

อัลกอริทึมการประมาณการเคลื่อนที่แบบขนานบนเอฟพีจีเอ

A PARALLEL MOTION ESTIMATION ALGORITHM ON FPGA

มานะ ทรัพย์ประพันธ์
MANA GRASPRAPUN

วิทยานิพนธ์เสนอแก่บัณฑิตวิทยาลัยของภาควิชาวิศวกรรมคอมพิวเตอร์
วิทยาลัยวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

สาขาวิชาวิศวกรรมคอมพิวเตอร์

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี กรุงเทพมหานคร

พ.ศ. 2550

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

อัลกอริทึมการประมาณการเคลื่อนที่แบบขนานบนเอฟพีจีเอ

A PARALLEL MOTION ESTIMATION ALGORITHM ON FPGA

มานะ กรัชประพันธ์

MANA GRASPRAPUN

เลขหมู่.....
เลขทะเบียน..... 74618
วัน,เดือน,ปี..... - 8 ต.ค. 2550

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2550

A PARALLEL MOTION ESTIMATION ALGORITHM FOR FPGA

MANA GRASPRAPUN

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN COMPUTER ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2007

COPYRIGHT 2007

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

หัวข้อวิทยานิพนธ์	อัลกอริทึมการประมาณการเคลื่อนที่แบบขนานบนเอฟพีจีเอ
นักศึกษา	นายมานะ กรีษประพันธ์
รหัสนักศึกษา	45061042
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
พ.ศ.	2550
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ. อภินันท์ อุณากร

บทคัดย่อ

งานวิจัยนี้เป็นการนำเสนออัลกอริทึมใหม่ในการประมาณการเคลื่อนที่ (Motion Estimation) ซึ่งเป็นขั้นตอนหนึ่งในการบีบอัดภาพวิดีโอ ที่เหมาะสมกับการนำไปออกแบบเป็นฮาร์ดแวร์เพื่อช่วยประมวลผลสำหรับระบบอุปกรณ์พกพา (Mobile Device) ซึ่งอัลกอริทึมนี้มีชื่อว่าอัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน 2 ระดับ (Two-Level Parallel Hierarchical One-Dimensional Search : 2LPHODS) เป็นอัลกอริทึมที่พัฒนามาจาก อัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน (Parallel Hierarchical One-Dimensional Search : PHODS) ซึ่งเป็นอัลกอริทึมที่ถูกออกแบบมาสำหรับรองรับการประมวลผลแบบขนาน จึงเหมาะสมสำหรับการนำมาสร้างเป็นชิพช่วยประมวลผลเป็นอย่างยิ่ง แต่อัลกอริทึมเดิมมีปัญหาในด้านความคลาดเคลื่อนของผลลัพธ์ งานวิจัยนี้จึงได้ทำการปรับปรุงแก้ไขข้อบกพร่องดังกล่าวเพื่อให้ประสิทธิภาพของการประมาณการเคลื่อนที่ด้วยวิธีนี้สูงขึ้น นอกจากนี้งานวิจัยนี้ยังนำเสนออัลกอริทึมที่ปรับปรุงขึ้นมาในรูปแบบของฮาร์ดแวร์ เพื่อนำข้อดีในการประมวลผลแบบขนานของฮาร์ดแวร์มาช่วยเพิ่มประสิทธิภาพของอัลกอริทึมนี้ เพื่อให้อุปกรณ์พกพาสามารถเข้ารหัสวิดีโอได้ในความเร็วระดับเรียลไทม์

Thesis Title	A parallel motion estimation algorithm on FPGA
Student	Mana Grasrapun
Student ID.	45061042
Degree	Master of Engineering
Program	Computer Engineering
Year	2007
Thesis Advisor	Asst. Prof. Apinetr Unakul

ABSTRACT

This research proposes a new algorithm for motion estimation that is a path of many videos compression. The algorithm is appropriate to design hardware for help processing mobile device. This algorithm called Two-Level Parallel Hierarchical One-Dimension Search (2LPHODS). The research algorithm is developed from Parallel Hierarchical One-Dimensional Search (PHODS) which designed to support the parallel processing, thus is suitable for implement ship in helping processing system. However, the PHODS has problem on precision errors.

Therefore, this research proposed algorithm improved from the PHODS for improving the efficiency of the motion estimation. Besides, this research has use this algorithm for utilize in hardware in order to increase the efficiency of algorithm and thus the mobile device can increase time for compressing the video.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้จะสำเร็จลุล่วงไปได้ หากปราศจากความช่วยเหลือจากผู้มีพระคุณหลายท่าน ขอกราบขอบพระคุณบิดามารดาของข้าพเจ้าที่ได้ให้กำลังใจ และสนับสนุนค่าใช้จ่ายในการทำงานวิจัยมาโดยตลอด

ขอกราบขอบพระคุณท่านอาจารย์ที่ปรึกษา ผศ.อภิเนตร อุณาภูล สำหรับวิชาความรู้ที่ได้มอบให้ การให้คำปรึกษา คอยให้แนวทางให้การแก้ปัญหา จนสามารถสำเร็จเป็นวิทยานิพนธ์นี้ รวมทั้งขอขอบพระคุณที่อาจารย์คอยให้ความช่วยเหลือต่าง ๆ กับข้าพเจ้า ถึงแม้ข้าพเจ้าจะสร้างความเดือดร้อนให้กับท่านหลายต่อหลายครั้งแล้วก็ตาม อันเนื่องจากความประพฤติกที่ไม่ค่อยดีของตัวข้าพเจ้า

ขอบคุณเพื่อน ๆ รุ่นพี่ และรุ่นน้องที่คอยเป็นกำลังใจ และให้ความช่วยเหลือ ให้คำแนะนำและเป็นกำลังใจที่ดีมาตลอด

ขอบคุณเจ้าหน้าที่บัณฑิตศึกษาคณะวิศวกรรมศาสตร์ ที่ให้ความช่วยเหลือให้ข้าพเจ้าสามารถยื่นสอบวิทยานิพนธ์ได้ทันเวลา

สุดท้ายขอกล่าวขอบ เจ้าหน้าที่บัณฑิตวิทยาลัยที่อำนวยความสะดวกและให้คำปรึกษาในการทำวิทยานิพนธ์ครั้งนี้

คุณความดีและประโยชน์ที่ได้จากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอบมอบให้บิดา มารดาที่รักและเคารพ ตลอดจนถึงผู้มีพระคุณทุกท่าน

มานะ กรีษประพันธ์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูปภาพ.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมติฐานของการศึกษา.....	2
1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย.....	3
1.5 ขอบเขตการวิจัย	3
1.6 ขั้นตอนของการศึกษา	4
บทที่ 2 อัลกอริทึมที่ใช้ในการประมาณการเคลื่อนที่	5
2.1 การประมาณการเคลื่อนที่	5
2.2 อัลกอริทึมการเปรียบเทียบทั้งกลุ่ม	6
2.3 อัลกอริทึมการเปรียบเทียบทั้งกลุ่มแบบรวดเร็ว	8
2.4 อัลกอริทึมการค้นหาแบบ 2 ขั้นตอน	9
2.5 อัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนาน.....	12
2.6 การเปรียบเทียบคุณภาพของอัลกอริทึม.....	15
2.6.1 อภิโยชะ.....	17
2.6.2 โคสท์การ์ด.....	18
2.6.3 คอนเทนเนอร์.....	19
2.6.4 โมบาย.....	20
2.6.5 เทมพีท.....	21

สารบัญ(ต่อ)

บทที่ 3 อัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้น 2 ระดับ.....	23
3.1 การปรับปรุงอัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน.....	23
3.2 อัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน 2 ระดับ.....	24
บทที่ 4 วิเคราะห์ประสิทธิภาพอัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้น 2 ระดับ.....	27
4.1 จำนวนพิกัดที่ต้องเปรียบเทียบ.....	27
4.2 ความเร็วที่ใช้ในการคำนวณจริง.....	28
4.3 เปรียบเทียบค่าความถูกต้อง.....	28
4.4 กราฟเปรียบเทียบอัตราการเปลี่ยนแปลงความถูกต้อง.....	31
บทที่ 5 การพัฒนางจรประมวลผล.....	34
5.1 สภาพแวดล้อมของระบบที่ใช้ทำการทดสอบ.....	34
5.2 วัตถุประสงค์ในการออกแบบวงจร.....	35
5.3 ลักษณะของข้อมูลที่ใช้ในการประมวลผล.....	36
5.4 โปรโตคอลสื่อสารระหว่างพีซีและเอฟพีจีเอ.....	36
5.5 ขั้นตอนในการประมวลผลเปรียบเทียบบิตของวงจรบนเอฟพีจีเอ.....	38
5.6 การพัฒนาระบบประมวลผลแบบขนาน.....	41
5.7 ข้อมูลสรุปของวงจร.....	43
5.8 ค่าความถูกต้องของการประมวลผลด้วยวงจรดิจิทัล.....	43
บทที่ 6 บทสรุปผลการวิจัยและเสนอแนะ.....	48
บรรณานุกรม.....	50
ภาคผนวก ก. ผลงานวิจัยที่ได้รับการตีพิมพ์.....	51
ภาคผนวก ข. ลำดับภาพมาตรฐานลำดับภาพที่ 1 ถึง 10 ที่ใช้ในงานวิจัย.....	61

สารบัญ(ต่อ)

ประวัติผู้เขียน.....	82
----------------------	----

สารบัญตาราง

ตารางที่	หน้า
2.1 จำนวนพิกัดที่ต้องเปรียบเทียบระหว่าง 2 อัลกอริทึมที่ความละเอียดแตกต่างกัน.....	9
2.2 จำนวนพิกัดที่ต้องเปรียบเทียบระหว่างอัลกอริทึมการค้นหาแบบ 2 ชั้นตอนและการค้นหาแบบเต็มพื้นที่.....	11
2.3 จำนวนพิกัดที่ต้องเปรียบเทียบระหว่างอัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนานและการค้นหาแบบเต็มพื้นที่.....	14
2.4 ค่า PSNR ระหว่างลำดับภาพต่าง ๆ ในลำดับมาตรฐานอาคิโยะ.....	17
2.5 ค่า PSNR ระหว่างลำดับภาพต่าง ๆ ในลำดับมาตรฐานโคสต์การ์ด.....	18
2.6 ค่า PSNR ระหว่างลำดับภาพต่าง ๆ ในลำดับมาตรฐานคอนเทนเนอร์.....	19
2.7 ค่า PSNR ระหว่างลำดับภาพต่าง ๆ ในลำดับมาตรฐานโมบาย.....	20
2.8 ค่า PSNR ระหว่างลำดับภาพต่าง ๆ ในลำดับมาตรฐานเทมพีท.....	21
4.1 จำนวนพิกัดที่ต้องเปรียบเทียบของแต่ละอัลกอริทึมที่ความละเอียดที่แตกต่างกัน.....	27
4.2 เวลาที่ใช้โดยเฉลี่ยของแต่ละอัลกอริทึมที่มีความละเอียดแตกต่างกัน.....	28
4.3 ค่า PSNR ของลำดับภาพอาคิโยะ.....	28
4.4 ค่า PSNR ของลำดับภาพโคสต์การ์ด.....	29
4.5 ค่า PSNR ของลำดับภาพคอนเทนเนอร์.....	29
4.6 ค่า PSNR ของลำดับภาพโมบาย.....	29
4.7 ค่า PSNR ของลำดับภาพเทมพีท.....	29
5.1 โปรโตคอลการส่งข้อมูลขจากพีซีไปยังชิพเอฟพีจีเอ.....	37
5.2 ตัวอย่างข้อมูลจากพีซีไปยังชิพเอฟพีจีเอ.....	38
5.3 ตัวอย่างข้อมูลจากชิพเอฟพีจีเอ ไปยังพีซี.....	38
5.4 จำนวนเกตที่ใช้ไปในการพัฒนางจร.....	43
5.5 ค่า PSNR ระหว่างอัลกอริทึม PHODS ที่เป็นซอฟต์แวร์และฮาร์ดแวร์ของลำดับภาพมาตรฐาน akiyo.....	44
5.6 ค่า PSNR ระหว่างอัลกอริทึม PHODS ที่เป็นซอฟต์แวร์และฮาร์ดแวร์ของลำดับภาพมาตรฐาน coastguard.....	44

สารบัญตาราง(ต่อ)

ตารางที่	หน้า
5.7 ค่า PSNR ระหว่างอัลกอริทึม PHODS ที่เป็นซอฟต์แวร์และฮาร์ดแวร์ของลำดับภาพมาตรฐาน container.....	44
5.8 ค่า PSNR ระหว่างอัลกอริทึม PHODS ที่เป็นซอฟต์แวร์และฮาร์ดแวร์ของลำดับภาพมาตรฐาน mobile.....	44
5.9 ค่า PSNR ระหว่างอัลกอริทึม PHODS ที่เป็นซอฟต์แวร์และฮาร์ดแวร์ของลำดับภาพมาตรฐาน tempete.....	44

สารบัญรูป

รูปที่	หน้า
2.1 ลำดับภาพที่ 1, 2 และ 3 ของภาพวิดีโอมิสอเมริกา.....	6
2.2 การประมาณการเคลื่อนที่ด้วยวิธีเปรียบเทียบกลุ่ม.....	6
2.3 การเปรียบเทียบพิกัดในขั้นตอนที่ 1 ของ 2SS.....	10
2.4 การเปรียบเทียบพิกัดในขั้นตอนที่ 2 ของ 2SS.....	10
2.5 เวกเตอร์การเคลื่อนที่ของ 2SS.....	11
2.6 ขั้นตอนที่ 1 ของวิธีการ PHODS.....	13
2.7 ขั้นตอนที่ 2 ของวิธีการ PHODS.....	13
2.8 เวกเตอร์การเคลื่อนที่ของ PHODS.....	14
2.9 เวกเตอร์ของลำดับภาพมิสอเมริการะหว่างภาพที่ 1 และ 4 ที่แตกต่างกันตามอัลกอริทึม.....	15
2.10 ลำดับภาพที่ 1 ของอาคิโยะ.....	17
2.11 ลำดับภาพที่ 1 ของคอสต์การ์ด.....	18
2.12 ลำดับภาพที่ 1 ของคอนเทนเนอร์.....	19
2.13 ลำดับภาพที่ 1 ของโมบาย.....	20
2.14 ลำดับภาพที่ 1 ของเทมพีท.....	21
2.15 ภาพลำดับที่ 4 ของลำดับภาพโมบายที่สร้างขึ้นจากค่าเวกเตอร์ของ FS.....	22
2.16 ภาพลำดับที่ 4 ของลำดับภาพโมบายที่สร้างขึ้นจากค่าเวกเตอร์ของ PHODS.....	22
3.1 การเปรียบเทียบในระดับแรกของ 2LPHODS.....	24
3.2 ตัวอย่างการกำหนดพื้นที่ค้นหาในระดับที่ 2	25
3.3 ผลลัพธ์ของการค้นหาในพื้นที่ค้นหาในระดับที่ 2	25
3.4 เวกเตอร์การเคลื่อนที่ของอัลกอริทึม 2LPHODS.....	26
4.1 ภาพที่สร้างจากผลลัพธ์ของอัลกอริทึม PHODS.....	30
4.2 ภาพที่สร้างจากผลลัพธ์ของอัลกอริทึม 2LPHODS.....	30
4.3 กราฟแสดงการเปลี่ยนแปลงค่า PSNR ของลำดับภาพ akiyo.....	31
4.4 กราฟแสดงการเปลี่ยนแปลงค่า PSNR ของลำดับภาพ coastguard.....	31
4.5 กราฟแสดงการเปลี่ยนแปลงค่า PSNR ของลำดับภาพ container.....	32
4.6 กราฟแสดงการเปลี่ยนแปลงค่า PSNR ของลำดับภาพ mpbile.....	32

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.7 กราฟแสดงการเปลี่ยนแปลงค่า PSNR ของลำดับภาพ tempete.....	33
5.1 การนำชีพอพพีจีเอมาช่วยประมวลผลผ่านทางบัสพีซีไอ.....	34
5.2 แผนภาพแสดงข้อมูลขนาด 16 บิต.....	37
5.3 สเตทแมทซิน.....	39
5.4 แผนภาพแสดงการประมวลผลแบบขนาน 4 ชุดของ PHODS.....	42
5.5 แผนภาพแสดงการปรับปรุงหน่วยความจำให้เป็นแบบขนาน 4 ชุด.....	42
5.6 กราฟเปรียบเทียบค่า PSNR ของลำดับภาพ akiyo ที่คำนวณโดยซอฟต์แวร์และฮาร์ดแวร์.....	45
5.7 กราฟเปรียบเทียบค่า PSNR ของลำดับภาพ coastguard ที่คำนวณโดยซอฟต์แวร์และฮาร์ดแวร์.....	45
5.8 กราฟเปรียบเทียบค่า PSNR ของลำดับภาพ container ที่คำนวณโดยซอฟต์แวร์และฮาร์ดแวร์.....	46
5.9 กราฟเปรียบเทียบค่า PSNR ของลำดับภาพ mobile ที่คำนวณโดยซอฟต์แวร์และฮาร์ดแวร์.....	46
5.10 กราฟเปรียบเทียบค่า PSNR ของลำดับภาพ tempete ที่คำนวณโดยซอฟต์แวร์และฮาร์ดแวร์.....	47
ข.1 ภาพเคลื่อนไหว akiyo ลำดับภาพที่ 1-3	62
ข.2 ภาพเคลื่อนไหว akiyo ลำดับภาพที่ 4-6	63
ข.3 ภาพเคลื่อนไหว akiyo ลำดับภาพที่ 7-9	64
ข.4 ภาพเคลื่อนไหว akiyo ลำดับภาพที่ 10	65
ข.5 ภาพเคลื่อนไหว coastguard ลำดับภาพที่ 1-3	66
ข.6 ภาพเคลื่อนไหว coastguard ลำดับภาพที่ 4-6	67
ข.7 ภาพเคลื่อนไหว coastguard ลำดับภาพที่ 7-9	68
ข.8 ภาพเคลื่อนไหว coastguard ลำดับภาพที่ 10	69
ข.9 ภาพเคลื่อนไหว container ลำดับภาพที่ 1-3	70
ข.10 ภาพเคลื่อนไหว container ลำดับภาพที่ 4-6	71
ข.11 ภาพเคลื่อนไหว container ลำดับภาพที่ 7-9	72
ข.12 ภาพเคลื่อนไหว container ลำดับภาพที่ 10	73
ข.13 ภาพเคลื่อนไหว mobile ลำดับภาพที่ 1-3	74
ข.14 ภาพเคลื่อนไหว mobile ลำดับภาพที่ 4-6	75
ข.15 ภาพเคลื่อนไหว mobile ลำดับภาพที่ 7-9	76

สารบัญรูป(ต่อ)

รูปที่	หน้า
ข.16 ภาพเคลื่อนไหว mobile ลำดับภาพที่ 10	77
ข.17 ภาพเคลื่อนไหว tempete ลำดับภาพที่ 1-3	78
ข.18 ภาพเคลื่อนไหว tempete ลำดับภาพที่ 4-6	79
ข.19 ภาพเคลื่อนไหว tempete ลำดับภาพที่ 7-9	80
ข.20 ภาพเคลื่อนไหว tempete ลำดับภาพที่ 10	81

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เทคโนโลยีทางการบีบอัดภาพวิดีโอได้มีการพัฒนาไปอย่างมากในปัจจุบัน วิธีการต่างๆ ได้ถูกคิดค้นขึ้นเพื่อลดขนาดของข้อมูลวิดีโอซึ่งเป็นข้อมูลที่มีขนาดใหญ่ให้มีขนาดเล็กที่สุด โดยที่คุณภาพของภาพวิดีโออยู่ในระดับที่ยอมรับได้ เพื่อให้สามารถจัดเก็บลงบนสื่อต่างๆ หรือส่งผ่านเครือข่ายที่มีข้อจำกัดด้านความเร็วของข้อมูล แต่การบีบอัดข้อมูลเพื่อลดขนาดให้ได้ในอัตราส่วนที่สูงนั้นจำเป็นต้องมีการคำนวณที่ซับซ้อน มีขั้นตอนในการลดขนาดหลายขั้นตอน และต้องใช้พื้นที่ในหน่วยความจำเป็นจำนวนมาก ทำให้การบีบอัดข้อมูลภาพวิดีโอในความเร็วที่สูงนั้น ต้องใช้ระบบประมวลผลที่มีทรัพยากรของระบบสูงไปด้วย โดยเฉพาะเมื่อต้องการบีบอัดภาพวิดีโอให้ได้ความเร็วระดับเรียลไทม์ (real time) ด้วยแล้ว อุปกรณ์ประมวลผลขนาดเล็ก เช่น สมาร์ทโฟน พ็อคเก็ตพีซี และพีดีเอ ซึ่งเป็นระบบประมวลผลที่มีทรัพยากรไม่สูงนั้นไม่มีทางทำได้เลย

การประมาณการเคลื่อนที่ (Motion Estimation : ME) เป็นขั้นตอนหนึ่งในกระบวนการบีบอัดภาพเคลื่อนไหวของมาตรฐานการเข้ารหัสวิดีโอหลายมาตรฐานเช่น MPEG-1, MPEG-2, MPEG-4, H.261 และ H.263 ซึ่งในมาตรฐานเหล่านี้ขั้นตอนการประมาณการเคลื่อนที่จะใช้เวลาอยู่ที่ประมาณ 60% ของขั้นตอนทั้งหมด ดังนั้นการปรับปรุงประสิทธิภาพในส่วนนี้จึงส่งผลต่อประสิทธิภาพ และความเร็วของการเข้ารหัสทั้งหมด การพัฒนาหน่วยประมวลผลเฉพาะทางขึ้นมาเพื่อใช้ในการคำนวณค่าการเคลื่อนที่ก็เป็นวิธีหนึ่งที่ถูกนำมาใช้เพื่อเพิ่มความเร็วในการเข้ารหัสวิดีโอ เนื่องจากการประมวลผลด้วยวงจรดิจิทัลที่ออกแบบมาโดยเฉพาะจึงมีความเร็วสูงกว่าการประมวลผลด้วยซอฟต์แวร์ผ่านหน่วยประมวลผลแบบธรรมดา นอกจากการพัฒนาหน่วยประมวลผลเฉพาะทางขึ้นมาแล้ว การเลือกใช้ หรือพัฒนาอัลกอริทึมสำหรับการประมาณการเคลื่อนที่ให้เหมาะสมกับการประมวลผลด้วยวงจรดิจิทัลก็มีส่วนช่วยเพิ่มความเร็วในการประมวลผลด้วยเช่นกัน

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์ฉบับนี้มุ่งหวังเพื่อศึกษาการพัฒนาหน่วยประมวลผลเฉพาะทางสำหรับใช้คำนวณการประมาณการเคลื่อนที่ซึ่งเป็นขั้นตอนหนึ่งที่สำคัญในการเข้ารหัสภาพวิดีโอหลายมาตรฐาน ซึ่งในส่วนของ การประมาณการเคลื่อนที่นั้นเป็นขั้นตอนที่มีการประมวลผลข้อมูลเป็น

จำนวนมาก และใช้เวลานาน การพัฒนาประสิทธิภาพในส่วนนี้จึงส่งผลกระทบต่อประสิทธิภาพโดยรวมของการเข้ารหัสทั้งหมด การเลือกอัลกอริทึมที่ใช้ประมาณการเคลื่อนที่ที่เหมาะสมกับการประมวลผลด้วยวงจรดิจิทัลก็มีส่วนช่วยเพิ่มประสิทธิภาพในด้านความเร็วด้วยเช่นกัน ดังนั้นวิทยานิพนธ์นี้จึงนำเสนอการพัฒนาอัลกอริทึมที่รองรับการประมวลผลแบบขนานซึ่งเป็นจุดเด่นของวงจรดิจิทัล มีการคำนวณที่ไม่ซับซ้อน มีความผิดพลาดของผลลัพธ์ในระดับที่ยอมรับได้ โดยเลือกใช้วิธีปรับปรุงอัลกอริทึมที่มีอยู่แล้ว ซึ่งมีจุดเด่นในแง่ของความเร็วที่สูง แต่ผลลัพธ์มีความคลาดเคลื่อนมากกว่าอัลกอริทึมอื่น ๆ จึงทำให้ไม่ค่อยเป็นที่นิยมใช้ ถ้าหากมีการปรับปรุงในแง่ของความถูกต้องให้มากยิ่งขึ้น โดยที่มีผลกระทบต่อความเร็วของอัลกอริทึมน้อยที่สุด ก็จะได้อัลกอริทึมที่เหมาะสมสำหรับนำไปสร้างเป็นหน่วยประมวลผลเฉพาะ เพื่อนำไปใช้เป็นส่วนหนึ่งในการเข้ารหัสวิดีโอสำหรับอุปกรณ์ประมวลผลที่มีความเร็วต่ำ

1.3 สมมติฐานของการศึกษา

ปัญหาของการเข้ารหัสข้อมูลวิดีโอในปัจจุบันอยู่ที่ความล่าช้าในการคำนวณ เนื่องจากความสามารถในการบีบอัดข้อมูลที่สูง จึงเกิดปัญหาในระบบประมวลผลที่มีความเร็วต่ำ เช่น อุปกรณ์พกพาต่าง ๆ ที่ไม่สามารถเข้ารหัสในความเร็วระดับเรียลไทม์ได้ จึงเกิดปัญหาในการจัดเก็บข้อมูลลงบนสื่อ หรือส่งผ่านไปยังอุปกรณ์ปลายทางผ่านเครือข่าย โดยเฉพาะภาพวิดีโอที่มีความละเอียดสูง ขั้นตอนที่ทำให้เกิดปัญหาความล่าช้าในการเข้ารหัสวิดีโอคือการประมาณการเคลื่อนที่ ซึ่งเป็นขั้นตอนที่ใช้หากการเปลี่ยนแปลงตำแหน่งของส่วนต่าง ๆ ในภาพวิดีโอเมื่อเทียบกับภาพอ้างอิง วิธีการที่ถูกนำมาใช้เพื่อหาความเปลี่ยนแปลงตำแหน่งของภาพวิดีโอ คือ วิธีการเปรียบเทียบแบบกลุ่ม (Block Matching Algorithm : BMA) เป็นการแบ่งภาพวิดีโอให้เป็นภาพขนาดเล็กหลายภาพ จากนั้นจึงนำไปเปรียบเทียบในภาพอ้างอิงเพื่อหาตำแหน่งที่ใกล้เคียงที่สุดโดยใช้วิธีเปรียบเทียบพิกเซลต่อพิกเซล (pixel) จึงเป็นผลให้เกิดขั้นตอนการเปรียบเทียบเป็นจำนวนมากในแต่ละภาพวิดีโอที่นำมาเข้ารหัส ถึงแม้ว่าจะมีการพัฒนาอัลกอริทึมการเปรียบเทียบให้มีตำแหน่งที่เปรียบเทียบจากเดิมที่เปรียบเทียบทุก ๆ ตำแหน่งที่เป็นไปได้ ให้เหลือเพียงบางตำแหน่ง โดยลดความถูกต้องของผลลัพธ์ลง แต่วิธีการดังกล่าวเมื่อมาใช้กับระบบประมวลผลที่ความเร็วต่ำก็ยังไม่สามารถทำให้เข้ารหัสวิดีโอในระดับเรียลไทม์ได้

วิธีการแก้ปัญหาดังกล่าวข้างต้น เราได้นำเสนอการนำหน่วยประมวลผลเฉพาะทางที่พัฒนาขึ้นมาสำหรับการเปรียบเทียบแบบกลุ่ม เพื่อนำไปใช้ในการประมาณการเคลื่อนที่ ซึ่งเป็นการนำความเร็วของการประมวลผลด้วยฮาร์ดแวร์มาใช้ นอกจากนี้ยังมีการพัฒนาอัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนาน 2 ระดับ (Two-Level Parallel Hierarchical One-Dimensional Search Algorithm : 2LPHODS) ที่รองรับการประมวลผลแบบขนาน มีตำแหน่งที่ต้องเปรียบเทียบน้อย

คำนวณตำแหน่งที่ต้องเปรียบเทียบกับคำสั่งคณิตศาสตร์พื้นฐาน จึงมีการคำนวณไม่ซับซ้อน และมีความผิดพลาดที่ไม่มากจนเกินไป ซึ่งทำให้ได้ความเร็วที่สูงและคุณภาพของภาพที่ได้อยู่ในเกณฑ์ดี

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

การประมาณการเคลื่อนที่เป็นขั้นตอนที่ใช้เวลาในการคำนวณที่สูงมากในการเข้ารหัส ข้อมูลวิดีโอ การนำวงจรดิจิทัลเข้ามาช่วยประมวลผลร่วมกับการพัฒนาอัลกอริทึมให้มีความเร็วที่สูงขึ้น มีการคำนวณที่ซับซ้อน และมีรูปแบบการประมวลผลที่เป็นแบบขนานจะช่วยเพิ่มประสิทธิภาพให้สูงขึ้นเป็นอย่างมาก อัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน 2 ระดับ ซึ่งนำเสนอในวิทยานิพนธ์นี้ พัฒนารวมได้เงื่อนไข 4 ข้อ คือ ต้องมีความเร็วสูง ความถูกต้องอยู่ในระดับที่ไม่แย่งกันไป ความซับซ้อนในการคำนวณน้อย และรองรับการประมวลผลแบบขนาน ซึ่งเมื่อนำไปสร้างเป็นหน่วยประมวลผลเฉพาะทาง จะทำให้มีประสิทธิภาพที่สูงในด้านความเร็ว

