

ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

การพัฒนาซอฟต์แวร์ชุดคำสั่งติดตาม  
การเคลื่อนไหวของมือเพื่อการโต้ตอบกับคอมพิวเตอร์

DEVELOPMENT OF HAND TRACKING  
SOFTWARE LIBRARY FOR HUMAN-COMPUTER INTERACTION



ON  
16737  
2549

อาจารย์ที่ปรึกษา  
รศ. ดร. นพพร โชติกกำธร



\*H004449\*

เลขหมู่.....  
เลขทะเบียน..... 04449  
วัน,เดือน,ปี... - 5 ส.ย. 2551

b. 119 22813
i. ....

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน  
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ  
คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ภาคเรียนที่ 2 ปีการศึกษา 2549  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**DEVELOPMENT OF HAND TRACKING  
SOFTWARE LIBRARY FOR HUMAN-COMPUTER INTERACTION**



**A SYSTEM DEVELOPMENT PROJECT  
OF THE REQUIREMENT FOR THE DEGREE OF  
MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY  
FACULTY OF INFORMATION TECNOLOGY  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
2/ 2006  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**COPYRIGHT 2007**

**FACULTY OF INFORMATION TECHNOLOGY**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG** ศูนย์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การพัฒนาซอฟต์แวร์ชุดคำสั่งติดตามการเคลื่อนไหวของมือ เพื่อการโต้ตอบกับคอมพิวเตอร์
นักศึกษา	นายกิตติคุณ ขอดรัก
รหัสนักศึกษา	48066519
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2549
อาจารย์ที่ปรึกษา	รศ. ดร. นพพร โชติกกำจร

### บทคัดย่อ

รายงานฉบับนี้นำเสนอการพัฒนาชุดคำสั่งติดตามการเคลื่อนไหวของมือให้สามารถควบคุมการโต้ตอบกับระบบคอมพิวเตอร์ได้ โดยศึกษากระบวนการประมวลผลภาพเพื่อใช้ในการแบ่งส่วนภาพมือที่ต้องการ กระบวนการตรวจจับลักษณะท่าทางมือ และกระบวนการติดตามการเคลื่อนไหวของมือ แล้วทำการวิเคราะห์หาตำแหน่งพิกัดของมือในระบบ 3 มิติ มุมการหมุนของมือ 180 องศา (4 Degree of Freedom) ลักษณะท่าทางมือประกอบด้วยการกำมือ และการแบมือ และทิศทางการเคลื่อนที่ของมือ 8 ทิศทาง โดยทำการส่งข้อมูลข้างต้นผ่านเหตุการณ์ของชุดคำสั่งไปให้โปรแกรมประยุกต์ที่เขียนคำสั่งรับค่าไปใช้งานต่อไป

การพัฒนาชุดคำสั่งข้างต้นพัฒนาโดยใช้ภาษา C# บน Microsoft Visual Studio .NET ร่วมกับชุดคำสั่ง OpenCV ในการประมวลผลภาพ ทำให้สามารถนำไปประยุกต์ใช้ในระบบการติดตามการเคลื่อนไหวของมือบน .NET Platform โดยทำการพัฒนาเป็นโปรแกรมต้นแบบในการวาดภาพโดยใช้กล้องจับภาพการเคลื่อนไหวของมือ ในการหยิบจับเคลื่อนย้ายวัตถุ หมุนวัตถุและย่อขยายวัตถุที่อยู่ในโปรแกรมได้ แต่การใช้งานจำเป็นต้องมีการปรับแต่งการตั้งค่าของชุดคำสั่ง และการตั้งค่าของกล้องวิดีโอให้ถูกต้อง ชุดคำสั่งการติดตามการเคลื่อนไหวของมือจึงสามารถทำงานได้อย่างมีประสิทธิภาพ

<b>Title</b>	Development of Hand Tracking Software Library For Human-Computer Interaction
<b>Student</b>	Mr. Kittikhun Yodrak
<b>Student ID.</b>	48066519
<b>Degree</b>	Master of Science
<b>Programme</b>	Information Science
<b>Academic Year</b>	2006
<b>Advisor</b>	Assoc.Prof. Dr.Nopporn Chotikakamthorn

## ABSTRACT

This paper presents the development of hand tracking software library to retrieve information of hand movement interacting with the computer system. The proposed development focuses on studying hand image segmentation, hand gesture detection and hand movement tracking by using image processing theory. Image analysis is performed to estimate hand 3D position and orientation, as well as recognizing different hand postures, namely flat hand and fist hand. The hand tracking software library allows an application to receive hand information by using the event-driven model.

The hand tracking software library was implemented in C# on Microsoft Visual Studio .NET with OpenCV library. This allows the supported features to be used in the .NET platform application. A sample application was developed for hand painting by using video camera to capture hand movement when grabbing, moving, rotating and resizing the object in the application. The hand tracking software library can be performed properly with efficiency if the calibration of camera device and software library are configured correctly.

# กิตติกรรมประกาศ

โครงการเรื่อง “การพัฒนาซอฟต์แวร์ชุดคำสั่งติดตามการเคลื่อนไหวของมือ เพื่อการโต้ตอบกับคอมพิวเตอร์” นี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา รศ.ดร.นพพร โชติกกำธร ที่กรุณาให้คำปรึกษาแนะนำ และช่วยตรวจสอบแก้ไขข้อบกพร่องของโครงการนี้ ตลอดจนให้ความรู้และข้อคิดเห็นที่เป็นประโยชน์อย่างยิ่งต่อโครงการ

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่ร่วมเป็นกำลังใจ และให้การสนับสนุนในทุกๆเรื่อง ด้วยดีเสมอมา

สำหรับคุณงานความดีและประโยชน์อันพึงมาจากโครงการฉบับนี้ ข้าพเจ้าขอมอบให้กับ บิดามารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า



กิตติคุณ ยอศรี

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป.....	VII
สารบัญตาราง.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ในการพัฒนาระบบ.....	1
1.3 ขอบเขตของการพัฒนาระบบ.....	1
1.4 ขั้นตอนการดำเนินงาน.....	2
บทที่ 2 ทฤษฎีการประมวลผลภาพและการติดตามการเคลื่อนไหวของมือ.....	3
2.1 ทฤษฎีสีและเงา.....	3
2.1.1 ระบบสี RGB .....	3
2.1.2 ระบบสี HLS.....	4
2.2 ขั้นตอนการเตรียมภาพ.....	5
2.2.1 การนำเข้าข้อมูลภาพวิดีโอ .....	5
2.2.2 การแบ่งส่วนภาพ .....	6
2.2.3 รูปภาพขาว-ดำ .....	7
2.2.4 การแปลงภาพสีให้เป็นภาพขาว-ดำ.....	8
2.2.5 การกรองสัญญาณรบกวน.....	9
2.2.6 การปรับแต่งภาพด้วยมอร์โฟโลยี.....	9
2.2.7 การหาขอบภาพด้วยตัวดำเนินการแคนนี่.. .....	11
2.3 การหาดำแหน่งมือ.....	12
2.4 การติดตามมือ.....	12
บทที่ 3 การออกแบบโปรแกรม.....	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์ที่ปรึกษาเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

3.1 ชุดคำสั่งการติดตามการเคลื่อนไหวของมือ..... 14

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
3.1.1 โครงสร้างของชุดคำสั่ง.....	14
3.1.2 คลาสไดอะแกรมของชุดคำสั่ง.....	15
3.1.3 พจนานุกรมข้อมูลคลาส.....	15
3.1.4 Activity Diagram of Hand Tracking Component.....	20
3.1.5 กระบวนการทำงานของชุดคำสั่ง.....	21
3.2 โปรแกรมประยุกต์.....	30
3.2.1 โครงสร้างของโปรแกรม.....	30
3.2.2 ยูสเคสของโปรแกรมประยุกต์.....	30
บทที่ 4 การพัฒนาโปรแกรม.....	38
4.1 เครื่องมือที่ใช้ในการพัฒนา.....	38
4.1.1 ซอฟต์แวร์.....	38
4.1.2 ฮาร์ดแวร์.....	38
4.2 การพัฒนาโปรแกรมประยุกต์โดยใช้ชุดคำสั่งติดตามการเคลื่อนไหวของมือ.....	38
4.2.1 การนำเข้าชุดคำสั่ง.....	38
4.2.2 การใช้งานคำสั่ง.....	39
4.2.3 การใช้งานเหตุการณ์.....	39
4.3 การใช้งานโปรแกรมประยุกต์.....	41
4.3.1 คุณลักษณะของโปรแกรม.....	41
4.3.2 การตั้งค่าระบบติดตามการเคลื่อนไหวของมือ.....	42
4.3.3 การวาดรูป.....	43
บทที่ 5 การออกแบบส่วนต่อประสานกับผู้ใช้.....	44
5.1 ปัญหาและอุปสรรค.....	44
5.2 แนวทางประยุกต์ใช้ร่วมกับงานอื่นในขั้นต่อไป.....	44
5.3 ข้อเสนอแนะ.....	44
บรรณานุกรม.....	45

## สารบัญ (ต่อ)

	หน้า
ภาคผนวก.....	46
ภาคผนวก ก. Hand Tracking Component API Reference.....	47
ภาคผนวก ข. คู่มือการติดตั้งชุดคำสั่งติดตามการเคลื่อนไหวของมือ.....	52
ประวัติผู้เขียน.....	56



# สารบัญรูป

รูปที่	หน้า
2.1 แสดงระบบสีแบบ RGB.....	4
2.2 แสดงระบบสีแบบ HSL.....	5
2.3 แสดงความสัมพันธ์ระหว่างอัตราการแสดงผลภาพต่อวินาทีกับเวลาที่นำมาใช้ในการประมวลผล.....	5
2.4 แสดง HSL Filtering .....	7
2.5 แสดงแต่ละ Pixel ที่ถูกกำหนดให้มีค่า 0 (สีดำ) และ 1 (สีขาว).....	8
2.6 แสดงข้อมูลนำเข้าสำหรับกระบวนการไคเลชัน.....	10
2.7 แสดงผลลัพธ์จากการไคเลชัน.....	10
2.8 แสดงผลลัพธ์จากการอีโรชัน.....	11
2.9 (ก) ภาพต้นแบบที่ต้องการหารูปทรงกลม.....	
(ข) ภาพที่ผ่านการแปลงฮัฟเบววงกลม.....	12
2.10 แสดงการติดตามด้วย CAMSHIFT Algorithms.....	
(ก) ภาพมือแบบ Gray Scale.....	
(ข) ภาพมือที่ผ่านการ Segmentation.....	
(ค) ภาพมือที่แสดงผลการติดตามและตำแหน่ง.....	13
2.11 แผนภาพการทำการติดตามสีของวัตถุ (Color Object Tracking of CAMSHIFT Method).....	13
3.1 แสดงโครงสร้างและกระบวนการทำงานของชุดคำสั่งส่วนประกอบ โปรแกรมในการติดตามการเคลื่อนไหวของมือ.....	14
3.2 แสดงคลาสไคอะแกรมของชุดคำสั่ง.....	15
3.3 Activity Diagram of Hand Tracking Component .....	20
3.4 แสดงภาพมือที่ได้จากกล้อง.....	21
3.5 แสดงจุดภาพแต่ละจุดภายในภาพในแนวแถวและแนวคอลัมน์.....	21
3.6 แสดงภาพมือที่ได้จากกล้องขนาด 320x240 Pixels .....	22
3.7 แสดงแผนภาพการหาค่าความแตกต่างของภาพ.....	22
3.8 แสดงการแปลงภาพไปเป็น Binary.....	23
3.9 แสดงภาพมือที่ผ่านการ HSL Filtering .....	24
3.10 แผนภาพการแปลงค่าสี RGB เป็น HSL.....	24

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.11 แผนภาพการแปลงค่าสี HSL เป็น RGB .....	25
3.12 แสดงภาพการหาตำแหน่งและมุมการหมุนจากภาพมือ.....	26
3.13 แสดงภาพทิศทางการเคลื่อนที่ของจุด.....	27
3.14 แสดงแผนภาพการหาทิศทางการเคลื่อนที่จากจุด 2 จุด.....	27
3.15 แสดงภาพมือที่ผ่านการแปลงฮัฟแมนวงกลม.....	28
3.16 แผนภาพการวาดเส้นวงกลม.....	29
3.17 USE CASE การทำงานของโปรแกรมประยุกต์ตัวอย่าง.....	30
3.17 แผนภาพการวาดเส้นวงกลม.....	29
3.16 แผนภาพการวาดเส้นวงกลม.....	29
4.1 แสดงการ Add Reference เพื่อใช้งาน Hand Tracker Component .....	38
4.2 แสดงการใช้งาน Hand Tracker Component บน Visual Studio .NET .....	39
4.3 แสดง Delegate Events เหตุการณ์ที่ถูกส่งออกมาจาก Hand Tracker Component .....	39
4.4 แสดงหน้าต่างแสดงผลของโปรแกรมประยุกต์ (Hand Paint).....	41
4.5 แสดงหน้าต่างสำหรับแก้ไข.....	42
4.6 แสดงหน้าต่างสำหรับตั้งค่า.....	42
4.7 แสดงหน้าต่างสำหรับการทดสอบการติดตามมือ.....	42
4.8 แสดงโครงสร้างและกระบวนการทำงานของชุดคำสั่งส่วนประกอบโปรแกรม.....	43

# สารบัญตาราง

ตารางที่	หน้า
2.1 การแปลงองศาตามแนวแกนสำหรับการหาขอบภาพ.....	11
3.1 รายการคลาสของระบบติดตามการเคลื่อนไหวของมือ.....	16
3.2 Component: HandTracker.....	16
3.3 Class: HandTrackerParameter.....	17
3.4 Class: BinaryParam.....	17
3.5 Class: MorphologyParam.....	17
3.6 Class: HSLParam.....	17
3.7 Class: SmoothParam.....	18
3.8 Class: CannyParam.....	18
3.9 Class: CircularHoughParam.....	18
3.10 Class: CamshiftParam.....	18
3.11 Class: CvImageEventArgs.....	18
3.12 Class: HandPositionEventArgs.....	18
3.13 Class: HandDirectionEvent.....	19
3.14 Enumeration: DirectionType.....	19
3.15 Enumeration: HandType.....	19
3.16 Structure: Point3D.....	19
3.17 รายละเอียดเมนู Select Menu.....	31
3.18 รายละเอียดเมนู Select Shape .....	31
3.19 รายละเอียดเมนู Select Color .....	32
3.20 รายละเอียดเมนู Select Action .....	32
3.21 รายละเอียดเมนู Draw .....	33
3.22 รายละเอียดเมนู Draw Color .....	33
3.23 รายละเอียดเมนู Draw Shape .....	34
3.24 รายละเอียดเมนู Move Shape .....	34
3.25 รายละเอียดเมนู Rotate Shape .....	35
3.26 รายละเอียดเมนู Resize Shape .....	36
3.27 รายละเอียดเมนู Clear Screen .....	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันอุปกรณ์ได้ตอบเพื่อใช้ในการควบคุมหรือป้อนข้อมูลเข้าสู่คอมพิวเตอร์ (Human-Computer Interface) ที่รู้จักกันโดยทั่วไปได้แก่ อุปกรณ์คีย์บอร์ดในการป้อนข้อมูลเข้าสู่ระบบ และอุปกรณ์เมาส์ในการควบคุมและชี้ตำแหน่ง ซึ่งยังไม่สามารถตอบสนองความต้องการของมนุษย์ในการที่จะควบคุมและชี้ตำแหน่งในระบบงาน 2 มิติ และ 3 มิติ ได้อย่างมีประสิทธิภาพ

โดยปกติแล้วมนุษย์มักเคยชินกับการที่จะใช้มือในการควบคุมการทำงานต่างๆ บนคอมพิวเตอร์ จึงมีการใช้ศาสตร์ด้านคอมพิวเตอร์วิชัน (Computer Vision) มาตรวจจับการเคลื่อนไหวต่างๆ ของมือเพื่อใช้เป็นข้อมูลนำเข้าสู่ระบบเพื่อใช้ในการควบคุมสั่งการทำงาน หรือชี้ตำแหน่งได้ ดังนั้นเพื่อทำให้มนุษย์สามารถใช้การเคลื่อนที่ของมือ หรือการเคลื่อนไหวในลักษณะต่างๆ ของมือ มาใช้ควบคุมเพื่อสั่งให้ระบบทำงาน หรือใช้ชี้ตำแหน่งต่างๆ ได้นั้น ยังต้องผ่านกระบวนการในด้านการประมวลผลภาพที่ถ่ายภาพการเคลื่อนที่ของมือเอาไว้แล้วนำมาประมวลผลเพื่อหาตำแหน่ง พิกัดของมือ และทิศทางของการเคลื่อนที่ของมือ รวมไปถึงการแสดงท่าทางต่างๆ ของมือ ให้ได้เสียก่อนจึงจะสามารถนำข้อมูลต่างๆ เหล่านี้มาใช้เป็นข้อมูลนำเข้าสู่ระบบเพื่อตีความหมายให้เป็นคำสั่ง หรือการชี้ตำแหน่งใดๆ ที่สามารถได้ตอบกับคอมพิวเตอร์ได้

### 1.2 วัตถุประสงค์ในการพัฒนาระบบ

เพื่อพัฒนาโปรแกรมที่สามารถควบคุมเพื่อสั่งให้ระบบทำงาน หรือชี้ตำแหน่งต่างๆ ในระบบ 2 มิติ หรือ 3 มิติ ในระบบคอมพิวเตอร์ได้โดยอาศัยการประมวลผลภาพดิจิทัลจากข้อมูลนำเข้าของการเคลื่อนไหวและแสดงท่าทางของมือ โดยใช้อุปกรณ์กล้องวิดีโอ ในการจับภาพเพื่อนำเข้าสู่โปรแกรมโดยพัฒนาโปรแกรมในลักษณะเชิงส่วนประกอบที่สามารถนำไปใช้เป็นส่วนติดต่อร่วมกับโปรแกรมประยุกต์ต่างๆ ที่ต้องการใช้การเคลื่อนที่ของมือในการชี้ตำแหน่ง หรือสั่งงาน

### 1.3 ขอบเขตของการพัฒนาระบบ

โครงการนี้เป็นการพัฒนาโปรแกรมที่สามารถควบคุมการชี้ตำแหน่งในระบบคอมพิวเตอร์ได้โดยใช้การเคลื่อนไหวของมือในการสั่งการแบบอัตโนมัติ โดยมีคุณลักษณะดังนี้

1. โปรแกรมสามารถรับข้อมูลนำเข้าจากข้อมูลภาพวิดีโอที่ถ่ายภาพการเคลื่อนไหวของมือ โดยใช้กล้องวิดีโอขนาดเล็ก (Web Camera) ในการบันทึกภาพแบบเวลาจริง (Real-Time)

2. ข้อมูลภาพนำเข้าเป็นข้อมูลภาพของมือถ่ายหน้าตรง (เห็นฝ่ามือ) ทั้งแบมือ และกำมือ ภายนอก  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ข้อมูลภาพนำเข้าเป็นข้อมูลภาพที่ไม่มีส่วนประกอบของฉากหลังที่เคลื่อนไหวอยู่ตลอดเวลา หรืออยู่ในสภาพแวดล้อมแบบเปิด (ไม่มีแสงแดดส่อง)

4. โปรแกรมสามารถตรวจจับหาคำแหน่งการเคลื่อนที่ของมือ 4 Degree of Freedom (4 DOF) คือ ตำแหน่งในระบบ 3 มิติ (แนวแกน X, Y, Z) การหมุน (ในแนวแกน Z) และทิศทาง การเคลื่อนที่ของมือ 8 ทิศทาง (ขึ้น, ลง, ซ้าย, ขวา, เฉียงขึ้นซ้าย, เฉียงลงซ้าย, เฉียงขึ้นขวา และ เฉียงลงขวา)

5. โปรแกรมมีลักษณะเป็นเชิงส่วนประกอบที่สามารถนำไปประกอบเข้ากับ โปรแกรมประยุกต์ทั่วไปที่พัฒนาด้วย .NET Platform โดยชุดโปรแกรมประกอบไปด้วย Class Libraries ซึ่งถูกรวมอยู่ใน DLL (Dynamic Link Libraries)

6. พัฒนาตัวอย่างโปรแกรมประยุกต์ที่สามารถใช้ชุดคำสั่ง โปรแกรมเชิงส่วนประกอบในการติดตามการเคลื่อนไหวของมือมาใช้งานร่วมกัน

#### 1.4 ขั้นตอนการดำเนินงาน

โครงการฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความจำเป็นของโครงการ ความมุ่งหมาย วัตถุประสงค์ในการ และขอบเขตในการพัฒนาโครงการ

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในการพัฒนาโครงการ กระบวนการแบ่งส่วนภาพ ตรวจจับองค์ประกอบภาพ ติดตามการเคลื่อนไหว เป็นต้น

บทที่ 3 กล่าวถึงการออกแบบโปรแกรมที่ทำหน้าที่ในการรับข้อมูลภาพมาประมวลผลภาพทางดิจิทัลเพื่อสั่งให้ระบบทำงาน

บทที่ 4 กล่าวถึงการพัฒนาโปรแกรม รายละเอียดการพัฒนาในเชิงเทคนิค โครงสร้างของโปรแกรม คุณลักษณะของอุปกรณ์ที่ใช้กับ โปรแกรม และส่วนการแสดงผลของระบบ การควบคุมการชี้ตำแหน่ง

บทที่ 5 กล่าวถึงบทสรุปและข้อเสนอแนะ สรุปผลการทดลองการควบคุมการชี้ตำแหน่งด้วยการเคลื่อนไหวของมือแบบต่างๆ ปัญหาและอุปสรรคที่เกิดขึ้น ไปจนถึงแนวทางในการนำไปศึกษาและประยุกต์ใช้ต่อไปในอนาคต

## บทที่ 2

# ทฤษฎีการประมวลผลภาพ และการติดตามการเคลื่อนไหวของมือ

ในหัวข้อนี้จะกล่าวถึงทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องในการพัฒนาโครงการ และทฤษฎีที่ถูกนำมาใช้ในการประมวลผลภาพเพื่อให้สามารถตรวจจับการเคลื่อนไหวของมือ แล้วหาตำแหน่งและทิศทาง เพื่อส่ง ไปยังระบบควบคุมเพื่อสั่งงานให้ระบบคอมพิวเตอร์ทำงาน

### 2.1 ทฤษฎีสีและเงา (Colors and Shading)

สี คือ ลักษณะความเข้มของแสงที่ปรากฏแก่สายตาให้เห็นเป็นสี โดยผ่านกระบวนการรับรู้ด้วยตามองจะรับข้อมูลจากตา โดยที่ตาได้ผ่านกระบวนการวิเคราะห์ข้อมูลพลังงานแสง มาแล้วผ่านประสาทสัมผัสการมองเห็นผ่านศูนย์สับเปลี่ยนในสมอง ไปสู่ศูนย์การมองเห็น ภาพ การสร้างภาพ หรือการมองเห็นก็คือ การที่ข้อมูลได้ผ่านการวิเคราะห์ แยกแยะให้เรารับรู้ถึงสรรพสิ่งรอบตัว โดยการมองเห็นภาพจากคอมพิวเตอร์นั้นเป็นการประกอบขึ้นของจุดภาพที่มีค่าของสีอยู่ในจุดภาพนั้น จำนวนสีสูงสุดที่เป็นไปได้ ของแต่ละจุดภาพขึ้นอยู่กับจำนวนบิตที่ใช้เมื่อมีการกำหนดให้ขนาดของบิตต่อจุด มากขึ้นจะทำให้จำนวนของสีมากขึ้นด้วย ตัวอย่างเช่น

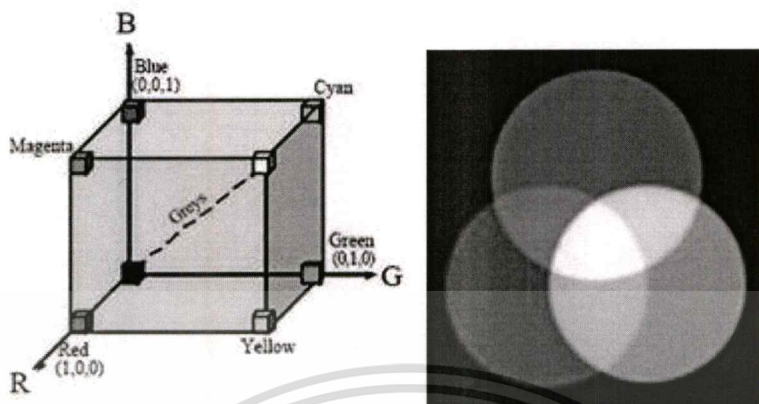
1 บิต =	2 <sup>1</sup> =	2	สี
2 บิต =	2 <sup>2</sup> =	4	สี
4 บิต =	2 <sup>4</sup> =	16	สี
8 บิต =	2 <sup>8</sup> =	256	สี
16 บิต =	2 <sup>16</sup> =	65536	สี

มาตรฐานของสีที่ใช้อยู่ในปัจจุบันมีอยู่หลายระบบด้วยกัน ทั้งนี้จะขึ้นอยู่กับนำไปใช้ แต่โดยทั่วไปแล้วทุกมาตรฐานจะมีแนวคิดเดียวกันคือ การแทนจุดสีด้วยจุดที่อยู่ภายในสเปส 3 มิติ โดยจะมีแกนอ้างอิงสำหรับจุดสีนั้นในสเปสซึ่งแต่ละแกนจะมีความเป็นอิสระต่อกัน

#### 2.1.1 ระบบสี RGB (Foley, James D. et.al. 1982)

ระบบสี RGB เป็นระบบสีที่เกิดจากการรวมกันของแสงสีแดง เขียวและน้ำเงิน โดยมีการรวมกันแบบ Additive ซึ่งโดยปกติจะนำไปใช้ในจอภาพแบบ CRT (Cathode ray tube) ในการใช้เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานระบบสี RGB ยังมีการสร้างมาตรฐานที่แตกต่างกันออกไปที่นิยมใช้งานได้แต่ RGB-CIE และ RGB-NTSC



รูปที่ 2.1 แสดงระบบสีแบบ RGB

### 2.1.2 ระบบสี HSL (Foley, James D. et.al. 1982)

HSL (Hue/Saturation/Luminance) เป็นการพิจารณาสีโดยใช้องค์ประกอบต่างๆดังนี้

2.1.2.1 Hue เป็นค่าตัวแทนสีที่ถูกแทนด้วย องศาต่างๆ Saturation เป็นค่าความเข้มของสี

$$H = \begin{cases} \theta & B \leq G \\ 360 - \theta & B > G \end{cases} \quad \dots (2.1)$$

$$\theta = \cos^{-1} \left[ \frac{\frac{1}{2}[(R-G) + (R-B)]}{[(R-G)^2 + (R-B)(G-B)]^{1/2}} \right] \quad \dots (2.2)$$

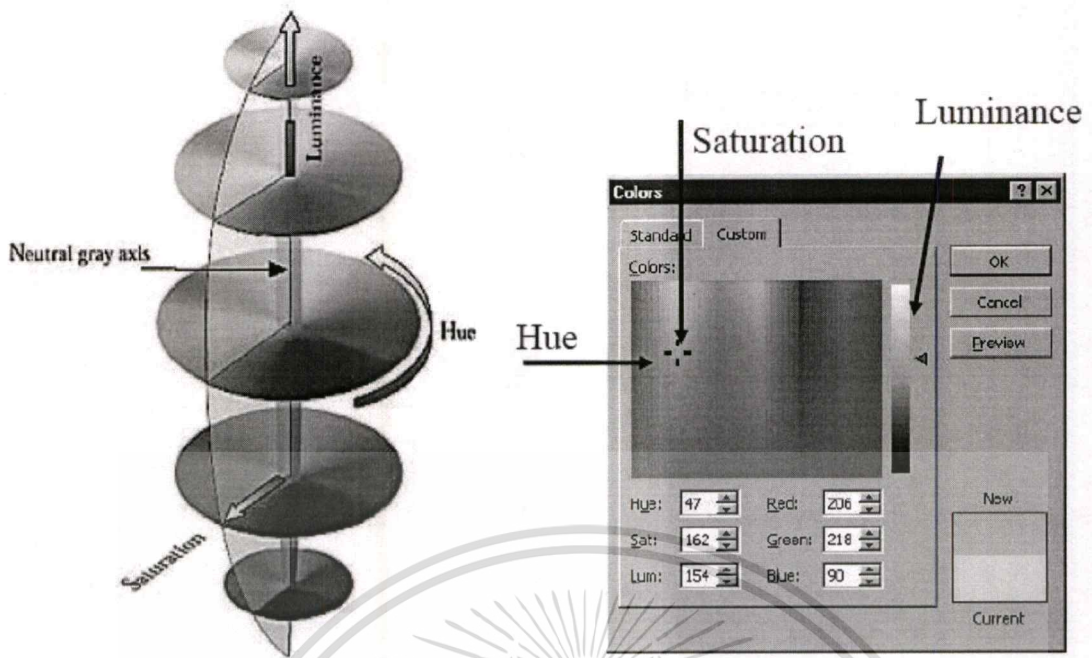
2.1.2.2 Saturation เป็นค่าความเข้มของสีจะมีค่าไม่เกิน 1

$$S = 1 - \frac{3}{(R+G+B)} [\min(R, G, B)] \quad \dots (2.3)$$

2.1.2.3 Luminance (ค่าความสว่างของสี) ซึ่งสามารถวัดได้? โดยค่าความเข้มของความสว่างของแต่ละสีที่ประกอบกัน

$$I = \frac{1}{3}(R+G+B) \quad \dots (2.4)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

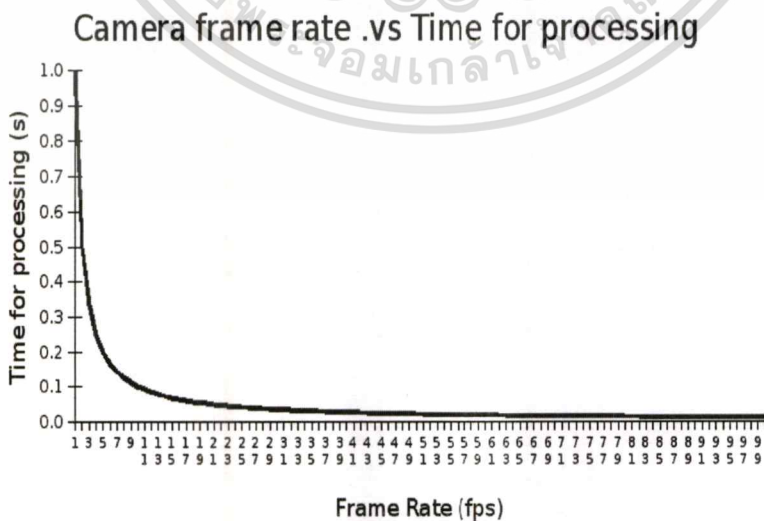


รูปที่ 2.2 แสดงระบบสีแบบ HSL

## 2.2 ขั้นตอนการเตรียมภาพ (Image Preprocessing)

### 2.2.1 การนำเข้าข้อมูลภาพวิดีโอ (Video Acquisition)

ในระบบการติดตามด้วยแสงโดยทั่วไปจะต้องมีช่วงเวลาที่ใช้ในการประมวลผลภาพระหว่างเฟรม เพื่อให้สามารถประมวลผลภาพได้ตามเวลาจริง (Real-Time Processing) ดังนั้นอัตราแสดงผลภาพต่อวินาทีจึงมีความสัมพันธ์กับเวลาที่ใช้ในการประมวลผลภาพในแต่ละเฟรม ดังรูปที่ 2.3



รูปที่ 2.3 แสดงความสัมพันธ์ระหว่างอัตราการแสดงผลภาพต่อวินาที

กับเวลาที่นำมาใช้ในการประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการแข่งขันเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2.2 การแบ่งส่วนภาพ (Image Segmentation)

### 2.1.2.1 การลบฉากหลัง (Background Subtraction)

การติดตามหาตำแหน่งของมือจากข้อมูลนำเข้าที่ได้จากภาพวิดีโอจะประกอบไปด้วยภาพของสิ่งที่ต้องการซึ่งอยู่ด้านหน้าของกล้องวิดีโอ (Foreground) และภาพฉากหลัง (Background) ที่ไม่ต้องการที่อาจเข้ามาในภาพด้วย ดังนั้นก่อนที่จะเข้าสู่ขั้นตอนของการหาตำแหน่งและติดตามการเคลื่อนไหวของมือ สิ่งสำคัญจึงอยู่ภาพที่จะเข้าสู่กระบวนการเหล่านี้จะต้องเป็นภาพที่มีข้อมูลของสิ่งที่เราสนใจเท่านั้น เพื่อให้การประมวลผลเป็นไปอย่างถูกต้อง จึงต้องมีกระบวนการในการลบฉากหลังที่ไม่ต้องการเหล่านี้ออกไป

(Hight, Ronald. 2004; Gerardin, Peggy. n.d.) โดยหลักการทำงานของเทคนิคนี้จะใช้วิธีการตรวจสอบภาพทั้งภาพเพื่อหาตำแหน่งของภาพที่ไม่มีการเปลี่ยนแปลงในช่วงเวลาที่กำหนด เพื่อทำการตัดข้อมูลที่ไม่มีเปลี่ยนแปลงเหล่านี้ทิ้งไปจากภาพ กล่าวคือจะทำการสนใจข้อมูลภาพที่มีการเคลื่อนไหวเท่านั้น ทำให้ในความเป็นจริงการใช้เทคนิคนี้อาจไม่มีประสิทธิภาพในการนำมาใช้ในระบบติดตามการเคลื่อนที่ของมือ อันเนื่องมาจากสาเหตุดังนี้

1) ภาพฉากหลังอาจมีการเปลี่ยนแปลงอยู่ตลอดเวลา หรืออาจไม่มีการเปลี่ยนแปลงเลยขึ้นอยู่กับสภาพแวดล้อมที่ใช้กล้องวิดีโอถ่ายภาพ รวมทั้งยังมีปัจจัยภายนอกอีกเช่น แสงสว่าง ทั้งที่เป็นแสงสว่างจากดวงอาทิตย์ หรือแสงสว่างจากแหล่งกำเนิดแสง ซึ่งอาจทำให้มีการเปลี่ยนแปลงของภาพฉากหลังในขณะที่กำลังประมวลผลภาพทำให้การตรวจสอบขาดประสิทธิภาพในการลบภาพฉากหลังออกจากภาพที่ได้

2) การเปลี่ยนแปลงจากสภาพแวดล้อมของโครงสร้างของภาพเช่น การเคลื่อนไหวของต้นไม้ การโบกสะบัดของธง รวมถึงการเปลี่ยนแปลงที่อาจเกิดขึ้นจากอุปกรณ์ที่ใช้ถ่ายภาพวิดีโอในตัวเอง ที่มีการสั่นไหวเกิดขึ้น หรือแม้กระทั่งสัญญาณรบกวน (noise) ที่อาจเกิดขึ้น ทำให้ภาพมีการเปลี่ยนแปลงอยู่ตลอดเวลาทั้งภาพ ทำให้ไม่สามารถทำการลบฉากหลังที่ไม่ต้องการออกไปได้

ดังนั้นหลักการทำงานของอัลกอริทึมในการลบภาพฉากหลังอย่างง่าย สามารถอธิบายได้ดังนี้ แบบจำลองของภาพฉากหลังอย่างง่ายนี้ ซึ่งให้สันนิษฐานว่าแสงสว่างในแต่ละจุดภาพ (Pixel) ของฉากหลังมีการกระจายที่แตกต่างกัน และเป็นอิสระต่อกัน ดังนั้นคุณลักษณะของภาพฉากหลังนี้สามารถคำนวณได้จากการบวกสะสมของเฟรมวิดีโอจำนวนมาก กล่าวคือต้องทำการหาผลรวมของจำนวนจุดภาพในแต่ละพื้นที่  $scene(x, y)$  และหาผลรวมของพื้นที่ของแต่ละภาพในพื้นที่นั้นๆ  $scene2(x, y)$  โดยให้  $N$  เป็นจำนวนของเฟรมวิดีโอที่ทำการถ่ายภาพ ดังนั้นค่าเฉลี่ยของจุดภาพสามารถคำนวณได้ด้วยดังนี้

$$mean_{(x,y)} = \frac{\sum scene_{(x,y)}}{N}$$

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) ของแต่ละจุดสามารถคำนวณได้ดังนี้

$$\sigma_{(x,y)} = \sqrt{\left( \frac{\sum scene_{(x,y)}^2}{N} - \left( \frac{\sum scene_{(x,y)}}{N} \right)^2 \right)} \quad \dots (2.6)$$

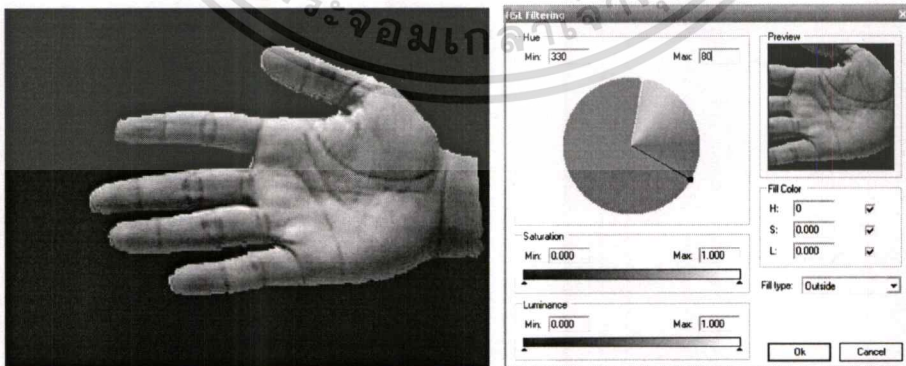
สุดท้ายสามารถพิจารณาจุดภาพใดๆ ที่มีการเคลื่อนไหว เมื่อจุดภาพดังกล่าวเป็นไปตามเงื่อนไขดังนี้

$$mean_{(x,y)} = \frac{1}{N} \sum_{j=i}^{i+N-1} scene_{(x,y)} \quad \dots (2.7)$$

### 2.1.2.2 การแบ่งส่วนภาพจากสีผิว (Skin Color Segmentation)

จากเทคนิคที่ใช้ในการคัดแยกภาพที่ไม่ต้องการออกไปจากองค์ประกอบภาพข้างต้นยังมีประสิทธิภาพไม่เพียงพอที่จะนำมาใช้ในกระบวนการติดตามการเคลื่อนไหวของมือ เนื่องจากภาพที่มีการเคลื่อนไหวหรือเปลี่ยนแปลง อาจมีองค์ประกอบอื่นที่เชื่อมโยงติดไปด้วยเช่น การเคลื่อนไหวของแขน เสื้อผ้า ศรีษะ หรือส่วนอื่นๆ ที่เข้ามาปะปนอยู่ในภาพ ทำให้ไม่สามารถคัดแยกภาพที่ต้องการออกมาได้ และส่งผลให้ไม่สามารถระบุหาตำแหน่งหรือขอบเขตของมือที่ต้องการได้

เทคนิคการคัดแยกภาพ โดยใช้สีในแต่ละจุดภาพเป็นตัวกำหนดนั้นจึงเป็นอีกทางเลือกหนึ่ง ซึ่งมีประสิทธิภาพเช่นเดียวกัน ในงานศึกษานี้ได้นำเอาวิธีการแบ่งจุดสีของภาพโดยใช้วิธีการ HSL filtering โดยกำหนดช่วงของค่า Hue ที่อยู่ในช่วงของสีผิวที่ต้องการ ดังรูปที่ 2.4



รูปที่ 2.4 แสดง HSL Filtering

### 2.2.3 รูปภาพขาว-ดำ (Binary Image)

รูปภาพขาว-ดำ คือ รูปภาพที่ประกอบด้วยสีเฉพาะสีขาวและสีดำ ใน Digital images ภาพจะอยู่ในรูปของแผ่นตาราง โดยแต่ละช่องจะเป็นส่วนหนึ่งของภาพหรืออักษร เรียกแต่ละจุดหรือไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่องนั้นว่า pixel แต่ละ pixel จะถูกกำหนดให้มีระดับของความเข้ม (สีดำ สีขาว สีเทาหรือสีอื่น ๆ) ซึ่งแสดงให้อยู่ในรูปของรหัส Binary (0 และ 1) แต่ละ pixel ก็จะถูกแทนด้วย Binary digital (bits) จะถูกเก็บเป็นลำดับในคอมพิวเตอร์ โดยส่วนใหญ่แล้วคอมพิวเตอร์สามารถวิเคราะห์ข้อมูลจากรูปภาพขาวดำได้ง่ายกว่ารูปภาพแบบอื่น ๆ เนื่องจากรูปภาพขาวดำมีค่าข้อมูลของสีเพียง 2 ค่า ซึ่งแทนด้วยค่า 0, 1 หรือ 0, 255 โดยที่ 0 แทนสีดำ และ 1 หรือ 255 แทนสีขาว

1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	0	0	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	0	0	0	0	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	0	0	0	1	1	0	0	0	1
1	1	1	1	1	1	1	1	1	1

รูปที่ 2.5 แสดงแต่ละ Pixel ที่ถูกกำหนดให้มีค่า 0 (สีดำ) และ 1 (สีขาว)

#### 2.2.4 การแปลงภาพสีให้เป็นภาพขาว-ดำ (Thresholding) (Gerardin, Peggy. n.d.)

Thresholding เป็นกระบวนการแปลงภาพสีให้มีการแสดงผลได้แค่ 2 ระดับ คือ ขาว และ ดำ โดยจะแปลงข้อมูลภาพให้เป็นภาพ binary (Binary Image) มีกระบวนการแปลงภาพที่มีความเข้มหลายระดับ (Multilevel Image) ให้เป็นภาพที่มีความเข้มเพียง 2 ระดับ หรือ 1 บิต (bit) คือ 0 และ 1 โดย 0 แทนด้วยจุดที่มีภาพสีขาว และ 1 แทนด้วยจุดที่มีภาพสีดำ

Thresholding Technique คือการพิจารณาจุด pixel ในภาพว่าจุดใดควรจะเป็นจุดขาว หรือ จุดใดควรจะเป็นจุดที่มีค่าเท่ากับ 1 โดยจะทำการเปรียบเทียบค่าของแต่ละ pixel ( $f(x,y)$ ) กับค่าคงที่ที่เรียกว่า Threshold (Threshold Value) เทคนิคนี้นิยมใช้กันมากในกรณีที่มีความแตกต่างระหว่างวัตถุ (Object) และพื้นหลัง (Background) ซึ่งเป็นลักษณะเดียวกับสีขาวของป้ายทะเบียน และสีดำของตัวอักษรบนป้ายทะเบียนรถยนต์ ค่า pixel ในภาพที่มีค่าน้อยกว่าค่า Threshold จะถูกกำหนดเป็น 1 (จุดดำ) และถ้าค่าของ pixel ใด ๆ ในภาพมีค่ามากกว่าหรือเท่ากับค่า Threshold จะถูกกำหนดให้เป็น 0 (จุดขาว)

ในการทำภาพ Binary โดยการทำ Thresholding ให้ได้ภาพดีและคมชัด ต้องเกิดจากการเลือกค่า Threshold ที่ถูกต้องและเหมาะสม ถ้าเลือกค่า Threshold ไม่เหมาะสม เช่น ค่า Threshold ที่มากหรือน้อยจนเกินไป ภาพที่ได้จะขาดความคมชัดหรืออาจทำให้รายละเอียดของภาพขาด

หายไป หรือภาพที่ได้อาจจะมืดเกินไป หรือสว่างเกินไป หรืออาจจะเป็นภาพที่มีสิ่งรบกวน (Noise) เกิดขึ้น ทำให้ภาพผลลัพธ์ที่ได้ไม่ชัดเจน และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.5 การกรองสัญญาณรบกวน (Noise Filtering)

เป็นกระบวนการในการลดสัญญาณรบกวนภายในภาพ เนื่องจากขั้นตอนการทำ Binary Thresholding ยังให้ค่าของข้อมูลที่ไม่ดีพอ จำได้ใช้กระบวนการในการเกลี่ยสี (Smoothing) เพื่อเฉลี่ยค่าสีของภาพที่มีส่วนต่างของค่าสีมากเกินไป (จุด หรือ สัญญาณรบกวน) โดยใช้เทคนิค การคอนโวลูชันโดยใช้ค่ามัธยฐาน (Median Convolution)

ในการประมวลผลภาพการคอนโวลูชันคือการกระทำกันระหว่างเทมเพลต (Template) กับภาพ (image) เทมเพลตคือเมตริกซ์ขนาด  $n \times m$  ของชุดตัวเลขที่จะนำไปซ้อนทับภาพที่ตำแหน่งต่างๆเพื่อหาผลลัพธ์ของการคอนโวลูชันถ้ากำหนดให้เทมเพลต  $T(x, y)$  เป็นเทมเพลตขนาด  $n \times m$  และภาพ  $I(X, Y)$  มีขนาด  $N \times M$  การคอนโวลูชันระหว่างเทมเพลตกับภาพสามารถแสดงได้ดังสมการต่อไปนี้

$$I'(X, Y) = T * I = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} T(i, j) \cdot I(X-i, Y-j) \quad \dots (2.8)$$

โดย  $I'(X, Y)$  คือภาพผลลัพธ์จากการคอนโวลูชัน

### 2.2.6 การปรับแต่งภาพด้วยมอร์โฟโลยี (Morphology) (Fisher, S. Perkins. et.al. 2003)

เป็นกระบวนการที่ประกอบด้วยกลุ่มของการดำเนินการพื้นฐานกับภาพ ด้วยตัวดำเนินการ Minkowski และกฎของ DeMorgan โดยใช้ตัวดำเนินการกับภาพแบบไบนารี (Binary Image) ที่มีค่าสี 2 ค่า คือ สีขาวและสีดำ มอร์โฟโลยีประกอบด้วยการดำเนินการหลายวิธี ได้แก่ ไคเลชัน (Dilation) อีโรชัน (Erosion) การเปิด (Opening) การปิด (Closing) เป็นต้น

เมื่อกำหนดให้  $A$  และ  $B$  เป็นเซต (Set) จะได้ตัวดำเนินการ Minkowski ดังสมการต่อไปนี้

$$\text{Minkowski addition } A \oplus B = \bigcup_{\beta \in B} (A + \beta) \quad \dots (2.9)$$

$$\text{Minkowski subtraction } A \ominus B = \bigcap_{\beta \in B} (A + \beta) \quad \dots (2.10)$$

#### 2.1.6.1 ไคเลชัน (Dilation)

ไคเลชันเป็นหนึ่งในตัวดำเนินการของมอร์โฟโลยีที่ทำงานเกี่ยวกับพื้นที่ของภาพ เป็นกระบวนการเพิ่มขอบเขตของภาพให้มีขนาดใหญ่ขึ้น และทำให้ช่องว่างของภาพมีขนาดเล็กลง โดยอ้างอิงจากตัวดำเนินการ Minkowski สามารถเขียนสมการไคเลชัน ดังนี้

$$\text{Dilation } D(A, B) = A \oplus B = \bigcup_{\beta \in B} (A + \beta) \quad \dots (2.11)$$

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ  $A$  และ  $B$  เป็นเซต และ  $B = \{-\beta \in B\}$

การทำงานของตัวดำเนินการไคเลชัน จะรับค่าข้อมูลนำเข้า 2 ค่า คือ ภาพที่ต้องการทำการไคเลชัน และคู่ลำดับของจุดบนภาพหรือที่เรียกว่า Structuring Element

1	1	1
1	1	1
1	1	1

Set of coordinate points =

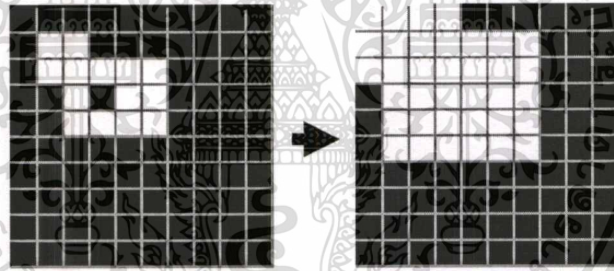
{ (-1, -1), (0, -1), (1, -1),

(-1, 0), (0, 0), (1, 0),

(-1, 1), (0, 1), (1, 1) }

รูปที่ 2.6 แสดงข้อมูลนำเข้าสำหรับกระบวนการไคเลชัน

เมื่อได้ข้อมูลนำเข้า ตัวดำเนินการจะตรวจสอบทุกจุดของภาพ จากนั้นจะเพิ่มขอบเขตของภาพใน 8 ทิศทาง (8-Connected) คือ บนซ้าย บน บนขวา กลางซ้าย กลางขวา ล่างซ้าย ล่าง และล่างขวา จะทำให้ได้ภาพที่มีขอบเขตกว้างขึ้น



รูปที่ 2.7 แสดงผลลัพธ์จากการไคเลชัน

### 2.1.6.2 อีโรชัน (Erosion)

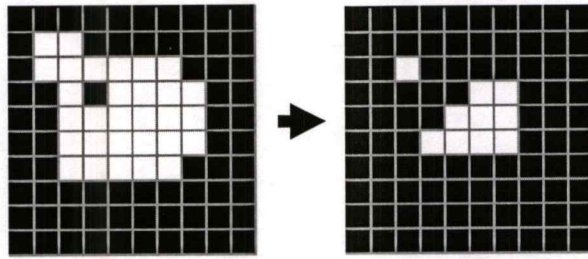
อีโรชันเป็นหนึ่งในตัวดำเนินการของมอร์โฟโลยีที่ทำงานเกี่ยวกับพื้นที่ของภาพ เป็นกระบวนการลดขอบเขตของภาพให้มีขนาดเล็กลง และทำให้ช่องว่างของภาพมีขนาดกว้างขึ้น โดยอ้างอิงจากตัวดำเนินการ Minkowski สามารถเขียนสมการอีโรชัน ดังนี้

$$\text{Erosion } E(A, B) = A \ominus (-B) = \bigcap_{\beta \in B} (A - \beta) \quad \dots (2.12)$$

เมื่อ  $A$  และ  $B$  เป็นเซต และ  $B = \{-\beta \in B\}$

การทำงานของตัวดำเนินการอีโรชัน จะรับค่าข้อมูลนำเข้า 2 ค่าเหมือนกับการไคเลชัน คือ ภาพที่ต้องการทำการอีโรชัน และคู่ลำดับของจุดบนภาพหรือที่เรียกว่า Structuring Element เมื่อได้ข้อมูลนำเข้า ตัวดำเนินการจะตรวจสอบทุกจุดของภาพ

จากนั้นจะลดขอบเขตของภาพออก 8 ทิศทาง (8-Connected) คือ บนซ้าย บน บนขวา กลางซ้าย กลาง กลางขวา ล่างซ้าย ล่าง และล่างขวา จะทำให้ได้ภาพที่มีขอบเขตลดลง



รูปที่ 2.8 แสดงผลลัพธ์จากการอีโรชัน

2.2.7 การหาขอบภาพด้วยตัวดำเนินการแคนนี่ (Canny Edge Detection) (Canny, J. 1986.)

การหาขอบด้วยวิธี Gradient Method เป็นการหาจุดต่ำสุดและจุดสูงสุดในรูปของอนุพันธ์อันดับหนึ่งของภาพ โดยจุดที่เป็นขอบจะอยู่ในส่วนที่เหนือค่า Threshold จึงอาจทำให้เส้นขอบที่ได้มีลักษณะหนา Canny Edge Detection นั้น สามารถจะหา edge ที่อยู่ในตำแหน่งที่ถูกต้องได้ และ edge ที่ได้มีขนาดแค่ 1 pixel โดยสามารถทำได้โดยการใช้รูปที่ทำการ Smooth จาก Gaussian filter แล้วมาคำนวณหา gradient magnitude และ gradient orientation โดยใช้ convolution masks

$$E_x = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ -1 & -1 \\ -2 & -2 \end{bmatrix} \text{ และ } E_y = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ -1 & -1 \\ -2 & -2 \end{bmatrix}$$

ซึ่ง gradient magnitude และ gradient orientation สามารถคำนวณได้จากสมการดังนี้

$$E(i, j) = \sqrt{E_x^2(i, j) + E_y^2(i, j)} \quad \dots (2.13)$$

$$\theta(i, j) = \arctan(E_x^2(i, j), E_y^2(i, j)) \quad \dots (2.14)$$

Gradient Orientation ที่คำนวณได้ จะถูกนำมาเปลี่ยนให้เป็น 4 แขน คือแกนแนวตั้ง แนวนอน และแนวทะแยงทั้งสอง โดยตารางดังต่อไปนี้

ตารางที่ 2.1 การแปลงองศาตามแนวแกนสำหรับการหาขอบภาพ

แนวแกน	องศา
แกนแนวนอน	-22.5 – 22.5 และ 157.5 – 202.5
แกนแนวทะแยง (ทิศตะวันออกเฉียงเหนือ)	22.5 – 67.5 และ 202.5 – 247.5
แกนแนวตั้ง	67.5 – 102.5 และ 247.5 – 292.5
แกนแนวทะแยง (ทิศตะวันตกเฉียงเหนือ)	102.5 – 157.5 และ 292.5 – 337.5

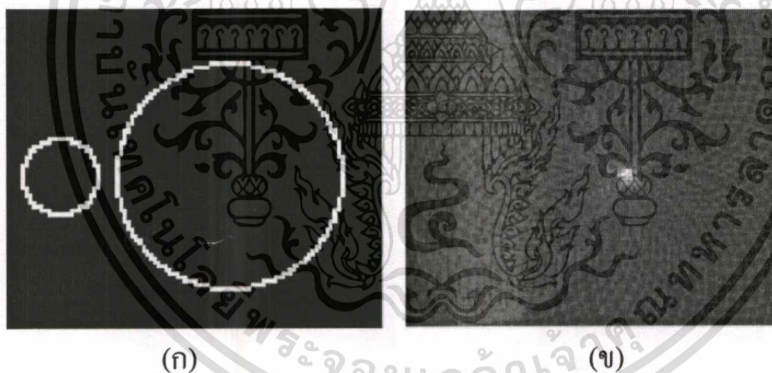
เอกสารนี้เป็นส่วนหนึ่งของการใช้งานเพื่อการศึกษาค้นคว้าวิจัยในโครงการวิจัยด้านการค้า  
ไม่ว่าการนำเข้าหรือการส่งออกสินค้าที่มีปริมาณการค้าสูง

แนวแกนนี้จะนำไปใช้ในการลดขนาดของ edge โดยในแต่ละ pixel ของ edge ถ้า pixels รอบข้างในแนวของ gradient orientation ใดที่ไม่ใช่ค่าสูงสุดจะไม่ถือว่าเป็น edge ดังนั้นจะมีเพียง pixel เดียวที่เป็น edge นอกจากนี้ยังสามารถทำการรวม edge เข้าเป็นกลุ่มเดียวกันได้โดยใช้ gradient orientation ที่ต่อเนื่องกันระหว่าง edge สอง pixels

## 2.3 การหาคำแหน่งมือ (Hand Detection)

### 2.3.1 การแปลงฮัฟแบบวงกลม (Circular Hough Transform)

กระบวนการแปลงฮัฟแบบวงกลมนิยมใช้ในการหารูปทรงที่มีลักษณะวงกลม โดยในงานวิจัยนี้ใช้การแปลงฮัฟแบบวงกลมในการหาจุดกึ่งกลางฝ่ามือ ด้วยการสร้างเส้นวงกลมให้มีรัศมีตามรูปทรงกลมที่ต้องการหา โดยให้ทำการสร้างเส้นวงกลมโดยมีจุดศูนย์กลางที่จุดพิกเซลของรูปภาพที่ต้องการค้นหา ดังรูปที่ 2.9 จะสังเกตเห็นว่ามีจุดของกลุ่มสีที่มีความเข้มสูงเนื่องจากบริเวณดังกล่าวเป็นจุดตัดกันระหว่างขอบรัศมีของเส้นรอบวงที่สร้างขึ้นจากจุดศูนย์กลางของขอบของภาพในแต่ละพิกเซล ทำให้เราสามารถหาจุดศูนย์กลางของรูปร่างทรงกลมที่ต้องการได้จากจุดพิกเซลที่มีค่าสีมากที่สุดนั่นเอง



รูปที่ 2.9 (ก) ภาพต้นแบบที่ต้องการหารูปทรงกลม

(ข) ภาพที่ผ่านการแปลงฮัฟแบบวงกลม

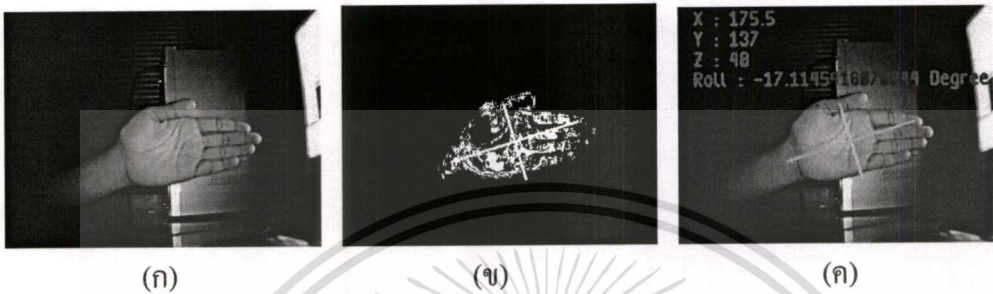
## 2.4 การติดตามมือ (Hand Tracking)

### 2.4.1 การติดตามด้วยอัลกอริทึม CAMSHIFT (Bradski, Gary R. 1998)

CAMSHIFT (Continuously Adaptive Mean Shift Algorithm) มีพื้นฐานมาจาก Mean Shift Algorithm ที่อาศัยความน่าจะเป็นในการกระจายตัวในการติดตามสีของวัตถุที่สนใจในแต่ละลำดับของ Frame ที่ดึงมาจากกล้องวิดีโอ ทั้งนี้การกระจายตัวของสีภายในภาพจะมีการเปลี่ยนแปลงอยู่ตลอดเวลา ดังนั้น Mean Shift Algorithm ต้องมีการเปลี่ยนแปลงตามไปด้วยอย่างต่อเนื่อง เพื่อให้สามารถติดตามหาตำแหน่งของสีของวัตถุได้อย่างถูกต้อง

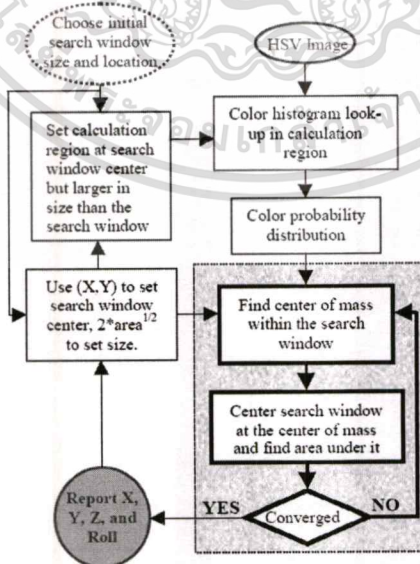
ในงานศึกษานี้ได้ใช้วิธีการ CAMSHIFT Algorithm ในการติดตามหาตำแหน่งของมือ เพื่อหาตำแหน่งของ พิกัด X, Y เป็นพิกัดที่อยู่บนภาพ ส่วน Z เป็นพิกัดของระยะห่างของกล้อง และ

แกนหมุน Roll เป็นมุมการหมุนในแนวแกน Z ดังนั้นในขั้นแรกต้องทำการหาความน่าจะเป็นในการกระจายตัวของสีภาพใน Frame ที่กำลังสนใจ โดยทำการแปลงอยู่ในระบบสี HSV (Hue Saturation Intensity) แล้วทำการสร้างฮิสโตแกรม เป็นกระบวนการที่เรียกว่าการทำภาพ Back-projection หลังจากนั้นทำการกำหนด Search window เพื่อใช้ในการคำนวณหาค่าเฉลี่ยของความน่าจะเป็นในการกระจายตัวของสีภายใน Search window นั้นๆ จะได้ตำแหน่งพิกัดมา ดังรูปที่ 2.10



รูปที่ 2.10 แสดงการติดตามด้วย CAMSHIFT Algorithms  
 (ก) ภาพมือแบบ Gray Scale (ข) ภาพมือที่ผ่านการ Segmentation  
 (ค) ภาพมือที่แสดงผลการติดตามและตำแหน่ง

หลังจากนั้นจึงทำการกำหนดจุดกึ่งกลางของ Search window ให้ไปอยู่ในตำแหน่งพิกัดที่ได้มาแล้วทำการคำนวณหาพื้นที่ใหม่ แล้วเริ่มขั้นตอนเดิมใหม่อีกครั้งไปเรื่อยๆ จนกว่าจะวนมาบรรจบกันที่กันกรวย (ค่าเฉลี่ยของความสามารในการกระจายตัวของสีมีค่าน้อยกว่าค่า Threshold ที่กำหนดไว้)



รูปที่ 2.11 แผนภาพการทำติดตามสีของวัตถุ  
 (Color Object Tracking of CAMSHIFT Method)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

# การออกแบบโปรแกรม

ในการพัฒนาโปรแกรมในโครงการนี้จะแบ่งออกเป็น “ชุดคำสั่งส่วนประกอบโปรแกรม” และ “โปรแกรมประยุกต์” ที่ใช้ชุดคำสั่งส่วนประกอบโปรแกรมข้างต้น ที่นำเอาผลลัพธ์ที่ต้องการนำมาประยุกต์ใช้ในงานหรือโปรแกรมที่ต้องการ

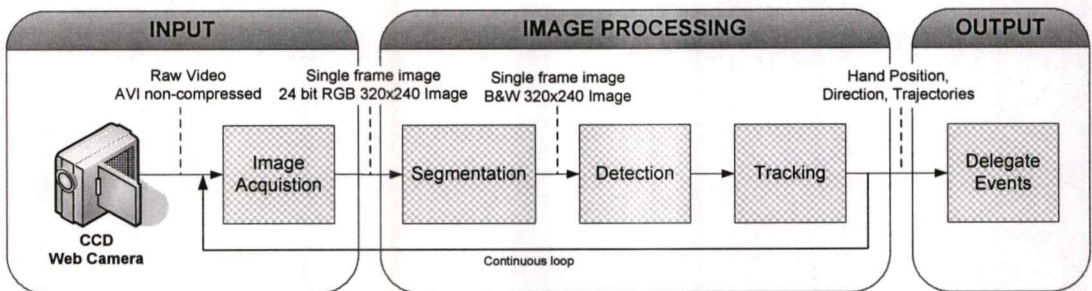
### 3.1 ชุดคำสั่งติดตามการเคลื่อนไหวของมือ (Hand Tracker Component)

การพัฒนาชุดคำสั่งสำหรับติดตามการเคลื่อนไหวของมือในเชิงส่วนประกอบ (Component) ซึ่งเมื่อทำการ Compile แล้วจะได้เป็น ไฟล์แบบ DLL (Dynamic Link Library) ที่จะถูกนำไปใช้เป็นส่วนประกอบหนึ่งใน โปรแกรมประยุกต์ที่ต้องการ โดยชุดคำสั่งส่วนประกอบสำหรับติดตามการเคลื่อนไหวของมือมีโครงสร้างดังนี้

#### 3.1.1 โครงสร้างของชุดคำสั่ง

แบ่งส่วนการทำงานออกเป็น 3 ส่วนใหญ่ๆ ดังนี้

- 1) ส่วนนำเข้าข้อมูลคือ ส่วนการนำภาพการเคลื่อนที่ของมือเข้าสู่ระบบ (Image Acquisition) เป็นการรับภาพจากกล้องวิดีโอที่มีอุปกรณ์ CCD ในการแปลงภาพมาเป็นสัญญาณดิจิทัล
- 2) ส่วนการประมวลผลภาพคือ ส่วนการนำภาพที่ได้มาทำการแบ่งส่วนภาพ (Image Segmentation) ตรวจสอบหาตำแหน่งของมือ (Hand Detection) และติดตามการเคลื่อนไหวของมือ (Hand Tracking)
- 3) ส่วนการส่งออกผลลัพธ์ของรูปภาพ พิกัดตำแหน่งในระบบ 3 มิติ และ ทิศทางการเคลื่อนที่ของมือ ออกไปให้โปรแกรมประยุกต์โดยผ่านเหตุการณ์ของชุดคำสั่ง

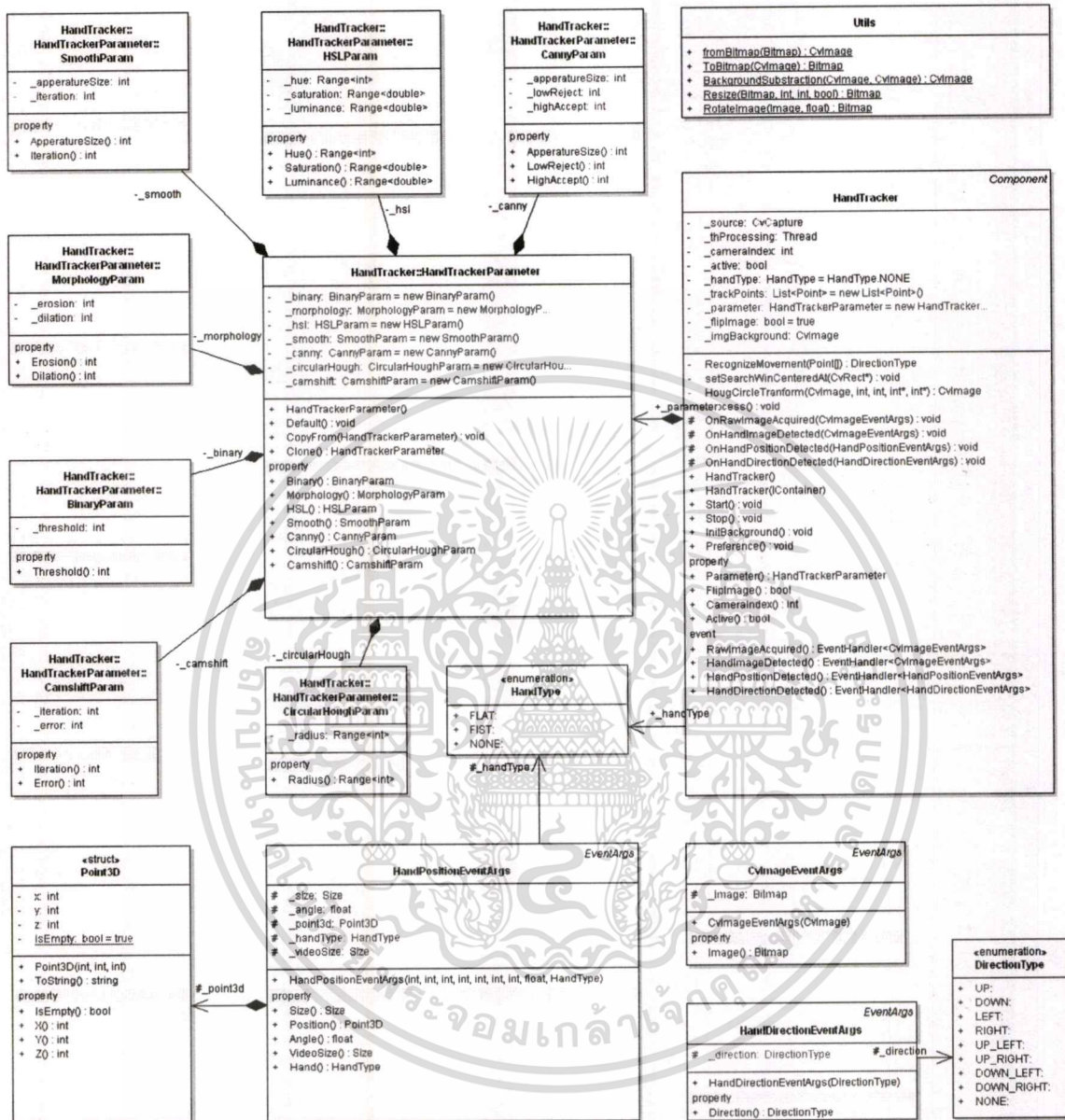


รูปที่ 3.1 แสดงโครงสร้างและกระบวนการทำงานของชุดคำสั่งส่วนประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า โปรแกรมในการติดตามการเคลื่อนไหวของมือ  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.2 คลาสไดอะแกรมของชุดคำสั่ง

แสดงกลุ่มข้อมูลโครงสร้างของคลาสภายในที่มีความสัมพันธ์กัน ดังแสดงได้ดังรูปที่ 3.2



รูปที่ 3.2 แสดงคลาสไดอะแกรมของชุดคำสั่ง

### 3.1.3 พจนานุกรมข้อมูลคลาส

จากคลาสไดอะแกรมที่ได้นั้นสามารถนำเสนอผ่านพจนานุกรมข้อมูลคลาสที่จะแสดงให้เห็นรายละเอียดของแอตทริบิวต์ต่างๆที่เกี่ยวข้องกับการทำงานของระบบ ซึ่งได้เป็นคลาสที่มีความสัมพันธ์กันทั้งหมด 16 คลาส ดังรายการตามตารางที่ 3.1

ตารางที่ 3.1 รายการคลาสของระบบติดตามการเคลื่อนไหวของมือ

ตารางที่	ชื่อคลาส	คำอธิบายคลาส
1	HandTracker	คลาสคอมโพเนนท์ในการทำงานของระบบการติดตามมือ
2	HandTrackerParameter	คลาสที่รวบรวมคลาสสำหรับตั้งค่าระบบติดตามมือ
3	BinaryParam	คลาสตั้งค่าการทำ Binary Threshold
4	SmoothParam	คลาสตั้งค่าการทำ Smoothing
5	MorphologyParam	คลาสตั้งค่าการปรับปรุงภาพด้วยวิธี Morphology
6	HSLParam	คลาสตั้งค่าการกรองภาพด้วย HSL Filtering
7	CannyParam	คลาสตั้งค่าการหาขอบภาพด้วยวิธี Canny Edge Detection
8	CircularHoughParam	คลาสตั้งค่าการแปลงฮัฟแบบวงกลม
9	CamshiftParam	คลาสตั้งค่าการติดตามด้วยวิธี CAMSHIFT Algorithm
10	CvImageEventArgs	คลาสเหตุการณ์สำหรับส่งภาพ
11	HandDirectionEventArgs	คลาสเหตุการณ์สำหรับส่งค่าทิศทางเคลื่อนที่
12	HandPositionEventArgs	คลาสเหตุการณ์สำหรับส่งค่าตำแหน่งพิกัดของมือ
13	DirectionType	ตัวแปรระบุชนิดของทิศทางเคลื่อนที่
14	HandType	ตัวแปรระบุชนิดของท่าทางมือ
15	Point3D	โครงสร้างข้อมูลตำแหน่งในระบบ 3 มิติ
16	Utils	คลาสเครื่องมือในการทำงานของระบบ

จากคลาสของระบบทั้ง 6 คลาสข้างต้นนั้น เมื่อมากำหนดคุณสมบัติต่างๆของแต่ละคลาสได้แก่ ชื่อและคำอธิบายแอททริบิวต์ ชนิดของข้อมูล ขนาดข้อมูล และการอ้างอิงข้อมูลไปยังคลาสที่มีความสัมพันธ์กัน เพื่อนำข้อมูลเหล่านี้ไปพัฒนาเป็นโปรแกรมใช้งานของระบบ โดยเราจะอธิบายรายละเอียดคุณสมบัติของคลาสไว้ที่พจนานุกรมข้อมูลคลาสดังรายละเอียดในตารางที่ 3.2 ถึงตารางที่ 3.16 ดังนี้

ตารางที่ 3.2 Component: HandTracker

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คลาสอ้างอิง
_source	โมดูลสำหรับจับภาพ	CvCapture	
_thProcessing		thread	
_handType	ชนิดของมือ	HandType	<b>HandType</b>
_trackPoints		List<Point>	
_imgBackground		CvImage	

ตารางที่ 3.2 (ต่อ)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คลาสอ้างอิง
Parameter	โมดูลสำหรับตั้งค่า	HandTrackerParameter	<b>HandTrackerParameter</b>
CameraIndex	หมายเลขอุปกรณ์กล้อง	integer	
FlipImage	การกลับข้างภาพ	boolean	
Active	สถานะการทำงาน	boolean	
_source	โมดูลสำหรับจับภาพ	CvCapture	

ตารางที่ 3.3 Class: HandTrackerParameter

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คลาสอ้างอิง
Binary	ตัวแปรตั้งค่า Binary Threshold	BinaryParam	<b>BinaryParam</b>
Morphology	ตัวแปรตั้งค่า Morphology	MorphologyParam	<b>MorphologyParam</b>
Hsl	ตัวแปรตั้งค่าสี HSL	HSLParam	<b>HSLParam</b>
Smooth	ตัวแปรตั้งค่า Smoothing	SmoothParam	<b>SmoothParam</b>
Canny	ตัวแปรตั้งค่า Canny Edge	CannyParam	<b>CannyParam</b>
CircularHough	ตัวแปรตั้งค่า Circular Hough	CircularHoughParam	<b>CircularHoughParam</b>
Camshift	ตัวแปรตั้งค่า CAMSHIFT	CamshiftParam	<b>CamshiftParam</b>

ตารางที่ 3.4 Class: BinaryParam

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คลาสอ้างอิง
Threshold	ค่าในการแปลงภาพเป็นไบนารี	integer	

ตารางที่ 3.5 Class: MorphologyParam

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คลาสอ้างอิง
Erosion	ค่าจำนวนรอบการทำ Erosion	integer	
Dilation	ค่าจำนวนรอบการทำ Dilation	integer	

ตารางที่ 3.6 Class: HSLParam

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คลาสอ้างอิง
Hue	ค่าสี Hue ต่ำสุด และสูงสุด ในระบบ HSL	Range<int>	
Saturation	ค่าสี Saturation ต่ำสุด และสูงสุด ในระบบ HSL	Range<double>	ใช้ประโยชน์ด้านการค้า
Luminance	ค่าสี Luminance ต่ำสุด และสูงสุด ในระบบ HSL	Range<double>	กรณีที่มีการนำไปใช้

ตารางที่ 3.7 Class: SmoothParam

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คลาสอ้างอิง
ApperatureSize	ขนาดของเมตริก	integer	
Iteration	จำนวนการทำซ้ำ	integer	

ตารางที่ 3.8 Class: CannyParam

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คลาสอ้างอิง
ApperatureSize	ขนาดของเมตริก	integer	
LowReject	ค่าต่ำสุดในการปฏิเสธ	integer	
HighAccept	ค่าสูงสุดในการยอมรับ	integer	

ตารางที่ 3.9 Class: CircularHoughParam

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คลาสอ้างอิง
Radius	ค่ารัศมีต่ำสุด และสูงสุด	Range<integer>	

ตารางที่ 3.10 Class: CamshiftParam

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คลาสอ้างอิง
Iteration	จำนวนการทำซ้ำ	integer	
Error	ค่าเปอร์เซ็นต์ความผิดพลาด	integer	

ตารางที่ 3.11 Class: CvImageEventArgs

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คลาสอ้างอิง
Image	ภาพ	Bitmap	

ตารางที่ 3.12 Class: HandPositionEventArgs

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คลาสอ้างอิง
Size	ขนาดของภาพมือ (กว้าง x สูง)	Size	
VideoSize	ขนาดของภาพวิดีโอ (กว้าง x สูง)	Size	
Angle	มุมการหมุนของมือ	float	
Position	ตำแหน่งพิกัดมือในระบบ 3 มิติ	Point3D	<b>Point3D</b>
Hand	ลักษณะท่าทางมือ	HandType	<b>HandType</b>

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.13 Class: HandDirectionEventArgs

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คลาสอ้างอิง
Direction	ทิศทาง การเคลื่อนที่	DirectionType	DirectionType

ตารางที่ 3.14 Enumeration: DirectionType

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คลาสอ้างอิง
UP	ขึ้น		
DOWN	ลง		
LEFT	ซ้าย		
RIGHT	ขวา		
UP_LEFT	เฉียงขึ้นซ้าย		
UP_RIGHT	เฉียงขึ้นขวา		
DOWN_LEFT	เฉียงลงซ้าย		
DOWN_RIGHT	เฉียงลงขวา		
NONE	ระบุทิศทางไม่ได้		

ตารางที่ 3.15 Enumeration: HandType

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คลาสอ้างอิง
FLAT	ลักษณะมือแบบ “แบนมือ”		
FIST	ลักษณะมือแบบ “กำมือ”		
NONE	ไม่มีลักษณะมือ		

ตารางที่ 3.16 Structure: Point3D

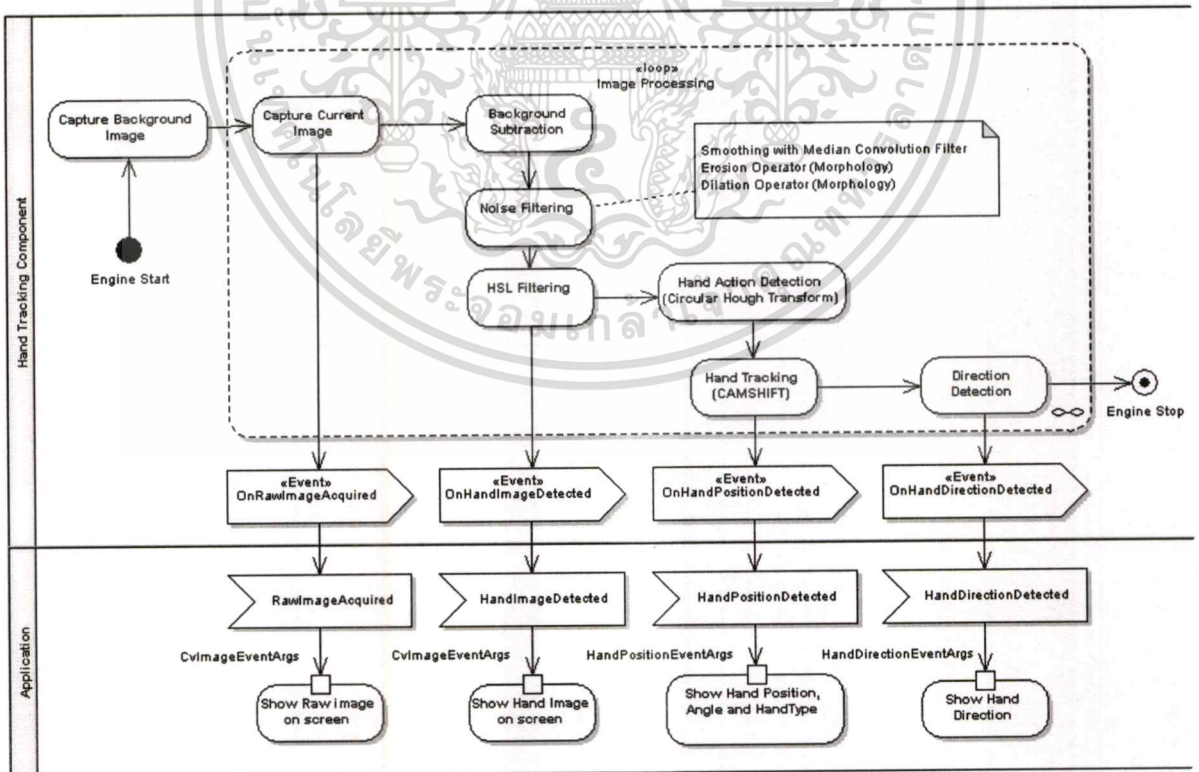
ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คลาสอ้างอิง
X	ค่าพิกัดในแนวแกน X	integer	
Y	ค่าพิกัดในแนวแกน Y	integer	
Z	ค่าพิกัดในแนวแกน Z	integer	
IsEmpty	สถานะข้อมูล	boolean	

### 3.1.4 Activity Diagram of Hand Tracking Component

ไดอะแกรมแสดงการทำงานของชุดคำสั่ง โดยเมื่อเริ่มการทำงานชุดคำสั่งจะทำการจับภาพพื้นหลังไว้ก่อน หลังจากนั้นจึงเริ่มเข้าสู่การวนลูปรการทำงาน โดยภายในลูปรการทำงานจะทำการจับภาพใหม่อยู่ตลอดเวลาแล้วนำมาเข้าสู่กระบวนการลบฉากหลัง การกรองสัญญาณรบกวน และการทำกรองภาพ HSL เพื่อหาภาพมือที่ต้องการ โดยระหว่างนั้นจะมีการส่งเหตุการณ์ไปด้วยโดยจะส่งภาพจริง และภาพมือที่ผ่านการตรวจจับผ่านเหตุการณ์ OnRawImageAcquired และ OnHandImageAcquired โดยส่งข้อมูลรูปภาพออกไป

ในขั้นต่อไปจะเป็นการตรวจจับลักษณะการแสดงของมือว่าเป็นการกำมือหรือแบมือโดยใช้อัลกอริทึมการแปลงฮัฟแบบวงกลมในการตรวจสอบ และทำการตรวจจับหาตำแหน่งของมือโดยใช้อัลกอริทึม CAMSHIFT เพื่อให้ได้ตำแหน่งพิกัดของมือ แล้วทำการส่งออกไปยังเหตุการณ์ OnHandPositionDetected โดยส่งข้อมูล ลักษณะท่าทางมือและพิกัดของมือ

สุดท้ายจะทำการตรวจจับทิศทางการเคลื่อนที่ของมือ โดยเมื่อเริ่มตรวจสอบตำแหน่งมือได้จะมีการเก็บรายการพิกัดของมือไว้เป็นลำดับ แล้วจึงนำเข้าอัลกอริทึมในการวิเคราะห์การทิศทางการเคลื่อนที่แล้วจึงส่งออกไปยังเหตุการณ์ OnHandDirectionDetected โดยส่งข้อมูล ทิศทางการเคลื่อนที่ของมือ 8 ทิศทาง

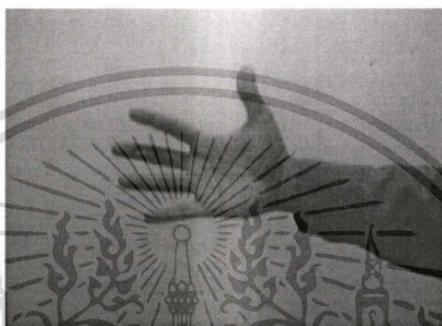


รูปที่ 3.3 Activity Diagram of Hand Tracking Component

### 3.1.5 กระบวนการทำงานของชุดคำสั่ง

#### 1) ส่วนนำเข้าข้อมูล

ข้อมูลภาพที่ได้จากกล้องวิดีโอจะอยู่ในรูปแบบของ Stream ของ AVI ที่เป็นแบบ non-compressed ดังนั้นในขั้นตอนนี้จะเป็นการเตรียมข้อมูลภาพ ให้อยู่ในรูปแบบที่ต้องการก่อนที่จะส่งต่อไปยังขั้นตอนต่อไป โดยทำการถ่ายภาพ หรือจับภาพ (Image Grabber) จากกล้องวิดีโอ ใน ช่วงเวลาที่ต้องการ แล้วทำการแปลงให้อยู่ในรูปแบบของระบบสี RGB ขนาด 24 bit ที่ความละเอียด 320x240 Pixels ดังรูปที่ 3.4



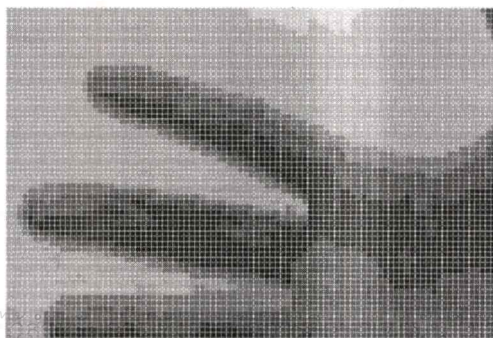
รูปที่ 3.4 แสดงภาพมือที่ได้จากกล้องขนาด 320x240 Pixels

ในการจับภาพจากกล้องวิดีโอจะใช้ฟังก์ชัน QueryFrame () ของคลาส CvCapture จาก OpenCV ในการจับภาพโดยเขียนคำสั่งได้ดังนี้

```
//Create instant of CvCapture, 0 as CameraIndex
CvCapture source = new CvCapture(0);

//Call QueryFrame() function for capture image frame
CvImage frame = source.QueryFrame().Copy();
```

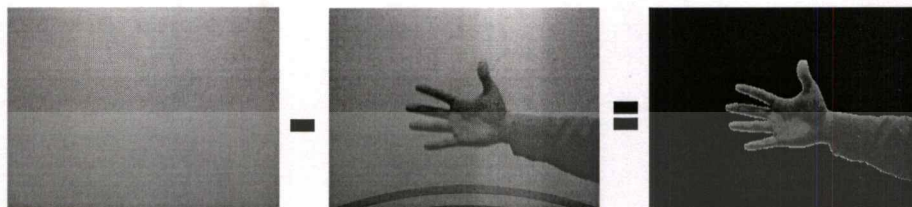
การเก็บข้อมูลสีในแต่ละจุดภาพ เรียกว่า Pixel เรียงกันในลักษณะแถว (Row) และคอลัมน์ (Column) ตามขนาดของภาพ (Image Aspect Ration) โดยภาพมีขนาดที่ได้มีขนาด 24 bit และความละเอียด 320 x 240 ก็จะมีจำนวนจุดภาพทั้งหมด 76,800 Pixel โดยแต่ละจุดภาพจะเก็บค่าสี RGB ไว้ สีละ 8 bit รวมจุดภาพละ 3 byte



รูปที่ 3.5 แสดงจุดภาพแต่ละจุดภายในภาพในแนวแถวและแนวคอลัมน์

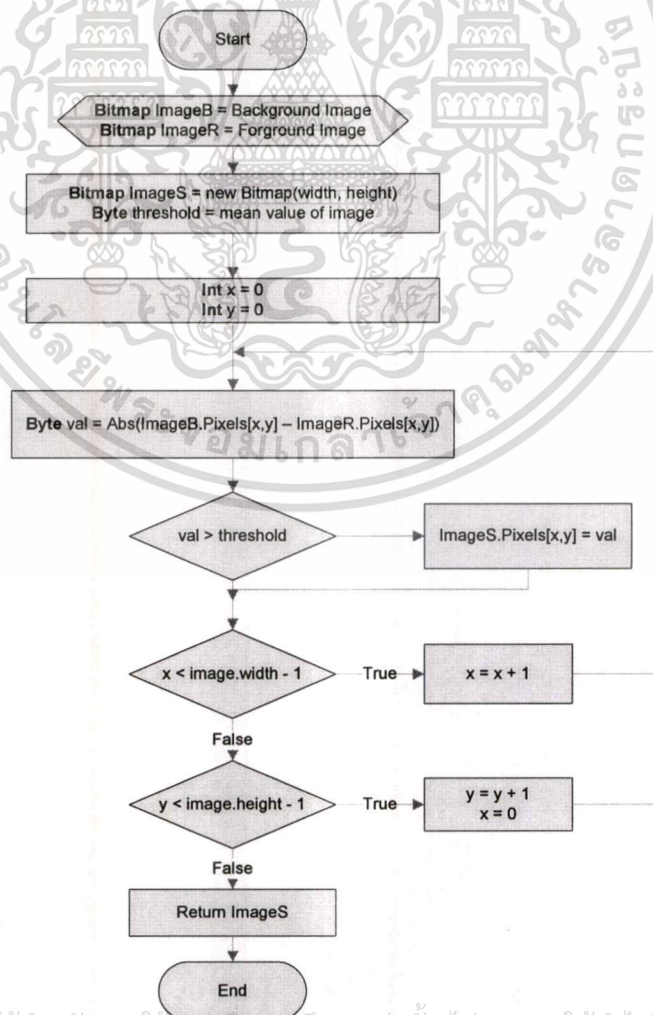
## 2) ส่วนการประมวลผลภาพ

เป็นขั้นตอนการเตรียมภาพให้อยู่ในรูปแบบที่เหมาะสมสำหรับนำไปใช้ขั้นตอนถัดไป โดยขั้นตอนแรกเป็นขั้นตอนการลบฉากหลังที่ไม่ต้องการให้เหลือแต่ส่วนของมือที่ต้องการ (Background Subtraction) โดยทำการหาค่าต่างของสีระหว่างภาพฉากหลังและภาพจริง แล้วทำการกรองภาพด้วยให้ Smooth ด้วยตัวกรอง Median ดังรูปที่ 3.6



รูปที่ 3.6 แสดงภาพมือที่ได้จากการลบฉากหลัง

โดยมีเงื่อนไขว่าภาพฉากหลังและภาพจริงต้องมีขนาดเท่ากัน ซึ่งจะสามารถเขียนอัลกอริทึมการทำ Background Subtraction เป็นแผนภาพได้ดังรูปที่ 3.7



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกรูปที่ 3.7 แสดงแผนภาพการหาค่าความแตกต่างของภาพทุกครั้งที่มีการนำไปใช้

ในการทำ Background Subtraction สามารถใช้ฟังก์ชัน AbsDiff() ของคลาส CvImage จาก OpenCV ในการหาภาพที่แตกต่างกันได้โดยเขียนคำสั่งได้ดังนี้

```
//Prepare background image by capture image on first time
CvImage imgBackground = source.QueryFrame().Copy()
```

```
//Declare forground image
CvImage imgRaw = source.QueryFrame().Copy();

//Find Absolute difference of foreground image from background image
CvImage imgDifference = imgBackground.AbsDiff(imgRaw);
```

เมื่อทำการหาค่าความต่างของภาพได้แล้วจึงทำการแปลงภาพให้เป็น Binary โดยการทำให้ Binary Threshold จะได้ภาพดังรูปที่ 3.8



รูปที่ 3.8 การแปลงภาพไปเป็น Binary

ในการทำ Binary Thresholding สามารถใช้ฟังก์ชัน Threshold() ของคลาส CvImage จาก OpenCV ในการหาภาพที่แตกต่างกันได้โดยเขียนคำสั่งได้ดังนี้

```
//Binary Tresholding by min value is 75 and max value is 255
CvImage imgBinary = imgSegmented.CvtToGrey().Threshold(75, 255,
ThresholdType.THRESH_BINARY);
```

หากภาพที่ได้มีสัญญาณรบกวนคิดมาด้วยจะต้องทำการลบสัญญาณรบกวนนั้นออกโดยใช้เทคนิค Noise Filtering ซึ่งในโครงงานนี้ใช้ Operator ในการจัดการ 3 ตัวคือ Median Convolution Filter (Smooth), Erosion Operator, Dilation Operator เพื่อทำการปรับแต่งภาพให้อยู่ในรูปแบบที่ต้องการโดยเขียนคำสั่งโดยใช้ฟังก์ชันของคลาส CvImage จาก OpenCV ได้ดังนี้

```
//Noise Filtering by Median Convolution Filter
CvImage imgSmooth = imgSegmented.Smooth(SmoothType.MEDIAN, 3, 10);

//Erosion Operator (Morphology) by iteration is 2 time
CvImage imgErode = imgSmooth.Erode(null, 2);

//Dilation Operator (Morphology) by iteration is 1 time
CvImage imgDilate = imgErode.Erode(null, 1);
```

หลังจากทำการปรับแต่งภาพให้เหมาะสมแล้ว จึงทำการ Masking เพื่อนำภาพจริงที่ต้องการออกมาแสดงโดยใช้ภาพ Binary ซึ่งสามารถเขียนคำสั่งโดยใช้ฟังก์ชันของคลาส CvImage จาก OpenCV ได้ดังนี้

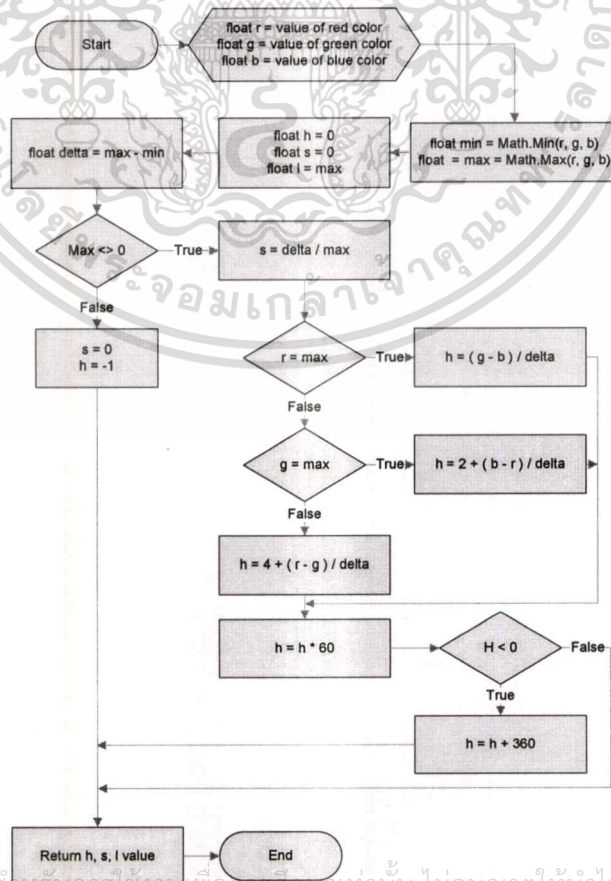
```
//Prepare empty image with same size of current image
CvImage imgSegmentedHand = imgRaw.Copy();
imgSegmentedHand.Zero();

//Masking by use funtion Copy(Destination Image, Mask Image)
imgRaw.Copy(imgSegmentedHand, imgSegmented);
```

หลังจากทำการลบฉากหลังออกแล้วภาพที่ได้ก็ยังไม่ดีพอที่จะนำไปใช้งานจึงต้องทำการกรองให้เหลือแต่ส่วนของมือที่ต้องการด้วย HSL Filtering โดยการกำหนดค่า Hue ให้มีค่าสีเฉพาะส่วนของสีมือที่ต้องการ ดังรูปที่ 3.9

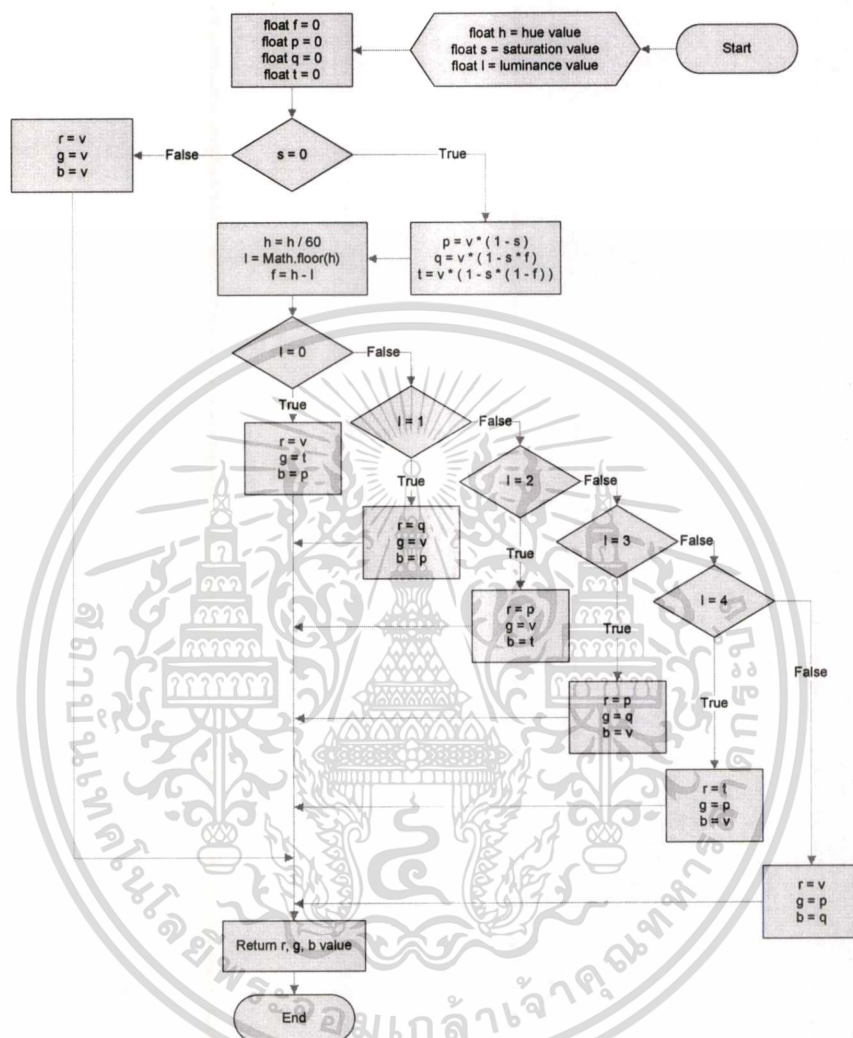


รูปที่ 3.9 แสดงภาพมือที่ผ่านการ HSL Filtering



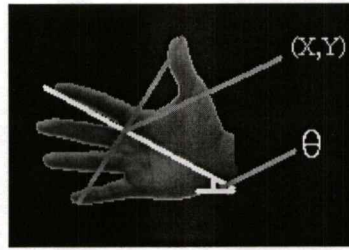
รูปที่ 3.10 แผนภาพการแปลงค่าสี RGB เป็น HSL

ในการแปลงค่าสี RGB ไปเป็น HSL สามารถเขียนอัลกอริทึมเป็นแผนภาพการทำงานได้ดังรูปที่ 3.10 และการแปลง HSL ไปเป็น RGB สามารถเขียนอัลกอริทึมเป็นแผนภาพการทำงานได้ดังรูปที่ 3.11 (Schaller, N. n.d.)



รูปที่ 3.11 แผนภาพการแปลงค่าสี HSL เป็น RGB

ในกระบวนการติดตามและหาตำแหน่งสามารถทำได้โดยใช้ CAMSHIF Algorithms โดยใช้ข้อมูลรูปภาพเฉพาะส่วนของมือที่ต้องการเท่านั้นมาประมวลผลเพื่อหาขนาดความกว้างและขนาดความสูงของมือ จะทำให้สามารถหาจุดกึ่งกลางมือได้ และมุมการหมุนของมือนั้นดังรูปที่ 3.12 ในส่วนของแกน Z จะใช้เป็นค่าอนุমানจากขนาดของความกว้างมือโดยมีค่าสูงขึ้นเมื่ออยู่ไกลจากกล้อง และมีค่าลดลงเมื่อเข้าใกล้กล้อง



รูปที่ 3.12 แสดงภาพการหาตำแหน่งและมุมการหมุนจากภาพมือ

ในการติดตามหาตำแหน่งของมือโดยใช้ CAMSHIFT Algorithm สามารถใช้คลาสที่เตรียมไว้แล้วใน OpenCV โดยขั้นตอนแรกต้องทำการประกาศตัวแปรที่จำเป็นต้องใช้ในการติดตามดังนี้

```
//Prepare search windows
CvRect search_window = new CvRect(0, 0, source.Size.width,
                                   source.Size.height);

//Create environment object for tracking
int[] tracker_histogram_bins = { 16, 16, 16 };
float[] [] tracker_histogram_ranges =
    { new float[] { 0, 255 }, new float[] { 0, 255 },
      new float[] { 0, 255 } };

CvHistogram tracker_target = new CvHistogram(tracker_histogram_bins,
                                             tracker_histogram_ranges, true);

//Create Criteria 100 Iterations || 10% Error
CvTermCriteria search_criteria = new CvTermCriteria(100, 10);
CvConnectedComp comp = new CvConnectedComp();
CvBox2D box = new CvBox2D();
CvImage backProjection = new CvImage(imgBackground.Size,
                                     BitDepths.IPL_DEPTH_8U, 1);
```

หลังจากนั้นใน Loop การทำงานจะสามารถติดตามการเคลื่อนไหวของมือโดยนำเข้าข้อมูลภาพที่ผ่านการลบจากหลัง การปรับแต่งภาพ และอยู่ในรูปแบบ Binary เข้าสู่ฟังก์ชันโดยเขียนคำสั่งได้ดังนี้

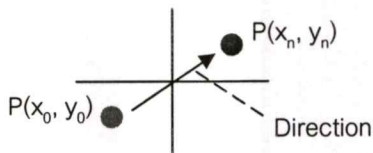
```
//Prepare search windows
tracker_target.CamShift(backProjection, search_window, search_criteria,
                       ref comp, ref box);
```

หลังจากทำการเรียกฟังก์ชัน CamShift() แล้วจะได้ค่ากลับมาซึ่งสามารถนำมาใช้ในการระบุตำแหน่งได้โดยเขียนคำสั่งได้ดังนี้

```
//Region of detected hand image and angle
float x = box.center.x;
float y = box.center.y;
float angle = box.angle;
float width = box.size.width;
float height = box.size.height;
```

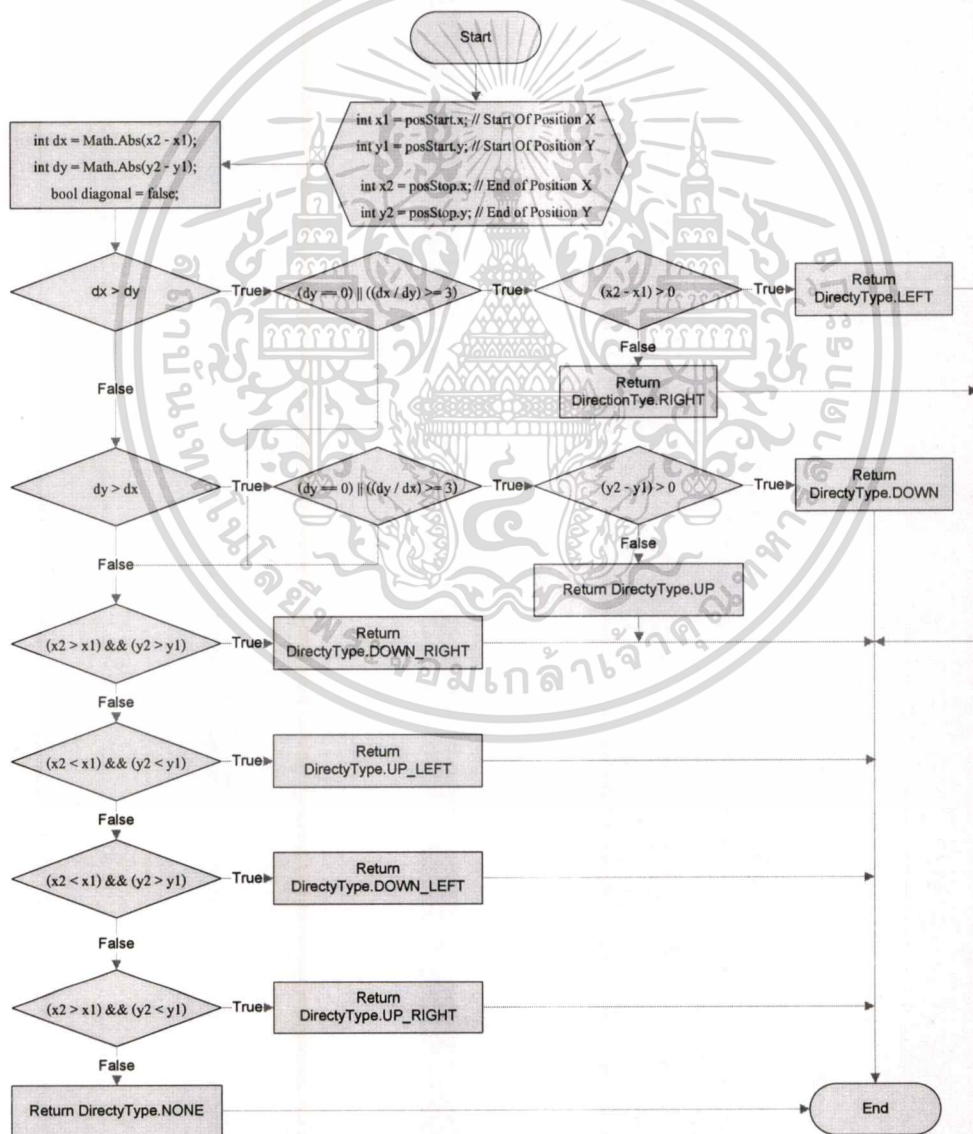
ในกระบวนการหาทิศทางของการเคลื่อนที่ของมือจะใช้ค่าพิกัด  $P(x, y)$  จากเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า กระบวนการติดตามข้างต้นนำมา เก็บบันทึกตามช่วงเวลา แล้วนำค่าดังกล่าวมา ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประมวลผล โดยค่าที่ถูกเก็บบันทึกจะเป็นลำดับของจุดพิกัด  $P(x, y)$  ตามเวลา  $t$  โดยทำการดูที่  $P_{t=0}$  เทียบกับ  $P_{t=n}$



รูปที่ 3.13 แสดงภาพทิศทางการเคลื่อนที่ของจุด

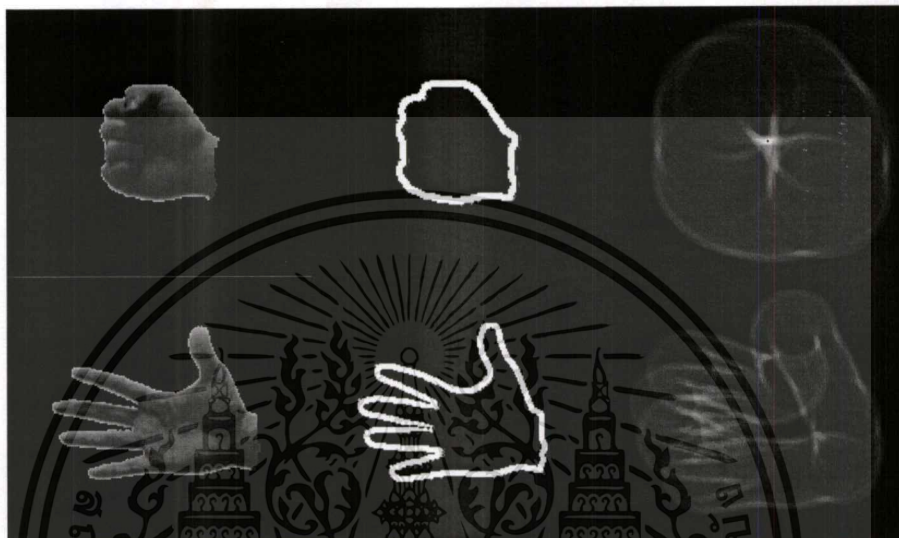
จากกระบวนการหาทิศทางของการเคลื่อนที่ของมือของพิกัด  $P(x, y)$  สามารถเขียนอัลกอริทึมในการหาทิศทางได้ดังรูปที่ 3.14



รูปที่ 3.14 แสดงแผนภาพการหาทิศทางการเคลื่อนที่จากจุด 2 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจจับการกำมือเป็นอีกขั้นตอนหนึ่งที่สามารถหาลักษณะของท่ามือได้ (Gesture Recognition) อย่างง่าย โดยใช้ตัวแปรงฮัฟแบบวงกลม เนื่องจากมือเมื่อทำการกำมือแล้วจะมีรูปทรงคล้ายวงกลมจึงสามารถใช้ตัวแปลงดังกล่าวในการหารูปทรงกลมและระบุได้ว่ารูปร่างมือในขณะนั้นมีการกำมืออยู่หรือไม่ ซึ่งภาพที่ได้จากการทำการแปรงฮัฟแบบวงกลมของลักษณะมือแต่ละแบบจะมีค่าต่างกัน ดังรูปที่ 3.15

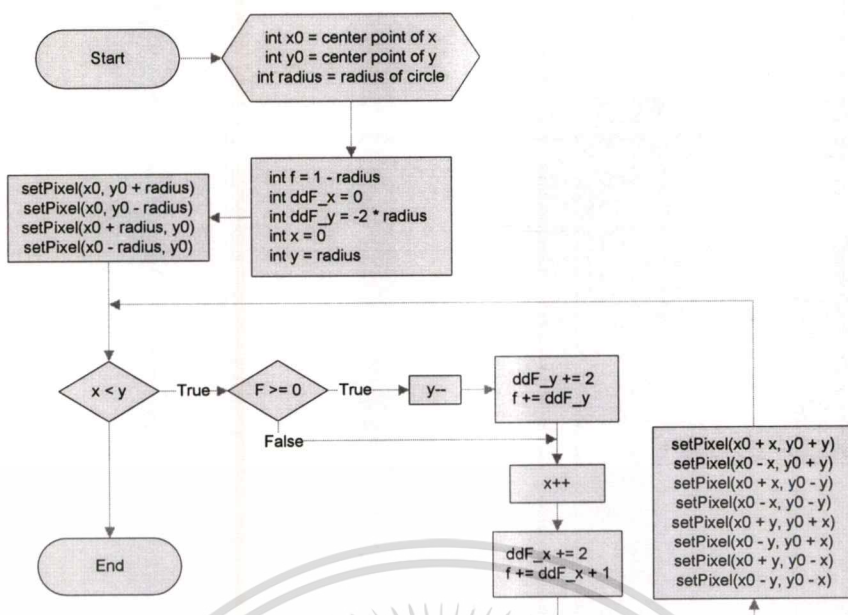


รูปที่ 3.15 แสดงภาพมือที่ผ่านการแปรงฮัฟแบบวงกลม

โดยทำกานำภาพมือที่ผ่านการแบ่งส่วนมาแล้วมาทำการหาขอบภาพโดยใช้วิธีการ Canny Edge Detection ซึ่งสามารถใช้ฟังก์ชัน Canny() ของคลาส CvImage จาก OpenCV ในการแปลงภาพ Binary เพื่อหาขอบของภาพ โดยเขียนคำสั่งได้ดังนี้

```
//Region of detected hand image and angle
//LowReject is 60
//HighAccept is 200
//AppertureSize is 3
CvImage imgEdge = imgSegmentedHand.Canny(60, 200, 3);
```

หลังจากทำการหาขอบภาพมาเรียบร้อยแล้วจะนำเข้าสู่กระบวนการแปลงฮัฟแบบวงกลมเพื่อหารูปทรงที่มีลักษณะเป็นวงกลม โดยอนุมานว่าภาพมือที่อยู่ในลักษณะกำมือจะมีรูปทรงคล้ายวงกลม โดยหลักการแปลงฮัฟแบบวงกลมนั้นจะทำการวาดเส้นวงกลมโดยกำหนดจุดศูนย์กลางตามตำแหน่งจุดของเส้นขอบของภาพมือข้างต้น และกำหนดรัศมีของเส้นวงกลม โดยอัลกอริธึมในการวาดเส้นวงกลมนั้นสามารถเขียนเป็นแผนภาพได้ดังรูปที่ 3.16



รูปที่ 3.16 แผนภาพการวาดเส้นวงกลม โดยกำหนดจุดกึ่งกลาง และรัศมี

เมื่อทำการวาดเส้นวงกลมโดยใช้จุดศูนย์กลางวงกลมอยู่ที่เส้นของขอบภาพมือข้างต้นโดยค่าจุดสีของเส้นรอบวงที่ทำการวาดลงไปให้ทำการเพิ่มค่าที่ละหนึ่งค่าในแต่ละเส้นวงกลม ทำให้เกิดลายเส้นซึ่งเรียกว่าภาพฮัฟ โดยถ้าเส้นขอบภาพมือข้างต้นมีลักษณะคล้ายวงกลม ภาพฮัฟที่ได้จะมีจุดที่มีความเข้มสูงซึ่งอยู่ที่จุดกึ่งกลางของภาพมือ และหากเส้นขอบภาพมือข้างต้นไม่มีลักษณะคล้ายวงกลม ภาพฮัฟที่ได้จะมีความเข้มของจุดสีกระจายกันไป ดังนั้นจึงนำภาพฮัฟที่ได้นำไป Threshold เพื่อหาจุดภาพที่มีความเข้มสูงตามที่กำหนดไว้ซึ่งอนุมานว่าภาพมือที่ได้เป็นภาพการกำมือก็จะมีค่าความเข้มสูงกว่าค่า Threshold ที่กำหนด แต่หากภาพมือเป็นการแบมือก็จะมีค่าความเข้มกระจายกันไปซึ่งมีค่าต่ำกว่าค่า Threshold นั้นเอง

### 3) ส่วนการส่งออกผลลัพธ์

จากการประมวลผลภาพทำให้สามารถหาค่าพิกัด 4 Degree of Freedom ได้ ทิศทางการเคลื่อนที่ของมือ และการตรวจจับการกำมือ ข้อมูลต่างๆ เหล่านี้จะถูกส่งออกผ่าน พารามิเตอร์ ของเหตุการณ์ที่เกิดขึ้นภายใน Hand Tracker Component โดยข้อมูลที่ได้นั้นประกอบไปด้วย

- I. ตำแหน่งพิกัดจุดกึ่งกลางมือในแนวแกน X, Y และ Z
- II. ขนาดความกว้าง (Width) และความสูง (Height) ของมือที่ตรวจจับ
- III. มุมการหมุนของมือ (Angle)
- IV. ลักษณะมือที่แสดงออก เช่น กำมือ (Fist) หรือ แบนมือ (Flat)
- V. ทิศทางการเคลื่อนที่ของมือ (Trajectories) 8 ทิศทาง

### 3.2 โปรแกรมประยุกต์

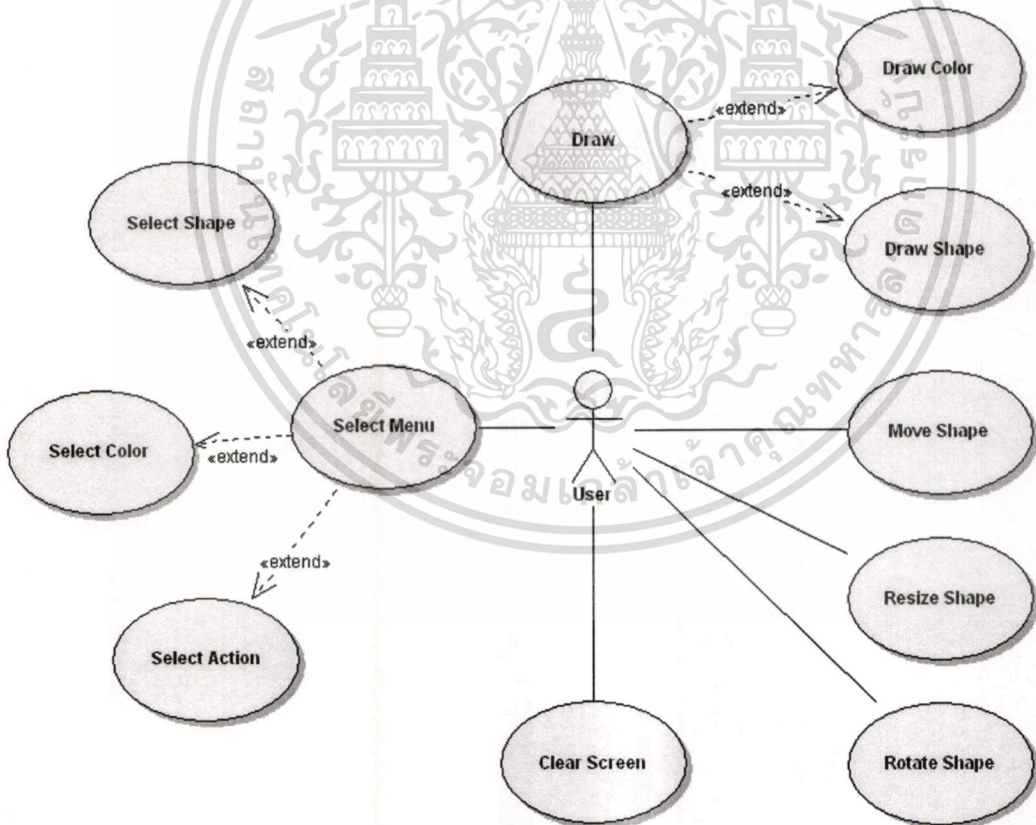
โปรแกรมประยุกต์ที่ออกแบบจะเป็นโปรแกรมสำหรับวาดรูปโดยใช้มือในการควบคุม หรือ วาดรูปตามรูปแบบที่ต้องการ โดยใช้ข้อมูลนำเข้าจาก ชุดคำสั่งติดตามการเคลื่อนไหวของมือ

#### 3.2.1 โครงสร้างของโปรแกรม

แบ่งส่วนการทำงานออกเป็น 3 ส่วนใหญ่ๆ ดังนี้

- 1) ส่วนการควบคุม คือ การใช้ชุดคำสั่งติดตามการเคลื่อนไหวของมือ (Hand Tracker Component) เป็นข้อมูลนำเข้าในระบบชี้ตำแหน่งในการเลือก อุปกรณ์หรือวาดรูปลงบนหน้าจอ
- 2) ส่วนการวาดรูป คือ ส่วนการวาดรูปหรือแสดงผลตามคำสั่งที่ได้รับจากส่วน ควบคุมในการชี้ตำแหน่ง และการกำมือในการเลือก

#### 3.2.2 ยูสเคสของโปรแกรมประยุกต์



รูปที่ 3.17 USE CASE การทำงานของโปรแกรมประยุกต์ตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.17 รายละเอียดยูสเคส Select Menu

ชื่อยูสเคส	Select Menu
คำอธิบาย	การเลือกเมนูการทำงาน
เหตุการณ์ที่กระตุ้นการทำงาน	เมื่อผู้ใช้ทำการก้ำมือ ณ บริเวณที่กำหนด
คำอธิบายยูสเคส	เป็นขั้นตอนการเลือกเมนูการทำงาน
แอกเตอร์	User
ยูสเคสที่เกี่ยวข้อง	Select Shape, Select Color, Select Action
ผู้เกี่ยวข้องอื่น	-
เงื่อนไขเริ่มต้น	-
เงื่อนไขเมื่อสำเร็จ	-
การทำงาน	1. ผู้ใช้เคลื่อนที่ตำแหน่งของมือในลักษณะแบมือไปยังบริเวณที่กำหนด 2. ผู้ใช้ทำการก้ำมือเพื่อทำการเลือก
เงื่อนไขการทำงาน	หากตำแหน่งมือของผู้ใช้ไม่ได้อยู่ในบริเวณที่กำหนด จะเลือกเมนูได้

ตารางที่ 3.18 รายละเอียดยูสเคส Select Shape

ชื่อยูสเคส	Select Shape
คำอธิบาย	การเลือกเมนูสำหรับการวาดวัตถุรูปทรง
เหตุการณ์ที่กระตุ้นการทำงาน	เมื่อผู้ใช้ทำการก้ำมือ ณ บริเวณที่กำหนด
คำอธิบายยูสเคส	เป็นขั้นตอนการเลือกเมนูสำหรับการวาดวัตถุรูปทรง
แอกเตอร์	User
ยูสเคสที่เกี่ยวข้อง	Select Menu
ผู้เกี่ยวข้องอื่น	-
เงื่อนไขเริ่มต้น	-
เงื่อนไขเมื่อสำเร็จ	พร้อมสำหรับการวาดวัตถุรูปทรงที่เลือก (ยูสเคส Draw Shape)
การทำงาน	1. ผู้ใช้เคลื่อนที่ตำแหน่งของมือในลักษณะแบมือไปยังบริเวณที่กำหนด 2. ผู้ใช้ทำการก้ำมือเพื่อทำการเลือก
เงื่อนไขการทำงาน	หากตำแหน่งมือของผู้ใช้ไม่ได้อยู่ในบริเวณที่กำหนด จะไม่สามารถเลือกเมนูรูปทรงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.19 รายละเอียดยูสเคส Select Color

ชื่อยูสเคส	Select Color
คำอธิบาย	การเลือกเมนูสำหรับการวาดจุดสี
เหตุการณ์ที่กระตุ้นการทำงาน	เมื่อผู้ใช้ทำการกำมือ ณ บริเวณที่กำหนด
คำอธิบายยูสเคส	เป็นขั้นตอนการเลือกเมนูสำหรับการวาดจุดสี
แอกเตอร์	User
ยูสเคสที่เกี่ยวข้อง	Select Menu
ผู้เกี่ยวข้องอื่น	-
เงื่อนไขเริ่มต้น	-
เงื่อนไขเมื่อสำเร็จ	พร้อมสำหรับการวาดจุดสีที่เลือก (ยูสเคส Draw Color)
การทำงาน	<ol style="list-style-type: none"> <li>1. ผู้ใช้เคลื่อนที่ตำแหน่งของมือในลักษณะแบมือไปยังบริเวณที่กำหนด</li> <li>2. ผู้ใช้ทำการกำมือเพื่อทำการเลือก</li> </ol>
เงื่อนไขการทำงาน	หากตำแหน่งมือของผู้ใช้ไม่ได้อยู่ในบริเวณที่กำหนด จะไม่สามารถเลือกเมนูสีได้

ตารางที่ 3.20 รายละเอียดยูสเคส Select Action

ชื่อยูสเคส	Select Action
คำอธิบาย	การเลือกเมนูสำหรับการกระทำกับวัตถุ
เหตุการณ์ที่กระตุ้นการทำงาน	เมื่อผู้ใช้ทำการกำมือ ณ บริเวณที่กำหนด
คำอธิบายยูสเคส	เป็นขั้นตอนการเลือกเมนูสำหรับการกระทำกับวัตถุ
แอกเตอร์	User
ยูสเคสที่เกี่ยวข้อง	Select Menu
ผู้เกี่ยวข้องอื่น	-
เงื่อนไขเริ่มต้น	-
เงื่อนไขเมื่อสำเร็จ	พร้อมสำหรับเคลื่อนย้าย หมุน ย่อ/ขยายวัตถุรูปทรง (Move, Rotate, Resize)
การทำงาน	<ol style="list-style-type: none"> <li>1. ผู้ใช้เคลื่อนที่ตำแหน่งของมือในลักษณะแบมือไปยังบริเวณที่กำหนด</li> <li>2. ผู้ใช้ทำการกำมือเพื่อทำการเลือก</li> </ol>
เงื่อนไขการทำงาน	หากตำแหน่งมือของผู้ใช้ไม่ได้อยู่ในบริเวณที่กำหนด จะไม่สามารถเลือกเมนูการกระทำได้

ตารางที่ 3.21 รายละเอียดยูสเคส Draw

ชื่อยูสเคส	Draw
คำอธิบาย	การวาด
เหตุการณ์ที่กระตุ้นการทำงาน	เมื่อผู้ใช้ทำการกำมือ
คำอธิบายยูสเคส	เป็นขั้นตอนการวาด
แอกเตอร์	User
ยูสเคสที่เกี่ยวข้อง	Draw Shape, Draw Color
ผู้เกี่ยวข้องอื่น	-
เงื่อนไขเริ่มต้น	-
เงื่อนไขเมื่อสำเร็จ	-
การทำงาน	<ol style="list-style-type: none"> <li>1. ผู้ใช้เคลื่อนที่ตำแหน่งของมือในลักษณะแบมือไปยังบริเวณที่ต้องการ</li> <li>2. ผู้ใช้ทำการกำมือเพื่อทำการวาด</li> <li>3. ผู้ใช้ทำการแบมือเพื่อทำการหยุดวาด</li> </ol>
เงื่อนไขการทำงาน	-

ตารางที่ 3.22 รายละเอียดยูสเคส Draw Color

ชื่อยูสเคส	Draw Color
คำอธิบาย	การวาดจุดสี
เหตุการณ์ที่กระตุ้นการทำงาน	เมื่อผู้ใช้ทำการกำมือแล้วเคลื่อนที่ตำแหน่งของมือ
คำอธิบายยูสเคส	เป็นขั้นตอนการวาดจุดสีที่เลือกไว้
แอกเตอร์	User
ยูสเคสที่เกี่ยวข้อง	Draw
เงื่อนไขเริ่มต้น	ผู้ใช้ทำการเลือกเงื่อนไขสีระบบจะเปลี่ยนการวาดเดิม (Draw) เป็นการวาดจุดสี (Draw Color)
การทำงาน	<ol style="list-style-type: none"> <li>4. ผู้ใช้เคลื่อนที่ตำแหน่งของมือในลักษณะแบมือไปยังบริเวณที่จะทำการวาดจุดสี</li> <li>5. ผู้ใช้ทำการกำมือเพื่อทำการวาดจุดสีโดยสามารถเคลื่อนที่ไปมาได้</li> <li>6. ผู้ใช้ทำการแบมือเพื่อทำการหยุดวาดจุดสี</li> </ol>
เงื่อนไขการทำงาน	หากผู้ใช้ไม่ได้ทำการเลือกสี หรืออยู่ในลักษณะการกระทำอื่นจะไม่สามารถวาดจุดสีได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตารางที่ 3.23 รายละเอียดยูสเคส Draw Shape

ชื่อยูสเคส	Draw Shape
คำอธิบาย	การวาดวัตถุรูปทรง
เหตุการณ์ที่กระตุ้นการทำงาน	เมื่อผู้ใช้ทำการกำมือ
คำอธิบายยูสเคส	เป็นขั้นตอนการวาดวัตถุรูปทรงที่เลือกไว้
แอกเตอร์	User
ยูสเคสที่เกี่ยวข้อง	Draw
ผู้เกี่ยวข้องอื่น	-
เงื่อนไขเริ่มต้น	ผู้ใช้ทำการเลือกเงื่อนไขรูปทรงที่ต้องการระบบจะเปลี่ยนการวาดเดิม (Draw) เป็นการวาดวัตถุรูปทรง (Draw Shape)
เงื่อนไขเมื่อสำเร็จ	เปลี่ยนเงื่อนไขไปอยู่ที่การเลือกการกระทำ (สำหรับการหมุน ย่อขยายวัตถุ)
การทำงาน	<ol style="list-style-type: none"> <li>1. ผู้ใช้เคลื่อนที่ตำแหน่งของมือ ในลักษณะแบมือไปยังบริเวณที่จะทำการวาดวัตถุรูปทรง</li> <li>2. ผู้ใช้ทำการกำมือเพื่อทำการวาดวัตถุรูปทรงที่ต้องการ</li> </ol>
เงื่อนไขการทำงาน	หากผู้ใช้ไม่ได้ทำการเลือกวัตถุรูปทรงไว้ หรืออยู่ในลักษณะการกระทำอื่น จะไม่สามารถวาดวัตถุรูปทรงได้

### ตารางที่ 3.24 รายละเอียดยูสเคส Move Shape

ชื่อยูสเคส	Move Shape
คำอธิบาย	การเคลื่อนย้ายตำแหน่งวัตถุรูปทรง
เหตุการณ์ที่กระตุ้นการทำงาน	เมื่อผู้ใช้ทำการกำมือ แล้วเคลื่อนที่
คำอธิบายยูสเคส	เป็นขั้นตอนการย้ายตำแหน่งวัตถุรูปทรงที่เลือกไว้
แอกเตอร์	User
ยูสเคสที่เกี่ยวข้อง	-
ผู้เกี่ยวข้องอื่น	-
เงื่อนไขเริ่มต้น	ผู้ใช้ทำการเลือกเงื่อนไขการกระทำ (Action) ระบบจะเปลี่ยนสถานะเป็นการกระทำกับวัตถุรูปทรง
เงื่อนไขเมื่อสำเร็จ	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตารางที่ 3.24 (ต่อ)

การทำงาน	<ol style="list-style-type: none"> <li>1. ผู้ใช้เคลื่อนที่ตำแหน่งของมือในลักษณะแบมือไปยังบริเวณวัตถุรูปทรงที่ต้องการ</li> <li>2. ผู้ใช้ทำการกำมือเพื่อทำการเลือกวัตถุรูปทรงที่จะทำการเคลื่อนย้าย</li> <li>3. ผู้ใช้ทำการเคลื่อนที่ตำแหน่งของมือไปยังบริเวณที่ต้องการ</li> <li>4. ผู้ใช้ทำการปล่อยมือเพื่อยกเลิกการเคลื่อนย้าย</li> </ol>
เงื่อนไขการทำงาน	หากผู้ใช้ไม่ได้ทำการเลือกการกระทำ (Action) ไว้จะไม่สามารถทำการเคลื่อนย้ายวัตถุรูปทรงได้

## ตารางที่ 3.25 รายละเอียดยูสเคส Rotate Shape

ชื่อยูสเคส	Rotate Shape
คำอธิบาย	การหมุนวัตถุรูปทรง
เหตุการณ์ที่กระตุ้นการทำงาน	เมื่อผู้ใช้ทำการกำมือ แล้วหมุนมือ
คำอธิบายยูสเคส	เป็นขั้นตอนการหมุนวัตถุรูปทรงที่เลือกไว้
แอกเตอร์	User
ยูสเคสที่เกี่ยวข้อง	-
ผู้เกี่ยวข้องอื่น	-
เงื่อนไขเริ่มต้น	ผู้ใช้ทำการเลือกเงื่อนไขการทำงาน (Action) ระบบจะเปลี่ยนสถานะเป็นการกระทำกับวัตถุรูปทรง
เงื่อนไขเมื่อสำเร็จ	-
การทำงาน	<ol style="list-style-type: none"> <li>1. ผู้ใช้เคลื่อนที่ตำแหน่งของมือในลักษณะแบมือไปยังบริเวณวัตถุรูปทรงที่ต้องการ</li> <li>2. ผู้ใช้ทำการกำมือเพื่อทำการเลือกวัตถุรูปทรงที่จะทำการหมุน</li> <li>3. ผู้ใช้ทำการหมุนลักษณะของมือตามต้องการ</li> <li>4. ผู้ใช้ทำการปล่อยมือเพื่อยกเลิกการหมุน</li> </ol>
เงื่อนไขการทำงาน	หากผู้ใช้ไม่ได้ทำการเลือกการกระทำ (Action) ไว้จะไม่สามารถทำการหมุนวัตถุรูปทรงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.26 รายละเอียดยูสเคส Resize Shape

ชื่อยูสเคส	Resize Shape
คำอธิบาย	การย่อหรือขยายวัตถุรูปทรง
เหตุการณ์ที่กระตุ้นการทำงาน	เมื่อผู้ใช้ทำการกำมือ แล้วเคลื่อนที่มือเข้า หรือออกจากกล้อง
คำอธิบายยูสเคส	เป็นขั้นตอนการย่อหรือขยายวัตถุรูปทรงที่เลือกไว้
แอกเตอร์	User
ยูสเคสที่เกี่ยวข้อง	-
ผู้เกี่ยวข้องอื่น	-
เงื่อนไขเริ่มต้น	ผู้ใช้ทำการเลือกเงื่อนไขการกระทำ (Action) ระบบจะเปลี่ยนสถานะเป็นการกระทำกับวัตถุรูปทรง
เงื่อนไขเมื่อสำเร็จ	-
การทำงาน	<ol style="list-style-type: none"> <li>1. ผู้ใช้เคลื่อนที่ตำแหน่งของมือในลักษณะแบมือ ไปยังบริเวณวัตถุรูปทรงที่ต้องการ</li> <li>2. ผู้ใช้ทำการกำมือเพื่อทำการเลือกวัตถุรูปทรงที่จะทำการย่อหรือขยาย</li> <li>3. ผู้ใช้ทำการเคลื่อนที่มือเข้าหรือออกจากกล้องตามต้องการ</li> <li>4. ผู้ใช้ทำการปล่อยมือเพื่อยกเลิกการย่อหรือขยาย</li> </ol>
เงื่อนไขการทำงาน	หากผู้ใช้ไม่ได้ทำการเลือกการกระทำ (Action) ว่าจะไม่สามารถทำการย่อหรือขยายวัตถุรูปทรงได้

ตารางที่ 3.27 รายละเอียดยูสเคส Clear Screen

ชื่อยูสเคส	Clear Screen
คำอธิบาย	การล้างหน้าจอวาดรูป
เหตุการณ์ที่กระตุ้นการทำงาน	เมื่อผู้ใช้ทำการกำมือ ณ บริเวณที่กำหนด
คำอธิบายยูสเคส	เป็นขั้นตอนการล้างหน้าจอวาดรูป
แอกเตอร์	User
ยูสเคสที่เกี่ยวข้อง	-
ผู้เกี่ยวข้องอื่น	-
เงื่อนไขเริ่มต้น	-
เงื่อนไขเมื่อสำเร็จ	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตารางที่ 3.27 (ต่อ)

การทำงาน	<ol style="list-style-type: none"> <li>1. ผู้ใช้เคลื่อนที่ตำแหน่งของมือในลักษณะแบมือไปยังบริเวณที่กำหนด</li> <li>2. ผู้ใช้ทำการกำมือเพื่อทำการล้างหน้าจอ</li> </ol>
เงื่อนไขการทำงาน	หากตำแหน่งมือของผู้ใช้ไม่ได้อยู่ในบริเวณที่กำหนด จะไม่สามารถล้างหน้าจอได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

# การพัฒนาโปรแกรม

### 4.1 เครื่องมือที่ใช้ในการพัฒนา

#### 4.1.1 ซอฟต์แวร์

- Microsoft Visual Studio .NET 2005 โดยใช้ภาษา C# ในการพัฒนาโปรแกรม ซึ่งเป็น
- OpenCV Library ชุดพัฒนาของด้านการประมวลผลภาพจาก Intel

#### 4.1.2 ฮาร์ดแวร์

##### Web Camera

- Samsung AnyCAM กล้องวิดีโอความละเอียด 340x240 Pixels และอัตราการแสดงผลภาพ 30 เฟรมต่อวินาที

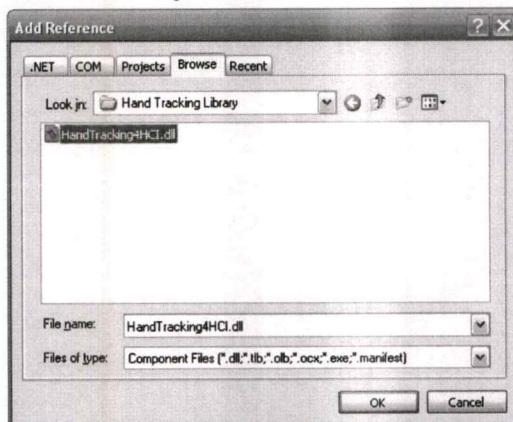
##### Personal Computer (Laptop)

- CPU : 1.73GHz Intel Pentium M processor
- RAM: 1GB
- Hardisk: 80GB
- Display Card: 64MB ATI MOBILITY RADEON

### 4.2 การพัฒนาโปรแกรมประยุกต์โดยใช้ชุดคำสั่งติดตามการเคลื่อนไหวของมือ

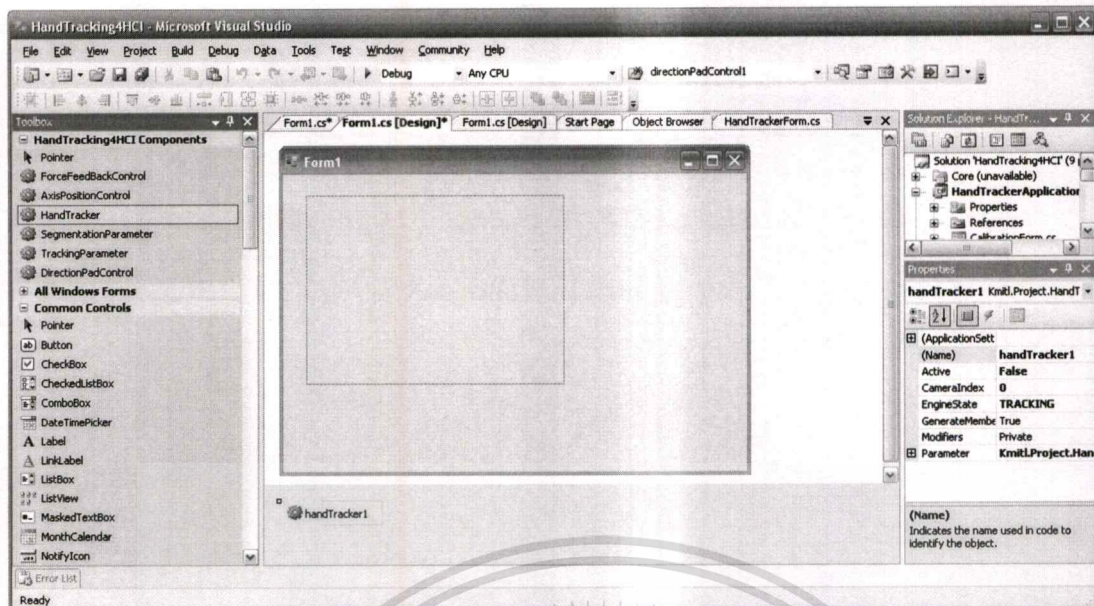
#### 4.2.1 การนำเข้าชุดคำสั่ง

การใช้งาน Component ทั่วไปสามารถทำได้โดยการ Add Reference เข้ามาใน Project ที่ต้องการ ดังรูปที่ 4.1 หลังจากนั้นที่หน้าต่าง Toolbox จะปรากฏรายการ Component ที่เพิ่มเข้ามาใหม่ ซึ่งสามารถนำมาใช้งานได้ทันที ดังรูปที่ 4.1-4.2



รูปที่ 4.1 แสดงการ Add Reference เพื่อใช้งาน Hand Tracker Component

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงการใช้งาน Hand Tracker Component บน Visual Studio .NET

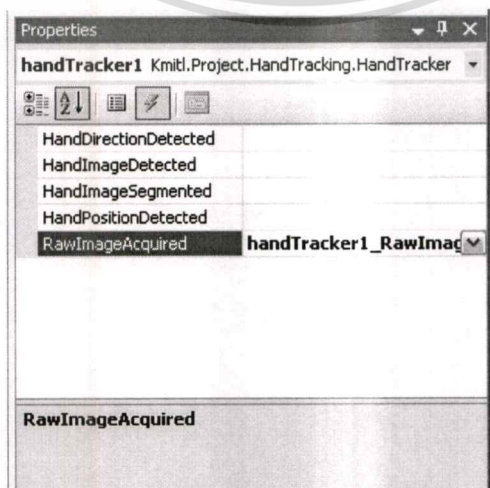
### 4.2.2 การใช้งานคำสั่ง

การใช้งาน Hand Tracker Component จะมี Function ที่ให้สามารถเรียกใช้งานในการควบคุมการทำงานของชุกคำสั่งได้ดังนี้ คือ

- Function Start() จะเป็นการเริ่มการทำงานติดตามการเคลื่อนไหวของมือ
- Function Stop() จะเป็นการหยุดการทำงานติดตามการเคลื่อนไหวของมือ
- Function Preference() จะเป็นการเรียกหน้าต่างสำหรับตั้งค่าระบบ

### 4.2.3 การใช้งานเหตุการณ์

ชุดคำสั่งภายใน Hand Tracker Component จะสามารถส่งข้อมูลผลลัพธ์ออกมาให้โปรแกรมประยุกต์ใช้งานได้โดยผ่านเหตุการณ์ (Events) ดังแสดงในรูปที่ 4.3



รูปที่ 4.3 แสดง Delegate Events เหตุการณ์ที่ถูกส่งออกมาจาก Hand Tracker Component

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นาเบะชอบประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1) การแสดงผลข้อมูลภาพ

โดยเหตุการณ์ที่ส่งออกข้อมูลภาพจะมีด้วยกัน 3 เหตุการณ์ คือ

- RawImageAcquired เมื่อได้รับข้อมูลภาพดิบจากกล้องโดยตรง
- HandImageDetected เมื่อได้รับรูปหลังจากหารูปมือพบ

ข้อมูลรูปภาพที่ส่งมาจาก Hand Tracker Component จะอยู่ในตัวแปร CvImageEventArgs ในรูปแบบ BITMAP ถ้าต้องการนำภาพดิบที่ได้จากกล้องผ่าน เหตุการณ์ RawImageAcquired นำมาแสดงบน PictureBox สามารถเขียนคำสั่งได้ดังนี้

```
private void handTracker1_RawImageAcquired(object sender,
                                           Kmitl.Project.HandTracking.CvImageEventArgs e)
{
    //Show image on picture box
    pictureBox1.Image = new Bitmap(e.Image);
}
```

### 2) การแสดงผลข้อมูลการติดตามตำแหน่งของมือ

ข้อมูลพิกัดตำแหน่งของมือ มุมการหมุน และขนาดของมือที่ตรวจพบสามารถนำ ข้อมูลเหล่านี้มาใช้งานได้จากเหตุการณ์ HandPositionDetected โดยข้อมูลดังกล่าวจะถูก เก็บอยู่ในตัวแปร HandPositionEventArgs สามารถเขียนคำสั่งได้ดังนี้

```
private void handTracker1_HandPositionDetected(object sender,
                                               Kmitl.Project.HandTracking.HandPositionEventArgs e)
{
    //Hand Position
    int posX = e.Position.X;
    int posY = e.Position.Y;
    int posZ = e.Position.Z;

    //Hand Rotation
    int angel = e.Angle;

    //Hand Size
    int handWidth = e.Size.Width;
    int handHeight = e.Size.Height;

    //Do something
    //..
}
```

### 3) การแสดงผลข้อมูลการทิศทางการเคลื่อนที่ของมือ

ทิศทางการเคลื่อนที่ของมือ 8 ทิศทาง (ขึ้น, ลง, ซ้าย, ขวา, เฉียงขึ้นซ้าย, เฉียงลงซ้าย, เฉียงขึ้นขวา และ เฉียงลงขวา) เป็นข้อมูลการเคลื่อนที่จากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่ง โดยไม่สนใจพิกัดตำแหน่ง ที่มาจากเหตุการณ์ HandDirectionDetected โดยข้อมูลทิศทางนั้น จะอยู่ในตัวแปร HandDirectionEventArgs สามารถเขียนคำสั่งได้ดังนี้

```
private void handTracker1_HandDirectionDetected(object sender,
                                                Kmitl.Project.HandTracking.HandDirectionEventArgs e)
{
    DirectionType direction = e.Direction;
    //Do something
    //..
}
```

## 4.3 การใช้งานโปรแกรมประยุกต์

### 4.3.1 คุณลักษณะของโปรแกรม

โครงสร้างของโปรแกรมแบ่งเป็น 2 ส่วน คือ ส่วนการควบคุมที่อยู่ด้านขวามือ และส่วนการวาดรูปอยู่ได้ซ้าย โดยมีรายละเอียดดังนี้

- 1) ส่วนการแสดงผลที่ได้รับจากชุดคำสั่งติดตามการเคลื่อนไหวของมือ โดยถูกนำมาแสดงผลบนหน้าจอเพื่อให้ผู้ใช้สามารถเห็นสถานะหรือตำแหน่งของมือในปัจจุบัน
- 2) ปุ่มการตั้งค่าระบบของชุดคำสั่งติดตามการเคลื่อนไหวของมือในกรณีที่ กระบวนการชี้ตำแหน่งที่แสดงในด้านขวามือ ไม่ถูกต้อง
- 3) หมวดการวาดรูปที่มีลักษณะรูปร่างอยู่แล้ว ซึ่งมีอยู่ด้วยกัน 3 รูปร่างคือ รูปร่างดาว รูปร่างสี่เหลี่ยม และรูปร่างหัวใจ
- 4) คำสั่งจับวัตถุเพื่อทำการหมุน หรือย่อขยายขนาดภาพ
- 5) ส่วนการเลือกสี ในกรณีที่ใช้แปรงในการวาดเส้น โดยตรง โดยมีสีให้เลือกทั้งหมด 4 สี คือ สีน้ำเงิน สีเขียว สีแดง และสีเหลือง
- 6) คำสั่งแปรง เป็นการเลือกที่จะใช้แปรงในการวาดรูปบนหน้าจอตามสีที่ถูกเลือกไว้
- 7) คำสั่งล้าง เป็นการล้างหน้าจอให้เป็นสีขาวทั้งหมด

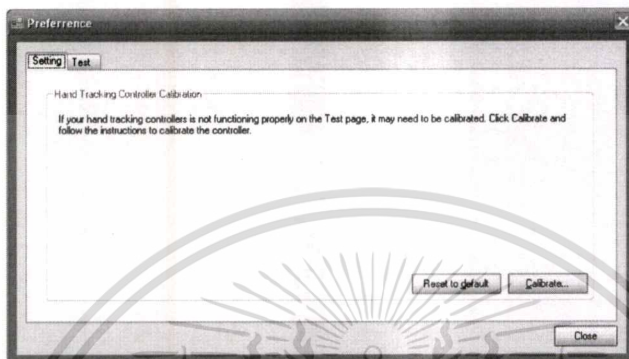


รูปที่ 4.4 แสดงหน้าต่างแสดงผลของโปรแกรมประยุกต์ (Hand Paint)

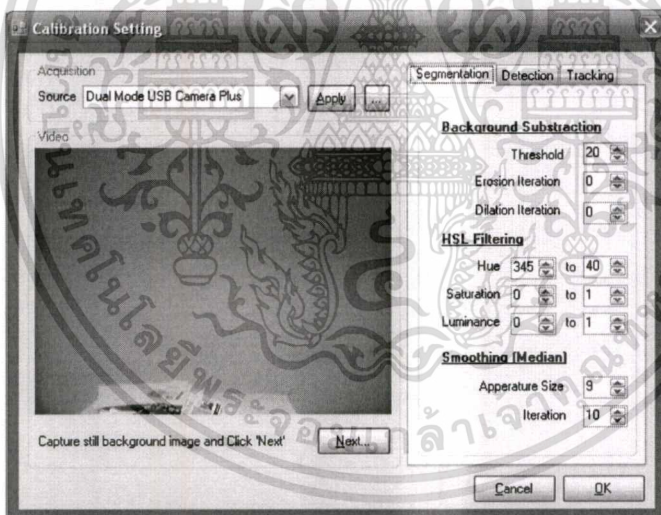
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3.2 การตั้งค่าระบบติดตามการเคลื่อนไหวของมือ

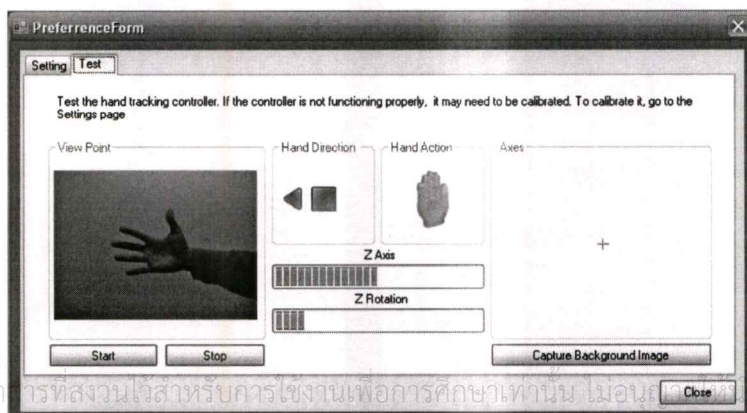
ในกรณีที่ระบบไม่สามารถส่งข้อมูลผลลัพธ์ออกมาได้อย่างถูกต้อง สามารถทำการตั้งค่าเริ่มต้นต่างๆ ได้โดยเรียกใช้คำสั่ง Function Preference() จาก Hand Tracker Component จะปรากฏหน้าต่างสำหรับตั้งค่าดังรูปที่ 4.5 โดยจะประกอบด้วย 2 ส่วนคือ ส่วนการ ตั้งค่า (Setting) ซึ่งสามารถทำการปรับค่าต่างๆ ได้ ดังรูปที่ 4.6 และส่วนการทดสอบระบบ (Test) ดังรูปที่ 4.7



รูปที่ 4.5 แสดงหน้าต่างสำหรับแก้ไข



รูปที่ 4.6 แสดงหน้าต่างสำหรับตั้งค่า

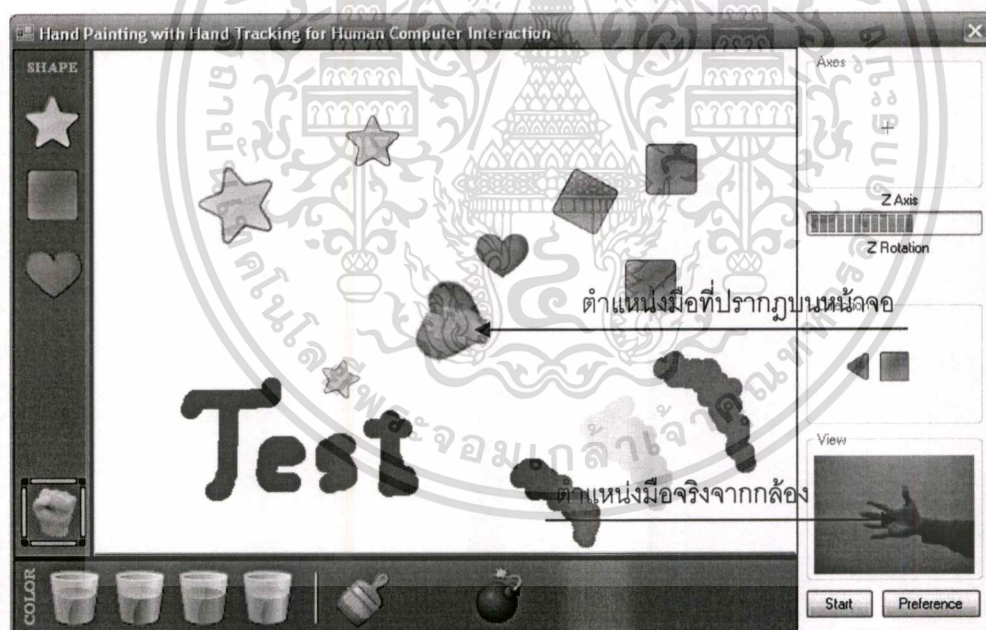


รูปที่ 4.7 แสดงหน้าต่างสำหรับการทดสอบการติดตามมือ

### 4.3.3 การวาดรูป

การวาดรูปโดยใช้ตัวควบคุมด้วยการมือเคลื่อนที่ของมือโดยมีองค์ประกอบดังนี้

- 1) การชี้ตำแหน่ง ในการเลือกคำสั่ง หรือ การวาดเส้นโดยใช้แปรง หรือการเลือกวัตถุ รูปทรงบนหน้าจอเพื่อขยายหรือหมุน จะใช้ตัวชี้ตำแหน่งจากตำแหน่งพิกัด X, Y ที่ได้จากการตรวจจับตำแหน่งของมือ
- 2) การสั่งการ จากการเลือกเมนูคำสั่ง การสั่งวาดเส้นลงบนหน้าจอ การเทสี การสั่งขยายหรือหมุนวัตถุบนหน้าจอที่มีการชี้ตำแหน่งไว้ จะใช้การควบคุมโดยท่าทางการกำมือ โดยการวาดเส้น การหมุน การขยาย ต้องทำการกำมือค้างไว้ตลอด
- 3) การหมุนรูปทรง ที่วาดบนหน้าจอแล้วสามารถทำได้โดยการชี้ตำแหน่งบนวัตถุที่ต้องการแล้วสั่งการค้างไว้ (กำมือ) จากนั้นจึงหมุนทิศทางของมือ
- 4) การขยายรูปทรง ที่วาดบนหน้าจอแล้วสามารถทำได้โดยชี้ตำแหน่งบนวัตถุที่ต้องการแล้วสั่งการค้างไว้ (กำมือ) จากนั้นจึงเลื่อนมือเข้า หรือออกจากกล้อง เพื่อทำการขยายรูปทรงตามต้องการ



รูปที่ 4.8 แสดงหน้าต่างแสดงผลโปรแกรมในการทดสอบวาดรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

# บทสรุปและข้อเสนอแนะ

### 5.1 ปัญหาและอุปสรรค

การใช้งานระบบในขั้นตอนติดตามหากมีการเลื่อนที่มือเลขวิสัยการจับภาพของกล้องออกไป จะไม่สามารถจับการเคลื่อนไหวได้ทำให้ความละเอียดในการหาพิกัดการเคลื่อนที่ถูกจำกัดโดยความละเอียดของกล้องถ่ายภาพกล่าวคือ กล้องมีความละเอียดในแนวแกน X มีขนาด 320 Pixels และแกน Y มีขนาด 240 Pixels ดังนั้นค่าความละเอียดของพิกัดในแนวแกน X ที่ติดตามได้จะมี 320 ค่า และแกน Y ที่ติดตามได้จะมี 240 ค่าเท่านั้น

การตรวจจับการกำมือหรือแบมือ โดยใช้ตัวแปลงฮัพแบบวงกลมนั้นยังมีข้อจำกัดในเรื่องของการกำหนดรัศมีของมือให้ถูกต้องเพื่อให้การตรวจจับมีความแม่นยำ หากปรับปรุงให้มีการตรวจสอบรัศมีเป็นแบบช่วง หากกำหนดช่วงกว้างเกินไปก็จะทำให้เสียเวลาในการประมวลผลนานมากขึ้นทำให้การแสดงผลพีธีเกิดความล่าช้าจากเวลาจริงออกไป

### 5.2 แนวทางการประยุกต์ใช้ร่วมกับงานอื่นในขั้นต่อไป

จากชุดคำสั่งติดตามการเคลื่อนไหวของมือ ซึ่งมีลักษณะเป็นชุดคำสั่งเชิงส่วนประกอบ (Component) อยู่แล้ว สามารถนำไปใช้ร่วมกับโปรแกรมประยุกต์ในรูปแบบอื่นได้อย่างสะดวกรวดเร็วโดยเฉพาะบน .NET Platforms ทำให้สามารถพัฒนาโปรแกรมหรือเกม ที่สามารถใช้การเคลื่อนที่ของมือ การกำมือ ทิศทางการเคลื่อนที่ของมือมาใช้ในการควบคุมภายใน โปรแกรม หรือ เกมที่ต้องการได้ทันที

โดยชุดคำสั่งสามารถนำไปใช้ในการพัฒนาต่อเพื่อเพิ่มเติมความสามารถที่ต้องการได้ เช่น พัฒนาต่อให้มีความสามารถในการรู้จำการเคลื่อนที่ของมือ โดยใช้ข้อมูลตำแหน่งของมือจากชุดคำสั่งเดิมได้ทันที ด้วยกระบวนการพัฒนาเชิงวัตถุ โดยสามารถสืบทอดคลาสชุดคำสั่งเพื่อนำมาเพิ่มเติมความสามารถ หรือปรับแต่งได้ตามต้องการ

### 5.3 ข้อสรุปและข้อเสนอแนะ

ในกระบวนการติดตามการเคลื่อนไหวของมือยังสามารถเพิ่มขีดความสามารถออกไปได้อีกในส่วนของการติดตามหาตำแหน่งของนิ้วซึ่งสามารถใช้ตัวแปลงฮัพแบบวงกลมนำไปประยุกต์ใช้ในการหาปลายนิ้วมือได้เช่นเดียวกัน อีกทั้งยังสามารถเพิ่มความสามารถในการรู้จำท่าทางลักษณะของมือแบบต่างๆ นอกเหนือจากการกำมือและแบมือ และให้สามารถรู้จำทิศทางการเคลื่อนที่ของมือ นำมาผสมผสานกันระหว่างทิศทางการเคลื่อนที่ของมือและลักษณะท่าทางมือ เพื่อเป็นการเพิ่มขีดความสามารถในการติดตามและรู้จำการเคลื่อนไหวของมือ ทำให้การโต้ตอบกับระบบคอมพิวเตอร์มีประสิทธิภาพและมีความแม่นยำมากขึ้น

ข้อเสนอแนะอื่นที่ผู้วิจัยได้ศึกษาเพื่อนำมาปรับปรุงงานวิจัยนี้ให้ดียิ่งขึ้นคือ การศึกษาเพื่อนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

- Barron, J. L., Fleet, D. J., and Beauchemin, S. 1994. “*Performance of optical flow techniques.*”, 12(1): 43-77 in *International Journal of Computer Vision.*
- Bradski, Gary R. 1998. “*Computer vision face tracking for use in a perceptual user interface.*”, *Intel Technology Journal*, 2nd Quarter.
- Canny, J. 1986. “*A Computational Approach to Edge Detection*”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:679-714.
- Fisher, S. Perkins, Walker, A. and Wolfart, E. 2003. “*Morphology*”. [Online]. Available: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/morops.htm>
- Foley, James D., Dam, Andries van. 1982. “*Fundamentals of Interactive Computer Graphics.*”, Boston, MA, USA: Addison-Wesley.
- Gerardin, Peggy. n.d. “*Color Image Segmentation*”, [Online]. Available: [http://lcavwww.epfl.ch/~gerardin/Papers/Color\\_Image\\_Segmentation.pdf](http://lcavwww.epfl.ch/~gerardin/Papers/Color_Image_Segmentation.pdf)
- Hight, Ronald. 2004. “*Optical vehicle tracking - A framework and tracking solution.*”, Computer Science Department Canterbury University, Christchurch, New Zealand.
- Malik, Shahzad. 2003. “*Real-time Hand Tracking and Finger Tracking for Interaction*”, CSC2503F Project Report of Toronto University.
- Schaller, N. n.d. “*Color Conversion Algorithms*”. [Online]. Available: [http://www.cs.rit.edu/~ncs/color/t\\_convert.html](http://www.cs.rit.edu/~ncs/color/t_convert.html)



## ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Namespace: Kmitl.Project.HandTracking****Class: HandTracker**

- **Constructor: HandTracker ()**

**Summary:** ทำการสร้าง Object ของ HandTracker.

- **Constructor: HandTracker (IContainer container)**

**Summary:** ทำการสร้าง Object ของ HandTracker.

**Parameters:            Description**

container                Container ของ Windows Form

- **Property: int CameraIndex**

**Summary:** ค่าลำดับของกล้องวิดีโอที่ถูกรู้ใช้งาน

**Return:** หมายเลขกล้องที่ใช้งาน

- **Property: Boolean Active**

**Summary:** สถานะการทำงาน

**Return:** true หมายถึงกำลังทำงาน และ false หมายถึงหยุดทำงาน

- **Property: Boolean FlipImage**

**Summary:** การกลับข้างของภาพวิดีโอที่นำเข้าก่อนการประมวลผล

**Return:** true หมายถึงกลับข้างภาพวิดีโอ และ false หมายถึงไม่ต้องกลับข้างภาพวิดีโอ

- **Property: HandTrackerParameter Parameter**

**Summary:** สำหรับการตั้งค่าตัวแปรชุดคำสั่ง

**Return:** Object ของ HandTrackerParameter ใช้สำหรับตั้งค่าชุดคำสั่ง

- **Method: void Start ()**

**Summary:** ทำการเริ่มการทำงานของชุดคำสั่งติดตามมือ

- **Method: void Stop ()**

**Summary:** ทำการหยุดการทำงานของชุดคำสั่งติดตามมือ

- **Method: void Preference ()**

**Summary:** ทำการเรียกหน้าต่างปรับแต่งการตั้งค่าชุดคำสั่งติดตามมือ

- **Method: void InitBackground ()**

**Summary:** ทำการจับภาพฉากหลังเริ่มต้นของการประมวลผลภาพ

- **Event: RawImageAcquired (CvImageEventArgs e)**

**Summary:** เหตุการณ์การรับภาพต้นฉบับจากกล้องวิดีโอ

**Parameters:**

**Description**

e

Object ของคลาส CvImageEventArgs

- **Event: HandImageDetected (CvImageEventArgs e)**

**Summary:** เหตุการณ์การรับภาพเมื่อสามารถตรวจจับภาพมือได้

**Parameters:**

**Description**

e

Object ของคลาส CvImageEventArgs

- **Event: HandPositionDetected (HandPositionEventArgs e)**

**Summary:** เหตุการณ์การตรวจจับหาตำแหน่งของมือ

**Parameters:**

**Description**

e

Object ของคลาส HandPositionEventArgs

- **Event: HandDirectionDetected (HandDirectionEventArgs e)**

**Summary:** เหตุการณ์การตรวจจับทิศทางเคลื่อนที่ของมือ

**Parameters:**

**Description**

e

Object ของคลาส HandDirectionEventArgs

**Class: CvImageEventArgs**

- **Constructor: CvImageEventArgs (CvImage image)**

**Summary:** สร้าง Object ของ CvImageEventArgs ใช้ส่งข้อมูลเหตุการณ์การจับภาพ

Parameters:	Description
Image	ภาพใน format RGB

- **Property: Bitmap Image**

**Summary:** ภาพที่ส่งออกมาจากเหตุการณ์

**Return:** object ของภาพ Bitmap

**Class: HandDirectionEventArgs**

- **Constructor: HandDirectionEventArgs (DirectionType direction)**

**Summary:** สร้าง Object ของ HandDirectionEventArgs ใช้ส่งข้อมูลทิศทางการเคลื่อนที่

Parameters:	Description
direction	ทิศทางการเคลื่อนที่

- **Property: DirectionType Direction**

**Summary:** ทิศทางการเคลื่อนที่ที่ส่งมาพร้อมกับเหตุการณ์

**Return:** ตัวแปรทิศทางการเคลื่อนที่ ของ DirectionType

**Class: HandPositionEventArgs**

- **Constructor: HandPositionEventArgs (int)**

**Summary:** สร้าง Object ของ HandPositionEventArgs ใช้ส่งข้อมูลทิศทางการเคลื่อนที่

Parameters:	Description
videoWidth	ขนาดความกว้างของภาพวิดีโอนำเข้า
videoHeight	ขนาดความสูงของภาพวิดีโอนำเข้า
handWidth	ขนาดความกว้างของภาพมือ
handHeight	ขนาดความสูงของภาพมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 posX ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

posY	ตำแหน่งมือแนวแกน Y
posZ	ตำแหน่งมือแนวแกน Z
angle	มุมการหมุนมือแนวแกน Z
handType	ลักษณะท่าทางมือ FIST หรือ FLAT

- **Property: Size Size**

**Summary:** ขนาดความกว้างและความสูงของภาพมือที่ส่งมาพร้อมกับเหตุการณ์

**Return:** ขนาดความกว้างและความสูงของภาพมือ

- **Property: Size VideoSize**

**Summary:** ขนาดความกว้างและความสูงของภาพวิดีโอที่ส่งมาพร้อมกับเหตุการณ์

**Return:** ขนาดความกว้างและความสูงของภาพวิดีโอ Width และ Height

- **Property: Point3D Position**

**Summary:** ตำแหน่งมือในระบบ 3 มิติที่ส่งมาพร้อมกับเหตุการณ์

**Return:** ตำแหน่งมือในระบบ 3 มิติ X, Y, Z

- **Property: float Angle**

**Summary:** มุมการหมุนของมือที่ส่งมาจากเหตุการณ์

**Return:** มุมการหมุนของมือ 180 องศา

- **Property: HandType Hand**

**Summary:** ลักษณะท่าทางมือที่ส่งมาพร้อมกับเหตุการณ์

**Return:** ตัวแปรลักษณะท่าทางมือของ HandType



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## คู่มือการติดตั้งชุดคำสั่งติดตามการเคลื่อนไหวของมือ

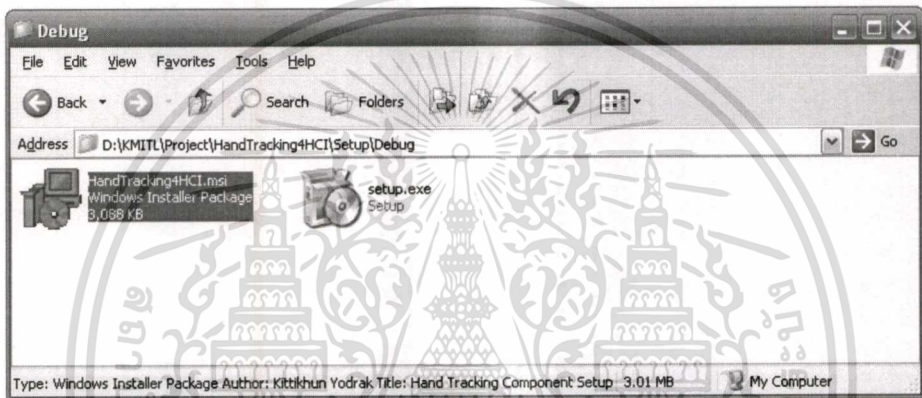
การติดตั้งระบบประกอบไปด้วย 2 ส่วนหลักๆที่เกี่ยวข้อง ได้แก่

1. การติดตั้งชุดคำสั่งเข้าสู่ระบบ
2. การติดตั้งชุดคำสั่งเข้าสู่โปรแกรมที่พัฒนา

### 1. การติดตั้งชุดคำสั่งเข้าสู่ระบบ

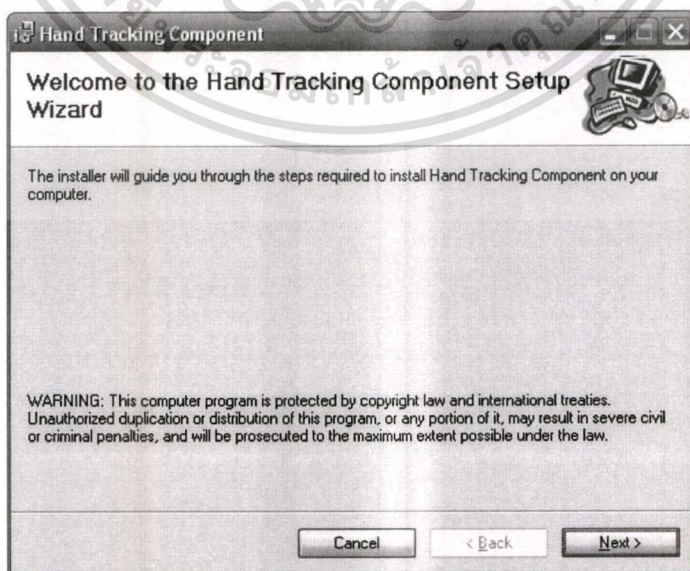
โดยขั้นตอนการติดตั้งโปรแกรมดำเนินการได้ดังนี้

- 1.1 ดับเบิลคลิกที่ตัวไอคอน HandTracking4HCI.msi ดังรูปที่ ข.1



รูปที่ ข.1 หน้าจอแสดงไอคอนตัวติดตั้ง โปรแกรม

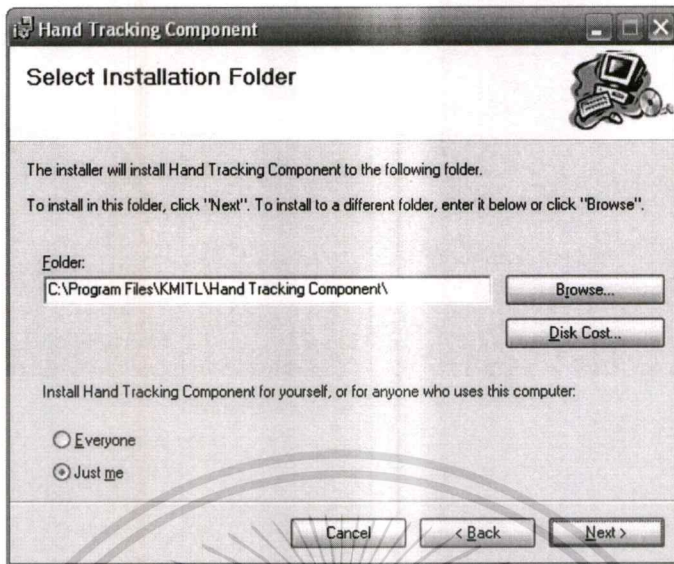
- 1.2 เมื่อดับเบิลคลิกที่ไอคอนตัวติดตั้งเรียบร้อยแล้ว จะปรากฏหน้าจอ ดังรูปที่ ข.2



รูปที่ ข.2 หน้าจอแสดงขั้นตอนการติดตั้งโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 หลังจากนั้นให้คลิก Next จะปรากฏหน้าจอ ดังรูปที่ ข.3



รูปที่ ข.3 หน้าจอแสดงการกำหนดพารามิเตอร์ติดตั้งโปรแกรม

1.4 หลังจากนั้นคลิก Next ต่อไปจนเสร็จสิ้นจะปรากฏไอคอนในหน้าต่าง Start> All Programs> KMITL> Hand Tracking Component ดังรูปที่ ข.4

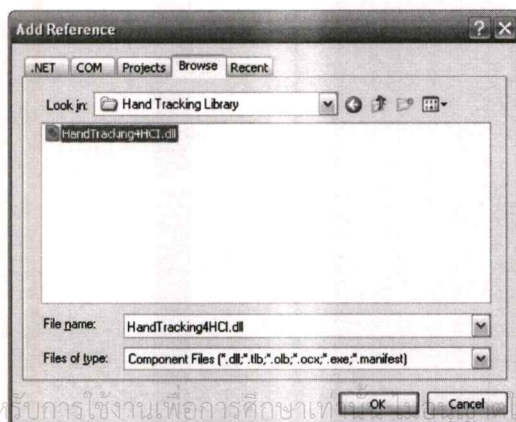


รูปที่ ข.4 หน้าจอแสดงหน้าต่างหลังจากติดตั้งชุดคำสั่ง

## 2. การติดตั้งชุดคำสั่งเข้าสู่โปรแกรมที่พัฒนา

โดยขั้นตอนการติดตั้งโปรแกรม สามารถจำแนกออกได้เป็นสองส่วน ได้แก่

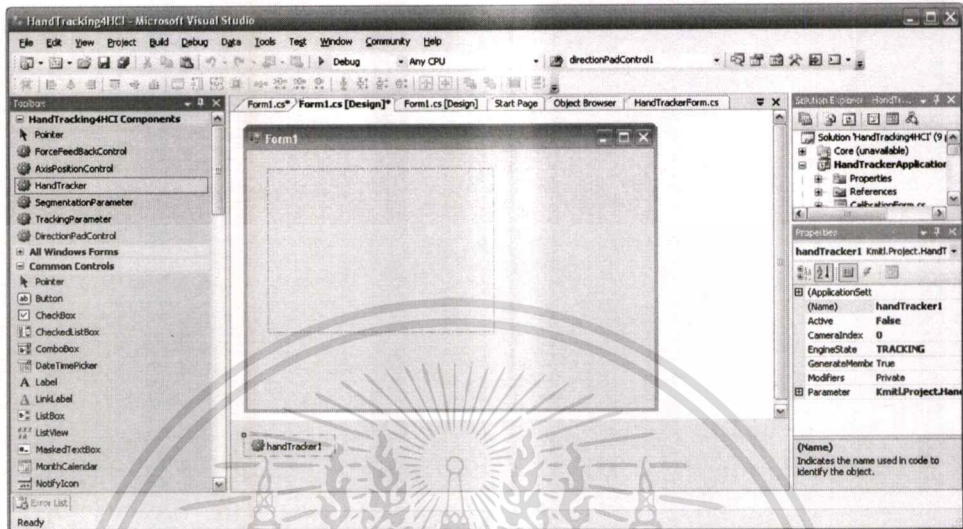
2.1 Microsoft Visual Studio สามารถติดตั้งชุดคำสั่งเข้าสู่โปรแกรมที่พัฒนาได้โดย การ Add Reference เข้ามาในโครงการที่ต้องการ ดังรูปที่ ข.5



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้นนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกรูปที่ ข.5 Add Reference เพื่อใช้งาน Hand Tracker Component รั้งที่มีการนำไปใช้

2.2 เมื่อทำการ Add Reference เข้าสู่โครงการเรียบร้อยแล้วจะปรากฏไอคอน Component อยู่ในหน้าต่าง Toolbox ดังรูปที่ ข.6



รูปที่ ข.6 การใช้งาน Hand Tracker Component บน Microsoft Visual Studio .NET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้เขียน

ชื่อผู้เขียน

นางกิตติคุณ ขอดรัก

สถานที่เกิด

จังหวัดกรุงเทพมหานครฯ

การศึกษา

ระดับปริญญาตรี

วท.บ. (วิทยาศาสตรบัณฑิต)

สาขาวิทยาการสารสนเทศคอมพิวเตอร์

มหาวิทยาลัยเกษมบัณฑิต

ประสบการณ์การทำงาน

โปรแกรมเมอร์

ผู้จัดการ โครงการ

บริษัท ไอทีเวิร์คส์ จำกัด กรุงเทพมหานครฯ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้