}{dt} \right)^T$ ในสมการที่ (2) เป็นความเร็ว ณ จุดภาพจุดหนึ่ง ซึ่งไม่เพียงพอที่จะคำนวณความเร็วของวัตถุได้ การแบ่งภาพออกเป็นแผงเล็ก ๆ โดยสมมติให้ความเร็วทุกจุดภาพภายในแผงนั้นเคลื่อนที่ด้วยความเร็วเท่ากัน สมมติฐานนี้สามารถคำนวณความเร็วของแผงภาพนั้นด้วยวิธีการกำลังสองน้อยที่สุด (The least squares approach) ดังสมการที่ (3) ซึ่งเป็นพื้นฐานที่ใช้ในงานวิจัยของ tomasi [2]

$$v_{LS} = \arg \min_p \sum_p \left(\nabla I^T(p) v + I_t(p) \right)^2 \quad (3)$$

โดย v_{LS} คือความเร็วกำลังสองน้อยที่สุดของแผนภาพ ผลรวมครอบคลุมทุกจุดภาพ จำนวน N จุดภาพ ในแผนภาพอื่นหนึ่ง ๆ ซึ่งแทนด้วย p เรารวบรวมสมการที่ (4) เพื่อความสะดวกต่อการวิเคราะห์

$$\begin{aligned} v_{LS} &= \arg \min_v \|Av + b\|^2 \\ &= \arg \min_v (v^T G v + 2v^T c) \end{aligned} \quad (4)$$

เมื่อ $G = A^T A$; $c = A^T b$;

$$A = \begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \vdots \\ I_{xN} & I_{yN} \end{bmatrix}; \quad b = \begin{bmatrix} I_{t1} \\ I_{t2} \\ \vdots \\ I_{tN} \end{bmatrix}$$

และ

$$G = \begin{bmatrix} \sum_p I_x^2(p) & \sum_p I_x(p) I_y(p) \\ \sum_p I_x(p) I_y(p) & \sum_p I_y^2(p) \end{bmatrix} \quad (5)$$

จากงานวิจัยของ Tomasi [2] ค่าลำดับชั้นของ G สามารถบอกสถานะของแผนภาพได้ โดย $\text{rank}(G) = 0$ เป็นแผนภาพที่ไม่มีการเคลื่อนที่ และ $\text{rank}(G) = 1$ กับ $\text{rank}(G) = 2$ เป็นแผนภาพที่มีการเคลื่อนที่ การนิยามสถานะของแผนภาพอีกวิธีหนึ่ง ที่สามารถเพิ่มประสิทธิภาพให้ถูกต้องยิ่งขึ้น ได้มาโดยการคำนวณความเร็วของแผนภาพด้วยวิธีการลดค้อยกำลังสองน้อยที่สุดทั้งหมด (Total least square regression) ดังสมการที่ (6)

$$r^{TLS} = \arg \min_r \sum_p \left(\left[\nabla I^T(p) | I_t(p) \right] r \right)^2, \|r\|^2 = 1 \quad (6)$$

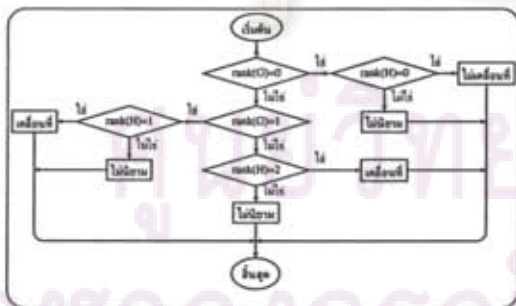
เมื่อ r^{TLS} คือความเร็วการลดค้อยกำลังสองน้อยที่สุดทั้งหมดของแผนภาพ และ r ประกอบด้วย 3 เวกเตอร์ ซึ่งมีความสัมพันธ์กับความเร็วดังนี้ $v_x^{TLS} = r_1^{TLS}$ กับ $v_y^{TLS} = r_2^{TLS}$ สมการที่ (6) ถูกจัดให้สะดวกต่อการวิเคราะห์ ดังสมการที่ (7)

$$r^{TLS} = \arg \min_r r^T H r; \|r\|^2 = 1 \quad (7)$$

เมื่อ

$$H = \begin{bmatrix} G & \begin{bmatrix} \sum_p I_x(p) I_t(p) \\ \sum_p I_y(p) I_t(p) \end{bmatrix} \\ \begin{bmatrix} \sum_p I_x(p) I_t(p) & \sum_p I_y(p) I_t(p) \end{bmatrix} & \sum_p I_t^2(p) \end{bmatrix} \quad (8)$$

