

โปรแกรมเสมือนสำหรับการวิเคราะห์และฟื้นฟูผู้ป่วยที่มีปัญหาช่วงไหล่และแขน
Virtual Reality for Upper Extremity Rehabilitation

นางสาวสโรชา กมลรังษีพร
MS.Sarocho Kamolrungsriporn
นางสาวอัญธิกา อ่อนพุทธา
MS.Antika Onputtha

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมชีวการแพทย์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2560

โปรแกรมเสมือนสำหรับการวิเคราะห์และฟื้นฟูผู้ป่วยที่มีปัญหาช่วงไหล่และแขน

Virtual Reality for Upper Extremity Rehabilitation

นางสาวสโรชา กมลรังษีพร

MS.Sarocho Kamolrungsriporn

นางสาวอัญธิกา อ่อนพุทรา

MS.Antika Onputtha

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมชีวการแพทย์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2560

Virtual Reality for Upper Extremity Rehabilitation

MS.Sarocha Kamolrungrsriporn

MS.Antika Onputtha

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE DEGREE OF BACHELOR OF ENGINEERING IN BIOMEDICAL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2017

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญาโท

หัวข้อปริญญาโท	โปรแกรมเสมือนสำหรับการวิเคราะห์และฟื้นฟูผู้ป่วยที่มีปัญหาช่วงไหล่และแขน Virtual Reality for Upper Extremity Rehabilitation
นักศึกษา	นางสาวโรชา กมลรังษีพร รหัสนักศึกษา 57011322 นางสาวอัญธิกา อ่อนพุทธา รหัสนักศึกษา 57011519
หลักสูตร	วิศวกรรมศาสตรบัณฑิต สาขาวิชาชีวการแพทย์
อาจารย์ผู้ควบคุมปริญญาโท	

(ดร. สุธเดช ตริไตรลักษณะ)

หัวข้อปริญญานิพนธ์	โปรแกรมเสมือนสำหรับการวิเคราะห์และฟื้นฟูผู้ป่วยที่มี ปัญหาช่วงไหล่และแขน
นักศึกษา	นางสาวสโรชา กมลรังษีพร นางสาวอัษฎิกา อ่อนพุทธา
หลักสูตร	วิศวกรรมศาสตรบัณฑิต สาขาวิชาชีวการแพทย์
ปีการศึกษา	2560
อาจารย์ผู้ควบคุมปริญญานิพนธ์	ดร. สุรเดช ตรีไตรลักษณะ

บทคัดย่อ

โรคที่เกี่ยวกับส่วนของ upper extremity ของร่างกายนั้นมีอยู่หลายโรคด้วยกัน อาทิ โรคเอ็นหัวไหล่อักเสบ โรคข้อไหล่ติด เป็นต้น ผู้ป่วยในกลุ่มนี้จะมีความผิดปกติในการเคลื่อนไหวของข้อต่อบริเวณไหล่หรือการเคลื่อนที่ของแขน ซึ่งในปัจจุบันนี้มีผู้ป่วยที่มีความผิดปกติบริเวณ upper extremity มากขึ้นเรื่อยๆ อาจเนื่องมาด้วยสาเหตุที่แตกต่างกัน อย่างไรก็ตามแนวทางหลักๆ ในการรักษาโรคในกลุ่มนี้คือการทำกายภาพบำบัดอย่างต่อเนื่อง เพื่อฟื้นฟูศักยภาพของร่างกายให้กลับมาใกล้เคียงกับคนปกติอีกครั้งหนึ่ง ซึ่งในการทำกายภาพบำบัดนั้นต้องกระทำอย่างสม่ำเสมอและมักจะใช้เวลาในการรักษาเป็นระยะเวลานาน หนึ่งในวิธีการวินิจฉัยอาการนั้นคือการดูเส้นทางการเคลื่อนที่ของมือของผู้ป่วยว่ามีลักษณะใกล้เคียงกับคนปกติมากน้อยเพียงใด โดยจะนำเส้นทางการเคลื่อนที่ของมือของผู้ป่วยมาเทียบกับของคนปกติก่อนและหลังการทำกายภาพบำบัดทุกครั้ง เพื่อนำไปใช้ในการวางแผนการรักษาของนักกายภาพบำบัดต่อไป โดยเครื่องที่ใช้ในการเก็บข้อมูลนี้เป็นเครื่องที่มีราคาแพง ดังนั้นแล้วงานวิจัยชิ้นนี้จึงได้สร้างโปรแกรมที่ใช้ในการวิเคราะห์เส้นทางการเคลื่อนที่ของมือแบบง่ายขึ้นมา เพื่อลดค่าใช้จ่ายและเก็บผลการรักษาเบื้องต้นได้ ซึ่งผลลัพธ์ที่ได้นั้นได้เปรียบเทียบระหว่างเส้นทางการเคลื่อนที่ของมือของผู้ป่วยก่อนและหลังการทำกายภาพ พบว่ามีความแตกต่างกันและแสดงแนวโน้มของอาการที่ดีขึ้น

Thesis Title	Virtual Reality for Stroke Rehabilitation
Student	Sarocho Kamolrungsriporn Antika Onputtha
Degree	Bachelor of Engineering in Biomedical Engineering King Mongkut's Institute of Technology Ladkrabang
Academic Year	2017
Thesis Advisor	Dr.Suradej Tretriluxana

Abstract

There are many diseases that concerning the upper extremity of the body ,for example the frozen shoulder, Shoulder Impingement etc.The patients get the problem to move their joints on the shoulder and their arm.Nowadays a number of the patients tends to increase.The main method to release the diseases is on continuously physical therapy and must take a long time Physiotherapist use the device (that use the 3D magnetic field) to record and guide the movement of hands (it called hand path).In this way physical therapist can check and compare between hand path in normal cases , abnormal cases , before the physical therapy and after physical therapy. But this device is very expensive. So, this research has created a program that is used to analyze the hand path easier and low cost.The result was a comparison between the hand movement of the patient before and after physical therapy. The findings were different and showed a trend of better symptoms.

กิตติกรรมประกาศ

ขอขอบคุณที่ปรึกษาวิทยานิพนธ์ ดร. สุรเดช ตรีไตรลักษณ์ ที่คอยชี้แนะและให้คำปรึกษามาโดยตลอด 1 ปีการศึกษาที่ผ่านมา จนกระทั่งงานวิจัยนี้เสร็จสิ้นลง ไม่ว่าจะในเรื่องแนวทางการดำเนินงาน การแก้ไขปัญหาระหว่างการดำเนินงาน ตลอดไปจนถึงประสานงานเพื่อให้กลุ่มของข้าพเจ้าสามารถไปเก็บข้อมูลจากผู้ป่วยจริงได้

ขอขอบคุณ รศ.ดร.ชูชาติ ปิณฑวิรุจน์ ที่คอยให้คำแนะนำในเรื่องต่างๆ ตลอดทุกสัปดาห์ ทำให้กลุ่มของข้าพเจ้าได้แนวคิดในการทำงานวิจัยที่หลากหลายมากยิ่งขึ้น

ขอขอบคุณ ผศ.ดร.กิติพล ชิตสกุล ที่คอยเสนอแนะแนวทางแก้ไขปัญหาต่างๆ คอยให้คำแนะนำเพิ่มเติมจากอาจารย์ที่ปรึกษา นอกจากนี้ยังคอยให้ความช่วยเหลือในด้านของอุปกรณ์ในการทำวิจัยโดยตลอด

ขอขอบคุณ รศ.ดร.จารุกุล ตรีไตรลักษณ์ คณะกายภาพบำบัด มหาวิทยาลัยมหิดลที่ช่วยให้คำปรึกษาในด้านการใช้งานและการทำงานของนักกายภาพบำบัด ทำให้งานวิจัยชิ้นนี้ตรงกับความต้องการของผู้ใช้ (นักกายภาพบำบัด) มากที่สุด อีกทั้งยังเอื้อเพื่อสถานที่และติดต่อกับผู้ป่วยเพื่อให้กลุ่มของข้าพเจ้าได้ไปเก็บตัวอย่างในสถานที่จริง

ขอขอบคุณครอบครัวที่คอยสนับสนุนการทำงานมาโดยตลอด โดยเฉพาะการเก็บข้อมูลจากตัวอย่างผู้ป่วยจริง เนื่องจากในครอบครัวมีผู้ป่วยเป็นโรคที่เกี่ยวข้องกับงานวิจัยชิ้นนี้ ทำให้การเก็บข้อมูลนั้นทำได้ง่ายขึ้น

สุดท้ายนี้ขอขอบคุณผู้ป่วยที่มาทำกายภาพบำบัดที่มหาวิทยาลัยมหิดล ที่ยอมสละเวลามาร่วมทำการทดลองกับกลุ่มของข้าพเจ้า ทำให้กลุ่มของข้าพเจ้าสามารถเก็บข้อมูลจากผู้ป่วยจริงได้

นางสาวสโรชา กมลรังสีพร

นางสาวอัญธิกา อ่อนพุทธา

สารบัญ

บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญ(ต่อ).....	จ
สารบัญตาราง	ฉ
สารบัญรูป	ช
สารบัญรูป(ต่อ).....	ซ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา	1
1.3 สมมุติฐานของการศึกษา.....	2
1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย	2
1.5 ขอบเขตการวิจัย.....	2
1.6 ขั้นตอนของการศึกษา.....	3
1.7 โครงสร้างวิทยานิพนธ์.....	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 Back ground subtraction.....	4
2.2 Edge Detection	14
2.2.1 Gradient and Edge Detection	16
2.2.2 Laplacian method	19
2.2.3 Canny Edge Detection	22
2.3 Color detection	27

สารบัญ(ต่อ)

2.3.1 HSV (Hue Saturation Value).....	27
2.3.2 RGB (Red Green Blue).....	31
2.3.3 Color detection Method	33
2.4 Python.....	36
2.5 Open CV.....	37
บทที่ 3 การออกแบบระบบ	39
3.1 แนวคิดของระบบที่นำเสนอ	39
3.2 ส่วนเก็บบันทึกผล	40
3.3 ส่วนวิเคราะห์ผล	42
บทที่ 4 การทดสอบและผลการทดสอบ	49
4.1 ทดสอบในกรณีของผู้ป่วยโรคไหล่ติด (frozen shoulder).....	49
4.1.1 ข้อมูลเบื้องต้นของผู้ป่วยที่ทำการทดสอบ	49
4.1.2 วิธีทำการทดสอบและผลเมื่อเทียบกับคนปกติ	50
4.1.3 วิธีทำการทดสอบและผลการทดสอบของผู้ป่วยเปรียบเทียบระหว่างก่อนการทำกายภาพบำบัด และหลังการทำกายภาพบำบัด.....	52
4.2 ทดสอบในกรณีของผู้ป่วยโรคเส้นเลือดในสมองตีบ (Brain Infarction).....	53
4.2.1 ข้อมูลเบื้องต้นของผู้ป่วยที่ทำการทดสอบ	53
4.2.2 วิธีทำการทดสอบและผลการทดสอบ	53
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	57
5.1 สรุปผลการดำเนินการ	57
5.2 ปัญหาระหว่างการศึกา.....	59
5.3 ข้อเสนอแนะ.....	59
เอกสารอ้างอิง	60

สารบัญตาราง

ตารางที่ 4. 1 แสดงการยกแขนของคนปกติเทียบกับการยกแขนของผู้ป่วย	49
ตารางที่ 4. 2 แสดงการเปรียบเทียบผลลัพธ์ของผู้ป่วยและคนปกติ	51
ตารางที่ 4.3 แสดงการทดสอบก่อนการทำกายภาพบำบัดและหลังการทำกายภาพบำบัด	52
ตารางที่ 4.4 แสดงผลการทดสอบครั้งที่ 1.....	55
ตารางที่ 4.5 แสดงผลการทดสอบครั้งที่ 2.....	55
ตารางที่ 4.6 แสดงผลการทดสอบครั้งที่ 3.....	56
ตารางที่ 4.7 แสดงค่าความแตกต่างของเวลาระหว่างเครื่องจริงและโปรแกรม	56

สารบัญรูป

รูปที่ 2.1 แสดงภาพถ่ายแบบปกติ.....	4
รูปที่ 2.2 แสดงภาพที่ผ่านกระบวนการ BACKGROUND SUBTRACTION	5
รูปที่ 2.3 ภาพแสดงกระบวนการการตัดภาพพื้นหลัง	6
รูปที่ 2.4 การไล่ระดับสีของ GRADIENT	16
รูปที่ 2.5 แสดงภาพของวิธีการ PREWITT (FILE DICOM).....	17
รูปที่ 2.6 แสดงภาพของวิธีการ SOBEL (FILE DICOM)	18
รูปที่ 2.7 แสดง MASK MATRIX ชนิด 3x3.....	20
รูปที่ 2.8 แสดงระบบ LOG	21
รูปที่ 2.9 ภาพแสดงผลลัพธ์.....	21
รูปที่ 2.10 แสดงผลลัพธ์ของ CANNY.....	27
รูปที่ 2.11 แสดงระบบสี HSV	28
รูปที่ 2.12 ความหมายของ HUE ใน HSV มีผลต่อการแปรปรวนของรูปหกเหลี่ยมเป็นวงกลม	29
รูปที่ 2.13 ใน HSV ความอิ่มตัวของสีเป็นเพียงการปรับค่าความเข้มของสีเพื่อเติมช่วง [0, 1] สำหรับการ รวมกันของ HUE และ LIGHTNESS หรือ VALUE	30
รูปที่ 2.14 ADDITIVE COLORS (RGB)	32
รูปที่ 2.15 แสดงผลลัพธ์ของ EROSION	34
รูปที่ 2.16 แสดงผลลัพธ์ของ DILATION.....	34
รูปที่ 2.17 แสดงผลลัพธ์ของการใช้ฟังก์ชัน IN RANGE โดยในภาพจะเป็นการจับสีแดง.....	35
รูปที่ 2.18 แสดงผลลัพธ์ของ CONTOUR APPROXIMATION METHOD.....	35
รูปที่ 3.1 ภาพแสดงการนั่งของผู้ป่วยก่อนเริ่มการเก็บผล.....	39
รูปที่ 3.2 ภาพการทำงานของโปรแกรมในส่วนเก็บบันทึกผล	40
รูปที่ 3.3 ภาพแสดงหน้าจอในขั้นตอนการเก็บผล	40
รูปที่ 3.4 ภาพแสดงหน้าจอเก็บผลจากโปรแกรมจริง	41
รูปที่ 3.5 แสดงการเคลื่อนที่ของแขนและความเจ็บปวดของผู้ป่วย	42
รูปที่ 3.6 ภาพแสดงการทำงานของโปรแกรมในส่วนของการประมวลผล	42
รูปที่ 3.7 ภาพปกติก่อนใช้ BACKGROUND SUBTRACTION	43

สารบัญรูป(ต่อ)

รูปที่ 3.8 ภาพแสดงผลลัพธ์ของ BACKGROUND SUBTRACTION	43
รูปที่ 3.9 ภาพปกติก่อนใช้ GAUSSIAN FILTER	44
รูปที่ 3.10 ภาพแสดงผลลัพธ์ของ GAUSSIAN FILTER	44
รูปที่ 3.11 ภาพแสดงปริซึมของรหัสสีแบบ HSV	45
รูปที่ 3.12 ภาพขาวดำเทาปกติก่อนใช้ EROSION	45
รูปที่ 3.13 ภาพแสดงผลลัพธ์ของ EROSION	46
รูปที่ 3.14 ภาพขาวดำเทาปกติก่อนใช้ DILATION	44
รูปที่ 3.15 ภาพแสดงผลลัพธ์ของ DILATION	46
รูปที่ 3.16 ภาพแสดงผลลัพธ์ของ DILATION	47
รูปที่ 3.17 ภาพแสดงหน้าจอแสดงผลในส่วนของการประมวลผลข้อมูล	48
รูปที่ 4.1 ภาพแสดงการนั่งของผู้เข้ารับการทดสอบจากหน้าจอแสดงผล.....	50
รูปที่ 4.2 ภาพแสดงการเคลื่อนที่ที่ผู้เข้ารับการทดสอบต้องทำบนหน้าจอแสดงผล	50
รูปที่ 4.3 ภาพแสดงการทดสอบโดยใช้เครื่อง MOTION MORNITOR.....	53
รูปที่ 4.4 ภาพแสดงการทดสอบโดยใช้โปรแกรม.....	54
รูปที่ 5.1 รูปเปรียบเทียบก่อนและหลังทำกายภาพ.....	58

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

คำจำกัดความของคำว่า Upper extremity ในงานวิจัยชิ้นนี้คือ บริเวณส่วนบนของร่างกาย อาทิ ไหล่ แขน เป็นต้น ซึ่งในปัจจุบันนี้ผู้ป่วยที่มีความผิดปกติในบริเวณของ upper extremity นั้นมีจำนวนเพิ่มสูงขึ้นมาก โดยอาจมีสาเหตุมาจากพฤติกรรมประจำวันของผู้ป่วย หรือเป็นความผิดปกติของร่างกาย ในส่วนของอาการนั้นผู้ป่วยจะมีความผิดปกติในด้านการเคลื่อนไหวของอวัยวะในส่วนของ upper extremity กล่าวคือเมื่อเทียบกับคนปกติแล้วผู้ป่วยจะไม่สามารถใช้งานอวัยวะในส่วนนี้ได้อย่างเต็มประสิทธิภาพ เช่น หากมีความผิดปกติบริเวณหัวไหล่แล้วผู้ป่วยจะไม่สามารถเคลื่อนไหวแขนได้อย่างมั่นคงและอิสระ ในผู้ป่วยบางรายอาจไม่สามารถยกมือขึ้นจรดศีรษะได้ เป็นต้น จากความผิดปกติดังกล่าวนี้สามารถรักษาและเยียวยาอาการได้ด้วยยา และการทำกายภาพบำบัด ซึ่งการทำกายภาพบำบัดนั้นมักจะ ต้องใช้เวลายาวนานในการรักษา ซึ่งในการทำกายภาพบำบัดในแต่ละครั้งจำเป็นต้องเก็บผลอาการเอาไว้ เพื่อดูแนวโน้มของอาการและการวางแผนการรักษาในอนาคตของนักกายภาพบำบัด ดังนั้นงานวิจัยชิ้นนี้ จึงถูกจัดทำขึ้นเพื่อช่วยในการเก็บบันทึกผลของอาการ ไม่ว่าจะเป็นการเก็บผลเพื่อเปรียบเทียบกับคนปกติ หรือการเก็บผลเพื่อเปรียบเทียบอาการก่อนและหลังการทำกายภาพบำบัดก็ตาม โดยงานวิจัยชิ้นนี้ทำการ วิเคราะห์ผลแบบออฟไลน์และจะแบ่งส่วนการทำงานออกเป็นสองส่วนคือ หนึ่งในส่วนของการอัดวิดีโอ ใน ส่วนนี้จะใช้กล้องจับการเคลื่อนไหวของผู้ป่วย โดยที่ผู้ป่วยจะทำการเคลื่อนที่ตามตำแหน่งของวัตถุเสมือน ที่ถูกยึดอยู่กับที่ในหน้าจอแสดงผล เมื่อทำการบันทึกเสร็จเรียบร้อยแล้วจึงนำเข้ามาในส่วนที่สองคือส่วนของการประมวลผลเพื่อแสดงเส้นทางการเคลื่อนที่ของมือ (hand path) ในระนาบ 2 มิติและเวลา เพื่อเก็บ เป็นข้อมูลต่อไป

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

จุดประสงค์ของงานวิจัยสามารถสรุปได้ดังนี้

1. ศึกษาลักษณะอาการของความผิดปกติในส่วนของ upper extremity
2. ศึกษาลักษณะและวิธีการทำกายภาพของผู้ป่วยที่มีความผิดปกติในส่วนของ upper extremity

3. ศึกษาหลักการและงานวิจัยเกี่ยวกับส่วนของ upper extremity เพื่อใช้เป็นแนวทางในการพัฒนา งานวิจัย

4. ทดสอบระบบกับผู้ป่วยจริงเพื่อดูผลลัพธ์และเปรียบเทียบกับอุปกรณ์ที่ใช้จริง

1.3 สมมุติฐานของการศึกษา

จากความเป็นมาและความสำคัญของปัญหาข้างต้นจะเห็นได้ว่า ในการทำกายภาพบำบัดนั้น จำเป็นต้องมีการบันทึกผลอยู่เป็นประจำเพื่อใช้ในการเปรียบเทียบผลการรักษาในระยะยาว ดังนั้นแล้วนัก กายภาพจึงจำเป็นต้องมีอุปกรณ์ที่ช่วยในการเก็บบันทึกและวิเคราะห์ผลดังกล่าว

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

จากสมมุติฐานของการศึกษาข้างต้น งานวิจัยนี้จึงได้นำเสนอโปรแกรมสำหรับช่วยในการ วิเคราะห์ผลและบันทึกผลก่อนและหลังทำกายภาพบำบัด หรือเปรียบเทียบอาการกับคนปกติ โดยยึดมั่น ในหลักการของการวิเคราะห์ผลเบื้องต้นเพื่อความสะดวกในการใช้งาน สามารถใช้ได้ทุกที่และมีราคาถูก แนวคิดเหล่านี้ถูกนำมาพัฒนาเป็นโปรแกรมสำหรับการหาเส้นทางการเคลื่อนที่ของมือ (hand path) เพื่อให้ นักกายภาพได้วิเคราะห์และดูผลได้อย่างง่ายดายมากยิ่งขึ้น

1.5 ขอบเขตการวิจัย

งานวิจัยนี้ทำขึ้นเพื่อพัฒนาโปรแกรมที่ใช้ในการเก็บและวิเคราะห์ผลเส้นทางการเคลื่อนที่ของมือ เพื่อให้ นักกายภาพได้ใช้ในการดูพัฒนาการของผู้ป่วย ความก้าวหน้าในการรักษา รวมไปถึงการวางแผนการรักษาล่วงหน้าในอนาคตอีกด้วย โปรแกรมนี้เป็นโปรแกรมที่ใช้วิเคราะห์ผลแบบออฟไลน์ซึ่งประ ครอบคลุมไปด้วยสองส่วนหลักๆ

ส่วนแรกคือส่วนที่ใช้ในการบันทึกผล ในส่วนนี้จะเป็นการอัดวิดีโอจากอุปกรณ์ (โน้ตบุ๊กหรือ webcam) เพื่อเก็บผลจากผู้ป่วย ในส่วนนี้เราจะให้ผู้ป่วนั่งอยู่ตรงหน้ากล้องและให้มองที่จอแสดงผลเป็น หลัก บนหน้าจอจะปรากฏกรอบสี่เหลี่ยมสองกรอบ เพื่อเป็นจุดอ้างอิงในการเคลื่อนที่ของมือของผู้ป่วย จากจุดหนึ่งไปอีกจุดหนึ่ง นอกจากนี้ยังต้องให้ผู้ป่วยถือลูกบอลสีแดงเอาไว้เพื่อใช้เป็นตัวอ้างอิงตำแหน่งใน ขั้นตอนวิเคราะห์ผลด้วย

ในส่วนถัดมาคือส่วนที่ใช้ในการวิเคราะห์ผล ส่วนนี้เราจะนำวิดีโอมาแสดงเส้นทางการเคลื่อนที่ของมือ ตัวโปรแกรมจะตรวจจับสีแดงเพื่อหาตำแหน่ง coordinate ก่อนที่จะทำการสร้างจุดตามลูกบอลสีแดงเพื่อ แสดงเส้นทางการเคลื่อนที่ออกมาให้เห็น เวลาในการเคลื่อนที่จะถูกแสดงที่มุมด้านล่างของจอ

1.6 ขั้นตอนของการศึกษา

ขั้นตอนการศึกษาเริ่มต้นจากการศึกษาค้นคว้าทฤษฎีและงานวิจัยเกี่ยวกับลักษณะของความผิดปกติของอวัยวะบริเวณ upper extremity รวมทั้งศึกษาโปรแกรมและอุปกรณ์ต่างๆ ที่ใช้ในการเขียนโปรแกรม เพื่อนำมาประยุกต์ให้เหมาะสมและเพื่อประโยชน์สูงสุดในการเก็บและวิเคราะห์ผลของเส้นทางการเคลื่อนที่ของมือ (Hand path) ได้ จากนั้นจึงนำไปทดสอบกับผู้ป่วยเพื่อดูผลลัพธ์ที่เกิดขึ้น

1.7 โครงสร้างวิทยานิพนธ์

ผลการศึกษาและวิจัยได้รวบรวมมานำเสนอในงานวิจัยฉบับนี้โดยแบ่งเนื้อหาเป็นบทตอน ดังนี้

บทที่ 1 กล่าวถึงความเป็นมาและความสำคัญของปัญหา ความมุ่งหมายและวัตถุประสงค์ ขอบเขตการวิจัยขั้นตอนการศึกษา เพื่อให้รู้ถึงขอบข่ายการทำงานวิจัยทั้งหมดนี้

บทที่ 2 กล่าวถึงทฤษฎีที่ใช้ในงานวิจัยทั้งหมด ทั้งการตรวจจับสีและการประมวลผลภาพ

บทที่ 3 กล่าวถึงแนวคิดที่ใช้ในการทำและนำเสนอ ระบบต่างๆ ที่ใช้ในงานวิจัยชิ้นนี้

บทที่ 4 กล่าวถึงการทดสอบและผลของการทดสอบในงานวิจัยชิ้นนี้

บทที่ 5 กล่าวถึงบทสรุปของงานวิจัย ปัญหาที่เกิดขึ้นและข้อเสนอแนะแนวต่างๆ เพื่อใช้ในการพัฒนางานวิจัยชิ้นอื่นๆ ต่อไปในอนาคต

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

เนื้อหาส่วนนี้ประกอบไปด้วยหลักการและวิธีการที่ใช้ในงานวิจัย รวมทั้งข้อมูลต่าง ๆ เกี่ยวกับโปรแกรม กระบวนการวิเคราะห์ข้อมูล การแสดงผลทางด้าน Motion detection

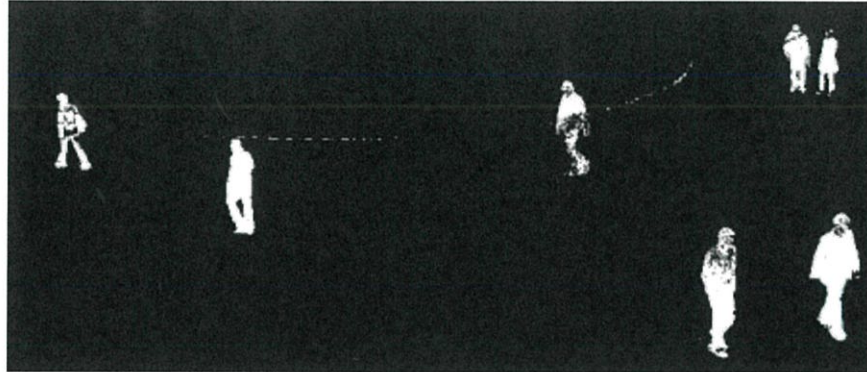
2.1 Back ground subtraction [1]

Back ground subtraction เป็นการลบภาพพื้นหลัง (Back ground) หรือที่เรียกว่าการตรวจหาภาพเบื้องหน้า (foreground) เป็นเทคนิคในด้านการประมวลผลภาพและการมองเห็นด้วยคอมพิวเตอร์ โดยจะทำการแยกภาพเบื้องหน้าออกจากภาพพื้นหลัง ภาพเบื้องหน้าส่วนใหญ่จะเป็นภาพประเภทวัตถุที่จับต้องได้ (มนุษย์ รถยนต์ ข้อความ ฯลฯ)

Back ground subtraction นี้เป็นวิธีการที่ใช้กันอย่างแพร่หลายในการตรวจจับการเคลื่อนที่ของวัตถุ โดยใช้การพิจารณาคือจากการเคลื่อนไหวของวัตถุ เพื่อหาความแตกต่างระหว่างเฟรมปัจจุบัน (current frame) หรือ ภาพเบื้องหน้า และเฟรมอ้างอิง (reference frame) หรือภาพพื้นหลัง เฟรมอ้างอิงนั้นจะมีลักษณะต่างจากเฟรมปัจจุบันอย่างเห็นได้ชัด เนื่องจากเฟรมปัจจุบันจะมีการเคลื่อนไหว ในขณะที่เฟรมอ้างอิงจะหยุดนิ่ง ไม่มีความเคลื่อนไหวหรือเป็นภาพนิ่ง ทำให้ง่ายต่อการแยกเฟรมทั้งสองออกจากกัน



รูปที่ 2.1 แสดงภาพถ่ายแบบปกติ



รูปที่ 2.2 แสดงภาพที่ผ่านกระบวนการ background subtraction

สำหรับอัลกอริทึมที่ใช้นั้นมีหลากหลายรูปแบบ โดยอัลกอริทึมเหล่านั้นควรจะเป็นอัลกอริทึมที่สามารถใช้ได้ในทุกๆ ความเปลี่ยนแปลงของเฟรม กล่าวคือในการวิเคราะห์และประมวลผลภาพนั้น ภาพที่นำมาประมวลผลจะมีความเปลี่ยนแปลงอย่างต่อเนื่อง อาทิ การเปลี่ยนทิศทางของแสง การเคลื่อนที่ที่เปลี่ยนแปลงไป การเปลี่ยนแปลงในระยะยาว เป็นต้น เพื่อดูความเปลี่ยนแปลงนี้เราจะใช้ฟังก์ชัน $V(x,y,t)$ เพื่อให้เห็นความเป็นไปในช่วงระยะเวลาที่กำหนด โดยตัวแปร t แทนค่าของเวลา และตัวแปร x และ y แทนตำแหน่งของพิกเซลในภาพ

Basic Motion Detection (Basic)

การตรวจจับการเคลื่อนไหว อย่างง่ายคือ การทำให้ภาพ back ground กลายเป็นภาพที่มีสีโทนเดียวหรือเป็นภาพขาวดำ ซึ่งต้องตัดการเคลื่อนไหวทั้งหมดออกไป, เป็นภาพ back ground ที่ไม่มีการเคลื่อนไหว หรืออาจประมาณผ่าน median filter [12, 17, 34] ซึ่งภาพดังกล่าวนี้จะต้องระวังต่อการเปลี่ยนแปลงของแสงโดยต้องมีการปรับปรุงภาพ ดังนี้

$$B_{s,t+1} = (1 - \alpha)B_{s,t} + \alpha \cdot I_{s,t}$$

เมื่อคือ α คงที่ ค่าที่ได้จะอยู่ในช่วง $[0,1]$ ซึ่ง Basic Motion Detection นี้จะอาศัยภาพพื้นหลังที่ปราศจากการเคลื่อนไหวและความสอดคล้องของพิกเซลของวัตถุที่มีการเคลื่อนไหวเบื้องหน้า การแยกภาพ back ground ออก และสามารถตรวจจับวัตถุได้โดยอาศัยค่า threshold และฟังก์ชันของระยะทาง

$$d_0 = |I_{s,t} - B_{s,t}|$$

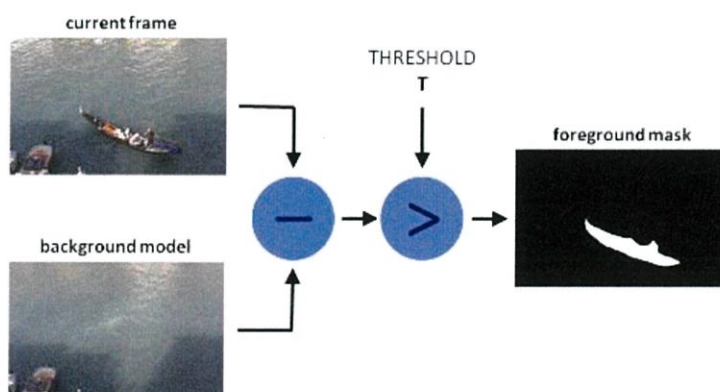
$$d_1 = |I_{s,t}^R - B_{s,t}^R| + |I_{s,t}^G - B_{s,t}^G| + |I_{s,t}^B - B_{s,t}^B|$$

$$d_2 = (I_{s,t}^R - B_{s,t}^R)^2 + (I_{s,t}^G - B_{s,t}^G)^2 + (I_{s,t}^B - B_{s,t}^B)^2$$

$$d_\infty = \max\{|I_{s,t}^R - B_{s,t}^R|, |I_{s,t}^G - B_{s,t}^G|, |I_{s,t}^B - B_{s,t}^B|\}$$

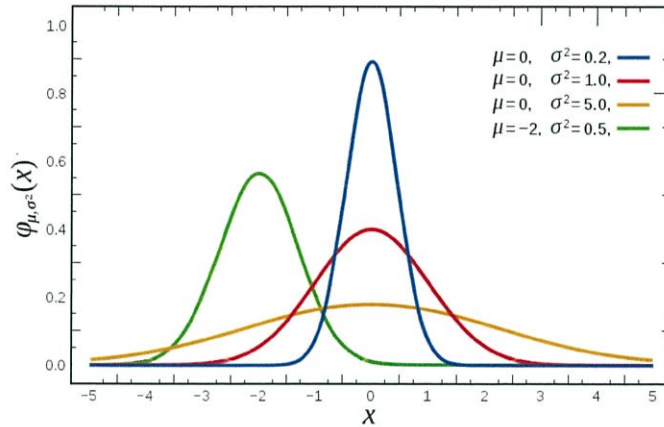
โดยให้ R, G, B แทน Red, Green, Blue Channel และ d_0 เป็นค่าของ operating ของภาพ grayscale

หากใช้เฟรมก่อนหน้า $t-1$ แทนภาพ Back ground B ด้วยการกำหนดค่านี้การตรวจจับการเคลื่อนไหวจะกลายเป็นการจับการเปลี่ยนแปลงระหว่างเฟรม กระบวนการที่มีประสิทธิภาพต่อการเปลี่ยนแปลงของแสง แต่อาจก่อให้เกิดจากปัญหาที่ทำการตรวจจับวัตถุได้เพียงบางส่วนเท่านั้น



รูปที่ 2.3 ภาพแสดงกระบวนการการตัดภาพพื้นหลัง

One Gaussian (1-G)



เป็นการสร้าง Back Ground ด้วยภาพเดี่ยวเช่นเดียวกับ Basic Motion Detection เพียงแต่จำเป็นต้องใช้ภาพพื้นหลังที่ปราศจากสัญญาณรบกวนใดๆ กล่าวคือภาพพื้นหลังที่นำมาใช้จะต้องเป็นภาพที่มีความเสถียรที่สูงมาก ดังนั้นการทำงานนี้จึงไม่สามารถใช้ได้ในความเป็นจริง เพราะการกำจัดสัญญาณรบกวนให้หมดไปอย่างหมดจดนั้นเป็นไปได้ยาก ดังนั้นจึงได้เกิดการจำลองแต่ละพิกเซลด้วย probability density function (PDF) โดยอาศัยความเกี่ยวข้องกันในแต่ละเฟรม ในกรณีนี้ปัญหาของ Basic Motion Detection กลายเป็น PDF-thresholding เนื่องจากพิกเซลมีความสอดคล้องกับพิกเซลของวัตถุที่มีการเคลื่อนที่ต่ำ อาทิ เมื่อต้องการอธิบายถึง noise จะมีการ model ทุกภาพพื้นหลังด้วย Gaussian distribution $(\mu_{s,t}, \Sigma_{s,t})$ ที่ $\mu_{s,t}$ และ $\Sigma_{s,t}$ สำหรับสีพื้นหลังเฉลี่ยและเมทริกซ์ความแปรปรวนร่วมกันที่พิกเซล s และเวลา t ซึ่งได้ออกมาเป็นดังนี้

$$d_G = \frac{1}{2} \log((2\pi)^3 |\Sigma_{s,t}|) + \frac{1}{2} (\mathbf{I}_{s,t} - \boldsymbol{\mu}_{s,t}) \Sigma_{s,t}^{-1} (\mathbf{I}_{s,t} - \boldsymbol{\mu}_{s,t})^T$$

$$d_M = |\mathbf{I}_{s,t} - \boldsymbol{\mu}_{s,t}| \Sigma_{s,t}^{-1} |\mathbf{I}_{s,t} - \boldsymbol{\mu}_{s,t}|^T$$

เนื่องจากเมทริกซ์ความแปรปรวนร่วมนั้นจะมีขนาดใหญ่เมื่ออยู่ในที่มีสัญญาณรบกวนมาก และมีขนาดเล็กเมื่ออยู่ในที่มีสัญญาณรบกวนน้อยหรือไม่มีสัญญาณรบกวน ดังนั้น Σ จะทำให้ค่าของ threshold ขึ้นอยู่กับปริมาณของ noise ในอีกแง่หนึ่ง ยิ่งทำให้พิกเซลมีขนาดใหญ่ ค่าของ temporal gradient ก็จะมีมากตามไปด้วย $\|I_{s,t} - \mu_{s,t}\|$ โดยที่พิกเซลนั้นต้องมีการเคลื่อนไหว ทั้งหมดนี้ทำให้วิธีการนี้มีประสิทธิภาพมากกว่าการใช้วิธีการจับการเคลื่อนไหวแบบพื้นฐาน (Basic motion detection) เนื่องจากแสงมักมีการเปลี่ยนแปลงอยู่ตลอดเวลา ดังนั้นแล้วค่าเฉลี่ยและค่าแปรปรวนร่วมของแต่ละพิกเซลสามารถปรับปรุงได้ตามสมการนี้

$$\begin{aligned}\mu_{s,t+1} &= (1 - \alpha) \cdot \mu_{s,t} + \alpha \cdot I_{s,t} \\ \Sigma_{s,t+1} &= (1 - \alpha) \cdot \Sigma_{s,t} \\ &\quad + \alpha \cdot (I_{s,t} - \mu_{s,t})(I_{s,t} - \mu_{s,t})^T\end{aligned}$$

Minimum, Maximum and Maximum Inter-Frame Difference (MinMax)

อีกวิธีหนึ่งคือการปรับปรุงระบบให้เข้ากับ noise ซึ่งเรียกว่าระบบ W^4 (คิดค้นโดย Haritaogluet) ทุกพิกเซลของพื้นหลัง (background pixel) s มักจะมีต่ำสุดสุด (m_s) ค่าสูงสุด (M_s) และความต่างสูงสุดของเฟรมต่อเนื่อง (maximum of consecutive frames difference) (D_s) โดยดูได้ตามช่วงเวลา วิธีการ MinMax นี้เป็นวิธีการที่เป็น “Static” หรือก็คือเป็นวิธีการที่คงที่ ไม่มีการแปรผัน ในทุกๆ พิกเซล s ซึ่งมีปริมาณเป็น $I_{s,t}$ ซึ่งจะเป็นไปตามเกณฑ์ดังต่อไปนี้

$$|M_s - I_{s,t}| < \tau d_\mu \quad \text{or} \quad |m_s - I_{s,t}| < \tau d_\mu$$

โดยที่ τ คือ threshold ที่กำหนดโดยผู้ใช้และ d_μ เป็นค่ามัธยฐานที่มากที่สุดของ interframe ของภาพทั้งหมด ซึ่งวิธีดังกล่าวนี้มีความคล้ายคลึงกับ 1-G พิกเซลในที่มี noise มากต้องการความแปรปรวนที่มากกว่าเพื่อให้เกิดการเคลื่อนไหว มากกว่าบริเวณที่มี noise น้อย (Stable) ในกรณีนี้แต่ละ

พิกเซลพื้นหลัง (Back ground pixel) มีความเกี่ยวข้องกับสามค่าทั้งสามอย่างมากซึ่งจะถูกแทนที่ด้วยเวกเตอร์เฉลี่ยและเวกเตอร์แปรปรวน ซึ่งอัลกอริทึมเดิมนั้นทำงานเฉพาะภาพวิดีโอที่เป็นภาพระดับเทาเท่านั้น (Gray scale) เท่านั้น ในมกมีการสูญเสียข้อมูลเมื่อเทียบกับลำดับวิดีโอ ผู้เขียนกล่าวถึงว่าพื้นหลังสามารถปรับปรุงได้ด้วยวิธีการ pixel-based และ object-based

Gaussian Mixture Model (GMM)

พิจารณาพื้นหลังที่เป็นพื้นหลังประเภทพื้นผิวที่มีการเคลื่อนไหว (เช่นคลื่นในน้ำหรือต้นไม้ที่ถูกลมพัด) ผู้เขียนบางคนเสนอให้ใช้ไฟล์ PDFs ของ Stauffer และ Grimson ตัวอย่างเช่นให้ทุกพิกเซลประกอบไปด้วย K Gaussians สำหรับวิธีการนี้ความน่าจะเป็นที่จะทำให้เกิดสีที่พิกเซลจะถูกกำหนดโดย

$$P(\mathbf{I}_{s,t}) = \sum_{i=1}^K \omega_{i,s,t} \cdot \mathcal{N}(\boldsymbol{\mu}_{i,s,t}, \boldsymbol{\Sigma}_{i,s,t})$$

ที่ $\mathcal{N}(\boldsymbol{\mu}_{i,s,t}, \boldsymbol{\Sigma}_{i,s,t})$ ให้ i^{th} Gaussian model และ $\omega_{i,s,t}$ เป็นน้ำหนัก สำหรับวัตถุประสงค์ในการคำนวณตามที่ Stauffer และ Grimson เสนอโดยสมมติว่าเมทริกซ์ความแปรปรวน $\boldsymbol{\Sigma}_{i,s,t}$ เป็นเส้นทแยงมุมได้ $\boldsymbol{\Sigma} = \boldsymbol{\sigma}^2 \text{Id}$ ในพารามิเตอร์ของคอมโพเนนต์ที่เข้าคู่กัน (คือ Gaussian ที่ใกล้ที่สุดซึ่งเป็น t อยู่ในค่าเบี่ยงเบนมาตรฐาน 2.5) ซึ่งสามารถปรับปรุงได้ตามสมการนี้

$$\omega_{i,s,t} = (1 - \alpha)\omega_{i,s,t-1} + \alpha$$

$$\boldsymbol{\mu}_{i,s,t} = (1 - \rho) \cdot \boldsymbol{\mu}_{i,s,t-1} + \rho \cdot \mathbf{I}_{s,t}$$

$$\sigma_{i,s,t}^2 = (1 - \rho) \cdot \sigma_{i,s,t-1}^2 + \rho \cdot d_2(\mathbf{I}_{s,t}, \boldsymbol{\mu}_{i,s,t})$$

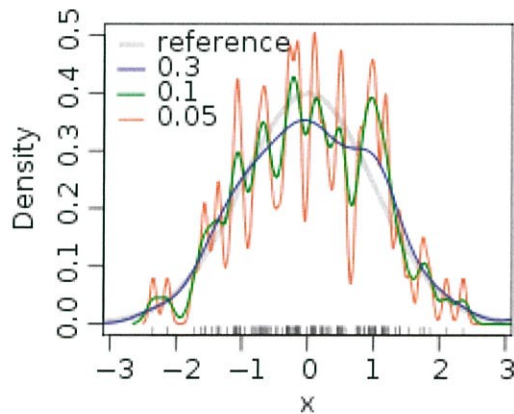
ที่ α ที่กำหนด ρ เป็น โดย $\rho = \alpha \mathcal{N}(\boldsymbol{\mu}_{i,s,t}, \boldsymbol{\Sigma}_{i,s,t})$ และ d_2 เป็นระยะทางที่กำหนดไว้ พารามิเตอร์ $\boldsymbol{\mu}$ และ $\boldsymbol{\sigma}$ ของการกระจายที่ไม่เหมือนกันยังคงเป็นเหมือนเดิม ในขณะที่น้ำหนักจะลดลงดังนี้ $\omega_{i,s,t} = (1 - \alpha) \omega_{i,s,t-1}$ เมื่อใดก็ตามที่ไม่มีองค์ประกอบตรงกันของสี $I_{s,t}$ ซึ่งเป็นหนึ่งที่มีน้ำหนักต่ำสุด

จะถูกแทนที่ด้วย Gaussian ที่มีค่าเฉลี่ย $I_{s,t}$ ความแปรปรวนเริ่มต้นเพิ่มขึ้น (σ^2_0) และมีน้ำหนักน้อย (ω_0) ทุกๆค่าของ Gaussian ที่หาได้แล้วนั้นน้ำหนักของ K คือ $\omega_{i,s,t}$ นั้นจะถูกทำให้เป็นปกติโดย ดังนั้นเราจึงให้ค่ารวมทั้งหมดเป็น 1 ซึ่งจะทำให้การกระจาย K จึงขึ้นอยู่กับค่า $\omega_{i,s,t} / \sigma_{i,s,t}$ และให้ H เป็นภาพบางส่วนของภาพพื้นหลัง

$$H = \underset{h}{\operatorname{argmin}} \left(\sum_{i=1}^h \omega_i > \tau \right)$$

ที่ τ เป็นค่า Threshold เมื่อนั้นพิกเซล

Kernel Density Estimation (KDE)



นอกจากนี้ยังสามารถใช้วิธีการแบบไม่มีโครงสร้างเพื่อใช้เป็นรูปแบบ PDF หลายรูปแบบ ในมุมมองนี้ Elgammal et al. เสนอ Parzen ประมาณการหน้าต่างที่พิกเซลแต่ละพื้นหลัง

$$P(I_{s,t}) = \frac{1}{N} \sum_{i=t-N}^{t-1} K(I_{s,t} - I_{s,i})$$

พิกเซลมีป้ายกำกับเป็นเบื้องหน้าถ้าไม่น่าจะมาจากการกระจายนี้เช่นเมื่อ P (คือ t) มีค่าน้อยกว่าค่าที่กำหนดไว้ โปรดทราบว่า σ_j สามารถกำหนดค่าได้หรือประมาณไว้ล่วงหน้า ตามวิธี Elgammal et al.