1.5 ขอบเขตการวิจัย

ในวิทยานิพนธ์ฉบับนี้นำเสนอการนำเอาอัลกอริทึมการเปรียบเทียบกลุ่มแบบขนานมาสร้างเป็นหน่วยประมวลผลเฉพาะทาง เพื่อนำไปใช้ในการประมาณการเคลื่อนที่ของภาพวิดีโอ โดยจะใช้ซีพียูเฟิร์มแวร์ (FPGA) มาสร้างในส่วนหนึ่งของหน่วยประมวลผลสำหรับเปรียบเทียบกลุ่มของภาพแล้วเชื่อมต่อเข้ากับเครื่องคอมพิวเตอร์ผ่านทางการ์ดพีซีไอ (PCI card) ในการทดลองใช้การเปรียบเทียบภาพวิดีโอในลำดับภาพ (frame) ที่แตกต่างกัน เพื่อเปรียบเทียบจำนวนจุดที่ต้องเปรียบเทียบ เวลาที่ใช้ในการคำนวณต่อ 1 ลำดับภาพ และค่าความผิดพลาดของผลลัพธ์ เปรียบเทียบกับอัลกอริทึมที่เป็นที่นิยมตัวอื่น ๆ อัลกอริทึมต้นแบบของอัลกอริทึมที่เสนอในวิทยานิพนธ์ รวมทั้งอัลกอริทึมสำหรับหน่วยประมวลผลเฉพาะทางที่มีผู้เสนอมาน

การประมาณการเคลื่อนที่ที่ใช้ในการทดลองจะใช้เพียงการเปลี่ยนแปลงตำแหน่งเพียงอย่างเดียวเท่านั้น ไม่มีการคำนวณในส่วนของการแก้ไขภาพ เพื่อให้เห็นความแตกต่างของอัลกอริทึมได้เด่นชัดขึ้น

ขนาดของภาพวิดีโอที่ใช้ในวิทยานิพนธ์นี้จะเป็นแบบ CIF (Common Intermediate Format) ซึ่งมีขนาดของภาพอยู่ที่ 352 X 288 พิกเซล และจะการใช้ภาพวิดีโอแบบ QCIF (Quarter Common Intermediate Format) ที่มีขนาดของภาพอยู่ที่ 176 X 144 พิกเซลในบางกรณี

1.6 ขั้นตอนของการศึกษา

วิทยานิพนธ์นี้ได้แบ่งเนื้อหาออกเป็น 6 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความเป็นมาของงานวิจัย ความมุ่งหมายและวัตถุประสงค์ สมมติฐาน ทฤษฎีที่ใช้ ขอบเขตของการวิจัย และขั้นตอนการศึกษา

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในการวิจัย และพื้นฐานของการประมาณการเคลื่อนที่ (motion estimation) ซึ่งประกอบไปด้วย อัลกอริทึมการเปรียบเทียบแบบกลุ่ม (block matching algorithm) คอสต์ฟังก์ชัน (cost function) อัลกอริทึมการเลือกตำแหน่งทั้งแบบเต็มพื้นที่ (full search) และแบบรวดเร็ว (fast search) ความเร็วและประสิทธิภาพ

บทที่ 3 กล่าวถึงวิธีการปรับปรุงอัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนาน ให้มีความถูกต้องของผลลัพธ์มากขึ้น

บทที่ 4 วิเคราะห์ประสิทธิภาพของอัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนาน 2 ระดับ

บทที่ 5 การพัฒนาวงจรประมวลผล

บทที่ 6 เป็นบทสรุปผลการวิจัยและข้อเสนอแนะ

บทที่ 2

อัลกอริทึมที่ใช้ในการประมาณการเคลื่อนที่ (Motion Estimation Algorithm)

ในหัวข้อนี้จะกล่าวถึงทฤษฎีและหลักการพื้นฐานต่าง ๆ ที่เกี่ยวข้องกับการประมาณการเคลื่อนที่ของภาพวิดีโอ ซึ่งเนื้อหาในบทนี้จะกล่าวถึงขั้นตอนในการเปรียบเทียบภาพ 2 ภาพ อัลกอริทึมที่ใช้ในการหาจุดพิกัดที่ต้องเปรียบเทียบแบบต่าง ๆ ฟังก์ชันสำหรับแสดงความแตกต่างของภาพ ค่าผลลัพธ์ที่ได้ และการนำค่าผลลัพธ์กลับมาสร้างเป็นภาพ

2.1 การประมาณการเคลื่อนที่ (Motion Estimation)

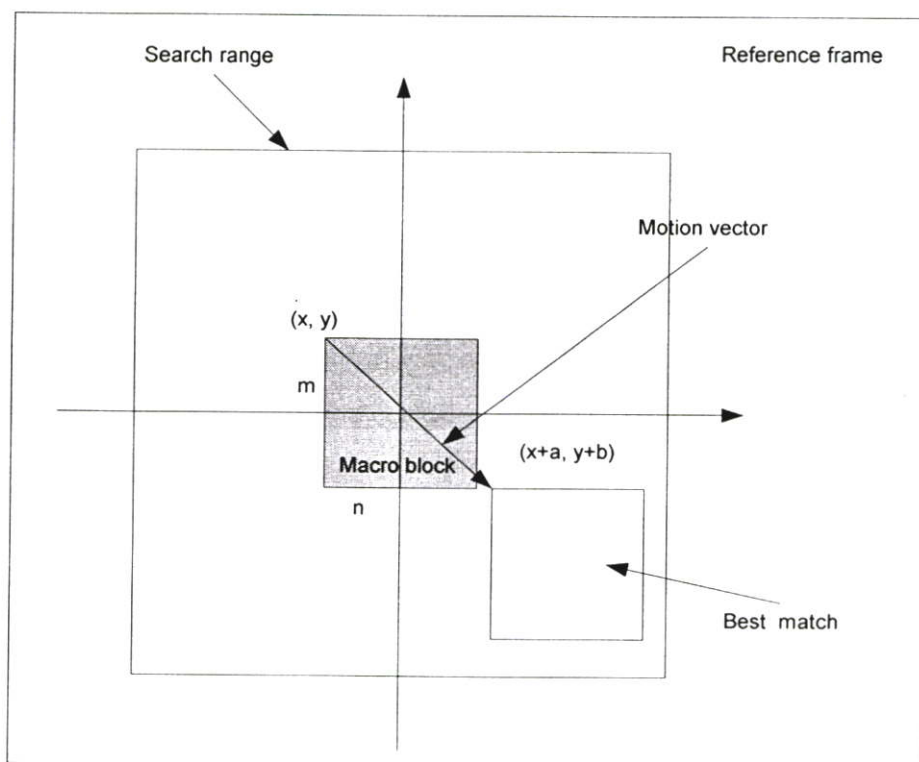
การประมาณการเคลื่อนที่เป็นวิธีการที่ถูกนำมาใช้เพื่อลดขนาดของข้อมูลภาพวิดีโอ โดยใช้พฤติกรรมที่ภาพวิดีโอในลำดับภาพที่ต่อเนื่องกันจะมีการเปลี่ยนแปลงน้อยมาก พื้นที่ส่วนใหญ่ของภาพจะยังเหมือนกับภาพก่อนหน้าหรือภาพถัดไปดังรูปที่ 2.1 จะเห็นได้ว่าในทั้ง 3 ภาพนั้นนอกจากบริเวณใบหน้าแล้ว ส่วนอื่น ๆ ในภาพจะไม่มีเปลี่ยนแปลงใด ๆ จากเหตุผลดังกล่าว จึงได้เกิดแนวคิดในการจัดเก็บภาพวิดีโอทั้งภาพเพียงบางลำดับภาพของวิดีโอ ส่วนภาพวิดีโออื่นนอกเหนือจากนั้นให้จัดเก็บเฉพาะการเปลี่ยนแปลงตำแหน่งภาพในภาพ เมื่อเทียบกับภาพในลำดับใกล้เคียงกันที่มีการจัดเก็บซึ่งเรียกว่าลำดับภาพอ้างอิง ซึ่งเมื่อมีทั้งภาพอ้างอิงและความเปลี่ยนแปลงตำแหน่งในส่วนต่าง ๆ ของภาพ ก็จะสามารถนำมาสร้างเป็นภาพที่ไม่ได้ถูกจัดเก็บไว้ได้ นอกจากการจัดเก็บแบบดังกล่าวแล้วยังมีการจัดเก็บอีกแบบที่ใช้ในมาตรฐานเอ็มเป็ก-1 (MPEG-1) คือจะใช้การจัดเก็บทั้งภาพเป็นบางลำดับภาพ โดยจะมีการเว้นช่วงที่ไม่จัดเก็บเป็นช่วงที่ไม่ยาวมาก เช่น จัดเก็บภาพในลำดับที่ 1 เว้นไป 3 ลำดับ และจัดเก็บใหม่ในลำดับที่ 5 จากนั้นเว้นไปอีก 3 ลำดับ แล้วจึงจัดเก็บลำดับที่ 9 เมื่อต้องการจะถอดรหัสออกมาเป็นลำดับภาพวิดีโอก็ให้คำนวณหาการประมาณการเคลื่อนที่ระหว่างภาพลำดับที่ 1 และ 5 เมื่อได้ทิศทางของการเปลี่ยนแปลงระหว่าง 1 กับ 5 แล้ว ก็สามารถจะนำไปคำนวณหาการเปลี่ยนแปลงระหว่างภาพที่ 1 กับภาพที่ 2 หรือ 3 หรือ 4 แล้วนำมาสร้างเป็นภาพเหล่านั้นได้



รูปที่ 2.1 ลำดับภาพที่ 1, 2 และ 3 ของภาพวิดีโออเมริกา (MISS_AM)

2.2 อัลกอริทึมการเปรียบเทียบทั้งกลุ่ม (Block Matching Algorithm : BMA)

เนื่องจากภาพวิดีโอจะมีลักษณะที่ไม่สามารถแบ่งแยกเป็นวัตถุต่าง ๆ ได้ชัดเจน การจะเปรียบเทียบความเปลี่ยนแปลงของวัตถุต่าง ๆ ระหว่างภาพ 2 ภาพจึงทำได้ยาก จึงต้องใช้วิธีแบ่งภาพออกเป็นภาพย่อย ๆ ขนาดเล็ก (macro block) ที่มีขนาดเท่ากัน และหาการเปลี่ยนแปลงตำแหน่งของภาพย่อย ๆ เหล่านี้แทน วิธีการที่เป็นที่นิยมสำหรับนำมาใช้เพื่อหาการเปลี่ยนแปลงตำแหน่งของภาพย่อยเหล่านี้คือการเปรียบเทียบแบบกลุ่ม ซึ่งเป็นการเปรียบเทียบภาพย่อยแบบภาพต่อภาพ ที่มีขนาดเดียวกัน วิธีการเปรียบเทียบแบบกลุ่มนี้จะใช้ในมาตรฐานการบีบอัดวิดีโอ เช่น เอ็มเป็ก 1 (MPEG-1), เอ็มเป็ก 2 (MPEG-2), เอ็มเป็ก 4 (MPEG-4), เอช 261 (H.261) และ เอช 263 (H.263)



รูปที่ 2.2 การประมาณการเคลื่อนที่ด้วยวิธีเปรียบเทียบกลุ่ม

รูปที่ 2.2 แสดงให้เห็นถึงกระบวนการพื้นฐานของการประมาณการเคลื่อนที่โดยใช้วิธีการเปรียบเทียบกลุ่มโดยจะประกอบไปด้วยลำดับภาพอ้างอิง ซึ่งเป็นลำดับภาพที่อยู่ก่อนหน้า และภาพย่อยขนาด $m \times n$ ซึ่งเป็นส่วนหนึ่งของลำดับภาพในปัจจุบัน ภาพย่อยของลำดับภาพที่แบ่งออกมานี้เรียกว่ามาโครบล็อก (macro block) จุดประสงค์ของการประมาณการเคลื่อนที่ด้วยวิธีนี้ คือ การหาภาพย่อยขนาด $m \times n$ ในลำดับภาพอ้างอิงที่มีความใกล้เคียงกับภาพย่อยขนาด $m \times n$ จากลำดับภาพปัจจุบัน รวมทั้งหาพฤติกรรมการเปลี่ยนแปลงของภาพย่อยนั้นในลำดับภาพปัจจุบัน ซึ่งโดยทั่วไปนั้นจะกำหนดให้ $m = n$ ถ้ากำหนดให้ลำดับภาพปัจจุบันเป็นลำดับภาพที่เวลา t ลำดับภาพอ้างอิงที่นำมาใช้สามารถใช้ได้ทั้งลำดับภาพในช่วงเวลาที่ผ่านมาแล้ว $t - p$ ซึ่งเรียกว่าการประมาณการเคลื่อนที่แบบไปข้างหน้า (forward motion estimation) หรือลำดับภาพในเวลาต่อไปข้างหน้า $t - q$ ซึ่งเรียกว่าการประมาณการเคลื่อนที่แบบย้อนหลัง (backward motion estimation) ก็ได้ เมื่อ p และ q เป็นหน่วยวัดเวลาใด ๆ ทั้งนี้ขึ้นอยู่กับความเหมาะสมของการใช้งาน ซึ่งส่วนมากจะใช้ในลักษณะการประมาณการเคลื่อนที่แบบไปข้างหน้าเสียเป็นส่วนใหญ่ เนื่องจากการบันทึกภาพจากกล้องวิดีโอจะต้องเป็นไปตามลำดับก่อนหลัง จึงไม่สามารถนำลำดับภาพถัดไปมาใช้ได้ นอกจากการเข้ารหัสจากไฟล์ข้อมูลจึงจะสามารถใช้แบบย้อนหลังได้

การอ้างอิงตำแหน่งของภาพย่อยจะใช้พิกัด (x, y) ของมุมบนซ้าย ในการอ้างอิง ในการค้นหาจะต้องมีการกำหนดขอบเขตในการค้นหา (search range) เนื่องจากการค้นหาจากลำดับภาพอ้างอิงทั้งภาพจะใช้เวลามากเกินความจำเป็น เพราะองค์ประกอบต่าง ๆ ในลำดับภาพของวิดีโอ นั้นจะมีการเปลี่ยนแปลงจากตำแหน่งเดิมไม่มากนัก ดังนั้นในการค้นหาจะทำการกำหนดขอบเขตโดยรอบของตำแหน่งพิกัดของภาพย่อยในลำดับภาพปัจจุบัน โดยการกำหนดขอบเขตนั้น จะต้องกำหนดขอบเขตค้นหาในแนวตั้ง V และขอบเขตค้นหาในแนวนอน H ขึ้นมา ซึ่งมาตรฐานการเข้ารหัสวิดีโอส่วนใหญ่ นั้น ค่า V และ H จะกำหนดให้เท่ากัน ขอบเขตการค้นหา นั้นเรียกว่าพื้นที่ค้นหา (search area) ในกรณีของทั่วไปของมาตรฐาน MPEG และ H.261/H.263 นั้นจะกำหนดให้ $V = H = 6$ และ $m = n = 16$ ถ้ากำหนดให้ $(x+a, y+b)$ เป็นพิกัดของภาพย่อยที่มีความใกล้เคียงที่สุดในลำดับภาพอ้างอิง เวกเตอร์จาก (x, y) ไปยัง $(x+a, y+b)$ คือเวกเตอร์การเคลื่อนที่ (motion vector) ของภาพย่อย (x, y) ค่าเวกเตอร์นี้คือสิ่งที่ใช้แสดงการเปลี่ยนแปลงระหว่างลำดับภาพในช่วงเวลาที่แตกต่างกัน ซึ่งเมื่อใช้ค่าเวกเตอร์นี้ร่วมกับลำดับภาพอ้างอิง ก็จะสามารถสร้างภาพที่ใกล้เคียงกับภาพปัจจุบัน

สำหรับวิธีการที่นำมาใช้เพื่อเปรียบเทียบความเหมือนหรือความแตกต่างของภาพย่อยนั้น เรียกว่าคอสต์ฟังก์ชัน (cost function) นิยมใช้วิธีเอ็มเอดี (MAD : Mean Absolute Difference) เป็นส่วนมาก เนื่องจากมี การคำนวณไม่ซับซ้อนมาก ซึ่งมีรูปแบบสมการดังสมการที่ 2.1

$$MAD(dx, dy) = \frac{1}{mn} \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} \sum_{j=-\frac{m}{2}}^{\frac{m}{2}} |F(i, j) - G(i + dx, j + dy)| \quad (2.1)$$

เมื่อ $F(i, j)$ คือ มาโครบล็อกขนาด $m \times n$ ของเฟรมที่กำลังบีบอัด
 $G(i, j)$ คือ มาโครบล็อกขนาด $m \times n$ ของเฟรมอ้างอิง
 (dx, dy) คือ โมชันเวกเตอร์ของพื้นที่ที่กำลังเปรียบเทียบ

ค่า MAD ที่ได้ออกมาแสดงถึงค่าความแตกต่างระหว่างภาพย่อที่นำมาเปรียบเทียบกัน ดังนั้นในการเปรียบเทียบกลุ่มโดยใช้ MAD เป็นคอสม์ฟังก์ชัน ภาพย่อในลำดับภาพอ้างอิงที่นำมาเปรียบเทียบแล้วที่มีค่า MAD น้อยที่สุด หรือมีค่าน้อยกว่าขอบเขตที่กำหนด จะถือว่าเป็นภาพย่อที่มีความใกล้เคียงมากที่สุด

การเปรียบเทียบภาพย่อนั้นใช้วิธีปรับเปลี่ยนค่า dx, dy ไปเป็นค่าพิกัดต่าง ๆ ภายในพื้นที่ค้นหาซึ่งวิธีที่ใช้กันในช่วงแรก ๆ นั้นจะใช้วิธีที่เรียกว่าการค้นหาแบบเต็มพื้นที่ (Full Search : FS) ซึ่งเป็นการค้นหาในทุก ๆ พิกัดตำแหน่งภายในพื้นที่ค้นหาเป็นวิธีที่ให้ผลลัพธ์ที่ถูกต้องที่สุด แต่เนื่องจากการที่ต้องเปรียบเทียบในทุก ๆ พิกัดตำแหน่งภายในพื้นที่ค้นหาจึงทำให้ต้องใช้เวลาในการคำนวณมาก นอกจากนี้การคำนวณค่า MAD ยังเป็นการเปรียบเทียบในลักษณะจุดต่อจุดในทั้งภาพย่อ ซึ่งถือเป็นการคำนวณที่ใช้เวลามากอยู่แล้ว จึงทำให้การคำนวณหาเวกเตอร์การเคลื่อนที่ด้วยวิธีการนี้ไม่เป็นที่นิยม เนื่องจากใช้เวลานาน และใช้ทรัพยากรที่สูง

2.3 อัลกอริทึมการเปรียบเทียบทั้งกลุ่มแบบรวดเร็ว (Fast Block Matching Algorithm : FBMA)

เนื่องจากการเปรียบเทียบพิกัดเพื่อหาเวกเตอร์การเคลื่อนที่ด้วยวิธีค้นหาแบบเต็มพื้นที่นั้นใช้เวลานาน จึงไม่เหมาะสมสำหรับการใช้งานจริง ๆ จึงได้มีการพัฒนาอัลกอริทึมสำหรับค้นหาตำแหน่งของภาพย่อภายในพื้นที่ค้นหาที่มีความใกล้เคียงกับภาพย่อที่นำมาหาเวกเตอร์การเคลื่อนที่ โดยจะใช้วิธีเลือกเปรียบเทียบเป็นบางตำแหน่ง อย่างเป็นขั้นเป็นตอน และใช้ความสัมพันธ์ของผลลัพธ์จากขั้นตอนก่อนหน้าในการเลือกจุดพิกัดที่ต้องทำการเปรียบเทียบในขั้นตอนต่อไป ซึ่งผลลัพธ์ในด้านความถูกต้องนั้นจะมีความคลาดเคลื่อนจากวิธีค้นหาแบบเต็มพื้นที่อยู่ในระดับหนึ่ง ขึ้นอยู่กับวิธีการค้นหาที่นำมาใช้ ตัวอย่างอัลกอริทึมในกลุ่มนี้ที่นิยมใช้กันทั่วไปคือ การค้นหาแบบ 3 ขั้นตอน (Three-Step Search : 3SS) ซึ่งมีจำนวนจุดที่ต้องเปรียบเทียบน้อยกว่าการค้นหาแบบเต็มพื้นที่มาก ดังตารางที่ 2.1

ตารางที่ 2.1 แสดงจำนวนพิกัดที่ต้องเปรียบเทียบระหว่าง 2 อัลกอริทึมที่ความละเอียดที่แตกต่างกัน โดยกำหนดขนาดของภาพย่อย 16 x 16 พิกเซล และระยะค้นหา 8 พิกเซล

อัลกอริทึม \ ขนาดของภาพ	QCIF (176 X 144)	CIF (352 X 288)
Full Search	28611	114444
Three-Step Search	2673	10692

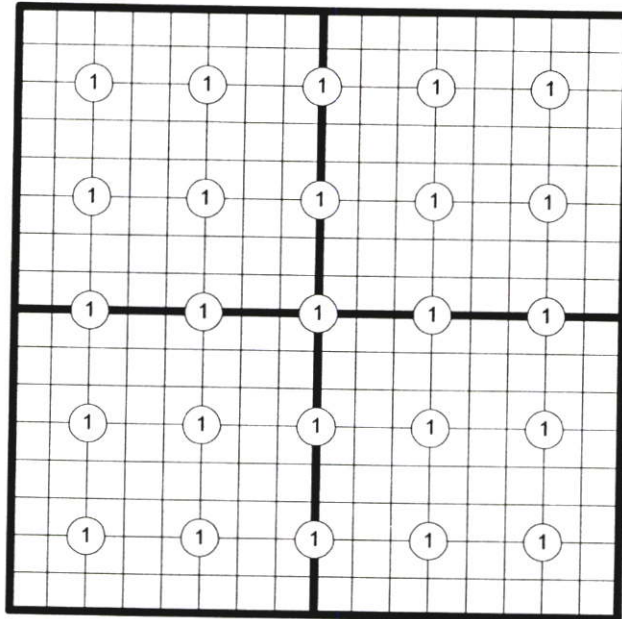
จากตารางที่ 2.1 จะเห็นได้ว่าการค้นหาแบบ 3 ขั้นตอนมีจำนวนพิกัดที่ต้องเปรียบเทียบเพียง 9.34% ของการค้นหาแบบเต็มพื้นที่ ซึ่งทำให้ความเร็วในการประมวลผลการเคลื่อนที่สูงขึ้นเป็นอย่างมาก ด้วยเหตุผลดังกล่าวจึงทำให้เกิดการพัฒนาอัลกอริทึมในกลุ่มนี้ออกมาเป็นจำนวนมาก แต่ละอัลกอริทึมก็มีข้อดี และข้อเสียที่แตกต่างกันไป นอกจากนี้ยังมีการพัฒนาอัลกอริทึมที่เหมาะสมสำหรับสภาพแวดล้อมการทำงานบางอย่างออกมามากด้วย ตัวอย่างเช่น การค้นหาแบบ 2 ขั้นตอนเป็นอัลกอริทึมที่เหมาะสมสำหรับการนำมาสร้างเป็นวงจรดิจิทัล เพื่อนำไปสร้างเป็นหน่วยประมวลผลเฉพาะงานขึ้นมา

2.4 อัลกอริทึมการค้นหาแบบ 2 ขั้นตอน (Two-Step Search : 2SS)

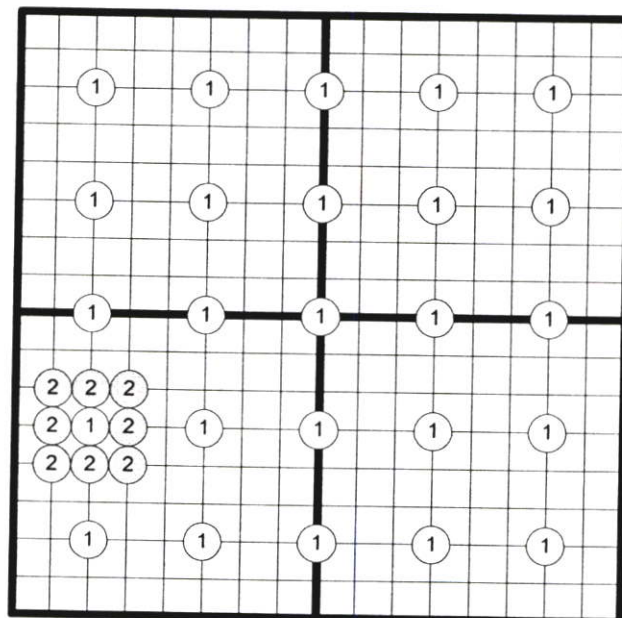
การค้นหาแบบ 2 ขั้นตอนเป็นอัลกอริทึมในกลุ่มของการค้นหาแบบรวดเร็ว จุดมุ่งหมายของการพัฒนาอัลกอริทึมนี้เน้นในด้านของความถูกต้องของผลลัพธ์ ซึ่งมีค่าความถูกต้องสูงใกล้เคียงกับการค้นหาแบบเต็มพื้นที่ และการคำนวณหาตำแหน่งพิกัดที่ต้องเปรียบเทียบในแต่ละขั้นตอนที่ใช้เพียงคณิตศาสตร์พื้นฐานในการคำนวณ หลักการพื้นฐานของอัลกอริทึมนี้คือ การแบ่งการค้นหาออกเป็น 2 ขั้นตอน ขั้นตอนแรกเป็นการค้นหาแบบกระจายทั่วพื้นที่ค้นหา เมื่อได้ค่าที่ดีที่สุดแล้ว จึงทำการค้นหาอย่างละเอียดในขั้นตอนต่อไป ผลลัพธ์ที่ได้จากขั้นตอนที่ 2 คือ เวกเตอร์การเคลื่อนที่ ซึ่งขั้นตอนการทำงานอย่างละเอียดของอัลกอริทึมการค้นหาแบบ 2 ขั้นตอน เมื่อกำหนดให้ระยะค้นหาเท่ากับ 8 พิกเซล มีดังต่อไปนี้

1. กำหนดจุดเริ่มต้น ซึ่งเป็นตำแหน่งเดิมของของภาพย่อยที่นำมาเปรียบเทียบให้เป็นพิกัด (0, 0) จากนั้นทำการหาค่า MAD ของทุก ๆ พิกัด (3a, 3b) เมื่อ a, b มีค่าเป็น -2, -1, 0, 1, 2 หาค่าตำแหน่งพิกัด (x, y) ที่มีค่า MAD น้อยที่สุด ดังรูปที่ 2.3

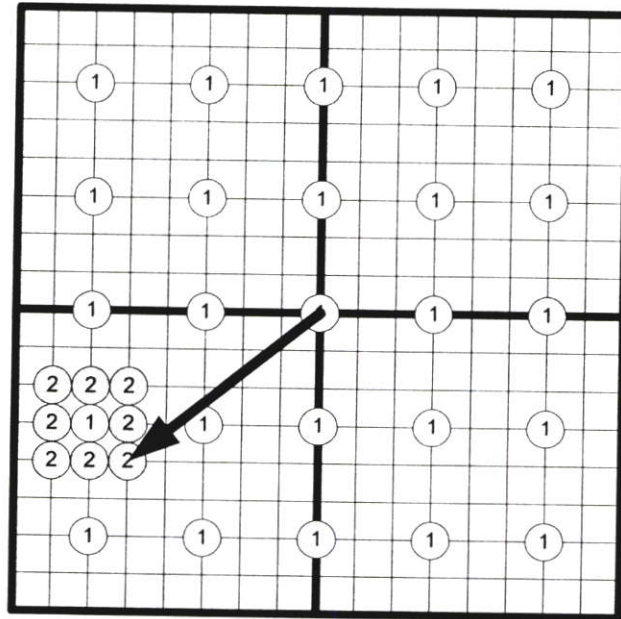
2. จากค่าพิกัด (x, y) ที่ได้จากขั้นตอนที่ 1 ทำการหาค่า MAD ของทุก ๆ ภาพย่อยที่อยู่บนพิกัด (x+a, y+b) เมื่อ a, b มีค่าเป็น -1, 0, 1 ดังรูป 2.4 ซึ่งค่าพิกัด (x+a, y+b) ที่มีค่า MAD น้อยที่สุด คือค่าเวกเตอร์การเคลื่อนที่ของอัลกอริทึมนี้ ดังรูป 2.5



รูปที่ 2.3 การเปรียบเทียบพิกัดในขั้นตอนที่ 1 ของ 2SS



รูปที่ 2.4 การเปรียบเทียบพิกัดในขั้นตอนที่ 2 ของ 2SS



รูปที่ 2.5 เวกเตอร์การเคลื่อนที่ของ 2SS

การปรับเปลี่ยนตำแหน่งพิกัดในแต่ละขั้นตอน สามารถปรับเปลี่ยนได้ตามความเหมาะสมของระยะการค้นหา และการใช้งาน เนื่องจากอัลกอริทึมนี้ไม่ได้มีการกำหนดสมการสำหรับหาตำแหน่งที่แน่นอน ในแต่ละค่าระยะการค้นหา จำนวนพิกัดที่ต้องเปรียบเทียบของอัลกอริทึมนี้ ตามการกำหนดตำแหน่งตามตัวอย่างข้างต้น จะพบว่าจะมีจำนวนพิกัดที่ต้องเปรียบเทียบเป็นจำนวน 11.76% ของการค้นหาแบบเต็มพื้นที่ ดังตารางที่ 2.2

ตารางที่ 2.2 แสดงจำนวนพิกัดที่ต้องเปรียบเทียบระหว่างอัลกอริทึมการค้นหาแบบ 2 ขั้นตอน และการค้นหาแบบเต็มพื้นที่ โดยกำหนดขนาดของภาพย่อย 16 x 16 พิกเซล และระยะค้นหา 8 พิกเซล

อัลกอริทึม \ ขนาดของภาพ	QCIF (176 X 144)	CIF (352 X 288)
Full Search	28611	114444
Two-Step Search	3366	13464

จากลักษณะของอัลกอริทึมที่ใช้การคำนวณหาตำแหน่งแบบพื้นฐาน คือ บวก ลบ และคูณ จึงทำให้มีผู้นำเสนอการนำเอาอัลกอริทึมนี้ไปใช้สร้างเป็นหน่วยประมวลผลเฉพาะทางสำหรับประมวลการเคลื่อนที่ โดยมีจุดมุ่งหมายเพื่อนำไปใช้ในอุปกรณ์มือถือ เพราะการคำนวณที่ไม่ซับซ้อนมาก จึงทำให้วงจรที่ออกมามีขนาดไม่ใหญ่มาก จึงสามารถนำไปประยุกต์ใช้ในอุปกรณ์ขนาดเล็กที่มีข้อจำกัดด้านขนาดและพลังงานได้

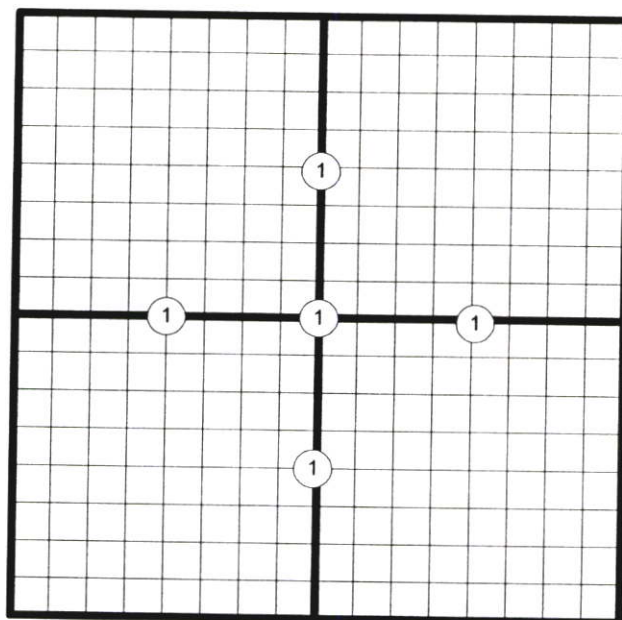
2.5 อัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน (Parallel Hierarchical One-Dimensional Search : PHODS)

อัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน เป็นอัลกอริทึมที่มีแนวคิดแตกต่างจากการหาเวกเตอร์โดยปกติ โดยใช้การค้นหาในแบบมิติเดียว มาหาผลลัพธ์ซึ่งเป็นเวกเตอร์แบบ 2 มิติ เวกเตอร์ที่เป็นผลลัพธ์ของอัลกอริทึมนี้ส่วนใหญ่จะ ไม่มีการเปรียบเทียบค่าคออสต์ฟังก์ชันที่ตำแหน่งดังกล่าวเลย จากเวกเตอร์ 1 มิติ 2 เวกเตอร์เป็นเวกเตอร์ในทิศทางแนวนอน 1 เวกเตอร์ และเป็นเวกเตอร์ในทิศทางแนวตั้ง 1 เวกเตอร์ ถูกนำมารวมกันเพื่อสร้างเป็นเวกเตอร์ 2 มิติในขั้นตอนสุดท้าย ในการขั้นตอนการเปรียบเทียบจะเป็นการคำนวณหาตำแหน่งบนแต่ละแกน x และ y แยกจากกันในลักษณะที่เป็นมิติเดียว ก่อนจะรวมกันเป็นเวกเตอร์ 2 มิติ ตามสมการที่ 2.2

$$(x,0) + (0,y) = (x,y) \quad (2.2)$$

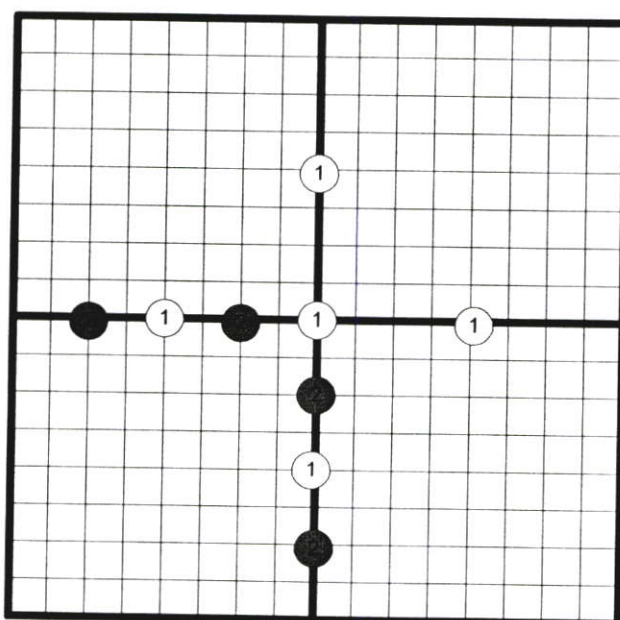
ขั้นตอนในการหาเวกเตอร์การเคลื่อนที่ด้วยวิธีการค้นหา 1 มิติเป็นลำดับชั้นแบบขนานมีดังต่อไปนี้

1. เริ่มต้นจากการคำนวณหาความแตกต่างที่จุดกึ่งกลางของพื้นที่ $(0, 0)$ จากนั้นกำหนดให้เป็นค่าที่ดีที่สุด จากนั้นทำการกำหนดจุดต่อไปที่ต้องเปรียบเทียบ โดยจุดบนแกน y คือ $(0, \frac{P}{2})$ และ $(0, -\frac{P}{2})$ และจุดบนแกน x คือ $(\frac{P}{2}, 0)$ และ $(-\frac{P}{2}, 0)$ จากนั้นคำนวณหาจุดที่ดีที่สุดด้วยคออสต์ฟังก์ชัน ซึ่งในที่นี้เลือกใช้ MAD แล้วทำการเลือกจุดที่ดีที่สุดบนแกน x และแกน y โดยเทียบรวมกับจุดกึ่งกลางด้วย ดังรูปที่ 2.6



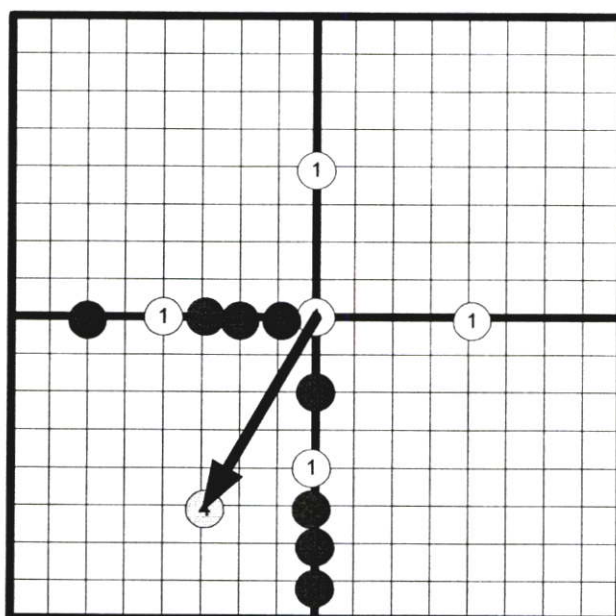
รูปที่ 2.6 แสดงขั้นตอนที่ 1 ของวิธีการ PHODS

2. เมื่อได้จุดที่ดีที่สุดบนทั้ง 2 แกนแล้ว จากนั้นกำหนดจุดต่อไปที่ต้องคำนวณ MAD บนแกน x และ y แกนละ 2 จุด โดยวัดระยะจากจุดที่ดีที่สุดทั้ง 2 จุดออกไปครึ่งหนึ่งของระยะเดิม จะได้เป็นช่วงละ $\frac{P}{4}$ จากนั้นคำนวณหาค่า MAD ของแต่ละจุดบนแกน x และ y จากนั้นเลือกค่าที่ดีที่สุด บนแต่ละแกนจากทั้ง 3 จุด รวมจุดเดิมด้วย จะได้จุดที่ดีที่สุด 2 จุดบนแต่ละแกน ดังรูปที่ 2.7



รูปที่ 2.7 แสดงขั้นตอนที่ 2 ของวิธีการ PHODS

3. กำหนดจุดที่ดีที่สุดบนแต่ละแกนเป็นจุดเริ่มต้น จากนั้นกำหนดจุดเปรียบเทียบบนทั้ง 2 แกนอีกแกนละ 2 จุด โดยลดระยะจากจุดเริ่มต้นไปครึ่งหนึ่ง ซึ่งจะได้เป็น $\frac{P}{8}$ จากนั้นทำซ้ำตามขั้นตอนที่ผ่านมาแล้ว จนกระทั่งระยะเป็น 1 จะถือว่าเป็นรอบสุดท้าย ซึ่งจะได้จุดที่ดีที่สุดบนแกน x และ y นำค่า x จากจุดที่ดีที่สุดเป็นแกน x และค่า y จากจุดที่ดีที่สุดบนแกน y มาจับเป็นคู่ลำดับ ก็จะได้ตำแหน่งที่เหมาะสมที่สุด ซึ่งเป็นเวกเตอร์การเคลื่อนที่ของวิธีการนี้ ดังรูปที่ 2.8



รูปที่ 2.8 เวกเตอร์การเคลื่อนที่ของวิธีการ PHODS

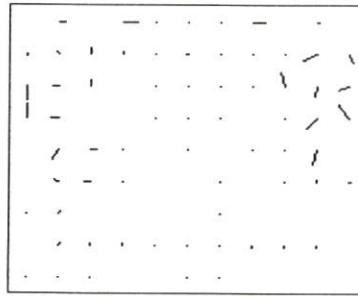
การหาเวกเตอร์การเคลื่อนที่ด้วยวิธีการนี้มีจุดเด่นในด้านของความเร็ว เนื่องจากมีจำนวนพิกัดที่ต้องทำการเปรียบเทียบน้อยมาก คือประมาณ 4.5% ของการค้นหาแบบเต็มพื้นที่ ดังตารางที่ 2.3 ด้วยลักษณะเด่นในด้านความเร็วอันเนื่องมาจากมีจำนวนพิกัดที่ต้องเปรียบเทียบน้อยของอัลกอริทึมนี้ และมีการคำนวณที่ไม่ซับซ้อน รวมทั้งลักษณะของอัลกอริทึมที่มีลักษณะจึงมีความเหมาะสมที่จะนำไปสร้างเป็นวงจรดิจิทัล เพื่อใช้เป็นหน่วยประมวลผลเฉพาะทางสำหรับการประมาณการเคลื่อนที่

ตารางที่ 2.3 แสดงจำนวนพิกัดที่ต้องเปรียบเทียบระหว่างอัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนานและการค้นหาแบบเต็มพื้นที่

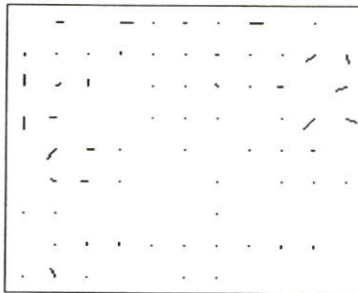
อัลกอริทึม \ ขนาดของภาพ	QCIF (176 X 144)	CIF (352 X 288)
Full Search	28611	114444
PHODS	1287	5148

2.6 การเปรียบเทียบคุณภาพของอัลกอริทึม

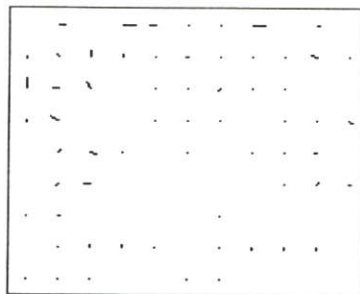
ในการเปรียบเทียบข้อดีและข้อเสียของอัลกอริทึมที่ใช้ในการเลือกตำแหน่งสำหรับเปรียบเทียบภาพย่อขนาดนั้น หลัก ๆ จะมีด้วยกัน 2 ด้าน คือในด้านของความเร็วที่ใช้ในการคำนวณ และในด้านของความคลาดเคลื่อนของผลลัพธ์ ซึ่งเป็นเหตุมาจากการที่วิธีการสุ่มเลือกเปรียบเทียบที่แตกต่างกัน ทำให้ค่าที่ได้แตกต่างกัน ดังรูปที่ 2.9



(FS)



(2SS)



(PHODS)

รูปที่ 2.9 เวกเตอร์ของลำดับภาพ miss_am ระหว่างภาพที่ 1 และ 4 ที่แตกต่างกันตามอัลกอริทึม

การเปรียบเทียบในด้านของความเร็วนั้นมีวิธีวัดที่นิยมใช้ด้วยกัน 2 แบบ คือ การวัดแบบจับเวลาที่ใช้คำนวณจริง และการวัดจากจำนวนพิกัดที่ต้องเปรียบเทียบในพื้นที่ค้นหาขนาดเท่ากัน วิธีการวัดแบบจับเวลาจริงนั้น ค่าเวลาจะเปลี่ยนแปลงไปตามระบบที่ใช้ประมวลผล จึงต้องมีอัลกอริทึมหลักสำหรับใช้อ้างอิง 1 อัลกอริทึม เพื่อใช้ในการอ้างอิงความเร็ว ซึ่งส่วนมากจะใช้

วิธีการค้นหาแบบเต็มพื้นที่ในการอ้างอิง การแสดงค่าของเวลาที่ใช้นั้น อาจจะเป็นเวลาตรง ๆ หรือแสดงเป็นอัตราส่วนเมื่อเทียบกับเวลาที่อัลกอริทึมอ้างอิงใช้ก็ได้

ในส่วนของค่าความถูกต้องนั้นจะใช้อ้างอิงกับวิธีการเปรียบเทียบแบบเต็มพื้นที่ ซึ่งยอมรับกันว่าเป็นวิธีที่ให้ผลลัพธ์เฉลี่ยดีที่สุด สำหรับวิธีการที่นำมาใช้วัดความถูกต้องของผลลัพธ์นั้นจะทำการคำนวณค่า PSNR (Peak Signal to Noise Ratio) ซึ่งมีรูปแบบสมการดังนี้

$$PSNR = 10 \log_{10} \left[\frac{(Peak - to - peak - value - of - the - original - data)^2}{MSE} \right] \quad (2.3)$$

ค่า MSE (Mean Square Error) สามารถคำนวณได้จากสมการ

$$MSE = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M [O(i, j) - R(i, j)]^2 \quad (2.4)$$

เมื่อ $O(i, j)$ คือภาพดั้งเดิม

$R(i, j)$ คือภาพขนาด $N \times M$ ที่สร้างขึ้นใหม่จากค่าเวกเตอร์การเคลื่อนที่ และภาพอ้างอิงที่เป็นภาพที่สมบูรณ์ ไม่สามารถใช้ภาพอ้างอิงที่ถูกสร้างขึ้นมาจากเวกเตอร์มาคำนวณค่าความผิดพลาดได้

ในส่วนของค่า Peak to peak value of the original data นั้นขึ้นอยู่กับค่าสีของพิกเซล เช่น ในกรณีที่เป็นค่า 8 หรือ 24 บิต ค่าที่เป็นไปได้ในแต่ละไบต์ คือ ต่ำสุด 0 และสูงสุด 255 ดังนั้นจึงใช้ค่า 255 ซึ่งเป็นผลต่างของค่าทั้ง 2 ในสมการที่ 2.3 หน่วยวัดของค่า PSNR คือเดซิเบล (dB) อัลกอริทึมที่มีค่าสูงกว่าจะมีความถูกต้องมากกว่าอัลกอริทึมที่มีค่าน้อย

ในการเปรียบเทียบในด้านความถูกต้องนั้นจะต้องใช้การเปรียบเทียบจากข้อมูลวิดีโอที่มีลักษณะการเคลื่อนที่ และการเปลี่ยนแปลงในภาพที่แตกต่างกัน ซึ่งในวิทยานิพนธ์นี้จะใช้ลำดับภาพมาตรฐานจำนวน 5 ชุด เพื่อเปรียบเทียบในด้านของของความถูกต้อง ซึ่งประกอบไปด้วย akiyo, coastguard, container, mobile และ tempete ลำดับภาพทั้งหมดจะเป็นมาตรฐาน CIF เนื่องจากมีขนาดของภาพที่ใหญ่พอสมควร ทำให้เห็นความแตกต่างของตัวเลขได้ชัดเจน

2.6.1 Akiyo

ลำดับภาพ akiyo เป็นลำดับภาพที่มีพื้นหลังนิ่ง มีการเปลี่ยนแปลงระหว่างภาพน้อยมากจะเกิดขึ้นส่วนใหญ่ที่กลางภาพ



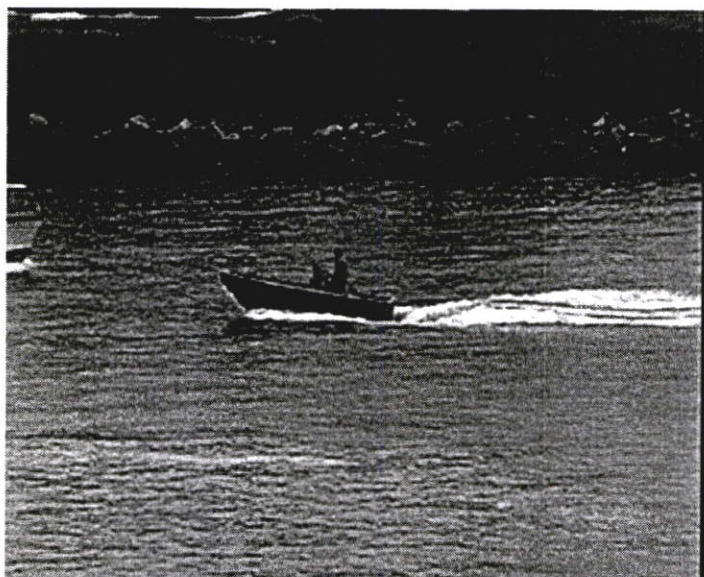
รูปที่ 2.10 ลำดับภาพที่ 1 ของ akiyo

ตารางที่ 2.4 แสดงค่า PSNR ระหว่างลำดับภาพต่าง ๆ ในลำดับมาตรฐาน akiyo

Ref. frame->Recon. frame	FS	3SS	2SS	PHODS
1->2	45.827	45.827	45.827	45.827
3->4	46.296	46.296	46.296	46.296
5->6	40.955	40.955	40.050	39.397
7->8	42.310	42.310	42.310	42.101
1->4	44.614	44.614	44.614	44.614
1->8	38.690	38.302	37.715	37.529

2.6.2 Coastguard

ลำดับภาพ coastguard เป็นลำดับภาพที่มีพื้นหลังเคลื่อนไปทางขวา มีการเปลี่ยนแปลงระหว่างภาพที่ค่อนข้างเร็วในทิศทางที่ไปด้านขวาไปซ้าย



รูปที่ 2.11 ลำดับภาพที่ 1 ของ coastguard

ตารางที่ 2.5 แสดงค่า PSNR ระหว่างลำดับภาพต่าง ๆ ในลำดับมาตรฐาน coastguard

Ref. frame->Recon. frame	FS	3SS	2SS	PHODS
1->2	30.476	30.476	30.441	29.989
3->4	30.690	30.690	30.674	30.599
5->6	30.818	30.818	30.838	30.430
7->8	30.304	30.304	30.273	30.175
1->4	25.503	25.457	25.457	23.993
1->8	22.3066	21.891	21.972	21.311

2.6.3 Container

ลำดับภาพ container เป็นลำดับภาพที่พื้นหลังมีการเปลี่ยนแปลงในลักษณะขยับลงเป็นบางส่วน มีการเปลี่ยนแปลงระหว่างภาพที่ค่อนข้างช้าในทิศทางที่ไปด้านซ้ายไปขวา



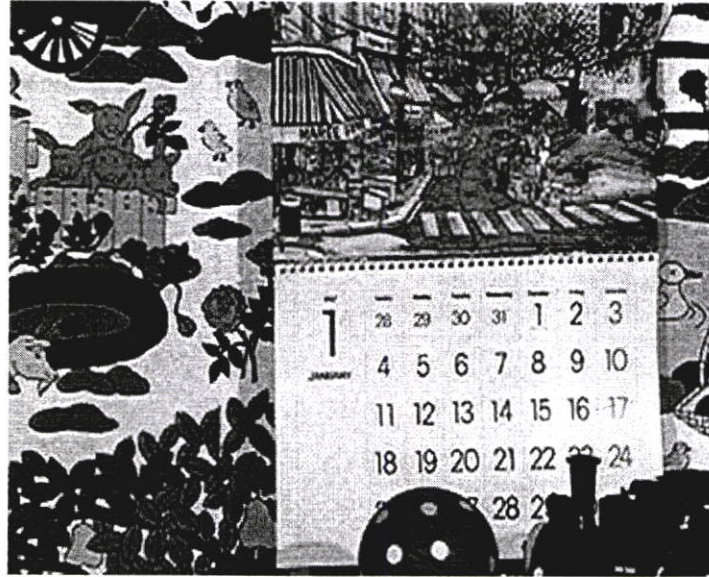
รูปที่ 2.12 ลำดับภาพที่ 1 ของ container

ตารางที่ 2.6 แสดงค่า PSNR ระหว่างลำดับภาพต่าง ๆ ในลำดับมาตรฐาน container

Ref. frame->Recon. frame	FS	3SS	2SS	PHODS
1->2	36.433	36.433	36.433	36.432
3->4	37.396	37.395	37.396	37.395
5->6	36.684	36.683	36.683	36.683
7->8	37.197	37.196	37.196	37.196
1->4	32.877	32.876	32.876	32.876
1->8	30.690	30.602	30.145	30.054

2.6.4 Mobile

ลำดับภาพ mobile เป็นลำดับภาพที่พื้นหลังมีการเปลี่ยนแปลงในลักษณะเลื่อนลงอย่างช้า ๆ มีการเปลี่ยนแปลงระหว่างภาพที่ค่อนข้างเร็วในทิศทางด้านขวาไปซ้ายและเกิดในบริเวณด้านล่างของภาพเป็นส่วนใหญ่



รูปที่ 2.13 ลำดับภาพที่ 1 ของ mobile

ตารางที่ 2.7 แสดงค่า PSNR ระหว่างลำดับภาพต่าง ๆ ในลำดับมาตรฐาน mobile

Ref. frame->Recon. frame	FS	3SS	2SS	PHODS
1->2	22.627	22.189	22.350	20.937
3->4	22.416	22.272	22.282	20.837
5->6	22.406	22.161	22.300	20.684
7->8	22.146	21.996	22.037	20.625
1->4	21.343	18.830	19.332	16.016
1->8	19.217	17.761	17.929	14.075

2.6.5 Tempete

ลำดับภาพ tempete เป็นลำดับภาพที่พื้นหลังมีการเปลี่ยนแปลงในลักษณะที่เป็นการซูมภาพออกอย่างช้า ๆ มีการเปลี่ยนแปลงระหว่างภาพที่ค่อนข้างเร็วในทิศทางจากบนลงล่างและเกิดการกระจายไปทั่วภาพ



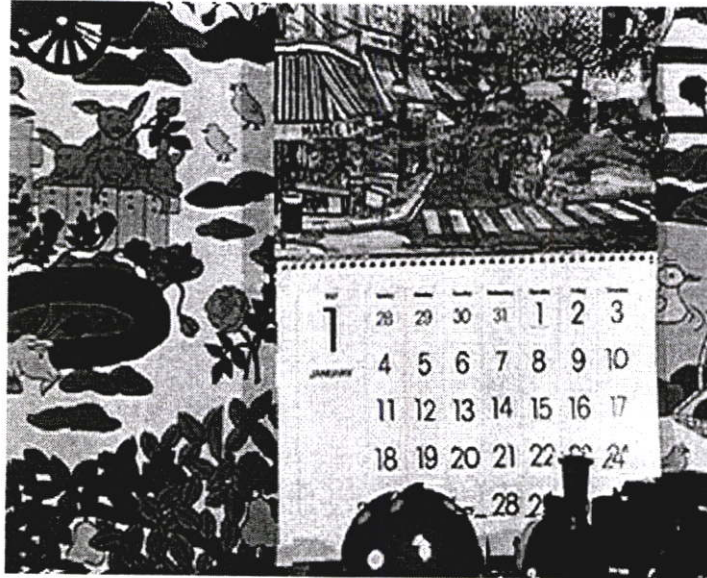
รูปที่ 2.14 ลำดับภาพที่ 1 ของ tempete

ตารางที่ 2.8 แสดงค่า PSNR ระหว่างลำดับภาพต่าง ๆ ในลำดับมาตรฐาน tempete

Ref. frame->Recon. frame	FS	3SS	2SS	PHODS
1->2	26.234	26.142	26.160	25.945
3->4	26.188	26.035	26.116	25.853
5->6	26.124	26.063	26.037	25.860
7->8	26.641	26.603	26.570	26.359
1->4	25.067	24.187	24.319	22.767
1->8	24.162	23.012	22.816	19.514

จากตารางแสดงค่า PSNR ตั้งแต่ตารางที่ 2.4 ถึง 2.8 จะเห็นได้ว่าอัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนานมีค่าความถูกต้องน้อยกว่าอัลกอริทึมอื่น ๆ ที่นำมาเปรียบเทียบอย่างเห็นได้ชัด โดยเฉพาะเมื่อระยะห่างจากลำดับภาพอ้างอิงมีค่ามาก เช่น ลำดับที่ 1 กับ 8 และในกรณีที่ลำดับภาพมีการเปลี่ยนแปลงอย่างรวดเร็วอย่างเช่น ลำดับภาพ mobile ถึงแม้ว่าอัลกอริทึมนี้จะมีความเร็วที่สูงกว่าอัลกอริทึมอื่น ๆ เป็นอย่างมาก แต่ค่าความผิดพลาดที่เกิดขึ้นดังกล่าว ทำให้เกิดปัญหาเมื่อนำไปใช้งานจริง ตัวอย่างดังรูปที่ 2.15 แสดงภาพลำดับที่ 4 ที่สร้างขึ้นจากเวกเตอร์ความเคลื่อนไหวที่

ด้วยวิธีการค้นหาแบบเต็มพื้นที่ โดยมีภาพลำดับที่ 1 เป็นภาพอ้างอิง และรูปที่ 2.16 แสดงภาพลำดับที่ 4 ที่สร้างขึ้นจากเวกเตอร์ของวิธีการค้นหา 1 มิติเป็นลำดับขั้นแบบขนาน ซึ่งจะเห็นได้ว่าคุณภาพของภาพต่างกันอย่างเห็นได้ชัด ซึ่งถ้าหากปรับปรุงในด้านของความถูกต้องได้ โดยกระทบกับความเร็วของอัลกอริทึมไม่มากนักเกินไป และยังคงลักษณะที่เป็นการประมวลผลแบบขนานเอาไว้ได้ อัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนาน ก็จะเป็นอัลกอริทึมที่เหมาะสมสำหรับนำไปใช้สร้างเป็นหน่วยประมวลผลเฉพาะทาง สำหรับไปใช้ในระบบประมวลผลที่มีความเร็วไม่สูงได้



รูปที่ 2.15 ภาพลำดับที่ 4 ของลำดับภาพ mobile ที่สร้างขึ้นจากเวกเตอร์ของ FS



รูปที่ 2.16 ภาพลำดับที่ 4 ของลำดับภาพ mobile ที่สร้างขึ้นจากเวกเตอร์ของ PHODS

บทที่ 3

อัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน 2 ระดับ

(Two-Level Parallel Hierarchical One-Dimensional Search : 2LPHODS)

จากหัวข้อที่ผ่านมาทำให้ทราบถึงลักษณะการทำงาน จุดดีและจุดด้อยของอัลกอริทึมการค้นหาแบบ 2 ชั้นตอนที่มีผู้เสนอก่อนหน้านี้ รวมทั้งลักษณะเด่นของอัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน ในหัวข้อนี้จะกล่าวถึงการนำอัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนานมาปรับปรุงเป็นอัลกอริทึมใหม่ที่มีชื่อว่าอัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน 2 ระดับ

3.1 การปรับปรุงอัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน

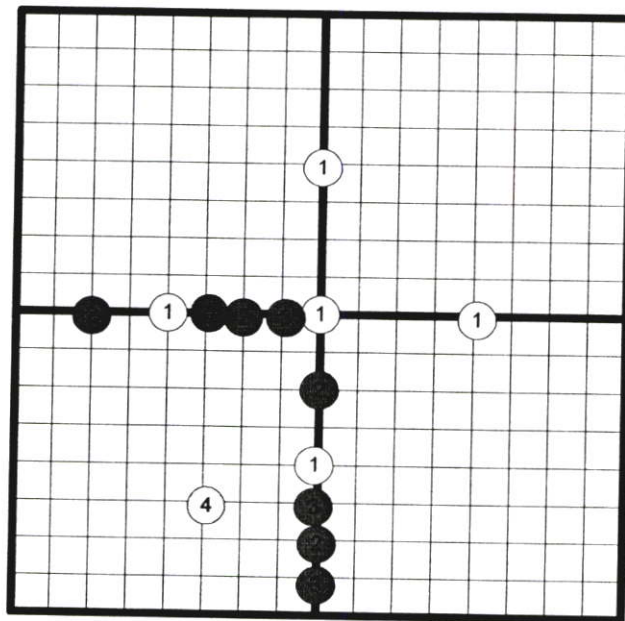
อัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนานนั้น จากหัวข้อที่ผ่านมาทำให้ทราบถึงข้อเสียในด้านของความถูกต้องของผลลัพธ์ของอัลกอริทึมดังกล่าว ซึ่งเมื่อระยะห่างระหว่างลำดับภาพอ้างอิง กับลำดับภาพปัจจุบันมีมาก จะทำให้ค่าความถูกต้องลดลงอย่างเห็นได้ชัดเมื่อเทียบกับอัลกอริทึมอื่นที่นำมาเปรียบเทียบ นอกจากนี้เมื่อภาพมีการเปลี่ยนแปลงที่รวดเร็ว ก็ส่งผลให้ค่าความผิดพลาดของอัลกอริทึมนี้ด้วยเช่นกัน จึงทำให้อัลกอริทึมนี้เมื่อนำไปใช้งานจริงแล้ว จะให้คุณภาพของภาพที่ค่อนข้างต่ำ ถึงแม้ว่าอัลกอริทึมนี้จะมีจุดเด่นในแง่ของความเร็วที่สูงมาก มีพิกัดที่ต้องทำการเปรียบเทียบน้อยมากเพียง 4.5% ของพิกัดที่ต้องทำการเปรียบเทียบด้วยวิธีการค้นหาแบบเดิมพื้นที่

ปัญหาความผิดพลาดดังกล่าว เกิดจากการที่เวกเตอร์การเคลื่อนที่ของอัลกอริทึมนี้แทบจะไม่เคยถูกหาค่าคอสต์ฟังก์ชันของภาพย่อยในพิกัดดังกล่าวเลย วิทยานิพนธ์นี้จึงได้นำเสนอให้มีการเพิ่มการเปรียบเทียบค่าที่พิกัดดังกล่าวและพิกัดใกล้เคียง เพื่อหาจุดพิกัดในที่ให้ผลลัพธ์ที่ดีที่สุดในบริเวณนั้น ซึ่งจะเป็นการลดข้อผิดพลาดของการค้นหาด้วยวิธีนี้

เพื่อให้มีจำนวนพิกัดที่ต้องเปรียบเทียบอีกครั้งเป็นจำนวนน้อยที่สุด และยังคงลักษณะที่เป็นการทำงานแบบขนานเอาไว้ จึงได้เลือกใช้อัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนานในการค้นหาอีกครั้งหนึ่ง ในตำแหน่งเริ่มต้น และระยะค้นหาที่แตกต่างออกไป จึงเรียกได้ว่าเป็นการทำงานแบบ 2 ระดับ

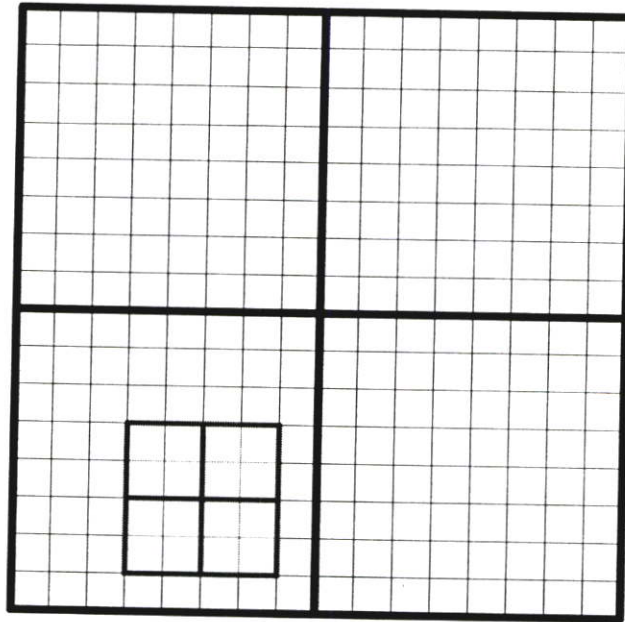
3.2 อัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน 2 ระดับ

จากแนวคิดในการปรับปรุงดังกล่าว ทำให้เกิดอัลกอริทึมสำหรับหาเวกเตอร์การเคลื่อนที่ใหม่ที่มีชื่อว่า อัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน 2 ระดับ (Two-Level Parallel Hierarchical One-Dimensional Search : 2LPHODS) เป็นการหาเวกเตอร์การเคลื่อนที่ด้วยอัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน 2 ครั้งบนพื้นที่ค้นหาที่มีขนาดของระยะค้นหาที่แตกต่างกัน โดยการค้นหาในพื้นที่แรกจะเป็นการค้นหาที่เหมือนกับปกติ คือกำหนดจุดเริ่มต้นที่กึ่งกลางของพื้นที่ค้นหา ซึ่งเป็นตำแหน่งเดียวกันกับ พิกัดของภาพย่อยในลำดับภาพปัจจุบัน จากนั้นทำการเปรียบเทียบภาพย่อย เพื่อหาพิกัด (x, y) ที่ดีที่สุดด้วยวิธีการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน ดังรูปที่ 3.1

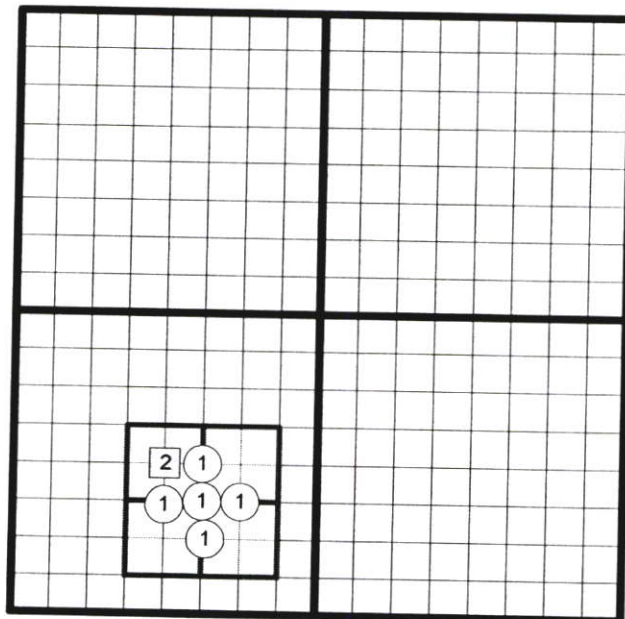


รูปที่ 3.1 การเปรียบเทียบในระดับแรกของ 2LPHODS

นำค่าที่ได้จากการเปรียบเทียบในระดับแรกมาเป็นจุดเริ่มต้นสำหรับการค้นหาในระดับที่ 2 จากนั้นตั้งแกน x' และ y' ขึ้นมาจากจุดดังกล่าว กำหนดขอบเขตการค้นหาของพื้นที่ค้นหาในระดับที่ 2 นี้ ในที่นี้กำหนดให้ระยะค้นหา $P'=2$ ดังรูปที่ 3.2 ทำการเปรียบเทียบพิกัดภายในพื้นที่ค้นหาที่ 2 ด้วยวิธีการค้นหา 1 มิติเป็นลำดับชั้นแบบขนานอีกครั้งจนได้ค่าตำแหน่งที่ดีที่สุดดังรูปที่ 3.3

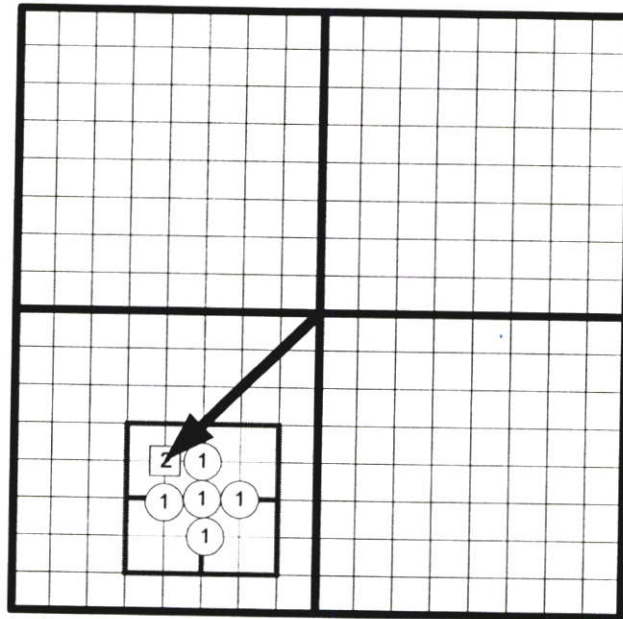


รูปที่ 3.2 ตัวอย่างการกำหนดพื้นที่ค้นหาในระดับที่ 2



รูปที่ 3.3 ผลลัพธ์ของการค้นหาในพื้นที่ค้นหาในระดับที่ 2

เส้นที่ลากจากจุดกึ่งกลางของพื้นที่ค้นหาในระดับที่ 1 ไปยังพิกัดที่ดีที่สุดในพื้นที่ค้นหาในระดับที่ 2 คือเวกเตอร์การเคลื่อนที่ของอัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนาน 2 ระดับ ดังรูปที่ 3.4



รูปที่ 3.4 เวกเตอร์การเคลื่อนที่ของอัลกอริทึม 2LPHODS

บทที่ 4

วิเคราะห์ประสิทธิภาพของอัลกอริทึมการค้นหา 1 มิติเป็น ลำดับชั้นแบบขนาน 2 ระดับ

ในส่วนนี้จะเป็นการกล่าวถึงการเปรียบเทียบผลลัพธ์ที่ได้จากอัลกอริทึมใหม่ เมื่อเทียบกับอัลกอริทึมต้นแบบ อัลกอริทึมการค้นหาแบบเต็มพื้นที่ และการค้นหาแบบ 2 ชั้นตอน โดยจะทำการเปรียบเทียบในด้านของจำนวนพิกัดที่ต้องเปรียบเทียบ เวลาที่ใช้ในการคำนวณบนระบบที่ใช้ทดสอบ และความถูกต้องของผลลัพธ์

4.1 จำนวนพิกัดที่ทำการเปรียบเทียบ

กำหนดขนาดของภาพย่อย 16 x 16 พิกเซล และระยะค้นหา 8 พิกเซล ในส่วนของระยะค้นหาระดับที่ 2 ของอัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน 2 ระดับ กำหนดให้เป็น 2 พิกเซล จำนวนพิกัดที่ต้องทำการเปรียบเทียบของภาพขนาด QCIF และ CIF แสดงไว้ในตารางที่ 4.1 ซึ่งจะเห็นได้ว่าการปรับปรุงอัลกอริทึมส่งผลให้มีจำนวนจุดที่ต้องเปรียบเทียบมากขึ้น แต่เมื่อเทียบกับอัลกอริทึมการค้นหาแบบ 2 ระดับที่ถูกเสนอมาก่อนหน้านั้น อัลกอริทึมที่ปรับปรุงใหม่จะมีจำนวนพิกัดที่ต้องเปรียบเทียบน้อยกว่ามากทีเดียว แสดงให้เห็นว่าอัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน 2 ระดับเป็นอัลกอริทึมที่มีความเร็วขึ้นน้อยกว่าอัลกอริทึมต้นแบบ แต่เร็วกว่าอัลกอริทึมอื่น ๆ ที่นำมาเปรียบเทียบ

ตารางที่ 4.1 แสดงจำนวนพิกัดที่ต้องเปรียบเทียบของแต่ละอัลกอริทึมที่ความละเอียดที่แตกต่างกัน

อัลกอริทึม \ ขนาดของภาพ	QCIF (176 X 144)	CIF (352 X 288)	อัตราส่วน
Full Search	28611	114444	100%
3-Step Search	2673	10692	9.34%
2-Step Search	3366	13464	11.76%
PHODS	1287	5148	4.50%
2LPHODS	2079	8316	7.27%

4.2 ความเร็วที่ใช้ในการคำนวณจริง

กำหนดขนาดของภาพย่อย 16 x 16 พิกเซล และระยะค้นหา 8 พิกเซล ในส่วนของระยะค้นหาระดับที่ 2 ของอัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนาน 2 ระดับ กำหนดให้เป็น 2 พิกเซล เวลาที่ใช้ในการคำนวณเพื่อทำการเปรียบเทียบของภาพขนาด QCIF และ CIF โดยเฉลี่ย แสดงไว้ในตารางที่ 4.2 ซึ่งผลที่ได้ ก็สอดคล้องกับจำนวนจุดพิกัดที่ต้องเปรียบเทียบ คือความเร็วของอัลกอริทึมที่ปรับปรุงขึ้นใหม่จะอยู่ในอันดับที่ 2 รองจากอัลกอริทึมต้นแบบ

ตารางที่ 4.2 แสดงเวลาที่ใช้โดยเฉลี่ยของแต่ละอัลกอริทึมที่ความละเอียดที่แตกต่างกัน หน่วยเป็นวินาที

อัลกอริทึม \ ขนาดของภาพ	QCIF (176 X 144)	CIF (352 X 288)	อัตราส่วน(CIF)
Full Search	0.625	2.453	100%
3-Step Search	0.110	0.422	17.20%
2-Step Search	0.125	0.469	18.78%
PHODS	0.047	0.250	10.19%
2LPHODS	0.078	0.312	12.71%

4.3 เปรียบเทียบค่าความถูกต้อง

กำหนดขนาดของภาพย่อย 16 x 16 พิกเซล และระยะค้นหา 8 พิกเซล ในส่วนของระยะค้นหาระดับที่ 2 ของอัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนาน 2 ระดับ กำหนดให้เป็น 2 พิกเซล ขนาดของภาพวิดีโอจะเป็นมาตรฐาน CIF ทำการเปรียบเทียบใน 5 ลำดับภาพมาตรฐาน ใช้ภาพลำดับที่ 1 เป็นภาพอ้างอิงเทียบกับภาพที่ 2-8 ซึ่งแสดงผลลัพธ์ออกมาใน 5 ตาราง ได้แก่ตารางที่ 4.3 ลำดับภาพ akiyo, ตารางที่ 4.4 ลำดับภาพ coastguard, ตารางที่ 4.5 ลำดับภาพ container, ตารางที่ 4.6 ลำดับภาพ mobile และตารางที่ 4.7 ลำดับภาพ tempete

ตารางที่ 4.3 ค่า PNSR ของลำดับภาพ akiyo

	2	3	4	5	6	7	8
PHODS	45.827	45.317	44.613	41.053	39.397	39.103	41.627
2LPHODS	45.827	45.317	44.613	41.053	39.520	39.218	41.734

ตารางที่ 4.4 ค่า PNSR ของลำดับภาพ coastguard

	2	3	4	5	6	7	8
PHODS	29.989	25.964	23.993	23.056	22.187	21.661	21.311
2LPHODS	29.029	26.118	24.566	23.555	22.717	21.942	21.519

ตารางที่ 4.5 ค่า PNSR ของลำดับภาพ container

	2	3	4	5	6	7	8
PHODS	36.432	31.993	32.876	33.543	32.493	30.389	30.054
2LPHODS	36.433	31.994	32.810	31.029	29.494	28.459	27.801

ตารางที่ 4.6 ค่า PNSR ของลำดับภาพ mobile

	2	3	4	5	6	7	8
PHODS	20.937	18.283	16.016	15.317	14.882	14.299	14.075
2LPHODS	21.729	18.378	17.196	16.869	16.330	15.867	15.335

ตารางที่ 4.7 ค่า PNSR ของลำดับภาพ tempete

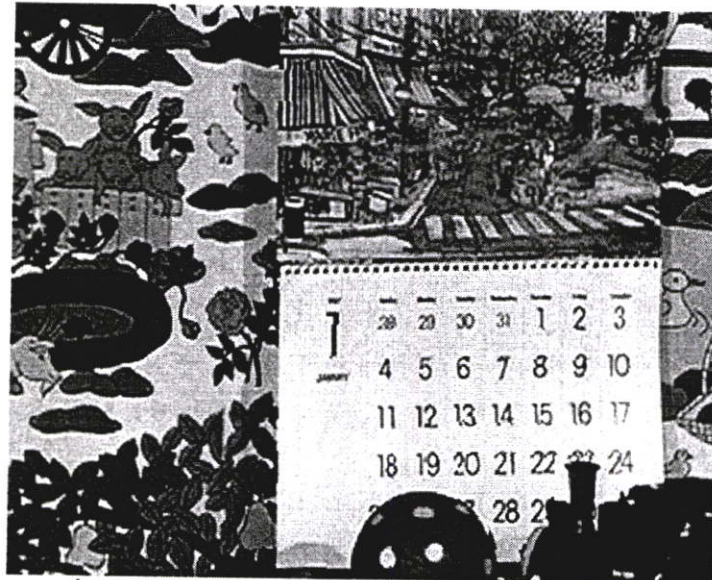
	2	3	4	5	6	7	8
PHODS	25.945	23.700	22.767	21.967	25.872	25.890	24.109
2LPHODS	26.102	23.941	22.896	22.447	25.997	25.928	23.925

จากผลลัพธ์ในตารางที่ 4.3 ถึง 4.7 แสดงให้เห็นว่าค่าความถูกต้องของอัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนาน 2 ระดับ จะสูงกว่าอัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนาน ซึ่งเป็นต้นแบบ ในเกือบทุกชุดของลำดับภาพทดสอบ ยกเว้นในลำดับภาพ container ที่ผลออกมาต่ำกว่าอัลกอริทึมเดิม เมื่อระยะห่างระหว่างลำดับภาพเพิ่มขึ้น ซึ่งจากผลโดยรวมแล้วสรุปได้ว่าอัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนาน 2 ระดับ เป็นอัลกอริทึมที่มีความถูกต้องของผลลัพธ์มากกว่า อัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนานอย่างเห็นได้ชัด

นอกจากการวิเคราะห์เชิงตัวเลขแล้ว ตัวอย่างของภาพที่สร้างขึ้นมาเวคเตอร์การเคลื่อนที่ที่เป็นผลลัพธ์ ดังรูปที่ 4.1 และ 4.2 จะเห็นว่าลูกบอลในรูป 4.2 คู่สมบูรณ์กว่า 4.1 ซึ่งเป็นการบ่งบอกได้อีกอย่างหนึ่งว่าอัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนาน 2 ระดับให้ผลลัพธ์ที่ดีกว่า



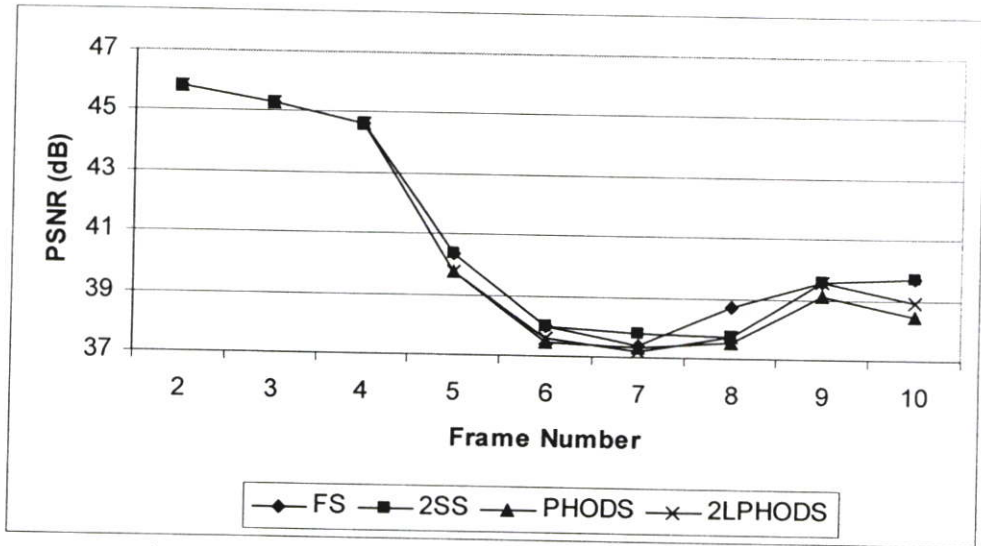
รูปที่ 4.1 ภาพที่สร้างจากผลลัพธ์ของอัลกอริทึม PHODS



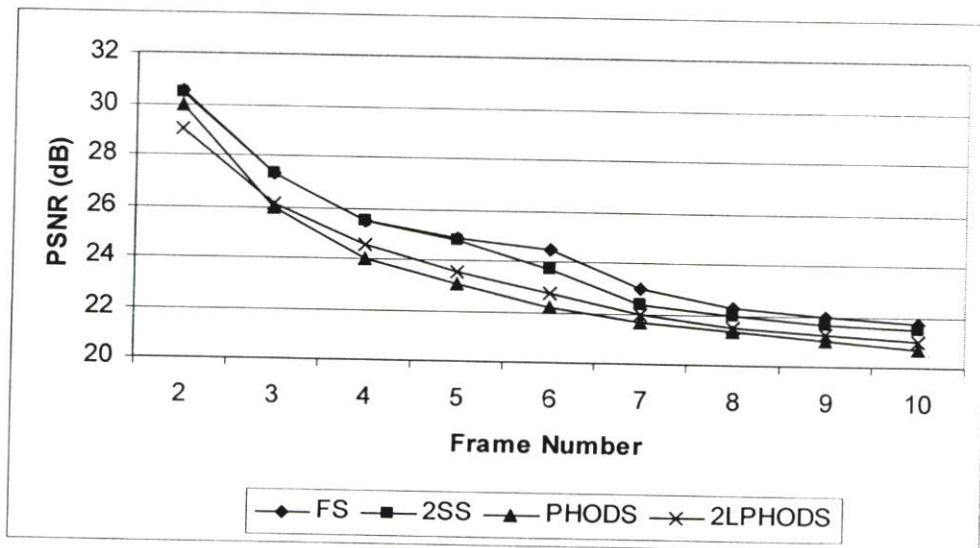
รูปที่ 4.2 ภาพที่สร้างจากผลลัพธ์ของอัลกอริทึม 2LPHODS

4.4 กราฟเปรียบเทียบอัตราการเปลี่ยนความถูกต้อง

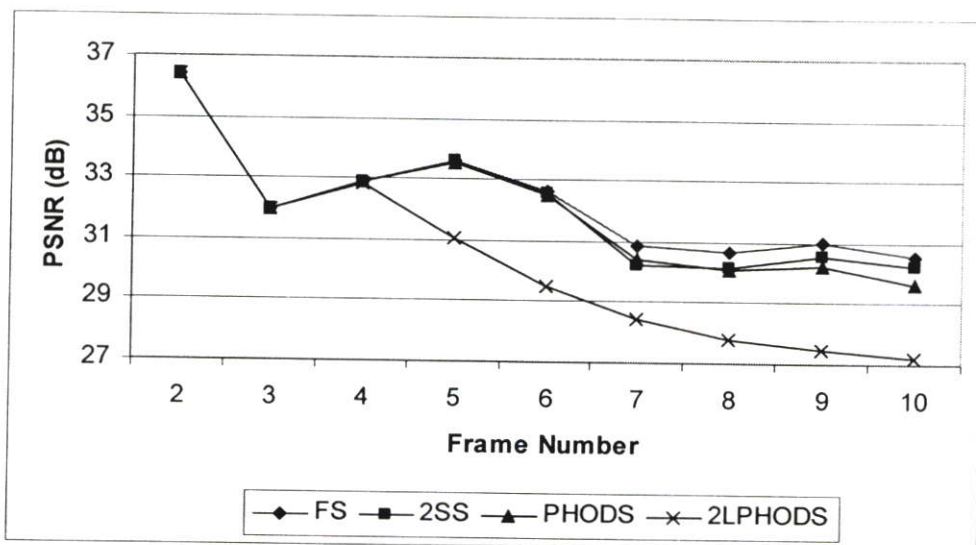
กำหนดให้ลำดับภาพที่ 1 เป็นภาพอ้างอิง จากนั้นหาการเปลี่ยนแปลงการเคลื่อนที่จากลำดับภาพที่ 2 ไปยังลำดับภาพที่ 10 ของลำดับภาพตัวอย่างทั้ง 5 ชุด แล้ววัดค่าความถูกต้องของผลลัพธ์ แล้วนำมาแสดงรูปแบบของกราฟจะทำให้สามารถเห็นความเปลี่ยนแปลงที่แตกต่างกันของแต่ละอัลกอริทึมได้ชัดเจนยิ่งขึ้น ดังรูปที่ 4.3 ถึง 4.7



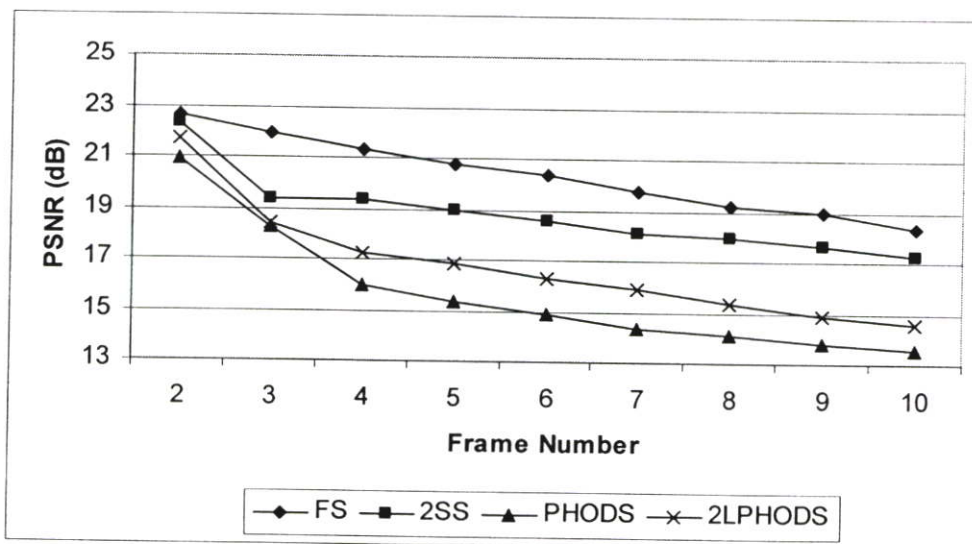
รูปที่ 4.3 กราฟแสดงการเปลี่ยนแปลงค่า PSNR ของลำดับภาพ akiyo



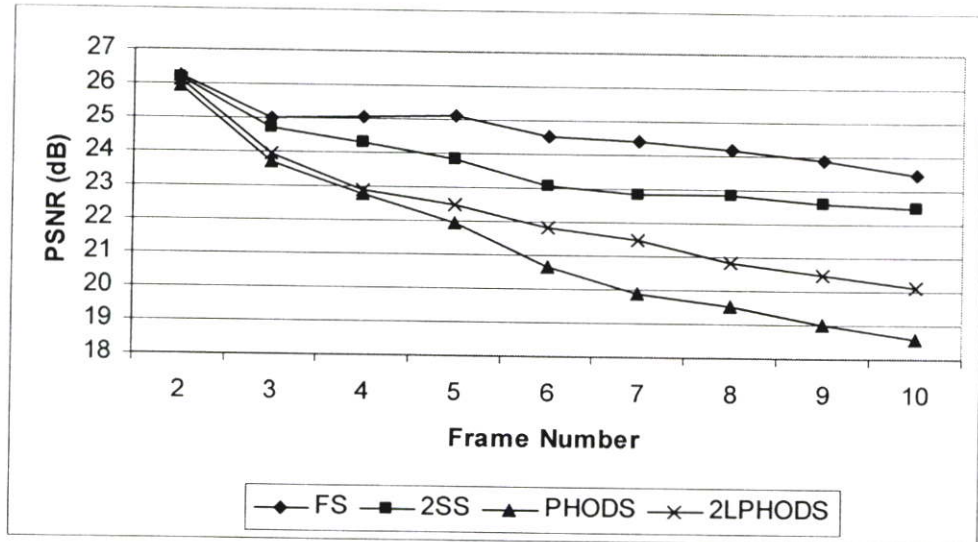
รูปที่ 4.4 กราฟแสดงการเปลี่ยนแปลงค่า PSNR ของลำดับภาพ coastguard



รูปที่ 4.5 กราฟแสดงการเปลี่ยนแปลงค่า PSNR ของลำดับภาพ container



รูปที่ 4.6 กราฟแสดงการเปลี่ยนแปลงค่า PSNR ของลำดับภาพ mobile



รูปที่ 4.7 กราฟแสดงการเปลี่ยนแปลงค่า PSNR ของลำดับภาพ tempete

เมื่อพิจารณาการเปลี่ยนแปลงจากกราฟตัวอย่างทั้ง 5 กราฟ จะเห็นได้ว่าอัลกอริทึมที่ปรับปรุงขึ้นมา ให้ผลลัพธ์ที่ดีกว่าอัลกอริทึมดั้งเดิมในหลาย ๆ ตัวอย่าง แต่ก็ยังมีบางกรณีที่อัลกอริทึมนี้ยังไม่ให้ผลที่ดีเท่าที่ควร เนื่องจากลักษณะของอัลกอริทึมที่เป็นการประมาณค่า จึงทำให้มีโอกาสที่จะเกิดผลลัพธ์ที่ผิดพลาดขึ้นมาได้ดังกราฟในรูปที่ 4.5 แต่เมื่อพิจารณาภาพรวมแล้ว อัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนาน 2 ระดับ จัดเป็นอัลกอริทึมให้ใช้งานได้ดีในระดับหนึ่ง

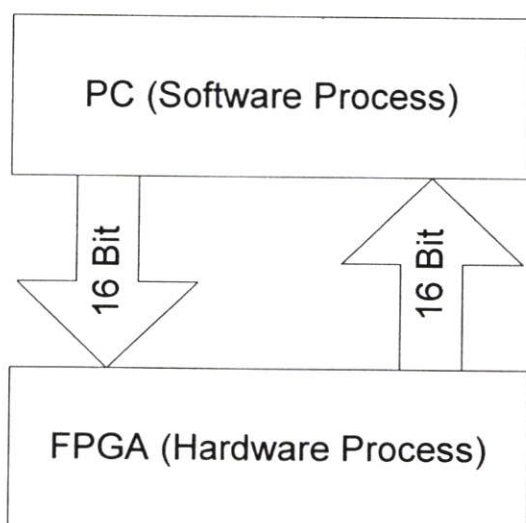
บทที่ 5

การพัฒนางจรประมวลผล

ในหัวข้อนี้จะเป็นการนำเสนอวิธีการนำอัลกอริทึมที่พัฒนาขึ้นมา มาสร้างเป็นวงจรประมวลผลด้วยวงจรถติคัล โดยเครื่องมือที่ใช้ในการพัฒนางจรดังกล่าวคือ ซิฟเอฟพีจีเอ (FPGA) ซึ่งเป็นไอซีที่สามารถจะปรับเปลี่ยนโครงสร้างภายในให้สามารถประมวลผลได้ตามต้องการ

5.1 สภาพแวดล้อมของระบบที่ใช้ทำในการทดสอบ

ในการออกแบบวงจรถติคัลบนเอฟพีจีเอนั้นจำเป็นต้องมีการกำหนดสภาพแวดล้อมของระบบที่ใช้ในการพัฒนาให้เรียบร้อยเสียก่อน จึงจะทำการออกแบบแบบวงจรถติคัลให้เข้ากับระบบทดสอบที่กำหนดไว้ เนื่องจากการประมวลผลด้วยวงจรถติคัลนั้นมีความยืดหยุ่นน้อย และมักจะออกแบบมาเพื่อให้รองรับกับระบบใดระบบหนึ่งเท่านั้น โดยระบบที่ใช้ในการพัฒนาอัลกอริทึมนี้จะใช้ซิฟเอฟพีจีเอมาเชื่อมต่อเข้ากับพีซี ผ่านทางบัสพีซีไอ (PCI Bus) เนื่องจากว่าอินเตอร์เฟซของพีซีไอเป็นแบบขนานทำให้ง่ายต่อการออกแบบวงจรถติคัล ซึ่งในการทดลองนี้จะใช้บัสข้อมูลขนาด 16 บิตในการเชื่อมต่อระหว่างพีซี และซิฟเอฟพีจีเอ ดังรูปที่ 5.1



รูปที่ 5.1 การนำซิฟเอฟพีจีเอมาช่วยประมวลผลผ่านทางบัสพีซีไอ

ในส่วนของซิฟเอฟพีจีเอนั้นจะใช้ชิพตระกูล Spartan-3 ของบริษัท Xilinx เบอร์ XC3S200 ซึ่งมีความจุวงจรถติคัลภายในอยู่ที่ 200,000 เกต ซึ่งเหมาะสำหรับใช้ในการออกแบบวงจรถติคัลขนาดใหญ่ นอกจากจำนวนเกตที่มากแล้ว ไอซีเบอร์ดังกล่าวยังมีหน่วยความจำแบบแรม (RAM) ภายในชิพซึ่ง

มีความจุรวมมากถึง 216Kb ซึ่งสามารถนำไปประยุกต์ใช้ในการประมวลผลข้อมูลขนาดใหญ่ เช่น รูปภาพ ภาพเคลื่อนไหว และข้อมูลเสียง โดยที่การประมวลผลโดยเก็บข้อมูลไว้ในหน่วยความจำภาพในชิพนั้นจะช่วยเพิ่มความเร็วในการประมวลผลได้อย่างมาก เนื่องจากการลดจำนวนครั้งในการดึงข้อมูลจากภายนอกเข้ามาประมวลผล

5.2 วัตถุประสงค์ในการออกแบบวงจร

หลังจากที่กำหนดสภาพแวดล้อมในการพัฒนาชิพประมวลผลสำหรับการประมวลผลเคลื่อนที่เสร็จเรียบร้อยแล้ว ในขั้นตอนต่อมาจะเป็นการกำหนดวัตถุประสงค์และเงื่อนไขที่ใช้ในการออกแบบ

1. วงจรจะต้องเป็นวงจรที่มีการประมวลผลแบบขนาน 4 ชุด เนื่องจากอัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน และ อัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน 2 ระดับ มีจุดเด่นในด้านการประมวลผลแบบขนาน โดยสามารถแยกการประมวลผลในแกน X และ แกน Y แยกออกจากกันได้อย่างอิสระ ในขณะเดียวกัน ในแต่ละแกนจะมีการประมวลผลทั้งในทิศทางลบและทิศทางบวก ซึ่งสามารถที่จะคำนวณทั้ง 2 ทิศทางไปพร้อมกันได้ จึงทำให้เกิดเป็นวงจรประมวลผลขนานกัน 4 ชุด ซึ่งจะทำได้ความเร็วในการประมวลผลสูงกว่าอัลกอริทึมแบบอื่น ๆ ถึง 4 เท่า
2. วงจรที่ออกแบบจะทำการประมวลผลในส่วนของการเปรียบเทียบบล็อก (Block matching) เท่านั้น เพื่อไม่ให้วงจรมีขนาดใหญ่เกินไปจนไม่สามารถจะบรรจุลงในชิพเอฟพีจีเอได้
3. ในการประมวลผลจะต้องทำการโหลดข้อมูลพื้นที่ค้นหา (Search area) และบล็อกย่อย (Macro block) เข้าไปในหน่วยความจำภายในตัวชิพเอฟพีจีเอ เพื่อความรวดเร็วในการประมวลผล โดยหน่วยความจำทั้ง 2 จะแยกจากกัน
4. ภาพที่ใช้ในการประมวลผลจะใช้ภาพที่มีความละเอียดสี 8 บิต เพื่อให้สามารถบรรจุภาพ 2 ภาพลงในหน่วยความจำภายในของชิพเอฟพีจีเอได้
5. ผลลัพธ์ที่ได้จะเป็นค่าพิกัด (X, Y) ซึ่งแสดงถึงค่าเวกเตอร์การเคลื่อนที่ (Motion Vector) ของบล็อกย่อยภายในพื้นที่ค้นหา
6. ในการพัฒนางจรประมวลผลนั้นจะพัฒนาขึ้นมาเพื่อให้ใช้กับอัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน ซึ่งสามารถนำไปประยุกต์ใช้สำหรับ อัลกอริทึมการค้นหา 1 มิติเป็นลำดับชั้นแบบขนาน 2 ระดับได้อีกด้วย

5.3 ลักษณะของข้อมูลที่ใช้ในการประมวลผล

เนื่องจากการประมวลผลด้วยวงจรดิจิทัลนั้นมีความยืดหยุ่นน้อยจึงต้องมีการระบุลักษณะของข้อมูลที่ใช้ประมวลผลให้ชัดเจน ดังนี้

1. รูปภาพที่ใช้ประมวลผลมีด้วยกัน 2 ขนาดคือ 16x16 พิกเซล และ 32x32 พิกเซล
2. รูปภาพใช้ความละเอียดของสี 8 บิต เพื่อลดขนาดของข้อมูลที่ต้องจัดเก็บในวงจร
3. ใช้ข้อมูลระบุพิกัด 2 ขนาดคือ 8 บิตสำหรับภาพขนาด 16x16 พิกเซล โดย 4 บิตแรกตั้งแต่บิตที่ 7 ถึงบิตที่ 4 เก็บค่า x และ 4 บิตหลังตั้งแต่บิตที่ 3 ถึงบิตที่ 0 เก็บค่า y และข้อมูลขนาด 10 บิตสำหรับภาพขนาด 32x32 พิกเซล โดยใช้ 5 บิตแรกตั้งแต่บิตที่ 9 ถึงบิตที่ 5 ในการเก็บค่า x และใช้ 5 บิตหลังตั้งแต่บิตที่ 4 ถึงบิตที่ 0 ในการเก็บค่า y
4. การจัดเก็บในหน่วยความจำจะใช้ค่า x และ y ที่นำมาต่อกันตามข้อที่ 3 มาใช้เป็นแอดเดรสอ้างอิงในหน่วยความจำ

5.4 โพรโทคอลสื่อสารระหว่างพีซีและเอฟพีจีเอ

ในการนำเอฟพีจีเอเข้ามาช่วยประมวลผลนั้นจำเป็นต้องกำหนดรูปแบบการส่งข้อมูลระหว่างกันให้ชัดเจนเพื่อให้ได้ผลลัพธ์ที่ถูกต้อง ในที่นี้ข้อมูลที่จะส่งระหว่างพีซีและเอฟพีจีเอมีขนาดข้อมูล 16 บิต โดยข้อมูลที่ส่งจากพีซีไปยังเอฟพีจีเอจะประกอบไปด้วย

1. ข้อมูลพื้นที่ค้นหา (Search Area) ที่มีขนาดข้อมูล 8 บิต กับแอดเดรส 10 บิต จำนวน 1024 ชุด
2. บล็อกย่อย (Macro block) ที่มีขนาดข้อมูล 8 บิต กับแอดเดรส 8 บิต จำนวน 256 ชุด
3. ข้อมูลที่ส่งจากเอฟพีจีเอมายังพีซีจะเป็นค่าพิกัด X, Y โดยจะเป็นค่า X จำนวน 9 บิต ค่า Y จำนวน 9 บิต ซึ่งสามารถไม่ส่งกลับมาได้ในชุดข้อมูล 16 บิตในครั้งเดียว จึงต้องมีการอ่านค่าผลลัพธ์ 2 ครั้ง โดยเริ่มส่งจากค่า X ก่อน แล้วจึงตามด้วยค่า Y

ในส่วนของการออกแบบโปรโตคอลของข้อมูล จึงมีความจำเป็นทั้งในส่วนของการส่งข้อมูลจากพีซีมายังเอฟพีจีเอ และการส่งข้อมูลจาก เอฟพีจีเอกลับมายังพีซี โดยเฉพาะในส่งของการส่งข้อมูลจากพีซีมายังเอฟพีจีเอ เนื่องจากข้อมูลมี 2 กลุ่ม และมีขนาดของแอดเดรสที่ไม่เท่ากัน จึงต้องมีการกำหนดรูปแบบข้อมูลเพื่อให้แยกแยะได้ว่า ข้อมูลใดเป็นของพื้นที่ค้นหา และข้อมูลใดเป็นของบล็อกย่อย นอกจากนี้ข้อมูลของพื้นที่ค้นหายังมีแอดเดรสขนาด 10 บิต ซึ่งเมื่อรวมกับข้อมูลที่มีขนาด 8 บิตแล้ว ทำให้ไม่สามารถส่งผ่านบัสข้อมูลขนาด 16 บิตได้ ซึ่งหมายความว่า จะต้องใช้การส่งข้อมูล 2 ชุด ต่อข้อมูลภาพ 1 ตำแหน่ง จากการวิเคราะห์ข้อมูลทำให้ได้รูปแบบข้อมูลดังรูปที่ 5.2 และตารางที่ 5.1

MSB											LSB				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

รูปที่ 5.2 แผนภาพแสดงข้อมูลขนาด 16 บิต

ตารางที่ 5.1 โพรโตคอลการส่งข้อมูลจากพีซีไปยังชิพเอฟพีจีเอ

ตำแหน่งของบิต	ความหมาย
15	RST (0 = Normal, 1 = Reset)
14	SYN (0 = Address, 1 = Data)
13	BLK (0 = Search area data, 1=Macro block)
12	END (0 = Normal, 1 = Load data complete)
9-0	Search area address
7-0	Macro block address
7-0	Pixel Data

การส่งข้อมูลจะเริ่มจากการส่งชุดแอดเดรสก่อนแล้วจึงตามด้วยชุดข้อมูล โดยจะใช้บิตที่ 14 เป็นตำแหน่งที่ใช้บอกว่าข้อมูลที่ส่งมาเป็นแอดเดรสของข้อมูลหรือว่าเป็นข้อมูลสีของภาพ ซึ่งหมายความว่าข้อมูลบิตที่ 14 จะมีค่าเป็น 0 และ 1 สลับกันไประหว่างที่มีการส่งข้อมูล นอกจากนี้ยังใช้บิตที่ 13 ในการระบุว่าข้อมูลที่ส่งมาเป็นข้อมูลของพื้นที่ค้นหาหรือว่าเป็นของบล็อกย่อย เพื่อให้สามารถนำข้อมูลดังกล่าวแยกเก็บไปยังหน่วยความจำของแต่ละส่วนได้อย่างถูกต้อง และยังใช้ในการระบุขนาดของแอดเดรสของข้อมูลด้วยว่าเป็นแอดเดรสขนาด 10 บิต หรือ 8 บิต โดยจะเริ่มนับจากบิต 0 เป็น LSB ของทั้งข้อมูลที่เป็นแอดเดรสและข้อมูลของภาพ นอกจากนี้ในส่วนที่เกี่ยวข้องกับการส่งข้อมูลแล้ว ยังมีส่วนของบิตควบคุม เพื่อใช้ในการควบคุมการทำงานของวงจรมอนิเตอร์พีจีเอ ซึ่งมีด้วยกันอยู่ 2 ค่าคือบิตที่ 15 ใช้ในการกำหนดค่าเริ่มต้นของการประมวลผล ซึ่งจะต้องมีการใช้ทุกครั้งก่อนที่จะมีการส่งข้อมูลมายังระบบเพื่อล้างข้อมูลเดิมที่มีอยู่ในวงจรมอนิเตอร์พีจีเอในรอบก่อนหน้า รวมทั้งยังเป็นการแจ้งให้วงจรมอนิเตอร์พีจีเอทราบว่ากำลังจะมีการส่งข้อมูลจากพีซี และข้อมูลบิตที่ 12 เป็นสัญญาณที่ใช้ในการแจ้งกับระบบว่าข้อมูลของภาพที่เป็นพื้นที่ค้นหา และภาพที่เป็นบล็อกย่อยได้ถูกจัดส่งมายังหน่วยความจำภายในของวงจรมอนิเตอร์พีจีเอครบแล้ว ให้เริ่มการประมวลผลได้ ในส่วนของข้อมูลบิตที่ 10 และ 11 นั้นเป็นส่วนที่ไม่มีการใช้งาน ดังตัวอย่างในตารางที่ 5.2

ตารางที่ 5.2 ตัวอย่างข้อมูลจากพีซีไปยังชิพเอฟพีจีเอ

ข้อมูล	แอดเดรส	ข้อมูล	ชนิดของภาพ	คำสั่ง
1000000000000000	-	-	-	รีเซต
0000001111111111	1111111111	-	พื้นที่ค้นหา	-
0100000010101010	-	10101010	พื้นที่ค้นหา	-
0010000000000001	00000001	-	บล็อกย่อย	-
0110000000000000	-	00000000	บล็อกย่อย	-
0001000000000000	-	-	-	สิ้นสุดการส่ง

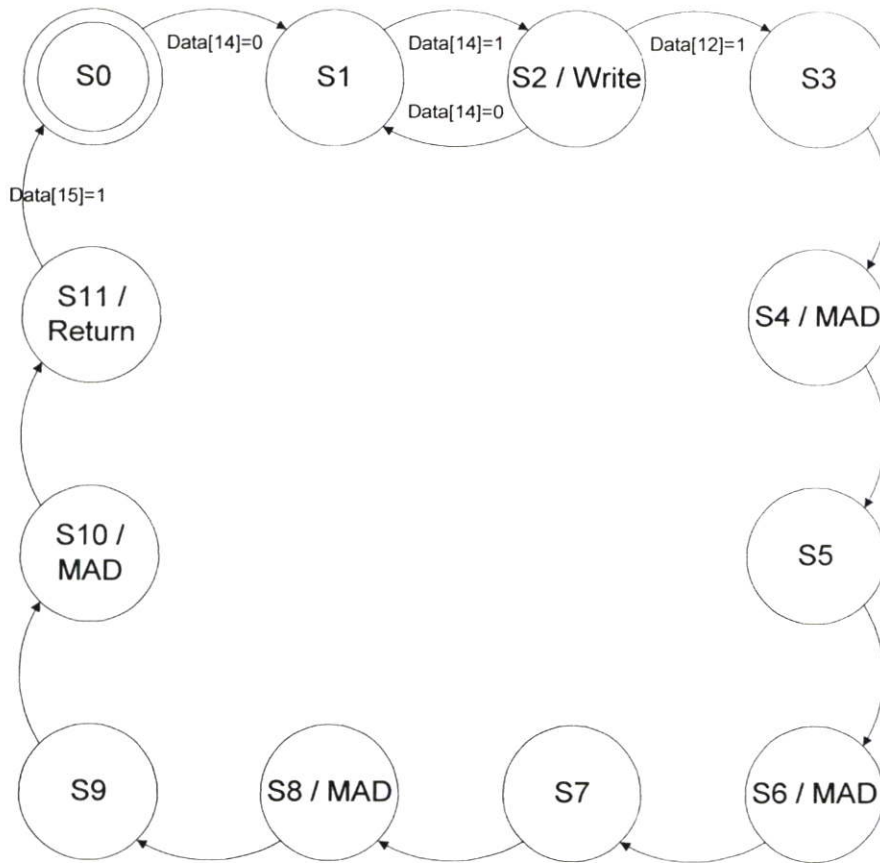
ในส่วนของการส่งข้อมูลผลลัพธ์จากชิพเอฟพีจีเอมายังพีซีนั้น มีรูปแบบที่ง่ายกว่า โดยจะใช้ข้อมูล บิตที่ 9 ถึงบิตที่ 0 ในการส่งข้อมูล และใช้บิตที่ 15 และบิตที่ 14 ในการระบุว่าข้อมูลนี้เป็นค่าแกน X หรือ ค่าแกน Y โดยกำหนดไว้ว่า ถ้าเป็นค่าแกน X ให้บิตที่ 15 เป็น 1 และถ้าเป็นค่าแกน Y ให้บิตที่ 14 เป็น 1

ตารางที่ 5.3 ตัวอย่างข้อมูลจากชิพเอฟพีจีเอไปยังพีซี

ข้อมูล	พิกัด X	พิกัด Y
1000001000010000	1000010000	-
0100001100001000	-	1100001000

5.5 ขั้นตอนในการประมวลผลเปรียบเทียบบล็อกของวงจรมอนอฟพีจีเอ

ในส่วนนี้จะเป็นการอธิบายขั้นตอนการประมวลผลของวงจรประมวลผลเปรียบเทียบบล็อก (Block matching) ตั้งแต่การรับข้อมูลรูปภาพทั้ง 2 รูปจากพีซี การจัดเก็บข้อมูลเข้าสู่หน่วยความจำ การประมวลผลข้อมูล และการจัดส่งผลลัพธ์ที่ได้ไปยังพีซี ในรูปของสเตทแมทซัน โดยกำหนดให้ข้อมูลจากพีซีเรียกว่า Data และจะใช้การระบุตำแหน่งบิตของข้อมูล เช่น ข้อมูลบิตที่ 15 จะแทนด้วย Data[15] เป็นต้น



รูปที่ 5.3 สเตตแมทชีน

S0 เป็นสถานะเริ่มต้น จะเป็นการรอสัญญาณ เมื่อสัญญาณ Data[14] มีค่าเป็น 0 จะเปลี่ยนสถานะไปยังสถานะ S1

S1 เป็นสถานะที่รับข้อมูลแอดเดรสจากพีซี โดยที่สถานะนี้จะทำการอ่านค่า Data[13] เพื่อหาชนิดของข้อมูลภาพที่ส่งมา แล้วทำการแยกหมายเลขแอดเดรสของข้อมูลตามชนิดของข้อมูลภาพ ถ้า Data[13]=0 หมายความว่า เป็นแอดเดรสของพื้นที่ค้นหาแสดงว่าข้อมูลตั้งแต่ Data[9] ถึง Data[0] เป็นแอดเดรส และถ้า Data[13]=1 หมายความว่า เป็นแอดเดรสของบล็อคอ้อยแสดงว่าข้อมูลตั้งแต่ Data[7] ถึง Data[0] เป็นแอดเดรส ให้ทำการแยกข้อมูลดังกล่าวไว้ แล้วรอจนกว่าสัญญาณ Data[14] จะมีค่าเป็น 1 จึงเปลี่ยนไปยังสถานะ S2

S2 เป็นสถานะหลังจากที่ได้รับทั้งแอดเดรสและข้อมูลมาครบทั้ง 2 ค่าแล้ว ที่สถานะนี้จะทำการเขียนข้อมูลไปยังหน่วยความจำของข้อมูลภาพแต่ละประเภท ซึ่งภายในวงจรจะมีหน่วยความจำอยู่ 2 ชุด หน่วยความจำชุดที่ 1 สำหรับเก็บพื้นที่ค้นหา มีขนาด 1024 ไบต์ และหน่วยความจำชุดที่ 4 มีขนาด 256 ไบต์สำหรับเก็บบล็อคอ้อย หลังจากเขียนข้อมูลไปยังหน่วยความจำเสร็จแล้ว จะเป็นการรอสัญญาณ Data[12] มีค่าเป็น 1 เพื่อเปลี่ยนไปยังสถานะ S3 หรือสัญญาณ Data[14] เป็น 0 แล้วเปลี่ยนสถานะไปยัง S1 เพื่อทำการรับข้อมูลชุดต่อไป

S3 จัดเป็นสถานะเริ่มแรกสำหรับส่วนการประมวลผล โดยที่สถานะนี้จะเป็นการกำหนดจุดเริ่มต้นสำหรับการเปรียบเทียบบล็อกย่อยภายในพื้นที่ค้นหา สำหรับอัลกอริทึม PHODS ก็คือพิกัด (0, 0) หลังจากกำหนดจุดเริ่มต้นแล้ว จะทำการเปลี่ยนสถานะไปยัง S4

S4 เป็นสถานะที่ระบบจะเริ่มคำนวณค่า MAD ของวงจรถัง 4 จุด ใน 4 ตำแหน่งพิกัด ซึ่งในการประมวลผลครั้งแรกนั้น ทั้ง 4 วงจรจะเปรียบเทียบที่พิกัดเดียวกัน เมื่อคำนวณเสร็จแล้ว ทำการจัดเก็บค่าผลลัพธ์ MAD ไว้ แล้วจึงเปลี่ยนไปยังสถานะ S5

S5 เป็นสถานะที่คำนวณในตำแหน่ง ± 4 ของตำแหน่งที่ดีที่สุดใแกน x และแกน y ซึ่งในที่นี้คือ (0, 0) ซึ่งจะทำให้ได้พิกัดที่ต้องเปรียบเทียบในทั้ง 4 จุดของวงจรถือ (-4, 0), (4, 0), (0, -4) และ (0, 4)

S6 เป็นสถานะที่คำนวณค่า MAD ของทั้ง 4 จุด แล้วเปรียบเทียบผลในแต่ละแกน และค่าที่ได้จากการคำนวณในรอบก่อนหน้านั้น เพื่อหาค่าที่ดีที่สุดใแกน x และแกน y

S7 เป็นสถานะที่คำนวณในตำแหน่ง ± 2 ของตำแหน่งที่ดีที่สุดใแกน x และแกน y

S8 เป็นสถานะที่ทำการคำนวณค่า MAD และทำการเปรียบเทียบ

S9 เป็นสถานะที่คำนวณในตำแหน่ง ± 1 ของตำแหน่งที่ดีที่สุดใแกน x และแกน y

S10 เป็นสถานะที่ทำการคำนวณค่า MAD และทำการเปรียบเทียบ

S11 เป็นสถานะที่ทำการรวมค่าผลลัพธ์ในแต่ละแกนเข้าด้วยกัน แล้วส่งค่ากลับไปยังพีซี หลังจากส่งค่าผลลัพธ์กลับไปยังพีซีเป็นที่เรียบร้อยแล้วจะทำการหยุดรอสัญญาณรีเซต Data[15]=1 จากทางพีซีเพื่อกลับไปยังสถานะเริ่มต้นใหม่อีกครั้ง

จากสเตทแมทซินดังกล่าวทำให้ทราบถึงจำนวนสัญญาณนาฬิกาที่ต้องใช้ในการทำงานเพื่อเปรียบเทียบบล็อกรูปภาพ 2 ภาพ เพื่อหาการเปลี่ยนแปลงการเคลื่อนที่ โดยแยกย่อยเป็นส่วน ๆ จากขั้นตอนหลัก ๆ ได้ดังนี้

1. ขั้นตอนการรับข้อมูลรูปภาพจากพีซีมายังหน่วยความจำภายในของเอฟพีจีเอ

- สัญญาณรีเซตเพื่อเริ่มต้นการทำงาน 1 สัญญาณนาฬิกา

- ข้อมูลพื้นที่ค้นหาขนาด 1,024 ไบต์ แต่ละไบต์ใช้ 2 สัญญาณนาฬิกา รวมทั้งสิ้น 2,048

สัญญาณ

- ข้อมูลบล็อกย่อยขนาด 256 ไบต์ แต่ละไบต์ใช้ 2 สัญญาณนาฬิกา รวมทั้งสิ้น 512

สัญญาณ

- สัญญาณสิ้นสุดการส่งข้อมูล 1 สัญญาณนาฬิกา

รวมสัญญาณนาฬิกาที่ต้องใช้ในส่วนนี้ทั้งสิ้น 2,562 สัญญาณ

2. ขั้นตอนการการหาค่า MAD

- กำหนดพิกัดที่ต้องทำการเปรียบเทียบ 1 สัญญาณนาฬิกา
- อ่านข้อมูลจากพื้นที่ค้นหา 4 สัญญาณนาฬิกา ทำทั้งหมด 256 ครั้งต่อการคำนวณค่า MAD 1 ครั้ง เท่ากับ 1,024 สัญญาณนาฬิกา
- ค่าวนค่า MAD ใช้สัญญาณนาฬิกา 2 สัญญาณต่อการคำนวณ 1 ตำแหน่ง ซึ่งต้องคำนวณ 256 ครั้ง รวมสัญญาณที่ต้องใช้ 512 สัญญาณ

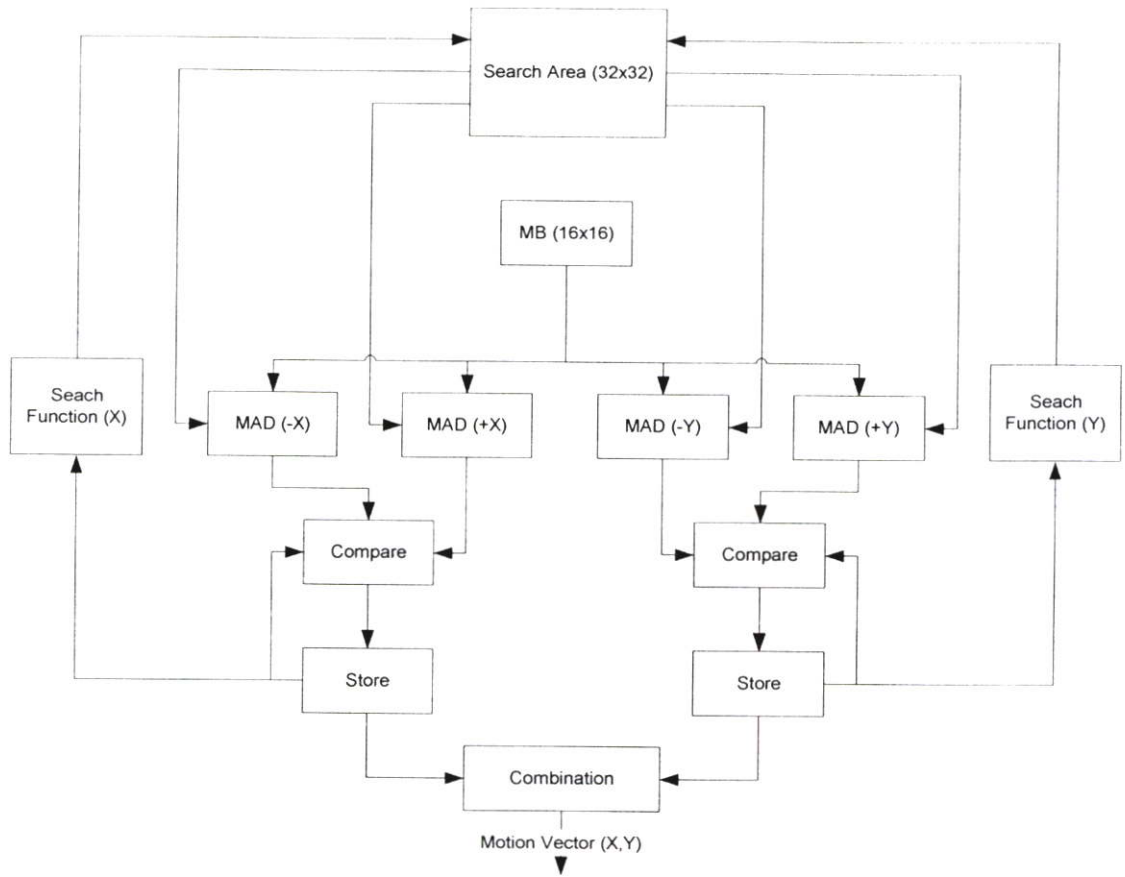
ในการคำนวณด้วยวิธีการ PHODS แบบขนาน 4 ชุด จะมีการคำนวณค่า MAD ทั้งหมด 4 ครั้ง ซึ่งคิดเป็นสัญญาณนาฬิกา 6,144 สัญญาณ

รวมสัญญาณนาฬิกาที่ต้องใช้ในการคำนวณเพื่อหาเวกเตอร์การเคลื่อนที่ทั้งสิ้น 8,706 สัญญาณ

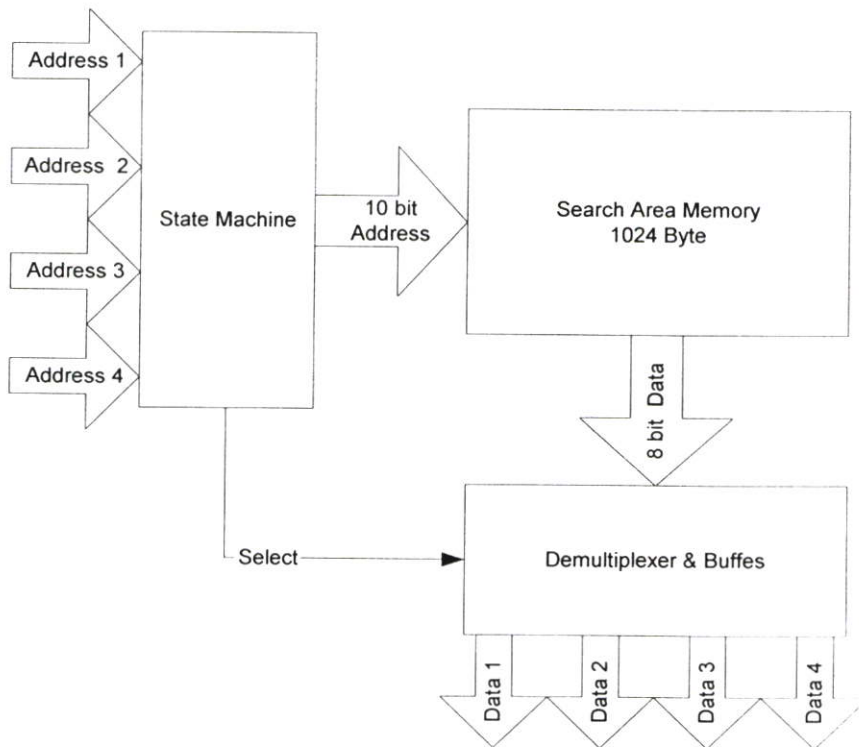
5.6 การพัฒนาระบบประมวลผลแบบขนาน

เพื่อให้การประมวลผลด้วยวงจรดิจิทัลมีประสิทธิภาพสูงยิ่งขึ้น การพัฒนาโครงสร้างการประมวลผลให้รองรับการประมวลผลแบบขนานจึงเป็นสิ่งที่สำคัญอย่างยิ่ง โดยการพัฒนากระบวนการประมวลผลแบบขนานบนชิพเอฟพีจีเอ นั้น มีข้อจำกัดที่สำคัญที่สุดคือความจุของวงจร เนื่องจากการประมวลผลแบบขนานเป็นการเพิ่มขนาดของวงจรขึ้นเป็นอย่างมาก ดังนั้นในการเลือกส่วนใดส่วนหนึ่งของงานมาปรับปรุงให้เป็นการประมวลผลแบบขนานนั้นจึงต้องพิจารณาให้ดี ถึงความเหมาะสม สำหรับการเปรียบเทียบบล็อกด้วยวิธีอัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนานนั้นจะทำในส่วนของการเปรียบเทียบบล็อก เพื่อหาค่า MAD ดังรูปที่ 5.4

ปัญหาที่พบในการเปรียบเทียบบล็อกแบบ 4 ชุดพร้อมกันคือ หน่วยความจำที่ใช้เก็บข้อมูลพื้นที่ค้นหาซึ่งใช้หน่วยความจำภายในของเอฟพีจีเอในการจัดเก็บ เป็นหน่วยความจำที่สามารถอ่านข้อมูลได้ที่ละ 1 ตำแหน่ง และมีทางออกของข้อมูลเพียง 1 ช่องทางเท่านั้น ไม่สามารถที่จะให้วงจรคำนวณหาค่า MAD ทั้ง 4 วงจรเข้าไปอ่านค่าพร้อม ๆ กัน 4 ค่าได้ จึงต้องมีการปรับปรุงหน่วยความจำให้มีลักษณะเป็นการทำงานคล้ายแบบขนาน โดยนำสเตทแมทชีนมาใช้ในการเลือกแอดเดรสที่จะเข้าไปยังหน่วยความจำในลักษณะวนรอบ และนำบัพเฟอร์มาใช้เก็บข้อมูลจากหน่วยความจำภายใน ดังรูปที่ 5.5



รูปที่ 5.4 แผนภาพแสดงการประมวลผลแบบขนาน 4 ชุดของ PHODS



รูปที่ 5.5 แผนภาพแสดงการปรับปรุงหน่วยความจำให้เป็นแบบขนาน 4 ชุด

จากการปรับปรุงหน่วยความจำด้วยวิธีการดังกล่าวทำให้ในขั้นตอนการอ่านข้อมูลจากหน่วยความจำต้องใช้สัญญาณนาฬิกา 4 สัญญาณ ต่อการอ่านข้อมูล 1 ครั้ง แต่วงจรคำนวณค่า MAD ทั้ง 4 ชุดจะสามารถทำงานได้พร้อมกัน โดยใช้สัญญาณนาฬิกาาร่วมกัน หลังจากการคำนวณหาค่า MAD ทั้ง 4 ชุดเสร็จแล้ว จะเป็นขั้นตอนการเปรียบเทียบหาค่าที่ดีที่สุดบนแต่ละแกน ซึ่งในขั้นตอนนี้วงจรที่ขนานกันจะเลือกเพียง 2 ชุด คือชุดของแกน x และชุดของแกน y และเมื่อเสร็จสิ้นการคำนวณทั้ง 4 รอบแล้ว ค่าจากวงจรทั้ง 2 ชุดจะถูกนำมารวมกันอีกครั้งเพื่อเป็นค่าเวกเตอร์การเคลื่อนที่

5.7 ข้อมูลสรุปของวงจร

ในหัวข้อนี้จะเป็นการนำเสนอรายละเอียดต่าง ๆ ของวงจรที่ได้จากการสังเคราะห์โค้ดของวงจรที่ทำการออกแบบด้วยภาษา VHDL บนเครื่องมือพัฒนา Xilinx ISE ซึ่งเป็นเครื่องมือที่รองรับการพัฒนาวงจรบน FPGA ด้วยภาษา VHDL และ Verilog ประกอบไปด้วยเครื่องมือในการเขียนโค้ดคำสั่งและการสังเคราะห์วงจร ซึ่งจากการสังเคราะห์วงจรด้วยเครื่องมือดังกล่าว ทำให้ทราบรายละเอียดการใช้งานเกตของชิพเอฟพีจีเอดังตารางที่ 5.4

ตารางที่ 5.4 จำนวนเกตที่ใช้ไปในการพัฒนาวงจร

	จำนวนที่ใช้ไป	จำนวนที่รองรับได้	อัตราส่วน
Slices	711	1920	37%
Flip Flops	279	3840	7%
4 input LUTs:	1352	3840	35%
bonded IOBs:	27	97	27%
GCLKs:	4	8	50%

5.8 ค่าความถูกต้องของการประมวลผลด้วยวงจรดิจิทัล

ในหัวข้อนี้จะเป็นการเปรียบเทียบค่าความถูกต้องระหว่างการประมวลผลการเคลื่อนที่ด้วยอัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนาน (PHODS) ในแบบที่ใช้ซอฟต์แวร์ประมวลผลบนเครื่องพีซีแต่เพียงอย่างเดียว กับแบบที่ใช้ชิพเอฟพีจีเอมาช่วยประมวลผลในส่วนของการเปรียบเทียบบิตคู่ โดยแสดงทั้งในรูปแบบตาราง และให้รูปของกราฟเพื่อให้สามารถเปรียบเทียบความแตกต่างระหว่างการประมวลผลทั้ง 2 รูปแบบได้อย่างชัดเจน ทั้งค่าความแตกต่างของผลลัพธ์ที่ได้

และแนวโน้มการเปลี่ยนแปลงของค่าความแตกต่างของผลลัพธ์ดังกล่าว โดยแสดงผลไว้ในตารางที่ 5.5 ถึง 5.9 และ รูปที่ 5.6 ถึง 5.10

ตารางที่ 5.5 ค่า PSNR ระหว่างอัลกอริทึม PHODS ที่เป็นซอฟต์แวร์และฮาร์ดแวร์ ของลำดับ

ภาพมาตรฐาน akiyo

ลำดับภาพที่	2	3	4	5	6	7	8	9	10
ซอฟต์แวร์	45.82	45.31	44.61	39.75	37.48	37.33	37.52	39.07	38.41
ฮาร์ดแวร์	45.60	45.08	44.51	40.15	37.51	37.57	37.52	38.44	38.01

ตารางที่ 5.6 ค่า PSNR ระหว่างอัลกอริทึม PHODS ที่เป็นซอฟต์แวร์และฮาร์ดแวร์ ของลำดับ

ภาพมาตรฐาน coastguard

ลำดับภาพที่	2	3	4	5	6	7	8	9	10
ซอฟต์แวร์	29.98	25.96	23.99	23.05	22.18	21.66	21.31	20.99	20.67
ฮาร์ดแวร์	27.72	23.54	21.94	21.08	20.55	20.27	19.95	19.93	19.73

ตารางที่ 5.7 ค่า PSNR ระหว่างอัลกอริทึม PHODS ที่เป็นซอฟต์แวร์และฮาร์ดแวร์ ของลำดับ

ภาพมาตรฐาน container

ลำดับภาพที่	2	3	4	5	6	7	8	9	10
ซอฟต์แวร์	36.43	31.99	32.87	33.54	32.49	30.38	30.05	30.23	29.62
ฮาร์ดแวร์	36.04	31.54	32.33	32.86	29.97	28.58	28.07	27.23	26.90

ตารางที่ 5.8 ค่า PSNR ระหว่างอัลกอริทึม PHODS ที่เป็นซอฟต์แวร์และฮาร์ดแวร์ ของลำดับ

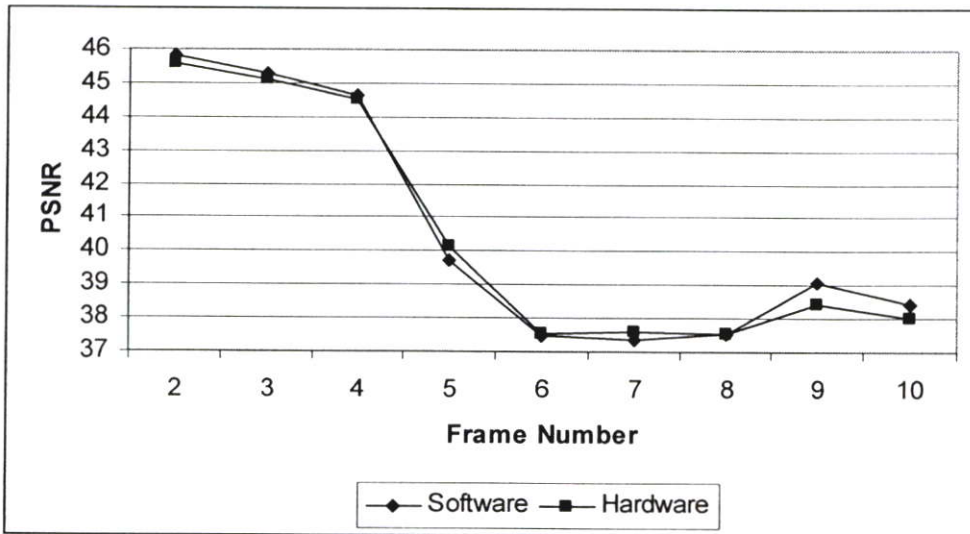
ภาพมาตรฐาน mobile

ลำดับภาพที่	2	3	4	5	6	7	8	9	10
ซอฟต์แวร์	20.93	18.28	16.01	15.31	14.88	14.29	14.07	13.74	13.53
ฮาร์ดแวร์	19.78	16.71	14.94	14.67	14.28	13.71	13.42	13.03	12.64

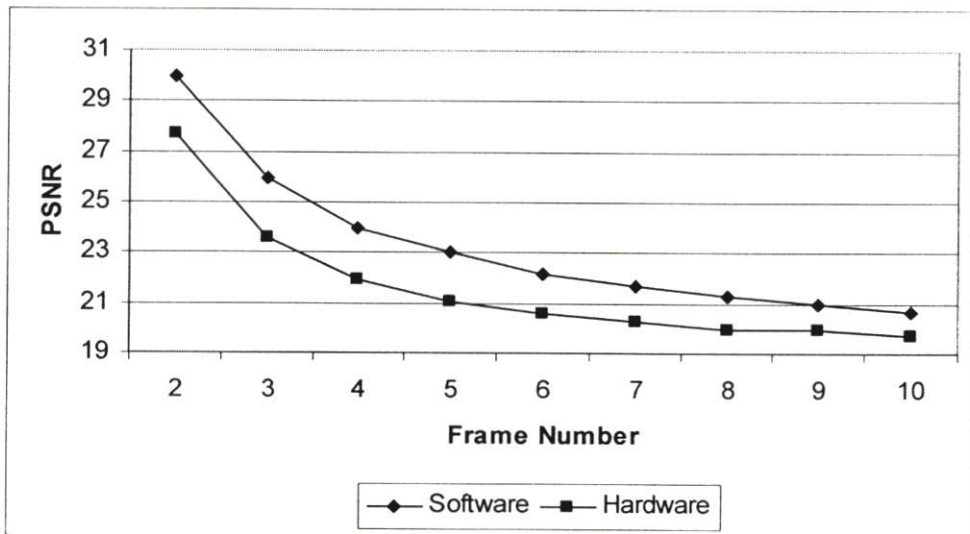
ตารางที่ 5.9 ค่า PSNR ระหว่างอัลกอริทึม PHODS ที่เป็นซอฟต์แวร์และฮาร์ดแวร์ ของลำดับ

ภาพมาตรฐาน tempete

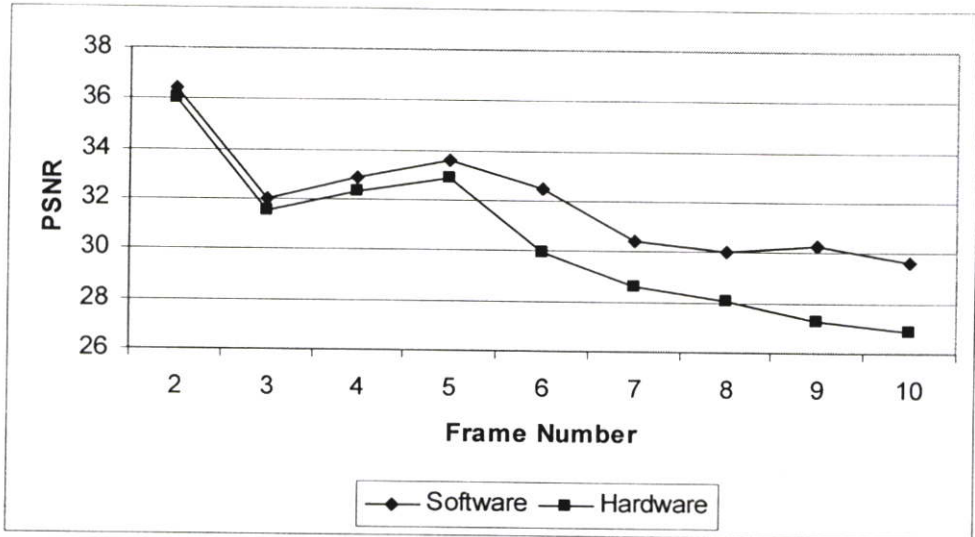
ลำดับภาพที่	2	3	4	5	6	7	8	9	10
ซอฟต์แวร์	25.94	23.70	22.76	21.96	20.67	19.90	19.51	19.02	18.56
ฮาร์ดแวร์	25.70	23.13	22.09	20.84	19.66	19.15	18.87	18.75	18.02



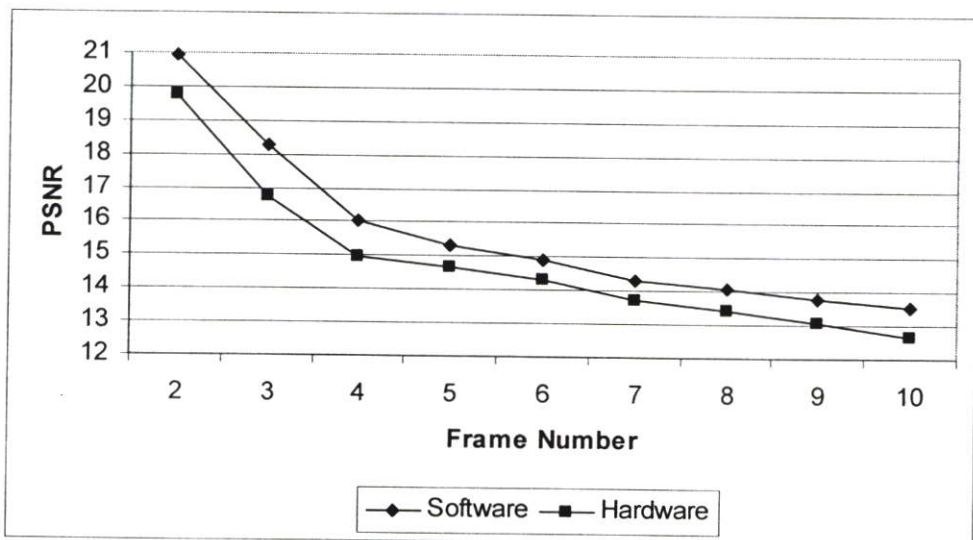
รูปที่ 5.6 กราฟเปรียบเทียบค่า PSNR ของลำดับภาพ akiyo ที่คำนวณโดยซอฟต์แวร์และฮาร์ดแวร์



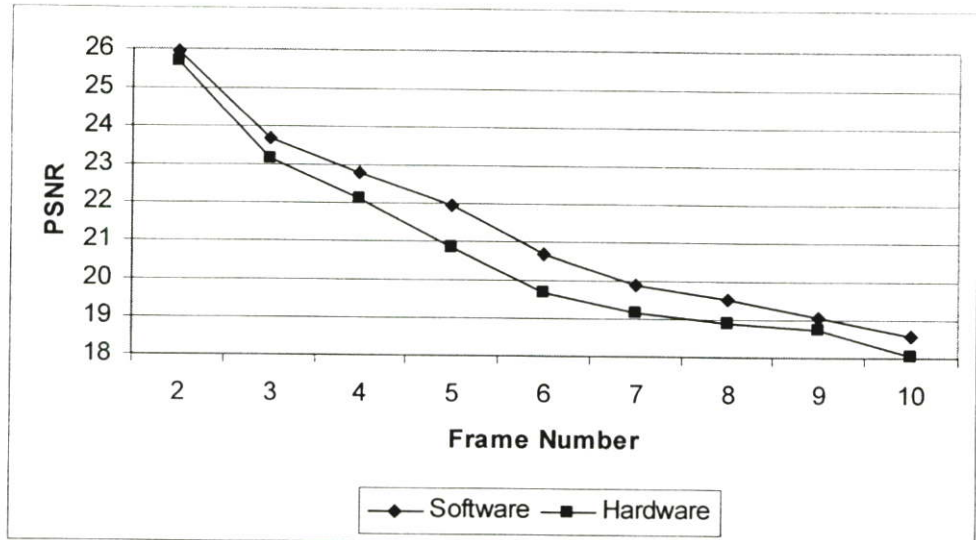
รูปที่ 5.7 กราฟเปรียบเทียบค่า PSNR ของลำดับภาพ coastguard ที่คำนวณโดยซอฟต์แวร์และฮาร์ดแวร์



รูปที่ 5.8 กราฟเปรียบเทียบค่า PSNR ของลำดับภาพ container ที่คำนวณโดยซอฟต์แวร์และฮาร์ดแวร์



รูปที่ 5.9 กราฟเปรียบเทียบค่า PSNR ของลำดับภาพ mobile ที่คำนวณโดยซอฟต์แวร์และฮาร์ดแวร์



รูปที่ 5.10 กราฟเปรียบเทียบค่า PSNR ของลำดับภาพ tempete ที่คำนวณโดยซอฟต์แวร์และฮาร์ดแวร์

จากข้อมูลค่าความถูกต้องดังกล่าวจะเห็นได้ว่าค่าความถูกต้องของการประมวลผลด้วยฮาร์ดแวร์มีน้อยกว่าการใช้ซอฟต์แวร์แต่เพียงอย่างเดียว เนื่องมาจากโครงสร้างการคำนวณของวงจรที่ออกแบบไว้ให้เป็นการคำนวณเลขจำนวนเต็ม จึงทำให้มีความคลาดเคลื่อนกว่าการคำนวณด้วยซอฟต์แวร์ซึ่งเป็นการคำนวณด้วยระบบจำนวนจริง

บทที่ 6

บทสรุปผลการวิจัยและข้อเสนอแนะ

เทคโนโลยีการบีบอัดข้อมูลวิดีโอ (Video compression) ได้มีการพัฒนาไปอย่างมาก สามารถลดขนาดของข้อมูลวิดีโอได้ในอัตราส่วนที่สูง แต่ในขณะเดียวกันก็มีความต้องการทรัพยากรของระบบประมวลผลที่สูงมากขึ้น เนื่องจากการคำนวณที่ซับซ้อน โดยเฉพาะในส่วนของ การบีบอัดข้อมูลวิดีโอ (Video encoding) เพราะเหตุผลดังกล่าวจึงทำให้ระบบประมวลผลที่มีความเร็วต่ำ เช่น อุปกรณ์พกพาต่าง ๆ มีปัญหาในการเข้ารหัสวิดีโอที่มีความละเอียดสูงไม่ได้ในความเร็วที่อยู่ในระดับเรียลไทม์ (Real time)

การประมาณการเคลื่อนที่ (Motion estimation) เป็นองค์ประกอบหลักในขั้นตอนการเข้ารหัสวิดีโอในหลาย ๆ มาตรฐานในปัจจุบัน ซึ่งในส่วนของ การประมาณการเคลื่อนที่นั้นจะใช้เวลาประมาณ 60% ของขั้นตอนการเข้ารหัสวิดีโอทั้งหมด ถ้าหากสามารถพัฒนาความเร็วในส่วนนี้ได้ ก็จะส่งผลต่อความเร็วในการเข้ารหัสทั้งหมดเช่นกัน การประมาณการเคลื่อนที่ที่เป็นกระบวนการหาความเปลี่ยนแปลงระหว่างลำดับภาพ (Frame) 2 ภาพ จากวิดีโอ เพื่อใช้ลดขนาดข้อมูลวิดีโอ เนื่องจากพฤติกรรมของของภาพวิดีโอในลำดับที่ใกล้เคียงกันจะมีส่วนที่แตกต่างกันน้อยมาก การจับเก็บข้อมูลเฉพาะบางลำดับภาพเป็นภาพอ้างอิง โดยลำดับภาพที่เหลือให้จับเก็บเฉพาะข้อมูลการเปลี่ยนแปลงเมื่อเทียบกับภาพอ้างอิงที่มีการจับเก็บ ซึ่งทำให้ลดปริมาณข้อมูลลงไปได้อย่างมาก วิธีการที่ใช้สำหรับหาความเปลี่ยนแปลงระหว่างภาพ คือ อัลกอริทึมการเปรียบเทียบแบบกลุ่ม (Block matching algorithm) ซึ่งประกอบไปด้วย 2 ส่วน คืออัลกอริทึมการค้นหา (Search algorithm) และคอสต์ฟังก์ชัน (Cost function) การเพิ่มความเร็วให้กับอัลกอริทึมการเปรียบเทียบแบบกลุ่ม จะใช้วิธีปรับปรุงในส่วนของอัลกอริทึมการค้นหา เพื่อลดจำนวนพิกัดที่ต้องเปรียบเทียบให้น้อยลง ในขณะเดียวกันก็ต้องรักษาคุณภาพของภาพเอาไว้ด้วย

นอกจากการปรับปรุงในเชิงอัลกอริทึมแล้ว การพัฒนาหน่วยประมวลผลเฉพาะทางสำหรับคำนวณการประมาณการเคลื่อนที่ ก็เป็นอีกวิธีหนึ่งที่ถูกนำมาใช้เพื่อเพิ่มความเร็วในการคำนวณ เนื่องจากการประมวลผลด้วยวงจรดิจิทัลที่ออกแบบมาเฉพาะจึงทำให้มีความเร็วที่สูงกว่าการประมวลผลด้วยหน่วยประมวลผลแบบทั่วไป จึงเหมาะสำหรับระบบที่มีหน่วยประมวลผลที่ความเร็วไม่สูง ที่การปรับปรุงในเชิงอัลกอริทึมอย่างเดียวไม่เพียงพอ แต่การพัฒนาหน่วยประมวลผลเฉพาะทาง ก็ควรที่จะเลือกอัลกอริทึมให้เหมาะสมกับการนำมาสร้างเป็นวงจรดิจิทัล ซึ่งสามารถเพิ่มความเร็วในประมวลผล ง่ายต่อการสร้างเป็นวงจร รวมทั้งยังสามารถลดขนาดของวงจรลงได้อีกด้วย

วิทยานิพนธ์ฉบับนี้ได้นำเสนออัลกอริทึมการค้นหา 1 มิติเป็นลำดับขั้นแบบขนาน 2 ระดับ ซึ่งเป็นอัลกอริทึมที่พัฒนาขึ้นมาเพื่อใช้สำหรับการนำไปสร้างเป็นหน่วยประมวลผลเฉพาะทาง สำหรับงานด้านการประมาณการเคลื่อนที่ เพื่อนำไปใช้ในงานบีบอัดภาพวิดีโอ โดยจุดเด่นของอัลกอริทึมนี้อยู่ตรงรูปแบบของอัลกอริทึมที่เป็นแบบขนาน สามารถแยกคำนวณค่าออกเป็น 2 ชุดอิสระจากกัน จึงทำให้มีความเร็วสูง ในขณะที่เดียวกันอัลกอริทึมก็มีพิกัดที่ต้องเปรียบเทียบน้อย จึงทำให้มีความเร็วสูงแม้จะไม่ได้คำนวณในรูปแบบที่ขนานกันก็ตาม นอกจากนี้ยังมีการคำนวณที่ไม่ซับซ้อนในอัลกอริทึมสำหรับค้นหาโดยประกอบไปด้วยบวก ลบ และหาร ในส่วนของการหารจะเป็นการหารด้วย 2 ซึ่งในวงจรดิจิทัลเป็นเพียงแค่การเลื่อนบิตไปทางขวาเท่านั้น ในส่วนของความถูกต้องของผลลัพธ์ อัลกอริทึมนี้ได้มีการพัฒนาในด้านของความถูกต้องของผลลัพธ์ โดยลดความคลาดเคลื่อนให้น้อยลงกว่าอัลกอริทึมดั้งเดิมที่เป็นต้นแบบ

การพัฒนาอัลกอริทึมให้เหมาะสมสำหรับการนำไปสร้างเป็นหน่วยประมวลผลเฉพาะสำหรับการประมาณการเคลื่อนที่ที่นำเสนอในวิทยานิพนธ์นี้เป็นเพียงวิธีการหนึ่งที่น่าสนใจเพื่อเพิ่มความเร็วในการเข้ารหัสวิดีโอ งานวิจัยทางด้านการเข้ารหัสวิดีโอ หรือด้านการประมาณการเคลื่อนที่ ก็ยังมีความน่าสนใจในอีกหลาย ๆ ด้าน การพัฒนาหน่วยประมวลผลเฉพาะทางก็เช่นกัน ถ้าหากมีการเพิ่มคำสั่งพื้นฐานของการบีบอัดวิดีโอ เช่น DCI/IDCT และ Quantization ก็อาจจะพัฒนาไปเป็นหน่วยประมวลผลสำหรับเข้ารหัสบีบอัดวิดีโอ (Video compression processing unit) ก็เป็นไปได้

บรรณานุกรม

- [1] Borko Furht, Joshua Greenberg, Raymond Westwater. **Motion Estimation Algorithms for Video Compression** .Massachusetts : Kluwer Academic Publishers, 1997.
- [2] Liang-Gee Chen, Wai-Ting Chen, Yeu-Shen Jehng, Tzi-Dar Chiuch “An Efficient Parallel Motion Estimation Algorithm for Digital Image Processing” **IEEE Transactions on Circuits and Systems for Video Technology**, Vol.1, Issue 4, Dec. 1991. pp.378 - 385
- [3] Nan Yu, Kichul Kim, Salcic Z., “A New Motion Estimation Algorithm for Mobile Real-Time Video and its FPGA Implementation” **TENCON**, Vol. 1, Nov. 2004. pp 383 - 386.
- [4] Rangarajan P., Prashanth G., Harish P.S. “FPGA Implementation of High Speed Parallel Architecture for Block Motion Estimation” **SIPS**, 2004. pp. 245 - 250.

ภาคผนวก ก.

ผลงานวิจัยที่ได้รับการตีพิมพ์



NCCIT'07



NCCIT'07

The 3rd National Conference
on Computing and Information Technology

ISBN 978-974-19-3281-8

การประชุมทางวิชาการระดับชาติ ด้านคอมพิวเตอร์ ครั้งที่ 3 และเทคโนโลยีสารสนเทศ

25 - 26 พ.ค. 2550



คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ



เกียรติบัตรนี้ไว้เพื่อแสดงว่า

มานะ กรัษประพັນ

ได้นำเสนอผลงานวิจัยในการประชุมทางด้านวิชาการระดับชาติ
ด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ครั้งที่ 3

วันที่ 25-26 พฤษภาคม พ.ศ. 2550

อานันท์ ปันยารชุน

รองศาสตราจารย์ ดร. มนต์ชัย เกตุทอง
ประธานงานวิชาการ



**Finalist for
Best Paper Award**

The Third National Conference on Computing and Information Technology
at the King Mongkut's Institute of Technology North Bangkok

To

Mana Krasrapun

for the paper entitled

A New Motion Estimation Algorithm for Hardware Implementation

25 June 2007

Manchai Hantong

Associate Professor Dr. Manchai Hantong
Conference Director

อัลกอริทึมการประมาณการเคลื่อนไหวสำหรับการประมวลผลด้วยฮาร์ดแวร์

มานะ กรีษประพูน

ผศ.อภิเนตร อุณาอุล

ภาควิชาวิศวกรรมคอมพิวเตอร์

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

m_krasrapun@hotmail.com

apinet@yahoo.com

บทคัดย่อ

เทคโนโลยีการพัฒนาหน่วยประมวลผลเฉพาะทางได้ถูกนำมาใช้เพื่อเพิ่มความเร็วให้กับงานประมวลผลที่ต้องมีการประมวลผลข้อมูลจำนวนมาก เช่น การเข้ารหัสวิดีโอ บทความนี้ได้นำเสนอการพัฒนาหน่วยประมวลผลเฉพาะทางเพื่อเพิ่มความเร็วให้กับการเข้ารหัสวิดีโอ โดยมีพัฒนาเฉพาะในส่วน of ขั้นตอนการประมาณการเคลื่อนไหว ซึ่งเป็นขั้นตอนที่มีการประมวลผลมากที่สุดในการเข้ารหัสวิดีโอที่ใช้ในปัจจุบัน ในการเพิ่มความเร็วให้กับการประมวลผลด้วยวิธีนี้ การเลือกอัลกอริทึมการประมาณการเคลื่อนไหวให้เหมาะสมกับการนำมาสร้างเป็นอุปกรณ์ฮาร์ดแวร์ ก็เป็นการเพิ่มประสิทธิภาพของหน่วยประมวลผลได้อีกระดับหนึ่ง โดยพัฒนาอัลกอริทึมที่เหมาะสมสำหรับการสร้างเป็นฮาร์ดแวร์ที่มีชื่อว่า Two-Level Parallel Hierarchical One-Dimensional Search (2LPHODS) โดยปรับปรุงมาจากอัลกอริทึม Parallel Hierarchical One-Dimensional Search (PHODS) ซึ่งเป็นอัลกอริทึมที่มีความเร็วสูง ไม่ซับซ้อน และยังมีรูปแบบอัลกอริทึมแบบขนาน ทำให้มีประสิทธิภาพสูงเมื่อนำไปสร้างเป็นฮาร์ดแวร์ แต่ในด้านความคลาดเคลื่อนของผลลัพธ์ยังมีมาก จึงได้พัฒนาอัลกอริทึมดังกล่าวขึ้นมาโดยลดความเร็วในการประมวลผลลงเล็กน้อย แต่เพิ่มความถูกต้องของผลลัพธ์ให้มากขึ้น เพื่อให้เหมาะสมกับการนำไปใช้งานจริง และยังคงคุณสมบัติเด่นของอัลกอริทึมดั้งเดิมเอาไว้

คำสำคัญ: การประมาณการเคลื่อนไหว, การเปรียบเทียบบล็อก, เอฟพีจีเอ

Abstract

Technology for specific processing unit has been used for improving the processing time that need high speed processing such as video encoding which requires a lot of data to be processed in limit of time. This research works and presents a development of specific processing unit to reduce video encoding time especially step in calculating motion estimation, which requires high calculating in processing standard video encoding. When ever the motion estimation step has been speed up, the overall time consuming will be reduced significantly. Through utilizing only specific processing unit for improving the processing time may not be enough. Therefore, selecting appropriate algorithm for calculating the motion estimation in developing hardware processing unit will also improve processing unit. This research presents new algorithm called Two-Level Parallel Hierarchical One-Dimensional Search (2LPHODS) based on algorithm called Parallel Hierarchical One-Dimensional Search (PHODS) which is high speed and simple algorithm and also in parallel form which increases the processing performance when developing in hardware unit. However the result from PHODS stills incorrect. Therefore the new algorithm has been proposed for increasing the correctness of result but reducing only a little bit processing speed for utilizing in real application and also still remain the main features in the old algorithm.

Keyword: motion estimation, Block matching algorithm, FPGA

1. บทนำ

ปัจจุบันเทคโนโลยีในงานทางด้านมัลติมีเดียบนอุปกรณ์พกพาขนาดเล็ก เช่น โทรศัพท์มือถือ และ PDA ได้มีการพัฒนาไปอย่างมาก แต่อย่างไรก็ตามงานเกี่ยวกับวิดีโอบนอุปกรณ์เหล่านี้ก็ยังมีปัญหาอีกมากในงานที่ต้องการคุณภาพสูง และต้องการความเร็วระดับเรียลไทม์ มาตรฐานการเข้ารหัสวิดีโอมีอยู่ด้วยกันหลายมาตรฐาน ซึ่งเมื่อนำมาใช้บนอุปกรณ์พกพาเหล่านี้จะมีปัญหาข้อจำกัดหลัก 2 อย่างคือข้อจำกัดทางด้านพลังงานที่มีอยู่จำกัด และข้อจำกัดทางทรัพยากรของระบบ เช่น ขนาดของหน่วยความจำ และความเร็วของหน่วยประมวลผลที่ไม่สูงมากนัก ถ้าวิเคราะห์ขั้นตอนในการเข้ารหัสวิดีโอแล้วจะพบว่าในขั้นตอนการเข้ารหัสวิดีโอกว่า 60% จะถูกใช้ไปในการคำนวณการประมาณการเคลื่อนไหวดังนั้นการปรับปรุงและพัฒนาประสิทธิภาพในส่วนของขั้นตอนการประมาณการเคลื่อนไหวจึงส่งผลต่อประสิทธิภาพของการเข้ารหัสวิดีโอโดยรวมเป็นอย่างมาก วิธีที่นำมาใช้ประมาณการเคลื่อนไหวนั้นใช้กันมากในมาตรฐานต่าง ๆ ที่มีอยู่ในปัจจุบัน คือ Block Matching Algorithm (BMA)

การพัฒนาอัลกอริทึมการค้นหาจากเดิมที่ใช้วิธีการที่เรียกว่า Full Search ซึ่งเป็นการเปรียบเทียบทุก ๆ ตำแหน่งในพื้นที่ที่กำหนดเพื่อหาค่าที่ดีที่สุด มาเป็นการเลือกเปรียบเทียบเป็นบางตำแหน่งภายใต้เงื่อนไขที่กำหนด ทำการค้นหาแบบเป็นลำดับขั้น ไปจนกระทั่งได้ผลลัพธ์ที่ใกล้เคียงกับค่าที่ดีที่สุดที่หาด้วยวิธีการ Full Search เช่น อัลกอริทึม Three-Step Search (3SS) และ Two-Dimensional Logarithm Search (2DLS) วิธีการเหล่านี้ทำให้ลดการเวลาประมวลผลในขั้นตอนนี้อย่างมาก รวมทั้งลดการใช้พลังงานของระบบ อันเนื่องมาจากขั้นตอนการทำงานที่ลดลง นอกจากการพัฒนาในด้านอัลกอริทึมแล้ว การพัฒนาหน่วยประมวลผลเฉพาะทางก็เป็นอีกวิธีหนึ่งที่ถูกนำมาใช้ วิธีการนี้จะให้ผลลัพธ์ในด้านความเร็วที่ค่อนข้างสูง เพราะเป็นการประมวลผลด้วยฮาร์ดแวร์โดยตรง รวมทั้งยังเป็นการลดภาระของหน่วยประมวลผลของระบบลงด้วย แต่วิธีการนี้เป็นการเพิ่มภาระทางด้านพลังงานให้กับระบบ

เพราะฮาร์ดแวร์ที่เพิ่มขึ้น ดังนั้นการใช้วิธีนี้จึงต้องคำนึงถึงเรื่องขนาดของวงจรและความซับซ้อนของขั้นตอนในการคำนวณเพื่อให้เกิดภาระในการใช้พลังงานน้อยที่สุด

การพัฒนาการคำนวณหาค่าการประมาณการเคลื่อนไหวดังกล่าวด้วยวงจรถอดลอกลงมีด้วยกันหลายแนวทางไม่ว่าจะเป็นการพัฒนาทางด้านสถาปัตยกรรมของการประมวลผล และการพัฒนาอัลกอริทึมที่นำมาใช้สร้างเป็นวงจร ในส่วนของ การพัฒนาสถาปัตยกรรมจะเป็นการนำเสนอโครงสร้างข้อมูล ขนาดของข้อมูล โครงสร้างเส้นทางของข้อมูล การจัดเก็บข้อมูล รูปแบบคำสั่ง โครงสร้างหน่วยประมวลผลทางคณิต และลำดับขั้นตอนการประมวลผล ในส่วนของการพัฒนาอัลกอริทึมนั้น เนื่องจากอัลกอริทึมที่ใช้ในการประมาณการเคลื่อนไหวนั้นมีเป็นจำนวนมาก การเลือกอัลกอริทึมเพื่อนำมาสร้างเป็นวงจรให้เหมาะสมกับการประมวลผลที่เป็นวงจรถอดลอกลงจะเป็นการเพิ่มประสิทธิภาพให้กับหน่วยประมวลผลที่สร้างขึ้นด้วยเช่นกัน โดยเฉพาะเมื่อมีเทคโนโลยีของชิป FPGA ซึ่งเป็นไอซีที่สามารถนำมาโปรแกรมโครงสร้างภายในให้สามารถประมวลผลข้อมูลได้ตามที่ต้องการ ทำให้การพัฒนาไอซีเฉพาะด้านนั้น มีความง่าย สะดวก และรวดเร็ว รวมทั้งสามารถรองรับวงจรที่มีขนาดใหญ่ได้ จึงทำให้เหมาะสมที่จะนำไปพัฒนาเป็นหน่วยประมวลผลเฉพาะทางเพื่อเพิ่มความเร็ว ให้กับ การประมวลผลที่ค่อนข้างซับซ้อน

บทความนี้แนะนำให้เสนอการพัฒนาอัลกอริทึมการประมาณการเคลื่อนไหวนั้นที่เหมาะสมกับการนำไปสร้างเป็นหน่วยประมวลผลเฉพาะทาง เพื่อนำไปใช้ในงานเข้ารหัสภาพวิดีโอ เพื่อให้ได้ความเร็วที่สูงขึ้น และใช้ทรัพยากรในการเข้ารหัสน้อยลง โดยเลือกใช้วิธีการปรับปรุงอัลกอริทึมที่มีอยู่แล้วที่มีความเหมาะสมในหลาย ๆ ด้าน แต่อย่างไรก็ตามมีจุดบกพร่องอยู่มาก มาแก้ไขปรับปรุงให้เหมาะสมกับการใช้งานมากขึ้น

2. การประมาณการเคลื่อนไหว (Motion Estimation)

การประมาณการเคลื่อนไหวเป็นกระบวนการหาความเปลี่ยนแปลงตำแหน่งของส่วนต่าง ๆ ในภาพระหว่างภาพ 2 ภาพที่มีความต่อเนื่องกัน เช่น ภาพเคลื่อนไหวดังกล่าว ซึ่งวิธีการประมาณการเคลื่อนไหวนั้นได้ถูกนำมาใช้ในงานเข้ารหัสภาพ

วิดีโอเพื่อลดขนาดของข้อมูลภาพวิดีโอซึ่งเป็นข้อมูลขนาดใหญ่ให้มีขนาดที่เล็กลง โดยอาศัยหลักการพื้นฐานที่ภาพวิดีโอที่มีลำดับใกล้เคียงกัน จะมีความแตกต่างกันน้อยมาก การจัดเก็บภาพวิดีโอทุกภาพจึงถือเป็นการจัดเก็บที่มากเกินไปจนเกินไป จึงได้เกิดแนวคิดที่จะจัดเก็บเต็มภาพเฉพาะบางภาพ เป็นช่วง ๆ เวลา และจัดเก็บภาพที่เหลือเฉพาะส่วนที่มีการเปลี่ยนแปลงเท่านั้น วิธีการนี้จะเป็นการลดขนาดของข้อมูลวิดีโอลงได้อย่างมาก ซึ่งวิธีการที่นำมาใช้เพื่อหาส่วนที่เปลี่ยนแปลงระหว่างภาพดังกล่าวคือ การประมาณการเคลื่อนไหว

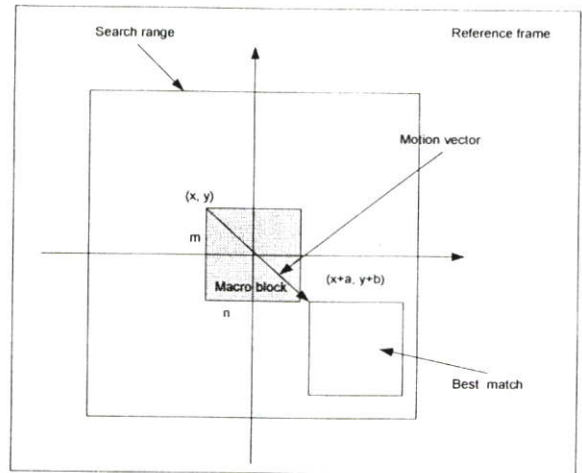
วิธีการประมาณการเคลื่อนไหวที่ใช้มากในมาตรฐานการเข้ารหัสวิดีโอในปัจจุบันคือ อัลกอริทึมเปรียบเทียบบล็อก (Block matching algorithm) ซึ่งจะประกอบไปด้วย 2 ส่วนหลัก ๆ คือการค้นหา และการเปรียบเทียบ การค้นหาเป็นการคำนวณเพื่อหาตำแหน่งพิกัดต่าง ๆ ในภาพอ้างอิงที่ต้องการเปรียบเทียบในแต่ละขั้นตอน ตามแต่อัลกอริทึมที่ใช้จะกำหนดอัลกอริทึมที่ใช้หาพิกัดต่าง ๆ ที่ต้องการเปรียบเทียบเรียกว่า อัลกอริทึมค้นหา (Search algorithm) ขั้นตอนการค้นหาอย่างคร่าว ๆ จะเริ่มจากการแบ่งภาพปัจจุบันที่ต้องการจะหาการเคลื่อนไหวเป็นบล็อกขนาด $m \times n$ ที่เรียกว่ามาโครบล็อก จากนั้นจึงนำบล็อกดังกล่าวไปเปรียบเทียบ เพื่อหาตำแหน่งที่ใกล้เคียงที่สุดในภาพอ้างอิง การเปรียบเทียบจะเริ่มเปรียบเทียบจากตำแหน่งเดิมของบล็อกนั้น จากนั้นจึงเปรียบเทียบกับพิกัดอื่นรอบ ๆ ตำแหน่งดั้งเดิมภายในรัศมีที่กำหนดซึ่งเรียกว่า รัศมีค้นหา เส้นที่ลากจากตำแหน่งเดิมของมาโครบล็อกไปยังตำแหน่งที่ใกล้เคียงที่สุดนั้นเรียกว่าเวกเตอร์การเคลื่อนไหวของมาโครบล็อกดังกล่าวที่ 1

การเปรียบเทียบจะใช้ฟังก์ชันที่เรียกว่าคอสต์ฟังก์ชัน ในการหาค่าความแตกต่างระหว่างบล็อก ซึ่งมีอยู่ด้วยกันมากมายหลายฟังก์ชัน แต่ที่นิยมใช้ และใช้ในบทความนี้คือ ค่า MAD (Mean Absolute Difference) ซึ่งมีรายละเอียดสมการดังนี้

$$MAD(dx, dy) = \frac{1}{mn} \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} \sum_{j=-\frac{m}{2}}^{\frac{m}{2}} |F(i, j) - G(i + dx, j + dy)|$$

เมื่อ $F(i, j)$ คือ มาโครบล็อกของภาพปัจจุบัน $G(i, j)$ คือ บล็อกที่กำลังเปรียบเทียบในภาพอ้างอิง และ (dx, dy) เป็นเวกเตอร์ไปยังตำแหน่งที่กำลังเปรียบเทียบ ค่า MAD ยิ่งน้อยยิ่งแสดงถึงความเหมือนระหว่างบล็อกทั้งสอง ค่า (dx, dy) ของตำแหน่งที่

มีค่า MAD น้อยที่สุดคือ เวกเตอร์การเคลื่อนไหวของบล็อก



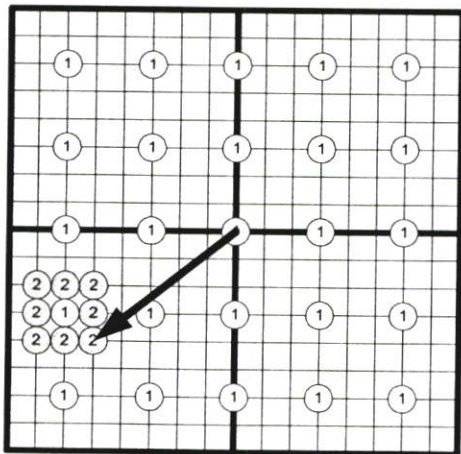
ภาพที่ 1: การประมาณการเคลื่อนไหว

ในการพัฒนาเพื่อเพิ่มความเร็วของการเปรียบเทียบนั้น ส่วนใหญ่จะพัฒนาในส่วนของการค้นหาเป็นหลัก โดยพัฒนารูปแบบของอัลกอริทึมให้เหมาะสมไปตามการใช้งาน ซึ่งมีทั้งที่เน้นด้านความเร็ว ความถูกต้อง และความง่ายในการคำนวณ นอกจากนี้ยังมีการพัฒนาในด้านความเหมาะสมกับรูปแบบการใช้งานอีกด้วยเช่น การประมวลผลด้วยอุปกรณ์ฮาร์ดแวร์

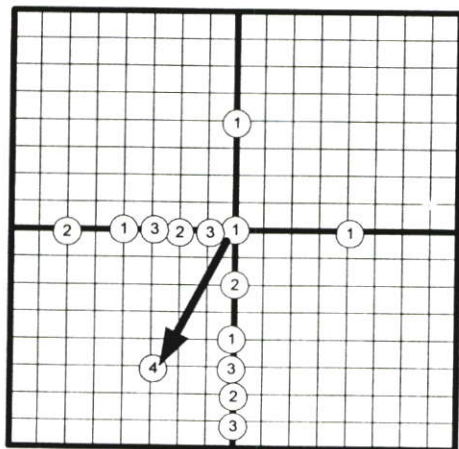
Two-Step Search (2SS) เป็นอัลกอริทึมที่ถูกพัฒนาขึ้นมาเพื่อใช้ในการประมวลผลด้วยฮาร์ดแวร์ อัลกอริทึมพัฒนาขึ้นโดยเน้นที่สมดุลระหว่างประสิทธิภาพ และขนาดของฮาร์ดแวร์ จุดเด่นของอัลกอริทึมนี้อยู่ที่การคำนวณที่ไม่ซับซ้อน ในขณะที่ผลลัพธ์ที่ได้ใกล้เคียงกับการหาด้วยวิธี Full search แต่จำนวนพิกัดที่เปรียบเทียบยังมีจำนวนมากกว่าภาพที่ 2 ซึ่งจะอยู่ที่ประมาณ 12.5 - 25% ของ Full search เนื่องจากตำแหน่งที่ต้องเทียบนั้นไม่มีสมการที่ตายตัว ขึ้นอยู่กับขนาดของพื้นที่ค้นหา และการออกแบบ นอกจากนี้การพัฒนาอัลกอริทึมเพื่อมาสร้างเป็นฮาร์ดแวร์นั้น ไม่เพียงแต่อัลกอริทึมต้องมีการคำนวณที่ไม่ซับซ้อนเท่านั้น ถ้าหากอัลกอริทึมมีลักษณะเป็นแบบขนาน ก็จะเป็นเพิ่มประสิทธิภาพให้กับการประมวลผลด้วยฮาร์ดแวร์เป็นอย่างมาก ซึ่งอัลกอริทึมนี้ไม่ได้รองรับในส่วนนั้น

อัลกอริทึม Parallel Hierarchical One-Dimensional Search (PHODS) นำเสนอวิธีการหาเวกเตอร์การเคลื่อนไหว ซึ่งเป็นค่าที่ใช้บ่งบอกขนาดและทิศทางของการเคลื่อนไหวที่เป็น 2

มิติ โดยแทนที่จะหาโดยตรง แต่ใช้การหาแบบมิติเดียว 2 ครั้ง ขนานไปพร้อมกันใน 2 แกน แยกอิสระออกจากกันภายใน พื้นที่ค้นหาถึงภาพที่ 3 จุดเด่นของอัลกอริทึมนี้คืออยู่ที่ความเร็วที่สูงกว่าอัลกอริทึมชนิดอื่น ๆ เนื่องจากตำแหน่งที่ต้องทำการเปรียบเทียบมีน้อย มีการคำนวณที่ไม่ซับซ้อน และมีการทำงานแบบขนานกัน จึงเหมาะสมอย่างมากที่จะนำไปใช้ในฮาร์ดแวร์ แต่ปัญหาของอัลกอริทึมนี้คือที่ด้านความถูกต้องของผลลัพธ์นั้นค่อนข้างต่ำเมื่อเทียบกับอัลกอริทึมอื่น เมื่อนำไปใช้ในการเข้ารหัสวิดีโอจะทำให้คุณภาพของภาพด้อยกว่าวิธีอื่น ๆ โดยเฉพาะในกรณีทีลำดับของภาพอ้างอิง และภาพปัจจุบันอยู่ห่างกันมาก ดังนั้นเมื่อนำวิธีนี้ไปใช้งานจริง ถ้าต้องการให้คุณภาพของภาพออกมาดี จึงต้องมีการเก็บภาพอ้างอิงในความถี่ที่สูงขึ้น ซึ่งทำให้ขนาดของข้อมูลที่ได้ออกมาใหญ่กว่าปกติ



ภาพที่ 2: ขั้นตอนการเปรียบเทียบด้วยวิธี 2SS ระยะค้นหา 8 พิกเซล



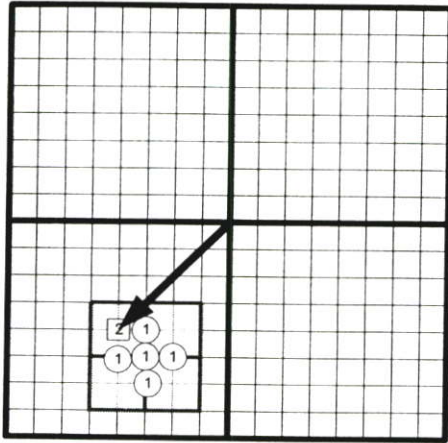
ภาพที่ 3: ขั้นตอนการเปรียบเทียบด้วยวิธี PHODS ระยะ 8 พิกเซล

3. อัลกอริทึม Two-Level Parallel Hierarchical One-Dimensional Search (2LPHODS)

อัลกอริทึม PHODS เป็นอัลกอริทึมที่มีจุดเด่นในแง่ของความเร็ว มีจำนวนพิกเซลที่ต้องทำการเปรียบเทียบน้อยมาก เมื่อเทียบกับอัลกอริทึมอื่น ๆ นอกจากนั้นยังมีลักษณะที่เป็นอัลกอริทึมแบบขนานซึ่งเมื่อนำไปสร้างเป็นวงจรดิจิทัล โดยใช้สถาปัตยกรรมที่เป็นแบบขนาน ก็จะช่วยให้ได้ความเร็วในการคำนวณที่สูงมากขึ้น แม้ว่าอัลกอริทึมนี้ยังมีจุดด้อยในแง่ของความถูกต้อง ซึ่งทำให้เมื่อนำไปใช้งานจะได้คุณภาพของภาพที่ต่ำกว่าอัลกอริทึมอื่น ๆ ที่นิยมใช้กันทั่วไป ซึ่งถ้าหากปรับปรุงในด้านของความถูกต้องให้กับอัลกอริทึมดังกล่าว โดยที่ไม่สูญเสียจุดเด่นของอัลกอริทึมเดิมมากเกินไป ก็จะได้อัลกอริทึมที่เหมาะสมสำหรับนำมาสร้างเป็นวงจรดิจิทัล เพื่อนำไปสร้างเป็นหน่วยประมวลผลเฉพาะทางสำหรับการประมวลการเคลื่อนไหว

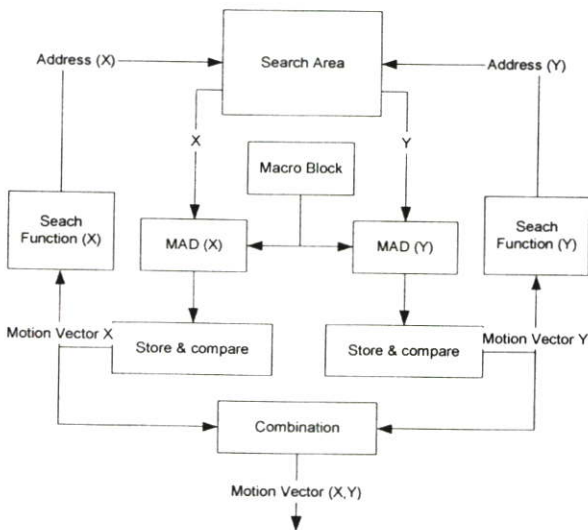
อัลกอริทึม 2LPHODS เป็นการปรับปรุงอัลกอริทึม PHODS ในด้านของความถูกต้องของผลลัพธ์ ให้สูงขึ้นกว่าเดิม โดยการเพิ่มจำนวนพิกเซลที่ต้องเปรียบเทียบอีกเล็กน้อยจากอัลกอริทึมเดิม ทำให้ความเร็วลดลงเพียงเล็กน้อยในขณะที่ความถูกต้องสูงขึ้น ภาพรวมของอัลกอริทึมนี้จะเป็นการหาเวกเตอร์การเคลื่อนไหวด้วยอัลกอริทึม PHODS เป็นจำนวน 2 ครั้ง ในพื้นที่ค้นหาที่มีขนาดต่างกัน และมีจุดเริ่มต้นที่ต่างกัน ซึ่งในที่นี้เรียกว่าเป็นระดับที่ 1 และระดับที่ 2 ในระดับที่ 1 นั้นจะทำการเปรียบเทียบด้วยวิธีการ PHODS ตามปกติ ซึ่งเมื่อได้ผลลัพธ์จากการค้นหาในระดับที่ 1 ด้วยวิธีการ PHODS แล้ว จะใช้ค่าผลลัพธ์ดังกล่าว เพื่อทำการกำหนดขอบเขตการค้นหาในระดับที่ 2 ต่อไป

ในการค้นหาในระดับที่ 2 จะใช้ผลลัพธ์จากระดับที่ 1 เป็นจุดเริ่มต้น จากนั้นจึงกำหนดขอบเขตค้นหาขนาดเล็กรอบจุดดังกล่าว แล้วทำการเปรียบเทียบด้วยวิธีการ PHODS อีกครั้งหนึ่ง เส้นเวกเตอร์จากจุดเริ่มต้นในระดับที่ 1 ไปยังผลลัพธ์ของระดับที่ 2 คือเวกเตอร์การเคลื่อนไหวของวิธีการ 2LPHODS ดังภาพที่ 4



ภาพที่ 4: การเปรียบเทียบด้วยวิธี 2LPHODS ระยะ 8 พิกเซล

ในส่วนของการนำไปสร้างเป็นวงจรดิจิทัลเพื่อนำไปใช้เป็นหน่วยประมวลผลเฉพาะทางนั้น จะใช้โครงสร้างการประมวลผลที่แยกค่าแกน X และแกน Y ออกจากกัน แล้วจึงนำผลลัพธ์ที่ได้มารวมกันในภายหลังดังภาพที่ 5 ซึ่งการออกแบบบนพื้นฐานของแผนผังดังกล่าวจะช่วยเพิ่มความเร็วในการค้นหาด้วยวิธีการนี้เป็นอย่างมาก



ภาพที่ 5: แผนภาพการประมวลผลแบบขนานของ 2LPHODS

4. ผลการทดลอง

การเปรียบเทียบประสิทธิภาพของอัลกอริทึมการประมวลการเคลื่อนไหวนั้น ประกอบไปด้วยการเปรียบเทียบในด้านของความเร็ว พิจารณาจากจำนวนพิกัดที่ต้องทำการเปรียบเทียบ และการเปรียบเทียบในด้านของความถูกต้องของผลลัพธ์ โดย

จะใช้ค่า PSNR (Peak Signal to Noise Ratio) ซึ่งเป็นค่าที่ใช้แสดงความถูกต้องของผลลัพธ์ มีหน่วยเป็นเดซิเบล (dB)

ข้อมูลที่ใช้ทดสอบจะใช้ลำดับภาพมาตรฐานจำนวน 4 ชุดที่มีลักษณะของการเคลื่อนไหวกว้างที่แตกต่างกัน ได้แก่ Akiyo เป็นลำดับภาพที่มีพื้นหลังนิ่ง มีการเปลี่ยนแปลงระหว่างภาพน้อย, Coastguard เป็นลำดับภาพที่มีพื้นหลังเคลื่อนไหวดลตลอดเวลา การเคลื่อนที่ของวัตถุค่อนข้างเร็ว, Mobile เป็นลักษณะของลำดับภาพที่มีการเคลื่อนที่ของวัตถุ 3 รูปแบบในภาพ คือมีการเคลื่อนที่ในแนวตั้ง การเคลื่อนที่ในแนวอน และการเคลื่อนที่แบบหมุน และ Tempete เป็นลำดับภาพที่มีการเคลื่อนที่แบบบนลงล่างอย่างรวดเร็วทั่วทั้งภาพ และมีการชุกก้องถอยห่างออกมา ลำดับภาพทั้งหมดที่ใช้เป็นมาตรฐาน CIF ซึ่งมีขนาด 352 X 288 พิกเซล

อัลกอริทึมที่นำมาใช้เปรียบเทียบมีอยู่ด้วยกัน 4 อัลกอริทึมคือ FS เป็นอัลกอริทึมที่ให้ผลลัพธ์ดีที่สุด, 2SS เป็นอัลกอริทึมที่มีผู้นำเสนอก่อนหน้า, PHODS เป็นอัลกอริทึมที่เป็นต้นแบบ และ 2LPHODS เป็นอัลกอริทึมที่เสนอในบทความนี้

ตารางที่ 1 แสดงการเปรียบเทียบจำนวนพิกัดที่ต้องทำการเปรียบเทียบของภาพที่มาตรฐาน CIF และอัตราส่วนเมื่อเทียบกับวิธี FS

ตารางที่ 1: จำนวนพิกัดที่ต้องเปรียบเทียบ

	FS	2SS	PHODS	2LPHODS
CIF	114444	13464	5148	8316
Ratio	100%	11.76%	4.50%	7.27%

ในส่วนของการวัดความถูกต้องของผลลัพธ์ จะใช้ลำดับภาพที่ 1 เป็นภาพอ้างอิง จากนั้นจึงเปรียบเทียบผลลัพธ์กับลำดับภาพที่ 2 ถึงลำดับภาพที่ 10 เพื่อให้เห็นถึงทิศทางการเปลี่ยนแปลง ดังภาพที่ 6 ถึง 9

จากผลลัพธ์ดังกล่าวจะเห็นได้ว่า 2LPHODS มีจำนวนพิกัดที่ต้องเปรียบเทียบมากกว่าอัลกอริทึมต้นแบบ แต่น้อยกว่าวิธีการ 2SS ในส่วนของค่า PSNR ซึ่งเป็นค่าความถูกต้องของผลลัพธ์จะเห็นได้ว่าเมื่อระยะห่างระหว่างภาพมีมากขึ้น ความถูกต้องของ PHODS จะมีแนวโน้มต่ำลงอย่างรวดเร็วเมื่อเทียบกับอัลกอริทึมอื่นที่นำมาเปรียบเทียบโดยเฉพาะในภาพที่มีการเปลี่ยนแปลงเป็นจำนวนมากอย่างภาพ Mobile และ Tempete แต่ในอัลกอริทึมที่ปรับปรุงขึ้นมานั้นแนวโน้มการเกิดความ

ผิดพลาดจะมีน้อยกว่ามาก แต่ไม่น้อยเท่ากับวิธีการ 2SS ซึ่งเมื่อพิจารณาทั้งในแง่ของความเร็ว และความถูกต้องแล้ว อัลกอริทึม 2LPHODS จัดได้ว่าเป็นอัลกอริทึมที่มีความสมดุลในด้านของความเร็วและความถูกต้อง นอกจากนี้ยังมีการคำนวณที่ไม่ซับซ้อนในส่วนของการคำนวณหาตำแหน่งนั้น มีแค่การบวก ลบ และหารเท่านั้น โดยการหารเป็นเพียงแค่การหาร 2 ซึ่งเพื่อนำไปสร้างเป็นวงจรดิจิทัลแล้วเป็นเพียงแค่การเลื่อนบิต และยังรองรับการประมวลผลแบบขนานอีกด้วย

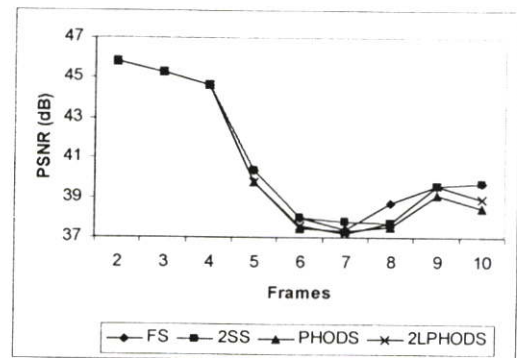
5. สรุป

บทความนี้นำเสนอการปรับปรุงอัลกอริทึมสำหรับประมวลผลการเคลื่อนไหวสำหรับการนำไปสร้างเป็นหน่วยประมวลผลเฉพาะทาง เพื่อนำไปใช้เพิ่มความเร็วให้การเข้ารหัสวิดีโอในมาตรฐานการบีบอัดภาพวิดีโอหลายมาตรฐานที่ใช้กันในปัจจุบัน สำหรับระบบประมวลผลที่มีความเร็วไม่สูงมาก โดยอัลกอริทึมดังกล่าวมีจุดเด่นในด้านของความเร็วที่สูง ในขณะที่ความผิดพลาดของผลลัพธ์มีไม่มากเกินไป อีกทั้งยังมีการคำนวณที่ไม่ซับซ้อน และรองรับการประมวลผลแบบขนาน

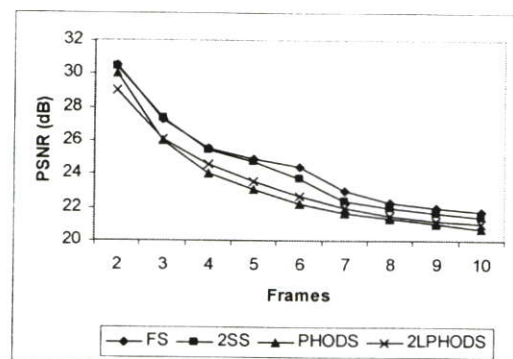
สำหรับการวิจัยในอนาคตสามารถพัฒนาเพิ่มฟังก์ชันพื้นฐานที่ใช้ในการบีบอัดภาพวิดีโอ เช่น DCT/IDCT เข้าไปในหน่วยประมวลผลเฉพาะทางให้พัฒนากลายเป็นหน่วยประมวลผลวิดีโอ (Video Processing Unit) ทำให้มีประโยชน์ในการใช้งานมากยิ่งขึ้น

6. เอกสารอ้างอิง

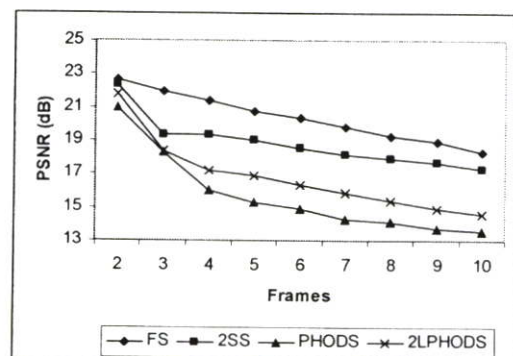
- [1] Borko Furht, Joshua Greenberg, Raymond Westwater, "Motion Estimation Algorithms for Video Compression". Massachusetts : Kluwer Academic Publishers, 1997.
- [2] Liang-Gee Chen, Wai-Ting Chen, Yeu-Shen Jehng, Tzi-Dar Chiuch "An Efficient Parallel Motion Estimation Algorithm for Digital Image Processing" *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.1, Issue 4, pp.378 – 385, Dec. 1991.
- [3] Nan Yu, Kichul Kim, Salcic Z., "A New Motion Estimation Algorithm for Mobile Real-Time Video and its FPGA Implementation" *TENCON*, Vol. 1, pp 383 – 386, Nov. 2004.
- [4] Rangarajan P., Prashanth G., Harish P.S. "FPGA Implementation of High Speed Parallel Architecture for Block Motion Estimation" *SIPS*, pp. 245 – 250, 2004.



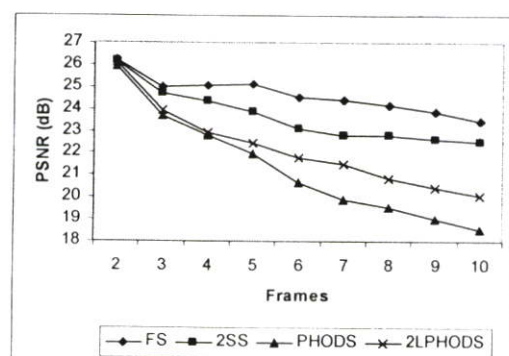
ภาพที่ 6: ค่า PSNR ของลำดับภาพ Akiyo



ภาพที่ 7: ค่า PSNR ของลำดับภาพ Coastguard



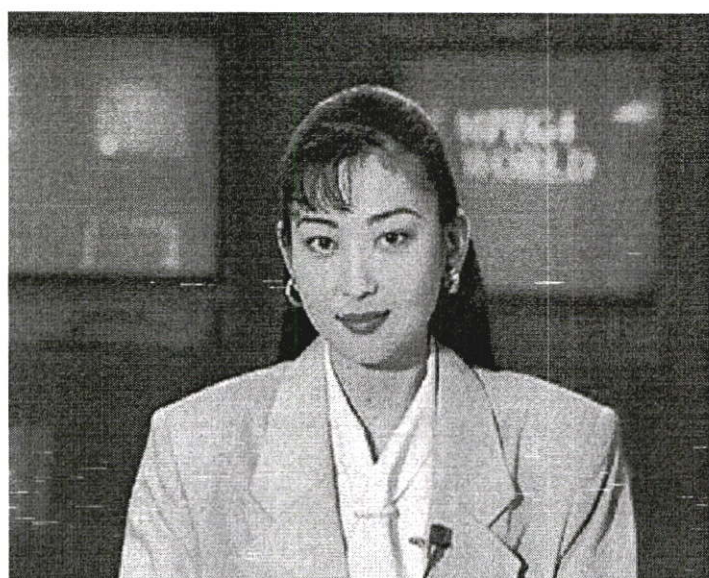
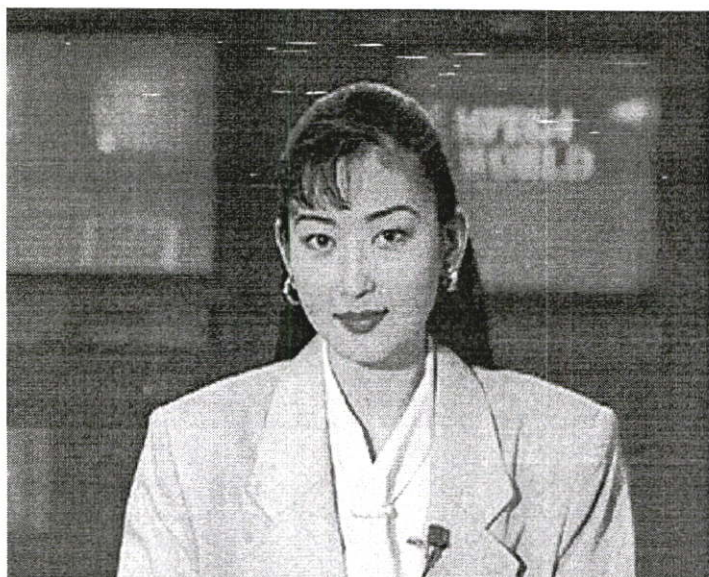
ภาพที่ 8: ค่า PSNR ของลำดับภาพ Mobile



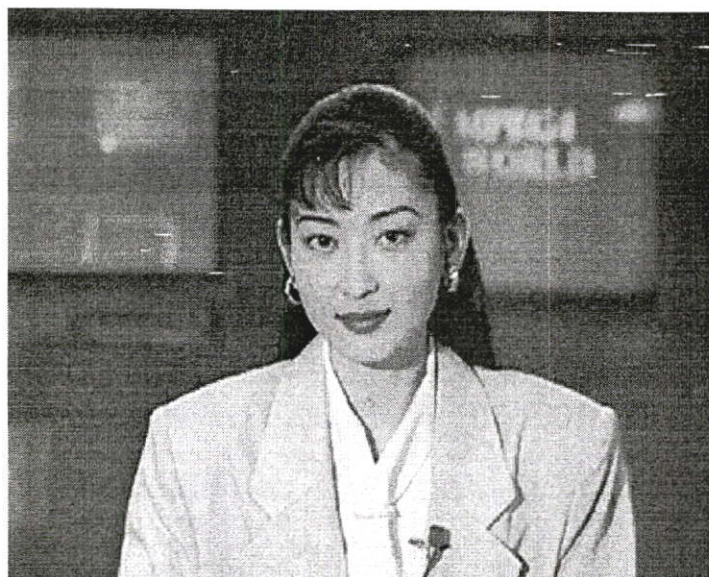
ภาพที่ 9: ค่า PSNR ของลำดับภาพ Tempete

ภาคผนวก ข.

ลำดับภาพมาตรฐานลำดับภาพที่ 1 ถึง 10 ที่ใช้ในงานวิจัย



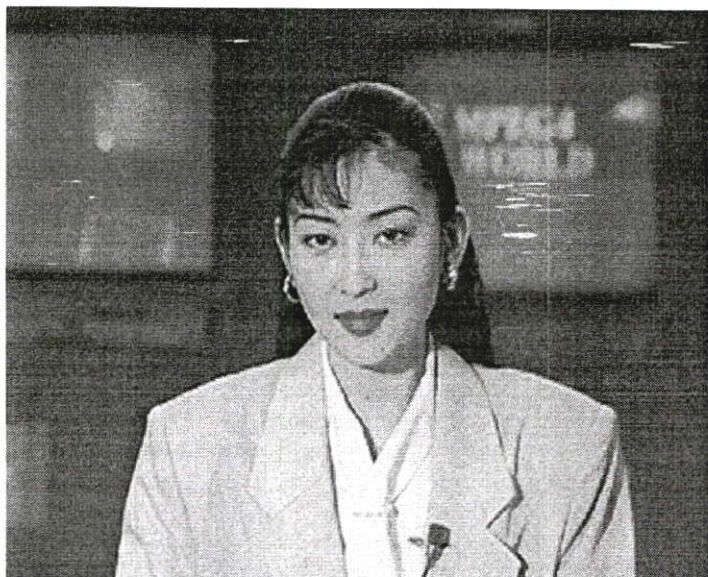
รูปที่ ข.1 ภาพเคลื่อนไหว akiyo ลำดับภาพที่ 1-3



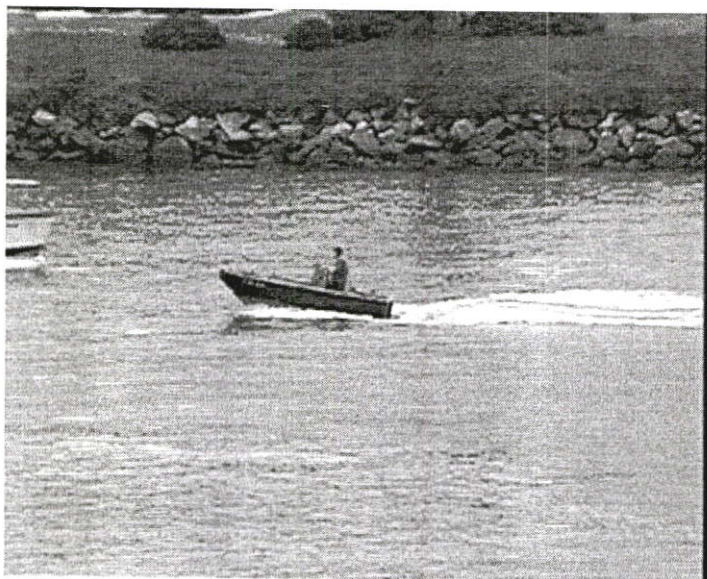
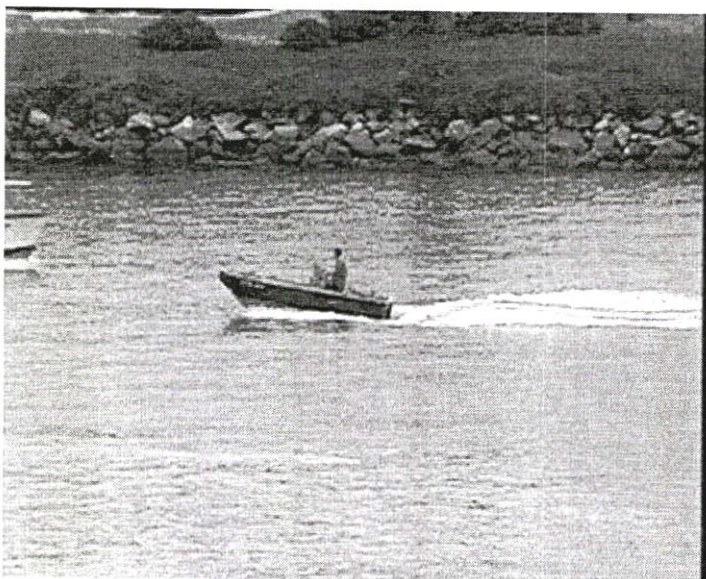
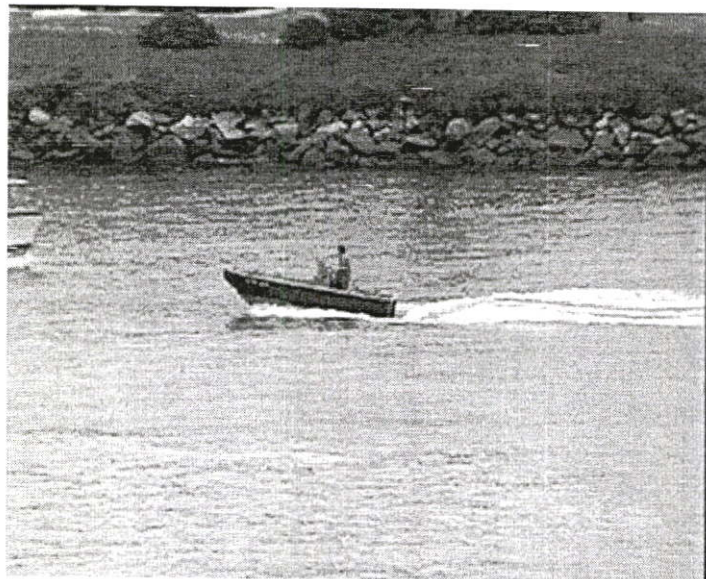
รูปที่ ข.2 ภาพเคลื่อนไหว akiyo ลำดับภาพที่ 4-6



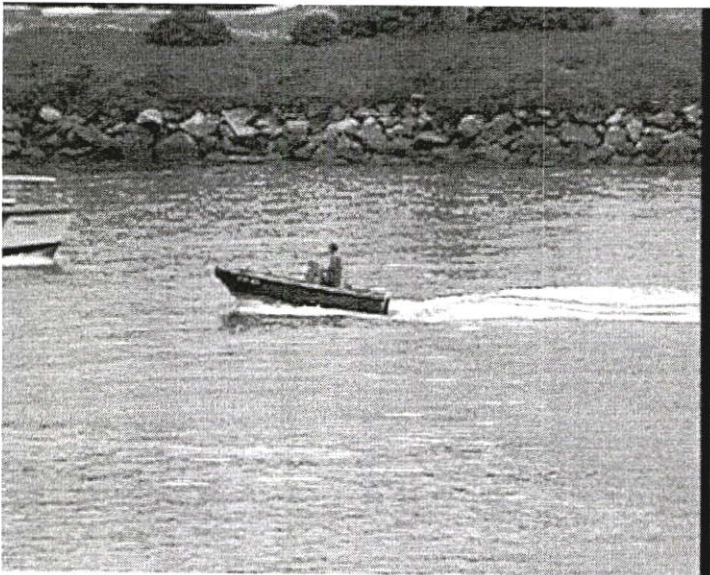
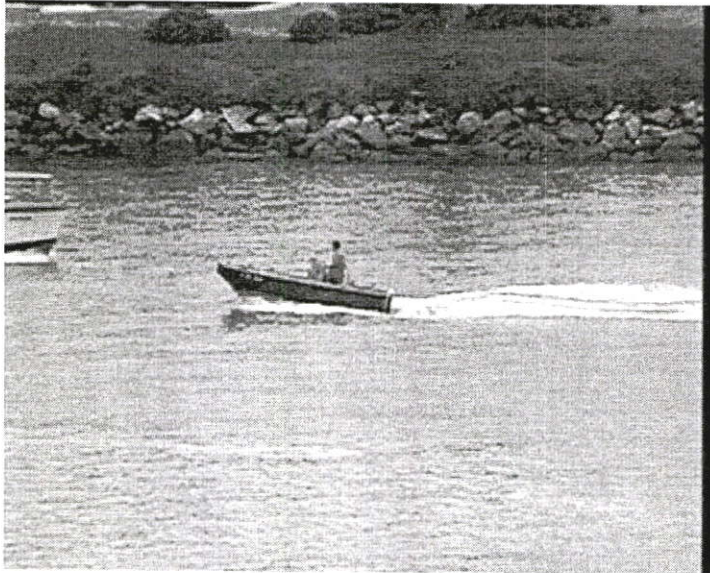
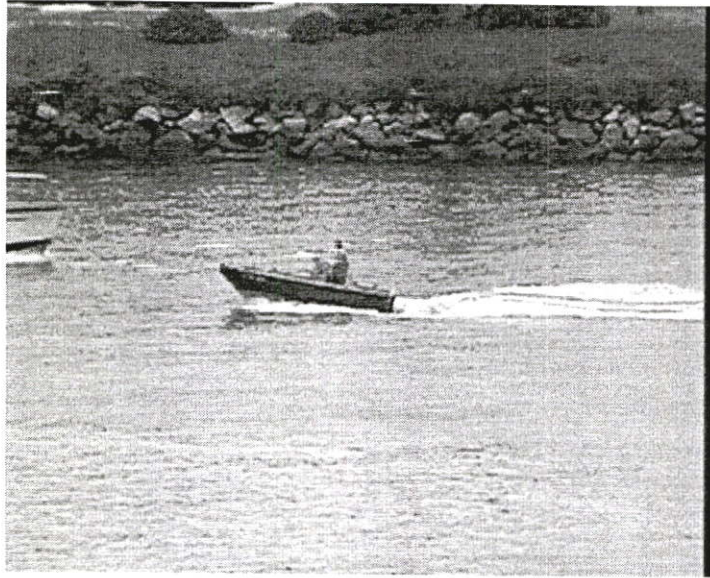
รูปที่ ข.3 ภาพเคลื่อนไหว akiyo ลำดับภาพที่ 7-9



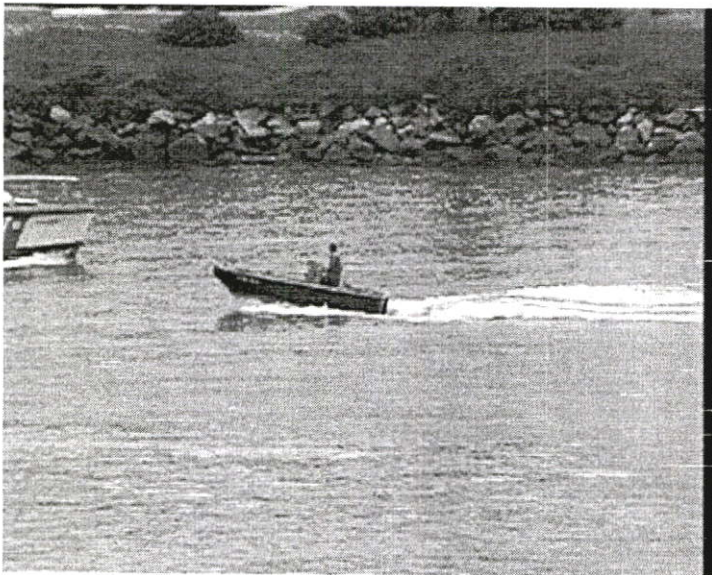
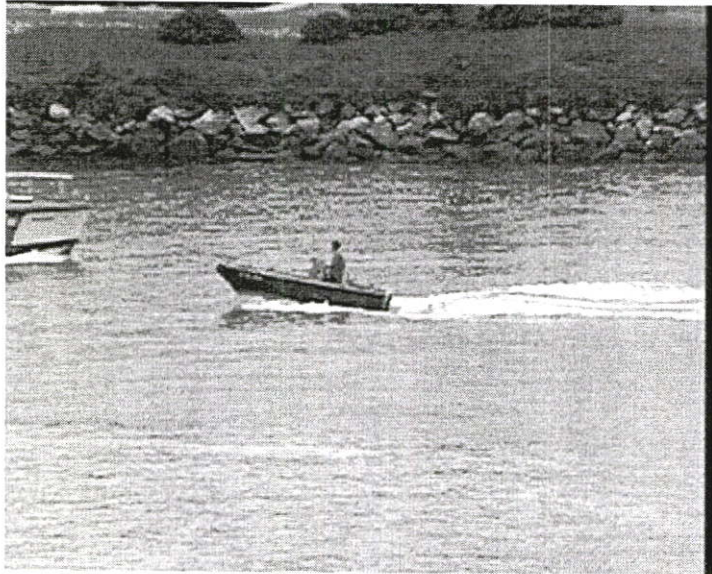
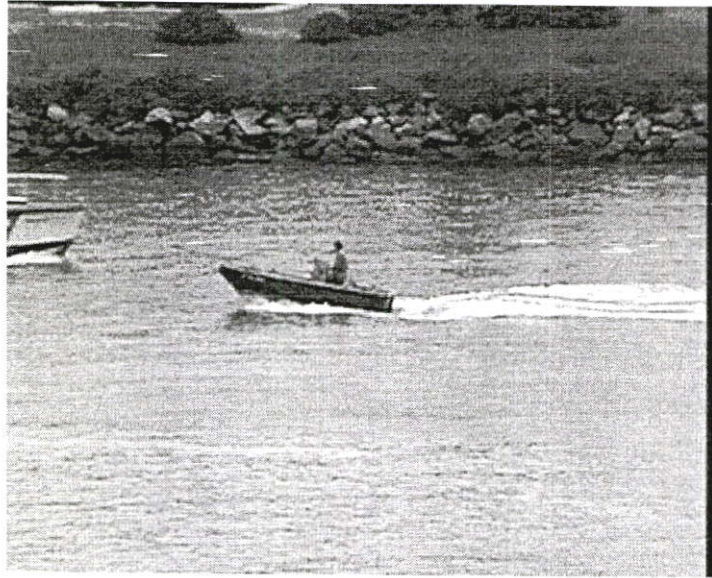
รูปที่ ข.4 ภาพเคลื่อนไหว akiyo ลำดับภาพที่ 10



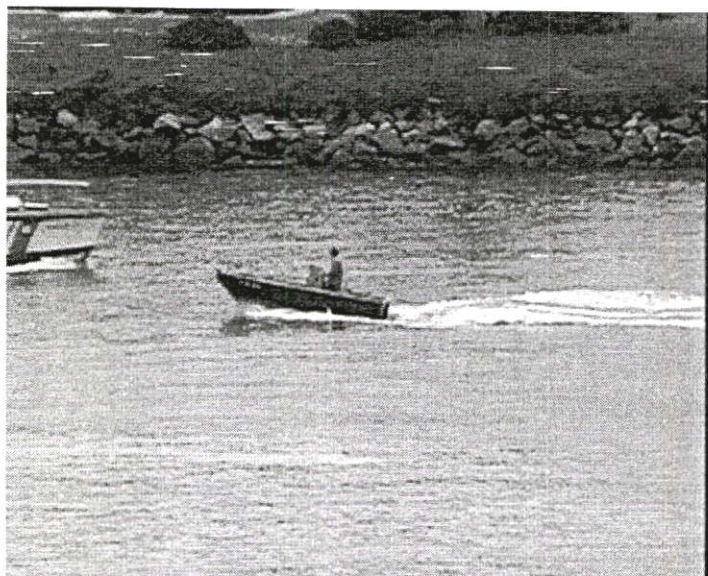
รูปที่ ข.5 ภาพเคลื่อนไหว coastguard ลำดับภาพที่ 1-3



รูปที่ ข.6 ภาพเคลื่อนไหว coastguard ลำดับภาพที่ 4-6



รูปที่ ข.7 ภาพเคลื่อนไหว coastguard ลำดับภาพที่ 7-9



รูปที่ ข.8 ภาพเคลื่อนไหว coastguard ลำดับภาพที่ 10



รูปที่ ข.9 ภาพเคลื่อนไหว container ลำดับภาพที่ 1-3



รูปที่ ข.10 ภาพเคลื่อนไหว container ดำดပ်ภาพที่ 4-6



รูปที่ ข.11 ภาพเคลื่อนไหว container ลำดับภาพที่ 7-9



รูปที่ ข.12 ภาพเคลื่อนไหว container ลำดับภาพที่ 10



รูปที่ ข.13 ภาพเคลื่อนไหว mobile ลำดับภาพที่ 1-3



รูปที่ ข.14 ภาพเคลื่อนไหว mobile ลำดับภาพที่ 4-6



รูปที่ ข.15 ภาพเคลื่อนไหว mobile ลำดับภาพที่ 7-9



รูปที่ ข.16 ภาพเคลื่อนไหว mobile ลำดับภาพที่ 10



รูปที่ ข.17 ภาพเคลื่อนไหว tempete ลำดับภาพที่ 1-3



รูปที่ ข.18 ภาพเคลื่อนไหว tempete ลำดับภาพที่ 4-6



รูปที่ ข.19 ภาพเคลื่อนไหว tempete ลำดับภาพที่ 7-9



รูปที่ ข.20 ภาพเคลื่อนไหว tempete ลำดับภาพที่ 10

ประวัติผู้เขียน

ชื่อ	นาย มานะ กรีษประพันธ์
เกิดวันที่	14 พฤษภาคม 2523 ที่จังหวัด ฉะเชิงเทรา
ประวัติ	จบการศึกษาระดับมัธยมศึกษาตอนปลายสายวิทยาศาสตร์ โรงเรียนพุทธรังสีพิบูล จังหวัดฉะเชิงเทรา ปี พ.ศ. 2541เกรดเฉลี่ย 3.44 จบการศึกษาระดับปริญญาตรี สำนักวิชาวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรม คอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ปี พ.ศ. 2545เกรดเฉลี่ย 2.80