จากงานวิจัยของ Haußecker [3] ค่าลำดับชั้นของ H สามารถบอกสถานะของแผนภาพ โดย $\text{rank}(H) = 0$ เป็นแผนภาพที่ไม่มีการเคลื่อนที่ และ $\text{rank}(H) = 1$ กับ $\text{rank}(H) = 2$ เป็นแผนภาพที่มีการเคลื่อนที่ สรุปขั้นตอนการนิยามแผนภาพดังรูปที่ 2



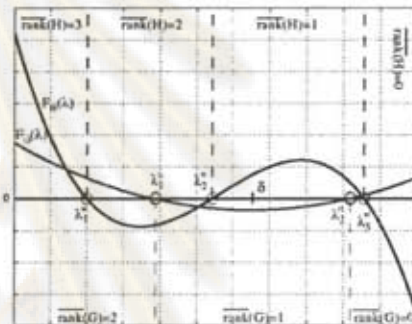
รูปที่ 2: ขั้นตอนวิธีการระบุสถานะของแผนภาพจากค่าลำดับชั้นของเมทริกซ์

2.1 การคำนวณค่าลำดับชั้น

การหาค่าลำดับชั้นของเมทริกซ์ที่มีสัญญาณรบกวนผสมอยู่ ปรกติจะใช้วิธีการ SVD (Singular Value Decomposition) กำหนดให้ $\lambda_1 \leq \lambda_2 \leq \lambda_3$ เป็นค่าลักษณะเฉพาะ (Eigenvalue) ของเมทริกซ์ H แต่การคำนวณค่าลักษณะเฉพาะ ต้องทราบ (Root) ของพหุนาม (Polynomial) เพื่อหลีกเลี่ยงการคำนวณแบบตรง ดังนั้นงานวิจัยนี้จึงเลือกการประมาณค่าลำดับชั้นของ Benedetti [4] ด้วยการเปรียบเทียบค่าลักษณะเฉพาะกับขีดเริ่มเปลี่ยนแทนการคำนวณโดยตรง กำหนดให้ $\overline{\text{rank}}$ แทนค่าลำดับชั้นที่ได้จากการคำนวณตามวิธีของ Benedetti [4]

พิจารณาพหุนามลักษณะเฉพาะ $F_G(\lambda) = |G - \lambda I_2|$ กับ $F_H(\lambda) = |H - \lambda I_3|$ โดยที่ I_n เป็นเมทริกซ์เอกลักษณ์ขนาด $n \times n$ ส่วน $F_G(\lambda)$ กับ $F_H(\lambda)$ เป็นพหุนามกำลังสอง กับ พหุนามกำลังสามตามลำดับ รูปที่ 3 เป็นตัวอย่างของ $F_G(\lambda)$ กับ $F_H(\lambda)$ และขีดเริ่มเปลี่ยน δ โดยมีกราฟหาค่าลำดับชั้นของเมทริกซ์ G ตามวิธีของ Benedetti [4] ดังนี้

- $\overline{\text{rank}}(G) = 0$ ก็ต่อเมื่อ $F_G(\delta) > 0$ และ $G_{11} < 0$
- $\overline{\text{rank}}(G) = 1$ ก็ต่อเมื่อ $F_G(\delta) < 0$
- $\overline{\text{rank}}(G) = 2$ ก็ต่อเมื่อ $F_G(\delta) > 0$ และ $G_{11} > 0$



