



การปรับปรุงการติดตามวัตถุให้มีประสิทธิภาพโดยใช้วิธีการ Mean Shift
Improve Object Tracking Efficiency using Mean Shift Method

เมธี บุญสิน
Matee Boonsin

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Prince of Songkla University

2553

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

๗

เลขที่	TA1634	ว/ร	2553	พ.2
Bib Key	344686			
	20 ธ.ย. 2554			

(1)

ชื่อวิทยานิพนธ์	การปรับปรุงการติดตามวัตถุให้มีประสิทธิภาพโดยใช้วิธีการ Mean Shift
ผู้เขียน	นายเมธี บุญสิน
สาขาวิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2553

บทคัดย่อ

การติดตามวัตถุโดยใช้วิธีการ Mean Shift เป็นที่นิยมใช้อย่างกว้างขวาง แต่อย่างไรก็ตามวิธีนี้ยังมีปัญหาที่สำคัญ คือ พื้นที่หลังที่มีสีเดียวกับวัตถุ ซึ่งลดความถูกต้องในการติดตาม ในงานวิทยานิพนธ์นี้ได้เสนอแบบจำลอง Mean Shift Tracking Algorithm Using Adaptive Candidate (MST_AC) แบบจำลองนี้ได้ใช้ตำแหน่งพื้นที่หลังของเฟรมก่อนหน้าและเฟรมปัจจุบันเพื่อคำนวณแบบจำลองคู่แข่งใหม่ โดยหน้าต่างที่ใช้ในการติดตามมีขนาดคงที่ ข้อมูลที่ใช้ทดสอบได้มาจาก Performance Evaluation of Tracking and Surveillance Benchmark 2006 จากการทดลองแสดงให้เห็นว่า แบบจำลอง MST_AC ให้ค่าความถูกต้องในการติดตามมากกว่า Mean Shift ทั่วไป และ Mean Shift ที่ใช้วิธี Independent Component Analysis

Thesis Title	Improve Object Tracking Efficiency Using Mean Shift Method
Author	Mr. Matee Boonsin
Major Program	Computer Science
Academic Year	2010

ABSTRACT

Mean shift tracking is used widely for object tracking. However, one of the main problems is that the background on a current frame has the same color as the target (object) which can reduce the correctness of tracking. This thesis proposes Improving of Mean Shift Tracking Algorithm Using Adaptive Candidate Model. This algorithm uses the background positions on the previous frame and the current frame to compute the new candidate model. The window size is fixed. The dataset is received from Performance Evaluation of Tracking and Surveillance Benchmark 2006. The experimental results show that the proposed MST_AC model receives higher correctness than those of the traditional mean shift algorithm and Independent Component Analysis Mean Shift Algorithm.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความช่วยเหลือและสนับสนุนจากบุคคลหลายฝ่าย ซึ่งผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูง คือ

ผศ.ดร. วิภาดา เวทย์ประสิทธิ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่กรุณาให้คำปรึกษาแนะนำและช่วยเหลือในการแก้ปัญหาต่างๆ พร้อมทั้งตรวจทานและแก้ไขวิทยานิพนธ์ให้แก่ผู้วิจัย

ดร.ลัดดา ปรีชาวีรกุล อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ที่กรุณาให้ข้อเสนอแนะต่างๆ รวมทั้งตรวจทานและแก้ไขวิทยานิพนธ์ให้แก่ผู้วิจัย

ดร.ฐิมาพร เพชรแก้วและ รศ.ดร. สาธิต อินทจักร์ ที่กรุณาช่วยตรวจทานและแก้ไขวิทยานิพนธ์ให้มีความสมบูรณ์

อาจารย์ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ทุกท่าน ที่ให้ความรู้ทางด้านวิชาการ ซึ่งสามารถนำมาใช้ในการทำวิทยานิพนธ์ได้อย่างดียิ่ง

เพื่อนๆ พี่ๆ และน้องๆ ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ ที่ให้คำปรึกษาและช่วยเหลือในการทำวิทยานิพนธ์ด้วยดีเสมอมา

ผู้วิจัยขอขอบคุณทุกท่านเป็นอย่างสูงมา ณ โอกาสนี้

เมธี บุญสิน

สารบัญ

	หน้า
สารบัญ.....	(6)
รายการตาราง.....	(8)
รายการภาพประกอบ.....	(9)
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมา	1
1.2 การตรวจเอกสาร.....	2
1.3 วัตถุประสงค์ของงานวิจัย.....	24
1.4 ขอบเขตและวิธีการดำเนินงาน.....	24
1.5 ขั้นตอนและระยะเวลาการดำเนินงาน	24
1.6 สถานที่และเครื่องมือที่ใช้.....	25
1.7 ประโยชน์ที่คาดว่าจะได้รับของการวิจัย	26
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	27
2.1 การประมวลผลภาพ	27
2.2 การวิเคราะห์วิดีโอ (Video Analysis).....	29
2.3 การจดจำรูปแบบ (Pattern Recognition).....	30
2.4 การติดตามวัตถุ (Object Tracking).....	31
บทที่ 3 แบบจำลอง Mean Shift Tracking Algorithm Using Adaptive Candidate.....	34
3.1 การเริ่มต้น	35
3.2 การสร้างแบบจำลอง.....	36
3.3 การประเมินความคล้าย	37
3.4 การเคลื่อนไปยังตำแหน่งใหม่	37
3.5 การกำหนดค่าพื้นหลัง	37
3.6 การปรับตำแหน่งของหน้าต่าง	46
บทที่ 4 ผลการทดลองและวิจารณ์	53
4.1 ชุดข้อมูลที่ใช้ในการทดลอง.....	53
4.2 การทดลองเปรียบเทียบขั้นตอนวิธีการติดตาม	54
4.3 ผลการทดลอง	56
4.4 เปรียบเทียบความถูกต้องในการติดตามโดยเปลี่ยนจำนวน bin	87
4.5 เปรียบเทียบความถูกต้องในการติดตามโดยเปลี่ยนความคมชัด (Resolution)	98
	(6)

สารบัญ (ต่อ)

	หน้า
4.6 เปรียบเทียบผลการทดลองและวิจารณ์ผลการทดลอง	106
บทที่ 5 บทสรุปและข้อเสนอแนะ	121
5.1 สรุปผลการวิจัย	121
5.2 ปัญหาและอุปสรรค	123
5.3 ข้อเสนอแนะและงานในอนาคต	123
บรรณานุกรม.....	124
ภาคผนวก.....	127
ก คู่มือการใช้โปรแกรม Analyze Tracking.....	128
ข ผลงานตีพิมพ์	137
ประวัติผู้เขียน.....	143

รายการตาราง

ตาราง	หน้า
1.1 ตัวอย่าง Kernel (Simakov, 2008).....	16
1.2 เปรียบเทียบข้อดี ข้อเสีย ของการติดตามโดยใช้วิธี Mean Shift.....	19
1.3 วิวัฒนาการการติดตามโดยวิธี Mean Shift.....	22
1.4 แผนการดำเนินการวิจัย.....	25
2.1 เปรียบเทียบขั้นตอนวิธีของการติดตามวัตถุ.....	32
3.1 การหาตำแหน่งพื้นหลัง.....	38
3.2 ตัวอย่างการปรับตำแหน่งของ \hat{z}_2 ตามทิศทางของวัตถุที่เป็นไปได้ทุกกรณี.....	40
3.3 เปรียบเทียบขั้นตอนวิธีของการติดตามวัตถุ.....	48
4.1 รายละเอียดชุดข้อมูลทดสอบ.....	54
4.2 รายละเอียดชุดข้อมูลทดสอบใหม่.....	98
4.3 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธี.....	106
4.4 เวลาเฉลี่ยที่ใช้ต่อการทำงาน 1 รอบของแต่ละขั้นตอนวิธี.....	107
4.5 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธี.....	110
4.6 เวลาเฉลี่ยที่ใช้ต่อการทำงาน 1 รอบของแต่ละขั้นตอนวิธี.....	111
4.7 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธี.....	116
4.8 เวลาเฉลี่ยที่ใช้ต่อการทำงาน 1 รอบของแต่ละขั้นตอนวิธี.....	116
5.1 ความถูกต้องในการติดตามของ MST_AC ในแต่ละคุณลักษณะของข้อมูล.....	122
5.2 ความถูกต้องในการติดตามและเวลาการทำงานของ MST_AC เมื่อปรับเปลี่ยน ขนาด bin.....	122
5.3 ความถูกต้องในการติดตามและเวลาการทำงานของ MST_AC เมื่อปรับเปลี่ยน ความคมชัดของภาพ.....	123

รายการภาพประกอบ

ภาพประกอบ	หน้า
1.1 ความหมายของคอมพิวเตอร์วิทัศน์.....	3
1.2 ระบบสี RGB.....	3
1.3 ระบบสี HSV	4
1.4 ระบบสี YIQ	4
1.5 การแสดงวัตถุในรูปแบบต่างๆ	5
1.6 ประเภทของการติดตาม.....	6
1.7 ตัวอย่างการติดตามแต่ละประเภท.....	7
1.8 แสดงข้อจำกัดในการเคลื่อนที่	8
1.9 ขั้นตอนการทำงานของ Mean Shift	9
1.10 การติดตามวัตถุด้วยวิธีการ Mean Shift	11
1.11 การสร้างแบบจำลอง.....	11
1.12 การวัดความคล้ายกันระหว่างแบบจำลองเป้าหมายกับคู่แข่ง	12
1.13 ตัวอย่างการสร้างแบบจำลอง	14
1.14 ผลจากการปรับหน้าตาด้วยวิธีต่าง ๆ.....	19
2.1 องค์ประกอบของภาพ.....	27
2.2 ตัวอย่างงานทางด้านการประมวลผลภาพ.....	28
2.3 สัญญาณจากหลายแหล่ง	28
2.4 Spectrum ของคลื่นแม่เหล็กไฟฟ้า.....	29
2.5 องค์ประกอบของวิดีโอ	29
2.6 การสร้างแบบจำลอง.....	30
2.7 การเปรียบเทียบความคล้ายระหว่าง Target Model และ Candidate Model....	31

รายการภาพประกอบ (ต่อ)

ภาพประกอบ	หน้า
3.1 แบบจำลองการทำงาน MST_AC.....	34
3.2 ตัวอย่างการเลือกพื้นที่ตำแหน่งเป้าหมาย.....	35
3.3 การหาตำแหน่ง \hat{y}_2 ในกรณี $\Delta x > 0$	41
3.4 การหาตำแหน่ง \hat{y}_2 ในกรณี $\Delta x < 0$	42
3.5 การหาตำแหน่ง \hat{y}_2 ในกรณี $\Delta x = 0 \Delta y > 0$	43
3.6 การหาตำแหน่ง \hat{y}_2 ในกรณี $\Delta x = 0 \Delta y < 0$	44
3.7 การหาตำแหน่ง \hat{y}_2 ในกรณี $\Delta x = 0 \Delta y = 0$	45
4.1 ขั้นตอนการทดสอบการติดตาม.....	55
4.2 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport1..	57
4.3 ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 1.....	58
4.4 ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC ลำดับของเฟรม 1018 1020 1030 1040 ตามแนวตั้งจากบนลงล่าง โดยใช้ข้อมูลทดสอบ Airport 1	59
4.5 ตัวอย่างการติดตามที่ยังมีปัญหของขั้นตอนวิธี MST_AC ลำดับของเฟรม 928 930 932 934 ตามแนวตั้งจากบนลงล่าง โดยใช้ข้อมูลทดสอบ Airport 1	60
4.6 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล Airport 1	61
4.7 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 2	62
4.8 ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 2	62
4.9 ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC ลำดับเฟรม 2700 2710 2719 และ 2735 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล Airport 2.....	63
4.10 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล Airport 2	64
4.11 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 3	65
4.12 ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 3.....	65
4.13 ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC ลำดับเฟรม 2765 2767 2769 และ 2771 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล Airport 3	66
4.14 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport 3.....	67

รายการภาพประกอบ (ต่อ)

ภาพประกอบ		หน้า
4.15	ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล Airport 4.....	68
4.16	ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 4.....	68
4.17	ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC ลำดับเฟรม 1070 1072 1074 และ 1076 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล Airport 4.....	69
4.18	เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอน โดยใช้ข้อมูลทดสอบ Airport 4	70
4.19	ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ ComSci 1.	71
4.20	ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ ComSci 1.....	72
4.21	ตัวอย่างการติดตามของขั้นตอนวิธี ลำดับเฟรม 224 264 324 404 484 และ 530 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล ComSci 1.....	73
4.22	เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล ComSci 1	74
4.23	ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล ComSci 2	75
4.24	ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ ComSci 2.....	76
4.25	ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC ลำดับเฟรม 394 396 398 และ 400 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล ComSci 2.....	77
4.26	ตัวอย่างการติดตามที่ยังมีปัญหาของขั้นตอนวิธี MST_AC ลำดับเฟรม 201 210 212 214 216 และ 218 ตามแนวตั้งจากบนลงล่าง	78
4.27	เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธีข้อมูลทดสอบ ComSci 2 ...	79
4.28	ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธีสำหรับ ข้อมูลทดสอบ ComSci 3	80
4.29	ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ ComSci 3.....	81
4.30	ตัวอย่างการติดตามที่ยังมีปัญหา (วงกลมหมายเลข 1) ลำดับเฟรม 52 54 และ 56 ตามแนวตั้งจากบนลงล่าง โดยใช้ข้อมูลทดสอบ ComSci 3.....	82

รายการภาพประกอบ (ต่อ)

ภาพประกอบ	หน้า
4.31 ตัวอย่างการติดตามที่ยังมีปัญหา (วงกลมหมายเลข 2) ลำดับเฟรม 214 216 และ 220 ตามแนวตั้งจากบนลงล่าง โดยใช้ข้อมูลทดสอบ ComSci 3	83
4.32 ตัวอย่างการติดตามที่ยังมีปัญหา (วงกลมหมายเลข 3) ลำดับเฟรม 374 และ 380 ตามแนวตั้งจากบนลงล่าง โดยใช้ข้อมูลทดสอบ ComSci 3.....	83
4.33 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธีสำหรับ โดยใช้ข้อมูลทดสอบ ComSci 3	84
4.34 ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ ComSci 4	85
4.35 ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ ComSci 4	85
4.36 ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC ลำดับเฟรม 100 105 110 115 และ 120 ตามแนวตั้งจากบนลงล่าง.	86
4.37 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธีสำหรับ ข้อมูลทดสอบ ComSci 4	87
4.38 ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 1 .	88
4.39 ค่าระยะทางผิดพลาดในแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 1	89
4.40 ตัวอย่างการติดตามที่ยังมีปัญหามาของขั้นตอนวิธี MST_AC ลำดับเฟรม 927 929 และ 931ตามแนวตั้งจากบนลงล่าง.....	90
4.41 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูลทดสอบ Airport 1	91
4.42 ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 2 .	91
4.43 ค่าระยะทางผิดพลาดในแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 2	92
4.44 ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC_16 ลำดับเฟรม 2719 2739 และ 2759 ตามแนวตั้งจากบนลงล่าง โดยใช้ข้อมูลทดสอบ Airport 2	93
4.45 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธีสำหรับข้อมูลทดสอบ Airport 2	94
4.46 ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 3 .	94
4.47 ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 3.....	95

รายการภาพประกอบ (ต่อ)

ภาพประกอบ	หน้า
4.48 ตัวอย่างการติดตามที่ดีของชั้นตอนวิถี MST_AC_16 ลำดับเฟรม 2729 และ 2769 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล Airport 3	96
4.49 ตัวอย่างการติดตามที่ยังมีปัญหาของชั้นตอนวิถี MST_AC_16 ลำดับเฟรม 2688 และ 2697 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล Airport 3	97
4.50 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละชั้นตอนวิถี โดยใช้ข้อมูลทดสอบ Airport 3.....	97
4.51 ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละชั้นตอนวิถี โดยใช้ข้อมูลทดสอบ Airport1_new	98
4.52 ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละชั้นตอนวิถี โดยใช้ข้อมูลทดสอบ Airport1_new	99
4.53 ตัวอย่างการติดตามที่ดีของชั้นตอนวิถี MST_AC ลำดับเฟรม 964 และ 984 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล Airport1_new.....	99
4.54 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละชั้นตอนวิถีสำหรับข้อมูล Airport1_new	100
4.55 ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละชั้นตอนวิถี โดยใช้ข้อมูลทดสอบ Airport 2_new	101
4.56 ค่าเวลาเฉลี่ยต่อการวนรอบในแต่ละชั้นตอนวิถีสำหรับข้อมูล Airport 2_new ...	101
4.57 ตัวอย่างการติดตามที่ดีของชั้นตอนวิถี MST_AC ลำดับเฟรม 2699 และ 2719 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล Airport2_new.....	102
4.58 ค่าเวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละชั้นตอนวิถีสำหรับ Airport2_new ...	103
4.59 ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละชั้นตอนวิถี โดยใช้ข้อมูลทดสอบ Airport 3_new	103
4.60 ค่าระยะทางผิดพลาดในแต่ละชั้นตอนวิถี โดยใช้ข้อมูลทดสอบ Airport 3_new.	104
4.61 ตัวอย่างการติดตามที่ดีของชั้นตอนวิถี MST_AC ลำดับเฟรม 2729 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล Airport3_new	104
4.62 ค่าเวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละชั้นตอนวิถีสำหรับ Airport3_new ...	105
4.63 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละชั้นตอนวิถีในแต่ละข้อมูลทดสอบ	108
4.64 เวลาเฉลี่ยที่ใช้ในการทำงาน 1 รอบของแต่ละชั้นตอนวิถีในแต่ละข้อมูลทดสอบ	109

รายการภาพประกอบ (ต่อ)

ภาพประกอบ		หน้า
4.65	ค่าระยะทางผิดพลาดของแต่ละขั้นตอนวิธีโดยเปรียบเทียบขนาด bin 32x32x32 กับขนาด bin 16x16x16 โดยใช้ข้อมูลทดสอบ Airport 1.....	111
4.66	ค่าเวลาเฉลี่ยของแต่ละขั้นตอนวิธีโดยเปรียบเทียบขนาด bin 32x32x32 กับขนาด bin 16x16x16 โดยใช้ข้อมูลทดสอบ Airport 1	112
4.67	ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยเปรียบเทียบขนาด bin 32x32x32 กับขนาด bin 16x16x16 โดยใช้ข้อมูลทดสอบ Airport 2.....	113
4.68	ค่าเวลาเฉลี่ยของแต่ละขั้นตอนวิธีโดยเปรียบเทียบขนาด bin 32x32x32 กับขนาด bin 16x16x16 โดยใช้ข้อมูลทดสอบ Airport 2	113
4.69	ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยเปรียบเทียบขนาด bin 32x32x32 กับขนาด bin 16x16x16 โดยใช้ข้อมูลทดสอบ Airport 3	114
4.70	ค่าเวลาเฉลี่ยของแต่ละขั้นตอนวิธีโดยเปรียบเทียบขนาด bin 32x32x32 กับขนาด bin 16x16x16 โดยใช้ข้อมูลทดสอบ Airport 3	115
4.71	ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยมีการปรับเปลี่ยนความคมชัดของวิดีโอ โดยใช้ข้อมูลทดสอบ Airport1_new และ Airport1	117
4.72	ค่าเวลาเฉลี่ยของแต่ละขั้นตอนวิธีโดยมีการปรับเปลี่ยนความคมชัดของวิดีโอ โดยใช้ข้อมูลทดสอบ Airport1_new และ Airport1	117
4.73	ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยมีการปรับเปลี่ยนความคมชัดของวิดีโอ โดยใช้ข้อมูลทดสอบ Airport2_new และ Airport2	118
4.74	ค่าเวลาเฉลี่ยของแต่ละขั้นตอนวิธีโดยมีการปรับเปลี่ยนความคมชัดของวิดีโอ โดยใช้ข้อมูลทดสอบ Airport2_new และ Airport2	119
4.75	ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยมีการปรับเปลี่ยนความคมชัดของวิดีโอ โดยใช้ข้อมูลทดสอบ Airport3_new และ Airport3	119
4.76	ค่าเวลาเฉลี่ยของแต่ละขั้นตอนวิธีโดยมีการปรับเปลี่ยนความคมชัดของวิดีโอ โดยใช้ข้อมูลทดสอบ Airport3_new และ Airport3	120

บทที่ 1

บทนำ

1.1 ความเป็นมา

การติดตามการเคลื่อนที่ของวัตถุ (Object Tracking) เป็นแขนงหนึ่งของคอมพิวเตอร์วิทัศน์ (Computer Vision) ในปัจจุบันการติดตามการเคลื่อนที่ของวัตถุสามารถทำได้หลายวิธีด้วยกัน Yilmaz (2006) ซึ่งสามารถแบ่งประเภทการติดตามออกเป็น 3 ประเภท คือ ประเภทแรกเป็นการติดตามโดยใช้จุดแบ่งออกได้เป็น 2 ประเภทย่อย คือ Statistical และ Deterministic Statistical วิธีการนี้เป็นการติดตามโดยอาศัยหลักทางสถิติเข้ามาช่วยในการติดตาม โดยจะพิจารณาข้อมูลจากในอดีตมาประกอบเช่น สีหรือ เส้นขอบ เป็นต้น สำหรับการทำนายหาวัตถุเป้าหมาย วิธีการแบบ Deterministic เป็นการติดตามโดยอาศัยข้อจำกัดที่ทราบล่วงหน้า เช่น ตำแหน่ง ทิศทางหรือความเร็ว เป็นต้น ประเภทที่สองเป็นการติดตามโดยใช้ Kernel ซึ่งที่นี้หมายถึง รูปร่างของวัตถุที่ใช้ในการติดตามอาจมีลักษณะ เป็น สีเหลี่ยม วงกลม เป็นต้น เป็นการติดตามที่อาศัยแบบจำลองแบ่งออกได้เป็น 2 ประเภทย่อย คือ Multi-view Based และ Template Based สำหรับ Multi-view Based จะใช้หลักการเรียนรู้ภาพวัตถุเป้าหมายล่วงหน้าหลาย ๆ รูปแบบก่อนทำการติดตาม ส่วน Template Based จะใช้การวัดความคล้ายระหว่าง Template กับภาพวัตถุเป้าหมายเพื่อใช้ในการติดตาม เช่น วิธีการ Mean Shift เป็นต้น และประเภทสุดท้ายเป็นการติดตามโดยอาศัยโครงสร้างสามารถแบ่งได้เป็น 2 ประเภทย่อยคือ Contour Evolution และ Shape Matching สำหรับ Contour Evolution จะอาศัย Contour ของเฟรมก่อนหน้าแล้วปรับเปลี่ยนตำแหน่งให้มีความคล้ายกับ Contour ของเฟรมปัจจุบันให้มากที่สุด ส่วน Shape Matching นั้นลักษณะการทำงานจะคล้าย ๆ กับ Template Based ดังที่ได้กล่าวมาแล้วข้างต้น

การติดตามโดยใช้วิธี Mean Shift เป็นวิธีการติดตามวิธีการหนึ่งที่จัดอยู่ในประเภท Kernel Tracking ในส่วนของ Template Based ซึ่งมีหลักการทำงาน คือ มีการสร้างแบบจำลองหรือ Template ในเฟรมแรก ส่วนเฟรมถัดไปก็จะสร้างแบบจำลองใหม่แล้วปรับตำแหน่งใหม่โดยเปรียบเทียบความคล้ายระหว่างแบบจำลองใหม่กับ Template

ข้อดีของการใช้วิธีการ Mean Shift ในการติดตาม คือ เป็นวิธีที่เร็วและไม่ต้องทราบข้อมูลล่วงหน้า เช่น รูปร่างหรือขนาดก็สามารถทำงานได้ สำหรับข้อเสียของ Mean Shift คือ ขนาดของหน้าต่างมีผลต่อค่าความถูกต้องในการติดตาม กล่าวคือ ถ้าขนาดของหน้าต่างเล็กเกินไปทำให้สามารถติดตามได้เฉพาะบางส่วน แต่ถ้าขนาดของหน้าต่างใหญ่เกินไปจะทำให้การติดตามมีความถูกต้องลดลงอันเนื่องมาจากภาพพื้นหลัง

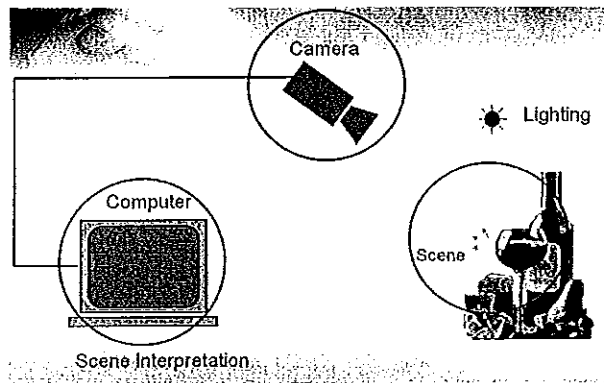
การติดตามโดยวิธีการ Mean Shift ถูกนำมาใช้ครั้งแรกโดย Comaniciu (2000) ซึ่งได้นำค่าใน Histogram มาสร้างแบบจำลอง และใช้ค่าสัมประสิทธิ์ของ Bhattacharyya Coefficient หรือ Similarity Function ในการปรับตำแหน่งสำหรับการติดตาม พร้อมทั้งได้เสนอการปรับเปลี่ยนหน้าต่างแบบ $\pm 10\%$ ตามขนาดวัตถุที่เปลี่ยนไป ต่อมา Collins (2003) ได้เสนอการปรับขนาดของหน้าต่าง โดยใช้วิธี Scale Space ในปีค.ศ. 2005 Birchfield เสนอเทคนิค Spatiogram ซึ่งเป็นการปรับ Histogram แบบเดิมโดยเพิ่มในส่วน of ค่าเฉลี่ย (Mean) และค่าความแปรปรวนร่วม (Covariance) ของตำแหน่ง ในปี ค.ศ. 2007 Yilmaz เสนอวิธีการปรับเปลี่ยนหน้าต่างตามโครงร่างของวัตถุ ซึ่งจะต้องมีการเรียนรู้โครงร่างก่อนที่จะทำการติดตาม นอกจากนั้น Zhuo (2007) ได้นำ Multiscale Images เข้ามาช่วยในการติดตามโดยใช้ในการปรับขนาดของหน้าต่าง Zhuo (2008) ได้ใช้วิธีการ Preprocess ข้อมูลก่อนสร้างแบบจำลองโดยทำการ Cluster สีด้วยวิธี Mean Shift และเลือกระบบสีแบบ HSV ในเรื่องปัญหาหาสีพื้นหลังมีสีเดียวกับวัตถุ Comaniciu (2003) ใช้วิธีขยายหน้าต่างเพื่อหาน้ำหนักของพื้นหลังแล้วไปลดค่าของแบบจำลอง

จากงานวิจัยข้างต้นจะเห็นได้ว่าการติดตามด้วยวิธีการ Mean Shift มีการพัฒนาอย่างต่อเนื่องเพื่อให้การทำงานมีประสิทธิภาพสูงสุด อย่างไรก็ตามสำหรับขั้นตอนวิธีนี้ยังมีปัญหาในเรื่องของความถูกต้องเมื่อสีของพื้นหลังมีสีเดียวกับวัตถุ จึงเป็นที่มาสำหรับวิทยานิพนธ์นี้ที่พัฒนาการติดตามด้วยวิธีการ Mean Shift เพื่อเพิ่มความสามารถในการติดตามทั้งในแง่ความถูกต้องและความเร็วในการทำงาน

1.2 การตรวจเอกสาร

1.4.2 คอมพิวเตอร์วิทัศน์ คือ การที่ทำให้คอมพิวเตอร์สามารถเข้าใจความหมายจากภาพหรือวิดีโอโดยข้อมูลที่รับเข้ามาเป็นภาพ ข้อมูลออกไปเป็นความหมาย แสดงได้ดัง

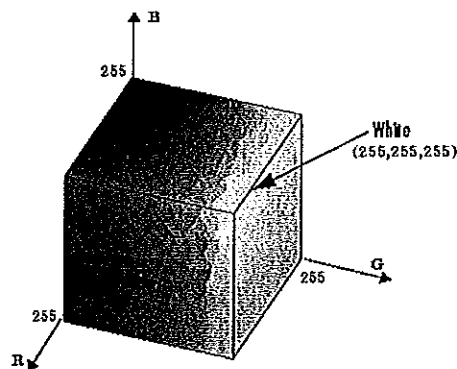
ภาพประกอบ 1.1 ซึ่งเมื่อกล้องได้รับภาพก็จะดำเนินการส่งข้อมูลที่เป็นภาพไปยังเครื่องคอมพิวเตอร์ ซึ่งข้างในมีขั้นตอนวิธีในการแปลความหมายภาพ



ภาพประกอบ 1.1 ความหมายของคอมพิวเตอร์วิทัศน์ (Narasimhan, 2006)

1.4.2 ตัวอย่างระบบสี เช่น RGB HSV และ YIQ โดยมีรายละเอียดดังนี้

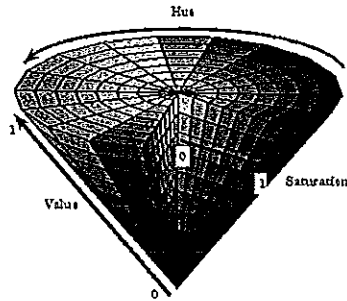
- RGB (Red, Green, Blue) เป็นระบบสีที่ประกอบด้วยสีหลัก ๆ คือ สีแดง สีเขียว และสีฟ้า ซึ่งระบบสี RGB สามารถแสดงสีได้ทั้งหมด 2^{24} สี ดังภาพประกอบ 1.2



ภาพประกอบ 1.2 ระบบสี RGB (Matlab, 2007)

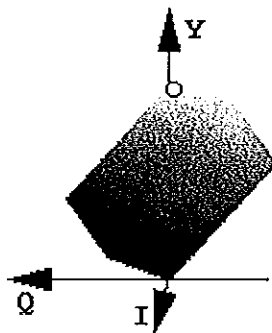
- HSV (Hue, Saturation, Value) เป็นระบบสีที่ประกอบด้วย Hue คือค่าของสี เช่น สีแดง สีเหลือง สีเขียว ซึ่งวัดเป็นมุมที่อยู่ระหว่าง 0 – 360 องศา ซึ่งสีแดง สีเหลือง และสีเขียวจะมีค่าต่างกันสีละ 60 องศา Saturation คือค่าความเข้มของเนื้อสี หรือค่าความบริสุทธิ์ของสี มีค่าตั้งแต่ 0 – 100 โดยสีจะมีความเข้มมากขึ้นเรื่อยๆ เมื่อค่า Saturation มีค่าเพิ่มขึ้น Value หรือ Brightness คือ ความสว่างของสีมีค่าตั้งแต่ 0 – 100 โดยภาพจะสว่างมากขึ้น เมื่อ Brightness มีค่าเพิ่มขึ้น สีในระบบ HSV มีความคงทนต่อการเปลี่ยนแปลงของแสง

สว่างมากกว่า RGB จึงเป็นระบบสีที่นิยมนำไปใช้ในการแบ่งกลุ่ม (Classification) ดังภาพประกอบ 1.3



ภาพประกอบ 1.3 ระบบสี HSV (Matlab, 2007)

- YIQ เป็นระบบสีที่ใช้ในการแพร่ภาพผ่านทางโทรทัศน์ประกอบด้วย 3 องค์ประกอบด้วยกัน คือ Y แทนความสว่าง I และ Q แทนสี ประโยชน์ของระบบสีนี้คือ ข้อมูลที่เป็นภาพ Gray Scale จะถูกแบ่งแยกจากข้อมูลสี ดังนั้นสัญญาณเดียวกันสามารถใช้ได้ทั้งเป็นภาพสีและภาพขาวดำแสดงได้ดังภาพประกอบ 1.4



ภาพประกอบ 1.4 ระบบสี YIQ (Wiedeman, 2008)

1.4.2 การแทนภาพวัตถุ (Object Representation) เป็นการแทนภาพของวัตถุในรูปแบบต่าง ๆ โดยสามารถจัดเป็นกลุ่มได้เป็น 5 แบบ คือ

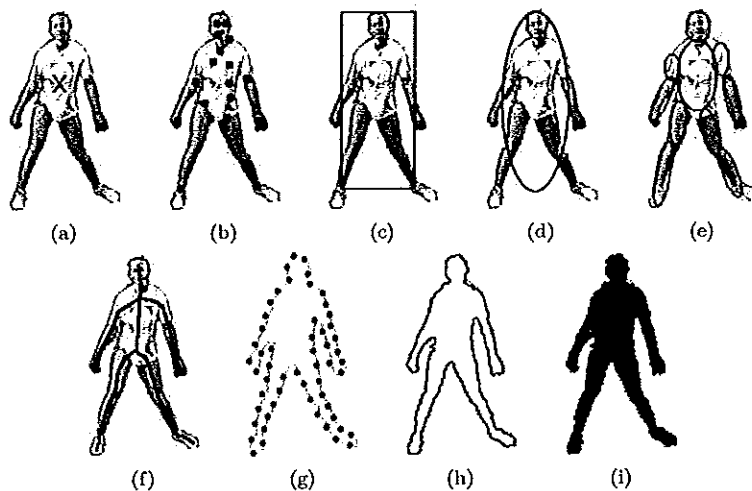
a) Point เป็นการแสดงภาพวัตถุเป้าหมายโดยการใช้จุด เช่น การใช้ Centroid หรือจุดกึ่งกลาง ในงานของ Veenman และคณะ (2001) การใช้กลุ่มของจุดในงานของ Serby และคณะ (2004) โดยสังเกตได้ว่าจุดจะเหมาะสำหรับการติดตามพื้นที่ขนาดเล็ก แสดงได้ดังภาพประกอบ 1.5 (a) (b) และ (g)

b) Primitive Geometric Shapes เป็นการแสดงวัตถุโดยแสดงเป็นรูปทรงทางเรขาคณิต เช่น วงรี สี่เหลี่ยม เช่น งานของ Comaniciu (2003) การแสดงภาพวัตถุแบบนี้ซึ่งเหมาะสำหรับภาพทั่วไป แสดงได้ดังภาพประกอบ 1.5 (c) และ (d)

c) Object Silhouette and Contour เป็นการแสดงวัตถุโดยอาศัยเส้นโครงร่างของวัตถุแสดงได้ดังภาพประกอบ 1.5 (g) (h) และ (i) เหมาะสำหรับวัตถุที่มีรูปร่างที่ไม่เป็นทรงตัน เช่น ในงานของ Yilmaz และคณะ (2004)

d) Articulated Shape Model เป็นการแสดงวัตถุโดยแสดงองค์ประกอบแต่ละส่วนของภาพเป้าหมาย แสดงได้ดังภาพประกอบ 1.5 (e)

e) Skeletal Models เป็นการแสดงวัตถุในรูปแบบแกนกลางของภาพวัตถุ แสดงได้ดังภาพประกอบ 1.5 (f)



ภาพประกอบ 1.5 การแสดงวัตถุในรูปแบบต่างๆ (Yilmaz, 2008)

(a) แสดงโดยใช้จุดศูนย์กลาง (b) แสดงโดยใช้กลุ่มของจุด (c) แสดงโดยใช้สี่เหลี่ยมผืนผ้า (d) แสดงโดยใช้วงรี (e) แสดงโดยใช้อ็องค์ประกอบหลาย ๆ ส่วน (f) แสดงโดยใช้เส้นแกนกลาง (g) แสดงโดยใช้กลุ่มจุดแสดงรูปร่าง (h) แสดงโดยใช้เส้นรูปร่าง (i) แสดงเป็นภาพวัตถุที่เป็นสีดำ

1.4.2 การเลือกคุณสมบัติเพื่อใช้ในการติดตาม

(Feature Selection for Tracking)

คุณสมบัติที่ใช้ในการติดตามมีหลายคุณสมบัติด้วยกันไม่ว่าจะเป็นสีซึ่งใช้ใน Histogram หรือ Optical Flow สำหรับภาพที่มีการเคลื่อนที่และในบางครั้งการติดตามก็สามารถ

ใช้หลายคุณสมบัติรวมกันมาใช้ในการติดตาม ขึ้นอยู่กับวัตถุประสงค์ของงานและสิ่งแวดล้อม คุณสมบัติที่ใช้ในการติดตามมีดังนี้

1) สีเป็นคุณสมบัติหนึ่งที่ยอมรับใช้ สิ่งที่มีผลต่อความคมชัดของสี คือ ความสว่างและการสะท้อนของพื้นผิว (ขึ้นอยู่กับคุณสมบัติของวัตถุ)

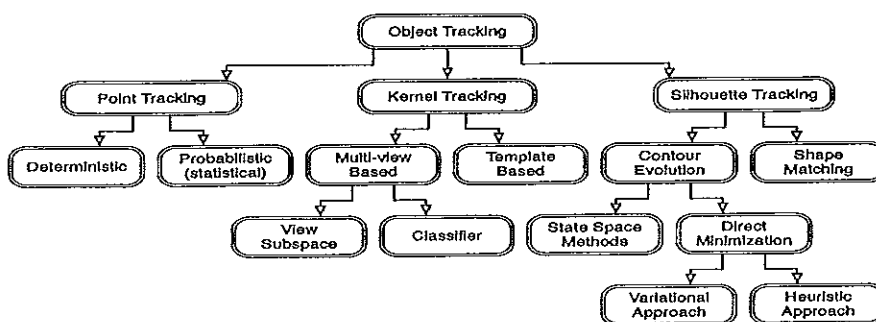
2) เส้นขอบ (Edge) เป็นคุณสมบัติที่อาศัยการเปลี่ยนแปลงความเข้มของสีที่อยู่บนเส้นขอบของโครงสร้างภาพวัตถุ ปกติจะใช้กับภาพ Gray Scale ซึ่งภาพสีที่เป็น Gray Scale นั้น จะเป็นภาพที่มีการเปลี่ยนแปลงของสีน้อยเมื่อแสงสว่างเปลี่ยนไปจึงทำให้ขั้นตอนวิธีที่ใช้ในการติดตามหาโครงสร้างของวัตถุ (Contour) มีความถูกต้องมาก

3) Optical Flow เป็นคุณสมบัติที่อาศัยเวกเตอร์ของแต่ละจุดในภาพโดย Optical Flow นั้น อาศัยสมมุติฐานที่ว่าความเข้มของสีจะไม่เปลี่ยนแปลงเมื่อเวลาผ่านไป ในปัจจุบัน Optical Flow มีหลายขั้นตอนวิธีด้วยกัน เช่น วิธีการของ Horn และคณะ (1981) และวิธีการของ Lucas และคณะ (1981)

4) พื้นผิว (Texture) เป็นคุณสมบัติที่ใช้การวัดความแตกต่างของสีบนพื้นผิวซึ่งวัดได้จากความสม่ำเสมอ และความเข้ากันได้ของพื้นผิวโดยเปรียบเทียบจากสี ในการเปรียบเทียบพื้นผิวจะต้องมีขั้นตอนวิธีที่เพิ่มเข้ามาเพื่อสร้างตัวอธิบาย (Descriptors) ซึ่งมีหลากหลายวิธีด้วยกัน เช่น วิธี Gray-Level Cooccurrence Matrices (GLCM's) (Haralick, 1973)

1.4.2 ประเภทของการติดตาม

เป็นการแบ่งกลุ่มประเภทของการติดตามแสดงได้ดังภาพประกอบ 1.6 โดยแบ่งตามคุณสมบัติที่ใช้ในการติดตาม ซึ่งมี 3 แบบด้วยกัน คือ แบบจุด แบบ Kernel และแบบโครงสร้าง



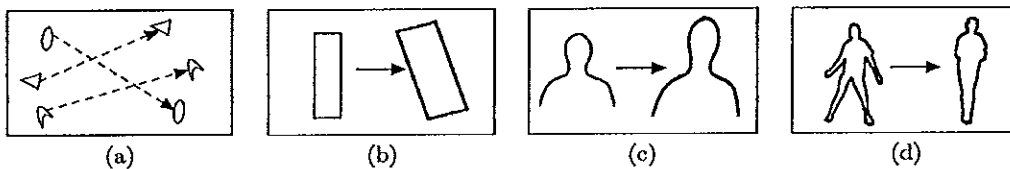
ภาพประกอบ 1.6 ประเภทของการติดตาม (Yilmaz, 2006)

ภาพประกอบ 1.7 แสดงตัวอย่างการทำงานในการติดตามวัตถุ ดังนี้

- Point Tracking เป็นการติดตามโดยใช้จุดและความสัมพันธ์ของแต่ละจุดหาได้จากสถานะของวัตถุก่อนหน้านี้ ซึ่งอาจจะเป็น ตำแหน่ง ความเร็ว เป็นต้น แสดงได้ดังภาพประกอบ 1.7 (a)

- Kernel Tracking เป็นการติดตามโดยใช้ Kernel ซึ่ง Kernel คือ รูปร่างของวัตถุ เช่น สีเหลี่ยมผืนผ้า วงรี เป็นต้น ซึ่งสัมพันธ์กับ Histogram วัตถุจะถูกติดตามโดยพิจารณาการเคลื่อนที่ของ Kernel แสดงได้ดังภาพประกอบ 1.7 (b)

- Silhouette Tracking เป็นการติดตามโดยอาศัยการประมาณพื้นที่ในแต่ละเฟรมโดยวัตถุหนึ่งวัตถุจะประกอบด้วยพื้นที่หลายส่วนประกอบกัน เช่น ภาพประกอบ 1.7 (c) และ (d)



ภาพประกอบ 1.7 ตัวอย่างการติดตามแต่ละประเภท (Yilmaz, 2006)

นอกจากนี้การติดตามด้วยจุด การติดตามด้วย Kernel และการติดตามด้วยรูปร่างมีรายละเอียดดังต่อไปนี้

1) การติดตามด้วยจุด (Point Tracking) เป็นการติดตามหาวัตถุในแต่ละเฟรมโดยใช้จุดซึ่งอาจจะหมายถึงจุดที่ใช้แสดงการเคลื่อนที่ของภาพก็ได้ในการติดตามด้วยจุดสามารถแบ่งได้ดังนี้

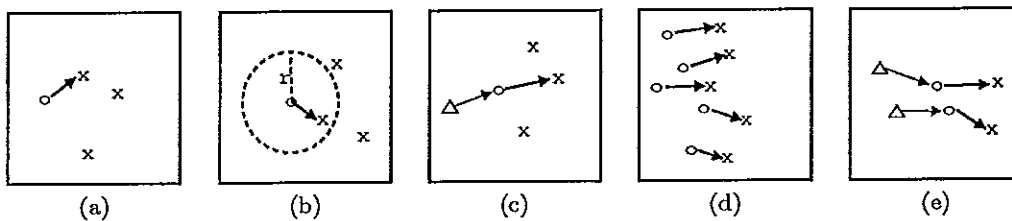
1.1) Deterministic Methods เป็นวิธีการติดตาม ที่อาศัยหลักการตั้งสมมุติฐานหรือ ข้อจำกัดที่ตั้งไว้ล่วงหน้า ได้แก่

- ความใกล้ชิดกัน โดยพิจารณาความใกล้ชิดกันของจุดที่อยู่ใกล้ ๆ ในเฟรมปัจจุบันกับเฟรมก่อนหน้านี้ ภาพประกอบ 1.8 (a)

- ความเร็วสูงสุด โดยพิจารณาความเร็วสูงสุดไม่เกินความเร็วที่ตั้งไว้ ดังภาพประกอบ 1.8 (b)

- ความเร็วเปลี่ยนแปลงเล็กน้อยเป็นความเร็วที่ตั้งไว้อยู่ในสมมุติฐานว่าความเร็วที่เปลี่ยนแปลงระหว่างเฟรม 2 เฟรม จะไม่แตกต่างกันและทิศทางเปลี่ยนแปลงไปไม่มาก ดังภาพประกอบ 1.8 (c)

- ความเร็วที่เหมือนกันของจุดที่อยู่ใกล้เคียงกัน โดยดูจากความเร็วของจุดที่อยู่ใกล้เคียงกัน ส่วนใหญ่จะไปในทิศทางเดียวกัน แสดงได้ดังภาพประกอบ 1.8 (d)



ภาพประกอบ 1.8 แสดงข้อจำกัดในการเคลื่อนที่ (Yilmaz, 2006)

(a) พิจารณาตามความใกล้ (b) พิจารณาตามค่าความเร็วสูงสุด (c) พิจารณาตามความเร็วที่เปลี่ยนแปลงเล็กน้อย (d) พิจารณาความเร็วของเพื่อนบ้านจะไปในทิศทางที่คล้าย ๆ กัน (e) พิจารณาจุดที่เคลื่อนที่ของวัตถุ 3D จะเห็นได้ว่าทิศทางการเคลื่อนที่ของจุด 2 จุด จะไม่เปลี่ยนแปลง

1.2) Statistical Methods เป็นวิธีการติดตามที่ใช้ข้อมูลทางสถิติมาช่วยโดยอาศัยข้อมูลที่รู้ก่อนหน้า เช่น ตำแหน่ง ความเร็ว แล้วนำมาทำนายเหตุการณ์ที่จะเกิดขึ้นข้างหน้า และจะมีตัววัดประสิทธิภาพโดยปกติมักจะเป็นตำแหน่งในภาพซึ่งหาได้จากวิธีการที่ใช้ในการตรวจหาตำแหน่ง การประมาณสถานะ ซึ่งสามารถแบ่งออกได้เป็น 2 ประเภทย่อย คือ Kalman Filtering ใช้ในการประมาณสถานะแบบเชิงเส้น โดยสถานะจะมีการกระจายแบบ Gaussian สำหรับอีกวิธีคือ Particle Filter เป็นวิธีแก้ปัญหาจากข้อจำกัดของ Kalman Filter เนื่องจากสถานะของตัวแปรจะเป็นการกระจายแบบ Gaussian ซึ่งถ้าสถานะของตัวแปรไม่กระจายตามแบบ Gaussian จะทำให้ความถูกต้องในการทำนายลดลง วิธีการ Particle Filter จึงเข้ามาช่วยในการแก้ปัญหา

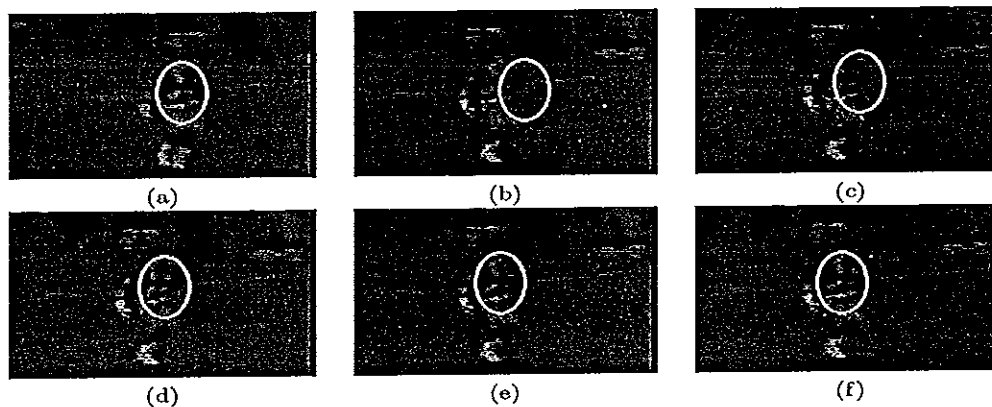
2) Kernel Tracking เป็นการติดตาม ซึ่ง Kernel เป็นรูปร่างของวัตถุ เช่น Kernel ที่เป็นสี่เหลี่ยมผืนผ้าหรือวงรี Kernel Tracking แบ่งออกได้เป็น 2 ประเภท คือ

2.1) การติดตามโดยใช้ Template เป็นการติดตามโดยใช้ค่าน้ำหนักที่ปรากฏบนแบบจำลอง ซึ่งนิยมใช้กันอย่างกว้างขวางเนื่องจากเป็นวิธีที่ง่ายและไม่สิ้นเปลืองหน่วยความจำ วิธีการนี้แบ่งออกได้เป็น 2 ประเภทย่อย คือ

- **Template Matching** เป็นการติดตามโดยใช้ Template ในการดูการเข้ากันได้ของภาพวัตถุกับ Template โดยใช้วิธีที่เรียกว่า Brute Force ในการค้นหาวัตถุ ซึ่งใช้เวลาในการประมวลผลมาก ดังนั้นเพื่อลดเวลาในการประมวลผลจึงใช้วิธีค้นหาวัตถุใกล้กับตำแหน่งเป้าหมายในเฟรมก่อนหน้านั้น

- **Mean Shift Tracking** เป็นการติดตาม โดยนำสีมาสร้างเป็นแบบจำลองบน Histogram ตัวอย่างการติดตามโดยใช้ Mean Shift แสดงได้ดังภาพประกอบ

1.9



ภาพประกอบ 1.9 ขั้นตอนการทำงานของ Mean Shift (Yilmaz, 2006)

(a) แสดงการประมาณตำแหน่งที่เวลา $t-1$, (b) แสดงการประมาณตำแหน่งที่เวลา t (c), (d), (e), (f) แสดงการปรับปรุงตำแหน่ง

2.2) **Tracking Using Multiview Appearance Models** เป็นวิธีที่ใช้ในการแก้ปัญหาของวิธีที่ 1 ที่ได้กล่าวมาแล้วซึ่งในระหว่างที่ทำการติดตามนั้นวัตถุที่ใช้เป็นแบบจำลองมีการเปลี่ยนรูปร่างอย่างรวดเร็ว เพื่อแก้ปัญหาที่เกิดขึ้นจึงใช้วิธีการเรียนรู้ลักษณะของวัตถุหลาย ๆ รูปแบบ ก่อนที่จะทำการติดตาม

3) การติดตามโดยใช้รูปร่าง (**Silhouette Tracking**) เป็นการติดตามโดยอาศัยรูปร่างสามารถแบ่งแยกได้เป็น 2 ประเภทย่อย คือ

3.1) Shape Matching เป็นการติดตามที่คล้ายกับการติดตามโดยใช้ Template โดยอาศัย Shape เป็นแบบจำลองหลัก

3.2) Contour Evolution เป็นวิธีที่อาศัย Contour ของเฟรมก่อนหน้าแล้วปรับเปลี่ยนตำแหน่งให้มีความคล้ายกับ Contour ของเฟรมปัจจุบันให้มากที่สุด

1.4.2 การติดตามโดยใช้วิธีการ Mean Shift

- ความเป็นมาของ Mean Shift

Mean Shift เป็นวิธีการหาความหนาแน่นของกลุ่มข้อมูล ซึ่งเป็นเทคนิคของการจดจำรูปแบบ (Pattern Recognition) เสนอโดย K. Fukunaga and L.D. Hosteler ในปี 1975 ได้นำ Mean Shift มาใช้กับภาพ วิธีการนี้เรียกว่า Kernel Density Estimation (Parzen Window) ซึ่งจะหาระยะห่างของทุกจุดกับจุดที่กำหนดเริ่มต้นในหน้าต่าง จากนั้นก็นำมาหาค่าเฉลี่ยแล้วเลื่อนไปยังตำแหน่งที่มีความหนาแน่นมากที่สุด

ในทางสถิติจะแบ่งวิธีการทางสถิติเป็น 2 แบบ คือ ทราบค่าทางสถิติล่วงหน้า (Parametric) และ ไม่ทราบค่าทางสถิติล่วงหน้า (Nonparametric) Mean Shift เป็นวิธีการทางสถิติแบบไม่ทราบค่าทางสถิติล่วงหน้า ซึ่งผู้ทดลองไม่ทราบค่าทางสถิติที่ได้จากกลุ่มตัวอย่างข้อมูล

ข้อดีของการไม่ทราบค่าทางสถิติล่วงหน้า

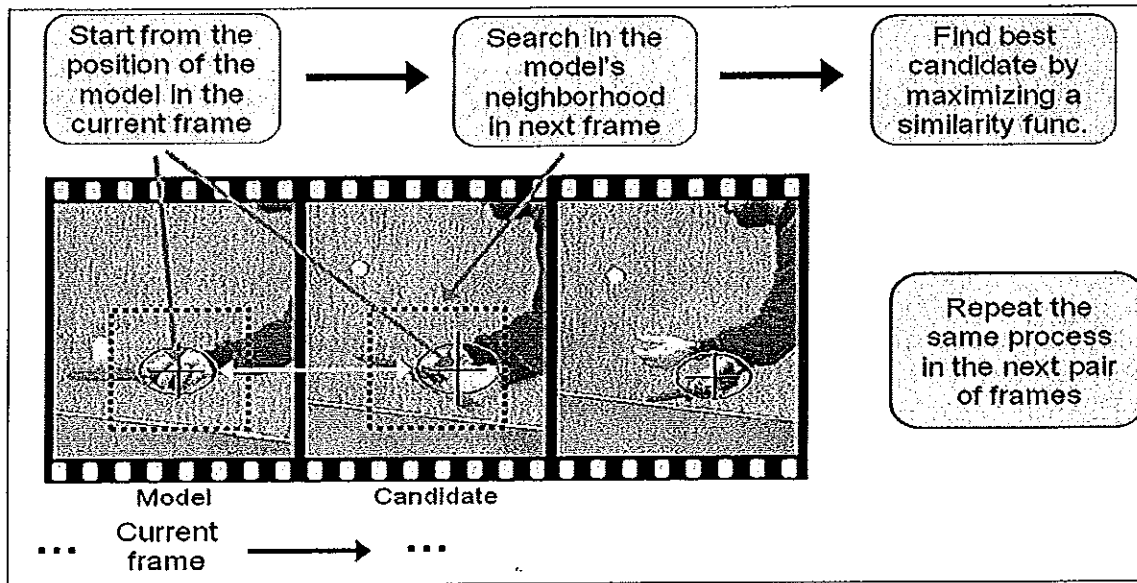
สูตรในการคำนวณง่ายไม่ซับซ้อนและใช้ได้ดีกับข้อมูลขนาดเล็กเมื่อเปรียบเทียบกับ ทราบค่าทางสถิติล่วงหน้า

ข้อเสียของการไม่ทราบค่าทางสถิติล่วงหน้า

ถ้ากลุ่มข้อมูลมีขนาดใหญ่สถิติแบบไม่ทราบค่าทางสถิติล่วงหน้าจะให้ค่าความถูกต้องน้อยกว่า

- ขั้นตอนการทำงานของ Mean Shift ในการติดตาม

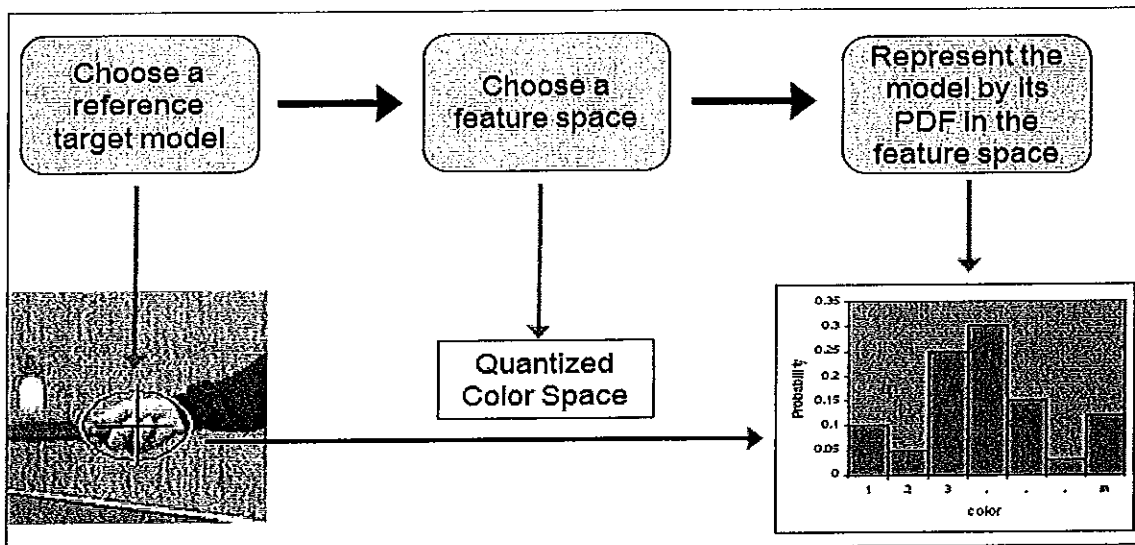
ขั้นตอนหลักในการทำงานของ Mean Shift แสดงได้ดังภาพประกอบ 1.10 โดยขั้นตอนในการทำงานหลักมีดังนี้ สร้างแบบจำลองที่ได้จากเฟรมแรก ค้นหาแบบจำลองที่มีความคล้ายมากที่สุดในเฟรมถัดไป และปรับตำแหน่งตามความคล้าย ส่วนขั้นตอนโดยละเอียดมีทั้งหมด 6 ขั้นตอนดังนี้



ภาพประกอบ 1.10 การติดตามวัตถุด้วยวิธีการ Mean Shift (Simakov, 2008)

ขั้นตอน 1) เลือกตำแหน่งจุดเริ่มต้น (Object Detection) ของภาพวัตถุที่สนใจในเฟรมแรก รวมทั้งความกว้างยาวของหน้าต่างและจุดศูนย์กลางของหน้าต่าง

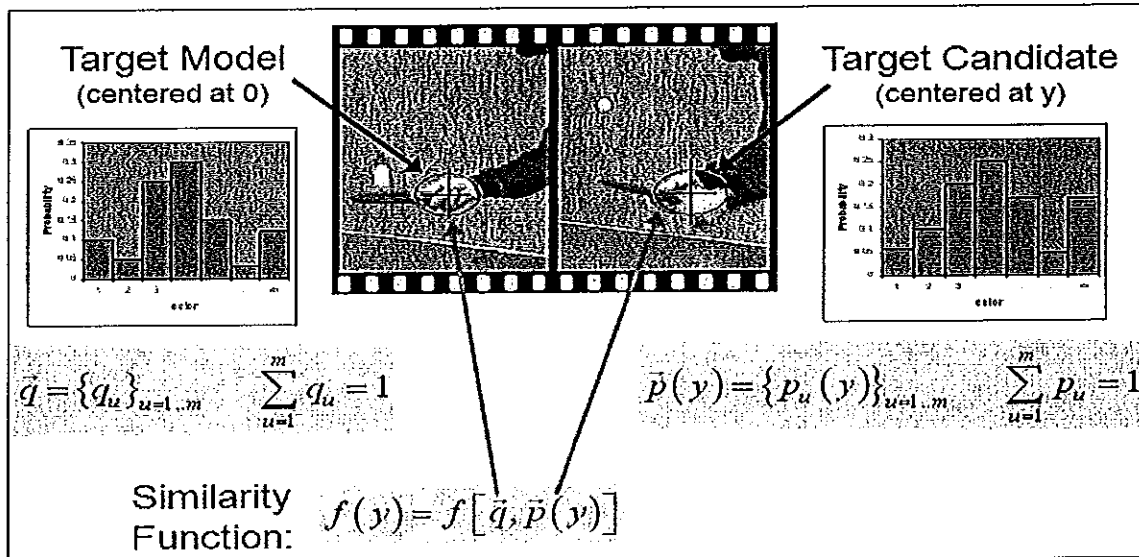
ขั้นตอน 2) สร้างแบบจำลองจากเฟรมแรก โดยเลือกคุณสมบัติ เช่น สีของพื้นที่ในหน้าต่างในขั้นตอนเลือกตำแหน่งจุดเริ่มต้นมาทำการ Quantization แล้วนำมาสร้าง Histogram ซึ่งใน Histogram ในแต่ละ bin เป็นความหนาแน่นของแต่ละสี (PDF) แสดงได้ดังภาพประกอบ 1.11



ภาพประกอบ 1.11 การสร้างแบบจำลอง (Simakov, 2008)

ขั้นตอน 3) ค้นหาแบบจำลองคู่แข่งจากเฟรมถัดไป

ขั้นตอน 4) วัดความคล้ายกันระหว่างแบบจำลองเป้าหมาย (Target Model) กับคู่แข่ง (Target Candidate) แสดงได้ดังภาพประกอบ 1.12



ภาพประกอบ 1.12 การวัดความคล้ายกันระหว่างแบบจำลองหลักกับแบบจำลองคู่แข่ง (Simakov, 2008)

แบบจำลองหลัก (Target Model) จากภาพประกอบ 1.12 ซ้ายมือที่เป็น Histogram ที่แสดงแบบจำลองหลักซึ่งสร้างขึ้นครั้งเดียวหลังจากขั้นตอนการหาวัตถุ (Object Detection) และใช้ตลอดการติดตามวัตถุ แบบจำลองหลักนี้ประกอบไปด้วย q ของแต่ละสีหรือ Bin (u) ตั้งแต่ 1 ถึง m $\vec{q} = \{q_u\}_{u=1..m}$ โดยค่าในแต่ละ bin หาได้จากค่าระยะทางระหว่างตำแหน่งและจุดศูนย์กลางของหน้าต่าง (y) แล้วส่งเข้าฟังก์ชัน kernel จากนั้น Normalize ค่าของ Bin ให้มีค่าอยู่ระหว่าง 0-1 ดังสมการที่ (1.1)

$$q_u = C \sum_{i=1}^n k\left(\left\|\frac{y-x_i}{h}\right\|\right) \delta_u(b(x_i)) \quad (1.1)$$

โดย C เป็นค่า Normalize ดังสมการที่ (1.2)

$$C = \frac{1}{\sum_{i=1}^n k\left(\left\|\frac{y-x_i}{h}\right\|\right)} \quad (1.2)$$

- n แทน จำนวนจุดที่เลือกในหน้าต่าง
 y แทน จุดศูนย์กลางของหน้าต่างหลังจากการตรวจหาวัตถุ
 h แทน ขอบเขตของแกน x และแกน y ในกรณีขนาดหน้าต่างเป็น

รูปสี่เหลี่ยม ขอบเขตแกน x และ y คือ (ความกว้าง/2 , ความยาว/2)

- x_i แทน ตำแหน่งของ Pixel
 $k()$ แทน ฟังก์ชัน Kernel
 $b(x_i)$ แทน หมายเลขสีที่ตำแหน่ง x_i
 $\delta_u(x)$ แทน Kronecker Delta Function ดังสมการที่ (1.3)

$$\delta_u(x) = \begin{cases} 1 & \text{if } u - x = 0 \\ 0 & \text{otherwise} \end{cases} \quad (1.3)$$

แบบจำลองคู่แข่ง (Target Candidate Model) จากภาพประกอบ 1.12 ขวามือที่เป็น Histogram ที่แสดงแบบจำลองคู่แข่ง (Target Candidate Model) ซึ่งจะสร้างทุกครั้งเมื่อมีเฟรมใหม่ และแต่ละเฟรมอาจสร้างได้มากกว่า 1 ครั้ง ขึ้นอยู่กับค่าความคล้ายระหว่างแบบจำลองหลักและแบบจำลองคู่แข่ง แบบจำลองคู่แข่งนี้ประกอบไปด้วย \hat{p} ของแต่ละสีหรือ Bin (u) ตั้งแต่ 1 ถึง m $\hat{p} = \{\hat{p}_u(y)\}_{u=1\dots m}$ วิธีในการสร้างแบบจำลองคู่แข่งเช่นเดียวกับการสร้างแบบจำลองหลัก โดย y แทนจุดศูนย์กลางของหน้าต่าง วิธีในการสร้างแบบจำลองดังสมการที่ (1.4)

$$\hat{p}_u(y) = C \sum_{i=1}^n k\left(\left|\frac{y-x_i}{h}\right|\right) \delta_u(b(x_i)) \quad (1.4)$$

ในการสร้างแบบจำลองต้องมีการทำ Quantize เพื่อลดจำนวนสี ซึ่งปกติจำนวนสีทั้งหมด $256 \times 256 \times 256$ Bin ถ้าต้องการลดจำนวน Bin ให้เหลือแค่ $32 \times 32 \times 32$ Bin ที่เป็นไปได้โดยไม่ซ้ำกันใช้สมการที่ (1.5) เป็นการทำให้ 3 มิติของสีของ (R,G,B) อยู่ในรูปแบบ 1 มิติ ซึ่ง $b(x_i)$ หมายถึงดัชนีที่อ้างอิงสีซึ่งได้จากสมการที่ (1.5)

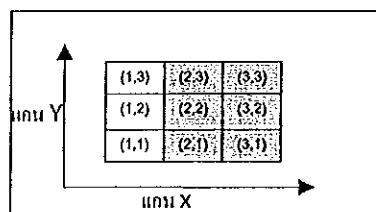
$$b(x_i) = R(x_i)/8 + 32G(x_i)/8 + 32 \times 32B(x_i)/8 \quad (1.5)$$

โดย $R(x_i)$ แทน สีแดงที่ตำแหน่ง x_i ซึ่งมีค่าตั้งแต่ 0 - 255

$G(x_i)$ แทน สีเขียวที่ตำแหน่ง x_i ซึ่งมีค่าตั้งแต่ 0 - 255

$B(x_i)$ แทน สีน้ำเงินที่ตำแหน่ง x_i ซึ่งมีค่าตั้งแต่ 0 - 255

ตัวอย่างในการสร้างแบบจำลองทั้งแบบจำลองหลักและแบบจำลองคู่แข่งมีวิธีการสร้างที่เหมือนกันดังภาพประกอบ 1.13 แทนตัวอย่างของเฟรม 1 เฟรม โดยในเฟรมนั้นมีวัตถุที่สนใจซึ่งแสดงเป็นตารางและมีหมายเลขพิกัดตำแหน่ง ในช่อง (1,3) (1,2) และ (1,1) แทนสีขาวย ในช่อง (2,3) (2,2) (2,1) (3,3) (3,2) และ (3,1) แทนสีเทา พิกัด (2,2) แทนจุดศูนย์กลางของหน้าต่าง (y) หน้าต่างนี้มีขนาด 3x3 จะเห็นได้ว่าสีที่ใช้ในการสร้างแบบจำลองมี 2 สี หรือมีจำนวน 2 bin คือ สีขาวและสีเทา



ภาพประกอบ 1.13 ตัวอย่างการสร้างแบบจำลอง

ให้ ตำแหน่งของสีขาวมีดังนี้ $x_1 = (1,3)$, $x_2 = (1,2)$, $x_3 = (1,1)$

ตำแหน่งของสีเทามีดังนี้ $x_4 = (2,1)$, $x_5 = (2,2)$, $x_6 = (2,3)$

$x_7 = (3,1)$, $x_8 = (3,2)$, $x_9 = (3,3)$

ขอบเขตของแกน x และแกน Y $= \left(\frac{3}{2}, \frac{3}{2} \right)$

$$\text{ค่า bin สีขาวรวมทั้งหมด} = k \left(\left\| \frac{y-x_1}{h} \right\| \right) + k \left(\left\| \frac{y-x_2}{h} \right\| \right) + k \left(\left\| \frac{y-x_3}{h} \right\| \right)$$

$$\begin{aligned} \text{ค่า bin สีเทารวมทั้งหมด} &= k \left(\left\| \frac{y-x_4}{h} \right\| \right) + k \left(\left\| \frac{y-x_5}{h} \right\| \right) + k \left(\left\| \frac{y-x_6}{h} \right\| \right) \\ &+ k \left(\left\| \frac{y-x_7}{h} \right\| \right) + k \left(\left\| \frac{y-x_8}{h} \right\| \right) + k \left(\left\| \frac{y-x_9}{h} \right\| \right) \end{aligned}$$

$$\begin{aligned} \text{ค่า bin สีขาวรวมทั้งหมด} &= k \left(\sqrt{\left(\frac{1-2}{\frac{3}{2}}\right)^2 + \left(\frac{3-2}{\frac{3}{2}}\right)^2} \right) + k \left(\sqrt{\left(\frac{1-2}{\frac{3}{2}}\right)^2 + \left(\frac{2-2}{\frac{3}{2}}\right)^2} \right) + k \left(\sqrt{\left(\frac{1-2}{\frac{3}{2}}\right)^2 + \left(\frac{1-2}{\frac{3}{2}}\right)^2} \right) \\ &= \frac{20}{9} \end{aligned}$$

$$\begin{aligned} \text{ค่า bin สีเทารวมทั้งหมด} &= k \left(\sqrt{\left(\frac{2-2}{\frac{3}{2}}\right)^2 + \left(\frac{3-2}{\frac{3}{2}}\right)^2} \right) + k \left(\sqrt{\left(\frac{2-2}{\frac{3}{2}}\right)^2 + \left(\frac{2-2}{\frac{3}{2}}\right)^2} \right) + k \left(\sqrt{\left(\frac{2-2}{\frac{3}{2}}\right)^2 + \left(\frac{1-2}{\frac{3}{2}}\right)^2} \right) \\ &\quad + k \left(\sqrt{\left(\frac{3-2}{\frac{3}{2}}\right)^2 + \left(\frac{3-2}{\frac{3}{2}}\right)^2} \right) + k \left(\sqrt{\left(\frac{3-2}{\frac{3}{2}}\right)^2 + \left(\frac{2-2}{\frac{3}{2}}\right)^2} \right) + k \left(\sqrt{\left(\frac{3-2}{\frac{3}{2}}\right)^2 + \left(\frac{1-2}{\frac{3}{2}}\right)^2} \right) \\ &= \frac{28}{9} \end{aligned}$$

k ในที่นี้คือ Kernel Profile โดยปกติใช้ Epanechnikov Kernel ดังสมการที่ (1.8) ซึ่งจะกล่าวในขั้นตอนต่อไป จากนั้นก็ทำการ Normalize

ค่า bin สีขาวที่มีการ Normalize = ค่า bin สีขาวรวมทั้งหมด / (ค่า bin สีขาวรวมทั้งหมด + ค่า bin สีเทารวมทั้งหมด)

$$\begin{aligned} &= \frac{\frac{20}{9}}{\left(\frac{20}{9} + \frac{28}{9}\right)} \\ &= 0.417 \end{aligned}$$

ค่า bin สีเทาที่มีการ Normalize = ค่า bin สีเทารวมทั้งหมด / (ค่า bin สีขาวรวมทั้งหมด + ค่า bin สีเทารวมทั้งหมด)

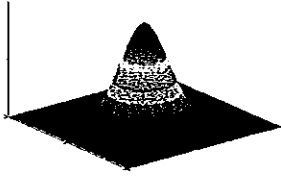

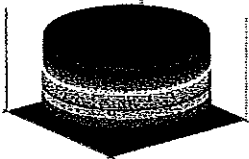
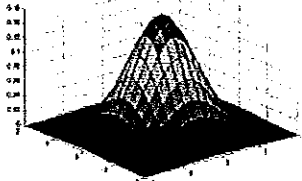
$$\begin{aligned} &= \frac{\frac{28}{9}}{\left(\frac{20}{9} + \frac{28}{9}\right)} \\ &= 0.583 \end{aligned}$$

Kernel ใช้สำหรับปรับค่าที่รับมา (Input) ให้อยู่ตามรูปแบบสมการที่เหมาะสม ซึ่งได้เป็นผลลัพธ์ในช่วงที่เหมาะสม ตัวอย่าง Kernel แสดงได้ดังตารางที่ 1.1

การเปรียบเทียบความคล้าย เป็นการวัดความคล้ายระหว่างแบบจำลองหลัก (q_u) และแบบจำลองคู่แข่ง ($p(y)$) ซึ่งมีวิธีการหลายวิธีด้วยกันแต่ที่นิยมใช้กันคือวิธี The Bhattacharyya Coefficient โดย Bhattacharyya (1983) ซึ่งเป็นวิธีวัดความคล้ายในทางสถิติ ดังสมการที่ (1.6)

$$\hat{p}(y) = f[\hat{p}(y), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}(y)_u q_u} \quad (1.6)$$

ตารางที่ 1.1 ตัวอย่าง Kernel (Simakov, 2008)

ชื่อ Kernel	ภาพจากสมการ	สมการ
Normal kernel		$K_N(x) = C \times \exp\left(-\frac{1}{2}\ x\ ^2\right) \quad (1.7)$
Epanechnikov Kernel		$K_E(x) = \begin{cases} c(1-\ x\ ^2) & \text{if } \ x\ \leq 1 \\ 0 & \text{Otherwise} \end{cases} \quad (1.8)$
Uniform Kernel		$K_U(x) = \begin{cases} c\ x\ & \text{if } \ x\ \leq 1 \\ 0 & \text{Otherwise} \end{cases} \quad (1.9)$
Gaussian Kernel		$K_G(x) = \frac{1}{\sqrt{2\pi\sigma}} \times e^{-\frac{x^2}{2\sigma^2}} \quad (1.10)$

จากสมการที่ (1.6) ปรับตำแหน่งใหม่โดยอาศัยความคล้าย จากการประมาณเชิงเส้นจะได้สมการที่ (1.11) y_0 แทนจุดศูนย์กลางเริ่มต้นของแต่ละเฟรม

$$\hat{p}(y) \approx \frac{1}{2} \sum_{u=1}^m \sqrt{p_u(y_0) q_u} + \frac{1}{2} \sum_{u=1}^m p_u(y) \sqrt{\frac{q_u}{p_u(y_0)}} \quad (1.11)$$

จากสมการที่ (1.11) เนื่องจาก $\frac{1}{2} \sum_{u=1}^m \sqrt{p_u(y_0)q_u}$ ไม่มีค่า y จึงตัดทิ้งได้ตั้งสมการที่ (1.12) และ (1.13) ส่วน $k(x)$ เป็นฟังก์ชัน Kernel ดังตารางที่ 1.1

$$\hat{p}(y) \approx \frac{C}{2} \sum_{u=1}^m \sum_{b(x_i)=u}^n k\left(\left\|\frac{y-x_i}{h}\right\|\right) \sqrt{\frac{q_u}{p_u(y_u)}} \quad (1.12)$$

$$\hat{p}(y) \approx \frac{C}{2} \sum_{i=1}^n w_i k\left(\left\|\frac{y-x_i}{h}\right\|\right) \quad (1.13)$$

โดย

$$w_i = \sum_{u=1}^m \sqrt{\frac{q_u}{p_u(y_u)}} \delta_u(b(x_i)) \quad (1.14)$$

ขั้นตอน 5) นำสมการที่ (1.13) มาหาจุดศูนย์กลางใหม่จะได้ตั้งสมการที่ (1.15) โดย k' คือ Kernel ที่ผ่าน Differentiation

$$y_1 = \frac{\sum_{i=1}^n x_i w_i k'\left(\left\|\frac{y-x_i}{h}\right\|\right)}{\sum_{i=1}^n w_i k'\left(\left\|\frac{y-x_i}{h}\right\|\right)} \quad (1.15)$$

ในกรณีที่ $k'(x)$ เป็น Uniform Kernel ดังสมการที่ (1.9) สามารถลดรูปสมการที่ (1.15) ได้เป็นสมการที่ (1.16)

$$y_1 = \frac{\sum_{i=1}^n x_i w_i}{\sum_{i=1}^n w_i} \quad (1.16)$$

นำจุดศูนย์กลางใหม่ (y_1) ที่ได้ไปหาความคล้อยใหม่ตั้งสมการที่ (1.6) จากนั้นปรับขนาดของหน้าต่างตามขั้นตอนวิธีข้างล่าง ε แทน ค่าผลต่างที่กำหนดไว้ล่วงหน้า

```

While  $\rho[\hat{p}(\hat{y}_1), \hat{q}] < \rho[\hat{p}(\hat{y}_0), \hat{q}]$ 
     $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_1 + \hat{y}_0)$ 
    Evaluate  $\rho[\hat{p}(\hat{y}_1), \hat{q}]$ 
End While
If  $\|\hat{y}_1 - \hat{y}_0\| < \varepsilon$  Stop
Else  $\hat{y}_0 \leftarrow \hat{y}_1$  Goto step 5

```

ขั้นตอน 6) ปรับขนาดหน้าต่าง

ในการปรับขนาดหน้าต่างในระหว่างติดตามนั้นวัตถุนั้นมีหลายวิธีด้วยกัน เช่น วิธีการของ Comaniciu (2003) ได้ใช้วิธีการปรับขนาดหน้าต่างโดยใช้สมการ (1.17) โดยใช้ค่า $\Delta = 0.1$

$$h = \begin{cases} h_{prev} + \Delta h_{prev} \\ h_{prev} - \Delta h_{prev} \\ h_{prev} \end{cases} \quad (1.17)$$

ซึ่งจะได้หน้าต่างทั้งหมด 3 หน้าต่าง แล้วเลือกหน้าต่างที่ให้ค่า Bhattacharyya Coefficient สูงที่สุด ซึ่งหน้าต่างนั้นจะเป็น h_{opt} แล้วนำไปหาหน้าต่างใหม่อีกครั้งเพื่อให้ได้ขนาดหน้าต่างที่เหมาะสมโดยใช้สมการที่ (1.18)

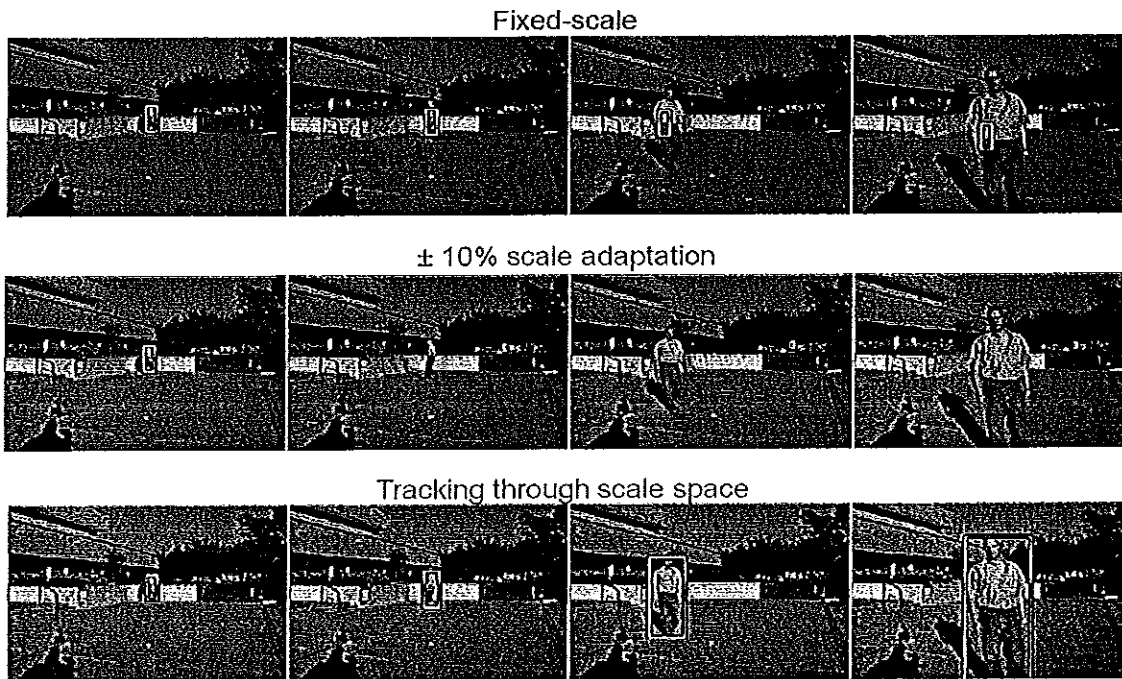
$$h_{new} = \lambda h_{opt} + (1 - \lambda)h_{prev} \quad (1.18)$$

λ มีค่า 0.1

h_{new} แทน ความกว้างของหน้าต่างหลังจากได้รับการปรับแล้ว

h_{prev} แทน ความกว้างของหน้าต่างในเฟรมก่อนหน้า

ตัวอย่างการปรับขนาดหน้าต่าง ดังภาพประกอบ 1.14 แถวบนใช้การกำหนดขนาดหน้าต่างแบบคงที่แถวกลางใช้วิธีการปรับขนาดหน้าต่างแบบ $\pm 10\%$ ของขนาดหน้าต่าง แถวล่างใช้วิธีการปรับหน้าต่างตาม Scale Space และสามารถแสดงข้อดีข้อเสียของวิธีการ Mean Shift ได้ดังตารางที่ 1.2



ภาพประกอบ 1.14 ผลจากการปรับหน้าต่างด้วยวิธีต่าง ๆ (Simakov, 2008)

ตารางที่ 1.2 เปรียบเทียบข้อดี ข้อเสียของการติดตามโดยใช้วิธี Mean Shift

ข้อดี	ข้อเสีย
<ol style="list-style-type: none"> 1. ใช้เวลาในการทำงานน้อย 2. สามารถติดตามได้ถึงแม้วัตถุเป้าหมายจะถูกบดบังเพียงบางส่วน 3. ไม่จำเป็นต้องรู้ข้อมูลล่วงหน้าเช่น รูปร่าง ขนาด 4. สามารถนำมาใช้กับข้อมูลที่มีหลายคุณลักษณะ 	<ol style="list-style-type: none"> 1. ขนาดของหน้าต่างที่ไม่เหมาะสมมีผลต่อความถูกต้องในการติดตาม 2. ถ้าพื้นหลังสีเดียวกับวัตถุเป้าหมายทำให้เกิดความผิดพลาดในการติดตาม

วิวัฒนาการของการนำ Mean Shift มาช่วยในการติดตามแสดงได้ดังตารางที่ 1.3 โดยแบ่งเป็น 8 ประเด็นได้ดังนี้

- ประเด็นที่ 1. ชนิดการติดตาม

ในการติดตามสามารถแบ่งชนิดการติดตามออกได้เป็น 2 ประเภท คือ การติดตาม 1 วัตถุ เช่น งานของ Comaniciu (2003) Collins (2003) Yang (2005) อีกประเภท คือ การติดตามวัตถุหลายวัตถุ เช่น งานของ Tang and Sang (2008) ได้เสนอวิธีการแก้ปัญหาคาดการติดตามในกรณีวัตถุ 2 วัตถุซ้อนกัน

- ประเด็นที่ 2. Preprocessing

เป็นประเด็นที่พิจารณาการ Preprocessing ภาพก่อนที่จะสร้างแบบจำลองไม่ว่าจะเป็นการเลือกรูปแบบสี การแบ่งกลุ่มสี ซึ่งรวมไปถึงการ Setup ระบบก่อนที่จะเริ่มกระบวนการติดตาม เช่น การเลือกวัตถุเพื่อใช้ในการติดตาม ระบบสีที่นิยมในการติดตามส่วนใหญ่มี 2 ระบบ คือ ระบบ RGB ในงานของ Comaniciu (2003) Birchfield (2005) และระบบสี HSV ในงานของ Zhuo (2008) และ ในงานของ Zhuo (2008) Li (2008) ได้ทำการแบ่งกลุ่มสีด้วยวิธีการ Mean Shift

- ประเด็นที่ 3. การสร้างแบบจำลอง

แบบจำลองสามารถแบ่งออกได้เป็น 2 ประเภทด้วยกัน คือ แบบจำลองหลักหรือ Template กับ แบบจำลองคู่แข่ง แบบจำลองหลักไม่มีการเปลี่ยนแปลง ซึ่งจะถูกสร้างในเฟรมแรก สำหรับแบบจำลองคู่แข่งมีการเปลี่ยนแปลงตลอดเวลาตามเฟรมที่เปลี่ยนไป โดยทั่วไปแล้วสมาชิกในแบบจำลอง มีแค่ค่าระยะห่างระหว่างตำแหน่ง Pixel กับตำแหน่งจุดศูนย์กลางของหน้าต่าง (Centroid) แล้วทำการ Normalize เพียงอย่างเดียว เช่น งานของ Comaniciu (2003) และ Li (2008) และมีการปรับปรุงมากขึ้น โดยมีการเพิ่มคุณสมบัติอื่นเข้าไป เช่น จำนวนค่าเฉลี่ย และค่าการกระจาย เช่น งานของ Birchfield (2007) และ Zhuo (2008) ในงานของ Chung-Wei Juan (2008) ได้ขยายแบบจำลองของ Birchfield โดยได้เพิ่มคุณสมบัติ ค่าเฉลี่ย และค่าการกระจายของสีเข้าไป ในงานของ Li (2008) ได้นำวิธีการของ Independent Component Analysis มาใช้กับแบบจำลอง และในงานของ Yilmaz (2007) ได้เพิ่มคุณสมบัติค่าตัวแปรอื่นเข้ามาคือ ค่ามุม เพื่อใช้พิจารณาการหมุนของหน้าต่าง

- ประเด็นที่ 4. Similarity Function

ฟังก์ชันที่ใช้ในการเปรียบเทียบความคล้ายของแบบจำลองส่วนใหญ่จะใช้ Bhattacharyya Coefficient ซึ่งเสนอครั้งแรกโดย Comaniciu ซึ่งใช้กับแบบจำลองที่มีคุณสมบัติคือค่าระยะห่างระหว่างตำแหน่ง Pixel กับตำแหน่ง Centroid แล้วทำการ Normalize ต่อมาภายหลังก็ได้มีการปรับเปลี่ยนแบบจำลอง ซึ่งทำให้ฟังก์ชันในการเปรียบเทียบมีหลายรูปแบบ

- ประเด็นที่ 5. การปรับเปลี่ยนขนาดหน้าต่าง

วิธีที่ใช้ในการปรับเปลี่ยนขนาดหน้าต่าง Comaniciu (2003) ได้เสนอวิธี $\pm 10\%$ จากขนาดหน้าต่างเดิม จากนั้น Collins (2003) ได้เสนอวิธี Scale Space ซึ่งดีกว่าวิธี $\pm 10\%$ Yilmaz (2007) ได้เสนอวิธีในการปรับเปลี่ยนหน้าต่างตามรูปร่างของวัตถุ Zhuo (2007) และ Zhuo (2008) ได้เสนอวิธีใหม่ซึ่งดีกว่าวิธี $\pm 10\%$ และ Scale Space

- ประเด็นที่ 6. การแก้ปัญหาพื้นหลัง

ปัญหาพื้นหลัง เป็นสาเหตุสำคัญที่ทำให้ความถูกต้องในการติดตามมีค่าน้อยลง วิธีในการแก้ปัญหาพื้นหลัง ส่วนใหญ่จะทำการขยายหน้าต่างเพื่อหาพื้นหลัง เช่น ในงานของ Comaniciu (2003) และ Chung-Wei Juan (2008)

- ประเด็นที่ 7. ความเร็ว

ในประเด็นของความเร็วมีหลายวิธีด้วยกันไม่ว่าจะเป็น วัดโดย Big O เช่น ในงานของ Comaniciu (2003) Li (2008) วัดโดยใช้จำนวน Iterative ต่อเฟรม เช่น งานของ Yang (2005) Zhuo (2007) และ Zhuo (2008) และวัดโดยใช้เวลาต่อ 1 รอบการทำงาน เช่น งานของ Chung-Wai Juan (2008)

- ประเด็นที่ 8. ความถูกต้อง

ในการเปรียบเทียบความถูกต้องมีหลายวิธีด้วยกัน เช่น ใช้วิธี Euclidean Distance ใช้ในงานของ Birchfield (2005) Birchfield (2007) และ Zhuo-Lin Jiang (2007) วิธี RMSE ใช้ในงานของ Zhuo-Lin Jiang (2008) และ วิธี Sum of Squared Difference ใช้ในงานของ Yang (2005)

ตารางที่ 1.3 วิวัฒนาการการติดตามโดยวิธี Mean Shift

ประเด็น		Comaniciu 2003	Colin 2003	Yang 2005	Birchfield 2005	Birchfield 2007	Yilmaz 2007	Zhuo-Lin Jiang 2007	Zhuo-Lin Jiang 2008	Li 2008	Chung-Wei Juan 2008	Li Tang Sang 2008
1. ขั้นตอนการติดตาม	ติดตามวัตถุที่จัดกลุ่ม	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	ติดตามวัตถุหลายวัตถุ											✓
2. Preprocessing	HSV							✓				
	RGB	✓		✓	✓		✓	✓		✓	✓	
	จัดกลุ่มสี								✓	✓		
	ระยะทาง	✓	✓					✓				✓
3. การสร้างแบบจำลอง	จำนวน, Mean, covariance of pixel					✓			✓			
	จำนวน, Mean and covariance of pixel, Mean and covariance of color										✓	
	ระยะทาง และสี			✓								
	ICA									✓		
	อื่นๆ				✓		✓					
	Bhattacharyya coefficient	✓					✓	✓				✓
4. Similarity function	อื่นๆ			✓	✓	✓			✓	✓	✓	
	±3%	✓								✓		
	Scale Space		✓									
5. ปริมาณค่าพารามิเตอร์	อื่นๆ						✓	✓	✓		✓	

ตารางที่ 1.3 วิวัฒนาการการติดตามโดยวิธี Mean Shift (ต่อ)

ประเด็น	Comaniciu 2003	Colin 2003	Yang 2005	Birchfield 2005	Birchfield 2007	Yilmaz 2007	Zhuo-Lin Jiang 2007	Zhuo-Lin Jiang 2008	Li 2008	Chung-Wen Juan 2008	Li Tang Sang 2008
6. แก้ไขความผิดพลาด	✓										
7. ความเร็ว	✓		✓				✓	✓	✓	✓	✓
8. ความถูกต้อง				✓	✓		✓	✓	✓	✓	
ประยุกต์ใช้เทคนิคต่าง อื่น ๆ										✓	
BigO											
Iterative/frame											
เวลา/รอบการทดลอง											
Euclidean distance					✓		✓		✓	✓	
RMSE								✓			
Sum of squared/difference			✓								

1.3 วัตถุประสงค์ของงานวิจัย

เพื่อพัฒนาขั้นตอนวิธีในการติดตามวัตถุโดยใช้วิธีการ Mean Shift ให้มีความถูกต้องมากขึ้น

1.4 ขอบเขตและวิธีการดำเนินงาน

- 1.4.1 เลือกวัตถุที่สนใจในการติดตามด้วยวิธี Manual
- 1.4.2 หน้าต่างที่ใช้แสดงวัตถุเป้าหมายในการติดตามเป็นรูปสี่เหลี่ยมและมีขนาดคงที่
- 1.4.3 วัตถุที่ใช้ในการติดตามเป็นวัตถุที่มีการเคลื่อนที่ เช่น คนเดินหรือวิ่ง
- 1.4.4 ข้อมูลนำเข้าเป็นไฟล์ Video

1.5 ขั้นตอนและระยะเวลาการดำเนินงาน

- 1.5.1 วิธีการดำเนินการวิจัย
 - 1) ศึกษางานวิจัย เอกสารที่เกี่ยวข้องและกำหนดขอบเขตของปัญหาให้ชัดเจน
 - 2) ศึกษาเทคนิคที่เกี่ยวข้องที่ใช้ในการติดตามรวมถึงการทดลองเบื้องต้น
 - 3) ศึกษาเครื่องมือที่ช่วยในการติดตาม
 - 4) วิเคราะห์และออกแบบการทดลอง
 - 5) เขียนบทความวิจัย
 - 6) สรุปและวิเคราะห์ผลการประเมินการใช้งานระบบ
 - 7) จัดทำเอกสารประกอบการวิจัย
- 1.5.2 ระยะเวลาที่ใช้ในการวิจัย
พฤศจิกายน พ.ศ. 2551 – พฤษภาคม พ.ศ. 2553
- 1.5.3 แผนการดำเนินการวิจัย
แผนการดำเนินการวิจัยแสดงดังตารางที่ 1.4

ตารางที่ 1.4 แผนการดำเนินการวิจัย

กิจกรรม/ขั้นตอนการดำเนินงาน	2551												2552					2553				
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	1	2	3	4	5
	1. ศึกษางานวิจัยและเอกสารที่เกี่ยวข้อง	■	■	■	■	■	■	■	■	■	■	■	■									
2. ศึกษาเทคนิคที่เกี่ยวข้องในงานวิจัย	■	■	■	■	■	■	■	■	■	■	■	■										
3. ศึกษาเครื่องมือที่ใช้			■	■	■	■	■	■	■	■	■	■										
4. เขียนบทความวิจัย									■	■	■	■	■	■								
5. วิเคราะห์และออกแบบระบบ							■	■	■	■	■	■	■	■								
6. พัฒนาระบบ									■	■	■	■	■	■								
7. ทดสอบและประเมินระบบ									■	■	■	■	■	■								
8. สรุปและวิเคราะห์ผล											■	■	■	■	■	■						
9. จัดทำเอกสารวิทยานิพนธ์												■	■	■	■	■	■					

1.6 สถานที่และเครื่องมือที่ใช้

1.6.1 สถานที่ดำเนินการวิจัย

ห้องปฏิบัติการ Artificial Intelligence Research LAB (AIR) คอมพิวเตอร์ CS207 ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์

1.6.2 เครื่องมือและอุปกรณ์ที่ใช้

1.6.2.1 ด้านฮาร์ดแวร์

- 1) CPU Intel Pentium 4 3.00 GHz.
- 2) RAM 1.5 GB
- 3) Hard disk 80 GB
- 4) Camera webcam

1.6.2.2 ด้านซอฟต์แวร์

- 1) Visual studio .NET 2008 (VC++, C#)
- 2) OpenCV (Library)
- 3) Microsoft Access 2007

1.7 ประโยชน์ที่คาดว่าจะได้รับของการวิจัย

นำหลักการที่ได้ไปประยุกต์ใช้ในการปรับปรุงประสิทธิภาพในการติดตามวัตถุในกรณีที่มีสีพื้นหลังมีสีเดียวกับวัตถุ ซึ่งเพิ่มความถูกต้องในการติดตามและรวดเร็ว สามารถนำมาใช้ในหลายด้าน เช่น ทางด้านการรักษาความปลอดภัยใช้การติดตามของกล้องวงจรปิด เพื่อใช้ในการตรวจหาเหตุการณ์หรือวัตถุที่น่าสงสัย

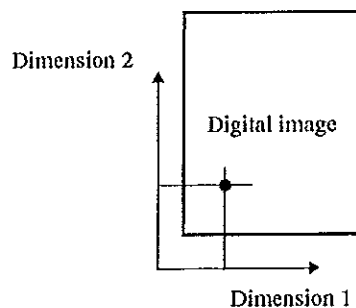
บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

บทนี้จะกล่าวถึงทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องกับการติดตามวัตถุ (Object Tracking) ซึ่งมีดังนี้ การประมวลผลภาพ (Image Processing) การวิเคราะห์วิดีโอ (Video Analysis) การจดจำรูปแบบ (Pattern Recognition) และ การติดตามวัตถุ (Object Tracking) รวมไปถึงขั้นตอนวิธีในการติดตามวัตถุ

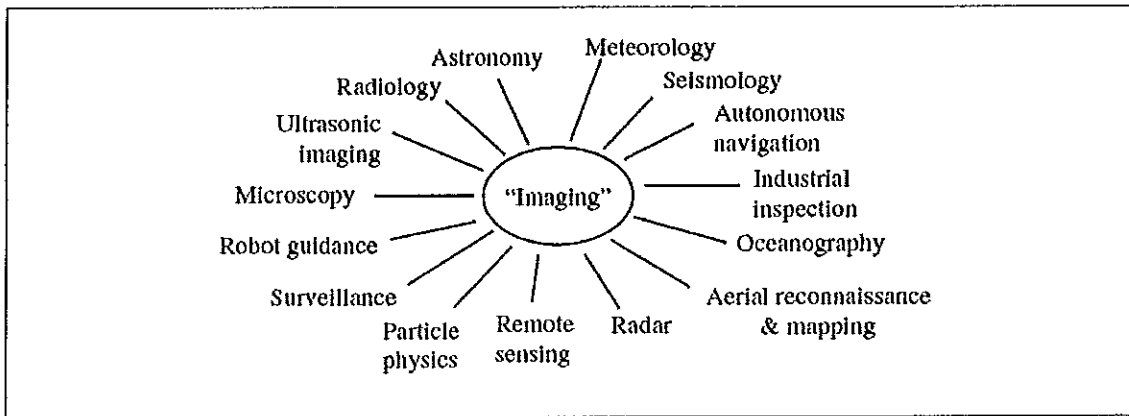
2.1 การประมวลผลภาพ

การประมวลผลภาพ คือ กระบวนการในการประมวลผลกับภาพโดยข้อมูลนำเข้า (Input) เป็นภาพ และ ข้อมูลออก (Output) เป็นภาพ โดยทั่วไปภาพจะประกอบด้วย 2 มิติ คือ ความกว้างและความสูง ดังภาพประกอบ 2.1



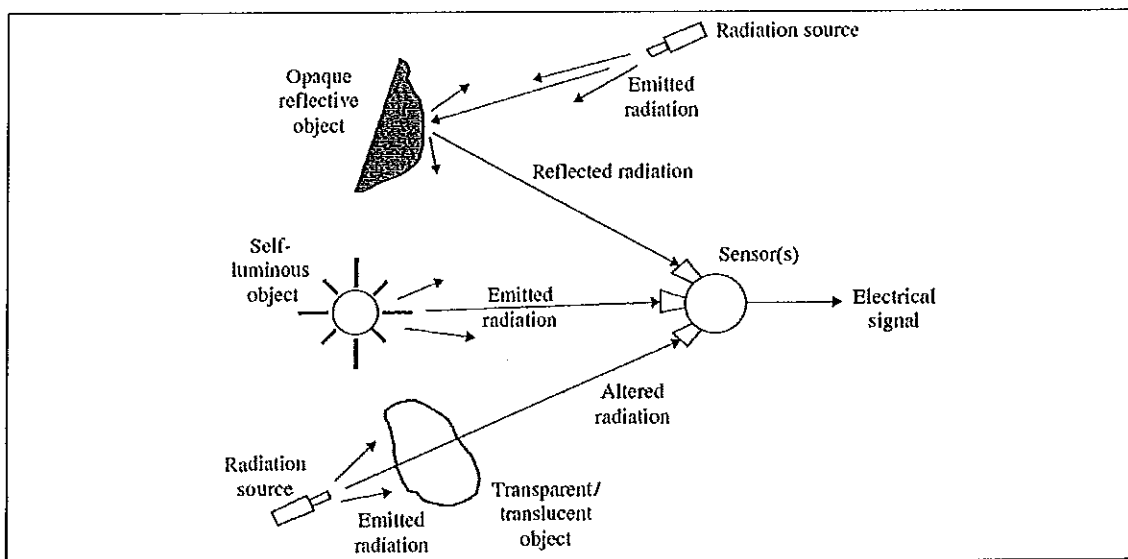
ภาพประกอบ 2.1 องค์ประกอบของภาพ (Bovik, 2009)

งานที่ใช้ภาพมีหลายงาน เช่น งานทางด้านการรักษาความปลอดภัยโดยใช้กล้องวงจรปิด (Video Surveillance) การใช้ภาพอุลตราซาวด์เพื่อใช้ในการวินิจฉัยโรค เป็นต้น ตัวอย่างงาน ดังภาพประกอบ 2.2



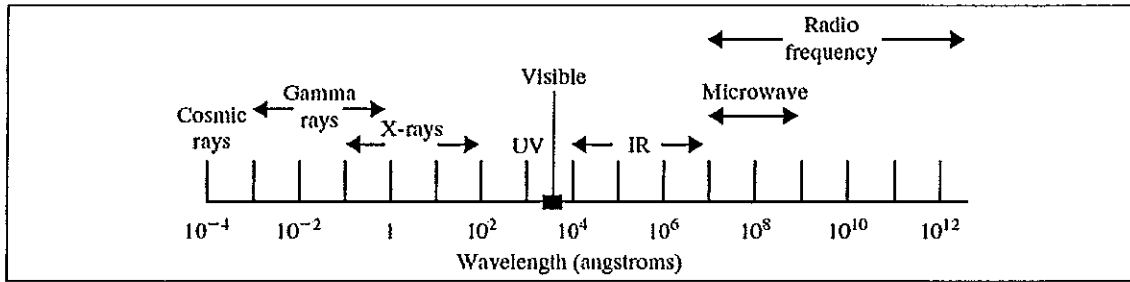
ภาพประกอบ 2.2 ตัวอย่างงานทางด้าน การประมวลผลภาพ (Bovik, 2009)

ในการรวบรวมสัญญาณก่อนที่จะมาเป็นภาพต้องใช้ตัวเก็บสัญญาณ (Sensor) สัญญาณส่วนใหญ่มาจากหลายแหล่งด้วยกันไม่ว่าจะเป็นรังสีที่มาจากดวงอาทิตย์ รังสีที่มาจากวัตถุเอง รังสีที่สะท้อนจากวัตถุ ดังภาพประกอบ 2.3



ภาพประกอบ 2.3 สัญญาณจากหลายแหล่ง (Bovik, 2009)

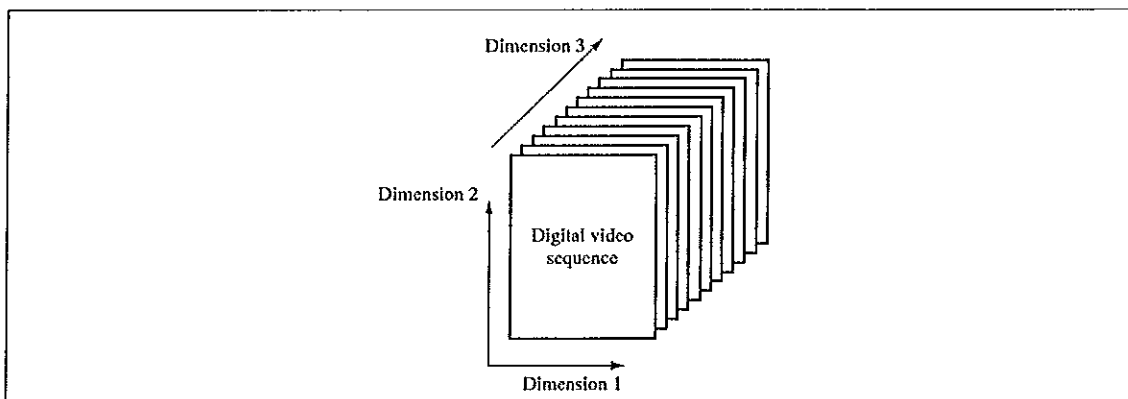
สัญญาณแสงที่ได้จากวัตถุก่อนที่จะสร้างเป็นภาพนั้น ตาของมนุษย์รับได้แค่ความยาวคลื่นช่วงหนึ่งที่เรียกว่าคลื่นแสงที่มองเห็น ซึ่งอยู่ในช่วง UV กับ IR ถือว่าน้อยมากเมื่อเปรียบเทียบกับช่วงคลื่นแสงทั้งหมด ตัวอย่างของ Spectrum ดังภาพประกอบ 2.4



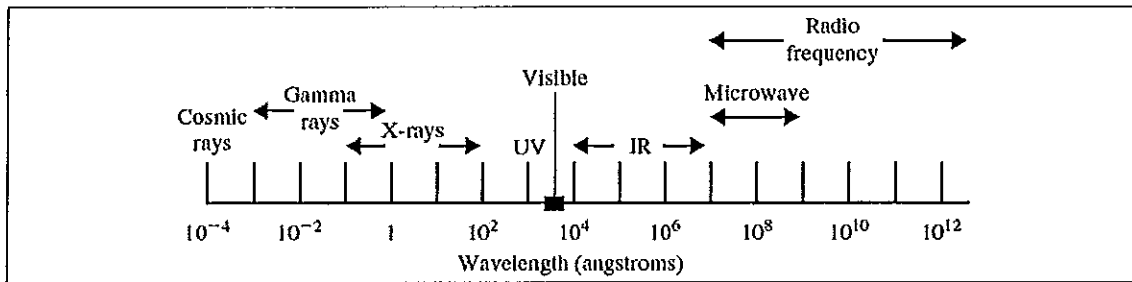
ภาพประกอบ 2.4 Spectrum ของคลื่นแม่เหล็กไฟฟ้า

2.2 การวิเคราะห์วิดีโอ (Video Analysis)

การวิเคราะห์วิดีโอ คือ การศึกษาขั้นตอนวิธีของการเคลื่อนที่ของภาพบนวิดีโอ ซึ่งแสดงในรูปแบบ Digital (0 และ 1) ซึ่งเป็นรูปแบบที่คอมพิวเตอร์สามารถอ่านได้ ในการวิเคราะห์วิดีโอจะเกี่ยวข้องกับเวลา ซึ่งโดยทั่วไปแล้วจะเป็นการเคลื่อนที่หรือการเปลี่ยนแปลงของวัตถุ และในปัจจุบันเนื่องจากอุปกรณ์ที่ใช้ในการจัดการเกี่ยวกับภาพมีความเร็วมากขึ้นและราคาถูกลง จึงทำให้การวิจัยเกี่ยวกับการวิเคราะห์ภาพที่ได้จากวิดีโอมีมากขึ้น โดยปกติแล้ววิดีโอประกอบด้วย 3 มิติ คือ (x,y,t) โดย x y แสดงตำแหน่ง และ t เป็นดัชนีเฟรม ดังภาพประกอบ 2.5 ซึ่งมีความแตกต่างจากการประมวลผลภาพซึ่งมี 2 มิติ



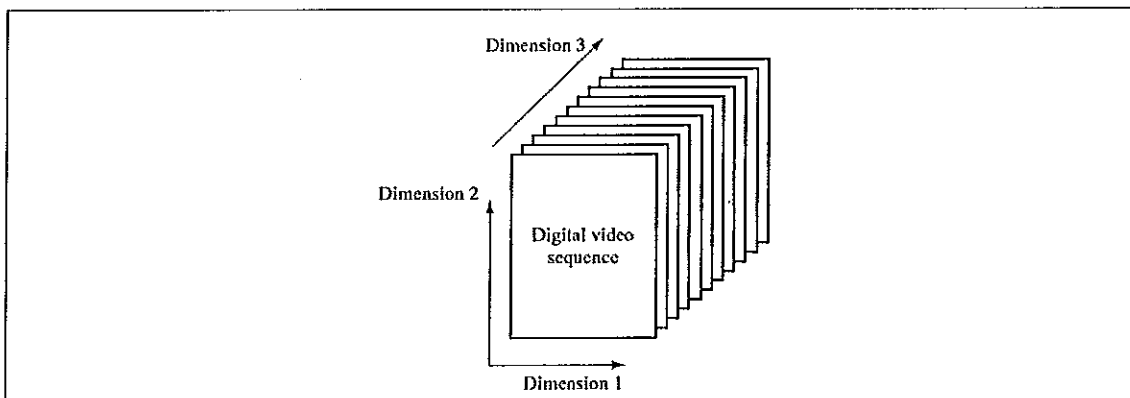
ภาพประกอบ 2.5 องค์ประกอบของวิดีโอ (Bovik, 2009)



ภาพประกอบ 2.4 Spectrum ของคลื่นแม่เหล็กไฟฟ้า

2.2 การวิเคราะห์วิดีโอ (Video Analysis)

การวิเคราะห์วิดีโอ คือ การศึกษาขั้นตอนวิธีของการเคลื่อนที่ของภาพบนวิดีโอ ซึ่งแสดงในรูปแบบ Digital (0 และ 1) ซึ่งเป็นรูปแบบที่คอมพิวเตอร์สามารถอ่านได้ ในการวิเคราะห์วิดีโอจะเกี่ยวข้องกับเวลา ซึ่งโดยทั่วไปแล้วจะเป็นการเคลื่อนที่หรือการเปลี่ยนแปลงของวัตถุ และในปัจจุบันเนื่องจากอุปกรณ์ที่ใช้ในการจัดการเกี่ยวกับภาพมีความเร็วมากขึ้นและราคาถูกลง จึงทำให้การวิจัยเกี่ยวกับการวิเคราะห์ภาพที่ได้จากวิดีโอมีมากขึ้น โดยปกติแล้ววิดีโอประกอบด้วย 3 มิติ คือ (x,y,t) โดย x y แสดงตำแหน่ง และ t เป็นดัชนีเฟรม ดังภาพประกอบ 2.5 ซึ่งมีความแตกต่างจากการประมวลผลภาพซึ่งมี 2 มิติ



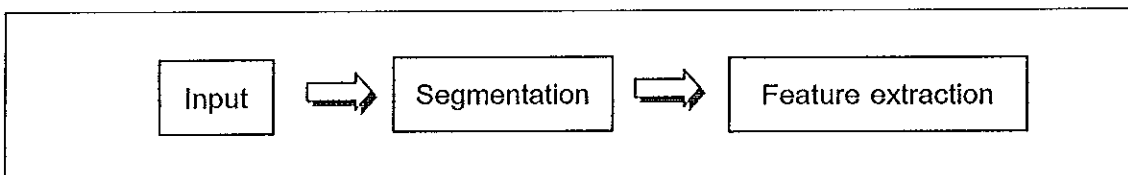
ภาพประกอบ 2.5 องค์ประกอบของวิดีโอ (Bovik, 2009)

2.3 การจดจำรูปแบบ (Pattern Recognition)

การจดจำรูปแบบจัดอยู่ในสาขา Machine Learning เป้าหมายของการจดจำรูปแบบ คือ การแบ่งแยก (Classification) จำแนก หรือ การหารูปแบบ (Pattern) การจดจำรูปแบบนำไปใช้ในหลายงานไม่ว่าจะเป็น การจดจำคำพูด (Speech Recognition) การติดตามวัตถุ (Object Tracking) ขั้นตอนในการหารูปแบบมีดังนี้

2.3.1 การสร้างแบบจำลองหลัก (Model)

การสร้างแบบจำลองหลัก สามารถสร้างโดยใช้ข้อมูลนำเข้า (Input) จากนั้นก็เลือกข้อมูลที่ต้องการ (Segmentation) แล้วนำข้อมูลที่ได้ไปหาคุณสมบัติ (Feature) ซึ่งจะได้ต้นแบบเพื่อใช้ในการเปรียบเทียบ ตัวอย่างการสร้างแบบจำลองดังภาพประกอบ 2.6



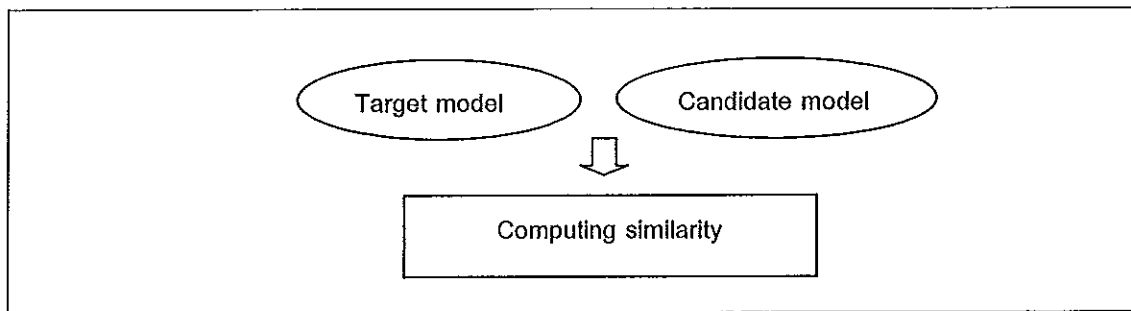
ภาพประกอบ 2.6 การสร้างแบบจำลอง

2.3.2 การสร้างแบบจำลองคู่แข่ง (Candidate Model)

ในการสร้างแบบจำลองคู่แข่งนั้นใช้ข้อมูลนำเข้า (Input) มีขั้นตอนการทำงานเช่นเดียวกันกับในข้อ 2.3.1

2.3.3 การเปรียบเทียบความคล้าย (Similarity)

เป็นขั้นตอนในการเปรียบเทียบ ความคล้ายระหว่างแบบจำลองหลักและแบบจำลองคู่แข่ง ในการเปรียบเทียบจะดูค่าความคล้าย ดังภาพประกอบ 2.7



ภาพประกอบ 2.7 การเปรียบเทียบความคล้ายระหว่าง Target Model และ Candidate Model

โดยทั่วไปแล้วในการแบ่งกลุ่มมีหลายขั้นตอนวิธีในการแบ่งกลุ่ม เช่น Neural Network และ Decision Tree ซึ่งในแต่ละวิธี ข้างในจะมีวิธีการในการสร้างแบบจำลองและการวัดความคล้ายแตกต่างกัน

2.4 การติดตามวัตถุ (Object Tracking)

การติดตามวัตถุ คือ การติดตามวัตถุบนวิดีโอซึ่งก่อนที่จะติดตามจะต้องมีการตรวจหาวัตถุที่สนใจ (Object Detection) จากนั้นก็จะเป็นการติดตามวัตถุ การติดตามวัตถุสามารถแบ่งออกได้เป็น 3 ประเภท ดังที่ได้กล่าวในบทที่ 1 ในที่นี้จะกล่าวถึงการติดตามโดยใช้ Mean Shift ซึ่งจัดอยู่ในประเภท Kernel Tracking ซึ่งเป็นการติดตามที่ใช้รูปเรขาคณิต เช่น สีเหลี่ยม วงกลม เพื่อแทนวัตถุ สามารถเปรียบเทียบการติดตามวัตถุได้ดังตารางที่ 2.1 โดยมีทั้งหมด 6 ขั้นตอน คือ 1) การเริ่มต้น 2) การสร้างแบบจำลอง 3) การประเมินความคล้าย 4) การเคลื่อนไปยังตำแหน่งใหม่ 5) การให้ค่าพื้นหลัง และ 6) การปรับตำแหน่งหน้าต่าง ตามขั้นตอนวิธีของ Comaniciu (2003) และ ขั้นตอนวิธีของ Li (2008) ดังนี้

ตารางที่ 2.1 เปรียบเทียบขั้นตอนวิธีของการติดตามวัตถุ

ขั้นตอน	(Gomanclu, 2003)	(LI, 2008)
1 การเริ่มต้น	<ol style="list-style-type: none"> 1. การเลือกตำแหน่งเป้าหมาย 2. การเลือกระบบสี (RGB) 3. การเลือกจำนวนของ bin (32x32x32) 4. การเลือกเคอร์เนล $k(x) = \begin{cases} 1-x^2 & \text{if } 0 \leq x < 1 \\ 0 & \text{otherwise} \end{cases}$ <ol style="list-style-type: none"> 5. การเลือกคุณสมบัติของแบบจำลองในที่นี้ใช้ระยะห่างระหว่างจุดในกรอบสี่เหลี่ยมกับจุดศูนย์กลางของหน้าต่างแล้วผ่านฟังก์ชันเคอร์เนลจากนั้นก็ทำการ Normalize 	<ol style="list-style-type: none"> 1. การเลือกตำแหน่งเป้าหมาย 2. การเลือกระบบสี (RGB) 3. การเลือกจำนวนของ bin (32x32x32) 4. การเลือกเคอร์เนล $k(x) = \begin{cases} 1-x^2 & \text{if } 0 \leq x < 1 \\ 0 & \text{otherwise} \end{cases}$ <ol style="list-style-type: none"> 5. การเลือกคุณสมบัติของแบบจำลองในที่นี้ใช้ระยะห่างระหว่างจุดในกรอบสี่เหลี่ยมกับจุดศูนย์กลางของหน้าต่างแล้วผ่านฟังก์ชันเคอร์เนลจากนั้นก็ทำการ Normalize 6. แบ่งสี RGB ในแต่ละช่องภายในเป็น 8 ส่วน ($K_{u,j} = 8$)
2 การสร้างแบบจำลอง	<p>เฟรมแรก สร้างแบบจำลองหลัก</p> $\hat{q}_u = C \sum_{i=1}^n k\left(\left \frac{y-x_i}{h}\right \right) \delta_u(b(x_i))$ <p>เฟรมอื่น สร้างแบบจำลองคู่แข่ง</p> $\hat{p}_u(\hat{y}_0) = C \sum_{i=1}^n k\left(\left \frac{y-x_i}{h}\right \right) \delta_u(b(x_i))$	<p>เฟรมแรก สร้างแบบจำลองหลัก</p> $\hat{q}_u = C \sum_{i=1}^n k\left(\left \frac{y-x_i}{h}\right \right) G_u \delta_u(b(x_i))$ <p>เฟรมอื่น สร้างแบบจำลองคู่แข่ง</p> $\hat{p}_u(\hat{y}_0) = C \sum_{i=1}^n k\left(\left \frac{y-x_i}{h}\right \right) G'_u \delta_u(b(x_i))$ <p>โดย</p> $G'_u = \prod_{j=1}^3 \sum_{k_j=1}^{K_{u,j}} q'_{u,j,k}$ $G_u = \prod_{j=1}^3 \sum_{k_j=1}^{K_{u,j}} q_{u,j,k}$ $q_{u,j,k} = \frac{m_{u,j,k_j}}{n_u}$ $q'_{u,j,k} = \frac{m'_{u,j,k_j}}{n'_u}$ <p>m_{u,j,k_j} แทน จำนวน pixel ที่อยู่ที่ j ในส่วนของ k_j</p> <p>n_u แทน จำนวน pixel ทั้งหมดที่อยู่ในสี u</p>
3 การประเมินความคล้าย	$\rho[\hat{p}(\hat{y}_0), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u}$	$\rho[\hat{p}(\hat{y}_0), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u} p(G_u G'_u)$ $p(G_u G'_u) = \prod_{j=1}^3 \sum_{k_j=1}^{K_{u,j}} \sqrt{q_{u,j,k} q'_{u,j,k}}$
4 การเคลื่อนไปยังตำแหน่งใหม่	$\hat{y}_1 = \frac{\sum_{i=1}^n \sum_{u=1}^m x_i \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \delta_u(b(x_i))}{\sum_{i=1}^n \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \delta_u(b(x_i))}$	$\hat{y}_1 = \frac{\sum_{i=1}^n \sum_{u=1}^m x_i \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} p(G_u G'_u) \delta_u(b(x_i))}{\sum_{i=1}^n \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} p(G_u G'_u) \delta_u(b(x_i))}$
5 การให้ค้ำขึ้นหลัง	ขยายหน้าต่าง 10 % เพื่อคำนวณ Weight Background แล้วไปลดค่าของแบบจำลอง	ไม่มี

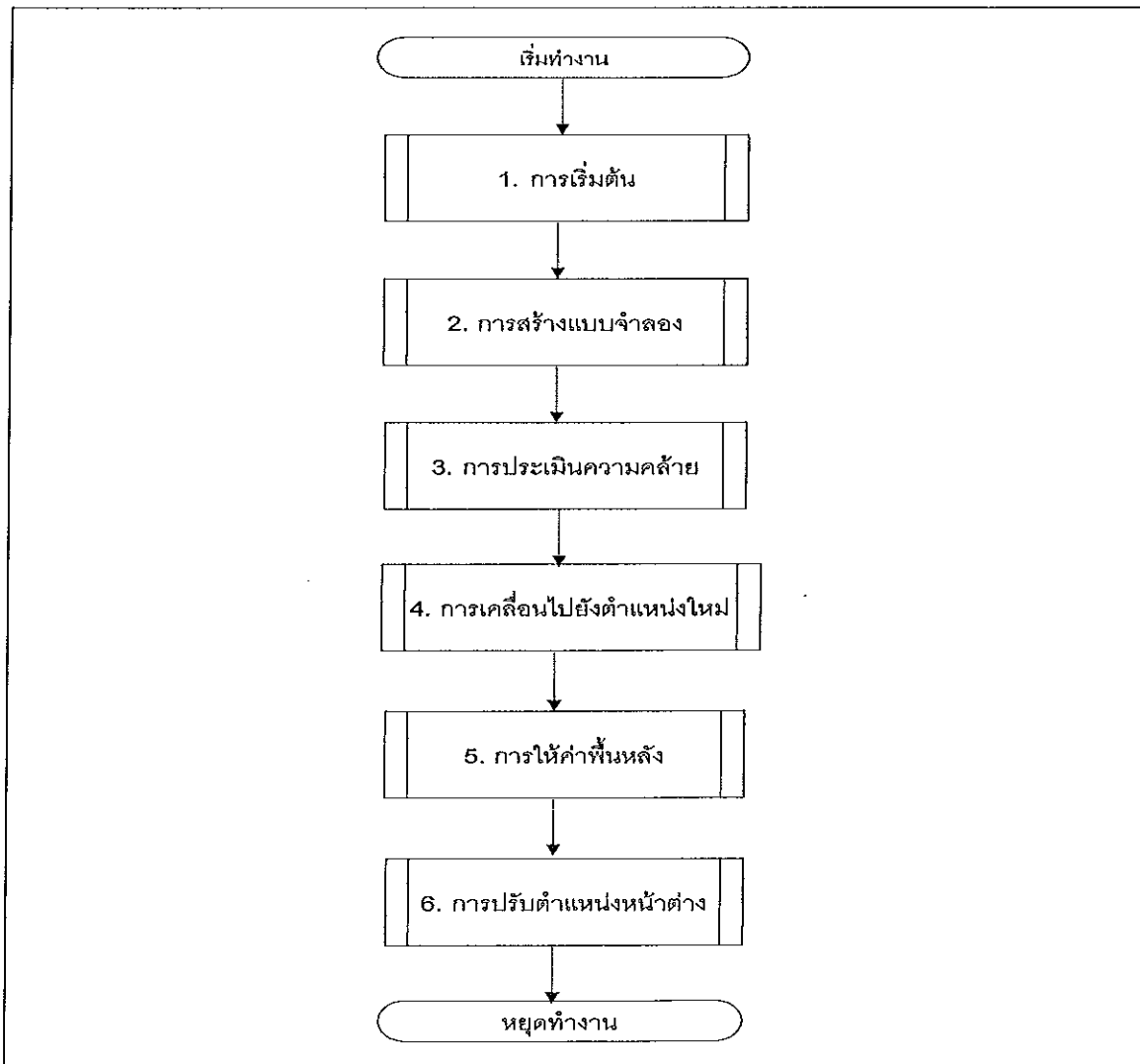
ตารางที่ 2.1 เปรียบเทียบขั้นตอนวิธีของการติดตามวัตถุ (ต่อ)

ขั้นตอน	(Comaniciu, 2003)	(LI, 2008)
<p>6 การปรับ ตำแหน่ง หน้าต่าง</p>	<p>1. สร้างแบบจำลองคู่แข่ง ตามขั้นตอนที่ 2</p> <p>2. ประเมินความคล้ายแบบจำลองคู่แข่ง</p> $\rho[\hat{p}(\hat{y}_1), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_1) \hat{q}_u}$ <p>3. ปรับตำแหน่งหน้าต่าง</p> <p>While $\rho[\hat{p}(\hat{y}_1), \hat{q}] < \rho[\hat{p}(\hat{y}_0), \hat{q}]$</p> $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_1 + \hat{y}_0)$ <p>Evaluate $\rho[\hat{p}(\hat{y}_1), \hat{q}]$</p> <p>End While</p> <p>If $\ \hat{y}_1 - \hat{y}_0\ < \varepsilon$ Stop</p> <p>Else $\hat{y}_0 \leftarrow \hat{y}_1$ Goto step 4</p>	<p>1. สร้างแบบจำลองคู่แข่ง ตามขั้นตอนที่ 2</p> <p>2. ประเมินความคล้ายแบบจำลองคู่แข่ง</p> $\rho[\hat{p}(\hat{y}_1), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_1) \hat{q}_u}$ <p>3. ปรับตำแหน่งหน้าต่าง</p> <p>While $\rho[\hat{p}(\hat{y}_1), \hat{q}] < \rho[\hat{p}(\hat{y}_0), \hat{q}]$</p> $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_1 + \hat{y}_0)$ <p>Evaluate $\rho[\hat{p}(\hat{y}_1), \hat{q}]$</p> <p>End While</p> <p>If $\ \hat{y}_1 - \hat{y}_0\ < \varepsilon$ Stop</p> <p>Else $\hat{y}_0 \leftarrow \hat{y}_1$ Goto step 4</p>

บทที่ 3

แบบจำลอง Mean Shift Tracking Algorithm Using Adaptive Candidate

วิทยานิพนธ์นี้ได้เสนอแบบจำลองสำหรับขั้นตอนวิธีในการติดตามวัตถุโดยมีชื่อว่าแบบจำลอง Mean Shift Tracking Algorithm Using Adaptive Candidate ซึ่งมีชื่อย่อว่า MST_AC กระบวนการทำงานของขั้นตอนวิธี MST_AC ประกอบไปด้วยขั้นตอนทั้งหมด 6 ขั้นตอน คือ 1) การเริ่มต้น 2) การสร้างแบบจำลอง 3) การประเมินความคล้าย 4) การเคลื่อนไปยังตำแหน่งใหม่ 5) การให้ค่าพื้นที่หลัง และ 6) การปรับตำแหน่งหน้าต่าง ดังแสดงในภาพประกอบ 3.1 โดยรายละเอียดของแบบจำลองของแต่ละขั้นตอนสามารถอธิบายได้ดังนี้

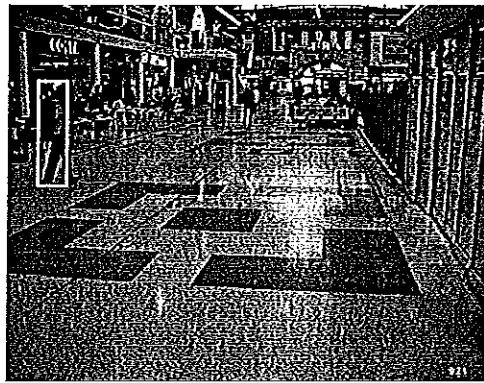


ภาพประกอบ 3.1 แบบจำลองการทำงาน MST_AC

3.1 การเริ่มต้น

วิธีการของการกำหนดค่าเริ่มต้นก่อนที่จะดำเนินการสำหรับการติดตามประกอบด้วยขั้นตอนการทำงานทั้งหมด 5 ขั้นตอนย่อย ดังนี้

3.1.1 การเลือกตำแหน่งเป้าหมาย (Object Detection) เป็นขั้นตอนในการกำหนดพื้นที่ของวัตถุเป้าหมายเพื่อที่จะสร้างแบบจำลองหลัก และรวมไปถึงการกำหนดจุดศูนย์กลางที่ใช้แทนวัตถุ ดังแสดงในภาพประกอบ 3.2 เป็นตัวอย่างการเลือกตำแหน่งเป้าหมายโดยในพื้นที่ของกรอบสี่เหลี่ยมแสดงถึงพื้นที่ของวัตถุที่จะนำไปใช้ในการติดตาม



ภาพประกอบ 3.2 ตัวอย่างการเลือกพื้นที่ตำแหน่งเป้าหมาย

3.1.2 การเลือกระบบสี ระบบสีที่ใช้มีหลายระบบเช่น HSV RGB ในที่นี้ใช้ระบบสี RGB

3.1.3 การเลือกจำนวนของ bin จำนวนของ bin ที่ใช้มีหลายขนาด ในที่นี้ ใช้ขนาด bin จำนวน $32 \times 32 \times 32$ การที่เลือกจำนวน bin นี้ สาเหตุเนื่องจาก ถ้าเลือกจำนวน bin น้อยเกินไป เช่น $16 \times 16 \times 16$ จะทำให้จำนวนสีน้อยลงซึ่งมีโอกาสที่สีพื้นหลังมีสีเดียวกับสีวัตถุเป้าหมายมาก แต่ถ้าเลือกจำนวน bin มากเกินไป เช่น $64 \times 64 \times 64$ ทำให้เสียเวลาในการคำนวณ

3.1.4 การเลือกเคอร์เนล ในที่นี้เลือก Epanechnikov Kernel ดังสมการที่ (1.8)

3.1.5 การเลือกคุณสมบัติของแบบจำลอง ในที่นี้คือ การกระจายของแต่ละสี (Bin) ตั้งแต่ $u=1$ ถึง m โดยค่าสีนั้นใช้สมการที่ (1.5) เพื่อคำนวณค่าสีโดยแปลงจาก 3 มิติ (R G B) ให้อยู่ในรูปมิติเดียว จากนั้นค่าในแต่ละสีหาได้จากค่าระยะห่างระหว่างตำแหน่งกับจุดศูนย์กลางของหน้าต่างแล้วส่งผ่านไปยังฟังก์ชันเคอร์เนลดังสมการที่ (1.8) แล้วทำการ Normalize

3.2 การสร้างแบบจำลอง

การสร้างสิ่งที่จะอธิบายวัตถุเป้าหมายโดยใช้คุณสมบัติในข้อ 3.1.5 ในการสร้างแบบจำลองสามารถแบ่งออกได้เป็น 2 ส่วนดังนี้

3.2.1 การสร้างแบบจำลองหลัก (\hat{q}_u) แบบจำลองหลักในที่นี้คือตัวต้นแบบที่เป็นตัวอธิบายวัตถุโดยใช้คุณสมบัติในข้อ 3.1.5 ในการสร้างแบบจำลองหลักนั้นจะสร้างแค่ครั้งเดียวตามสมการที่ (1.1) โดยทั่วไปการสร้างแบบจำลองหลักนั้นจะสร้างจากเฟรมเริ่มต้นที่ใช้ในการติดตาม จากนั้นก็ทำการเก็บสำรองเฟรม (pf) และตำแหน่งจุดศูนย์กลางของหน้าต่าง \hat{y}_{old}

3.2.2 การสร้างแบบจำลองคู่แข่ง (\hat{p}_u) เป็นแบบจำลองที่สร้างขึ้นเพื่อใช้เปรียบเทียบกับแบบจำลองหลัก ซึ่งมีความแตกต่างกับแบบจำลองคู่แข่งของ Comanicu (2003) โดยมีการเพิ่ม $g(x_i)$ เพื่อลบค่าส่วนที่เป็นพื้นหลังที่มีสีเดียวกับวัตถุออกไปวิธีในการสร้างดังสมการที่ (3.1) (3.2) และ (3.3) โดยในเฟรมแรกนั้น $g(x_i)$ มีค่าเป็น 0

$g(x_i)$ แทน ค่าตำแหน่งพื้นหลังที่ตำแหน่ง x_i

cf แทน เฟรมปัจจุบัน

pf แทน เฟรมก่อนหน้า

m แทน จำนวน bin ทั้งหมด

$b(x, y)$ แทน สีที่ตำแหน่ง x บนเฟรม y

$$\hat{p}_u(\hat{y}_0) = \begin{cases} \sum_{i=1}^n k \left(\left\| \frac{y-x_i}{h} \right\| \right) \delta_u(b(x_i, cf)) - g(x_i) \delta_u(b(x_i, pf)) & \text{if } b(x_i, cf) = b(x_i, pf) \\ \sum_{i=1}^n k \left(\left\| \frac{y-x_i}{h} \right\| \right) \delta_u(b(x_i, cf)) & \text{otherwise} \end{cases} \quad (3.1)$$

$$\hat{p}_u(\hat{y}_0) = \begin{cases} \hat{p}_u(\hat{y}_0) & \text{if } \hat{p}_u(\hat{y}_0) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.2)$$

$$\hat{p}_u(\hat{y}_0) = \frac{\hat{p}_u(\hat{y}_0)}{\sum_{u=1}^m \hat{p}_u(\hat{y}_0)} \quad (3.3)$$

3.3 การประเมินความคล้าย

สิ่งที่ใช้ในการเปรียบเทียบแบบจำลองหลักและแบบจำลองคู่แข่ง คือ Bhattacharyya Coefficient ดังสมการที่ (1.6)

3.4 การเคลื่อนไปยังตำแหน่งใหม่

วิธีการที่ใช้ในการขยับตำแหน่งจุดศูนย์กลางใหม่ (\hat{y}_i) ของหน้าต่าง ดังสมการที่ (1.16)

3.5 การกำหนดค่าพื้นหลัง

การกำหนดค่าตำแหน่งของพื้นหลังสำหรับการติดตามวัตถุ ประกอบด้วยการทำงานทั้งหมด 4 ขั้นตอนย่อย ดังนี้



3.5.1 การกำหนดตำแหน่งพื้นหลัง เป็นการหาตำแหน่งพื้นหลังที่มีสีเดียวกับวัตถุซึ่งหาได้จากนำตำแหน่งในพื้นที่สี่เหลี่ยมของ \hat{y}_i ที่ตำแหน่ง x_i บนเฟรมปัจจุบัน (Current Frame) หรือ cf ลบด้วยตำแหน่งพื้นที่สี่เหลี่ยมของ \hat{y}_{old} ที่ตำแหน่ง x_i บนเฟรมก่อนหน้า (Previous Frame) หรือ pf โดยที่ตำแหน่งของ x_i นั้นมีสีเดียวกับแบบจำลองหลัก ดังขั้นตอนวิธีข้างล่าง โดย ถ้าเป็นตำแหน่งพื้นหลังที่มีสีเดียวกับวัตถุค่า $g(x_i)=1$ แต่ถ้าไม่ใช่ตำแหน่งพื้นหลัง $g(x_i)=0$

area(x,y,z) แทน ตำแหน่ง x บนพื้นที่สี่เหลี่ยม y บนเฟรม z

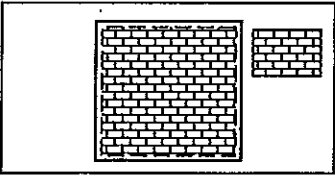
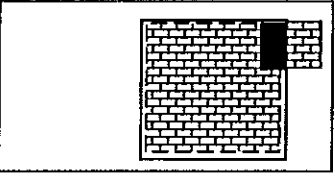
```

For i = 1 to n
If area(xi, ŷ1, cf) - area(xi, ŷold, pf) ≠ ∅ and  $\hat{g}(b(x_i, pf)) > 0$ 
g(xi) = 1
Else If area(xi, ŷold, pf) - area(xi, ŷ1, cf) ≠ ∅
g(xi) = 0
End If

```

ตัวอย่างแสดงการหาตำแหน่งพื้นหลัง ดังตารางที่ 3.1 โดย  แทนตำแหน่งบนเฟรมที่มีสีเดียวกับวัตถุ และ  แทนพื้นที่พื้นหลังที่มีสีเดียวกับวัตถุซ้อนทับกันพอดี

ตารางที่ 3.1 การหาดำแหน่งพื้นหลัง

 <p>เฟรมที่ T</p>	<p>จากภาพ เฟรมที่ T จะเห็นได้ว่ากรอบสี่เหลี่ยมล้อมรอบวัตถุ และทางด้านขวามีพื้นหลังที่มีสีเดียวกับวัตถุ</p>
 <p>เฟรมที่ T+1</p>	<p>จากภาพ เฟรมที่ T+1 กรอบสี่เหลี่ยมขยับตำแหน่งตามวัตถุ และพื้นที่ที่มีสีดำเข้ม ■ แทนตำแหน่งพื้นหลังที่มีสีเดียวกับวัตถุซ้อนทับกัน (วัตถุเคลื่อนที่ไปทางขวามือ)</p>

3.5.2 การคำนวณค่าระยะห่าง คือขั้นตอนในการคำนวณค่าระยะห่างซึ่งประกอบด้วยสมการ (3.4) (3.5) และ (3.6) เพื่อใช้ในขั้นตอนต่อไป Δx คือระยะห่างจากจุด $\hat{y}_{(old,x)}$ ในแนวแกน x Δy คือระยะห่างจากจุด $\hat{y}_{(old,y)}$ ในแนวแกน y

$$\Delta x = \frac{\hat{y}_{(l,x)} - \hat{y}_{(old,x)}}{2} \quad (3.4)$$

$$\Delta y = \frac{\hat{y}_{(l,y)} - \hat{y}_{(old,y)}}{2} \quad (3.5)$$

$$l = \sqrt{\Delta x^2 + \Delta y^2} \quad (3.6)$$

3.5.3 สร้างจุดศูนย์กลางชั่วคราว (\hat{y}_2) จุดศูนย์กลางชั่วคราวคือตำแหน่งจุดศูนย์กลางที่สร้างขึ้นใหม่เพื่อใช้คำนวณค่าพื้นหลัง ซึ่งเป็นตำแหน่งที่แตกต่างจาก \hat{y}_{old} และ \hat{y}_1 คือ ตำแหน่งของ \hat{y}_2 ไม่จำเป็นต้องมีตำแหน่งอยู่ตรงกลางของกรอบสี่เหลี่ยม ตำแหน่งของ \hat{y}_2 จะขยับไปแนวทางที่มีทิศเดียวกับทิศที่วัตถุเป้าหมายเคลื่อนที่ไปดังสมการ (3.4) และ สมการ (3.5) ส่วนขนาดของการเคลื่อนที่จะเป็นไปตามสมการ (3.6) ในการหาดำแหน่ง \hat{y}_2 หาได้จากสมการที่ (3.7) และ (3.8)

โดย m แทน ความชันของ จุด \hat{y}_2 กับ จุด \hat{y}_1

l แทน ระยะห่างระหว่าง จุด \hat{y}_2 กับ จุด \hat{y}_1

แสดงการหาดำแหน่ง $\hat{y}_{(2,x)}$

$$m = \frac{\hat{y}_{(2,y)} - \hat{y}_{(1,y)}}{\hat{y}_{(2,x)} - \hat{y}_{(1,x)}} = \frac{\Delta y}{\Delta x} \quad \text{if } \Delta x \neq 0 \quad (3.7)$$

$$l^2 = (\hat{y}_{(2,y)} - \hat{y}_{(1,y)})^2 + (\hat{y}_{(2,x)} - \hat{y}_{(1,x)})^2 \quad (3.8)$$

จากสมการที่ (3.7) แทน $\hat{y}_{(2,y)} - \hat{y}_{(1,y)}$ ด้วย $m(\hat{y}_{(2,x)} - \hat{y}_{(1,x)})$ ของสมการ (3.8)

$$l^2 = m^2 (\hat{y}_{(2,x)} - \hat{y}_{(1,x)})^2 + (\hat{y}_{(2,x)} - \hat{y}_{(1,x)})^2$$

$$(\hat{y}_{(2,x)} - \hat{y}_{(1,x)})^2 = \frac{l^2}{(m^2 + 1)} \quad \text{if } \Delta x \neq 0$$

$$\hat{y}_{(2,x)} = \hat{y}_{(1,x)} \pm \frac{l}{\sqrt{(m^2 + 1)}} \quad \text{if } \Delta x \neq 0$$

แสดงการหาค่าแห่ง $\hat{y}_{(2,y)}$

$$l^2 = (\hat{y}_{(2,y)} - \hat{y}_{(1,y)})^2 + \left(\frac{\hat{y}_{(2,y)} - \hat{y}_{(1,y)}}{m}\right)^2 \quad \text{if } \Delta x \neq 0$$

$$l^2 = (\hat{y}_{(2,y)} - \hat{y}_{(1,y)})^2 \left(\frac{1+m^2}{m^2}\right) \quad \text{if } \Delta x \neq 0$$


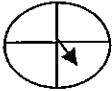



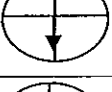
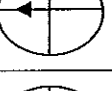
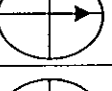
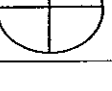
$$\hat{y}_{(2,y)} = \hat{y}_{(1,y)} \pm \frac{l \times m}{\sqrt{m^2 + 1}} \quad \text{if } \Delta x \neq 0$$

สรุปการหาค่าแห่ง \hat{y}_2 ทุกกรณี ดังสมการ (3.9)

$$\hat{y}_2 = \begin{cases} \hat{y}_{(2,x)} = \hat{y}_{(1,x)} + \frac{l}{\sqrt{m^2 + 1}}, & \hat{y}_{(2,y)} = \hat{y}_{(1,y)} + \frac{l \times m}{\sqrt{m^2 + 1}} \\ & \text{if } \Delta x > 0 \\ \hat{y}_{(2,x)} = \hat{y}_{(1,x)} - \frac{l}{\sqrt{m^2 + 1}}, & \hat{y}_{(2,y)} = \hat{y}_{(1,y)} - \frac{l \times m}{\sqrt{m^2 + 1}} \\ & \text{if } \Delta x < 0 \\ \hat{y}_{(2,x)} = \hat{y}_{(1,x)}, & \hat{y}_{(2,y)} = \hat{y}_{(1,y)} + l \quad \text{if } \Delta x = 0 \text{ and } \Delta y > 0 \\ \hat{y}_{(2,x)} = \hat{y}_{(1,x)}, & \hat{y}_{(2,y)} = \hat{y}_{(1,y)} - l \quad \text{if } \Delta x = 0 \text{ and } \Delta y < 0 \\ \hat{y}_{(2,x)} = \hat{y}_{(1,x)}, & \hat{y}_{(2,y)} = \hat{y}_{(1,y)} \quad \text{if } \Delta x = 0 \text{ and } \Delta y = 0 \end{cases} \quad (3.9)$$

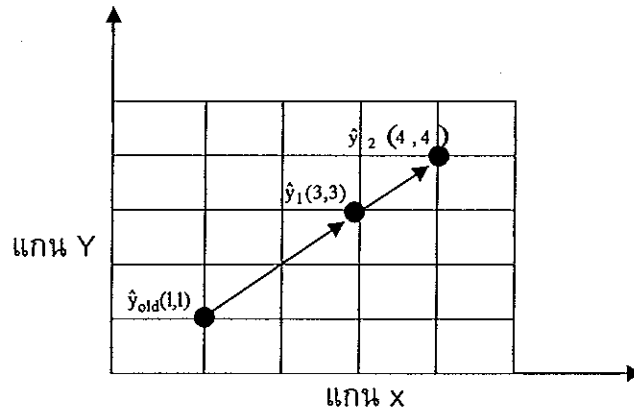
ตารางที่ 3.2 แสดงตัวอย่างการปรับตำแหน่งของ \hat{y}_2 ตามทิศทางของวัตถุ โดยในคอลัมน์ 2 แสดงตำแหน่งของจุดศูนย์กลาง \hat{y}_1 ซึ่งอยู่กึ่งกลางของวงกลม ส่วนปลายลูกศร แสดงตำแหน่งของจุดศูนย์กลางชั่วคราว \hat{y}_2

ตารางที่ 3.2 ตัวอย่างการปรับตำแหน่งของ \hat{v}_2 ตามทิศทางของวัตถุที่เป็นไปได้ทุกกรณี

ทิศการเคลื่อนที่ของวัตถุ	ตำแหน่งและทิศทางของ \hat{v}_2
ไปทางขวาบน	
ไปทางขวาล่าง	
ไปทางซ้ายบน	
ไปทางซ้ายล่าง	
ไปทางบน	
ไปทางล่าง	
ไปทางซ้าย	
ไปทางขวา	
อยู่กับที่	

ตัวอย่างการหาดำแหน่ง \hat{y}_2

กรณีที่ 1 $\Delta x > 0$ โดยกำหนดให้จุดศูนย์กลางปัจจุบัน \hat{y}_1 ที่ตำแหน่งพิกัด (3,3) และจุดศูนย์กลางของเฟรมก่อนหน้า \hat{y}_{old} ที่ตำแหน่งพิกัด (1,1) ดังภาพประกอบ 3.3



ภาพประกอบ 3.3 การหาดำแหน่ง \hat{y}_2 ในกรณีที่ $\Delta x > 0$

หาค่า Δx Δy และ m

$$\Delta x = \frac{\hat{y}_{(1,x)} - \hat{y}_{(old,x)}}{2} = \frac{(3-1)}{2} = 1$$

$$\Delta y = \frac{\hat{y}_{(1,y)} - \hat{y}_{(old,y)}}{2} = \frac{(3-1)}{2} = 1$$

เนื่องจาก $\Delta x > 0$

$$m = \frac{\Delta y}{\Delta x} = \frac{1}{1} = 1$$

หาค่า l

$$l = \sqrt{\Delta x^2 + \Delta y^2} = \sqrt{1^2 + 1^2} = 1.414$$

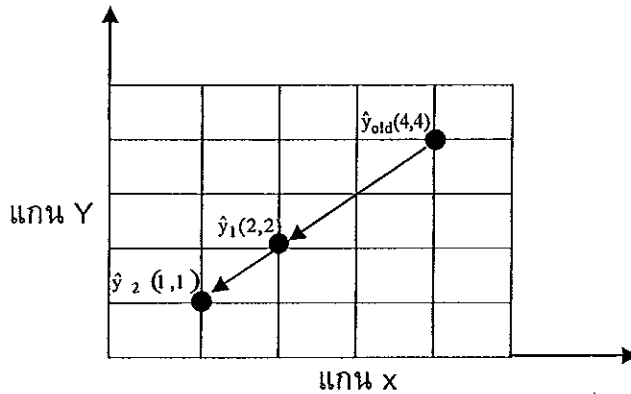
หาค่า $\hat{y}_{(2,x)}$ และ $\hat{y}_{(2,y)}$

$$\hat{y}_{(2,x)} = \hat{y}_{(1,x)} + \frac{l}{\sqrt{m^2 + 1}} = 3 + \frac{1.414}{\sqrt{1^2 + 1}} = 4$$

$$\hat{y}_{(2,y)} = \hat{y}_{(1,y)} + \frac{l \times m}{\sqrt{m^2 + 1}} = 3 + \frac{1.414 \times 1}{\sqrt{1^2 + 1}} = 4$$

ดังนั้นตำแหน่ง $\hat{y}_2 = (4,4)$

กรณีที่ 2 $\Delta x < 0$ โดยกำหนดให้จุดศูนย์กลางปัจจุบัน \hat{y}_1 ที่ตำแหน่งพิกัด (2,2) และจุดศูนย์กลางของเฟรมก่อนหน้า \hat{y}_{old} ที่ตำแหน่งพิกัด (4,4) ดังภาพประกอบ 3.4



ภาพประกอบ 3.4 การหาดำแหน่ง \hat{y}_2 ในกรณีที่ $\Delta x < 0$

หาค่า Δx Δy และ m

$$\Delta x = \frac{\hat{y}(1,x) - \hat{y}(old,x)}{2} = \frac{(2-4)}{2} = -1$$

$$\Delta y = \frac{\hat{y}(1,y) - \hat{y}(old,y)}{2} = \frac{(2-4)}{2} = -1$$

เนื่องจาก $\Delta x < 0$

$$m = \frac{\Delta y}{\Delta x} = \frac{-1}{-1} = 1$$

หาค่า l

$$l = \sqrt{\Delta x^2 + \Delta y^2} = \sqrt{(-1)^2 + (-1)^2} = 1.414$$

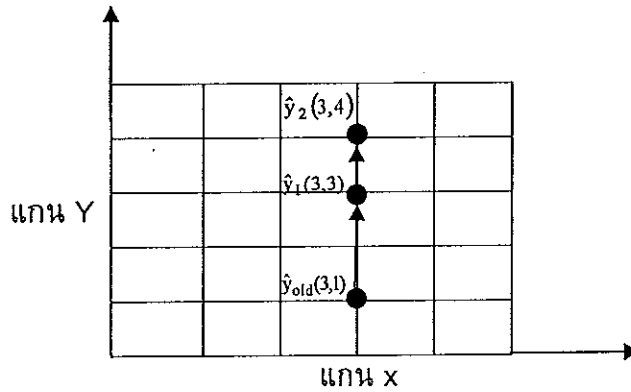
หาค่า $\hat{y}(2,x)$ และ $\hat{y}(2,y)$

$$\hat{y}(2,x) = \hat{y}(1,x) - \frac{l}{\sqrt{m^2 + 1}} = 2 - \frac{1.414}{\sqrt{1^2 + 1}} = 1$$

$$\hat{y}(2,y) = \hat{y}(1,y) - \frac{l \times m}{\sqrt{m^2 + 1}} = 2 - \frac{1.414 \times 1}{\sqrt{1^2 + 1}} = 1$$

ดังนั้นตำแหน่ง $\hat{y}_2 = (1,1)$

กรณีที่ 3 $\Delta x = 0$ และ $\Delta y > 0$ โดยกำหนดให้จุดศูนย์กลางปัจจุบัน \hat{y}_1 ที่ตำแหน่งพิกัด (3,3) และจุดศูนย์กลางของเฟรมก่อนหน้า \hat{y}_{old} ที่ตำแหน่งพิกัด (3,1) ดังภาพประกอบ 3.5



ภาพประกอบ 3.5 การหาตำแหน่ง \hat{y}_2 ในกรณีที่ $\Delta x = 0$ $\Delta y > 0$

หาค่า Δx Δy และ m

$$\Delta x = \frac{\hat{y}_{(1,x)} - \hat{y}_{(old,x)}}{2} = \frac{(3-3)}{2} = 0$$

$$\Delta y = \frac{\hat{y}_{(1,y)} - \hat{y}_{(old,y)}}{2} = \frac{(3-1)}{2} = 1$$

เนื่องจาก $\Delta x = 0$ และ $\Delta y > 0$

หาค่า l

$$l = \sqrt{\Delta x^2 + \Delta y^2} = \sqrt{0^2 + 1^2} = 1$$

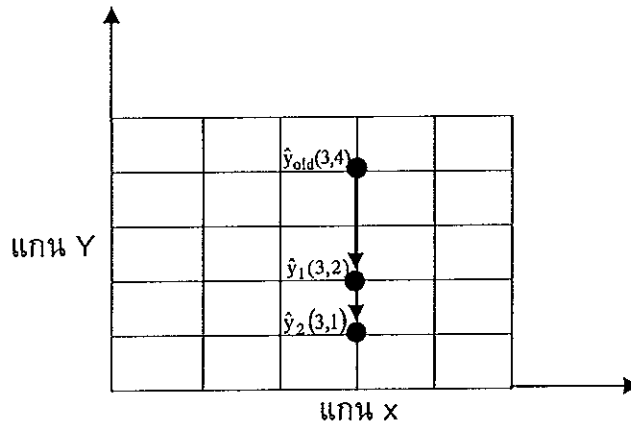
หาค่า $\hat{y}_{(2,x)}$ และ $\hat{y}_{(2,y)}$

$$\hat{y}_{(2,x)} = \hat{y}_{(1,x)} = 3$$

$$\hat{y}_{(2,y)} = \hat{y}_{(1,y)} + l = 3 + 1 = 4$$

ดังนั้นตำแหน่ง $\hat{y}_2 = (3,4)$

กรณีที่ 4 $\Delta x = 0$ และ $\Delta y < 0$ โดยกำหนดให้จุดศูนย์กลางปัจจุบัน \hat{y}_1 ที่ตำแหน่งพิกัด (3,2) และจุดศูนย์กลางของเฟรมก่อนหน้า \hat{y}_{old} ที่ตำแหน่งพิกัด (3,4) ดังภาพประกอบ 3.6



ภาพประกอบ 3.6 การหาตำแหน่ง \hat{y}_2 ในกรณีที่ $\Delta x = 0$ $\Delta y < 0$

หาค่า Δx Δy และ m

$$\Delta x = \frac{\hat{y}_{(1,x)} - \hat{y}_{(old,x)}}{2} = \frac{(3-3)}{2} = 0$$

$$\Delta y = \frac{\hat{y}_{(1,y)} - \hat{y}_{(old,y)}}{2} = \frac{(2-4)}{2} = -1$$

เนื่องจาก $\Delta x = 0$ และ $\Delta y < 0$

หาค่า l

$$l = \sqrt{\Delta x^2 + \Delta y^2} = \sqrt{0^2 + (-1)^2} = 1$$

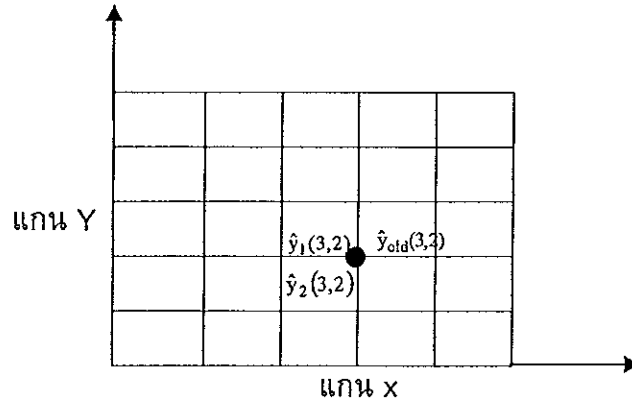
หาค่า $\hat{y}_{(2,x)}$ และ $\hat{y}_{(2,y)}$

$$\hat{y}_{(2,x)} = \hat{y}_{(1,x)} = 3$$

$$\hat{y}_{(2,y)} = \hat{y}_{(1,y)} - l = 2 - 1 = 1$$

ดังนั้นตำแหน่ง $\hat{y}_2 = (3,1)$

กรณีที่ 5 $\Delta x = 0$ และ $\Delta y = 0$ โดยกำหนดให้จุดศูนย์กลางปัจจุบัน \hat{y}_1 ที่ตำแหน่งพิกัด (3,2) และจุดศูนย์กลางของเฟรมก่อนหน้า \hat{y}_{old} ที่ตำแหน่งพิกัด (3,2) ดังภาพประกอบ 3.7



ภาพประกอบ 3.7 การหาดำแหน่ง \hat{y}_2 ในกรณี $\Delta x = 0$ $\Delta y = 0$

หาค่า Δx Δy และ m

$$\Delta x = \frac{\hat{y}_{(1,x)} - \hat{y}_{(old,x)}}{2} = \frac{(3-3)}{2} = 0$$

$$\Delta y = \frac{\hat{y}_{(1,y)} - \hat{y}_{(old,y)}}{2} = \frac{(2-2)}{2} = 0$$

เนื่องจาก $\Delta x = 0$ และ $\Delta y = 0$

หาค่า l

$$l = \sqrt{\Delta x^2 + \Delta y^2} = \sqrt{0^2 + 0^2} = 0$$

หาค่า $\hat{y}_{(2,x)}$ และ $\hat{y}_{(2,y)}$

$$\hat{y}_{(2,x)} = \hat{y}_{(1,x)} = 3$$

$$\hat{y}_{(2,y)} = \hat{y}_{(1,y)} = 2$$

ดังนั้นตำแหน่ง $\hat{y}_2 = (3,2)$

3.5.4 การให้ค่าพื้นหลัง เป็นการกำหนดค่าให้กับตำแหน่งของพื้นหลัง ซึ่งค่าพื้นหลังนี้คือความน่าจะเป็นพื้นหลัง โดยความน่าจะเป็นพื้นหลังนี้เป็นการทำนายโดยเฟรมก่อนหน้าที้นำร่องเอาไว้ ซึ่งเป็นเฟรมที่มีความน่าเชื่อถือมากที่สุดเนื่องจากเป็นเฟรมที่มีเวลาใกล้เคียงกับเฟรมปัจจุบันมากที่สุด และในขณะที่เฟรมปัจจุบันไม่สามารถที่จะหาตำแหน่งพื้นหลังเนื่องจากยังอยู่ในกระบวนการค้นหาตำแหน่งวัตถุ ในการให้ค่าตำแหน่งของพื้นหลังนั้นจะพิจารณาจากตำแหน่งที่ขยับหน้าต่างซึ่งหาได้จาก Mean Shift ซึ่งการขยับหน้าต่างจะไปในทิศทางที่เข้าใกล้กับวัตถุเป้าหมายจึงถือได้ว่าตำแหน่งใหม่ที่ได้นั้นเป็นตำแหน่งของวัตถุ ซึ่งค่าความน่าจะเป็นพื้นหลังนั้นจะมีค่ามากเมื่อตำแหน่งพื้นหลังนั้นอยู่ในทิศตรงข้ามกับการเคลื่อนที่ของวัตถุ วิธีที่ใช้ในการกำหนดค่าพื้นหลังดังขั้นตอนวิธีข้างล่าง

- W แทน ความกว้างของกรอบสี่เหลี่ยม
- H แทน ความสูงของกรอบสี่เหลี่ยม
- n แทน จำนวน Pixels ที่อยู่บนหน้าต่าง
- i แทน ดัชนีในการอ้างถึงตำแหน่งบนหน้าต่าง
- (x_i, y_i) แทน พิกัดตำแหน่งที่ i

For $i = 1$ to n

If $g(x_i) > 0$

$$g(x_i) = \sqrt{((x_{(i,x)} - \hat{y}_{(2,x)})^2 / W)^2 + ((x_{(i,y)} - \hat{y}_{(2,y)})^2 / H)^2}$$

3.6 การปรับตำแหน่งของหน้าต่าง

การปรับตำแหน่งเพื่อหาแบบจำลองคู่แข่งที่มีความเหมือนใกล้เคียงกับแบบจำลองหลักมากที่สุดสำหรับการทำงานสามารถแบ่งออกเป็น 4 ขั้นตอนย่อยดังนี้

3.6.1 สร้างแบบจำลองคู่แข่งใหม่ ขั้นตอนการสร้างแบบจำลองคู่แข่งใหม่จะมีขั้นตอนการสร้างเช่นเดียวกันกับในขั้นตอน 3.2.2 โดยใช้กรอบสี่เหลี่ยม (\hat{y}_1)

3.6.2 ประเมินความคล้ายใหม่ ดังสมการที่ (1.6)

3.6.3 ปรับตำแหน่งหน้าต่างใหม่ ขั้นตอนในการทำงานสำหรับการปรับตำแหน่งหน้าต่างใหม่สามารถอธิบายได้ดังนี้

```

While  $\rho[\hat{p}(\hat{y}_1), \hat{q}] < \rho[\hat{p}(\hat{y}_0), \hat{q}]$ 
 $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_1 + \hat{y}_0)$ 
Evaluate  $\rho[\hat{p}(\hat{y}_1), \hat{q}]$ 
End While
If  $\|\hat{y}_1 - \hat{y}_0\| < \varepsilon$  Stop
Else  $\hat{y}_0 \leftarrow \hat{y}_1$  and calculate new  $\hat{y}_1$ 

```

3.6.4 เก็บสำรองเฟรม (pf) และ จุดศูนย์กลางของหน้าต่าง $\hat{y}_{(old)}$

ตารางที่ 3.3 แสดงแบบจำลองที่นำเสนอคือ MST_AC (Mean Shift Tracking Using Adaptive Candidate Model) เปรียบเทียบกับงานเดิม (ตารางที่ 2.1) คือ MS_BG_10 (Comaniciu, 2003) เป็นการทำงานของ Mean Shift กับการพิจารณาพื้นที่หลังโดยขยายหน้าต่าง 10 % และ MS_ICA (Peihua, 2008) เป็นการทำงานของ Mean Shift แบบ Independent Component Analysis ส่วนที่มีการแรงใน คอลัมน์ ของ MST_AC คือส่วนที่แตกต่างกับ Mean Shift ทั่วไป

ตารางที่ 3.3 เปรียบเทียบขั้นตอนวิธีของการติดตามวัตถุ

ขั้นตอน	MS_EG_10 (Comaniciu, 2003)	MS_CA (Péhu, 2008)	MST_AC
การเริ่มต้น	<ol style="list-style-type: none"> 1. การเลือกตำแหน่งเป้าหมาย 2. การเลือกแบบสี (RGB) 3. การเลือกขนาดของ bin (32x32x32) 4. การเลือก เคอร์เนล $k(x) = \begin{cases} 1-x^2 & \text{if } 0 \leq x < 1 \\ 0 & \text{otherwise} \end{cases}$ <ol style="list-style-type: none"> 5. การเลือกคุณสมบัติของแบบจำลองใหม่ที่ใช้ระหว่างจุดศูนย์กลางของหน้าต่างและผ่านฟังก์ชันเคอร์เนล จากนั้นทำการ Normalize 	<ol style="list-style-type: none"> 1. การเลือกตำแหน่งเป้าหมาย 2. การเลือกแบบสี (RGB) 3. การเลือกขนาดของ bin (32x32x32) 4. การเลือก เคอร์เนล $k(x) = \begin{cases} 1-x^2 & \text{if } 0 \leq x < 1 \\ 0 & \text{otherwise} \end{cases}$ <ol style="list-style-type: none"> 5. การเลือกคุณสมบัติของแบบจำลองใหม่ที่ใช้ระหว่างจุดศูนย์กลางของหน้าต่างและผ่านฟังก์ชันเคอร์เนล จากนั้นทำการ Normalize 	<ol style="list-style-type: none"> 1. การเลือกตำแหน่งเป้าหมาย 2. การเลือกแบบสี (RGB) 3. การเลือกขนาดของ bin (32x32x32) 4. การเลือก เคอร์เนล $k(x) = \begin{cases} 1-x^2 & \text{if } 0 \leq x < 1 \\ 0 & \text{otherwise} \end{cases}$ <ol style="list-style-type: none"> 5. การเลือกคุณสมบัติของแบบจำลองใหม่ที่ใช้ระหว่างจุดศูนย์กลางของหน้าต่างและผ่านฟังก์ชันเคอร์เนล จากนั้นทำการ Normalize
การสร้างแบบจำลอง	<p>เฟรมแรก (สร้างแบบจำลองหลัก)</p> $\hat{q}_u = C \sum_{i=1}^n k\left(\frac{y-x_i}{h}\right) \delta_u(b(x_i))$ <p>เฟรมอื่น (สร้างแบบจำลองคู่แข่ง)</p> $\hat{p}_u(\hat{y}_0) = C \sum_{i=1}^n k\left(\frac{y-x_i}{h}\right) \delta_u(b(x_i))$ <p>โดย</p> $G'_u = \prod_{j=1}^{K_{u,j}} q'_{u,j,k}$ $G_u = \prod_{j=1}^{K_{u,j}} q_{u,j,k}$	<p>เฟรมแรก (สร้างแบบจำลองหลัก)</p> $\hat{q}_u = C \sum_{i=1}^n k\left(\frac{y-x_i}{h}\right) G_u \delta_u(b(x_i))$ <p>เฟรมอื่น (สร้างแบบจำลองคู่แข่ง)</p> $\hat{p}_u(\hat{y}_0) = C \sum_{i=1}^n k\left(\frac{y-x_i}{h}\right) G'_u \delta_u(b(x_i))$ <p>โดย</p> $G'_u = \prod_{j=1}^{K_{u,j}} q'_{u,j,k}$ $G_u = \prod_{j=1}^{K_{u,j}} q_{u,j,k}$	<p>เฟรมแรก (สร้างแบบจำลองหลัก)</p> $\hat{q}_u = C \sum_{i=1}^n k\left(\frac{y-x_i}{h}\right) \delta_u(b(x_i))$ <p>เก็บค่าของเฟรม (pf) และ จุดศูนย์กลางของหน้าต่างต่าง (\hat{y}_0)</p> <p>เฟรมอื่น (สร้างแบบจำลองคู่แข่ง)</p> $\hat{p}_u(\hat{y}_0) = \begin{cases} \sum_{i=1}^n k\left(\frac{y-x_i}{h}\right) \delta_u(b(x_i, cf)) - g(x_i) \delta_u(b(x_i, pf)) & \text{if } b(x_i, cf) = b(x_i, pf) \\ \sum_{i=1}^n k\left(\frac{y-x_i}{h}\right) \delta_u(b(x_i, cf)) & \text{otherwise} \end{cases}$

ตารางที่ 3.3 เปรียบเทียบขั้นตอนวิธีของการติดตามวัตถุ (ต่อ)

ขั้นตอน	MS_EG_10 (Comaniciu, 2003)	MS_ICA (Painua, 2008)	MST_AC
2 การสร้าง แบบจำลอง		$q_{u,j,k} = \frac{m_{u,j,k_j}}{n_u}$ $q'_{u,j,k} = \frac{m'_{u,j,k_j}}{n'_u}$ <p>m_{u,j,k_j} แทน จำนวน pixel ที่อยู่สี j ในส่วนของ k_j n_u แทน จำนวนสีที่อยู่ในสี u</p>	$\hat{p}_u(\hat{y}_0) = \begin{cases} \hat{p}_u(\hat{y}_0) & \text{if } \hat{p}_u(\hat{y}_0) \geq 0 \\ 0 & \text{otherwise} \end{cases}$ $\hat{p}_u(\hat{y}_0) = \frac{\hat{p}_u(\hat{y}_0)}{\sum_{u=1}^m \hat{p}_u(\hat{y}_0)}$
3 การประเมิน ความคล้าย	$\rho[\hat{p}(\hat{y}_0), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u}$	$\rho[\hat{p}(\hat{y}_0), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u} p(G_u G'_u)$ $p(G_u G'_u) = \prod_{j=1}^3 \sum_{k_j=1}^{K_{u,j}} \sqrt{q_{u,j,k_j} q'_{u,j,k_j}}$	$\rho[\hat{p}(\hat{y}_0), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u}$
4 การเคลื่อนไป ยังตำแหน่ง ใหม่	$\hat{y}_1 = \frac{\sum_{i=1}^n \sum_{u=1}^m x_i \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)} \delta_u(b(x_i))}}{\sum_{i=1}^n \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)} \delta_u(b(x_i))}}$	$\hat{y}_1 = \frac{\sum_{i=1}^n \sum_{u=1}^m x_i \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)} p(G_u G'_u) \delta_u(b(x_i))}}{\sum_{i=1}^n \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)} p(G_u G'_u) \delta_u(b(x_i))}}$	$\hat{y}_1 = \frac{\sum_{i=1}^n \sum_{u=1}^m x_i \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)} \delta_u(b(x_i))}}{\sum_{i=1}^n \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)} \delta_u(b(x_i))}}$

ตารางที่ 3.3 เปรียบเทียบขั้นตอนวิธีการติดตามวัตถุ (ต่อ)

ขั้นตอน	MS_BC_10 (Comaniciu, 2003)	MS_CA (Pelhuu, 2008)	MST_AC
5 การหาค่าพื้นที่	ใช้วิธีขยายหน้าต่าง 10 % เพื่อคำนวณ Weight Background	ไม่มี	<p>แบ่งออกเป็น 4 ขั้นตอนย่อย คือ</p> <ol style="list-style-type: none"> การกำหนดตำแหน่งพื้นที่ <ul style="list-style-type: none"> For $j = 1$ to n If $\text{area}(x_j, \hat{y}_j, cf) - \text{area}(x_j, \hat{y}_{old}, pf) \neq \emptyset$ and $\hat{q}(e(x_j, pf)) > 0$ $g(x_j) = 1$ Else If $\text{area}(x_j, \hat{y}_{old}, pf) - \text{area}(x_j, \hat{y}_j, cf) \neq \emptyset$ $g(x_j) = 0$ End If คำนวณค่าระยะห่าง $\Delta x = \frac{\hat{y}(l,x) - \hat{y}(old,x)}{2}$ $\Delta y = \frac{\hat{y}(l,y) - \hat{y}(old,y)}{2}$ $l = \sqrt{\Delta x^2 + \Delta y^2}$ สร้างจุดศูนย์กลางชั่วคราว (\hat{y}_2)

ตารางที่ 3.3 เปรียบเทียบขั้นตอนวิธีของการจัดตามวัตถุ (ต่อ)

ขั้นตอน	MS_EG_10 (Comanicu, 2003)	MS_ICA (Paihua, 2008)	MST_AC
<p>5</p> <p>การให้ค่าน้ำหนัก</p>			$\hat{y}_{(2,y)} = \hat{y}_{(1,x)} + \frac{l}{\sqrt{m^2 + 1}}, \hat{y}_{(2,y)} = \hat{y}_{(1,y)} + \frac{l \times m}{\sqrt{m^2 + 1}}$ <p style="text-align: center;">if $\Delta x > 0$</p> $\hat{y}_{(2,y)} = \hat{y}_{(1,x)} - \frac{l}{\sqrt{m^2 + 1}}, \hat{y}_{(2,y)} = \hat{y}_{(1,y)} - \frac{l \times m}{\sqrt{m^2 + 1}}$ <p style="text-align: center;">if $\Delta x < 0$</p> $\hat{y}_{(2,x)} = \hat{y}_{(1,x)}, \hat{y}_{(2,y)} = \hat{y}_{(1,y)} + l \quad \text{if } \Delta x = 0 \text{ and } \Delta y > 0$ $\hat{y}_{(2,x)} = \hat{y}_{(1,x)}, \hat{y}_{(2,y)} = \hat{y}_{(1,y)} - l \quad \text{if } \Delta x = 0 \text{ and } \Delta y < 0$ $\hat{y}_{(2,x)} = \hat{y}_{(1,x)}, \hat{y}_{(2,y)} = \hat{y}_{(1,y)} \quad \text{if } \Delta x = 0 \text{ and } \Delta y = 0$ <p>4. การกำหนดค่าพื้นที่หลัง</p> <p>For $i = 1$ to n</p> <p> if $g(x_i) > 0$</p> $g(x_i) = \sqrt{((x_{(1,x)} - \hat{y}_{(2,x)})^2 / W)^2 + ((x_{(1,y)} - \hat{y}_{(2,y)})^2 / H)^2}$

ตารางที่ 3.3 เปรียบเทียบขั้นตอนวิธีของการติดตามวัตถุ (ต่อ)

ขั้นตอน	MS_BG-10 (Comanicu, 2003)	MS_ICA (Peihua, 2008)	MST_AC
<p>6</p> <p>การปรับ ตำแหน่ง หน้าต่าง</p>	<ol style="list-style-type: none"> 1. สร้างแบบจำลองคู่แข่ง ตามขั้นตอนที่ 2 2. ประเมินความคล้ายแบบจำลองคู่แข่ง $\rho[\hat{p}(\hat{y}_1), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_1) \hat{q}_u}$ <ol style="list-style-type: none"> 3. ปรับตำแหน่งหน้าต่าง <p>While $\rho[\hat{p}(\hat{y}_1), \hat{q}] < \rho[\hat{p}(\hat{y}_0), \hat{q}]$</p> $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_1 + \hat{y}_0)$ <p>Evaluate $\rho[\hat{p}(\hat{y}_1), \hat{q}]$</p> <p>End While</p> <p>If $\ \hat{y}_1 - \hat{y}_0\ < \varepsilon$ Stop</p> <p>Else $\hat{y}_0 \leftarrow \hat{y}_1$ Goto step 4</p>	<ol style="list-style-type: none"> 1. สร้างแบบจำลองคู่แข่ง ตามขั้นตอนที่ 2 2. ประเมินความคล้ายแบบจำลองคู่แข่ง $\rho[\hat{p}(\hat{y}_1), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_1) \hat{q}_u}$ <ol style="list-style-type: none"> 3. ปรับตำแหน่งหน้าต่าง <p>While $\rho[\hat{p}(\hat{y}_1), \hat{q}] < \rho[\hat{p}(\hat{y}_0), \hat{q}]$</p> $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_1 + \hat{y}_0)$ <p>Evaluate $\rho[\hat{p}(\hat{y}_1), \hat{q}]$</p> <p>End While</p> <p>If $\ \hat{y}_1 - \hat{y}_0\ < \varepsilon$ Stop</p> <p>Else $\hat{y}_0 \leftarrow \hat{y}_1$ Goto step 4</p>	<ol style="list-style-type: none"> 1. สร้างแบบจำลองคู่แข่ง ตามขั้นตอนที่ 2 2. ประเมินความคล้ายแบบจำลองคู่แข่ง $\rho[\hat{p}(\hat{y}_1), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_1) \hat{q}_u}$ <ol style="list-style-type: none"> 3. ปรับตำแหน่งหน้าต่าง <p>While $\rho[\hat{p}(\hat{y}_1), \hat{q}] < \rho[\hat{p}(\hat{y}_0), \hat{q}]$</p> $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_1 + \hat{y}_0)$ <p>Evaluate $\rho[\hat{p}(\hat{y}_1), \hat{q}]$</p> <p>End While</p> <p>If $\ \hat{y}_1 - \hat{y}_0\ < \varepsilon$ Stop</p> <p>Else $\hat{y}_0 \leftarrow \hat{y}_1$ Goto step 4</p> <p>เก็บสำรองแฟรม (pf) และ จุดศูนย์กลางของหน้าต่างต่าง (\hat{y}_0)</p>

บทที่ 4

ผลการทดลองและวิจารณ์

สำหรับบทนี้เกี่ยวกับผลการทดลอง ซึ่งแบ่งออกเป็น 3 การทดลองด้วยกัน คือ การทดลองแรกเป็นการทดลองเปรียบเทียบค่าความถูกต้องในการติดตามโดยดูจากค่าระยะทางผิดพลาดและเวลาเฉลี่ยที่ใช้ในการวนรอบ 1 ครั้งในแต่ละขั้นตอนวิธี ซึ่งประกอบด้วยขั้นตอนวิธี Mean Shift Tracking Algorithm Using Adaptive Candidate (MST_AC) เป็นขั้นตอนวิธีที่ผู้วิจัยได้นำเสนอในวิทยานิพนธ์เล่มนี้ ขั้นตอนวิธี MS (Mean Shift ทั่วไปที่ไม่มีปัญหาพื้นหลัง) ขั้นตอนวิธี MS_BG_10 (Mean Shift ทั่วไปที่มีการแก้ปัญหาพื้นหลังโดยขยายหน้าต่าง 10 % เพื่อหาหน้าหนักพื้นหลัง) และขั้นตอนวิธี MS_ICA (Mean Shift ที่นำหลักการ Independent Component Analysis มาใช้) ในการทดลองได้ใช้ข้อมูลทั้งหมด 8 ชุดข้อมูล การทดลองที่ 2 เป็นการทดลองเปรียบเทียบค่าความถูกต้องในการติดตามและเวลาเฉลี่ยที่ใช้ในการวนรอบ 1 ครั้ง โดยมีการลดจำนวน bin จาก $32 \times 32 \times 32$ เป็น $16 \times 16 \times 16$ ของแต่ละขั้นตอนวิธี และการทดลองที่ 3 สุดท้ายเป็นการทดลองเปรียบเทียบค่าความถูกต้องในการติดตามและเวลาเฉลี่ยที่ใช้ในการวนรอบ 1 ครั้ง โดยมีการปรับความคมชัดของวิดีโอที่เป็นข้อมูลทดสอบ

4.1 ชุดข้อมูลที่ใช้ในการทดลอง

ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพการทำงานของแต่ละขั้นตอนวิธี สำหรับตรวจสอบความถูกต้องในการติดตามและระยะเวลาที่ใช้ในการวนรอบ ประกอบไปด้วยชุดข้อมูล 8 ชุดข้อมูล ได้แก่ ชุดข้อมูล Airport 1 (วัตถุเป้าหมายเป็นคนเดินในสนามบินในทิศทางห่างจากกล้องสพายเป็สีดำ) ชุดข้อมูล Airport 2 (วัตถุเป้าหมายเป็นคนเดินในสนามบินจากขวาไปซ้าย ใส่เสื้อสีขาว) ชุดข้อมูล Airport 3 (วัตถุเป้าหมายเป็นคนเดินในสนามบินจากขวาไปซ้ายใส่ชุดดำ) ชุดข้อมูล Airport 4 (วัตถุเป้าหมายเป็นคนเดินในสนามบินจากขวาไปซ้าย ใส่เสื้อสีดำ กางเกงสีขาว) ชุดข้อมูล ComSci 1 (วัตถุเป้าหมายเป็นคนเดินนอกสถานที่ในทิศทางห่างจากกล้อง ใส่เสื้อสีขาว) ชุดข้อมูล ComSci 2 (วัตถุเป้าหมายเป็นคนเดินนอกสถานที่ในทิศทางห่าง

จากกล้อง ใส่อีเสื่อสีฟ้า) ชุดข้อมูล ComSci 3 (วัตถุเป้าหมายเป็นคนวิ่งนอกสถานที่ในทิศทางห่าง และใกล้จากกล้อง ใส่อีเสื่อสีขาว) และสุดท้าย ชุดข้อมูล ComSci 4 (วัตถุเป้าหมายเป็นคนเดินนอกสถานที่ ใส่อีเสื่อสีดำเดินตัดพื้นหลังสีดำ) ซึ่งแต่ละชุดข้อมูลแสดงรายละเอียดดังตารางที่ 4.1

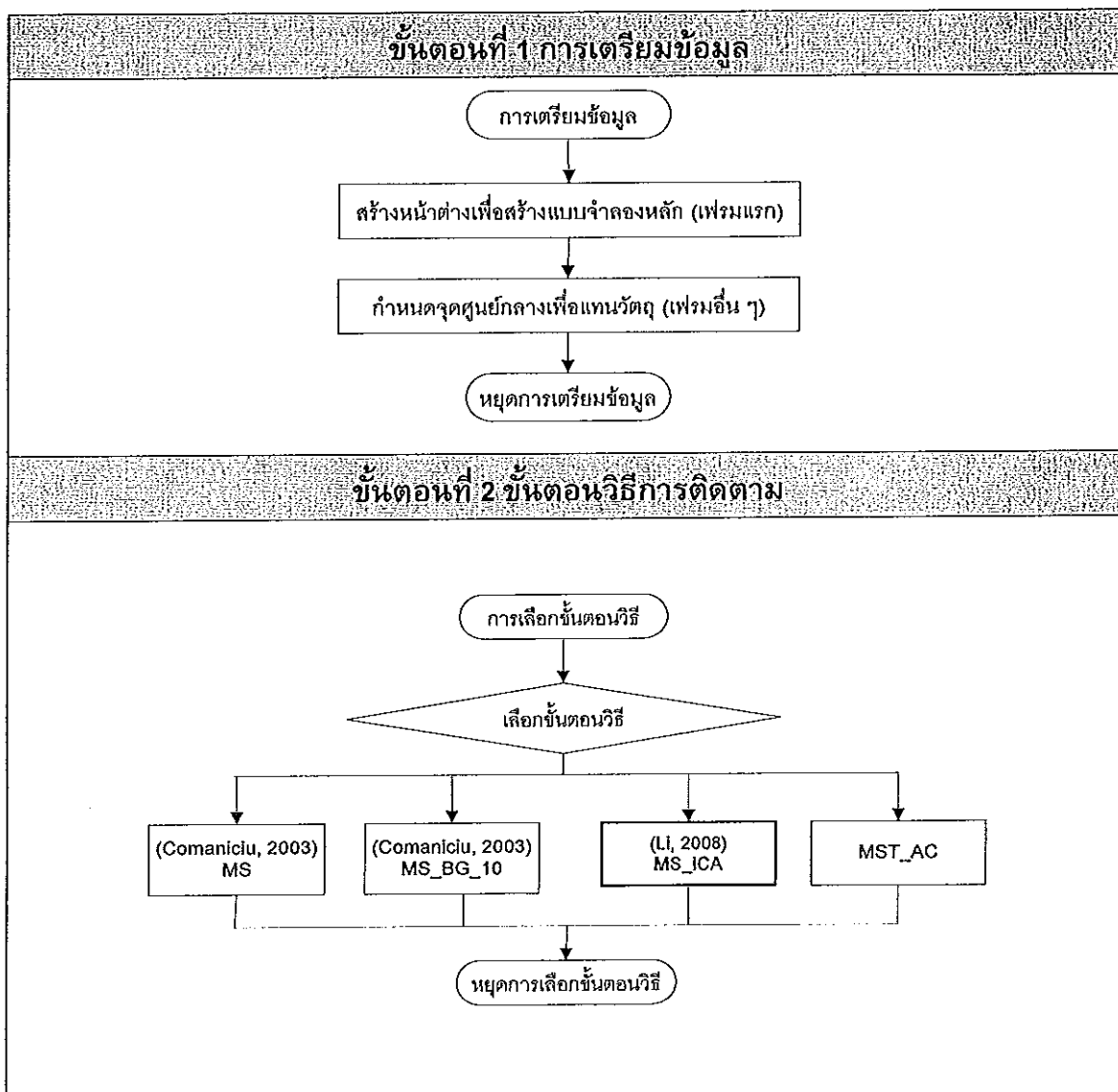
ตารางที่ 4.1 รายละเอียดชุดข้อมูลทดสอบ

ชื่อ VDO	จำนวนเฟรม	fps	ขนาดเฟรม	ขนาดหน้าต่าง	Skip	รายละเอียด
Airport 1	380	30	720x576	43x158	2	คนเดินในสนามบินในทิศทางห่างจากกล้อง สพายเป้สีดำ
Airport 2	140	30	720x576	72x71	2	คนเดินในสนามบินจากขวาไปซ้ายใส่อีเสื่อสีขาว
Airport 3	150	30	720x576	63x166	2	คนเดินในสนามบินจากขวาไปซ้ายใส่ชุดดำ
Airport 4	100	30	720x576	66x151	2	คนเดินในสนามบินจากขวาไปซ้ายใส่อีเสื่อสีดำ ทางเกงสีขาว
ComSci 1	400	30	640x480	34x61	2	คนเดินนอกสถานที่ในทิศทางห่างจากกล้อง ใส่อีเสื่อสีขาว
ComSci 2	660	30	640x480	50x94	2	คนเดินนอกสถานที่ในทิศทางห่างจากกล้อง ใส่อีเสื่อสีฟ้า
ComSci 3	600	30	640x480	53x77	2	คนวิ่งนอกสถานที่ในทิศทางห่างและใกล้จากกล้อง ใส่อีเสื่อสีขาว
ComSci 4	150	30	640x480	46x100	2	คนเดินนอกสถานที่ ใส่อีเสื่อสีดำเดินตัดพื้นหลังสีดำ

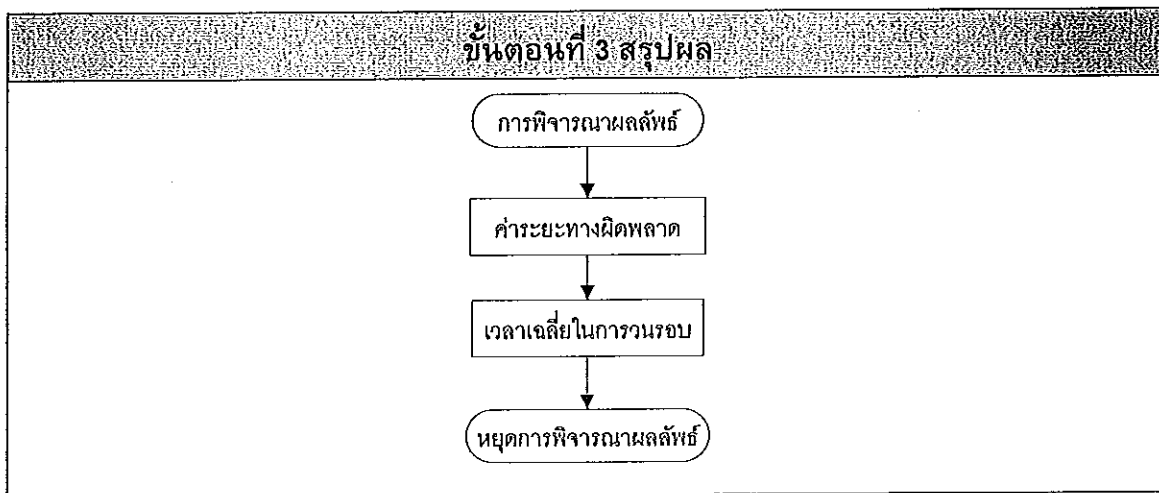
4.2 การทดลองเปรียบเทียบขั้นตอนวิธีการติดตาม

สำหรับการทดลองเปรียบเทียบขั้นตอนวิธีการติดตามจะประกอบด้วยขั้นตอนทั้งหมด 3 ขั้นตอนคือ ขั้นตอนแรกเป็นการเตรียมข้อมูลเพื่อใช้ในการทดสอบ ในการเตรียมข้อมูลทดสอบ

นั้นใช้โปรแกรมที่พัฒนาขึ้นมาดัง ภาคผนวก ก ในการเตรียมข้อมูลจะได้ตำแหน่งพื้นที่บนเฟรม ซึ่งเป็นพื้นที่ของวัตถุเป้าหมาย และจุดศูนย์กลางที่ใช้แทนวัตถุในแต่ละเฟรม ขั้นตอนที่ 2 เป็นขั้นตอนเลือกขั้นตอนวิธีที่ใช้ในการติดตาม ซึ่งมี 4 ขั้นตอนวิธีด้วยกัน คือ MS เป็นขั้นตอนวิธี Mean Shift ทัวไปที่ไม่มีการแก้ปัญหาพื้นหลัง MS_BG_10 เป็นขั้นตอนวิธี Mean Shift ทัวไปที่แก้ปัญหาพื้นหลังโดยขยายหน้าต่าง 10 % MS_ICA เป็นขั้นตอนวิธี Mean Shift ที่ใช้วิธีการ Independent Component Analysis และ MST_AC เป็นขั้นตอนวิธีที่ผู้วิจัยนำเสนอ ขั้นตอนสุดท้ายคือการสรุปผลโดยพิจารณาถึงผลลัพธ์ที่ได้จากการทดลองซึ่งแสดงค่าระยะทางผิดพลาดและเวลาที่ใช้ในการทำงาน ขั้นตอนการทดสอบทั้งหมดสามารถแสดงดังภาพประกอบ 4.1



ภาพประกอบ 4.1 ขั้นตอนการทดสอบการติดตาม



ภาพประกอบ 4.1 ขั้นตอนการทดสอบการติดตาม (ต่อ)

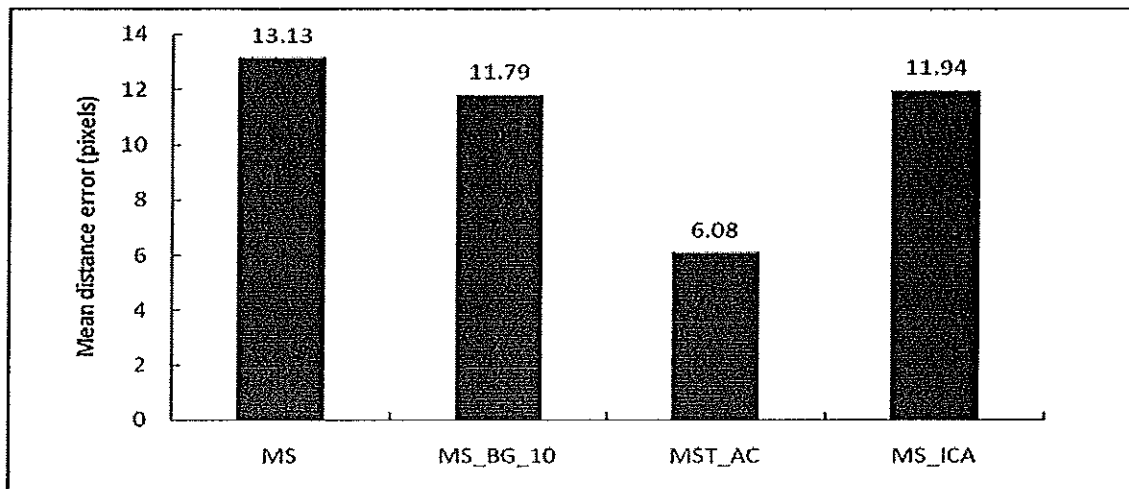
4.3 ผลการทดลอง

4.3.1 ชุดข้อมูล Airport1

เป็นชุดข้อมูลที่วัตถุประสงค์เป้าหมายเป็นคนเดินในสนามบินในทิศทางห่างจากกล้อง สพายเป็สีดำ

4.3.1.1 ค่าระยะทางผิดพลาดเฉลี่ย

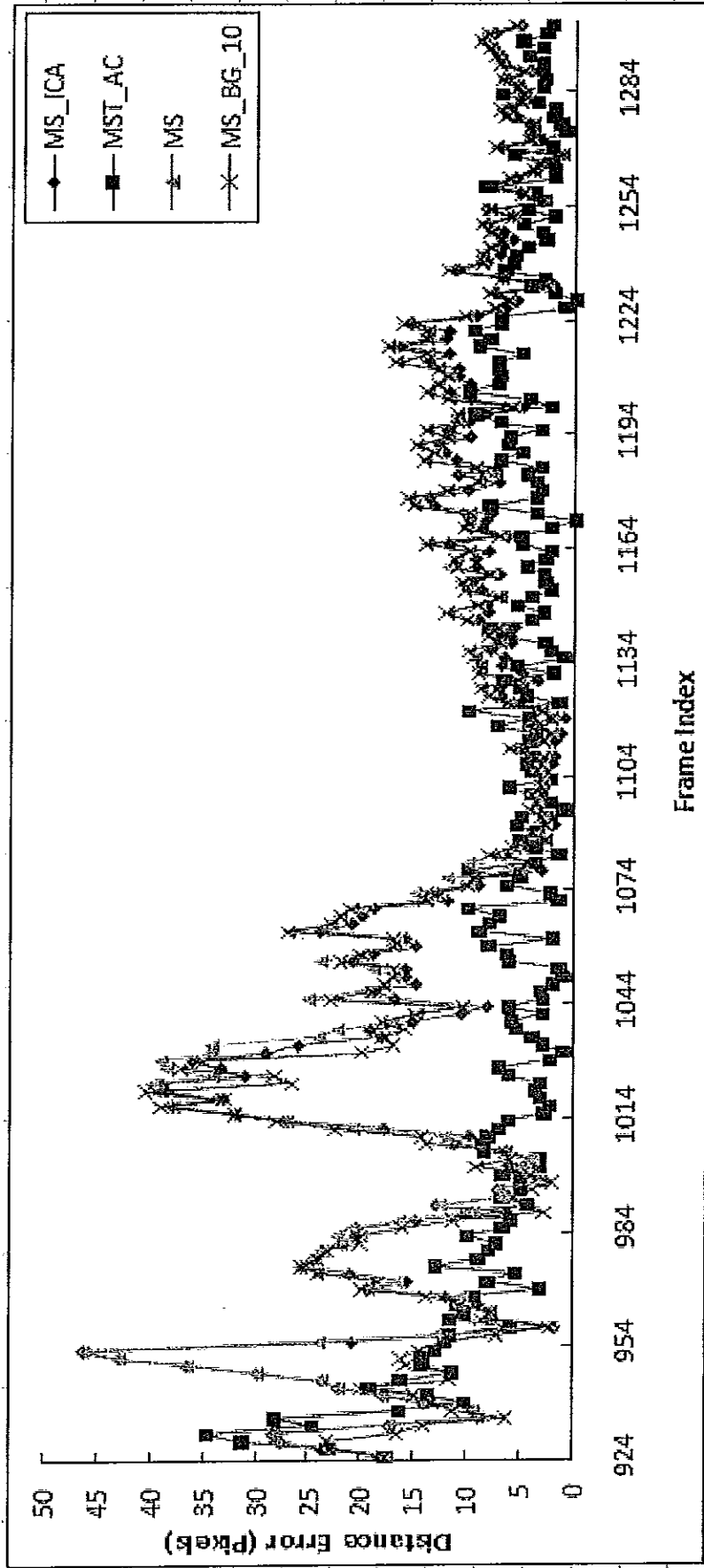
ภาพประกอบ 4.2 แสดงค่าระยะทางผิดพลาดเฉลี่ยของแต่ละ ขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport1 จะเห็นได้ว่า MST_AC ให้ค่าระยะทางผิดพลาดเฉลี่ย น้อยที่สุดคือ 6.08 ในขณะที่ MS ให้ค่า 13.13 MS_BG_10 ให้ค่า 11.79 และ MS_ICA ให้ค่า 11.94



ภาพประกอบ 4.2 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport 1

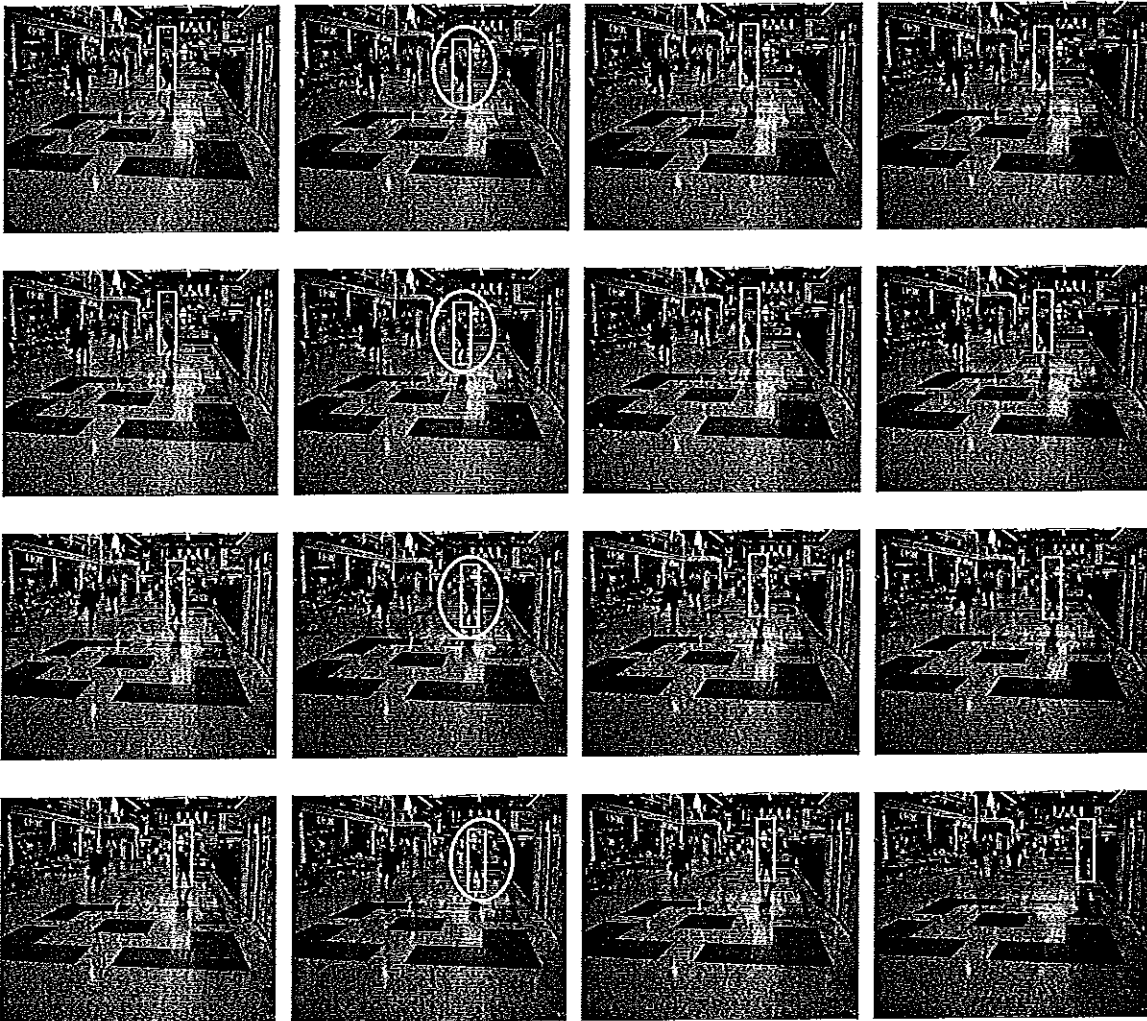
4.3.1.2 ค่าระยะทางผิดพลาด

ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport1 ดังภาพประกอบ 4.3 จะเห็นได้ว่าจุดส่วนใหญ่ของขั้นตอนวิธี MST_AC ต่ำกว่าขั้นตอนวิธีอื่น



ภาพประกอบ 4.3 ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 1

ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC ที่แสดงในภาพประกอบ 4.4 จะเห็นได้ว่าเฟรมที่ 1018 1020 1030 และเฟรมที่ 1040 สำหรับขั้นตอนวิธีของ MST_AC ให้ความถูกต้องในการติดตามมากกว่าขั้นตอนวิธีอื่น สังเกตได้จากวงกลมด้านล่าง เนื่องจากในการทำงานของขั้นตอนวิธี MST_AC มีการแก้ปัญหาพื้นหลังสังเกตได้จากภายในกรอบสี่เหลี่ยมเมื่อภาพวัตถุเล็กลงส่วนของพื้นหลังแสดงออกมา แต่ในขั้นตอนวิธีอื่น ๆ กรอบสี่เหลี่ยมจะไม่พอดีกับวัตถุ



MS_ICA

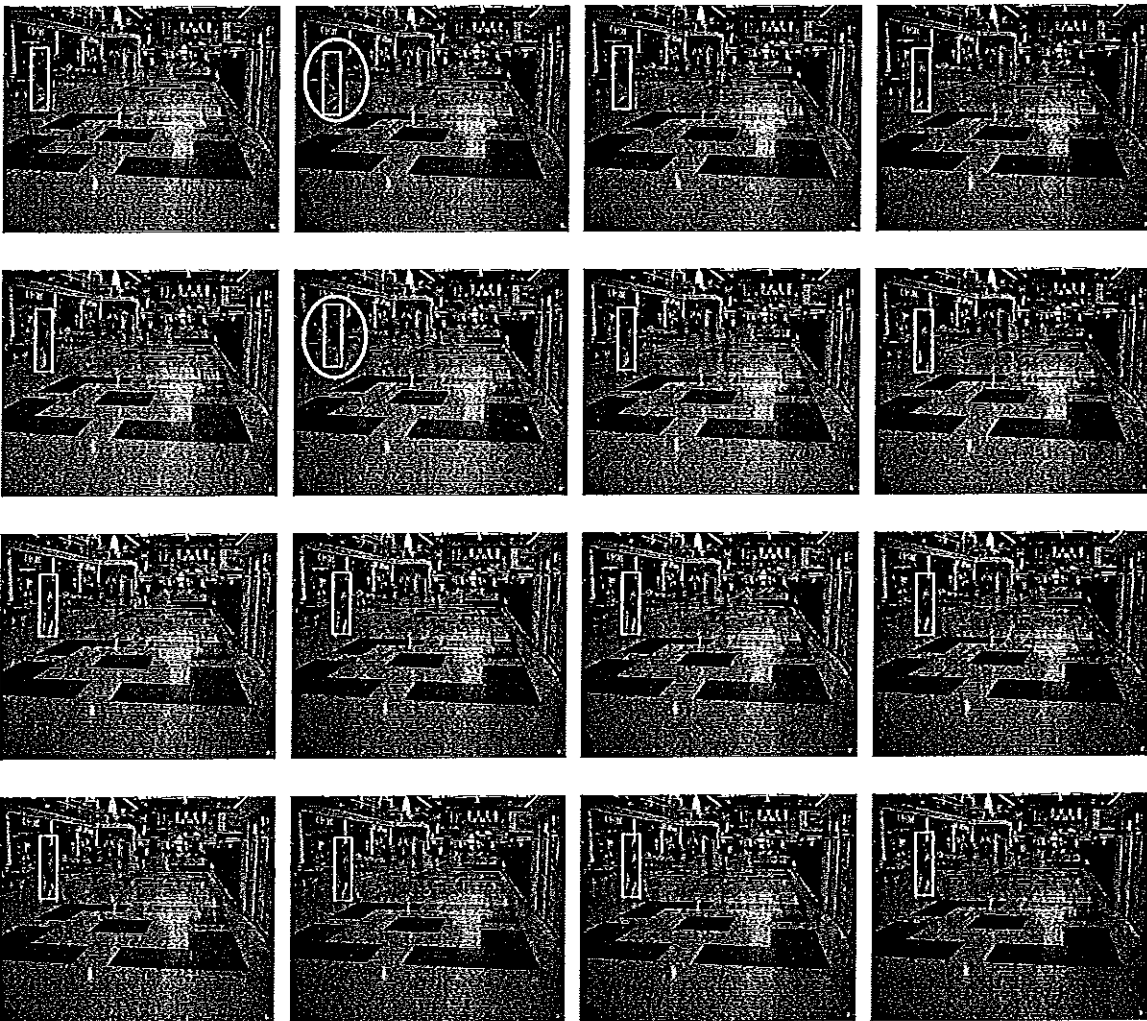
MST_AC

MS

MS_BG_10

ภาพประกอบ 4.4 ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC ลำดับของเฟรม 1018 1020 1030 1040 ตามแนวตั้งจากบนลงล่างโดยใช้ข้อมูลทดสอบ Airport 1

ตัวอย่างการติดตามที่ยังเป็นปัญหาของ MST_AC แสดงดังภาพประกอบ 4.5 จะเห็นว่าในเฟรมที่ 928 และเฟรมที่ 930 สำหรับขั้นตอนวิธีของ MST_AC ให้ความถูกต้องในการติดตามน้อยกว่าขั้นตอนวิธีอื่นสังเกตได้จากวงกลมด้านล่าง สาเหตุเนื่องจากว่าในการสร้างแบบจำลองหลักในเฟรมแรกนั้นลักษณะของคนสพายเป้จะมีส่วนของพื้นหลังอยู่มาก ทำให้เข้าใจผิด หลังจากนั้นในเฟรมที่ 932 และเฟรมที่ 934 สำหรับขั้นตอนวิธีของ MST_AC ให้ค่าระยะทางผิดพลาดใกล้เคียงกับวิธีอื่น



MS_ICA

MST_AC

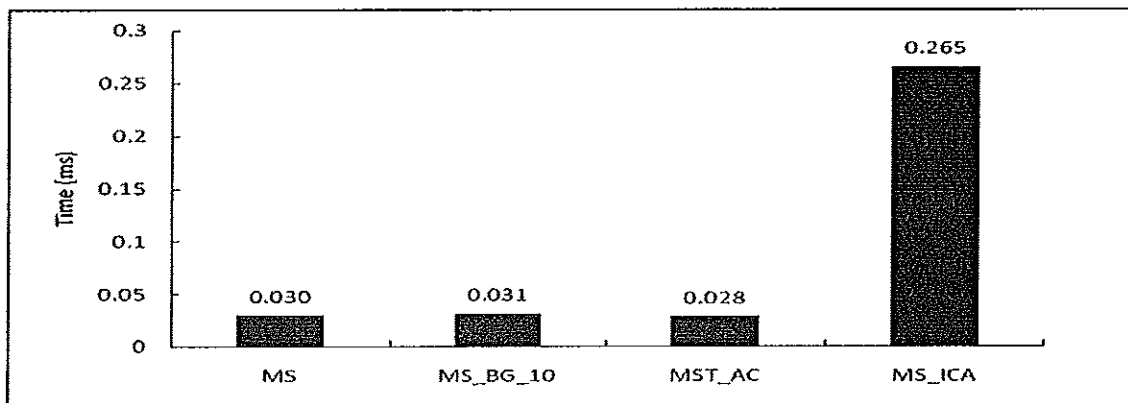
MS

MS_BG_10

ภาพประกอบ 4.5 ตัวอย่างการติดตามที่ยังมีปัญหของขั้นตอนวิธี MST_AC ลำดับของเฟรม 928 930 932 934 ตามแนวตั้งจากบนลงล่าง โดยใช้ข้อมูลทดสอบ Airport 1

4.3.1.3 ค่าเวลาเฉลี่ย

ค่าเวลาเฉลี่ยในการทำงาน 1 รอบ ของชุดข้อมูล Airport1 ดังภาพประกอบ 4.6 จะเห็นได้ว่าขั้นตอนวิธี MST_AC ใช้เวลาในการวนรอบ 1 ครั้งใช้เวลาน้อยที่สุดคือ 0.028 ms และมีค่าใกล้เคียงกับขั้นตอนวิธี MS ให้ค่า 0.030 ms MS_BG_10 ให้ค่า 0.031 และสำหรับขั้นตอนวิธี MS_ICA ใช้เวลาในการวนรอบมากที่สุดคือ 0.265 ms



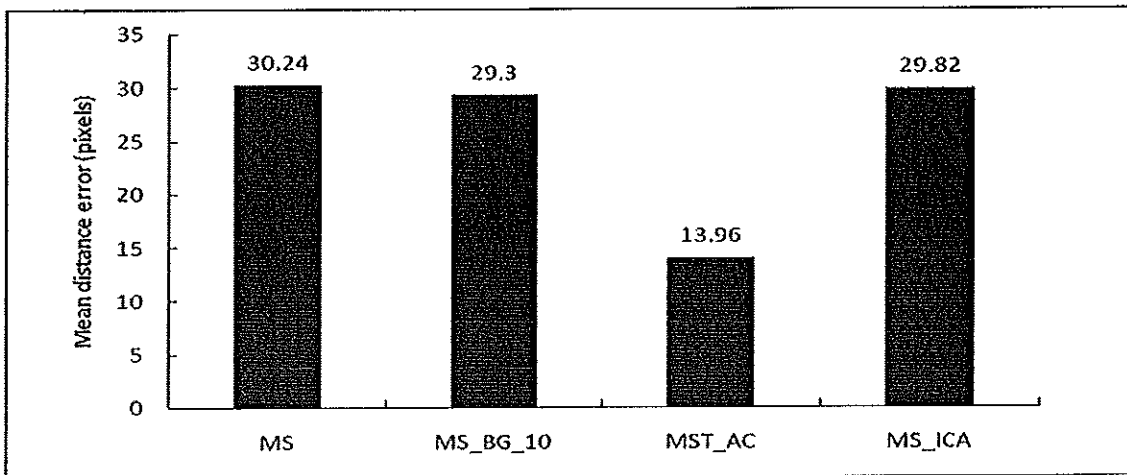
ภาพประกอบ 4.6 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธีโดยใช้ชุดข้อมูล Airport 1

4.3.2 ชุดข้อมูล Airport 2

เป็นชุดข้อมูลที่วัตถุประสงค์เป้าหมายเป็นคนเดินในสนามบินจากขวาไปซ้ายใส่เสื้อสีขาว

4.3.2.1 ค่าระยะทางผิดพลาดเฉลี่ย

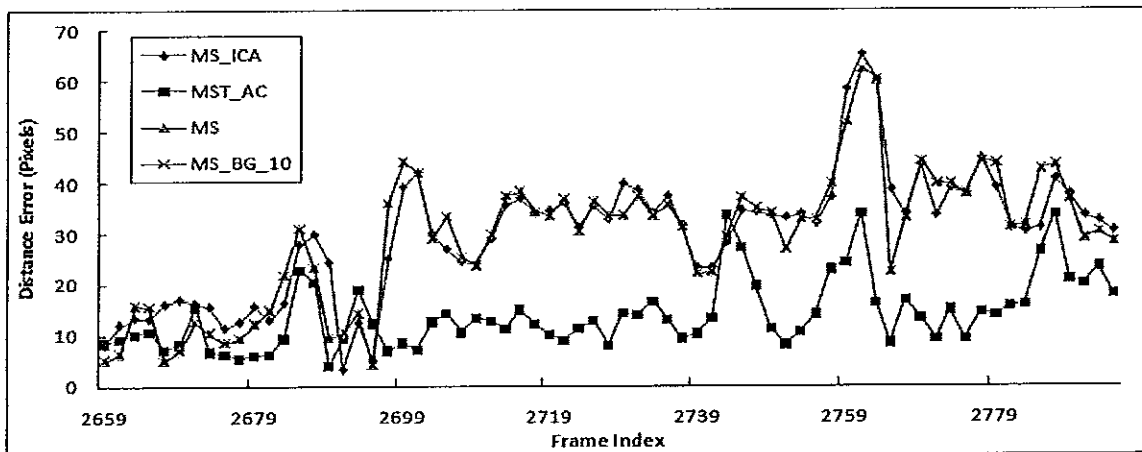
ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport 2 ดังภาพประกอบ 4.7 จะเห็นได้ว่า MST_AC ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยที่สุดคือ 13.96 โดยที่ MS ให้ค่า 30.24 MS_BG_10 ให้ค่า 29.3 และ MS_ICA ให้ค่า 29.82



ภาพประกอบ 4.7 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 2

4.3.2.2 ค่าระยะทางผิดพลาด

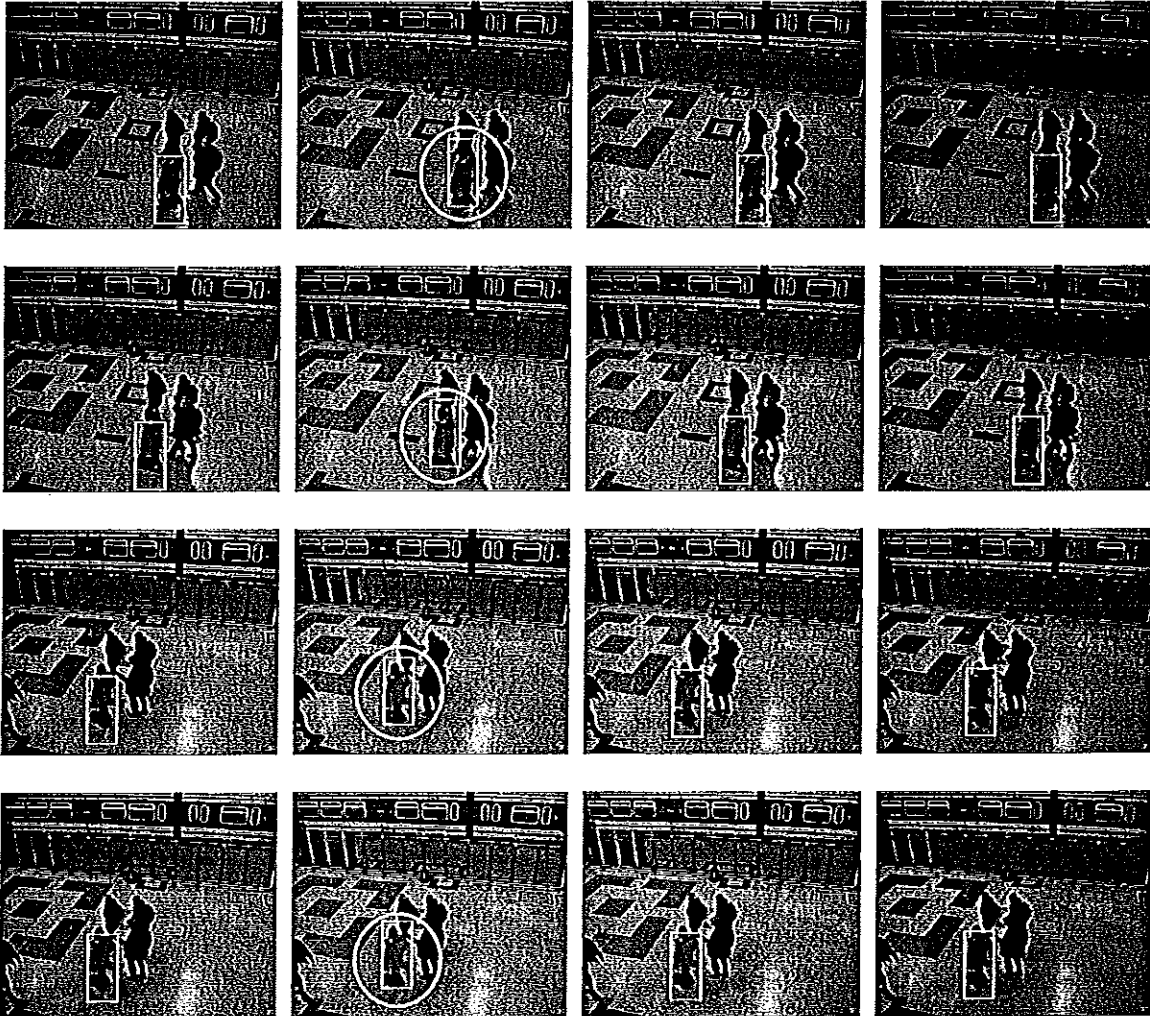
ค่าระยะทางผิดพลาด ของชุดข้อมูล Airport2 ดังภาพประกอบ 4.8 จะเห็นได้ว่าจุดส่วนใหญ่ของ MST_AC ต่ำกว่าขั้นตอนวิธีอื่น



ภาพประกอบ 4.8 ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 2

ตัวอย่างการติดตามที่ดีของ MST_AC แสดงดังภาพประกอบ 4.9 จะเห็นได้ว่าในเฟรมที่ 2700 2710 2719 และ 2735 สำหรับขั้นตอนวิธีของ MST_AC ให้ความถูกต้องในการติดตาม

มากกว่าขั้นตอนวิธีอื่นๆ จากวงกลมด้านล่าง ซึ่งสามารถสังเกตได้จากกรอบสี่เหลี่ยมที่อยู่ใกล้กับวัตถุเป้าหมายมากที่สุด



MS_ICA

MST_AC

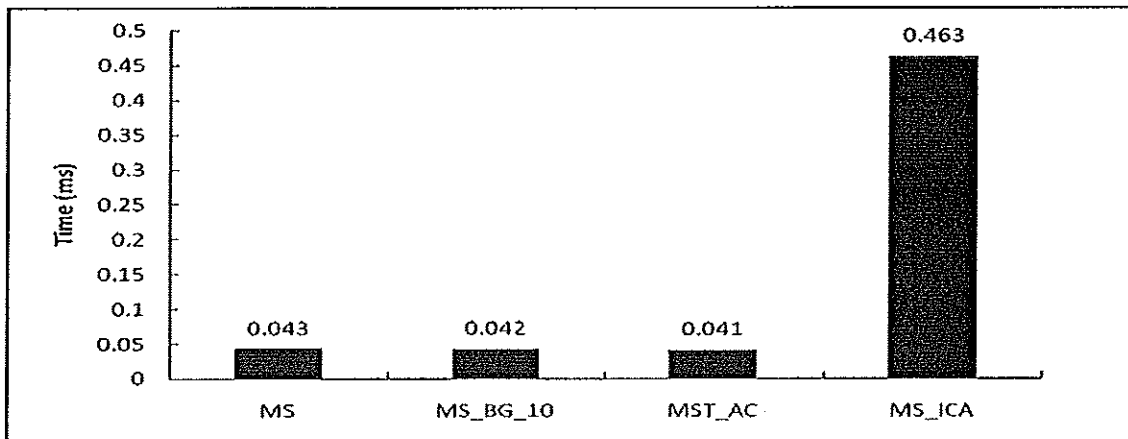
MS

MS_BG_10

ภาพประกอบ 4.9 ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC ลำดับเฟรม 2700 2710 2719 และ 2735 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล Airport 2

4.3.2.3 ค่าเวลาเฉลี่ย

ค่าเวลาเฉลี่ยในการทำงาน 1 รอบ ของชุดข้อมูล Airport2 ดังภาพประกอบ 4.10 จะเห็นได้ว่าขั้นตอนวิธี MST_AC ใช้เวลาในการวนรอบ 1 ครั้งใช้เวลาน้อยที่สุดคือ 0.041 ms และมีค่าใกล้เคียงกับขั้นตอนวิธี MS คือ 0.043 ms MS_BG_10 คือ 0.042 ms และสำหรับขั้นตอนวิธี MS_ICA ใช้เวลาในการวนรอบมากที่สุดคือ 0.463 ms



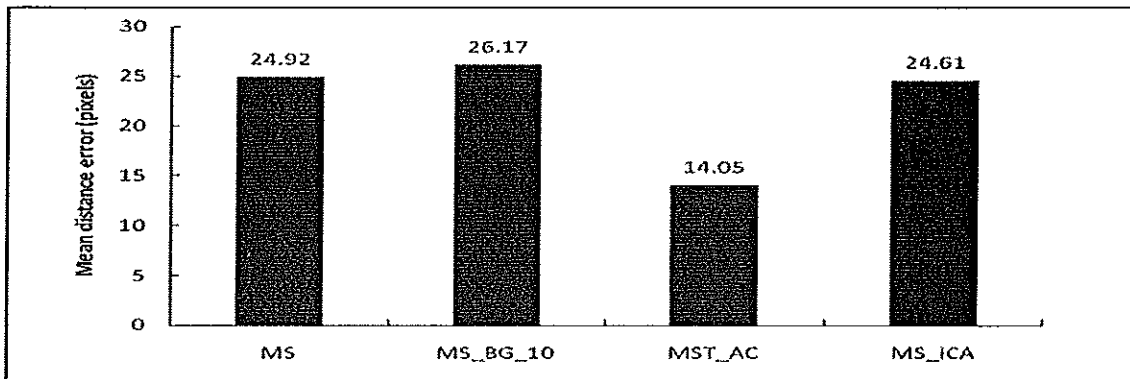
ภาพประกอบ 4.10 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล Airport 2

4.3.3 ชุดข้อมูล Airport 3

เป็นชุดข้อมูลที่มีวัตถุประสงค์เป้าหมายคือคนเดินในสนามบินจากขวาไปซ้ายใส่ชุดดำ

4.3.3.1 ค่าระยะทางผิดพลาดเฉลี่ย

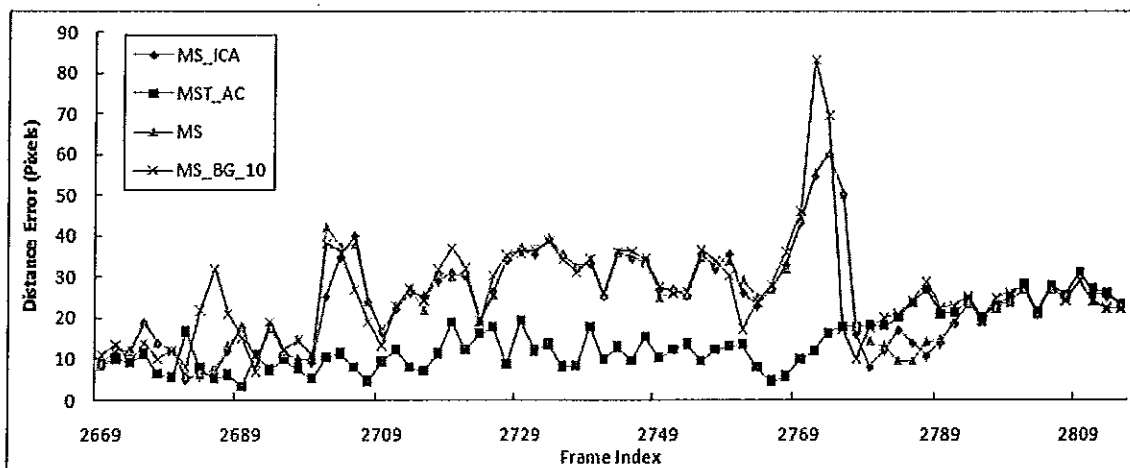
ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport 3 ดังภาพประกอบ 4.11 จะเห็นได้ว่า MST_AC ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยที่สุดคือ 14.05 ในขณะที่ MS ให้ค่า 24.92 MS_BG_10 ให้ค่า 26.17 และ MS_ICA ให้ค่า 24.61



ภาพประกอบ 4.11 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 3

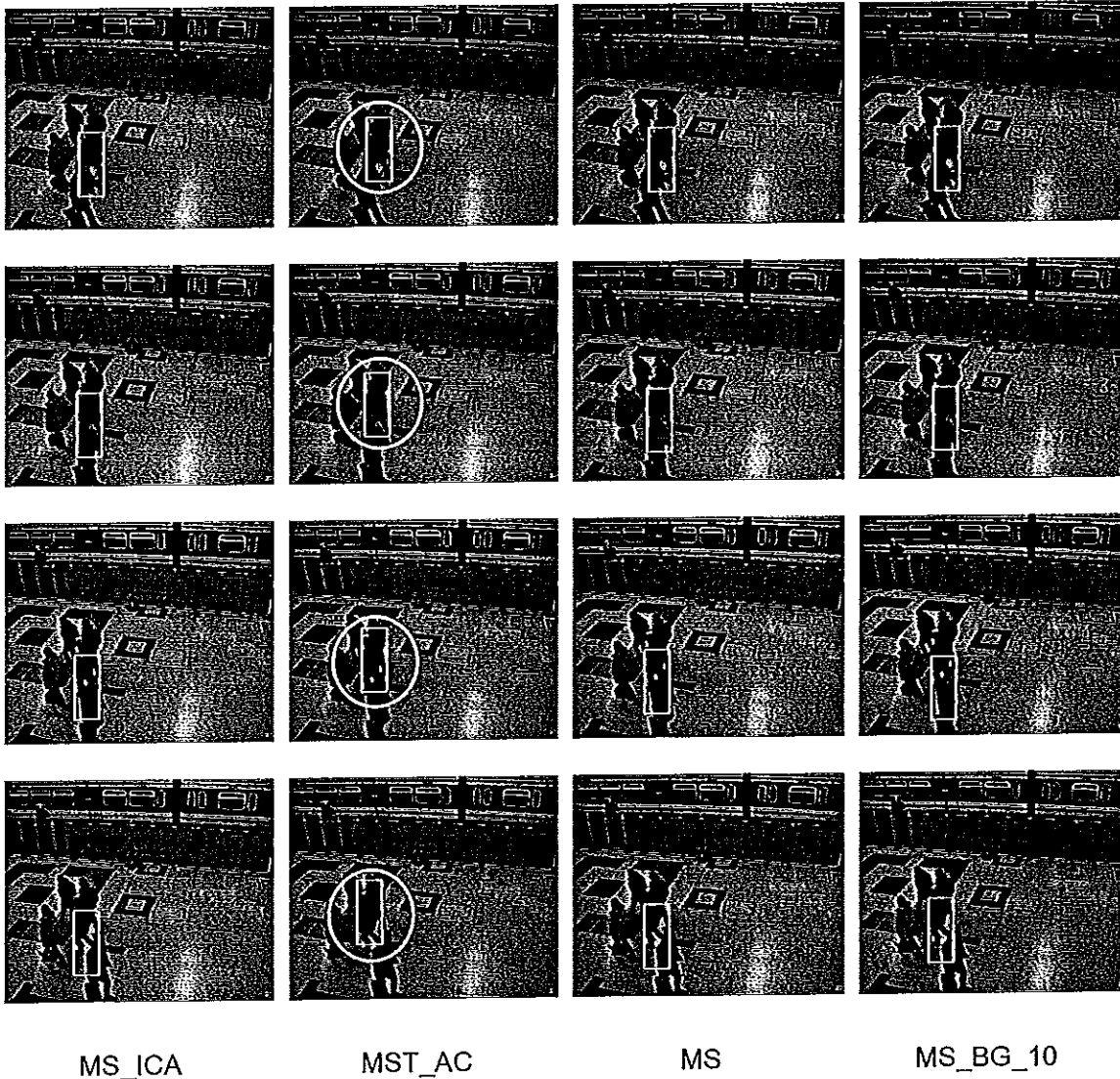
4.3.3.2 ค่าระยะทางผิดพลาด

ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport 3 ดังภาพประกอบ 4.12 จะเห็นได้ว่าจุดส่วนใหญ่ของขั้นตอนวิธี MST_AC ต่ำกว่าขั้นตอนวิธีอื่น



ภาพประกอบ 4.12 ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 3

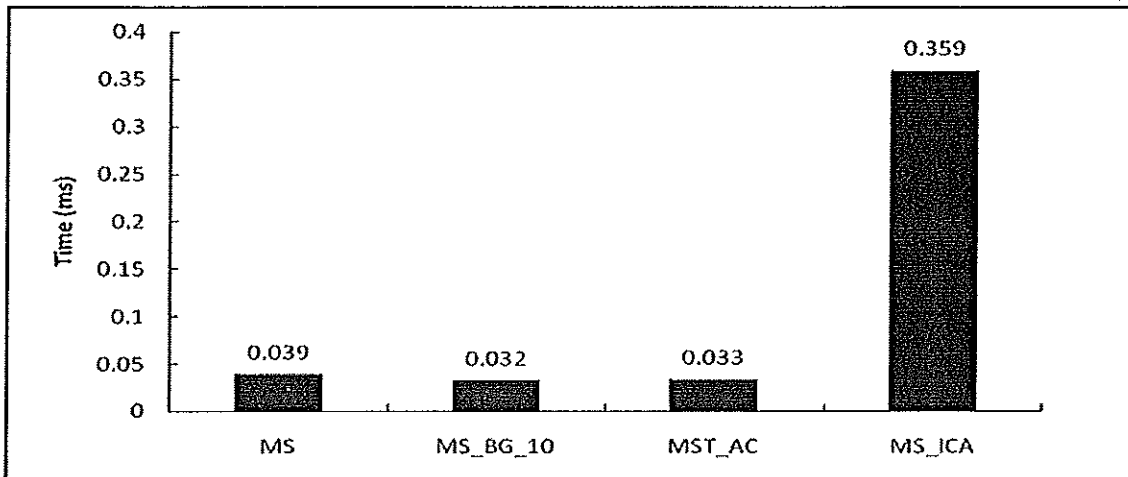
ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC แสดงดังภาพประกอบ 4.13 ซึ่งจะเห็นได้ว่าในเฟรมที่ 2765 2767 2769 และเฟรมที่ 2771 วัตถุเป้าหมาย (คนชุดดำ) ถูกบดบังด้วยอีกวัตถุ (คนชุดดำ) ซึ่งมีสีเดียวกับวัตถุเป้าหมาย ทำให้ผลของการทดสอบของขั้นตอนวิธี MST_AC ให้ความถูกต้องในการติดตามมากกว่าขั้นตอนอื่น ดูจากวงกลมด้านล่าง โดยสามารถสังเกตได้จากกรอบสี่เหลี่ยมที่อยู่ใกล้กับวัตถุเป้าหมายมากที่สุด



ภาพประกอบ 4.13 ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC ลำดับเฟรม 2765 2767 2769 และ 2771 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล Airport 3

4.3.3.3 ค่าเวลาเฉลี่ย

ค่าเวลาเฉลี่ยในการทำงาน 1 รอบ ของชุดข้อมูล Airport3 ดังภาพประกอบ 4.14 จะเห็นได้ว่าขั้นตอนวิธี MS_BG_10 ใช้เวลาเฉลี่ยในการวนรอบ 1 ครั้งใช้เวลาน้อยที่สุดคือ 0.032 ms และมีค่าใกล้เคียงกับขั้นตอนวิธี MS คือ 0.039 ms MST_AC คือ 0.033 ms และสำหรับขั้นตอนวิธี MS_ICA ใช้เวลาในการวนรอบมากที่สุดคือ 0.359 ms



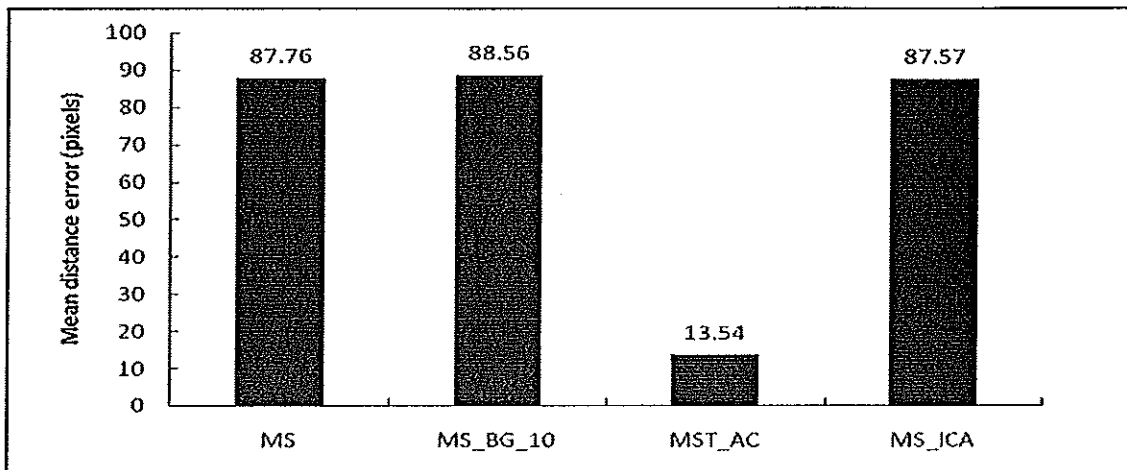
ภาพประกอบ 4.14 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล Airport 3

4.3.4 ชุดข้อมูล Airport 4

เป็นชุดข้อมูลที่วัตถุประสงค์เป้าหมายเป็นคนเดินในสนามบินจากขวาไปซ้าย ใส่เสื้อสีดำ กางเกงสีขาว

4.3.4.1 ค่าระยะทางผิดพลาดเฉลี่ย

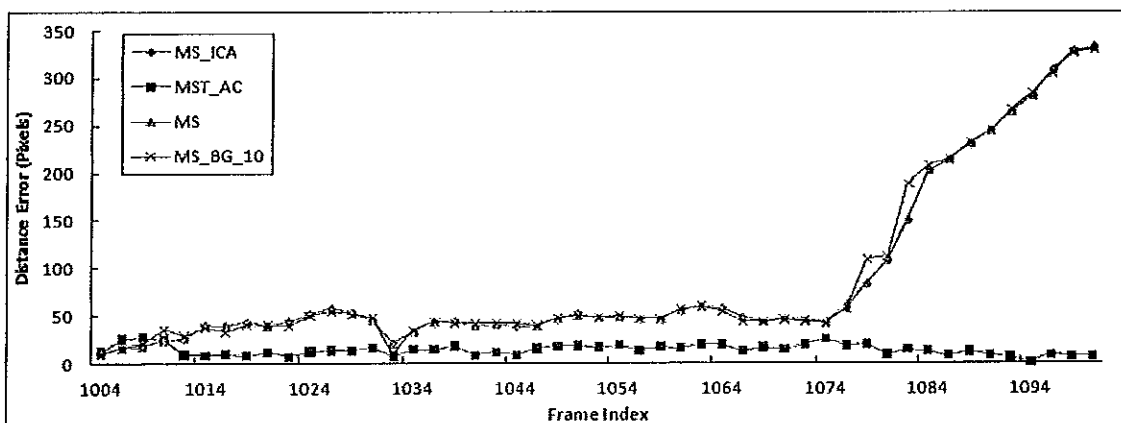
ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport 4 ดังภาพประกอบ 4.15 จะเห็นได้ว่าขั้นตอนวิธี MST_AC ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยที่สุด คือ 13.54 และขั้นตอนวิธี MS_BG_10 ให้ค่าระยะทางผิดพลาดเฉลี่ยมากที่สุดคือ 88.56 ซึ่งใกล้เคียงกับขั้นตอนวิธี MS คือ 87.76 และขั้นตอนวิธี MS_ICA คือ 87.57 จะเห็นได้ว่าแตกต่างจากค่าที่ได้จากขั้นตอนวิธี MST_AC มาก อันเนื่องจากการติดตามที่ผิดพลาด



ภาพประกอบ 4.15 ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล Airport 4

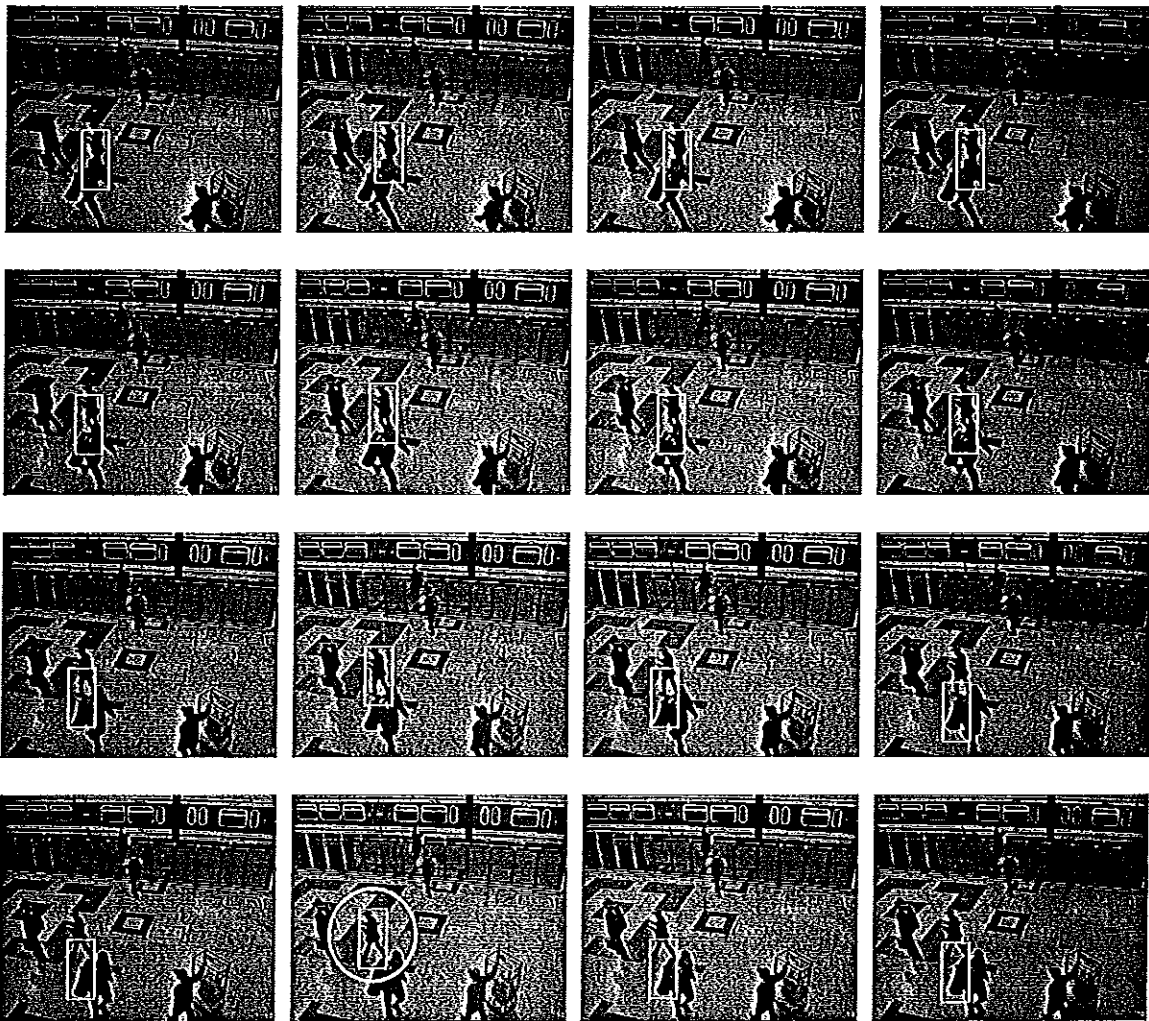
4.3.4.2 ค่าระยะทางผิดพลาด

ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport 4 ดังภาพประกอบ 4.16 จะเห็นว่าจุดส่วนใหญ่ของ MST_AC ต่ำกว่าขั้นตอนวิธีอื่น



ภาพประกอบ 4.16 ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 4

ตัวอย่างการติดตามที่ดีของ MST_AC จากภาพประกอบ 4.17 ในช่วงเฟรมที่ 1070 1072 1074 และเฟรมที่ 1076 จะพบว่าวัตถุเป้าหมายถูกบังโดยอีกวัตถุหนึ่ง ซึ่งมีบางส่วนที่มีสีเดียวกับวัตถุเป้าหมาย (สีกระเป๋าส) และในเฟรมที่ 1076 MST_AC ยังสามารถติดตามวัตถุจากวงกลมด้านล่าง ในขณะที่การติดตามโดยวิธีอื่น ไม่สามารถติดตามวัตถุ



MS_ICA

MST_AC

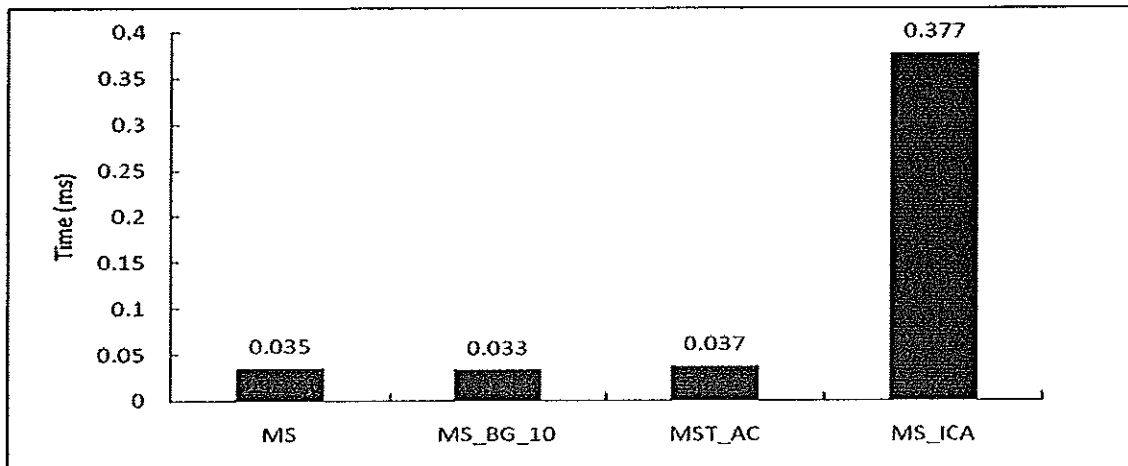
MS

MS_BG_10

ภาพประกอบ 4.17 ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC ลำดับเฟรม 1070 1072 1074 และ 1076 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล Airport 4

4.3.4.3 ค่าเวลาเฉลี่ย

ค่าเวลาเฉลี่ยในการทำงาน 1 รอบ ของชุดข้อมูล Airport4 ดังภาพประกอบ 4.18 จะเห็นได้ว่าขั้นตอนวิธี MS_BG_10 ใช้เวลาเฉลี่ยน้อยที่สุดคือ 0.033 ms และใกล้เคียงกับขั้นตอนวิธี MS คือ 0.035 ms และขั้นตอนวิธี MST_AC คือ 0.037 ms แต่สำหรับขั้นตอนวิธี MS_ICA นั้นใช้เวลาเฉลี่ยในการวนรอบมากที่สุดคือ 0.377 ms



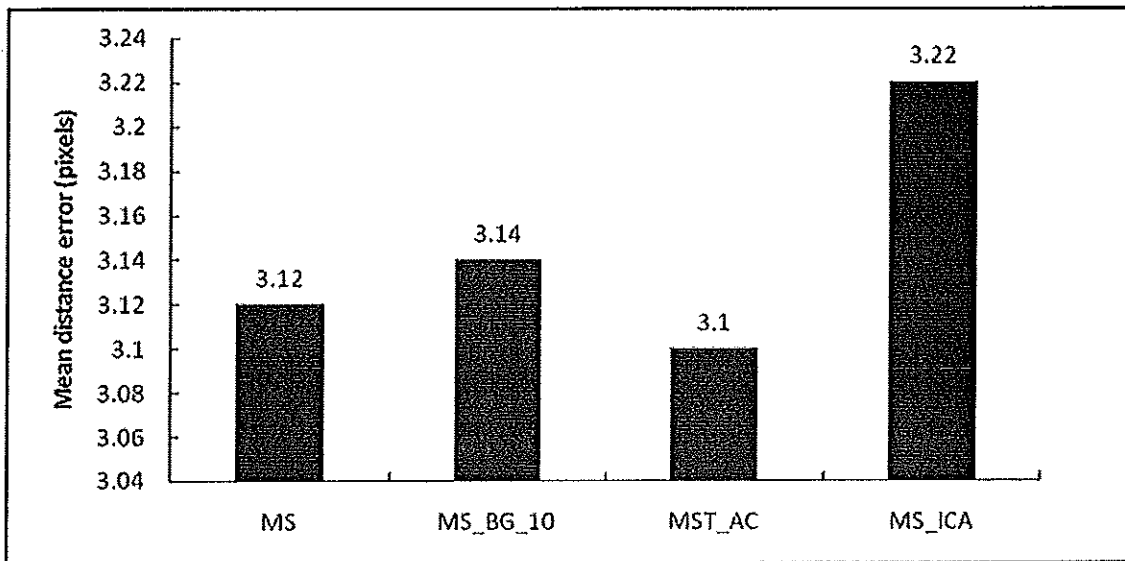
ภาพประกอบ 4.18 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล Airport 4

4.3.5 ชุดข้อมูล ComSci 1

เป็นชุดข้อมูลที่วัตถุประสงค์เป้าหมายเป็นคนเดินนอกสถานที่ในทิศทางห่างจากกล้องใส่อีเสื้อสีขาว

4.3.5.3 ค่าระยะทางผิดพลาดเฉลี่ย

ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ ComSci 1 ดังภาพประกอบ 4.19 จะเห็นได้ว่าขั้นตอนวิธี MST_AC ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยที่สุด คือ 3.1 ในขณะที่ MS ให้ค่า 3.12 MS_BG_10 ให้ค่า 3.14 และ MS_ICA ให้ค่า 3.22

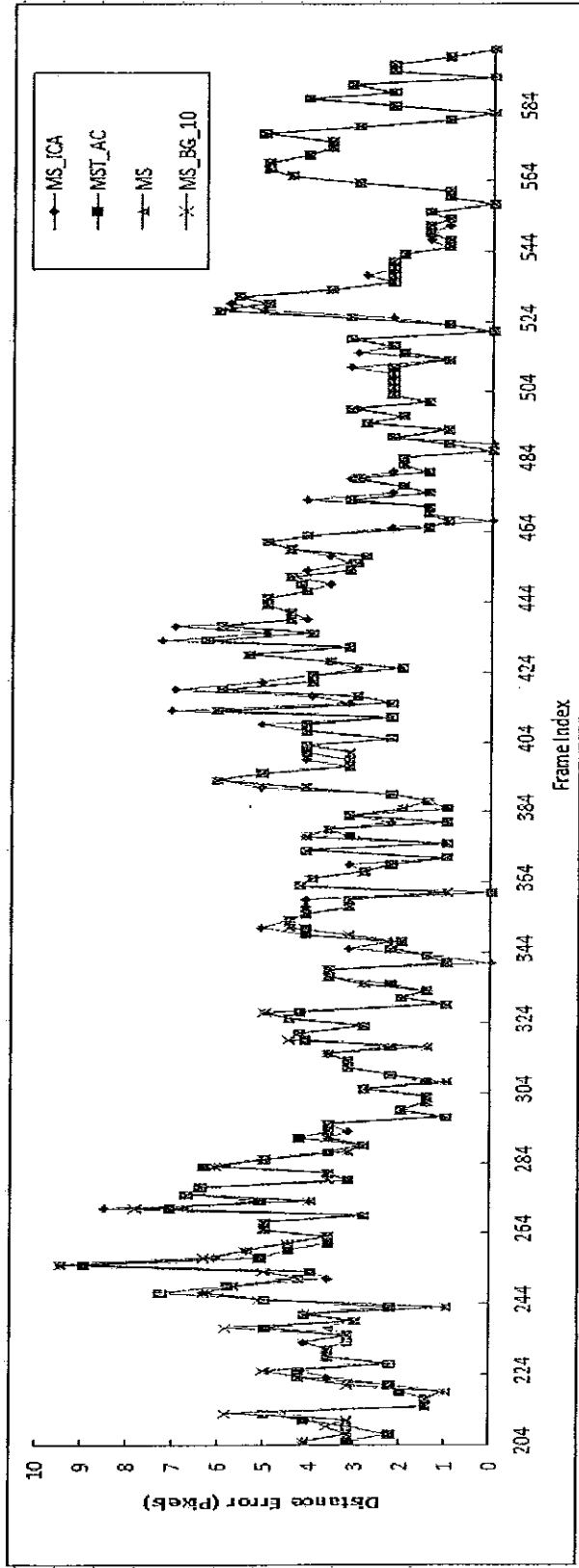


ภาพประกอบ 4.19 ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ ComSci 1

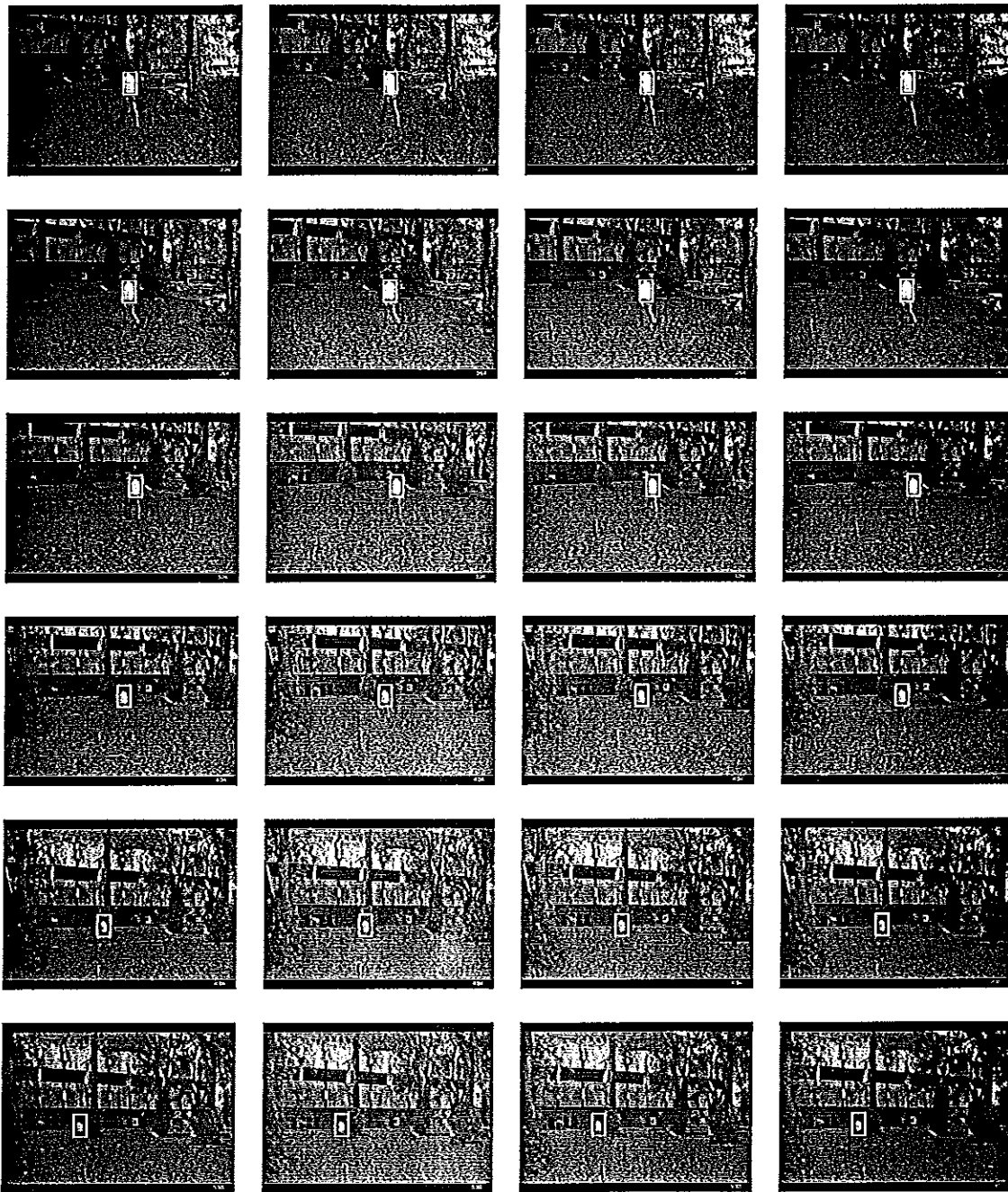
4.3.5.2 ค่าระยะทางผิดพลาด

ค่าระยะทางผิดพลาดของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ ComSci 1 ดังภาพประกอบ 4.20 จะเห็นว่าผลจากการติดตามในแต่ละขั้นตอนวิธีใกล้เคียงกัน เนื่องจากวัตถุที่ใช้สร้างแบบจำลองมีสีที่แตกต่างจากสีพื้นหลังอย่างชัดเจน

ตัวอย่างการติดตามของขั้นตอนวิธีดังภาพประกอบ 4.21 จะเห็นได้ว่าผลจากการติดตามในแต่ละขั้นตอนวิธีใกล้เคียงกัน เนื่องจากวัตถุที่ใช้สร้างแบบจำลองมีสีที่แตกต่างจากพื้นหลังโดยสิ้นเชิง วัตถุที่ใช้สร้างแบบจำลองมีสีขาว



ภาพประกอบ 4.20 ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ ComSci 1



MS_ICA

MST_AC

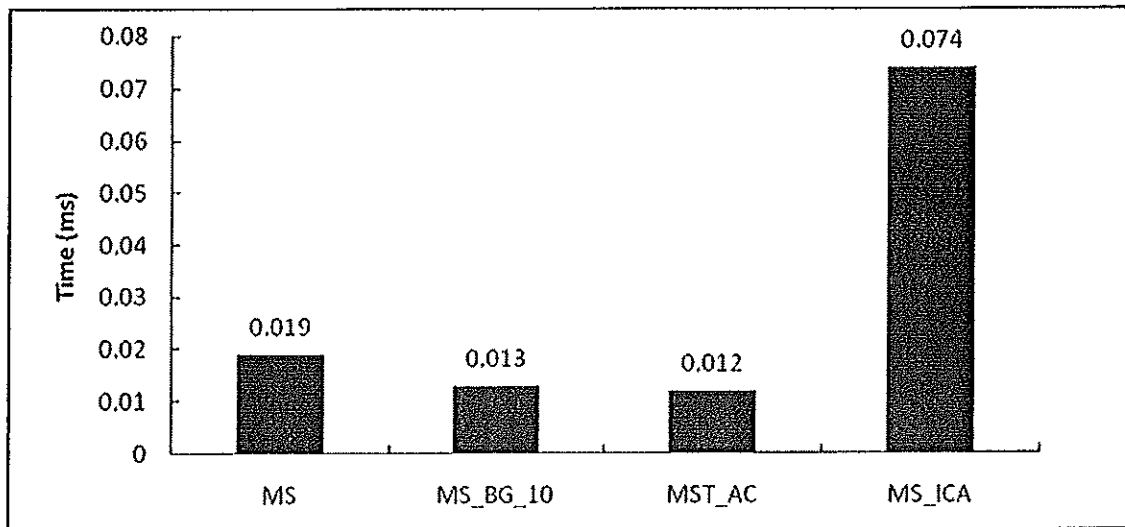
MS

MS_BG_10

ภาพประกอบ 4.21 ตัวอย่างการติดตามของขั้นตอนวิธี ลำดับเฟรม 224 264 324 404 484 และ 530 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล ComSci 1

4.3.5.3 ค่าเวลาเฉลี่ย

ค่าเวลาเฉลี่ยในการทำงาน 1 รอบ ของชุดข้อมูล ComSci 1 ดังภาพประกอบ 4.22 จะเห็นได้ว่าขั้นตอนวิธี MST_AC ใช้เวลาในการวนรอบ 1 ครั้งน้อยที่สุด คือ 0.012 ms และใกล้เคียงกับขั้นตอนวิธี MS ให้ค่า 0.019 ms และขั้นตอนวิธี MS_BG_10 ให้ค่า 0.013 ms แต่สำหรับขั้นตอนวิธี MST_ICA นั้นใช้เวลาในการวนรอบมากที่สุดคือ 0.074 ms



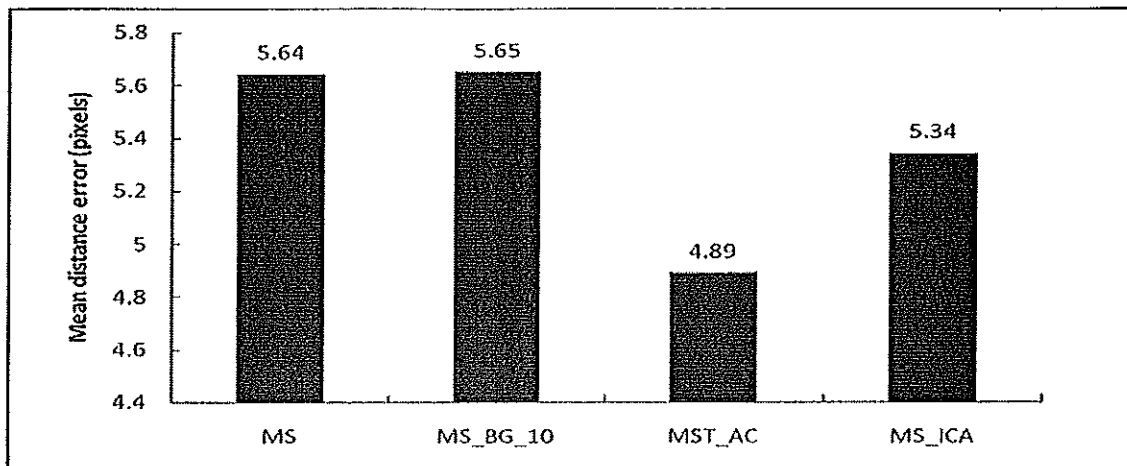
ภาพประกอบ 4.22 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล ComSci 1

4.3.6 ชุดข้อมูล ComSci 2

เป็นชุดข้อมูลที่วัตถุเป้าหมายเป็นคนเดินนอกสถานที่ในทิศทางห่างจากกล้องใส่เสื้อสีฟ้า

4.3.6.1 ค่าระยะทางผิดพลาดเฉลี่ย

ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธีโดยใช้ชุดทดสอบ ComSci 2 ดังภาพประกอบ 4.23 จะเห็นได้ว่าขั้นตอนวิธี MST_AC ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยที่สุด คือ 4.89 ในขณะที่ MS ให้ค่า 5.64 MS_BG_10 ให้ค่า 5.65 MS_ICA ให้ค่า 5.34 แต่อย่างไรก็ตามค่าความแตกต่างของระยะทางผิดพลาดไม่แตกต่างกันมากเนื่องจากสีของวัตถุเป้าหมายกับสีของพื้นหลังต่างกันมาก

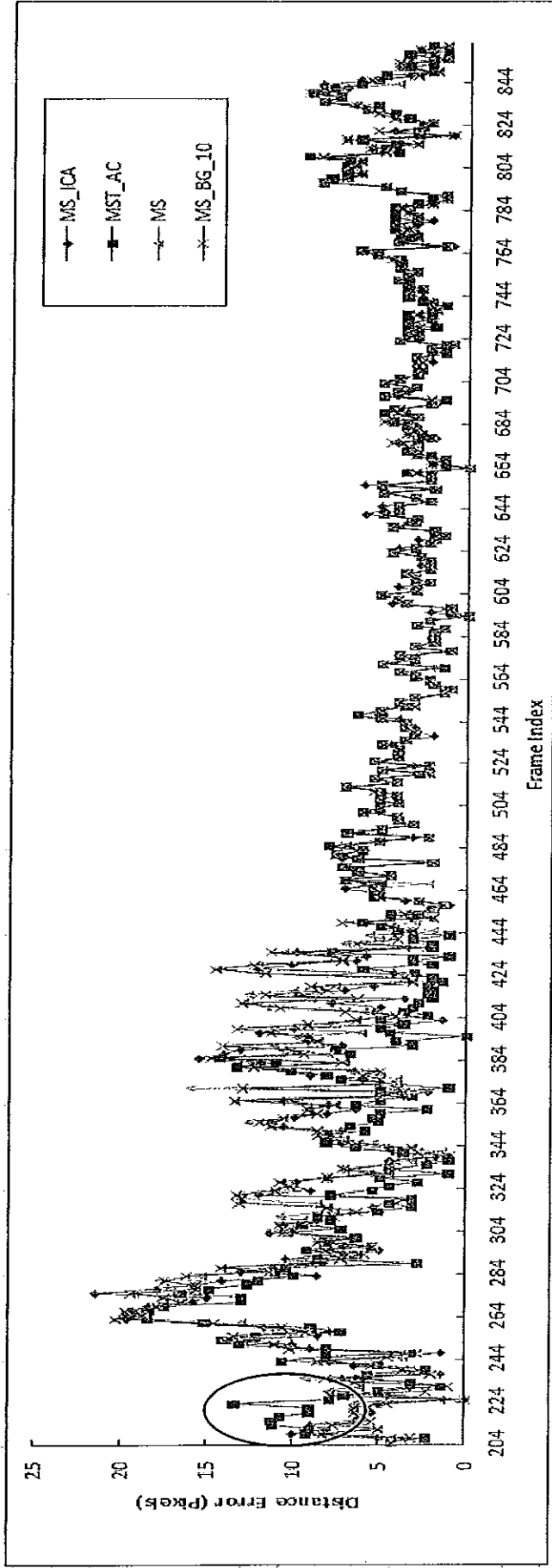


ภาพประกอบ 4.23 ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล ComSci 2

4.3.6.2 ค่าระยะทางผิดพลาด

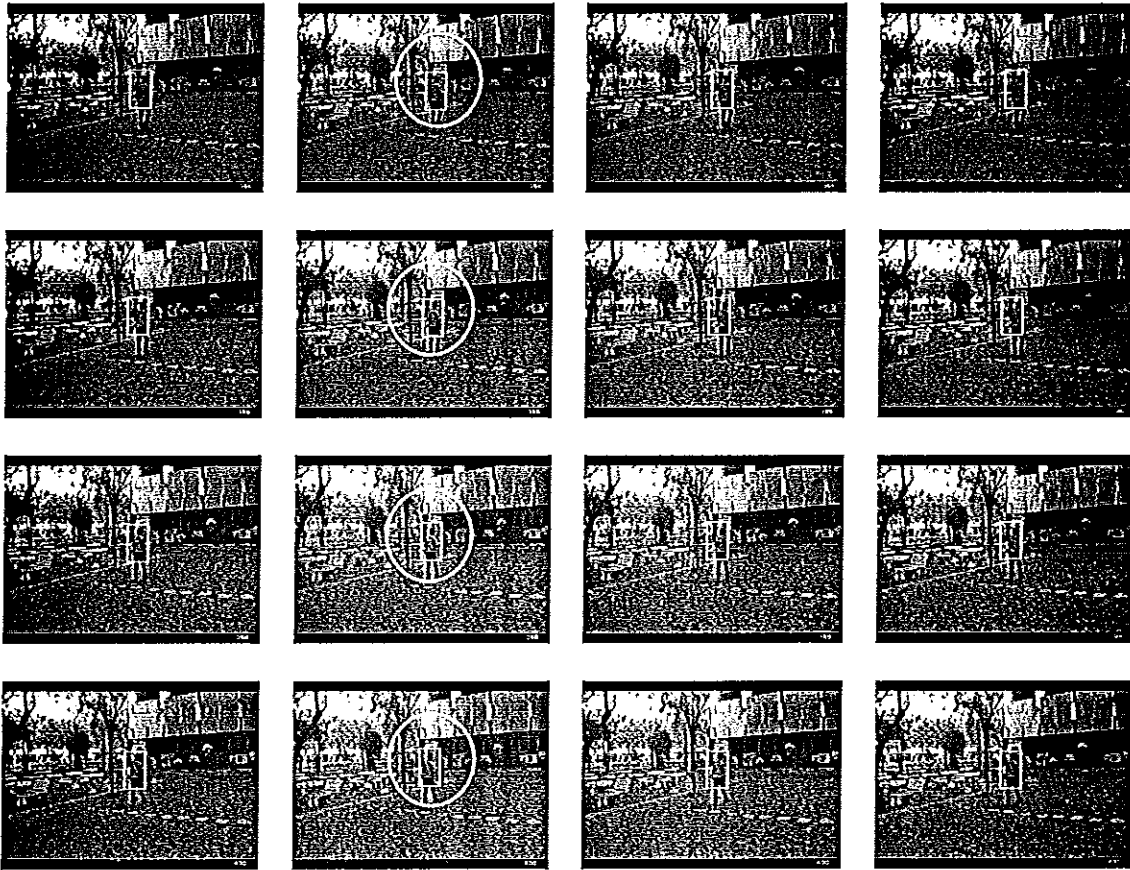
ค่าระยะทางผิดพลาดของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ

ComSci 2 ดังภาพประกอบ 4.24 จะเห็นได้ว่าในช่วงเฟรม 204 ถึง 444 กราฟของ MST_AC ส่วนใหญ่จะต่ำกว่าขั้นตอนวิธีอื่น แต่หลังจากเฟรม 444 เป็นต้นไปกราฟของขั้นตอนวิธีทุกขั้นตอนวิธีใกล้เคียงกัน



ภาพประกอบ 4.24 ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ ComSci 2

ตัวอย่างการติดตามที่ดีของ MST_AC จากภาพประกอบ 4.25 ในเฟรมที่ 394 396 398 และเฟรมที่ 400 จะเห็นว่าขั้นตอนวิธี MST_AC ให้ความถูกต้องในการติดตามมากกว่าขั้นตอนวิธีอื่น ๆ



MS_ICA

MST_AC

MS

MS_BG_10

ภาพประกอบ 4.25 ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC ลำดับเฟรม 394 396 398 และ 400 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล ComSci 2

ตัวอย่างการติดตามที่ยังมีปัญหาของ MST_AC จากภาพประกอบ 4.24 ด้านบน ในช่วงเฟรมที่ 204-212 ตำแหน่งที่วงกลมด้านบนขั้นตอนวิธี MST_AC ให้ค่าระยะทางผิดพลาดมากกว่าขั้นตอนวิธีอื่น ๆ เนื่องจากภาพวัตถุ (เสื้อสีฟ้า) มีการเปลี่ยนรูปแบบโดยมีการเอียงตั้ง ตัวอย่างในภาพประกอบ 4.26 ในเฟรมที่ 214 216 และเฟรมที่ 218 ตำแหน่งที่มีวงกลม



MS_ICA

MST_AC

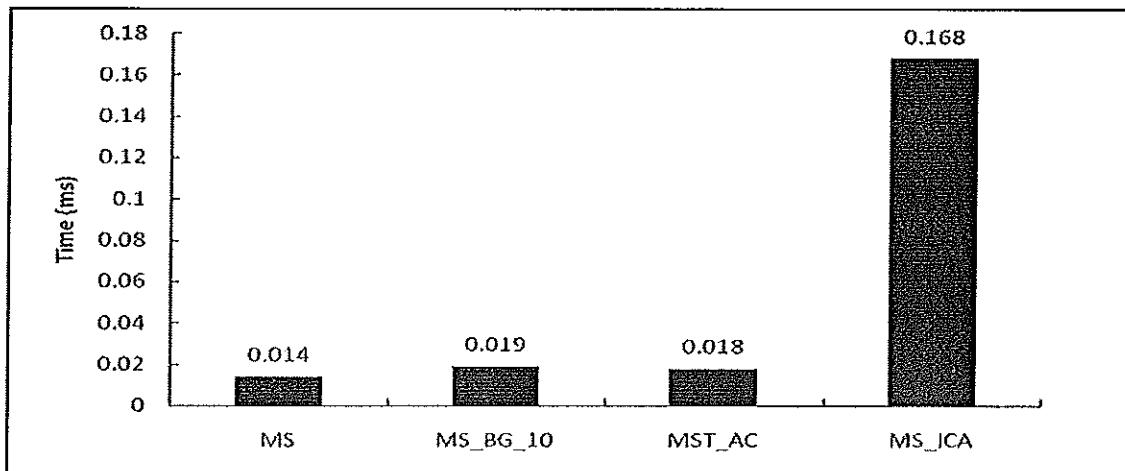
MS

MS_BG_10

ภาพประกอบ 4.26 ตัวอย่างการติดตามที่ยังมีปัญหของขั้นตอนวิธี MST_AC
ลำดับเฟรม 201 210 212 214 216 และ 218 ตามแนวตั้งจากบนลงล่าง

4.3.6.3 ค่าเวลาเฉลี่ย

ค่าเวลาเฉลี่ยในการทำงาน 1 รอบ ของชุดข้อมูล ComSci 2 ดังภาพประกอบ 4.27 จะเห็นได้ว่า MS ใช้เวลาในการวนรอบ 1 ครั้งน้อยที่สุด คือ 0.014 ms และใกล้เคียงกับขั้นตอนวิธี MS_BC_10 คือ 0.019 ms และขั้นตอนวิธี MST_AC คือ 0.018 ms แต่สำหรับขั้นตอนวิธี MS_ICA นั้นใช้เวลาในการวนรอบมากที่สุดคือ 0.168 ms



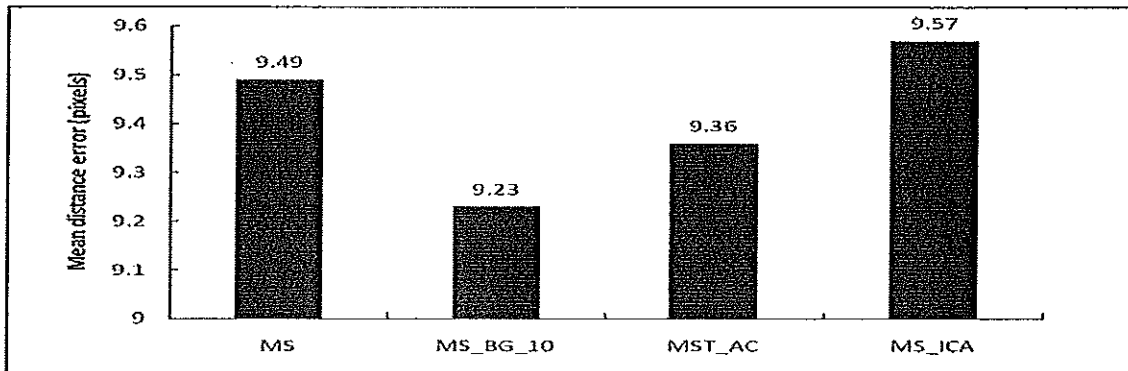
ภาพประกอบ 4.27 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล ComSci 2

4.3.7 ชุดข้อมูล ComSci 3

ชุดข้อมูลนี้วัตถุประสงค์เป้าหมายคือคนวิ่งนอกสถานที่ในทิศทางห่างและใกล้จากกล้องใส่อีสือขาว

4.3.7.1 ค่าระยะทางผิดพลาดเฉลี่ย

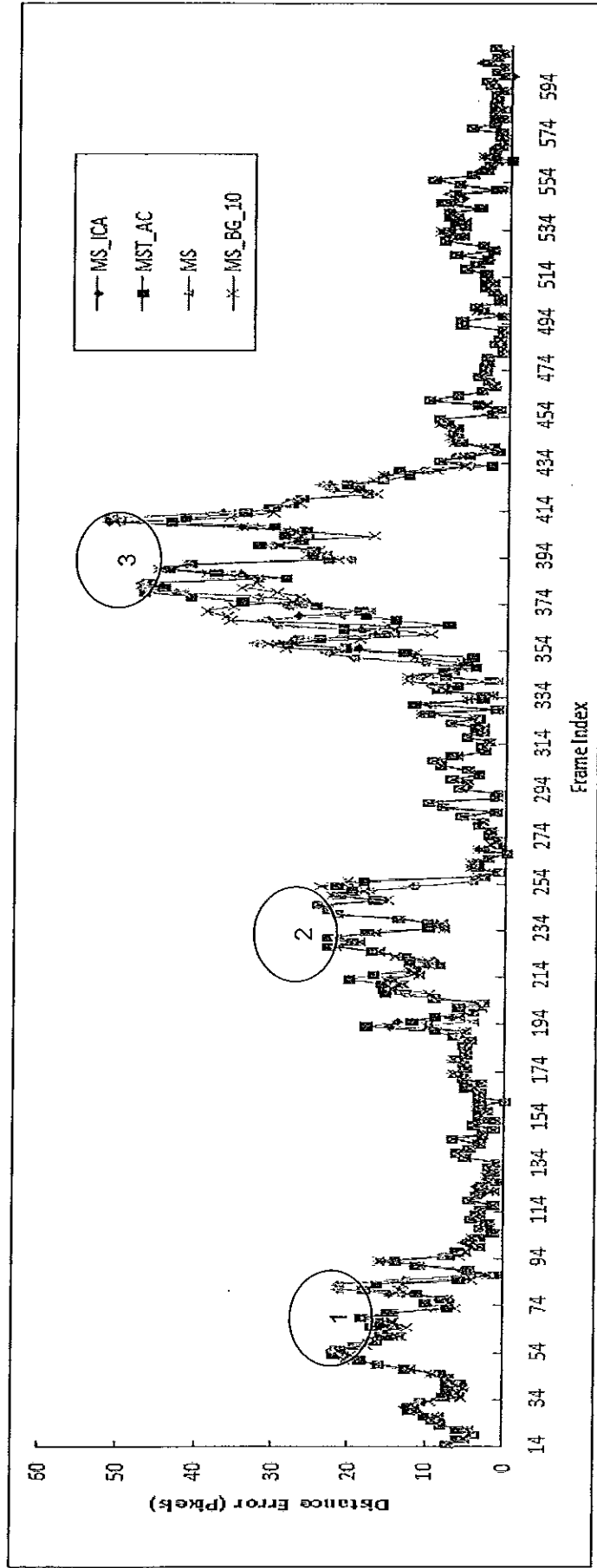
ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยใช้ชุดทดสอบ ComSci 3 ดังภาพประกอบ 4.28 จะเห็นได้ว่าขั้นตอนวิธี MS_BG_10 ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยที่สุด 9.23 MS ให้ค่า 9.49 MST_AC ให้ค่า 9.36 และ MS_ICA ให้ค่า 9.57 แต่อย่างไรก็ตามค่าความแตกต่างของระยะทางผิดพลาดไม่แตกต่างกันมากเนื่องจากสีของวัตถุเป้าหมายกับสีของพื้นหลังแตกต่างกันมาก



ภาพประกอบ 4.28 ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธีสำหรับ ข้อมูลทดสอบ ComSci 3

4.3.7.2 ค่าระยะทางผิดพลาด

ค่าระยะทางผิดพลาดของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ ComSci 3 ดังภาพประกอบ 4.29 ข้างล่างจะเห็นว่าในแต่ละขั้นตอนวิธีให้ค่าระยะทางผิดพลาดใกล้เคียงกัน



ภาพประกอบ 4.29 ค่าระยะทางผิดพลาดในแต่ละเฟรมของต้นขั้วคอนกรีต โดยใช้ข้อมูลทดสอบ ComSci 3

ตัวอย่างการติดตามที่มีปัญหา จากภาพประกอบ 4.29 ข้างบน ที่วงกลมหมายเลข 1 จะเห็นว่า เป็นจุดที่ค่าระยะทางผิดพลาดของทุกขั้นตอนวิธีมีค่าสูง สาเหตุเนื่องจากแบบจำลองมีการเปลี่ยนแปลงจะเห็นได้จากเส้นสีขาวที่เป็นเป้าหมายมีการเอียง ตัวอย่างในภาพประกอบ 4.30 ในเฟรมที่ 52 54 และ 56 จากภาพประกอบ 4.29 ข้างบน วงกลมหมายเลข 2 เป็นจุดที่ค่าระยะทางผิดพลาดของทุกขั้นตอนวิธีมีค่าสูง ค่าระยะทางผิดพลาดของแต่ละขั้นตอนวิธีมีค่าสูง เนื่องจากวัตถุเป้าหมายอยู่ไกลทำให้ภาพวัตถุมีขนาดเล็กไม่ชัดเจน ตัวอย่างดังภาพประกอบ 4.31 ในเฟรมที่ 214 216 และ 220 จากภาพประกอบ 4.29 ข้างบน วงกลมหมายเลข 3 ระยะทางผิดพลาดของแต่ละขั้นตอนวิธีมีค่าสูง สาเหตุเนื่องจากพื้นที่ภาพของวัตถุเป้าหมายมีขนาดใหญ่เกินกว่าขนาดของหน้าต่าง ทำให้ค่าระยะทางผิดพลาดมีค่าสูง ตัวอย่างดังภาพประกอบที่ 4.32 ในเฟรมที่ 374 และ 380



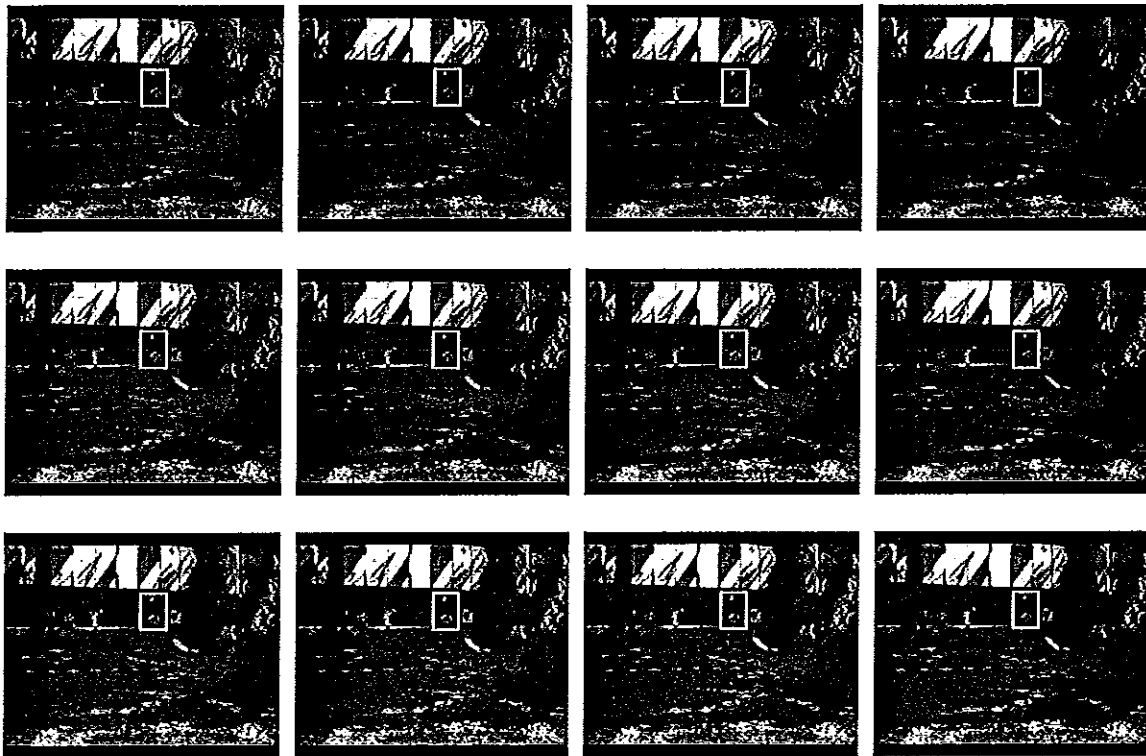
MS_ICA

MST_AC

MS

MS_BG_10

ภาพประกอบ 4.30 ตัวอย่างการติดตามที่ยังมีปัญห (วงกลมหมายเลข 1) ลำดับเฟรม 52 54 และ 56 ตามแนวตั้งจากบนลงล่าง โดยใช้ข้อมูลทดสอบ ComSci 3



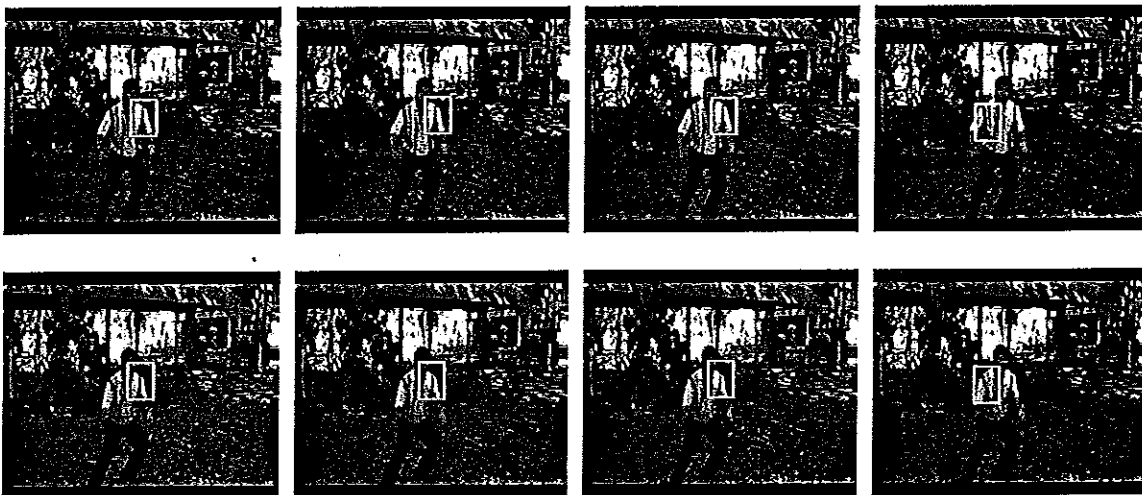
MS_ICA

MST_AC

MS

MS_BG_10

ภาพประกอบ 4.31 ตัวอย่างการติดตามที่ยังมีปัญห (วงกลมหมายเลข 2) ลำดับเฟรม 214 216 และ 220 ตามแนวตั้งจากบนลงล่าง โดยใช้ข้อมูลทดสอบ ComSci 3



MS_ICA

MST_AC

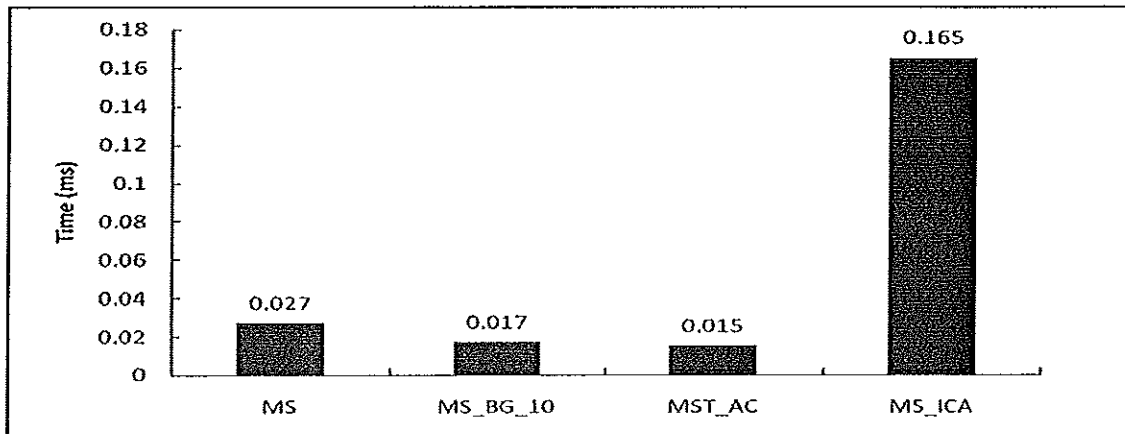
MS

MS_BG_10

ภาพประกอบ 4.32 ตัวอย่างการติดตามที่ยังมีปัญห (วงกลมหมายเลข 3) ลำดับเฟรม 374 และ 380 ตามแนวตั้งจากบนลงล่าง โดยใช้ข้อมูลทดสอบ ComSci 3

4.3.7.3 ค่าเวลาเฉลี่ย

ค่าเวลาเฉลี่ยในการทำงาน 1 รอบ ของชุดข้อมูล ComSci 3 ดังภาพประกอบ 4.33 จะเห็นว่าขั้นตอนวิธี MST_AC ใช้เวลาน้อยที่สุดคือ 0.015 ms และใกล้เคียงกับขั้นตอนวิธี MS คือ 0.027 ms และขั้นตอนวิธี MS_BG_10 คือ 0.017 ms แต่สำหรับขั้นตอนวิธี MST_ICA ใช้เวลาในการวนรอบมากที่สุดคือ 0.165 ms



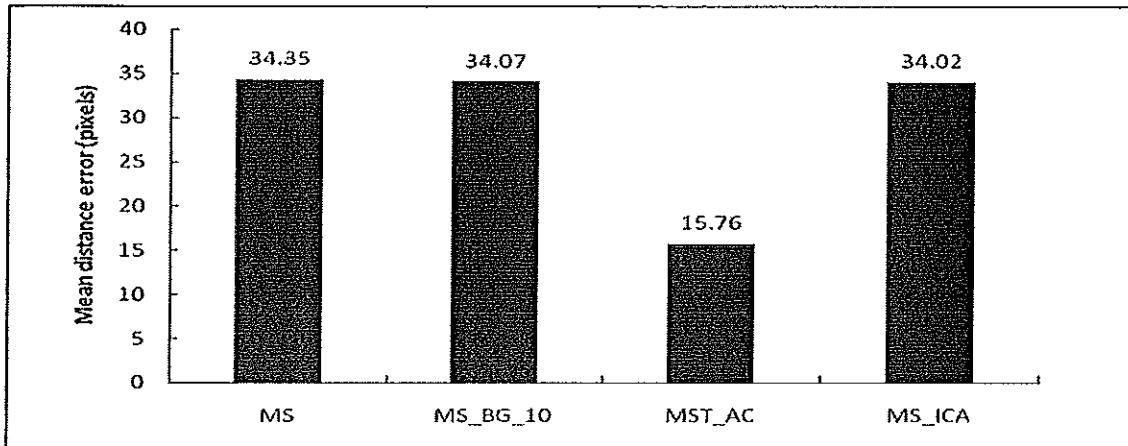
ภาพประกอบ 4.33 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล ComSci 3

4.3.8 ชุดข้อมูล ComSci 4

ชุดข้อมูลนี้วัตถุประสงค์เป้าหมายคือคนวิ่งนอกสถานที่ใส่เสื้อสีดำเดินผ่านพื้นหลังสีดำ

4.3.8.1 ค่าระยะทางผิดพลาดเฉลี่ย

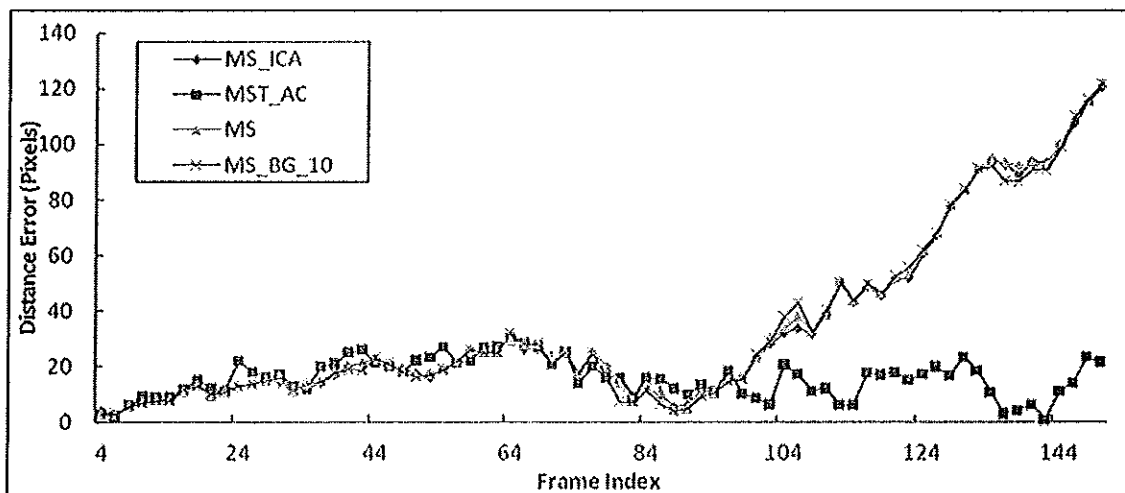
ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยใช้ชุดข้อมูลทดสอบ ComSci 4 ดังภาพประกอบ 4.34 จะเห็นว่าขั้นตอนวิธี MST_AC ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยที่สุด คือ 15.76 MS คือ 34.35 MS_BG_10 คือ 34.07 และ MS_ICA คือ 34.02 สาเหตุที่ทำให้ขั้นตอนวิธี MST_AC ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยกว่าขั้นตอนวิธีอื่นมากเนื่องจากการติดตามของขั้นตอนวิธีอื่นๆ มีการติดตามที่ผิดพลาด



ภาพประกอบ 4.34 ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ ComSci 4

4.3.8.2 ค่าระยะทางผิดพลาด

ค่าระยะทางผิดพลาดของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ ComSci 4 ดังภาพประกอบ 4.35 ข้างล่างจะเห็นว่าในช่วงเฟรมที่ 4 ถึง 100 ขั้นตอนวิธี MST_AC ให้ค่าระยะทางผิดพลาดใกล้เคียงกับขั้นตอนวิธีอื่นแต่หลังจากเฟรมที่ 100 ขั้นตอนวิธี MST_AC ให้ค่าระยะทางผิดพลาดน้อยกว่าขั้นตอนวิธีอื่น ๆ



ภาพประกอบ 4.35 ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ ComSci 4

ตัวอย่างการติดตามที่ดีของ MST_AC จากภาพประกอบ 4.36 ในเฟรมที่ 100 105 110 115 และเฟรมที่ 120 จะเห็นว่าขั้นตอนวิธี MST_AC ให้ความถูกต้องในการติดตามมากกว่าขั้นตอนวิธีอื่น ๆ สังเกตจากวงกลมข้างล่าง



MS_ICA

MST_AC

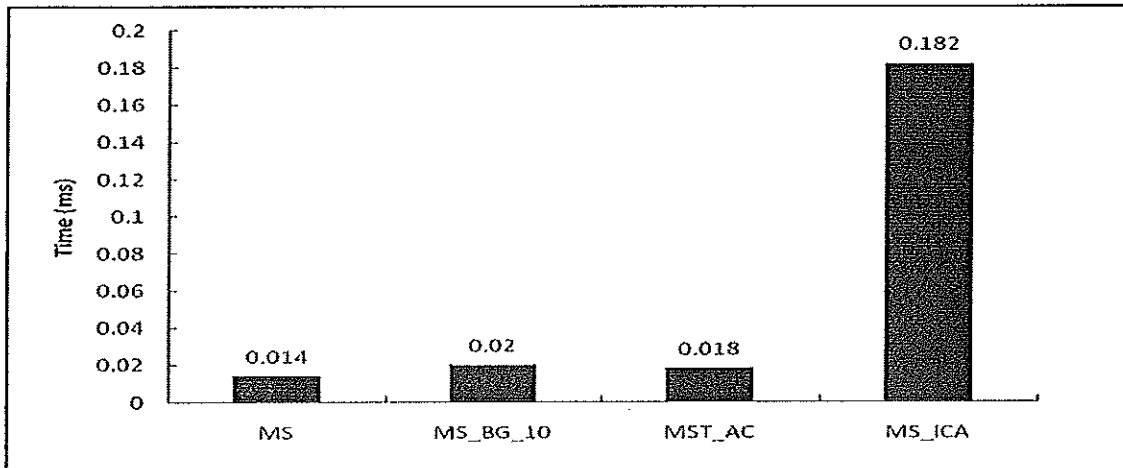
MS

MS_BG_10

ภาพประกอบ 4.36 ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC ลำดับเฟรม 100 105 110 115 และ 120 ตามแนวตั้งจากบนลงล่าง

4.3.8.3 ค่าเวลาเฉลี่ย

ค่าเวลาเฉลี่ยในการทำงาน 1 รอบ ของชุดข้อมูล ComSci 4 ดังภาพประกอบ 4.37 จะเห็นได้ว่าขั้นตอนวิธี MS ใช้เวลาน้อยที่สุดคือ 0.014 ms และใกล้เคียงกับขั้นตอนวิธี MS_BG_10 คือ 0.02 ms และขั้นตอนวิธี MST_AC คือ 0.018 ms แต่สำหรับขั้นตอนวิธี MST_ICA ใช้เวลาในการวนรอบมากที่สุดคือ 0.182 ms



ภาพประกอบ 4.37 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล ComSci 4

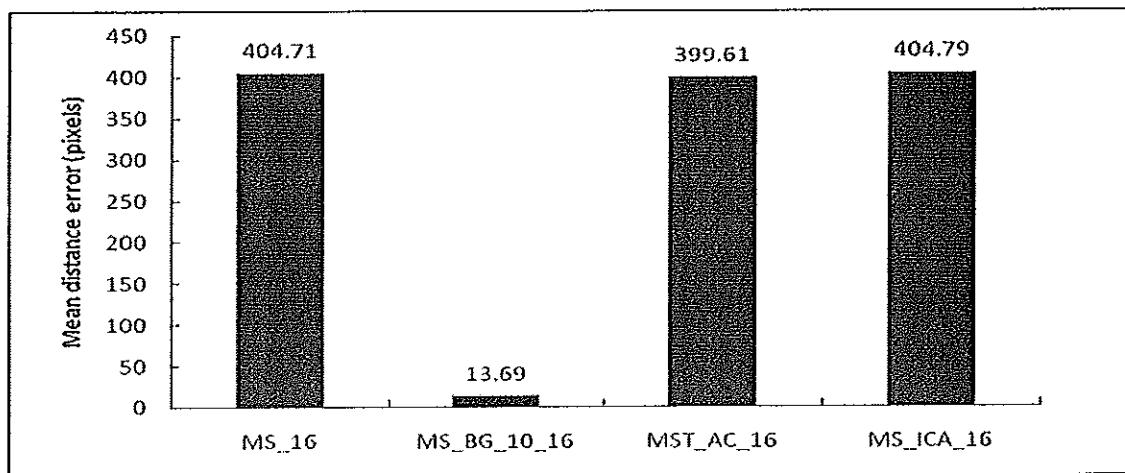
4.4 เปรียบเทียบความถูกต้องในการติดตามโดยเปลี่ยนจำนวน bin

ในการทดลองนี้ใช้จำนวน bin ที่แตกต่างกัน คือ จำนวน bin 32x32x32 และจำนวน bin 16x16x16 โดยใช้ขั้นตอนวิธี MS MS_BG_10 MST_AC และ MS_ICA ขั้นตอนวิธีที่ใช้จำนวน bin จำนวน 32x32x32 คือ MS_32 MS_BG_10_32 MST_AC_32 และ MS_ICA_32 ขั้นตอนวิธีที่ใช้จำนวน bin จำนวน 16x16x16 คือ MS_16 MS_BG_10_16 MST_AC_16 และ MS_ICA_16

ข้อมูลทดสอบที่ใช้ คือ Airport 1 Airport 2 และ Airport 3 คุณสมบัติของข้อมูล ดังตาราง 4.1 ข้างต้น

4.4.1 ค่าระยะทางผิดพลาดเฉลี่ย

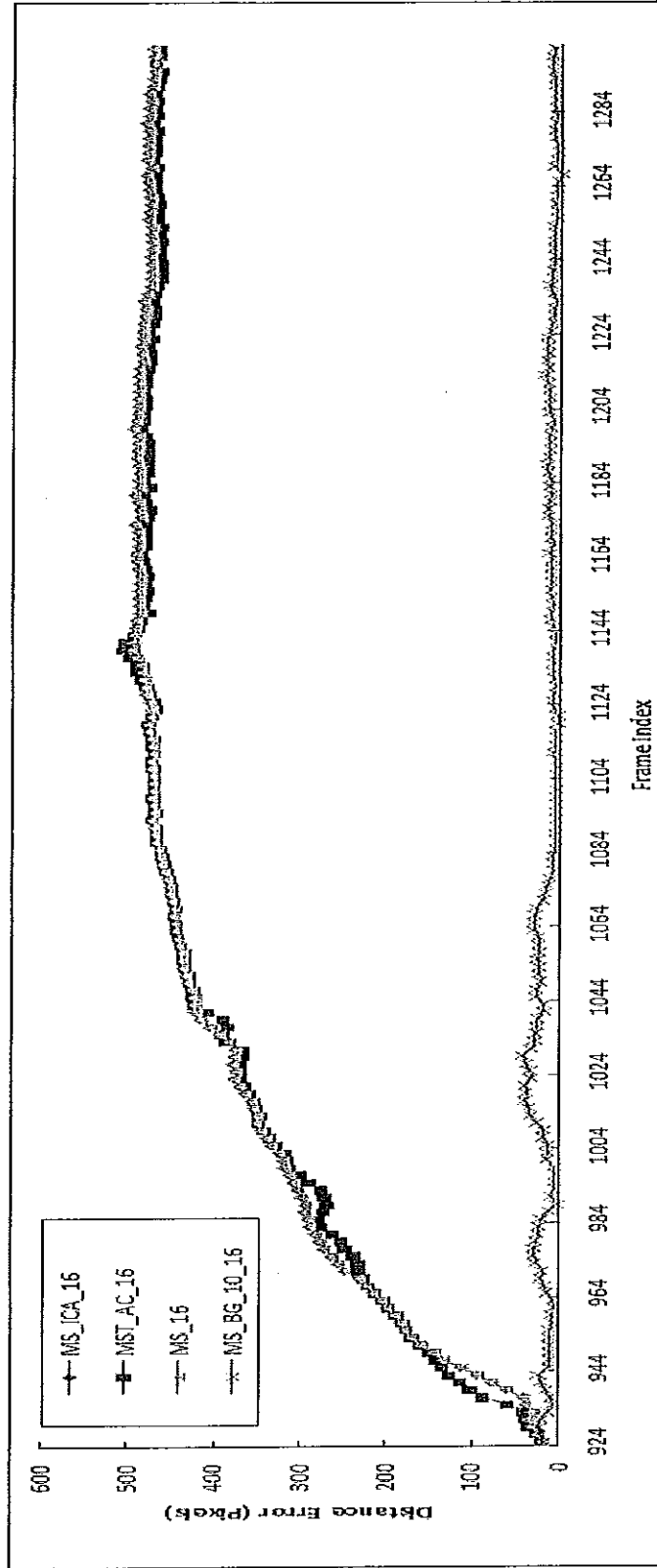
ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport 1 ดังภาพประกอบ 4.38 จะเห็นได้ว่า MS_BG_10_16 ให้ค่าระยะทางผิดพลาดน้อยที่สุด คือ 13.69 MS_16 ให้ค่า 404.71 MST_AC_16 ให้ค่า 399.61 และ MS_ICA_16 ให้ค่า 404.79



ภาพประกอบ 4.38 ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 1

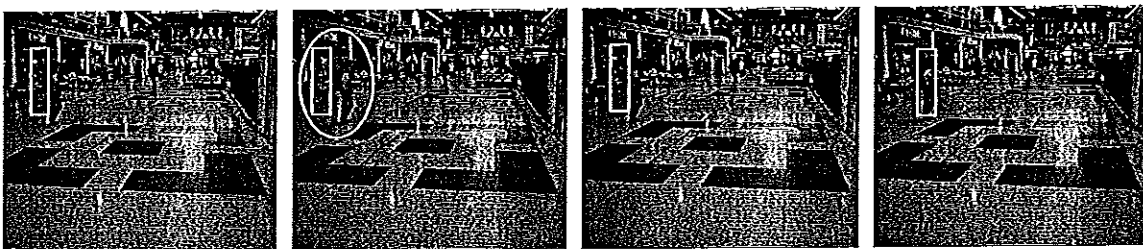
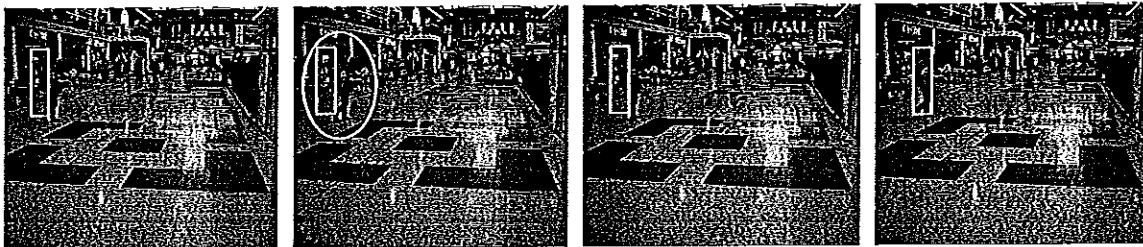
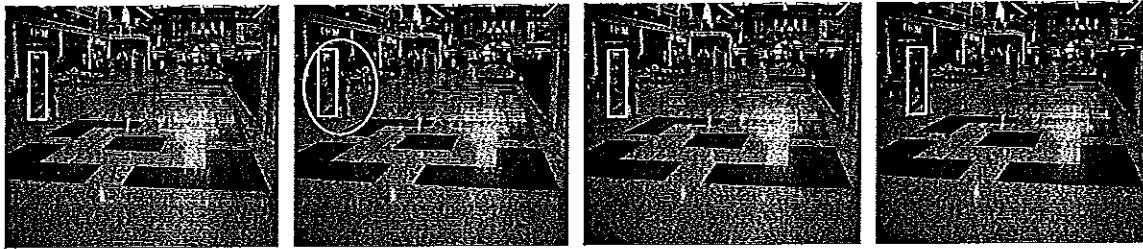
4.4.2 ค่าระยะทางผิดพลาด

ค่าระยะทางผิดพลาดของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport 1 ดังภาพประกอบ 4.39 จะเห็นได้ว่าตอนเริ่มต้นของเฟรมตั้งแต่ 921 ถึง 926 ทุกขั้นตอนวิธีให้ค่าใกล้เคียงกันแต่พอหลังจากเฟรมที่ 926 ขั้นตอนวิธี MS_BG_10_16 ให้ค่าระยะทางผิดพลาดน้อย ในขณะที่ขั้นตอนวิธีอื่น ๆ ให้ค่าระยะทางผิดพลาดสูงเนื่องมาจากการติดตามผิดพลาดเนื่องจากสีของพื้นหลังมีสีเดียวกับสีของวัตถุและรวมไปถึงการใช้ขนาด bin $16 \times 16 \times 16$ ทำให้สีของพื้นหลังที่มีสีคนละสีกับสีของวัตถุมีโอกาสที่มีสีเดียวกับวัตถุมากขึ้น



ภาพประกอบ 4.39 ค่าระยะทางผิดพลาดในแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 1

ตัวอย่างการติดตามที่ยังมีปัญหของ MST_AC_16 โดยใช้ขนาดของ bin จำนวน 16x16x16 จากภาพประกอบ 4.40 ในเฟรมที่ 927 929 และเฟรมที่ 931 จะเห็นว่าขั้นตอนวิธี MST_AC_16 MS_ICA_16 และ MS_16 ให้ความถูกต้องในการติดตามน้อยกว่าขั้นตอนวิธี MS_BG_10_16 สังเกตจากวงกลมดังรูปข้างล่าง



MS_ICA_16

MST_AC_16

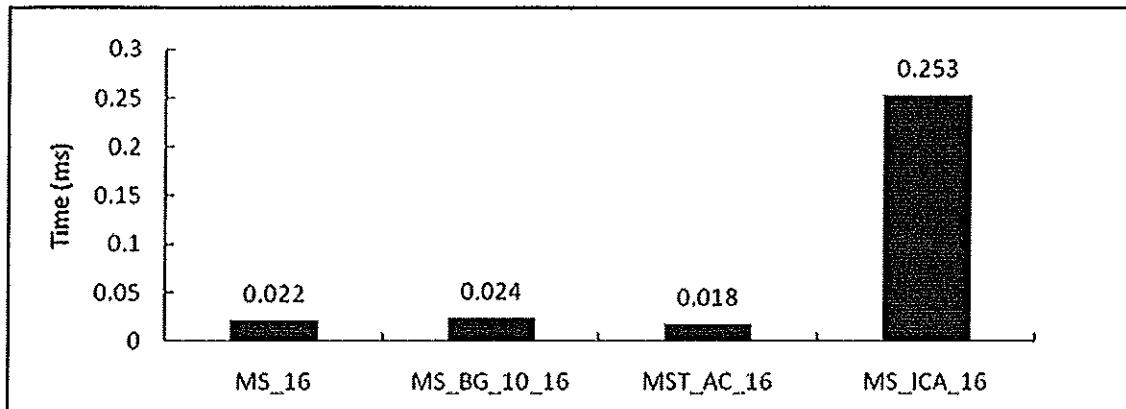
MS_16

MS_BG_10_16

ภาพประกอบ 4.40 ตัวอย่างการติดตามที่ยังมีปัญหของขั้นตอนวิธี MST_AC ลำดับเฟรม 927 929 และ 931 ตามแนวตั้งจากบนลงล่าง

4.4.3 ค่าเวลาเฉลี่ย

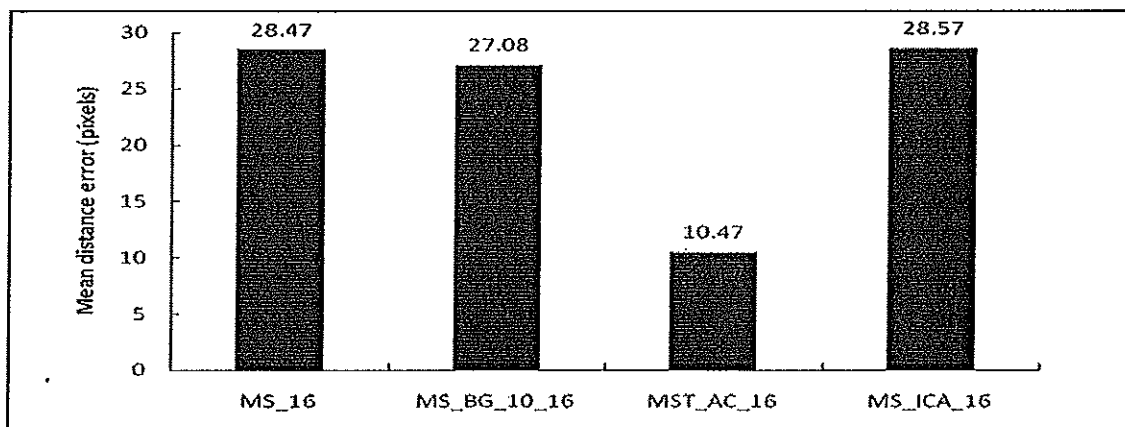
ค่าเวลาเฉลี่ยในการทำงาน 1 รอบ ของชุดข้อมูล Airport 1 ดังภาพประกอบ 4.41 จะเห็นว่า MST_AC_16 ให้ค่าเวลาเฉลี่ยน้อยที่สุด คือ 0.018 ms และใกล้เคียงกับ MS_16 คือ 0.022 ms MS_BG_10_16 คือ 0.024 ms สำหรับขั้นตอนวิธี MS_ICA_16 ให้ค่าเวลาเฉลี่ยมากที่สุด คือ 0.253 ms



ภาพประกอบ 4.41 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล Airport 1

4.4.4 ค่าระยะทางผิดพลาดเฉลี่ย

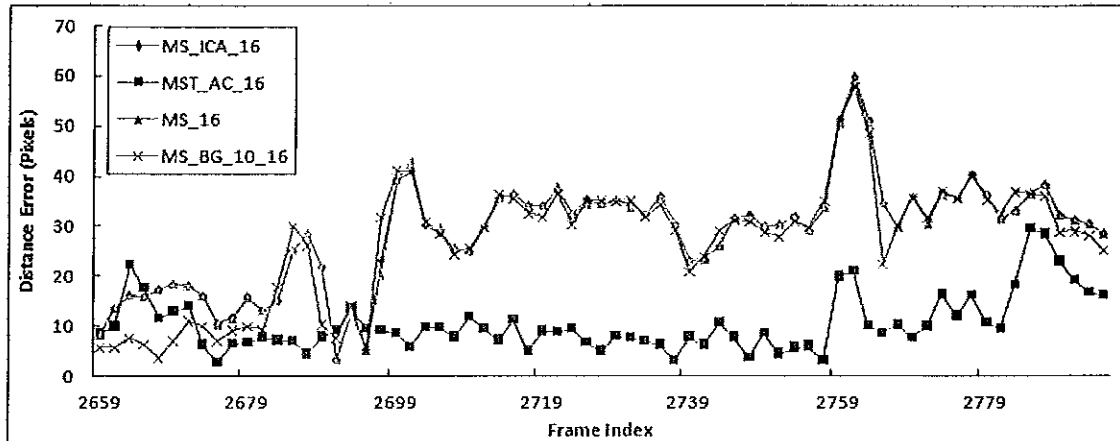
ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport 2 ดังภาพประกอบ 4.42 จะเห็นได้ว่า MST_AC_16 ให้ค่าระยะทางผิดพลาดน้อยที่สุด คือ 10.47 MS_16 ให้ค่า 28.47 MS_BG_10_16 ให้ค่า 27.08 และ MS_ICA_16 ให้ค่า 28.57



ภาพประกอบ 4.42 ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 2

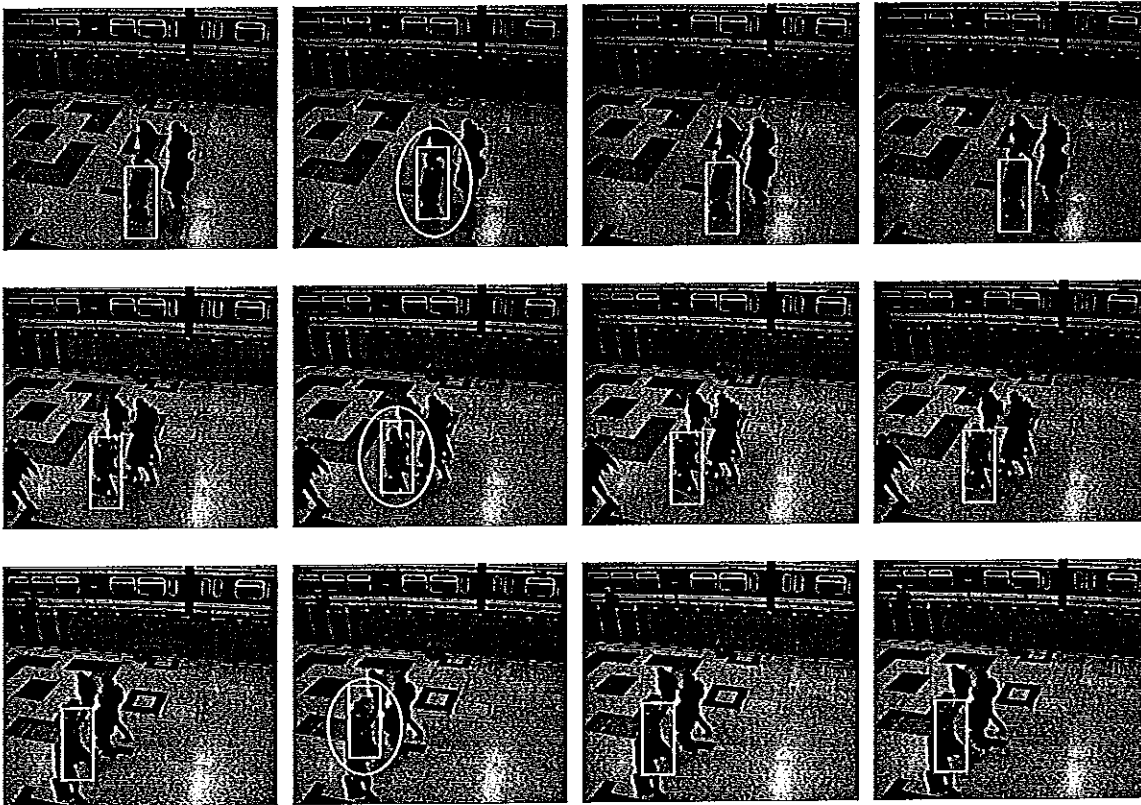
4.4.5 ค่าระยะทางผิดพลาด

ค่าระยะทางผิดพลาดของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport 2 ดังภาพประกอบ 4.43 จะเห็นจุดส่วนใหญ่ของขั้นตอนวิธี MST_AC_16 ต่ำกว่าขั้นตอนวิธีอื่น ๆ



ภาพประกอบ 4.43 ค่าระยะทางผิดพลาดในแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 2

ตัวอย่างการติดตามที่ดีของ MST_AC_16 จากภาพประกอบ 4.44 ในเฟรมที่ 2719 2739 และเฟรมที่ 2759 จะเห็นได้ว่าขั้นตอนวิธี MST_AC_16 ให้ความถูกต้องในการติดตามมากกว่าขั้นตอนวิธีอื่น ๆ สังเกตจากวงกลมข้างล่าง



MS_ICA_16

MST_AC_16

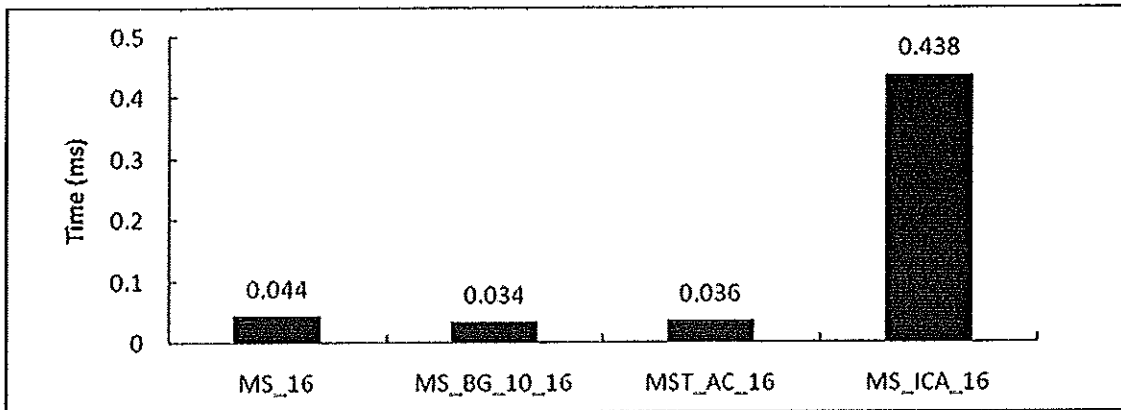
MS_16

MS_BG_10_16

ภาพประกอบ 4.44 ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC_16 ลำดับเฟรม 2719 2739 และ 2759 ตามแนวตั้งจากบนลงล่าง โดยใช้ข้อมูลทดสอบ Airport 2

4.4.9 ค่าเวลาเฉลี่ย

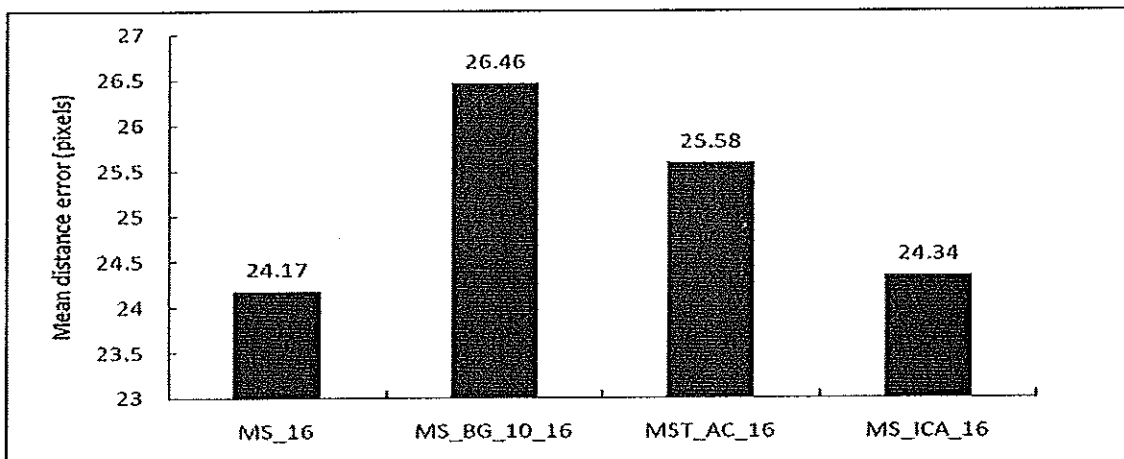
ค่าเวลาเฉลี่ยในการทำงาน 1 รอบ ของชุดข้อมูล Airport 2 ดังภาพประกอบ 4.45 จะเห็นได้ว่า MS_BG_10_16 ให้ค่าเวลาเฉลี่ยน้อยที่สุด คือ 0.034 ms และใกล้เคียงกับ MS_16 คือ 0.044 ms MST_AC_16 คือ 0.036 ms สำหรับขั้นตอนวิธี MS_ICA_16 ให้ค่าเวลาเฉลี่ยมากที่สุด คือ 0.438 ms



ภาพประกอบ 4.45 ค่าเวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล Airport 2

4.4.7 ค่าระยะทางผิดพลาดเฉลี่ย

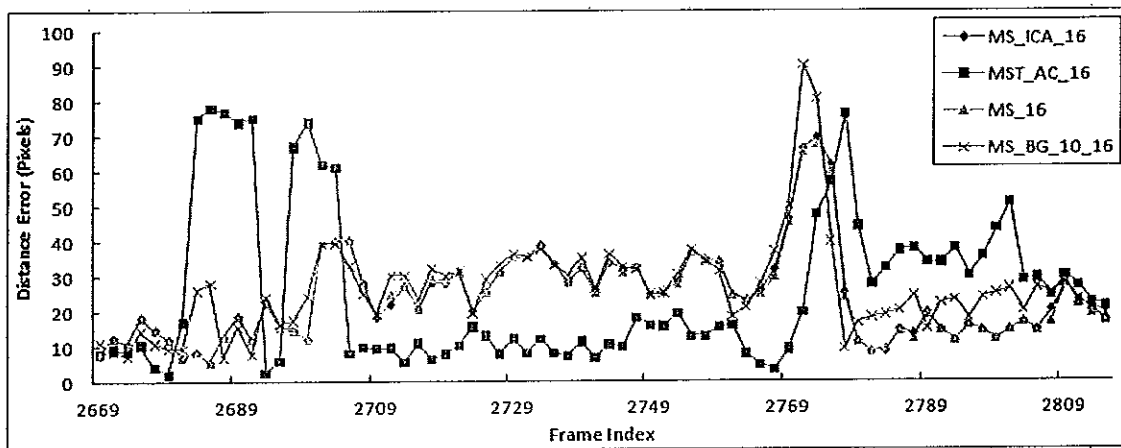
ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport 3 ดังภาพประกอบ 4.46 จะเห็นได้ว่า MS_16 ให้ค่าระยะทางผิดพลาดน้อยที่สุด คือ 24.17 MS_BG_10_16 ให้ค่า 26.46 MST_AC_16 ให้ค่า 25.58 และ MS_ICA_16 ให้ค่า 24.37



ภาพประกอบ 4.46 ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 3

4.4.8 ค่าระยะทางผิดพลาด

ค่าระยะทางผิดพลาดของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport 3 ดังภาพประกอบ 4.47 จะเห็นได้ว่าในเฟรมช่วง 2679 ถึง 2688 และ 2694 ถึง 2700 ขั้นตอนวิธี MST_AC_16 ให้ค่าระยะทางผิดพลาดสูงกว่าขั้นตอนวิธีอื่น ๆ แต่พอในเฟรมช่วง 2709 ถึง 2769 MST_AC_16 ให้ค่าระยะทางผิดพลาดต่ำกว่าขั้นตอนวิธีอื่น ๆ



ภาพประกอบ 4.47 ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 3

ตัวอย่างการติดตามที่ดีของ MST_AC_16 จากภาพประกอบ 4.48 ในเฟรมที่ 2729 และ 2769 จะเห็นได้ว่าขั้นตอนวิธี MST_AC_16 ให้ความถูกต้องในการติดตามมากกว่าขั้นตอนวิธีอื่น ๆ สังเกตจากวงกลมข้างล่าง



MS_ICA_16

MST_AC_16

MS_16

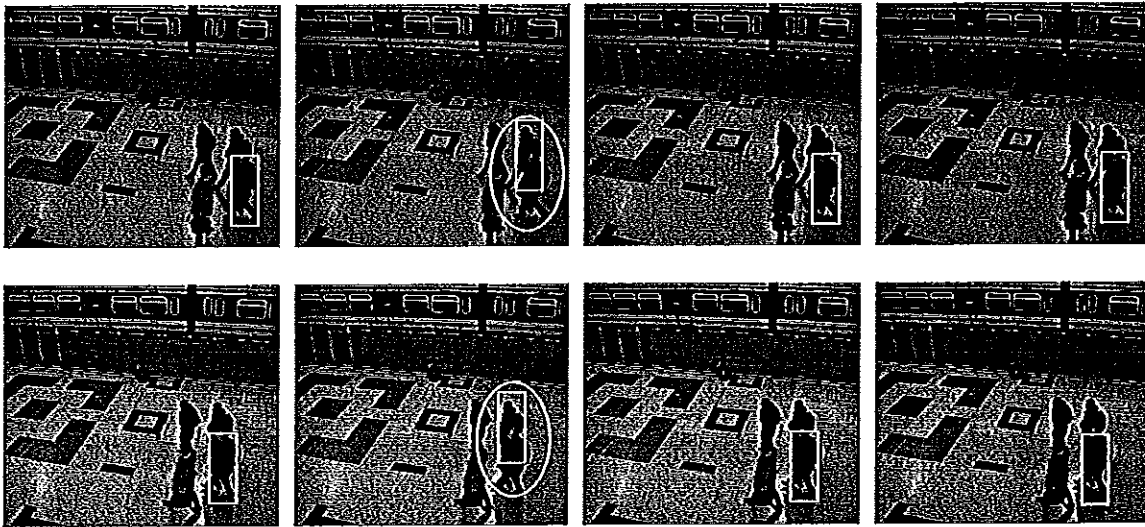
MS_BG_10_16

ภาพประกอบ 4.48 ตัวอย่างการติดตามที่ดีที่สุดของขั้นตอนวิธี MST_AC_16 ลำดับเฟรม 2729 และ 2769 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล Airport 3

ตัวอย่างการติดตามยังมีปัญหาของ MST_AC_16 จากภาพประกอบ 4.49 ข้างล่างในเฟรมที่ 2688 และ 2697 จะเห็นได้ว่าขั้นตอนวิธี MST_AC_16 ให้ความถูกต้องในการติดตามน้อยกว่าขั้นตอนวิธีอื่น ๆ เนื่องจากจำนวน bin มีน้อยและสีของพื้นหลังมีสีเดียวกับวัตถุ คือ สีดำ สังเกตจากวงกลมข้างล่าง

4.4.9 ค่าเวลาเฉลี่ย

ค่าเวลาเฉลี่ยในการทำงาน 1 รอบ ของชุดข้อมูล Airport 3 ดังภาพประกอบ 4.50 MS_BG_10_16 ให้ค่าเวลาน้อยที่สุด คือ 0.026 ms และใกล้เคียงกับ MS_16 คือ 0.045 ms และ MST_AC_16 คือ 0.031 ms สำหรับ MS_ICA_16 ให้ค่าเวลาเฉลี่ยมากที่สุด คือ 0.35 ms



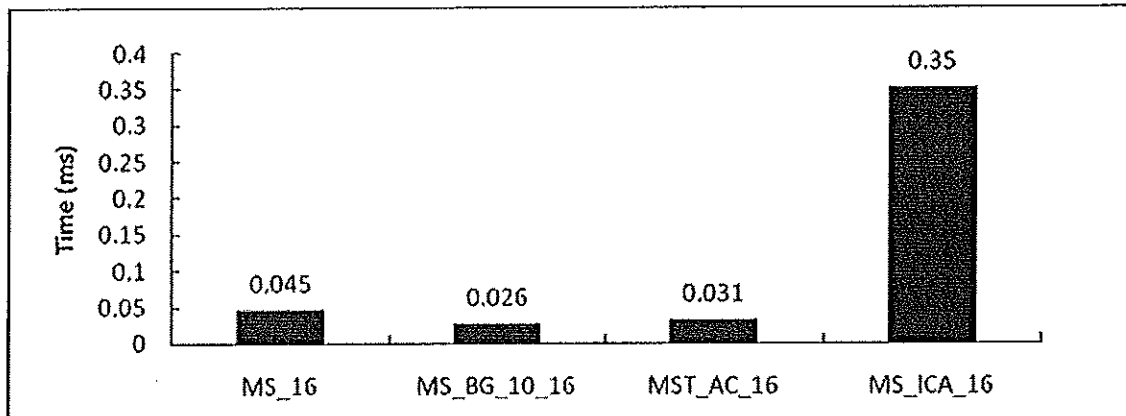
MS_ICA_16

MST_AC_16

MS_16

MS_BG_10_16

ภาพประกอบ 4.49 ตัวอย่างการติดตามที่ยังมีปัญหของขั้นตอนวิธี MST_AC_16 ลำดับเฟรม 2688 และ 2697 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล Airport 3



ภาพประกอบ 4.50 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล Airport 3

4.5 เปรียบเทียบความถูกต้องในการติดตามโดยเปลี่ยนความคมชัด (Resolution)

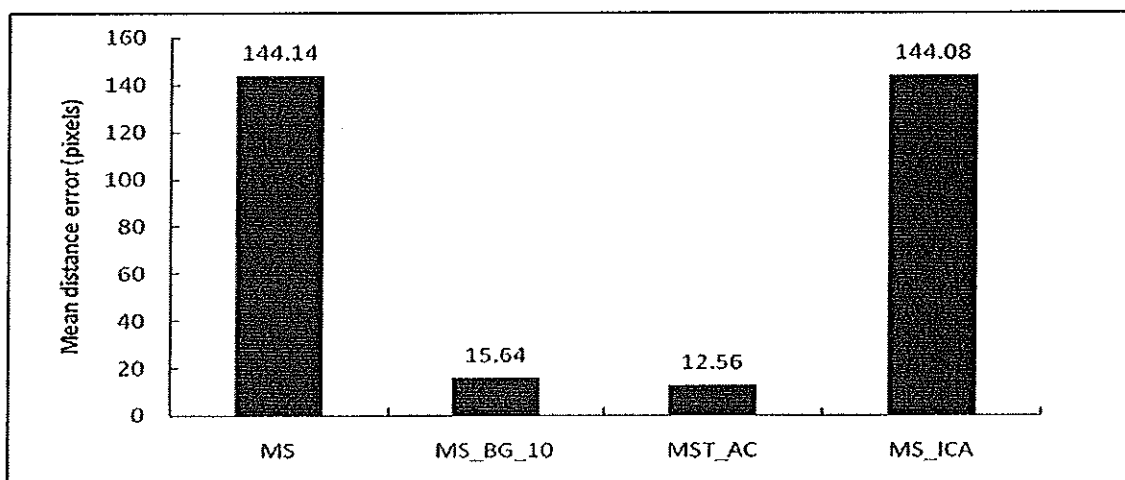
ในการทดลองนี้เพื่อเปรียบเทียบความถูกต้องในการติดตามโดยดูจากค่าระยะทางผิดพลาดเฉลี่ยของขั้นตอนวิธี เมื่อเปลี่ยนความคมชัดของวิดีโอ การทดลองใช้ขั้นตอนวิธี MST_AC MS MS_BG_10 และ MS_ICA ชุดข้อมูลทดสอบที่ใช้ คือ Airport1_new Airport 2_new และ Airport 3_new ซึ่งเป็นวิดีโอที่ได้ใหม่ที่ได้จากการนำชุดข้อมูลทดสอบ Airport1 Airport2 และ Airport3 (ตาราง 4.1) มาทำการบีบอัดใหม่ทำให้ความคมชัดของวิดีโอต่ำโดยคุณสมบัติของชุดข้อมูลใหม่ ดังตาราง 4.2 ข้างล่าง

ตารางที่ 4.2 รายละเอียดชุดข้อมูลทดสอบใหม่

ชื่อ VDO	จำนวนเฟรม	ขนาดหน้าต่าง	ความคมชัด
Airport 1_new	100	43x139	ต่ำ
Airport 2_new	100	65x156	ต่ำ
Airport 3_new	100	50x155	ต่ำ

4.5.1 ค่าระยะทางผิดพลาดเฉลี่ย

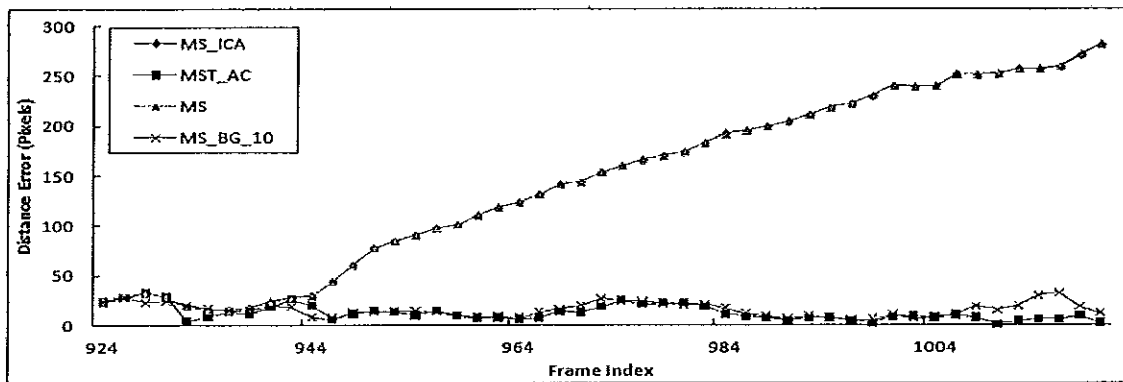
ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport1_new ดังภาพประกอบ 4.51 จะเห็นได้ว่า MST_AC ให้ค่าระยะทางผิดพลาดน้อยที่สุดคือ 12.56 MS ให้ค่า 144.14 MS_BG_10 ให้ค่า 15.64 และ MS_ICA ให้ค่า 144.08 การที่ MS และ MS_ICA ให้ค่าสูงมากกว่าปกติเนื่องจากการติดตามผิดพลาด



ภาพประกอบ 4.51 ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 1_new

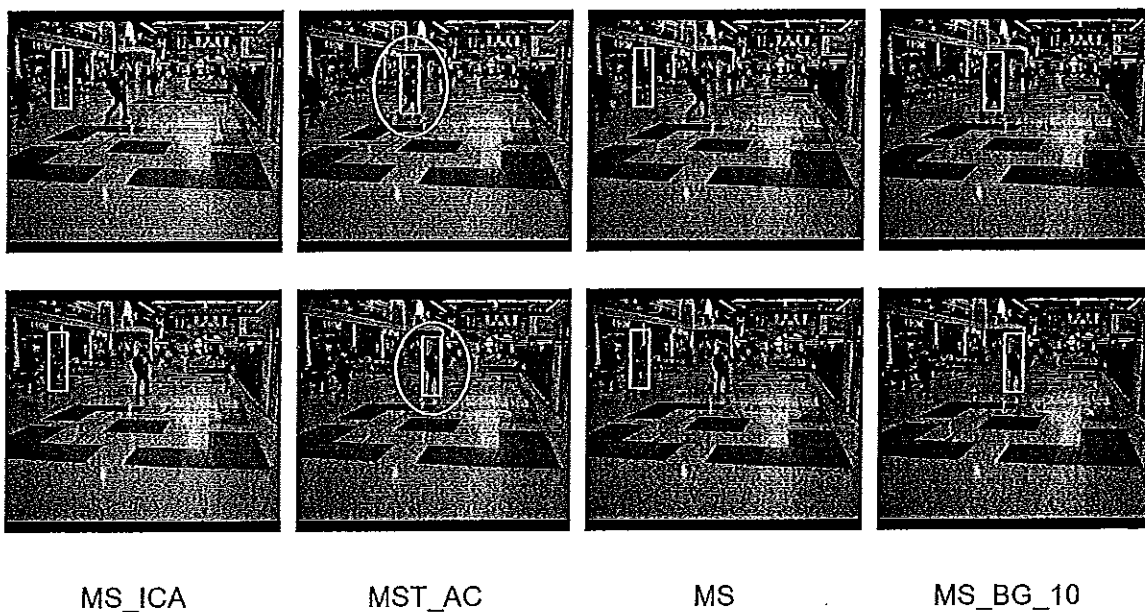
4.5.2 ค่าระยะทางผิดพลาด

ค่าระยะทางผิดพลาดของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport1_new ดังภาพประกอบ 4.52 จะเห็นว่าจุดส่วนใหญ่ของขั้นตอนวิธี MST_AC จะอยู่ต่ำกว่าขั้นตอนวิธีอื่น ๆ



ภาพประกอบ 4.52 ค่าระยะทางผิดพลาดในแต่ละเฟรมของแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport1_new

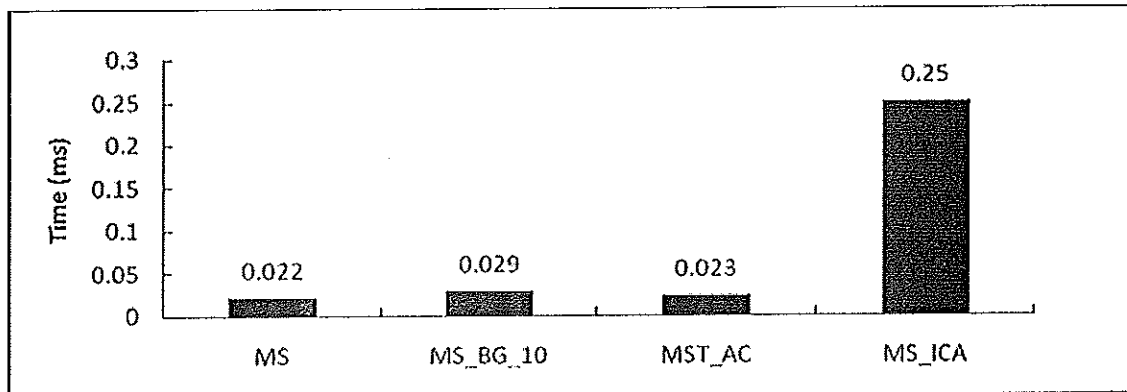
ตัวอย่างการติดตามที่ดีของ MST_AC จากภาพประกอบ 4.53 ในเฟรมที่ 964 และ 984 จะเห็นได้ว่าขั้นตอนวิธี MST_AC ให้ความถูกต้องในการติดตามมากกว่าขั้นตอนวิธี MS_ICA และ MS และใกล้เคียงกับขั้นตอนวิธี MS_BG_10 สังเกตจากวงกลมข้างล่าง



ภาพประกอบ 4.53 ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC ลำดับเฟรม 964 และ 984 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล Airport1_new

4.5.3 ค่าเวลาเฉลี่ย

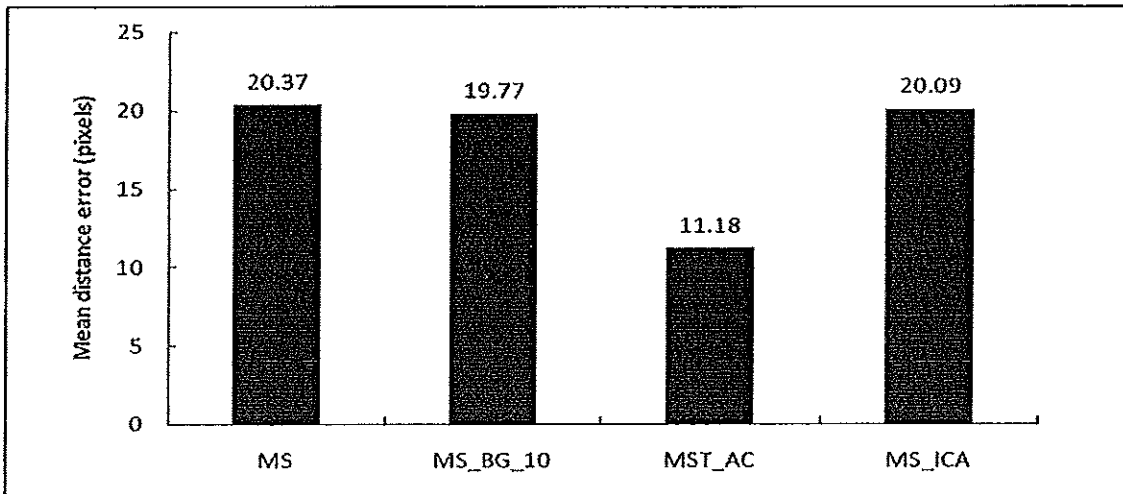
ค่าเวลาเฉลี่ยในการทำงาน 1 รอบ ของชุดข้อมูล Airport1_new จากภาพประกอบ 4.54 MS ให้ค่าเวลาเฉลี่ยน้อยที่สุด คือ 0.022 ms และใกล้เคียงกับ MS_BG_10 ms คือ 0.029 ms และ MST_AC คือ 0.023 ms สำหรับ MS_ICA ให้ค่าเวลาเฉลี่ยมากที่สุด คือ 0.25 ms



ภาพประกอบ 4.54 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล Airport1_new

4.5.4 ค่าระยะทางผิดพลาดเฉลี่ย

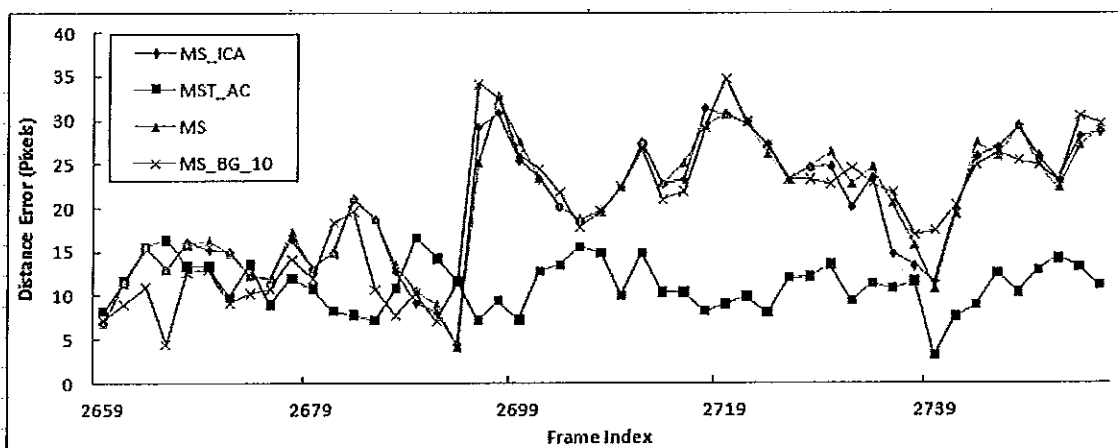
ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport2_new จากภาพประกอบ 4.55 จะเห็นได้ว่า MST_AC ให้ค่าระยะทางผิดพลาดน้อยที่สุด คือ 11.18 MS ให้ค่า 20.37 MS_BG_10 ให้ค่า 19.77 และ MS_ICA ให้ค่า 20.09



ภาพประกอบ 4.55 ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 2_new

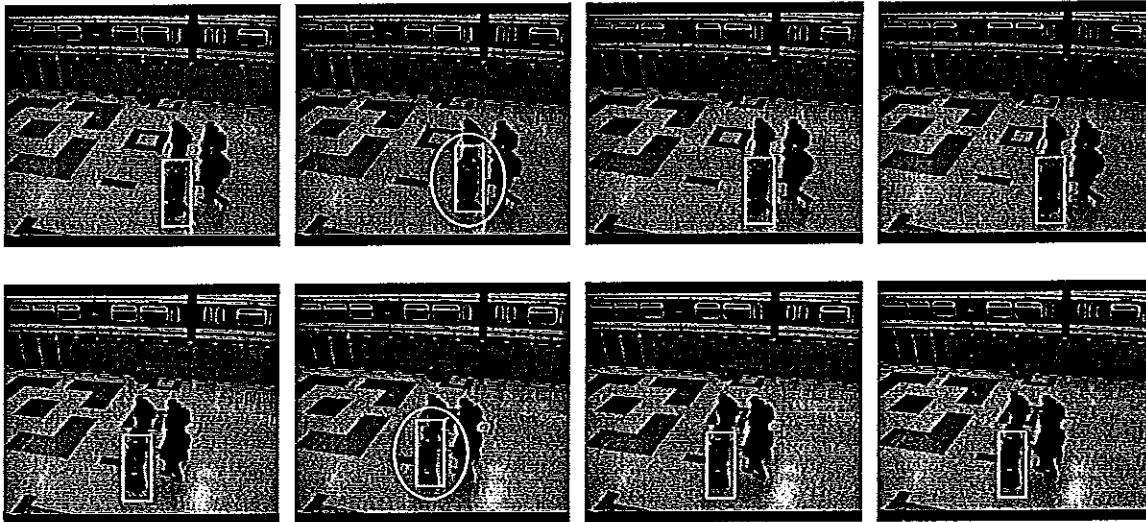
4.5.5 ค่าระยะทางผิดพลาด

ค่าระยะทางผิดพลาดของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport2_new จากภาพประกอบ 4.56 จะเห็นว่าจุดส่วนใหญ่ของขั้นตอนวิธี MST_AC จะอยู่ต่ำกว่าขั้นตอนวิธีอื่น ๆ



ภาพประกอบ 4.56 ค่าเวลาเฉลี่ยต่อการวนรอบในแต่ละขั้นตอนวิธีสำหรับข้อมูล Airport 2_new

ตัวอย่างการติดตามที่ดีของ MST_AC จากภาพประกอบ 4.57 ในเฟรมที่ 2699 และ 2719 จะเห็นว่าขั้นตอนวิธี MST_AC ให้ความถูกต้องในการติดตามมากกว่าขั้นตอนวิธี MS_ICA MS และ MS_BG_10 สังเกตจากวงกลมข้างล่าง



MS_ICA

MST_AC

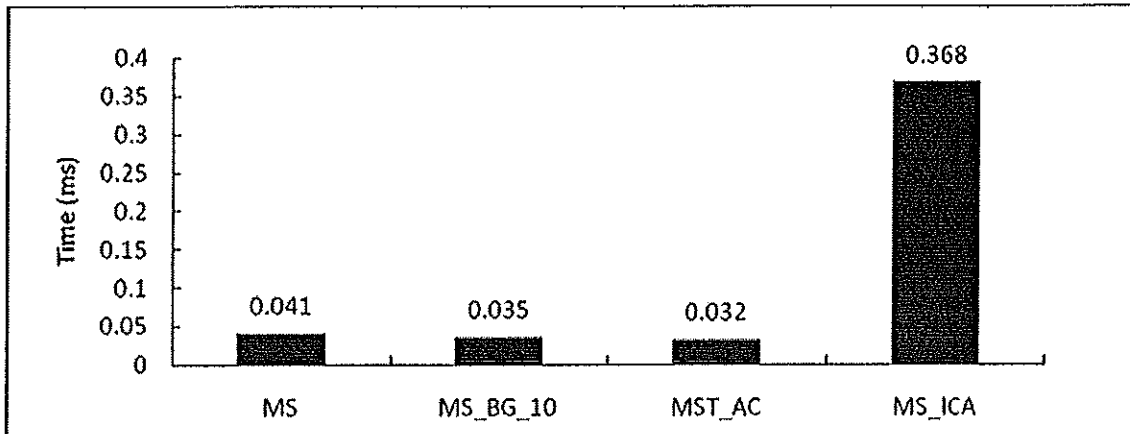
MS

MS_BG_10

ภาพประกอบ 4.57 ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC ลำดับเฟรม 2699 และ 2719 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล Airport2_new

4.5.6 ค่าเวลาเฉลี่ย

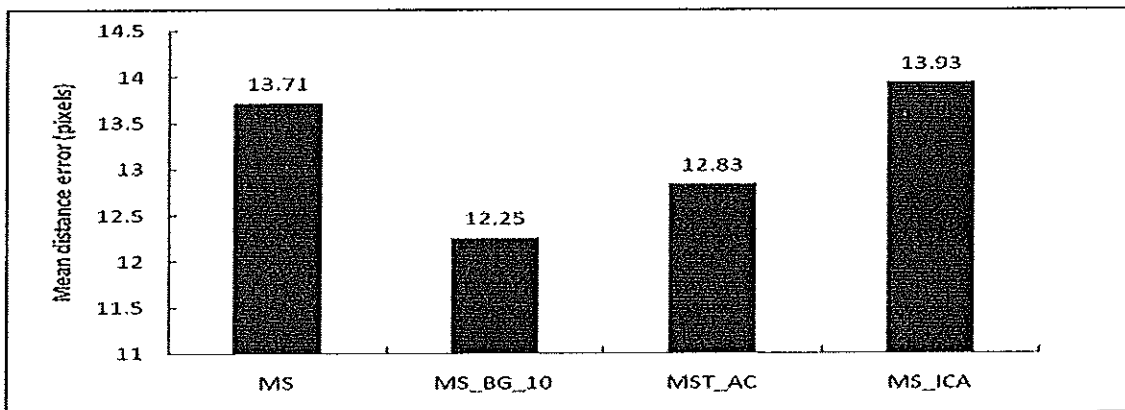
ค่าเวลาเฉลี่ยในการทำงาน 1 รอบ ของชุดข้อมูล Airport2_new จากภาพประกอบ 4.58 MST_AC ให้ค่าเวลาเฉลี่ยน้อยที่สุด คือ 0.032 ms และใกล้เคียงกับ MS คือ 0.041 ms และ MS_BG_10 คือ 0.035 ms สำหรับ MS_ICA ให้ค่าเวลาเฉลี่ยมากที่สุด คือ 0.368 ms



ภาพประกอบ 4.58 ค่าเวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล Airport2_new

4.5.7 ค่าระยะทางผิดพลาดเฉลี่ย

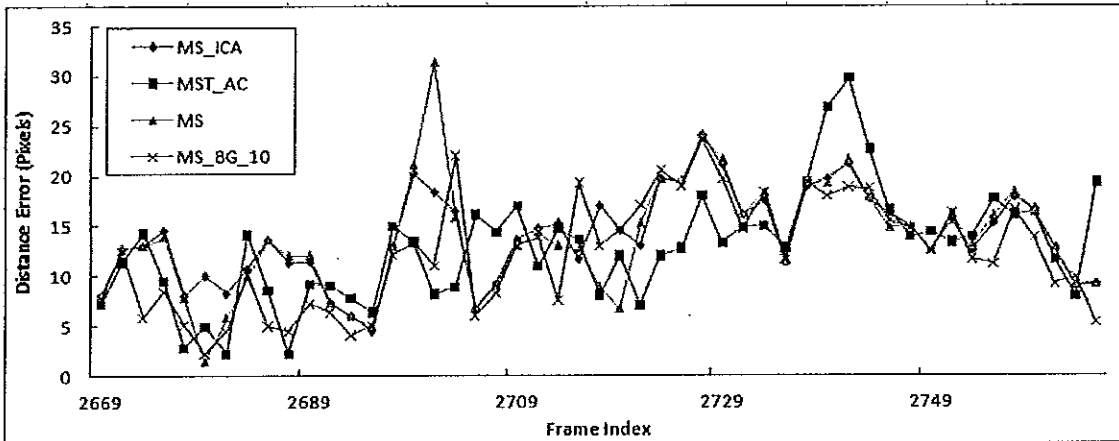
ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport3_new จากภาพประกอบ 4.59 จะเห็นได้ว่า MS_BG_10 ให้ค่าระยะทางผิดพลาดน้อยที่สุด คือ 12.25 MS ให้ค่า 13.71 MST_AC ให้ค่า 12.83 และ MS_ICA ให้ค่า 13.93



ภาพประกอบ 4.59 ค่าระยะทางผิดพลาดเฉลี่ยในแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 3_new

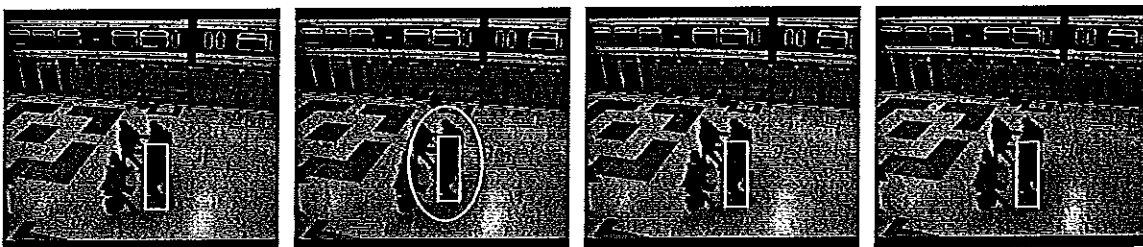
4.5.8 ค่าระยะทางผิดพลาด

ค่าระยะทางผิดพลาดของแต่ละขั้นตอนวิธีโดยใช้ข้อมูลทดสอบ Airport3_new จากภาพประกอบ 4.60 จะเห็นว่าจุดส่วนใหญ่ของขั้นตอนวิธี MST_AC จะอยู่ใกล้เคียงกับขั้นตอนวิธีอื่น ๆ



ภาพประกอบ 4.60 ค่าระยะทางผิดพลาดในแต่ละขั้นตอนวิธี โดยใช้ข้อมูลทดสอบ Airport 3_new

ตัวอย่างการติดตามที่ดีของ MST_AC จากภาพประกอบ 4.61 ในเฟรมที่ 2729 จะเห็นได้ว่าขั้นตอนวิธี MST_AC ให้ความถูกต้องในการติดตามมากกว่าขั้นตอนวิธี MS MS_BG_10 และ MS_ICA สังเกตจากวงกลมข้างล่าง



MS_ICA

MST_AC

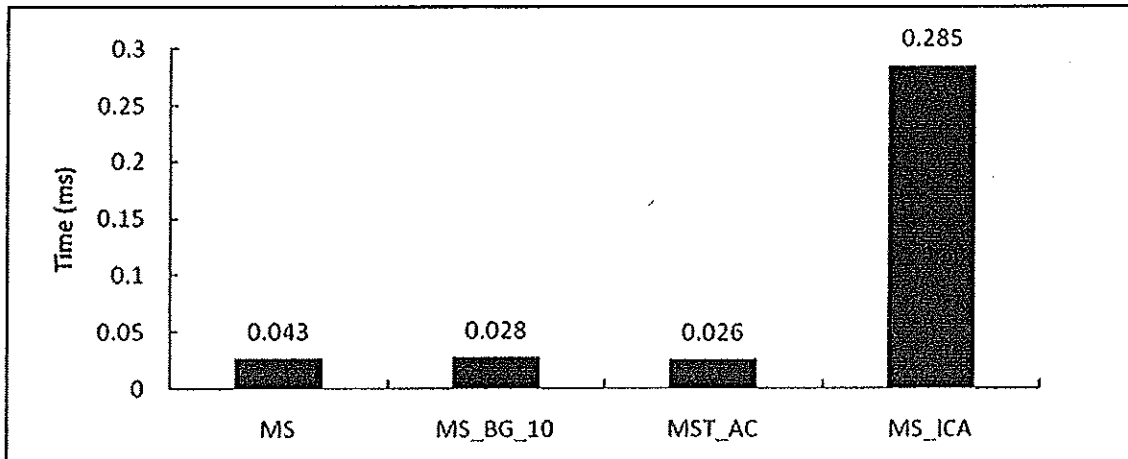
MS

MS_BG_10

ภาพประกอบ 4.61 ตัวอย่างการติดตามที่ดีของขั้นตอนวิธี MST_AC ลำดับเฟรม 2729 ตามแนวตั้งจากบนลงล่าง โดยใช้ชุดข้อมูล Airport3_new

4.5.9 ค่าเวลาเฉลี่ย

ค่าเวลาเฉลี่ยในการทำงาน 1 รอบ ของชุดข้อมูล Airport3_new จากภาพประกอบ 4.62 MST_AC ให้ค่าเวลาเฉลี่ยน้อยที่สุด คือ 0.026 ms และใกล้เคียงกับ MS คือ 0.043 ms และ MS_BG_10 คือ 0.028 ms สำหรับ MS_ICA ให้ค่าเวลาเฉลี่ยมากที่สุด คือ 0.285 ms



ภาพประกอบ 4.62 ค่าเวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธี โดยใช้ชุดข้อมูล Airport3_new

4.6 เปรียบเทียบผลการทดลองและวิจารณ์ผลการทดลอง

4.6.1 เปรียบเทียบผลการทดลองและวิจารณ์

ผลการทดลองการติดตามของแต่ละขั้นตอนวิธี เมื่อใช้ข้อมูลทดสอบ Airport 1 Airport 2 Airport 3 Airport 4 ComSci 1 ComSci 2 ComSci 3 และ ComSci 4 โดยในการทดสอบใช้วิธีการหาค่าระยะทางผิดพลาดเฉลี่ยเปรียบเทียบผลการทดลอง ซึ่งผลลัพธ์แสดงดังตารางที่ 4.3 จะได้ว่า MST_AC ให้ค่าระยะทางผิดพลาดน้อยที่สุด เมื่อเปรียบเทียบกับขั้นตอนวิธีอื่น ๆ ในข้อมูลทดสอบ Airport 1 Airport 2 Airport 3 Airport 4 ComSci 1 ComSci 2 และ ComSci 4 ในเรื่องของเวลาเฉลี่ยที่ใช้ในการทำงาน ดังตารางที่ 4.4 จะเห็นว่า MST_AC ใช้เวลาในการทำงานน้อยและใกล้เคียงกับขั้นตอนวิธี MS_BG_10 และ MS สำหรับ MS_ICA ใช้เวลาในการทำงานมากที่สุด

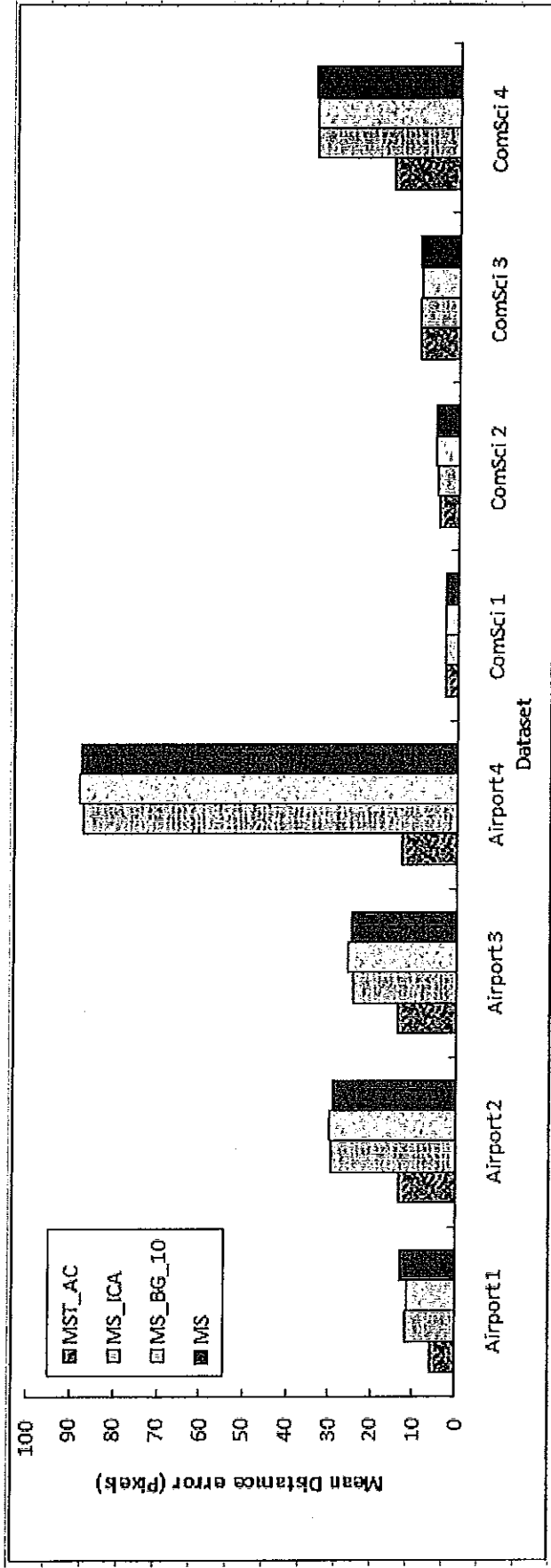
ตารางที่ 4.3 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธี

	Airport 1	Airport 2	Airport 3	Airport 4	ComSci 1	ComSci 2	ComSci 3	ComSci 4
MST_AC	6.08	13.96	14.05	13.54	3.10	4.89	9.36	15.76
MS_ICA	11.94	29.82	24.61	87.57	3.22	5.34	9.57	34.02
MS_BG_10	11.79	29.30	26.17	88.56	3.14	5.65	9.23	34.07
MS	13.13	30.24	24.92	87.76	3.12	5.64	9.49	34.35

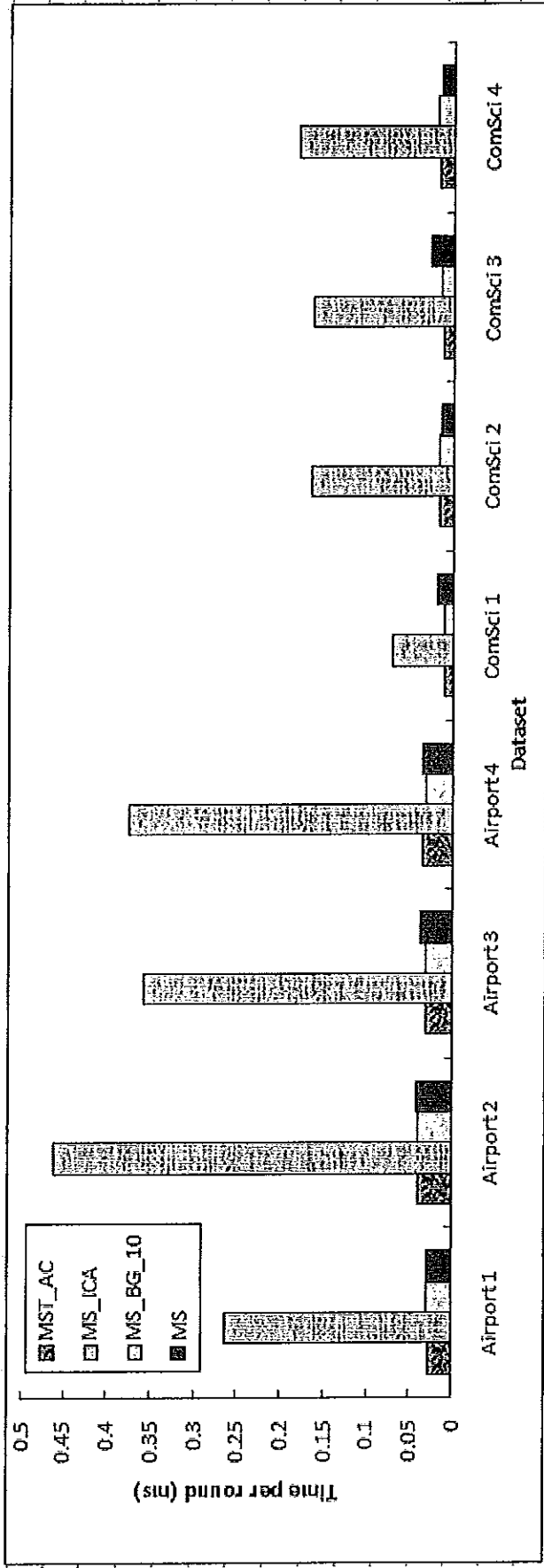
ตารางที่ 4.4 เวลาเฉลี่ยที่ใช้ต่อการทำงาน 1 รอบ (ms) ของแต่ละขั้นตอนวิธี

	Airport1	Airport 2	Airport 3	Airport 4	ComSci 1	ComSci 2	ComSci 3	ComSci 4
MST_AC	0.028	0.041	0.033	0.037	0.012	0.018	0.015	0.018
MS_JCA	0.265	0.463	0.359	0.377	0.074	0.168	0.165	0.182
MS_BG_10	0.031	0.042	0.032	0.033	0.013	0.019	0.017	0.020
MS	0.030	0.043	0.039	0.035	0.019	0.014	0.027	0.014

ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีในแต่ละข้อมูลทดสอบ ดังภาพประกอบ 4.63 และเวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธีในแต่ละข้อมูลทดสอบดังภาพประกอบ 4.64



ภาพประกอบ 4.63 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีในแต่ละชุดทดสอบ



ภาพประกอบ 4.64 เวลาเฉลี่ยต่อการทำงาน 1 รอบในแต่ละขั้นตอนวิธีในแต่ละชุดทดสอบ

4.6.2 เปรียบเทียบผลการทดลองและวิจารณ์ขนาด bin 32x32x32 และ 16x16x16

ผลการทดลองการติดตามของแต่ละขั้นตอนวิธีโดยปรับขนาด bin จาก 32x32x32 เป็น 16x16x16 เมื่อใช้ข้อมูลทดสอบ Airport 1 Airport 2 และ Airport 3 โดยในการทดสอบใช้วิธีการหาค่าระยะทางผิดพลาดเฉลี่ยเปรียบเทียบผลการทดลอง ซึ่งผลลัพธ์แสดงดังตารางที่ 4.5 จะได้ว่า ชุดข้อมูลทดสอบ Airport 1 ขนาด bin 32x32x32 ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยกว่าขนาด bin 16x16x16 ตามขั้นตอนวิธี MST_AC MS_ICA MS_BG_10 และ MS ซึ่งให้ค่าดังนี้ 6.08 11.94 11.79 และ 13.13 ในชุดข้อมูล Airport 2 ขนาด bin 16x16x16 ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยกว่าขนาด bin 32x32x32 ตามขั้นตอนวิธี MST_AC MS_ICA MS_BG_10 และ MS ซึ่งให้ค่าดังนี้ 10.47 28.57 27.08 และ 28.47 ในชุดข้อมูล Airport 3 ขนาด bin 32x32x32 ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยกว่าขนาด bin 16x16x16 ในขั้นตอนวิธี MST_AC และ MS_BG_10 ซึ่งให้ค่า 14.05 และ 26.17 ในเรื่องของเวลาเฉลี่ยที่ใช้ในการทำงาน 1 รอบในแต่ละขั้นตอนวิธี ดังตาราง 4.6 ในชุดข้อมูล Airport1 เวลาเฉลี่ยของขนาด bin 16x16x16 ใช้เวลาเฉลี่ยน้อยกว่าขนาด bin 32x32x32 ในขั้นตอนวิธี MST_AC MS_ICA MS_BG_10 และ MS ซึ่งให้ค่าดังนี้ 0.018 ms 0.253 ms 0.024 ms และ 0.022 ms ในชุดข้อมูล Airport 2 เวลาส่วนใหญ่ของขนาด bin 16x16x16 ใช้เวลาเฉลี่ยน้อยกว่าขนาด bin 32x32x32 ในขั้นตอนวิธี MST_AC MS_ICA MS_BG_10 และ MS ซึ่งให้ค่าดังนี้ 0.036 ms 0.438 ms 0.034 ms และ 0.044 ms และในชุดข้อมูล Airport 3 เวลาส่วนใหญ่ของขนาด bin 16x16x16 ใช้เวลาเฉลี่ยน้อยกว่าขนาด bin 32x32x32 ในขั้นตอนวิธี MST_AC MS_ICA MS_BG_10 และ MS ซึ่งให้ค่าดังนี้ 0.031 ms 0.35 ms 0.026 ms และ 0.045 ms

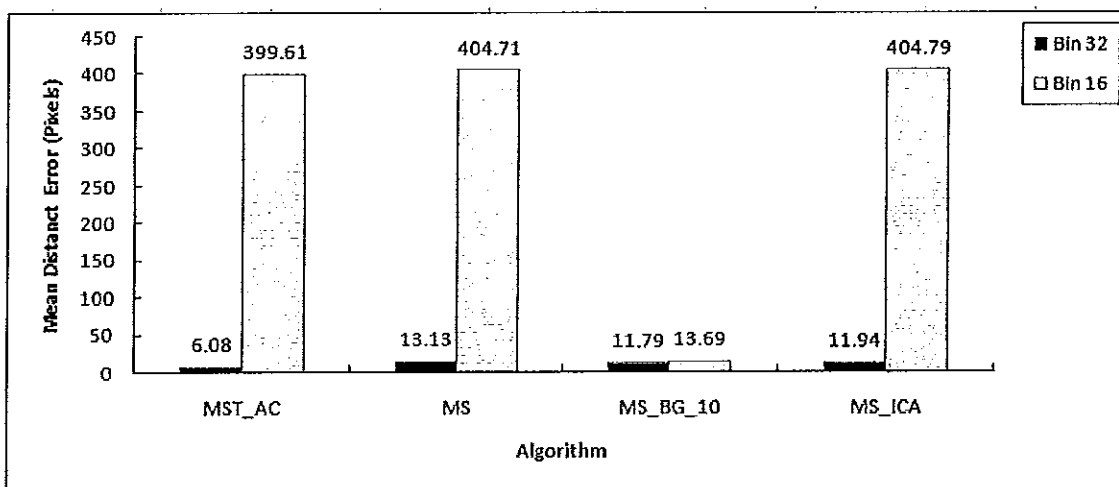
ตารางที่ 4.5 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธี

ขนาด bin	Airport 1		Airport 2		Airport 3	
	16x16x16	32x32x32	16x16x16	32x32x32	16x16x16	32x32x32
MST_AC	399.61	6.08	10.47	13.96	25.58	14.05
MS_ICA	404.79	11.94	28.57	29.82	24.34	24.61
MS_BG_10	13.69	11.79	27.08	29.30	26.46	26.17
MS	404.71	13.13	28.47	30.24	24.17	24.92

ตารางที่ 4.6 เวลาเฉลี่ยที่ใช้ต่อการทำงาน 1 รอบของแต่ละขั้นตอนวิธี

	Airport 1 (ms)		Airport 2 (ms)		Airport 3 (ms)	
	16x16x16	32x32x32	16x16x16	32x32x32	16x16x16	32x32x32
MST_AC	0.018	0.028	0.036	0.041	0.031	0.033
MS_ICA	0.253	0.265	0.438	0.463	0.35	0.359
MS_BG_10	0.024	0.031	0.034	0.042	0.026	0.032
MS	0.022	0.03	0.044	0.043	0.045	0.039

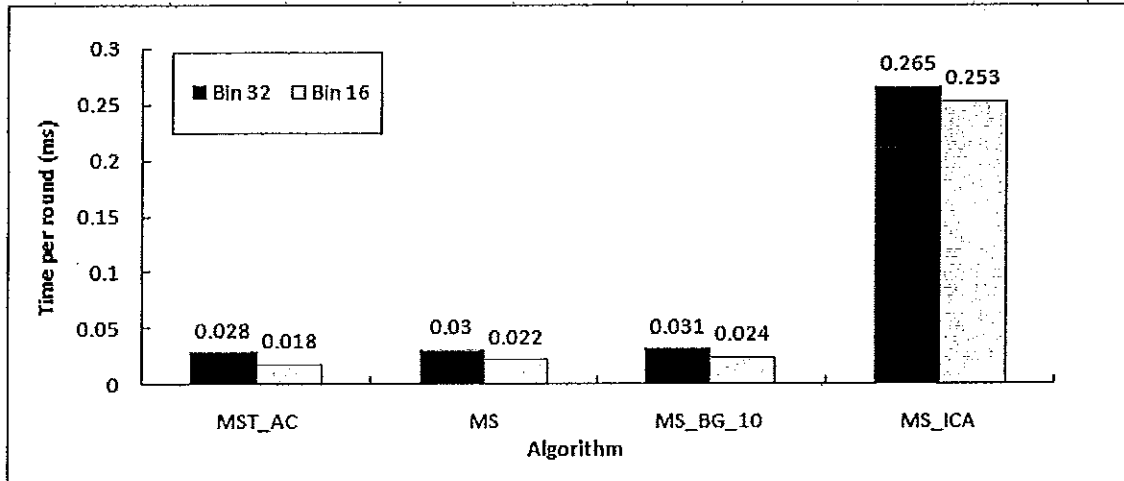
ค่าระยะทางผิดพลาดของแต่ละขั้นตอนวิธีโดยเปรียบเทียบขนาด bin 32x32x32 กับขนาด bin 16x16x16 โดยใช้ข้อมูลทดสอบ Airport 1 ดังภาพประกอบ 4.65 จะเห็นว่า โดยใช้ขนาด bin 32x32x32 ให้ค่าระยะทางผิดพลาดน้อยกว่าใช้ขนาด bin 16x16x16 ตามขั้นตอนวิธี MST_AC MS MS_BG_10 และ MS_ICA ซึ่งให้ค่าดังนี้ 6.08 13.13 11.79 และ 11.94



ภาพประกอบ 4.65 ค่าระยะทางผิดพลาดของแต่ละขั้นตอนวิธีโดยเปรียบเทียบขนาด bin 32x32x32 กับขนาด bin 16x16x16 โดยใช้ข้อมูลทดสอบ Airport 1

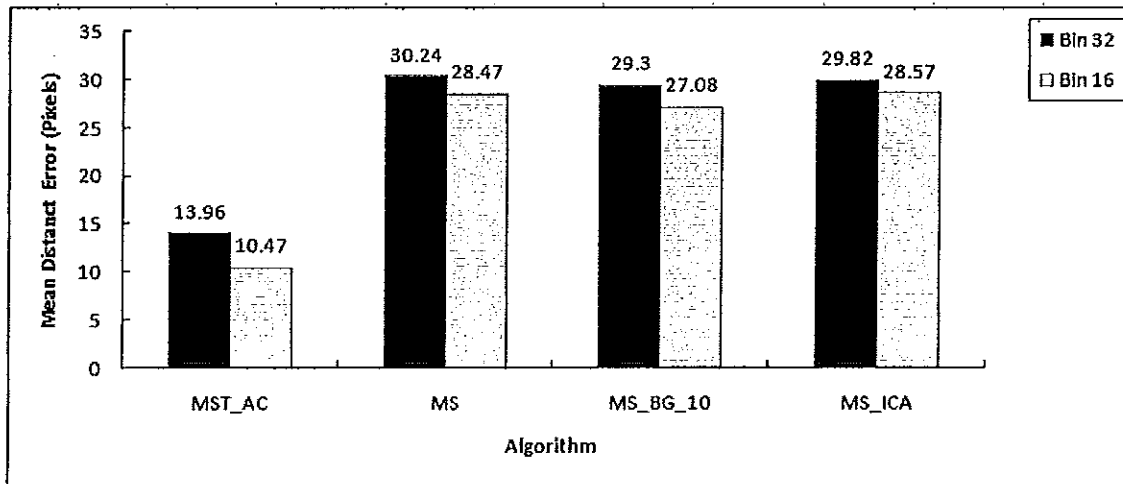
การเปรียบเทียบค่าเวลาเฉลี่ยของแต่ละขั้นตอนวิธีโดยเปรียบเทียบขนาด bin 32x32x32 กับขนาด bin 16x16x16 โดยใช้ข้อมูลทดสอบ Airport 1 แสดงได้ดังภาพประกอบ 4.66 เวลาเฉลี่ยของขนาด bin 16x16x16 ใช้เวลาเฉลี่ยน้อยกว่าขนาด bin 32x32x32 ตาม

ขั้นตอนวิธี MST_AC MS MS_BG_10 และ MS_ICA ซึ่งให้ค่าดังนี้ 0.018 ms 0.022 ms 0.024 ms และ 0.253 ms



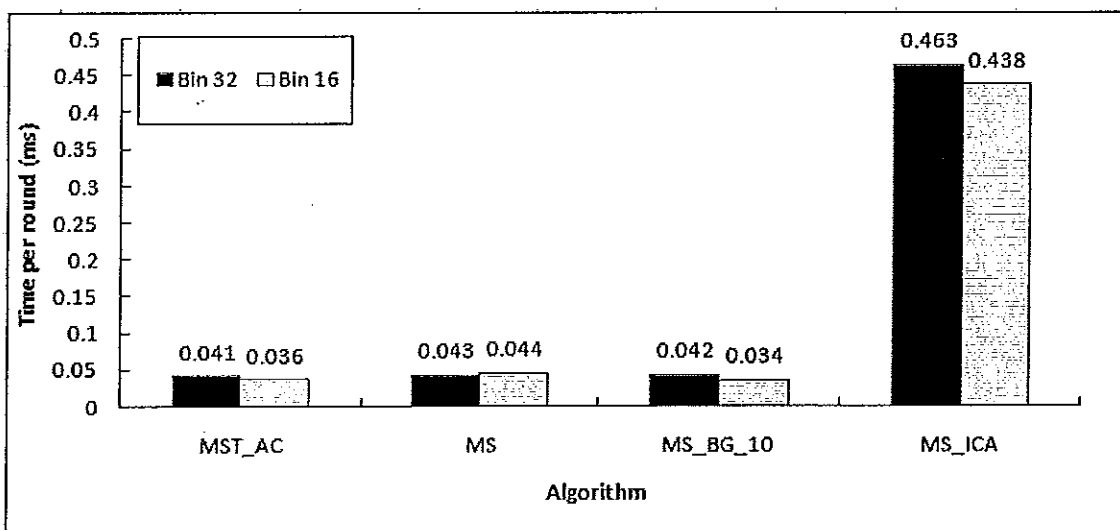
ภาพประกอบ 4.66 ค่าเวลาเฉลี่ยของแต่ละขั้นตอนวิธีโดยเปรียบเทียบขนาด bin 32x32x32 กับขนาด bin 16x16x16 โดยใช้ข้อมูลทดสอบ Airport 1

ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยเปรียบเทียบขนาด bin 32x32x32 กับขนาด bin 16x16x16 โดยใช้ข้อมูลทดสอบ Airport 2 ดังภาพประกอบ 4.67 จะเห็นว่า โดยใช้ขนาด bin 16x16x16 ให้ค่าระยะทางผิดพลาดน้อยกว่าใช้ขนาด bin 32x32x32 ตามขั้นตอนวิธี MST_AC MS MS_BG_10 และ MS_ICA ซึ่งให้ค่าดังนี้ 10.47 28.47 27.08 และ 28.57



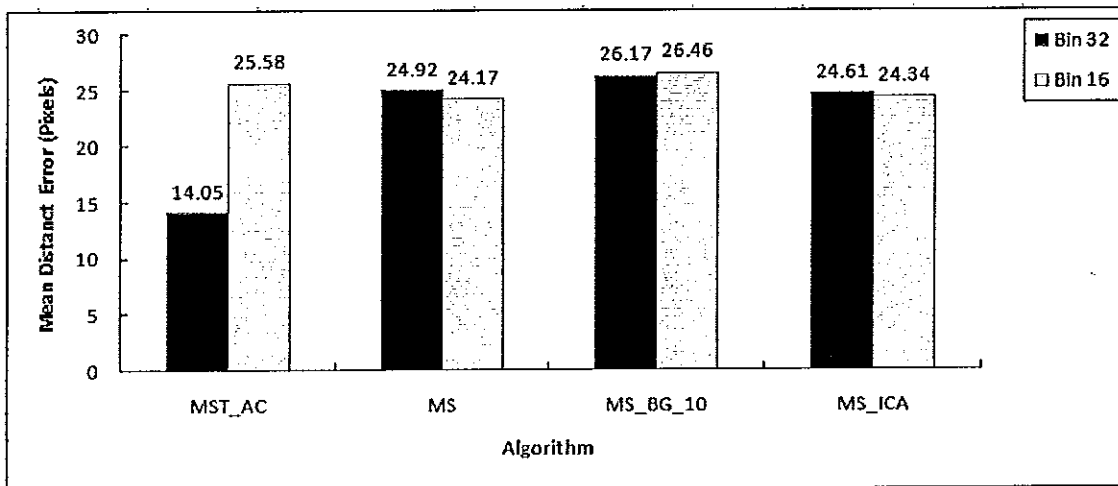
ภาพประกอบ 4.67 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยเปรียบเทียบขนาด bin 32x32x32 กับขนาด bin 16x16x16 โดยใช้ข้อมูลทดสอบ Airport 2

การเปรียบเทียบค่าเวลาเฉลี่ยของแต่ละขั้นตอนวิธีโดยเปรียบเทียบขนาด bin 32x32x32 กับขนาด bin 16x16x16 โดยใช้ข้อมูลทดสอบ Airport 2 แสดงได้ดังภาพประกอบ 4.68 เวลาเฉลี่ยของขนาด bin 16x16x16 ส่วนใหญ่ใช้เวลาเฉลี่ยน้อยกว่าขนาด bin 32x32x32 ตามขั้นตอนวิธี MST_AC MS MS_BG_10 และ MS_ICA ซึ่งให้ค่าดังนี้ 0.036 ms 0.044 ms 0.034 ms และ 0.438 ms



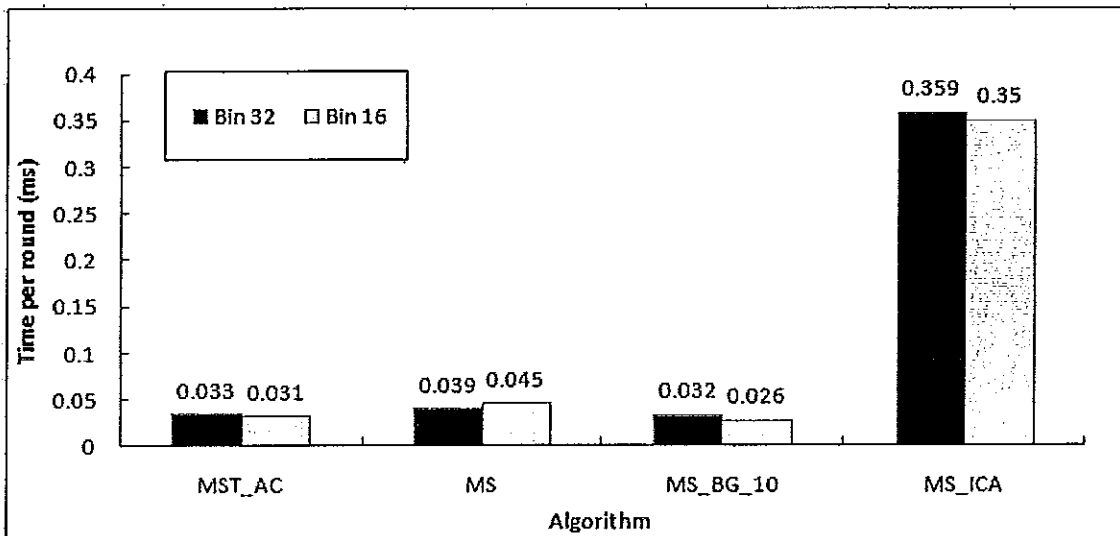
ภาพประกอบ 4.68 ค่าเวลาเฉลี่ยของแต่ละขั้นตอนวิธีโดยเปรียบเทียบขนาด bin 32x32x32 กับขนาด bin 16x16x16 โดยใช้ข้อมูลทดสอบ Airport 2

ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยเปรียบเทียบขนาด bin 32x32x32 กับขนาด bin 16x16x16 โดยใช้ข้อมูลทดสอบ Airport 3 ดังภาพประกอบ 4.69 จะเห็นว่า โดยใช้ขนาด bin 32x32x32 ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยกว่าใช้ขนาด bin 16x16x16 สำหรับขั้นตอนวิธี MST_AC



ภาพประกอบ 4.69 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยเปรียบเทียบขนาด bin 32x32x32 กับขนาด bin 16x16x16 โดยใช้ข้อมูลทดสอบ Airport 3

การเปรียบเทียบค่าเวลาเฉลี่ยของแต่ละขั้นตอนวิธีโดยเปรียบเทียบขนาด bin 32x32x32 กับขนาด bin 16x16x16 โดยใช้ข้อมูลทดสอบ Airport 3 แสดงได้ดังภาพประกอบ 4.70 เวลาเฉลี่ยของขนาด bin 16x16x16 ส่วนใหญ่ใช้เวลาเฉลี่ยน้อยกว่าขนาด bin 32x32x32 ตามขั้นตอนวิธี MST_AC MS MS_BG_10 และ MS_ICA ซึ่งให้ค่าดังนี้ 0.031 ms 0.045 ms 0.026 ms และ 0.35 ms



ภาพประกอบ 4.70 ค่าเวลาเฉลี่ยของแต่ละขั้นตอนวิธีโดยเปรียบเทียบขนาด bin 32x32x32 กับขนาด bin 16x16x16 โดยใช้ข้อมูลทดสอบ Airport 3

4.6.3 เปรียบเทียบผลการทดลองและวิจารณ์โดยเปลี่ยนความคมชัดของข้อมูลทดสอบ

ในการทดลองนี้เพื่อเปรียบเทียบความถูกต้องในการติดตามโดยดูจากค่าระยะทางผิดพลาดเฉลี่ยของขั้นตอนวิธี เมื่อเปลี่ยนความคมชัดของวิดีโอ การทดลองใช้ขั้นตอนวิธี MST_AC MS MS_BG_10 และ MS_ICA ชุดข้อมูลทดสอบที่ใช้ คือ Airport1_new Airport2_new และ Airport3_new โดยคุณสมบัติของชุดข้อมูลใหม่จำนวน 100 เฟรมที่มีค่าความคมชัดต่ำ (ภาพแตก) เปรียบเทียบกับข้อมูลชุดเก่า คือ Airport1 Airport2 และ Airport3 โดยคุณสมบัติของชุดข้อมูลเก่าจำนวน 100 เฟรมที่มีค่าความคมชัดสูง (ภาพปกติ) ดังตาราง 4.2 ค่าระยะทางผิดพลาดแสดงได้ดังตารางที่ 4.7 ในชุดข้อมูลแรก (Airport1_new และ Airport1) Airport1 ส่วนใหญ่ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยกว่า Airport1_new ตามขั้นตอนวิธี MST_AC MS_ICA MS_BG_10 และ MS ซึ่งให้ค่า 10.84 18.68 15.61 และ 19.07 ในชุดข้อมูลที่ 2 (Airport2_new และ Airport2) Airport2_new ส่วนใหญ่ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยกว่า Airport2 ตามขั้นตอนวิธี MST_AC MS_ICA MS_BG_10 และ MS ซึ่งให้ค่า 11.18 20.09 19.77 และ 20.37 และชุดข้อมูลที่ 3 (Airport3_new และ Airport3) Airport3_new ส่วนใหญ่ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยกว่า Airport3 ตามขั้นตอนวิธี MST_AC MS_ICA MS_BG_10 และ MS ซึ่งให้ค่า 12.83 13.93 12.25 และ 13.71 ในเรื่องของเวลาเฉลี่ยที่ใช้ในการทำงาน 1 รอบ ในแต่ละขั้นตอนวิธี ดังตาราง 4.8 ในชุดข้อมูลแรก (Airport1_new และ Airport1) Airport1_new ใช้เวลาเฉลี่ยน้อยกว่า Airport1 ตามขั้นตอนวิธี MST_AC MS_ICA MS_BG_10 และ MS ซึ่งให้ค่า 0.023 ms 0.25 ms 0.029 ms และ 0.022 ms ในชุดข้อมูลที่ 2

(Airport2_new และ Airport2) Airport2_new ใช้เวลาเฉลี่ยน้อยกว่า Airport2 ตามขั้นตอนวิธี MST_AC MS_ICA MS_BG_10 และ MS ซึ่งให้ค่า 0.032 ms 0.368 ms 0.035 ms และ 0.041 ms และชุดข้อมูลที่ 3 (Airpot3_new และ Airport3) Airport3_new ส่วนใหญ่ใช้เวลาเฉลี่ยน้อยกว่า Airport3 ตามขั้นตอนวิธี MST_AC MS_ICA MS_BG_10 และ MS ซึ่งให้ค่า 0.026 ms 0.285 ms 0.028 ms และ 0.043 ms

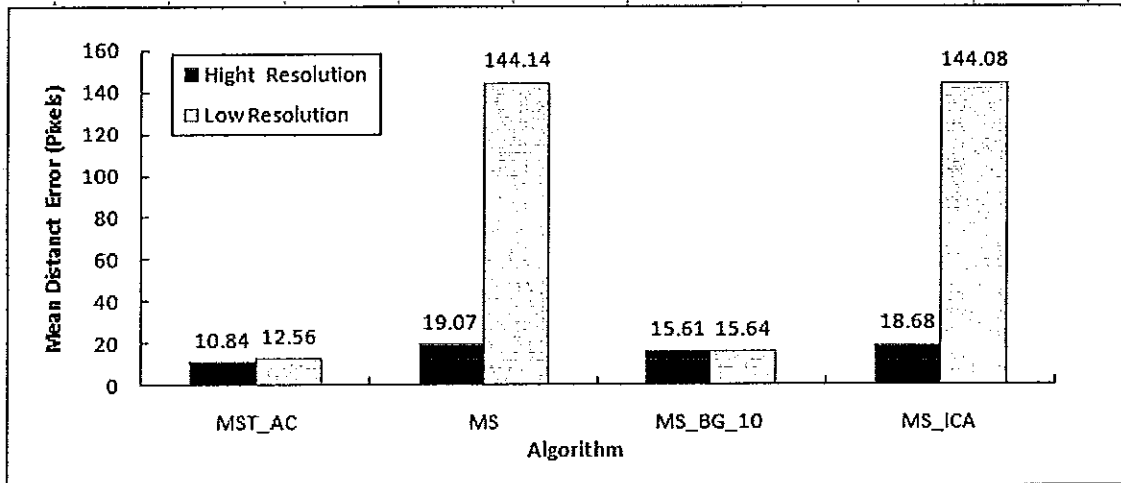
ตารางที่ 4.7 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธี

	Airport1_new	Airport1	Airport2_new	Airport2	Airport3_new	Airport3
MST_AC	12.56	10.84	11.18	12.04	12.83	10.36
MS_ICA	144.08	18.68	20.09	25.68	13.93	23.90
MS_BG_10	15.64	15.61	19.77	25.05	12.25	24.94
MS	144.14	19.07	20.37	26.36	13.71	24.57

ตารางที่ 4.8 เวลาเฉลี่ยที่ใช้ต่อการทำงาน 1 รอบของแต่ละขั้นตอนวิธี

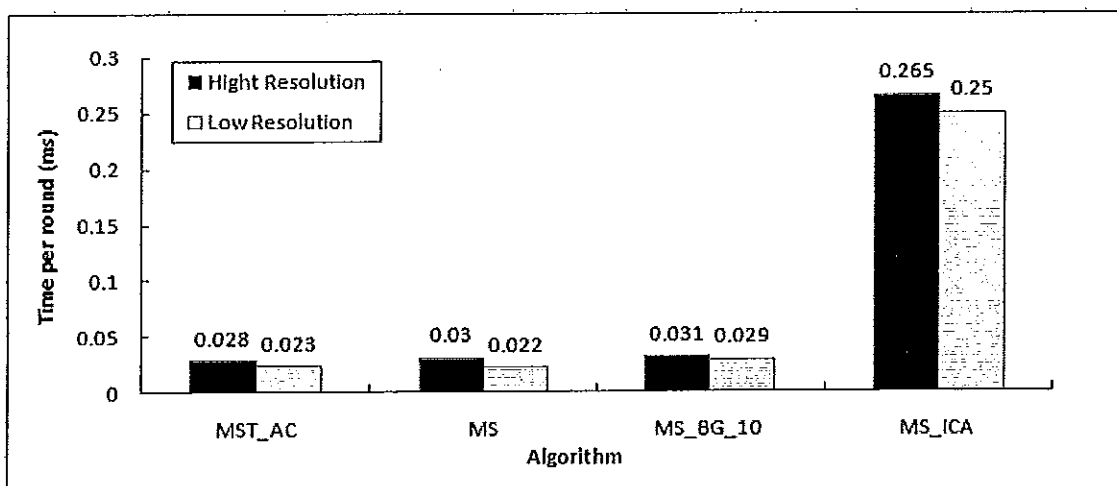
	Airport1_new (ms)	Airport1 (ms)	Airport2_new (ms)	Airport2 (ms)	Airport3_new (ms)	Airport3 (ms)
MST_AC	0.023	0.028	0.032	0.041	0.026	0.033
MS_ICA	0.25	0.285	0.368	0.463	0.285	0.359
MS_BG_10	0.029	0.031	0.035	0.042	0.028	0.032
MS	0.022	0.03	0.041	0.043	0.043	0.039

ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยมีการปรับเปลี่ยนความคมชัดของวิดีโอ โดยใช้ข้อมูลทดสอบ Airport1_new และ Airport1 ดังภาพประกอบ 4.71 จะเห็นว่า Airport1 ส่วนใหญ่ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยกว่า Airport1_new ตามขั้นตอนวิธี MST_AC MS MS_BG_10 และ MS ซึ่งให้ค่า 10.84 19.07 15.61 และ 18.68



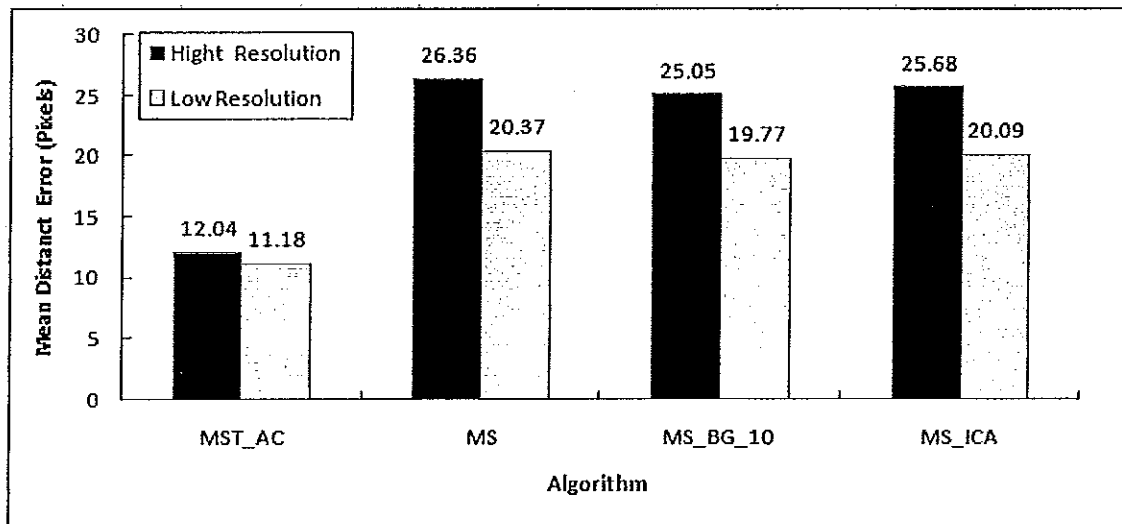
ภาพประกอบ 4.71 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยมีการปรับเปลี่ยนความคมชัดของวิดีโอ โดยใช้ข้อมูลทดสอบ Airport1_new และ Airport1

การเปรียบเทียบค่าเวลาเฉลี่ยของแต่ละขั้นตอนวิธีโดยมีการปรับเปลี่ยนความคมชัดของวิดีโอ โดยใช้ข้อมูลทดสอบ Airport1_new และ Airport1 แสดงได้ดังภาพประกอบ 4.72 Airport1_new ใช้เวลาเฉลี่ยน้อยกว่า Airport1 ตามขั้นตอนวิธี MST_AC MS MS_BG_10 และ MS_ICA ซึ่งให้ค่า 0.023 ms 0.022 ms 0.029 ms และ 0.25 ms



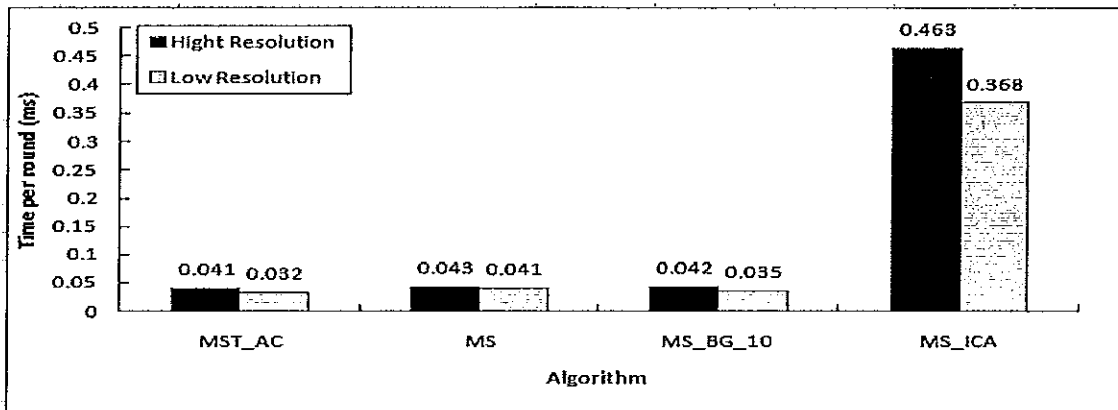
ภาพประกอบ 4.72 ค่าเวลาเฉลี่ยของแต่ละขั้นตอนวิธีโดยมีการปรับเปลี่ยนความคมชัดของวิดีโอ โดยใช้ข้อมูลทดสอบ Airport1_new และ Airport1

ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยมีการปรับเปลี่ยนความคมชัดของวิดีโอ โดยใช้ข้อมูลทดสอบ Airport2_new และ Airport2 ดังภาพประกอบ 4.73 จะเห็นว่า Airport2_new ส่วนใหญ่ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยกว่า Airport2 ตามขั้นตอนวิธี MST_AC MS MS_BG_10 และ MS_ICA ซึ่งให้ค่า 11.18 20.37 19.77 และ 20.09



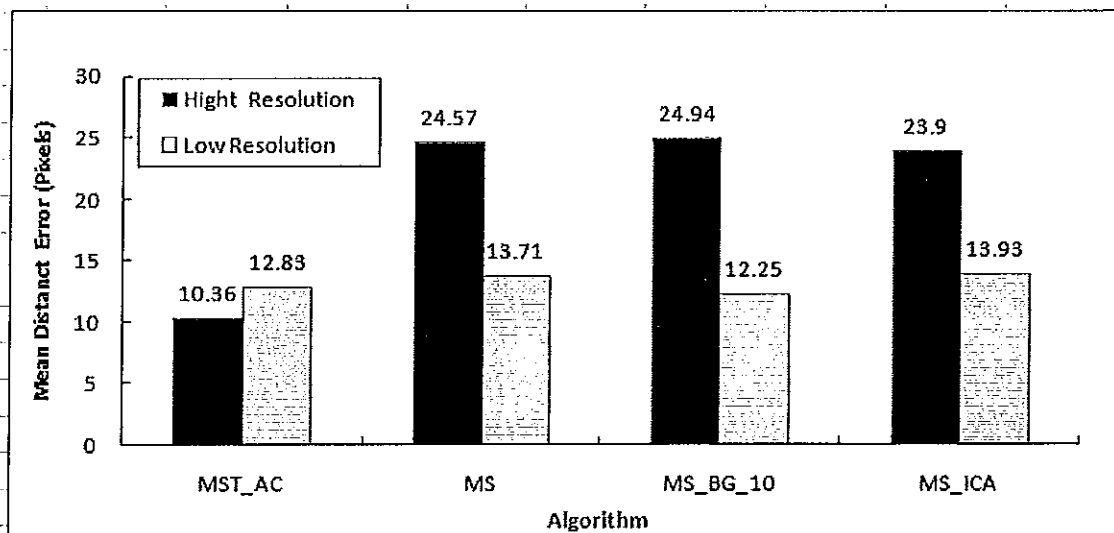
ภาพประกอบ 4.73 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยมีการปรับเปลี่ยนความคมชัดของวิดีโอ โดยใช้ข้อมูลทดสอบ Airport2_new และ Airport2

การเปรียบเทียบค่าเวลาเฉลี่ยของแต่ละขั้นตอนวิธีโดยมีการปรับเปลี่ยนความคมชัดของวิดีโอ โดยใช้ข้อมูลทดสอบ Airport2_new และ Airport2 แสดงได้ดังภาพประกอบ 4.74 Airport2_new ใช้เวลาเฉลี่ยน้อยกว่า Airport2 ตามขั้นตอนวิธี MST_AC MS MS_BG_10 และ MS_ICA ซึ่งให้ค่า 0.032 ms 0.041 ms 0.035 ms และ 0.368 ms



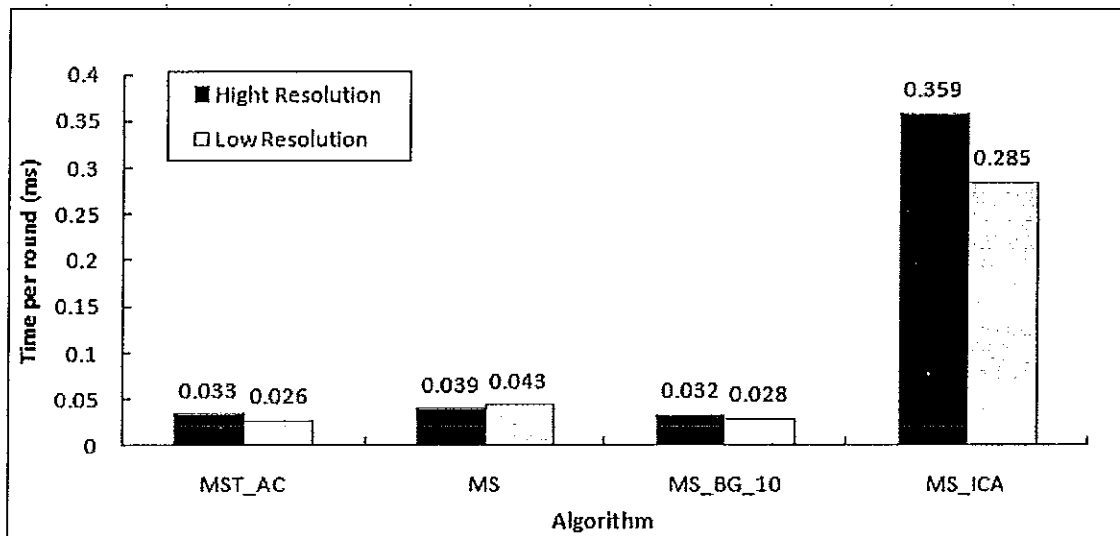
ภาพประกอบ 4.74 ค่าเวลาเฉลี่ยของแต่ละขั้นตอนวิธีโดยมีการปรับเปลี่ยนความคมชัดของวิดีโอ โดยใช้ข้อมูลทดสอบ Airport2_new และ Airport2

ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยมีการปรับเปลี่ยนความคมชัดของวิดีโอ โดยใช้ข้อมูลทดสอบ Airport3_new และ Airport3 ดังภาพประกอบ 4.75 จะเห็นว่า Airport3_new ส่วนใหญ่ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยกว่า Airport2 ตามขั้นตอนวิธี MST_AC MS MS_BG_10 และ MS_ICA ซึ่งให้ค่า 12.83 13.71 12.25 และ 13.93



ภาพประกอบ 4.75 ค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธีโดยมีการปรับเปลี่ยนความคมชัดของวิดีโอ โดยใช้ข้อมูลทดสอบ Airport3_new และ Airport3

การเปรียบเทียบค่าเวลาเฉลี่ยของแต่ละขั้นตอนวิธีโดยมีการปรับเปลี่ยนความคมชัดของวิดีโอ โดยใช้ข้อมูลทดสอบ Airport3_new และ Airport3 แสดงได้ดังภาพประกอบ 4.76 Airport2_new ส่วนใหญ่ใช้เวลาเฉลี่ยน้อยกว่า Airport2 ตามขั้นตอนวิธี MST_AC MS MS_BG_10 และ MS_ICA ซึ่งให้ค่า 0.026 ms 0.043 ms 0.028 ms และ 0.285 ms



ภาพประกอบ 4.76 ค่าเวลาเฉลี่ยของแต่ละขั้นตอนวิธีโดยมีการปรับเปลี่ยนความคมชัดของวิดีโอ โดยใช้ข้อมูลทดสอบ Airport3_new และ Airport3

บทที่ 5

บทสรุปและข้อเสนอแนะ

วิทยานิพนธ์นี้ได้เสนอแบบจำลอง Mean Shift Tracking Algorithm Using Adaptive Candidate โดยใช้ชื่อเรียกว่าแบบจำลอง MST_AC เพื่อใช้ในการติดตามวัตถุ โดยเฉพาะอย่างยิ่งสำหรับแก้ปัญหาวัตถุที่มีสีเดียวหรือสีใกล้เคียงกับพื้นหลัง เพื่อเพิ่มประสิทธิภาพและความถูกต้องในการติดตามวัตถุ สำหรับวิธีการทำงานของขั้นตอนวิธี MST_AC นั้นจะเก็บเฟรมสำรองก่อนหน้าและตำแหน่งศูนย์กลางก่อนหน้าเพื่อใช้ในการคำนวณตำแหน่งพื้นหลังใหม่ แบบจำลองที่นำเสนอให้ค่าระยะทางผิดพลาดน้อยและเวลาที่ใช้ในการประมวลผลน้อย

ซึ่งผลงานวิจัยได้ตีพิมพ์ในชื่อว่า "Improving of Mean Shift Tracking Algorithm Using Adaptive Candidate Model" ในงานประชุมนานาชาติ (ECTI-CON 2010) ที่จังหวัดเชียงใหม่ เมื่อวันที่ 19-21 พฤษภาคม 2553 ดังภาคผนวก ข

5.1 สรุปผลการวิจัย

ขั้นตอนวิธี MST_AC เป็นขั้นตอนวิธี ซึ่งมีองค์ประกอบสำคัญที่ช่วยเพิ่มความถูกต้องในการติดตาม 3 ส่วน คือ 1) การหาดำแหน่งพื้นหลัง 2) การให้ค่าตำแหน่งพื้นหลัง และ 3) การปรับแบบจำลองคู่แข่ง จากการวิเคราะห์และผลการทดลองจะเห็นได้ว่าขั้นตอนวิธี MST_AC มีความถูกต้องในการติดตามมากกว่าขั้นตอนวิธี MS ขั้นตอนวิธี MS_BG_10 และขั้นตอนวิธี MS_ICA ในสิ่งแวดล้อมที่พื้นหลังมีสีเดียวกันกับวัตถุเป้าหมายและสิ่งแวดล้อมที่มีการบังโดยวัตถุที่มีสีเดียวกันกับวัตถุเป้าหมาย ขั้นตอนวิธี MST_AC ใช้เวลาในการประมวลผลใกล้เคียงกับ ขั้นตอนวิธี MS ขั้นตอนวิธี MS_BG_10 และมีค่าน้อยเมื่อเปรียบเทียบกับเวลาการประมวลผลของขั้นตอนวิธี MS_ICA

เมื่อคุณลักษณะของข้อมูลในแต่ละข้อมูลทดสอบโดยใช้ขนาด bin 32x32x32 สามารถแบ่งกลุ่มคุณลักษณะของข้อมูลได้ดังนี้ ดังตารางที่ 5.1 แสดงความถูกต้องในการติดตามของ MST_AC ในแต่ละคุณลักษณะของข้อมูล

ตารางที่ 5.1 ความถูกต้องในการติดตามของ MST_AC ในแต่ละคุณลักษณะของข้อมูล

	พื้นที่วัตถุมีขนาดใกล้เคียงกับหน้าต่าง
พื้นหลังมีสีเดียวกับวัตถุเป้าหมายและไม่มีการบัง	ขั้นตอนวิธี MST_AC ให้ความถูกต้องในการติดตามดีกว่าขั้นตอนวิธีอื่น ดังตัวอย่างตามการทดลองที่ 1 ชุดข้อมูลทดสอบ Airport 1 Airport 2 และ ComSci 4
พื้นหลังมีสีเดียวกับวัตถุและวัตถุที่บังมีสีเดียวกับวัตถุเป้าหมาย	ขั้นตอนวิธี MST_AC จะให้ความถูกต้องในการติดตามมากกว่าขั้นตอนวิธีอื่น ดังตัวอย่างตามการทดลองที่ 1 ชุดข้อมูลทดสอบ Airport 3 Airport 4
พื้นหลังมีสีแตกต่างจากสีวัตถุเป้าหมายอย่างชัดเจน	ขั้นตอนวิธี MST_AC จะให้ความถูกต้องในการติดตามใกล้เคียงกับขั้นตอนวิธีอื่น ดังตัวอย่างตามการทดลองที่ 1 ชุดข้อมูลทดสอบ ComSci 1 ComSci 2 และ ComSci 3

5.1.1 เมื่อปรับขนาดของ bin

ความถูกต้องในการติดตามและเวลาการทำงานของ MST_AC เมื่อปรับเปลี่ยนขนาด bin ดังตารางที่ 5.2 ที่ใช้ ขนาด bin (16x16x16) และขนาดของ bin (32x32x32) ดังตัวอย่างตามการทดลองที่ 2

ตารางที่ 5.2 ความถูกต้องในการติดตามและเวลาการทำงานของ MST_AC เมื่อปรับเปลี่ยนขนาด bin

ความถูกต้องในการติดตาม	เวลาที่ใช้ในการทำงาน
ความถูกต้องในการติดตามขึ้นอยู่กับลักษณะของข้อมูลแต่ถ้าขนาดของ bin 32x32x32 มีโอกาสที่ได้ความถูกต้องในการติดตามมาก ดังตารางที่ 4.5	ในเรื่องของเวลาขนาด bin 16x16x16 ใช้เวลาน้อยกว่าขนาดของ bin 32x32x32 ดังตารางที่ 4.6

5.1.2 เมื่อปรับความละเอียดของภาพ

ความถูกต้องในการติดตามและเวลาการทำงานของ MST_AC เมื่อปรับเปลี่ยนความคมชัดของวิดีโอ ดังตารางที่ 5.3 ความคมชัดที่ใช้ในที่นี้คือ ความคมชัดต่ำและความคมชัดสูง ตัวอย่างตามการทดลองที่ 3

ตารางที่ 5.3 ความถูกต้องในการติดตามและเวลาการทำงานของ MST_AC เมื่อปรับเปลี่ยนความคมชัดของภาพ

ความถูกต้องในการติดตาม	เวลาที่ใช้ในการทำงาน
ความถูกต้องในการติดตามมากขึ้นเมื่อความคมชัดสูง ความถูกต้องในการติดตามลดลงเมื่อความคมชัดต่ำ ดังตารางที่ 4.7	ความคมชัดต่ำใช้เวลาการทำงานน้อยกว่าความคมชัดสูง ดังตารางที่ 4.8

5.2 ปัญหาและอุปสรรค

เนื่องจากเครื่องมือที่ใช้พัฒนาโปรแกรมที่มีชื่อว่า OpenCV เป็น library ที่พัฒนาจากภาษา Visual C++ ซึ่งรวบรวมฟังก์ชันทางด้าน Image Processing และ Computer Vision ซึ่งผู้วิจัยไม่ถนัดทางด้านภาษานี้และบวกกับเอกสารที่อธิบายการทำงานของแต่ละฟังก์ชันเขียนไม่ชัดเจน จึงทำให้เสียเวลานานสำหรับการศึกษาเครื่องมือ

5.3 ข้อเสนอแนะและงานในอนาคต

5.4.1 แบบจำลอง

การสร้างแบบจำลองการติดตามวัตถุในการทดลองนี้ใช้สีในการพิจารณาเป็นหลัก นอกจากนี้ผู้วิจัยสามารถใช้คุณสมบัติอย่างอื่นในการสร้างแบบจำลองได้ด้วย เช่น เส้นขอบ พื้นผิว เป็นต้น

5.4.2 การปรับขนาดหน้าต่างตามวัตถุเป้าหมาย

เนื่องจากการติดตามวัตถุที่นำเสนอ MST_AC ใช้กับขนาดหน้าต่างคงที่ ในความเป็นจริงวัตถุที่เคลื่อนที่มีการเปลี่ยนขนาดหน้าต่างได้ตลอดเวลา จึงสามารถทำการทดลองแบบปรับขนาดหน้าต่างอัตโนมัติได้

บรรณานุกรม

- Birchfield, S.T., Rangarajan, S. 2005. Spatiograms Versus Histograms for Region-based Tracking. Conference on Computer Vision and Pattern Recognition. Vol. 2 pp.1158-1163.
- Bovik, A. 2009. The Essential Guide to Image Processing. Academic Press Is An Imprint of Elsevier. 30 Corporate Drive, Suite 400, Burlington, USA.
- Bovik, A. 2009. The Essential Guide to Video Processing. Academic Press Is An Imprint of Elsevier. 30 Corporate Drive, Suite 400, Burlington, USA.
- Caulfield, D. 2008. Evaluation of Multi-part Models for Mean-Shift Tracking. Conference on Machine Vision and Image Processing. pp. 77.
- Cheng, Y., 1995. Mean Shift, Mode Seeking, and Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol.17, Issue.8, pp.790-799.
- Chu, H., Wang, K. 2009. Target Tracking Based on Mean Shift and Improved Kalman Filtering Algorithm. IEEE International Conference on Automation and Logistics. pp. 808-812.
- Collins, R. 2003. Mean-Shift Blob Tracking Through Scale Space. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp.230-240.
- Comaniciu, D., Ramesh, V., and Meer, P. 2000. Real-Time Tracking of Non-Rigid Objects using Mean Shift. IEEE Proceedings on Computer Vision and Pattern Recognition, pp. 637-678.
- Comaniciu, D., Ramesh, V., and Meer, P. 2002. Mean Shift: A Robust Approach Towards Feature Space Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence. pp. 603-619.
- Comaniciu, D., Ramesh, V., and Meer, P. 2003. Kernel-base Object Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence. pp.565-577.
- Fukanaka, K., Hostetler, L. 1975. The Estimation of The Gradient of A Density Function, with Application in Pattern Recognition. IEEE Transactions on Information Theory, pp.32-40.
- Gonzalez, R., Woods, R. 2002. Digital Image Processing. Prentice Hall Upper Saddle River, New Jersey.

- Jeyakar, J., Babu, R.V. and Ramakrishnan, K.R. 2007. Robust Object Tracking using Local Kernels and Background Information. IEEE International Conference on Image Processing. Vol. 5, pp. v49-v52.
- Jiang, z., Li, S. and Gao, D. 2007. An Adaptive Mean Shift Tracking Method Using Multiscale Images. Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition. Beijing, China. pp.1060-1066.
- Jiang, Z., Li, S., Jia, X. and Zhu, H. 2008. An improved Mean Shift Tracking Method based on Nonparametric Clustering and Adaptive Bandwidth. International Conference on Machine Learning and Cybernetics. Vol. 5, pp. 2779-2784.
- Juan, C., Hu, J. 2008. A New Spatial-Color Mean-Shift Object Tracking Algorithm with Scale and Orientation Estimation. IEEE International Conference on Robotics and Automation, pp. 2265-2270.
- Liang, D., Huang, Q. and Jiang, S. 2007. Mean Shift Blob Tracking with Adaptive Feature Selection and Scale Adaptation. IEEE International Conference on Image Processing. Vol. 3, pp. III369-III372.
- Li, Peihua. 2008. An Adaptive Bining Color Model for Mean Shift Tracking. IEEE Transaction on Circuits and Systems for Video Technology. Vol.18, Issue. 9, pp.1293-1299.
- Li, Z., Tang, Q.L. and Sang, N. 2008. Improved Mean Shift Algorithm for Occlusion Pedestrian Tracking. Electronics Letters, Vol. 44, Issue. 10, pp. 622-623.
- Lowe, D.G. 1999. Object Recognition From Local Scale-Invariant Features. The proceedings of the Seventh IEEE International Conference on Computer vision. Vol.2, pp. 1150-1157.
- Oshima, N., Saitoh, T. and Konishi, R. 2006. Real Time Mean Shift Tracking Using Optical Flow Distribution. International Joint Conference, pp.4316-4320.
- Performance Evaluation of Tracking and Surveillance (PETS) 2006 Benchmark Data. <http://pets2006.net/>. (accessed 21/7/2010).
- Sen, G., Wei, L., Xin, L. and Yongsen, L. 2009. An Adaptive Weight Values Updating Mean Shift Tracking Algorithm. International Conference on Information and Automation, pp. 790-794.
- Srinivasa, N. Computer Vision 2006. www.cs.cmu.edu/~srinivas/ (accessed 21/7/2010).
- Yilmaz, A., Javed, O. and Shah, M. 2006. Object Tracking: A Survey. ACM Comput, Vol. 38.

- Yilmaz, A. 2007. Object Tracking by Asymmetric kernel Mean Shift with Automatic Scale and Orientation Selection. IEEE Conference on Computer Vision and Pattern Recognition, pp.1-6.
- Yang, C., Duraiswami, R. and Davis, L. 2005. Efficient Mean-shift Tracking Via A New Similarity Measure. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Vol. 1, pp. 176-183.
- Yang, G., Liu, H. 2009. Visual Tracking Algorithm Based on Camshift and Multi-cue Fusion for Human Motion Analysis. IEEE International Conference on Systems Man and Cybernetics. pp.1280-1285.
- Yi, K.M., Ahn, H.S. and Choi, J.Y. 2008. Orientation and Scale Invariant Mean Shift Using Object Mask-Based Kernel. 19th International Conference on Pattern Recognition. pp.1-4.
- Zhang, L., Zhao, H. 2007. Real Time Mean Shift Tracking Using the Gabor Wavelet. International Conference on Mechatronics and Automation, pp.1617-1621.
- Zhou, Z. 2009. Object Tracking using SIFT Features and Mean Shift. Computer Vision and Image Understanding. Vol.113, Issue. 3, pp. 345-352.

ภาคผนวก

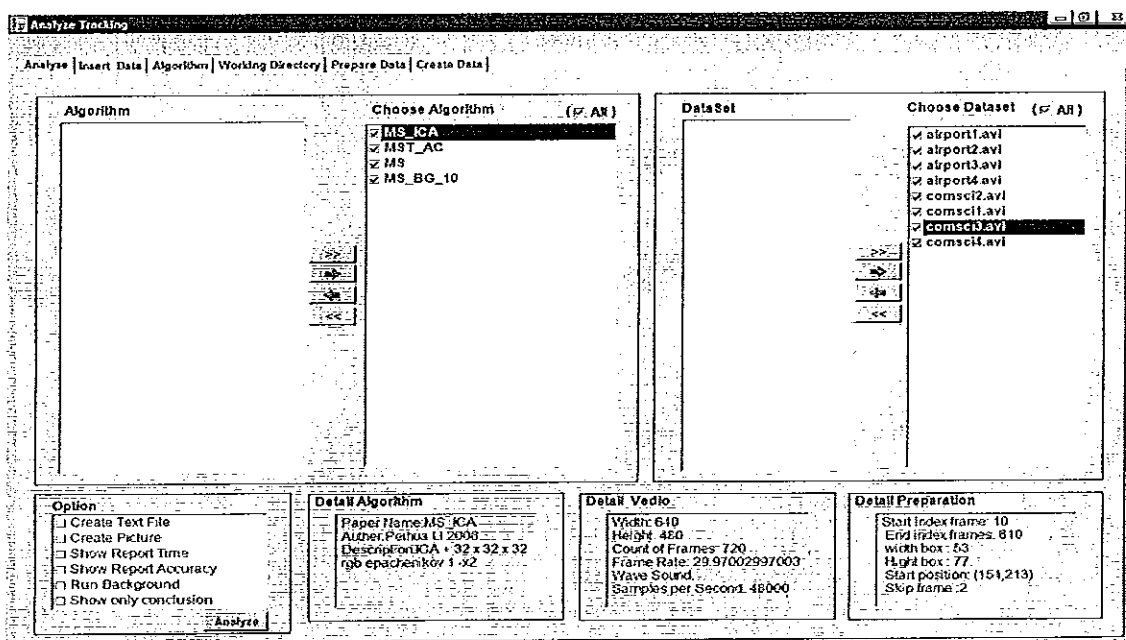
ภาคผนวก ก

คู่มือการใช้โปรแกรม Analyze Tracking

โปรแกรม Analyze Tracking เป็นโปรแกรมที่สร้างขึ้นเพื่ออำนวยความสะดวกในการวิเคราะห์ขั้นตอนวิธีการติดตามวัตถุ มีองค์ประกอบที่สำคัญ 5 ส่วนดังนี้ คือ

1. ส่วนที่เกี่ยวกับการกำหนด ตำแหน่งเริ่มต้น (Working Directory) ของข้อมูลวิดีโอที่ใช้ในการทดสอบ ซึ่งจะเป็นส่วนที่เก็บรวบรวมไฟล์วิดีโอ ตำแหน่งเริ่มต้นของขั้นตอนวิธี โดยเป็นตำแหน่งที่ใช้ในการเก็บ Library ของขั้นตอนวิธี ตำแหน่งเริ่มต้นของ Output ที่ได้ ตำแหน่งที่ใช้เก็บกลุ่มของภาพ ไฟล์ระยะทางผิดพลาดของแต่ละขั้นตอนวิธี
2. ส่วนที่เกี่ยวข้องกับการ Upload ข้อมูลทดสอบ ส่วนนี้ใช้ในการ Upload ข้อมูลวิดีโอที่ต้องการทดสอบ ข้อมูลวิดีโอที่ใช้จะต้องเป็นนามสกุล avi
3. ส่วนที่เกี่ยวข้องกับการเตรียมข้อมูลทดสอบ ส่วนนี้ใช้ในการเตรียมข้อมูลวิดีโอเพื่อทดสอบ
4. ส่วนที่เกี่ยวข้องกับการ Upload ขั้นตอนวิธี ซึ่งเป็นไฟล์ dll
5. ส่วนที่เกี่ยวข้องกับการทดสอบขั้นตอนวิธี

เมื่อเปิดโปรแกรมแสดงดังภาพประกอบ ก.1



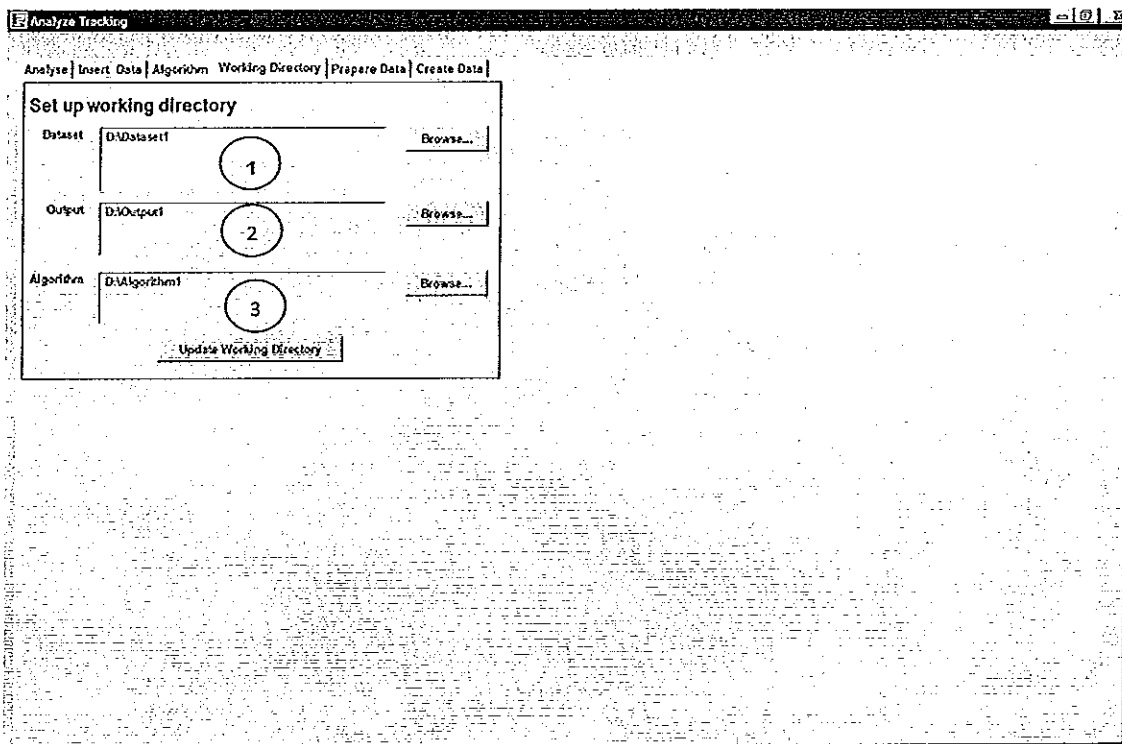
ภาพประกอบ ก.1 หน้าจอหลักของโปรแกรม

ก.1 การทำงานของโปรแกรม

แสดงการทำงานของโปรแกรม ซึ่งมี 5 ส่วนคือ 1) การกำหนดตำแหน่ง 2) การ Upload ไฟล์วิดีโอ 3) การเตรียมข้อมูลทดสอบ 4) การ Upload library และ 5) การทดสอบ ขั้นตอนวิธี ซึ่งมีรายละเอียดดังต่อไปนี้

1. การกำหนดตำแหน่ง

การกำหนดตำแหน่งของ Working Directory โดยหน้าต่างการทำงานแสดงดังภาพประกอบ ก.2 โดย วงกลม 1 เป็นเส้นทางไปยังไฟล์วิดีโอ วงกลม 2 เป็นเส้นทางไปยัง Output วงกลม 3 เป็นเส้นทางไปยัง library ซึ่ง library ในที่นี้คือขั้นตอนวิธี จากนั้นก็กดปุ่ม Upload Working Directory

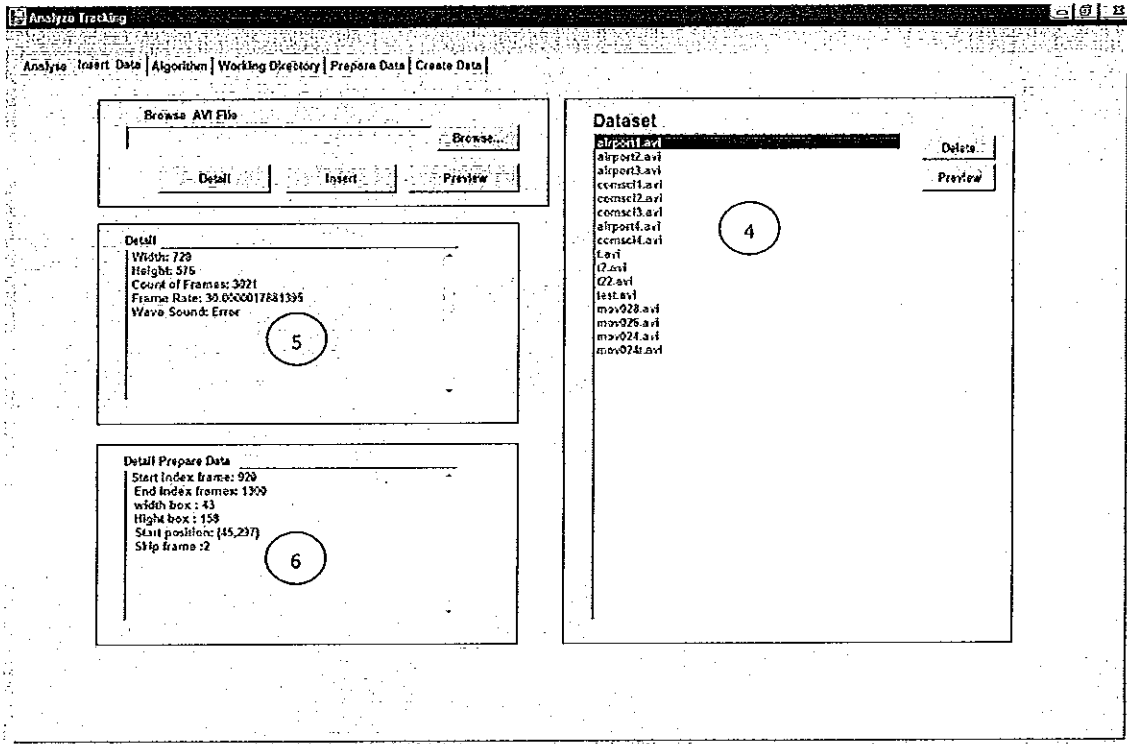


ภาพประกอบ ก.2 ระบุตำแหน่ง Working Directory

2. การ Upload ข้อมูลวิดีโอ

หน้าต่างที่ใช้ในการ Upload ข้อมูลวิดีโอ แสดงดังภาพประกอบ ก.3 โดยกดปุ่ม Browse เพื่อหาตำแหน่งของวิดีโอ จากนั้นก็กดปุ่ม Insert เพื่อย้ายไฟล์วิดีโอไปยังตำแหน่ง

ของ dataset ที่หาจากขั้นตอนที่ 1 หลังจาก Upload ข้อมูลวิดีโอแล้วชื่อวิดีโอจะไปแสดงที่วงกลมหมายเลข 4 ที่วงกลมหมายเลข 5 ใช้สำหรับแสดงคุณสมบัติของวิดีโอ สำหรับวงกลมหมายเลข 6 ใช้แสดงรายละเอียดการเตรียมข้อมูลซึ่งจะกล่าวในขั้นตอนต่อไป

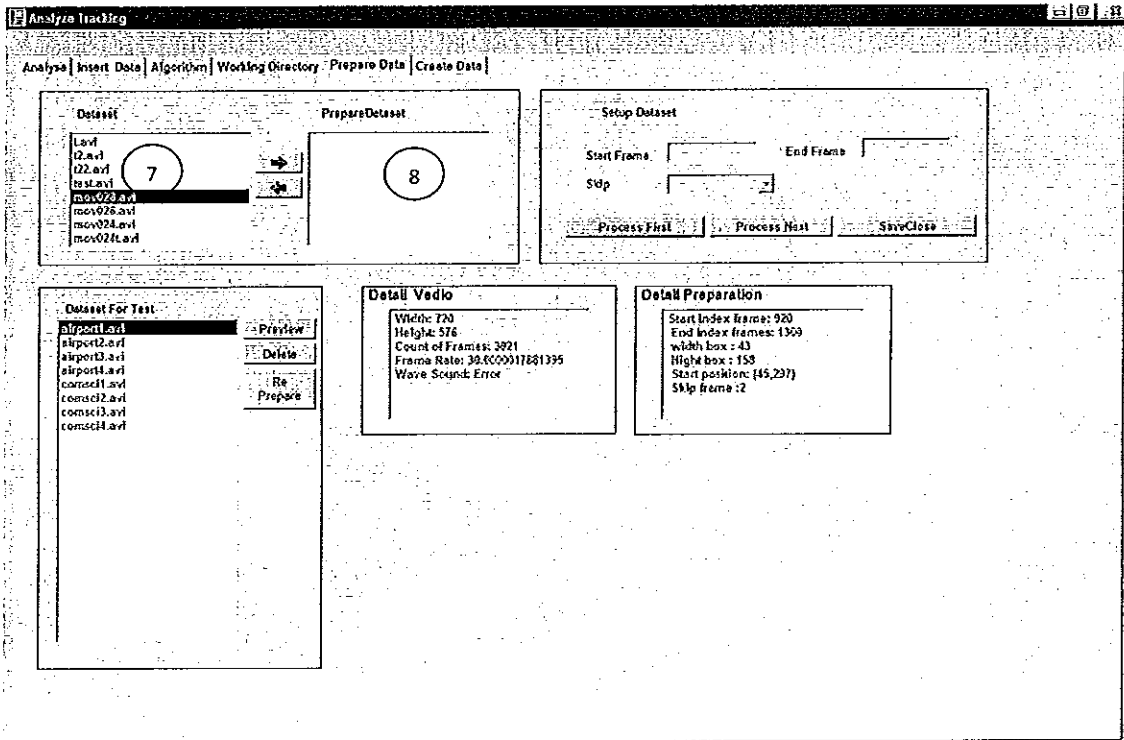


ภาพประกอบ ก.3 หน้าต่างที่ใช้ในการ Upload ข้อมูลทดสอบ

3. เตรียมข้อมูลทดสอบ

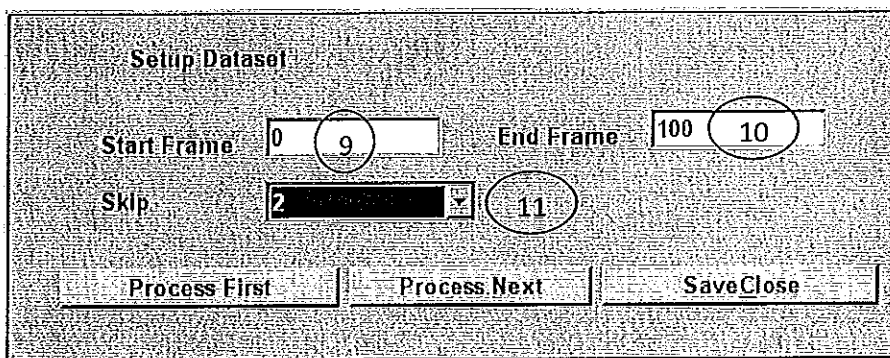
หน้าต่างที่ใช้ในการเตรียมข้อมูลแสดงดังภาพประกอบ ก.4 ซึ่งสามารถแบ่งการทำงานได้ดังนี้

- 3.1 เลือกข้อมูล avi จากวงกลม 7
- 3.2 ย้ายข้อมูล avi ที่ใช้เตรียมข้อมูล จากวงกลม 7 ไปหาวงกลม 8



ภาพประกอบ ก.4 หน้าต่างที่ใช้ในการเตรียมข้อมูลที่ทดสอบ

3.3 กำหนดตำแหน่งเริ่มต้นของเฟรมที่จะสร้างเป็นแบบจำลองหลักที่วงกลม 9 กำหนดตำแหน่งสิ้นสุดของเฟรมที่วงกลม 10 และกำหนดช่วงของการ Skip เฟรมที่จะใช้ทดสอบว่าทุก ๆ ก็เฟรม ที่วงกลม 11 จากนั้นก็กดปุ่ม Process First ดังภาพประกอบ ก.5 แสดงหน้าต่างที่ใช้ในการเตรียมข้อมูลที่ทดสอบ



ภาพประกอบ ก.5 หน้าต่างที่ใช้ในการเตรียมข้อมูลที่ทดสอบ

3.4 สร้างพื้นที่ที่จะใช้เป็นแบบจำลอง ดังภาพประกอบ ก.6 โดยใช้ Mouse ลากสี่เหลี่ยมจากนั้นก็กดปุ่ม SaveClose เสร็จแล้วก็กดปุ่ม Process Next

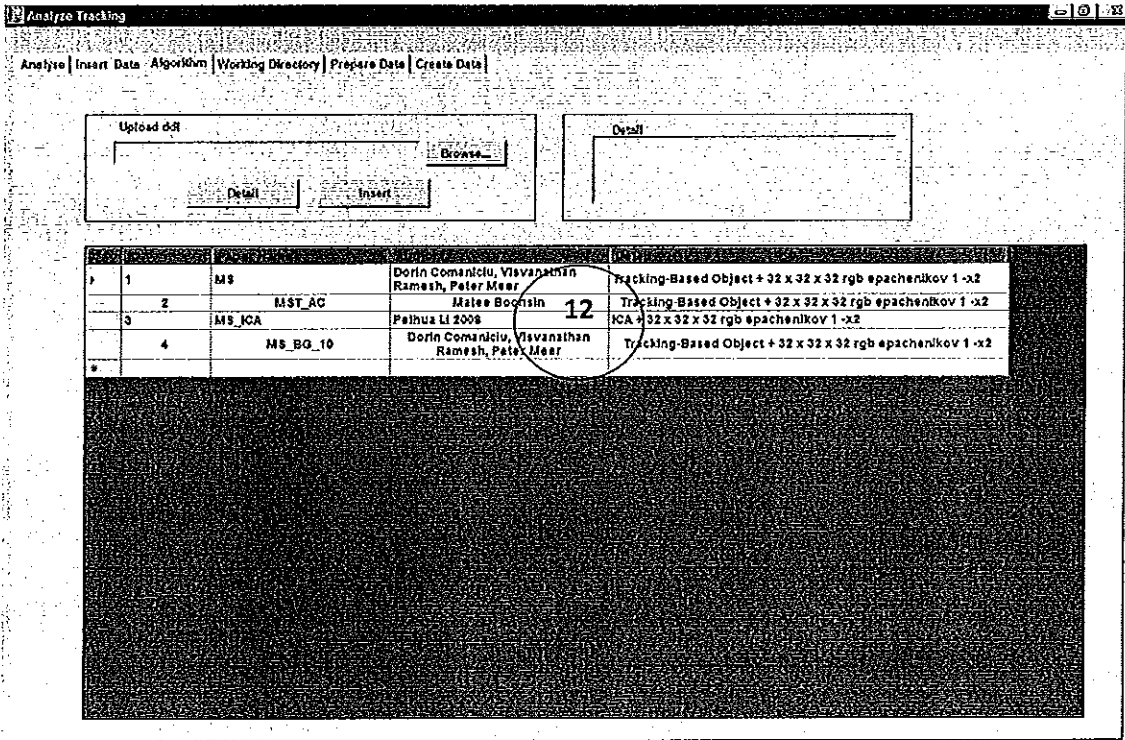


ภาพประกอบ ก.6 ตัวอย่างหน้าต่างที่ใช้ในการเตรียมข้อมูลที่จะทดสอบ

3.5 สร้างพื้นที่สี่เหลี่ยม เช่นเดียวกับข้อ 3.4 เพื่อหาจุดศูนย์กลาง หลังจากนั้นกดปุ่ม SaveClose เพื่อเก็บตำแหน่งจุดศูนย์กลางของวัตถุ ทำแบบนี้ไปเรื่อย ๆ จนกว่าถึงเฟรมสุดท้ายที่กำหนดในข้อ 3.3

4. การ Upload Library

การทำงานเริ่มจากการกดปุ่ม Browse เพื่อหาตำแหน่งของ Library จากนั้นกดปุ่ม Insert เพื่อย้าย library ไปยัง Working Directory ที่ใช้เก็บ Algorithm หลังจากนั้นชื่อและรายละเอียดของขั้นตอนวิธีแสดงดังวงกลมหมายเลข 12 ดังภาพประกอบ ก.7



ภาพประกอบ ก.7 หน้าต่างที่ Upload Library

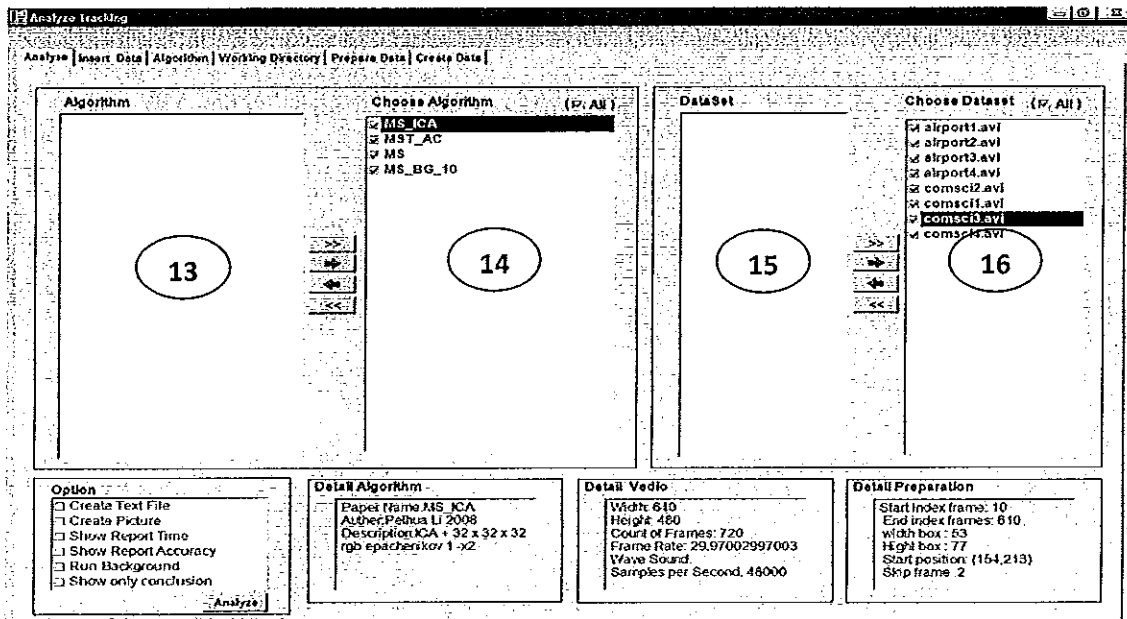
5. การทดสอบขั้นตอนวิธี

5.1 เลือกขั้นตอนวิธีที่ต้องการทดสอบวงกลมที่ 13

5.2 ย้ายขั้นตอนวิธีจากวงกลมที่ 13 ไปยังวงกลมที่ 14

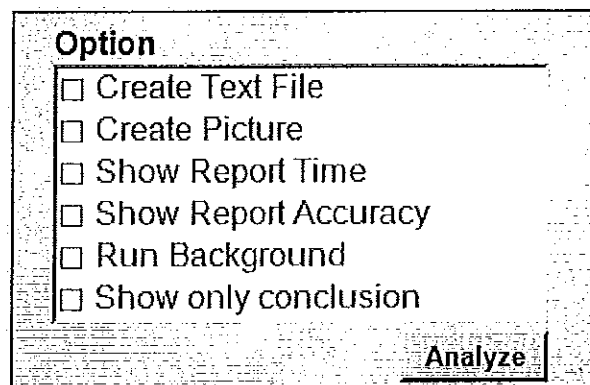
5.3 เลือกข้อมูลที่ใช้ทดสอบวงกลมที่ 15

5.4 ย้ายข้อมูลที่ใช้ทดสอบจากวงกลมที่ 15 ไปยังวงกลมที่ 16 ดังภาพประกอบ ก.8



ภาพประกอบ ก.8 หน้าต่างที่ใช้ทดสอบขั้นตอนวิธี

- 5.5 ใส่เครื่องหมายถูกใน Check Box ที่วงกลม 14 เพื่อเลือกขั้นตอนวิธี
- 5.6 ใส่เครื่องหมายถูกใน Check Box ที่วงกลม 16 เพื่อเลือกข้อมูลที่จะทดสอบ
- 5.7 การเลือก Option ในการวิเคราะห์ โดยหน้าต่าง Option แสดงดังภาพประกอบ ก.9 จากนั้นก็กดปุ่ม Analyze เพื่อทำการวิเคราะห์ ซึ่ง Option ที่ใช้ในการวิเคราะห์ทั้งหมดสำหรับการทดสอบครั้งนี้สามารถแสดงได้ดังตารางที่ ก.1

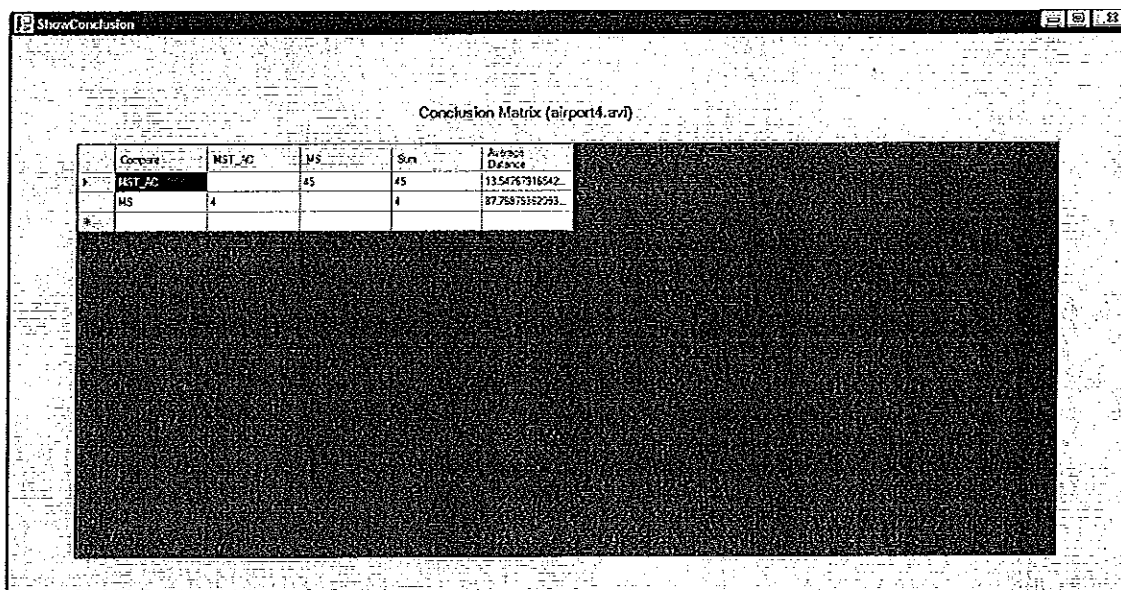


ภาพประกอบ ก.9 Option

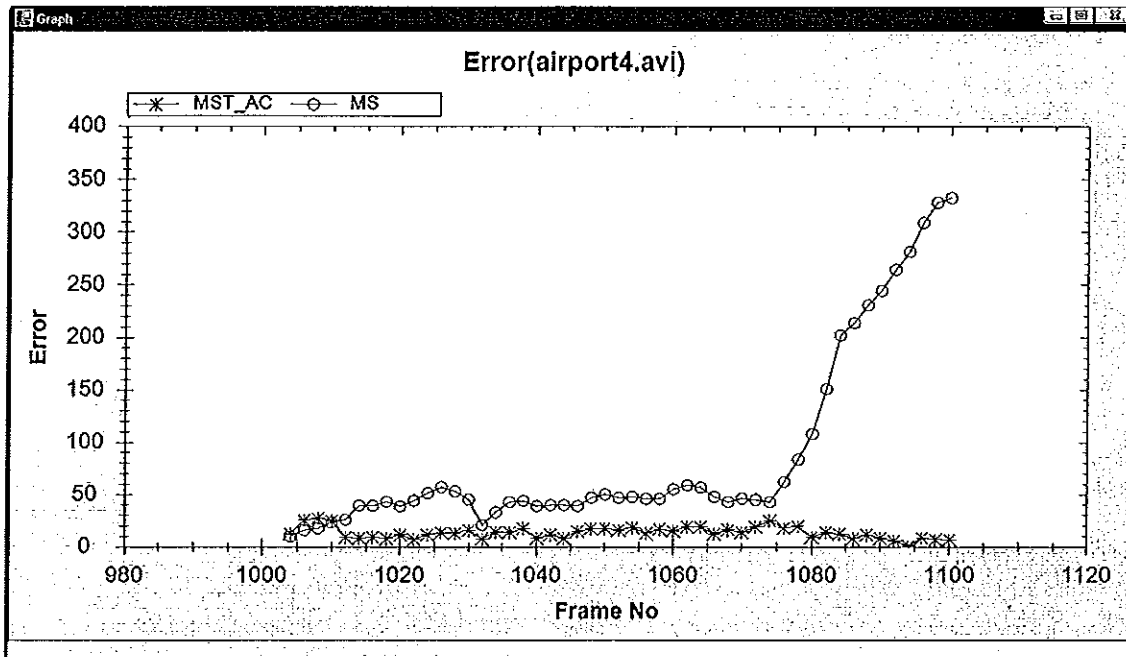
ตารางที่ ก.1 Option การวิเคราะห์

Option	ความหมาย
Create Text File	สร้างไฟล์ซึ่งมีข้อมูลค่าระยะทางผิดพลาดในแต่ละเฟรม
Create Picture	สร้างไฟล์รูปภาพขณะทำงาน
Show Report Time	แสดงผลการวิเคราะห์เรื่องเวลา ซึ่งเวลาในที่นี้เป็นเวลาที่ใช้ในการทำงานของแต่ละขั้นตอนวิธี
Show Report Accuracy	แสดงกราฟค่าระยะทางผิดพลาดในแต่ละเฟรม
Run Background	ไม่แสดงภาพขณะทำงาน
Show only Conclusion	แสดงค่าระยะทางผิดพลาดเฉลี่ยของแต่ละขั้นตอนวิธี

ภาพประกอบ ก. 10 แสดงค่าระยะทางผิดพลาดเฉลี่ย ภาพประกอบ ก. 11 แสดงค่าระยะทางผิดพลาด



ภาพประกอบ ก.10 ค่าระยะทางผิดพลาดเฉลี่ย



ภาพประกอบ ก.11 ค่าระยะทางผิดพลาด

ภาคผนวก ข

ผลงานตีพิมพ์

เรื่อง	Improving of Mean Shift Tracking Algorithm Using Adaptive Candidate Model
งานประชุมวิชาการ	7 th Annual International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON 2010)
สถานที่	จังหวัดเชียงใหม่ ประเทศไทย
วันที่	19 - 21 พฤษภาคม 2553

Improving of Mean Shift Tracking Algorithm Using Adaptive Candidate Model

Matee Boonsin
Artificial Intelligence Research Laboratory
Computer Science Department
Prince of Songkla University, Thailand
matee_b@hotmail.com

Wiphada Wettayaprasit
Artificial Intelligence Research Laboratory
Computer Science Department
Prince of Songkla University, Thailand
wwettayaprasit@yahoo.com

Ladda Preechaveerakul
iSTAR Research Laboratory
Computer Science Department
Prince of Songkla University, Thailand
ladda.p@psu.ac.th

Abstract— Mean shift tracking is used widely for object tracking. However, one of the main problems is that the background on a current frame has the same color as the target (object) which can reduce the correctness of tracking. This paper proposes Improving of Mean Shift Tracking Algorithm Using Adaptive Candidate (MST_AC) Model. This algorithm uses the background positions on the previous frame and the current frame to compute the new candidate model. The window size is fixed. The dataset is received from Performance Evaluation of Tracking and Surveillance (PETS) 2006 Benchmark. The experimental results show that the proposed MST_AC model receives higher correctness than those of the traditional mean shift algorithm (MS) and ICA mean shift algorithm (MS_ICA).

Keywords—Object tracking; mean shift tracking; candidate model.

I. INTRODUCTION

The mean shift algorithm was proposed in 1975 by Fukunaga [1]. D. Comaniciu *et al.* [2-4] was the beginner who adapted mean shift for tracking. Mean shift tracking is categorized as a kernel tracking by Yilmaz [5], which used geometric window such as circle shape or rectangular shape to represent the target. Mean shift tracking can be considered into 4 items as follows: 1) model, 2) similarity function, 3) window size adjustment and 4) weight-background information. For the first item, the model is the property of the input features of each bin of histogram. For example, D. Comaniciu *et al.* [4] used spatial a radially symmetric normalized distance from the object center as a feature. Li [6] proposed model by extending model [4] using a space of bin. Birchfield and Rangarajan [7] proposed a model which used features: number, mean, and covariance of the pixel location as features. Juan and Hu [8] used an extended model of Birchfield and Rangarajan [7] by additional mean and covariance of the color pixels. The models [7-8] are more descriptive of the target than that of Comaniciu *et al.* [4]. The second item is the similarity function. The similar function was used to compute the similarity of the model between target model and candidate model. There are many similarity functions that can be used for tracking algorithm. For example, D. Comaniciu *et al.* [4] used the Bhattacharyya. The third item is the window size adjustment that was depended on the size of target. Comaniciu *et al.* [4] chose a different window size ($\pm 3\%$) to find the highest similarity value to represent a target. Linderberg [9] used T. Linderberg's theory based on local maxima of differential scale-space filters for the adaptation

of window. Jiang *et al.* [10] used multiscale images from a Gaussian pyramid. The last item is weight-background information. Juan and Hu [8] and D. Comaniciu *et al.* [4] extended target window to find background weight.

One of the main problems of the mean shift tracking is the background on the current frame that has the same color as the target, which can reduce correctness of tracking. This paper proposes a mean shift tracking algorithm using a new adaptive candidate model by computing the background positions on the previous frame and the current frame. The rest of this paper is organized as follows. Section II is mean shift tracking model. Section III will describe on the MST_AC model. Section IV is experimental result. Section V is conclusion.

II. MEAN SHIFT TRACKING MODEL

Mean shift tracking model is a general model used to tracking, which consists of 3 steps as follows: 1) initializing, 2) creating model, 3) measuring of similarity as showed in Fig. 1.

1) Initializing

The first step is initializing steps such as color system, bin size, kernel function, and specifying the target area.

2) Creating a model.

The model can be categorized into 2 groups: a target model and a candidate model. A target model is a primary model which represents a description of target. It will be created after the target is detected. A candidate model is used to compare with a target model by using similarity function. If a candidate model is not a target model then the position of current window is shifted by a mean shift algorithm.

3) Computing similarity.

Similarity function is used to compute similarity of two models between a target model and a candidate model. Fig. 2 shows a mean shift tracking algorithm [4].

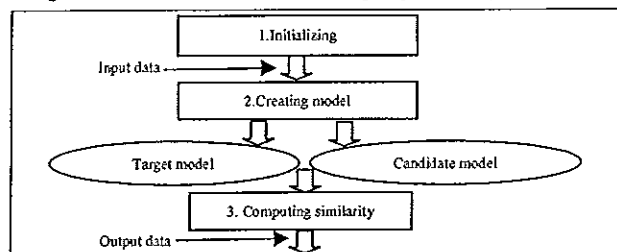


Figure 1. Mean shift tracking model.

```

1: For frame = 1 to a number of frames
2: If frame = 1
3:   Detect object
4:   Create a target model
5: Else
6:   Create a new candidate model ( $\hat{y}_0$ )
7:    $S_0 = \text{Similarity}(\hat{y}_0)$ 
8:   Compute a new center ( $\hat{y}_1$ )
9:   Create a new candidate model ( $\hat{y}_1$ )
10:   $S_1 = \text{Similarity}(\hat{y}_1)$ 
11:  While  $S_1 < S_0$ 
12:     $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_1 + \hat{y}_0)$ 
13:     $S_1 = \text{Similarity}(\hat{y}_1)$ 
14:  End While
15:  If  $\|\hat{y}_1 - \hat{y}_0\| < \epsilon$  Stop
16:  Else  $\hat{y}_0 \leftarrow \hat{y}_1$ , Goto line 8
17: End If
18: End If
19: End For

```

Figure 2. Mean shift tracking algorithm [4].

III. MST_AC ALGORITHM

The correctness of mean shift tracking depended on creating candidate model which used color position in current window. The position can be categorized into 3 groups: 1) the position is a target, 2) the position is not a target and it has the same color as a target, and 3) The position is not a target and it does not have the same color as a target. The position that reduces the correctness of the tracking algorithm is a position that is not a target and it has the same color as a target.

The method of finding the background position will use the information from the previous frame. First frame is detecting a target and then creates a target model as shows in Fig. 3 (a). The next frame is creating a candidate model. The background position (in black) is the area which is the result of an area of new window (that has been shifted) minus an area of the current window (old frame) as shows in Fig. 3 (b).

Improving of Mean Shift Tracking Algorithm Using Adaptive Candidate (MST_AC) model has 6 steps as shows in Fig. 4.

Step 1 is initializing using Epanechnikov kernel as shows in (1).

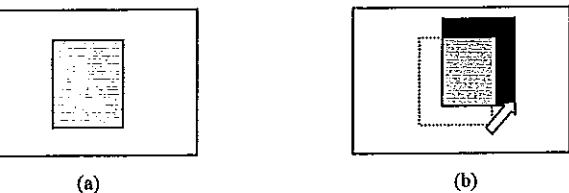


Figure 3. Movement of the target (a) A current window on first frame. Movement of window and the interested background (in black).

Step 2 is creating models of a target model \hat{q}_u as shows in (2) and a candidate model \hat{p}_u as shows in (5)-(7). The method of creating a new candidate model is different from the traditional mean shift [4] by (5)-(7).

Step 3 is computing similarity using Bhattacharyya similarity function as shows in (8).

Step 4 is shifting to a new center point (\hat{y}_1) as shows in (9) where \hat{y}_0 is a current center point.

Step 5 is assigning a background position $g[i]$ where i is the index of pixel. Step 5 consists of 4 parts.

- 1) Setting flag value of background position.
- 2) Calculating a vector from \hat{y}_1 to \hat{y}_{old} . The vector is used to identifying a direction and size of the moving target.
- 3) Creating a new temporary center point (\hat{y}_2). This step is used to computing value of the background position. The direction of background position can be found from the background position minus with a center point (\hat{y}_2). A new center point (\hat{y}_2) is created by the vector from step 5.2 and 5.3. The reason of computing value of background position in (15) is that the probability of raising background position on the current window is higher when the direction of background position is opposite the direction of moving target.
- 4) Assigning a background position value.

Step 6 is adjusting a current window position until the similarity between a target model \hat{q}_u and candidate model $\hat{p}_u(\hat{y}_0)$ is matched then store a current frame and a current window.

Note that the highlight areas in Fig. 4 are the new proposed steps or equations comparing with the traditional mean shift algorithm.

<p>Step 1: Initializing</p> <ol style="list-style-type: none"> 1.1 Initializing an area of a target model where the center position is y. 1.2 Choosing a color system. 1.3 Choosing a bin size. 1.4 The feature spatial a radially symmetric normalized distance from the object center. 1.5 Using the Epanechnikov kernel. $k(x) = \begin{cases} 1-x^2 & \text{if } 0 \leq x \text{ and } x \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$
<p>Step 2: Creating a model</p> <p>Creating a target model on the first frame (fixed)</p> <p>2.1 Store a current frame (of) and current window (\hat{y}_{old}).</p> <p>2.2 Create a target model $\{\hat{q}_u\}_{u=1..m}$ for all bin.</p> <p>Let $b(x_i)$ be an index color of bin on a position (x_i)</p> <p>h be a bandwidth of window.</p> <p>m be a number of bin.</p> <p>n be a number of pixels on current window.</p> $\hat{q}_u = C \sum_{i=1}^n k\left(\left\ \frac{y-x_i}{h}\right\ \right) \delta_u(b(x_i)) \quad (2)$

Figure 4. MST_AC algorithm.

$$C = \left(\sum_{i=1}^n k \left(\left| \frac{y-x_i}{h} \right| \right) \right)^{-1} \quad (3)$$

$$\delta_u(x) = \begin{cases} 1 & \text{if } x-u=0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Creating a candidate model on other frame

2.1 Create a candidate model

Let $b(x_i, of)$ be an index of bin on a previous frame

$b(x_i, cf)$ be an index of bin on a current frame

$$\hat{p}_u(\hat{y}_0) = \begin{cases} \sum_{i=1}^n k \left(\left| \frac{y-x_i}{h} \right| \right) \delta_u(b(x_i, cf)) - g[x_i] \delta_u(b(x_i, of)) & \text{if } b(x_i, cf) = b(x_i, of) \\ \sum_{i=1}^n k \left(\left| \frac{y-x_i}{h} \right| \right) \delta_u(b(x_i, cf)) & \text{otherwise} \end{cases} \quad (5)$$

$$\hat{p}_u(\hat{y}_0) = \begin{cases} \hat{p}_u(\hat{y}_0) & \text{if } \hat{p}_u(\hat{y}_0) \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$\hat{p}_u(\hat{y}_0) = \frac{\hat{p}_u(\hat{y}_0)}{\sum_{u=1}^m \hat{p}_u(\hat{y}_0)} \quad (7)$$

step 3 : Computing similarity

1 Evaluate similarity function on current window (\hat{y}_0).

$$\rho[\hat{p}(\hat{y}_0), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u} \quad (8)$$

step 4 : Shift to a new position

1 Shift to a new position (\hat{y}_1).

Let n be a number of pixels on current window

$$\hat{y}_1 = \frac{\sum_{i=1}^n \sum_{u=1}^m x_i \sqrt{\hat{q}_u} \delta_u(b(x_i))}{\sum_{i=1}^n \sum_{u=1}^m \sqrt{\hat{q}_u} \delta_u(b(x_i))} \quad (9)$$

step 5 : Assigning a background position

1 Set flag value of $g[i]$.

Take an area of new window (\hat{y}_1) on current frame and an area of old window (\hat{y}_{old}) on previous frame to find a new area which is a position of a previous background and set flag $g[i]$ to 1 (1 background 0 no background)

Let area (x, y, z) where position x is element of area y on frame z

(\hat{y}_{old}) be a center point of previous frame.

cf be a current frame.

of be an old frame.

For $i = 1$ to n

If $\text{area}(x_i, \hat{y}_1, cf) = \text{area}(x_i, \hat{y}_{old}, of) \neq \emptyset$ and $\hat{q}(b(x_i, of)) > 0$

$g[i] = 1$

Else If $\text{area}(x_i, \hat{y}_{old}, of) = \text{area}(x_i, \hat{y}_1, cf) \neq \emptyset$

$g[i] = 0$

End If

Figure 4 MST_AC algorithm (Continue).

5.2 Calculating a vector from \hat{y}_1 to \hat{y}_{old} .

Let l represent a length of vector.

$$\Delta x = \frac{\hat{y}(1, x) - \hat{y}(old, x)}{2} \quad (10)$$

$$\Delta y = \frac{\hat{y}(1, y) - \hat{y}(old, y)}{2} \quad (11)$$

$$l = \sqrt{\Delta x^2 + \Delta y^2} \quad (12)$$

5.3 Creating a new temporary center point \hat{y}_2 by considering distance and slope of vector \hat{y}_2 to \hat{y}_1 .

$$m = \frac{\hat{y}(2, y) - \hat{y}(1, y)}{\hat{y}(2, x) - \hat{y}(1, x)} = \frac{\Delta y}{\Delta x} \quad \text{if } \Delta x \neq 0 \quad (13)$$

$$l^2 = (\hat{y}(2, y) - \hat{y}(1, y))^2 + (\hat{y}(2, x) - \hat{y}(1, x))^2 \quad (14)$$

From (10), (11), (12), (13) and (14) will be (15).

$$\hat{y}_2 = \begin{cases} \hat{y}(2, x) = \hat{y}(1, x) \pm \frac{l}{\sqrt{m^2+1}}, \quad \hat{y}(2, y) = \hat{y}(1, y) \pm \frac{l \times m}{\sqrt{m^2+1}} & \text{if } \Delta x \neq 0 \\ \hat{y}(2, x) = \hat{y}(1, x), \quad \hat{y}(2, y) = \hat{y}(1, y) + l & \text{if } \Delta x = 0 \text{ and } \Delta y > 0 \\ \hat{y}(2, x) = \hat{y}(1, x), \quad \hat{y}(2, y) = \hat{y}(1, y) - l & \text{if } \Delta x = 0 \text{ and } \Delta y < 0 \\ \hat{y}(2, x) = \hat{y}(1, x), \quad \hat{y}(2, y) = \hat{y}(1, y) & \text{if } \Delta x = 0 \text{ and } \Delta y = 0 \end{cases} \quad (15)$$

5.4 Assigning a value position $g[i]$

Let W be a width of window, H be a high of window and (x, y) be a position on window.

For $i = 1$ to n

If $g[i] > 0$

$$g[i] = \sqrt{((x - \hat{y}(2, x))^2 / W)^2 + ((y - \hat{y}(2, y))^2 / H)^2}$$

Step 6 : Adjust a position of window

6.1 Creating a candidate model on a new window (\hat{y}_1).

6.2 Evaluate similarity function on a new window (\hat{y}_1).

$$\rho[\hat{p}(\hat{y}_1), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_1) \hat{q}_u} \quad (16)$$

6.3 Adjusting a current window position.

While $\rho[\hat{p}(\hat{y}_1), \hat{q}] < \rho[\hat{p}(\hat{y}_0), \hat{q}]$

$$\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_1 + \hat{y}_0)$$

Evaluate $\rho[\hat{p}(\hat{y}_1), \hat{q}]$

End While

If $\|\hat{y}_1 - \hat{y}_0\| < \varepsilon$ Goto step 6.4

Else $\hat{y}_0 \leftarrow \hat{y}_1$ Goto step 4

6.4 Store a current frame (of) and a current window (\hat{y}_{old}).

Figure 4. MST_AC algorithm (Continue).

IV. EXPERIMENT RESULTS

The Mean Shift Tracking Algorithm using Adaptive Candidate (MST_AC) model, MS_ICA [6] and MS [4] have been implemented in visual C++ and tested on a 2.40 GHz Intel(R) Core (TM) 2 Quad CPU Q6600 with 2.00 GB memory. The number of bins is 32x32x32 bins, the color system is RGB. A window size is fixed. Initializing a target object is marked manually on the first frame. The target position is manually marked every two frames. The Euclidean distance is used to find an error. The data set is received from sequence pictures of PETS 2006 Benchmark data [11]. The names of sequence pictures are walknarecamera I and walknarecamera II. The details of sequence pictures were showed on TABLE I.

Fig. 5 shows the example of the tracking algorithms where (a) is MST_AC, (b) is MS_ICA (intra-cluster = 8),

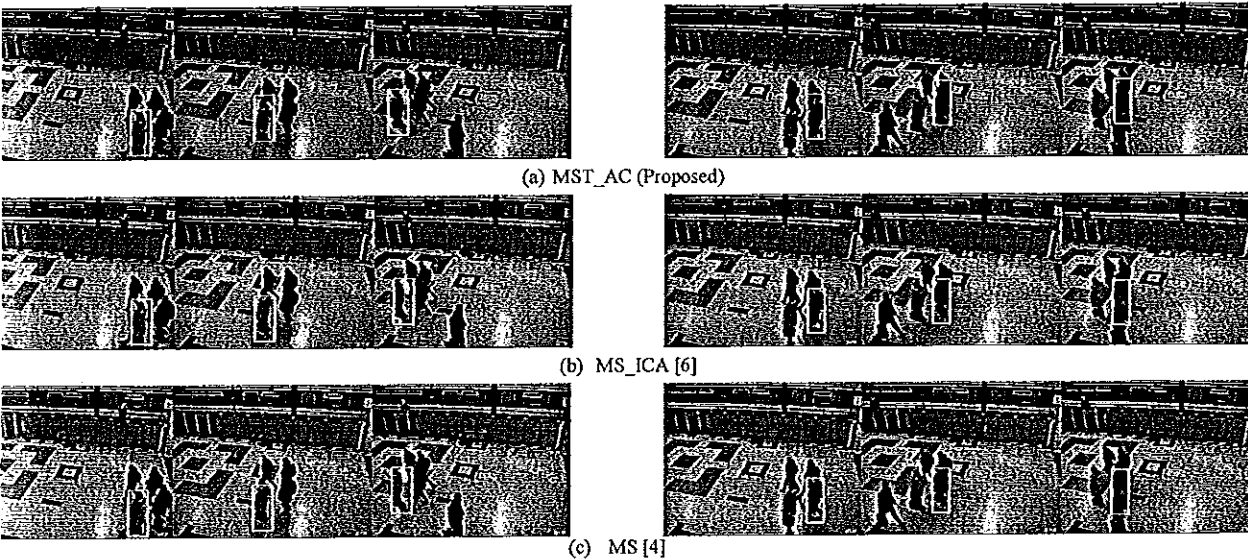


Figure 5. Example of tracking result sequence walknarecamera I (First column) frame index 2663, 2719 and 2779, walknarecamera II (Second column) frame index 2689, 2749, and 2767.

TABLE I. DETAILS OF SEQUENCE DATA [11].

Sequence pictures	Number of frame	Size	Current window first frame (start)	(σ)
Walknarecamera I	140	720 x 576	72x171	30
Walknarecamera II	170	720 x 576	45x137	30

and (c) is MS (extended window +10 % to compute weight background). The tracking results for both Walknarecamera I sequence and Walknarecamera II sequence, MST_AC is outperformed. Fig. 6. shows distance error of walknarecamera I sequence where MST_AC gives less error than MS [4] and MS_ICA [6]. Fig. 7 shows distance error of walknarecamera II sequence where MST_AC gives less error than MS [4] and MS_ICA [6].

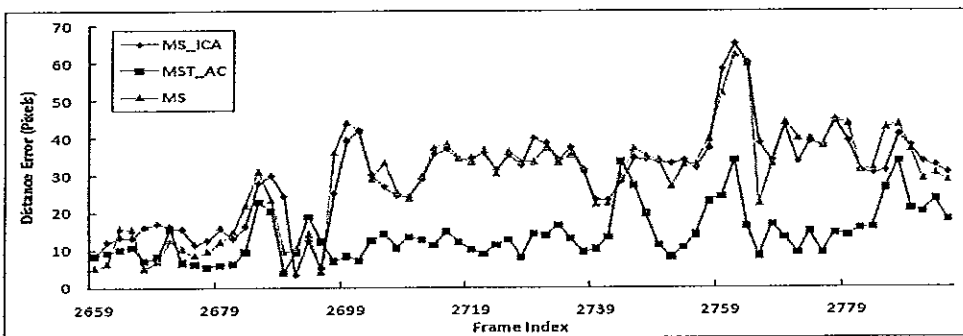


Figure 6. Distance error of walknarecamera I sequence.

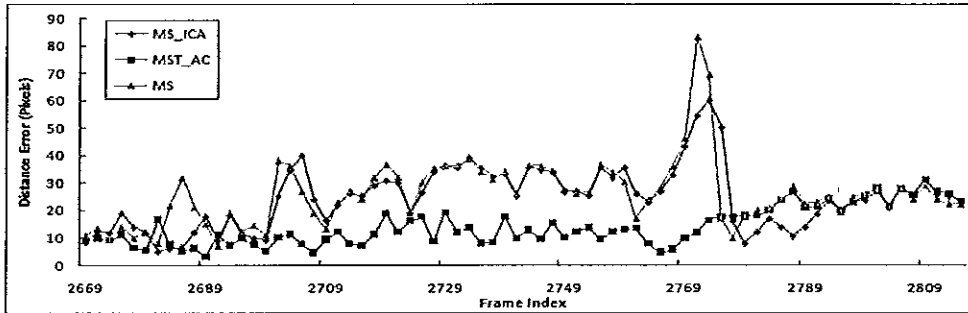


Figure 7. Distance error of walknearcamera II sequence.

TABLE II. AVERAGE TRACKING ERRORS

Sequence pictures	MST_AC (Proposed)	MS [4]	MS_ICA [6]
Walknearcamera I	13.96	29.30	29.82
Walknearcamera II	14.05	26.17	24.61

TABLE III. AVERAGE TIME

Sequence pictures	MST_AC (Proposed)	MS [4]	MS_ICA [6]
Walknearcamera I	0.041	0.042	0.463
Walknearcamera II	0.033	0.032	0.359

Average tracking errors from MST_AC, MS [4], and MS_ICA [6] are summarized in TABLE II. Average tracking errors of MST_AC gives the least value when compare with MS_ICA [6] and MS [4]. For example, in walknearcamera I sequence, average tracking errors of MST_AC is 13.96, average tracking errors of MS [4] is 29.30, average tracking errors of MS_ICA [6] is 29.82, in walknearcamera II sequence, average tracking errors of MST_AC is 14.05, average tracking errors of MS [4] is 26.17, average tracking errors of MS_ICA [6] is 24.61.

TABLE III shows the average time of iteration 3 times. Each of the sequence pictures used the first 100 frames. The average time of iteration of MST_AC is small and similar to the average time of MS [4]. The average time of MS_ICA [6] is the highest. For example, in walknearcamera I sequence, the average time of MST_AC is 0.041 second, the average time of MS [4] is 0.042 second and the average time of MS_ICA [6] is 0.463 second.

V. CONCLUSION

In this paper, we proposed a new method to solve background problem in mean shift algorithm using the background position on the previous frame and the current frame to compute the new candidate model. The experimental results show that this proposed method (MST_AC) gives the highest correctness of tracking algorithm and less average time.

REFERENCES

- [1] K. Fukunaga and L. Hostetter, "The Estimation of The Gradient of A Density Function, with Application in Pattern Recognition," *IEEE Transactions on information Theory*, pp. 32-40, 1975.
- [2] D. Commaniciu, V. Ramesh, and P. Meer, "Real-Time Tracking of Non-rigid Objects using Mean Shift," *IEEE Proceedings on Computer Vision and Pattern Recognition*, pp. 673-678, 2000.
- [3] D. Commaniciu, V. Ramesh, and P. Meer, "Mean Shift: A robust approach towards feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 603-619, 2002.
- [4] D. Commaniciu, V. Ramesh, and P. Meer, "Kernel-based Object Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 546-577, May 2003.
- [5] Alper Yilmaz, "Object Tracking: A Survey," *ACM Comput. Surv.*, December 2006.
- [6] Peihua Li, "An Adaptive Binning Color Model for Mean Shift tracking," *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 18, September 2008.
- [7] S.T. Birchfield and S. Rangarajan, "SpatioGrams Versus Histograms for Region-Based Tracking," *Proc. CVPR*, pp.1158-1163, June 2005.
- [8] C. W. Juan and J. S. Hu, "A New Spatial-Color Mean-Shift Object Tracking Algorithm with Scale and Orientation Estimation," *IEEE International Conference on Robotics and Automation Pasadena, Ca, USA*, May 2008.
- [9] R. Collins, "Mean-Shift Blob Tracking Through Scale Space," *Proc IEEE Conf. Comp. Vision Pattern Recognition*, pp.230-240, 2003.
- [10] Jiang, Shao-Fa Li, Dong-Fa Gao, "An Adaptive Mean Shift Tracking Method Using Multiscale Images," *Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition*, Beijing, China, November 2007.
- [11] Performance Evaluation of Tracking and Surveillance (PETS) 2006 Benchmark Data, <http://pets2006.net/>.

ประวัติผู้เขียน

ชื่อ สกุล นายเมธี บุญสิน
รหัสประจำตัวนักศึกษา 5010220188
วุฒิการศึกษา
วุฒิ ชื่อสถาบัน ปีที่สำเร็จการศึกษา
วท.บ. (วิทยาการคอมพิวเตอร์) มหาวิทยาลัยสงขลานครินทร์ 2548

การตีพิมพ์เผยแพร่ผลงาน

1. Boonsin, M., Preechaveerakul, L., and Wettayaprasit, W. 2010. Improving of Mean Shift Tracking Algorithm Using Adaptive Candidate Model. 7th Annual International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON 2010) Chiang Mai, Thailand, May 19-21, 2010.