การนำเสนอเทคนิคการประมวลภาพเพื่อการจำลองการเคลื่อนไหวตัวละคร 3 มิติ

Proposal of an Image Processing Technique for 3D Character Animation

นายดนัยชาติ แจ่มจิตรตรง

นายต่อศักดิ์ รักอารมณ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2548

การนำเสนอเทคนิกการประมวลผลภาพเพื่อการจำลองการเคลื่อนไหวตัวละคร3มิติ

Proposal of an Image Processing Technique for 3D Character Animation

นายดนัยชาติ แจ่มจิตรตรง
นายต่อศักดิ์ รักอารมณ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2548

การนำเสนอเทคนิคการประมวลผลภาพเพื่อการจำลองการเคลื่อนไหวตัวละคร3มิติ

Proposal of an Image Processing Technique for 3D Character Animation

โดย

นายดนัยชาติ แจ่มจิตรตรง

นายต่อศักดิ์ รักอารมณ์

อาจารย์ที่ปรึกษา

ดร. สมศักดิ์ วลัยรัชต์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2548

ปริญญานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การนำเสนอเทคนิกการประมวลผลภาพเพื่อการจำลองการเคลื่อนไหวตัวละคร 3 มิติ

Proposal of an Image Processing Technique for 3D Character Animation

ผู้จัดทำ

1. นายคนัยชาติ แจ่มจิตรตรง รหัสนักศึกษา 45010262
2. นายต่อศักดิ์ รักอารมณ์ รหัสนักศึกษา 45010279

_____ อาจารย์ที่ปรึกษา

(สมศักดิ์ วลัยรัชต์)

การนำเสนอเทคนิกการประมวลผลภาพเพื่อการจำลองการเคลื่อนไหวตัวละคร 3 มิติ

นายดนัยชาติ แจ่มจิตรตรง 45010262
นายต่อศักดิ์ รักอารมณ์ 45010279
สมศักดิ์ วลัยรัชต์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2548

บทคัดย่อ

ระบบการตรวจจับการเคลื่อนไหวของมนุษย์แบบโมชั่นแคปเจอร์โดยใช้กล้อง (Optical Motion Capture) คือการใช้กล้องวีดีโอจำนวนอย่างน้อย 2 ตัวในการเก็บภาพการเคลื่อนไหวของมนุษย์ และบันทึกไว้เป็นไฟล์วีดีโอ จากนั้นนำไฟล์วีดีโอทั้งหมดมาประมวลผลหาค่าตำแหน่งของข้อต่อเหล่านั้นโดยใช้ทฤษฎีการประมวลผลภาพ จากนั้นนำตำแหน่งบนภาพวิดีโอทั้งหมดมาคำนวณเพื่อหาตำแหน่งจริงในสามมิติ โดยตำแหน่งเหล่านี้จะถูกบันทึกลงไฟล์การเคลื่อนไหวแบบต่างๆ ซึ่งสามารถนำไปสร้างการเคลื่อนไหวให้กับตัวละคร 3 มิติในโปรแกรมสามดีสตูดิโอแม็กซ์ (3ds Studio Max) ต่อไป

การนำเสนอเทคนิกการประมวลผลภาพเพื่อการจำลองการเคลื่อนไหวตัวละคร 3 มิติ เป็นโครงงานการพัฒนาระบบการตรวจจับการเคลื่อนไหวของมนุษย์แบบโมชั่นแคปเจอร์โดยใช้กล้อง (Optical Motion Capture) ซึ่งการพัฒนาระบบจากโครงงานก่อนหน้า โดยจะเป็นการพัฒนา โดยการเพิ่มกล้องที่ใช้ในการตรวจจับการเคลื่อนไหว เป็น 4 กล้อง และ นำเสนอเทคนิคของการทำการประมวลผลภาพ ที่จะถูกนำไปใช้ในการตรวจจับมาร์กเกอร์สีต่างๆ สำหรับการจำลองการทำการเคลื่อนไหวของตัวละคร โดยในการทดลองนี้ระบบจะทำการแสดงการปรับปรุงความสามารถในการตรวจจับมาร์กเกอร์สีต่างๆ เนื่องจากความถูกต้องของการประมวลผลการเคลื่อนไหวนั้น จะส่งผลให้การเคลื่อนไหวของตัวละครนั้นถูกต้องและ เสมือนจริงมากขึ้นด้วย

Proposal of an Image Processing Technique for 3D Character Animation

Danaichart	Jamjittrong	45010262
Torsak	Rak-arom	44010279
Somsak	Walairacht	Advisor

ABSTRACT

The Optical Motion Capture is a system that uses at least 2 cameras to capture a human motion and record it in video files. Image sequences are then brought to calculate for corresponding positions in the 3D world. The motion positions are recorded in a motion file for generating the animation of a 3D character using 3D Studio Max.

This Project, Proposal of an Image Processing Technique for 3D Character Animation, is an alternative development based on the previous Optical Motion Capture System from the previous project. The different is a technique of using 4 cameras. Also, a new technique of image processing is employed to detect color markers used for simulating the animation of the 3D character. In the experiments, the improvement of capable for capturing color markers can be noticed. The results show the correct analysis of motion that resulting in smooth animation of the 3D character.

กิตติกรรมประกาศ

โครงงานและปริญญานิพนธ์นี้ มิอาจเกิดขึ้นได้หากขาดการสนับสนุนจากอาจารย์ที่ปรึกษาโครงงาน อาจารย์ ดร.สมศักดิ์ วลัยรัชต์ รวมถึงอาจารย์ ดร.อรัญญา วลัยรัชต์ ที่ได้ให้คำปรึกษา คำแนะนำ และความช่วยเหลือในเรื่องต่างๆ ในการทำโครงงานนี้ และบุคคลอื่นที่ไม่ได้กล่าวถึง ณ ที่นี้ ที่ได้มีส่วนร่วมในการส่งเสริม และสนับสนุนการทำโครงงานนี้ไม่ว่าในรูปใดๆ ก็ตาม จนทำให้สามารถดำเนินโครงงานจนสำเร็จเสร็จสิ้นไปด้วยดี ดังนั้นข้าพเจ้าจึงขอขอบคุณทุกท่านมาไว้ ณ ที่นี้

นอกเหนือจากบุคคลข้างต้นแล้ว ข้าพเจ้าขอขอบคุณบุพการีที่ทำให้ข้าพเจ้าได้มาถึง ณ จุดนี้ ซึ่งท่านมีส่วนสำคัญต่อข้าพเจ้าในการชุบเลี้ยงจนเติบโต และให้ข้าพเจ้าได้รับการศึกษา รวมถึงการสนับสนุนในด้านต่างๆ ดังนั้นข้าพเจ้าจึงขอระลึกถึงพระคุณ และขอกราบขอบพระคุณท่านมาไว้ ณ ที่นี้

ท้ายสุดนี้ข้าพเจ้าขอขอบคุณเพื่อนร่วมงาน ที่ได้ร่วมมือกันทำงาน และแก้ไขปัญหาต่างๆ จนทำให้โครงงานสำเร็จลุล่วงด้วยดี

นายคนัยชาติ แจ่มจิตรตรง
นายต่อศักดิ์ รักอารมณ์

สารบัญ

สารบัญ
(ต่อ)

สารบัญ
(ต่อ)

สารบัญรูปภาพ

สารบัญรูปภาพ
(ต่อ)

สารบัญรูปภาพ
(ต่อ)

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

ปัจจุบัน เทคโนโลยีในการสร้างตัวละคร 3 มิติ เริ่มมีบทบาทสำคัญมากขึ้น โดยเฉพาะการสร้างภาพยนตร์, การสร้างโฆษณา, การพัฒนาเกมส์คอมพิวเตอร์ และอื่นๆ อีกมากมาย

ในประเทศไทยได้มีการเริ่มนำเทคโนโลยีนี้มาใช้แล้ว แต่ยังมีการให้บริการด้านนี้อยู่น้อยมาก เพราะมีปัญหาในการที่ต้องซื้อเทคโนโลยีนี้มาในราคาแพงจากต่างประเทศ เพราะอุปกรณ์ต่างๆนั้นต้องเป็นอุปกรณ์ที่มีประสิทธิภาพ และมีความละเอียดในการทำงานสูง

ดังนั้นการทำโครงงานนี้ จะเน้นทางด้านการพัฒนาระบบและการทำงาน เพื่อเป็นพื้นฐานในการพัฒนาในอนาคต ให้ดียิ่งขึ้นต่อไป

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อเพิ่มประสิทธิภาพในการตรวจจับการเคลื่อนไหวให้ดียิ่งขึ้น
2. เพิ่มประสิทธิภาพในการวิเคราะห์หาการเคลื่อนไหวให้ดียิ่งขึ้น
3. เพื่อให้การทำงานได้ภาพ 3 มิติ ที่เคลื่อนไหวได้สมบูรณ์ยิ่งขึ้น

1.3 ขอบเขตของโครงงาน

เป็นการนำเสนอวิธีการประมวลผลภาพ โดยการพัฒนากระบวนการในการตรวจจับการเคลื่อนไหว เพื่อให้สามารถทำการจำลองการเคลื่อนไหวของตัวละคร 3 มิติ ได้สมบูรณ์ และให้มีความสะดวกในการใช้งานยิ่งขึ้น

โดยในส่วนของการบันทึกภาพการเคลื่อนไหวของนักแสดงของโครงงานนี้จะเป็นการจำลอง (Simulation) โดยใช้โปรแกรม 3ds Studio Max ในการสร้างผู้แสดงสมมติ และกล้องที่ใช้ในการบันทึกภาพเคลื่อนไหว โดยผลของการบันทึกข้อมูลภาพการเคลื่อนไหวจะอยู่ในรูปของไฟล์วีดีโอ จากนั้นจะนำไฟล์วีดีโอข้อมูลการเคลื่อนไหวของนักแสดงที่ได้มาวิเคราะห์หาตำแหน่งของข้อต่อ โดยมีการนำเสนอการออกแบบการทำงานในส่วนของการประมวลผลต่างๆ เพื่อให้ระบบสามารถทำงานได้อย่างต่อเนื่อง และให้ดียิ่งขึ้น แล้วบันทึกข้อมูลในรูปแบบของไฟล์ข้อมูลการเคลื่อนไหวแบบ CSM เพื่อไว้ใช้ในการสร้างการเคลื่อนไหวให้กับตัวละคร 3 มิติต่อไป

1.4 วิธีการดำเนินการ

1. ศึกษาเกี่ยวกับกระบวนการประมวลผลภาพ
2. ศึกษาการสร้างโมเดล 3 มิติ และการสร้างการเคลื่อนให้กับตัวละคร 3 มิติ
3. ศึกษาเกี่ยวกับโครงสร้างของไฟล์ที่ใช้ในการบันทึกข้อมูล
4. ศึกษาการใช้โปรแกรม 3D Studio Max เบื้องต้น
5. ออกแบบวิธีการทำงาน ในส่วนของการวิเคราะห์ตำแหน่งต่างๆ
6. สร้างผู้แสดง และกล้องที่ใช้ในการตรวจจับการเคลื่อนไหวของผู้แสดง ด้วยโปรแกรม 3D Studio Max
7. พัฒนาโปรแกรมที่ใช้ในการวิเคราะห์หาตำแหน่งมาร์กเกอร์ต่างๆ
8. วิเคราะห์ผลของระบบตรวจจับการเคลื่อนไหวที่ได้ทำการพัฒนาขึ้น และแก้ไขส่วนที่ผิดพลาดของระบบ เพื่อให้ระบบสามารถสร้างไฟล์ข้อมูลการเคลื่อนไหวที่สามารถนำไปสร้างการเคลื่อนไหวให้ตัวละครได้สมจริงมากที่สุด

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับความรู้ความเข้าใจเกี่ยวกับกระบวนการประมวลผลภาพ
2. ได้รับความรู้ความเข้าใจเกี่ยวกับการสร้างโมเดล 3 มิติ
3. ได้รับความรู้ความเข้าใจเกี่ยวกับการสร้างการเคลื่อนไหวให้กับตัวละคร 3 มิติ
4. ได้รับความรู้ความเข้าใจเกี่ยวกับการใช้โปรแกรม 3ds Studio Max เบื้องต้น

1.6 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 บทนำ คือบทที่ 1 นี้

บทที่ 2 ทฤษฎีพื้นฐานและการออกแบบ ซึ่งประกอบด้วยไปด้วย 6 บท ได้แก่

2.1 ความรู้เบื้องต้นเกี่ยวกับการตรวจจับการเคลื่อนไหว อธิบายถึงการตรวจจับการเคลื่อนไหวด้วยวิธีโมชั่นแคปเจอร์ด้วยวิธีต่างๆ ตลอดจนการนำไปประยุกต์ใช้งาน

2.2 การประมวลผลภาพ แสดงความรู้เกี่ยวกับการประมวลผลภาพ อัลกอริธึมที่ใช้ในโครงงาน

2.3 Stereopsis เป็นการอธิบายในส่วนทฤษฎีการคำนวณหาตำแหน่งจากภาพ 2 ภาพ โดยมีการคำนวณแบบขาไปคือหาตำแหน่งจริงจากภาพ 2 ภาพและการคำนวณขากลับคือการหาตำแหน่งบนภาพจากตำแหน่งจริงที่รู้

2.4 Character Studio Motion File เป็นการอธิบายถึงไฟล์ข้อมูลการเคลื่อนไหวที่ระบบจะใช้เป็นเอาท์พุท เพื่อนำไปสร้างการเคลื่อนไหวให้กับตัวละครต่อไป

2.5 การออกแบบการทำงาน เป็นการอธิบายถึงการวางตำแหน่งกล้อง การแสดงขั้นตอนและวิธีการในการทำงานในส่วนต่างๆ ของการทำการสร้างข้อมูลการเคลื่อนไหวให้กับตัวละคร

บทที่ 3 ส่วนการทดสอบ เป็นส่วนที่แสดงการทดลองระบบ ตั้งแต่การสร้างตัวละครเคลื่อนไหว จนถึงขั้นตอนในการทำการวิเคราะข้อมูล และสุดท้ายเป็นการนำข้อมูลที่ได้ไปใส่ในตัวละคร 3 มิติ ในการแสดงการเคลื่อนไหวอีกทีหนึ่ง

บทที่ 4 ส่วนผลการทดสอบ เป็นการทดสอบระบบที่ได้สร้างขึ้น เพื่อแสดงถึงผลที่ได้จากโครงงาน รวมถึง ข้อผิดพลาดต่างๆที่เกิดขึ้นในระบบ

บทที่ 5 เป็นบทวิจารณ์และสรุป เป็นการสรุปโครงงานทั้งหมดและแนะนำแนวทางการพัฒนาต่อ

บทที่ 2
ทฤษฎีพื้นฐานและการออกแบบ

ในหัวข้อนี้จะกล่าวถึงรายละเอียดเกี่ยวกับทฤษฎีต่างๆ ที่เกี่ยวข้องกับโครงงาน ซึ่งเนื้อหาในบทนี้จะกล่าวถึงความรู้เบื้องต้นเกี่ยวกับการตรวจจับการเคลื่อนไหว การประมวลผลภาพ Stereopsis Character Studio Motion File การสร้างการเคลื่อนไหวให้กับตัวละคร 3 มิติ และการสร้างกล้อง และควบคุมกล้อง ในโปรแกรม 3D Studio Max ซึ่งเนื้อหาทั้งหมดนี้เป็นการปูพื้นฐานความเข้าใจต่อการทำโครงงาน

2.1. ความรู้เบื้องต้นเกี่ยวกับการตรวจจับการเคลื่อนไหว

2.1.1 การตรวจจับการเคลื่อนไหว (Motion Capture)

การตรวจจับการเคลื่อนไหวเป็นกระบวนการในการบันทึกการเคลื่อนไหวจริง แล้วเปลี่ยนให้อยู่ในรูปทางคณิตศาสตร์ที่สามารถนำไปใช้ได้ โดยการติดตามจุดสำคัญ (Key Point) ที่เคลื่อนไหวในบริเวณ แล้วนำข้อมูลเหล่านั้นมารวมกันเพื่อสร้างการแสดงผลในรูปของสามมิติ

2.1.2 การตรวจจับการเคลื่อนไหวโดยใช้กล้องวีดีโอ (Optical Motion Capture Systems)

การตรวจจับการเคลื่อนไหวโดยใช้กล้องวีดีโอนั้น เป็นการถ่ายภาพวีดีโอ เพื่อใช้ในการติดตามตำแหน่งของมาร์กเกอร์ โดยจำนวนของกล้องที่ใช้นั้น มักจะมีไม่ต่ำกว่า 4 กล้อง ถึง 32 กล้อง โดยทั่วไปกล้องจะถูกติดตั้งไว้กับที่ ไว้กับต้นกำเนิดแสง ซึ่งจะสร้างทิศทางการสะท้อนจากมาร์กเกอร์ ที่เป็นพื้นที่ที่ปิดไว้ด้วยวัสดุสะท้อนแสง เช่น เทปสี หรืออื่นๆ ตามจุดต่างๆ โดยสามารถแบ่งวิธีในการตรวจจับได้ ตามลักษณะของมาร์กเกอร์ที่ใช้ได้แก่

1) มาร์กเกอร์แบบสะท้อนแสงอินฟราเรด (Reflective Marker) ใช้แสงอินฟราเรดจากแหล่งกำเนิดแสงที่ติดไว้รอบๆ กล้องถ่ายภาพ แสงจะสะท้อนที่มาร์กเกอร์ทำให้เห็นเป็นจุดที่มีความเข้มแสงมากกว่าบริเวณอื่นๆ
2) มาร์กเกอร์แบบหลอด LED (Pulsed-LED) วิธีนี้แหล่งกำเนิดแสงจะอยู่ที่ตัวมาร์กเกอร์ การตรวจจับจะใช้วิธีวัดความเข้มแสงจากหลอด LED โดยตรง

ในการหาตำแหน่งของจุด 1 จุดใน 3 มิตินั้น จะใช้อย่างน้อยภาพ 2 ภาพในการค้นหาตำแหน่ง และกล้องหลายๆ อันนั้นจำเป็นในการรักษาระยะการมองเห็น โดยต้องใช้อย่างน้อย 2 กล้องเพื่อมองไปยังมาร์กเกอร์ทุกๆ อัน

ข้อดีในการใช้กล้องวีดีโอในการตรวจจับการเคลื่อนไหว

1) ข้อมูลของกล้องนั้นมีความถูกต้องแม่นยำมากในหลายกรณี
2) สามารถนำมาใช้ได้กับมาร์กเกอร์จำนวนมาก
3) ง่ายที่จะทำการเปลี่ยนแปลงองค์ประกอบต่างๆ ของมาร์กเกอร์
4) สามารถทำลักษณะให้ใกล้เคียงกับโครงกระดูกได้ โดยใช้กลุ่มของมาร์กเกอร์
5) สามารถทำงานได้โดยไม่ถูกจำกัดด้วยสายเคเบิล
6) มีประสิทธิภาพในการใช้พื้นที่ได้มากกว่าวิธีการอื่นๆ
7) มีความถี่ในการตรวจจับได้สูง

ข้อเสียในการใช้กล้องวีดีโอในการตรวจจับการเคลื่อนไหว

1) อุปกรณ์มีราคาแพง
2) ไม่สามารถตรวจจับการเคลื่อนไหวได้ เมื่อมาร์กเกอร์นั้นหายไปเป็นเวลานานเกินไป
3) ในการตรวจจับนั้น ต้องอยู่ภายใต้สภาพแวดล้อมที่ควบคุมไว้ โดยต้องอยู่ห่างจากแสง และแสงสะท้อนรบกวนอื่นๆ

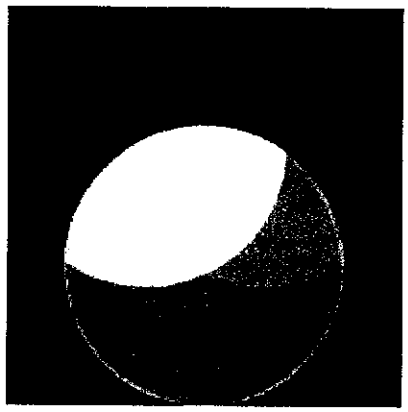
2.2. การประมวลผลภาพ

โครงงานนี้ จะนำทฤษฎีการประมวลผลภาพ มาใช้เพื่อหาตำแหน่งของมาร์กเกอร์ จากเฟรมภาพวิดีโอที่ได้บันทึกไว้ เพื่อนำตำแหน่งเหล่านี้ไปคำนวณหาตำแหน่งจริงต่อไป

2.2.1 รูปแบบสี (Color Model)

1) รูปแบบ RGB

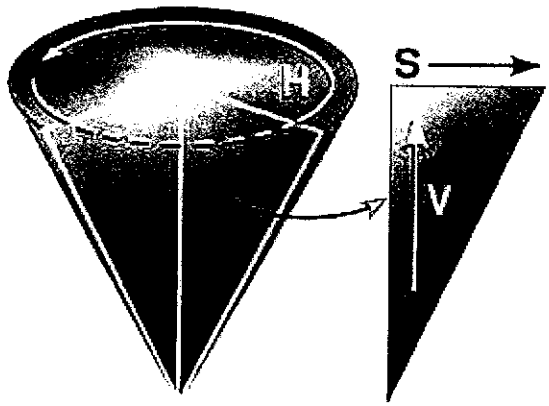
เป็นระบบสีพื้นฐานของคอมพิวเตอร์ที่ใช้ในการแสดงผล โดยจุดย่อยของภาพ (Pixel) จะประกอบด้วยค่าสี 3 ค่า คือ แดง (R) เขียว (G) และน้ำเงิน (B) การผสมสีทั้งสามนี้ด้วยค่าต่างๆ กัน จะก่อให้เกิดสีที่แตกต่างกัน โดยคอมพิวเตอร์จะเก็บค่าสีนี้แยกกัน โดยใช้ขนาดข้อมูล 1 ไบต์ต่อ 1 สี ทำให้ค่าของสีนั้นมีได้ 256 ระดับ และผสมได้สีทั้งหมด 16 ล้านสี



รูปที่ 2.1 รูปแบบสีแบบ RGB

2) รูปแบบ HSV

เป็นระบบสีที่ประกอบไปด้วยค่า 3 ค่าคือ ค่าสี (Hue) บอกความเป็นสีใดๆ ค่าความอิ่มตัวของสี (Saturation) บอกความสีหรือขาว-ดำ และค่าความสว่าง (Intensity, Value) บอกความขาวหรือดำ รูปแบบสีนี้ จะเหมาะกับการประมวลผลภาพ ที่ต้องแยกแยะสี เพราะสามารถใช้ค่า Hue เพียงค่าเดียว ก็สามารถดูความแตกต่างของสีได้



รูปที่ 2.2 รูปแบบสีแบบ HSV โดยแสดงในรูปของกรวย

2.2.2 Image Segmentation

เป็นการประมวลผลภาพเพื่อแยกวัตถุหรือส่วนของภาพที่เราสนใจ ออกมาจากส่วนอื่นๆ ซึ่งมีหลาก หลาย วิธีการตามแต่ลักษณะของภาพ และวัตถุที่ต้องการแยกออกมา สำหรับในโครงงานนี้ จะต้องแยกมาร์กเกอร์ซึ่งเป็นสีต่างๆ ออกมาจากพื้นหลังซึ่งประกอบไปด้วยสีขาวส่วนใหญ่ แล้วนำเอาไปหาตำแหน่งบนภาพ ดังนั้นจึงสามารถแบ่งได้เป็น 2 ขั้นตอน ดังนี้

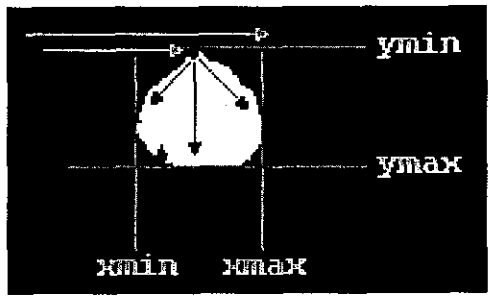
1) Threshold

เป็นการประมวลผลภาพเพื่อแยกวัตถุหรือส่วนของภาพที่เราสนใจออกมาจากส่วนอื่นๆ ซึ่งมีหลากหลายวิธีการตามแต่ลักษณะของภาพ และวัตถุที่ต้องการแยกออกมา สำหรับในโครงงานนี้ จะต้องแยกมาร์กเกอร์ซึ่งเป็นสีต่างๆ ออกมาจากพื้นหลังซึ่งประกอบไปด้วยสีขาวส่วนใหญ่ แล้วนำเอาไปหาตำแหน่งบนภาพ

ดังนั้นหลังจากการทำ Threshold แล้ว ภาพที่ได้จะเป็น Binary Image คือมีเพียงสีขาวและสีดำ โดยสีขาวจะเป็นจุดที่เราสนใจ ซึ่งต่อไปจะต้องนำไปประมวลผลหาตำแหน่งจุดที่เราสนใจ

2) Blob Analysis

เป็นการประมวลผลหาตำแหน่งของ Blob (กลุ่มของจุดสีขาว) และข้อมูลอื่นๆ สำหรับประกอบการพิจารณา โดยวิธีการ Floodfill คือจะไล่ทีละจุดสีจนกว่าจะเจอจุดสีขาวแล้วจากนั้นจะวนกระจายไปตามจุดสีขาวจนครบ (เหมือนกับการเทสีในโปรแกรมแต่งภาพ) ในระหว่างที่วนนั้นก็นับจำนวนจุดและหาขอบเขตบนและล่างเพื่อให้สามารถหาจุดกึ่งกลางได้ในภายหลัง ดังรูปที่ 2.3



รูปที่ 2.3 แสดงการทำงานของ Blob Analysis โดยใช้วิธี Floodfill

แต่เนื่องจากการทำ Threshold นั้น มักจะมีจุดที่เป็นจุดรบกวน เข้ามาด้วย ขึ้นกับช่วงสีที่ระบุว่ามีความเหมาะสมมากน้อยเพียงใด ดังนั้นการ Floodfill หาตำแหน่ง ทำให้ยังไม่สามารถจำแนกจุดที่ต้องการออกมาได้โดยตรง ทำให้ต้องมีข้อมูลอื่นเข้ามาประกอบ เช่น ขนาดของ Blob โดยอาจพิจารณาจุดที่มีขนาดเล็กว่าเป็น Noise และจุดที่มีขนาดใหญ่เป็นจุดที่เราสนใจ ก็จะทำให้สามารถหาตำแหน่งของวัตถุที่เราต้องการออกมา

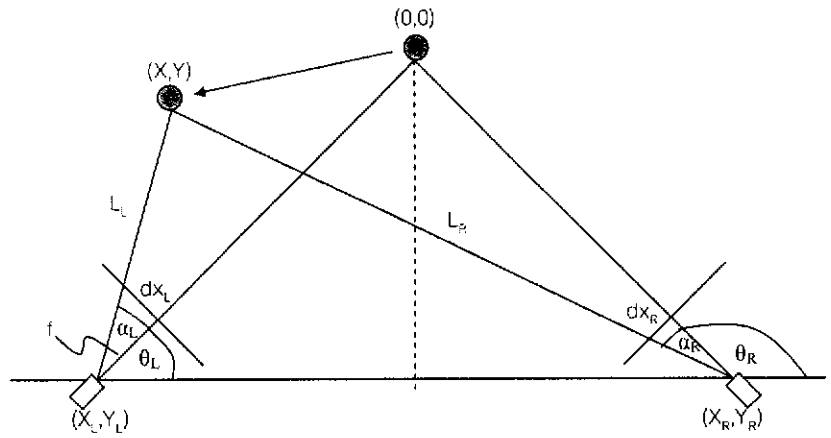
2.3. Stereopsis

Stereopsis เป็นการดึงข้อมูล 3 มิติ จาก รูปภาพ 2 มิติ โดยการใช้ทฤษฎี stereopsis แทนที่จะใช้ภาพหลายๆรูป ที่แบ่งตามเวลาแบบ การเคลื่อนไหวของวัตถุจากการเปลี่ยนตำแหน่งของวัตถุตามตำแหน่งที่ผู้มองเปลี่ยนไป (motion parallax) แต่ stereopsis ใช้ภาพหลายๆรูปใน world space

สาเหตุที่เราสามารถเห็นเป็นภาพ 3 มิติได้ เพราะตาทั้ง 2 ของคนเราทำให้เกิด stereoscopic vision ซึ่งทำให้เราเห็นความลึกได้

ฉะนั้นการคำนวณของ stereopsis จะใช้ในการหาความเหมือนกันของลักษณะทั้ง 2 ภาพ

ประโยชน์ของ stereopsis คือ สามารถหาค่าความลึกได้ง่าย เนื่องจากเรารู้ระยะห่างระหว่างกล้อง และจุดที่เส้นของสายตาตัดกัน โดยใช้การคำนวณอย่างง่ายในการหาความลึกของแกน z สำหรับตำแหน่ง(x ,y)ใดๆ ตามหลักเรขาคณิตดังรูป



รูปที่ 2.4 การหาตำแหน่งของมาร์กเกอร์โดยวิธี Stereopsis

โดยทั่วไปสามารถแบ่งลักษณะของการคำนวณหาตำแหน่งได้เป็น 2 รูปแบบ คือ การคำนวณแบบขาไปซึ่งเป็นการหาตำแหน่งจริงจากตำแหน่งบนภาพ 2 ภาพ และการคำนวณแบบย้อนกลับซึ่งเป็นการหาตำแหน่งที่จะปรากฏบนภาพจากจุดที่เรารู้ตำแหน่งจริง

2.4. Character Studio Motion File

ไฟล์ข้อมูลการเคลื่อนไหว (Motion Data File) คือ ไฟล์ที่เก็บข้อมูลลักษณะการเคลื่อนไหวของโครงกระดูก (Skeleton) ซึ่งมีใช้งานอยู่ในโปรแกรมที่ใช้สร้างตัวละครแบบ 3 มิติ ทำให้ผู้ใช้งานสามารถสร้างลักษณะการเคลื่อนไหวของตัวละครที่ออกแบบไว้แล้วได้อย่างง่ายดาย โดยที่ไม่ต้องคอยกำหนดตำแหน่งของร่างกายของตัวละครในแต่ละเฟรมของการสร้าง Animation ของตัวละครนั้นๆ เอง แต่หันมาใช้ไฟล์ข้อมูลการเคลื่อนไหวมาช่วยในการสร้างการเคลื่อนไหวที่ต้องการ โดยการเก็บเอาลักษณะการเคลื่อนไหวจากคนหรือโมเดลที่มีรูปร่างใกล้เคียงกับตัวละครที่สร้างไว้แทน ด้วยการติดมาร์กเกอร์ไว้ตามตำแหน่งที่ต้องการ และมีความสอดคล้องกับตัวละคร 3 มิติที่สร้างขึ้น แล้วจึงนำข้อมูลที่ได้นั้นไปใช้ในการกำหนดการเคลื่อนไหวของตัวละครอีกทีหนึ่ง สำหรับการทำงานในลักษณะนี้นั้นคือการสร้างโครงกระดูกสำเร็จรูป (Biped) ซึ่งจะทำให้สามารถสร้างการเคลื่อนไหวของตัวละครได้อย่างเร็ว และยังมีความสมจริงมากขึ้นอีกด้วย

สำหรับการเก็บข้อมูลของตำแหน่งของมาร์กเกอร์แต่ละจุดนั้น จะได้มาจากการคำนวณตามทฤษฎี Stereopsis และไฟล์ข้อมูลที่ใช้นั้นมีโครงสร้างอยู่หลายรูปแบบด้วยกัน ซึ่งลักษณะโครงสร้างนั้นจะแตกต่างกัน แต่ที่เป็นที่นิยมใช้กันได้แก่ ไฟล์สกุล *.csm, *.bvh, *.trd, *.c3d และ *.fbx ซึ่งในที่นี้จะอธิบายเฉพาะไฟล์ที่อยู่ในรูปไฟล์ที่มีสกุลเป็น *.csm และ *.bvh เนื่องจากไฟล์ประเภทนี้นั้น เป็นไฟล์มาตรฐานของ Character Studio Marker

2.4.1 ไฟล์ CSM

เป็นไฟล์ข้อมูลที่จะเก็บอยู่ในรูปของ ASCII-based และเก็บเพียงแค่ค่าตำแหน่ง x, y และ z ของมาร์กเกอร์แต่ละอัน ที่ติดอยู่กับผู้แสดงไว้เท่านั้น ทำให้โครงสร้างของไฟล์นั้นเก็บได้ง่าย และรวดเร็วในการตอบสนองอีกด้วย ตัวอย่างดังรูป

สำหรับการแปลข้อมูลจากไฟล์ข้อมูลการเคลื่อนไหวแบบ CSM ที่ใช้ในการสร้างการเคลื่อนไหวให้กับตัวละครนั้น ข้อมูลที่ได้จากการแปลไฟล์ประเภทนี้จะไม่มีการกำหนดความสัมพันธ์ของมาร์กเกอร์แต่ละตัว แต่จะอ่าน และอ้างอิงค่าต่างๆ ตามลำดับที่ได้กำหนดไว้เท่านั้น จะไม่มีการเปลี่ยนอัตราส่วนของข้อมูลใดๆ ทั้งสิ้น โดยข้อมูลที่ใส่ไว้นั้นจะเป็นข้อมูลของตำแหน่งมาร์กเกอร์ใน World Space หากต้องมีการเปลี่ยนแปลงอัตราส่วนของข้อมูล ผู้ที่นำไฟล์ไปใช้จะต้องเป็นคนเปลี่ยนแปลงอัตราส่วนของข้อมูลเอง

```
$Date 21/02/2549
$Time 2:56:22
$Filename C:\Documents and Settings\USER\Desktop\Tor1\ccc.csm
$Actor Noname

$Comments
Created by Motion Analyser : 3D Character Animation Studio v2.2

$FirstFrame 1
$LastFrame 51

$Rate 25

$Order
LFHD LBHD RFHD RBHD C7 CLAV ........................ LHEL LMT5 LMTI RKNI RHEL
RMT5 RMTI

$Points
1  49.01 -103.79 933.15  49.00 35.17 941.22  -21.00 ........ -51.99 -71.78 -192.73
2  49.01 -103.79 933.15  49.00 35.17 941.22  -21.00 ........ -51.99 -71.78 -192.73
3  49.01 -103.79 933.15  49.00 35.17 941.22  -21.00 ........ -51.99 -71.78 -192.73
4  49.01 -103.79 933.15  49.00 35.17 941.22  -21.00 ........ -51.99 -71.78 -192.73
5  49.01 -103.79 933.15  49.00 35.17 941.22  -21.00 ........ -51.99 -71.78 -192.73
6  49.01 -103.79 933.15  49.00 35.17 941.22  -21.00 ........ -51.99 -71.78 -192.73
   .        .                .                .                      .
   .        .                .                .                      .
   .        .                .                .                      .
   .        .                .                .                      .
   .        .                .                .                      .
   .        .                .                .                      .
   .        .                .                .                      .
   .        .                .                .                      .
   .        .                .                .                      .
   .        .                .                .                      .
   .        .                .                .                      .
   .        .                .                .                      .
50 49.01 -103.79 933.15  49.00 35.17 941.22  -21.00 ........ -51.99 -71.78 -192.73
51 49.01 -103.79 933.15  49.00 35.17 941.22  -21.00 ........ -51.99 -71.78 -192.73
52 49.01 -103.79 933.15  49.00 35.17 941.22  -21.00 ........ -51.99 -71.78 -192.73
53 49.01 -103.79 933.15  49.00 35.17 941.22  -21.00 ........ -51.99 -71.78 -192.73
54 49.01 -103.79 933.15  49.00 35.17 941.22  -21.00 ........ -51.99 -71.78 -192.73
55 49.01 -103.79 933.15  49.00 35.17 941.22  -21.00 ........ -51.99 -71.78 -192.73
56 49.01 -103.79 933.15  49.00 35.17 941.22  -21.00 ........ -51.99 -71.78 -192.73
57 49.01 -103.79 933.15  49.00 35.17 941.22  -21.00 ........ -51.99 -71.78 -192.73
58 49.01 -103.79 933.15  49.00 35.17 941.22  -21.00 ........ -51.99 -71.78 -192.73
59 49.01 -103.79 933.15  49.00 35.17 941.22  -21.00 ........ -51.99 -71.78 -192.73
60 49.01 -103.79 933.15  49.00 35.17 941.22  -21.00 ........ -51.99 -71.78 -192.73
```

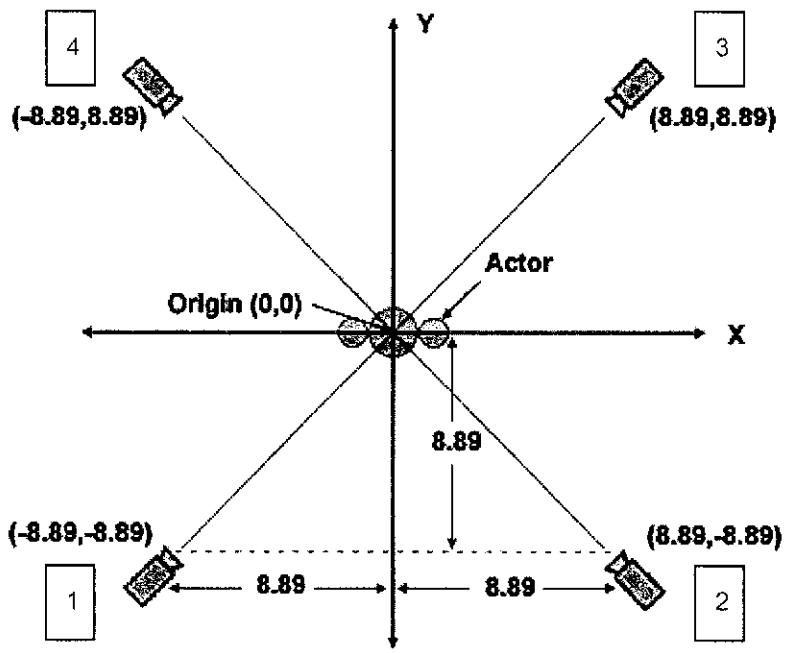
รูปที่ 2.5 แสดงข้อมูลไฟล์ CSM

2.4.2 ไฟล์ BVH

เป็นไฟล์ข้อมูลแบบไบโอวิชัน (Biovision) โปรแกรม 3D studio Max และอื่นๆ สามารถนำเอาไฟล์ชนิดนี้ไปใช้งานได้ โดย ไฟล์ชนิดนี้จะประกอบไปด้วยการจัดลำดับของข้อมูล (hierarchy) โดยไฟล์จะถูกแบ่งเป็น 2 ส่วนใหญ่ๆ คือส่วน การจัดลำดับข้อมูล (hierarchy) และส่วนของการเคลื่อนไหว (Motion) โดยส่วนของการจัดลำดับข้อมูล เป็นส่วนที่ประกอบด้วยส่วนทุกๆส่วน ที่เป็นตัวบอกในการสร้างโครงกระดูกในโปรแกรมเคลื่อนไหว และส่วนของการเคลื่อนไหวจะเก็บข้อมูลการเคลื่อนไหว

2.5. การออกแบบการทำงาน

2.5.1 การวางตำแหน่งของกล้อง



รูปที่ 2.6 แสดงการจัดวางตำแหน่งของกล้อง

การวิเคราะห์หาตำแหน่งจากคู่กล้องจะใช้ทฤษฎี stereopsis ในการหาตำแหน่งของมาร์กเกอร์ (x, y, z) โดยจะทำการคำนวณหาทีละคู่กล้องดังนี้

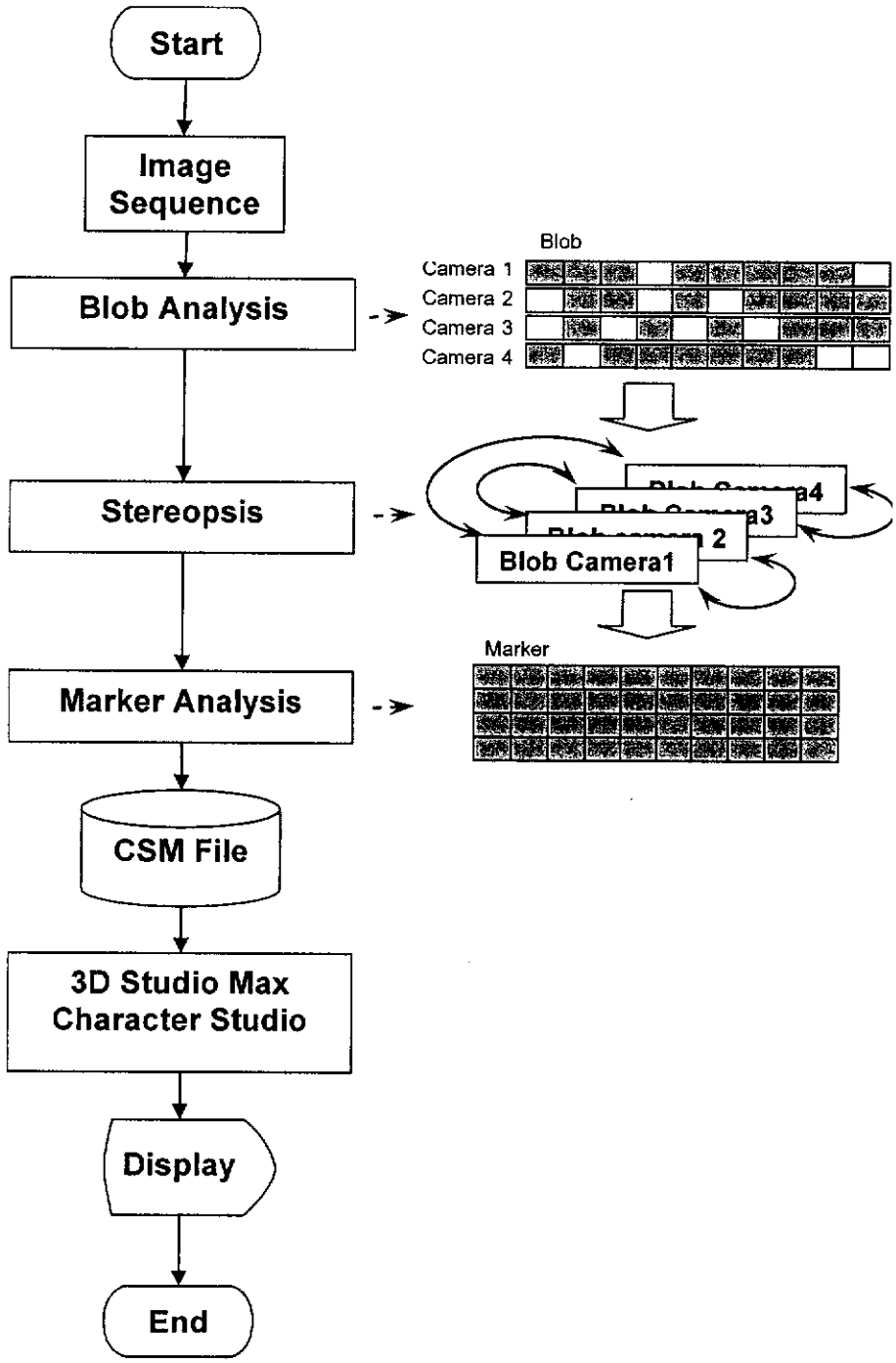
- กล้องคู่ที่ 1 จะใช้กล้องที่ 1 และกล้องที่ 2
- กล้องคู่ที่ 2 จะใช้กล้องที่ 2 และกล้องที่ 3
- กล้องคู่ที่ 3 จะใช้กล้องที่ 3 และกล้องที่ 4
- กล้องคู่ที่ 4 จะใช้กล้องที่ 1 และกล้องที่ 4

คู่กล้องที่เลือกไว้นั้น ถ้าเห็นจุด Blob นั้นๆ ทั้งคู่แล้วจะทำการ Stereopsis เก็บไว้

หลังจากการทำ Stereopsis แล้ว จะได้ตำแหน่งมาร์กเกอร์ต่างๆ แล้ว หากยังมีมาร์กเกอร์ใดที่ยังไม่ทราบตำแหน่งอีก แสดงว่า กล้องไม่สามารถมองเห็นได้เลย ดังนั้น มาร์กเกอร์ตัวนั้นไม่สามารถหาได้จากการ Stereopsis ดังนั้น จะทำการระบุตำแหน่งมาร์กเกอร์นั้นๆ เอง โดยการใช้ตำแหน่งของมาร์กเกอร์อื่น มาช่วยในการระบุตำแหน่งแทน

2.5.2 การทำงานของระบบ

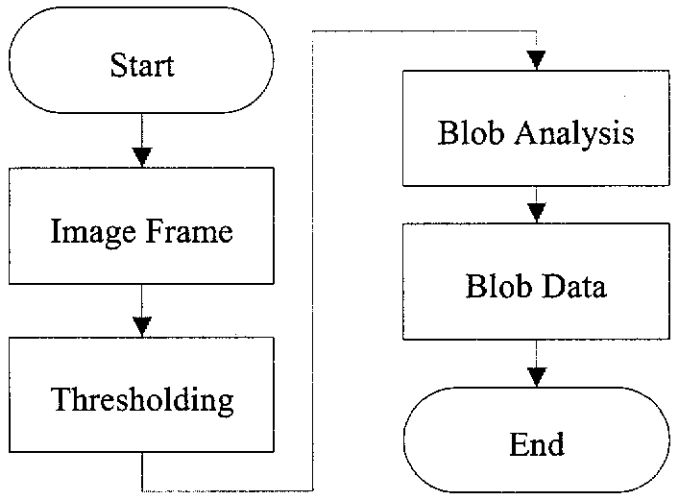
สำหรับโปรแกรมวิเคราะห์การเคลื่อนไหวนั้น จะเป็นโปรแกรมสำหรับหาตำแหน่งของข้อต่อต่างๆจากภาพวิดีโอทั้งสี่ภาพในทุกๆเฟรม เพื่อสร้างข้อมูลการเคลื่อนไหวแล้วบันทึกลงไฟล์การเคลื่อนไหว CSM ดังนั้นโปรแกรมจะแบ่งออกเป็น 3 ขั้นตอนหลักๆ คือขั้นตอนแรกเป็นการเปิดไฟล์วิดีโอขึ้นมา แล้วทำการประมวลผลเพื่อหาตำแหน่งของ blob ในทุกๆ จุดสีและทุกๆ เฟรมจากกล้องทั้งหมด 4 กล้อง ซึ่งจะมีการใช้อัลกอริธึมและวิธีการต่างๆ ในการช่วยการตรวจจับ หลังจากนั้นเมื่อตรวจจับตำแหน่งของ blob ได้หมดแล้ว ก็จะนำไปคำนวณเพื่อหาตำแหน่งของข้อต่อ (marker) ต่างๆ และบันทึกลงไฟล์ CSM ต่อไป



รูปที่ 2.7 แสดงการทำงานของระบบ

2.5.3 การวิเคราะห์หาจุด Blob บนไฟล์ Video

ขั้นตอนการวิเคราะห์หาตำแหน่งจุด Blob มาจาก การประมวลผลภาพจากแต่ละกล้องวีดีโอ (ทั้งหมดมี 4 กล้อง) ทีละเฟรมแล้ววนทำงานจนครบทุกเฟรมในไฟล์วีดีโอ โดยทำการหาจากจุดสีที่มีสีเดียวกับจุด blob ที่ต้องการ โดยการ threshold เอาเฉพาะจุดสีที่ต้องการนั้นๆ ออกมาก่อนแล้วทำการ floodfill เลือกมาทีละจุดสี แล้วนำมาประมวลผล Blob Analysis จนได้ตำแหน่งของจุดสี Blob ออกมา โดยมีขั้นตอนดังนี้



รูปที่ 2.8 แผนผังแสดงการประมวลผลภาพเพื่อหาตำแหน่งของมาร์กเกอร์

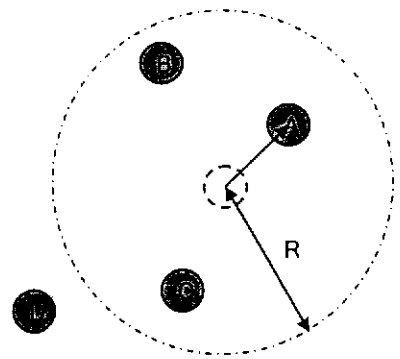
จากรูปที่ 2.8 จะสามารถแบ่งขั้นตอนการประมวลผลได้ 3 ขั้นตอน ดังต่อไปนี้

1) การทำ Threshold โดยจะทำในช่วงสี HSV ทั้งหมด 4 ช่วง เนื่องจากมีมาร์กเกอร์ที่ใช้ทั้งหมด 4 สี คือ Blue, Pink, Green, Yellow โดยมีการตั้งค่าไว้ดังนี้

- Blue - มีค่า Hue = 230-260
  - มีค่า Sat = 50-255
  - มีค่า Val = 55-255
- Red - มีค่า Hue = 270-340
  - มีค่า Sat = 45-255
  - มีค่า Val = 75-255
- Green - มีค่า Hue = 90-155
  - มีค่า Sat = 35-255
  - มีค่า Val = 40-255
- Yellow - มีค่า Hue = 0-45
  - มีค่า Sat = 25-255
  - มีค่า Val = 70-255

สุดท้ายจะได้ภาพแบบขาวดำ มาทั้งหมด 4 ภาพตามจำนวณชุดของจุดสีที่ทำ Threshold ไว้

2) จากนั้นนำภาพทั้งสี่ไปหาตำแหน่งของจุดสีแต่ละจุด



รูปที่ 2.9 แสดงการติดตามมาร์กเกอร์

3) เป็นกระบวนการเลือก Blob เฉพาะที่เราสนใจออกมา โดยใช้วิธีการติดตามมาร์กเกอร์แต่ละตัว (Tracking) โดยจะดูตำแหน่งของมาร์กเกอร์จากเฟรมก่อนหน้าว่าอยู่ตำแหน่งใด จากนั้นจะเลือกจุด Blob ที่มีขนาดใหญ่อยู่ในช่วงที่กำหนด ที่อยู่ใกล้ตำแหน่งเดิมมากที่สุด แต่ต้องไม่เกินรัศมีที่กำหนด ดังรูปที่ 2.9

จากรูปจุด Blob ที่ใกล้ที่สุดคือจุด A ส่วนสำหรับจุด B และ C ซึ่งอยู่ห่างออกไปจะไม่ถูกตรวจจับ และสำหรับจุด D ซึ่งอยู่นอกเหนือขอบเขต R จะไม่นำมาคิดเช่นกัน ในที่นี้ในโปรแกรมผู้พัฒนาได้กำหนดค่ารัศมีการตรวจจับให้มีค่าเท่ากับ 40 พิกเซล

ในการหาตำแหน่งจุดสีในครั้งแรกนั้นจะเทียบกับตำแหน่งเริ่มต้นที่ได้ทำการกำหนดไว้เพื่อสะดวกในการใช้งาน (สามารถแก้ไขการตั้งจุดสีเริ่มแรกเองได้)

การกำหนดรัศมี R จะมีผลต่อการติดตามมาร์กเกอร์ การกำหนด R ให้น้อย จะช่วยทำให้ลดปัญหาการตรวจจับผิดพลาดได้ แต่ก็มีข้อเสียที่สำคัญคือ จะไม่สามารถติดตามมาร์กเกอร์ที่เคลื่อนไหวเร็วมากได้ แต่ถ้าหากใช้ R กว้างเกินไปก็อาจจะทำให้ไปตรวจจับจุดอื่นๆ ที่ไม่ใช่จุดที่สมควรได้ ซึ่งถ้าหากเป็นจุดรบกวน (Noise Blob) จำเป็นที่จะต้องปรับช่วงสีเพื่อลดจุดรบกวนออกไปในขั้นการทำ Threshold แต่หากเป็นจุดอื่นๆ ที่อยู่ใกล้กันและมีสีเดียวกัน เช่น ไหล่กับข้อศอก หากจุดที่ไหล่หายไป และเมื่อทำการตรวจจับจุดที่ไหล่อาจจะไปจับเอาจุดที่ข้อศอกแทน ซึ่งจะแก้ไขโดยการเปรียบเทียบระยะห่าง โดยเมื่อไปหาตำแหน่งของข้อศอกอีกครั้ง และพบว่าจุดเดียวกันที่เคยตรวจจับเป็นไหล่นั้นไปแล้วเป็นจุดของข้อศอกด้วย จะทำการเปรียบเทียบระยะทางจากจุดเดิมมาจุดใหม่ หากจุดใดมีระยะทางน้อยกว่าแสดงว่าจุด Blob นั้นน่าจะเป็นจุดของมาร์กเกอร์นั้นมากกว่า และทำให้จุดเดิมนั้นเป็นจุดที่ไม่สามารถตรวจจับได้แทน ซึ่งต้องมีการแก้ไขในหัวข้อต่อๆ ไป

Camera 1

Blob #0

x, y, value	x, y, value	x, y, value	.....	x, y, value	x, y, value
Frame 0	Frame 1	Frame 2		Frame N-1	Frame N

Blob #1

x, y, value	x, y, value	x, y, value	.....	x, y, value	x, y, value
Frame 0	Frame 1	Frame 2		Frame N-1	Frame N

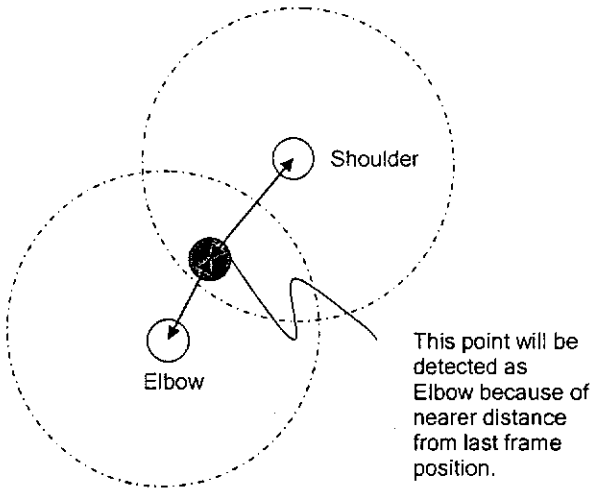
Blob #23

x, y, value	x, y, value	x, y, value	.....	x, y, value	x, y, value
Frame 0	Frame 1	Frame 2		Frame N-1	Frame N

รูปที่ 2.10 แสดงตารางจุดสีของภาพจาก กล้องตัวที่ 1

จากรูป เป็นการแสดงข้อมูลของตารางจุด Blob ของกล้อง Camera 1 โดยมีการเก็บค่าจุด blob ทั้งหมด 24 ตัว (ตามจำนวณจุดสีที่เราใช้) โดยในแต่ละตัวจะเก็บข้อมูลในแต่ละเฟรมไว้เป็นค่า x, y, value

2.5.4 การระบุตำแหน่งเมื่อจุดสีเดียวกันมาซ้อนกัน



รูปที่ 2.11 แสดงการเกิดการซ้อนทับกันของ Blob

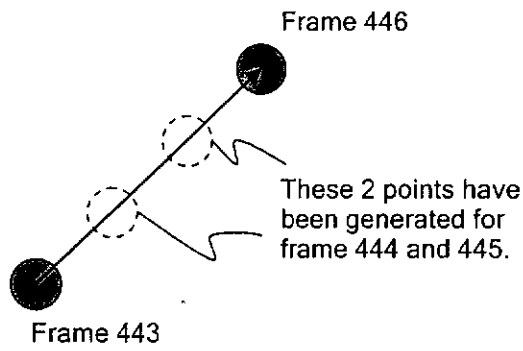
วิธีในการระบุตำแหน่งจะทำการดูกล้องด้านข้างทั้ง 2 ด้าน โดยหากเฟรมก่อนหน้า Blob ยังอยู่ทั้ง 2 กล้อง และเฟรมปัจจุบันยังอยู่ทั้ง 2 กล้อง แต่กล้องตัวเองมองไม่เห็น จะสรุปได้ว่ามีการซ้อนทับกันของจุดสีนั้นๆ จะทำการใส่ตำแหน่งให้กับ blob นั้นด้วยค่าตำแหน่งที่ซ้อนกันเลย เช่น กล้องตัวที่ 1 เกิดการซ้อนทับกัน จะใช้กล้องตัวที่ 2 และกล้องตัวที่ 4 เป็นกล้องด้านข้างของทั้ง 2 ด้าน

2.5.5 การแก้ไขตำแหน่งเมื่อตรวจไม่พบมาร์กเกอร์

ในการติดตามมาร์กเกอร์นั้น จะช่วยในการหาตำแหน่งและแยกแยะจุด Blob แต่ละตัวได้ แต่อาจเกิดเหตุการณ์ที่ไม่สามารถตรวจพบจุด Blob ได้ ซึ่งการตรวจจับจุด Blob นั้นไม่สามารถดำเนินต่อไปได้ ถึงแม้ว่าจุด Blob นั้นสามารถพบได้ในเฟรมต่อมาก็ตาม เพราะการตรวจจับการติดตามมาร์กเกอร์นั้นต้องใช้ตำแหน่งของจุด Blob ของเดิม (เฟรมก่อนหน้า) ดังนั้นจึงสามารถแก้ปัญหานี้ได้ด้วยวิธีการละเว้นได้

การละเว้นคือการให้การตรวจจับดำเนินต่อไป แม้ว่าจุด Blob นั้นไม่สามารถตรวจจับได้ และไปตรวจจับหาในเฟรมต่อๆ ไปแทน โดยเราจะใช้ตำแหน่งสุดที่ที่ยังสามารถตรวจจับได้เป็นจุดอ้างอิง เมื่อเราพบจุด Blob ในเฟรมถัดๆ ไป ดังนั้นจะทำการทำการสร้างค่าโดยการเฉลี่ยระยะทางที่เปลี่ยนไปกับจำนวนเฟรมที่ละเว้นไว้ แต่วิธีนี้สามารถช่วยได้ในเหตุการณ์ที่ไม่สามารถตรวจจับได้น้อยเฟรมเท่านั้น เพราะหากละเว้นไว้เป็นระยะเวลานาน หรือหลายเฟรมเกินไป ตำแหน่งของจุด Blob อาจเปลี่ยนไปมากเกินจนไม่สามารถตรวจจับได้ หรือตรวจจับได้จุด Blob ผิดไป ดังนั้น หากมีการเคลื่อนไหวระหว่างเฟรมที่ไม่สามารถตรวจจับได้เพียง 2-3 เฟรม โดย

ส่วนมากการเคลื่อนไหวใน 2-3 เฟรมนั้น มักมีลักษณะเป็นเส้นตรง ดังนั้นการสร้างจุดที่ขาดหายไปด้วยสมการเส้นตรงนั้น จะให้ผลได้ใกล้เคียงและชดเชยได้



รูปที่ 2.12 แสดงการแก้ไขจุดมาร์กเกอร์ที่ตรวจจับไม่พบ

แต่หากใช้วิธีนี้แล้วยังไม่สามารถกลับมาตรวจจับพบจุด Blob นั้นได้ ระบบจะทำการหยุดการตรวจจับจุด Blob นั้นไป โดยระบบจะกำหนดให้ละเว้นได้สูงสุดไม่เกิน 2 เฟรม ซึ่งจากการทดสอบพบว่าเป็นระยะที่ตำแหน่งของจุด Blob ส่วนมากยังเคลื่อนที่อยู่ในระยะที่ยังเคลื่อนที่ไปไม่มากจนเกินไป

2.5.6 เงื่อนไขในการกำหนด หรือ เลือกจุดตำแหน่ง

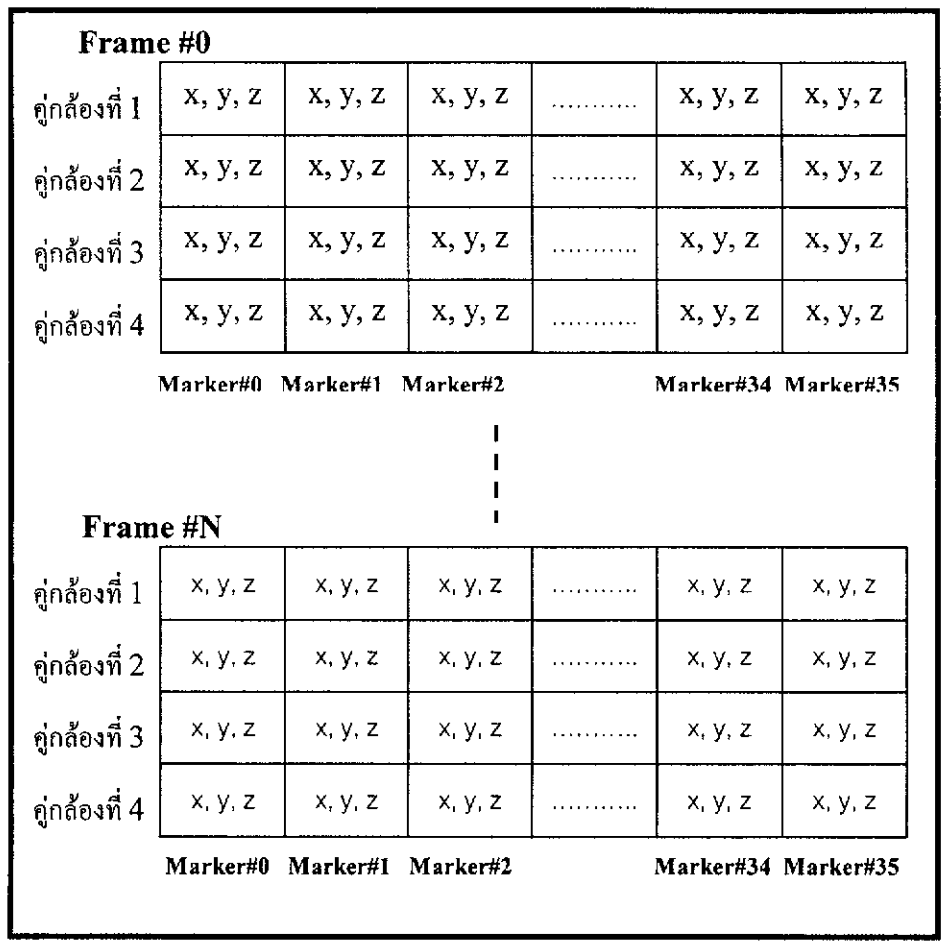
การกำหนด หรือเลือกจุดตำแหน่งของ Marker (x, y, z) ทำโดยการพิจารณาข้อมูลของ Blobที่ได้มาจากการ Stereopsis ที่ได้มาจากคู่กล้องต่างๆ โดยจะนำมาพิจารณาดังนี้

- **ได้ค่า Stereopsis ทั้งหมด 1 ค่า**

  จะทำการเลือกเป็นตำแหน่งของมาร์กเกอร์เลย

- **ได้ค่า Stereopsis ทั้งหมดมากกว่าหรือเท่ากับ 2 ค่า**

  จะทำการหาระยะห่าง ระหว่างจุดมาร์กเกอร์ในเฟรมก่อนหน้า และจุดมาร์กเกอร์ในเฟรมปัจจุบัน ในแนวแกน x, แกน y และแกน z แล้วนำตำแหน่งที่มีระยะห่างในแต่ละแนวแกนที่ใกล้เคียงกับระยะในแนวแกนต่างๆ ที่เราทำไว้ของเฟรมที่แล้วมากที่สุด มาเป็นค่าตำแหน่ง marker จริง



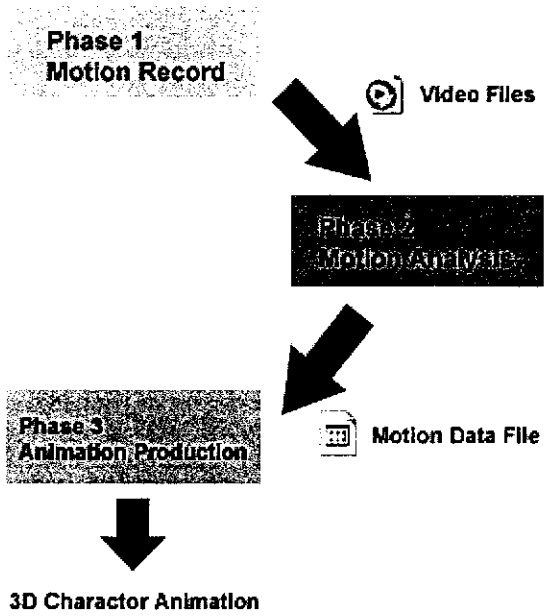
รูปที่ 2.13 แสดงตารางจุดมาร์กเกอร์

จากรูป เป็นการแสดงข้อมูลของตารางจุดมาร์กเกอร์ โดยมีการเก็บค่าของเฟรมทุกเฟรม โดยในแต่ละเฟรมจะเก็บข้อมูลของมาร์กเกอร์ของแต่ละคู่กล้อง (4 คู่กล้อง) ไว้เป็นค่า x, y, z ตามจำนวนมาร์กเกอร์ทั้งหมด (36 ตำแหน่ง)

บทที่ 3

การทดสอบระบบ

3.1 ภาพรวมของระบบ(System Overview)



รูปที่ 3.1 แผนผังแสดงการทำงานของระบบ

1. การบันทึกภาพการเคลื่อนไหว (Motion Record) เป็นส่วนที่ต้องจัดเตรียมกล้อง สถานที่ และผู้แสดง (Actor) เพื่อทำการบันทึกเป็นภาพวิดีโอจากกล้องทั้ง 4 ตัวเป็นไฟล์ชนิด AVI ซึ่งอาจจะมีการเข้ารหัสเพื่อบีบอัดข้อมูลในรูปแบบต่างๆ เช่น DIVX เป็นต้น
2. การวิเคราะห์หาการเคลื่อนไหว (Motion Analysis) โดยใช้กระบวนการการประมวลผลภาพเพื่อหาตำแหน่งมาร์กเกอร์แต่ละจุดจากไฟล์วิดีโอที่ได้จากข้อ 1 ในแต่ละเฟรม จากนั้นจะทำการจับคู่เพื่อนำไปคำนวณหาตำแหน่งจริง (World Coordinate) แล้วบันทึกลงไฟล์บันทึกการเคลื่อนไหว (Motion Data File) ชนิด CSM โดยมีการปรับแต่งแก้ไขข้อมูลในขั้นนี้ด้วย
3. การสร้างการเคลื่อนไหวให้กับตัวละคร (Animation Production) เป็นขั้นตอนที่จะนำเอาไฟล์บันทึกการเคลื่อนไหว (Motion Data File) ไปใส่ให้กับตัวละคร 3 มิติในโปรแกรม 3ds Studio Max เพื่อสร้างการเคลื่อนไหวให้กับตัวละคร

3.2 การบันทึกภาพการเคลื่อนไหว (Motion Record)

ในส่วนการบันทึกภาพการเคลื่อนไหวจะเป็นการจำลอง (Simulation) ด้วยโปรแกรม 3ds Studio Max โดยการสร้างผู้แสดงสมมุติขึ้นมา จากนั้นจึงสร้างกล้องเพื่อทำการจับภาพแล้วบันทึกลงไฟล์ตามรูปแบบการเข้ารหัสที่ได้กำหนดไว้ ซึ่งในส่วนรายละเอียดของการบันทึกภาพเคลื่อนไหวจะมีดังต่อไปนี้

- อัตราภาพ (Frame rate) เท่ากับ 25 เฟรมต่อวินาที
- ขนาดภาพที่บันทึกนั้น จะใช้ขนาด 320x240 พิกเซล
- ไฟล์ฟอร์แมตที่ใช้ในการบันทึก จะใช้ไฟล์ AVI ที่มีการเข้ารหัสแบบ DIVX

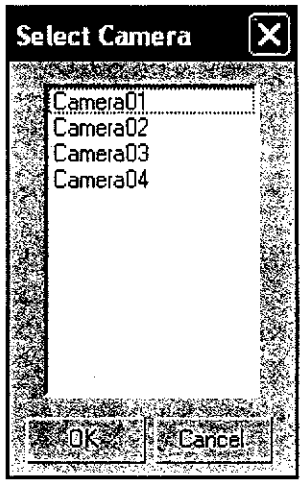
3.2.1 การสร้างและวางตำแหน่งกล้อง

ในโปรแกรม 3ds max จะให้กล้องมาใช้งาน 2 ชนิดคือ

- **Target Camera** เป็นกล้องที่ถูกสร้างมาพร้อมตัว Target สำหรับล็อคเป้าหมาย สามารถควบคุมได้ทั้งตำแหน่งของตัวกล้อง และตัว Target
- **Free Camera** เป็นกล้องที่ถูกสร้างเอาไว้ลอยๆ สามารถหมุนและเลื่อนไปมาได้อย่างอิสระ โดยไม่มีการล็อคเป้าหมายเหมือน Target Camera

โดยในโครงงานนี้จะใช้กล้องแบบ Target Camera ซึ่งมีขั้นตอนการสร้างและวางตำแหน่งกล้องดังต่อไปนี้

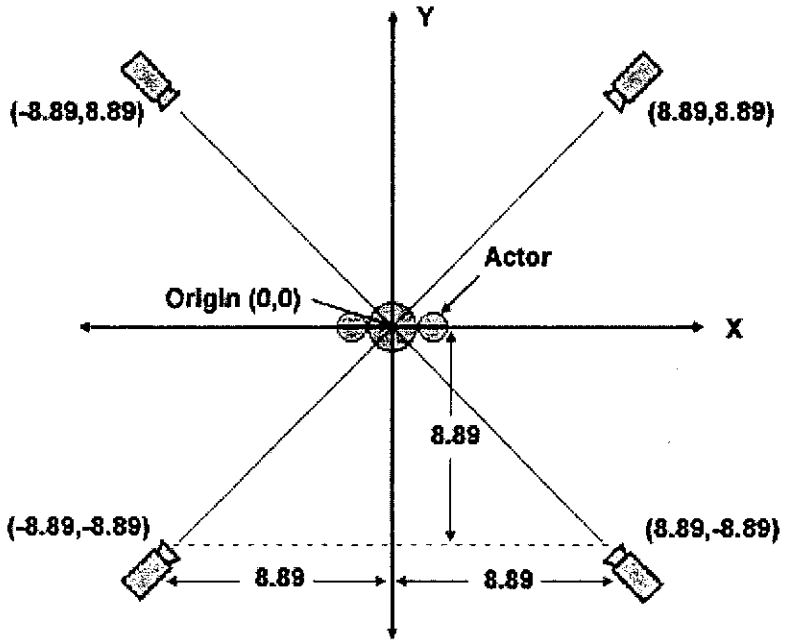
1. ภายใน (Create Panel) > (Camera) คลิกปุ่ม เพื่อสร้างกล้องแบบ Target Camera
2. ภายใน Viewport ต่างๆ แดรกเมาส์กำหนดทิศทางของกล้อง ให้ตำแหน่งที่วางกล้องและเป้าหมายที่ชี้ไป เพื่อให้ได้มุมมองที่ต้องการ
3. เปลี่ยน Viewport Perspective ให้เป็น Viewport Camera ที่มองจากกล้องด้วยการคลิกขวาเลือก Viewport Perspective และกดปุ่ม <C> บนคีย์บอร์ด เพื่อแสดงมุมมองจากกล้องที่เราสร้างไว้ โดยในกรณีที่มีกล้องมากกว่า 1 ตัว โปรแกรมจะมีหน้าต่าง Select Camera ปรากฏขึ้นมาให้เราเลือกกล้องสำหรับเปลี่ยน Viewport ก็ให้คลิกเลือกกล้องตัวที่ต้องการใช้งานแล้วคลิกปุ่ม OK ดังรูปที่ 3.2
4. เราสามารถเปลี่ยนมุมมองที่สร้างไว้ก่อนหน้านี้ โดยการเปลี่ยนตำแหน่งกล้องได้ใหม่ตามที่ต้องการ ซึ่งภาพที่ปรากฏใน Viewport Camera จะเปลี่ยนตามตำแหน่งของกล้องตลอดเวลา



รูปที่ 3.2 หน้าต่าง Select Camera

3.2.2 การติดตั้งกล้องบันทึกภาพการเคลื่อนไหว

โครงงานนี้จะติดตั้งกล้องจำนวน 4 ตัวให้อยู่บนระนาบเดียวกัน โดยกล้องทั้ง 4 ตัวจะอยู่บริเวณด้านหน้าซ้าย ด้านหน้าขวา ด้านหลังซ้าย และด้านหลังขวา และหันให้แกนกลางมาตัดกันที่จุดกำเนิดพอดี ดังรูปที่ 3.3



รูปที่ 3.3 การวางตำแหน่งกล้อง

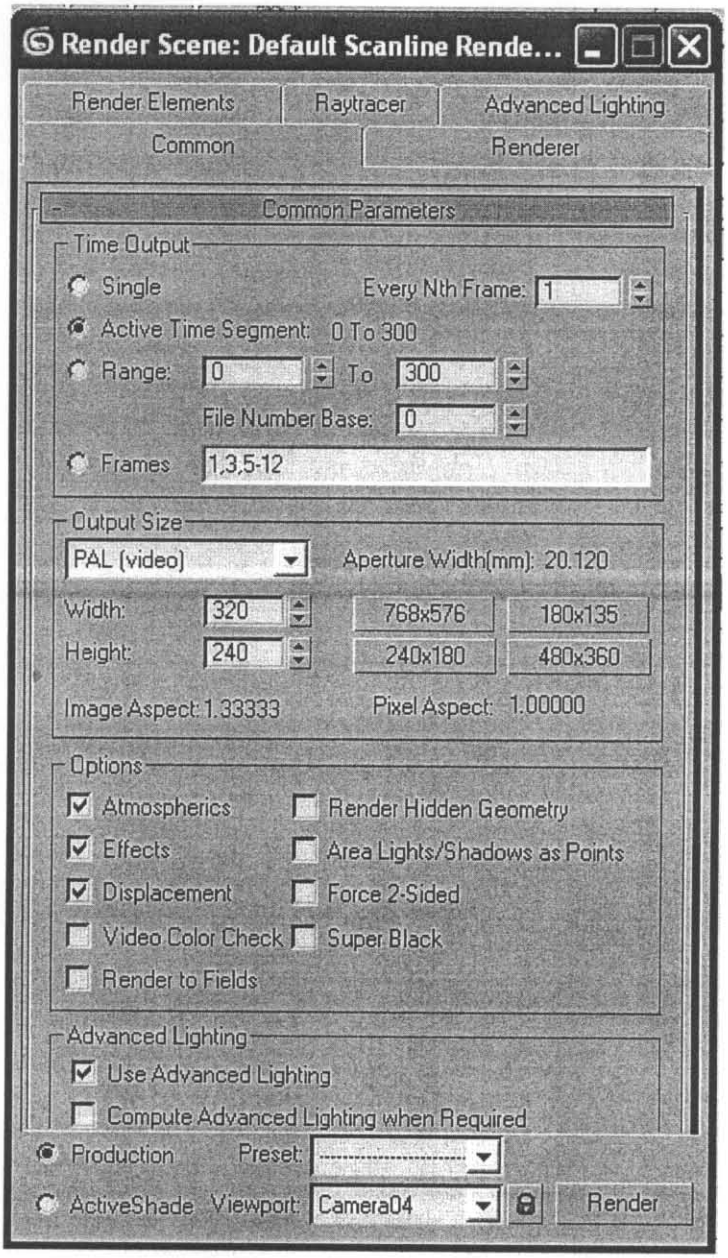
โดยการวางตำแหน่งกล้องที่ดีที่สุดจำเป็นจะต้องให้เห็นผู้แสดง (Actor) ได้ทั้งตัว รวมถึงต้องเผื่อบริเวณไว้ส่วนหนึ่งสำหรับการเคลื่อนไหวร่างกายด้วย ในที่นี้จะเลือกใช้ระยะห่างในแนวแกน Y 8.89 เมตร ระยะห่างในแนวแกน X 8.89 เมตร และความสูงจากพื้น 1.27 เมตร สำหรับมุมการวางกล้องนั้นจะใช้มุม 45 องศา เพราะจะทำให้ได้มุมมองบริเวณในแนวกว้างและลึกเท่าๆกัน ไม่แคบด้านใดด้านหนึ่ง ซึ่งจะทำให้สามารถเคลื่อนไหวในบริเวณที่กว้างกว่ามุมอื่นๆ

3.2.3 การบันทึกภาพเคลื่อนไหว

การบันทึกภาพเคลื่อนไหว ด้วยโปรแกรม 3ds Studio Max หรือการเรนเดอร์ภาพเคลื่อนไหว จะแบ่งออกเป็น 2 แบบใหญ่ๆคือ เรนเดอร์ภาพเคลื่อนไหวออกมาเป็นไฟล์เดียว และเรนเดอร์ออกมาเป็นภาพที่เรียงต่อกันไปตามลำดับโดยแยกไฟล์รูปภาพแต่ละเฟรมออกจากกัน หรือเรียกกันว่าการเรนเดอร์แบบ Sequence โดยในโครงงานนี้จะใช้แบบเรนเดอร์ภาพเคลื่อนไหวออกมาเป็นไฟล์เดียว ซึ่งมีขั้นตอนการทำงานดังต่อไปนี้

1. เปลี่ยน Viewport name ให้เป็น View ที่ต้องการบันทึกภาพเคลื่อนไหว โดยในกรณีที่ต้องการบันทึกภาพจากกล้องที่ได้สร้างไว้จะใช้ Camera View

2. คลิกคำสั่ง Rendering > Render... หรือกดปุ่ม <F10> บนคีย์บอร์ด เพื่อเปิดหน้าต่าง Render Scene



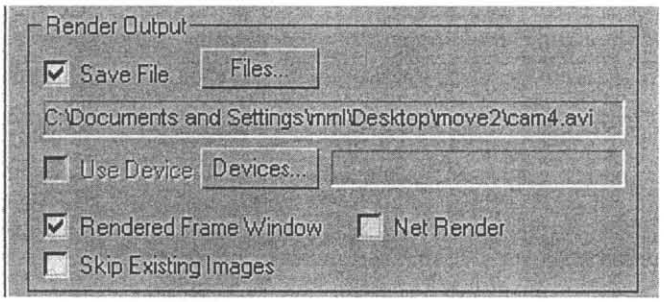
รูปที่ 3.4 หน้าต่าง Render Scene

3. การกำหนดเฟรมที่ต้องการเรนเดอร์จากช่อง Time Output โดยคลิกเลือกรายละเอียดต่อไปนี้

- **Active Time Segment** เพื่อเรนเดอร์ภาพจากเฟรมที่มีอยู่ทั้งหมดตั้งแต่ต้นจนจบ
- **Range** เพื่อกำหนดให้เรนเดอร์ภาพจากเฟรมที่กำหนด โดยกำหนดเฟรมแรกจนถึงเฟรมสุดท้ายที่ต้องการเรนเดอร์ได้จากช่องว่างด้านหลัง
- **Frames** เพื่อกำหนดให้เรนเดอร์เฉพาะเฟรมที่ต้องการได้โดยไม่ต้องต่อเนื่องกันตามลำดับ

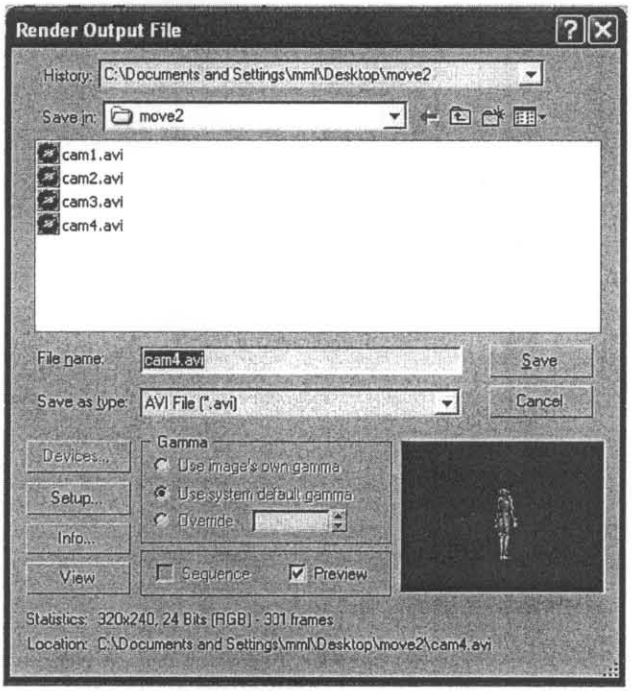
4. กำหนดขนาดของภาพที่ต้องการจากกรอบ Output Size

5. คลิกปุ่ม Files... เพื่อกำหนดรายละเอียดสำหรับบันทึกภาพเคลื่อนไหว



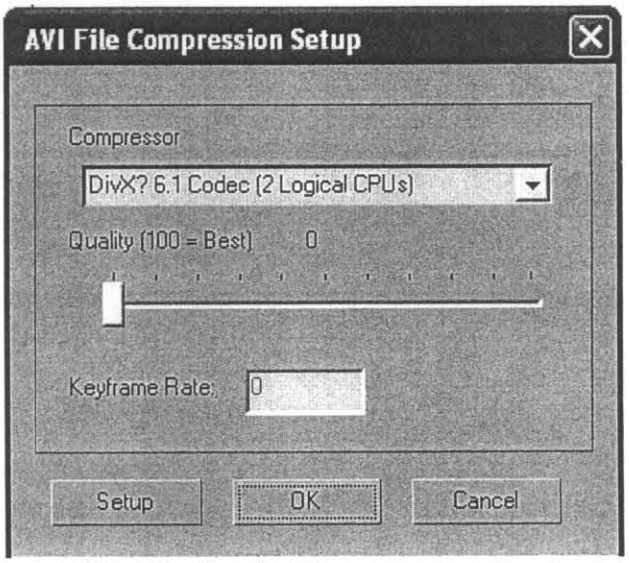
รูปที่ 3.5 ช่อง Render Output

6. กำหนดชื่อและรูปแบบไฟล์จากช่อง File name และ Save as type ในที่นี้ให้เลือกเป็นแบบ AVI ซึ่งเป็นรูปแบบไฟล์สำหรับภาพเคลื่อนไหว จากนั้นคลิกปุ่ม Save



รูปที่ 3.6 หน้าต่าง Render Output File

7. คลิกเลือกรูปแบบการบีบอัดไฟล์จากหน้าต่าง AVI File Compression Setup



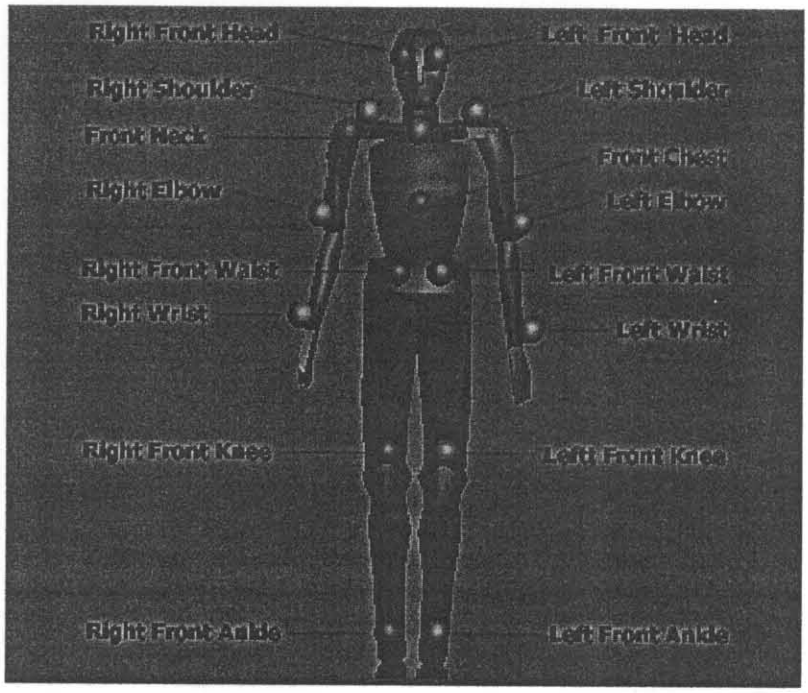
รูปที่ 3.7 หน้าต่าง AVI File Compression Setup

8. คลิกปุ่ม Render เพื่อให้เริ่มทำการบันทึกภาพเคลื่อนไหว

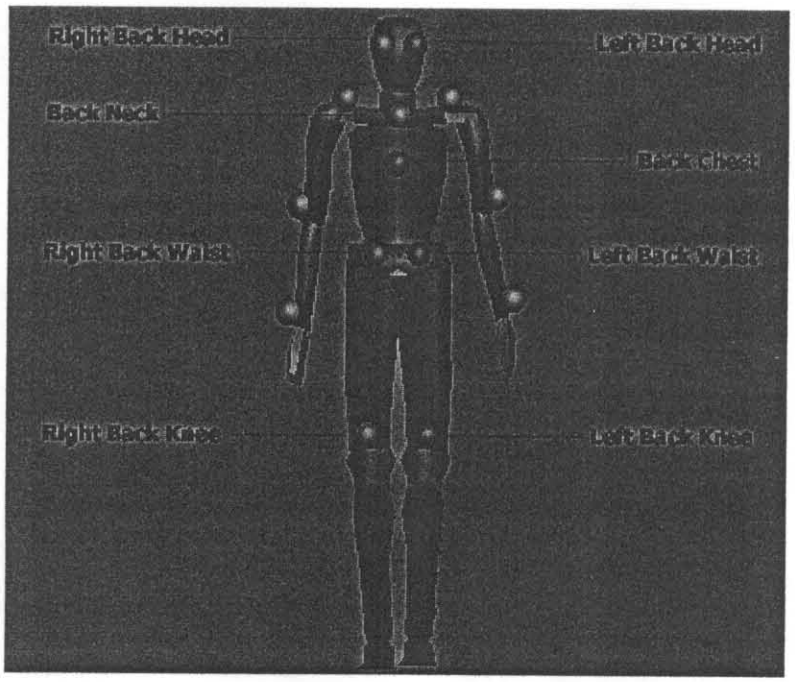
3.2.4 ผู้แสดง มาร์กเกอร์ และการจัดสภาพแวดล้อม

การเลือกผู้แสดงนั้นจะมีผลต่อการตั้งกล้อง เนื่องจากการตั้งกล้องในตำแหน่งและมุมต่างๆ จะได้บริเวณที่จำกัด การคัดเลือกผู้แสดงที่มีรูปร่างเหมาะสมจะทำให้สามารถเคลื่อนไหวในบริเวณที่จำกัดนั้นได้มากกว่าผู้ที่มีขนาดร่างกายใหญ่ แต่เนื่องจากผู้แสดงมาจากการสร้างตัวละคร 3 มิติด้วยโปรแกรม 3ds Studio Max จึงทำให้สามารถกำหนดขนาดผู้แสดงให้มีรูปร่างเหมาะสมได้ โดยในการจับการเคลื่อนไหวของผู้แสดงนั้น ระบบจะหาตำแหน่งข้อต่อสำคัญเพื่อให้สามารถนำไปสร้างการเคลื่อนไหวให้กับตัวละคร 3 มิติได้ ในการหาตำแหน่งข้อต่อของผู้แสดงจากภาพวิดีโอที่บันทึกได้นั้น จำเป็นที่จะต้องใช้สัญลักษณ์หรือมาร์กเกอร์ (Marker) เป็นจุดสังเกต เพื่อให้โปรแกรมสามารถประมวลผลหาตำแหน่งได้

ดังนั้นโครงงานนี้จึงใช้มาร์กเกอร์ที่มีลักษณะเป็นทรงกลม (Sphere) เชื่อมติดกับผู้แสดงตรงบริเวณข้อต่อในแต่ละข้อ โดยจะใช้สีที่แตกต่างกันไปในแต่ละส่วน เพื่อช่วยแยกแยะมาร์กเกอร์ในกระบวนการประมวลผลภาพ



รูปที่ 3.8 ตำแหน่งและสีของมาร์กเกอร์ที่ติดกับชุดแสดงด้านหน้า



รูปที่ 3.9 ตำแหน่งและสีของมาร์กเกอร์ที่ติดกับชุดแสดงด้านหลัง

จากรูปที่ 3.4 และ รูปที่ 3.5 เป็นรูปแสดงตำแหน่งและสีที่ใช้ของมาร์กเกอร์ในบริเวณข้อต่อต่างๆ ทั้งสิ้น 24 จุดดังนี้

1. หัวด้านซ้ายหน้า และหลัง ใช้สีชมพู

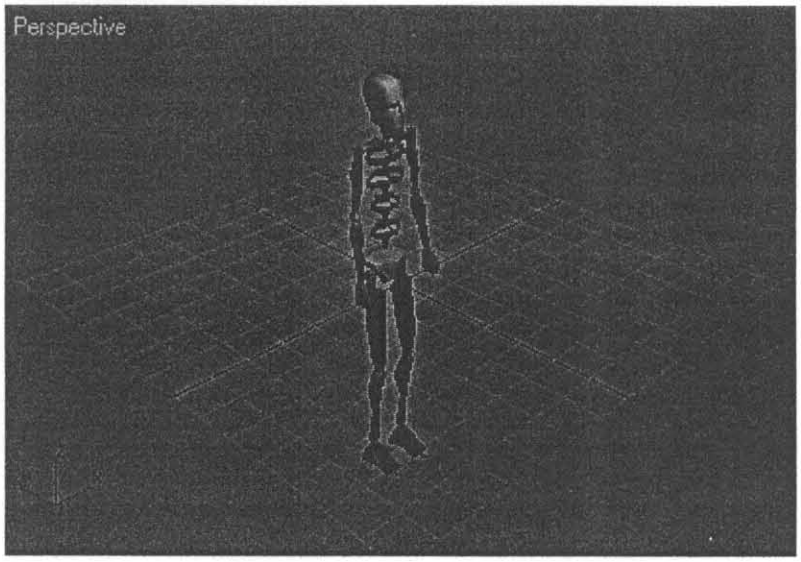
2. หัวด้านขวาหน้า และหลัง ใช้สีฟ้า
3. คอด้านหน้า และหลัง ใช้สีชมพู
4. หัวไหล่ ข้อศอก ข้อมือด้านซ้าย ใช้สีส้มเหลือง
5. หัวไหล่ ข้อศอก ข้อมือด้านขวา ใช้สีเขียว
6. อกด้านหน้า และหลัง ใช้สีฟ้า
7. สะโพก เข่า ข้อเท้าด้านหน้าซ้าย ใช้สีชมพู
8. สะโพก เข่า ข้อเท้าด้านหน้าขวา ใช้สีฟ้า
9. สะโพก เข่า ด้านหลังซ้าย ใช้สีชมพู
10. สะโพก เข่า ด้านหลังขวา ใช้สีฟ้า

สำหรับตำแหน่งที่จะตรวจจับทั้งสิ้นจำนวน 24 จุดนั้น เป็น 24 จุดหลักที่เพียงพอต่อการตรวจจับการเคลื่อนไหวของผู้แสดง โดยทั้งนี้จะเน้นไปที่การเคลื่อนไหวหัว แขน ขา และลำตัวก่อน ทำให้การตรวจจับการเคลื่อนไหวของ ข้อมือ และข้อเท้ายังไม่สามารถทำได้ดีนัก การติดมาร์กเกอร์นั้นจะติดไว้ตรงข้อต่อพอดี เพื่อเวลาตรวจจับจะได้หาตำแหน่งตรงกลางของแถบสีตรงกับตำแหน่งข้อต่อพอดี แต่อย่างไรก็ตามก็อาจมีความคลาดเคลื่อนได้จากการคลื่อนไหวร่างกายแบบต่างๆ ซึ่งถือเป็นข้อจำกัดของระบบ

นอกจากนี้สภาพแวดล้อมก็มีส่วนสำคัญต่อการตรวจจับ เพราะหากบันทึกภาพในที่ที่มีแสงมากเกินไป ก็อาจทำให้มาร์กเกอร์มีสภาพเป็นสีขาว หรือหากถ้ามืดเกินไปก็อาจจะออกเป็นสีดำ แต่เนื่องจากสภาพแวดล้อมถูกสมมติด้วยโปรแกรม 3ds Studio Max ทำให้สามารถควบคุมแสงได้ง่าย เพื่อให้สีที่ชัดเจนที่สุด ส่วนฉากหลังหรือสิ่งใดๆ ที่จะปรากฏในภาพควรเป็นสีขาวหรือดำที่ไม่รบกวนต่อการตรวจจับด้วย

3.2.5 การสร้างผู้แสดง และมาร์กเกอร์

ในการสร้างผู้แสดง สำหรับโครงงานนี้จะใช้ Plug-in ย่อยของCharacter Studio 2 ส่วน คือ Biped โดย Biped เป็นชุดโครงกระดูกของมนุษย์สำเร็จรูปที่เอาไว้ใช้งานแทนกระดูก (Bone) แบบเดิมที่ใช้งานในโปรแกรม 3ds Studio Max ซึ่งทำให้ช่วยลดขั้นตอนในการสร้างกระดูก และการเชื่อมต่อกระดูกให้เป็นรูปร่างตัวละคร การทำงานที่เกี่ยวข้องกับการ Setup IK รวมทั้งการตั้งชื่อให้กับกระดูกแต่ละชิ้น ซึ่งเป็นขั้นตอนที่เสียเวลามาก โดยเฉพาะงานที่ต้องมีตัวละครมากๆ



รูปที่ 3.10 ตัวอย่างของโครงกระดูกสำเร็จรูป (Biped)

หลังจากที่ได้ลงโปรแกรม 3D Studio Max และ Plug-in Character Studio แล้ว เมื่อเปิดโปรแกรมขึ้นมาเพื่อใช้งานจะสามารถสร้างหุ่นที่เป็นโครงกระดูกสำเร็จรูปได้ โดยเลือกที่แถบเครื่องมือใน Tab Create ดังรูปที่ 3.11 แล้วลากเพื่อสร้างโครงกระดูกสำเร็จรูป (Biped) บนพื้นที่ทำงาน

รูปที่ 3.11 เครื่องมือที่ใช้ในการสร้างโครงกระดูกสำเร็จรูป (Biped)

ในส่วนของการสร้างมาร์กเกอร์นั้น เมื่อเปิดโปรแกรมขึ้นมาเพื่อใช้งาน จะสามารถสร้างวัตถุทรงกลมได้ โดยเลือกที่แถบเครื่องมือใน Tab Create ดังรูปที่ 3.12 แล้วลากเพื่อสร้างวัตถุทรงกลม (Sphere) บนพื้นที่ทำงาน

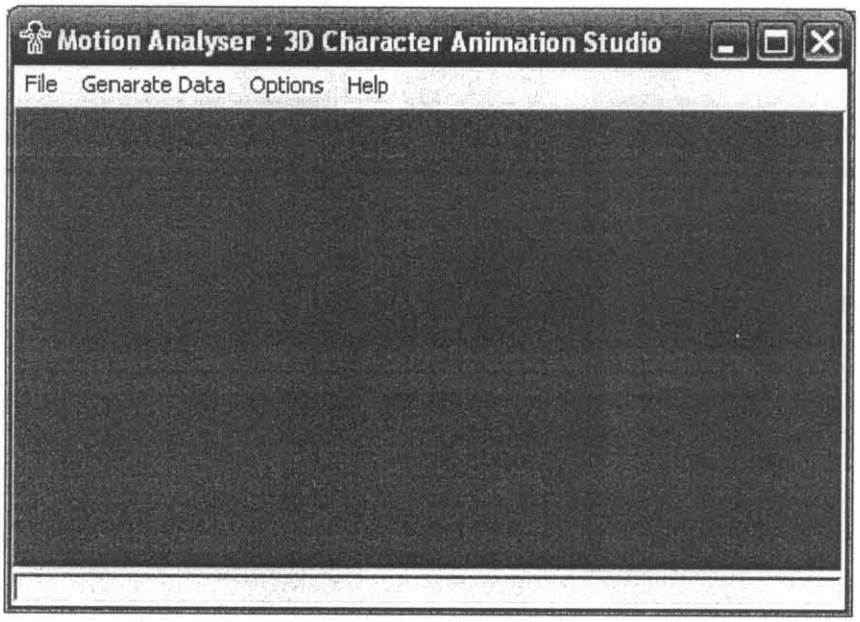
รูปที่ 3.12 เครื่องมือที่ใช้ในการสร้างวัตถุทรงกลม (Sphere)

ในส่วนของการติดมาร์กเกอร์ที่ตัวผู้แสดง หลังจากที่สร้างวัตถุทรงกลมแล้ว เราจะนำมาติดเข้ากับร่างกายของผู้แสดง โดยการใช้คำสั่ง Select and Link () กดที่ object ลูก (มาร์กเกอร์) แล้วไปเชื่อมที่ object แม่ (ชิ้นส่วนต่างๆของร่างกาย เช่น แขน) ซึ่งข้อดีของคำสั่งนี้คือ เวลาเราขยับมาร์กเกอร์ จะไม่มีผลต่อแขน แต่ถ้าเราขยับแขน marker จะตามมาด้วย ทำให้เราสามารถปรับแก้ตำแหน่งของมาร์กเกอร์ได้ง่าย

3.3 การวิเคราะห์การเคลื่อนไหว (Motion Analysis)

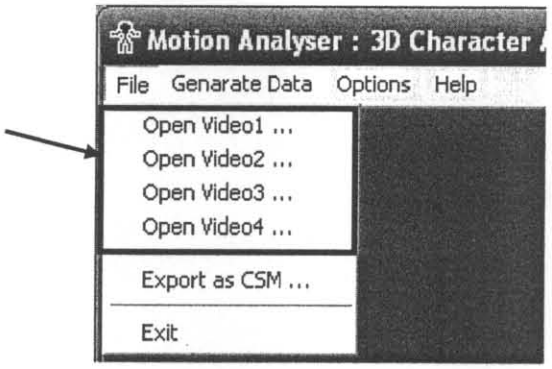
3.3.1 การเปิดไฟล์วีดีโอ

ในส่วนของการวิเคราะห์การเคลื่อนไหวนั้น จะเป็นการใช้งานของโปรแกรมที่สร้างขึ้น ในการทำการแปลงภาพวีดีโอ จากกล้องทั้ง 4 ตัว ไปเป็นข้อมูลของตำแหน่งต่างๆ ของมาร์กเกอร์ โดยจะมีวิธีในการทำงานดังนี้



รูปที่ 3.13 แสดงโปรแกรม Motion Analysis

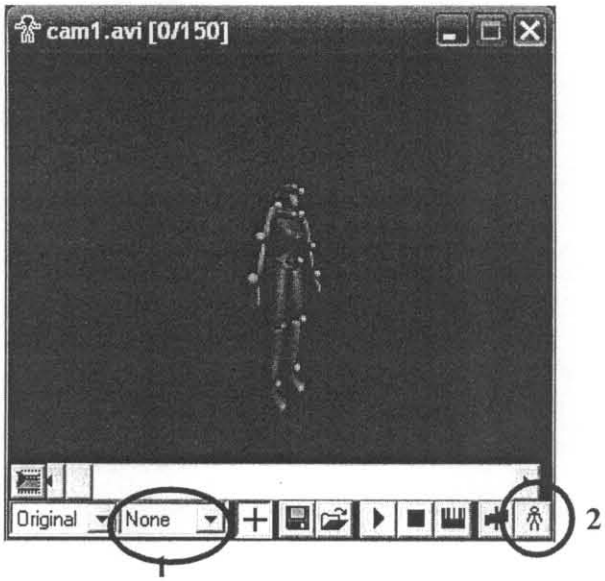
การเปิดไฟล์วีดีโอ เป็นการเปิดไฟล์วีดีโอ ที่มาจากกล้องวีดีโอ โดยจะมีทั้งหมด 4 กล้อง ด้วยกัน โดยใช้งานโดยกดเปิดเมนูวีดีโอที่ต้องการจาก เมนู File ที่แถบ menu bar ดังรูป



รูปที่ 3.14 แสดงการเปิดไฟล์วีดีโอ จากกล้อง

3.3.2 การตั้งตำแหน่งจุด Blob เริ่มต้น

เป็นการกำหนดจุด Blob ของกล้องวีดีโอ แต่ละกล้องที่เราได้ทำการเปิดไฟล์ของวีดีโอนั้นขึ้นมาก่อน

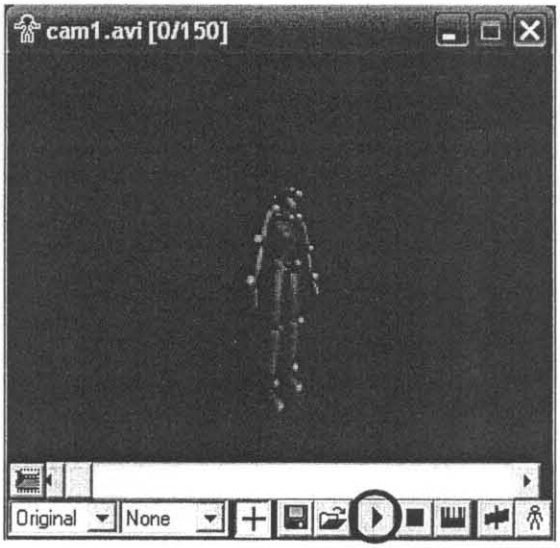


รูปที่ 3.15 แสดงปุ่มที่ใช้ตั้งค่าจุด Blob เริ่มต้นของโปรแกรมวีดีโอของ Camera 1

จากรูปเราสามารถค่อยๆ ระบุตำแหน่งของจุด Blob ทีละจุด โดยการใช้แถบในส่วนที่ 1 หรือจะใช้ตำแหน่งที่ 2 ในการระบุตำแหน่งที่ทำไว้แล้วในท่าพื้นฐาน คือ ยืนตัวตรงโดยไม่กางแขน ได้

3.3.3 การวิเคราะห์ตำแหน่งจุด Blob

เป็นการทำการวิเคราะห์หาตำแหน่งของจุด Blob ในแต่ละกล้องวีดีโอ ทำได้โดยการกดปุ่ม Start ของแต่ละกล้องวีดีโอ

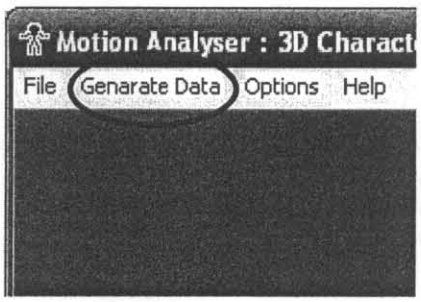


รูปที่ 3.16 แสดงปุ่ม Start ที่โปรแกรมวีดีโอของ Camera 1

ต้องทำการวิเคราะห์ตำแหน่งของจุด Blob ในทุกๆ กล้องก่อนที่เราจะทำงานต่อๆ ไป โดยผลที่ได้จะได้ข้อมูลจุด Blob ทั้งหมด ในทุกๆ เฟรมเพื่อนำไปหาตำแหน่งมาร์กเกอร์ต่อไป

3.3.4 การวิเคราะห์หาตำแหน่งของมาร์กเกอร์

เป็นการวิเคราะห์หาตำแหน่งจริงของจุดมาร์กเกอร์ จากการนำเอาข้อมูลของจุด Blob ของกล้องทั้ง 4 ตัว นำมาวิเคราะห์ วิธีในการใช้งานคือใช้เมนู "Genetate Data" ในแถบ menu bar

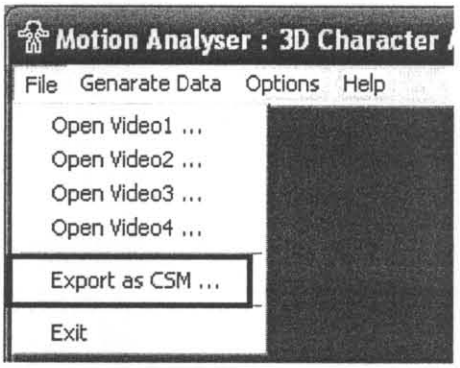


รูปที่ 3.17 แสดงปุ่มที่ใช้วิเคราะห์ตำแหน่งมาร์กเกอร์

เราต้องทำการกำหนดตำแหน่ง Blob ของข้อมูลกล้อง ให้ครบทุก 4 กล้องก่อนจึงจะสามารถทำงานได้ โดยผลลัพธ์ที่ได้ จะเป็นข้อมูลของมาร์กเกอร์ทั้งหมดในแต่ละเฟรม

3.3.5 การสร้างไฟล์การเคลื่อนไหว

เป็นการนำข้อมูลตำแหน่งมาร์กเกอร์ทั้งหมด มาทำการเขียนใหม่ ให้เป็นไฟล์แบบ CSM เพื่อนำไปใส่ให้ตัวละคร 3 มิติ ที่เราต้องการแสดง โดยการกด "Export as CSM" ที่เมนู File ที่แถบ menu bar



รูปที่ 3.18 แสดงปุ่มที่ใช้สร้างไฟล์การเคลื่อนไหว

เราต้องทำการหาตำแหน่งของมาร์กเกอร์ก่อน จึงจะสามารถทำงานได้ โดยเราจะได้ไฟล์ออกมาเป็นไฟล์การเคลื่อนไหวแบบ CSM ออกมา

3.4 การสร้างการเคลื่อนไหวให้กับตัวละคร (Animation Production)

สำหรับการสร้างการเคลื่อนไหวให้กับตัวละคร (Character Animation) โดยการใช้ไฟล์ข้อมูลการเคลื่อนไหวช่วยในการสร้างการเคลื่อนไหวนั้นแบ่งออกเป็นขั้นตอนหลัก 4 ขั้นตอน ดังนี้

1. สร้างตัวละครที่มีลักษณะตามต้องการ
2. สร้างโครงกระดูกสำเร็จรูป(Biped) ที่มีรูปร่าง และข้อต่อสัมพันธ์กับโมเดลที่สร้างไว้
3. นำเข้าไฟล์ข้อมูลการเคลื่อนไหวแบบ CSM ให้กับโครงกระดูกสำเร็จรูปที่สร้างไว้
4. ทำการยึดโครงกระดูกสำเร็จรูปเข้ากับโมเดลตัวละครที่ได้สร้างไว้

ขั้นตอนเหล่านี้เป็นหลักการของโปรแกรมออกแบบโมเดล 3 มิติทั่วไป สำหรับในการศึกษาและพัฒนาระบบได้ใช้โปรแกรม 3D Studio Max และโปรแกรม Character Studio ซึ่งเป็น Plug-in ของโปรแกรมเป็นหลักซึ่งสามารถอธิบายขั้นตอนการสร้างการเคลื่อนไหวให้กับตัวละครด้วยไฟล์ข้อมูลการเคลื่อนไหวแบบ CSM ที่ได้จากการวิเคราะห์หาตำแหน่งของจุดมาร์กเกอร์ในกระบวนการก่อนหน้านี้ได้ ดังนี้

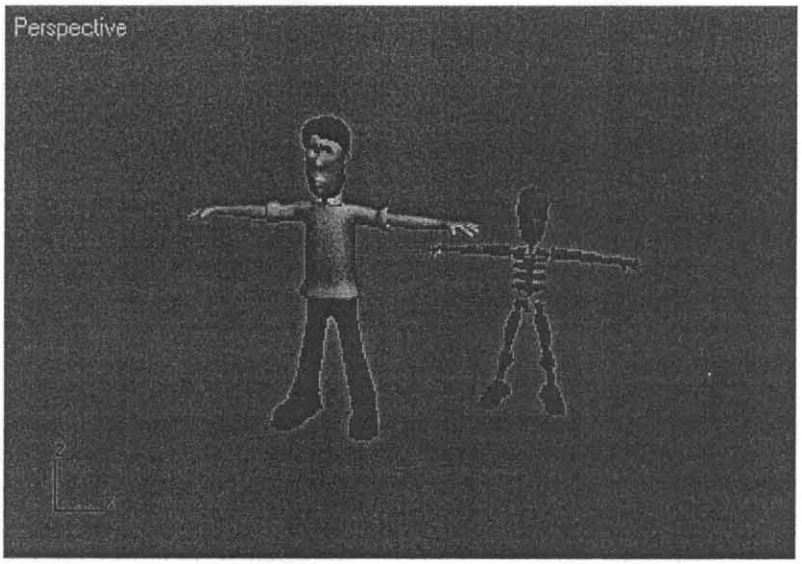
3.4.1 การสร้างโครงกระดูกสำเร็จรูป

การสร้างตัวละครและโครงกระดูกสำเร็จรูปนั้น ได้อธิบายไปแล้วในก่อนหน้านี้

3.4.2 การนำไฟล์ข้อมูลการเคลื่อนไหวแบบ CSM มาใช้

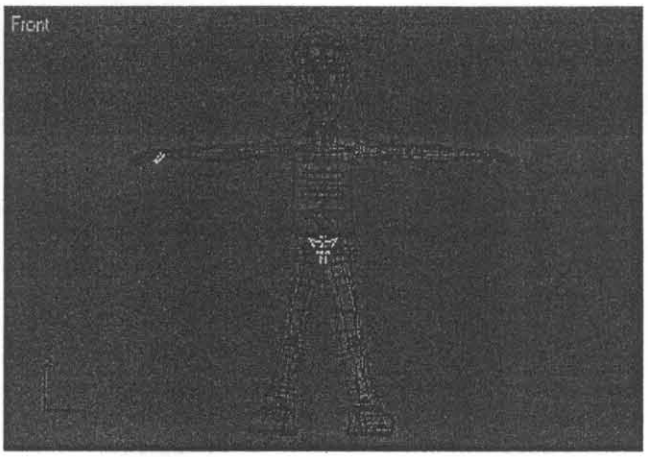
เป็นขั้นตอนสุดท้ายของการสร้างการเคลื่อนไหวให้กับโมเดลตัวละครที่ได้ออกแบบ และทำไว้แล้วด้วยการนำเข้าไฟล์ข้อมูลบันทึกการเคลื่อนไหว (Motion Data File) ให้กับโครงกระดูกสำเร็จรูป (Biped) แล้วนำโครงกระดูกสำเร็จรูปไปยึดเข้ากับโมเดลตัวละครที่ได้สร้างไว้อีกทีหนึ่ง โดยการใช้เครื่องมือ Freeze Selection ใน Popup Menu กับโมเดลตัวละคร หลังจากนั้นทำการจัดรูปร่างท่าทาง ขนาด และตำแหน่งของข้อต่อส่วนต่างๆ ของโครงกระดูกสำเร็จรูป (Biped) ให้เข้ากับโมเดลตัวละคร 3 มิติ เมื่อจัดรูปร่างของโครงกระดูกสำเร็จรูปได้เหมาะสมแล้ว ให้ทำการ Unfreeze โมเดลตัวละครด้วยเครื่องมือ Unfreeze all ใน Popup Menu และสุดท้ายคือการทำ Attach to Node ในแถบเครื่องมือ Modify ในส่วนของ Physique ซึ่งมีขั้นตอนดังต่อไปนี้

1. สร้างตัวละคร 3 มิติ และโครงกระดูกสำเร็จรูปขึ้นมา



รูปที่ 3.19 โมเดลตัวละคร และโครงกระดูกสำเร็จรูปที่ทำการนำเข้าไฟล์การเคลื่อนไหวไว้

2. นำโครงกระดูกสำเร็จรูปที่นำเข้าข้อมูลการเคลื่อนไหวแล้ว จัดตำแหน่งและขนาดให้พอดีกับตัวละคร 3 มิติ ดังรูปที่ 3.19 และยึดเข้ากับตัวละคร 3 มิติตามขั้นตอนในรูปที่ 3.20



รูปที่ 3.20 โครงกระดูกสำเร็จรูปที่จัดรูปร่างให้เหมาะกับโมเดลตัวละครแล้ว

บทที่ 4

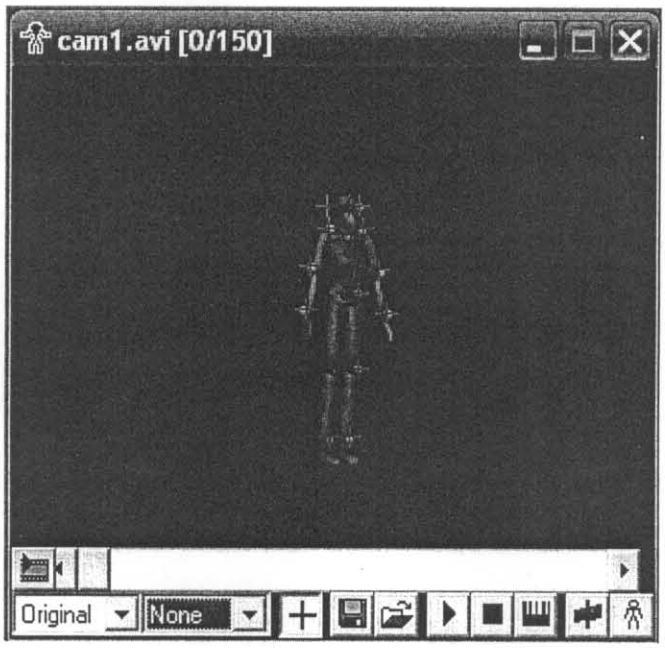
ผลการทดสอบระบบ

ในบทนี้จะเป็นนำเสนอผลการทดสอบระบบในการตรวจจับจุด Blob จากภาพวิดีโอไปจนถึงการนำไฟล์ข้อมูลการเคลื่อนไหว CSM มาสร้างการเคลื่อนไหวให้กับตัวละคร พร้อมสรุปผลในแต่ละหัวข้อ โดยการทดสอบในส่วนนี้ทั้งหมด ได้ทำการบันทึกการเคลื่อนไหวโดยการ simulate ด้วย โปรแกรม 3D Studio Max ออกมาเป็นไฟล์ AVI เพื่อนำมาใช้ประมวลผลต่อไป

4.1 การตรวจจับ ติดตาม และแยกจุด Blob

การทดสอบนี้เป็นการทดสอบการตรวจจับเพื่อหาตำแหน่งของจุด Blob จากภาพวิดีโอของกล้องทั้ง 4 กล้อง ว่าระบบสามารถทำงานได้ตามที่ออกแบบไว้หรือไม่และได้ผลเป็นเช่นไร โดยการบันทึกภาพวิดีโอการเคลื่อนไหวของตัวละครในลักษณะท่าทางแบบต่างๆ จากนั้นนำมาให้โปรแกรมทำการวิเคราะห์การเคลื่อนไหว โดยทำการตรวจจับและติดตามจุด Blob ผลการทดสอบจะรวมถึงการประมวลผลภาพ การตรวจจับตำแหน่งจุด Blob และการติดตามจุด Blob

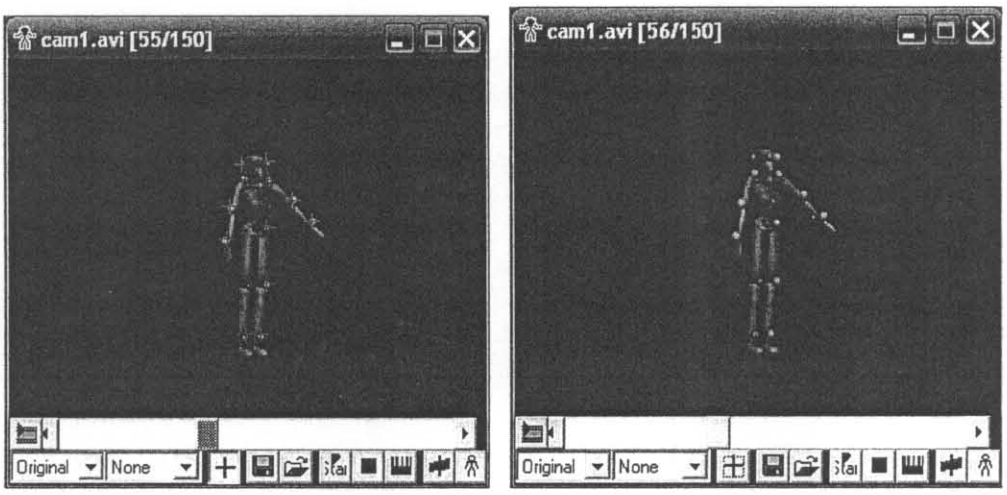
4.1.1 การตรวจจับตำแหน่งจุด Blob ในท่าเริ่มต้น



รูปที่ 4.1 การตรวจจับตำแหน่งจุด Blob ในท่าเริ่ม

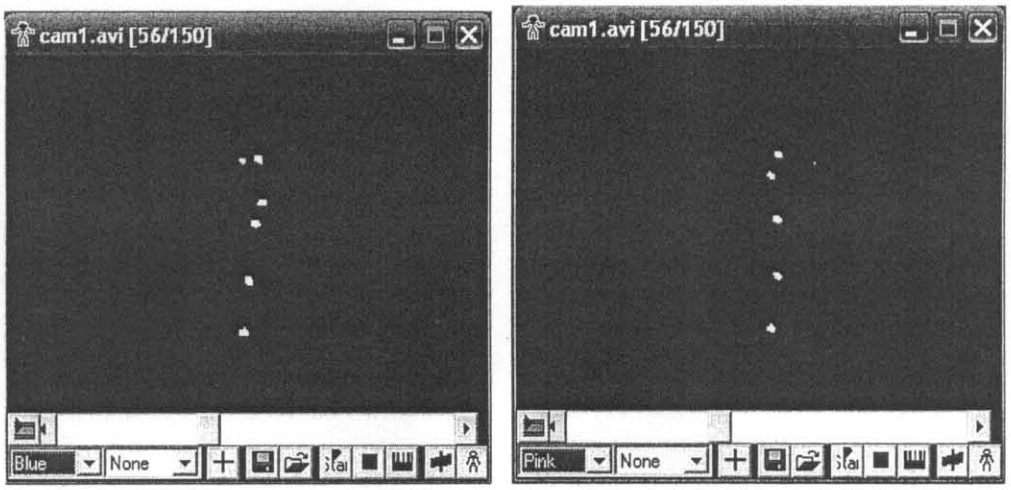
ในเฟรมแรกผู้ใช้จะต้องทำการกำหนดตำแหน่งของจุด Blob ทุกจุด เพื่อใช้เป็นตำแหน่งเริ่มต้นในการติดตาม แต่เพื่อความสะดวก โปรแกรมสามารถที่จะตรวจจับตำแหน่งให้เองได้ แต่จะต้องอยู่ในท่าทางที่กำหนดคือยืนตรงแขนแนบชิดลำตัวดังรูปที่ 4.1 ซึ่งโปรแกรมสามารถตรวจจับได้อย่างถูกต้องแม่นยำ

4.1.2 การประมวลผลภาพเพื่อหาตำแหน่ง

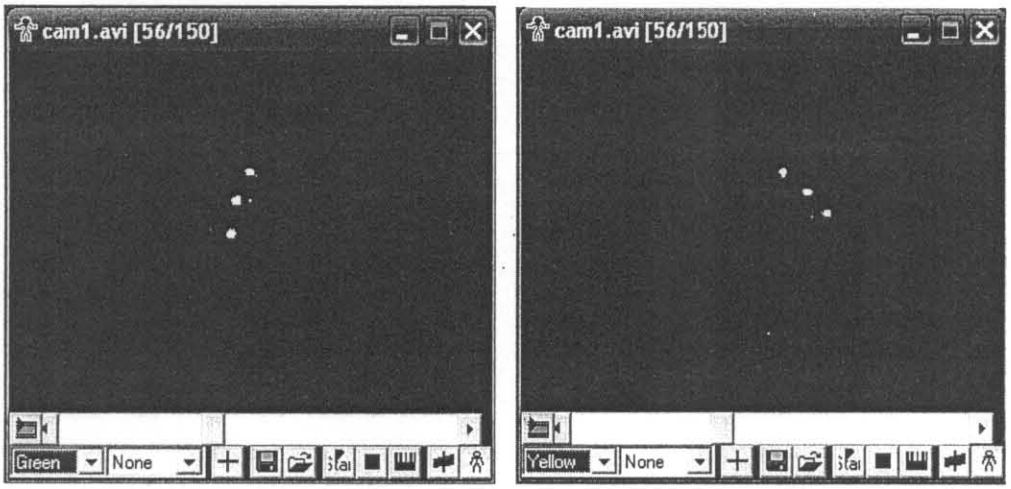


(ก) (ข)

รูปที่ 4.2 (ก) แสดงตำแหน่งจุด Blob ที่ตรวจจับได้ในเฟรมที่ 55
(ข) แสดงภาพที่ต้องตรวจจับในเฟรมที่ 56

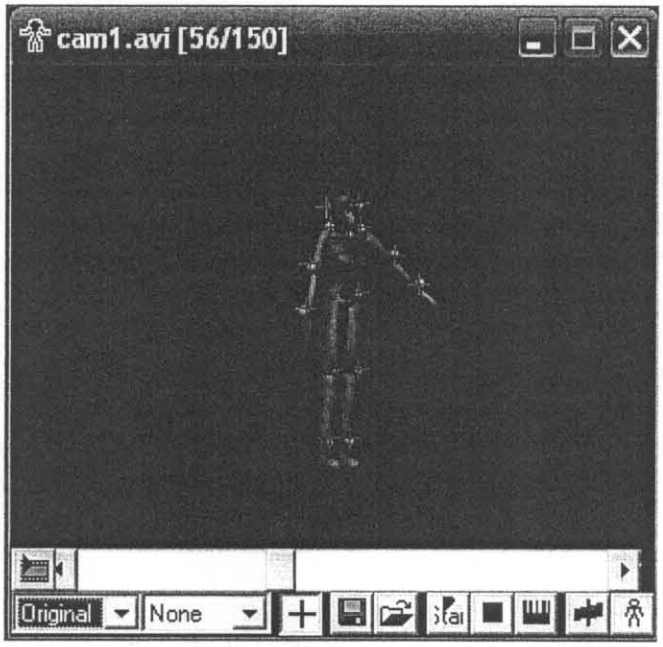


(ก) (ข)



(ค) (ง)

รูปที่ 4.3 (ก) ภาพที่ได้หลังการทำ Threshold ในช่วงสีน้ำเงิน
(ข) ภาพที่ได้หลังการทำ Threshold ในช่วงสีชมพู
(ค) ภาพที่ได้หลังการทำ Threshold ในช่วงสีเขียว
(ง) ภาพที่ได้หลังการทำ Threshold ในช่วงสีเหลือง

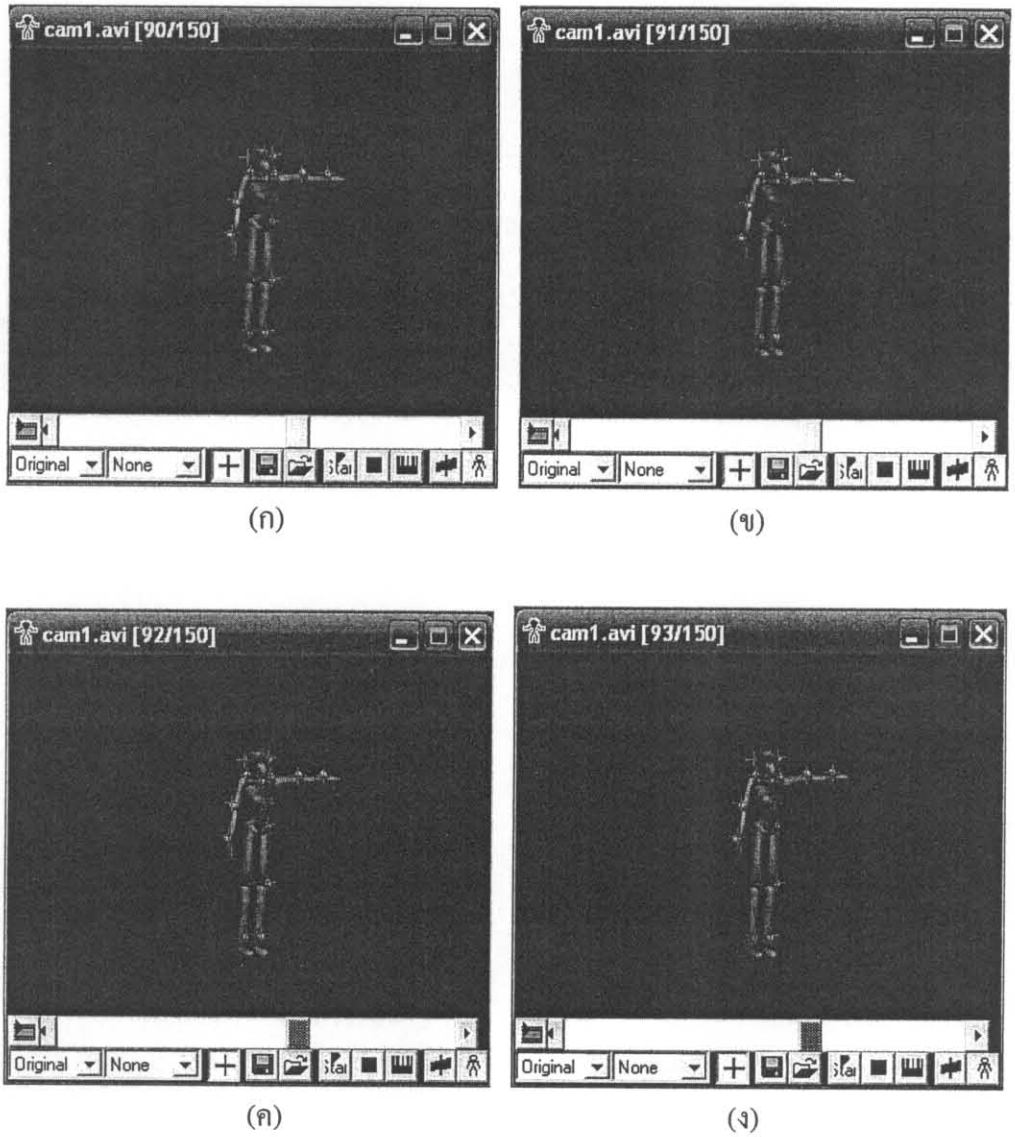


รูปที่ 4.4 แสดงตำแหน่งจุด Blob ที่ตรวจจับได้ในเฟรมที่ 56

จากรูปที่ 4.2 (ก) เป็นตำแหน่งที่โปรแกรมสามารถตรวจจับได้ในเฟรมก่อนหน้า และในรูปที่ 4.2 (ข) เป็นภาพในเฟรมถัดมาที่ต้องการทำตรวจจับ โดยในรูปที่ 4.3 แสดงภาพขาว-ดำที่ได้จาก

การทำ Threshold ในช่วงสีต่างๆ ซึ่งพบว่าโปรแกรมสามารถตัดช่วงสีที่ต้องการออกมาได้อย่างถูกต้อง ถึงแม้ว่าจะมีจุดรบกวนบ้าง แต่ก็มีปริมาณที่น้อย ซึ่งโปรแกรมสามารถแยกแยะได้อย่างถูกต้อง ดังในรูปที่ 4.4

4.1.3 การติดตามจุด Blob



(ก) (ข)

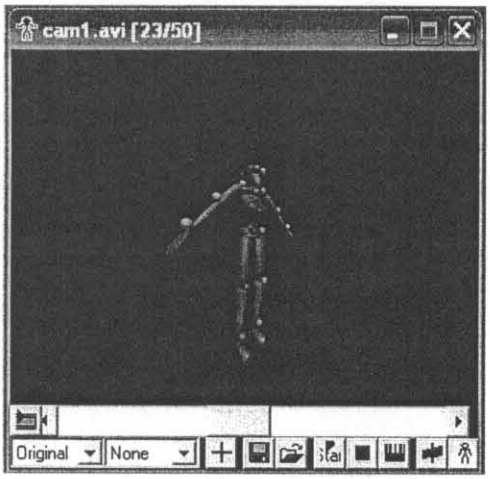
(ค) (ง)

รูปที่ 4.5 (ก) แสดงตำแหน่งจุด Blob ที่ตรวจจับได้ในเฟรมที่ 90
(ข) แสดงตำแหน่งจุด Blob ที่ตรวจจับได้ในเฟรมที่ 91
(ค) แสดงตำแหน่งจุด Blob ที่ตรวจจับได้ในเฟรมที่ 92
(ง) แสดงตำแหน่งจุด Blob ที่ตรวจจับได้ในเฟรมที่ 93

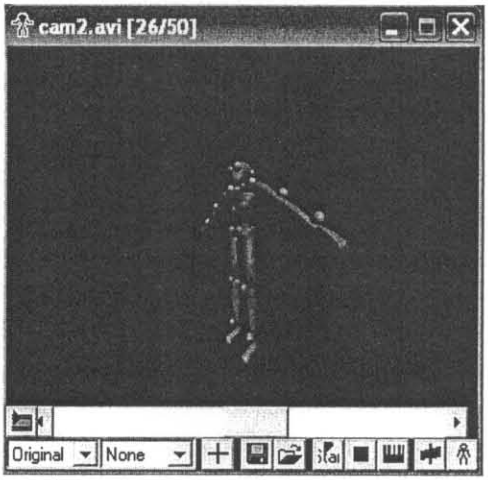
จากการให้โปรแกรมทำการตรวจจับอย่างต่อเนื่อง พบว่าสามารถตรวจจับจุด Blob แต่ละจุดได้อย่างถูกต้อง และสามารถติดตามได้อย่างต่อเนื่อง แต่ก็จะมีบ้างบางครั้งหากจุด Blob ถูกบดบังหรือจุด Blob ที่มีสีเดียวกันอยู่ติดกัน (ซ้อนทับกัน) ก็จะทำให้การตรวจจับมีปัญหาได้

4.2 การทดลองการทำงานโดยไม่มีการบดบัง

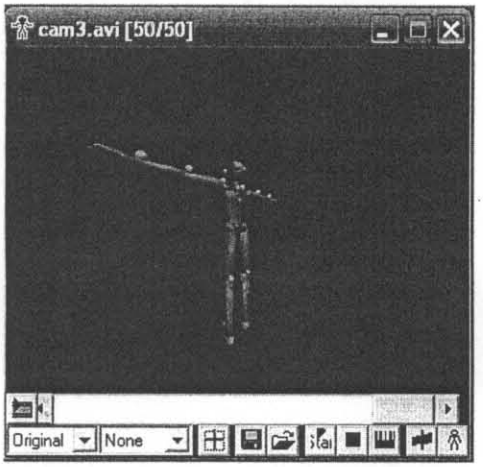
จากการทดลอง ในการตรวจจับการเคลื่อนไหวนั้น หากจุด Blob ไม่เกิดการบดบังเลย เราสามารถตรวจจับการเคลื่อนไหวได้อย่างถูกต้อง ตามตัวอย่างต่อไปนี้



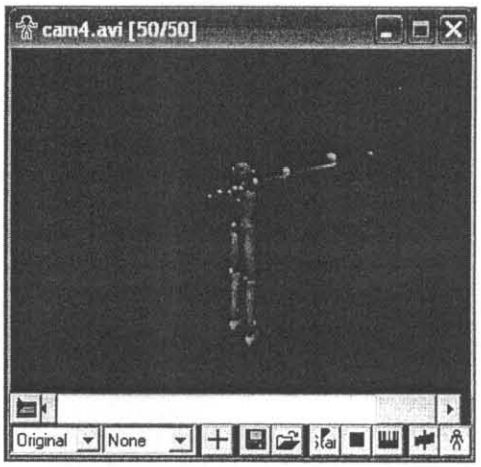
(ก)



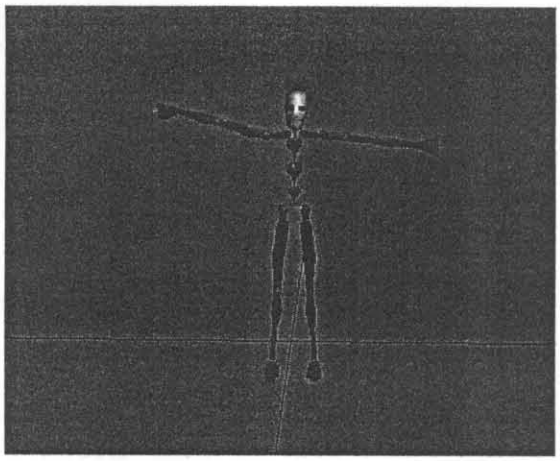
(ข)



(ค)



(ง)



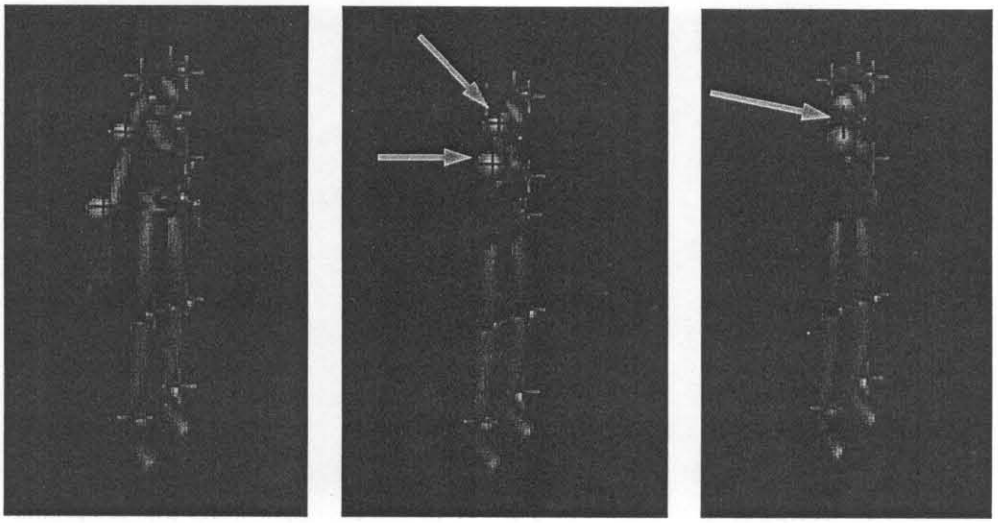
(จ)

รูปที่ 4.6 (ก) ภาพจากกล้อง 1 แสดงท่าการยกแขนออกทั้ง 2 ข้าง
(ข) ภาพจากกล้อง 2 แสดงท่าการยกแขนออกทั้ง 2 ข้าง
(ค) ภาพจากกล้อง 3 แสดงท่าการยกแขนออกทั้ง 2 ข้าง
(ง) ภาพจากกล้อง 4 แสดงท่าการยกแขนออกทั้ง 2 ข้าง
(จ) ภาพท่าทางของโครงกระดูกสำเร็จรูปที่ได้จากการนำไฟล์ข้อมูลไปใช้งาน

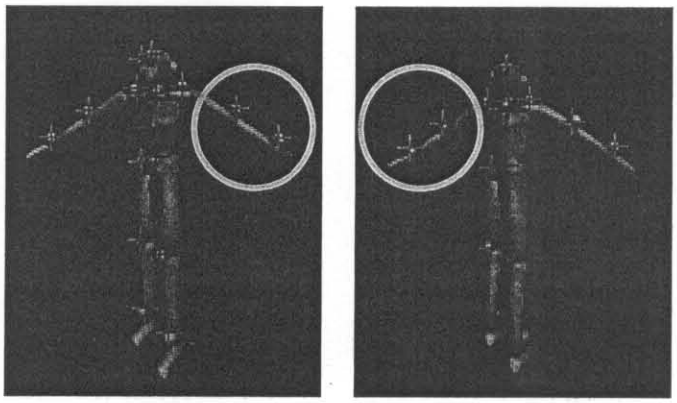
4.3 การทดลองแก้ปัญหาในการวิเคราะห์ในการหาตำแหน่งจริง

ในการหาตำแหน่งจริงนั้น ข้อมูลจุด Blob ที่เราติดตาม หากมีข้อมูลมากๆ จะทำให้เราสามารถนำมาวิเคราะห์หาตำแหน่งของมาร์กเกอร์ได้ดียิ่งขึ้น แต่การที่จุด Blob นั้นมีน้อยกว่าที่ควรจะเห็นนั้น อาจเกิดมาจากสาเหตุต่างๆ เช่น ซ้อนทับกัน, การที่จุด Blob นั้นมาจากมุมกล้องอื่น หรือจุด Blob ถูกบดบังเป็นเวลานานหลายเฟรมเกินไป อย่างไรก็ตามเราจึงทำการทดลอง โดยการให้ตัวละครแสดงท่าที่เราต้องการ และทำการตรวจจับดู ดังต่อไปนี้

4.3.1 การติดตามจุด Blob ที่ทับซ้อนกัน



(ก) (ข) (ค)



(ง) (จ)

รูปที่ 4.7 (ก) แสดงท่าแกว่งแขนไปตามปกติ

(ข) การแสดงท่าแกว่งแขนในเฟรมที่ 28 ก่อนเกิดการ ซ้อนทับ

(ค) แสดงท่าแกว่งแขนในเฟรมที่ 29 เกิดการซ้อนทับ

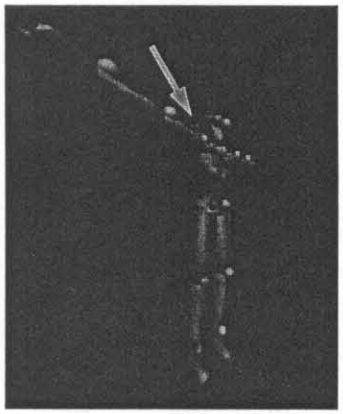
(ง) ภาพจากกล้องที่ 2ที่มองเห็นจุด Blob ในเฟรมที่ 29ในขณะที่ กล้อง 1 เกิดการซ้อนทับ

(จ) ภาพจากกล้องที่ 4 ที่มองเห็นจุด Blob ในเฟรมที่ 29 ในขณะที่กล้อง 1 เกิดการซ้อนทับ

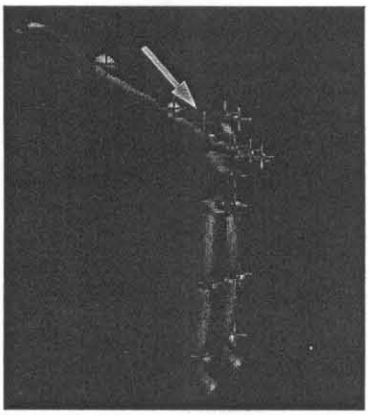
ภาพ 4.7(ก) ถึงภาพที่ 4.7(ค) เป็นภาพจากกล้องที่ 1 โดยจากภาพที่ 4.7(ข) ภาพการแกว่งแขนไปนั้น จะเหตุการณ์ที่จุด Blob จะทำการทับซ้อนกัน โดยในที่นี้ จุด Blob ของข้อศอกขวาจะไปซ้อนทับกับจุด Blob ของข้อมือขวา และเมื่อเกิดการซ้อนทับกันในรูป 4.7(ค) แล้วตำแหน่งจุด Blob ของข้อศอกจะถูกซ้อนทับโดยตำแหน่งข้อมือ ซึ่งเราได้ทำการตรวจสอบได้ว่ามีการเกิดการทับซ้อน

กันได้ ตามวิธีการตรวจจับในบทก่อนหน้าแล้ว ทำให้สามารถเก็บข้อมูลต่อไปได้ โดยการดูจากคู่กล้องข้างเคียง

4.3.2 การติดตามจุด Blob ที่เริ่มจะมองเห็น



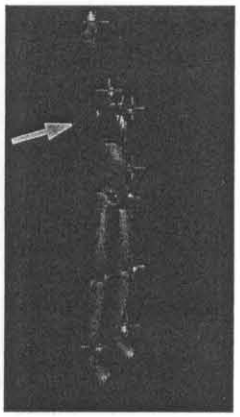
(ก)



(ข)

(ค)

(ง)



(จ)

รูปที่ 4.8 (ก) แสดงท่ากางแขนหมุนตัวตามปกติ

(ข) แสดงท่ากางแขนหมุนตัวและดูการติดตามจุด Blob ที่หัวด้านหลังขวา

(ค) แสดงท่ากางแขนหมุนตัวและเกิดการซ้อนทับที่ตำแหน่งหัวด้านหลังขวา

(ง) แสดงตำแหน่งของท่ากางแขนหมุนตัว และเกิดการซ้อนทับที่ตำแหน่งหัวด้านหลังขวา

(จ) แสดงตำแหน่งของท่ากางแขนหมุนตัว โดยจุด Blob ที่หัวด้านหลังขวาโผล่กลับมา

รูปที่ 4.8 เป็นภาพจากกล้องที่ 1 โดยเป็นการแสดงให้เห็นการเกิดเหตุการณ์การบดบัง และเมื่อโปรแกรมตรวจพบจุด Blob ที่ถูกบดบังนั้นแล้ว โปรแกรมจะทำการระบุตำแหน่งให้ได้ ทำให้การทำงานในเฟรมต่อๆ ไป จะสามารถระบุจุด Blob นั้นต่อไปได้ (การติดตามจุด Blob ต้องใช้ตำแหน่งของจุด Blob นั้นในเฟรมก่อนหน้ามาช่วยด้วย)

ในการทดลองต่างๆ จะเห็นว่าเราสามารถทำการติดตาม และระบุตำแหน่งต่างๆ ของจุด Blob ได้ ทำให้การทำงานของระบบเป็นไปได้ โดยไม่ต้องมีการหยุดการทำงาน และเมื่อเราระบุตำแหน่งของ Blob ได้ทุกอันแล้ว การหาตำแหน่งจริงของมาร์กเกอร์นั้นก็ไม่เป็นปัญหาอีก ทำให้เราได้ข้อมูลที่สมบูรณ์ขึ้นเพื่อนำไปใช้ในการสร้างไฟล์ CSM ต่อไป

บทที่ 5
บทวิจารณ์และสรุป

5.1.ข้อสรุป

สำหรับโครงงานระบบจำลองการเคลื่อนไหวตัวละคร 3 มิติ (3D Character Animation) นี้นั้น ในส่วนของการออกแบบของระบบจะเน้นในส่วนของการปรับปรุงเพื่อเพิ่มประสิทธิภาพให้ระบบเดิม โดยได้ทำการเพิ่มกล้องเพื่อลดปัญหาการหายไปของจุด Blob ซึ่งเกิดจากการที่จุดมาร์กเกอร์ถูกบดบังโดย จำนวนกล้องที่ใช้ในโครงงานนี้คือ 4 ตัว นำมาติดตั้งในมุมมองรอบตัวผู้แสดง นอกจากนี้ยังออกแบบในส่วนของการประมวลผลภาพเพื่อหาตำแหน่ง ปรับปรุงให้ระบบสามารถทำงานได้อย่างต่อเนื่องแม้ว่าจะมีมาร์กเกอร์ใดถูกบดบังไปเป็นเวลานานๆ และสามารถทำงานได้เองโดยอัตโนมัติ โดยไม่ต้องทำการกำหนดตำแหน่งจุดเองในขณะประมวลผลข้อมูล

ในส่วนของการบันทึกภาพการเคลื่อนไหว ที่ได้สร้างขึ้นมาจากการ simulation ด้วยโปรแกรม 3D Max Studio แทนการใช้กล้องวิดีโอถ่ายภาพการเคลื่อนไหวของคนจริงๆ นั้น มีข้อดี ข้อเสียต่างๆ ดังนี้

ข้อดี

- ทำให้ไม่ต้องสนใจสภาพแวดล้อม เพราะการที่จำลองการเคลื่อนไหวขึ้นมานั้น ไม่มีสิ่งรบกวนจากภายนอก ทำให้การทำทดลองง่ายขึ้น
- ง่ายต่อการทำการเพิ่มจำนวนกล้อง และการวางกล้อง เพราะได้ตำแหน่งที่ถูกต้อง และแม่นยำมาก
- อัตราการให้การที่กล้องทำการถ่ายภาพ สามารถทำงานได้พร้อมกันทุกตัว
- สามารถกำหนดขนาดต่างๆ ของตัวละครได้ตามความต้องการ

ข้อเสีย

- การเคลื่อนไหวของตัวละครไม่ค่อยสมจริง เพราะเป็นการสร้างขึ้นมาเอง
- ในการทดลอง ทำได้เพียงทดสอบระบบ และการทำงานที่ได้ออกแบบไว้เท่านั้น แต่ในการนำมาใช้จริงอาจเกิดปัญหาต่างๆ ได้
- การวางมาร์กเกอร์ต่างๆ ไม่สมจริงเพราะมาร์กเกอร์ที่สร้างขึ้นมา ลักษณะของมัน ไม่เหมือนกับการติดมาร์กเกอร์จริงๆ ได้

5.2 แนวทางการพัฒนาต่อ

5.2.1 การหาตำแหน่งข้อต่อจริงจากภาพ

โครงงานนี้ใช้ทฤษฎีในการหาตำแหน่งจริงของภาพโดยใช้ Stereopsis นั้น เป็นการหาตำแหน่งของมาร์กเกอร์โดยใช้กล้อง 2 กล้อง โดยจะต้องอยู่ในระนาบเดียวกันเท่านั้น ทำให้ต้องใช้พื้นที่ในการทดลองมาก หากใช้ทฤษฎีที่ไม่จำเป็นต้องวางเป็นระนาบเดียวกันได้ น่าจะทำให้การพัฒนาเป็นไปได้ในทางที่ดีขึ้นอีก ในหลายมุมมองของกล้อง จะทำให้ได้ภาพที่ดียิ่งขึ้น

5.2.2 การสร้างการเคลื่อนไหวโดยใช้โครงกระดูก

ในการตรวจจับในโครงงานนั้น จะเป็นการตรวจจับจุดแต่ละจุดแยกออกจากกัน ทำให้เวลานำไปสร้างการเคลื่อนไหวจริงๆ จะเกิดความผิดพลาดขึ้นบ้าง เช่น การเคลื่อนไหวที่ไม่สมจริง ทั้งนี้ไม่ได้คำนึงถึงความสัมพันธ์ของจุดแต่ละจุด และการตรวจจับที่ต้องการแปลงจุดที่ตรวจจับได้ไปเป็นจุดตามมาตรฐานไฟล์ CSM นั้นทำให้เกิดข้อจำกัด เพราะไฟล์ CSM กำหนดจุดมาตรฐานไว้จำกัดจำนวนและตำแหน่ง ทำให้เราต้องใช้มาร์กเกอร์ซึ่งวางในตำแหน่งแบบเดียวกันซึ่งอาจไม่เหมาะสมกับการตรวจจับ ดังนั้นในระบบส่วนมากมักจะแยกส่วนที่ตรวจจับได้กับส่วนที่จะไปสร้างเป็นไฟล์ออกจากกัน โดยใช้โครงกระดูกเพื่อคำนวณการเคลื่อนไหว

โครงกระดูกนั้นจะทำหน้าที่ในการรับค่าจุดหลายๆ จุดที่ตรวจจับได้มาสร้างการเคลื่อนไหว โดยมีการคำนวณความสัมพันธ์ระหว่างจุด ข้อต่อ ทำให้การเคลื่อนไหวนั้นสมจริงและถูกต้องมากขึ้น ประกอบกับเราสามารถเพิ่มการปรับปรุงจุดหรือข้อกำหนดบางอย่างในการเคลื่อนที่ได้ในขั้นตอนนี้ จากนั้นจึงเอาการเคลื่อนไหวของโครงกระดูกนี้ไปสร้างเป็นไฟล์เพื่อไปใช้งานต่อไป ซึ่งไฟล์ที่เหมาะสม และเป็นที่นิยมใช้จะเป็นไฟล์แบบ BVH (Biovision Hierarchy) ซึ่งจะบันทึกการเคลื่อนไหวแบบความสัมพันธ์ระหว่างข้อต่อต่างๆ แทนที่จะบันทึกตำแหน่งแต่ละจุดแยกกันแบบ CSM และไฟล์แบบ BVH ยังสามารถนำไปใช้กับโปรแกรมสร้างการเคลื่อนไหวได้มากกว่า เพราะเป็นมาตรฐานและเป็นที่นิยมใช้งาน

5.2.3 การทำงานแบบรีลไทม์

ในการพัฒนาแบบรีลไทม์นั้นจำเป็นต้องอาศัยส่วนประกอบต่างๆ จำนวนมากเพื่อให้ระบบสามารถทำงานได้อย่างสมบูรณ์ ตั้งแต่อุปกรณ์ฮาร์ดแวร์ อุปกรณ์จับภาพที่ต้องมีความรวดเร็วในการจับภาพ การประมวลผล และการคำนวณเพื่อหาตำแหน่งออกมา ที่ต้องทำให้ได้รวดเร็วเพียงพอที่จะตรวจจับอย่างต่อเนื่อง ซึ่งในระบบจริงนั้นจะใช้การตรวจจับที่ฮาร์ดแวร์ โดยการใช้กล้องพิเศษที่สามารถประมวลผลภาพเพื่อหาตำแหน่งของจุดมาร์กเกอร์บนภาพแต่ละจุดได้ จากนั้นข้อมูลตำแหน่งจากกล้องแต่ละกล้องจะถูกส่งไปรวมกันที่อุปกรณ์อีกชุด ผ่านระบบเครือข่ายความเร็วสูงขนาด Gigabit เพื่อทำการคำนวณหาตำแหน่งจริงและแยกแยะจุดแต่ละจุด จากนั้นอุปกรณ์ชุดนี้จะถูกส่งเข้าคอมพิวเตอร์เพื่อนำจุดต่างๆ ไปสร้างการเคลื่อนไหวต่อไป

บรรณานุกรม

[1] นายชัยพร พรพุทธศรี, นายชัยรัตน์ อ่อนแย้ม: "ระบบจำลองการเคลื่อนไหวตัวละคร 3 มิติ", ปริญญานิพนธ์วิศวกรรมศาสตร์บัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 2547.

[2] กฤษฎา สวัสดี และ ภาณุพงศ์ ศิริพร ณ ราชสีมา: "โปรแกรมจับรูปแบบการเคลื่อนไหว", ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2545.

[3] Alberto Menache (2000): "Understanding Motion Capture for Computer Animation and Video Games", Morgan Kaufmann Publishers, San Francisco. 2000.

[4] Rafael C. Gonzalez and Richard E. Woods (2002): "Digital Image Processing Second Edition", Prentice-Hall, Inc., New Jersey. 2002.

[5] ปิยะบุตร สุทธิดารา (2547): "3ds Max 6 Basic", บริษัท ด่านสุทธากาพิมพ์ จำกัด, กรุงเทพฯ. 2547.