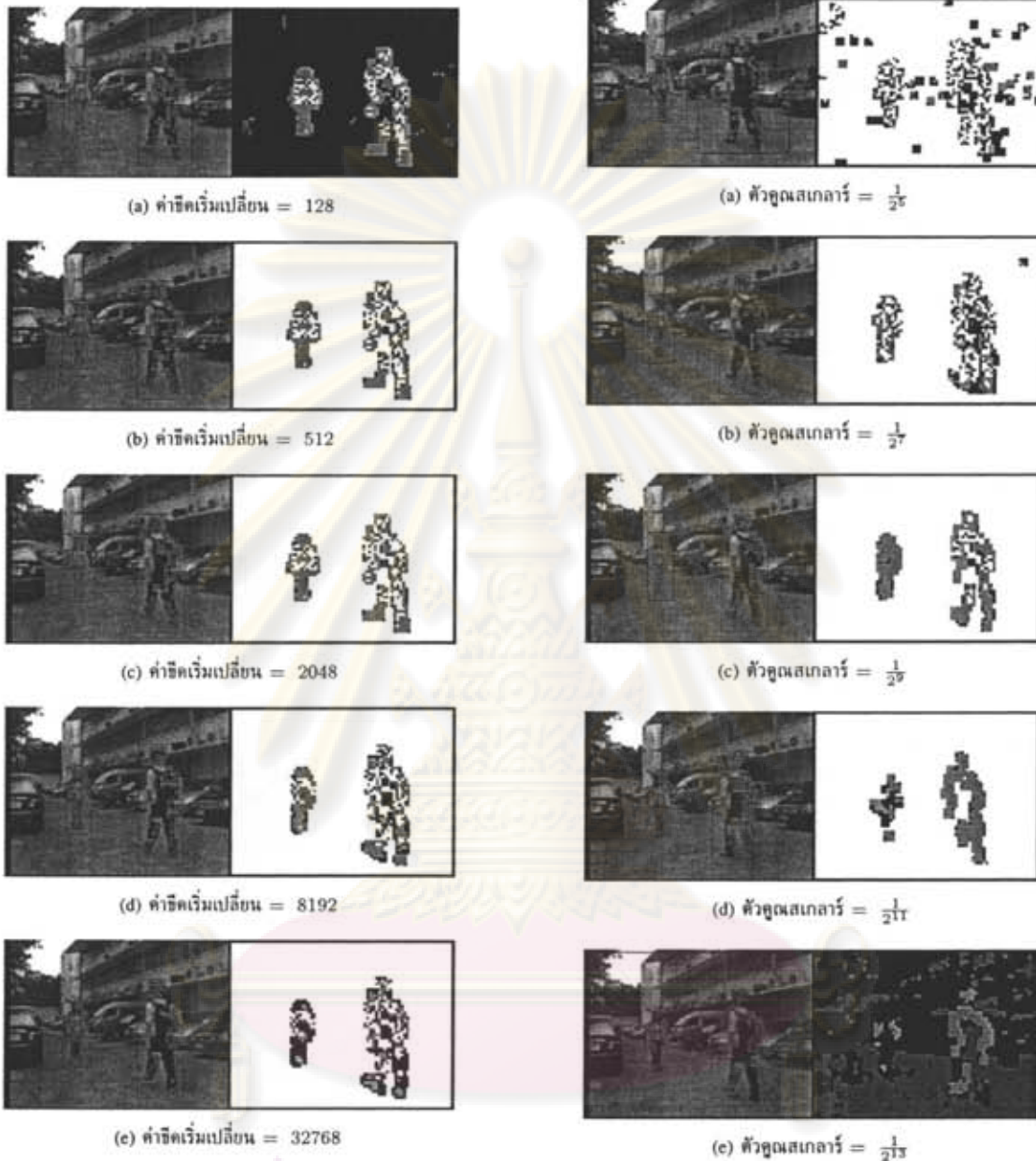
รูปที่ 3: จุดลักษณะพหุนาม $F_G(\lambda)$ (สีแดง) กับ $F_H(\lambda)$ (สีดำ) การเลือกค่าขีดเริ่มเปลี่ยน δ มีผลต่อค่าลำดับชั้นของ G กับ H (รูปประกอบคำอธิบายนำมาจาก Lu [5])

ในรูปที่ 3 $\overline{\text{rank}}(H)$ มีค่าเท่ากับ $\overline{\text{rank}}(G)$ หรือ $\overline{\text{rank}}(G) + 1$ การคำนวณค่า $\overline{\text{rank}}(H)$ ที่ชัดเจนนั้น ต้องทราบค่าที่แน่นอนของ $\overline{\text{rank}}(G)$ ก่อน แล้วพิจารณาเครื่องหมายของ $F_H(\delta)$ ตัวอย่างเช่น $\overline{\text{rank}}(G) = 1$ และ $F_H(\delta) > 0$ ดังนั้น $\overline{\text{rank}}(H) = 1$

กรณีที่ค่าลำดับชั้นของ $H = 0$ สามารถทำให้การตรวจสอบแผนภาพทำงานได้เร็วขึ้น เพราะรอยเมทริกซ์ H (Trace of H) เท่ากับผลรวมของค่าลักษณะเฉพาะ ดังนั้นถ้า $\text{trace}(H) < \delta$ ทำให้ทราบได้ว่า $\lambda_3 < \delta$ หรือ $\overline{\text{rank}}(H) = 0$ เป็นแผนภาพหยุดนิ่ง



รูปที่ 4: กรอบสี่เหลี่ยมคือแผนภาพเคลื่อนที่ กรอบสีแดงคือขอบล้อมของวัตถุเคลื่อนที่ที่ได้จากการขยายบริเวณ



รูปที่ 5: ตัวอย่างผลลัพธ์ที่ค่าขีดเริ่มเปลี่ยนมีค่าตั้งแต่ 128 - 32768 กำหนดให้ขนาดตัวคูณสเกลาร์เท่ากับ $\frac{1}{25}$ คงที่ หน้าต่างด้านขวา พื้นที่สีแดงกับพื้นสีเทาแทนแมงภาพที่เคลื่อนที่ พื้นที่สีขาวแทนแมงภาพไม่เคลื่อนที่ พื้นที่สีดำแทนแมงภาพที่นิยามไม่ได้ หน้าต่างด้านซ้าย กรอบสีเขียวเป็นบริเวณที่มีการเคลื่อนที่ กรอบสีแดงบริเวณวัตถุเคลื่อนที่ซึ่งเป็นผลลัพธ์จากการขยายบริเวณความเงื่อนที่เสนอ

รูปที่ 6: ผลลัพธ์ที่ได้จากการปรับตัวคูณสเกลาร์ในช่วง $\frac{1}{25} - \frac{1}{15}$ กำหนดให้ค่าขีดเริ่มเปลี่ยนเท่ากับ 4096 คงที่ หน้าต่างด้านขวา พื้นที่สีแดงกับพื้นสีเทาแทนแมงภาพที่เคลื่อนที่ พื้นที่สีขาวแทนแมงภาพไม่เคลื่อนที่ พื้นที่สีดำแทนแมงภาพที่นิยามไม่ได้ หน้าต่างด้านซ้าย กรอบสีเขียวเป็นบริเวณที่มีการเคลื่อนที่ กรอบสีแดงบริเวณวัตถุเคลื่อนที่ซึ่งเป็นผลลัพธ์จากการขยายบริเวณความเงื่อนที่เสนอ

