

การประมาณค่าทางของศีรษะด้วยการจับคู่การเคลื่อนไหวของจุดสำคัญบนใบหน้า
สำหรับระบบโต้ตอบ

นายณวัฒน์ อรรถานุกูล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2555
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)
are the thesis authors' files submitted through the Graduate School.

Head Pose Estimation Using Facial Feature Motion Matching for Interactive Systems

Mr. Nawarat Auttanugune

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2012

Copyright of Chulalongkorn University

นวนรัตน์ อรรถานุกูล : การประมาณท่าทางของศีรษะด้วยการจับคู่การเคลื่อนไหวของจุดสำคัญบนใบหน้าสำหรับระบบโต้ตอบ. (HEAD POSE ESTIMATION USING FACIAL FEATURE MOTION MATCHING FOR INTERACTIVE SYSTEMS)
 อ. ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ. ดร.ธนาวรัตน์ ชลิดาพงศ์, 62 หน้า.

งานวิจัยนี้นำเสนอขั้นตอนวิธีสำหรับการประมาณท่าทางของศีรษะจากภาพวิดีโอที่ศรัณ 2 มิติให้เป็นพิกัด 3 มิติ ในมุม ก้ม, เงย และหมุนหัน ด้วยการจับคู่การเคลื่อนไหวระหว่างเวกเตอร์การเคลื่อนที่จากส่วนภาพใบหน้าของผู้ใช้งานโปรแกรมและตัวแบบศีรษะ 3 มิติที่องคิต โดยสามารถทำงานได้อย่างอัตโนมัติในระบบทันกาล ใช้เพียงกล้องรับภาพธรรมดา และไม่ต้ออาศัยการกำหนดค่าเริ่มต้นใดๆของโปรแกรมก่อนการใช้งาน

ขั้นตอนวิธีที่นำเสนอประกอบไปด้วย 3 ขั้นตอนหลัก ประกอบด้วย ขั้นตอนเตรียมการ ขั้นตอนประมาณท่าทางของศีรษะ และขั้นตอนการแสดงผล ซึ่งขั้นตอนเตรียมการทำหน้าที่ค้นหาส่วนภาพใบหน้าของผู้ใช้งานและค้นหาตัวทรงของสัญญาณรบกวนของกล้องรับภาพโดยอัตโนมัติ จากนั้นในขั้นตอนของการประมาณท่าทางของศีรษะ จะอาศัยการหมุนของตัวแบบที่องคิตไปในทิศทางต่างๆ แล้วฉายให้อยู่บนพิกัด 2 มิติ เพื่อสร้างเวกเตอร์การเคลื่อนที่แล้วนำไปเปรียบเทียบกับเวกเตอร์การเคลื่อนที่ของส่วนภาพใบหน้า โดยขั้นตอนนี้จะทำซ้ำจนกระทั่งได้เวกเตอร์การเคลื่อนที่ที่มีความใกล้เคียงกับเวกเตอร์การเคลื่อนที่จากส่วนภาพใบหน้าของผู้ใช้งานมากที่สุด จากนั้น จึงแสดงผลท่าทางของศีรษะที่ประมาณได้ออกจากจอภาพ

ผลการทดลองพบว่าขั้นตอนวิธีที่นำเสนอนี้สามารถทำงานในระบบทันกาลได้ โดยมีค่าความคลาดเคลื่อนต่ำกว่า 5 องศาบนแต่ละแกนหมุนของศีรษะ นอกจากนี้ยังได้นำไปประยุกต์ใช้กับการสังเคราะห์ภาพเคลื่อนไหว 3 มิติด้วย

ภาควิชา.....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต.....
 สาขาวิชา....วิทยาศาสตร์คอมพิวเตอร์... ลายมือชื่ออ.ที่ปรึกษาวิทยานิพนธ์หลัก.....
 ปีการศึกษา....2555.....

5371420221 : MAJOR COMPUTER SCIENCE

KEYWORDS : HEAD POSE ESTIMATION / MOTION MATCHING / FACE DETECTION
/ GPGPU

NAWARAT AUTTANUGUNE : HEAD POSE ESTIMATION USING FACIAL
FEATURE MOTION MATCHING FOR INTERACTIVE SYSTEMS . ADVISOR :
ASST.PROF. THANARAT H. CHALIDABHONGSE, Ph.D., 62 pp.

This research proposes an algorithm for head pose estimation. The algorithm takes a 2D image sequence and returns 3D angles of yaw, pitch and roll of each input frame. The algorithm is based on motion vector matching between motions extracted from face image and motions computed by rotating the CANDIDE 3D head model. The implemented system is suitable for real time processing via general camera devices such as a web camera. The system works automatically so any prior configuration and setup are not required.

The proposed algorithm is composed of 3 main parts, initialization, head pose estimation, and result display. The initialization is for the face detection and for automatic camera noise calibration. The head pose estimation phase begins with the computing of the optical flow of a facial image. Later, the process rotates the CANDIDE model into a predicted direction and projects the model into 2D to generate optical flows. These two sources of optical flow are compared and repeated until the best match is found.

The experiment results show that the system is suitable for real-time interactive processing. The error of head pose estimation is lower than 5 degrees on each rotation angle of the head. The proposed algorithm has also been applied to a 3D character animation.

Department : Computer Engineering Student's Signature

Field of Study : Computer Science Advisor's Signature

Academic Year : 2012

กิตติกรรมประกาศ

ความสำเร็จสมบูรณ์ของวิทยานิพนธ์ฉบับนี้เกิดขึ้นได้ด้วยคำชี้แนะแนวทางการศึกษาและวิจัยอย่างทุ่มเทของ ผศ.ดร.ธนรัตน์ ชลิตาพงศ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ จึงขอขอบพระคุณอย่างสูงไว้ ณ ที่นี้

ขอขอบคุณคณาจารย์ เจ้าหน้าที่ภาควิชาฯ เจ้าหน้าที่ห้องสมุด ตลอดจนเพื่อนๆ ทั้งหลักสูตรในเวลาและภาคนอกเวลาราชการ ที่คอยสนับสนุนการทำงาน ให้ความเห็นและกำลังใจในการทำวิจัย รวมทั้ง ขอขอบพระคุณ บิดา มารดา ผู้ให้การศึกษาระดับปริญญาตรีและเลี้ยงดูเป็นอย่างดีเสมอมา

สารบัญ

| | หน้า |
|--|------|
| บทคัดย่อภาษาไทย..... | ง |
| บทคัดย่อภาษาอังกฤษ..... | จ |
| กิตติกรรมประกาศ..... | ฉ |
| สารบัญ..... | ช |
| สารบัญรูป..... | ญ |
| บทที่ | |
| 1. บทนำ..... | 1 |
| 1.1 ความเป็นมาและความสำคัญของปัญหา..... | 1 |
| 1.2 วัตถุประสงค์ของการวิจัย..... | 3 |
| 1.3 ขอบเขตงานวิจัย..... | 4 |
| 1.4 วิธีการดำเนินงานวิจัย..... | 4 |
| 1.5 ประโยชน์ที่คาดว่าจะได้รับ..... | 5 |
| 1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์..... | 5 |
| 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง..... | 6 |
| 2.1 ทฤษฎีที่เกี่ยวข้อง..... | 6 |
| 2.1.1 ตัวแบบกิ่งกอดิต (CANDIDE)..... | 6 |
| 2.1.2 ออปติคัลโฟลว์ (Optical flows)..... | 7 |
| 2.1.2.1 ความสว่างของจุดภาพระหว่างเฟรมคงที่ (Brightness Constancy)..... | 8 |
| 2.1.2.2 มีการเปลี่ยนแปลงของภาพในแต่ละเฟรมเพียงเล็กน้อย (Temporal Persistence)..... | 8 |
| 2.1.2.3 มีความสัมพันธ์เชิงพื้นที่ระหว่างเฟรม (Spatial Coherence)..... | 9 |
| 2.1.3 Haar – like Feature..... | 10 |
| 2.1.4 การค้นหาแบบต้นทุนต่ำสุด (Least Cost Search)..... | 11 |
| 2.2 งานวิจัยที่เกี่ยวข้อง..... | 12 |
| 2.2.1 Head Pose Estimation Survey..... | 12 |
| 2.2.1.1 Appearance Template Methods..... | 12 |

| บทที่ | หน้า |
|--|------|
| 2.2.1.2 Detector Array Methods..... | 13 |
| 2.2.1.3 Nonlinear Regression Methods..... | 14 |
| 2.2.1.4 Manifold Embedding Methods..... | 15 |
| 2.2.1.5 Flexible Model Methods..... | 16 |
| 2.2.1.6 Geometric Methods..... | 17 |
| 2.2.1.7 Tracking Methods..... | 18 |
| 2.2.1.8 Hybrid Methods..... | 19 |
| 2.2.2 Estimating 3D Facial Pose in Video with Just Three Points..... | 21 |
| 2.2.2.1 การค้นหาตำแหน่งของส่วนภาพใบหน้าจากภาพอินพุต..... | 21 |
| 2.2.2.2 การค้นหาจุดสำคัญของส่วนภาพใบหน้า..... | 22 |
| 2.2.2.3 การติดตามใบหน้าและจุดสำคัญ..... | 23 |
| 2.2.2.4 การประมาณท่าทางของศีรษะ..... | 23 |
| 2.2.3 Motion regularization for model-based head tracking..... | 27 |
| 2.2.3.1 ตัวแบบศีรษะ 3 มิติ..... | 28 |
| 2.2.3.2 ตัวแบบการเคลื่อนที่..... | 28 |
| 2.2.3.3 การปรับขนาดของตัวแบบศีรษะ 3 มิติ..... | 29 |
| 2.2.3.4 การฉายตัวแบบศีรษะ 3 มิติ ลงบนภาพ 2 มิติ..... | 29 |
| 2.2.3.5 การเคลื่อนที่ของตัวแบบ (Model Flow)..... | 29 |
| 2.2.3.6 การเปรียบเทียบการเคลื่อนที่..... | 30 |
| 2.2.3.7 การเลือกสถานะที่เป็นคำตอบ..... | 30 |
| 3 ขั้นตอนวิธีที่เสนอ..... | 32 |
| 3.1 ขั้นเตรียมการ..... | 33 |
| 3.1.1 ค้นหาส่วนภาพใบหน้า..... | 33 |
| 3.1.2 ค้นหาจุดสำคัญบนใบหน้าด้วย Active Shape Model (ASM)..... | 33 |
| 3.1.3 การปรับค่าความคลาดเคลื่อน..... | 35 |
| 3.1.4 ปรับขนาดตัวแบบก็องดิด..... | 39 |
| 3.2 การประมาณท่าทางของศีรษะ..... | 41 |
| 3.2.1 การติดตามจุดสำคัญบนใบหน้า..... | 41 |
| 3.2.2 การคำนวณการเคลื่อนที่ของศีรษะ..... | 41 |

| บทที่ | หน้า |
|--|------|
| 3.2.2.1 สถานะของปัญหา (State of a Problem)..... | 42 |
| 3.2.2.2 ฟังก์ชันต้นทุน (Cost Function)..... | 44 |
| 3.2.2.3 การค้นหาสถานะคำตอบแบบขนาน..... | 45 |
| 3.2.3 การคำนวณการเคลื่อนที่ส่วนประกอบของใบหน้า..... | 48 |
| 3.3 การแสดงผล..... | 49 |
| 4 การทดลองและผลการทดลอง..... | 51 |
| 4.1 การทดสอบความแม่นยำของขั้นตอนวิธี..... | 51 |
| 4.2 การทดสอบประสิทธิภาพการปรับค่าความคลาดเคลื่อน..... | 52 |
| 4.2 การทดสอบประสิทธิภาพการประมวลผลด้วย GPU เทียบกับ CPU..... | 55 |
| 4.3 การแสดงผลเป็นภาพแอนิเมชัน 3 มิติ..... | 57 |
| 5 สรุปผลการวิจัยและข้อเสนอแนะ..... | 58 |
| 5.1 สรุปผลการวิจัย..... | 58 |
| 5.2 ข้อเสนอแนะ..... | 59 |
| รายการอ้างอิง..... | 60 |
| ประวัติผู้เขียนวิทยานิพนธ์..... | 62 |

สารบัญรูป

| รูปที่ | | หน้า |
|--------|--|------|
| 1-1 | การประมาณท่าทางของศีรษะกับการใช้งานด้านความปลอดภัย โดย Erik และคณะ..... | 1 |
| 1-2 | ตัวอย่างการประยุกต์ใช้งานด้านความบันเทิง..... | 2 |
| 1-3 | ศีรษะในท่าทางต่างๆบนพิกัด 3 มิติสามารถทำให้เกิดเส้นรอบขอบศีรษะบนพิกัดสองมิติที่เหมือนกันได้..... | 3 |
| 2-1 | วิวัฒนาการของตัวแบบกึ่งอติตทั้ง 3 รุ่น I II และ III..... | 7 |
| 2-2 | สมมุติฐานความสว่างของจุดภาพคงที่..... | 8 |
| 2-3 | การประมาณฟังก์ชันการเคลื่อนที่..... | 9 |
| 2-4 | สมมุติฐานความสัมพันธ์เชิงพื้นที่ระหว่างเฟรม..... | 10 |
| 2-5 | Haar - Like Features..... | 10 |
| 2-6 | การตรวจจับใบหน้าด้วย Haar-Like Feature..... | 11 |
| 2-7 | Appearance Template Methods..... | 12 |
| 2-8 | Detector Array Methods..... | 13 |
| 2-9 | Nonlinear Regression Methods..... | 14 |
| 2-10 | Manifold Embedding Methods..... | 15 |
| 2-11 | Flexible Model Methods..... | 16 |
| 2-12 | Geometric Methods..... | 17 |
| 2-13 | Tracking Methods..... | 18 |
| 2-14 | Hybrid Methods..... | 19 |
| 2-15 | การคัดกรองตัวเลือกพื้นที่ของใบหน้า..... | 21 |
| 2-16 | การทำโพรไฟล์เพื่อตรวจสอบใบหน้า..... | 22 |
| 2-17 | การค้นหาค่าประกอบของใบหน้า..... | 23 |
| 2-18 | ตัวแบบการเกิดภาพ..... | 24 |
| 2-19 | แกนเอกของวงรีบนส่วนภาพใบหน้าเป็นค่าคงที่..... | 25 |
| 2-20 | ตัวแบบศีรษะที่มีลักษณะคล้ายกับแขนของหุ่นยนต์..... | 26 |
| 2-21 | การทดลองการประมาณท่าทางของศีรษะ..... | 27 |
| 2-22 | ผลการทดลองของ Sumit Basu และคณะ..... | 31 |

| รูปที่ | | หน้า |
|--------|---|------|
| 3-1 | ภาพรวมระเบียบวิธีขั้นตอนที่นำเสนอ..... | 32 |
| 3-2 | การค้นหาส่วนภาพใบหน้า..... | 33 |
| 3-3 | การปรับตัวแบบที่สร้างไว้อย่างวนซ้ำเพื่อให้เข้ากับเส้นขอบของวัตถุบนภาพ อินพุต..... | 34 |
| 3-4 | ตัวแบบใบหน้า ASM ที่นำมาประยุกต์ใช้ในวิทยานิพนธ์ฉบับนี้..... | 34 |
| 3-5 | คู่จุดสำคัญบนใบหน้าทั้ง 19 ตำแหน่งระหว่างตัวแบบ ASM และตัวแบบสี่ระยะกั องติด..... | 35 |
| 3-6 | กราฟของฟังก์ชันซิกมอยด์ ณ ค่า t ต่างๆ..... | 37 |
| 3-7 | ขนาดของวัตถุกับขนาดของเวกเตอร์การเคลื่อนที่..... | 39 |
| 3-8 | คู่จุดที่ใช้ปรับขนาดตัวแบบก็องติดและส่วนภาพใบหน้าของผู้ใช้งาน..... | 40 |
| 3-9 | โครงสร้าง T-Shape..... | 41 |
| 3-10 | โครงสร้างต้นไม้การค้นหา..... | 42 |
| 3-11 | การสร้างสถานะจาก $S_i \rightarrow S_j$ | 44 |
| 3-12 | เส้นทางที่นำไปสู่สถานะคำตอบที่ดีที่สุดอาจจะอยู่นอกเส้นทางการค้นหาที่มี ต้นทุนต่ำสุดเสมอ..... | 46 |
| 3-13 | การค้นหาสถานะคำตอบที่ดีที่สุดแบบขนาน..... | 47 |
| 3-14 | การค้นหาสถานะคำตอบด้วยฟังก์ชันต้นทุนภายในแต่ละปริภูมิสถานะย่อย..... | 48 |
| 3-15 | ตัวอย่างการแสดงผลของโปรแกรม..... | 49 |
| 3-16 | หน้าจอแสดงสถานะการทำงานของโปรแกรม..... | 50 |
| 4-1 | ผลการทดสอบความแม่นยำของขั้นตอนวิธี..... | 52 |
| 4-2 | ความแปรปรวนของสัญญาณกล้องรับภาพทำให้ตรวจจับการเคลื่อนที่ที่ แปรปรวนตามไปด้วย..... | 53 |
| 4-3 | การสะสมค่าผิดพลาดที่เกิดจากความแปรปรวนของสัญญาณกล้องรับภาพ | 53 |
| 4-4 | ภาพของโปรแกรมขณะที่โปรแกรมกำลังค้นหาพารามิเตอร์เพื่อปรับความ คลาดเคลื่อนของสัญญาณภาพ..... | 54 |
| 4-5 | ภาพของโปรแกรมเมื่อปรับความคลาดเคลื่อนของสัญญาณภาพแล้ว..... | 54 |
| 4-6 | ผลการทดลองใช้งานโปรแกรม..... | 54 |
| 4-7 | ผลการประมาณท่าทางศีรษะเปรียบเทียบในมุมหัน (Yaw)..... | 56 |
| 4-8 | ผลการประมาณท่าทางศีรษะเปรียบเทียบในมุมก้ม (Pitch)..... | 56 |

| รูปที่ | | หน้า |
|--------|---|------|
| 4-9 | ผลการประมาณท่าทางศีรษะเปรียบเทียบในมุมเงย (Roll)..... | 56 |
| 4-10 | ผลการทดลองร่วมกับโปรแกรมสร้างภาพแอนิเมชัน 3 มิติ..... | 57 |
| 5-1 | การตรวจจับส่วนภาพใบหน้าอัตโนมัติ..... | 58 |
| 5-2 | การเริ่มต้นตรวจจับใบหน้าใหม่อัตโนมัติเมื่อการติดตามใบหน้ามีข้อผิดพลาด.... | 58 |
| 5-3 | การหมุนศีรษะไปในทิศทางต่างๆอย่างต่อเนื่อง..... | 59 |

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ท่าทางของศีรษะสามารถสื่อสารบอกความหมายได้ เช่น การพยักหน้า แสดงถึงการรับรู้ แทนคำว่าใช่ หรือการหันหน้าไปในทางใดทางหนึ่งเพื่อบอกทิศทางที่กำลังกล่าวถึง เป็นต้น ซึ่งท่าทางเหล่านี้เป็นสากล ไม่ขึ้นกับภาษาพูด ดังนั้น ถ้าเราสามารถคิดค้นขั้นตอนวิธีที่ช่วยให้คอมพิวเตอร์สามารถรับรู้ท่าทางของศีรษะ จะสามารถนำไปใช้ในการพัฒนาระบบสำหรับการสื่อสารกับคอมพิวเตอร์ หรือใช้เพื่อควบคุมเครื่องจักรด้วยท่าทางของศีรษะได้ โดยไม่ขึ้นกับภาษาท้องถิ่นอย่างภาษาพูด หรือภาษาเขียน ซึ่งในปัจจุบันมีการประยุกต์ใช้งานการประมาณท่าทางของศีรษะอย่างหลากหลาย เช่น การนำไปประยุกต์ใช้งานกับภาษามือสำหรับผู้พิการทางการได้ยิน ตลอดจนการประยุกต์ใช้ในด้านความปลอดภัยของการขับขี่ดังรูปที่ 1-1 และด้านความบันเทิงในรูปที่ 1-2 ตามลำดับ



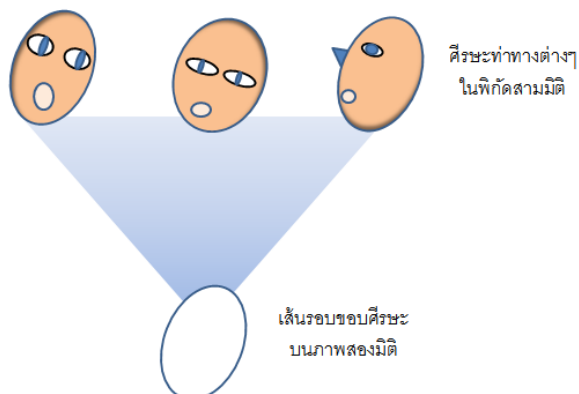
รูปที่ 1-1 การประมาณท่าทางของศีรษะกับการใช้งาน

ด้านความปลอดภัย โดย Erik และคณะ [1]



รูปที่ 1-2 ตัวอย่างการประยุกต์ใช้งานด้านความบันเทิง [2]

อย่างไรก็ตาม ถึงแม้ว่ามนุษย์สามารถรับรู้ความหมายจากท่าทางของศีรษะได้อย่างง่ายดาย แต่ไม่ใช่เรื่องง่ายสำหรับคอมพิวเตอร์ การตรวจจับและติดตามตำแหน่งของศีรษะโดยใช้วิธีการติดตามเส้นขอบ (Edge) ระหว่างวัตถุและพื้นหลัง หรือการติดตามจากเส้นรอบขอบวัตถุ (Contour) นั้น ใช้ได้เฉพาะการเคลื่อนที่ในระนาบ 2 มิติเท่านั้น ไม่สามารถนำมาใช้กับการติดตามการเคลื่อนที่ของศีรษะแบบอิสระใน 3 มิติ ที่ศีรษะมนุษย์สามารถหมุนได้ในสามแกน ได้แก่ การก้มเงย การหันหน้า และการเอียงศีรษะได้ ดังนั้น เส้นรอบขอบของศีรษะในภาพสองมิติจะแปรเปลี่ยนรูปร่างอยู่ตลอดเวลา เมื่อมีการขยับหรือเคลื่อนที่ในแกนหมุนใดๆ เนื่องด้วย เมื่อถ่ายภาพด้วยกล้องวีดิทัศน์ ข้อมูล 3 มิติ ได้แก่ ความกว้าง ความยาว และ ความลึก จะหายไปเหลือเพียงข้อมูล 2 มิติ ที่มีเพียงความกว้าง และ ความยาวเท่านั้น ซึ่งสามารถทำให้เกิดปัญหาการผิดรูป (Ill-Posed Problem) ของเส้นรอบขอบวัตถุ กล่าวคือ การหมุนศีรษะไปตามแกนหมุนทั้งสามด้วยท่าทางที่แตกต่างกัน แต่สามารถก่อให้เกิดเส้นรอบขอบของศีรษะบนภาพสองมิติที่มีความคล้ายคลึงกันได้ ดังแสดงในรูปที่ 1-3 ดังนั้น เราจึงไม่สามารถแยกแยะท่าทางของศีรษะที่เกิดขึ้นใน 3 มิติ โดยอาศัยเพียงข้อมูลเส้นรอบขอบที่ปรากฏบนภาพ 2 มิติ



รูปที่ 1-3 ศีรษะในท่าทางต่างๆบนพิกัด 3 มิติสามารถทำให้เกิดเส้นรอบขอบศีรษะบนพิกัดสองมิติที่เหมือนกันได้

นอกจากนี้ จุดสำคัญต่างๆที่ปรากฏบนใบหน้า (Facial Features) ได้แก่ คิ้ว ตา และปาก ยังสามารถเปลี่ยนรูปร่างไปได้ ตามอารมณ์ความรู้สึกของผู้พูด เช่น รูปร่างของคิ้ว ตา และปาก ในอารมณ์โกรธ ย่อมแตกต่างออกไปจากอารมณ์ดีใจ เป็นต้น ดังนั้น ทั้งข้อมูลเส้นรอบขอบและข้อมูลเชิงพื้นที่ของส่วนภาพใบหน้าที่ปรากฏบนภาพสองมิติแทบจะไม่สามารถนำไปใช้ในการประมาณท่าทางของศีรษะและจุดสำคัญต่างๆที่ปรากฏบนใบหน้าโดยตรงได้เลย

