ระบบช่วยฝึกการออกกำลังกายแบบแอโรบิคโดยการประยุกต์ใช้เทคนิคการตรวจจับการเคลื่อนไหวของมนุษย์จากวีดิทัศน์

An Interactive Aerobic Training System Using Vision-based Motion Capture

โดย

นายอลงกต น้อยใจบุญ

รหัส 45066085

H002182

อาจารย์ที่ปรึกษา

ดร. ธนารัตน์ ชลิดาพงศ์

วัน เดือน ปี 06 ก.พ. 2550
เลขทะเบียน 02182
เลขเรียกหนังสือ วพ. ว422 ธ 2546
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2546

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	ระบบช่วยฝึกการออกกำลังกายแบบแอโรบิคโดยการประยุกต์ใช้เทคนิคการตรวจจับการเคลื่อนไหวของมนุษย์จากวีดิทัศน์
นักศึกษา	นายอลงกต น้อยใจบุญ
อาจารย์ที่ปรึกษา	ดร. ธนารัตน์ ชลิดาพงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2546

บทคัดย่อ

ระบบช่วยฝึกการออกกำลังกายแบบแอโรบิคโดยการประยุกต์ใช้เทคนิคการตรวจจับการเคลื่อนไหวของมนุษย์จากวีดิทัศน์ ขั้นตอนการทำงานของระบบเริ่มจากการนำข้อมูลที่ได้จากวีดิทัศน์ที่จับท่าทางของผู้ใช้นำมาแยกส่วนที่สนใจออกจากสิ่งแวดล้อมโดยใช้เทคนิค Background Subtraction ซึ่งจะได้ข้อมูลโครงร่างของของผู้ใช้ หลังจากนั้นนำโครงร่างของผู้ใช้ที่ได้นำไปวิเคราะห์ท่าทางโดยใช้เทคนิค Cardboard Model ในการวิเคราะห์ว่าตรงกับท่าทางของการ์ตูนหรือไม่ โดยนำผลที่ได้จากการวิเคราะห์ไปสร้างการโต้ตอบกับผู้ใช้โดยการ์ตูน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	An Interactive Aerobic Training System Using Vision-based Motion Capture
Student	Mr. Alongot Noichaiboon
Advisor	Dr.Thanarat H Chalidabhongse
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2003

ABSTRACT

In this report, we describe the development of an interactive aerobic training system using vision-based motion capture. The system applies the background subtraction technique for extracting the particular parts in the motion capture video, thereby resulting the silhouette. Subsequently, human gesture is analyzed by this silhouette. Cardboard Model is used for detecting and examining the human gesture compared with an interactive cartoon animation, thereby illustrating the result by the interactive cartoon animation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานพัฒนาระบบช่วยฝึกการออกกำลังกายแบบแอโรบิคโดยการประยุกต์ใช้เทคนิคการตรวจจับการเคลื่อนไหวของมนุษย์จากวีดิทัศน์ จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดีถ้าขาดความร่วมมือและการให้ความช่วยเหลือของบุคคลหลายท่านดังนี้

ขอขอบพระคุณ ดร.ธนารัตน์ ชลิดาพงศ์ ที่ได้ให้คำแนะนำในการพัฒนาโครงงานนี้ ตั้งแต่การศึกษาทฤษฎีที่เกี่ยวข้องจนถึงแนะนำขั้นตอนและหลักการในการพัฒนาระบบ

ขอขอบคุณเพื่อนๆ IS13.2 ทุกคนที่เป็นกำลังใจให้กันโดยเฉพาะ คุณพูลศักดิ์ จีรบุณย์ และคุณพนมพร สาครที่ได้ให้คำปรึกษาและช่วยทดสอบระบบทำให้เป็นแรงผลักดันให้สามารถพัฒนาระบบจนลุล่วง

และสุดท้ายที่มิอาจลืมที่จะกล่าวถึง คือ กำลังใจจากมารดา ยาย และพี่สาว ของข้าพเจ้า

อลงกต น้อยใจบุญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญภาพ VII

บทที่

1. บทนำ 1
   - 1.1. ความสำคัญและความเป็นมาของปัญหา 1
   - 1.2. วัตถุประสงค์ของการพัฒนาระบบงาน 1
   - 1.3. ขอบเขตของการพัฒนาระบบงาน 2
   - 1.4. ประโยชน์ที่คาดว่าจะได้รับ 2
2. ทฤษฎีที่เกี่ยวข้อง 3
   - 2.1. การตรวจจับและติดตามการเคลื่อนไหวของมนุษย์ 3
   - 2.2. การแยกส่วนที่สนใจออกจากสิ่งแวดล้อม (Target Segmentation) 3
   - 2.3. การวิเคราะห์หาส่วนต่างๆของร่างกายจากวีดิโอ 5
     - 2.3.1. การตรวจจับส่วนของร่างกายโดยใช้โครงร่าง (Silhouette) 5
     - 2.3.2. การตรวจจับและติดตามส่วนของร่างกายโดยใช้ Cardboard Model ใน W4 10
   - 2.4. หลักการเขียนภาษาเชิงวัตถุ และการสร้างโปรแกรมด้วย MS Visual C++ 16
   - 2.5. Microsoft Vision SDK 18
     - 2.5.1. โครงสร้างหลักของ MS Vision SDK 19
     - 2.5.2. คลาส CVisImage 20
       - 2.5.2.1. Pixel และ Image Type 21
       - 2.5.2.2. การสร้าง CVisImage Object 24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.5.2.3. Operations พื้นฐานของคลาส CVisImage 26

2.5.3. การแสดงข้อมูลภาพออกจากจอมอนิเตอร์ 27

2.5.3.1. แสดงผลโดยใช้ Windows HDC 27

2.5.3.2. แสดงผลโดยใช้คลาส CVisPane และ VisDisplayImage 28

2.5.4. การจัดการติดต่อกับอุปกรณ์ต่อพ่วง 28

3. การออกแบบ การพัฒนาโครงงาน และการทดลอง 30

3.1. การออกแบบระบบ 30

3.2. การพัฒนาโครงงาน 31

3.3. การทดลอง 32

3.3.1. อุปกรณ์ที่ใช้ในการทดลองระบบ 32

3.3.2. ขั้นตอนการทดลอง 32

3.3.2.1. การทดลองการสร้าง Background Model 32

3.3.2.2. การทดลองการโต้ตอบกับผู้ใช้ 33

3.4. สรุปผลการทดลอง 35

4. บทสรุปและข้อเสนอแนะ 36

4.1. บทสรุปของโครงการพัฒนาระบบงาน 36

4.2. ปัญหาและข้อจำกัดในการทำงานของระบบ 36

4.3. ข้อเสนอแนะในการพัฒนาระบบต่อไป 37

บรรณานุกรม 38

ประวัติผู้เขียน 39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

2.1. แสดงรายละเอียดของ pixel แต่ละชนิดในกลุ่ม Gray Scale Pixel Type.....22

2.2. แสดงรายละเอียดของ pixel แต่ละชนิดในกลุ่ม RGBA Color Pixel Types.....23

2.3. แสดงรายละเอียดของ pixel แต่ละชนิดในกลุ่ม YUVA Color Pixel Types.....24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้า

ภาพที่

2.1. การลบฉากหลังโดยใช้ รูปแบบจำลองของฉากหลัง 4

2.2. ขอบเขตโครงร่าง (Silhouette Boundary) ของมนุษย์ในหลายๆท่าทาง 5

2.3. ตัวอย่างของลำดับของส่วนของร่างกายบนเส้นขอบเขตโครงร่าง 5

2.4. ท่าทางต่างๆเมื่อแสดงให้เห็นมุมมองที่ต่างๆ 6

2.5. Horizontal and vertical Projection ที่ทำการ normalized แล้วของท่า ยืน, คลาน (ท่าโก้งโค้ง) และ ท่านอน (หมอบ) ที่ใช้ในการคาดเดาท่าทาง 7

2.6. การแสดงลำดับของตำแหน่งของส่วนต่างๆ 8

2.7. ท่ายืน Input image (a), convex-hull vertices (b), เส้นเชื่อมระหว่าง convex-hull vertices (c), convex และ concave hull vertices (d), และ เส้นเชื่อมจุด convex และ concave hull vertices (e) 9

2.8. การคาดเดาการเคลื่อนไหวของร่างกายโดยใช้โครงร่างจับคู่กัน 12

2.9. Cardboard model ที่ใช้ใน W^4 14

2.10. เป็นโครงร่างอย่างการใช้ Cardboard Model 15

2.11. แสดงโครงสร้างหลักของ MS Vision SDK 19

3.1. แสดงโครงสร้างของโครงงาน 30

3.2. Component diagram 31

3.3. แสดงตัวอย่างผลการทดลองสร้าง Background Model 33

3.4. แสดงตัวอย่างการระบุตำแหน่งของศีรษะ และมือทั้งสองข้างจากท่ายืนกางแขน 34

3.5. แสดงตัวอย่างการระบุตำแหน่งของศีรษะ และมือทั้งสองข้างจากท่าย่อขากางแขน 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและความเป็นมาของปัญหา

ในโครงงานนี้เป็นการออกแบบและพัฒนา ระบบช่วยฝึกการออกกำลังกายแบบแอโรบิค โดยการประยุกต์ใช้เทคนิคการตรวจจับการเคลื่อนไหวของมนุษย์จากวีดิทัศน์ เป็นระบบที่นำเอาทฤษฎีของComputer Vision มาประยุกต์ใช้กับการฝึกสอนการออกกำลังกายแบบแอโรบิค โดยแนวคิดที่จะพัฒนาระบบนี้เกิดขึ้นเนื่องจากได้พบว่า ระบบหรือสื่อช่วยฝึกการออกกำลังกายแบบแอโรบิคในปัจจุบันนั้นไม่สามารถโต้ตอบกับผู้ใช้ได้ ถึงแม้ว่าในบางสื่อจะมีการพูดคุยกับผู้ใช้ แต่สื่อช่วยสอนในการทำงานใช้การคาดเดาว่าผู้ใช้จะทำตามที่ผู้ฝึกสอนแนะนำ ซึ่งในความเป็นจริงแล้วผู้ใช้อาจจะไม่ได้ทำตามหรือทำท่าทางที่ไม่ตรงกับที่ผู้สอนสอนอยู่ จากเหตุผลดังกล่าวจึงมีแนวคิดที่จะพัฒนาระบบที่สามารถโต้ตอบกับผู้ใช้ได้จริง โดยนำเอาแนวคิดของการวิเคราะห์การเคลื่อนไหวมาใช้ร่วมกันกับการฝึกการออกกำลังกายแบบแอโรบิคโดยใช้วีดิทัศน์จับภาพผู้ใช้และนำข้อมูลที่ได้มาวิเคราะห์ว่าการเคลื่อนไหวของผู้ใช้เพื่อตรวจสอบความถูกต้องของท่าทางของผู้ใช้ โดยนำผลที่ได้จากการวิเคราะห์ไปสร้างการโต้ตอบกับผู้ใช้ เช่น บอกชมเมื่อทำท่าทางถูกต้อง และเปลี่ยนท่าไปทำท่าอื่นแทน ซึ่งจะทำให้ระบบสามารถวิเคราะห์ท่าทางของผู้ใช้ว่าทำท่าทางเดียวกันกับผู้ฝึกสอนหรือไม่ และสามารถโต้ตอบกับผู้ใช้ได้โดยใช้การ์ตูนในการโต้ตอบกับผู้ใช้ ซึ่งจะช่วยให้ผู้ใช้เกิดความเพลิดเพลินและปฏิบัติท่าทางได้ถูกต้องตามผู้ฝึกสอน

1.2 วัตถุประสงค์ของการพัฒนาระบบงาน

ในการพัฒนาการวิเคราะห์และออกแบบระบบช่วยฝึกการออกกำลังกายแบบแอโรบิคโดยการประยุกต์ใช้เทคนิคการตรวจจับการเคลื่อนไหวของมนุษย์จากวีดิทัศน์มีวัตถุประสงค์เพื่อ

1.2.1 เพื่อศึกษาหลักการและทฤษฎีทางด้าน Computer Vision ในเรื่องเกี่ยวกับการแยกส่วนที่สนใจออกจากสิ่งแวดล้อม

1.2.2 เพื่อศึกษาหลักการและทฤษฎีทางด้าน Computer Vision ในเรื่องเกี่ยวกับในการวิเคราะห์หาส่วนต่างๆของร่างกายจากวีดิทัศน์

1.2.3 เพื่อศึกษาการใช้ Microsoft Vision System development Kit (Vision SDK) ในการสร้าง application สำหรับการรับภาพจากวีดิทัศน์และการประมวลผลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.4 สร้างโปรแกรมประยุกต์ที่มีความสามารถในการตรวจจับและวิเคราะห์ท่าทางของผู้ใช้งานระบบช่วยฝึกการออกกำลังกายแบบแอโรบิค และ สามารถโต้ตอบกับผู้ใช้ระบบได้แบบทันทีทันใด (Real-Time)

1.3 ขอบเขตของการพัฒนาระบบงาน

ในการพัฒนาระบบช่วยฝึกการออกกำลังกายแบบแอโรบิคโดยการประยุกต์ใช้เทคนิคการตรวจจับการเคลื่อนไหวของมนุษย์จากวีดิทัศน์มีขอบเขตของการศึกษาและพัฒนาระบบ ดังต่อไปนี้

1.3.1 ในการวางตำแหน่งของกล้องตายตัว ไม่มีการเคลื่อนไหวกล้อง ณ เวลาที่ระบบเริ่มทำงาน

1.3.2 ท่าแอโรบิคที่ใช้ในการพัฒนาระบบมีจำนวน 3 ท่า

1.3.3 ตำแหน่งการของศีรษะของมนุษย์ในภาพวีดิโอที่ใช้ในการประมวลผล จะอยู่ด้านบนสุด และจำนวนผู้ใช้เพียงคนเดียว

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ได้ฝึกฝนการใช้ภาษาเชิงวัตถุในการพัฒนาโปรแกรมประยุกต์ที่สามารถนำไปใช้งานได้

1.4.2 ได้นำทฤษฎีด้าน Computer Vision ไปใช้ในการสร้างโปรแกรมประยุกต์ที่สามารถทำงานได้จริง และเป็นแนวทางในการพัฒนาโปรแกรมประยุกต์ในด้านอื่นๆต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงหลักการและทฤษฎีที่ได้นำมาใช้ในการพัฒนาโปรแกรมประยุกต์

2.1. การตรวจจับและติดตามการเคลื่อนไหวของมนุษย์

ขั้นตอนในการตรวจจับและติดตามการเคลื่อนไหวของมนุษย์จากวีดิทัศน์ นี้ จะต้องมีการเตรียมอุปกรณ์ ต่างๆในการทำงานโดยเฉพาะ โดยระบบกล้องจะอยู่กับที่ในช่วงเวลาที่ทำงาน จากนั้นมีขั้นตอนต่างๆดังนี้

2.1.1. การแยกส่วนที่สนใจออกจากสิ่งแวดล้อม (Target segmentation) เทคนิคที่เป็นที่นิยมใช้ในการแยกส่วนของร่างกายออกจากฉากหลังคือ เทคนิคการลบฉากหลัง (Background Subtraction) เทคนิคนี้เหมาะสำหรับกรณีที่ฉากหนังในวิดีโอไม่มีการเปลี่ยนแปลง หรือมีการเปลี่ยนแปลงน้อยเท่านั้น ในกรณีที่ฉากหลังมีการเคลื่อนไหวมาก จำเป็นต้องใช้เทคนิคอื่นในการแยกเป้าหมายได้แก่ การแยกโดยใช้การคำนวณหา Optical Show เป็นต้น

2.1.2 หลังจากได้ส่วนที่เป็นเป้าหมาย ซึ่งในที่นี้คือ ส่วนที่เป็นตัวโครงร่างมนุษย์ (Silhouette) ออกจากรูปแล้วก็จะทำการวิเคราะห์หาส่วนต่างๆของของร่างกายต่อไป (Body Part Localization) หรือใช้วิธีหา Motion Template เพื่อใช้ในการวิเคราะห์การเคลื่อนไหวแทนการหาตำแหน่งของส่วนของร่างกาย

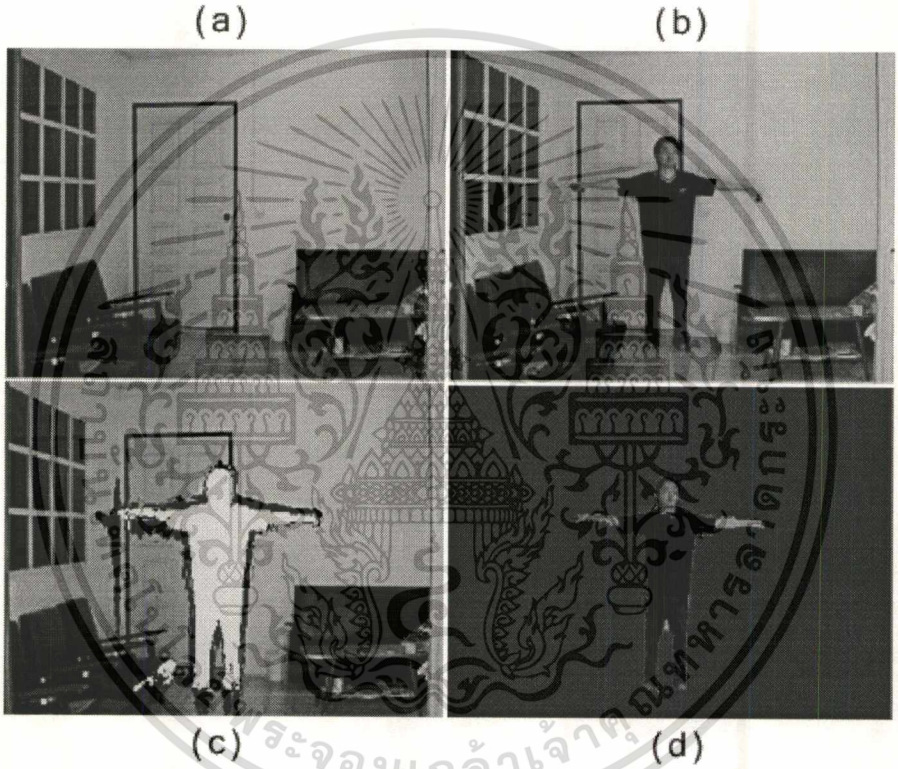
2.2. การแยกส่วนที่สนใจออกจากสิ่งแวดล้อม (Target Segmentation)

ในการแยกส่วนที่เราสนใจออกจากสิ่งแวดล้อมนั้นจะช่วยให้การคำนวณและการวิเคราะห์ในขั้นตอนต่อไปสามารถทำได้ง่ายขึ้นโดยเทคนิคที่เป็นที่นิยมและเหมาะสมกับในส่วนที่ศึกษาคือ เทคนิคการลบฉากหลัง(Background Subtraction) เป็นเทคนิคที่ง่าย และมีประสิทธิภาพ ในการแยกโครงร่าง (Silhouette) ของเป้าหมายหรือวัตถุที่เราสนใจออกจากฉากหลังในภาพ ซึ่งความสามารถในการแยกเป้าหมายออกจากฉากหลังหรือสิ่งแวดล้อมนั้น มีความสำคัญอย่างมากต่อขั้นตอนการวิเคราะห์อื่นๆ ในลำดับต่อไป ดังนั้นหากการลบฉากหลังทำได้ไม่ดีการวิเคราะห์โครงร่างเพื่อค้นหาอวัยวะส่วนต่างๆก็จะผิดตามไปด้วย

หลักการในการลบฉากหลังนั้น เริ่มจากการคำนวณเพื่อสร้างรูปแบบจำลองของฉากหลัง (background model) จาก วิดีโอ วิธีการนี้เหมาะสม ในกรณีที่กล้องตั้งอยู่กับที่ (stationary camera)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยฉากหลังไม่มีการเคลื่อนไหว หรือ เคลื่อนไหวน้อย เช่น ใบไม้หรือต้นไม้ที่โยกไปมาเนื่องจากแรงลม หลังจากสร้างรูปแบบจำลองของฉากหลัง (background model) แล้ว ในการแยกแยะเป้าหมาย (ในที่นี้คือร่างกายมนุษย์) จากวิดีโอ ให้นำแต่ละเฟรมของวิดีโอลบออกจากรูปแบบจำลองฉากหลัง ผลลัพธ์ที่ได้คือ โครงร่างของเป้าหมาย (target's silhouette) ดังแสดงในภาพที่ 2.1 โดยโครงร่างที่ได้จะนำไปใช้ในการวิเคราะห์ คำนวณหาตำแหน่งอวัยวะที่สำคัญของร่างกายในขั้นตอนต่อไป



ภาพที่ 2.1 การลบฉากหลังโดยใช้ รูปแบบจำลองของฉากหลัง

(a) Background model

(b) Input Image

(c)(d) Output (Silhouette)

จากภาพที่ 2.1 (a) แสดงแบบจำลองของฉากหลัง (Background model) ที่ถูกสร้างขึ้นจากช่วงเวลาที่แต่ละเฟรมที่ไม่มีเป้าหมาย (ในที่นี้คือร่างกายมนุษย์) นำช่วงเวลาดังกล่าวแต่ละเฟรมมาหาค่าเฉลี่ยในแต่ละ Pixel สร้างเป็นแบบจำลองของฉากหลัง (Background model)

ภาพที่ 2.1 (b) เป็นการแสดงภาพ วีดีโอที่มีเป้าหมายเข้าสู่ระบบเรียกว่า Input Image ข้อมูลดังกล่าวไปลบกับแบบจำลองของฉากหลัง (Background model) จะได้ส่วนที่แตกต่างซึ่งในการลบ

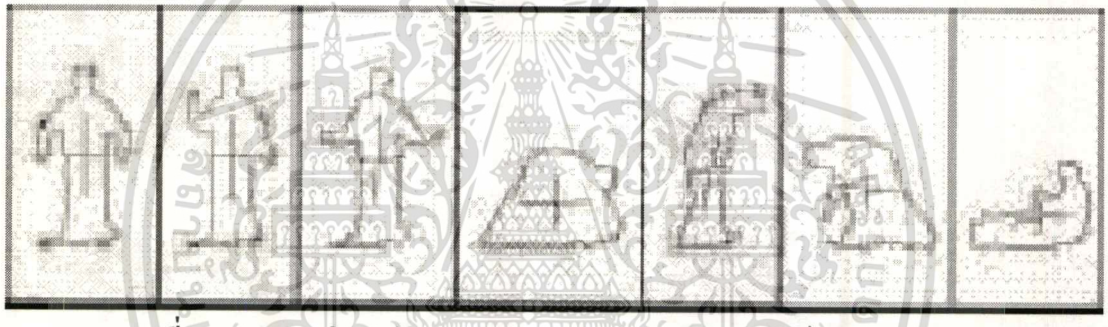
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กัน จะตรวจสอบถึงสีและแสงของข้อมูลด้วย หากมีสีเดียวกัน แต่แสงน้อยกว่าก็อาจคาดเดาได้ว่าเป็นเงาที่เกิดขึ้น หรือหากมีสีเดียวกัน แต่แสงมากกว่าก็อาจคาดเดาได้ว่าเป็นแสงสะท้อนที่เกิดขึ้นได้

2.3. การวิเคราะห์หาส่วนต่างๆของร่างกายจากวีดิโอ

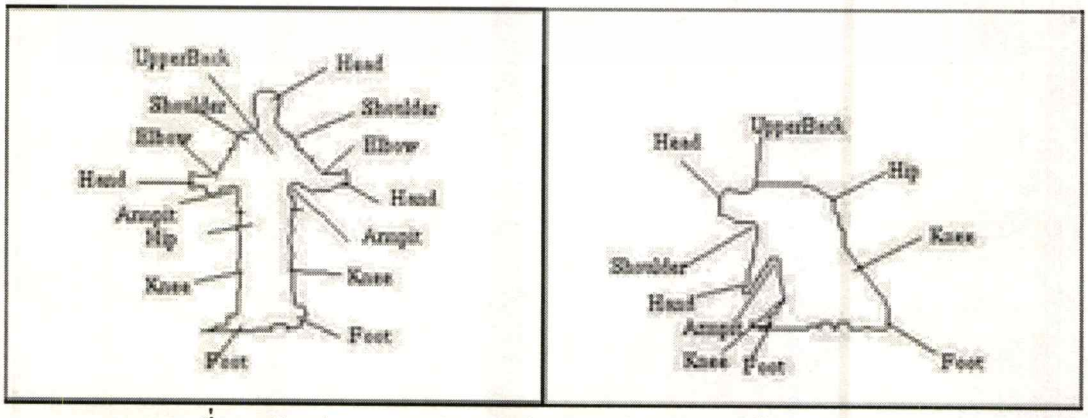
การวิเคราะห์เพื่อคำนวณหาตำแหน่งของอวัยวะที่สำคัญจากมุมมองแต่ละมุมมองหรือใน 2 มิตินั้นมีวิธีการ หรือเทคนิคในการหาตำแหน่งอวัยวะจากโครงร่างมนุษย์ในวิดีโอมีหลายวิธีซึ่งจะขึ้นอยู่กับการนำไปใช้เพราะในแต่ละวิธีจะมีข้อดีและข้อเสีย ดังตัวอย่าง 3 ตัวอย่างคือ

2.3.1 การตรวจจับส่วนของร่างกายโดยใช้โครงร่าง (Silhouette) หลังจากทำการลบฉากหลังแล้วผลลัพธ์ที่ได้ก็จะเป็นโครงร่างของร่างกายของเป้าหมาย ซึ่งเมื่อนำโครงร่างมาวิเคราะห์เส้นขอบเขตโครงร่าง (Silhouette Boundary) ดังภาพที่ 2.2 จะพบว่า



ภาพที่ 2.2 ขอบเขตโครงร่าง (Silhouette Boundary) ของมนุษย์ในหลายๆท่าทาง

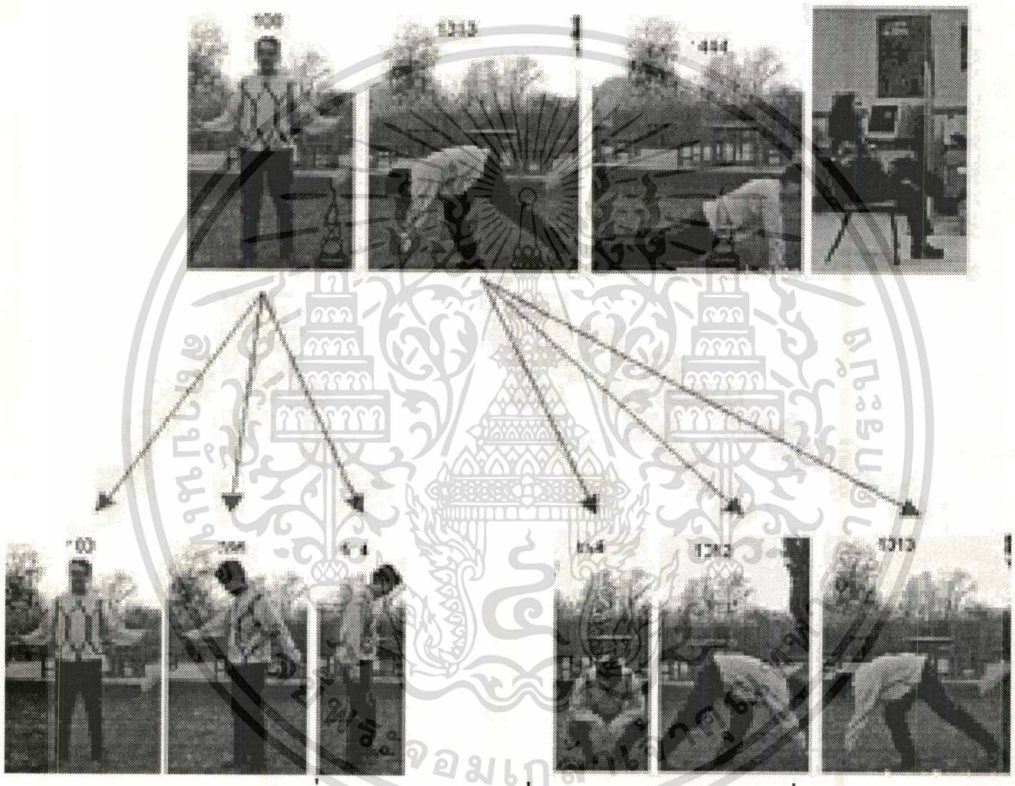
- ความสัมพันธ์ของตำแหน่งของร่างกายมนุษย์ในแต่ละท่าทางที่คล้ายคลึงกัน หรือ ท่าทางท่าเดียวกัน ตำแหน่งของ ศีรษะ, ลำตัว, มือ และ เท้า ทั้งสองบนเส้นขอบเขตโครงร่าง (Silhouette Boundary) จะอยู่ลำดับเดียวกันดังภาพที่ 2.3



ภาพที่ 2.3 ตัวอย่างของลำดับของส่วนของร่างกายบนเส้นขอบเขตโครงร่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตำแหน่งของร่างกายลำดับของส่วนต่างๆบนขอบเขตโครงร่าง (Silhouette Boundary) มักไม่มีการเปลี่ยนแปลงนักเมื่อมนุษย์ทำท่าทางทั่วๆไป เช่น ท่าเดิน ท่านั่ง แต่อย่างไรก็ดีลำดับก็จะเปลี่ยนแปลงเมื่อมีการเปลี่ยนท่าทาง เช่น จากเดินเปลี่ยนเป็นท่านั่ง ซึ่งท่าทางสามารถแบ่งได้เป็นท่าหลักๆคือท่า ยืน, นั่ง, คลาน (ท่าโก้งโค้ง) และ ท่านอน (หมอบ) และในแต่ละท่าทางสามารถแบ่งมุมมองได้อีก คือ หน้า, หลัง, ซ้าย และ ขวา ดังภาพที่ 2.4

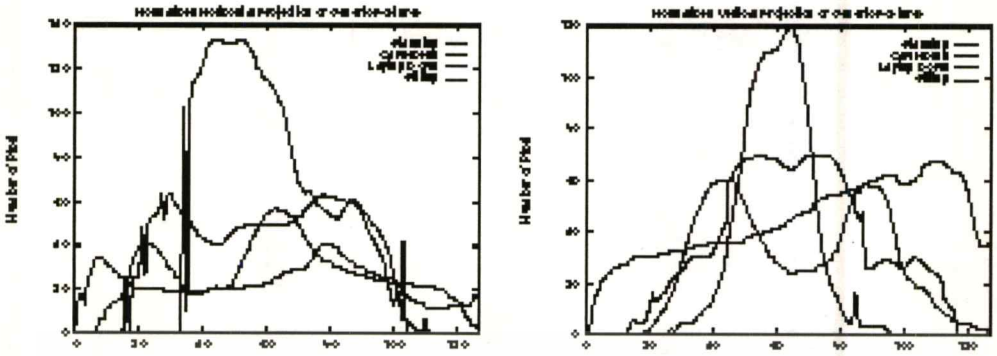


ภาพที่ 2.4 ท่าทางต่างๆเมื่อแสดงให้เห็นมุมมองที่ต่างๆ

จากข้อสังเกตดังกล่าว จึงแนะนำวิธีการในการหาตำแหน่งของส่วนประกอบของร่างกายโดยมีหลักการดังนี้

1. การวิเคราะห์ท่าทางมนุษย์แบบลำดับชั้น (Hierarchical Body Posture) คือ การนำโครงร่างมาคำนวณหาความคล้ายคลึงกับ Projection ของ Pixel ในภาพตามแนวนอนและแนวตั้ง (Horizontal and vertical Projection) เพื่อใช้ตรวจจับโครงร่างและท่าหลัก ท่าทางของร่างกายที่ได้ผลคล้ายคลึงที่สุดที่วัดได้ก็จะคาดเดาได้ว่าเป็นท่าทางนั้นดังภาพที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.5 Horizontal and vertical Projection ที่ทำการ normalized แล้วของท่า ยืน, คลาน (ท่าโก้งโค้ง) และ ท่านอน (หมอบ) ที่ใช้ในการคาดเดาท่าทาง

2. ใช้ Convex-hull algorithm ทำซ้ำๆเพื่อใช้ในการหาตำแหน่งที่น่าจะเป็นไปได้จากขอบเขตโครงร่าง (Silhouette Boundary)

3. หาตำแหน่งของศีรษะ ทำนายได้จาก แกนหลักของโครงร่าง, จุดตัดส่วนเว้า (Hull vertices) และ Topology ของส่วนของร่างกายมนุษย์ที่คาดเดาไว้

4. เมื่อได้ตำแหน่งที่แน่นอนของศีรษะ แล้วจึงทำการวิเคราะห์รูปทรง (Topological Analysis) เพื่อคาดเดาจุดตัดส่วนเว้า (Hull vertices) ที่ยังไม่พบมาจับคู่กับจุดตัดส่วนเว้า (Hull vertices) ที่ยังเหลืออยู่ โดยใช้ ลำดับรูปทรง (Topological-Order) คำนวณหาระยะห่างที่เปลี่ยนแปลง

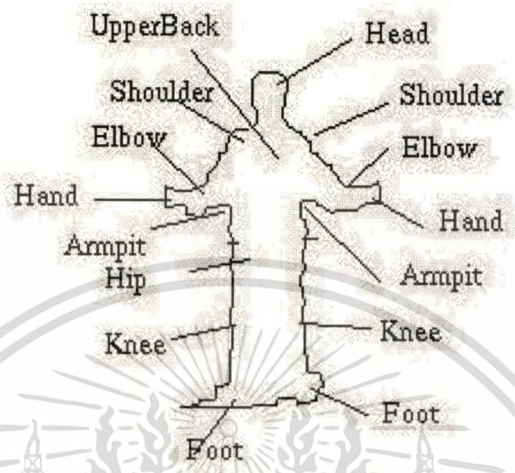
ซึ่งแต่ละขั้นตอนในการตรวจจับส่วนต่างๆของร่างกายจะอธิบายโดยละเอียดดังนี้ การตรวจจับส่วนต่างๆของร่างกาย หลังจากที่ลบพื้นหลังแล้วมีขั้นตอนดังนี้

- สร้าง แบบจำลอง 2 มิติโดยใช้ โครงร่าง (Silhouette) ในระบบนี้ จะใช้เทคนิคการทำแบบจำลองร่างกายโดยใช้ Silhouette-base ซึ่งประกอบด้วย 6 ส่วนประกอบหลักของร่างกาย (ศีรษะ, มือ, เท้า, ลำตัว) และส่วนประกอบย่อย 11 ส่วน (ข้อศอก, เข่า, ไหล่, รักแร้, ตะโพก และ Upper Back) ซึ่งส่วนประกอบย่อยนี้จะช่วยในการกำหนดตำแหน่งส่วนประกอบหลักได้ดังภาพที่ 2.3 ซึ่งในความเป็นจริงไม่สามารถหาส่วนต่างๆได้ครบทุกส่วนได้จากเฟรมเดียวเพราะส่วนประกอบบางส่วนไม่สามารถมองเห็นได้ในบางท่าทาง

จากข้อสังเกตที่ลำดับของตำแหน่งของส่วนต่างๆจะไม่เปลี่ยนแปลง จากภาพที่ 2.6 หากเริ่มจาก ศีรษะ เรียงลำดับตามเข็มนาฬิกาจากภาพท่ายืนจะได้ลำดับดังนี้ ศีรษะ-หัวไหล่-ข้อศอก-มือ-รักแร้-เข่า-เท้า-เท้า-เข่า-รักแร้-มือ-ข้อศอก-หัวไหล่-ศีรษะ ซึ่งในแต่ละท่าทางจะไม่มีการสลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งของลำดับ เช่น ไม่สามารถมีลำดับของ มือ-เท้า-หัวเข่า ได้ในท่าทางปกติ แต่จะสามารถข้ามตำแหน่งได้และอาจจะมีบางส่วนหายไปได้ เช่น ศีรษะ-หัวไหล่-มือ ได้



ภาพที่ 2.6 การแสดงลำดับของตำแหน่งของส่วนต่างๆ

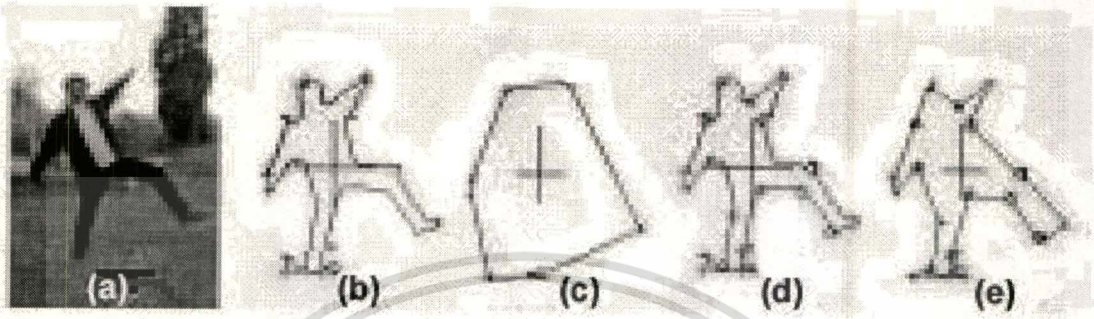
- การคาดเดาท่าทาง มนุษย์ สามารถมีท่าทางได้หลายท่าทางในการกระทำในแต่ละการกระทำ และในแต่ละท่าทางยังสามารถได้หลายมุมมองขึ้นกับองศาที่เห็น ซึ่งในการทำงานของระบบนี้จะใช้การเก็บตัวอย่างท่าทางจาก ระยะไกลจากโครงร่างมนุษย์ (Silhouette) เพื่อค้นหาลำดับของส่วนของร่างกายในท่าทางหลัก (ยืน, นั่ง, คลาน (ท่าโก้งโค้ง), ท่านอน (หมอบ)) โดยแบ่งแยกประเภทโดยสร้างแบบโครงร่างแบบลำดับชั้นดังภาพที่ 2.4 ซึ่งจะได้ท่าทาง 4 หลักและในแต่ละท่าหลักจะสามารถแบ่งมุมมองได้อีกคือ หน้าตรง ด้านซ้าย ด้านขวา

ท่าทางต่างๆจะสามารถแสดงได้ในรูปของProjection ของ Pixel ในภาพตามแนวนอนและแนวตั้ง (Horizontal and vertical Projection) ที่ทำการ normalized แล้วดังภาพที่ 2.5 ในการตรวจจับภาพจากวีดีโอก็จะนำแต่ละเฟรมไปคำนวณหาค่า Projection ของ Pixel ในภาพตามแนวนอนและแนวตั้ง (Horizontal and vertical Projection) แล้วนำมาเทียบกับกราฟเดิมที่มีอยู่ว่ามีค่าใกล้เคียงกับกราฟท่าทางใด ก็จะคาดเดาว่าเป็นท่าทางนั้น

- การตรวจสอบ ส่วนที่ยื่นออกมา (Convex hulls) ส่วนที่เว้าเข้าข้างใน (Concave hulls) ของโครงร่างมนุษย์ (Silhouettes) โดยใช้ Graham scan convex-hull algorithm มาปรับปรุงสามารถที่จะใช้ประโยชน์ของคุณสมบัติของโครงร่างและการคำนวณให้เร็วขึ้น ซึ่งการใช้การคำนวณส่วนที่ยื่นออกมา (Convex hulls) โดยใช้วิธีเดียวไม่พอที่จะใช้ในการตรวจสอบโครงร่างได้ จึงต้องใช้หลายวิธีการเพื่อที่จะตรวจสอบได้ดีขึ้น โดยลำดับแรกหาส่วนที่ยื่นออกมาสูงสุด และ เชื่อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดสูงสุดนั้นโดยใช้ algorithm ที่พัฒนาขึ้นเพื่อนำไปหาส่วนเว้าเข้าข้างใน (Concave hulls) และทำซ้ำอีก ดังภาพที่ 2.7



ภาพที่ 2.7 ท่ายืน Input Image (a), convex-hull vertices (b), เส้นเชื่อมระหว่าง convex-hull vertices (c), convex และ concave hull vertices (d), และ เส้นเชื่อมจุด convex และ concave hull vertices (e)

- การคาดเดาตำแหน่งของส่วนต่างๆ โดยหลักการคิดจำเป็นที่จะต้องค้นหาส่วนหนึ่งในโครงร่างเพื่อเป็นจุดเริ่มต้นในการหาส่วนอื่นๆ ในส่วนนี้จะการคาดเดาตำแหน่งของศีรษะเป็นลำดับแรก และ การคาดเดาตำแหน่งของ เท้า มือ และ ลำตัวเป็นลำดับต่อไป

การคาดเดาส่วนที่เป็นศีรษะ ใช้หลักการของความสัมพันธ์ระหว่างตำแหน่งของร่างกายจากโครงร่างที่ได้ทำการคาดเดาท่าทางแล้ว กับ แกนหลักของโครงร่าง และ ข้อมูลของระนาบพื้นเพื่อใช้ในการคาดเดาส่วนย่อยของโครงร่างที่อยู่ตรงข้ามกับศีรษะ

การคาดเดาตำแหน่งของ เท้า มือ และ ลำตัว หลังจากที่ทำการคาดเดาตำแหน่งของศีรษะแล้วก็จะใช้การคาดเดาส่วนอื่นๆจากลำดับของตำแหน่งของส่วนต่างๆในแต่ละท่าทางที่ทำเอาไว้โดยใช้ข้อกำหนดของลำดับของตำแหน่งของส่วนของร่างกาย และ ความสัมพันธ์ของระยะระหว่างส่วนต่างๆ

สรุป มีใช้หลักการ ของการวิเคราะห์ท่าทางมนุษย์แบบลำดับชั้น (Hierarchical Body Posture) ท่าทางของร่างกายที่ได้ผลคล้ายคลึงที่สุดที่วัดได้ก็จะคาดเดาได้ว่าเป็นท่าทางใดแล้วนำไปวิเคราะห์ลำดับของตำแหน่งของส่วนต่างๆจะไม่เปลี่ยนแปลงในท่าทางเดียวกันนำมาการคาดเดาตำแหน่งของส่วนต่างๆ โดยหลักการคิดจำเป็นที่จะต้องค้นหาส่วนหนึ่งในโครงร่างเพื่อเป็นจุดเริ่มต้นในการหาส่วนอื่นๆ ในส่วนนี้จะการคาดเดาตำแหน่งของศีรษะเป็นลำดับแรก และ การคาดเดาตำแหน่งของ เท้า มือ และ ลำตัวเป็นลำดับต่อไป วิธีนี้เป็นวิธีการที่มีความสามารถในการระบุท่าทางและตำแหน่งของส่วนประกอบของร่างกายมนุษย์ได้ในหลายท่าทางแต่จำเป็นต้องเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ท่าทางต่างๆก่อนและมีโอกาสผิดพลาดสูงขึ้นถ้าขั้นตอนแรกที่ระบุว่าเป็นท่าทางใดระบุไม่ตรงกับท่าทางจริง

1.3.2 การตรวจจับและติดตามส่วนของร่างกายโดยใช้ Cardboard Model ใน W4

W4 เป็นระบบลาดตะเวนเพื่อใช้ในการตรวจจับและติดจามการเคลื่อนไหวของมนุษย์จากวีดีโอ พัฒนาโดย UNIVERSITY OF MARYLAND เมื่อปี 1998

หลักการการทำงานของ W4 กล่าวคือ W4 จะทำการตรวจจับและติดตามวัตถุที่เคลื่อนไหวใน Scene โดยทำการตรวจจับและติดตามโครงร่างโดยรวม (Whole silhouette region) ของวัตถุนั้นๆ ในขณะเดียวกันก็จะทำการตรวจจับและติดตามส่วนประกอบของร่างกายด้วยถ้าวัตถุที่ตรวจพบนั้นเป็นมนุษย์

ก่อนอื่นจะกล่าวถึงหลักการในการตรวจจับและติดตามโครงร่างโดยรวม (Whole silhouette region) ของแต่ละวัตถุที่เคลื่อนไหวใน Scene ก่อน จากนั้นจะกล่าวถึงเทคนิคที่ W4 ใช้ในการตรวจจับและติดตามส่วนของร่างกายในกรณีที่ ภาพโครงสร้างนั้นเป็นร่างกายมนุษย์โดยใช้ Card Board Model ใน

การตรวจจับและติดตามโครงร่างที่เคลื่อนไหวที่ปรากฏใน Scene นั้นมีขั้นตอนดังนี้

- หาว่าวัตถุใหม่เข้ามาในมุมมองของระบบเมื่อไหร่ และเริ่มสร้างตัวต้นแบบการเคลื่อนไหวสำหรับการติดตามการเคลื่อนไหววัตถุนั้น
- คำนวณความสัมพันธ์ระหว่างบริเวณโครงร่างที่ตรวจพบโดยการลบพื้นหลัง และ วัตถุที่ถูกติดตามการเคลื่อนไหวโดย W4
- ใช้กระบวนทำงานของการติดตามการเคลื่อนไหวเพื่อประมาณตำแหน่ง (ของร่างกาย) ของวัตถุแต่ละโครงร่าง และทำการอัพเดทตัวต้นแบบการเคลื่อนไหวที่ใช้ในการติดตามการเคลื่อนไหว W4 ใช้โครงร่างต้นแบบการเคลื่อนไหวแบบอนุพันธ์อันดับสอง (รวมไปถึง ความเร็ว และ ค่าความเร่ง ที่อาจเป็นศูนย์) ในการสร้างโครงร่างต้นแบบของการเคลื่อนไหวทั้งโครงร่างและการเคลื่อนไหวของส่วนต่างๆของร่างกาย

W4 จะติดตามการเคลื่อนไหววัตถุต่อไปถึงแม้ว่าความสามารถในการตรวจจับจะอยู่ในระดับต่ำไม่สามารถแบ่งผู้คนเป็นวัตถุจากโครงร่างโดยรวม(Whole silhouette region)ได้ ซึ่งอาจจะเกิดขึ้นเนื่องจากวัตถุถูกบดบังชั่วคราวในแต่ละเฟรม หรือ วัตถุแยกเป็นหลายชิ้น (อาจจะเนื่องมาจากผู้คนเอาถือวัตถุเข้ามาในฉากใน หรือ คนถูกปิดบังโดยวัตถุชิ้นเล็กๆ) ในที่สุด วัตถุที่ระบบติดตามการเคลื่อนไหวที่แยกกันอยู่ก็อาจจะมารวมเป็นชิ้นเดียวกันเนื่องจากการปฏิสัมพันธ์กันระหว่างบุคคล ภายใต้เงื่อนไขนี้การวิเคราะห์รูปร่างโดยรวมและกระบวนการทำงานการติดตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเคลื่อนไหวโดย W4 อาจจะเกิดข้อผิดพลาด ระบบก็จะใช้เทคนิคความสัมพันธ์ในการพยายามที่จะติดตามการเคลื่อนไหวของวัตถุที่ปฏิสัมพันธ์กัน

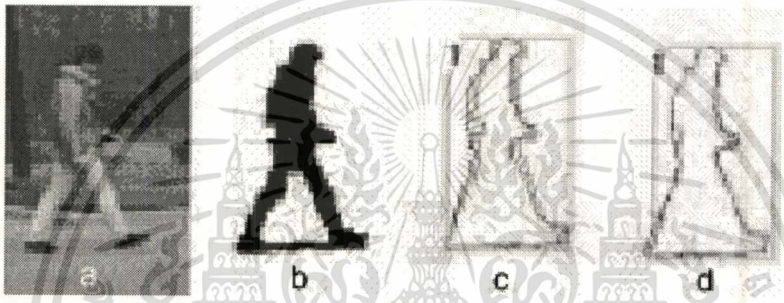
อันดับแรก W4 จะจับคู่วัตถุกับบริเวณโครงร่างโดยการหาบริเวณที่ซ้อนทับกันระหว่างกรอบที่ประมาณค่าโดยใช้ต้นแบบการเคลื่อนไหวโดยรวมของวัตถุ และ กรอบของบริเวณโครงร่างจากเฟรมปัจจุบัน สำหรับวัตถุแต่ละโครงร่าง บริเวณโครงร่างทั้งหมด ซึ่งกรอบซ้อนทับกันมากพอที่จะเหมาะนำมาจับคู่กับวัตถุนั้น ในทางอุดมคติ การจับคู่แบบหนึ่งต่อหนึ่ง (การติดตามการเคลื่อนไหว) จะถูกพบในระหว่างการติดตามการเคลื่อนไหวหนึ่งๆ อย่างไรก็ตาม การจับคู่แบบหนึ่งต่อจำนวนมาก (วัตถุที่ถูกติดตามการเคลื่อนไหวถูกแบ่งแยกเป็นบริเวณโครงร่างหลายอัน) การจับคู่แบบจำนวนมากกับหนึ่ง (วัตถุที่ถูกติดตามการเคลื่อนไหวรวมโครงร่างกันเป็นบริเวณโครงร่างเพียงหนึ่งบริเวณ) การจับคู่แบบหนึ่งกับศูนย์ (การหายไป) และ การจับคู่แบบศูนย์กับหนึ่ง (การปรากฏขึ้น) อาจจะเกิดขึ้นอยู่บ่อยๆ W4 จะใช้การติดตามการเคลื่อนไหวโดยใช้วิธีที่ต่างกันสำหรับการจับคู่แต่ละแบบ

1. การที่วัตถุปรากฏขึ้น (Appearing Objects) เมื่อบริเวณโครงร่าง ถูกตรวจพบ เราไม่สามารถบอกได้เลยทีเดียวว่ามันเป็นวัตถุจริงหรือแค่สัญญาณรบกวนเท่านั้น ถ้าบริเวณนั้นสามารถถูกติดตามการเคลื่อนไหวได้ในหลายๆเฟรม มันก็จะถูกเพิ่มเข้าไปในรายการวัตถุที่จะต้องถูกติดตามการเคลื่อนไหวด้วย

2. การติดตามการเคลื่อนไหว (Tracking) ส่วนนี้ เราจะพิจารณาสถานการณ์ที่วัตถุถูกติดตามการเคลื่อนไหวที่เป็นโครงร่างเพียงโครงร่างเดียว W4 ใช้รูปแบบการเคลื่อนไหวแบบอนุพันธ์อันดับสองสำหรับวัตถุแต่ละโครงร่างในการประมาณตำแหน่งในเฟรมต่อมา การพยากรณ์จากโครงร่างต้นแบบนี้ถูกนำมาใช้ในการประมาณหาตำแหน่งของกรอบสำหรับวัตถุแต่ละโครงร่าง ตำแหน่งของกรอบที่ถูกพยากรณ์ก็จะถูกนำมาเปรียบเทียบกับกรอบจริงของบริเวณโครงร่างที่ตรวจจับได้ กำหนดให้วัตถุถูกจับคู่กับบริเวณโครงร่างโครงร่างเดียว (และขนาดของบริเวณเหล่านั้นก็เกือบเท่ากัน) W4 ต้องหาตำแหน่งปัจจุบันของวัตถุเพื่อที่จะทำการอัพเดตโครงร่างการเคลื่อนไหวต้นแบบ แม้ว่าการเคลื่อนไหวทั้งหมดของวัตถุจะค่อนข้างน้อยในแต่ละเฟรม สำหรับโครงร่างที่มีเปลี่ยนแปลงมาก(เคลื่อนไหวมาก)จะใช้เทคนิคง่ายๆ เช่น การติดตามจุดศูนย์กลางรวมของบริเวณโครงร่างผิดพลาดได้ แทนที่จะให้เป็นแบบนั้น W4 ใช้วิธีจับคู่แบบสองชั้นเพื่อที่จะอัพเดตการประมาณตำแหน่งโดยรวมของวัตถุ การประมาณตำแหน่งตอนเริ่มต้นถูกคำนวณเป็นการเคลื่อนไหวของจุดพิกัดมัธยฐาน (median) ของวัตถุ ค่าพิกัดมัธยฐาน (median) นี้เป็นการประมาณตำแหน่งของวัตถุที่มีประสิทธิภาพมากกว่า และไม่ถูกกระทบโดยการเคลื่อนไหวมากๆ (ซึ่งมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวโน้มที่จะกระทบจุดศูนย์กลางรวมอย่างมาก) มันทำให้เราบีบการค้นหาที่ว่างสำหรับการเคลื่อนไหวของวัตถุ อย่างไรก็ตาม การประมาณนี้ไม่เพียงพอสำหรับการติดตามการเคลื่อนไหวในระยะยาว ดังนั้นหลังจากที่ย้ายภาพตัดขวางของวัตถุจากเฟรมที่ผ่านมาโดยใช้การประมาณโดยใช้ median-base เราใช้ความสัมพันธ์ของขอบแบบฐานสองระหว่างโครงร่างปัจจุบันและโครงร่างก่อนหน้านี้ ความสัมพันธ์นี้ถูกคำนวณโดยใช้ชุดของจุดตำแหน่ง 5x3 ความสัมพันธ์จะถูกครอบครองโดยขอบของร่างกายและหัว ซึ่งรูปร่างของมันจะเปลี่ยนไปอย่างช้าๆจากเฟรมหนึ่งไปยังอีกเฟรมหนึ่ง กระบวนการติดตามความเคลื่อนไหวได้ แสดงดัง ภาพที่ 2.8



ภาพที่ 2.8 การคาดเดาการเคลื่อนไหวของร่างกายโดยใช้โครงร่างจับคู่กันระหว่าง 1 เฟรมที่เลือกมา ภาพที่เข้ามา (a), การตรวจจับโครงร่าง (b), alignment ของขอบโครงร่างโดยใช้ความแตกต่างของ Median (c), alignment สุดท้ายหลังจากทำความสัมพันธ์โครงร่าง (d)

3. การแบ่งแยกบริเวณ (Region splitting) วัตถุที่ถูกติดตามการเคลื่อนไหวอาจจะแบ่งเป็นบริเวณโครงร่างหลายๆบริเวณ ไม่ว่าจะเป็นการบดบังเพียงบางส่วน หรือ เพราะว่าคนได้นำวัตถุเข้ามาในฉากด้วย ในกรณีนี้ โครงร่างหนึ่งจะถูกจับคู่กับบริเวณสองบริเวณหรือมากกว่านั้น W4 จะหาว่าการแบ่งแยกนี้เป็นการแบ่งแยกของจริงหรือหลอก (เนื่องมาจากสัญญาณรบกวน) โดยการสังเกตดูเฟรมต่อมา ในขณะที่ติดตามดูวัตถุที่ถูกแบ่งแยกออกมาเป็นวัตถุแต่ละโครงร่าง ถ้า W4 สามารถติดตามดูวัตถุย่อยๆได้ในหลายๆเฟรม แล้วเราสามารถสรุปได้ว่าพวกมันเป็นวัตถุแต่ละโครงร่างและเริ่มติดตามดูความเคลื่อนไหวของมันแต่ละโครงร่าง

4. การรวมบริเวณ (Region merging) ในส่วนนี้จะพิจารณาสถานการณ์ที่วัตถุถูกติดตามการเคลื่อนไหวที่เป็นโครงร่าง 2 โครงร่างหรือ คน 2 คน เมื่อคนสองคนพบกัน(เดินสวนกัน) พวกเขาจะถูกแบ่งเป็นบริเวณโครงร่างเพียงหนึ่งโครงร่างโดยกระบวนการทำงานการลบพื้นหลัง W4 จำได้ว่านี่มันเกิดขึ้นโดยมีพื้นฐานมาจากการวิเคราะห์อย่างง่ายของกรอบที่ถูกพยากรณ์ของวัตถุที่เราติดตามการเคลื่อนไหวและกรอบของบริเวณโครงร่างที่รวมแล้ว บริเวณที่รวมแล้วจะถูกจับตาดูความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เคลื่อนไหวจนกระทั่งมันแยกกลับเป็นวัตถุย่อย เนื่องจากภาพตัดขวางของบริเวณมีแนวโน้มที่จะเปลี่ยนรูปอย่างรวดเร็วและไม่สามารถคาดเดาได้ W4 จึงใช้วิธีเทียบค่าอย่างง่ายในการสร้างโครงร่างต้นแบบการพยากรณ์การเคลื่อนไหวสำหรับวัตถุที่ได้รวมมานี้ ปัญหาที่เกิดขึ้นเมื่อบริเวณที่รวมกันได้แบ่งแยกออก และ คนปรากฏโครงร่างขึ้น ก็คือการหาความสัมพันธ์ระหว่างคนที่ถูกติดตามการเคลื่อนไหวก่อนที่จะมีการปฏิสัมพันธ์และคนที่ได้มารวมกันจากการปฏิสัมพันธ์ ในการแก้ปัญหานี้ W4 ใช้โครงร่างต้นแบบการปรากฏสองชนิดที่สร้างขึ้นในขณะที่มันกำลังติดตามดูการเคลื่อนไหวของคนที่แยกออกมา

W4 ได้สร้างแผ่นเทมเพลทเชิงเวลา หรือเรียกว่า temporal texture template เป็นโครงร่างต้นแบบการเคลื่อนไหว temporal texture template สำหรับวัตถุสามารถหาได้จากสูตร

$$\Psi^t(x,y) = \frac{I(x,y) + w^{t-1}(x,y) \times \Psi^{t-1}(x,y)}{w^{t-1}(x,y) + 1}$$

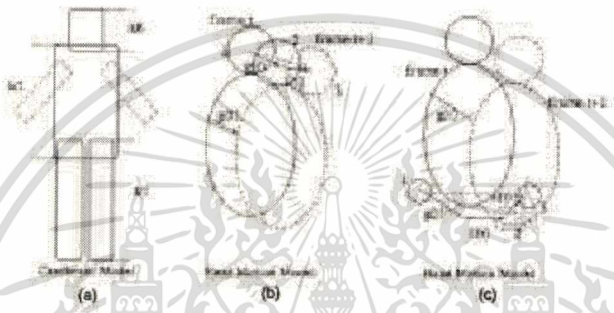
ในที่นี้ ขออ้างถึงบริเวณโครงร่างที่ถูกตรวจพบในระหว่างการติดตามการเคลื่อนไหววัตถุและจุดพิกัดก็ถูกแสดงโดยสัมพันธ์กับค่า มัธยฐาน (median) ของแผ่นเทมเพลทหรือบริเวณโครงร่าง น้ำหนักจากสูตร ก็คือความถี่ที่ ค่าพิกเซล ใน Ψ ถูกตรวจพบเป็นพิกเซลของโครงร่างในระหว่างการติดตามการเคลื่อนไหว ค่าน้ำหนักเริ่มต้น $w^t(x,y)$ ของ Ψ มีค่าเป็นศูนย์และจะเพิ่มขึ้นในแต่ละครั้งที่ตำแหน่งที่สัมพันธ์กับค่าพิกัดของค่ามัธยฐาน (median)ของเทมเพลทถูกตรวจพบในลักษณะของพิกเซลของโครงร่างในภาพของ Input

หลังจากแยกออกจากกัน วัตถุย่อยก็ถูกจับคู่กับวัตถุที่แยกออกจากกันโดยทำให้ temporal templates ของพวกมันสัมพันธ์กัน เนื่องจาก temporal texture template ได้มาจากการดู มันอาจจะไม่สามารถจับคู่ได้ถ้ามันมีการเปลี่ยนแปลงการขยับของวัตถุมากในระหว่างเหตุการณ์บดบัง ดังนั้นวิธีที่ไม่ได้ใช้การดูซึ่งใช้การแสดงผลวัตถุเป็นสัญลักษณ์ ก็ได้นำมาใช้ในการวิเคราะห์การบดบังนี้ด้วย ยกโครงร่างอย่าง เช่น ถ้า temporal texture templates ไม่สามารถให้ค่าความสัมพันธ์ที่สูงพอ ดังนั้นเมื่อเราจับคู่วัตถุโดยใช้ความเข้มเฉลี่ยในส่วนบน ล่าง และกลาง ในการพยายามที่จะระบุวัตถุเมื่อมันแยกออกไปแล้ว

5. การติดตามการเคลื่อนไหวของส่วนต่างๆของคน (Tracking People's Parts) ในส่วนเพิ่มเติมของการติดตามการเคลื่อนไหวของคนทั้งหมด เราต้องการที่จะระบุตำแหน่งของส่วนต่างๆของร่างกาย เช่น ศีรษะ ร่างกาย ขา และ เท้า และ ติดตามการเคลื่อนไหวของมันเพื่อที่จะเข้าใจการกระทำ W4 ใช้การวิเคราะห์ร่างกายร่วมกับการจับคู่เท็มเพลทเพื่อที่จะดูการเคลื่อนไหวของส่วนต่างๆนี้ (เมื่อคนถูกบัง และรูปร่างของมันไม่สามารถพยากรณ์ได้ง่ายนัก เราจะใช้แค่การจับคู่เท็มเพลทในการดูการเคลื่อนไหวของส่วนต่างๆของร่างกาย) โครงร่างต้นแบบของรูปร่างนำมาใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

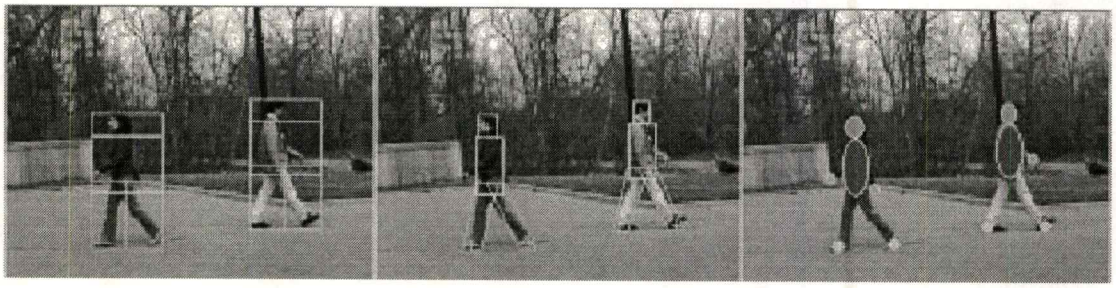
โดยการใช้ Cardboard Model ซึ่งแสดงถึงตำแหน่งและขนาดสัมพัทธ์ของร่างกายและส่วนต่างๆ ของมัน เมื่อใช้งานร่วมกับโครงร่างต้นแบบการพยากรณ์การเคลื่อนไหวแบบอนุพันธ์อันดับสอง Cardboard Model สามารถนำมาใช้ในการพยากรณ์หาตำแหน่งของส่วนต่างๆของร่างกายในแต่ละเฟรมได้ ภาพที่ 2.9 ได้แสดงถึงโครงร่างต้นแบบการเคลื่อนไหวที่ใช้สำหรับมือและศีรษะ ตำแหน่งเหล่านี้ถูกระบุและกลั่นกรองโดยใช้การจับคู่เท็มเพลทที่คงที่จาก temporal texture templates ของส่วนของร่างกายที่เราสังเกต



ภาพที่ 2.9 Cardboard model ที่ใช้ใน W^4 (a), และ แบบจำลองที่เครื่องไหวที่ใช้สำหรับศีรษะ (b) และ มือ(c)

Cardboard model เป็นแบบจำลองร่างกายมนุษย์ในท่ายืน ดังรูป 2.9(a) มันถูกนำมาใช้ในการพยากรณ์ค่าตำแหน่งของส่วนต่างๆของร่างกาย (ศีรษะ ร่างกาย เท้า มือ ขา) ความสูงของกรอบของวัตถุก็คือความสูงของ cardboard model จากนั้นเราก็ใช้ค่าขนาดที่คงที่ในแนวตั้งการหาตำแหน่งเริ่มต้น (กรอบ) ของส่วนของร่างกายแต่ละส่วน ดังภาพที่ 2.10 ความยาวของกรอบตอนเริ่มต้นของศีรษะ ร่างกายและขาจะถูกคำนวณเป็น 1/5, 1/2 และ 1/2 ของความยาวของกรอบของวัตถุตามลำดับ ความกว้างของกรอบของศีรษะ ร่างกายและขาถูกคำนวณโดยการหาความกว้างมัธยฐาน (median) ภายในกรอบตอนเริ่ม นอกจากการหาขนาดและตำแหน่ง ค่าโมเมนต์ของค่าพิกเซลของโครงร่างภายในกรอบก็ต้องถูกคำนวณเพื่อประมาณค่าแกนหลักของมัน แกนหลักนี้จะให้ข้อมูลเกี่ยวกับท่าทางของส่วนต่างๆของร่างกาย ศีรษะจะถูกหาก่อน ตามมาด้วย ร่างกายและ ขา มือจะถูกหาหลังจากร่างกายโดยการหาบริเวณที่เข้มข้นที่เชื่อมต่อกับร่างกายและอยู่ข้างนอกร่างกาย เท้าก็จะถูกพบในบริเวณเข้มในทิศทางของแกนหลักของขา ภาพที่ 2.10 ได้แสดงถึงโครงร่างอย่างของวิธีนำ cardboard model ไปใช้ในการพยากรณ์ตำแหน่งของส่วนต่างๆของร่างกายในสองขั้นตอน (การประมาณตำแหน่งตอนเริ่มและตำแหน่งสุดท้ายที่ถูกประมาณ) และแสดงมันในแบบที่ละเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a) (b) (c)

ภาพที่ 2.10 เป็นโครงร่างอย่างการใช้ Cardboard Model แสดงการบอกตำแหน่งส่วนต่างๆ ของร่างกาย (a) แสดงค่าเริ่มต้นของกรอบบอกตำแหน่งขอบเขตของโครงร่าง, (b) ใช้ Cardboard Model วิเคราะห์ตำแหน่งของส่วนของร่างกาย, (c) แสดงการระบุตำแหน่งของร่างกาย (ศีรษะ, ลำตัว, มือทั้งสองข่าง และ เท่าทั้งสองข้าง)

หลังจากที่พยากรณ์ค่าตำแหน่งของศีรษะและมือโดยใช้ cardboard model แล้ว , ตำแหน่งของพวกมันก็จะถูกระบุและกลั่นกรองโดยใช้ temporal texture templates , temporal texture templates เหล่านี้ก็จะถูกอัพเดทดังที่ได้อธิบายไปก่อนหน้านี้ เว้นเสียแต่ว่ามันถูกพบในรูปด้านของของร่างกาย ในกรณีนี้ ค่าพิกเซลที่เกี่ยวข้องกับศีรษะและมือจะถูกฝังอยู่ในส่วนประกอบที่ใหญ่กว่าที่เกี่ยวข้องกับร่างกาย นี้ทำให้มันยากในการที่ประมาณตำแหน่งมัธยฐาน (median) ของแต่ละส่วนได้อย่างถูกต้องหรือหาว่าพิกเซลไหนที่อยู่ในร่างกายที่เป็นพิกเซลของส่วนอื่น ในกรณีเหล่านี้ ส่วนต่างๆของร่างกายก็จะถูกติดตามการเคลื่อนไหวโดยใช้การสัมพันธ์กัน แต่แผ่นเทมเพลทจะไม่ถูกอัพเดท

สรุป หลักการการทำงานของ W4 กล่าวคือ W4 จะทำการตรวจจับและติดตามวัตถุที่เคลื่อนไหวใน Scene โดยทำการตรวจจับและติดตามโครงร่างโดยรวม (Whole silhouette region) ของวัตถุนั้นๆ ในขณะเดียวกันก็จะทำการตรวจจับและติดตามส่วนประกอบของร่างกายด้วยถ้าวัตถุที่ตรวจพบนั้นเป็นมนุษย์ใช้หลักการ template-matching ในการติดตามส่วนต่างๆและใช้แบบจำลอง Cardboard model ในการพยากรณ์ค่าตำแหน่งของส่วนต่างๆของร่างกาย (ศีรษะ ร่างกาย เท้า มือ และ ขา) โดยความสูงของกรอบของวัตถุก็คือความสูงของ Cardboard model จากนั้นเราก็ใช้ค่าขนาดที่คงที่ในแนวตั้ง การหาตำแหน่งเริ่มต้นของส่วนของร่างกายแต่ละส่วน โดยความยาวของกรอบ ศีรษะ ร่างกายและขาจะถูกคำนวณเป็น 1/5, 1/2 และ1/2 ของความยาวของกรอบของวัตถุตามลำดับ วิธีนี้เป็นวิธีการที่มีความเร็วในการตรวจจับและติดตามสูง และสามารถที่จะตรวจจับมนุษย์ได้มากกว่า 1 คน แต่มีข้อจำกัดที่ใช้งานได้เฉพาะในท่ายืน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4. หลักการเขียนภาษาเชิงวัตถุ และการสร้างโปรแกรมด้วย MS Visual C++

การเขียนโปรแกรมแบบ Object Oriented Programming (OOP) เป็นแนวคิดในการเขียนโปรแกรมแบบหนึ่ง โดยจะไม่มุ่งเน้นไปในการเขียนโปรแกรมเพียงอย่างเดียว แต่เป็นการมองภาพรวมของสิ่งที่อยู่รอบๆ ตัวเราโดยสามารถให้คำจำกัดความได้ดังนี้ "วัตถุแต่ละอย่างต่างก็มีลักษณะ และวิธีการใช้งานเป็นของตัวมันเอง" กล่าวคือ วัตถุแต่ละชนิดแต่ละอัน ต่างก็มีรูปร่างลักษณะ และการใช้งานที่แตกต่างกันออกไป เราจะเรียกคุณลักษณะของวัตถุนี้ว่า Attribute และเราจะเรียกวิธีการใช้งานวัตถุว่า Method ยกตัวอย่างเช่น จักรยานคันหนึ่งมีสีแดง สามารถใช้ขับขี่ได้ สีแดงจัดว่าเป็น Attribute ของจักรยาน ในขณะที่การใช้งานขับขี่ถือเป็น Method เป็นต้น

จากที่กล่าวมาข้างต้นจะเห็นได้ว่าการเขียนโปรแกรมแบบ OOP เป็นแนวคิดที่คล้ายธรรมชาติของสิ่งของสิ่งหนึ่งซึ่งเราสามารถแบ่งแยกสิ่งต่างๆ ออกเป็นประเภทๆ ได้ ถ้าเราได้นำเอาแนวคิดของ OOP มาใช้ในการเขียนโปรแกรม และการจัดการข้อมูล เราจะพบว่าโปรแกรม หรือฟังก์ชั่น จะมีความเป็นอิสระต่อกันอย่างเห็นได้ชัด อธิบายง่ายๆ ก็คือ โปรแกรม หรือฟังก์ชั่นแต่ละตัวถึงแม้จะมาจากที่เดียวกัน แต่สามารถทำงานในคนละหน้าที่หรือเก็บข้อมูลคนละค่าได้

คลาสใน OOP คือกลุ่มของวัตถุที่คุณลักษณะพื้นฐาน และมีฟังก์ชั่นพื้นฐานเหมือนกัน เช่น คลาสคน กล่าวคือไม่ว่าคนนั้นจะมีชื่อใด มีอาชีพใด หรือมีส่วนสูงเท่าไหร่ ก็จัดอยู่ในคลาสคน ซึ่งจะมีคุณลักษณะ และฟังก์ชั่นพื้นฐานเหมือนกัน โดยผู้เขียนโปรแกรมไม่สามารถนำคลาสไปใช้งานได้โดยตรง ต้องมีการสร้าง Object (หรือ Instance) ขึ้นมาก่อน ซึ่ง Object ที่สร้างจากคลาสใดจะมีคุณลักษณะพื้นฐานมาจากคลาสนั้นด้วย เพราะฉะนั้นการสร้างโปรแกรมด้วยวิธีการแบบ OOP จะทำให้การสร้างโปรแกรมมีความยืดหยุ่น และมีประสิทธิภาพสูง

คุณสมบัติพิเศษอีกอย่างหนึ่งที่ทำให้การเขียนโปรแกรมในแบบ OOP คือคุณสมบัติ การสืบทอด (Inheritance) การที่ OOP นำแนวคิดนี้มาใช้ทำให้ผู้เขียนโปรแกรมสามารถนำเอา Source Code (อาจเป็นคลาส) ที่เขียนไว้แล้วสามารถนำกลับมาใช้ใหม่ (Code Reuse) หรือมาเปลี่ยนแปลงแก้ไขให้เหมาะสมกับงานเฉพาะอย่าง ทำให้การเขียนโปรแกรมเป็นไปได้อย่างรวดเร็วและเป็นการใช้ Source Code ที่มีอยู่ให้เกิดประโยชน์สูงสุด

OOP มีการกำหนดระดับการเข้าถึงข้อมูลภายในคลาส เพื่อความปลอดภัยในการเข้าถึงข้อมูลภายในคลาส ซึ่งการกำหนดระดับการเข้าถึงข้อมูลภายในคลาสนี้เป็นคุณสมบัติอีกอย่างที่มีในแนวคิดแบบ OOP การเข้าถึงข้อมูลภายใน (Member Variable) หรือฟังก์ชั่นภายใน (Member Function หรืออาจเรียกว่า Method) แบ่งออกได้ 3 ระดับ ดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

- ระดับ Private เป็นระดับที่จะถูกกำหนดไว้อัตโนมัติ (Default) หากผู้เขียนโปรแกรมไม่กำหนด ระดับนี้เป็นระดับการป้องกันสูงสุด กล่าวคือเป็นการป้องกันไม่ให้กระบวนการใดๆ ที่อยู่นอกคลาสเรียกใช้สิ่งที่อยู่ในคลาสไม่ได้ การจะเข้าถึงข้อมูลภายในจะต้องผ่านจากฟังก์ชั่นที่เป็นของคลาสนั้นอนุญาตให้ใช้ได้
- ระดับ Public เป็นระดับที่ตรงข้ามกับระดับ Private กล่าวคือกระบวนการใดๆ ที่อยู่นอกคลาสนั้นสามารถเข้าถึงข้อมูลภายใน หรือฟังก์ชั่นภายในได้อย่างอิสระ ไม่จำเป็นต้องได้รับอนุญาตก่อนการเข้าใช้งาน
- ระดับ Protected คล้ายกับในระดับ Private แต่มีข้อแตกต่างคือระดับ Protected นั้น คลาสลูกที่สืบทอดคุณลักษณะจากคลาสแม่ จะสามารถเข้าถึงข้อมูลภายในคลาสแม่ได้โดยตรง

ความสามารถในการกำหนดระดับการเข้าถึงข้อมูลภายในคลาสอาจเรียกว่า คุณสมบัติการห่อหุ้มวัตถุ(Data Hiding)

การเขียนโปรแกรมแบบ OOP ผู้เขียนโปรแกรมสามารถใช้คุณสมบัติของฟังก์ชั่นคอนสตรัคเตอร์ (Constructor) และ ดีสตรัคเตอร์ (Destructor) เพื่อเพิ่มความยืดหยุ่นให้กับโปรแกรมได้ ในการสร้างคลาสแต่ละครั้งเราไม่จำเป็นต้องสร้างคอนสตรัคเตอร์และดีสตรัคเตอร์ทุกครั้งไป การสร้างคอนสตรัคเตอร์และดีสตรัคเตอร์นั้นขึ้นอยู่กับความจำเป็นในแต่ละสถานการณ์และการดำเนินงานของคลาส และในคลาสหนึ่งสามารถมีคอนสตรัคเตอร์ได้หลายตัวโดยฟังก์ชั่นคอนสตรัคเตอร์ต้องมีชื่อฟังก์ชั่นเหมือนชื่อของคลาส และ ไม่มีการคืนค่าใดๆออกมาเลย ฟังก์ชั่นคอนสตรัคเตอร์จะถูกเรียกโดยอัตโนมัติเมื่อมีการสร้าง object และการสร้าง object สามารถผ่านพารามิเตอร์ (หรือไม่ผ่านก็ได้เป็น default) ซึ่ง complier จะเรียกฟังก์ชั่นคอนสตรัคเตอร์ได้ตรงกัน คุณสมบัตินี้เรียกว่า สามารถโอเวอร์โหลดฟังก์ชั่นได้

ยังมีคุณสมบัติ Dynamic Binding (Late Binding) และ Polymorphism ซึ่งเป็นคุณสมบัติพิเศษที่ทำให้การเขียนโปรแกรมแบบ OOP แตกต่างจากการเขียนแบบโครงสร้าง ซึ่งจะไม่ขอกล่าวในรายงานฉบับนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5. Microsoft Vision SDK

Microsoft Vision System Development Kit หรือที่จักกันในนามของ MS Vision SDK ถูกพัฒนาขึ้นโดย Vision Technology Research Group เพื่อใช้ในการสนับสนุนการพัฒนาโปรแกรมทางด้าน Image Processing, Real-time Image Processing และ Computer Vision บนระบบปฏิบัติการ Microsoft Windows

MS Vision SDK มีคลาสและฟังก์ชั่นในภาษา C++ สำหรับการทำงานที่เกี่ยวข้องกับงานด้านภาพ ผู้เขียนโปรแกรมที่ติดตั้ง MS Vision SDK สามารถพัฒนาโปรแกรม โดยใช้ MS Visual C++ ในชุด Visual Studio ได้อย่างอิสระ ไม่ขึ้นกับอุปกรณ์ต่อเชื่อมที่นำสัญญาณเข้าเครื่องคอมพิวเตอร์ เพราะ MS Vision SDK อาศัย Windows Driver Model (WDM) ที่ทำหน้าที่เป็นตัวกลางในการติดต่อระหว่าง Application Software กับอุปกรณ์ต่อเชื่อม ดังนั้นผู้เขียนโปรแกรมจึงสามารถใช้อุปกรณ์ที่เป็นตัวนำข้อมูลภาพ ทั้งภาพนิ่งและภาพเคลื่อนไหวมาใช้ได้หลากหลายรูปแบบ เช่น กล้องถ่ายรูปดิจิตอล กล้องถ่ายวิดีโอดิจิตอล ที่ต่อกับอุปกรณ์คอมพิวเตอร์ผ่านทาง USB Port หรือผ่านทาง Port IEEE 1394 (Fire Wire) รวมทั้งกล้องวิดีโออนาล็อกที่ต่อผ่านทาง Capture card ที่ต่อกับคอมพิวเตอร์ผ่านทาง PCI bus เป็นต้น

สรุปข้อดีและข้อเสียของ MS Vision SDK เมื่อเปรียบเทียบกับ ชุดพัฒนา Application Software ทางด้าน Image Processing อื่นๆ

ข้อดี

- สามารถประมวลผลได้รวดเร็วเหมาะสมกับงานแบบ real-time Image Processing
- การติดต่อระหว่าง Application กับระบบปฏิบัติการ MS Windows เป็นไปได้อย่างมีประสิทธิภาพ ในเรื่องของการใช้ Image memory ร่วมกันระหว่าง processes ต่างๆ ภายใน Application รวมทั้งสามารถใช้งาน Graphic Device Interface
- ผู้ใช้งานสามารถสร้าง pixel data type ได้เอง (User-definable pixel) ซึ่งเหมาะสมกับงานวิจัยทางด้าน Image Processing
- ผู้เขียนโปรแกรมสามารถสร้าง Application โดยไม่ต้องคำนึงถึงอุปกรณ์ต่อเชื่อมที่ใช้ในการนำสัญญาณภาพเข้ามา ซึ่งรวมถึงสามารถต่อเชื่อมอุปกรณ์ใหม่ๆ ได้โดยไม่ต้องแก้ไขตัวโปรแกรม

ข้อเสีย

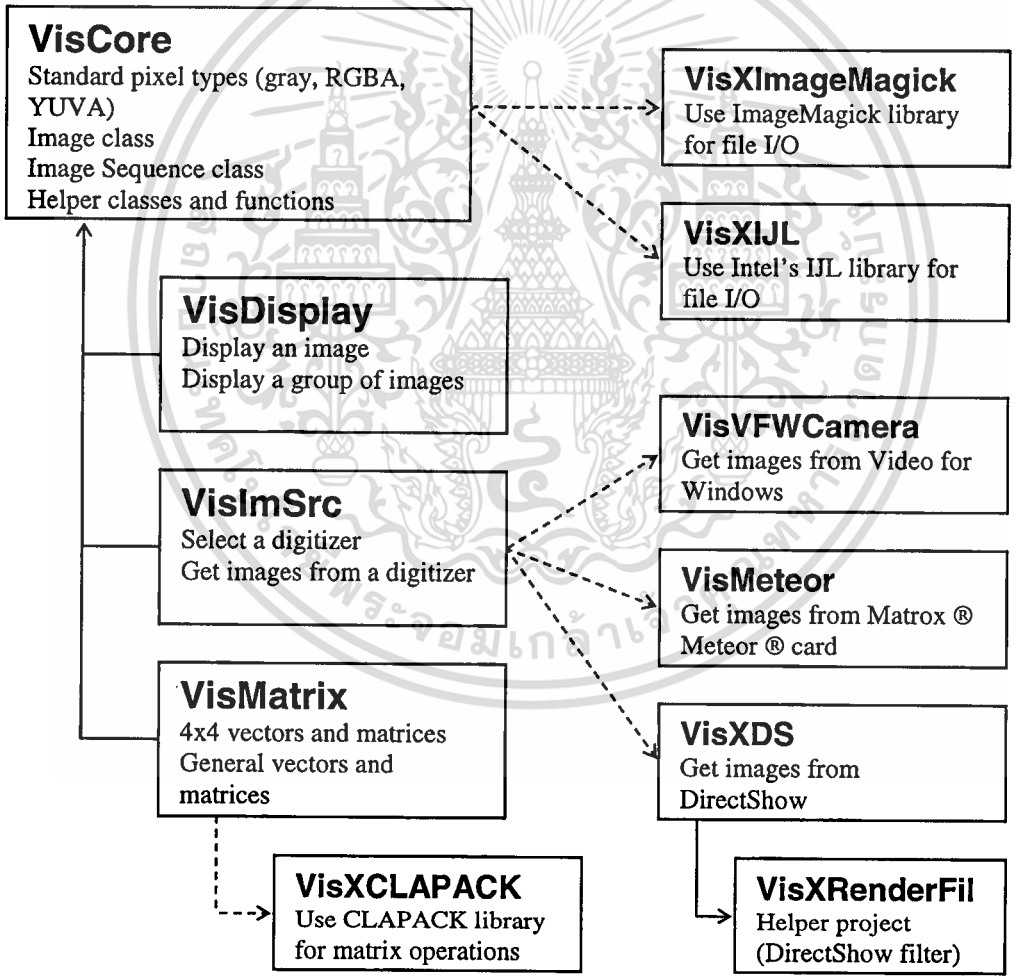
- ข้อมูลภาพทั้งหมดจะอยู่ในหน่วยความจำ (RAM) ทำให้ไม่สามารถรองรับแฟ้มข้อมูลภาพ (Image files) ชนิดที่มีขนาดใหญ่เกินกว่าที่จะอยู่ในหน่วยความจำได้หมด MS Vision SDK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถรองรับขนาดข้อมูลภาพได้ประมาณ 2 GB (แต่ถ้าเป็น MS Windows 9x จะสามารถรับได้ประมาณ 1 GB)

- MS Vision SDK ไม่ได้เตรียมฟังก์ชั่นสำหรับการทำ Image Processing ในระดับ high-level ไว้ให้ เช่น การทำงาน threshold, erosion, dilation เป็นต้น ผู้สร้างจะต้องเขียนฟังก์ชั่นเหล่านี้ไว้ใช้งานเอง หรืออาจใช้ library ของ Intel's Image Processing Library แต่ได้เตรียมฟังก์ชั่นพื้นฐานในระดับ low-level สำหรับการใช้งานด้าน Image Processing ไว้แล้ว

2.5.1. โครงสร้างหลักของ MS Vision SDK



ภาพที่ 2.11 แสดงโครงสร้างหลักของ MS Vision SDK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 2.11 แสดงโครงสร้างหลักของ MS Vision SDK ซึ่งประกอบไปด้วย header files ที่ใช้สำหรับการ include เพื่อให้ precomplier ทราบว่าจะมีการเรียกใช้ components หรือคลาสที่ได้ define ไว้มาใช้ในโปรแกรม โดยในที่นี้จะอธิบายในส่วนของ header file ที่สำคัญๆ ดังนี้คือ

- VisWin.h ซึ่งไม่ได้อยู่รวมกับ MS Vision SDK แต่จะอยู่ใน Microsoft Foundation Class (MFC) ซึ่งเป็น header file ที่ define macros พื้นฐาน และคลาสที่ต้องการใช้ใน MS Vision SDK

- VisCore.h ประกอบไปด้วยคลาสหลักๆ หลายคลาส เช่น คลาส CVisImage เป็นตัวที่เก็บข้อมูลภาพซึ่งใช้สำหรับการจัดเก็บ การนำเอาข้อมูลไปใช้ในการประมวลผล รวมทั้งการนำเสนอข้อมูลที่ผ่านการประมวลผลแล้ว คลาส CVisSequence ใช้สำหรับการจัดเก็บข้อมูลภาพในแบบต่อเนื่อง และการบันทึกภาพต่อเนื่องที่ผ่านการ Process แล้วในรูปแบบ AVI file และสุดท้ายฟังก์ชั่น และฟังก์ชั่น DisplayInHdc เป็นการนำเสนอข้อมูลภาพที่ผ่านการ process แล้วด้วย Windows GDI

- VisImSrc.h สนับสนุนการ capture ภาพจากอุปกรณ์เชื่อมต่อ โดยมี Class CVisImageSource และฟังก์ชั่น VisFIndImageSource ในการค้นหาแหล่งที่มาของสัญญาณภาพ

- VisDisplay.h ใช้ในการนำเสนอภาพที่ผ่านการ process แล้วเพื่อนำเสนอผ่านทางหน้าจอมอนิเตอร์ซึ่งมี Class CVisPane, CVisPaneArray รวมถึงฟังก์ชั่น VisDisplayImage

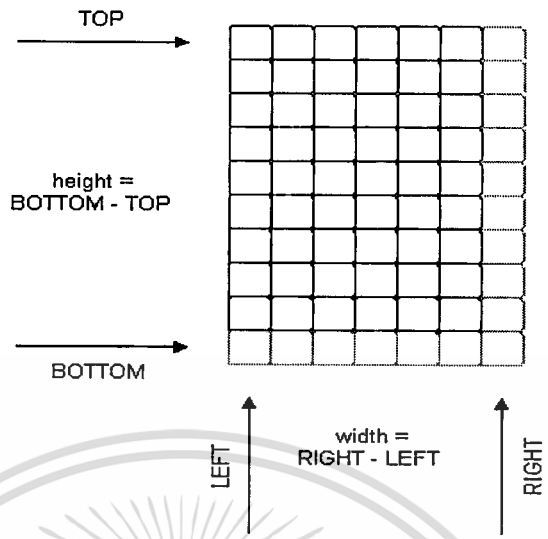
2.5.2. คลาส CVisImage

คลาส CVisImage เป็น คลาสหลักที่ใช้ในการทำงานของ MS Vision SDK คลาสนี้ถูกกำหนดไว้ใน Project "VisCore.h" คลาส CVisImage คล้ายคลึงกับ Windows bitmap header กล่าวคือ คลาส CVisImage หน้าที่เก็บคุณสมบัติหลายอย่างเกี่ยวกับภาพ อีกทั้งยังเก็บ pointer ที่ชี้ยังไปหน่วยความจำที่เก็บข้อมูลภาพจริง ข้อแตกต่างจาก bitmap คือ คลาสนี้ถูกเขียนโดยใช้ภาษา C++ และเขียนเป็นแบบ templates ที่สามารถเก็บและสามารถ process pixel format ชนิดต่างๆ กันได้

ข้อมูลภาพที่จัดเก็บใน Vision SDK จัดเก็บในรูป array ของ pixels ซึ่ง pixels ทั้งหมดจะมี data type ชนิดเดียวกันขึ้นอยู่กับผู้เขียนโปรแกรมว่าต้องการใช้ data type แบบใด pixel type (data type ของ pixel นั้นๆ) มีหลายแบบและมีทั้งแบบที่ MS Vision SDK กำหนดไว้แล้ว (Predefined Pixel and Image Types) และผู้เขียนโปรแกรมกำหนดเองภายหลัง (Adding a New Pixel Type)

MS Vision SDK ใช้โครงสร้างแบบ standard Windows RECT structure สำหรับการระบุข้อมูลเกี่ยวกับพื้นที่ของข้อมูลภาพ ดังแสดงในภาพที่ 2.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.12 โครงสร้างของการเก็บภาพใน Vision SDK

จากภาพตำแหน่ง (0, 0) คือตำแหน่ง TOP, LEFT ในขณะที่ pixel สุดท้ายของภาพคือตำแหน่ง BOTTOM, RIGHT โดยค่า height สามารถหาได้จากการนำค่า BOTTOM ลบด้วยค่า TOP และค่า width หาได้จากการนำค่า RIGHT ลบด้วยค่า LEFT

ผู้เขียนโปรแกรมสามารถอ้างถึงตำแหน่งของแต่ละ pixel ได้โดยตรง โดย MS Vision SDK จะกำหนดค่า x เป็นคอลัมน์ของภาพและค่า y เป็นแถวของภาพ โดยค่า x จะเริ่มต้นที่ 0 และเพิ่มค่าขึ้นทีละ 1 จากด้านซ้ายไปขวา ส่วนค่า y ก็จะเพิ่มค่าขึ้นทีละ 1 เหมือนค่า x ต่างกันตรงที่นับจากบนลงล่าง แต่จุด origin หรือจุด (0, 0) เป็นเพียงจุดสมมุติผู้เขียนสามารถกำหนดจุด origin ได้เองที่ตำแหน่งใดๆ ก็ได้ในภาพ ซึ่งทำให้การทำงานมีความยืดหยุ่นมากขึ้น

Object หรือ Instance ที่สร้างจากคลาส CVisImage นั้น ที่จริงแล้วไม่ได้เก็บข้อมูลของแต่ละ pixel จริงๆ แต่จะเก็บ pointer ที่ชี้ไปยัง block ข้อมูลจริงอีกทีหนึ่ง เพราะฉะนั้นผู้เขียนสามารถสร้าง objects หลายๆ objects โดยที่แต่ละ objects สามารถอ้างอิงไปที่ข้อมูลจริง โดยไม่คัดลอกข้อมูลไปยังที่ใหม่ โดยเมื่อไม่มีการใช้ object นั้นๆ แล้ว MS Vision SDK จะทำการตรวจสอบว่าถ้าไม่มีการอ้างอิงถึงข้อมูลจริงๆ อีกต่อไปแล้ว ก็จะทำการคืนหน่วยความจำส่วนนั้นไป

นอกจากนี้ในคลาส CVisImage ยังสามารถเลือกที่จะ Process เพียงแค่บางพื้นที่ของภาพเท่านั้น โดยใช้ method SubImage() ซึ่งสามารถกำหนดให้มีการ process เฉพาะตำแหน่งที่กำหนด

2.5.2.1 Pixel และ Image Type

CVisImage ถูกสร้างในแบบ template ในโครงสร้างของภาษา C++ ซึ่งเราสามารถกำหนดชนิดของ object ที่สร้างว่าจะให้เป็นแบบชนิดใดแล้วแต่ความต้องการโดยใช้คำสั่ง CVisImage <pixeltype>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดของพิกเซล (Pixel Types)

- **Gray Scale Types** แต่ละ pixel ประกอบไปด้วยค่าความสว่างของแต่ละ pixel ซึ่งจะมีระดับความสว่าง 256 ระดับ (0-255) ซึ่งใช้หน่วยความจำขนาด 8 บิต ตารางที่ 2.1 แสดงรายละเอียดของ pixel แต่ละชนิดในกลุ่มนี้

ตารางที่ 2.1 แสดงรายละเอียดของ pixel แต่ละชนิดในกลุ่ม Gray Scale Pixel Type

Name	Component Size	Data type	Pixel Size	Display	ImageSource
CVis(Gray)BytePixel	8 bits	Unsigned char	8 bits	Yes	Yes
CVis(Gray)CharPixel	8 bits	Signed char	8 bits	No	No
CVis(Gray)ShortPixel	16 bits	Signed short	16 bits	No	No
CVis(Gray)UShortPixel	16 bits	Unsigned short	16 bits	Yes	No
CVis(Gray)IntPixel	32 bits	Signed int	32 bits	No	No
CVis(Gray)LongPixel	32 bits	Signed long	32 bits	No	No
CVis(Gray)UIntPixel	32 bits	Unsigned int	32 bits	Yes	Yes
CVis(Gray)ULongPixel	32 bits	Unsigned long	32 bits	Yes	Yes
CVis(Gray)FloatPixel	32 bits	Float	32 bits	No	No
CVis(Gray)DoublePixel	64 bits	Double	64 bits	No	No

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **RGBA Color Pixel Types** จะประกอบไปด้วย component ย่อยๆ ภายในแต่ละ pixel จำนวน 4 ค่าได้แก่ ค่าความสว่างในย่านสีแดง (method R()) ค่าความสว่างในย่านสีน้ำเงิน (method B()) ค่าความสว่างในย่านสีเขียว (method G()) และค่าอัลฟ่า (method A()) ซึ่งเป็นค่าที่กำหนดระดับการ transparency ตารางที่ 2.2 แสดงรายละเอียดของแต่ละ pixel ภายในกลุ่ม RGBA Color pixel types โดยลำดับในการเก็บคือ B-G-R-A ซึ่งตรงกับค่า RGBQUAD ซึ่งใช้ใน 32-bits Windows Bitmaps

ตารางที่ 2.2 แสดงรายละเอียดของ pixel แต่ละชนิดในกลุ่ม RGBA Color Pixel Types

Name	Component Size	Data type	Pixel Size	Display	ImageSource
CVisRGBABytePixel	8 bits	Unsigned char	32 bits	Yes	Yes
CVisRGBACharPixel	8 bits	Signed char	32 bits	No	No
CVisRGBAShortPixel	16 bits	Signed short	64 bits	No	No
CVisRGBAUShortPixel	16 bits	Unsigned short	64 bits	No	No
CVisRGBAIntPixel	32 bits	Signed int	128 bits	No	No
CVisRGBALongPixel	32 bits	Signed long	128 bits	No	No
CVisRGBAUIntPixel	32 bits	Unsigned int	128 bits	No	No
CVisRGBAULongPixel	32 bits	Unsigned long	128 bits	No	No
CVisRGBAFloatPixel	32 bits	Float	128 bits	No	No
CVisRGBADoublePixel	64 bits	Double	256 bits	No	No

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **YUVA Color Pixel Types** คล้ายกลับกลุ่ม RGBA Color Pixel Types ที่ได้กล่าวมาแล้ว เพียงแต่ค่าความสว่างจัดเก็บในแบบ YUV component แทนที่จะเป็น RGB เท่านั้น ตารางที่ 2.3 แสดงรายละเอียดของ pixel แต่ละชนิดในกลุ่ม YUVA Color Pixel Types

ตารางที่ 2.3 แสดงรายละเอียดของ pixel แต่ละชนิดในกลุ่ม YUVA Color Pixel Types

Name	Component Size	Data type	Pixel Size	Display	ImageSource
CVisYUVABytePixel	8 bits	(Un)signed char	32 bits	Yes	Yes
CVisYUVACharPixel	8 bits	(Un)signed char	32 bits	Yes	Yes
CVisYUVAShortPixel	16 bits	(Un)signed short	64 bits	No	Yes
CVisYUVAIntPixel	32 bits	(Un)signed int	128 bits	No	No
CVisYUVALongPixel	32 bits	(Un)signed long	128 bits	No	No
CVisYUVAULongPixel	32 bits	(Un)signed long	128 bits	No	No
CVisYUVAUIntPixel	32 bits	(Un)signed int	128 bits	No	No
CVisYUVAFloatPixel	32 bits	Float	128 bits	No	No
CVisYUVADoublePixel	64 bits	Double	256 bits	No	No

2.5.2.2 การสร้าง CVisImage Object

การสร้าง CVisImage Object สามารถระบุขนาดของภาพได้โดยใช้ CVisImage constructor ในการกำหนดขนาดคอลัมน์ และแถวของภาพได้ เช่น

```
CVisByteImage Image(100, 50);
```

คำสั่งข้างต้นเป็นการกำหนดขนาดของ Object ให้มีขนาด 100 คอลัมน์ และ 50 แถว หรือ กว้าง 100 pixels และสูง 50 pixels แต่ผู้เขียนโปรแกรมสามารถสร้าง object ก่อนแล้วระบุขนาดในภายหลังโดยใช้ method Allocate เช่น

```
CVisByteImage Image;
Image.Allocate(100, 50);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่ได้กล่าวมาแล้วข้างต้นว่า CVisImage จะทำการเก็บ pointer ที่อ้างอิงถึงตำแหน่งของหน่วยความจำที่เก็บข้อมูลภาพจริง ในบางครั้งผู้เขียนโปรแกรมต้องการเก็บข้อมูลภาพชุดนั้นไปไว้ที่ตำแหน่งแยกกันต่างหาก method Copy() จะเป็นการสร้างหน่วยความจำที่ตำแหน่งใหม่และมีการสร้างชุดข้อมูลใหม่สำหรับเก็บข้อมูลเดิม โดยที่การแก้ไข การ process ที่ข้อมูลชุดนี้จะไม่มีผลกระทบกับข้อมูลต้นฉบับแต่อย่างใด แต่ method Copy() จะทำงานได้เฉพาะกับข้อมูลในชนิดเดียวกัน แต่ถ้าเป็นกรณี เช่น เป็น pixel type ต่างชนิดกันต้องใช้คำสั่งอื่น คือ method CopyPixelTo() ดังตัวอย่างต่อไปนี้

```
// assign ImageFaceOriginal to the subImage in the original data.
CVisByteImage ImageFaceOriginal = ImageOriginal.SubImage(rectFace);
// fill ImageFaceCopy with a new Image data block containing just the subImage
CVisByteImage ImageFaceCopy.Copy(ImageFaceOriginal);
// convert and copy the data into an RGBA version of the face subImage
CVisRGBAByteImage ImageFaceRGBA(rectFace);
ImageFaceOriginal.CopyPixelsTo(ImageFaceRGBA);
```

จากตัวอย่างข้างต้นมีการใช้ method Copy() และ method CopyPixelTo() จะเห็นได้ว่า Copy() สามารถใช้ได้เฉพาะ pixel ในแบบเดียวกัน ส่วน CopyPixelTo() สามารถใช้เปลี่ยนแปลงข้ามกันไปมาได้ซึ่งคุณลักษณะนี้มีประโยชน์มากในการทำขบวนการทาง Image Processing อย่างการทำ binary threshold เป็นต้น

แต่ method CopyPixelTo() ไม่สามารถเปลี่ยน pixel จากกลุ่ม RGBA หรือ YUVA ไปเป็นกลุ่ม grayscale ได้ หรือแม้แต่ RGBA ไปเป็น YUVA ได้ แต่ MS Vision SDK ก็มีฟังก์ชั่นที่ใช้ในการแปลงค่าจาก RGBA ไปเป็น grayscale ได้คือ VisBrightnessFromRGBA ซึ่งมีสูตรในการคำนวณคือ

$$G = 0.299R + 0.587G + 0.114B$$

โดยที่ G คือ ค่าความสว่างในแถบ grayscale

R คือ ค่าความสว่างในย่านสีแดง

G คือ ค่าความสว่างในย่านสีเขียว

B คือ ค่าความสว่างในย่านสีน้ำเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2.3 Operations พื้นฐานของคลาส CVisImage

method Pixel(x, y) เป็น method ในการเข้าถึงค่าความสว่างของแต่ละ pixel ในข้อมูลภาพทุกๆ ชนิด แต่ในบางครั้งการเขียนโปรแกรมต้องมีการเข้าถึงทุกๆ pixel ในภาพ method RowPointer(row) จึงมีความสะดวก และรวดเร็วกว่าในการทำงาน ดังตัวอย่างต่อไปนี้

```
float FindAverageGray(CVisFloatImage Image)
{
        assert(Image.NBands() == 1);
        float fltTotalIntensity = 0;
        for (int y = Image.Top(); y < Image.Bottom();y++)
        {
                float *pflt = Image.RowPointer(y);
                for (int x = Image.Left(); x < Image.Right();x++)
                fltTotalIntensity += pflt[x];
        }
        return (fltTotalIntensity / Image.NPoints());
}
```

จากตัวอย่างข้างต้น pflt เป็น pointer ที่รับค่าตำแหน่งของแถวของข้อมูลภาพที่แต่ละแถว เวลาต้องการเข้าถึงค่าในแต่ละ pixel จริงก็สามารถสั่งให้ pflt เลื่อนไปทีละตำแหน่งจนสุดแนวแกน x ได้

คลาส CVisByteImage และ CVisRGBABtyeImage มีความคล้ายกับ Windows Bitmaps เพราะฉะนั้น คุณสมบัติพื้นฐานบางอย่างจึงสามารถใช้ร่วมกันได้ อย่างเช่น การใช้ Windows GDI functions ในการแก้ไข การเสริมแต่งภาพได้โดยตรง การเรียกใช้ Windows GDI functions นั้นต้องมีการใช้ method Hdc เพื่อให้ได้ object ของคลาส HDC ซึ่งสามารถเรียกใช้ Windows GDI functions ได้ดังตัวอย่างต่อไปนี้

```
CVisRGBAByteImage Image(100, 100);
HDC hdcImage = Image.Hdc();
if (hdcImage != 0)
{
        // Note that we use (0, 0) to refer to the top-left corner of the
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Image when working with Windows GDI functions.
RECT rect;
rect.left = rect.top = 0;
rect.right = Image.Width();
rect.bottom = Image.Height();

HBRUSH hBrush = CreateSolidBrush((COLORREF) 0xff0000);
if (hBrush != 0)
{
        FillRect(hdcImage, &rect, hBrush);
        DeleteObject(hBrush);
}

DrawText(hdcImage, "Hello World!", - 1, &rect,
                DT_CENTER | DT_SINGLELINE | DT_VCENTER);
Image.DestroyHdc();
}
```

จากตัวอย่างข้างต้นเป็นการสร้าง object ของคลาส CVisRGBAByteImage ขนาดความกว้าง 100 คอลัมน์ และสูง 100 แถว (100 x 100 pixels) โดยให้สีพื้นภายในเป็นสีน้ำเงิน และเขียนตัวอักษร "Hello World!" ไว้ตรงกลาง สังเกตว่ามีการใช้ Windows GDI functions เช่น DrawText() โดยต้องมีการสร้าง object ในคลาส HDC โดยรับค่าจาก method Hdc()

2.5.3 การแสดงข้อมูลภาพออกจากจอมอนิเตอร์

เมื่อมีการสร้าง object ในคลาส CVisImage และมีการ process ข้อมูลภายในแล้ว การจะแสดงภาพที่ได้สามารถทำได้ 3 วิธีได้แก่

2.5.3.1 แสดงผลโดยใช้ Windows HDC

ถ้า object ที่สร้างจากคลาส CVisImage มีคุณสมบัติเหมือนใน Windows bitmaps ผู้เขียนโปรแกรมสามารถใช้ method DisplayInHdc() สำหรับการแสดงภาพบน Windows ได้ โดยจะต้องมีการสร้าง object ของคลาส HDC จากนั้นผ่านค่า window's HWND ไปให้กับฟังก์ชั่น GetDC() โดยผู้เขียนโปรแกรมสามารถตรวจสอบผลลัพธ์ที่ได้ หากมีชนิดของข้อมูลภาพแตกต่างออกไป ผู้เขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมสามารถใช้ method CopyPixelTo() สำหรับการเปลี่ยนแปลงชนิดของข้อมูลภาพได้ชนิดของข้อมูลภาพที่ DisplayInHdc สนับสนุน

- CVisRGBAByteImage
- CVisByteImage
- CVisUShortImage (Displayed as RGB555)
- CVisUIntImage
- CVisYUVACharPixel (Displayed as gray scale)

2.5.3.2 แสดงผลโดยใช้คลาส CVisPane และ VisDisplayImage

คลาส CVisPane และ CVisPaneArray กำหนดอยู่ใน VisDisplay.h ผู้เขียนโปรแกรมสามารถ include VisDisplay.h และสามารถสร้าง object จากคลาส CVisPane และ CVisPaneArray โดยสามารถนำ object ที่สร้างจากคลาส CVisImage มาแสดงผลได้

2.5.4 การจัดการติดต่อกับอุปกรณ์ต่อพ่วง

Vision SDK สามารถติดต่อกับอุปกรณ์เชื่อมต่อจากภายนอกได้โดยที่ไม่ขึ้นกับอุปกรณ์ โดย MS Vision SDK ได้เตรียม VisImSrc DLL ไว้ให้ โดยผู้ใช้โปรแกรมสามารถเลือกอุปกรณ์เชื่อมต่อในขณะที่กำลังต่ออยู่กับคอมพิวเตอร์ได้

คลาส CVisImageSource ใน VisImSrc Project โดยสามารถใช้ฟังก์ชั่น VisFindImageSource() ซึ่ง return เป็น instance ของคลาส CVisImageSource ดังตัวอย่าง

CVisImageSource ImSrc = VisFindImageSource("");

จากตัวอย่างข้างต้น ImSrc จะทำการติดต่อกับอุปกรณ์เชื่อมต่อโดยจะเก็บข้อมูลภาพในลักษณะข้อมูลดิบเพื่อรอการ process ต่อไป ในกรณีต้องการติดต่อกับอุปกรณ์ที่เก็บภาพต่อเนื่อง เช่น กล้องถ่ายวีดิโอ สามารถใช้ method SetUseContinuousGrab() ได้

คลาส CVisSequence เป็นคลาสที่มีการใช้มากในการจัดการเกี่ยวกับ process ภาพต่อเนื่อง รวมถึงการอ่าน และการเขียนข้อมูลภาพที่เป็นภาพต่อเนื่อง เช่น AVI file คลาส CVisSequence อยู่ใน VisCore Project เหมือนกับคลาส CVisImage และมีลักษณะเป็น template เช่นเดียวกัน เพราะฉะนั้นจึงสามารถทำงานกับ data type ในชนิดต่างๆ กันได้ การทำงานของคลาส CVisSequence จะทำงานกับข้อมูลภาพที่มีลักษณะเป็นชุดโดยในแต่ละชุดจะต้องเก็บข้อมูลในประเภทเดียวกัน คลาส CVisSequence ใช้ Standard Template library deque class ในการจัดเก็บกลุ่มของข้อมูลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

deque เป็นโครงสร้างข้อมูลที่เก็บ objects ไว้ภายใน และสามารถเข้าถึงข้อมูลโดยการใช้ดัชนี (index) สามารถเพิ่ม และสามารถลบข้อมูลออกจากตำแหน่งหัว หรือท้ายของ deque ได้ คุณสมบัติคล้ายในโครงสร้างข้อมูลแบบคิว

การใช้คลาส CVisSequence สามารถกำหนดขนาดของจำนวน objects ที่จะบรรจุไว้ใน deque ได้ โดยค่า default คือค่า 0 ซึ่งเวลาดึง object ไปใช้จะได้ object เป็นชุดล่าสุดที่เข้ามาใน deque การกำหนดว่าจะใช้ buffer เท่าไหร่ขึ้นกับชนิดของการทำงาน ตัวอย่างเช่น ถ้าต้องการ capture ภาพที่ได้สดๆ มาแสดงบนจอภาพคอมพิวเตอร์ควรตั้งไว้ในขนาดที่ไม่มาก คือ ประมาณ 0 ถึง 5 แต่ถ้าต้องการเก็บภาพจากส่วนอื่นของโปรแกรมไว้ก่อนทำการ process ก็ควรตั้งค่า buffer ขนาดปานกลางคือ ประมาณ 30 ถึง 200 และการตั้งค่า buffer ขนาดสูงคือ ประมาณมากกว่า 200 สำหรับการเก็บภาพแต่ละเฟรมเพื่อนำมาสร้างเป็นไฟล์วีดิโอ สำหรับ method ในการตั้งค่าขนาดของ buffer ของคลาส CVisSequence คือ SetLenghtMax() และยังมี method อื่นๆ เช่น ReadStream(), InsertStream(), AppendStream() และ WriteStream() ในการจัดการเกี่ยวกับไฟล์ AVI ด้วย

การนำเอา object แต่ละ object ที่อยู่ในคลาส CVisSequence มาทำการ process สามารถใช้ method Pop() โดยผู้เขียนสามารถสร้าง object ในคลาส CVisImage มารับไป process ต่ออีกทางหนึ่ง ดังตัวอย่างต่อไปนี้

```
CVisImageSource Imagesource = VisFindImageSource("");
if(Imagesource.IsValid()) {
        CVisSequence<CVisRGBABytePixel> sequence;
        Sequence.ConnectToSource(Imagesource, true, false);
        CVisRGBAByteImage ImageT;
        if(sequence.Pop(Image, 2000) {ImageT.FWriteFile("out.bmp");}
        sequence.DisconnectFromSource();
}
```

จากตัวอย่างโปรแกรมข้างต้นมีการใช้ method ConnectToSource() เพื่อการเชื่อมต่อ object ในคลาส CVisSequence กับ Image Source จากนั้นจึงมีการใช้ method Pop() สำหรับการนำข้อมูลภาพภายใน deque ของ CVisSequence มาเก็บไว้ใน object ของคลาส CVisRGBAByteImage เพื่อทำการ process ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบ การพัฒนาโครงงาน และการทดลอง

ในบทนี้จะเป็นการนำเอาหลักการ และทฤษฎีที่ได้กล่าวเมื่อบทที่ผ่านมานำมาทำการสร้างโปรแกรมประยุกต์โดยมีขั้นตอนดังนี้

3.1. การออกแบบ

โครงสร้างของโครงงาน จะประกอบด้วย กล้องวีดิทัศน์ซึ่งใช้จับภาพการเคลื่อนไหวของผู้ใช้ โดยผู้ใช้จะทำท่าทางตามการ์ตูน ซึ่งตัวการ์ตูนจะเป็นผู้ฝึกสอนการเต้น และมีหน้าที่โต้ตอบกับผู้ใช้โดยใช้คำพูด เช่นการกล่าวชมเมื่อผู้ใช้ทำท่าทางถูกต้อง และเปลี่ยนท่าเมื่อผู้ใช้ทำท่าทางถูกต้องแล้วดังภาพที่ 3.1

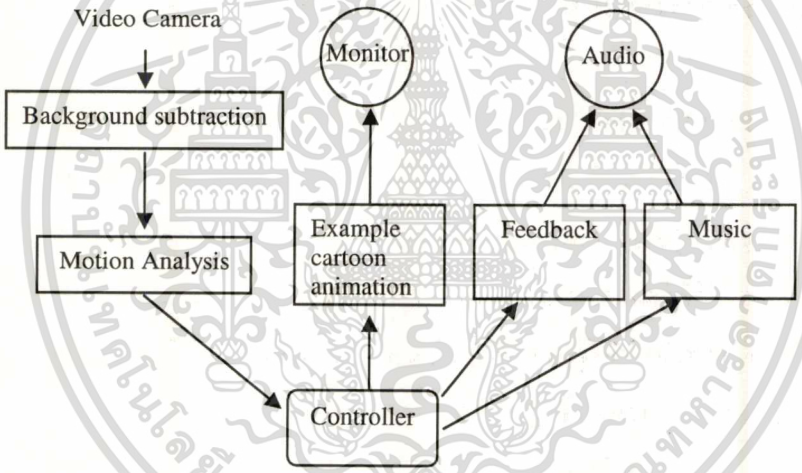
ภาพที่ 3.1 แสดงโครงสร้างของโครงงาน

ขั้นตอนการทำงาน เมื่อเปิดโปรแกรมระบบก็จะเริ่มจากการกล่าวต้อนรับผู้ใช้ด้วยตัวการ์ตูนโดยแนะนำให้ผู้ใช้สร้าง Background Model โดยให้ผู้ใช้กดปุ่ม Build Background Model หลังจากนั้นให้กดปุ่ม Start เพื่อเริ่มตรวจจับการเคลื่อนไหว เมื่อผู้ใช้เข้าในส่วนที่สามารถตรวจจับที่ทางได้ ระบบก็จะเริ่มวิเคราะห์ท่าทางการเคลื่อนไหวของผู้ใช้ หากการเคลื่อนไหวถูกต้องระบบจะกล่าวชมและเปลี่ยนท่าการเต้น แต่หากผู้ใช้ทำท่าทางไม่ตรงกับการ์ตูนที่นำเต้นระบบจะแนะนำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำใหม่จนกว่าจะทำท่าทางได้ถูกต้อง เมื่อทำครบทุกท่าทางที่มีอยู่ระบบจะสิ้นสุดการนำเต้นและกล่าวขอบคุณ

Component diagram แสดงการสื่อสารของ สื่อ(มอนิเตอร์,เสียง) ส่วนควบคุม การรับภาพวีดิทัศน์ การลบฉากหลังและการวิเคราะห์การเคลื่อนไหว ดังภาพที่ 3.2 ระบบจะรับข้อมูลการเคลื่อนไหวของผู้ใช้จากวีดิทัศน์และนำไปประมวลผลโดยการตัดผู้ใช้ออกจากฉากหลังด้วยวิธี Background Subtraction หลังจากนั้นก็จะนำไปวิเคราะห์ท่าทางโดยใช้วิธี Cardboard Model ซึ่งผลที่ได้จากการวิเคราะห์จะใช้ในการทำ Feedback กับผู้ใช้ หากผู้ใช้ทำท่าทางที่ถูกต้อง ระบบจะสุ่มคำกล่าวชม และ เปลี่ยนท่าทางการเต้นในท่าต่อไป แต่หากผู้ใช้ทำท่าทางผิด ระบบจะทำท่าทางนั้นซ้ำอีกจนกว่าผู้ใช้จะทำถูก หรือใช้เวลาเกินเวลาที่กำหนดในแต่ละท่าทาง



ภาพที่ 3.2 Component diagram

3.2. การพัฒนาโครงงาน

ประกอบด้วยส่วนต่างๆมีรายละเอียดดังนี้

- การรับข้อมูลวีดิทัศน์ โดยอาศัยหลัก Video for Windows ทำหน้าที่เป็นตัวกลางในการติดต่อระหว่าง Application program กับ อุปกรณ์ต่อเชื่อม

- การตัดผู้ใช้ออกจากฉากหลังใช้หลักการ Background Subtraction และทำการลบข้อมูลที่ไม่ใช่ส่วนที่ต้องการโดยใช้หลักการทาง Image Processing โดยการทำ Close Morphology

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Operation เพื่อลบส่วนที่เป็นสัญญาณรบกวน (Noise) และ การหา Connected Component เพื่อเลือกส่วนที่ใหญ่ที่ติดกันมาประมวลผล

- การวิเคราะห์ท่าทาง ใช้หลักการ Cardboard Model ในการหาตำแหน่งของส่วนของร่างกายแต่ละส่วน และนำตำแหน่งต่างๆในแต่ละเฟรมมาวิเคราะห์การเปลี่ยนแปลงเทียบกับท่าทางของตัวการ์ตูน

- การแสดงภาพการ์ตูนเคลื่อนไหว และส่วนการโต้ตอบผู้ใช้ ใช้ ActiveX Control Shockwave Flash Object ในการแสดงผล File Flash

3.3. การทดลอง

ในการทดลองระบบช่วยฝึกการออกกำลังกายแบบแอโรบิคโดยการประยุกต์ใช้เทคนิคการตรวจจับการเคลื่อนไหวของมนุษย์จากวีดิทัศน์นั้น ไม่สามารถใช้ตัวเลขหรือการคำนวณทางคณิตศาสตร์ในหารวัดค่าความถูกต้องในการวิเคราะห์ท่าทางว่าถูกต้องมากเพียงใดได้ในการทดลองจึงใช้การทดลองว่าระบบสามารถโต้ตอบได้ตามความเป็นจริงหรือไม่โดยมีรายละเอียดดังนี้

3.3.1 อุปกรณ์ที่ใช้ในการทดลองระบบ

- กล้อง Web Cam ผ่าน USB Port ใช้ในการส่งภาพวีดิโอที่จับภาพได้เข้าสู่เครื่องคอมพิวเตอร์ที่ใช้ในการประมวลผลระบบ

- เครื่องคอมพิวเตอร์ Notebook CPU Pentium 4-M 1.8 GHz หน่วยความจำหลัก 512 MB

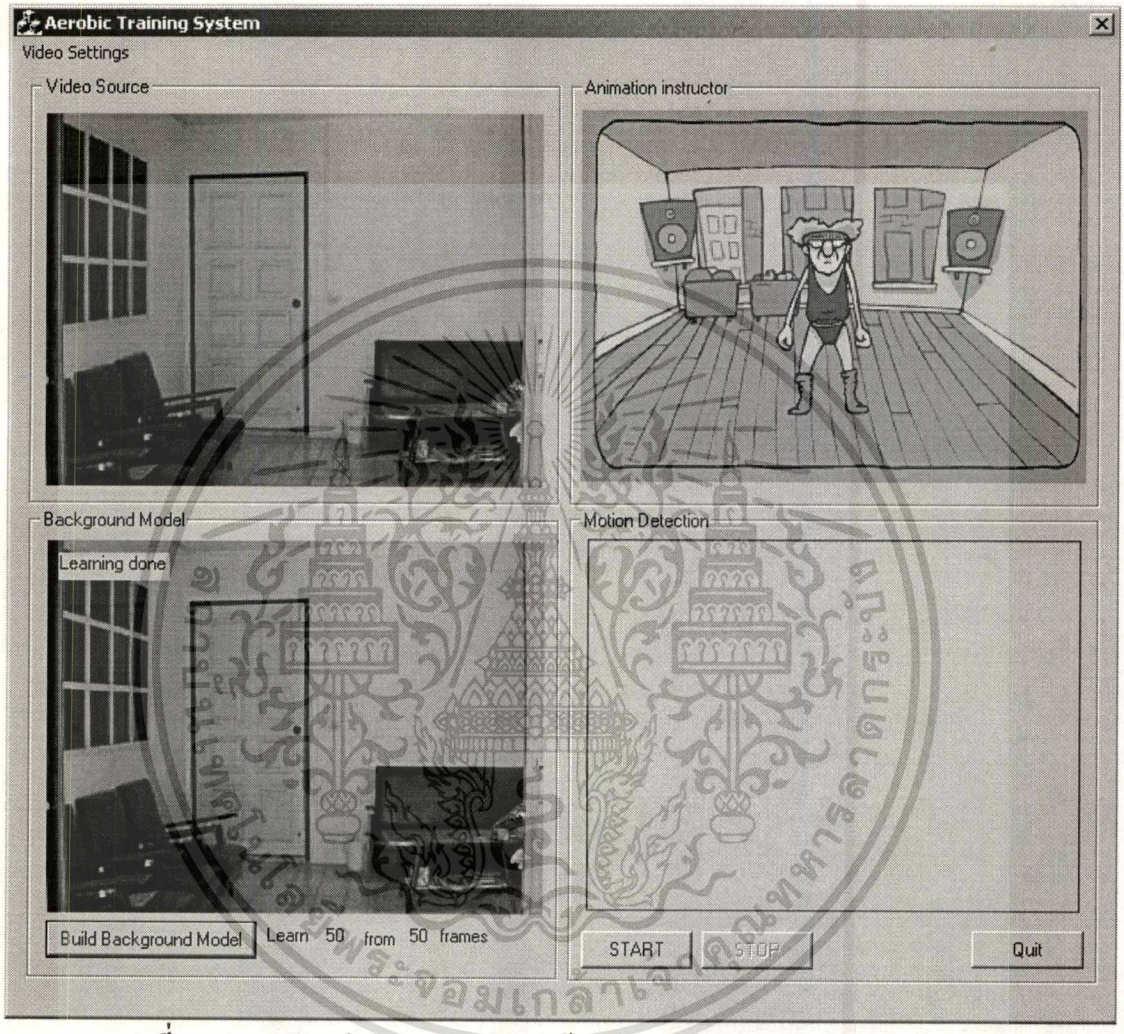
3.3.2 ขั้นตอนการทดลอง

3.3.2.1 การทดลองการสร้าง Background Model

ในการทดลองเริ่มด้วยการเปิดโปรแกรมแล้วสร้าง Background Model ดังภาพที่ 3.3 เป็นภาพหน้าจอของโปแกรมในภาพประกอบด้วย 4 ส่วนหลักคือ มุมขวาบนจะแสดงภาพการ์ตูนกล่าวคำต้อนรับพร้อมการแนะนำผู้ใช้ในให้สร้าง Background Model มุมซ้ายบนเป็นภาพที่ได้จากวีดีทัศน์ หรือเป็นข้อมูลเข้า เมื่อกดปุ่ม Build Background Model ระบบก็จะเริ่มสร้าง Background Model โดยแสดงจำนวนเฟรมที่ทำการ Learn และแสดงข้อความ “Learning…” เมื่อสร้างเสร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะแสดงผล ภาพ Background Model ภาพซ้ายล่าง ที่ถูกสร้าง ณ เวลาที่ไม่มีผู้ใช้อยู่ในจอภาพ และแสดงผล Learn done



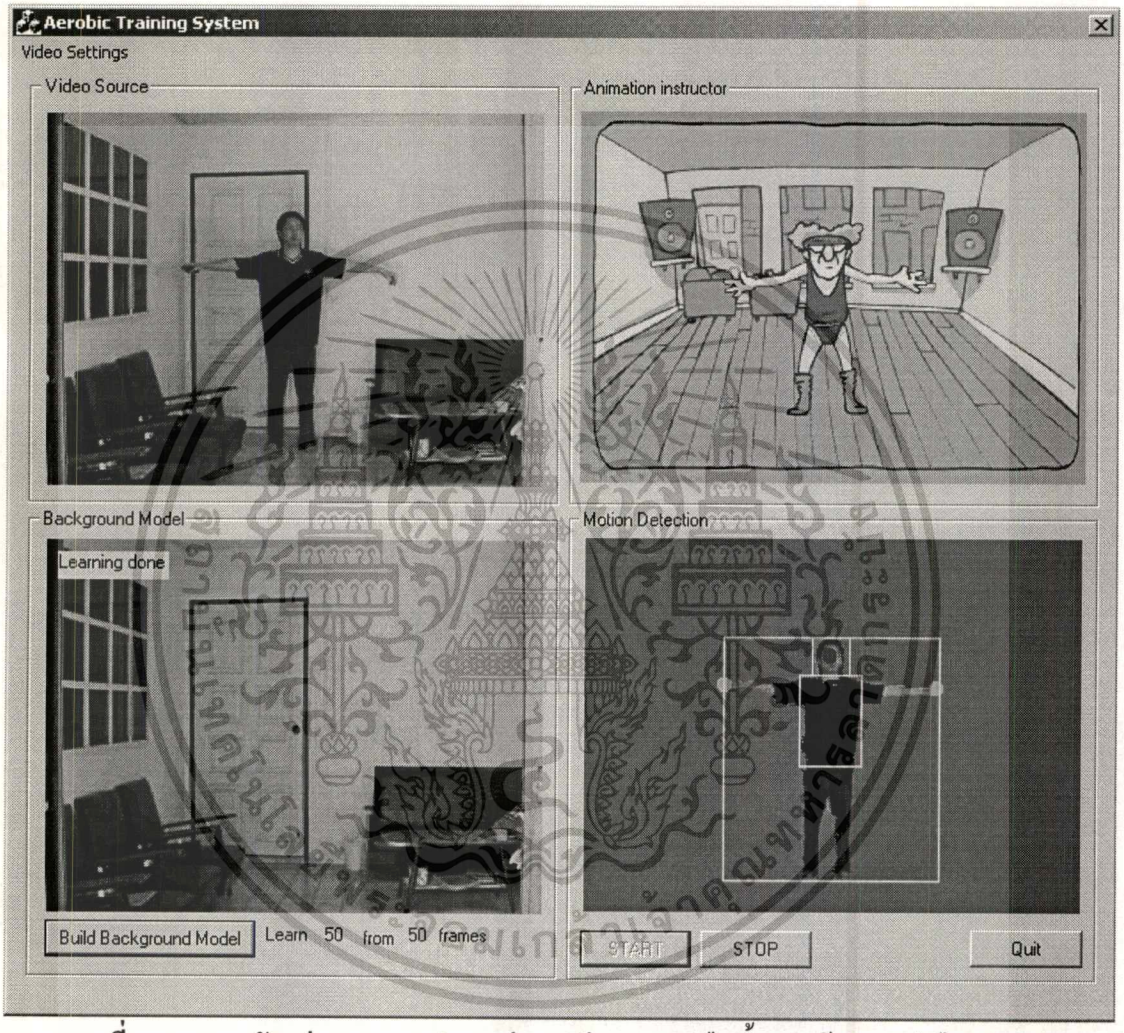
ภาพที่ 3.3 แสดงตัวอย่างผลการทดลองสร้าง Background Model

3.3.2.1 การทดลองการโต้ตอบกับผู้ใช้

เริ่มด้วยการกดปุ่ม START เพื่อเริ่มการตรวจจับและวิเคราะห์การเคลื่อนไหวเมื่อมีผู้ใช้เข้าสู่ระบบ ระบบจะเริ่มวิเคราะห์ท่าทางโดยการหาตำแหน่งและแสดงจุดที่หาได้โดยจุด ณ ส่วนที่ตรวจจับการเคลื่อนไหวนั่นคือศีรษะ และมือทั้งสองข้าง ดังภาพที่ 3.4 แสดงตัวอย่างการระบุตำแหน่งของศีรษะ และมือทั้งสองข้างจากท่ายืนกางแขน และ ภาพที่ 3.5 แสดงตัวอย่างการระบุตำแหน่งของศีรษะและมือทั้งสองข้างจากท่าย่อขากางแขน ซึ่งจะสังเกตว่าระบบสามารถหาตำแหน่ง

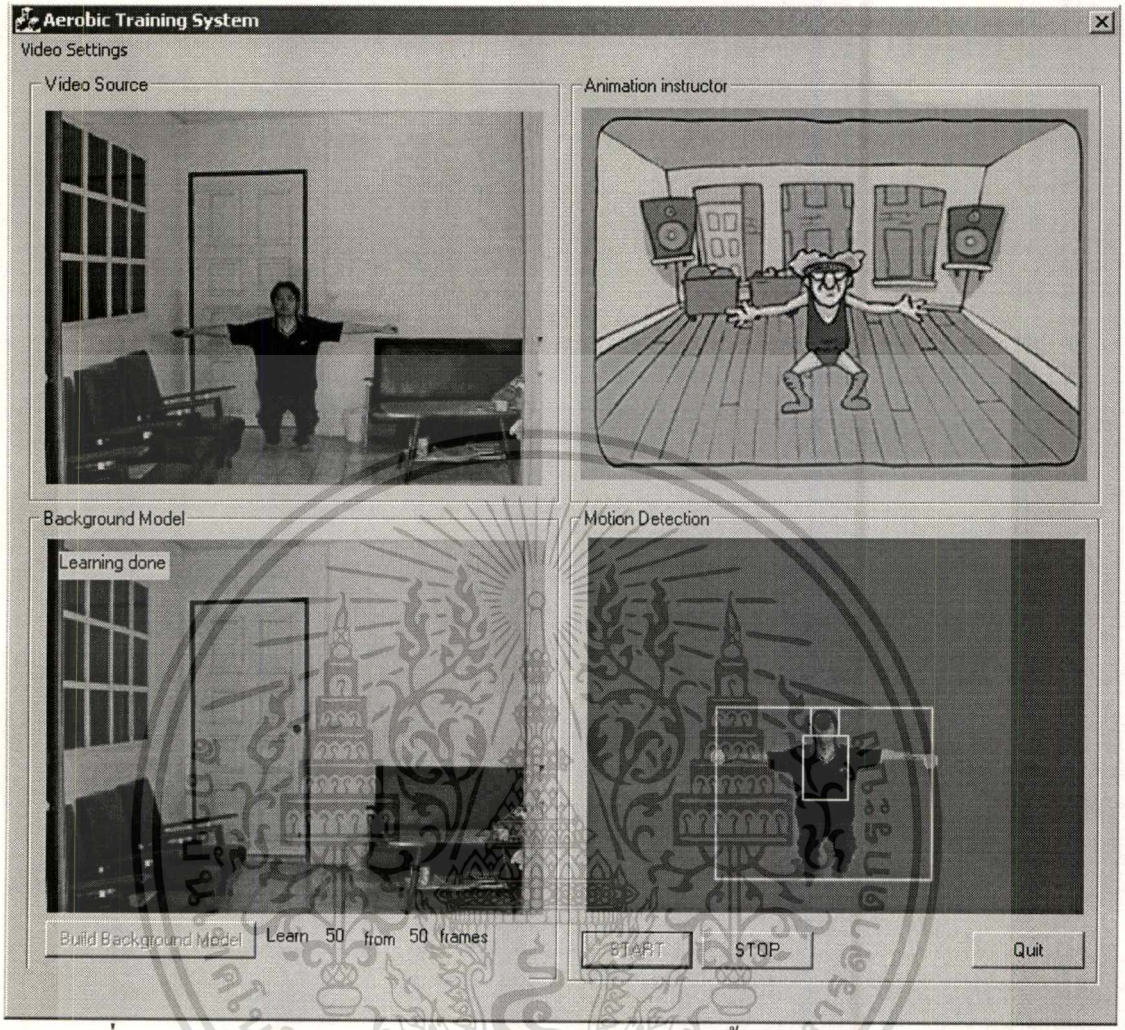
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของส่วนทั้งสองได้หลังจากนั้น ระบบก็จะเริ่มทำการวิเคราะห์ท่าทางว่าตรงกับท่าทางของการ์ตูนหรือไม่และทำการโต้ตอบกับผู้ใช้โดยใช้เสียงในการบอกว่าทำท่าทางผิดหรือไม่ตรงกันที่การ์ตูนทำเป็นต้น



ภาพที่ 3.4 แสดงตัวอย่างการระบุตำแหน่งของศีรษะ และมือทั้งสองข้างจากท่ายืนกางแขน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.5 แสดงตัวอย่างการระบุตำแหน่งของศีรษะ และมือทั้งสองข้างจากท่าย่อขากางแขน

3.4. สรุปผลการทดลอง

ระบบสามารถทำงานได้อย่างถูกต้องเมื่อฉากหลังมีการเปลี่ยนแปลงของแสงและเงาน้อย ในการโต้ตอบกับผู้ใช้ระบบสามารถโต้ตอบได้ โดยจะขึ้นอยู่กับท่าทางของผู้ใช้และคุณภาพที่ได้จากการลบฉากหลังออก โดยหากในบางท่าทางผู้ใช้เคลื่อนไหวน้อยไป เช่น ย่อเข่าน้อยไปกว่าที่ระบบกำหนดเอาได้ ระบบจะวิเคราะห์ว่าทำท่าทางไม่ถูกเป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
บทสรุปและข้อเสนอแนะ

4.1 บทสรุปของโครงการพัฒนาระบบงาน

โครงการพัฒนาระบบช่วยฝึกการออกกำลังกายแบบแอโรบิคโดยการประยุกต์ใช้เทคนิคการตรวจจับการเคลื่อนไหวของมนุษย์จากวีดิทัศน์เป็นการศึกษาทฤษฎีที่เกี่ยวข้องกับการพัฒนาระบบในส่วนต่างๆ และเป็นการสร้างระบบให้สามารถทำงานได้ตามที่กำหนดไว้ในขอบเขตของการพัฒนาระบบ โดยระบบประกอบไปด้วยส่วนหลักๆ คือ การนำข้อมูลจากวีดิทัศน์เข้าสู่ระบบ การตัดส่วนที่สนใจออกจากฉากหลัง การตรวจจับการเคลื่อนไหวของมนุษย์ และการโต้ตอบกับผู้ใช้ในการพัฒนาระบบยังพบปัญหาและข้อจำกัดในการพัฒนาระบบหลายอย่าง ดังจะได้กล่าวในหัวข้อถัดไป

4.2 ปัญหาและข้อจำกัดในการทำงานของระบบ

ในการพัฒนาระบบระบบช่วยฝึกการออกกำลังกายแบบแอโรบิคโดยการประยุกต์ใช้เทคนิคการตรวจจับการเคลื่อนไหวของมนุษย์จากวีดิทัศน์ มีปัญหาและข้อจำกัดในการพัฒนาหลายอย่าง เนื่องจากระบบมีปัจจัยและข้อจำกัดที่ส่งผลต่อการทำงานของระบบ ดังต่อไปนี้

4.2.1 ในส่วนของการนำเข้าข้อมูลจากวีดีทัศน์ยังพบปัญหาว่าไม่สามารถนำข้อมูลจากวีดิทัศน์เข้าในระบบได้ทุกยี่ห้อทุกรุ่น จากการสังเกตพบว่าระบบต้องการวีดิทัศน์ที่มี Diver Support การแสดงผลภาพแบบ RGB

4.2.2 ในส่วนของการตัดส่วนที่สนใจออกจากฉากหลังทำงานได้ดีเมื่อแสงและเงามีการเปลี่ยนแปลงน้อย ไม่เหมาะกับพื้นผิวที่มีการสะท้อนแสงเช่นกระจก หรือวัสดุที่มีความมันวาว และในส่วนของการประมวลผลโดยการหา Connected Component ทำให้ระบบต้องประมวลผลมากทำให้ไม่สามารถประมวลผลได้ทุกๆเฟรมที่เข้ามาในระบบได้ซึ่งมีผลกระทบในข้อต่อไป

4.2.3 การตรวจจับการเคลื่อนไหวของมนุษย์ที่ต้องนำข้อมูลจากการตัดส่วนที่สนใจออกจากฉากหลังที่มีการประมวลหากผู้ใช้เคลื่อนที่ด้วยความเร็วระบบจะไม่สามารถทาบการเคลื่อนไหวได้เนื่องจากบางเฟรมหายไปซึ่งมีสาเหตุมาจาก 2 ส่วนคือ วีดิทัศน์ไม่สามารถจับภาพได้ และการประมวลผลการลบฉากหลังทำได้ไม่ทุกเฟรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 การตรวจจับการเคลื่อนไหวของมนุษย์ในโครงงานนี้จะตรวจจับได้เฉพาะการเคลื่อนไหวที่ไม่ซับซ้อนมากโดยจะไม่สามารถจรวจจับท่าทางที่ใช้มือชูเหนือระดับศีรษะ เพราะระบบใช้การคาดเดาว่าส่วนที่อยู่สูงที่สุดคือส่วนของศีรษะและใช้การคาดเดาตำแหล่งจากสัดส่วนที่ควรจะเป็นในส่วนอื่นๆต่อไป

4.2.5 การตรวจจับการเคลื่อนไหวของมนุษย์ไม่สามารถตรวจจับความเคลื่อนไหวของมนุษย์ได้มากกว่า 1 คน

4.2.6 ระบบต้องการทำงานแบบตอบสนองทันทีทันใด(Real-time) ทำให้มีการประมวลสูง ใช้ทรัพยากรต่างๆ เช่น CPU, RAM และ HARDDISK สูง จำเป็นต้องใช้เครื่องที่มีประสิทธิภาพและมีหน่วยความจำสูงในการประมวลผลระบบ

4.3 ข้อเสนอแนะในการพัฒนาระบบต่อไป

4.3.1 พัฒนา Algorithm ในการประมวลผลให้สามารถตอบสนองได้เร็วหรือสามารถวิเคราะห์ข้อมูลที่ได้ในทุกเฟรมเพื่อความถูกต้อง

4.3.2 เพิ่มเติมจำนวนท่าทางในระบบโดยอาจจะต้องใช้วิธีในการวิเคราะห์การเคลื่อนไหวที่สามารถวิเคราะห์ท่าทางที่ซับซ้อนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Haritaoglu, I. Harwood, D. and Davis, L.S. 1998 "Ghost: A Human Body Part Labeling System Using Silhouettes" **14th International Conference on Patten Recognition.**

Haritaoglu, I. Harwood, D. and Davis, L.S. 1998 "W4: Who? When? Where? What? A Real time System for Detection and tracking People" **International Conference on Face and Gesture Recognition**.

Horprasert, T. Harwood, D. and Davis, L.S. 1999 "A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection", **Proc. IEEE ICCV'99 FRAME-RATE Workshop, Kerkyra, Greece.**

Kruglinski, D.J. 1997. **Inside Visual C++.** 4th edition. Redmond: Microsoft Press.

Vision Technology Group, Microsoft Research, 2000. **The Microsoft Vision SDK Version 1.2, May 2000.** [Online]. Available:
http://research.microsoft.com/projects/VisSDK/VisSDK.doc

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ นามสกุล	นายอลงกต น้อยใจบุญ
เกิดวันที่	9 พฤศจิกายน 2520
ที่อยู่	22/4 หมู่ 8 ต.บางพระ อ.เมือง จ.ฉะเชิงเทรา 24000
สถานที่เกิด	จังหวัด กรุงเทพฯ
การศึกษา	การจัดการสารสนเทศ มหาวิทยาลัย รังสิต
การทำงาน	เจ้าหน้าที่พัฒนาระบบข้อมูล มหาวิทยาลัยรังสิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้