

การสร้างระบบคลังวีดิทัศน์ติดตามเป้าหมายแบบเวลาจริงบนบอร์ด DSP TMS320C6713



นายประวิทย์ ชินประเสริฐสุข

ศูนย์วิทยทรัพยากร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2552

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

IMPLEMENTATION OF REAL TIME TARGET TRACKING VIDEO CAMERA ON DSP
TMS320C6713 BOARD



Mr.Pravit Chinprasertsuk

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 2009

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์

การสร้างระบบกล้องวิดีโอที่บันทึกตามเป้าหมายแบบเวลาจริง
บนบอร์ด DSP TMS320C6713

โดย

นายประวิช ชินประเสริฐสุข

สาขาวิชา

วิศวกรรมไฟฟ้า

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

รองศาสตราจารย์ ดร.สมชาย จิตะพันธ์กุล

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

อาจารย์ สุวิทย์ นาคพิระยุทธ

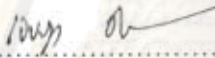
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้ให้นักศึกษานี้เป็นส่วน
ส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาโท



คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร.บุญสม เลิศศิริวงษ์)

คณะกรรมการสอบวิทยานิพนธ์



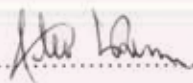
ประธานกรรมการ

(รองศาสตราจารย์ ดร.เจษฎา ชินรุ่งเรือง)



อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(รองศาสตราจารย์ ดร.สมชาย จิตะพันธ์กุล)



อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

(อาจารย์ สุวิทย์ นาคพิระยุทธ)



กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.วิทยากร อัครวิเศษ)



กรรมการภายนอกมหาวิทยาลัย

(รองศาสตราจารย์ ดร.วุฒิพงศ์ อารีกุล)

ศูนย์วิทยุโทรคมนาคม
จุฬาลงกรณ์มหาวิทยาลัย

ประวิช ชินประเสริฐสุข : การสร้างระบบกล้องวีดิทัศน์ติดตามเป้าหมายแบบเวลาจริงบนบอร์ด DSP TMS320C6713 (IMPLEMENTATION OF REAL TIME TARGET TRACKING VIDEO CAMERA ON DSP TMS320C6713 BOARD) อ.ที่ปรึกษา
 วิทยานิพนธ์หลัก : รศ.ดร.สมชาย จิตะพันธ์กุล, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม : อ. สุวิทย์ นาคพิระยุทธ, 81 หน้า.

ระบบสำรวจและตรวจการที่ใช้กล้องวีดิทัศน์ที่ติดตั้งบนยานพาหนะที่เคลื่อนที่ กำลังแพร่หลายในปัจจุบัน โดยเฉพาะยานพาหนะไร้คน (unmanned vehicle) หนึ่งในยานพาหนะที่นิยมใช้ในการนี้คือ อากาศยานไร้คนบินหรือ UAV วิทยานิพนธ์นี้กล่าวถึงการพัฒนาาระบบติดตามเป้าหมายแบบเวลาจริงในสภาพแวดล้อมของการปฏิบัติการของอากาศยานไร้คนบินที่มีการรบกวนสูง โดยเลือกใช้อัลกอริธึมมินชิฟต์สำหรับติดตามเป้าหมายเพราะเป็นอัลกอริธึมที่เหมาะสมสำหรับกล้องบนพาหนะที่เคลื่อนที่และไม่ซับซ้อนเกินกว่าการประมวลผลในเวลาจริง อัลกอริธึมมินชิฟต์เมื่อใช้ติดตามภาพเป้าหมายจะติดตามร่องรอยของคุณลักษณะของเป้าหมาย เช่น สีของเป้าหมาย การติดตามร่องรอยจะทำให้ระบบติดตามทนทานต่อการเปลี่ยนแปลงขนาดและรูปร่างของเป้าหมายรวมถึงการเบลอที่เกิดจากการสั่นไหวของกล้องเมื่อเปรียบเทียบกับวิธีใช้การแยกแยะวัตถุเป้าหมายจากฉากหลัง

ในวิทยานิพนธ์นี้ได้พัฒนาโปรแกรมการติดตามเป้าหมายด้วยมินชิฟต์ 2 วิธีคือ มินชิฟต์แบบทั่วไปซึ่งใช้ติดตามคุณลักษณะของเป้าหมาย และมินชิฟต์แบบเคอร์เนลเบสท์ซึ่งใช้การเปรียบเทียบการกระจายความหนาแน่นความน่าจะเป็น โดยทั้งสองวิธีถูกนำไปพัฒนาใช้งานบนบอร์ด DSP TMS320C6713 ที่ความเร็ว 225 MHz เพื่อติดตามเป้าหมายในเวลาจริงและควบคุมกล้องแบบกัมเมยสายโดยใช้การควบคุมป้อนกลับแบบ PI ผลลัพธ์ที่ได้แสดงให้เห็นว่าการติดตามเป้าหมายด้วยมินชิฟต์แบบเคอร์เนลเบสท์มีประสิทธิภาพดีกว่ามินชิฟต์แบบทั่วไปถึงแม้ว่าจะใช้เวลาประมวลผลมากกว่า โดยที่มีการประมวลผลต่อรอบเท่ากับ 3.95 และ 2.76 ms ตามลำดับ สำหรับหน้าต่างค้นหาขนาด 50x50 พิกเซล

จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา.....วิศวกรรมไฟฟ้า
 สาขาวิชา.....วิศวกรรมไฟฟ้า
 ปีการศึกษา..... 2552

ลายมือชื่อนิสิตร *Ken Kveanit*
 ลายมือชื่ออ.ที่ปรึกษาวิทยานิพนธ์หลัก *[Signature]*
 ลายมือชื่ออ.ที่ปรึกษาวิทยานิพนธ์ร่วม *[Signature]*

5070693821 : MAJOR ELECTRICAL ENGINEERING

KEYWORDS : UAV / TARGET TRACKING / MEAN SHIFT / DSP / REAL TIME

PRAVIT CHINPRASERTSUK : IMPLEMENTATION OF REAL TIME TARGET TRACKING VIDEO CAMERA ON DSP TMS320C6713 BOARD. ADVISOR : ASSOC. PROF. SOMCHAI JITAPUNKUL, Dr. Ing., CO-ADVISOR : MR. SUVIT NAKPEERAYUTH, 81 pp.

Mobile surveillance and reconnaissance systems using video camera are gaining popularity, especially, Unmanned Aerial Vehicle or UAV. This thesis describes the development of real time target tracking video camera for UAV noisy operating environment. Mean shift algorithm is selected since it is suitable for mobile applications but simple enough for real time video processing. Mean shift algorithm, for video target tracking, traces the trail of target's features between frames, e.g., target color. In comparison to target segmentation based algorithm, tracing the target trail by mean shift provides robustness against size and orientation changes, including motion blur.

In this thesis, two mean shift target tracking algorithms were implemented, conventional mean shift which tracks target feature only, and kernel-based mean shift which tracks pdf instead. Both were implemented on 225 MHz TMS320C6713 DSP board to track target image in real time and control PTZ camera using PI controller. The results show that kernel-based is more robust than conventional mean shift but required more processing time, 3.95 and 2.76 ms per iteration respectively for 50x50 pixels search window.

Department : Electrical Engineering.....

Field of Study : Electrical Engineering.....

Academic Year : 2009.....

Student's Signature 

Advisor's Signature 

Co-Advisor's Signature 

กิตติกรรมประกาศ

ขอขอบพระคุณรองศาสตราจารย์ ดร.สมชาย จิตะพันธ์กุล อาจารย์ที่ปรึกษา
วิทยานิพนธ์ และ อาจารย์ สุวิทย์ นาคไพระยุทธ อาจารย์ที่ปรึกษาร่วมที่ให้คำปรึกษาและให้ความ
ช่วยเหลืออย่างดียิ่งแก่ผู้วิจัยมาโดยตลอด ขอขอบคุณอาจารย์ทุกท่านที่ได้ให้ความรู้อันมีค่าให้แก่
ผู้วิจัย ทำให้งานวิจัยและวิทยานิพนธ์ฉบับนี้สำเร็จลงด้วยดี

ขอขอบคุณคุณปิติจิตร์ สุทธาโรจน์ และคุณณัฐชัย วัชรภินชัย ที่ให้ข้อมูลที่เป็น
ประโยชน์ต่อการวิจัย

สุดท้าย ผู้วิจัยขอขอบพระคุณ คุณพ่อ คุณแม่ ภรรยา และบุตรที่ให้ความ
สนับสนุนและกำลังใจมาโดยตลอด



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญตาราง.....	ญ
สารบัญภาพ	ฎ
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 เป้าหมายและขอบเขตของการวิจัย	2
1.4 ขั้นตอนและวิธีดำเนินการวิจัย.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 ระบบกล้องวีดิทัศน์ (camera system).....	4
2.1.1 แบบจำลองกล้องวีดิทัศน์.....	4
2.1.1.1 การฉายภาพแบบทัศนมิติ (perspective projection).....	5
2.1.1.2 การฉายภาพแบบขนาน (parallel projection หรือ orthogonal projection)	5
2.1.2 การเคลื่อนที่ของกล้องวีดิทัศน์.....	5
2.1.2.1 การเคลื่อนที่ของกล้องที่ตำแหน่งศูนย์กลางการฉายภาพมีการเคลื่อนที่	5
2.1.2.2 การเคลื่อนที่ของกล้องที่ตำแหน่งศูนย์กลางการฉายภาพไม่เปลี่ยนแปลง	6
2.1.3 การซูม (zoom)	6
2.2 ระบบปรับเสถียรวีดิทัศน์	6
2.2.1 ระบบปรับเสถียรทางกล.....	7
2.2.1.1 ระบบแพสซีฟ (passive stabilizer)	7
2.2.1.2 ระบบแอ็กทีฟ (active stabilizer)	7
2.2.2 ระบบปรับเสถียรแบบอิเล็กทรอนิกส์	8
2.3 ปรีกวมิตี	8
2.3.1 ปรีกวมิตี RGB.....	9

2.3.2	ปริภูมิสี YCbCr.....	9
2.3.3	ปริภูมิสี HSV	10
2.4	มินชิฟต์.....	12
2.5	ระยะห่าง Bhattacharyya	14
2.6	งานวิจัยที่เกี่ยวข้อง	15
บทที่ 3	วิธีดำเนินการวิจัย.....	17
3.1	ระบบติดตามเป้าหมาย	17
3.1.1	กล้องวีดิทัศน์ประเภทสายก้มเงยและซูม (Pan-Tilt-Zoom Camera, PTZ)	18
3.1.2	บอร์ดประมวลผลวีดิทัศน์.....	19
3.2	อัลกอริทึมการติดตามเป้าหมาย.....	21
3.2.1	การใช้มินชิฟต์แบบทั่วไปจากปริภูมิคุณลักษณะ (feature space)	21
3.2.1.1	มินชิฟต์แบบทั่วไปในการติดตามภาพเป้าหมาย	22
3.2.1.2	การเลือกปริภูมิสี.....	23
3.2.1.3	การสร้างฮิสโตแกรมและการถ่วงน้ำหนัก.....	26
3.2.1.4	มินชิฟต์แบบทั่วไปในการติดตามเป้าหมายภาคปฏิบัติ.....	27
3.2.1.5	การปรับขนาดหน้าต่างขณะจับยึดเป้าหมาย.....	29
3.2.2	การใช้มินชิฟต์แบบเคอร์เนลเบสท์	31
3.2.2.1	แบบจำลองเป้าหมาย (target model)	34
3.2.2.2	ตัวแทนเป้าหมาย (target candidate).....	35
3.2.2.3	การติดตามเป้าหมายภาคปฏิบัติ.....	35
3.3	การควบคุมการสายก้มเงยของกล้องวีดิทัศน์	38
3.3.1	การควบคุมแบบ PID.....	38
3.3.1.1	ตัวควบคุมแบบสัดส่วน.....	39
3.3.1.2	ตัวควบคุมอินทิกรัล.....	39
3.3.1.3	ตัวควบคุมแบบอนุพันธ์	39
3.3.2	การหาพารามิเตอร์ด้วยวิธี Ziegler-Nicole.....	40
3.3.3	ตัวควบคุมแบบ PID แบบดีสครีต	41
3.3.4	การควบคุมกล้อง	42
3.3.5	การหาค่าพารามิเตอร์ค่า Kp และ Ki ที่อัตราซุ่มต่างๆ	43
3.3.6	การไวด์อิฟ (windup) และการป้องกัน.....	44

3.4 โครงสร้างซอฟต์แวร์.....	45
3.4.1 การโอนย้ายข้อมูล.....	46
3.4.2 การประมวลผลการปรับเสถียรวิถีทัศน์.....	47
3.4.3 การติดตามเป้าหมาย.....	53
บทที่ 4 ผลการวิจัย.....	55
4.1 การติดตามเป้าหมาย.....	55
4.1.1 เวลาในการประมวลผล.....	55
4.1.2 เปรียบเทียบสมรรถนะของการติดตามเป้าหมายทั้งสองวิธี.....	56
4.1.3 ข้อเปรียบเทียบในการติดตามเป้าหมายแบบทั่วไปและแบบเคอร์เนลเบสส์.....	58
4.1.3.1 ความสำเร็จในการยึดจับเป้าหมาย.....	58
4.1.3.2 ความทนทานในการเกาะติดเป้าหมาย.....	59
4.1.3.3 ความเร็วในการประมวลผล.....	59
4.1.4 จำนวนวนรอบต่อเฟรมในการติดตามเป้าหมาย.....	59
4.1.5 ความทนทานต่อการรบกวน.....	60
4.1.6 ความทนทานต่อการโดนบังบางส่วน (partial occlusion).....	60
4.1.7 การปรับหน้าตาต่างโดยอัตโนมัติ.....	61
4.2 การหาค่าพารามิเตอร์ Kp และ Ki โดยอัตโนมัติด้วยวิธี Ziegler-Nicole.....	61
4.2.1 ตัวอย่างการตอบสนองต่ออินพุตแบบสแต็ป.....	62
บทที่ 5 สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ.....	64
5.1 สรุปผลการวิจัย.....	64
5.2 อภิปรายผลการวิจัย.....	64
5.3 ข้อเสนอแนะ.....	65
รายการอ้างอิง.....	67
ภาคผนวก.....	69
ภาคผนวก ก.ผังการทำงานของโปรแกรมระบบติดตามเป้าหมาย.....	70
ภาคผนวก ข. โครงสร้างโปรแกรมบนบอร์ด DSP.....	71
ภาคผนวก ค. บทความทางวิชาการที่ได้รับการเผยแพร่.....	76
ประวัติผู้เขียนวิทยานิพนธ์.....	81

สารบัญตาราง

หน้า

ตาราง 3.1 แสดงการหาค่าพารามิเตอร์ของตัวควบคุมจากค่า K_c และ P_c 40

ตาราง 4.1 เวลาประมวลผลต่อการวนรอบ (time per iteration) ด้วยบอร์ด DSP ของการติดตาม
เป้าหมายด้วยวิธีมินิชิฟต์แบบทั่วไป..... 55

ตาราง 4.2 เปรียบเทียบเวลาประมวลผลในการติดตามเป้าหมายทั้งสองวิธี 56

ตาราง 4.3 คะแนนเปรียบเทียบความทนทานด้วยวิธีการติดตามเป้าหมายทั้งสองวิธี..... 57



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญญภาพ

	หน้า
รูปที่ 2.1 (ก) การฉายภาพแบบทัศนมิติ (ข) การฉายภาพแบบขนาน	4
รูปที่ 2.2 การเคลื่อนที่ของกล้อง.....	6
รูปที่ 2.3 กิมบอลส์สำหรับงานถ่ายภาพวีดิทัศน์ทางอากาศ.....	7
รูปที่ 2.4 ตัวอย่างปริภูมิสีที่อยู่ในแบบจำลองสี RGB	9
รูปที่ 2.5 แสดงการผสมแม่สีแสงแดงเขียวน้ำเงินในปริภูมิสี RGB	9
รูปที่ 2.6 ปริภูมิสี HSV	10
รูปที่ 2.7 แสดงการหาความหนาแน่นของการกระจายด้วยวิธีมินิพิต.....	12
รูปที่ 3.1 ผังของระบบติดตามเป้าหมาย	18
รูปที่ 3.2 กล้องวีดิทัศน์ในการทดลอง (ก) กล้องวีดิทัศน์ EVID-100 (ข) ลักษณะการกัมเมย (ค) ลักษณะการส่าย.....	18
รูปที่ 3.3 แสดงผังการทำงานของ micro-line C6713Compact	20
รูปที่ 3.4 แสดงการทำงานในบอร์ด DSP ในระบบปรับเสถียรวีดิทัศน์และติดตามเป้าหมาย	21
รูปที่ 3.5 ภาพต้นแบบ.....	24
รูปที่ 3.6 ภาพในปริภูมิสีแบบ HSV โดยให้ V มีค่าคงที่.....	25
รูปที่ 3.7 ภาพในปริภูมิสีแบบ YCbCr โดยให้ Y มีค่าคงที่	25
รูปที่ 3.8 (ก) แสดงภาพสีของเป้าหมายที่ถูกจับยึด และ (ข) ภาพคุณลักษณะที่สร้างมาจากสีของ เป้าหมาย	28
รูปที่ 3.9 แสดงแนวคิดในการปรับภาพแบบอัตโนมัติ.....	30
รูปที่ 3.10 ผังการทำงานของระบบควบคุมการกัมส่ายกล้อง.....	38
รูปที่ 3.11 ผังควบคุมแบบป้อนกลับ.....	38
รูปที่ 3.12 ผังส่วนควบคุมแบบ PID	39
รูปที่ 3.13 ระบบควบคุมกล้องโดยแยกแกน X Y	43
รูปที่ 3.14 โครงสร้างการโอนย้ายข้อมูลบนบอร์ด DSP	46
รูปที่ 3.15 การหาเวกเตอร์โมชันในภาพความละเอียดระดับ L3	48
รูปที่ 3.16 การกำหนดบล็อกอ้างอิงของภาพความละเอียดระดับ L2 L1 และ L0.....	48
รูปที่ 3.17 การใช้เวกเตอร์โมชันของระดับความละเอียดก่อนหน้าเพื่อกำหนดขอบเขตบล็อกค้นหา	48
รูปที่ 3.18 แสดงการหาเวกเตอร์โมชันแบบใช้ภาพหลายความละเอียด.....	49

รูปที่ 3.19 รูปแบบตัวกรองปรับแก้การเคลื่อนที่ของภาพ.....	49
รูปที่ 3.20 ผลตอบสนองทางความถี่และเฟสของ filter ปรับเสถียรภาพ	51
รูปที่ 3.21 ผลตอบสนองทางเวลาของตัวกรองปรับเสถียรภาพเมื่อป้อน ramp input.....	51
รูปที่ 3.22 ตัวอย่างการชดเชยการสั่นของภาพจากภาพที่ถ่ายมาจริง	52
รูปที่ 3.23 การประมาณพิกัดใหม่ของภาพเมื่อมีการชดเชยการสั่น	53
รูปที่ 3.24 แสดงการย้ายข้อมูลจากหน้าต่างจากหน่วยความจำภายนอกสู่หน่วยความจำภายใน	54
รูปที่ 4.1 แสดงการติดตามเป้าโดยวิธีมินชีฟต์แบบเคอร์เนลเบสส์	58
รูปที่ 4.2 แสดงจำนวนรอบในการติดตามเป้าหมายกับดัชนีเฟรม	59
รูปที่ 4.3 แสดงความทนทานต่อการรบกวนจากการเบลอและการกระโดด.....	60
รูปที่ 4.4 แสดงความทนทานเมื่อเป้าหมายที่ถูกบังบางส่วน.....	60
รูปที่ 4.5 แสดงการปรับหน้าต่างโดยอัตโนมัติของการติดตามด้วยวิธีการมินชีฟต์แบบทั่วไป.....	61
รูปที่ 4.6 ค่า K_p ที่ได้จากการการหาค่าด้วยวิธี Ziegler-Nicole.....	62
รูปที่ 4.7 ค่า K_i ที่ได้จากการการหาค่าด้วยวิธี Ziegler-Nicole	62
รูปที่ 4.8 แสดงตัวอย่างการตอบสนองต่ออินพุตแบบสแต็ปของตัวควบคุม	63

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันการใช้อากาศยานไร้คนขับ (UAV) กำลังเป็นที่นิยมมากในวงการทหาร และพลเรือน ทั้งนี้เพราะระบบดังกล่าวมีความสามารถในการตรวจการ สอดแนม และสำรวจได้ดีจากระยะไกล โดยมีค่าใช้จ่ายต่ำกว่าการใช้อากาศยานแบบใช้คนขับ อีกทั้งยังสามารถปฏิบัติการภารกิจที่เป็นงานประจำเป็นระยะเวลานานหรือภารกิจที่อาจมีอันตรายต่อนักบินได้อย่างเหมาะสมและมีประสิทธิภาพ

ในปี พ.ศ. 2547 ได้มีโครงการ การพัฒนาระบบอากาศยานไร้คนขับ ที่มี สกว.เป็นผู้ให้ทุนวิจัยหลัก ได้แยกองค์ประกอบในการพัฒนาระบบอากาศยานไร้คนขับเป็น 4 ส่วน คือ

1. ระบบอากาศยาน ซึ่งหมายถึงตัวอากาศยานหรือเครื่องบิน (airframe) ซึ่งเป็นพาหนะ (vehicle) ในการนำอุปกรณ์ในการปฏิบัติการไปสู่เป้าหมาย
2. ระบบควบคุมการบิน ได้แก่ระบบควบคุมการบิน ทั้งส่วนนักบินกล (autopilot) และส่วนควบคุมด้วยคน (manual control) สำหรับกรณีพนักงานต้องการควบคุมอากาศยานเอง เช่น ขณะทำการบินขึ้น (takeoff) หรือร่อนลง (landing)
3. ระบบสื่อสาร คือระบบเชื่อมต่อสัญญาณวิทยุ (RF link) ระหว่างอากาศยานและสถานีภาคพื้น (ground station)
4. ระบบประมวลผลการสื่อสารและอุปกรณ์การภาพ ซึ่งประกอบด้วยระบบการถ่ายภาพ ระบบปรับเสถียรวิดีโอ ระบบการเข้ารหัสและถอดรหัสภาพ รวมถึงระบบจัดเก็บ แสดง และประมวลผลภาพ

ทางคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ได้รับผิดชอบในการพัฒนาในหัวข้อที่ 4 วิทยานิพนธ์นี้เป็นผลจากการวิจัยเพื่อปรับปรุงระบบปรับเสถียรภาพวิดีโอ ซึ่งเป็นส่วนหนึ่งในระบบอุปกรณ์การภาพ โดยพัฒนาระบบติดตามภาพเป้าหมายเพิ่มเข้าไปในระบบ เพื่อเพิ่มความสามารถในการสังเกตการณ์ (monitor) เป้าหมายของพนักงาน ทั้งนี้ระบบติดตามเป้าหมายจะมีประโยชน์อย่างยิ่งต่อพนักงานที่กำลังติดตามเป้าหมาย เพราะระบบจะทำการปรับ

ทิศทางของกล้องเพื่อติดตามเป้าหมายโดยอัตโนมัติ ช่วยลดภาระงานของพนักงานที่จะต้องปรับแก้ทิศทางของกล้องเพื่อให้เห็นเป้าหมายอยู่ตลอดเวลา โดยเฉพาะอย่างยิ่งเมื่อพนักงานทำการเพิ่มอัตราซูม (zoom) สูงมาก การควบคุมทิศทางผ่านคันบังคับ (control stick) นั้นจะทำได้ยากมากหรือทำไม่ได้เลยทั้งนี้เพราะการสั่นไหวเพียงเล็กน้อยจะทำให้ภาพสั่นหรือตกออกนอกจอภาพอย่างรวดเร็ว

ระบบติดตามเป้าหมายนี้ยังสามารถนำไปประยุกต์ใช้ในกิจการอื่นได้เช่น การติดตามเป้าหมายจากยานยนต์ทั้งทางบก ทางน้ำ อีกทั้งยังสามารถประยุกต์ใช้ในหุ่นยนต์กู้ภัยได้ด้วย

1.2 วัตถุประสงค์ของการวิจัย

1. ปรับปรุงอัลกอริทึมและเพิ่มความสามารถให้กับระบบปรับเสถียรของภาพวีดิทัศน์สำหรับระบบอากาศยานไร้คนขับ
2. พัฒนาระบบกล้องวีดิทัศน์ติดตามเป้าหมายแบบเวลาจริงบนระบบข้างต้นที่พัฒนาขึ้นร่วมกับ สกว.

1.3 เป้าหมายและขอบเขตของการวิจัย

1. พัฒนาระบบติดตามภาพเป้าหมายที่ทำงานในเวลาจริง และทนทานต่อการรบกวน บนบอร์ด DSP
2. พัฒนาระบบควบคุมกล้องแบบเงยส่ายให้ติดตามเป้าหมาย เพื่อไม่ให้เป้าหมายหลุดจากจอภาพ

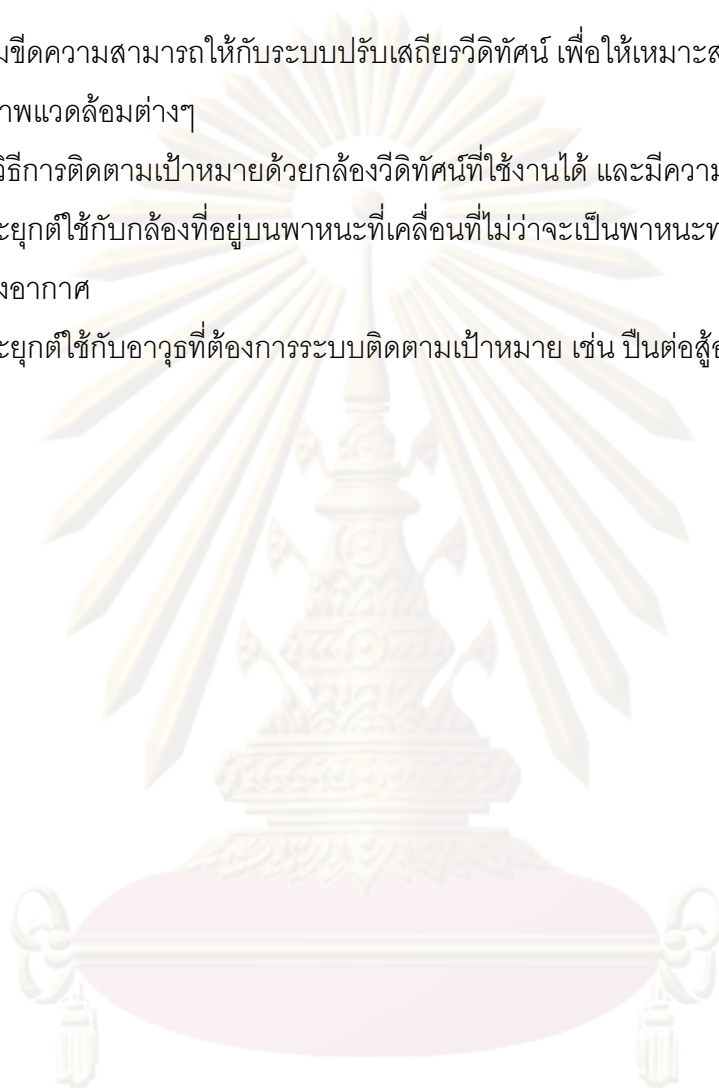
1.4 ขั้นตอนและวิธีดำเนินการวิจัย

1. กำหนดขอบเขต ตั้งสมมุติฐานของสภาพแวดล้อมและขีดจำกัดต่างๆ ในการใช้งาน
2. เก็บข้อมูลภาพจากสภาพแวดล้อมตามที่กำหนดไว้ในข้อ 1
3. ทำการวิเคราะห์ วิจัยบนคอมพิวเตอร์โดยใช้ โปรแกรมวิเคราะห์ เช่น MATLAB และภาษา C++ ร่วมกับคลังรหัสโปรแกรม (program library) OpenCV
4. ทำการพัฒนาโปรแกรมสำหรับระบบที่ใช้บนบอร์ด DSP
5. ทดลองเพื่อหาผลลัพธ์ในภาคสนาม
6. เปรียบเทียบและวิเคราะห์ผล

7. จัดทำรายงานฉบับสมบูรณ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เพิ่มขีดความสามารถให้กับระบบปรับเสถียรวิถีทัศนียภาพ เพื่อให้เหมาะสมกับการใช้งานได้ในสภาพแวดล้อมต่างๆ
2. ได้วิธีการติดตามเป้าหมายด้วยกล้องวิถีทัศนียภาพที่ใช้งานได้ และมีความเชื่อถือได้
3. ประยุกต์ใช้กับกล้องที่อยู่บนพาหนะที่เคลื่อนที่ไม่ว่าจะเป็นพาหนะทางบก ทางเรือ หรือทางอากาศ
4. ประยุกต์ใช้กับอาวุธที่ต้องการระบบติดตามเป้าหมาย เช่น ปืนต่อสู้อากาศยาน เป็นต้น



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

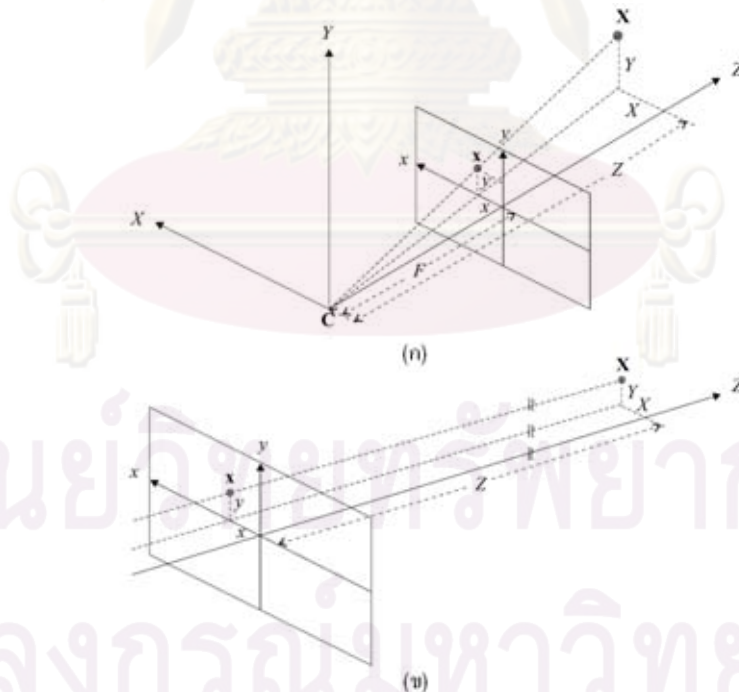
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีพื้นฐานและทฤษฎีอื่นๆ รวมถึงงานวิจัยอื่นๆ ที่เกี่ยวข้องกับงานวิจัยนี้

2.1 ระบบกล้องวิดิทัศน์ (camera system)