จุฬาลงกรณ์มหาวิทยาลัย

3 การขยายบริเวณของแผนภาพเคลื่อนที่

เราสามารถระบุขอบของวัตถุเคลื่อนที่ในลำดับภาพวิดีโอที่แสดงในรูปที่ 4 โดยที่กรอบสี่เหลี่ยมที่ล้อมรอบที่มีการเคลื่อนที่ และวัตถุในกรอบสี่เหลี่ยมแสดงขอบของขอบของวัตถุเคลื่อนที่ด้วยวิธีการขยายบริเวณ เนื่องจากแผนภาพที่มีการเคลื่อนที่กระจายตัวและเรียงชิดติดกันทั่ววัตถุหมายเลขสอง แต่สำหรับวัตถุในกรอบสี่เหลี่ยมหมายเลขหนึ่งแผนภาพที่มีเคลื่อนที่กระจายตัวแต่ไม่เรียงชิดติดกัน จึงไม่สามารถใช้วิธีการขยายบริเวณตามปกติได้ แต่เราเสนอวิธีการปรับปรุงข้อกำหนดการขยายบริเวณสำหรับแก้ปัญหา เริ่มจากการเปลี่ยนข้อมูลจากเดิมที่ต้องการแผนภาพที่มีการเคลื่อนที่เท่านั้น เปลี่ยนเป็นขอบสี่เหลี่ยมที่ครอบคลุมการขยายบริเวณของแผนภาพที่มีการเคลื่อนที่ของวัตถุนั้น โดยแผนภาพที่มีการเคลื่อนที่ที่ใกล้ที่สุดของแผนภาพที่มีการเคลื่อนที่หนึ่ง ๆ ต้องมีระยะห่างกันไม่เกินค่าขีดเริ่มเปลี่ยนที่กำหนด ดังนั้นภายในขอบของวัตถุเคลื่อนที่ประกอบด้วยแผนภาพที่มีการเคลื่อนที่และแผนภาพที่ไม่มีการเคลื่อนที่ ดังนั้นผลลัพธ์ในกรอบสี่เหลี่ยมสีแดงหมายเลข 1 กับ 2 ในรูปที่ 4

4 ผลการทดลอง

การทดลองนี้เลือกแผนภาพที่มีขนาด 4×4 จุดภาพ การคำนวณผลรวมที่แผนภาพหนึ่ง ๆ ไม่มีปัญหาการล้น (overflow) เกิดขึ้นเพราะ ผลรวมทั้ง 16 ครั้งของผลคูณความเข้มแสงขนาดมากที่สุด 8 บิต (bit) ต้องการพื้นที่สำหรับการเก็บจุดภาพขนาด 20 บิตเป็นอย่างน้อย ซึ่งการทดลองใช้ภาษา C/C++ ร่วมกับไลบรารี OpenCV [6] ที่มีพารามิเตอร์เก็บข้อมูลจุดภาพขนาด 32 บิต แต่การล้นจะเป็นปัญหาเมื่อการประมวลผลแผนภาพหนึ่ง ต้องนำแผนภาพรอบข้างมาคำนวณด้วย การคำนวณค่าลำดับชั้นจึงเกิดปัญหาได้ เราเสนอการแก้ปัญหาการล้นโดยใช้การคูณเลขยกกำลังกับเมทริกซ์ที่ต้องการคำนวณค่าลำดับชั้น ตัวคูณเลขยกกำลังที่ใช้มีขนาดเป็น 2^k โดยที่ $k = 1, 2, \dots$ ซึ่งเป็นการเลื่อนบิตของพารามิเตอร์ไปทางขวาเท่านั้น