[13] สามารถใช้วิธีการที่ซับซ้อนมากขึ้น เช่น Mittal และ Paragios's [26] ซึ่งอิงกับ "ตัวแปร Bandwidth Kernels"

Codebook (CBRGB)

อีกวิธีหนึ่งที่ใช้กับ multimodal backgrounds คือสิ่งที่เรียกว่า วิธี codebook (โดย Kim et al) ขึ้นอยู่กับค่า training sequence ที่กำหนดให้ วิธีการนี้จะระบุค่าชุดสีของแต่ละพิกเซลของพื้นหลัง (เรียกว่า codewords) ที่เก็บไว้ใน codebook โค้ดเหล่านี้อธิบายถึงสีที่พิกเซลมีแนวโน้มที่จะใช้เวลาในช่วงเวลาหนึ่งๆ ตัวอย่างเช่นพิกเซลในพื้นที่ที่มีเสถียรภาพ (stable area) อาจสรุปได้โดย codeword แต่สำหรับที่ไม่เสถียร ตัวอย่างเช่น ต้นไม้ที่ถูกลมพัดจนพลิ้วไหว ที่นี้อาจสรุปได้ว่ามีค่าโค้ดสีทั้งหมด 3 สี คือสีเขียวจากใบไม้ สีฟ้าจากท้องฟ้าและสีน้ำตาลจากเปลือกไม้ และเงาที่มีความสอดคล้องกับการเปลี่ยนทิศทางของแสงและการเคลื่อนไหวของวัตถุเบื้องหน้า โดยพื้นฐานวิธีการนี้ได้รับการออกแบบมาเพื่อกำจัด false positives ที่เกิดจากการเปลี่ยนแปลงการของแสง การส่องสว่าง นี่ทำโดยการประเมินผลแยกสีและการบิดเบือน ดังนี้

$$\sqrt{I_{s,t}^R + I_{s,t}^G + I_{s,t}^B - \frac{(\mu_{i,s}^R \cdot I_{s,t}^R + \mu_{i,s}^G \cdot I_{s,t}^G + \mu_{i,s}^B \cdot I_{s,t}^B)^2}{\mu_{i,s}^R + \mu_{i,s}^G + \mu_{i,s}^B}} < \tau$$

และ brightness distortion คือ

$$\alpha_{i,s} \leq I_{s,t}^R + I_{s,t}^G + I_{s,t}^B \leq \beta_{i,s}$$

โดยที่ $\mu_{i,s}^R$, $\mu_{i,s}^G$, $\mu_{i,s}^B$, $\alpha_{i,s}$ และ $\beta_{i,s}$ เป็นพารามิเตอร์ของตัวแปรรหัสที่สองของพิกเซลและ τ คือ threshold เมื่อใดก็ตามที่ค่าของพิกเซล s เป็นไปตามสมการดังกล่าว ก็จะสามารถบ่งชี้ได้ว่าพิกเซลมีค่าตรงกับ codeword ที่ i^{th}

อย่างไรก็ตาม เราจะเห็นได้ว่าการการดูความเปลี่ยนแปลงความเข้มของสีนั้นค่อนข้างจำกัดและมีค่าของ false negatives มาก ตัวอย่างเช่น เมื่อตรวจจับจรรยาบรรณท้องถนน พบว่ามีเพียงการเคลื่อนที่ของวัตถุที่มีสีเท่านั้นที่ถูกตรวจจับอย่างสมบูรณ์ ในขณะที่บางส่วนของวิดีโอที่นำมาตรวจจับพบว่ายา

ภาพที่มีสีเทาเข้ม ยานพาหนะที่มีการสะท้อนของเงาและยานพาหนะสีขาวนั้นมีค่า intensity เพิ่มขึ้นมากกว่า ดังที่กล่าวมาจะเห็นได้ว่า codebook original ยังมีข้อเสียอยู่

ในการใช้งาน codeword แต่ละตัวคือ RGB Gaussian distribution โดยขึ้นอยู่กับ Nframe long training sequence และ $C_s = \{c_{1,s}, \dots, c_{L,s}\}$, codebook ที่เชื่อมโยงกับพิกเซล s ที่มาจาก L codewords แต่ละ codeword $c_{i,s}$ จะสามารถหาได้จาก Gaussian ซึ่งมาจากค่าเฉลี่ยของ $\mu_{i,s}$ และ เมทริกซ์ความแปรปรวนร่วม $\Sigma_{i,s}$ (ซึ่งสันนิษฐานว่าเป็นเส้นทแยงมุม) พารามิเตอร์ Gaussian นั้นจะเหมือนกันกับการประมาณจำนวนระหว่าง training phase

ในช่วงระยะของ phase นั้น ค่า code book ของแต่ละพิกเซลจะเริ่มต้นด้วยสี ที่เวลา 0 คือ $\mu_{1,s} = I_{s,0}$ และ $\Sigma_{1,s} = \sigma^2_{0,Id}$ โดยที่ σ^2_0 เป็นค่าคงที่และ I_d เป็นเมทริกซ์เอกลักษณ์ จากนั้นแต่ละสีใหม่ จะถูกเปรียบเทียบกับค่า codewords ที่ประเมินไว้ล่วงหน้า $c_{i,s}$ และสำหรับการจับคู่ของแต่ละครั้ง พารามิเตอร์ของ codebook ที่เกี่ยวข้องจะเป็นไปตามสมการ เมื่อใดก็ตามที่เป็น $I_{s,t}$ นั้นมีค่าไม่ตรงกันกับ ค่าพารามิเตอร์ของ codebook ค่าของ code book ใหม่ $c_{j,s}$ นั้นจะถูกเป็นไปตามนี้

$$\mu_{j,s} = I_{s,t} \text{ และ } \Sigma_{j,s} = \sigma^2_0 I_d$$

ในระหว่างขั้นตอนการตรวจสอบแต่ละพิกเซลจะเป็นไปตามเงื่อนไขนี้

$$\mathcal{X}_t(s) = \begin{cases} 1 & \text{if } d_M(\mathbf{I}_{s,t}, c_{i,s}) > \tau \quad \forall i \\ 0 & \text{otherwise.} \end{cases}$$

โดยที่ τ เป็น threshold และ d_M คือ Mahalanobis distance

Eigen Backgrounds (Eigen)

วิธีการนี้เป็นวิธีการ non pixel-level method เขียนขึ้นโดย Oliver et al โดยใช้ eigenspace เพื่อสร้างพื้นหลัง องค์ประกอบหลักของวิธีการนี้อยู่ในความสามารถในการเรียนรู้รูปแบบพื้นหลังจาก

ลำดับภาพที่ไม่จำกัดของวิดีโอแม้จะมีการเคลื่อนที่ของวัตถุด้านหน้า ในขณะที่วิธีการก่อนหน้านี้จะใช้สถิติระดับพิกเซล Eigen ซึ่งคาดว่าจะทำให้พื้นหลังมีเสถียรภาพมากขึ้น ซึ่งให้ $\{I_i\}_{i=1:N}$ เป็นตัวแทนคอลัมน์ N-frames long training sequence ค่าเฉลี่ย μ สามารถคำนวณได้โดย

$$\mu = \frac{1}{N} \sum_{i=1}^N I_i$$

จากนั้นจึงลบด้วยภาพนำเข้าแต่ละภาพเพื่อสร้าง zero-mean vector $\{X_i\}_{i=1:N}$ โดย $X_i = I_i - \mu$ จากนั้นเมทริกซ์ความแปรปรวนร่วม Σ หาได้ด้วย $\Sigma = E[XX^T]$ โดยมี $X = [X_1, \dots, X_n]$ อ้างอิงจาก การแปลงการแปลง Karhunen-Loeve ซึ่งสามารถคำนวณสมการเวกเตอร์เมทริกซ์ ϕ ด้วย diagonalizes the covariance

$$D = \phi \Sigma \phi^T$$

ที่ D เป็น corresponding diagonal matrix จากการวิเคราะห์ Principal Component Analysis (PCA) เมทริกซ์รูปสี่เหลี่ยมใหม่ ϕ_M ถูกสร้างขึ้นจาก M eigenvectors ที่มีค่า eigenvalues ที่มากที่สุด เมื่อนำ eigen background ϕ_M และค่าเฉลี่ย μ มาคำนวณ คอลัมน์ของภาพแต่ละภาพที่นำเข้า (I_t) จะถูกแสดงบน first projected ไปยัง M -dimensional subspace

$$B_t = \phi_M(I_t - \mu)$$

และสร้างขึ้นดังนี้

$$I'_t = \phi_M^T B_t + \mu.$$

สุดท้ายวัตถุด้านหน้าที่มีการเคลื่อนไหวจะถูกตรวจจับโดยการคำนวณระยะห่างระหว่างภาพนำเข้า (I_t) และภาพที่ถูกสร้างขึ้นใหม่ (I'_t)

$$\mathcal{X}_t(s) = \begin{cases} 1 & \text{if } d_2(I_t, I'_t) > \tau \\ 0 & \text{otherwise,} \end{cases}$$

โดยที่ τ เป็นค่า threshold และ d_2 เป็น Euclidian distance การ Eigen decomposition สามารถคำนวณได้อย่างรวดเร็วด้วย Single Value Decomposition แต่ที่น่าเสียดายที่การหาค่า Φ_M ใน video streams เป็นไปได้ยาก

2.2 Edge Detection [2]

การหาขอบภาพ (Edge Detection) เป็นการหาเส้นรอบวัตถุที่อยู่ในภาพ เมื่อทราบเส้นรอบวัตถุ ก็จะสามารถคำนวณหาพื้นที่ (ขนาด) รูหรือจำนวนของวัตถุนั้นได้ อย่างไรก็ตามการหาขอบภาพที่ถูกต้องสมบูรณ์ไม่ใช่เป็นเรื่องที่ง่าย โดยเฉพาะอย่างยิ่งการหาขอบของภาพที่มีคุณภาพต่ำ ภาพที่มีความแตกต่างระหว่างพื้นหน้า กับ พื้นหลัง น้อย หรือภาพมีความสว่างไม่สม่ำเสมอทั่วภาพ

ขอบภาพเกิดจากความแตกต่างของความเข้มแสงจากจุดหนึ่งเทียบกับอีกจุดหนึ่ง หากค่าความต่างนี้มีค่ามากขอบภาพก็จะเห็นได้อย่างชัดเจน แต่ถ้าความแตกต่างมีค่าน้อย ขอบภาพก็จะไม่ชัดเจน โดยในการหาขอบภาพจะใช้หลักการในการหาความชันของความเข้มสี (intensity) เนื่องจากขอบรูปเป็นบริเวณที่มีความต่างของค่าสีมาก

ถาต้องการหาขอบภาพในแนวนอนอย่างง่าย วิธีการก็คือหาผลต่างระหว่างจุดหนึ่งกับจุดที่อยู่ข้างล่าง (หรือข้างบน) ของจุดนั้น ดังนี้

$$Y_{diff}(x,y) = I(x,y) - I(x,y+1)$$

โดยที่ Y_{diff} คือค่าความแตกต่างในแนวแกนตั้ง และ $I(x,y)$ คือค่าความเข้มแสงของจุดภาพที่ตำแหน่ง (x,y)

ผลของการใช้สมการมีค่าเทียบเท่ากับการ convolution ภาพด้วย template

$$\begin{bmatrix} -1 & 1 \end{bmatrix}$$

การหาขอบภาพในแนวตั้งก็สามารถหาได้เช่นเดียวกันคือ

$$X_{diff}(x,y) = I(x,y) - I(x-,y) \quad (8.2)$$

โดยที่ X_{diff} คือค่าความแตกต่างในแนวนอน และสมการที่ (8.2) มีให้ผลเทียบเท่ากับการ convolution ด้วย template

$$\begin{matrix} -1 & 1 \end{matrix}$$

convolution เขาด้วยกัน เพื่อที่จะได้มีตัววัดความแรงของขอบภาพ (Gradient Magnitude) เพียงตัวเดียว เนื่องจากค่าความแตกต่างอาจมีค่าเป็นบวกหรือลบ ดังนั้น การบวกค่าความแตกต่างของทั้งสองแกนอาจทำให้ขอบภาพเกิดการหักล้างกันเอง ในทางปฏิบัติ เราจะต้องนำค่าสัมบูรณ์ (Absolute Value) หรือค่ากำลังสอง (Squared Value) ของค่าความแตกต่างของทั้งสองแกนมาบวกกันแทน นอกจากหาความแรงของขอบภาพแล้ว การหาทิศทางของขอบภาพ (Gradient Direction) ก็มีประโยชน์เช่นกัน การหาทิศทางของขอบภาพสามสามารถทำได้โดยการใช้สมการดังต่อไปนี้

$$GD(x,y) = \tan^{-1} \left\{ \frac{Y_{diff}(x,y)}{X_{diff}(x,y)} \right\}$$

โดยให้ $GD(x,y)$ ทิศทางของขอบภาพที่ตำแหน่ง (x,y)

วิธีการหาขอบภาพที่กล่าวมามีข้อเสียคือ การหาขอบภาพไม่ได้นำค่าความเข้มแสงของจุดรอบข้างมาใช่ ซึ่งในความเป็นจริงแล้ว ภาพที่ได้มาส่วนใหญ่จะไม่มีขอบภาพชัดเจน ขอบเขตของพื้นที่มักมีการเปลี่ยนแปลงอย่างซ้าๆ วิธีที่กล่าวมาจะไม่สามารถหาขอบภาพได้อย่างแม่นยำกับภาพลักษณะนี้ การใช้ template ที่มีขนาดใหญ่ขึ้นจะสามารถแก้ปัญหานี้ได้

2.2.1 Gradient and Edge Detection



รูปที่ 2.4 การไล่ระดับสีของ Gradient

Gradient ของภาพคือรูปแบบการไล่สีที่มีการไล่ระดับความเข้มของสี ความแตกต่างของ gray level ของ pixels รอบๆ ทั้งในแกน x และแกน y ซึ่ง Gradient ของรูป 2D, $I(x,y)$ สามารถนำมาอธิบายในรูปของ vector ได้ดังนี้

$$\nabla I = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix}$$

ซึ่ง $\nabla I(x, y)$ จะสามารถหาได้จาก

$$G_x = \frac{\partial I(x, y)}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{I(x + \Delta x, y) - I(x, y)}{\Delta x}$$

$$G_y = \frac{\partial I(x, y)}{\partial y} = \lim_{\Delta y \rightarrow 0} \frac{I(x, \Delta y) - I(x, y)}{\Delta y}$$

เนื่องจากรูปภาพเป็นระบบ digital ดังนั้นจึงนับ pixel เป็น discrete เช่นกัน ทำให้สามารถหาความแตกต่างของ gray level ระหว่าง กับ pixel รอบข้างได้ เช่น

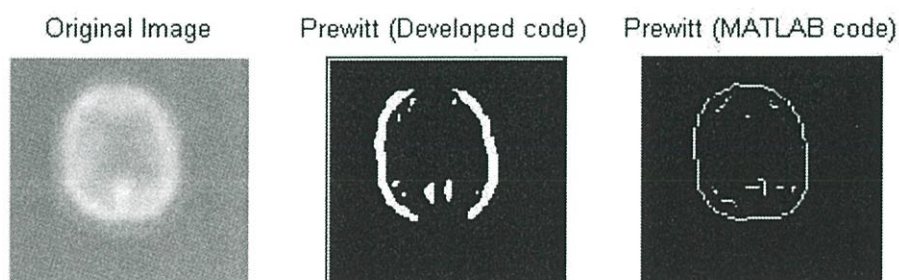
$$\frac{\partial I(x, y)}{\partial x} = \frac{I(x + 1, y) - I(x - 1, y)}{2}$$

$$\frac{\partial I(x,y)}{\partial y} = \frac{I(x,y+1) - I(x,y-1)}{2}$$

PREWITT

$$H_x^P = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad H_y^P = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

การคำนวณจะคล้ายการหา first order derivative โดยจะเทียบ pixel ระหว่างฝั่งซ้ายและฝั่งขวาของจุดที่เราสนใจ โดยให้น้ำหนักก่อนและหลังจุดที่เราสนใจแตกต่างกัน ทำให้เราสามารถหาขอบทั้งในแนวแกน x โดยใช้ H_y และ หาขอบในแนวแกน y โดยใช้ H_x



รูปที่ 2.5 แสดงภาพของวิธีการ Prewitt (File Dicom)

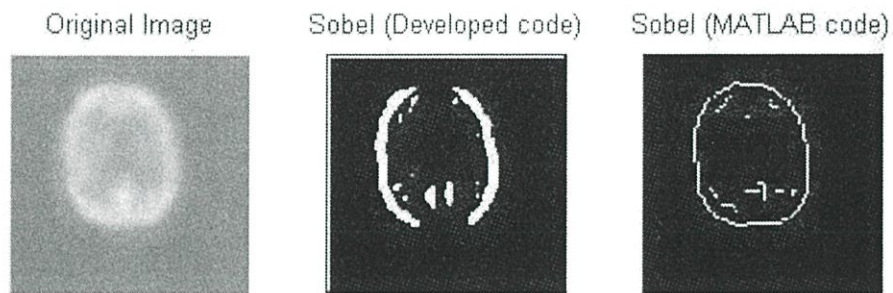
Sobel

$$H_x^S = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad H_y^S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

การคำนวณจะคล้ายกับของ Prewitt operator โดยจะเทียบ pixel ระหว่างฝั่งซ้ายและฝั่งขวา เหมือนกัน แต่น้ำหนักที่ให้จะเน้นไปที่บริเวณ center line ของจุดที่เราสนใจ ทำให้บริเวณที่หาขอบจะเด่นชัดขึ้นมา สามารถหาขอบทั้งในแนวแกน x โดยใช้ H_y^P และ ในแนวแกน y โดยใช้ H_x^P

จากนั้นนำไปใช้หาขอบโดยใช้หลักการของ Gradient operator ซึ่งน้ำหนักที่ใช้ใน Prewitt กับ sobel ก็แตกต่างกันอีกเช่นกัน

$$\nabla I(u, v) \approx \frac{1}{6} \cdot \begin{bmatrix} (I * H_x^P)(u, v) \\ (I * H_y^P)(u, v) \end{bmatrix} \quad \nabla I(u, v) \approx \frac{1}{8} \cdot \begin{bmatrix} (I * H_x^S)(u, v) \\ (I * H_y^S)(u, v) \end{bmatrix}$$

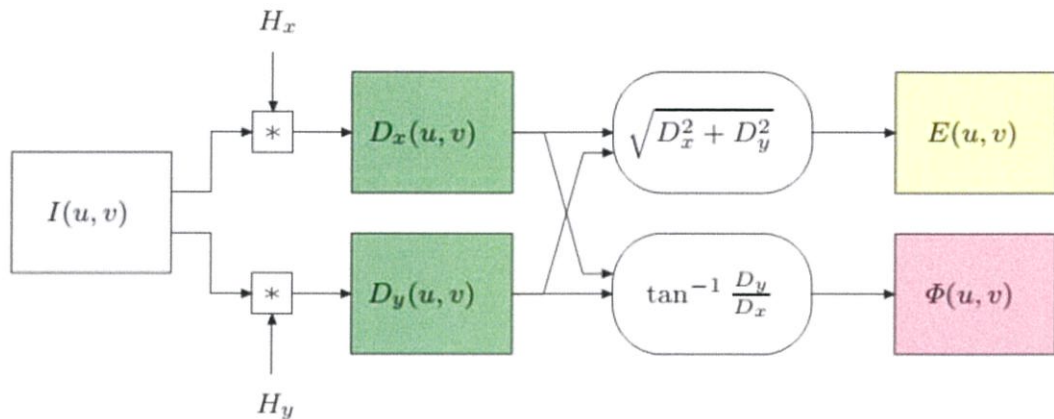


รูปที่ 2.6 แสดงภาพของวิธีการ Sobel (File Dicom)

สรุปชนิดของ mask ที่ใช้

ชนิดของ mask	Row gradient	column gradient
Pixel difference	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Prewitt	$\frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$
Sobel	$\frac{1}{4} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

สรุปขั้นตอนการหาขอบภาพ



2.2.2 Laplacian method

วิธีการนี้เป็นวิธีการที่ใช้การหาอนุพันธ์อันดับที่ 2 เป็นตัวช่วยในการหาความต่างโดยใช้จุดที่ y เป็น ศูนย์ และอาศัยหลักการของลาปลาซ เพื่อหาบริเวณของขอบดังนี้

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

วิธีการนี้จะใช้เวลาในการคำนวณมากกว่าการใช้ Gradient method ตัวอย่างการหาขอบของ Laplacian method เช่น Laplacian of Gaussian และ Marrs-Hildreth เป็นต้น

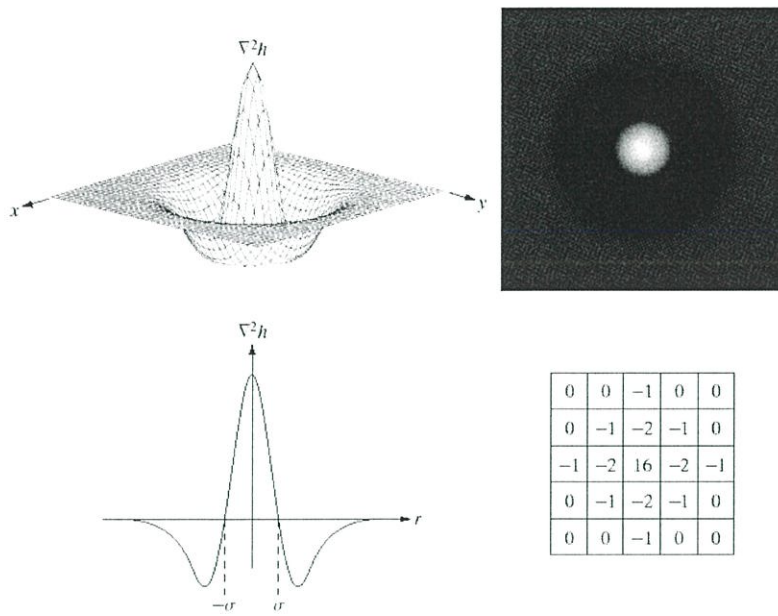
วิธีการนี้ทำได้โดยการสร้าง mask หรือก็คือการสร้างเมทริกซ์จัตุรัส (สมมาตร) ขึ้นเพื่อใช้เป็น filter ในการประมวลผลเพื่อหาขอบของวัตถุ โดยจะขอยกตัวอย่างด้วยเมทริกซ์ชนิด 3×3 ซึ่งจะมีทั้งหมดแปด ทิศทาง

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

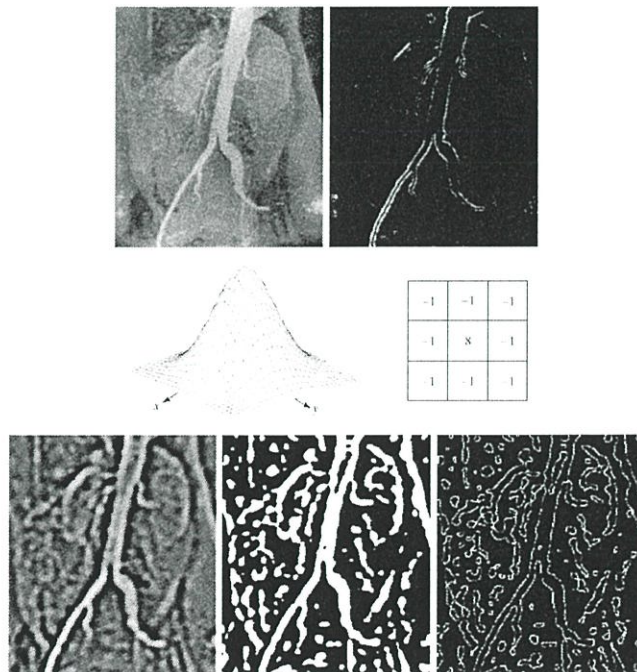
รูปที่ 2.7 แสดง mask matrix ชนิด 3x3

ในการประมวลผลในเทคนิคนี้สามารถทำได้ด้วยการ convolution ชั้นพื้นฐาน และสำหรับ Laplacian mask ที่มีขนาดใหญ่มาก เราจะใช้เทคนิค Laplacian of Gaussian (LOG) เข้ามาช่วยในการสร้าง mask โดยจะใช้ตาม function นี้

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}$$



รูปที่ 2.8 แสดงระบบ LOG



รูปที่ 2.9 ภาพแสดงผลลัพธ์

2.2.3 Canny Edge Detection

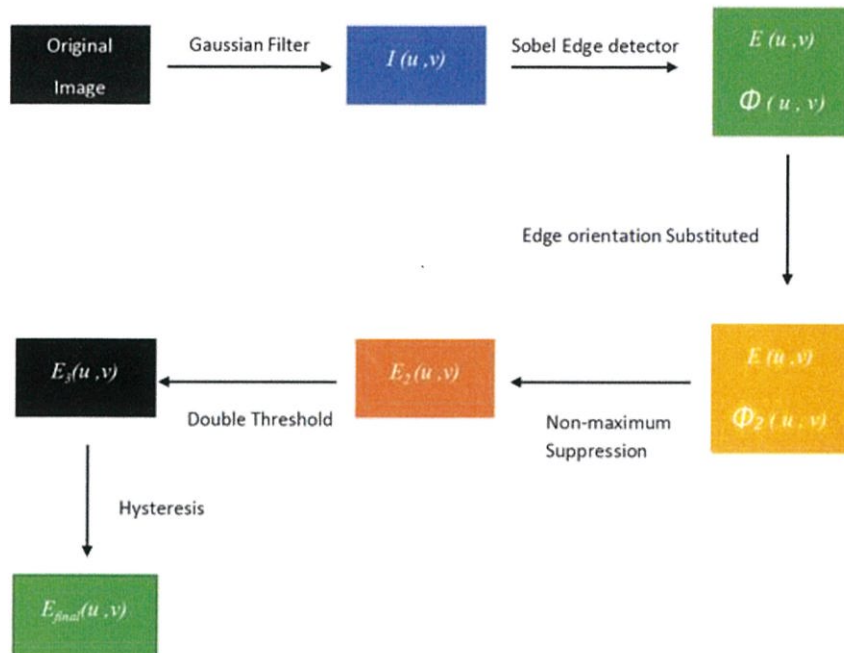
การตรวจจับขอบ Canny คือเทคนิคการดึงข้อมูลโครงสร้างที่มีประโยชน์จากวัตถุวิสัยทัศน์ที่ต่างกัน และลดปริมาณข้อมูลที่จะประมวลผลลงอย่างมาก มีการใช้กันอย่างแพร่หลายในระบบวิสัยทัศน์ของคอมพิวเตอร์หลายเครื่อง Canny พบว่าข้อกำหนดสำหรับการประยุกต์ใช้การตรวจหาขอบบนระบบวิชันทัศน์ที่หลากหลายมีลักษณะคล้ายกันมาก ดังนั้นโซลูชันการตรวจหาขอบเพื่อตอบสนองความต้องการเหล่านี้จึงสามารถใช้งานได้หลากหลายสถานการณ์ เกณฑ์ทั่วไปสำหรับการตรวจจับขอบประกอบด้วย

1. การตรวจจับขอบด้วยอัตราความผิดพลาดต่ำซึ่งหมายความว่า การตรวจจับจะต้องจับขอบได้มากที่สุดเท่าที่จะเป็นไปได้ในภาพ
2. จุดขอบที่ตรวจพบจากโอเปอเรเตอร์ควรอยู่ตรงกลางของขอบ
3. ขอบที่ระบุในภาพควรทำเครื่องหมายเพียงครั้งเดียวและถ้าเป็นไปได้ภาพรบกวนจะไม่สร้างขอบเท็จ

เพื่อตอบสนองความต้องการเหล่านี้ Canny ใช้แคลคูลัสของรูปแบบต่างๆ ซึ่งเป็นเทคนิคที่พบฟังก์ชันที่ปรับการทำงานให้เหมาะสม ฟังก์ชันที่ดีที่สุดในเครื่องตรวจจับ Canny ถูกอธิบายโดยการรวมสี่ค่าแทน แต่สามารถประมาณโดยอนุพันธ์แรกของ Gaussian

ในวิธีการตรวจจับขอบที่ได้รับการพัฒนาขึ้นมาแล้ว ขั้นตอนการตรวจหาขอบ Canny คือหนึ่งในวิธีที่กำหนดโดยเคร่งครัดที่สุดซึ่งให้การตรวจจับที่ดีและเชื่อถือได้ เนื่องจากความสามารถในการตอบสนองกับเกณฑ์การตรวจจับขอบ 3 ข้อและความเรียบง่ายของกระบวนการในการติดตั้งจึงกลายเป็นอัลกอริทึมที่ได้รับความนิยมมากที่สุดสำหรับการตรวจจับขอบ

การทำงานของ Canny edge detection นั้นเริ่มต้นจากการปรับภาพให้เรียบ (Smoothing) ด้วยตัวกรองเกาส์เซียน (Gaussian filter) เพื่อกำจัดสัญญาณรบกวน หลังจากนั้น คำนวณค่าขนาด (magnitude) และทิศทาง (orientation) ของ gradient โดยใช้การหาอนุพันธ์อันดับหนึ่ง ในถัดมาจึงใช้ nonmaxima suppression กับ gradient magnitude เพื่อทำให้ได้ขอบที่บางลง และในขั้นตอนสุดท้ายใช้ double thresholding algorithm เพื่อระบุพิกเซลที่เป็นขอบและช่วยเชื่อมต่อขอบ



กระบวนการสำหรับการหา Canny Edge Detection มีทั้งหมด 5 ขั้นตอน

2.2.3.1 ใช้ Gaussian filter เพื่อขจัดสัญญาณรบกวนออก ซึ่งหาได้จากกรใช้ mask ขนาดเล็กของ Gaussian mask โดยจะต้องใช้ mask ให้เหมาะกับงาน เนื่องจากขนาดของ mask มีผลกับประสิทธิภาพในการกำจัดสัญญาณรบกวนของระบบ



โดย Gaussian filter เป็นไปตาม function นี้

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i - (k + 1))^2 + (j - (k + 1))^2}{2\sigma^2}\right); 1 \leq i, j \leq (2k + 1)$$

ตัวอย่าง 5x5 Gaussian filter โดยให้ค่า $\sigma = 1.14$

$$\mathbf{B} = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * \mathbf{A}.$$

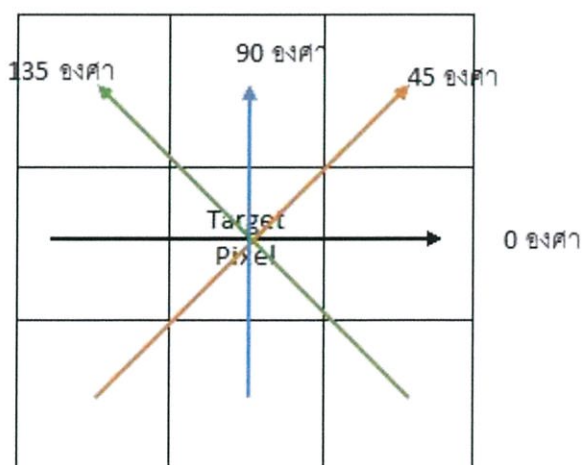
สิ่งสำคัญคือต้องเข้าใจว่าการเลือกขนาดของ Gaussian จะส่งผลต่อประสิทธิภาพของเครื่องตรวจจับ ยังมีขนาดใหญ่เท่าไรความไวของเครื่องตรวจจับจะลดลง นอกจากนี้ข้อผิดพลาดในการแปลเพื่อตรวจหาขอบจะเพิ่มขึ้นเล็กน้อยเมื่อเพิ่มขนาด ขนาด 5×5 มีขนาดที่นิยมใช้กัน แต่อย่างก็ควรคิดพิจารณาในแต่ละงานด้วย

2.2.3.2 หา intensity gradient ของของภาพ

ขอบในภาพอาจชี้ไปในทิศทางต่างๆดังนั้นอัลกอริทึม Canny จะใช้ตัวกรองสี่ตัวเพื่อตรวจจับขอบแนวนอนแนวตั้งและเส้นทแยงมุมในภาพเบลอ ตัวตรวจจับขอบ (เช่น Roberts, Prewitt หรือ Sobel) จะส่งกลับค่าสำหรับอนุพันธ์แรกในทิศทางแนวนอน (G_x) และทิศทางแนวตั้ง (G_y) จากนั้นการไล่ระดับสีและทิศทางของขอบสามารถทำได้ดังนี้

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$$

$$\Theta = \text{atan2}(\mathbf{G}_y, \mathbf{G}_x)$$



2.2.3.3 Non-maximum suppression

Non-maximum suppression จะใช้กับขอบ "บาง" หลังจากใช้การคำนวณการไล่ระดับสีแล้วขอบที่ดึงออกมาจากการไล่ระดับสีจะยังคงเบลอลอยอยู่ สำหรับเกณฑ์ที่ 3 ควรมีการตอบสนองที่ถูกต้องเพียงอย่างเดียว ดังนั้น Non-maximum suppression สามารถช่วยในการยับยั้งค่าการไล่ระดับสีทั้งหมด (โดยการตั้งค่าเป็น 0) ยกเว้นค่าสูงสุดเฉพาะจุด ซึ่งระบุตำแหน่งที่มีการเปลี่ยนความเข้มของค่าที่คมชัดที่สุด อัลกอริทึมสำหรับแต่ละพิกเซลในภาพไล่ระดับสีคือ

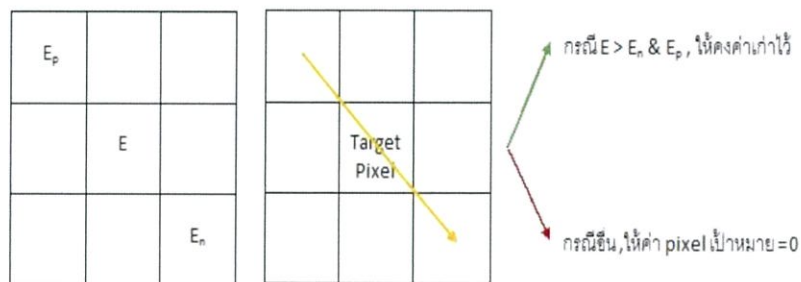
1. เปรียบเทียบความแข็งแรงของขอบของพิกเซลปัจจุบันกับความแรงของขอบพิกเซลในทิศทางการไล่ระดับสีบวกและลบ
2. ถ้าความแรงของขอบพิกเซลปัจจุบันมีค่ามากที่สุดเมื่อเทียบกับพิกเซลอื่น ๆ ในตัวพรางที่มีทิศทางเดียวกัน (เช่นพิกเซลที่ขึ้นไปในทิศทาง y - มันจะเปรียบเทียบกับพิกเซลด้านบนและด้านล่างใน แกนแนวตั้ง) ค่าจะถูกเก็บรักษาไว้ มิฉะนั้นค่าจะถูกระงับ

ในขั้นตอนการใช้งานบางอย่างอัลกอริทึมจะแบ่งประเภทของทิศทางการไล่ระดับสีอย่างต่อเนื่องเป็นชุดทิศทางเล็ก ๆ และย้ายตัวกรอง 3×3 ไปที่เอาต์พุตของขั้นตอนก่อนหน้านี้ (นั่นคือความแรงของขอบและทิศทางการไล่ระดับสี) ในแต่ละพิกเซลจะเป็นการยับยั้งความแรงของขอบของพิกเซลศูนย์กลาง (โดยการตั้งค่าเป็น 0) ถ้าขนาดของมันไม่มากกว่าขนาดของสองแห่งที่ตั้งอยู่ใกล้เคียงในทิศทางการไล่ระดับสี ตัวอย่างเช่น ถ้ามุมของการไล่ระดับสีที่โค้งมนเป็น 0° (คือขอบอยู่ในทิศเหนือ - ใต้) จุดจะถือว่าอยู่ในขอบถ้าความแรงของการไล่ระดับสีมีค่ามากกว่าขนาดที่พิกเซลทางทิศตะวันออกและทิศตะวันตก ถ้ามุมของการไล่ระดับสีที่โค้งมนเป็น 90° (คือขอบอยู่ในทิศทางตะวันออก - ตะวันตก) จุดจะถือว่าอยู่ที่ขอบถ้าความแรงของการไล่ระดับสีมากกว่าขนาดที่พิกเซลในทิศทางเหนือและใต้ ถ้ามุมของการไล่ระดับสีแบบกลมเป็น 135° (ขอบคือทิศตะวันออกเฉียงใต้ไปทางทิศตะวันตกเฉียงใต้) จุดจะถือว่าอยู่ที่ขอบถ้าความลาดเอียงของมันสูงกว่าขนาดที่พิกเซลทางทิศตะวันตกเฉียงเหนือและทิศตะวันออกเฉียงใต้ ถ้ามุมของการไล่ระดับสีเป็นมุม 45 องศา (คือขอบอยู่ทางทิศตะวันตกเฉียงเหนือไปทางทิศตะวันออกเฉียงใต้) จุดนี้จะถือว่าอยู่บนขอบถ้าความลาดเอียงสูงกว่าขนาดที่พิกเซลทางทิศ

ตะวันออกเฉียงเหนือและทิศตะวันตกเฉียงใต้ ในการใช้งานที่แม่นยำยิ่งขึ้นจะมีการใช้การสอดแทรกเชิงเส้นระหว่างพิกเซล

สองแห่งที่ตั้งอยู่ใกล้เคียงกับทิศทางการไล่ระดับสี ตัวอย่างเช่นถ้ามุมลาดอยู่ระหว่าง 45° ถึง 90° การสอดแทรกระหว่างการไล่ระดับสีที่พิกเซลทางทิศเหนือและทิศตะวันออกจะให้ค่า interpolated และ interpolation ระหว่างพิกเซลทางใต้และทางตะวันตกเฉียงใต้จะให้ค่าอื่น ๆ (โดยใช้ข้อตกลงของย่อหน้าสุดท้าย) ขนาดการไล่ระดับสีที่พิกเซลกลางต้องใหญ่กว่าทั้งสองแบบเพื่อให้มีการทำเครื่องหมายเป็นขอบ

โปรดสังเกตว่าสัญลักษณ์ของทิศทางไม่เกี่ยวข้อง เช่น ทิศตะวันตกเฉียงใต้เหมือนกับทิศใต้และอื่น ๆ



2.2.3.4 Double threshold

หลังจากที่เราผ่านกระบวนการทั้งหมดมาแล้วนั้น ภาพที่ได้ยังมีความเป็นไปไม่ได้ที่จะยังหลงเหลือผลกระทบจาก noise ที่ไม่เกี่ยวข้องอยู่จำนวนหนึ่ง เพื่อตัดสิ่งรบกวนเหล่านี้ออกไปจึงมีการนำ Hysteresis เข้ามาช่วยโดยทำให้เราสามารถเลือกช่วงค่า Edge strength ที่ต้องการแสดงได้และค่าที่ต่ำกว่าในขอบเขตที่ระบุให้กำจัดค่าของ pixel นั้นทิ้ง(ให้เท่ากับ 0) ส่วนค่าที่มากกว่าให้เป็นขอบภาพความเข้มสูง(strong edge) และส่วนที่อยู่ในช่วงให้เป็นขอบความเข้มจาง(weak edge)ซึ่งจะนำมาพิจารณาในขั้นตอนต่อไป

2.2.3.5 Hysteresis

จนถึงตอนนี้พิกเซลขอบที่แข็งแกร่งน่าจะมีส่วนเกี่ยวข้องกับภาพขอบสุดท้ายเนื่องจากภาพเหล่านี้ถูกดึงออกจากขอบที่แท้จริงของภาพ อย่างไรก็ตามจะมีการอภิปรายเกี่ยวกับพิกเซลที่

ไม่แข็งแรงเนื่องจากพิกเซลเหล่านี้สามารถสกัดได้จากขอบจริงหรือความต่างของระดับเสียง / สี เพื่อให้ได้ผลลัพธ์ที่แม่นยำขอบที่ไม่แข็งแรงซึ่งเกิดจากเหตุผลหลังควรถูกลบออก โดยปกติขอบพิกเซลที่ไม่แข็งแรงที่เกิดจากขอบที่แท้จริงจะเชื่อมต่อกับพิกเซลขอบที่แข็งแรงในขณะที่ยังคงตอบสนองต่อเสียงไม่ได้เชื่อมต่อ ในการติดตามการเชื่อมต่อขอบการ blob analysis จะถูกนำไปใช้โดยการดูพิกเซลที่อ่อนแอและพิกเซลที่เชื่อมต่อกัน 8 จุด ตรวจจับที่มีพิกเซลขอบที่แข็งแรงที่มีส่วนเกี่ยวข้องใน blob จุดขอบอ่อนที่สามารถระบุได้ว่าเป็นจุดที่ควรได้รับการเก็บรักษาไว้



รูปที่ 2.10 แสดงผลลัพธ์ของ canny

2.3 Color detection [3]

2.3.1 HSV (Hue Saturation Value) [4][5]

HSV เป็นทางเลือกที่ใช้แทนสีแบบ RGB ซึ่งออกแบบโดยนักวิจัยกราฟิกคอมพิวเตอร์ในยุค 70 เพื่อให้สอดคล้องกับวิสัยทัศน์ของมนุษย์มากที่สุดในการรับรู้ถึงคุณลักษณะการสร้างสี

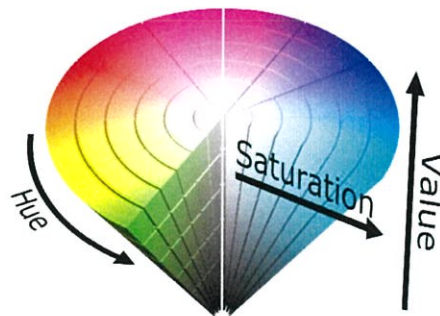
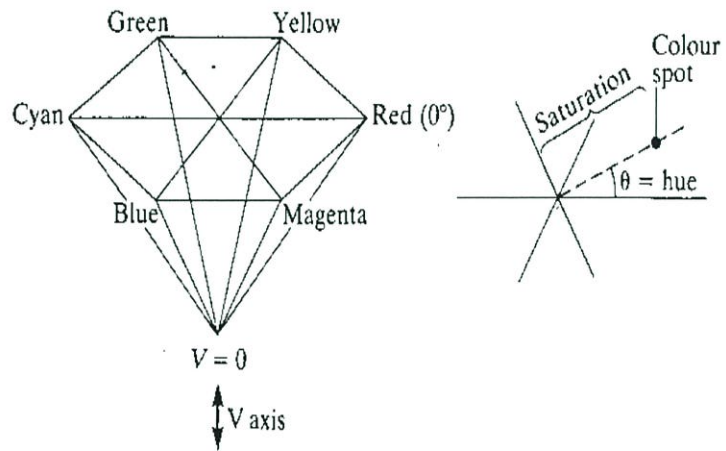
โดยที่ระบบสี HSV (Hue Saturation Value) เป็นการพิจารณาสีโดยใช้ Hue Saturation และ Value ซึ่ง Hue คือค่าสีของสีหลัก(แดง เขียวและน้ำเงิน)ในทางปฏิบัติจะอยู่ระหว่าง 0 และ 255 ซึ่งถ้า Hue มีค่าเท่ากับ 0 จะแทนสีแดงและเมื่อ Hue มีค่าเพิ่มขึ้นเรื่อย ๆ สีก็จะเปลี่ยนแปลงไปตามสเปกตรัมของสีจนถึง 256 จึงจะกลับมาเป็นสีแดงอีกครั้ง ซึ่งสามารถแทนให้อยู่ในรูปขององศาได้ ดังนี้คือ สีแดง = 0 องศา สีเขียวเท่ากับ 120 องศา สีนํ้าเงินเท่ากับ 240 องศา

Hue สามารถคำนวณได้จากระบบสี RGB ได้ดังนี้

$$red_h = red - \min(red, green, blue)$$

$$green_h = green - \min(red, green, blue)$$

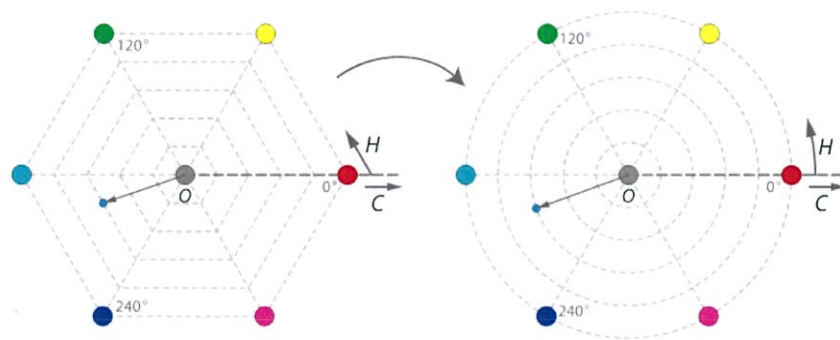
$$blue_h = blue - \min(red, green, blue)$$



รูปที่ 2.11 แสดงระบบสี HSV

จากลักษณะโมเดลของระบบ Hue พบว่าจะมีค่าน้อยหนึ่งค่าที่จะเท่ากับ 0 แต่ถ้ามีสองค่าเท่ากับ 0 แล้ว hue จะเป็นมุมของสี(ค่าสี)มีค่าเป็นไปตามสีที่สามและถ้าทั้งสามสีมีค่าเท่ากับ 0 แล้วจะทำให้ไม่มีค่าของ Hue หรือสีที่ได้จะมีค่าเท่ากับสีขาวนั่นเอง ตัวอย่างเช่น จอภาพขาว-ดำ ถ้าเกิดมีสีแดงสีหนึ่งมีค่าเท่ากับ 0 จะทำให้ค่าสีที่ได้เป็นไปตามสีที่เหลือ การให้นำหนักในการพิจารณาเมื่อสีแดงมีค่าเท่ากับ 0

$$\frac{(240 \times blue_k) + (120 \times green_k)}{blue_k + green_k}$$



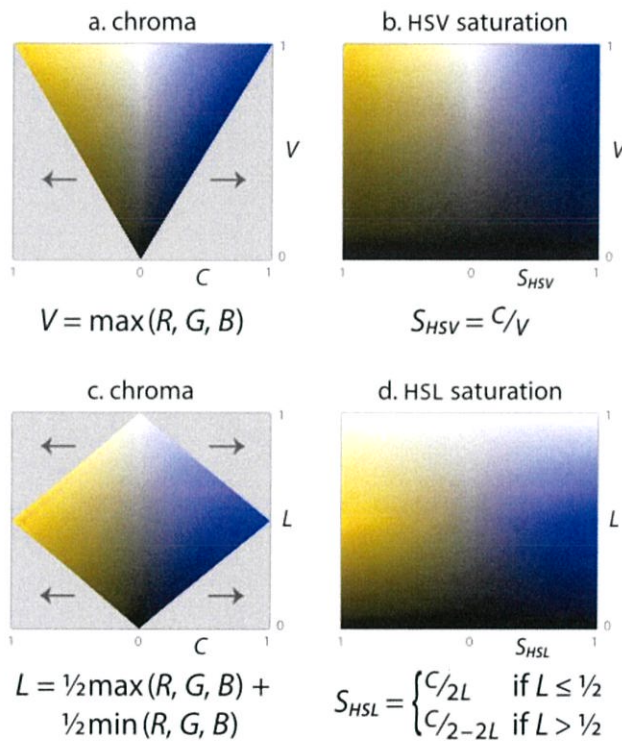
รูปที่ 2.12 ความหมายของ hue ใน HSV มีผลต่อการแปรปรวนของรูปหกเหลี่ยมเป็นวงกลม (fivedots, 2010)

Saturation คือค่าการวัดความบริสุทธิ์ของสี สีที่มีความอิ่มตัวมากจะมีความเข้มของสีมากสีจะสดใส เช่น สีปรากฏในสเปกตรัมจะมีความอิ่มตัวสีสูงสุด สีที่มีความอิ่มตัวน้อยจะมีความเข้มของสีน้อยหรือจางซึ่งถ้า Saturation มีค่าเท่ากับ 0 แล้วสีที่ได้จะไม่มี Hue ซึ่งจะเป็นสีขาวล้วนแต่ถ้า Saturation มีค่าเท่ากับ 255 แสดงว่าจะไม่มีแสงสีขาวผสมอยู่เลย

Saturation สามารถคำนวณได้ดังนี้

$$Saturation = \frac{\max(red, green, blue) - \min(red, green, blue)}{\max(red, green, blue)}$$

$H = 50^\circ / 230^\circ$



รูปที่ 2.13 ใน HSV ความอิ่มตัวของสีเป็นเพียงการปรับค่าความเข้มของสีเพื่อเติมช่วง [0, 1] สำหรับการรวมกันของ hue และ lightness หรือ value (fivedots, 2010)

Value คือค่าการวัดความสว่างหรือความมืดของสี สีที่มีความอิ่มตัวมากจะมีความสว่างน้อยกว่าสีที่มีความอิ่มตัวน้อย ซึ่งสามารถวัดได้โดยค่าความเข้มของความสว่างของแต่ละสีที่ประกอบกันสามารถคำนวณได้จาก

$$value = \max(red, green, blue)$$

Use in image analysis

HSL, HSV, HSI หรือรูปแบบอื่น ๆ ที่เกี่ยวข้องมักใช้ในการมองเห็นและการวิเคราะห์ภาพ คอมพิวเตอร์เพื่อตรวจจับคุณลักษณะหรือการแบ่งส่วนภาพ การประยุกต์ใช้เครื่องมือดังกล่าวรวมถึงการตรวจจับวัตถุเช่นในการมองเห็นของหุ่นยนต์ การจดจำวัตถุเช่นใบหน้าข้อความหรือป้ายทะเบียน การดึงข้อมูลตามเนื้อหา และการวิเคราะห์ภาพทางการแพทย์

ส่วนใหญ่อัลกอริทึมการมองเห็นของคอมพิวเตอร์ที่ใช้กับภาพสีเป็นส่วนขยายที่ซับซ้อนสำหรับอัลกอริทึมที่ออกแบบมาสำหรับภาพสีเทาตัวอย่างเช่น k-means ,fuzzy clustering ของสีพิกเซล หรือ canny edge detection องค์ประกอบสีแต่ละตัวจะถูกแยกจากกันผ่านทางอัลกอริทึมเดียวกัน ดังนั้นคุณลักษณะที่น่าสนใจสามารถแยกแยะได้ในมิติสีที่ใช้ เนื่องจากองค์ประกอบ R, G และ B ของสีของวัตถุในภาพดิจิทัลมีความสัมพันธ์กับปริมาณแสงที่ตกกระทบวัตถุทำให้การอธิบายองค์ประกอบภาพยากลำบาก ดังนั้น hue/lightness/chroma หรือ hue/lightness/saturation มักมีความเกี่ยวข้องมากกว่า

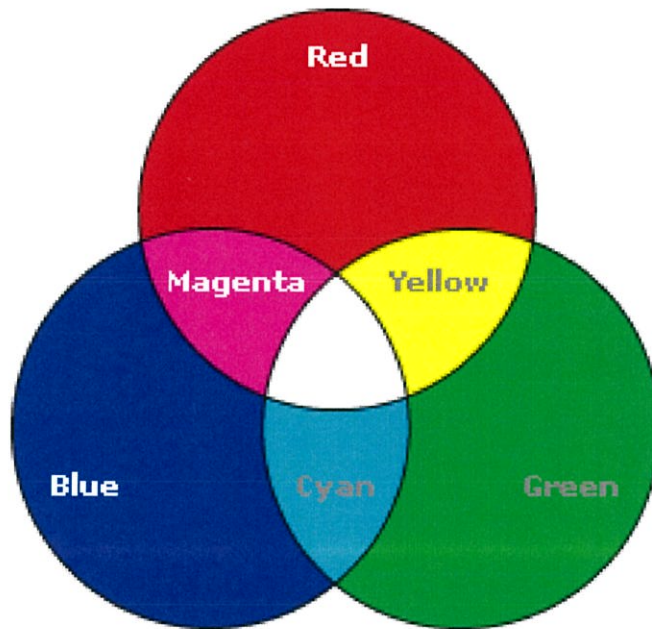
ในปลายปี 1970 การแปลงเช่น HSV หรือ HSI ถูกใช้เพื่อประนีประนอมระหว่างประสิทธิภาพสำหรับการแบ่งส่วนและความซับซ้อนของการคำนวณ เพื่อการประมวลผลประสาทที่ใช้โดยการมองเห็นสีของมนุษย์โดยไม่ได้ตกลงในรายละเอียด นั่นคือถ้าเป้าหมายคือการตรวจจับวัตถุประมาณสีแยกความสว่างและความเข้มหรือความอิ่มตัวของสีมีประสิทธิภาพ แต่ไม่ต้องการที่จะเลียนแบบการตอบสนองต่อสีของมนุษย์อย่างเคร่งครัด วิทยานิพนธ์ต้นแบบของ John Kender ปี 1976 เสนอแบบจำลอง HSI Ohta et al. (1980) แทนใช้แบบจำลองที่มีขนาดใกล้เคียงที่เราเรียกว่า I , α , และ β . ในช่วงไม่กี่ปีที่ผ่านมาโมเดลดังกล่าวยังคงมีการใช้งานกันอย่างแพร่หลายเนื่องจากการเปรียบเทียบประสิทธิภาพกับโมเดลที่ซับซ้อนมากขึ้นและความเรียบง่ายในการคำนวณยังคงเป็นที่น่าสนใจ

2.3.2 RGB (Red Green Blue)

RGB ย่อมาจาก red, green และ blue คือ กระบวนการผสมสีจากแม่สี 3 สี คือสีแดง สีเขียว และสีน้ำเงิน การใช้สัดส่วนของสี 3 สีนี้ต่างกัน จะทำให้เกิดสีต่าง ๆ ได้อีกมากมาย

ระบบสี RGB เป็นระบบสีที่เกิดจากการรวมกันของแสงสีแดง เขียว และน้ำเงินโดยมีการรวมกันแบบ Additive ซึ่งโดยปกติจะนำไปใช้ในจอภาพแบบ CRT (Cathode ray tube) ในการใช้งานระบบสี RGB ยังมีการสร้างมาตรฐานที่แตกต่างกันออกไปที่นิยมใช้งานได้แก่ RGBCIE และ RGBNTSC

Additive colors



รูปที่ 2.14 Additive colors (RGB)

ในการสร้างสีที่มี RGB ต้องผสมลำแสงสามสี คือ สีแดง สีเขียวและสีน้ำเงิน แต่ละลำแสงเรียกว่าส่วนประกอบของสีนั้น ๆ และแต่ละลำแสงจะมี arbitrary intensity ตั้งแต่ปิดอย่างเต็มที่จนเปิดอย่างเต็มที่ในส่วนผสม

รูปแบบสี RGB เป็น additive ในแง่ที่ว่าทั้งสามแสงสีถูกเพิ่มเข้าด้วยกันและสเปกตรัมแสงจะเพิ่มความยาวคลื่นสำหรับความยาวคลื่นเพื่อให้เกิดสเปกตรัมสีสุดท้าย เป็นหลักการที่ตรงกันข้ามกับแบบ subtractive color ที่ใช้กับสีหมึก หมึกสีย้อม และสารอื่น ๆ ที่มีสีขึ้นอยู่กับการสะท้อนแสงที่เราเห็น

เนื่องจากคุณสมบัติของสามสีเหล่านี้ในการสร้างสีขาวจึงมีความแตกต่างอย่างสิ้นเชิงกับสีทางกายภาพเช่น สีอ้อมซึ่งเป็นสีดำเมื่อผสม

ความเข้มข้นสำหรับแต่ละองค์ประกอบให้สีที่มีมากที่สุด (ไม่มีแสงถือว่าเป็นสีดำ) และความเข้มสูงสุดของแต่ละสีจะให้สีขาว คุณภาพของสีขาวนี้ขึ้นอยู่กับลักษณะของแหล่งกำเนิดแสงหลัก แต่ถ้ามีความสมดุลผลลัพธ์ที่เกิดขึ้นคือสีขาวที่เป็นกลางซึ่งตรงกับจุดขาวของระบบ เมื่อความเข้มของชิ้นส่วนทั้งหมดเหมือนกันผลที่ได้คือสีเทาเข้มหรือสว่างขึ้นอยู่กับความเข้มสี เมื่อความเข้มแตกต่างกันผลที่ได้คือ hue ที่อ้อมตัวมากหรือน้อยขึ้นอยู่กับความแตกต่างของความเข้มที่มากที่สุดและความเข้มน้อยที่สุดของสีหลักที่ใช้

เมื่อสีใดสีหนึ่งมีความเข้มสูงสุด สีคือ hue ที่อยู่ใกล้กับสีหลัก (สีแดง สีเขียวหรือสีน้ำเงิน) และเมื่อส่วนประกอบทั้งสองมีความเข้มเท่ากัน hue ที่เป็นสีรองหรือเฉดสีของสีฟ้าอมเขียว(cyan) สีแดงอมม่วง(magenta) และเหลือง(yellow) สีรองจะถูกสร้างขึ้นโดยการรวมสองสีหลักที่มีความเข้มเท่ากันนั่นคือสีฟ้าอมเขียวมาจากสีเขียว + สีน้ำเงิน สีแดงอมม่วงมาจากสีแดง + สีน้ำเงินและสีเหลืองมาจากสีแดง + สีเขียว ทุกสีรองเป็นส่วนประกอบของสีหลัก เมื่อสีหลักและสีรองเสริมจะถูกเพิ่มเข้าด้วยกันผลลัพธ์จะเป็นสีขาว ฟ้าเติมลงในสีแดงและแดงอมม่วงเติมลงในสีเขียวและสีเหลืองเติมลงในสีฟ้า

โมเดลสี RGB เองไม่ได้ระบุว่าอะไรคือสีแดงสีเขียวและสีน้ำเงินดังนั้นผลลัพธ์ของการผสมสีเหล่านี้จึงไม่ได้ระบุเป็นค่าสัมบูรณ์ แต่มีความสัมพันธ์กับสีหลัก เมื่อมีการกำหนดสีที่แน่นอนของสีแดง, สีเขียวและสีฟ้าแล้วแบบสีจะกลายเป็น absolute color space เช่น sRGB หรือ Adobe RGB

2.3.3 Color detection Method [6]

2.3.3.1 Erosion

Erosion คือกระบวนการพื้นฐานของการประมวลผลภาพ โดยจะเป็นกร่อนขนาดบริเวณขอบวัตถุซึ่งที่วิธีคล้ายกับการขยายขนาดภาพ กล่าวคือในกระบวนการจะเริ่มจากการขยายขนาดภาพขึ้น ก่อนจะตรวจสอบความเชื่อมโยงกันระหว่างภาพต้นแบบและภาพที่ทำการขยาย จากนั้นจึงทำการแปลงค่าข้อมูลในระดับพิกเซลโดยให้ข้อมูลส่วนที่ต้องการเป็น 1 และข้อมูลที่ไม่ต้องการเป็น 0 เพื่อให้ได้ภาพที่ต้องการ



รูปที่ 2.15 แสดงผลลัพธ์ของ Erosion

2.3.3.2 Dilation

Dilation คือ กระบวนการขยายภาพซึ่งจะทำงานตรงกันข้ามกับกระบวนการ Erosion ซึ่งการขยายภาพในกระบวนการนี้จะเป็นการขยายภาพแบบไบนารีโดยจะใช้ kernel ในกระบวนการ ถ้าหากบริเวณใดภายใต้ kernel มีค่าเป็น 1 หรือก็คือบริเวณภาพที่ต้องการ บริเวณนั้นจะเพิ่มขนาดขึ้น สำหรับกระบวนการนี้เป็นกระบวนการที่ใช้ร่วมกับ Erosion เพื่อกำจัดจุดขาวในภาพ

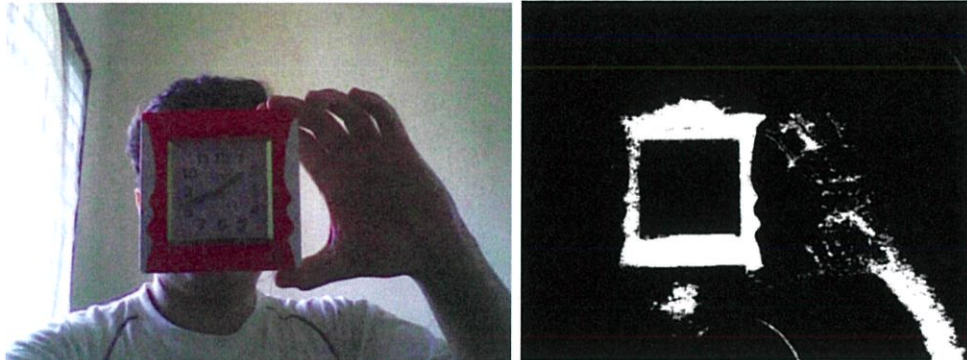


รูปที่ 2.16 แสดงผลลัพธ์ของ Dilation

2.3.3.3 in Range

in Range คือกระบวนการที่ใช้ในการตรวจจับสีของวัตถุในภาพ โดยการกำหนดชนิดของสีและ

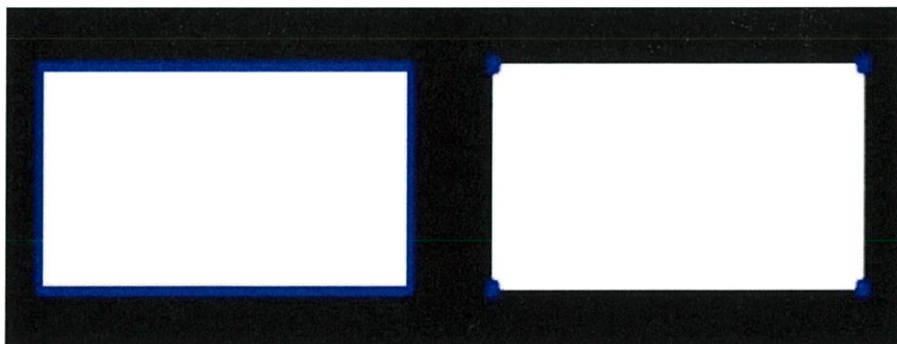
ช่วงค่าสูงและต่ำของสี จากนั้นผลลัพธ์ที่ได้จะเป็นวัตถุที่มีสีในช่วงที่กำหนดเท่านั้น ส่วนอื่นๆ ที่ไม่เกี่ยวข้องหรือไม่อยู่ในช่วงสีจะถูกตัดออกไปจากภาพ กลายเป็นพื้นที่สีดำ



รูปที่ 2.17 แสดงผลลัพธ์ของการใช้ฟังก์ชัน in range โดยในภาพจะเป็นการจับสีแดง

2.3.3.4 Contour Approximation Method

Contour Approximation Method นี้เป็นกระบวนการที่ใช้ในการหารูปทรงต่างๆ ในภาพ โดยใช้หลักการของความต่อเนื่องกันของขอบวัตถุซึ่งจะมีสีหรือความเข้มเดียวกัน ทำให้สามารถหารูปร่างของวัตถุนั้นๆ ได้ โดยในกระบวนการนี้จำเป็นต้องใช้ภาพแบบไบนารีในการดำเนินการ



รูปที่ 2.18 แสดงผลลัพธ์ของ Contour Approximation Method

2.3.3.5 Find center of object

สำหรับการทำงานในกระบวนการนี้คือการหาจุดกึ่งกลางของวัตถุที่ตรวจจับได้ในภาพ เพื่อหาค่า coordinate ของวัตถุเมื่อเคลื่อนที่ไปในตำแหน่งนั้น ๆ

2.4 Python [7]



รูปที่ 2.19 ภาพแสดง Logo ของโปรแกรม Python

Python เป็นภาษาเขียนโปรแกรมระดับสูงที่ใช้กันอย่างกว้างขวางในการเขียนโปรแกรมสำหรับวัตถุประสงค์ทั่วไป ภาษา Python นั้นสร้างโดย Guido van Rossum และถูกเผยแพร่ครั้งแรกในปี 1991 Python นั้นเป็นภาษาแบบ interpreter ที่ถูกออกแบบโดยมีเป้าหมายที่จะทำให้สามารถอ่านได้ง่ายยิ่งขึ้น และโครงสร้างของภาษานั้นจะทำให้โปรแกรมเมอร์สามารถเข้าใจแนวคิดการเขียนโค้ดโดยใช้บรรทัดที่น้อยกว่าภาษาอื่นๆ (เช่น C++ และ Java เป็นต้น)

Python นั้นมีคุณสมบัติเป็นภาษาเขียนโปรแกรมแบบไดนามิกส์และมีระบบการจัดการหน่วยความจำอัตโนมัติและสนับสนุนการเขียนโปรแกรมหลายรูปแบบที่ประกอบไปด้วย การเขียนโปรแกรมเชิงวัตถุ imperative การเขียนโปรแกรมแบบฟังก์ชัน และการเขียนโปรแกรมแบบขั้นตอน มันมี library ที่ครอบคลุมการทำงานอย่างหลากหลาย ตัวแปรในภาษา Python นั้นมีให้ใช้งานได้ในหลายระบบปฏิบัติการ ทำให้โค้ดของภาษา Python สามารถทำงานบนระบบต่างๆ ได้

ภาษา Python นั้นถูกพัฒนาขึ้นมาโดยมีความตั้งใจจะให้มันเป็นภาษาที่อ่านง่าย มันถูกออกแบบมาให้มีโครงสร้างที่มองเห็นได้โดยไม่ซับซ้อน โดยมักจะใช้คำในภาษาอังกฤษในขณะที่ภาษาอื่นใช้เครื่องหมายวรรคตอน นอกจากนี้ Python มีข้อยกเว้นของโครงสร้างทางภาษาน้อยกว่าภาษา C และ Pascal

2.5 Open CV [8]



รูปที่ 2.20 ภาพแสดง Logo ของ opencv

OpenCV (Open Source Computer Vision Library) เป็นโปรแกรมที่เปิดให้ใช้กันอย่างแพร่หลาย โดยไม่มีค่าใช้จ่ายในการใช้งาน รวมทั้งยังมีเครื่องมือที่ใช้สำหรับการเรียนรู้อีกด้วย OpenCV ถูกสร้างขึ้นเพื่อสร้างเป็น library พื้นฐานทั่วไปสำหรับการสร้างโปรแกรมการประมวลผลบนคอมพิวเตอร์ และเพื่อลดเวลาในการทำงานอีกด้วย ซึ่ง OpenCV นี้เป็นผลิตภัณฑ์ที่ได้รับอนุญาตจาก BSD OpenCV ทำให้สามารถนำไปประยุกต์กับงานต่างๆ ได้อย่างแพร่หลาย

Library ของ OpenCV นี้มีอัลกอริทึมมากกว่า 2500 ชนิด ซึ่งรวมถึงชุดวิสัยทัศน์คอมพิวเตอร์ เครื่องคอมพิวเตอร์และขั้นตอนการเรียนรู้ที่ทันสมัยทั้งแบบพื้นฐานและแบบประยุกต์ อัลกอริทึมเหล่านี้สามารถใช้ในการตรวจจับและจำแนกใบหน้า ระบุชนิดวัตถุ หรือแบ่งประเภทการกระทำของมนุษย์ในวิดีโอจากกล้องติดตามวัตถุที่เคลื่อนที่ตึงโมเดล 3 มิติของวัตถุสร้างจุด 3D จากกล้องสเตอริโอและสร้างภาพที่มีความละเอียดสูงได้ OpenCV มีผู้ใช้มากกว่า 47,000 คน ชุมชนและจำนวนการดาวน์โหลดโดยประมาณกว่า 14 ล้านคน library นี้ถูกใช้อย่างกว้างขวางใน บริษัท กลุ่มวิจัยและหน่วยงานของรัฐด้วย

นอกเหนือจากองค์กรที่มีชื่อเสียงระดับโลกเช่น Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota ที่ใช้ library นี้แล้วยังมีหลายบริษัทอีกหลายที่ที่ใช้ OpenCV อีก เช่น Applied Minds, VideoSurf และ Zeitera การใช้งานของ OpenCV ใช้ช่วงจากการรวมภาพ StreetView ไปด้วยกัน การตรวจจับการบุกรุกในวิดีโอการเฝ้าระวังในอิสราเอล การตรวจสอบอุปกรณ์เหมืองแร่ในประเทศจีนช่วยให้หุ่นยนต์นำทางและรับวัตถุที่ Willow Garage การตรวจหาอุบัติเหตุการจมน้ำในสระ

ว่ายนํ้าในยุโรป สเปนและนิวยอร์ก ตรวจสอบรันเวย์สำหรับเศษซากในตุรกีตรวจสอบฉลากสินค้าใน
โรงงานทั่วโลกเพื่อตรวจหาใบหน้าอย่างรวดเร็วในประเทศญี่ปุ่น

OpenCV นั้นสามารถใช้งานได้หลายภาษา C + +, Python, Java และ MATLAB นอกจากนี้
ยังรองรับระบบ Windows, Linux, Android และ Mac OS OpenCV

ขณะนี้มีการพัฒนา CUDA และ OpenCL ที่มีคุณลักษณะเต็มรูปแบบ มีขั้นตอนวิธีมากกว่า 500
และประมาณ 10 เท่าของฟังก์ชันที่เขียนหรือสนับสนุนอัลกอริทึมเหล่านั้น OpenCV เขียนโดยกำเนิดใน
C + + และมีอินเตอร์เฟซเพิ่มเติมที่สามารถทำงานร่วมกับคอนเทนเนอร์ STL ได้อย่างสมบูรณ์แบบ

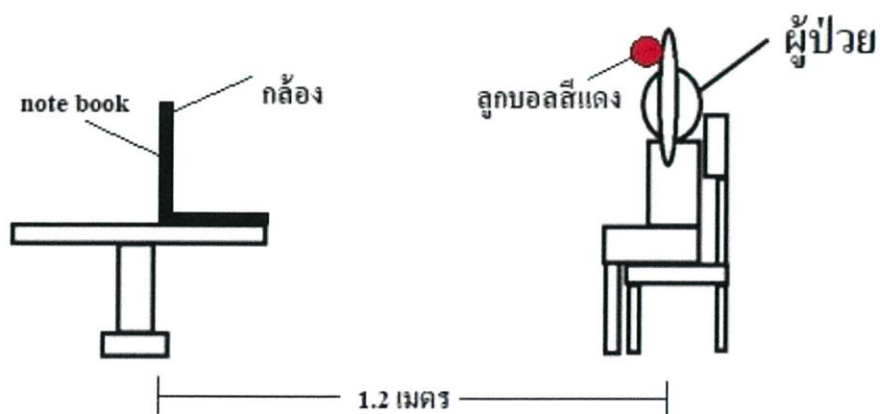
บทที่ 3

การออกแบบระบบ

เนื้อหาส่วนนี้ประกอบไปด้วยหลักการและวิธีการที่ใช้ในงานวิจัย รวมทั้งข้อมูลต่างๆ เกี่ยวกับโปรแกรม กระบวนการวิเคราะห์ข้อมูล การแสดงผลทางด้าน Motion detection

3.1 แนวคิดของระบบที่นำเสนอ

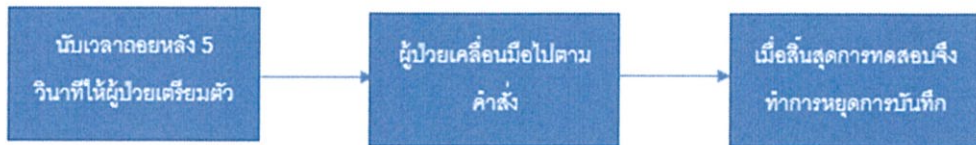
งานวิจัยนี้เป็นงานวิจัยที่จัดทำขึ้นเพื่อการเก็บบันทึกผลและวิเคราะห์เส้นทางการเคลื่อนที่ของมือ โดยใช้กล้องในการเก็บตัวอย่างการเคลื่อนที่ของมือจากผู้ป่วยในรูปแบบของวิดีโอและนำมาประมวลผลในภายหลัง เนื่องจากมีความเสถียรภาพสูงกว่าการแสดงผลแบบ real time และตัวอย่างการเคลื่อนที่ของมือนี้มีระยะเวลาที่สั้น ทำให้ง่ายต่อการเก็บผลมาวิเคราะห์แบบออฟไลน์ ในส่วนของการวิเคราะห์นั้น งานวิจัยนี้เลือกใช้สีในการอ้างอิงตำแหน่ง (หา coordinate) ของมือในหน้าจอแสดงผล เนื่องจากการใช้สีอ้างอิงตำแหน่งมีความเสถียรมากกว่าการอ้างอิงตำแหน่งด้วยมือเพียงอย่างเดียว ดังนั้นแล้วในขั้นตอนของการเก็บผลผู้ป่วยจะต้องถือลูกบอลสีแดงเอาไว้ในมือด้วย ก่อนการเก็บผลจะให้ผู้ป่วยนั่งห่างจากกล้องเป็นระยะประมาณ 1.2 เมตร (เนื่องจากระยะ 1.2 เมตรเป็นระยะที่เมื่อผู้ป่วยยืดแขนจนสุดแล้วภาพที่ได้จะไม่หลุดออกจากหน้าจอแสดงผล) ดังภาพ จากนั้นจึงเริ่มเก็บผลต่อไป โดยโปรแกรมแสดงเส้นทางการเคลื่อนที่ของมือนี้แบ่งออกเป็น 2 ส่วนใหญ่ๆ คือ ส่วนของการเก็บผลและส่วนของการวิเคราะห์และแสดงผลเส้นทางการเคลื่อนที่ของมือ



รูปที่ 3.1 ภาพแสดงการนั่งของผู้ป่วยก่อนเริ่มการเก็บผล

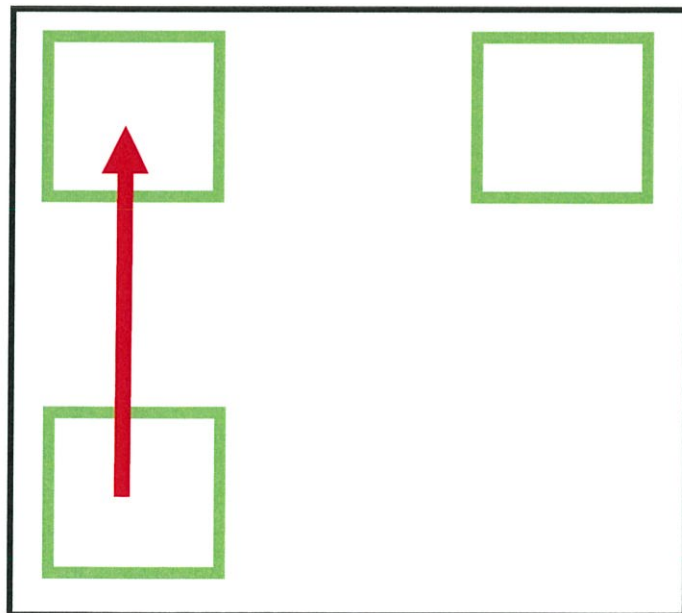
3.2 ส่วนเก็บบันทึกผล

ในส่วนแรกนี้คือส่วนของการบันทึกผลการเคลื่อนที่ของมือหรือแขนของผู้ป่วย เนื่องจากโปรแกรมเป็นการประมวลผลแบบออฟไลน์ ดังนั้นจึงต้องทำการอัดวิดีโอก่อนนำไปประมวลผลเพื่อหาเส้นทางการเคลื่อนที่ของมือ โดยมีลำดับการทำงานของโปรแกรมดังรูปที่ 3.2



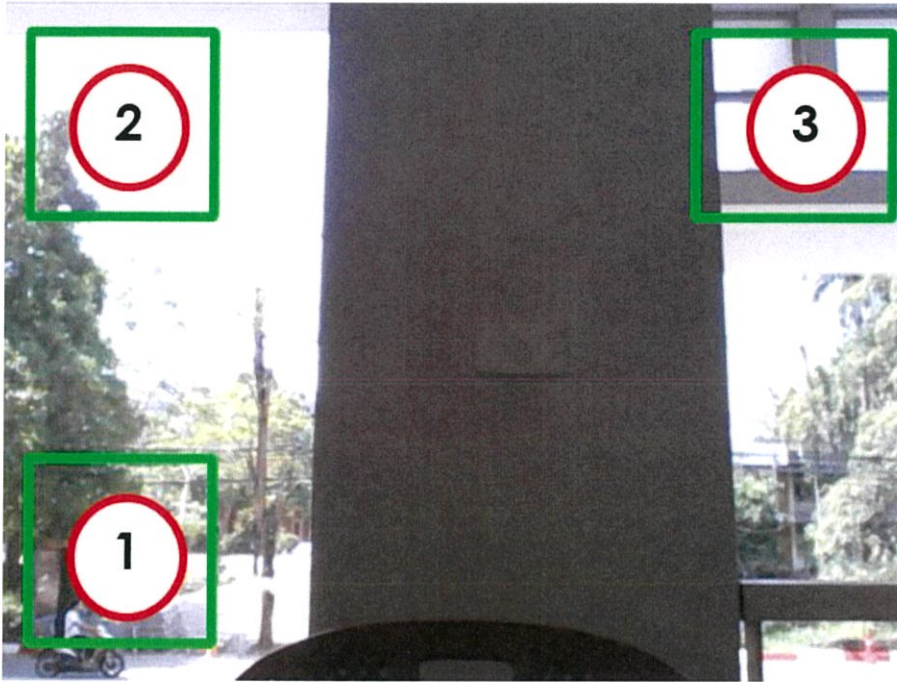
รูปที่ 3.2 ภาพการทำงานของโปรแกรมในส่วนเก็บบันทึกผล

การทำงานจะเริ่มจากการให้ผู้ผู้ป่วยนั่งหลังพิงพนักพิง โดยให้ตำแหน่งที่นั่งของผู้ป่วยเยื้องไปทางด้านตรงข้ามกับมือที่มีปัญหาเล็กน้อยเพื่อให้สามารถเก็บเส้นทางการเคลื่อนที่ของมือได้อย่างชัดเจนและครบถ้วน (วงของการเคลื่อนที่ของมือพอดีกับหน้าจอ ไม่หลุดออกไปจากหน้าจอแสดงผล) และให้มือข้างที่ต้องการทดสอบถือลูกบอลสีแดงเอาไว้ จากนั้นให้ผู้ผู้ป่วยวางลูกบอลให้อยู่ในตำแหน่งเริ่มต้นแล้วรอสัญญาณนับถอยหลังเป็นเวลา 5 วินาทีก่อนเริ่มเคลื่อนที่ไปยังตำแหน่งที่กำหนด ดังภาพ 3.4



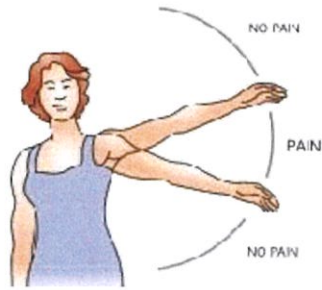
รูปที่ 3.3 ภาพแสดงหน้าจอในขั้นตอนการเก็บผล

ในการทดสอบนั้นจะให้ผู้ป่วยเคลื่อนมือจากกล่องสีเขียวหมายเลข 1 ไปยังกล่องสีเขียวหมายเลข 2 ที่อยู่ด้านบน เพื่อดูลักษณะการเคลื่อนที่ของมือของผู้ป่วยว่ามีลักษณะอย่างไร ในบางกรณีอาจให้เคลื่อนมือจากกล่องสีเขียวหมายเลข 1 ไปยังกล่องสีเขียวหมายเลข 3 ก็ได้ซึ่งทั้งหมดนี้ขึ้นอยู่กับดุลพินิจของนักกายภาพทั้งสิ้น



รูปที่ 3.4 ภาพแสดงหน้าจอเก็บผลจากโปรแกรมจริง

การที่เลือกวางกล่องสีเขียวในตำแหน่งดังกล่าวนี้เนื่องจากการนำลักษณะของความผิดปกติของผู้ป่วยมาร่วมพิจารณาด้วย กล่าวคือสำหรับผู้ป่วยที่มีปัญหาด้าน upper extremity ส่วนมากจะไม่สามารถโยกแขนออกนอกลำตัวหรือยกแขนขึ้นเหนือหัวได้ จากจุดนี้จึงได้นำมาเป็นแบบทดสอบเพื่อทำการบันทึกผลในขั้นตอนดังกล่าวนี้

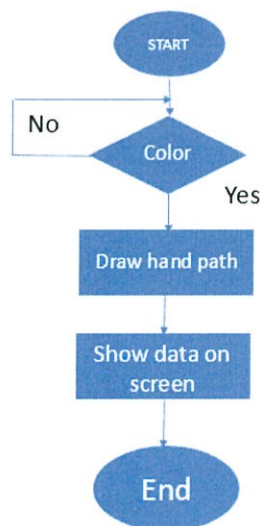


(fivedots, 2010)

รูปที่ 3.5 แสดงการเคลื่อนที่ของแขนและความเจ็บปวดของผู้ป่วย

3.3 ส่วนวิเคราะห์ผล

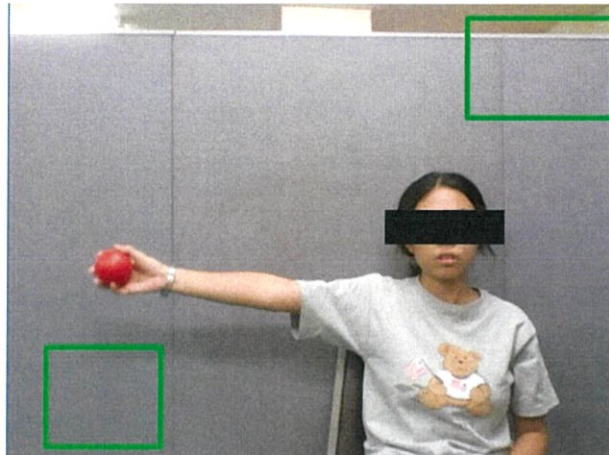
ในส่วนของการวิเคราะห์ผลนั้นจะนำวิดีโอจากส่วนแรกมาประมวลผล โดยที่ตรวจโปรแกรมจะตรวจจับสีแดงที่อยู่ภายในวิดีโอ ก่อนที่จะใช้คำสั่งสร้างจุดตามเส้นทางการเคลื่อนที่ของลูกบอลสีแดงดังกล่าว สำหรับการวิเคราะห์ผลนั้นเหตุที่เลือกใช้จุดแทนเส้นลากยาวก็เนื่องมาจากการวิเคราะห์จุดที่เป็นปัญหาของผู้ป่วย กล่าวคือหากผู้ป่วยเคลื่อนที่เข้าจุดที่เกิดขึ้นก็จะเรียงตัวกันอย่างหนาแน่นต่างจากจุดที่สามารถเคลื่อนที่ได้ได้อย่างเป็นปกติซึ่งลักษณะจุดที่ปรากฏจะเป็นจุดเรียงเดียวเป็นระเบียบแถวยาว จากจุดเหล่านี้ทำให้นักกายภาพสามารถวิเคราะห์ได้ว่าจุดใดคือจุดที่ผู้ป่วยควรเน้นหนักหรือทำกายภาพเพิ่มเติมที่จุดใด รวมถึงสามารถดูพัฒนาการของผู้ป่วยหลังการทำกายภาพบำบัดได้อีกด้วย ขั้นตอนการทำงานของโปรแกรมในขั้นตอนนี้ถูกแสดงในภาพที่ 3.4



รูปที่ 3.6 ภาพแสดงการทำงานของโปรแกรมในส่วนของการประมวลผล

ขั้นตอนการทำงานของโปรแกรมนั้นเริ่มจากการเรียกวิดีโอที่ต้องการที่จะวิเคราะห์ (วิดีโอจากขั้นตอนการเก็บผล) เข้ามาในส่วนของกรวิเคราะห์ผล โดยตัวโปรแกรมจะเริ่มทำการวิเคราะห์ผลโดยใช้ อัลกอริทึมดังต่อไปนี้

- 1) Background subtraction ขั้นตอนนี้จะขั้นตอนในการตัดภาพพื้นหลังออกจากผู้ป่วย จะได้ภาพผู้ป่วยในพื้นที่สีดำ ทำให้ง่ายต่อการตรวจหาสีแดง ผลลัพธ์แสดงเป็นภาพแบบไบนารีดังภาพ 3.9

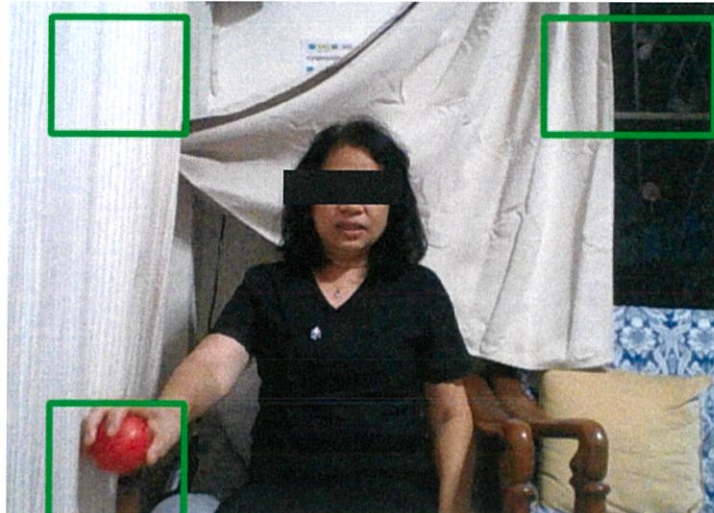


รูปที่ 3.7 ภาพปกติก่อนใช้ Background subtraction

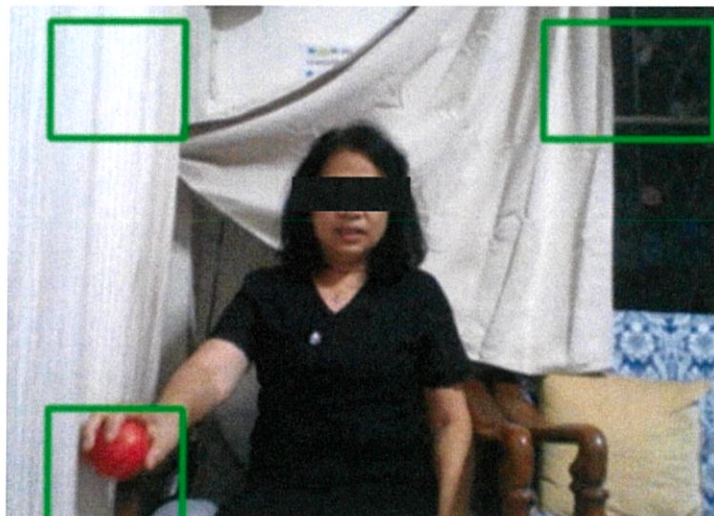


รูปที่ 3.8 ภาพแสดงผลลัพธ์ของ Background subtraction

- 2) Gaussian Filter ในขั้นตอนนี้จะเป็นการใช้อัลกอริทึมที่เป็นฟิลเตอร์ในการกรองภาพ โดยในงานวิจัยชิ้นนี้เลือก Gaussian Filter ประเภท kernel เข้ามาใช้งาน ผลลัพธ์ที่ได้จะทำให้ภาพเบลอขึ้นดังภาพ รูปที่ 3.11



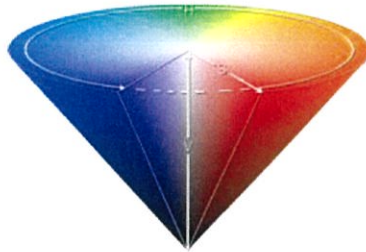
รูปที่ 3.9 ภาพปกติก่อนใช้ Gaussian Filter



รูปที่ 3.10 ภาพแสดงผลของ Gaussian Filter

- 3) In Range คืออัลกอริทึมที่ใช้ในการเลือกช่วงสีที่ต้องการตรวจจับ โดยในงานวิจัยชิ้นนี้เลือกใช้รหัสสีประเภท HSV เนื่องจากมีความละเอียดและมีเสถียรภาพมากกว่า

รหัสสีแบบ RGB ซึ่งสีที่ใช้จะอยู่ในช่วงของสีแดง เนื่องจากลูกบอลที่ใช้เป็นลูกบอลสีแดง



รูปที่ 3.11 ภาพแสดงปริซึมของรหัสสีแบบ HSV (colorizer, n.d.)s

- 4) Erosion สำหรับขั้นตอนนี้จะเป็นขั้นตอนในการลดทอนขอบภาพ เพื่อกำจัด noise หรือสัญญาณรบกวนออกไปจากภาพ เพื่อเพิ่มเสถียรภาพในการประมวลผลให้มากยิ่งขึ้น ในขั้นตอนนี้ภาพจะมีลักษณะผอมบางลงดังภาพที่ 3.14



รูปที่ 3.12 ภาพขาวดำเทาปกติก่อนใช้ Erosion



รูปที่ 3.13 ภาพแสดงผลลัพธ์ของ Erosion

- 5) Dilation คือกระบวนการในการขยายภาพ เป็นขั้นตอนต่อเนื่องจากขั้นตอนที่แล้ว Erosion กล่าวคือเมื่อเราทำการลดขนาดภาพเพื่อกำจัดสัญญาณรบกวนแล้วจึงใช้ Dilation ในการขยายภาพได้สัดส่วนเดิมอีกครั้ง ดังภาพที่ 3.16

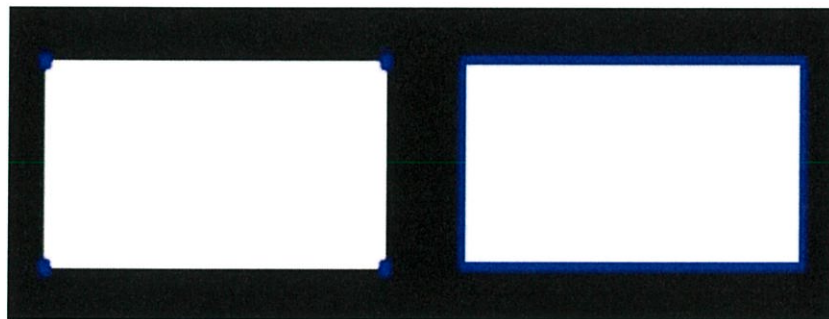


รูปที่ 3.14 ภาพขาวดำเทาปกติก่อนใช้ Dilation



รูปที่ 3.15 ภาพแสดงผลลัพธ์ของ Dilation

- 6) Contour Approximation Method คือกระบวนการในการลากเส้นล้อมรอบรูปร่าง โดยหามุมของภาพที่ต้องการลากเส้นรอบรอบ ก่อนจะกำหนดจุดที่ตำแหน่งนั้นๆ แล้วจึงลากเส้น ดังตัวอย่างในภาพที่ 3.17

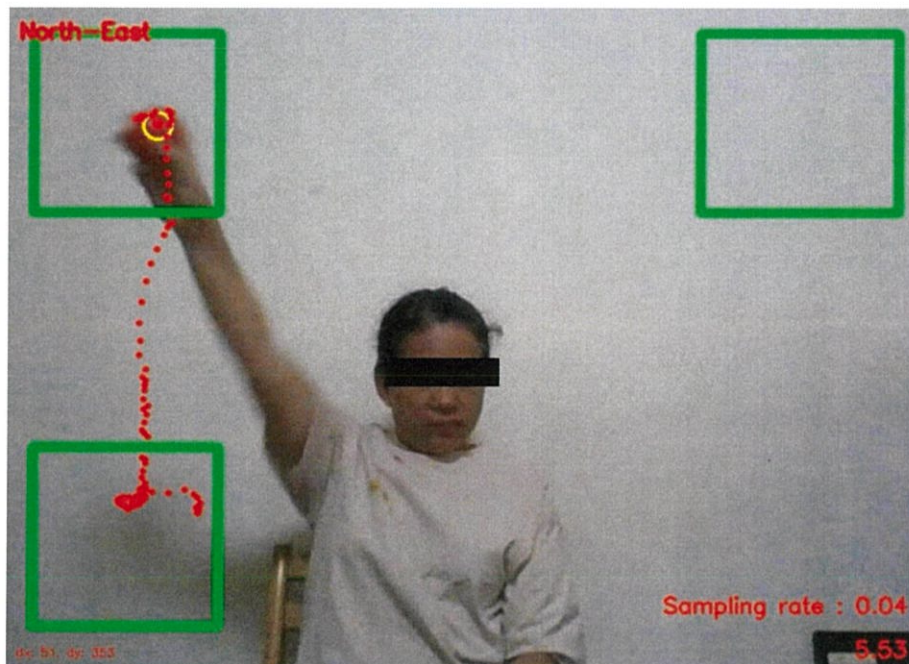


รูปที่ 3.16 ภาพแสดงผลลัพธ์ของ Dilation

- 7) Find center of red ball คือกระบวนการที่ใช้ในการหาจุดกึ่งกลางของลูกบอล เพื่อหาดำแหน่ง X-Y ที่ใช้อ้างอิงในการวาดเส้น hand path
- 8) Draw dots เป็นคำสั่งที่ใช้ในการวาดจุดเพื่อสร้างเป็นเส้นแสดงการเคลื่อนที่ของมือของผู้ป่วย การที่เลือกใช้เส้นในการแสดงผลนั้นเป็นผลเนื่องมาจากการเปรียบเทียบ

เพื่อให้เห็นผลว่าผู้ป่วยมีพัฒนาการในการทำกายภาพบำบัดอย่างไรและใช้ในการวินิจฉัยอาการเบื้องต้นอีกด้วย โดยถ้าผู้ป่วยมีปัญหาในการยกแขนขึ้นลงจุดที่ได้ ออกมาจะมีลักษณะเรียงชิดติดกัน ซึ่งต่างจากคนปกติที่จุดจะเรียงตัวสวยงามและห่างกันอย่างสม่ำเสมอ

นอกจากนี้ยังมีการแสดงเวลาการเคลื่อนที่ของมือเอาไว้ที่บริเวณมุมขวาล่างของจอ เพื่อที่จะใช้เปรียบเทียบพัฒนาการได้เช่นเดียวกันกับความหนาแน่นและการเรียงตัวของจุดตามการเคลื่อนที่ของมือ ซึ่งหากผู้ป่วยใช้เวลาสั้นลงก็จะแปลได้ว่าผู้ป่วยมีพัฒนาการด้านการเคลื่อนที่ในเชิงบวก ส่วนบริเวณด้านซ้ายบนจะแสดงทิศทางการเคลื่อนที่ของลูกบอลสีแดงหรือมือของผู้ป่วย ทำให้นักกายภาพเห็นได้ว่าผู้ป่วยสามารถเคลื่อนมือตรงได้มากน้อยเพียงใด



รูปที่ 3.17 ภาพแสดงหน้าจอแสดงผลในส่วนของการประมวลผลข้อมูล

บทที่ 4

การทดสอบและผลการทดสอบ

สำหรับงานวิจัยชิ้นนี้ ในการทดสอบนั้นได้ทดสอบในผู้ป่วยทั้งหมด 2 กรณี คือในผู้ป่วยที่เป็นโรคไหล่ติด (frozen shoulder) และผู้ป่วยที่เป็นโรคเส้นเลือดในสมองตีบ (Brain Infarction) โดยใช้การทดสอบที่แตกต่างกันออกไปดังหัวข้อต่อไปนี้

4.1 ทดสอบในกรณีของผู้ป่วยโรคไหล่ติด (frozen shoulder)

4.1.1 ข้อมูลเบื้องต้นของผู้ป่วยที่ทำการทดสอบ

สำหรับผู้ป่วยที่มาทำการทดสอบในกรณีนี้เป็นผู้ป่วยเพศหญิง อายุ 52 ปีมีปัญหาที่บริเวณไหล่ซ้าย ป่วยเป็นโรคไหล่ติดเรื้อรังเป็นระยะเวลากว่า 3 ปี ทำให้ยกแขนได้ไม่เต็มวงเหมือนคนปกติ โดยตารางที่ 4.1 แสดงการยกแขนข้างที่ปกติเทียบกับข้างที่ไม่ปกติ

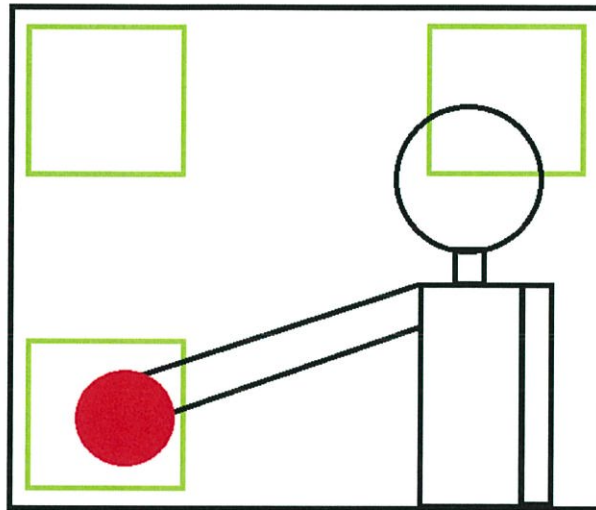


ตารางที่ 4. 1 แสดงการยกแขนของคนปกติเทียบกับการยกแขนของผู้ป่วย

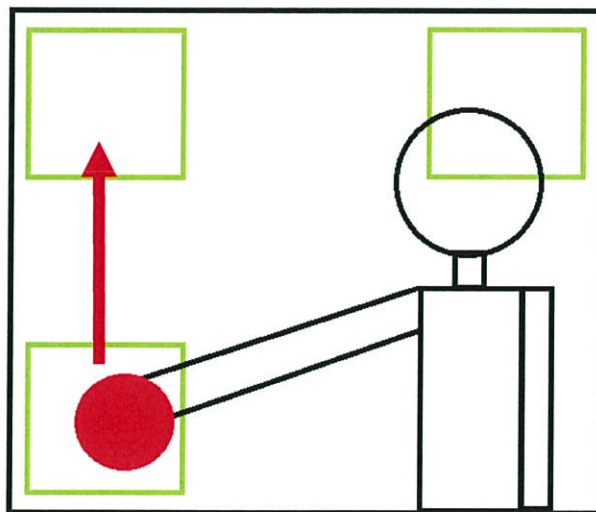
จากตารางการเปรียบเทียบจะเห็นว่าผู้ป่วยมีปัญหาในการยกแขนข้างซ้ายขึ้นเป็นอย่างมาก แต่การมองเปรียบเทียบกันเพียงรูปถ่ายเช่นนี้ยังไม่สามารถวินิจฉัยได้อย่างละเอียดว่าผู้ป่วยมีปัญหาที่จุดใดบ้างจึงได้นำไปสู่การทดสอบในลำดับถัดไป

4.1.2 วิธีทำการทดสอบและผลเมื่อเทียบกับคนปกติ

เพื่อให้ได้ภาพผู้ป่วยแบบเต็มๆ นั้น ในการทดสอบแรกจะให้ผู้ป่วยนั่งในตำแหน่งที่ห่างจากกล้องเป็นระยะ 1.2 เมตร โดยนั่งเอียงเอียงไปทางด้านตรงข้ามกับแขนที่ต้องการทำการทดสอบ เพื่อแขนอยู่ตรงกับกรอบสีเขียวที่เป็นจุดเริ่มต้นที่อยู่ด้านล่างของจอภาพ ดังภาพที่ 4.1 และ 4.2

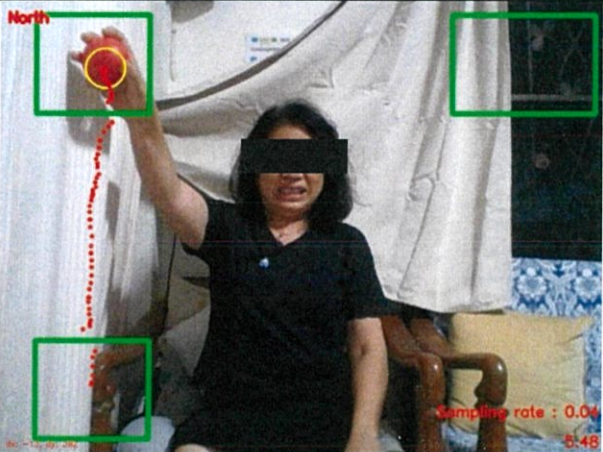
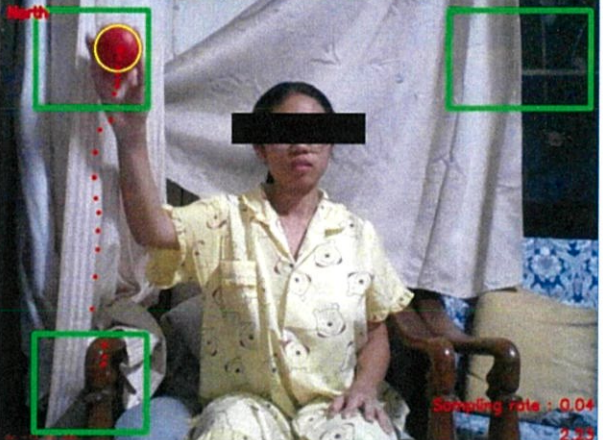


รูปที่ 4.1 ภาพแสดงการนั่งของผู้เข้ารับการทดสอบจากหน้าจอแสดงผล



รูปที่ 4.2 ภาพแสดงการเคลื่อนที่ของผู้เข้ารับการทดสอบต้องทำบนหน้าจอแสดงผล

เมื่อเริ่มทำการทดสอบ โปรแกรมจะนับถอยหลัง 5 วินาที เมื่อสิ้นสุดการนับถอยหลังผู้ป่วยจะต้องขยับวัตถุทรงกลมสีแดงขึ้นไปยังกรอบสีเขียวด้านบนให้เร็วที่สุด หากผู้ป่วยสามารถขยับลูกบอลสีแดงไปยังกรอบสีเขียวด้านบนได้ วิดีโอจะสิ้นสุดการบันทึกเป็นอันจบการทดสอบ เมื่อเก็บผลการทดสอบของผู้ป่วยเสร็จสิ้นจึงเก็บผลจากคนปกติด้วยวิธีการเดียวกันนี้ดังผลในตารางที่ 4.2

การเปรียบเทียบระหว่างผลจากผู้ป่วยและคนปกติ		
	เวลา (วินาที)	ผลการทดสอบ
ผลจากผู้ป่วย	5.48	
ผลจากคนปกติ	2.23	

ตารางที่ 4. 2 แสดงการเปรียบเทียบผลลัพธ์ของผู้ป่วยและคนปกติ

จากผลการทดสอบจะเห็นได้ว่าผู้ป่วยใช้เวลาในการเคลื่อนวัตถุทรงกลมสีแดงมากกว่าคนปกติมากกว่า 2 เท่า รวมถึงการเรียงตัวของจุดสีแดงของผู้ป่วยที่ไม่สม่ำเสมอและไม่เป็นระเบียบเหมือนคนปกติ เนื่องจากการเคลื่อนที่ที่ช้าและปัญหาของไหล่ บริเวณที่มีจุดกระจุกกันหนาแน่นทำให้สามารถวินิจฉัยได้ว่าการเคลื่อนที่ของแขนในบริเวณนั้นทำได้ยากและมีปัญหามากกว่าบริเวณอื่น

4.1.3 วิธีทำการทดสอบและผลการทดสอบของผู้ป่วยเปรียบเทียบระหว่างก่อนการทำ
 ภายภาพบำบัดและหลังการทำกายภาพบำบัด

ในการทดสอบในหัวข้อถัดไปคือการทดสอบที่เปรียบเทียบผลของตัวผู้ป่วยเอง โดยจะทำการ
 ทดสอบก่อนการทำกายภาพบำบัดและหลังการทำกายภาพบำบัด และนำมาเปรียบเทียบกันเพื่อดูความ
 ต่างของผลลัพธ์ที่ได้ดังตารางที่ 4.3

การเปรียบเทียบระหว่างผลจากผู้ป่วยและคนปกติ		
	เวลา (วินาที)	ผลการทดสอบ
ผลจาก ผู้ป่วย	8.88	
ผลจาก คน ปกติ	5.53	

ตารางที่ 4.3 แสดงการทดสอบก่อนการทำกายภาพบำบัดและหลังการทำกายภาพบำบัด

จากผลการทดสอบจะเห็นได้ว่าก่อนทำกายภาพบำบัดนั้นใช้เวลาที่มากกว่าหลังการทำกายภาพบำบัดถึง 3.35 วินาทีรวมถึงลักษณะของจุดที่ปรากฏขึ้นบนหน้าจอแสดงผลด้วย ผลการทดสอบก่อนการทำกายภาพบำบัดนั้นจุดจะเรียงถี่จะเกือบจะกลายเป็นเส้นตรง ส่วนผลการทดสอบหลังการทำกายภาพบำบัด จุดที่ปรากฏออกมามีความห่างและเป็นระเบียบมากกว่าผลก่อนการทำกายภาพบำบัดอยู่มาก แสดงให้เห็นถึงผลในเชิงบวกของผู้ป่วย

4.2 ทดสอบในกรณีของผู้ป่วยโรคเส้นเลือดในสมองตีบ (Brain Infarction)

4.2.1 ข้อมูลเบื้องต้นของผู้ป่วยที่ทำการทดสอบ

สำหรับผู้ป่วยที่มาทำการทดสอบในกรณีนี้เป็นผู้ป่วยเพศชาย อายุ 60 ปี มีประวัติเป็นโรคหัวใจมาก่อน ก่อนที่จะเป็นโรคเส้นเลือดในสมองตีบเนื่องจากหินปูนเกาะในเส้นเลือด ทำให้ไม่สามารถเคลื่อนไหวร่างกายได้โดยสะดวก หลังทำกายภาพบำบัดได้ระยะหนึ่งจึงมีอาการดีขึ้น แต่ก็ยังไม่หายเป็นปกติ

4.2.2 วิธีการทดสอบและผลการทดสอบ

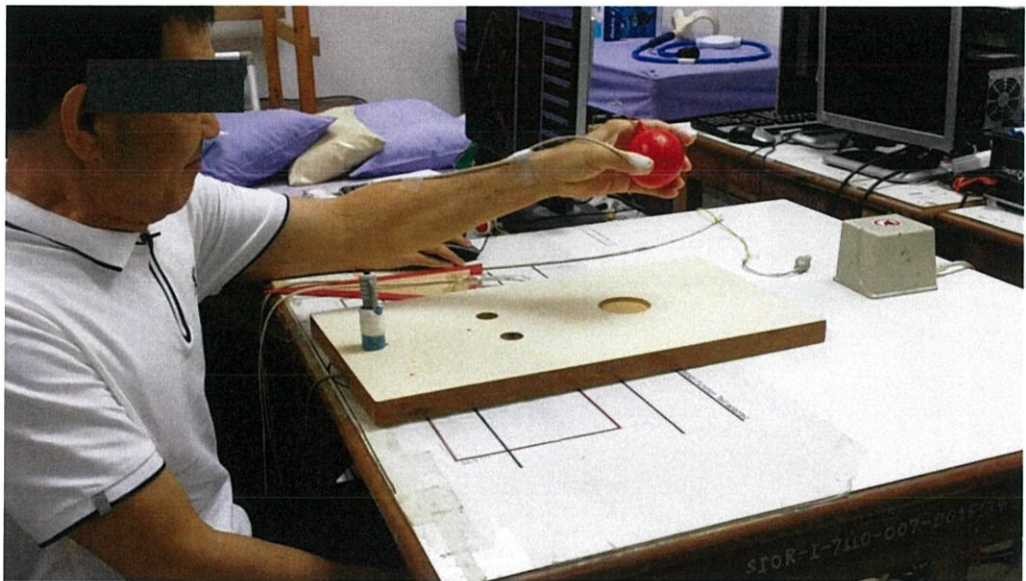
ในการทดสอบในกรณีนี้จะอ้างอิงวิธีการทดสอบมาจากการทดสอบจริงของ [9] เครื่อง motion monitor เป็นเครื่องที่ใช้เก็บบันทึกทิศทางการเคลื่อนไหวในแกนสามมิติด้วยแม่เหล็กไฟฟ้า ใช้เซนเซอร์ 3 ตัวติดที่มือเพื่อดูการเคลื่อนที่ดังภาพที่ 4.3



รูปที่ 4.3 ภาพแสดงการทดสอบโดยใช้เครื่อง motion monitor

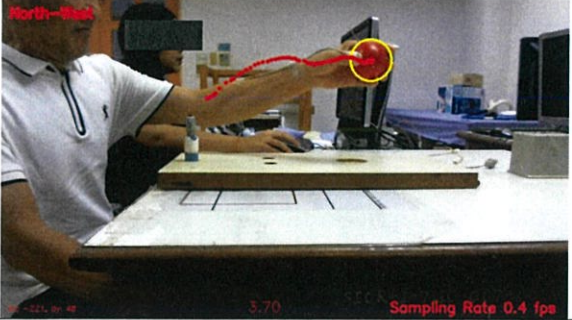
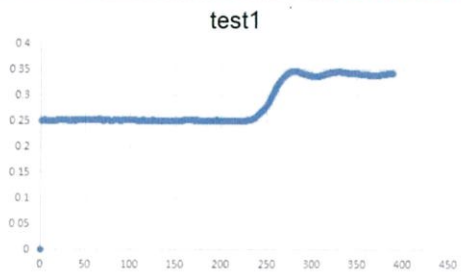
การทดสอบจริงนั้นผู้ทดสอบจะให้ผู้ป่วยใช้มือกดปุ่มเอาไว้ เมื่อเสียงสัญญาณออกตัวดังขึ้น ผู้ป่วยจะต้องยื่นมือไปหยิบวัตถุให้เร็วที่สุดเท่าที่จะเป็นไปได้ ซึ่งวัตถุในการทดสอบนี้เป็นแท่งไม้ทรงกระบอกที่อยู่ด้านหน้า ผลลัพธ์ที่ได้ออกมาจะเป็นเวลาและค่าพิกัดในแกนสามมิติเพื่อนำไปแสดงผลในรูปแบบกราฟต่อไป

ส่วนในการทดสอบสำหรับงานวิจัยนี้ได้ทำการดัดแปลงจากการทดสอบจริงเล็กน้อย โดยเปลี่ยนจากการยื่นมือไปยกวัตถุทรงกระบอกเป็นถือวัตถุทรงกลมสีแดงที่โปรแกรมใช้ในการตรวจจับและหาค่าพิกัดในแกนสองมิติเอาไว้ ก่อนจะให้ยื่นแขนออกไปในตำแหน่งที่ระบุโดยเร็วที่สุดแทนดังภาพที่ 4.4

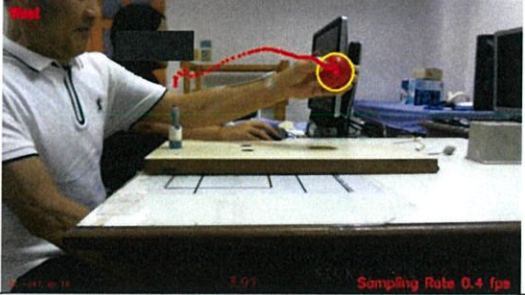
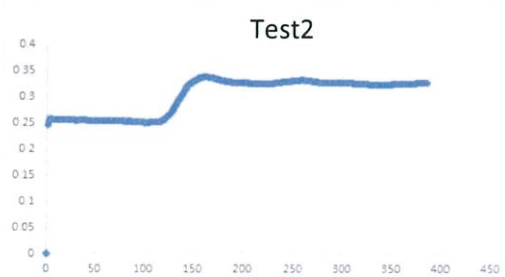


รูปที่ 4.4 ภาพแสดงการทดสอบโดยใช้โปรแกรม


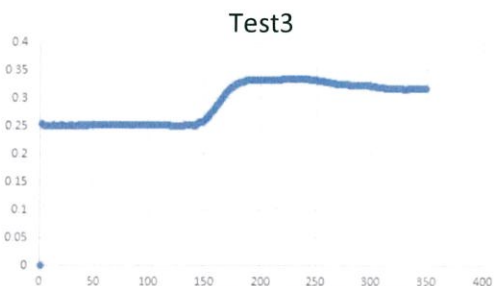
ก่อนเริ่มต้นการทดสอบนั้นผู้ทดสอบยังให้ผู้ป่วยกดปุ่มก่อนเริ่มเสียงสัญญาณอยู่เช่นเดิม เพื่อนำผลจากเครื่องมาพล็อตกราฟและเทียบกับเส้นทางการเคลื่อนที่ของมือจากโปรแกรม ซึ่งทำการทดสอบทั้งสิ้น 3 ครั้ง และได้ผลดังตารางที่ 4.4, 4.5 และ 4.6

	Test1	
	เวลา (วินาที)	ผลการทดสอบ
ผลจากโปรแกรม	3.70	
ผลจากเครื่อง	3.86	

ตารางที่ 4.4 แสดงผลการทดสอบครั้งที่ 1

	Test2	
	เวลา (วินาที)	ผลการทดสอบ
ผลจากโปรแกรม	3.91	
ผลจากเครื่อง	3.83	

ตารางที่ 4.5 แสดงผลการทดสอบครั้งที่ 2

	Test3	
	เวลา (วินาที)	ผลการทดสอบ
ผลจากโปรแกรม	3.30	
ผลจากเครื่อง	3.47	

ตารางที่ 4.6 แสดงผลการทดสอบครั้งที่ 3

ในการประมวลผลจากเครื่องจริงนั้นจะดึงข้อมูลเพียงสองแกนมาใช้ในการแสดงกราฟ เพื่อความสอดคล้องกับผลที่ได้จากโปรแกรม จากการทดสอบจะพบว่าจากเครื่องและผลจากโปรแกรมมีความใกล้เคียงกัน รวมถึงเวลาที่ได้ก็ใกล้เคียงกัน โดยในแต่ละการทดสอบ ค่าของเวลา มีความแตกต่างจากเครื่องจริงดังตารางที่ 4.7

เวลาจากเครื่องจริง	3.86	3.91	3.47
เวลาจากโปรแกรม	3.70	3.83	3.30
ค่าความแตกต่าง	2%	2%	4%

ตารางที่ 4.7 แสดงค่าความแตกต่างของเวลาระหว่างเครื่องจริงและโปรแกรม

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

งานวิจัยชิ้นนี้จัดทำขึ้นเพื่อศึกษาลักษณะการของผู้ป่วยที่มีปัญหาเกี่ยวกับส่วนของ upper extremity รวมทั้งแนวทางการวิจัย และวัดผลการรักษาเพื่อนำเทคโนโลยีที่มีราคาถูกลงและง่ายต่อการใช้งานมาประยุกต์ใช้ในการรักษาผู้ป่วยเหล่านี้

5.1 สรุปผลการดำเนินการ

1. จากการศึกษาลักษณะอาการของความผิดปกติในส่วนของ upper extremity ทำให้ได้ทราบถึงปัญหาที่เกิดขึ้นกับผู้ป่วย ดังนี้

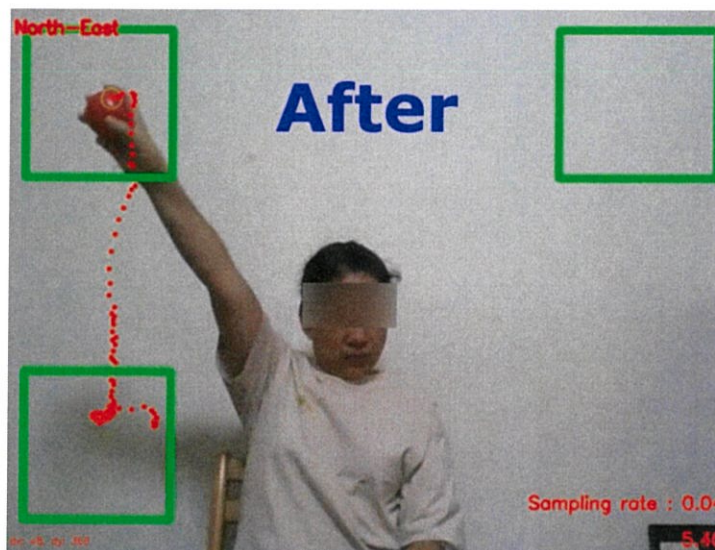
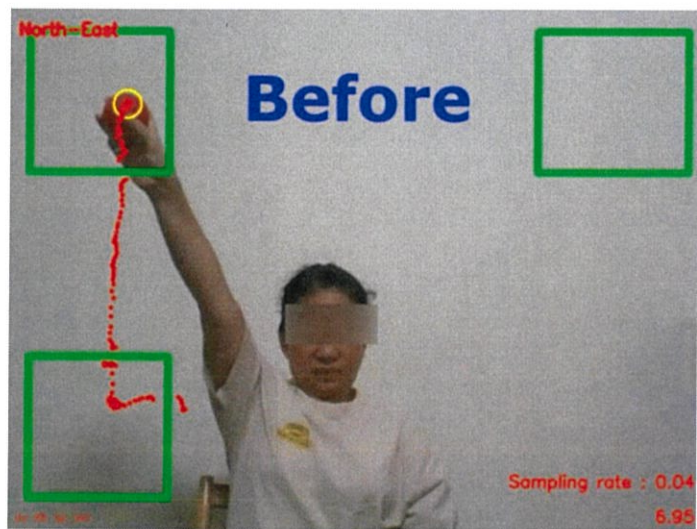
- 1.) ผู้ป่วยที่มีอาการผิดปกติในส่วนของ upper extremity จะไม่สามารถยกแขนได้ถึงจุดที่สูงที่สุดของการยกปกติได้
- 2.) ผู้ป่วยที่มีอาการผิดปกติในส่วนของ upper extremity ในขณะที่ยกแขนขึ้นบางรายอาจมีการเจ็บปวดบริเวณแขน
- 3.) ผู้ป่วยมีการเคลื่อนไหวที่ช้ากว่าคนปกติ

2. จากการศึกษาลักษณะและวิธีการทำกายภาพของผู้ป่วยที่มีความผิดปกติในส่วนของ upper extremity จากศูนย์กายภาพบำบัด คณะกายภาพบำบัด มหาวิทยาลัยมหิดล ทำให้ทราบถึงขั้นตอนการรักษาเบื้องต้นคือ ทางนักกายภาพจะทดสอบการเคลื่อนที่ด้วยเครื่อง motion monitor ที่ใช้หลักการของอิเล็กทรอนิกส์และติดตั้งเซ็นเซอร์ไว้ที่แขนของผู้ป่วยในจุดที่ต้องการ หลังจากนั้นทำการวัดแกน x, y และ z เมื่อวัดแกนเสร็จ จะให้ผู้ป่วยทำการทดลองด้วยการเคลื่อนมือไปข้างหน้าให้เร็วที่สุดซ้ำ ๆ จำนวนครั้งที่ทำนั้นขึ้นอยู่กับเจ้าหน้าที่กำหนด และจะมีการเพิ่มอุปสรรคขึ้นเรื่อย ๆ ผลที่ได้จากการทดลองจะเห็นความเร็วในการเคลื่อนที่ของมือ แกน 3 แนว และเวลาที่ใช้ในการเคลื่อนที่

3. เมื่อทราบถึงหลักการรักษาและการวัดผลของนักกายภาพแล้ว จึงได้จัดทำโปรแกรมเพื่อใช้ในการหาเส้นทางเดินของมือ (Hand Path) ในแนวแกน 2 มิติ โดยจะใช้การจับสีของวัตถุทรงกลมสีแดงเพื่อหาเส้นทางเดินของมือ (Hand Path) การวิเคราะห์จะใช้เป็นแบบออฟไลน์ โดยจะแบ่งเป็น 2 ส่วนคือ ส่วนการบันทึกวิดีโอคือจะกำหนดจุดเริ่มต้นของการเคลื่อนที่และจุดสุดท้ายไว้ หลังจากนั้นจะเป็นส่วนที่ 2 คือ

การนำวิดีโอที่ได้มาวิเคราะห์หาเส้นทางการเดินของมือ (Hand Path) ผลที่ได้จะแสดงเส้นทางการเดินของมือ (Hand Path) เวลาที่ใช้ในการเคลื่อนที่ทั้งหมด และ sampling rate ของวิดีโอ

4. การทดลองใช้กับผู้ป่วยจริงที่มีปัญหาความผิดปกติในส่วนของ upper extremity ในกรณีนี้คือผู้ป่วยมีอาการไหล่ติด เมื่อทำการทดลองจะเห็นว่าก่อนทำกายภาพ (ครั้งแรก) จะเห็นว่าจุดสีแดงที่แสดงเส้นทางการเดินของมือ (Hand Path) จะติดกัน ซึ่งหมายความว่าผู้ป่วยมีการเคลื่อนที่จากจุดแรกไปยังจุดสุดท้ายช้า และเมื่อหลังทำกายภาพ (ครั้งสุดท้าย) จะเห็นว่าจุดสีแดงที่แสดงเส้นทางการเดินของมือ (Hand Path) จะมีระยะห่างที่มากขึ้น ซึ่งหมายความว่าผู้ป่วยมีการเคลื่อนที่มีเร็วขึ้น



รูปที่ 5.1 รูปเปรียบเทียบก่อนและหลังทำกายภาพ

5.2 ปัญหาระหว่างการศึกษา

ในระหว่างที่ได้ศึกษาและวิจัยงานชิ้นนี้มีปัญหาที่เกิดขึ้นสามารถสรุปต่อไปนี้

1. ภาษาที่ใช้ในการเขียนโปรแกรมไม่แน่นอน เนื่องด้วยความไม่ชำนาญยกตัวอย่างเช่น ภาษา C# ทำให้เมื่อได้ทำการเขียนโปรแกรมเกิดปัญหาที่ตัวคำสั่งจนไม่สามารถดำเนินการในส่วนต่อไปได้ จึงทำให้เสียเวลาไปมากกว่าจะมาใช้ภาษา Python ซึ่งมีผู้เชี่ยวชาญหลายท่านคอยให้คำแนะนำ

2. แนวคิดในการเขียนโปรแกรมในตอนแรก คือ การให้ผู้ป่วยนั่งตรงกลางจอแล้วให้แขนสองข้างทำการทดลองพร้อมกัน แต่เมื่อได้ปรึกษารศ.ดร.จากรุกุล ตรีไตรลักษณ์ ซึ่งเป็นอาจารย์ของคณะกายภาพบำบัด มหาวิทยาลัยมหิดล จึงได้ทราบว่าขณะที่กายภาพผู้ป่วนั้นจะทำเพียงแขนข้างใดข้างหนึ่งที่ละข้าง ซึ่งเป็นผลให้ต้องทำรูปแบบของโปรแกรมใหม่เพื่อให้ตอบสนองความต้องการของผู้ใช้งานจริงให้มากที่สุด

3. เมื่อโปรแกรมเสร็จสมบูรณ์ได้มีการนำได้ทดลองกับผู้ป่วยจริง ซึ่งมีอาการของโรคพาร์กินสัน และกล้ามเนื้ออ่อนแรงทำให้ ผู้ป่วยมีความยากลำบากในการหยิบวัตถุทรงกลมสีแดงที่ได้จัดเตรียมไว้

5.3 ข้อเสนอแนะ

1. ในงานวิจัยชิ้นนี้ผลที่ได้จากการวิเคราะห์สามารถแสดงให้เห็นเพียงแค่แกน 2 มิติ (x, y) แต่ในการใช้งานจริงเพื่อให้ได้ผลที่แน่นอนและแม่นยำ การที่สามารถแสงแกน 3 มิติ (x, y และ z) ได้นั้นจะมีประโยชน์ในการวินิจฉัยการรักษามากขึ้น

2. ในการทดลองการผู้ป่วยจริงในบางครั้งผู้อาจจะมีอาการอื่นร่วมด้วย เช่น กล้ามเนื้ออ่อนแรง พาร์กินสัน ฯลฯ ซึ่งอาจจะมีผลทำให้ผู้ป่วยไม่สามารถจับวัตถุสีแดงได้ ดังนั้นการทำให้สามารถหาเส้นทางเดินของมือ (Hand Path) โดยไม่พวงวัตถุดังกล่าวจึงเป็นทางเลือกหนึ่งที่ควรศึกษาเพิ่มเติมเพื่อพัฒนาต่อไป

เอกสารอ้างอิง

- [1] OpenCV. (ม.ป.ป.). *Back ground subtraction*. เข้าถึงได้จาก <https://docs.opencv.org>
- [2] tutorialspoint. (ม.ป.ป.). *Edge detection* . เข้าถึงได้จาก <https://www.tutorialspoint.com>
- [3] OpenCV. (ม.ป.ป.). *Color Detection, VideoWriter, and DrawLine use OpenCv with Python* . เข้าถึงได้จาก <http://opencv-python-tutroals.readthedocs.io/en/latest>
- [4] normankoren. (มกราคม 2004). *Color management* . เข้าถึงได้จาก Norman Koren Photography: <http://www.normankoren.com>
- [5] colorizer. (ม.ป.ป.). *HSV color*. เข้าถึงได้จาก <http://colorizer.org>
- [6] OpenCV. (ม.ป.ป.). *Erosion and dilatation*. เข้าถึงได้จาก <https://docs.opencv.org>
- [7] marcuscode. (ม.ป.ป.). *Python*. เข้าถึงได้จาก <http://marcuscode.com/lang/python/introduction>
- [8] OpenCV . (ม.ป.ป.). เข้าถึงได้จาก OpenCV Library: <https://opencv.org/>
- [9] MotionMonitor. (2014). *The motion monitor*. เข้าถึงได้จาก <https://www.innsport.com>
- [10] pygame. (ม.ป.ป.). *Sound and Time on Video*. เข้าถึงได้จาก <https://www.pygame.org/news>