2.1.1 แบบจำลองกล้องวิดิทัศน์

[1] การทำงานของกล้องถ่ายภาพจะทำการฉายภาพ (แสง) ที่มาจากแหล่งกำเนิดหรือจากการสะท้อนแสงของวัตถุที่เป็น 3 มิติ มาสู่ฉากของกล้อง (camera image plane) ซึ่งเป็นแบบ 2 มิติ ดังนั้นจึงต้องกำหนดแบบจำลองของกล้อง (camera model) เพื่ออธิบายการทำงานและใช้ในการคำนวณ มีแบบจำลองหลายรูปแบบที่ใช้บรรยายการสร้างภาพ เช่น แบบจำลองกล้องรูเข็ม (pinhole camera model) แบบจำลองกล้อง CAHV (CAHV camera model) เป็นต้น แต่ในที่นี้จะพิจารณาเฉพาะแบบจำลองกล้องรูเข็มเท่านั้น



รูปที่ 2.1 (ก) การฉายภาพแบบทัศนมิติ (ข) การฉายภาพแบบขนาน

แบบจำลองกล้องรูเข็ม เป็นแบบจำลองที่นิยมใช้กันอย่างมากมาใช้ในการวิเคราะห์การสร้างภาพและงานประยุกต์ใน 3 มิติ แสดงได้ดังรูปที่ 2.1

2.1.1.1 การฉายภาพแบบทัศนมิติ (perspective projection)

แบบจำลองการฉายภาพนี้จะใช้เมื่อวัตถุอยู่ใกล้กับกล้อง ที่จุดพิกต์วัตถุในพิกต์ 3 มิติ สามารถฉายลงมายังจุดในระนาบภาพ (imaging plane) 2 มิติได้ดังสมการที่ 2.1 โดยมีจุด c เป็นจุดศูนย์กลางในการฉายภาพ (center of projection)

$$x = F \frac{X}{Z}, \quad y = F \frac{Y}{Z} \quad (2.1)$$

โดยที่ F คือ ระยะโฟกัส (focal length)

2.1.1.2 การฉายภาพแบบขนาน (parallel projection หรือ orthogonal projection)

เป็นแบบจำลองเมื่อวัตถุอยู่ห่างไกลจากกล้องมาก ซึ่งถือว่าเป็นระยะอนันต์ และถือว่าแสงจากวัตถุจะขนานกับรูกล้องและตั้งฉากกับฉากรับภาพ จะได้ความสัมพันธ์ดังสมการที่ 2.2

$$x = X, \quad y = Y \quad (2.2)$$

เนื่องจากการถ่ายภาพจากอากาศยานมักจะมีระยะห่างจากเป้าหมายมาก ดังนั้นแบบจำลองที่ใช้ในการคำนวณจะใช้แบบการฉายภาพแบบขนานเป็นหลักอีกทั้งเป้าหมายที่ติดตามมักจะเคลื่อนที่ที่ความเร็วต่ำเมื่อเทียบกับความเร็วอากาศยาน ดังนั้นจึงถือได้ว่าเป้าหมายอยู่นิ่งเมื่อเทียบกับฉากหลัง (background)

2.1.2 การเคลื่อนที่ของกล้องวิดีโอทัศน์

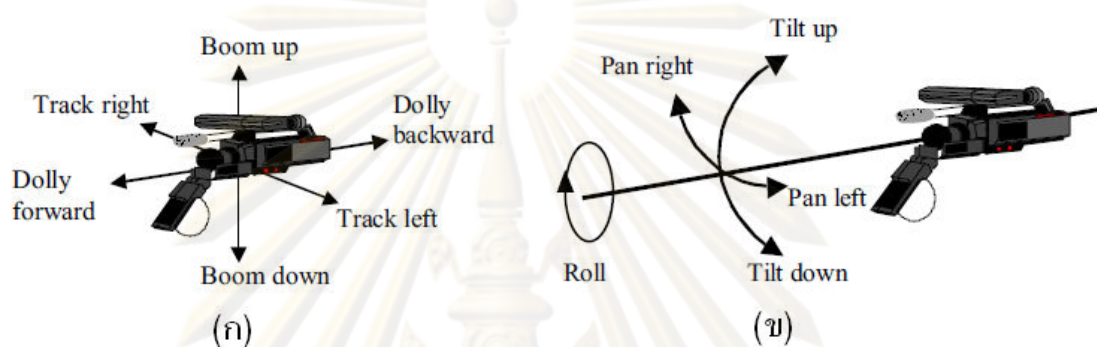
การเคลื่อนที่ของกล้องวิดีโอทัศน์ [1] โดยทั่วไปจะแบ่งออกได้เป็น 2 รูปแบบ คือ

2.1.2.1 การเคลื่อนที่ของกล้องที่ตำแหน่งศูนย์กลางการฉายภาพมีการเคลื่อนที่

ดังรูปที่ 2.2 (ก) ประกอบด้วย การติดตาม (track) เป็นลักษณะการเคลื่อนที่กล้องในลักษณะการเลื่อนขนาน (translation) ตามแนวแกนนอนของระนาบภาพ การยก (boom) เป็นลักษณะการเคลื่อนที่กล้องในลักษณะการเลื่อนขนานตามแนวแกนตั้งของระนาบภาพ และ การดอลลี (dolly) เป็นลักษณะการเคลื่อนที่กล้องในลักษณะการเลื่อนขนานตามแนวแกนเชิงแสง

2.1.2.2 การเคลื่อนที่ของกล้องที่ตำแหน่งศูนย์กลางการฉายภาพไม่เปลี่ยนแปลง

ดังรูปที่ 2.2 (ข) ประกอบด้วย การส่าย (pan) เป็นลักษณะการเคลื่อนที่กล้องโดยยึดแกนตั้ง (vertical Axis) เป็นแกนหมุนรอบ การก้มเงย (tilt) เป็นลักษณะการเคลื่อนที่กล้องโดยยึดแกนนอน (horizontal axis) เป็นแกนหมุนรอบ และการหมุน (roll) เป็นลักษณะการเคลื่อนที่กล้องโดยยึดแกนเชิงแสง (optical axis) เป็นแกนหมุนรอบ



รูปที่ 2.2 การเคลื่อนที่ของกล้อง

2.1.3 การซูม (zoom)

กำหนดให้ F และ F' เป็นความยาวโฟกัสก่อนและหลังการซูม จะได้สมการการเคลื่อนที่ที่เกิดจากการซูมดังสมการที่ 2.3

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \rho \cdot x \\ \rho \cdot y \end{bmatrix} \quad (2.3)$$

โดยที่ $\rho = \frac{F'}{F}$ คือ ตัวประกอบการซูม (zoom factor) ดังนั้นได้สมการสนามการเคลื่อนที่ดังสมการที่ 2.4

$$\begin{bmatrix} d_x(x, y) \\ d_y(x, y) \end{bmatrix} = \begin{bmatrix} (1 - \rho)x \\ (1 - \rho)y \end{bmatrix} \quad (2.4)$$

2.2 ระบบปรับเสถียรวิดิทัศน์

เป็นระบบที่ปรับให้ภาพที่ไปปรากฏบนจอภาพนิ่งไม่สั่นไหวไปตามการสั่นไหวจากสภาพแวดล้อมที่กล้องติดตั้งอยู่

2.2.1 ระบบปรับเสถียรทางกล

ระบบนี้มักรวมกับระบบปรับหันทิศของกล้อง เรียกว่ากิมบอลส์ (gimbals) ดังตัวอย่างรูปที่ 2.3 ในการปรับเสถียรจะใช้อุปกรณ์ทางกล เช่น ให้คาน สปริง เป็นตัวหน่วง (damper) เพื่อลดการสั่นไหวของ เปรียบเสมือนตัวกรองผ่านต่ำ (low pass filter) ในระบบไฟฟ้า ทำให้กล้องสั่นไหวน้อยลงมาก จนพนักงานไม่สามารถสังเกตเห็นได้ ระบบปรับเสถียรทางกลยังแบ่งได้ เป็น 2 แบบหลักๆ คือ



รูปที่ 2.3 กิมบอลส์สำหรับงานถ่ายภาพวีดิทัศน์ทางอากาศ

2.2.1.1 ระบบแพสซีฟ (passive stabilizer)

โดยใช้สปริงหรือคานในการหน่วง เพื่อลดทอนความถี่สูงทางกล ทำให้ภาพไม่สั่นไหวไปมา ระบบนี้เป็นระบบที่ง่ายไม่ซับซ้อน แต่ต้องการการบำรุงรักษามาก ต้องปรับแต่งมาก และมีน้ำหนักมาก เหมาะกับพาหนะขนาดใหญ่

2.2.1.2 ระบบแอ็กทีฟ (active stabilizer)

ระบบนี้จะใช้เซอร์โวมอเตอร์หมุนแก้มการสั่นไหวผ่านระบบควบคุมที่มีเซนเซอร์ (sensor) เป็นไจโรสโคป และตัววัดความเร่ง โดยระบบจะตรวจจับการสั่นไหวโดยเซนเซอร์ดังกล่าว ข้อมูลจะถูกส่งให้ระบบควบคุมแบบป้อนกลับ (feedback system) เพื่อทำการควบคุมเซอร์โวมอเตอร์เพื่อแก้มการสั่นไหว โดยมอเตอร์จะหมุนในทิศตรงกันข้ามกับการสั่นไหวหรือมีการเคลื่อนที่กลับเฟส 180 องศาของการสั่นไหวนั่นเอง ซึ่งเสมือนการหักล้างทางเฟส (phase cancellation) ข้อดีของระบบนี้ จะมีความแม่นยำสูงกว่าระบบแรก น้ำหนักเบา แต่มีข้อเสียคือ ต้นทุนสูงและมีความซับซ้อนมาก

นอกจากนี้ ยังมีระบบที่นำทั้งสองระบบมารวมกัน (hybrid) ทำให้ระบบมีประสิทธิภาพมากขึ้น ปัจจุบันระบบปรับเสถียรมักเป็นระบบทางกล โดยมีใช้ในอากาศยานทั้งที่มีนักบินและไร้นักบิน แต่เนื่องจากระบบมีน้ำหนักมากและมีขนาดใหญ่ ดังนั้นจึงนิยมใช้ในอากาศยานที่ไร้นักบิน หรืออากาศยานไร้นักบินขนาดใหญ่ เช่น Predator หรือ Global Hawk เป็นต้น

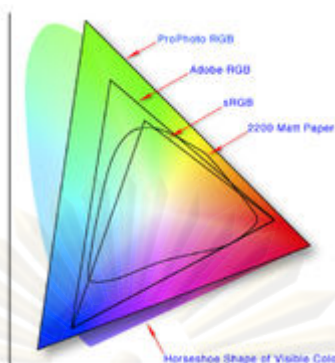
2.2.2 ระบบปรับเสถียรแบบอิเล็กทรอนิกส์

แทนที่จะลดการสั่นไหวของตัวกล้อง ระบบนี้จะนำภาพที่ถ่ายได้จากกล้องมาประมวลเพื่อทำการปรับเสถียร โดยมีหลักการดังนี้ เริ่มต้นจะนำภาพเฟรมแรกเป็นภาพอ้างอิง เมื่อได้ภาพจากเฟรมถัดมาซึ่งภาพอาจไม่อยู่ในตำแหน่งเดียวกันกับภาพเฟรมแรกเนื่องจากการสั่นไหว ระบบจะทำการตรวจสอบฉากหลังเทียบกับภาพจากเฟรมแรก และจะทำการย้ายภาพใหม่มาแสดง (display) ในตำแหน่งที่ใกล้เคียงกับเฟรมแรกมากที่สุด ในมุมมองของพนักงานจะเสมือนว่าภาพไม่ได้เคลื่อนไหว กระบวนการนี้จะดำเนินการต่อเนื่องกันไป โดยคงสถานะการเคลื่อนที่ในการเคลื่อนย้าย (translation) ไว้ ดังนั้นระบบเสมือนเป็นตัวกรองผ่านต่ำ เพื่อลดการสั่นไหวจากภาพที่ถ่ายได้จากกล้อง ข้อดีของระบบคือ มีน้ำหนักเบา ขนาดเล็ก ถ้าผลิตเป็นจำนวนมากจะมีต้นทุนต่ำ ปัจจุบันมีใช้ในกล้องถ่ายภาพดิจิทัล กล้องวิดีโอแบบพกพา เป็นต้น เนื่องจากข้อดีดังกล่าว ระบบนี้จึงเหมาะกับอากาศยานไร้นักบินขนาดเล็ก หรือแบบพกพา

ปัจจุบันทางภาควิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ได้พัฒนาระบบปรับเสถียรดังกล่าวให้กับโครงการอากาศยานไร้นักบิน ซึ่งงานวิจัยนี้จะพัฒนาระบบติดตามเป้าหมายเพื่อต่อ ยอดความสามารถให้กับระบบปรับเสถียรนี้

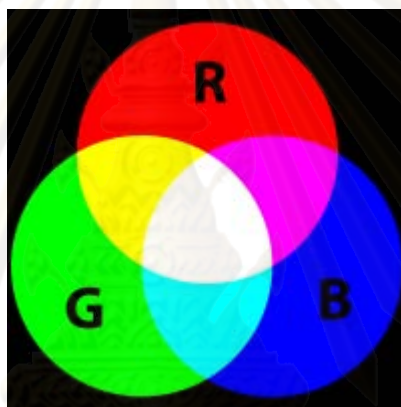
2.3 ปริภูมิสี

[2] หมายถึงแบบจำลองของสี (color model) และรวมถึงขอบเขตของสี (absolute color space) แบบจำลองสี คือการใช้คณิตศาสตร์เพื่อเป็นตัวแทนค่าของสี สีหนึ่งสี มักจะแทนด้วยชุดตัวเลข 3 ชุด ดังเช่น แบบจำลองสี RGB หรือ แบบจำลองสี CMYK จะแทนด้วยตัวเลข 4 ชุด เป็นต้น การกำหนดปริภูมิสีนั้นนอกจากจะกำหนดแบบจำลองสีแล้วยังกำหนดขอบเขตสี เช่นปริภูมิสี Adobe RGB จะมีขอบเขตกว้างกว่า ปริภูมิสี sRGB ซึ่งปริภูมิสีทั้งสองมีแบบจำลองสีแบบเดียวกันคือ RGB ดังรูปที่ 2.4



รูปที่ 2.4 ตัวอย่างปริภูมิสีที่อยู่ในแบบจำลองสี RGB

2.3.1 ปริภูมิสี RGB



รูปที่ 2.5 แสดงการผสมแม่สีแสงแดงเขียวน้ำเงินในปริภูมิสี RGB

[3] เป็นการผสมสีของแม่สีแสง แดง เขียว และ น้ำเงิน ในสัดส่วนต่างๆ เพื่อให้เกิดแสงสีอื่นๆ ระบบสี RGB จะใช้ในการรับรู้ (sensing) และแสดงผลในรูปภาพในระบบอิเล็กทรอนิกส์ เช่น โทรศัพท์ จอคอมพิวเตอร์ ฯลฯ ระบบสีแบบ RGB เป็นระบบที่ขึ้นกับอุปกรณ์หมายความว่า อุปกรณ์ต่างกัน จะรับรู้ และแสดงผล ค่า RGB ที่ต่างกัน ซึ่งค่าเหล่านี้จะแปรตามผู้ผลิตอุปกรณ์ หรือแม้แต่อุปกรณ์เดียวกันแต่คนละช่วงเวลา ดังนั้นค่าข้อมูลเดียวกันอาจแสดงสีที่แตกต่างกันได้ อุปกรณ์ที่รับรู้สีแบบ RGB ได้แก่ กล้องโทรทัศน์ (TV camera) กล้องวิดีโอ เครื่องสแกนภาพ กล้องดิจิทัล เป็นต้น ส่วนอุปกรณ์แสดงผลด้วยระบบ RGB ได้แก่ จอทีวี จอซีอาร์ที จอพลาสมา โทรศัพท์มือถือ เครื่องฉายภาพวิดีโอ (video projector) ฯลฯ

2.3.2 ปริภูมิสี YCbCr

ปริภูมิสีแบบ YCbCr [4] เป็นหนึ่งในหลายปริภูมิสี ปริภูมิสี YCbCr ถูกพัฒนามาจากปริภูมิสีแบบ YUV [5] และเป็นที่ยอมรับใช้ในการถ่ายทอดข้อมูลภาพในระบบวิดีโอและ

ภาพถ่ายดิจิทัล โดย Y คือค่าความสว่าง (luminance) Cb คือ ความแตกต่างน้ำเงิน (blue-difference) ส่วน Cr คือความแตกต่างแดง (red-difference) ประโยชน์อย่างหนึ่งของปริภูมิสี YCbCr เกิดจากธรรมชาติการมองเห็นของมนุษย์ โดยมนุษย์จะมีความไวในการมองเห็นแถบเทาหรือสี ขาวดำได้ดีกว่าแถบสีอื่นๆ ทั้งนี้เพราะมนุษย์มีเซลล์ประสาทรับแสงแบบรีด (rod) ไวต่อสีเทา มากกว่าเซลล์ประสาทรับสีแบบโคน (cone) ที่ไวต่อการรับสีมาก ดังนั้นจึงสามารถลดข้อมูลสีหรือ แถบสัญญาณสี (chroma bandwidth) ขององค์ประกอบสี (chroma component) Cb และ Cr ได้ ซึ่งอาจทำได้โดย ลดจำนวนสุ่ม (subsampling) บีบอัด (compressed) องค์ประกอบสีเหล่านี้ YCbCr ถูกกำหนดในมาตรฐาน ITU-R BT.601 โดยมีสูตรแปลงสีจากปริภูมิสี RGB ดังนี้ และยัง กำหนดให้ค่าขององค์ประกอบ Y อยู่ในช่วง 16-235 ค่า Cb และ Cr อยู่ในช่วง 16-240 ดังนั้น องค์ประกอบของระบบ YCbCr จึงสามารถบรรจุในข้อมูลขนาด 8 บิตได้

การแปลงปริภูมิสีจาก RGB ไปเป็น YCbCr

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (2.5)$$

การแปลงปริภูมิสีจาก YCbCr กลับเป็น RGB

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 0 & 1.596 \\ 1.164 & -0.392 & -0.813 \\ 1.164 & 2.017 & 0 \end{bmatrix} \begin{bmatrix} Y-16 \\ Cb-128 \\ Cr-128 \end{bmatrix} \quad (2.6)$$

2.3.3 ปริภูมิสี HSV



รูปที่ 2.6 ปริภูมิสี HSV

เป็นปริภูมิสีที่แสดงความสัมพันธ์ในการรับรู้สีของมนุษย์ได้ละเอียดกว่าปริภูมิสีแบบ RGB ปริภูมิสี HSV [6] จะแสดงสีในแบบรูปกรวยหงาย ดังรูปที่ 2.6 โดยแกนกลางของกรวย

จะเป็นส่วนของ V หรือค่า (value) แสดงความสว่าง ซึ่งใกล้เคียงกับ Y ในระบบ YCbCr โดย
 ด้านล่างของกรวยจะเป็นสีดำด้านบนจะเป็นสีขาวระหว่างกลางจะเป็นเทา ส่วนมุมรอบแกนกลาง
 จะเป็นตัวแทนมุมของสีเรียกว่า hue ส่วนระยะระหว่างแกนกลางไปยังขอบของสี เรียกว่า
 saturation ซึ่งแสดงความบริสุทธิ์ของสีนั้น ถ้าค่านี้น้อยสีจะมีความซีดมาก หรือใกล้สีเทา แต่ถ้าค่า
 นี้มีค่ามาก สีจะสดหรือเข้มขึ้นมาก HSV นิยมใช้ในงานกราฟิกส์บนคอมพิวเตอร์ โดยที่ผู้ใช้จะจัด
 วางสีต่างๆ ไว้รอบๆจานสี ดังนั้นส่วน hue จะเหมือนวงของจานสีที่เตรียมไว้ ส่วน saturation จะ
 แสดงความสดหรือซีดของสีนั้นๆ จากนั้นค่า value หรือความสว่างจะถูกเลือกอีกที เพื่อให้ได้
 องค์ประกอบของสีตามต้องการ

การแปลงปริภูมิสีจาก RGB ไปเป็น HSV

ให้ R,G,B มีค่าระหว่าง [0,1] และฟังก์ชัน $\max(R,G,B)$ ให้ค่าของตัวแปรที่มีค่า
 มากที่สุด และฟังก์ชัน $\min(R,G,B)$ ให้ค่าต่ำสุดจะได้

$$H = \begin{cases} 0, & \max = \min \\ \left(60 \times \frac{G - B}{\max - \min} + 360 \right) \bmod 360, & \max = R \\ 60 \times \frac{B - R}{\max - \min} + 120, & \max = G \\ 60 \times \frac{R - G}{\max - \min} + 240, & \max = B \end{cases} \quad (2.7)$$

โดย H คือมุมสี มีค่า [0,360) องศา

$$S = \begin{cases} 0, & \max = 0 \\ \frac{\max - \min}{\max}, & \text{otherwise} \end{cases} \quad (2.8)$$

$$V = \max \quad (2.9)$$

ทั้งนี้ค่า S และ V มีค่าอยู่ระหว่าง [0,1]

การแปลงปริภูมิสีจาก HSV กลับเป็น RGB

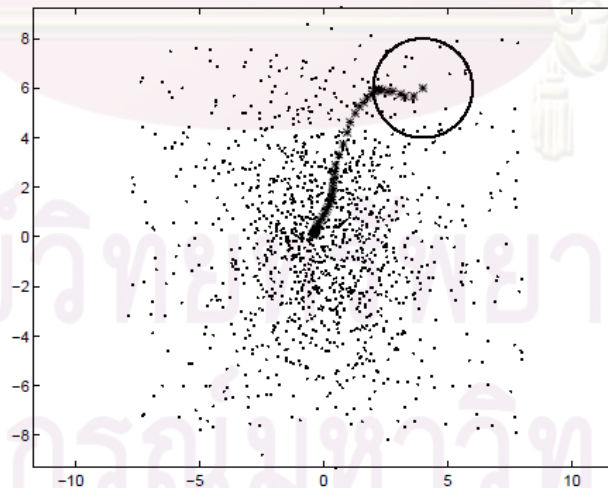
$$\begin{aligned}
 h_i &= \left\lfloor \frac{H}{60} \right\rfloor \bmod 6 \\
 f &= \frac{H}{60} - \left\lfloor \frac{H}{60} \right\rfloor \\
 p &= V \times (1 - S) \\
 q &= V \times (1 - f \times S) \\
 t &= V \times (1 - (1 - f) \times S)
 \end{aligned} \tag{2.10}$$

$$(R, G, B) = \begin{cases} (v, t, p), h_i = 0 \\ (q, v, p), h_i = 1 \\ (p, v, t), h_i = 2 \\ (p, q, v), h_i = 3 \\ (t, p, v), h_i = 4 \\ (v, p, q), h_i = 5 \end{cases} \tag{2.11}$$

ทั้งนี้ค่า R,G,B,S และ V อยู่ระหว่าง [0,1] ส่วนค่า H อยู่ระหว่าง [0,360)

2.4 มินชิฟต์

มินชิฟต์ (mean shift) ถูกพัฒนาเพื่อนำมาใช้ในทางสถิติการหาค่าฐานนิยม (mode) ของการกระจาย (distribution) ความน่าจะเป็น (probability) โดยใช้เทคนิคการไล่ความลาดชันของความหนาแน่น (climbing density gradients) [7]



รูปที่ 2.7 แสดงการหาความหนาแน่นของการกระจายด้วยวิธีมินชิฟต์

ให้เซต $\{\mathbf{x}_i\}_{i=1,\dots,n}$ คือจำนวนจุดที่มีขนาด d มิติในปริภูมิ R^d โดยมีเคอร์เนล (kernel) เป็น $K(\mathbf{x})$ และหน้าต่างที่มีรัศมี h จะสามารถเขียนสมการการประมาณการกระจายความหนาแน่น (density distribution estimation) ได้ดังนี้ [8]

$$\hat{f}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \quad (2.12)$$

ถ้าเคอร์เนลสามารถหาอนุพันธ์ได้ จะสามารถหาการประมาณความลาดชันของความหนาแน่นในรูปของการประมาณความลาดชันของเคอร์เนลได้ดังนี้

$$\hat{\nabla} f(\mathbf{x}) \equiv \nabla \hat{f}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^n \nabla K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \quad (2.13)$$

ถ้า $K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$ อยู่ในรูป $ck\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)$ และ $g(\mathbf{x}) = -k'(\mathbf{x})$

ในที่สุดจะได้

$$\nabla \hat{f}_k(\mathbf{x}) = \frac{2}{nh^{d+2}} \left[\frac{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)} \left[\frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x} \right] \right] \quad (2.14)$$

พิจารณาวงเล็บสุดท้าย ซึ่งเป็นเวกเตอร์มินิพท์แบบไต่ความชัน

$$M = \left[\frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x} \right] \quad (2.15)$$

ซึ่งสมการนี้จะถูกนำไปใช้ในการติดตามเป้าหมายแบบเคอร์เนลเบสส์ ดังจะกล่าวโดยละเอียดในบทที่ 3

นอกจากนี้ยังมีวิธีมินิพท์แบบทั่วไป [9] โดยมีหลักการดังนี้

ถ้าข้อมูลเป็นเซตจำกัด (finite set) S ที่มีขนาด n มิติ ในปริภูมิยูคลิดเดียน X และ K เป็นเคอร์เนลแบบแบนราบที่มีฟังก์ชันคุณลักษณะของ λ -ball ใน X [9]

$$K(x) = \begin{cases} 1, & \|x\| \leq \lambda \\ 0, & \|x\| > \lambda \end{cases} \quad (2.16)$$

จะได้ค่าเฉลี่ยของตัวอย่างที่ $x \in X$ ดังนี้

$$m(x) = \frac{\sum_{s \in S} K(s-x)s}{\sum_{s \in S} K(s-x)} \quad (2.17)$$

โดยส่วนต่าง $m(x)-x$ จะเรียกว่า เวกเตอร์มีนชิฟต์ (mean shift) ดังปรากฏใน [7] และการเคลื่อนที่ของจุดข้อมูลที่เป็นเป็นค่าเฉลี่ยจะเรียกว่าอัลกอริธึมมีนชิฟต์ โดยในการเคลื่อนแต่ละครั้งจะทำให้ $s \leftarrow m(s)$ ซึ่งในที่สุดจะนำไปสู่ค่าสูงสุด (peak) ของการกระจายความหนาแน่นของเซตข้อมูลนั้น ดังรูปที่ 2.7 วิธีมีนชิฟต์แบบนี้จะถูกนำไปประยุกต์ในการติดตามเป้าหมายโดยวิธีมีนชิฟต์แบบทั่วไปดังจะกล่าวต่อไปในบทที่ 3

2.5 ระยะห่าง Bhattacharyya

ปัญหาสำคัญอย่างหนึ่งในงานวิสัยทัศน์คอมพิวเตอร์ (computer vision) คือการวัดความคล้ายกันของการกระจายความหนาแน่นของคุณลักษณะ (feature density distribution) ระยะห่าง Bhattacharyya เป็นวิธีวัดความคล้ายกันที่ไม่ซับซ้อน มีประสิทธิภาพสูง และที่สำคัญไม่ก่อให้เกิดปัญหาเอกฐาน (singularity problem) ที่เกิดขึ้นกับวิธีวัดอื่นๆ เช่น Chi-square เป็นต้น

กำหนดให้ $p(i)$ และ $q(i)$ เป็นตัวแทนประชากรที่มี N ลำดับโดยมีความน่าจะเป็นเท่ากับ $p(i=1), \dots, p(i=N)$ และ $q(i=1), \dots, q(i=N)$ ตามลำดับ เนื่องจาก p และ q เป็นการกระจายความน่าจะเป็น ดังนั้น $\sum_{i=1}^N p(i) = 1$ และ $\sum_{i=1}^N q(i) = 1$ ดังนั้นสัมประสิทธิ์ Bhattacharyya [10] เป็นการวัดแบบลู่ออก (divergence-type measure) ระหว่างการกระจาย p และ q จะถูกนิยามไว้ดังนี้

$$\rho(p, q) = \sum_{i=1}^N \sqrt{p(i)q(i)} \quad (2.18)$$

ถ้าแทนความหมายการกระจายด้วยเวกเตอร์ขนาด N มิติ $[\sqrt{p(1)}, \dots, \sqrt{p(N)}]^T$
และ $[\sqrt{q(1)}, \dots, \sqrt{q(N)}]^T$

โคซายด์ของมุมระหว่างการกระจายทั้งสองจะเท่ากับ

$$\cos \theta = \sum_{i=1}^N \sqrt{p(i)q(i)} \quad \text{โดย } 0 \leq \cos \theta \leq 1 \quad (2.19)$$

ส่วนระยะห่าง Bhattacharyya ถูกนิยามไว้ดังนี้

$$d(p, q) = \sqrt{1 - \rho(p, q)} \quad (2.20)$$

การประยุกต์ใช้สัมประสิทธิ์และระยะห่าง Bhattacharyya จะแสดงโดยละเอียด
ในบทที่ 3

2.6 งานวิจัยที่เกี่ยวข้อง

G.R.Bradski และคณะ [11] นักวิจัยทำการพัฒนาการติดตามหน้ามนุษย์ที่ทำงานกับคอมพิวเตอร์เพื่อนำตำแหน่งและการหันหน้าของมนุษย์เพื่อการติดต่อ (interface) กับเครื่องคอมพิวเตอร์ นักวิจัยใช้วิธีมีนชิฟต์ (mean shift) ในการหาตำแหน่งของหน้าโดยหาค่าฐานานิยมของสีของผิวหน้ามนุษย์ โดยการตรวจสอบจะใช้วิธีมีนชิฟต์จากข้อมูลการประมาณการกระจายความน่าจะเป็นของสี ซึ่งจะกล่าวถึงในบทถัดไป จากนั้นยังได้พัฒนาการตรวจสอบขนาดและมุมการเอียงของใบหน้าโดยเรียกว่า Continuous Adaptive Mean Shift หรือ CAMSHIFT นักวิจัยได้ใช้ปริภูมิสีแบบ HSV โดยใช้เฉพาะองค์ประกอบ H หรือ hue ในการแยกสีผิวมนุษย์จากวัตถุอื่นๆ โดยให้เหตุผลของการใช้เฉพาะองค์ประกอบ hue เพราะสีมนุษย์ จะมี hue เหมือนกันไม่ว่าจะเป็นคนผิวขาว ผิวดำ หรือ ผิวเหลือง แต่ต่างกันตรง saturation ซึ่งวิธีการติดตามเป้าหมายสามารถนำมาประยุกต์ได้ในวิทยานิพนธ์นี้ โดยเพิ่มองค์ประกอบ saturation

D. Comaniciu และคณะ [12,13] ได้ทำการวิจัยเพื่อติดตามเป้าหมาย ด้วยวิธีมีนชิฟต์แบบคอร์เนลเบลสตัดเพื่อติดตามคุณลักษณะของเป้าหมาย โดยใช้ฟังก์ชันความคล้ายคลึงด้วยวิธี ระยะห่าง Bhattacharyya คุณลักษณะที่ใช้คือสีของเป้าหมาย สิ่งที่แตกต่างกันจากงานวิจัยของ Bradski คือจะใช้มีนชิฟต์โดยการเปรียบเทียบฮิสโตแกรม ซึ่งทำให้มีคุณสมบัติในการเปรียบเทียบรูปแบบแฝงอยู่ แทนที่จะเป็นน้ำหนักจากคุณลักษณะ(สี)เพียงอย่างเดียว ทำให้การ