พารามิเตอร์ที่สำคัญที่ใช้ตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอ คือ ค่าขีดเริ่มเปลี่ยนสำหรับการคำนวณค่าลำดับชั้นกับขนาดของตัวคูณเลขยกกำลังเพื่อหลีกเลี่ยงการล้น เทคนิคที่ใช้ในการเลือกปรับค่าขีดเริ่มเปลี่ยนนี้ คือ การเลือกค่าของพารามิเตอร์ทั้งสองที่ทำให้แผนภาพเคลื่อนที่สอดคล้องกับวัตถุเคลื่อนที่จริงในลำดับภาพวิดีโอ โดยมีแผนภาพที่นิยามไม่ได้น้อยที่สุด ตัวอย่างบางส่วนของผลการทดลองปรับค่าขีดเริ่มเปลี่ยนแสดงในรูปที่ 5 และตัวอย่างบางส่วนของผลการทดลอง การปรับขนาดตัวคูณเลขยกกำลังแสดงในรูปที่ 6 สำหรับการปรับค่าขีดเริ่มเปลี่ยนแสดงในรูปที่ 5(a) กับรูปที่ 5(c) สังเกตจากหน้าคางคางของแต่ละรูป ผลของการประมวลผลของแผนภาพเป็นพื้นที่ ที่ไม่นิยามเพราะค่าลำดับชั้นของ G กับ H ไม่เท่ากัน แม้ผลของการประมวลผลจะผิดพลาดค่อนข้างมาก แต่การขยายบริเวณที่เสนอในวิธีการของงานวิจัยนี้ช่วยทำให้ผลลัพธ์หรือขอบสอดคล้องกับวัตถุเคลื่อนที่จริงในลำดับภาพวิดีโอ สำหรับการปรับขนาดตัวคูณเลขยกกำลังเป็นการหารชนิดหนึ่ง ถ้าตัวหารน้อยหรือมากเกินไปจะทำให้ผลลัพธ์การประมวลผลของแผนภาพผิดพลาดได้ ดังตัวอย่างในรูปที่ 6(a) เป็นตัวอย่างผลลัพธ์การประมวลผลของภาพที่ผิดพลาดเพราะตัวหารมีขนาดเล็กเกินไป ส่วนตัวอย่างในรูปที่ 6(c) เป็นตัวอย่างผลลัพธ์การประมวลผลของภาพที่ผิดพลาดเพราะตัวหารมีขนาดใหญ่เกินไป

ในการทดลองนี้ค่าที่เหมาะสมที่สามารถให้ผลลัพธ์ตามเกณฑ์ที่ต้องการเป็นค่าขีดเริ่มเปลี่ยนที่มีค่าในช่วง $256 - 8192$ ดังตัวอย่างในรูปที่ 5(b) ถึงรูปที่ 5(d) และขนาดตัวคูณเลขยกกำลังมีค่าในช่วง $2^4 - 2^{16}$ ดังตัวอย่างในรูปที่ 6(b) ถึง รูปที่ 6(d) ช่วงของค่าที่เหมาะสมทั้งสองนี้มีผลต่อการตรวจหาวัตถุเคลื่อนที่ในลำดับภาพวิดีโอเป็นช่วงที่กว้าง แสดงว่าขนาดของพารามิเตอร์ค่าหนึ่งสามารถใช้ได้หลายสถานะการณ

5 สรุป

การตรวจหา วัตถุ เคลื่อนที่ ใน ลำดับ ภาพ วิดีทัศน์ สามารถ คำนวณ โดย ใช้ เลขคณิตจุดคงที่ (fixed-point arithmetic) และมีผลลัพธ์เป็นขอบของวัตถุ

เคลื่อนที่ที่ก่อเมื่อเกิดการเคลื่อนที่ของวัตถุนั้น ที่สำคัญไม่ต้องเก็บภาพจากหลังทุกเฟรม และ ไม่ต้องคำนวณผลกระทบการเปลี่ยนแปลงของแสงที่เล็กน้อย จึงเหมาะกับการทำงานในระบบฝังตัว พร้อมทั้งวิธีการขยายบริเวณแบบใช้ค่าขีดเริ่มเปลี่ยนในการพิจารณาทำให้พารามิเตอร์มีความคงทนและสามารถใช้ได้กับสถานะการณที่แตกต่างกัน

วิธีการตรวจหาวัตถุเคลื่อนที่ด้วยการไหลเชิงแสงในบทความนี้ มีผลลัพธ์ที่ต่างจากงานวิจัยของ Lu [5] เพราะ Lu [5] สนใจผลลัพธ์เป็นแผนภาพที่มีการเคลื่อนที่เท่านั้น จึงใช้การแพร่กระจายความเชื่อ (belief propagation) พร้อมการทำมัลติสเกล (multiscale) กับแผนภาพที่ถูกระบุสถานะจากค่าลำดับชั้น ซึ่งอาจไม่ได้มีความเร็วที่สอดคล้องกันทั่วทั้งวัตถุ ทำให้ข้อมูลของวัตถุที่สนใจนั้นขาดหาย แต่วิธีที่เราเสนอในบทความนี้เราสนใจผลลัพธ์เป็นขอบสี่เหลี่ยมที่ครอบคลุมบริเวณของวัตถุเคลื่อนที่ เราแก้ปัญหาการกระจายที่ไม่เรียงชิดติดกันของแผนภาพเคลื่อนที่ของวัตถุหนึ่ง ด้วยการขยายบริเวณแบบใช้ค่าขีดเริ่มเปลี่ยน ทำให้ได้ข้อมูลของวัตถุเคลื่อนที่อย่างครบถ้วน

