

อัลกอริทึมการค้นหาแบบเพรชชนิดไม่สมมาตรแบบปรับตัวได้โดยใช้การปรับย้ายจุดศูนย์กลาง  
สำหรับการประมาณการเคลื่อนที่ในการเข้ารหัสสัญญาณดิจิทัล



นายวิศักดิ์ สรรพสิทธิวงศ์

สถาบันวิทยบริการ

จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า

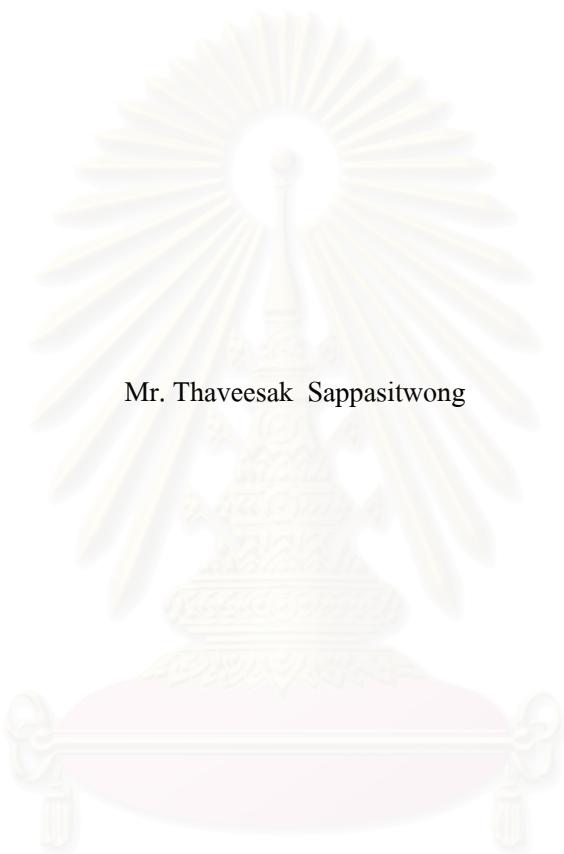
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2544

ISBN 974-03-1517-8

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

ADAPTIVE ASYMMETRIC DIAMOND SEARCH ALGORITHM USING  
ADAPTIVE SEARCH CENTER FOR MOTION ESTIMATION IN VIDEO CODING



Mr. Thaveesak Sappasitwong

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

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

Department of Electrical Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2001

ISBN 974-03-1517-8



ทวิศักดิ์ สรรพสิทธิ์วงศ์ : อัลกอริทึมการค้นหาแบบเพชรชนิดไม่สมมาตรแบบปรับตัวได้  
 โดยใช้การปรับย้ายจุดศูนย์กลางสำหรับการประมาณการเคลื่อนที่ในการเข้ารหัสสัญญาณ  
 วิดีทัศน์. (ADAPTIVE ASYMMETRIC DIAMOND SEARCH ALGORITHM USING  
 ADAPTIVE SEARCH CENTER FOR MOTION ESTIMATION IN VIDEO CODING)  
 อ. ที่ปรึกษา : รศ. ดร.สมชาย จิตะพันธ์กุล , 121 หน้า. ISBN 974-03-1517-8.

การประมาณการเคลื่อนที่มีบทบาทสำคัญในกระบวนการเข้ารหัสสัญญาณวิดีโอ การ  
 ประมาณการเคลื่อนที่ด้วยอัลกอริทึมการค้นหาแบบเพชรชนิดไม่สมมาตร  
 สัญญาณวิดีโอทั้งหมด วิทยานิพนธ์ฉบับนี้เสนออัลกอริทึมการค้นหาแบบเพชรชนิดไม่สมมาตร  
 แบบปรับตัวได้โดยใช้การปรับย้ายจุดศูนย์กลาง ซึ่งใช้ประโยชน์จากความสัมพันธ์กันระหว่างเวกเตอร์  
 การเคลื่อนที่ของบล็อกข้างเคียงทั้งเชิงพื้นที่และเชิงเวลา เพื่อปรับรูปแบบการค้นหาให้เหมาะสมกับ  
 ลักษณะการเคลื่อนที่ของแต่ละลำดับภาพ เทคนิคการค้นหาที่นำเสนอในวิทยานิพนธ์ฉบับนี้มุ่งเน้นใน  
 การลดความซับซ้อนของการประมาณการเคลื่อนที่โดยยังคงรักษาความถูกต้องในการประมาณเอาไว้  
 ผลการจำลองระบบ แสดงให้เห็นว่า อัลกอริทึมนี้สามารถลดความซับซ้อนในการคำนวณได้มากและมี  
 ความถูกต้องในการคำนวณอยู่ในระดับที่น่าพอใจ โดยในลำดับภาพที่เคลื่อนที่ช้าสามารถลดจำนวนจุด  
 การค้นหาได้มากกว่าอัลกอริทึมการค้นหาแบบทั้งหมดได้สูงสุดประมาณ 33 เท่า โดยที่ค่าอัตราส่วน  
 สัญญาณต่อสัญญาณรบกวนที่ต่ำที่สุดมีค่าประมาณ 38.66 db

สถาบันวิทยบริการ  
 จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา.....วิศวกรรมไฟฟ้า.....ลายมือชื่อนิสิต.....  
 สาขาวิชา.....วิศวกรรมไฟฟ้า.....ลายมือชื่ออาจารย์ที่ปรึกษา.....  
 ปีการศึกษา.....2544.....ลายมือชื่ออาจารย์ที่ปรึกษาร่วม.....

## 4370310421 : MAJOR ELECTRICAL ENGINEERING

KEY WORD: VIDEO CODING / MOTION ESTIMATION / BLOCK MATCHING  
ALGORITHM / MOTION VECTOR PREDICTION / ADAPTIVE

THAVEESAK SAPPASITWONG : ASYMMETRIC DIAMOND SEARCH

ALGORITHM USING ADAPTIVE SEARCH CENTER FOR MOTION

ESTIMATION IN VIDEO CODING. THESIS ADVISOR:

ASSOC. PROF. SOMCHAI JITAPUNKUL, Dr.Eng., 121 pp. ISBN 974-03-1517-8.

Motion estimation is playing a significant role in digital video coding preprocess. The block-based motion estimation has been widely used in general video-coding standard. This thesis proposed an adaptive asymmetric diamond search algorithm using adaptive search center which exploits the correlation of motion vectors between adjacent blocks in order to set the search pattern suitable with the case of motion object in each sequences. The proposed algorithm focuses on computational complexity reducing in motion estimation whilst maintain estimation accuracy. Simulation results show that this algorithm decreased an amount of complexity while keeping satisfactory accuracies, MSE and PSNR. In the gentle motion sequence, the proposed algorithm can reduce number of searchpoints from that of full search algorithm up to 33 times while the smallest PSNR is 38.66 db.

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

Department.....Electrical Engineering.....Student's Signature.....

Field of study.....Electrical Engineering.....Advisor's signature.....

Academic year.....2001.....Co-advisor's signature.....

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความช่วยเหลืออย่างดียิ่งของท่านอาจารย์ที่ปรึกษา วิทยานิพนธ์ รองศาสตราจารย์ ดร.สมชาย จิตะพันธ์กุล ซึ่งได้ให้ความกรุณาแนะนำและเสนอแนะ ข้อคิดเห็นต่าง ๆ ในการทำวิจัยมาตลอด ผู้วิจัยจึงขอกราบขอบพระคุณ

กราบขอบพระคุณบิดามารดา ผู้ซึ่งทำให้ผู้วิจัยได้รับแต่สิ่งดี ๆ เสมอมา ขอบพระคุณสำหรับ ความรัก ความห่วงใยอาทรที่ไม่เคยลดน้อยลงไปเลย ขอบคุณพี่สาวของผู้วิจัยที่คอยให้กำลังใจในการทำงานอย่างต่อเนื่อง

ขอบพระคุณ ดร.วุฒิพงศ์ อารีกุล ที่ได้ให้ความอนุเคราะห์ชุดภาพทดสอบมาตรฐาน เพื่อใช้ สำหรับงานวิจัยในวิทยานิพนธ์ฉบับนี้

ขอบคุณ คุณอาทิตย์ ธรรมตระการ สำหรับคำแนะนำด้านเทคนิคการเขียน โปรแกรมอันมี ประสิทธิภาพยิ่ง

ขอขอบคุณเพื่อน ๆ และคนรอบตัวของผู้วิจัยทุก ๆ กลุ่มและทุก ๆ คน ไม่ว่าจะเป็นที่อยู่ ภายในห้องปฏิบัติการวิจัยกรรมวิธีสัญญาณดิจิทัล (Digital Signal Processing Research Laboratory) หรือที่ใดก็ตาม สำหรับความช่วยเหลือและแรงบันดาลใจ ขอขอบคุณที่ทำให้วันเวลาของผู้วิจัยผ่านไป อย่างมีความหมาย

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ฅ
สารบัญภาพ.....	ฉ
บัญชีคำศัพท์.....	ณ
บทที่	
1 บทนำ.....	1
1.1 การใช้งานของวิดิทัศน์ของระบบดิจิทัล.....	2
1.2 มาตรฐานการเข้ารหัสสัญญาณวิดิทัศน์.....	3
1.2.1 มาตรฐานของ ITU-T.....	3
1.2.2 มาตรฐานของ MPEG.....	4
1.3 แนวทางในการเข้ารหัสสัญญาณวิดิทัศน์.....	5
1.4 ความรู้เบื้องต้นเกี่ยวกับการประมวลผลเคลื่อนที่.....	5
1.5 แนวทางของวิทยานิพนธ์.....	7
1.6 เป้าหมายและขอบเขตวิทยานิพนธ์.....	7
1.7 ขั้นตอนและวิธีดำเนินงาน.....	7
1.8 ประโยชน์ที่คาดว่าจะได้รับ.....	8
1.9 โครงสร้างวิทยานิพนธ์.....	8
2 ความรู้พื้นฐาน.....	9
2.1 วิดิทัศน์ระบบดิจิทัล.....	9
2.2 การเข้ารหัสสัญญาณวิดิทัศน์.....	10
2.2.1 การเข้ารหัสภายในเฟรม.....	10
2.2.2 การเข้ารหัสระหว่างเฟรม.....	11
2.3 การลดความซ้ำซ้อนทางปริภูมิ.....	12
2.4 การลดความซ้ำซ้อนทางเวลา.....	12

## สารบัญ (ต่อ)

บทที่	หน้า
2.5	การประมาณการเคลื่อนที่.....14
2.6	การชดเชยการเคลื่อนที่.....15
2.7	อัลกอริทึมการเข้าคู่บล็อก.....17
2.7.1	สมมติฐานของอัลกอริทึมการเข้าคู่บล็อก.....17
2.7.2	ขั้นตอนของอัลกอริทึมการเข้าคู่บล็อก.....18
2.7.3	การวัดค่าความผิดพลาด.....20
2.7.3.1	ค่าความผิดพลาดกำลังสองเฉลี่ย.....20
2.7.3.2	ค่าความผิดพลาดสัมบูรณ์เฉลี่ย.....20
2.7.3.3	ค่าผลรวมค่าสัมบูรณ์ความแตกต่าง.....21
2.8	กระบวนการประมาณการเคลื่อนที่อย่างรวดเร็ว.....21
2.8.1	สมมติฐานของการประมาณการเคลื่อนที่อย่างรวดเร็ว.....21
2.8.2	งานวิจัยด้านการเข้าคู่บล็อกอย่างรวดเร็ว.....22
2.8.2.1	การค้นหาแบบสามขั้นตอน.....23
2.8.2.2	การค้นหาแบบลอกการิทึม.....24
2.8.2.3	การค้นหาแบบสามขั้นตอนวิธีใหม่.....25
2.8.2.4	การค้นหาแบบลดระดับเกรเดียนต์เชิงบล็อก.....27
2.8.2.5	การค้นหาแบบสี่ขั้นตอน.....28
2.8.2.6	การค้นหาหารูปเพชร.....29
2.9	การวัดสมรรถนะของระเบียบวิธี.....31
2.9.1	การวัดคุณภาพของภาพ.....31
2.9.1.1	ค่าความผิดพลาดกำลังสองเฉลี่ย.....31
2.9.1.2	ค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน.....32
2.9.2	การวัดความซับซ้อนในการคำนวณของระเบียบวิธี.....32
3	การค้นหาแบบเพชรชนิดไม่สมมาตรแบบปรับตัวได้โดยใช้การปรับย้ายจุดศูนย์กลาง.....34
3.1	สหสัมพันธ์กันระหว่างบล็อก.....34
3.2	การเลือกเวกเตอร์การเคลื่อนที่ตอนเริ่มต้นค้นหา.....37
3.2.1	การทำนายโดยค่าเฉลี่ย.....37
3.2.2	การทำนายแบบเน้นค่าเฉลี่ย.....37



สารบัญ (ต่อ)

บทที่	หน้า
3.3	การค้นหาแบบเพชรชนิดไม่สมมาตรแบบปรับตัวได้
	โดยใช้การปรับย้ายจุดศูนย์กลาง.....38
3.3.1	การจำแนกลักษณะการเคลื่อนที่และการทำนายเวกเตอร์การเคลื่อนที่.....38
	3.3.1.1 ประเภทที่ 1 การเคลื่อนที่แบบทั่วไป.....40
	3.3.1.2 ประเภทที่ 2 การเคลื่อนที่แบบเร็ว.....40
3.3.2	การปรับรูปแบบการค้นหาของอัลกอริทึมการค้นหารูปเพชร.....41
	3.3.2.1 รูปแบบการค้นหาแบบเพชรชนิดไม่สมมาตร.....41
3.3.3	การปรับย้ายจุดศูนย์กลางการค้นหา.....43
3.3.4	ขั้นตอนการค้นหา.....45
3.4	การค้นหาแบบเพชรไม่สมมาตร.....47
4	ผลการทดสอบ.....48
4.1	ระบบที่ใช้ในการทดสอบและชุดภาพที่ใช้ในการทดสอบ.....48
	4.1.1 ระบบที่ใช้ในการทดสอบ.....49
	4.1.2 ชุดภาพที่ใช้ในการทดสอบ.....50
4.2	การทดลองเพื่อทดสอบแนวความคิด.....53
	4.2.1 ความน่าจะเป็นในการพบเวกเตอร์การเคลื่อนที่ที่แท้จริง.....53
	4.2.1.1 วิธีการทดสอบ.....54
	4.2.2 ระยะห่างเฉลี่ยระหว่างเวกเตอร์การเคลื่อนที่ที่แท้จริง.....54
4.3	การทดสอบสมรรถนะของอัลกอริทึมการค้นหาแบบเพชรชนิดไม่สมมาตร
	แบบปรับตัวได้โดยใช้การปรับย้ายจุดศูนย์กลาง.....56
	4.3.1 ผลการทดสอบเบื้องต้น.....56
	4.3.1.1 ผลการทดสอบกับชุดภาพ Akiyo.....58
	4.3.1.2 ผลการทดสอบกับชุดภาพ Claire.....62
	4.3.1.3 ผลการทดสอบกับชุดภาพ Miss America.....65
	4.3.1.4 ผลการทดสอบกับชุดภาพ Carphone.....68
	4.3.1.5 ผลการทดสอบกับชุดภาพ Salesman.....71
	4.3.1.6 ผลการทดสอบกับชุดภาพ Suzie.....74
	4.3.1.7 ผลการทดสอบกับชุดภาพ Foreman.....77
	4.3.1.8 ผลการทดสอบกับชุดภาพ Trevor.....80

## สารบัญ (ต่อ)

บทที่	หน้า
4.3.1.9 ผลการทดสอบกับชุดภาพ Mother and daughter.....	83
4.3.1.10 ผลการทดสอบกับชุดภาพ Grandmother.....	86
4.3.1.11 ผลการทดสอบกับชุดภาพ Table tennis.....	89
4.3.2 ตัวอย่างภาพที่สร้างขึ้นจากอัลกอริทึมต่าง ๆ.....	92
4.3.2.1 ตัวอย่างภาพที่สร้างขึ้นของลำดับภาพ Akiyo.....	92
4.3.2.2 ตัวอย่างภาพที่สร้างขึ้นของลำดับภาพ Carphone.....	93
4.3.3 สนามการเคลื่อนที่ของลำดับภาพ.....	95
4.3.3.1 การทดสอบกับลำดับภาพ Carphone.....	95
4.3.3.2 การทดสอบกับลำดับภาพ Table tennis.....	97
4.3.3.3 การทดสอบกับลำดับภาพ Suzie.....	99
5 สรุปผลการวิจัยและข้อเสนอแนะ.....	102
5.1 สรุปผลการวิจัย.....	102
5.2 ข้อเสนอแนะสำหรับการวิจัยในอนาคต.....	102
รายการอ้างอิง.....	104
ภาคผนวก ก.....	106
ภาคผนวก ข.....	112
ประวัติผู้เขียนวิทยานิพนธ์.....	121

## สารบัญตาราง

	หน้า
ตารางที่ 1.1	มาตรฐานการเข้ารหัสสัญญาณวิดีโอของ ITU-T ตระกูลเอช.....3
ตารางที่ 1.2	มาตรฐานในการเข้ารหัสสัญญาณวิดีโอของ MPEG.....4
ตารางที่ 2.1	รูปแบบของสัญญาณวิดีโอที่ใช้งานในปัจจุบัน.....10
ตารางที่ 3.1	มุมมองของเวกเตอร์การเคลื่อนที่ที่ใช้กำหนดทิศทาง.....42
ตารางที่ 3.2	การกำหนดทิศทางให้กับเวกเตอร์การเคลื่อนที่ที่ทำนายไว้.....42
ตารางที่ 4.1	ผลการทดลองเพื่อทดสอบแนวความคิด.....55
ตารางที่ 4.2	การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Akiyo.....60
ตารางที่ 4.3	การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Claire.....64
ตารางที่ 4.4	การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Miss America.....67
ตารางที่ 4.5	การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Carphone.....70
ตารางที่ 4.6	การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Salesman.....73
ตารางที่ 4.7	การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Suzie.....76
ตารางที่ 4.8	การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Foreman.....79
ตารางที่ 4.9	การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Trevor.....82
ตารางที่ 4.10	การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Mother.....85
ตารางที่ 4.11	การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Grandmother.....88
ตารางที่ 4.12	การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Table.....91

## สารบัญภาพ

ภาพประกอบ	หน้า
รูปที่ 1.1 ความต้องการแบนด์วิดท์สำหรับเทคโนโลยีประเภทต่าง ๆ.....	2
รูปที่ 1.2 ตัวอย่างการหาเวกเตอร์การเคลื่อนที่ในการประมาณการเคลื่อนที่.....	6
รูปที่ 2.1 ความสัมพันธ์ระหว่างเฟรมแบบ I, P และ B ในการทำนายการเคลื่อนที่.....	13
รูปที่ 2.2 แผนภาพของการเข้ารหัสสัญญาณวิดีโอที่มีการบีบอัด.....	14
รูปที่ 2.3 ประเภทของวิธีการประมาณการเคลื่อนที่.....	15
รูปที่ 2.4 ลักษณะการชดเชยการเคลื่อนที่.....	16
รูปที่ 2.5 การแบ่งบล็อกของเฟรมขนาด $N \times N$ .....	18
รูปที่ 2.6 ลักษณะการเข้าคู่บล็อก.....	19
รูปที่ 2.7 ลักษณะหน้าต่างการค้นหาขนาด $\pm 7$ จุดภาพ.....	19
รูปที่ 2.8 ภาพรวมของกระบวนการประมาณการเคลื่อนที่อย่างรวดเร็ว.....	22
รูปที่ 2.9 ลักษณะวิธีการค้นหาแบบสามขั้นตอน.....	23
รูปที่ 2.10 ลักษณะวิธีการค้นหาแบบลอการิทึม.....	24
รูปที่ 2.11 ลักษณะวิธีการค้นหาแบบสามขั้นตอนวิธีใหม่.....	26
รูปที่ 2.12 ลักษณะวิธีการค้นหาแบบ BBGDS.....	27
รูปที่ 2.13 ลักษณะการค้นหาแบบสี่ขั้นตอน.....	28
รูปที่ 2.14 รูปแบบการค้นหาหารูปเพชร.....	29
รูปที่ 2.15 การย้ายจุดการค้นหาของอัลกอริทึมการค้นหาหารูปเพชร.....	30
รูปที่ 3.1 รูปแบบของบล็อกข้างเคียงที่ใช้ในการวัดสหสัมพันธ์ของเวกเตอร์ การเคลื่อนที่.....	35
รูปที่ 3.2(ก) สหสัมพันธ์ของเวกเตอร์การเคลื่อนที่ระหว่างบล็อก เมื่อทดสอบกับลำดับภาพ “Carphone”.....	35
รูปที่ 3.2(ข) สหสัมพันธ์ของเวกเตอร์การเคลื่อนที่ระหว่างบล็อก เมื่อทดสอบกับลำดับภาพ “Mother and daughter”.....	36
รูปที่ 3.2(ค) สหสัมพันธ์ของเวกเตอร์การเคลื่อนที่ระหว่างบล็อก เมื่อทดสอบกับลำดับภาพ “Foreman”.....	36
รูปที่ 3.3 รูปแบบของบล็อกข้างเคียงในเชิงปริภูมิ.....	39
รูปที่ 3.4 รูปแบบของบล็อกต่าง ๆ ที่ใช้ในการจำแนกลักษณะการเคลื่อนที่.....	39
รูปที่ 3.5 รูปแบบการค้นหาแบบเพชรชนิดไม่สมมาตร.....	43

สารบัญภาพ (ต่อ)

	หน้า
รูปที่ 3.6 ลักษณะการย้ายจุดศูนย์กลางการค้นหา.....	44
รูปที่ 3.7 รูปแบบการค้นหารูปเพชร.....	44
รูปที่ 3.8 ขั้นตอนการทำงานของวิธีการค้นหาแบบเพชรชนิดไม่สมมาตรแบบปรับตัวได้ โดยใช้การปรับย้ายจุดศูนย์กลาง.....	45
รูปที่ 3.9 รูปแบบของบล็อกข้างเคียงที่ใช้ในการทำนายเวกเตอร์การเคลื่อนที่.....	47
รูปที่ 4.1 บล็อกโคโอะแกรมแสดงระบบการทดสอบที่ใช้ในวิทยานิพนธ์.....	49
รูปที่ 4.2 ชุดลำดับภาพทดสอบมาตรฐานทั้งสิบเอ็ดลำดับภาพ.....	50
รูปที่ 4.3(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Akiyo.....	58
รูปที่ 4.3(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับ ลำดับภาพ Akiyo.....	58
รูปที่ 4.3(ค) ค่าอัตราส่วนกำลังสัญญาณขอดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Akiyo.....	59
รูปที่ 4.4(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบ กับลำดับภาพ Claire.....	62
รูปที่ 4.4(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบ กับลำดับภาพ Claire.....	62
รูปที่ 4.4(ค) ค่าอัตราส่วนกำลังสัญญาณขอดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Claire.....	63
รูปที่ 4.5(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบ กับลำดับภาพ Miss America.....	65
รูปที่ 4.5(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบ กับลำดับภาพ Miss America.....	65
รูปที่ 4.5(ค) ค่าอัตราส่วนกำลังสัญญาณขอดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Miss America.....	66
รูปที่ 4.6(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบ กับลำดับภาพ Carphone.....	68
รูปที่ 4.6(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบ กับลำดับภาพ Carphone.....	68

## สารบัญญภาพ (ต่อ)

	หน้า
รูปที่ 4.6(ค) ค่าอัตราส่วนกำลังสัญญาณยอดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Carphone.....	69
รูปที่ 4.7(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบ กับลำดับภาพ Salesman.....	71
รูปที่ 4.7(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับ ลำดับภาพ Salesman.....	71
รูปที่ 4.7(ค) ค่าอัตราส่วนกำลังสัญญาณยอดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Salesman.....	72
รูปที่ 4.8(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบ กับลำดับภาพ Suzie.....	74
รูปที่ 4.8(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบ กับลำดับภาพ Suzie.....	74
รูปที่ 4.8(ค) ค่าอัตราส่วนกำลังสัญญาณยอดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Suzie.....	75
รูปที่ 4.9(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบ กับลำดับภาพ Foreman.....	77
รูปที่ 4.9(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับ ลำดับภาพ Foreman.....	77
รูปที่ 4.9(ค) ค่าอัตราส่วนกำลังสัญญาณยอดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Foreman.....	78
รูปที่ 4.10(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบ กับลำดับภาพ Trevor.....	80
รูปที่ 4.10(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบ กับลำดับภาพ Trevor.....	80
รูปที่ 4.10(ค) ค่าอัตราส่วนกำลังสัญญาณยอดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Trevor.....	81
รูปที่ 4.11(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบ กับลำดับภาพ Mother.....	83

## สารบัญภาพ (ต่อ)

	หน้า
รูปที่ 4.11(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบ กับลำดับภาพ Mother.....	83
รูปที่ 4.11(ค) ค่าอัตราส่วนกำลังสัญญาณขดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Mother.....	84
รูปที่ 4.12(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบ กับลำดับภาพ Grandmother.....	86
รูปที่ 4.12(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบ กับลำดับภาพ Grandmother.....	86
รูปที่ 4.12(ค) ค่าอัตราส่วนกำลังสัญญาณขดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Grandmother.....	87
รูปที่ 4.13(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบ กับลำดับภาพ Table.....	89
รูปที่ 4.13(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบ กับลำดับภาพ Table.....	89
รูปที่ 4.13(ค) ค่าอัตราส่วนกำลังสัญญาณขดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Table.....	90
รูปที่ 4.14 ตัวอย่างภาพที่สร้างขึ้นจากอัลกอริทึมต่าง ๆ ของลำดับภาพ Akiyo.....	92
รูปที่ 4.15 ตัวอย่างภาพที่สร้างขึ้นจากอัลกอริทึมต่าง ๆ ของลำดับภาพ Carphone.....	93
รูปที่ 4.16 ตัวอย่างสนามการเคลื่อนที่ของลำดับภาพ Carphone.....	95
รูปที่ 4.17 ตัวอย่างสนามการเคลื่อนที่ของลำดับภาพ Table tennis.....	97
รูปที่ 4.18 ตัวอย่างสนามการเคลื่อนที่ของลำดับภาพ Suzie.....	99
รูปที่ ก.1.1 ค่าความผิดพลาดกำลังสองเฉลี่ยของลำดับภาพ Akiyo เมื่อเปลี่ยนพารามิเตอร์ บางค่าในอัลกอริทึม AADS.....	108
รูปที่ ก.1.2 จำนวนจุดการค้นหาของลำดับภาพ Akiyo เมื่อเปลี่ยนพารามิเตอร์ บางค่าในอัลกอริทึม AADS.....	108
รูปที่ ก.2.1 ค่าความผิดพลาดกำลังสองเฉลี่ยของลำดับภาพ Carphone เมื่อเปลี่ยนพารามิเตอร์ บางค่าในอัลกอริทึม AADS.....	109

## สารบัญภาพ

หน้า

รูปที่ ก.2.2	จำนวนจุดการค้นหาของลำดับภาพ Carphone เมื่อเปลี่ยนพารามิเตอร์ บางค่าในอัลกอริทึม AADS.....	109
รูปที่ ก.3.1	ค่าความผิดพลาดกำลังสองเฉลี่ยของลำดับภาพ Foreman เมื่อเปลี่ยนพารามิเตอร์ บางค่าในอัลกอริทึม AADS.....	110
รูปที่ ก.3.2	จำนวนจุดการค้นหาของลำดับภาพ Foreman เมื่อเปลี่ยนพารามิเตอร์ บางค่าในอัลกอริทึม AADS.....	110



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



## บัญชีคำศัพท์

Analog	แอนะล็อก
Aspect ratio	อัตราส่วนจอภาพ
Bandwidth	แถบความถี่
Block Matching Algorithm	อัลกอริทึมการเข้าคู่บล็อก
Candidate Block	บล็อกในอดีตที่เลือกมา
Chrominance	ความเข้มสี
Compression Ratio	อัตราการบีบอัด
Current Block	บล็อกปัจจุบัน
Digital Video	วีดิทัศน์ระบบดิจิทัล
Half Pixel Motion Vector	เวกเตอร์การเคลื่อนที่แบบครึ่งจุดสี
Intra frame coding	การเข้ารหัสในเฟรม
Inter frame coding	การเข้ารหัสระหว่างเฟรม
Luminance	ความเข้มแสง
Motion Compensation	การชดเชยการเคลื่อนที่
Motion Estimation	การประมาณการเคลื่อนที่
Motion Vector	เวกเตอร์การเคลื่อนที่: ค่าที่บ่งบอกขนาดการเคลื่อนที่ของบล็อก
Multimedia	สื่อผสม
Panning	การเลื่อนกล้อง
Peak Signal to Noise Ratio(PSNR)	ค่าอัตราส่วนสัญญาณรบกวนต่อสัญญาณรบกวน
Previous Frame	เฟรมในอดีต
Random Media Access	การเข้าถึงแบบสุ่ม
Redundancy	ส่วนเกิน
Residue	เรซิดิว ภาพข้อมูลผิดพลาด ความเพี้ยนของภาพที่ประมาณขึ้น
Search Algorithm	อัลกอริทึมการค้นหา

Search Pattern	รูปแบบการค้นหา
Search Window	หน้าต่างการค้นหา
Spatial Domain	โดเมนปริภูมิ โดเมนพื้นที่
Spatial Redundancy	ส่วนเกินในโดเมนปริภูมิ
Spatial-temporal Correlation	ความสัมพันธ์ในเชิงพื้นที่และเวลา
Speed-up Ratio	อัตราการเพิ่มความเร็ว
Three Step Search	วิธีการค้นหาแบบ 3 ชั้น
Threshold	ค่าจุดเปลี่ยน
Test Sequence	ลำดับภาพทดสอบ
True Motion Vector	เวกเตอร์การเคลื่อนที่ที่แท้จริง (เวกเตอร์การเคลื่อนที่ที่ได้จากวิธีการค้นหาแบบทั้งหมด)
Variable Length Coding: VLC	การเข้ารหัสแบบหลายค่าความยาว
Video	วิดีโอ
Video Conferencing	การประชุมผ่านวิดีโอ
Zooming	การซูม

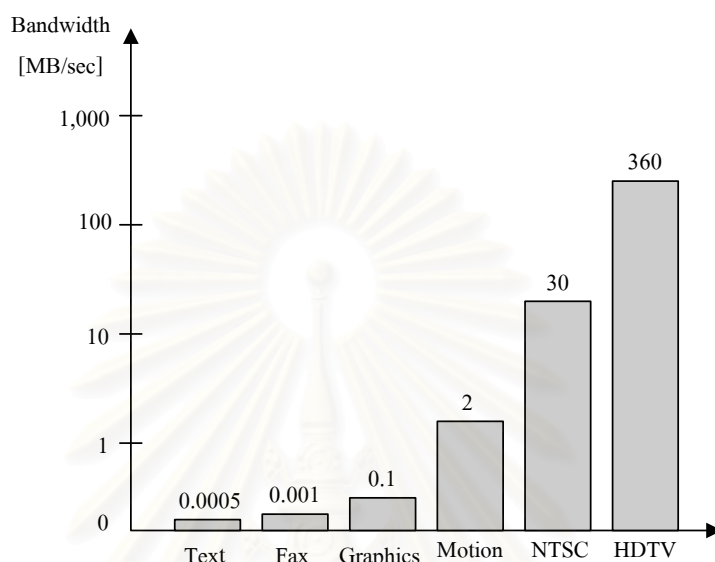
## บทที่ 1

### บทนำ

เนื่องจากในปัจจุบันนี้ แทบไม่มีแบนด์วิดท์ที่สูงพอที่จะรองรับการส่งสัญญาณวิดีโอที่ขยับไม่ได้ ทำการบีบอัดข้อมูล [1] ซึ่งแบนด์วิดท์ของโครงข่ายก็มีความสำคัญเทียบเท่ากับความต้องการหน่วยความจำ [2] รูปที่ 1.1 แสดงถึงความต้องการแบนด์วิดท์ของเทคโนโลยีแบบต่าง ๆ สมมติว่าสัญญาณวิดีโอที่ต้องการส่งเป็นรูปแบบ QCIF ใช้ข้อมูล 24 บิตต่อจุดภาพ และมีอัตราการแสดงผล 30 เฟรมต่อวินาที ต้องใช้อัตราบิตในการส่งสูงถึง 18.3 Mb/s ซึ่งเป็นตัวอย่างที่เห็นได้ชัดเจนว่า การบีบอัดสัญญาณวิดีโอมีความสำคัญเพียงใด การประมาณการเคลื่อนที่เป็นการลดความซับซ้อนของลำดับภาพวิดีโอ และเป็นส่วนสำคัญของมาตรฐานการบีบอัดสัญญาณวิดีโอ ISO/IEC MPEG-1, MPEG-2, MPEG-4 และ CCITT H.261 / ITU-T H.263 [2] ระเบียบวิธีการประมาณการเคลื่อนที่ได้รับความสนใจในการทำวิจัย เนื่องจากเป็นกระบวนการที่ต้องการการคำนวณของตัวเข้ารหัสสูง (ประมาณร้อยละ 60-80 ของเวลาทั้งหมดในการคำนวณ) ความซับซ้อนของตัวเข้ารหัสจึงขึ้นอยู่กับการประมาณการเคลื่อนที่เป็นหลัก [3] และเนื่องจาก ไม่ได้มีการจำกัดระเบียบวิธีในการหาเวกเตอร์การเคลื่อนที่เอาไว้ จึงทำให้มีการแข่งขันในด้านนี้เป็นอย่างสูงสำหรับการประมาณการเคลื่อนที่ด้วยวิธีค้นหาทั้งหมด (full search) ค่าของเวกเตอร์การเคลื่อนที่ที่เป็นไปได้ทั้งหมดจะถูกนำมาคิดค่าความเพี้ยนต่อบล็อกเพื่อเปรียบเทียบหาค่าต่ำสุด มีข้อเสียคือ ต้องใช้การคำนวณสูงเพราะต้องค้นหาทุกจุดการค้นหา ตัวอย่างเช่น ถ้าใช้กับการเข้ารหัสวิดีโอในรูปแบบ CIF ในอัตรา 30 เฟรมต่อวินาที ต้องใช้การดำเนินการทางคณิตศาสตร์ถึง 9.34 พันล้านครั้งต่อวินาที [2] จึงเป็นเรื่องที่น่าเสียดายที่ในปัจจุบันนี้ ยังไม่สามารถนำวิธีการดังกล่าวมาประยุกต์ใช้งานได้จริง โดยเฉพาะสำหรับระบบสื่อสารแบบเวลาจริง ดังนั้น ในช่วงระยะเวลาสี่ปีที่ผ่านมา ระเบียบวิธีการประมาณการเคลื่อนที่อย่างรวดเร็ว (fast motion estimation algorithm) จึงได้ถูกพัฒนาขึ้นมาเพื่อช่วยลดความซับซ้อนในการคำนวณลง อย่างไรก็ตาม ความซับซ้อนในการคำนวณที่ลดลงด้วยระเบียบวิธีเหล่านี้ มักจะทำให้คุณภาพของภาพ ลดน้อยลงไป หลาย ๆ ระเบียบวิธีที่ได้พัฒนาขึ้น ได้แก่ 2-D logarithmic search (LOGS) [4], three-step search (TSS) [5], new three-step search (NTSS) [6], four-step search (FSS) [3], unrestricted center-biased diamond search (UCBDS) [7] เป็นต้น

ในบทนี้อธิบายการใช้งานของวิดีโอระบบดิจิทัล มาตรฐานของการเข้ารหัสวิดีโอระบบดิจิทัล แนวทางในการเข้ารหัสสัญญาณวิดีโอ ความรู้เบื้องต้นของการประมาณการเคลื่อนที่ แนวทางของ

วิทยานิพนธ์ เป้าหมายและขอบเขตของวิทยานิพนธ์ ขั้นตอนและวิธีดำเนินงานวิจัย ประโยชน์ที่คาดว่าจะได้รับและเค้าโครงวิทยานิพนธ์



รูปที่ 1.1 ความต้องการแบนด์วิดท์สำหรับเทคโนโลยีประเภทต่าง ๆ

### 1.1 การใช้งานของวิดิทัศน์ระบบดิจิทัล

ช่วงเวลาหลายสิบปีที่ผ่านมาได้มีการประยุกต์และพัฒนา เทคโนโลยีทางด้านภาพและเสียง (audiovisual) ขึ้นอย่างมากมาย ทั้งยังมีความต้องการเพิ่มมากขึ้นตามไปด้วย จากการสื่อสารในอดีตของมนุษย์ เช่น โทรเลข ได้พัฒนาเป็นการสื่อสารผ่านสัญญาณวิทยุโทรทัศน์ แต่ด้วยพื้นฐานเทคโนโลยีของสัญญาณต่าง ๆ เหล่านี้เป็นแบบแอนะล็อก (Analog) จึงมีข้อจำกัดทางกายภาพ รวมถึงระบบแอนะล็อกมีการสูญเสียคุณภาพระหว่างการรับส่งได้ง่าย และเป็นจำนวนมาก รวมถึงจำกัดจำนวนข้อมูลในการส่งด้วยขนาดของความกว้างแถบ (Bandwidth) ส่งผลให้ระบบดิจิทัลซึ่งมีความทนทานต่อสัญญาณรบกวนที่สูงกว่า สามารถบีบอัดข้อมูลได้มากกว่า และคุณลักษณะบางอย่างที่ระบบแอนะล็อกไม่สามารถทำได้ เช่น การเข้าถึงแบบสุ่ม (Random media access) ความสะดวกต่อการเก็บและการเผยแพร่ ทำให้วิดิทัศน์ระบบดิจิทัลถูกนำมาใช้งานเผยแพร่สัญญาณโทรทัศน์ความละเอียดสูง (HDTV) งานทางด้านอินเทอร์เน็ตที่ฟแผ่นวีดีโอซีดีและแผ่นดีวีดี (Digital Versatile Disk: DVD) รวมไปถึงการใช้งานเพื่อการสื่อสาร เช่น การประชุมผ่านวิดิทัศน์ การส่งสัญญาณภาพเคลื่อนไหวผ่านโทรศัพท์เคลื่อนที่ [8] อย่างไรก็ตาม ความต้องการของมนุษย์ไม่เคยจำกัด ความคิดสร้างสรรค์ของมนุษย์ไม่เคยหยุดนิ่ง เป็นตัวผลักดันให้

เทคโนโลยีของวิดีโอระบบดิจิทัลได้รับการพัฒนาขึ้นอย่างต่อเนื่องและรวดเร็ว ในวิทยานิพนธ์ฉบับนี้ เกี่ยวข้องกับการเข้ารหัสสัญญาณวิดีโอระบบดิจิทัล ดังนั้นต่อไปนี้จะกล่าวสั้น ๆ ว่า การเข้ารหัสสัญญาณวิดีโอ ให้หมายถึง การเข้ารหัสสัญญาณวิดีโอในระบบดิจิทัล

## 1.2 มาตรฐานการเข้ารหัสสัญญาณวิดีโอ

มีสองหน่วยงานใหญ่ที่ทำหน้าที่ในการจัดทำมาตรฐานการเข้ารหัสสัญญาณวิดีโอ [9] ได้แก่ ITU-T และ MPEG (Motion Picture Expert Group) รายละเอียดของแต่ละหน่วยงาน มีดังต่อไปนี้

### 1.2.1 มาตรฐานของ ITU-T

ตั้งแต่ปลายทศวรรษที่ 1980 มาตรฐาน ITU-T ตระกูลเอช (H-series) ได้รับการจัดทำขึ้น โดยมีจุดประสงค์มุ่งเน้นในการใช้งานกับเครื่องโทรศัพท์ภาพ หรือ การประชุมสัมมนาทางโทรศัพท์ มาตรฐานต่าง ๆ ในตระกูลนี้ เป็นดังตารางที่ 1.1

ตารางที่ 1.1 : มาตรฐานการเข้ารหัสสัญญาณวิดีโอของ ITU-T ตระกูลเอช

มาตรฐาน	ปีที่กำหนดใช้	วัตถุประสงค์
H . 2 6 1	ค.ศ. 1990	เพื่อจัดการระบบการประชุมทางไกลผ่านระบบ ISDN (ISDN Video Conferencing)
H.262 (ร่วมกับ ISO/IEC)	ค.ศ. 1995	เหมือน MPEG-2
H.263	ค.ศ. 1996	สำหรับจัดการภาพเคลื่อนไหวผ่านระบบโทรศัพท์เคลื่อนที่ (Mobile Telephony Video)
H.263+	ค.ศ. 1998	พัฒนาเพิ่มเติมจาก H.263
H.263++	ค.ศ. 2000	พัฒนามาจาก H.263++
H.26L (H.263L)	ค.ศ. 2002	เพื่องานเกี่ยวกับภาพเคลื่อนไหวสำหรับอินเทอร์เน็ต (internet video) และการเก็บข้อมูลภาพเคลื่อนไหว (video storage and retrieval services)

## 1.2.2 มาตรฐานของ MPEG

ในปีคริสต์ศักราช 1988 ISO/IEC ได้ก่อตั้ง Moving Pictures Experts Group (MPEG) ขึ้นเพื่อกำหนดมาตรฐานในการบีบอัดสัญญาณภาพและเสียง ต้องการให้ได้การส่งและบันทึกสัญญาณภาพและเสียงที่รวดเร็ว ไม่สะดุด ที่อัตราบิต 1.5 Mbps การจัดตั้งมาตรฐานเช่นนี้ทำให้ผลิตภัณฑ์ต่าง ๆ มีการใช้งานร่วมกันได้กว้างและง่ายขึ้นกว่าเดิม ทั้งยังทำให้เกิดการแข่งขันกันในการพัฒนาตามมา มาตรฐาน MPEG ได้รับการยอมรับจาก ISO IEC/JTC SC29 ในปี 1991 ตั้งแต่นั้นมา มาตรฐานของ MPEG ได้รับการพัฒนาเรื่อยมาดังในตารางที่ 1.2 สำหรับ MPEG-3 เริ่มด้วยวัตถุประสงค์เพื่อการประยุกต์ใช้งานกับ HDTV แต่ในภายหลัง มาตรฐาน MPEG-2 ก็สามารถทำได้เช่นเดียวกัน จึงได้ตัดมาตรฐาน MPEG-3 ทิ้งไป

ตารางที่ 1.2 : มาตรฐานในการเข้ารหัสสัญญาณวิดีโอของ MPEG

มาตรฐาน	ปีที่กำหนดใช้	วัตถุประสงค์
MPEG-1	ค.ศ. 1992	เพื่อการเข้ารหัสข้อมูลภาพเคลื่อนไหวสำหรับเก็บข้อมูลในสื่อเก็บข้อมูล เช่น VCD (Video CD)
MPEG-2	ค.ศ. 1995	พัฒนาจาก MPEG-1 เพื่อการส่งกระจายสัญญาณ เช่น Digital TV, HDTV, Satellite TV และ cable TV ให้เก็บข้อมูลได้ชัดเจนและมีคุณสมบัติที่หลากหลายขึ้น
MPEG-4	ค.ศ. 1999 (Version 1 ) ค.ศ. 2000 (Version 2)	แบ่งเป็นสองส่วนใหญ่ คือ เพื่อการส่งแบบอัตราบิตต่ำมาก (very low bit rate) และเพื่อนำไปใช้งานทางอินเทอร์เน็ตแบบอินเทอร์แอคทีฟ (Interactive video) ถือได้ว่าเป็นมาตรฐานที่มุ่งเน้นในการบีบอัดเป็นมาตรฐานสุดท้ายของ MPEG
MPEG-7	ค.ศ. 2001	เพื่อขยายมาตรฐานสู่การค้นหาและจัดเก็บข้อมูลในรูปแบบของสื่อผสม (Multimedia content description interface)
MPEG-21	N/A	Multimedia Environment

### 1.3 แนวทางในการเข้ารหัสสัญญาณวิดีโอ

ไม่ว่าจะเป็นมาตรฐานในการเข้ารหัสวิดีโอแบบใดก็ตาม สิ่งสำคัญของการเข้ารหัสสัญญาณวิดีโออยู่ที่การประมาณการเคลื่อนที่ (Motion Estimation: ME) ซึ่งช่วยลดข้อมูลส่วนที่ไม่จำเป็นซึ่งเกิดขึ้นจากการที่ลำดับภาพหลาย ๆ ภาพมีความซ้ำกันในช่วงเวลาหนึ่ง (Spatial-Temporal Redundancy) ระบบใดที่มีกระบวนการการประมาณการเคลื่อนที่ที่มีความถูกต้องมากก็มีความสามารถในการบีบอัดข้อมูลได้สูงตามไปด้วย วิธีการประมาณการเคลื่อนที่ที่ใช้กันอย่างแพร่หลายในปัจจุบัน คือ อัลกอริทึมการเข้าคู่บล็อก (Block Matching Algorithm: BMA) เนื่องจากวิธีนี้ไม่ยุ่งยากซับซ้อนมากนักและยังมีประสิทธิภาพ ซึ่งใช้เป็นเทคนิคการประมาณการเคลื่อนที่ในวิทยานิพนธ์ฉบับนี้ด้วย มาตรฐานในการเข้ารหัสทั่วไปนิยมใช้เทคนิคการค้นหาเวกเตอร์การเคลื่อนที่ (Motion Vector: MV) [9] ซึ่งอัลกอริทึมการค้นหาแบบทั้งหมด (Full Search Algorithm) เป็นวิธีการที่ให้ความถูกต้องสูงมากที่สุด ซึ่งในปัจจุบันมีงานวิจัยทางด้านเทคนิคการค้นหาเวกเตอร์การเคลื่อนที่เป็นจำนวนมาก การประมาณการเคลื่อนที่โดยใช้วิธีการค้นหาแบบทั้งหมดต้องคำนวณหาค่าความเพี้ยนต่อบล็อก (Block Distortion Measure: BDM) ทุกบล็อกที่เกิดจากการค้นหา (Searchpoint) ที่เป็นไปได้ทั้งหมดในหน้าต่างการค้นหา (Search Window) ทำให้ต้องการการคำนวณสูงและใช้เวลาในการเข้ารหัสนาน ทำให้ไม่สามารถนำมาประยุกต์ใช้งานในแบบเวลาจริง (Real Time) ได้ ถึงแม้ว่าวิธีการดังกล่าวให้ความถูกต้องสูงที่สุดในการประมาณการเคลื่อนที่ก็ตาม

### 1.4 ความรู้เบื้องต้นเกี่ยวกับการประมาณการเคลื่อนที่

เป็นการหาความแตกต่างระหว่างจุดภาพในตำแหน่งที่สอดคล้องกัน (มีพิกัดทางตำแหน่งเดียวกัน) ระหว่างเฟรมสองเฟรมที่ต่อเนื่องกัน ส่งผลให้สามารถทำการเข้ารหัสเพียง “ค่าของความแตกต่าง” ระหว่างสองเฟรมนั้น แทนที่จะต้องเข้ารหัสด้วยภาพทั้งภาพ [10] ซึ่งถ้าค่าความแตกต่างดังกล่าวมีค่าน้อยเท่าใดก็ทำให้การบีบอัดสัญญาณภาพยิ่งมีประสิทธิภาพมากขึ้นเท่านั้น โดยทั่วไป เราสามารถสมมุติว่าค่าของจุดภาพในภาพที่มีการเคลื่อนที่ของวัตถุได้เลื่อนตำแหน่งจากจุดหนึ่งในเฟรมหนึ่ง ไปยังจุดอื่นในเฟรมถัดไป ถ้าเราสามารถหาตำแหน่งที่เลื่อนไปได้ เราก็สามารถลดค่าความแตกต่างลงไปได้ วิธีนี้เรียกว่า การชดเชยการเคลื่อนที่ (motion compensation) ซึ่งในสัญญาณวิดีโอส่วนใหญ่ เฟรมที่อยู่ติด ๆ กันมักจะมีความคล้ายคลึงกันเป็นอย่างมาก หรือกล่าวได้ว่า ลำดับภาพมีความซ้ำซ้อนกันทางเวลา ยกตัวอย่างเช่น ภาพของผู้ประกาศข่าวมีฉากหลังที่ซ้ำซ้อนนั่นเอง การชดเชยการเคลื่อนที่จึงเกิดขึ้นเพื่อลดความซ้ำซ้อนในส่วนนี้ แต่การชดเชยการเคลื่อนที่กับทุกจุดภาพไม่เป็นการง่ายต่อการบีบอัด จึงได้คิดค้น

วิธีการชดเชยการเคลื่อนที่โดยการจัดกลุ่มให้กับจุดภาพ (กลุ่มของจุดภาพนี้ เรียกว่าบล็อก) ค่าของตำแหน่งที่เลื่อนไปนี้จะแสดงในรูปของตัวเลขสองตัว ซึ่งแสดงถึงการเคลื่อนที่ในแนวแกนอนและแนวแกนตั้งตามลำดับ เรียกว่า เวกเตอร์การเคลื่อนที่ (motion vector)

ส่วนใหญ่แล้วขนาดของบล็อกที่ใช้ในการชดเชยการเคลื่อนที่ มีขนาด  $8 \times 8$  หรือ  $16 \times 16$  จุดภาพ ดังตัวอย่างในรูปที่ 1.2 [10] แสดงถึงบล็อกข้อมูลขนาด  $8 \times 8$  ในตำแหน่งที่สอดคล้องกันระหว่างเฟรมสองเฟรมที่อยู่ลำดับติดกัน โดยบล็อกข้อมูลในรูปที่ 1.2(ก) แสดงถึงภาพในเฟรมอ้างอิงและบล็อกข้อมูลและรูปที่ 1.2(ข) แสดงถึงภาพในเฟรมถัดไป จากตัวอย่างถ้าคำนวณหาความแตกต่างของจุดสีระหว่างบล็อกในบริเวณตำแหน่งที่สอดคล้องกันตรง ๆ จะได้ค่าความแตกต่างไม่เป็นศูนย์ทั้งหมด แต่ถ้าลองเปรียบเทียบความแตกต่างโดยใช้บล็อกข้อมูลที่อยู่ในกรอบเส้นประของเฟรมอ้างอิงพบว่าได้ค่าความแตกต่างเป็นศูนย์ทุกจุดภาพภายในเฟรม ค่าของตำแหน่งที่เลื่อนไปนี้เรียกว่า เวกเตอร์การเคลื่อนที่นั่นเอง นั่นคือ เวกเตอร์การเคลื่อนที่ในแนวแกนอนและแกนตั้ง ก็คือ สองจุดสีทางด้านขวาและหนึ่งจุดสีทางด้านล่าง ตามลำดับ

40	42	43	47	51	47	54	53	52	49
39	41	45	45	50	45	58	56	50	48
40	42	43	50	43	45	50	56	56	53
41	43	43	53	48	50	61	53	63	56
42	54	48	50	58	53	53	48	68	50
43	44	45	48	56	56	50	45	68	61
47	46	48	53	50	61	61	45	61	50
48	49	50	53	56	56	58	53	63	61
50	51	50	48	56	56	63	53	61	61

(ก) ค่าของจุดภาพจากเฟรมอ้างอิง

45	45	50	45	58	56	50	48
43	50	43	45	50	56	56	53
43	53	48	50	61	53	63	56
48	50	58	53	53	48	68	50
45	48	56	56	50	45	68	61
48	53	50	61	61	45	61	50
50	53	56	56	58	53	63	61
50	48	56	56	63	53	61	61

(ข) ค่าของจุดภาพในตำแหน่งเดิมของเฟรมถัดไป

รูปที่ 1.2 ตัวอย่างการหาเวกเตอร์การเคลื่อนที่ในการประมาณการเคลื่อนที่



## 1.5 แนวทางของวิทยานิพนธ์

ในยุคของข้อมูลข่าวสาร ตลาดสำหรับการประยุกต์ใช้งานสื่อผสม เช่น โทรทัศน์ดิจิทัลแบบไร้สาย เติบโตขึ้นอย่างรวดเร็ว แต่เนื่องจากปัญหาของข้อจำกัดในด้านความสามารถของระบบ และข้อจำกัดของกำลังในการใช้งานที่ต้องใช้กำลังต่ำในอุปกรณ์บีบอัดสัญญาณดิจิทัลแบบ VLSI การเข้ารหัสสัญญาณดิจิทัลตามมาตรฐาน MPEG-4 จึงเป็นอีกแนวทางในการส่งผ่านสัญญาณ อย่างไรก็ตาม กระบวนการประมาณการเคลื่อนที่ที่ใช้ในมาตรฐานดังกล่าวอาจใช้กำลังในการคำนวณสูงถึงร้อยละ 80 ของตัวเข้ารหัส [11] จึงผลักดันให้เกิดงานวิจัยจำนวนมาก เพื่อลดความซับซ้อนในการประมาณการเคลื่อนที่ลงอย่างต่อเนื่อง ดังนั้นวิทยานิพนธ์นี้จึงเสนอการนำกระบวนการทำนายเวกเตอร์การเคลื่อนที่มาใช้ในการปรับรูปแบบการค้นหาอัลกอริทึมรูปเพชร ร่วมกับการปรับย้ายตำแหน่งเริ่มต้น เพื่อมุ่งเน้นในการลดความซับซ้อนในการประมาณการเคลื่อนที่ โดยให้คุณภาพของสัญญาณภาพไม่ด้อยไปกว่าวิธีประมาณการเคลื่อนที่อย่างรวดเร็วชนิดอื่น ๆ มาก

## 1.6 เป้าหมายและขอบเขตวิทยานิพนธ์

พัฒนาวิธีการประมาณการเคลื่อนที่ (Motion Estimation) แบบใช้อัลกอริทึมการเข้าคู่บล็อก ด้วยการปรับรูปแบบอัลกอริทึมการค้นหารูปเพชรในการค้นหาเวกเตอร์การเคลื่อนที่ร่วมกับการปรับย้ายจุดศูนย์กลางการค้นหาเพื่อมุ่งเน้นในการลดความซับซ้อนในการคำนวณ (Computational Complexity) เมื่อเปรียบเทียบกับจำนวนจุดการค้นหาที่ใช้กับเทคนิคการค้นหาอย่างรวดเร็วชนิดอื่น ๆ และรักษาความถูกต้องในการประมาณการเคลื่อนที่ โดยอ้างอิงจากผลการประมาณด้วยอัลกอริทึมแบบค้นหาทั้งหมด

## 1.7 ขั้นตอนและวิธีดำเนินงาน

1. ศึกษาความรู้พื้นฐานในการเข้ารหัสสัญญาณดิจิทัลและการประมาณการเคลื่อนที่
2. ศึกษาระเบียบวิธีและอัลกอริทึมที่ใช้ในการค้นหาชนิดต่าง ๆ
3. ศึกษาและวิเคราะห์ลักษณะของเวกเตอร์การเคลื่อนที่เพื่อให้สามารถนำมาใช้ประโยชน์ในการปรับรูปแบบการค้นหา
4. พัฒนาอัลกอริทึมการค้นหาด้วยวิธีที่นำเสนอ
5. ทดสอบวิธีที่นำเสนอกับอัลกอริทึมในงานวิจัยที่มีอยู่แล้ว เช่น การค้นหาแบบทั้งหมด การค้นหาแบบสาม

ขั้นตอนและการค้นหาแบบเพชฌฆาต เป็นต้น โดยใช้ชุดภาพทดสอบมาตรฐานต่าง ๆ พร้อมทั้งวิเคราะห์ผล

6. สรุปและรวบรวมผลงานวิจัย พร้อมทั้งจัดทำรูปเล่มวิทยานิพนธ์

### 1.8 ประโยชน์ที่คาดว่าจะได้รับ

1. มีความรู้พื้นฐานและมีความเข้าใจเทคนิคต่าง ๆ ของการประมวลการเคลื่อนที่
2. ซอฟต์แวร์อัลกอริทึมเพื่อใช้ประมวลการเคลื่อนที่ในการเข้ารหัสสัญญาณวิดีโอ
3. สามารถนำไปเป็นแนวความคิดในการใช้งานที่ต้องการลดความซับซ้อนในการคำนวณ

### 1.9 โครงสร้างวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้แบ่งรายละเอียดออกเป็น 5 บท ดังต่อไปนี้

บทที่ 1 บทนำ กล่าวถึงการใช้งานวิดีโอระบบดิจิทัล มาตรฐานการเข้ารหัสสัญญาณวิดีโอ ความรู้เบื้องต้นของการประมวลการเคลื่อนที่ เป้าหมายและขอบเขตของวิทยานิพนธ์ ขั้นตอนและวิธีการดำเนินงาน และ ประโยชน์ที่คาดว่าจะได้รับ

บทที่ 2 ความรู้พื้นฐาน กล่าวถึงวิดีโอระบบดิจิทัลที่มีในปัจจุบัน การเข้ารหัสสัญญาณวิดีโอ การลดความซับซ้อนเชิงปริภูมิ การลดความซับซ้อนทางเวลา การประมวลการเคลื่อนที่ การชดเชยการเคลื่อนที่ อัลกอริทึมการเข้ารหัสบล็อก เทคนิคการประมวลการเคลื่อนที่และงานวิจัยที่มีอยู่แล้ว การวัดสมรรถนะของระเบียบวิธี

บทที่ 3 การค้นหาแบบเพชฌฆาตไม่สมมาตรแบบปรับตัวได้โดยใช้การปรับย้ายจุดศูนย์กลาง กล่าวถึง สหสัมพันธ์ของเวกเตอร์การเคลื่อนที่ระหว่างบล็อก การจำแนกลักษณะการเคลื่อนที่ และการทำนายเวกเตอร์การเคลื่อนที่ เทคนิคการค้นหาแบบเพชฌฆาตไม่สมมาตร เทคนิคการปรับรูปแบบการค้นหา การปรับย้ายจุดศูนย์กลางการค้นหา ขั้นตอนในการค้นหาและอัลกอริทึมการค้นหาแบบเพชฌฆาตไม่สมมาตร

บทที่ 4 ผลการวิจัย กล่าวถึง ผลการทดลองเพื่อทดสอบแนวความคิด ผลของเทคนิคที่นำเสนอ รวมทั้งผลของการทดสอบเปรียบเทียบกับงานวิจัยที่ผ่านมาเกี่ยวกับลำดับภาพทดสอบมาตรฐานต่าง ๆ

บทที่ 5 บทสรุป กล่าวถึง สรุปผลการวิจัย ข้อเสนอแนะ

## บทที่ 2

### ความรู้พื้นฐาน

เนื้อหาในบทนี้กล่าวถึง หลักการพื้นฐานของวิดีโอ ดิจิทัล ทฤษฎีการเข้ารหัสสัญญาณ วิดีทัศน์ การลดความซ้ำซ้อนเชิงปริภูมิ การลดความซ้ำซ้อนเชิงเวลา การประมาณการเคลื่อนที่และการชดเชยการเคลื่อนที่ อัลกอริทึมการเข้ารหัสแบบบล็อก ตัวอย่างงานวิจัยในด้านการประมาณการเคลื่อนที่อย่างรวดเร็วที่ผ่านมา และการวัดสมรรถนะของอัลกอริทึม

#### 2.1 วิดีทัศน์ดิจิทัล (Digital Video)

ภาพดิจิทัล ประกอบด้วย ส่วนความเข้มแสง (luminance) และส่วนความเข้มสี (chrominance) ซึ่งมาจากกล้องถ่ายภาพระบบดิจิทัล หรือการสแกนรูปจากภาพแอนะล็อก ข้อมูลเหล่านี้อยู่ในรูปแบบของจุดภาพต่อความยาวหนึ่งหน่วย เช่น จุดภาพต่อนิ้ว หรือ จุดภาพต่อเซนติเมตร โดยแต่ละจุดภาพได้มาจากการสุ่มตัวอย่างและผ่านการควอนไทซ์ โดยทั่วไป ความเข้มแสงของแต่ละภาพเมื่อผ่านการควอนไทซ์แล้วถูกแบ่งออกเป็น 256 ระดับ ดังนั้น จุดภาพแต่ละจุดสามารถแสดงด้วย ความเข้มแสง ตั้งแต่ 0 ถึง 255 ซึ่งแทนระดับสีเทา (gray level) ตั้งแต่สีดำถึงสีขาว เป็นจำนวน 256 ระดับ หรือ 8 bits สำหรับความเข้มสีนั้นไม่มีความจำเป็นต้องแสดงทุกจุดภาพเนื่องจากความสามารถในการรับสีของระบบประสาทในการมองเห็นของมนุษย์น้อยกว่าความเข้มแสง จึงมีจำนวนจุดความเข้มสีแล้วแต่กำหนด [8]

วิดีโอทัศน์ดิจิทัล เป็นกลุ่มของภาพดิจิทัลที่มีหน่วยเป็น เฟรมต่อหนึ่งหน่วยเวลา เช่น เฟรมต่อวินาที (frame per second) รูปแบบของสัญญาณวิดีโอทัศน์บางชนิดกำหนดไว้ดังตารางที่ 2.1 [12]

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 2.1 รูปแบบของสัญญาณวิดีโอที่ใช้งานในปัจจุบัน

Application	Luminance Resolution	Chrominance Resolution	Aspect Ratio	Temporal (frames/s)	Bitrate (Mbit/s)
HDTV	1920 x 1152	960 x 576	16/9	50	1800
TV (broadcast)	720 x 576	360 x 576	4/3	25	166
TV (CD record)	360 x 288	180 x 144	4/3	25	31
Video phone	360 x 288	180 x 144	4/3	10	12.4
Mobile Video	180 x 144	90 x 72	4/3	5	1.6

## 2.2 การเข้ารหัสสัญญาณวิดีโอ (Video Coding)

กระบวนการสำคัญในการเข้ารหัสสัญญาณวิดีโอมีอยู่สองส่วน ได้แก่ การเข้ารหัสช่องสัญญาณ (channel coding) และการเข้ารหัสแหล่งกำเนิดสัญญาณ (source coding)

การเข้ารหัสช่องสัญญาณต้องการความเชื่อมั่นสูงและตรวจสอบความผิดพลาดได้ ส่วนการเข้ารหัสแหล่งกำเนิดสัญญาณมุ่งเน้นในด้านการบีบอัดข้อมูล การเข้ารหัสสัญญาณวิดีโอแบบบีบอัดข้อมูล แบ่งตามชนิดข้อมูลได้สองประเภท ดังนี้

### 2.2.1 การเข้ารหัสภายในเฟรม (Intra-Frame Coding)

การเข้ารหัสชนิดนี้ลดส่วนเกินในโดเมนปริภูมิ (Spatial Redundancy) ของเฟรมภาพด้วยการบีบอัด โดยอาศัยความซ้ำซ้อนของข้อมูลภายในหนึ่งเฟรม โดยใช้เทคนิคต่าง ๆ ดังนี้

1. การสุ่มข้อมูล (Subsampling) คือ แทนการเข้ารหัสจุดภาพทุก ๆ จุด (pixel) ก็ใช้วิธีเข้ารหัสเพียงแค่บางจุดเท่านั้น หรือเข้ารหัสด้วยค่าเฉลี่ยของกลุ่มจุดภาพ เพื่อช่วยลดอัตราบิตในการส่ง แต่ผลเสียก็คือ ความคมของภาพจะลดลง สำหรับจุดภาพที่ไม่ถูกเข้ารหัสก็จะถูกสร้างขึ้นที่ภาครับโดยการประมาณ (Interpolation)
2. การควอนไทซ์แบบหยาบหรือการลดความลึกของจุดภาพ (Coarse Quantization or Depth Reduction) เป็นการควอนไทซ์โดยใช้จำนวนบิตต่ำลง แต่จะทำให้ข้อมูลบางส่วนของภาพ

หายไป เช่น ในภาพขาวดำ ส่วนที่เป็นสีเทาจะใช้วิธีนี้ในการควอนไทซ์ เนื่องจากสายตาของมนุษย์ส่วนใหญ่ไม่สามารถเห็นความแตกต่างได้ชัดเจน

3. การเข้ารหัสแบบการแปลง (Transform Coding) เป็นการแปลงข้อมูลของภาพจากโดเมนปริภูมิ (Spatial Domain) เป็นโดเมนความถี่ (Frequency Domain)
4. การทำควอนไทซ์แบบเวกเตอร์ (Vector Quantization) ข้อมูลภาพจะถูกแบ่งกลุ่มและแสดงแทนด้วยสัญลักษณ์จากชุดรหัส (codebook) เพื่อทำการส่งสัญลักษณ์แทนเท่านั้น ซึ่งเป็นการประหยัดบิตในการส่งได้เช่นกัน

### 2.2.2 การเข้ารหัสระหว่างเฟรม (Inter-Frame Coding)

เนื่องจากในสัญญาณวิดีโอทัศน์มีคุณลักษณะที่ว่า เฟรมภาพที่อยู่ติดกันมักจะมีความสัมพันธ์กันระหว่างข้อมูลของภาพทางปริภูมิ-เวลา (spatial temporal correlation) ซึ่งลักษณะดังกล่าวเรียกว่าเป็นความซ้ำซ้อนของข้อมูลทางปริภูมิ-เวลา (spatial temporal redundancy) ดังนั้นจึงเกิดกรรมวิธีในการลดข้อมูลเหล่านี้ เครื่องมือที่ใช้ในการเข้ารหัสมีหลายเทคนิค เช่น

1. การเข้ารหัสแบบสุ่มทางเวลา (Time Subsampling) ด้วยการเข้ารหัสไม่ครบทุกลำดับเฟรมภาพ แต่ใช้ตัวถอดรหัสช่วยสร้างเฟรมที่หายไปขึ้นแทน
2. การเข้ารหัสค่าความแตกต่าง (Different Coding) ด้วยการเข้ารหัสเพียงแค่ค่าความแตกต่างระหว่างเฟรมที่อยู่ติดกันแทนการเข้ารหัสทั้งเฟรมภาพ เพราะค่าความแตกต่างมีข้อมูลน้อยกว่า เนื่องจากภาพมีความคล้ายคลึงกันมาก แต่อาจต้องใช้ ส่วนเพิ่มเติม (overhead) ในการบ่งบอกว่าจุดภาพใดมีการเปลี่ยนแปลง จุดภาพใดไม่มีการเปลี่ยนแปลง
3. การเข้ารหัสค่าความแตกต่างระหว่างบล็อก (Block-Based Different Coding) วิธีการนี้คล้ายกับการเข้ารหัสความแตกต่างระหว่างเฟรม ต่างกันตรงที่ ภาพจะถูกแบ่งออกเป็นบล็อกย่อย ๆ ก่อน หลังจากนั้นแต่ละบล็อกจะถูกทำการเปรียบเทียบ
4. การประมาณการเคลื่อนที่ และการชดเชยการเคลื่อนที่ (Motion Estimation and Motion Compensation) เป็นวิธีการที่สร้างขึ้นเพื่อลดข้อมูลที่ต้องใช้ในการส่งสัญญาณวิดีโอทัศน์ ข้อมูลส่วนที่ต้องส่งไป คือ ภาพอ้างอิง (Reference Image) เวกเตอร์การเคลื่อนที่ (Motion Vector) และ ค่าความผิดพลาดของภาพ (Residue) สังเกตได้ว่า ยิ่งข้อมูลภาพแสดงความผิดพลาดน้อยเท่าใด การเข้ารหัสระหว่างเฟรมก็ยิ่งบีบอัดได้มากขึ้นเท่านั้น หรืออีกแง่มุมหนึ่งคือ ยิ่งทำการประมาณการเคลื่อนที่ได้แม่นยำมากขึ้นก็ส่งผลให้มีความผิดพลาดน้อยลงตามมา ทำให้บีบอัด

ได้มากขึ้นเช่นกัน ดังนั้น การประมาณการเคลื่อนที่จึงมีบทบาทสำคัญในส่วนของ การบีบอัดข้อมูลสัญญาณวิดีโอ

### 2.3 การลดความซ้ำซ้อนทางปริภูมิ (Spatial Redundancy Reduction)

ในการลดความซ้ำซ้อนทางปริภูมิของภาพ มาตรฐาน MPEG ใช้วิธีการเข้ารหัสด้วยพื้นฐานแบบการแปลงดิคกริตโคซายน์ (DCT-based Coding Technique) สมการ DCT 2 มิติขนาด  $8 \times 8$  สามารถเขียนในรูปค่าของจุดภาพ  $f(i,j)$  ดังนี้

$$F(u,v) = \frac{1}{4} C(u)C(v) \sum_{i=0}^7 \sum_{j=0}^7 f(i,j) \cos\left(\frac{(2i+1)u\pi}{16}\right) \cos\left(\frac{(2j+1)v\pi}{16}\right) \quad (2.1)$$

$$\text{โดย } C(x) = \begin{cases} \frac{1}{\sqrt{2}} & , x=0 \\ 1 & , x \neq 0 \end{cases} \quad (2.2)$$

ค่าสัมประสิทธิ์ DCT ที่ผ่านการแปลงแล้วจะถูกควอนไทซ์เพื่อลดจำนวนบิต และเพิ่มจำนวนสัมประสิทธิ์ที่เป็นศูนย์ การทำควอนไทซ์ร่วมกับการเข้ารหัสตามความยาว (run-length coding) ช่วยให้บีบอัดข้อมูลได้มาก [13]

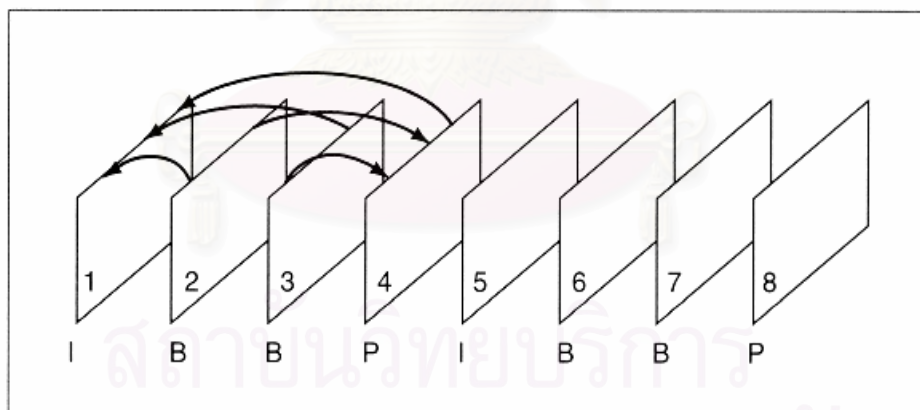
### 2.4 การลดความซ้ำซ้อนทางเวลา (Temporal Redundancy Reduction)

วิธีการทำนายในทางเวลา (temporal prediction) ช่วยให้รองรับความสามารถในการเข้าถึงแบบสุ่มของสัญญาณวิดีโอที่เก็บไว้ ในขณะที่ยังคงต้องการลดความซ้ำซ้อน (redundancy) ให้ได้มากที่สุด ตามมาตรฐาน MPEG กำหนดชนิดของรูปภาพไว้สามแบบ ได้แก่ intra ( $I$ ) pictures, predicted ( $P$ ) pictures และ bidirectionally interpolated ( $B$ ) pictures [14] โดยมีรายละเอียด ดังนี้

1. Intra ( $I$ ) pictures เป็นภาพที่เข้ารหัสโดยไม่ต้องใช้ข้อมูลของภาพจากเฟรมอื่น ดังนั้น ในลำดับภาพวิดีโอ เฟรมแรกจะเป็นภาพแบบ I เสมอ โดยเป็นเฟรมที่ถูกบีบอัดภายในเฟรม ใช้เป็นจุดเข้าถึง (access point) ของการเข้าถึงแบบสุ่ม ภาพแบบ I มีอัตราการบีบอัดข้อมูลต่ำที่สุดเมื่อเปรียบเทียบกับภาพทั้งสามแบบ

2. Predicted (*P*) pictures เฟรมแบบนี้สามารถถอดรหัสโดยใช้ข้อมูลจากเฟรมอื่นที่ได้แสดงไปก่อนหน้า ซึ่งภาพที่แสดงไปก่อนหน้า เรียกว่า ภาพอ้างอิง (reference picture) โดยสามารถเป็นเฟรมแบบ I หรือว่าจะเป็นเฟรมแบบ P ด้วยกันเองก็ได้ สำหรับบล็อกรหัสของภาพที่ได้มาจากภาพอ้างอิงจะต้องระบุเวกเตอร์การเคลื่อนที่มาด้วย ผลของการบีบอัดภาพแบบ P นี้จะทำให้ขนาดของเฟรมแบบ P มีขนาดประมาณร้อยละ 30 ถึง 50 ของเฟรมแบบ I
3. Bidirectionally interpolated (*B*) pictures เป็นภาพที่ใช้การอ้างอิงข้อมูลทั้งจากอดีตและอนาคต รวมทั้งค่าการเคลื่อนที่ของเฟรมดังกล่าวทั้งสองเฟรม โดยละเลยค่าความเพี้ยนของภาพที่ได้จากการประมาณ ดังนั้นเฟรมชนิดนี้จะไม่ใช้เป็นเฟรมอ้างอิง ภาพชนิดนี้จึงให้ผลของการบีบอัดข้อมูลที่สูงที่สุด กล่าวคือ มีขนาดเหลือเพียงร้อยละ 50 ของภาพแบบ P เท่านั้น เฟรมแบบนี้มุ่งเน้นเพื่อบีบอัดแบบอัตราบิตต่ำ

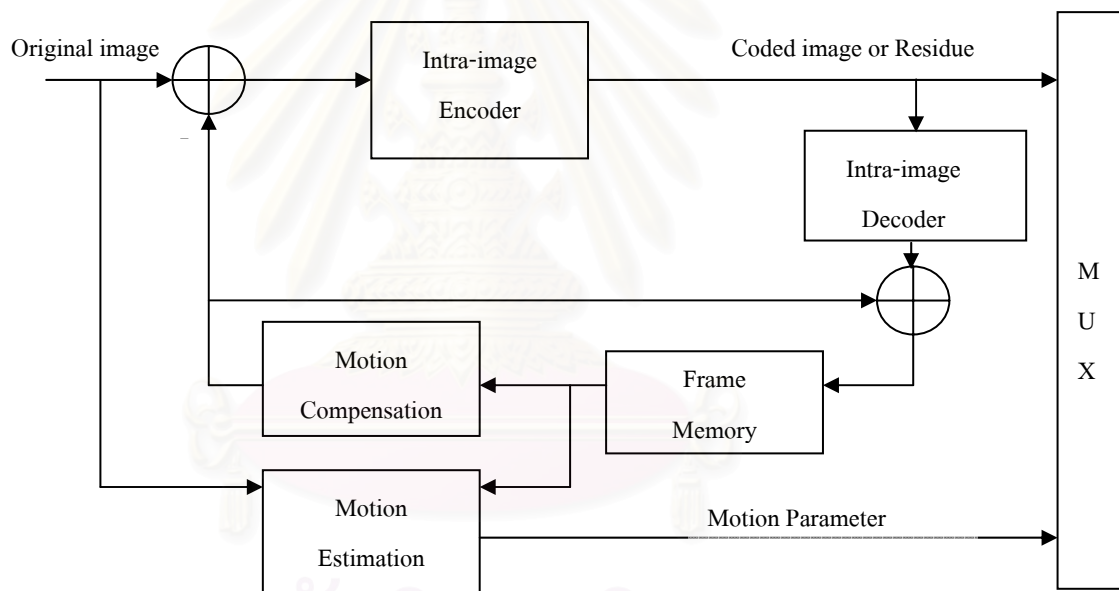
ตามมาตรฐาน MPEG ไม่ได้กำหนดจำนวนภาพแบบ B ในระหว่างสองเฟรมอ้างอิงใด ๆ รวมทั้งไม่ได้กำหนดจำนวนรูปภาพระหว่างภาพแบบ I สองภาพใด ๆ [14] ความสัมพันธ์ระหว่างภาพแบบ I, P และ B เป็นดังรูปที่ 2.1 ภาพเฟรมลำดับที่ 4 ซึ่งเป็นภาพแบบ P ได้จากการอ้างอิงเฟรมลำดับที่ 1 (เฟรมแบบ I) ส่วนเฟรมแบบ B ทั้งสองเฟรมได้จากเฟรมแบบ I และ P ที่เฟรมเหล่านี้อยู่ระหว่างกลาง



รูปที่ 2.1 ความสัมพันธ์ระหว่างเฟรมแบบ I, P และ B ในการทำนายการเคลื่อนที่

## 2.5 การประมาณการเคลื่อนที่ (Motion Estimation)

หลักการของการประมาณการเคลื่อนที่ คือ การตรวจจับการเคลื่อนที่ของวัตถุในเฟรมระหว่างสองเฟรมใด ๆ ซึ่งค่าการเคลื่อนที่นี้นำไปใช้สร้างเฟรมถัดไปได้ นอกจากนี้ยังสามารถใช้เวกเตอร์การเคลื่อนที่ที่ได้นี้ไปวิเคราะห์เพื่อการประยุกต์ใช้งานด้านอื่นได้อีก เช่น การแบ่งภาพ การประมาณการเคลื่อนที่ที่ใช้ข้อมูลของภาพเพียงบางเฟรมร่วมกับเวกเตอร์การเคลื่อนที่ในการสร้างข้อมูลของภาพเฟรมอื่น ๆ ได้ โดยทั่วไป การประมาณการเคลื่อนที่นี้ทำควบคู่ไปกับการชดเชยการเคลื่อนที่ โดยการประมาณการเคลื่อนที่ทำงานที่ด้านส่งสัญญาณเพื่อเข้ารหัสออกมาเป็นเวกเตอร์การเคลื่อนที่ ซึ่งมีการชดเชยการเคลื่อนที่ดังรูปที่ 2.2 เพื่อช่วยในการหาค่าความผิดพลาดตกค้าง (residue) จากนั้นที่ด้านรับสัญญาณก็มีการชดเชยการเคลื่อนที่อีกเพื่อทำหน้าที่สร้างภาพประมาณ (estimated picture) จากภาพอ้างอิง (reference picture) ร่วมกับเวกเตอร์การเคลื่อนที่ [8]



รูปที่ 2.2 แผนภาพของการเข้ารหัสสัญญาณวิดีโอที่มีการบีบอัด

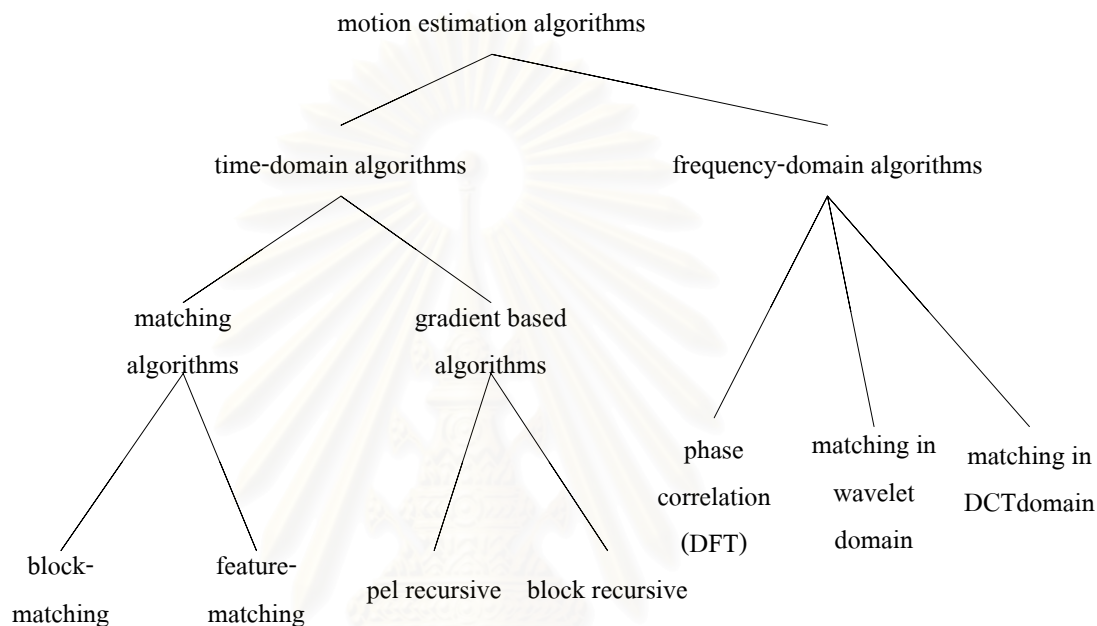
การประมาณการเคลื่อนที่เป็นหัวข้อที่เป็นที่สนใจทั้งทางการวิจัยและในทางอุตสาหกรรม เนื่องจากสาเหตุที่ว่า

1. การประมาณการเคลื่อนที่ที่ต้องการการคำนวณส่วนใหญ่จากตัวเข้ารหัส (ประมาณร้อยละ 60-80 ของเวลาทั้งหมดที่ใช้ในการคำนวณ) ซึ่งเป็นตัวจำกัดสมรรถนะของตัวเข้ารหัส
2. มีผลกระทบอย่างมากต่อคุณภาพของภาพสำหรับอัตราบิตที่กำหนด
3. ยังไม่มีการกำหนดมาตรฐานในการหาเวกเตอร์การเคลื่อนที่ เป็นการชักนำให้เกิดการแข่งขัน



นอกจากนี้ การประมาณการเคลื่อนที่ยังสามารถนำไปประยุกต์ใช้งานอื่น ๆ ได้อีก เช่น การตัดแบ่งภาพ (motion segmentation) และการวิเคราะห์สัญญาณวิดีโอ (video analysis) เป็นต้น

การประมาณการเคลื่อนที่ที่มีอยู่สามารถแบ่งได้เป็นวิธีการในเชิงเวลา กับ เชิงความถี่ ดังในรูปที่ 2.3 [2] ซึ่งในวิทยานิพนธ์ฉบับนี้ใช้วิธีการของอัลกอริทึมการเข้าคู่บล็อกในโดเมนเวลาเพราะมีความซับซ้อนของอัลกอริทึมไม่มากนักและให้ความถูกต้องในระดับที่ยอมรับได้

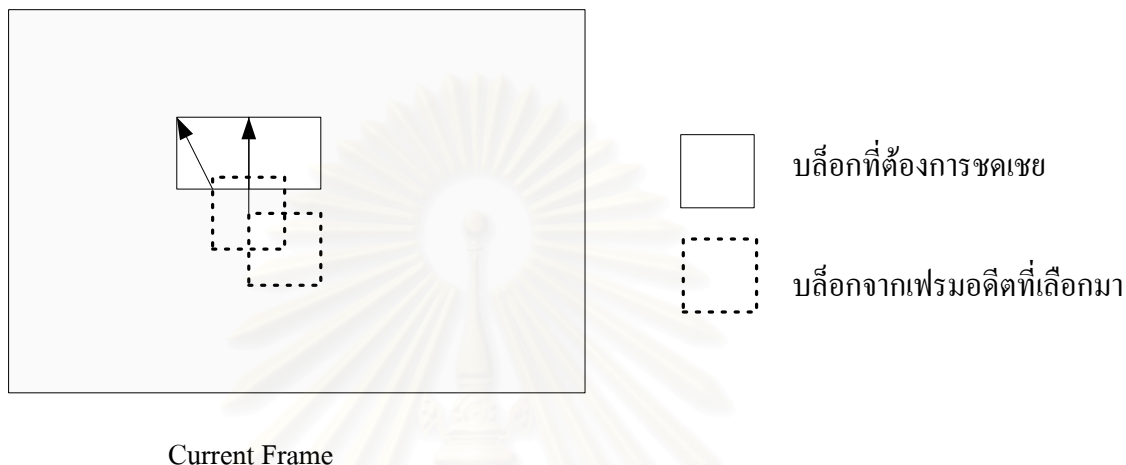


รูปที่ 2.3 ประเภทของวิธีการประมาณการเคลื่อนที่

## 2.6 การชดเชยการเคลื่อนที่ (Motion Compensation)

ด้วยสมมติฐานว่า แต่ละบล็อกของข้อมูลภาพประกอบจากจุดภาพที่มีความเกี่ยวข้องกันมาก ทำให้มีความน่าจะเป็นที่การเคลื่อนที่ภายในเฟรมทำให้จุดภาพส่วนใหญ่ภายในบล็อกเคลื่อนที่ไปด้วยระยะทางและทิศทางที่สอดคล้องกัน [15] หรือกล่าวได้ว่า เฟรมหนึ่ง ๆ มีความน่าจะเป็นที่เหมือนกับเฟรมก่อนหน้า ซึ่งเป็นจริงสำหรับลำดับภาพที่มีการเคลื่อนที่ไม่มาก การเปลี่ยนแปลงเพียงบางส่วนของภาพทำให้สามารถทำนายภาพจากเฟรมอดีตได้โดยตรง เป็นการลดความซ้ำซ้อนทางเวลาของภาพในแบบ P และ B ดังที่ได้กล่าวมาข้างต้น ซึ่งในทางทฤษฎีแล้ว การเก็บข้อมูลเพียงแค่ค่าความแตกต่างระหว่างเฟรมปัจจุบันกับเฟรมที่ทำนายได้จะใช้เนื้อที่น้อยกว่าการเก็บข้อมูลทั้งภาพ

รูปที่ 2.4 แสดงตัวอย่างของการชดเชยการเคลื่อนที่ โดยบล็อกสี่เหลี่ยมที่เป็นจุดประแทน บล็อกของภาพจากเฟรมอดีตซึ่งเป็นบล็อกที่เลือกมาเพื่อสร้างเฟรมประมาณขึ้น ส่วนบล็อกที่เป็น เส้นทึบแทนบล็อกปัจจุบัน จะเห็นได้ว่า บล็อกที่นำมาจากเฟรมอดีตมีความเป็นไปได้ที่มีพื้นที่ ซ้อนทับกัน แต่บล็อกในเฟรมที่ทำการชดเชยการเคลื่อนที่ไม่มีการซ้อนทับกัน



รูปที่ 2.4 ลักษณะการชดเชยการเคลื่อนที่

การชดเชยการเคลื่อนที่แบ่งได้เป็นสองแบบ ได้แก่ การชดเชยการเคลื่อนที่แบบไปข้างหน้า (forward motion compensation) และการชดเชยการเคลื่อนที่แบบไปข้างหลัง (backward motion compensation) การชดเชยการเคลื่อนที่แบบไปข้างหน้าอาศัยการเข้ารหัสโดยอ้างอิงจากเฟรม ก่อนหน้านั้นในทางลำดับเวลา ส่วนแบบไปข้างหลังนั้นตรงกันข้าม คือ ใช้เฟรมที่อยู่ถัดไปเป็น เฟรมอ้างอิงในการทำนาย [15] เทคนิคทั้งสองแบบนี้สามารถใช้ร่วมกันหรือแยกกันได้ ดังในกรณี ของภาพแบบ B ต้องใช้การชดเชยการเคลื่อนที่ทั้งสองแบบ ทั้งตัวเข้ารหัสและตัวถอดรหัสในระบบ วิดีทัศน์แบบบีบอัดมีความสามารถในการทำการชดเชยการเคลื่อนที่ แต่ตัวเข้ารหัสมักจะ ทำงานมากกว่าในการประมาณการเคลื่อนที่ สรุปได้ว่า

- การประมาณการเคลื่อนที่เป็นกระบวนการของการประมาณเวกเตอร์การเคลื่อนที่ในขณะที่ทำ การเข้ารหัส
- การชดเชยการเคลื่อนที่เป็นกระบวนการที่ใช้เวกเตอร์การเคลื่อนที่ในการสร้างภาพประมาณ กลับขึ้นมา

## 2.7 อัลกอริทึมการเข้าคู่บล็อก (Block-Matching Algorithm: BMA)

วิธีการนี้เป็นวิธีที่ได้รับความนิยมใช้กันมากในการหาค่าเวกเตอร์การเคลื่อนที่ของมาตรฐานต่าง ๆ ไม่ว่าจะเป็นมาตรฐาน MPEG-1, MPEG-2 และยังมีแนวโน้มใน MPEG-4 อีกด้วย หรือจะเป็นมาตรฐาน H.261 / 262 / 263 / 263+ / 263++ เพราะจุดเด่นที่ว่า วิธีการนี้มีความซับซ้อนในการคำนวณต่ำกว่าวิธีอื่น ๆ และยังให้คุณภาพในการประมาณการเคลื่อนที่ที่เป็นที่ยอมรับได้

### 2.7.1 สมมติฐานของอัลกอริทึมการเข้าคู่บล็อก

สมมติฐานของอัลกอริทึมการเข้าคู่บล็อก คือ จุดภาพทุกจุดในบล็อกเดียวกันเคลื่อนที่ไปในทิศทางเดียวกันด้วยขนาดเท่า ๆ กัน (มีเวกเตอร์การเคลื่อนที่เดียวกัน) ซึ่งสมมติฐานนี้ไม่เป็นจริงเมื่อบล็อกมีขนาดใหญ่เกินไป แต่อย่างไรก็ตาม ขนาดของบล็อกนี้ต้องใหญ่พอสมควรเพื่อให้แยกแยะจุดเด่นของแต่ละบล็อกได้ ขนาดของบล็อกทั่วไป คือ  $8 \times 8$  และ  $16 \times 16$  จุดภาพ

ปัจจัยอื่น ๆ ที่ทำให้อัลกอริทึมการเข้าคู่บล็อกใช้งานได้ดีขึ้น ได้แก่

- วัตถุที่อยู่ในภาพเคลื่อนที่ไปในแนวขนานกับระนาบของภาพ (Translation) ไม่มีการเปลี่ยนแปลงในรูปแบบอื่น เช่น การหมุนของวัตถุ (Rotation) การขยาย (Scaling) และการย่อ-ขยาย (Zoom)
- ความสว่างของภาพมีลักษณะเป็นรูปแบบเดียวกันทั้งในทางเวลา และทางตำแหน่ง นั่นคือ ระดับแสงภายในภาพคงที่ ไม่มีการเปลี่ยนแปลง
- ไม่มีวัตถุใหม่เกิดขึ้นในภาพ หรือมีการหายไป (uncover) ของภาพเบื้องหลัง

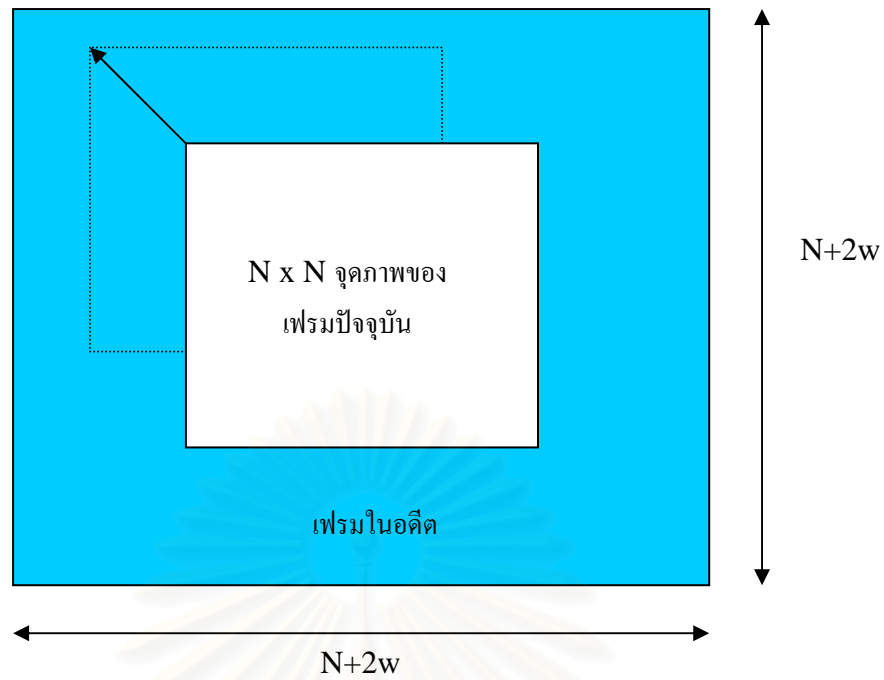
สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## 2.7.2 ขั้นตอนของอัลกอริทึมการเข้าคู่บล็อก

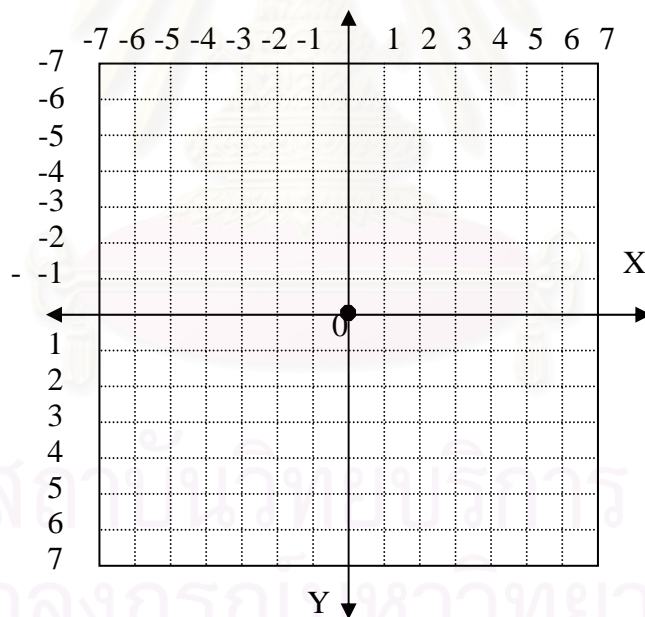
NxN													

รูปที่ 2.5 การแบ่งเฟรมของภาพออกเป็นบล็อกขนาด  $N \times N$

1. แบ่งภาพในเฟรมปัจจุบัน ออกเป็นบล็อกย่อย ๆ ที่ไม่มีการซ้อนทับกัน (non-overlapped block) ขนาดเท่า ๆ กัน ซึ่งเรียกว่า บล็อกปัจจุบัน (current block) ดังในรูปที่ 2.5
2. หาค่าเวกเตอร์การเคลื่อนที่สำหรับแต่ละบล็อก โดยอ้างอิงจากบล็อกอดีต (previous block) ดังตัวอย่างในรูปที่ 2.6 ตามกระบวนการดังต่อไปนี้
  - 2.1 กำหนดขอบเขตของการค้นหาสำหรับการเคลื่อนที่ของแต่ละบล็อกปัจจุบัน ซึ่งเป็นขอบเขตของขนาดเวกเตอร์การเคลื่อนที่สูงสุดที่เป็นได้ ขอบเขตนี้เรียกว่า หน้าต่างการค้นหา (search window) แทนด้วยตัวแปร  $w$  โดยทั่วไปนิยมใช้ที่  $\pm 7$  จุดภาพ ( $w = 7$ ) ดังในรูปที่ 2.7
  - 2.2 กำหนดขอบเขตพื้นที่การค้นหา (search area) บนเฟรมอดีต ซึ่งพื้นที่การค้นหามีขนาดเท่ากับ  $(N+2w) \times (N+2w)$  โดยที่บล็อกปัจจุบันมีขนาดเท่ากับ  $N \times N$
  - 2.3 เปรียบเทียบบล็อกปัจจุบันกับพื้นที่บนเฟรมอดีตที่ขนาดเท่ากับบล็อกปัจจุบัน ( $N \times N$ ) เลือกค่าเวกเตอร์การเคลื่อนที่ที่ให้ค่าความเพี้ยน (Distortion Measurement) ต่ำที่สุด เป็นเวกเตอร์การเคลื่อนที่ของบล็อกที่กำลังค้นหา



รูปที่ 2.6 ลักษณะการเข้าคู่บล็อก

รูปที่ 2.7 ลักษณะหน้าต่างการค้นหาค่า  $\pm 7$  จุดภาพ

กล่าวโดยนัยหนึ่ง คือ ในแต่ละบล็อกใช้วิธีการค้นหาบล็อกในพื้นที่การค้นหาที่มีความเข้ากันได้กับบล็อกปัจจุบันมากที่สุด เลือกค่าเวกเตอร์ที่ให้ค่าความผิดพลาดต่ำสุดเป็นเวกเตอร์การเคลื่อนที่ ซึ่งการวัดค่าความผิดพลาดมีหลายวิธีด้วยกัน รายละเอียดเป็นดังหัวข้อถัดไป

### 2.7.3 การวัดค่าความผิดพลาด

คือ การประเมินถึงค่าความแตกต่างหรือความเหมือนกันระหว่างบล็อกสองบล็อก วิธีการที่ใช้ในการตรวจสอบค่าความผิดพลาดมีหลายวิธีด้วยกัน แต่ละวิธีมีความซับซ้อนและประสิทธิภาพแตกต่างกันออกไป ตัวอย่างของวิธีที่ได้รับความนิยมใช้ เช่น

#### 2.7.3.1 ค่าความผิดพลาดกำลังสองเฉลี่ย (Mean Square Error: MSE)

วิธีการนี้ต้องใช้การคำนวณสูงแต่เป็นวิธีที่ให้ผลลัพธ์ที่ดี เป็นการวัดระยะทางแบบยูคลิดเดียน (Euclidean distance) ระหว่างจุดภาพของบล็อกสองบล็อก ใช้หนึ่งตัวดำเนินการคูณต่อหนึ่งค่าความแตกต่างของจุดภาพ มีสมการเป็นดังนี้

$$MSE(dx, dy) = \frac{1}{N \cdot N} \sum_{m=x}^{x+N-1} \sum_{n=y}^{y+N-1} [I_k(m, n) - I_{k-1}(m + dx, n + dy)]^2 \quad (2.3)$$

$$(MV_x, MV_y) = \min_{(dx, dy) \in R^2} MSE(dx, dy) \quad (2.4)$$

โดยที่  $N \times N$  เป็นขนาดของบล็อก

$MSE(dx, dy)$  เป็นค่าความผิดพลาดกำลังสองเฉลี่ย เมื่อเวกเตอร์การเคลื่อนที่เท่ากับ  $(dx, dy)$

$I_k(m, n)$  เป็นค่าความเข้มของแสงของเฟรมที่  $k$  ที่ตำแหน่ง  $(m, n)$

#### 2.7.3.2 ค่าความผิดพลาดสัมบูรณ์เฉลี่ย (Mean Absolute Error: MAE)

วิธีนี้อาจเรียกว่า ความแตกต่างสัมบูรณ์เฉลี่ย (Mean Absolute Difference: MAD) เป็นวิธีที่ได้รับความนิยมเป็นอย่างมาก [2] ในอัลกอริทึมการเข้าคู่บล็อกเนื่องจากความง่าย ให้ประสิทธิภาพที่ใกล้เคียงกับค่าผิดพลาดกำลังสองเฉลี่ย ดังสมการด้านล่างนี้

$$MAE(dx, dy) = \frac{1}{N \cdot N} \sum_{m=x}^{x+N-1} \sum_{n=y}^{y+N-1} |I_k(m, n) - I_{k-1}(m + dx, n + dy)| \quad (2.5)$$

### 2.7.3.3 ค่าผลรวมค่าสัมบูรณ์ความแตกต่าง (Sum of Absolute Difference: SAD)

เพื่อให้สามารถใช้งานได้ง่ายขึ้น วิธีนี้จะเลยพจน์ตัวหาร  $N \times N$  ของสมการที่ (2.5) ทิ้งไป ทำให้ได้สมการเป็น

$$SAD(dx, dy) = \sum_{m=x}^{x+N-1} \sum_{n=y}^{y+N-1} |I_k(m, n) - I_{k-1}(m+dx, n+dy)| \quad (2.6)$$

## 2.8 กระบวนการประมาณการเคลื่อนที่อย่างรวดเร็ว (Fast Motion Estimation Algorithms)

สำหรับการประยุกต์ใช้งานการสื่อสารภาพแบบเวลาจริงที่ใช้กำลังไฟฟ้าจากแบตเตอรี่ มีข้อจำกัดว่า อุปกรณ์ที่ใช้ต้องกินกำลังงานต่ำ ในขณะที่วิธีการค้นหาแบบทั้งหมดต้องใช้การคำนวณทางคณิตศาสตร์ 9.34 พันล้านครั้งต่อวินาที ใช้แบนด์วิดท์ 6.22 Gb/s สำหรับการเข้ารหัสวิดีโอที่สนับสนุนในรูปแบบ CIF ที่การแสดงผล 30 เฟรมในหนึ่งวินาที ซึ่งเห็นได้ชัดเจนว่า การค้นหาแบบทั้งหมดไม่เหมาะสมสำหรับข้อจำกัดนี้ ถึงแม้ให้ค่าความผิดพลาดต่ำสุดก็ตาม

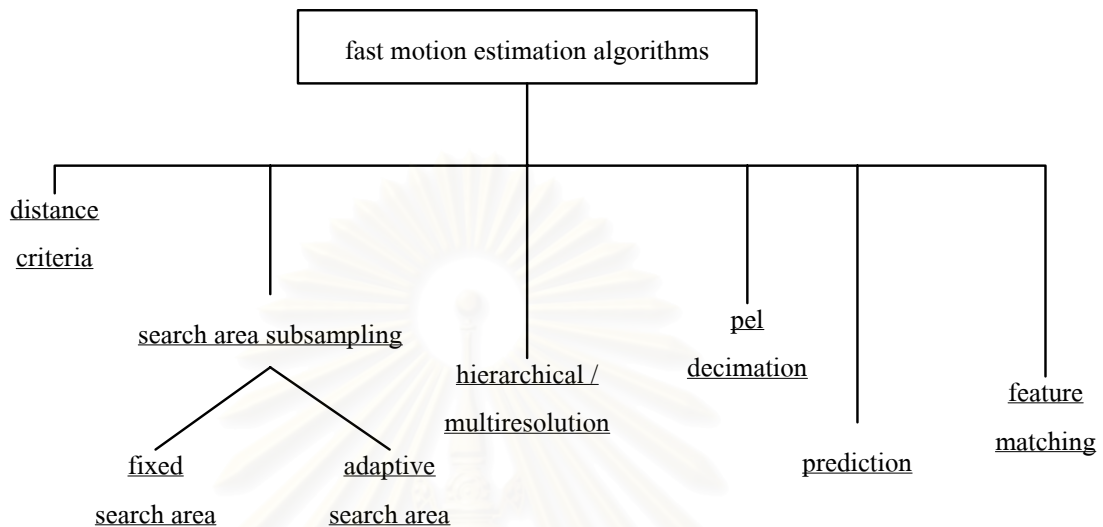
ดังนั้น กระบวนการประมาณการเคลื่อนที่อย่างรวดเร็วจึงเกิดขึ้นมาเพื่อลดความซับซ้อนในการคำนวณเป็นหลัก อย่างไรก็ตาม การลดความซับซ้อนเช่นนี้ทำให้ต้องเสียคุณภาพของสัญญาณตามมา การลดความซับซ้อนในการคำนวณในอีกแง่หนึ่ง ก็คือ การลดจำนวนจุดการค้นหา โดยที่ยังรักษาคุณภาพของการประมาณการเคลื่อนที่เอาไว้ให้อยู่ในระดับที่ยอมรับได้

### 2.8.1 สมมติฐานของการประมาณการเคลื่อนที่อย่างรวดเร็ว

เมื่อตั้งสมมติฐานว่า “บล็อกหนึ่ง ๆ มีจุดที่เหมาะสมที่สุดเพียงจุดเดียวในบริเวณที่กำหนดให้ ถ้าจุดค้นหาอยู่ห่างจากจุดต่ำสุดนี้มากเท่าใด ค่าความเพี้ยนต่อบล็อกก็จะมีค่าสูงขึ้นเท่านั้น” กระบวนการค้นหาอย่างรวดเร็วทุกแบบ ใช้หลักการหาจุดต่ำสุดดังกล่าว โดยการจำกัดจำนวนของจุดการค้นหาข้างเคียงที่ต้องทำการตรวจสอบหาค่าต่ำสุด ยกตัวอย่างเช่น ถ้าใช้ขอบเขตการค้นหา  $p = 6$  การใช้เทคนิคการค้นหาอย่างรวดเร็วสามารถลดจำนวนจุดการค้นหาจาก 169 จุด เหลือแค่ประมาณ 20 จุดเท่านั้น [15]

## 2.8.2 งานวิจัยด้านการเข้าคู่บล็อกอย่างรวดเร็ว

ภาพรวมของกระบวนการประมาณการเคลื่อนที่อย่างรวดเร็ว เป็นดังรูปที่ 2.8



รูปที่ 2.8 ภาพรวมของกระบวนการประมาณการเคลื่อนที่อย่างรวดเร็ว

การประมาณการเคลื่อนที่อย่างรวดเร็วที่มีอยู่ส่วนใหญ่ใช้กระบวนการเข้าคู่ โดยอาศัยวิธีการอย่างหนึ่งอย่างใดหรือหลายอย่าง ต่อไปนี้ [2]

- Distance criterion: ใช้สำหรับการคำนวณระยะห่างระหว่างบล็อกก่อนหน้ากับบล็อกที่ทำการค้นหาในรูปค่าความแตกต่างสัมบูรณ์เฉลี่ย เนื่องจากความง่าย ดังที่กล่าวในหัวข้อที่ 2.7.3
- Search strategy: เพื่อหาค่าต่ำสุดของค่าความผิดพลาดจากทุกตำแหน่งบล็อกที่เป็นไปได้โดยใช้ขั้นตอนต่ำสุดในการค้นหา
- Pel decimation: เป็นการสุ่มตัวอย่างของบล็อกที่ถูกนำมาหาค่าความผิดพลาด
- Block-matching: เป็นการจับคู่ภาพของบล็อกปัจจุบันกับบล็อกในพื้นที่การค้นหา
- Feature-matching: เป็นการจับคู่ข้อมูลภาพที่แบ่งแยกออกมาระหว่างบล็อกปัจจุบันกับขอบเขตการค้นหา
- Temporal prediction: เป็นการทำนายเวกเตอร์การเคลื่อนที่ หรือจุดศูนย์กลางการค้นหาโดยใช้ความสัมพันธ์ทางเวลา
- Spatial prediction: เป็นการทำนายเวกเตอร์การเคลื่อนที่ หรือจุดศูนย์กลางการค้นหาโดยใช้ความสัมพันธ์ทางปริภูมิ

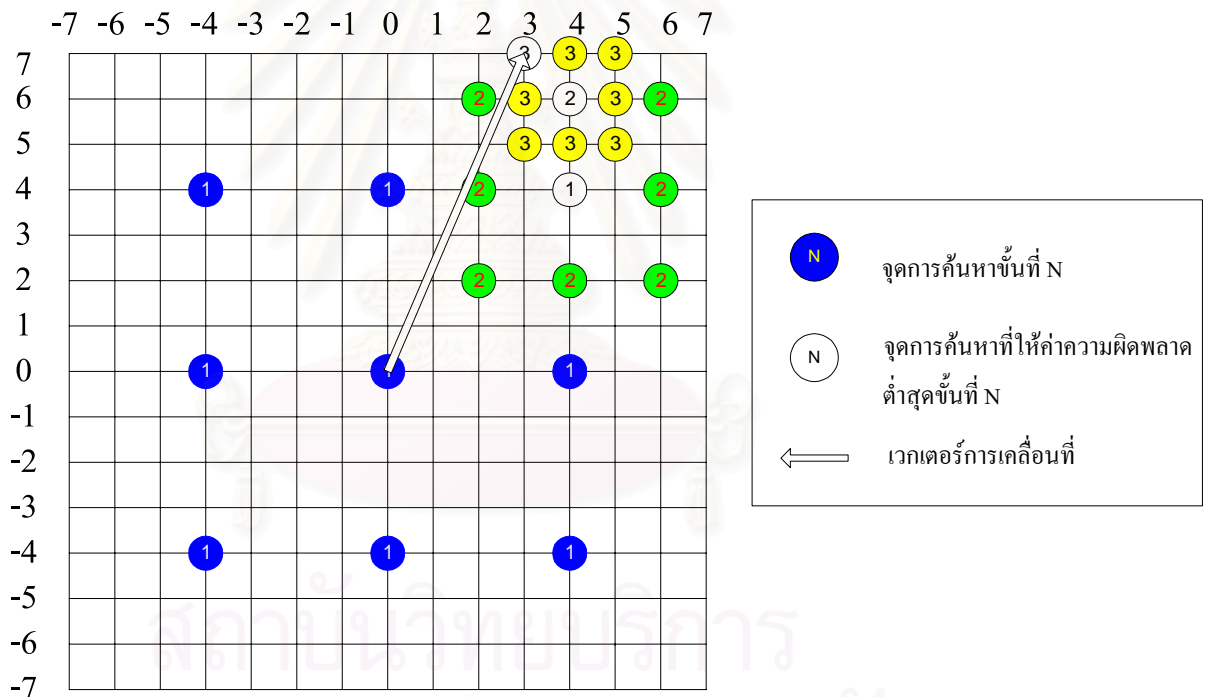


- Frequency-domain search: ทำการค้นหาเวกเตอร์การเคลื่อนที่หลังจากทำการแปลงบล็อกปัจจุบันและพื้นที่ค้นหาให้อยู่ในโดเมนความถี่
- Multi-resolution algorithm: เป็นกระบวนการประมาณการเคลื่อนที่ ที่ทำกับรูปภาพที่ความละเอียดแตกต่างกัน โดยผ่านกระบวนการอื่นมาก่อน เช่น ผ่านตัวกรอง หรือสุ่มตัวอย่าง

ตัวอย่างของงานวิจัยที่ผ่านมาที่เป็นที่นิยมในการประมาณการเคลื่อนที่ ได้แก่

2.8.2.1 การค้นหาแบบสามขั้นตอน (TSS) [5]

เป็นอัลกอริทึมที่ให้ความเร็วในการค้นหา โดยใช้จำนวนขั้นตอนในการค้นหาและจำนวนจุดการค้นหากที่ คือ 25 จุด ดังในรูปที่ 2.9



รูปที่ 2.9 ลักษณะวิธีการค้นหาแบบสามขั้นตอน

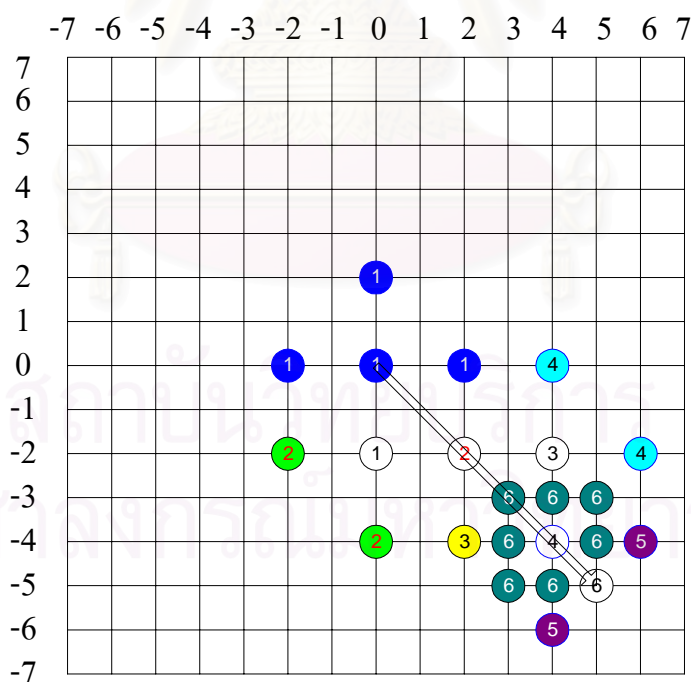
การค้นหาแบบสามขั้นตอน ใช้เงื่อนไขของ MAD แทนที่จะเป็น MSE ในแต่ละขั้นตอน เพิ่มความละเอียดของการค้นหาขึ้นเรื่อย ๆ หากหน้าต่างการค้นหามีขนาดมากกว่า  $\pm 7$  จุดภาพ ต้องเพิ่มจำนวนขั้นตอนในการค้นหาขึ้น โดยระยะห่างระหว่างจุดการค้นหาในแต่ละขั้นตอน คือ  $2^{L-n}$  เมื่อ n คือ ลำดับขั้นตอนระหว่างที่ทำการค้นหา L คือ จำนวนขั้นตอนที่ใช้ในการค้นหา มีค่าเท่ากับ

$\log_2(d+1)$  และจำนวนจุดการค้นหารวมทั้งหมด คือ  $1+8 \log_2(d+1)$  เมื่อ  $d$  คือขนาดของหน้าต่างการค้นหา

เมื่อหน้าต่างการค้นหาที่มีขนาด  $\pm 7$  จุดภาพ อัลกอริทึมนี้มี อัตราความเร็วที่เพิ่มขึ้น (speed-up ratio) เทียบกับการค้นหาแบบค้นหาทั้งหมด เท่ากับ  $225/25 = 9.0$  ให้ความแม่นยำของการค้นหาอยู่ในระดับปานกลาง ได้รับความนิยมในการใช้เป็นฐานในการเปรียบเทียบกับอัลกอริทึมอื่น ๆ เช่นเดียวกับ อัลกอริทึมการค้นหาแบบทั้งหมด

### 2.8.2.2 การค้นหาแบบลอการิทึม (LOGS) [4]

ใช้รูปแบบการค้นหาเป็นรูปเครื่องหมายบวก (+) ในแต่ละขั้นตอนการค้นหา โดยขั้นตอนแรกใช้ระยะการค้นหา (step size) เท่ากับ  $\lceil d/4 \rceil$  ( $\lceil x \rceil$  คือ จำนวนเต็มที่น้อยที่สุดที่มากกว่า  $x$ ) การค้นหาค่าเนินไปเรื่อย ๆ จนกว่าจะได้จุดต่ำสุดอยู่ที่จุดศูนย์กลางของรูป + จึงลดระยะการค้นหาลงครึ่งหนึ่ง จนกระทั่งขนาดช่วงก้าว (step size) ลดเหลือ 1 จึงทำการค้นหาอีก 8 จุดรอบ ๆ จุดต่ำสุดที่ได้ในขณะนั้น เลือกจุดที่ให้ค่าต่ำสุดเป็นเวกเตอร์การเคลื่อนที่ จำนวนจุดการค้นหาลดต่ำสุดที่เป็นไปได้ คือ 13 จุด สำหรับ  $d=7$  และจำนวนจุดการค้นหามากที่สุด คือ 23 จุด ดังรูปที่ 2.10



รูปที่ 2.10 ลักษณะวิธีการค้นหาแบบลอการิทึม

### 2.8.2.3 การค้นหาแบบสามขั้นตอนวิธีใหม่ (NTSS) [6]

เป็นวิธีที่ปรับปรุงจากกระบวนการค้นหาแบบสามขั้นตอน เพื่อแก้ไขปัญหาของ TSS ซึ่งไม่เหมาะสมกับภาพที่มีการเคลื่อนที่น้อย ประกอบกับผลลัพธ์ที่ได้จากการสังเกตว่า ในลำดับภาพตามความเป็นจริงมีการเคลื่อนที่แบบช้าและค่อนข้างสม่ำเสมอ [2] ส่งผลให้เกิดการกระจายตัวของเวกเตอร์การเคลื่อนที่แบบเน้นที่จุดศูนย์กลาง (center-biased) แทนที่เป็นการกระจายตัวอย่างสม่ำเสมอ

ข้อดีของวิธีการค้นหาแบบนี้ คือ

- ใช้เทคนิคการหยุดครึ่งทาง (half way stop technique) เพื่อลดจำนวนครั้งในการคำนวณ
- ไม่ต้องใช้ค่าจุดเปลี่ยน (threshold) ในการกำหนดการเปลี่ยนแปลง เหมือนวิธีหน้าต่างการค้นหาแบบปรับตัวได้ (Adaptive Search Window) ซึ่งมีข้อเสียคือ ค่าจุดเปลี่ยนที่เหมาะสมกับภาพต่าง ๆ กันมีค่าแตกต่างกันไป ทำให้ยากต่อการใช้งานจริง
- ให้ค่าความผิดพลาดน้อยกว่า TSS

ในขณะที่มีข้อเสีย คือ มีขั้นตอนในการค้นหาช้ากว่า TSS กรณีที่ลำดับภาพเป็นแบบเคลื่อนที่เร็ว

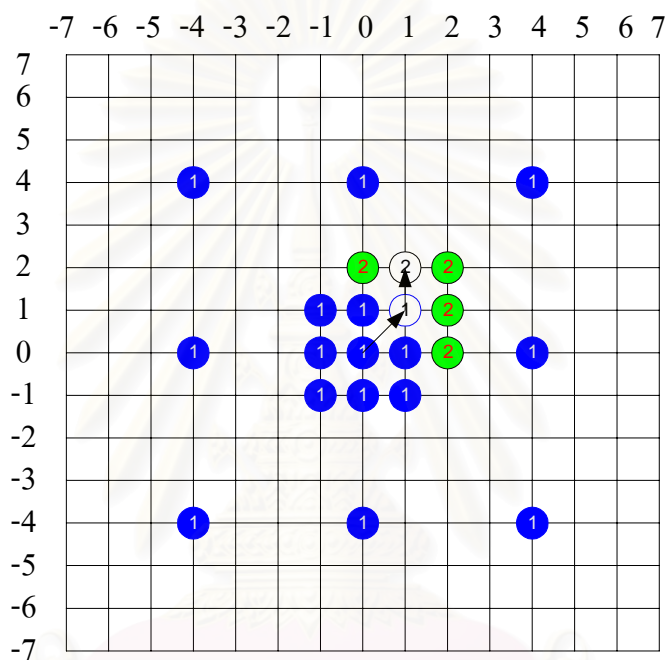
#### หลักการ

พื้นผิวความผิดพลาดมีการเปลี่ยนแปลงไปในทางเดียวกัน (monotonic) ในพื้นที่เล็ก ๆ บริเวณรอบ ๆ จุดต่ำสุดรวม (Global Minimum) ซึ่งแตกต่างจาก TSS ที่มีสมมติฐานว่า พื้นผิวความผิดพลาดเป็นแบบเปลี่ยนแปลงไปในทางเดียวกันทั่วทั้งหน้าต่างการค้นหา ดังนั้น NTSS เหมาะกับภาพที่มีการเคลื่อนที่น้อย โดยเฉพาะภาพการประชุมผ่านวิดีโอ (video conference)

#### วิธีการ

1. ขั้นตอนแรกเหมือนกับ TSS แต่เพิ่มจุดการค้นหาอีก 8 จุด รอบจุด (0,0) ดังรูปที่ 2.11
2. ถ้าจุดต่ำสุดอยู่ที่จุด (0,0) ให้หยุดการค้นหา แล้วให้เวกเตอร์การเคลื่อนที่ที่เป็น (0,0) ได้จำนวนจุดการค้นหาทั้งหมด คือ 17 จุด

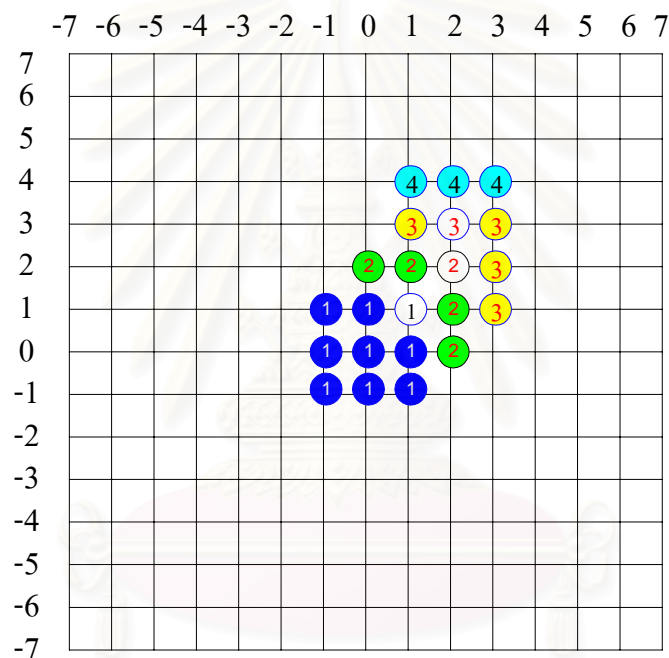
3. ถ้าจุดต่ำสุดอยู่รอบจุด (0,0) ให้ตรวจสอบรอบจุดดังกล่าว (เลือกเฉพาะจุดที่ยังไม่ได้ทำการค้นหา จาก 8 จุด) จุดต่ำสุดที่ได้ คือ ค่าเวกเตอร์การเคลื่อนที่ ในกรณีจุดศูนย์กลางการค้นหาไม่ใช่ จุดตำแหน่งมุม จำนวนจุดการค้นหาทั้งหมดคือ 20 จุด (17+3) สำหรับกรณีจุดศูนย์กลางการค้นหาเป็นจุดมุมได้จำนวนจุดการค้นหาเป็น 22 จุด (17+5)
4. ถ้าจุดต่ำสุดอยู่ที่ตำแหน่ง 8 จุดที่เหลือ (บริเวณรอบนอก) จึงทำการค้นหาแบบ TSS จนเสร็จสิ้น ซึ่งใช้จำนวนจุดการค้นหาทั้งหมด 33 จุด



รูปที่ 2.11 ลักษณะวิธีการค้นหาแบบสามขั้นตอนวิธีใหม่

### 2.8.2.4 การค้นหาแบบลดระดับเกรเดียนต์เชิงบล็อก (BBGDS) [16]

เป็นวิธีการที่อ้างอิงจากสมมติฐานของการเน้นจุดศูนย์กลาง มีลักษณะเป็นดังรูปที่ 2.12 มีรูปแบบการค้นหาเป็นรูปสี่เหลี่ยมจัตุรัสขนาด  $3 \times 3$  จุดภาพ ซึ่งเคลื่อนที่ไปในทิศทางที่มีค่า SAD ต่ำที่สุด โดยขั้นแรก ทำการค้นหาที่จุด  $(0,0)$  และจุดรอบ ๆ อีก 8 จุด ขั้นต่อไปก็ค้นหาอีกแปดจุด รอบจุดต่ำสุดที่ได้ในขั้นตอนก่อนหน้า ยกเว้นจุดที่ทำการค้นหาไปแล้ว ดังนั้น จำนวนจุดการค้นหาที่เพิ่มขึ้นในแต่ละขั้นตอน คือ 3 จุดในกรณีจุดต่ำสุดของขั้นก่อนหน้าอยู่บริเวณขอบ และ 5 จุดในกรณีที่อยู่บริเวณมุม การค้นหาคำเนินต่อไปจนกว่าได้จุดต่ำสุดอยู่ตรงกลาง จึงหยุดการค้นหา และให้จุดนั้นเป็น เวกเตอร์การเคลื่อนที่



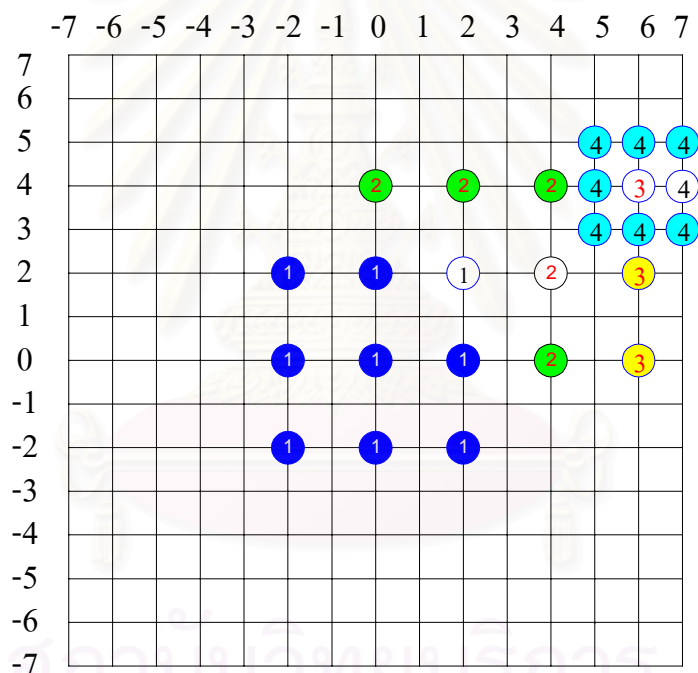
รูปที่ 2.12 ลักษณะวิธีการค้นหาแบบ BBGDS

เห็นได้ชัดเจนว่า อัลกอริทึมนี้เหมาะสำหรับลำดับภาพที่มีการเคลื่อนที่ไม่มาก เนื่องจากถ้า เวกเตอร์การเคลื่อนที่มีขนาดมาก ต้องใช้จำนวนจุดการค้นหามาก ในขณะที่เดียวกันหากเวกเตอร์การเคลื่อนที่อยู่บริเวณใกล้กับจุด  $(0,0)$  ใช้จำนวนจุดการค้นหาอย่างมาก ซึ่งเรียกว่าเป็น อัลกอริทึมแบบ Center-biased

ข้อเสียของวิธีนี้ คือ การค้นหาถูกดักให้ตกอยู่ใน จุดต่ำสุดท้องถิ่น (Local Minimum) ได้ง่าย

### 2.8.2.5 การค้นหาแบบสี่ขั้นตอน (FSS) [3]

วิธีการค้นหาแบบสี่ขั้นตอน ถูกนำเสนอขึ้นเพื่อปรับปรุงประสิทธิภาพของ TSS โดยเฉพาะอย่างยิ่ง ในกรณีที่เวกเตอร์การเคลื่อนที่มีค่าน้อย ๆ ดังที่ได้กล่าวมาแล้ว ในขั้นตอนแรก FSS ตรวจสอบจุดการค้นหา 9 จุดในหน้าต่างขนาด  $5 \times 5$  อัลกอริทึมนี้มีลักษณะการค้นหาคล้ายกับ BBGDS เพียงแต่เพิ่มระยะห่างระหว่างจุดการค้นหาเป็น 2 สำหรับ  $d = 7$  และ  $\lceil d/4 \rceil$  สำหรับค่า  $d$  อื่น ๆ หน้าต่างขนาด  $5 \times 5$  นี้ถูกขยับไปในทิศทางที่เข้าคู่กันได้ดีที่สุดถ้าหากการค้นหาสามครั้งแรก ไม่ได้มีจุดต่ำสุดอยู่ตรงกลาง และหากถึงครั้งที่สี่หรือได้จุดต่ำสุดอยู่บริเวณตรงกลางให้ตรวจสอบจุดการค้นหา 8 จุดรอบ ๆ จุดต่ำสุดในขณะนั้น เพื่อเลือกเวกเตอร์การเคลื่อนที่ จำนวนจุดการค้นหาที่น้อยที่สุด คือ 17 จุด และมากที่สุด คือ 27 จุด

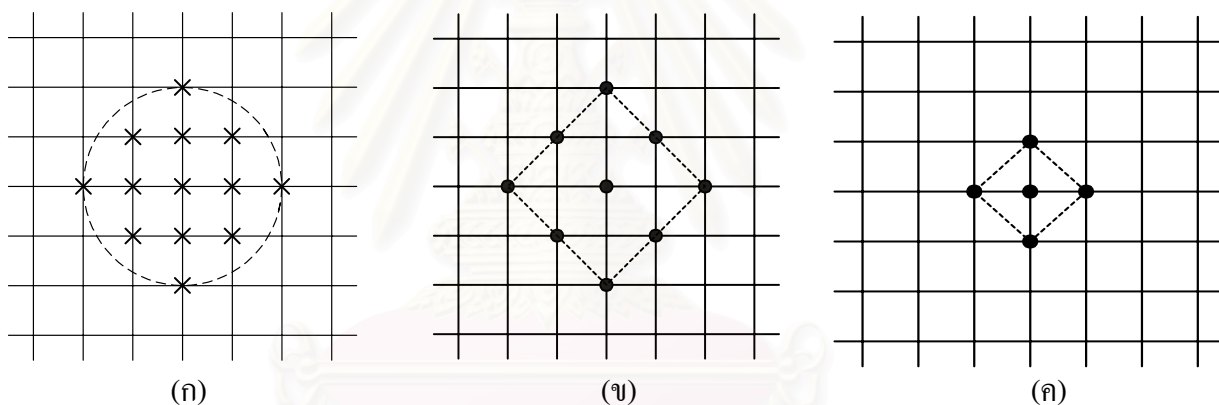


รูปที่ 2.13 ลักษณะการค้นหาแบบสี่ขั้นตอน

### 2.8.2.6 การค้นหารูปเพชร (Diamond Search Algorithm) [7]

วิธีการค้นหาอย่างรวดเร็วแบบต่าง ๆ นั้นมีข้อเสียแตกต่างกันไป เช่น การค้นหาแบบสามขั้นตอนใช้รูปแบบการค้นหาที่กว้าง มีขนาด  $9 \times 9$  จุดภาพ และมีจุดการค้นหาในตอนแรกไม่แน่นอน ทำให้มีความเป็นไปได้มากที่จะค้นหาในทิศทางที่ผิดพลาดตั้งแต่แรก สำหรับการค้นหาแบบสี่ขั้นตอนใช้รูปแบบการค้นหาขนาด  $5 \times 5$  จุดภาพและใช้ประโยชน์จากจุดการค้นหาที่ซ้ำกับขั้นตอนแรกเพื่อลดความซับซ้อนในการคำนวณ แต่สำหรับบล็อกที่ไม่มีการเคลื่อนที่ วิธีการค้นหาแบบสี่ขั้นตอนต้องใช้จุดการค้นหาถึง 17 จุด เป็นต้น

จากผลการสังเกตพบว่า ตั้งแต่ร้อยละ 53 ถึงร้อยละ 99 ของเวกเตอร์การเคลื่อนที่ตกอยู่ในพื้นที่วงกลมรัศมีสองจุดภาพรอบ ๆ ค่าเวกเตอร์การเคลื่อนที่ศูนย์ จึงเกิดการพัฒนาอัลกอริทึมการค้นหารูปเพชรเพื่อให้รูปแบบการค้นหาสอดคล้องกับการกระจายตัว ซึ่งอัลกอริทึมแบบนี้อาศัยรูปแบบในการค้นหาสองรูปแบบ ดังรูปที่ 2.14

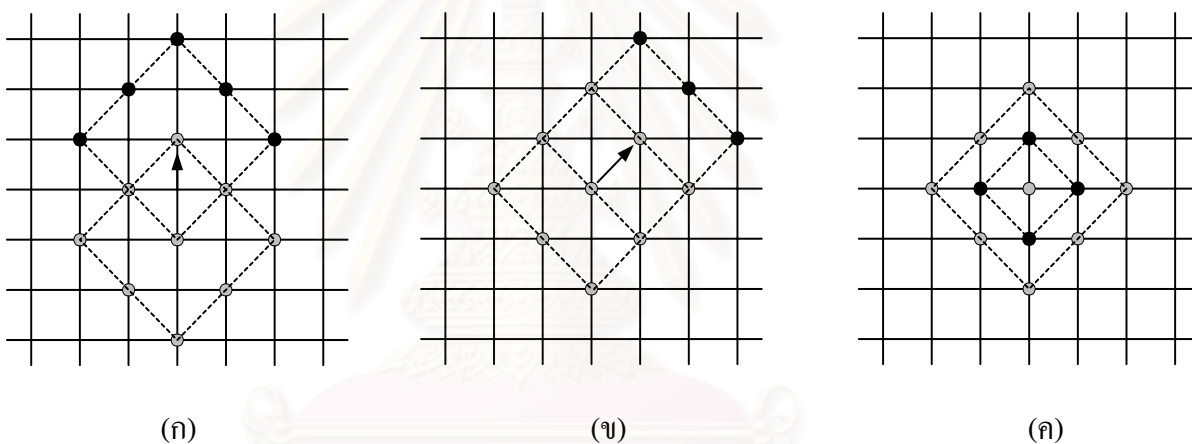


รูปที่ 2.14 รูปแบบการค้นหารูปเพชร (ก) จุดการค้นหาทั้ง 13 จุดอยู่ในวงกลมรัศมี 2 จุดภาพ (ข) รูปแบบการค้นหารูปเพชรใหญ่ จำนวนเก้าจุด (ค) รูปแบบการค้นหารูปเพชรเล็ก มีจำนวนห้าจุดการค้นหา

รูปแบบแรกคือ รูปแบบการค้นหารูปเพชรใหญ่ (Large Diamond Search Pattern: LDSP) ใช้สำหรับการค้นหาตำแหน่งจุดต่ำสุดแบบหยาบ ๆ ใช้จุดการค้นหาเก้าจุด (รูปที่ 2.14ข) เมื่อได้จุดต่ำสุดอยู่ที่ จุดศูนย์กลางของ LDSP จึงใช้รูปแบบการค้นหารูปเพชรเล็ก (Small Diamond Search Pattern: SDSP) เพื่อค้นหาเวกเตอร์การเคลื่อนที่แบบละเอียด

### อัลกอริทึมการค้นหาหารูปเพชร มีวิธีการดังนี้

- ขั้นตอนที่ 1. ใช้การค้นหาหารูปเพชรใหญ่ที่จุดศูนย์กลางของหน้าต่างการค้นหา คำนวณค่าความผิดพลาดทั้งเก้าจุด ถ้าจุดที่ให้ค่าผิดพลาดต่ำสุดอยู่ที่จุดศูนย์กลางไปที่ขั้นตอนที่ 3 นอกจากนี้ไปที่ขั้นตอนที่ 2
- ขั้นตอนที่ 2. ให้จุดต่ำสุดจากขั้นตอนที่ 1 เป็นจุดศูนย์กลางการค้นหาใหม่ โดยถ้าจุดต่ำสุดอยู่บริเวณของมุม ให้ค้นหาเพิ่มอีกห้าจุด ดังในรูปที่ 2.15(ก) และถ้าอยู่บริเวณของด้านให้ค้นหาเพิ่มอีกสามจุด ดังรูปที่ 2.15(ข)
- ขั้นตอนที่ 3. เปลี่ยนจากการค้นหาหารูปเพชรใหญ่ เป็นการค้นหาหารูปเพชรเล็ก เลือกจุดที่ให้ค่าผิดพลาดต่ำสุดเป็นเวกเตอร์การเคลื่อนที่ ดังรูปที่ 2.15(ค)



รูปที่ 2.15 การย้ายจุดการค้นหาของอัลกอริทึมการค้นหาหารูปเพชร (ก) กรณีจุดต่ำสุดอยู่บริเวณของมุม (ข) กรณีจุดต่ำสุดอยู่บริเวณของด้าน (ค) การปรับเปลี่ยนเป็นรูปแบบการค้นหาหารูปเพชรเล็ก

อัลกอริทึมการค้นหาหารูปเพชรนี้มีการเลื่อนตัว ทำให้ไม่ถูกจำกัดด้วยเงื่อนไขการหยุดเหมือนกับอัลกอริทึมอื่น ๆ และได้รับการทดสอบกับมาตรฐาน MPEG-4 แล้ว [2] ทั้งในด้านของจำนวนจุด การค้นหาเฉลี่ย และค่าความผิดพลาดกำลังสองเฉลี่ย วิธีการค้นหาหารูปเพชรให้ผลที่ดีกว่าเมื่อเปรียบเทียบกับวิธี TSS FSS และ NTSS นอกจากนี้สำหรับบล็อกที่ไม่เคลื่อนที่ การค้นหาแบบเพชรให้จำนวนจุดการค้นหาเพียง 13 จุดเท่านั้น



## 2.9 การวัดสมรรถนะของระเบียบวิธี (Performance Measurement)

การวัดสมรรถนะของระเบียบวิธีมีอยู่สองลักษณะ คือ ค่าคุณภาพของภาพ และความซับซ้อนในการคำนวณ ดังมีรายละเอียด ดังต่อไปนี้

### 2.9.1 การวัดคุณภาพของภาพ (Quality Measurement)

ในการวัดคุณภาพของภาพที่สร้างขึ้นจากกระบวนการชดเชยการเคลื่อนที่ ด้วยวิธีการค้นหาแบบต่าง ๆ นั้น เป็นการตรวจสอบว่า ภาพที่ได้นั้นมีความใกล้เคียงกับภาพที่ต้องการประมาณมากน้อยเพียงใด ค่าที่ใช้วัดคุณภาพนิยมใช้กันอยู่สองชนิด คือ ค่าความผิดพลาดกำลังเฉลี่ย และค่าอัตราส่วนสัญญาณยอดต่อสัญญาณรบกวน

#### 2.9.1.1 ค่าความผิดพลาดกำลังสองเฉลี่ย (Mean Square Error: MSE)

เป็นการวัดคุณภาพในแบบ Objective ซึ่งคำนวณจากสมการ

$$MSE = \frac{1}{M \times N} \sum_{m=1}^M \sum_{n=1}^N (X(m,n) - X^R(m,n))^2 \quad (2.7)$$

โดยที่  $M \times N$  เป็นขนาดของเฟรม

$X(m,n)$  เป็นค่าความเข้มแสงของภาพปัจจุบันที่จุด  $(m,n)$

$X^R(m,n)$  เป็นค่าความเข้มแสงของภาพอ้างอิงที่จุด  $(m,n)$

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

### 2.9.1.2 ค่าอัตราส่วนสัญญาณยอดต่อสัญญาณรบกวน (Peak Signal to Noise Ratio: PSNR)

เป็นค่าที่วัดคุณภาพในแบบ objective ที่พยายามให้ความหมายเชิง subjective [9] โดยมีสมการดังต่อไปนี้

$$PSNR = 10 \log \left[ \frac{R^2}{\frac{1}{M \times N} \sum_{m=1}^M \sum_{n=1}^N (X(m,n) - X^R(m,n))^2} \right] \quad (2.8)$$

โดยที่  $R=2^L$  เมื่อ L แทนจำนวนบิต และ R แทนค่ายอดสูงสุดที่เป็นไปได้ของสัญญาณ  
 $M \times N$  เป็นขนาดของเฟรม  
 $X(m,n)$  เป็นค่าความเข้มแสงของภาพปัจจุบันที่จุด (m,n)  
 $X^R(m,n)$  เป็นค่าความเข้มแสงของภาพอ้างอิงที่จุด (m,n)

### 2.9.2 การวัดความซับซ้อนในการคำนวณของระเบียบวิธี (Computational Complexity Measurement)

การวัดความซับซ้อนของระเบียบวิธีสามารถทำได้หลายแบบ เช่น เวลาในการคำนวณ หรือ จำนวนตัวปฏิบัติการในการคำนวณ สำหรับระเบียบวิธีการเข้าคู่บล็อกนิยมใช้การเปรียบเทียบจากจำนวนจุดการค้นหาที่ใช้ เพราะว่าแต่ละจุดการค้นหาใช้การคำนวณเท่ากันทุกครั้ง และในแต่ละครั้งยังต้องใช้ตัวปฏิบัติการในการคำนวณสูง ดังนั้น จำนวนจุดการค้นหาจึงมีความหมายสำคัญอย่างมากต่อความซับซ้อนในการคำนวณ ส่งผลให้ค่าที่ใช้วัดผลความซับซ้อนจึงเกี่ยวข้องกับจำนวนจุดการค้นหา ได้แก่ จำนวนจุดการค้นหาเฉลี่ยในลำดับภาพแต่ละชุด จำนวนจุดการค้นหาน้อยที่สุดที่เป็นไปได้ หรือจำนวนจุดการค้นหามากที่สุดที่เป็นไปได้

วิธีการค้นหาแบบทั้งหมดทำการค้นหาทุกค่าของเวกเตอร์ที่เป็นไปได้ในขอบเขตการค้นหา โดยใช้เงื่อนไขของค่า SAD ดังในสมการที่ (2.6) ที่ต่ำสุด ซึ่งต้องใช้การคำนวณสูงมาก โดยที่ SAD ต้องใช้ตัวดำเนินการพื้นฐาน 3 ชนิดด้วยกัน ได้แก่ การลบ การบวก และการหาค่าสัมบูรณ์ ทำให้ได้จำนวนการปฏิบัติการในหนึ่งวินาที (เมื่อคิดผลของค่า SAD เท่านั้น) ของการค้นหาแบบทั้งหมดเป็นดังสมการด้านล่างนี้ [8]

$$Op=3 \cdot 2p \cdot 2p \cdot N_h \cdot N_v \cdot f \quad (2.9)$$

โดยที่ขนาดของภาพในแนวแกนนอนและแกนตั้ง แทนด้วย  $N_h$  และ  $N_v$  ตามลำดับ  $f$  แสดงจำนวนเฟรมในหนึ่งวินาที และ  $p$  แทนขอบเขตการค้นหา  $Op$  แทนจำนวนครั้งในการคำนวณ สำหรับการประยุกต์ใช้งานวิดีโอ ในเวลาจริงทั่วไป ใช้  $p = 16$ ,  $f = 30$ ,  $N_h = 352$ ,  $N_v = 288$  (CIF) พบว่า ใช้การคำนวณถึง 9.34 พันล้านครั้งทีเดียว

สำหรับในงานวิทยานิพนธ์นี้ ใช้จำนวนจุดการค้นหาเป็นตัวบ่งบอกความซับซ้อนของระเบียบวิธีเพราะเทคนิคที่ใช้เปรียบเทียบใช้จำนวนปฏิบัติการเท่ากันทุกจุดการค้นหาแตกต่างกันที่เพียงรูปแบบการค้นหา



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

### บทที่ 3

#### การค้นหารูปเพชรชนิดไม่สมมาตรแบบปรับตัวได้โดยใช้การปรับย้ายจุดศูนย์กลาง Adaptive Asymmetric Diamond Search Algorithm Using Adaptive Search Center

ดังที่ได้กล่าวมาในบทที่แล้วว่า เทคนิคในการประมาณการเคลื่อนที่อย่างรวดเร็วได้รับการพัฒนาขึ้นอย่างมากและรวดเร็ว เพื่อให้สามารถนำมาประยุกต์ใช้งานได้จริง โดยเฉพาะอย่างยิ่ง เพื่อให้สอดคล้องกับการใช้งานในเวลาจริง เทคนิคการค้นหารูปเพชรชนิดไม่สมมาตรแบบปรับตัวได้โดยใช้การปรับย้ายจุดศูนย์กลางที่นำเสนอในวิทยานิพนธ์นี้ เป็นเทคนิคที่มุ่งเน้นการลดความซับซ้อนในการประมาณการเคลื่อนที่ โดยที่ยังคงรักษาความถูกต้องของสัญญาณวิถีที่ค้นได้ให้อยู่ในระดับที่ยอมรับได้ เมื่อเปรียบเทียบกับอัลกอริทึมการค้นหาลักษณะอื่น เทคนิคที่นำเสนอนี้เป็นการรวม 2 แนวความคิดเข้าด้วยกัน ได้แก่ เทคนิคการปรับรูปแบบการค้นหาโดยอาศัยการคาดคะเนขนาดของเวกเตอร์การเคลื่อนที่ และเทคนิคการปรับย้ายจุดศูนย์กลางในการเริ่มต้นค้นหา

เนื้อหาในบทนี้นำเสนอ สหสัมพันธ์กันระหว่างเวกเตอร์การเคลื่อนที่ของแต่ละบล็อก วิธีการทำนายเวกเตอร์การเคลื่อนที่ และ หลักการรวมทั้งขั้นตอนของอัลกอริทึมการค้นหารูปเพชรชนิดไม่สมมาตรแบบปรับตัวได้โดยใช้การปรับย้ายจุดศูนย์กลางที่นำเสนอ

#### 3.1 สหสัมพันธ์กันระหว่างบล็อก (Interblock Motion Correlation)

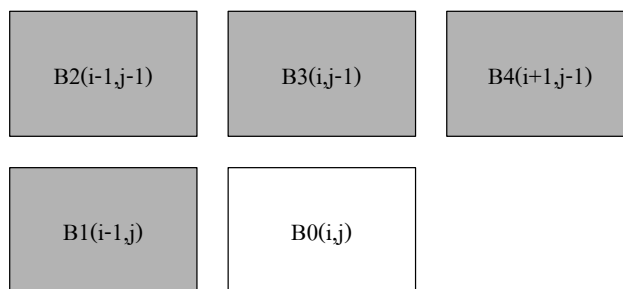
เนื่องจากวัตถุที่มีการเคลื่อนที่มักจะครอบคลุมหลาย ๆ บล็อกภายในหนึ่งเฟรมทำให้การเคลื่อนที่ของบล็อกที่ติดกันในเชิงพื้นที่มีความคล้ายคลึงกันมาก นอกจากนี้ เนื่องจากความต่อเนื่อง (continuity) ของการเคลื่อนที่ในเชิงเวลาทำให้การเคลื่อนที่ของบล็อกที่อยู่ใกล้เคียงกันในทางเวลา มีความสัมพันธ์กันเป็นอย่างมาก [17]

กำหนด สหสัมพันธ์ของการเคลื่อนที่ระหว่างบล็อก (interblock motion correlation) ด้วยระยะห่างระหว่างเวกเตอร์การเคลื่อนที่ของบล็อกปัจจุบันกับค่าเฉลี่ยของเวกเตอร์การเคลื่อนที่ของบล็อกข้างเคียงทั้งสี่บล็อก [17] ดังแสดงในรูปที่ 3.1 ดังนี้

$$\bar{D}_{mv} = \bar{V}_{B0} - \frac{1}{4} \sum_{i=1}^4 \bar{V}_{Bi} \quad (3.1)$$

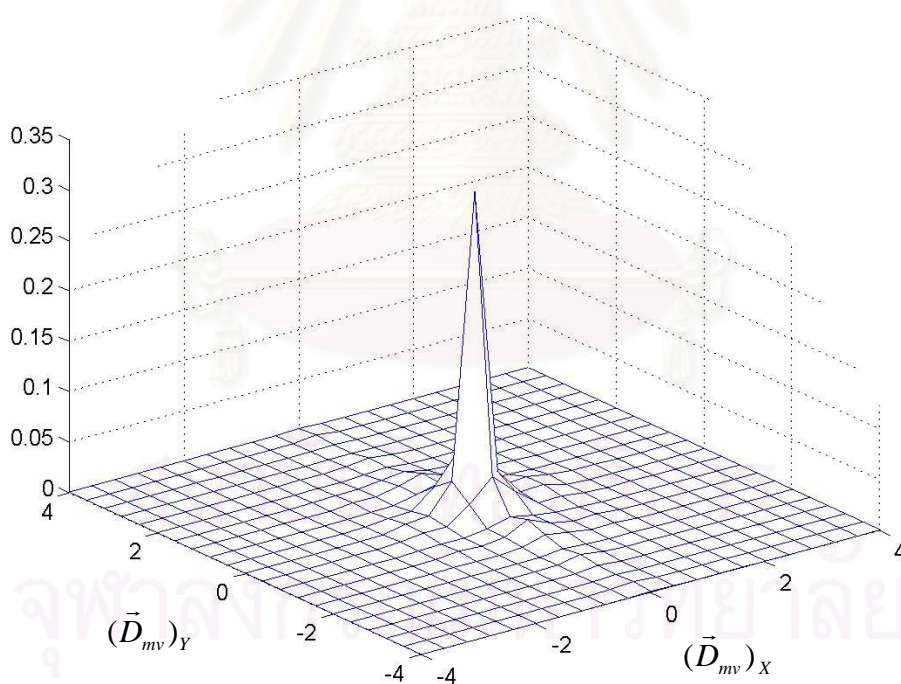
เมื่อ  $\bar{D}_{mv}$  คือ ระยะขจัด

$\bar{V}_{Bi}$ ,  $i = 0, 1, 2, 3$  และ 4 คือ ค่าเวกเตอร์การเคลื่อนที่ของบล็อกต่าง ๆ ในรูปที่ 3.1

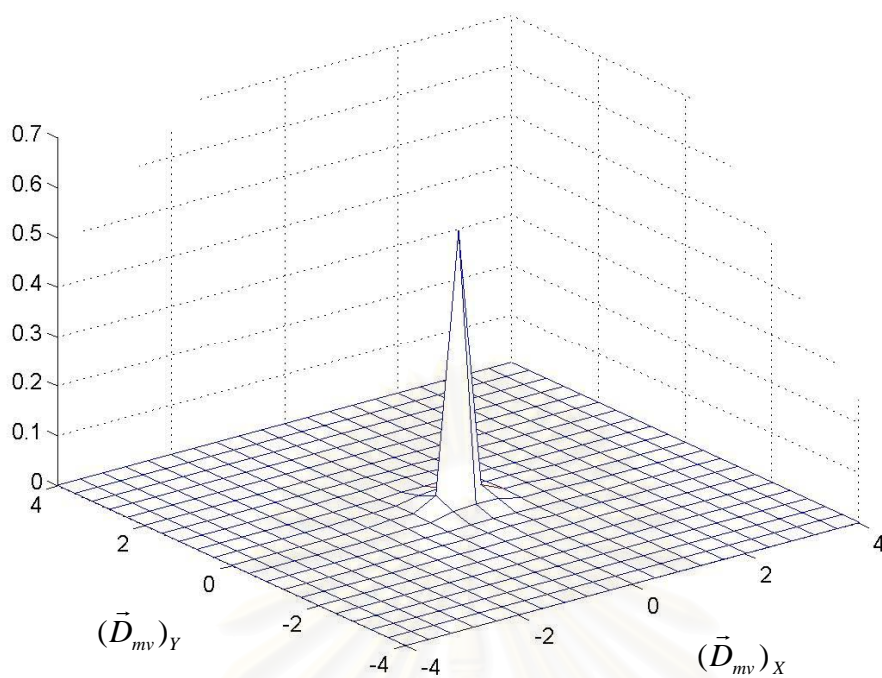


รูปที่ 3.1 รูปแบบของบล็อกข้างเคียงที่ใช้ในการวัดสหสัมพันธ์กันของเวกเตอร์การเคลื่อนที่

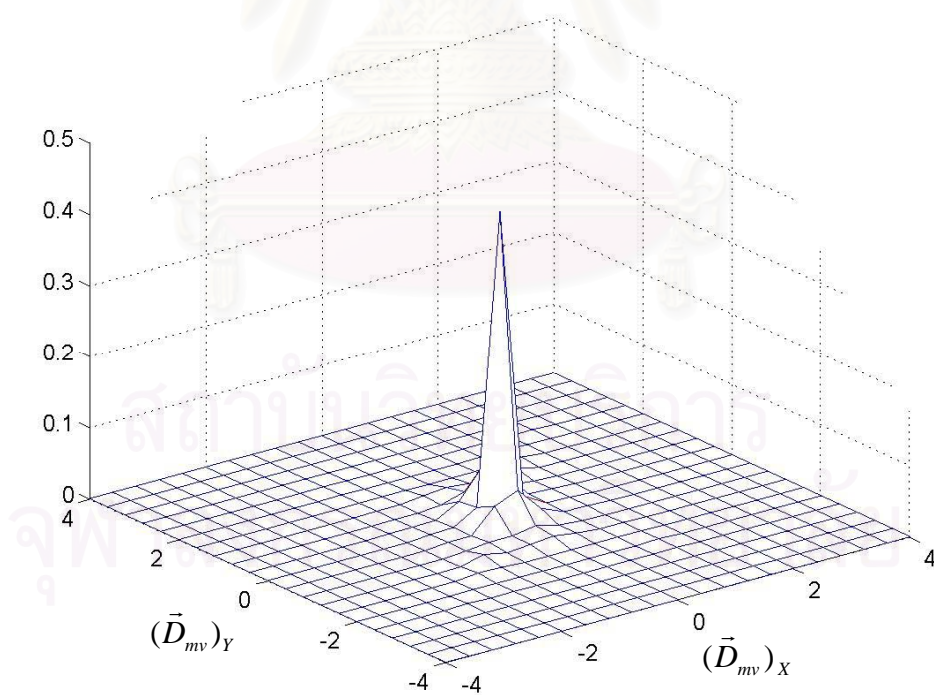
กรณีที่  $\bar{D}_{mv}$  มีขนาดเล็ก แสดงว่าเวกเตอร์การเคลื่อนที่ของบล็อกปัจจุบันมีความสัมพันธ์กับเวกเตอร์การเคลื่อนที่ของบล็อกข้างเคียงสูง ความหนาแน่นในการกระจายตัวเมื่อใช้เวกเตอร์การเคลื่อนที่ซึ่งคำนวณจากกระบวนการค้นหาแบบทั้งหมดของลำดับภาพ “Carphone” “Mother and daughter” และ “Foreman” เป็นดังในรูปที่ 3.2(ก) 3.2(ข) และ 3.2(ค) ตามลำดับ โดยใช้หน้าต่างการค้นหาเป็น  $\pm 7$  จุดภาพ ขนาดบล็อกแต่ละบล็อกเป็น  $8 \times 8$  จุดภาพ ทดสอบกับเฟรมจำนวน 49 เฟรม การหาความสัมพันธ์คิดจากเวกเตอร์การเคลื่อนที่จำนวน 17493 เวกเตอร์ในแต่ละลำดับภาพ



รูปที่ 3.2(ก) สหสัมพันธ์ของเวกเตอร์การเคลื่อนที่ระหว่างบล็อกเมื่อทดสอบกับลำดับภาพ “Carphone”



รูปที่ 3.2(ข) สหสัมพันธ์ของเวกเตอร์การเคลื่อนที่ระหว่างบล็อกเมื่อทดสอบกับ  
ลำดับภาพ “Mother and daughter”



รูปที่ 3.2(ค) สหสัมพันธ์ของเวกเตอร์การเคลื่อนที่ระหว่างบล็อกเมื่อทดสอบกับ  
ลำดับภาพ “Foreman”

ผลการทดสอบแสดงว่า เวกเตอร์การเคลื่อนที่ของบล็อกหนึ่ง ๆ มีสหสัมพันธ์กับเวกเตอร์การเคลื่อนที่ของบล็อกข้างเคียงเป็นอย่างมากแสดงถึงคุณลักษณะของสหสัมพันธ์ระหว่างกันในเวกเตอร์การเคลื่อนที่

### 3.2 การเลือกเวกเตอร์การเคลื่อนที่ตอนเริ่มต้นค้นหา

โดยทั่วไปแล้ว กระบวนการประมาณการเคลื่อนที่จะเลือกเวกเตอร์การเคลื่อนที่ศูนย์ให้เป็นเวกเตอร์การเคลื่อนที่เริ่มแรกในการค้นหา แต่ภาพส่วนใหญ่มีลักษณะการเคลื่อนที่ที่มีขนาดไม่มากและค่อนข้างมีความสม่ำเสมอ [18] เมื่ออาศัยสหสัมพันธ์ระหว่างเวกเตอร์การเคลื่อนที่ตามที่กล่าวในหัวข้อ 3.1 ช่วยให้ได้เวกเตอร์การเคลื่อนที่เริ่มต้นซึ่งมีความใกล้เคียงกับเวกเตอร์การเคลื่อนที่ที่เหมาะสมที่สุดมากขึ้น ตัวอย่างของอัลกอริทึมการทำนายเวกเตอร์การเคลื่อนที่ในช่วงเริ่มแรก ได้แก่

#### 3.2.1 การทำนายโดยค่าเฉลี่ย (Mean prediction) [19]

อ้างอิงจากรูปที่ 3.1 อัลกอริทึมนี้พิจารณาบล็อกข้างเคียง 3 บล็อก ได้แก่ B1 B3 และ B4 ได้เวกเตอร์การเคลื่อนที่เริ่มต้นจากการทำนาย  $MV_{init}$  ซึ่งเป็นค่าเฉลี่ยของทั้งสามบล็อกดังกล่าว

$$MV_{init} = \frac{1}{3}(MV_1 + MV_3 + MV_4) \quad (3.2)$$

#### 3.2.2 การทำนายแบบเน้นค่าเฉลี่ย (Mean-biased prediction) [17]

พิจารณาบล็อกข้างเคียง 4 บล็อก ได้แก่ B1 B2 B3 และ B4 โดยในขั้นตอนแรกหาค่าเฉลี่ยของเวกเตอร์การเคลื่อนที่ทั้ง 4 เวกเตอร์ ได้เป็น

$$MV_{mean} = \frac{1}{4} \sum_{i=1}^4 MV_i \quad (3.3)$$

ถ้าเวกเตอร์การเคลื่อนที่ของทั้งสี่บล็อกมีค่าใกล้เคียงกับค่าเฉลี่ย จะตั้งสมมติฐานว่า ความน่าจะเป็นที่ทั้งห้าบล็อกในรูปที่ 3.1 เป็นของวัตถุเดียวกันมีสูง หรือไม่ก็ทั้งห้าบล็อกดังกล่าวเป็นส่วนหนึ่งของฉากหลังที่ไม่มีการเคลื่อนที่ทั้งหมด ดังนั้น เวกเตอร์การเคลื่อนที่ของ 4 บล็อกข้างเคียงนี้สามารถใช้ทำนายเวกเตอร์การเคลื่อนที่ได้ ดังสมการ

$$MV_{init} = \begin{cases} MV_{pred}, \max_{i=1to4} \|MV_{mean} - MV_i\| < T \\ (0,0), otherwise \end{cases} \quad (3.4)$$

โดยที่ T เป็นค่าจุดเปลี่ยนกำหนดความคล้ายคลึงกันของเวกเตอร์การเคลื่อนที่ทั้งสี่บล็อก และเวกเตอร์จากการทำนายมีค่าเป็น

$$MV_{pred} = \arg \min_{MV_i} \|MV_i - MV_{mean}\|, i = 1,2,3,4 \quad (3.5)$$

### 3.3 การค้นหารูปเพชรชนิดไม่สมมาตรแบบปรับตัวได้โดยใช้การปรับย้ายจุดศูนย์กลาง

อัลกอริทึมการค้นหารูปเพชรได้รับการยอมรับในโครงสร้างของรูปแบบการค้นหาที่มีขนาดเล็กซึ่งมีความเหมาะสมกับลักษณะการกระจายตัวแบบเน้นจุดศูนย์กลางของเวกเตอร์การเคลื่อนที่ [20] มีงานวิจัยต่าง ๆ ที่พัฒนาจากอัลกอริทึมการค้นหารูปเพชรธรรมดา [21, 22]

ขั้นตอนของวิธีการค้นหาที่นำเสนอนี้ มีการแบ่งเป็นส่วนต่าง ๆ ได้แก่ ส่วนของการจำแนกลักษณะการเคลื่อนที่ของลำดับภาพ ส่วนของการทำนายเวกเตอร์การเคลื่อนที่ ส่วนของการปรับรูปแบบการค้นหา รวมถึงส่วนของการปรับย้ายจุดศูนย์กลางในการเริ่มต้นค้นหา

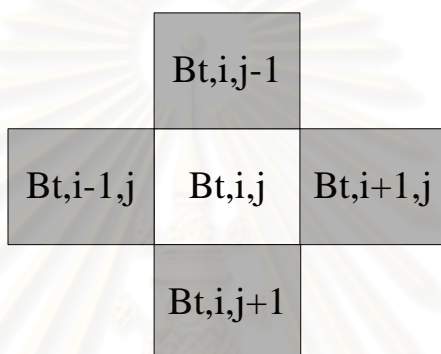
#### 3.3.1 การจำแนกลักษณะการเคลื่อนที่และการทำนายเวกเตอร์การเคลื่อนที่ [21]

Ming-gang Liu ได้เสนอการนำเทคนิคการจำแนกลักษณะการเคลื่อนที่มาใช้ร่วมกับการทำนายเวกเตอร์การเคลื่อนที่ การจำแนกลักษณะการเคลื่อนที่เป็นการบ่งบอกให้ทราบว่า วัตถุซึ่งเคลื่อนที่อยู่ในลำดับภาพที่สนใจนั้นมีแนวโน้มของอัตราเร็วในการเคลื่อนที่เป็นเช่นไร ซึ่งสามารถใช้ประโยชน์ในการปรับรูปแบบการค้นหาในกระบวนการประมาณการเคลื่อนที่ต่อไป รูปแบบของบล็อกที่เลือกใช้ในการจำแนกลักษณะการเคลื่อนที่พัฒนามาจากรูปแบบของบล็อกข้างเคียงในรูปที่ 3.3



0	1	2
3	B	4
5	6	7

รูปที่ 3.3 รูปแบบของบล็อกข้างเคียงในเชิงปริภูมิ



รูปที่ 3.4 รูปแบบของบล็อกต่าง ๆ ที่ใช้ในการจำแนกลักษณะการเคลื่อนที่

รูปที่ 3.3 แสดงถึงรูปแบบของบล็อกข้างเคียงในเชิงปริภูมิทั้งแปดบล็อก เมื่อกำหนดให้บล็อกตำแหน่งตรงกลาง (บล็อก B ดังในรูป) เป็นบล็อกปัจจุบันที่สนใจ ซึ่ง Lijun Luo [19] แสดงให้เห็นว่า บล็อกหมายเลข 0 มีความสัมพันธ์กับบล็อกหมายเลข 1 และหมายเลข 3 เป็นอย่างมาก ทำนองเดียวกับบล็อกหมายเลข 2 5 และ 7 รูปแบบของบล็อกที่ใช้ในการจำแนกและทำนายเวกเตอร์การเคลื่อนที่ได้ปรับปรุงให้มีลักษณะดังรูปที่ 3.4 และเพื่อให้สามารถใช้ประโยชน์จากสหสัมพันธ์ของเวกเตอร์การเคลื่อนที่ในเชิงปริภูมิ-เวลา จึงมีการเลือกใช้เวกเตอร์การเคลื่อนที่จากเฟรมอดีตร่วมด้วย

- กำหนดให้  $f_t$  แทนเฟรมปัจจุบันในลำดับภาพสัญญาณวิดีโอ  
 $f_{t-1}$  แทนเฟรมก่อนหน้าในลำดับภาพสัญญาณวิดีโอ  
 $B_{t,i,j}$  แทนบล็อกในตำแหน่ง  $(i,j)$  ในเฟรมลำดับที่  $t$   
 $b_{t,i,j}$  แทนเวกเตอร์การเคลื่อนที่ของบล็อกที่  $i$  ในเฟรมลำดับที่  $t$

$$\begin{aligned} \text{ทั้งนี้ กำหนดให้ } B &= [b_{t-1,i,j} \quad b_{t-1,i+1,j} \quad b_{t,i,j-1} \quad b_{t,i-1,j} \quad b_{t-1,i,j+1}] \\ &= [b_0 \quad b_1 \quad b_2 \quad b_3 \quad b_4] \end{aligned} \quad (3.6)$$

เป็นบล็อกต่าง ๆ ที่ใช้จำแนกลักษณะการเคลื่อนที่ โดยทุก ๆ องค์ประกอบของ B เป็นสองมิติ เพราะเวกเตอร์การเคลื่อนที่มีทั้งแนวแกน X และแนวแกน Y ประเภทของลักษณะการเคลื่อนที่ที่ใช้จำแนก มีดังนี้ [21]

### 3.3.1.1 ประเภทที่ 1 การเคลื่อนที่แบบทั่วไป

การเคลื่อนที่แบบทั่วไปในที่นี้ หมายถึง วัตถุในลำดับภาพไม่เคลื่อนไหว คงที่ หยุดนิ่งหรือเกือบจะหยุดนิ่ง (stationary or quasi-stationary) การเคลื่อนที่ประเภทนี้จำแนกด้วยเงื่อนไขว่า

$$\|b_i - b_0\| \leq R_0 \quad \text{สำหรับ } \forall i \in [0,4] \quad (3.7)$$

และทำนายเวกเตอร์การเคลื่อนที่ได้ดังสมการ

$$b_t^p = \text{sign}(\cdot) \times I[|b_{t-1,i,j} + b_{t,i,j-1} + b_{t,i-1,j}| / 3.0 + 0.35] \quad (3.8)$$

โดยที่  $\text{sign}(\cdot)$  เป็นฟังก์ชันเครื่องหมาย ขึ้นอยู่กับเครื่องหมายของ  $(b_{t-1,i,j} + b_{t,i,j-1} + b_{t,i-1,j})$   
 $I[\cdot]$  เป็นฟังก์ชันการประมาณให้เป็นจำนวนเต็มทีละที่สุดทิศทางเข้าหาศูนย์  
 (rounding toward zero)

### 3.3.1.2 ประเภทที่ 2 การเคลื่อนที่แบบเร็ว

การเคลื่อนที่แบบนี้เป็นลักษณะการเคลื่อนที่ที่ไม่ใช่แบบทั่วไป มีเงื่อนไขในการจำแนก คือ

$$\|b_i - b_0\| > R_0 \quad \text{สำหรับ } \exists i \in [0,4] \quad (3.9)$$

และทำนายเวกเตอร์การเคลื่อนที่ได้ดังสมการ

$$b_{i,j}^p = I[\lambda B^T] \quad (3.10)$$

โดยที่  $\lambda = [a_0 \ a_1 \ a_2 \ a_3 \ a_4]$  เป็นสัมประสิทธิ์ในการถ่วงน้ำหนัก ซึ่งสอดคล้องกับสมการ

$$\sum_{i=0}^4 a_i = 1 \quad (3.11)$$

ตามหลักการของการถ่วงน้ำหนักเชิงเส้น ซึ่งในวิทยานิพนธ์ฉบับนี้ได้ทำการเปลี่ยนค่า  $\lambda$  ต่าง ๆ กัน ซึ่งกล่าวถึงในภาคผนวก ก

### 3.3.2 การปรับรูปแบบการค้นหาของอัลกอริทึมการค้นหารูปเพชร

จากขั้นตอนที่ 3.3.1 ได้ตรวจสอบลักษณะการเคลื่อนที่ของลำดับภาพแล้ว ในขั้นตอนนี้ นำข้อมูลที่ได้มาใช้ในการปรับรูปแบบการค้นหาของอัลกอริทึมการค้นหารูปเพชร

#### 3.3.2.1 รูปแบบการค้นหารูปเพชรชนิดไม่สมมาตร

เมื่อได้การทำนายเวกเตอร์การเคลื่อนที่แล้ว ขั้นตอนต่อไปจะทำการคำนวณมุมของเวกเตอร์การเคลื่อนที่ ด้วยสมการต่อไปนี้

$$\theta_{rad} = \arcsin \left[ \frac{|y|}{\sqrt{x^2 + y^2}} \right] \quad (3.12)$$

$$\theta_{deg} = \theta_{rad} \times \left( \frac{180}{\pi} \right) \quad (3.13)$$

ค่าของมุมที่นำมาใช้ พิจารณาจากควอטרันท์ของเวกเตอร์ด้วยตารางที่ 3.1

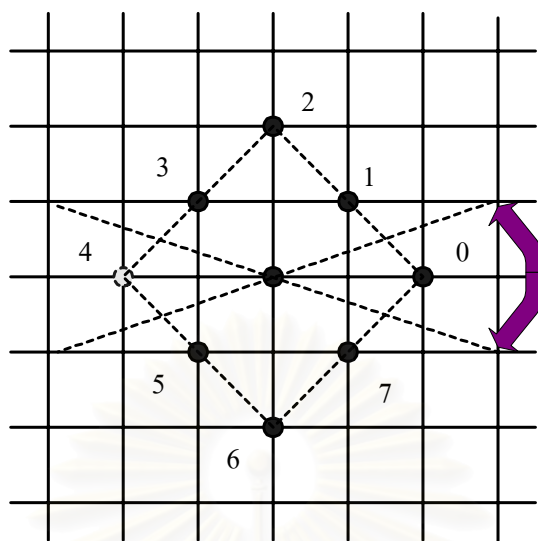
ตารางที่ 3.1 มุมของเวกเตอร์การเคลื่อนที่ที่ใช้กำหนดทิศทาง

ควอטרันท์ของเวกเตอร์การเคลื่อนที่	มุมของเวกเตอร์การเคลื่อนที่
ควอטרันท์ที่ 1	$\theta_{deg}$
ควอทรันท์ที่ 2	$180 - \theta_{deg}$
ควอทรันท์ที่ 3	$180 + \theta_{deg}$
ควอทรันท์ที่ 4	$360 - \theta_{deg}$

ขั้นตอนต่อไปจะทำการกำหนดทิศทางให้กับเวกเตอร์การเคลื่อนที่โดยอาศัยมุมที่คำนวณได้ ดังแสดงในตารางที่ 3.2 เมื่อได้กำหนดทิศทางแล้ว รูปแบบการค้นหารูปเพชรแบบปกติถูกเปลี่ยนแปลงเป็นรูปแบบการค้นหารูปเพชรชนิดไม่สมมาตร กล่าวคือ จุดการค้นหาด้านที่อยู่ตรงข้ามกับทิศทางที่ได้ระบุไว้จะถูกตัดทิ้งไป เพราะเป็นจุดการค้นหาที่มีโอกาสที่จะไม่เป็นจุดต่ำสุดที่สูงสุด เวกเตอร์การเคลื่อนที่มีแนวโน้มว่าไม่อยู่ในบริเวณนั้น

ตารางที่ 3.2 การกำหนดทิศทางให้กับเวกเตอร์การเคลื่อนที่ที่ทำนายไว้

มุมของเวกเตอร์การเคลื่อนที่ที่ทำนายไว้	หมายเลขทิศทาง	จุดการค้นหาที่ถูกตัด
มุม = $0^\circ$	-	-
$337.5^\circ \leq \text{มุม}$ หรือ $\text{มุม} < 22.5^\circ$	0	4
$22.5^\circ \leq \text{มุม} < 67.5^\circ$	1	5
$67.5^\circ \leq \text{มุม} < 112.5^\circ$	2	6
$112.5^\circ \leq \text{มุม} < 157.5^\circ$	3	7
$157.5^\circ \leq \text{มุม} < 202.5^\circ$	4	0
$202.5^\circ \leq \text{มุม} < 247.5^\circ$	5	1
$247.5^\circ \leq \text{มุม} < 292.5^\circ$	6	2
$292.5^\circ \leq \text{มุม} < 337.5^\circ$	7	3

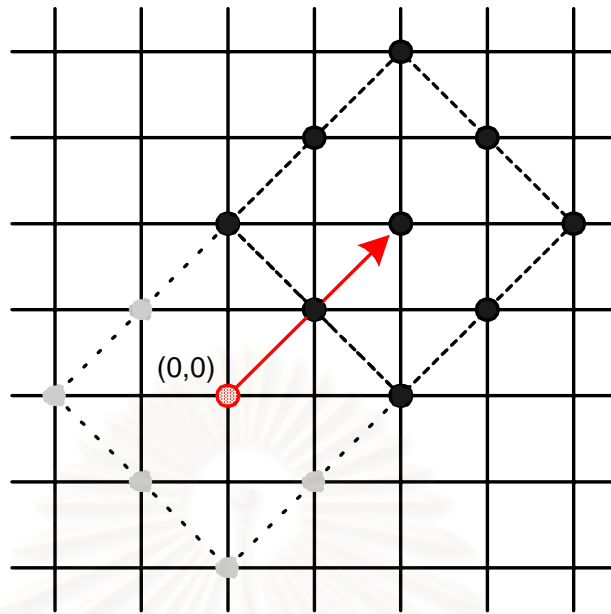


รูปที่ 3.5 รูปแบบการค้นหารูปเพชรชนิดไม่สมมาตร

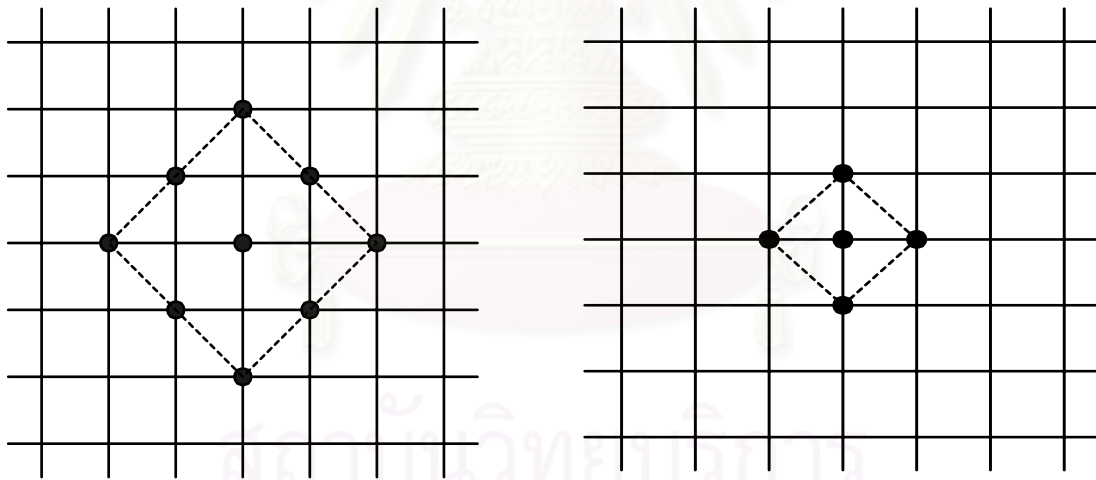
รูปที่ 3.5 เป็นตัวอย่างของรูปแบบการค้นหารูปเพชรชนิดไม่สมมาตร สมมุติว่าทำนายทิศทางเวกเตอร์การเคลื่อนที่ได้เป็นทิศทางหมายเลข 0 (นั่นคือ มุม  $< 22.5^\circ$  หรือมุม  $\geq 337.5^\circ$ ) จุดการค้นหาหมายเลข 4 ในรูปแบบการค้นหารูปเพชรปกติจะถูกตัดทิ้งไป ไม่นำมาคิดในกระบวนการตรวจหาเวกเตอร์การเคลื่อนที่

### 3.3.3 การปรับย้ายจุดศูนย์กลางการค้นหา

เพื่อให้สามารถค้นหาเวกเตอร์การเคลื่อนที่ได้รวดเร็วและลดความซับซ้อนในการคำนวณยิ่งขึ้นพบว่าสามารถใช้ประโยชน์จากเวกเตอร์การเคลื่อนที่ที่ทำนายไว้แล้วมากำหนดตำแหน่งเริ่มต้นในการค้นหาได้ แทนที่จะเริ่มต้นการค้นหาที่จุด (0,0) ตลอดเวลา โดยการเปรียบเทียบระหว่างค่าความผิดพลาดของจุดการค้นหาที่ตำแหน่ง (0,0) กับจุดที่ได้จากการทำนาย ถ้าจุดใดให้ความผิดพลาดต่ำกว่าก็เลือกจุดนั้นเป็นจุดศูนย์กลางใหม่ในการค้นหา ดังในรูปที่ 3.6 จุดแรกที่ทำตำแหน่ง (0,0) คือจุดศูนย์กลางการค้นหาเดิม เวกเตอร์ในรูป คือ เวกเตอร์ที่ได้จากการทำนาย ในที่นี้สมมติว่า จุดศูนย์กลางใหม่ให้ค่าความผิดพลาดต่ำกว่า ในกรณีนี้จึงย้ายรูปแบบการค้นหามาเป็นเส้นทึบ



รูปที่ 3.6 ลักษณะการย้ายจุดศูนย์กลางการค้นหา



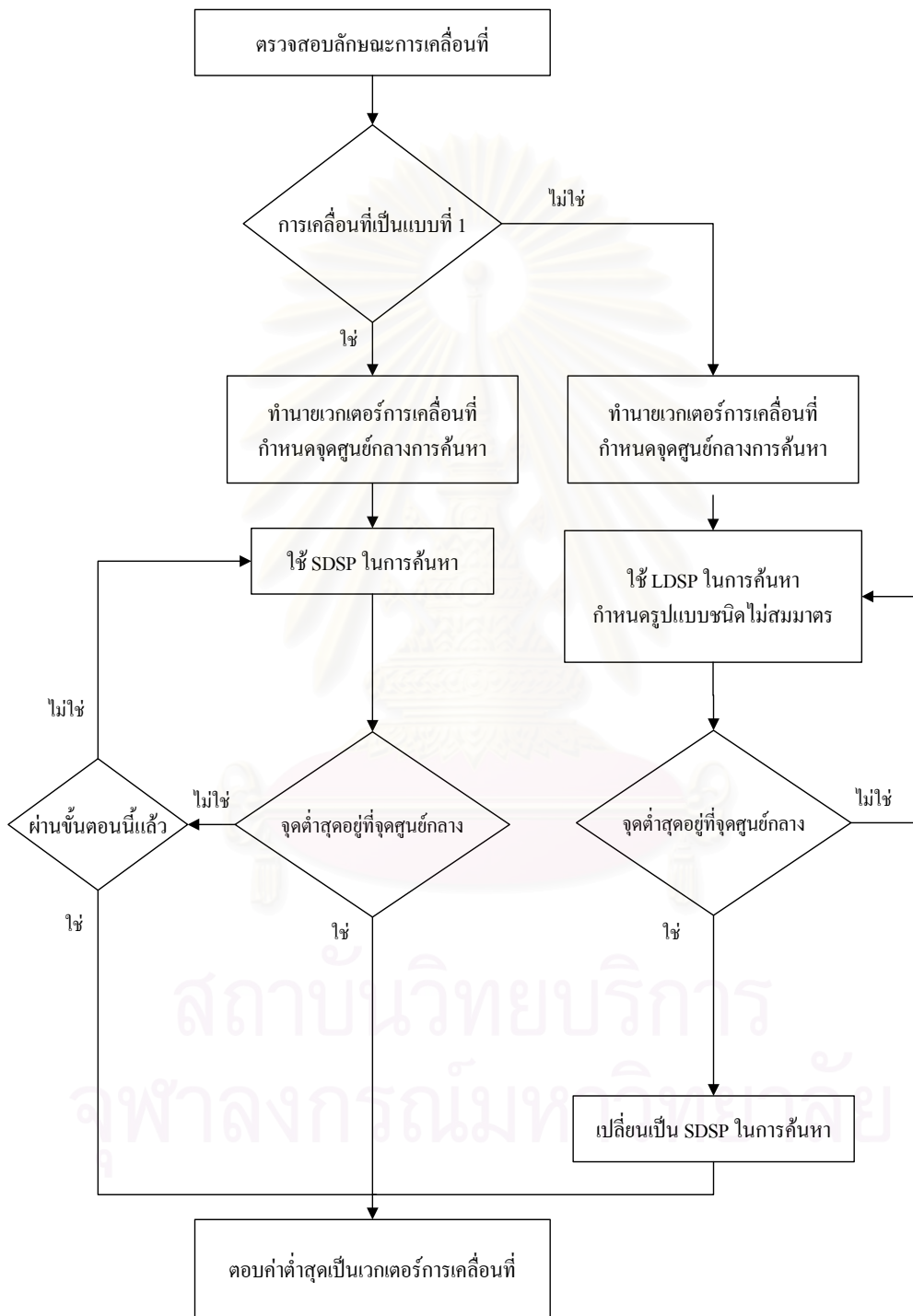
(ก)

(ข)

รูปที่ 3.7 รูปแบบการค้นหารูปเพชร (ก) รูปแบบการค้นหารูปเพชรใหญ่ (Large Diamond Search Pattern: LDSP) (ข) รูปแบบการค้นหารูปเพชรเล็ก (Small Diamond Search Pattern: SDSP)

3.3.4 ขั้นตอนการค้นหา

ขั้นตอนของวิธีการค้นหารูปเพชรชนิดไม่สมมาตรแบบปรับตัวได้โดยใช้การปรับย้ายจุดศูนย์กลาง แสดงไว้ในรูปที่ 3.8



รูปที่ 3.8 ขั้นตอนการทำงานของวิธีการค้นหารูปเพชรชนิดไม่สมมาตรแบบปรับตัวได้โดยใช้การปรับย้ายจุดศูนย์กลาง

ขั้นที่ 1 : ตรวจสอบลักษณะการเคลื่อนที่ของวัตถุในภาพ โดยใช้เวกเตอร์การเคลื่อนที่เป็นตัวบ่งบอก โดยเลือกรวมกลุ่มของบล็อกข้างเคียงที่สนใจ คำนวณหาระยะห่างแบบยูคลิดเดียวระหว่างเวกเตอร์การเคลื่อนที่ทุกตัวในกลุ่มกับเวกเตอร์ที่ตำแหน่งเดียวกันบนพรมอคิด ถ้าพบว่ามีค่าน้อยกว่าค่าจุดเปลี่ยนที่กำหนดไว้ จัดแบ่งเป็นลักษณะการเคลื่อนที่แบบทั่วไป (แบบที่ 1) ในทางกลับกัน หากพบว่าเวกเตอร์การเคลื่อนที่ใดในกลุ่มมีค่ามากกว่าค่าจุดเปลี่ยนดังกล่าว จัดเป็นลักษณะการเคลื่อนที่แบบเร็ว (แบบที่ 2)

ขั้นที่ 2(ก) : กรณีที่ตรวจสอบพบว่าลักษณะการเคลื่อนที่เป็นแบบทั่วไป (แบบที่ 1) ใช้สมการ (3.4) ในการทำนายเวกเตอร์การเคลื่อนที่ของบล็อกปัจจุบัน  $b_t^p$  และใช้รูปแบบการค้นหาเพชรแบบเล็ก (SDSP) ดังแสดงในรูปที่ 3.7(ข) ในการค้นหาเวกเตอร์การเคลื่อนที่ ซึ่งจุดศูนย์กลางการค้นหาเลือกจากตำแหน่งที่ให้ค่าต่ำสุดระหว่างจุดศูนย์กลางการค้นหาเดิม (0,0) กับตำแหน่งที่ได้จากการทำนาย  $b_t^p$  คำนวณหาค่าความผิดพลาด กรณีที่ได้จุดต่ำสุดอยู่บริเวณจุดศูนย์กลางของรูปแบบการค้นหา ตอบค่านั้นเป็นเวกเตอร์การเคลื่อนที่ นอกจากนี้ไปที่ขั้นตอน 3

(ข) : กรณีที่ตรวจสอบพบว่าลักษณะการเคลื่อนที่เป็นแบบเร็ว (แบบที่ 2) ใช้สมการ (3.6) ในการทำนายเวกเตอร์การเคลื่อนที่ของบล็อกปัจจุบัน  $b_t^p$  กำหนดรูปแบบการค้นหารูปเพชรชนิดไม่สมมาตร ดังแสดงในหัวข้อ 3.3.2.1 จุดเริ่มต้นในการค้นหาทำการปรับย้ายในทำนองเดียวกับขั้นตอนที่ 2(ก) กรณีที่ได้จุดต่ำสุดอยู่บริเวณจุดศูนย์กลางของรูปแบบการค้นหาให้ไปที่ขั้นตอนที่ 5 นอกจากนี้ไปที่ขั้นตอนที่ 4

ขั้นที่ 3 : กำหนดให้ จุดที่ให้ค่าความผิดพลาดต่ำสุดในขั้นตอนก่อนหน้านี้ เป็นจุดศูนย์กลางของรูปแบบการค้นหาใหม่ ตำแหน่งที่ให้ค่าต่ำสุดในขั้นตอนนี้ให้ถือเป็น เวกเตอร์การเคลื่อนที่

ขั้นที่ 4 : กำหนดให้ จุดที่ให้ค่าความผิดพลาดต่ำสุดในขั้นตอนก่อนหน้านี้ เป็นจุดศูนย์กลางของรูปแบบการค้นหาใหม่ ถ้าพบว่าจุดต่ำสุดอยู่ที่จุดศูนย์กลางไปที่ขั้นตอนที่ 5 นอกจากนี้ให้ทำขั้นตอนนี้ซ้ำอีกครั้ง

ขั้นที่ 5 : เปลี่ยนรูปแบบการค้นหาจากรูปแบบการค้นหาเพชรใหญ่ มาเป็นรูปแบบการค้นหาเพชรเล็ก (SDSP) ซึ่งเป็นการหาเวกเตอร์การเคลื่อนที่ขั้นสุดท้าย ค่าต่ำสุดที่ได้ตอบเป็นเวกเตอร์การเคลื่อนที่



### 3.4 การค้นหาแบบเพชรมิสมมาตร

อัลกอริทึมที่นำเสนอนี้ไม่มีการปรับตัว กล่าวคือ ใช้รูปแบบการค้นหาแบบเพชรมิสมมาตรใหญ่ในการค้นหาก่อนเสมอ ขั้นตอนในการค้นหาเหมือนกับอัลกอริทึมรูปเพชรมิสมมาตร แต่แตกต่างตรงที่มีการใช้รูปแบบมิสมมาตร ดังในหัวข้อที่ 3.3.2.1 มาใช้ในการค้นหา

สำหรับการทำนายเวกเตอร์การเคลื่อนที่ ใช้การหามัชฐานของเวกเตอร์การเคลื่อนที่จากบล็อกข้างเคียง ดังแสดงในรูปที่ 3.9

MV2 ↗	MV3 ↗
MV1 ↗	MV

รูปที่ 3.9 รูปแบบของบล็อกข้างเคียงที่ใช้ในการทำนายเวกเตอร์การเคลื่อนที่

โดย MV : เวกเตอร์การเคลื่อนที่ของบล็อกปัจจุบัน

MV1 : เวกเตอร์การเคลื่อนที่ของบล็อกทางซ้าย

MV2 : เวกเตอร์การเคลื่อนที่ของบล็อกบนซ้าย

MV3 : เวกเตอร์การเคลื่อนที่ของบล็อกด้านบน

ให้เวกเตอร์การเคลื่อนที่ที่ทำนายเป็น  $\vec{P} = (P_x, P_y)$  โดย

$$\vec{P}_x = \text{median}(MV1_x, MV2_x, MV3_x) \quad (3.14)$$

$$\vec{P}_y = \text{median}(MV1_y, MV2_y, MV3_y) \quad (3.15)$$

วิธีการต่อจากนี้เหมือนกับอัลกอริทึมการค้นหาแบบเพชรมิสมมาตร ดังได้กล่าวมาแล้ว

## บทที่ 4

### ผลการทดสอบ

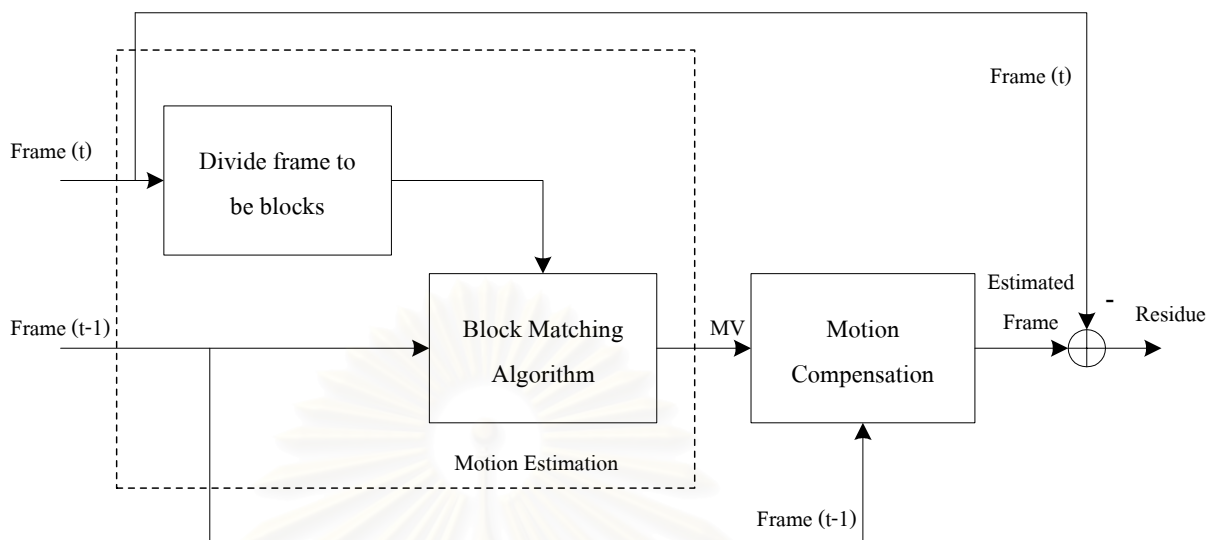
ในบทนี้เป็นการทดสอบสมรรถนะอัลกอริทึมที่นำเสนอเปรียบเทียบกับอัลกอริทึมการค้นหาแบบทั้งหมด รวมทั้งอัลกอริทึมแบบเร็วชนิดต่าง ๆ ได้แก่ อัลกอริทึมการค้นหาแบบสามขั้นตอน อัลกอริทึมการค้นหาแบบสี่ขั้นตอน อัลกอริทึมการค้นหารูปเพชร และทำการวิเคราะห์และวิจารณ์ผลการทดสอบต่าง ๆ

#### 4.1 ระบบที่ใช้ในการทดสอบและชุดภาพที่ใช้ในการทดสอบ

บล็อกไดอะแกรมของระบบที่ใช้ในการทดสอบการทำงานของอัลกอริทึมการเข้าสู่บล็อกในการประมาณการเคลื่อนที่ เป็นดังแสดงในรูปที่ 4.1 มีรายละเอียด ดังต่อไปนี้

1. แบ่งเฟรมภาพปัจจุบันออกเป็นบล็อกสี่เหลี่ยมจัตุรัสย่อย ๆ ที่ไม่ซ้อนทับกัน
2. ทำการเข้าสู่บล็อกเปรียบเทียบเฟรมปัจจุบันกับเฟรมในอดีต เพื่อทำการหาเวกเตอร์การเคลื่อนที่ตามอัลกอริทึมที่เลือกใช้
3. นำเวกเตอร์การเคลื่อนที่ที่หาได้จากขั้นตอนที่ 2 มาสร้างเฟรมประมาณ
4. นำเฟรมที่ได้จากการประมาณเปรียบเทียบกับเฟรมในอดีตเพื่อตรวจสอบค่าความเพี้ยนจากการประมาณ

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.1 บล็อกไดอะแกรมแสดงระบบการทดสอบที่ใช้ในวิทยานิพนธ์

#### 4.1.1 ระบบที่ใช้ในการทดสอบ

ระบบที่ใช้ในการทดสอบ มีข้อกำหนดต่าง ๆ รวมถึงลักษณะการทดสอบเบื้องต้นที่ใช้ในงานวิจัย ดังนี้

สัญญาณภาพที่ใช้ในการทดสอบเป็นสัญญาณภาพตามมาตรฐาน QCIF ขนาด 144 x 176 จุดภาพ โดยใช้เฉพาะค่าของความเข้มแสง (luminance value) ซึ่งมี 256 ระดับหรือ 8 บิต ทำการทดสอบตั้งแต่เฟรมที่ 1 ไปถึงเฟรมที่ 50 เป็นจำนวนทั้งสิ้น 49 เฟรม ขนาดของบล็อกกำหนดไว้ที่ 8 x 8 จุดภาพ กระบวนการเข้าคู่บล็อกใช้หน้าต่างการค้นหาค้นหาขนาด 15 x 15 จุดภาพ วัดค่าความผิดพลาดของแต่ละบล็อกด้วยฟังก์ชันค่าความผิดพลาดสัมบูรณ์เฉลี่ย (Mean Absolute Error: MAE)

#### 4.1.2 ชุดภาพที่ใช้ในการทดสอบ

ในวิทยานิพนธ์ฉบับนี้ ใช้ลำดับภาพมาตรฐาน 11 ชุด ดังในรูปที่ 4.2 ซึ่งได้รับความเอื้อเฟื้อจาก ดร. วุฒิพงษ์ อารีกุล ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ เพื่อการทดสอบสมรรถนะของอัลกอริทึมแต่ละแบบ ชุดภาพมาตรฐานแต่ละแบบ มีลักษณะการเคลื่อนไหวของวัตถุในภาพแตกต่างกันไป เช่น การซูม การเลื่อนของกล้อง วัตถุเคลื่อนไหวช้า วัตถุเคลื่อนไหวเร็ว แต่ละเฟรมแบ่งออกเป็นบล็อกย่อยเป็นจำนวนทั้งสิ้น  $18 \times 22$  หรือ 396 บล็อก ดังนั้น ในแต่ละชุดภาพมีค่าเวกเตอร์การเคลื่อนที่ทั้งหมดเท่ากับจำนวนบล็อกที่ค้นหา กล่าวคือ  $49 \times 396$  หรือ 19404 เวกเตอร์การเคลื่อนที่



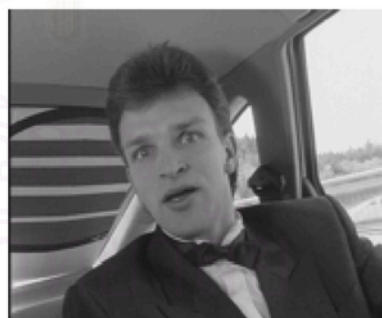
(ก) Akiyo



(ข) Claire



(ค) Miss America



(ง) Carphone



(จ) Salesman



(ฉ) Suzie



(ช) Foreman



(ฅ) Trevor



(ฉ) Mother and daughter



(ค) Grandmother

ทยบ  
จุฬาลงกรณ์มหาวิทยาลัย



(จ) Table Tennis

## รูปที่ 4.2 ชุดลำดับภาพทดสอบมาตรฐานทั้งสิบเอ็ดลำดับภาพ

ลักษณะโดยทั่วไปของชุดภาพต่าง ๆ ที่ใช้ทดสอบ มีดังนี้

1. ลำดับภาพ Akiyo : เป็นลำดับภาพที่มีการเคลื่อนที่น้อย มีฉากหลังที่ไม่เคลื่อนไหว มีเพียงภาพตัวคนที่เป็นผู้ประกาศข่าวขยับเพียงเล็กน้อย
2. ลำดับภาพ Claire : มีการเคลื่อนไหวของวัตถุช้า คล้ายกับลำดับภาพ Akiyo การเคลื่อนที่ส่วนใหญ่อยู่ที่บริเวณใบหน้า
3. ลำดับภาพ Miss America : มีการเคลื่อนที่ของวัตถุปานกลาง มีการเคลื่อนไหวของศีรษะคน ภาพฉากหลังไม่เคลื่อนที่
4. ลำดับภาพ Carphone : เป็นลำดับภาพที่มีการเคลื่อนไหวของวัตถุเร็วเพราะเป็นภาพที่บันทึกบนรถยนต์ขณะเคลื่อนที่ มีการเคลื่อนไหวของวัตถุพื้นหน้า (ตัวคน) และภาพฉากหลังบางส่วนมีการเลื่อนตำแหน่ง
5. ลำดับภาพ Salesman : มีการเคลื่อนไหวของวัตถุปานกลาง มีการเคลื่อนไหวของวัตถุหลายองค์ประกอบและบางวัตถุมีการเคลื่อนที่ซับซ้อน เช่น มีการพับแขนและการขยี้มือ
6. ลำดับภาพ Suzie : มีการเคลื่อนที่บริเวณศีรษะและไหล่ (head and shoulder) เป็นบางช่วง มีการเงยและก้มหน้า
7. ลำดับภาพ Foreman : มีการเคลื่อนที่ของวัตถุด้วยความเร็วปานกลาง มีการเลื่อนและหมุนกล้อง ตัวคนมีการหมุนคอ เอียงคอ

8. ลำดับภาพ Trevor : มีวัตถุภายในเฟรมเดียวกันหลายวัตถุซึ่งเคลื่อนที่แตกต่างกันออกไป และมีการตัดภาพ
9. ลำดับภาพ Mother and daughter : มีการเคลื่อนที่ของวัตถุปานกลาง มีการเคลื่อนไหวหลายวัตถุ เช่น การยกแขนพร้อมกับการขยับหน้า
10. ลำดับภาพ Grandmother : มีการเคลื่อนที่ของวัตถุน้อย การเคลื่อนที่มีความเร็วไม่มาก
11. ลำดับภาพ Table tennis : การเคลื่อนที่ของวัตถุมีความเร็วสูง เช่น การเคลื่อนที่ของลูกปิงปอง การเคลื่อนที่ของแขน มีการซูมกล้อง มีการตัดภาพไปมา

#### 4.2 การทดลองเพื่อทดสอบแนวความคิด

เนื่องจากเทคนิคการค้นหาแบบเพชร ไม่สมมาตรแบบปรับตัวได้โดยใช้การปรับย้ายจุดศูนย์กลางที่นำเสนอไว้ ใช้เทคนิคในการทำนายเวกเตอร์การเคลื่อนที่เป็นหลัก ในหัวข้อนี้เป็นการตรวจสอบว่า การทำนายเวกเตอร์การเคลื่อนที่และการตัดจุดการค้นหาที่พัฒนาขึ้น มีความผิดพลาดมากน้อยเพียงใด เมื่อเทียบกับอัลกอริทึมการค้นหาแบบทั้งหมดซึ่งเป็นวิธีการที่ได้รับการยอมรับว่า ให้ค่าความถูกต้องในการประมาณการเคลื่อนที่สูงที่สุด การทดสอบแนวความคิดดังกล่าว ได้แก่

##### 4.2.1 ความน่าจะเป็นในการพบเวกเตอร์การเคลื่อนที่ที่แท้จริง

เวกเตอร์การเคลื่อนที่ที่แท้จริง (true motion vector) ในที่นี้หมายถึง เวกเตอร์การเคลื่อนที่ที่ได้จากอัลกอริทึมการค้นหาแบบทั้งหมด (full search) ดังนั้น หัวข้อนี้เป็นการตรวจสอบความน่าจะเป็นของเวกเตอร์การเคลื่อนที่ที่ได้ว่า ให้ผลลัพธ์ตรงกับเวกเตอร์การเคลื่อนที่ที่ได้จากวิธีการค้นหาแบบทั้งหมดมากน้อยเพียงใด และยังเป็นทดสอบว่า อัลกอริทึมที่สนใจนั้นมีความไวต่อการตกอยู่ในจุดต่ำสุดท้องถิ่นหรือไม่ ถ้ามีความน่าจะเป็นสูง หมายความว่า อัลกอริทึมมีความทนทานต่อการตกอยู่ในจุดต่ำสุดท้องถิ่น [20]

#### 4.2.1.1 วิธีการทดสอบ

1. เปรียบเทียบเวกเตอร์การเคลื่อนที่ที่ได้จากวิธีการค้นหาแบบทั้งหมดกับวิธีการที่ต้องการเปรียบเทียบ แล้วนับจำนวนเวกเตอร์การเคลื่อนที่ที่มีค่าเท่ากันได้ ในกรณีที่ค่าไม่เท่ากันให้คำนวณหาระยะห่างระหว่างเวกเตอร์ทั้งสอง
2. เมื่อเปรียบเทียบครบทุกเวกเตอร์การเคลื่อนที่แล้ว คำนวณค่าความน่าจะเป็นในการพบเวกเตอร์ที่แท้จริง โดยนำเอาจำนวนที่นับได้จากขั้นตอนที่หนึ่งมาหารด้วยเวกเตอร์การเคลื่อนที่ทั้งหมด
3. คำนวณค่าระยะห่างเฉลี่ยระหว่างเวกเตอร์การเคลื่อนที่ ดังจะกล่าวในหัวข้อถัดไป

#### 4.2.2 ระยะห่างเฉลี่ยระหว่างเวกเตอร์การเคลื่อนที่ที่แท้จริง

เป็นการวัดระยะทางแบบยูคลิดเดียนระหว่างเวกเตอร์การเคลื่อนที่ที่ทำการประมาณขึ้นกับเวกเตอร์การเคลื่อนที่ที่ได้จากการค้นหาแบบทั้งหมด ดังในสมการที่ 4.1

$$\Delta = \|A_i - F_i\| \quad (4.1)$$

- เมื่อ  $A_i$  เป็นเวกเตอร์การเคลื่อนที่ที่ได้จากอัลกอริทึมที่ต้องการทดสอบ  
 $F_i$  เป็นเวกเตอร์การเคลื่อนที่ที่ได้จากอัลกอริทึมการค้นหาแบบทั้งหมด  
 $i$  เป็นเวกเตอร์การเคลื่อนที่ที่ต้องการทดสอบ

แล้วนำมาหาค่าเฉลี่ย ผลของการทดลองเพื่อทดสอบแนวความคิด ดังตารางที่ 4.1 ซึ่งได้ทดลองกับวิธีการที่นำเสนอ (AADSC) กับวิธีการค้นหาแบบเพชรูปกติ (DS) วิธีการที่นำเสนอใช้สัมประสิทธิ์ในการถ่วงน้ำหนัก  $\lambda = [0.4 \ 0 \ 0.3 \ 0.3 \ 0]$  [21] เลือกใช้ค่าจุดเปลี่ยนเป็นค่าประมาณของรากที่สองของสอง (1.40) และแปด (2.83)



ตารางที่ 4.1 ผลการทดลองเพื่อทดสอบแนวความคิด

Sequences	AADSC				DS	
	Th=1.40		Th=2.83		Prob.	Av. distance
	Prob.	Av. distance	Prob.	Av. distance		
1. Akiyo	0.9988	0.0018	0.9964	0.0042	0.9994	0.0012
2. Claire	0.8482	0.9397	0.8410	0.9486	0.8585	0.9234
3. Missam	0.7471	1.2479	0.7163	1.2954	0.7643	1.2145
4. Carphone	0.8261	0.7327	0.7511	0.8179	0.8470	0.6856
5. Salesman	0.9919	0.0165	0.9822	0.0272	0.9941	0.0150
6. Suzie	0.9428	0.1410	0.8712	0.2186	0.9467	0.1448
7. Foreman	0.8604	0.5916	0.7367	0.7109	0.8547	0.6472
8. Trevor	0.9230	0.4482	0.8773	0.5009	0.9284	0.4383
9. Mother	0.8409	0.7392	0.7998	0.7875	0.8465	0.7402
10. Grandmother	0.8447	0.7593	0.8310	0.7788	0.8507	0.7461
11. Table	0.9167	0.4076	0.8532	0.8532	0.9160	0.4306

จากผลการทดลองกับลำดับภาพแบบต่าง ๆ พบว่าอัลกอริทึมการค้นหาแบบเพชรรชนิดไม่สมมาตรแบบปรับตัวได้โดยใช้การปรับย้ายจุดศูนย์กลาง ส่วนใหญ่มีความน่าจะเป็นที่จะได้เวกเตอร์การเคลื่อนที่เป็นเวกเตอร์การเคลื่อนที่ที่แท้จริงต่ำกว่าวิธีการค้นหาแบบเพชรรธรรมดาเพียงเล็กน้อย ส่วนผลของระยะห่างเฉลี่ยจากเวกเตอร์การเคลื่อนที่ที่แท้จริงได้ค่าต่ำกว่าการค้นหาแบบเพชรรธรรมดา จึงกล่าวได้ว่า อัลกอริทึมที่นำเสนอมีความผิดพลาดไม่แตกต่างจากอัลกอริทึมการค้นหาแบบเพชรรมากนัก

จุฬาลงกรณ์มหาวิทยาลัย

### 4.3 การทดสอบสมรรถนะของอัลกอริทึมการค้นหาแบบเพชรชนิดไม่สมมาตรแบบปรับตัวได้โดยใช้การปรับย้ายจุดศูนย์กลาง

เทคนิคนี้ใช้การปรับรูปร่างของโครงสร้างของจุดการค้นหาแบบเพชร ซึ่งอัลกอริทึมการค้นหาแบบเพชรปกติใช้รูปแบบในการค้นหาแบบใหญ่ก่อนเสมอ หลังจากได้ค่าต่ำสุดอยู่ที่จุดศูนย์กลางของรูปแบบการค้นหา จึงใช้รูปแบบการค้นหาแบบเล็กเป็นการหาเวกเตอร์การเคลื่อนที่แบบละเอียด ในวิธีการที่นำเสนอ เมื่อพัฒนาวิธีการทำนายลักษณะของการเคลื่อนที่ของวัตถุในลำดับภาพแล้ว สามารถนำมาใช้ประโยชน์ในการปรับรูปแบบการค้นหาของรูปแบบการค้นหาแบบเพชรดังกล่าวได้ กล่าวคือ ถ้าลักษณะการเคลื่อนที่เป็นแบบหยุดนิ่งหรือเกือบจะหยุดนิ่ง (stationary หรือ quasi-stationary) เวกเตอร์การเคลื่อนที่มีแนวโน้มที่มีขนาดไม่เกิน 1 จุดภาพสูง การใช้รูปแบบการค้นหาแบบเล็กย่อมเหมาะสมกว่าการเริ่มต้นค้นหาด้วยรูปแบบการค้นหาแบบใหญ่ก่อนตลอด นอกจากนี้ เมื่อตัดจุดการค้นหาบริเวณตำแหน่งตรงข้ามกับทิศทางของเวกเตอร์การเคลื่อนที่ที่ทำนายได้ออก ร่วมกับการปรับย้ายจุดศูนย์กลางการค้นหา ช่วยให้ความซับซ้อนลงไปได้อีก ทั้งยังลดโอกาสในการตกอยู่ในจุดต่ำสุดท้องถิ่นเมื่อทำการค้นหาได้ด้วย

#### 4.3.1 ผลการทดสอบเบื้องต้น

ผลการทดสอบแสดงในรูปแบบของกราฟด้วยการวัดค่าของจำนวนจุดการค้นหาที่ใช้ ค่าความผิดพลาดกำลังสองเฉลี่ย และค่าอัตราส่วนกำลังของสัญญาณขอดีต่อสัญญาณรบกวน สำหรับในส่วนนี้ทำการเปรียบเทียบผลที่ได้จากอัลกอริทึมการค้นหาแบบสามขั้นตอน สี่ขั้นตอนและการค้นหาแบบเพชรปกติ กราฟที่แสดงผลเลือกใช้ค่าสัมประสิทธิ์ในการถ่วงน้ำหนักเป็น  $\lambda=[0.4 \ 0 \ 0.3 \ 0.3 \ 0]$  และค่าจุดเปลี่ยนในการแบ่งแยกลักษณะการเคลื่อนที่เป็น 1.40 ซึ่งการเปลี่ยนค่าเหล่านี้ส่งผลต่อผลการทดสอบเพียงเล็กน้อยเท่านั้น ดังแสดงในภาคผนวก ก ดังนั้นจึงเลือกใช้ค่าดังกล่าวตลอดการทดสอบ สำหรับการค้นหาแบบทั้งหมดจะไม่ทำการแสดงค่าจำนวนจุดการค้นหาเพราะมีค่าสูงกว่าวิธีการค้นหาแบบอื่นมากและมีค่าคงตัว คือ เฟรมละ 89100 จุดการค้นหา

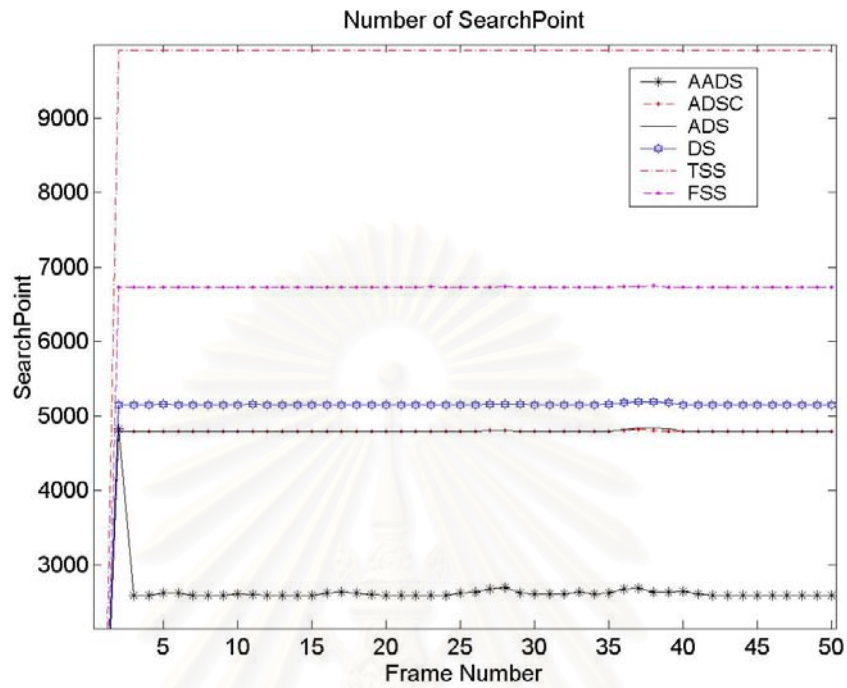
อัลกอริทึมต่าง ๆ ใช้สัญลักษณ์ ดังนี้

AADS	อัลกอริทึมการค้นหาแบบเพชรปรับตัวได้ชนิดไม่สมมาตร โดยใช้การปรับย้ายจุดศูนย์กลาง
ADSC	อัลกอริทึมการค้นหาแบบเพชรชนิดไม่สมมาตร โดยใช้การปรับย้ายจุดศูนย์กลาง
ADS	อัลกอริทึมการค้นหาแบบเพชรชนิดไม่สมมาตร
DS	อัลกอริทึมการค้นหาแบบเพชร
TSS	อัลกอริทึมการค้นหาแบบสามขั้นตอน
FSS	อัลกอริทึมการค้นหาแบบสี่ขั้นตอน
FS	อัลกอริทึมการค้นหาแบบทั้งหมด

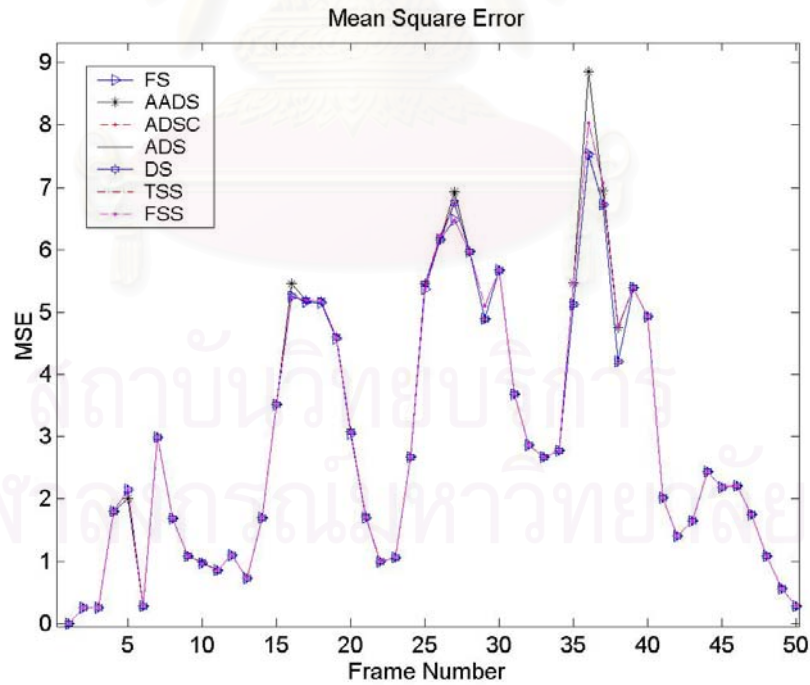


สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

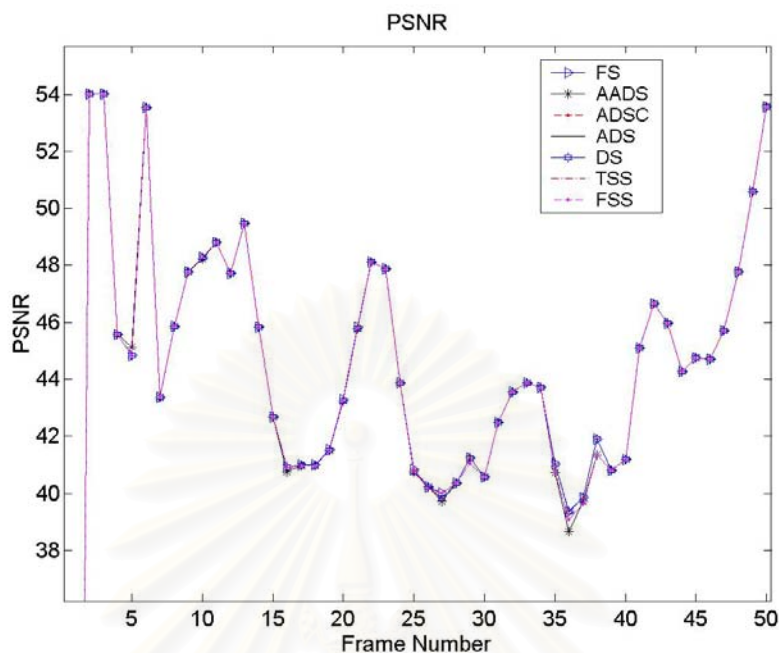
#### 4.3.1.1 ผลการทดสอบกับชุดภาพ Akiyo



รูปที่ 4.3(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Akiyo



รูปที่ 4.3(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Akiyo



รูปที่ 4.3(ค) ค่าอัตราส่วนกำลังสัญญาณยอต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Akiyo

รูปที่ 4.3(ก) แสดงผลของจำนวนจุดการค้นหาที่ใช้เปรียบเทียบระหว่างอัลกอริทึมชนิดต่าง ๆ โดยทดสอบด้วยลำดับภาพ 50 เฟรมแรกของลำดับภาพ Akiyo เมื่อไม่นับจำนวนจุดการค้นหาของอัลกอริทึมการค้นหาแบบทั้งหมด อัลกอริทึมการค้นหาแบบสามขั้นตอน (Three Step Search: TSS) ใช้จำนวนจุดการค้นหามากที่สุดและมีค่าคงที่ คือ ในแต่ละบล็อก ทำการค้นหาตลอดทั้งสามขั้นตอนต้องใช้จำนวนจุดการค้นหา เท่ากับ  $9+8+8 = 25$  จุดการค้นหา ซึ่งภายในหนึ่งเฟรมภาพ ประกอบด้วยบล็อกย่อย ๆ เป็นจำนวน  $18 \times 22 = 396$  บล็อก ดังนั้น ในแต่ละเฟรม การค้นหาแบบสามขั้นตอนใช้จำนวนจุดค้นหารวมเป็นค่าคงที่เท่ากับ  $396 \times 25 = 9,900$  จุดการค้นหา สำหรับการค้นหาแบบสี่ขั้นตอน (Four Step Search: FSS) ใช้จำนวนจุดค้นหารองลงมา ซึ่งมีค่าเฉลี่ยคือ 17 จุดการค้นหา ต่อหนึ่งเวกเตอร์การเคลื่อนที่ อัลกอริทึมการค้นหาแบบเพชรชนิดไม่สมมาตร (Asymmetric Diamond Search: ADS) สามารถลดจำนวนจุดการค้นหาจากอัลกอริทึมการค้นหาแบบเพชรธรรมดา (Diamond Search: DS) ประมาณร้อยละ 7 ผลการทดสอบกับลำดับภาพ Akiyo พบว่า อัลกอริทึมการค้นหาแบบเพชรชนิดไม่สมมาตรแบบปรับย้ายจุดศูนย์กลาง (Asymmetric Diamond Search using adaptive search Center: ADSC) และแบบไม่ปรับย้ายจุดศูนย์กลางใช้จำนวนจุดการค้นหาเกือบจะเท่ากัน เนื่องจากลำดับภาพ Akiyo เป็นภาพที่มีการเคลื่อนที่ช้าและไม่ซับซ้อน กล่าวคือ เป็นเพียงภาพ

ผู้ประกาศข่าวและฉากหลังที่ไม่เคลื่อนที่เท่านั้น เวกเตอร์การเคลื่อนที่มีขนาดเล็กไม่มาก วิธีการสุดท้ายคือ อัลกอริทึมการค้นหาแบบเพชรชนิดไม่สมมาตรแบบปรับตัวได้โดยใช้การปรับย้ายจุดศูนย์กลาง (Adaptive Asymmetric Diamond Search: AADS) สามารถลดจำนวนจุดการค้นหาจากวิธีการค้นหาแบบเพชรธรรมดาได้ถึงประมาณร้อยละ 50 จากค่าที่ได้จากการทดสอบนี้ พบว่า เมื่อลำดับภาพเป็นลักษณะที่มีวัตถุเคลื่อนที่ไม่ซับซ้อน ในทำนองเดียวกับลำดับภาพ Akiyo เช่นนี้ ใช้จำนวนจุดการค้นหาในแต่ละเฟรมค่อนข้างสม่ำเสมอ ราบเรียบเกือบเป็นเส้นตรง

รูปที่ 4.3(ข) แสดงค่าความผิดพลาดกำลังสองเฉลี่ย (Mean Square Error: MSE) ของอัลกอริทึมชนิดต่าง ๆ จากผลการทดสอบเห็นได้ว่า ค่า MSE ของแต่ละวิธีการมีค่าใกล้เคียงกัน มีเพียงบางเฟรมที่วิธีการ AADS ให้ค่าความผิดพลาดกำลังสองเฉลี่ยสูงกว่าวิธีการอื่น ๆ เล็กน้อย สำหรับเฟรมที่ให้ค่าความผิดพลาดสูงที่สุด คือ เฟรมที่ 36 ซึ่งมีค่า MSE มากกว่าอัลกอริทึมการค้นหาแบบทั้งหมดเพียง 1.31 เท่านั้น รูปที่ 4.3(ค) แสดงค่าอัตราส่วนกำลังสัญญาณขดต่อสัญญาณรบกวน เปรียบเทียบระหว่างภาพต้นแบบกับภาพที่สร้างขึ้นจากกระบวนการประมาณการเคลื่อนที่ด้วยอัลกอริทึมต่าง ๆ ซึ่งค่าที่ได้ก็มีลักษณะใกล้เคียงกันเป็นส่วนใหญ่

ตารางที่ 4.2 การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Akiyo

Algorithm	Average MSE	Prob.	Average Distance	Average Searchpoints	Speed up ratio
FS	2.9609	-	-	225	1.0
TSS	3.0057	0.9992	0.0012	25	9.0
FSS	3.0057	0.9992	0.0012	17.00	13.23
DS	2.9711	0.9994	0.0012	13.01	17.29
ADS	2.9711	0.9994	0.0012	12.11	18.58
ADSC	2.9725	0.9994	0.0013	12.10	18.59
AADS	3.0260	0.9988	0.0018	6.71	33.54

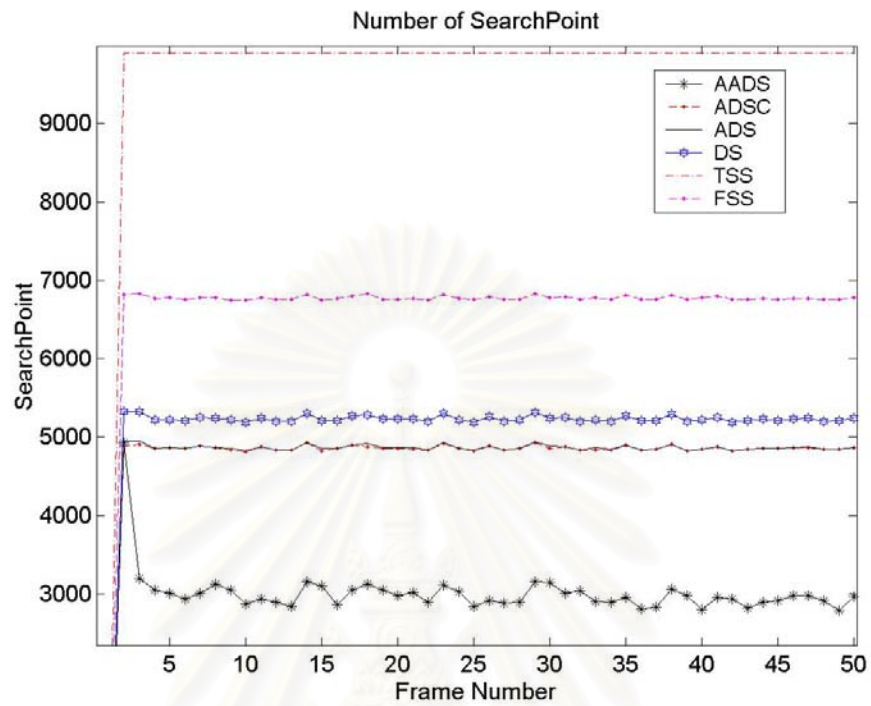
ตารางที่ 4.2 แสดงการเปรียบเทียบสมรรถนะของอัลกอริทึมแบบต่าง ๆ โดยทำการวัดค่าเฉลี่ยของค่าความผิดพลาดกำลังสองเฉลี่ยของแต่ละเฟรม ความน่าจะเป็นที่จะพบเวกเตอร์การเคลื่อนที่ตรงกับเวกเตอร์การเคลื่อนที่ของอัลกอริทึมการค้นหาแบบทั้งหมด ระยะห่างเฉลี่ยระหว่างเวกเตอร์ที่ทำได้

กับเวกเตอร์ที่ได้จากอัลกอริทึมการค้นหาแบบทั้งหมด จำนวนจุดการค้นหาเฉลี่ยที่ใช้ต่อหนึ่งเวกเตอร์ การเคลื่อนที่ และค่าอัตราส่วนความเร็วที่เพิ่มขึ้น (Speed up ratio ซึ่งเป็น อัตราส่วนระหว่างจำนวน จุดการค้นหาของอัลกอริทึมแบบทั้งหมดกับจำนวนจุดการค้นหาที่ได้จากแต่ละอัลกอริทึม) สำหรับ ลำดับภาพ Akiyo เวกเตอร์การเคลื่อนที่ที่หาได้จากแต่ละอัลกอริทึมค่อนข้างตรงกับอัลกอริทึมแบบ ทั้งหมด เพราะลำดับภาพดังกล่าว เป็นลำดับภาพที่มีการเคลื่อนไหวไม่ซับซ้อน อัลกอริทึมการค้นหา แบบเพชรชนิดไม่สมมาตรแบบปรับตัวได้โดยใช้การปรับย้ายจุดศูนย์กลาง (AADS) สามารถลดจำนวน จุดการค้นหาจากวิธีการค้นหาแบบเพชรธรรมดาได้ประมาณร้อยละ 50

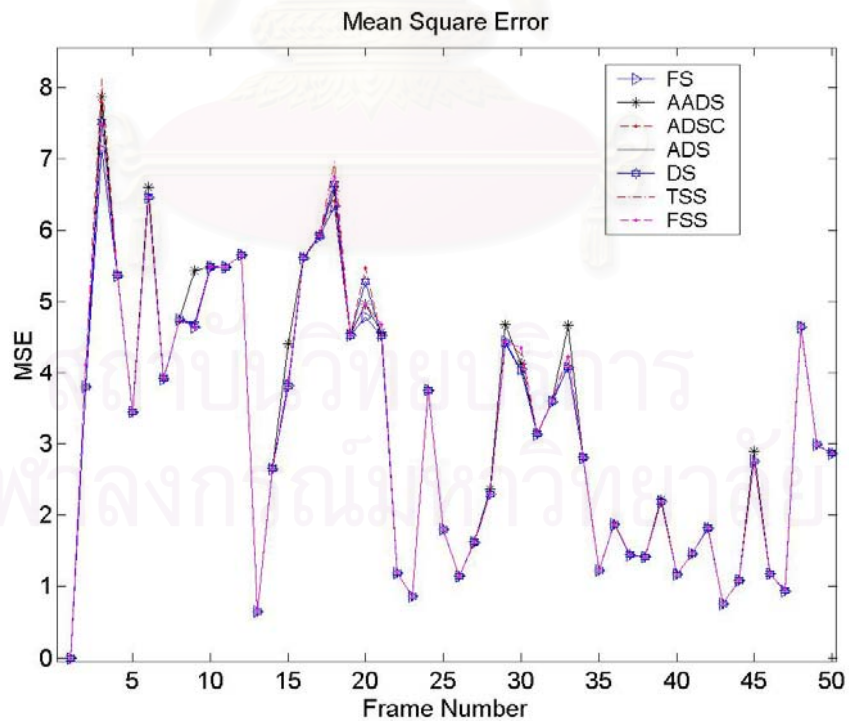


สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

#### 4.3.1.2 ผลการทดสอบกับชุดภาพ Claire

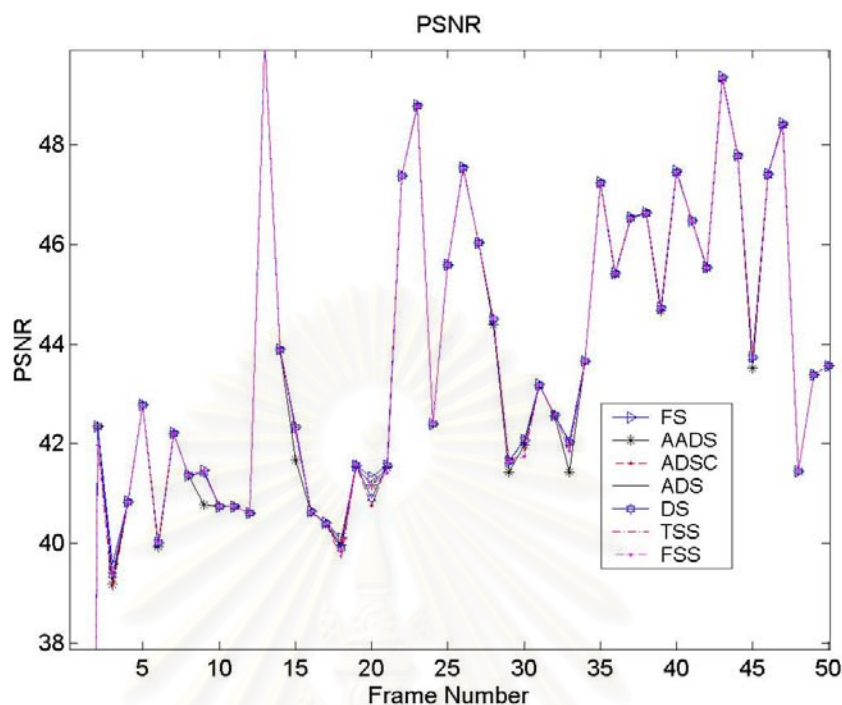


รูปที่ 4.4(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Claire



รูปที่ 4.4(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Claire





รูปที่ 4.4(ค) ค่าอัตราส่วนกำลังสัญญาณขดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Claire

รูปที่ 4.4(ก) การค้นหาแบบสามขั้นตอนใช้จำนวนจุดการค้นหาที่ ดังที่ได้กล่าวในผลการทดสอบกับลำดับภาพ Akiyo อัลกอริทึมการค้นหาแบบสี่ขั้นตอนค่อนข้างใช้จำนวนจุดการค้นหาที่สอดคล้องกับการที่ลำดับภาพ Claire ยังคงเป็นลำดับภาพที่วัตถุภายในภาพเคลื่อนไหวซ้ำคล้ายคลึงกับลำดับภาพ Akiyo เพียงแต่วัตถุในลำดับภาพ Claire มีการเคลื่อนที่เร็วกว่าเล็กน้อย อัลกอริทึมการค้นหาแบบเพชรชนิดไม่สมมาตร (ADS) สามารถลดจุดการค้นหาจากอัลกอริทึมการค้นหาแบบเพชรธรรมดา (DS) ประมาณ 7 เปอร์เซ็นต์ ในลำดับภาพ Claire การปรับย้ายจุดศูนย์กลางในการค้นหายังคงใช้จำนวนจุดการค้นหาใกล้เคียงกับการไม่ปรับย้ายจุดศูนย์กลาง ส่วนวิธี AADS ใช้จำนวนจุดการค้นหาไม่สม่ำเสมอเหมือนกับผลการทดสอบกับลำดับภาพ Akiyo

รูปที่ 4.4(ข) แสดงค่าความผิดพลาดกำลังสองเฉลี่ย (MSE) ของอัลกอริทึมชนิดต่าง ๆ จากผลการทดสอบเห็นได้ว่า ค่า MSE ของแต่ละวิธีการมีค่าใกล้เคียงกัน มีเพียงบางเฟรมที่วิธีการ AADS ให้ค่าความผิดพลาดกำลังสองเฉลี่ยสูงกว่าวิธีการอื่น ๆ เล็กน้อย สำหรับเฟรมที่ให้ความผิดพลาดสูงสุด คือเฟรมที่ 3 มีค่า MSE เท่ากับ 7.86

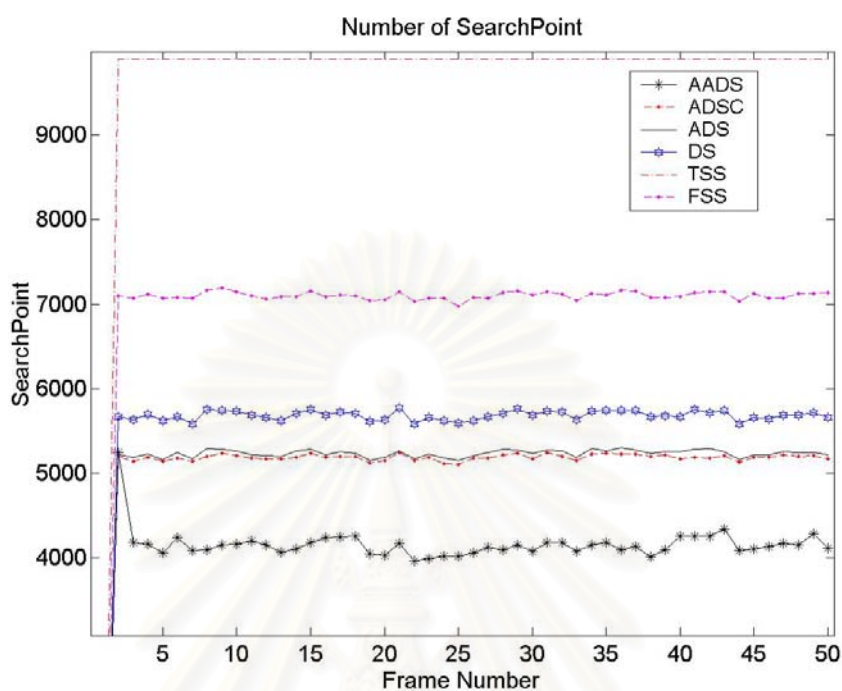
รูปที่ 4.4(ค) แสดงค่าอัตราส่วนกำลังสัญญาณขดต่อสัญญาณรบกวน เปรียบเทียบระหว่างภาพต้นแบบ กับภาพที่สร้างขึ้นจากกระบวนการประมาณการเคลื่อนที่ด้วยอัลกอริทึมต่าง ๆ ซึ่งในแต่ละวิธียังคงให้ค่า PSNR ใกล้เคียงกัน เฟรมที่มีค่า PSNR น้อยกว่าอัลกอริทึมการค้นหาแบบทั้งหมดมากที่สุด คือ เฟรมที่ 9 ซึ่งมีค่า PSNR ต่ำกว่า 0.684 db

ตารางที่ 4.3 การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Claire

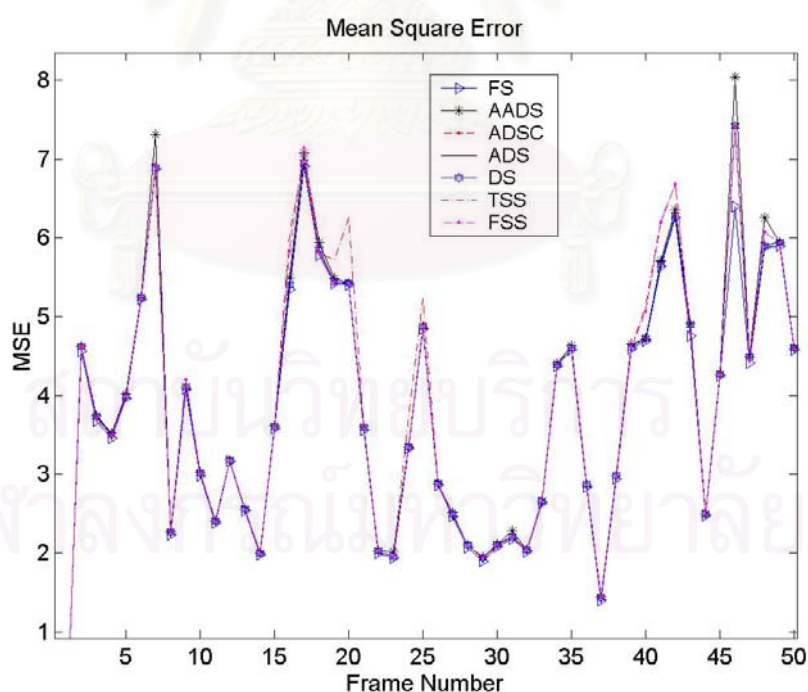
Algorithm	Average MSE	Prob.	Average Distance	Average Searchpoints	Speed up ratio
FS	3.2562	-	-	225	1.0
TSS	3.3146	0.8654	0.8906	25	9.0
FSS	3.3002	0.8581	0.9228	17.12	13.14
DS	3.2843	0.8585	0.9234	13.22	17.02
ADS	3.2908	0.8579	0.9251	12.29	18.30
ADSC	3.2904	0.8577	0.9252	12.28	18.33
AADS	3.3396	0.8482	0.9397	7.62	29.54

ตารางที่ 4.3 แสดงการเปรียบเทียบสมรรถนะของอัลกอริทึมแบบต่าง ๆ เช่นเดียวกับในตาราง ที่ 4.2 สำหรับลำดับภาพ Claire มีการเคลื่อนที่มากกว่าในลำดับภาพ Akiyo ทำให้ได้ความน่าเป็นที่ได้ เวกเตอร์การเคลื่อนที่ตรงกับเวกเตอร์การเคลื่อนที่ได้ จากอัลกอริทึมแบบทั้งหมดลดน้อยลงไป อัลกอริทึมการค้นหาแบบเพชรรณีไม่สมมาตรแบบปรับตัวได้ โดยใช้การปรับย้ายจุดศูนย์กลาง (AADS) สามารถลดจำนวนจุดการค้นหาจากวิธีการค้นหาแบบเพชรรณีมาได้ถึงร้อยละ 42 โดยมีอัตราการเพิ่มความเร็วนั้นคือ 29.54

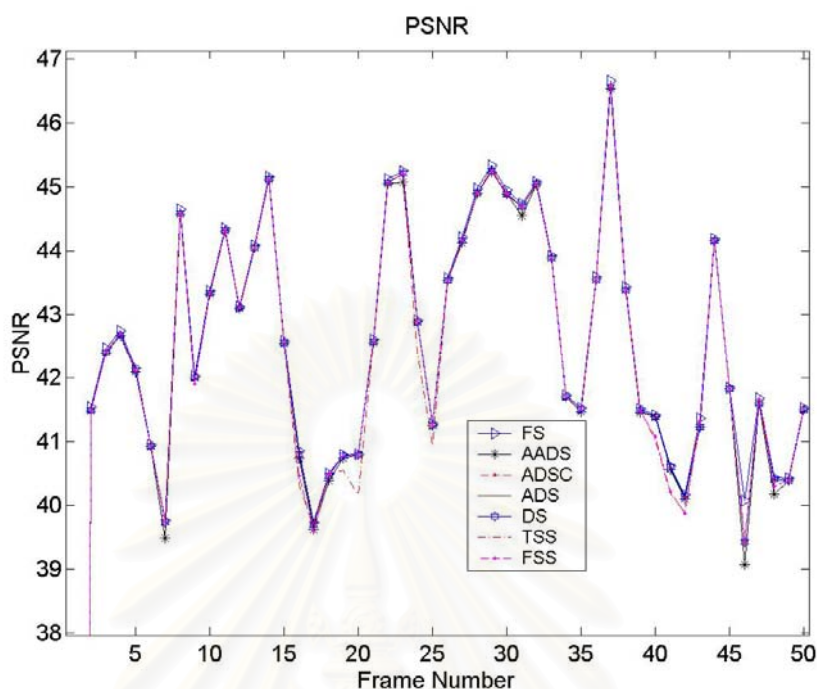
#### 4.3.1.3 ผลการทดสอบกับชุดภาพ Miss America



รูปที่ 4.5(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Miss America



รูปที่ 4.5(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Miss America



รูปที่ 4.5(ค) ค่าอัตราส่วนกำลังสัญญาณขดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Miss America

รูปที่ 4.5(ก) อัลกอริทึมการค้นหาแบบเพชรรชนิดไม่สมมาตร โดยการใช้การปรับย้ายจุดศูนย์กลาง (ADSC) เริ่มให้จำนวนจุดการค้นหาแตกต่างจากวิธีการที่ไม่ได้ปรับย้ายจุดศูนย์กลาง (ADS) แต่เป็นค่าเพียงเล็กน้อยเท่านั้น อัลกอริทึมการค้นหาแบบเพชรรชนิดไม่สมมาตร (ADS) สามารถลดจุดการค้นหาจากอัลกอริทึมการค้นหาแบบเพชรรธรรมดา (DS) ประมาณร้อยละ 8 และอัลกอริทึม ADSC ลดได้ประมาณร้อยละ 9

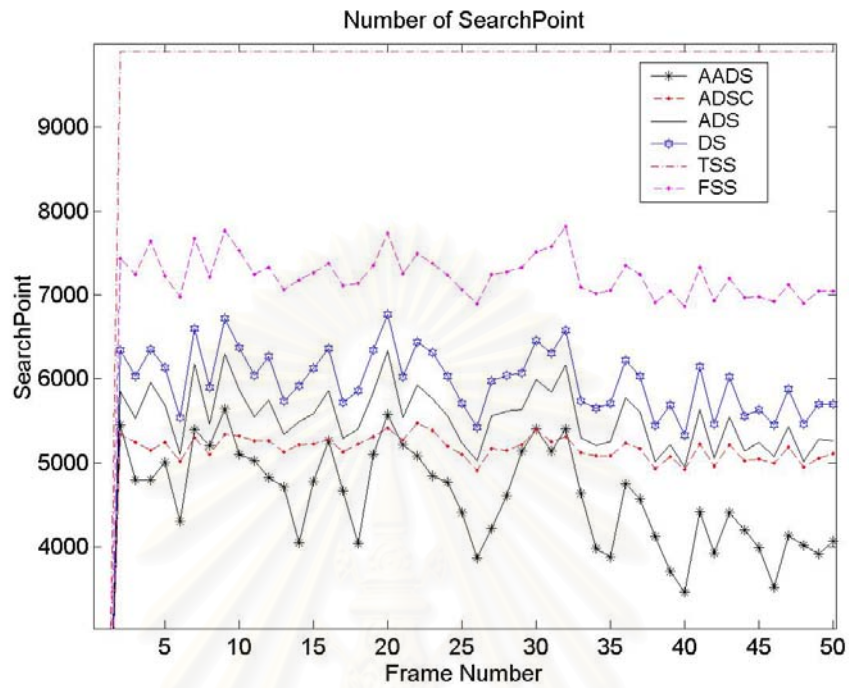
รูปที่ 4.5(ข) แสดงค่าความผิดพลาดกำลังสองเฉลี่ย (MSE) ของอัลกอริทึมชนิดต่าง ๆ จากผลการทดสอบเห็นได้ว่า ค่า MSE ของแต่ละวิธีการมีค่าใกล้เคียงกัน มีเพียงบางเฟรมที่วิธีการ AADS ให้ค่าความผิดพลาดกำลังสองเฉลี่ยสูงกว่าวิธีการอื่น ๆ เล็กน้อย สำหรับเฟรมที่ให้ความผิดพลาดสูงที่สุด คือ เฟรมที่ 46 ซึ่งมีค่า MSE เพียง 8.04 รูปที่ 4.5(ค) แสดงค่าอัตราส่วนกำลังสัญญาณขดต่อสัญญาณรบกวนเปรียบเทียบระหว่างภาพต้นแบบกับภาพที่สร้างขึ้นจากกระบวนการการประมาณการเคลื่อนที่ด้วยอัลกอริทึมต่าง ๆ เฟรมที่มีค่า PSNR น้อยกว่าอัลกอริทึมการค้นหาแบบทั้งหมดมากที่สุด คือ เฟรมที่ 46 ซึ่งมี PSNR ต่ำกว่า 0.994 db

ตารางที่ 4.4 การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Miss America

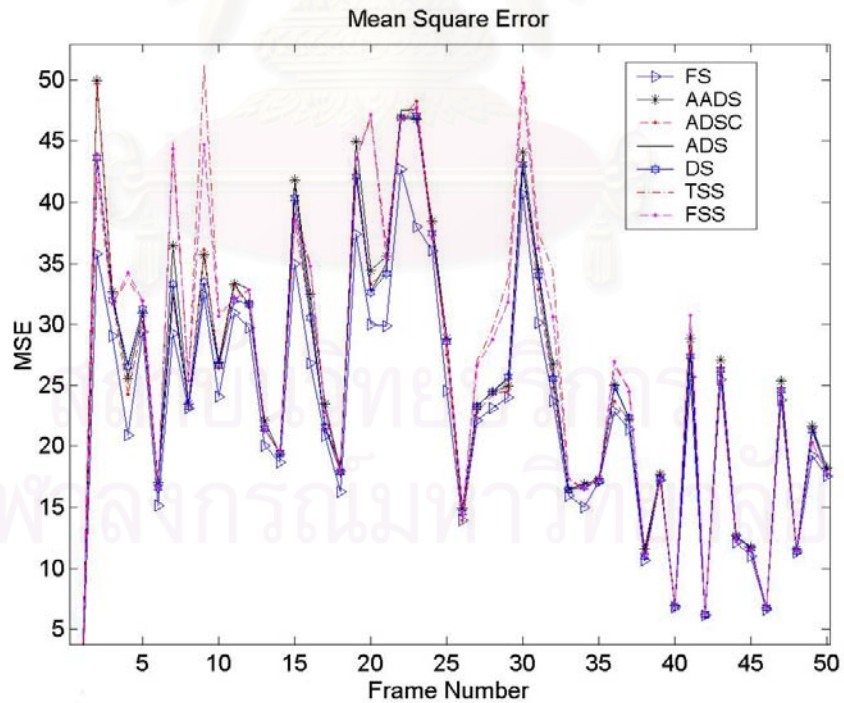
Algorithm	Average MSE	Pr ob .	Average Distance	Average Searchpoints	Speed up ratio
FS	3.8664	-	-	225	1.0
TSS	3.9967	0.7705	1.1863	25	9.0
FSS	3.9571	0.7630	1.2087	17.94	12.54
DS	3.9191	0.7643	1.2145	14.35	15.68
ADS	3.9214	0.7553	1.2337	13.22	17.02
ADSC	3.9178	0.7555	1.2350	13.10	17.17
AADS	3.9646	0.7471	1.2479	10.51	21.41

ในลำดับภาพ Miss America ความน่าจะเป็นที่ได้เวกเตอร์การเคลื่อนที่ตรงกับเวกเตอร์การเคลื่อนที่ที่ได้จากอัลกอริทึมการค้นหาแบบทั้งหมดลดลงจากลำดับภาพที่ผ่านมา เนื่องมาจากการเคลื่อนที่ที่เร็วขึ้นของวัตถุภายในภาพ อย่างไรก็ตาม อัลกอริทึม AADS ให้อัตราส่วนความเร็วที่เพิ่มขึ้นสูงถึง 21.41 ในขณะที่อัลกอริทึม DS ให้ค่าเป็น 15.68

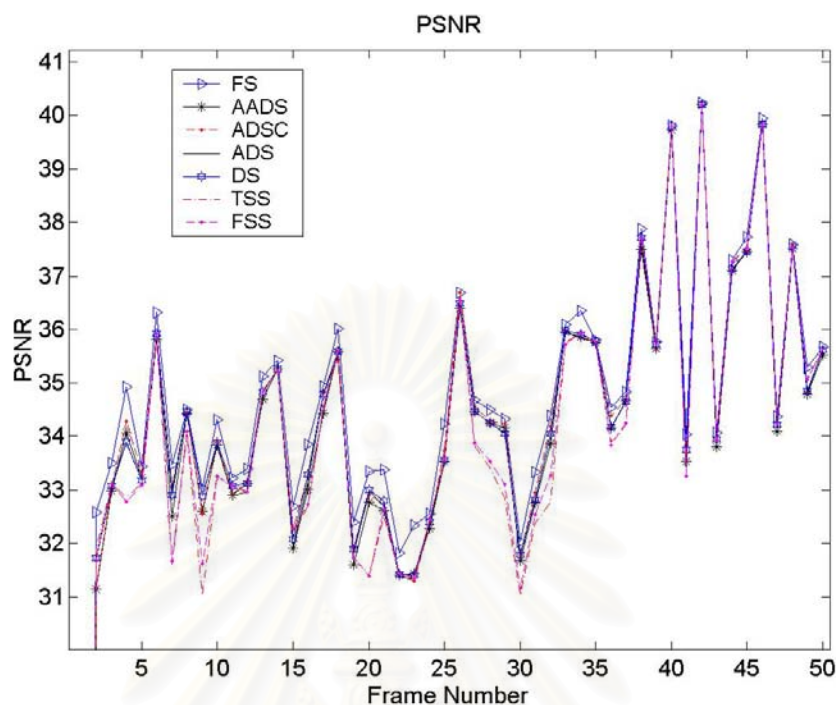
#### 4.3.1.4 ผลการทดสอบกับชุดภาพ Carphone



รูปที่ 4.6(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Carphone



รูปที่ 4.6(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Carphone



รูปที่ 4.6(ค) ค่าอัตราส่วนกำลังสัญญาณยอดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Carphone

ด้วยลักษณะการเคลื่อนที่ที่ซับซ้อนของลำดับภาพ Carphone ดังที่ได้กล่าวมาแล้วว่า มีลักษณะเป็นภาพคนเคลื่อนที่ด้วยความเร็วและภาพฉากหลังที่มีการเคลื่อนที่สวนทาง ทำให้จำนวนจุดการค้นหาที่ต้องใช้ในทุก ๆ อัลกอริทึมไม่สม่ำเสมอ ดังในรูปที่ 4.6(ก) พบว่า ในเฟรมลำดับที่ต้องใช้จำนวนจุดการค้นหามากเมื่อเปรียบเทียบกับเฟรมข้างเคียงก็จะมีรูปภาพในทำนองเดียวกันทุก ๆ อัลกอริทึม อันเป็นผลมาจาก การที่วัตถุในภาพเกิดการเปลี่ยนแปลงอย่างรวดเร็ว

พิจารณารูปที่ 4.6(ข) ช่วงเฟรมที่ 1 ถึง 25 อัลกอริทึมแบบต่าง ๆ ให้ค่าความผิดพลาดกำลังสองเฉลี่ยค่อนข้างแตกต่างกัน ต่างกับช่วงเฟรมที่ 26-50 ที่มีค่าใกล้เคียงกันมากขึ้น เพราะว่า ช่วงแรก ๆ ของลำดับภาพ ภายในรถยนต์มีการสั่นขึ้นลงมากกว่าในช่วงหลังเมื่อรถเคลื่อนตัวไปแล้ว เป็นสาเหตุสำคัญที่ทำให้การประมาณการเคลื่อนที่มีความผิดพลาดได้ง่าย เฟรมที่ 22 ให้ค่า MSE สูงที่สุด คือ 46.91 ในรูปที่ 4.6(ค) แสดงถึงค่าอัตราส่วนกำลังสัญญาณยอดต่อสัญญาณรบกวนเมื่อทดสอบกับลำดับภาพนี้ เฟรมที่มีค่า PSNR น้อยกว่าอัลกอริทึมการค้นหาแบบทั้งหมดมากที่สุด คือ เฟรมที่ 23 ซึ่งมี PSNR ต่ำกว่า 0.903 db

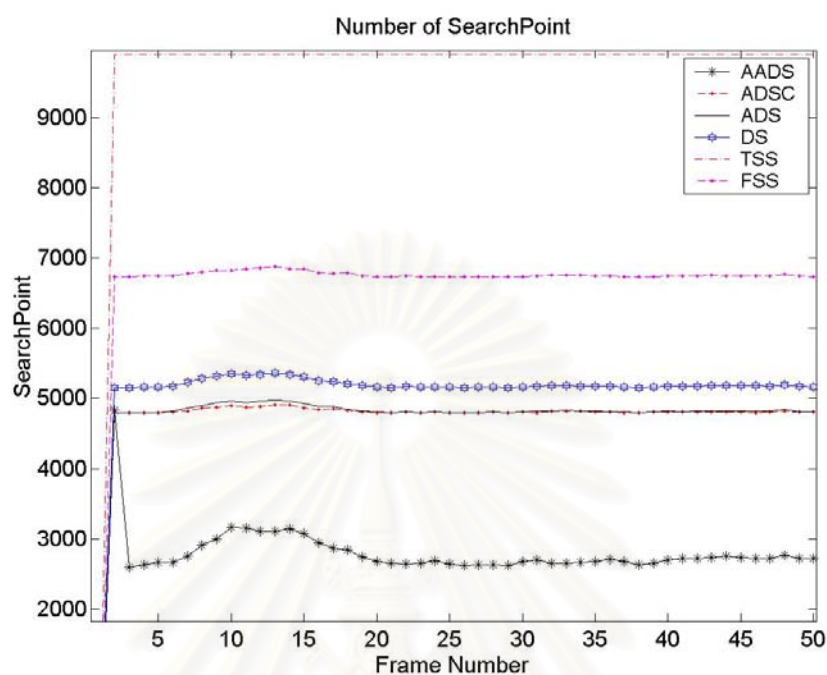
ตารางที่ 4.5 การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Carphone

Algorithm	Average MSE	Pr ob .	Average Distance	Average Searchpoints	Speed up ratio
FS	23.2540	-	-	225	1.0
TSS	27.5192	0.8172	0.8405	25	9.0
FSS	27.1974	0.8162	0.8311	18.28	12.31
DS	25.2929	0.8470	0.6856	15.12	14.88
ADS	25.5169	0.8398	0.7030	13.98	16.09
ADSC	25.3227	0.8460	0.6853	13.07	17.21
AADS	25.9876	0.8261	0.7327	11.62	19.36

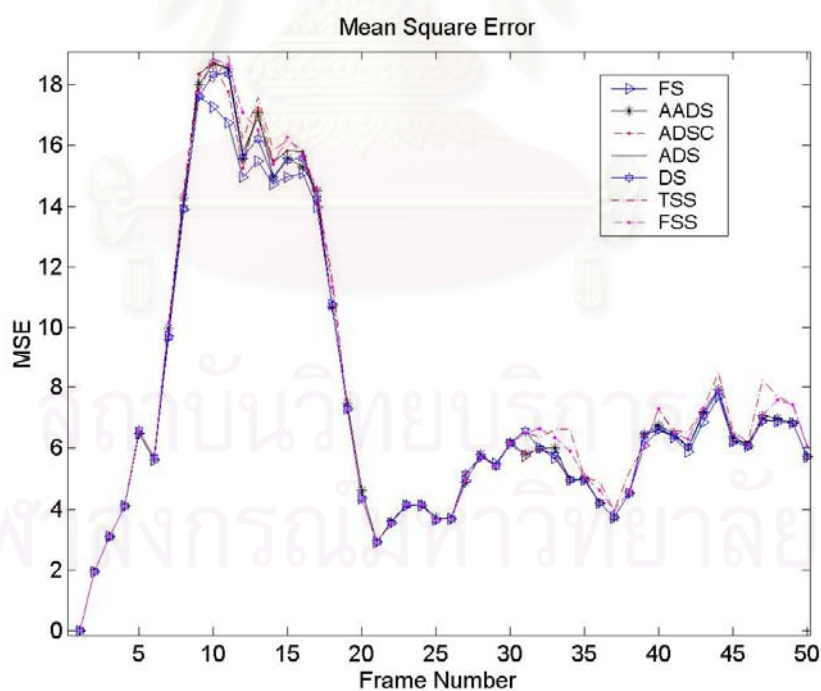
จากตารางที่ 4.5 พบว่า ค่าความผิดพลาดกำลังสองเฉลี่ยของแต่ละวิธีมีค่าไม่แตกต่างกันมากนัก แม้ว่าลำดับภาพ Carphone นี้มีการเคลื่อนที่ของวัตถุที่ซับซ้อนก็ตาม AADS ใช้อัตราความเร็วที่เพิ่มขึ้นเป็น 19.36



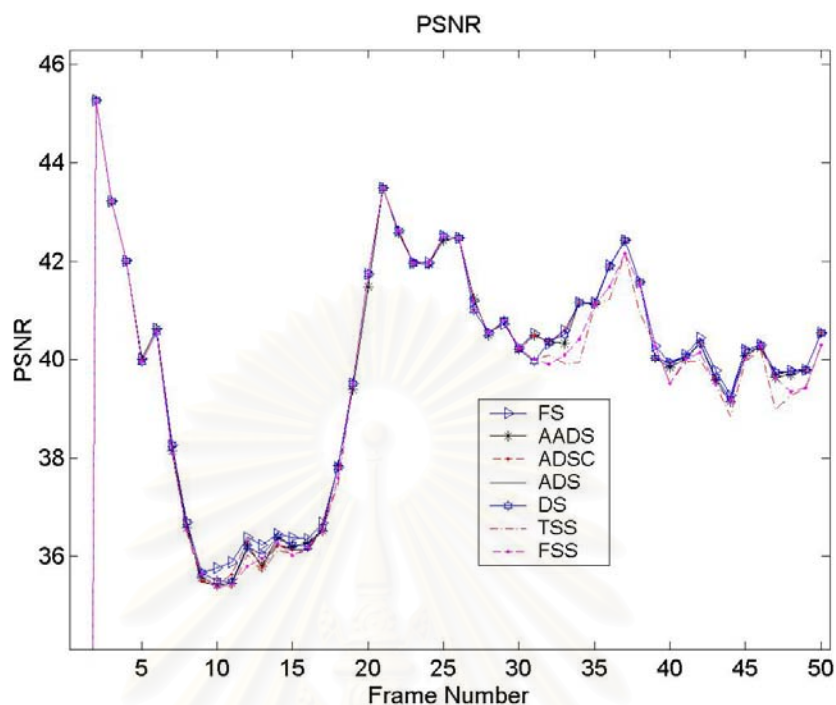
#### 4.3.1.5 ผลการทดสอบกับชุดภาพ Salesman



รูปที่ 4.7(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Salesman



รูปที่ 4.7(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Salesman



รูปที่ 4.7(ค) ค่าอัตราส่วนกำลังสัญญาณขดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Salesman

จากหลังของลำดับภาพ Salesman นี้ไม่เคลื่อนที่ มีการเคลื่อนที่มากบริเวณมือและแขน จากรูปที่ 4.7(ก) จำนวนจุดการค้นหาที่ได้จากอัลกอริทึมแต่ละวิธีจึงค่อนข้างสม่ำเสมอ ในลำดับภาพนี้ อัลกอริทึม AADS ที่นำเสนอให้ค่าความถูกต้องใกล้เคียงกับวิธีการค้นหาแบบเร็วชนิดอื่น ๆ แต่สามารถลดจำนวนจุดการค้นหาลงจากวิธีการค้นหาแบบเพชรรวมได้ถึงประมาณร้อยละ 46

รูปที่ 4.7(ข) แสดงค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมต่าง ๆ เฟรมที่ 10 ให้ค่า MSE สูงที่สุด คือ 18.68 ในรูปที่ 4.6(ค) แสดงถึงค่า PSNR โดยเฟรมที่มีค่า PSNR น้อยกว่าอัลกอริทึมการค้นหาแบบทั้งหมดมากที่สุด คือ เฟรมที่ 11 ซึ่งมี PSNR ต่ำกว่า 0.439 db

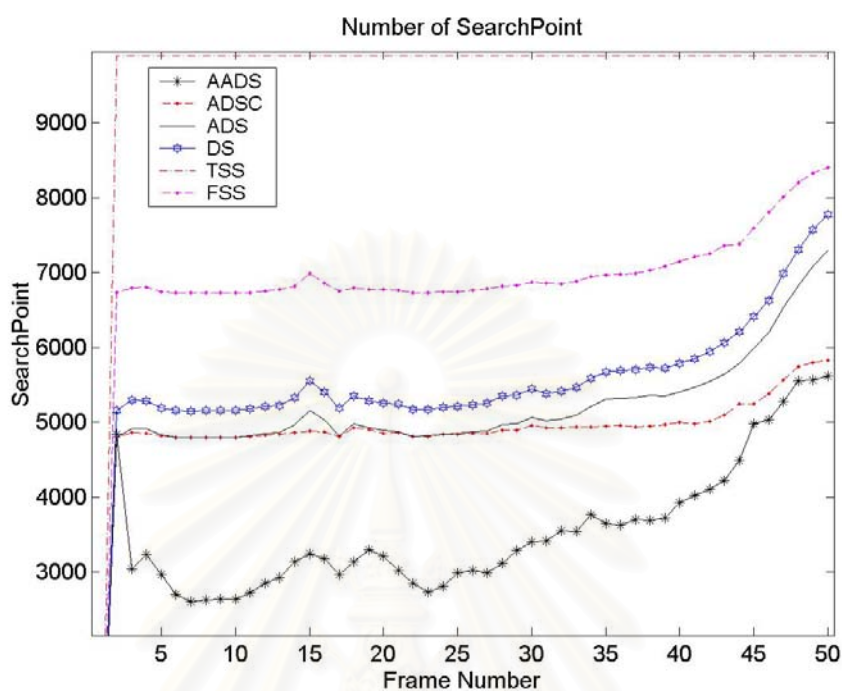
ตารางที่ 4.6 การค้นหาแบบสามขั้นตอนให้ค่าความผิดพลาดกำลังสองเฉลี่ย และระยะห่างเฉลี่ยระหว่างเวกเตอร์การเคลื่อนที่ที่แท้จริงสูงกว่าวิธีอื่นๆ เล็กน้อย อัลกอริทึม AADS มีอัตราความเร็วที่เพิ่มขึ้นเป็น 31.79

ตารางที่ 4.6 การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Salesman

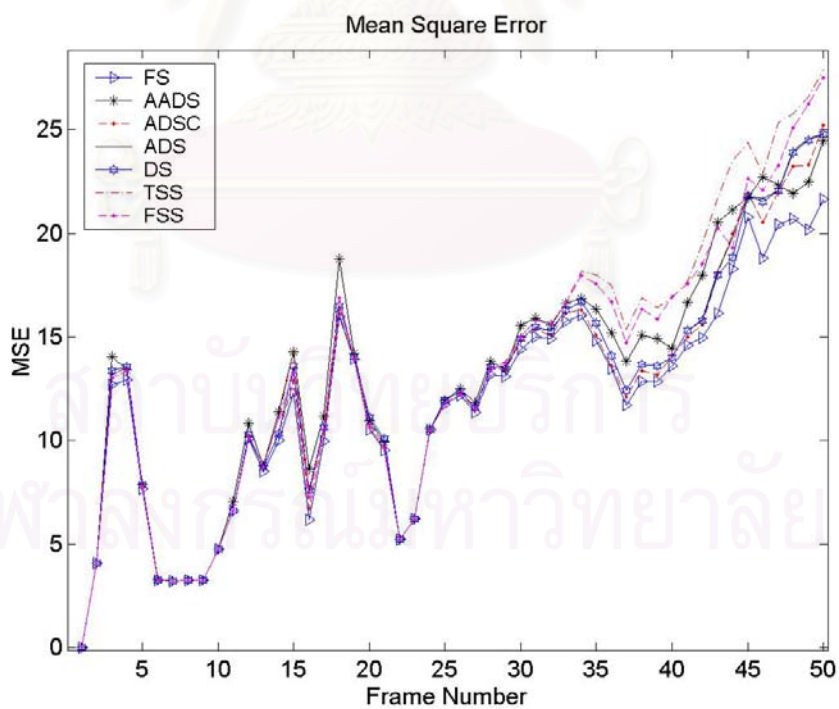
Algorithm	Average MSE	Prob.	Average Distance	Average Searchpoints	Speed up ratio
FS	7.5549	-	-	225	1.0
TSS	8.0533	0.9897	0.0241	25	9.0
FSS	7.9499	0.9911	0.0198	17.08	13.18
DS	7.7152	0.9941	0.0150	13.13	17.14
ADS	7.7840	0.9926	0.0171	12.21	18.43
ADSC	7.7428	0.9932	0.0160	12.17	18.50
AADS	7.7752	0.9919	0.0165	7.08	31.79

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

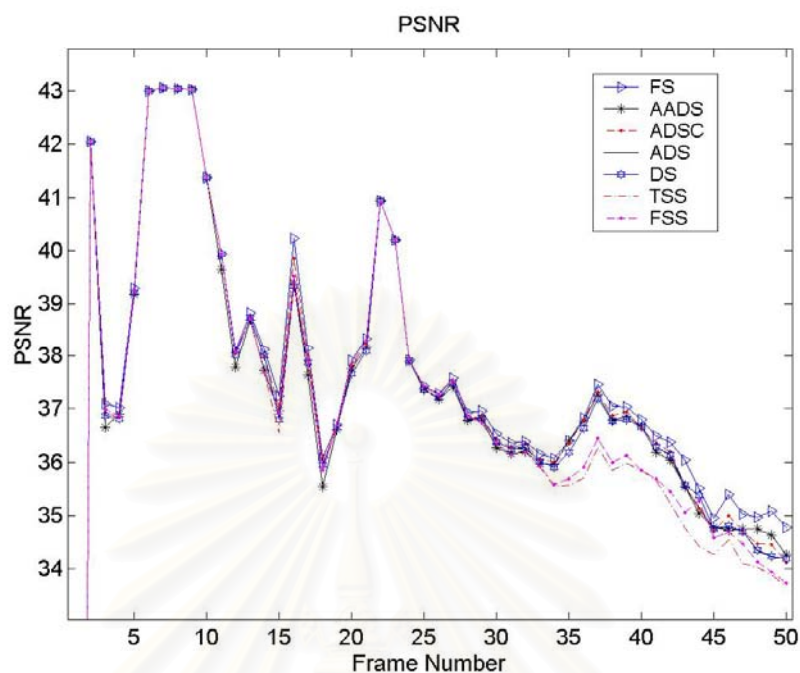
#### 4.3.1.6 ผลการทดสอบกับชุดภาพ Suzie



รูปที่ 4.8(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Suzie



รูปที่ 4.8(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Suzie



รูปที่ 4.8(ค) ค่าอัตราส่วนกำลังสัญญาณขดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Suzie

ดังในรูปที่ 4.8(ก) ช่วงเฟรมที่ 14 พบว่าต้องใช้จำนวนจุดการค้นหามากกว่าเฟรมข้างเคียง เนื่องจากหน้าคนในลำดับภาพเกิดการเคลื่อนที่อย่างรวดเร็ว สำหรับในช่วงปลายของการทดสอบช่วงตั้งแต่เฟรมที่ 35 เป็นต้นไป หน้าคนในลำดับภาพมีการเคลื่อนที่อย่างกะทันหันและต่อเนื่องอีกครั้งหนึ่ง ส่งผลให้อัลกอริทึมการค้นหาแต่ละวิธีต้องใช้จำนวนจุดการค้นหามากขึ้นตาม รูปที่ 4.8(ข) ช่วงเฟรมที่ 35 ขึ้นไปก็ให้ค่าความผิดพลาดกำลังสองเฉลี่ยมากกว่าปกติเช่นกัน ซึ่งเป็นข้อดีของการประมาณการเคลื่อนที่ด้วยอัลกอริทึมการเข้าคู่บล็อก ที่ตั้งสมมติฐานว่า การเคลื่อนที่ต้องเป็นแบบเลื่อนที่ของวัตถุแข็งแกร่ง เฟรมที่ให้ค่า MSE สูงที่สุด คือ เฟรมที่ 50 มีค่าเป็น 24.38

ในรูปที่ 4.6(ค) แสดงถึงค่า PSNR โดยเฟรมที่มีค่า PSNR น้อยกว่าอัลกอริทึมการค้นหาแบบทั้งหมดมากที่สุด คือ เฟรมที่ 16 ซึ่งมี PSNR ต่ำกว่าอยู่ 0.8449 db

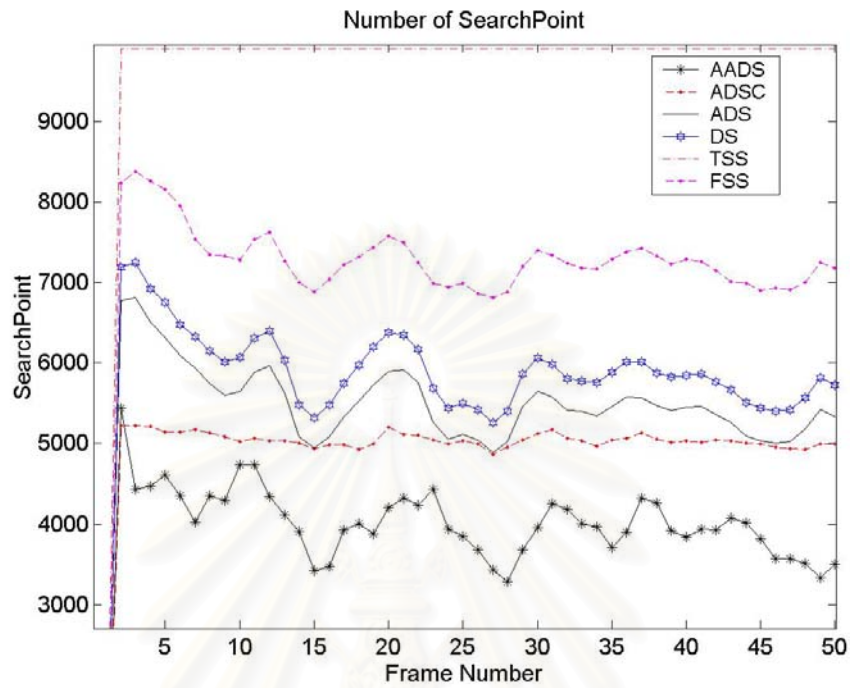
ตารางที่ 4.7 พิจารณาในแง่ของความผิดพลาดกำลังสองเฉลี่ย กระบวนการ AADS ยังให้ผลดีกว่ากระบวนการค้นหาแบบสามและสี่ขั้นตอน ในขณะที่ให้อัตราความเร็วที่เพิ่มขึ้นเป็น 25.17

ตารางที่ 4.7 การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Suzie

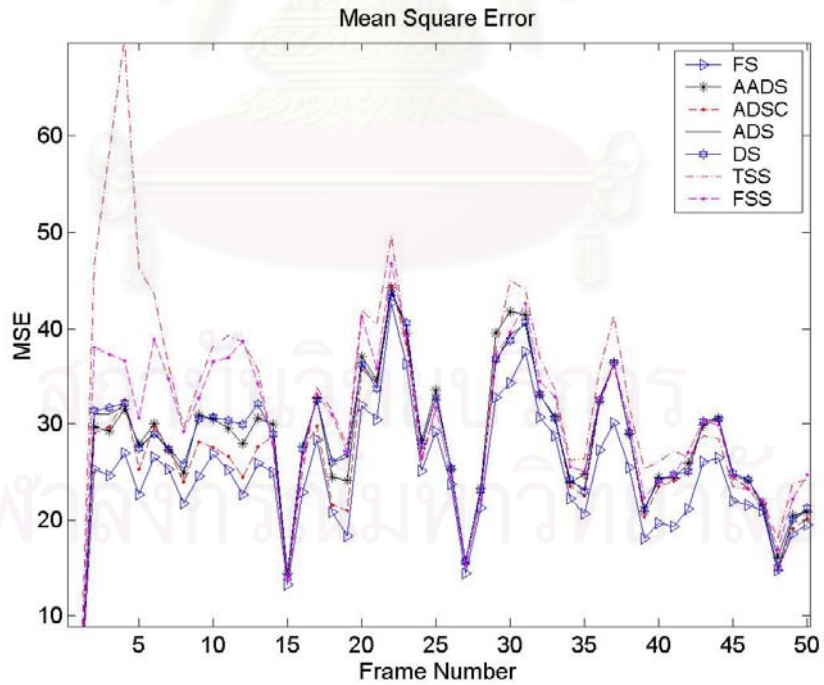
Algorithm	Average MSE	Prob.	Average Distance	Average Searchpoints	Speed up ratio
FS	12.1271	-	-	225	1.0
TSS	13.8378	0.9150	0.2200	25	9.0
FSS	13.4861	0.9244	0.1835	17.72	12.70
DS	12.8470	0.9467	0.1448	14.18	15.86
ADS	12.8771	0.9456	0.1464	13.22	17.02
ADSC	12.6791	0.9531	0.1321	12.57	17.91
AADS	12.8674	0.9428	0.1410	8.94	25.17

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

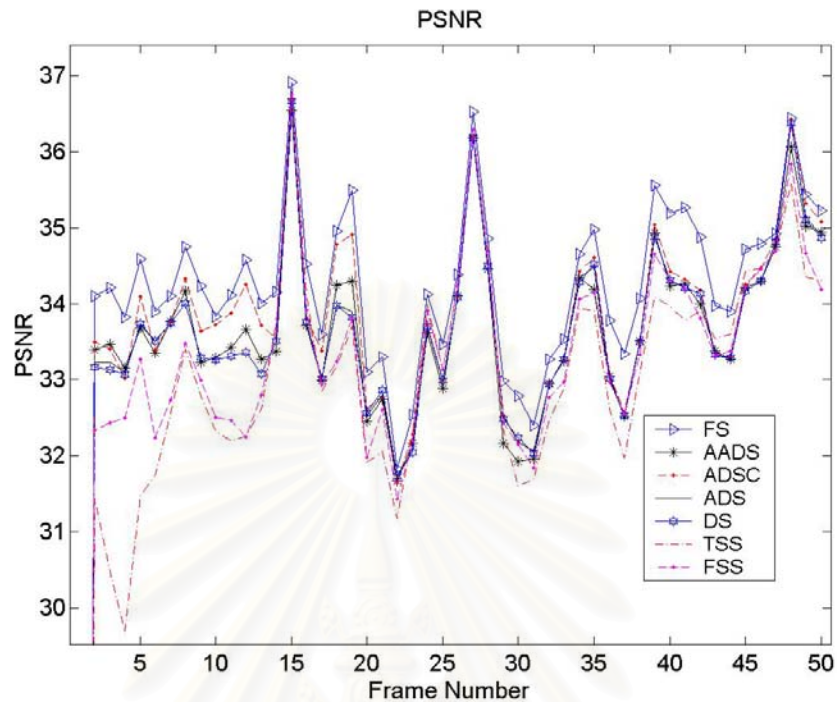
#### 4.3.1.7 ผลการทดสอบกับชุดภาพ Foreman



รูปที่ 4.9(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Foreman



รูปที่ 4.9(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Foreman



รูปที่ 4.9(ค) ค่าอัตราส่วนกำลังสัญญาณขดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Foreman

ลำดับภาพ Foreman มีการเคลื่อนที่ที่ซับซ้อน พร้อมทั้งมีการหมุนกลิ้ง ดังที่ได้กล่าวมาแล้ว ทำให้ในรูปที่ 4.9(ก) จำนวนจุดการค้นหาไม่ราบเรียบสม่ำเสมอ ค่าความผิดพลาดกำลังสองเฉลี่ยแสดงดังในรูปที่ 4.9(ข) ค่าที่ได้จากอัลกอริทึมแบบต่าง ๆ เกิดการกระจายตัว กล่าวคือ มีกราฟที่แตกต่างกันอย่างชัดเจนต่างจากลำดับภาพที่มีการเคลื่อนที่ช้าที่ผ่านมา นอกจากช่วงเฟรมที่ 20 ถึง 30 มีการหมุนกลิ้งน้อยกว่าช่วงอื่น ๆ จึงได้ค่าความผิดพลาดกำลังสองเฉลี่ยใกล้เคียงกันทุกอัลกอริทึม เฟรมที่ 22 ให้ค่า MSE สูงที่สุด คือ 44.29 ในรูปที่ 4.9(ค) แสดงถึงค่า PSNR โดยเฟรมที่มีค่า PSNR น้อยกว่าอัลกอริทึมการค้นหาแบบทั้งหมดมากที่สุด คือ เฟรมที่ 19 ซึ่งมี PSNR ต่ำกว่า 1.197 db

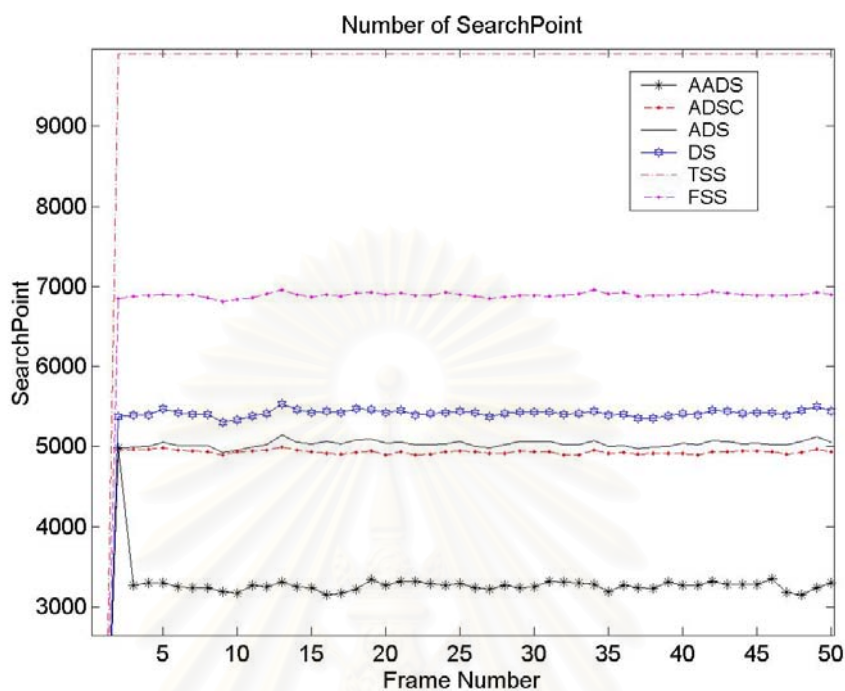


ตารางที่ 4.8 การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Foreman

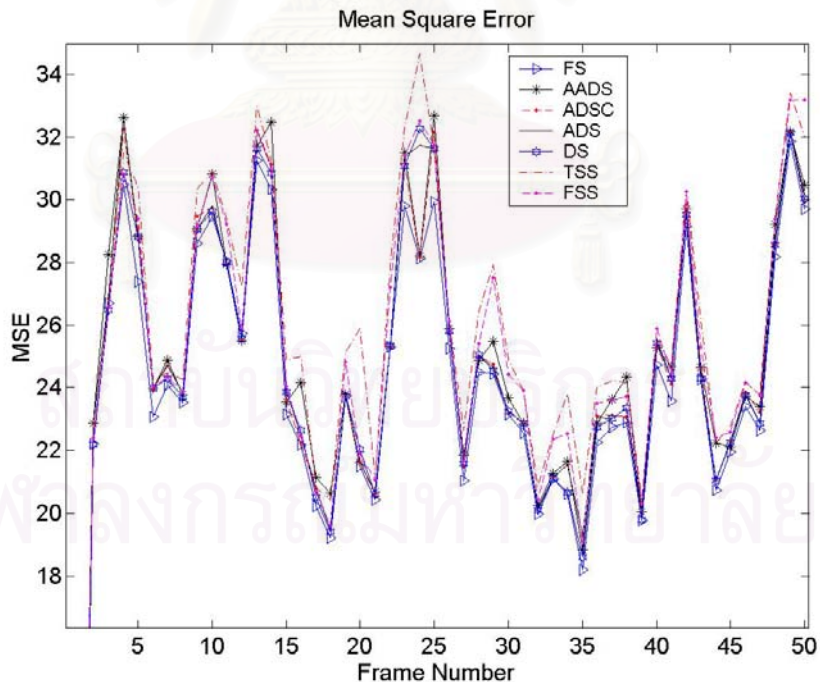
Algorithm	Average MSE	Pr ob .	Average Distance	Average Searchpoints	Speed up ratio
FS	24.8782	-	-	225	1.0
TSS	33.0779	0.7991	0.8686	25	9.0
FSS	30.4993	0.8203	0.7329	18.44	12.20
DS	28.5719	0.8547	0.6472	14.98	15.02
ADS	28.5535	0.8537	0.6434	13.94	16.14
ADSC	27.5551	0.8819	0.5343	12.74	17.66
AADS	28.6014	0.8604	0.5916	10.16	22.14

ตารางที่ 4.8 แสดงการเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ เมื่อใช้ลำดับภาพนี้เป็นตัวแทนของลำดับภาพที่มีการหมุนกลิ้ง พบว่า ค่าความผิดพลาดกำลังสองเฉลี่ยมีค่าสูงขึ้นมากเมื่อเปรียบเทียบกับลำดับภาพที่ผ่านมา อัลกอริทึม AADS ให้อัตราความเร็วที่เพิ่มขึ้นเป็น 22.14

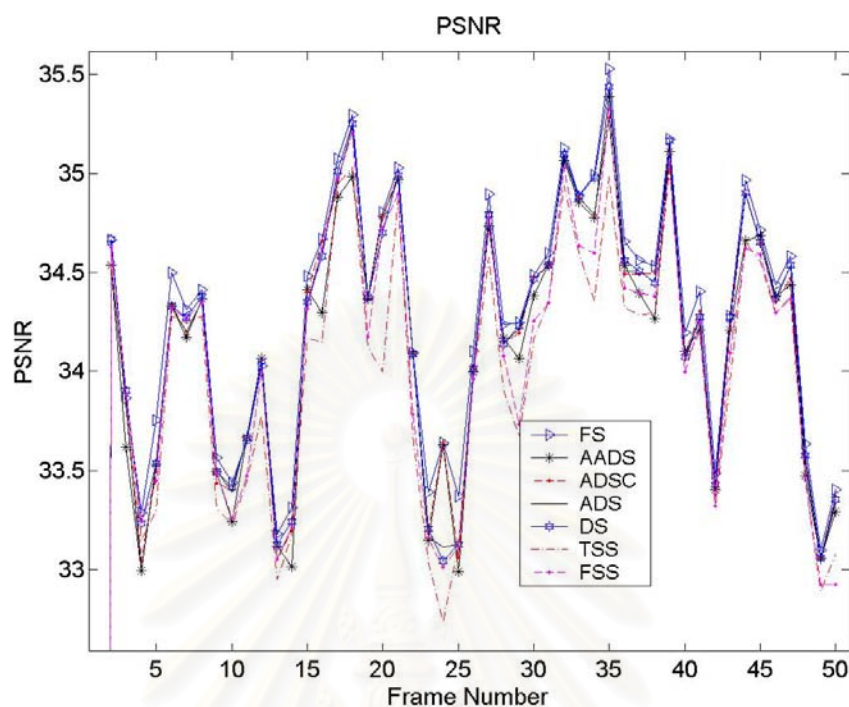
#### 4.3.1.8 ผลการทดสอบกับชุดภาพ Trevor



รูปที่ 4.10(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Trevor



รูปที่ 4.10(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Trevor



รูปที่ 4.10(ค) ค่าอัตราส่วนกำลังสัญญาณยอดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Trevor

รูปที่ 4.10(ก) แสดงจำนวนจุดการค้นหาซึ่งค่อนข้างสม่ำเสมอ เพราะการเคลื่อนที่ของแต่ละวัตถุภายในภาพไม่เร็วมากนัก รูปที่ 4.10(ข) แสดงค่า MSE ของแต่ละอัลกอริทึม พบว่า อัลกอริทึม AADS ให้ค่าความผิดพลาดกำลังสองเฉลี่ยสูงกว่าอัลกอริทึมอื่น ๆ เล็กน้อย อย่างไรก็ตาม จำนวนจุดการค้นหาที่ใช้ก็ลดลงได้มาก สำหรับเฟรมที่ให้ค่าความผิดพลาดสูงที่สุด คือ เฟรมที่ 25 ซึ่งมีค่า MSE เท่ากับ 32.67

รูปที่ 4.10(ค) แสดงถึงค่า PSNR ของแต่ละอัลกอริทึมโดยเฟรมที่มีค่า PSNR น้อยกว่าอัลกอริทึมการค้นหาแบบทั้งหมดมากที่สุด คือ เฟรมที่ 25 ซึ่งมี PSNR ต่ำกว่าอยู่ 0.379 db

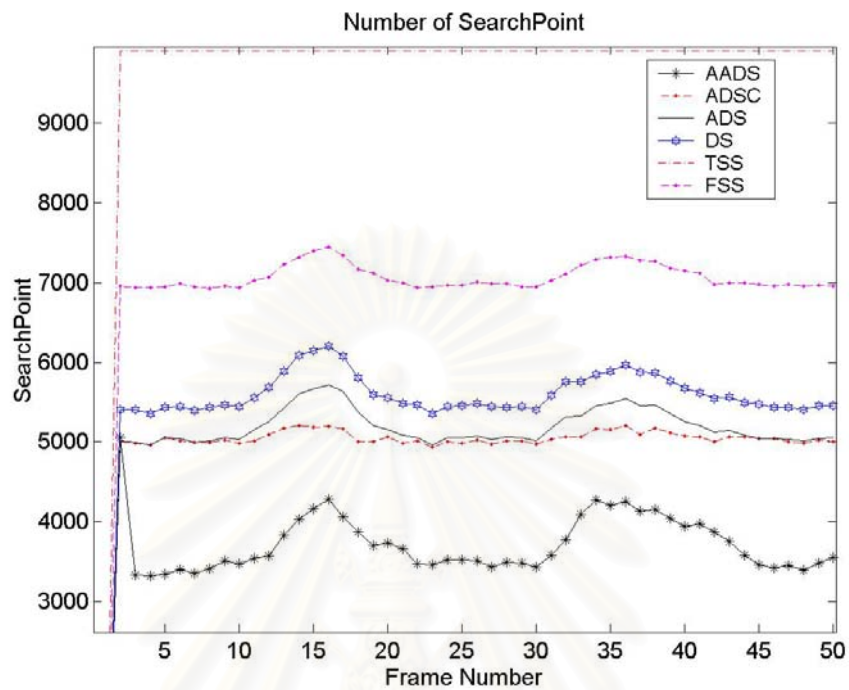
ตารางที่ 4.9 พบว่าความน่าจะเป็นในการพบเวกเตอร์การเคลื่อนที่ตรงกับเวกเตอร์การเคลื่อนที่ที่แท้จริงมีค่าใกล้เคียงกับที่ได้ในลำดับภาพที่เคลื่อนที่ช้าอื่น ๆ อย่างไรก็ตาม การที่ลำดับภาพ Trevor มีวัตถุหลายชิ้นภายในภาพเคลื่อนที่ไปในทิศทางที่แตกต่างกัน ทำให้ค่าความผิดพลาดกำลังสองเฉลี่ยสูงพอ ๆ กับลำดับภาพ Carphone ซึ่งมีการเคลื่อนที่เร็ว อัลกอริทึม AADS มีอัตราความเร็วที่เพิ่มขึ้นเป็น 27.04

ตารางที่ 4.9 การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Trevor

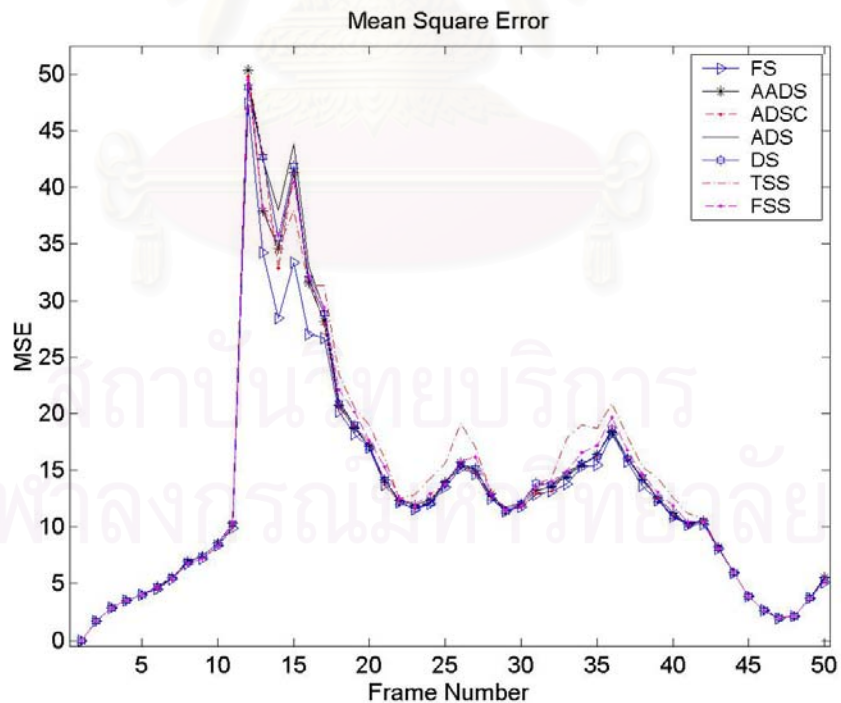
Algorithm	Average MSE	Prob.	Average Distance	Average Searchpoints	Speed up ratio
FS	24.5515	-	-	225	1.0
TSS	26.1816	0.9139	0.4710	25	9.0
FSS	25.6364	0.9160	0.4637	17.41	12.92
DS	24.9950	0.9284	0.4383	13.68	16.44
ADS	25.1470	0.9241	0.4475	12.71	17.70
ADSC	25.0618	0.9242	0.4470	12.45	18.07
AADS	25.3288	0.9230	0.4482	8.32	27.04

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

#### 4.3.1.9 ผลการทดสอบกับชุดภาพ Mother and daughter

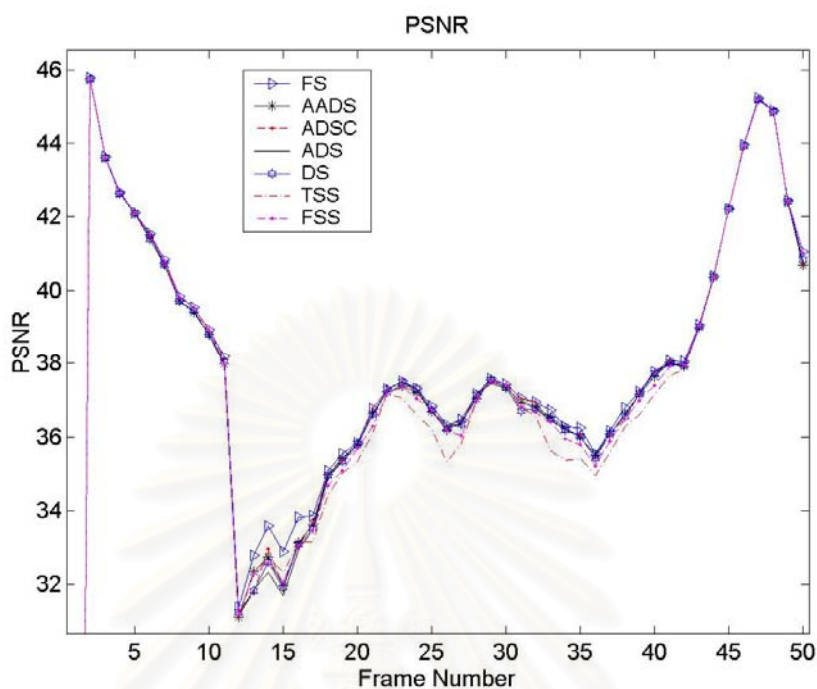


รูปที่ 4.11(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Mother



รูปที่ 4.11(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับ

ลำดับภาพ Mother



รูปที่ 4.11(ค) ค่าอัตราส่วนกำลังสัญญาณยอดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Mother

ในรูปที่ 4.11(ค) ช่วงเฟรมที่ 10 ถึง 20 ต้องใช้จุดการค้นหามากกว่าบริเวณข้างเคียง เนื่องจากในลำดับภาพมีวัตถุใหม่เกิดขึ้น กล่าวคือ มีมือปรากฏขึ้นมาอย่างรวดเร็ว และช่วงเฟรมประมาณ 30 ถึงเฟรมที่ 45 มีการหันหน้าอย่างรวดเร็ว ทำให้จำนวนจุดการค้นหามากกว่าบริเวณข้างเคียงเช่นกัน

ในรูปที่ 4.11(ข) ช่วงเฟรมที่ 10 ถึง 20 ในแต่ละอัลกอริทึมมีการกระจายตัวของค่าความผิดพลาดกำลังสองเฉลี่ยมากกว่าบริเวณข้างเคียงเช่นกัน เฟรมที่ 12 ให้ค่า MSE สูงที่สุดเท่ากับ 50.68

ค่า PSNR ของแต่ละอัลกอริทึมแสดงในรูปที่ 4.11(ค) ซึ่งกราฟที่ได้จากแต่ละอัลกอริทึมมีความใกล้เคียงกัน เฟรมที่มีค่า PSNR น้อยกว่าอัลกอริทึมการค้นหาแบบทั้งหมดมากที่สุด คือ เฟรมที่ 15 ซึ่งมี PSNR ต่ำกว่าอยู่ 0.922 db

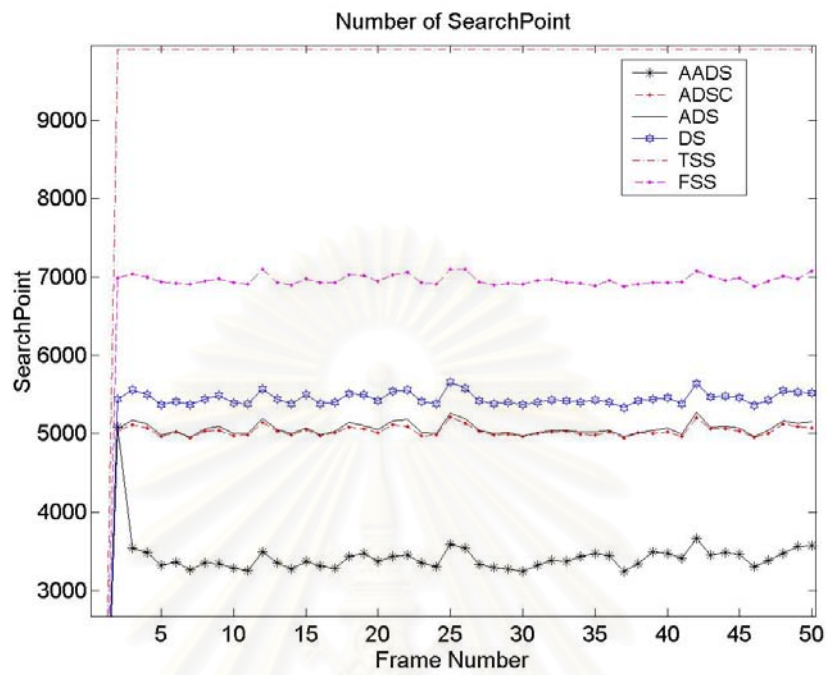
ตารางที่ 4.10 แสดงการเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ กับลำดับภาพ Mother and daughter ค่าความผิดพลาดกำลังสองเฉลี่ยที่ได้ใกล้เคียงกับลำดับภาพ Suzie แต่ค่าระยะห่างเฉลี่ยจากเวกเตอร์การเคลื่อนที่ที่แท้จริงมีค่าสูงกว่า เนื่องจากการเคลื่อนที่ของวัตถุในภาพที่ซับซ้อนกว่า อัลกอริทึม AADS มีอัตราความเร็วที่เพิ่มขึ้นเป็น 23.94

ตารางที่ 4.10 การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Mother

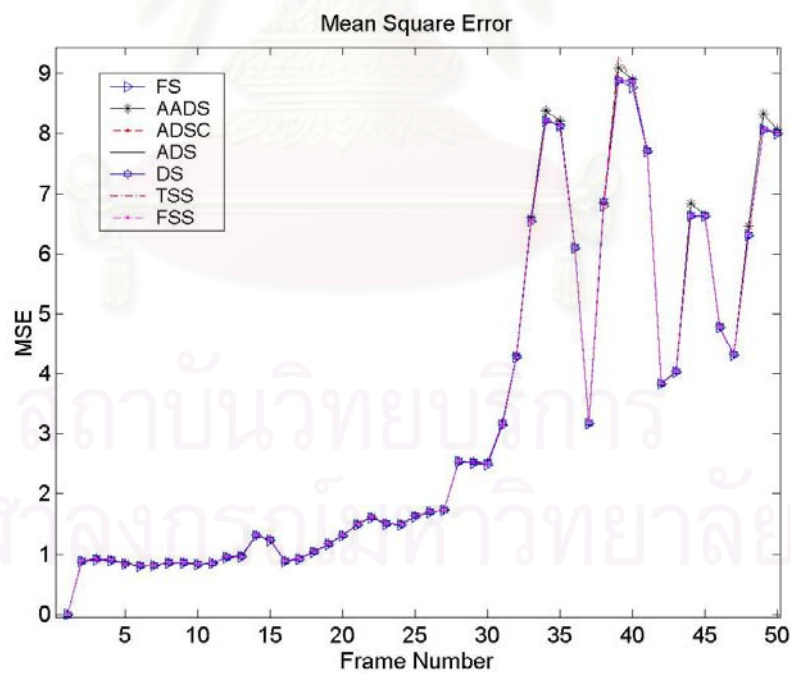
Algorithm	Average MSE	Prob.	Average Distance	Average Searchpoints	Speed up ratio
FS	12.8570	-	-	225	1.0
TSS	14.3864	0.8388	0.7950	25	9.0
FSS	13.8972	0.8422	0.7363	17.85	12.60
DS	13.7454	0.8465	0.7402	14.16	15.90
ADS	13.8966	0.8420	0.7497	13.11	17.16
ADSC	13.6444	0.8436	0.7387	12.76	17.64
AADS	13.6224	0.8409	0.7392	9.40	23.94

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

#### 4.3.1.10 ผลการทดสอบกับชุดภาพ Grandmother

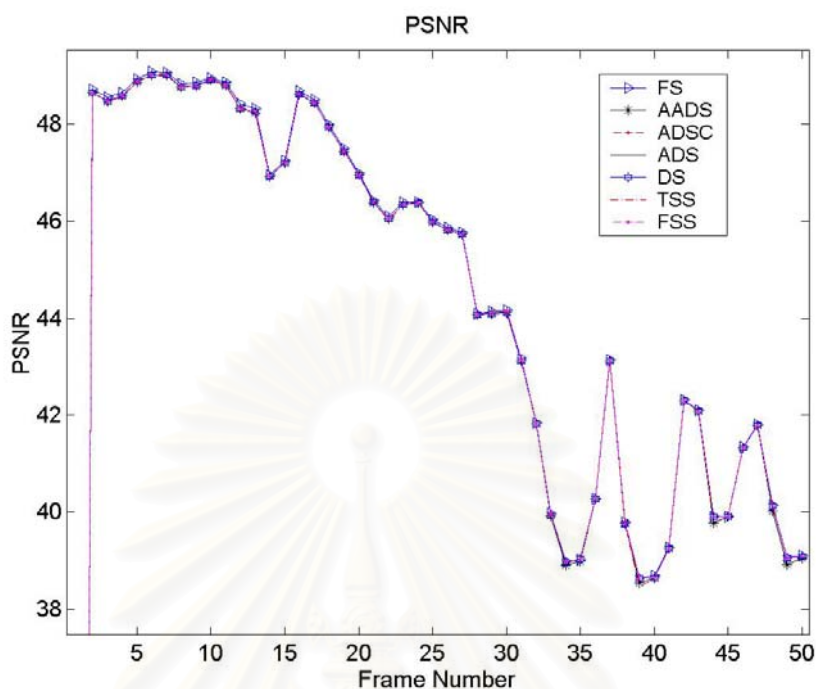


รูปที่ 4.12(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Grandmother



รูปที่ 4.12(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Grandmother





รูปที่ 4.12(ค) ค่าอัตราส่วนกำลังสัญญาณขดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Grandmother

พิจารณารูปที่ 4.12(ก) จำนวนจุดการค้นหาของแต่ละวิธีค่อนข้างสม่ำเสมอ อัลกอริทึมการค้นหาแบบ ADS กับ ADSC ให้จำนวนจุดการค้นหาไม่แตกต่างกันมากนัก เนื่องจากว่า ในลำดับภาพที่มีการเคลื่อนไหวช้า การย้ายจุดศูนย์กลางการค้นหามีโอกาสที่จะอยู่ตำแหน่งเดิมสูง การปรับย้ายจุดศูนย์กลางการค้นหาจึงไม่ส่งผลเด่นชัด รูปที่ 4.12(ข) แสดงค่า MSE ของแต่ละอัลกอริทึม ช่วงเฟรมที่ 1 ถึง 30 มีค่าความผิดพลาดกำลังสองเฉลี่ยค่อนข้างใกล้เคียงกันสำหรับทุก ๆ อัลกอริทึมและยังมีค่าสม่ำเสมอกว่าช่วงเฟรมที่ 30 เป็นต้นไป เพราะการเคลื่อนที่ของคนในช่วงท้ายมีมากกว่า และยังมีการสายศีรษะมากกว่าในช่วงต้น อัลกอริทึม AADS มีค่า MSE สูงกว่าอัลกอริทึมแบบอื่น ๆ เพียงเล็กน้อย เฟรมที่ 39 ให้ค่า MSE สูงที่สุดเท่ากับ 9.09

รูปที่ 4.12(ค) แสดงค่า PSNR ของอัลกอริทึมแต่ละวิธี ซึ่งค่าที่ได้ใกล้เคียงกัน เฟรมที่มีค่า PSNR น้อยกว่าอัลกอริทึมการค้นหาแบบทั้งหมดมากที่สุด คือ เฟรมที่ 49 ซึ่งมี PSNR ต่ำกว่าอยู่ 0.140 db

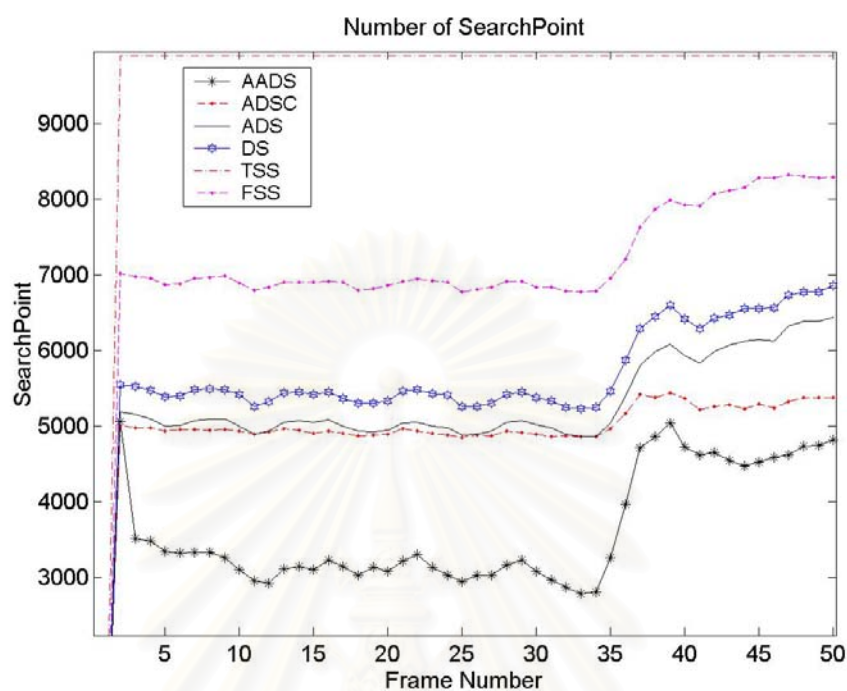
ตารางที่ 4.11 เห็นได้ว่า ค่าความผิดพลาดกำลังสองเฉลี่ยมีค่าไม่มากเท่ากับการทดสอบกับลำดับภาพที่มีการเคลื่อนที่สูง อัลกอริทึม AADS มีอัตราความเร็วที่เพิ่มขึ้นเป็น 25.93

ตารางที่ 4.11 การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Grandmother

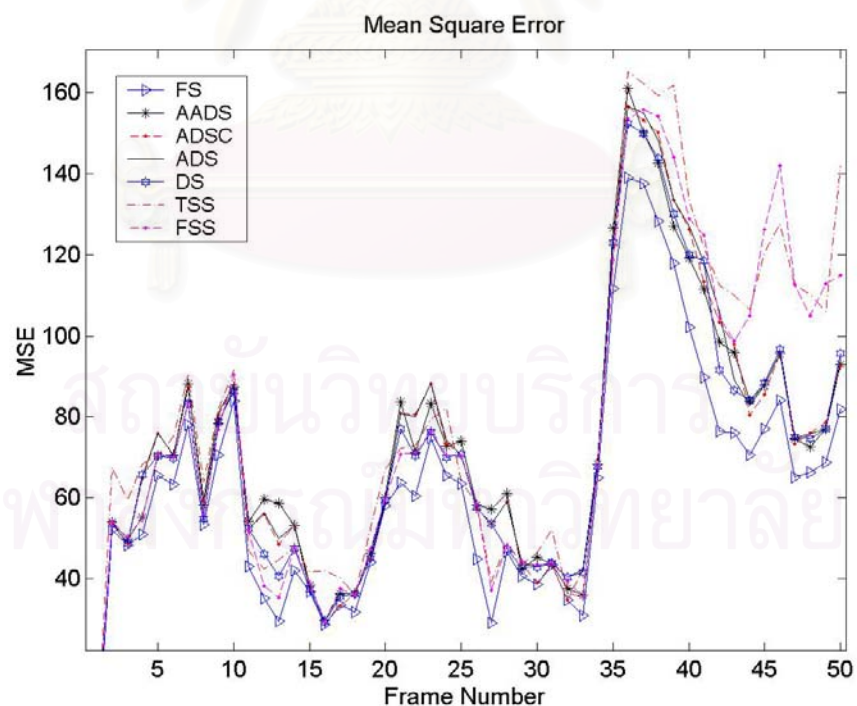
Algorithm	Average MSE	Prob.	Average Distance	Average Searchpoints	Speed up ratio
FS	3.2900	-	-	225	1.0
TSS	3.3120	0.8543	0.7377	25	9.0
FSS	3.3044	0.8508	0.7447	17.59	12.79
DS	3.3068	0.8507	0.7461	13.77	16.34
ADS	3.3072	0.8474	0.7526	12.80	17.58
ADSC	3.3064	0.8472	0.7573	12.72	17.70
AADS	3.3296	0.8447	0.7593	8.68	25.93

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

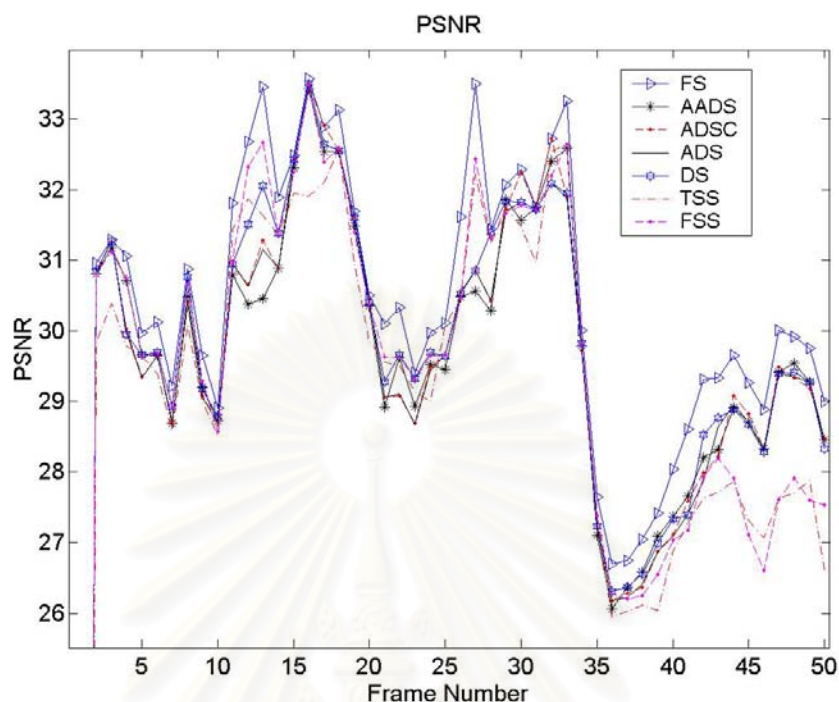
#### 4.3.1.11 ผลการทดสอบกับชุดภาพ Table tennis



รูปที่ 4.13(ก) จำนวนจุดการค้นหาของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Table tennis



รูปที่ 4.13(ข) ค่าความผิดพลาดกำลังสองเฉลี่ยของอัลกอริทึมชนิดต่าง ๆ เมื่อทดสอบกับลำดับภาพ Table tennis



รูปที่ 4.13(ค) ค่าอัตราส่วนกำลังสัญญาณขดต่อสัญญาณรบกวนของอัลกอริทึมชนิดต่าง ๆ  
เมื่อทดสอบกับลำดับภาพ Table tennis

สำหรับในลำดับภาพ Table tennis ดังที่ได้กล่าวมาแล้วว่า วัตถุภายในเฟรมมีการเคลื่อนที่อย่างรวดเร็ว ประกอบกับการซูมกล้องในการถ่ายทำด้วย ซึ่งเห็นได้ชัดในช่วงเฟรมที่ 35 เป็นต้นมา ในรูปที่ 4.13(ค) ช่วงเฟรมที่ 35 จึงเริ่มใช้จำนวนจุดการค้นหามากกว่าปกติ พิจารณารูปที่ 4.13(ข) ซึ่งแสดงกราฟของ MSE ในแต่ละอัลกอริทึม ช่วงที่มีการซูมกล้องดังกล่าว เห็นได้ชัดว่า อัลกอริทึมการค้นหาแบบเร็วต่าง ๆ นั้นให้ผลดีน้อยกว่าอัลกอริทึมการค้นหาแบบทั้งหมดอย่างชัดเจน นอกจากนี้ อัลกอริทึมการค้นหาแบบสามขั้นตอนและสี่ขั้นตอนยังให้ค่าความผิดพลาดกำลังสองเฉลี่ยสูงกว่าอัลกอริทึมอื่น ๆ ที่ทดสอบด้วย อัลกอริทึมแบบเพชรชนิดไม่สมมาตรแบบต่าง ๆ ยังให้ MSE ใกล้เคียงกับ DS อีกด้วย สำหรับเฟรมที่ให้ค่า MSE สูงที่สุด คือ เฟรมที่ 36 ซึ่งมีค่าเท่ากับ 161.05

รูปที่ 4.13(ค) แสดงค่า PSNR ของแต่ละอัลกอริทึม เฟรมที่มีค่า PSNR น้อยกว่าอัลกอริทึมการค้นหาแบบทั้งหมดมากที่สุด คือ เฟรมที่ 13 ซึ่งมี PSNR ต่ำกว่าอยู่ 2.99 db

ตารางที่ 4.12 การเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ ของลำดับภาพ Table

Algorithm	Average MSE	Prob.	Average Distance	Average Searchpoints	Speed up ratio
FS	64.0622	-	-	225	1.0
TSS	81.5568	0.8153	0.9215	25	9.0
FSS	77.6466	0.8542	0.6786	18.29	12.30
DS	72.3526	0.9160	0.4306	14.48	15.53
ADS	74.9374	0.9119	0.4409	13.47	16.71
ADSC	74.5298	0.9179	0.4225	12.74	17.67
AADS	74.1246	0.9167	0.4076	9.18	24.50

จากตารางที่ 4.12 แสดงการเปรียบเทียบสมรรถนะของอัลกอริทึมต่าง ๆ พบว่า ลำดับภาพนี้ให้ค่าความผิดพลาดกำลังสองเฉลี่ยสูงกว่าทุก ๆ ลำดับภาพที่ทดสอบมา เมื่อให้ลำดับภาพ Table tennis เป็นตัวแทนของลำดับภาพที่มีการเคลื่อนที่เร็วและมีการซูมกล้อง กล่าวได้ว่า อัลกอริทึมการเข้าคู่บล็อกมีข้อด้อยที่ให้สมรรถนะไม่ดีเมื่อใช้กับลำดับภาพที่มีการซูมกล้อง

### 4.3.2 ตัวอย่างภาพที่สร้างขึ้นจากอัลกอริทึมต่าง ๆ

หัวข้อนี้แสดงตัวอย่างภาพที่สร้างขึ้นด้วยอัลกอริทึมแบบต่าง ๆ ที่นำเสนอไป ยกเว้นภาพจากอัลกอริทึม ADS และ ADSC เพราะผลที่ได้ใกล้เคียงกับอัลกอริทึม DS ลำดับภาพที่เลือกมาใช้ในการทดสอบในหัวข้อนี้ ได้แก่ ลำดับภาพ Akiyo และ Carphone ซึ่งเป็นตัวแทนของลำดับภาพที่มีการเคลื่อนที่ช้าและเร็ว ตามลำดับ

#### 4.3.2.1 ตัวอย่างภาพที่สร้างขึ้นของลำดับภาพ Akiyo



(ก)



(ข)



(ค)



(ง)



(จ)

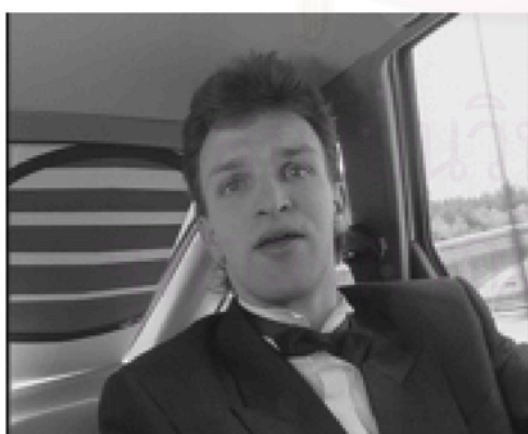


(ข)

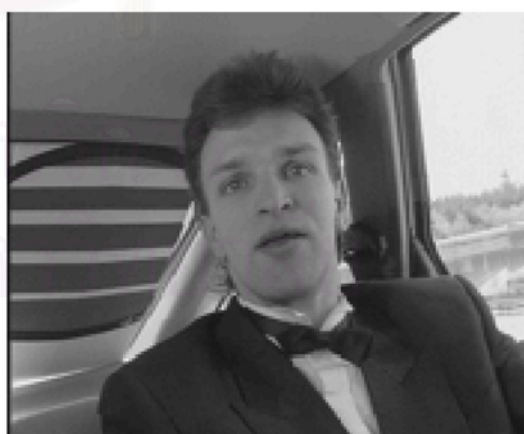
รูปที่ 4.14 (ก) ภาพต้นแบบเฟรมที่ 36 (ข) ตัวอย่างภาพที่สร้างขึ้นจากอัลกอริทึม FS (ค) TSS (ง) FSS (จ) DS (ฉ) AADS

รูปที่ 4.14 แสดงภาพต้นแบบและภาพที่สร้างขึ้นจากอัลกอริทึมต่าง ๆ การทดสอบเลือกใช้ลำดับภาพเฟรมที่ 36 ซึ่งเป็นเฟรมที่อัลกอริทึม AADS ให้ค่า MSE สูงที่สุด คือ 8.66 ผลที่ได้จากทุกอัลกอริทึมมีความใกล้เคียงกัน เนื่องจากวัตถุในลำดับภาพ Akiyo เคลื่อนไหวช้าและไม่ซับซ้อน

#### 4.3.2.2 ตัวอย่างภาพที่สร้างขึ้นของลำดับภาพ Carphone



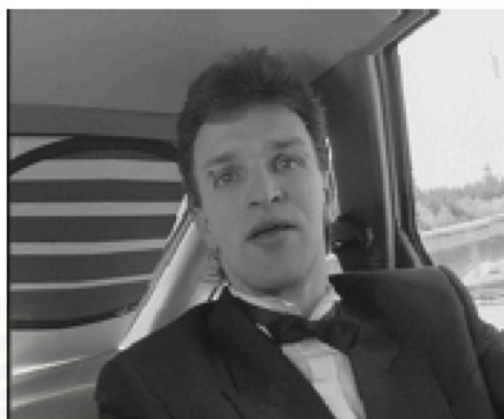
(ก)



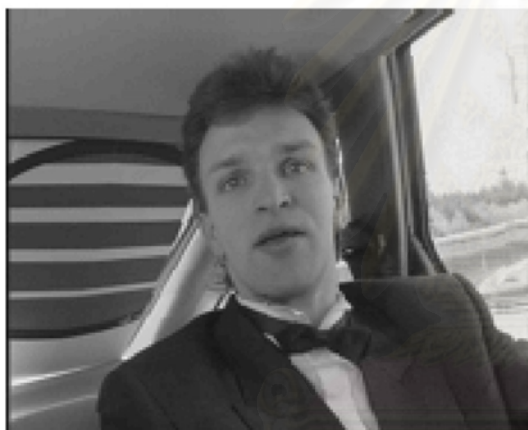
(ข)



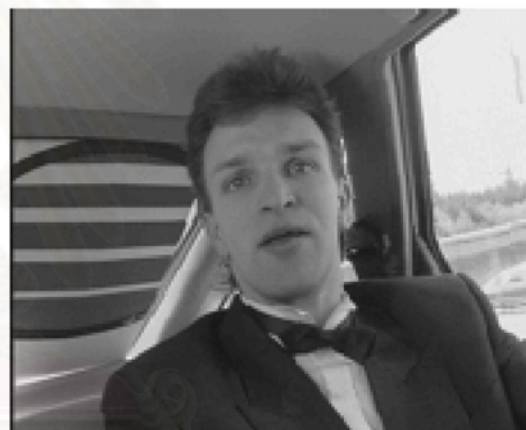
(ก)



(ง)



(จ)



(ฉ)

รูปที่ 4.15 (ก) ภาพต้นแบบเฟรมที่ 21 (ข) ตัวอย่างภาพที่สร้างขึ้นจากอัลกอริทึม FS (ค) TSS (ง) FSS (จ) DS (ฉ) AADS

ในลำดับภาพ Carphone เฟรมที่ให้ค่า MSE สูงที่สุด คือ เฟรมที่ 21 ซึ่งให้ค่าเป็น 29.86 ดังแสดงในรูปที่ 4.15 โดยภาพของเสาที่อยู่บริเวณมุมบนขวาของภาพต้นแบบ เมื่อใช้อัลกอริทึมต่าง ๆ สร้างขึ้นแล้วเกิดความผิดพลาดที่สังเกตเห็นได้ เนื่องจากเสาดังกล่าวเป็นวัตถุใหม่ที่เกิดขึ้นในเฟรมภาพ ทำให้การประมาณการเคลื่อนที่เกิดความผิดพลาดขึ้น



### 4.3.3 สนามการเคลื่อนที่ของลำดับภาพ

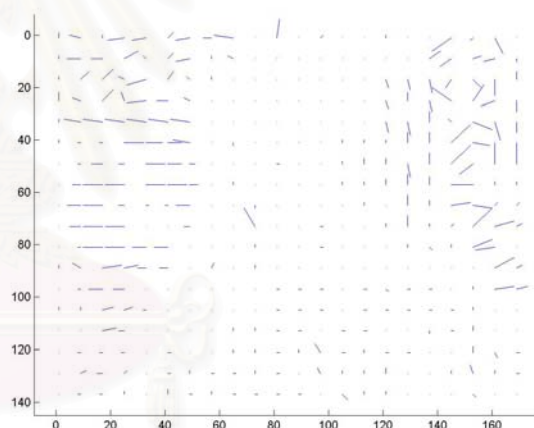
หัวข้อนี้ทำการทดสอบสนามการเคลื่อนที่ของลำดับภาพต่าง ๆ โดยเปรียบเทียบระหว่าง เวกเตอร์การเคลื่อนที่ที่ได้จากอัลกอริทึมชนิดต่าง ๆ ลำดับภาพที่เลือกใช้ในการทดสอบ คือ ลำดับภาพ Carphone ซึ่งมีวัตถุในภาพเคลื่อนที่เร็ว และฉากหลังมีการเลื่อนตำแหน่ง ลำดับภาพ Table tennis ซึ่งมีวัตถุในภาพเคลื่อนที่เร็ว และมีการซูมกล้อง และลำดับภาพ Suzie ซึ่งมีวัตถุในภาพเคลื่อนที่ไม่เร็วเท่าสองลำดับภาพแรก แต่มีการหมุนศีรษะ

#### 4.3.3.1 การทดสอบกับลำดับภาพ Carphone

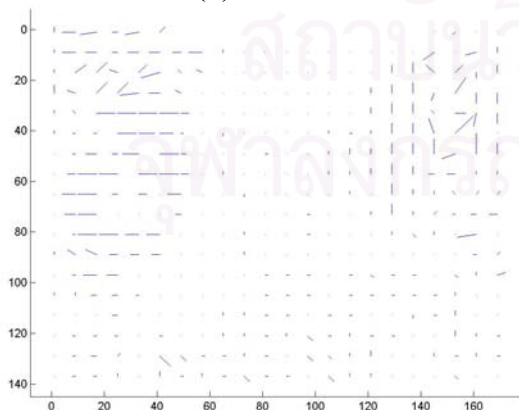
การทดสอบใช้เวกเตอร์การเคลื่อนที่ของลำดับภาพ Carphone เฟรมที่ 23 ผลการทดสอบกับอัลกอริทึมชนิดต่าง ๆ มีดังนี้



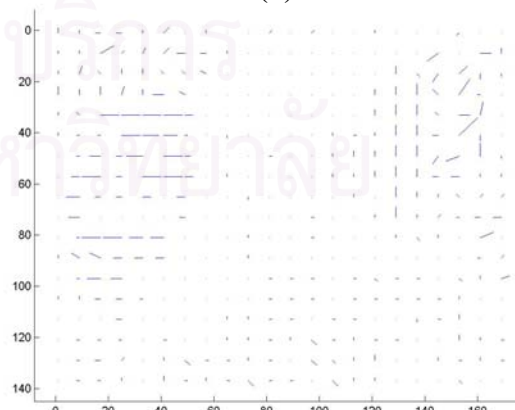
(ก) ภาพเฟรมที่ 23



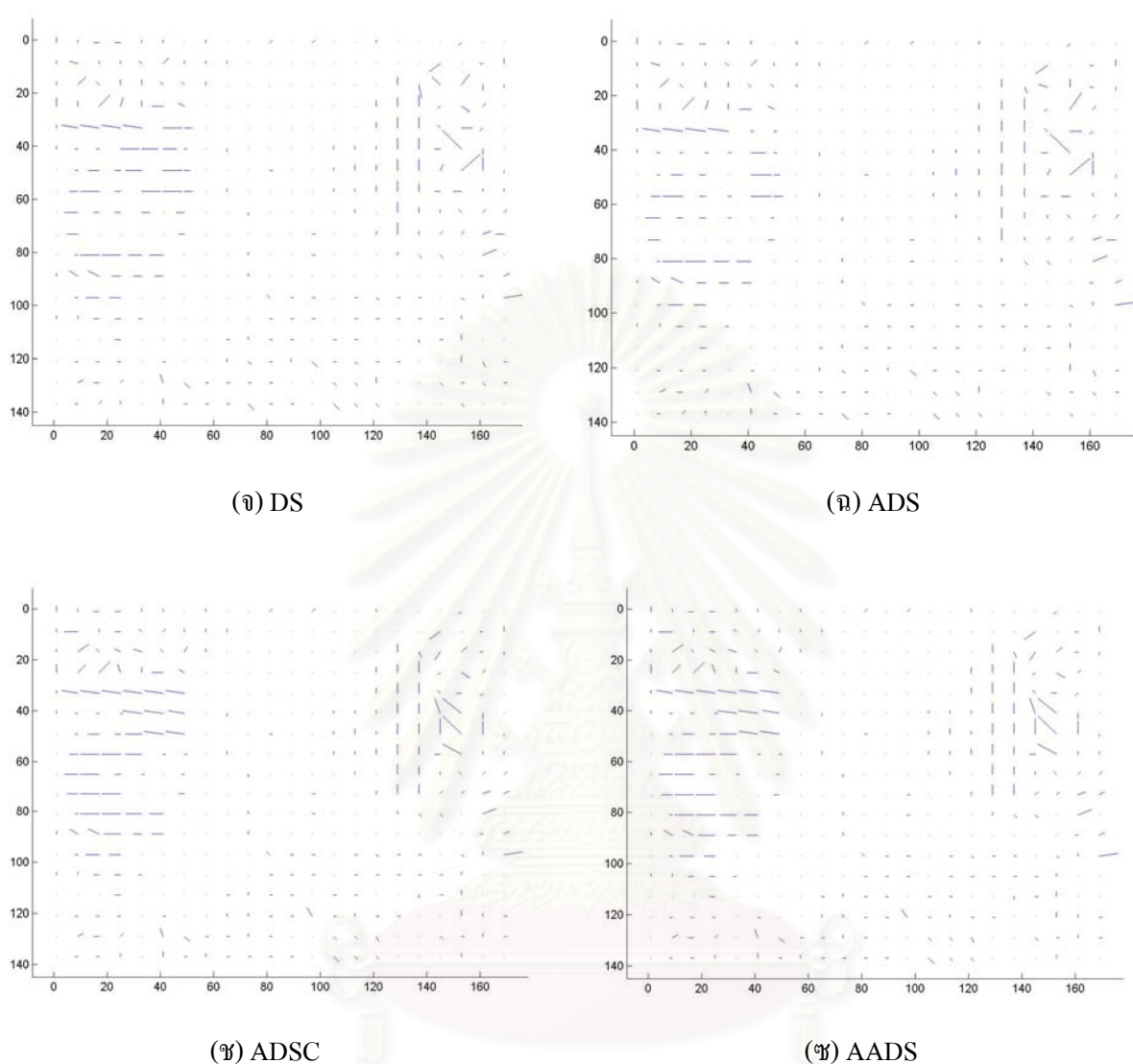
(ข) FS



(ค) TSS

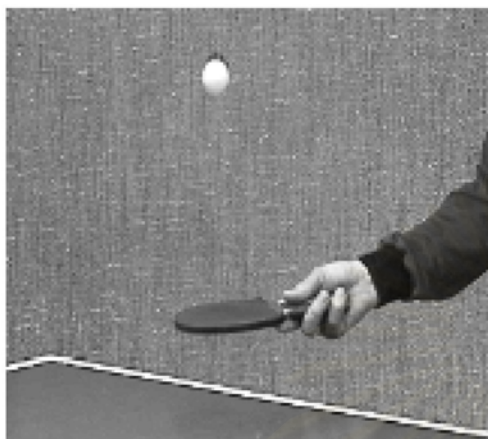


(ง) FSS

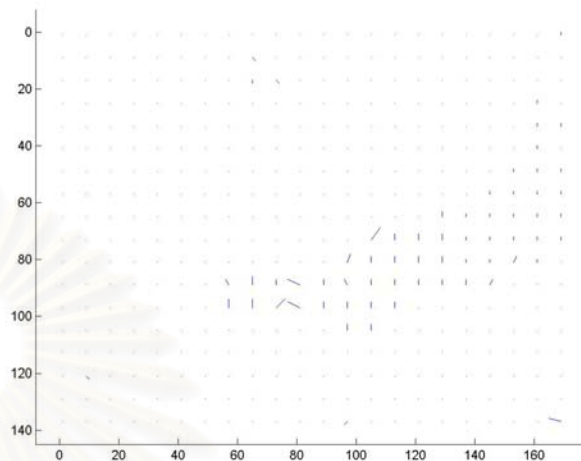


รูปที่ 4.16 (ก) การตรวจสอบสนามการเคลื่อนที่ของลำดับภาพ “Carphone” (ก) ภาพต้นแบบเฟรมที่ 23 สนามการเคลื่อนที่จากอัลกอริทึม (ข) การค้นหาแบบทั้งหมด (ค) การค้นหาแบบสามขั้นตอน (ง) การค้นหาแบบสี่ขั้นตอน (จ) การค้นหาแบบเพชร (ฉ) การค้นหาแบบเพชรไม่สมมาตร (ช) การค้นหาแบบเพชรไม่สมมาตรแบบย้ายจุดศูนย์กลาง (ซ) การค้นหาแบบเพชรไม่สมมาตรแบบปรับตัวได้

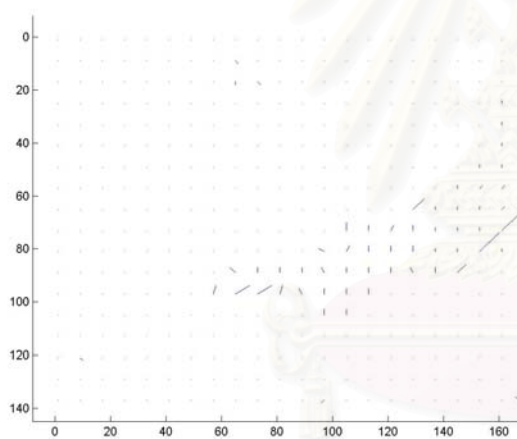
### 4.3.3.2 การทดสอบกับลำดับภาพ Table tennis



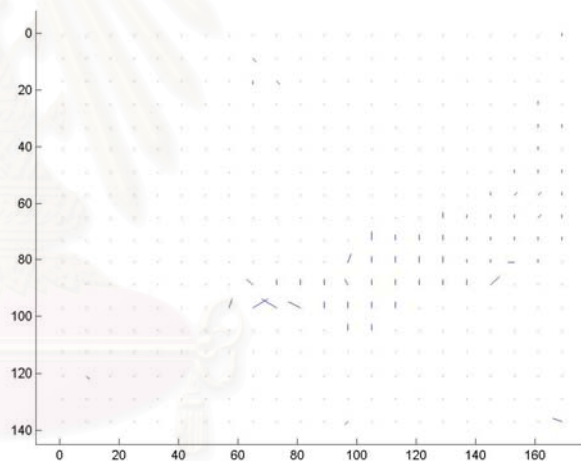
(ก) ภาพเฟรมที่ 16



(ข) FS

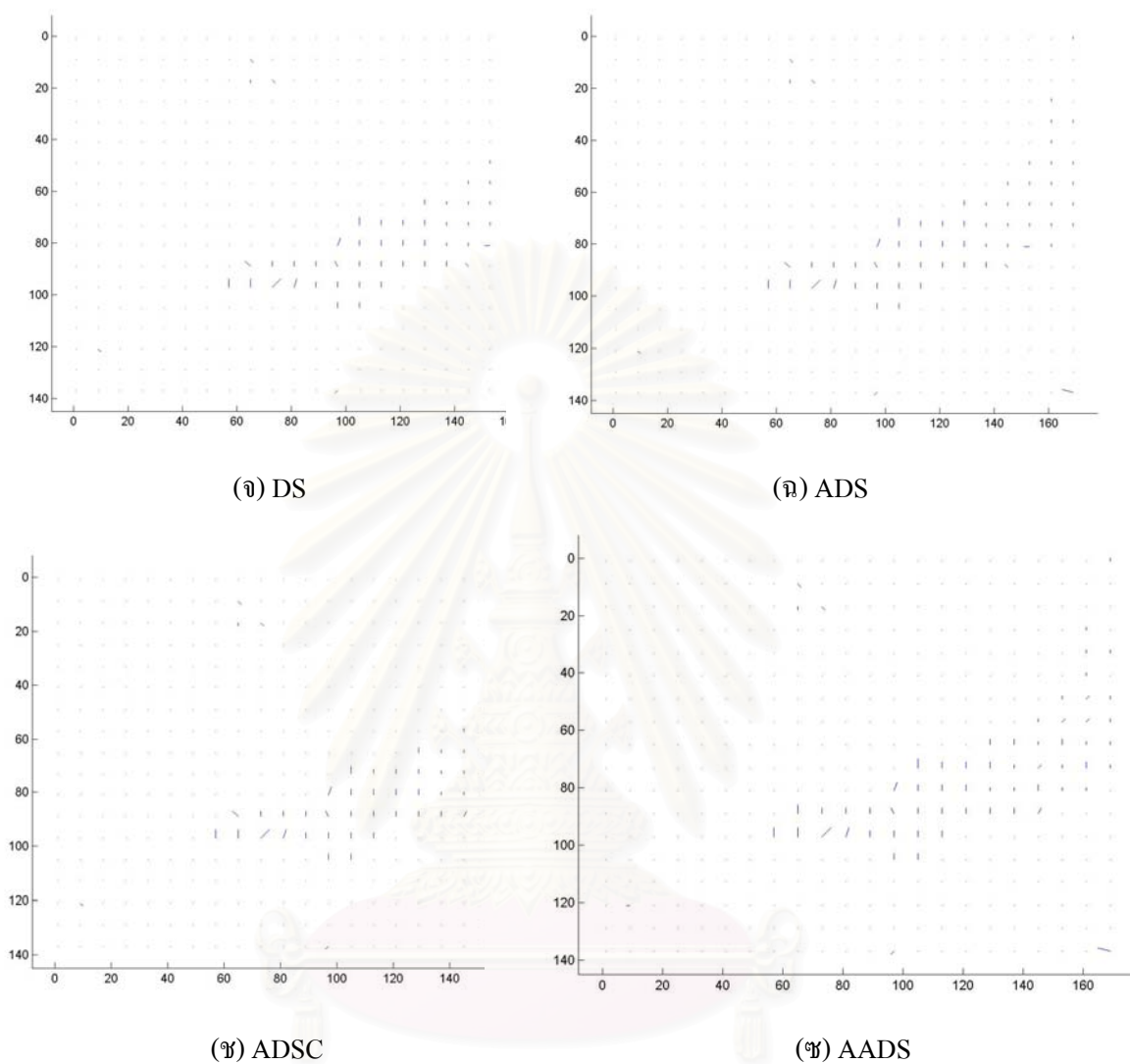


(ค) TSS



(ง) FSS

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

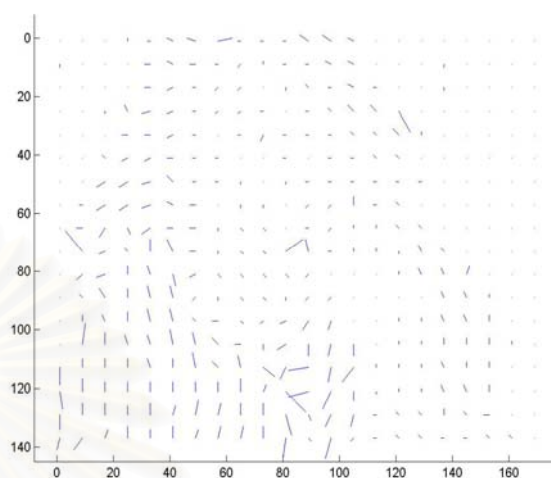


รูปที่ 4.17 (ก) การตรวจสอบสนามการเคลื่อนที่ของลำดับภาพ “Table tennis” (ก) ภาพต้นแบบเฟรมที่ 23 สนามการเคลื่อนที่จากอัลกอริทึม (ข) การค้นหาแบบทั้งหมด (ค) การค้นหาแบบสามขั้นตอน (ง) การค้นหาแบบสี่ขั้นตอน (จ) การค้นหาแบบเพชร (ฉ) การค้นหาแบบเพชรไม่สมมาตร (ช) การค้นหาแบบเพชรไม่สมมาตรแบบย้ายจุดศูนย์กลาง (ซ) การค้นหาแบบเพชรไม่สมมาตรแบบปรับตัวได้

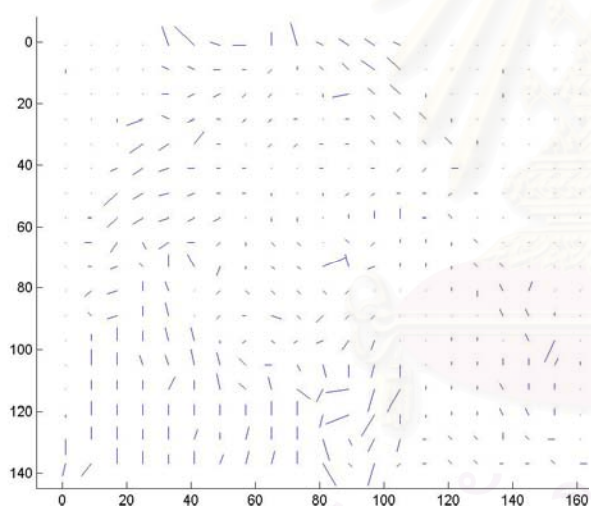
## 4.3.3.3 การทดสอบกับลำดับภาพ Suzie



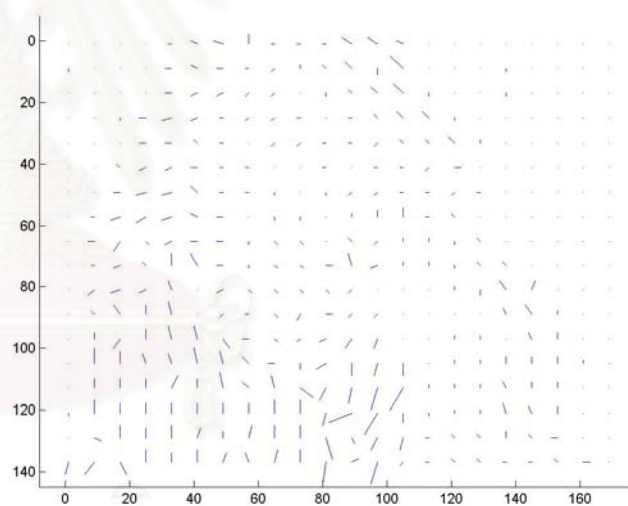
(ก) ภาพเฟรมที่ 46



(ข) FS

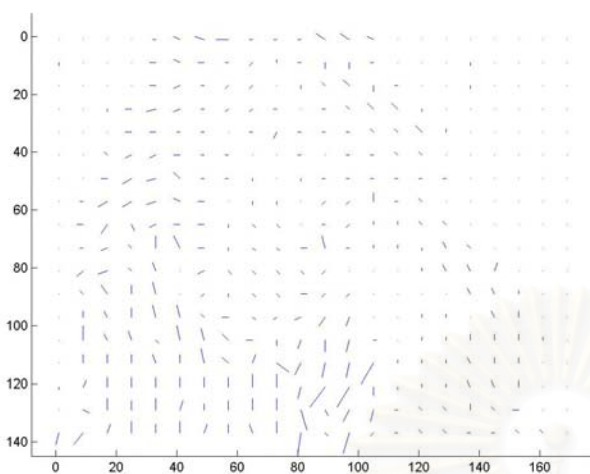


(ค) TSS

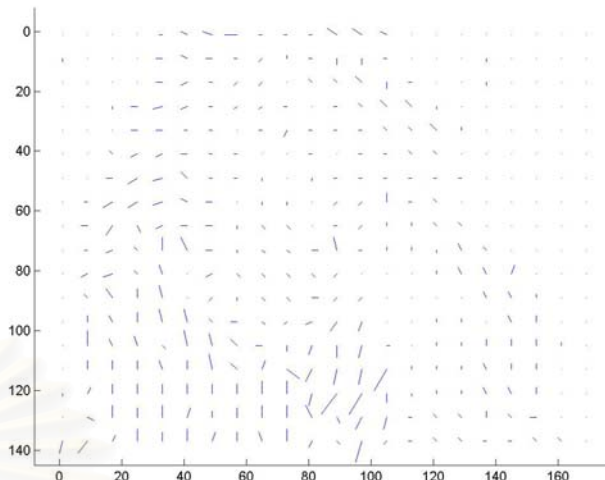


(ง) FSS

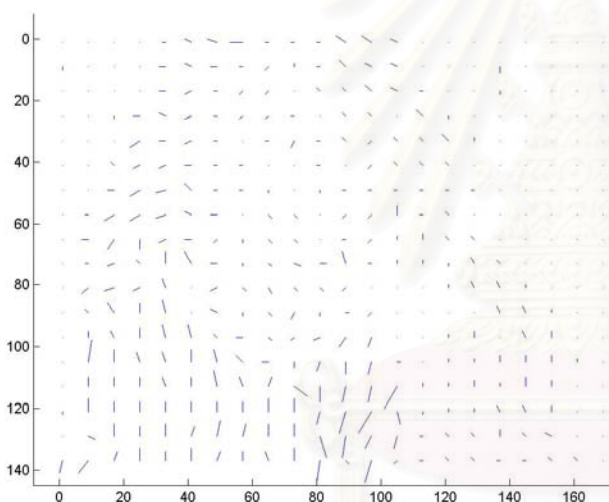
สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



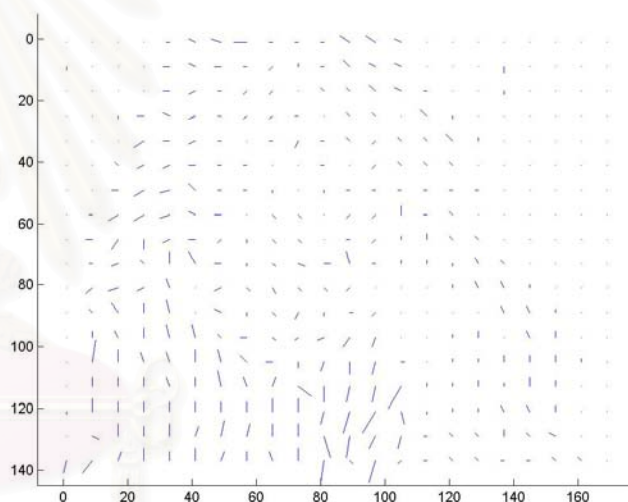
(จ) DS



(ฉ) ADS



(ช) ADSC



(ฅ) AADS

รูปที่ 4.18 (ก) การทดสอบสนามการเคลื่อนที่ของลำดับภาพ “Suzie” (ก) ภาพต้นแบบเฟรมที่ 46 สนามการเคลื่อนที่จากอัลกอริทึม (ข) การค้นหาแบบทั้งหมด (ค) การค้นหาแบบสามขั้นตอน (ง) การค้นหาแบบสี่ขั้นตอน (จ) การค้นหาแบบเพชร (ฉ) การค้นหาแบบเพชรไม่สมมาตร (ช) การค้นหาแบบเพชรไม่สมมาตรแบบย้ายจุดศูนย์กลาง (ฅ) การค้นหาแบบเพชรไม่สมมาตรแบบปรับตัวได้

จากผลการทดสอบทั้งสามลำดับภาพ สรุปได้ว่า อัลกอริทึมการค้นหาแบบทั้งหมดให้เวกเตอร์การเคลื่อนที่ในลักษณะกระจายตัวมากกว่าอัลกอริทึมชนิดอื่น ๆ แต่อย่างไรก็ดี ขนาดของเวกเตอร์การเคลื่อนที่ที่ได้ส่วนใหญ่มีขนาดมากกว่าอัลกอริทึมชนิดอื่น ๆ เนื่องจากอัลกอริทึมการค้นหาแบบทั้งหมดค้นหาทั่วทั้งหน้าตาของการค้นหา จุดต่ำสุดที่ได้จึงเป็นจุดต่ำสุดรวม ไม่ใช่เป็นจุดต่ำสุดท้องถิ่น เหมือนกับอัลกอริทึมการค้นหาอย่างรวดเร็วอื่น ๆ



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## บทที่ 5

### สรุปผลการวิจัยและข้อเสนอแนะ

#### 5.1 สรุปผลการวิจัย

วิทยานิพนธ์ฉบับนี้นำเสนออัลกอริทึมการเข้าคู่บล็อกด้วยเทคนิคการค้นหาแบบเพชชชนิดไม่สมมาตรแบบปรับตัวได้โดยใช้การปรับย้ายจุดศูนย์กลาง เพื่อมุ่งเน้นในการลดความซับซ้อนในกระบวนการประมาณการเคลื่อนที่ในระบบการเข้ารหัสสัญญาณดิจิทัล โดยยังคงรักษาความถูกต้องในการคำนวณเวกเตอร์การเคลื่อนที่ไว้ให้อยู่ในระดับที่ยอมรับได้ เทคนิคที่นำเสนอนี้เป็นการรวม 2 แนวความคิดเข้าด้วยกัน ได้แก่ เทคนิคการปรับรูปแบบ การค้นหาโดยอาศัยการคาดคะเนขนาดของเวกเตอร์การเคลื่อนที่ และเทคนิคการปรับย้ายจุดศูนย์กลางในการเริ่มต้นค้นหา

ผลการจำลองระบบในส่วนของการทดสอบแนวความคิดว่า การทำนายและการตัดจุดการค้นหาที่พัฒนาขึ้นมีความผิดพลาดมากน้อยเพียงใดเมื่อเทียบกับอัลกอริทึมการค้นหาแบบทั้งหมด ทั้งในแง่ของความน่าจะเป็นในการพบเวกเตอร์การเคลื่อนที่ที่แท้จริง และระยะห่างเฉลี่ยระหว่างเวกเตอร์การเคลื่อนที่ที่แท้จริงให้ผลการทดลองที่ยืนยันความน่าเชื่อถือของการทำนายเวกเตอร์การเคลื่อนที่ได้

ผลการจำลองระบบเพื่อเปรียบเทียบสมรรถนะของอัลกอริทึมที่นำเสนอกับอัลกอริทึมแบบอื่น ๆ พบว่า ในลำดับภาพที่วัตถุเคลื่อนที่ช้าให้ค่า MSE ไม่แตกต่างกันมาก ในขณะที่เดียวกัน อัลกอริทึมการค้นหาแบบเพชชชนิดไม่สมมาตรแบบปรับตัวได้โดยใช้การปรับย้ายจุดศูนย์กลางสามารถลดจำนวนจุดการค้นหาที่ต้องการใช้ลงได้เป็นที่น่าพอใจ

#### 5.2 ข้อเสนอแนะสำหรับงานวิจัยในอนาคต

ปัญหาที่เกิดขึ้นในการทำงานวิจัยในวิทยานิพนธ์เล่มนี้ ได้แก่

1. เอกสารและข้อมูลที่เกี่ยวข้องกับงานทางด้านการศึกษาการเข้ารหัสสัญญาณดิจิทัลยังมีอยู่น้อยมากในประเทศไทย ควรจะมีแผนงานในการจัดหาเอกสารและเข้าร่วมการประชุมสัมมนาขององค์กรต่าง ๆ ที่วิจัยในเรื่องที่เกี่ยวข้อง
2. บุคลากรที่สนใจในการทำงานต่อเนื่องมีอยู่น้อยทำให้งานวิจัยใด ๆ เกิดการขาดช่วง



งานที่ควรได้รับการศึกษาหรือพัฒนาต่อไปในอนาคต คือ

1. การพัฒนาความสามารถในการทำนายเวกเตอร์การเคลื่อนที่ เพื่อให้ปรับรูปแบบการค้นหาให้สอดคล้อง หรือลดความซับซ้อนในการคำนวณได้มากยิ่งขึ้น
2. ขยายวัตถุประสงค์ในการประมาณการเคลื่อนที่ จากเดิมที่มุ่งเน้นการเข้ารหัสให้เร็วขึ้นและรักษาความถูกต้องของภาพ ให้เพิ่มขึ้น เช่น ใช้การประมาณการเคลื่อนที่เพื่อการแบ่งวัตถุภาพ (motion estimation for object segmentation)



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## รายการอ้างอิง

1. Schuster, G. M., and Katsaggelos, A. K. Rate-Distortion Based Video Compression.  
Netherlands : Kluwer Academic Publishers, 1999.
2. Kuhn, P. Algorithms, Complexity Analysis and VLSI Architecture for MPEG-4 Motion Estimation. Netherlands : Kluwer Academic Publishers, 1999.
3. Po, L.-M., and Ma, W.-C. A Novel Four-Step Search Algorithm for Fast Block Motion Estimation. IEEE Trans. Circuits and Systems for Video Technology Vol. 6 No. 3 (June 1996) : 313-317.
4. Jain, J.R., and Jain, A.K. Displacement measurement and its application in interframe image coding. IEEE Trans. Communications Vol. COM-29 (Dec. 1981) : 1799-1808.
5. Lu, J., and Liou, M. L. A simple and Efficient Search Algorithm for Block-Matching Motion Estimation. IEEE Trans. Circuits and Systems for Video Technology Vol. 7 No. 2 (April 1997) : 429-433.
6. Li, R., Zeng, B., and Liou, M. L. A New Three-Step Search Algorithm for Block Motion Estimation. IEEE Trans. Circuits and Systems for Video Technology Vol. 4 No. 4 (August 1994) : 438-442.
7. Zhu, S., and Ma, K.-K. A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation. IEEE Trans. Image Processing Vol. 9 No. 2 (Feb. 2000) : 287-290.
8. พณชิต กิตติปัญญางาม. อัลกอริทึมการเข้าคู่บล็อกการค้นหาแบบไฮบริดชนิดปรับตัวได้แบบใช้การปรับย้ายจุดศูนย์กลางสำหรับการประมาณการเคลื่อนที่ในการเข้ารหัสสัญญาณวิดีโอ. วิทยานิพนธ์ระดับมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2543.
9. Rao, K. R., and Hwang, J. J. Techniques and Standards for Image, Video and Audio Coding.  
USA. : Prentice Hall, 1996.
10. Brown, C. W., and Shepherd, B. J. Graphics File Formats, reference and guide. USA. : Prentice Hall, 1995.
11. Nam, J. Y., Seo, J. S., Kwak, J. S., and Lee, M. H. New Block-matching Algorithm for Motion Estimation Based on Predicted Direction Information. SPIE Proc. Visual Communications and Image Processing 2000 Vol. 4067 (June 2000) : 1212-1220.

12. Marichal, X. Motion Estimation and Compensation for very low bit rate video coding. Doctoral Dissertation from University catholique de Louvain, 1998.
13. Lee, W. Spatial Redundancy Reduction.  
<http://icsl.ee.washington.edu/~woobin/papers/General/node6.html> , 1995.
14. Lee, W. Temporal Redundancy Reduction.  
<http://icsl.ee.washington.edu/~woobin/papers/General/node4.html> , 1995.
15. Furht, W., Greenberg, J., and Westwater, R. Motion Estimation Algorithm for Video Compression. USA. : Kluwer Academic Publishers, 1997.
16. Liu, L. K., and Feig, E. A Block-based Gradient Descent Search Algorithm for block-based motion estimation in video coding. IEEE Trans. Circuits and Systems for Video Technology Vol. 6 No. 4 (Aug. 1996) : 419-422.
17. Xu, J.-B., Po, L. M., and Cheung, C.-K. Adaptive Motion Tracking Block Matching Algorithms for Video Coding. IEEE Trans. Circuits and Systems for Video Technology Vol. 9 No. 7 (Oct. 1999) : 1025-1029.
18. Du, C., and He, Y. A Comparative Study of Motion Estimation for Low Bit Rate Video Coding. SPIE Proc. Visual Communications and Image Processing 2000 Vol. 4067 (June 2000) : 1239-1249.
19. Lou, L., Zou, C., Gao, X., and He, Z. A new prediction search algorithm for block motion estimation in video coding. IEEE Trans. Consumer Electronics Vol. 43 (1997) : 56-61.
20. Tham, J. Y., Ranganath, S., Ranganath, M., and Kassim, A. A. A Novel Unrestricted Center-Biased Diamond Search Algorithm for Block Motion Estimation. IEEE Trans. Circuits and Systems for Video Technology Vol. 8 No. 4 (August 1998) : 369-377.
21. Liu, M.-G., Hou, C.-H. A Fast Block-Matching Motion Estimation Algorithm Based on Spatial-Temporal Motion Vector Correlation. Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing (May 2-4 2001) : 498-501.
22. Zheng, W., Ahmad, I., and Liou, M. L. Adaptive Motion Search With Elastic Diamond for MPEG-4 Video Coding. International Conference on Image Processing 2001 Vol. 1 (2001) : 377-380.



ภาคผนวก

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## ภาคผนวก ก

### ผลของการเปลี่ยนพารามิเตอร์ในการทดสอบ

Ming-gang Liu [21] นำเสนอการทำนายเวกเตอร์การเคลื่อนที่โดยอ้างอิงจากบล็อกข้างเคียง ทั้งในเชิงพื้นที่และเชิงเวลา

$$\begin{aligned} \text{โดยกำหนดให้ } B &= [b_{t-1,i,j} \quad b_{t-1,i,j+1} \quad b_{t,i-1,j} \quad b_{t,i,j-1} \quad b_{t-1,i+1,j}] \\ &= [b_0 \quad b_1 \quad b_2 \quad b_3 \quad b_4] \end{aligned} \quad (\text{ก.1})$$

เมื่อ  $b_{t,i,j}$  แทนเวกเตอร์การเคลื่อนที่ของบล็อกในตำแหน่ง  $(i,j)$  ในเฟรมลำดับที่  $t$

ทั้งนี้ จำแนกเวกเตอร์การเคลื่อนที่ออกเป็นสองประเภท ได้แก่ การเคลื่อนที่แบบทั่วไปและการเคลื่อนที่แบบเร็ว โดยกำหนดจากค่าจุดเปลี่ยน  $R_0$  เมื่อจำแนกลักษณะการเคลื่อนที่เป็นแบบเร็วเวกเตอร์การเคลื่อนที่ที่ได้จากการทำนายเป็น

$$b_{t,i,j}^p = I[\lambda B^T] \quad (\text{ก.2})$$

เมื่อ  $\lambda = [a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4]$  เป็นสัมประสิทธิ์ในการถ่วงน้ำหนัก ซึ่งสอดคล้องกับสมการ

$$\sum_{i=0}^4 a_i = 1 \quad (\text{ก.3})$$

ภาคผนวก ก แสดงผลการทดสอบเมื่อใช้ค่าพารามิเตอร์  $R_0$  และ  $\lambda$  บางค่า

กำหนดให้สัญลักษณ์ที่ใช้ในกราฟ มีดังนี้

AADS คือ อัลกอริทึมการค้นหารูปเพชรชนิดไม่สมมาตรแบบปรับตัวได้โดยใช้การปรับย้าย

จุดศูนย์กลาง

A มีค่าเป็น  $[0.4 \quad 0 \quad 0.3 \quad 0.3 \quad 0]$

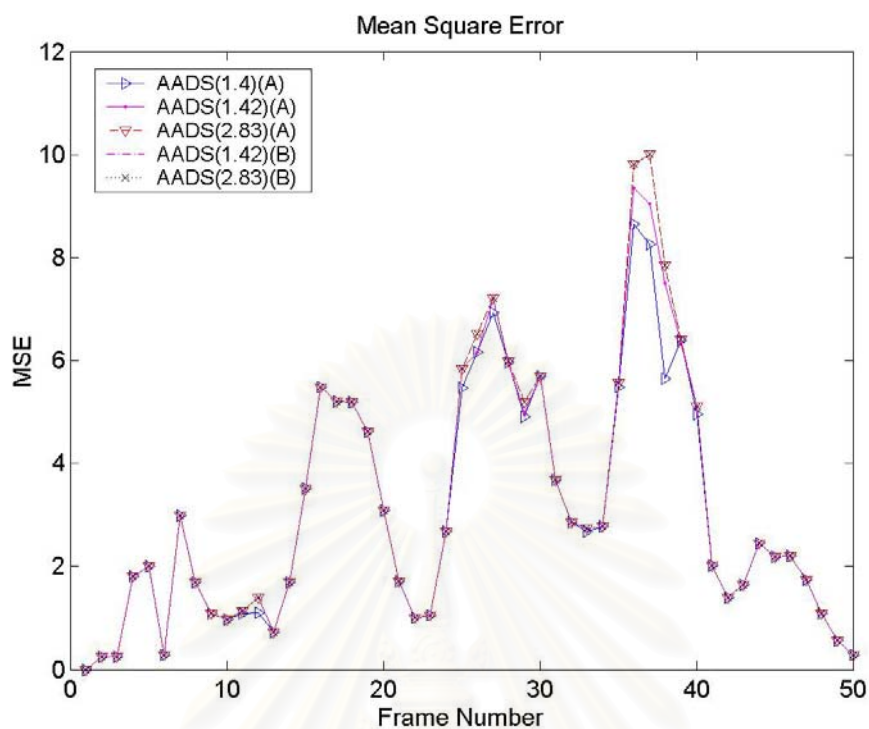
B มีค่าเป็น  $[0.4 \quad 0.1 \quad 0.2 \quad 0.2 \quad 0.1]$

การทดสอบกับลำดับภาพ Akiyo ซึ่งมีลักษณะการเคลื่อนที่ของวัตถุในภาพช้า จากหลัง  
หยุดนิ่ง ลำดับภาพ Carphone ซึ่งวัตถุในภาพเคลื่อนที่เร็ว ภาพจากหลังเคลื่อนที่ และลำดับภาพ  
Foreman ซึ่งวัตถุเคลื่อนที่ค่อนข้างเร็ว และมีการหมุนกลิ้ง

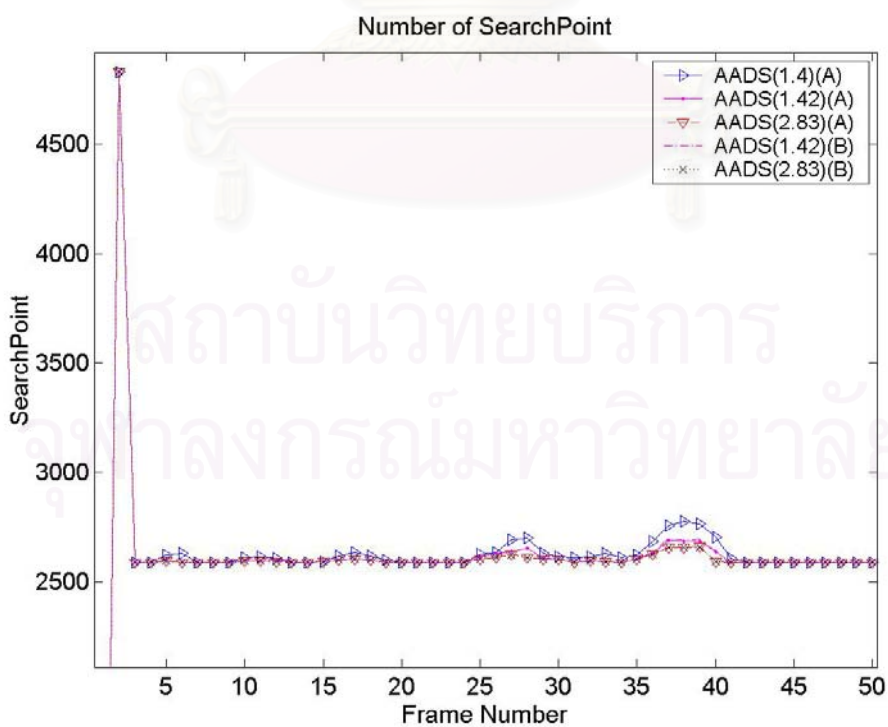


สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

### ก.1 ผลการทดสอบกับลำดับภาพ Akiyo

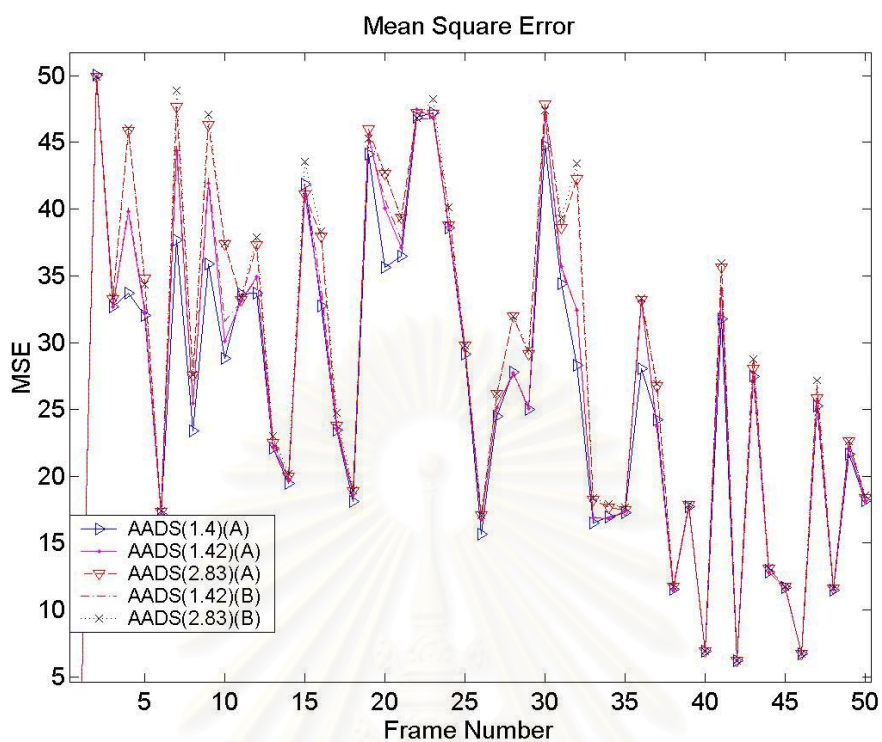


รูปที่ ก.1.1 ค่าความผิดพลาดกำลังสองเฉลี่ยของลำดับภาพ Akiyo เมื่อเปลี่ยนพารามิเตอร์บางค่า  
ในอัลกอริทึม AADS

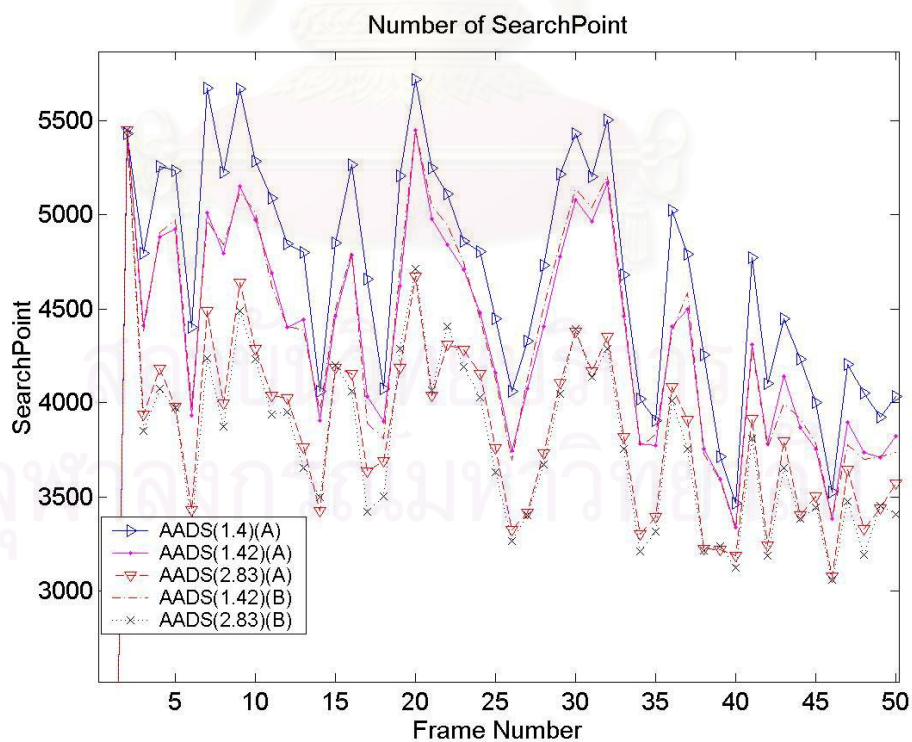


รูปที่ ก.1.2 จำนวนจุดการค้นหาของลำดับภาพ Akiyo เมื่อเปลี่ยนพารามิเตอร์บางค่า  
ในอัลกอริทึม AADS

## ก.2 ผลการทดสอบกับลำดับภาพ Carphone



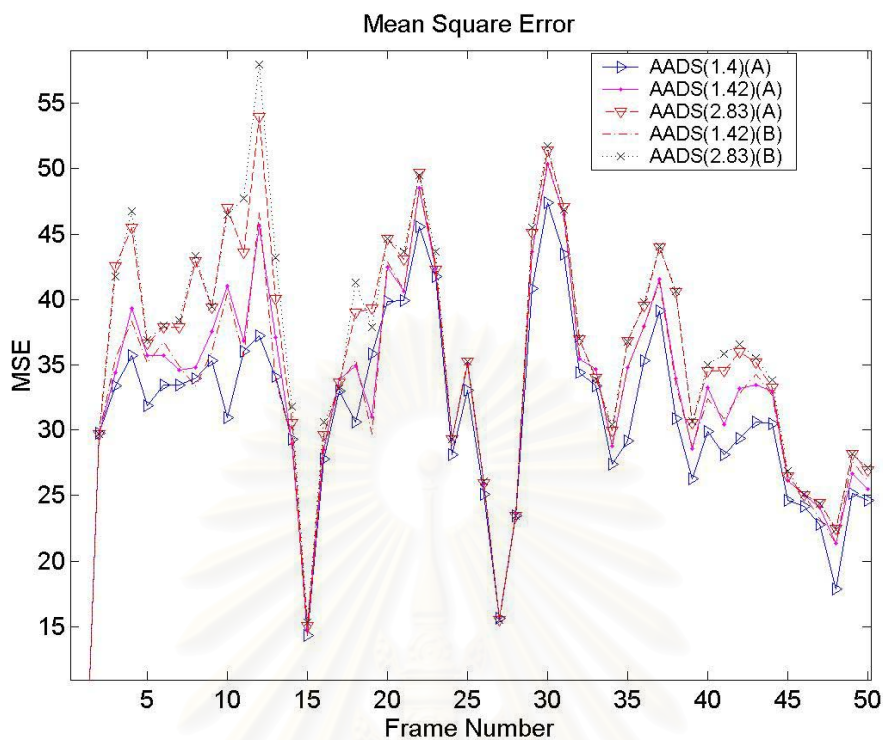
รูปที่ ก.2.1 ค่าความผิดพลาดกำลังสองเฉลี่ยของลำดับภาพ Carphone เมื่อเปลี่ยนพารามิเตอร์  
บางค่าในอัลกอริทึม AADS



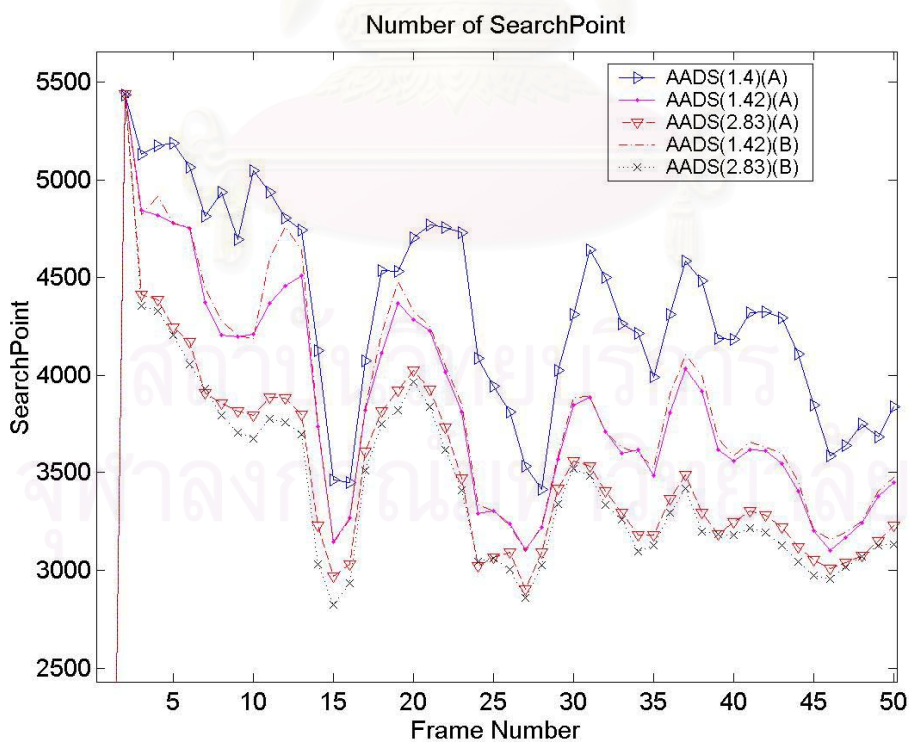
รูปที่ ก.2.2 จำนวนจุดการค้นหาของลำดับภาพ Carphone เมื่อเปลี่ยนพารามิเตอร์บางค่า  
ในอัลกอริทึม AADS



### ก.3 ผลการทดสอบกับลำดับภาพ Foreman



รูปที่ ก.3.1 ค่าความผิดพลาดกำลังสองเฉลี่ยของลำดับภาพ Foreman เมื่อเปลี่ยนพารามิเตอร์บางค่าในอัลกอริทึม AADS



รูปที่ ก.3.2 จำนวนจุดการค้นหาของลำดับภาพ Foreman เมื่อเปลี่ยนพารามิเตอร์บางค่าในอัลกอริทึม AADS

รูปที่ ก.1.1 แสดงค่าความผิดพลาดกำลังสองเฉลี่ยของลำดับภาพ Akiyo เมื่อค่าจุดเปลี่ยน  $R_0$  เพิ่มมากขึ้น มีแนวโน้มว่าความผิดพลาดกำลังสองเฉลี่ยจะสูงขึ้นตามไปด้วย รูปที่ ก.1.2 แสดงจำนวนจุดการค้นหาที่ใช้ เมื่อค่าจุดเปลี่ยน  $R_0$  มีค่าเพิ่มสูงขึ้นมีแนวโน้มว่าจะใช้จำนวนจุดการค้นหาลดลง เนื่องจากอัลกอริทึมจำแนกลักษณะการเคลื่อนที่เป็นการเคลื่อนที่แบบทั่วไปมากขึ้น ส่งผลให้ใช้รูปแบบการค้นหาแบบเพชรเล็กในการค้นหามากขึ้น จำนวนจุดการค้นหาจึงลดลง ช่วงเฟรมที่ 2 ใช้จำนวนจุดการค้นหาสูงกว่าจุดการค้นหาของเฟรมที่ 3 เป็นต้นไป ซึ่งเป็นเฟรมที่เริ่มต้นใช้อัลกอริทึม AADS เพราะอัลกอริทึมนี้ต้องใช้ข้อมูลเวกเตอร์การเคลื่อนที่จากเชิงพื้นที่และเชิงเวลา

รูปที่ ก.2.1 และรูปที่ ก.3.1 มีลักษณะเช่นเดียวกับรูปที่ ก.1.1 รูปที่ ก.2.2 และ รูปที่ ก.3.2 มีลักษณะเช่นเดียวกับรูปที่ ก.1.2



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## ภาคผนวก ข

## บทความทางวิชาการที่ได้รับการเผยแพร่

1. บทความทางวิชาการจากงานประชุมทางวิชาการ IEEE Circuits and Systems Colloquium on Signal Processing ซึ่งจัดขึ้นในวันที่ 22 มีนาคม พ.ศ. 2545 ณ มหาวิทยาลัยมหานคร, อาคาร วาณิช ประเทศไทย
2. ได้รับการตอบรับจากงานประชุมทางวิชาการ IEEE International Symposium on Video / Image Processing and Multimedia Communications (IEEE-VIPromcom-2002) ซึ่งจะจัดขึ้นในวันที่ 16-19 มิถุนายน พ.ศ. 2545 ณ เมือง Zadar ประเทศโครเอเชีย
3. ได้รับการตอบรับจากงานประชุมทางวิชาการ IEEE International Conference on Digital Signal Processing (IEEE-DSP-2002) ซึ่งจะจัดขึ้นในวันที่ 1-3 กรกฎาคม พ.ศ. 2545 ณ เมือง Santorini ประเทศกรีซ



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## ADAPTIVE ASYMMETRIC DIAMOND SEARCH ALGORITHM FOR BLOCK-BASED MOTION ESTIMATION

*Thaveesak Sappasitwong, Somchai Jitapunkul, Atit Tamtrakarn,  
Punnachit Kitti-punyangam, Supavadee Aramvith and Hatairat Kortrakulkij*

Department of Electrical Engineering,  
Chulalongkorn University, Bangkok, Thailand

### ABSTRACT

Based on the correlation of motion vectors between adjacent blocks, an adaptive asymmetric diamond search algorithm (AADS) for block based motion estimation is proposed. The proposed method exploits adaptive search pattern depend on predicted motion vector. In addition, the accuracy of the prediction is improved by changing initial search center position. Simulation results show that the proposed algorithm can reduce search points up to about 50% compared to the regular diamond search while keeping the picture quality similar to other fast search algorithms.

### 1. INTRODUCTION

The memory and bandwidth requirements of full-motion video are almost impractical. Video coding is an necessary process for several visual communication applications. Fortunately, the high correlation between successive frames of a video sequence makes it possible to achieve high compression by reducing the temporal redundancy. Motion estimation (ME) has proven to be effective to exploit the temporal redundancy of video sequences and is therefore a integral part of the ISO/IEC MPEG-1, MPEG-2, MPEG-4, CCITT H.261 and ITU-T H.263 video compression standard [1]. The block-matching algorithm (BMA's) is most commonly used to implement standardized block-based coding schemes which searched the best match block of the current frame from the candidate blocks inside the search window in the previous frame. The Full Search (FS) algorithm can find the optimal matching block by exhaustively checking all candidate blocks within the search window. However, its computational requirement is too large. To overcome this problem, many fast block-matching algorithms

have been developed, such as *2-D logarithmic search (LOGS)* [2], *three-step search (TSS)* [3], *new three-step search (NTSS)* [4], *four-step search (FSS)* [5], *new diamond search (DS)* [6], *block-based gradient descent search (BBGDS)* [7], and etc. These fast BMA's have different search patterns and search strategies for finding the motion vector with much less number of search points compared to the FS algorithm.

In most video sequences, motion objects usually cover several blocks, so the motion vector of the spatial neighbor blocks may be very closely. In addition, the continuity of the motion in the temporal domain makes motion vectors of the temporal neighbor blocks may be highly correlated. In this paper, we proposed a motion estimation algorithm that exploits new diamond search pattern which adaptively select shape of the search pattern. Besides that, the predicted motion vector helps the motion estimation to start search at a better position so that the average search point could be further reduced and increase chance to jump out of the local minimum point.

This paper is organized as follows. In the following section, The proposed algorithm is described. Simulation results are shown in section 3. Finally, in section 4, the conclusions are given.

### 2. ADAPTIVE ASYMMETRIC DIAMOND SEARCH ALGORITHM (AADS)

Due to the fact that the motion vector in video sequences are highly correlated to its neighbor's motion vector. Some linear prediction of motion vectors utilizing spatial neighbor blocks are proposed [8]. Based on the spatial-temporal motion vector correlation, the prediction by the use of temporal neighbor blocks is included [9].

Our proposed algorithm has two stages. The first stage is a motion vector prediction. The second stage is an adaptive asymmetric

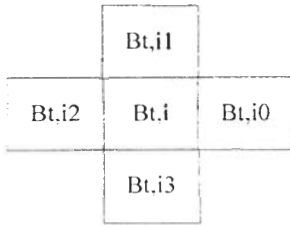


Fig. 1. The neighboring blocks in the motion vector prediction stage.

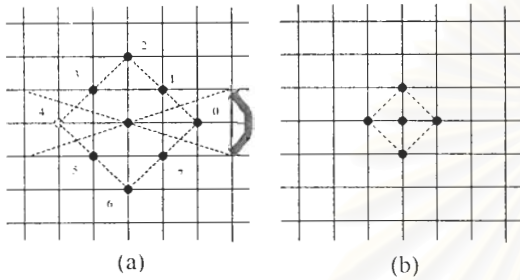


Fig. 2. Two search patterns which selectively used. (a) Large Asymmetric Diamond Search Pattern: LADS (b) Small Diamond Search Pattern: SDSP

diamond search algorithm using adaptive search center.

### Stage 1. Motion vector prediction

Let  $b_{t,i}$  is the current block that is being processed of the  $t$ 'th frame. The other neighboring blocks is shown in figure. 1.

$$\text{Let } B = \begin{bmatrix} b_{t-1,i} & b_{t-1,i0} & b_{t,i1} & b_{t,i2} & b_{t-1,i3} \\ b_0 & b_1 & b_2 & b_3 & b_4 \end{bmatrix} \quad (1)$$

denote the neighboring blocks used in prediction process. And  $\lambda = [a_0 \ a_1 \ a_2 \ a_3 \ a_4]$  means the weighting matrix of the neighboring blocks B, with the normalization

$$\sum_{i=0}^4 a_i = 1 \quad (2)$$

The determination of the search pattern is depend on type of motion in each block. Two type of motion are as follow:

**Type 1.** gentle motion (stationary or quasi-stationary) by the criterion

$$\text{If } \|b_i - b_0\| \leq R_0, \forall i \in [0,4], \text{ then} \\ b_{i,j}^p = \text{sign}(\cdot) \times I[\text{abs}(b_{t-1,j} + b_{t,i1} + b_{t,i2}) / 3.0 + 0.35] \quad (3)$$

is the predicted motion vector, where  $\text{sign}(\cdot)$  is a sign function, assign from the sign of  $b_{t-1,j} + b_{t,i1} + b_{t,i2}$ ,  $I[\cdot]$  denote rounding toward zero (rounds to the nearest integer toward zero)

**Type 2.** fast motion, which satisfies

$$\|b_i - b_0\| > R_0, \exists i \in [0,4], \text{ then}$$

$$b_{i,j}^p = I[\lambda B^T] \quad (4)$$

### Stage 2. AADS algorithm

As illustrated in fig.2, The proposed algorithm is summarized as follows:

Step 1.(a) While the motion is type 1, eq.(3) is used to predict motion vector. Small pattern is selected.

(b) While the motion is type 2, eq.(4) is chosen. A search point opposite to the angle of predicted motion vector is omitted. LADS pattern is used. This pattern is hold until the motion vector is obtain.

Step 2.MAE of the original search center (0,0) and the position found in prediction stage is compared. The point which obtains the minimum BDM is chosen as the search center.

Step 3.(a) If small pattern is applied, if the minimum block distortion (MBD) point is located at the center, it is returned as motion vector; if not, go to step 4(a).

(b) If LADS is used, If the MBD point is located at the center, go to Step 5; otherwise, go to Step 4(b).

Step 4.(a) The MBD point in the previous search step is repositioned to form a new pattern.The position of MBD point found in this step is the final solution.(one iteration is allowed)

(b) If the MBD point is located at the center, go to Step 5; otherwise, iterate this search step.

Step 5.Change the search pattern to SDSP. The MBD point is selected to be the motion vector.

## 3. SIMULATION RESULTS

Simulation results using the luminance component of the first 50 frames of various sequences. The size of each frame is  $176 \times 144$  pixels. The block size is fixed at  $8 \times 8$ . The matching algorithm is proceed within  $15 \times 15$  search window. The mean absolute error (MAE)

distortion function is used as the block distortion measurement. For the blocks in the first row and column, they have no prediction and use conventional diamond search as the search algorithm.

We compare AADS with other well-known algorithms such as FS, TSS, FSS and DS, as given in Table 1, using the following criteria.

1) Average MSE per pels: This shows the distortion per pixels between reconstructed frames and original frames.

2) Probability of finding motion vector as that of FS: This displays probability of the algorithms to obtain the same motion vector as those found using FS algorithm.

3) Average distance from motion vector of FS per block: This measures the Euclidean distance between obtained motion vector and that of FS.

4) Average number of search points per motion vector: This can be represents the complexity of algorithm. In addition, speed up ratio which is the ratio of the number of search points obtained from FS and algorithm is included.

we can see that the proposed algorithm reduces the number of search points in the search procedure. For a stationary or quasi-stationary sequence such as "Akiyo", the proposed algorithm reduces the number of search point while MSE is almost the same as that of full search algorithm. For the sequences with large displacement and fast motion, for example, "Carphone" and "Foreman", AADS works very well in term of accuracy and computation complexity as shown in fig. 3. Besides that, we show the 12<sup>th</sup> frame of the original and reconstructed images which has the largest distortion of 2.59 dB in fig. 4.

#### 4. CONCLUSION

In this paper, we proposed an adaptive asymmetric diamond search algorithm to reduce complexity of motion estimation procedure. This algorithm exploits spatial-temporal correlation to predict motion vector and adapt search pattern suitable with each sequence. Simulation results show improvement in speed up ratio up to 21 times of FS, while quality is almost that of other fast search algorithm.

#### 6. REFERENCES

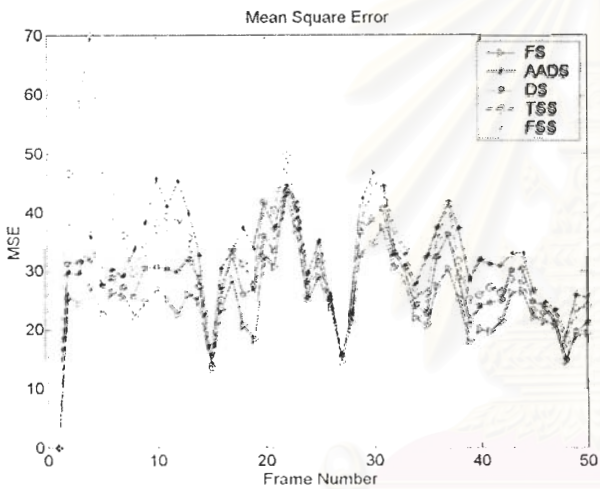
- [1] P. Kuhn, "Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation," Kluwer Academic Publishers, Netherlands, p.17, 1999.
- [2] J. Jain and A. Jain, "Displacement measurement and its application in interframe image coding." *IEEE Trans. Commun.*, vol. COMM-29, pp. 1799-1808, Dec. 1981.
- [3] T. Koga, K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, "Motion compensated interframe coding for video conferencing," *Proc.Nat. Telecommun. Conf.*, New Orleans, LA, Nov. 29-Dec. 3 1981, pp. G5.3.1-5.3.5.
- [4] S. Zhu and K. K. Ma, "A new three-step search algorithm for block motion estimation," *IEEE Trans.Circuits Syst.Video Technol.*, Vol. 4, pp. 438-442, Aug. 1994.
- [5] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, Vol.6, pp. 313-317, June 1996.
- [6] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. Image Processing*, vol. 9, pp. 287-290, Feb. 2000.
- [7] L. K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding." *IEEE Trans. Circuits Syst. Video Technol.*, Vol.6, pp. 419-423, Aug. 1996.
- [8] Lijun Luo, Cairong Zou and Xiqi Gao. "A New Prediction Search Algorithm for Block Motion Estimation in Video Coding." *IEEE Trans. Consumer Electronics*, Vol. 43, No. 1, Feb. 1997.
- [9] Ming-gang Liu and Chao-huan Hou, "A Fast Block-Matching Motion Estimation Algorithm Based on Spatial-Temporal Motion Vector Correlation." *Proc. 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing*, Hong Kong, May 2001.

Table 1.  
Performance Comparisons of FS, TSS, FSS and AADS using “Foreman” sequence.

	Average MSE	Prob.	Average Distance	Aver. Search Points	Speed up ratio
FS	25.09	-	-	225	1.0
TSS	33.08	0.997	3.16	25.00	9.0
FSS	30.05	0.997	3.49	18.44	12.2
DS	28.57	0.998	4.70	14.98	15.0
AADS	32.94	0.996	2.91	10.67	21.1

Table 2.  
Performance Comparisons of FS, TSS, FSS and AADS using “Carphone” sequence.

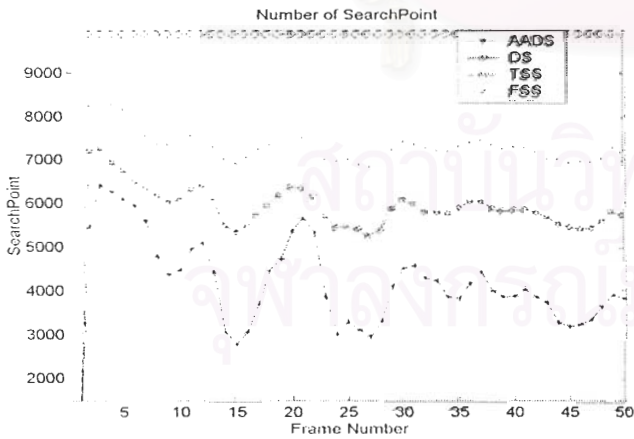
	Average MSE	Prob.	Average Distance	Aver. Search Points	Speed up ratio
FS	23.42	-	-	225	1.0
TSS	27.52	0.998	5.82	25.00	9.0
FSS	27.20	0.998	5.91	18.28	12.31
DS	25.29	0.997	5.32	15.12	14.88
AADS	27.41	0.997	4.58	10.88	21.09



(a)



(a)



(b)



(b)

Figure 3. Simulation results of “Foreman” sequence (a) mean square error (b) number of search points

Figure 4. Performance evaluation of 12<sup>th</sup> of “Foreman” sequence (a) an original image (b) reconstructed image of AADSC algorithm.

## ADAPTIVE ASYMMETRIC DIAMOND SEARCH ALGORITHM FOR BLOCK-BASED MOTION ESTIMATION

*Thaveesak Sappasitwong, Supavadee Aramvith, Somchai Jitapunkul, Atit Tamtrakarn, Punnachit Kitti-punyangam,  
and Hatairat Kortrakulkij*

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

supava@ee.eng.chula.ac.th

**Abstract:** In this paper, we present an adaptive asymmetric diamond search (AADS) for block based motion estimation. The proposed method exploits adaptive search pattern based on the predicted motion vector. In addition, the accuracy of the prediction is improved by changing initial search center position. Simulation results show that the proposed algorithm can reduce the number of search points up to about 50% compared to the regular diamond search while keeping the picture quality similar to other fast search algorithms.

### 1. INTRODUCTION

Due to limited channel bandwidth and very large sizes of uncompressed video, video compression plays a significant role in video coding systems. Motion estimation (ME) has proven to be effective to exploit the temporal redundancy of video sequences and is therefore an integral part of MPEG-1, MPEG-2, MPEG-4, H.261 and H.263 video compression standard [1]. The block-matching algorithm (BMA), the most commonly used in the standardized block-based coding schemes, searches the best match block of the current frame from the candidate blocks inside the search window in the previous frame. The Full Search (FS) algorithm can find the optimal matching block by exhaustively checking all candidate blocks within the search window. However, its computational requirement is expensive. To overcome this problem, many fast block-matching algorithms have been developed, such as 2-D logarithmic search (LOGS) [2], three-step search (TSS) [3], new three-step search (NTSS) [4], four-step search (FSS) [5], new diamond search (DS) [6], block-based gradient descent search (BBGDS) [7], and etc. These fast algorithms use different search patterns and search strategies to find motion vector with much less number of search points compared to the FS algorithm.

In most video sequences, motion vector of a block is highly correlated to the motion vectors of the adjacent blocks and also the motion vector of the temporal neighboring blocks. In this paper, we propose a motion estimation algorithm that exploits new diamond search pattern which adaptively selects shape of a search pattern. Besides that, the predicted motion vector helps the motion estimation to start search at a better position so that the average search point could be further reduced and increase the chance to jump out of the local minimum point. This paper is organized as follows. In the following section, the proposed algorithm is described. Simulation results are shown in section 3. The conclusions are given in section 4.

### 2. ADAPTIVE ASYMMETRIC DIAMOND SEARCH ALGORITHM (AADS)

Due to the fact that the motion vector in video sequences are highly correlated to its neighbor's motion vector, some linear prediction of motion vectors utilizing spatial neighbor blocks are proposed [8]. Based on the spatial-temporal motion vector correlation, the temporal neighboring blocks are used in the motion vector prediction [9]. Our proposed algorithm has two stages which are described as follows:



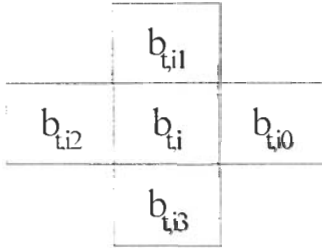


Figure 1 The neighboring blocks in the motion vector prediction stage.

### Stage 1. Motion vector prediction

Let  $b_{t,i}$  is the current block in the  $t^{\text{th}}$  frame. The neighboring blocks are shown in Figure 1. Let

$$B = \begin{bmatrix} b_{t-1,i} & b_{t-1,i+1} & b_{t,i} & b_{t,i+1} & b_{t-1,i+2} \\ b_0 & b_1 & b_2 & b_3 & b_4 \end{bmatrix} \quad (1)$$

denotes the neighboring blocks used in prediction process. Define  $\lambda = [a_0 \ a_1 \ a_2 \ a_3 \ a_4]$  as the weighting matrix of the neighboring blocks  $B$ , with

$$\text{the normalization: } \sum_{i=0}^4 a_i = 1 \quad (2)$$

There are two search patterns, large asymmetric diamond search (LADS) pattern and small diamond search (SDS) pattern, as shown in Figure 2. The determination of the search pattern is based on motion types in each block. Two types of motion are defined as follows:

**Type 1.** The motion is gentle motion (stationary or quasi-stationary)

if  $\|b_i - b_0\| \leq R_0$ ,  $\forall i \in [0, 4]$ , then assign,

$$b_{t,i}^p = \text{sign}(\cdot) \times I[\text{abs}(b_{t-1,i} + b_{t,i+1} + b_{t,i+2}) / 3.0 + 0.35] \quad (3)$$

as the predicted motion vector, where  $\text{sign}(\cdot)$  is a sign function for  $b_{t-1,i} + b_{t,i+1} + b_{t,i+2}$ ,  $I[\cdot]$  denotes rounding toward zero, i.e., rounding to the nearest integer toward zero.

**Type 2.** The motion is fast motion, which satisfies  $\|b_i - b_0\| > R_0$ ,  $\exists i \in [0, 4]$ , then

$$b_{t,i}^p = I[\lambda B^T] \quad (4)$$

**Stage 2.** AADS algorithm is summarized as follows:

Step 1: If the motion is type 1, eq. (3) is used to predict motion vector with SDS pattern. If the motion is type 2, eq.(4) is chosen with LADS pattern. A search point opposite to the angle of

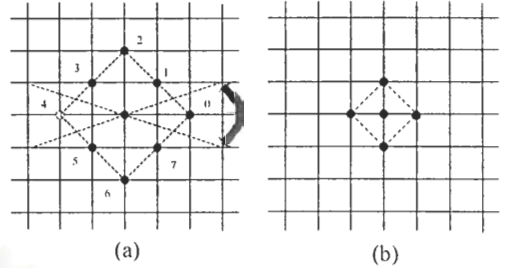


Figure 2 Two search patterns. (a) Large Asymmetric Diamond Search Pattern: (LADS) (b) Small Diamond Search Pattern: (SDS)

predicted motion vector is omitted. This pattern is used until the motion vector is obtained.

Step 2: Mean absolute error (MAE) of the original search center (0,0) and the position found in prediction stage are compared. The point which has minimum block distortion (MBD), i.e., minimum MAE, is chosen as the search center.

Step 3: (a) If SDS pattern is used, and if MBD point is located at the center, it is a motion vector; otherwise, go to step 4 (a).

(b) If LADS is used, check the MBD point is located at the center, go to Step 5; otherwise, go to Step 4 (b).

Step 4: (a) The MBD point in the previous search step is repositioned to form a new pattern. The position of MBD point found in this step is the final solution (only one iteration is allowed).

(b) If the MBD point is located at the center, go to Step 5; otherwise, iterate this search step.

Step 5. Change the search pattern to SDS. The MBD point is selected to be the motion vector.

### 3. SIMULATION RESULTS

Simulation results using the luminance component of the first 50 frames of various sequences. The size of each frame is 176x144 pixels. The block size is 8x8 pixels. Search window of 15x15 pixels is used. MAE is used as the block distortion measurement. For the blocks in the first row and column, they have no prediction and use conventional diamond search as the search algorithm. In the simulations, we use  $\lambda = [0.4 \ 0 \ 0.3 \ 0.3 \ 0]$ , and  $R_0 = \sqrt{2}$ .

We compare AADS with other well-known algorithms such as FS, TSS, FSS and DS, as shown in Table 1 and 2, for Salesman and Akiyo sequences, respectively, using the following criteria:

- Average mean square error (MSE) per pixels: This shows the distortion per pixels between original and reconstructed frames.
- Probability of finding motion vector as that of FS: This value represents probability of the algorithms to obtain the same motion vector as those found using FS algorithm.
- Average distance from motion vector of FS per block: This measures the Euclidean distance between obtained motion vector and that of FS.
- Average number of search points per motion vector: This value represents the complexity of algorithm. In addition, the speed-up ratio, which is the ratio of the number of search points obtained from FS and algorithm, is included.

We can see that the proposed algorithm can reduce the number of search points of up to ~50% in the search procedure compared to DS scheme. Figure 3 shows the MSE and the number of search points for the sequences with large displacement and fast motion, "Carphone". We can see that AADS works very well in terms of accuracy and computation complexity. Besides that, the comparison of the 23<sup>th</sup> frame between an original and reconstructed image with the largest value of MSE of 47.19 is shown with a little noticeable visible distortion in Figure 4.

#### 4. CONCLUSION

In this paper, we proposed an adaptive asymmetric diamond search algorithm to reduce complexity of motion estimation procedure. This algorithm exploits spatial-temporal correlation to predict motion vector and adaptively select suitable search pattern for each sequence. Simulation results show our proposed AADS has less computational complexity than other fast search algorithms, while maintaining almost about the same quality.

#### REFERENCES

- [1] P. Kuhn, "Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation," Kluwer Academic Publishers, Netherlands, p.17, 1999.
- [2] J. Jain and A. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. COMM-29, pp. 1799-1808, Dec. 1981.
- [3] T. Koga, K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, "Motion compensated interframe coding for video conferencing," *Proc.Nat. Telecommun. Conf.*, New Orleans, LA, Nov. 29-Dec. 3 1981, pp. G5.3.1-5.3.5.
- [4] S. Zhu and K. K. Ma, "A new three-step search algorithm for block motion estimation," *IEEE Trans.Circuits Syst.Video Technol.*, Vol. 4, pp. 438-442, Aug. 1994.
- [5] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, Vol.6, pp. 313-317, June 1996.
- [6] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. Image Processing*, vol. 9, pp. 287-290, Feb. 2000.
- [7] L. K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," *IEEE Trans. Circuits Syst. Video Technol.*, Vol.6, pp. 419-423, Aug. 1996.
- [8] Lijun Luo, Cairong Zou and Xiqi Gao, "A New Prediction Search Algorithm for Block Motion Estimation in Video Coding," *IEEE Trans. Consumer Electronics*, Vol. 43, No. 1, Feb. 1997.
- [9] Ming-gang Liu and Chao-huan Hou, "A Fast Block-Matching Motion Estimation Algorithm Based on Spatial-Temporal Motion Vector Correlation," *Proc. 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing*, Hong Kong, May 2001.

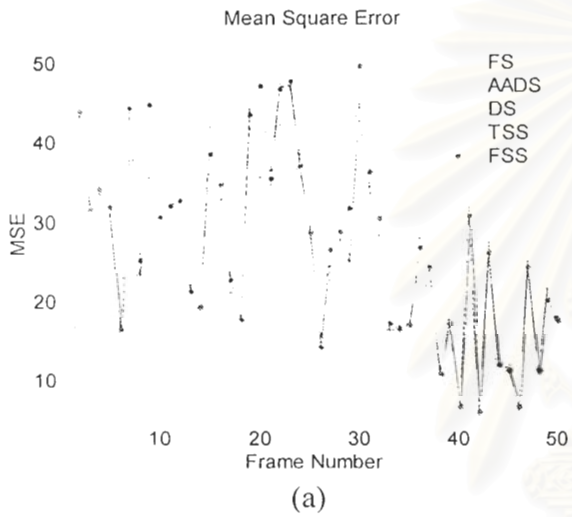
\*Table 1

Performance Comparisons of FS, TSS, FSS and AADS schemes using "Salesman" sequence.

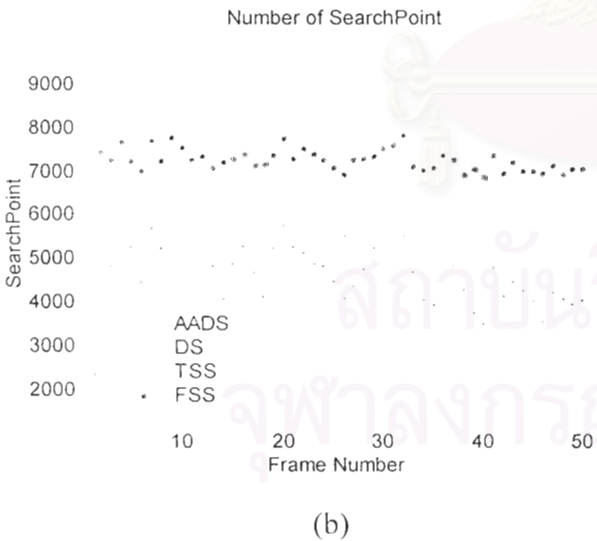
	Average MSE	Prob.	Average Distance	Average Search points	Speed-up ratio
FS	7.5549	-	-	225	1.0
TSS	8.0533	0.9897	0.0241	25	9.0
FSS	7.9499	0.9911	0.0198	17.08	13.18
DS	7.7152	0.9941	0.0150	13.13	17.14
AADS	7.9618	0.9890	0.0196	7.15	31.48

Table 2  
Performance Comparisons of FS, TSS, FSS and AADS schemes using "Akiyo" sequence.

	Average MSE	Prob.	Average Distance	Average Search points	Speed-up ratio
FS	2.9609	-	-	225	1.0
TSS	3.0057	0.9992	0.0012	25	9.0
FSS	3.0057	0.9992	0.0012	17.00	13.23
DS	2.9711	0.9994	0.0012	13.01	17.29
AADS	3.0920	0.9979	0.0027	6.73	33.42



(a)



(b)

Figure 4 Performance evaluation of 23<sup>th</sup> of "Carphone" sequence (a) an original image (b) a reconstructed image of AADS algorithm.

Figure 3. Simulation results of "Carphone" sequence (a) mean square error (b) number of search points

## ประวัติผู้เขียนวิทยานิพนธ์

นายทวีศักดิ์ สรรพสิทธิ์วงศ์ เกิดวันที่ 14 สิงหาคม พ.ศ. 2522 ที่จังหวัดกรุงเทพมหานคร เข้าศึกษาในหลักสูตรวิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2539 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ที่ห้องปฏิบัติการวิจัยกรรมวิธีสัณฐานคิเจิ้ล ภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย ในปี การศึกษา 2543



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย