



การประยุกต์ใช้สเตอริโอวิชันและการประมวลผลแบบขนาน  
เพื่อช่วยเหลือในการเดินทางของผู้พิการทางสายตา  
**Applying Stereo Vision and Parallel Computing  
for Supporting the Journey of the Visually Impaired**

ธนาธิป ลิ้มนา  
**Thanathip Limna**

วิทยานิพนธ์นี้สำหรับการศึกษาตามหลักสูตรปริญญา  
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์  
มหาวิทยาลัยสงขลานครินทร์

**A Thesis Submitted in Fulfillment of the Requirements for the Degree of  
Master of Engineering in Computer Engineering  
Prince of Songkla University**

**2553**

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

ชื่อวิทยานิพนธ์      การประยุกต์ใช้สเตอริโอไอซันและการประมวลผลแบบขนานเพื่อช่วยเหลือใน  
 การเดินทางของผู้พิการทางสายตา

ผู้เขียน                นายธนชาติ ลิ้มนา

สาขาวิชา              วิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

คณะกรรมการสอบ

.....  
 (ผู้ช่วยศาสตราจารย์ ดร.พิชญา ตันฑัยย์)

.....ประธานกรรมการ  
 (ผู้ช่วยศาสตราจารย์ ดร.พรชัย พุกภัยภัทรานนต์)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

.....กรรมการ  
 (ผู้ช่วยศาสตราจารย์ ดร.พิชญา ตันฑัยย์)

.....  
 (ดร.นิคม สุวรรณวรร)

.....กรรมการ  
 (ดร.นิคม สุวรรณวรร)

.....กรรมการ  
 (ดร.ประกาศิต กายะสิทธิ์)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้รับวิทยานิพนธ์ฉบับนี้  
 สำหรับการศึกษา ตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรม  
 คอมพิวเตอร์

.....  
 (รองศาสตราจารย์ ดร.เกริกชัย ทองหนู)  
 คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์	การประยุกต์ใช้สเตอริโอวิชันและการประมวลผลแบบขนานเพื่อช่วยเหลือในการเดินทางของผู้พิการทางสายตา
ผู้เขียน	นายธนาริพ ลิ้มนา
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	2552

### บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอระบบตรวจสอบสิ่งกีดขวางสำหรับการเดินทางของผู้พิการทางสายตาโดยใช้เทคนิคทางการประมวลผลภาพสเตอริโอ โดยจะเน้นส่วนตรวจจับวัตถุสิ่งกีดขวางด้วยกล้องสเตอริโอและประมาณระยะทางของสิ่งกีดขวางเท่านั้น เทคนิคทางการประมวลผลภาพสเตอริโอที่ใช้ในวิทยานิพนธ์นี้คือ Depth discontinuities by pixel-to-pixel stereo โดยที่ผลลัพธ์จากการประมวลผลภาพสเตอริโอนั้นสามารถใช้ตรวจหาตำแหน่งสิ่งกีดขวางในการเดินทางได้ด้วยการแปลงข้อมูลให้อยู่ในรูปของพิกัด  $u$  และ  $v$  โดยที่นำข้อมูลที่ได้จากพิกัด  $u$  และ  $v$  ไปแทนค่าในสมการเพื่อประมาณค่าระยะทางต่อไป สำหรับการประมาณระยะทางจากสิ่งกีดขวางถึงตัวผู้พิการทางสายตานั้น วิทยานิพนธ์นี้เสนอสมการที่ได้สร้างขึ้นมาโดยเฉพาะสำหรับกล้องสเตอริโอราคาถูกลงที่สร้างขึ้นเองจากเว็บแคมยี่ห้อ Logitech มีลักษณะเป็นกล้องสเตอริโอแบบขนานมีระยะห่างระหว่างกล้อง 12 เซนติเมตร ระบบสามารถตรวจจับวัตถุได้ดีในระยะ 1-5 เมตร และระดับคิสพาริตีที่เหมาะสมอยู่ที่ 50 ถึง 100 เนื่องจากการประมวลผลภาพสเตอริโอที่ใช้ในวิทยานิพนธ์นี้ใช้เวลานานจึงปรับปรุงอัลกอริทึมการประมวลผลภาพสเตอริโอให้สามารถทำงานได้เร็วขึ้นโดยเสนอเทคนิคการประมวลผลแบบขนานที่เหมาะสมด้วยวิธีการส่งผ่านข้อความบนคอมพิวเตอร์ที่มีหน่วยประมวลผลหลายแกน จากการทดสอบระบบด้วยการประมวลผลแบบขนานบนคอมพิวเตอร์ที่มีหน่วยประมวลผลสองแกน โดยใช้ภาพขนาด 320x240 พิกเซล ที่จำนวนระดับคิสพาริตีที่สูงที่สุด 100 มีค่าสปีดอัปเป็น 1.936 และค่าประสิทธิภาพในการประมวลผล 0.968 ในขณะที่ทดสอบบนคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วย ค่าสปีดอัปคิดเป็น 1.930, 3.603 และ 6.567 ค่าประสิทธิภาพในการประมวลผลคิดเป็น 0.965, 0.901 และ 0.821 เมื่อใช้โปรเซสจำนวน 2, 4 และ 8 โพรเซสตามลำดับ

**คำสำคัญ** ผู้พิการทางสายตา, การประมวลผลภาพสเตอริโอ, การประมวลผลแบบขนาน, การตรวจจับสิ่งกีดขวาง

**Thesis Title**            Applying Stereo Vision and Parallel Computing for supporting the journey of the Visually Impaired

**Author**                    Mr. Thanathip Limna

**Major Program**        Computer Engineering

**Academic Year**        2009

## **ABSTRACT**

This thesis presents an obstacles detection system for supporting the journey of the visually impaired applying a stereo vision technique that focuses on detecting obstacles using stereo cameras and approximating the obstacle distances. The stereo image processing technique used in this thesis is the depth discontinuities by pixel-to-pixel stereo. The results from the stereo image processing can detect the positions of obstacles by transforming the disparity map into the u and v coordinates systems where the data can be used to approximate the distance equation. The approximate distances from the obstacles to the visually impaired can be detected using the equation specifically proposed for the stereo cameras created from low cost Logitech web cams which have the base line of 12 cm. Our obstacle detection system can accurately detect obstacles in the range of 1 to 5 meters further from the visually impaired and the disparity level range is from 50 to 100. The response time for computing the stereo image processing algorithm used in this thesis has been improved by employing parallel processing technique using messages passing interface tested on computers with multi-core processors. Applying parallel processing along with the image size of 320x240 pixels and the maximum amount disparity of 100, the speedup is 1.936 and the efficiency is 0.968 on a two-core personal computer and the speedups on an 8-core server using 2, 4 and 8 processes are 1.930, 3.603 and 6.567, and the efficiencies are 0.965, 0.901 and 0.821 respectively.

**Keywords:** the visually impaired, stereo image processing, parallel computing, obstacles detection

## กิตติกรรมประกาศ

งานวิจัยและวิทยานิพนธ์นี้สำเร็จลุล่วงลงได้ต้องขอกราบขอบพระคุณ ผู้ที่ให้คำปรึกษา แนะนำและเสนอแนะข้อบกพร่องต่างๆ มีรายชื่อดังต่อไปนี้ ผู้ช่วยศาสตราจารย์ ดร. พิชญาดัณห์ชัย ประธานกรรมการที่ปรึกษาวิทยานิพนธ์ ที่ได้ให้คำแนะนำวิธีการดำเนินงานวิจัยต่างๆ ซึ่งแนะข้อผิดพลาดระหว่างการดำเนินงาน ให้คำปรึกษาเกี่ยวกับการประมวลแบบขนานและการประมวลผลภาพเพื่อก่อให้เกิดประโยชน์แก่ผู้พิการทางสายตา ตลอดจนตรวจสอบและแก้ไขจุดผิดพลาดต่างๆ ในวิทยานิพนธ์เล่มนี้จนสมบูรณ์

ดร. นิคม สุวรรณวร กรรมการที่ปรึกษาวิทยานิพนธ์ที่ได้แนะนำและเสนอแนะเทคนิคทางด้านสเตอริโอวิชัน และจุดประกายให้สนใจประยุกต์ใช้เทคโนโลยีนี้เพื่อช่วยในการเดินทางสำหรับผู้พิการทางสายตา ผู้ช่วยศาสตราจารย์ ดร. พรชัย พฤกษ์ภัทรานนท์ กรรมการสอบวิทยานิพนธ์ และ ดร. ประภาสิต กายะสิทธิ์ จากศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ กรรมการสอบวิทยานิพนธ์ที่ให้คำแนะนำและตรวจสอบวิทยานิพนธ์เพื่อความสมบูรณ์แห่งวิทยานิพนธ์ฉบับนี้

ขอกราบขอบพระคุณภาควิชาวิศวกรรมคอมพิวเตอร์ และคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ที่สนับสนุนทุนค่าเล่าเรียนในระดับบัณฑิตศึกษา และค่าใช้จ่ายในการนำเสนอและเผยแพร่ผลงานในที่ประชุมทางวิชาการในระดับนานาชาติ ตลอดจนบัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ ที่มอบทุนอุดหนุนงานวิจัยเพื่อวิทยานิพนธ์นี้

ขอกราบขอบพระคุณครูบาอาจารย์ที่มีได้กล่าวถึงที่ให้ความรู้และมีส่วนให้วิทยานิพนธ์นี้เสร็จสมบูรณ์ ตลอดจน ญาติมิตร ที่ให้กำลังใจและสนับสนุนการศึกษามาด้วยดี ทำให้นี้ขอกราบขอบพระคุณบิดา มารดา ที่ให้การอบรมเลี้ยงดูและส่งเสริมการศึกษาเป็นอย่างดีตลอดมาในอดีต จนทำให้ผู้วิจัยประสบความสำเร็จในชีวิตตลอดมา

ธนชาติ ลิ้มนา

## สารบัญ

บทคัดย่อ.....	(3)
ABSTRACT.....	(4)
กิตติกรรมประกาศ.....	(5)
สารบัญ .....	(6)
รายการภาพประกอบ .....	(10)
รายการตาราง.....	(14)
สัญลักษณ์คำย่อและตัวย่อ.....	(16)
ศัพท์บัญญัติ.....	(17)
บทที่ 1 บทนำ .....	1
1.1. ความสำคัญและที่มาของวิทยานิพนธ์ .....	1
1.2. การตรวจเอกสาร .....	3
1.2.1. เทคโนโลยีที่ใช้ช่วยเหลือในการเดินทางของผู้พิการทางสายตา.....	3
1.2.2. สเตอริโอวิชั่น.....	4
1.2.3. การประมวลผลแบบขนาน.....	5
1.3. วัตถุประสงค์ .....	6
1.4. ขอบเขตของการวิจัย.....	6
1.5. ประโยชน์ที่คาดว่าจะได้รับ .....	6
บทที่ 2 ทฤษฎีและหลักการ .....	7
2.1. การประมวลผลภาพแบบสเตอริโอ .....	7
2.1.1. ลักษณะทางเรขาคณิตของ Binocular stereo.....	7
2.1.2. การจับคู่ภาพสเตอริโอ.....	9

## สารบัญ (ต่อ)

2.1.2.1. Intensity-based stereo matching technique.....	9
2.1.2.2. Feature-based stereo matching technique.....	10
2.1.3. Depth Discontinuities by Pixels-to-Pixels Stereo.....	11
2.2. การประมาณระยะทางของวัตถุจากแผนผังคิสพาริตี.....	14
2.2.1. U-V-disparity.....	15
2.2.2. การประมาณระยะทางของวัตถุด้วย V-disparity .....	17
2.3. การประมวลผลแบบขนาน.....	18
2.3.1. มิติในการประมวลผลแบบขนาน .....	19
2.3.2. การวัดประสิทธิภาพของการประมวลผลแบบขนาน .....	19
2.3.3. Message Passing Interface.....	20
2.3.4. MPICH .....	22
2.4. สรุป .....	23
บทที่ 3 การออกแบบและพัฒนาระบบ .....	25
3.1. ภาพรวมของระบบ.....	25
3.2. การออกแบบระบบ .....	25
3.2.1. กล้องสเตอริโอวิชัน.....	26
3.2.2. การประมวลผลภาพสเตอริโอ .....	29
3.2.3. การประมาณค่าระยะทางสิ่งกีดขวาง .....	31
3.3. การออกแบบอัลกอริทึมการประมวลผลแบบขนาน.....	32
3.3.1. การออกแบบการประมวลผลแบบขนาน P2P ส่วนการจับคู่พิกเซลในแต่ละแถว .....	32
3.3.2. การออกแบบการประมวลผลแบบขนาน P2P ส่วนการประมวลผลขั้นปลาย .....	36
3.4. การประมาณค่าระยะทางของวัตถุในระบบตรวจจับวัตถุเพื่อผู้พิการทางสายตา .....	42

## สารบัญ (ต่อ)

3.4.1. สมการประมาณค่าระยะทางวัตถุ .....	42
3.4.2. การแปลผลข้อมูลจาก U และ V disparity .....	49
3.4.3. อัตราส่วนลักษณะของภาพ .....	54
3.5. สรุป .....	56
บทที่ 4 ผลการทดสอบและการวิเคราะห์ผล .....	57
4.1. ความเร็วการประมวลผล .....	57
4.1.1. คุณลักษณะของคอมพิวเตอร์ที่ใช้ทดสอบ .....	58
4.1.1.1. คอมพิวเตอร์ที่มีหน่วยประมวลผลสองแกน .....	58
4.1.1.2. คอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วย .....	58
4.1.2. การประมวลผลอัลกอริทึม P2P แบบตามลำดับ .....	59
4.1.2.1. เวลาการประมวลผลอัลกอริทึม P2P แบบตามลำดับบนคอมพิวเตอร์ที่มีหน่วย ประมวลผลสองแกน .....	59
4.1.2.2. เวลาการประมวลผลอัลกอริทึม P2P แบบตามลำดับบนคอมพิวเตอร์สมรรถนะ สูงที่มีหน่วยประมวลผล 8 หน่วย .....	61
4.1.3. การประมวลผลอัลกอริทึม P2P แบบขนานในกระบวนการจับคู่พิกเซลในแต่ละแถว .....	62
4.1.3.1. เวลาการประมวลผลแบบขนานของอัลกอริทึม P2P ในส่วนของการจับคู่พิกเซล ในแต่ละแถวบนคอมพิวเตอร์ที่มีหน่วยประมวลผลสองแกน .....	62
4.1.3.2. เวลาการประมวลผลแบบขนานของอัลกอริทึม P2P ส่วนการจับคู่พิกเซลในแต่ ละแถวบนคอมพิวเตอร์ที่มีหน่วยประมวลผล 8 หน่วย .....	64
4.1.4. การประมวลผลอัลกอริทึม P2P แบบขนานกระบวนการประมวลผลขึ้นปลาย .....	65
4.1.5. วิเคราะห์ผลการประมวลผลอัลกอริทึม P2P แบบขนาน .....	67
4.2. การประมาณค่าระยะทางและตรวจสอบตำแหน่งของวัตถุ .....	71
4.2.1. สภาพแวดล้อมจริงที่ทดสอบ .....	71



## สารบัญ (ต่อ)

4.2.2. สภาพแวดล้อมที่เป็นปัญหาสำหรับการตรวจสอบ.....	79
4.2.3. การทดสอบปรับอัตราส่วนลักษณะ .....	81
4.2.4. การปรับช่วงในแนวตั้งเพื่อหาวัตถุที่มีความกว้างขนาดเล็ก.....	86
4.3. วิเคราะห์ผลการทดสอบ .....	88
บทที่ 5 บทสรุป .....	91
5.1. สรุปผลการวิจัย .....	91
5.2. ข้อเสนอแนะ .....	94
เอกสารอ้างอิง .....	96
ภาคผนวก .....	100
ภาคผนวก ก. ไบรารี MPI ที่ใช้ในวิทยานิพนธ์ทั้งหมด .....	101
ภาคผนวก ข. ตัวอย่างภาพแผนผังคิซพาริตีที่ระยะทางต่างๆ .....	106
ภาคผนวก ค. ลักษณะโครงสร้างโปรแกรมพัฒนาขึ้นในวิทยานิพนธ์ .....	112
ภาคผนวก ง. ผลงานตีพิมพ์เผยแพร่จากวิทยานิพนธ์ .....	117
ประวัติผู้เขียน .....	129

## รายการภาพประกอบ

รูปที่ 2-1 กระบวนการจับคู่ภาพสเตอริโอซึ่งผลลัพธ์ที่ได้คือแผนผังคิสพาริตี.....	7
รูปที่ 2-2 ลักษณะทางเรขาคณิตของ Binocular stereo [11].....	8
รูปที่ 2-3 ความเข้มของภาพซ้ายและภาพขวา ณ เส้นกราดตรวจเดียวกัน.....	10
รูปที่ 2-4 ตัวอย่างพิกเซลที่เหมือนกัน จากภาพซ้ายที่ตรงกับภาพขวา.....	11
รูปที่ 2-5 ภาพรวมของการทำงานของอัลกอริทึม P2P [15] .....	13
รูปที่ 2-6 พิกัด $u$ และ $v$ ใหม่สำหรับแผนผังคิสพาริตี.....	14
รูปที่ 2-7 การแปลงแผนผังคิสพาริตีไปเป็น U-V-disparity โดยใช้ฟังก์ชัน $H$ และ $V$ .....	16
รูปที่ 2-8 ความสัมพันธ์ระหว่างวัตถุในแผนผังคิสพาริตีกับเส้นตรงใน U-V disparity .....	16
รูปที่ 2-9 สถาปัตยกรรมคอมพิวเตอร์แบบซิมเมตริกมัลติโปรเซสเซอร์ [16] .....	18
รูปที่ 3-1 ส่วนตรวจสอบวัตถุสิ่งกีดขวางด้วยกล้องสเตอริโอวิชันและประมาณระยะทางของสิ่งกีดขวาง .....	25
รูปที่ 3-2 กระบวนการประมวลผลภาพสเตอริโอ เริ่มจากจับภาพจากกล้องสเตอริโอ ผ่านอัลกอริทึมการประมวลผลภาพสเตอริโอจนได้แผนผังคิสพาริตี แล้วจึงนำแผนผังคิสพาริตีที่ได้ไปประมวลผลต่อตามที่ต้องการ .....	26
รูปที่ 3-3 โครงสร้างกล้องสเตอริโอที่ใช้สำหรับจับภาพสเตอริโอในวิทยานิพนธ์นี้โดยใช้ระยะห่างระหว่างกล้อง 12 เซนติเมตร.....	27
รูปที่ 3-4 คุณลักษณะของภาพที่ใช้ในการประมวลผลที่บันทึกจากกล้องเว็บแคม Logitech.....	27
รูปที่ 3-5 โครงสร้างกล้องสเตอริโอที่จัดทำขึ้นเพื่อทดสอบระบบโดยใช้เว็บแคมยี่ห้อ Logitech จำนวน 2 กล้องมีระยะห่างในระนาบแกน $x$ ขนาด 12 เซนติเมตร.....	28
รูปที่ 3-6 ตัวอย่างการนำกล้องสเตอริโอไปใช้กับผู้พิการ ซึ่งสามารถประดิษฐ์เป็นแว่นหรือที่ครอบศีรษะก็ได้ .....	28
รูปที่ 3-7 กล้องสเตอริโอที่จัดทำขึ้นเมื่อนำมาสวมใส่.....	28
รูปที่ 3-8 กราฟแสดงความสัมพันธ์ระหว่างเวลาที่ใช้ในการประมวลผลกับจำนวนระดับคิสพาริตีสูงสุดของ P2P โดยใช้ซอร์สโค้ดแบบตามลำดับซึ่งทำงานอยู่บนคอมพิวเตอร์ที่มีหน่วยประมวลผลแบบสองแกน ภาพขนาด 320x240 พิกเซล .....	30
รูปที่ 3-9 การทำงาน โดยรวมของกระบวนการจับคู่พิกเซลในแต่ละแถว.....	33

## รายการภาพประกอบ (ต่อ)

รูปที่ 3-10 การแบ่งงานออกเป็นสองโปรเซสให้ประมวลผลพร้อมกันเพื่อลดเวลาของผลตอบสนอง ใน ส่วนการจับคู่พิกเซลในแต่ละแถว.....	34
รูปที่ 3-11 ตัวอย่างการแบ่งงานใน ส่วนการจับคู่พิกเซลในแต่ละแถว.....	34
รูปที่ 3-12 ส่วนของโปรแกรมแสดงลักษณะการแบ่งงาน.....	35
รูปที่ 3-13 ขั้นตอนการทำงานและการไหลของข้อมูลในกระบวนการประมวลผลขั้นปลาย.....	37
รูปที่ 3-14 การแบ่งงานออกเป็นสองโปรเซสให้ประมวลผลพร้อมกันเพื่อลดเวลาของผลตอบสนอง ใน ส่วนการประมวลผลขั้นปลายในการคำนวณที่ใช้ข้อมูลทั้งคอลัมน์ .....	39
รูปที่ 3-15 การหาความสัมพันธ์ระหว่างระยะทางกับระดับดิสพาริตีด้วยสมการพหุนามกำลัง 4 ตาม สมการที่ (12).....	46
รูปที่ 3-16 ความสัมพันธ์ระหว่างระยะทางกับดิสพาริตีด้วยสมการพหุนามกำลัง 4 ตามสมการที่ (13) ที่คำนึงถึงบริเวณที่ปลอดภัยสำหรับการเดินทางของผู้พิการทางสายตา.....	49
รูปที่ 3-17 กระบวนการระบุตำแหน่งของวัตถุโดยใช้ U-disparity (ก) แผนผังดิสพาริตี (ข) แปลง แผนผังดิสพาริตีมาเป็น U-disparity (ค) เลือกระดับดิสพาริตีที่มีค่าระหว่าง 0 ถึง 35 (ง) นำระดับดิส พาริตีจาก (ค) มาประมาณระยะทางด้วยสมการ แล้วนำมาแสดงผล.....	51
รูปที่ 3-18 ตัวอย่างผลที่ได้จากกระบวนการประมาณค่าระยะทางและหาตำแหน่งของวัตถุ (ก) ภาพ ซ้าย (ข) U-disparity ที่สนใจ (ง) ค่าดิสพาริตีที่หาได้ในแต่ละช่วง (จ) ระยะทางที่ได้จากสมการ (13) (ฉ) แผนผังดิสพาริตี (ซ) แปลงผลจาก (จ) ให้คนสายตาปรกติเข้าใจได้ง่าย .....	52
รูปที่ 3-19 ใกล้เคียงของกระบวนการประมาณระยะทางสิ่งกีดขวาง .....	53
รูปที่ 3-20 ความแตกต่างระหว่างอัตราส่วน 4:3 รูป (ก-ง) และ 16:9 (จ-ซ).....	55
รูปที่ 3-21 การแปลงอัตราส่วนลักษณะ 4:3 ภาพวีดีโอมาตรฐานไปเป็น 16:9.....	55
รูปที่ 4-1 ภาพซ้ายและภาพขวาที่ใช้สำหรับการทดสอบ.....	57
รูปที่ 4-2 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ในลักษณะโปรแกรมแบบตามลำดับบน คอมพิวเตอร์ส่วนบุคคลที่มีหน่วยประมวลผลสองแกนเมื่อเปลี่ยนแปลงจำนวนระดับดิสพาริตี สูงสุด.....	59
รูปที่ 4-3 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ในลักษณะโปรแกรมแบบตามลำดับบน คอมพิวเตอร์ส่วนบุคคลที่มีหน่วยประมวลผลสองแกนเมื่อเปลี่ยนแปลงขนาดภาพ.....	60

## รายการภาพประกอบ (ต่อ)

รูปที่ 4-4 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ในลักษณะโปรแกรมแบบตามลำดับบนคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วย เมื่อเปลี่ยนแปลงจำนวนระดับคิซพารีที่สูงสุด.....	61
รูปที่ 4-5 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ในลักษณะโปรแกรมแบบตามลำดับบนคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วยเมื่อเปลี่ยนแปลงขนาดรูปภาพ.....	62
รูปที่ 4-6 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ส่วนการจับคู่พิกเซลในแต่ละแถวในลักษณะโปรแกรมแบบขนานบนคอมพิวเตอร์ส่วนบุคคลที่มีหน่วยประมวลผลสองแกน โดยใช้ 1 และ 2 โพรเซส.....	63
รูปที่ 4-7 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ในลักษณะโปรแกรมแบบขนานบนคอมพิวเตอร์ส่วนบุคคลที่มีหน่วยประมวลผล 2 หน่วยโดยใช้ 1, 2, 4 และ 8 โพรเซส.....	64
รูปที่ 4-8 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ส่วนของการจับคู่พิกเซลในแต่ละแถว ในลักษณะโปรแกรมแบบขนานบนคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วย ด้วยภาพขนาด 320x240 พิกเซล โดยใช้ 1, 2, 4 และ 8 โพรเซส.....	65
รูปที่ 4-9 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ส่วนของการประมวลผลขั้นปลายในลักษณะโปรแกรมแบบขนานบนคอมพิวเตอร์ที่มีหน่วยประมวลผล 2 แกน ด้วยภาพขนาด 320x240 พิกเซล จำนวน 1 และ 2 โพรเซส.....	66
รูปที่ 4-10 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ส่วนของการประมวลผลขั้นปลายในโปรแกรมแบบขนานบนคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วย ด้วยภาพขนาด 320x240 พิกเซล จำนวน 1, 2, 4 และ 8 โพรเซส.....	66
รูปที่ 4-11 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ในลักษณะโปรแกรมแบบขนานบนคอมพิวเตอร์ส่วนบุคคลที่มีหน่วยประมวลผลสองแกน ด้วยขนาดภาพ 320x240 พิกเซล จำนวน 2 โพรเซส.....	67
รูปที่ 4-12 เวลาเฉลี่ยรวมของการประมวลผลอัลกอริทึม P2P ลักษณะโปรแกรมแบบขนานบนคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วย ด้วยภาพขนาด 320x240 พิกเซล โดยใช้ 1, 2, 4 และ 8 โพรเซส.....	68
รูปที่ 4-13 เวลาเฉลี่ยทั้งหมดของการประมวลผลอัลกอริทึม P2P ในลักษณะโปรแกรมแบบตามลำดับ และโปรแกรมแบบขนานที่ใช้ 2 โพรเซสในการประมวลผล บนคอมพิวเตอร์ที่มีหน่วยประมวลผลสองแกน ด้วยภาพขนาด 320x240 พิกเซล.....	68

## รายการภาพประกอบ (ต่อ)

รูปที่ 4-14 ตัวอย่างการเพิ่มช่วงให้กับ U-disparity ภาพ (ก) ภาพซ้าย (ข) แผนผังคิสพาริตี (ค) การแบ่ง U-disparity ออกเป็น 10 ช่วง (ง) ผลลัพธ์ที่ได้จากการแบ่ง U-disparity ออกเป็น 10 ช่วง (จ) การแบ่ง U-disparity ออกเป็น 15 ช่วง (ฉ) ผลลัพธ์ที่ได้จากการแบ่ง U-disparity ออกเป็น 15 ช่วง (ช) การแบ่ง U-disparity ออกเป็น 20 ช่วง (ซ) ผลลัพธ์ที่ได้จากการแบ่ง U-disparity ออกเป็น 20 ช่วง.....	87
รูปที่ 4-15 เวลาเฉลี่ยทั้งหมดของการประมวลผลอัลกอริทึม P2P ในลักษณะ โปรแกรมแบบขนานที่ใช้ 2 โพรเซสในการประมวลผลบนคอมพิวเตอร์ที่มีหน่วยประมวลผลสองแกน ด้วยอัตราส่วนลักษณะต่างๆ .....	89
รูปที่ ข-1 ภาพตั้งต้นที่ใช้สำหรับหาค่าจำนวนระดับคิสพาริตีสูงสุดที่เหมาะสมต่อระบบตรวจจับสิ่งกีดขวาง.....	107
รูปที่ ค-1 แผนภาพคลาสที่สร้างขึ้นสำหรับส่วนการประมวลผลภาพสเตอริโอและการประมาณค่าระยะทาง.....	113
รูปที่ ค-2 แผนภาพลำดับการเรียกใช้งานส่วนต่างๆ ในระบบที่พัฒนาขึ้น .....	114
รูปที่ ค-3 แผนภาพกิจกรรมต่างๆ ที่เกิดขึ้นขณะการประมวลผล .....	115

## รายการตาราง

ตารางที่ 3-1 การใช้งานฟังก์ชันการรวบรวมข้อมูลกลับในแต่ละฟังก์ชันของการประมวลผลขั้น ปลาย.....	41
ตารางที่ 3-2 ตัวอย่างรูปภาพแผนผังคิสพาริตี ภาพ V-disparity ที่ใช้ในการเก็บข้อมูลเพื่อประมาณ ระยะทางของวัตถุที่คขวาง .....	42
ตารางที่ 3-3 ระดับคิสพาริตีที่ตรงกับระยะทางจริงของวัตถุ.....	44
ตารางที่ 3-4 ระยะทางที่ได้จากสมการที่ (12) เมื่อเทียบกับระยะทางเฉลี่ยที่ระดับคิสพาริตีต่างๆ....	47
ตารางที่ 3-5 ระดับคิสพาริตีที่ตรงกับระยะทางจริงของวัตถุ โดยเลือกข้อมูลระยะทางที่น้อยที่สุดจาก แต่ละระดับของคิสพาริตี.....	48
ตารางที่ 4-1 เวลาในการประมวลผลโปรแกรมในแต่ละกระบวนการของอัลกอริทึม P2P บน คอมพิวเตอร์ที่มีหน่วยประมวลผลแบบสองแกนที่จำนวนระดับคิสพาริตีสูงสุด 100 ภาพขนาด 320x240 พิกเซล .....	69
ตารางที่ 4-2 ค่าค่าสปีดอัปและค่าประสิทธิภาพการประมวลผลในแต่ละกระบวนการของอัลกอริทึม P2P บนคอมพิวเตอร์ที่มีหน่วยประมวลผลแบบสองแกนโดยใช้ข้อมูลจากตารางที่ 4-1 .....	69
ตารางที่ 4-3 เวลาในการประมวลผลโปรแกรมในแต่ละกระบวนการของอัลกอริทึม P2P บน คอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผลแบบ 8 หน่วย ที่จำนวนระดับคิสพาริตีสูงสุด 100 ภาพขนาด 320x240 พิกเซล.....	70
ตารางที่ 4-4 ค่าค่าสปีดอัปและค่าประสิทธิภาพการประมวลผลในแต่ละกระบวนการของอัลกอริทึม P2P บนคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผลแบบ 8 หน่วย โดยใช้ข้อมูลจากตารางที่ 4-3 .....	70
ตารางที่ 4-5 รูปในแต่ละสถานที่ตัวอย่างที่ใช้ทดสอบหาตำแหน่งของวัตถุในสภาพแสงกลางวัน ภายนอกอาคาร .....	72
ตารางที่ 4-6 รูปในแต่ละสถานที่ตัวอย่างที่ใช้ทดสอบหาตำแหน่งของวัตถุในสภาพแสงกลางวัน ภายในอาคาร .....	76
ตารางที่ 4-7 สภาพแวดล้อมที่มีปัญหาในการตรวจจับด้วยอัลกอริทึม P2P.....	79
ตารางที่ 4-8 อัตราส่วนลักษณะ 4:3 ภาพขนาด 320x240 พิกเซล.....	81
ตารางที่ 4-9 อัตราส่วนลักษณะ 16:9 ภาพขนาด 320x180 พิกเซล.....	82
ตารางที่ 4-10 อัตราส่วนลักษณะ 8:3 ภาพขนาด 320x120 พิกเซล.....	83

## รายการตาราง (ต่อ)

ตารางที่ 4-11 อัตราส่วนลักษณะ 16:5 ภาพขนาด 320x100 พิกเซล.....	84
ตารางที่ 4-12 อัตราส่วนลักษณะ 4:1 ภาพขนาด 320x80 พิกเซล.....	85
ตารางที่ ข-1 ตัวอย่างภาพคิสพาริตีเมื่อกำหนดจำนวนระดับคิสพาริตีสูงสุดต่างๆ ที่ระยะทางต่างๆ .....	108

## สัญลักษณ์คำย่อและตัวย่อ

CCD	Charge Coupled Device
CMOD	Complementary Metal Oxide Semiconductor
CPU	Central Processing Unit
ENVS	Electro Neural Vision System
GPS	Global Positioning System
MPI	Message Passing Interface
P2P	Depth discontinuities by pixel-to-pixel stereo
RAM	Random Access Memory
SMP	Symmetric Multiprocessor



## ศัพท์บัญญัติ

post-processing	การประมวลผลขั้นปลาย
dynamic programming	กำหนดการพลวัต
interpolation	การประมาณค่าในช่วง
stereo vision	สเตอริโอวิชัน
scanline	เส้นกราดตรวจ
disparity map	แผนผังคิสภาพริที

# บทที่ 1

## บทนำ

### 1.1. ความสำคัญและที่มาของวิทยานิพนธ์

การช่วยให้ผู้พิการสามารถดำรงชีวิตประจำวันได้อย่างปกติ และลดการพึ่งพาคนรอบข้างได้นั้นจำเป็นต้องอาศัยเครื่องอำนวยความสะดวกต่างๆ เพื่อลดทอนความไม่สะดวกจากความพิการนั้นลงและสามารถพึ่งพาตัวเองได้มากขึ้น ผู้พิการทางสายตาเป็นกลุ่มผู้พิการที่ต้องการความช่วยเหลือเป็นพิเศษในการอำนวยความสะดวกในการเดินทางเนื่องจากตาที่มองไม่เห็นทำให้การเดินทางของผู้พิการเองเป็นไปด้วยความลำบาก ประกอบกับไม้เท้าขาวซึ่งเป็นเครื่องมือตรวจสอบสิ่งกีดขวางที่ได้รับความนิยมในปัจจุบัน สามารถตรวจหาวัตถุกีดขวางและสภาพแวดล้อมรอบตัวผู้พิการได้ในระดับหนึ่ง แต่ไม่สามารถตรวจจับวัตถุที่แขวนไว้หรือวัตถุที่ยื่นออกจากกำแพงที่สูงกว่าหน้าอกของผู้พิการขึ้นไปได้ อาทิเช่น กั้นสาด ป้ายโฆษณา เป็นต้น การให้ผู้พิการทางสายตาแหว่งไม้เท้าเพื่อตรวจสอบสภาพแวดล้อมต่างๆ อาจส่งผลให้ผู้ร่วมทางอื่นๆ ได้รับอันตรายจากการกระทำได้กล่าว ผู้ดูแลผู้พิการจึงไม่วางใจให้ผู้พิการทางสายตาดำเนินการโดยลำพังไปยังสถานที่ต่างๆ ที่ตัวผู้พิการเองไม่เคยไปมาก่อน เนื่องจากผู้พิการทางสายตาอาจจะประสบอุบัติเหตุระหว่างการเดินทางได้

เทคโนโลยีสำหรับผู้พิการทางสายตาได้รับการพัฒนาอย่างต่อเนื่อง โดยที่อุปกรณ์อำนวยความสะดวกสำหรับการเดินทางของผู้พิการทางสายตาที่แพร่หลายในปัจจุบันประดิษฐ์ขึ้นจากเทคโนโลยีสองประเภทคือ อัลตราโซนิก และ ระบบกำหนดตำแหน่งบนโลก (Global Positioning System: GPS) สำหรับระบบนำทางผู้พิการทางสายตาโดยใช้ GPS นั้นสามารถนำทางผู้พิการที่ต้องการเดินทางในระยะไกลให้ไปถึงที่หมายได้ แต่ไม่สามารถตรวจสอบสภาพแวดล้อมรอบตัวผู้พิการได้ ในขณะที่ไม้เท้าที่ติดตั้งอัลตราโซนิกสามารถตรวจสอบสภาพแวดล้อมรอบตัวผู้พิการทางสายตาได้ โดยอาศัยคุณสมบัติการสะท้อนของคลื่นอัลตราโซนิกเป็นตัวตรวจสอบสิ่งกีดขวาง ส่งผลให้ผู้พิการทางสายตาสามารถรับทราบตำแหน่งของสิ่งกีดขวางดังกล่าวและหลบหลีกได้ดีขึ้น อย่างไรก็ตามถึงแม้ว่าอัลตราโซนิกจะสามารถตรวจจับสิ่งกีดขวางและหาระยะทางได้แม่นยำ แต่ไม้เท้าอัลตราโซนิกที่ใช้กันอยู่ในปัจจุบันไม่สามารถบอกขนาดของวัตถุได้ อีกทั้งยังไม่สามารถบอกสถานที่ได้ชัดเจนว่าในขณะที่ผู้พิการทางสายตามาถึงยังบริเวณที่ต้องการแล้วหรือไม่ รวมไปถึงจนถึงไม้เท้าอัลตราโซนิกไม่สามารถตรวจสอบได้ว่าบริเวณที่มีสิ่งกีดขวางนั้นสามารถเดินเข้าไปได้หรือไม่

เมื่ออัลตราโซนิกไม่สามารถตอบสนองความต้องการที่มากขึ้น ผู้วิจัยจึงสนใจนำระบบการมองเห็นที่ใช้ในหุ่นยนต์มาประยุกต์ใช้ในการเดินทางของผู้พิการทางสายตาในที่นี้คือเทคนิคการประมวลผลภาพ การประมวลผลภาพสามารถวิเคราะห์ลักษณะบริเวณที่ผู้พิการจะสามารถเดินทางเข้าไปได้ว่ามีอุปสรรคกีดขวางหรือไม่ แต่การใช้ภาพเพียงภาพเดียวในการประมวลผลไม่สามารถบอกระยะทางจากผู้พิการถึงสิ่งกีดขวางได้ จึงต้องใช้เทคนิคการประมวลผลภาพสเตอริโอเข้ามาช่วย เพื่อที่จะหาระยะของวัตถุได้ถูกต้อง การประมวลผลภาพสเตอริโอ คือการประมวลผลภาพมากกว่าหนึ่งภาพ ข้อมูลที่ได้จากการประมวลผลภาพสเตอริโอคือระยะลึก ซึ่งสามารถนำระยะลึกนี้ไปใช้หาระยะทางจากผู้พิการถึงสิ่งกีดขวางได้ อย่างไรก็ตามระยะเวลาในการตอบสนองของการประมวลผลภาพแบบสเตอริโอนั้นใช้เวลานาน หากนำเทคนิคนี้มาใช้งานกับผู้พิการทางสายตา ความเร็วในการเดินทางของผู้พิการทางสายตาจะไม่สัมพันธ์กับผลตอบสนองของการประมวลผลภาพสเตอริโอซึ่งช้ากว่ามากอาจก่อให้เกิดอันตรายแก่ตัวผู้พิการได้

เทคโนโลยีหน่วยประมวลผลกลางของคอมพิวเตอร์ในปัจจุบันได้พัฒนาไปอย่างต่อเนื่อง มีการพัฒนาหน่วยประมวลผลกลางของคอมพิวเตอร์ให้สามารถบรรจุหน่วยประมวลผลได้หลายหน่วยประมวลผลโดยเรียกเทคโนโลยีประเภทนี้ว่าเทคโนโลยีมัลติคอร์ ยกตัวอย่างเช่น Intel Core 2 Duo, Intel CORE i7, AMD Phenom และ AMD Athlon64 X2 เป็นต้น หน่วยประมวลผลแบบมัลติคอร์นี้ไม่สามารถทำงานได้อย่างเต็มประสิทธิภาพหากยังใช้การพัฒนาโปรแกรมแบบตามลำดับโดยทั่วไปที่ใช้กันอยู่ในปัจจุบัน จึงจำเป็นต้องใช้เทคนิคการโปรแกรมแบบขนานเพื่อกระจายงานให้แต่ละหน่วยประมวลผลสามารถทำงานได้อย่างเต็มที่ ซึ่งการนำเทคนิคการโปรแกรมแบบขนานมาประยุกต์ใช้กับการประมวลผลภาพสเตอริโอย่อมส่งผลให้เพิ่มผลตอบสนองต่อหนึ่งหน่วยเวลาให้มากขึ้นได้

วิทยานิพนธ์ฉบับนี้มุ่งนำเสนอระบบตรวจจับสิ่งกีดขวางที่เหมาะสมกับการเดินทางของผู้พิการทางสายตาโดยใช้การประมวลผลภาพสเตอริโอ เน้นส่วนตรวจจับวัตถุสิ่งกีดขวางด้วยกล้องสเตอริโอและการหาตำแหน่งและประมาณระยะทางของสิ่งกีดขวาง ปรับปรุงประสิทธิภาพการประมวลผลภาพสเตอริโอโดยใช้เทคนิคการโปรแกรมแบบขนาน เพื่อเพิ่มปริมาณของผลตอบสนองในหนึ่งหน่วยเวลาให้สอดคล้องใกล้เคียงกับความเร็วในการเดินทางของผู้พิการทางสายตา เป็นต้นแบบเพื่อนำไปใช้งานจริงได้ในอนาคต

## 1.2. การตรวจเอกสาร

### 1.2.1. เทคโนโลยีที่ใช้ช่วยเหลือในการเดินทางของผู้พิการทางสายตา

เทคโนโลยีที่ใช้ช่วยเหลือในการเดินทางของผู้พิการทางสายตาสามารถจำแนกออกเป็นสองลักษณะใหญ่ๆ คืออุปกรณ์ตรวจสอบอุปสรรคสิ่งกีดขวางรอบตัวผู้พิการและระบบนำทางไปยังจุดหมายต่างๆ ที่ผู้พิการต้องการเดินทาง ซึ่งระบบนำทางประเภทนี้จะต้องอาศัยอุปกรณ์ตรวจสอบสิ่งกีดขวางด้วย ผู้พิการทางสายตาจึงจะสามารถเดินทางไปยังจุดหมายปลายทางได้อย่างปลอดภัย เนื่องจากระบบนำทางกลุ่มนี้จะใช้ข้อมูลจาก GPS [1], [2], [3] เป็นหลัก ซึ่งการใช้เทคโนโลยีนี้เพียงอย่างเดียวจะสามารถนำทางผู้พิการให้เดินทางไปยังจุดหมายได้ตามที่ต้องการ แต่ตัวผู้พิการเองห้ามออกจากเส้นทางที่ระบุไว้ เนื่องจากอาจเกิดอันตรายได้ ซึ่งในความเป็นจริงแล้วผู้พิการทางสายตาไม่อาจรู้ได้เลยว่าข้างหน้ามีอุปสรรคสิ่งกีดขวางหรือไม่ นอกจากการใช้ GPS ดังกล่าวเพียงอย่างเดียวแล้วยังมีงานวิจัยอื่นที่เพิ่มทางเลือกให้แก่ผู้พิการโดยใช้ GPS ร่วมกับสถานีฐานของโทรศัพท์เคลื่อนที่ [4] นอกจากนี้ยังมีงานวิจัยที่สร้างทางเดินให้กับผู้พิการทางสายตาโดยใช้งานร่วมกับไม้เท้าขาว แต่ผู้พิการจะต้องเดินทางภายในบริเวณที่กำหนดไว้เท่านั้น ซึ่งแบ่งออกเป็นลักษณะต่างๆ ดังต่อไปนี้คือการใช้อินฟราเรด [5] หรือ IC tags [6] ตรวจจับเส้นทางที่กำหนดไว้ หรือการใช้เทคนิคการประมวลผลภาพประมวลผลสัญลักษณ์ที่นำไปคิดไว้ยังที่ต่างๆ ได้แก่ Tag-camera-based Identification [7] ถึงแม้ระบบนำทางที่ได้กล่าวไปแล้วนี้จะมีมีความสำคัญ แต่การที่ผู้พิการทางสายตาสามารถเดินทางไปยังสถานที่ต่างๆ ได้อย่างปลอดภัยและเป็นอิสระนั้นมีความจำเป็นมากกว่า ดังนั้นวิทยานิพนธ์นี้จะเน้นไปที่อุปกรณ์ตรวจสอบสิ่งกีดขวาง

ผู้วิจัยพบว่าการประดิษฐ์อุปกรณ์ตรวจสอบสิ่งกีดขวางสำหรับผู้พิการทางสายตามีหลายรูปแบบ แต่รูปแบบที่มีผู้ใช้งานมากที่สุดคือการใช้คลื่นอัลตราโซนิก ซึ่งมีการประยุกต์ใช้ในหลายๆ อุปกรณ์ เช่น ไม้เท้าขาว [8] นอกจากคลื่นอัลตราโซนิกแล้ว ยังมีวิธีการประมวลผลภาพ [9] เพื่อช่วยในการเดินทางของผู้พิการทางสายตา หากแต่การประมวลผลภาพนี้เป็นการตรวจสอบข้อมูลเฉพาะเส้นทางบนพื้นผิวจราจร เช่น ทางม้าลาย เส้นขอบผิวจราจร หรือสัญญาณไฟจราจร จะไม่สามารถบอกได้ว่าวัตถุมีระยะห่างเท่าไรจากตัวผู้พิการ

สำหรับระบบตรวจสอบสิ่งกีดขวางที่ใช้การประมวลผลภาพสเตอริโอสำหรับผู้พิการทางสายตานั้นเสนอโดย S. Meer และ K. Ward โดยนำเสนอระบบที่ชื่อว่า Electro Neural Vision System (ENVS) [10] ENVS นำผลลัพธ์ที่ได้จากการประมวลผลภาพสเตอริโอคือแผนผังคิสพาริตีที่นำเสนอแก่ผู้พิการทางสายตาผ่านทาง Electro-tactile display ตู้นิ้วมือผู้พิการทั้งสองนิ้ว [10] ซึ่งข้อมูลที่นำเสนอแก่ผู้พิการนี้เป็นการคำนวณปริมาณคิสพาริตีที่พบในแต่ละช่วง ระบบนี้ถึงแม้จะมี

ข้อดีหลายประการแต่ยังมีจุดด้อยอยู่ที่ผู้พิการทางสายตาใช้ไม้เท้าขาวจนเคยชิน การเปลี่ยนพฤติกรรมการใช้งานจากไม้เท้ามาใช้ระบบนี้โดยทันที อาจทำให้ผู้พิการไม่มั่นใจหรือหากใช้งานร่วมกับไม้เท้าขาวอาจสร้างความไม่สะดวก อีกทั้งระบบนี้ใช้การประมวลผลภาพสเตอริโอในกลุ่ม feature-based stereo matching จึงไม่สามารถตรวจจับวัตถุที่ไม่มีลักษณะเด่น เช่น ตู้ กระดานดำ กำแพง ฯลฯ ได้เองจึงอาจต้องเพิ่มอุปกรณ์ช่วยตรวจจับวัตถุเพิ่มเติม รวมไปถึงจนถึงการใช้ปริมาณคิสพาริตีที่เป็นเครื่องกำหนดระยะทางวัตถุอาจจะส่งผลให้ผู้พิการไม่สามารถทราบระยะทางที่แท้จริงของวัตถุเพื่อการตัดสินใจเลือกเส้นทางที่ดีในการเดินทางได้

### 1.2.2. สเตอริโอวิชัน

สเตอริโอวิชัน (Stereo Vision) เป็นการนำภาพสองภาพ (stereo) ที่ถ่ายจากกล้องสองกล้อง มาประมวลผลเพื่อนำข้อมูลที่ได้ไปใช้งานต่อ โดยที่กระบวนการประมวลผลภาพสเตอริโอที่ให้ผลลัพธ์เป็นแผนผังคิสพาริตี (disparity map) มีอยู่ด้วยกันหลายวิธี ซึ่งสามารถจัดกลุ่มได้สองลักษณะใหญ่ๆ คือ เทคนิค intensity-based stereo matching (ISM) เน้นการหาความเข้มที่เหมือนกันในแต่ละคู่แถวเดียวกันของภาพสองภาพ และ เทคนิค feature-based stereo matching (FSM) เป็นการเน้นหาลักษณะเด่นของภาพในแต่ละภาพเพื่อหาลักษณะเด่นที่เหมือนกันในภาพนั้นๆ มาจับคู่กัน ซึ่งทั้งสองเทคนิคนี้มีข้อดีข้อเสียที่แตกต่างกัน แต่งานวิจัยนี้เน้นไปที่ ISM เนื่องจาก ISM เป็นเทคนิคที่มุ่งไปที่ความเข้มของภาพโดยไม่สนใจลักษณะเด่นที่อยู่ในภาพ ส่งผลให้ ISM สามารถตรวจสอบหาอุปสรรคประเภทไร้ลักษณะเด่นเช่น ตู้ ประตู กระดานดำ ฯลฯ ได้ โดยอัลกอริทึมที่วิทยานิพนธ์นี้สนใจคือ Depth Discontinuities by Pixels-to-Pixels Stereo (P2P) [13], [14] เป็นอัลกอริทึมที่สามารถนำไปแกรมต้นฉบับมาปรับปรุงต่อได้ อีกทั้งอัลกอริทึมนี้ไม่ต้องเตรียมภาพก่อนการใช้งานเพียงแต่ปรับภาพให้เป็นระดับสีเทาเท่านั้น

ผลลัพธ์จากการประมวลผลภาพสเตอริโอ P2P คือแผนผังคิสพาริตี ข้อมูลที่บรรจุอยู่ในแผนผังคิสพาริตีคือระดับของคิสพาริตีที่เป็นตัวแทนของระยะลึกต่างๆ ซึ่งสามารถนำข้อมูลส่วนนี้ไปใช้งานต่อให้เกิดประโยชน์ได้ เช่น การตรวจจับวัตถุ การประมาณค่าระยะทางของวัตถุที่ตรวจพบ เป็นต้น สำหรับการตรวจจับวัตถุและการประมาณระยะทางของวัตถุ สามารถทำได้ด้วยกันได้ 2 วิธีหลักคือ วิธี Euclidean [27], [28] ซึ่งเป็นการใช้งานโครงสร้างของข้อมูลพิกัด 3 มิติ และ U-V-Disparity [25], [26], [27] เป็นกระบวนการแปลงข้อมูลแผนผังให้อยู่ในรูปของระบบพิกัด u และ v แล้วนำข้อมูลที่ไปใช้งานต่อไป ซึ่ง U-V-disparity [30] เหมาะแก่การประมาณค่าระยะทางของวัตถุหรือสิ่งกีดขวางในวิทยานิพนธ์นี้ เนื่องจากสามารถนำเอาผลลัพธ์จากอัลกอริทึม P2P มาใช้งานต่อได้ จากการศึกษาการตรวจจับวัตถุบนท้องถนนสำหรับระบบความปลอดภัยเพื่อการขับขี่

(Driving Safety Assistance) [29] พบว่าการใช้กระบวนการทางสเตอริโอวิชันและ V-disparity สามารถทำงานได้ดีแม้สภาพอากาศและสภาพแวดล้อมเปลี่ยนแปลงไป เช่น สภาพอากาศที่มีหมอกเล็กน้อย หรือ ฝนตกเบาบาง ทั้งนี้ขึ้นอยู่กับคุณภาพของแผนผังคิสพาริตีด้วย นอกจากนี้แผนผังคิสพาริตีในระบบพิกัด  $u$  และ  $v$  หรือ U-V-disparity [30] ยังสามารถบอกขนาดความกว้าง และ ระยะทางของวัตถุได้อีกด้วย

### 1.2.3. การประมวลผลแบบขนาน

เทคโนโลยีหน่วยประมวลผลกลาง (Central Processing Unit: CPU) พัฒนาไปอย่างรวดเร็ว โดยในปัจจุบันผู้ผลิตเตียงข้อจำกัดด้านความเร็วของสัญญาณนาฬิกา โดยเน้นการเพิ่มแกนของ CPU แทนซึ่งรู้จักกันในนามของหน่วยประมวลผลแบบหลายแกน หรือหน่วยประมวลผลแบบมัลติคอร์ (multi-core CPU) อีกทั้งได้รับความนิยมและสามารถเลือกซื้อได้ทั่วไปตามท้องตลาดทั้ง คอมพิวเตอร์ตั้งโต๊ะส่วนบุคคล (Desktop PC) และคอมพิวเตอร์แบบพกพา (Computer Notebook) แต่วิธีการพัฒนาโปรแกรมในปัจจุบันนั้นไม่สามารถใช้ประโยชน์จาก CPU ชนิดนี้ได้อย่างเต็มประสิทธิภาพ เนื่องจากการพัฒนาโปรแกรมแบบตามลำดับจะใช้งาน CPU เพียงแกนเดียวในการประมวลผล หากต้องการเพิ่มความเร็วให้กับงานที่สามารถแบ่งออกเป็นส่วนๆ ซึ่งแยกการคำนวณออกจากกันได้จำเป็นต้องอาศัยเทคนิคการประมวลผลแบบขนานเพื่อช่วยกระจายการทำงานออกไปยังแต่ละแกนของ CPU

การประมวลผลแบบขนานนั้นสามารถทำได้หลายวิธี แต่วิธีที่นิยมมากที่สุดคือการเขียนโปรแกรมโดยระบุการใช้งานไลบรารีการประมวลผลแบบขนานอย่างชัดเจน โดยแบ่งออกเป็น การพัฒนาโปรแกรมแบบส่งผ่านข้อความ (Message Passing Interface: MPI) [18] เน้นการใช้งานและการสื่อสารระหว่างโพรเซส และการพัฒนาโปรแกรมแบบใช้ตัวแปรร่วม (Share Variable Programming) ซึ่งใช้เซรคเป็นหลัก การโปรแกรมทั้งสองแบบนี้สามารถใช้งานบนสถาปัตยกรรมคอมพิวเตอร์แบบซิมเมตริกมัลติโพรเซสเซอร์ (Symmetric Multiprocessor: SMP) ได้ โดยที่คอมพิวเตอร์ในปัจจุบันจัดอยู่ในประเภทนี้ด้วยเช่นกัน โดยวิทยานิพนธ์นี้สนใจการใช้งาน MPI เนื่องจาก MPI สามารถรองรับการทำงานและการสื่อสารระหว่างโพรเซสได้ดี เหมาะแก่การย้ายโปรแกรมไปประมวลผลบนสถาปัตยกรรมหน่วยประมวลผลกลางในหลายรูปแบบ วิทยานิพนธ์นี้เลือกใช้งานไลบรารีของ MPICH [19] ซึ่งเป็นการพัฒนาไลบรารีสำหรับการประมวลผลแบบขนานตามมาตรฐานของ MPI ไลบรารีของ MPICH สามารถใช้งานได้หลายระบบปฏิบัติการ เพียงแต่เลือกชนิดอุปกรณ์สื่อสาร (communication device) ให้ตรงกับความต้องการเท่านั้น

### 1.3. วัตถุประสงค์

- 1) หากความเหมาะสมของระบบสเตอริโอวิชันที่ใช้งานสำหรับผู้พิการทางสายตา สามารถตรวจจับวัตถุ บอกระยะทางจากกล้องจนถึงวัตถุได้ บอกตำแหน่งของวัตถุ ในระยะทาง 1- 6 เมตร
- 2) หาวิธีการตรวจสอบขนาดวัตถุที่สามารถตรวจจับได้ในระยะทางต่างๆ รวมถึงความเที่ยงตรงและแม่นยำของระบบสเตอริโอวิชันที่จัดทำขึ้น
- 3) เพิ่มประสิทธิภาพของระบบสเตอริโอวิชันให้สามารถทำงานได้เร็วขึ้นด้วยการประมวลผลแบบขนาน

### 1.4. ขอบเขตของการวิจัย

- 1) กล้องที่ใช้ในการทำวิจัยเป็นเว็บแคม 2 ตัว มีความละเอียด 640x480 พิกเซล
- 2) สภาพแสงที่ใช้ทดสอบเป็นสภาพแสงกลางวัน สภาพอากาศปลอดโปร่ง หรือสภาพแสงที่ได้รับการควบคุม
- 3) ทดสอบและเปรียบเทียบการประมวลผลบนหน่วยประมวลผลเดี่ยวและ 2 ถึง 8 หน่วยประมวลผลเท่านั้น

### 1.5. ประโยชน์ที่คาดว่าจะได้รับ

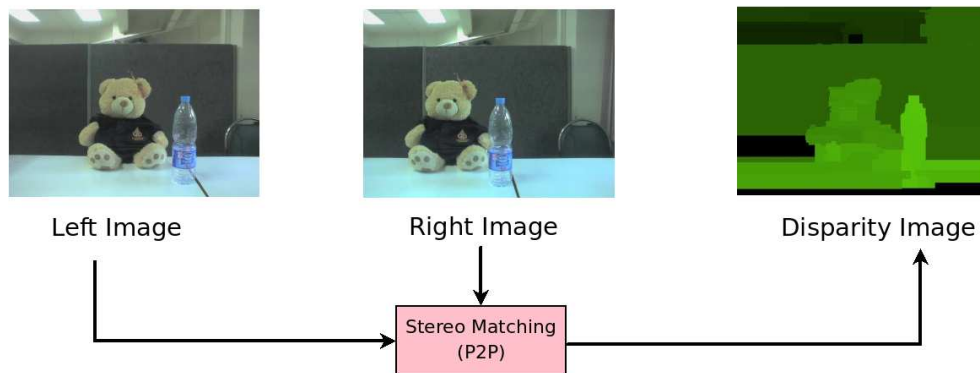
- 1) ได้ระบบตรวจสอบสิ่งกีดขวางด้วยสเตอริโอวิชันที่เหมาะสมต่อผู้พิการทางสายตา สามารถนำไปประยุกต์ใช้งานกับผู้พิการทางสายตาได้ในอนาคต
- 2) มีระบบตรวจสอบสิ่งกีดขวางด้วยสเตอริโอวิชันที่ประมวลผลได้เร็วเพียงพอต่อการใช้งานสำหรับผู้พิการทางสายตา

## บทที่ 2

### ทฤษฎีและหลักการ

#### 2.1. การประมวลผลภาพแบบสเตอริโอ

คอมพิวเตอร์สเตอริโอวิชัน (computer stereo vision) เป็นส่วนหนึ่งของคอมพิวเตอร์วิชัน ซึ่งมักนำมาใช้กับระบบการมองเห็นของหุ่นยนต์เพื่อใช้ตรวจจับอุปสรรคสิ่งกีดขวางต่างๆ โดยเป็นเทคนิคที่ประมวลผลภาพเลียนแบบระบบการมองเห็นของมนุษย์ ระบบการมองเห็นของมนุษย์นั้นเริ่มจากแสงสะท้อนจากวัตถุเข้าตามนุษย์ จากนั้นตามนุษย์ส่งข้อมูลภาพที่ได้ไปยังสมองส่วนที่มีชื่อว่า ซีรีบรัม เพื่อประมวลผลข้อมูลภาพที่ได้รับ เช่นเดียวกับคอมพิวเตอร์สเตอริโอวิชันที่เกิดจากการนำภาพสองภาพจากกล้องถ่ายภาพสองกล้องที่ถ่ายภาพทัศนียภาพเดียวกันโดยมีระยะห่างระหว่างกล้องระยะหนึ่ง (ลักษณะเดียวกันกับตามนุษย์) มาประมวลผลภาพ โดยที่เปรียบเทียบพิกเซลในแต่ละภาพเพื่อหาตำแหน่งของพิกเซลจากภาพด้านซ้ายที่เป็นตำแหน่งเดียวกันของภาพด้านขวา ระยะระหว่างสองพิกเซลนี้เรียกว่าคิสพาริตี (disparity) โดยเรียกกระบวนการนี้ว่าการจับคู่ภาพสเตอริโอ (stereo matching) ซึ่งคิสพาริตีที่บริเวณวัตถุหรือสิ่งกีดขวางที่มีคุณภาพดี สามารถนำมาคำนวณหาระยะลึกของวัตถุได้ ลักษณะของแผนผังคิสพาริตีเป็นดังรูปที่ 2-1



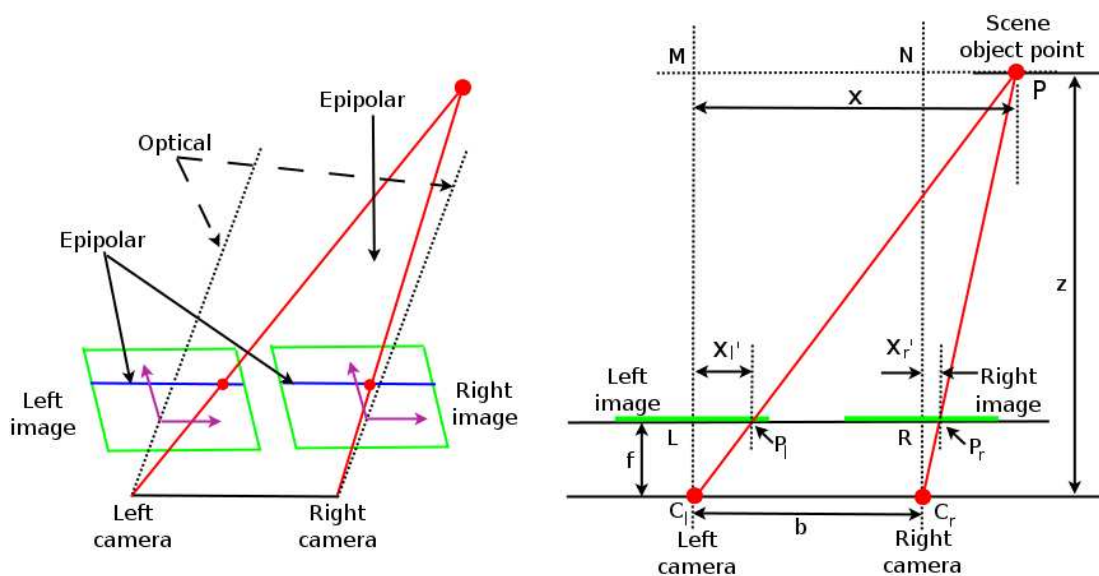
รูปที่ 2-1 กระบวนการจับคู่ภาพสเตอริโอซึ่งผลลัพธ์ที่ได้คือแผนผังคิสพาริตี

##### 2.1.1. ลักษณะทางเรขาคณิตของ Binocular stereo

จากรูปที่ 2-2 ซึ่งเป็นลักษณะทางเรขาคณิตของ binocular stereo [11] จะเห็นได้ว่าแบบจำลองพื้นฐานของกล้องทั้งสองจะอยู่ในระนาบเดียวกัน (แกน x) โดยมีระยะห่างเท่ากับระยะ b หรือ base line และระนาบของภาพ (image plane) ของกล้องทั้งสองตัวนั้นอยู่ในระนาบเดียวกัน



ดังนั้นจุดที่เราสนใจของภาพจากภายนอกจะปรากฏบนระนาบของกล้องทั้งสองที่ตำแหน่งต่างกัน ซึ่งระยะห่างระหว่างตำแหน่งที่ปรากฏบนภาพนั้นเรียกว่าดิสพาริตี และระนาบที่ผ่านจุดศูนย์กลางของกล้องทั้งสองและจุดที่เราสนใจนั้นเรียกว่า epipolar plane เส้นที่เกิดขึ้นจากการตัดกันของระนาบภาพกับ epipolar plane นั้นถูกกำหนดให้เป็น epipolar line ซึ่งจากแบบจำลองที่แสดงไว้ในรูปที่ 2-2 นั้น ลักษณะจุดของภาพจากระนาบหนึ่งจะอยู่ในแถวเดียวกันกับอีกระนาบหนึ่งหมายความว่าเส้น epipolar line จะอยู่ที่ตำแหน่งแถวเดียวกันของระนาบภาพของกล้องทั้งสอง แต่ในการกำหนดให้เป็นแบบนี้ได้นั้นต้องกำหนดให้ vertical disparity มีค่าเท่ากับศูนย์



รูปที่ 2-2 ลักษณะทางเรขาคณิตของ Binocular stereo [11]

จากรูปที่ 2-2 จุด P จะปรากฏเป็นจุด  $P_L$  และ  $P_R$  บนระนาบภาพด้านซ้าย และระนาบภาพด้านขวาตามลำดับ  $f$  คือระยะโฟกัสของกล้อง  $C_L$  และ  $C_R$  คือตำแหน่งศูนย์กลางเลนส์ของกล้องด้านซ้ายและกล้องด้านขวา  $x$  คือระยะทางจากจุดกึ่งกลางกล้องซ้ายถึงจุด P ส่วน จากนั้นกำหนดให้จุดกำเนิดของพิกัดอยู่บนตำแหน่งเดียวกันกับตำแหน่งศูนย์กลางเลนส์ของกล้องด้านซ้าย ( $C_L$ ) หากเปรียบเทียบสามเหลี่ยมคล้าย  $PMC_L$  และ  $P_L C_L$  จะได้สมการที่ (1)

$$\frac{x}{z} = \frac{x_L'}{f} \quad (1)$$

ในทำนองเดียวกัน จากสามเหลี่ยมคล้าย  $PNC_R$  และ  $P_R C_R$  จะได้สมการที่ (2)

$$\frac{x - b}{z} = \frac{x_r'}{f} \quad (2)$$

รวมสมการที่ (1) และสมการที่ (2) เข้าด้วยกันจะได้สมการที่ (3) ดังนี้

$$z = \frac{bf}{(x_l' - x_r')} \quad (3)$$

ดังนั้นหาระยะ  $z$  หรือระยะห่างระหว่างวัตถุกับจุดศูนย์กลางของภาพได้จากค่าคิสมารีทีของภาพนั่นเอง ในการมองเห็นของมนุษย์นั้นมีหลักการมองเห็นคล้ายกับหลักการของสเตอริโอวิชันที่ได้กล่าวมาแล้วข้างต้น เพียงแต่จากหลักการที่กล่าวมา แขน optical ของกล้องทั้งสองนั้นขนานกัน ซึ่งการมองของคนเราจะมีลักษณะการมองเหมือนกับนำกล้องสองตัวมาติดตั้งให้แกน optical ตัด

### 2.1.2. การจับคู่ภาพสเตอริโอ

กระบวนการจับคู่ภาพสเตอริโอ [12] (stereo matching) นี้สามารถแบ่งออกได้สองกลุ่มใหญ่ๆ ด้วยกัน คือ เทคนิค intensity-based stereo matching (ISM) และ เทคนิค feature-based stereo matching (FSM) แต่ละเทคนิคมีรายละเอียดดังต่อไปนี้

#### 2.1.2.1. Intensity-based stereo matching technique

เป็นเทคนิคที่พิจารณาความเข้มของภาพ โดยตั้งสมมุติฐานไว้ว่า หากเรานำภาพจากกล้องสองกล้องที่ขนานกันมาประมวลผล ภาพที่ได้จะมีเส้นกราดตรวจ (scanline) ที่มีพิกเซลเดียวกันบรรจุอยู่ แต่ระยะห่างต่างกัน ดังนั้นเราสามารถหาพิกเซลได้จากการหาความเข้มของแต่ละพิกเซลที่อยู่บนเส้นกราดตรวจเดียวกัน

ในกรณีที่กล้องที่ใช้จับภาพขนานกันจริง เส้น epipolar จะเหมือนกับเส้นกราดตรวจแนวนอน จุดที่สัมพันธ์กันในทั้งสองภาพจะอยู่บนตำแหน่งเส้นกราดตรวจแนวนอนเดียวกัน การติดตั้งกล้องสเตอริโอที่ขนานกันจะช่วยลดการค้นหาค่าความสัมพันธ์จากสองมิติ (รูปทั้งหมด) เป็นมิติเดียว ในความเป็นจริงจากแถวความสัมพันธ์ของรูปภาพทั้งคู่แสดงให้เห็นว่า intensity profile ของทั้งสองภาพแตกต่างกัน โดยมี horizontal shift และ local foreshortening รูปที่ 2-3 (ก) และ (ข) ทำให้เห็นว่าภาพที่ถ่ายจากกล้องที่มีตำแหน่งต่างกัน ในแนวราบ ดังนั้นภาพคู่นี้จะสอดคล้องกันกับการติดตั้งกล้องที่ขนานกัน เส้นกราดตรวจสองเส้นอยู่ที่ตำแหน่ง 80 และ 230 ทั้งสองรูป รูปที่ 2-3 (ค)

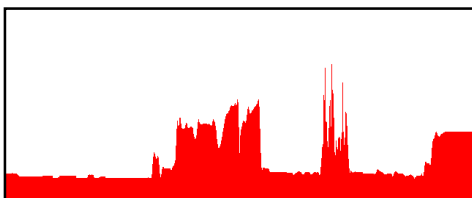
และ (ง), (จ) และ (ฉ) แสดง intensity profile ของแถวที่ 250 และแถวที่ 380 ของทั้งสองภาพตามลำดับ



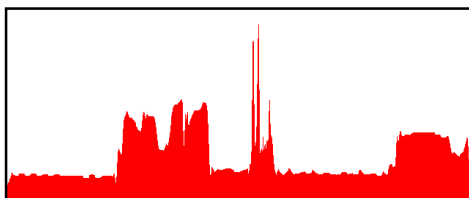
(ก) ภาพด้านซ้าย



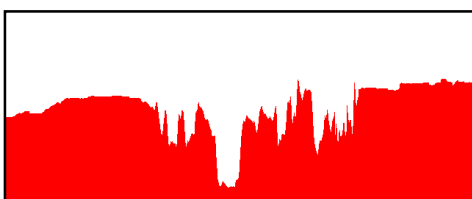
(ข) ภาพด้านขวา



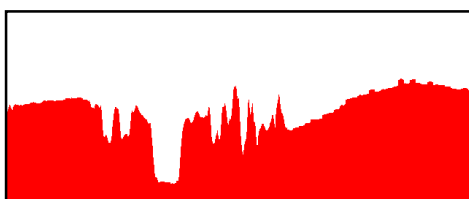
(ค) Intensity profile ของแถวที่ 250 ภาพซ้าย



(ง) Intensity profile ของแถวที่ 250 ภาพขวา



(จ) Intensity profile ของแถวที่ 380 ภาพซ้าย



(ฉ) Intensity profile ของแถวที่ 380 ภาพขวา

รูปที่ 2-3 ความเข้มของภาพซ้ายและภาพขวา ณ เส้นกราดตรวจเดียวกัน

ข้อดีของวิธีการนี้คือได้ dense disparity map ซึ่งให้ dense depth map ออกมา วิธีการที่ใช้กันใน intensity-based stereo matching ที่รู้จักกันทั่วไปคือ window-based method เป็นการจับคู่บริเวณในรูปภาพที่สนใจ ยกตัวอย่างเช่น บริเวณที่มีการเปลี่ยนแปลงค่าความเข้มสูงในแนวราบ แนวตั้งและแนวทแยงมุม

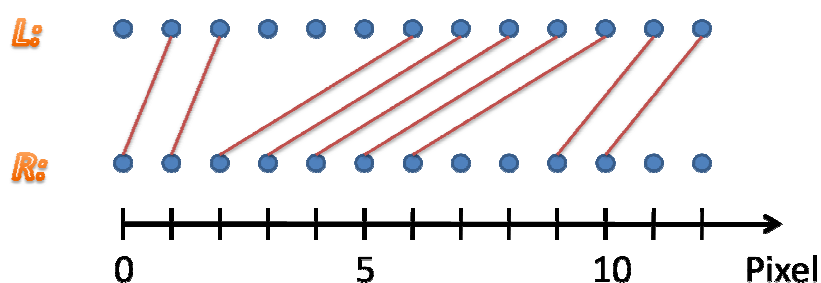
#### 2.1.2.2. Feature-based stereo matching technique

ในลำดับแรกของวิธีการ feature-based stereo matching จะแยกลักษณะเด่นที่มีความคงที่ภายใต้มุมมองที่เปลี่ยนแปลงไปออกมาก่อนการประมวลผลภาพทั้งคู่ หลังจากนั้นจะเข้าสู่

กระบวนการจับคู่คุณลักษณะที่สัมพันธ์กันด้วยการตรวจสอบจากลักษณะเด่นในข้างต้น ลักษณะเหล่านี้ถูกใช้อย่างกว้างขวางในหลายๆ งาน ลักษณะเด่นขั้นต้นที่ใช้งานกันคือ ขอบภาพ มุม กลุ่มเส้น กลุ่มเส้นโค้ง ส่วนลักษณะเด่นขั้นสูง ได้แก่ วงกลม วงรี และ บริเวณหลายเหลี่ยม ทั้งนี้ขึ้นอยู่กับข้อจำกัดของรูปภาพด้วย ระบบ feature-based stereo matching มักไม่ได้จำกัดอยู่เพียงชนิดของลักษณะเด่นอย่างเฉพาะเจาะจงอย่างใดอย่างหนึ่ง สามารถใช้ชนิดของลักษณะเด่นได้หลายลักษณะ

### 2.1.3. Depth Discontinuities by Pixels-to-Pixels Stereo

Depth Discontinuities by Pixels-to-Pixels Stereo (P2P) [13], [14] เป็นอัลกอริทึมที่ใช้หาพิกเซลที่มีความสัมพันธ์กันระหว่างภาพสองภาพ สามารถจัดอยู่ในกลุ่มของเทคนิค intensity-based stereo matching เนื่องจากอัลกอริทึมนี้หาพิกเซลที่มีความสัมพันธ์กันในแต่ละแถวของภาพ โดยใช้ความเข้มของพิกเซลในการตรวจสอบว่ามีความสัมพันธ์กันหรือไม่ โดยยึดภาพด้านซ้ายเป็นหลักแล้วหาพิกเซลที่ตรงกันในภาพด้านขวาดังรูปที่ 2-4 ผลที่ได้จากอัลกอริทึมเป็นแผนผังคิสพาริตี (disparity map) ซึ่งเป็นข้อมูลความลึกของภาพบรรจุอยู่ ข้อดีของอัลกอริทึมนี้คือภาพต้นฉบับที่นำมาใช้เพื่อหาแผนผังคิสพาริตีนั้นไม่จำเป็นต้องมีการเตรียมภาพก่อน เพียงแต่แปลงภาพให้เป็นภาพในระดับสีเทาเท่านั้น



รูปที่ 2-4 ตัวอย่างพิกเซลที่เหมือนกัน จากภาพซ้ายที่ตรงกับภาพขวา

อัลกอริทึม P2P มีมาตรวัดความลึกของทุกๆ คู่ลำดับของพิกเซลที่ตรงกันในแต่ละเส้นกราดตรวจโดยสามารถวัดได้โดยใช้ cost function [13] ดังสมการที่ (4)

$$\gamma(M) = N_{occ}k_{occ} - N_mk_r + \sum_{i=1}^{N_m} d(x_i, y_i) \quad (4)$$

$\gamma(M)$  คือ cost function สำหรับลำดับ  $M$  จากรูปที่ 2-4 ลำดับ  $M$  คือ  $M = \{(1, 0), (2, 1), (6, 2), (7,3), (8, 4), (9, 5), (10,6), (11,9), (12,10)\}$  โดยที่

- $N_{occ}$  คือจำนวนของ occlusion
- $k_{occ}$  คือค่าคงที่ occlusion penalty
- $N_m$  จำนวนพิกเซลที่มีความสัมพันธ์กัน
- $k_r$  คือค่าคงที่ match reward
- $d(x_i, y_i)$  คือความแตกต่างระหว่างพิกเซล  $x_i$  และ  $y_i$

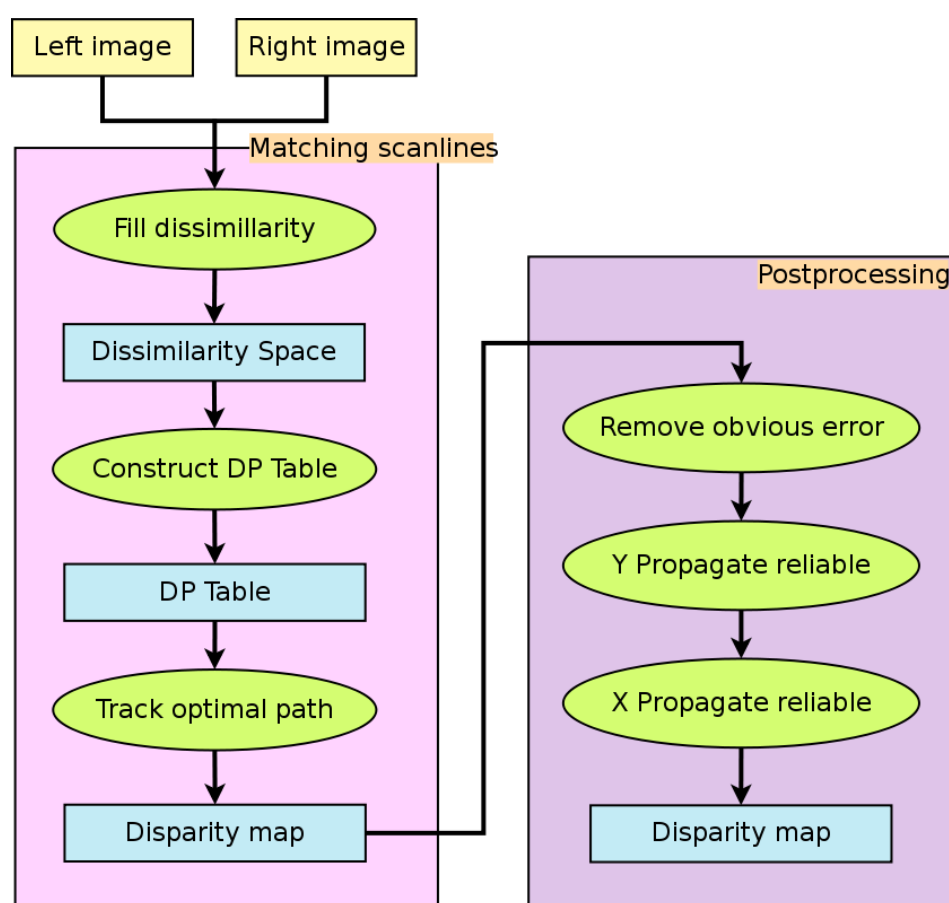
อัลกอริทึม P2P ประกอบไปด้วยสองกระบวนการหลักๆ แยกออกเป็นกระบวนการจับคู่พิกเซลอย่างเป็นอิสระในแต่ละแถว (matching scanlines independently) และ การกระจายข้อมูลระหว่างแถว (propagation information between scanlines) เพื่อให้ง่ายต่อการเข้าใจกระบวนการจับคู่พิกเซลอย่างเป็นอิสระในแต่ละแถวจะเรียกว่า “การจับคู่พิกเซลในแต่ละแถว” และการกระจายข้อมูลระหว่างแถวจะเรียกว่า “การประมวลผลขั้นปลาย” (post-processing) เนื่องจากเป็นการประมวลผลภายหลังที่ได้แผนผังคิสพาริตีจากกระบวนการจับคู่พิกเซลในแต่ละแถวมาแล้ว โดยที่การจับคู่พิกเซลในแต่ละแถวเป็นกระบวนการหาพิกเซลด้านขวาที่มีความสัมพันธ์กับพิกเซลด้านซ้ายในแต่ละแถวของภาพที่คำนวณ ซึ่งแต่ละแถวของภาพที่คำนวณเป็นอิสระจากกันคือไม่มีการแลกเปลี่ยนข้อมูลใดๆ ระหว่างการคำนวณ ในแต่ละแถวที่คำนวณหาพิกเซลที่ตรงกันนั้น ลักษณะของการเขียน โปรแกรมตรงส่วนนี้ใช้เทคนิคการกำหนดการพลวัต (dynamic programming) เพื่อหาลำดับของพิกเซลที่เหมาะสมที่สุดในแต่ละแถวของภาพโดยใช้ cost function  $\gamma(M)$  ช่วยลดความแตกต่างความเข้มของพิกเซล จำนวนและขนาดของ occlusion ด้วย ส่วนกระบวนการประมวลผลขั้นปลายปรับปรุงข้อมูลที่ได้จากกระบวนการแรกให้ดีขึ้นโดยใช้การเปลี่ยนแปลงความเข้มเพื่อปรับปรุงรอยต่อของขอบภาพและบริเวณที่เป็นไปได้ที่จะมีคิสพาริตีที่บรรจุอยู่

ทั้งสองกระบวนการของอัลกอริทึม P2P มีการทำงานหลายขั้นตอนด้วยกันโดยอธิบายตามขั้นตอนการเขียน โปรแกรมได้ดังรูปที่ 2-5 เริ่มต้นด้วยการนำภาพซ้ายและภาพขวาเข้าสู่อัลกอริทึม P2P โดยผ่านกระบวนการจับคู่พิกเซลในแต่ละแถวเป็นกระบวนการแรก ซึ่งกระบวนการนี้เริ่มด้วยการนำข้อมูลภาพซ้ายและภาพขวาในแต่ละแถวที่ตรงกันมาหา dissimilarity ซึ่งเป็นไปดังสมการที่ (5) เมื่อผ่านกระบวนการนี้จะได้ชุดข้อมูล dissimilarity เพื่อนำไปสร้างตาราง DP โดยที่ชุดข้อมูล dissimilarity มีขนาด  $\Delta \times w$  ( $\Delta$  คือจำนวนระดับพาริตีสูงสุด ซึ่งตัวแปรนี้ผู้ใช้งานอัลกอริทึม P2P สามารถระบุได้ และ  $w$  แทนความกว้างของภาพ) ตาราง DP สร้างจากการนำชุดข้อมูล dissimilarity มาผ่าน cost function ในสมการที่ (4) หลังจากนั้นหาคำตอบที่เหมาะสมที่สุดโดยใช้อัลกอริทึม

Faster ซึ่งผลที่ได้คือแผนผังดิสพาริตีซึ่งเป็นผลลัพธ์ของกระบวนการนี้ อัลกอริทึม Faster ใช้เวลาในการประมวลผลเฉลี่ยโดยประมาณ  $O(w\Delta \log \Delta)$  ในแต่ละแถวของภาพ

$$d(x_i, y_i) = |I_L(x_i) - I_R(y_i)| \quad (5)$$

โดยที่  $d(x_i, y_i)$  คือ dissimilarity  $I_L$  ความเข้มของพิกเซลบนเส้นกราดตรวจภาพซ้าย  $I_R$  ความเข้มของพิกเซลบนเส้นกราดตรวจภาพขวา



รูปที่ 2-5 ภาพรวมของการทำงานของอัลกอริทึม P2P [15]

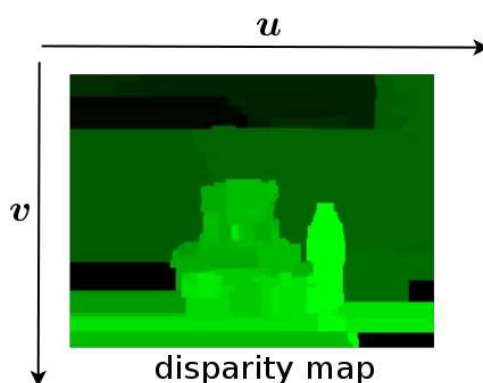
เมื่อได้แผนผังดิสพาริตีจากกระบวนการจับคู่พิกเซลในแต่ละแถวและนำผลที่ได้ผ่านกระบวนการประมวลผลขั้นปลายเพื่อให้ได้แผนผังดิสพาริตีที่มีคุณภาพดีขึ้น โดยการแลกเปลี่ยนข้อมูลระหว่างแถวและคอลัมน์โดยเริ่มจากการลดจำนวนข้อผิดพลาดแล้วเริ่มการแลกเปลี่ยนข้อมูลระหว่างคอลัมน์ และการแลกเปลี่ยนข้อมูลระหว่างแถวเพื่อหาพิกเซลที่น่าเชื่อถือได้ว่าจะปรากฏอยู่

ในบริเวณนั้นๆ จริงโดยอ้างอิงจากภาพซ้าย การคำนวณทั้งสองส่วนนี้ใช้ข้อมูลเกรเดียนต์ในแนวแกน  $x$  และแกน  $y$  จากภาพซ้ายเป็นข้อมูลในการคำนวณ เวลาเฉลี่ยในการประมวลผลโดยประมาณทั้งอัลกอริทึมคือ  $O(wh)$  เมื่อ  $h$  คือจำนวนแถว  $w$  คือจำนวนพิกเซลในแต่ละแถวหรือความกว้างของภาพ

อัลกอริทึม P2P ได้รับการยอมรับจนปรากฏเป็นไลบรารีของ Open Source Computer Vision (OpenCV) [17] ในชุด CvAux ชื่อ cvFindStereoCorrespondence โดยใช้ค่าคงที่ระบุชื่อ CV\_DISPARITY\_BIRCHFIELD อีกทั้งยังสามารถดาวน์โหลดโปรแกรมต้นฉบับมาทดลองใช้งานได้จากเว็บไซต์ส่วนตัวของผู้พัฒนาอัลกอริทึม<sup>1</sup>

## 2.2. การประมาณระยะทางของวัตถุจากแผนผังคิสพาริตี

มีผู้เสนอวิธีการประมาณระยะทางของวัตถุโดยใช้ภาพสเตอริโอหลายวิธีด้วยกันเช่น การใช้งาน Euclidean [27], [28] และ V-Disparity [25], [27] สำหรับ Euclidean นั้นเป็นการใช้งานโครงสร้างของข้อมูลในพิกัด 3 มิติ ซึ่งเป็นวิธีการประมวลผลภาพสเตอริโอที่แตกต่างจากอัลกอริทึม P2P เนื่องจาก P2P เป็นวิธีที่ให้ผลลัพธ์ออกมาเป็นแผนผังคิสพาริตี ดังนั้นวิทยานิพนธ์นี้จึงสนใจการใช้งาน V-Disparity เพื่อสร้างสมการประมาณระยะทางของวัตถุ เนื่องจาก V-disparity เป็นการแปลงแผนผังคิสพาริตีให้อยู่ในพิกัด  $u$  และ  $v$  ( $u, v$  coordinate system ดังรูปที่ 2-6) นั่นเอง



รูปที่ 2-6 พิกัด  $u$  และ  $v$  ใหม่สำหรับแผนผังคิสพาริตี

Labayrade และคณะ ได้เสนอการแปลงแผนผังคิสพาริตีให้อยู่ในรูปของ V-disparity [25] เพื่อใช้สำหรับตรวจจับอุปสรรคต่างๆ บนถนนที่มีลักษณะไม่เป็นระนาบ โดยเมื่อแปลงแผนผังคิส

<sup>1</sup> เว็บไซต์ส่วนตัวของ Stan Birchfield คือ [http://www.ces.clemson.edu/~stb/research/stereo\\_p2p](http://www.ces.clemson.edu/~stb/research/stereo_p2p)

พาริตีมาอยู่ในรูป V-disparity แล้ว ถนนจะมีลักษณะเป็นเส้นตรงที่ทำมุมแหลมกับแนวนอน ส่วนอุปสรรค หรือ ทิวทัศน์ต่างๆ ที่อยู่บนถนน หรือข้างถนนจะมีลักษณะเป็นเส้นตรงตามแนวตั้ง ซึ่งเทคนิคการใช้งาน V-disparity เพื่อใช้ตรวจจับสิ่งกีดขวางมีความทนทานและแม่นยำต่อการเปลี่ยนแปลงของสภาพแวดล้อม [29] ทั้งนี้ขึ้นอยู่กับคุณภาพของแผนผังคิสพาริตีด้วย เราสามารถแยกเส้นตรงในแนวตั้งเพื่อนำข้อมูลไปใช้งานต่อได้โดยใช้ Hough Transform นอกจากนี้ยังมีการขยายการศึกษาคุณลักษณะของแผนผังคิสพาริตีเมื่อแปลงข้อมูลให้อยู่ในพิกัด  $u$  และ  $v$  เพิ่มเติมเสนอโดย Zhencheng Hu และคณะ [30] เพื่อให้ครอบคลุมทั้ง U และ V-disparity

นอกจาก V-disparity จะใช้สำหรับตรวจจับอุปสรรคแล้ว V-disparity ยังสามารถบอกระดับความลึกของวัตถุขึ้นนั้นได้เช่นกัน ซึ่งเราสามารถนำเส้นตรงใน V-disparity มาประมาณค่าระยะทางจริงได้ ความยาวของเส้นตรงในแนวตั้งยังสามารถประมาณความสูงของวัตถุได้ด้วยเช่นกัน สำหรับ U-disparity นั้น สามารถบอกตำแหน่งของวัตถุในภาพได้ว่าวัตถุต้องอยู่บริเวณใดในภาพ ซึ่งส่วนนี้สามารถนำมาช่วยให้ผู้พิการทางสายตาสามารถรับรู้ข้อมูลสิ่งกีดขวางในการเดินทางไปยังสถานที่ต่างๆ ได้ ส่วนความยาวของเส้นตรงแนวนอนที่อยู่ใน U-disparity สามารถนำมาประมาณความกว้างของวัตถุได้

### 2.2.1. U-V-disparity

V-disparity [25] คือข้อมูลที่ประมวลผลมาจากแผนผังตามแนวแกน  $v$  (แนวตั้ง) โดยให้  $I_{\Delta}$  คือแผนผังคิสพาริตีที่ได้จากการประมวลผลภาพสเตอริโอ  $H$  เป็นฟังก์ชันของตัวแปร  $I_{\Delta}$  ดังนั้นจะได้สมการที่ (6) ดังนี้

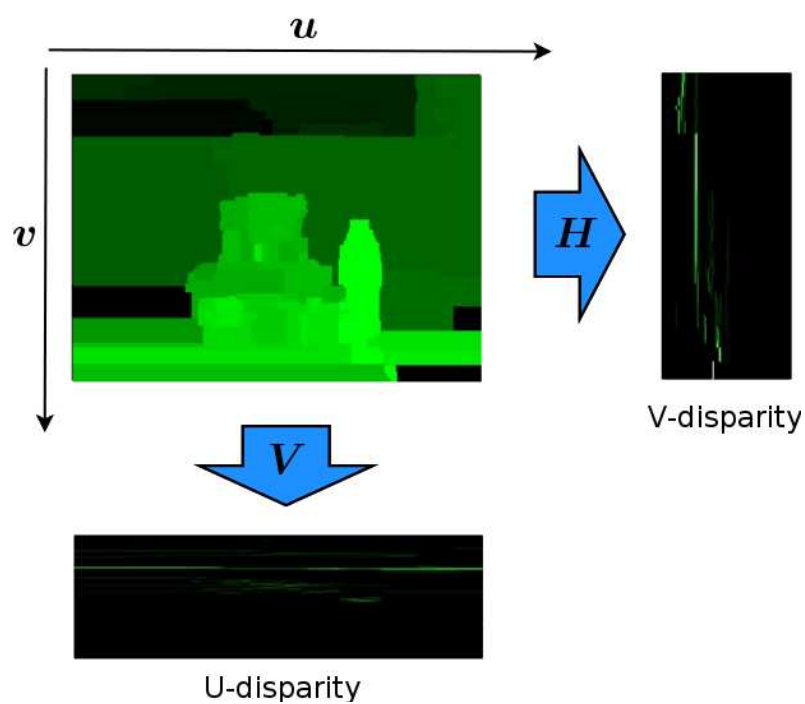
$$H(I_{\Delta}) = I_{\Delta v} \quad (6)$$

$I_{\Delta v}$  เรียกว่า V-disparity โดยที่  $H$  คือการสะสมจำนวนคิสพาริตีที่มีค่าคิสพาริตีเดียวกันที่ปรากฏอยู่บนรูปภาพแถวที่  $i$  เช่นเดียวกับ U-disparity [30] คือข้อมูลที่ประมวลผลมาจากแผนผังคิสพาริตีตามแนวแกน  $u$  และ  $V$  เป็นฟังก์ชันของ  $I_{\Delta}$  ดังนั้นจะได้สมการที่ (7) ดังนี้

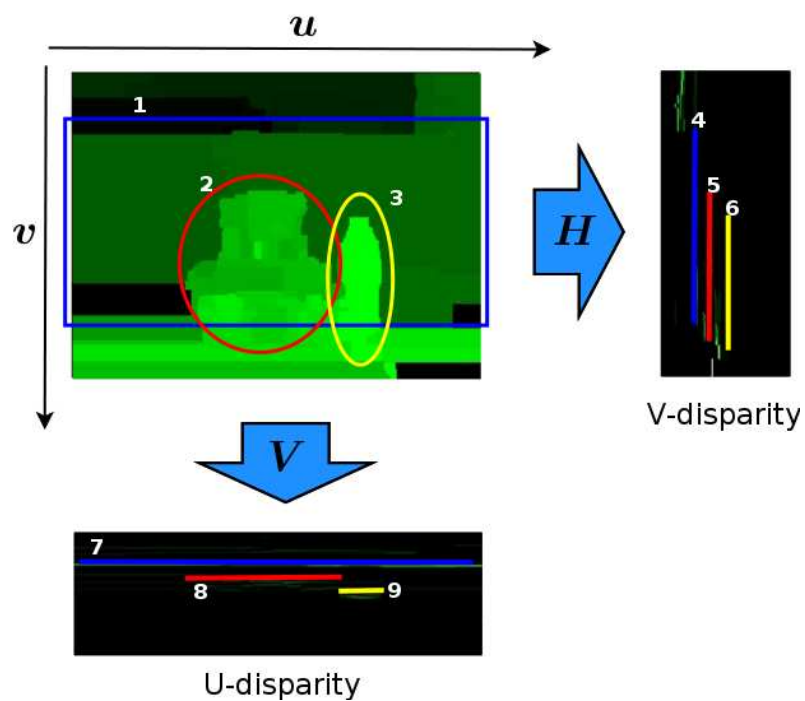
$$V(I_{\Delta}) = I_{\Delta u} \quad (7)$$

$I_{\Delta u}$  เรียกว่า U-disparity โดยที่  $V$  คือการสะสมจำนวนคิสพาริตีที่มีค่าคิสพาริตีเดียวกันที่ปรากฏอยู่บนรูปภาพคอลัมน์ที่  $j$  ซึ่งการใช้งานฟังก์ชัน  $H$  และ  $V$  เพื่อแปลงแผนผังคิสพาริตีดังรูปที่ 2-7





รูปที่ 2-7 การแปลงแผนผังดิสพาริตีไปเป็น U-V-disparity โดยใช้ฟังก์ชัน  $H$  และ  $V$



รูปที่ 2-8 ความสัมพันธ์ระหว่างวัตถุในแผนผังดิสพาริตีที่กับเส้นตรงใน U-V disparity

จากรูปที่ 2-8 แสดงความสัมพันธ์ระหว่างวัตถุที่อยู่ในแผนผังคิสพาริตีกับเส้นตรงในแนวตั้งของ V-disparity และเส้นตรงในแนวนอนของ U-disparity โดยที่

- เส้นตรงสีเหลืองใน U-V-disparity และวงรีสีเหลืองหรือหมายเลข 1 4 และ 7 ในแผนผังคิสพาริตีคือขวคน้ำ
- เส้นตรงสีแดงใน U-V-disparity และวงรีสีแดงหรือหมายเลข 2 5 และ 8 ในแผนผังคิสพาริตีคือตุ๊กตาหมี
- เส้นตรงสีน้ำเงิน ใน U-V-disparity และกรอบสีเหลี่ยมสีน้ำเงินหรือหมายเลข 3 6 และ 9 ในแผนผังคิสพาริตีคือผนังห้อง

V-disparity สามารถบอกได้ว่ามีวัตถุอยู่ในภาพหรือไม่ ส่วน U-disparity สามารถบอกได้ว่าวัตถุอยู่บริเวณใดของภาพ

### 2.2.2. การประมาณระยะทางของวัตถุด้วย V-disparity

เนื่องจากในแต่ละระดับของคิสพาริตีที่ได้จากการประมวลผลภาพสเตอริโอ นั้นมีความสัมพันธ์กับระยะทางจริงของวัตถุ การที่จะประมาณค่าระยะทางจริงของวัตถุได้จะต้องมีการเก็บข้อมูลระยะทางที่สอดคล้องกับแต่ละระดับของคิสพาริตี เพื่อนำข้อมูลที่ได้มาสร้างสมการในการประมาณค่าระยะทาง เนื่องจากคิสพาริตีที่ใช้มีหลายระดับจึงใช้วิธีเก็บข้อมูลระยะทางจริงเป็นช่วงๆ แล้วใช้วิธีการประมาณค่าในช่วง (interpolation) ด้วยสมการพหุนาม

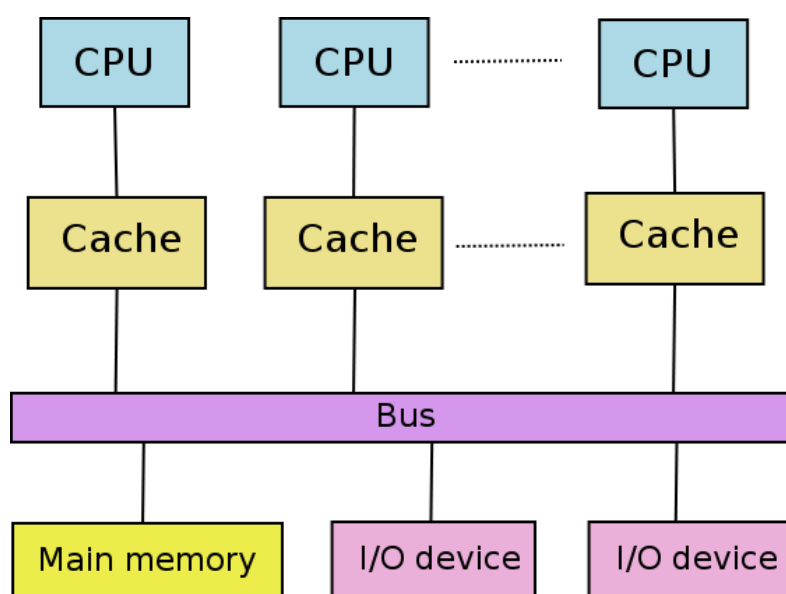
สำหรับข้อมูลที่เราเลือกเพื่อมาแทนค่าในสมการที่กล่าวไว้ข้างต้นนั้นคือค่าตามแนวแกน x ของเส้นตรงในแนวตั้งจาก V-disparity หรือค่าตามแนวแกน y ของเส้นตรงในแนวนอนจาก U-disparity เนื่องจากค่าดังกล่าว คืออุปสรรคหรือสิ่งกีดขวางที่ปรากฏในภาพตั้งต้น เมื่อแทนค่าที่สนใจลงในสมการจะได้ระยะทางของวัตถุโดยประมาณออกมา เพื่อนำไปใช้งานต่อไป โดยสามารถนำข้อมูลนี้เสนอต่อผู้พิการโดยใช้เสียงหรือเทคโนโลยีอื่น

จากรูปที่ 2-8 จะเห็นได้ว่า U-V-disparity นั้นมีการแสดงวัตถุทั้งสามออกมาเป็นลำดับโดยที่ภาพจริงจะมีวัตถุเรียกกัน โดยวัตถุที่อยู่ใกล้กับกล้องที่ใช้จับภาพมากที่สุดคือ ขวคน้ำ ตุ๊กตาหมี และผนังตามลำดับ หมายเลข 4, 5 และ 6 จาก V-disparity คือข้อมูลที่แทนความสูงของวัตถุเหล่านั้นโดยเรียงตามระดับความลึก ซึ่งก็คือระยะทางจากกล้องจนถึงวัตถุนั่นเอง หมายเลข 4 จะมีระยะทางใกล้ที่สุดในขณะที่หมายเลข 6 จะมีระยะทางใกล้ที่สุด เช่นเดียวกับหมายเลข 7, 8 และ 9 ที่แทนความกว้างของของวัตถุดังกล่าวตามระยะทางจากกล้องจนถึงวัตถุ หมายเลข 9 จะอยู่ใกล้กล้องที่สุด และหมายเลข 7 จะอยู่ไกลกล้องที่สุด

### 2.3. การประมวลผลแบบขนาน

ปัจจุบันเทคโนโลยีหน่วยประมวลผลกลางเน้นการพัฒนาให้หนึ่งหน่วยประมวลผลกลางมีหน่วยประมวลผลมากกว่าหนึ่งแกนหรือที่เรียกว่าเทคโนโลยีมัลติคอร์ (Multi-core) เพื่อแก้ปัญหาข้อจำกัดของสัญญาณนาฬิกา ซึ่งใช้เป็นตัวกำหนดความเร็วของการประมวลผล หน่วยประมวลผลดังกล่าวได้รับความนิยมอย่างรวดเร็วส่งผลให้มี CPU มีประสิทธิภาพสูงขึ้นในราคาที่ถูกลงอย่างต่อเนื่อง ดังนั้นหากนำโปรแกรมแบบตามลำดับที่เคยใช้สำหรับคอมพิวเตอร์ที่มีหน่วยประมวลผลเดี่ยวมาประมวลผลบนคอมพิวเตอร์ที่มีหลายหน่วยประมวลผลจะไม่สามารถใช้ประสิทธิภาพคอมพิวเตอร์ที่มีหน่วยประมวลผลแบบมัลติคอร์ในปัจจุบันได้อย่างเต็มที่ ดังนั้นเพื่อให้สามารถใช้ประโยชน์จากหน่วยประมวลผลที่เพิ่มขึ้น การพัฒนาโปรแกรมแบบขนานจึงมีความสำคัญควบคู่กันไปด้วย

คอมพิวเตอร์ส่วนบุคคลที่ใช้หน่วยประมวลผลกลางแบบมัลติคอร์ในปัจจุบันจัดได้ว่าเป็นสถาปัตยกรรมคอมพิวเตอร์แบบซิมเมตริกมัลติโพรเซสเซอร์ เนื่องจากใน CPU มีหน่วยประมวลผลที่เหมือนกันมากกว่าหนึ่งหน่วย โดยใช้งานหน่วยความจำหลักร่วมกันมีโครงสร้างดังรูปที่ 2-9



รูปที่ 2-9 สถาปัตยกรรมคอมพิวเตอร์แบบซิมเมตริกมัลติโพรเซสเซอร์ [16]

### 2.3.1. มิติในการประมวลผลแบบขนาน

การพัฒนาโปรแกรมแบบขนานสามารถแบ่งได้สองลักษณะด้วยกัน [16] คือการประมวลผลขนานเชิงข้อมูล (Data parallelism) และการประมวลผลขนานเชิงฟังก์ชัน (Functional parallelism) โดยมีรายละเอียดดังต่อไปนี้

#### 1) การประมวลผลขนานเชิงข้อมูล

การประมวลผลขนานเชิงข้อมูลนั้นสามารถทำได้ในกรณีที่สามารถแบ่งข้อมูลที่ใช้ประมวลผลออกเป็นส่วนย่อยๆ ได้หลายส่วน และสามารถประมวลผลข้อมูลส่วนย่อยเหล่านั้นพร้อมกันได้ด้วยโปรแกรมชุดเดียวกันบนหลายหน่วยประมวลผลแยกจากกัน เช่น การบวกเวกเตอร์ การประมวลผลภาพ (Image Processing) บางลักษณะ เช่น การหาขอบภาพ (Edge Detection) การแปลงภาพสีเป็นภาพระดับสีเทา เป็นต้น

#### 2) การประมวลผลขนานเชิงฟังก์ชัน

การประมวลผลขนานเชิงฟังก์ชันสามารถทำได้ในกรณีที่โปรแกรมสามารถแบ่งออกเป็นหลายๆ งานย่อยได้ โดยในแต่ละส่วนของโปรแกรมประมวลผลเป็นอิสระต่อกัน ตัวอย่างเช่น การทำงานของคอมพิวเตอร์ ที่มีการตรวจสอบข้อผิดพลาดของซอร์สโค้ดแล้วแปลงซอร์สโค้ดไปเป็นโปรแกรมที่พร้อมจะทำงาน หากมีการคอมไพล์หลายไฟล์อาจจะแยกส่วนของการตรวจสอบออกจากส่วนที่ทำหน้าที่แปลงซอร์สโค้ด เป็นต้น

สำหรับวิทยานิพนธ์นี้เลือกใช้การประมวลผลขนานเชิงข้อมูลในการประมวลผลภาพสเตอริโอ เนื่องจากข้อมูลที่นำเข้าของระบบเป็นข้อมูลภาพวิดีโอซึ่งมีปริมาณมาก และเวลาที่ใช้กับกระบวนการทำงานทั้งหมดของระบบเสียไปกับการประมวลผลภาพสเตอริโอ ส่งผลให้ผลตอบสนองต่อหนึ่งภาพนั้นใช้เวลานาน หากแบ่งข้อมูลภาพออกเป็นส่วนย่อยๆ แล้วนำไปประมวลผลแยกจากกันบนหลายหน่วยประมวลผลจะช่วยลดเวลาของผลตอบสนองลงได้

### 2.3.2. การวัดประสิทธิภาพของการประมวลผลแบบขนาน

ในปัจจุบันการวัดประสิทธิภาพของการประมวลผลแบบขนานที่ใช้กันทั่วไปคือการหาค่าสปีดอัป [16] (speedup) ซึ่งค่านี้จะบอกถึงอัตราส่วนของประสิทธิภาพที่เพิ่มมากขึ้นจากการประมวลผลแบบขนานเมื่อเทียบกับโปรแกรมแบบตามลำดับดังแสดงในสมการที่ (8)

$$s = \frac{T_s}{T_p} \quad (8)$$

เมื่อ  $s$  คือค่าสปีดอัป  $T_s$  คือเวลาในการประมวลผลแบบตามลำดับ  $T_p$  คือเวลาในการประมวลผลแบบขนาน  $p$  คือจำนวนหน่วยประมวลผล โดยที่ค่าสปีดอัปสูงสุดที่สามารถทำได้คือ  $p$  ดังนั้นค่าสปีดอัปที่ได้จะไม่เกิน  $p$  ดังสมการที่ (9)

$$s \leq p \quad (9)$$

ค่าสปีดอัปสามารถนำมาใช้ในการคำนวณค่าประสิทธิภาพการประมวลผล [16] (efficiency:  $e$ ) ได้ โดยค่าดังกล่าววัดได้จากสัดส่วนของเวลาที่หน่วยประมวลผลถูกใช้ในงานคำนวณที่เป็นประโยชน์เทียบกับเวลาว่างที่ต้องหยุดรอการแลกเปลี่ยนข้อมูลหรือ โอเวอร์เฮด (overhead) ประเภทอื่นๆ กล่าวอีกนัยหนึ่งได้ว่าค่าประสิทธิภาพการประมวลผลจะบอกถึงอัตราการใช้ประโยชน์ของหน่วยประมวลผลนั่นเอง จากตัวแปรที่ใช้ในการคำนวณค่าสปีดอัป ค่าประสิทธิภาพการประมวลผลสามารถคำนวณได้ดังสมการที่ (10)

$$e = \frac{s}{p} \quad (10)$$

### 2.3.3. Message Passing Interface

Message Passing Interface (MPI) [18] คือข้อกำหนดมาตรฐานของไลบรารีสำหรับการพัฒนาโปรแกรมแบบขนานด้วยรูปแบบการส่งผ่านข้อความระหว่างโพรเซส มาตรฐานต่างๆ ของ MPI มี MPI Forum เป็นกลุ่มที่ประกอบด้วยผู้แทนจากหน่วยงานต่างๆ หลายหน่วยงาน เป็นผู้ดูแลและกำหนดมาตรฐานต่างๆ ในปัจจุบัน MPI Forum ออกมาตรฐาน MPI รุ่น 2.1 เป็นรุ่นล่าสุด เมื่อวันที่ 4 กันยายน 2551 MPI มีข้อกำหนดรูปแบบในหลายภาษา เช่น C, C++, Fortran90 และ Fortran77 เป็นต้น ตัวอย่างไลบรารีที่พัฒนาจากข้อกำหนด MPI มีดังนี้

- MPICH [19] เป็นการพัฒนาตามมาตรฐาน MPI ให้สามารถใช้งานได้หลายระบบปฏิบัติการ เช่น Windows (NT/2000/XP/2003), Solaris และ Linux เป็นต้น
- HP MPI [21] เป็นอีกหนึ่งการพัฒนาไลบรารีตามมาตรฐาน MPI สำหรับระบบคอมพิวเตอร์ของ Hewlett Packard

- MacMPI [22] ไลบรารีตามมาตรฐาน MPI สำหรับระบบปฏิบัติการ Mac OS
- OpenMPI [23] ไลบรารีที่พัฒนาตามมาตรฐาน MPI-1.2 และ MPI-2 โดยใช้แนวทางการพัฒนาแบบเปิดสามารถใช้งานได้บนระบบปฏิบัติการ Unix ทั่วไป

โดยในวิทยานิพนธ์นี้เลือกใช้ MPICH เนื่องจาก MPICH มีการพัฒนาไลบรารีมาอย่างต่อเนื่องตามมาตรฐานของ MPI อีกทั้งยังสามารถใช้งานได้หลายระบบปฏิบัติการ จึงสามารถเปลี่ยนระบบปฏิบัติการในการทดสอบหรือใช้งานจริงได้ในอนาคตโดยไม่ต้องเปลี่ยนแปลงซอร์สโค้ดมากนัก อีกทั้ง MPICH มีชื่อเสียงสำหรับการพัฒนาโปรแกรมแบบขนานมาควบคู่กับ MPI จึงมั่นใจได้ว่าไลบรารีนี้จะไม่มีการหยุดพัฒนาในอนาคตอันใกล้

เพื่อให้ครอบคลุมการพัฒนาโปรแกรมแบบขนานในลักษณะต่างๆ MPI ได้เตรียมฟังก์ชันมาตรฐานเอาไว้มากมาย โดยสามารถจัดกลุ่มได้ดังต่อไปนี้

- การติดต่อสื่อสารระหว่างเครื่องสองเครื่อง (Point to Point Communication)
- การติดต่อสื่อสารระหว่างกลุ่มเครื่อง (Collective Communication)
- การจัดการกลุ่มข้อมูล (Data Grouping Management)
- การจัดการสภาพแวดล้อม (Environment Management)

ฟังก์ชัน MPI ที่ใช้ในวิทยานิพนธ์นี้มีหลายฟังก์ชันด้วยกันสามารถดูได้จากภาคผนวก ก แต่มีฟังก์ชันหลักที่เกี่ยวข้องกับการออกแบบระบบในวิทยานิพนธ์ได้แก่

- MPI::Bcast เป็นการกระจายข้อความจากโพรเซสหนึ่งไปยังกลุ่มของโพรเซสหนึ่ง โดยที่ทุกๆ โพรเซสที่ติดต่อด้วยจะได้รับข้อความเดียวกันทั้งหมด
- MPI::Scatter เป็นการกระจายข้อความจากโพรเซสหนึ่งไปยังกลุ่มของโพรเซสหนึ่ง คล้ายกับ MPI::Bcast แต่ทุกๆ โพรเซสที่ติดต่อด้วยจะไม่ได้รับข้อความเดียวกัน
- MPI::Gather เป็นการทำงานที่ตรงกันข้ามกับ MPI::Scatter คือเป็นการรวบรวมข้อมูลกลับจากกลุ่มของโพรเซสมายังโพรเซสที่ต้องการ
- MPI::Allgather มีลักษณะการทำงานคล้ายกับ MPI::Gather แต่ต่างกันที่ทุกๆ โพรเซสจะทำการรวบรวมข้อมูลในลักษณะเดียวกันทั้งหมด ดังนั้นเมื่อเสร็จสิ้นกระบวนการ ทุกๆ โพรเซสจะมีข้อมูลเหมือนกัน
- MPI::Reduce เป็นการรวบรวมข้อมูลกลับมายังโพรเซสที่ต้องการ โดยข้อมูลดังกล่าวจะผ่านการประมวลผลตามตัวดำเนินการที่ระบุไว้ในพารามิเตอร์ของฟังก์ชัน ซึ่งที่ใช้งานในวิทยานิพนธ์นี้มีดังต่อไปนี้

- MPI::MAX หาค่าสูงสุดของข้อมูลทั้งหมดจากทุกๆ โพรเซส
- MPI::MIN หาค่าต่ำสุดของข้อมูลทั้งหมดจากทุกๆ โพรเซส
- MPI::SUM หาผลรวมของข้อมูลทั้งหมด
- MPI::Allreduce มีลักษณะการใช้งานคล้ายกับ MPI::Reduce มาก เพียงแต่เมื่อจบการดำเนินการทุกๆ โพรเซสจะมีข้อมูลชุดเดียวกัน

### 2.3.4. MPICH

MPICH [19] เป็นไลบรารีที่พัฒนาตามข้อกำหนดของ MPI โดยสามารถใช้งานได้บนหลายระบบปฏิบัติการ ทั้ง Windows (NT/XP/2003), ลินุกซ์, ยูนิกซ์ ฯลฯ แต่ต้องเลือกตัวจัดการโปรเซสและช่องทางสื่อสาร (communication device) ให้เหมาะสมกับระบบปฏิบัติการและสถาปัตยกรรมคอมพิวเตอร์ที่จะใช้งานนั้นๆ MPICH ได้พัฒนามาจนถึงเวอร์ชัน 2 เพื่อรองรับมาตรฐาน MPI 2 ซึ่งเป็นที่นิยมแพร่หลายในปัจจุบัน

MPICH มีหลายช่องทางสื่อสารให้เลือกใช้เพื่อให้เหมาะสมสำหรับการใช้งานคอมพิวเตอร์ในแต่ละสถาปัตยกรรม โดย MPICH 2 ได้เตรียมช่องทางสื่อสารพื้นฐานไว้ให้ หรือที่รู้จักกันคือ “ch3” (เป็นรุ่นที่ 3 ของ “channel” interface) [20] เป็นค่าปริยาย ซึ่งสามารถเลือกรูปแบบการสื่อสาร (communication method) ย่อยลงไปได้อีกดังนี้

- ch3:nemesis เป็นวิธีการใหม่ที่มีประสิทธิภาพสูงโดยเริ่มใช้เป็นมาตรฐานตั้งแต่ MPICH 2 เวอร์ชัน 1.1 วิธีการนี้ใช้หน่วยความจำร่วม (shared-memory) เพื่อส่งผ่านข้อความระหว่างโปรเซสคอมพิวเตอร์บนคอมพิวเตอร์เครื่องเดียวกัน และใช้เครือข่ายคอมพิวเตอร์ (computer network) สำหรับส่งผ่านข้อความไปยังโปรเซสที่อยู่บนเครื่องอื่น อีกทั้งยังสนับสนุนการใช้งาน MPI\_THREAD\_MULTIPLE อีกด้วย
- ch3:sock วิธีการนี้ใช้ซ็อกเก็ตสำหรับทุกๆ การติดต่อสื่อสารระหว่างโปรเซส นอกจากนี้ยังสนับสนุน MPI\_THREAD\_MULTIPLE เช่นกัน
- ch3:ssm วิธีการนี้ใช้ซ็อกเก็ตในการสื่อสารระหว่างคอมพิวเตอร์ และใช้หน่วยความจำร่วมในการสื่อสารระหว่างโปรเซสบนคอมพิวเตอร์เครื่องเดียวกัน
- ch3:shm วิธีการนี้ใช้หน่วยความจำร่วมเท่านั้นต้องใช้งานบนคอมพิวเตอร์เครื่องเดียวในสถาปัตยกรรม SMP เท่านั้น อีกทั้งไม่สนับสนุน MPI dynamic process routines เช่น MPI\_Comm\_spawn

โดยทั่วไป MPICH 2 จะตั้งค่าปริยายในการติดตั้งไว้ที่ `ch3:nemesis` เนื่องจากวิธีการสื่อสารนี้ให้ประสิทธิภาพสูงสุด สำหรับแพลตฟอร์มที่ไม่สามารถใช้งานวิธีการนี้ได้ผู้เชี่ยวชาญแนะนำให้ใช้งานวิธี `ch3:sock` แทน

MPICH 2 เน้นการแบ่งงานไปยังหลายๆ โพรเซส โดยใน MPICH 2 สนับสนุนวิธีการจัดการโพรเซส 4 ลักษณะ ดังนี้

- `mpd` เป็นค่าปริยายของ MPICH ซึ่งพัฒนาตามข้อกำหนด Process Management Interface (PMI) เป็นตัวจัดการโพรเซสที่อาศัยโพรเซสดีมอน (daemon process) ช่วยจัดการ
- `smpd` เป็นตัวจัดการโพรเซสที่สามารถใช้งานได้ทั้ง Linux และ Windows อีกทั้งเป็นตัวจัดการโพรเซสแบบเดียวที่สามารถใช้งานได้บนระบบปฏิบัติการ Windows
- `gforker` เป็นอีกหนึ่งตัวจัดการโพรเซสที่สามารถสร้างโพรเซสบนเครื่องๆ เดียวได้ดี มีประโยชน์ในการแก้ไขจุดบกพร่องของวิธีการสื่อสารเมื่อใช้เครื่องคำนวณเพียงเครื่องเดียวแต่มีหลายหน่วยประมวลผล โดยใช้วิธีการสื่อสารระหว่างโพรเซสผ่านหน่วยความจำร่วม
- `hydra` เป็นตัวจัดการโพรเซสที่สร้างโพรเซสโดยใช้ native launcher daemons ไม่สนับสนุน `MPI_Comm_spawn` และรูทีนโพรเซสพลวัตชนิดอื่น (dynamic process routines)

โดยสรุปแล้วสำหรับคอมพิวเตอร์ส่วนบุคคลที่ใช้งาน CPU แบบหลายแกนนั้นสามารถจัดอยู่ในสถาปัตยกรรมแบบ SMP จึงควรใช้ตัวจัดการโพรเซสแบบ `gforker` เนื่องจากไม่จำเป็นต้องอาศัยโพรเซสดีมอนและง่ายต่อการใช้งาน ในส่วนช่องทางการสื่อสารระหว่างโพรเซสนั้นเลือกใช้ `ch3:nemesis` เนื่องจากจะให้ประสิทธิภาพสูงสุดตามที่ระบุไว้ในคู่มือ โดยช่องทางการสื่อสารนี้จะใช้งานหน่วยความจำร่วมเพื่อแลกเปลี่ยนข้อความแทนการใช้ซ็อกเก็ตของเครือข่ายคอมพิวเตอร์ ส่งผลให้มีการแลกเปลี่ยนข้อมูลที่ดีขึ้นเมื่อใช้งานบนสถาปัตยกรรมคอมพิวเตอร์แบบ SMP

## 2.4. สรุป

ในบทนี้นำเสนอทฤษฎีและหลักการพื้นฐานที่จำเป็นต่อการสร้างระบบตรวจสอบสิ่งกีดขวางสำหรับผู้พิการทางสายตาด้วยสเตอริโอวิชัน โดยเริ่มจากการประมวลผลภาพสเตอริโอไปจนถึงอัลกอริทึม P2P ซึ่งใช้จับคู่ภาพสเตอริโอที่ใช้ในงานวิจัยนี้ ในส่วนการประมวลผลแบบขนานนั้นแนะนำไลบรารีที่สำหรับการประมวลผลแบบขนานโดยการส่งผ่านข้อความ ซึ่งเลือกใช้ MPICH



โดยเลือก gforker เป็นตัวจัดการโปรเซส และใช้ ch3:nemesis เป็นช่องทางการสื่อสารระหว่างโปรเซสเพื่อให้สามารถดึงประสิทธิภาพของหน่วยประมวลผลแบบมัลติคอร์มาใช้งานได้มากที่สุด อีกทั้งแนะนำการวัดประสิทธิภาพของการประมวลผลแบบขนานที่ใช้กันอยู่ในปัจจุบันเพื่อใช้วัดประสิทธิภาพของโปรแกรมที่ได้จัดทำขึ้นอีกด้วย

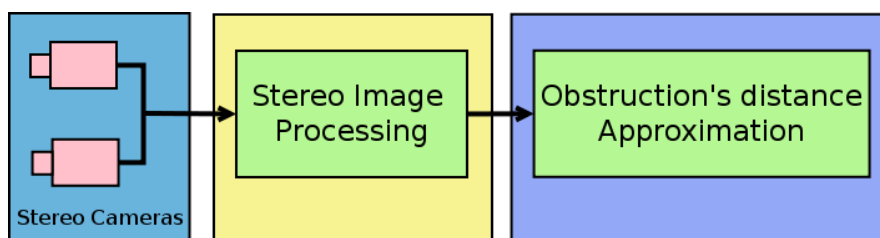
## บทที่ 3

### การออกแบบและพัฒนาระบบ

#### 3.1. ภาพรวมของระบบ

ระบบตรวจสอบสิ่งกีดขวางสำหรับผู้พิการทางสายตาแบ่งออกเป็นสองส่วนหลักคือ ส่วนตรวจสอบวัตถุสิ่งกีดขวางด้วยกล้องสเตอริโอวิชันและประมาณระยะทางของสิ่งกีดขวาง และส่วนติดต่อกับผู้พิการทางสายตา ซึ่งวิทยานิพนธ์นี้จะเน้นส่วนตรวจสอบวัตถุสิ่งกีดขวางด้วยกล้องสเตอริโอวิชันและประมาณระยะทางของสิ่งกีดขวางเท่านั้น

ระบบตรวจสอบสิ่งกีดขวางสำหรับผู้พิการทางสายตาในส่วนตรวจสอบวัตถุสิ่งกีดขวางด้วยกล้องสเตอริโอวิชันและประมาณระยะทางของสิ่งกีดขวาง ประกอบด้วยสามส่วนย่อยดังแสดงในรูปที่ 3-1 คือ กล้องสเตอริโอซึ่งทำหน้าที่บันทึกทัศนียภาพหน้ากล้องสเตอริโอเพื่อนำภาพที่ได้ไปประมวลผลภาพต่อไป ส่วนต่อมาคือการประมวลผลภาพสเตอริโอซึ่งผลที่ได้จากกระบวนการนี้คือแผนผังคิสพาริตีที่ส่วนสุดท้ายของกระบวนการคือ การประมาณระยะทางและตำแหน่งของวัตถุจากแผนผังคิสพาริตี



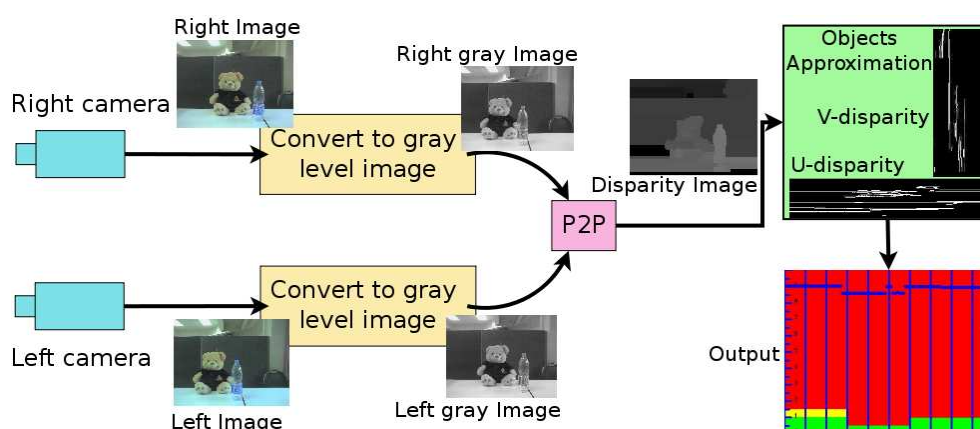
รูปที่ 3-1 ส่วนตรวจสอบวัตถุสิ่งกีดขวางด้วยกล้องสเตอริโอวิชันและประมาณระยะทางของสิ่งกีดขวาง

#### 3.2. การออกแบบระบบ

เนื่องจากระบบตรวจสอบสิ่งกีดขวางสำหรับผู้พิการทางสายตาที่ได้จัดทำขึ้นนี้ ไม่ต้องการให้ผู้พิการใช้งานระบบนี้แทนไม้เท้าขาวซึ่งเป็นอุปกรณ์การเดินทางของผู้พิการที่มีอยู่เดิม แต่เป็นการเพิ่มข้อมูลของสิ่งแวดล้อมที่มีระยะไกลกว่าไม้เท้าขาวให้แก่ผู้พิการทางสายตาเพื่อให้ผู้พิการทางสายตาสามารถเลือกทิศทางในการเคลื่อนที่ได้ดีขึ้น ดังนั้นระบบที่ได้ออกแบบและจัดทำขึ้นนี้ จะมีความสามารถในการตรวจสอบวัตถุตั้งแต่ระยะหนึ่งเมตรเป็นต้นไป โดยระยะห่างจากตัวผู้

พิการออกไปหนึ่งเมตรนั้น ผู้พิการจะต้องใช้ไม้เท้าตรวจสอบเอง ซึ่งการใช้งานไม้เท้าตรวจสอบสิ่งกีดขวาง ผู้พิการทางสายตาสามารถใช้งานไม้เท้าขาได้อย่างชำนาญอยู่แล้ว อีกทั้งสาเหตุสำคัญที่ผู้วิจัยไม่ได้ออกแบบให้ใช้งานระบบตรวจสอบสิ่งกีดขวางสำหรับผู้พิการทางสายตาแทนไม้เท้าขา เนื่องจากผู้พิการทางสายตาคุ้นเคยกับไม้เท้าขาเป็นอย่างดี หากนำเอาระบบเหล่านี้ไปแทนที่ไม้เท้าขาในทันที อาจส่งผลให้ผู้พิการเกิดความไม่มั่นใจต่อระบบและอาจปฏิเสธระบบตรวจสอบสิ่งกีดขวางที่ได้จัดทำขึ้นในลำดับต่อมา นอกจากนี้ยังเป็นการสนับสนุนให้ผู้พิการไม่พึ่งพาเทคโนโลยีมากเกินไป เนื่องจากหากไม่มีอุปกรณ์ดังกล่าวช่วยเหลือแล้วผู้พิการอาจไม่สามารถใช้ชีวิตได้ตามปกติ

ภาพรวมของระบบแสดงไว้ดังรูปที่ 3-2 โดยเริ่มจากการบันทึกภาพจากกล้องเว็บแคม เมื่อได้ภาพจากกล้องซ้ายและกล้องขวามาแล้วปรับภาพจากภาพสีให้กลายเป็นภาพระดับสีเทา จากนั้นนำมาหาความสัมพันธ์โดยผ่านอัลกอริทึม P2P เมื่อได้ภาพความสัมพันธ์แล้วเข้าสู่กระบวนการประมวลผลระยะทางด้วย U หรือ V-disparity ซึ่งจะมีการแบ่งช่วงเพื่อให้สะดวกต่อการรับรู้ของผู้ใช้งาน ข้อมูลที่ได้จากส่วนนี้เป็นระยะทางของสิ่งกีดขวางที่สามารถนำไปเสนอต่อผู้พิการทางสายตาต่อไป

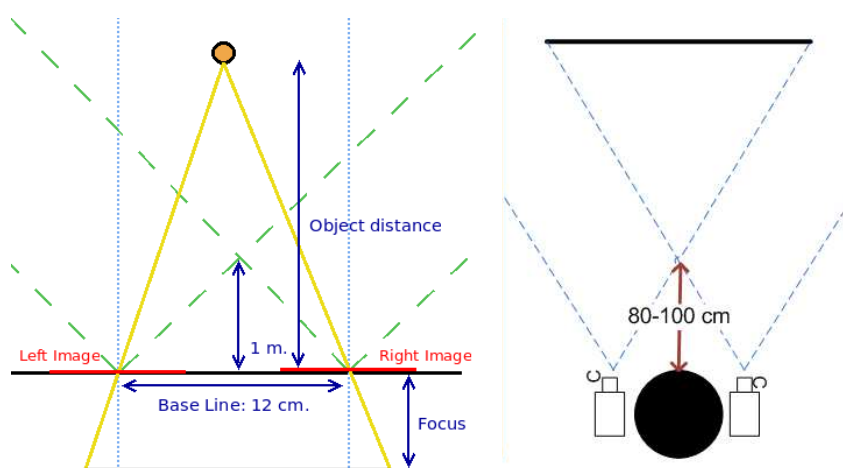


รูปที่ 3-2 กระบวนการประมวลผลภาพสเตอริโอ เริ่มจากจับภาพจากกล้องสเตอริโอ ผ่านอัลกอริทึมการประมวลผลภาพสเตอริโอจนได้แผนผังความสัมพันธ์ แล้วจึงนำแผนผังความสัมพันธ์ที่ได้ไปประมวลผลต่อตามที่ต้องการ

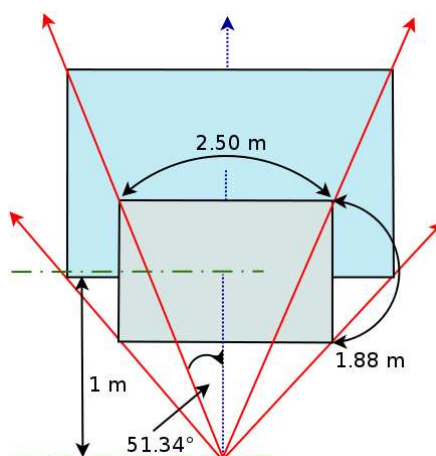
### 3.2.1. กล้องสเตอริโอวิชัน

โดยปรกติกล้องสเตอริโอในท้องตลาดมีราคาสูงมาก เนื่องจากมีผู้ใช้งานกล้องประเภทนี้น้อยและนิยมใช้งานในการวิจัยเป็นหลัก อีกทั้งกล้องที่ใช้ประกอบขึ้นเป็นกล้องสเตอริโอเป็นมักจะมีคุณภาพของอุปกรณ์รับภาพ (Sensor: CCD, CMOS) สูงเพื่อให้ได้ภาพที่มีคุณภาพดีตามไปด้วย

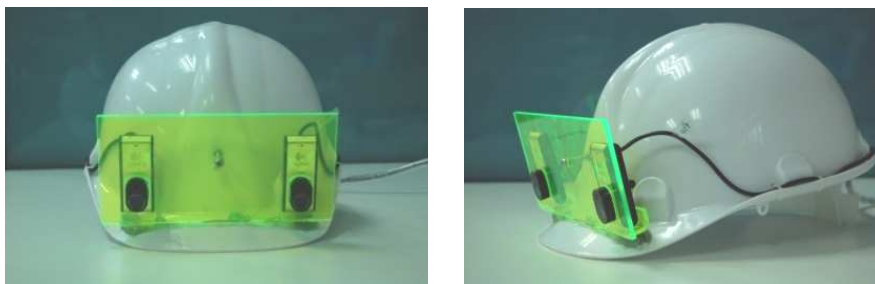
แต่ในวิทยานิพนธ์นี้เลือกประดิษฐ์กล้องสเตอริโอจากเว็บแคมซึ่งเป็นกล้องที่สามารถหาซื้อได้ทั่วไปในท้องตลาด ถึงแม้จะให้คุณภาพของภาพไม่ดีเท่ากล้องสเตอริโอจากโรงงาน แต่มีราคาถูกกว่ามาก โดยเลือก Logitech QuickCam for Notebook Pro [24] ซึ่งสามารถถ่ายภาพวิดีโอขนาด 640x480 พิกเซลได้ อีกทั้งมีขนาดเล็ก น้ำหนักเบา ซึ่งเหมาะแก่การพกพา โดยผู้วิจัยได้ออกแบบโครงสร้างกล้องสเตอริโอในงานวิจัยนี้ดังรูปที่ 3-3 และมีคุณลักษณะของภาพที่ใช้ประมวลผลดังรูปที่ 3-4 เมื่อสร้างกล้องสเตอริโอเรียบร้อยแล้วจะมีลักษณะดังรูปที่ 3-5 และเมื่อได้กล้องสเตอริโอเรียบร้อยแล้วสามารถนำไปสวมใส่จะมีลักษณะดังตัวอย่างในรูปที่ 3-7 นอกจากนี้ยังนำไปประดิษฐ์ในรูปร่างอื่นได้เช่น อุปกรณ์ครอบศีรษะ แวนหรือแม้แต่อุปกรณ์ห้อยคอ ดังรูปที่ 3-6



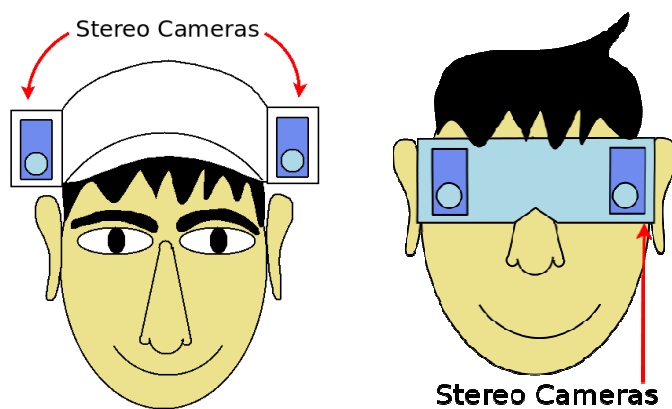
รูปที่ 3-3 โครงสร้างกล้องสเตอริโอที่ใช้สำหรับจับภาพสเตอริโอในวิทยานิพนธ์นี้โดยใช้ระยะห่างระหว่างกล้อง 12 เซนติเมตร



รูปที่ 3-4 คุณลักษณะของภาพที่ใช้ในการประมวลผลที่บันทึกจากกล้องเว็บแคม Logitech



รูปที่ 3-5 โครงสร้างกล้องสเตอริโอที่จัดทำขึ้นเพื่อทดสอบระบบโดยใช้เว็บแคมยี่ห้อ Logitech จำนวน 2 กล้องมีระยะห่างในระนาบแกน x ขนาด 12 เซนติเมตร



รูปที่ 3-6 ตัวอย่างการนำกล้องสเตอริโอไปใช้กับผู้พิการ ซึ่งสามารถประดิษฐ์เป็นแว่นหรือที่ครอบศีรษะก็ได้



รูปที่ 3-7 กล้องสเตอริโอที่จัดทำขึ้นเมื่อนำมาสวมใส่

กล้องสเตอริโอที่สร้างขึ้นมีระยะห่าง 12 เซนติเมตร (base line) เพื่อให้สอดคล้องกับโครงหน้ามนุษย์ซึ่งสามารถนำไปประดิษฐ์เป็นแว่นหรือที่ครอบศีรษะได้ สำหรับระยะ 12 เซนติเมตรนี้เมื่อนำภาพเข้าประมวลผลแล้ว สามารถตรวจจับวัตถุที่มีขนาด 60x60 เซนติเมตรได้ในระยะ 7 เมตร ด้วยขนาดภาพ 320x240 พิกเซลแต่ทั้งนี้จะขึ้นอยู่กับลักษณะของทัศนียภาพด้วย จากรูปที่ 3-4 เมื่อพิจารณาความกว้างของบริเวณที่สามารถบันทึกภาพได้จะเป็นไปตามสมการ (11) โดยที่  $w$  คือความกว้างของภาพ  $d$  คือระยะทางจากกล้องจนถึงบริเวณที่สนใจ

$$w = 2 d \tan 51.34^\circ \quad (11)$$

จากการนำภาพที่ได้จากกล้องสเตอริโอที่ประดิษฐ์ขึ้นใช้ในงานวิจัยดังในรูปที่ 3-5 เมื่อนำมาสวมใส่ดังรูปที่ 3-7 ไปทดสอบกับอัลกอริทึมการประมวลผลภาพสเตอริโอ ปรากฏว่าให้แผนผังคิสพาริตีที่สามารถใช้งานได้ แต่คุณภาพอาจไม่เทียบเท่าคุณภาพกล้องสเตอริโอจากโรงงาน เนื่องจากกล้องเว็บแคมที่ใช้ประดิษฐ์กล้องสเตอริโอแต่ละกล้องอาจมีข้อผิดพลาดทางโครงสร้างที่เกิดจากกระบวนการผลิตที่ไม่ได้คุณภาพมาตรฐานของเว็บแคม ระบายของภาพที่ได้จากกล้องไม่ขนานกันจริง หรืออาจเกิดการเหลื่อมล้ำระหว่างภาพเล็กน้อย คุณภาพของภาพที่ได้จากกล้องมีคุณภาพไม่ดีเท่ากล้องสเตอริโอจากโรงงาน เป็นต้น

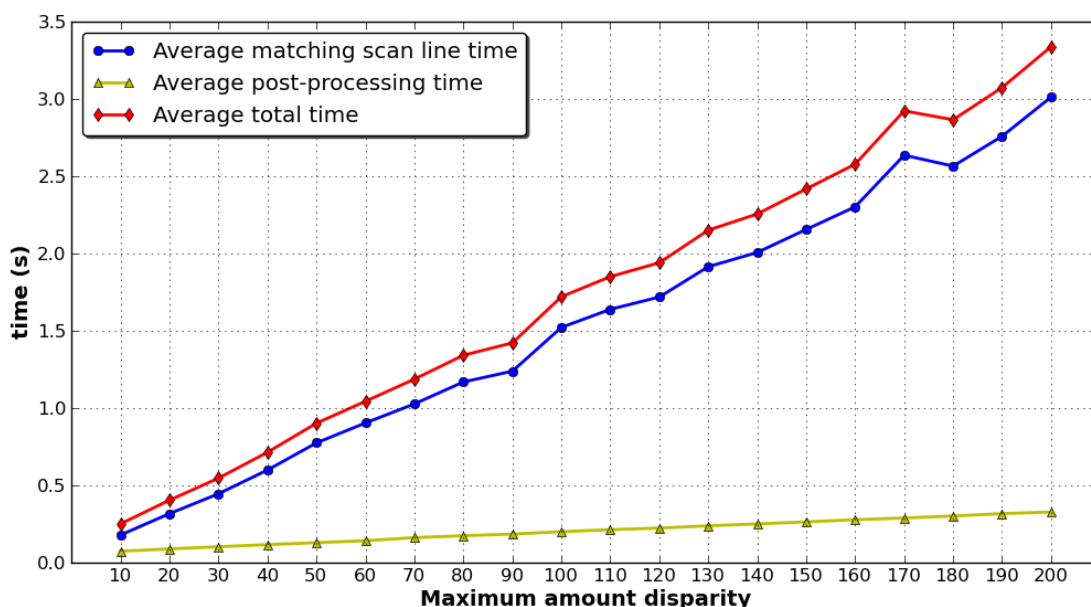
### 3.2.2. การประมวลผลภาพสเตอริโอ

การประมวลผลภาพสเตอริโอสำหรับระบบตรวจสอบสิ่งกีดขวางนี้เน้นเทคนิค intensity-based stereo matching เพื่อแก้ปัญหาที่ปรากฏใน ENVIS [10] ซึ่งไม่สามารถตรวจสอบวัตถุที่ไม่มีลักษณะเด่นได้ เช่น ไวท์บอร์ด ตู้ หรือ กำแพง เป็นต้น เนื่องจาก ENVIS ใช้เทคนิค feature-based stereo matching สำหรับเทคนิค intensity-based stereo matching นั้นมีผู้ทำวิจัยไว้หลายท่านด้วยกัน แต่บางเทคนิคยังไม่เปิดเผยโปรแกรมต้นฉบับจึงไม่อาจนำมาใช้งานกับระบบที่จัดทำขึ้นได้ เพื่อหลีกเลี่ยงปัญหาเรื่องสัญญาอนุญาตจึงใช้งานอัลกอริทึมเฉพาะที่มีการเผยแพร่โปรแกรมต้นฉบับเท่านั้น ซึ่งหนึ่งในนั้นคือ Depth discontinuities by pixel-to-pixel stereo (P2P) โดย S. Birchfield และ C. Tomasi [13], [14] ซึ่งจัดเป็นหนึ่งในอัลกอริทึม ISM ที่ได้รับการยอมรับและปรากฏใน OpenCV ในส่วนของ CvAux ฟังก์ชัน cvFindStereoCorrespondence

จากการทดสอบอัลกอริทึม P2P เพื่อหาพารามิเตอร์ที่เหมาะสมสำหรับการใช้งานระบบนี้ โดยทดสอบหาวัตถุที่มีระยะทาง 1 เมตรเป็นต้นไปจนถึงระยะทางประมาณ 7 เมตรปรากฏว่าระดับคิสพาริตีที่เหมาะสมแก่การใช้งานคือตั้งแต่ 50 ถึง 100 สามารถดูภาพตัวอย่างจากการทดสอบได้

ที่ภาคผนวก ข สำหรับพารามิเตอร์อื่นๆ ที่ใช้งานในระบบคือ Occlusion penalty, Match reward และ Reliability threshold จะใช้ค่าระบุไว้ในงานวิจัยผู้คิดค้นอัลกอริทึม [13] นี้คือ 25, 5 และ 14 ตามลำดับ เนื่องจากเมื่อทดสอบภาพด้วยพารามิเตอร์เหล่านี้ที่เปลี่ยนไป ไม่ส่งผลให้เกิดการเปลี่ยนแปลงคิสพาริตีที่ชัดเจนจึงใช้ค่าที่ผู้คิดค้นเสนอแนะแทน

ถึงแม้ว่าอัลกอริทึม Depth discontinuities by pixel-to-pixel stereo จะสามารถหาแผนผังคิสพาริตีได้รวดเร็วเมื่อเทียบกับอัลกอริทึมในหมวดหมู่เดียวกัน (ISM) แต่ผลตอบสนองที่ได้จากอัลกอริทึม P2P ยังไม่เร็วเพียงพอที่จะใช้งานในระบบตรวจสอบสิ่งกีดขวางสำหรับผู้พิการทางสายตา ซึ่งความเร็วในการตรวจจับสิ่งกีดขวางเป็นตัวแปรสำคัญที่ทำให้ผู้พิการปลอดภัยจากอุปสรรคที่กีดขวางอยู่



รูปที่ 3-8 กราฟแสดงความสัมพันธ์ระหว่างเวลาที่ใช้ในประมวลผลกับจำนวนระดับคิสพาริตีสูงสุดของ P2P โดยใช้ซอร์สโค้ดแบบตามลำดับซึ่งทำงานอยู่บนคอมพิวเตอร์ที่มีหน่วยประมวลผลแบบสองแกน ภาพขนาด 320x240 พิกเซล

จากการทดสอบอัลกอริทึม P2P บนคอมพิวเตอร์ที่ใช้ CPU ตระกูล Intel® Core™ 2 Dual ซึ่งเป็น CPU ที่สามารถหาซื้อได้ทั่วไป รุ่น Intel® Core™ 2 Dual 6320 1.86 GHz RAM 1010.7 MB ใช้ Linux kernel 2.6.26 เป็นระบบปฏิบัติการ ที่จำนวนคิสพาริตีสูงสุด 100 และขนาดรูปที่ 320x240 พิกเซล เวลาในการประมวลผลต่อหนึ่งเฟรมโดยเฉลี่ยอยู่ที่ 1.719 วินาทีต่อภาพ โดยแบ่งเป็นการ

จับคู่พิกเซลในแต่ละแถว 1.520 วินาทีและการประมวลผลขั้นปลาย 0.199 วินาที นอกจากนี้สามารถดูรายละเอียดกราฟแสดงความสัมพันธ์ระหว่างเวลาที่ใช้ในการประมวลผลกับจำนวนระดับดิสพาริตีสูงสุดที่กำหนดให้อัลกอริทึม P2P ได้จากรูปที่ 3-8

จากรูปที่ 3-8 หากพิจารณาจะพบว่าเมื่อเพิ่มจำนวนระดับดิสพาริตีสูงสุดในขณะที่ขนาดภาพไม่เปลี่ยนแปลง เวลาในการประมวลผลจะเพิ่มตามไปด้วย ส่วนของการจับคู่พิกเซลในแต่ละแถวจะใช้เวลาในการประมวลผลส่วนใหญ่ของเวลาทั้งหมดในขณะที่การประมวลผลขั้นปลายใช้เวลาในการประมวลผลน้อยกว่ามาก กระบวนการจับคู่พิกเซลในแต่ละแถวจึงเป็นส่วนที่น่าสนใจศึกษาเพื่อเพิ่มความเร็ว อีกทั้งลักษณะการประมวลผลของกระบวนการนี้เอื้อต่อการประมวลผลแบบขนาน โดยอธิบายวิธีการแบ่งงานเพื่อออกแบบการประมวลผลแบบขนานไว้ในหัวข้อ 3.3 การออกแบบอัลกอริทึมการประมวลผลแบบขนาน ส่วนหัวข้อ 3.3.1 เน้นการออกแบบการประมวลผลแบบขนาน P2P ในส่วน การจับคู่พิกเซลในแต่ละแถว สำหรับกระบวนการประมวลผลขั้นปลายนั้นจากการพิจารณากราฟจากรูปที่ 3-8 แล้วพบว่ากราฟของเวลาที่ใช้ประมวลผลมีความชันน้อยมากแต่สามารถเพิ่มความเร็วด้วยการประมวลผลแบบขนานได้เช่นกัน และอาจจะมีวิธีในการแบ่งงานที่ซับซ้อนมากกว่า เนื่องจากในกระบวนการทำงานมีการแลกเปลี่ยนข้อมูลจากการประมวลผลทั้งข้อมูลระหว่างแถว และข้อมูลระหว่างคอลัมน์ดังอธิบายไว้ใน การออกแบบอัลกอริทึมการประมวลผลแบบขนานในหัวข้อที่ 3.3.2 การออกแบบการประมวลผลแบบขนาน P2P ส่วนการประมวลผลขั้นปลาย

### 3.2.3. การประมาณค่าระยะทางสิ่งกีดขวาง

วิทยานิพนธ์นี้สนใจการประมาณค่าระยะทางสิ่งกีดขวางสำหรับการประมวลผลภาพสเตอริโอโดยใช้ U และ V-disparity ซึ่งเป็นข้อมูลที่ได้จากการแปลงแผนผังดิสพาริตีให้อยู่ในรูปพิกัด u และ v เมื่อได้ U และ V-disparity แล้วอุปสรรคหรือสิ่งกีดขวางจะปรากฏออกมาอยู่ในรูปเส้นตรงในแนวนอนสำหรับ U-disparity และในแนวตั้งสำหรับ V-disparity ซึ่งสามารถนำค่าตามแนวแกน y และค่าตามแนวแกน x ของเส้นตรงนั้นมาแทนค่าในสมการประมาณค่าระยะทาง เพื่อหาระยะทางของวัตถุได้ โดยระยะทางที่ได้จากสมการจะเป็นระยะทางจากสิ่งกีดขวางจนถึงตัวผู้พิจารณา

สมการประมาณค่าระยะทางเป็นการประมาณค่าในช่วงโดยใช้สมการพหุนามประมาณค่าระยะทางจริงที่สอดคล้องกับค่าดิสพาริตีในแต่ละช่วง โดยค่าระยะทางจริงที่ใช้ในการประมาณค่านี้อยู่ในช่วง 1 ถึง 6 เมตร เหตุที่ใช้ระยะทางช่วงนี้เนื่องจากระยะทางตั้งแต่ตัวผู้พิจารณาจนถึงหนึ่งเมตรเป็นระยะทางที่ผู้พิจารณาทางสายตาสามารถใช้ไม่เท่าตรวจจับสิ่งกีดขวางได้ ส่วนเกินจากระยะ 6



เมตรเป็นต้นไป เมื่อนำภาพที่ได้จากกล้องสเตอริโอเข้าอัลกอริทึมประมวลผลภาพสเตอริโอ P2P แล้วนำมาแปลงเป็น V-disparity จะแยกสิ่งที่เป็นอุปสรรคสิ่งกีดขวางกับสิ่งรอบกวนได้ยาก

ตำแหน่งของวัตถุที่ปรากฏอยู่ในภาพนั้นสามารถอธิบายได้ด้วย U-disparity โดยที่ตำแหน่งของอุปสรรคและสิ่งกีดขวางจะปรากฏอยู่ในรูปของเส้นตรงหรือเส้นโค้งใน U-disparity หากนำตำแหน่งของเส้นตรงใน U-disparity มาแบ่งออกเป็นส่วนๆ ตามแนวตั้ง จะสามารถระบุตำแหน่งของวัตถุอย่างคร่าวๆ ได้

### 3.3. การออกแบบอัลกอริทึมการประมวลผลแบบขนาน

เนื่องจากอัลกอริทึมประมวลผลภาพสเตอริโอ P2P ที่ใช้ในงานวิจัยนี้ ถึงแม้จะสามารถตรวจจับสิ่งกีดขวางได้ดี แต่ความเร็วของผลตอบสนองจากอัลกอริทึมช้ามาก จากหัวข้อการประมวลผลภาพสเตอริโอ เวลาที่ใช้ในการประมวลผลอัลกอริทึม P2P โดยเฉลี่ยอยู่ที่ 1.719 วินาทีต่อภาพซึ่งการตอบสนองที่ได้ไม่สามารถใช้ได้ในเวลาจริง หากนำอัลกอริทึม P2P มาให้ผู้พิจารณาทางสายตาใช้งานอาจส่งผลให้ผู้พิจารณาได้รับอันตรายหรือได้ข้อมูลสภาพแวดล้อมคลาดเคลื่อนจากความเป็นจริงได้

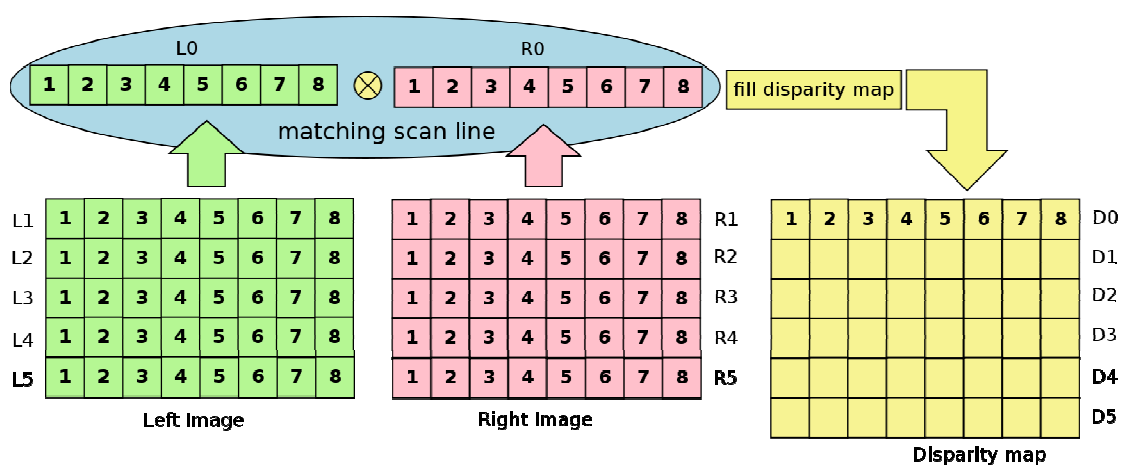
ในปัจจุบันเทคโนโลยีหน่วยประมวลผลกลางได้พัฒนาให้หนึ่งหน่วย CPU มีหลายแกน (multi-core CPU) อีกทั้งยังใช้ CPU ชนิดนี้กันแพร่หลายทั้งคอมพิวเตอร์ตั้งโต๊ะและแบบพกพา การใช้งาน CPU เหล่านี้ให้เต็มประสิทธิภาพต้องอาศัยการประมวลผลแบบขนาน ซึ่งเทคนิคนี้ต้องการการออกแบบลักษณะงานที่เหมาะสมด้วยจึงจะสามารถใช้งานได้ดีและให้ประสิทธิภาพสูงสุด การประมวลผลแบบขนานนี้เป็นเทคนิคที่เหมาะสมต่อการลดเวลาการประมวลผลภาพ เนื่องจากข้อมูลภาพมีลักษณะเป็นเมตริกซ์สองมิติ การประมวลผลภาพส่วนใหญ่จะเป็นการดำเนินการในแต่ละพิกเซลหรือกลุ่มของพิกเซลในภาพ ซึ่งบางเทคนิคสามารถแบ่งการทำงานออกเป็นอิสระจากกันในแต่ละแถวของภาพหรือในแต่ละคอลัมน์ของภาพได้

อัลกอริทึม P2P แยกกระบวนการทำงานออกเป็นสองส่วนหลัก คือ การจับคู่พิกเซลในแต่ละแถว และการประมวลผลขั้นปลาย ซึ่งมีลักษณะการทำงานที่แตกต่างกัน ผู้วิจัยจึงได้ออกแบบโปรแกรมแบบขนานให้เหมาะสมและสอดคล้องกับลักษณะการประมวลผลในแต่ละกระบวนการดังต่อไปนี้

#### 3.3.1. การออกแบบการประมวลผลแบบขนาน P2P ส่วนการจับคู่พิกเซลในแต่ละแถว

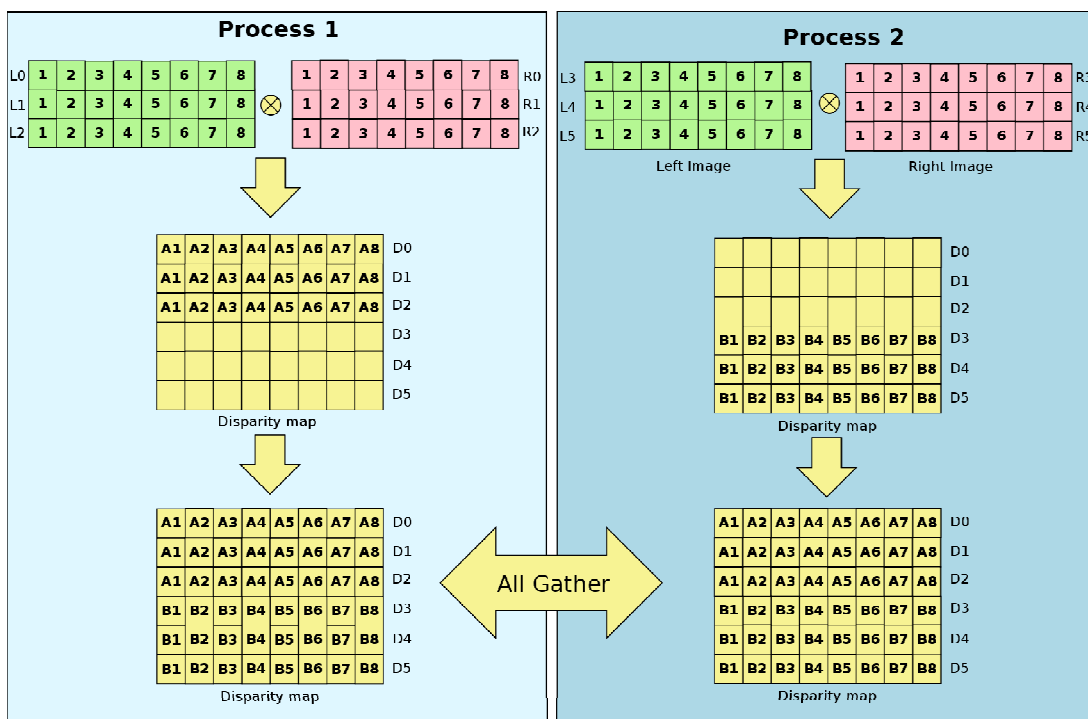
กระบวนการจับคู่พิกเซลในแต่ละแถวเป็นกระบวนการจับคู่พิกเซล โดยใช้เทคนิคการกำหนดการพลวัตเพื่อหาพิกเซลที่เหมาะสมที่สุดระหว่างเส้นกราดตรวจจากภาพซ้ายและภาพขวา

เพื่อให้ได้มาซึ่งแผนผังคิสพาริตีที่ดีที่สุด ตัวอย่างการทำงานของกระบวนการนี้แสดงในรูปที่ 3-9 จากรูปที่ 3-9 กระบวนการนี้จะใช้ข้อมูลจากภาพทั้งสองภาพในแถวเดียวกันเพื่อประมวลผล โดยข้อมูลในแต่ละแถวเป็นอิสระจากกันซึ่งการทำงานในลักษณะนี้เหมาะแก่การจัดการให้อัลกอริทึมสามารถประมวลผลแบบขนานได้ ในลักษณะการประมวลผลแบบขนานเชิงข้อมูล คือลักษณะการประมวลผลจะเหมือนกันทุกอย่างในทุกๆ โปรเซสแต่แตกต่างกันเฉพาะข้อมูลที่ใช้ประมวลผลเท่านั้น

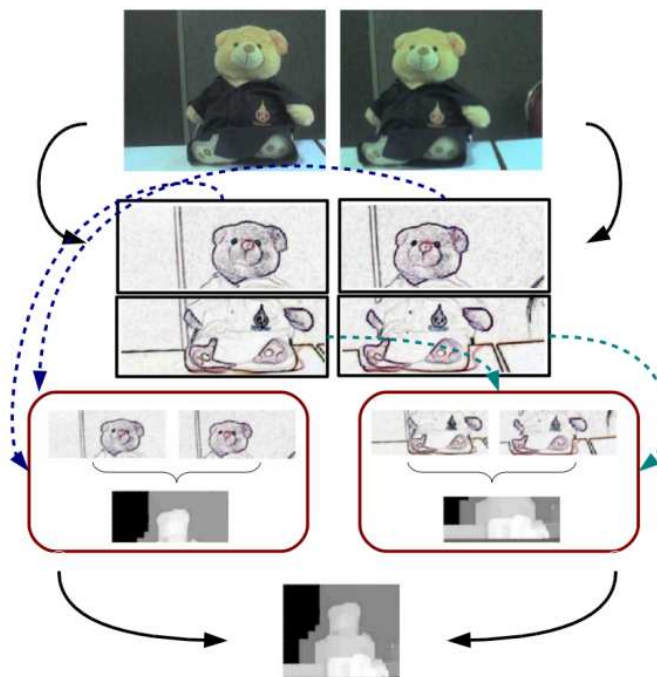


รูปที่ 3-9 การทำงานโดยรวมของกระบวนการจับคู่พิกเซลในแต่ละแถว

ดังนั้นสำหรับกระบวนการนี้วิธีแบ่งข้อมูลที่ดีที่สุดคือการแบ่งจำนวนในแต่ละแถวข้อมูล โดยแบ่งออกเป็นกลุ่มของแถวกลุ่มละเท่า ๆ กันเพื่อประสิทธิภาพสำหรับรวบรวมข้อมูลกลับที่ดี ตัวอย่างการแบ่งงานตามรูปที่ 3-10 ซึ่งเป็นตัวอย่างการแบ่งงานออกเป็นสองโปรเซส โดยที่โปรเซสแรกทำหน้าที่จับคู่ความสัมพันธ์ของพิกเซลที่เหมาะสมสำหรับกลุ่มของแถวของส่วนบนของภาพ และโปรเซสที่สองทำหน้าที่ในจำนวนส่วนล่างของภาพ หลังจากหาแผนผังคิสพาริตีเสร็จสิ้นเรียบร้อยแล้ว โปรเซสแรกจะมีข้อมูลแผนผังคิสพาริตีส่วนครึ่งบน และโปรเซสที่สองจะมีข้อมูลแผนผังคิสพาริตีส่วนครึ่งล่าง จากนั้นจึงใช้การรวบรวมข้อมูลกลับแล้วกระจายไปยังทุกโปรเซสเพื่อเตรียมนำแผนผังคิสพาริตีที่รวบรวมไว้อย่างสมบูรณ์แล้วไปคำนวณต่อในกระบวนการถัดไป จากวิธีการดังกล่าวเมื่อนำมาใช้งานจริงจะมีการแบ่งงานในการคำนวณดังรูปที่ 3-11



รูปที่ 3-10 การแบ่งงานออกเป็นสองโปรเซสให้ประมวลผลพร้อมกันเพื่อลดเวลาของผลตอบสนองในส่วนการจับคู่พิกเซลในแต่ละแถว



รูปที่ 3-11 ตัวอย่างการแบ่งงานในส่วนการจับคู่พิกเซลในแต่ละแถว

```

ROWS : number of rows in the image
id : process id
numProcs : number of processes
startIndex := ROWS*id/ numProcs;           //starting scan line
endIndex := ROWS*(id+1)/numProcs;         //finishing scan line

//scatter left and right images at Process 0 to other processes from
startIndex to ROWS/ numProcs
MPI::Broadcast(imgL, ...);
MPI::Broadcast(imgR, ...);

for (scanline = startIndex; scanline < endIndex; scanline++)
    matching scanline independently

// collect disparity map to all process
MPI::Allgather(disparity_map,...);
// collect depth discontinuities images from all the processes into
Process 0
MPI::Gather(depth_discontinuities,...);

```

### รูปที่ 3-12 ส่วนของโปรแกรมแสดงลักษณะการแบ่งงาน

จากการออกแบบ โปรแกรมในรูปที่ 3-10 สามารถเขียนเป็นโปรแกรมได้ดังรูปที่ 3-12 โดยมีลำดับขั้นตอนดังต่อไปนี้

- 1) ให้ค่าเริ่มต้นแก่ตัวแปรที่จำเป็นในการประมวลผลดังต่อไปนี้
  - ROWS เป็นจำนวนแถวของภาพ ซึ่งใช้เป็นตัวแปรสำหรับการแบ่งงานให้แก่แต่ละโปรเซส
  - id คือหมายเลขที่ใช้ระบุโปรเซส โดยเป็นเลขจำนวนเต็มค่าหนึ่งเริ่มจาก 0 เป็นต้นไป สามารถอ่านค่าได้จาก MPI::COMM\_WORLD.Get\_rank()
  - numProcs จำนวนโปรเซสทั้งหมดที่ใช้สำหรับคำนวณ
  - startIndex เป็นหมายเลขแถวรูปภาพเริ่มต้นที่โปรเซสนั้นๆ ใช้เริ่มการคำนวณ โดยค่านี้มาจาก  $ROWS * id / numProcs$
  - endIndex เป็นหมายเลขแถวรูปภาพหลังแถวสุดท้ายที่โปรเซสนั้นๆ จะประมวลผล โดยค่านี้มาจาก  $ROWS * (id + 1) / numProcs$
- 2) เมื่อได้ค่าเริ่มต้นสำหรับตัวแปรที่จำเป็นแล้ว ขั้นตอนต่อไปเป็นการกระจายข้อมูลที่จำเป็นสำหรับการประมวลผลไปยังโปรเซสต่างๆ ซึ่งในที่นี้คือข้อมูลรูปภาพที่บันทึกจากกล้องซ้ายและขวา โดยปรกติเราควรแบ่งข้อมูลออกเป็นส่วนๆ เพื่อลดปริมาณข้อมูลที่ใช้สร้างข้อความเพื่อการติดต่อระหว่างโปรเซส แต่เนื่องจากข้อมูลรูปภาพทั้งภาพจะถูกนำมาใช้งานต่อในการประมวลผลของการประมวลผลขั้นปลายหากใช้วิธีการแบ่งส่วนข้อมูลด้วย

คำสั่ง `MPI::Scatter` ในกระบวนการนี้แล้วยังคงต้องส่งข้อมูลทั้งหมดใหม่ในกระบวนการประมวลผลขั้นปลายอีกทำให้โปรแกรมใช้เวลาในการรับส่งข้อมูลมากขึ้นโดยไม่จำเป็น ดังนั้นในกรณีนี้ผู้พัฒนาจึงเลือกส่งข้อมูลทั้งหมดไปทีเดียวโดยใช้คำสั่ง `MPI::Bcast`

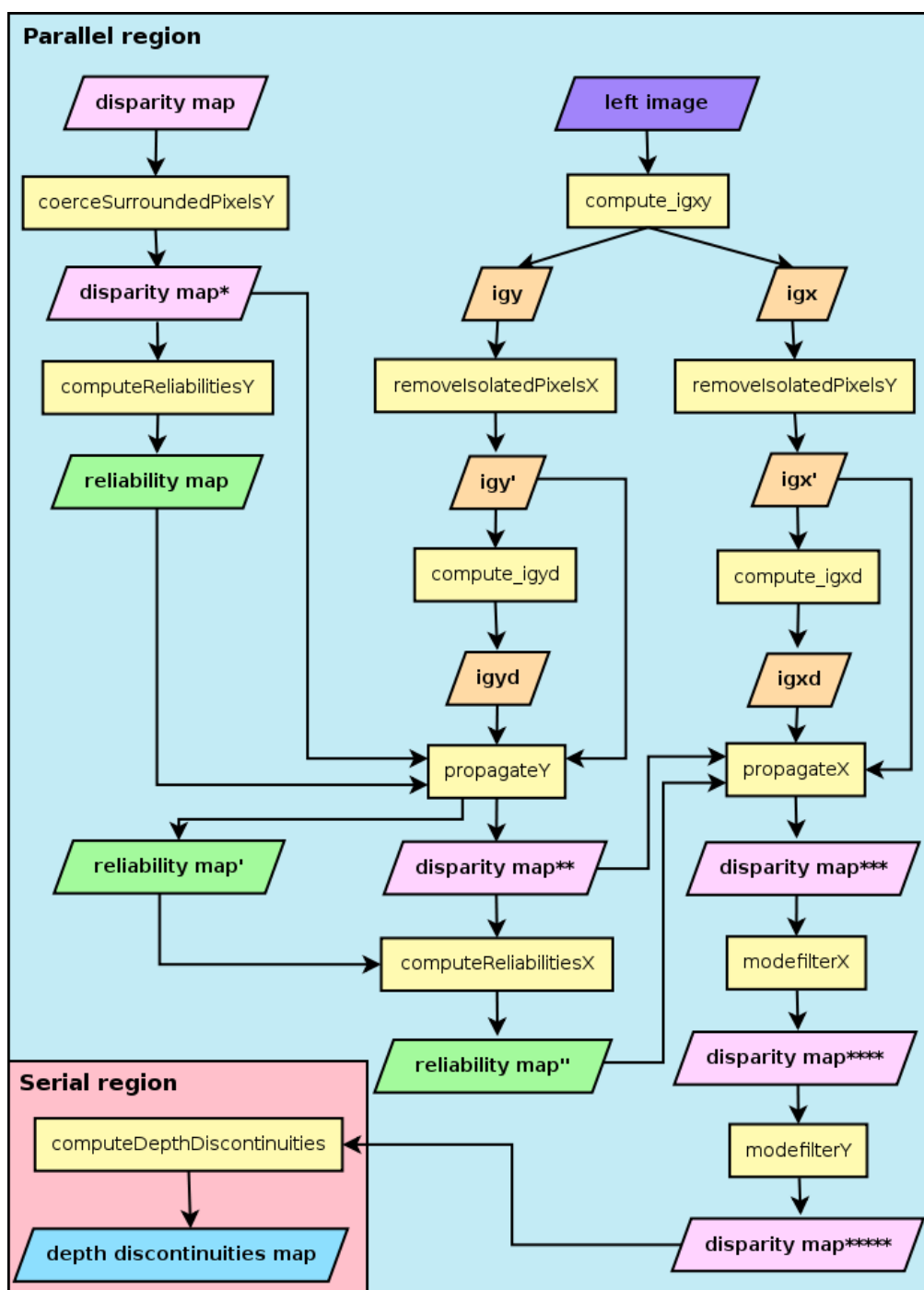
- 3) ต่อมาคือการประมวลผลกระบวนการจับคู่พิกเซลในแต่ละแถว จากที่กล่าวไปแล้วในข้างต้นคือกระบวนการนี้แต่ละโปรเซสจะแยกกันประมวลผลตามดัชนีที่กำหนดค่าไว้ในตอนเริ่มต้น โดยเริ่มตั้งแต่แถวที่มีดัชนีตรงกับ `startIndex` แล้วสิ้นสุดแถวที่มีดัชนีก่อนหน้า `endIndex` ผลลัพธ์ที่ได้จากกระบวนการนี้มีคือแผนผังคิสพาริตีซึ่งจะถูกเก็บข้อมูลไว้ในแต่ละโปรเซส
- 4) ขั้นตอนสุดท้ายคือการรวบรวมข้อมูลในแต่ละโปรเซสเพื่อใช้เป็นข้อมูลสำหรับการประมวลผลในส่วนของการประมวลผลขั้นปลายต่อไป สำหรับผลลัพธ์ที่ได้จากกระบวนการจับคู่พิกเซลในแต่ละแถวคือแผนผังคิสพาริตีและ `depth discontinuities` แต่ทั้งสองอย่างนี้มีลักษณะการใช้งานที่แตกต่างกันแผนผังคิสพาริตีจะถูกนำไปใช้งานต่อในกระบวนการประมวลผลขั้นปลาย โดยที่ทุกๆ โปรเซสต้องใช้งานทุกๆ ข้อมูลในแผนผังคิสพาริตี ดังนั้น `MPI::Allgather` จึงเหมาะสมกับการกระจายข้อมูลในขั้นตอนนี้ เนื่องจาก `MPI::Allgather` จะรวบรวมข้อมูลจากทุกๆ โปรเซสแล้วกระจายข้อมูลที่ได้รับไปยังทุกๆ โปรเซสด้วยเช่นกัน สำหรับ `depth discontinuities` นั้นเป็นผลลัพธ์ที่ได้จากกระบวนการนี้ แต่ `depth discontinuities` จะไม่ถูกนำไปใช้งานต่อในกระบวนการประมวลผลขั้นปลาย ดังนั้นจึงใช้เพียง `MPI::Gather` เพื่อรวมข้อมูลไว้ที่โปรเซสหลักเพียงโปรเซสเดียวเท่านั้น

### 3.3.2. การออกแบบการประมวลผลแบบขนาน P2P ส่วนการประมวลผลขั้นปลาย

กระบวนการประมวลผลขั้นปลายนั้นมีความซับซ้อนในการแบ่งงานมากกว่ากระบวนการจับคู่พิกเซลในแต่ละแถว แต่ใช้เวลาประมวลผลคิดเป็นร้อยละ 12.38 ของเวลาการประมวลผลทั้งหมดของอัลกอริทึม P2P จากรูปที่ 3-8 จะเห็นได้ว่ากระบวนการประมวลผลขั้นปลายนั้นใช้เวลาน้อยมาก เมื่อเทียบกับกระบวนการจับคู่พิกเซลในแต่ละแถว แต่เนื่องจากกว่าร้อยละ 90 ของกระบวนการนี้สามารถจัดสรรงานให้ประมวลผลแบบขนานได้ เพื่อช่วยลดเวลารวมในการประมวลผลทั้งหมดจึงได้เปลี่ยนกระบวนการประมวลผลขั้นปลายให้เป็นแบบขนานด้วยเช่นกัน

กระบวนการประมวลผลขั้นปลายมีฟังก์ชันที่ใช้งานทั้งหมด 13 ฟังก์ชัน ซึ่งทั้ง 13 ฟังก์ชันมีลำดับการทำงานและการไหลของข้อมูลดังแสดงในรูปที่ 3-13 กระบวนการประมวลผลขั้นปลายนี้ในแต่ละฟังก์ชันจะมีการปรับปรุงข้อมูลเดิมหรือสร้างข้อมูลใหม่เพื่อใช้ในการคำนวณหลายชุดข้อมูลด้วยกัน โดยเริ่มจากการนำข้อมูลแผนผังคิสพาริตีจากกระบวนการจับคู่พิกเซลในแต่ละแถว

มาเตรียมข้อมูลตามแนวแกน y ด้วยฟังก์ชัน `coerceSurroundedPixelsY` จะได้แผนผังคิสพาริตีขั้นแรก (disparity map\*) จากนั้นนำแผนผังคิสพาริตีที่ได้จากฟังก์ชันนี้ไปหาความน่าเชื่อถือของแต่ละระดับคิสพาริตีที่จะปรากฏในภาพบริเวณนั้นๆ ตามแนวแกน y ด้วยฟังก์ชัน `computeReliabilitiesY` จากฟังก์ชันนี้จะได้ข้อมูลหนึ่งชุดก็คือ reliability map เพื่อเตรียมพร้อมใช้คำนวณในฟังก์ชันต่อไป

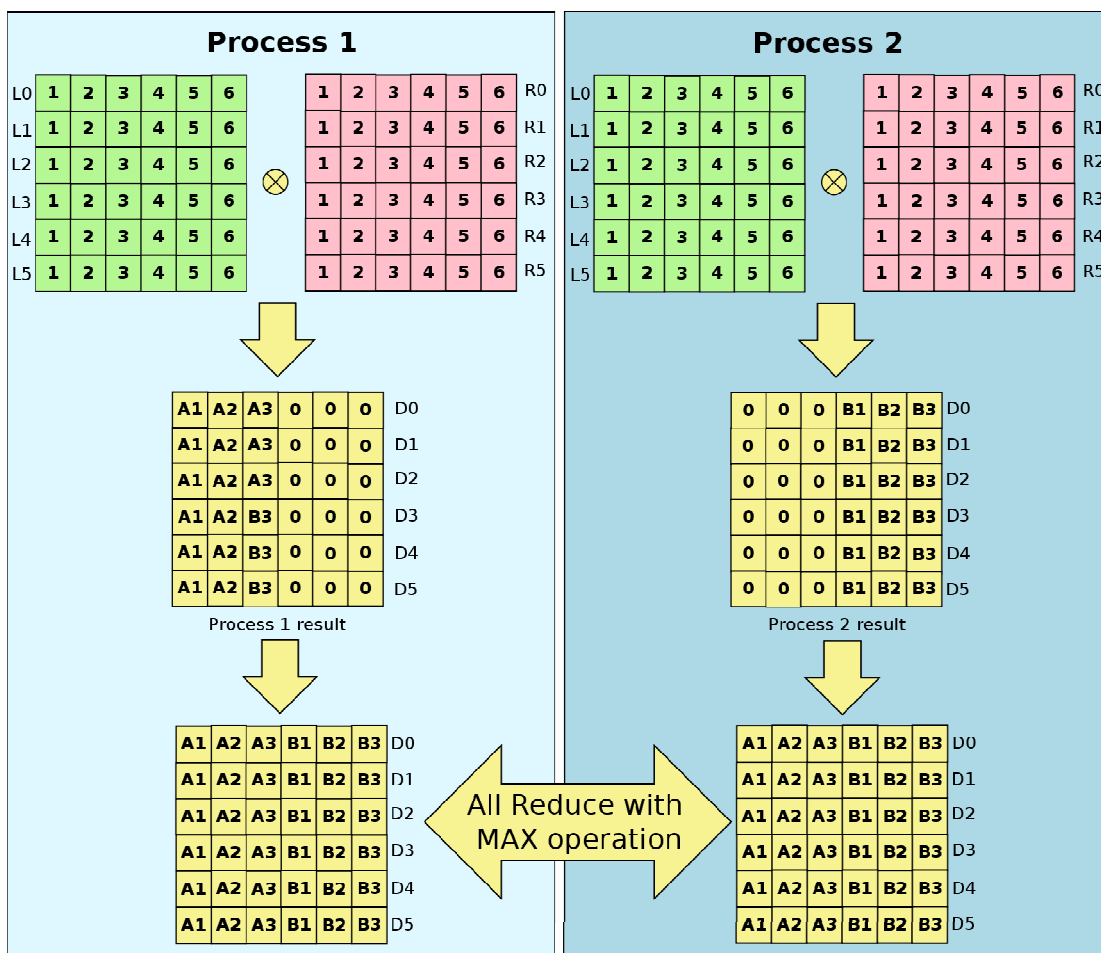


รูปที่ 3-13 ขั้นตอนการทำงานและการไหลของข้อมูลในกระบวนการประมวลผลขั้นปลาย

สำหรับการประมวลผลอีกชุดหนึ่งเพื่อเตรียมข้อมูลสำหรับการประมวลผลในลำดับต่อไป เริ่มจากการนำข้อมูลภาพด้านซ้ายเข้าสู่ฟังก์ชัน `compute_igxy` เพื่อหาเกรเดียนต์ความเข้มตามแนวแกน x และแนวแกน y ซึ่งผลที่ได้ในฟังก์ชันนี้จะเป็น `igy` และ `igx` หลังจากนั้นนำ `igy` มาจัดข้อมูลหรือกลุ่มของข้อมูลที่มีจำนวนไม่ครบตามที่กำหนดในแนวแกน x ด้วยฟังก์ชัน `removeIsolatedPixelsX` ซึ่งผลที่ได้จากฟังก์ชันนี้คือข้อมูล `igy'` จากนั้นนำข้อมูลนี้ไปหาแผนผังจำนวนเต็มที่ใช้ระบุตำแหน่งระยะทางของเกรเดียนต์ความเข้ม โดยที่ 0 แทนเกรเดียนต์ความเข้ม ณ ตำแหน่งดังกล่าว ค่าบวกแทนระยะทางที่อยู่ด้านล่างของเกรเดียนต์ความเข้ม ค่าลบแทนระยะทางด้านบนของเกรเดียนต์ความเข้มในฟังก์ชัน `compute_igyd` ผลที่ได้จากฟังก์ชันนี้คือ `igyd` สำหรับข้อมูล `igx` จะมีการคำนวณในลักษณะเดียวกันกับ `igy` แต่เปลี่ยนเป็นในแนวแกน x แทน โดยผ่านฟังก์ชัน `removeIsolatedPixelsY` จะได้ข้อมูล `igx'` แล้วนำข้อมูลดังกล่าวมาคำนวณด้วยฟังก์ชัน `compute_igxd` ได้ข้อมูล `igxd` โดยที่ 0 แทนเกรเดียนต์ความเข้ม ณ ตำแหน่งดังกล่าว ค่าบวกแทนระยะทางที่อยู่ด้านขวาของเกรเดียนต์ความเข้ม ค่าลบแทนระยะทางด้านซ้ายของเกรเดียนต์ความเข้ม ข้อมูลเกรเดียนต์ความเข้มที่ได้จากฟังก์ชันเหล่านี้ จะถูกนำไปปรับปรุงแผนผังดิสพาริตีต่อไป

ในลำดับถัดไปเป็นการคำนวณข้อมูลที่ได้จากข้อมูลที่สร้างไว้ โดยเริ่มที่ฟังก์ชัน `propagateY` ใช้ข้อมูล `disparity map*` ข้อมูล `reliability map` ข้อมูล `igy'` ข้อมูล `igyd` เพื่อปรับปรุงข้อมูลแผนผังดิสพาริตีในแนวแกน y โดยข้อมูลที่ได้จากฟังก์ชันนี้คือ `disparity map**` และ `reliability map'` เมื่อได้ข้อมูลในส่วนนี้แล้ว นำแผนผังดิสพาริตี (`disparity map**`) และ `reliability map'` ที่ได้จากฟังก์ชัน `propagateY` ไปหาความน่าเชื่อถือของแต่ละระดับดิสพาริตีที่จะปรากฏในบริเวณพิกเซลนั้นๆ ตามแนวแกน x ด้วยฟังก์ชัน `computeReliabilitiesX` ผลที่ได้จากฟังก์ชันนี้คือ `reliability map''` เพื่อนำไปใช้คำนวณต่อไป ในส่วนการแลกเปลี่ยนข้อมูลในแนวแกน x ด้วยฟังก์ชัน `propagateX` ซึ่งฟังก์ชันนี้จะใช้ข้อมูลข้อมูล `disparity map**` ข้อมูล `reliability map'` ข้อมูล `igx'` ข้อมูล `igxd` เพื่อปรับปรุงข้อมูลแผนผังดิสพาริตีในแนวแกน x ผลที่ได้จากฟังก์ชันนี้คือ `disparity map***` ส่วนสุดท้ายเป็นการกรองข้อมูลระดับดิสพาริตีที่มีค่าเกินกว่าจำนวนดิสพาริตีสูงสุดทั้งแนวแกน x และ y โดยใช้ฟังก์ชัน `modefilterX` และ `modefilterY` ตามลำดับ ฟังก์ชันทั้งสองให้ผลออกมาเป็นข้อมูล `disparity map****` และ `disparity map*****` ซึ่ง `disparity map*****` นี้เป็นแผนผังดิสพาริตีในลำดับสุดท้ายที่จะนำไปใช้งานในส่วนตรวจสอบตำแหน่งและประมาณระยะทางวัตถุต่อไป ทั้งหมดที่กล่าวมานี้เป็นการปรับปรุงคุณภาพของแผนผังดิสพาริตีในส่วนที่สามารถปรับปรุงให้ประมวลผลแบบขนานได้ สำหรับส่วนที่ไม่สามารถปรับปรุงเป็นการประมวลผลแบบขนานได้คือ `computeDepthDiscontinuities` เนื่องจากเป็นการนำแผนผังดิสพาริตี

ในขั้นสุดท้าย (disparity map\*\*\*\*) มาใช้คำนวณแผนผัง depth discontinuities ที่ต้องพิจารณาข้อมูลรอบคอบที่นั่นๆ ทั้งด้านบน ด้านล่าง ด้านซ้าย และด้านขวาในคราวเดียวกัน



รูปที่ 3-14 การแบ่งงานออกเป็นสองโปรเซสให้ประมวลผลพร้อมกันเพื่อลดเวลาของผลตอบสนองในส่วนการประมวลผลขั้นปลายในการคำนวณที่ใช้ข้อมูลทั้งคอลัมน์

การประมวลผลขั้นปลายทั้ง 13 ฟังก์ชันนั้นสามารถเขียนโปรแกรมเป็นแบบขนานได้ทั้งหมด 12 ฟังก์ชัน โดยที่ทั้ง 13 ฟังก์ชันสามารถแบ่งออกเป็นสี่กลุ่มตามลักษณะการคำนวณได้ดังต่อไปนี้

- 1) คำนวณโดยใช้ข้อมูลทั้งแถว
  - modefilterX
  - compute\_igxd



- computeReliabilitiesX
  - propagateX
  - removeIsolatedPixelsX
- 2) คำนวณ โดยใช้ข้อมูลทั้งคอลัมน์
- coerceSurroundedPixelsY
  - modefilterY
  - compute\_igyd
  - computeReliabilitiesY
  - propagateY
  - removeIsolatedPixelsY
- 3) คำนวณ โดยใช้ข้อมูลทั้งแถวและคอลัมน์
- compute\_igxy
- 4) ส่วนที่ไม่สามารถเขียนโปรแกรมแบบขนานได้
- computeDepthDiscontinuities

ลักษณะการคำนวณในกระบวนการประมวลผลขั้นปลายที่วิทยานิพนธ์นี้สนใจนำมาปรับปรุงให้สามารถคำนวณแบบขนานได้คือการใช้ข้อมูลทั้งแถวและการใช้ข้อมูลทั้งคอลัมน์ เนื่องจากลักษณะการประมวลผลในทั้งสองกลุ่มดังกล่าวจะใช้ข้อมูลในการคำนวณจากแถวแต่ละแถวและคอลัมน์แต่ละคอลัมน์เป็นอิสระจากกัน กระทำการคำนวณเป็นลำดับโดยในแต่ละฟังก์ชันจะสร้างข้อมูลเพื่อใช้ในการประมวลผลในฟังก์ชันต่อไป ลักษณะดังกล่าวจึงเหมาะแก่การแบ่งงานในลักษณะการประมวลผลแบบขนานเชิงข้อมูล สำหรับส่วนการแบ่งงานในการคำนวณโดยใช้ข้อมูลทั้งแถวนั้นมีลักษณะเดียวกับกระบวนการจับคู่พิกเซลในแต่ละแถว ซึ่งมีหลักการปรับเปลี่ยนเป็นโปรแกรมแบบขนานคล้ายกันจึงไม่ขอกว่าในที่นี้ อีก ส่วนการคำนวณ โดยใช้ข้อมูลทั้งคอลัมน์ นั้น จะมีลักษณะคล้ายกับการคำนวณ โดยใช้ข้อมูลทั้งแถว แต่จะไล่ประมวลผลไปที่ละคอลัมน์ แทนที่จะเป็นทีละแถว จาก

รูปที่ 3-14 เป็นตัวอย่างการแบ่งงานการคำนวณตามคอลัมน์ ซึ่งลักษณะการคำนวณจะเหมือนกับรูปที่ 3-10 แต่เปลี่ยนจากการคำนวณ โดยใช้ข้อมูลทั้งแถวเป็นการคำนวณ โดยใช้ข้อมูลทั้งคอลัมน์แทน แต่ MPI ไม่มีฟังก์ชันที่สนับสนุนการรวมข้อมูลกลับจากคอลัมน์ในลักษณะนี้ โดยเฉพาะ จึงใช้ฟังก์ชัน MPI::Allreduce เข้าช่วยรวมข้อมูลกลับแทน โดยเลือกตัวดำเนินการให้เหมาะสมกับลักษณะการประมวลผลในแต่ละฟังก์ชันของกระบวนการประมวลผลขั้นปลาย จากการวิเคราะห์ตัว

ดำเนินการที่เหมาะสมในการใช้งานมีทั้งหมดสามชนิดด้วยกันคือ MPI::MAX, MPI::MIN และ MPI::SUM ในกรณีดังต่อไปนี้

- MPI::MAX จะใช้เมื่อผลลัพธ์ มีค่ามากกว่าหรือเท่ากับ 0 ทั้งหมด
- MPI::MIN จะใช้เมื่อผลลัพธ์ มีค่าน้อยกว่าหรือเท่ากับ 0 ทั้งหมด
- MPI::SUM จะใช้เมื่อผลลัพธ์มีค่าปนกันทั้ง 0 มากกว่า 0 หรือน้อยกว่า 0

จากการศึกษากระบวนการทำงานของการประมวลผลขั้นปลายทั้งหมด ผู้วิจัยได้รวบรวมวิธีการรวบรวมข้อมูลกลับที่เหมาะสมสำหรับการแบ่งงานในแต่ละฟังก์ชันที่สามารถปรับปรุงการทำงานเป็นแบบขนานได้ดังตารางที่ 3-1

ตารางที่ 3-1 การใช้งานฟังก์ชันการรวบรวมข้อมูลกลับในแต่ละฟังก์ชันของการประมวลผลขั้นปลาย

ชื่อฟังก์ชัน	ลักษณะการแบ่งงาน	ข้อมูลที่รวบรวมกลับ	วิธีการรวบรวมข้อมูลกลับ
coerceSurroundedPixelsY	กลุ่มของคอตัมน์	disparity map	MPI:: Allreduce, MPI::MAX
compute_igxy	กลุ่มของแถว	igx	MPI::Allgather
	กลุ่มของคอตัมน์	igy	MPI:: Allreduce, MPI::MAX
removeIsolatedPixelsX	กลุ่มของแถว	igy	MPI::Allgather
compute_igyd	กลุ่มของคอตัมน์	igyd	MPI:: Allreduce, MPI::SUM
computeReliabilitiesY	กลุ่มของคอตัมน์	reliability map	MPI:: Allreduce, MPI::SUM
propagateY	กลุ่มของคอตัมน์	reliability map	MPI:: Allreduce, MPI::MAX
		disparity map	MPI:: Allreduce, MPI::MAX
removeIsolatedPixelsY	กลุ่มของคอตัมน์	igx	MPI:: Allreduce, MPI::MIN
compute_igxd	กลุ่มของแถว	igxd	MPI::Allgather
computeReliabilitiesX	กลุ่มของแถว	reliability map	MPI::Allgather
propagateX	กลุ่มของแถว	reliability map	MPI::Allgather
		disparity map	MPI::Allgather
modefilterY	กลุ่มของคอตัมน์	disparity map	MPI:: Allreduce, MPI::MAX
modefilterX	กลุ่มของแถว	disparity map	MPI::Allgather

### 3.4. การประมาณค่าระยะทางของวัตถุในระบบตรวจจับวัตถุเพื่อผู้พิการทางสายตา



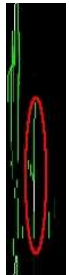

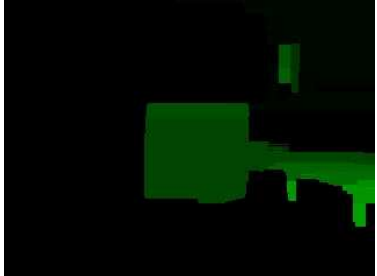
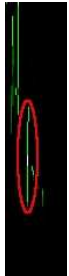

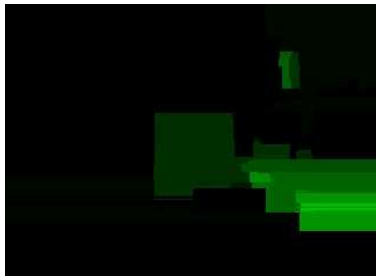
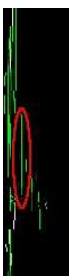
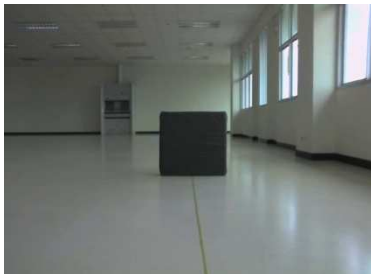
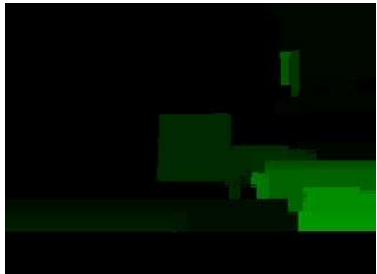
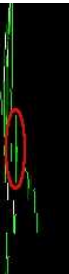


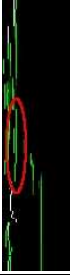
หากต้องการทราบว่าวัตถุอยู่ที่บริเวณใดของภาพและมีระยะทางเท่าไรจากตัวผู้พิการทางสายตาจนถึงวัตถุชิ้นนั้น สามารถทำได้โดยการแทนค่าระดับคิสพาริตีที่บริเวณที่มีวัตถุปรากฏอยู่ลงในสมการพหุนามที่ใช้สำหรับประมาณค่าระยะทาง ผลที่ได้คือระยะทางของวัตถุโดยประมาณ ส่วนสมการประมาณค่าระยะทางนั้น เป็นการประมาณค่าในช่วงโดยใช้สมการพหุนาม ซึ่งประมาณค่าระยะทางจริงที่สอดคล้องกับระดับคิสพาริตีที่นั้นๆ โดยมีรายละเอียดดังต่อไปนี้

#### 3.4.1. สมการประมาณค่าระยะทางวัตถุ

ข้อมูลที่ใช้สำหรับการประมาณค่าในช่วงเพื่อสร้างสมการพหุนามสำหรับใช้ประมาณระยะทางวัตถุนั้น ได้จากการเก็บข้อมูลจากระดับคิสพาริตีที่มีความสัมพันธ์กับระยะทางจริงในแต่ละช่วงของคิสพาริตีที่นั้นๆ โดยนำแผนผังคิสพาริตีมาแปลงข้อมูลให้อยู่ในรูป V-disparity ก่อน จากนั้นจึงเก็บข้อมูลเส้นตรงในแนวดิ่งซึ่งเป็นตัวแทนของวัตถุมาใช้สร้างสมการ วัตถุที่ใช้เป็นตัวอย่งในการเก็บข้อมูลมีลักษณะเป็นสี่เหลี่ยมจัตุรัสขนาด 60x60 เซนติเมตร โดยที่ตัวอย่างภาพที่ใช้เก็บข้อมูลนั้นแสดงไว้ในตารางที่ 3-2 ซึ่งแสดงภาพซ้าย แผนผังคิสพาริตี และ V-disparity ที่สัมพันธ์กับระยะทางที่กำหนดไว้ โดยที่ข้อมูลที่น่าสนใจนี้มีระยะทางตั้งแต่ระยะทาง 1 – 7 เมตร ดังแสดงข้อมูลทั้งหมดไว้ในตารางที่ 3-3

ตารางที่ 3-2 ตัวอย่างรูปภาพแผนผังคิสพาริตี ภาพ V-disparity ที่ใช้ในการเก็บข้อมูลเพื่อประมาณระยะทางของวัตถุที่ขีดขวาง

ระยะทาง (m)	ภาพซ้าย	แผนผังคิสพาริตี	V-disparity
1.0			

ระยะทาง (m)	ภาพถ่าย	แผนผังดิสพาริตี	V-disparity
1.5			
2.0			
2.6			
3.0			
3.6			

ตารางที่ 3-3 ระดับคิสปาริที่ที่ตรงกับระยะทางจริงของวัตถุ

ระยะทาง (m)	ระดับคิสปาริที่		
	ค่าน้อยสุด	ฐานนิยม	ค่ามากที่สุด
1.0	34	35	35
1.1	31	32	32
1.2	29	30	31
1.3	28	28	29
1.4	26	26	27
1.5	24	25	25
1.6	23	23	24
1.7	22	23	24
1.8	20	21	21
1.9	20	21	21
2.0	19	19	19
2.2	18	18	18
2.4	17	17	17
2.6	15	16	16
2.8	15	15	15
3.0	14	14	14
3.2	13	13	13
3.4	13	13	13
3.6	12	12	12
3.8	12	12	12
4.0	12	12	12
4.2	12	12	12
4.4	12	12	12
4.6	11	11	11
4.8	11	11	11

ระยะทาง (m)	ระดับดิสพาริตี		
	ค่าน้อยสุด	ฐานนิยม	ค่ามากที่สุด
5.0	11	11	11
5.2	11	11	11
5.4	10	10	10
5.6	10	10	10
5.8	10	10	10
6.0	10	10	10
6.2	10	10	10
6.4	10	10	10
6.6	8	8	10
6.8	8	8	8
7.0	8	8	8

จากตารางที่ 3-3 ระดับดิสพาริตีที่เราสนใจในที่นี้คือฐานนิยมเนื่องจากข้อมูลที่เกิดขึ้นซ้ำกันบ่อยๆ แสดงว่าอัลกอริทึมการประมวลผลภาพสเตอริโอให้ผลลัพธ์เป็นข้อมูลชุดนั้นมีความถี่มากกว่าชุดอื่นๆ ในระยะทางที่เก็บข้อมูล ข้อมูลจากฐานนิยมในแต่ละระยะทางนั้นจะนำมาประมาณค่าในช่วงโดยใช้สมการพหุนาม เพื่อนำสมการที่ได้ไปใช้ในการประมาณค่าระยะทางต่อไป สมการที่ได้จากขั้นตอนนี้คือดังสมการที่ (12) ดังนี้

$$d = 1.28187598 \times 10^{-5}x^4 - 1.91706518 \times 10^{-3}x^3 + 9.87373726 \times 10^{-2}x^2 - 2.17953355x + 19.1983965, x \leq 35 \quad (12)$$

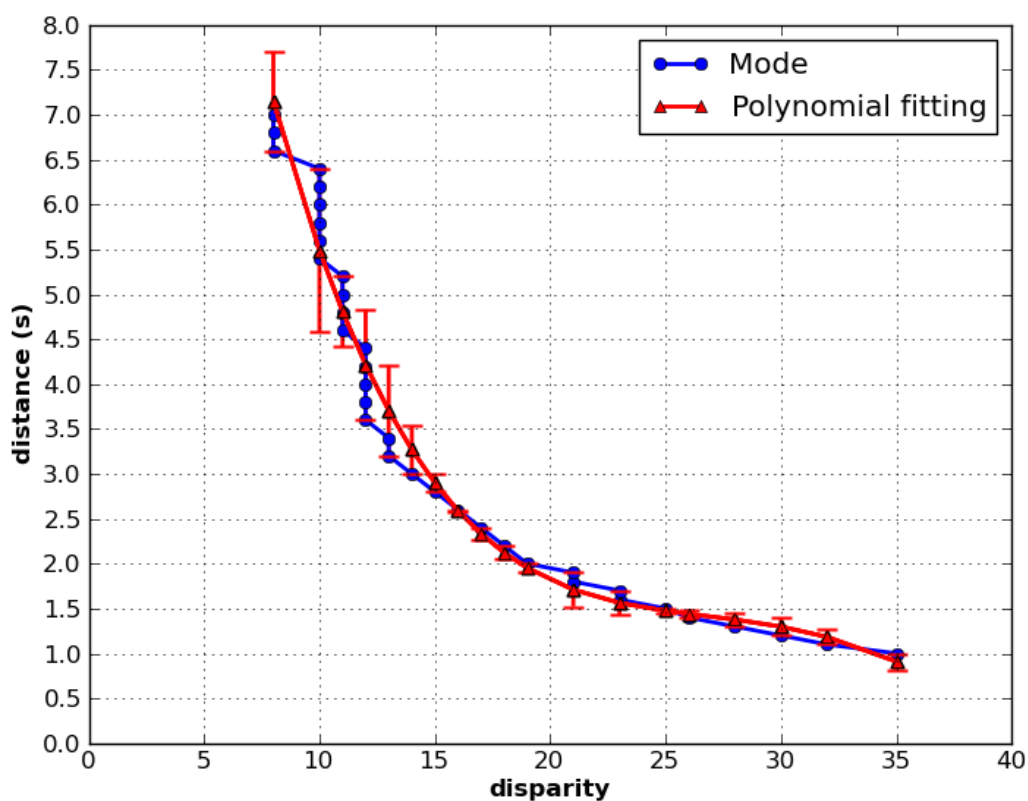
โดยที่  $d$  คือระยะทางของวัตถุที่ได้จากสมการมีหน่วยเป็นเมตร(m)  $x$  คือระดับของดิสพาริตี

เมื่อแทนค่าระดับดิสพาริตีลงในสมการที่ (12) จะได้ดังตารางที่ 3-4 ปรากฏว่าสำหรับวัตถุที่อยู่ใกล้ผู้พิกานั้นสมการนี้จะสามารถประมาณระยะทางได้แม่นยำกว่าสิ่งกีดขวางที่อยู่ไกลออกไป สังเกตได้จากค่าความผิดพลาดที่แสดงไว้ในตารางที่ 3-4 และเมื่อพิจารณารูปที่ 3-15 จะเห็นได้ว่าสมการที่ (12) นั้นสามารถประมาณระยะทางได้ โดยที่บริเวณที่อยู่ตอนล่างของกราฟเป็นระยะทางที่มีความปลอดภัยในการเดินทางสำหรับผู้พิกar แต่หากสังเกตช่วงระยะทางตั้งแต่ 1 ถึง 1.5 เมตร ซึ่งเป็นระยะทางที่มีความสำคัญต่อการตรวจสอบสิ่งกีดขวางสำหรับผู้พิกar เนื่องจาก

ระยะนี้เป็นระยะที่เกิดผู้พิการทางสายตามากอาจส่งผลให้ผู้พิการทางสายตาอาจหลบหลีกวัตถุได้ยาก สมการประมาณระยะทางวัตถุสมการที่ (12) นั้นอาจจะส่งผลให้ผู้พิการประสบอุบัติเหตุได้ ดังนั้นจึงเปลี่ยนข้อมูลที่ใช้ประมาณค่าใหม่โดยเลือกเอา ข้อมูลระยะทางที่น้อยที่สุดในแต่ละระดับ disparities ที่ดังแสดงในตารางที่ 3-5 เพื่อไปประมาณระยะทางวัตถุใหม่ จึงได้สมการประมาณระยะวัตถุใหม่ดังสมการที่ (13)

$$d = 3.57636692 \times 10^{-5}x^4 - 3.78767563 \times 10^{-3}x^3 + 1.50193793 \times 10^{-1}x^2 - 2.70909311x + 20.5754034, x \leq 35 \quad (13)$$

โดยที่  $d$  คือระยะทางของวัตถุที่ได้จากสมการมีหน่วยเป็นเมตร(m)  $x$  คือระดับของ disparity



รูปที่ 3-15 การหาความสัมพันธ์ระหว่างระยะทางกับระดับ disparities ที่ด้วยสมการพหุนามกำลัง 4 ตามสมการที่ (12)

ตารางที่ 3-4 ระยะทางที่ได้จากสมการที่ (12) เมื่อเทียบกับระยะทางเฉลี่ยที่ระดับดิสพาริตีต่างๆ

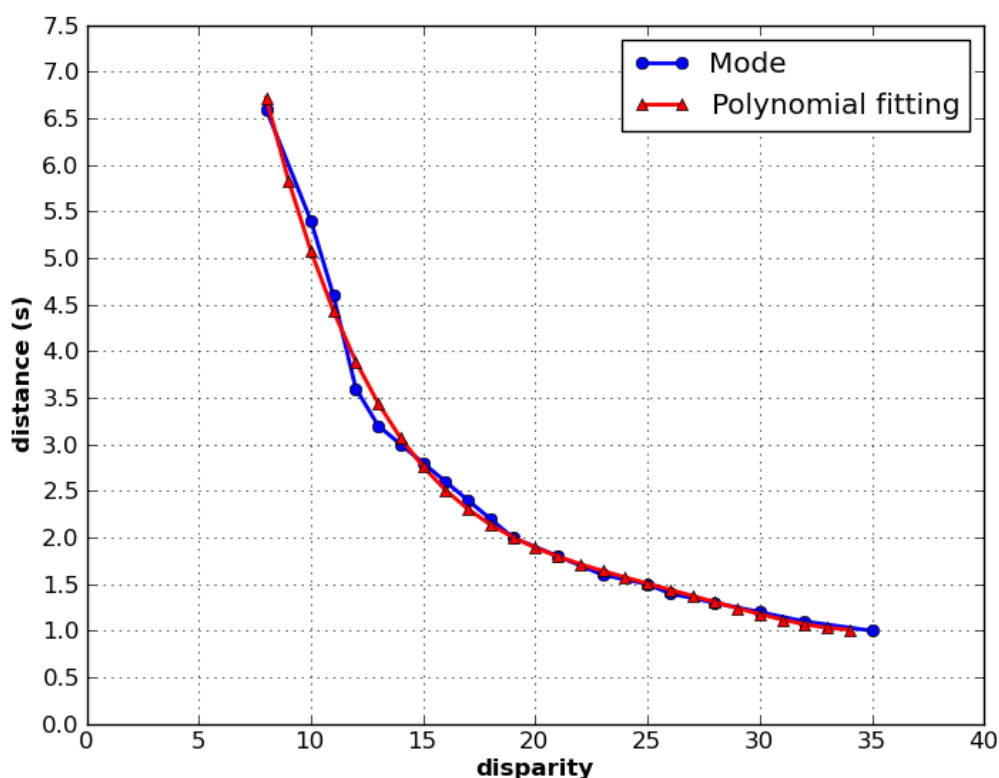
ระดับดิสพาริตี	ระยะทาง (m)		ความผิดพลาด
	ระยะทางจริงเฉลี่ย	ระยะทางจากสมการ	
35	1.00	0.910	0.090
32	1.10	1.183	0.083
30	1.20	1.298	0.098
28	1.30	1.377	0.077
26	1.40	1.440	0.040
25	1.50	1.474	0.026
23	1.65	1.563	0.137
21	1.85	1.7104	0.190
19	2.00	1.953	0.047
18	2.20	2.123	0.077
17	2.40	2.334	0.066
16	2.60	2.590	0.010
15	2.80	2.900	0.100
14	3.00	3.269	0.269
13	3.30	3.705	0.505
12	3.8	4.215	0.615
11	4.9	4.807	0.393
10	5.9	5.488	0.912
8	6.8	7.152	0.552

จากข้อมูลในตารางที่ 3-5 และสมการที่ (13) แสดงผลการลัพท์ในรูปแบบที่ 3-16 โดยสมการที่ได้จะให้ผลต่ำกว่าสมการที่ (12) ซึ่งอาจจะให้ระยะทางที่น้อยกว่าระยะทางของวัตถุจริง แต่จากสมการที่ (13) ตั้งแต่ระยะทาง 1 ถึง 3 เมตร จะไม่มีวัตถุใดมีระยะทางน้อยกว่าค่าที่ได้จากสมการนี้ แต่จะมีระยะทาง 3 ถึง 4 เมตรที่อาจจะมิวัตถุที่มีระยะน้อยกว่าค่าที่ได้จากสมการได้ แต่หากมีวัตถุในระยะนี้จริงเมื่อผู้พิจารณาเดินทางไปในระยะห่าง 1-3 เมตร วัตถุจะถูกลบประมาณค่าใหม่จนได้ค่าที่ถูกต้อง



ตารางที่ 3-5 ระดับคิสปาริทีที่ตรงกับระยะทางจริงของวัตถุ โดยเลือกข้อมูลระยะทางที่น้อยที่สุดจากแต่ละระดับของคิสปาริที

ระยะทาง (m)	ระดับคิสปาริที (ฐานนิยม)
1.0	35
1.1	32
1.2	30
1.3	28
1.4	26
1.5	25
1.6	23
1.7	23
1.8	21
2.0	19
2.2	18
2.4	17
2.6	16
2.8	15
3.0	14
3.2	13
3.6	12
4.6	11
5.4	10
6.6	8



รูปที่ 3-16 ความสัมพันธ์ระหว่างระยะทางกับคิสปาริตีด้วยสมการพหุนามกำลัง 4 ตามสมการที่ (13) ที่คำนึงถึงบริเวณที่ปลอดภัยสำหรับการเดินทางของผู้พิการทางสายตา

จากที่กล่าวมาแล้วในข้างต้นสมการที่ได้จากขั้นตอนี่จึงมีสองสมการคือ สมการที่ (12) และสมการที่ (13) โดยสมการที่ (12) นั้นสามารถประมาณระยะทางวัตถุได้ใกล้เคียงกว่าสมการที่ (13) แต่อาจมีบางช่วงของคิสปาริตีที่อาจส่งผลให้เกิดอันตรายแก่ผู้พิการทางสายตาได้อีกทั้งช่วงดังกล่าวเป็นระยะทางที่ใกล้กับผู้พิการมากคือระยะ 1-3 เมตร ดังนั้นสมการที่ (13) จึงให้ระยะทางที่มีความปลอดภัยในการเดินทางแก่ตัวผู้พิการมากกว่าถึงแม้ว่าอาจคลาดเคลื่อนจากระยะจริงเล็กน้อยก็ตาม

#### 3.4.2. การแปลผลข้อมูลจาก U และ V disparity

มนุษย์สามารถคาดคะเนระยะทางหรือตำแหน่งของวัตถุต่างๆ ได้ด้วยการมองเห็น แต่สำหรับผู้พิการทางสายตานั้นไม่สามารถทำได้ การที่จะเสนอข้อมูลตำแหน่งของวัตถุในภาพแก่ผู้พิการทางสายตานั้นจึงมีข้อจำกัดตามไปด้วย ระบบ ENVIS [10] ได้เสนอวิธีการบอกตำแหน่งของวัตถุจากแผนผังคิสปาริตี โดยแบ่งแผนผังคิสปาริตีออกเป็น 10 ช่วงตามแนวตั้งเท่าๆ กัน ในแต่ละช่วงเป็นการคำนวณจำนวนคิสปาริตีในแต่ละระดับ แล้วแสดงผลออกเป็นระดับความสูงในแต่ละ

ช่วง หากสูงมากแสดงว่าบริเวณช่วงที่คำนวณมีจำนวนคิสปาริตีอยู่มากซึ่งในระบบนี้จะตั้งสมมุติฐานว่ามีวัตถุอยู่ใกล้ตัวผู้พิจารณาตามไปด้วยและระบบนี้ไม่เน้นการประมาณระยะทางของวัตถุตามจริง

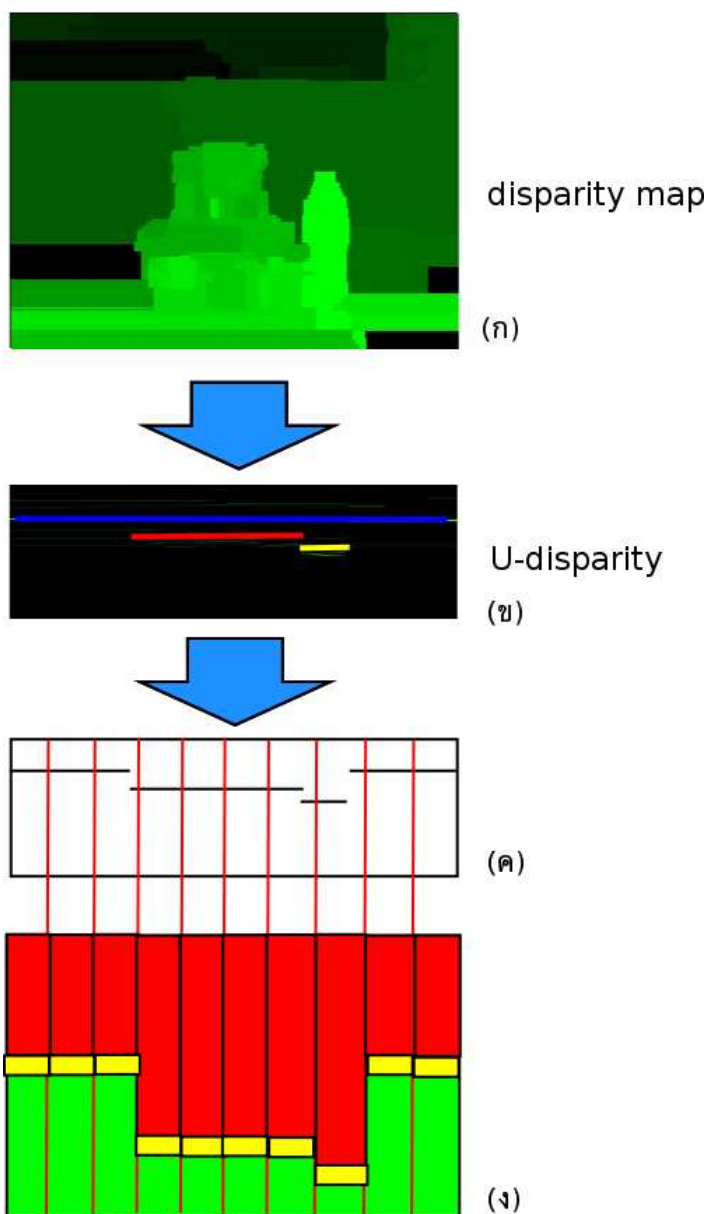
การแบ่งช่วงของแผนผังคิสปาริตีตามแนวตั้งเพื่อเสนอข้อมูลแก่ผู้พิจารณาทางสายตาเป็นวิธีการที่เหมาะสม เนื่องจากการแบ่งออกเป็นช่วงๆ ในแนวตั้ง ส่งผลให้ผู้พิจารณาทางสายตาสามารถรับรู้ตำแหน่งของวัตถุได้ง่าย แต่การใช้จำนวนของระดับคิสปาริตีเพียงอย่างเดียวอาจจะส่งผลให้ผู้พิจารณาทางสายตาได้รับข้อมูลสภาพแวดล้อมที่ผิดพลาด เพื่อให้ผู้พิจารณาสามารถรับรู้สภาพแวดล้อมรวมถึงระยะทางได้แม่นยำขึ้นจึงนำ U-disparity เข้ามาช่วยในส่วนนี้

โดยปรกติแล้ว U และ V disparity มีความสัมพันธ์กัน โดยที่ V-disparity สามารถระบุวัตถุได้ง่ายโดยปรากฏเป็นเส้นตรงในแล้วคิง ในขณะที่ U-disparity สามารถบอกได้ว่าวัตถุอยู่บริเวณใดของภาพ เพื่อให้ผู้พิจารณาสามารถทราบถึงสภาพแวดล้อมรอบตัวได้ง่าย ดังนั้นวิทยานิพนธ์นี้จึงเสนอใช้บริเวณ U-disparity มาเป็นส่วนช่วยบอกว่าวัตถุอยู่ตำแหน่งใดบ้าง เมื่อได้ตำแหน่งของวัตถุหรือสิ่งกีดขวางมาแล้ว จึงนำระดับคิสปาริตีในส่วนนั้นมาประมาณค่าระยะทางและแสดงออกเป็นช่วงๆ เพื่อนำไปใช้สร้างส่วนนำเสนอข้อมูลแก่ผู้พิจารณาต่อไป

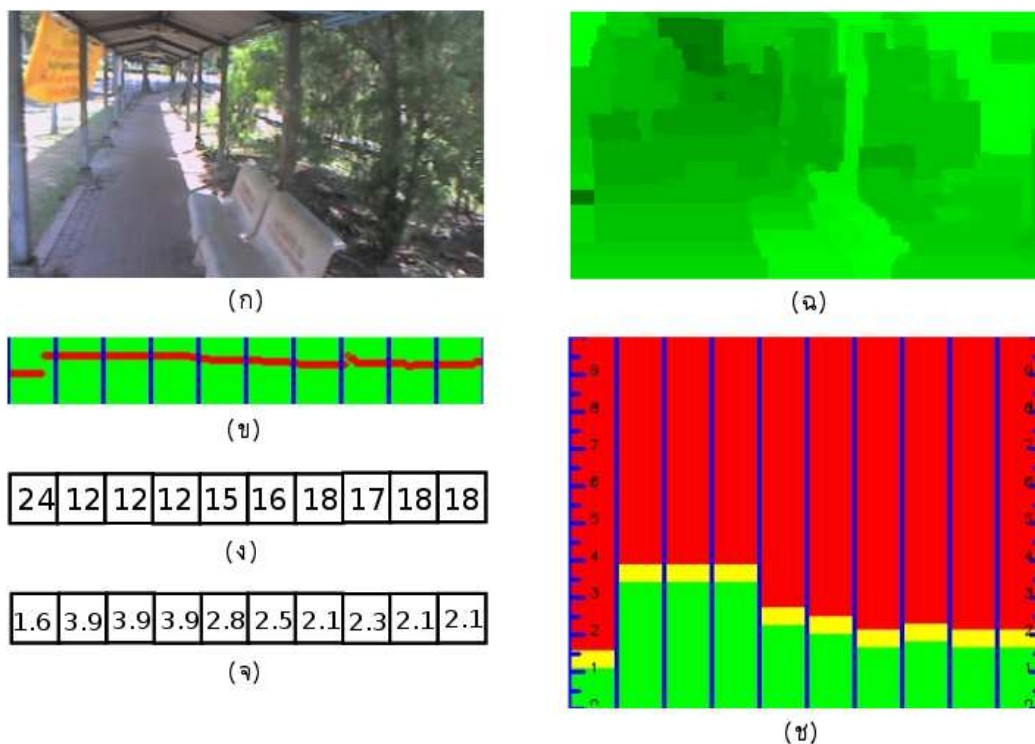
ขั้นตอนกระบวนการในการสร้างส่วนตรวจสอบสภาพแวดล้อมแสดงได้ดังรูปที่ 3-17 มีรายละเอียดดังต่อไปนี้

- 1) แปลงแผนผังคิสปาริตีให้อยู่ในรูปของ U-disparity
- 2) กำหนดช่วงให้กับ U-disparity ในแนวคิงโดยแบ่งออกเป็นช่วงๆ เท่าๆ กัน โดยในที่นี้แบ่งออกเป็น 10 ช่วง
- 3) ในแต่ละช่วงที่กำหนดให้กับ U-disparity นั้น หาพิกเซลที่มีระดับคิสปาริตีที่สนใจปรากฏอยู่ ระดับคิสปาริตีที่สนใจดังกล่าวจะให้ระยะทางตั้งแต่ 1 ถึง 6 เมตร โดยที่ค่าคิสปาริตีนั้นจะมีค่าตั้งแต่ 0 ถึง 35 นั้นเอง ขั้นตอนนี้เป็น ไปดังรูปที่ 3-17 (ค)
- 4) หาความถี่ของระดับคิสปาริตีในแต่ละช่วง แล้วเลือกระดับคิสปาริตีที่เป็นฐานนิยมเพื่อเป็นข้อมูลที่ใช้เป็นแทนคิสปาริตีในแต่ละช่วงนั้นๆ
- 5) เปลี่ยนระดับคิสปาริตีที่ได้จากกระบวนการก่อนหน้าในแต่ละช่วงคิสปาริตีให้อยู่ในรูประยะทาง โดยใช้สมการประมาณระยะทาง สมการที่ (13)
- 6) แปลผลระยะทางของวัตถุในแต่ละช่วงให้สามารถเข้าใจได้ง่ายดังรูปที่ 3-17 (ง)

ข้อมูลที่ได้จากกระบวนการนี้เป็นข้อมูลอาร์เรย์แบบ float ที่ใช้เก็บระยะทางของวัตถุตามช่วงที่กำหนดให้ในกับ U-disparity โดยสามารถนำข้อมูลส่วนนี้ไปสร้างส่วนติดต่อกับผู้พิจารณาต่อไปได้ดังรูปที่ 3-18 (จ)



รูปที่ 3-17 กระบวนการระบุตำแหน่งของวัตถุโดยใช้ U-disparity (ก) แผนผังคิสพาริตี (ข) แปลงแผนผังคิสพาริตีมาเป็น U-disparity (ค) เลือกระดับคิสพาริตีที่มีค่าระหว่าง 0 ถึง 35 (ง) นำระดับคิสพาริตีจาก (ค) มาประมาณระยะทางด้วยสมการ แล้วนำมาแสดงผล



รูปที่ 3-18 ตัวอย่างผลที่ได้จากกระบวนการประมาณค่าระยะทางและหาตำแหน่งของวัตถุ (ก) ภาพซ้าย (ข) U-disparity ที่สนใจ (ง) ค่าคิสพาริตีที่หาได้ในแต่ละช่วง (จ) ระยะทางที่ได้จากสมการ (13) (ฉ) แผนผังคิสพาริตี (ซ) แปลงผลจาก (จ) ให้คนสายตาปรกติเข้าใจได้ง่าย

จากขั้นตอนที่อธิบายไว้สามารถเขียนเป็นสมการการเลือกคิสพาริตีในแต่ละช่วงที่พร้อมจะนำไปประมาณระยะทางได้ดังสมการที่ (15) เมื่อได้  $disp$  ซึ่งเป็นระดับคิสพาริตีที่ต้องการแล้วนำไปแทนค่าในสมการที่ (13) จะได้ระยะทางออกมา ในส่วนการเลือกระดับคิสพาริตีที่สนใจก่อนการหาคิสพาริตีที่เหมาะสมในการประมาณค่าระยะทางนั้นเป็นไปดังสมการที่ (14) เมื่อผ่านกระบวนการนี้จะได้อัปเดต U-disparity ที่สนใจ

$$y_i = \max_{i \in [0, width]} (I_{\Delta u_i}), 0 \leq I_{\Delta u_i} \leq 35 \quad (14)$$

จากสมการที่ (14) เป็นการหาค่า  $I_{\Delta u_i}$  (ข้อมูลที่บรรจุอยู่ใน U-disparity แถวที่  $i$ ) ที่มากที่สุดที่ไม่เกิน 35 ในทุกๆ คอลัมน์  $i$  ตั้งแต่ 0 ไปจนถึง  $width$  ซึ่งเป็นความกว้างของ U-disparity ซึ่งค่าที่ได้คือ  $y_i$  จากนั้นนำ  $y_i$  มาแทนค่าในสมการที่ (15) ก็จะได้  $disp$  ซึ่งเป็นข้อมูลที่ได้จากการหาสถิติที่มากที่สุด (ฐานนิยม) ของชุดข้อมูล  $y_i$  ในแต่ละช่วงโดยเริ่มต้นจากคอลัมน์ที่  $b$  ถึงคอลัมน์

ที่  $e$  โดยที่  $b$  คือดัชนีคอลัมน์เริ่มต้นในช่วงที่ต้องคำนวณไปจนถึงดัชนีคอลัมน์  $e$  จากกระบวนการทั้งหมดที่ได้กล่าวไปแล้วนี้สามารถเขียนเป็นโค้ดเทียม (pseudocode) ได้ดังรูปที่ 3-19 และตัวอย่างผลการทำงานแสดงดังรูปที่ 3-18

$$disp = \max_{i \in [b, e]} (Frequency(y_i))$$

(15)

```

segmentSize = 10 // number of segment
divideSize = u-disparity width / segmentSize
maxDisparity = 35
interestDisp [ u-disparity width ] initial 0 all element

for i along to u-disparity width:
  for j = maxDisparity to 0:
    if interestDisp[i] > 0:
      continue
    if u-disparity[i,j] contain disparity:
      interestDisp[i] = j

for i = 0 to segmentSize - 1:
  b = i * divideSize
  e = i * divideSize + divideSize - 1

  initial all element of frequency to 0

  for j = b to e:
    frequency [interestDisp[j]]++

  result[i] = max(frequency)

```

รูปที่ 3-19 โค้ดเทียมของกระบวนการประมาณระยะทางสิ่งกีดขวาง

การแปลผลระยะทางของวัตถุในแต่ละช่วงให้สามารถเข้าใจได้ง่ายขึ้นนี้เป็นข้อมูลที่แสดงให้คนสายตาปรกติดูเท่านั้น แต่ในความเป็นจริงแล้วข้อมูลที่ได้จะเป็นอาร์เรย์ที่มีขนาดเท่ากับช่วงที่กำหนดให้กับ U-disparity โดยในอาร์เรย์จะบรรจุระยะทางที่ตรวจสอบพบวัตถุ จากข้อมูลดังกล่าวผู้วิจัยจึงนำเสนอออกมาให้อยู่ในลักษณะรูปภาพดังรูปที่ 3-18 (ข) โดยที่แบ่งรูปนี้ออกเป็นช่วงๆ ตามแนวตั้งเท่ากับจำนวนช่วงที่ได้กำหนดให้ U-disparity ในแต่ละช่วงนั้นจะมีสีต่างๆ สามสีด้วยกัน มีความหมายดังต่อไปนี้

- สีแดงหมายถึงบริเวณที่ผู้พิการทางสายตาไม่สามารถเดินทางเข้าไปได้
- สีเขียวคือบริเวณที่ผู้พิการสามารถเดินทางได้

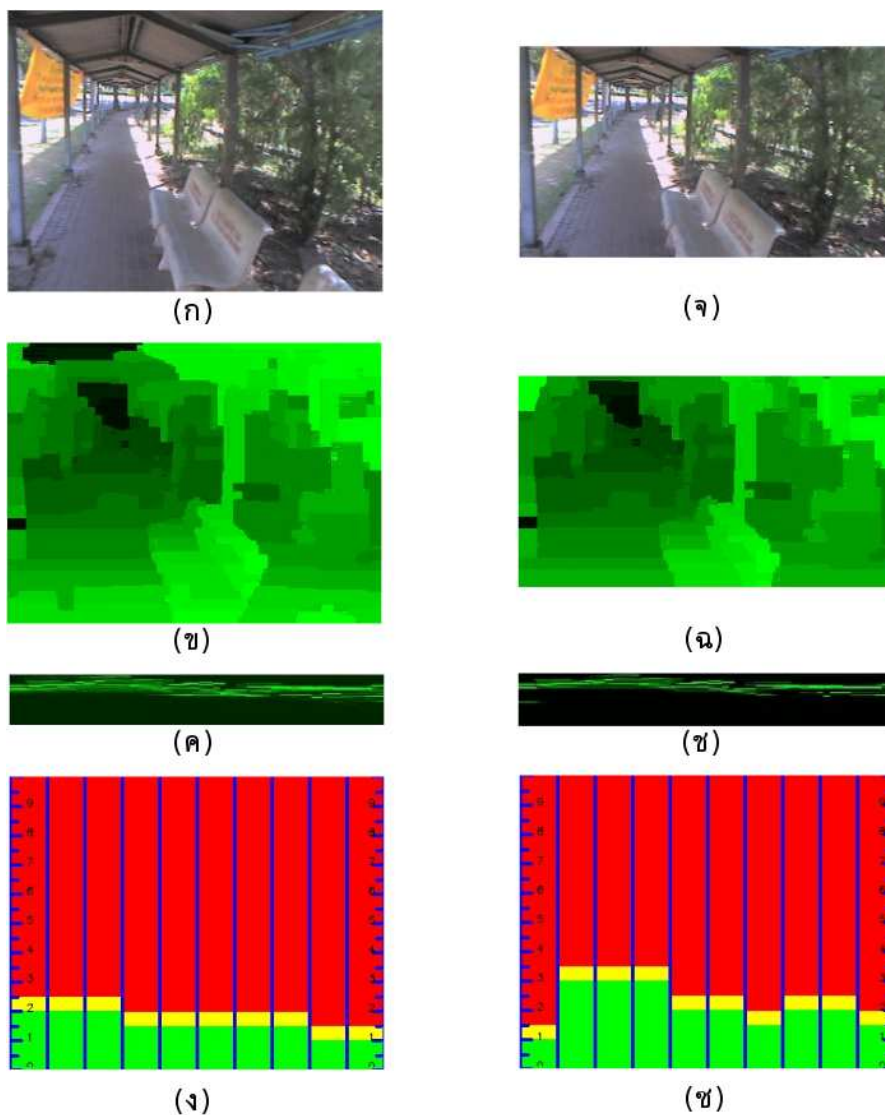
- สีเหลืองเป็นบริเวณที่อยู่ใกล้กับวัตถุ ผู้พิจารณาควรเลือกบริเวณที่จะเดินใหม่ มิฉะนั้น อาจพบวัตถุหรือสิ่งกีดขวางได้ โดยกำหนดให้บริเวณนี้อยู่ที่ระยะประมาณ 0.5 เมตร

ทางด้านซ้ายและขวาของรูปที่ 3-18 (ข) นั้นจะมีตัวเลขขนาดเล็กรากฏอยู่ โดยที่ตัวเลขดังกล่าวเป็นมาตรวัดระยะทางที่มีหน่วยเป็นเมตรเพื่อให้สามารถอ่านข้อมูลได้สะดวกขึ้น

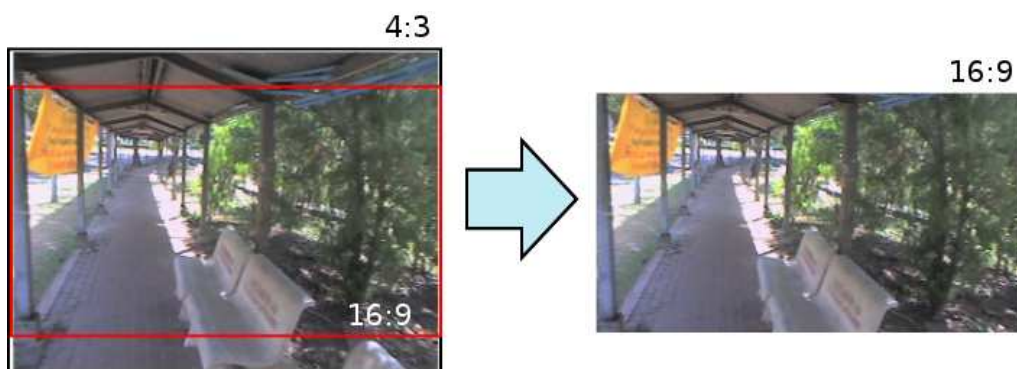
จากรูปที่ 3-18 เป็นตัวอย่างผลที่ได้จากกระบวนการประมาณค่าระยะทางและหาตำแหน่งของวัตถุ โดยที่ภาพ (ก) เป็นภาพซ้าย หลังจากผ่านอัลกอริทึม P2P จะได้ดังภาพ (ข) ซึ่งเป็นแผนผังคิสพาริตีเสร็จแล้วแปลงข้อมูลให้อยู่ในรูปของ U-disparity แล้วเลือกระดับคิสพาริตีที่สนใจ ข้อมูลตัวอย่างดังกล่าวสามารถแสดงได้ดังภาพ (ข) เลือกคิสพาริตีจากฐานนิยามในแต่ละช่วงจะได้อาร์เรย์ข้อมูลดังแสดงในภาพ (ง) นำระดับคิสพาริตีจาก (ง) มาประมาณค่าระยะทางด้วยสมการ (13) จะได้อาร์เรย์เก็บข้อมูลระยะทางของวัตถุซึ่งค่าที่ได้แสดงไว้ในภาพ (จ) เสร็จแล้วแปลงข้อมูลให้คนสายตาปรกติเข้าใจดังภาพ (ข)

### 3.4.3. อัตราส่วนลักษณะของภาพ

ภาพที่ใช้สำหรับทดสอบระบบมีขนาด 320x240 พิกเซล และอัตราส่วนลักษณะ (aspect ratio) เป็น 4:3 ซึ่งเป็นขนาดมาตรฐานสำหรับภาพเคลื่อนไหวที่ถ่ายจากกล้อง VDO ทั่วไป ซึ่งเว็บแคม Logitech ที่ใช้ในงานวิจัยนี้จะให้ภาพในอัตราส่วนนี้เช่นกัน แต่ภาพในลักษณะนี้ หากผู้ใช้งานระบบมองตรงไปข้างหน้า ขอบล่างและขอบบนของภาพจะบรรจบทัศนียภาพได้ดังรูปที่ 3-4 ทั้งนี้ขึ้นอยู่กับความสูงของผู้ใช้งานเองด้วย ซึ่งเมื่อนำภาพเหล่านี้ไปคำนวณหาแผนผังคิสพาริตีแล้วแปลงเป็น U-disparity เพื่อนำไปใช้งานต่อไปนั้น จะพบพื้นดินหรือเพดานในปริมาณมากส่งผลให้เกิดการประมาณค่าตำแหน่งของวัตถุผิดพลาดได้ โดยมีกระบวนว่าพื้นดินเป็นวัตถุดังรูปที่ 3-20 (ก-ง) ดังนั้นระบบที่ได้ออกแบบไว้จะแจ้งว่ามีวัตถุอยู่ใกล้ตลอดเวลาซึ่งสามารถแก้ไขได้โดยเมื่ออยู่ในที่โล่งแจ้งไม่มีหลังคาผู้ใช้งานควรเงยหน้าขึ้นเล็กน้อย เพื่อให้มีส่วนของพื้นในภาพน้อยลง แต่หากเป็นบริเวณดังแสดงไว้ในรูปที่ 3-20 (ก) จะไม่สามารถใช้วิธีการนี้ได้ เพื่อลดพื้นที่บริเวณด้านบนและด้านล่างลง ผู้วิจัยจึงเสนอให้ใช้วิธีการเปลี่ยนอัตราส่วนลักษณะ โดยใช้อัตราส่วน 16:9 แทนวิธีการปรับเปลี่ยนอัตราส่วนลักษณะแสดงดังรูปที่ 3-21 ซึ่งยังคงความกว้างของรูปในอัตราส่วน 4:3 ไว้แล้วปรับลดความยาวลง เมื่อนำภาพขนาด 320x240 พิกเซลมาปรับอัตราส่วนก็จะได้ภาพขนาด 320x180 พิกเซลแทน



รูปที่ 3-20 ความแตกต่างระหว่างอัตราส่วน 4:3 รูป (ก-ง) และ 16:9 (จ-ซ)



รูปที่ 3-21 การแปลงอัตราส่วนลักษณะ 4:3 ภาพวิดีโอมาตรฐานไปเป็น 16:9



จากการทดสอบโดยใช้ภาพที่มีอัตราส่วนลักษณะ 16:9 หรือ 320x180 พิกเซลเพื่อคำนวณหาแผนผังคิสพาริตี ผลที่ได้จากการทดสอบจะเป็นไปดังรูปที่ 3-20 (จ-ข) หากพิจารณารูปที่ 3-20 (จ) และ (ข) ซึ่งเป็นแผนผังคิสพาริตีในอัตราส่วนลักษณะ 4:3 และ 16:9 นั้น จะพบระดับคิสพาริตีที่มีค่าสูงๆ บริเวณพื้นทางเดินและหลังคาในแผนผังคิสพาริตีที่ได้ดังรูปที่ 3-20 (ข) เนื่องจากบริเวณนี้อยู่ใกล้ผู้ใช้งาน ส่งผลให้ส่วนการแปลผลที่ออกแบบไว้ประมาณระยะทางของสิ่งกีดขวางใช้โดยใช้คิสพาริตีบริเวณพื้นทางเดินและหลังคาเป็นหลัก ซึ่งให้ระยะทางที่ 1.5 ถึง 2.5 เมตรเท่านั้น แต่เมื่อเปลี่ยนมาใช้อัตราส่วนลักษณะ 16:9 ขนาดภาพที่ได้จะมีความยาวลดลง ส่งผลให้แผนผังคิสพาริตีที่ได้จากการคำนวณด้วยอัลกอริทึม P2P นั้นมีบริเวณพื้นทางเดินและหลังคาน้อยลงได้ ซึ่งทำให้ระบบสามารถประมาณระยะทางที่ผู้พิการสามารถเดินได้เหมาะสมมากขึ้น จากรูปที่ 3-20 (ข) เป็นการกำหนดอัตราส่วนลักษณะ 16:9 สามารถประมาณระยะทางได้ตั้งแต่ 1.5 ถึง 3.5 เมตร และจากการลดอัตราส่วนลักษณะลงส่งผลให้เวลาของการตอบสนองของอัลกอริทึม P2P ที่ใช้เพื่อคำนวณคิสพาริตีลดลงตามไปด้วยเนื่องจากอัลกอริทึม P2P จะคำนวณทุกๆ แถวของพิกเซลที่เป็นความยาวของภาพ การลดความยาวของภาพลงจึงทำให้เวลาในการประมวลผลลดลงตามไปด้วย

### 3.5. สรุป

ในบทนี้นำเสนอแนวคิดการออกแบบระบบตรวจสอบสิ่งกีดขวางสำหรับผู้พิการทางสายตาด้วยสเตอริโอวิชัน โดยเริ่มจากการประดิษฐ์กล้องสเตอริโออย่างง่ายด้วยกล้องเว็บแคมราคาถูก ในส่วนต่อมาเป็นวิธีการที่สำคัญที่สุดในระบบคือการนำภาพสเตอริโอมาประมวลผลด้วยอัลกอริทึม P2P จากการทดสอบค่าระดับคิสพาริตีที่เหมาะสมในการใช้งานสำหรับระบบนี้คือ 50 ถึง 100 ส่วนพารามิเตอร์อื่นนั้นใช้ค่าที่ผู้คิดค้นเสนอไว้ เนื่องจากพบว่าเมื่อทดลองปรับเปลี่ยนค่าต่างๆ แผนผังคิสพาริตีที่มีการเปลี่ยนแปลงน้อยมาก ในส่วนของผลตอบสนองจากอัลกอริทึม P2P นั้นใช้เวลาในการคำนวณนาน ซึ่งทำให้ไม่สามารถรายงานสิ่งกีดขวางได้ทันเวลาและจึงอาจเป็นอันตรายแก่ผู้ใช้งานระบบได้ เพื่อแก้ปัญหาดังกล่าวจึงเสนอวิธีการปรับปรุงซอร์สโค้ดของอัลกอริทึม P2P ให้สามารถประมวลผลแบบขนานได้โดยแยกการปรับปรุงซอร์สโค้ดออกเป็นสองรูปแบบคือ ลักษณะการคำนวณโดยใช้ข้อมูลทั้งแถว และลักษณะการคำนวณโดยใช้ข้อมูลทั้งคอลัมน์ และในที่สุดท้ายคือการนำข้อมูลจาก U-disparity มาแปลผลเพื่อให้ผู้พิการทางสายตาสามารถเข้าใจได้ง่ายขึ้น โดยผลจากการทดสอบระบบที่ออกแบบไว้นี้จะเสนอบทที่ 4 ต่อไป

## บทที่ 4

### ผลการทดสอบและการวิเคราะห์ผล

บทนี้นำเสนอผลการทดสอบโปรแกรมตามทีออกแบบไว้ในบทที่ 3 ประกอบไปด้วย การทดสอบโปรแกรมที่ได้ปรับปรุงให้สามารถประมวลผลแบบขนานได้ และผลการตรวจหาตำแหน่งและระยะทางของวัตถุ โดยใช้แผนผังดิสพาริตีที่ได้จากการประมวลผลภาพสเตอริโอด้วยอัลกอริทึม P2P โดยนำระบบที่ได้จัดสร้างขึ้นไปทดสอบในสภาพแวดล้อมต่างๆ ที่ผู้พิจารณาทางสายตาอาจพบเจอทั้งภายในและภายนอกอาคาร

#### 4.1. ความเร็วการประมวลผล

การทดสอบความเร็วของการประมวลผลของโปรแกรมแบบขนานในกระบวนการทำงานของอัลกอริทึม P2P เนื่องจากระบบตรวจจับสิ่งกีดขวางนี้ใช้อัลกอริทึม P2P เป็นหลัก ซึ่งอัลกอริทึมนี้ใช้เวลาประมวลผลต่อภาพหนึ่งชุดมากที่สุดเมื่อเทียบกับส่วนอื่นๆ ในระบบ อีกทั้งซอร์สโค้ดส่วนนี้สามารถปรับปรุงให้สามารถประมวลผลแบบขนานได้ สำหรับคอมพิวเตอร์ที่ใช้ในการทดสอบนั้นใช้คอมพิวเตอร์สองชุดคือ คอมพิวเตอร์ที่มีหน่วยประมวลผลแบบสองแกนซึ่งเป็นคอมพิวเตอร์ที่คาดว่าจะเป็นแพลตฟอร์มที่จะนำระบบนี้ไปใช้งานจริง และคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผลกลางสองหน่วย 8 แกน เพื่อหาแนวโน้มว่าในอนาคตเมื่อมีคอมพิวเตอร์พกพามีจำนวนหน่วยประมวลผลมากขึ้น เวลาในการประมวลผลจะเป็นอย่างไร สำหรับภาพวิดีโอที่ใช้ทดสอบนั้นจะใช้ภาพที่มีลักษณะดังแสดงไว้ดังรูปที่ 4-1 เป็นภาพตั้งต้น



(ก) ภาพซ้าย



(ข) ภาพขวา

รูปที่ 4-1 ภาพซ้ายและภาพขวาที่ใช้สำหรับการทดสอบ

#### 4.1.1. คุณลักษณะของคอมพิวเตอร์ที่ใช้ทดสอบ

คอมพิวเตอร์ที่ใช้ทดสอบการประมวลผลแบบขนานประกอบด้วยคอมพิวเตอร์ส่วนบุคคลที่มีหน่วยประมวลผลสองแกน และคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วย โดยมีคุณลักษณะดังต่อไปนี้

##### 4.1.1.1. คอมพิวเตอร์ที่มีหน่วยประมวลผลสองแกน

คอมพิวเตอร์ส่วนบุคคลที่มีหน่วยประมวลผลสองแกนที่ใช้ในการทดสอบนี้ใช้หน่วยประมวลผลกลางที่สามารถหาซื้อได้ทั่วไปในตระกูล Intel® Core™ 2 Dual โดยมีรายละเอียดฮาร์ดแวร์ดังต่อไปนี้

- Intel® Core™ 2 Dual 6320 1.86 GHz
- หน่วยความจำ RAM 1010.7 MB

เคอร์เนลและไลบรารีสำคัญๆ ที่ใช้ทดสอบการประมวลผลภาพสเตอริโอ

- Linux Kernel 2.6.32 สถาปัตยกรรม AMD 64
- MPICH 2.1.1 ใช้ gforker เป็นตัวจัดการโปรเซส และ ch3:nemesis เป็นช่องทางสื่อสาร
- OpenCV 2.0.0

##### 4.1.1.2. คอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วย

คอมพิวเตอร์ที่มีหน่วยประมวลผล 8 หน่วย ที่ใช้ทดสอบได้รับการสนับสนุนจากบริษัทอินเทลในตระกูล Intel® Xeon® โดยมีรายละเอียดฮาร์ดแวร์ดังต่อไปนี้

- Intel® Xeon® E5440 2.83 GHz
- แรม 8200 MB

เคอร์เนลและไลบรารีสำคัญๆ ที่ใช้ทดสอบการประมวลผลภาพสเตอริโอ

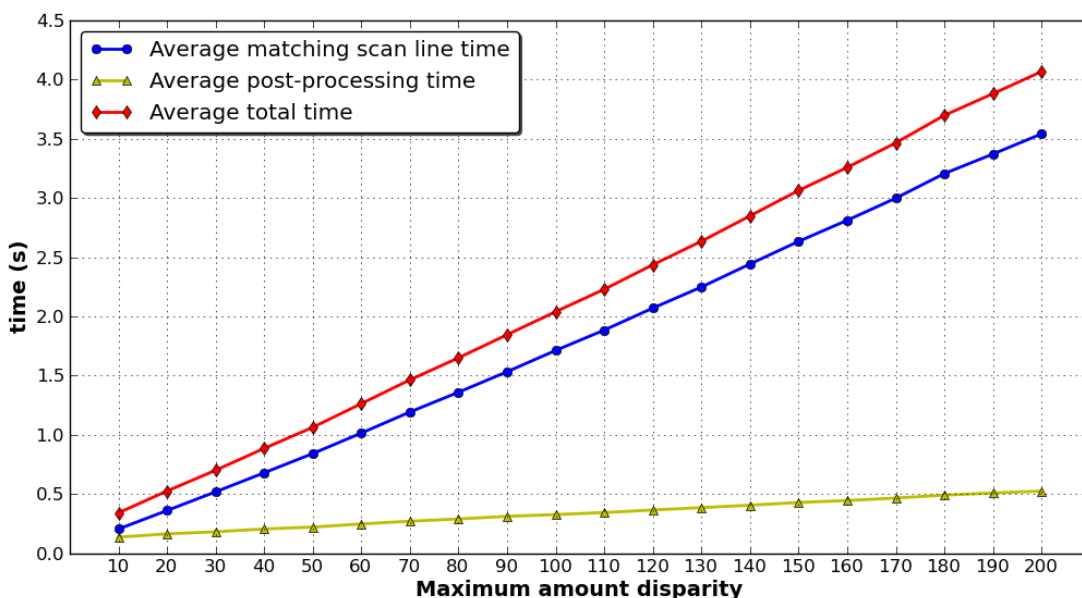
- Linux Kernel 2.6.18 สถาปัตยกรรม AMD 64
- MPICH 2.1.1 ใช้ gforker เป็นตัวจัดการโปรเซส และ ch3:nemesis เป็นช่องทางสื่อสาร
- OpenCV 2.0.0

#### 4.1.2. การประมวลผลอัลกอริทึม P2P แบบตามลำดับ

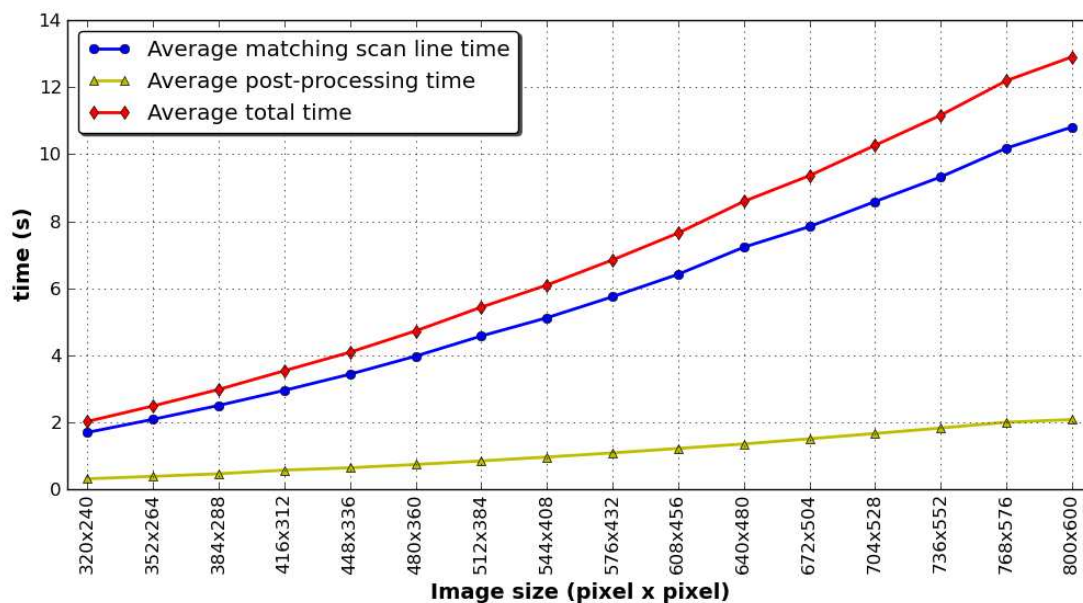
การประมวลผลโปรแกรมแบบตามลำดับที่นำเสนอในส่วนนี้ใช้ซอร์สโค้ดเดียวกับการโปรแกรมแบบขนานเพียงแต่นำฟังก์ชันที่ระบุว่าเป็นการโปรแกรมแบบขนานออก เพื่อให้เวลาของผลตอบสนองจากการประมวลผลของโปรแกรมแบบตามลำดับเป็นตัวเปรียบเทียบกับเวลาของผลตอบสนองจากการประมวลผลของโปรแกรมแบบขนาน

##### 4.1.2.1. เวลาการประมวลผลอัลกอริทึม P2P แบบตามลำดับบนคอมพิวเตอร์ที่มีหน่วยประมวลผลสองแกน

ในส่วนนี้นำเสนอเวลาในการประมวลผลอัลกอริทึม P2P แบบตามลำดับบนคอมพิวเตอร์ที่มีหน่วยประมวลผลสองแกนเพื่อใช้เปรียบเทียบกับเวลาของการประมวลผลแบบขนาน โดยที่รูปที่ 4-2 เป็นกราฟแสดงเวลาของการประมวลผลอัลกอริทึม P2P เมื่อเพิ่มจำนวนระดับคิสพาริตีสูงสุด โดยกำหนดขนาดของภาพไว้ที่ 320x240 พิกเซลและรูปที่ 4-3 เป็นกราฟแสดงเวลาของการประมวลผลอัลกอริทึม P2P เมื่อเปลี่ยนแปลงขนาดภาพโดยกำหนดจำนวนสูงสุดของคิสพาริตีไว้ที่ 100



รูปที่ 4-2 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ในลักษณะโปรแกรมแบบตามลำดับบนคอมพิวเตอร์ส่วนบุคคลที่มีหน่วยประมวลผลสองแกนเมื่อเปลี่ยนแปลงจำนวนระดับคิสพาริตีสูงสุด



รูปที่ 4-3 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ในลักษณะโปรแกรมแบบตามลำดับบนคอมพิวเตอร์ส่วนบุคคลที่มีหน่วยประมวลผลสองแกนเมื่อเปลี่ยนแปลงขนาดภาพ

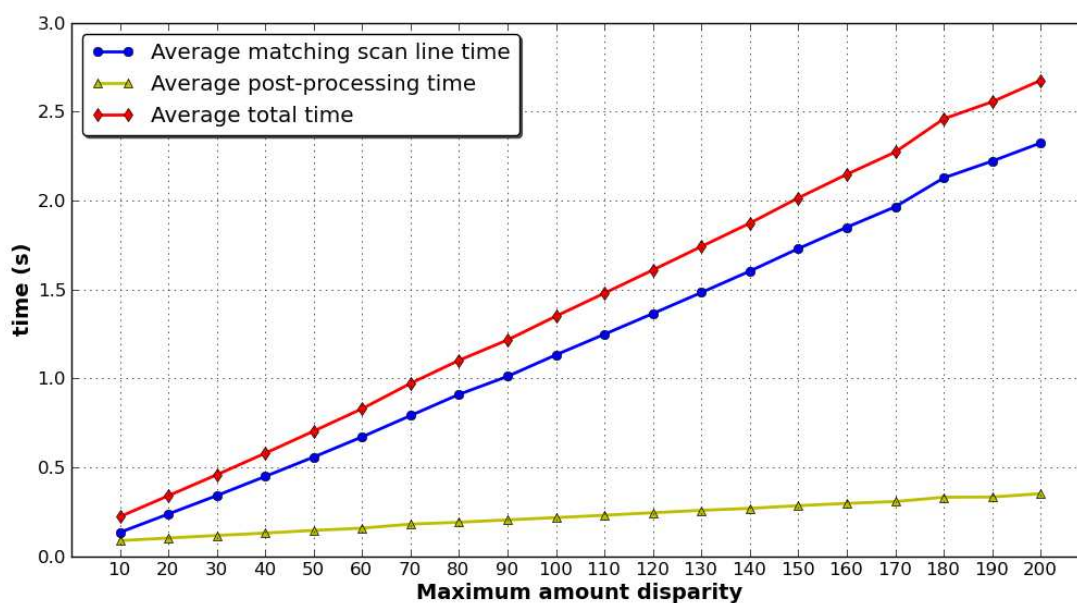
จากรูปที่ 4-2 เวลาในการประมวลผลของอัลกอริทึม P2P ที่มีจำนวนระดับดิสพาริตีสูงสุดเท่ากับ 100 มีค่าดังต่อไปนี้ เวลาเฉลี่ยของกระบวนการจับคู่พิกเซลในแต่ละแถวใช้เวลา 1.712 วินาที เวลาเฉลี่ยของกระบวนการประมวลผลขั้นปลายใช้เวลา 0.327 วินาทีเมื่อรวมเวลาทั้งสองกระบวนการเข้าด้วยกัน เวลาเฉลี่ยของทั้งอัลกอริทึมคือ 2.040 วินาที จะเห็นได้ว่าการจับคู่พิกเซลในแต่ละแถวใช้เวลาในการประมวลผลมากที่สุดคิดเป็นร้อยละ 83.92 ของเวลาในการประมวลผลอัลกอริทึม P2P ต่อภาพหนึ่งชุด

จากรูปที่ 4-2 และรูปที่ 4-3 ลักษณะกราฟที่ได้จะให้ผลในลักษณะเดียวกันคือ เมื่อจำนวนระดับดิสพาริตีสูงสุดหรือขนาดภาพเพิ่มขึ้นเวลาที่ใช้ในการประมวลผลจะเพิ่มขึ้นด้วย โดยมีลักษณะเป็นกราฟเส้นตรงที่มีความชันมีค่าเป็นบวก และเมื่อเปรียบเทียบกราฟที่แสดงในรูปที่ 3-8 และรูปที่ 4-2 จะพบว่าเส้นกราฟที่ได้ในรูปที่ 3-8 ไม่ราบเรียบเหมือนรูปที่ 4-2 เนื่องมาจากรูปที่ 3-8 เป็นการใช้งานซอร์สโค้ดดั้งเดิมซึ่งใช้อาร์เรย์สองมิติสำหรับเก็บข้อมูลรูปภาพซึ่งข้อมูลส่วนนี้มีปริมาณมาก เนื่องจากหนึ่งอีลีเมนต์ของอาร์เรย์เก็บข้อมูลภาพหนึ่งพิกเซล อีกทั้งการคำนวณในแต่ละแถวของภาพต้องสร้างอาร์เรย์สองมิติขนาดเท่ากับจำนวนพิกเซลในแต่ละแถวควบคู่กับจำนวนระดับดิสพาริตีสูงสุด อีกทั้งตัวแปรทั้งหมดที่ใช้งานเป็นตัวแปรท้องถิ่น (local variable) ส่งผลให้ในแต่ละครั้งที่มี

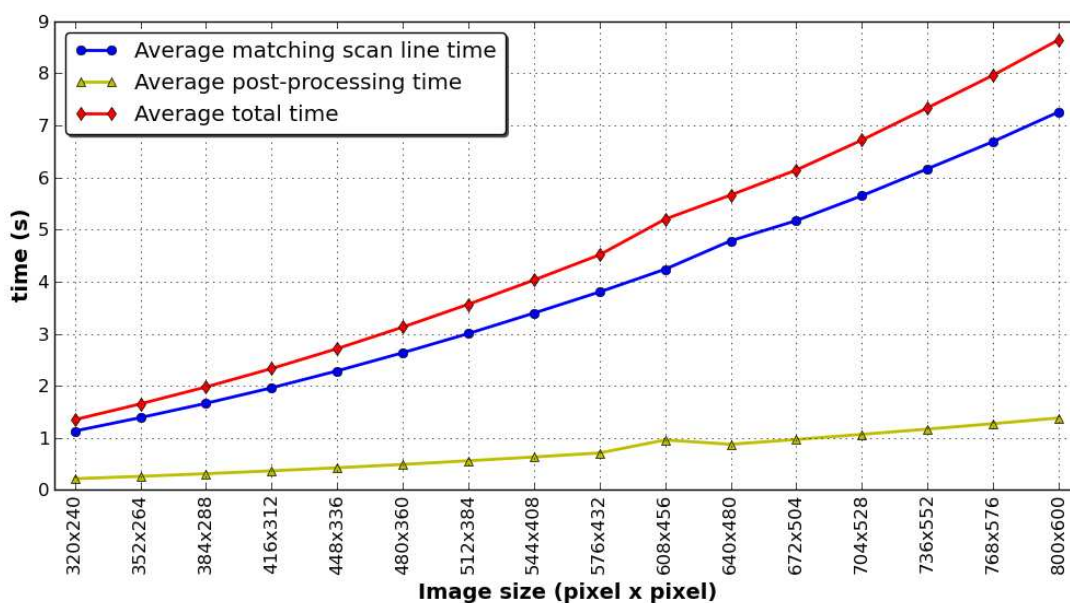
การเรียกใช้งานอัลกอริทึมจะต้องมีการกำหนดสภาพแวดล้อมใหม่ทุกครั้งที่เราเรียกใช้งานและทำลายตัวแปรทุกครั้งที่เราสิ้นสุดการทำงานในแต่ละรอบ ดังนั้นเวลาเฉลี่ยในแต่ละส่วนจึงไม่คงที่ตามไปด้วย

#### 4.1.2.2. เวลาการประมวลผลอัลกอริทึม P2P แบบตามลำดับบนคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วย

เวลาในการประมวลผลอัลกอริทึม P2P แบบตามลำดับบนคอมพิวเตอร์ที่มีหน่วยประมวลผล 8 หน่วยดังแสดงในรูปที่ 4-4 และ รูปที่ 4-5 เนื่องจากคอมพิวเตอร์สมรรถนะสูงที่ใช้ทดสอบมีประสิทธิภาพสูงกว่าคอมพิวเตอร์ส่วนบุคคลมาก เวลาในการประมวลผลจึงน้อยกว่า แต่มีลักษณะแนวโน้มของกราฟเพิ่มขึ้นในทิศทางเดียวกันเมื่อเพิ่มจำนวนระดับคิสพาริตีสูงสุด จากการทดสอบนี้เมื่อพิจารณาจำนวนระดับคิสพาริตีสูงสุดที่ 100 ภาพขนาด 320x240 พิกเซล จะใช้เวลาเฉลี่ยในการประมวลผลทั้งหมดอยู่ที่ 1.350 วินาที โดยกระบวนการจับคู่พิกเซลในแต่ละแถวใช้เวลาเฉลี่ย 1.133 วินาที และการประมวลผลขั้นปลายใช้เวลาเฉลี่ย 0.218 วินาที



รูปที่ 4-4 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ในลักษณะโปรแกรมแบบตามลำดับบนคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วย เมื่อเปลี่ยนแปลงจำนวนระดับคิสพาริตีสูงสุด



รูปที่ 4-5 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ในลักษณะโปรแกรมแบบตามลำดับบนคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วยเมื่อเปลี่ยนแปลงขนาดรูปภาพ

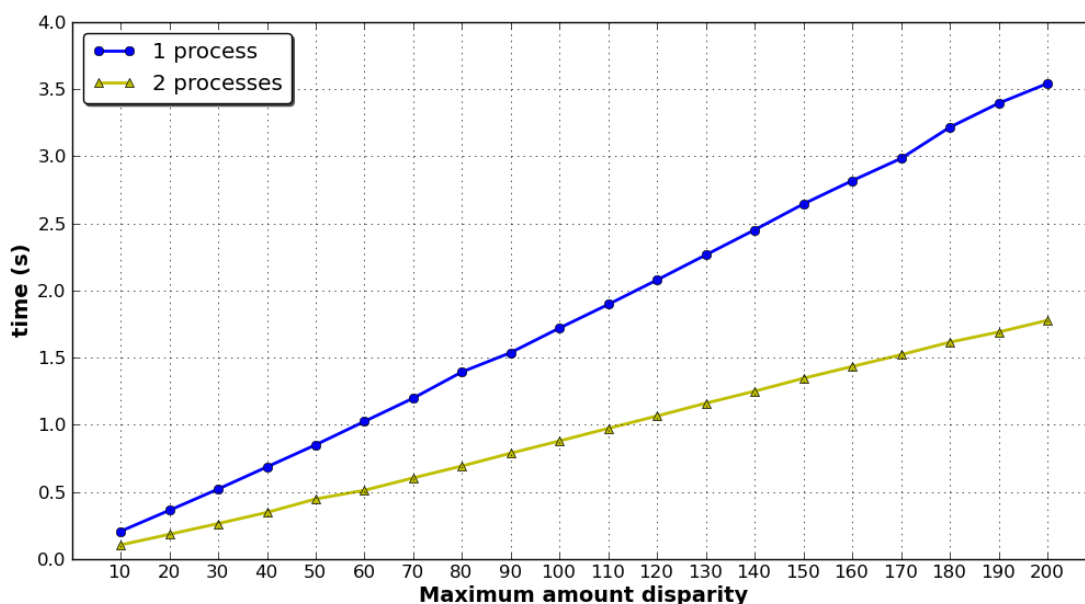
#### 4.1.3. การประมวลผลอัลกอริทึม P2P แบบขนานในกระบวนการจับคู่พิกเซลในแต่ละแถว

กระบวนการจับคู่พิกเซลในแต่ละแถวเป็นกระบวนการสำคัญสำหรับอัลกอริทึม P2P และใช้เวลาในการประมวลผลมากที่สุดเมื่อเทียบกับขั้นตอนอื่นๆ ของอัลกอริทึม P2P การปรับเปลี่ยนการเขียนโปรแกรมในส่วนนี้ด้วยการโปรแกรมแบบขนานจึงสามารถลดเวลาในการคำนวณลงได้มากตามไปด้วย การทดสอบซอร์สโค้ดที่ได้ออกแบบให้สามารถทำงานแบบขนานไว้ในบทที่ 3 หัวข้อที่ 3.3.1 แบ่งการทดสอบออกเป็นสองส่วน ส่วนแรกเป็นการทดสอบบนคอมพิวเตอร์ที่มีหน่วยประมวลผลสองแกนที่ใช้กันอย่างแพร่หลายในปัจจุบัน และคอมพิวเตอร์สมรรถนะสูงซึ่งมีหน่วยประมวลผลทั้งหมด 8 หน่วย เพื่อศึกษาแนวโน้มเวลาการตอบสนองการทำงานของโปรแกรมเมื่อมีการเพิ่มหน่วยประมวลผล

##### 4.1.3.1. เวลาการประมวลผลแบบขนานของอัลกอริทึม P2P ในส่วนของการจับคู่พิกเซลในแต่ละแถวบนคอมพิวเตอร์ที่มีหน่วยประมวลผลสองแกน

จากการทดสอบโปรแกรมแบบขนานบนคอมพิวเตอร์ส่วนบุคคลที่มีหน่วยประมวลผลสองแกนในส่วนการจับคู่พิกเซลในแต่ละแถว นั้นแสดงไว้ดังรูปที่ 4-6 ซึ่งเป็นการทดสอบโปรแกรมโดยแบ่งการประมวลผลออกเป็นสองชุดคือ นำโปรแกรมแบบขนานมาทดสอบการประมวลผลโดยใช้

โปรเซสเพียงโปรเซสเดียว และใช้สองโปรเซส ผลที่ได้จากการทดสอบคือ โปรแกรมที่ใช้เพียงโปรเซสเดียวนั้นใช้เวลาในการประมวลผลใกล้เคียงกับการโปรแกรมแบบตามลำดับ แต่เมื่อทดสอบโปรแกรมแบบขนานโดยใช้สองโปรเซสในการประมวลผลพบว่าสามารถลดเวลาผลตอบสนองลงได้

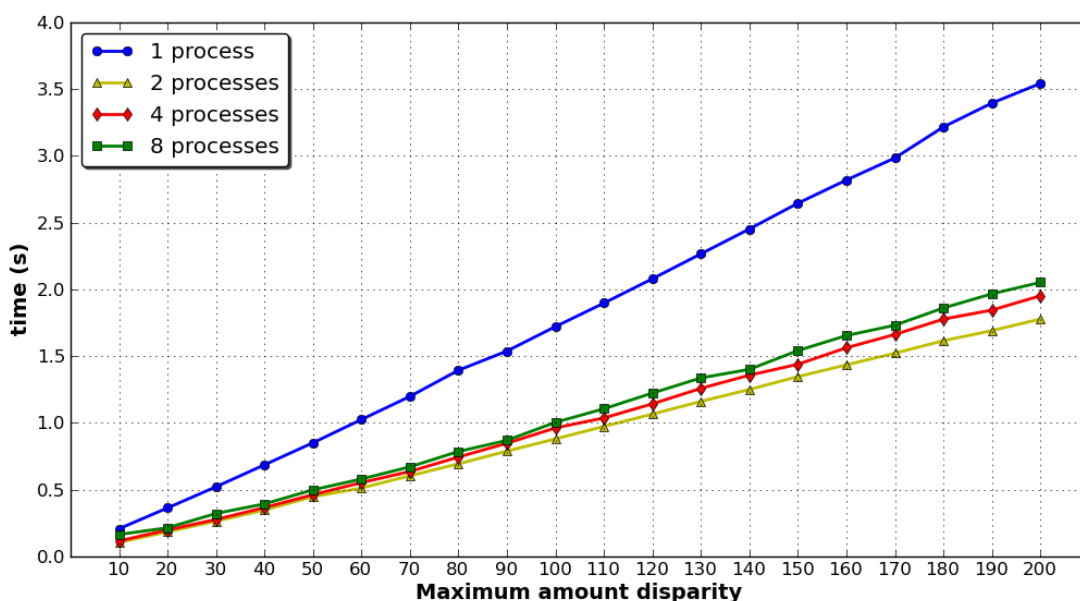


รูปที่ 4-6 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ส่วนการจับคู่พิกเซลในแต่ละแถวในลักษณะโปรแกรมแบบขนานบนคอมพิวเตอร์ส่วนบุคคลที่มีหน่วยประมวลผลสองแกนโดยใช้ 1 และ 2 โปรเซส

จากรูปที่ 4-6 เมื่อเปรียบเทียบเวลาในการประมวลผลของอัลกอริทึม P2P ในกระบวนการจับคู่พิกเซลในแต่ละแถวที่มีจำนวนระดับคิซพาริตีสูงสุดเท่ากับ 100 การประมวลผลโปรแกรมที่ใช้โปรเซสเดียวใช้ความเร็วเฉลี่ยที่ 1.720 วินาที และสองโปรเซสให้ความเร็วเฉลี่ยที่ 0.879 วินาที ซึ่งการประมวลผลแบบขนานโดยใช้สองโปรเซสในการประมวลผล สามารถลดเวลาลงได้ 0.841 วินาที แต่การเพิ่มโปรเซสที่มากเกินไปกว่าหน่วยประมวลผลที่มีอยู่จะส่งผลให้เวลาในการประมวลผลเพิ่มขึ้นจากที่ควรจะเป็นดังปรากฏในรูปที่ 4-7 เมื่อเพิ่มโปรเซสขึ้นเป็น 4 โปรเซส และ 8 โปรเซส ซึ่งจำนวนโปรเซสที่เพิ่มขึ้นนี้มีจำนวนมากกว่าจำนวนหน่วยประมวลผลคิดเป็น 2 และ 4 เท่าตามลำดับ ซึ่งส่งผลให้เวลาในการประมวลผลของกระบวนการนี้เพิ่มขึ้นตามไปด้วย โดยการประมวลผลที่ 4 โปรเซสใช้เวลา 0.961 วินาที และการประมวลผลที่ 8 โปรเซสใช้เวลา 1.003 วินาที



ดังนั้นหากงานที่แบ่งให้หน่วยประมวลผลมีมากเพียงพอต่อความสามารถของหน่วยประมวลผลแล้ว ไม่ควรใช้โปรเซสในการประมวลผลเกินกว่าจำนวนหน่วยประมวลผลที่มี มิฉะนั้นแทนที่เวลาในการคำนวณจะลดลงกลับเป็นการเพิ่มเวลาของการตอบสนองให้มากขึ้น เวลาที่เพิ่มขึ้นนี้เป็นโอเวอร์เฮดของ MPI ในการส่งข้อความเพื่อการแบ่งงานไปยัง โปรเซสต่างๆ อีกทั้งจำนวนโปรเซสที่ใช้นี้มีจำนวนมากกว่าหน่วยประมวลผล จึงเพิ่มเวลาในการสลับงานของแต่ละหน่วยประมวลผลเข้ามาด้วย

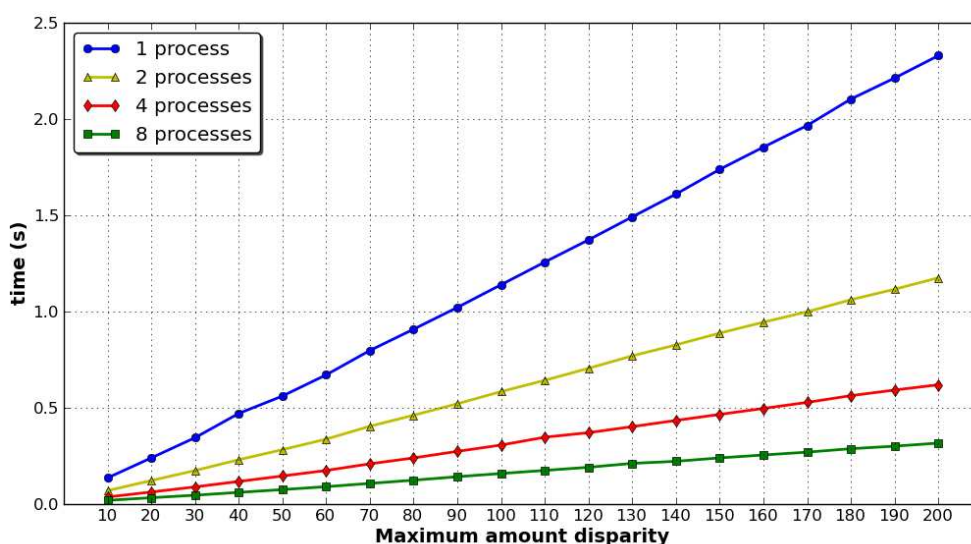


รูปที่ 4-7 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ในลักษณะโปรแกรมแบบขนานบนคอมพิวเตอร์ส่วนบุคคลที่มีหน่วยประมวลผล 2 หน่วยโดยใช้ 1, 2, 4 และ 8 โปรเซส

#### 4.1.3.2. เวลาการประมวลผลแบบขนานของอัลกอริทึม P2P ส่วนการจับคู่พิกเซลในแต่ละแฉบบคอมพิวเตอร์ที่มีหน่วยประมวลผล 8 หน่วย

สำหรับการประมวลผลแบบขนานของอัลกอริทึม P2P ในส่วนของกระบวนการจับคู่พิกเซลในแต่ละแฉบบคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วย ด้วยขนาดภาพ 320x240 พิกเซล โดยใช้จำนวนโปรเซสเป็น 1, 2, 4 และ 8 โปรเซส หากพิจารณาเวลาในการประมวลผลในกระบวนการนี้ที่จำนวนระดับดิสพาริตีสูงสุดที่ 100 จะใช้เวลาเฉลี่ย 1.138 วินาที 0.583 วินาที 0.305 วินาที และ 0.158 วินาทีตามลำดับ ซึ่งจากการทดสอบเวลาการตอบสนองของอัลกอริทึมจะลดลงตามจำนวนโปรเซสที่เพิ่มขึ้น โดยเวลาในการประมวลผลที่ลดลงจากการ

ประมวลผลโดยใช้เพียงโปรเซสเดียวเมื่อเพิ่มโปรเซสเป็น 2, 4 และ 8 คือ 0.555 วินาที 0.833 วินาที 0.980 วินาทีตามลำดับ สำหรับผลตอบสนองที่ประมวลผลด้วยจำนวนระดับคิสพาริตีสูงสุดอื่นๆ นั้นแสดงไว้ดังรูปที่ 4-8

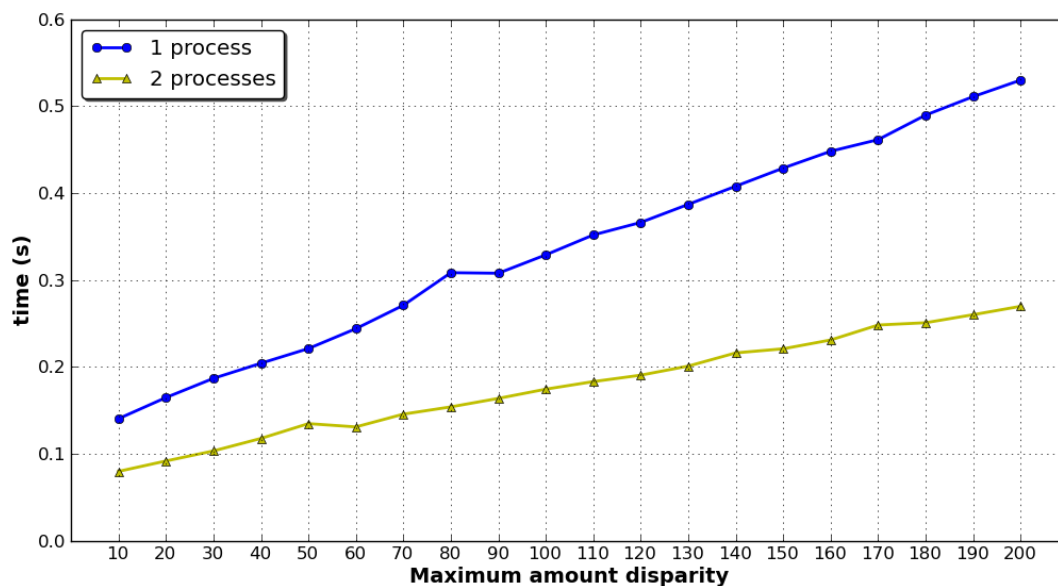


รูปที่ 4-8 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ส่วนของการจับคู่พิกเซลในแต่ละแถว ในลักษณะโปรแกรมแบบขนานบนคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วย ด้วยภาพขนาด 320x240 พิกเซล โดยใช้ 1, 2, 4 และ 8 โปรเซส

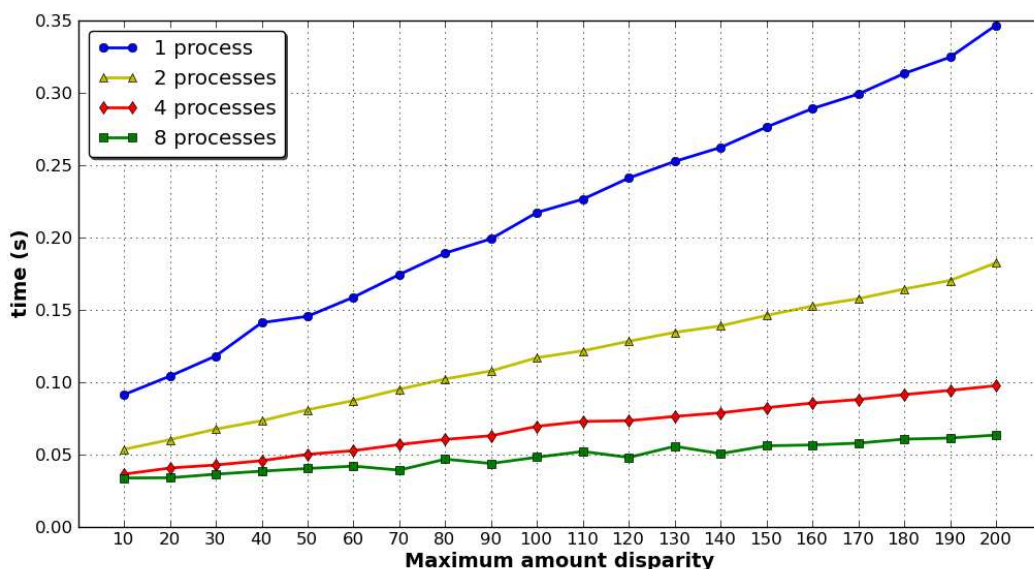
#### 4.1.4. การประมวลผลอัลกอริทึม P2P แบบขนานกระบวนการประมวลผลขั้นปลาย

อัลกอริทึม P2P ใช้เวลาในการประมวลผลกระบวนการประมวลผลขั้นปลายบนคอมพิวเตอร์ส่วนบุคคลที่มีหน่วยประมวลผลสองแกนคิดเป็นร้อยละ 16.03 จากเวลาในการประมวลผลอัลกอริทึมแบบตามลำดับ ซึ่งใช้เวลาเฉลี่ยเพียง 0.327 วินาทีจากเวลาเฉลี่ยทั้งหมด 2.040 วินาทีต่อภาพ เมื่อประมวลผลด้วยจำนวนระดับคิสพาริตีสูงสุดที่ 100 ภาพขนาด 320x240 พิกเซล จากการปรับเปลี่ยนมาใช้ในการประมวลผลแบบขนานให้เวลาตอบสนองที่ 0.174 วินาที ซึ่งเร็วขึ้น 0.153 วินาที สำหรับจำนวนระดับคิสพาริตีสูงสุดอื่นๆ แสดงไว้ดังรูปที่ 4-9 ส่วนคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วยเวลาที่ใช้ในการประมวลผลที่จำนวนระดับคิสพาริตีสูงสุดต่างๆ แสดงไว้ในรูปที่ 4-10 หากพิจารณารูปที่ 4-10 กราฟเวลาของการประมวลผลโดยใช้ 4 โปรเซสและกราฟการประมวลผลโดยใช้ 8 โปรเซส จะสังเกตได้ว่าเวลาในส่วนนี้จะลดลงน้อยกว่าในช่วงอื่นๆ เนื่องจากในส่วนของการประมวลผลขั้นปลายมีการส่งผ่านข้อความเพื่อติดต่อระหว่างโปรเซสบ่อยครั้งในขณะที่มีจำนวนหน่วยประมวลผลมากขึ้น แต่จำนวนข้อมูลที่ใช้ประมวลผล

น้อยลงอาจส่งผลให้มีเวลาโอเวอร์เฮดมากขึ้นด้วย เวลาที่ใช้ในการประมวลผลของ 8 โพรเซสจึงมีอัตราการลดลงน้อยตามไปด้วย



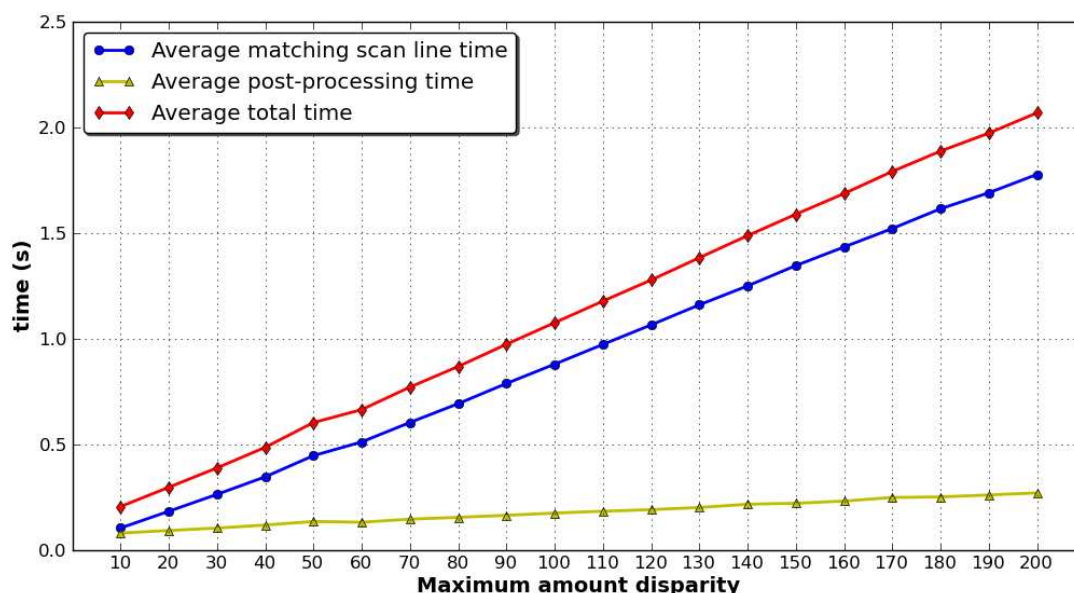
รูปที่ 4-9 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ส่วนของการประมวลผลขั้นปลายในลักษณะโปรแกรมแบบขนานบนคอมพิวเตอร์ที่มีหน่วยประมวลผล 2 แกน ด้วยภาพขนาด 320x240 พิกเซล จำนวน 1 และ 2 โพรเซส



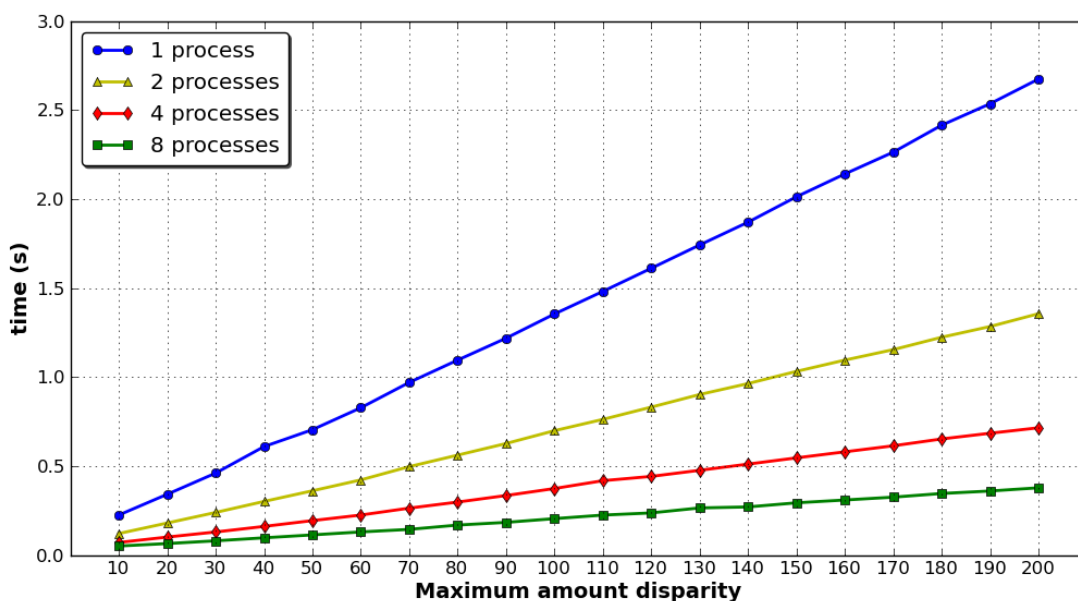
รูปที่ 4-10 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ส่วนของการประมวลผลขั้นปลายในโปรแกรมแบบขนานบนคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วย ด้วยภาพขนาด 320x240 พิกเซล จำนวน 1, 2, 4 และ 8 โพรเซส

#### 4.1.5. วิเคราะห์ผลการประมวลผลอัลกอริทึม P2P แบบขนาน

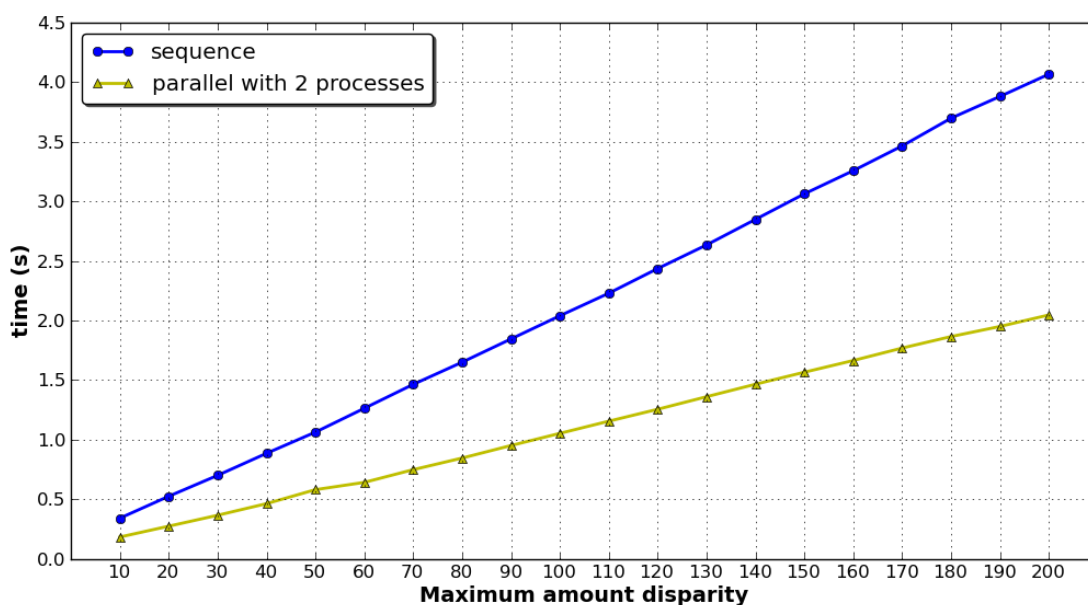
จากการทดสอบการประมวลผลแบบขนานที่ได้นำเสนอในหัวข้อ 4.1.3 และ 4.1.4 เมื่อนำเวลาของการประมวลผลในแต่ละกระบวนการบนคอมพิวเตอร์ที่มีหน่วยประมวลผลแบบสองแกนมารวมกันจะได้เวลาเฉลี่ยของอัลกอริทึม P2P ดังรูปที่ 4-11 และคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วยจะได้เวลาเฉลี่ยของอัลกอริทึม P2P ดังรูปที่ 4-12 เพื่อให้ตรงกับจุดประสงค์ที่จะนำระบบนี้ไปใช้งานบนคอมพิวเตอร์ที่มีหน่วยประมวลผลแบบสองแกน และสังเกตแนวโน้มของผลตอบสนองเมื่อมีจำนวนหน่วยประมวลผลเพิ่มขึ้น การวิเคราะห์ประสิทธิภาพจะวิเคราะห์บนคอมพิวเตอร์ทั้งสองแบบ เมื่อสร้างกราฟเปรียบเทียบเวลาในการประมวลผลของอัลกอริทึม P2P บนคอมพิวเตอร์ที่มีหน่วยประมวลผลแบบสองแกนระหว่างการประมวลผลแบบตามลำดับและการประมวลผลแบบขนานที่ใช้สองโปรเซสในการประมวลผลจะได้ดังรูปที่ 4-13 เมื่อพิจารณาเวลาในการประมวลผลโปรแกรมในแต่ละกระบวนการของอัลกอริทึม P2P บนคอมพิวเตอร์ที่มีหน่วยประมวลผลแบบสองแกนที่จำนวนระดับคิสพาริตีสูงสุดที่ 100 ภาพขนาด 320x240 พิกเซล จะมีค่าดังแสดงไว้ในตารางที่ 4-1 ซึ่งข้อมูลนี้จะใช้ในการวิเคราะห์ประสิทธิภาพของการประมวลผลแบบขนานต่อไป



รูปที่ 4-11 เวลาเฉลี่ยของการประมวลผลอัลกอริทึม P2P ในลักษณะโปรแกรมแบบขนานบนคอมพิวเตอร์ส่วนบุคคลที่มีหน่วยประมวลผลสองแกน ด้วยขนาดภาพ 320x240 พิกเซล จำนวน 2 โปรเซส



รูปที่ 4-12 เวลาเฉลี่ยรวมของการประมวลผลอัลกอริทึม P2P ลักษณะโปรแกรมแบบขนานบนคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วย ด้วยภาพขนาด 320x240 พิกเซล โดยใช้ 1, 2, 4 และ 8 โพรเซส



รูปที่ 4-13 เวลาเฉลี่ยทั้งหมดของการประมวลผลอัลกอริทึม P2P ในลักษณะโปรแกรมแบบตามลำดับ และโปรแกรมแบบขนานที่ใช้ 2 โพรเซสในการประมวลผล บนคอมพิวเตอร์ที่มีหน่วยประมวลผลสองแกน ด้วยภาพขนาด 320x240 พิกเซล

ตารางที่ 4-1 เวลาในการประมวลผลโปรแกรมในแต่ละกระบวนการของอัลกอริทึม P2P บนคอมพิวเตอร์ที่มีหน่วยประมวลผลแบบสองแกนที่จำนวนระดับดิสพาริตีสูงสุด 100 ภาพขนาด 320x240 พิกเซล

กระบวนการ	เวลาในการประมวลผล (วินาที)	
	แบบตามลำดับ	แบบขนานจำนวนสองโปรเซส
การจับคู่พิกเซลในแต่ละแถว	1.712480	0.878809
การประมวลผลขั้นปลาย	0.327071	0.174473
เวลารวมทั้งหมด	2.039551	1.053282

การวัดประสิทธิภาพของการประมวลผลอัลกอริทึม P2P แบบขนานสามารถหาค่าสปีดอัปและค่าประสิทธิภาพการประมวลผลโดยใช้ข้อมูลจากตารางที่ 4-1 เวลาในการประมวลผลของโปรแกรมในแต่ละกระบวนการของอัลกอริทึม P2P บนคอมพิวเตอร์ที่มีหน่วยประมวลผลแบบสองแกนที่จำนวนระดับดิสพาริตีสูงสุด 100 ภาพขนาด 320x240 พิกเซลได้ดังรูปที่ 4-2

ตารางที่ 4-2 ค่าสปีดอัปและค่าประสิทธิภาพการประมวลผลในแต่ละกระบวนการของอัลกอริทึม P2P บนคอมพิวเตอร์ที่มีหน่วยประมวลผลแบบสองแกนโดยใช้ข้อมูลจากตารางที่ 4-1

กระบวนการ	ค่าสปีดอัป	ค่าประสิทธิภาพการประมวลผล
การจับคู่พิกเซลในแต่ละแถว	1.949	0.975
การประมวลผลขั้นปลาย	1.875	0.938
เวลารวมทั้งหมด	1.936	0.968

จากตารางที่ 4-2 เมื่อพิจารณาค่าสปีดอัปของเวลารวมทั้งอัลกอริทึมมีค่าเท่ากับ 1.936 ซึ่งมีค่าเข้าใกล้จำนวนหน่วยประมวลผลคือสองหน่วย ส่วนค่าประสิทธิภาพการประมวลผลเท่ากับ 0.968 ซึ่งมีค่าเข้าใกล้ 1 แสดงว่าอัลกอริทึมแบบขนานที่ออกแบบไว้สามารถใช้งานได้เต็มประสิทธิภาพการประมวลผลของหน่วยประมวลผลแล้ว

ตารางที่ 4-3 เวลาในการประมวลผลโปรแกรมในแต่ละกระบวนการของอัลกอริทึม P2P บนคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผลแบบ 8 หน่วย ที่จำนวนระดับดิสพาริตีสูงสุด 100 ภาพขนาด 320x240 พิกเซล

กระบวนการ	เวลาในการประมวลผล (วินาที)			
	แบบตามลำดับ	การประมวลผลแบบขนาน		
		2 โพรเซส	4 โพรเซส	8 โพรเซส
การจับคู่พิกเซลในแต่ละแถว	1.132692	0.582908	0.305413	0.157589
การประมวลผลขั้นปลาย	0.217673	0.116806	0.069353	0.048038
เวลารวมทั้งหมด	1.350364	0.699715	0.374766	0.205627

ตารางที่ 4-4 ค่าค่าสปีดอัปและค่าประสิทธิภาพการประมวลผลในแต่ละกระบวนการของอัลกอริทึม P2P บนคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผลแบบ 8 หน่วย โดยใช้ข้อมูลจากตารางที่ 4-3

กระบวนการ	ค่าสปีดอัป			ค่าประสิทธิภาพการประมวลผล		
	2	4	8	2	4	8
	โพรเซส	โพรเซส	โพรเซส	โพรเซส	โพรเซส	โพรเซส
การจับคู่พิกเซลในแต่ละแถว	1.943	3.709	7.188	0.972	0.927	0.899
การประมวลผลขั้นปลาย	1.864	3.139	4.531	0.932	0.785	0.566
เวลารวมทั้งหมด	1.930	3.603	6.567	0.965	0.901	0.821

จากตารางที่ 4-3 เป็นการแสดงเวลาในการตอบสนองของอัลกอริทึม P2P เมื่อประมวลผลบนคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผล 8 หน่วย ในแต่ละกระบวนการใช้จำนวนโพรเซสในการประมวลผล 2, 4 และ 8 โพรเซสตามลำดับ เมื่อเพิ่มจำนวนโพรเซสขึ้นจะสามารถลดเวลาการตอบสนองของอัลกอริทึม P2P ลงได้จากเดิม โดยเวลารวมของอัลกอริทึม P2P ในการประมวลผลแบบตามลำดับอยู่ที่ 1.350 วินาที แต่เมื่อเปลี่ยนเป็นการประมวลผลแบบขนานโดยใช้จำนวนโพรเซสเป็น 2, 4 และ 8 จะใช้เวลา 0.610, 0.374 และ 0.206 ตามลำดับ ซึ่งข้อมูลเวลาการ

ตอบสนองจากตารางที่ 4-3 นี้สามารถนำไปหาค่าสปีดอัปและค่าประสิทธิภาพการประมวลผลได้ดังตารางที่ 4-4

จากตารางที่ 4-3 และตารางที่ 4-4 พบว่าเมื่อมีคอมพิวเตอร์ที่มีหน่วยประมวลผลมากขึ้น ค่าสปีดอัปจะให้ค่าที่ใกล้เคียงจำนวนหน่วยประมวลผลสำหรับการใช้งาน 2 และ 4 โพรเซส แต่สำหรับ 8 โพรเซสนั้นค่าสปีดอัปจะลดลงเหลือเพียง 6.567 เท่านั้น เนื่องจากข้อมูลขนาด 320x240 พิกเซล เมื่อแบ่งออกเป็น 8 ส่วน จะเหลือข้อมูลเพียง 320x30 พิกเซลเท่านั้น ส่งผลให้แต่ละหน่วยประมวลผลมีข้อมูลในการคำนวณน้อยลงและเพิ่มปริมาณการรับส่งข้อมูลบ่อยขึ้นนั่นเอง แต่เมื่อพิจารณาค่าประสิทธิภาพแล้วการแบ่งงานที่ได้ทำไว้ยังให้ค่าประสิทธิภาพที่ดี

## 4.2. การประมาณค่าระยะทางและตรวจสอบตำแหน่งของวัตถุ

ส่วนนี้เป็นการนำเสนอผลการทดสอบระบบที่ได้ออกแบบไว้ในหัวข้อ 3.4 ซึ่งเป็นส่วนที่นำสมการที่ (13) มาใช้เพื่อการประมาณระยะทางของสิ่งกีดขวางสำหรับการเดินทางของผู้พิการทางสายตา โดยทดสอบในสภาพแวดล้อมจริงที่ใช้งานในชีวิตประจำวัน ในส่วนถัดมานำเสนอสภาพแวดล้อมที่ไม่สามารถตรวจจับสิ่งกีดขวางได้

### 4.2.1. สภาพแวดล้อมจริงที่ทดสอบ



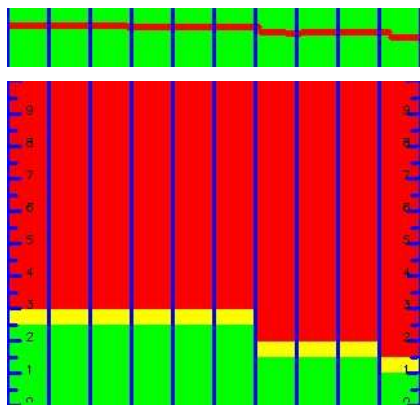
ผลจากการทดสอบนี้แสดงไว้ในตารางที่ 4-5 และตารางที่ 4-6 โดยเลือกสถานที่ที่มักใช้เดินทางในชีวิตประจำวัน ทั้งนี้ทดสอบกับทางเดินทั้งภายในและภายนอกอาคารในสภาพแสงที่แตกต่างกัน โดยมีรายละเอียดสถานที่ดังตารางที่ 4-5 และตารางที่ 4-6 ต่อไปนี้



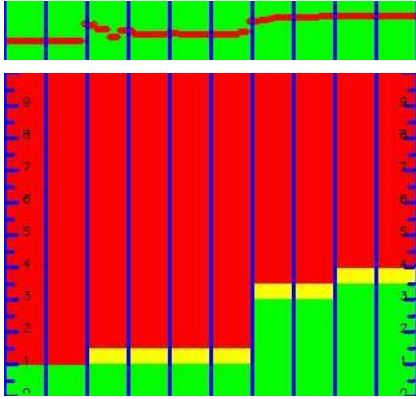


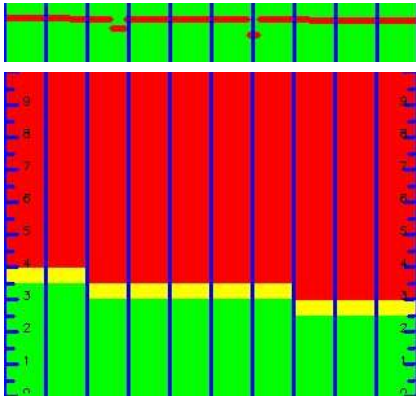



- ตารางที่ 4-5 รูปในแต่ละสถานที่ตัวอย่างที่ใช้ทดสอบหาตำแหน่งของวัตถุในสภาพแสงกลางวันภายนอกอาคาร
  - รูปที่ 1 เป็นลานจอดรถ โดยที่สภาพแสงมีเงาปนแดด ความสว่าง 3870- 58700 lux
  - รูปที่ 2 เป็นถนนที่มีรถจอดอยู่ด้านข้าง โดยที่สภาพแสงมีเงาปนแดด ความสว่าง 989-5500 lux
  - รูปที่ 3 ถนนทางขึ้นเนิน โดยที่สภาพแสงมีเงาปนแดด ความสว่าง 3810-52800 lux
  - รูปที่ 4 ถนนทางลงเนิน โดยที่สภาพแสงมีเงาปนแดด ความสว่าง 3810-52800 lux
  - รูปที่ 5 ป้ายสัญญาณ โดยที่สภาพแสงมีเงาปนแดด ความสว่าง 3810-52800 lux
  - รูปที่ 6 ทางเดินในสวน ความสว่าง 1064-3730 lux
  - รูปที่ 7 ทางเดินที่มีหลังคา ความสว่าง 876-17560 lux



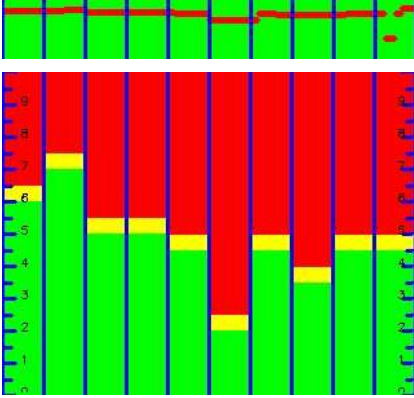


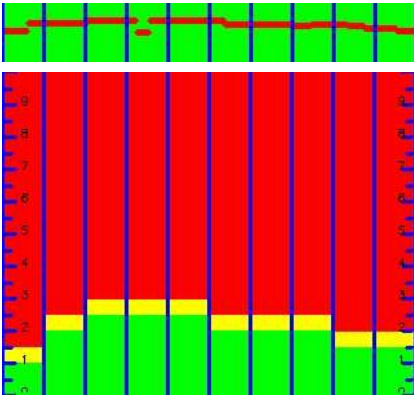


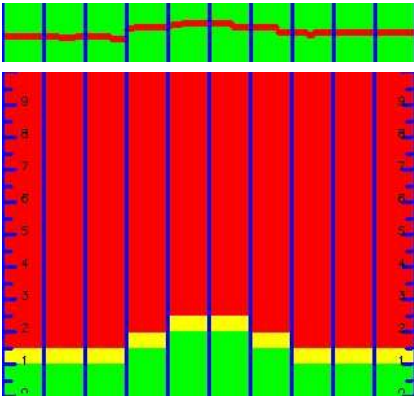



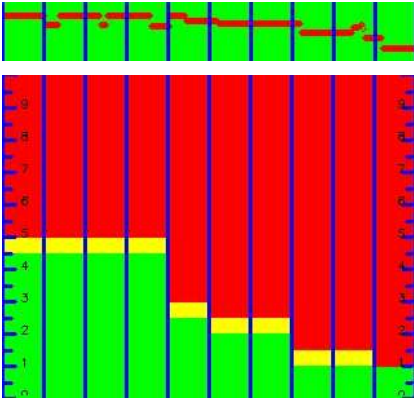


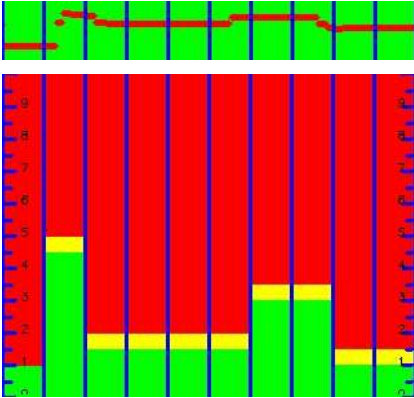
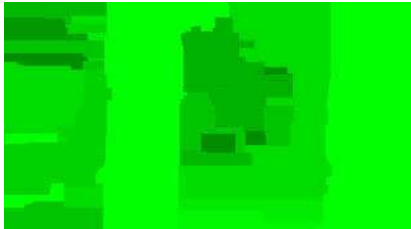
- รูปที่ 8 ทางเดินบริเวณอาคารมีช่องทางสามารถเดินลอดผ่านได้ ความสว่าง 1077-1377 lux
- รูปที่ 9 ทางเดินบริเวณภายนอกอาคารมีเสาอาคารสามารถเดินผ่านได้ ความสว่าง 1077-1377 lux
- ตารางที่ 4-6 รูปในแต่ละสถานที่ตัวอย่างที่ใช้ทดสอบหาตำแหน่งของวัตถุในสภาพแสงกลางวันภายในอาคาร
  - รูปที่ 1 ห้องทำงานที่มีผนังกันห้องมีช่องทางเดิน ความสว่าง 140-300 lux
  - รูปที่ 2 บริเวณห้องทำงานมีทางเดินแคบ ความสว่าง 140-300 lux
  - รูปที่ 3 บริเวณห้องทำงานที่ตู้และขาเก้าอี้มีช่องทางเดินเล็กน้อย ความสว่าง 140-300 lux
  - รูปที่ 4 ทางเดินภายในอาคารความสว่าง 2200-5510 lux
  - รูปที่ 5 ทางเดินภายในอาคารมีชั้นวางรองเท้าอยู่ด้านซ้าย ความสว่าง 2200-5510 lux
  - รูปที่ 6 ห้องกระจกใสภายในอาคารที่มีม่านปิด ความสว่าง 2200-5510 lux
  - รูปที่ 7 บันไดทางลงภายในอาคาร ความสว่าง 2200-5510 lux
  - รูปที่ 8 บันไดทางขึ้นภายในอาคาร ความสว่าง 2200-5510 lux

ตารางที่ 4-5 รูปในแต่ละสถานที่ตัวอย่างที่ใช้ทดสอบหาตำแหน่งของวัตถุในสภาพแสงกลางวันภายนอกอาคาร



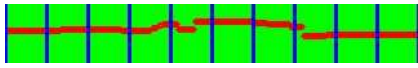
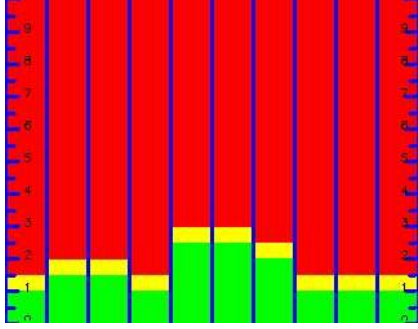


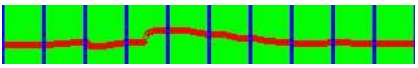
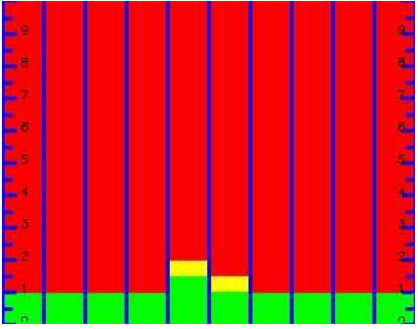
รูปที่	ความสว่าง	รูปด้านซ้าย	ตำแหน่งของวัตถุ
		รูปศิลปะาริที่	
1	3870-58700 lux	 	


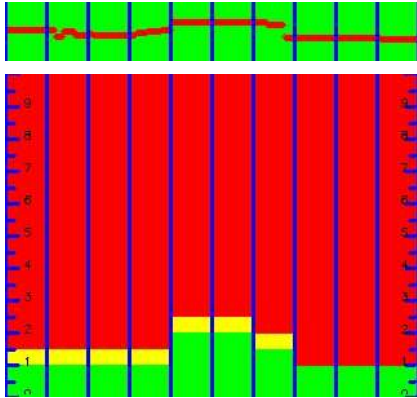


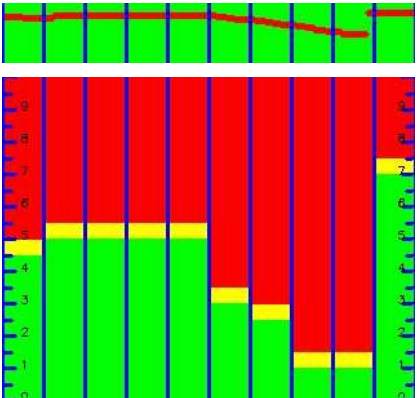


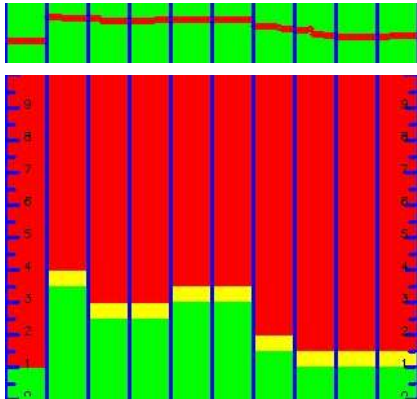

รูปที่	ความสว่าง	รูปด้านซ้าย	ตำแหน่งของวัตถุ
		รูปดิสพาริตี	
2	989-5500 lux	 	
3	3810-52800 lux	 	
4	3810-52800 lux	 	


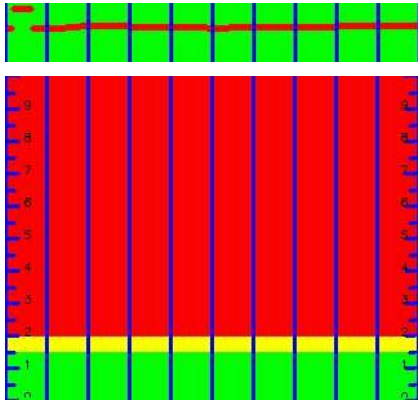


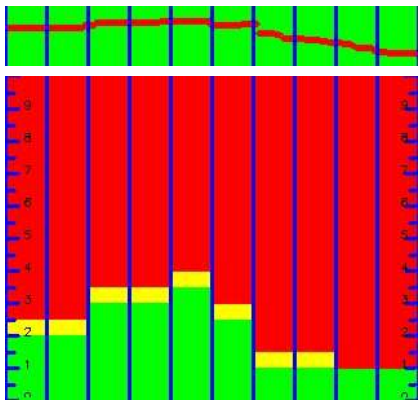


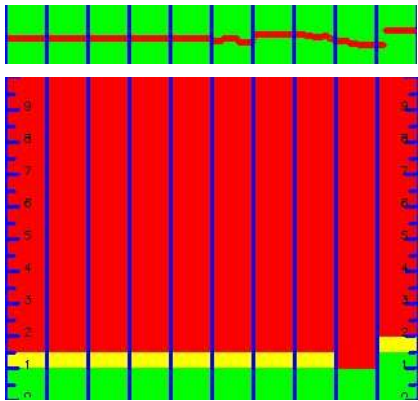

รูปที่	ความสว่าง	รูปด้านซ้าย	ตำแหน่งของวัตถุ
		รูปดิสพาริตี	
5	1600-3730 lux	 	
6	1064-3730 lux	 	
7	876-17560 lux	 	

รูปที่	ความสว่าง	รูปด้านซ้าย	ตำแหน่งของวัตถุ
		รูปศิลปะวิธี	
8	1077-1377 lux		
			
9	1077-1377 lux		
			

ตารางที่ 4-6 รูปในแต่ละสถานที่ตัวอย่างที่ใช้ทดสอบหาตำแหน่งของวัตถุในสภาพแสงกลางวัน  
ภายในอาคาร

รูปที่	ความสว่าง	รูปด้านซ้าย	ตำแหน่งของวัตถุ
		รูปศิลปะวิธี	
1	140-300 lux	 	 
2	140-300 lux	 	 

รูปที่	ความสว่าง	รูปด้านซ้าย	ตำแหน่งของวัตถุ
		รูปศิลปะาริที่	
3	140-300 lux		
			
4	2200-5510 lux		
			
5	2200-5510 lux		
			

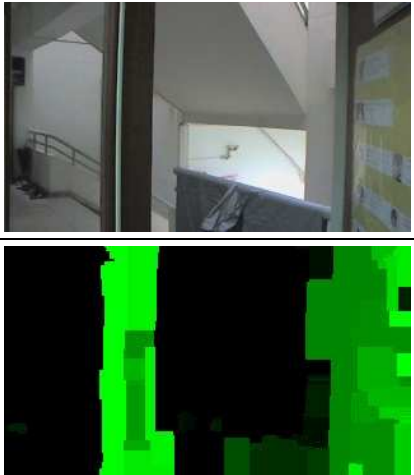
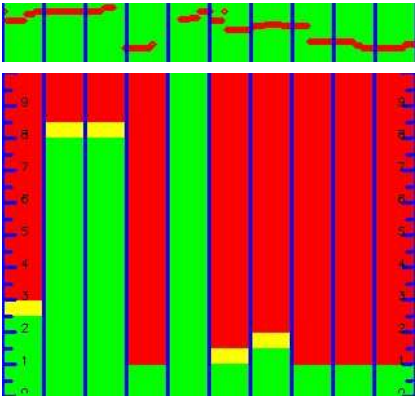
รูปที่	ความสว่าง	รูปด้านซ้าย	ตำแหน่งของวัตถุ
		รูปดิสพาริตี	
6	2200-5510 lux		
			
7	2200-5510 lux		
			
8	2200-5510 lux		
			

#### 4.2.2. สภาพแวดล้อมที่เป็นปัญหาสำหรับการตรวจสอบ



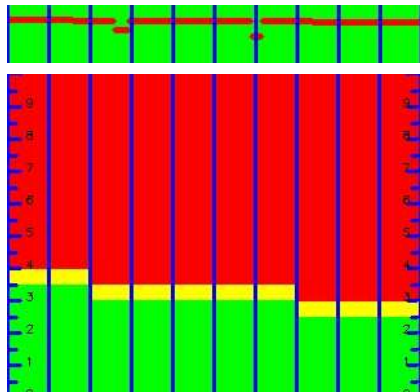





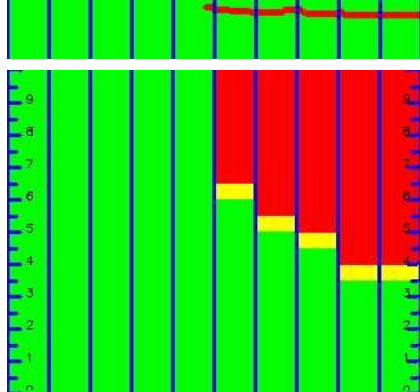
จากการทดสอบสถานที่ต่างๆ มีวัตถุหรือสภาพแวดล้อมบางคุณลักษณะที่อัลกอริทึม P2P ไม่สามารถหาติสพาริติได้ ดังแสดงไว้ในตารางที่ 4-7 โดยมีรายละเอียดดังต่อไปนี้

- รูปที่ 1 วัตถุที่มีลักษณะโปร่งใสโดยสามารถบอกเห็นภาพข้างนอกได้โดยไม่มีเงาสะท้อน แต่กระจกที่มีมันติดตั้งไว้ด้านหลังดังในตารางที่ 4-6 รูปที่ 6 อัลกอริทึมสามารถหาติสพาริติได้
- รูปที่ 2 บริเวณถนนทั้งถนนขึ้นเนินและลงเนิน เนื่องจากอัลกอริทึม P2P สามารถหาติสพาริติบริเวณถนนได้ ดังนั้นหากถนนที่ไม่สิ่งกีดขวางระบบที่สร้างขึ้นจะยังตีความว่ามีสิ่งกีดขวางอยู่ เพียงแต่จะอยู่ไกลออกไปเท่านั้น ดังนั้นผู้พิการยังสามารถเดินทางได้อยู่
- รูปที่ 3 บริเวณที่มีกิ่งไม้หรือวัตถุยื่นออกมา ถ้าหากกิ่งไม้หรือวัตถุเหล่านั้นอยู่ใกล้ระบบจะตีความว่ามีวัตถุใกล้ตัวผู้ใช้งาน ถึงแม้ว่าวัตถุที่มีลักษณะดังกล่าวผู้ใช้งานจะสามารถเดินผ่านไป ได้ แต่ทัศนียภาพลักษณะนี้สามารถแก้ไขได้โดยการปรับอัตราส่วนของขนาดภาพใหม่ให้ความยาวลดลง เมื่อภาพที่ได้มีความยาวลดลงภาพกิ่งไม้จะหายไป
- รูปที่ 4 สภาพแสงน้อยโดยบริเวณที่อัลกอริทึมจับภาพได้มีความสว่างประมาณ 82 lux ส่วนบริเวณอื่นอัลกอริทึม P2P ไม่สามารถหาติสพาริติได้

ตารางที่ 4-7 สภาพแวดล้อมที่มีปัญหาในการตรวจจับด้วยอัลกอริทึม P2P

รูปที่	ความเข้มแสง	รูปด้านซ้าย	ตำแหน่งของวัตถุ
		รูปติสพาริติ	
1	140-300 lux		





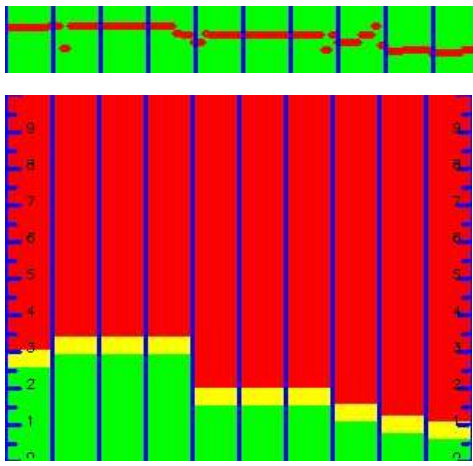
รูปที่	ความเข้มแสง	รูปด้านซ้าย	ตำแหน่งของวัตถุ
		รูปดิสพาริตี	
2	3810-52800 lux	 	
3	989-55000 lux	 	
4	1-82 lux	 	

#### 4.2.3. การทดสอบปรับอัตราส่วนลักษณะ


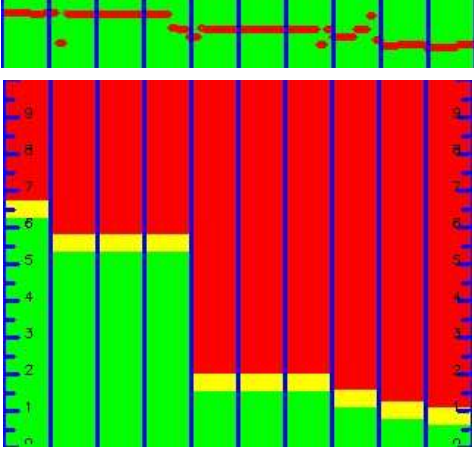

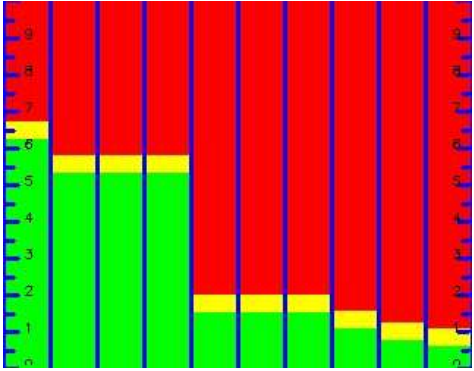

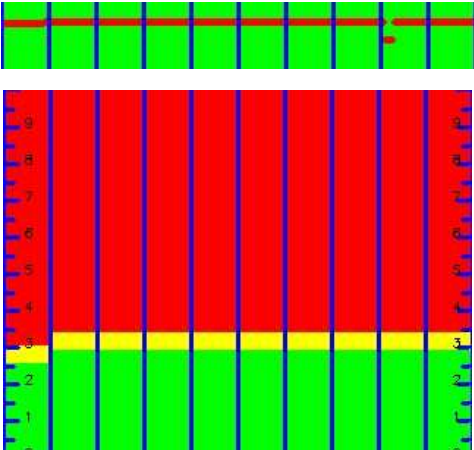

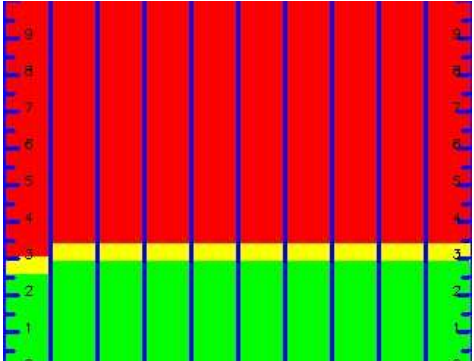
ส่วนนี้เป็นการทดสอบการปรับเปลี่ยนอัตราส่วนลักษณะของภาพเพื่อช่วยให้ผู้ใช้งานระบบสามารถทำความเข้าใจสภาพแวดล้อมได้ดีขึ้น โดยแบ่งอัตราส่วนลักษณะดังต่อไปนี้

- อัตราส่วน 4:3 ภาพขนาด 320x240 พิกเซล ดังตารางที่ 4-8
- อัตราส่วน 16:9 ภาพขนาด 320x180 พิกเซล ดังตารางที่ 4-9
- อัตราส่วน 8:3 ภาพขนาด 320x120 พิกเซล ดังตารางที่ 4-10
- อัตราส่วน 16:5 ภาพขนาด 320x100 พิกเซล ดังตารางที่ 4-11
- อัตราส่วน 4:1 ภาพขนาด 320x80 พิกเซล ดังตารางที่ 4-12




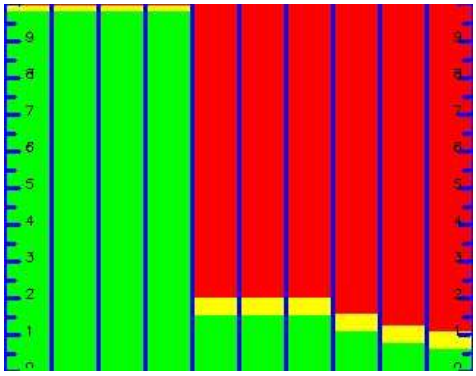



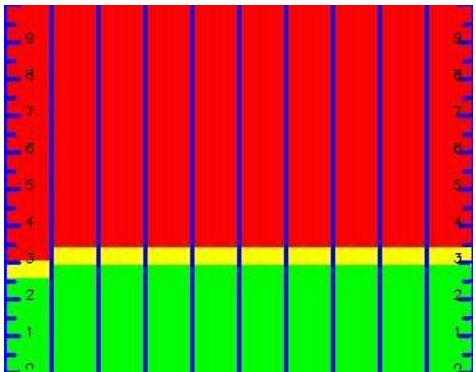
ตารางที่ 4-8 อัตราส่วนลักษณะ 4:3 ภาพขนาด 320x240 พิกเซล

รูปที่	รูปด้านซ้าย	ตำแหน่งของวัตถุ
	รูปดิสพาริตี	
1	 	



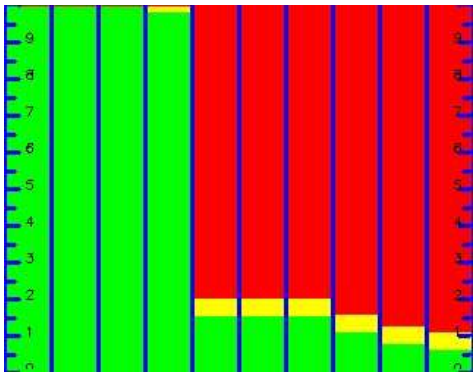
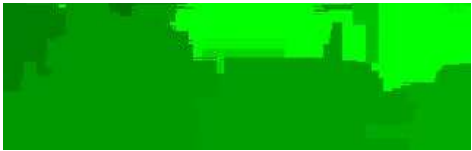

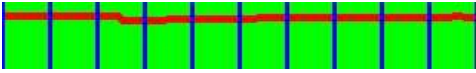
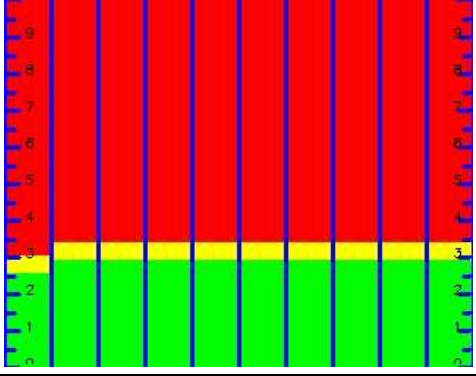



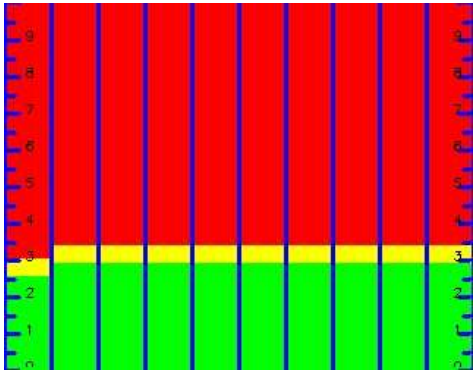

ตารางที่ 4-9 อัตราส่วนลักษณะ 16:9 ภาพขนาด 320x180 พิกเซล

รูปที่	รูปด้านซ้าย	ตำแหน่งของวัตถุ
	รูปดิสพาริตี	
1		
		
2		
		


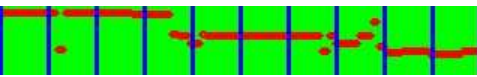

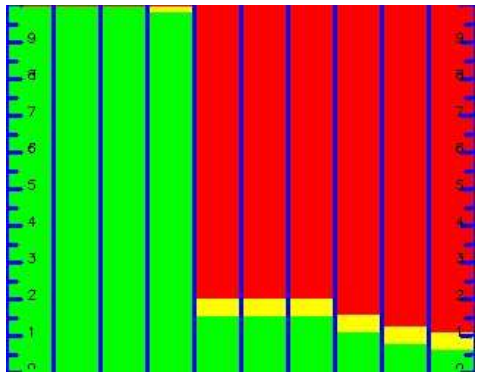

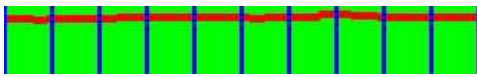

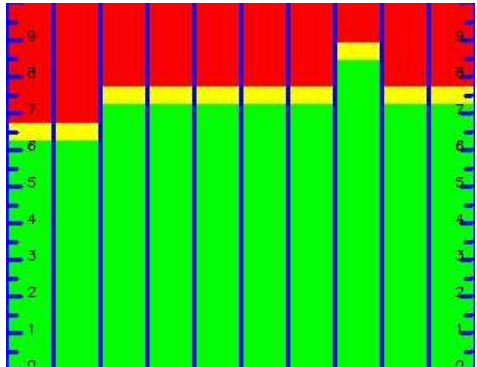

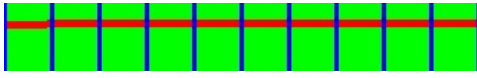
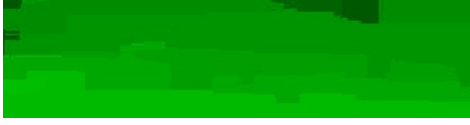
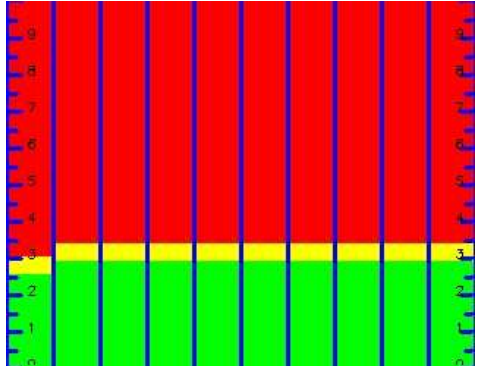
ตารางที่ 4-10 อัตราส่วนลักษณะ 8:3 ภาพขนาด 320x120 พิกเซล

รูปที่	รูปด้านซ้าย	ตำแหน่งของวัตถุ
	รูปดิสพาริตี	
1		
		
2		
		

ตารางที่ 4-11 อัตราส่วนลักษณะ 16:5 ภาพขนาด 320x100 พิกเซล

รูปที่	รูปด้านซ้าย	ตำแหน่งของวัตถุ
	รูปดิสพาริตี	
1		 
		
2		 
		
3		 
		

ตารางที่ 4-12 อัตราส่วนลักษณะ 4:1 ภาพขนาด 320x80 พิกเซล

รูปที่	รูปด้านซ้าย	ตำแหน่งของวัตถุ
	รูปดิสพาริตี	
1		
		
2		
		
3		
		

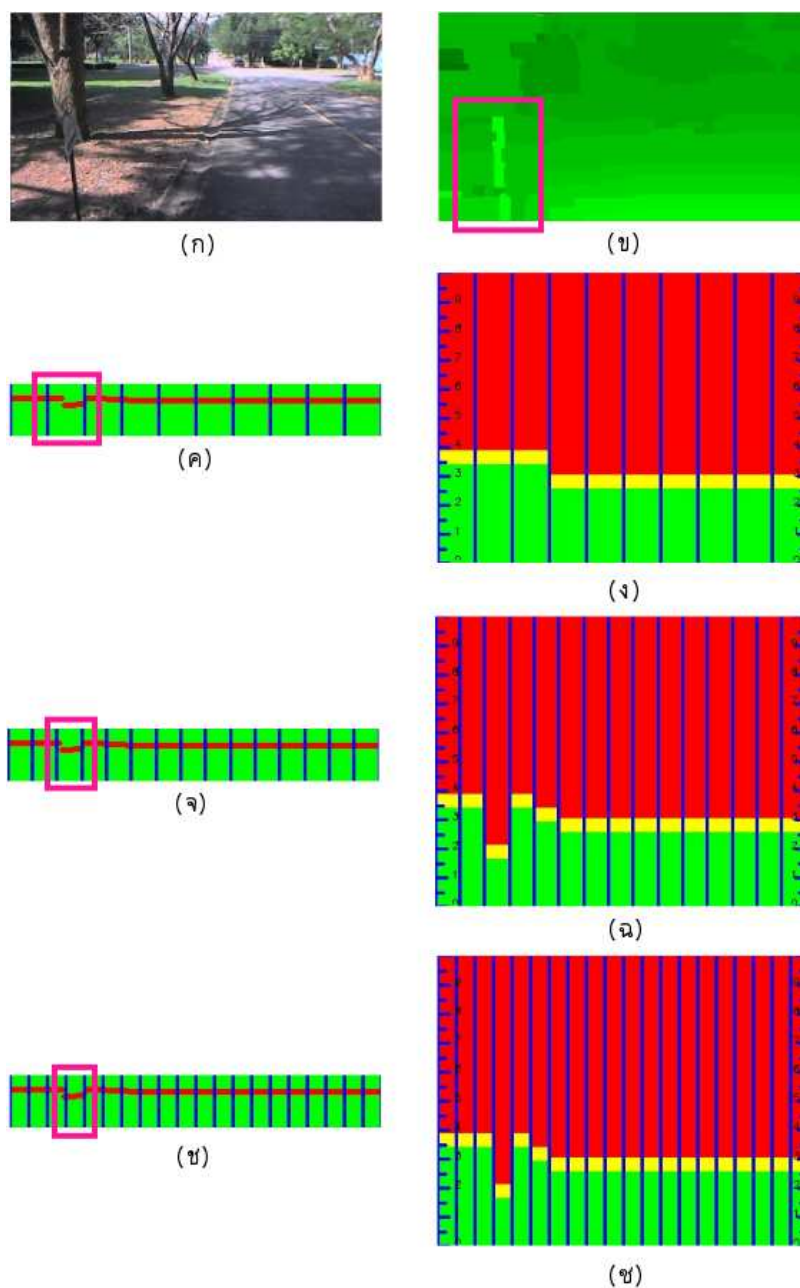
ภาพที่ใช้ทดสอบเป็นภาพที่ใช้มีอัตราส่วนระหว่างท้องฟ้าประมาณร้อยละ 40 และพื้นถนนร้อยละ 60 ซึ่งบริเวณท้องฟ้าจะมีกิ่งไม้ที่ยื่นออกมาให้เห็นได้ชัดเจน จากสถานการณ์นี้ผู้ใช้งานระบบสามารถเดินผ่านกิ่งไม้เหล่านี้ไปได้ จากการใช้ภาพขนาด 4:3 ดังตารางที่ 4-8 รูปที่ 1 ระบบจะพบว่าไม่มีวัตถุปรากฏอยู่ แต่ทางซ้ายจะสามารถผ่านไปได้นี้เนื่องจากไม่มีวัตถุใดๆ หากเราเปลี่ยนอัตราส่วนลักษณะจาก 4:3 เป็น 16:9 และ 8:3 ดังตารางที่ 4-9 และตารางที่ 4-10 มีการปรับตำแหน่งในการประมวลผลภาพโดยเริ่มจากความสูงด้านบนดังรูปที่ 1 และเริ่มจากความสูงด้านล่างดังรูปที่ 2 จากตารางที่ 4-9 และตารางที่ 4-10 ซึ่งมีอัตราส่วนลักษณะใกล้เคียงกันเมื่อถัวเฉลี่ยจำนวนภาพให้ครอบคลุมบริเวณภาพต้นฉบับทั้งหมดจะได้ 2 ภาพ ผลคือรูปที่ 1 ไม่สามารถผ่านด้านขวาได้แต่สามารถผ่านไปทางด้านซ้ายได้ซึ่งจะเหมือนกับตารางที่ 4-8 รูปที่ 1 แต่เมื่อเป็นรูปที่ 2 ระยะทางจะเสมอกันหมดและสามารถผ่านไปตลอดความกว้างของภาพ ผู้ใช้งานระบบจึงสามารถอนุมานได้ว่าน่าจะมีวัตถุที่ลอยอยู่เหนือพื้น

เมื่อปรับขนาดภาพโดยลดความสูงของภาพลงเป็น 16:5 ดังตารางที่ 4-11 และ 4:1 ดังตารางที่ 4-12 หากถัวเฉลี่ยจำนวนภาพให้ครอบคลุมบริเวณภาพต้นฉบับทั้งหมดจะได้ 3 ภาพ โดยที่แบ่งออกเป็นบน กลาง และล่าง หากพิจารณาจากตารางที่ 4-11 และตารางที่ 4-12 รูปที่ 1 จะเหมือนกับตารางที่ 4-9 และตารางที่ 4-10 รูปที่ 1 คือพบบริเวณกิ่งไม้ที่ระบบจะระบุว่าไม่สามารถผ่านไปได้ รูปที่ 2 บริเวณกลางภาพ ตรงนี้จะเห็นถนนและท้องฟ้าที่อยู่ไกลออกไป ส่งผลให้ผู้ใช้งานระบบสามารถตัดสินใจเดินทางต่อได้ส่วนรูปที่ 3 จะเป็นบริเวณพื้นเป็นหลักจะเหมือนกับตารางที่ 4-9 และตารางที่ 4-10 รูปที่ 2 นั่นเอง ซึ่งหากเราพิจารณารูปที่ 1 และ 2 จากตารางที่ 4-11 และตารางที่ 4-12 ผู้ใช้งานระบบจะสามารถอนุมานได้ว่าน่าจะมีวัตถุที่ลอยอยู่เหนือพื้นและสามารถผ่านสิ่งกีดขวางเหล่านั้นไปได้เนื่องจากรูปที่สองจะเป็นบริเวณกลางภาพที่มีระดับกล้องอยู่พอดีกับระดับสายตาเมื่อนำกล้องไปสวมไว้บนศีรษะ จึงผ่านได้อย่างไม่มีปัญหานั้นเอง แต่จากผลที่ได้จากตารางที่ 4-11 และตารางที่ 4-12 รูปที่ได้จากตารางที่ 4-12 มีความเหมาะสมในการใช้งานในลักษณะนี้มากกว่า เนื่องจากสามารถใช้ประมาณระยะทางของวัตถุที่แม่นยำกว่านั่นเอง ซึ่งอัตราส่วนลักษณะคือ 4:1 ภาพขนาด 320x80 พิกเซล

#### 4.2.4. การปรับช่วงในแนวตั้งเพื่อหาวัตถุที่มีความกว้างขนาดเล็ก

เนื่องจากการแบ่งช่วง U-disparity ออกเป็น 10 ช่วง ส่งผลให้วัตถุที่มีความกว้างน้อยนั้นระบบที่สร้างขึ้นไม่สามารถตรวจจับได้ถึงแม้ว่าอัลกอริทึม P2P จะสามารถหาคิสปาริตีได้ก็ตาม เนื่องจากระบบที่ได้ออกแบบไว้ใช้สถิติในการเลือกว่ากลุ่มคิสปาริตีใดเป็นผลลัพธ์ของช่วงนั้นๆ

ดังนั้นเพื่อแก้ปัญหาที่ได้กล่าวไว้ในข้างต้นจึงใช้วิธีการเพิ่มช่วงให้กับ U-disparity โดยผลที่ได้ดังรูปที่ 4-14



รูปที่ 4-14 ตัวอย่างการเพิ่มช่วงให้กับ U-disparity ภาพ (ก) ภาพซ้าย (ข) แผนผังคิสพาริตี (ค) การแบ่ง U-disparity ออกเป็น 10 ช่วง (ง) ผลลัพธ์ที่ได้จากการแบ่ง U-disparity ออกเป็น 10 ช่วง (จ) การแบ่ง U-disparity ออกเป็น 15 ช่วง (ฉ) ผลลัพธ์ที่ได้จากการแบ่ง U-disparity ออกเป็น 15 ช่วง (ช) การแบ่ง U-disparity ออกเป็น 20 ช่วง (ซ) ผลลัพธ์ที่ได้จากการแบ่ง U-disparity ออกเป็น 20 ช่วง



จากรูปที่ 4-14 ภาพ (ก) คือภาพด้านซ้ายที่ใช้ในการทดสอบนี้ เมื่อนำเข้าไปคำนวณด้วย อัลกอริทึม P2P จะได้ภาพ (ข) คือแผนผังดิสพาริตี ส่วนภาพ (ค) คือการแบ่ง U-disparity ออกเป็น 10 ช่วง ภาพ (ง) คือผลลัพธ์ที่ได้จากการแบ่ง U-disparity ออกเป็น 10 ช่วง ภาพ (จ) คือการแบ่ง U-disparity ออกเป็น 15 ช่วง ภาพ (ฉ) คือผลลัพธ์ที่ได้จากการแบ่ง U-disparity ออกเป็น 15 ช่วง ภาพ (ช) คือการแบ่ง U-disparity ออกเป็น 20 ช่วง ส่วนภาพ (ซ) คือผลลัพธ์ที่ได้จากการแบ่ง U-disparity ออกเป็น 20 ช่วง ซึ่งสังเกตได้ว่าหากเพิ่มการแบ่งช่วงให้ U-disparity จะช่วยในการตรวจสอบวัตถุที่มีขนาดเล็กหรือความกว้างน้อยได้ แต่การแบ่งช่วงที่มากเกินไปอาจส่งผลให้ผู้ใช้งานระบบยากต่อการคาดการณ์ว่าวัตถุอยู่ตำแหน่งใดถึงจะถูกต้อง ดังนั้นจึงใช้การแบ่งช่วง U-disparity ขนาด 10 ช่วง ไว้เป็นค่าปริยาย แต่เพิ่มทางเลือกให้ผู้ใช้งานสามารถเลือกการแบ่งช่วงที่เยอะขึ้นได้ตามความต้องการ

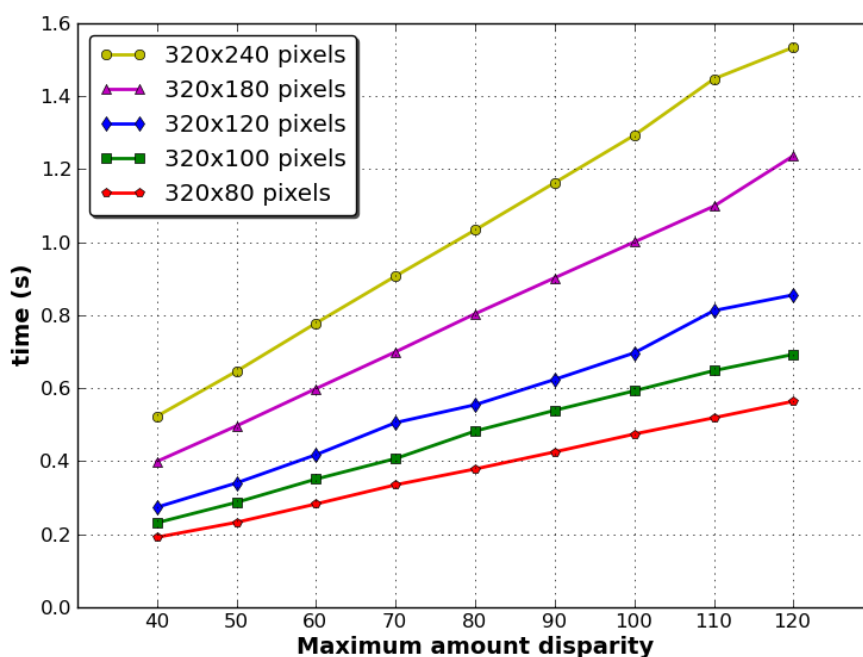
#### 4.3. วิเคราะห์ผลการทดสอบ

จากผลการทดสอบในหัวข้อ 4.1 และ 4.2 ในการประมวลผลของระบบตรวจสอบสิ่งกีดขวางด้วยสเตอริโอวิชัน ความเร็วของผลตอบสนองของอัลกอริทึม P2P ในลักษณะโปรแกรมแบบตามลำดับบนคอมพิวเตอร์ที่มีหน่วยประมวลผลสองแกนคือ 2.040 วินาทีเมื่อใช้วิธีการประมวลผลแบบขนานเวลาของผลตอบสนองจะอยู่ที่ 1.053 วินาทีโดยใช้ภาพขนาด 320x240 พิกเซล ระดับดิสพาริตีสูงสุดที่ 100 ค่าสปีดอัปเป็น 1.936 และ ค่าประสิทธิภาพของการประมวลผลคือ 0.968 ซึ่งความเร็วที่ได้จะตกอยู่ที่หนึ่งเฟรมต่อหนึ่งวินาที

การตรวจจับสิ่งกีดขวางนั้นอัลกอริทึม P2P สามารถทำงานได้ในบริเวณที่มีความสว่างมากกว่า 82 lux ในขณะที่วัตถุบางประเภทอัลกอริทึม P2P ไม่สามารถหาดิสพาริตีได้ ได้แก่วัตถุโปร่งใสเช่น กระจก แต่ถ้าเป็นกระจกที่มีมันบังอยู่ด้านหลัง อัลกอริทึม P2P นั้นยังสามารถหาดิสพาริตีได้เช่นกัน การใช้ภาพขนาดขนาดใหญ่ในการประมวลผลส่งผลให้อัลกอริทึม P2P ใช้เวลาในการประมวลผลนานขึ้น ผลตอบสนองที่ได้รับจะล่าช้าอาจก่อให้เกิดอันตรายแก่ผู้ใช้งานระบบได้อีกทั้งกรณีที่มีกิ่งไม้ที่ยื่นออกมาดังตารางที่ 4-7 รูปที่ 3 หากพิจารณาแล้วผู้ใช้งานระบบน่าจะเดินผ่านไปได้ แต่ระบบจะตรวจพบกิ่งไม้ซึ่งอยู่ด้านหน้าว่าเป็นวัตถุที่อยู่ใกล้แทน ในกรณีนี้สามารถแก้ไขได้ด้วยการปรับลดอัตราส่วนความสูงของแผนผังดิสพาริตีลง จากการทดสอบในกรณีนี้หากเลือกใช้อัตราส่วน 4:1 และการเลือกตรวจสอบสามบริเวณคือ บน กลาง ล่าง ในหนึ่งภาพ จะช่วยให้ผู้พิจารณาสามารถคาดคะเนได้ว่าสามารถจะเดินทางต่อไปในบริเวณดังกล่าวได้หรือไม่

จากการปรับอัตราส่วนระหว่างความกว้างและความยาวของภาพก่อนการคำนวณหาแผนผังดิสพาริตีส่งผลให้สามารถลดเวลาตอบสนองของของอัลกอริทึม P2P ได้อีกด้วย ดังแสดงใน

รูปที่ 4-15 เนื่องจากเวลาที่มากที่สุดของอัลกอริทึม P2P คือกระบวนการจับคู่พิกเซลในแต่ละแถว โดยที่ในแต่ละแถวใช้เวลาเฉลี่ย  $O(w\Delta \log \Delta)$  ถ้าหากมีจำนวนแถวขนาด  $h$  จะใช้เวลา  $O(hw\Delta \log \Delta)$  การลดจำนวนแถวลงจะเป็นการลดเวลาในการประมวลผลตามไปด้วย เมื่อเปลี่ยนขนาดภาพจาก 320x240 พิกเซลเป็น 320x180 พิกเซล คิดเป็นอัตราส่วน 16:9 จะใช้เวลาประมวลผลเพียง 0.970 วินาที เมื่อใช้จำนวนระดับดิสพาริตีสูงสุดที่ 100 การปรับภาพลงเหลืออัตราส่วน 16:9 จาก 4:3 นอกจากจะลดเวลาของการตอบสนองลงได้แล้วยังสามารถช่วยให้ผู้ใช้งานระบบสามารถเข้าใจได้ง่ายว่าตอนนี้ระดับของกล้องทำมุมอย่างไรกับพื้น เพื่อเป็นข้อมูลในการตัดสินใจแก่ผู้ใช้งานระบบเองว่าผู้ใช้งานระบบในขณะนั้นควรเลือกเดินไปในทิศทางใด นอกจากอัตราส่วนลักษณะที่กล่าวไปแล้วนี้สามารถดูเวลาที่ใช้ในการประมวลผลอัตราส่วนลักษณะอื่นได้ดังรูปที่ 4-15



รูปที่ 4-15 เวลาเฉลี่ยทั้งหมดของการประมวลผลอัลกอริทึม P2P ในลักษณะโปรแกรมแบบขนานที่ใช้ 2 โพรเซสในการประมวลผลบนคอมพิวเตอร์ที่มีหน่วยประมวลผลสองแกน ด้วยอัตราส่วนลักษณะต่างๆ

จากการทดสอบระบบโดยรวมผลปรากฏว่าจำนวนระดับดิสพาริตีสูงสุดที่เหมาะสมในการสร้างระบบตรวจสอบสิ่งกีดขวางสำหรับผู้พิการทางสายตาด้วยสเตอริโอวิชันคือ 60 เนื่องจากค่าดังกล่าวสามารถตรวจหาวัตถุตั้งแต่ 1 ถึง 6 เมตรได้ดี อีกทั้งเมื่อใช้จำนวนระดับดิสพาริตีสูงสุดดังกล่าวจะได้ผลตอบสนองจากการประมวลผลภาพสเตอริโอเพียงแค่ 0.598 วินาทีต่อภาพเท่านั้น

เมื่อใช้ภาพขนาด 320x180 พิกเซล ซึ่งสามารถใช้ตรวจจับวัตถุในการเดินทางของมนุษย์ได้ ภาพขนาดดังกล่าวจึงเหมาะแก่การใช้เป็นค่าปริยายของระบบ หากต้องการตรวจสอบวัตถุที่มีความเร็วมากขึ้น เช่นรถยนต์จักรยานยนต์ที่วิ่งช้าๆ หรือคนวิ่งควรรใช้อัตราส่วน 4:1 แทนแต่อาจจะทำให้ภาพที่ได้ขาดรายละเอียดไปบางส่วน เนื่องจากอัตราส่วนนี้ให้ความเร็วที่ 0.282 วินาทีต่อภาพ ดังนั้น 1 วินาทีจะประมวลผลได้ประมาณ 3 ภาพ ซึ่งภาพจำนวน 3 ภาพนี้เพียงพอต่อการประมวลผลวัตถุเคลื่อนไหวในระยะ 1 ถึง 6 เมตร

## บทที่ 5

### บทสรุป

ในบทนี้กล่าวสรุปผลการวิจัยที่ได้ดำเนินการสำหรับวิทยานิพนธ์นี้ รวมทั้งข้อเสนอแนะต่างๆ เพื่อเป็นประโยชน์ต่อการวิจัยเกี่ยวกับระบบตรวจสอบสิ่งกีดขวางสำหรับผู้พิการทางสายตาด้วยสเตอริโอวิชัน และการพัฒนาส่วนเสนอข้อมูลต่อผู้พิการทางสายตาต่อไป

#### 5.1. สรุปผลการวิจัย

งานวิจัยในวิทยานิพนธ์ฉบับนี้ได้นำเสนอระบบตรวจสอบสิ่งกีดขวางสำหรับผู้พิการทางสายตาด้วยสเตอริโอวิชันในส่วนของกระบวนการประมวลผลภาพสเตอริโอและการประมาณค่าระยะทางและตำแหน่งของวัตถุเพื่อนำข้อมูลที่ได้ไปเสนอต่อผู้พิการทางสายตาต่อไป ระบบที่ได้สร้างขึ้นนี้ใช้อัลกอริทึม P2P ซึ่งจัดอยู่ในเทคนิค FSM เพื่อป้องกันปัญหาที่อาจเกิดจากการตรวจจับวัตถุที่ไร้ลักษณะเด่นเช่น กระดานดำ ตู้ หรือ ประตู เป็นต้น

จากการสร้างกล่องสเตอริโอที่ติดตั้งบนหมวกที่มีระยะห่างระหว่างกล้อง 12 เซนติเมตรเมื่อได้ทดสอบอัลกอริทึม P2P ในระยะทางต่างๆ เพื่อหาค่าตัวแปรของอัลกอริทึม P2P ที่เหมาะสมต่อการสร้างระบบนี้พบว่าในระยะตั้งแต่ 1 ถึง 6 เมตรห่างจากผู้ใช้งานระบบ ระดับคิสพาริตีที่สูงสุดที่เหมาะสมในการใช้งานมีค่าอยู่ระหว่าง 50 ถึง 100 แต่จากการพิจารณาความสัมพันธ์ระหว่างระดับคิสพาริตีกับการตอบสนองของอัลกอริทึม P2P ระดับคิสพาริตีที่สูงสุดที่มีค่าสูงจะส่งผลต่อความเร็วในการประมวลผล โดยที่ยิ่งจำนวนระดับคิสพาริตีที่สูงสุดมีค่าสูงๆ การตอบสนองของอัลกอริทึม P2P จะใช้เวลานาน ดังนั้นจึงเลือกกำหนดค่ามาตรฐานไว้ที่ 60 เนื่องจากที่ระดับคิสพาริตีที่สูงสุดที่ 50 นั้นอัลกอริทึม P2P ยังไม่สามารถตรวจสอบสิ่งกีดขวางได้อย่างชัดเจนในระยะ 6 เมตรในส่วนตัวแปรอื่นๆ ที่อัลกอริทึมนี้ต้องการพบว่าค่าตัวแปรเดิมที่ผู้คิดค้นได้เสนอไว้ในการทำงานได้ดีอยู่แล้วเมื่อเปลี่ยนแปลงสภาพแวดล้อมไปจะส่งผลต่อแผนผังคิสพาริตีที่น้อยมาก

การนำเสนอข้อมูลที่ได้จากแผนผังคิสพาริตีซึ่งเป็นผลลัพธ์จากอัลกอริทึม P2P แก่ผู้พิการทางสายตานั้นไม่สามารถทำได้โดยตรง เนื่องจากแผนผังคิสพาริตีที่ประกอบขึ้นจากคิสพาริตีหลายระดับ ซึ่งแต่ละระดับหมายถึงวัตถุชิ้นต่างๆ ที่มีระยะทางที่แตกต่างกัน ผู้วิจัยจึงเสนอให้แปลงแผนผังคิสพาริตีให้อยู่ในรูปของ U-disparity เมื่อเสร็จสิ้นกระบวนการดังกล่าวแล้ว วัตถุหรือสิ่งกีดขวางต่างๆ จะปรากฏเป็นเส้นตรงแนวนอน โดยที่วัตถุที่อยู่ใกล้ผู้ใช้งานจะปรากฏอยู่ด้านล่าง ส่วนวัตถุที่อยู่ไกลออกไปจะอยู่ด้านบนของภาพ จากนั้นหาคิสพาริตีที่อยู่ในระดับล่างสุดที่มีค่าไม่เกิน

35 เพื่อใช้ในขั้นตอนต่อไป เพื่อให้ผู้ใช้งานระบบสามารถเข้าใจตำแหน่งของวัตถุได้ง่ายจึงได้แบ่ง U-disparity ออกเป็น 10 ช่วง โดยในแต่ละช่วงจะหาความสัมพันธ์ที่เป็นตัวแทนของช่วงนั้นๆ ผู้วิจัยเสนอให้ใช้วิธีการหาความถี่ในแต่ละช่วงที่กำหนดให้ U-disparity ในแต่ละช่วงหากระดับความสัมพันธ์ใดมีความถี่สูงสุดจะใช้ระดับนั้นเป็นข้อมูลในการประมาณค่าระยะทางด้วยสมการพหุนามต่อไป เมื่อผ่านขั้นตอนนี้จะได้อาร์เรย์ข้อมูลขนาด 10 หน่วยที่บรรจุระยะทางจากตัวผู้ใช้งานจนถึงบริเวณที่พบวัตถุ ซึ่งสามารถนำข้อมูลนี้ไปเสนอแก่ผู้ใช้งานระบบเพื่อประกอบการตัดสินใจในการเลือกทิศทางที่จะต้องเดินทางต่อไป อีกทั้งวิธีการนี้จะทำให้ผู้ใช้งานระบบสามารถคาดคะเนได้ว่ามีวัตถุอยู่บริเวณใดของภาพ การแบ่ง U-disparity ออกเป็น 10 ช่วงนี้อาจส่งผลให้ไม่สามารถตรวจจับวัตถุที่มีขนาดเล็กได้ เนื่องจากเมื่อวัตถุมีขนาดเล็ก จะปรากฏจำนวนระดับความสัมพันธ์ใน U-disparity น้อยตามไปด้วย เพื่อให้ครอบคลุมทุกขนาดของวัตถุ ผู้วิจัยจึงได้พัฒนาให้ส่วนนี้สามารถปรับจำนวนช่วงให้ลดลงหรือเพิ่มขึ้นได้ตามความเหมาะสม แต่การปรับขนาดให้ละเอียดเกินไปอาจส่งผลให้ผู้พิจารณาทางสายตาคาดคะเนตำแหน่งได้ยากเช่นกัน ดังนั้นจึงปรับตั้งค่ามาตรฐานไว้ที่ 10 ช่วง

การประมาณค่าระยะทางของสิ่งกีดขวางนั้นสามารถทำได้โดยหาความสัมพันธ์ระหว่างระดับความสัมพันธ์กับระยะทางจริงในระยะทางต่างๆ ผู้วิจัยได้ทดสอบหาระดับความสัมพันธ์ของวัตถุที่ระยะทางต่างๆ ตั้งแต่ 1 ถึง 7 เมตร จากการหาความสัมพันธ์ระหว่างระยะทางจริงกับระดับความสัมพันธ์ ผู้วิจัยจึงเสนอสมการพหุนามกำลัง 4 ที่ใช้ประมาณค่าระยะทางของวัตถุโดยสมการนี้ต้องอาศัยตัวแปรคือระดับความสัมพันธ์ในการคำนวณ สำหรับระดับความสัมพันธ์ที่ให้แก่สมการนี้มีค่าตั้งแต่ 0 ถึง 35 เท่านั้น เนื่องจากที่ระดับความสัมพันธ์ที่ 35 นั้นจะให้ระยะทาง 1 เมตรพอดี ทั้งนี้ข้อมูลที่เกินจาก 35 นอกจากเป็นระดับความสัมพันธ์ของวัตถุที่ใกล้กว่า 1 เมตรแล้ว อาจจะเป็นสิ่งรบกวนที่เกิดจากการประมวลผลของอัลกอริทึม P2P ได้ด้วยเช่นกัน

เมื่อนำระบบที่สร้างขึ้นทั้งหมดไปทดสอบใช้งานในสถานที่จริง การใช้ภาพขนาด 4:3 หรือ 320x240 พิกเซลนั้น ระบบจะให้ค่าระยะทางที่ผิดพลาดในหลายสถานการณ์ เนื่องจากทัศนียภาพที่ปรากฏในภาพขนาดนี้จะมีพื้นถนนและท้องฟ้าเป็นบริเวณกว้าง โดยเฉพาะพื้นถนนนั้นอัลกอริทึม P2P สามารถหาความสัมพันธ์ได้เช่นกัน ส่งผลให้ระบบบอกว่าพื้นถนนคือสิ่งกีดขวาง เพื่อแก้ปัญหาดังกล่าว ผู้วิจัยจึงเสนอให้ปรับลดอัตราส่วนลักษณะของภาพเหลือเพียง 16:9 หรือ 320x180 พิกเซลแทน ด้วยภาพขนาดนี้ระบบสามารถประมาณระยะทางได้ถูกต้องมากยิ่งขึ้น แต่ก็ยังพบปัญหาเกี่ยวกับทัศนียภาพที่มีวัตถุแขวนอยู่ เพื่อให้ผู้ใช้งานระบบสามารถตรวจสอบวัตถุว่ามีวัตถุแขวนอยู่หรือไม่ได้อย่างถูกต้อง ผู้วิจัยจึงเสนอให้ใช้อัตราส่วนลักษณะ 4:1 หรือ 320x80 พิกเซลโดยแบ่งการตรวจสอบออกเป็น ส่วนบน ส่วนกลาง และส่วนล่าง จากภาพที่ต้องการตรวจสอบ หากระบบพบว่า มีวัตถุในส่วนบนอยู่ใกล้ แต่ส่วนกลางและส่วนล่างไม่ปรากฏว่ามีวัตถุแขวนอยู่ ผู้ใช้งานระบบ

สามารถอนุมานได้ว่ามีวัตถุแขวนอยู่และอยู่ที่ตำแหน่งใด แต่ด้วยภาพขนาด 4:1 นี้ไม่สามารถใช้งานในสถานการณ์ทั่วไปได้ เนื่องจากความยาวเพียง 80 พิกเซลจะทำให้ภาพที่ใช้ในการประมวลผลมีมุมมองแคบลง ภาพที่ใช้ประมวลผลมีทัศนียภาพไม่สมบูรณ์ อาจก่อให้เกิดอันตรายต่อผู้ใช้งานได้ ดังนั้นขนาดภาพ 16:9 จึงเหมาะสมที่จะใช้เป็นค่าปริยายของระบบ

ถึงแม้ว่าอัลกอริทึม P2P จะสามารถหาแผนผังดิสพาร์ทิได้ดี แต่การตอบสนองของอัลกอริทึมใช้เวลานาน ดังนั้นผู้วิจัยจึงนำการประมวลผลแบบขนานด้วยการส่งผ่านข้อความโดยใช้ไลบรารี MPICH บนคอมพิวเตอร์ที่มีหน่วยประมวลผลแบบสองแกนเพื่อช่วยลดเวลาการตอบสนองของอัลกอริทึม โดยเข้าไปปรับปรุงกระบวนการจับคู่พิกเซลในแต่ละแถว และกระบวนการประมวลผลขั้นปลาย ซึ่งในแต่ละส่วนมีลักษณะการคำนวณที่แตกต่างกัน สำหรับกระบวนการจับคู่พิกเซลในแต่ละแถวนั้นสามารถแบ่งงานได้ง่ายเนื่องจากมีการแยกข้อมูลในแต่ละแถวที่เป็นอิสระจากกันอยู่แล้ว จึงแบ่งข้อมูลในแต่ละแถวออกเป็นกลุ่มของข้อมูลตามจำนวนหน่วยประมวลผล แต่ในส่วนการประมวลผลขั้นปลายนั้นมีการแลกเปลี่ยนข้อมูลระหว่างแถวและระหว่างคอลัมน์จึงยากต่อการแบ่งงานในทันที จากการที่ผู้วิจัยได้ศึกษาลักษณะการทำงานในส่วนนี้จึงสรุปรูปแบบออกมาเป็นสองส่วนคือ การคำนวณโดยใช้ข้อมูลทั้งแถวและการคำนวณโดยใช้ข้อมูลทั้งคอลัมน์ ซึ่งการคำนวณโดยใช้ข้อมูลทั้งแถวนั้นสามารถแบ่งงานได้โดยใช้วิธีการเช่นเดียวกับกระบวนการจับคู่พิกเซลในแต่ละแถว ในส่วนการคำนวณโดยใช้ข้อมูลทั้งคอลัมน์นั้นถึงแม้ว่าจะสามารถแบ่งงานให้คำนวณตามกลุ่มคอลัมน์ได้ แต่ไม่มีฟังก์ชันใดใน MPI สนับสนุนการรวบรวมข้อมูลกลับโดยตรง ผู้วิจัยจึงประยุกต์ใช้ฟังก์ชัน MPI::Allreduce และใช้โอเปอเรชันสำหรับการรวบรวมข้อมูลกลับให้เหมาะสมกับลักษณะการคำนวณในแต่ละแบบเพื่อแก้ปัญหาดังกล่าว

จากการทดสอบความเร็วของระบบที่ได้สร้างขึ้นบนคอมพิวเตอร์แบบสองแกนพบว่า หากใช้ภาพขนาด 320x240 พิกเซล จำนวนระดับดิสพาร์ทิที่สูงสุดที่ 100 วิธีการประมวลผลแบบขนานจะใช้เวลาเพียง 1.053 วินาที ในขณะที่การประมวลผลแบบตามลำดับใช้เวลาผลตอบสนองที่ 2.040 วินาที คิดเป็นค่าสปีดอัปได้ 1.936 และค่าประสิทธิภาพในการประมวลผล 0.968 และเมื่อทดสอบบนคอมพิวเตอร์สมรรถนะสูงที่ใช้หน่วยประมวลผล 8 หน่วยพบว่า การประมวลผลแบบตามลำดับใช้เวลาผลตอบสนองที่ 1.350 วินาที แต่เมื่อใช้การประมวลผลแบบขนานโดยใช้จำนวนโปรเซสเท่ากับ 2, 4 และ 8 จะใช้เวลา 0.700, 0.375 และ 0.206 วินาทีตามลำดับ โดยค่าสปีดอัปคิดเป็น 1.930, 3.603 และ 6.567 ค่าประสิทธิภาพในการประมวลผลคิดเป็น 0.965, 0.901 และ 0.821 ตามลำดับ จากการทดสอบโปรแกรมบนคอมพิวเตอร์สมรรถนะสูงที่ใช้หน่วยประมวลผล 8 หน่วยพบว่าเมื่อมีจำนวนหน่วยประมวลผลที่เพิ่มขึ้นจะสามารถลดเวลาตอบสนองของอัลกอริทึม P2P ลง

ได้อีก แต่อัตราการลดจะไม่สามารถลดลงได้มากกว่าการใช้จำนวนหน่วยประมวลผลน้อยๆ เนื่องจากจำนวนหน่วยประมวลผลที่มากขึ้นจะส่งผลให้ปริมาณข้อมูลที่ใช้ในการประมวลผลของแต่ละโปรเซสชันน้อยลงในขณะที่มีการรับส่งข้อมูลบ่อยขึ้นนั่นเอง ดังนั้นโอเวอร์เฮดของการรับส่งข้อมูลจะเพิ่มเวลาในการตอบสนอง และส่งผลให้ลดทอนประสิทธิภาพของการประมวลผลในที่สุด

เมื่อพิจารณาจากระดับคิสพาร์ทีสูงสุดที่ 60 และใช้ภาพขนาด 320x180 พิกเซลซึ่งทั้งสองตัวแปรนี้จะเป็ค่าที่จะใช้จริงในระบบ การประมวลผลแบบขนานใช้เวลาเพียง 0.598 วินาทีต่อภาพเท่านั้น ในขณะที่การประมวลผลแบบตามลำดับใช้เวลา 1.136 วินาที

โดยสรุปแล้วงานในวิทยานิพนธ์นี้สามารถหาความเหมาะสมของระบบสเตอริโอวิชันที่ใช้ในการตรวจจับสิ่งกีดขวางสำหรับผู้พิการทางสายตาได้ และสามารถตรวจจับวัตถุ และบอกระยะทางและตำแหน่งของวัตถุได้ ในระยะทาง 1 ถึง 6 เมตร มีวิธีการตรวจสอบขนาดวัตถุที่สามารถตรวจจับได้ในระยะทางต่างๆ และสามารถเพิ่มประสิทธิภาพของระบบสเตอริโอวิชันให้สามารถทำงานได้เร็วขึ้นด้วยการประมวลผลแบบขนาน

## 5.2. ข้อเสนอแนะ

จากการทดสอบโปรแกรมระบบตรวจสอบสิ่งกีดขวางสำหรับผู้พิการทางสายตาด้วยสเตอริโอ-โอวิชันในส่วนของการประมวลผลภาพสเตอริโอและการประมาณค่าระยะทางและตำแหน่งของวัตถุมีข้อเสนอแนะดังต่อไปนี้

- 1) ควรปรับอัตราส่วนของภาพก่อนการคำนวณหาแผนผังคิสพาร์ทีเพื่อช่วยลดเวลาการตอบสนองอัลกอริทึม P2P และช่วยให้ผู้ใช้งานรับรู้ระดับการทำมุมกล้องสเตอริโอกับพื้นเพื่อช่วยในการตัดสินใจในการเลือกเส้นทาง หากต้องการใช้ตรวจจับวัตถุที่มีความเร็วสูงขึ้นต้องปรับอัตราส่วนลักษณะของภาพลง เพื่อลดข้อมูลในการประมวลผลภาพสเตอริโอ แต่อาจส่งผลให้ไม่สามารถตรวจสอบสภาพแวดล้อมโดยรวมได้
- 2) หากสถานที่ที่ผู้ใช้งานระบบเดินทางอยู่มีทัศนียภาพในรูปแบบเดียวกันไปโดยตลอดเช่นทางเดินเชื่อมระหว่างอาคาร ทางเดินภายในอาคาร ฯลฯ อาจจะลดจำนวนภาพที่ใช้ในการประมวลผลลงได้ แต่ต้องมีระบบตรวจสอบพื้นที่ที่เหมาะสม เพื่อไม่ให้เกิดอันตรายแก่ตัวผู้ใช้งานระบบ
- 3) ในการใช้งานระบบอาจเกิดข้อผิดพลาดจากอัลกอริทึม P2P ขึ้นได้ เนื่องจากความไม่สอดคล้องกันในการจับภาพเพื่อใช้ในการประมวลผล จึงควรมีระบบทำนายทัศนียภาพที่ฉลาดเพียงพอ เพื่อป้องกันไม่ให้ผู้ใช้งานระบบเข้าใจสภาพแวดล้อมคลาดเคลื่อนในการใช้งานจริงในอนาคต

- 4) อาจใช้อัลกอริทึมการประมวลผลภาพสเตอริโออื่นๆ มาประมวลผลภาพเดียวกันเพื่อเพิ่มความแม่นยำ
- 5) ในบริเวณที่ความสว่างน้อยโดยเฉพาะเวลากลางคืนอาจเพิ่มไฟฉายหรือแหล่งกำเนิดแสงอื่นที่ให้ความสว่างกว่า 82-100 lux บริเวณวัตถุหรือสิ่งกีดขวางอื่น เพื่อให้อัลกอริทึม P2P สามารถคำนวณแผนผังคิสพารีที่ออกมาได้
- 6) จากการทดสอบการประมวลผลแบบขนานโดยใช้วิธีการส่งผ่านข้อความด้วยไลบรารีของ MPICH เพื่อให้สามารถใช้งานให้เต็มประสิทธิภาพต้องใช้ช่องทางสื่อสารให้เหมาะสมกับสถาปัตยกรรมของคอมพิวเตอร์นั้นๆ มิฉะนั้นอาจเป็นการเพิ่มโอเวอร์เฮดให้แก่การคำนวณสำหรับคอมพิวเตอร์ตั้งโต๊ะส่วนบุคคลทั่วไป หรือคอมพิวเตอร์พกพาเครื่องเดียวนั้นสามารถใช้งานช่องทางสื่อสาร ch3:nemesis และ ch3:ssm สำหรับไลบรารี MPICH 2 ซึ่งช่องทางสื่อสารทั้งหมดเป็นการใช้หน่วยความจำร่วมแทนซ็อกเก็ตในการแลกเปลี่ยนข้อมูล
- 7) ในการกำหนดจำนวนโปรเซสเพื่อใช้ประมวลผลงานใดงานหนึ่งพึงคำนึงถึงจำนวนหน่วยประมวลผลที่อยู่ในระบบ จากการทดลองในหัวข้อที่ 4.1.3.1 พบว่าในการประมวลผลอัลกอริทึม P2P โดยการกำหนดจำนวนโปรเซสที่มากกว่าหน่วยประมวลผลที่มีอยู่จริงไม่สามารถลดเวลาในการประมวลผลลงได้ แต่กลับเป็นการเพิ่มเวลาในการประมวลผลอันเนื่องมาจากการสลับงานบนหน่วยประมวลผลเพื่อให้สามารถประมวลผลโปรเซสต่างๆ ได้ครบถ้วน
- 8) พึงระวังอาร์เรย์ข้อมูลขนาดใหญ่ในการเขียนโปรแกรมด้วยภาษา C/C++ เนื่องจากโครงสร้างข้อมูลชนิดนี้ได้รับการออกแบบให้เก็บอยู่ในสแต็กเซ็กเมนต์ (stack segment) ซึ่งระบบปฏิบัติการได้กำหนดค่าคงที่ปรียายไว้ค่าหนึ่ง ในกรณีการประมวลผลภาพการสร้างอาร์เรย์ที่มีขนาดใหญ่หลายๆ อาร์เรย์ อาจส่งผลให้เกิดการใช้งานเกินขนาดของสแต็กเซ็กเมนต์ที่กำหนดส่งผลให้เกิดข้อผิดพลาด "Segmentation fault" ขณะเวลาโปรแกรมกำลังทำงานได้ แต่สามารถแก้ไขได้สองกรณีคือใช้การจัดสรรหน่วยความจำพลวัตแทนซึ่งส่วนนี้จะถูกเก็บอยู่ในฮีปเซ็กเมนต์ (heap segment) แทนหรือใช้คำสั่ง ulimit ในระบบปฏิบัติการยูนิกซ์ เพื่อเพิ่มขนาดเซ็กเมนต์ดังกล่าว



## เอกสารอ้างอิง

- [1] H. Hashimoto, K. Magatani, and K. Yanashima, "The development of the navigation system for visually impaired persons," in *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, vol.2, 2001, pp. 1481-1483.
- [2] K. Soeda, S. Aoki, K. Yanashima, and K. Magatani, "Development of the visually impaired person guidance system using GPS," in *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, 2004, pp. 4870-4873.
- [3] A. Helal, S.E. Moore, and B. Ramachandran, "Drishti: An Integrated Navigation System for Visually Impaired and Disabled," in *Proceedings of the 5th IEEE International Symposium on Wearable Computers*, IEEE Computer Society, 2001, p. 149.
- [4] F. Cecelja, V. Garaj, Z. Hunaiti, and W. Balachandran, "A Navigation System for Visually Impaired," in *Proceedings of Instrumentation and Measurement Technology Conference*, 2006, pp. 1690-1693.
- [5] T. Harada, Y. Kaneko, Y. Hirahara, K. Yanashima, and K. Magatani, "Development of the navigation system for visually impaired," in *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, 2004, pp. 4900-4903.
- [6] N. Takatori, K. Nojima, M. Matsumoto, K. Yanashima, and Magatani K, "Development of voice navigation system for the visually impaired by using IC tags," in *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, 2006, pp. 5181-5184.
- [7] Y. Ebrahim, W. Abdelsalam, M. Ahmed, and Siu-Cheung Chau, "Proposing a hybrid tag-camera-based identification and navigation aid for the visually impaired," in *Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE*, 2005, pp. 172-177.
- [8] B. Ando, "Electronic sensory systems for the visually impaired," *Instrumentation & Measurement Magazine, IEEE*, vol. 6, 2003, pp. 62-67.

- [9] S. Kaluwahandi and Y. Tadokoro, "Portable traveling support system using image processing for the visually impaired," in *Proceedings of 2001 International Conference on Image Processing*, vol.1, 2001, pp. 337-340.
- [10] S. Meers and K. Ward, "A vision system for providing 3D perception of the environment via transcutaneous electro-neural stimulation," in *Proceedings of Eighth International Conference on Information Visualisation, 2004. IV 2004.*, 2004, pp. 546-552.
- [11] สมาคมวิชาการหุ่นยนต์ไทย. "โครงการการศึกษาการพัฒนาารถอัจฉริยะ: เสนอต่อ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ, สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ," *สมาคมวิชาการหุ่นยนต์ไทย*, 29 March, 2008. [Online]. Available: <http://www.tris.or.th/download/IntelligentVehicle/studyreport.pdf>. [Accessed: November 19, 2008].
- [12] R. Owens, "Stereo matching," *Computer Vision*, 29 October, 1997. [Online]. Available: [http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/OWENS/LECT11/lect11.html](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT11/lect11.html). [Accessed: June 20, 2008].
- [13] S. Birchfield and C. Tomasi, "Depth discontinuities by pixel-to-pixel stereo," in *Proceedings of the Sixth IEEE International Conference on Computer Vision*, 1998, pp. 1073-1080.
- [14] S. Birchfield and C. Tomasi, "A pixel dissimilarity measure that is insensitive to image sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 401-406, 1998.
- [15] H. Adachi, F. Nakatsuji and Ta. Nakamura, "StereoMatching," [Online]. Available: <http://cell.fixstars.com/opencv/index.php/StereoMatching>. [Accessed: November 19, 2009].
- [16] ชีรณี อจลากุล และ ราชวิรัช สโรชวิกสิต, *เทคโนโลยีการประมวลผลแบบกระจาย*. กรุงเทพฯ:ท็อป, 2551.
- [17] G. Bradski and A. Kaehler, *OpenCV*. USA:O'Reilly Media, 2008.
- [18] W. Gropp, et al., "MPI:A Message-Passing Interface Standard Version 2.2," *MPI 2.2 Documents*, September 4, 2009. [Online]. Available: <http://www.mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf>. [Accessed: November 19, 2009].

- [19] W. Gropp, et al., "MPICH2 User Guide Version 1.2," *MPICH2 DOCS*, October 7, 2009. [Online]. Available: <http://www.mcs.anl.gov/research/projects/mpich2/documentation/files/mpich2-1.2-userguide.pdf>. [Accessed: November 19, 2009].
- [20] W. Gropp, et al. "MPICH2 Installer's Guide Version 1.2," *MPICH2 DOCS*, November 18, 2009. [Online]. Available: <http://www.mcs.anl.gov/research/projects/mpich2/documentation/files/mpich2-1.2.1-installguide.pdf>. [Accessed: November 19, 2009].
- [21] "HP-MPI V2.3 for Linux Release Note.," [Online]. Available: <http://h21007.www2.hp.com/portal/download/files/unprot/mpi/T1919-90017.pdf>. [Accessed: November 19, 2009].
- [22] "What is MacMPI?," *AppleSeed Development Page*, August 21, 2006. [Online]. Available: <http://exodus.physics.ucla.edu/appleseed/dev/developer.html#MacMPI>. [Accessed: November 19, 2009].
- [23] E. Gabriel, et al., "Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation," in *Proceedings of the 11th European PVM/MPI Users' Group Meeting*, 2004, pp. 97–104.
- [24] S. Clark, "Logitech QuickCam for Notebooks Pro.," 2009. [Online]. Available: [http://www.everythingusb.com/logitech\\_quickcam\\_for\\_notebooks\\_pro.html](http://www.everythingusb.com/logitech_quickcam_for_notebooks_pro.html). [Accessed: January 4, 2009].
- [25] R. Labayrade, D. Aubert, and J. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation," *Intelligent Vehicle Symposium, 2002. IEEE*, vol.2, pp. 646-651, 2002.
- [26] R. Labayrade and D. Aubert, "In-vehicle obstacles detection and characterization by stereovision," in *Proceeding of the 1<sup>st</sup> International Workshop on In-Vehicle Cognitive Computer Vision Systems*, 2003, pp. 13-19.
- [27] A. Sappa, R. Herrero, F. Dornaika, D. Gerónimo, and A. López, "Road Approximation in Euclidean and v -Disparity Space: A Comparative Study," in *Computer Aided Systems Theory – EUROCAST 2007*, 2007, pp. 1105-1112.
- [28] A. Sappa, D. Geronimo, F. Dornaika, and A. Lopez, "On-board camera extrinsic parameter estimation," *Electronics Letters*, vol. 42, pp. 745-747, 2006.

[29] R. Labayrade and D. Aubert, "Robust and Fast Stereovision Based Road Obstacles Detection for Driving Safety Assistance," in *Proceeding of IAPR Work Mach Vis Appl 2002*, 2002, pp. 624-627.

[30] Zhencheng Hu, F. Lamosa, and K. Uchimura, "A complete U-V-disparity study for stereovision based 3D driving environment analysis," in *the Fifth International Conference on 3-D Digital Imaging and Modeling*, 2005, pp. 204-211.

**ภาคผนวก**

ภาคผนวก ก.

ไลบรารี MPI ที่ใช้ในวิทยานิพนธ์ทั้งหมด

## ไลบรารี MPI ที่ใช้ในวิทยานิพนธ์ทั้งหมด

MPI ได้กำหนดมาตรฐานของฟังก์ชันไว้หลายกลุ่มด้วยกันเพื่อให้สามารถใช้งานได้ครอบคลุมกับความต้องการของกันพัฒนาโปรแกรมแบบขนาน โดยฟังก์ชันที่ใช้ในวิทยานิพนธ์นี้มีดังต่อไปนี้ โดยอ้างอิงฟังก์ชันสำหรับภาษา C++ จาก MPI: A Message-Passing Interface Standard Version 2.2 [18]

### **void MPI::Init(int& argc, char\*\*& argv)**

ตามข้อกำหนดของ MPI ที่ต้องการให้สามารถย้ายซอร์สโค้ดจากระบบคอมพิวเตอร์หนึ่งไปยังอีกระบบคอมพิวเตอร์หนึ่งได้โดยไม่ต้องแก้ไขซอร์สโค้ดใหม่เพียงแต่ ระบุตัวแปรสภาพแวดล้อมที่ต้องการลงไปในขณะที่เริ่มการทำงานของโปรแกรมโดยใช้คำสั่ง mpiexec อีกทั้งเพื่อช่อนการจัดการในส่วนการสร้างโปรเซสขณะเริ่มการทำงานเอาไว้ ทุกๆ ฟังก์ชันของ MPI จะใช้งานได้จำเป็นต้องเรียกใช้งานฟังก์ชัน MPI::Init ก่อนอย่างน้อยหนึ่งครั้ง เพื่อกำหนดค่าเริ่มต้นของสภาพแวดล้อม ฟังก์ชัน MPI::Init รับพารามิเตอร์ 2 โดยที่พารามิเตอร์ 2 คำนี้นักเป็นพารามิเตอร์ของฟังก์ชัน main ด้วย

### **void MPI::Finalize()**

เป็นฟังก์ชันสำหรับจบการทำงานของ MPI ตามปกติ

### **void MPI::Comm::Abort(int errorcode)**

เป็นฟังก์ชันสำหรับจบการทำงานของ MPI แบบไม่ปกติ โดยสามารถแสดงรหัสข้อผิดพลาดให้แก่ระบบปฏิบัติการได้เมื่อสิ้นสุดโปรแกรม

### **int MPI::Comm::Get\_rank() const**

แสดงค่าลำดับโปรเซสว่าโปรเซสปัจจุบันจัดเป็นลำดับเป็นเท่าใดจากจำนวนโปรเซสทั้งหมด

### **int MPI::Comm::Get\_size() const**

แสดงขนาดจำนวนโปรเซสที่ใช้ในการประมวลผล

**double MPI::Wtime()**

แสดงเวลาปัจจุบัน

**void MPI::Comm::Bcast(**

**void\* buffer, int count, const Datatype& datatype, int root**

**) const = 0**

กระจายข้อความจาก root โพรเซสไปยังโพรเซสอื่นๆ โดยที่มีพารามิเตอร์ต่างๆ ดังต่อไปนี้

- **buffer** ตำแหน่งที่เริ่มส่งข้อมูล
- **count** จำนวนข้อมูลที่ส่ง
- **datatype** ชนิดข้อมูลที่ส่ง
- **root** หมายเลขลำดับโพรเซสที่จะเริ่มส่งข้อมูล

**void MPI::Comm::Gather(**

**const void\* sendbuf, int sendcount, const MPI::Datatype& sendtype,**

**void\* recvbuf, int recvcount, const MPI::Datatype& recvtype, int root**

**) const = 0**

รวบรวมข้อความจากทุกๆ โพรเซสมายัง root โพรเซส โดยที่มีพารามิเตอร์ต่างๆ ดังต่อไปนี้

- **sendbuf** ตำแหน่งข้อมูลที่เริ่มรวบรวมกลับ
- **sendcount** จำนวนข้อมูลที่รวบรวมกลับ
- **sendtype** ชนิดข้อมูลที่รวบรวมกลับ
- **recvbuf** ตำแหน่งข้อมูลที่ใช้เพื่อการรวบรวมกลับ
- **recvcount** จำนวนข้อมูลที่ใช้เพื่อการรวบรวมกลับ
- **recvtype** ชนิดข้อมูลที่รวบรวมกลับ
- **root** หมายเลขลำดับโพรเซสที่ต้องการให้รวบรวมข้อมูลกลับ

**void MPI::Comm::Allgather(**

**const void\* sendbuf, int sendcount, const Datatype& sendtype,**

**void\* recvbuf, int recvcount, const Datatype& recvtype**



) const = 0

คล้ายกับ Gather แต่เมื่อรวบรวมแล้วทุกๆ โพรเซสจะมีข้อมูลเหมือนกัน โดยที่มีพารามิเตอร์ต่างๆ ดังต่อไปนี้

- **sendbuf** ตำแหน่งข้อมูล que เริ่มรวบรวมกลับ
- **sendcount** จำนวนข้อมูลที่รวบรวมกลับ
- **sendtype** ชนิดข้อมูลที่จะรวบรวมกลับ
- **recvbuf** ตำแหน่งข้อมูลที่ใช้เพื่อการรวบรวมกลับ
- **recvcount** จำนวนข้อมูลที่ต่อรวบรวมกลับ
- **recvtype** ชนิดข้อมูลที่จะรวบรวมกลับ

**void MPI::Comm::Reduce(**

**const void\* sendbuf, void\* recvbuf, int count,**

**const MPI::Datatype& datatype, const MPI::Op& op, int root**

**) const = 0**

รวบรวมข้อความจากทุกๆ โพรเซสมายัง root โพรเซส ผ่านการกระทำบางอย่างที่ระบุไว้ใน Op โดยที่มีพารามิเตอร์ต่างๆ ดังต่อไปนี้

- **sendbuf** ตำแหน่งข้อมูล que เริ่มรวบรวมกลับ
- **recvbuf** ตำแหน่งข้อมูลที่ใช้เพื่อการรวบรวมกลับ
- **count** จำนวนข้อมูลที่ได้จากการรวบรวมกลับ
- **datatype** ชนิดข้อมูลที่จะรวบรวมกลับ
- **op** ตัวดำเนินการที่ต้องการ
  - MPI::MAX หาค่าสูงสุดของข้อมูลทั้งหมดจากทุกๆ โพรเซส
  - MPI::MIN หาค่าต่ำสุดของข้อมูลทั้งหมดจากทุกๆ โพรเซส
  - MPI::SUM หาผลรวมของข้อมูลทั้งหมด
- **root** หมายเลขลำดับโพรเซสที่ต้องการรวบรวมข้อมูลกลับ

**void MPI::Comm::Allreduce(**

**const void\* sendbuf, void\* recvbuf, int count,**

**const MPI::Datatype& datatype, const MPI::Op& op**

) **const = 0**

คล้ายกับ Reduce เพียงแต่เมื่อรวบรวมข้อมูลกลับแล้วทุกๆ โพรเซสจะมีข้อมูลเหมือนกันทั้งหมด ดังต่อไปนี้

- **sendbuf** ตำแหน่งข้อมูลที่จะรวบรวมกลับ
- **recvbuf** ตำแหน่งข้อมูลที่ได้จากการรวบรวมกลับ
- **count** จำนวนข้อมูลที่ได้จากการรวบรวมกลับ
- **datatype** ชนิดข้อมูลที่จะรวบรวมกลับ
- **op** ตัวดำเนินการที่ต้องการ
  - **MPI::MAX** หาค่าสูงสุดของข้อมูลทั้งหมดจากทุกๆ โพรเซส
  - **MPI::MIN** หาค่าต่ำสุดของข้อมูลทั้งหมดจากทุกๆ โพรเซส
  - **MPI::SUM** หาผลรวมของข้อมูลทั้งหมด

ภาคผนวก ข.

ตัวอย่างภาพแผนผังนิทรรศการที่ระยะทางต่างๆ

## ตัวอย่างภาพแผนผังดิสพาร์ตีที่ระยะทางต่างๆ

ส่วนนี้เสนอภาพตัวอย่างดิสพาร์ตีที่ระยะทางต่างๆ โดยเปลี่ยนระดับดิสพาร์ตีเพื่อหาความเหมาะสมสำหรับใช้งานในระบบตรวจจับสิ่งกีดขวางเพื่อผู้พิการทางสายตาด้วยสเตอริโอวิชัน จากการทดสอบในระยะทาง 1 ถึง 7 เมตร พบว่าจำนวนระดับดิสพาร์ตีสูงสุดที่เหมาะสมในการใช้งานคือ 50 ถึง 100 ทั้งนี้ขึ้นอยู่กับระยะทาง และความแม่นยำในการประมาณค่าระยะทางที่ต้องการและขนาดวัตถุด้วย สำหรับการทดสอบนี้ใช้ภาพตั้งต้นในแต่ละระยะทางดังรูปที่ ข-1 โดยตัวอย่างที่ได้จากการทดสอบแสดงดังตารางที่ ข-1



1 m



2 m



3 m



4 m



5 m



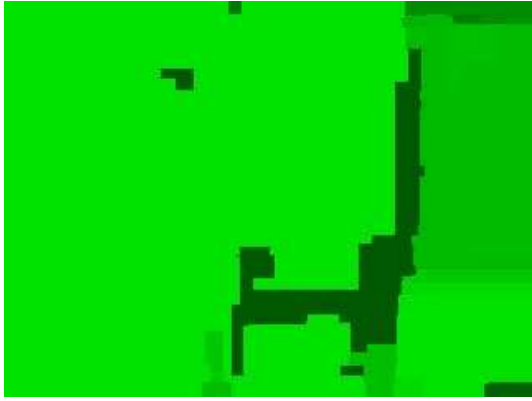
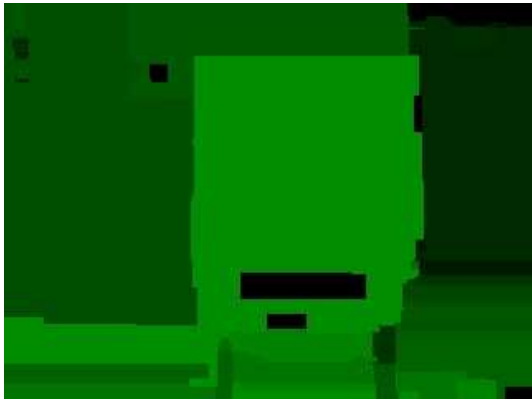
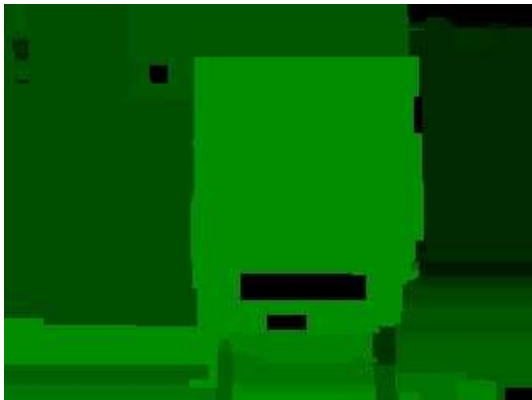
6 m

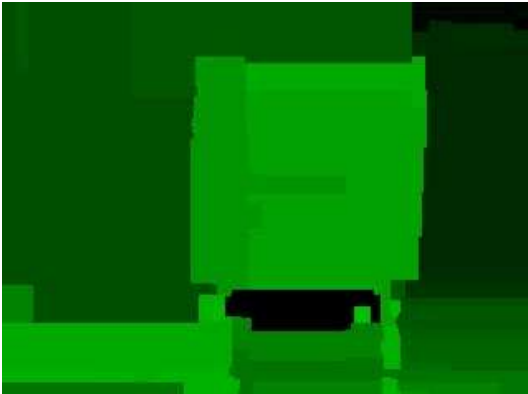
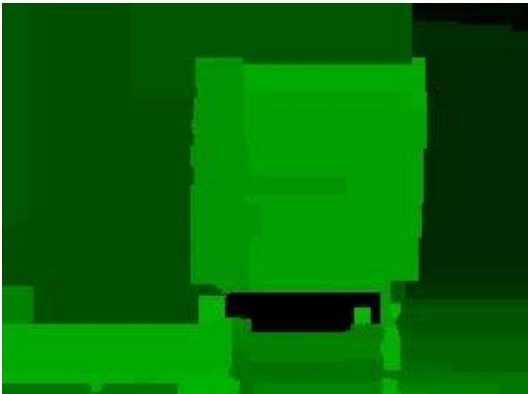
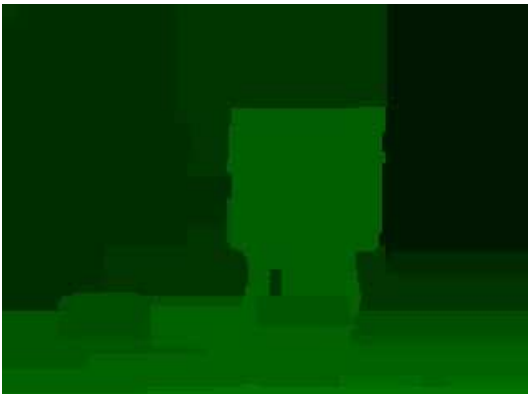







7 m

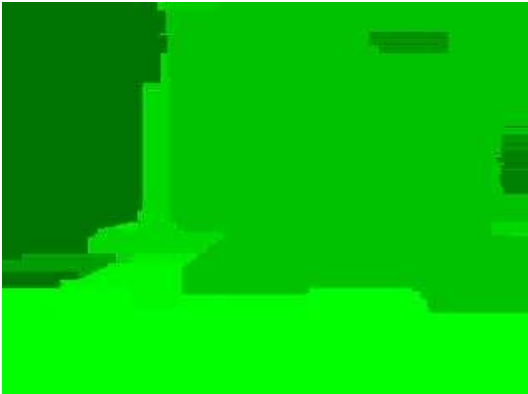
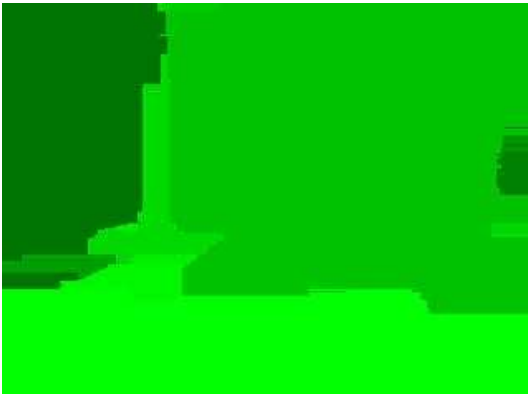

รูปที่ ข-1 ภาพตั้งต้นที่ใช้สำหรับหาค่าจำนวนระดับดิสพาร์ตีสูงสุดที่เหมาะสมต่อระบบตรวจจับสิ่งกีดขวาง

ตารางที่ ข-1 ตัวอย่างภาพดิสพาริตีเมื่อกำหนดจำนวนระดับดิสพาริตีสูงสุดต่างๆ ที่ระยะทางต่างๆ

ระยะทาง (m)	ระดับดิสพาริตี	ภาพดิสพาริตี
1	20	
1	30	
1	40	

ระยะทาง (m)	ระดับตีสฟารีที่	ภาพตีสฟารีที่
1	50	
1	60	
2	60	
3	60	

ระยะทาง (m)	ระดับตีสภาพริที่	ภาพตีสภาพริที่
4	60	
4	100	
5	60	
6	60	

ระยะทาง (m)	ระดับตีสภาพิที่	ภาพตีสภาพิที่
7	60	
7	100	
7	120	



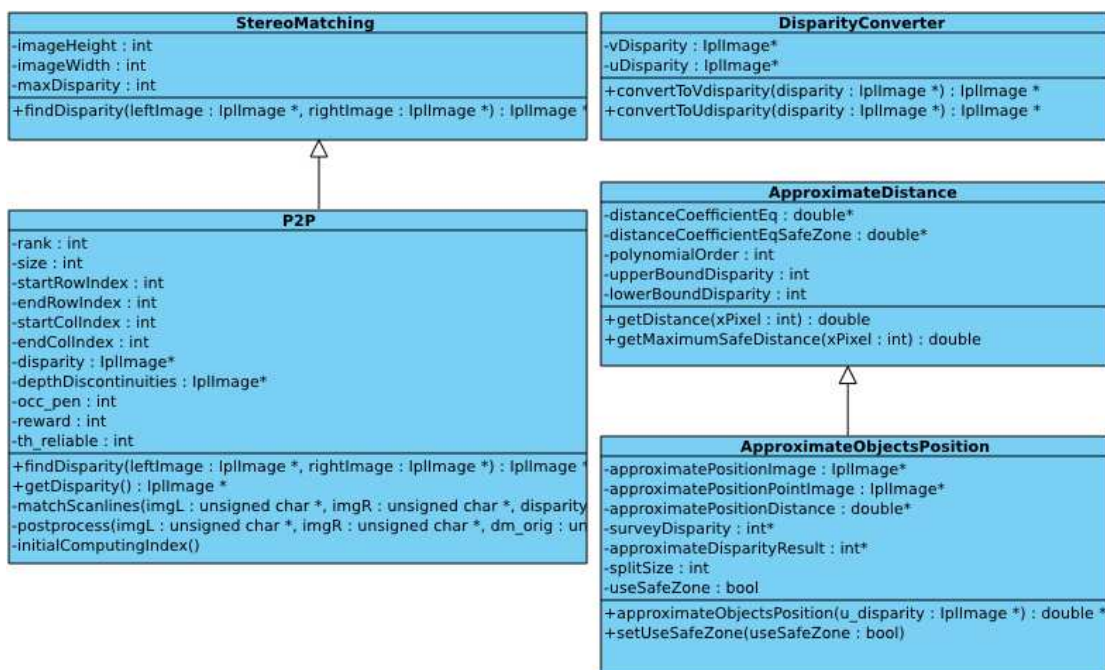
ภาคผนวก ก.

ลักษณะโครงสร้างโปรแกรมพัฒนาขึ้นในวิทยานิพนธ์

## ลักษณะโครงสร้างโปรแกรมพัฒนาขึ้นในวิทยานิพนธ์

ส่วนนี้แนะนำลักษณะโครงสร้างโปรแกรมที่ได้พัฒนาขึ้นเพื่อใช้ในวิทยานิพนธ์ โดยแบ่งออกเป็น 3 แผนภาพคือ

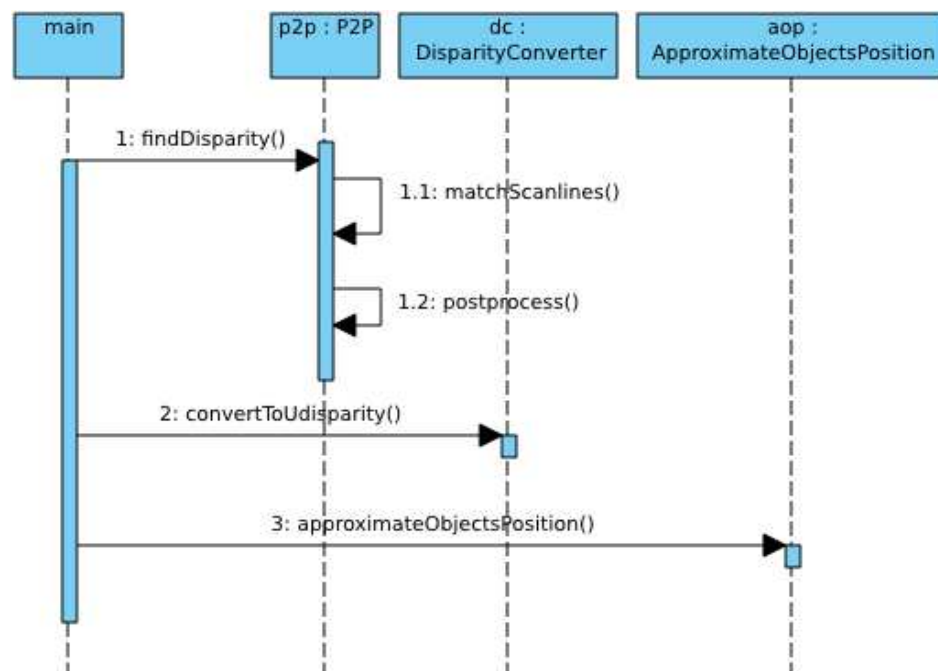
- แผนภาพคลาสที่สร้างขึ้นสำหรับการประมวลผลภาพสเตอริโอและการประมาณค่าระยะทาง (รูปที่ ค-1)
- แผนภาพลำดับการเรียกใช้งานส่วนต่างๆ ในระบบที่พัฒนาขึ้น (รูปที่ ค-2)
- แผนภาพกิจกรรมต่างๆ ที่เกิดขึ้นขณะการประมวลผล (รูปที่ ค-3)



รูปที่ ค-1 แผนภาพคลาสที่สร้างขึ้นสำหรับการประมวลผลภาพสเตอริโอและการประมาณค่าระยะทาง

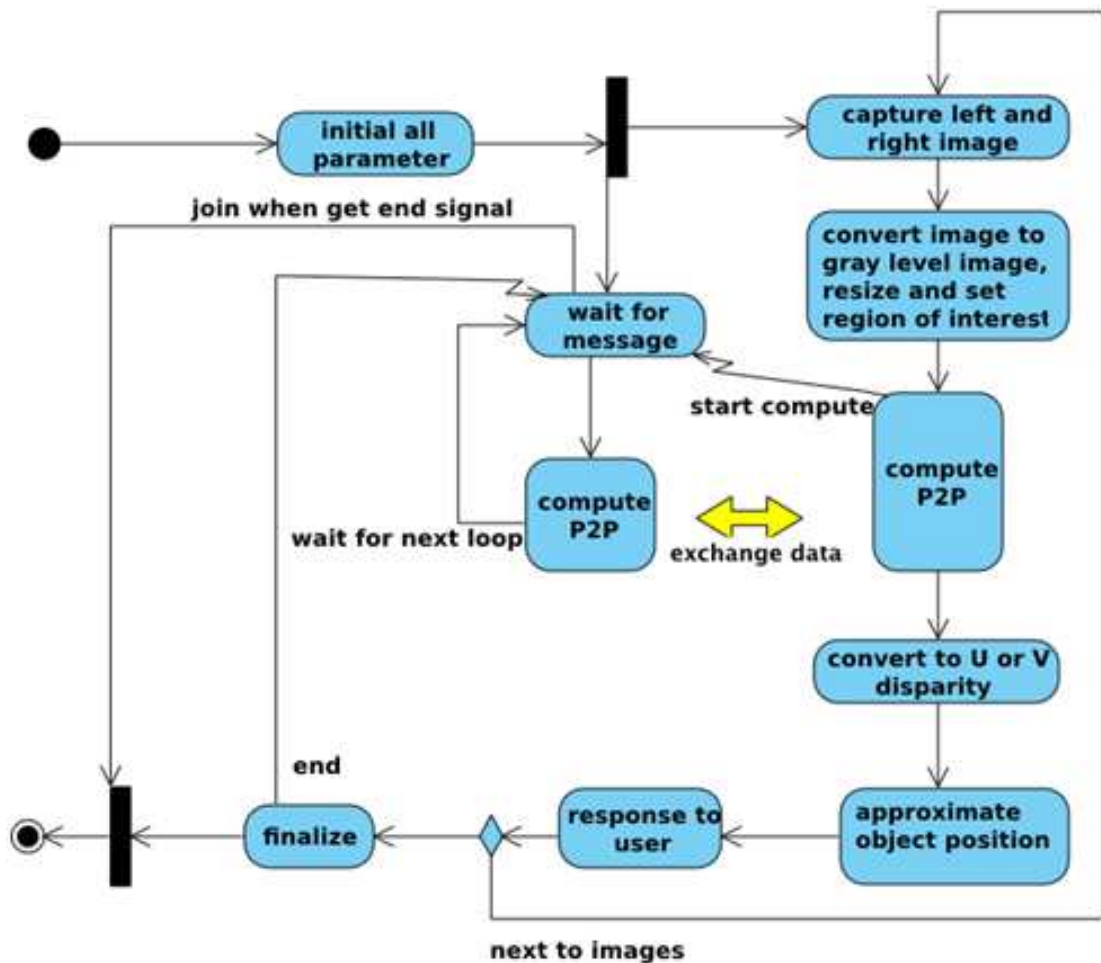
จากรูปที่ ค-1 ประกอบด้วยสามคลาสหลักที่ใช้ในการประมวลผลครั้งนี้โดยมีรายละเอียดคือ P2P ซึ่งเป็นคลาสที่ใช้สำหรับการประมวลผลภาพสเตอริโอโดยใช้อัลกอริทึม P2P ซึ่งในคลาสนี้มีส่วนที่ช่วยให้เกิดการประมวลผลแบบขนานอยู่ภายในแต่ละฟังก์ชันสมาชิก ซึ่งที่สำคัญมีฟังก์ชันสองฟังก์ชันคือ matchScanline และ postprocessing ซึ่งทั้งสองฟังก์ชันนี้ไม่สามารถเรียกใช้ได้โดยตรงจากภายนอกคลาส การเรียกใช้งานฟังก์ชันสลับกันอาจส่งผลให้ได้ผลลัพธ์ที่ผิดพลาด แต่สามารถใช้งานโดยผ่านฟังก์ชัน findDisparity โดยส่งพารามิเตอร์เป็นภาพซ้ายและขวาให้กับ

ฟังก์ชันโดยที่ผลลัพธ์ของฟังก์ชันนี้คือแผนผังดิสพาริตี คลาสต่อมาคือ DisparityConverter ทำหน้าที่แปลงข้อมูลจากแผนผังดิสพาริตีให้อยู่ในรูปของพิกัด u หรือ v คลาสสุดท้ายคือ ApproximateObjectsPosition ทำหน้าที่ตรวจหาวัตถุว่าอยู่บริเวณใดของภาพและประมาณ ระยะทางตามสมการที่กำหนดไว้ ซึ่งคลาส ApproximateObjectsPosition สืบทอดมาจาก ApproximateDistance ซึ่งเป็นคลาสที่ทำหน้าที่คำนวณระยะทางจากระดับดิสพาริตีต่างๆ ที่ต้องการ คลาสที่กล่าวไว้ทั้งหมดนี้มีลำดับการทำงานอย่างคร่าวๆ ดังรูปที่ ค-2



รูปที่ ค-2 แผนภาพลำดับการเรียกใช้งานส่วนต่างๆ ในระบบที่พัฒนาขึ้น

จากรูปที่ ค-2 ในการประมวลผลภาพแต่ละครั้งเริ่มต้นจากการบันทึกภาพจากกล้องวิดีโอ แล้วนำภาพมาเข้ากระบวนการต่างๆ โดยมีลำดับการเรียกใช้งานคลาสดังต่อไปนี้ เรียกใช้งานฟังก์ชันสมาชิก findDisparity จากคลาส P2P เพื่อหาแผนผังดิสพาริตีจากภาพซ้ายและภาพขวา ภายใน findDisparity จะเรียกใช้งาน matchScanline กับ postprocess ตามลำดับ หลังจากได้แผนผังดิสพาริตีมาแล้วปรับเปลี่ยนแผนผังดิสพาริตีให้เป็น U-disparity โดยใช้ฟังก์ชันสมาชิกของคลาส DisparityConverter ชื่อ convertToUdisparity หลังจากนั้นกระบวนการสุดท้ายคือหาตำแหน่งของวัตถุจาก U-disparity ซึ่งกระบวนการนี้เรียกใช้งานฟังก์ชันสมาชิก approximate ObjectsPosition จากคลาส ApproximateObjectsPosition มาถึงกระบวนการนี้จะได้อาร์เรย์ข้อมูลที่เก็บค่าระยะทางต่างๆของวัตถุไว้เรียบร้อยแล้วซึ่งสามารถนำไปเสนอต่อผู้พิจารณาต่อไป



รูปที่ ค-3 แผนภาพกิจกรรมต่างๆ ที่เกิดขึ้นขณะการประมวลผล

รูปที่ ค-3 อธิบายกิจกรรมที่เกิดขึ้นขณะที่ประมวลผล โดยเมื่อเริ่มการทำงานกระบวนการแรกคือทำให้ค่าเริ่มต้นต่างๆ แก่ระบบหลังจากให้ค่าเรียบร้อยแล้วโปรแกรมหลักจะแตกออกเป็นสองโปรเซส (หรือหลายโปรเซสตามที่ต้องการ เนื่องจากวิทยานิพนธ์นี้กล่าวถึงการใช้งานบนคอมพิวเตอร์สองแกนเป็นหลักจึงเน้นการใช้งานสองโปรเซส) โดยที่โปรเซสแรกทำหน้าที่จับภาพซ้ายและขวาเพื่อนำไปเตรียมภาพก่อนการประมวลผลได้แก่ การเปลี่ยนภาพให้อยู่ในระดับสีเทา ปรับขนาดให้ได้ตามที่ต้องการคือ 320x240 พิกเซล แล้วกำหนดค่าพื้นที่สนใจเพื่อให้เหลืออัตราส่วนลักษณะตามต้องการ หลังจากนั้นก็เข้าสู่การคำนวณหาแผนผังดิสพาริตีด้วยอัลกอริทึม P2P โดยเริ่มส่งข้อมูลภาพไปให้โปรเซสที่สองก่อนเริ่มคำนวณ ในขณะที่โปรเซสที่สองจะรอจนกว่าโปรเซสแรกทำงานในขั้นตอนการเตรียมภาพให้เรียบร้อยแล้ว โดยรอจนกว่าจะได้รับ

ข้อมูลจากโปรเซสที่หนึ่งจัดส่งมาให้ แล้วเริ่มการคำนวณ ในการคำนวณจะมีการแลกเปลี่ยนเปลี่ยนข้อมูลกันเป็นระยะๆ เพื่อให้ทั้งสองโปรเซสสามารถคำนวณได้อย่างถูกต้อง เมื่อเสร็จสิ้นการคำนวณแล้วโปรเซสที่สองจะกลับไปสู่สถานะของการรอข้อมูลเพื่อประมวลผลต่อไป ในขณะที่โปรเซสที่หนึ่งทำหน้าที่ในการหาตำแหน่งของวัตถุและประมาณระยะทางต่อไป หลังจากเสร็จสิ้นกระบวนการตรงส่วนนี้กระบวนการต่อไปเป็นการติดต่อกับผู้ใช้งาน เมื่อเสร็จสิ้นแล้วหากต้องการให้ระบบทำงานต่อไปโปรเซสที่หนึ่งจะกลับไปจับภาพใหม่เวียนอย่างนี้เรื่อยๆ แต่หากเลือกจะจบโปรแกรมก็จะเข้าสู่การรวมโปรเซสกลับ เพื่อให้เหลือแค่โปรเซสเดียวแล้วจบโปรแกรม

ภาคผนวก ง.  
ผลงานตีพิมพ์เผยแพร่จากวิทยานิพนธ์

# i-CREATE <sup>2009</sup>

22-26 April • Singapore Management University • Conference Guide • ISBN: 978-981-08-3071-7

WELCOME

CONTENTS

AUTHORS

SEARCH



International **C**onvention on **R**ehabilitation **E**ngineering  
& **A**ssistive **T**echnology

# Low-cost Stereo Vision System for Supporting the Visually Impaired's Walk

Thanathip Limna, Pichaya Tandayya and Nikom Suvanvorn  
Department of Computer Engineering, Faculty of Engineering,  
Prince of Songkla University

Hat Yai, Songkhla 90112, Thailand  
Tel. +66 74287352 Fax. +66 74212895

boatkrap@gmail.com, pichaya@coe.psu.ac.th, kom@coe.psu.ac.th

## ABSTRACT

This paper presents an obstacle detection system for the visually impaired to use with a walking stick. The system in this work can find obstacles applying the depth discontinuities Pixel-to-Pixel (P2P) stereo algorithm which is one of the Intensity-based Stereo Matching (ISM) techniques in stereo vision that can find objects with featureless surface such as whiteboard, door, etc. However, the ISM technique is time consuming and not fast enough for real-time usage. Applying parallel computing using Message Passing Interface (MPI) helps reduce the computing time and enables real-time usage. In this work, we estimate the distance between objects and the visually impaired by applying the V-disparity. Our system can accurately detect objects within the range of 5 meters using 12-centimeter based-line low-cost webcams.

## Categories and Subject Descriptors

I.2.10 [Vision and Scene Understanding]: video analysis

## General Terms

Experimentation

## Keywords

Stereo Vision, Parallel computing, the visually impaired, Pixel-to-Pixel stereo algorithm, Intensity-based Stereo Matching technique, Message Passing Interface, Calibration.

## 1. INTRODUCTION

Normally, a visually impaired person walks to places using a walking stick for checking obstacles, especially which lie on the floor. However, visually impaired people still often get accidents because of hung signs, floating objects, a protection extending from the eaves against rain or sun usually found in Asia, tree branches sticking out into the walk way, etc. A walking stick with an ultrasonic sensor detecting obstacles was made to warn the visually impaired [1]. However, it can only detect whether there are obstacles ahead but cannot inform the distance, size and position of the objects.

There are several inventions using ultrasonic or infrared devices attached to walking sticks and glasses, in order to find the object distance. Although they can detect whether there are obstacles in the current surrounding environments and find the distances, they do not provide sufficient information to enable the visually impaired to continue their journey. The important information lacked is the size.

Stereo vision technology can help solving the problem by finding the distance between the object to the stereo cameras, and the

object's size and position. The accuracy of information depends on the light condition and stereo vision technique.

Electro Neural Vision System (ENVS) [5] is an application of stereo vision for the visually impaired that presents the obstacles and distances via different signals alerting at their ten figures. If the object is close, a high frequency signal will be used. Likewise, if the object is far away, a lower frequency signal will be sent. By this way, the visually impaired can walk to places and avoid obstacles by themselves. However, the Feature-based stereo matching (FSM) techniques [5] used in the ENVS, although can be quickly processed, cannot detect objects with featureless surface which cause dangers to the visually impaired.

On the other hand, the Intensity-based Stereo Matching (ISM) [10] techniques can detect featureless objects. However, there are also problems in applying the techniques in real time due to complex and time consuming image processing techniques and processes, especially when more accuracy is required. However, no reports about using parallel computing with the ISM techniques to reduce computing time have been found.

Parallel computing technology can help reducing the processing time at some parts of stereo vision, i.e., finding the image edges and scan line matching. Message Passing Interface (MPI) [12] offers a parallel programming interface specification that supports several platforms and can specify the number of processes at run time.

Our work applies parallel computing using MPI and of-the-shelf multi-core computers with the Intensity-based Stereo Matching (ISM) techniques [10] in order to reduce computing time. We also investigate for suitable calibration of our low-cost stereo cameras using webcams for supporting walking journeys of the visually impaired.

This paper presents an obstacle detection system for the visually impaired to use with a walking stick. The system finds obstacles applying the depth discontinuities Pixel-to-Pixel (P2P) stereo algorithm [7][8] of the ISM techniques that can detect featureless surface objects and estimate the object distance using V-disparity [9].

The next section presents the background knowledge of stereo vision. The third section describes system design including system overview, low-cost stereo cameras, object distance estimation, enhancing P2P using MPI, and limitations. The final section will be conclusions and future work.



## 2. STEREO VISION

The stereo vision system imitates the visual perception of human being by processing images from a pair of cameras. The human being's vision is 3-Dimensional (3D). Human beings can perceive the object's size and position, and the distance between an object and themselves. Likewise, in the stereo vision system, the 3D images can be processed in order to find an object's size, position and distance.

In this work, we apply stereo disparity to find the depth of the images. The method to find the images' disparity can be classified into two categories [11].

- Intensity-based Stereo Matching techniques (ISM)
- Feature-based Stereo Matching techniques (FSM)

In ISM techniques, the matching process is directly applied to the intensity profiles of the two images. In FSM techniques, features are first extracted from the images and the matching process is applied to the features. In this research, we are interested in the ISM techniques as they match corresponding pixels using the image intensity that can find the disparity even in the case of featureless surface objects. For our work, the most appropriate and recognized algorithm for processing the image disparity in the ISM techniques is the Depth Discontinuities

Stereo (P2P) by Birchfield and Tomasi [7][8] as it appears as one of the OpenCV functions [14] and the source code can be downloaded freely.

In the P2P algorithm, there are two processes, scan line matching and post-process. Scan line matching is a process to match corresponding pixels on left and right images on the same scan lines. In each scan line, processes match corresponding pixels independently from one another. The post-process is a process to exchange data between the scan lines in order to select the best disparity image. This post-process concerns processing data across rows and columns and is not suitable for applying parallel programming. When running the P2P algorithm sequentially on a computer with the following specification, Intel® Core™ 2 Dual 6320 1.86 GHz 1010.7 MB RAM, running on the Linux operation system kernel 2.6.26, at the maximum disparity of 100 and the image size 320x240 pixels, the average computing times are 1.168 seconds for the scan line matching process and 0.165 second for the post-process. As a result, the scan line matching process took most computing time or about 70%. Therefore, we focus on scan line matching in this work.

## 3. SYSTEM DESIGN

This section describes our stereo vision system for the visually impaired as follows.

### 3.1 System Overview

For the safety of the visually impaired, this object detection system is required to give warnings to the visually impaired when there are obstacles in the distance of 10 meters ahead as shown in Figure 1. However, the most critical distance is between 1 meter and 6 meters that is out of the walking stick enough to cover dangerous areas while walking so that the visually impaired can avoid obstacles. This system is not to replace the walking stick but aims to be used in order to provide more confidence with used to.

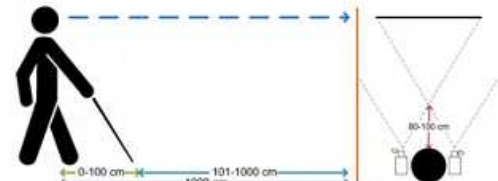


Figure 1. System usage overview.

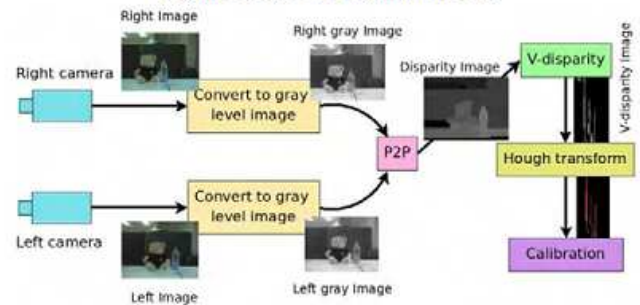


Figure 2. Stereo vision processes

Figure 2 shows the stereo vision processes. First, the color images from left and right cameras will be transformed into gray level image for faster processing and reducing unnecessary information. The disparity image created by the P2P algorithm is a binary image containing pixels that inform the relationship between left and right images. The disparity image is used to find the V-disparity image which is the summary of the disparity values in each scan lines. After that, the next process is finding depth lines from the V-disparity image using the Hough transform algorithm. Then, comparing the lines with the V-disparity can inform the object distances.

### 3.2 Low-cost Stereo Cameras

As normal standard stereo cameras for general stereo vision systems are rather expensive, we applied a pair of much cheaper web cams to be stereo cameras. These low-cost stereo cameras, however, are less accurate than the standard ones and may produce more noises.

Our stereo cameras are Logitech QuickCam webcams for Notebooks Pro [15]. As we want to attach the stereo cameras at the head of the visually impaired like wearing glasses or a helmet, the distance of the base line should not be more than the diameter of the head. In our work, we specify the maximum diameter to be not more than 12 centimeters which is the average head size. The design of the stereo cameras is as shown in Figure 3 and Figure 4 shows how they look like.

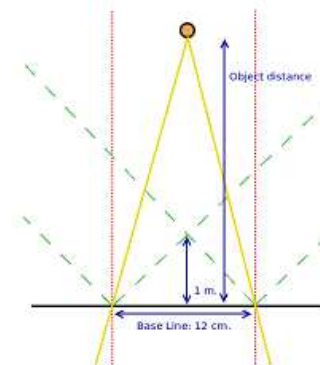


Figure 3. Design of our stereo cameras



Figure 4. Our stereo cameras using webcams

### 3.3 Object Distance Estimation with V-disparity

The disparity image created by the P2P algorithm is a binary image that contains corresponding pixels of left and right images. The V-disparity image is calculated from the summation of the disparities in each scan line. Each vertical straight line in V-disparity image presents a depth distance. The Hough transform algorithm is then used for finding depth lines from the V-disparity image. Finally, the distance information in the depth line will be compared with the V-disparity in order to find the object distance.

## 4. RESULT AND DISCUSSION

### 4.1 Calibration and Distance Estimation

#### 4.1.1 Stereo camera calibration

The V-disparity image transformed from the disparity image contains several straight lines. Each straight line has its length equals to the height of the associated object from the disparity image and is far from the left edge with an  $x$  distance from the stereo cameras to the object distance from the straight line from the by comparing with the distance of the object measured for comparison in prior experiments.

Table 1 shows the relationship between the object distances and V-disparity line pixel positions (minimum, medium and maximum). Each distance range refers to the prior testing dataset. The data is processed to find the medium for distance estimation. The matches between object distances and pixel positions in the V-disparity can be estimated by the 6<sup>th</sup> order polynomial equation shown in Equation 1. Figure 5 shows a graph plotting to fit Equation 1 using the medium dataset. In Equation 1,  $d$  is the distance to the obstacle and  $x$  is the value of the vertical line from the  $x$  axis in the V-disparity.

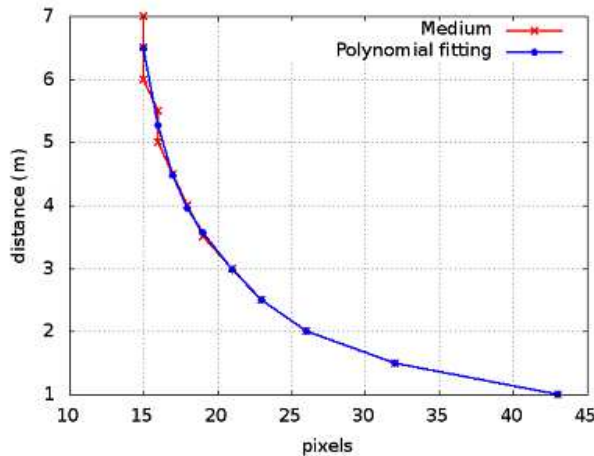


Figure 5. Graph shows the medium dataset fit for a 6<sup>th</sup> order polynomial equation.

Table 1. Matches between real object distances and pixel positions from the V-disparity by finding the medium

Object distance (meters)	Pixel Position		
	Minimum	Medium	Maximum
1	42	43	44
1.5	31	32	34
2	26	26	27
2.5	21	23	24
3	20	21	21
3.5	19	19	20
4	18	18	18
4.5	17	17	17
5	16	16	16
5.5	16	16	16
6	15	15	15
6.5	15	15	15
7	15	15	15

$$d = 1.7938 \times 10^{-6}x^6 - 2.9101 \times 10^{-4}x^5 + 0.019294x^4 - 0.67064x^3 + 12.925x^2 - 131.57x + 559.10 \quad \dots(1)$$

#### 4.1.2 Object distance estimation

In order to estimate the object distance, we use the V-disparity from the P2P algorithm to find straight lines using the openCV function *cvHoughLine* that gives 2 sequence pairs which are the beginning and ending sequence pairs of the straight lines. The value in the  $x$  axis represents the object depth and is used for replacing the variable  $x$  in Equation 1 in order to find the object distance. Currently, our system can find object distances within the range of 5 meters from the stereo cameras.

### 4.2 Enhancing P2P using MPI

Nowadays, multi-core computers are produced and sold widely in the market, i.e., Intel Core 2 Duo, Intel CORE i7, AMD Phenom, AMD Athlon64 X2, etc. In order to fully utilize the CPU, parallel computing technique is needed. In this work, we applied the parallel computing using MPI (MPICH implementation) to help reduce the computing time. In our work, we use the *ch\_shmem* device of the MPICH [13] that is suitable for the symmetric multiprocessors (SMPs) architecture used in the 2-core PC notebook.

The workflow of scan line matching process is suitable for parallelism because it independently computes each scan line. Figure 6 shows the example of the data distribution in the scan line matching process of the P2P algorithm. For each image frame in the scan line matching process, we divide left and right image data into two segments and then distribute them to be separately processed in two processes. The first process computes the top left and right image data, and the second process computes the bottom left and right. After we combine the outputs and create a disparity image.

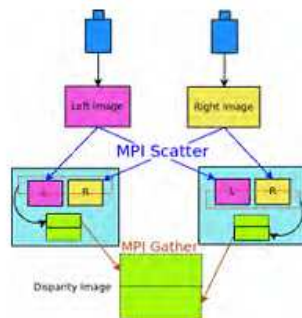


Figure 6. Overview diagram of data distribution

The parallelized P2P algorithm can reduce the total response time of the program to 0.79 second at the maximum disparity of 100, 320x240 pixels when applying two processes on a 2-core computer. The total computing time of the sequential algorithm was 1.365 second per frame. The parallelized version can reduce the computing time to about a half. Using more processes than the number of CPU cores is not recommended since it is not faster.

### 4.3 Limitations

Using the P2P algorithm which is an algorithm using ISM techniques requires the most parallel images so that the algorithm can accurately detect obstacles. Our prototype of stereo cameras is made of webcams that have the maximum based line diameter not more than 12 centimeters in order to enable the attachment to the head of the visually impaired. Consequently, the prototype can accurately detect obstacles only within the range of 5 meters. For further distances than 5 meters, the accuracy of the system is variable. It can detect an object of 65x65 centimeters at the distance of 10 meters but the object appeared so small that may be mixed up with noises. However, using better quality cameras will increase the detecting distance and accuracy.

The speed of the program is 0.790 second for the image of 320x240 pixels at the maximum disparity of 100. It can be used to detect objects with slow movement, i.e., human being's walk. It cannot properly detect objects with fast movement, i.e., car, in real time.

## 5. CONCLUSION

Our obstacle detection for the visually impaired using stereo vision and parallel computing reduces the response time of the depth discontinuities P2P stereo algorithm in the part of scan line matching using MPI running on a 2-core computer. The object distances are found by applying the V-disparity.

Our system can accurately detect objects with slow movement within the range of 5 meters using 12-centimeter based-line low-cost webcams. However, more development and detail experiments are needed to provide more information to the users.

The possible future work can be building a suitable interface for the visually impaired and a warning system in case of getting close to dangerous objects. Applying stereo vision with pattern recognition can also help informing more details of the environment to the users. In the future, when prices allow, using more processes up to the number of cores on a higher performance multi-core computer, says 4-core or 8-core, will certainly reduce much of the computing time.

## 6. ACKNOWLEDGMENTS

The authors are grateful for the support of the PSU Grid Center of Prince of Songkla University and the Intel Company.

## 7. REFERENCES

- [1] Ando, B. Electronic Sensory Systems for the Visually Impaired. *Instrumentation & Measurement Magazine*, IEEE Volume 6, Issue 2, pp. 62 - 67, June 2003.
- [2] Cristina Nicolescu, Pieter Jonker Lecture. Parallel low-level image processing on a distributed-memory system. *Proceedings of the 15 IPDPS 2000 Workshops on Parallel and Distributed Processing*, pp. 226 - 233, 2000.
- [3] Hashimoto, H., Magatani, K., Yanashima, K. The development of the navigation system for visually impaired persons. *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE Volume 2*, pp. 1481 - 1483 vol.2, 25-28 Oct. 2001.
- [4] Helal, A., Moore, S.E., Ramachandran, B. Drishti: an integrated navigation system for visually impaired and disabled. *Wearable Computers, 2001. Proceedings. Fifth International Symposium*, pp. 149 - 156, 8-9 Oct. 2001.
- [5] Meers S., Ward, K. A vision system for providing 3D perception of the environment via transcutaneous electro-neural stimulation. *Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Conference*, pp. 546 - 552, 14-16 July 2004.
- [6] Thomas Bräunl. Tutorial in Data Parallel Image Processing. *Australian Journal of Intelligent Information Processing Systems (AJIIPS)*, vol. 6, no. 3, pp. 164-174 . 2001
- [7] Birchfield, S., Tomasi, C. Depth discontinuities by pixel-to-pixel stereo. *Computer Vision, 1998. Sixth International Conference*. pp. 1073-1080. 4-7 Jan 1998.
- [8] Birchfield, S., Tomasi, C. A pixel dissimilarity measure that is insensitive to image sampling. *Pattern Analysis and Machine Intelligence*, IEEE Transactions. pp. 401-406, Volume: 20, 4, Apr 1998.
- [9] Labayrade, R. Aubert, D. Tarel, J.-P. Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation. *This paper appears in: Intelligent Vehicle Symposium*, IEEE. pp. 646- 651 vol.2, 17-21 June 2002.
- [10] Robyn Owens. Stereo. [http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/OWENS/LECT11/lect11.html](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT11/lect11.html). Available on 4 Jan 2009.
- [11] Stereo vision start with two views. <http://www.vision3d.com/stereo.html>. Available on 4 Jan 2009.
- [12] The Message Passing Interface (MPI) standard. <http://www-unix.mcs.anl.gov/mpi/>. Available on 4 Jan 2009.
- [13] MPICH-A Portable Implementation of MPI. <http://www-unix.mcs.anl.gov/mpi/mpich1/>. available on 4 Jan 2009
- [14] CvAux. <http://opencv.willowgarage.com/wiki/CvAux#FindStereoCorrespondence>. Available on 4 Jan 2009.
- [15] Scott Clark. Logitech QuickCam for Notebooks Pro. [http://www.everythingusb.com/logitech\\_quickcam\\_for\\_notebooks\\_pro.html](http://www.everythingusb.com/logitech_quickcam_for_notebooks_pro.html). Available on 4 Jan 2009.

**the Sixth International Joint Conference on Computer Science and Software  
Engineering (JCSSE2009)  
May 13-15, 2009, at Laguna Beach Hotel, Phuket, Thailand.**

## Enhancing the Depth Discontinuities by Pixel-to-Pixel Algorithm Using MPI

Thanathip Limna and Pichaya Tandayya  
 Department of Computer Engineering, Faculty of Engineering,  
 Prince of Songkla University, Hat Yai, Songkhla, 90112 Thailand  
 Email: boatkrap@gmail.com, pichaya@coe.psu.ac.th

### Abstract

*Stereo vision, especially the Feature-based Stereo Matching (FSM) technique, has been used to enable the visually impaired people to detect obstacles with feature surface. However, it cannot help the visually impaired to detect objects with featureless surface. The Intensity-based Stereo Matching (ISM) technique can detect objects in with featureless surface but it takes longer computing time to process and cannot be used in real-time situation. This paper presents an investigation on the enhancement of the Depth Discontinuities by Pixel-to-Pixel algorithm using parallel computing on a 2-core personal computer and an 8-core server. The results show that parallel computing using Message Passing Interface (MPI) significantly reduces the response time and it is possible to use the ISM technique in real-time.*

**Key Words:** MPI, Parallel computing, Depth Discontinuities by Pixel-to-Pixel, ISM, Multi-core

### 1. Introduction

There are many ideas to help the visually impaired with navigation. Electro Neural Vision System (ENVS) [1] is an application of stereo vision for the visually impaired by presenting the obstacles and distances via different signals alerting at their ten fingers. If the object is close, a signal with a high frequency will be used. Likewise, if the object is far away, a signal with a lower frequency will be sent. In this way, the visually impaired can walk to places and avoid obstacles by themselves. However, the technique of the Feature-based Stereo Matching (FSM) [2] used in the ENVS, although can be quickly processed, cannot detect objects with featureless surface which usually cause dangers to the visually impaired.

On the other hand, the Intensity-based Stereo Matching (ISM) [2] techniques can detect objects with featureless surface and else. However, it takes

much longer computing time and requires running on a computer with high specification. Personal computers nowadays have higher specification than before with acceptably less expensive prices. The market trend is now for 2-core and 4-core Central Processing Units (CPUs) and prices are getting cheaper. Sequential processing cannot fully utilize the maximum performance of multi-core CPUs. Parallel processing techniques are used for enhancing the performance of applications running on multi-core CPUs.

So far, there are two methods in parallel programming, including using threads and processes. In our system, the safety of the visually impaired is the key. Using threads, the programmer somehow cannot predict the order of tasks to be processed and cannot specify the number of processes at run-time while parallel programming using Message Passing Interface (MPI) [3][4] has a better process management. MPICH which is one of MPI implementations that has the *ch\_shmem* device that is suitable for the Symmetric Multiple Processors (SMPs) architecture [5] that is used in our stereo vision system. It is also quite convenient to change to other architectures. The source code can be recompiled with other device configurations and used on different architectures.

Using parallel computing to help reducing processing time in the stereo vision techniques is interesting and optimistic. Most image processing algorithms including stereo vision can be parallelized. There are a few standards and tools in parallel computing. MPI is used in this work as it is widely known, well documented and stable.

In this paper, we present a novel investigation on computing time reduction when applying the parallel computing approach to enhance the Depth Discontinuities by Pixel-to-Pixel stereo (P2P) [6][7] algorithm running on a 2-core personal computer and an 8-core server in order to see the example performances of current PCs and future computers.

In the first section, background knowledge of stereo vision, the Depth Discontinuities by Pixel-to-Pixel Stereo algorithm, and Message Passing Interface will be given. Then, the next section will discuss the designed parallel algorithm used in the investigation. After that the experiments will be described and the results will be analyzed. The final section will be the conclusion and future work.

## 2. Background

### 2.1 Stereo Vision

Human beings can perceive a 3-dimensional (3D) vision through two eyes. Each eye captures the image separately and sends the vision signals to be processed at the brain. When the two images arrive at the back of the brain, the brain will merge the two images into one [8]. This process is also similar in stereo vision. Stereo vision can process two images to find the distances between the video cameras and the objects. Stereo vision also includes the image segmentation. However, stereo vision system needs high enough processing power in order to produce the results in real time because it needs to process images from two cameras.

Stereo matching approaches to the corresponding problem can be broadly be classified into two categories. [2]

- Intensity-based Stereo Matching (ISM)
- Feature-based Stereo Matching (FSM)

In ISM, the matching process is applied directly to the intensity profiles of the two images. In FSM, features are first extracted from the images and the matching process is applied to the features. This paper focuses at the Depth Discontinuities by Pixel-to-Pixel Stereo (P2P) of ISM which is also provided as an OpenCV function [9] as it can detect featureless objects. This approach is used in this work for a guiding system for the visually impaired. Figure 1 shows an example disparity image of the stereo vision system.



**Figure 1** The left image, right image and disparity image

### 2.2 Depth Discontinuities by Pixel-to-Pixel Stereo

Depth Discontinuities by Pixel-to-Pixel Stereo (P2P) [4] is an algorithm that can produce the disparity image from stereo images using Intensity-based Stereo Matching techniques which match

pixels on paired scan lines applying a cost function in order to find the M sequence as shown in Equation 1 [4].

$$\gamma(M) = N_{occ}K_{occ} - N_mk_r + \sum_{i=1} d(x_i, y_i) \dots (1)$$

From Equation 1,  $K_{occ}$  is the occlusion penalty constant,  $k_r$  is the match reward constant,  $d(x_i, y_i)$  is the distance between pixel  $x_i$  and pixel  $y_i$ , and, finally,  $N_{occ}$  and  $N_m$  are the occlusion and the number of related pairs respectively.

The P2P algorithm processes left and right images projecting on the same scan lines. The process can be divided into 2 parts which are scan line matching and post-process algorithms. The data processing for each pair of scan lines in the matching algorithm is independent. On the other hand, the post process algorithm needs to compute data from rows and columns together which is difficult to be parallelized.

Therefore, in this work, data parallelism is applied only in the process of matching scan lines.

### 2.3 Message passing interface

Message passing interface (MPI) is a standard and specification for the message passing library designed for parallel computing. MPI implementations are on C and FORTRAN. MPI has the role to facilitate parallel processing by distributing tasks and data to be processed at processing units or processors for higher performance. MPI is applied in various works such as computational science that analyze high quantity of data; i.e., finding relationships of base sequence in human DNA, finding drug formula for destroying cancer cells, animation rendering and image processing.

MPICH is one of the MPI implementations [3]. MPICH supports 4 different computer architectures by providing 4 devices as follows.

- *ch\_p4* device for Workstation Networks
- *ch\_p4mpd* device for Workstation Networks and Clusters
- *ch\_shmem* device for Shared Memory Processors
- *globus2* device for Grids

In our case, the suitable device is *ch\_shmem* for a 2-core personal computer and an 8-core server which are SMPs that share memories together.

## 3. Parallel P2P Algorithm

In order to compare sequential computing and parallel computing, we tested the sequential and parallel algorithms on the same computers, a 2-core personal computer and an 8-core server. The 2-core

computer specification is Intel® Core™ 2 Dual 6320 1.86 GHz with 1010.7 MB RAM running on the Linux Operating System Kernel 2.6.26. The 8-core server specification is Intel® Xeon® E5440 2.83 GHz with 8200 MB RAM running on the Linux Operating System Kernel 2.6.26.

The P2P algorithm has been added as a function in the OpenCV library. When using this algorithm with motion pictures, especially in case of big images or images with a high maximum disparity, it will take a long time. Figure 2 shows the experimental results of the parallel P2P algorithm on a personal computer. The image size is 320x240 pixels.

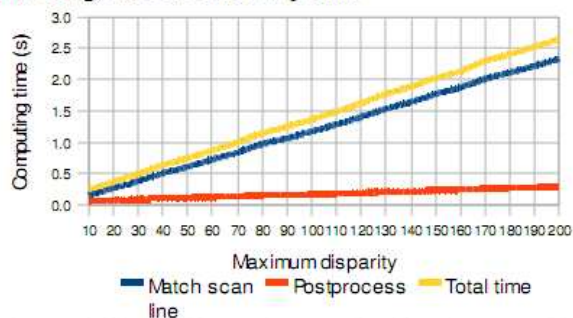


Figure 2 Graph shows the relationship between the computing time and maximum disparity of the sequential computing P2P algorithm running on a 2-core computer.

In Figure 3, the program is designed to be able to run on a computer with more than two processors. The user can specify the number of process at run time. The scan lines are used for work distribution. The scatter command is used for distributing portions of left and right images to processes. Each process will manage the same number of scan lines. For example, for the 320x240 pixels and 4 processes, each process will match scan lines from left and right images for 80 rows. After the matching, the disparity image and depth discontinuities will be gathered at Process 0 before moving to the next step of calculation. Figure 4 shows the pseudo code of the parallel program mentioned above.

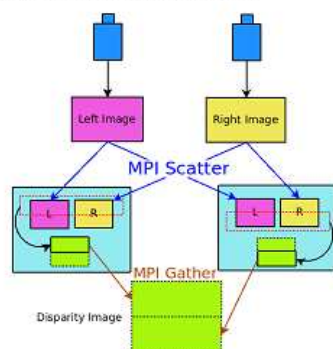


Figure 3 Overview diagram of data distribution [10].

```

startCalc : starting number of lines for
computing
endCalc : last number of lines for computing
ROWS : number of rows of image
id : process id
size : number of all processes
startCalc := ROWS*id/size;
endCalc := (ROWS*id+ROWS)/size;
scatter left and right images at Process 0
to other processes from startCalc to
ROWS/size
MPI::Scatter(imgL);
MPI::Scatter(imgR);
for (scanline = startCalc;
    scanline < endCalc; scanline++){

    scan line matching code

}
gather disparity image and depth
discontinuities image from all processes to
Process 0
MPI::Gather(disparity_map);
MPI::Gather(depth_discontinuities);
  
```

Figure 4 Pseudo code of our parallel program.

## 4. Experimental Results

The testing has been done on two types of computers. First is on a 2-core computer in order to find a suitable number of processes in real use. Second is on an 8-core server in order to test the relationship between the number of cores and the speedup of the program.

### 4.1 Running on a 2-core computer

The image size is 320x240 pixels. The result for sequential computing is shown in Figure 2. Scan line matching takes up most of the computing time. The results for parallel computing using MPICH with the *ch\_shmem* device are shown in Figure 5 and Figure 6.

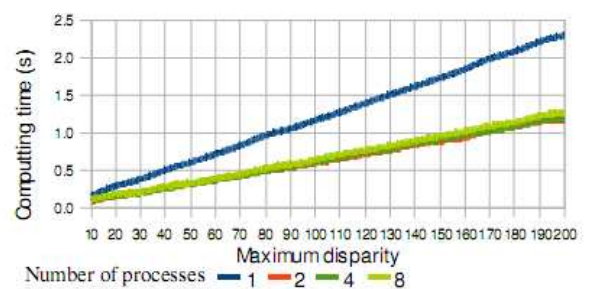
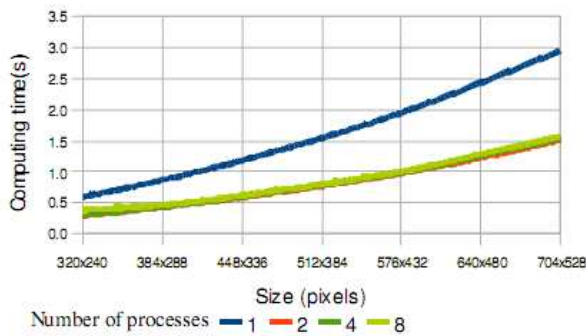


Figure 5 Relationship between the computing time and the maximum disparity. Graph lines are the number of processes of the parallel computing scan line matching of the P2P algorithm using MPICH and a 2-core computer.



**Figure 6 Relationship between the computing time and the size of images. Graph lines are the number of processes of the parallel computing scan line matching of the P2P algorithm using MPICH and an 8-core server.**

In Figure 5, the computing time is reduced if the number of processes is higher than 1 process. However, when the number of processes is more than 2, the computing time is longer than that of 2 processes. The computing time of the scan line matching of the maximum disparity of 100 is significantly reduced for 49.23% or 0.575 second that be effective in real use.

**4.2 Running on an 8-core computer**

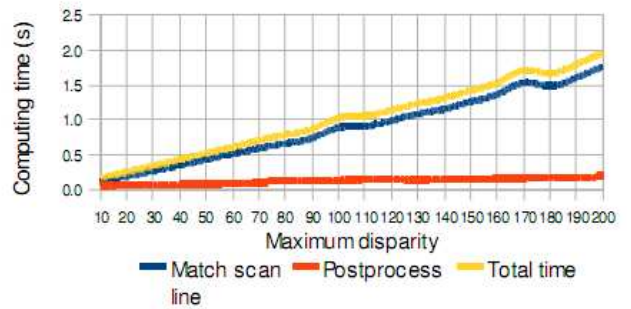
The image size is 320x240 pixels. For sequential computing, the result is shown in Figure 7. The result is similar to the result running on a 2-core computer that scan line matching is the main computing load. Figure 8 and Figure 9 show the results of parallel computing. The computing time is also reduced when the number of processes is higher as shown in Figure 7. The computing time at the maximum disparity of 100 is reduced for 0.779 second or 88.21% when comparing sequential computing and parallel computing with 2 processes.

**Table 1 average computing time of serial computing and parallel computing on a 2-core computer and an 8-core server**

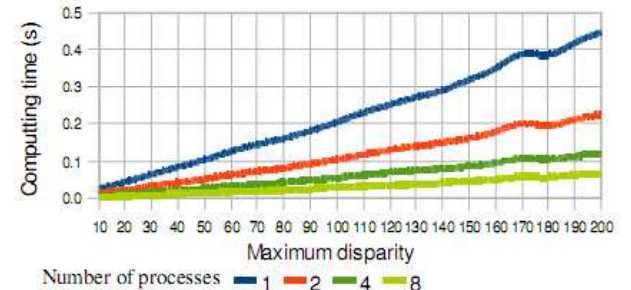
Type of computer	Computing time (s)			
	Sequence computing	Parallel computing with		
		2 processes	4 processes	8 processes
2-core	1.168308994	0.593164271	0.602813994	0.636913673
8-core	0.883306714	0.104114369	0.055412835	0.028512324

From the experimental running on the 2-core computer at the maximum disparity of 100 and the image size of 320x240 pixels, the average sequential computing time of the scan line matching of the P2P algorithm is 1.168 seconds while the average parallel computing time with 2 processes is 0.593 second. For

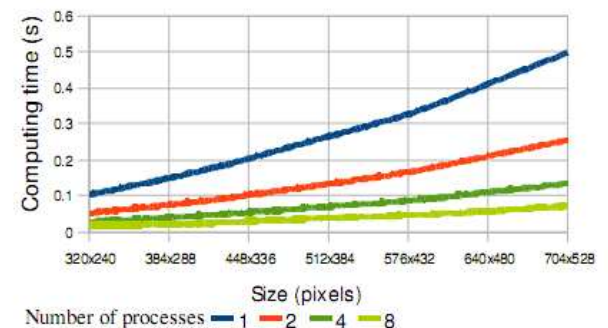
the 8-core server, the average sequential computing time of the scan line matching of the P2P algorithm is 0.883 second while the average parallel computing time with 2 processes is 0.104 second. Increasing the number of processes more than 2 in the 2-core computer cannot reduce the computing time while increasing the number of processes in the 8-core server up to 8 can significantly reduce the computing time.



**Figure 7 Graph shows the relationship of the computing time and maximum disparity of the sequential computing P2P algorithm on an 8-core server.**



**Figure 8 Relationship between the computing time and the maximum disparity. Graph lines are the number of processes of the parallel computing scan line matching of the P2P algorithm using MPICH and an 8-core server**



**Figure 9 Relationship between the computing time and the image size. Graph lines are the number of processes of the parallel computing scan line matching of the P2P algorithm using MPICH and an 8-core server**



## 5. Result Analysis

In order to apply the parallel computing P2P algorithm in real-time, there are 3 factors to consider, including the response time, the image size and the image disparity. In case of the response time, the computing time that is less than 1 second is preferable. In order to detect obstacles within the range of 1-6 meters from the video cameras, the acceptable image disparity that can be computed within the response time about 1 second is in the range of 100-150. Finally, the suitable image sizes for the acceptable disparity in this case are in between 320x240 pixels to 512x384 pixels.

The results show that the number of processes to be run in parallel on multi-core computers should not exceed the number of cores in order to provide best performance as according to the parallel computing theory.

## 6. Conclusion

Our work shows that applying parallel computing using MPI on multi-core computers significantly reduces the computing time in the P2P depth discontinuities algorithm of the ISM technique and enables the usage of the ISM to detect the technique and enables the usage of the ISM to detect the obstacles for the visually impaired in real-time. The future work will then be the calibration of the stereo system.

## 7. Acknowledgment

The authors are grateful for the support of PSU Grid Center of Prince of Songkla University, Thai National Grid Center, and Intel Company.

## 8. References

- [1] Meers S., Ward, K. "A vision system for providing 3D perception of the environment via transcutaneous electro-neural stimulation," *Information Visualisation, 2004. Proceedings. Eighth International Conference*, pp. 546-552, 14-16 Jul 2004.
- [2] Robyn Owens. "Stereo" [http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/OWENS/LECT11/lect11.html](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT11/lect11.html). Available on 4 Jan 2009.
- [3] "MPICH-A Portable Implementation of MPI," <http://www-unix.mcs.anl.gov/mpi/mpich1/>. Available on 4 Jan 2009.
- [4] "MPICH Documents," <http://www-unix.mcs.anl.gov/mpi/mpich1/docs.html>. Available on 4 Jan 2009.
- [5] "Symmetric multiprocessing," [http://en.wikipedia.org/wiki/Symmetric\\_multiprocessing](http://en.wikipedia.org/wiki/Symmetric_multiprocessing). Available on 4 Jan 2009.
- [6] Birchfield, S., Tomasi, C. "Depth discontinuities by pixel-to-pixel stereo," *Computer Vision, 1998. Proceedings. Sixth International Conference*. pp. 1073-1080. 4-7 Jan 1998.
- [7] Birchfield, S., Tomasi, C. "A pixel dissimilarity measure that is insensitive to image sampling," *Pattern Analysis and Machine Intelligence, IEEE Transactions*. pp 401-406, Volume: 20, 4, Apr 1998.
- [8] "Stereo vision start with two views", <http://www.vision3d.com/stereo.html>. Available on 4 Jan 2009.
- [9] "CvAux," <http://opencv.willowgarage.com/wiki/CvAux#FindStereoCorrespondence>. Available on 4 Jan 2009.
- [10] Limna T., Tandayya P., Suvanvorn N. "Low-cost Stereo Vision System for Supporting the Visually Impaired's Walk". *International Convention on Rehabilitation Engineering and Assistive Technology, 2009. Proceedings. Third International Conference*. 22-26 Apr 2009. To be published.

## ประวัติผู้เขียน

ชื่อ สกุล	นายธนชาติป ลิ้มนา	
รหัสประจำตัวนักศึกษา	5010120028	
วุฒิการศึกษา		
วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วิศวกรรมศาสตรบัณฑิต (วิศวกรรมคอมพิวเตอร์)	มหาวิทยาลัยสงขลานครินทร์	2550

## ทุนการศึกษา (ที่ได้รับในระหว่างการศึกษา)

ทุนบัณฑิตศึกษาระดับปริญญาโท สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
มหาวิทยาลัยสงขลานครินทร์

## การตีพิมพ์เผยแพร่ผลงาน

- T. Limna, P. Tandayya, and N. Suvanvorn, “Low-cost stereo vision system for supporting the visually impaired's walk,” in *Proceedings of the 3rd International Convention on Rehabilitation Engineering & Assistive Technology*, Singapore: ACM, 2009, pp 1-4.
- T. Limna and P. Tandayya, “Enhancing the Pixel-to-Pixel Depth Discontinuities Algorithm Using MPI,” in *Proceedings of the 6th International Joint Conference on Computer Science and Software Engineering (JCSSE2009)*, Phuket, Thailand, 13<sup>th</sup> – 15<sup>th</sup> May 2009, pp 86-90.