งานที่จะทำต่อไป คือการนำวิธีนี้ไปใช้ในระบบฝังตัว โดยนำขอบที่คำนวณได้เป็นตัวจำกัดบริเวณของภาพวิดีโอที่ต้องการบีบอัดก่อนส่งออกผ่านเครือข่าย (network) พร้อมกับการรายงานผลตำแหน่งของขอบในภาพเฟรมหนึ่ง ๆ เพื่อคอมพิวเตอร์แม่ข่ายอัปเดตข้อมูลที่เป็นที่แน่นอน

เอกสารอ้างอิง

- [1] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice Hall, 1982.
- [2] C. Tomasi and T. Kanade, "Detection and tracking of point features," Tech. Rep. CMU-CS-91-132, Carnegie Mellon University, April 1991.
- [3] H. Haußecker and B. Jähne, "A tensor approach for precise computation of dense displacement vector fields," in *Mustererkennung 1997, 19. DAGM-Symposium*, (London, UK), pp. 199-208, Springer-Verlag, 1997.
- [4] A. Benedetti and P. Perona, "Real-time 2-d feature detection on a reconfigurable computer," 1998.
- [5] X. Lu and R. Manduchi, "Fast image motion computation on an embedded computer," in *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, (Washington, DC, USA), p. 120, IEEE Computer Society, 2006.
- [6] "Open source computer vision library," <http://www.intel.com/technology/computing/opencv/index.htm>.



ชัยพร ชัยवालกิจกุล จบการศึกษาระดับปริญญาตรี สาขาวิศวกรรมไฟฟ้าจาก จุฬาลงกรณ์มหาวิทยาลัยเมื่อปี พ.ศ. 2546 ปัจจุบันกำลังศึกษาต่อระดับปริญญาโท สาขาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย



สุริย์ พุมรินทร์ จบการศึกษาระดับปริญญาตรี สาขาวิศวกรรมอิเล็กทรอนิกส์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ระดับปริญญาโท สาขาวิศวกรรมไฟฟ้า มหาวิทยาลัยอริโซนา สหรัฐอเมริกา และระดับปริญญาเอก สาขาวิศวกรรมไฟฟ้า มหาวิทยาลัยวอชิงตัน ปัจจุบันดำรงตำแหน่งอาจารย์ประจำภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย มีความสนใจทางด้าน การประมวลผลภาพ คอมพิวเตอร์วิทัศน์ และ ระบบฝังตัว

ประวัติผู้เขียนวิทยานิพนธ์

ว่าที่ ร้อยตรี ชัยพร ชัชวาลกิจกุล เกิดเมื่อวันจันทร์ที่ 20 เมษายน พ.ศ. 2524 จังหวัดกำแพงเพชร เป็นบุตรของนายชัยศิลป์ และนางประทีน ชัชวาลกิจกุล สำเร็จการศึกษาระดับปริญญาตรี หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า จากจุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2546 หลังจากสำเร็จการศึกษา ได้เข้าร่วมวิจัย และพัฒนาเครื่องวัดพลังงานไฟฟ้าเก็บหน่วยไฟฟ้าตามช่วงเวลา สังกัดศูนย์เชี่ยวชาญเฉพาะทางเทคโนโลยีไฟฟ้ากำลัง คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เป็นเวลาหนึ่งปี จากนั้นทำงานให้กับ NR. INDUSTRIES co., Ltd. จังหวัดสมุทรปราการ ก่อนเข้าศึกษาหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ในปีการศึกษา 2548 ณ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย สังกัดห้องปฏิบัติการวิจัยออกแบบและประยุกต์วงจรรวม



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย