



การวิเคราะห์และรู้จำคำสั่งท่าทางของมือสำหรับระบบเฝ้าระวัง
ด้วยกล้องวงจรปิด

**Hand Gesture Analysis and Recognition for Video Surveillance
Command System**

กฤตศิลป์ ติลานนท์
Kittasil Silanon

วิทยานิพนธ์นี้สำหรับการศึกษาตามหลักสูตรปริญญา
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

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

2554

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

ชื่อวิทยานิพนธ์	การวิเคราะห์และรู้จำคำสั่งท่าทางของมือสำหรับระบบเฝ้าระวังด้วยกล้อง วงจรปิด
ผู้เขียน	นายกฤตศิลป์ ศิลานนท์
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	2553

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้เป็นงานวิจัยเพื่อประยุกต์ใช้เทคนิคด้านการประมวลผลภาพวิเคราะห์ท่าทางการเคลื่อนที่ของมือสำหรับการรู้จำชุดคำสั่งเพื่อใช้สำหรับการติดต่อระหว่างระบบคอมพิวเตอร์หรือเครื่องจักรกลกับผู้ใช้งาน โดยในงานชิ้นนี้จะมุ่งเน้นไปที่เทคนิคของการรู้จำชุดคำสั่งเพียงอย่างเดียว ซึ่งชุดคำสั่งนั้นผู้ใช้สามารถกำหนดขึ้นเองได้และนำไปประยุกต์ใช้กับงานตามแต่ความต้องการของผู้ใช้ในภายหลัง โดยเทคนิคการรู้จำชุดคำสั่งด้วยท่าทางมือประกอบด้วย (1) กระบวนการตรวจจับและติดตามรูปร่างลักษณะของมือภายในภาพวิดีโอ เพื่อให้ได้บริเวณที่สนใจ (2) กระบวนการหาคุณลักษณะเด่นที่ได้จากการติดตามมือ ซึ่งผลจากการติดตามจะให้ข้อมูลของทิศทางการเคลื่อนที่ ขนาด และมุมของแต่ละท่าทาง ข้อมูลดังกล่าวจะถูกนำไปใช้เพื่ออธิบายถึงลักษณะเด่นท่าทางมือที่จะใช้ในการรู้จำ (3) กระบวนการรู้จำท่าทางมือ ซึ่งแบ่งเป็น 2 กลุ่ม ได้แก่ (3.1) การรู้จำท่าทางมืออย่างง่าย เช่น การโบกมือไปทางซ้าย-ขวา หรือขึ้น-ลง ได้ใช้วิธีไฟไนต์สเตตแมชชีนในการกำหนดเงื่อนไขการรู้จำท่าทางจากคุณลักษณะเด่นที่หามาได้ โดยได้มีการทดลองใช้การรู้จำชุดคำสั่งในการควบคุมการทำงานของโปรแกรมคอมพิวเตอร์เพื่อใช้งานจริง และ (3.2) การรู้จำท่าทางมือแบบสัญลักษณ์ เช่น การโบกมือเป็นท่าทางวงกลม สี่เหลี่ยม สามเหลี่ยม หรือท่าทางอื่นๆ โดยใช้แบบจำลองฮิดเดนมาร์คอฟสร้างแบบจำลองที่ใช้อธิบายคุณลักษณะของท่าทางนั้นๆ จากตัวอย่างชุดข้อมูลสำหรับการฝึกสอนเพื่อใช้ในการรู้จำ ซึ่งสัญลักษณ์ในที่นี้คือตัวอักษรภาษาไทยที่แบ่งกลุ่มตามลักษณะความคล้ายคลึงกันได้ 12 กลุ่ม และเลือกเอาเฉพาะตัวอักษรหนึ่งตัวที่ง่ายที่สุดในแต่ละกลุ่มมาใช้เป็นสัญลักษณ์ในการรู้จำ โดยผลการรู้จำท่าทางมือให้ความถูกต้องโดยเฉลี่ย 84 เปอร์เซ็นต์ แต่ทั้งนี้ความสามารถในการรู้จำท่าทางมือของระบบจะขึ้นอยู่กับชุดข้อมูลท่าทางมือที่จะนำมาทำการฝึกสอนระบบ

Thesis Title	Hand Gesture Analysis and Recognition for Video Surveillance Command System
Author	Mr.Kittasil Silnon
Major Program	Computer Engineering
Academic Year	2010

ABSTRACT

This thesis applies image processing techniques for hand gesture recognition system based on hand motion analysis. The system is established into the following steps. Firstly, hand detection and tracking from image sequences to obtain region of interest. Secondly, feature extraction from hand motion that provides the important information of hand movement including size, angle and orientation which are reformulated into the features description of hand gesture characteristic. Finally, hand gesture recognition that we focus on two study cases: simple hand gestures recognition and hand symbols recognition. For simple hand gestures recognition such as moving hand to up, down, left or right direction. In this case, the finite state machine is applied to define the condition of recognition command from the hand's features. We have used hand command for controlling computer application in real situation. In the case of hand symbols recognition such as moving hand as square circle or other symbols. Hidden markov model is applied to generate the model that describe gesture characteristic from data training. The symbols is chosen from the 12 groups of Thai alphabet that considering from its similarities. The experimentation result of hand symbols recognition is around 84%. However, the recognition rate will depend on data training that used to train the system.

สารบัญ

	หน้า
สารบัญ.....	(6)
รายการรูปภาพ.....	(9)
รายการตาราง.....	(12)
บทที่ 1 บทนำ.....	13
1.1. ความสำคัญและที่มาของการวิจัย	13
1.2. งานวิจัยที่เกี่ยวข้อง	13
1.3. วัตถุประสงค์	18
1.4. ขอบเขตของการวิจัย.....	18
1.5. ประโยชน์ที่คาดว่าจะได้รับ	18
บทที่ 2 ทฤษฎีและหลักการ	19
2.1. กระบวนการตรวจจับวัตถุที่สนใจ	19
2.1.1. คุณลักษณะเด่นแบบฮาไลต์	19
2.1.2. อัลกอริทึมในการเรียนรู้แบบอดาปต	20
2.1.3. การแยกประเภทแบบลำดับชั้น	22
2.2. กระบวนการติดตามวัตถุเคลื่อนไหว.....	23
2.2.1. คุณสมบัติที่ใช้ในกระบวนการติดตาม.....	23
2.2.2. กระบวนการติดตามโดยการย้ายไปยังค่าเฉลี่ย	25
2.3. ตัวกรองแบบคาลแมน	27
2.4. แบบจำลองฮิดเดนมาร์คอฟ	30
2.4.1. การฝึกสอนแบบจำลอง	31
2.4.2. การทดสอบการรู้จำแบบจำลอง.....	33
2.5. สรุป.....	34

สารบัญ

	หน้า
บทที่ 3 การออกแบบและพัฒนาระบบ.....	35
3.1. กระบวนการตรวจจับและติดตามลักษณะมือ.....	36
3.1.1. การฝึกสอนด้วยการเรียนรู้แบบอคาบาสท.....	36
3.1.2. การตรวจจับวัตถุด้วยตัวแยกประเภทแบบลำดับชั้น.....	39
3.1.3. การติดตามมือโดยใช้คุณสมบัติของสี.....	40
3.1.4. การหาค่าประมาณที่ดีที่สุดด้วยตัวกรองกาลแมน.....	42
3.2. กระบวนการดึงคุณลักษณะเด่น.....	43
3.2.1. ลักษณะเด่นของมือ.....	43
3.2.2. การลดจำนวนข้อมูลของท่าทางมือ.....	44
3.2.2. การควอนไทซ์เวกเตอร์.....	46
3.3. กระบวนการรู้จำท่าทางมือ.....	47
3.3.1. การรู้จำท่าทางมือด้วยไฟไนต์สเตตแมชชีน.....	47
3.3.2. การรู้จำท่าทางมือด้วยแบบจำลองฮิดเดนมาร์คอฟ.....	49
3.4. สรุป.....	52
บทที่ 4 ผลการทดลอง.....	53
4.1. ผลการทดลองกระบวนการตรวจจับลักษณะมือ.....	53
4.2. ผลการทดลองกระบวนการติดตามการเคลื่อนที่ของมือ.....	56
4.3. ผลการทดลองกระบวนการลดจำนวนข้อมูลท่าทางมือ.....	61
4.4. ผลการทดลองกระบวนการรู้จำท่าทางมือ.....	63
4.4.1. ผลการทดลองการรู้จำท่าทางมืออย่างง่าย.....	63
4.4.2. ผลการทดลองการรู้จำท่าทางมือแบบสัญลักษณ์.....	63

สารบัญ

	หน้า
4.5. สรุป.....	77
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	78
5.1. สรุปผล	78
5.1.1. สรุปผลการทดลองกระบวนการตรวจจับลักษณะมือ	78
5.1.2. สรุปผลการทดลองกระบวนการติดตามการเคลื่อนที่ของมือ	79
5.1.3. สรุปผลการทดลองกระบวนการรู้จำท่าทางมือ.....	79
5.2. บทวิจารณ์	80
5.2.1. บทวิจารณ์ผลการทดลองกระบวนการตรวจจับลักษณะมือ.....	80
5.2.2. บทวิจารณ์ผลการทดลองกระบวนการติดตามการเคลื่อนที่ของมือ	80
5.2.3. บทวิจารณ์ผลการทดลองกระบวนการรู้จำท่าทางมือ	80
บรรณานุกรม	81
ภาคผนวก	83
ภาคผนวก ก. ผลงานตีพิมพ์เผยแพร่จากวิทยานิพนธ์.....	84
ประวัติผู้เขียน	98

รายการรูปภาพ

	หน้า
ภาพประกอบ 1-1 ผลลัพธ์ของการแยกค่าจุดสี	14
ภาพประกอบ 1-2 คุณลักษณะ 2 ชนิดที่ถูกดึงออกมา	15
ภาพประกอบ 1-3 กลุ่มภาพตัวอย่าง.....	15
ภาพประกอบ 1-4 ผลลัพธ์การรู้จำรูปร่างมือโดยใช้คุณลักษณะเด่นแบบฮาร์ไลท	16
ภาพประกอบ 1-5 การแยกจุดสีที่เป็นผิวหนังและการระบุตำแหน่งมือ	16
ภาพประกอบ 1-6 ตำแหน่งมือในทุกๆ เฟรมภาพตามช่วงเวลา (t).....	17
ภาพประกอบ 1-7 แบบจำลองฮิดเดนมาร์คอฟประเภทลำดับชั้นซ้าย-ขวา.....	17
ภาพประกอบ 2-1 คุณลักษณะเด่นแบบฮาร์ไลทสำหรับการตรวจจับในลักษณะแบบต่างๆ	19
ภาพประกอบ 2-2 วิธีการคำนวณภาพแบบอินทิกรัล	20
ภาพประกอบ 2-3 การคำนวณหาผลรวมค่าจุดสีในสี่เหลี่ยม D.....	20
ภาพประกอบ 2-4 การทำงานอดาบูสท (AdaBoost)	22
ภาพประกอบ 2-5 การทำงานของการแยกประเภทแบบลำดับชั้น (Cascade Classifiers).....	22
ภาพประกอบ 2-6 ลูกบาศก์ระบบสี RGB	23
ภาพประกอบ 2-7 ระบบสี HSV	25
ภาพประกอบ 2-8 กระบวนการย้ายไปยังค่าเฉลี่ย	26
ภาพประกอบ 2-9 การทำงานของตัวกรองคาลแมน (Kalman Filter)	30
ภาพประกอบ 2-10 ขั้นตอนการฝึกสอนแบบจำลอง (Training)	31
ภาพประกอบ 2-11 ขั้นตอนการทดสอบการจดจำของแบบจำลอง (Evaluation)	34
ภาพประกอบ 3-1 กระบวนการโดยรวมเทคนิคการรู้จำคำสั่งโดยท่าทางมือ	35
ภาพประกอบ 3-2 ชุดข้อมูลสำหรับการฝึกสอน	36
ภาพประกอบ 3-3 การกำหนดลำดับให้กับกลุ่มตัวอย่าง	37
ภาพประกอบ 3-4 คุณลักษณะเด่นมุมตั้งตรง (Upright) และมุมเอียง 45° องศา (45° Rotated).....	37
ภาพประกอบ 3-5 ตัวแยกประเภทแบบแข็งแกร่ง	38
ภาพประกอบ 3-6 กระบวนการตรวจจับวัตถุด้วยตัวแยกประเภทแบบลำดับชั้น	39
ภาพประกอบ 3-7 กระบวนการตรวจจับลักษณะมือ.....	40
ภาพประกอบ 3-8 การคำนวณหากราฟแสดงความหนาแน่นของสี	40

รายการรูปภาพ

	หน้า
ภาพประกอบ 3-9 การฉายภาพย้อนกลับ	41
ภาพประกอบ 3-10 กระบวนการติดตามโดยใช้คุณสมบัติของสี	41
ภาพประกอบ 3-11 การทำนายค่าสถานะ.....	42
ภาพประกอบ 3-12 การประมาณค่าสถานะ	43
ภาพประกอบ 3-13 คุณลักษณะเด่นของมือ	44
ภาพประกอบ 3-14 การลดจำนวนข้อมูลของท่าทาง.....	45
ภาพประกอบ 3-15 ขั้นตอนการทำงานของการประมาณค่ารูปหลายเหลี่ยม.....	45
ภาพประกอบ 3-16 การควอนไทซ์เวกเตอร์	46
ภาพประกอบ 3-17 ไฟไนต์สเตตแมชชีนสำหรับการรู้จำคำสั่ง.....	48
ภาพประกอบ 3-18 การรู้จำท่าทางมือสำหรับใช้เป็นชุดคำสั่ง	48
ภาพประกอบ 3-19 ระบบรู้จำท่าทางมือที่ประยุกต์ใช้เพื่อควบคุม โปรแกรมวินโดวมีเดียเซ้นเทอ...49	
ภาพประกอบ 3-20 ท่าทางทั้ง 12 แบบจากตัวอักษรภาษาไทย.....	51
ภาพประกอบ 3-21 ขั้นตอนการทำงานของกระบวนการรู้จำท่าทาง.....	51
ภาพประกอบ 4-1 ตัวอย่างบันทึกข้อมูลของการฝึกสอน	54
ภาพประกอบ 4-2 กราฟ ROC ของการฝึกสอนแต่ละสถานะ	55
ภาพประกอบ 4-3 การตรวจจับลักษณะมือ	56
ภาพประกอบ 4-4 ทดสอบความผิดพลาดในการติดตามการเคลื่อนที่ของมือ.....	58
ภาพประกอบ 4-5 แบบจำลองฮิดเดนมาร์คอฟแบบการเชื่อมโยงทั่วถึง	64
ภาพประกอบ 4-6 มุมท่าทางมือ ก.....	65
ภาพประกอบ 4-7 มุมท่าทางมือ ข	65
ภาพประกอบ 4-8 มุมท่าทางมือ ง	66
ภาพประกอบ 4-9 มุมท่าทางมือ ด	66
ภาพประกอบ 4-10 มุมท่าทางมือ ท	67
ภาพประกอบ 4-11 มุมท่าทางมือ ธ.....	67
ภาพประกอบ 4-12 มุมท่าทางมือ น	68
ภาพประกอบ 4-13 มุมท่าทางมือ ป	68
ภาพประกอบ 4-14 มุมท่าทางมือ พ	69
ภาพประกอบ 4-15 มุมท่าทางมือ ม	69

รายการรูปภาพ

	หน้า
ภาพประกอบ 4-16 มุมทำทางมือ ล.....	70
ภาพประกอบ 4-17 มุมทำทางมือ อ.....	70
ภาพประกอบ 4-18 ส่วนการเปลี่ยนแปลงมุมที่คล้ายกันของทำทางมือ.....	72
ภาพประกอบ 4-19 ทำทางมือ ก ที่ระบบรู้จำเป็นทำทางมือ ค.....	73
ภาพประกอบ 4-20 มุมทำทางมือ ก ที่ระบบรู้จำเป็นทำทางมือ ค	73
ภาพประกอบ 4-21 ทำทางมือ ค ที่ระบบรู้จำเป็นทำทางมือ ค	74
ภาพประกอบ 4-22 มุมทำทางมือ ค ที่ระบบรู้จำเป็นทำทางมือ ค	74
ภาพประกอบ 4-23 ประสิทธิภาพการรู้จำทำทางมือด้วยแบบจำลองฮิดเดนมาร์คอฟ	76

รายการตาราง

หน้า

ตาราง 3-1 แสดงให้เห็นถึงจำนวนคุณลักษณะเด่นทั้งหมดที่สามารถหาได้ภายในภาพกลุ่มตัวอย่างขนาด 24x24 จุดภาพ.....	38
ตาราง 3-2 การแปลงข้อมูลจุดของท่าทาง (ภาพประกอบที่ 3-14(b)) ให้เป็นชุดรหัส	46
ตาราง 3-3 การแยกกลุ่มตัวอักษรภาษาไทยตามลักษณะความคล้ายคลึง	50
ตาราง 4-1 ประสิทธิภาพการตรวจจับลักษณะมือของการฝึกสอนแต่ละสถานะ	54
ตาราง 4-2 การทดสอบความผิดพลาดในการติดตามเคลื่อนที่ของมือ	56
ตาราง 4-3 แสดงจำนวนจุดข้อมูลของท่าทางมือ และค่าประมาณท่าทางมือ.....	60
ตาราง 4-4 ผลการทดลองการรู้จำท่าทางมืออย่างง่าย.....	66
ตาราง 4-5 ประสิทธิภาพการรู้จำท่าทางมือด้วยแบบจำลองฮิดเดนมาร์คอฟ 10 สถานะ	64
ตาราง 4-6 ประสิทธิภาพการรู้จำท่าทางมือด้วยแบบจำลองฮิดเดนมาร์คอฟที่สถานะต่างๆ.....	76

บทที่ 1

บทนำ

1.1. ความสำคัญและที่มาของการวิจัย

สภาพแวดล้อมอัจฉริยะ (Smart Environment) คือ การที่สภาพแวดล้อมที่อยู่รอบตัวเรา ไม่ว่าจะเป็นอุปกรณ์หรือเครื่องใช้ต่างๆ สามารถที่จะตอบสนองต่อความต้องการของมนุษย์ได้อย่างชาญฉลาด เช่น การเปิดปิดอุปกรณ์เครื่องใช้ต่างๆ โดยอัตโนมัติเมื่อมนุษย์ต้องการใช้งาน ซึ่งความชาญฉลาดเหล่านี้ อาจเกิดมาจากการใช้อุปกรณ์แบบต่างๆ ในการควบคุมไม่ว่าจะเป็น การใช้รีโมทคอนโทรลเลอร์ (Remote Controller) อุปกรณ์เซนเซอร์ (Sensor Device) ต่างๆ ที่ช่วยเพิ่มความสะดวกสบายให้กับผู้ใช้ แต่ก็ยังมีข้อจำกัดในการใช้งานอยู่บ้างเช่น อุปกรณ์เซนเซอร์และรีโมทคอนโทรลเลอร์อาจใช้ในการควบคุมได้แค่การเปิดปิด แต่มีวิธีควบคุมอีกรูปแบบหนึ่งสามารถนำมาประยุกต์ใช้กับงานได้ในหลายๆด้าน นั่นก็คือการควบคุมด้วยข้อมูลภาพวิดีโอ (Video-based Controller) ตัวอย่างงานด้านต่างๆ ที่นำข้อมูลภาพวิดีโอเข้ามาใช้ เช่น การจดจำใบหน้า (Face Recognition) การตรวจจับการเคลื่อนไหว (Motion Detection) และการรู้จำท่าทางมือ (Hand Gesture Recognition) แต่ในงานวิจัยชิ้นนี้จะเน้นที่การรู้จำท่าทางมือ ซึ่งการรู้จำท่าทางมือเป็นหัวข้องานวิจัยที่กำลังได้รับความสนใจเป็นอย่างมาก [17][18][19][20] เนื่องจากมือเป็นส่วนที่สามารถแสดงท่าทางต่างๆ ได้มากกว่าส่วนอื่นๆ ของร่างกายและสามารถสื่อความหมายได้อย่างชัดเจน ทำให้ง่ายต่อการรู้จำดังนั้นจึงได้มีการนำท่าทางของมือมาประยุกต์ใช้ในงานหลายๆ ด้าน เช่น การจดจำภาษามือ (Sign Language Recognition) การใช้มือชี้ไปยังที่ต่างๆ บนหน้าจอคอมพิวเตอร์ (Pointing Gesture Recognition) ซึ่งในงานแต่ละด้านก็จะมีการออกแบบท่าทางของมือที่แตกต่างกัน เพื่อให้เหมาะสมกับงานนั้นๆ

งานวิจัยชิ้นนี้เป็นความรู้จำคำสั่งจากท่าทางมือเพื่อใช้ในระบบเฟียร์ะวังผ่านกล้องวงจรปิด โดยจะเน้นไปที่ลักษณะการเคลื่อนที่ของมือในรูปแบบต่างๆ เพื่อใช้กำหนดเป็นชุดคำสั่ง เช่น การเคลื่อนมือไปทางซ้าย ทางขวา เป็นวงกลม หรือสี่เหลี่ยม เป็นต้น โดยสามารถนำไปประยุกต์ใช้งานจริงภายในบ้านหรืออาคารสำนักงานต่างๆ ที่ติดระบบกล้องวงจรปิดได้

1.2. งานวิจัยที่เกี่ยวข้อง

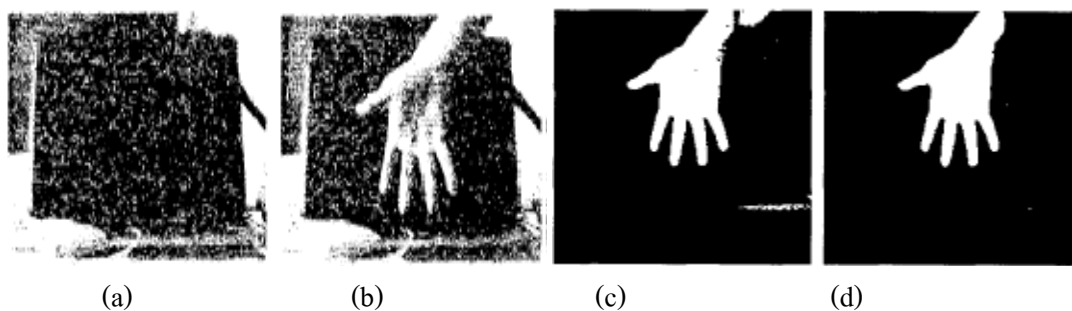
การค้นหาลักษณะของมือและติดตาม (Hand Detection and Tracking) การรู้จำท่าทางต่างๆ ของมือจากข้อมูลภาพวิดีโอเป็นหัวข้องานวิจัยที่กำลังได้รับความสนใจเป็นอย่างมาก เนื่องจากทำ

ให้ผู้ใช้สามารถโต้ตอบกับคอมพิวเตอร์ได้อย่างเป็นธรรมชาติ โดยที่ไม่จำเป็นต้องมีอุปกรณ์ภายนอกอื่นๆ เข้ามาเกี่ยวข้อง ซึ่งจากการศึกษางานวิจัยที่เกี่ยวข้อง จะแบ่งได้เป็น 2 แบบ คือ

- 1). การรู้จำรูปร่างของมือ (Hand Posture Recognition) จะเน้นไปที่การจดจำรูปร่างลักษณะของมือโดยใช้การ กำมือ แบนมือ หรือการชูนิ้วมือในลักษณะต่างๆ เพื่อกำหนดเป็นชุดคำสั่งที่เหมาะสมสำหรับโปรแกรมต่างๆ ซึ่งจะต้องมีภาพของมือที่ใหญ่เพียงพอเพื่อให้สามารถหาคุณลักษณะเช่น ปลายนิ้ว ง่ามนิ้ว หรือฝ่ามือ เพื่อใช้สำหรับการรู้จำท่าทางต่างๆ ของมือได้
- 2). การรู้จำท่าทางมือ จะเน้นไปที่การจดจำทิศทาง การเคลื่อนที่มือในรูปแบบ ต่างๆ เช่น เคลื่อนมือไปทางซ้าย ทางขวา หรือขึ้นลงซึ่งจะมีคุณลักษณะเช่น ทิศทาง มุม หรือความเร็วที่จะนำเข้ามาใช้ในการจดจำท่าทางของมือ

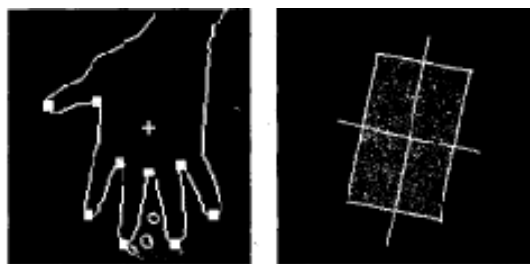
โดยวิธีการในการรู้จำลักษณะท่าทางต่างๆ ของมือ อาจใช้วิธีการอย่างเช่น แบบจำลองฮิดเดนมาร์คอฟ (Hidden Markov Model) นิวรอลเน็ตเวิร์ค (Neural Network) หรือสร้างแบบจำลองการรู้จำขึ้นมาเอง เพื่อใช้เป็นต้นแบบของท่าทางในลักษณะต่างๆ ที่จะนำมาเปรียบเทียบกับท่าทางต่างๆ ที่เกิดขึ้นภายในภาพวิดีโอ

Wei Du และคณะ [1] ได้เสนองานวิจัยเกี่ยวกับการรู้จำรูปร่างของมือ โดยใช้ตัวกรองสีผิวหนัง (Skin Color Filter) ในการดึงจุดสีที่เป็นค่าสีเป็นผิวหนังออกมา โดยจุดสีที่เป็นส่วนของผิวหนังจะถูกกำหนดค่าสีเป็น 1 ส่วนที่เหลือมีค่าสีเป็น 0 ภาพที่ออกมาจะได้เป็นภาพขาวดำ (Binary Image) แต่ยังคงต้องใช้การกรองภาพแบบมีเดียเนียนฟิลเตอร์ (Median Filter) เพื่อทำการลดสัญญาณรบกวนต่างๆ ที่ยังคงมีอยู่



ภาพประกอบ 1-1 ผลลัพธ์ของการแยกค่าจุดสี (a) ภาพพื้นหลังที่ยังไม่มีมือ (b) ภาพอินพุตพร้อมด้วยมือ (c) ภาพหลังการกรองสีผิวหนัง (d) ผลลัพธ์สุดท้ายหลังการทำมีเดียเนียนฟิลเตอร์

คุณลักษณะของมือ 2 ชนิดจะถูกดึงออกมาจากภาพขาวดำที่ได้ อย่างแรกคือ คุณลักษณะทางด้านข้อมูล (Statistic-based Feature) ใช้หา จุดศูนย์กลางมวล (Center of Gravity), แกนหลักของมือ (Primary Axis) คุณลักษณะที่สอง คือ คุณลักษณะทางด้านโครงร่าง (Contour-based Feature) ใช้สำหรับหาเส้นขอบของมือและนำจุดที่ได้ของเส้นขอบมาหาปลายนิ้ว และ โคนนิ้ว โดยวัดค่าความโค้ง (K-curvature) ของแต่ละจุด



(a)

(b)

ภาพประกอบ 1-2 คุณลักษณะ 2 ชนิดที่ถูกดึงออกมา (a) คุณลักษณะทางด้านโครงร่าง
(b) คุณลักษณะทางด้านข้อมูล

ซึ่งแต่ละท่าทางที่จะใช้จดจำจะถูกกำหนดเงื่อนไขขึ้นมาโดยใช้คุณลักษณะที่หามาได้ เพื่อแยกแยะความแตกต่างระหว่างท่าทาง จากนั้นจึงใช้ไฟไนต์สเตตแมชชีน (Finite State Machine) โดยออกแบบการทำงานของระบบเป็นหลายๆ สถานะเพื่อใช้ในการจดจำคำสั่งท่าทางต่างๆ ตามที่ได้ออกแบบไว้

Qing Chen และคณะ [2] ได้นำเสนองานวิจัยเกี่ยวกับการรู้จำรูปร่างของมือ โดยใช้คุณลักษณะของฮาร์ไลท (Haar-like Feature) ในการหาลักษณะเด่นของวัตถุที่เราสนใจภายในภาพ การที่จะได้ลักษณะเด่นของวัตถุที่เราสนใจ จำเป็นต้องมีกลุ่มตัวอย่างภาพ ที่ภายในภาพมีวัตถุที่เราสนใจอยู่ (Positive Sample) และภาพอื่นๆ ที่ไม่ใช่ (Negative Sample) มาทำการฝึกสอนเพื่อให้ได้ลักษณะเด่นของวัตถุนั้นๆ ที่จะนำมาใช้ในการรู้จำ



(a)

(b)

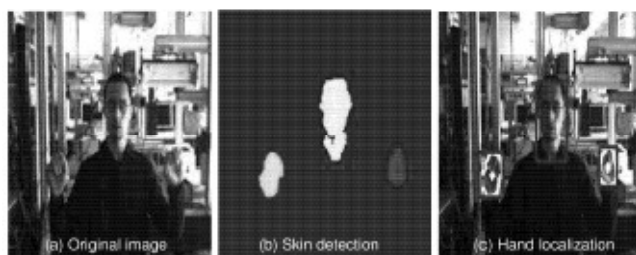
ภาพประกอบ 1-3 กลุ่มตัวอย่างภาพ (a) ภาพตัวอย่างที่ใช่ (b) ภาพตัวอย่างที่ไม่ใช่

เมื่อต้องการจะหาว่าภายในข้อมูลภาพอื่นๆ มีวัตถุที่เราสนใจอยู่หรือไม่ ก็จะนำข้อมูลภาพเหล่านั้นมาเปรียบเทียบกับคุณลักษณะเฉพาะที่หามาได้จากกลุ่มภาพตัวอย่างที่ได้ทำการสอนเอาไว้แล้วเพื่อใช้ในการตรวจจับวัตถุที่กำลังสนใจภายในภาพ



ภาพประกอบ 1-4 ผลลัพธ์ของการรู้จำรูปร่างมือโดยใช้คุณลักษณะแบบฮาร์ไลท

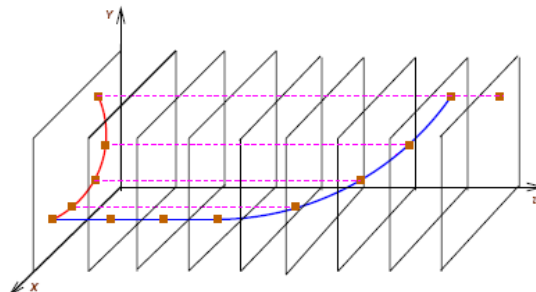
Mahmoud Elmezain และคณะ [3] ได้เสนองานวิจัยที่เกี่ยวข้องกับการแปลความหมายของทิศทางการเคลื่อนที่มือเพื่อใช้แสดงเป็นตัวอักษร โดยตัวกรองสีผิวหนังแยกแยะว่าจุดสีใดเป็นผิวหนังหรือไม่ใช่ผิวหนัง ความลึกของจุดสีที่ถูกแยกแยะออกมาจะช่วยลดปัญหาการซ้อนทับกันระหว่างมือและใบหน้า เนื่องจากพื้นที่ของมือจะอยู่ใกล้กับตัวกล้องมากกว่าพื้นที่ของใบหน้า เพื่อเป็นการกำจัดสัญญาณรบกวนต่างๆ จึงได้ใช้กระบวนการเปลี่ยนแปลงลักษณะรูปร่างหรือโครงสร้างของภาพ (Morphological) เข้ามาช่วยเนื่องจากยังมีพื้นที่เล็กที่อยู่ใกล้บริเวณผิวหนังแต่ไม่ใช่ส่วนที่เป็นผิวหนังเกิดขึ้นอยู่



ภาพประกอบ 1-5 การแยกจุดสีที่เป็นผิวหนังและการระบุตำแหน่งมือ

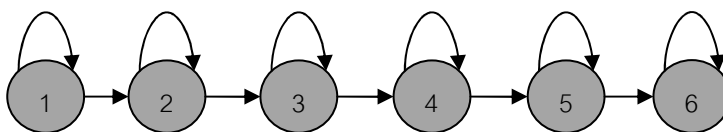
การระบุตำแหน่งของมือทั้ง 2 จะถูกพบโดยการเลือกพื้นที่เล็ก 2 พื้นที่ โดยที่ใบหน้าจะถูกแสดงเป็นพื้นที่ใหญ่ๆ และอยู่ห่างจากกล้อง นอกจากนี้ยังใช้การวิเคราะห์กลุ่มของจุดภาพที่เป็นขาวดำ (BLOB Analysis) เพื่อใช้ในการคำนวณหาตำแหน่งและขนาดมือ โดยจะนำข้อมูลที่ได้นำไปใช้กับภาพที่ได้ในเฟรมถัดไปเพื่อทำให้เกิดการติดตาม ซึ่งจะนำไปกำหนดเป็นท่าทางในลักษณะต่างๆ คุณลักษณะ 3 อย่างที่จะนำมาใช้คือ ตำแหน่ง ทิศทาง และความเร็ว โดยที่เส้นทางของการเคลื่อนที่

ของมือ (Gesture Path) จะเป็นตำแหน่งของมือในทุกๆ เฟรมภาพตามช่วงเวลา (t) ที่ฉายบนระนาบ xy (Spatio-temporal Space)



ภาพประกอบ 1-6 ตำแหน่งของมือในทุกๆ เฟรมภาพตามช่วงเวลา (t)

งานวิจัยนี้ใช้คุณลักษณะของทิศทางเป็นหลัก จึงได้มีการคำนวณมุมและทิศทางจากจุดศูนย์กลางมวล (Center of Gravity) ของเส้นทางการเคลื่อนที่ (Gesture Path) ซึ่งจะนำไปใช้ในแบบจำลองฮิดเดนมาร์คอฟโดยมีการกำหนด สถานะของในฮิดเดนมาร์คอฟโมเดลเป็นแบบลำดับชั้นซ้าย-ขวา (Left-Right Banded) คือแต่ละสถานะสามารถย้อนกลับไปยังสถานะเดิมได้หรือเปลี่ยนสถานะไปตามลำดับ



ภาพประกอบ 1-7 แบบจำลองฮิดเดนมาร์คอฟประเภทลำดับชั้นซ้าย-ขวา

ใน HMM ประกอบไปด้วยตัวแปรต่างๆ ดังนี้ คือ ความน่าจะเป็นในการเปลี่ยนสถานะจากสถานะ i ไปสถานะ j (A), ค่าความน่าจะเป็นของเหตุการณ์ j ที่เกิดขึ้นในสถานะ i (B) และ π ความน่าจะเป็นที่จะเกิดสถานะแรก ส่วนประกอบต่าง ๆ ของโมเดลสามารถรวมเข้าด้วยกัน แล้วแทน ด้วย $\lambda = (A, B, \pi)$ ที่แสดงเป็นเซตของตัวแปรที่เสร็จสมบูรณ์ของโมเดล เมื่อกำหนดโมเดลเริ่มแรก λ ซึ่งประกอบด้วย (A, B, π) จากนั้นก็เข้าสู่กระบวนการประมาณตัวแปร (Parameter Estimation) ซึ่ง ใช้วิธีของ Baum-Welch เพื่อทำการปรับโมเดลจนกระทั่งได้โมเดลที่ดีที่สุดสำหรับทำทางนั้นๆ

นอกจากนี้ยังมีงานวิจัยอื่นๆ ที่เกี่ยวข้องอีก ยกตัวอย่างเช่น Segan และคณะ [4] อธิบายระบบด้วยกล้อง 2 กล้องที่สามารถจดจำท่าทางและการติดตามตำแหน่งของมือใน 3 มิติ ระบบจะตรวจจับนิ้วมือโดยการดึงเอาคุณลักษณะของจุดบนโครงร่างของมือเพื่อกำหนดเป็นท่าทางในลักษณะต่างๆ Utsumi และคณะ [5] อธิบายระบบการจดจำท่าทางมือโดยใช้กล้อง 4 กล้อง ซึ่งภายในระบบแบบจำลองของมืออย่างง่ายถูกสร้างขึ้นจากตำแหน่ง ทิศทาง และมุมของมือ Quek

และคณะ [6] อธิบายระบบที่สามารถใช้แทนเมาส์ด้วยท่าทางการเคลื่อนที่ของมือ (Finger Mouse) Triesch และคณะ [7] นำเสนออัลกอริทึมที่ใช้ในการจดจำท่าทางมือโดยใช้คุณลักษณะต่างๆ เช่น การเคลื่อนไหว สี และภาพ 3 มิติ เพื่อใช้กับการหยิบจับวัตถุของหุ่นยนต์

1.3. วัตถุประสงค์

- 1.3.1. พัฒนาวิธีการค้นหาและติดตามตำแหน่งของมือจากภาพวิดีโอได้อย่างถูกต้อง
- 1.3.2. พัฒนาโมเดลการรู้จำลักษณะและท่าทางของมือ
- 1.3.3. พัฒนาชุดคำสั่งโดยใช้ท่าทางของมือในลักษณะต่างๆ เพื่อใช้ในระบบเฟืองระวางผ่านกล้องวิดีโอวงจรปิด

1.4. ขอบเขตงานวิจัย

งานวิจัยนี้จะมุ่งเน้นไปยัง 3 ส่วน ดังต่อไปนี้

- 1.4.1. การค้นหาและติดตามตำแหน่งของมือ เป็นการค้นหาตำแหน่งของมือและทำการติดตามจากภาพวิดีโอ โดยผู้ใช้จะแบ่มือข้างเดียวเข้าหากล้อง ซึ่งกล้องที่ใช้จับภาพจะอยู่ห่างจากตำแหน่งของผู้ใช้งานเป็นระยะพอสมควร
- 1.4.2. การรู้จำท่าทางมือ เป็นการดึงคุณลักษณะเฉพาะของท่าทาง เช่น ตำแหน่ง หรือทิศทางของการเคลื่อนที่ของมือ เพื่อใช้แบมือจำลองการรู้จำลักษณะ และท่าทางของมือ
- 1.4.3. การรู้จำชุดคำสั่งด้วยท่าทางมือ เป็นการสร้างชุดคำสั่งด้วยท่าทางต่างๆ ของมือ โดยจะเน้นไปที่การเคลื่อนที่ของมือ ได้แก่ เคลื่อนมือไปทางซ้าย ทางขวา ขึ้นบน ลงล่าง เคลื่อนมือเป็นวงกลม สามเหลี่ยม หรือสี่เหลี่ยม เป็นต้น

ซึ่งในขั้นตอนนี้จะเป็นการรู้จำเพียงอย่างเดียวคือ เมื่อผู้ใช้แสดงท่าทางมือตามชุดคำสั่งในลักษณะต่างๆ ระบบสามารถบอกได้อย่างถูกต้องว่าผู้ใช้งานกำลังออกคำสั่งอยู่ แต่คำสั่งเหล่านั้นจะใช้ควบคุมอะไรก็ขึ้นอยู่กับนำไปประยุกต์ใช้ในภายหลัง

1.5. ประโยชน์ที่คาดว่าจะได้รับ

ได้วิธีการค้นหาตำแหน่งและการติดตามมือ พร้อมทั้งโมเดลการรู้จำชุดคำสั่งโดยใช้ท่าทางมือในลักษณะต่างๆ เพื่อใช้กับระบบเฟืองระวางผ่านกล้องวิดีโอวงจรปิดที่ติดตั้งอยู่ภายในบ้านหรืออาคารสำนักงานได้

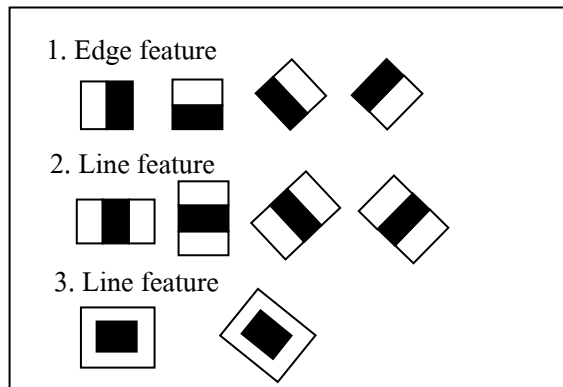
บทที่ 2

ทฤษฎีและหลักการ

2.1. กระบวนการตรวจจับวัตถุที่สนใจ (Object Detection)

2.1.1. คุณลักษณะเด่นแบบฮาไลท์ (Haar-like Feature)

คุณลักษณะเด่นแบบฮาไลท์ ถูกนำเสนอครั้งแรกโดย Viola และคณะ [8] แล้วได้ถูกพัฒนาต่อโดย Rainer และคณะ [9] คุณลักษณะเด่น (Feature) จะประกอบไปด้วยสี่เหลี่ยม 2-3 ชั้น (สีดำและสีขาว) ที่อยู่ติดกันและสามารถเปลี่ยนแปลงขนาดและตำแหน่งภายในภาพได้ โดยค่าของคุณลักษณะเด่นก็คือความแตกต่างระหว่างผลรวมของค่าจุดสีภายในพื้นที่สีขาวและสีดำ



ภาพประกอบ 2-1 คุณลักษณะเด่นแบบฮาไลท์สำหรับการตรวจจับในลักษณะแบบต่างๆ

การคำนวณค่าของคุณลักษณะเด่นนั้น ใช้หลักการคำนวณภาพแบบอินทิกรัล (Integral Image) ซึ่งทำให้การคำนวณค่าคุณลักษณะเด่นนั้นทำได้เร็วมาก โดยการคำนวณภาพแบบอินทิกรัลที่ตำแหน่ง (x, y) ใดๆ คือผลรวมของค่าในทุกๆ จุดสีจนไปถึงตำแหน่ง (x, y) นั้นๆ

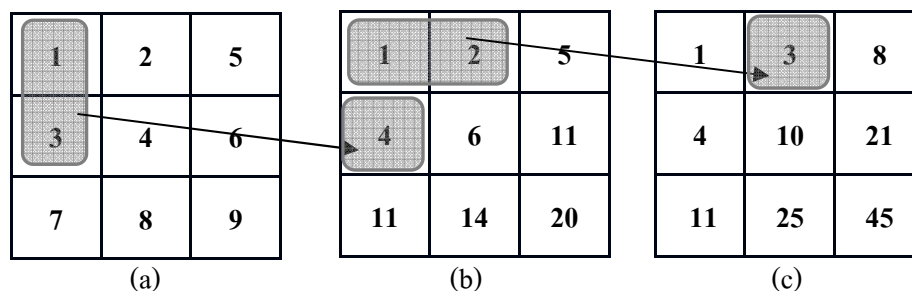
$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (2.1)$$

ซึ่ง $ii(x, y)$ คือการคำนวณภาพแบบอินทิกรัล และ $i(x, y)$ คือภาพต้นฉบับ โดยใช้วิธีการคำนวณดังสมการที่ 2.2 และ 2.3

$$S(x, y) = S(x, y-1) + i(x, y) \quad (2.2)$$

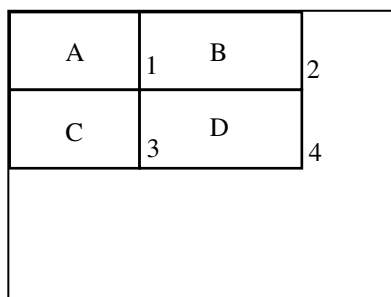
$$ii(x, y) = ii(x-1, y) + S(x, y) \quad (2.3)$$

โดยที่ $S(x, y)$ คือค่าผลรวมค่าจุดสีสะสม ส่วน $S(x, -1) = 0$ และ $ii(-1, y) = 0$ (ภาพประกอบ 2-1)



ภาพประกอบ 2-2 วิธีการคำนวณภาพแบบอินทิกรัล (a) ภาพต้นฉบับ $(i(x, y))$ (b) ผลรวมค่าจุดสีในแนวแกน y ($S(x, y)$) (c) ผลรวมค่าจุดสีในแนวแกน x ($ii(x, y)$)

การใช้การคำนวณภาพแบบอินทิกรัลทำให้สามารถคำนวณหาผลรวมค่าจุดสีที่ตำแหน่ง (x, y) ใดๆ ได้โดยใช้ตำแหน่งอ้างอิง 4 ตำแหน่ง การผลรวมค่าจุดสีในสี่เหลี่ยม D สามารถคำนวณได้โดยใช้ตำแหน่งอ้างอิง 4 ตำแหน่ง ค่าของการคำนวณภาพแบบอินทิกรัลในตำแหน่งที่ 1 ก็คือผลรวมค่าจุดสีในสี่เหลี่ยม A ค่าในตำแหน่งที่ 2 คือ $A+B$ ในตำแหน่งที่ 3 คือ $A+C$ และ ตำแหน่งที่ 4 คือ $A+B+C+D$ ซึ่งผลรวมค่าจุดสีในสี่เหลี่ยม D คือ $4-3-2+1$ (ภาพประกอบ 2-3) ดังนั้นถ้าต้องการหาผลต่างของค่าผลรวมจุดสีของพื้นที่สี่เหลี่ยม 2 ชั้นที่อยู่ติดกันสามารถทำได้โดยใช้ตำแหน่งอ้างอิง 6 ตำแหน่ง ใช้ตำแหน่งอ้างอิง 8 ตำแหน่งสำหรับพื้นที่สี่เหลี่ยม 3 ชั้นที่อยู่ติดกัน และใช้ตำแหน่งอ้างอิง 9 ตำแหน่งสำหรับพื้นที่สี่เหลี่ยม 4 ชั้นที่อยู่ติดกัน



ภาพประกอบ 2-3 การคำนวณหาผลรวมค่าจุดสีในสี่เหลี่ยม D

2.1.2. อัลกอริธึมในการเรียนรู้แบบอดาบูสท (AdaBoost learning algorithm)

หลักการของอดาบูสท (AdaBoost) [8] เป็นกระบวนการหาคุณลักษณะเด่นที่มีลักษณะใกล้เคียงและแตกต่างกับภาพนำเข้า ซึ่งมีสองลักษณะคือ ภาพที่มีวัตถุต่างๆ ที่สนใจประกอบอยู่ในภาพ และภาพใดๆที่ไม่มีวัตถุที่สนใจอยู่ในภาพ โดยมีการถ่วงน้ำหนักให้คุณลักษณะเด่นต่างๆ ตัวภายในภาพ ทั้งบนภาพทั้ง 2 แบบเพื่อใช้หาลักษณะของวัตถุที่ “ใช่” และ “ไม่ใช่” ในลักษณะต่างๆ มีกระบวนการดังนี้

- กำหนดภาพตัวอย่าง $(x_1, y_1), \dots, (x_n, y_n)$ โดยที่ $y_i = 0, 1$ สำหรับภาพอื่นๆ และภาพที่มีวัตถุที่สนใจอยู่ตามลำดับ
- กำหนดค่าน้ำหนักเริ่มต้น $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ สำหรับ $y_i = 0, 1$ โดยที่ m และ l คือ จำนวนของภาพอื่นๆ และ ตามลำดับ
- ที่เวลา $t = 1, \dots, T$

1. ทำค่าน้ำหนักให้อยู่ในรูปแบบที่เป็นบรรทัดฐานเดียวกัน (Normalize)

$$w_{1,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$

w_t คือ การแจกแจงความน่าจะเป็น (Probability distribution)

2. สำหรับค่าของแต่ละคุณลักษณะ ($f_j(x)$) จะมีฟังก์ชันการแยกประเภทแบบอ่อน (Weak Classifier : $h_j(x)$) และความผิดพลาด (ϵ_j) ขึ้นอยู่กับค่าน้ำหนักที่ถ่วงไว้ w_t

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases} \quad (2.4)$$

$$\epsilon_j = \sum_i w_i |h_j(x_i) - y_i| \quad (2.5)$$

3. เลือกเอาเฉพาะคุณลักษณะเด่นที่มีค่าความผิดพลาดน้อยที่สุด
4. ปรับค่าน้ำหนัก

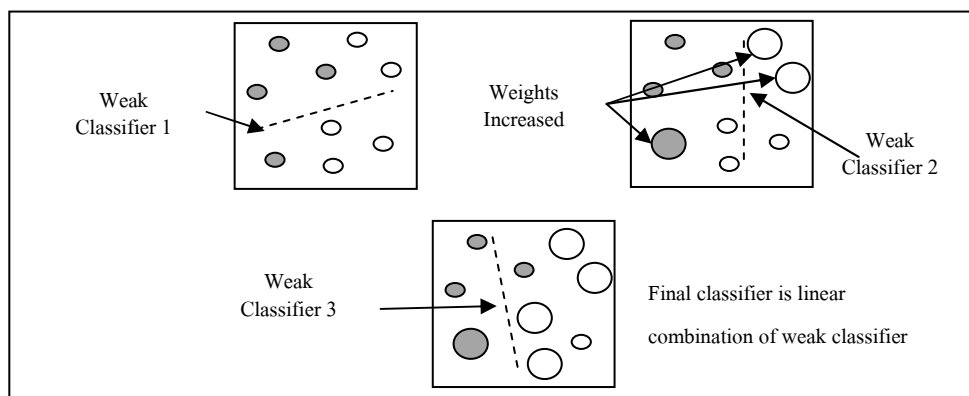
$$w_{t+1,i} = w_{t,i} \beta_t^{1-\epsilon_i} \quad (2.6)$$

โดยที่ $\epsilon_i = 0$ ถ้าตัวอย่าง x_i ถูกแยกประเภทได้อย่างถูกต้อง

ถ้าไม่เช่นนั้น $\epsilon_i = 1$ และ $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$

- ฟังก์ชันสำหรับการแยกประเภทขั้นสุดท้ายแบบแข็งแกร่ง (Strong Classifier)

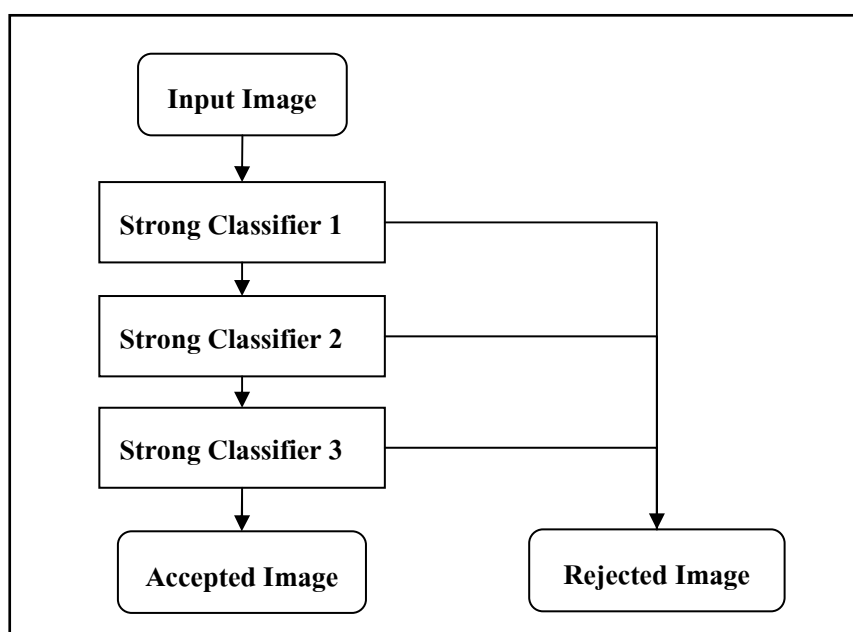
$$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases}, \quad \alpha_t = \log \frac{1}{\beta_t} \quad (2.7)$$



ภาพประกอบ 2-4 การทำงานอดามูสท (AdaBoost)

2.1.3. การแยกประเภทแบบลำดับขั้น (Cascade Classifiers)

การแยกประเภทแบบลำดับขั้น[8] (Cascade Classifiers) เป็นกระบวนการตรวจจับวัตถุที่เราสนใจภายในภาพ โดยเริ่มต้นจากการตัดส่วนของภาพออกเป็นภาพย่อยๆ (Sub Image) และนำแต่ละภาพย่อยๆ ไปเปรียบเทียบกับคุณลักษณะเด่นที่หามาได้ในกระบวนการเรียนรู้ โดยคุณลักษณะเด่นจะถูกจัดไว้เป็นลำดับขั้น ถ้าภาพย่อยๆ ที่ถูกนำไปเปรียบเทียบกับคุณลักษณะเด่นในลำดับใดๆ แสดงผลว่าไม่ใช่วัตถุที่เราสนใจก็จะถูกตัดทิ้งไปโดยไม่นำไปเปรียบเทียบกับลำดับถัดไป แต่ถ้าภาพย่อยๆ ใดสามารถผ่านการเปรียบเทียบได้ทุกลำดับขั้นก็จะแสดงผลว่าภาพนั้นน่าจะเป็นส่วนของวัตถุที่เราสนใจ (ภาพประกอบ 2-5)



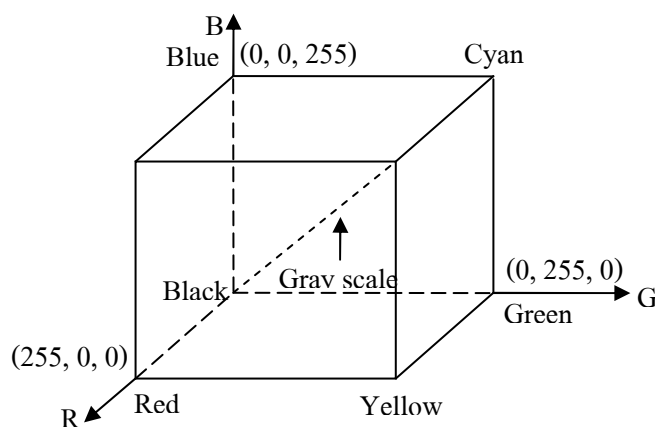
ภาพประกอบ 2-5 การทำงานของการแยกประเภทแบบลำดับขั้น (Cascade Classifiers)

2.2. กระบวนการติดตามวัตถุเคลื่อนไหว

2.2.1. คุณสมบัติสีที่ใช้ในกระบวนการติดตาม

ในกระบวนการแบ่งส่วนวัตถุโดยอาศัยคุณสมบัติด้านสีของวัตถุ เพื่อความถูกต้องของกระบวนการจึงจำเป็นที่จะต้องมีการพิจารณาคูสมบัตินี้ของสีที่มีความสัมพันธ์กับแสงสว่างซึ่งระบบสี RGB [10] นั้นไม่สามารถอธิบายได้ ดังนั้นระบบสี HSV [10] ซึ่งเป็นระบบสีที่เหมาะสมสำหรับการถ่วงน้ำหนักของสีของวัตถุ

ในระบบสี RGB แต่ละสีจะแสดงในรูปของส่วนผสมระหว่างสี แดง เขียว และน้ำเงิน ระบบสีนี้มีพื้นฐานอยู่บนการระบุค่าเฉพาะแต่ละจุดด้วยคู่ของตัวเลขระยะพิคค ซึ่งค่าสี RGB ต่าง ๆ สามารถแสดงตัวอย่างในภาพประกอบ 2-6



ภาพประกอบ 2-6 ลูกบาศก์ระบบสี RGB

การแสดงผลภาพสี ในระบบสี RGB จะพิจารณาสีหลักแต่ละสี คือ แดง เขียวและน้ำเงิน ถ้าหากให้ความละเอียดของสีหลักแต่ละสีขนาด 8 บิต รวม 3 สี จึงใช้ข้อมูลขนาด 24 บิต จำนวนสีทั้งหมด จึงมีจำนวน เท่ากับ $(2^8)^3 = 16,777,216$ สี

ระบบสี RGB นั้นเหมาะกับการใช้งานในฮาร์ดแวร์ และยังเข้ากันได้ดีกับการมองเห็นสีหลักของตามนุษย์ คือ สีแดง เขียวและน้ำเงิน แต่น่าเสียดายที่ระบบสี RGB ไม่เหมาะสมกับการอธิบายสีในลักษณะที่มีความสัมพันธ์กับค่าความสว่าง ในระบบสี HSV นั้นจะประกอบด้วย 3 ส่วน คือ ค่าเฉดสี (Hue) ค่าความบริสุทธิ์ของสี (Saturate) และ ค่าความสว่าง (Value) โดยค่าเฉดสีคือคุณสมบัติที่อธิบายถึงสีบริสุทธิ์ เช่น สีเหลืองบริสุทธิ์ สีแดงบริสุทธิ์ ส่วน ค่าความบริสุทธิ์ของสีอธิบายถึงความเข้มของสีที่ถูกผสมด้วยแสงขาว และ ค่าความสว่าง คือ คุณสมบัติของสี ที่นิยามในลักษณะของการมองเห็นของมนุษย์ต่อการสะท้อนแสงของตัววัตถุ ซึ่งจะมีค่าอยู่บนแกนความสว่างและความมืด ระบบสี HSV สามารถแสดงตัวอย่างได้ในภาพประกอบ 2-11

การแปลงค่าสี RGB เป็น HSV

เมื่อกำหนดค่าสีในระบบ RGB สามารถแปลงเป็นระบบสี HSV ได้ดังนี้

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases} \quad (2.8)$$

$$\theta = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2 + (R-B)(G-B)\frac{1}{2}}} \right\} \quad (2.9)$$

ความบริสุทธิ์ของสี (S) สามารถคำนวณได้โดย

$$S = 1 - \frac{1}{(R+G+B)} [\min(R, G, B)] \quad (2.10)$$

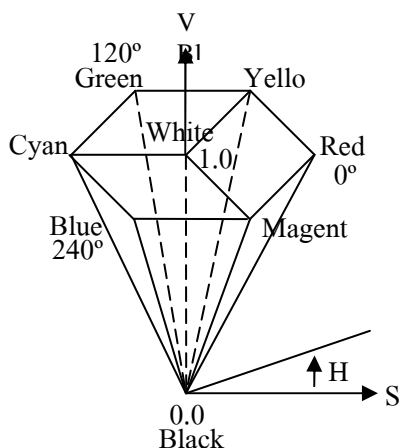
และสุดท้าย ความสว่าง (V) สามารถคำนวณได้โดย

$$V = \frac{1}{3} (R + G + B) \quad (2.11)$$

ค่าสี (H) สามารถคำนวณได้จากค่าสีในระบบ RGB ดังสมการ (2.8) โดยค่าจะอยู่ในรูปองศาระหว่าง 0° ถึง 360° ดังในภาพประกอบ 2-11 แต่ละองศาที่เปลี่ยนไปจะแสดงถึงสีที่ค่อย ๆ เปลี่ยนไปสู่อีกสีหนึ่ง ซึ่งสามารถจำแนกออกเป็นสีหลักได้ 6 สี ได้แก่ สีแดง สีเหลือง สีเขียว สีน้ำเงินเขียว สีน้ำเงิน และ สีแดงม่วง

ความบริสุทธิ์ของสี (S) คำนวณได้จากค่าสีในระบบ RGB ดังสมการ (2.10) โดยความบริสุทธิ์ของสีจะมีความสัมพันธ์กับค่าความสว่าง (V) ด้วย ซึ่งเป็นไปตามสามัญสำนึก คือ ในกรณีที่มีแสงสว่างปรกติสายตามนุษย์จะเห็นสีของวัตถุตามความเป็นจริงของสีวัตถุนั้น ๆ (ถ้าวัตถุนั้น ๆ ไม่เกิดแสงจ้าที่พื้นผิวอันเนื่องมาจากคุณสมบัติการสะท้อนแสง) และในกรณีที่มีแสงน้อย (V มีค่าน้อย) ยังมีความสว่างน้อยสีของวัตถุก็ย่อมมีลดลงเช่นเดียวกัน

ความสว่าง (V) คำนวณได้จากค่าสีในระบบ RGB ดังสมการ (2.11) โดยแกน V นั้นจะมีค่าระหว่างสีดำ สีเทา ถึง สีขาวเป็นลักษณะมาตราส่วนสีเทา (Gray Scale) ในสภาพแวดล้อมที่มีแสงน้อยหรือวัตถุมีสีใกล้เคียงสีดำ จะทำให้ V มีค่าน้อยลง สีก็จะเข้าใกล้สีดำมากขึ้น และที่ V มีค่าสูงสุดจะหมายถึงสีขาว



ภาพประกอบ 2-7 ระบบสี HSV

2.2.2. กระบวนการติดตามโดยการย้ายไปยังค่าเฉลี่ย (MeanShift Tracking)

ภายในกระบวนการติดตามวัตถุ โดยอาศัยคุณสมบัติสีของวัตถุ ค่าของกลุ่มสีวัตถุที่ต้องการติดตามนั้นจะอยู่ในรูปของจุดข้อมูลที่มีการกระจายตัวอยู่ภายในภาพและขั้นตอนวิธีการเคลื่อนย้ายบริเวณที่สนใจให้ลู่เข้าสู่ค่าเฉลี่ย เป็นวิธีที่ใช้เพื่อค้นหาตำแหน่งกลุ่มข้อมูลที่มีการกระจุกตัวกันอย่างหนาแน่น ภายในกลุ่มข้อมูลที่กระจายอย่างกว้างขวาง ดังนั้นวิธีการนี้จึงสามารถนำมาประยุกต์ใช้เพื่อติดตามวัตถุที่มีสีที่เราสนใจได้

วิธีการนี้ได้ถูกนำมาประยุกต์ใช้กับ กราฟแสดงความหนาแน่น (Density Histogram) เพื่อใช้หาตำแหน่งวัตถุตามคุณลักษณะสีที่ต้องการติดตาม ซึ่งถูกใช้เพื่อติดตามวัตถุเคลื่อนไหวภายในภาพวิดีโอ การย้ายไปยังค่าเฉลี่ย ได้ประยุกต์ใช้กระบวนการทางสถิติ ซึ่งจะทำให้ จุดข้อมูลที่กระจายตัวอยู่ห่างจากกลุ่มข้อมูลที่มีความหนาแน่นถูกละทิ้งและค่อย ๆ เลื่อนหน้าต่างค้นหา (Search Window) ลู่เข้าสู่บริเวณกลุ่มข้อมูลที่มีความหนาแน่น

กระบวนการย้ายไปยังค่าเฉลี่ย จะเกี่ยวข้องกับเคอร์เนลการประมาณค่าความหนาแน่นของสีที่ต้องการติดตาม ซึ่งเคอร์เนลจะหมายถึงฟังก์ชันที่เจาะจงการคำนวณเฉพาะบางพื้นที่ โดยกระบวนการคำนวณมินชิฟ (Mean-Shift) จะทำให้เคอร์เนลค่อย ๆ เปลี่ยนตำแหน่งเข้าสู่กลุ่มข้อมูลที่มีความหนาแน่นสูง และเมื่อการเปลี่ยนตำแหน่งใกล้เคียงศูนย์แสดงว่าบริเวณนั้นเป็นบริเวณที่ข้อมูลมีความหนาแน่นสูง

ภาพประกอบ 2-12 แสดงสมการที่ได้รวมขั้นตอนวิธีการย้ายไปยังค่าเฉลี่ย สมการสามารถมองให้อยู่ในรูปอย่างง่ายได้โดยพิจารณาที่สมการในบริเวณสี่เหลี่ยมแรเงาและการลดลงของสมการมินชิฟเวกเตอร์ (Mean-shift Vector Equation) เพื่อจะทำการคำนวณศูนย์กลางมวลของการกระจายตัวของจุดภาพ เราสามารถคำนวณหาจุดศูนย์กลางมวล (x_c, y_c) ได้โดย

$$x_c = \frac{M_{10}}{M_{00}}, y_c = \frac{M_{01}}{M_{00}} \quad (2.12)$$

โมเมนต์ลำดับที่ ศูนย์ สามารถคำนวณได้โดย

$$M_{00} = \sum_x \sum_y I(x,y) \quad (2.13)$$

และโมเมนต์ลำดับที่ หนึ่ง สามารถคำนวณได้โดย

$$M_{10} = \sum_x \sum_y xI(x,y) \quad (2.14)$$

$$M_{01} = \sum_x \sum_y yI(x,y) \quad (2.15)$$

กำหนดให้

$I(x,y)$ คือ ค่าสี ณ จุดภาพที่พิกัด(x,y)

เริ่มต้นโดยคำนวณ $K(x - x_i) = ck \left[\left\| \frac{x - x_i}{h} \right\|^2 \right]$ เป็นค่าประมาณของความน่าจะเป็น

การกระจายของ $P(x) = \frac{1}{n} \sum_{i=1}^n K(x - x_i)$

เกรเดียนต์ $\nabla P(x) = \sum_{i=1}^n \nabla K(x - x_i)$

กำหนดให้ : $g(x) = -k'(x)$ เป็นอนุพันธ์ของ $K(x - x_i)$ ดังนั้น เราจะได้ สมการ

$$\nabla P(x) = \frac{c}{n} \sum_{i=1}^n \nabla K_i = \frac{c}{n} \left[\sum_{i=1}^n g_i \left[\frac{\|x - x_i\|^2}{h} \right] \right] \left[\frac{\sum_{i=1}^n x_i g_i \left[\frac{\|x - x_i\|^2}{h} \right]}{\sum_{i=1}^n g_i \left[\frac{\|x - x_i\|^2}{h} \right]} \right] x$$

ภาพประกอบ 2-8 กระบวนการย้ายไปยังค่าเฉลี่ย

จากสมการมินชิฟเวกเตอร์ในกรณีนี้แสดงให้เห็นถึงการเปลี่ยนตำแหน่งศูนย์กลางของหน้าต่างมินชิฟ (พื้นที่ภายในวงกลมภาพประกอบ 2-8) บนการคำนวณค่าศูนย์กลางใหม่ของกลุ่ม

ข้อมูลที่อยู่ภายในหน้าต่าง ซึ่งการเปลี่ยนตำแหน่งศูนย์กลางนี้จะทำให้มีชิฟเวคเตอร์เข้าสู่ค่าศูนย์เสมอ จึงทำให้ได้ตำแหน่งศูนย์กลางที่กลุ่มข้อมูลที่มีความหนาแน่นที่สุดภายในหน้าต่างถ้าหากเปลี่ยนขนาดหน้าต่างจะทำให้ได้ตำแหน่งที่ได้ต่างออกไปเนื่องจากโดยพื้นฐานแล้ววิธีการนี้สร้างจากกระบวนการที่มีความอ่อนไหวด้านการเปลี่ยนแปลงขนาดพื้นที่การคิดคำนวณกลุ่มข้อมูล

จากพื้นฐานกระบวนการมินชิฟ ได้มีการพัฒนาเพิ่มเติมโดย Gary R. Bradski [11] ซึ่งได้พัฒนาเป็นขั้นตอนวิธีที่เรียกว่า การย้ายเข้าสู่ค่ากลางโดยการปรับตัวอย่างต่อเนื่อง (CAMSHIFT : Continuously Adaptive Mean Shift) กระบวนการนี้ได้ถูกใช้เพื่อติดตามวัตถุภายในภาพจากกล้องวิดีโอโดยอาศัยคุณลักษณะสีของวัตถุซึ่งมีการกระจายความน่าจะเป็นของสีที่มีการเปลี่ยนแปลงอยู่เสมอ ขั้นตอนวิธีการย้ายเข้าสู่ค่ากลางโดยการปรับตัวอย่างต่อเนื่องเริ่มต้นจากกำหนดตำแหน่งค่าเริ่มต้นของหน้าต่างค้นหาและใช้ขั้นตอนวิธีมินชิฟ จากนั้นเก็บค่าโมเมนต์ลำดับที่ศูนย์ไว้ และกำหนดขนาดหน้าต่างค้นหาให้เท่ากับฟังก์ชันของโมเมนต์ลำดับที่ศูนย์ และทำซ้ำขั้นตอนวิธีมินชิฟ และกำหนดขนาดหน้าต่างค้นหาใหม่จนกระทั่ง การเปลี่ยนตำแหน่งของบริเวณที่เป็นค่ากลางมีค่าน้อยกว่าค่าขีดแบ่งที่ถูกกำหนดไว้ก่อนแล้วจึงหยุดกระบวนการซึ่งจะทำให้ได้บริเวณพื้นที่ที่มีสีใกล้เคียงกับบริเวณที่ถูกกำหนดให้เป็นตำแหน่งเริ่มต้นของหน้าต่างค้นหา

2.3. ตัวกรองแบบคาลแมน (Kalman Filter)

ตัวกรองคาลแมน (Kalman Filter) [12] คือสูตรทางคณิตศาสตร์ใช้สำหรับหาค่าประมาณที่ดีที่สุดของสถานะของระบบโดยนำข้อมูลเกี่ยวกับความไม่แน่นอน เช่น ความไม่แน่นอนของระบบกลศาสตร์ (Process Noise), ความไม่แน่นอนของการวัดด้วยตัวตรวจ (Measurement Noise) มาประกอบการพิจารณาบนพื้นฐานของความน่าจะเป็นในลักษณะที่เกือถูกันอย่างดีที่สุด (Optimal) การใช้ตัวกรองคาลแมนเริ่มต้นด้วยสมมุติฐานว่าเราสามารถเขียนแบบจำลองทางคณิตศาสตร์ของระบบในรูปของความไม่ต่อเนื่องทางเวลา (Discrete-Time System) ด้วยสมการ

$$x_k = \Phi_{k-1}x_{k-1} + w_k \quad (2.16)$$

$$z_k = H_k x_k + v_k \quad (2.17)$$

สมการที่ 2.16 และ 2.17 คือสมการที่แสดงความสัมพันธ์ของการเปลี่ยนแปลงสถานะของระบบกับเวลา (Discrete-Time Process Model) และสมการแสดงความสัมพันธ์ระหว่างสถานะของระบบกับค่าที่เซนเซอร์วัดได้ (Measurement Model) ตามลำดับ ส่วน w_k และ v_k คือ เป็นตัวแปรสุ่มแบบเกาส์เซียน (Gaussian) มีค่าเฉลี่ย (Mean) เป็นศูนย์ และค่าความแปรปรวน (Variance) ของ w_k

และ v_k มีค่าเป็น Q_k และ R_k ซึ่ง w_k และ v_k ไม่เกี่ยวข้องกันโดยสิ้นเชิง (Uncorrelated) (หรืออีกนัยหนึ่ง แม้ว่าจะรู้ค่าของ w หรือ v ณ เวลาหนึ่งๆ ก็ไม่สามารถนำมาใช้ทำนายค่าของ w หรือ v ณ เวลาอื่นๆ ได้) ในทางคณิตศาสตร์สามารถเขียนอธิบาย w และ v ได้ ดังนี้

$$\text{ค่าเฉลี่ยของ } w = E[w] = 0, \text{ ค่าแปรปรวนของ } w = E[w_k w_j] = \begin{cases} Q_k, k = j \\ 0, k \neq j \end{cases} \quad (2.18)$$

$$\text{ค่าเฉลี่ยของ } v = E[v] = 0, \text{ ค่าแปรปรวนของ } v = E[v_k v_j] = \begin{cases} R_k, k = j \\ 0, k \neq j \end{cases} \quad (2.19)$$

$E[.]$ หมายถึง ค่าคาดหมายของตัวแปรสุ่ม (Expected Value)

การทำงานของตัวกรองคาลแมนมีขั้นตอนดังนี้ ในขณะที่เซนเซอร์ยังไม่สามารถวัดค่าอะไรได้ เราสามารถประมาณค่าสถานะของระบบ x_k ณ เวลา $t_k = t$ ได้จากสมการ

$$\hat{x}_k^- = \Phi_{k-1} \hat{x}_{k-1} \quad (2.20)$$

เครื่องหมายลบเป็นการระบุว่า \hat{x}_k^- เป็นค่าประมาณโดยไม่มีค่าที่เซนเซอร์วัดได้มาประกอบการพิจารณา การคาดการณ์ไปในอนาคตย่อมมีความไม่แน่นอน ให้ความคลาดเคลื่อนในการคาดการณ์คือ

$$e_k^- = x_k - \hat{x}_k^- \quad (2.21)$$

สมการที่ 2.21 ความแตกต่างระหว่างสถานะจริงของระบบกับค่าที่เราคาดการณ์จากการคำนวณ เนื่องจากสมมุติฐานที่ว่า w_k เป็นตัวแปรสุ่มแบบเกาส์เซียน เราสามารถพิสูจน์ได้ว่า \hat{x}_k^- และ e_k^- จึงเป็นตัวแปรสุ่มแบบเกาส์เซียนด้วย โดยที่มีค่าเฉลี่ยเป็นศูนย์ และมีค่าแปรปรวนคือ

$$P_k^- = E[(e_k^-)^2] = \Phi_{k-1}^2 P_{k-1} + Q_k \quad (2.22)$$

เราสามารถใช้สมการที่ (2.21) และ (2.22) เพื่อคาดการณ์ค่าของ x_k และความน่าเชื่อถือของการคาดการณ์ (P_k^-) ไปเรื่อยๆ จนกระทั่งเซนเซอร์สามารถวัดค่าได้ เมื่อเซนเซอร์วัดค่าได้ ตัวกรองคาลแมนจะนำค่าที่วัดได้มาประกอบการพิจารณาเพื่อหาค่าประมาณสถานะของระบบ ตามสมการ

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (Z_k - H_k \hat{x}_k^-) \quad (2.23)$$

โดยที่

$$K_k = P_k^- H_k (P_k^- H_k^2 + R_k)^{-1} \quad (2.24)$$

เครื่องหมายบอกเป็นการระบุว่า \hat{x}_k เป็นค่าประมาณโดยได้นำค่าที่เซนเซอร์วัดได้มาประกอบการพิจารณา สมการที่ 2.23 คือสมการที่ใช้ปรับค่า \hat{x}_k^- ที่เราคาดการณ์ไว้ล่วงหน้า (ก่อนที่จะเซนเซอร์จะวัดค่าได้) เมื่อพิจารณาจากสมการที่ 2.23 และ 2.24 แล้วจะเห็นว่า $H_k \hat{x}_k^-$ คือ สมการที่ใช้ประมาณค่าที่เซนเซอร์ควรวัดได้ (\hat{Z}_k) คำนวณโดยใช้ค่าประมาณ \hat{x}_k^- ตัวกรองคาลแมนจะทำงานโดยหาผลต่างระหว่างค่าที่เซนเซอร์ควรวัดได้กับค่าที่วัดได้จริง (Residual: $Z_k - H_k \hat{x}_k^-$) นำมาให้น้ำหนักโดยการคูณด้วยค่าคาลแมนเกน (Kalman Gain: K) แล้วนำผลที่ได้มาใช้แก้ไขค่า \hat{x}_k^- ที่ได้คาดการณ์ไว้ล่วงหน้า

จะสังเกตได้ว่าค่าที่นำมาใช้แก้ไข \hat{x}_k^- ขึ้นอยู่กับขนาดของผลต่างระหว่างค่าที่เซนเซอร์ควรวัดได้กับค่าที่วัดได้จริง และค่าคาลแมนเกน กล่าวโดยทั่วไปถ้าผลต่างระหว่างค่าที่คาดการณ์กับค่าที่วัดได้จริงมีมากและข้อมูลมีความน่าเชื่อถือน้อยค่า คาลแมนเกนที่คำนวณได้จะมีค่าสูง ทำให้ต้องแก้ไข \hat{x}_k^- มาก ($K_k (Z_k - H_k \hat{x}_k^-)$ มีค่ามาก) ในทางกลับกันถ้าผลต่างระหว่างค่าที่คาดการณ์ไว้กับค่าที่วัดได้จริงมีน้อยและข้อมูลมีความน่าเชื่อถือมาก ค่าคาลแมนเกนที่คำนวณได้จะมีค่าต่ำ ทำให้แก้ไข \hat{x}_k^- เพียงเล็กน้อย ($K_k (Z_k - H_k \hat{x}_k^-)$ มีค่าน้อย) โดยอัตโนมัติ การคำนวณค่าคาลแมนเกนตามสมการที่ 2.24 เป็นการคำนวณที่ได้นำข้อมูลทางสถิติของระบบมาพิจารณาแล้วและเป็นค่า ที่จะทำให้ค่าความคลาดเคลื่อนของการประมาณในการหาค่าประมาณโดยใช้สมการที่ 2.23 มีค่าต่ำสุด

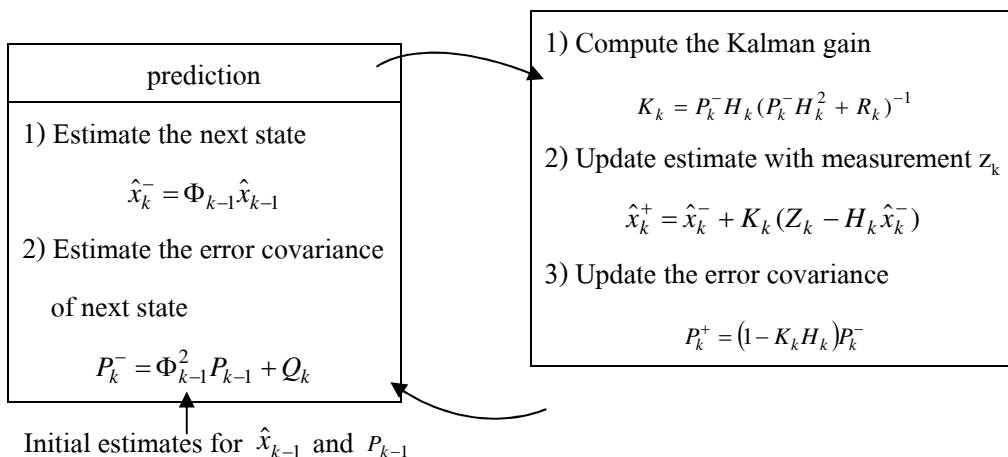
$$e_k^+ = x_k - \hat{x}_k^+ \quad (2.25)$$

หรือกล่าวอีกนัยหนึ่ง e_k^+ เป็นตัวแปรแบบเกาส์เซียนด้วย โดยมีค่าเฉลี่ยเป็นศูนย์ และมีค่าแปรปรวนคือ

$$P_k^+ = E[e_k^+ e_k^{+T}] = (1 - K_k H_k) P_k^- \quad (2.26)$$

สุดท้ายจะสังเกตได้ว่าตัวกรองคาลแมนทำงานในลักษณะที่เรียกว่าวนซ้ำ (Recursive) มีการคาดการณ์ล่วงหน้าว่าสถานะของระบบ \hat{x}_k^- น่าจะเป็นอย่างไรโดยไม่ใช้ค่าจากเซนเซอร์ (Predict) และการหาค่าประมาณ \hat{x}_k^+ เมื่อเซนเซอร์วัดค่าได้ (Measurement) ทำให้เป็นที่ยอมรับว่าตัวกรองคาลแมนเป็นตัวประมาณค่าสถานะของระบบเหมาะสมที่สุด (Optimal State Estimator)

Measurement



ภาพประกอบ 2-9 การทำงานของตัวกรองคาลแมน

2.4. แบบจำลองฮิดเดนมาร์คอฟ (Hidden markov model)

แบบจำลองฮิดเดนมาร์คอฟ[13] เป็นแบบจำลองทางสถิติรูปแบบหนึ่ง สร้างอยู่บนพื้นฐานของทฤษฎีมาร์คอฟ ที่แสดงความเป็นไปได้ของการกระทำหรือว่าเหตุการณ์ที่เป็นไปได้ ฮิดเดนมาร์คอฟให้คำจำกัดความของการกระทำหรือเหตุการณ์ที่เป็นไปได้นั้นว่าเป็น สถานะซ่อน (Hidden State) ซึ่งแต่ละสถานะซ่อนจะมีเส้นโยงถึงกันรวมทั้งมีความน่าจะเป็นกำกับ ฮิดเดนมาร์คอฟได้มีการนำมาประยุกต์ ใช้กับงานทางด้านการจับคู่รูปแบบ (Pattern matching) ที่เปลี่ยนแปลงตามเวลา เช่น การรู้จำเสียงพูด รู้จำท่าทาง ฯลฯ โดยมีองค์ประกอบที่สำคัญดังนี้

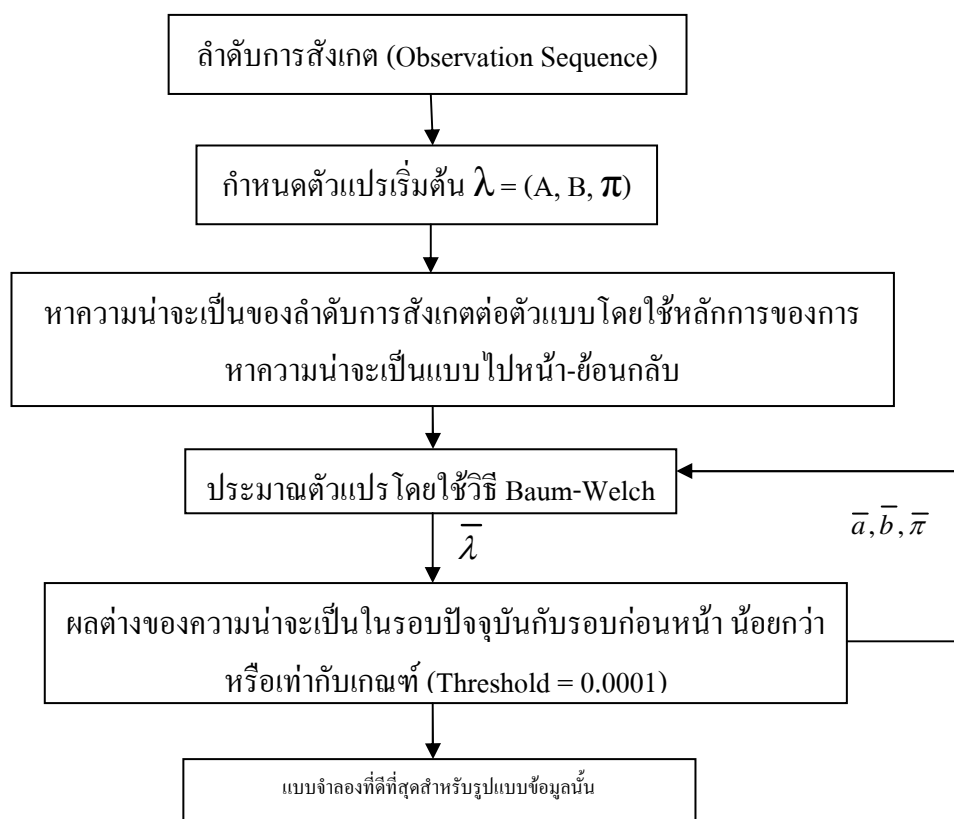
- N คือ จำนวนสถานะในตัวแบบ สถานะจะเปลี่ยนไปตามเวลา t
- M คือ จำนวนของเหตุการณ์ต่อสถานะ เหตุการณ์ที่ได้จะสอดคล้องกับอินพุตที่ป้อนให้กับแบบจำลอง
- a_{ij} คือ ความน่าจะเป็นในการเปลี่ยนสถานะจากสถานะ i ไปสถานะ j
- $b_j(O)$ คือ ค่าความน่าจะเป็นของเหตุการณ์ O ที่เกิดขึ้นในสถานะ j
- π_i ความน่าจะเป็นที่จะเกิดสถานะแรก

ส่วนประกอบต่าง ๆ ของแบบจำลองสามารถรวมเข้าด้วยกันแล้วแทนด้วยสมการ (2.27) ที่แสดงเป็นเซตของตัวแปรที่เสร็จสมบูรณ์ของแบบจำลองฮิดเดนมาร์คอฟในการจัดจำรูปแบบของข้อมูล ซึ่งแบบจำลองฮิดเดนมาร์คอฟสามารถแบ่งออกเป็น 2 ส่วนด้วยกันคือ การฝึกฝนแบบจำลอง (Training) และการทดสอบการจดจำของแบบจำลอง (Evaluation)

$$\lambda = (A, B, \pi) \tag{2.27}$$

2.4.1. การฝึกสอนแบบจำลอง (Training)

การฝึกฝนแบบจำลองให้จดจำรูปแบบข้อมูล เป็นการประมาณตัวแปรของแบบจำลองที่ดีที่สุดสำหรับรูปแบบของข้อมูลในแต่ละรูปแบบ ซึ่งขั้นตอนการฝึกฝนให้กับแบบจำลองแสดงดังภาพประกอบ 2-9 โดยอินพุตของแบบจำลอง คือ ลำดับการสังเกตของข้อมูล (Observation Sequence) จากนั้นกำหนดแบบจำลองเริ่มแรก $\lambda = (A, B, \pi)$ แล้วคำนวณหาค่าความน่าจะเป็นของอินพุตต่อแบบจำลองนั้น โดยใช้หลักการของการหาความน่าจะเป็นแบบไปหน้า-ย้อนกลับ (Forward-Backward Algorithm) เมื่อได้ค่าความน่าจะเป็นของอินพุตต่อแบบจำลอง จึงเข้าสู่กระบวนการประมาณตัวแปรโดยใช้วิธีของ Baum-Welch เพื่อทำการปรับตัวแบบจนกระทั่งได้แบบจำลองที่ดีที่สุดสำหรับอินพุตนั้น



ภาพประกอบ 2-10 ขั้นตอนการฝึกสอนแบบจำลอง

การคำนวณหาค่าความน่าจะเป็นของอินพุตหรือลำดับการสังเกตของแบบจำลอง โดยใช้หลักการหาความน่าจะเป็นแบบไปหน้า-ย้อนกลับ มีรายละเอียดดังนี้

ในกระบวนการหาความน่าจะเป็นแบบไปหน้า (Forward Algorithm) มีการกำหนดตัวแปร ดังนี้

- $\alpha_t(i)$ เป็นตัวแปรแบบไปหน้า (Forward) ณ เวลา t ที่สถานะ i
- π_i เป็นค่าความน่าจะเป็น ณ เวลา $t = 1$ ที่สถานะ i
- $b_j(O_t)$ เป็นค่าความน่าจะเป็นของข้อมูล O ณ เวลา t ที่สถานะ j
- a_{ij} เป็นค่าความน่าจะเป็นในการเปลี่ยนจากสถานะ i ไปที่สถานะ j
- ขั้นเริ่มต้น

$$\alpha_1(i) = \pi_i b_i(O_1) \quad , \quad 1 \leq i \leq N \quad (2.28)$$

- ขั้นการวนซ้ำ

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}) \quad (2.29)$$

$$1 \leq t \leq T - 1, \quad 1 \leq j \leq N$$

- ขั้นสิ้นสุด

$$P(O | \lambda) = \sum_{i=1}^N \alpha_T(i) \quad (2.30)$$

ส่วนของกระบวนการหาความน่าจะเป็นแบบย้อนกลับ (Backward Algorithm) ได้มีการกำหนดตัวแปรเพิ่มเติมจากกระบวนการหาความน่าจะเป็นแบบไปหน้า ดังนี้

- $\beta_t(i)$ เป็นตัวแปรแบบย้อนกลับ ณ เวลา t ที่สถานะ i
- ขั้นเริ่มต้น

$$\beta_T(i) = 1 \quad , \quad 1 \leq i \leq N \quad (2.31)$$

- ขั้นอุปนัย

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \quad (2.31)$$

$$t = T-1, T-2, \dots, 1 \quad , \quad 1 \leq i \leq N$$

หลังจากกระบวนการหาความน่าจะเป็นแบบไปหน้า-ย้อนกลับ แล้วจะเข้าสู่กระบวนการประมาณค่าพารามิเตอร์โดยใช้วิธี Baum-Welch โดยนำผลลัพธ์ที่ได้จากก่อนหน้า มาใช้ซึ่งมีรายละเอียด ดังนี้

- กำหนดให้ $\xi_t(i,j)$ คือความน่าจะเป็นของการอยู่ในสถานะ i ที่เวลา t และสถานะ j ที่เวลา $t+1$

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \quad (2.32)$$

- กำหนดตัวแปร $\gamma_t(i)$ เป็นความน่าจะเป็นของการ อยู่ในสถานะ i ที่เวลา t โดยที่ตัวแบบ λ และลำดับของเหตุการณ์ O มีความสัมพันธ์กับ $\xi_t(i, j)$ โดยการบวกกันทุก j

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j) \quad (2.33)$$

- การประมาณตัวแปรใหม่ของ A, B และ π

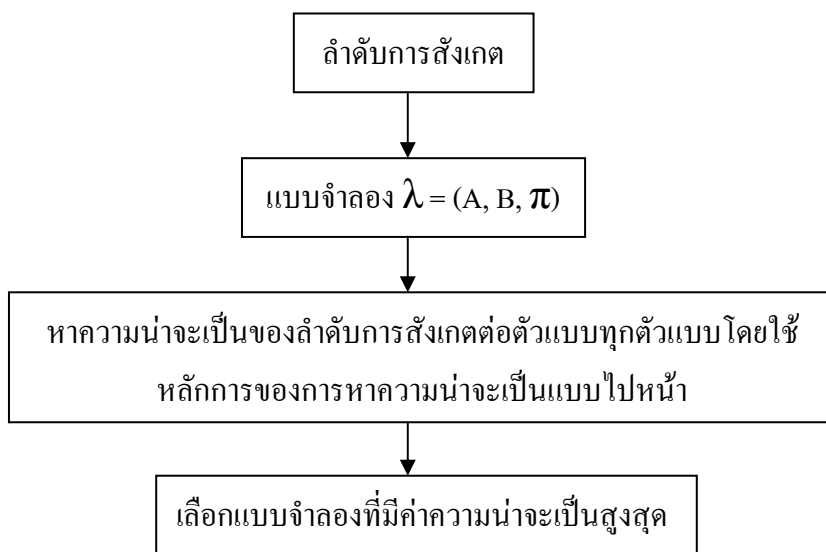
$$\bar{\pi}_i = \gamma_1(i) \quad (2.34)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (2.35)$$

$$\bar{b}_j(k) = \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \quad (2.36)$$

2.4.2. การทดสอบการจดจำของแบบจำลอง (Evaluation)

หลังจากที่ได้ทำการฝึกฝนรูปแบบข้อมูลให้กับแบบจำลองแล้ว จึงทำการทดสอบการจดจำของแบบจำลองโดยการคำนวณหาความน่าจะเป็นของเหตุการณ์ต่อแบบจำลองโดยใช้หลักการหาความน่าจะเป็นแบบไปหน้ากับทุกๆ แบบจำลอง แบบจำลองใดที่ได้ค่าความน่าจะเป็นสูงสุด ถือว่าเป็นคำตอบของลำดับสังเกตนั้น รายละเอียดของขั้นตอนการทดสอบการจดจำของแบบจำลองอธิบายได้ด้วยภาพประกอบ 2-11



ภาพประกอบ 2-11 ขั้นตอนการทดสอบการรู้จำของแบบจำลอง

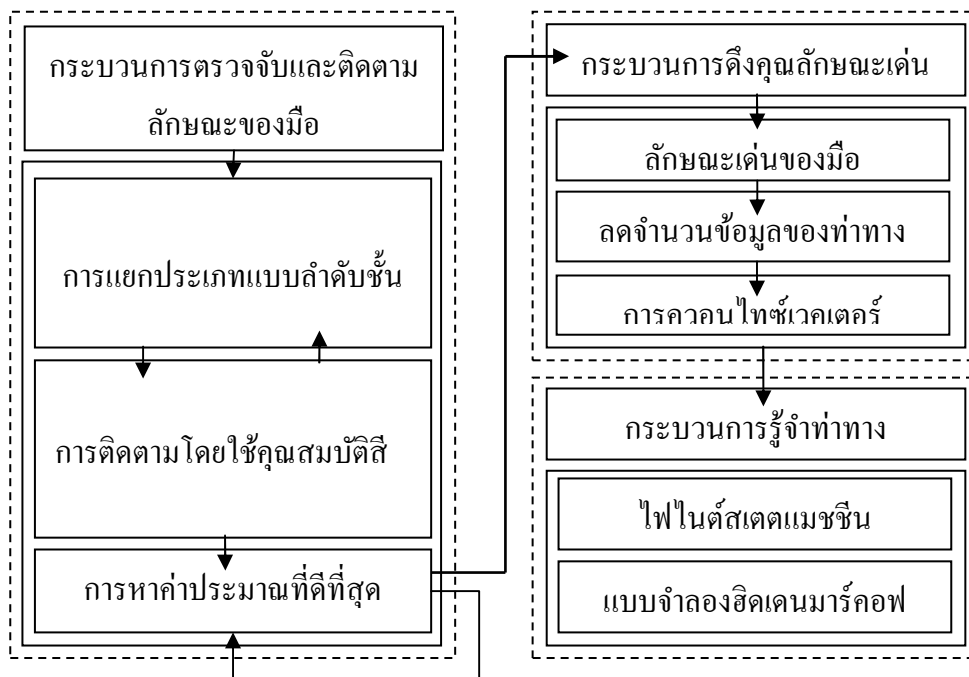
2.5. สรุป

เนื้อหาภายในบทนี้กล่าวถึงกระบวนการด้านการประมวลผลภาพด้วยคอมพิวเตอร์ที่เกี่ยวข้องกับงานวิจัยนี้ได้แก่ กระบวนการตรวจจับวัตถุที่สนใจภายในภาพด้วยวิธีการเรียนรู้แบบอดาบาสต การบวนการติดตามวัตถุเคลื่อนไหวโดยใช้คุณสมบัติของสี การดำเนินการของตัวกรองกาลแมนในการตัวประมาณค่าสถานะของระบบเหมาะสมที่สุด และกล่าวถึงกระบวนการพื้นฐานเกี่ยวกับการจัดจํารูปแบบของข้อมูลโดยใช้แบบจำลองฮิดเดนมาร์คอฟ

บทที่ 3

การออกแบบและพัฒนาระบบ

งานวิจัยนี้ได้ประยุกต์ใช้กระบวนการจับและติดตามวัตถุเคลื่อนไหวกจากภาพในกล้องวิดีโอเพื่อพัฒนาระบบการรู้จำคำสั่งโดยท่าทางมือ โดยมุ่งเน้นไปที่กระบวนการรู้จำ กระบวนการโดยรวมแสดงในภาพประกอบ 3-1 เริ่มต้นจากการตรวจจับตำแหน่งของวัตถุที่สนใจภาพในภาพวิดีโอโดยในที่นี้ก็คือลักษณะของมือที่ตั้งตรงและขนานกับตัวกล้อง ซึ่งได้นำหลักการการตรวจจับวัตถุที่ใช้คุณลักษณะเด่นแบบฮาไลต์และการเรียนรู้แบบอคาบสมุทรมาทำการสอนระบบ จากนั้นจึงทำการติดตามการเคลื่อนที่ของมือโดยใช้คุณสมบัติของสีของวัตถุซึ่งมีการกระจายความน่าจะเป็นของสีอยู่ภายในภาพ ผลจากการติดตามการเคลื่อนที่ของมือจะทำให้ได้คุณลักษณะที่สำคัญของท่าทางที่ผู้ใช้แสดงออกมาและจากคุณลักษณะดังกล่าวถูกนำไปใช้ในกระบวนการรู้จำท่าทาง ซึ่งท่าทางจะถูกนิยามโดยการเคลื่อนที่ของมือในลักษณะต่างๆ แบ่งได้ 2 แบบ 1)การเคลื่อนที่อย่างง่าย เช่น โบกมือไปทางซ้าย-ขวา หรือขึ้น-ลง 2)การเคลื่อนที่เป็นสัญลักษณ์ เช่น เคลื่อนที่มือเป็นรูปทรงสี่เหลี่ยม สามเหลี่ยม หรือวงกลม เป็นต้น ผลจากกระบวนการรู้จำนี้จะทำให้ระบบทราบว่าผู้ใช้ได้แสดงท่าทางใดบ้าง โดยท่าทางที่แสดงออกมาสามารถนำไปใช้แทนชุดคำสั่งซึ่งผู้ใช้งานสามารถที่จะกำหนดเองได้เพื่อนำไปควบคุมระบบคอมพิวเตอร์หรือและนำไปประยุกต์ใช้ตามความเหมาะสมของงานแต่ละประเภท



ภาพประกอบ 3-1 กระบวนการโดยรวมเทคนิคการรู้จำคำสั่งโดยท่าทางมือ

3.1. กระบวนการตรวจจับและติดตามลักษณะของมือ (Hand Detection and Tracking)

3.1.1. กระบวนการฝึกสอนด้วยการเรียนรู้แบบอดาบูสท (AdaBoost Training)

ในกระบวนการตรวจจับลักษณะของมือได้นำหลักการการเรียนรู้แบบอดาบูสทซึ่งเป็นหนึ่งในกระบวนการการเรียนรู้ของเครื่อง (Machine Learning) ที่เป็นวิธีเพื่อให้คอมพิวเตอร์สามารถเรียนรู้จากการวิเคราะห์ชุดข้อมูล การเรียนรู้แบบอดาบูสทเป็นการเรียนรู้แบบมีผู้สอน (Supervised Learning) ซึ่งมีตัวอย่างของชุดข้อมูลเพื่อใช้สำหรับการฝึกสอน (Training Data) โดยชุดข้อมูลประกอบด้วยกลุ่มตัวอย่าง 2 ชนิด คือ ตัวอย่างที่ใช่ เป็นภาพของวัตถุที่ต้องการให้คอมพิวเตอร์เรียนรู้ซึ่งในที่นี้คือ ภาพของมือในลักษณะตั้งตรง และตัวอย่างที่ไม่ใช่ เป็นภาพของสิ่งแวดล้อมอื่นๆ โดยต้องไม่มีวัตถุที่สนใจอยู่ในภาพ



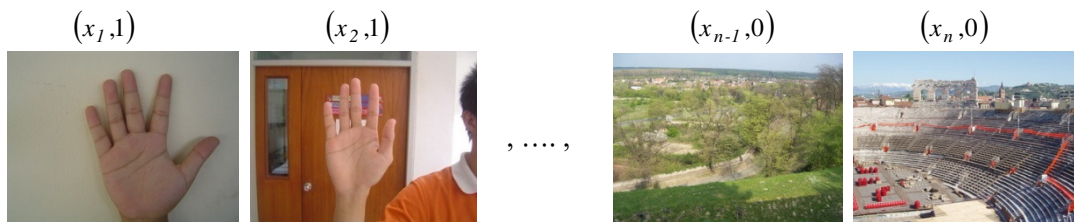
(a)

(b)

ภาพประกอบ 3-2 ชุดข้อมูลสำหรับการฝึกสอน (a) กลุ่มตัวอย่างที่ใช่

(b) กลุ่มตัวอย่างที่ไม่ใช่

โดยชุดข้อมูลที่ใช้สำหรับการฝึกสอนควรมีจำนวนของกลุ่มตัวอย่างในแต่ละประเภทมากในระดับหนึ่ง เพื่อให้สามารถตรวจจับวัตถุที่สนใจได้อย่างถูกต้องและแม่นยำมากขึ้น ได้มีการเก็บกลุ่มตัวอย่างจากสภาพแวดล้อมต่างๆ เช่น กลางแจ้ง ที่ร่ม และภายในอาคาร เป็นต้น โดยได้ทำการเก็บกลุ่มตัวอย่างที่ใช่ประมาณ 5,000 ตัวอย่าง และเก็บกลุ่มตัวอย่างที่ไม่ใช่ประมาณ 10,000 ตัวอย่าง กลุ่มตัวอย่างทั้งหมดจะถูกกำหนดเป็นคู่ลำดับ $(x_i, y_i) \dots (x_n, y_n)$ โดยที่ x_i เป็นลำดับของกลุ่มตัวอย่างเมื่อ $i = 1, 2, \dots, n$ และ y_i มีค่าเท่ากับ 0, 1 สำหรับกลุ่มตัวอย่างที่ไม่ใช่และกลุ่มตัวอย่างที่ใช่ตามลำดับ

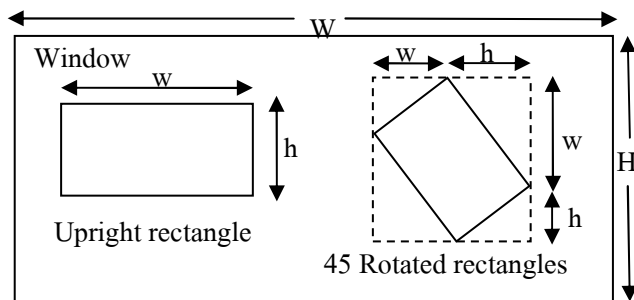


ภาพประกอบ 3-3 การกำหนดลำดับให้กับกลุ่มตัวอย่าง

กำหนดขนาดของภาพกลุ่มตัวอย่างเป็น $W \times H$ จุดภาพ (Pixel) และใช้คุณลักษณะเด่นแบบฮาร์ไลทเพื่อหาลักษณะของวัตถุที่สนใจจากผลรวมของค่าจุดสีภายในแต่ละคุณลักษณะเด่น ซึ่งแต่ละคุณลักษณะเด่นสามารถอธิบายได้ดังนี้

$$f = (x, y, w, h, \alpha) \tag{3.1}$$

กำหนดให้ x, y, w, h คือ จุดเริ่มต้น w, h คือ ความกว้างกับความยาว และ $\alpha \in \{0^\circ, 45^\circ\}$ คือ ค่ามุมของแต่ละคุณลักษณะเด่น โดยที่ $x, y \geq 0$, $w, h > 0$, $x + w < W$ และ $y + h < H$



ภาพประกอบ 3-4 ตัวอย่างคุณลักษณะเด่นมุมตั้งตรง (Upright) และมุมเอียง 45 องศา (45° Rotated)

คุณลักษณะเด่นแบบฮาร์ไลทที่ถูกนำมาใช้มีอยู่ด้วยกันได้เป็น 14 แบบ แบ่งได้เป็น 4 แบบสำหรับคุณลักษณะเด่นแบบขอบ (Edge Features) 8 แบบสำหรับคุณลักษณะเด่นแบบเส้นตรง (Line Features) และ 2 แบบสำหรับคุณลักษณะเด่นแบบมีจุดศูนย์กลาง (Center-Surround Features) ดังที่แสดงตามภาพประกอบ 2-1 ซึ่งจำนวนของคุณลักษณะเด่นทั้ง 14 แบบที่ถูกสร้างขึ้นภายในภาพกลุ่มตัวอย่างหนึ่งภาพด้วยขนาดและตำแหน่งที่กันออกไปสามารถคำนวณได้จาก

$$XY \left(W + 1 - w \frac{X+1}{2} \right) \left(H + 1 - h \frac{Y+1}{2} \right) \tag{3.2}$$

สมการที่ (3.2) เป็นการคำนวณหาคุณลักษณะเด่นทั้ง 14 แบบ ที่มีมุมตั้งตรง กำหนดให้ $X = [W/w]$ และ $Y = [H/h]$ คือ ตัวแปรในการขยายขนาดสูงสุดของคุณลักษณะเด่นในทิศทางแกน x และ y ส่วนคุณลักษณะเด่นที่มีมุมเอียง 45 องศา สามารถคำนวณได้จาก

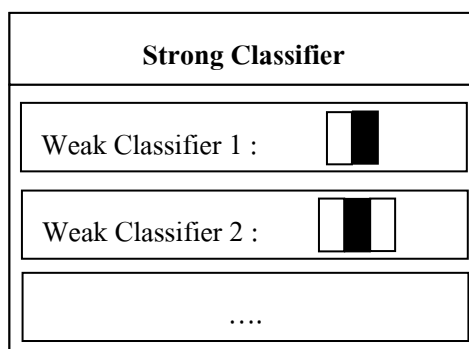
$$XY \cdot \left(W + 1 - z \frac{X+1}{2} \right) \left(H + 1 - z \frac{Y+1}{2} \right) \quad \text{เมื่อ } z = w + h \quad (3.3)$$

ตารางที่ 3-1 แสดงให้เห็นถึงจำนวนคุณลักษณะเด่นทั้งหมดที่สามารถหาได้ภายในภาพกลุ่มตัวอย่าง

ขนาด 24x24 จุดภาพ

แบบ	w/h	X/Y	จำนวน
1a ; 1b	2/1 ; 1/2	12/24 ; 24/12	43,200
1c ; 1d	2/1 ; 1/2	8/8	8,464
2a ; 2c	3/1 ; 1/3	8/24 ; 24/8	27,600
2b ; 2d	4/1 ; 1/4	6/24 ; 24/6	20,736
2e ; 2g	3/1 ; 1/3	6/6	4,356
2f ; 2h	4/1 ; 1/4	4/4	3,600
3a	3/3	8/8	8,464
3b	3/3	3/3	1,521
ทั้งหมด			117,941

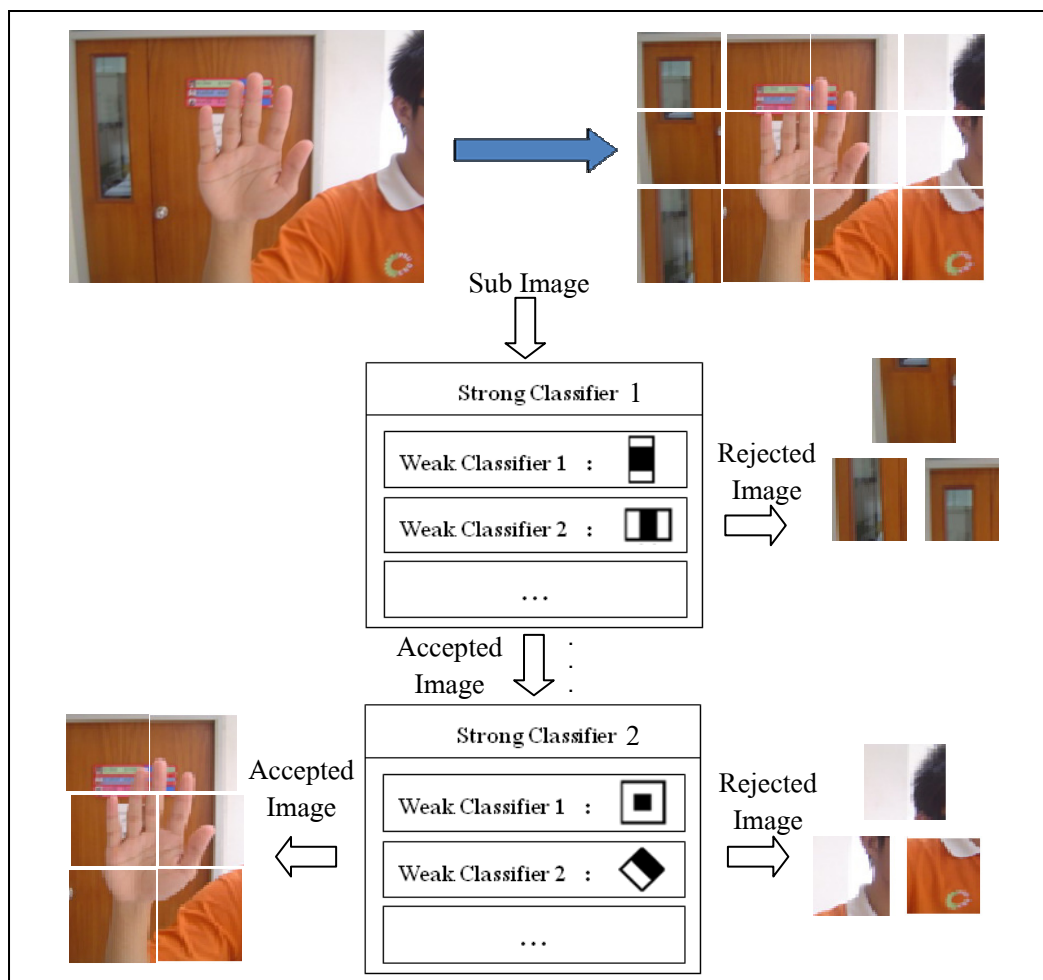
แต่ละคุณลักษณะเด่นที่ถูกสร้างขึ้นจะถูกกำหนดด้วยค่าน้ำหนักเริ่มต้น $w_{i,i} = \frac{1}{2m}, \frac{1}{2l}$ สำหรับ $y_i = 0, 1$ เมื่อ m และ l คือ จำนวนกลุ่มตัวอย่างที่ไม่ใช่ และ กลุ่มตัวอย่างที่ใช่ตามลำดับ ในกระบวนการฝึกสอนในแต่ละรอบ คุณลักษณะเด่นที่มีความสามารถแยกประเภทระหว่างกลุ่มตัวอย่างที่ใช่กับกลุ่มตัวอย่างที่ไม่ใช่ได้อย่างชัดเจนจะถูกเลือก ซึ่งจะเรียกว่าตัวแยกประเภทแบบอ่อน (Weak Classifier) แต่ในทางปฏิบัติคุณลักษณะเด่นเพียงหนึ่งตัวมีความแม่นยำต่ำในการตรวจจับ ดังนั้นจึงได้นำคุณลักษณะเด่นทั้งหมดที่ถูกเลือกนำมาเรียงต่อกันเพื่อเพิ่มความแม่นยำในการตรวจจับ ซึ่งถูกเรียกว่าตัวแยกประเภทแบบแข็งแกร่ง (Strong Classifier)



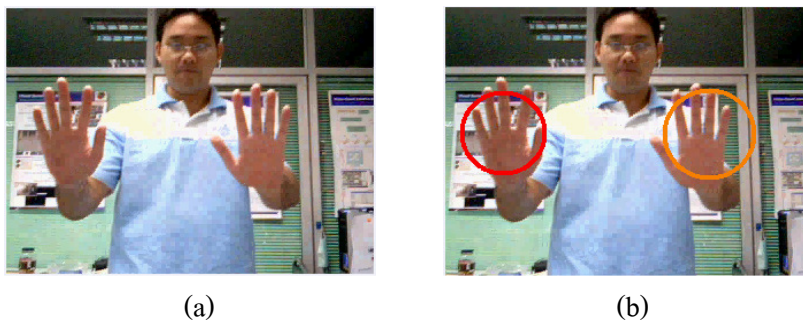
ภาพประกอบ 3-5 ตัวแยกประเภทแบบแข็งแกร่ง

3.1.2. กระบวนการตรวจจับวัตถุด้วยตัวแยกประเภทแบบลำดับชั้น (Cascade Classifier Detection)

ตัวแยกประเภทแบบลำดับชั้น คือ โครงสร้างของแผนผังการตัดสินใจแบบปล่อยทิ้ง (Degenerate Decision Tree) ซึ่งเกิดจากลำดับของตัวแยกประเภทแบบแข็งแกร่งที่ได้จากการฝึกสอนในแต่ละชั้น โดยในกระบวนการตรวจจับวัตถุผลลัพธ์ของการตรวจจับที่บวกจากตัวแยกประเภทแบบแข็งแกร่งตัวแรกจะนำไปสู่การตรวจจับในตัวแยกประเภทแบบแข็งแกร่งตัวที่สอง และตัวที่สองนำไปสู่ตัวที่สาม แต่ถ้าเกิดมีผลลัพธ์ของการตรวจจับที่ตำแหน่งใดเป็นลบก็จะทำการปล่อยภาพนั้นๆ ทิ้งทันที ดังนั้นภาพจะต้องผ่านการตรวจจับด้วยตัวแยกประเภทแบบแข็งแกร่งทุกลำดับชั้นก่อนจึงจะถือว่าภาพนั้นมีวัตถุที่สนใจอยู่ภายในภาพ



ภาพประกอบ 3-6 กระบวนการตรวจจับวัตถุด้วยตัวแยกประเภทแบบลำดับชั้น

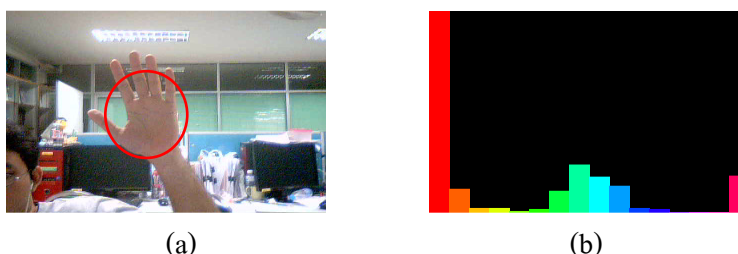


ภาพประกอบ 3-7 กระบวนการตรวจจับลักษณะมือ (a) ภาพต้นฉบับ (b) ผลลัพธ์หลังจาก
กระบวนการตรวจจำ

กระบวนการตรวจจำจะให้ผลลัพธ์ที่ดีเมื่อลักษณะของมือในภาพต้นฉบับตรงตามรูปแบบของกลุ่มตัวอย่างที่ใช่ นั่นก็คือมือที่อยู่ในลักษณะตั้งตรงวางขนานกับตัวกล้อง (ภาพประกอบที่ 3-6) แต่เมื่อมือมีการเคลื่อนที่ทำให้เปลี่ยนรูปร่างไปจะทำให้กระบวนการตรวจไม่สามารถตรวจจับได้ จึงได้กำหนดลักษณะมือที่ตั้งตรงให้เป็นจุดเริ่มต้นของระบบ เมื่อระบบสามารถตรวจจับลักษณะมือได้ ขั้นตอนต่อไปจะใช้กระบวนการติดตามมือโดยใช้คุณสมบัติของสีแทนเพื่อแก้ปัญหาของกระบวนการตรวจจับที่ได้กล่าวไปแล้วข้างต้น

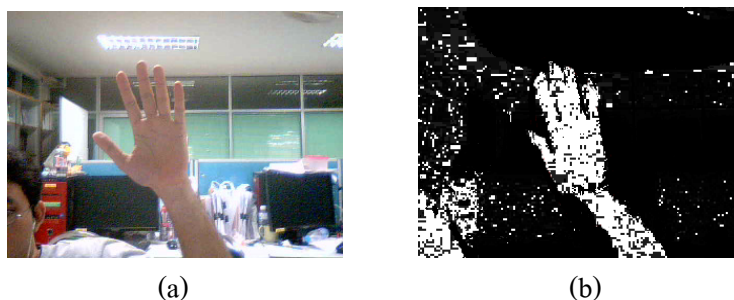
3.1.3. กระบวนการติดตามมือโดยใช้คุณสมบัติของสี (Hand Tracking Using CAMSHIFT)

จากปัญหาที่กล่าวไปในกระบวนการตรวจจับลักษณะของมือ จึงได้ใช้กระบวนการติดตามมือโดยใช้คุณสมบัติของสีจากบริเวณมือที่ตรวจจับได้ในตอนเริ่มต้นของระบบมาช่วยแก้ไขปัญหาคะบวนการติดตามมือโดยใช้คุณสมบัติของสีจะใช้วิธีการที่เรียกว่า การย้ายเข้าสู่ค่าเฉลี่ยโดยปรับตัวอย่างต่อเนื่อง ที่พัฒนามาจากวิธีการย้ายเข้าสู่ค่าเฉลี่ย ซึ่งใช้เฉพาะค่าสี ในระบบสี HSV มาคำนวณค่ากราฟแสดงความหนาแน่นของสี (Color Density Histogram)



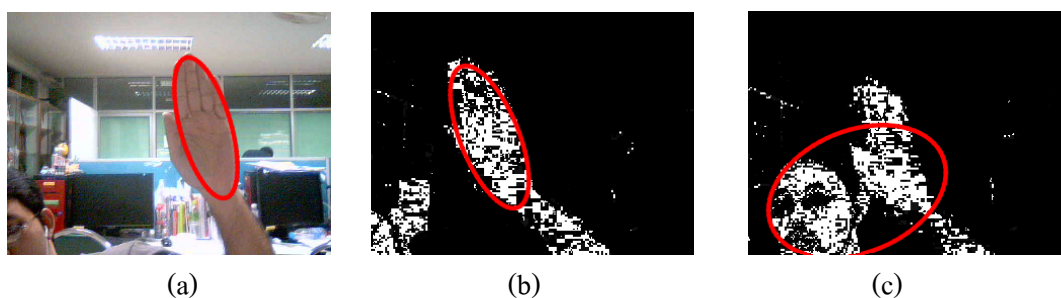
ภาพประกอบ 3-8 การคำนวณหากราฟแสดงความหนาแน่นของสี (a) บริเวณของมือที่ได้จาก
กระบวนการตรวจจับ (b) กราฟแสดงความหนาแน่นของสีที่ได้จากบริเวณมือ

ในระหว่างกระบวนการการติดตามกราฟแสดงความหนาแน่นของสีถูกใช้เป็นแบบตัวอย่างหรือตารางการค้นหา เพื่อแปลงภาพวิดีโอที่เข้ามาเป็นภาพความน่าจะเป็นของสีนั้นๆ (Color Probability image) ซึ่งใช้วิธีการฉายย้อนกลับ (Back Projection) ก็คือการหาว่าจุดภาพหรือกลุ่มของจุดภาพ (Patch) ของภาพที่เข้ามาเข้ากันได้กับกราฟแสดงความหนาแน่นของสีที่ได้จากตัวอย่างหรือไม่ โดยกำหนดช่วงของความน่าจะเป็นอยู่ช่วง 0 (ความน่าจะเป็น 0.0) จนถึงค่าที่มากที่สุดของจุดภาพ (ความน่าจะเป็น 1.0) สำหรับภาพ 8 บิตของค่าสีจะอยู่ในช่วง 0 ถึง 255



ภาพประกอบ 3-9 การฉายภาพย้อนกลับ (a) ภาพต้นฉบับ (b) ภาพความน่าจะเป็นของสี

จากนั้นจะทำการเข้าสู่บริเวณกลุ่มสีที่มีความหนาแน่นสูงสุด และคำนวณหากราฟแสดงความหนาแน่นของสีใหม่ เพื่อใช้สำหรับการเข้าสู่บริเวณกลุ่มสีที่มีความหนาแน่นสูงสุดของภาพวิดีโอเฟรมถัดไป วิธีการนี้จะทำวนซ้ำไปเรื่อยๆ ทำให้ได้กระบวนการในการติดตามวัตถุโดยใช้คุณสมบัติของสี แต่กระบวนการติดตามวัตถุโดยใช้คุณสมบัติของสีก็สามารถเกิดข้อผิดพลาดในการติดตามได้ถ้าเกิดมีวัตถุหรือพื้นหลังที่มีค่าสีอยู่ในช่วงสีที่สนใจก็อาจทำให้เกิดการเข้าสู่บริเวณนั้นๆ ได้เช่นกัน



ภาพประกอบ 3-10 กระบวนการติดตามโดยใช้คุณสมบัติของสี (a) ภาพบริเวณที่สนใจ (b) การเข้าสู่บริเวณกลุ่มสีที่มีความหนาแน่นสูงสุด (c) ความผิดพลาดของการเข้าสู่

จากความผิดพลาดดังกล่าวจึงได้มีการนำเอาวิธีการการหาค่าประมาณที่ดีที่สุดของสถานะของระบบที่เรียกว่า ตัวกรองคาลแมนเข้ามาช่วยแก้ไขปัญหานี้ ซึ่งจะกล่าวถึงในหัวข้อถัดไป

3.1.4. การหาค่าประมาณที่ดีที่สุดด้วยตัวกรองคาลแมน (Optimal State Estimation Using Kalman Filter)

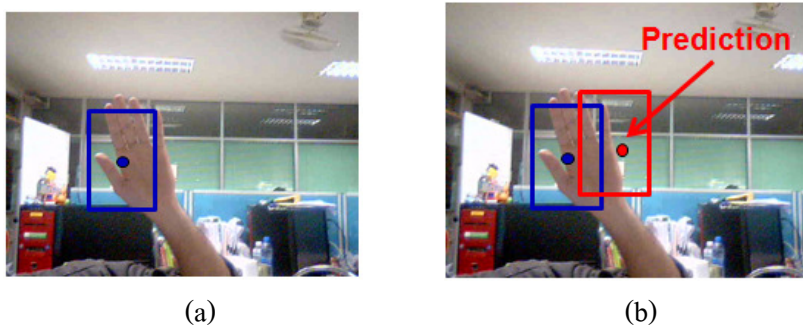
ตัวกรองคาลแมนถูกนำมาใช้เพื่อเพิ่มประสิทธิภาพในกระบวนการติดตามมือเมื่อมีสัญญาณรบกวนเกิดขึ้นภายในเฟรมภาพ เช่น มีวัตถุหรือพื้นหลังที่มีค่าสีอยู่ในช่วงสีที่สนใจ เป็นต้น การทำงานของตัวกรองคาลแมนมีอยู่ 2 ขั้นตอน คือ

1). การทำนายค่าสถานะ (Prediction) เป็นการนำข้อมูลจากสถานะที่เวลาปัจจุบันไปใช้ทำนายสถานะของระบบที่อาจเป็นไปได้ที่เวลาถัดไป สถานะในที่นี้สามารถนิยามได้ดังนี้

$$X_t = [x(t) \ y(t) \ v_x(t) \ v_y(t)]^T \quad (3.4)$$

เมื่อ $x(t), y(t)$ คือ ตำแหน่งจุดศูนย์กลางมวลของมือ และ $v_x(t), v_y(t)$ คือ ความเร็วในแนวแกน x, y โดยสามารถทำนายค่าสถานะได้จากสมการที่ (2.16) ซึ่งมีเมทริกซ์ของการเปลี่ยนสถานะ (Φ : State Transition Matrix) กับสัญญาณรบกวนสถานะ (w : noise) เป็นตัวแปรในการทำนายค่าสถานะที่เวลาถัดไป ในที่นี้ได้ทำการประมาณการเคลื่อนที่ของมือในระหว่างเฟรมภาพที่ติดกันให้อยู่ในรูปแบบเส้นตรงเนื่องจากความแตกต่างระหว่างช่วงเวลาของเฟรมภาพมีน้อยมาก (ΔT) ดังนั้นสามารถกำหนดค่าเมทริกซ์ของการเปลี่ยนสถานะได้ดังนี้

$$\Phi = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

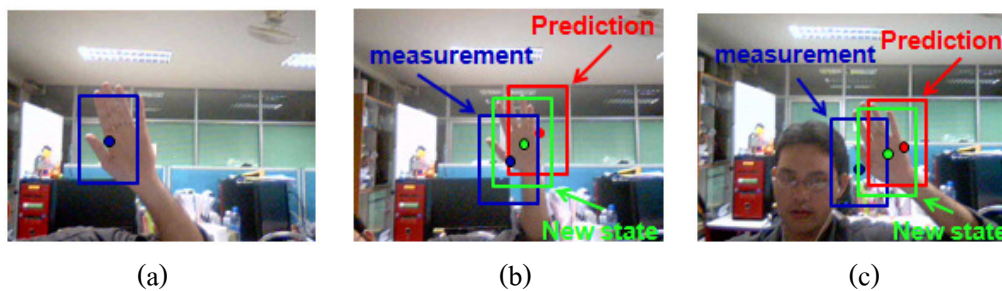


ภาพประกอบ 3-11 การทำนายค่าสถานะ (a) ตำแหน่งมือที่ได้จากการติดตามด้วยคุณสมบัติของสี (b) การทำนายตำแหน่งมือที่อาจเป็นไปได้

2). การวัดค่าสถานะ (Measurement) เป็นการนำข้อมูลที่วัดได้จากเวลาปัจจุบันกับข้อมูลที่ได้อีกจากการทำนายจากเวลาที่ผ่านมามาทำการประมาณค่าใหม่ของสถานะเพื่อให้ได้ค่าที่เหมาะสมที่สุด โดยวัดค่าสถานะได้จากสมการที่ (2.17) มีค่าของการวัด (z_k) เป็นค่าตำแหน่งมือที่วัดได้จากการติดตามวัตถุด้วยคุณสมบัติของสีซึ่งก็คือ ค่าสถานะ (x_k) บวกด้วยสัญญาณรบกวนในการวัด (v_k) และมีเมตริกซ์ของการวัด (H : Measurements Matrix) ที่สอดคล้องกับค่าสถานะและค่าของการวัด โดยสามารถกำหนดค่าได้ดังนี้

$$z_k = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}^T \quad (3.6)$$

ตัวกรองคาลแมนจะทำงานโดยหาผลต่างระหว่างค่าที่ได้จากการทำนายกับค่าที่วัดได้จริง จากนั้นนำมาให้น้ำหนักโดยการคูณด้วยค่าคาลแมนเกน (Kalman Gain: K) แล้วนำผลที่ได้มาใช้แก้ไขค่าที่ได้จากการทำนายเพื่อประมาณค่าสถานะของระบบใหม่ให้เหมาะสมที่สุด



ภาพประกอบ 3-12 การประมาณค่าสถานะ (a) เฟรมภาพเริ่มต้น (b) การประมาณค่าสถานะของระบบ (c) การประมาณค่าสถานะเมื่อมีสัญญาณรบกวนอยู่ในระบบ

3.2. กระบวนการดึงคุณลักษณะเด่น (Feature Extraction)

3.2.1. ลักษณะเด่นของมือ (Hand's Feature)

ผลจากกระบวนการติดตามมือจะทำให้ได้บริเวณที่เป็นพื้นที่ของมือ จากบริเวณที่ได้มานี้สามารถที่จะนำมาหาหลักขเด่นของมือได้โดยใช้หลักการของภาพโมเมนต์ (Image Moment) ซึ่งก็คือ น้ำหนักเฉลี่ยของความเข้มของสีภายในจุดภาพ โดยภาพโมเมนต์ถูกใช้เพื่ออธิบายถึงคุณลักษณะของวัตถุหลังจากการแยกวัตถุออกจากพื้นหลัง คุณลักษณะที่สามารถหาได้จากภาพโมเมนต์ ได้แก่ จุดศูนย์กลางมวล (Centriod) และ ทิศทาง (Orientation) เป็นต้น สามารถนิยามภาพโมเมนต์ได้ดังนี้

$$M_{i,j} = \sum_x \sum_y x^i y^j I(x,y) \quad (3.7)$$

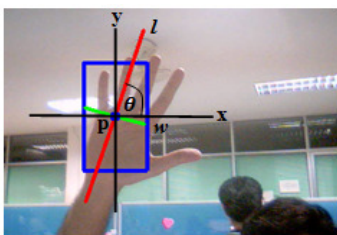
เมื่อ i, j เป็นลำดับของภาพโมเมนต์ x, y เป็นตำแหน่งของจุดภาพ และ $I(x, y)$ เป็นความเข้มของค่าสีภายในจุดภาพ การหาคุณลักษณะเด่นของมือสามารถทำได้โดยคำนวณจากภาพโมเมนต์ในลำดับต่างๆ สมการที่ (2.13) เป็นภาพโมเมนต์ลำดับที่ศูนย์ สมการที่ (2.14) และ (2.15) เป็นภาพโมเมนต์ลำดับที่หนึ่ง ซึ่งสามารถใช้ภาพโมเมนต์ทั้งสองลำดับมาคำนวณหาตำแหน่งศูนย์กลางมวลของบริเวณที่สนใจได้จากสมการที่ (2.12) ส่วนทิศทางของมือสามารถหาได้ภาพโมเมนต์ลำดับที่สองคำนวณได้จากสมการดังต่อไปนี้

$$\theta = \frac{1}{2} \arctan \left[\frac{2 \left(\frac{M_{11}}{M_{00}} - X_c Y_c \right)}{\left(\frac{M_{20}}{M_{00}} - X_c^2 \right) - \left(\frac{M_{02}}{M_{00}} - Y_c^2 \right)} \right] \quad (3.8)$$

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}}, \quad w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}} \quad (3.9)$$

$$a = \frac{M_{20}}{M_{00}} - X_c^2, \quad b = 2 \left(\frac{M_{11}}{M_{00}} - X_c Y_c \right), \quad c = \frac{M_{02}}{M_{00}} - Y_c^2 \quad (3.10)$$

เมื่อ θ เป็นมุมของวัตถุ l เป็นแกนยาวของวัตถุ และ w เป็นแกนสั้นของวัตถุ ซึ่งคุณลักษณะเด่นของมือที่หาได้จะถูกนำไปใช้ต่อไปในกระบวนการของการรู้จำท่าทางมือ

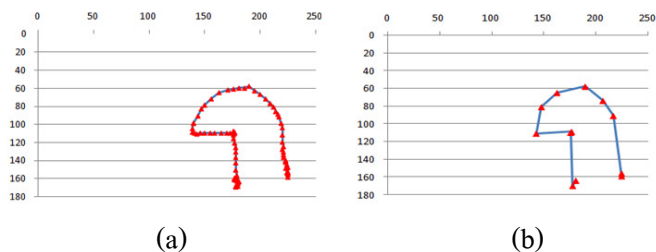


ภาพประกอบ 3-13 คุณลักษณะเด่นของมือ

3.2.2. การลดจำนวนข้อมูลของท่าทางมือ

ข้อมูลของท่าทางมือในที่นี้ก็คือ จุดศูนย์กลางมวลของมือที่ได้ทำการติดตามตลอดทุกเฟรมภาพของลำดับท่าทางนั้นๆ ซึ่งในระหว่างการทำท่าทางมือไม่ได้เคลื่อนที่ด้วยความเร็วมากนักจึงทำให้ตำแหน่งของมือไม่เปลี่ยนแปลงมากนักระหว่างเฟรมภาพเช่นกัน ดังนั้นข้อมูลจุดของท่าทางไม่จำเป็นต้องถูกพิจารณาทุกๆ เฟรมภาพของลำดับท่าทาง ด้วยสาเหตุดังกล่าวจึงได้มีการเลือกเฉพาะข้อมูลจุดที่สำคัญเพื่อเก็บเป็นลำดับของท่าทางนั้นใหม่แต่ยังคงลักษณะของรูปร่างเอาไว้ให้ใกล้เคียง

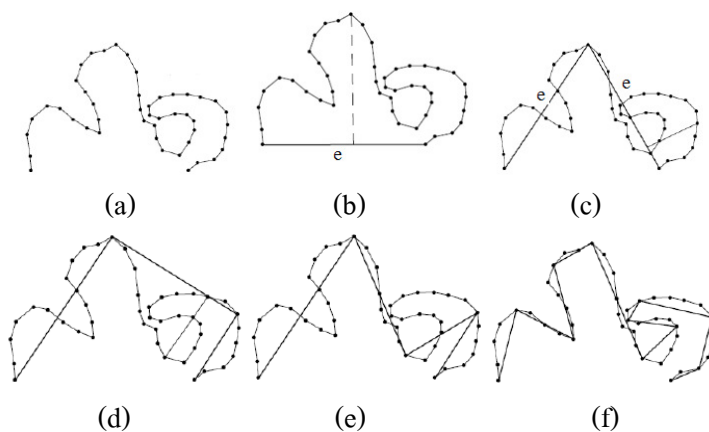
ของเดิม การทำเช่นนี้จะเป็นการช่วยลดหน่วยความจำและเพิ่มความเร็วในการเทียบเท่าของท่าทาง
ในกระบวนการรู้จำท่าทาง



ภาพประกอบที่ 3-14 การลดจำนวนข้อมูลของท่าทาง (a) ข้อมูลจุดตั้งต้น (97 จุดข้อมูล) (b) ผลลัพธ์
ของการลดจำนวนข้อมูล (12 จุดข้อมูล)

วิธีการที่นำมาใช้เพื่อเลือกข้อมูลจุดสำคัญคือ การประมาณค่ารูปหลายเหลี่ยม (Polygon Approximations) ซึ่งถูกนำเสนอโดย Douglas และ Peucker [14] โดยวิธีการประมาณค่ารูปหลายเหลี่ยมมีขั้นตอนการทำงานดังต่อไปนี้

1. ให้ลำดับข้อมูลจุดของท่าทางเป็นดังภาพประกอบที่ 3-15 (a)
2. ลากเส้นจากจุดเริ่มต้นของข้อมูลจุดไปยังจุดสุดท้ายของข้อมูลจุดดังภาพประกอบที่ 3-14 (b) กำหนดให้เส้นนี้เป็นเส้นตรง e
3. จากนั้นจะทำการหาจุดที่มีระยะห่างที่มากที่สุดจากกับเส้นตรง e ที่มีค่ามากกว่าค่าที่กำหนดไว้มากที่สุด กำหนดให้ ϵ เป็นระดับของค่าระยะห่างที่ยอมรับได้ เพื่อทำการประมาณเส้นตรง e ใหม่ (ภาพประกอบที่ 3-15 (c)) โดยการลากเส้นเพิ่มไปยังตรงจุดที่มีค่าสูงที่สุดนั้น ทำเช่นนี้ไปเรื่อยๆ จนกระทั่งแนวเส้นที่มากที่สุดจากมีค่าภายใต้ค่าที่กำหนดไว้จึงได้ลำดับของท่าทางนั้นใหม่แต่ยังคงลักษณะของรูปร่างเอาไว้ให้ใกล้เคียงของเดิมดังภาพประกอบที่ 3-15 (f)



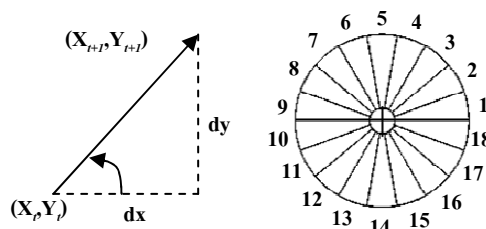
ภาพประกอบ 3-15 ขั้นตอนการทำงานของการประมาณค่ารูปหลายเหลี่ยม

3.2.3. การควอนไทซ์เวกเตอร์ (Vector Quantization)

เนื่องจากท่าทางมืออธิบายได้เป็นตำแหน่งของมือในทุกๆ เฟรมภาพตามช่วงเวลาที่ต่อเนื่องกัน เป็นเวกเตอร์ ซึ่งจะบอกถึงทิศทางและการเคลื่อนที่ของมือในท่าทางนั้นๆ ทิศทางสามารถหาได้จากการคำนวณหามุมที่เกิดขึ้นระหว่างจุด 2 จุด ดังสมการที่ (3.15)

$$\theta_t = \arctan\left(\frac{Y_{t+1} - Y_t}{X_{t+1} - X_t}\right); t = 1, 2, \dots, T-1 \quad (3.11)$$

เมื่อ T เป็นความยาวทั้งหมดของท่าทาง โดยในงานวิจัยชิ้นนี้จะกำหนดทิศทางออกเป็นรหัส 18 แบบ ซึ่งเกิดจากการแบ่งมุมตั้งแต่ 0° - 360° องศา ออกเป็น 18 ช่วง แต่ละช่วงมีค่า 20° องศา



ภาพประกอบที่ 3-16 การควอนไทซ์เวกเตอร์

รหัสทั้ง 18 แบบจะใช้ในการอธิบายลักษณะของท่าทางที่ใช้เป็นคุณลักษณะเด่นสำหรับกระบวนการรู้จำท่าทางโดยใช้แบบจำลองฮิดเดนมาร์คอฟ จากภาพประกอบที่ 3-14 (b) สามารถอธิบายเป็นชุดรหัสได้ดังนี้ $v = \{13, 5, 3, 10, 5, 3, 1, 16, 16, 14, 5\}$

ตาราง 3-2 การแปลงข้อมูลจุดของท่าทาง (ภาพประกอบที่ 3-14(b)) ให้เป็นชุดรหัส

ข้อมูลจุด	มุม(องศา)	รหัสข้อมูล
(181,164)	-	-
(178,170)	243.43	13
(176,110)	91.90	5
(177,109)	45.00	3
(143,111)	183.36	10
(148,81)	80.53	5
(163,65)	46.84	3
(190,58)	14.53	1
(207,74)	316.73	16
(217,91)	300.46	16

ข้อมูลจุด	มุม(องศา)	รหัสข้อมูล
(225,159)	276.70	14
(225,156)	90.00	5

3.3. กระบวนการรู้จำท่าทางมือ (Hand Gesture Recognition)

ส่วนสุดท้ายของระบบจะเป็นส่วนของกระบวนการรู้จำท่าทางมือเพื่อใช้เป็นชุดคำสั่งในการควบคุมระบบคอมพิวเตอร์หรือเครื่องกล โดยท่าทางมือในที่นี้จะแบ่งได้เป็น 2 แบบ คือ

1. ท่าทางมืออย่างง่าย คือ การเคลื่อนมือไปทางซ้าย-ขวา บน-ล่าง เป็นต้น ซึ่งจะใช้วิธีการไฟไนต์สเตตแมชชีนในการรู้จำท่าทางมือ

2. ท่าทางมือแบบสัญลักษณ์ คือ การเคลื่อนมือเป็นรูปทรงต่างๆ เช่น วงกลม สามเหลี่ยม หรือสี่เหลี่ยม เป็นต้น ซึ่งจะใช้วิธีการแบบจำลองฮิดเดนมาร์คอฟในการรู้จำท่าทางมือ

3.3.1. การรู้จำท่าทางมือด้วยไฟไนต์สเตตแมชชีน (Hand Gesture Recognition Using Finite State Machine)

ไฟไนต์สเตตแมชชีน[15] เป็นการออกแบบการทำงานของระบบเป็นหลายๆ สถานะ แต่ละสถานะมีเงื่อนไขในการทำงานที่ต่างกัน โดยสถานะจะถูกกำหนดเป็นท่าทางมือในรูปแบบต่างๆ เพื่อใช้ในการรู้จำเป็นชุดคำสั่ง ซึ่งเงื่อนไขจะถูกกำหนดจากคุณลักษณะเด่นของมือที่หามาได้คือ จุดศูนย์กลางมวล (p_t) มุม (θ_t) ขนาด (z_t) และระยะทาง (d_t) สามารถสรุปได้เป็น

$$f(t) = [p_t \ \theta_t \ z_t \ d_t] \quad (3.12)$$

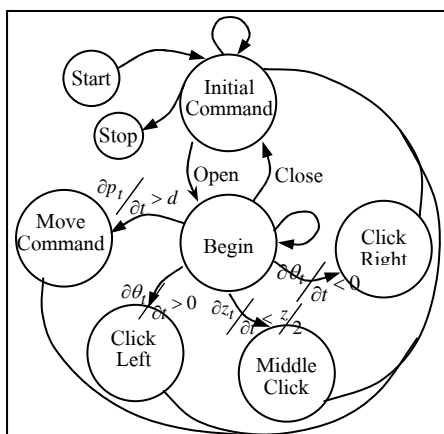
โดยจุดศูนย์กลางมวลสามารถหาได้จากสมการที่ (2.12) มุมหาได้จากสมการที่ (3.9) ส่วนขนาดและระยะทางสามารถหาได้จาก

$$z = l x w \quad (3.13)$$

$$d = \sqrt{l^2 x w^2} \quad (3.14)$$

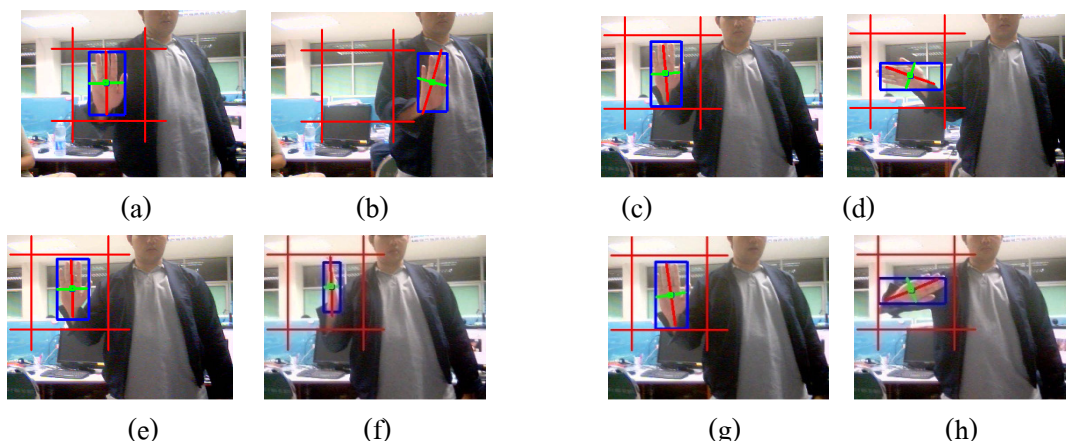
ชุดคำสั่งท่าทางมือที่ใช้ในการรู้จำจะประกอบไปด้วย (1) คำสั่งของการเคลื่อนที่ 4 ทิศทาง (Move Command) คือ บน-ล่าง ซ้าย-ขวา (2) คำสั่งของการกดปุ่มเมาส์ (Click Command) คือ กดปุ่มเมาส์ ซ้าย ขวา และกลาง ซึ่งชุดคำสั่งจะทำหน้าที่เหมือนอุปกรณ์เมาส์ที่ใช้ในการควบคุมระบบคอมพิวเตอร์ สำหรับคำสั่งของการเคลื่อนที่จะใช้การเคลื่อนที่ของคุณลักษณะเด่นจุดศูนย์กลางมวล ($\frac{dp_t}{dt}$) เป็นเงื่อนไขในการรู้จำคำสั่ง เมื่อระบบตรวจจับลักษณะมือได้รอบสี่เหลี่ยมขนาดความ

กว้างยาวเท่ากับคุณลักษณะเด่นระยะทางจะถูกสร้างล้อมรอบบริเวณมือ เมื่อมีการเคลื่อนที่ของจุดศูนย์กลางมวลออกนอกกรอบสี่เหลี่ยมไปยังทิศทางใดๆ ระบบจะทำการรู้จำคำสั่งของการเคลื่อนที่ตามทิศทางที่ถูกระบุเอาไว้ สำหรับคำสั่งของการกดปุ่มเมาส์ จะใช้การเปลี่ยนแปลงของคุณลักษณะเด่นมุม ($\partial\theta/\partial t$) และคุณลักษณะเด่นขนาด ($\partial z_1/\partial t$) เป็นเงื่อนไขในการรู้จำคำสั่ง โดยคำสั่งกดปุ่มเมาส์ซ้ายมีเงื่อนไขคือ คุณลักษณะเด่นมุมต้องเปลี่ยนแปลงเป็นค่าบวก คำสั่งกดปุ่มเมาส์ขวามีเงื่อนไขคือคุณลักษณะเด่นมุมต้องเปลี่ยนแปลงเป็นค่าลบ และคำสั่งกดปุ่มเมาส์กลางมีเงื่อนไขคือ คุณลักษณะเด่นขนาดต้องน้อยกว่าครึ่งหนึ่งของคุณลักษณะเด่นขนาดตอนเริ่มต้น สามารถเขียนสรุปให้อยู่ในรูปแบบของไฟไนต์สเตตแมชชีนได้ดังนี้



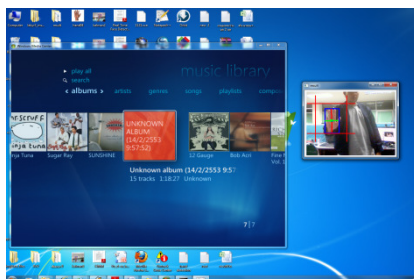
ภาพประกอบ 3-17 ไฟไนต์สเตตแมชชีนสำหรับการรู้จำคำสั่ง

ภาพประกอบ 3-18 แสดงให้เห็นถึงท่าทางมือที่ใช้แทนคำสั่งของการเคลื่อนที่ คำสั่งกดปุ่มเมาส์ซ้าย กลาง และขวา ตามลำดับ



ภาพประกอบ 3-18 การรู้จำท่าทางมือสำหรับใช้เป็นชุดคำสั่ง (a-b) คำสั่งของการเคลื่อนที่ (c-d) คำสั่งกดปุ่มเมาส์ซ้าย (e-f) คำสั่งกดปุ่มเมาส์กลาง (g-h) คำสั่งกดปุ่มเมาส์ขวา

ชุดคำสั่งถูกนำไปประยุกต์ใช้กับโปรแกรมวินโดวมีเดียเซ็นเตอร์ (Window Media Center) ในระบบปฏิบัติการวินโดว 7 (Windows 7) ซึ่งสามารถเข้าไปดูตัวอย่างของระบบได้ที่ <http://www.youtube.com/watch?v=loVxPEP5fME>



ภาพประกอบ 3-19 ระบบรู้จำท่าทางมือที่ประยุกต์ใช้เพื่อควบคุมโปรแกรมวินโดวมีเดียเซ็นเตอร์

3.3.2. การรู้จำท่าทางมือด้วยแบบจำลองฮิดเดนมาร์คอฟ (Hand Gesture Recognition Using Hidden Markov Model)

แบบจำลองฮิดเดนมาร์คอฟ เป็นแบบจำลองทางสถิติที่แสดงความเป็นไปได้ของการกระทำหรือเหตุการณ์ที่เป็นไปได้ ซึ่งความเป็นไปได้ของการกระทำหรือเหตุการณ์ที่เป็นไปได้อาจถูกกำหนดให้เป็นสถานะซ่อนและมีความน่าจะเป็นของการเปลี่ยนสถานะซ่อนเชื่อมโยงอยู่ถึงกันทุกๆ สถานะซ่อน องค์ประกอบที่สำคัญของแบบจำลองฮิดเดนมาร์คอฟประกอบด้วย

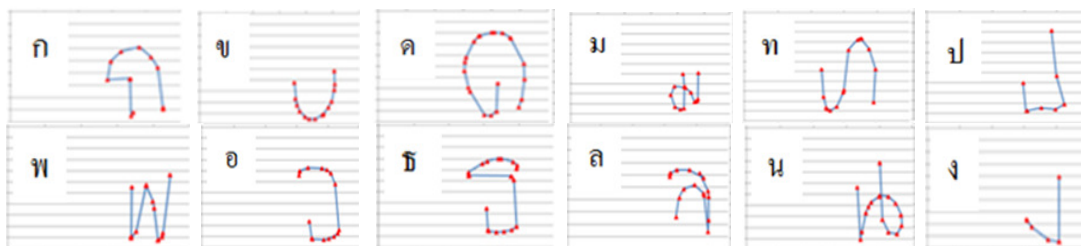
1. จำนวนสถานะ (N) ในที่นี้จะใช้สถานะทั้งหมด 10 สถานะ โดยแบ่งเป็น 8 สถานะซ่อน ส่วนอีก 2 สถานะสำหรับสถานะเริ่มต้นและสถานะสิ้นสุด โดยสถานะซ่อนจะพิจารณาจากทิศทางการเคลื่อนที่ของมือในท่าทางต่างๆ ซึ่งมีการเคลื่อนที่ใน 8 ทิศทางหลัก คือ บน-ล่าง ซ้าย-ขวา เฉียงบนซ้าย-ขวา และเฉียงล่างซ้าย-ขวา

2. จำนวนเหตุการณ์ต่อสถานะ (M) คือ เหตุการณ์ทั้งหมดที่มีอยู่ในระบบ ซึ่งได้จากจำนวนรหัสข้อมูลทั้ง 18 แบบ $O = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18\}$ ที่ใช้เป็นชุดรหัสเพื่อบอกลำดับของท่าทาง

3. เมตริกซ์ขนาด 10 แถว 10 หลัก ที่ภายในจะระบุถึงความน่าจะเป็นในการเปลี่ยนสถานะจากสถานะ i ไปสถานะ j (A)

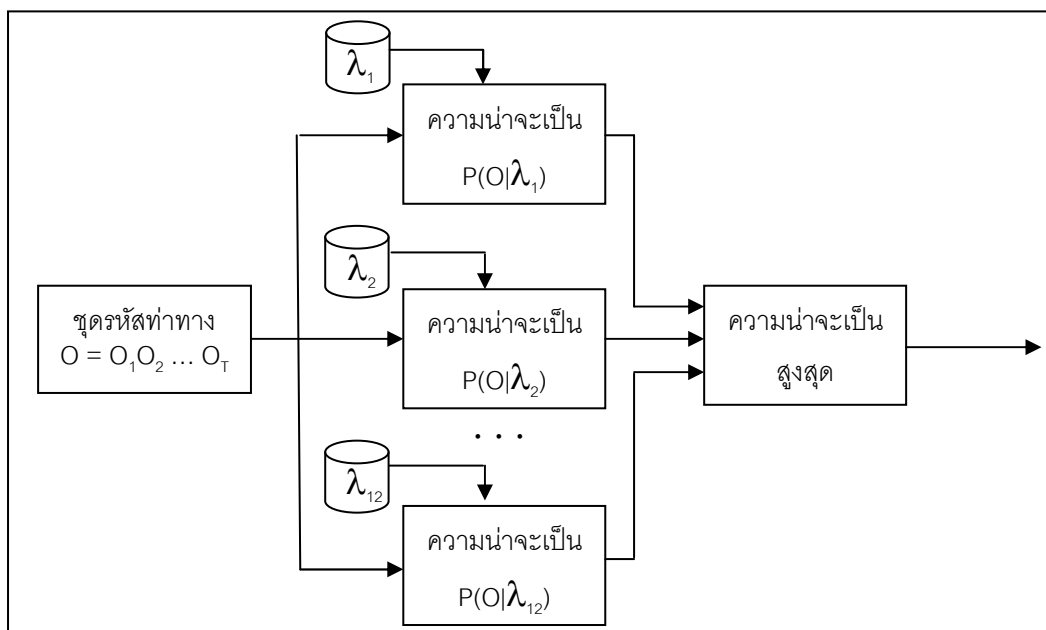
4. เมตริกซ์ขนาด 10 แถว 18 หลักที่ภายในจะระบุถึงความน่าจะเป็นที่จะเกิดเหตุการณ์ O_i ที่เกิดในสถานะ j (B)

5. เมตริกซ์ขนาด 1 แถว 10 หลักที่ภายในจะระบุถึงความน่าจะเป็นของสถานะแรกเริ่ม (π_i)



ภาพประกอบ 3-20 ท่าทางทั้ง 12 แบบจากตัวอักษรภาษาไทย

ในขั้นตอนการฝึกสอนจะใช้ชุดรหัสทั้งหมด 50 ชุดสำหรับการฝึกสอนในแต่ละท่าทางทั้ง 12 แบบ เมื่อขั้นตอนการฝึกสอนเสร็จสิ้นจะทำให้ได้แบบจำลองทั้ง 12 แบบของแต่ละท่าทางเพื่อใช้หาความน่าจะเป็นเมื่อมีชุดรหัสท่าทางที่ต้องการรู้จำเข้ามาภายในระบบ การรู้จำท่าทางเกิดจากการนำชุดรหัสท่าทางไปหาความน่าจะเป็นด้วยวิธีการหาความน่าจะเป็นแบบไปหน้ากับแบบจำลองทั้ง 12 แบบ และเลือกแบบจำลองที่ให้ความน่าจะเป็นสูงสุดเพื่อว่าบอกว่าชุดรหัสท่าทางที่เข้ามามีความเป็นไปได้สูงสุดที่จะตรงกับแบบจำลองของท่าทางที่เลือกมา ซึ่งสามารถเข้าไปดูตัวอย่างของระบบรู้จำท่าทางตัวอักษรภาษาไทยได้ที่ <http://www.youtube.com/watch?v=jEZi02EuweY>



ภาพประกอบที่ 3-21 ขั้นตอนการทำงานของกระบวนการรู้จำท่าทาง

3.4. สรุป

เนื้อหาบทนี้กล่าวถึงกระบวนการรู้จำท่าทางมือซึ่งสามารถแบ่งออกเป็นขั้นตอนหลักๆ 3 ขั้นตอน ได้แก่ (1) กระบวนการตรวจจับและติดตามลักษณะมือสำหรับใช้ระบุตำแหน่งของมือที่กำลังเคลื่อนที่อยู่ภายในภาพวิดีโอ (2) กระบวนการดึงคุณลักษณะเด่นสำหรับหาลักษณะเด่นต่างๆ ของมือ เช่น จุดศูนย์กลางมวล ทิศทาง ขนาด เป็นต้น เพื่อใช้เป็นเงื่อนไขในกระบวนการรู้จำ (3) กระบวนการรู้จำท่าทางสำหรับการนำท่าทางที่ได้ออกแบบมาใช้เป็นชุดคำสั่งต่างๆ เพื่อนำไปประยุกต์ใช้เข้ากับงานในด้านต่างๆ อย่างเหมาะสม

บทที่ 4

ผลการทดสอบ

4.1. ผลการทดลองกระบวนการตรวจจับลักษณะมือ (Hand Detection)

การทดลองจะใช้กลุ่มภาพตัวอย่างที่ใช่ 5,058 ภาพ เป็นภาพมือในลักษณะตั้งตรงขนานกับกล้อง และใช้กลุ่มภาพตัวอย่างที่ไม่ใช่ 10,448 ภาพ เป็นภาพของสภาพแวดล้อมต่างๆ เช่น อาคาร สิ่งก่อสร้าง ทิวทัศน์ หรือส่วนอื่นๆ ของร่างกาย โดยไม่มีสิ่งที่กำลังสนใจอยู่ในภาพ (ภาพประกอบที่ 3-1) ความละเอียดของภาพทั้ง 2 กลุ่มตัวอย่างมีขนาด 320x240 จุดภาพ กลุ่มตัวอย่างทั้งหมดจะถูกนำเข้าสู่กระบวนการฝึกสอนด้วยการเรียนรู้แบบออบดาบัส ในกระบวนการฝึกสอนภาพกลุ่มตัวอย่างทั้ง 2 กลุ่มจะลดขนาดของภาพลงและแปลงให้อยู่ในรูปแบบของภาพระดับสีเทา (Gray Scale Image) เพื่อลดหน่วยความจำที่จะนำมาใช้ในระหว่างกระบวนการฝึกสอน ในระหว่างกระบวนการฝึกสอนจะมีค่ากำหนดอยู่ 2 ค่าที่ใช้เป็นเกณฑ์ในการตัดสินใจว่าควรหยุดทำการฝึกสอนหรือทำการฝึกสอนต่อไป ซึ่งค่ากำหนดทั้ง 2 คือ ค่าอัตราการเจอ (Hit Rate) เป็นค่าที่ได้จากการตรวจจับวัตถุว่าใช่สิ่งที่สนใจและวัตถุนั้นก็เป็นสิ่งที่สนใจ และอัตราความผิดพลาดเชิงบวก (False Positive Rate) เป็นค่าที่ได้จากการตรวจจับวัตถุว่าใช่แต่วัตถุนั้นไม่ใช่สิ่งที่สนใจ ซึ่งค่าอัตราการเจอและอัตราความผิดพลาดเชิงบวกทั้งหมดของการฝึกสอนสามารถคำนวณได้จาก

$$\text{Hit Rate} = \text{Min Hit Rate}^{n\text{Stages}} \quad (4.1)$$

$$\text{False Positive Rate} = \text{Max False Positive Rate}^{n\text{Stages}} \quad (4.2)$$

ค่าอัตราการเจอ คือ ค่าของอัตราการเจอน้อยสุด (Min Hit Rate) ซึ่งกำหนดค่ามาตรฐานเป็น 0.995000 ยกกำลังด้วยจำนวนสถานะที่ใช้ในการฝึกสอนทั้งหมด (nStages) ส่วนค่าอัตราความผิดพลาดเชิงบวก คือ ค่าอัตราความผิดพลาดเชิงบวกมากที่สุด (Max False Positive Rate) ซึ่งกำหนดค่ามาตรฐานเป็น 0.500000 ยกกำลังด้วยจำนวนสถานะที่ใช้ในการฝึกสอนทั้งหมด ค่าอัตราการเจอและอัตราความผิดพลาดเชิงบวกสังเกตได้จากผลที่แสดงระหว่างการฝึกสอนในแต่ละสถานะ โดยค่าอัตราการเจอจะอยู่หลังค่าจำนวนภาพที่ใช่และค่าอัตราผิดพลาดเชิงบวกจะอยู่หลังค่าจำนวนภาพที่ไม่ใช่ในการฝึกสอนตามสถานะนั้นๆ (ภาพประกอบ 4-1) ในการฝึกสอนได้มีทดลองประสิทธิภาพในการตรวจจับของการฝึกสอนในแต่ละแบบ โดยได้ทำการฝึกสอนเป็นจำนวน 4 ครั้ง ด้วยจำนวนภาพกลุ่มตัวอย่างที่เท่ากันแต่จำนวนสถานะการฝึกสอนแตกต่างกัน ผลลัพธ์ของประสิทธิภาพการตรวจจับลักษณะของมือในการฝึกสอนแต่ละแบบสามารถแสดงได้ดังตารางที่ 4-1

Tree Classifier															
Stage															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 --- 1 --- 2 --- 3 --- 4 --- 5 --- 6 --- 7 --- 8 --- 9 --- 10 --- 11 --- 12 --- 13 --- 14 --- 15															
Parent node: 15															
*** 1 cluster ***															
POS: 4684 5058 0.926058															
NEG: 9675 3.17986e-005															
BACKGROUND PROCESSING TIME: 1485.07															
Precalculation time: 9.69															

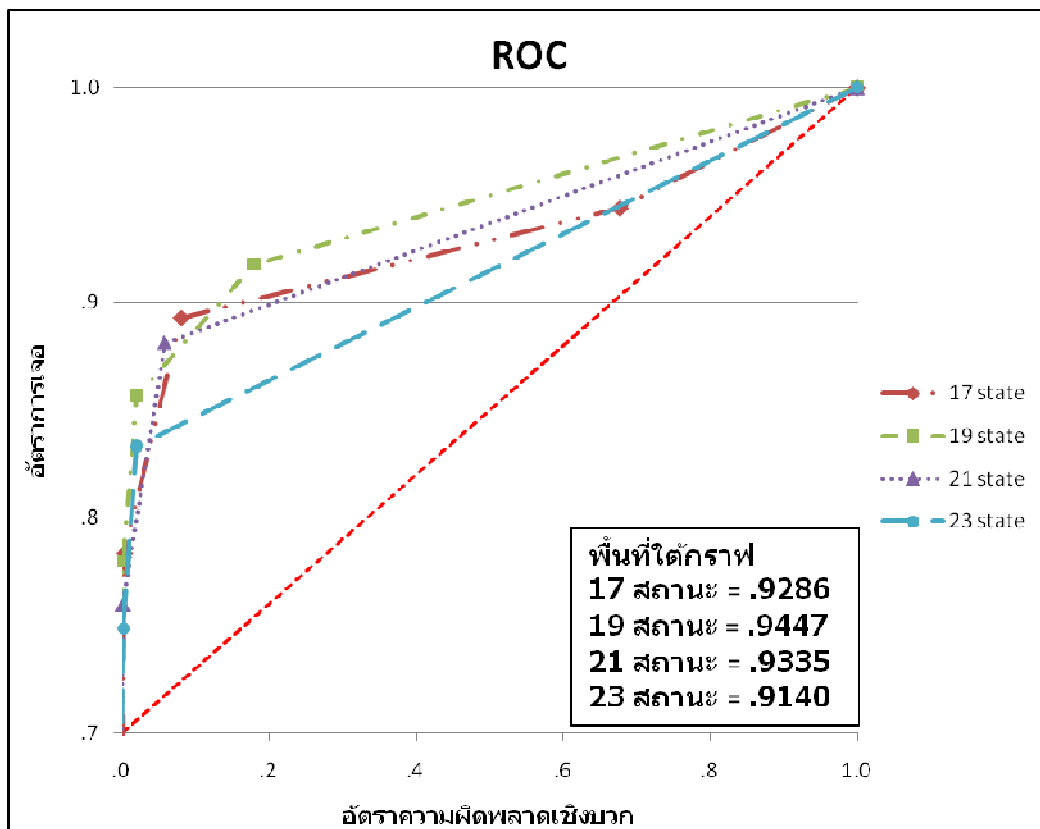
ภาพประกอบ 4-1 ตัวอย่างบันทึกข้อมูลของการฝึกสอน

ตาราง 4-1 ประสิทธิภาพการตรวจจับลักษณะมือของการฝึกสอนแต่ละสถานะ

	17 สถานะ	19 สถานะ	21 สถานะ	23 สถานะ
อัตราการเจอ	94.3%	91.7%	88.1%	83.3%
อัตราความผิดพลาดเชิงบวก	67.7%	17.8%	5.8%	1.9%

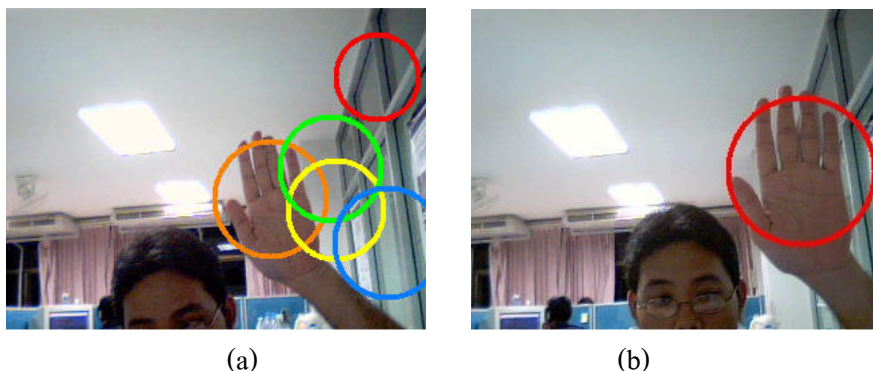
จากตาราง 4-1 ทำการทดสอบประสิทธิภาพการตรวจจับของการฝึกสอนแบบต่างๆ โดยทดสอบกับกลุ่มภาพตัวอย่างที่ใช่จำนวน 5,058 ภาพ การฝึกสอนที่ 17 สถานะจะให้อัตราการเจอที่มีค่าสูงสุดแต่อัตราความผิดพลาดเชิงบวกก็มีค่าสูงเช่นกัน ซึ่งทำให้ตัวตรวจจับยังไม่สามารถแยกแยะภาพพื้นหลังกับวัตถุที่สนใจได้อย่างชัดเจนจึงไม่เหมาะที่จะนำมาใช้งาน (ภาพประกอบ 4-3 (a)) เมื่อทำการฝึกสอนต่อไปการฝึกสอนที่ 21 และ 23 สถานะจะทำให้อัตราความผิดพลาดลดลงแต่ก็ส่งผลให้อัตราการเจอลดลงเช่นกันเนื่องจากเมื่อทำการฝึกสอนด้วยสถานะที่มากขึ้นต่อไปจะให้ฟังก์ชันการตรวจจับมีความเฉพาะมากขึ้น คือ วัตถุที่จะนำมาตรวจจับจะต้องมีลักษณะที่คล้ายกับกลุ่มตัวอย่างที่ใช้ในการฝึกสอนมากยิ่งขึ้นจึงจะสามารถตรวจจับได้ ซึ่งก็ไม่เหมาะสมที่จะนำมาใช้งานเช่นกัน ดังนั้นในระบบจึงเลือกใช้ตัวตรวจจับลักษณะของมือที่ได้ทำการฝึกสอนเป็นจำนวน 19 สถานะ ซึ่งให้อัตราการเจอที่ 91.7 % และอัตราความผิดพลาดเชิงบวกที่ 17.8 % (ภาพประกอบ 4-3 (b)) ซึ่งน่าจะเป็นค่าที่เหมาะสมที่จะนำมาใช้งานจริง เพื่อเป็นการยืนยันถึงประสิทธิภาพค่าการฝึกสอนที่ได้เลือกใช้จึงได้นำข้อมูลที่ได้จากการฝึกสอนแต่ละสถานะมาแสดงให้อยู่ในรูปแบบ

กราฟ ROC (Receiver Operating Characteristic) ซึ่งแสดงความสัมพันธ์ระหว่างอัตราการเจอและอัตราความผิดพลาดเชิงบวกเพื่อบอกถึงประสิทธิภาพการทำงานของระบบ โดยประสิทธิภาพการทำงานหาได้จากพื้นที่ใต้กราฟ ROC ซึ่งมีค่าอยู่ระหว่าง 0.0 ถึง 1.0 ถ้าพื้นที่ใต้กราฟเข้าใกล้ค่า 1.0 แสดงถึงระบบมีประสิทธิภาพการทำงานสูงแต่ถ้าพื้นที่ใต้กราฟต่ำกว่า 0.5 แสดงถึงระบบมีประสิทธิภาพในการทำงานต่ำ



ภาพประกอบ 4-2 กราฟ ROC ของการฝึกสอนแต่ละสถานะ

จากภาพประกอบ 4-2 แสดงกราฟ ROC ของการฝึกสอนแต่ละสถานะ ซึ่งพื้นที่ใต้กราฟของแต่ละสถานะมีค่าเข้าใกล้ 1.0 เหมือนกันหมดแต่จะเห็นได้ว่าการฝึกสอน 19 สถานะมีพื้นที่ใต้กราฟมากที่สุดและค่อยลดต่ำลงเมื่อจำนวนสถานะเพิ่มขึ้น ดังนั้นค่าการฝึกสอนที่ 19 สถานะจึงเป็นค่าที่เหมาะสมที่จะนำมาใช้งานจริง



ภาพประกอบ 4-3 การตรวจจับลักษณะมือ (a) การฝึกสอน 17 สถานะ (b) การฝึกสอน 19 สถานะ

4.2. ผลการทดลองกระบวนการติดตามการเคลื่อนที่ของมือ (Hand Tracking)

การทดลองจะทำการเคลื่อนที่มือเป็น 3 รูปแบบคือ เส้นตรง (Straight Line) เส้นโค้ง (Curve) และเส้นรูปวี (V-shape) เพื่อเปรียบเทียบความผิดพลาดในการติดตามการเคลื่อนที่ของมือระหว่างผลที่ได้จากการติดตามมือโดยใช้คุณสมบัติของสี และผลที่ได้จากการติดตามมือโดยใช้ตัวกรองคาลแมน กับเส้นทางอุดมคติทั้ง 3 แบบ (Ideal Path) โดยทำการทดลองเคลื่อนที่มือในแต่ละแบบเป็นจำนวน 10 ครั้ง การหาความผิดพลาดในการติดตามการเคลื่อนที่ของมือหาได้จากสมการที่

$$e = |v - v_{measure}| \quad (4.3)$$

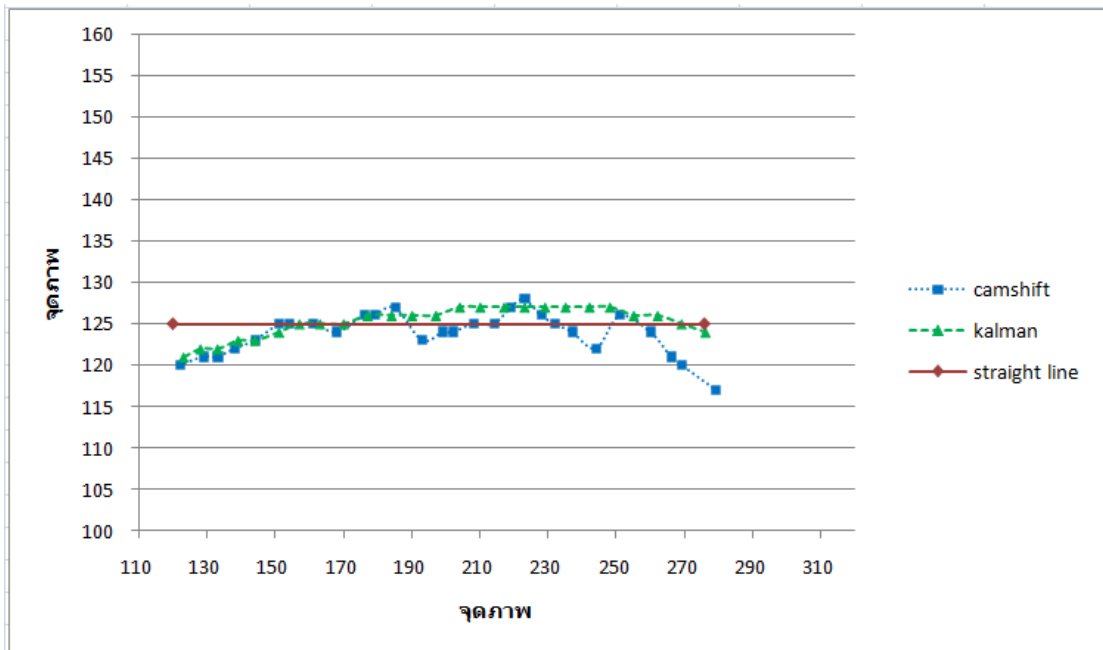
ความผิดพลาดในการติดตามการเคลื่อนที่ของมือ (e) คือ ความแตกต่างระหว่างค่าตำแหน่งการเคลื่อนที่ของมือที่ได้จากการวัด ($v_{measure}$) กับค่าตำแหน่งของเส้นทางอุดมคติ (v) ณ เวลาเดียวกัน

ตาราง 4-2 การทดสอบความผิดพลาดในการติดตามการเคลื่อนที่ของมือ

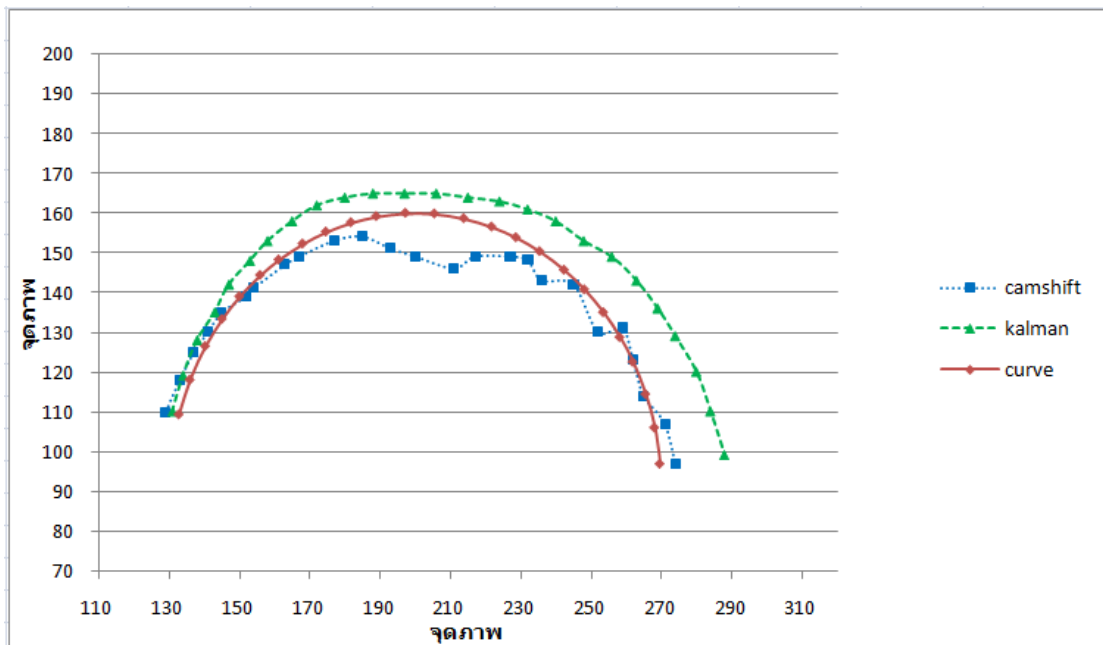
ทดสอบความผิดพลาด (จุดภาพ)						
จำนวน	เส้นตรง		เส้นโค้ง		เส้นรูปตัววี	
	Camshift	Kalman	Camshift	Kalman	Camshift	Kalman
1	1.40	1.54	3.43	3.79	8.44	8.05
2	1.66	1.70	3.96	3.61	3.39	3.31
3	1.53	1.39	7.30	6.87	4.31	4.22
4	1.74	1.59	6.46	6.68	9.22	9.13

ทดสอบความผิดพลาด (จุดภาพ)						
จำนวน	เส้นตรง		เส้นโค้ง		เส้นรูปตัววี	
	Camshift	Kalman	Camshift	Kalman	Camshift	Kalman
5	2.10	1.97	7.17	7.55	5.38	4.98
6	1.02	0.97	8.39	8.93	3.44	3.51
7	1.50	1.40	7.68	8.09	4.50	4.31
8	2.06	2.02	4.54	4.23	4.58	4.44
9	1.81	1.76	5.61	6.05	7.19	7.05
10	1.71	1.84	6.33	6.97	6.55	6.38
μ	1.65	1.61	6.08	6.27	5.70	5.53
σ	0.30	0.29	1.57	1.74	1.94	1.89

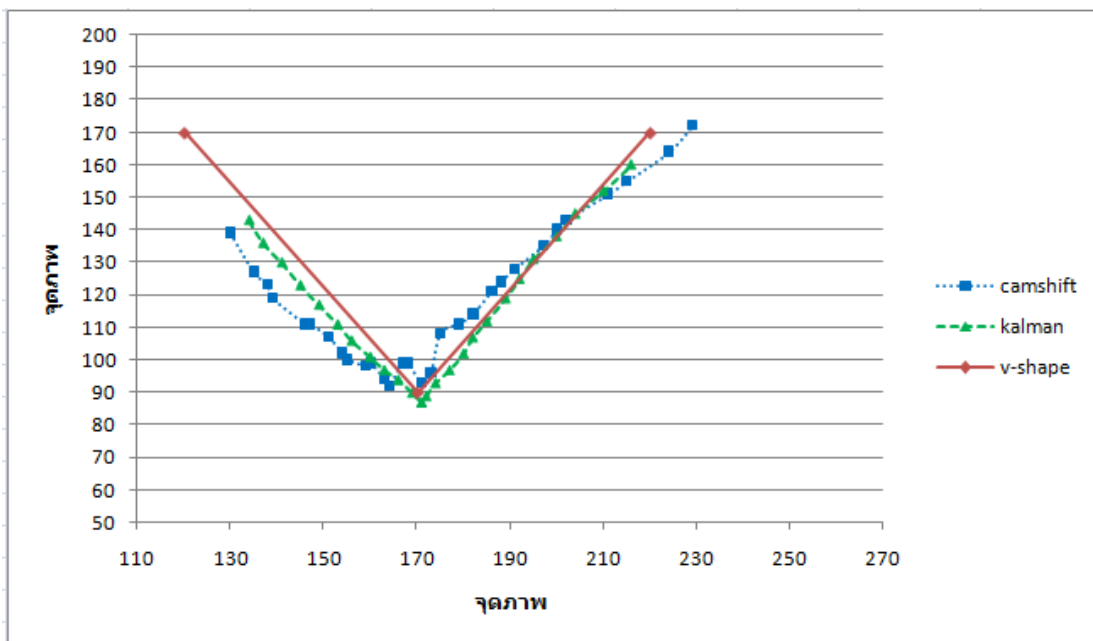
จากภาพประกอบ 4-3 (a) (b) และ (c) จะแสดงตำแหน่งของมือที่ได้จากการติดตามโดยใช้คุณสมบัติของสี (เส้นสีฟ้า) และใช้ตัวกรองคาลแมน (เส้นสีเขียว) เพื่อหาความผิดพลาดกับเส้นอุดมคติ (เส้นสีน้ำตาล) จะเห็นได้ว่าการติดตามมือโดยใช้คุณสมบัติของสีเมื่อทดลองเคลื่อนที่ตามเส้นอุดมคติทั้ง 3 แบบจุดข้อมูลที่ได้มีเกิดการแกว่งของจุดข้อมูลอยู่มากเนื่องจากการเปลี่ยนแปลงของสีที่เกิดขึ้นอยู่ตลอดทุกเฟรมภาพทำให้เกิดการเปลี่ยนตำแหน่งไปยังทิศทางต่างๆ ตามความน่าจะเป็นของสีภายในเฟรมภาพนั้นๆ ส่วนการติดตามมือโดยใช้ตัวกรองคาลแมนจุดข้อมูลที่ได้จะมีค่าคงที่มากกว่าเนื่องจากตัวกรองคาลแมนจะทำการประมาณค่าจุดข้อมูลที่เหมาะสมจากค่าจากทำนายและค่าที่วัดได้จริง จากตารางที่ 4-2 แสดงถึงความผิดพลาดในการติดตามการเคลื่อนที่ของมือในรูปแบบของจุดภาพตามการเคลื่อนที่ทั้ง 3 แบบ พบว่าความผิดพลาดเฉลี่ยของการเคลื่อนที่แบบเส้นโค้งมีมากกว่าการเคลื่อนที่แบบเส้นตรงและรูปตัววี เนื่องจากตัวกรองคาลแมนจะประมาณค่าในลักษณะเป็นเส้นตรงจึงทำให้จุดข้อมูลมีการเบนออกจากวิถีโค้งออกไป (ภาพประกอบ 4-3 (c)) แต่ถึงแม้จะมีความผิดพลาด ระบบก็ยังจำเป็นต้องใช้ค่าที่ได้จากตัวกรองคาลแมนเนื่องจากการประมาณค่าที่เหมาะสมทำให้การเคลื่อนที่ของมือใกล้เคียงกับเส้นอุดมคติมากขึ้นจึงเป็นค่าที่เหมาะสมที่จะนำไปใช้งาน



(a)



(b)



(c)

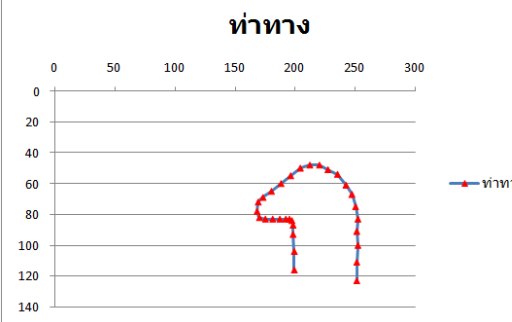
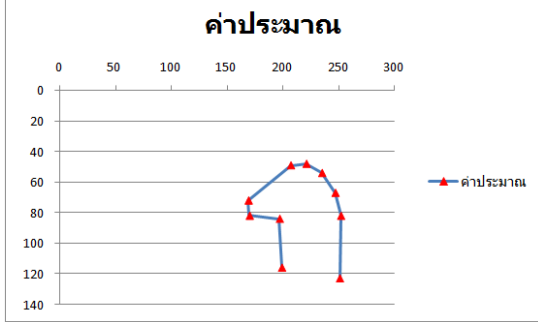
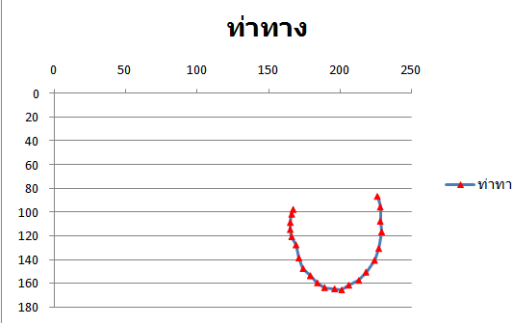
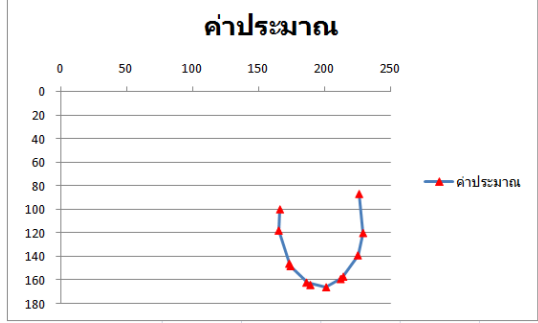
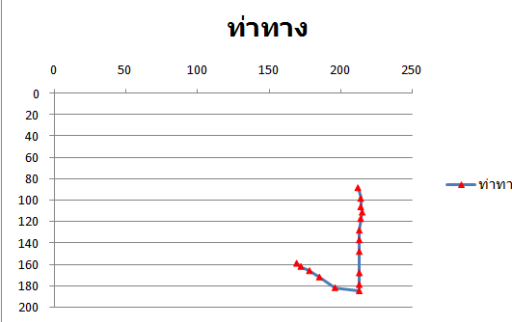
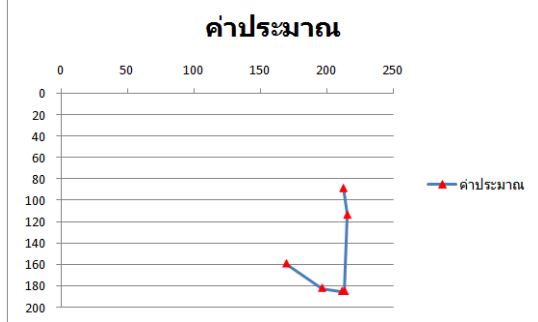
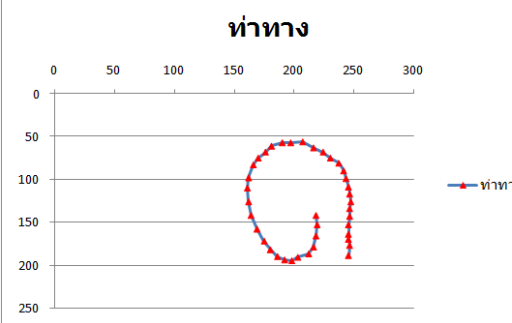
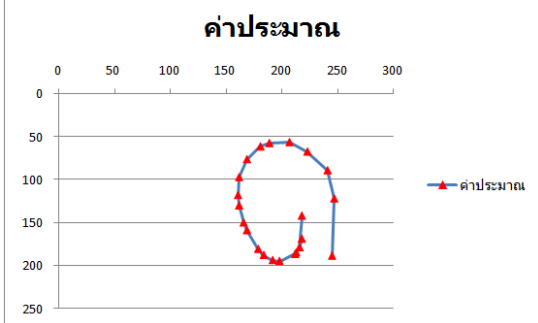
ภาพประกอบ 4-4 ทดสอบความผิดพลาดในการติดตามการเคลื่อนที่ของมือ

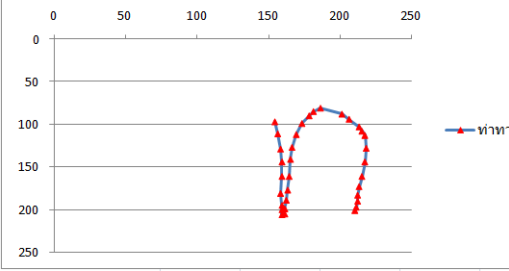
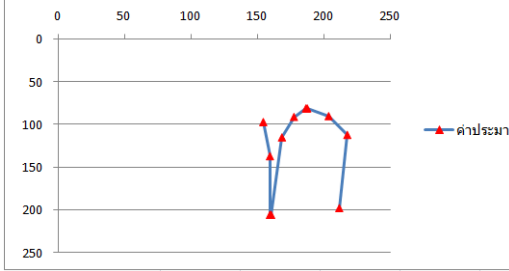
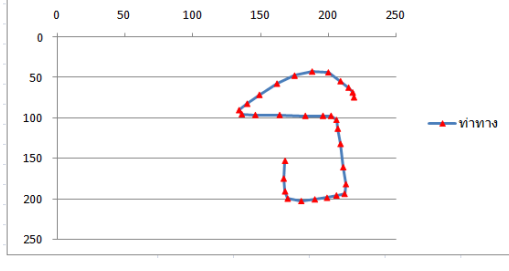
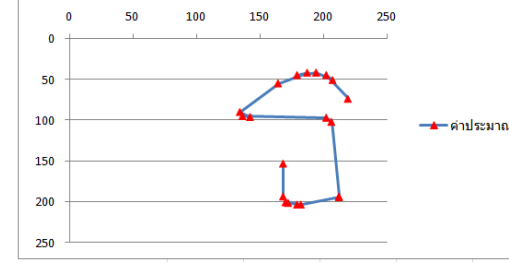
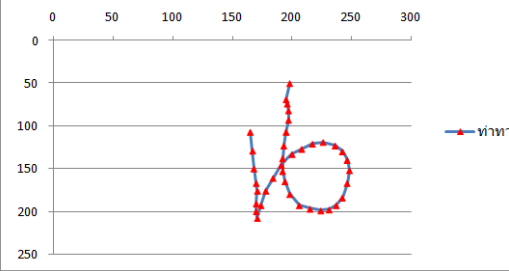
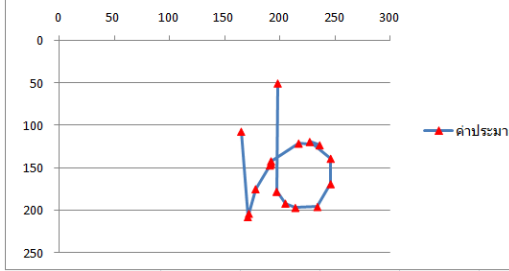
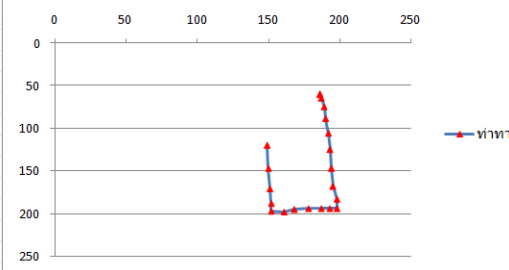
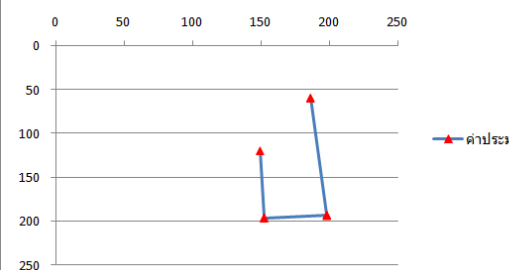
(a) เส้นตรง (b) เส้นโค้ง (c) เส้นตัววี

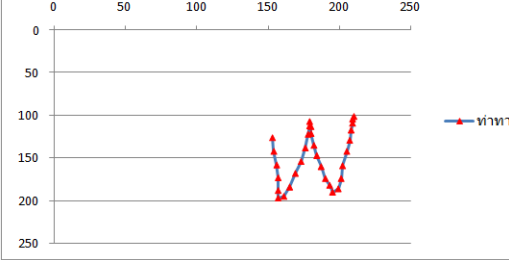
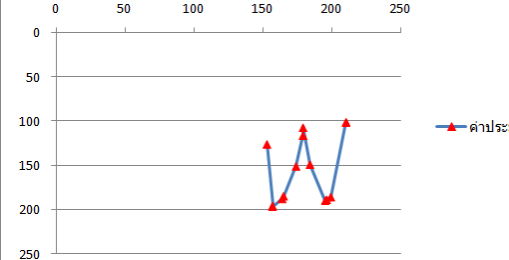
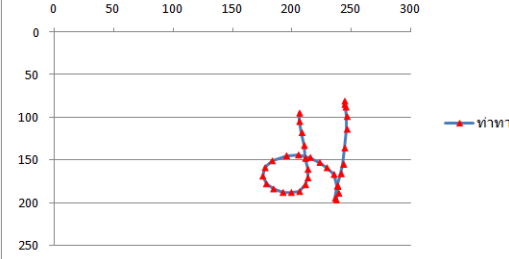
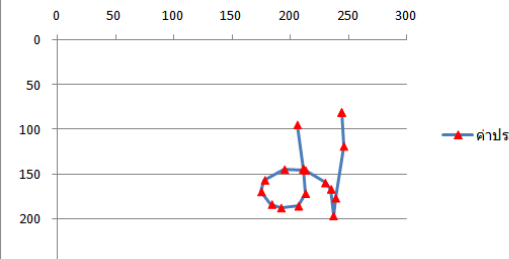
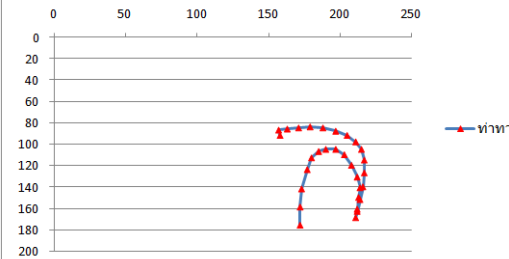
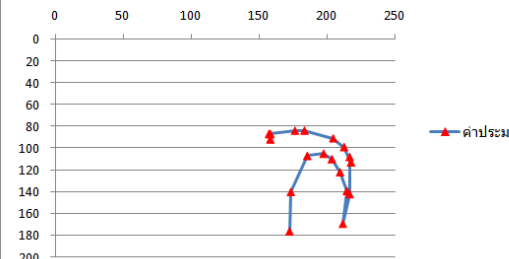
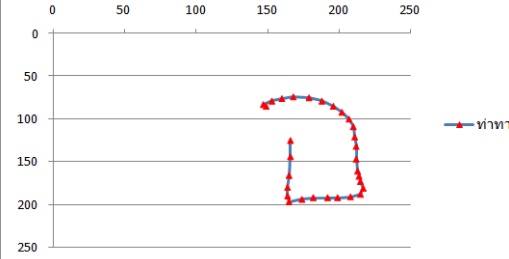
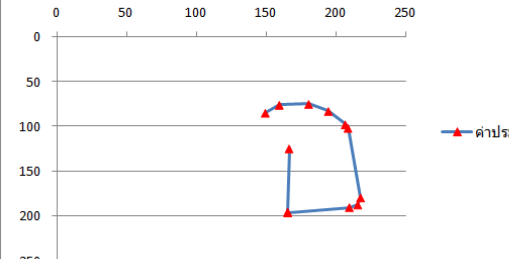
4.3. ผลการทดลองกระบวนการลดจำนวนข้อมูลท่าทางมือ

เนื่องจากการเคลื่อนที่ของมือไม่ได้มีการเปลี่ยนแปลงไปอย่างรวดเร็วทำให้มีตำแหน่งของจุดข้อมูลที่อยู่ติดกันเป็นจำนวนมาก ซึ่งไม่จำเป็นที่จะต้องนำจุดข้อมูลเหล่านั้นมาพิจารณาทุกจุดข้อมูล ดังนั้นจึงจำเป็นที่จะต้องทำการลดจำนวนจุดข้อมูลของท่าทางมือลงเพื่อประมาณค่าของท่าทางมือโดยใช้จำนวนจุดข้อมูลที่น้อยที่สุดแต่ยังคงให้ลักษณะของท่าทางเหมือนเดิม ซึ่งทำให้ช่วยลดจำนวนหน่วยความจำที่ใช้ในการเก็บค่าจุดข้อมูลและยังช่วยเพิ่มความรวดเร็วในการจับคู่ท่าทางมือกับแบบจำลองท่าทางในกระบวนการรู้จำ ในตาราง 4-3 แสดงตัวอย่างท่าทางมือ และค่าประมาณท่าทางมือหลังจากทำการลดจำนวนจุดข้อมูลของท่าทางทั้ง 12 แบบที่ได้จากตัวอักษรภาษาไทย

ตาราง 4-3 แสดงจำนวนจุดข้อมูลของท่าทางมือ และค่าประมาณท่าทางมือ

ท่าทางมือ	ค่าประมาณท่าทางมือ
<p style="text-align: center;">ท่าทาง</p>  <p style="text-align: center;">ก 30 จุด</p>	<p style="text-align: center;">ค่าประมาณ</p>  <p style="text-align: center;">ก 10 จุด</p>
<p style="text-align: center;">ท่าทาง</p>  <p style="text-align: center;">ข 22 จุด</p>	<p style="text-align: center;">ค่าประมาณ</p>  <p style="text-align: center;">ข 12 จุด</p>
<p style="text-align: center;">ท่าทาง</p>  <p style="text-align: center;">ง 16 จุด</p>	<p style="text-align: center;">ค่าประมาณ</p>  <p style="text-align: center;">ง 6 จุด</p>
<p style="text-align: center;">ท่าทาง</p>  <p style="text-align: center;">ค 39 จุด</p>	<p style="text-align: center;">ค่าประมาณ</p>  <p style="text-align: center;">ค 22 จุด</p>

ท่าทางมือ	ค่าประมาณท่าทางมือ
<p style="text-align: center;">ท่าทาง</p>  <p style="text-align: center;">ท 34 จุด</p>	<p style="text-align: center;">ค่าประมาณ</p>  <p style="text-align: center;">ท 11 จุด</p>
<p style="text-align: center;">ท่าทาง</p>  <p style="text-align: center;">ธ 31 จุด</p>	<p style="text-align: center;">ค่าประมาณ</p>  <p style="text-align: center;">ธ 20 จุด</p>
<p style="text-align: center;">ท่าทาง</p>  <p style="text-align: center;">น 38 จุด</p>	<p style="text-align: center;">ค่าประมาณ</p>  <p style="text-align: center;">น 17 จุด</p>
<p style="text-align: center;">ท่าทาง</p>  <p style="text-align: center;">ป 21 จุด</p>	<p style="text-align: center;">ค่าประมาณ</p>  <p style="text-align: center;">ป 7 จุด</p>

ท่าทางมือ	ค่าประมาณท่าทางมือ
<p style="text-align: center;">ท่าทาง</p>  <p style="text-align: center;">พ 31 จุด</p>	<p style="text-align: center;">ค่าประมาณ</p>  <p style="text-align: center;">พ 14 จุด</p>
<p style="text-align: center;">ท่าทาง</p>  <p style="text-align: center;">ม 35 จุด</p>	<p style="text-align: center;">ค่าประมาณ</p>  <p style="text-align: center;">ม 18 จุด</p>
<p style="text-align: center;">ท่าทาง</p>  <p style="text-align: center;">ถ 32 จุด</p>	<p style="text-align: center;">ค่าประมาณ</p>  <p style="text-align: center;">ถ 18 จุด</p>
<p style="text-align: center;">ท่าทาง</p>  <p style="text-align: center;">อ 30 จุด</p>	<p style="text-align: center;">ค่าประมาณ</p>  <p style="text-align: center;">อ 12 จุด</p>

4.4. ผลการทดลองกระบวนการรู้จำท่าทางมือ (Hand Gesture Recognition)

4.4.1. ผลการทดลองการรู้จำท่าทางมืออย่างง่าย

การทดลองการรู้จำท่าทางมืออย่างง่ายที่นำไปประยุกต์ใช้ในการควบคุมโปรแกรมวินโดว มีเดียเซ็นเซอร์ในระบบปฏิบัติการวินโดว 7 จะทดลองที่สภาพแวดล้อมภายในอาคาร ไม่มีการเปลี่ยนแปลงของแสงไฟ และทำการทดลองด้วยระยะห่างระหว่างมือกับกล้องกับกับความเร็วในการเคลื่อนที่ของมือที่แตกต่างกัน

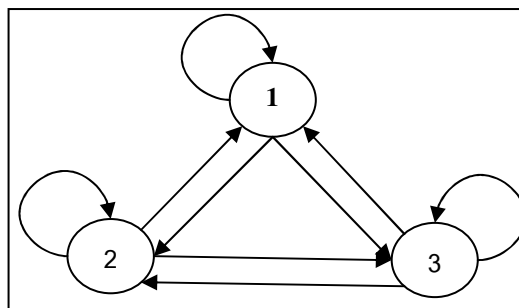
ตารางที่ 4-4 ผลการทดลองการรู้จำท่าทางมืออย่างง่าย

ระยะทาง (ซ.ม.)	ความเร็ว (วินาที)		
	0.5	1	2
30	10/10	10/10	10/10
60	10/10	10/10	10/10
90	7/1	8/10	9/10

ตาราง 4-4 แสดงผลการทดลองการรู้จำท่าทางมืออย่างง่าย โดยระยะห่างระหว่างมือกับกล้องแบ่งเป็น 3 ระดับ คือ ระยะห่าง 30 ซ.ม. 60 ซ.ม. และ 90 ซ.ม. ส่วนความเร็วในการเคลื่อนที่ของมือแบ่งเป็น 3 ระดับเช่นกัน คือ ความเร็ว 0.5 วินาที 1 วินาที และ 2 วินาที การทดลองจะทดลองเป็นจำนวน 10 ครั้งในแต่ละกรณี จากผลการทดลองสังเกตได้ว่าที่ความเร็วต่ำท่าทางมือจะถูกรู้จำได้ดีกว่าที่ความเร็วสูง เนื่องจากที่ความเร็วต่ำในกระบวนการการรู้จำจะใช้เฟรมภาพจำนวนมากกว่าในการวิเคราะห์ทำให้เพิ่มประสิทธิภาพในการรู้จำท่าทางมือได้ดีขึ้น และยังสังเกตได้อีกว่าระยะห่างระหว่างมือกับกล้องที่ใกล้ให้ผลในการรู้จำท่าทางมือได้ดีกว่าระยะห่างที่ไกล เนื่องจากพื้นที่ของมือเมื่ออยู่ไกลจากกล้องมากก็จะทำให้กระบวนการตรวจจับลักษณะมือตรวจจับไม่ได้

4.4.2. ผลการทดลองการรู้จำท่าทางมือแบบสัญลักษณ์

การทดสอบการรู้จำท่าทางมือทั้ง 12 ท่าทางจะทำการทดสอบเป็นจำนวน 100 ชุดข้อมูลต่อหนึ่งท่าทาง โดยชุดข้อมูลภาพวิดีโอถูกถ่ายทำในสภาพแวดล้อมภายในอาคาร ซึ่งใช้กล้องจำนวน 1 ตัว ความละเอียดของภาพขนาด 320x240 จุดภาพ จำนวนสถานะของแบบจำลองฮิดเดนมาร์คอฟที่ใช้ในการทดสอบเป็น 10 สถานะ แบ่งเป็น 8 สถานะซ่อนและอีก 2 สถานะสำหรับจุดเริ่มต้นและจุดสิ้นสุด ชนิดของแบบจำลองฮิดเดนมาร์คอฟเป็นแบบการเชื่อมโยงทั่วถึง (Fully Connected Model) ซึ่งทุกๆ สถานะจะถูกเชื่อมโยงด้วยความน่าจะเป็นถึงกันหมด

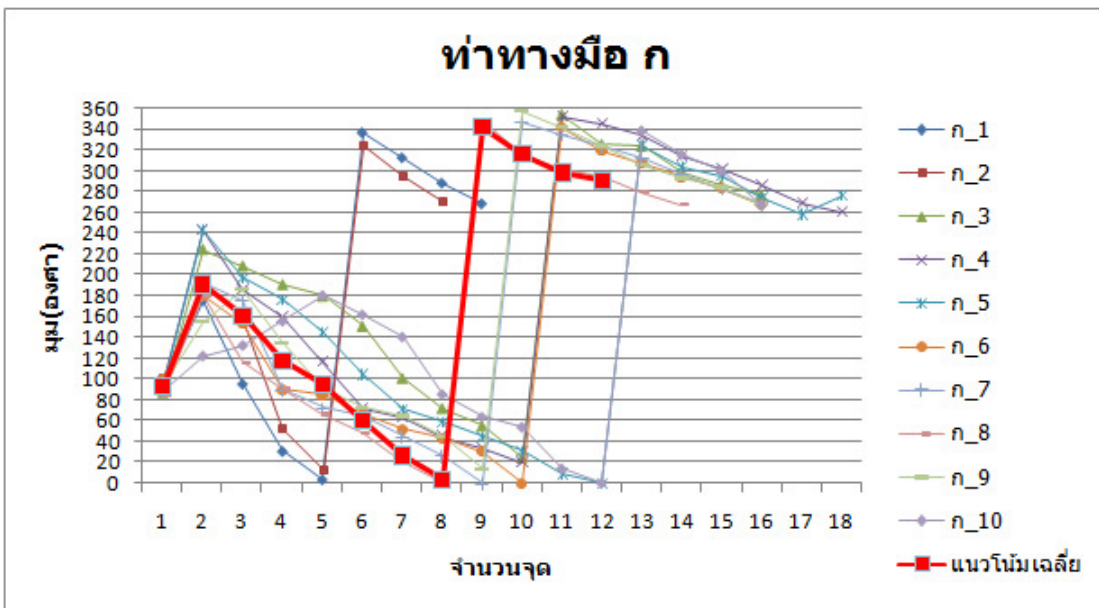


ภาพประกอบที่ 4-5 แบบจำลองฮิดเดนมาร์คอฟแบบการเชื่อมโยงทั่วถึง

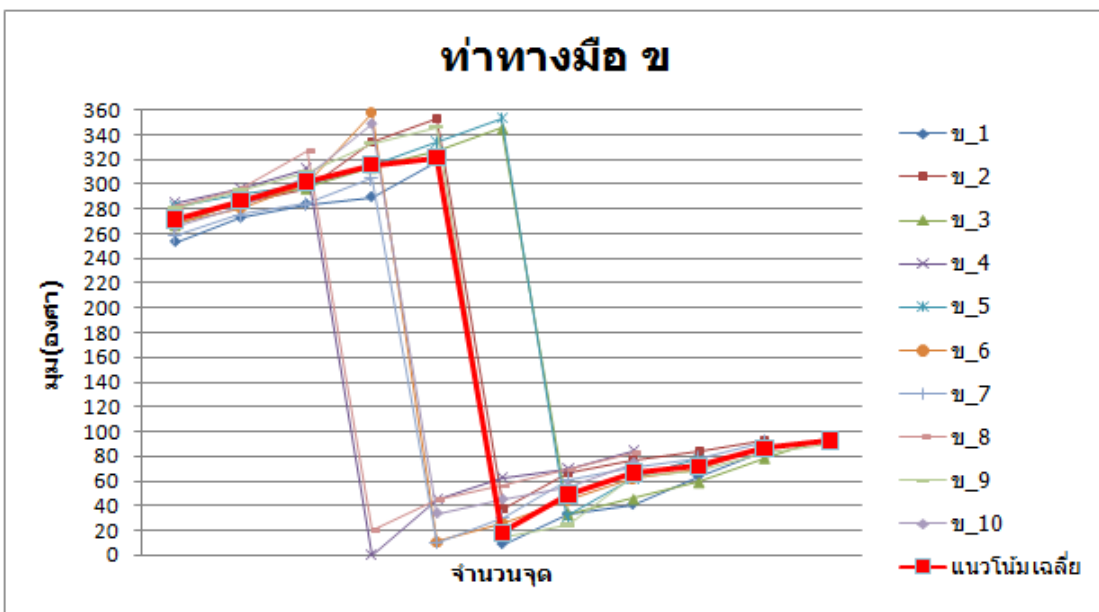
ตาราง 4-5 ประสิทธิภาพการรู้จำท่าทางมือด้วยแบบจำลองฮิดเดนมาร์คอฟ 10 สถานะ

ท่าทาง (100 ชุดข้อมูล/ท่าทาง)	การฝึกสอน 10 สถานะ	
	ถูกต้อง (ชุดข้อมูล)	ผิดพลาด (ชุดข้อมูล)
ก	82	ด = 16, น = 1, ม = 1
ข	90	ท = 9, อ = 1
ง	95	ด = 2, อ = 3
ค	98	ก = 1, น = 1
ท	65	ก = 3, ข = 6, ค = 6, น = 4, พ = 8, ม = 2, ล = 3, อ = 3
ธ	69	ข = 1, ป = 2, อ = 28
น	93	ค = 1, ท = 3, พ = 1, อ = 2
ป	75	ข = 9, ธ = 1, อ = 15
พ	70	ข = 11, ท = 14, น = 1, ป = 3, ล = 1
ม	69	ก = 12, ข = 2, ท = 11, น = 5, อ = 1
ล	95	น = 2, ม = 1, อ = 2
อ	97	ธ = 2, ป = 1

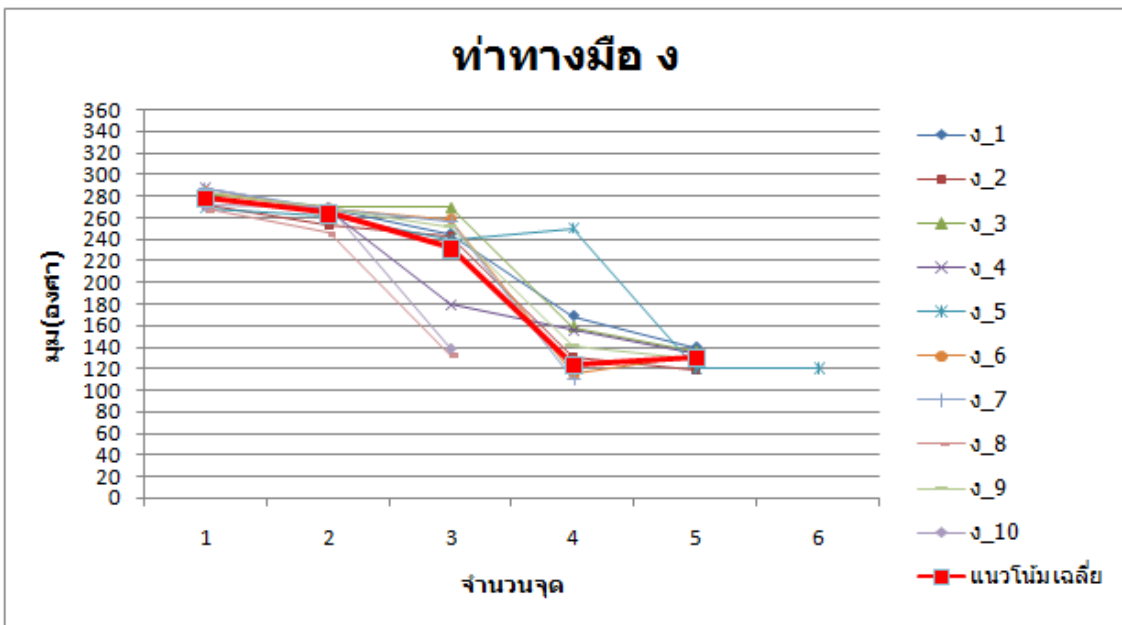
ตาราง 4-5 แสดงถึงความความถูกต้องในการรู้จำท่าทางมือแต่ละแบบ โดยท่าทางที่นำมาใช้รู้จำเลือกมาจากการแบ่งกลุ่มตัวอักษรภาษาไทยที่มีความคล้ายคลึงกัน แต่ในการรู้จำท่าทางก็ยังมีผลผิดพลาดที่ท่าทางแบบหนึ่งอาจถูกรู้จำเป็นท่าทางอีกแบบหนึ่งเกิดขึ้นได้ เช่น ในกรณีท่าทางมือ ก อาจถูกรู้จำเป็นท่าทางมือ ค ท่าทางมือ น หรือท่าทางมือ ม ถึงแม้ว่าท่าทางมือทั้ง 4 แบบจะดูแตกต่างกันอย่างชัดเจนเมื่อนำจุดข้อมูลมาแสดงภาพ (ตาราง 4-3) แต่สิ่งที่ถูกนำมาใช้พิจารณาในการรู้จำที่แท้จริงก็คือ มุมที่เกิดขึ้นระหว่างจุด 2 จุดที่ติดกันภายในลำดับชุดข้อมูล



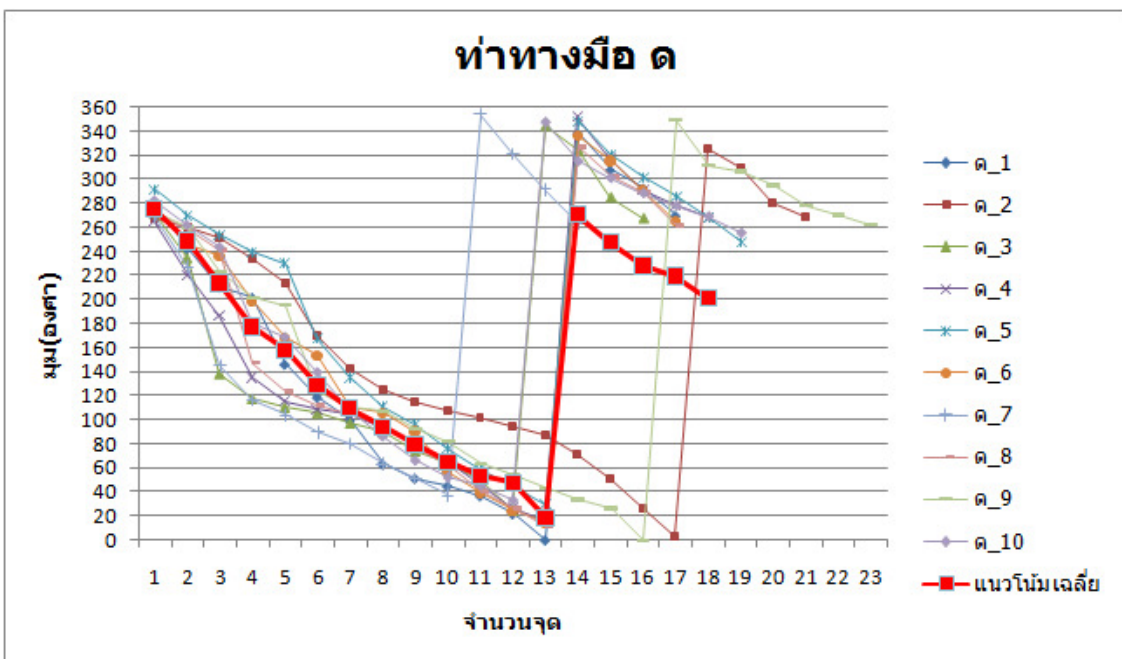
ภาพประกอบ 4-6 มุมท่าทางมือ ก



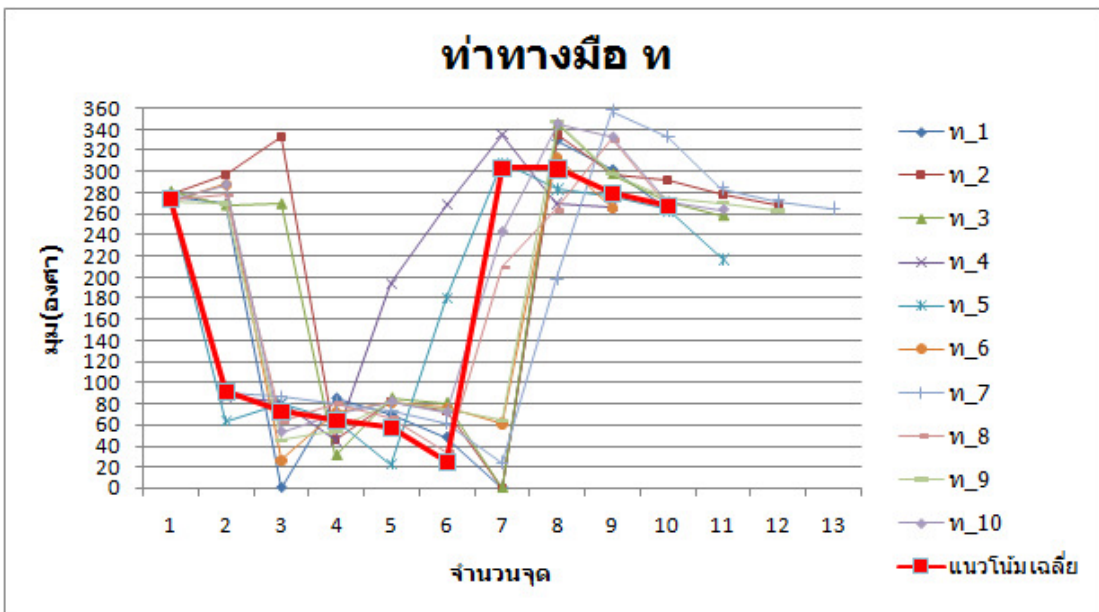
ภาพประกอบ 4-7 มุมท่าทางมือ ข



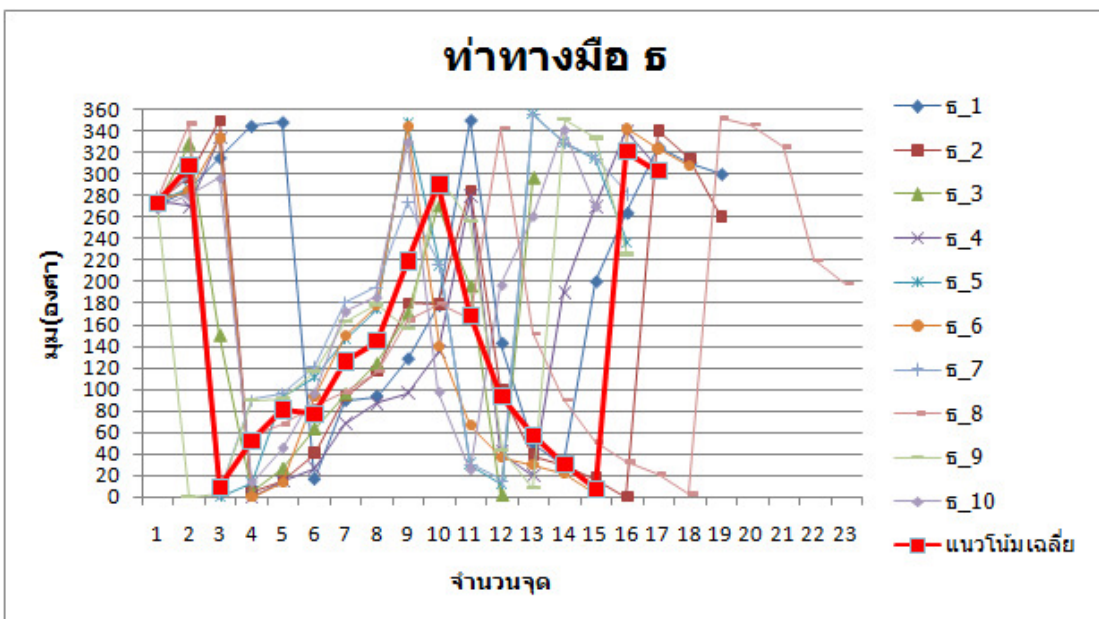
ภาพประกอบ 4-8 มุมท่าทางมือ ง



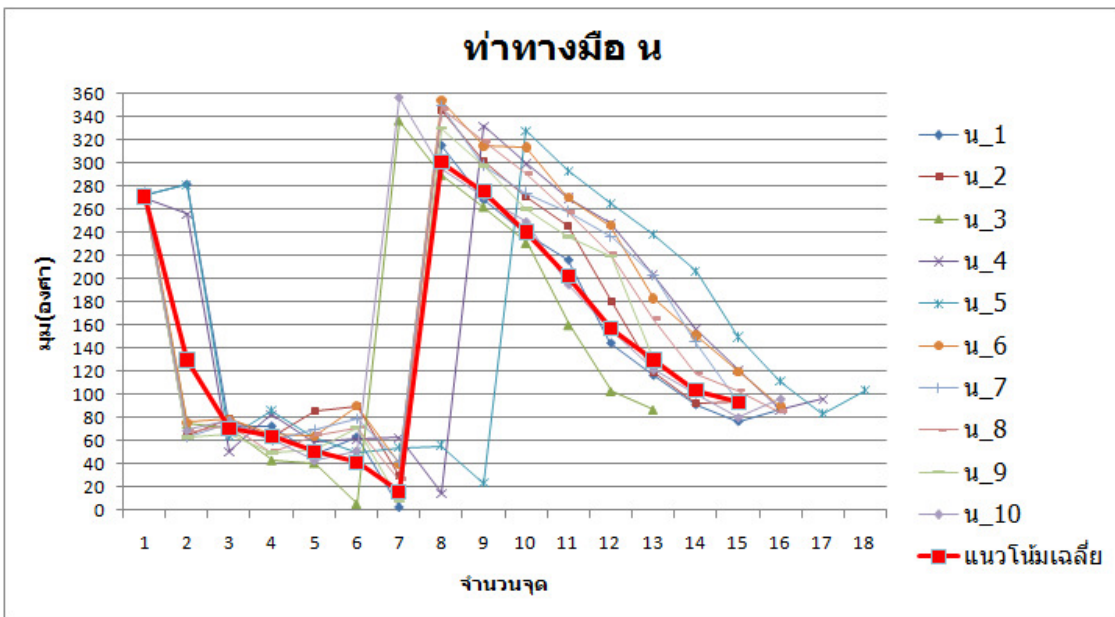
ภาพประกอบ 4-9 มุมท่าทางมือ ด



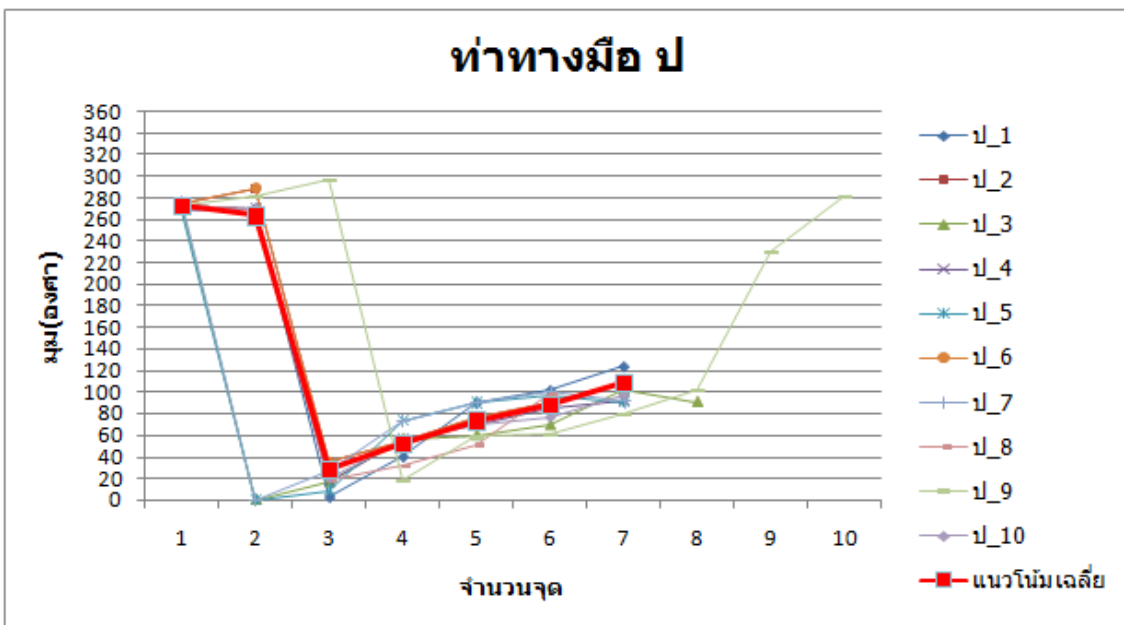
ภาพประกอบ 4-10 มุมท่าทางมือ ท



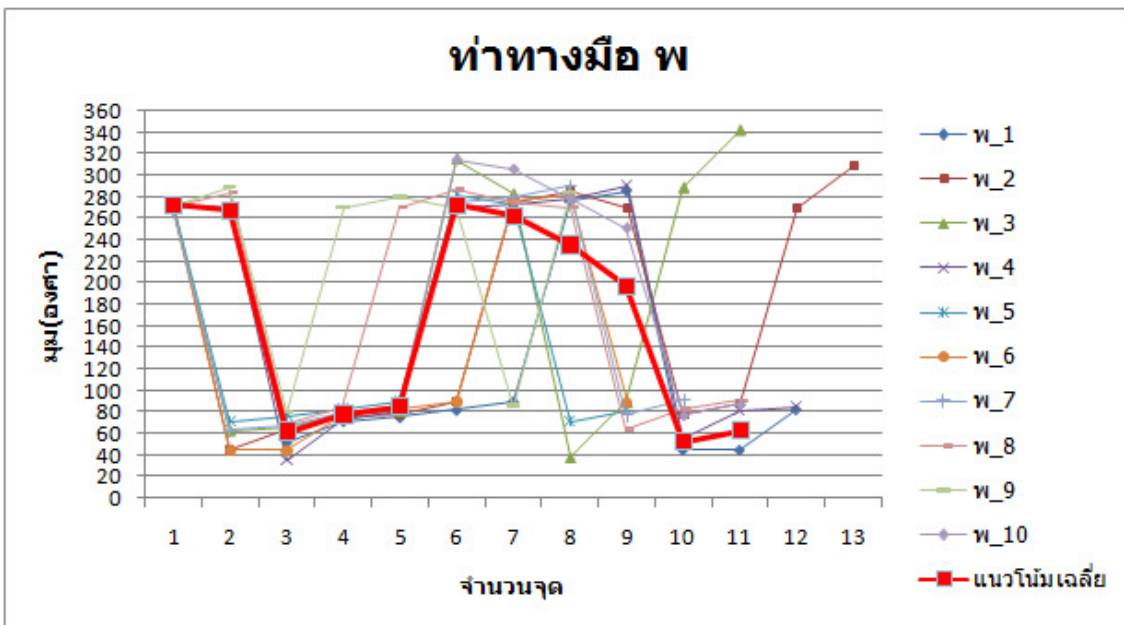
ภาพประกอบ 4-11 มุมท่าทางมือ ธ



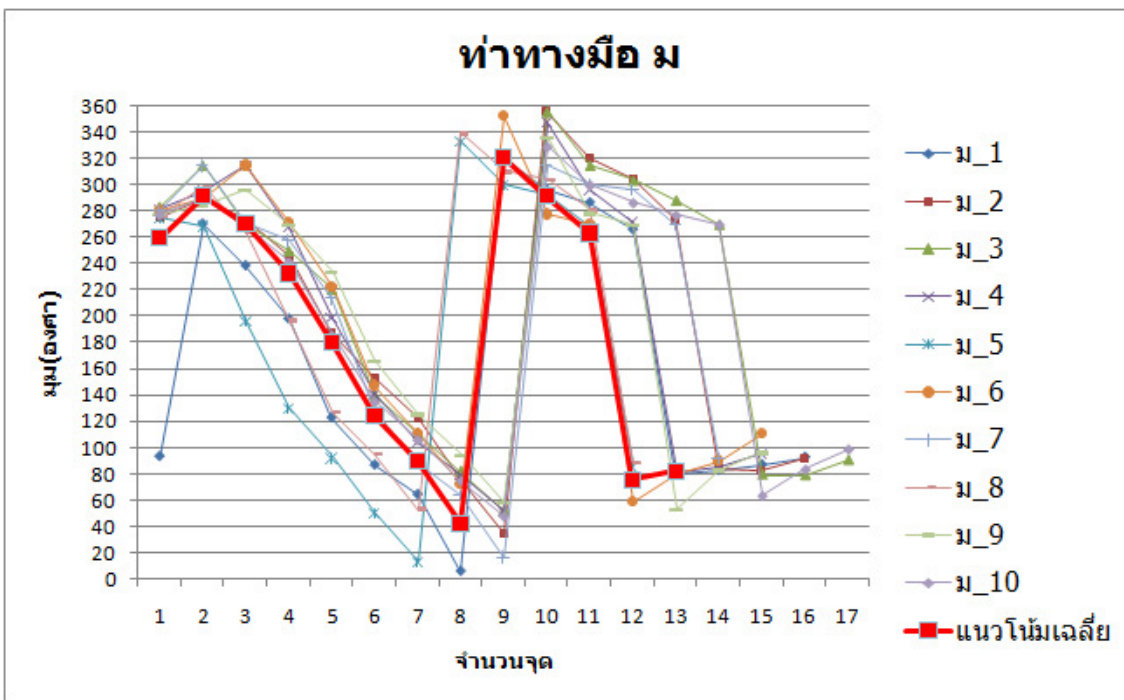
ภาพประกอบ 4-12 มุมท่าทางมือ น



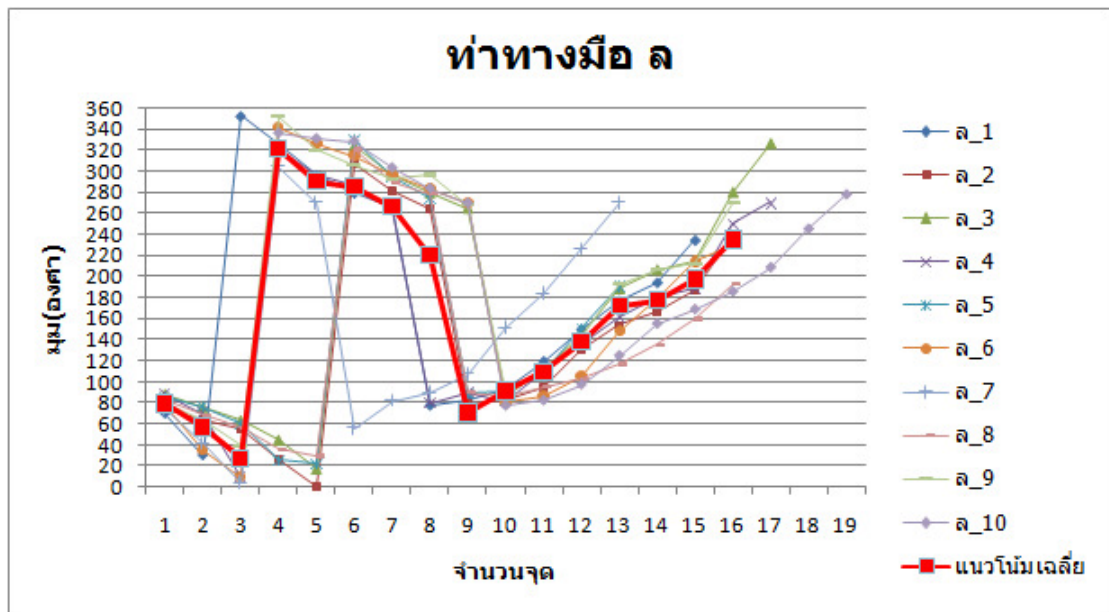
ภาพประกอบ 4-13 มุมท่าทางมือ ป



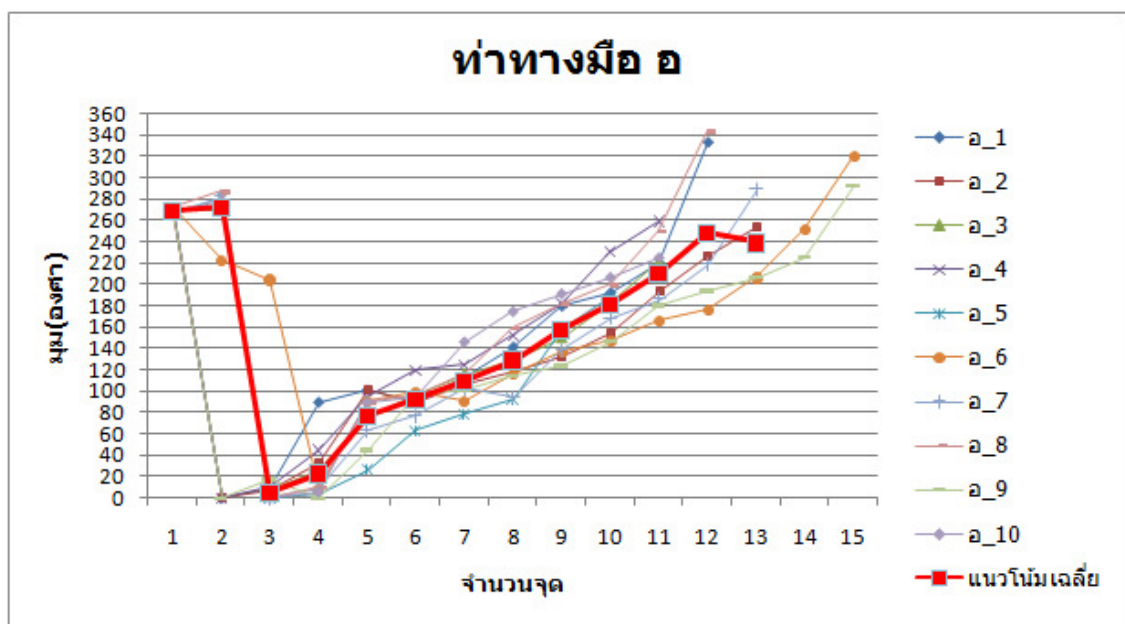
ภาพประกอบ 4-14 มุมท่าทางมือ พ



ภาพประกอบ 4-15 มุมท่าทางมือ ม

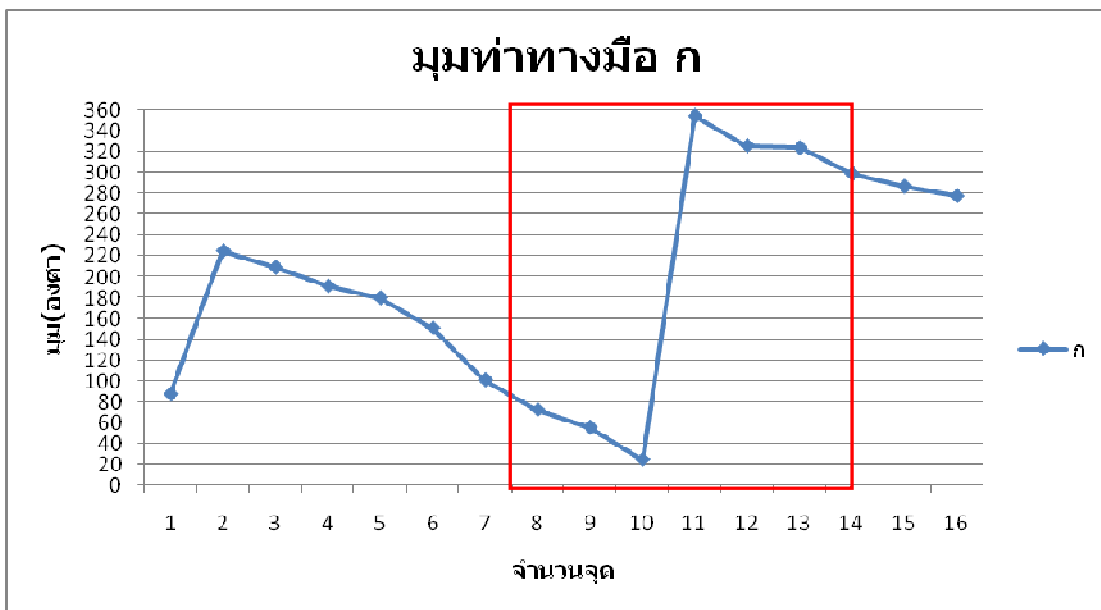


ภาพประกอบ 4-16 มุมท่าทางมือ ล

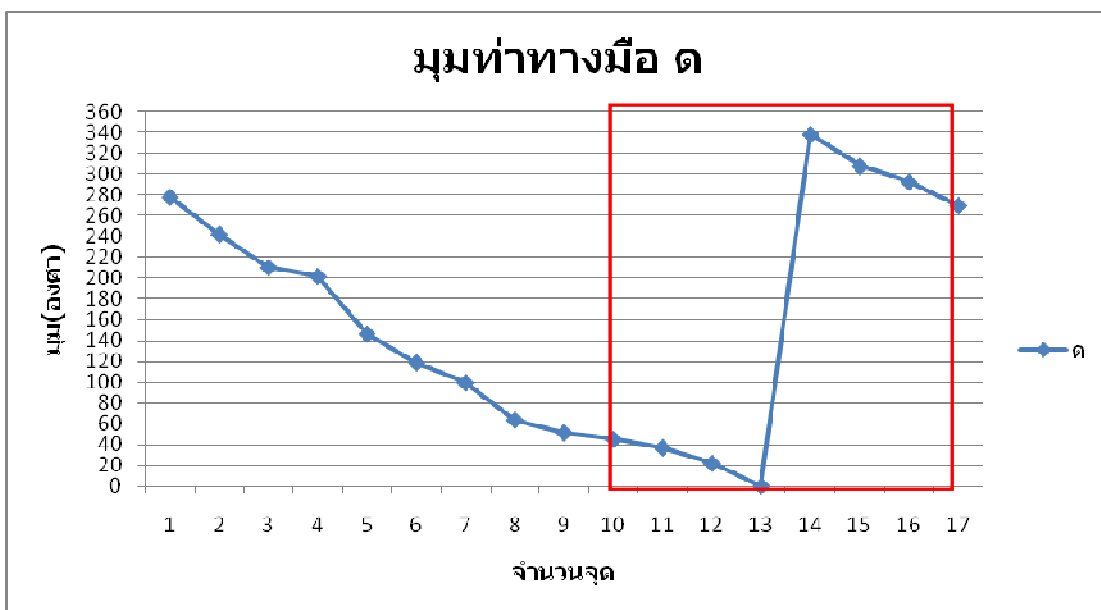


ภาพประกอบ 4-17 มุมท่าทางมือ อ

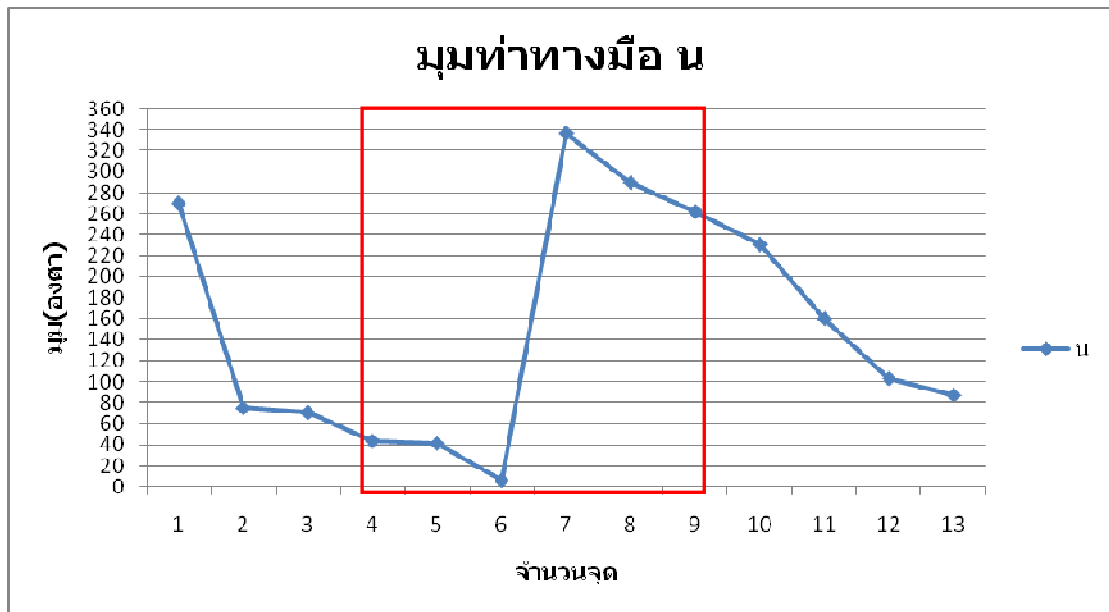
ภาพประกอบ 4-6 ถึง 4-17 แสดงให้เห็นถึงลำดับของมุมที่เกิดขึ้นในท่าทางมือแบบต่างๆ ที่จะถูกแปลงเป็นรหัสข้อมูลเพื่อใช้ในการฝึกสอนด้วยแบบจำลองฮิดเดนมาร์คอฟ ซึ่งจากในกรณีความผิดพลาดที่กล่าวไปข้างต้น เมื่อพิจารณาถึงมุมที่เกิดขึ้นของท่าทางมือทั้ง 4 แบบ จะเห็นว่ามีส่วนของการเปลี่ยนแปลงมุมที่มีลักษณะคล้ายกันอยู่ (ภาพประกอบ 4-18)



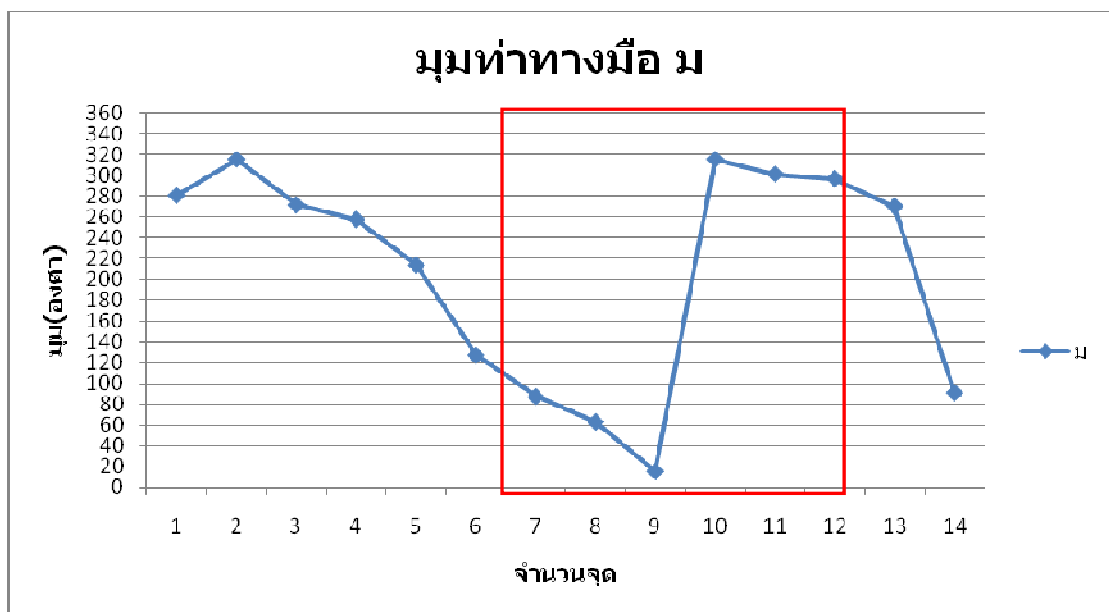
(a)



(b)



(c)



(d)

ภาพประกอบ 4-18 ส่วนการเปลี่ยนแปลงมุ่มที่คล้ายกันของท่าทางมือ (a) ท่าทางมือ ก (b) ท่าทางมือ ค (c) ท่าทางมือ น (d) ท่าทางมือ ม

ภาพประกอบ 4-18 แสดงส่วนของการเปลี่ยนแปลงมุ่มที่คล้ายกันของท่าทางมือทั้ง 4 แบบ (ภายในกรอบสี่เหลี่ยม) อาจจะทำให้เกิดความผิดพลาดในการรู้จำท่าทางขึ้นได้เนื่องจากแบบจำลองฮิดเดนมาร์คอฟจะพยายามเรียนรู้รูปแบบของลำดับข้อมูลว่ามีความเหมือนหรือคล้ายกันมากน้อยแค่ไหน



ภาพประกอบ 4-21 ท่าทางมือ ด ที่ระบบรู้จำเป็นท่าทางมือ ด



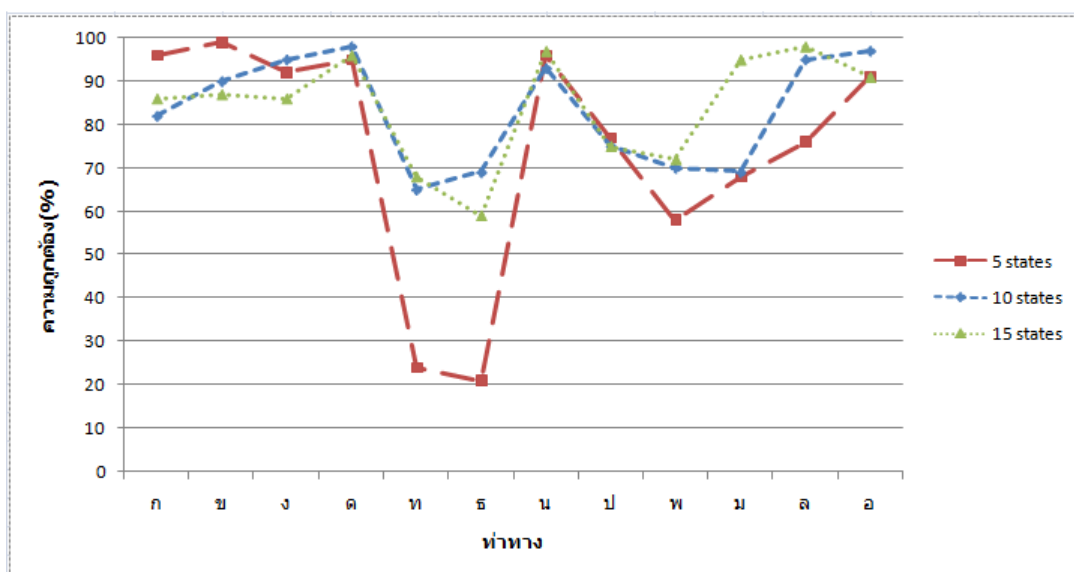
ภาพประกอบ 4-22 มุมท่าทางมือ ด ที่ระบบรู้จำเป็นท่าทางมือ ด

เปรียบเทียบภาพประกอบ 4-19 กับ 4-21 จะเห็นได้ว่าข้อมูลจุดของท่าทางทั้ง 2 แบบดูแตกต่างกันอย่างเห็นได้ชัด แต่เมื่อนำภาพประกอบ 4-20 มาเปรียบเทียบกับภาพประกอบที่ 4-22 จะเห็นได้ว่ามีส่วนของการเปลี่ยนแปลงมุมที่คล้ายกันอยู่(ภายในกรอบสี่เหลี่ยม)และเมื่อนำมุมท่าทางมือ ก ที่ถูกรู้จำเป็นท่าทางมือ ค มาแปลงเป็นชุดรหัสข้อมูลจะได้เป็น $V_n = \{4, 0, 10, 9, 8, 8, 7, 6, 5, 3, 3, 2, 0, 17, 16, 15, 15, 14, 13\}$ ส่วนมุมท่าทางมือ ค ที่ระบบรู้จำเป็นท่าทางมือ ค จะได้เป็น $V_n = \{14, 14, 13, 12, 10, 8, 6, 6, 5, 5, 4, 4, 3, 2, 1, 0, 17, 16, 15, 15, 14, 13, 12\}$ ซึ่งเมื่อนำชุดรหัสข้อมูล V_n ไปหาความน่าจะเป็นกับแบบจำลองท่าทางมือ ก ที่ได้ฝึกสอนไว้อาจเป็นไปได้ว่าลำดับของชุดรหัสข้อมูลในส่วนแรก (ไม่ขีดเส้นใต้) ให้ความน่าจะเป็นออกมามีค่าน้อยและลำดับของชุดรหัสข้อมูลในส่วนที่สอง (ขีดเส้นใต้) ก็มีความคล้ายคลึงกับลำดับชุดรหัสข้อมูล V_n ดังนั้นชุดรหัสข้อมูล V_n จึงอาจให้ความน่าจะเป็นที่จะเป็นท่าทางมือ ค มากกว่าท่าทางมือ ก จึงทำให้ระบบเกิดความผิดพลาดในการรู้จำท่าทางมือ ซึ่งจากสาเหตุดังกล่าวก็สามารถเกิดขึ้นได้กับท่าทางมือแบบอื่นๆ ได้เช่นกันดังที่ได้แสดงไว้ในตาราง 4-5

เนื่องจากแบบจำลองฮิดเดนมาร์คคอฟไม่สามารถที่จะบอกได้ว่าจำนวนสถานะควรจะเป็นเท่าไรถึงจะดีต่อระบบ ดังนั้นจึงได้มีการทดสอบการรู้จำท่าทางมือกับแบบจำลองฮิดเดนมาร์คคอฟที่ใช้จำนวนสถานะเป็น 5 สถานะกับ 15 สถานะเพื่อเปรียบเทียบผลการทดลองกับแบบจำลองฮิดเดนมาร์คคอฟที่ใช้จำนวนสถานะเป็น 10 สถานะตามที่ได้ออกแบบไว้ จากตาราง 4-6 เมื่อนำผลการทดสอบแบบจำลองฮิดเดนมาร์คคอฟ 5 สถานะเทียบกับผลการทดสอบแบบจำลองฮิดเดนมาร์คคอฟ 10 สถานะความถูกต้องของการรู้จำท่าทางมือจะเพิ่มขึ้นประมาณ 9 % แต่เมื่อนำผลการทดสอบแบบจำลองฮิดเดนมาร์คคอฟ 10 สถานะเทียบกับแบบจำลองฮิดเดนมาร์คคอฟ 15 สถานะความถูกต้องของการรู้จำท่าทางมือเพิ่มขึ้นเพียง 1% เท่านั้น ดังนั้นจึงไม่มีความจำเป็นที่จะต้องใช้จำนวนสถานะที่มากกว่านี้ทำให้แบบจำลองฮิดเดนมาร์คคอฟที่ใช้สถานะ 10 สถานะเพียงพอสำหรับระบบของการรู้จำท่าทางมือ

ตาราง 4-6 ประสิทธิภาพการรู้จำท่าทางมือด้วยแบบจำลองฮิดเดนมาร์คอฟที่สถานะต่างๆ

ท่าทาง	ความถูกต้องของการรู้จำ (%)		
	5 สถานะ	10 สถานะ	15 สถานะ
ก	96	82	86
ข	99	90	87
ง	92	95	86
ด	95	98	96
ท	24	65	68
ธ	21	69	59
น	96	93	97
ป	77	75	75
พ	58	70	72
ม	68	69	95
ล	76	95	98
อ	91	97	91
μ	74.41	83.16	84.16
σ	27.40	12.81	12.79



ภาพประกอบ 4-23 ประสิทธิภาพการรู้จำท่าทางมือด้วยแบบจำลองฮิดเดนมาร์คอฟ

4.5. สรุป

เนื้อหาในบทนี้ได้กล่าวถึงผลการทดสอบกระบวนการต่าง ๆ ที่ถูกอธิบายไว้ในเนื้อหาบทที่ 3 โดยจากผลการทดสอบจะแบ่งแยกออกเป็นการทดสอบกระบวนการตรวจจับลักษณะมือในการฝึกสอนแต่ละสถานะพบว่ามีความถูกต้องโดยเฉลี่ยประมาณ 91 เปอร์เซ็นต์ การทดสอบกระบวนการติดตามการเคลื่อนที่ของมือที่ได้ทดลองเคลื่อนที่มือใน 3 รูปแบบ คือ เส้นตรง เส้นโค้ง และเส้นตัววี เนื่องจากเส้นทั้ง 3 แบบเป็นส่วนประกอบหลักของท่าทางมือที่จะใช้ในการรู้จำ และการทดสอบกระบวนการรู้จำท่าทางมือ ซึ่งแบ่งได้ 2 ลักษณะคือ การรู้จำท่าทางมืออย่างง่ายที่นำไปประยุกต์ใช้ควบคุมโปรแกรมวินโดวมีเดียเซ้นเทอในระบบปฏิบัติการวินโดว 7 และการรู้จำท่าทางมือที่เป็นสัญลักษณ์ โดยสัญลักษณ์ที่นำมาใช้ในการรู้จำจะเป็นตัวอักษรภาษาไทยที่ถูกแบ่งกลุ่มตามลักษณะความคล้ายคลึงกัน ซึ่งได้ทดสอบการรู้จำด้วยการฝึกสอนด้วยแบบจำลองฮิดเดนมาร์คอฟที่ได้แบ่งการฝึกสอนออกเป็นสถานะที่แตกต่างกันเพื่อเปรียบเทียบผลการรู้จำพบว่ามีความถูกต้องโดยเฉลี่ย 83 เปอร์เซ็นต์

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1. สรุปผล

ภายในงานวิจัยนี้ได้พัฒนากระบวนการรู้จำท่าทางมือ โดยท่าทางมือในที่นี้สามารถอธิบายได้ด้วยการเคลื่อนที่ของมือเป็นสัญลักษณ์ต่างๆ เช่น วงกลม สามเหลี่ยม หรือสี่เหลี่ยม เพื่อนำท่าทางมือเหล่านั้นมาใช้เป็นชุดคำสั่งที่สามารถนำไปประยุกต์ใช้กับงานในประเภทต่างๆ ได้อย่างเหมาะสม ซึ่งผลจากการวิจัยสามารถที่จะนำเอาระบบมาใช้สำหรับกระบวนการรู้จำท่าทางมือที่มีความซับซ้อน นอกเหนือการรู้จำท่าทางมือพื้นฐานดังที่ได้กล่าวไปข้างต้นเช่น รู้จำท่าทางจากตัวอักษรภาษาไทย เป็นต้น โดยกระบวนการรู้จำท่าทางมือสามารถแบ่งได้เป็น 3 ขั้นตอนหลัก คือ (1) ขั้นตอนการตรวจจับลักษณะมือเป็นการฝึกสอนให้คอมพิวเตอร์รู้จักถึงวัตถุที่สนใจเพื่อทำการตรวจจับและแยกแยะออกจากภาพพื้นหลัง ซึ่งวัตถุที่สนใจในที่นี้ก็คือ ลักษณะของมือที่ตั้งตรงขนานกับตัวกล้อง (2) ขั้นตอนการติดตามการเคลื่อนที่ของมือ เนื่องจากท่าทางมืออธิบายได้ด้วยการเคลื่อนที่ของมือ ดังนั้นจึงต้องมีการติดตามการเคลื่อนที่ของมือ ซึ่งผลจากการติดตามทำให้สามารถบอกได้ถึงข้อมูลทิศทาง การเคลื่อนที่ ขนาด และมุม ซึ่งจากข้อมูลทั้งหมดดังกล่าวข้างต้นจะถูกนำไปใช้ในขั้นตอนสุดท้ายคือ (3) ขั้นตอนการรู้จำท่าทางมือ เป็นการสร้างแบบจำลองที่ใช้ในการอธิบายถึงท่าทางมือในรูปแบบต่างๆ เพื่อให้คอมพิวเตอร์สามารถจดจำและจำแนกได้ว่าท่าทางมือที่ผู้ใช้ได้แสดงออกมาเป็นท่าทางในลักษณะใด ซึ่งภายในงานวิจัยนี้ได้ทำการทดลองขั้นตอนทั้ง 3 ขั้นตอน โดยสรุปผลการทดลองมีดังนี้

5.1.1. สรุปผลการทดลองกระบวนการตรวจจับลักษณะมือ

การฝึกสอนได้ใช้ภาพกลุ่มตัวอย่างข้อมูลที่ใช่และไม่ใช่เป็นจำนวน 5,058 และ 1,048 ภาพ โดยได้ทำการฝึกสอนเป็นจำนวน 4 ครั้งแต่ละครั้งมีสถานะในการฝึกสอนที่แตกต่างกัน จากการฝึกสอนที่สถานะต่ำทำให้อัตราการเจอสูงแต่ความผิดพลาดเชิงบวกก็สูงเช่นกัน แต่เมื่อทำการฝึกสอนที่สถานะสูงขึ้นอัตราการเจอจะลดลงและความผิดพลาดเชิงบวกก็ลดลงเช่นกัน เนื่องจากการฝึกสอนที่สถานะสูงจะทำให้ฟังก์ชันการแยกประเภทมีลักษณะเฉพาะมากขึ้นทำให้วัตถุที่นำมาตรวจจับต้องมีลักษณะที่ใกล้เคียงกับกลุ่มตัวอย่างที่นำมาฝึกสอนมากขึ้นจึงจะสามารถตรวจจับได้ ผลจากการฝึกสอนในการตรวจจับลักษณะมีความถูกต้องโดยเฉลี่ยอยู่ที่ 91 เปอร์เซ็นต์ โดยลักษณะ

ที่ตรวจจับได้จะเป็นลักษณะของมือที่ตั้งตรงขนานกับตัวกล้องสามารถตรวจจับได้ทั้งมือซ้ายและมือขวาแต่จะเกิดความผิดพลาดในการตรวจจับเมื่อมีการเอียงมือเกิดขึ้นหรือระยะห่างระหว่างมือกับตัวกล้องมากเกินไป

5.1.2. สรุปผลการทดลองทดลองกระบวนการติดตามการเคลื่อนที่ของมือ

การติดตามการเคลื่อนที่ของมือจะถูกนำมาใช้เนื่องจากตัวตรวจจับลักษณะมือไม่สามารถที่จะตรวจจับได้เมื่อมือเกิดการเอียงหรืออยู่ห่างจากกล้องไกลเกินไป ซึ่งการติดตามจะใช้คุณสมบัติของสีของบริเวณมือที่ถูกตรวจจับได้ การทดสอบเบื้องต้นได้ทำการเคลื่อนที่มือไปใน 3 รูปแบบคือเคลื่อนที่เป็นเส้นตรง เส้นโค้ง และเส้นรูปตัววี แล้วเก็บตำแหน่งของมือ ณ เวลาต่างๆ ผลก็คือ มีการแกว่งของจุดข้อมูลทำให้เมื่อเคลื่อนที่มือเป็นเส้นตรงหรือเส้นอื่นๆ ไม่ได้เป็นไปตามรูปแบบเส้นนั้นๆ ในอุดมคติจริง ดังนั้นจึงได้ทำการปรับปรุงการติดตามมือที่นำตัวกรองคาลแมนเข้ามาใช้งานเพื่อทำการประมาณค่าตำแหน่งมือใหม่โดยประมาณค่าจะค่าที่ทำนายได้กับค่าที่วัดได้จริงทำให้ได้ตำแหน่งที่เหมาะสมและไม่เกิดการแกว่งของจุดข้อมูลส่งผลให้การเคลื่อนที่มือเป็นเส้นตรงหรือเส้นอื่นๆ มีความใกล้เคียงกับรูปแบบเส้นนั้นๆ ในอุดมคติมากยิ่งขึ้น ผลจากการติดตามการเคลื่อนที่มือจะทำให้ทราบถึงทิศทางการเคลื่อนที่ ขนาด และมุมของมือ ที่จะถูกนำไปใช้ต่อในกระบวนการรู้จำท่าทางมือ

5.1.3. สรุปผลการทดลองกระบวนการรู้จำท่าทางมือ

การรู้จำท่าทางมือจะถูกแบ่งออกเป็น 2 แบบคือ (1) การรู้จำท่าทางมืออย่างง่ายเช่น การเคลื่อนมือไปทางซ้าย-ขวา บน-ล่าง เป็นต้น โดยใช้ไฟในดีสเคดแมชชีนแบ่งท่าทางมือต่างๆ ออกเป็นสถานะการทำงาน และใช้คุณสมบัติของมือที่ได้จากการติดตามเป็นตัวกำหนดในการเปลี่ยนแปลงสถานะการทำงาน (2) การรู้จำท่าทางมือแบบสัญลักษณ์เช่น การเคลื่อนที่มือเป็นรูปทรงต่างๆ โดยใช้แบบจำลองฮิดเดนมาร์คอฟเพื่อสร้างแบบจำลองที่ใช้อธิบายท่าทางมือต่างๆ ภายใต้สภาวะแวดล้อมภายในอาคาร โดยใช้กล้องเพียงตัวเดียวและกล้องไม่เคลื่อนที่ การรู้จำท่าทางมืออย่างง่ายถูกนำไปประยุกต์ใช้กับโปรแกรมวินโดวมีเดียเซ็นเทอในระบบปฏิบัติการวินโดว 7 ความถูกต้องในการรู้จำจะสูงถ้ามีการเคลื่อนที่มือช้าๆ และระยะห่างระหว่างกล้องกับมือไม่มากจนเกินไป ส่วนการรู้จำท่าทางมือแบบสัญลักษณ์ ซึ่งสัญลักษณ์ในที่นี้ได้ใช้เป็นตัวอักษรภาษาไทยที่มีความแตกต่างกันอย่างชัดเจน 12 แบบเป็นสัญลักษณ์ในการทดสอบการรู้จำ ผลการทดสอบความถูกต้องในการรู้จำโดยเฉลี่ยอยู่ที่ 83 เปอร์เซ็นต์ โดยใช้กลุ่มตัวอย่างท่าทาง 50 ชุดในแต่ท่าทางเพื่อทำการฝึกสอนและฝึกสอนด้วยจำนวนสถานะ 10 สถานะซึ่งเพียงพอสำหรับการรู้จำท่าทางมือ

5.2. บทวิจารณ์

5.2.1. บทวิจารณ์ผลการทดลองการตรวจจับลักษณะมือ

การเรียนรู้แบบอนุกรมเป็นวิธีการที่ถูกลำเอียงมาใช้ในกระบวนการตรวจจับลักษณะวัตถุที่สนใจ นอกจากนี้ยังมีวิธีการเรียนรู้แบบอื่นๆ อีกเช่น การเรียนรู้แบบอนุกรมเชิงไม่ต่อเนื่อง (Discrete Ada Boost) การเรียนรู้แบบอนุกรมเชิงแท้ (Real Ada Boost) และการเรียนรู้แบบอนุกรมเชิงตรรก (Logit Boost) ซึ่งสามารถนำมาใช้กระบวนการตรวจจับวัตถุที่สนใจได้เช่นกัน ดังนั้นจึงควรมีการเปรียบเทียบผลลัพธ์ของการเรียนรู้ในแต่ละแบบ และหาว่าอัตราการเจอและความผิดพลาดเชิงบวกของการเรียนรู้ในแต่ละแบบมีความแตกต่างกันอย่างไร เพื่อหาว่าวิธีการไหนที่มีประสิทธิภาพในกระบวนการตรวจจับมากที่สุด โดยใช้กลุ่มตัวอย่างที่เท่ากัน

5.2.2. บทวิจารณ์ผลการทดลองกระบวนการติดตามการเคลื่อนที่ของมือ

กระบวนการติดตามด้วยตัวกรองคาลแมนที่นำมาใช้เพื่อช่วยเพิ่มประสิทธิภาพในการติดตามด้วยคุณสมบัติของสีถึงแม้ในบางกรณีจะมีความผิดพลาดในการติดตามมากกว่า แต่ก็ยังคงต้องใช้ค่าที่ได้จากการติดตามด้วยตัวกรองคาลแมนเนื่องจากจะไม่เกิดการแกว่งของจุดข้อมูลที่ได้ ทำให้การติดตามเคลื่อนที่ใกล้เคียงกับที่มือเคลื่อนที่จริง ซึ่งทำให้สามารถบอกทิศทางของการเคลื่อนที่ได้ถูกต้องมากยิ่งขึ้น

5.2.3. บทวิจารณ์ผลการทดลองกระบวนการรู้จำท่าทางมือ

กระบวนการรู้จำโดยใช้แบบจำลองฮิดเดนมาร์คอฟที่ใช้ในงานวิจัยชิ้นนี้เป็นแบบจำลองประเภทเชื่อมโยงถึงกันหมด ซึ่งเป็นแบบจำลองพื้นฐานที่ใช้กันมากในกระบวนการรู้จำ แต่ยังมีแบบจำลองฮิดเดนมาร์คอฟประเภทอื่นๆ อีกเช่น แบบจำลองฮิดเดนมาร์คอฟประเภทซ้าย-ขวา แบบจำลองฮิดเดนมาร์คอฟประเภทลำดับชั้นซ้าย-ขวา ซึ่งสามารถนำมาใช้งานในกระบวนการรู้จำได้เช่นกัน ดังนั้นควรมีการเปรียบเทียบผลการรู้จำระหว่างแบบจำลองฮิดเดนมาร์คอฟที่ใช้กับแบบจำลองฮิดเดนมาร์คอฟประเภทอื่นๆ ของแบบจำลองฮิดเดนมาร์คอฟ เพื่อหาความถูกต้องในการรู้จำของแบบจำลองฮิดเดนมาร์คอฟแต่ละประเภทให้ผลลัพธ์ที่แตกต่างกันอย่างไร และในส่วนของคุณลักษณะเด่นที่จะนำมาใช้เป็นตัวอย่งในการฝึกสอนควรมีการทดลองเพิ่มคุณลักษณะเด่นแบบอื่นๆ เช่น ความเร็ว ความเร่ง นอกเหนือจากทิศทางและการเคลื่อนที่ของมือ ซึ่งอาจช่วยเพิ่มประสิทธิภาพในกระบวนการรู้จำให้มีความแม่นยำมากยิ่งขึ้น

บรรณานุกรม

- [1] Wei Du and Hua Li, "Vision based gesture recognition system with single camera," 5th International Conference on ICSP, pp.1351-1357 vol.2, 2000.
- [2] Qing Chen, Georganas N.D., Petriu E.M., "Real-time Vision-based Hand Gesture Recognition Using Haar-like Features," Instrumentation and Measurement Technology Conference Proceedings, 2007.IMCT 2007. IEEE, pp.1-6, 1-3 May 2007.
- [3] Elmezain M., Al-Hamadi, "Gesture Recognition for Alphabets from Hand Motion Trajectory Using Hidden Markov Models," Signal Processing and Information Technology, 2007 IEEE International Symposium, pp.1192-1197, 15-18 Dec. 2007.
- [4] Jakub Segen and Senthil Kumar, "Human-Computer Interaction using gesture Recognition and 3D Hand Tracking," Proceedings of ICIP'98, Vol. 3, pp188-192, Chicago, October 1998.
- [5] Akira Utsumi, Tsutomu Miyasato, Fumio Kishino, Ryohei Nakatsu, "Hand Gesture Recognition System using Multiple Cameras," Proceedings of ICPR'96, Vol. 1, pp667-671, August 1996
- [6] Francis K.H.Quek, "Non-Verbal Vision-Based Interfaces," Vision Interfaces and Systems Laboratory Technical Report 1994, Electrical Engineering and Computer Science Department, The University of Illinois at Chicago.
- [7] Jochen Triesch, Christoph von der Malsburg, "Robotic Gesture Recognition," Proceedings of ECCV'98, pp233-244, 1998.
- [8] Viola P. and Jones M., "Rapid object detection using a boosted cascade of simple features," Proceedings of the IEEE CVPR, vol.1, pp. I577-I-518 vol.1, 2001.
- [9] Lienhart R. and Maydt J., "An extended set of Haar-like features for rapid object detection," International Conference on Image Processing, Vol.1, pp. I-900-I-903 vol.1, 2002.
- [10] R. C. Gonzalez, R. E. Woods, *Digital Image Processing*. Prentice Hall Inc., 2002.
- [11] G. Bradski and A. Kaebler, *Learning OpenCV Computer Vision with the OpenCV Library*. O'Reilly Media Inc., 2008.

- [12] Greg Welch and Gary Bishop, "An Introduction to the Kalman Filter," Annual Conference on Computer Graphics & Interactive Techniques. ACM Press, Addison-Wesley, Los Angeles, CA, USA (August 12-17), SIGGRAPH 2001 course pack edition.
- [13] L.R. Rabiner, "A tutorial on hidden markov models and selected application in speech recognition," Proc. IEEE 77, 1989, pp. 267-293
- [14] Ramer-Douglas-Peucker algorithm, July 2010, Available at (online): http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorithm.
- [15] Finite-state machine, March 2011, Available at (online): http://en.wikipedia.org/wiki/Finite-state_machine
- [16] Thai Script, October 2010, Available at (online): http://en.wikipedia.org/wiki/Thai_script
- [17] Ahmad, A.M.; Bade, A.; Abidin, L.A.-H.Z.; , "Using Principal Component Analysis and Hidden Markov Model for Hand Recognition Systems," Information and Multimedia Technology, 2009. ICIMT '09. International Conference on , vol., no., pp.323-326, 16-18 Dec. 2009
- [18] Elmezain, M.; Al-Hamadi, A.; Pathan, S.S.; Michaelis, B.; , "Spatio-temporal feature extraction-based hand gesture recognition for isolated American Sign Language and Arabic numbers," Image and Signal Processing and Analysis, 2009. ISPA 2009. Proceedings of 6th International Symposium on , vol., no., pp.254-259, 16-18 Sept. 2009
- [19] Matsuo, T.; Shirai, Y.; Shimada, N.; , "Automatic generation of HMM topology for sign language recognition," Pattern Recognition, 2008. ICPR 2008. 19th International Conference on , vol., no., pp.1-4, 8-11 Dec. 2008
- [20] Van Hieu, D.; Nitsuwat, S.; , "Image Preprocessing and Trajectory Feature Extraction based on Hidden Markov Models for Sign Language Recognition," Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. SNPD '08. Ninth ACIS International Conference on , vol., no., pp.501-506, 6-8 Aug. 2008

ภาคผนวก

ภาคผนวก ก.

ผลงานตีพิมพ์เผยแพร่จากวิทยานิพนธ์

1. Kittasil Silanon and Nikom Suvonvorn, “Real Time Hand Tracking as a User Input Device,” *Proceedings of the Fifth International Conference on Knowledge, Information and Creativity Support Systems*, November 25-27, 2010, Chiang Mai, Thailand, pp. 153 - 158.
2. Kittasil Silanon, Nikom Suvonvorn and Montri Karnchanadecha, “Thai Alphabet Recognition from Hand Motion Trajectory Using HMM,” *2011 3rd International Conference on Signal Acquisition and Processing*, Singapore, February 26-28, 2011, pp. V1-189 – V1-193.

KICSS²⁰¹⁰

Proceedings of the Fifth International Conference on
Knowledge, Information and Creativity Support Systems

KICSS 2010

November 25–27, 2010
Chiang Mai, Thailand

Thanaruk Theeramunkong, Virach Sornlertlamvanich,
and Susumu Kunifuji (Eds.)

Real Time Hand Tracking as a User Input Device

Kittasil Silanon¹ Nikom Suvonvorn²

^{1,2}Department of Computer Engineering

Faculty of Engineering, Prince of Songkla University,
Hat Yai, Thailand

jalego3@hotmail.com¹, kom@coe.psu.ac.th²

Abstract

In this paper, we propose a system that facilitates the two dimension user-input as general mouse device for controlling applications. The method is based on hand movement analysis by applying image processing technique. The Haar-like with a cascade of boost classifiers is applied for hand detection, and then tracked with skin color using CamShift. Extracted hand features are computed and used for recognizing commands via a finite state machine. We evaluated the performance of system under real time constraint in real environment with demonstration application.

Keywords: Hand detection, Hand Tracking, Command Recognition

1. Introduction

Trends in human machine interfacing have invented very rapid in consumer devices such as multi-touch of mini-computer like iPad, mobile phone iPhone, motion sensing of game devices, and etc. Relatively, many applications to support these platforms are actively developed. In this paper, we introduce a hand motion capture system using single camera that enables to track 2D hand position for commanding the desirable applications in real-time. The system offers very easy way of short range human-machine interfacing, which is very effective in terms of application constraints and prices, and generally very appropriate for the PC-based applications without any additional integrated material devices.

There are many research try to solve the same problem. Wei Du [1] introduced a system with one camera that can recognize four gestures and tracking hand by extracting the feature points on hand contour. Nebojsa Jovic [2] developed a real-time system for detecting pointing gestures and estimating the direction of pointing using stereo cameras for controlling the cursor on a wall screen. Juan [3] proposed a vision-based system that can interpret a user's hand gesture in

real time to manipulate objects and optimized for the medical data visualization environment.

We propose a very simple system that used only a consumer grade webcam as additional material, generally installed on top of the computer screen looking down towards the user's hands, for interfacing with applications. Our main contribution is then focused on the image processing techniques that try to detect hand pose and extract hand features, such as position, axis, and angle, which are used later in the command recognition process. Figure 1 shows example of system.

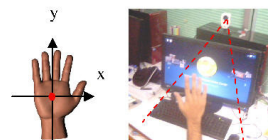


Figure 1. Hand movement tracking as user input device.

2. Proposed System

In this section, we introduce the system for command recognition based on hand motion by analyzing the image sequences, obtained from the consumer grade webcam attached over the computer screen. Images are captured and analyzed in order to extract the essential parameters of the specific pose of hand. The parameters are reformulated into hand features which is necessary for command recognition.

There are two main parts of our system: hand detection and tracking, and command recognition. The overall of system is illustrated in the figure 2. First part, a hand detector is implemented using a cascade of boost classifiers, which allows obtaining very robust object detector. Two types of hand pose with vertical position are considered such as open and close hand, which is necessary for executing a command as start and stop symbol respectively. However, hand pose can have many figures caused by translation and rotation in 3D, which needs to be found. The tracking method, CamShift and Kalman filter, is then applied in order to extract hand through image sequence. In the second

part, hand parameters are determined such as position, axis, area, and angle, which are defined as hand feature. A Finite Stage Machine is established based on these features for recognizing commands.

Figure 2 (left) shows the sequence of steps for hand detection and tracking; figure 2 (right) shows the recognition process. These two parts will be detailed in the section 3 and 4 respectively. In section 5, the experimentation result is discussed for evaluating the performance of system. For testing in real environment, we have implemented the system in windows 7, which our module can control any applications as a mouse interface. In the last section, we conclude our method.

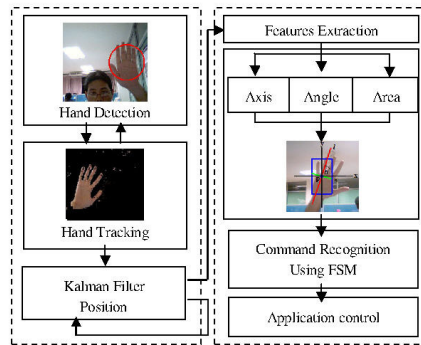


Figure 2. System overview: (left) detection and tracking. (right) recognition process.

3. Hand Detection and Tracking

3.1 Hand Detection

A Boost Cascade of Classifier [4][5] for object detection is originally developed by Viola and Jones. This method uses Haar-like features and a cascade of boosted tree classifier as a supervised statistical model of object recognition. Haar-like feature consists of two or three connected “black” and “white” rectangles. The feature value is defined by the difference between the sum of pixel values within the black and white rectangles. Figure 2 shows a basic set of Haar-like features.

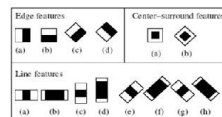


Figure 3. A set of Haar-like feature.

The AdaBoost algorithm is introduced to improve the classification performance that designed to select rectangle feature which best separates the positive and negative example. In the first iteration, the algorithm train a weak classifier $h(x)$ using one Haar-like feature that achieves the best recognition performance for the training samples. The classifier consist of a feature $f(x)$, a threshold θ_i and parity p_i indicating the direction of inequality sign.

$$h_i(x) = \begin{cases} 1 & \text{if } p_i f_i(x) < p_i \theta_i \\ 0 & \text{otherwise} \end{cases}$$

In the second iteration, the training samples that were misclassified by the first weak classifier receive higher weights. The iteration goes on and the final result is a cascade of linear combinations of the selected weak classifiers $H(x)$, which achieves the required accuracy.

$$H(x) = \sum_{t=1}^T \alpha_t h_t(x)$$

In the detection process by using the trained cascade, the sub-windows must be test each stage of the cascade. A negative outcome at any point leads to the immediate rejection of the sub-window.

In our technique, a command begins with open hand gesture and stop using close hand gesture. Each frame, these two gestures must be detected. In the training processing, we have collected ~5,000 positive samples for each hand style from students in our university in various conditions such as: indoor with neon light and outdoor with natural light. Systematically, around 10,000 negative samples are selected from landscape, building, and human faces or body images. Figure 4 show some of these samples.

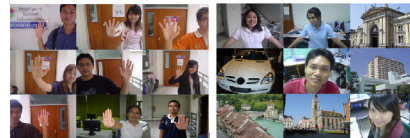


Figure 4. (left) positive samples. (right) negative samples.

Each gesture type was trained two rounds. The first round is for discriminating quickly the target gesture from the outliers with small samples. By observing the false positive and false negative, we determine the additional positives and negative samples in order to overcome the better recognition rate while reducing noise.



Figure 5. Hand detection results: (left) 1st round. (right) 2nd round.

Table 1 show the detection rate results form our experimentation of the two hand gestures with respective configurations. We notice that a good detection rate is risen at least more than 91% with 16 training stage at minimum.

Table 1. Hand detection rate.

Hand Gesture	Window Size	Training and Performance			
		Positive	Negative	Stage	Rec. Rate
Open	32x32	5,058	10,448	19	91.17%
Close	32x32	5,678	10,448	16	99.06%

3.2 Hand Tracking

CamShift [6] is a non-parametric technique using for color object tracking deriving from the Mean Shift algorithm. The main difference between CamShift and Mean Shift algorithm is that CamShift updates continuously its probability distributions; in generally the target object in image sequences changes significantly its shape size or color, while Mean Shift is based on static distributions. That why CamShift is suitable for tracking the rigid object.

In the hand tracking, the process can be described as the following.

Step 1: the color probability distribution of detected hand image is determined from its histogram via hue component of HSV color space, related to skin color.

Step 2: this target distribution of detected hand is systematically tracked on the searching window of next frame in image using mean shift algorithm. The mean shift vector, which is aimed for finding an optimized path that climbs the gradient of a probability distribution to the mode (peak) of nearest dominant peak, is necessary to be computed.

Step 3: the back-projection technique, which associates the pixel values in the image (tracking hand) with the value of the corresponding distribution, is applied.

Step 4: the center of mass and size of tracking hand (projected image) is computed and defined as

hand features. This step will be detailed in the next section (hand features).

Step 5: on the next iterative, the current position of hand in image is used for defining the searching window on the next frame. The process is repeated at step 2 continuously.

Note that the step 1 will be re-executed systematically if the detected hand by Haar-like features with boost cascade of classifier found in the searching window. We found that Haar-like method provides very accurate results when hand is paralleled to the vertical axis, compared with CamShift, but missed mostly in other directions, so that CamShift is applied in order to solve the problem.



Figure 6. Hand tracking with CamShift.

3.3 Hand Movement Estimation

The Kalman filter [7] is a recursive linear filtering method. It addresses the general problem of trying to estimate the state of discrete time process that is described by the linear stochastic differential equation by the following.

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$$

With a measurement $z \in \mathfrak{R}^m$ that is

$$z_k = Hx_k + v_k$$

The random variables w_k and v_k represent Gaussian noise of the process and measurement respectively. The algorithm of Kalman filter estimates a process by using feedback control technique: estimating the process state by an appropriate model and doing feedback by noisy measurements. As such, the equations of Kalman filter are formed into two groups: prediction and correction equations.

In the post tracking of hand, the algorithm can be described as the following.

Step 1: Initialization state

Initial estimates for \hat{x}_{k-1} and P_{k-1}

Step 2: Prediction state

1) Estimate the next state $\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$

2) Estimate the error covariance of next state

$$P_k^- = AP_{k-1}A^{-1} + Q$$

Step 3: Measurement state

- 1) Compute the Kalman gain

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$

- 2) Correct with measurement w_k

$$\hat{x}_k = \hat{x}_{k-1} + K_k (z_k - H u_{k-1} \hat{x}_{k-1})$$

- 3) Update the error covariance

$$P_k = (I - K_k H) P_k^-$$

Step4. Repeat the prediction at step 2.

Experimentally, we found that hand detector by Haar-like features with boost cascade of classifier cannot provide total results and CamShift may wrongly track especially when there are other parts of body such as face or background, having color in skin color range, move close to tracking hand. Therefore, the Kalman filter is applied in order to predict hand position in the next frame based on previous frame, obtained by CamShift. To apply the Kalman algorithm, two principle equations needed to be declared: tracking process and measurement. The state of tracking process is measured from hand position and velocity in each image frame. So, we define the process of state x_k by the following:

$$\begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_{k,i} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_{k-1,i} + w_{k-1,i}$$

Where x , y , v_x , v_y are the position and the velocity of hand in the k^{th} image frame respectively. Here, we assume that that the motion of hand between two successive frames can be uniformly approximated as straight line, the frame interval Δt is very short. For the measurement z_k , we define directly by the position value obtained from CamShift algorithm.

4. Command Recognition

4.1 Hand features

During the step 4 of hand tracking process by CamShift algorithm, in each frame, the hand characteristics are computed. The following parameters are considered: center of mass (p), axis angle (θ), area (z) and distance (d). The center of mass of the hand in a frame can be easily computed from the 0^{th} and 1^{st} order moments of pixels in hand's region.

Step 1: Find the 0^{th} moment: $M_{00} = \sum_x \sum_y I(x,y)$

Find the 1^{st} moment:

$$M_{10} = \sum_x \sum_y x I(x,y), \quad M_{01} = \sum_x \sum_y y I(x,y)$$

Find 2^{nd} moment:

$$M_{20} = \sum_x \sum_y x^2 I(x,y), \quad M_{02} = \sum_x \sum_y y^2 I(x,y)$$

Step 2: the center of mass X_c, Y_c is calculated as

$$X_c = \frac{M_{10}}{M_{00}} \text{ and } Y_c = \frac{M_{01}}{M_{00}}$$

In the above equations, $I(x,y)$ is the pixel value at the position (x,y) of the image, x and y are range over the hand's region. At the same time, calculate the 2^{nd} moment; obtain length-axis, short-axis and angle of the hand's region.

Step 3: the angle of hand is

$$\theta = \frac{1}{2} \arctan \left[\frac{2 \left(\frac{M_{11}}{M_{00}} - X_c Y_c \right)}{\left(\frac{M_{20}}{M_{00}} - X_c^2 \right) - \left(\frac{M_{02}}{M_{00}} - Y_c^2 \right)} \right]$$

Step 4: the long-axis l , short-axis w of tracking hand can be presented as :

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}}, \quad w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}}$$

$$a = \frac{M_{20}}{M_{00}} - X_c^2, \quad b = 2 \left(\frac{M_{11}}{M_{00}} - X_c Y_c \right), \quad c = \frac{M_{02}}{M_{00}} - Y_c^2$$

When used in tracking, the above equations give us angle, length-axis and width-axis. Then, we use axis for determined area (z) is $l \times w$ and distance (d) is $\sqrt{l^2 + w^2}$.

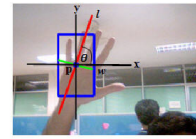


Figure 7. Features of the hand's region.

By considering the parameters with respect to time t , we formulate the hand feature $f(t)$ by the following :

$$f(t) = [p_i(x_c, y_c) \quad \theta_i \quad z_i \quad d_i]$$

4.2 Command recognition

A simple recognition system of commands based on grid is established in order to interface the hand movement as user input of interactive application. The following commands are considered: (1) cursor displacement in four directions, and (2) clicks: left, middle, right. The main idea for designing the command recognition system is that to use the extracted parameters as features f .

For displacement commands, only $\partial p_i / \partial t$ is used for constructing command grid. A dynamic grid is used for command recognition. When hand has detected, which the 3x3 grid is created enclosing its center $p(x_c, y_c)$. Then, move hand from center to the neighborhood cells will activate the displacement command with respect to the specific direction. For the click commands, we consider the value of $\partial \theta_i / \partial t$ and $\partial \alpha_i / \partial t$ defined by the following: click left if only if $\partial \theta_i / \partial t$ is positive, click right if only if $\partial \theta_i / \partial t$ is negative, and middle click if only if $\partial \alpha_i / \partial t$ is less than $\pi/2$. Figure 8 summarizes the FSM states and transitions.

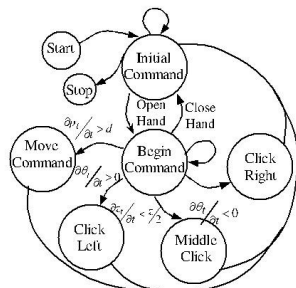


Figure 8. Finite State Machine for command recognition.

Figure 9 shows the example result of move command and click commands, for click left, middle click, and click right respectively.

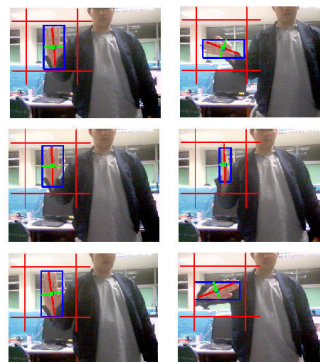


Figure 9. Example of move command, left click, middle click and right click commands respectively.

5. System Performance and Discussion

Our system is implemented using OpenCV library. Testing system is run on Intel processor Core 2 Duo, 2 Ghz, 2 GB memory. The video sequence is analyzed with image resolutions 320x240 pixels at 30 fps.

5.1 Hand tracking

We evaluate the precision of trajectory obtained from hand tracking process. The test is done by executing the hand movement following three types of ideal path: straight line, curve and v-shape respective. Then, we compute the errors, defined by the difference between ideal path and tracking position. Table 2 shows results of ten times test with average. We found that the error in average of curve is bigger than v-shape and line respectively.

Table2. Hand tracking with errors

No	Testing errors (pixels)		
	Line	Curve	V-shape
1	1.54	3.79	8.05
2	1.70	3.61	3.31
3	1.39	6.87	4.22
4	1.59	6.68	9.13
5	1.97	7.55	4.98
6	0.97	8.93	3.51
7	1.40	8.09	4.31
8	2.02	4.23	4.44
9	1.76	6.05	7.05
10	1.84	6.97	6.38
μ	1.61	6.27	5.53
σ	0.29	1.74	1.89

5.2 Command recognition

The performance of command recognition is tested depends on two factors: distance of hand from camera, and hand speed for executing a command. Testing is done under indoor environment, no change of light condition, with the following test: three fix distances with three levels of speed, ten times of testing in each case.

Table3. Experimentation results

Distance (cm)	Hand speed (s)		
	0.5s	1s	2s
30	10/10	10/10	10/10
60	10/10	10/10	10/10
90	7/1	8/10	9/10

Table 3 shows our testing results. We note that at low speed commands can be executed more precise than high speed one. Intuitively, during recognition process more frames are analyzed under low speed that increases eventually the effectiveness of the method. We found that the commands at near distance provide high recognition rate than far distance. Certainly, when the image region of hand in image is too small, then hand cannot be detected.

We also demonstrate our system as a user input that interface to Window Media Center application on Windows 7. You can see video showing the real-time interactive at

<http://www.youtube.com/watch?v=loVxPEP5fME>.

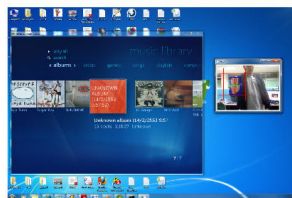


Figure 10. Our module is implemented in Windows 7 for controlling Window Media Center.

6. Conclusion

We have introduced the image processing methods for hand detection tracking and recognition aimed for human applications interfacing like a general mouse input. The hand trajectory and other features are precisely extracted, using for command recognition. Eleven basic commands are recognized at very high rate with respect to different speeds and

distances. The implementation of our system on Windows 7 shows an example test under real-time constraint with real application.

References

- [1] Wei Du and Hua Li , "Vision based gesture recognition system with single camera", 5th International Conference on ICSP, vol.2, no., pp.1351-1357 vol.2, 2000
- [2] Jojic N., Brumitt B., Meyers B., Harris S. and Huang T. , "Detection and estimation of pointing gestures in dense disparity maps," Fourth IEEE International Conference on Automatic Face and Gesture Recognition, vol., no., pp.468-475,
- [3] J. Wachs, H. Stern, Y. Edan, M. Gillam, C. Feied, M. Smith, and J. Handler,"A Real-Time Hand Gesture System Based on Evolutionary Search," *Vision*. vol. 22, no. 3, Oct. 2006, Dearborn, Mich, Society of Manufacturing Engineers.
- [4] Viola, P. and Jones M., "Rapid object detection using a boosted cascade of simple features," *Proceedings of the IEEE CVPR*, vol.1, no., pp. I-511- I-518 vol.1, 2001
- [5] Lienhart R. and Maydt J., "An extended set of Haar-like features for rapid object detection," *International Conference on Image Processing*, vol.1, no., pp. I-900- I-903 vol.1, 2002
- [6] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal*, Q2 1998.
- [7] Greg Welch and Gary Bishop, "An Introduction to the Kalman Filter," *Annual Conference on Computer Graphics & Interactive Techniques*. ACM Press, Addison-Wesley, Los Angeles, CA, USA (August 12–17), SIGGRAPH 2001 course pack edition.

2011 3rd International Conference on Signal Acquisition and Processing

ICSSAP 2011

Singapore, February 26-28, 2011

Editor: Dr. Steve Thatcher and Dr. Venkatesh

© 2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.



IEEE Catalog Number: CFP1140G-PRT

ISBN: 978-1-4244-9385-2

Thai Alphabet Recognition from Hand Motion Trajectory Using HMM

Kittasil Silanon¹, Nikom Suvonvorn², Montri Karnchanadecha³

^{1,2,3}Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University,
Hat Yai, Songkhla, Thailand 90112
jalego3@hotmail.com¹, kom@coe.psu.ac.th², montri@coe.psu.ac.th³

Abstract — In this paper, we propose a system for Thai alphabet recognition from hand movement trajectory, as a human-computer interaction method. Thai characters drawing by hand movement are analyzed and recognized by our system, which can apply for controlling any specific tasks. The method is based on hand motion analysis combining with Haar-like with a cascade of boost classifiers, as hand detection method. Hand is tracked with skin color using CamShift and Kalman filter. Trajectory features of hand are extracted and used for recognizing 12 Thai alphabet letters through the Hidden Markov Model.

Keywords- Thai Alphabet; Hand Movement Trajectory Recognition; HMM.

I. INTRODUCTION

In the field of computer vision, communication between human and computer becomes more important. Human-Computer Interaction (HCI), which allows humans communicate with computer have been a popular research field for many years. Some active research in this field are human face recognition, eye gaze tracking, lip reading, hand gesture recognition and body pose tracking. In this paper, we emphasize on hand movement, which can freely moves and gesticulates more than other parts of body therefore hand movement gesture recognition can be applied in many applications such as sign language recognition, computer-controlled game, teleconference, and so on. Many researches of hand gesture recognition have been proposed, covering a wide variety of methods and approaches. For example, Mahmoud et. al. [1] has developed a system that could recognize gesture for alphabets from hand motion using Hidden Markov Model (HMM). Juan et. al. [2] introduced a vision-based system that can interpret a user's hand gesture in real time to manipulate objects within a medical data visualization environment. Dinh et. al. [3] proposes a hand gesture classification system that able to efficiently recognize 24 basic signs of American Sign Language with Haar-like feature and AdaBoost learning algorithm.

In this paper, we proposed a system that can recognize the 12 important Thai alphabet letters selecting from the specific group using hand movement trajectory features. The paper is organized in six sections by the following: proposed hand movement recognition system, hand detection and tracking, trajectory feature extraction, experimentation results, and conclusion respectively.

II. PROPOSED SYSTEM

In this section, we introduce the system for hand gesture recognition based on hand motion by analyzing the image sequences, obtained from the webcam attached over the computer screen. There are three main parts of our system: hand detection and tracking, hand feature extraction and gesture recognition. The overall of system is illustrated in the figure 2. First part, a hand detector is implemented using a cascade of boost classifiers, which allows obtaining very robust object detector. Two types of hand pose with vertical position are considered such as open and close hand, which is necessary for executing a command as start and stop symbol respectively. However, hand pose can have many figures caused by translation and rotation in 3D, which needs to be found. The tracking method, CamShift and Kalman filter, is then applied in order to extract hand through image sequence. In the second part, the gesture trajectory features are extracted and enhanced by the Douglas-Peucker algorithm which gives us a smallest number of control points trajectory. The features are selected to give a discrete vector that use as input to discrete HMM for recognizing commands in the final part.

These three parts will be detailed in the section 3, 4 and 5 respectively. In section 6, the experimentation result is discussed for evaluating the performance of system.

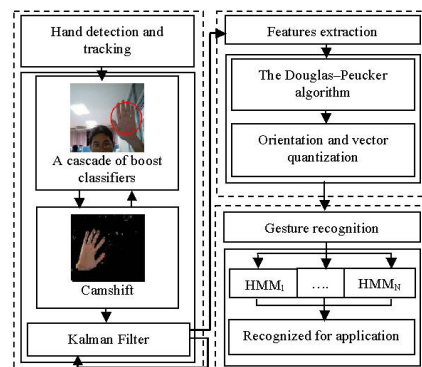


Figure 1. System overview : (left) detection and tracking, (right) feature extraction and recognition process.

III. HAND DETECTION AND TRACKING

A. Hand Detection

A Cascade of Boost Classifier [4][5] for object detection is originally developed by Viola and Jones. This method uses Haar-like features and a cascade of boosted tree classifier as a supervised statistical model of object recognition. Haar-like feature consists of two or three connected "black" and "white" rectangles. The feature value is defined by the difference between the sums of pixel values within the black and white rectangles. Figure 2 shows a basic set of Haar-like features.

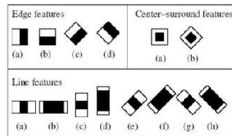


Figure 2. A set of Haar-like feature.

The AdaBoost algorithm is introduced to improve the classification performance that designed to select rectangle feature which best separates the positive and negative example. In the first iteration, the algorithm train a weak classifier $h(x)$ using one Haar-like feature that achieves the best recognition performance for the training samples. The classifier consist of a feature $f(x)$, a threshold θ , and parity p , indicating the direction of inequality sign.

$$h_t(x) = \begin{cases} 1 & \text{if } p_t f_t(x) < p_t \theta_t \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

In the second iteration, the training samples that were misclassified by the first weak classifier receive higher weights. The iteration goes on and the final result is a cascade of linear combinations of the selected weak classifiers $H(x)$, which achieves the required accuracy.

$$H(x) = \sum_{t=1}^T \alpha_t h_t(x) \quad (2)$$

In the detection process by using the trained cascade, the sub-windows must be test each stage of the cascade. A negative outcome at any point leads to the immediate rejection of the sub-window.

In our technique, a command begins with open hand posture and stop using close hand posture. Each frame, these two gestures must be detected. In the training processing, we have collected around 5,000 positive samples for each hand style from students in our university in various conditions such as indoor with neon light and outdoor with natural light. Systematically, around 10,000 negative samples are selected from landscape, building, and human faces or body images. Figure 3 show some of these samples. Each posture type was trained two rounds. The first round is for discriminating quickly the target posture from the outliers with small samples. By observing the false positive

and false negative, we determine the additional positives and negative samples in order to overcome the better recognition rate while reducing noise as shown in figure 4.



Figure 3. (left) positive samples, (right) negative samples.

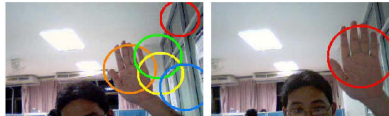


Figure 4. Hand detection results : (left) 1st round, (right) 2nd round.

Table 1 show the detection rate results form our experimentation of the two hand postures with respective configurations.

Table1. Hand detection rate.

Hand Posture	Window Size	Training and Performance			
		Positive	Negative	Stage	Rec Rate
Open	32x32	5,058	10,448	19	91.17%
Close	32x32	5,678	10,448	16	99.06%

B. Hand Tracking

CamShift [6] is a non-parametric technique using for color object tracking deriving from the Mean Shift algorithm. The main difference between CamShift and Mean Shift algorithm is that CamShift updates continuously its probability distributions; in generally the target object in image sequences changes significantly its shape size or color, while Mean Shift is based on static distributions. That why CamShift is suitable for tracking the rigid object. In the hand tracking, the process can be described as the following.

Step 1: the color probability distribution of detected hand image is determined from its histogram via hue component of HSV color space, related to skin color.

Step 2: this target distribution of detected hand is tracked on the searching window of next frame in image using mean shift algorithm. The mean shift vector, which is aimed for finding an optimized path that climbs the gradient of a probability distribution to the mode (peak) of nearest dominant peak, is necessary to be computed.

Step 3: the back-projection technique, which associates the pixel values in the image (tracking hand) with the value of the corresponding distribution, is applied.

Step 4: the center of mass of tracking hand is computed and defined as hand trajectory. This step will be detailed in the next section (trajectory parameters).

Step 5: on the next iterative, the current position of hand in image is used for defining the searching window on the next frame. The process is repeated at step 2 continuously.

Note that the step 1 will be re-executed systematically if the detected hand by Haar-like features with boost cascade of classifier found in the searching window. We found that Haar-like method provides very accurate results when hand is paralleled to the vertical axis, compared with CamShift, but missed mostly in other directions, so that CamShift is applied in order to solve the problem.



Figure 5. Hand tracking with CamShift

C. Hand Movement Estimation

The Kalman filter [7] is a recursive linear filtering method. It addresses the general problem of trying to estimate the state of discrete time process that is described by the linear stochastic differential equation by the following.

$$x_k = Ax_{k-1} + Bw_{k-1} + v_{k-1} \quad (3)$$

With a measurement $z \in \mathbb{R}^m$ that is

$$z_k = Hx_k + v_k \quad (4)$$

The random variables w_k and v_k represent Gaussian noise of the process and measurement respectively. The algorithm of Kalman filter estimates a process by using feedback control technique: estimating the process state by an appropriate model and doing feedback by noisy measurements. As such, the equations of Kalman filter are formed into two groups: prediction and correction equations. In the post tracking of hand, the algorithm can be described as the following figure 6.

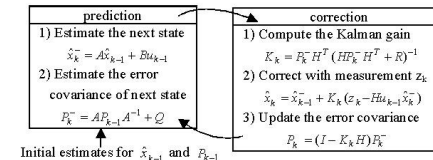


Figure 6. The operation of the Kalman filter.

Experimentally, we found that hand detector by Haar-like features with boost cascade of classifier cannot provide total results and CamShift may wrongly track especially when there are other parts of body such as face or background, having color in skin color range, move close to tracking hand. Therefore, the Kalman filter is applied in order to predict hand position in the next frame based on previous frame, obtained by CamShift. To apply the Kalman algorithm, two principle equations needed to be declared: tracking process and measurement. The state of tracking process is measured from hand position and velocity in each image frame. So, we define the process of state x_k by the following:

$$\begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_{k,i} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_{k-1,i} + v_{k-1,i} \quad (5)$$

Where x, y, v_x, v_y are the position and the velocity of hand in the k^{th} image frame respectively. Here, we assume that the motion of hand between two successive frames can be uniformly approximated as straight line, the frame interval Δt is very short. For the measurement z_k , we define directly by the position value obtained from CamShift algorithm.

IV. FEATURE EXTRACTION

A. Trajectory parameters

In order to form the trajectory of hand position from the step 4 of hand tracking process by CamShift algorithm and Kalman filter. We choose to use the center point of hand and the trajectory can be obtained by joining this point in every frame in the sequence, in each frame, we obtained the center point of hand's region that can be easily computed from the moments of pixels in hand's region, which is defined as:

$$M_{ij} = \sum_{x,y} x^i y^j I(x,y) \quad (6)$$

In the above equations, $I(x,y)$ is the pixel value at the position (x,y) of the image, x and y are range over the hand's region. The center point of hand (X_c, Y_c) is calculated as:

$$X_c = \frac{M_{10}}{M_{00}}, \quad Y_c = \frac{M_{01}}{M_{00}} \quad (7)$$

B. Trajectory approximation

Since during gesturing the hand does not move very fast, the position of hand does not change much from one frame to the next one. Therefore, for trajectory formation it may not be necessary to consider all the frames in a gesture sequence. Accordingly, we propose to select key trajectory point to be stored in the new sequence. This reduces the memory requirement as well as speeds up the trajectory matching during recognition process. The algorithm that we use to extract key trajectory point is the Douglas-Peucker algorithm [8]. The algorithm recursively divides the line. Initially it is given all the points between the first and last point (figure 8 (a)). It marks the first and last point. It then finds the point that is furthest from the line segment with the first and last points as end points (this point is obviously furthest on the curve from the approximating line segment between the end points). If the point is closer than threshold to the line segment then any points not currently marked to keep can be discarded without the smoothed curve being worse than threshold. If the point furthest from the line segment is greater than threshold from the approximation then that point must be kept. The algorithm recursively calls itself with the first point and the selected point and then with the selected point and the last point (which includes marking

the selected point being marked as kept).When the recursion is completed a new output curve can be generated.

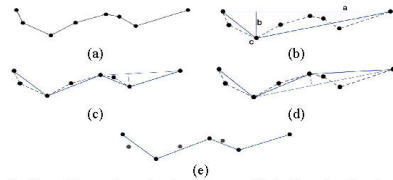


Figure 7. Smoothing a piecewise linear curve with the Douglas-Peucker algorithm.

C. Features

After getting the key trajectory points, we calculated the orientation between consecutive points to obtain a sequence of angle.

$$\theta_t = \arctan\left(\frac{Y_{t+1} - Y_t}{X_{t+1} - X_t}\right); t = 1, 2, \dots, T-1 \quad (8)$$

Where T represents the length of gesture trajectory. The angle's domain is [0, 360] degrees. We divide this angle by 20° to quatize them to 18 directional codewords from 1 to 18. The codewords is used as input to HMM recognition model.

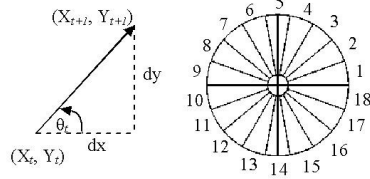


Figure 8. Codewords form orientation

Figure 9 show the result of gesture that was approximate by the Douglas-Peucker algorithm and define a gesture as a sequence of directional codewords which are O = { 13, 5, 3, 10 }

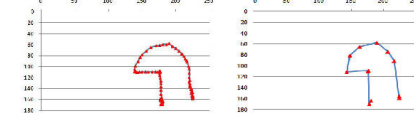


Figure 9. (a) gesture curve (b) gesture curve after approximate by the Douglas-Peucker algorithm

V. RECOGNITION MODEL

A. Hidden Markov Model

In the recognition process, the probabilistic approach, such as the Hidden Markov model [9], is applied for characterizing gestures representing the Thai alphabet letters. The HMM model can be defined by the following:

1). The set of states $S = \{s_1, s_2, \dots, s_N\}$ where N is number of states.

2). An initial probability for each state $\pi_i, i = 1, 2, \dots, N$ such that $\sum \pi_i = P(s_i)$ at the initial step.

3). An N-by-N transition probability matrix, $A = \{a_{ij}\}$, where a_{ij} is the transition probability of taking the transition from state i to state j.

4). The set of observation symbols $O = \{o_1, o_2, \dots, o_M\}$ representing our 18 directional codewords, M is the number of observation symbols.

5). An N-by-M observation matrix, $B = \{b_j(o_k)\}$ where $b_j(o_k)$ give the probability of emitting observation symbol o_k from state j.

An HMM requires specification of two model parameters (N and M), specification of observation symbols, and the specification of the three probability measures: A, B, and π . For convenience, we use the compact notation to indicate the complete parameter set of model.

$$\lambda = (A, B, \pi) \quad (9)$$

There are three basic problems for HMM. These problems are the following:

Evaluation problem: Given the observation sequence $O = O_1 O_2 \dots O_T$, and model $\lambda = (A, B, \pi)$, calculate the probability that model λ has generated sequence O.

Decoding problem: Given the observation sequence $O = O_1 O_2 \dots O_T$, and the model λ , calculate the most likely sequence of hidden states s_t that produced this observation sequence O.

Learning problem: How do we adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$.

The solutions to these three problems are Forward-Backward algorithm, the Viterbi algorithm, and the Baum-Welch algorithm respectively.

VI. EXPERIMENTATION RESULT

Our system is implemented using OpenCV library. Testing system is run on Intel processor Core 2 Duo, 2 Ghz, 2 GB memory. The video sequence is analyzed with image resolutions 320x240 pixels at 30 fps. The 12 Thai alphabet are chosen as command symbols, selecting from the 12 groups of 44 thai alphabet [10] considering from its similarities, as shown in table 2.

Table2. The 12 groups of Thai alphabet

Group 1	ก กฏ กฏ กฏ กฏ	Group 7	ฉ ฉ ฉ ฉ ฉ
Group 2	ข ข ข ข ข	Group 8	ช ช ช ช ช
Group 3	ค ค ค ค ค	Group 9	ซ ซ ซ ซ ซ
Group 4	ด ด ด ด ด	Group 10	ฌ ฌ ฌ ฌ ฌ
Group 5	ต ต ต ต ต	Group 11	ฎ ฎ ฎ ฎ ฎ
Group 6	ถ ถ ถ ถ ถ	Group 12	ฏ ฏ ฏ ฏ ฏ



Figure 10. Samples of hand trajectory of the 12 Thai alphabet.

A. Thai Alphabet Recognition

In the evaluation, HMM topology is a fully connected with 3 types of states. About the number of state we take in our consideration the number of segment parts that are contained in possible gesture. The main direction part of all gesture in configuration consists of 8 directions (up, down, left, right, up-right, up-left, down-right and down-left). Intuitively, we use 8 hidden states with 2 auxiliary states as initialization and output. However, the number of HMM cannot defined precisely by that consideration, it need to be tested with variation. Therefore, we also evaluate the HMM with 5 and 15 states for comparison with 10 states.

For each alphabet, we use 50 observation sequences for training and 100 observation sequences for testing. The table3 and figure 11 show the result, we can notice that the HMM with 10 states improvement significantly the result comparing to HMM with 5 state (9%). Although the HMM with 15 states give in global better result than HMM with 10 states but the improvement of correction rate is very insignificant (1%). In conclusion, the HMM with 10 states is good enough for our Thai alphabet recognition system. The demonstration of our system in real-time show in the video at

<http://www.youtube.com/watch?v=jEzi02EuweY>

Table3. Recognition rate for the 12 Thai alphabet.

Gesture model	Correct recognition		
	5 state	10 state	15 state
ก	96	82	86
ข	99	90	87
ค	92	95	86
ด	95	98	96
อ	24	65	68
โ	21	69	59
ฃ	96	93	97
ฅ	77	75	75
ง	58	70	72
จ	68	69	95
ฉ	76	95	98
ช	91	97	91

Gesture model	Correct recognition		
	5 state	10 state	15 state
ก	74.41	83.16	84.16
ข	27.40	12.81	12.79

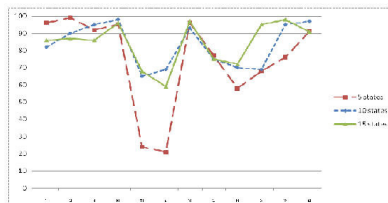


Figure 11. Recognition rate comparison for the 12 Thai alphabet.

VII. CONCLUSION

We have introduced the recognition system for the 12 Thai alphabets using hand trajectory features with Hidden Markov Model. The appropriate feature parameter and HMM topology were applied in our system. The result shows that the system can recognize in overall about 84 % of recognition rate. For the best case ก alphabet. Additionally, more features may be introduced in order to improve the correction rate.

REFERENCES

- [1] Elmezain, M.; Al-Hamadi, "Gesture Recognition for Alphabets from Hand Motion Trajectory Using Hidden Markov Models," Signal Processing and Information Technology, 2007 IEEE International Symposium on , vol., no., pp.1192-1197, 15-18 Dec. 2007
- [2] J. Wachs, H. Stern, Y. Edan, M. Gillam, C. Feied, M. Smith, and J. Handler, "A Real-Time Hand Gesture System Based on Evolutionary Search," Vision, vol. 22, no. 3, Oct. 2006, Dearborn, Mich, Society of Manufacturing Engineers.
- [3] Qing Chen; Georganas, N.D.; Petriu, E.M.; , "Real-time Vision-based Hand Gesture Recognition Using Haar-like Features," Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE , vol., no., pp.1-6, 1-3 May 2007
- [4] Viola, P. and Jones M., "Rapid object detection using a boosted cascade of simple features," Proceedings of the IEEE CVPR, vol.1, no., pp. I-511- I-518 vol.1, 2001
- [5] Lienhart R. and Maydt J., "An extended set of Haar-like features for rapid object detection," International Conference on Image Processing, vol.1, no., pp. I-900- I-903 vol.1, 2002
- [6] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," Intel Technology Journal, Q2 1998.
- [7] Greg Welch and Gary Bishop, "An Introduction to the Kalman Filter," Annual Conference on Computer Graphics & Interactive Techniques. ACM Press, Addison-Wesley, Los Angeles, CA, USA (August 12-17), SIGGRAPH 2001 course pack edition
- [8] Ramer-Douglas-Peucker algorithm, july,2010, Available at (online): http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorithm
- [9] L.R. Rabiner, "A tutorial on hidden markov models and selected application in speech recognition," Proc. IEEE 77, 1989, pp 267-293
- [10] Thai script, october, 2010, Available at (online): http://en.wikipedia.org/wiki/Thai_script

ผู้เขียน

ชื่อ สกุล	นายกฤตศิลป์ ศิลานนท์		
รหัสประจำตัวนักศึกษา	5210120055		
วุฒิการศึกษา			
	วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
	วิศวกรรมศาสตรบัณฑิต (วิศวกรรมคอมพิวเตอร์)	มหาวิทยาลัยสงขลานครินทร์	2551

การตีพิมพ์เผยแพร่ผลงาน

1. Kittasil Silanon and Nikom Suvonvorn, "Real Time Hand Tracking as a User Input Device," In *Proceedings of the Fifth International Conference on Knowledge, Information and Creativity Support Systems*, November 25-27, 2010, Chiang Mai, Thailand, pp. 153 - 158.
2. Kittasil Silanon, Nikom Suvonvorn and Montri Karnchanadecha, "Thai Alphabet Recognition from Hand Motion Trajectory Using HMM," In *2011 3rd International Conference on Signal Acquisition and Processing*, Singapore, February 26-28, 2011, pp. V1-189 – V1-193.