ติดตามเป้าหมายได้ถูกต้องมากกว่า มีความทนทานสูงกว่า แต่ต้องแลกมาด้วยการคำนวณที่มากกว่าเช่นกัน

X. Chen และคณะ [14] ที่วิจัยนี้ได้ประยุกต์อัลกอริทึมสองอัลกอริทึมในการติดตามเป้าหมายที่เคลื่อนที่ โดยใช้วิธีแยกแยะวัตถุ (object segmentation) ของภาพเป้าหมายจากฉากหลัง จากนั้นทำการติดตามเป้าหมายด้วยวิธีมินชิฟต์ เช่นเดียวกับ D.Comaniciu ทั้งนี้ นักวิจัยให้เหตุผลว่า ส่วนฉากหลังอาจมีผลกระทบต่อติดตามเป้าหมาย โดยเฉพาะอย่างยิ่งเมื่อเป้าหมายมีลักษณะพิเศษ เช่น เรียวยาว ทำให้ปริมาณจุดของฉากหลังมีค่าใกล้เคียงกับปริมาณจุดของเป้าหมาย เมื่อเป้าหมายเคลื่อนที่ไปอาจเกิดการเปลี่ยนฉากหลัง เช่นจากถนนไปพื้นหญ้า ทำให้สูญเสียการติดตามเป้าหมายได้ แต่เนื่องจากข้อกำหนดที่ต้องการการติดตามเป้าหมายแบบเวลาจริงดังนั้นการแยกแยะวัตถุจะต้องเป็นอัลกอริทึมที่ง่ายทำการประมวลผลได้อย่างรวดเร็ว ซึ่งนักวิจัยได้ใช้วิธีการ ตัดกราฟ (graph cut) โดยเมื่อแยกแยะแล้วจะใส่ค่าศูนย์แทนที่บริเวณที่เป็นฉากหลังแล้วทำการติดตามทำให้การติดตามมีความทนทานมากขึ้น

N. Oshima และคณะ [15] งานวิจัยนี้มีจุดประสงค์เพื่อตรวจจับเป้าหมายที่เคลื่อนที่โดยอัตโนมัติ โดยใช้วิธีการลบภาพเพื่อแยกฉากหลังออกจากภาพ จากนั้นใช้วิธีการจัดกลุ่ม (clustering) แบบบลิค เพื่อแยกเป้าหมายจากการรบกวน (noise) และใช้การติดตามเป้าหมายตามแนวทางของ D.Comaniciu ในการติดตามเป้าหมาย ข้อกำหนดในงานวิจัยนี้ คือ ตรวจสอบเป้าหมายที่เคลื่อนที่โดยอัตโนมัติ และใช้กล่องที่ติดตรงกับที่ ด้วยข้อกำหนดหลังนี้ทำให้การตรวจหาและติดตามเป้าหมายเป็นไปอย่างมีประสิทธิภาพ แต่ด้วยข้อจำกัดที่ให้กล่องติดตรง ดังนั้นงานวิจัยนี้จึงไม่เหมาะที่จะประยุกต์ใช้ในวิทยานิพนธ์นี้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

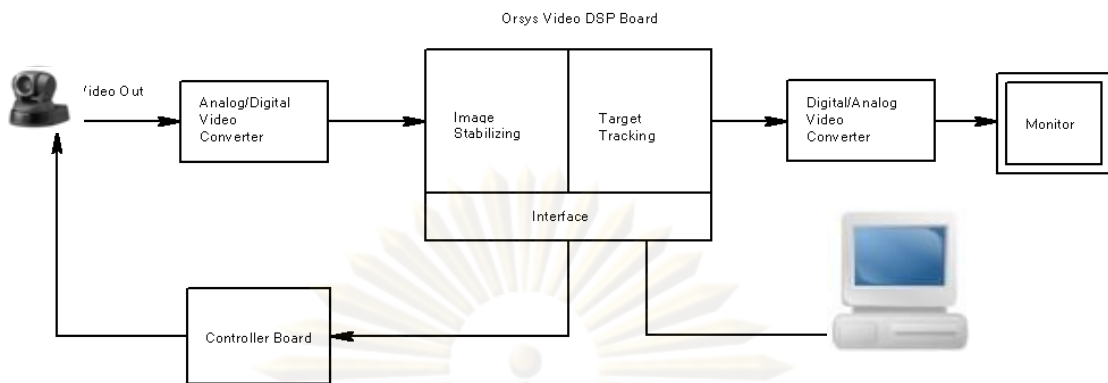
บทที่ 3

วิธีดำเนินการวิจัย

ในการปฏิบัติการถ่ายภาพทางอากาศของอากาศยานไร้คนขับ เป็นการปฏิบัติงานภายใต้สภาวะแวดล้อมที่ไม่เสถียรนั่นคือสภาวะที่มีการรบกวนทางกล (mechanical noise) สูง เช่น การสั่นจากเครื่องยนต์ การสั่นไหวจากสภาพอากาศ (turbulence) การรบกวนเนื่องจากการควบคุม เป็นต้น ดังนั้นกล้องถ่ายภาพจึงต้องมีระบบปรับเสถียรที่เหมาะสมและมีระบบการควบคุมที่ดี ซึ่งระบบดังกล่าวจะประกอบอยู่ในระบบปรับเสถียรหรือกิมบอลส์ (gimbals) ดังที่กล่าวมาแล้วในบทที่ 2 นอกจากนี้เพื่อให้การถ่ายภาพเป็นไปอย่างมีประสิทธิภาพระบบติดตามเป้าหมายจึงเป็นส่วนประกอบที่สำคัญ เนื่องจากระบบดังกล่าวจะช่วยลดภาระให้กับพนักงานในการติดตามเป้าหมาย โดยเฉพาะอย่างยิ่งเมื่อถ่ายภาพที่อัตราสูง วิทยานิพนธ์นี้มีเป้าหมายในการประยุกต์ระบบติดตามเป้าหมายที่ปฏิบัติการในเวลาจริง โดยระบบติดตามดังกล่าวจะต้องกระชับแต่มีความทนทาน ซึ่งอัลกอริทึมที่นำมาพิจารณาใช้คือมินชิฟต์ (mean shift) [7] มินชิฟต์ถูกพัฒนาเพื่อใช้ในการหาค่าฐานนิยมในการกระจายความหนาแน่นของความน่าจะเป็น (probability density distribution) ซึ่งต่อมาถูกนำมาประยุกต์ใช้งานทางการประมวลผลภาพ

3.1 ระบบติดตามเป้าหมาย

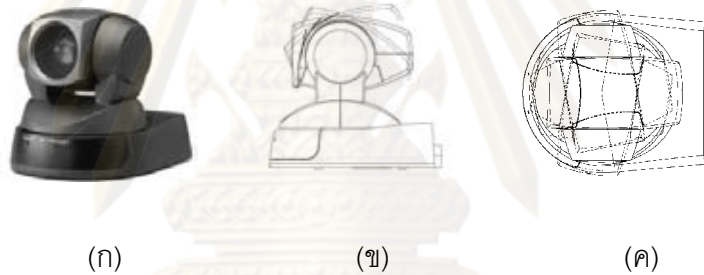
รูปที่ 3.1 แสดงถึงระบบติดตามภาพเป้าหมาย ในวิทยานิพนธ์นี้ ซึ่งประกอบด้วย กล้องแบบ สายก้มเงยซูม (pan-tilt-zoom) หรือ PTZ เป็นตัวถ่ายภาพเพื่อส่งไปประมวลผล และ ขณะเดียวกันก็เป็นอุปกรณ์ดำเนินการควบคุม (process control) กล้องในการสายก้มเงยซูมกล้องไปยังตำแหน่งที่ต้องการ ส่วนบอร์ดประมวลผลจะบอร์ดประมวลผลวีดิทัศน์ด้วยตัวประมวลผลสัญญาณดิจิทัล (digital signal processor) หรือ DSP บอร์ดประมวลผลจะทำการติดตามเป้าหมายที่ปรากฏในภาพที่ได้จากกล้องวีดิทัศน์ และทำการคำนวณการควบคุมและส่งผลการควบคุมไปยังกล้อง PTZ นอกจากนี้ระบบติดตามเป้าหมายจะทำการติดต่อกับพนักงานผ่าน คอมพิวเตอร์ส่วนบุคคล



รูปที่ 3.1 ผังของระบบติดตามเป้าหมาย

3.1.1 กล้องวิดีโอประเภทสายก้มเงยและซูม (Pan-Tilt-Zoom Camera, PTZ)

ในงานวิจัยนี้ใช้กล้องติดตามเป้าหมายเป็นกล้องวิดีโอ SONY รุ่น EVID-100P ดังรูปที่ 3.2 มีคุณสมบัติที่สำคัญเกี่ยวกับการสาย การก้มเงยและการซูม ดังนี้



รูปที่ 3.2 กล้องวิดีโอชนิดในการทดลอง (ก) กล้องวิดีโอ EVID-100 (ข) ลักษณะการก้มเงย (ค) ลักษณะการสาย

- ในการควบคุมการเคลื่อนที่ของกล้องอาศัยคำสั่ง (command) ควบคุมผ่านทางสายสัญญาณ RS-232

- สามารถซูมเชิงแสง (Optical zoom) ได้สูงสุด 10 เท่า

- แนวการสายสูงสุด ± 100 องศา

- แนวการก้มเงยสูงสุด ± 25 องศา

- ความเร็วสูงสุดในการเคลื่อนที่แบบสาย 300 องศา/วินาที

- ความเร็วสูงสุดในการเคลื่อนที่แบบก้มเงย 125 องศา/วินาที

- รองรับการเชื่อมต่อสัญญาณระหว่างกล้องกับส่วนประมวลผลได้ 2 รูปแบบ คือ แบบสายสัญญาณภาพรวม (composite video cable) และสายสัญญาณวีดิทัศน์แบบเอสวีดีโอ (S-video cable)

กล้อง PTZ ที่นำมาใช้ในงานวิจัยนี้ เป็นกล้องสำหรับงานเฝ้าตรวจทั่วไปไม่เหมาะกับงานในพาหนะที่เคลื่อนที่ แต่เนื่องจากระบบгимบอลล์ที่ใช้ในการตรวจการทางอากาศมีราคาสูงมาก ดังนั้นจึงจำเป็นต้องใช้กล้อง PTZ ในการจำลองการทำงาน (simulation) แทนระบบгимบอลล์

3.1.2 บอร์ดประมวลผลวีดิทัศน์

รูปที่ 3.3 เป็นผังบอร์ดประมวลผลทางวีดิทัศน์ของบริษัท Orsys รุ่น micro-line C6713Compact โดยมีคุณสมบัติดังนี้

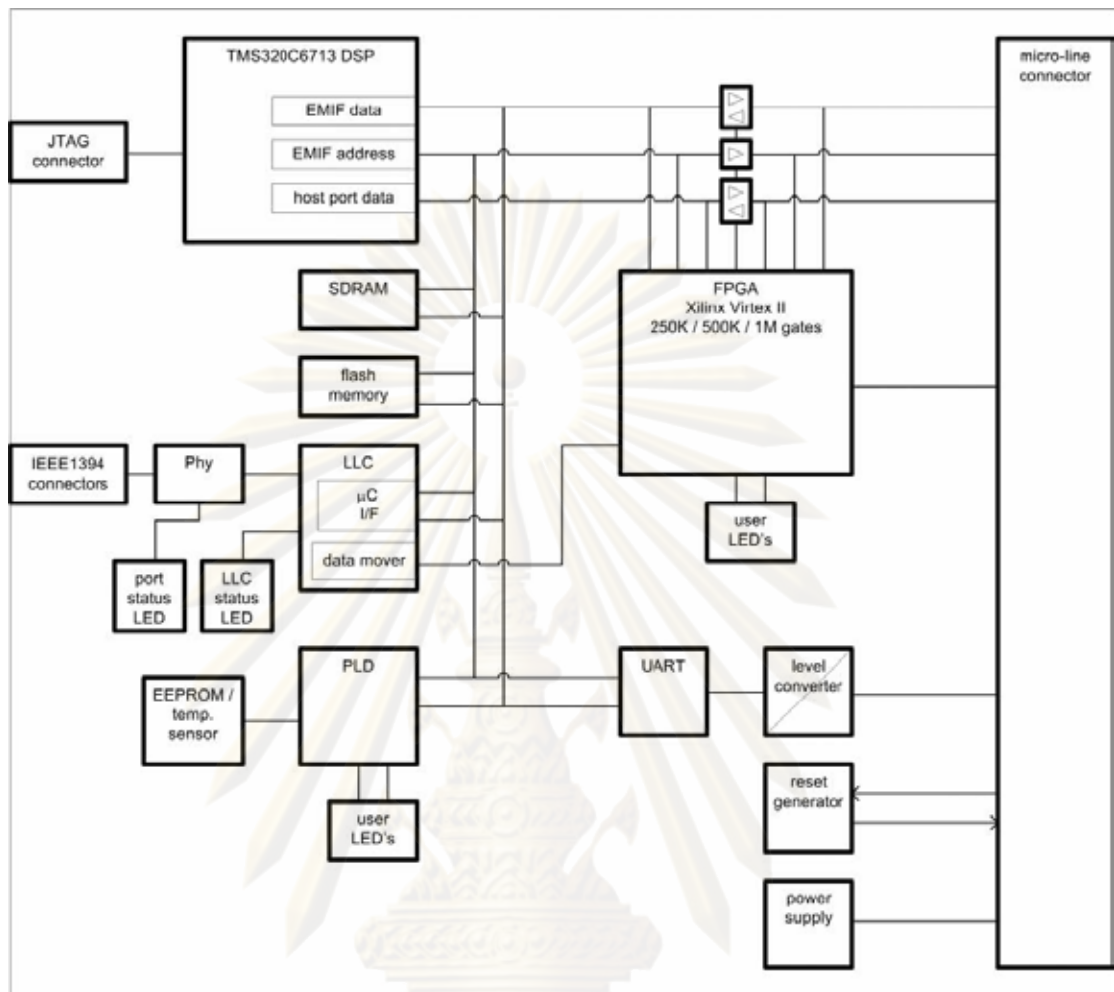
- ใช้ตัวประมวลผล DSP รุ่น TMS320C6713 ของบริษัทเท็กซัสอินสตรูเมนต์ส์ ที่มีขนาดหน่วยความภายในจำแบบ RAM 256KB ทำงานที่คล็อก 225 MHz (1800 MIPS / 1350 MFLOPSS)

- ใช้ชิปเซต FPGA รุ่น Xilinx Virtex II ที่มีเกต (gate) 1M เกต

- ใช้ชิปเซตสื่อสาร 1394a เพื่อสื่อสารหรืออ่านข้อมูลวีดิทัศน์ได้ถึง 400 Mbit/s

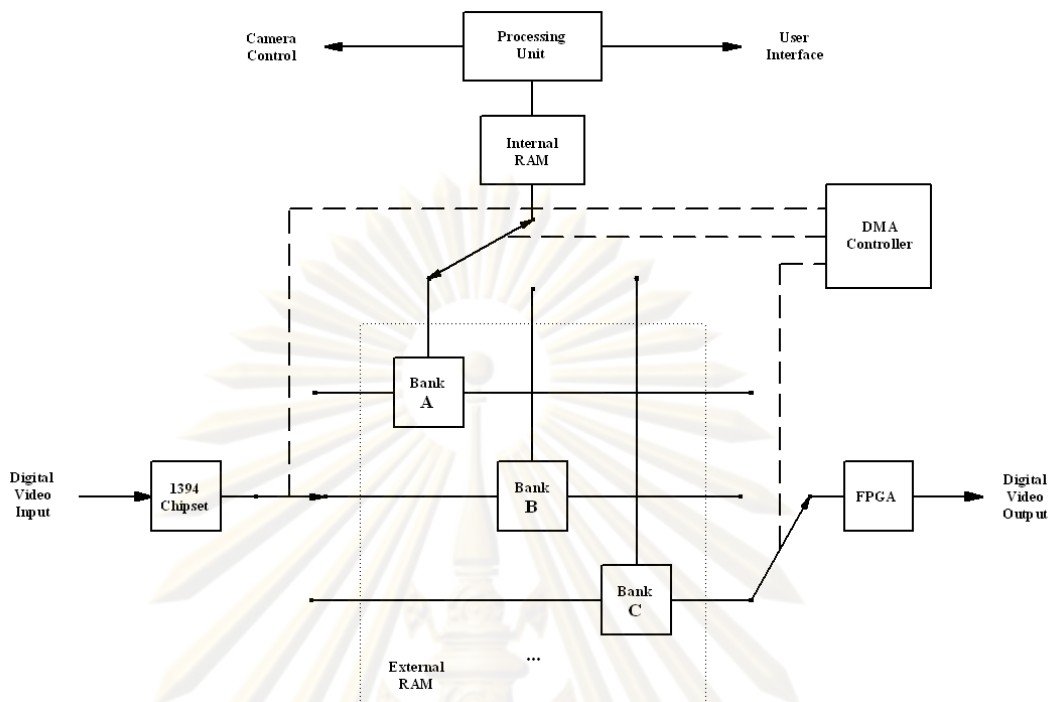
- มีหน่วยความจำภายนอกแบบ RAM 64MB

- มีหน่วยความจำแบบ FLASH 32 MB



รูปที่ 3.3 แสดงผังการทำงานของ micro-line C6713Compact

ระบบการทำงานในบอร์ดจะประกอบด้วยระบบนำภาพเข้า ที่เป็นกระแส (stream) ข้อมูลของวีดิทัศน์แบบดิจิทัล ผ่านบัสแบบ IEEE 1394 จากนั้นข้อมูลจะถูกถอดรหัสเป็น ข้อมูลดิจิทัลแบบขนานโดยชิปเซต แล้วนำข้อมูลไปเก็บไว้ในที่พักข้อมูล (buffer) จากนั้นตัวประมวลผลจะทำการส่งข้อมูลตรงสู่หน่วยความจำ (direct memory access) สู่หน่วยความจำ ภายนอก (external memory) โดยแบ่งส่วนของหน่วยความจำเป็น 3 ส่วนย่อยดังรูปที่ 3.4 ซึ่งทำงานขนานกันไป เช่น ขณะที่ส่วน B กำลังรับข้อมูลภาพจากกล้อง ส่วน A ที่เป็นฟิลด์ภาพที่แล้ว จะถูกประมวลผลโดยการนำข้อมูลบางส่วนคือส่วนที่เป็นบริเวณเป้าหมายไปใช้ในการคำนวณหา ตำแหน่งของเป้าหมายในภาพเมื่อพบแล้วจะทำการวาดกรอบหน้าต่างกลับลงมา และในส่วน C ซึ่งเป็นส่วนภาพก่อนหน้า B จะถูกนำไปประมวลผลหลัง (post processing) แล้วทำการแปลงเป็น ข้อมูลอนุกรมอีกครั้งเพื่อส่งออกโดยใช้วงจรรวมแบบ FPGA ในส่วนการติดต่อกับพนักงานจะ ติดต่อผ่านระบบสื่อสารแบบ RS-232 เช่นเดียวกับกับข้อมูลที่นำไปควบคุมการกัมส่ายชুমของ กล้อง PTZ



รูปที่ 3.4 แสดงการทำงานในบอร์ด DSP ในระบบปรับเสถียรวิดีโอและติดตามเป้าหมาย

เนื่องจากงานวิจัยนี้จะเน้นการติดตามเป้าหมายภาพ ดังนั้นจะไม่ลงรายละเอียดของกระบวนการประมวลผลก่อน (preprocessing) และประมวลผลหลัง ในหัวข้อถัดไปจะกล่าวถึงเฉพาะการใช้อัลกอริทึมในการติดตามเป้าหมายและการควบคุมกล้อง

3.2 อัลกอริทึมการติดตามเป้าหมาย

ในงานวิจัยนี้จะทำการประยุกต์การติดตามเป้าหมายด้วยวิธีมินชิฟต์ ซึ่งแบ่งออกเป็นสองอัลกอริทึมคือ การติดตามเป้าหมายด้วยมินชิฟต์แบบทั่วไปจากปริภูมิคุณลักษณะและการใช้มินชิฟต์แบบคอร์เนลเบสด์ตามคุณลักษณะของเป้าหมาย ทั้งนี้ปริภูมิคุณลักษณะ (feature space) ที่แปลงมาจากปริภูมิสี ดังนั้นจึงจำเป็นต้องเลือกปริภูมิสีที่เหมาะสม ดังจะกล่าวในหัวข้อต่อไป

3.2.1 การใช้มินชิฟต์แบบทั่วไปจากปริภูมิคุณลักษณะ (feature space)

[11] กำหนดให้ X_i เป็นสมาชิกของเซตจำกัดในหน้าต่างที่มีขนาด h ที่มี n มิติในปริภูมิยูคลีเดียน (Euclidean space) ที่มีฟังก์ชันความหนาแน่นของความน่าจะเป็น $f(x)$ ที่มีอนุพันธ์

ดังนั้นตำแหน่งของค่าเฉลี่ยการกระจาย [9] ใน h คือ

$$m(\mathbf{x}) = \frac{\sum_{\mathbf{x}_i \in h} \mathbf{x}_i f(\mathbf{x}_i)}{\sum_{\mathbf{x}_i \in h} f(\mathbf{x}_i)} \quad (3.1)$$

โดย \mathbf{x} เป็นจุดกึ่งกลางในหน้าต่าง h และ $m(\mathbf{x}) - \mathbf{x}$ คือเวกเตอร์มินิชิฟต์ ถ้าทำการเลื่อนกึ่งกลางของหน้าต่าง h ไปยังจุด $m(\mathbf{x})$ แล้วทำการคำนวณซ้ำจะได้ว่า $m(\mathbf{x})$ จะเข้าสู่ \mathbf{x} ที่มีค่าการกระจายสูงสุด นั่นคือ $m(\mathbf{x})$ เลื่อนไปหาค่าสูงสุดของ $f(\mathbf{x})$ นั่นเอง จะเห็นได้ว่ามินิชิฟต์เป็นการไต่ความชัน (gradient climbing) ซึ่งจะไปสู่ค่าสูงสุดค่าหนึ่งเสมอ ดังนั้นมินิชิฟต์จะลู่เข้า (convergence) เสมอ [11] ซึ่งพิสูจน์ได้ดังนี้

จากข้อกำหนดให้หน้าต่างมีขนาด h และมีค่ากึ่งกลาง \mathbf{x} ถ้าให้ h เข้าใกล้ศูนย์ จะพบว่า

$$m(\mathbf{x}) - \mathbf{x} \approx \frac{f'(\mathbf{x})}{f(\mathbf{x})} \quad (3.2)$$

$f'(\mathbf{x})$ เป็นอนุพันธ์ของ $f(\mathbf{x})$ โดยที่ $f(\mathbf{x})$ คือการกระจายความหนาแน่นที่มีค่าเป็นบวก ทั้งนี้ $m(\mathbf{x}) - \mathbf{x}$ อาจมีเครื่องหมายบวกหรือลบ แต่จะมีเครื่องหมายเดียวกับ $f'(\mathbf{x})$ เสมอ นั่นหมายความว่ามินิชิฟต์จะไต่ขึ้นเสมอ และค่า $f(\mathbf{x})$ จะมีค่าเป็นศูนย์ที่ยอดสูงสุดของฟังก์ชันการกระจายความหนาแน่นเสมอ ซึ่งคุณสมบัติลู่เข้าเสมอนี้เป็นสิ่งที่สำคัญมาก ทำให้ง่ายต่อการประยุกต์และเพิ่มความทนทาน (robust) ในการติดตามเป้าหมาย

3.2.1.1 มินิชิฟต์แบบทั่วไปในการติดตามภาพเป้าหมาย

การนำมาประยุกต์ในการประมวลผลภาพเพื่อติดตามเป้าหมาย การติดตามจะเป็นติดตามการกระจายคุณลักษณะ (feature distribution) ของภาพ มิใช่การกระจายความหนาแน่นของความน่าจะเป็น คุณลักษณะที่นำมาใช้ในการติดตามเป้าหมายคือ สี (chrome) ของเป้าหมาย ซึ่งจะกล่าวถึงในหัวข้อถัดไป

ขั้นตอนในการติดตามเป้าหมาย

1. พนักงานเลือกเป้าหมายโดยการครอบหน้าต่างลงบนเป้าหมาย ทั้งนี้ต้องให้พื้นที่สีส่วนใหญ่ที่ครอบบนเป้าหมายแตกต่างจากสีฉากหลังอย่างชัดเจน
2. ระบบจะสร้างฮีสโตแกรม จากสีในกรอบหน้าต่าง ซึ่งจะถูกลบเป็นคุณลักษณะของเป้าหมาย ซึ่งฮีสโตแกรมนี้จะเป็นการกระจายความน่าจะเป็นของสีของเป้าหมาย

3. ระบบจะแปลงข้อมูลสีของพิกเซลต่างๆ ที่อยู่ในหน้าต่างไปเป็นค่าในปริภูมิคุณลักษณะ (feature space) จากนั้นหาค่าเฉลี่ยของตำแหน่งในกรอบหน้าต่างใหม่ $m(x)$
4. ตรวจสอบกฎการหยุด (stop rule) โดยตรวจว่า $|m(x)-x| < \text{ค่ากำหนด}$ หรือจำนวนรอบการคำนวณถึงค่าที่กำหนดไว้หรือไม่ ถ้าใช่ไปขั้นตอนที่ 6
5. เลื่อนตำแหน่งหน้าต่างไปอยู่ที่ตำแหน่ง $m(x)$ ทำตามขั้นตอนที่ 3 ถึง 5
6. แสดงผลและควบคุมการก้มสายของกล้อง
7. รอภาพเฟรมใหม่แล้วดำเนินการตามขั้นตอน 3 ถึง 7

3.2.1.2 การเลือกปริภูมิสี

ปริภูมิสี ที่ใช้ในระบบการแสดงผลภาพแบบดิจิทัลที่นิยมในจอคอมพิวเตอร์สีต่างๆไปคือระบบ RGB หรือแดง-เขียว-น้ำเงิน ซึ่งจะมี 3 ช่องสี โดยที่ใช้งานทั่วไปจะมีค่า 8 บิตในแต่ละช่องสี ในกรณี RGB จะมีสีแดงค่าระหว่าง 0-255 เช่นเดียวกับกับสีเขียวและสีน้ำเงิน โดยเมื่อนำสีมารวมกันก็จะได้สีต่างๆ 256 ยกกำลัง 3 หรือเท่ากับ 16.8 ล้านสี ในการส่งข้อมูลภาพของระบบกล้องดิจิทัลทั่วไปจะใช้ระบบ YCbCr โดย Y จะเป็นช่องแทนความสว่าง (intensity) ซึ่งมีค่ามืดถึงสว่างภาพที่มีค่า Y อย่างเดียวคือภาพขาวดำ ส่วน Cb และ Cr เป็นองค์ประกอบของสี นอกจากนี้ยังมีปริภูมิสีที่มีพฤติกรรมคล้ายการมองเห็นของมนุษย์นั่นคือแบบ HSV โดย H หรือ hue คือสีที่สดที่สุดของสีต่างแทนค่าเป็นมุมมีค่า 0-360 องศา S หรือ saturation คือค่าความเข้มข้นของสีของสีนั้นๆ โดยทั่วไปจะมีค่า 0-1 เช่นถ้า hue คือสีแดงแล้ว S มีค่าเป็น 1 จะได้สีแดงสด ถ้า S เท่ากับ 0.2 จะได้สีแดงซีด ถ้า S เป็น 0 จะได้สีเทาโดยจะขึ้นอยู่กับค่า V ค่า V จะเหมือน Y คือค่าความสว่าง

ในความเป็นจริงยังมีปริภูมิสีอีกมาก แต่ปริภูมิสีที่เลือกมานี้เป็นปริภูมิสีพบเห็นได้ทั่วไปหรือเกี่ยวข้องกับการประมวลผลภาพ พิจารณาการถ่ายภาพจากที่สูงโดยอากาศยานไร้คนบินซึ่งมักจะปฏิบัติการในวันที่อากาศแจ่มใส ในเวลาที่มีแสงเต็มที่และกำหนดให้สีของเป้าหมายแตกต่างจากฉากหลังดังนั้นจึงนำเฉพาะสีของเป้าหมายและฉากหลังมาใช้ในการติดตามเป้าหมายและละทิ้งค่าความสว่างเพื่อให้การประมวลผลรวดเร็วยิ่งขึ้น ปริภูมิสี RGB จึงไม่เหมาะสมเพราะมีสี 3 ช่องที่รวมความสว่างไว้ในแต่ละสี จึงต้องทำการประมวลผลทั้ง 3 ช่อง ขณะที่ปริภูมิสี YCbCr และ HSV มีช่องความสว่างคือ Y และ V ที่สามารถตัดทิ้งได้ YCbCr ดูเหมือนน่าสนใจ

เนื่องจากเป็นปริภูมิสีจากกล้องถ่ายภาพวีดิทัศน์โดยตรงสามารถนำมาประมวลผลได้ทันทีโดยไม่ต้องแปลงปริภูมิสี ทำให้กระบวนการประมวลผลรวดเร็ว แต่จากการวิจัยพบว่า CbCr บรรจุรายละเอียดของภาพน้อยกว่า HS โดยรูปที่ 3.5 คือภาพต้นแบบ ส่วนรูปที่ 3.6 และ 3.7 เป็นภาพปริภูมิสี HSV และ YCbCr ตามลำดับ โดยที่ค่า Y และ V ถูกกำหนดให้เป็นค่าคงที่ค่าหนึ่ง สังเกตรูปที่ 3.6 และ 3.7 จะเห็นว่ารูปที่ 3.6 มีรายละเอียดมากกว่า 3.7 อย่างชัดเจน ซึ่งรายละเอียดเหล่านี้จะใช้นำมาแยกวัตถุเป้าหมายจากฉากหลัง เมื่อคำนวณค่าแปรปรวน (variance) ของขนาดที่ทำการนอร์มอไลซ์ (normalize) ของ $\|CbCr\|$ และ $\|HS\|$ จะพบว่า $\sigma_{CbCr}^2 = 0.0009$ และ $\sigma_{HS}^2 = 0.0256$ โดยที่อัตราส่วน $\sigma_{HS}^2 / \sigma_{CbCr}^2 = 28.44$ ซึ่งผลที่ได้สอดคล้องกับสมมุติฐาน นอกจากนี้ผู้วิจัยยังได้พัฒนาโปรแกรมประยุกต์ติดตามเป้าหมายโดยใช้ปริภูมิสีทั้งสองแบบ ซึ่งผลการทดลองยืนยันว่าการใช้ปริภูมิสีแบบ HSV ให้ความทนทานกว่าปริภูมิสีแบบ YCbCr



รูปที่ 3.5 ภาพต้นแบบ

ศูนย์วิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.6 ภาพในปริภูมิสีแบบ HSV โดยให้ V มีค่าคงที่



รูปที่ 3.7 ภาพในปริภูมิสีแบบ YCbCr โดยให้ Y มีค่าคงที่

3.2.1.3 การสร้างฮิสโตแกรมและการถ่วงน้ำหนัก

ข้อมูลสีในแต่ละพิกเซลที่อยู่ในหน้าต่างจะถูกนำมาใช้เพื่อสร้างฮิสโตแกรม โดยข้อมูลสีที่ใช้คือส่วนของ H และ S ซึ่งมีสมการดังนี้

$$f_b = \frac{1}{N} \sum_{i=1}^N B(\mathbf{I}(\mathbf{X}_i)) \quad (3.3)$$

โดย

$$B(\mathbf{x}) = \begin{cases} 1, & \mathbf{I}(\mathbf{x}) = b \in U \\ 0, & \text{otherwise} \end{cases} \quad (3.4)$$

f_b คือฮิสโตแกรมที่ถูกสร้างจากพิกเซลในหน้าต่าง I คือเวกเตอร์ของ H ขนาด 16 และ S ขนาด 16 ดังนั้น $I = [H/16 \ S/16]^T$ ของพิกเซล X_i ส่วน U คือตะกร้าสองมิติ (2D-bin) ของเวกเตอร์ I และ b คือดัชนีตะกร้า เพื่อให้การคำนวณเป็นไปอย่างรวดเร็วจะใช้ตะกร้าขนาด 16×16

ในการเลือกเป้าหมาย ในบางกรณีเป้าหมายอาจมีรูปร่างพิเศษ เช่น เรียวยาวหรือเป็นรูปทรงไม่สมมาตร การปรับขนาดหน้าต่างให้เล็กเพื่อจะได้ครอบคลุมเฉพาะบริเวณที่มีสีแตกต่างกันนั้นอาจไม่เหมาะสม เพราะขนาดหน้าต่างที่เล็กจะส่งผลต่อความทนทาน ในกรณีที่ขนาดหน้าต่างใหญ่กว่าเป้าหมายมาก ทำให้พิกเซลส่วนของฉากหลัง เข้ามาอยู่ในหน้าต่างเป็นจำนวนมากและอาจมากกว่าพิกเซลเป้าหมาย เมื่อข้อมูลถูกนำมาสร้างฮิสโตแกรม บ้างครั้งกลายเป็นว่าข้อมูลฉากหลังเป็นข้อมูลหลัก (majority) และเกิดปรากฏการณ์ติดตามฉากหลังแทน

ดังนั้นจึงต้องมีการถ่วงน้ำหนักพิกเซลในหน้าต่างเพื่อเพิ่มความสำคัญของพิกเซลตามตำแหน่งของพิกเซลในหน้าต่าง โดยมักจะให้พิกเซลบริเวณตรงกึ่งกลางหน้าต่างเป็นบริเวณที่มีน้ำหนักมากที่สุด เพราะพนักงานจะครอบหน้าต่างบนเป้าหมายโดยให้เป้าหมายอยู่ตรงกลางหน้าต่าง และให้น้ำหนักลดลงเมื่อระยะห่างจากจุดกึ่งกลาง ในวิทยานิพนธ์นี้ได้เลือกฟังก์ชันการถ่วงน้ำหนัก 3 แบบคือ

1. ฟังก์ชันแบบแบนราบ (flat) หรือแบบคงที่ เพื่อเป็นตัวควบคุมหรือตัวอ้างอิงโดยมีสมการดังนี้

$$w(\mathbf{x}) = \begin{cases} 1, & \mathbf{I}(\mathbf{x}) \in U \\ 0, & \text{otherwise} \end{cases} \quad (3.5)$$

2. ฟังก์ชันแบบพาราโบลอยด์ การถ่วงน้ำหนักแบบนี้ จะให้น้ำหนักมากบริเวณกึ่งกลางหน้าต่างแล้วลดลงเมื่อระยะทางห่างออกโดยมีความสัมพันธ์ของน้ำหนักแบบพาราโบล่า โดยมีสมมุติฐานว่าเป้าหมายมีขนาดใหญ่ออกไปรอบทุกทิศทางและครอบคลุมพื้นที่หน้าต่างมากพอสมควร โดยมีสมการดังนี้

$$w(\mathbf{x}) = \begin{cases} r^2 - |\mathbf{x} - \mathbf{x}_0|^2, & r = \frac{h}{2}, \mathbf{I}(\mathbf{x}) \in U \\ 0, & \text{otherwise} \end{cases} \quad (3.6)$$

3. ฟังก์ชันแบบเกาส์เซียน (Gaussian) การถ่วงน้ำหนักแบบนี้จะให้น้ำหนักบริเวณกึ่งกลางหน้าต่างมากและลดลงอย่างรวดเร็วเมื่อห่างจากจุดกึ่งกลาง ฟังก์ชันแบบเกาส์เซียนจะเหมาะกับเป้าหมายที่อัตราส่วนความยาวต่อความกว้างมาก เช่น เรือ อากาศยาน เป็นต้น

$$w(\mathbf{x}) = \begin{cases} \frac{1}{K} e^{-\frac{|\mathbf{x}-\mathbf{x}_0|^2}{2\sigma^2}}, & \sigma^2 \propto h, \mathbf{I}(\mathbf{x}) \in U \\ 0, & \text{otherwise} \end{cases} \quad (3.7)$$

โดย K เป็นสัมประสิทธิ์นอร์มอลไลซ์ และ \mathbf{x}_0 คือจุดกึ่งกลาง

ทั้งนี้สมการการสร้างฮิสโตแกรมแบบถ่วงน้ำหนักจะเป็นดังนี้

$$f_{wb} = \frac{1}{C} \sum_{i=1}^N w(\mathbf{X}_i) B(\mathbf{I}(\mathbf{X}_i)) \quad (3.8)$$

โดย

$$C = \sum_{i=1}^N w(\mathbf{X}_i) \quad (3.9)$$

3.2.1.4 มินิฟต์แบบทั่วไปในการติดตามเป้าหมายภาคปฏิบัติ

เมื่อพนักงานเลือกเป้าหมายและวางหน้าต่างตรงกึ่งกลางเป้าหมายแล้ว ระบบจะสร้างฮิสโตแกรมของเป้าหมายนั้นในภาพเฟรมแรก เก็บไว้ในฐานข้อมูล จากนั้นจะนำค่าสีในพิกเซลที่อยู่ในหน้าต่าง มาแปลงเป็นค่าความน่าจะเป็นตามข้อมูลในฮิสโตแกรม ข้อมูลที่ได้ใหม่นี้จะเป็นข้อมูลคุณลักษณะของภาพ (feature) ซึ่งนำมาจากสีของเป้าหมาย ดังรูปที่ 3.8



(ก)

(ข)

รูปที่ 3.8 (ก) แสดงภาพสีของเป้าหมายที่ถูกจับยึด และ (ข) ภาพคุณลักษณะที่สร้างมาจากสีของเป้าหมาย

ข้อมูลนี้จะถูกนำมาหาค่าจุดศูนย์กลางของพิกเซลตามคุณลักษณะ ดังรูปที่ 2.7 ในบทที่ 2 เพื่อหาตำแหน่งกึ่งกลางของจุดในหน้าต่าง เพื่อให้เกิดความชัดเจนจะใช้ฟังก์ชัน (x,y) แทนการใช้เวกเตอร์ในการบรรยายต่อไป โดยในการสร้างคุณลักษณะจะมีสมการดังนี้

$$F(x, y) = f_{wb}(\mathbf{I}(x, y)) \quad (3.10)$$

โดย $F(x,y)$ คือค่าความน่าจะเป็นที่ได้จากฮิสโตแกรมที่สะท้อนถึงสีของเป้าหมาย ตัวอย่างเช่นถ้าสีของเป้าหมายที่อยู่ในหน้าต่างเป็นสีแดงทั้งหมด จุดใดๆในกรอบหน้าต่างจะมีค่า $F(x,y) = 1$ โดยจุดที่มีสีอื่นจะมีค่า $F(x,y) = 0$ เมื่อได้ค่าคุณลักษณะแล้วจะทำการหาโมเมนต์อันดับศูนย์ (zeroth moment)

$$M_0 = \frac{1}{N} \sum_{x=1}^W \sum_{y=1}^L F(x, y) \quad (3.11)$$

และหา โมเมนต์อันดับหนึ่ง (first moment) ทางแกน x และแกน y ดังนี้

$$M_{1x} = \frac{1}{N} \sum_{x=1}^W \sum_{y=1}^L xF(x, y) \quad (3.12)$$

$$M_{1y} = \frac{1}{N} \sum_{x=1}^W \sum_{y=1}^L yF(x, y) \quad (3.13)$$

โดย N คือจำนวนพิกเซลในหน้าต่างที่มีขนาด L คือความสูงและ W คือความกว้างของหน้าต่าง จากนั้นคำนวณหาตำแหน่งค่าเฉลี่ย \bar{x} , \bar{y} โดยสมการ

$$\bar{x} = M_{1x}/M_0 \quad (3.14)$$

$$\bar{y} = M_{1y}/M_0 \quad (3.15)$$

เมื่อได้ตำแหน่งค่าเฉลี่ยแล้วให้ย้ายหน้าต่างโดยให้จุดกึ่งกลางของหน้าต่างอยู่บนจุด \bar{x} , \bar{y} จากนั้นทำการตรวจสอบเพื่อหยุดการหาโดยใช้วิธีการ 2 วิธีดังนี้

1. ค่าความแตกต่างของจุดกึ่งกลางน้อยกว่าค่าที่กำหนด (threshold) ดังสมการ

$$|\bar{x}_k - \bar{x}_{k-1}| \leq err \quad \text{และ} \quad |\bar{y}_k - \bar{y}_{k-1}| < err \quad (3.16)$$

2. เมื่อจำนวนการคำนวณ k ถึงค่าที่กำหนด เช่น $k = 10$ เป็นต้น

โดย k คือรอบที่ของการคำนวณ และ err คือค่าพิสัยที่กำหนด

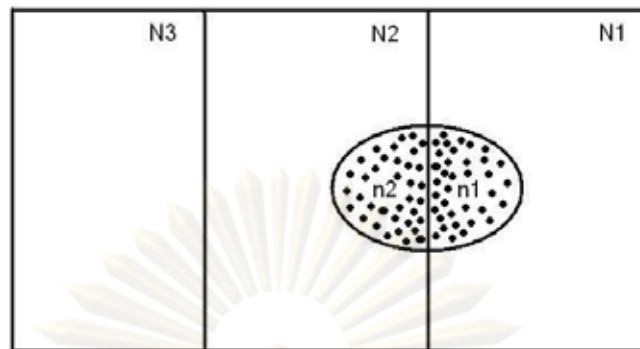
จากนั้นนำค่าจุดกึ่งกลางใหม่ที่ได้ไปใช้ในการควบคุมการกัมส่ายของกล้องและไปเป็นค่าตั้งต้นหน้าต่างในเฟรมภาพถัดไป

3.2.1.5 การปรับขนาดหน้าต่างขณะจับยึดเป้าหมาย

ในการติดตามเป้าหมายจากที่สูง วัตถุที่เป็นเป้าหมายมักมีขนาดไม่เปลี่ยนแปลง ดังนั้นการติดตามเป้าหมายจึงสามารถให้ขนาดหน้าต่างมีขนาดคงที่ได้ แต่ในบางกรณีจะต้องปรับเปลี่ยนขนาดหน้าต่าง เช่น เมื่อมีการปรับเปลี่ยนอัตราซูม เพื่อให้การติดตามเป้าหมายมีความทนทาน การปรับเปลี่ยนขนาดหน้าต่างทำได้สองวิธีคือการปรับด้วยคน (manual-adjustment) หรือการปรับแบบอัตโนมัติ (auto-adjustment) ในการปรับค่าแบบอัตโนมัติ จะใช้คนเป็นผู้ออกคำสั่ง จากนั้นระบบจะปรับขนาดหน้าต่างให้ครอบคลุมเป้าหมาย โดยมีวิธีการดังนี้ พิจารณารูปที่

3.9

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.9 แสดงแนวคิดในการปรับภาพแบบอัตโนมัติ

สมมติว่าหน้าต่างครอบเป้าหมายในกรอบทางขวามือ จะเห็นได้ว่าพิกเซลที่เป็นเป้าหมายเท่ากับ n_1 และจำนวนพิกเซลในหน้าต่างทั้งหมดมีจำนวน N_1 ถ้าสมมุติอีกว่าพิกเซลที่เป็นฉากหลังมีค่าความน่าจะเป็นเท่ากับ 0 จะได้ว่า

$$M_0 = c \frac{n_1}{N_1} \quad (3.17)$$

โดย c คือค่าความน่าจะเป็นของพิกเซลที่เป็นเป้าหมาย ถ้าขนาดหน้าต่างขยายไปครอบคลุมถึงกรอบกลาง และถ้า $N_2 = N_1$ และ $n_2 = n_1$ จะได้ว่า M_0 ดังนี้

$$c \frac{n_1 + n_2}{N_1 + N_2} = c \frac{n_1}{N_1} = M_0 \quad (3.18)$$

ถ้าขนาดหน้าต่างเพิ่มขึ้นและครอบจำนวนพิกเซลเป้าหมายในสัดส่วนเดียวกันจะเห็นว่าค่าส่วนต่างของโมเมนต์อันดับที่ศูนย์ หรือ $\Delta M_0 = 0$ แต่ถ้าขนาดหน้าต่างขยายไปครอบคลุมถึงส่วนซ้ายมือ ให้ $N_1 + N_2 + N_3 = a(N_1 + N_2)$ ทั้งนี้โดยไม่มีจำนวนพิกเซลของเป้าหมายเพิ่มขึ้น หรือพิกเซลเป้าหมายยังมีจำนวน $n_1 + n_2$ จะได้ว่า

$$M_0 = c \frac{n_1}{aN_1} \quad (3.19)$$

$$M_{0_new} - M_0 = c \frac{n_1}{aN_1} - c \frac{n_1}{N_1} = M_0 \left(\frac{1}{a} - 1 \right) \quad (3.20)$$

$$-\Delta M_0 / M_0 = \left(1 - \frac{1}{a} \right) \quad (3.21)$$

ถ้าอัตราการขยายของหน้าต่างเป็น 20% จะได้ว่า $a = 1.2$ ดังนั้นจะพบว่า $-\Delta M_0/M_0 = 0.17$ ซึ่งขั้นตอนที่กล่าวมานี้จะใช้เป็นขั้นตอนในการหยุดการขยายหน้าต่าง เพราะถ้าการขยายหน้าต่างยังคงครอบคลุมบางส่วนของพิทเชลของเป้าหมายค่า $-\Delta M_0/M_0$ จะน้อยกว่า 0.17 ในทางปฏิบัติเนื่องจากภาพมักมีการรบกวน (noise) ดังนั้นค่าที่เหมาะสมจะอยู่ระหว่าง 0.12 ถึง 0.15 และในกรณีที่ภาพมีการรบกวนมากวิธีการที่กล่าวมาจะใช้ไม่ได้ผล ดังนั้นจึงต้องกำหนดขนาดการขยายสูงสุดเพื่อหยุดการปรับขนาดของหน้าต่าง

เราสามารถสรุปขั้นตอนการปรับขนาดหน้า ต่างแบบอัตโนมัติได้ดังนี้

- พนักงานทำการออกคำสั่ง
- ระบบทำการขยายตามสัดส่วนที่กำหนด เช่น $a = 1.2$
- ระบบตรวจข้อจำกัดการหยุดว่าได้ค่ามากกว่าที่กำหนดหรือไม่ ถ้าใช่ให้หยุด
- ระบบตรวจว่าค่าการขยายเกินขีดจำกัดหรือไม่ ถ้าใช่ให้หยุด
- ทำการขยายขนาดหน้าต่างต่อ

3.2.2 การใช้มินชิฟต์แบบเคอร์เนลเบสส์

ใน [12,13] ได้กล่าวถึงการใช้มินชิฟต์ในการติดตามเป้าหมาย แทนที่จะใช้น้ำหนักจากคุณลักษณะ แต่ใช้การเปรียบเทียบการประมาณการกระจายความหนาแน่น (density distribution estimation) ของแบบจำลองเป้าหมาย (target model) กับตัวแทนเป้าหมาย (candidate target) โดยมีทฤษฎีดังนี้

ให้เซต $\{\mathbf{x}_i\}_{i=1\dots n}$ คือจำนวนจุดที่มีขนาด d มิติในปริภูมิ R^d โดยมีแกนหรือเคอร์เนลเป็น $K(\mathbf{x})$ และหน้าต่างที่มีรัศมี h จะสามารถเขียนสมการการประมาณการกระจายความหนาแน่น (density distribution estimation) ได้ดังนี้

$$\hat{f}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \quad (3.22)$$

ถ้าเคอร์เนลมีสมการเป็น

$$K_E(\mathbf{x}) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2) (1 - \|\mathbf{x}\|^2), & \|\mathbf{x}\| < 1 \\ 0, & \text{otherwise} \end{cases} \quad (3.23)$$

ทั้งนี้ ถ้า $K(\mathbf{x})$ เป็นเคอร์เนลแบบอีพานีคินคอฟ (Epanechnikov kernel) แบบสหแปรปรวน (multivariate) ถ้าให้ $k(x)$ เป็นโปรไฟล์ของเคอร์เนล K โดย $k : [0, \infty) \rightarrow R$ จะได้ $K(x) = k(\|\mathbf{x}\|^2)$ โดยเขียนเป็นสมการได้

$$k_E(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2) (1 - x), & x < 1 \\ 0, & \text{otherwise} \end{cases} \quad (3.24)$$

เราสามารถเขียน การประมาณการกระจายความหนาแน่นใหม่ได้ดังนี้

$$\hat{f}_K(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^n k\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right) \quad (3.25)$$

ถ้ากำหนดให้

$$g(\mathbf{x}) = -k'(\mathbf{x}) \quad (3.26)$$

โดย $g(\mathbf{x})$ คือเกรเดียนของ $k(x)$ โดยมีสมมุติฐานว่าอนุพันธ์ของ $k(x)$ เป็นจริงทุกค่า เราสามารถเขียนเคอร์เนล G ได้ดังนี้

$$G(\mathbf{x}) = Cg(\|\mathbf{x}\|^2) \quad (3.27)$$

โดย C เป็นสัมประสิทธิ์นอร์มอลไลซ์ (normalized coefficient)

เราจะหาค่าเกรเดียน (gradient) ของการประมาณการกระจายความหนาแน่นได้ดังนี้

$$\nabla \hat{f}_K(\mathbf{x}) = \frac{2}{nh^{d+2}} \sum_{i=1}^n (\mathbf{x} - \mathbf{x}_i) k'\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right) \quad (3.28)$$

$$\nabla \hat{f}_K(\mathbf{x}) = \frac{2}{nh^{d+2}} \sum_{i=1}^n (\mathbf{x}_i - \mathbf{x}) g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right) \quad (3.29)$$

$$\nabla \hat{f}_K(\mathbf{x}) = \frac{2}{nh^{d+2}} \sum_{i=1}^n \left[g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) \right] \left[\frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x} \right] \quad (3.30)$$

พิจารณาวงเล็กสุดท้าย ซึ่งคือเวกเตอร์มินิซิปต์แบบไต่ความชัน

$$M = \left[\frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x} \right] \quad (3.31)$$

ในการหาค่าฐานนิยมของ $\hat{f}_K(\mathbf{x})$ จะพบว่าค่า $\nabla \hat{f}_K(\mathbf{x}) = 0$ จากสมการ 3.32
จะได้

$$0 = \left[\frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x} \right] \quad (3.32)$$

$$\mathbf{x}_p = \left[\frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} \right] \quad (3.33)$$

ดังนั้น \mathbf{x}_p จึงเป็นตำแหน่งสูงสุดของการกระจายความหนาแน่นในตำบลที่ใกล้เคียงกับ \mathbf{x}_i (local peak)

ระยะห่าง Bhattacharyya

ฟังก์ชันคล้ายคลึง (similarity function) คือฟังก์ชันที่วัดความคล้ายคลึงระหว่างการกระจายความหนาแน่น 2 ชุด การหาความคล้ายคลึงมีหลายวิธีแต่ในวิทยานิพนธ์นี้จะกล่าวเฉพาะระยะห่าง Bhattacharyya ที่มีค่าจัดความดังนี้

$$d(\mathbf{y}) = \sqrt{1 - \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}]}, 0 \leq d(\mathbf{y}) \leq 1 \quad (3.34)$$

โดย $\hat{\mathbf{q}}$ คือการประมาณการกระจายความหนาแน่นของแบบจำลองเป้าหมาย

และ $\hat{\mathbf{p}}(\mathbf{y})$ คือการประมาณการกระจายความหนาแน่นของตัวแทนเป้าหมายที่ตำแหน่ง \mathbf{y}

ให้ ρ คือสัมประสิทธิ์ Bhattacharyya จะได้

$$\hat{\rho}(\mathbf{y}) \equiv \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\mathbf{y})\hat{q}_u} \quad \text{โดย } 0 \leq \hat{\rho}(\mathbf{y}) \leq 1 \quad (3.35)$$

เราสามารถแทนความหมายของ $\hat{\mathbf{p}}$ และ $\hat{\mathbf{q}}$ ในรูปของเวกเตอร์ขนาด m มิติที่มีขนาดหนึ่งหน่วยได้ดังนี้

$$\hat{\mathbf{p}} = [\sqrt{\hat{p}_1} \dots \sqrt{\hat{p}_m}]^T \quad \text{และ} \quad \hat{\mathbf{q}} = [\sqrt{\hat{q}_1} \dots \sqrt{\hat{q}_m}]^T$$

ถ้าพิจารณาจะพบว่าสัมประสิทธิ์ Bhattacharyya คือการทำสหสัมพันธ์ (correlation) ของเวกเตอร์ $\hat{\mathbf{p}}$ และ $\hat{\mathbf{q}}$ นั้นเอง จะพบว่าความคล้ายคลึงมากที่สุดเมื่อระยะห่าง Bhattacharyya เป็น 0 นั่นคือ $\hat{\mathbf{p}} = \hat{\mathbf{q}}$ และจะได้ว่าสัมประสิทธิ์ Bhattacharyya เท่ากับ 1 ซึ่งเป็นค่ามากที่สุด

3.2.2.1 แบบจำลองเป้าหมาย (target model)

แบบจำลองเป้าหมายจะถูกกำหนดจากฟังก์ชันในหน้าต่างซึ่งอาจเป็นวงรีหรือวงกลม จากนั้นจะทำการนอลมอดไลซ์ให้อยู่ในรูปของวงกลมหนึ่งหน่วย กำหนดให้ $\{\mathbf{x}_i^*\}_{i=1 \dots n}$ เป็นฟังก์ชันที่ถูกนอลมอดไลซ์ในบริเวณที่กำหนดให้เป็นเป้าหมาย และมีจุดศูนย์กลางอยู่ที่ 0 โดยการใส่โปรไฟล์ของเคอร์เนล $k(\mathbf{x})$ แบบไอโซทรอปิก (isotropic) โค้งนูน (convex) ลดค่าลงอย่างเดียว (decreasing monotonic) ทั้งนี้จะใช้ฮิสโตแกรมที่มีขนาดตะกร้าเท่ากับ m เป็นตัวแทนการประมาณการกระจายความหนาแน่น โดยกำหนดให้ฟังก์ชัน $b(\mathbf{x})$ เป็นฟังก์ชันแปลงค่าสีจากฟังก์ชัน \mathbf{x} ไปเป็นดัชนีตะกร้าโดย $b: \mathbb{R}^2 \rightarrow \{1 \dots m\}$ และ u เป็นหมายเลขตะกร้า จะได้

$$\hat{q}_u = C \sum_{i=1}^n k(\|\mathbf{x}_i^*\|^2) \delta[b(\mathbf{x}_i^*) - u] \quad (3.36)$$

ทั้งนี้ δ คือฟังก์ชันเดลต้าของ Kronecker และ C คือสัมประสิทธิ์นอลมอดไลซ์

$$C = \frac{1}{\sum_{i=1}^n k(\|\mathbf{x}_i^*\|^2)} \quad (3.37)$$

โดยมีเงื่อนไขว่า $\sum_{u=1}^m \hat{q}_u = 1$

3.2.2.2 ตัวแทนเป้าหมาย (target candidate)

เช่นเดียวกับแบบจำลองเป้าหมาย แต่ต่างกันที่ตัวแทนเป้าหมายคือกลุ่มพิกเซลในเฟรมปัจจุบันโดยมีจุดศูนย์กลางที่ตำแหน่ง \mathbf{y} กำหนด $\{\mathbf{x}_i\}_{i=1 \dots n}$ เป็นพิกเซลที่ถูกนอลมอดไลซ์อยู่ในหน้าต่างที่มีจุดศูนย์กลางอยู่ที่ \mathbf{y} ในเฟรมปัจจุบัน โดยใช้โปรไฟล์ $k(\mathbf{x})$ เดียวกับแบบจำลองเป้าหมาย แต่อาจมีขนาดหน้าต่าง h ต่างกัน แต่ต้องมีจำนวนตะกร้าในฮิสโตแกรมเท่ากันเท่ากับ m จะได้การประมาณการกระจายความหนาแน่นของตัวแทนเป้าหมายที่ตำแหน่ง \mathbf{y}

$$\hat{p}_u(\mathbf{y}) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{\mathbf{y} - \mathbf{x}_i}{h}\right\|^2\right) \delta[b(\mathbf{x}_i) - u] \quad (3.38)$$

และ

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{\mathbf{y} - \mathbf{x}_i}{h}\right\|^2\right)} \quad (3.39)$$

โดย $\sum_{u=1}^m \hat{p}_u = 1$

โดย C_h เป็นสัมประสิทธิ์นอลมอดไลซ์ของการกระจายของตัวแทนเป้าหมาย

จะเห็นได้ว่าการเปรียบเทียบ ขนาดหน้าต่างของตัวแทนเป้าหมาย (จำนวนพิกเซล) ไม่จำเป็นต้องเท่ากับแบบจำลองเป้าหมาย

3.2.2.3 การติดตามเป้าหมายภาคปฏิบัติ

ในการหาตำแหน่งเป้าหมายในเฟรมปัจจุบัน จะใช้ฟังก์ชันความคล้ายคลึง $d(\mathbf{y})$ โดยหาจากตำแหน่งที่ให้ค่า $d(\mathbf{y})$ น้อยที่สุด การที่ค่า $d(\mathbf{y})$ น้อยที่สุด นั่นคือสัมประสิทธิ์ Bhattacharyya จะมีค่ามากที่สุด

1. การหาสัมประสิทธิ์ Bhattacharyya

ในเฟรมปัจจุบันเป้าหมายได้เคลื่อนไปอยู่ที่ตำแหน่ง \mathbf{y} ในการค้นหาเป้าหมายนั้น จะต้องเริ่มต้นจากตำแหน่งของเป้าหมายของภาพที่แล้ว โดยกำหนดให้เป็นตำแหน่ง \mathbf{y}_0 ในเฟรมปัจจุบัน จะได้ $\{\hat{p}_u(\mathbf{y}_0)\}_{u=1\dots m}$ เป็นการประมาณการกระจายความหนาแน่นของตัวแทนเป้าหมายที่ตำแหน่ง \mathbf{y}_0 โดยใช้สมการการขยายเทเลอร์รอบ $\hat{p}_u(\mathbf{y}_0)$ จะได้ค่าประมาณเชิงเส้น (linear approximation) ของสัมประสิทธิ์ Bhattacharyya ดังนี้

$$\rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\mathbf{y}_0) \hat{q}_u} + \frac{1}{2} \sum_{u=1}^m \hat{p}_u(\mathbf{y}) \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\mathbf{y}_0)}} \quad (3.40)$$

โดยการประมาณนี้จะไม่ผิดพลาดมากนัก ถ้า $\hat{p}_u(\mathbf{y})$ กับ $\hat{p}_u(\mathbf{y}_0)$ ไม่แตกต่างกันมาก ดังนั้นเราสามารถเขียนสมการใหม่ได้ดังนี้

$$\rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\mathbf{y}_0) \hat{q}_u} + \frac{C_h}{2} \sum_{i=1}^{n_h} w_i k \left(\left\| \frac{\mathbf{y} - \mathbf{x}_i}{h} \right\|^2 \right) \quad (3.41)$$

$$w_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\mathbf{y}_0)}} \delta[b(\mathbf{x}_i) - u] \quad (3.42)$$

เพื่อที่จะหาตำแหน่งเป้าหมาย เราจะใช้เฉพาะเทอมที่สองของสมการ 3.41 ทั้งนี้ เพราะเทอมแรกจะไม่ขึ้นกับ \mathbf{y}

2. มินชิฟต์แบบขยายขอบเขต (extended meanshift)

ในการหามินชิฟต์แบบทั่วไปของการประมาณค่าการกระจายความหนาแน่น $\hat{f}_K(\mathbf{x})$ ในสมการ 3.25 จะเป็นดังสมการ 3.33 แต่เนื่องจากการหาค่าสูงสุดของเทอมที่สองของสมการ 3.41 ดังนั้นจะต้องปรับปรุงสมการ 3.33 เป็น

$$\hat{\mathbf{y}}_1 = \frac{\left[\sum_{i=1}^{n_h} \mathbf{x}_i w_i g \left(\left\| \frac{\mathbf{y}_0 - \mathbf{x}_i}{h} \right\|^2 \right) \right]}{\left[\sum_{i=1}^{n_h} w_i g \left(\left\| \frac{\mathbf{y}_0 - \mathbf{x}_i}{h} \right\|^2 \right) \right]} \quad (3.43)$$

สมการ 3.43 เป็นสมการมินชิฟต์แบบขยายขอบเขต

ถ้าเราใช้คอร์เนลแบบอี่พานิคนี่คอฟ จะพบว่า $g(\mathbf{x})$ จะเป็นค่าคงที่ ดังนั้นสมการ 3.43 จะลดรูปเหลือสมการ 3.44 ที่เหมาะสมนำไปใช้เพื่อติดตามเป้าหมาย

$$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i w_i}{\sum_{i=1}^{n_h} w_i} \quad (3.44)$$

ขั้นตอนการติดตามเป้าหมาย

ในการติดตามเป้าหมายในแต่ละเฟรมนั้น จะมีขั้นตอนดังนี้

1. พนักงานกำหนดเป้าหมายโดยครอบหน้าต่างลงบนกึ่งกลางเป้าหมาย ระบบ จะทำการสร้างการประมาณการกระจายความหนาแน่น $\{\hat{q}_u\}_{u=1\dots m}$ ไว้

2. ในเฟรมปัจจุบัน สร้างการประมาณการกระจายความหนาแน่นตัวแทนเป้าหมาย y_0 โดย $\{\hat{p}_u(y_0)\}_{u=1\dots m}$ และหาสัมประสิทธิ์ Bhattacharyya ที่ y_0

$$\rho[\hat{\mathbf{p}}(y_0), \hat{\mathbf{q}}] = \sum_{u=1}^m \sqrt{\hat{p}_u(y_0) \hat{q}_u}$$

3. หา $\{w_i\}_{i=1\dots m}$

4. หา y_1 จากสมการ 3.44

5. หา $\{\hat{p}_u(y_1)\}_{u=1\dots m}$ และ $\rho[\hat{\mathbf{p}}(y_1), \hat{\mathbf{q}}]$

6. ถ้า $\rho[\hat{\mathbf{p}}(y_1), \hat{\mathbf{q}}] < \rho[\hat{\mathbf{p}}(y_0), \hat{\mathbf{q}}]$ ให้ $y_1 \leftarrow 1/2(y_0 + y_1)$ หาก $\rho[\hat{\mathbf{p}}(y_1), \hat{\mathbf{q}}]$