ในช่วงหลายปีที่ผ่านมา มีการคิดค้นพัฒนาขั้นตอนวิธีสำหรับการประมาณท่าทางของศีรษะจำนวนมาก โดยแต่ละขั้นตอนวิธีมีข้อดีและข้อเสียที่แตกต่างกัน ดังจะได้กล่าวต่อไปในบทที่ 2 บททบทวนวรรณกรรม ซึ่งในโครงร่างวิทยานิพนธ์ฉบับนี้ได้นำเสนอขั้นตอนวิธีที่ใช้การวิเคราะห์ข้อมูลการเคลื่อนที่ของศีรษะที่ปรากฏบนภาพ 2 มิติ โดยไม่ต้องอาศัยอุปกรณ์พิเศษ หรือ การทำการสอน (Training) ใดๆ กล่าวคือ ขั้นตอนวิธีที่นำเสนออาศัยตัวแบบของใบหน้า 3 มิติ ที่มีชื่อว่า กิ่งดิด (CANDIDE) ทำการเลียนแบบการเคลื่อนที่ที่เกิดขึ้นในภาพอินพุตสองมิติ โดยทำการหมุนตัวแบบศีรษะไปในทิศทางต่างๆ แล้วเปรียบเทียบค่าความคลาดเคลื่อนที่เกิดขึ้นจากการคำนวณผลรวมระยะทางระหว่างทุกๆเวกเตอร์การเคลื่อนที่จากภาพและตัวแบบกิ่งดิด

1.2 วัตถุประสงค์ของการวิจัย

- เพื่อพัฒนาขั้นตอนวิธีประมาณท่าทางศีรษะ โดยการประมาณมุมการหมุนทั้ง 3 แกน ได้แก่ ก้มเงย หันหน้า และเอียง เพื่อใช้ในโปรแกรมประยุกต์แบบทันที (Real-Time)
- เพื่อพัฒนาขั้นตอนวิธีตรวจจับคุณลักษณะเด่นบนใบหน้า (Facial Feature Extraction) บางอย่าง ได้แก่ คิ้วทั้งสองข้าง ดวงตาทั้งสองข้าง และปาก

1.3 ขอบเขตงานวิจัย

ขอบเขตของงานวิจัยในวิทยานิพนธ์ฉบับนี้ คือคิดค้นขั้นตอนวิธีสำหรับการค้นหามุมก้มเงย มุมหันหน้า และมุมเอียงของศีรษะจากภาพวีดิทัศน์ 2 มิติด้วยกล้องรับภาพชนิดพื้นฐาน โดยผลลัพธ์ของมุมที่ประมาณได้ มีหน่วยเป็นองศาของแต่ละแกนหมุนในปริภูมิ 3 มิติแยกจากกัน รวมไปถึงการตรวจจับลักษณะองค์ประกอบของใบหน้า (Facial Feature) เช่น การตรวจจับการเคลื่อนไหวที่ของคิ้วทั้งสองข้าง, ดวงตาและปาก แต่ไม่รวมถึงการตีความอารมณ์จากสีหน้า เช่น การบ่งบอกว่าเป็นอารมณ์โกรธ, ดีใจ, เสียใจ เป็นต้น รวมไปถึงข้อกำหนดต่างๆ ดังนี้

- วีดิทัศน์อินพุตมีภาพของบุคคลเดียว
- สภาพแสงที่เพียงพอและถ่ายภายในอาคาร
- ขนาดของภาพอินพุตมีความละเอียดไม่ต่ำกว่า 640*480 พิกเซล
- สามารถทำงานได้ในเวลาจริง อย่างน้อย 25 เฟรมต่อวินาที
- สามารถรองรับกรณีภาพใบหน้าถูกบังบางส่วน (Partial Occlusion) ในบางขณะ

1.4 วิธีการดำเนินงานวิจัย

การดำเนินการวิจัยและทดลอง เริ่มจากการบันทึกภาพวีดิทัศน์สำหรับเป็นอินพุตของขั้นตอนวิธี โดยการบันทึกจะทำในสภาพแวดล้อมปิด เพื่อควบคุมปัจจัยที่ส่งผลต่อภาพของวีดิทัศน์ เช่น ความเข้มของแสง ระยะห่างระหว่างกล้องและผู้แสดงท่าทาง เป็นต้น โดยการพัฒนาในส่วนของขั้นตอนวิธีการสามารถแบ่งออกได้เป็น 2 ขั้นตอน ประกอบด้วย ขั้นตอนการตรวจสอบความแม่นยำของขั้นตอนวิธี และการนำไปใช้งานกับภาพวีดิทัศน์ที่ได้เตรียมนั้น โดยในขั้นตอนการตรวจสอบความแม่นยำของขั้นตอนวิธีนั้น ทำขึ้นเพื่อชี้วัดประสิทธิภาพของขั้นตอนวิธี โดยการทดลองจะจำลองการหมุนศีรษะในพิภัก 3 มิติในคอมพิวเตอร์ โดยกำหนดมุมที่จะหมุนในแต่ละแกนไว้ล่วงหน้า จากนั้นจึงใช้ขั้นตอนวิธีที่คิดค้นขึ้น ทำการหามุมการหมุนในแต่ละแกน แล้วเปรียบเทียบผลลัพธ์ที่ได้กับมุมที่กำหนดไว้ล่วงหน้า

เมื่อผลการทดสอบความแม่นยำของขั้นตอนวิธีมีค่าความคลาดเคลื่อนต่ำกว่าที่กำหนด คือ มีผลการประมาณท่าทางของศีรษะที่ได้คลาดเคลื่อนน้อยกว่า 5 องศา ตามแนวการพัฒนาขั้นตอนวิธีเรื่องความแม่นยำ (Accuracy) ที่ E. Murphy-Chutorian และ M.M. Trivedi แนะนำ ดังที่ได้แสดงไว้ในหน้าที่ 19 จากนั้น การดำเนินงานในขั้นถัดไป จะนำขั้นตอนวิธีการมาประยุกต์ใช้กับภาพวีดิทัศน์ที่เตรียมไว้ในแล้ววิเคราะห์ผลลัพธ์ทั้งในแง่ของประสิทธิภาพของขั้นตอนวิธีการ และประสิทธิภาพในด้านความซับซ้อนเชิงเวลา (Time Complexity) ที่กำหนดไว้ให้เทียบเท่าหรือต่ำกว่า $\log(n)$ เพื่อให้สามารถนำไปใช้ในสภาพแวดล้อมทันกาล (Real-Time) ได้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- ได้ขั้นตอนวิธีตรวจสอบการแสดงสีหน้าและท่าทางของศีรษะที่ทำงานได้อย่างอัตโนมัติ และมีความแม่นยำ โดยสามารถประมวลผลได้แบบทันที
- สามารถนำไปใช้สังเคราะห์ภาพการเคลื่อนไหวที่ใบหน้า 3 มิติ ได้
- สามารถนำไปใช้เพื่อสกัดข้อมูลการเคลื่อนไหวของสีหน้าเพื่อพัฒนาขั้นตอนวิธีการรู้จำภาษามือ

1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

- “Head Pose Estimation using Motion Matching”, in Proc. 15th International Workshop on Advanced Image Technology (IWAIT), Ho Chi Minh City, Vietnam, 2012
- “Head Pose Estimation Using Motion Subspace Matching on GPU”, in Proc. Asia Pacific Signal and Information Processing Association (APSIPA), Hollywood, California, USA 2012

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

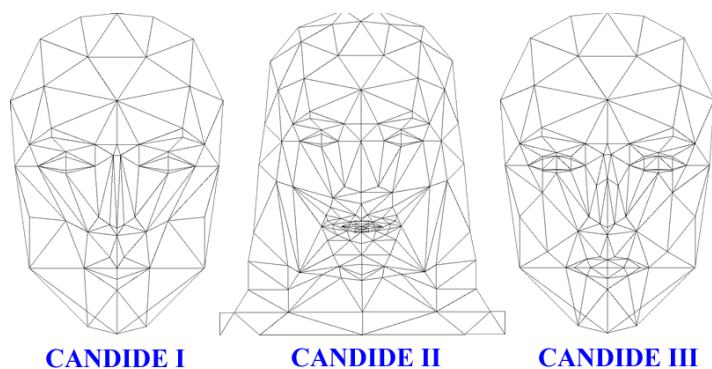
การประมาณท่าทางของศีรษะแบบ 3 มิตินั้น ไม่สามารถทำได้ด้วยการพิจารณาจากเส้นรอบขอบ หรือ ข้อมูลเชิงพื้นที่โดยตรงจากภาพ 2 มิติดังที่ได้กล่าวไว้ในหัวข้อก่อนหน้า แนวคิดที่นำเสนอในวิทยานิพนธ์ฉบับนี้จึงคิดค้นขั้นตอนวิธีสำหรับการประมาณท่าทางของศีรษะด้วยเวกเตอร์การเคลื่อนที่ โดยหลักการทำงานเริ่มจาก ระบบทำการคำนวณเวกเตอร์การเคลื่อนที่จากภาพอินพุต 2 มิติ จากนั้น จึงใช้ตัวแบบกราฟฟิกรูปศีรษะ 3 มิติ ทำการหมุนไปในแกนหมุนทั้ง 3 แกน คือ การก้มเงย การหันหน้า และการเอียงศีรษะ จากนั้นระบบจะคำนวณเวกเตอร์การเคลื่อนที่ที่ได้จากการฉาย (Perspective Projection) เพื่อเปรียบเทียบกับเวกเตอร์การเคลื่อนที่ที่เกิดขึ้นในภาพอินพุต 2 มิติ ซึ่งวิธีการนี้สามารถทำงานได้โดยอัตโนมัติ ไม่ต้องใช้ข้อมูลฝึกสอนใดๆ และสามารถนำไปใช้ประมวลผลแบบทันที (Real Time) ได้ ซึ่งทฤษฎีที่เกี่ยวข้องกับขั้นตอนวิธีการประมาณท่าทางของศีรษะ ด้วยการเปรียบเทียบเวกเตอร์ของการเคลื่อนที่นี้ ประกอบไปด้วย ตัวแบบของศีรษะ (กึ่งอดิต), การอ่านการเคลื่อนที่ของวัตถุด้วยออฟติคอลลโฟลว์, Haar – like Feature, การเปรียบเทียบการเคลื่อนที่ของวัตถุ และการค้นหาในปริภูมิสถานะการหมุนของศีรษะที่ใช้กำหนดทิศการหมุนของตัวแบบกึ่งอดิตและประมาณท่าทางของศีรษะจากภาพอินพุตวีดิทัศน์

2.1.1 ตัวแบบกึ่งอดิต (CANDIDE)

ตัวแบบกึ่งอดิต เป็นตัวแบบ 3 มิติของศีรษะ พัฒนาโดย M. Rydfalk [3] เพื่อใช้สำหรับการทำภาพเคลื่อนไหว (Animation) ซึ่งประกอบไปด้วยจุดยอด (Vertex) ทั้งหมด 79 จุด ซึ่งแบ่งออกเป็น 2 กลุ่ม คือ จุดยอดท้องถิ่น (Local vertex) และ จุดยอดครอบคลุม (Global vertex) โดย จุดยอดท้องถิ่นใช้สำหรับควบคุมการเคลื่อนที่องค์ประกอบของใบหน้า เช่น ควบคุมการเคลื่อนที่ของ คิ้ว ตา จมูก และปาก เป็นต้น ในขณะที่จุดยอดครอบคลุมใช้สำหรับควบคุมการหมุนของศีรษะทั้ง 3 แกน

ตัวแบบกึ่งอดิตที่นิยมใช้ในปัจจุบันมี 3 รุ่น โดยแต่ละรุ่นถูกพัฒนาขึ้นด้วยวัตถุประสงค์การใช้งานที่แตกต่างกัน ดังแสดงในรูปที่ 2-1 ซึ่งตัวแบบกึ่งอดิตรุ่นแรก ประกอบด้วยจุดยอด (Vertex) จำนวน 79 จุด ใช้สำหรับงานการสร้างภาพเคลื่อนไหวของใบหน้าทั่วไป และตัว

แบบ กิ่งดิดในรุ่นที่ 2 ได้เพิ่มรายละเอียดเกี่ยวกับลำคอไปจนถึงไหล่และฟันโดยการเพิ่มจำนวนจุดยอดเป็น 160 จุด ซึ่งเหมาะสำหรับการสร้างภาพเคลื่อนไหวที่มีรายละเอียดของปาก ลำคอ และฟัน ส่วนตัวแบบกิ่งดิดในรุ่นที่ 3 ไม่ได้พัฒนาต่อจากตัวแบบรุ่นที่ 2 แต่พัฒนามาจากตัวแบบรุ่นที่ 1 โดยมีวัตถุประสงค์เพื่อประยุกต์ใช้กับการแสดงภาพด้วยชุดพารามิเตอร์เกี่ยวกับใบหน้า MPEG-4 “MPEG-4 Facial Animation Parameters” ซึ่งได้เพิ่มจุดยอดเข้าไปอีก 20 จุดอย่างไรก็ตาม การทดลองในขั้นต้นของวิทยานิพนธ์ ต้องการหาการเคลื่อนที่ของศีรษะซึ่งไม่ครอบคลุมถึงรายละเอียดของฟันและ MPEG-4 จึงใช้เพียงตัวแบบกิ่งดิดเวอร์ชันแรกเท่านั้น



รูปที่ 2-1 วิวัฒนาการของตัวแบบกิ่งดิดทั้ง 3 รุ่น I II และ III

2.1.2 ออปติคัลโฟลว์ (Optical flows)

ออปติคัลโฟลว์เป็นวิธีการคำนวณหาเวกเตอร์การเคลื่อนที่ของวัตถุในภาพที่กล้องรับภาพตั้งอยู่กับที่ (Fixed Camera) โดยใช้ภาพอินพุตตั้งแต่ 2 เฟรมขึ้นไป ซึ่งขั้นตอนวิธีที่ได้รับการยอมรับและนำมาประยุกต์ใช้ในงานวิจัยนี้คือ ขั้นตอนวิธีของลูคาส-คานาด (Lucas-Kanade) ซึ่งพัฒนาขั้นตอนวิธีโดยตั้งอยู่บนสมมติฐาน 3 ประการ คือ

- ความสว่างของจุดภาพระหว่างเฟรมคงที่ (Brightness Constancy)
- มีการเปลี่ยนแปลงของภาพในแต่ละเฟรมเพียงเล็กน้อย (Temporal Persistence)
- มีความสัมพันธ์เชิงพื้นที่ระหว่างเฟรม (Spatial Coherence)

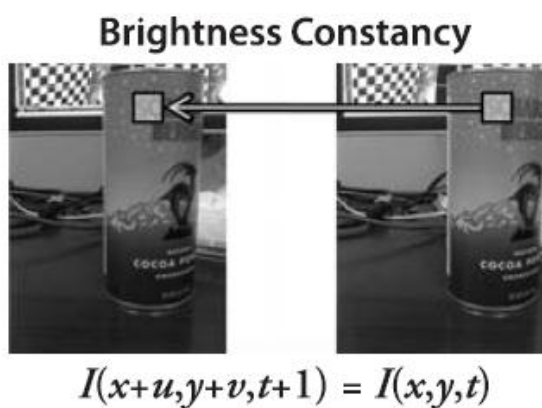
2.1.2.1 ความสว่างของจุดภาพระหว่างเฟรมคงที่ (Brightness Constancy)

กำหนดให้ $f(x,t)$ เป็นค่าความสว่างของจุด x ณ เฟรมที่ t ซึ่งมีค่าคงที่ ถึงแม้จะเคลื่อนไปยังตำแหน่งอื่นในเฟรมที่ $t + dt$ ดังนั้น จึงเขียนเป็นรูปสมการได้ว่า

$$f(x,t) \equiv I(x(t),t) = I(x(t+dt),t+dt) \quad (2.1)$$

ดังนั้น ค่าของพิกเซลใดๆที่สนใจจะไม่เปลี่ยนแปลงไปในแต่ละเฟรมดังสมการ

$$\frac{\partial f(x)}{\partial t} = 0 \quad (2.2)$$



รูปที่ 2-2 สมมติฐานความสว่างของจุดภาพคงที่

2.1.2.2 มีการเปลี่ยนแปลงของภาพในแต่ละเฟรมเพียงเล็กน้อย (Temporal Persistence)

การเคลื่อนที่ หรือการเปลี่ยนแปลงของพิกเซลที่สนใจจากเฟรมที่ t ไปยังเฟรมที่ $t + dt$ จะต้องเปลี่ยนแปลงไปที่เล็กน้อย โดยจากสมมติฐานในข้อแรก (Brightness Constancy) เราเขียนแทนสมการความสัมพันธ์ในรูปแบบของอนุพันธ์เพียงย่อย (Partial Differential Equation) ที่อาศัยกฎลูกโซ่ (Chain rule) ได้ว่า

$$\frac{\partial I}{\partial x} \left(\frac{\partial x}{\partial t} \right) + \frac{\partial I}{\partial t} \Big|_{x(t)} = 0 \quad (2.3)$$

เมื่อ กำหนดให้

$$I_x = \left. \frac{\partial I}{\partial x} \right|_t$$

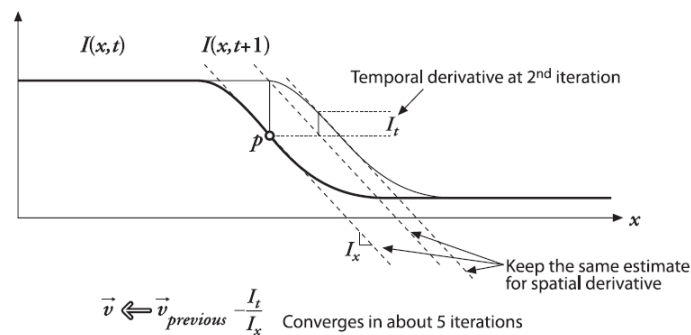
$$v = \left(\frac{\partial x}{\partial t} \right)$$

$$I_t = \left. \frac{\partial I}{\partial t} \right|_{x(t)}$$

ดังนั้น ความเร็ว (Velocity) ของจุดภาพใดๆที่สนใจสามารถคำนวณจาก

$$v = -\frac{I_t}{I_x} \quad (2.4)$$

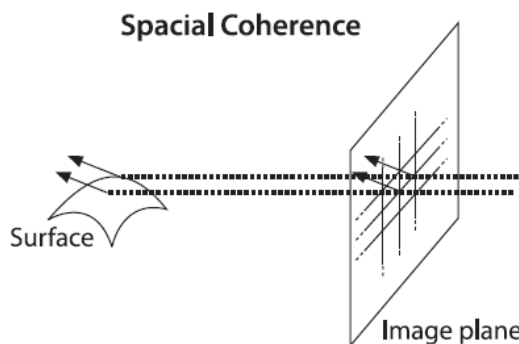
โดยผลลัพธ์ที่ได้ เป็นค่าประมาณการ ซึ่งเราสามารถเพิ่มความแม่นยำของการประมาณการด้วยวิธีการคำนวณซ้ำของนิวตัน (Newton's method) ดังแสดงในรูปที่ 2-3



รูปที่ 2-3 การประมาณฟังก์ชันการเคลื่อนที่

2.1.2.3 มีความสัมพันธ์เชิงพื้นที่ระหว่างเฟรม (Spatial Coherence)

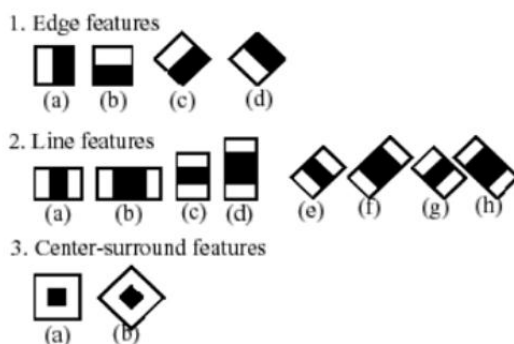
ข้อกำหนดนี้ คล้ายๆกับข้อกำหนดใน 2.1.2.1 กล่าวคือ พิกเซลรอบๆจุดพิกเซลที่สนใจ (neighbors pixel) ในเฟรมที่ t ยังคงปรากฏเป็นพิกเซลรอบๆจุดพิกเซลที่สนใจในเฟรมที่ $t + dt$ และมีค่าไม่เปลี่ยนแปลงไปจากเดิมมากนัก ดังแสดงในรูปที่ 2-4



รูปที่ 2-4 สมมุติฐานความสัมพันธ์เชิงพื้นที่ระหว่างเฟรม

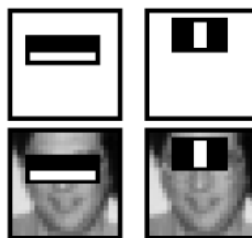
2.1.3 Haar – like Feature

Haar – like Feature เป็นขั้นตอนวิธีสำหรับการตรวจจับรูปแบบในภาพระดับสีเทา (Gray Image) โดยอาศัยชุดของโครงสร้างรูปสี่เหลี่ยมที่ได้จากการสอน (Training) ด้วยข้อมูลตัวอย่างของรูปแบบที่ต้องการตรวจค้นจำนวนมาก โดยภาพต้นแบบที่ใช้สอนจะต้องมีทั้งตัวอย่างสอนที่ถูกต้อง (Positive Data) และตัวอย่างลบ (Negative Data) เพื่อให้ทราบถึงลักษณะที่ใช่และไม่ใช่ของรูปแบบที่จะตรวจหาในลักษณะต่างๆ ซึ่งโครงสร้างรูปสี่เหลี่ยมที่ได้จากการสอนนี้เรียกว่า คุณสมบัติ (Feature) มีลักษณะเป็นรูปสี่เหลี่ยมที่มีพื้นที่สีขาวและสีดำสลับกันไปมาในแนวต่างๆ ดังแสดงใน รูปที่ 2-5 ที่แบ่งคุณสมบัติออกเป็นสามกลุ่ม คือ คุณสมบัติสำหรับการตรวจจับขอบ, คุณสมบัติสำหรับการตรวจจับเส้น และ คุณสมบัติสำหรับการหาบริเวณที่ล้อมรอบจุดศูนย์กลาง



รูปที่ 2-5 Haar - Like Features

ในงานวิทยานิพนธ์นี้ นำ Haar-Like Feature มาตรวจหาใบหน้าและจุดสำคัญบนส่วนภาพใบหน้า ได้แก่ ดวงตา จมูก และปาก กล่าวคือ ลักษณะของใบหน้าในภาพระดับสีเทา ช่วงด้านบนของใบหน้า ตั้งแต่หน้าผากถึงกึ่งกลางใบหน้าจะมีข้อมูลเชิงพื้นที่โดยรวมเป็นสีดำมากกว่าช่วงของใบหน้าด้านล่าง ดังนั้นจึงสามารถแสดงรูปแบบของใบหน้าด้วย Haar-Like Feature ดังในรูปที่ 2-6



รูปที่ 2-6 การตรวจจับใบหน้าด้วย Haar-Like Feature [4]

2.1.4 การค้นหาแบบต้นทุนต่ำสุด (Least Cost Search)

เมื่อพิจารณาการหมุนศีรษะของมนุษย์ในรูปแบบของปัญหาการค้นหา (Searching problem) จะพบว่าศีรษะมนุษย์สามารถหมุนได้ 3 แกน (3 Degrees of freedom) โดยแต่ละแกนสามารถหมุนได้ประมาณ 180 องศา ดังนั้น ขนาดของปริภูมิสถานะทั้งหมดคิดเป็น 180^3 ซึ่งมีขนาดใหญ่เกินกว่าจะค้นหาด้วยการค้นหาแบบเรียงลำดับ (Sequential search)

เพื่อให้การค้นหาสถานะมีประสิทธิภาพ จึงได้นำเอาการค้นหาแบบต้นทุนต่ำสุด (Least cost search) มาประยุกต์ใช้ โดยต้นทุนของการค้นหาในที่นี้ คำนวณจากผลรวมค่าความคลาดเคลื่อนของเวกเตอร์การเคลื่อนที่ ที่เกิดขึ้นจากการหมุนแบบจำลองศีรษะไปในทิศทางนั้นๆ เมื่อเทียบกับเวกเตอร์การเคลื่อนที่ที่คำนวณได้จากภาพอินพุต

เมื่อกำหนดให้ ภาพวิดีโอที่ค้นอินพุตที่ความเร็ว 30 เฟรมต่อวินาที และการหมุนศีรษะของมนุษย์ในแต่ละแกนสามารถทำได้ไม่เกิน 180 องศาในหนึ่งวินาที ดังนั้น ในแต่ละเฟรม จะมีการหมุนของศีรษะในแต่ละแกนไม่เกิน $180/30 = 6$ องศา แต่การหมุนสามารถเป็นได้ทั้งการเพิ่มหรือลดองศาการหมุน ดังนั้นพิสัย (Range) ของการหมุนต่อเฟรม คือ $[-6, 6]$ องศา โดย

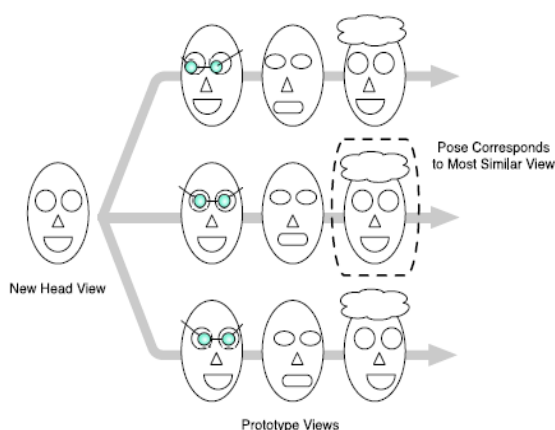
การสร้างสถานะการค้นหาในแต่ละรุ่นจะเริ่มจากพิสัยที่กว้างที่สุดก่อน แล้วจึงขยับเข้าไปในพิสัยที่แคบลงจนกระทั่งเข้าใกล้ศูนย์ โดยในแต่ละลำดับขั้นของการค้นหา จะมีพิสัยที่แคบลงครึ่งหนึ่ง เช่น จาก $[-6, 6] \rightarrow [-3, 3] \rightarrow [-1.5, 1.5] \rightarrow \dots \rightarrow [0, 0]$ เป็นต้น ซึ่งในแต่ละขั้นของสถานะการค้นหา จะเลือกสถานะที่มีต้นทุนหรือค่าผิดพลาดต่ำที่สุดเป็นสถานะแม่สำหรับการแตกกิ่งออกไป ดังนั้น ความซับซ้อนเชิงเวลาของการค้นหาจะลดลงจาก $O(n^3)$ เป็น $\log_2(n)$

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 Head Pose Estimation Survey

E. Murphy-Chutorian และ M.M. Trivedi [5] ได้สำรวจงานวิจัยเกี่ยวกับการประมาณท่าทางของศีรษะจำนวนทั้งสิ้น 151 บทความที่ตีพิมพ์ตั้งแต่ปี 1994 ถึง 2008 โดยมีวัตถุประสงค์เพื่อจัดกลุ่มขั้นตอนวิธีที่มีอยู่ทั้งหมด และวิเคราะห์ให้เห็นถึงข้อดีและข้อเสียของแต่ละกลุ่มขั้นตอนวิธี โดย E. Murphy Chutorian และ M.M. Trivedi ได้จัดกลุ่มขั้นตอนวิธีการของการประมาณท่าทางของศีรษะทั้งหมดออกเป็น 8 กลุ่ม ดังนี้

2.2.1.1 Appearance Template Methods



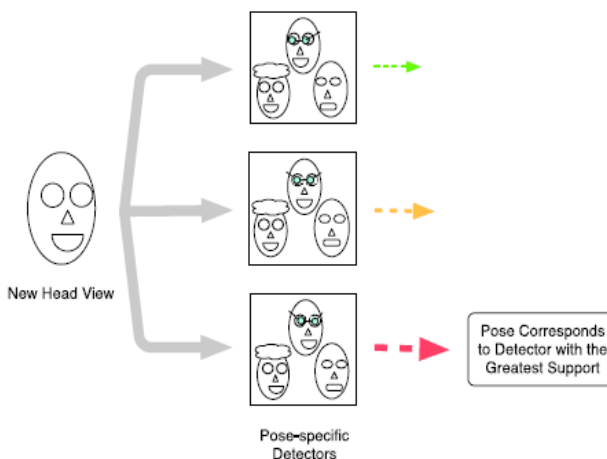
รูปที่ 2-7 Appearance Template Methods

เป็นกลุ่มของขั้นตอนวิธีที่ทำการประมวลผลโดยตรงจากข้อมูลพิกเซล โดยนำข้อมูลเชิงพื้นที่ของส่วนภาพใบหน้าในท่าทางต่างๆมาสร้างตัวแบบ (Template) ของ

ท่าทางศีรษะแต่ละแบบ ซึ่งการนำไปใช้ทำได้โดยการเปรียบเทียบระหว่างตัวแบบที่มีกับ ภาพอินพุตจากนั้นจึงเลือกตัวแบบที่มีลักษณะความคล้ายคลึงมากที่สุดเป็นคำตอบ

| ข้อดี | ข้อเสีย |
|---|--|
| <ul style="list-style-type: none"> • สามารถเพิ่มจำนวนตัวแบบ (Template) ได้ง่าย • สามารถปรับเปลี่ยนวิธีการรู้จำตัวแบบได้ง่าย • ไม่ต้องอาศัยการเรียนรู้จากตัวอย่างลบ (Negative data) • ไม่ต้องทำการสกัดจุดสำคัญของใบหน้า เช่น ลักษณะของตา, คิ้ว, ปาก • รองรับภาพอินพุตได้ทั้งความละเอียดต่ำและความละเอียดสูง | <ul style="list-style-type: none"> • ให้ผลลัพธ์เป็นการประมาณท่าทางของศีรษะในรูปแบบจำกัดเท่าที่ตัวแบบกำหนดไว้ (Discrete pose) จึงมีความแม่นยำต่ำ • ระบบยังคงประมวลผลและให้ผลลัพธ์ที่คลาดเคลื่อนเมื่อใช้ภาพอินพุตที่ไม่สมบูรณ์ • ในการประมาณท่าทางของศีรษะแต่ละครั้ง ต้องคำนวณจากตัวแบบ (Template) ทุกตัวแบบที่มี ดังนั้น หากมีจำนวนตัวแบบมากจะต้องใช้เวลาในการประมวลผลมากตามไปด้วย |

2.2.1.2 Detector Array Methods



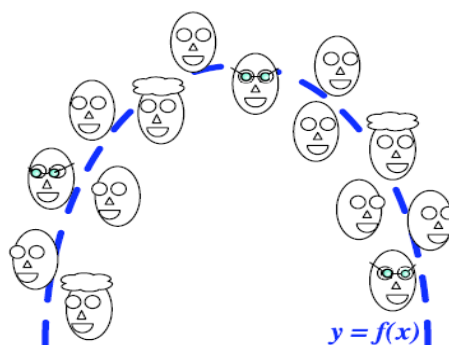
รูปที่ 2-8 Detector Array Methods

คือกลุ่มขั้นตอนวิธีที่นำชุดข้อมูลสอนของใบหน้าในมุมต่างๆมาสร้างเป็นอาเรย์ สำหรับการตรวจหา (Detector Array) โดยสมาชิกในอาเรย์ตรวจหาแต่ละตัวจะจดจำรูปแบบของท่าทางศีรษะเพียงท่าทางเดียว ซึ่งการสร้างอาเรย์ตรวจหานี้จะสร้างด้วยวิธีการเรียนรู้แบบมีผู้สอน (Supervised learning) เพื่อให้ระบบสามารถจดจำรูปแบบของข้อมูลสอน เช่น สร้างด้วยวิธีการเรียนรู้แบบเบย์เซียน (Bayesian learning) เป็นต้น

และ เมื่อต้องการนำไปใช้ประมาณค่าทางของศึระะ โปรแกรมจะต้องประมวลผลสมาชิกทุกตัวในอาเรย์ตรวจหา จากนั้นจึงเลือกสมาชิกตัวที่ได้คะแนนสูงสุดเป็นคำตอบ

| ข้อดี | ข้อเสีย |
|---|---|
| <ul style="list-style-type: none"> ● ไม่ต้องอาศัยขั้นตอนการค้นหาส่วนภาพของใบหน้าและองค์ประกอบของใบหน้า เช่น ตา, ปาก, จมูก เป็นต้น ● สามารถบ่งบอกได้ว่าอินพุตเป็นภาพของศึระะที่สมบูรณ์หรือไม่โดยอาศัยการวิเคราะห์ค่าขีดแบ่งจากอาเรย์ตรวจหาที่เรียนรู้มา ● สามารถกำหนดขั้นตอนวิธีการได้มาของแต่ละส่วนย่อยอาเรย์ตรวจหาได้เอง โดยแต่ละส่วนไม่จำเป็นต้องขึ้นต่อกัน ● รองรับภาพอินพุตได้ทั้งความละเอียดต่ำและความละเอียดสูง | <ul style="list-style-type: none"> ● ต้องอาศัยการเรียนรู้ด้วยข้อมูลภาพอินพุตของศึระะที่ไม่ต่อเนื่อง (Discrete pose) จำนวนมาก ● ต้องอาศัยการเรียนรู้จากตัวอย่างลบ (Negative data) ● อาจเกิดปัญหาเมื่อตัวอย่างเรียนรู้ที่ถูกต้อง (Positive data) และตัวอย่างลบ (Negative data) มีความใกล้เคียงกันมาก ● โดยมาก งานวิจัยในกลุ่มนี้ คุณสมบัติที่มี จะไม่เกิน 12 คอลัมน์ เนื่องจากจำนวนคอลัมน์ที่ขึ้นนี้ กลับทำให้ความสามารถในการแยกแยะความแตกต่างของแต่ละท่าทางมีประสิทธิภาพลดลง ● มีองศาแห่งความอิสระ (Degree of freedom) ของท่าทางศึระะที่รับได้ค่อนข้างจำกัด |

2.2.1.3 Nonlinear Regression Methods

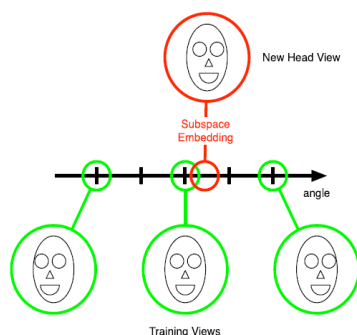


รูปที่ 2-9 Nonlinear Regression Methods

เป็นกลุ่มของขั้นตอนวิธีการประมาณท่าทางของศีรษะ จากการเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning) ผ่านฟังก์ชันไม่เชิงเส้น (Non-linear function) เช่น การเรียนรู้ด้วยโครงข่ายประสาทเทียม (Neural network) ซึ่งการเรียนรู้จะอาศัยการฝึก (Training) ด้วยภาพต้นแบบที่รู้มุมของท่าทางศีรษะแบบไม่ต่อเนื่อง (Discrete Pose)

| ข้อดี | ข้อเสีย |
|--|--|
| <ul style="list-style-type: none"> • สามารถให้ผลลัพธ์เป็นท่าทางของศีรษะแบบต่อเนื่อง (Continuous pose) ได้ จึงมีความแม่นยำสูง • สามารถแยกผลลัพธ์การคำนวณของแต่ละองศาแห่งความอิสระ (Degree of freedom) ออกจากกัน ได้แก่ มุมเอียง (Roll), มุมก้ม (Pitch) และมุมหัน (Yaw) • รองรับภาพอินพุตได้ทั้งความละเอียดต่ำและความละเอียดสูง • ให้ผลลัพธ์ที่แม่นยำสูงเมื่อเทียบกับขั้นตอนวิธีในกลุ่มอื่นๆ | <ul style="list-style-type: none"> • ไม่สามารถชี้วัดได้ชัดเจนว่าการเรียนรู้ด้วยสมการไม่เชิงเส้นแบบใด เป็นการคำนวณที่เหมาะสมกับลักษณะของข้อมูลการเรียนรู้ที่เตรียมไว้ • เมื่อนำไปใช้กับภาพวิดีโอที่คนอาจเกิดปัญหาการสะสมของข้อผิดพลาดที่เกิดขึ้นจากแต่ละเฟรมได้ |

2.2.1.4 Manifold Embedding Methods



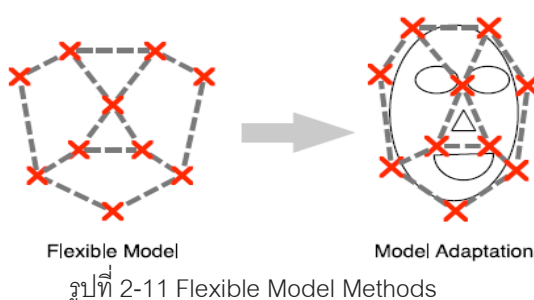
รูปที่ 2-10 Manifold Embedding Methods

เป็นกลุ่มของขั้นตอนวิธีการลดมิติการคำนวณจากภาพตัวอย่างท่าทางของศีรษะต่างๆจำนวนมากมาสร้างเป็นฟังก์ชันเชิงเส้นแบบต่อเนื่องที่มีขนาดของมิติการคำนวณ

น้อยลง เช่น วิธีการลดมิติการวิเคราะห์องค์ประกอบหลัก (Principal component Analysis: PCA) เป็นต้น

| ข้อดี | ข้อเสีย |
|---|--|
| <ul style="list-style-type: none"> ใช้เวลาในการประมวลผลน้อยลงเมื่อเทียบกับวิธีการในกลุ่มอื่น ๆ ที่มีจำนวนข้อมูลใช้ฝึกเท่ากัน | <ul style="list-style-type: none"> อาจเกิดข้อผิดพลาดจากการคำนวณสูง เมื่อภาพอินพุตมีความละเอียดต่ำหรือมีสิ่งรบกวน (Noise) การเตรียมภาพตัวอย่างทำทางศีรษะที่จะใช้สอนให้ครอบคลุมกับรูปแบบของศีรษะผู้ใช้ที่มีความหลากหลายเป็นเรื่องยาก |

2.2.1.5 Flexible Model Methods

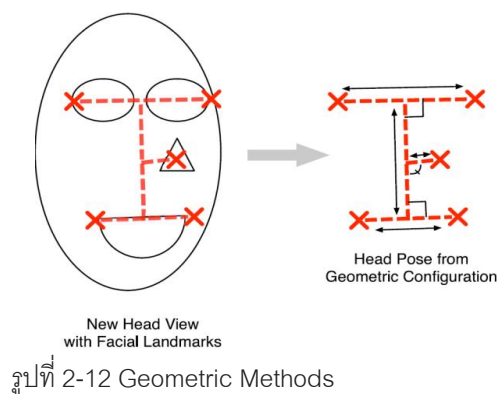


เป็นกลุ่มของขั้นตอนวิธีที่นำกลุ่มของจุดสำคัญต่างๆจากส่วนภาพบนใบหน้า มาสร้างเป็นกราฟยืดหยุ่น (Elastic Graph) ใช้สำหรับเป็นตัวแบบการประมาณท่าทางของศีรษะ โดยการใช้งานจะหาจุดสำคัญต่างๆตามที่กราฟยืดหยุ่นกำหนดไว้จากส่วนภาพใบหน้า แล้วคำนวณท่าทางของศีรษะจากโครงสร้างของกราฟที่ตรวจพบ

| ข้อดี | ข้อเสีย |
|--|---|
| <ul style="list-style-type: none"> กำหนดตำแหน่งบนใบหน้าที่จะใช้เป็นคุณสมบัติชัดเจน จึงมีความแม่นยำในการประมาณท่าทางของศีรษะกว่าขั้นตอนวิธีที่ใช้ข้อมูลเชิงพื้นที่ของใบหน้าเพียงอย่างเดียว มีวิธีที่สามารถนำมาประยุกต์ใช้เพื่อระบุตำแหน่งสำคัญบนใบหน้าที่มีประสิทธิภาพหลากหลายวิธี เช่น ใช้ร่วมกับ Active | <ul style="list-style-type: none"> ใช้เวลาในการคำนวณ (Time complexity) สูงจึงไม่เหมาะกับการประมวลผลภาพวิดีโอแบบทันที (Real time) ในกรณีที่ใช้กับภาพวิดีโอจำเป็นต้องค้นหาตำแหน่งของจุดสำคัญบนใบหน้า เช่น ตา, จมูก, ปาก ให้ได้อย่างแม่นยำเพื่อใช้เป็นข้อมูลอินพุต |

| ข้อดี | ข้อเสีย |
|--|--|
| Appearance Model (AAM) เพื่อหาตำแหน่งบนใบหน้าจากกลดลายของพื้นผิว (Texture) | <ul style="list-style-type: none"> ยังไม่ปรากฏผลการทดลองที่ดีกับภาพศีรษะระยะไกลหรือภาพอินพุตที่มีความละเอียดต่ำ |

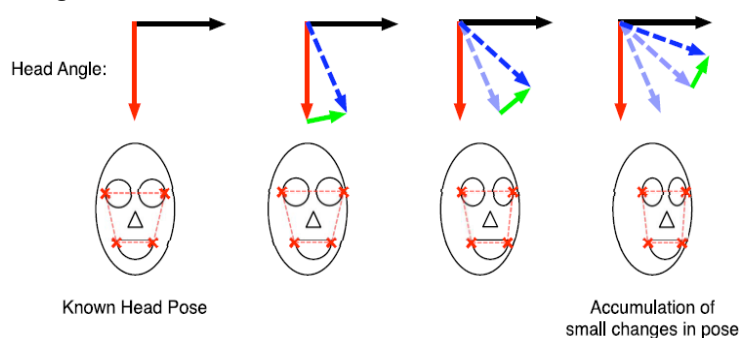
2.2.1.6 Geometric Methods



เป็นกลุ่มของขั้นตอนวิธีที่ประมาณท่าทางของศีรษะด้วยการคำนวณจากตำแหน่งของจุดสำคัญที่ปรากฏบนส่วนภาพใบหน้าแล้วนำมาคำนวณกับค่าโครงแบบ (Configuration) ที่กำหนดไว้ เช่น โครงแบบทางเรขาคณิต, โครงแบบที่สร้างจากการวัดสัดส่วนของใบหน้า (Facial Anthropometry) เป็นต้น

| ข้อดี | ข้อเสีย |
|--|---|
| <ul style="list-style-type: none"> สามารถนำตัวแบบเรขาคณิตต่างๆมาประยุกต์ใช้นอกเหนือจากการคำนวณด้วยข้อมูลพิกเซลของส่วนภาพใบหน้าโดยตรง ตัวแบบคณิตศาสตร์ด้านเรขาคณิตสามารถทำความเข้าใจได้ง่าย | <ul style="list-style-type: none"> การค้นหาตำแหน่งของจุดบนใบหน้าต้องมีความแม่นยำสูง ต้องสามารถค้นหาจุดสำคัญบนใบหน้าที่กำหนดได้ครบจึงจะสามารถประมวลผลได้ ต้องใช้กับภาพอินพุตที่มีความละเอียดสูง |

2.2.1.7 Tracking Methods

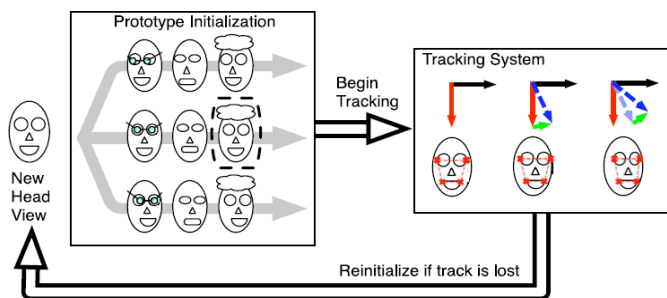


รูปที่ 2-13 Tracking Methods

เป็นกลุ่มของขั้นตอนวิธีที่ประมาณท่าทางของศีรษะจากการติดตามการเปลี่ยนแปลงที่เกิดขึ้นทีละน้อยจากภาพแต่ละเฟรม ซึ่งวิธีการในกลุ่มนี้จึงเหมาะกับการประมาณท่าทางของศีรษะจากภาพวิดีโอ โดยมียุทธศาสตร์ที่ว่า การเคลื่อนที่ของศีรษะในแต่ละเฟรมจะเปลี่ยนแปลงทีละน้อย

| ข้อดี | ข้อเสีย |
|---|---|
| <ul style="list-style-type: none"> ● ค้นหาท่าทางของศีรษะจากการเปลี่ยนแปลงทีละน้อยในทุกๆเฟรม ผลลัพธ์จึงมีความแม่นยำสูง ● ไม่มีข้อจำกัดด้านลักษณะรูปร่างของศีรษะที่มีรูปร่างแตกต่างกันไปจากแต่ละผู้ใช้งาน ● สามารถใช้งานเป็นระบบแบบทันที (Real time) ได้ | <ul style="list-style-type: none"> ● ต้องเริ่มต้นจากเฟรมที่รู้ท่าทางของศีรษะก่อนเสมอ เช่น ต้องเริ่มจากภาพวิดีโอที่ผู้ใช้หันหน้าตรง เป็นต้น ● ต้องเริ่มต้นการติดตามใหม่ (Reinitialize) เมื่อการติดตามส่วนภาพของใบหน้าสูญหาย (Tracking is lost) ● อาจเกิดปัญหาการสะสมของความผิดพลาดในการประมาณมุมของศีรษะที่เกิดขึ้นจากแต่ละเฟรม |

2.2.1.8 Hybrid Methods



รูปที่ 2-14 Hybrid Methods

เป็นกลุ่มขั้นตอนวิธีที่ผสมผสานวิธีการจากกลุ่มต่างๆเข้าด้วยกันเพื่อใช้ข้อได้เปรียบจากแต่ละกลุ่มขั้นตอนวิธี ซึ่งจะช่วยลดปัญหา ข้อเสียหรือข้อจำกัดของแต่ละกลุ่มวิธีการได้ เช่น การผสมผสานระหว่างขั้นตอนวิธีการของการประมาณท่าทางของศีรษะแบบไม่ต่อเนื่อง (Discrete Pose) กับการติดตาม (Tracking) ซึ่งช่วยให้ได้วิธีการประมาณท่าทางศีรษะที่มีความแม่นยำและรองรับภาพอินพุตที่มีความละเอียดต่ำได้เป็นต้น อย่างไรก็ตาม ในทางกลับกัน วิธีการผสมผสานนี้อาจจะทำให้เกิดการรวมผลเสียของแต่ละวิธีเข้าไว้ด้วยกันได้เช่นกัน

| ข้อดี | ข้อเสีย |
|--|---|
| <ul style="list-style-type: none"> ได้รับจุดเด่น, ข้อดีของแต่ละกลุ่มขั้นตอนวิธีที่นำมาประยุกต์ใช้ | <ul style="list-style-type: none"> เนื่องจากเป็นวิธีที่นำขั้นตอนวิธีการหลากหลายชนิดมารวมกัน จึงอาจรับเอาข้อเสียทั้งหมดมารวมกันด้วย เช่นกัน |

ในตอนท้ายขอบทความ E. Murphy-Chutorian และ M.M. Trivedi ได้เสนอความเห็นเกี่ยวกับแนวทางในการพัฒนาขั้นตอนวิธีการประมาณท่าทางศีรษะเพื่อที่สามารถนำไปประยุกต์ใช้งานได้ตรงกับความต้องการที่มีมากขึ้นในอนาคตไว้ ดังนี้

- **ความแม่นยำ (Accuracy)**

ต้องมีความผิดพลาดของการประมาณท่าทางศีรษะต่ำ คือ มีความคลาดเคลื่อนไม่เกิน 5 องศา

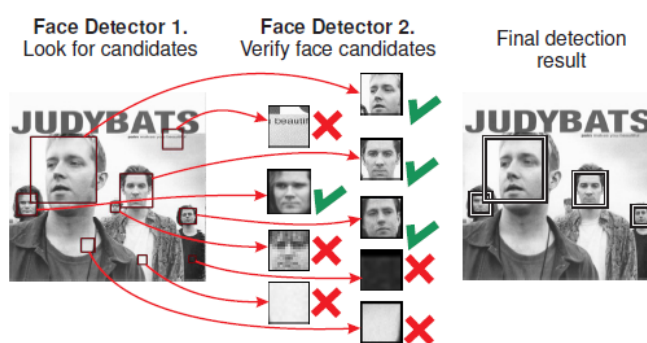
- **ใช้กล้องรับภาพเพียงตัวเดียว (Monocular)**
ระบบต้องสามารถทำงานได้ด้วยอุปกรณ์พื้นฐาน เช่น สามารถประมาณท่าทางศีรษะด้วยกล้องรับภาพ 2 มิติธรรมดา เป็นต้น
- **ทำงานโดยอัตโนมัติ (Autonomous)**
ระบบต้องสามารถทำงานได้อย่างอัตโนมัติโดยไม่ต้องอาศัยการควบคุมหรือกำหนดค่าใดๆจากผู้ใช้งาน
- **รองรับผู้ใช้งานได้พร้อมกันหลายๆคน (Multi-Person)**
ระบบควรมีความสามารถในการประมาณท่าทางศีรษะของผู้ใช้หลายๆคนที่ปรากฏอยู่ในภาพอินพุตเพียงภาพเดียว
- **ไม่ขึ้นต่อสภาพแสงและตัวบุคคล (Identity & Lighting Invariant)**
ระบบต้องสามารถทำงานได้ดีในสภาพแสงต่างๆ และไม่ขึ้นกับบุคคล
- **ใช้งานได้โดยไม่ขึ้นต่อความละเอียด (Resolution Independent)**
ระบบต้องสามารถทำงานได้ทั้งภาพโบนักรระยะใกล้ (Near Field) และระยะไกล (Far Field) ได้โดยไม่มีข้อจำกัดด้านความละเอียดของภาพอินพุต
- **ครอบคลุมทุกการเคลื่อนที่ของศีรษะ (Full Range of Head Motion)**
ระบบต้องสามารถประมาณท่าทางของศีรษะได้อย่างราบรื่นและต่อเนื่องโดยไม่จำกัดท่าทางของผลลัพธ์ (Discrete Pose)
- **ประมวลผลได้แบบทันที (Real-Time)**
ระบบสามารถประมวลผลกับภาพวีดิทัศน์ด้วยความเร็วสูง เช่น ใช้งานกับภาพวีดิทัศน์ที่ความเร็ว 30 ต่อวินาทีหรือสูงกว่าได้อย่างทันที เป็นต้น

2.2.2 Estimating 3D Facial Pose in Video with Just Three Points

G. Garica-Mateos และคณะ [6] ได้นำเสนอวิธีการประมาณท่าทางของศีรษะด้วยจุดสำคัญบนใบหน้าเพียง 3 ตำแหน่ง คือ จากดวงตาทั้งสองและปาก ซึ่งจัดเป็นขั้นตอนวิธีในกลุ่มวิธีเรขาคณิต (Geometric Methods) โดยมีวัตถุประสงค์ที่จะนำเสนอขั้นตอนวิธีที่ใช้ข้อมูลในการประมวลผลให้น้อยที่สุด และไม่จำเป็นต้องกำหนดค่าโครงแบบ (Configuration) ล่วงหน้าก่อนการใช้งาน ซึ่งขั้นตอนวิธีที่นำเสนอนี้ ประกอบไปด้วย 4 ขั้นตอนหลัก ได้แก่ การค้นหาตำแหน่งของใบหน้าจากภาพอินพุต, การค้นหาจุดสำคัญจากส่วนภาพใบหน้า, การติดตามจุดสำคัญ และขั้นตอนการประมาณท่าทางของศีรษะ

2.2.2.1 การค้นหาตำแหน่งของส่วนภาพใบหน้าจากภาพอินพุต

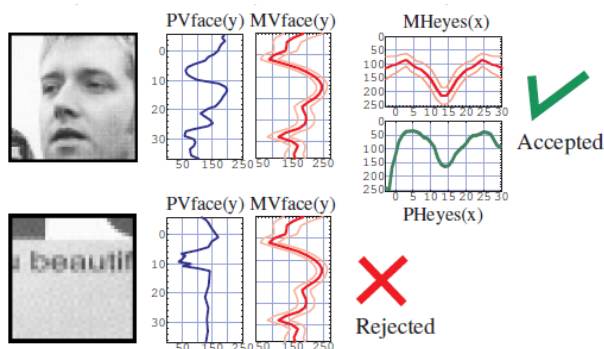
การค้นหาส่วนภาพใบหน้าแบ่งออกเป็น 2 ขั้นตอนย่อย โดยขั้นตอนแรกอาศัย Haar-like feature ในการประมาณตำแหน่งของบริเวณส่วนรูปที่มีความน่าจะเป็นใบหน้าจากภาพอินพุต โดยการประมาณในขั้นแรกนี้อาจตรวจพบบริเวณที่คาดว่าจะจะเป็นใบหน้าได้หลายตัวเลือก ซึ่งส่วนภาพของภาพอินพุตที่คาดว่าจะจะเป็นใบหน้าทั้งหมด จะถูกส่งผ่านไปยังขั้นตอนย่อยที่สองคือการคัดกรองตัวเลือกของส่วนภาพใบหน้าที่แสดงในรูปที่ 2-15



รูปที่ 2-15 การคัดกรองตัวเลือกพื้นที่ของใบหน้า

การคัดกรองตัวเลือกของใบหน้าใช้ขั้นตอนวิธีจากบทความชื่อเรื่อง “การประมวลผลใบหน้านามนุษย์ด้วยตัวแบบ 1.5D” ที่นำเสนอโดยผู้วิจัยคณะเดียวกันนี้ [7] เพื่อตรวจสอบส่วนของภาพอินพุตที่เป็นใบหน้าที่ถูกต้อง โดยหลักการทำงานจะตรวจสอบจากการคำนวณโปรไฟล์ความเข้มของพิกเซลทั้งในแนวนอนและแนวตั้ง (Projection profile) ดัง

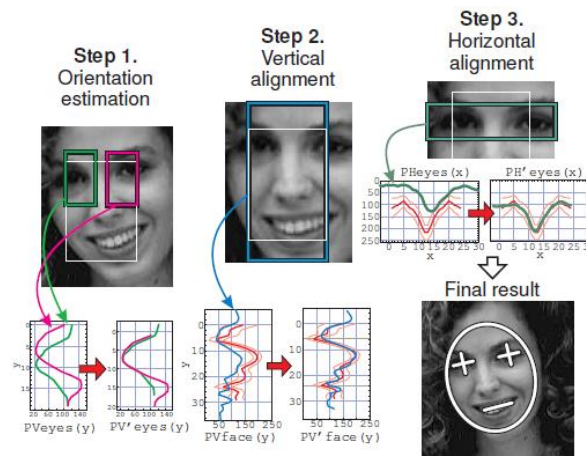
รูปที่ 2-16 แล้วนำไปเปรียบเทียบกับข้อมูลอ้างอิงที่มีการค้นหาตำแหน่งของใบหน้าตามวิธีดังกล่าวข้างต้นนี้ จะทำเฉพาะในเฟรมแรกของภาพวิดีโอ หรือเมื่อเกิดความผิดพลาดจากการติดตามตำแหน่งในแต่ละเฟรมแล้วต้องเริ่มต้นตรวจจับใหม่



รูปที่ 2-16 การทำโปรไฟล์เพื่อตรวจสอบใบหน้า

2.2.2.2 การค้นหาจุดสำคัญของส่วนภาพใบหน้า

การค้นหาจุดสำคัญของส่วนภาพใบหน้า ได้แก่ ดวงตาทั้งสองข้าง และปาก ของขั้นตอนวิธีที่น่าเสนอนี้อาศัย Haar-like feature สำหรับดวงตาและปาก โดยจะค้นหาจากพื้นที่ที่สนใจที่ตรวจพบว่าเป็นใบหน้าจากขั้นตอนที่แล้ว และเมื่อตรวจพบพื้นที่ที่คาดว่าจะ เป็นดวงตาและปากแล้วจะตรวจสอบความถูกต้องซ้ำอีกครั้งด้วยการคำนวณโปรไฟล์ ความเข้มของพิกเซลทั้งในแนวนอนและแนวตั้ง เช่นเดียวกับขั้นตอนวิธีในการค้นหา ตำแหน่งของใบหน้า ดังแสดงผลลัพธ์ที่ได้ในรูปที่ 2-17 โดยผลลัพธ์ที่ได้จะกำหนดให้เป็นพื้นที่ที่สนใจ (Region of Interest) สำหรับการประมวลผลในขั้นต่อไป



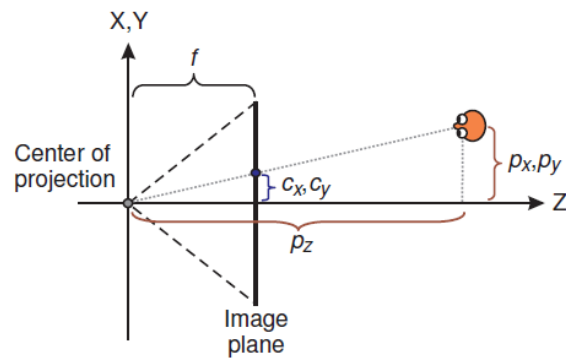
รูปที่ 2-17 การค้นหาองค์ประกอบของใบหน้า

2.2.2.3 การติดตามใบหน้าและจุดสำคัญ

การติดตามใบหน้าและจุดสำคัญใช้หลักการของการคำนวณปริพันธ์ความเข้มของพิกเซลภายในส่วนรูปที่สนใจ (Region of Interest) ทั้งในแนวนอนและแนวตั้งจากภาพอินพุตในแต่เฟรม ซึ่งได้นำเสนอโดย G. Garcia-Mateos [8] โดยผลการคำนวณปริพันธ์ที่ได้ จะถูกใช้เพื่อบอกตำแหน่งใหม่ของใบหน้าและจุดสำคัญของใบหน้าจากอินพุตเฟรมล่าสุด

2.2.2.4 การประมาณท่าทางของศีรษะ

การประมาณท่าทางของศีรษะอาศัยทฤษฎีของการเกิดภาพ 2 มิติ ซึ่งจุดกึ่งกลางของใบหน้าคำนวณจากจุดกึ่งกลางระหว่างดวงตาทั้งสองข้างและปาก โดยในระบบพิกัด 3 มิติ ตำแหน่งของใบหน้าแทนเขียนแทนด้วยสัญลักษณ์ $p = (p_x, p_y, p_z)$ ส่วนในระนาบ 2 มิติของภาพอินพุตเขียนแทนด้วย $C = (c_x, c_y)$ ดังแสดงในรูปที่ 2-18



รูปที่ 2-18 ตัวแบบการเกิดภาพ

กำหนดให้

- f = ระยะโฟกัส
 e_1 = ตำแหน่งของตาซ้ายบนภาพ
 e_2 = ตำแหน่งของตาขวาบนภาพ
 m = ตำแหน่งของปากบนภาพ

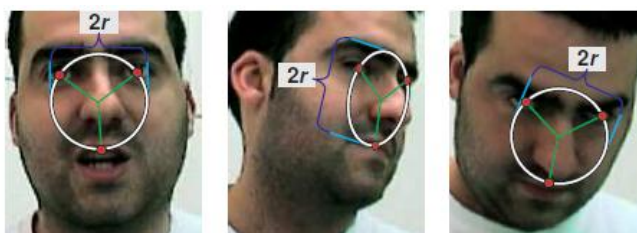
ดังนั้น ความสัมพันธ์ระหว่างตำแหน่งของใบหน้าในพิกัด 3 มิติและภาพอินพุต 2 มิติ คือ

$$p_x = \frac{e1_x + e2_x + m_x}{3} \times \frac{p_z}{f} \quad p_y = \frac{e1_y + e2_y + m_y}{3} \times \frac{p_z}{f} \quad (2.5)$$

ซึ่งจะพบว่า ขนาดของใบหน้าที่ปรากฏบนภาพ เป็นอัตราส่วนแปรผันตามระยะทางของ f ดังนั้น เมื่อกำหนดให้ขนาดของใบหน้าจริงในพิกัด 3 มิติ เขียนแทนด้วย t และขนาดของใบหน้าที่ปรากฏบนภาพอินพุต 2 มิติ เขียนแทนด้วย r จะได้ว่าความสัมพันธ์ว่า

$$p_z = f \frac{t}{r} \quad (2.6)$$

อย่างไรก็ตาม G. Garica-Mateos และคณะพบว่า r มีความสัมพันธ์กับ ระยะห่างระหว่างดวงตาทั้งสองข้างและปาก ดังเช่นรูปที่ 2-19 ระยะห่างระหว่างดวงตาทั้งสองข้างจะแปรเปลี่ยนไปตามแต่ละท่าทางของศีรษะ แต่แกนเอกของวงรีที่วาดผ่านจุดศูนย์กลางของดวงตาทั้งสองข้างและปากยังคงเป็น $2r$ เท่าเดิม



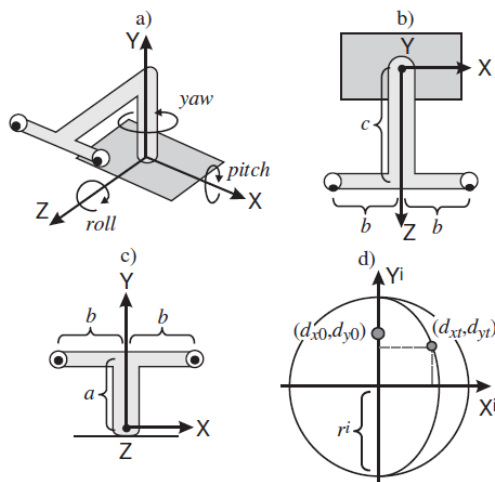
รูปที่ 2-19 แกนเอกของวงรีบนส่วนภาพใบหน้าเป็นค่าคงที่

ดังนั้น จึงสามารถสรุปเป็นเมทริกซ์ความแปรปรวน (Covariance Matrix) ระหว่างจุดศูนย์กลางของดวงตาและปากกับวงรีได้ ดังนี้

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} = \frac{1}{3} P^T \cdot P; P = \begin{bmatrix} e1_x - c_x & e1_y - c_y \\ e2_x - c_x & e2_y - c_y \\ m_x - c_x & m_y - c_y \end{bmatrix} \quad (2.7)$$

ค่าลักษณะเฉพาะ (Eigen Value) ของ Σ ตัวแรก จะสมนัยกับความยาวแกนเอกของวงรีดังนั้น r จึงสามารถคำนวณได้จาก $r = \sigma_x^2 + \sigma_y^2 + \sqrt{(\sigma_x^2 - \sigma_y^2)^2 + 4\sigma_{xy}^2}$ และเมื่อแทนค่า r นี้กลับไปยัง $p_z = f \frac{t}{r}$ จึงทำได้ข้อมูล 3 มิติของ (p_x, p_y, p_z) ครบทุกจำนวน ซึ่งทำให้สามารถคำนวณหาตำแหน่งของใบหน้าและจุดสำคัญในพิคัด 3 มิติ จากภาพ 2 มิติด้วยการแทนสมการข้างต้นเข้าด้วยกัน ทั้งนี้ งานวิจัยนี้กำหนดค่าคงที่ t เป็น 4 เซนติเมตร และมีการกำหนดคงที่ f ก่อนทำการทดลอง (แต่ไม่ปรากฏค่าที่ใช้ในบทความ) นอกจากนี้ ยังมีข้อกำหนดเพิ่มเติมเกี่ยวกับร่างกายที่จะต้องอยู่กับที่ (Fixed body assumption) และเมื่อนำศีรษะไปเปรียบเทียบกับแกนกลหมุนยนต์ที่มีองศาอิสระของการหมุน 3 แกน (3 degrees of freedom) ดังปรากฏในรูปที่ 2-20 โดยกำหนดให้ a

แทน ความสูงของดวงตาจากคอในแนวตั้งฉาก, b แทนระยะห่างครึ่งหนึ่งของดวงตา, c แทนระยะทางระหว่างกึ่งกลางของดวงตาถึงคอในแนวตั้งขนานกับพื้น และ (d_x, d_y) แทนค่าเฉลี่ยระยะห่างของดวงตาที่ปรากฏบนภาพ



รูปที่ 2-20 ตัวแบบศีระะที่มีลักษณะคล้ายกับแขนของหุ่นยนต์

จากลักษณะของแขนกลหุ่นยนต์ที่คล้ายคลึงกับแกนการหมุนของศีระะมนุษย์ G. Garica-Mateos และคณะจึงสรุปเป็นสูตรการคำนวณท่าทางของศีระะสำหรับมุมเอียง (roll), มุมก้ม (pitch) และมุมหัน (yaw) ตามลำดับ ดังนี้

$$\arctan\left(\frac{e2_y - e1_y}{e2_x - e1_x}\right) \tag{2.8}$$

$$\arcsin\left(\frac{d_{yt} - \left(d_{y0} - a \cdot \frac{f}{p_z}\right)}{r^i}\right) - \arcsin(a/c) \tag{2.9}$$

$$\arcsin\left(\frac{d_{xt} - d_{x0}}{r^i \cdot \cos\left(\text{pitch} + \arcsin\left(\frac{a}{c}\right)\right)}\right) \tag{2.10}$$

ผลการทดลองของขั้นตอนวิธีที่เสนอมานี้แสดงในรูปที่ 2-21 ซึ่งขั้นตอนวิธีนี้มีข้อดีคือใช้การคำนวณน้อย จึงสามารถประมวลผลได้เร็ว อย่างไรก็ตามมีข้อจำกัดเรื่องร่างกายจะต้องนั่งนิ่งอยู่กับที่และต้องกำหนด ค่าโครงแบบ (Configuration) ได้แก่ ค่า f และ t ซึ่งไม่สอดคล้องกับวัตถุประสงค์ของวิจัยที่กำหนดไว้ในตอนต้นว่าจะนำเสนอขั้นตอนวิธีที่ไม่ใช้โครงแบบ



รูปที่ 2-21 การทดลองการประมาณท่าทางของศีรษะ

2.2.3 Motion regularization for model-based head tracking

Sumit Basu และคณะ [9] นำเสนอขั้นตอนวิธีการประมาณท่าทางของศีรษะ โดยมีวัตถุประสงค์เพื่อที่จะคิดค้นขั้นตอนวิธีที่ไม่อาศัยจุดสำคัญเพียงบางจุดบนส่วนภาพใบหน้า อ้างอิงไปตลอดทั้งภาพวีดิทัศน์เนื่องจากในบางท่าทางของศีรษะจุดอ้างอิงเหล่านั้นอาจเกิดปัญหาถูกบดบัง (Occlusion) ได้ นอกจากนี้ยังต้องการออกแบบให้ขั้นตอนวิธีไม่มีข้อจำกัดเกี่ยวกับลำตัวและศีรษะของผู้ใช้งาน กล่าวคือ ผู้ใช้งานสามารถขยับไปในทิศทางต่างๆได้อย่างอิสระ อย่างไรก็ตาม วิธีการนี้มีภาระการคำนวณที่สูงกว่าการประมาณท่าทางของศีรษะจากการคำนวณที่อาศัยจุดอ้างอิงบนส่วนภาพของใบหน้าเพียงไม่กี่จุด

แนวคิดที่นำเสนอในงานวิจัยนี้อาศัยแนวคิดพื้นฐานจาก Motion Regularization กล่าวคือ ระบบจะคำนวณเวกเตอร์การเคลื่อนที่จากภาพอินพุต จากนั้นจะหมุนตัวแบบศีรษะสามมิติแล้วแปลงตัวแบบให้เป็นภาพสองมิติเพื่อคำนวณการเคลื่อนที่เพื่อนำมาเปรียบเทียบกับเคลื่อนที่ที่คำนวณได้จากภาพ โดยมุมการหมุนของตัวแบบศีรษะ 3 มิติ ที่ให้ความคลาดเคลื่อนของเวกเตอร์การเคลื่อนที่น้อยที่สุดจะถือเป็นท่าทางของศีรษะที่ประมาณได้ซึ่งมีรายละเอียดต่างของขั้นตอนวิธีการ ดังนี้

2.2.3.1 ตัวแบบสี่ระ 3 มิติ

ตัวแบบ 3 มิติที่ใช้แทนสี่ระในงานวิจัยนี้ คือ รูปทรงเรขาคณิตที่มีลักษณะเป็นรูปทรงรี (ellipsoid) ซึ่งมีพารามิเตอร์ คือ r_x , r_y และ r_z ใช้กำหนดขนาดในพิกัด 3 มิติ โดยการนำตัวแบบไปใช้ในการคำนวณใดๆ จะอาศัยการสุ่มเลือกจุด (x, y, z) บนพื้นผิวมาจำนวน k จุด โดยแต่ละจุดที่สุ่มเลือกสามารถเขียนอยู่ในรูปของเวกเตอร์

$P_k = [x_k \ y_k \ z_k \ 1]^T$ และกำหนดให้ P_o แทนเซตของจุดที่สุ่มเลือกมาทั้งหมด

2.2.3.2 ตัวแบบการเคลื่อนที่

ในพิกัด 3 มิติ กำหนดให้

α = องศาในการหมุนบนแกน z

β = องศาในการหมุนบนแกน y

γ = องศาในการหมุนบนแกน x

t_x = การเลื่อนขนาน (translation) บนแกน x

t_y = การเลื่อนขนานบนแกน y

t_z = การเลื่อนขนานบนแกน z

เวกเตอร์การเคลื่อนที่ใดๆเขียนแทนด้วย $a = [\alpha \ \beta \ \gamma \ t_x \ t_y \ t_z]^T$ และ

เมทริกซ์ของการแปลง (Transformation Matrix) มีค่าเท่ากับ

$$T = \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma & t_x \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma & t_y \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

เมื่อ $c = \cos()$, $s = \sin()$ ตามลำดับดังนั้น ตำแหน่งใหม่ของ P_o เมื่อทำการแปลงการเคลื่อนที่ด้วยเวกเตอร์ a แล้วจะเท่ากับ $P = T \cdot P_o$

2.2.3.3 การปรับขนาดของตัวแบบสี่ระะ 3 มิติ

เนื่องจากขั้นตอนวิธีที่เสนอ อาศัยการเปรียบเทียบเวกเตอร์การเคลื่อนที่ จึงมีความจำเป็นต้องปรับขนาดของตัวแบบสี่ระะ 3 มิติ ให้เท่ากับส่วนภาพสี่ระะของผู้ใช้งานที่ปรากฏบนภาพอินพุต โดยการปรับขนาดตัวแบบสี่ระะเริ่มจากภาพอินพุตที่ผู้ใช้จะต้องมีใบหน้าตรง แล้วค้นหาตำแหน่งของตาทั้งสองข้าง, จมูกและปากด้วยวิธีการของ A. Pentland และคณะ [10] เพื่อนำไปวิเคราะห์หาขนาดของตัวแบบสี่ระะที่เหมาะสมจากฐานข้อมูลของการทดลองที่เตรียมไว้

2.2.3.4 การฉายตัวแบบสี่ระะ 3 มิติ ลงบนภาพ 2 มิติ

กำหนดให้ z_d แทนพารามิเตอร์ที่มีผลต่อขนาดของวัตถุที่ฉายลงฉาก 2 มิติ ซึ่งสมมติให้มีค่าเทียบเคียงได้กับระยะโฟกัส ดังนั้น ตำแหน่ง (x, y, z) ในปริภูมิ 3 มิติ สามารถแปลงเป็นพิกัด 2 มิติ ได้จากสมการ

$$x_v = \frac{x}{1 - z/z_d}, y_v = \frac{y}{1 - z/z_d} \quad (2.12)$$

โดยกำหนดให้ เมทริกซ์ของพิกัด 2 มิติที่คำนวณได้จากการแปลงจากพิกัด 3 มิติ P ไตๆนี้ เขียนแทนด้วย Q

2.2.3.5 การเคลื่อนที่ของตัวแบบ (Model Flow)

กำหนดให้ออปติคัลโฟลว์ ของจุด (x, y) ที่เคลื่อนที่จากจุด $(x-u, y-v)$ ในเฟรมก่อนหน้า เขียนแทนด้วย $[u \ v]^T$ เพราะฉะนั้น การจำลองการเคลื่อนที่ของตัวแบบสี่ระะ จะต้องเลือกจุดใดๆ ที่สามารถพบได้ทั้งในเฟรมปัจจุบัน และเฟรมก่อนหน้า ดังนั้นเมื่อกำหนดให้ V_i และ V_j แทนขั้วเขตของจุด 2 มิติที่เกิดจากการฉายตัวแบบสี่ระะ 3 มิติในเขต Q_i และ Q_j ตามลำดับ โดยที่ j แทนเฟรมปัจจุบัน และ i แทนเฟรมก่อนหน้า ดังนั้นเวกเตอร์ของตัวแบบการเคลื่อนที่ (Model Flow, F_M) ระหว่าง 2 เฟรมที่ติดกันแทนด้วย $F_M = V_j - V_i$ และเขียนแทนเวกเตอร์การเคลื่อนที่ตัวที่ k ด้วย $[u_{M,k} \ v_{M,k}]^T$

2.2.3.6 การเปรียบเทียบการเคลื่อนที่

การเปรียบเทียบการเคลื่อนที่จะทำการเปรียบเทียบระหว่างการเคลื่อนที่ของส่วนภาพศีรษะของผู้ใช้งาน และการเคลื่อนที่ของตัวแบบศีรษะ โดยกำหนดให้เซตของเวกเตอร์การเคลื่อนที่จากภาพอินพุต 2 มิติเขียนแทนด้วย F_A และเซตของเวกเตอร์การเคลื่อนที่จากตัวแบบศีรษะ 3 มิติ ที่ฉายลงจากเป็นพิกัด 2 มิติแล้วเขียนแทนด้วย F_M ดังนั้น การเปรียบเทียบการเคลื่อนที่คำนวณจาก ผลรวมค่าเฉลี่ยกำลังสองคงทน (Robust mean square) ของทุกๆเวกเตอร์สมาชิกใน F_A และ F_M โดยค่าเฉลี่ยกำลังสองคงทนของแต่ละเวกเตอร์การเคลื่อนที่คำนวณจากสมการ

$$u_k = (u_{M,k} - u_A(x_k, y_k))^2 + (v_{M,k} - v_A(x_k, y_k))^2$$

$$e_k = \begin{cases} u_k; u_k < u_t \\ u_t; u_k \geq u_t \end{cases} \quad (2.13)$$

เมื่อ

u_k = ความคลาดเคลื่อนระหว่างเวกเตอร์การเคลื่อนที่ของภาพและตัวแบบศีรษะ

u_t = ค่าขีดแบ่งความคลาดเคลื่อนค่าเฉลี่ยกำลังสองคงทน (Error Threshold)

e_k = ค่าความคลาดเคลื่อนของการเคลื่อนที่ที่ปรับด้วยค่าขีดแบ่งแล้ว

ดังนั้น ค่าความผิดพลาดเฉลี่ยของเวกเตอร์การเคลื่อนที่ที่เข้าเป็นตัวแทนของสถานะการหมุนตัวแบบศีรษะที่ P_o ที่มีพารามิเตอร์เซตการเคลื่อนที่เริ่มต้นที่ a_i แล้วหมุนตัวแบบศีรษะด้วยพารามิเตอร์เซต a_j โดยที่มีจำนวนคู่ของเวกเตอร์การเคลื่อนที่ที่ตรงกับสมาชิกใน F_A ทั้งหมด n_c (number of common) ตัว จะมีค่าเท่ากับ

$$E(P_o, a_i, a_j, F_A) = \frac{1}{n_c} \sum_{k=1}^{n_c} e_k \quad (2.14)$$

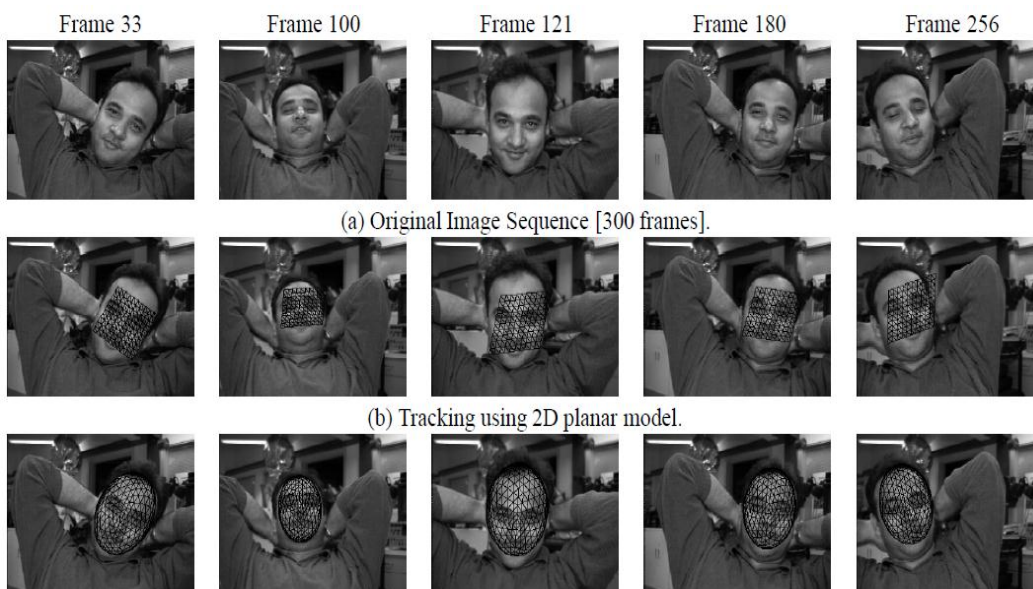
2.2.3.7 การเลือกสถานะที่เป็นคำตอบ

สถานะที่เป็นคำตอบ คือ สถานะของตัวแบบศีรษะที่ทำให้เกิดค่าความผิดพลาดของเวกเตอร์การเคลื่อนที่เฉลี่ยต่ำที่สุด โดยงานวิจัยชิ้นนี้อาศัยวิธีการค้นหาพารามิเตอร์ที่

ให้ผลลัพธ์ที่เหมาะสมที่สุด (Optimal Solution) ด้วยวิธีการเคลื่อนลงตามความชัน (Gradient Descent) โดยเขียนแทนสถานะคำตอบที่ดีที่สุด ด้วย

$$a_j^* = \arg \left(\min_j E(P_o, a_i, a_j, F_A) \right) \quad (2.15)$$

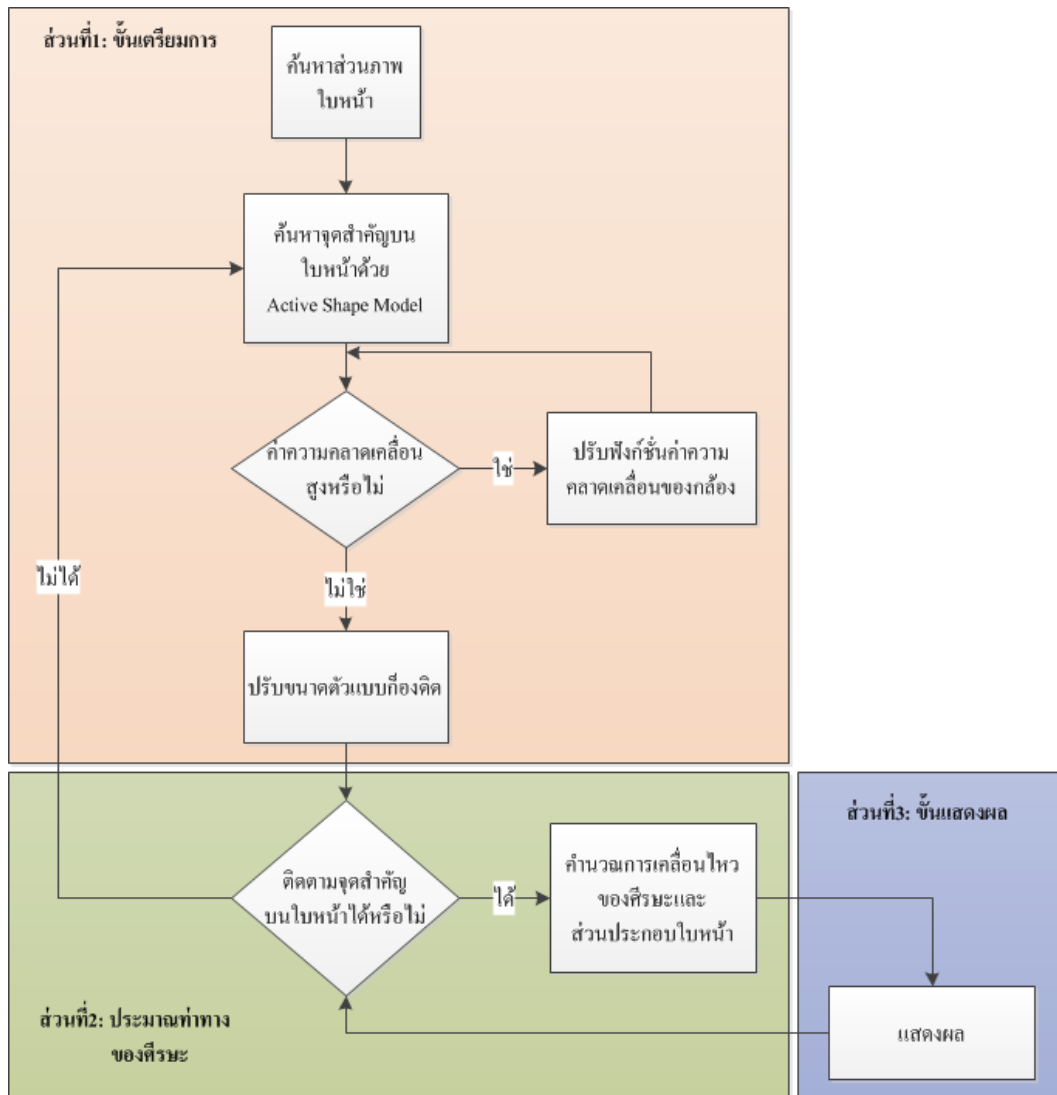
ผลการทดลองขั้นตอนวิธีการที่นำเสนอในงานวิจัยนี้ แสดงในรูปที่ 2-22 ในบทความระบุว่าสามารถใช้ประมาณท่าทางของศีรษะได้ดีโดยองศาในแต่ละแกนหมุนของศีรษะโดยจะมีความคลาดเคลื่อนไม่เกิน 10 องศา อย่างไรก็ตาม ขั้นตอนวิธีที่นำเสนอมีความซับซ้อนเชิงเวลาสูง เนื่องจากการคำนวณมาก โดยเฉพาะการค้นหาพารามิเตอร์สำหรับการเลือกสถานะที่เป็นคำตอบที่ให้ผลลัพธ์ที่เหมาะสมที่สุดด้วยการเคลื่อนลงตามความชัน ซึ่งไม่สามารถระบุความซับซ้อนเชิงเวลา (Time Complexity) ที่ต้องใช้ในแต่ละครั้งของการประมาณท่าทางศีรษะได้อย่างชัดเจนจึงมีโอกาสเป็นไปได้ที่จะไม่สามารถประมวลผลให้ทันต่อการแสดงผลแบบทันที



รูปที่ 2-22 ผลการทดลองของ Sumit Basu และคณะ [9]

บทที่ 3 ขั้นตอนวิธีที่เสนอ

ขั้นตอนวิธีขั้นตอนที่เสนอ ประกอบด้วย 3 ส่วนหลัก คือ ขั้นตอนเตรียมการ, การประมาณท่าทางของศีรษะ และการแสดงผล ดังแสดงในรูปที่ 3-1 โดย ขั้นตอนเตรียมการจะทำเพียงครั้งเดียวเมื่อตอนเริ่มใช้โปรแกรมและเมื่อสูญเสียการติดตามจุดสำคัญบนใบหน้า ส่วนของการประมาณท่าทางของศีรษะ และขั้นตอนการแสดงผล จะประมวลผลทุกอินพุตเฟรม ด้วยการประมวลผลแบบขนานบนหน่วยประมวลผลกราฟฟิก (GPU: Graphic Processing Unit)



รูปที่ 3-1 ภาพรวมขั้นตอนวิธีที่นำเสนอ

3.1 ชั้นเตรียมการ

3.1.1 ค้นหาส่วนภาพใบหน้า

เนื่องจากระบบที่พัฒนาเป็นการใช้งานในรูปแบบทันที การลดขนาดส่วนภาพอินพุตเพื่อประมวลผลส่วนภาพเฉพาะที่จำเป็นจึงเป็นสิ่งสำคัญ ดังนั้นในขั้นตอนแรกจึงเริ่มจากการค้นหาส่วนภาพใบหน้า เพื่อตัดส่วนภาพอินพุตที่ไม่เกี่ยวข้องออกไป โดยการหาตำแหน่งของใบหน้าผู้ใช้งานจากภาพอินพุตด้วยเทคนิค Haar-like feature เมื่อพบตำแหน่งของใบหน้าแล้ว ระบบจะกำหนดให้เป็นพื้นที่ที่สนใจ (Region of interest: ROI) เขียนแทนด้วย R ดังแสดงในรูปที่ 3-2 เพื่อใช้ในการปรับตัวอย่างก็องดิดตลอดจนใช้เพื่อการประมาณการหมุนของศีรษะต่อไป



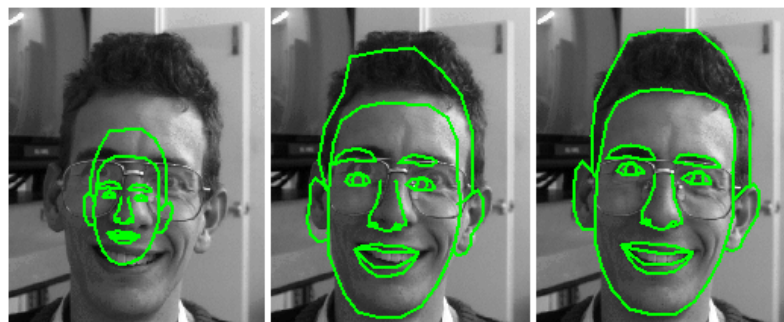
รูปที่ 3-2 การค้นหาส่วนภาพใบหน้า

3.1.2 ค้นหาจุดสำคัญบนใบหน้าด้วย Active Shape Model (ASM)

Active Shape Model (ASM) คือ ตัวแบบทางสถิติสำหรับการค้นหารูปร่างของวัตถุ ด้วยการปรับตัวอย่างที่สร้างไว้อย่างวนซ้ำเพื่อให้เข้ากับเส้นขอบของวัตถุจากภาพอินพุตให้มากที่สุด ดังแสดงในรูปที่ 3-3 คิดค้นโดย Tim Cootes และ Chris Taylor ในปี 1995 ซึ่งขั้นตอนการพัฒนาตัวแบบของวัตถุ เริ่มจากการฝึก (Training) ด้วยชุดของภาพเค้าโครงร่างตัวอย่างของวัตถุ โดยผลลัพธ์ของตัวแบบที่ได้ จะได้เป็นเซตของจุดที่สามารถลากเส้นเชื่อมต่อกันเป็นโครงร่างเส้นขอบของวัตถุได้

ในแต่ละภาพตัวอย่างที่ใช้ในการฝึก ขั้นตอนวิธีจะนำเซตของจุดดังกล่าว มาคำนวณด้วยหลักการของตัวแบบการกระจายจุด (Point Distribution Model) กับภาพตัวอย่างใหม่ ซึ่งจะเป็นตัวกำหนดว่า เซตของจุดเหล่านี้ จะจัดเรียงตัวอย่างไร และปรับความสัมพันธ์ของจุดต่างๆว่าสามารถปรับไปในทิศทางใดได้บ้าง ซึ่งในวิทยานิพนธ์ฉบับนี้

ได้นำขั้นตอนวิธีของ ASM มาใช้กับการค้นหาเค้าโครงของใบหน้า โดยตัวต้นแบบเค้าโครงของใบหน้าแสดงอยู่ในรูปที่ 3-4

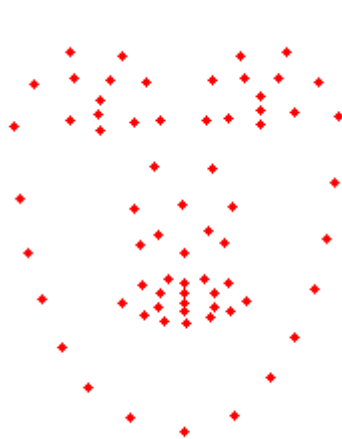


Initial Pose

After 5 iterations

Convergence

รูปที่ 3-3 การปรับตัวแบบที่สร้างไว้อย่างวนซ้ำ เพื่อให้เข้ากับเส้นขอบของวัตถุบนภาพอินพุต¹



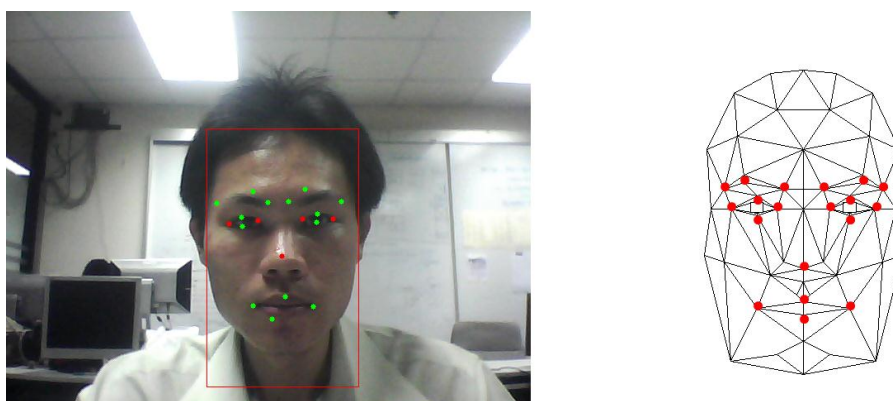
รูปที่ 3-4 ตัวแบบใบหน้า ASM ที่นำมาประยุกต์ใช้ในวิทยานิพนธ์ฉบับนี้

เนื่องจาก ในแต่ละภาพอินพุต ขั้นตอนวิธีของ ASM จะต้องปรับรูปร่างอย่างวนซ้ำ ดังนั้นในงานประมวลผลภาพวิถีทัศน์ที่ต้องการผลแบบทันที จะค้นหาและปรับรูปร่างตัวแบบ ASM เฉพาะเฟรมเริ่มต้น แล้วทำการติดตาม (Tracking) จุดสำคัญเหล่านี้แทน ดังจะได้กล่าวในหัวข้อ 3.2.1 การติดตามจุดสำคัญบนใบหน้า อย่างไรก็ตาม หากการติดตาม

¹ ที่มา: <http://personalpages.manchester.ac.uk/staff/timothy.f.cootes/Models/asms.html>

จุดสำคัญเหล่านี้เกิดข้อผิดพลาด ระบบก็จะค้นหาและปรับรูปร่างตัวแบบ ASM ใหม่ ดังที่
ได้แสดงไว้ในรูปที่ 3-1

การประยุกต์ใช้ขั้นตอนวิธีของ ASM ในวิทยานิพนธ์ฉบับนี้มีประเด็นที่สำคัญอีก
ประเด็นหนึ่ง คือ จำนวนจุดบนตัวแบบ ASM กล่าวคือ การประมาณท่าทางของศีรษะ มี
การใช้ตัวแบบศีรษะ 3 มิติที่ประกอบด้วย ซึ่งจำนวนคู่จุดสำคัญบนใบหน้าระหว่าง
ตัวแบบก็องติดกับจำนวนจุดของตัวแบบ ใบหน้า ASM มีไม่เท่ากัน โดยจำนวนจุดสำคัญ
บนใบหน้าของตัวแบบศีรษะก็องติด มีจำนวนน้อยกว่า ดังนั้น จึงได้ดัดแปลง ตัวแบบ
จุดสำคัญบนตัวแบบใบหน้า ASM ด้วยการลดรูปให้เหลือเพียง 19 จุด ให้ตรงกับ จุดยอด
(vertex) ที่ปรากฏบนตัวแบบก็องติด ดังแสดงในรูปที่ 3-5



รูปที่ 3-5 คู่จุดสำคัญบนใบหน้าทั้ง 19 ตำแหน่งระหว่าง
ตัวแบบ ASM และตัวแบบศีรษะก็องติด

3.1.3 การปรับค่าความคลาดเคลื่อน

ขั้นตอนวิธีการประมาณท่าทางของศีรษะที่นำเสนอในวิทยานิพนธ์ฉบับนี้ จัดอยู่ใน
กลุ่มเรขาคณิตดังที่ได้กล่าวไว้ในหัวข้อที่ 2.2.1.6 ซึ่งอาศัยการคำนวณการเคลื่อนที่ของ
วัตถุจากกลุ่มของจุดพิกเซลที่สนใจซึ่งมีภาระการคำนวณต่ำ โปรแกรมจึงสามารถทำงาน
ได้อย่างรวดเร็ว อย่างไรก็ตาม หากการติดตาม (tracking) หรือตรวจหาตำแหน่งของ
พิกเซล (detection) ที่นำมาใช้คำนวณการเคลื่อนที่ดังกล่าว มีความคลาดเคลื่อนแม้แต่
เพียงเล็กน้อย เมื่อนำไปใช้คำนวณเวกเตอร์การเคลื่อนที่ของศีรษะ ก็จะทำให้ผลลัพธ์ของ
เวกเตอร์การเคลื่อนที่ที่ผิดเพี้ยนไปอย่างมาก ซึ่งจากการวิจัย พบว่า ปัจจัยสำคัญที่ทำให้
เกิดความคลาดเคลื่อนของการตรวจจับตำแหน่งของพิกเซลก็คือ สัญญาณภาพของกล้อง

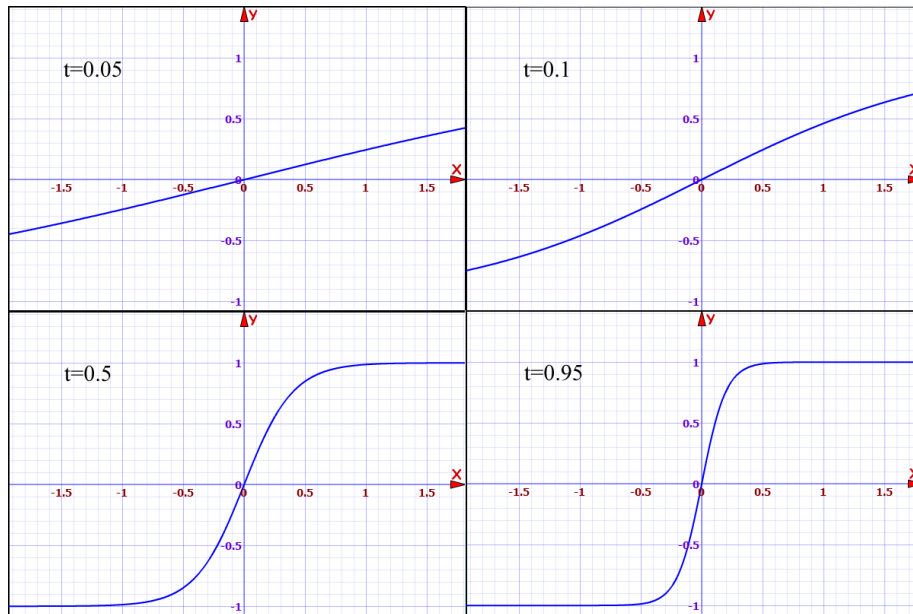
ที่มีสัญญาณรบกวน ดังนั้น จึงจำเป็นต้องตรวจหาค่าความคลาดเคลื่อนของสัญญาณภาพก่อน เพื่อให้สามารถประมาณตำแหน่งของพิกเซลที่เกิดขึ้นจริงได้อย่างแม่นยำ

กระบวนการปรับค่าความคลาดเคลื่อนของสัญญาณภาพที่นำเสนอในวิทยานิพนธ์ฉบับนี้ สามารถทำงานได้โดยอัตโนมัติ หลักการทำงานคือ ในขั้นต้นให้ผู้ใช้นั่งอยู่นิ่งๆเป็นระยะเวลา 2 – 3 วินาที (Initialization) ซึ่งระบบจะค้นหาจุดสำคัญบนใบหน้าด้วย ASM ดังนั้น หากกล้องรับภาพที่ความเร็ว 30 ภาพต่อวินาที จำนวนจุดทั้งหมดที่ใช้คำนวณเท่ากับ $19 \times 30 = 570$ จุดต่อ 1 วินาที ทั้งนี้การปรับค่าความคลาดเคลื่อนโดยอัตโนมัติ ซึ่งมีสมมติฐานว่า ณ ช่วงเวลาดังกล่าว แต่ละจุดสำคัญบนใบหน้าไม่มีการเคลื่อนที่เนื่องจากผู้ใช้สามารถเคลื่อนไหวได้ไม่เกินค่าที่กำหนด ดังนั้น การเปลี่ยนแปลงตำแหน่งของจุดสำคัญบนใบหน้าใดๆ เกิดจากความไม่นิ่งของสัญญาณภาพ

กำหนดให้

- จุดสำคัญบนใบหน้าที่ p ในเฟรมที่ j เขียนแทนด้วย p_j
- ค่าความคลาดเคลื่อนของ p ในเฟรมถัดไปเขียนแทนด้วย $|p_j - p_{j+1}|$
- จำนวนจุดสำคัญบนใบหน้าที่น่าสนใจทั้งหมดมีจำนวน n จุด

จากสมมติฐานว่า การเปลี่ยนแปลงตำแหน่งของจุดสำคัญบนใบหน้าใดๆ เกิดจากความไม่นิ่งของสัญญาณภาพ ดังนั้น การปรับค่าความคลาดเคลื่อนอัตโนมัติ คือการหาตัวคูณ α ใดๆที่ทำให้ $|p_j - \alpha(p_{j+1})|$ มีค่าต่ำหรือใกล้เคียงศูนย์มากที่สุด โดยการหาตัวคูณดังกล่าวอาศัยคุณสมบัติพื้นฐานของฟังก์ชันซิกมอยด์ที่สามารถปรับความยืดหยุ่นของกราฟรูปตัว S ได้ตามพารามิเตอร์ t จึงเหมาะที่จะนำมาใช้หาค่าตัวคูณกับระยะห่างของจุดสำคัญบนใบหน้าที่แตกต่างกันเล็กน้อยในแต่ละอินพุตเฟรม ดังแสดงในรูปที่ 3-6



รูปที่ 3-6 กราฟของฟังก์ชันซิกมอยด์ ณ ค่า t ต่างๆ

จากที่ได้กล่าวมาในข้างต้น เมื่อแทนตัวคุณด้วยฟังก์ชันซิกมอยด์ สมการการปรับค่าความคลาดเคลื่อนอัตโนมัติอยู่ในรูปของฟังก์ชันได้ สามารถเขียนได้ ดังนี้

$$\begin{aligned}
 \xi(t) &= \sum_{i=1}^n (p_j) - \left(\frac{1-e^{-t}}{1+e^{-t}} \right) \sum_{i=1}^n (p_j) \\
 \xi(t) &= \sum_{i=1}^n (p_j) \left[1 - \left(\frac{1-e^{-t}}{1+e^{-t}} \right) \right] \\
 \xi(t) &= \sum_{i=1}^n (p_j) \left[\frac{1+e^{-t} - 1 + e^{-t}}{1+e^{-t}} \right] \\
 \xi(t) &= \sum_{i=1}^n (p_j) \left[\frac{2e^{-t}}{1+e^{-t}} \right] \\
 \xi(t) &= \sum_{j=1}^n \left(\left[p_j - p_{j+1} \left[\frac{2e^{-t}}{1+e^{-t}} \right] \right] \right) \approx 0
 \end{aligned} \tag{3.1}$$

ตัวอย่าง

สมมติ ตำแหน่งหัวคิวซ้ายที่ตรวจพบจาก 30 เฟรมแรก ในขณะที่ผู้ใช้งานไม่เคลื่อนไหว เป็นดังนี้

| เฟรมที่ | ตำแหน่ง | เฟรมที่ | ตำแหน่ง | เฟรมที่ | ตำแหน่ง |
|---------|----------|---------|----------|---------|----------|
| 1 | 126, 215 | 2 | 127, 212 | 3 | 121, 213 |
| 4 | 129, 213 | 5 | 124, 217 | 6 | 127, 214 |
| 7 | 123, 212 | 8 | 122, 210 | 9 | 125, 219 |
| 10 | 126, 216 | 11 | 125, 214 | 12 | 123, 213 |
| 13 | 122, 218 | 14 | 128, 210 | 15 | 129, 219 |
| 16 | 124, 215 | 17 | 126, 213 | 18 | 125, 213 |
| 19 | 126, 210 | 20 | 123, 219 | 21 | 122, 218 |
| 22 | 125, 212 | 23 | 129, 215 | 24 | 128, 214 |
| 25 | 128, 215 | 26 | 126, 210 | 27 | 124, 218 |
| 28 | 121, 219 | 29 | 121, 213 | 30 | 126, 216 |

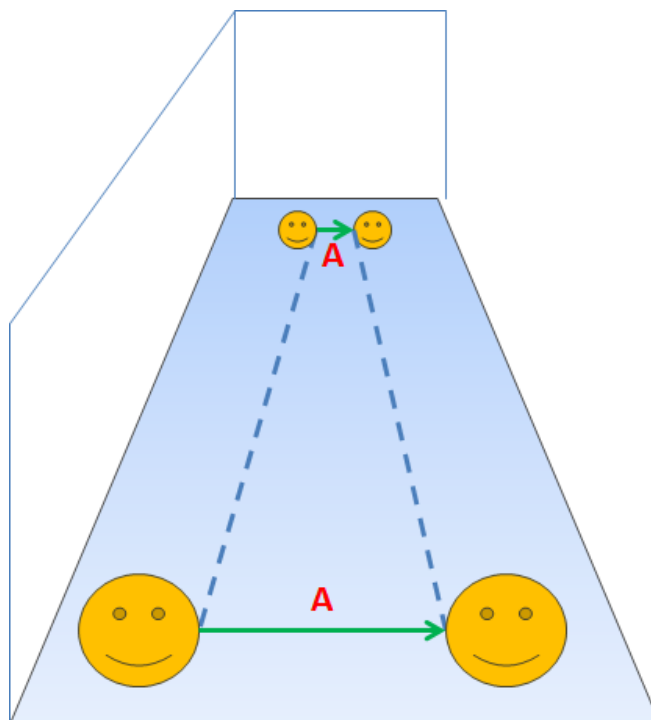
จากตัวอย่างในข้างต้น ผลเฉลี่ยระยะห่างทั้ง 30 จุด เท่ากับ 4.7639 จากนั้น จึงทำการหาค่าพารามิเตอร์ t ของฟังก์ชันปรับค่าความคลาดเคลื่อนที่ได้กล่าวไว้ในข้างต้น ที่ทำให้ผลรวมระยะห่างทั้ง 30 จุดมีค่าต่ำที่สุด โดยค่าของ t ที่ทดสอบอยู่ในช่วงระหว่าง 0.05 ถึง 0.95 โดยแต่ละค่าแตกต่างกัน 0.05 หน่วย

| t | ผลลัพธ์ |
|------|---------|
| 0.95 | 2.657 |
| 0.9 | 2.754 |
| ... | ... |
| 0.05 | 4.644 |

จากตัวอย่างนี้ ค่า t ที่เหมาะสมที่สุดคือ 0.95 ซึ่งค่าคงที่ของฟังก์ชันการปรับค่าความคลาดเคลื่อนที่ได้นี้ จะนำไปคูณกับค่าของตำแหน่งจุดสำคัญบนใบหน้าในทุกๆเฟรมที่ได้จากการตรวจจับ เพื่อให้ ตำแหน่งของจุดที่ได้ มีความคงตัวไม่แปรปรวนตามสัญญาณรบกวน

3.1.4 ปรับขนาดตัวแบบกึ่งดัด

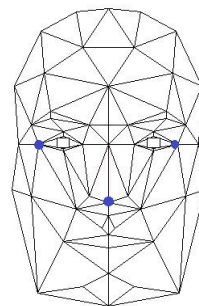
เนื่องจากขั้นตอนวิธีในงานวิจัยนี้ อาศัยการเปรียบเทียบเวกเตอร์การเคลื่อนที่ระหว่างส่วนภาพในหน้าของผู้ใช้งาน และตัวแบบกึ่งดัดศีรษะ 3 มิติ ซึ่งนอกเหนือจากทิศทางการหมุนของตัวแบบกึ่งดัดที่จะมีเวกเตอร์การเคลื่อนที่แล้ว ขนาดของตัวแบบกึ่งดัดเอง ก็มีผลต่อเวกเตอร์การเคลื่อนที่ด้วยเช่นกัน เนื่องจากวัตถุสามมิติที่อยู่ไกลจากผู้สังเกตการณ์ เมื่อหมุนไปในทิศทางใดๆ แล้วทำการฉายภาพลงฉากรับภาพ 2 มิติ เมื่อเปรียบเทียบกับเหตุการณ์เดียวกันนี้ แต่ขยับวัตถุสามมิติให้เข้ามาใกล้ผู้สังเกตการณ์มากขึ้น จะได้เวกเตอร์การเคลื่อนที่ ที่มีขนาดต่างกัน กล่าวคือ วัตถุที่อยู่ใกล้ผู้สังเกตการณ์จะได้เวกเตอร์การเคลื่อนที่ ที่มีทิศทางเดียวกันกับเมื่อวัตถุอยู่ห่างออกไปแต่มีขนาดใหญ่กว่า ดังแสดงในรูปที่ 3-7 เมื่อวัตถุอยู่ไกล การเคลื่อนที่ไปในระยะทาง A หน่วยจะมีการเปลี่ยนแปลง หรือ มีเวกเตอร์การเคลื่อนที่ ที่ดูเสมือนว่าน้อยกว่า เมื่อเทียบกับกรณีที่วัตถุขยับมาอยู่ใกล้กับกล้องรับภาพมากขึ้น



รูปที่ 3-7 ขนาดของวัตถุกับขนาดของเวกเตอร์การเคลื่อนที่

ดังนั้น ถ้าตัวแบบกึ่งดัดมีขนาดที่ใหญ่หรือเล็กกว่าส่วนภาพใบหน้าของผู้ใช้ เมื่อทำการหมุนแล้วคำนวณออกพิกัดคัลโฟลว์ จะทำให้ได้เวกเตอร์การเคลื่อนที่ที่มีขนาดใหญ่หรือ

เล็กเกินไป และจะส่งผลให้การประมาณท่าทางของศีรษะผิดเพี้ยนในที่สุด ดังนั้นจึงจำเป็นต้องปรับขนาดตัวแบบก็องดิดก่อนการประมาณท่าทางของศีรษะ โดยขั้นตอนการปรับขนาดของตัวแบบ ก็องดิดจะทำในขณะที่ผู้ใช้มีใบหน้าตรง โดยอาศัยตำแหน่งอ้างอิงจากขอบของดวงตาทั้งสองข้าง ข้างละสี่จุด และจุดกึ่งกลางบนปลายจมูก รวมทั้งหมด 3 จุด โดยเปรียบเทียบกับระยะห่างของจุดยอด (vertex) บนตัวแบบก็องดิด ดังแสดงในรูปที่ 3-8



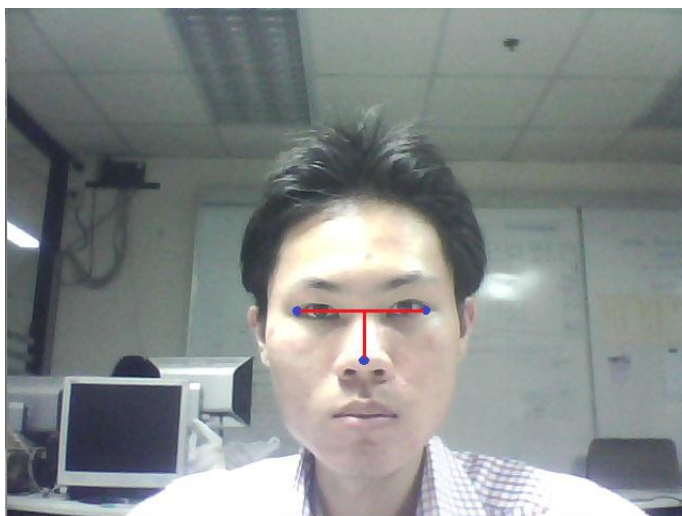
รูปที่ 3-8 คู่จุดที่ใช้ปรับขนาดตัวแบบก็องดิดและส่วนภาพใบหน้าของผู้ใช้งาน

การปรับขนาดตัวแบบก็องดิดจะคำนวณจากตัวแบบที่เรียกว่า T-Shape เนื่องจากมีลักษณะคล้ายตัวอักษร T ดังแสดงในรูปที่ 3-9 โดยส่วนหัวของตัว T แทนด้วยสัญลักษณ์ T_h ใช้สำหรับกำหนดความกว้างและส่วนลำตัวของ T แทนด้วยสัญลักษณ์ T_v ใช้ในการกำหนดความยาวของตัวแบบ และ กำหนดให้

- จุดปลายทางซ้ายของดวงตาซ้ายแทนด้วย v_{le}
- จุดปลายทางขวาของดวงตาขวาแทนด้วย v_{re}
- จุดกึ่งกลางระหว่าง v_{le} และ v_{re} แทนด้วย v_{ce}
- จุดกึ่งกลางของจมูกแทนด้วย v_n

ดังนั้น

$$T_h = \|v_{le} - v_{re}\| \quad T_v = \|v_{ce} - v_n\| \quad (3.2)$$



รูปที่ 3-9 โครงสร้าง T-Shape

3.2 การประมาณท่าทางของศีรษะ

3.2.1 การติดตามจุดสำคัญบนใบหน้า

เมื่อสามารถตรวจพบจุดสำคัญบนใบหน้าทั้ง 19 จุด ด้วย ASM และปรับค่าความเคลื่อนไหวของกล้องโดยอัตโนมัติดังที่ได้กล่าวมาแล้วในข้างต้น โปรแกรมจะเริ่มติดตามจุดสำคัญบนใบหน้าโดยใช้หลักการออปติคัลโฟลว์ ด้วยขั้นตอนวิธีของลูคาส-คานาเดะ โดยกำหนดให้ จุดสำคัญบนใบหน้าที่ p ในอินพุตเฟรมที่ i เขียนแทนด้วย p_i กระบวนการติดตามจุดสำคัญบนใบหน้าจะยุติลง แล้วเริ่มการค้นหาจุดสำคัญบนใบหน้าใหม่ เมื่อมีจุดสำคัญบนใบหน้าใดๆ หลุดออกไปนอกส่วนภาพของใบหน้า R

3.2.2 การคำนวณการเคลื่อนที่ของศีรษะ

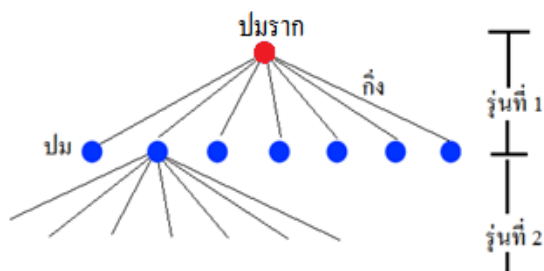
ขั้นตอนวิธีการประมาณท่าทางของศีรษะที่จะนำเสนอนี้ นำหลักการของการค้นหาคำตอบด้วยต้นทุนน้อยที่สุด (Least Cost Search) มาประยุกต์ใช้ กล่าวคือ ท่าทางของศีรษะใดๆ เกิดจากการก้มเงย (pitch) การหันหน้า (yaw) และการเอียงศีรษะ (roll) ผสมกัน โดยเรียกท่าทางของศีรษะใดๆว่า เป็นสถานะปัญหา (Problem State) และเมื่อกำหนดให้องศาของการก้มเงย การหันหน้า และการเอียงศีรษะนั้นสามารถทำได้อย่างละ 180 องศา

ดังนั้น ปริภูมิสถานะของปัญหาทั้งหมดที่สามารถเกิดขึ้นได้ (Global State Space) จะมีมากถึง 180^3 แบบ

อย่างไรก็ตาม เราสามารถนำฮิวริสติกที่มีอยู่ (Heuristic Knowledge) เกี่ยวกับท่าทางของศีรษะเข้ามาประยุกต์ใช้เพื่อลดทอนขนาดของสถานะทั้งหมดที่จะต้องค้นจริง กล่าวคือ เมื่อกำหนดให้ภาพอินพุตมีอัตราความเร็วที่ N ภาพต่อวินาที และข้อจำกัดทางกายภาพของมนุษย์ที่สามารถหมุนศีรษะไปในทิศทางใดทิศทางหนึ่งได้ไม่เกิน 180 องศาต่อวินาที ดังนั้น ปริภูมิสถานะท้องถิ่น (Local State Space) ต่อภาพ จะมีไม่เกิน $\left(\frac{180}{N}\right)^3$ กรณี โดยมีพิสัยของแต่ละแกนหมุนของศีรษะไม่เกิน $\frac{180}{N}$ องศา เช่น ภาพวีดิทัศน์ที่ความเร็ว 30 ภาพต่อวินาที ดังนั้น จะมีปริภูมิสถานะที่ต้องค้นหาจริงเท่ากับ $\left(\frac{180}{30}\right)^3 = 216$ สถานะ โดยมีพิสัยในแต่ละแกนหมุนของศีรษะไม่เกิน $\frac{180}{30} = 6$ องศาต่อภาพอินพุต 1 ภาพเท่านั้น

3.2.2.1 สถานะของปัญหา (State of a Problem)

ต้นไม้การค้นหา (Search Tree) ประกอบด้วย ปมราก (Root Node) ปม (Node) และ กิ่ง (Branche) ดังแสดงในรูปที่ 3-10 โดยปมในต้นไม้การค้นหาสำหรับการประมาณท่าทางศีรษะนี้เป็นส่วนเก็บข้อมูลของการหันหน้า มุมของการหันหน้า (Yaw) แทนด้วย γ มุมของการก้มเงย (Pitch) แทนด้วย β และ มุมของการเอียงศีรษะ (Roll) แทนด้วย α ซึ่งปมสามารถมีกิ่งที่เชื่อมโยงไปยังปมรุ่นลูกได้ โดยปมรุ่นลูกจะสืบคุณสมบัติของข้อมูลมาจากปมแม่ ซึ่งในที่นี้ คือ พิสัยขององศาของการหันหน้า การก้มเงย และ การเอียงศีรษะ



รูปที่ 3-10 โครงสร้างต้นไม้การค้นหา

กำหนดให้ ปมสถานะของการค้นหาใดๆที่ไม่ใช่ปมราก เขียนแทนด้วย $S_i(\gamma, \beta, \alpha)$ และปมสมาชิกในรุ่นถัดไปเขียนแทนด้วย S_j ซึ่งสมาชิกภายในรุ่นเกิดจากการนำพิสัยการค้นหาบนแต่ละแกนหมุนของสี่ระฆังของปมแม่มาลดลงครึ่งหนึ่ง ดังนั้น เมื่อกำหนดให้ พิสัยการค้นหาจากปมแม่ของการหันหน้าแทนด้วย δ_γ พิสัยการค้นหาจากปมแม่ของการก้มเงยแทนด้วย δ_β และ พิสัยการค้นหาจากปมแม่ของการเอียงสี่ระฆังแทนด้วย δ_α ปมสถานะแม่การค้นหาที่ S_i ใดๆ สามารถสร้างปมสมาชิกรุ่นลูก S_j ออกมาได้ 26 สถานะ ดังนี้

$$S_j \in \left\{ \begin{array}{l} S_i \left(\gamma \pm \frac{\delta_\gamma}{2}, \beta, \alpha \right), S_i \left(\gamma, \beta \pm \frac{\delta_\beta}{2}, \alpha \right), S_i \left(\gamma, \beta, \alpha \pm \frac{\delta_\alpha}{2} \right) \\ S_i \left(\gamma \pm \frac{\delta_\gamma}{2}, \beta \pm \frac{\delta_\beta}{2}, \alpha \right), S_i \left(\gamma \pm \frac{\delta_\gamma}{2}, \beta, \alpha \pm \frac{\delta_\alpha}{2} \right) \\ S_i \left(\gamma, \beta \pm \frac{\delta_\beta}{2}, \alpha \pm \frac{\delta_\alpha}{2} \right), S_i \left(\gamma \pm \frac{\delta_\gamma}{2}, \beta \pm \frac{\delta_\beta}{2}, \alpha \pm \frac{\delta_\alpha}{2} \right) \end{array} \right. \quad (3.3)$$

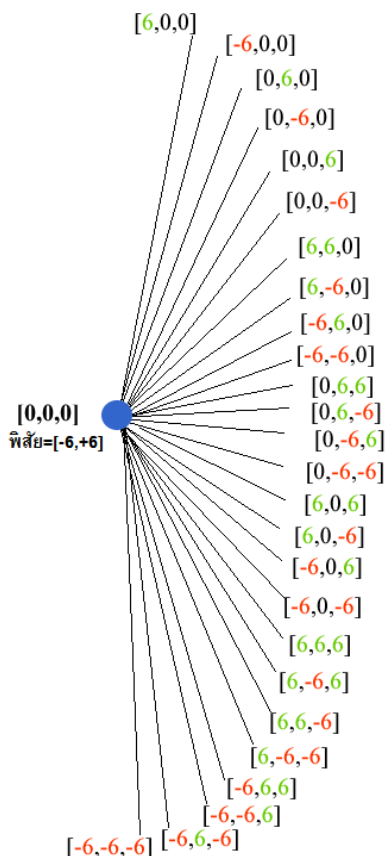
ตัวอย่าง กำหนดให้

- ภาพอินพุตจากวีดิทัศน์ที่ความเร็ว 30 ภาพต่อวินาที

- ผู้ใช้งานสามารถหมุนสี่ระฆังไปในทิศทางต่างๆได้ไม่เกิน 180 องศาต่อวินาที

ในเฟรมที่ 1 ผู้ใช้งานมีใบหน้าตรง, $S_1 = (0, 0, 0)$, เมื่อตรวจพบการเปลี่ยนแปลงของท่าทางสี่ระฆังในเฟรมที่ 2 ระบบจะสร้างสถานะปัญหา รุ่นที่ 1 โดยมีพิสัยการค้นหาแต่

ละแกนหมุนของสี่ระยะเท่ากับ $\frac{180}{30} = 6$ องศา แต่เนื่องจากการค้นหาเป็นแบบสองทิศทาง คือ สามารถเป็นได้ทั้ง +6 หรือ -6 องศา ดังนั้นพิสัยของการค้นหาในรอบที่ 1 เท่ากับ [-6, +6] ดังแสดงในรูปที่ 3-11



รูปที่ 3-11 การสร้างสถานะจาก $S_i \rightarrow S_j$

3.2.2.2 ฟังก์ชันต้นทุน (Cost Function)

ดังที่กล่าวไว้ในข้างต้น ปริภูมิสถานะท้องถิ่น (Local State Space) ต่อภาพจะมีไม่เกิน $\left(\frac{180}{N}\right)^3$ เมื่อ N แทนจำนวนของอัตราเฟรมต่อวินาที อย่างไรก็ตาม จำนวนดังกล่าวคำนวณจากหน่วยของมุมต่อ 1 องศา ซึ่งถ้าหากเราต้องการให้แต่ละสถานะมีความละเอียดมากขึ้นที่ระดับ 0.5 องศา ปริภูมิสถานะท้องถิ่นที่จะต้องตรวจสอบจะเพิ่มขึ้นเป็น $\left(\frac{180 \times 2}{N}\right)^3$ สถานะ ซึ่งการตรวจสอบสถานะปัญหาที่เป็นไปได้ทั้งหมดนี้

(Brute Force Search) จะใช้เวลามากเกินไปจนไม่สามารถประมวลผลแบบทันกาลได้ (Real Time) ดังนั้น เพื่อเป็นการแก้ไขปัญหาดังกล่าว จึงได้นิยามฟังก์ชันต้นทุน เพื่อเป็นตัวนำทางในการเลือกสถานะที่ควรจะค้นหา และนำไปสู่สถานะของคำตอบที่เหมาะสมที่สุด (The Optimal Solution) ได้อย่างรวดเร็ว โดยเราจะสร้าง S_j ใหม่ จาก S_i ที่มีต้นทุนต่ำที่สุด โดย กำหนดให้

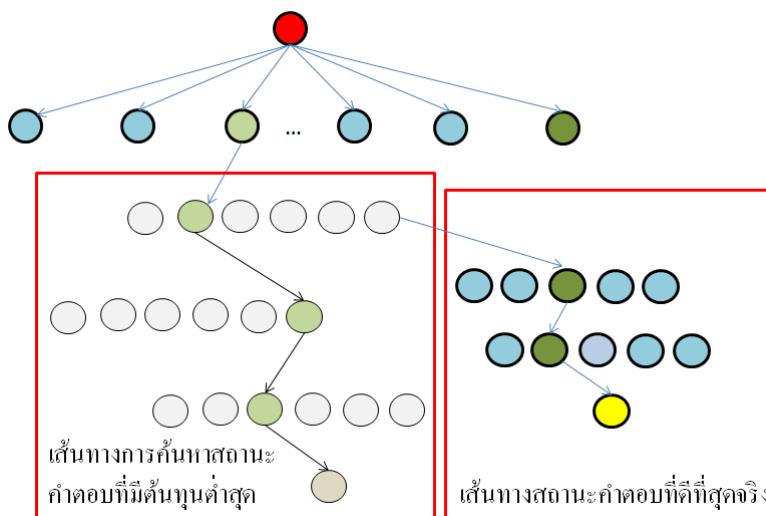
- ขนาดเวกเตอร์การเคลื่อนที่ตัวที่ i บนภาพอินพุต = $\|I\|_i$
- ทิศทางเวกเตอร์การเคลื่อนที่ตัวที่ i บนภาพอินพุต = θ_i
- ขนาดเวกเตอร์การเคลื่อนที่ตัวที่ j ของตัวแบบกึ่งองคิต = $\|C\|_j$
- ทิศทางเวกเตอร์การเคลื่อนที่ตัวที่ j ของตัวแบบกึ่งองคิต = θ_{C_j}
- จำนวนของเวกเตอร์การเคลื่อนที่จากภาพอินพุต = n

ต้นทุนของสถานะปัญหาที่ k เขียนแทนด้วย $E_k(\gamma, \beta, \alpha)$ คำนวณจากผลรวมระยะทางที่น้อยที่สุดของแต่ละคู่เวกเตอร์การเคลื่อนที่บนระบบพิกัดเชิงขั้ว (Polar coordinate system) ดังนี้

$$E_k(\gamma, \beta, \alpha) = \sum_{i=1}^n \left[\min_{1 \leq j \leq m} \left(\sqrt{\|I\|_i^2 + \|C\|_j^2 - 2\|I\|_i \|C\|_j \cos(\theta_i - \theta_{C_j})} \right) \right] \quad (3.4)$$

3.2.2.3 การค้นหาสถานะคำตอบแบบขนาน

เราสามารถนำฟังก์ชันต้นทุนดังที่ได้กล่าวไว้ในหัวข้อที่แล้วใช้เป็นตัวบ่งชี้เพื่อลดจำนวนสถานะปัญหาที่จะต้องตรวจสอบ (Prune Search) เพื่อเพิ่มความเร็วในการประมวลผลแบบทันกาล แต่จะทำให้ผลลัพธ์ที่ได้ เป็นเพียงคำตอบที่เหมาะสมที่สุด (The Optimal Solution) เท่านั้น เนื่องจากสถานะคำตอบที่อาจจะไม่ได้อยู่บนเส้นทาง การค้นหาที่มีค่าความคลาดเคลื่อนของฟังก์ชันต้นทุนที่น้อยที่สุดเสมอ จึงมีโอกาสเป็นคำตอบท้องถิ่น (Local Minimum Problem) ดังแสดงในรูปที่ 3-12



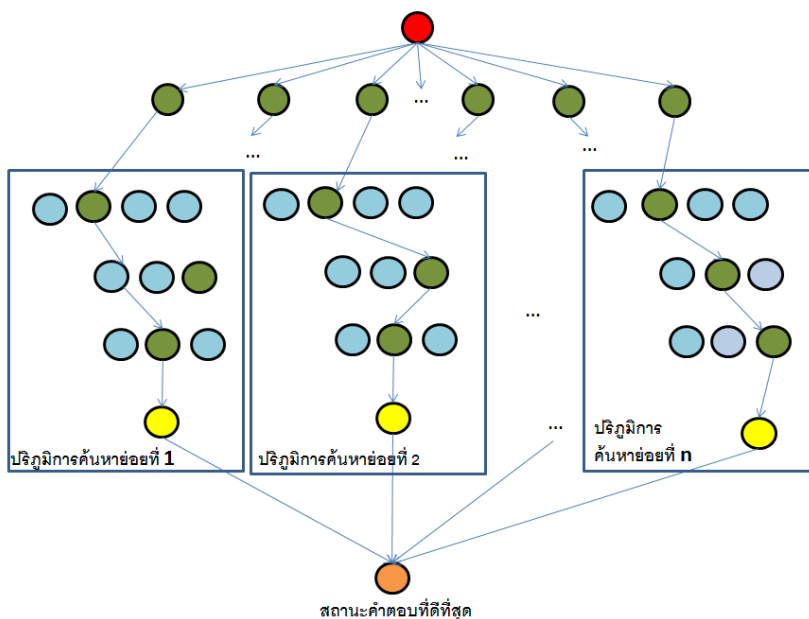
รูปที่ 3-12 เส้นทางที่นำไปสู่สถานะคำตอบที่ดีที่สุดอาจจะอยู่นอกเส้นทางที่ค้นหาที่มีต้นทุนต่ำสุดเสมอ

ผู้วิจัยจึงได้นำวิธีการประมวลผลแบบขนานด้วยหน่วยประมวลผลกราฟฟิค (GPGPU: General Purpose Graphic Processing Unit) ช่วยในการค้นหาสถานะคำตอบที่ดีที่สุด โดยขั้นตอนในการค้นหาสถานะคำตอบแบบขนาน เริ่มจากที่สถานะปรมากร (Root node) โดยสร้างสถานะรุ่นที่ 1 ทั้งหมดที่สามารถเป็นไปได้ แล้วกำหนดให้แต่ละสถานะ เป็นปรมากรของปริภูมิการค้นหาย่อย (Root node of Subspace) โดยแต่ละปริภูมิการค้นหาย่อย จะส่งไปประมวลผลพร้อมกันด้วยหน่วยประมวลผลกราฟฟิค จากนั้นหน่วยประมวลผลกลาง (CPU: Central Processing Unit) จะทำหน้าที่เปรียบเทียบสถานะคำตอบที่ดีที่สุด จากแต่ละปริภูมิการค้นหาย่อย แล้วเลือกสถานะที่มีต้นทุนต่ำที่สุดเป็นสถานะคำตอบ ดังแสดงในรูปที่ 3-13

กำหนดให้

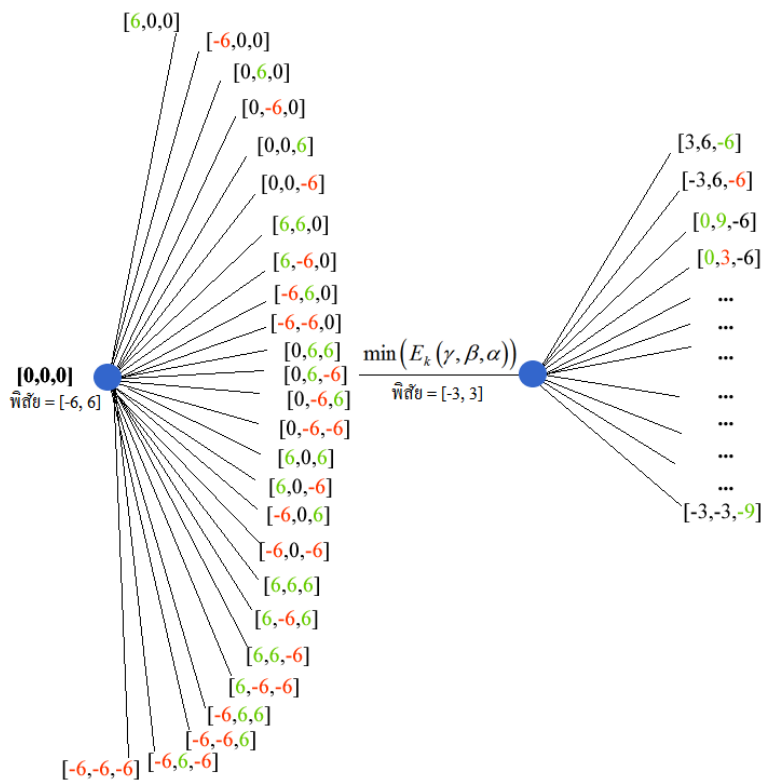
- ปริภูมิการค้นหาแบ่งออกเป็นปริภูมิการค้นหาย่อยทั้งหมด N ปริภูมิ
- ปริภูมิการค้นหาย่อยที่ k เขียนแทนด้วย L_k
- สถานะคำตอบที่ดีที่สุดของ L_k เขียนแทนด้วย l_k

ดังนั้น สถานะคำตอบที่ดีที่สุดของปริภูมิการค้นหาทั้งหมดในการประมาณค่าทาง ศีรษะจากภาพอินพุตใดๆ คือ $E = \min_{1 < k < N} (l_k)$



รูปที่ 3-13 การค้นหาสถานะคำตอบที่ดีที่สุดแบบขนาน

ภายใต้แต่ละปริภูมิการค้นหาลำดับที่ L ระบบจะสร้างสถานะลูกที่เป็นไปได้ทั้งหมด ด้วยหลักการตามที่ได้กล่าวไว้แล้วในหัวข้อ “3.2.2.1 สถานะของปัญหา” โดยพิสัยของมุม ในแต่ละแกนหมุน จะลดครึ่งละครึ่งหนึ่งจากสถานะรุ่นก่อน คือ $\delta_\gamma^i = \frac{\delta_\gamma^i}{2}$, $\delta_\beta^j = \frac{\delta_\beta^j}{2}$ และ $\delta_\alpha^k = \frac{\delta_\alpha^k}{2}$ โดยพิสัยรุ่นแรกที่ปมรากของแต่ละแกนหมุนของสี่ระฆังมีค่าตั้งแต่ -6 ไปจนถึง +6 องศาบนแกนหมุนทั้งสามแกน คิดเป็น 13^3 ดังนั้น ความสูงของต้นไม้หรือจำนวนรอบในการวนซ้ำ (Recursion) เท่ากับลอการิทึมความกว้างของพิสัยทั้งสามแกนหมุนของสี่ระฆังที่ปมรากฐานสองโดยประมาณคิดเป็น $\log_2(13^3) \approx 12$ รอบ ซึ่งแต่ละสถานะที่สร้างขึ้นนี้จะนำไปหมุนตัวแบบกึ่งอัตโนมัติ, คำนวณออกพิตคัลโฟลว์แล้วคำนวณต้นทุนของสถานะนั้นๆ แล้วตัดสินใจเลือกสถานะที่มีต้นทุนน้อยที่สุด เขียนแทนด้วย $\min(E_x(\gamma, \beta, \alpha))$ เป็นสถานะแม่ในรุ่นต่อไป ดังแสดงในรูปที่ 3-14



รูปที่ 3-14 การค้นหาสถานะคำตอบด้วยฟังก์ชันต้นทุน
ภายในแต่ละปริภูมิสถานะย่อย

3.2.3 การคำนวณการเคลื่อนที่ส่วนประกอบของใบหน้า

ส่วนประกอบของใบหน้าที่น่าสนใจในงานวิทยานิพนธ์ฉบับนี้ ได้แก่ คิ้วทั้งสองข้าง ตาทั้งสองข้าง และปาก ซึ่งล้วนเป็นที่อยู่บนพื้นผิวของศีรษะ ดังนั้น เมื่อมีการเคลื่อนที่ไปในทิศทางใด ส่วนประกอบของใบหน้าเหล่านี้ย่อมเคลื่อนที่ตามไปด้วย อย่างไรก็ตาม เราสามารถขยับคิ้ว ตา และปากได้อย่างเป็นอิสระจากการหมุนศีรษะ การเคลื่อนที่ของจุดสำคัญบนใบหน้าสามารถทำได้โดยการประมาณท่าทางของศีรษะก่อน แล้วบวกด้วยเวกเตอร์การเคลื่อนที่เฉพาะจุดสำคัญบนใบหน้า

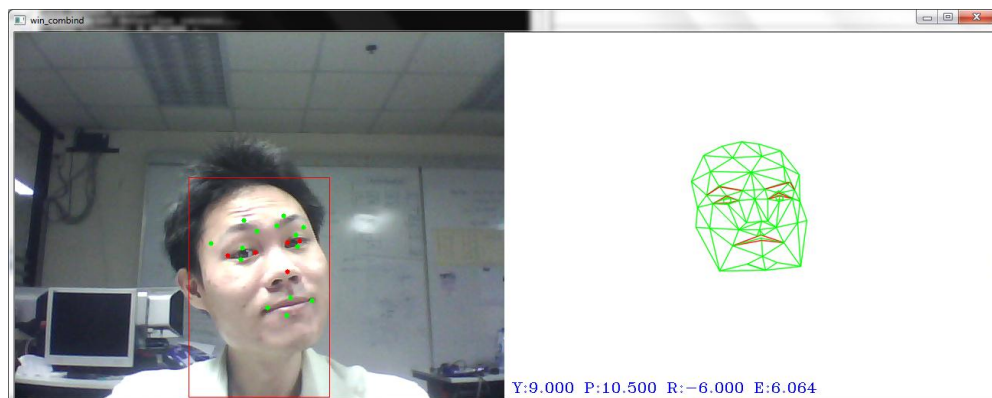
เนื่องจาก ความลึกของคิ้ว, ตา จมูก และ ปาก เมื่อเทียบกับความลึกของศีรษะแล้ว มีขนาดที่เล็กกว่ามาก และการแสดงออกทางส่วนประกอบของใบหน้า มีการเปลี่ยนแปลงความลึกที่น้อยมาก ดังนั้น จึงมีข้อสมมติฐานว่า การเคลื่อนที่ของส่วนประกอบบนใบหน้าใดๆ จะอยู่บนระนาบเดียวกันเสมอ ดังนั้น เมื่อกำหนดให้ เวกเตอร์การเคลื่อนที่จุดสำคัญบนใบหน้า p_i บนภาพอินพุต 2 มิติ เขียนแทนด้วย $t_i = [\Delta x \ \Delta y]^T$ เมื่อประมาณท่าทาง

ของศีรษะบนตัวแบบก็องติดแล้ว ตำแหน่งของจุดสำคัญบนใบหน้าบนตัวแบบก็องติดนั้นๆ ซึ่งเขียนแทนด้วย $c_i = (X, Y, Z)$ จะมีค่าของตำแหน่งใหม่บนพิกัด 3 มิติ เท่ากับ

$$c_i = (X + \Delta x, Y + \Delta y, Z) \quad (3.5)$$

3.3 การแสดงผล

การแสดงผลท่าทางของศีรษะที่ประมาณได้ในงานวิทยานิพนธ์ฉบับนี้ วาดออกมาเป็น ภาพโครงร่างตัวแบบศีรษะก็องติด โดยด้านใต้ของภาพจะแสดงค่าของมุม ก้มเงย เอียง และหัน หน้าโดยมีหน่วยเป็นองศา ส่วนองค์ประกอบของใบหน้า ได้แก่ คิ้ว ตา และปาก อาศัยการขยับ ตำแหน่งจุดยอดของโครงร่างตัวแบบศีรษะก็องติดเช่นเดียวกัน โดยเมื่อมีการขยับของ ส่วนประกอบบนใบหน้าใดๆ โปรแกรมจะเปลี่ยนสีของเส้นโครงร่างที่สัมพันธ์กับส่วนประกอบบน ใบหน้านั้นๆที่มีการเคลื่อนที่ เพื่อให้เห็นสังเกตเห็นได้ชัดเจนมากขึ้น ดังแสดงในรูปที่ 3-15 และมี หน้าจอแสดงผลการทำงานของโปรแกรมดังแสดงรูปที่ 3-16 เพื่อความสะดวกในการตรวจสอบการ ทำงานของระบบ



รูปที่ 3-15 ตัวอย่างการแสดงผลของโปรแกรม

```

Initializing GPU .....
  Get platform info      .... OK
  Get device info       .... OK
  Create context        .... OK
  Read kernel souce     .... OK
  Kernel compilation    .... OK
  Create command queue  .... OK
  Allocate cl buffer    .... OK
Reading active shape models from file D:/Testing/TestHeadPoseASM/TestHeadPoseASM
/my.amf ... (Done)
Facial point detection success...
: Time elapsed: 0.015000 s.
: Time elapsed: 0.002000 s.
: Time elapsed: 0.001000 s.
: Time elapsed: 0.002000 s.
: Time elapsed: 0.002000 s.
: Time elapsed: 0.002000 s.
: Time elapsed: 0.002000 s.
: Time elapsed: 0.002000 s.
: Time elapsed: 0.002000 s.
Facial point detection success...
Facial point detection success...
Facial point detection success...
: Time elapsed: 0.002000 s.
Facial point detection success...
: Time elapsed: 0.003000 s.
: Time elapsed: 0.003000 s.
: Time elapsed: 0.002000 s.
: Time elapsed: 0.002000 s.
: Time elapsed: 0.002000 s.
: Time elapsed: 0.002000 s.
Facial point detection success...
: Time elapsed: 0.002000 s.
: Time elapsed: 0.002000 s.
: Time elapsed: 0.002000 s.
: Time elapsed: 0.002000 s.
: Time elapsed: 0.001000 s.
: Time elapsed: 0.002000 s.
: Time elapsed: 0.002000 s.
: Time elapsed: 0.002000 s.
: Time elapsed: 0.003000 s.
: Time elapsed: 0.002000 s.
: Time elapsed: 0.003000 s.
: Time elapsed: 0.002000 s.

```

รูปที่ 3-16 หน้าจอแสดงสถานะการทำงานของโปรแกรม

บทที่ 4

การทดลองและผลการทดลอง

ภาพวีดิทัศน์ที่ใช้ในการทดสอบสำหรับเป็นอินพุตของขั้นตอนวิธีที่ได้นำเสนอในวิทยานิพนธ์ฉบับนี้ ได้บันทึกในสภาพแวดล้อมปิด เพื่อควบคุมปัจจัยที่ส่งผลกระทบต่อภาพของวีดิทัศน์ เช่น ความเข้มของแสง ระยะห่างระหว่างกล้องและผู้แสดงท่าทาง เป็นต้น โดยอุปกรณ์ที่ใช้ในการทดลอง ประกอบไปด้วย

- เครื่องคอมพิวเตอร์แล็ปท็อป

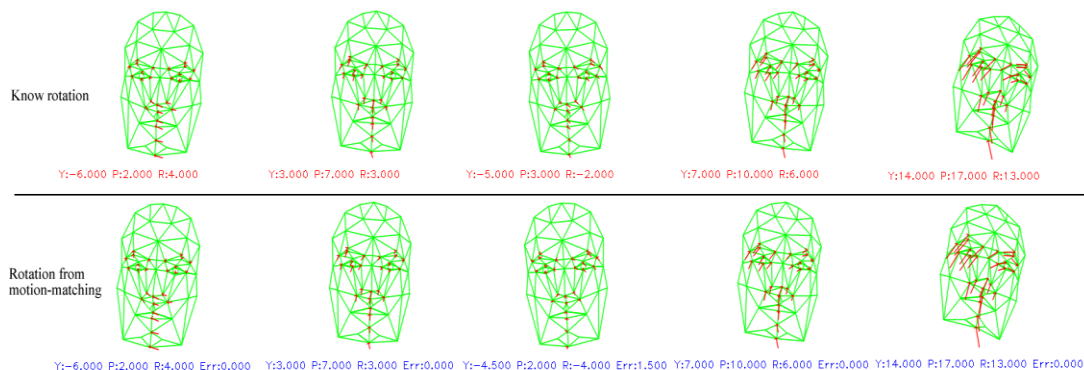
| | |
|----------------|----------------|
| ซีพียู | AMD A6 – 3400M |
| หน่วยความจำ | 8 GB |
| ระบบปฏิบัติการ | Windows 7 x64 |

- กล้องรับภาพ ความละเอียด 1.3 ล้านพิกเซล

ทั้งนี้ ผู้วิจัยได้ดำเนินการทดสอบตามขั้นตอนต่างๆ ดังนี้

4.1 การทดสอบความแม่นยำของขั้นตอนวิธี

ในขั้นตอนการตรวจสอบความแม่นยำของขั้นตอนวิธีนั้น ทำขึ้นเพื่อชี้วัดประสิทธิภาพของขั้นตอนวิธี โดยการทดลองจะจำลองการหมุนศีรษะในพิกัด 3 มิติในคอมพิวเตอร์ โดยกำหนดมุมที่จะหมุนในแต่ละแกนไว้ล่วงหน้า จากนั้นจึงใช้ขั้นตอนวิธีที่คิดค้นขึ้น ทำการหามุมการหมุนในแต่ละแกน แล้วเปรียบเทียบผลลัพธ์ที่ได้กับมุมที่กำหนดไว้ล่วงหน้า และเมื่อผลการทดสอบความแม่นยำของขั้นตอนวิธีมีค่าความคลาดเคลื่อนต่ำกว่าที่กำหนด คือ มีผลการประมาณท่าทางของศีรษะที่ได้คลาดเคลื่อนน้อยกว่า 5 องศา ตามแนวการพัฒนาขั้นตอนวิธีเรื่องความแม่นยำ (Accuracy) ที่ E. Murphy-Chutorian และ M.M. Trivedi แนะนำ ดังที่ได้แสดงไว้ในหน้าที่ 19 ซึ่งผลการทดลองแสดงดังรูปที่ 4-1

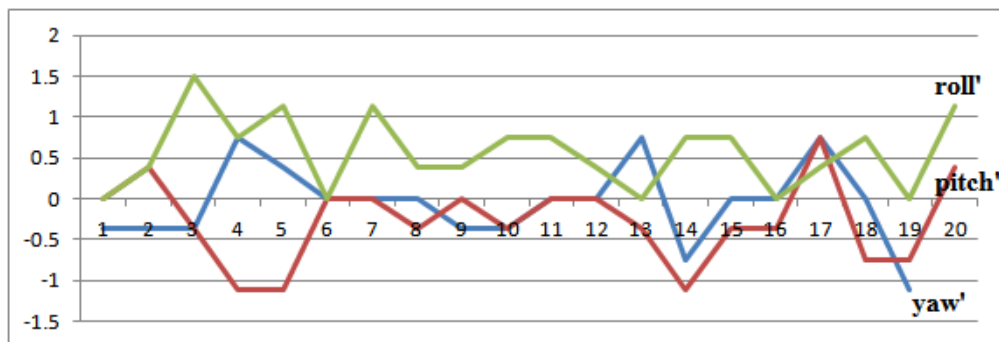


รูปที่ 4-1 ผลการทดสอบความแม่นยำของขั้นตอนวิธี

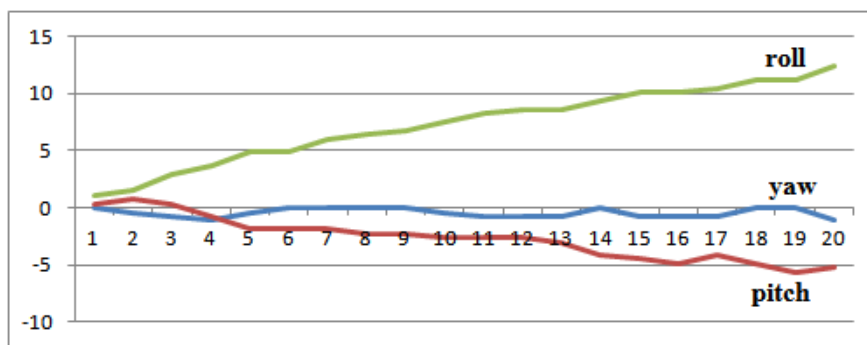
จากผลการทดลอง ความคลาดเคลื่อนที่วัดได้ มีไม่เกิน 1.5 องศา ซึ่งเป็นตัวเลขที่ต่ำกว่าเกณฑ์ที่ตั้งไว้ที่ 5 องศา จากการวิเคราะห์พบว่าสภาพแวดล้อมการทดสอบ อาศัยการสังเคราะห์ภาพศีรษะ จึงไม่มีสัญญาณรบกวนซึ่งต่างจากการใช้ภาพจริงจากวิดีโอ ดังนั้น การดำเนินงานในขั้นถัดไป จะนำขั้นตอนวิธีการมาประยุกต์ใช้กับภาพวิดีโอที่เตรียมไว้ในแล้ววิเคราะห์ผลลัพธ์ทั้งในแง่ของประสิทธิภาพของขั้นตอนวิธีการ และประสิทธิภาพในด้านความซับซ้อนเชิงเวลา (Time Complexity) ที่กำหนดไว้ให้เทียบเท่าต่ำกว่า $\log(n)$ เพื่อให้สามารถนำไปใช้ในสภาพแวดล้อมแบบทันที (Real-Time) ได้

4.2 การทดสอบประสิทธิภาพการปรับค่าความคลาดเคลื่อน

ในกรณีที่โปรแกรมเริ่มการทำงานโดยไม่มีค่าความคลาดเคลื่อนของสัญญาณ จากกล้องรับภาพจะพบว่า เมื่อผู้ใช้นั่งอยู่กับที่ และไม่มี การเคลื่อนที่ของศีรษะ แต่ท่าทางของศีรษะที่ประมาณได้จะผิดเพี้ยนไปจากความจริง คือ มีค่าเพิ่มขึ้นอยู่ตลอดเวลา เนื่องจาก ในแต่ละภาพ อินพุตจะมีการแกว่งของสัญญาณภาพอยู่เล็กน้อยเสมอ ดังนั้น เมื่อทำการติดตามจุดสำคัญบนใบหน้าบนจากภาพอินพุตที่มีสัญญาณรบกวน จะทำให้โปรแกรมเห็นการเคลื่อนที่ของจุดเหล่านั้น เป็นเวกเตอร์การเคลื่อนที่ของผู้ใช้ ดังแสดงในรูปที่ 4-2 ซึ่งทำให้การประมาณท่าทางของศีรษะจึงมีความคลาดเคลื่อนตามไปด้วย และค่าความผิดพลาดนี้ จะสะสมต่อไปเรื่อยๆตลอดการใช้งานโปรแกรม ดังแสดงในรูปที่ 4-3



รูปที่ 4-2 ความแปรปรวนของสัญญาณกล้องรับภาพทำให้
ตรวจจับการเคลื่อนที่ที่แปรปรวนตามไปด้วย



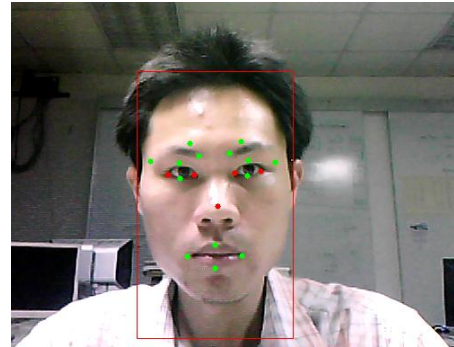
รูปที่ 4-3 การสะสมค่าผิดพลาดที่เกิดจากความแปรปรวนของสัญญาณกล้องรับภาพ

เมื่อทำการทดลองด้วยโปรแกรมที่เพิ่มความสามารถปรับค่าความคลาดเคลื่อนของสัญญาณจากกล้องรับภาพโดยอัตโนมัติดังที่กล่าวมาแล้ว จะพบว่าโปรแกรมใช้เวลาประมาณ 1 – 2 วินาที ในการค้นหาพารามิเตอร์เพื่อปรับลดการแกว่งของสัญญาณภาพ โดยในขณะที่โปรแกรมค้นหาพารามิเตอร์ดังกล่าว จุดสำคัญบนใบหน้าที่ตรวจพบ จะแสดงเป็นสีขาวเพื่อให้ผู้ใช้งานทราบว่า ควรหยุดอยู่นิ่งๆ เพื่อให้ระบบทำการปรับค่าความคลาดเคลื่อนของสัญญาณภาพก่อน ดังแสดงในรูปที่ 4-4

หลังจากนั้น เมื่อโปรแกรมสามารถปรับค่าความคลาดเคลื่อนของสัญญาณจากกล้องรับภาพได้แล้ว จุดสำคัญบนใบหน้าที่ตรวจพบจะเปลี่ยนเป็นสีเขียว และสีแดง ดังแสดงในรูปที่ 4-5 เพื่อให้ผู้ใช้งานโปรแกรมทราบว่า สามารถเริ่มขยับศีรษะและใบหน้าที่ได้แล้ว โดยจากการทดลองเมื่อเปรียบเทียบกับกรทดลองที่ไม่มีการปรับค่าความคลาดเคลื่อน พบว่า ไม่เกิดปัญหาการสะสมของข้อผิดพลาดแต่อย่างใด รวมทั้งให้ผลการประมาณท่าทางของศีรษะที่แม่นยำขึ้น โดยสามารถสังเกตได้จากภาพสังเคราะห์ของตัวแบบศีรษะที่โปรแกรมจำลองขึ้น ดังแสดงในรูปที่ 4-6



รูปที่ 4-4 ภาพของโปรแกรมขณะที่โปรแกรมกำลังค้นหาพารามิเตอร์เพื่อปรับความคลาดเคลื่อนของสัญญาณภาพ



รูปที่ 4-5 ภาพของโปรแกรมเมื่อปรับความคลาดเคลื่อนของสัญญาณภาพแล้ว



รูปที่ 4-6 ผลการทดลองใช้งานโปรแกรม

4.2 การทดสอบประสิทธิภาพการประมวลผลด้วย GPU เทียบกับ CPU

ในการทดสอบการประมวลผลทำทางของศีรษะด้วยการประมวลผลแบบขนาน ผู้วิจัย ได้เขียนโปรแกรมเพิ่มเติมในส่วนของการค้นหาสถานะของคำตอบที่ดีที่สุดด้วย OpenCL และใช้อุปกรณ์เพิ่มเติมคือ หน่วยประมวลผลกราฟฟิก AMD Radeon HD 6520G ซึ่งยังคงโครงสร้างการทำงานเดิมของโปรแกรมในส่วนอื่น ๆ ไว้เพื่อความแม่นยำในการเปรียบเทียบกับผลการประมวลผลด้วยซีพียู โดยมีผลการทดลองสรุปได้ ดังนี้

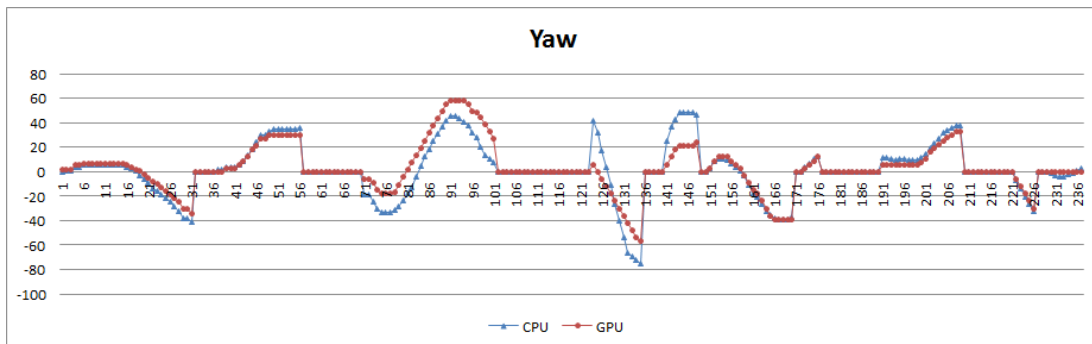
| | เวลาที่ใช้โดยเฉลี่ยในการประมวลผลทำทางของศีรษะต่อ 1 อินพุตเฟรม (หน่วย microseconds) | จำนวนสถานะที่ค้นหา | เวลาที่ใช้ต่อการคำนวณต่อ 1 หน่วยสถานะ (หน่วย microseconds) |
|-----|--|--------------------|--|
| CPU | 1,317.86 | 84 | 15.69 |
| GPU | 2,143.14 | 208 | 10.30 |

ผลการทดลองพบว่า จำนวนสถานะที่ค้นหาเมื่อประมวลผลด้วย CPU จะขึ้นอยู่กับฟังก์ชันค่าความคลาดเคลื่อน กล่าวคือ ตรวจจับที่ยังพบสถานะที่มีค่าความคลาดเคลื่อนต่ำลง โปรแกรมก็จะยังคงค้นหาต่อไป ซึ่งแตกต่างจากจำนวนสถานะการค้นหาด้วย GPU ซึ่งกำหนดไว้ชัดเจนเพื่อความสะดวกในการเขียนโปรแกรม โดยแบ่งออกเป็น 4 ปริภูมิสถานะย่อย โดยในแต่ละปริภูมิสถานะย่อย จะมี 26 สถานะ และทำการค้นหา 2 ครั้ง จึงมีจำนวนสถานะที่คำนวณคร่าวๆ $4 * 2 * 26 = 208$ สถานะ

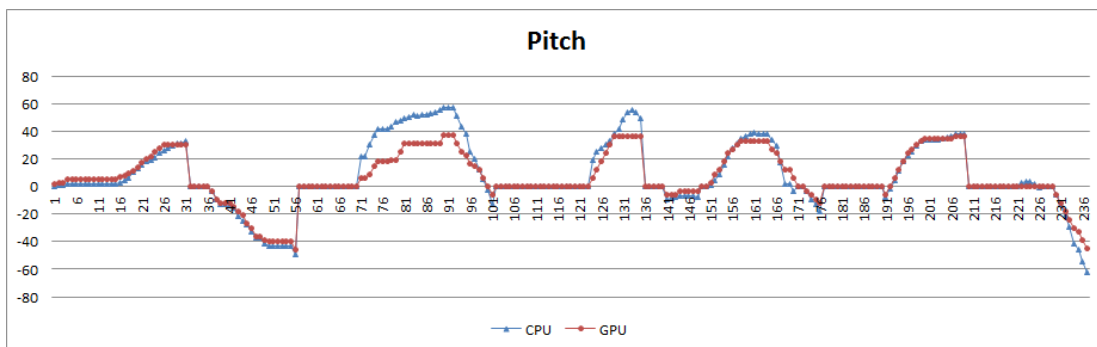
อย่างไรก็ตาม โดยเฉลี่ยแล้ว การประมวลผลด้วย GPU ใช้เวลามากขึ้นประมาณ 62.63% คือ เพิ่มขึ้นโดยเฉลี่ย 825.28 microseconds ต่อ 1 อินพุตเฟรม แต่ระยะเวลาการคำนวณที่ใช้เมื่อเทียบต่อ 1 หน่วยสถานะของการค้นหาคำตอบ กลับใช้เวลาลดลง 5.39 ไมโครเซคคั่น เนื่องจากมีจำนวนสถานะการค้นหาที่มากขึ้นประมาณ 2.5 เท่าต่อ 1 อินพุตเฟรม อย่างไรก็ตามระยะเวลา 825.28 ไมโครเซคคั่นต่อ 1 อินพุตเฟรมที่ใช้เพิ่มขึ้นนี้ เป็นช่วงระยะเวลาที่สั้นมาก ผู้ใช้จึงไม่รู้สึกรถึงการเปลี่ยนแปลง หรือส่งผลให้การทำงานของโปรแกรมช้าลงอย่างมีนัยสำคัญ

รูปที่ 4-7, 4-8 และ 4-9 แสดงถึงกราฟผลลัพธ์ทำทางของศีรษะที่ประมวลได้ในมุมหัน (Yaw), มุมก้ม (Pitch) และมุมเอียง (Roll) ตามลำดับ โดยมีหน่วยเป็นองศาทั้งหมดจำนวน 236 เฟรม ซึ่งภาพวีดิทัศน์ที่ใช้ ได้บันทึกไว้ล่วงหน้าแล้วจึงมาทำการประมวลผลเพื่อนำข้อมูลมาเปรียบเทียบระหว่างการคำนวณด้วย CPU และ GPU โดยพบว่าในภาพรวม ทั้ง 2 วิธี ประมาณทิศทางของศีรษะ ไปในทิศทางเดียวกัน แต่องศาการเคลื่อนที่ด้วยการคำนวณบน GPU จะ

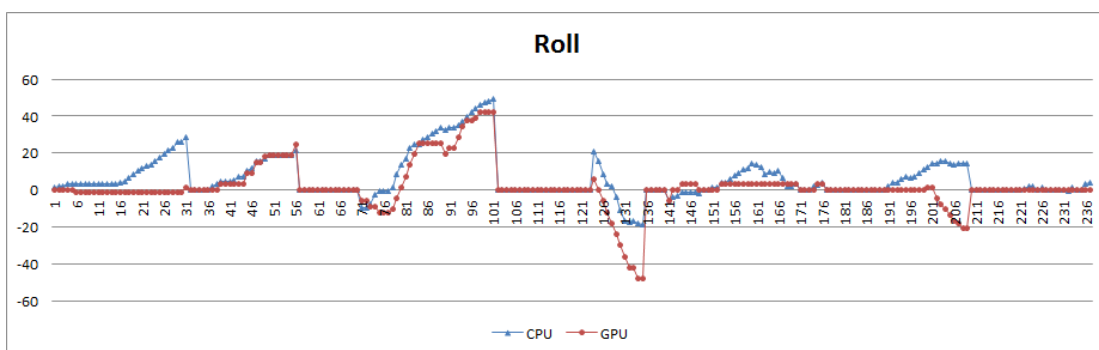
เปลี่ยนแปลงในช่วงที่แคบกว่า ทำให้ผลลัพธ์ที่ได้ เมื่อนำไปแสดงผลเป็นภาพสังเคราะห์ 3 มิติ มีการเคลื่อนไหวที่นุ่มนวลและต่อเนื่องมากกว่า



รูปที่ 4-7 ผลการประมาณท่าทางศีรษะเปรียบเทียบในมุมหัน (Yaw)



รูปที่ 4-8 ผลการประมาณท่าทางศีรษะเปรียบเทียบในมุมก้ม (Pitch)



รูปที่ 4-9 ผลการประมาณท่าทางศีรษะเปรียบเทียบในมุมเียง (Roll)

อย่างไรก็ตาม ยังมีบางกรณีที่ผลลัพธ์ระหว่าง CPU และ GPU แตกต่างกัน ดังแสดงในรูปที่ 4-8 ช่วงเฟรมที่ 201 ถึง 211 ซึ่งพบว่า ค่าตอบที่ได้จาก CPU และ GPU มีองศาที่คำนวณได้

ใกล้เคียงกัน แต่ไปในทิศทางตรงกันข้าม ซึ่งปัญหานี้เกิดจาก การประมาณท่าทางของศีรษะด้วย GPU มีโอกาสที่จะค้นพบสถานะที่ให้ค่าความคลาดเคลื่อนเท่ากันมากกว่า 1 สถานะ ระบบจึงจำเป็นต้องเลือกสถานะใดสถานะหนึ่งมาเป็นคำตอบ ซึ่งในงานวิจัยฉบับนี้ ในกรณีดังกล่าว โปรแกรมจะเลือกสถานะที่มีลำดับการค้นหาล่าสุดเป็นสถานะคำตอบ ดังนั้น จึงมีโอกาสที่ได้ผลลัพธ์ที่แตกต่างจากการคำนวณด้วย CPU

4.3 การแสดงผลเป็นภาพแอนิเมชัน 3 มิติ

ขั้นตอนวิธีที่นำเสนอในวิทยานิพนธ์ฉบับนี้ ได้นำไปประยุกต์ใช้งานกับงานวิจัยสำหรับภาษามือไทย ภายใต้ทุนสนับสนุนการวิจัยของ คณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ (กสทช.) โดยท่าทางของศีรษะที่ประมาณได้ มีหน่วยเป็นเรเดียน (Radian) บนแต่ละแกนหมุนของศีรษะ จะส่งไปยังโปรแกรมสร้างภาพแอนิเมชัน 3 มิติ จึงสามารถจำลองท่าทางของศีรษะผู้ใช้งานได้ โดยการแสดงผลของโปรแกรมสร้างภาพเสมือนนี้ อยู่ในรูปแบบทันทกาล ดังแสดงในรูปที่ 4-10



รูปที่ 4-10 ผลการทดลองร่วมกับโปรแกรมสร้างภาพแอนิเมชัน 3 มิติ

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

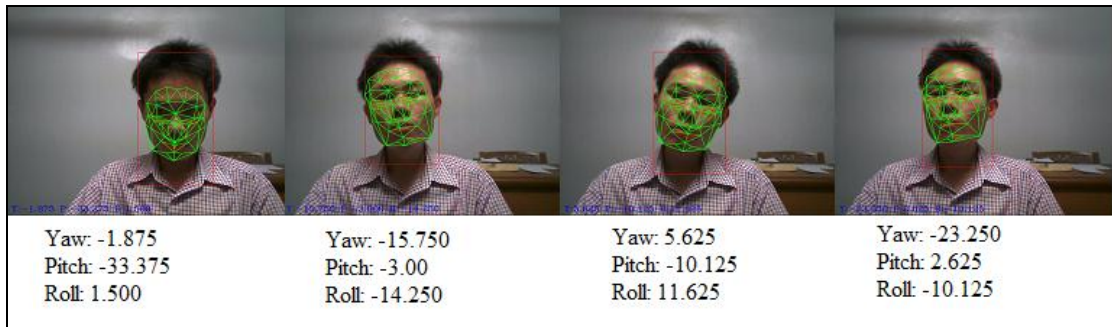
วิทยานิพนธ์ฉบับนี้ ได้นำเสนอขั้นตอนวิธีสำหรับการประมาณท่าทางของศีรษะและการเคลื่อนที่ขององค์ประกอบบนใบหน้าบนพิกัด 3 มิติ ด้วยข้อมูลภาพ 2 มิติ สำหรับบุคคลเดี่ยวที่สามารถประยุกต์ใช้ร่วมกับการประมวลผลแบบขนานบนหน่วยประมวลผลกราฟฟิกได้ โดยไม่ต้องอาศัยการกำหนดค่าโครงแบบ (Configuration) ใดๆ และมีความสามารถในการประมวลผลได้แบบทันกาล (Real-time) โดยให้ผลลัพธ์ของการประมาณท่าทางของศีรษะซึ่งสามารถทำงานได้ดีเมื่ออยู่ในช่วง -35 ถึง 35 องศาในแต่ละแกนหมุนของศีรษะ รวมทั้งยังสามารถคำนวณเวกเตอร์การเคลื่อนที่ส่วนประกอบของใบหน้า ได้แก่ คิ้ว ตา และปากด้วย โดยรูปที่ 5-1 ถึง 5-3 แสดงผลการวิจัยของขั้นตอนวิธีที่นำเสนอกับการใช้งานในกรณีต่างๆ ดังนี้



รูปที่ 5-1 การตรวจจับส่วนภาพใบหน้าอัตโนมัติ



รูปที่ 5-2 การเริ่มต้นตรวจจับใบหน้าใหม่อัตโนมัติเมื่อการติดตามใบหน้ามีข้อผิดพลาด



รูปที่ 5-3 การหมุนศีรษะไปในทิศทางต่างๆอย่างต่อเนื่อง

5.2 ข้อเสนอแนะ

ข้อเสนอแนะเพื่อนำวิทยานิพนธ์นี้ไปพัฒนาต่อไปในอนาคตมีดังนี้

- การปรับปรุงฟังก์ชันค่าความคลาดเคลื่อน

จากการทดสอบการประมาณท่าทางของศีรษะด้วย GPU พบว่า ในกรณีของการค้นหาสถานะจำนวนมาก มีโอกาสสูงที่จะได้ค่าความคลาดเคลื่อนต่ำที่สุด ที่มีค่าเท่ากับมากกว่า 1 สถานะขึ้นไป ดังนั้น แนวทางการการปรับปรุงฟังก์ชันค่าความคลาดเคลื่อน จะต้องคำนึงถึงปัจจัยอื่นๆ ที่ทำให้มีความแม่นยำและลดโอกาสเกิดค่าความคลาดเคลื่อนต่ำที่สุดมากกว่า 1 สถานะ

- การรู้จำการแสดงออกทางใบหน้า

ขั้นตอนวิธีที่นำเสนอในวิทยานิพนธ์ฉบับนี้ สามารถคำนวณเวกเตอร์การเคลื่อนที่ขององค์ประกอบใบหน้าได้ แต่นำมาใช้เพียงแสดงผลจำลองท่าทางของใบหน้าเท่านั้น ไม่ได้นำไปใช้กับการรู้จำความรู้สึก เช่น อารมณ์ดีใจ เสียใจ หรือ โกรธ เป็นต้น ดังนั้น แนวทางการพัฒนาขั้นต่อไป ระบบจะต้องสามารถรู้จำความรู้สึก ที่แสดงออกจากทางสีหน้าได้ด้วย

รายการอ้างอิง

- [1] Erik Murphy-Chutorian, Mohan M. Trivedi. Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation. IEEE Conference on Intelligent Transportation Systems, pp. 709–714, 2007.
- [2] Red Bull Formula Face. [ออนไลน์]. 2554. แหล่งที่มา:
http://www.redbull.com/cs/Satellite/en_INT/Game/Red-Bull-Formula-Face-021243076152177 [15 กันยายน 2555]
- [3] M. Rydfalk. Candide, a parameterized face. Doctoral dissertation, Dept. of Electrical Engineering, Linköping University, Sweden, 1987.
- [4] Paul Viola and Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features, IEEE Conference on Computer Vision and Pattern Recognition, vol.1, pp.511-518, 2001.
- [5] E. Murphy-Chutorian, M.M. Trivedi. Head Pose Estimation in Computer Vision: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.31, no.4, pp.607-626, 2009.
- [6] G. Garcia-Mateos, A. Ruiz, P.E. Lopez-de-Teruel, A.L. Rodriguez, L. Fernandez. Estimating 3D facial pose in video with just three points. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp.1-8, 2008.
- [7] G. Garcia-Mateos, A. Ruiz, and P.E. Lopez-de-Teruel. Human face processing with 1.5D models. The 3rd Int. Workshop on Analysis and Modeling of Faces and Gestures, vol. LNCS 4778, pp. 220–234, 2007.
- [8] G. Garcia-Mateos. Refining face tracking with integral projections. The 4th Intl. Conf. on Audio- and Video-based Biometric Person Authentication, vol. LNCS 2688, pp. 360–368, 2003.

- [9] S. Basu, I. Essa, A. Pentland. Motion regularization for model-based head tracking. International Conference on Pattern Recognition, vol.3, pp.611-616, 1996.
- [10] A. Pentland, B. Moghaddam, T. Starner, O. Oliyide, and M. Turk. View-based and modular eigenspaces for face recognition, IEEE Int Conf. in Computer Vision and Pattern Recognition, pp.84-91, 1994

ประวัติผู้เขียนวิทยานิพนธ์

นายนวิรัตน์ อรรถานุกูล เกิดเมื่อวันที่ 17 เมษายน 2526 ในกรุงเทพมหานคร จบการศึกษาชั้นมัธยมศึกษาที่โรงเรียนเทพศิรินทร์ กรุงเทพมหานคร และจบการศึกษาระดับปริญญาบัณฑิตจากภาควิชาบริหารสารสนเทศ คณะพาณิชยศาสตร์และการบัญชี มหาวิทยาลัยธรรมศาสตร์ เมื่อปีการศึกษา 2548