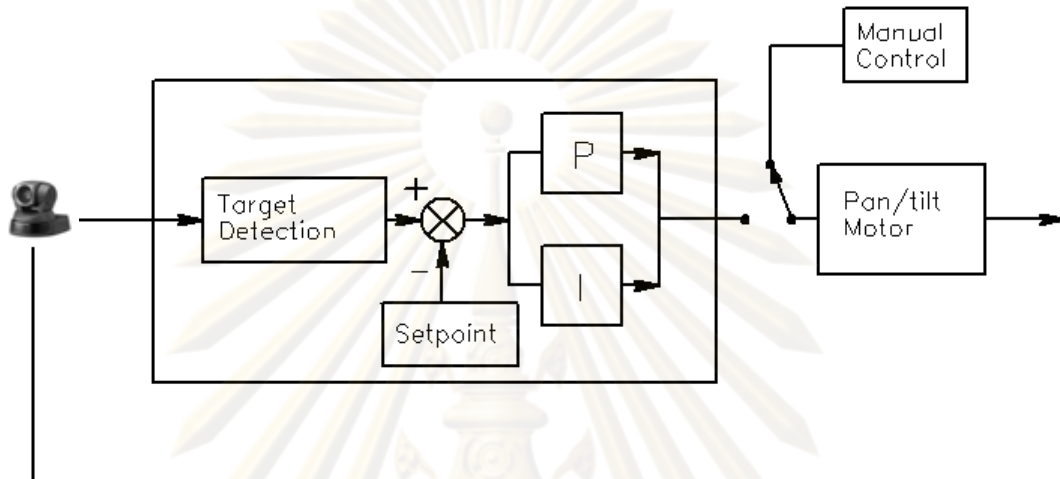
7. ตรวจ $\|y_1 - y_0\| < \text{err}$ หรือจำนวนรอบสูงสุดถ้าใช่ให้หยุด รอเฟรมใหม่

8. ให้ $y_0 \leftarrow y_1$ กับไปเริ่มใหม่ที่ขั้นตอนที่ 2

การคำนวณในขั้นตอนที่ 6 ทำไปเพื่อหลีกเลี่ยงปัญหาจากตัวเลขที่ได้จากการคำนวณ แต่จากการทดลอง [13] พบว่าโอกาสผิดพลาดมีเพียง 0.1% ดังนั้นในการปฏิบัติจริง ขั้นตอน 2, 5 และ 6 จึงไม่ต้องนำมาใช้

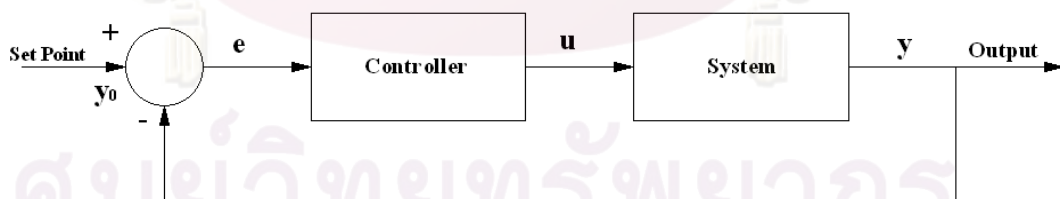
3.3 การควบคุมการส่ายก้มเงยของกล้องวิดีโอ

การควบคุมการส่ายก้มเงยของกล้องในวิทยานิพนธ์นี้จะใช้การควบคุมแบบป้อนกลับ ที่ใช้ตัวควบคุมแบบ PI มีการทำงานของระบบดังรูปที่ 3.10 ซึ่งมีหลักการดังนี้



รูปที่ 3.10 ผังการทำงานของระบบควบคุมการก้มส่ายกล้อง

จากรูปที่ 3.11 ขณะทีระบบอยู่ในสภาวะคงตัว (steady state) สัญญาณขาออก (output) y จะถูกนำมาป้อนกลับที่ผลต่างเพื่อหักล้างกับค่าตั้ง (set point) y_0 โดยมีค่าผิดพลาด (error) $e = y_0 - y$ เพื่อป้อนให้กับตัวควบคุม (controller) ตัวควบคุมจะพยายามปรับค่าเพื่อให้ค่า y เข้าหาค่า y_0 โดยจะได้สัญญาณขาออกจากตัวควบคุมคือ u จากนั้นค่านี้จะถูกนำไปใช้โดยระบบในการปรับเปลี่ยน y

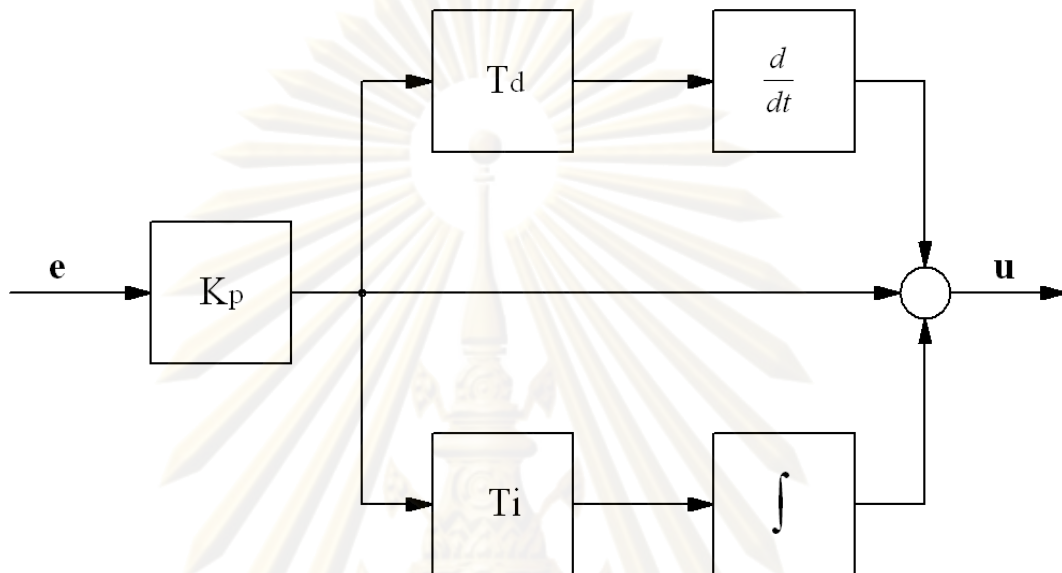


รูปที่ 3.11 ผังควบคุมแบบป้อนกลับ

3.3.1 การควบคุมแบบ PID

หรือ สัดส่วน-อินทิกรัล-อนุพันธ์ [16] (proportional-integral-derivative) ทั้งนี้เพื่อปรับค่าผิดพลาด e เพื่อให้ได้ค่า u ที่เหมาะสมกับระบบ การควบคุมแบบ PID จะเริ่มจากการอ่านค่าจากตัวเซนเซอร์ (sensor) แล้วนำมาหักออกจากค่าตั้ง นำค่าผิดพลาดที่ได้ไปควบคุม โดยการควบคุมมี 3 ลักษณะคือ การจัดการกับปัจจุบันผ่านค่าสัดส่วน การแก้ไขค่าจากในอดีตผ่านตัว

อินทิกรัล และป้องกันปัญหาในอนาคตผ่านส่วนอนุพันธ์ รูปที่ 3.12 แสดงถึงส่วนควบคุมแบบ PID โดยค่า K_p คือสัมประสิทธิ์ของส่วน P ส่วน T_i และ T_d ค่าตัวคงที่ทางเวลา (time constant) ของส่วน I และ D ตามลำดับ



รูปที่ 3.12 ฟังก์ชันควบคุมแบบ PID

3.3.1.1 ตัวควบคุมแบบสัดส่วน

ตัวควบคุมสัดส่วน มีค่าขาออกเป็นสัดส่วนกับค่าผิดพลาด ซึ่งเขียนในรูปสมการได้ $u_p = K_p \cdot e$ การใช้ตัวควบคุมแบบสัดส่วนอย่างเดียว จะทำให้เกิดค่าความต่างระหว่างค่าตั้งกับค่าขาออกแบบคงตัว (stationary) เสมอ ยกเว้นเมื่อค่าขาเข้าเป็นศูนย์ การตั้งค่าสัมประสิทธิ์สัดส่วน K_p สูงเกินไปจะทำให้ระบบไม่เสถียร

3.3.1.2 ตัวควบคุมอินทิกรัล

ตัวควบคุมแบบอินทิกรัลจะนำผลรวมของค่าผิดพลาดในอดีต เพื่อใช้ในการควบคุมทางขาออก ดังสมการ $U_i = K_p T_p \int e(\tau) d\tau$ โดยผลของการควบคุมจะหยุดลงเมื่อผลรวมของค่าผิดพลาดเป็นศูนย์ ซึ่งจะเกิดเมื่อค่าขาออก y เท่ากับค่าตั้ง y_0 ดังนั้นตัวควบคุมแบบอินทิกรัลจะกำจัดค่าความแตกต่างแบบคงตัวที่ปรากฏในการควบคุมแบบสัดส่วน โดยมีข้อแม้ว่าค่าตั้งจะต้องเสถียร ตัวควบคุมแบบอินทิกรัลมักใช้ร่วมกับตัวควบคุมแบบสัดส่วนเรียกว่า PI ถ้าใช้ตัวควบคุมแบบอินทิกรัลอย่างเดียวจะให้ผลตอบสนองที่ช้ามาก

3.3.1.3 ตัวควบคุมแบบอนุพันธ์

ตัวควบคุมแบบอนุพันธ์จะนำอัตราการเปลี่ยนแปลงของค่าผิดพลาด โดยมีสมการคือ $u_d = de(t)/dt$ ซึ่งค่าที่ได้มักจะมีค่าตรงข้ามของบ่อนกลับ ทำให้ตัวควบคุมแบบอนุพันธ์เป็นตัวหน่วงที่ดีในกรณีที่มีการเปลี่ยนแปลงในระบบอย่างรวดเร็ว ตัวควบคุมแบบอนุพันธ์มักใช้ร่วมกับตัวควบคุมแบบสัดส่วน หรือตัวควบคุมแบบสัดส่วนและอินทิกรัล เรียกว่าแบบ PD และ PID ข้อเสียของตัวควบคุมแบบอนุพันธ์ คือ ตัวมันจะมีพฤติกรรมเป็นตัวกรองผ่านสูง (high pass filter) ซึ่งมักจะขยายสัญญาณรบกวนและทำให้ระบบไม่เสถียร ดังนั้นตัวควบคุมแบบอนุพันธ์จึงไม่นิยมใช้ในระบบที่มีการรบกวนมาก

3.3.2 การหาพารามิเตอร์ด้วยวิธี Ziegler-Nicole

[16] วิธีที่ดีที่สุดในการพารามิเตอร์ของการควบคุมแบบ PID (K_p T_i และ T_d) คือการหาแบบจำลองทางคณิตศาสตร์ (mathematic model) แต่บางครั้งเป็นการยากที่จะหาแบบดังกล่าว ดังนั้นจึงมีผู้ค้นคิดวิธีการหาค่าพารามิเตอร์ดังกล่าว ซึ่งหนึ่งในวิธีดังกล่าวที่เป็นที่นิยมมากคือวิธีการของ Ziegler-Nicole โดยมีขั้นตอนดังนี้

- กำหนดให้ค่าการขยายแบบ I และ D เท่ากับศูนย์
- เพิ่มการขยายแบบ P
- ตรวจสอบว่าระบบเกิดการแกว่งหรือไม่ ถ้าไม่ให้เพิ่มค่า P จนระบบแกว่งแบบต่อเนื่องเป็นครั้งแรก อัตราขยาย P นี้จะเรียกว่าอัตราขยายวิกฤต (K_c) วัดคาบการสั่น P_c
- นำค่า K_c และ P_c มาหาพารามิเตอร์ต่างๆ ในตาราง 3.1

ตาราง 3.1 แสดงการหาค่าพารามิเตอร์ของตัวควบคุมจากค่า K_c และ P_c

ตัวควบคุม \ พารามิเตอร์	K_p	T_i	T_d
P	$0.5K_c$	-	-
PI	$0.45K_c$	$0.85P_c$	-
PID	$0.65K_c$	$0.5P_c$	$0.12P_c$

3.3.3 ตัวควบคุมแบบ PID แบบดิสครีต

[16] ระบบการควบคุมแบบดิจิทัลที่ใช้ไมโครโปรเซสเซอร์ จะมีการทำงานเป็นช่วง ซึ่งจัดเป็นแบบไม่ต่อเนื่อง (non-continuous) หรือแบบดิสครีต (discrete) การควบคุมด้วย PID สามารถประยุกต์บนแบบดิสครีตได้โดยระบบจะอ่านค่า คำนวณค่าผิดพลาดให้สัญญาณขาออกตามช่วงคาบ (interval) T ที่สุ่มสัญญาณ (sampling) จากรูปที่ 3.12 เราสามารถเขียนสมการโอนย้าย (transfer function) ของตัวควบคุมได้ดังนี้

$$\frac{U(s)}{E(s)} = H(s) = K_p \left[1 + \frac{1}{T_i s} + T_d s \right] \quad (3.45)$$

ซึ่งสามารถเขียนให้อยู่ในรูปโดเมนเวลา (time domain) ได้ดังนี้

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (3.46)$$

เปลี่ยนรูปการอินทิเกรตและการอนุพันธ์ให้อยู่ในรูปแบบดิสครีตได้ดังนี้

$$\int_0^t e(\tau) d\tau \approx T \sum_{k=0}^n e(k) \quad (3.47)$$

$$\frac{de(t)}{dt} \approx \frac{e(n) - e(n-1)}{T} \quad (3.48)$$

โดย

$$t = nT \quad (3.49)$$

โดย n คือขั้นดิสครีต (discrete step) ที่เวลา t ซึ่งนำไปสู่สมการควบคุมแบบดิสครีต

$$u(n) = K_p e(n) + K_i \sum_{k=0}^n e(k) + K_d (e(n) - e(n-1)) \quad (3.50)$$

เนื่องจาก $e(n) = y(n) - y_0(n)$ ดังนั้น ประมาณว่า $e(n) - e(n-1) = y(n) - y(n-1)$ จะได้

$$u(n) = K_p e(n) + K_i \sum_{k=0}^n e(k) + K_d (y(n) - y(n-1)) \quad (3.51)$$

โดยที่

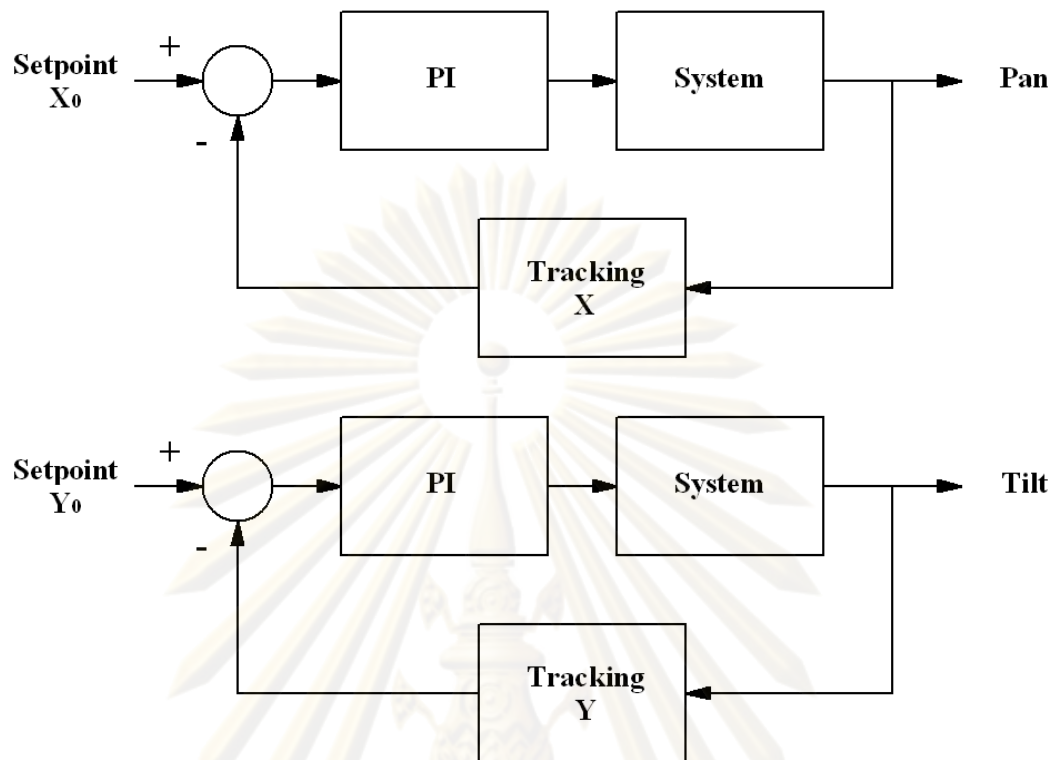
$$K_i = \frac{K_p T}{T_i} \quad (3.52)$$

$$K_d = \frac{K_p T_d}{T} \quad (3.53)$$

ซึ่งสมการ 3.52 จะใช้ในการควบคุมในระบบตีสครีต

3.3.4 การควบคุมกล้อง

ในการควบคุมกล้องจะใช้ตัวควบคุมแบบ PI เท่านั้นเนื่องจากระบบติดตามเป้าหมายมีสัญญาณรบกวนต่างๆมาก ซึ่งการเพิ่มตัวควบคุมแบบ D จะทำให้ระบบอาจไม่เสถียรได้ การควบคุมจะเกิดขึ้นเมื่อพนักงานทำการจับยึด (lock) เป้าหมาย เป้าหมายอาจกำลังเคลื่อนที่ออกจากจอภาพ ดังนั้นระบบควบคุมจะทำการสายกัมเมยกกล้อง เพื่อให้เป้าหมายกลับสู่กลางจอภาพแสดงผล เมื่อพนักงานจับยึดเป้าหมาย โดยเป้าหมายห่างจากจุดกึ่งกลางภาพเป็นระยะ $(x-x_0, y-y_0)$ โดย (x_0, y_0) คือจุดกึ่งกลางภาพ ค่าความแตกต่าง (dx, dy) จะถูกนำไปป้อนให้กับระบบควบคุม โดยการควบคุมจะแยกแกน ดังรูปที่ 3.13 จากที่กล่าวมาแล้ว ในการควบคุมแบบ PI ค่า K_p และ K_i จะต้องมีค่าที่เหมาะสม ถ้าค่าคงที่สองค่านี้น้อยไประบบจะติดตามเป้าหมายช้าเกินไป อาจทำให้เป้าหมายเคลื่อนที่ออกนอกจอภาพได้ แต่ถ้ามีค่ามากเกินไปอาจทำให้ระบบไม่เสถียร และเกิดการแกว่ง (oscillate) ได้



รูปที่ 3.13 ระบบควบคุมกลิ้งโดยแยกแกน X Y

3.3.5 การหาค่าพารามิเตอร์ค่า K_p และ K_i ที่อัตราซุ่มต่างๆ

เนื่องจากอัตราซุ่มมีผลกระทบต่อโดยตรงกับระยะขจัดของเป้าหมายที่ปรากฏในจอแสดงผล เช่นเมื่อกลิ้งมีอัตราซุ่มมาก การขยับกลิ้งเพียงเล็กน้อยจะทำให้ระยะทางในภาพเคลื่อนที่ไปมาก ดังนั้นจึงต้องหาค่าพารามิเตอร์ K_p และ K_i ให้เหมาะสมกับแต่ละอัตราซุ่ม เพื่อลดความซับซ้อนของการหาค่าพารามิเตอร์ จึงทำการแบ่งอัตราซุ่มออกเป็นส่วนๆ จากนั้นทำการหาค่าพารามิเตอร์ค่า K_p และ K_i ในแต่ละอัตราซุ่มนั้นๆ ซึ่งการหาค่าพารามิเตอร์จะทำโดยวิธีการ Ziegler-Nicole ดังที่กล่าวมาแล้ว

ขั้นตอนการหาพารามิเตอร์ K_p และ K_i ด้วยวิธี Ziegler-Nicole

1. พนักงานทำการจับยึดเป้าหมายที่กำหนดไว้ โดยเป้าหมายต้องมีขนาดใกล้เคียงกับหน้าต่างและมีสีที่แตกต่างจากฉากหลังอย่างชัดเจน ขนาดหน้าต่างเป็นขนาดมาตรฐาน และชุดควบคุมกลิ้งต้องตั้งให้ขนานกับพื้นดิน การหันสายให้ขนาดกึ่งพื้นดิน การก้มเงยให้ตั้งฉากกับพื้นดิน เพื่อกันการรบกวนกันระหว่างแกน

2. ปรับอัตราซุ่มที่ต้องการตั้งค่า

3. ระบบจะกำหนดค่า K_p และ K_i ทางแกนนอน หรือ $K_{p_H} =$ ค่าตั้งต้น และ $K_{i_H} = 0$ และค่า K_p และ K_i ทางแกนตั้งหรือ $K_{p_V} = 0$ และ $K_{i_V} = 0$ เพื่อทำการหาค่าพารามิเตอร์ทางแนวราบ

4. หั่นกล่องออกจากกึ่งกลาง แล้วปล่อยให้ระบบควบคุมทำงาน เสมือนการใส่ค่าอินพุตแบบขั้น (step)

5. วัดว่าเกิดค่าสูงสุด-ต่ำสุดหรือไม่ เพื่อตรวจสอบการเกิดการแกว่ง

6. ถ้าเกิดการแกว่งอย่างต่อเนื่อง ตลอดช่วงเวลาที่กำหนด ให้บันทึกค่า K_c และ P_c ไปยังข้อ 8

7. เพิ่มค่า K_{p_H} กลับไปยังข้อ 4

8. คำนวณหาค่า K_{p_H} และ K_{i_H} ตามตาราง 3.1

9. กำหนดให้ $K_{p_H} = 0$ $K_{i_H} = 0$ $K_{p_V} =$ ค่าตั้งต้น $K_{i_V} = 0$ เพื่อทำการหาค่าพารามิเตอร์ทางแนวตั้ง

10. ทำตามข้อ 4 – 6 เพื่อหาค่า K_{p_V} และ K_{i_V}

11. ทำการหาค่าพารามิเตอร์แต่ละอัตราห้วงจนครบ

3.3.6 การไวด์อัฟ (windup) และการป้องกัน

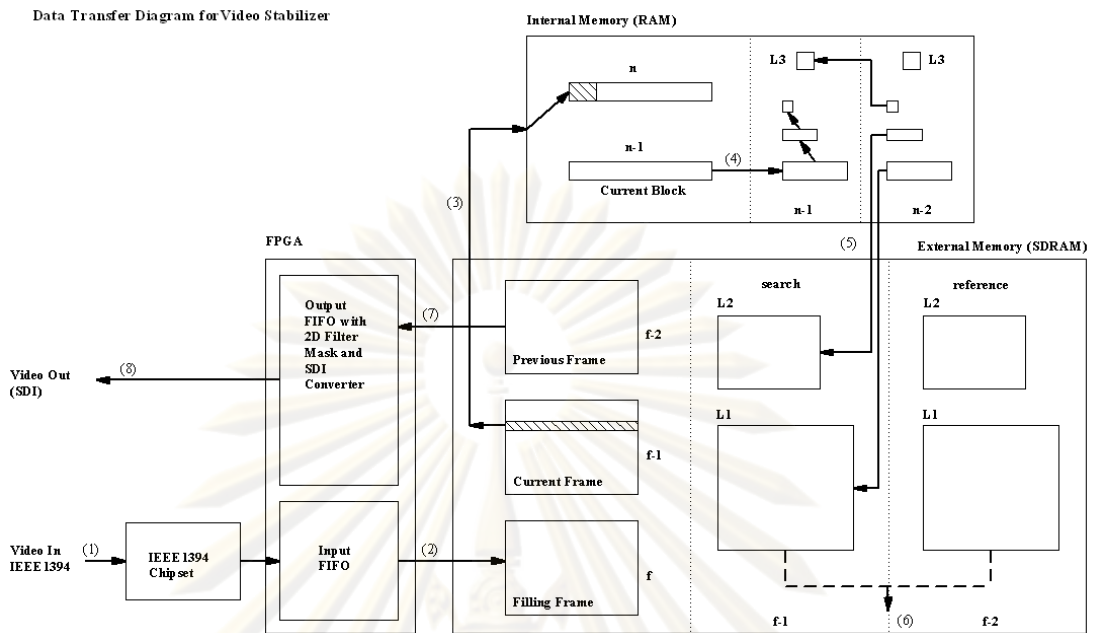
บางครั้งเมื่อระบบกัมส่ายตามเป้าหมายไปจนสุดขอบเขต จะทำให้เป้าหมายอยู่ห่างจากกึ่งกลางจอภาพ ซึ่งมีผลให้ตัวควบคุมแบบ I สะสมค่าผิดพลาดไว้เป็นจำนวนมาก ทำให้ค่า u_i มีค่าสูงมาก ถ้าบังเอิญเป้าหมายกลับเข้ามาอยู่ในขอบเขตการควบคุม จะเกิดการควบคุมเกินต้องการ (overshoot) ทำให้ระบบไม่เสถียร ซึ่งปรากฏการณ์ดังกล่าวเรียกว่าการไวด์อัฟ การแก้ปัญหาสามารถทำได้โดยกำหนดค่า u_i สูงสุดไว้ค่าหนึ่ง เพื่อให้ระบบไม่มีค่า u_i สูงเกินไป

3.4 โครงสร้างซอฟต์แวร์

หัวข้อนี้จะกล่าวถึงภาพโครงสร้างของซอฟต์แวร์ทั้งหมด ทั้งระบบปรับเสถียรวิดีโอ ที่ศน์ที่พัฒนาโดยอาจารย์สุวิทย์ นาคพีระยุทธและนายปิติฉัตร สุทธาโรจน์ โดยทุนวิสัยของ สกว. ตามที่กล่าวไว้ในบทที่ 1 และระบบติดตามเป้าหมาย ทั้งนี้ผู้วิจัยจึงนำเสนอไว้ ณ ที่นี้เพื่อให้ข้อมูลครบถ้วนสมบูรณ์

ในภาคผนวก ข เป็นโครงสร้างของโปรแกรมระบบทั้งระบบปรับเสถียรวิดีโอที่ศน์ และติดตามเป้าหมาย โดยแสดงถึงฟังก์ชันของโปรแกรม การทำงานของระบบมีองค์ประกอบหลัก อยู่สองประการคือ การโอนย้ายข้อมูล (data transfer) และการประมวลผล การโอนย้ายข้อมูลเป็นส่วนสำคัญมาก เนื่องจากการประมวลผลภาพวิดีโอที่ศน์แบบเวลาจริง มีข้อมูลที่เคลื่อนย้ายเป็นจำนวนมากด้วยความเร็วสูงมาก ดังนั้นการออกแบบระบบการโอนย้ายข้อมูลจำเป็นต้องมีประสิทธิภาพมากเช่นกัน

ในภาคผนวก ข ส่วนที่ 1 (PART 1) จะเป็นการตั้งค่าเริ่มต้นบอร์ด DSP (initialize C6317pt part) ซึ่งจะทำการตั้งค่าอุปกรณ์ต่างๆ บนบอร์ดและตัว DSP เช่นการตั้งค่า DMA และ EDMA [17] ทำการติดต่อและตรวจสอบกล่องหรืออุปกรณ์แปลงสัญญาณภาพที่ต่ออยู่กับบอร์ด ในส่วนที่ 2 เป็นส่วนประมวลผลหลัก โดยเริ่มจากการตั้งค่าในฟังก์ชัน Init_Before_MainProcess เริ่มจากระบบแปลงภาพหลายความละเอียด (multiresolution) เพื่อหาเวกเตอร์โมชัน (motion vector) ใน Init_MultiRes ซึ่งประกอบด้วยการตั้งค่า DMA เพื่อย้ายข้อมูลระหว่างหน่วยความจำ หรือเรจิสเตอร์ใน FPGA ต่างๆ กำหนดบริเวณหน่วยความจำที่เป็นที่พักข้อมูล (buffer) ทั้งที่เก็บภาพจากกล่องวิดีโอและที่เก็บภาพแบบหลายความละเอียด ดังแสดงในรูปที่ 3.14 ซึ่งจะกล่าวต่อไป จากนั้นทำการกำหนดตำแหน่งการทำ DMA ในฟังก์ชัน Init_DMA_target กำหนดตัวชี้ไปยังตำแหน่งหน่วยความจำภาพ Init_FieldParam การตั้งค่าอุปกรณ์แปลงสัญญาณภาพ Init_IEEE1394_Input_converter_part และการตั้งค่า FPGA ใน Init_FPGA_module ตั้งค่าเฟสล็อกลูป (phase lock loop) เพื่อทำการซิงโครไนส์อัตราเฟรมระหว่างภาพขาเข้ากับภาพขาออก ให้ตรงกันในฟังก์ชัน Open_PLL_timer Init_PLL_Param และ Init_PLL_time จากนั้นทำการตั้งค่าการรับส่งข้อมูลภาพ โดยทำการลบค่าในที่พักข้อมูล ตั้งค่าเวกเตอร์การขัดจังหวะ (interrupt vector) กำหนดอัตราเฟรม และเริ่มการรับข้อมูลภาพ



รูปที่ 3.14 โครงสร้างการโอนย้ายข้อมูลบนบอร์ด DSP

3.4.1 การโอนย้ายข้อมูล

เมื่อกำหนดค่าตั้งต้นแล้ว จะดำเนินการย้ายข้อมูลเพื่อการประมวลผล โดยเริ่มจากการรับภาพจากกล้องวิดีโอที่ส่งจากบัส (BUS) IEEE1394 ชิพเซต ดังในส่วนที่ (1) ในรูปที่ 3.14 โดยเป็นสัญญาณแบบ YCbCr แบบ 4:2:2 จากนั้นข้อมูลจะถูกเก็บเป็นแพคเกจ (packet) เล็กๆ ใน FPGA เมื่อที่พิกัดเต็ม FPGA จะทำการขัดจังหวะซีพียูเพื่อขอส่งข้อมูล การส่งข้อมูลจะส่งด้วยวิธี EDMA [17] (2) ไปพักไว้ที่หน่วยความจำภายนอกที่เป็นบริเวณเฟรมที่ f กระบวนการนี้จะทำไปเรื่อยๆจนกว่าข้อมูลจะเต็ม เมื่อเต็มแล้วจะทำการสลับตำแหน่งเก็บภาพดังรูปที่ 3.4 ในส่วนเฟรมปัจจุบัน (current frame) หรือเฟรม f-1 ซีพียูจะดึงข้อมูลผ่าน QDMA [17] (3) ไปสู่หน่วยความจำภายในครั้งละบล็อกๆละ 8 เส้นแนวระดับโดยนำข้อมูลทั้งความสว่างและสีเข้า ในขณะที่กำลังส่งข้อมูลบล็อกที่ n บล็อกที่ n-1 จะถูกประมวลโดยจะเลือกเฉพาะส่วนความสว่าง Y เท่านั้นเพื่อสร้างภาพหลายความละเอียด (4) เป็น 3 ระดับคือ L1 L2 และ L3 ตามลำดับ โดย L3 มีความละเอียดน้อยที่สุด ในขณะที่ประมวลผลบล็อกที่ n-1 บล็อกที่ n-2 ที่ถูกประมวลผลเสร็จแล้วจะถูกส่งไปเก็บไว้ในที่พัก (5) พิงก์ชันที่เกี่ยวข้องคือ MultiRes_1stFrame_BeforeStabilization ซึ่งจะสร้างภาพหลายความละเอียดของเฟรมเริ่มต้นโดยมีพิงก์ชันภายในคือ ImgMultiResolution และพิงก์ชันนี้จะใช้ในการประมวลผลภายในรูปหลักด้วย เมื่อการประมวลผลภาพ f-1 เสร็จแล้ว ข้อมูลภาพหลายความละเอียดของเฟรม f-1 และ f-2 จะถูกนำไปหาเวกเตอร์โมชัน (6) ซึ่งจะกล่าวต่อไป ใน

ส่วนสุดท้ายคือที่เฟรม f-2 หรือเฟรมก่อนหน้า (previous frame) จะถูกนำออกไปจากที่พัก ไปยัง FPGA (7) ที่ถูกโปรแกรมทำการส่งข้อมูลภาพขาออก โดยจะได้รับข้อมูลตำแหน่งเริ่มต้นของภาพ ข้อมูลภาพ ข้อมูลในการทำประมาณการภาพ (interpolate) โดยใช้หน้ากากตัวกรอง (filter mask) ขนาด 5x5 ทั้งนี้ FPGA จะส่งภาพขาออกโดยโปรโตคอลแบบ SDI (8)

3.4.2 การประมวลผลการปรับเสถียรวิดิทัศน์

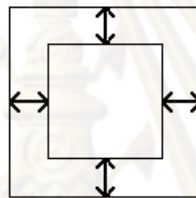
ประกอบด้วยการหาเวกเตอร์โมชัน (motion vector) ของภาพเฟรมปัจจุบันและก่อนหน้า การหาตำแหน่งในการแสดงผลและการปรับปรุงภาพ การหาเวกเตอร์โมชัน เริ่มต้นด้วยฟังก์ชัน `ImgMultiResolution` ที่จัดการระบบ DMA และ `BuildMultiResImgIntlvPlane` ที่เป็นตัวคำนวณ ตอนนี้จะกำหนดให้ข้อมูลภาพในเฟรม f-2 เป็นตัวอ้างอิง (reference) ที่ใช้ในการหาเวกเตอร์โมชัน ส่วนข้อมูลในเฟรม f-1 จะเป็นบริเวณค้นหา (search area) เมื่อได้ภาพหลายความละเอียด เริ่มจากระดับ L3 ในส่วนภาพอ้างอิงจะทำการตัดขอบออกให้ภาพอ้างอิงมีขนาดเล็กลงเพื่อให้เกิดบริเวณค้นหาขนาด NxM เมื่อเปรียบเทียบกับภาพ f-2 ทั้งเฟรม โดยตัดขอบซ้ายและขอบขวา N/2 ขอบบนและล่าง M/2 ตามลำดับ ฟังก์ชัน `Find_lv3_MaxCorrPoint` แล้วทำการค้นหากับภาพค้นหาดังรูปที่ 3.15 โดยวิธีการ `mad` หรือ `minimum absolute difference` [1] จำนวนการค้นหาจะเท่ากับ NxM เมื่อได้เวกเตอร์โมชันในระดับ L3 แล้ว ทำการหาเวกเตอร์โมชันในระดับ L2 โดยเริ่มจากเพิ่มขนาดเวกเตอร์โมชันเป็นสองเท่า เพื่อนำมากำหนดบริเวณค้นหาในระดับ L2 จะกำหนดพื้นที่อ้างอิงและค้นหาออกเป็นส่วนๆ ดังรูปที่ 3.16 โดยจะเรียกว่าบล็อกอ้างอิง (reference block) และบล็อกค้นหา (search block) ตามลำดับ โดยตำแหน่งของบล็อกอ้างอิงจะถูกกำหนดไว้ก่อน (predefine) กระจายสม่ำเสมอทั่วทั้งภาพจำนวน 32 บล็อก จากนั้นใช้เวกเตอร์โมชันเพื่อกำหนดตำแหน่งบล็อกค้นหาดังรูปที่ 3.17 ทั้งนี้บล็อกอ้างอิงจะใช้ขนาด 8x8 ส่วนบล็อกค้นหาจะมีขนาด 16x16 ในการหาเวกเตอร์โมชันระดับ L2 ถึง L0 จะต้องเลือกใช้บล็อกอ้างอิงที่มีขาวสารมากพอ นั่นคือค่าความแปรปรวนในบล็อกอ้างอิงต้องมีความมาก เพื่อให้การหาเวกเตอร์โมชันถูกต้องมากที่สุด ระบบจะทำการเลือกบล็อกอ้างอิงที่มีความแปรปรวนมากที่สุด 16 ชุดแรกจากบล็อกอ้างอิงทั้งหมด โดยหาจากค่า `sad` หรือ `sum absolute difference` ระหว่าง พิกเซลปัจจุบันและพิกเซลถัดไป ดังสมการ 3.54 ถึง 3.56

$$sad_{x_i} = \sum_{y=1}^N \sum_{x=1}^{N-1} |Y(x, y) - Y(x+1, y)| \quad (3.54)$$

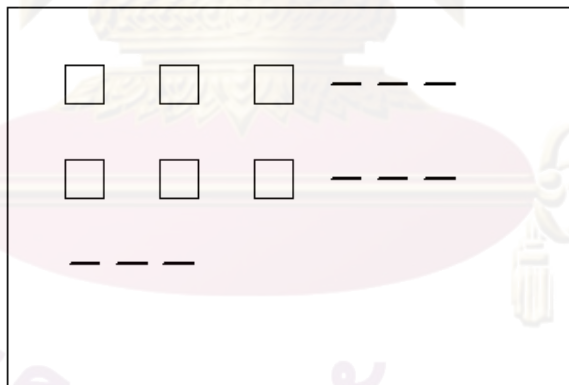
$$sad_{y_i} = \sum_{x=1}^N \sum_{y=1}^{N-1} |Y(x, y) - Y(x, y+1)| \quad (3.55)$$

$$sad_i = \max(sad_{x_i}, sad_{y_i}) \quad (3.56)$$

จากนั้นทำการเรียงลำดับ (sort) ค่าจากสูงไปต่ำ แล้วเลือกค่าสูงสุด 16 ลำดับแรก ในการหาเวกเตอร์โมชัน ครั้งนี้จะทำการรวมผล sad ของแต่ละตำแหน่งหรือ acc_sad เมื่อหาครบทุกบล็อกนำค่าต่ำสุดที่ได้มากำหนดหาเวกเตอร์โมชันในระดับ L2 ทำซ้ำกระบวนการเพื่อหาเวกเตอร์ L1 และ L0 ตามลำดับ โดยมีฟังก์ชัน Find_lv_MaxCorrPoint เป็นตัวดำเนินการ ดังรูปที่ 3.18 ในส่วนระดับ L0 เมื่อได้ค่า acc_sad แล้วจะนำข้อมูลนี้ไปหาตำแหน่งซัพพิกเซล (subpixel) ในฟังก์ชัน SubPixelMaxCorrPoint ทำให้ค่าเวกเตอร์โมชันมีความละเอียดถึง 0.1 พิกเซล ซึ่งเวกเตอร์โมชันที่ได้คือระยะทางระหว่างตำแหน่งภาพปัจจุบันกับภาพก่อนหน้า



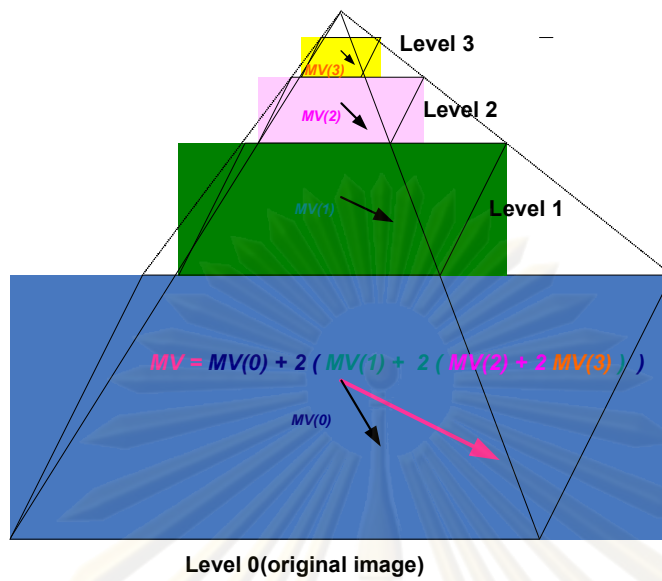
รูปที่ 3.15 การหาเวกเตอร์โมชันในภาพความละเอียดระดับ L3



รูปที่ 3.16 การกำหนดบล็อกอ้างอิงของภาพความละเอียดระดับ L2 L1 และ L0



รูปที่ 3.17 การใช้เวกเตอร์โมชันของระดับความละเอียดก่อนหน้าเพื่อกำหนดขอบเขตบล็อกค้นหา



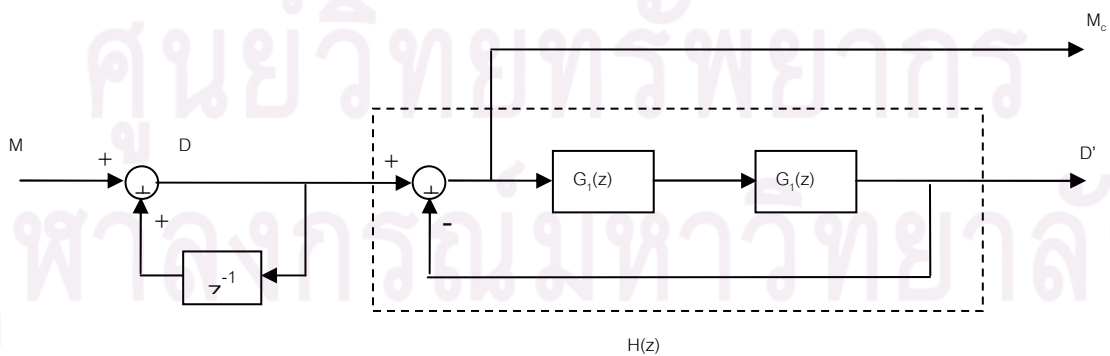
$$MV = MV(0) + 2 (MV(1) + 2 (MV(2) + 2 MV(3)))$$

Level 0(original image)

รูปที่ 3.18 แสดงการหาเวกเตอร์โมชันแบบใช้ภาพหลายความละเอียด

ในการปรับเสถียรวิดีโอที่ใช้นั้น คือการแก้การสั่นของภาพโดยนำภาพในเฟรมปัจจุบันนำมาแสดงในตำแหน่งเดียวกันกับภาพเฟรมก่อนหน้า ภาพที่ปรากฏจึงเสมือนนิ่งไม่สั่นไหวตามการสั่นของกล้อง แต่ถ้าพนักงานบังคับกล้องโดยทำการก้มส่าย ซึ่งเป็นความตั้งใจที่จะเปลี่ยนตำแหน่งภาพ ถ้าระบบพยายามนำภาพมาไว้ที่ตำแหน่งเดิมจะทำให้พนักงานไม่สามารถเห็นภาพในตำแหน่งใหม่ที่ต้องการได้ เพื่อแก้ปัญหานี้พิจารณาการควบคุมการก้มส่าย คือการเคลื่อนที่ตามวิถีที่ถือว่าเป็นการเคลื่อนที่ที่มีความถี่ต่ำมาก ดังนั้นระบบจะทำการคำนวณหาตำแหน่งภาพใหม่เพื่อปรับแก้ตำแหน่งภาพโดยฟังก์ชัน GetCompensateVector โดยมีทฤษฎีดังนี้

กำหนดตัวกรอง (filter) $H(z)$ ที่นำมาใช้ปรับแก้การเคลื่อนที่ของภาพ เพื่อนำผลลัพธ์เวกเตอร์โมชันที่ปรับแก้แล้ว (compensated motion vector) ไปใช้ประมาณตำแหน่งใหม่ของภาพที่เสถียรแล้วจะมีลักษณะดังรูปที่ 3.19



รูปที่ 3.19 รูปแบบตัวกรองปรับแก้การเคลื่อนที่ของภาพ

เมื่อ M คือเวกเตอร์โมชันของเฟรมปัจจุบันเทียบกับเฟรมที่แล้วที่ได้มาจากขั้นตอนแรก

D คือ Distance หรือ ระยะทางการเลื่อนของจุดกึ่งกลางภาพจากจุดเริ่มต้นของภาพ (เฟรมที่ 1)

D' คือ Stabilized distance หรือ ระยะทางการเลื่อนจุดกึ่งกลางของภาพที่มีการปรับเสถียรแล้ว

M_c คือ compensated motion vector ที่นำมาใช้ปรับตำแหน่งภาพในเฟรมปัจจุบันให้เสถียร

$$\text{โดย } H(z) = \frac{D'(z)}{D(z)} = \frac{G_1(z)G_2(z)}{1 + G_1(z)G_2(z)} \quad (3.57)$$

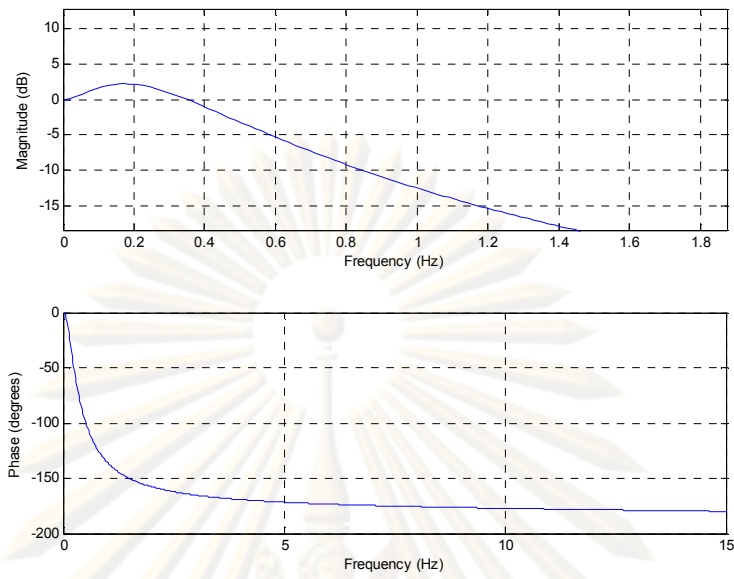
$$\text{เมื่อ } G_1(z) = \frac{(1+z^{-1})^2}{(1-z^{-1})^2} \quad (3.58)$$

$$\text{และ } G_2(z) = \hat{k} \frac{(aT+2) + (aT-2)z^{-1}}{(bT+2) + (bT-2)z^{-1}} \quad (3.59)$$

$$\text{โดย } \hat{k} = \frac{\pi f_c b T^2}{2}, \quad a = \frac{2\pi f_c}{\Delta}, \quad b = 2\pi f_c \Delta$$

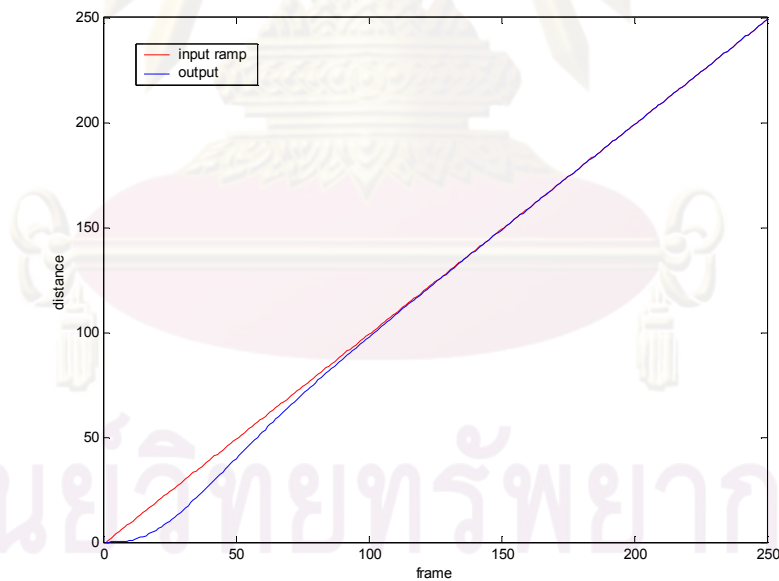
เมื่อ f_c คือ cutoff frequency ของ filter

Δ คือ ค่าระยะห่างจาก $\omega_c = 2\pi f_c$ ไปยัง zero(a) และ pole(b) เมื่อพิจารณาบน analog domain ซึ่งจะใช้ค่าเท่ากับ 3 โดยผลตอบสนองทางความถี่และเฟสของตัวกรองปรับเสถียร $H(z)$ เมื่อใช้ค่า $f_c = 0.3$ จะเป็นดังรูปที่ 3.20



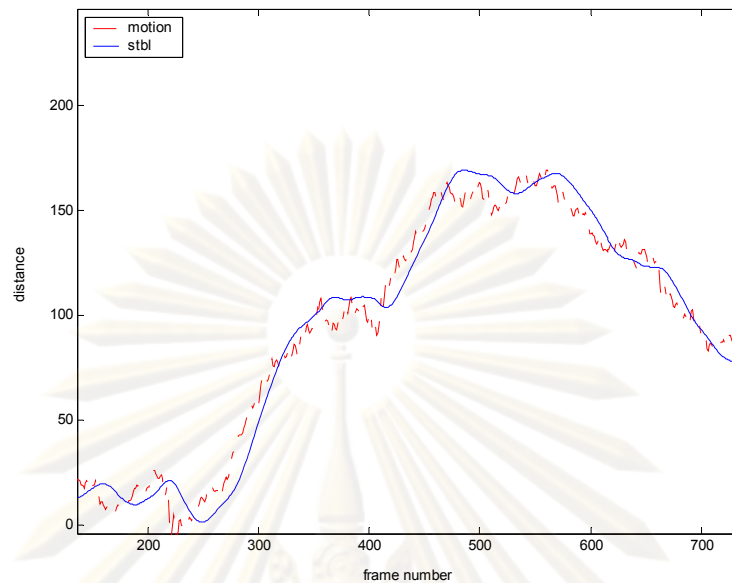
รูปที่ 3.20 ผลตอบสนองทางความถี่และเฟสของ filter ปรับเสถียรภาพ

โดยผลตอบสนองทางเวลาของ $H(z)$ ต่อ ramp input จะเป็นดังรูปที่ 3.21



รูปที่ 3.21 ผลตอบสนองทางเวลาของตัวกรองปรับเสถียรภาพเมื่อป้อน ramp input

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.22 ตัวอย่างการชดเชยการสั่นของภาพจากภาพที่ถ่ายมาจริง

เมื่อได้เวกเตอร์โมชันที่ได้รับการปรับแก้แล้ว ซีพียูจะทำการประมวลหาข้อมูลการประมาณเพื่อปรับแก้ภาพในส่วนซับพิกเซล ฟังก์ชัน `GetInterpolateParam_5x5` จะทำการประมาณค่าจะหน้าจากตัวกรองขนาด 5×5 ซึ่งมีวิธีการดังนี้ การหาจุดบนพิกัดใหม่ของภาพ Q เมื่อมีการเปลี่ยนตำแหน่งของภาพ ดังรูปที่ 3.23 จะใช้การประมาณเชิงเส้นจากจุดภาพบนพิกัดเดิม 4 จุดรอบๆ จุดภาพที่จะประมาณ (P_1, P_2, P_3 และ P_4) โดยจุด Q จะมีค่าดังสมการที่ 3.60

$$Q = V_3(H_1P_2 + H_2P_1) + V_1(H_1P_4 + H_2P_3) \quad (3.60)$$

เมื่อ H_i คือ ระยะห่างในแนวนอนจาก P_i ไปยังจุด Q

V_i คือ ระยะห่างในแนวตั้งจาก P_i ไปยังจุด Q

โดยที่ $H_1 + H_2 = 1$ และ $V_1 + V_3 = 1$

ส่วนการเพิ่มความละเอียดให้กับพิกัดใหม่ที่มีการเปลี่ยนตำแหน่ง จะประมาณค่า Q_1, Q_2, Q_3 และ Q_4 จากจุดบนพิกัดเก่ารอบๆ (P_1, P_2, P_3 และ P_4) ได้ดังสมการที่ 3.61

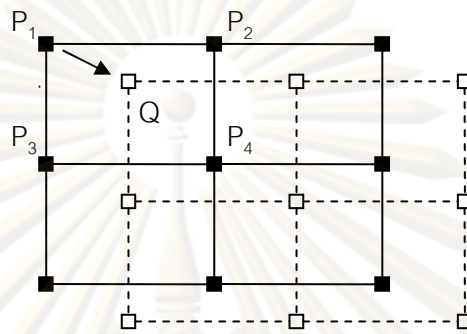
$$Q_i = V_{3i}(H_{1i}P_{2i} + H_{2i}P_{1i}) + V_{1i}(H_{1i}P_{4i} + H_{2i}P_{3i}) \quad (3.61)$$

โดย i มีค่าตั้งแต่ 1 ถึง 4

เมื่อ H_{ji} คือ ระยะห่างในแนวนอนจาก P_j ไปยังจุด Q_i

V_{ji} คือ ระยะห่างในแนวตั้งจาก P_j ไปยังจุด Q_i

โดยที่ $H_{1i} + H_{2i} = 1$ และ $V_{1i} + V_{3i} = 1$



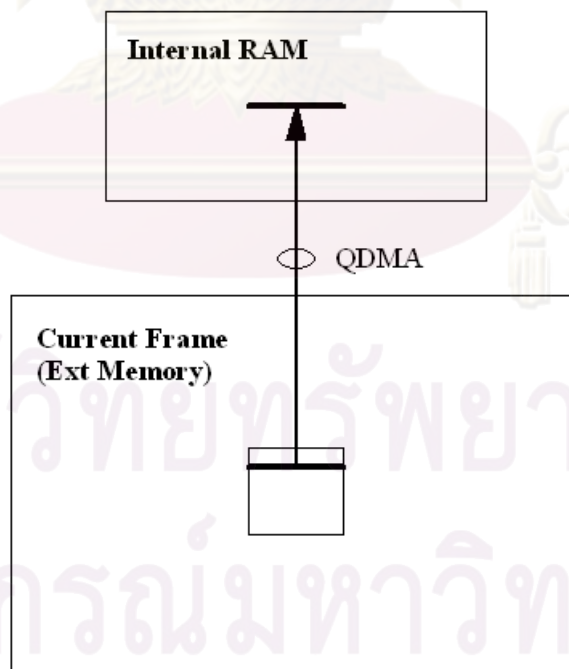
รูปที่ 3.23 การประมาณพิกัดใหม่ของภาพเมื่อมีการขดเชยการสั่น

จากนั้นส่งข้อมูลให้กับ FPGA โดยส่งข้อมูลภาพ ตำแหน่งภาพ และ ข้อมูลการประมาณเพื่อปรับแก้ภาพในส่วนซับพิกเซล ฟังก์ชัน `Get_SDI_Origin_And_PixelCondition`

3.4.3 การติดตามเป้าหมาย

จากรูปที่ 3.4 เฟรมปัจจุบันที่กำลังถูกประมวลผลจะมีตัวชี้ (pointer) ชื่อ `curr_param` ซึ่งไปเฟรมนี้เสมอ เมื่อพนักงานวางหน้าต่างลงบนเป้าหมาย ซึ่งถูกกำหนดด้วยจุด x y ความกว้าง (w) ความสูง (h) ไปยังตำแหน่งที่ต้องการและเริ่มการติดตามเป้าหมาย ในฟังก์ชัน `MeanShiftMain` ข้อมูลในหน้าต่างจะถูกส่งไปยังหน่วยความจำภายในที่ละเส้น ด้วย QDMA ดังรูปที่ 3.24 โดยถ้าเป็นเฟรมเริ่มต้น (initial frame) ระบบจะทำการสร้างฮิสโตแกรมของต้นแบบ (เป้าหมาย) ฟังก์ชัน `HistGen` ในกรณีมีน็พท์แบบทั่วไปจะสร้าง f_{wb} หรือ q ในมีน็พท์แบบเคอร์เนลเบสส์ จากนั้นทำการหาตำแหน่งเซนทรอยด์ โดยเริ่มกลับไปอ่านข้อมูลจากหน่วยความจำภายนอก ทำการคำนวณจนได้ตำแหน่งเป้าหมายที่ต้องการ ซึ่งอยู่ในฟังก์ชัน `MeanShiftWindowSearch` `MeanShiftPeaking` และ `MeanShiftLoop` ตามลำดับ สำหรับการแปลงปริภูมิสี ฟังก์ชัน `ColorConverter` จะถูกเรียกโดย `MeanShiftLoop` และ `HistGen` เนื่องจากข้อมูลสีของภาพวิดีโอที่เข้าเป็นแบบ YCbCr รูปแบบ 4:2:2 โดยมีการเรียงข้อมูลดังนี้ Cb0 Y0 Cr0 Y1 Cb1 Y2 Cr1... ในการแปลงค่าจะต้องนำค่า Y ในสองพิกเซลมาหาค่าเฉลี่ยเท่ากับ $(Y_i + Y_{i+1})/2$ ดังนั้นข้อมูลที่ได้จึงเป็นข้อมูลภาพความละเอียดระดับ L1 ทั้งนี้ส่วนของสีถูกแปลงเป็นแบบหลายความละเอียดก่อนนำเข้าแล้ว จากนั้นจะต้องแปลงเป็นปริภูมิสีแบบ RGB และมาเป็น

HSV ในที่สุด ในส่วน HS ที่จะถูกนำมาใช้จะแปลงค่ามาให้อยู่ในช่วง 0-255 เพื่อให้การประมวลผลมีประสิทธิภาพมากที่สุด สำหรับการสร้างฮีสโตแกรม จะนำเอาค่า H และ S ของพิกเซลนั้นมาหาร 16 จะทำให้ได้ค่าเวกเตอร์ $I=[H/16, V/16]^T$ แล้วนำค่านี้ไปสร้างอาเรย์แบบ 2 มิติ เพื่อเก็บค่าการกระจายในฮีสโตแกรม โดยการหาค่าความน่าจะเป็นของแต่ละส่วนน้ำหนักหรือเคอร์เนล ขึ้นอยู่กับวิธีที่ใช้ในการติดตามเป้าหมาย ส่วนการหาเซนทรอยด์จะนำค่า I จากพิกเซลเพื่อมาประมวลหาเซนทรอยด์ โดยนำข้อมูลจากฮีสโตแกรมเป้าหมายมาใช้ตามหัวข้อการติดตามเป้าหมาย โพรแกรมจะทำการหาเซนทรอยด์โดยวิธีวนลูป และจะหยุดตามกฎการหยุดสองประการคือเมื่อผลต่างของระยะเป้าหมายอยู่ในค่าที่กำหนด และประมวลผลจนถึงจำนวนลูปสูงสุด ในส่วนสุดท้ายซึ่งเป็นการควบคุมกล้อง บอร์ด DSP จะไม่ทำการควบคุมเองแต่จะให้ตัวควบคุมกล้อง ซึ่งในงานวิจัยนี้ใช้คอมพิวเตอร์เป็นตัวควบคุม ดังนั้นในส่วนบอร์ด DSP จะให้ค่าผลต่างของตำแหน่งกึ่งกลางหน้าต่างกับตำแหน่งกึ่งกลางภาพ ออกไปทางพอร์ตสื่อสารแบบอนุกรม RS-232 โดยส่งข้อมูลในรูปแบบตัวอักษร (text) ไปแต่ละเฟรม ซึ่งมีตัวอย่างดังนี้ $\langle -15,7 \rangle$ โดย -15 คือผลต่างบนแกนระดับ ส่วน 7 คือผลต่างบนแกนตั้ง ในส่วนตัวควบคุมจะนำค่าผลต่างไปคำนวณเพื่อนำไปสั่งกล้อง การติดต่อกับกล้องจะใช้โปรโตคอล VISCA ของบริษัท SONY เพื่อสั่งการก้มเงยสายและซูมกล้อง ทั้งนี้การใช้การสื่อสารชั้นกายภาพ (physical layer) ระหว่างกล้องกับคอมพิวเตอร์เป็นแบบ RS-232



รูปที่ 3.24 แสดงการย้ายข้อมูลจากหน้าต่างจากหน่วยความจำภายนอกสู่หน่วยความจำภายใน

บทที่ 4

ผลการวิจัย

ระบบติดตามเป้าที่กล่าวมานี้ถูกพัฒนาขึ้นโดยใช้โปรแกรมภาษา C++ บนสองระบบคือ

1. คอมพิวเตอร์แบบตั้งโต๊ะที่ทำงานโดยซีพียูเพนเทียม 4 ความเร็ว 2.4 GHz โดยเครื่องนำเข้าภาพจากเครื่องแปลงสัญญาณวีดิทัศน์แบบอะนาล็อกไปเป็นระบบภาพดิจิทัลที่มีปริภูมิสีแบบ RGB ที่มีขนาดภาพ 640x480 พิกเซล อัตราเฟรม (fps) เท่ากับ 25 เฟรมต่อวินาที สำหรับการทดสอบและการประเมินผลอัลกอริทึมต่างๆที่จะนำมาใช้

2. บอร์ดประมวลผลแบบ DSP ที่ใช้ซีพียูรุ่น TMS320C6713 ของบริษัทเท็กซัสอินสตรูเมนต์ส์ที่ทำงานที่ความถี่ 225 MHz โดยรับสัญญาณภาพแบบดิจิทัลที่มีปริภูมิสีแบบ YCrCb อัตราภาพ 50 เฟรมต่อวินาที ขนาดภาพ 768 พิกเซล กว้าง 576 เส้น

4.1 การติดตามเป้าหมาย

4.1.1 เวลาในการประมวลผล

ในการติดตามภาพด้วยวิธีมินิชิฟต์โดยอัลกอริทึมแบบทั่วไปจะได้ความเร็วในการประมวลผลดังตารางที่ 4.1 ส่วนอัลกอริทึมแบบเคอร์เนลเบสค์จะได้ผลดังตารางที่ 4.2

ตาราง 4.1 เวลาประมวลผลต่อการวนรอบ (time per iteration) ด้วยบอร์ด DSP ของการติดตามเป้าหมายด้วยวิธีมินิชิฟต์แบบทั่วไป

ขนาดหน้าต่าง(พิกเซล)*	30x30		50x50	
	YCbCr	HSV	YCbCr	HSV
ปริภูมิสี**	YCbCr	HSV	YCbCr	HSV
เวลาประมวลผล(ms)	0.68	1.08	1.47	2.76

*ขนาดหน้าต่างที่แท้จริงเท่ากับ 60x60 และ 100x100 ตามลำดับ ทั้งนี้ด้วยการใช้วิธีมัลติเรสโลูชัน (multiresolution) ลำดับหนึ่งทำให้การประมวลผลจึงเหลือครึ่ง

**YCrCb เป็นปริภูมิสีที่ถูกนำเข้านั้นจึงไม่ต้องเสียเวลาในการแปลง

ตาราง 4.2 เปรียบเทียบเวลาประมวลผลในการติดตามเป้าหมายทั้งสองวิธี

ขนาดหน้าต่าง(พิกเซล)*	30x30	50x50
เวลาประมวลผลด้วยวิธีมินซีฟต์แบบทั่วไป (ms)	1.08	2.76
เวลาประมวลผลด้วยวิธีมินซีฟต์แบบคอร์เนลเบสต์ (ms)	1.61	3.95

*ใช้ปริภูมิสี HSV

4.1.2 เปรียบเทียบสมรรถนะของการติดตามเป้าหมายทั้งสองวิธี

ในการทดลองยังพบว่าปริภูมิสีแบบ YCbCr จะมีประสิทธิภาพในการเกาะติดเป้าหมายต่ำมากเมื่อเทียบกับปริภูมิสีแบบ HSV โดยจะเกาะติดได้เฉพาะสีที่ความโดดเด่นมากเท่านั้น ดังปรากฏในหัวข้อ 3.2.1.2 การเลือกปริภูมิสี

ในตารางที่ 4.3 แสดงการเปรียบเทียบคะแนนความสำเร็จในการจับยึดเป้าหมาย (successful target locked) เมื่อพนักงานทำการครอบหน้าต่างลงบนเป้าหมาย และคะแนนความทนทานในการเกาะติดเป้าหมายโดยเทียบกับเป้าหมายแบบต่างๆ เนื่องจากการจับยึดเป้าหมายจะต้องทำการเลือกตำแหน่งที่แม่นยำการปรับขนาดที่เหมาะสมกับเป้าหมายที่เคลื่อนที่ซึ่งต้องใช้ทักษะในการทำงาน ทำให้การทดลองไม่สามารถทำซ้ำกันในแต่ละเงื่อนไขได้ ผู้วิจัยจึงใช้วิธีการให้คะแนนแทนการวัดผลเชิงตัวเลขแทน โดยให้พิสัยขนาดอยู่ระหว่าง 1 ถึง 5 ทั้งนี้ 1 คือสภาพการณ์ที่ระบบไม่สามารถทำงานได้ เช่น จับยึดหรือเกาะติดไม่ได้เลย พนักงานไม่ได้ประโยชน์ในการทำงาน และ 5 คือ พนักงานได้ประโยชน์เต็มที่นั่นคือ การจับยึดหรือเกาะติดมีความทนทานเชื่อถือได้เป็นเวลานานแม้มีการเปลี่ยนแปลงสภาพของเป้าหมาย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตาราง 4.3 คะแนนเปรียบเทียบความทนทานด้วยวิธีการติดตามเป้าหมายทั้งสองวิธี

	ความสำเร็จในการจับยึด		ความทนทานในการเกาะติด	
	มินซีฟต์แบบทั่วไป	มินซีฟต์แบบเคอร์เนลเบสท์	มินซีฟต์แบบทั่วไป	มินซีฟต์แบบเคอร์เนลเบสท์
เป้าหมายเป็นภาพเงา*	1	2	1	2
เป้าหมายมีหลายสี	2	5	2	4
เป้าหมายนิ่งสีแตกต่างจากฉากหลังชัดเจน	5	5	5	5
เป้าหมายเคลื่อนที่ฉากหลังนิ่งสีแตกต่างจากฉากหลังชัดเจน	4	5	4	4
เป้าหมายและฉากหลังนิ่งแต่ภาพเคลื่อนที่สีแตกต่างจากฉากหลังชัดเจน**	5	5	5	5
เป้าหมายเคลื่อนที่ฉากหลังนิ่งสีต่างกันในแถบเทา (gray scale)	2	4	2	4

* เช่นภาพอากาศยานที่บินสูงเมื่อมองขึ้นไปจะเห็นเป็นเงาเทาดำ

** ภาพที่เกิดจากการกัมม่ายกล้อง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



(ก)



(ข)



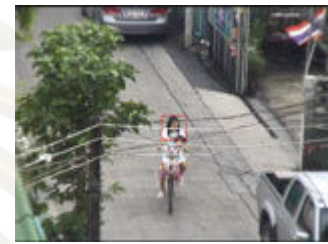
(ค)



(ง)



(จ)



(ฉ)

รูปที่ 4.1 แสดงการติดตามเป้าโดยวิธีมินชีฟต์แบบเคอร์เนลเบสส์

ในส่วนการตามภาพด้วยวิธีมินชีฟต์แบบทั่วไป การใช้ตัวถ่วงน้ำหนักฮิสโตแกรมแบบพาราโบลอยด์จะมีประสิทธิภาพดีเมื่อใช้กับภาพที่มีอัตราความยาวต่อความกว้างภาพต่ำ (low aspect ratio) เช่น รถ บ้าน เป็นต้น ดังปรากฏในรูปที่ 3.8 รูปที่ 3.8 (ก) จะแสดงส่วนที่หน้าต่างจับยึดเป้าหมายที่มีส่วนฉากหลังในปริมาณมาก ส่วนรูปที่ 3.8 (ข) แสดงให้เห็นว่าบริเวณกึ่งกลางเป้าหมายจะมีค่าความน่าจะเป็นสูงสุดเพราะน้ำหนักถูกให้ไว้สูงสุด ในขณะที่ตัวถ่วงน้ำหนักฮิสโตแกรมแบบเกาส์จะดีกับภาพที่มีอัตราส่วนความยาวต่อความกว้างสูง (high aspect ratio) เช่น อากาศยาน เรือ เป็นต้น ดังปรากฏในรูปที่ 4.3 ส่วนตัวถ่วงน้ำหนักแบบแบนจะเหมาะกับวัตถุที่มีสี่เหลี่ยมที่มีพื้นที่มาก ในการติดตามแบบเคอร์เนลเบสส์ จะพบว่าเคอร์เนลแบบอ็พานิคินคอฟนั้นใช้ได้กับเป้าหมายทุกรูปแบบ ทั้งนี้เนื่องจากการเปรียบเทียบการกระจายของสีในหน้าต่างซึ่งจะเปรียบเทียบทั้งเป้าหมายและฉากหลัง

4.1.3 ข้อเปรียบเทียบในการติดตามเป้าหมายแบบทั่วไปและแบบเคอร์เนลเบสส์

โดยจะเปรียบเทียบในสามหัวข้อคือ

4.1.3.1 ความสำเร็จในการยึดจับเป้าหมาย

จากผลการทดลองจะพบว่าวิธีการแบบเคอร์เนลเบสท์ จะมีอัตราส่วนความสำเร็จในการยึดจับเป้าหมายดีกว่ามินชีฟต์แบบทั่วไป ทั้งนี้เพราะการเปรียบเทียบการกระจายความหนาแน่น จะทำการเปรียบเทียบรูปแบบ (pattern) ของการกระจาย ที่มาจากพิกเซลทั้งเป้าหมายและฉากหลังที่อยู่ในหน้าต่างๆ ถึงแม้พิกเซลฉากหลังจะมีเป็นจำนวนมากก็จะมีผลกระทบต่อมากนัก ส่วนการถ่วงน้ำหนัก ถ้าจำนวนพิกเซลฉากหลังที่อยู่ในหน้าต่างๆมีปริมาณมาก ในเฟรมถัดมา ถ้าเป้าหมายเคลื่อนที่ไปแล้วปริมาณพิกเซลฉากหลังเพิ่มขึ้นมากจนทำให้ผลของของเวกเตอร์มินชีฟต์ที่ไปที่ฉากหลังแทนจึงทำให้สูญเสียการจับยึดเป้าหมายได้

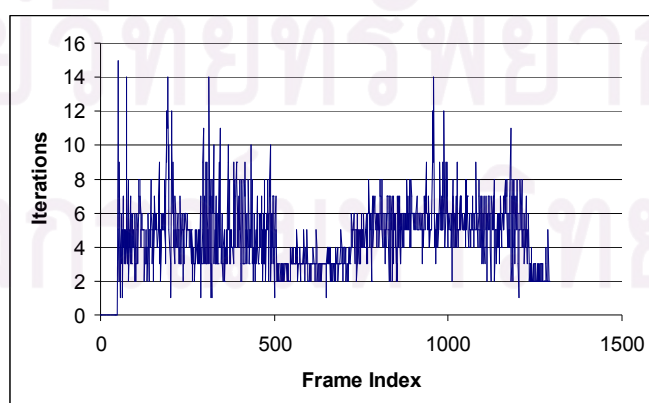
4.1.3.2 ความทนทานในการเกาะติดเป้าหมาย

หลังจากการจับยึดเป้าหมายแล้ว จะเห็นว่าการเกาะติดเป้าหมายของวิธีการแบบเคอร์เนลเบสท์จะดีกว่าแบบทั่วไป โดยเฉพาะเมื่อเป้าหมายที่มีหลายสีหรือมีลวดลาย (texture) มากหรือเป้าหมายที่มีแถบสีเทา ซึ่งเหตุผลก็เหมือนกับข้อแรก ส่วนความสามารถในการติดตามเป้าหมายที่มีสีชัดเจนทั้งสองวิธีมีประสิทธิภาพใกล้เคียง ปัญหาที่พบในการเกาะติดเป้าหมายด้วยวิธีการแบบเคอร์เนลเบสท์ โดยเฉพาะเมื่อการจับยึดที่มีส่วนของพิกเซลฉากหลังมาก ถ้าเกิดเป้าหมายเคลื่อนที่ไปบริเวณที่ฉากหลังเปลี่ยนสีไปจากเดิม เช่นจากถนนไปสู่สนามหญ้า ความทนทานในการติดตามจะลดลงและอาจสูญเสียการเกาะติดได้

4.1.3.3 ความเร็วในการประมวลผล

เวลาในการประมวลผลด้วยมินชีฟต์แบบเคอร์เนลเบสท์จะใช้เวลามากกว่าการประมวลผลด้วยวิธีแบบทั่วไปประมาณ 1.45 เท่า

4.1.4 จำนวนรอบต่อเฟรมในการติดตามเป้าหมาย



รูปที่ 4.2 แสดงจำนวนรอบในการติดตามเป้าหมายกับดัชนีเฟรม

รูปที่ 4.2 แสดงจำนวนการวนรอบ (iteration) กับดัชนีเฟรม ของภาพอากาศยาน ดังรูปที่ 4.3 ในช่วงต้นของการติดตามเป้าหมายจะสังเกตเห็นว่าจำนวนการวนรอบมีค่ามากทั้งนี้ เพราะหน้าต่างมีขนาดเล็ก แต่เมื่อปรับขนาดหน้าต่างให้เหมาะสมจำนวนการวนรอบจะลดลง (เฟรม 500 ถึง 700) ในช่วงเฟรม 900 มีจำนวนวนรอบที่สูงขึ้นทั้งนี้เกิดจากการสั่นไหวของกล้องทำให้เกิดการกระโดดดังรูปที่ 4.3 (ข) และ 4.3 (ค)

4.1.5 ความทนทานต่อการรบกวน



(ก) ดัชนีภาพที่ 107



(ข) ดัชนีภาพที่ 955

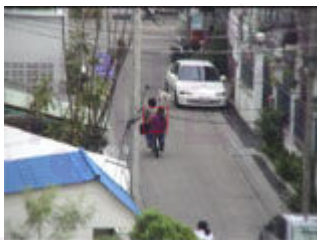


(ค) ดัชนีภาพที่ 958

รูปที่ 4.3 แสดงความทนทานต่อการรบกวนจากการเบลลและการกระโดด

ความทนทานต่อการเปลี่ยนขนาด และรูปร่าง รวมถึงการเบลลเนื่องจากการเปลี่ยนตำแหน่งกล้องอย่างรวดเร็ว ในรูปที่ 4.3 (ก) แสดงการเกาะติดภาพอากาศยาน รูปที่ 4.3 (ข) เป็นอากาศยานลำเดียวกันที่เปลี่ยนลักษณะ (orientation) และเกิดการเบลล รูปที่ 4.3 (ค) เป็นการย้ายตำแหน่งอย่างรวดเร็วของเป้าหมายเนื่องจากการสั่นจากการถ่ายภาพ สังเกตจากดัชนีเฟรม (ข) และ (ค) ที่เปลี่ยนไปเล็กน้อยแต่ตำแหน่งเป้าหมายเปลี่ยนไปมาก

4.1.6 ความทนทานต่อการ โคนบังบางส่วน (partial occlusion)



(ก)



(ข)



(ค)

รูปที่ 4.4 แสดงความทนทานเมื่อเป้าหมายที่ถูกบังบางส่วน

4.1.7 การปรับหน้าต่างโดยอัตโนมัติ



(ก)

(ข)

(ค)

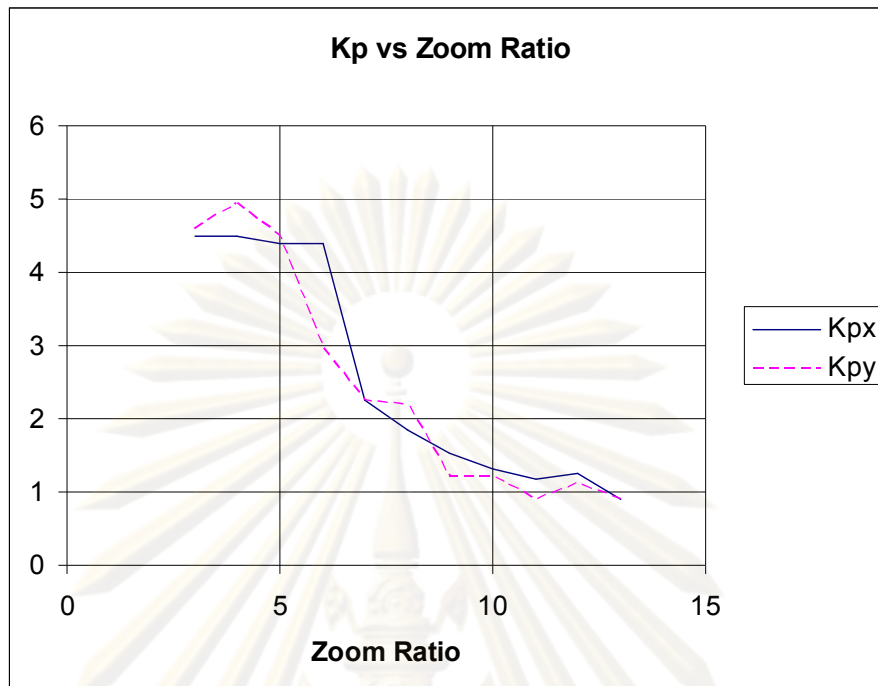
รูปที่ 4.5 แสดงการปรับหน้าต่างโดยอัตโนมัติของการติดตามด้วยวิธีการมินิซีพต์แบบทั่วไป

เมื่อพนักงานสั่งการระบบจะขยายหน้าต่างไปในแนวนอนตามลักษณะของเป้าหมาย แต่เนื่องจากถูกจำกัดขนาดการขยายจึงหยุดที่กึ่งการขยายของเป้าหมายดังรูปที่ 4.5(ข) เมื่อมีการสั่งการอีกครั้งการขยายจึงครอบคลุมเป้าหมายทั้งหมด 4.5(ค)

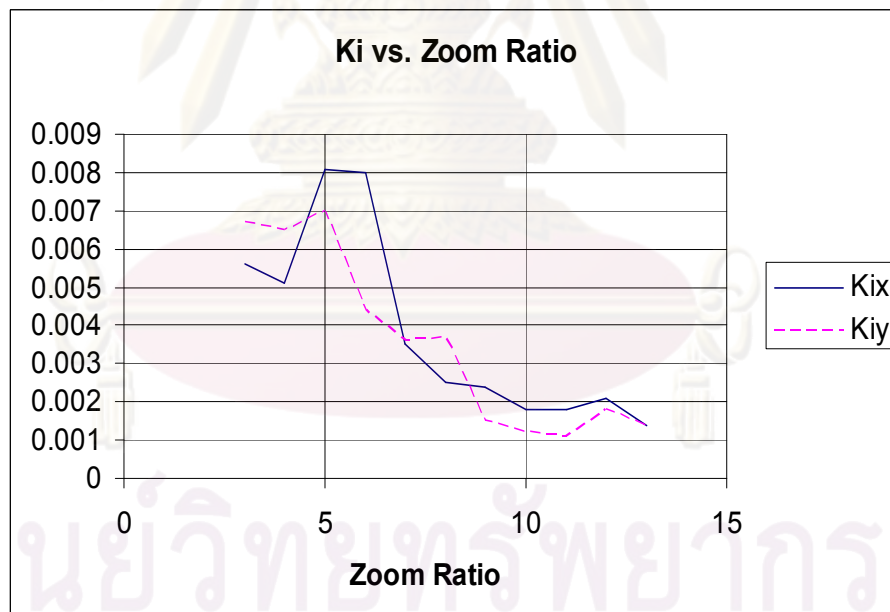
4.2 การหาค่าพารามิเตอร์ K_p และ K_i โดยอัตโนมัติด้วยวิธี Ziegler-Nicole

จากรูปที่ 4.6 และ 4.7 เป็นตัวอย่างค่า K_p และ K_i ที่ได้ด้วยวิธี Ziegler-Nicole กับอัตราซุม จะสังเกตได้ว่า ค่าพารามิเตอร์ที่ได้ไม่ต่อเนื่องและราบเรียบ (smooth) ทั้งนี้สาเหตุเกิดจากแถบการหยุด (stop band) ในบริเวณกึ่งกลางเป้าหมาย ทำให้บางครั้งการแกว่งหยุดลงกระทันหันเมื่อผ่านกึ่งกลางภาพ ทำให้ต้องปรับเพิ่มค่า K_c ทำให้ผลผิดพลาด อีกทั้งระบบสั่งการควบคุมกล็องเป็นแบบดิจิตัล ในการควบคุมจะเกิดการหน่วงเวลาแบบ latency ทำให้คาบเวลา T ไม่คงที่ ดังนั้นจึงอาจปรับแก้ได้ดังนี้

1. ทำการตั้งค่าหลายๆ ครั้ง แล้วนำค่าที่ได้มาเฉลี่ยกัน วิธีนี้จะใช้เวลานาน
2. สร้างแบบจำลองทางคณิตศาสตร์ที่เหมาะสมมาแทนผลลัพธ์ (curve fitting) แล้วนำไปใช้งานแทน



รูปที่ 4.6 ค่า K_p ที่ได้จากผลการหาค่าด้วยวิธี Ziegler-Nicole

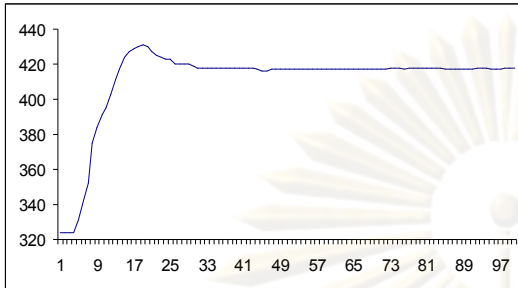


รูปที่ 4.7 ค่า K_i ที่ได้จากผลการหาค่าด้วยวิธี Ziegler-Nicole

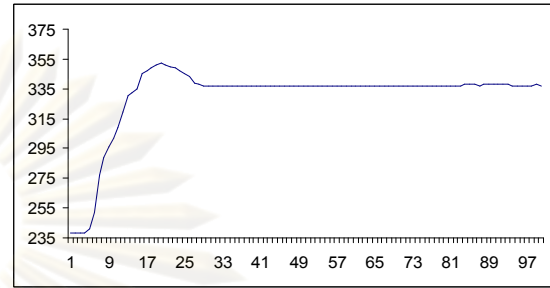
4.2.1 ตัวอย่างการตอบสนองต่ออินพุตแบบสแต็ป

รูปที่ 4.8 แสดงตัวอย่างการตอบสนองของกล่องทั้งแกนนอน (X) และแกนตั้ง (Y) ที่ใช้ค่า K_p และ K_i ที่ได้จากผลการหาค่าด้วยวิธี Ziegler-Nicole ที่อัตราซุ่ม 12 โดยที่ขนาดของสแต็ปเท่ากับ 100 และ 200 พิกเซลตามลำดับ การตอบสนองจะมีลักษณะของการควบคุมแบบ PI โดย

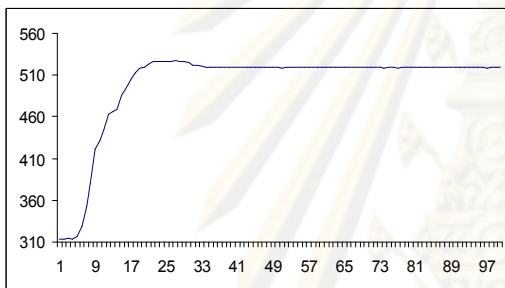
จะมีการเกิดโอเวอร์ชูทก่อนเข้าสู่ภาวะคงตัว (steady state) แต่จะมีลักษณะการเกิดความไม่ต่อเนื่องขึ้นเป็นช่วงๆ ทั้งนี้สาเหตุมาจากความไม่เป็นเชิงเส้นของระบบการส่ายเงยของกล้อง



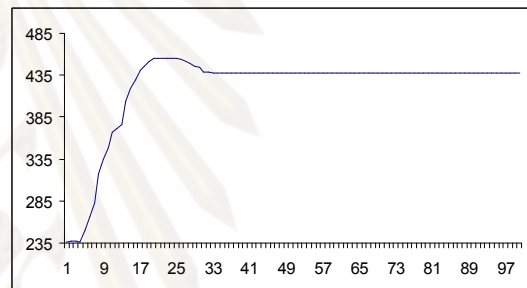
(ก) ผลตอบสนองแกนนอน สเต็ป 100 พิกเซล



(ข) ผลตอบสนองแกนตั้ง สเต็ป 100 พิกเซล



(ค) ผลตอบสนองแกนนอน สเต็ป 200 พิกเซล



(ง) ผลตอบสนองแกนตั้ง สเต็ป 200 พิกเซล

รูปที่ 4.8 แสดงตัวอย่างการตอบสนองต่ออินพุทแบบสเต็ปของตัวควบคุม

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

5.1 สรุปผลการวิจัย

การติดตามเป้าหมายภาพด้วยวิธีมินิซีฟต์ทั้งสองวิธี เป็นการติดตามร่องรอย (trail) ของคุณลักษณะของเป้าหมาย ซึ่งแปลงมาจากสีของเป้าหมาย ไม่ใช่การแยกแยะเป้าหมาย ออกจากฉากหลัง ดังนั้นการเปลี่ยนแปลงขนาด รูปร่าง หรือแม้แต่การรบกวนที่ทำให้ลักษณะของเป้าหมายเปลี่ยนไปจะไม่มีผลกระทบต่อความทนทานในการติดตามมากนัก ดังนั้นจึงเหมาะสมอย่างยิ่งสำหรับงานติดตามเป้าหมายในสภาพที่มีการรบกวนและการเปลี่ยนแปลงลักษณะของเป้าหมายสูง เหมาะกับการนำมาประยุกต์ใช้ในอากาศยานไร้คนบิน หรือแม้แต่ในพาหนะอื่นๆ

เปรียบเทียบการติดตามทั้งสองวิธีจะพบว่าการติดตามร่องรอยของเป้าหมายด้วยวิธีมินิซีฟต์แบบเคอร์เนลเบสท์จะมีความทนทานและเชื่อถือได้กว่าการติดตามด้วยวิธีมินิซีฟต์แบบทั่วไปทั้งการยึดจับเป้าหมายและการเกาะติดเป้าหมาย แต่วิธีแรกใช้เวลาในการประมวลผลมากกว่าทำให้ประสิทธิภาพลดลง แต่โดยภาพรวมแล้ววิธีมินิซีฟต์แบบเคอร์เนลเบสท์จะมีประสิทธิภาพดีกว่า และเป็นที่ยอมรับในการใช้งานมากกว่า

การควบคุมด้วยตัวควบคุมแบบ PID สามารถนำมาประยุกต์ใช้ในการควบคุมกล้องเพื่อติดตามเป้าหมาย ทั้งนี้เนื่องจากพารามิเตอร์ D นั้นไวต่อการรบกวน ดังนั้นจึงเลือกใช้เฉพาะ PI ซึ่งให้การควบคุมที่เหมาะสม ส่วนการตั้งค่าพารามิเตอร์โดยอัตโนมัติด้วยวิธี Ziegler-Nicole นั้นใช้ได้ดีในระดับหนึ่ง เพราะในบางกรณียังให้ผลที่ผิดพลาดบ้าง เนื่องจากระบบดำเนินการควบคุมมีความไม่เป็นเชิงเส้น และหน่วงเวลา (latency)

5.2 อภิปรายผลการวิจัย

ปัจจัยที่ส่งผลต่อความทนทานในการติดตามเป้าหมาย

1. ระยะขจัด (displacement) ของเป้าหมายในแต่เฟรมภาพ ถ้าเป้าหมายเคลื่อนที่เร็ว จะมีระยะขจัดสูง ร่องรอย (trail) ของเป้าหมายในการติดตามจะต่ำหรือหายไป ทำให้โอกาสสูญเสียการเกาะติดสูง

2. ความเร็วของเฟรมภาพ ถ้าความเร็วของเฟรมภาพต่อวินาทีสูง ระยะเวลาจัดของเป้าหมายที่ปรากฏในภาพจะลดลง โอกาสสูญเสียการติดตามจะต่ำ ทั้งนี้ยังขึ้นกับความเร็วในการประมวลผล

3. ถ้าความเร็วในการประมวลผลสูง โอกาสที่จะเกิดการประมวลผลข้ามเฟรมต่ำ จะทำให้โอกาสสูญเสียการติดตามต่ำ ถ้าการประมวลผลเร็วมาก และยังสามารถเพิ่มความเร็วเฟรมภาพต่อวินาทีได้ โอกาสสูญเสียการเกาะติดจะลดน้อยลง

4. ขนาดกรอบหน้าต่าง กรอบหน้าต่างที่มีขนาดใหญ่จะครอบคลุมพื้นที่เป้าหมายมาก ทำให้ติดตามร่องรอยได้ดี แต่ขนาดหน้าต่างที่ใหญ่จะเพิ่มเวลาในการประมวลผล ซึ่งอาจทำให้โอกาสสูญเสียการเกาะติดมากขึ้น ดังนั้นการปรับขนาดหน้าต่างที่เหมาะสมจะให้ผลดีที่สุด

5. ความแตกต่างระหว่างสีของเป้าหมายและสีของฉากหลัง ถ้ามีความแตกต่างน้อย โอกาสที่สีของฉากหลังจะทำการรบกวนการติดตามเป้าหมายเป็นไปได้สูง ลดความทนทานในการติดตามเป้าหมาย

6. การรบกวนจากสภาพแวดล้อม ทั้งการรบกวนเชิงกล เช่น การสั่นต่างๆ และการรบกวนการภาพ เช่น เงา หมอก ฯลฯ การรบกวนเชิงกลจะทำให้เกิดระยะขาดมากอีกทั้งยังทำให้เกิดการเบลอ ส่วนการรบกวนการภาพจะเพิ่มความผิดพลาดในปริภูมิคุณลักษณะ ทำให้ความทนทานลดลง

5.3 ข้อเสนอแนะ

การติดตามเป้าหมายด้วยวิธีมีนชิฟต์นั้นถึงแม้จะมีความทนทาน แต่ก็มีจุดอ่อน โดยเฉพาะในสภาพแวดล้อมที่มีการรบกวนมาก ในบางครั้งถ้ามีวัตถุในภาพที่มีลักษณะคล้ายเป้าหมายและอยู่ใกล้เป้าหมาย โอกาสที่จะเกิดการกระโดดจากเป้าหมายไปยังวัตถุอื่นมีโอกาสอยู่มาก การแก้ไขจะต้องผสมวิธีการแยกแยะและระบุตัวตนเพื่อเพิ่มประสิทธิภาพในการติดตามเป้าหมาย หรือผสมวิธีการติดตามวิถี (trajectory) ของเป้าหมาย ซึ่งวิธีการติดตามวิถีของเป้าหมายยังนำมาช่วยในการแก้ปัญหาการโดนบังบางส่วนและการโดนบังแบบชั่วคราว (temporary occlusion) ได้

นอกจากนี้เราสามารถเพิ่มความทนทานในการติดตามเป้าหมายโดยใช้การแยกแยะเป้าหมายจากฉากหลังแล้วใส่สีดำในฉากหลังดังในงานวิจัย [14] หรือแยกแยะและระบุ

เป้าหมาย (identification) วิธีดังกล่าวจะเพิ่มความทนทานมากขึ้น แต่จะทำให้การประมวลผลช้าลงเช่นกัน



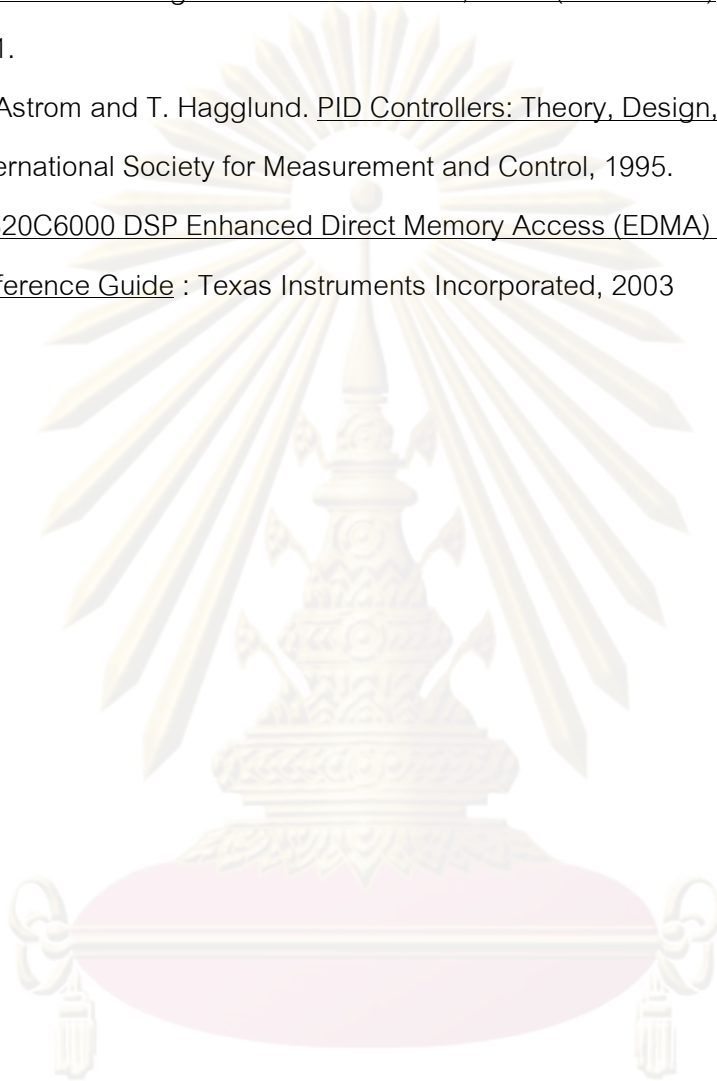
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

- [1] Yao Wang, Joern Ostermann, and Ya-Qin Zhang. Video Processing and Communications. : Prentice-Hall, 2002.
- [2] Wikipedia, Color Space[Online].2007. Available from:
http://en.wikipedia.org/wiki/color_space. [2009, May 19]
- [3] Wikipedia, RGB color model[Online].2007. Available from:
<http://en.wikipedia.org/wiki/RGB>. [2009, May 18]
- [4] Wikipedia, YCbCr[Online].2007. Available from: <http://en.wikipedia.org/wiki/YCbCr>.
[2009, Jun. 2]
- [5] Wikipedia, YUV[Online].2007. Available from: <http://en.wikipedia.org/wiki/YUV>.
[2008, Nov. 25]
- [6] Wikipedia, HSL and HSV[Online].2007. Available from:
http://en.wikipedia.org/wiki/HSL_color_space. [2009, Feb. 10]
- [7] K. Fukunaga. Introduction to Statistical Pattern Recognition. Boston : Academic Press, 1990.
- [8] D. Comaniciu and P. Meer. Mean Shift Analysis and Applications. In Proc. Intl Conf: Computer Vision (1999) : 1197-1203.
- [9] Y. Cheng. Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Machine Intell. (1995) : 790-799.
- [10] A.Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distribution. Bulletin of the Calcutta Mathematical Society (1943) : 99-110.
- [11] G.R.Bradschi. Computer Vision Face Tracking as a Component of a Perceptual User Interface. IEEE work. on Application Computer Vision (1998) : 214-219.
- [12] D. Comaniciu and P. Meer. Real-Time Tracking of Non-Rigid Objects using Mean Shift. Computer Vision and Pattern Recognition (2000) : 142-149.
- [13] D. Comaniciu, V. Ramesh and P. Meer. Kernel-based object tracking. IEEE Trans. On Pattern Analysis and Machine Intelligence (2003) : 564-575.
- [14] Xiaoping Chen, Shengsheng Yu and Zhilong Ma. An Improved Mean Shift Algorithm for Moving Object Tracking", Intelligent Control and Automation. 7th

World Congress 2008 (WCICA 2008) (June 2008) : 5111 - 5114.

- [15] Naoya Oshima, Takeshi Saitoh and Ryosuke Konishi. Automatic Moving Object Detection and Tracking with Mean Shift for Surveillance System. Intelligent Signal Processing and Communications, 2006. (ISPACS '06) (Dec. 2006) : 578 - 581.
- [16] K. J. Astrom and T. Hagglund. PID Controllers: Theory, Design, and Tuning. : International Society for Measurement and Control, 1995.
- [17] TMS320C6000 DSP Enhanced Direct Memory Access (EDMA) Controller Reference Guide : Texas Instruments Incorporated, 2003



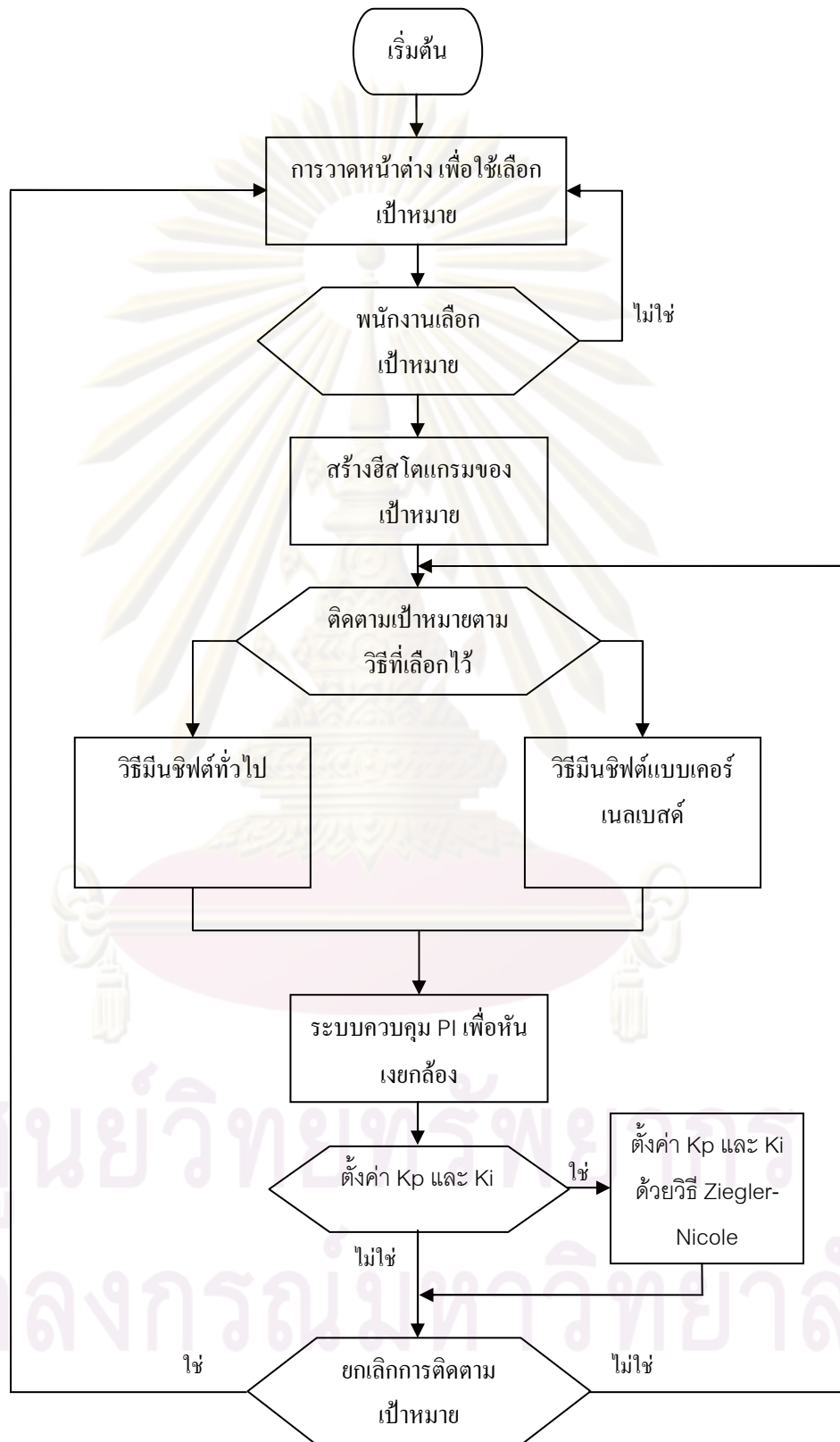
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก. ฟังก์ชันการทำงานของโปรแกรมระบบติดตามเป้าหมาย



ภาคผนวก ข. โครงสร้างโปรแกรมบนบอร์ด DSP

```

***** PART 1 *****
main program (File name: RLT_Mark_Target8_61.c)

**** initialized C6713Cpt part ****

InitSystem()                hardware dependent
                           initialization

CSL_init()                  init CSL library(TI's timer -
                           EDMA - others library)

Init_C6713Cpt_on_IEEE1394Bus_NodeInfo()  Init IEEE1394 chipset

GetNodeID(..)              get chip ID

sbiCauseBusReset()         issue bus reset to get self
                           IDs, etc.

AllocateAddress_on_IEEE1394Bus();      create addresses on IEEE.1394
bus and load Callback
func.when other nodes access
these addresses allocate
address range for incoming
transactions(from PC)

InitialCCRPixelink(..)     initial CCR value for acting
                           like DCAM(Pixelink camera
                           capabilities)

sbiAllocateAddressRange(..)  Allocate Address Range.

sbiCauseBusReset()

**** load camera ****

Load_CameraHandle_on_Bus_New(..)

pActCameraHandle = GetCam(0)      get camera(only '1' on bus for
now)

**** init EDMA module ****

EDMA_Reg_Assignment()          DMA initialize [DMA.h] for
program that use EDMA
parameters as variables

***** PART 2 *****
main video processing function(field-based)

**** initial for all process ****

```

```

Init_Before_MainProcess()          initial all module and all
                                   configuration before main loop
                                   processing

>InitUserAdjustableParam()        user adjustable parameter

>Init_MultiRes()                  init multires(finding motion
                                   vector) part

>DMA_PaRAM_InitMulti()            init DMA parameters for
                                   multires

>EDMA_reg_InitMulti()            EDMA register initialization

>ClearSAD_Map(..)                 Clear SAD memory map

>MultiRes_Load_CPU_IO_Buffer(..)  map N 8-line pointers for
                                   multires

>MultiRes_Load_FrameBuffer(..)   map memory buffer.to frame
                                   buffer

>Init_FieldParam()                set field parameter (map
                                   buffer frame to time frame)

>Init_IEEE1394_Input_Converter_part()  init video converter

>Init_FPGA_Module(..)            set for FPGA module

**** init PLL ****

>Open_PLL_Timer()                video input/output frame sync
                                   (50Hz approx.)

>Init_PLL_Param()

>Init_PLL_Timer()

>ClearAllCIPRVal()               <DMA.c>confirm CIPR's bits are
                                   set to '0'

>sbiIsoListen(..)                set IEEE1394 port

***** start transmission *****
***** setup board hardware for isochronous receive *****

1.) reset streaming port to
ensure empty FIFO
2.) set up for receive
operation

```

```

3.) isochronous interrupt
setup (preparation only)

4.) set up FIFO fill level for
almost full condition

>Set_Converter_InputChannel(..)      Input Composite 1 Input S-Video

>Set_Converter_VideoMode(..)         1) set mode

>Set_Converter_FieldRate(..)         2) set rate

>Enable_Converter_Tx(..)              3) enable converter

>Enable_StreamingPort_C6713Cpt()     enable port

**** main process loop ****

**** read initial field ****

MultiRes_1stFrame_BeforeStabilization(..) create multi-resolution for
(curr)frame that will be ref
frame for first time
stabilization

>ImgMultiResolution(..)              creating multiresolution image
routine, input : interleaved
'UYVY' -> 'L0', output :
'multiresolution L1,L2 and L3
buffers

>BuildMultiResImgIntlvPlane(..)      build multires blocks L1,L2
and L3 from N-8-line block

**** main loop ****

**** 1) DSP analysis motion vector ****

FindMotionVector(..)

1) MOTION VECTOR PART

>ImgMultiResolution(..)              Generate multiresolution

>Find_lv3_MaxCorrPoint(..)           [LEVEL 3] : only one block
search only!

>Find_lv_MaxCorrPoint(..)            [LEVEL 2] : get chosen block
from candidate block

>Find_lv_MaxCorrPoint(..)            [LEVEL 1] : get chosen block
from candidate block

```

```

>Find_lv_MaxCorrPoint(..)          [LEVEL 0] :
                                   "currFrameBuff_ERAM" is ref
                                   frame in YUV4:2:2 interleave
                                   version "lastFrameBuff_ERAM"
                                   is srch frame in YUV4:2:2
                                   interleave version

>SubPixelMaxCorrPoint(..)         [special] interpolate for
                                   subpixel max correlation point

**** 2) filter motion ****

GetCompensateVector(..)

**** 3) extract filter coefficient ****

GetInterpolateParam_5x5(..)       get value of
                                   'intpl_coeff[5][5]' 2D filter
                                   mask

Get_SDI_Origin_And_PixelCondition(..)

**** 6.1) Meanshift target tracking ****

                                   4) and 5) are cancelled 6) is
                                   old target tracking

----- Initial frame -----

Define_Rect(..)                   just define rectangular - not
                                   auto tracking now

MeanshiftMain(..)                 initial frame generate target
                                   model and found new window
                                   center

>HistGen(..)                      generate histogram for target
                                   model, initial frame only!

>MeanShiftWindowSearch(..)        search new window location

>MeanShiftPeaking(..)             checking for stop rules

>MeanShiftLoop(..)               mean shift calculation

----- 2nd and more frames -----

MeanshiftMain(..)

```

```
----- Output for camera control -----
control: output dx,dy for
camera control via rs-232

diff_center_ver = (..)          locate center window from
                                center of frame

diff_center_hor = (..)

----- Output to serial port -----
### function to send data to serial port ###
### function to read command from serial port or keyboard ###
### go to main loop ###

***** end of program *****
```

NOTE: '>' is the function called by above untab function



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค. บทความทางวิชาการที่ได้รับการเผยแพร่

Pravit Chinprasertsuk Suvit Nakpeerayuth and Somchai Jitapunkul Real Time Target Tracking for Moving Camera Running on DSP Platform , การประชุมทางวิศวกรรมไฟฟ้าครั้งที่ 32 (2009) pp.



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Real Time Target Tracking for Moving Camera Running on DSP Platform

Pravit Chinprasertsuk¹, Suvit Nakpeerayuth² and Somchai Jitapunkul³

Digital Signal Processing Research Laboratory, Department of Electrical Engineering, Chulalongkorn University
Bangkok, 10330, Thailand.

pravit1c@yahoo.com¹, Suvit.N@chula.ac.th², Somchai.J@chula.ac.th³

Abstract

Real time camera target tracking on moving platform, such as UAV, requires robustness against severe environment. In such case, Digital Signal Processing is the best choice compares to other processing platforms, due to its compactness, low power requirement, weight and processing power. Mean shift algorithm is simple and suitable for real time video processing. Mean shift video target tracking traces the feature of target in image frame. Tracing the target feature trail provides robustness against size and orientation changes, including blur. In this paper, two mean shift algorithms were implemented, conventional and kernel-based, on DSP video processing board for real time tracking. The results show that the processing time on DSP is much lower than desktop computer and kernel-based is more robust than conventional mean shift.

Keywords: target tracking / mean shift / DSP / real time

1. Introduction

Moving camera, especially, in mobile surveillance vehicles, such as patrol helicopter, unmanned aerial vehicle (UAV) and many other types of drone provide valuable data. Since these systems operate in mechanical noisy environment e.g. engine vibration, atmosphere turbulence noise, control noise, etc, the camera should be properly stabilized and controlled. A system that provides stability and control of moving camera is called stabilized gimbals. Target tracking capability is also an important part of gimbals system. It helps operator track the target, especially, during viewing in high zoom ratio or in noisy environment. Many video tracking algorithms have been published and employed. The purpose of this article is the development of fast real time video target tracking using digital signal processing board. Therefore, the algorithm must be simple but robust. Mean shift was chosen, due to its simplicity and robustness, as expressed in [1]. Originally, mean shift was developed by Fugunaka [2] to seek for peak value or mode in probability density distribution. In this paper, we implemented two mean shift algorithms. One is based on G.Bradski using conventional mean shift [1], another is kernel-based tracking mean shift [3,4] introduced by D. Comaniciu et al. Both algorithms were implemented on DSP video processing board for real time target tracking.

2. Color space selection

To track the target in real time, we consider only color or more specific chrominance of the target and background. This will save processing time and also reduce the effect of intensity varying. Therefore, color space must, carefully, be selected. There are varieties of color space in image and video processing. But three of them were

considered, RGB, YCbCr [5] and HSV. RGB is mostly found in monitor display system. But RGB is inappropriate, since processing must be done in all three channels. YCbCr and HSV have a separate channel for intensity; those are Y and V and can be ignored. YCbCr is the part of video streaming and available at the digital video input, without any conversion required, YCbCr should be selected. But after some researches, the results showed that CbCr contains less detail than HS space. Figure 1 and 2 illustrate the image of YCbCr and HSV which Y and V were fixed at a certain value. It can be seen that, much more detail appeared in HSV than YCbCr image. This detail is used to separate foreground from background. The variances of normalized $\|CbCr\|$ and $\|HS\|$ in figure 1 and 2 are $\sigma_{CbCr}^2 = 0.0009$ and $\sigma_{HS}^2 = 0.0256$. The ratio of $\sigma_{HS}^2 / \sigma_{CbCr}^2 = 28.44$. Examinations were performed in 8-bit scale uniform quantization. The results agreed to the assumption above. Two applications were developed for target tracking. One used CbCr another used HS and results again have confirmed that the application using HS was much robust in tracking than CbCr.



Figure 1 HSV color space image with fixed V value



Figure 2 YCbCr color space image with fixed Y value

3. Target tracking using conventional mean shift

Let \mathbf{Z}_i be a member of finite set in window of size h embedded in n -dimension Euclidean space and has PDF $f(\mathbf{z})$ that can be differentiated [1]. Mean location in h is [6]

$$m(\mathbf{z}) = \frac{\sum_{\mathbf{Z}_i \in h} \mathbf{Z}_i f(\mathbf{Z}_i)}{\sum_{\mathbf{Z}_i \in h} f(\mathbf{Z}_i)} \quad (1)$$

\mathbf{z} is the center location of window h . Mean shift vector is $m(\mathbf{z}) - \mathbf{z}$. Set window center location to $m(\mathbf{z})$ and repeat the process, finally, $m(\mathbf{z})$ will reach the peak of distribution. It can be seen that $m(\mathbf{z})$ always climb up the hill of the distribution and always reach the peak which has zero gradient. This guarantee convergence [1] an important characteristic of mean shift. To proof this, let $h \rightarrow 0$ then,

$$m(\mathbf{z}) - \mathbf{z} \approx \frac{f'(\mathbf{z})}{f(\mathbf{z})} \quad (2)$$

$f'(\mathbf{z})$ can be positive or negative and $m(\mathbf{z}) - \mathbf{z}$ has the same direction of $f'(\mathbf{z})$, means that the mean shift will always climb up. Set $m(\mathbf{z}) \rightarrow \mathbf{z}$ and repeat the calculation, finally, $f'(\mathbf{z})$ will always be zero or on top of distribution.

3.1 Histogram generation and weighting

In initial frame, hue (H) and saturation (S) components of pixels in the target window are used to generate the histogram. Let \mathbf{X}_i be the location vector of a pixel in the window surrounding the target of N pixels.

$$f_b = \frac{1}{N} \sum_{i=1}^N B(\mathbf{I}(\mathbf{X}_i)) \quad (3)$$

$$B(\mathbf{x}) = \begin{cases} 1, \mathbf{I}(\mathbf{x}) = b \in U \\ 0, otherwise \end{cases} \quad (4)$$

f_b is the histogram. \mathbf{I} is color vector composed of hue divided by 16 (H/16) and saturation divided by 16 (S/16), $\mathbf{I} = [H/16, S/16]^T$. U is 2-D bin of vector \mathbf{I} . For fast processing, bin size of 16x16 was implemented. When operator places the window over the target, usually, there are some parts of background left within the window. Especially, high aspect ratio or long-narrow shape target that most pixels in the window are background. In such case, the background colors become majority and the tracking will track background. To avoid this problem, weighted histogram must be implemented. Three types of weight functions were chosen.

1. Flat weight, this weight function is used as reference.

$$w(\mathbf{x}) = \begin{cases} 1, \mathbf{I}(\mathbf{x}) \in U \\ 0, otherwise \end{cases} \quad (5)$$

2. Paraboloid, for the objects that have low aspect ratio such as car, house, etc. The weight is mostly on the window center and gradually reduces to zero at the window boundary.

$$w(\mathbf{x}) = \begin{cases} r^2 - |\mathbf{x} - \mathbf{x}_0|^2, r = \frac{h}{2}, \mathbf{I}(\mathbf{x}) \in U \\ 0, otherwise \end{cases} \quad (6)$$

3. Gaussian, provide extra high weight at the window center. It is suitable for long-narrow object such as, ship, airplane, etc.

$$w(\mathbf{x}) = \begin{cases} \frac{1}{K} e^{-\frac{|\mathbf{x} - \mathbf{x}_0|^2}{2\sigma^2}}, \sigma^2 \propto h, \mathbf{I}(\mathbf{x}) \in U \\ 0, otherwise \end{cases} \quad (7)$$

K is the normalized factor and \mathbf{x}_0 is the center location. The weighted histogram can be expressed as

$$f_{wb} = \frac{1}{C} \sum_{i=1}^N w(\mathbf{X}_i) B(\mathbf{I}(\mathbf{X}_i)) \quad (8)$$

$$C = \sum_{i=1}^N w(\mathbf{X}_i) \quad (9)$$

3.2 Implementation of conventional mean shift

Tracking starts from reading color values of pixels in the window. For the sake of clarity, window coordinate (x, y) will be used. Start at the first frame, the pixels will be converted to feature space as

$$F(x, y) = f_{wb}(\mathbf{I}(x, y)) \quad (10)$$

$F(x, y)$ is weighted probability from histogram which was reflected to color of the target. If target is all red, the red pixels will have value of 1 while other colors will have value equal to 0. Then compute zeroth-moment and first-moment from pixels in window.

$$M_0 = \frac{1}{N} \sum_{x=1}^W \sum_{y=1}^L F(x, y) \quad (11)$$

$$M_{1x} = \frac{1}{N} \sum_{x=1}^W \sum_{y=1}^L xF(x, y) \quad (12)$$

$$M_{1y} = \frac{1}{N} \sum_{x=1}^W \sum_{y=1}^L yF(x, y) \quad (13)$$

L is window height, H is window width and $N=LxW$. The mean location or centroid \bar{x} , \bar{y} are

$$\bar{x} = M_{1x}/M_0 \quad (14)$$

$$\bar{y} = M_{1y}/M_0 \quad (15)$$

The tracking will be processed until a certain criteria or stop rule. To stop tracking in each frame, two rules have been implemented.

1. Error threshold

$$|\bar{x}_k - \bar{x}_{k-1}| \leq err \text{ and } |\bar{y}_k - \bar{y}_{k-1}| < err \quad (16)$$

Where k is the iteration. Error threshold of 4 pixels were applied.

2. Iteration count, by limit iteration counts to a certain number.

The mean location \bar{x}, \bar{y} will then be used as the center of new window.

4. Kernel-based target tracking mean shift

D. Comaniciu et al. [3,4] proposed a real-time tracking of non-rigid objects based on mean shift algorithm. The mean shift is employed to search for target candidate that is most similar to the target model, using metric based on Bhattacharya coefficient similarity function.

4.1 Target model

After target was selected, the color-based feature histogram of m bins is generated in initial frame. The target object model can be presented by

$$\hat{q}_u = C \sum_{i=1}^n k \left(\left\| \frac{\mathbf{x}_i^*}{h} \right\|^2 \right) \delta[b(\mathbf{x}_i^*) - u] \quad (17)$$

Where $\{\mathbf{x}_i^*\}_{i=1 \dots n}$ are the normalized pixels of target in the window center at 0. $k(\mathbf{x})$ is kernel profile which is isotropic, convex and monotonic decreasing, h is window size, δ is a Kronechker delta function, C is the normalized coefficient which satisfies $\sum_{u=1}^m \hat{q}_u = 1$, $b(\mathbf{x})$ is the mapping function of a pixel to feature bin number and u is the bin index number.

4.2 Target candidate

Let $\{\mathbf{x}_i\}_{i=1 \dots n}$ are the normalized pixels of target candidate, center at \mathbf{y} in current frame. The candidate model can be presented by

$$\hat{p}_u(\mathbf{y}) = C_h \sum_{i=1}^{n_h} k \left(\left\| \frac{\mathbf{y} - \mathbf{x}_i}{h} \right\|^2 \right) \delta[b(\mathbf{x}_i) - u] \quad (18)$$

Where n_h is the number of pixels in the candidate window, C_h is the normalized coefficient that must also satisfy $\sum_{u=1}^m \hat{p}_u = 1$

The similarity of $\hat{\mathbf{p}}(\mathbf{y})$ and $\hat{\mathbf{q}}$ can be measured by Bhattacharya coefficient $\rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\mathbf{y}) \hat{q}_u}$ which can be presented after linear Taylor series expansion as [4]

$$\rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\mathbf{y}_0) \hat{q}_u} + \frac{C_h}{2} \sum_{i=1}^{n_h} w_i k \left(\left\| \frac{\mathbf{y} - \mathbf{x}_i}{h} \right\|^2 \right) \quad (19)$$

$$w_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\mathbf{y}_0)}} \delta[b(\mathbf{x}_i) - u] \quad (20)$$

Where \mathbf{y}_0 is the estimated location of target in previous frame. $\hat{\mathbf{p}}(\mathbf{y}_0)$ is the feature histogram of target candidate center at \mathbf{y}_0 in current frame. The mean shift equation is

$$\hat{\mathbf{y}}_1 = \frac{\left[\sum_{i=1}^{n_h} \mathbf{x}_i w_i g \left(\left\| \frac{\mathbf{y}_0 - \mathbf{x}_i}{h} \right\|^2 \right) \right]}{\left[\sum_{i=1}^{n_h} w_i g \left(\left\| \frac{\mathbf{y}_0 - \mathbf{x}_i}{h} \right\|^2 \right) \right]} \quad (21)$$

Where $g(\mathbf{x})$ is the gradient of kernel profile $k(\mathbf{x})$. If $g(\mathbf{x}) = -k'(\mathbf{x})$, such as Epanechnikov kernel profile.

$$k_E(\|\mathbf{x}\|^2) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2)(1 - \|\mathbf{x}\|^2), & \|\mathbf{x}\| < 1 \\ 0, & \text{otherwise} \end{cases} \quad (22)$$

Then, mean shift is simplified to

$$\hat{\mathbf{y}}_1 = \frac{\left[\sum_{i=1}^{n_h} \mathbf{x}_i w_i \right]}{\left[\sum_{i=1}^{n_h} w_i \right]} \quad (23)$$

4.3 implementation

1. In initial frame, generate target model $\{\hat{q}_u\}_{u=1 \dots m}$
2. In current frame, compute $\{\hat{p}_u(\mathbf{y}_0)\}_{u=1 \dots m}$ and $\rho[\hat{\mathbf{p}}(\mathbf{y}_0), \hat{\mathbf{q}}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\mathbf{y}_0) \hat{q}_u}$
3. Derive for \mathbf{y}_1 using (23)
4. Update $\{\hat{p}_u(\mathbf{y}_1)\}_{u=1 \dots m}$ and $\rho[\hat{\mathbf{p}}(\mathbf{y}_1), \hat{\mathbf{q}}]$
5. If $\rho[\hat{\mathbf{p}}(\mathbf{y}_1), \hat{\mathbf{q}}] < \rho[\hat{\mathbf{p}}(\mathbf{y}_0), \hat{\mathbf{q}}]$ update $\mathbf{y}_1 \leftarrow 1/2(\mathbf{y}_0 + \mathbf{y}_1)$ update $\rho[\hat{\mathbf{p}}(\mathbf{y}_1), \hat{\mathbf{q}}]$.
6. If $\|\mathbf{y}_1 - \mathbf{y}_0\| < \text{err}$ or max iteration then stop, wait for new frame, go to 2
7. $\mathbf{y}_0 \leftarrow \mathbf{y}_1$ go to 2

5. Implementation Results

Target tracking applications were developed using C++ on two platforms. One on desktop computer with Pentium 2.4GHz CPU the image size was 640x480 pixels, 25 fps for evaluation and testing. Another, on DSP video processing board using Texas Instruments TMS320C6713 digital signal processor chip running at clock speed 225 MHz, video input rate 50 fields per second with 768 pixels by 576 lines.

Table 1 illustrates tracking time per iteration of conventional tracking. Table 2 compares processing time between two mean shift algorithms on DSP platform. The processing times measured on desktop PC are in order of tens millisecond, due to the Windows OS nature

Figure 3 illustrates the iterations vs. frame index of airplane shown on figure 4. Figure 4a window that tracked airplane was small and caused the high number of iterations during start frames. After window size adjustment the number of iterations had dropped (frame 500 to 700). During frame index 955 to 958, there was mechanical surge caused blurred (figure 4b) and jumped target (figure 4c). Figure 5 proves that mean shift tracking

is also robust to partial occlusion. It can be seen that mean shift tracking is robust against object orientation varying and deformation from mechanical noise and partial occlusion. The above results were achieved by both algorithms.

For weighted histogram in the conventional mean shift, the paraboloid is best suit for low aspect ratio object e.g. car, house etc. Where Gauss is best for long-narrow object e.g. ship, airplane. For kernel-based, Epanechnikov kernel profile is suitable for any type of target shape.

Comparing between these two algorithms, the kernel-based is more robust than conventional mean shift.

Windows size (pixel)*	30x30		50x50	
Color space**	YCbCr	HSV	YCbCr	HSV
Processing time (ms)	0.68	1.08	1.47	2.76

Table 1 Average processing time per iteration on DSP board

*Actual window sizes are 60x60 and 100x100 respectively. By using multi-resolution level 1 to reduce processed pixels.

**YCbCr is the default input color space, no conversion required.

Window size (pixel)*	30x30	50x50
Processing time of conventional tracking (ms)	1.08	2.76
Processing time of kernel-based tracking (ms)	1.61	3.95

Table 2 Comparing processing time of both algorithms

* HSV color space.

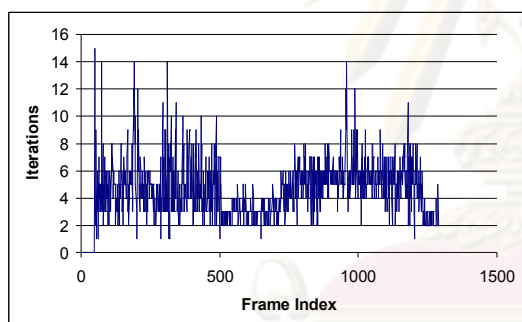


Figure 3 Iterations per frame of conventional tracking



Figure 4 Tracking an airplane



Figure 5 Partial occlusion

6. Discussions and conclusions

Mean shift, in video target tracking, traces the trail of feature of the target, which is converted from target colors, not by extracting target from the background. Therefore, changing in size and orientation will not much affect tracking robustness. Target displacement per frame effects tracking robustness. The displacement of target depends on target itself, frame rate, zoom ratio and environment noise. The higher the displacement the lower the tracking robustness. High processing speed and frame rate will compensate this weakness. Image noise also affects tracking performance.

Comparing between two mean shift algorithms, the kernel-based tracking is much robust than conventional. Due to the inherit pattern search, even the later required less processing time.

One major weakness of mean shift trail tracking, especially in noisy environment, if two objects have similar colors, chance of tracking to jump from the real target to similar object is high. Combining object segmentation [7], trajectory estimation, etc. to mean shift target tracking algorithm should improve this weakness.

References

- [1] G.R. Bradski, "Real Time Face and Object Tracking as a Component of a Perceptual User Interface", Proc. IEEE Workshop on Applications of Comp., pp. 214-219, 1998.
- [2] K. Fukunaga, "Introduction to Statistical Pattern Recognition", Academic Press, Boston, 1990.
- [3] D. Comaniciu and P. Meer, "Real-Time Tracking of Non-Rigid Objects using Mean Shift.", Computer Vision and Pattern Recognition pp. 142-149, 2000.
- [4] D. Comaniciu, V. Ramesh, P. Meer, "Kernel-based objects tracking", IEEE Trans. On Pattern Analysis and Machine Intelligence, pp. 564-575, 2003.
- [5] Yao Wang, Joern Ostermann, and Ya-Qin Zhang, "Video Processing and Communications", Prentice-Hall, 2002.
- [6] Y. Cheng, "Mean Shift, Mode Seeking, and Clustering", IEEE Trans. Pattern Anal. Machine Intell., vol.17, pp.790-799, 1995.
- [7] Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing Second Edition", Prentice-Hall Inc., 2002.
- [8] D. Comaniciu, Peter Meer, "Mean Shift Analysis and Applications", Department of Electrical and Computer Engineering, Rutgers University.
- [9] G.R. Bradski and Adrian Kaehler, "Learning OpenCV", O'Reilly Media Inc., 2008

ประวัติผู้เขียนวิทยานิพนธ์

นายประวิช ชินประเสริฐสุข เกิดวันที่ 26 พฤษภาคม พ.ศ. 2509 ที่จังหวัด กรุงเทพมหานคร เข้ารับการศึกษาในหลักสูตรวิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2528 สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมไฟฟ้าคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2531 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ในสังกัดห้องปฏิบัติการวิจัยกรรมวิธีสังเคราะห์นาโนเทคโนโลยี ภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2550



ศูนย์วิทยุทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย