สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# โปรแกรมจับรูปแบบการเคลื่อนไหว

## MOTION CAPTURE

T049907

นาย กฤษฎา สวัสดี

นาย ภาณุพงศ์ ศิริพร ณ ราชสีมา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน 4990 7
วัน,เดือน,ปี - 2 มิ.ย. 2547
.b.....
.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# โปรแกรมจับรูปแบบการเคลื่อนไหว

# MOTION CAPTURE


โดย

นาย กฤษฎา สวัสดี

นาย ภาณุพงศ์ ศิริพร ณ ราชสีมา

อาจารย์ที่ปรึกษา

อาจารย์ เกียรติกูล เจียรนัยธนะกิจ


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมจับรูปแบบการเคลื่อนไหว

MOTION CAPTURE


ผู้จัดทำ

นายกฤษฎา สวัสดี รหัสประจำตัว 43015349

นายภาณุพงศ์ ศิริพร ณ ราชสีมา รหัสประจำตัว 43015375


_____ อาจารย์ที่ปรึกษา

(อาจารย์เกียรติกูล เจียรนัยธนะกิจ)


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# โปรแกรมจับรูปแบบการเคลื่อนไหว

นายกฤษฎา สวัสดี 43015349
นายภาณุพงศ์ ศิริพร ณ ราชสีมา 43015375
เกียรติกูล เจียรนัยธนะกิจ อาจารย์ที่ปรึกษา
ปีการศึกษา 2545

## บทคัดย่อ

ในปัจจุบันการพัฒนาทางทางด้านเทคโนโลยีคอมพิวเตอร์นั้นได้ก้าวหน้าไปอย่างรวดเร็ว โดยเฉพาะในด้านกราฟิกแอนิเมชัน ซึ่งในปัจจุบันนี้เราจะเห็นว่าภาพแอนิเมชัน 3 มิติได้เข้าไปมีบทบาทในการทำภาพยนตร์หรือภาพยนตร์โฆษณาหรือนำไปสร้างเป็นเกม และโดยส่วนมากจะเป็นการเคลื่อนไหวของสิ่งมีชีวิตเช่น มนุษย์ โดยการจะได้มาซึ่งภาพเคลื่อนไหวที่สมจริงนั้นเราจำเป็นต้องมีอุปกรณ์ที่เรียกว่าโมชันแคปเจอร์

โมชันแคปเจอร์คือการที่เราทำการแคปเจอร์การเคลื่อนไหวของมนุษย์(หรือการเคลื่อนไหวของสิ่งอื่นๆ) แล้วทำการการส่งข้อมูลที่ได้ทำการแคปเจอร์ไปยังคอมพิวเตอร์เพื่อจัดการกับข้อมูลเหล่านั้น ตัวอย่างการแคปเจอร์ เช่น การแคปเจอร์การเคลื่อนไหวส่วนต่างๆของร่างกาย แคปเจอร์การแสดงสีหน้า แคปเจอร์การเคลื่อนไหวของสัตว์เป็นต้น

ปริญญานิพนธ์ฉบับนี้จะทำการศึกษาการพัฒนาโปรแกรมจับรูปแบบการเคลื่อนไหว โดยใช้ภาษาแมกซ์สคลิปของโปรแกรมสามดีสตูดิโอแมกซ์ในการพัฒนาในส่วนของการทำแอนิเมชัน และใช้โปรแกรมวิชวลซีพลัสพลัสในส่วนของการคำนวณหาตำแหน่งการเคลื่อนที่จากไฟล์วีดีโอ โดยเราต้องทำการติดมาคร์เกอร์ตามส่วนต่างๆของร่างกาย และใช้กล้องในการแคปเจอร์การเคลื่อนไหวนั้นเพื่อนำมาคำนวณหาการเคลื่อนที่ของมาคร์เกอร์เหล่านั้นและเราจะนำผลที่ได้ไปทำเป็นภาพแอนิเมชัน 3 มิติที่มีการเคลื่อนไหวที่สมจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Motion capture**

Kridsada Sawasdee

Phanupong Siriporn na rachseema

Kietikul Jearanitanakij Advisor

**Abstract**

At present, Computer Technology is fast developed rapidly both in performance and ability such as the animation graphics. 3D animation to become for film, publicity and games development. At mostly that is human movement. We must have Motion Capture to create the realistic motion.

Motion Capture is the process of recording human movement (or other movement) in physical space, and transforming that information in a computer-usable form. Examples are measurements of full body motions, facial expressions, or animal motions.

This thesis is to study the motion capture application development that uses MaxScript of the 3D Studio Max to create animation development. And use C++ to process markers movement from video files. We must bind markers at the body of human. And use two cameras to capture that human movement. Take the result to make the realistic 3D motion.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้เกิดขึ้นได้เนื่องมาจากการทำงานร่วมกันในหลายๆ ส่วน บุคคลแรกที่ต้องกล่าวถึงคือ อาจารย์เกียรติกูล เจียรนัยธนะกิจ อาจารย์ที่ปรึกษาที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอ รวมไปถึงอาจารย์ท่านอื่นๆ ที่มิได้กล่าวถึง ที่ได้คอยแนะนำ และให้คำปรึกษาจนคลายความข้องใจ ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

และต้องขอขอบพระคุณบุคคลที่สำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจเอาใจใส่เสมอมา ในทุกๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณและขอกราบขอบพระคุณมา ณ ที่นี้

นายกฤษฎา สวัสดี

นายภาณุพงศ์ ศิริพร ณ ราชสีมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ


| | หน้าที่ |
|---|---|
| บทคัดย่อภาษาไทย | I |
| บทคัดย่อภาษาอังกฤษ | II |
| กิตติกรรมประกาศ | III |
| สารบัญ | IV |
| สารบัญตาราง | VII |
| สารบัญภาพ | VIII |
| **บทที่ 1 บทนำ** | 1 |
| 1.1 ความสำคัญและที่มา | 1 |
| 1.2 วัตถุประสงค์ของงานโครงงาน | 1 |
| 1.3 ขอบเขตในการดำเนินงาน | 2 |
| 1.4 วิธีดำเนินการ | 2 |
| **บทที่ 2 อิมเมจโปรเซสชิ่ง** | 3 |
| 2.1 การปรับปรุงคุณภาพของภาพ | 3 |
| 2.2 การแบ่งส่วนของภาพ | 5 |
| 2.2.1 ในการทำการแบ่งส่วน(Segment) | 5 |
| 2.2.2 การใช้ค่าเทรดโฮล (Thresholding) | 6 |
| 2.2.3 รีเจียนกโรอิง (Region Growing) | 9 |
| **บทที่ 3 แม็กซ์สคลิป (MaxScript)** | 11 |
| 3.1 การเรียกใช้ แม็กส์คลิป | 11 |
| 3.2 ข้อมูลต่างๆในแม็กส์สคลิป | 13 |
| 3.2.1 การกำหนดค่าตัวเลขในแม็กสคลิป | 13 |
| 3.2.2 การเรียกใช้สตริงในแม็กส์สคลิป | 14 |
| 3.2.3 การใช้อาร์เรย์ในแม็กสคลิป | 15 |
| 3.3 การกำหนดค่าให้กับตัวแปร | 15 |
| 3.4 การทำงานคณิตศาสตร์ในในแม็กส์สคลิป | 17 |
| 3.4.1 การทำงานพื้นฐานทางคณิตศาสตร์ | 17 |
| 3.4.2 การทำงานกับสตริง | 19 |
| 3.4.3 การสุ่มค่าตัวเลข | 19 |
| 3.4.4 การเพิ่มค่า | 20 |
| 3.5 การเรียกใช้และทำงานกับ สคลิปไฟล์ | 20 |
| 3.5.1 การทำงานกับสคลิปที่คุณได้ เก็บไว้แล้ว | 20 |
| 3.5.2 การเรียกใช้สคลิปอื่นๆ | 21 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

## (ต่อ)


3.6 การทำแอนิเมชันด้วยแมกซ์สคลิป 21

3.6.1 การวาดกล่องด้วย แม็กสคลิป 22

3.6.2 การแก้ไขค่าต่างๆของกล่อง 24

3.6.3 การเปลี่ยนสีของกล่อง 24

3.6.4 การเปลี่ยนตำแหน่งของกล่อง 25

3.6.5 การเปลี่ยนขนาดของกล่อง 25

**บทที่ 4 Stereopsis** 27

4.1 ปัญหาที่เกิดขึ้นในการทำ Stereopsis 28

4.2 ระบบ Stereopsis อย่างง่าย 30

4.3 การแก้ปัญหาของการหาจุดที่แสดงถึงจุดในโลกจริงเดียวกัน 31

4.3.1 วิธี Correlation – based 31

4.3.2 วิธี Feature – based 32

4.4 การสร้าง Cartesian coordinate เมื่อทราบตำแหน่งที่ต้องการ 32

4.5 เรขาคณิตของภาพ 34

4.6.1 โมเดลพื้นฐานของเรขาคณิตของภาพ 35

**บทที่ 5 รูปแบบของไฟล์เอวีไอ** 36

5.1 ส่วนหัวของไฟล์ เอวีไอ (AVI Header Format) 36

5.2 ส่วนหัวของ เฟรม เอวีไอ (Frame AVI Header Format) 37

**บทที่ 6 การสร้างโมเดล 3 มิติ** 41

6.1 หลักการสร้างกราฟิก 3 มิติด้วยโปรแกรม 3 มิติ สตูดิโอ แมกซ์ 41

6.2 โครงสร้างและหน้าจอการทำงานภายในโปรแกรม 3 41

6.3 การ เชื่อมต่อ (Merge) ไฟล์ 43

6.4 การ อิมพอร์ต / เอ็กซ์พอร์ตไฟล์ 43

6.5 การสร้างวัตถุ 3 มิติรูปทรงพื้นฐาน 44

6.5.1 สร้างวัตถุโดยการใช้เมาส์ในจอภาพย่อย 44

6.5.2 สร้างกล่องโดยการกำหนดค่าจากคีย์บอร์ด 45

6.6 การกำหนดพื้นที่ผิววัตถุ 45

**บทที่ 7 การออกแบบโปรแกรมจับรูปแบบการเคลื่อนไหว** 47

7.1 ภาพรวมของโครงงาน 47

7.2 การออกแบบและการเขียนโปรแกรม 48


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**สารบัญ**

**(ต่อ)**


7.2.1 การบันทึกภาพเคลื่อนไหว (Image capture) 48

7.2.2 อิมเมจโปร์เซ็สซิ่ง (Image processing) 49

7.2.3 การวิเคราะห์ค่าความผิดพลาด (Error analysis) 50

7.2.4 ซเทริโอออฟซิส (Stereopsis) 51

7.2.5 การสร้างไฟล์แมกซ์สคลิป (Generate MaxScript) 51

7.2.6 การทำแอนิเมชัน (Make animation) 53

7.3 แผนภาพต่างๆ ของการออกแบบระบบ (Diagram) 59

7.3.1 ยูสเครสไดอะแกรม (Use Case Diagram) 59

7.3.2 ซีเควนไดอะแกรม (Sequence Diagram) 60

7.3.3 คลาสไดอะแกรม(Class Diagram) 62

**บทที่ 8 ผลการทดลองและการทดสอบ** 64

8.1 วัตถุประสงค์ในการทดสอบ 64

8.2 การทดสอบโปรแกรมจับรูปแบบการเคลื่อนไหว 64

**บทที่ 9 บทวิจารณ์และสรุป** 76

9.1 ประเมินผล 76

9.2 แนวทางการพัฒนาต่อ 76

9.3 สรุป 76

**ภาคผนวก ก** 77

**บรรณานุกรม** 83


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง


| | หน้าที่ |
|---|---|
| ตารางที่ 5.1 รูปแบบส่วนหัวของไฟล์ เอวีไอ | 36 |
| ตารางที่ 5.2 รูปแบบส่วนหัวของไฟล์ เอวีไอ | 38 |




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูปภาพ


| | หน้าที่ |
|---|---|
| รูปที่ 2.1 การใช้ลิเนียร์ทรานฟอร์มแบบที่หนึ่ง | 3 |
| รูปที่ 2.2 การใช้ลิเนียร์ทรานฟอร์มแบบที่สอง | 4 |
| รูปที่ 2.3 การใช้ลิเนียร์ทรานฟอร์มแบบที่สาม | 4 |
| รูปที่ 2.4 การทำการแบ่งส่วนของภาพ | 5 |
| รูปที่ 2.5 ภาพที่ได้จากการทำเทรดโฮล | 6 |
| รูปที่ 2.6 ผลจากการปรับค่าเทรดโฮลเป็นค่าต่างๆ | 7 |
| รูปที่ 2.7 ผังขั้นตอนในการหาค่าเทรดโฮล ( flow chart thresholds ) | 8 |
| รูปที่ 2.8 ขั้นตอนการทำรีเจียนกโรอิง (Region Growing) | 9 |
| รูปที่ 2.9 ผังขั้นตอนในการทำรีเจียนกโรอิง(Region Growing flow ) | 10 |
| รูปที่ 3.1 พานัลของแม็กส์สคลิป | 12 |
| รูปที่ 3.2 หน้าต่างรอรับคำสั่ง แม็กสคลิป | 12 |
| รูปที่ 3.3 ประเภทตัวแปรแบบตัวเลข | 13 |
| รูปที่ 3.4 การเรียกใช้สตริงในแม็กส์สคลิป | 14 |
| รูปที่ 3.5 เมื่อใส่สตริงในรูปแบบที่ถูกต้อง | 14 |
| รูปที่ 3.6 การใช้อาร์เรย์ | 15 |
| รูปที่ 3.7 แสดงการกำหนดค่าให้กับตัวแปรแบบสตริง | 16 |
| รูปที่3.8 การเรียกใช้ตัวแปรแบบสตริง | 16 |
| รูปที่ 3.9 การใช้คำสั่งทางคณิตศาสตร์ ในแม็กส์สคลิป | 17 |
| รูปที่ 3.10 การป้อนค่าคงที่ทางคริตศาสตร์ | 18 |
| รูปที่ 3.11 การคำนวรทางคณิตศาสตร์ด้วยหลายๆคำสั่ง | 18 |
| รูปที่ 3.12 การคำนวณทางคณิตศาสตร์กัน สตริง | 19 |
| รูปที่ 3.13 รูปแบบการสุ่มค่า | 20 |
| รูปที 3.14 แสดง Code MaxScript ที่ใช้ทำภาพแอนิเมชัน | 21 |
| รูปที่ 3.15 การสร้างกล่องและกำหนดค่าพารามิเตอร์ | 23 |
| รูปที่ 3.16 กล่องที่ถูกวาดโดยใช้แม็กส์สคลิป | 24 |
| รูปที่ 3.17 การเปลี่ยนตำแหน่งของกล่อง | 25 |
| รูปที่ 3.18 คำสั่งแสดงค่าพารามิเตอร์ต่างๆของออปเจ็ก | 26 |
| รูปที่ 4.1 รูปภาพที่ใช้ในการหา ตำแหน่งของวัตถุ (Cartesian coordinate) | 27 |
| รูปที่ 4.2 จุดภาพบน (ก) (ข) ที่เป็นจุดเดียวกันในโลกจริง | 28 |
| รูปที่ 4.3 แสดงมุมมองที่ได้จากกล้องสองตัว | 29 |
| รูปที่ 4.4 โมเดลของเรขาคณิตของจุด | 30 |
| รูปที่ 4.5 โมเดลของระบบ Sstereo เพื่อใช้ในสมการเพื่อหาค่า X | 32 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูปภาพ

## (ต่อ)

รูปที่ 4.6 โมเดลของระบบ Stereo เพื่อใช้ในการสร้างสมการเพื่อหาค่า Y 33
รูปที่ 4.7 โมเดลของระบบ Stereopsis เพื่อใช้อธิบายการหาค่า Z 33
รูปที่ 4 .8 โมเดลของระบบ Stereo ที่ถูกสร้างขึ้นเพื่อหาค่า X3 X4 34
รูปที่ 4.9 เรขาคณิตภาพ 35
รูปที่ 5.1 แสดง รูปแบบ ของเฟรม เอวีไอ 37
รูปที่ 6.1 แสดงหน้าจอการทำงานภายในโปรแกรม 3 มิติ สตูดิโอ แมกซ์ เวอร์ชัน 5.0 41
รูปที่ 6.2 แสดงส่วนควบคุมหน้าจอย่อย 42
รูปที่ 6.3 แสดงสร้างวัตถุโดยการใช้เมาส์ในจอภาพย่อย 44
รูปที่ 6.4 แสดงแถบคำสั่ง Rollout หัวข้อ Parameters 45
รูปที่ 6.5 แสดงคุณสมบัติเฉดสีของวัตถุ 45
รูปที่ 6.6 แสดงการกำหนดคุณสมบัติ 46
รูปที่7.1 แสดงภาพโดยรวมของระบบ 47
รูปที่ 7.2 แสดงอุปกรณ์มาคร์เกอร์ที่ใช้ติดตามส่วนต่างๆของร่างกาย 48
รูปที่7.3 แสดงตำแหน่งที่เราทำการติดอุปกรณ์มาคร์เกอร์ 49
รูปที่ 7.4 แสดง Format ของเฟรม AVI 50
รูปที่ 7.5 แสดงตัวอย่างโค้ดแมกซ์สคลิป 52
รูปที่ 7.6 แสดงโครงสร้างร่างกายของมนุษย์ 53
รูปที่ 7.7 ไฟล์สคลิปส่วนที่ 1 54
รูปที่ 7.8 ไฟล์สคลิปส่วนที่ 2 55
รูปที่ 7.9 ไฟล์สคลิปส่วนที่ 3 56
รูปที่7.10 ไฟล์สคลิปส่วนที่ 4 57
รูปที่ 7.11 แสดงวิธีการรันไฟล์แมกซ์สคลิป 58
รูปที่ 7.12 แสดงยูสเคสไดอะแกรมของระบบ 59
รูปที่ 7.13 แสดงซีเควนไดอะแกรมของระบบ 60
รูปที่ 7.14 แสดงคลาสไดอะแกรมของโปรแกรมจับรูปแบบการเคลื่อนไหว 62
รูปที่ 8.1 กราฟของภาพแอนิเมชันเมื่อคีย์เฟรมเป็น 1 65
รูปที่ 8.2 กราฟของภาพแอนิเมชันเมื่อคีย์เฟรมเป็น 5 66
รูปที่ 8.3 กราฟของภาพแอนิเมชันเมื่อคีย์เฟรมเป็น 10 67
รูปที่ 8.4 กราฟของภาพแอนิเมชันเมื่อคีย์เฟรมเป็น 15 68
รูปที่ 8.5 คีย์เฟรมเป็น 1 70
รูปที่ 8.6 คีย์เฟรมเป็น 5 70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูปภาพ

(ต่อ)

รูปที่ 8.7 คีย์เฟรมเป็น 10 71
รูปที่ 8.8 คีย์เฟรมเป็น 15 71
รูปที่ 8.9 คีย์เฟรมเป็น 1 72
รูปที่ 8.10 คีย์เฟรมเป็น 5 73
รูปที่ 8.11 คีย์เฟรมเป็น 10 74
รูปที่ 8.12 คีย์เฟรมเป็น 15 75




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

## 1.1 ความสำคัญและที่มา

ในปัจจุบันการพัฒนาทางด้านเทคโนโลยีคอมพิวเตอร์นั้นได้ก้าวหน้าไปอย่างรวดเร็ว โดยเฉพาะในด้านกราฟิกแอนิเมชัน ซึ่งในปัจจุบันนี้เราจะเห็นว่าภาพแอนิเมชันได้เข้าไปมีบทบาททางด้านสื่อต่างๆ ไม่ว่าจะเป็นทางโทรทัศน์หรือทางเว็บไซท์ต่างๆ หรือการนำภาพแอนิเมชัน 3 มิติไปใช้ในการทำภาพยนตร์หรือภาพยนตร์โฆษณาหรือนำไปสร้างเป็นเกม และโดยส่วนมากจะเป็นการเคลื่อนไหวของสิ่งมีชีวิตเช่น มนุษย์ หรือ สัตว์ โดยการจะได้มาซึ่งภาพเคลื่อนไหวที่สมจริงนั้นเราจำเป็นต้องมีอุปกรณ์ที่เรียกว่าโมชันแคปเจอร์ (Motion Capture) ซึ่งมีราคาสูงมากเกินกว่าความสามารถของบริษัทจำนวนมากที่จะหามาใช้ได้

โครงงานโปรแกรมจับรูปแบบการเคลื่อนไหวของวัตถุนี้ในขั้นแรกได้ทำการศึกษาในเรื่องของมุมกล้องที่จะนำมาใช้ในการคำนวณหาค่าพิกัดในโลก 3 มิติ และได้ศึกษาซอฟแวร์ ที่ใช้ในการพัฒนาโครงการ เช่น โปรแกรมสามดีสตูดิโอแมกซ์ เป็นต้น โดยจะใช้การติดมาร์คเกอร์ที่บริเวณข้อต่อต่างๆที่สนใจ แล้วไฟล์ภาพที่ได้จากกล้องจะอยู่ในรูปของไฟล์วีดีโอเอวีไอ แล้วเราจะนำแต่ละ เฟรมที่ได้มาทำการคำนวณหาค่าตำแหน่งในโลก 3 มิติ แล้วจึงนำตำแหน่งที่ได้ มาทำการบังคับตัวละครที่เราต้องการให้มีการเคลื่อนไหวได้เหมือนตัวตนแบบให้มากที่สุด

สำหรับโครงงานนี้จะทำการพัฒนาในส่วนของกราฟิก 3 มิติซึ่งในโครงงานนี้ได้เลือกใช้โปรแกรมสามดีสตูดิโอแมกซ์ เวอร์ชัน 5.0 ( 3D Studio Max 5.0 ) สร้างโมเดลรูปคนให้มีความเสมือนจริง รวมทั้งการเขียนโปรแกรมเพื่อใช้วิเคราะห์ภาพถ่ายวีดีโอการเคลื่อนไหวของมนุษย์ที่ได้จากกล้องทั้งสองตัว โดยใช้ วิชวลซีพลัสพลัส (Visual C++ Version 6.0) ในการเขียนเพื่อวิเคราะห์ผลภาพที่ได้ ทำให้ได้พิกัดตำแหน่งต่างๆ ของร่างกายบุคลที่เราใช้กล้องจับ เพื่อนำมาแม็ชกับตัวละครที่เราได้สร้างไว้ในโปรแกรมสามดีสตูดิโอแมกซ์ โดยใช้แมกซ์สคลิปในการทำ แมปปิ้ง(mapping) และ แอนิเมชัน ทำให้ได้ภาพแอนิเมชันของตัวละครมีความสมจริงมีการเคลื่อนไหวได้ใกล้เคียงกับการเคลื่อนไหวของมนุษย์มากที่สุด

## 1.2 วัตถุประสงค์ของโครงงาน

1.2.1 สามารถสร้างงานแอนิเมชัน ที่สมจริงจากการเคลื่อนไหวจริงของวัตถุ

1.2.2 สามารถนำความรู้มาประยุกต์ใช้ในการพัฒนาโปรแกรม 3 มิติได้

1.2.3 สามารถนำภาพแอนิเมชันที่ได้ไปช่วยในอุตสาห์กรรมด้านภาพยนตร์,การกีฬา,เกมและอื่นๆอีกมากมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.4 เพื่อเรียนรู้ขั้นตอนต่างๆของการเขียนและพัฒนาซอฟท์แวร์ตั้งแต่การวิเคราะห์ความต้องการ การออกแบบ การเขียนโปรแกรม และการทดสอบ

**1.3 ขอบเขตในการดำเนินงาน**

โครงงานนี้จะเป็นการสร้างโปรแกรมจับรูปแบบการเคลื่อนไหว ซึ่งจะนำเสนอในรูปแบบภาพกราฟิกแอนิเมชัน 3 มิติ โดยจะสร้างตัวโมเดลการ์ตูนรูปคนที่อยู่ในรูป 3 มิติ และเราจะใช้กล้องวีดีโอสองตัวในการจับการเคลื่อนไหวของมนุษย์โดยเราจะนำอุปกรณ์เปล่งแสงมาติดตามส่วนต่างๆของร่างกาย และผลลัพธ์ที่ออกมาจากการแคปเจอร์จะได้อยู่ในรูปไฟล์วีดีโอเอวีไอ ออกมาสองไฟล์จากกล้องสองตัว แล้วนำไฟล์นี้ไปคำนวณหาตำแหน่งส่วนต่างๆของร่างกายในโลกจริงแล้วนำไปแปลงเป็นพิกัดในโลก 3 มิติเพื่อนำไปแม็ชกับตัวหุ่น 3 มิติที่เราได้สร้างไว้แล้วโดยใช้โปรแกรมแมกซ์สคลิปในการทำแอนิเมชันซึ่งจะได้ภาพออกมาเป็นภาพแอนิเมชันที่เหมือนกับการเคลื่อนไหวของตัวต้นแบบ

แต่จะมีบางอิริยาบถที่เราไม่สามารถจับได้เนื่องจากมีกล้องเพียงแค่ 2 ตัว ทำให้จุดที่เราติดอุปกรณ์ไว้บางจุด กล้องไม่สามารถจับได้อาจเนื่องมาจากหลุดออกจากกรอบของกล้องหรือร่างกายของเราไปบังส่วนนั้นไว้ จึงทำให้ไม่สามารถตรวจจับตรงอุปกรณ์ที่ติดตรงส่วนนั้นได้ทำให้ไม่รู้ว่าตรงส่วนนั้นของร่างกายอยู่ในลักษณะท่าทางแบบใด

**1.4 วิธีการดำเนินงาน**

1.4.1 ศึกษาทฤษฎีด้านอิมเมจโปรเซสซิ่ง (Image Processing)

1.4.2 ศึกษาโปรแกรม ประยุกต์ 3 มิติ เช่น โปรแกรมสามดีสตูดิโอแมกซ์เวอร์ชัน 5.0

1.4.3 ศึกษาภาษาที่จะนำมาใช้เขียนในการควบคุมการเคลื่อนไหวกระดูกของตัวการ์ตูนเช่น ไดเรกเอ็ค, แมกซ์สคลิป

1.4.4 ศึกษารูปแบบฟอร์แมตของไฟล์เอวีไอ

1.4.5 วิเคราะห์และออกแบบระบบ รวมถึงกำหนดขอบเขตในการทำงานของโปรแกรมจับรูปแบบการเคลื่อนไหว

1.4.6 กำหนดและเลือกเครื่องมือที่จะนำไปพัฒนาโปรแกรมจับรูปแบบการเคลื่อนไหว

1.4.7 เริ่มเขียนโปรแกรมโดยจะแบ่งโครงงานออกเป็น 2 ส่วนคือ ส่วนของการควบคุมการเคลื่อนไหวและส่วนของการทำอิมเมจโปรเซสซิ่ง

1.4.8 ทำการทดสอบโปรแกรมทั้ง 2 ส่วน

1.4.9 แก้ไขส่วนที่ผิดพลาดให้มีการคำนวณที่ผิดพลาดน้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่2
# อิมเมจโปรเซสชิ่ง
## (Image processing)

ในงานวิจัยนี้ใช้ทฤษฎีเกี่ยวกับอิมเมจโปรเซสชิ่งในหลายส่วน เพื่อใช้ในการปรับปรุงคุณภาพของภาพ และ การแบ่งส่วนของภาพ ที่ได้จาก แต่ละเฟรมของไฟล์ วีดีโอ เพื่อหาตำแน่งของจุดเซ็นเซอร์ที่เราสนใจ โดยจะมีการใช้ ทฤษฎีต่างๆดังนี้

การปรังปรุงของภาพ (Image Enhancement) ซึ่งเป็นการปรังปรุงของภาพ ที่จะนำมาใช้ในการค้นหาตำแหน่งของเซ็นเซอร์ ซึ่งมีเทคนิคในการปรับปรุงอยู่ 2 แบบ คือ

- บนพื้นฐานพิกเซล (Pixel - based)
- บนพื้นฐานความถี่(Frequency - based)

ในงานวิจัยนี้เราจะใช้การปรับปรุงคุณภาพของภาพแบบพิกเซลเบส

การแบ่งส่วนของภาพ (Image Segmentation) ซึ่งจะใช้ในการแบ่งกลุ่มของเซ็นเซอร์ในภาพที่ได้จากแต่ละเฟรมของไฟล์วีดีโอ ซึ่งใช้เทคนิคในการแบ่งส่วนของภาพอยู่ 2 วิธี

- การใช้ค่าเทคร์โฮล (Thresholding)
- รีเจียน โกรน ( Region Growing )

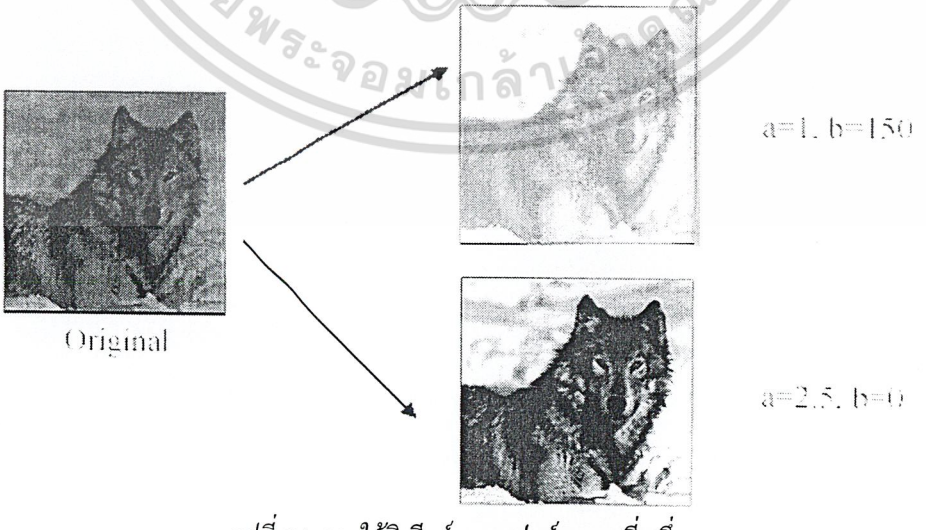
โดยจะกล่าวถึงรายละเอียด ดังนี้

**2.1 การปรับปรุงคุณภาพของภาพ (Image Enhancement)**

โดยจะมีการใช้ปรับปรุงคุณภาพของภาพเพื่อใช้ในการหาค่าความเข้มแสงที่เหมะสม โดยจะเลือกดังต่อไปนี้

ลิเนียร์ทรานฟอรม์ (Linear Transformation)

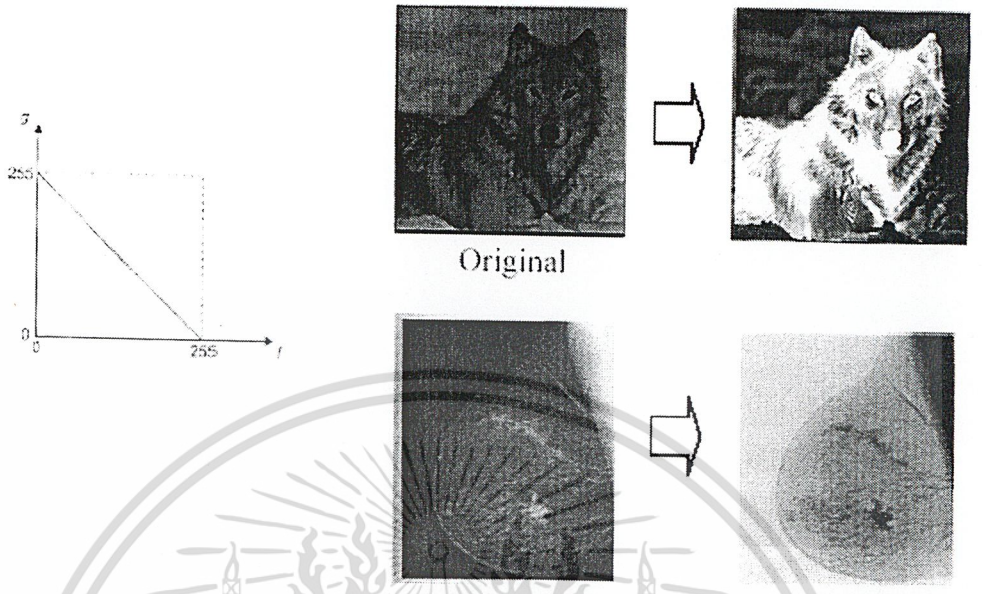
$$G(x, y) = af(x, y) + b$$




*รูปที่ 2.1 การใช้ลิเนียร์ทรานฟอร์มแบบที่หนึ่ง*

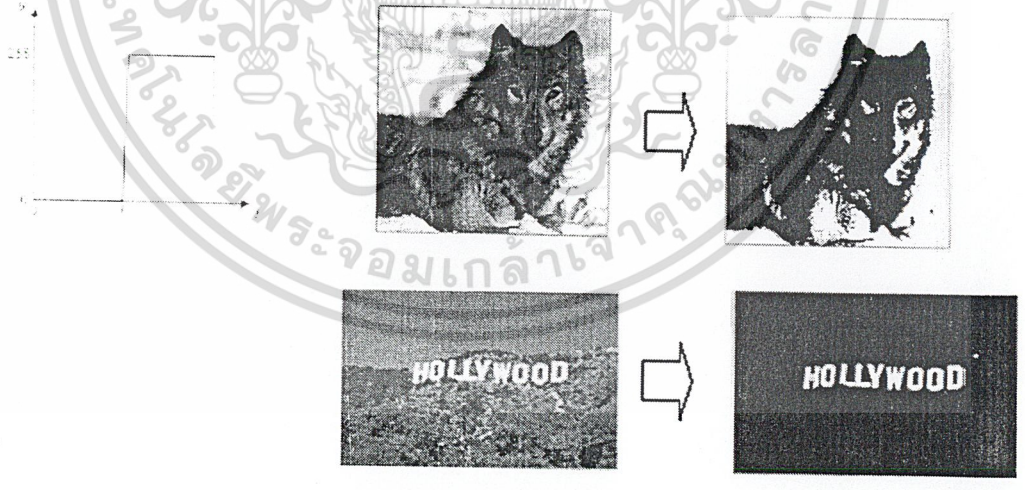
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการใช้ลิเนียร์ทรานฟอร์มสร้างภาพนีกาทีฟ (Negative)




*รูปที่ 2.2 การใช้ลิเนียร์ทรานฟอร์มแบบที่สอง*

เป็นการใช้ลิเนียร์ทรานฟอร์มแบบกำหนดค่าความเข้มแสงผ่าน คือ ถ้ามีค่าความเข้มแสงมากกว่าหรือเท่ากับให้มีค่าความเข้มแสงเท่ากับ 255 หรือถ้าต่ำกว่าให้ค่าความเข้มแสงเท่ากับ 0




*รูปที่ 2.3 การใช้ลิเนียร์ทรานฟอร์มแบบที่ สาม*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

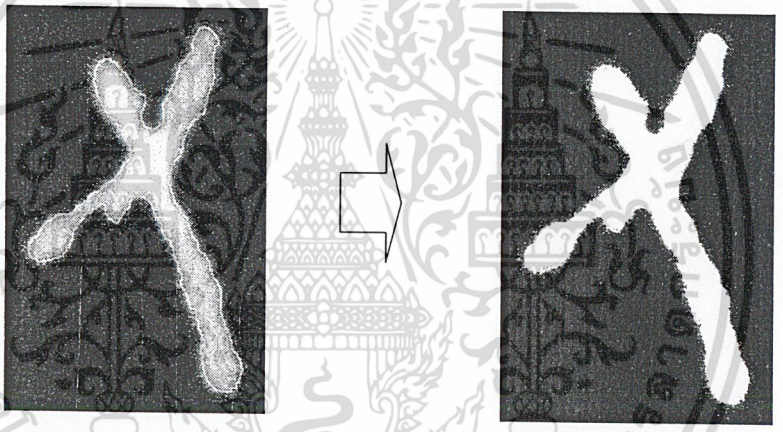
**2.2 การแบ่งส่วนของภาพ (Image Segmentation)**

ในการแบ่งกลุ่มของสี ในภาพที่ได้จากแต่ละเฟรมของไฟล์วีดีโอ ซึ่งใช้เทคนิคในการแบ่งส่วนของภาพซึ่งมีการแบ่งตามคุรสมบัติต่อไปนี้

- ความเข้มแสง
- สี
- ลักษณะพื้นผิว (texture)
- ลักษณะการเปลี่ยนแปลงของค่าความเข้มแสง (Gradient)

2.2.1 เทคนิคในการทำการแบ่งส่วน(Segment)

- ตรวจหาตำแหน่งการเปลี่ยนแปลง ของคุณสมบัติที่ทำให้เกิดการไม่ต่อเนื่อง (feature discontinuity)
- จับกลุ่มพิกเซลที่มี คุณสมบัติ (feature) ที่คล้ายคลึงกันเข้าด้วยกัน

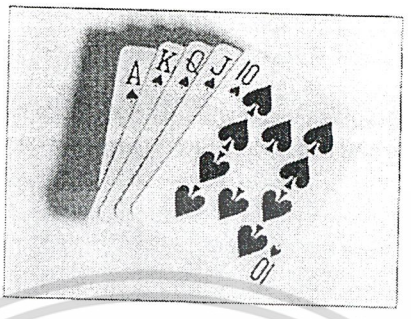


*รูปที่ 2.4 การทำการแบ่งส่วนของภาพ*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การใช้ค่าเทรดโฮล (Thresholding)

ค่าเทรดโฮล นั้นจะมีผลต่อภาพที่เราต้องการตัด เพราะ ฉะนั้นเราจึงควรรู้ว่าควรจะกำหนดค่า เทรดโฮล นี้ เท่าไหร่ ซึ่งมีวิธีการกำหนดดังนี้ คือ




*รูปที่ 2.5 ภาพที่ได้จากการทำเทรดโฮล*

จากรูปที่ 2.5 เรากำหนดให้ค่าเทรดโฮล ( T ) เท่ากับ 90 โดย

ถ้า $F(x,y) \leq T$ ให้ $G(x, y) = 255$

ถ้า $F(x,y) > T$ ให้ $G(x, y) = 0$

ให้เราเซ็ทค่า เทรดโฮล นี้ขึ้นมาค่าหนึ่ง ก่อนแล้วทำการแบ่งกลุ่มโดยที่ถ้าน้อยว่าค่า เทรดโฮลนี้ให้อยู่กลุ่มที่1 ถ้ามากกว่าให้อยู่กลุ่มที่2 แล้วทำการหาค่ากลาง (mean) ของแต่ละกลุ่ม แล้วคำนวณหาค่าเทรดโฮลนี้ใหม่โดย

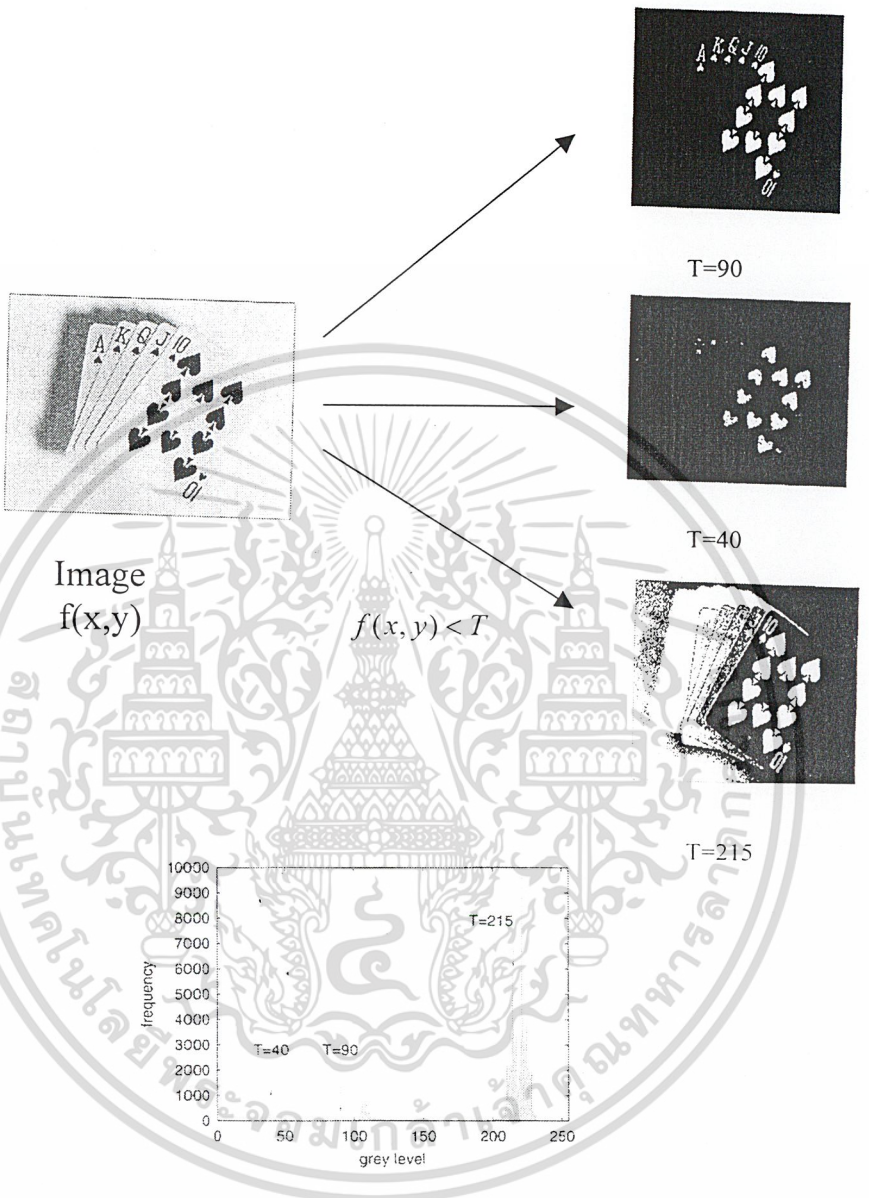
$$T = \frac{1}{2}(M1 + M2)$$

โดยที่ T =ค่าเทรดโฮล

M1 = ค่ากลางของกลุ่มที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

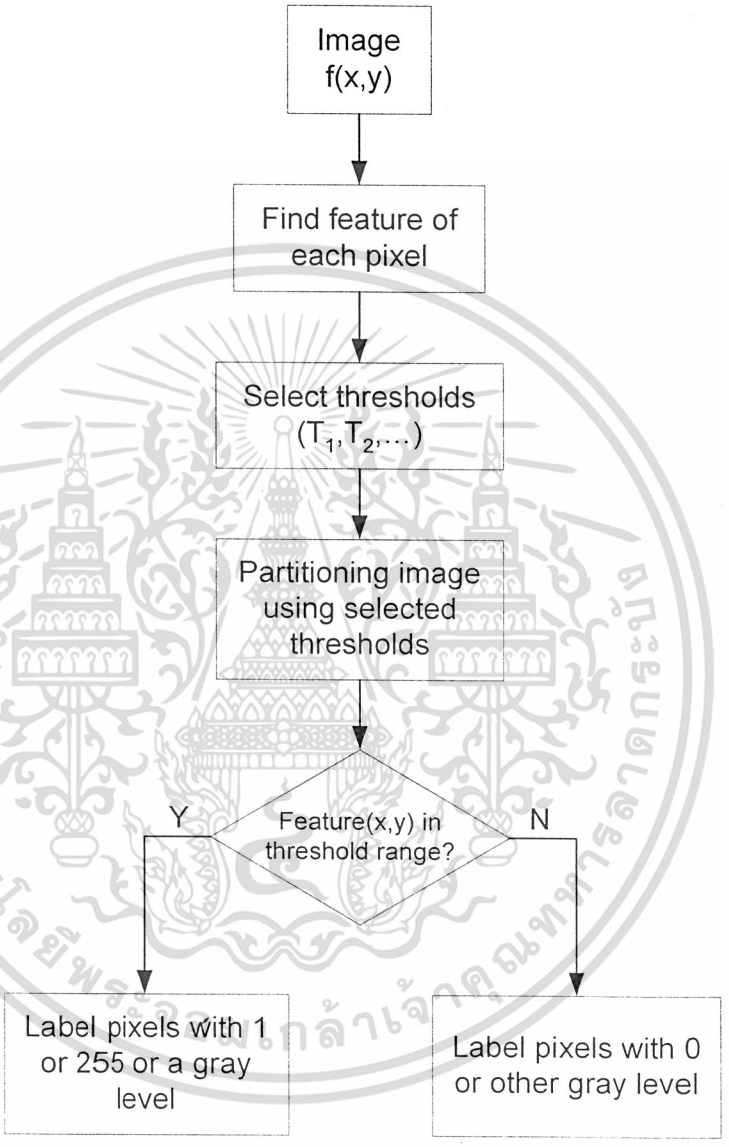
M2 = ค่ากลางของกลุ่มที่ 2




*รูปที่ 2.6 ผลจากการปรับค่าเทรดโฮลเป็นค่าต่างๆ*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งมี ผังการทำงานดังนี้

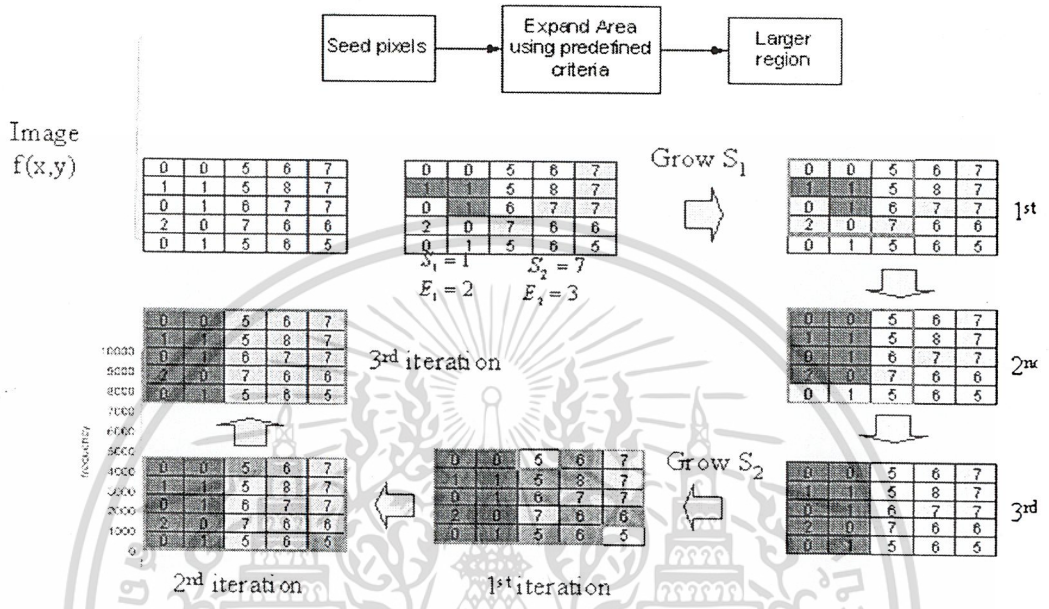



*รูปที่ 2.7 ผังขั้นตอนในการหาค่าเทรดโฮล (flow chart threshold )*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 รีเจียนกโรอิง (Region Growing)

จะใช้วิธีการขยายพื้นที่ๆสนใจ ไป 1 พิกเซลที่ติดกัน แล้วดูค่าความแตกต่างของความเข้มแสง ว่าต่างกันมากหรือน้อย ซึ่งขึ้นอยู่กับเราว่าจะ กำหนดลงไปว่าจะให้เป็นเท่าไรซึ่งในที่นี้กำหนดให้เป็นค่า E ซึ่งถ้าค่าที่พิจารณาน้อยกว่าค่า E ให้ถือว่าเป็นพื้นที่เดียวกับ




*รูปที่ 2.8 ขั้นตอนการทำรีเจียนกโรอิง( Region Growing )*

ข้อเสีย

- เสียเวลาในการคำนวณ
- ได้ผลช้า

ข้อดี

- ได้กลุ่มของพิกเซลที่มีความเข้มแสงใกล้เคียงกันจริงๆ

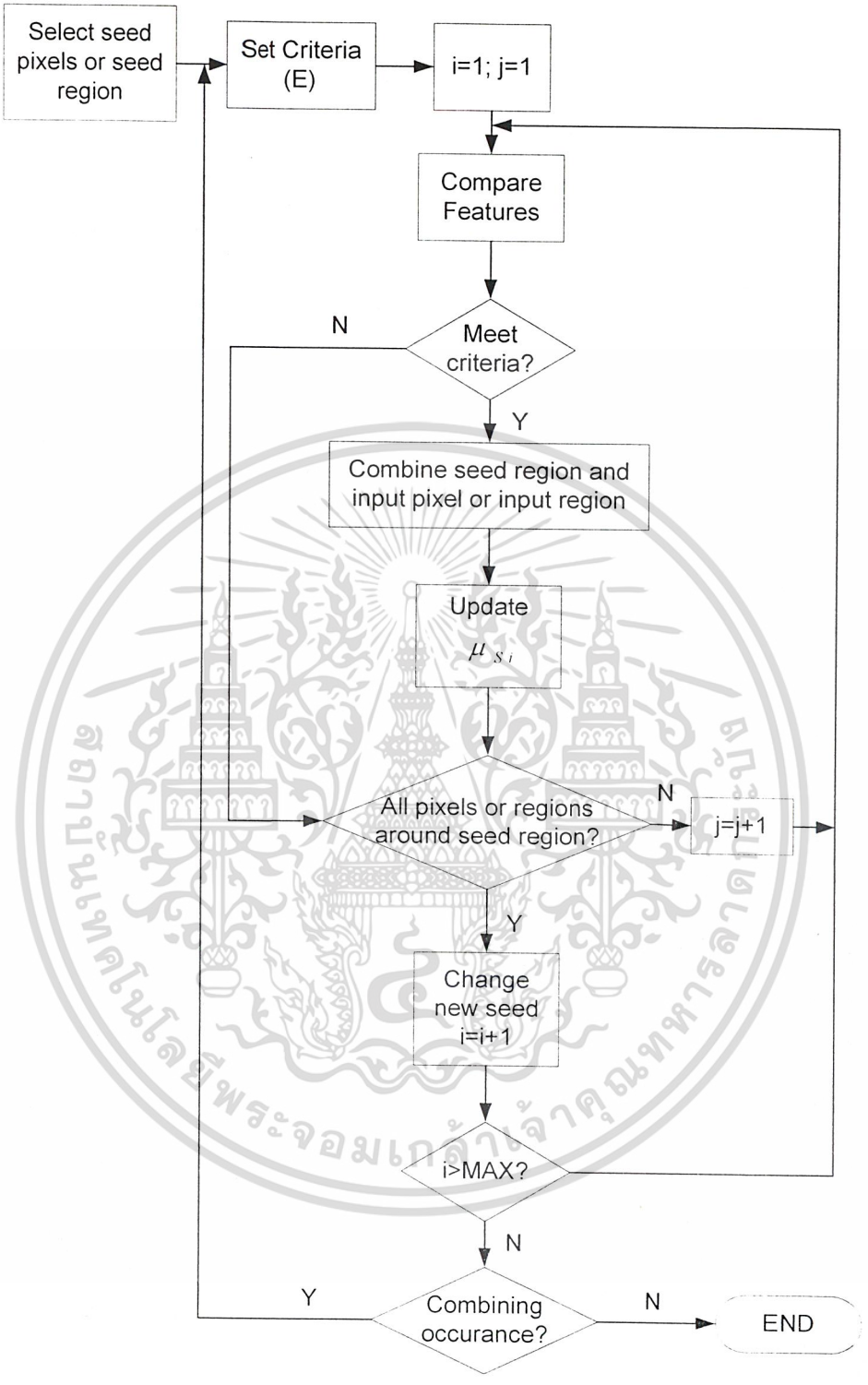
ขั้นตอนในการทำ รีเจียนกโรอิง (Region Growing)

1.กำหนดตำแหน่งที่จะทำการพิจารณา

2.กำหนดว่าจะให้ค่าความแตกต่างของความเข้มแสงเท่าไรจึงจะให้รวมกลุ่ม(กำหนดค่า E)

3.ทำการขยายพื้นที่ๆอยู่รอบๆ (growing) แล้วกำหนดค่าที่จะหยุดการขยายพื้นที่ด้วยว่าจะให้หยุดเมื่อไหร่ เช่น จำนวนรอบ , เวลา , ใช้คุณสมบัติอื่นๆเข้าช่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




*รูปที่ 2.9 ผังขั้นตอนในการทำรีเจียนกโรอิง (Region Growing flow )*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3
# แมกซ์สคลิป
# (MaxScript)

แม็กซ์สคลิปเป็นภาษาสคลิปที่สร้างขึ้นมาสำหรับสามดีสตูดิโอแม็กซ์ที่ใช้ช่วยในการทำงานภายในโปรแกรมโดยแม็กซ์สคลิปมีความสามารถดังนี้นำมาใช้ดังนี้

- สคริป เป็นรูปแบบหนึ่งในการใช้งานโปรแกรม เช่น การทำโมเดลลิ่ง,แอนิเมชัน,วัสดุ,การเรนเดอร์
- ในการควบคุมการทำงานของโปรแกรมนั้นจะเป็นแบบโต้ตอบกันทันทีซึ่งจะมีการใช้งานในลักษณะ คอมมานด์ไลน์
- สคริปถูกสร้างขึ้นมาใช้สำหรับการรวมวัตถุ,การแก้ไข,และการเรนเดอร์
- ผู้ใช้สามารถ อิมพอร์ต/เอ็กพอร์ต โดยการใช้ฟังก์ชั่น ไฟล์ อินพุต/เอาต์พุต
- ในการเขียนการควบคุมนั้นสามารถเข้าถึงสถานะของฉากได้
- เราสามารถสร้างเครื่องมือในการทำงานให้เป็นแบบ แบตช์โปรเซสซิ่งซึ่งเหมือนกับการทำสคริปแบตช์เรนเดอร์
- สามารถบันทึกการกระทำในสามดีสตูดิโอแม็กส์ ให้มาเป็น คำสั่ง แมกซ์สคลิป ได้

ภาษาแม็กซ์สคลิป เป็นภาษาที่ถูกออกแบบเพื่องานสามมิติอย่างครบถ้วนเพื่อใช้ใน สามดีสตูดิโอแม็กซ์ และยังสามารถทำงานในระบบพิกัดฉากได้ดีอีกด้วย, วัตถุที่ถูกสร้างจะถูกแสดงพร้อมกับแสดงลักษณะพื้นผิวออกมาผ่านทางยูสเซอร์อินเทอเฟรด และยังมีโหมดการทำแอนิเมชันอัตโนมัติคีย์เฟรม และการเข้าถึง วัตถุโดยใช้ ชื่อพาสต์ของวัตถุ ในสามดีสตูดิโอแม็กซ์ นั้นมีโครงสร้างของภาษาที่ง่ายโดยที่ผู้เขียนไม่จำเป็นต้องเป็นโปรแกรมเมอร์เลย ผู้ใช้สามารถป้อนคำสั่งต่างๆผ่านหน้าต่างที่รอรับคำสั่ง(MAXScript Listening Windows) และสามารถทำการบันทึกการกระทำผู้ใช้ มาเป็นคำสั่งแม็กซ์ สคลิปได้อีกด้วย แม็กซ์คลิป สามารถบันทึกบันทึกการกระทำขณะที่ผู้ใช้กำลังใช้งานอยู่ในสามดีสตูดิโอแม็กส์

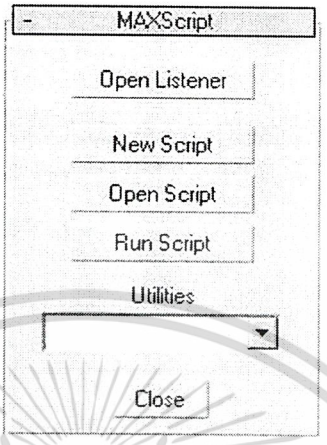
แม็กส์คลิปนั้นยังสามารถรองรับ อินพุตและเอาท์พุตในรูปแบบเท็กซ์ ได้ด้วยซึ่งมันอาจจะเป็นการรายงานการกระทำโดยตรงจาก สามดีสตูดิโอแม็กส์ หรือ สามารถอ่านไฟล์ที่บรรจุ การออกแบบฉาก หรือ รายละเอียดพื้นผิว และยังสามารถแปลงออกมาเพื่อนำไปใช้ในโปรแกรมอื่นๆได้อีกด้วย

### 3.1 การเรียกใช้ แม็กส์คลิป

แม็กสคลิปเป็นภาษาสคลิป ที่สร้างขึ้นมาใช้ใน สามดีสตูดิโอแม็กส์ เพราะว่าคุณสามารถใช้แม็กสคลิปได้เมื่อเปิด สามดีสตูดิโอแม็กเท่านั้น ในการเรียกใช้แม็กสคลิปนั้น ทางที่ง่ายที่สุดคือเปิด หน้าต่างที่คอยรอรับคำสั่งแม็กสคลิป จากแม็กสคลิปเมนู

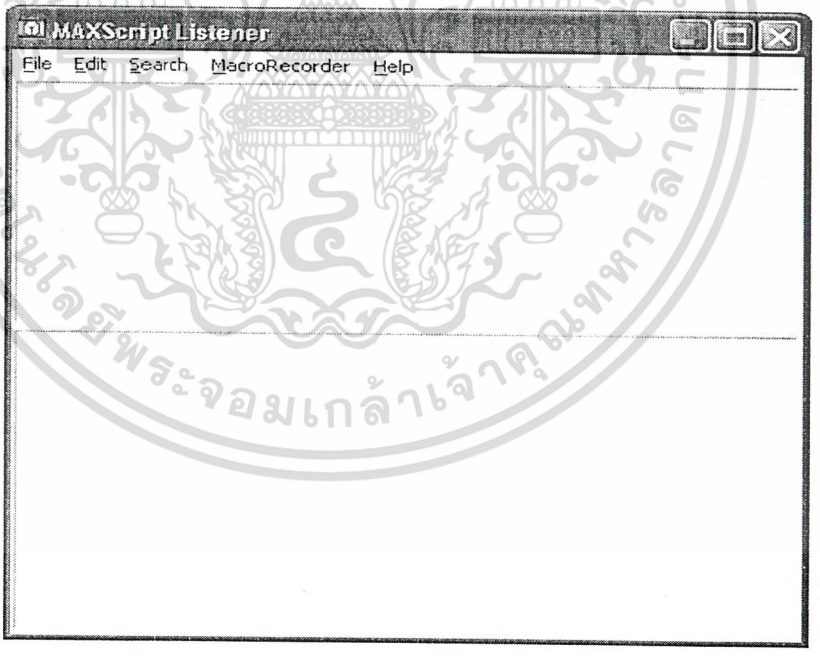
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. บน คอมมานด์พาแนล เลือกที่เครื่องมือแล้วคลิกที่ แม็กส์คลิป จะปรากฏไดอล็อกดังรูปที่ 3.1




*รูปที่ 3.1 พานัลของแม็กส์สคลิป*

2. คลิกเปิดหน้าต่างรอรับคำสั่ง( listener )




*รูปที่ 3.2 หน้าต่างรอรับคำสั่ง แม็กสคลิป*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่างรอรับคำสั่งแม็กส์สคลิป จะมีการทำงานแบบโต้ตอบกันทันที ภาษาแม็กสคลิป์เป็นภาษาที่ง่าย เมื่อป้อนเข้าสู้หน้าต่างคำสั่งแม็กส์สคลิปในหน้าต่างนี้ แล้วทำการกดเอ็นเทอร์มันก็จะทำการประมวลผลในทันที

หน้าต่างรอรับคำสั่งแม็กส์สคลิป สามารถเปิดขึ้นมาเมื่อเวลาใดก็ได้เพราะหน้าต่างนี้มีการทำงานแบบ โมเดลเลส ( Modeless ) เราจึงสามารถที่จะสลับหน้าต่างการทำงานไปมาได้ ถ้าคุณปิดหน้าต่างที่รอรับคำสั่งนี้แล้วปิดมันขึ้นมาอีกครั้ง ข้อความที่อยู่ในหน้าต่างก่อนหน้าที่จะปิด นั้นจะปรากฏขึ้นมาอีกครั้ง

เมื่อเราให้ มาโครเร็กคอร์ดนั้นเริ่มการทำงาน ทุกๆคำสั่งจะถูกบันทึกและแสดงใน มาโครเร็คคอรด์ เอ้าพุตที่เป็นผลลัพท์ที่มาจากสคลิปจะถูกแสดงในหน้าต่างเอ้าท์พุต และเอ้าพุตที่ได้จากการประมวลผลในมาโครเร็กคอรด์จะถูกแสดงผลโดยตรงไปยังหน้าต่างเอ้าท์พุต หน้าต่างแสดงผลนี้สามารถทำงานได้ทั้ง คัดลอกแล้ววาง , แก้ไข , เลือก และประมวลผล

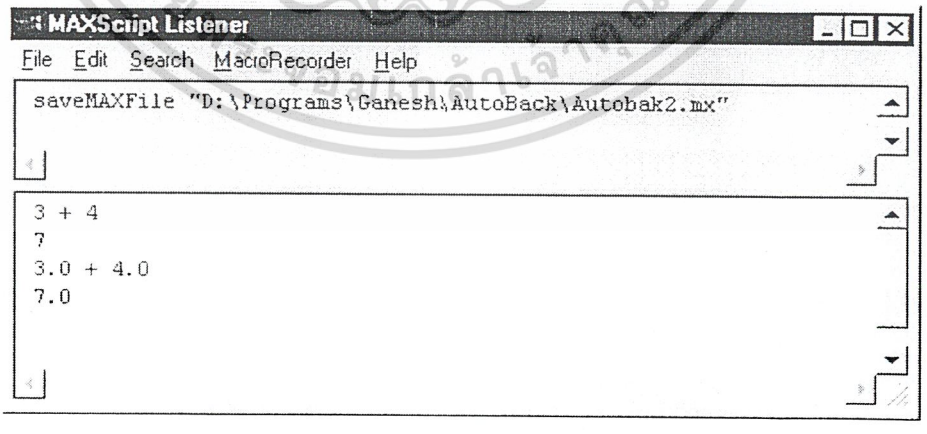
**3.2 ข้อมูลต่างๆในแม็กส์สคลิป**

ในแม็กส์สคลิปนั้นมีการใช้ ข้อมูลอยู่ สามแบบ คือ ตัวเลข สตริง และ อาร์เรย์ โดยทั่วไปการกำหนดค่าต่างๆในแม็กส์สคลิปนั้นจะทำโดยการ พิมพ์ หรือ การใช้เงื่อนไขและการส่งผ่านค่าโดยการใช้เอ็นเทอร์ อย่างไรก็ตามข้อมูลแต่ละประเภทก็ มีการกำหนดค่าที่แตกต่างกันออกไป

3.2.1 การกำหนดค่าตัวเลขในแม็กสคลิป

แม็กสคลิปจะมีการแบ่งการใช้ตัวเลขออกเป็น 2 ประเภทคือ อินทีเจอร์(integers) และ โฟรต(floats) อินทีเจอร์ นั้นเป็นเลขจำนวนเต็ม เช่น 0 , 1 , 2 , 10 , 527 โฟรต จะเป็นเลขทศนิยมที่บรรจุเลขฐานสิบ เช่น 2.5 , 72.0 ,และ 0.33

เมื่อ แม็กส์สคลิป อ้างถึงการทำงานกันตัวเลข ผลที่ได้โดยทั่วไปจะเหมือนกับ ค่าที่ส่งเข้าไปทำ สำหรับตัวอย่างเช่นส่ง 3+4 จะ ส่งค่า 7 กลับมา ,ขณะที่ 3.0 + 4.0 จะมีการส่งค่า 7.0 คืนกลับมา ดังรูปที่ 3.3




*รูปที่ 3.3 ประเภทตัวแปรแบบตัวเลข*

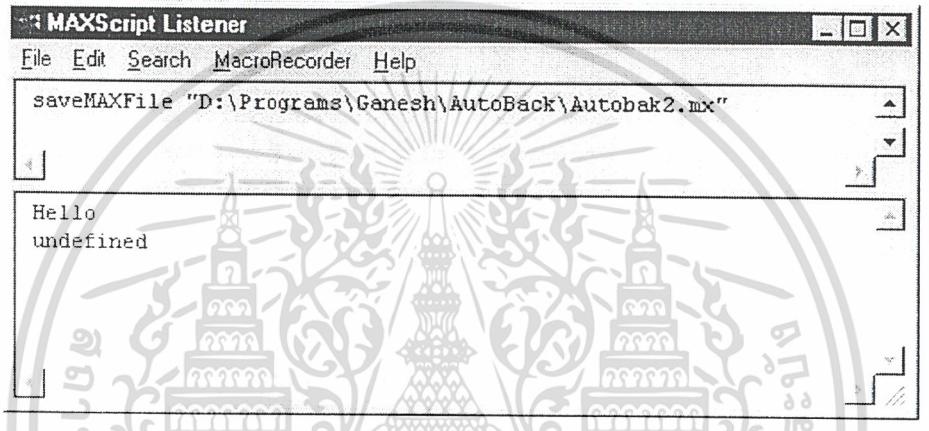
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 การเรียกใช้สตริงในแม็กส์สคลิป

แม็กส์สคลิก เป็นภาษาที่มีพื้นฐานมาจากการวางเงื่อนไข ทุกๆครั้งที่มีการสร้างจะต้องมีการกำหนดค่า การสร้างสตริงขึ้นมาในแม็กส์สคลิป นั้นจะมีการกำหนดค่าขึ้นมาให้อยู่ในเครื่องหมายอัญประกาศ

1. เมื่อเราพิมพ์ Hello แล้วกดเอ็นเทอร์

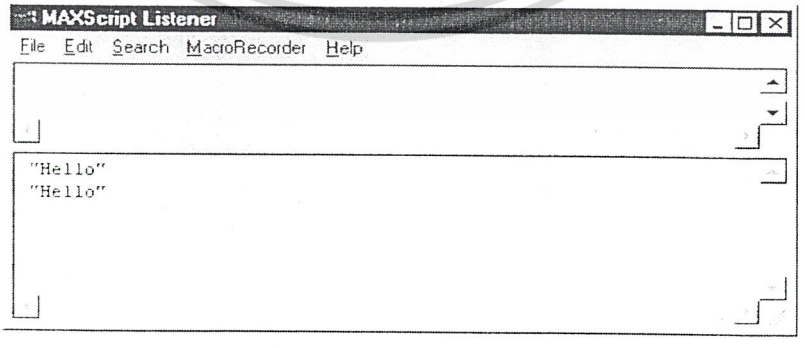
แม็กส์สคลิปจะตอบ ไม่รู้จัก(undefined) ซึ่งเป็นเหมือนคำสั่งจะถูกเรียกออกมายังเอ้าท์พุท ซึ่งจะปรากฏเป็นตัวสีฟ้าเสมอ เหตุที่มีการปรากฏเอ้าท์พุตเช่นนี้เพราะ แม็กส์สคลิปนั้นไม่รู้จักคำว่า hello มันไม่ได้กำหนดไว้ในภาษา เนื่องจาก แม็กส์สคลิปไม่สามารถอธิบายได้ว่าจะทำอะไรกับมัน




*รูปที่ 3.4 การเรียกใช้สตริงในแม็กส์สคลิป*

2. การเขียน คำว่า "Hello" อีกครั้ง ในครั้งนี้เราจะใส่อัญประกาศ แล้วกดเอ็นเทอร์ ผลลัพธ์ที่ออกคือ "Hello"

ตอนนี้แม็กส์สคลิป รู้จักสตริง "Hello" เนื่องจากมันตอบกับมาว่า "hello" อย่างถูกต้อง มันเพียงรู้ว่า สิ่งที่คุณป้อนเข้าไปนั้นเป็น สตริง เอ้าท์พุทของคำสั่งที่ป้อนเข้าไปนั้นเป็น สตริงดังนั้นค่าที่ได้กลับมาจึงเป็นสตริงที่ป้อนเข้าไปเสมอ




*รูปที่ 3.5 เมื่อใส่สตริงในรูปแบบที่ถูกต้อง*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

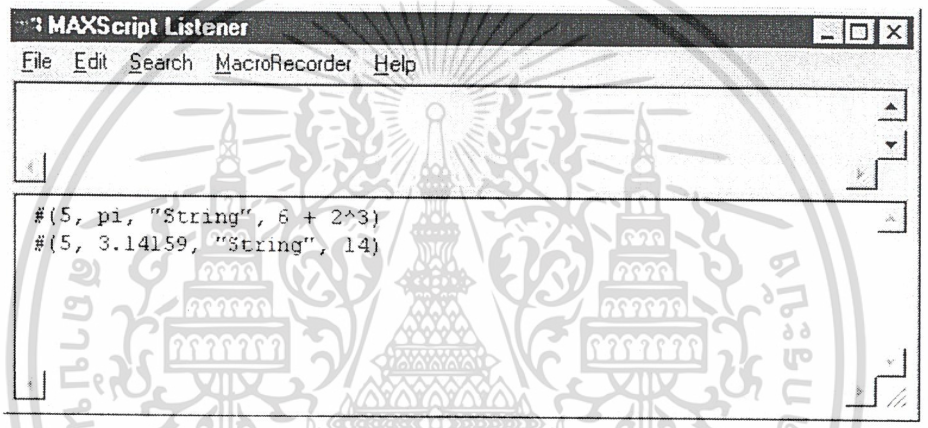
3.2.3 การใช้อาร์เรย์ในแม็กสคลิป

การเรียกใช้ อาร์เรย์ : ในการประกาศใช้ อาร์เรย์ทุกครั้งต้องมีตัวอักษร ( # ) และคู่กับข้อความในวงเล็บ

#( )

#( < expr > , < expr> )

ในแม็กส์สคลิปนั้น อาร์เรย์นั้นสามารถเก็บค่าของสมาชิกแต่ละตัวได้โดยที่ไม่จำเป็นต้องมี ข้อมูลเป็นประเภทเดียวกัน เช่นใน อาร์เรย์นั้นสมาชิกตัวที่หนึ่งอาจจะเก็บ อินทีเจอร์ แล้วตัวที่สองอาจจะเก็บเป็นสตริงก็ได้ดังรูปที่ 3.6




*รูปที่ 3.6 การใช้อาร์เรย์*

3.3 การกำหนดค่าให้กับตัวแปร

ในการเรียนรู้ที่ผ่านมาคุณได้ทำการป้อนสตริง "Hello" ในแม็กส์สคลิป และ แม็กส์สคลิปจะแสดง สตริงที่เหมือนกันนั้นออกมาทางเอ้าท์พุต เมื่อคุณป้อนสตริงอื่นเข้ามาแม็กส์สคลิป จะแสดงสตริงนั้น และค่าที่อยู่ก่อนหน้านี้ก็จะถูกลืมไป แม็กส์สคลิปจะจำค่าของสตริงที่ป้อนให้กับตัวแปร การป้อนตัวแปรให้กับแม็กส์สคลิปจะเป็นดังนี้ :

Variable_name = variable_value

Variable_name นั้นจะขึ้นต้นด้วยอักขระ ตัวอักษร หรือ "_" , ตามด้วยตัวเลข และ อักขระที่เป็นตัวอักษร

variable_value นั้นสามารถที่จะเป็น สตริง , ตัวเลข , หรือ อื่นๆ ตามเงื่อนไข

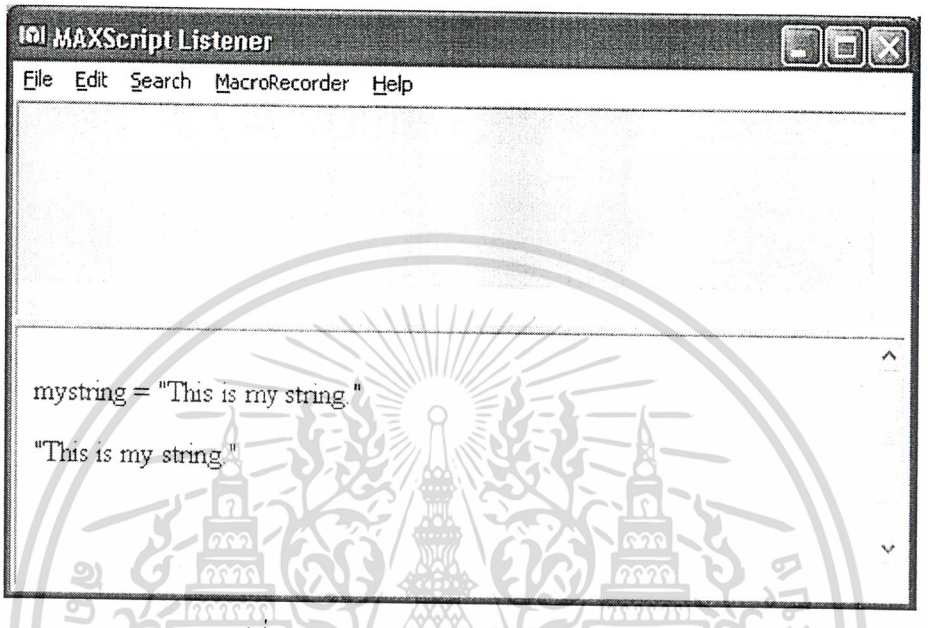
การกำหนดค่า สตริง ให้กับตัวแปร :

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

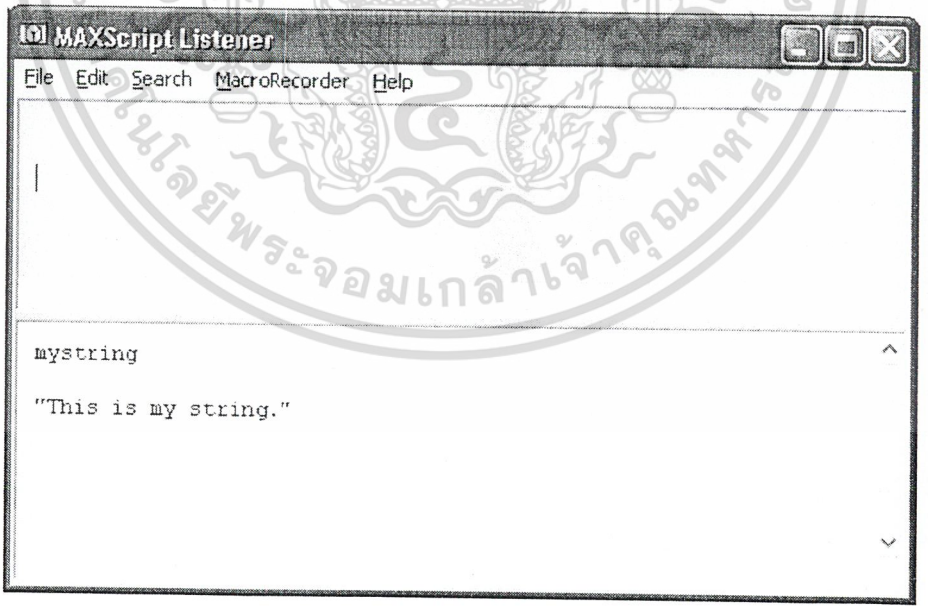
1. ที่หน้าต่างรอรับคำสั่ง เราสามารถป้อนคำสั่งตามนี้ :

mystring = " this is is my string."

แม็กส์สคลิปจะคืนค่า ของ ตัวแปรสตริง สตริงจะถูกเก็บเข้าไปในตัวแปร mystring ดังรูปที่ 3.7 และรูปที่ 3.8




*รูปที่ 3.7 แสดงการกำหนดค่าให้กับตัวแปรแบบสตริง*




*รูปที่ 3.8 การเรียกใช้ตัวแปรแบบสตริง*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

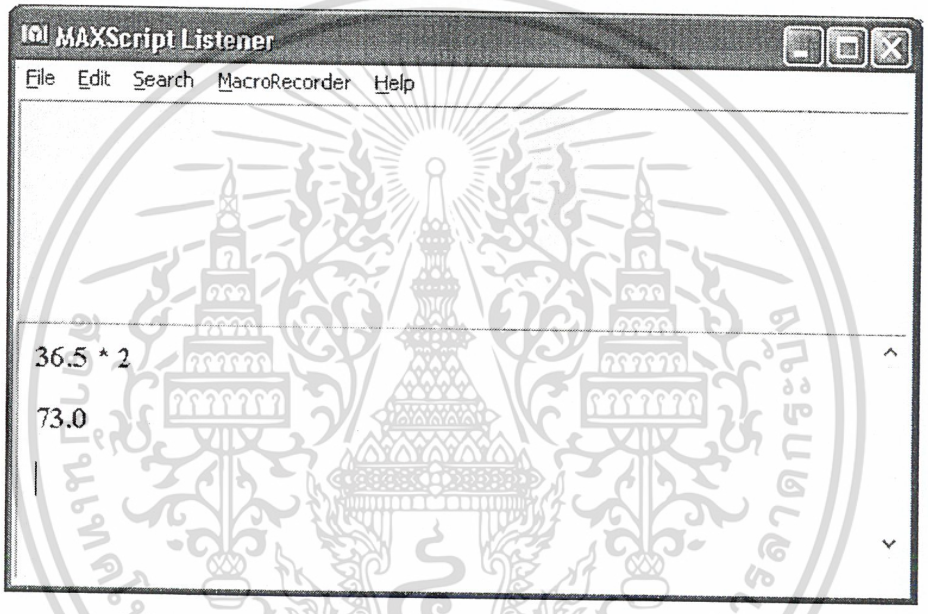


แม็กส์สคลิป จะไม่ให้ความแตกต่างระหว่าง อักษร ตัวเล็ก กับตัวใหญ่ ในการกำหนดชื่อตัวแปร เพราะ ชื่อ ใน แม็กส์สคลิป ไม่เป็น เคสเซนต์ซิทิป( casesensitive )

**3.4 การทำงานคณิตศาสตร์ในในแม็กส์สคลิป**

3.4.1 การทำงานพื้นฐานทางคณิตศาสตร์

สามารถนำแม็กส์สคลิปมาใช้ในการทำการคำนวณทางคณิตศาสตร์ได้ เช่นป้อน 36.5*2 เข้าไปซึ่งเห็นได้จากการคูณค่าขนาดใน3มิติ ผลที่ได้ออกมาเป็นค่า ดับเบิ้ล

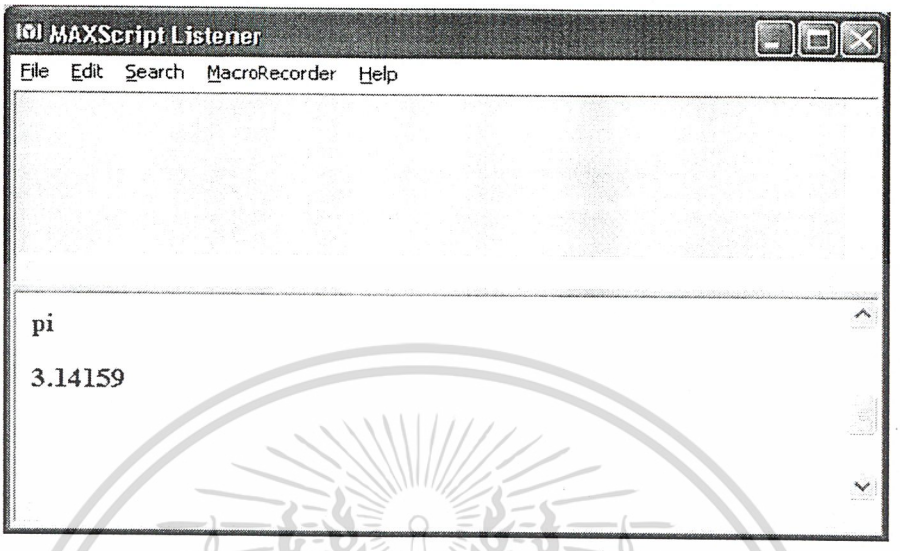



*รูปที่ 3.9 การใช้คำสั่งทางคณิตศาสตร์ ในแม็กส์สคลิป*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

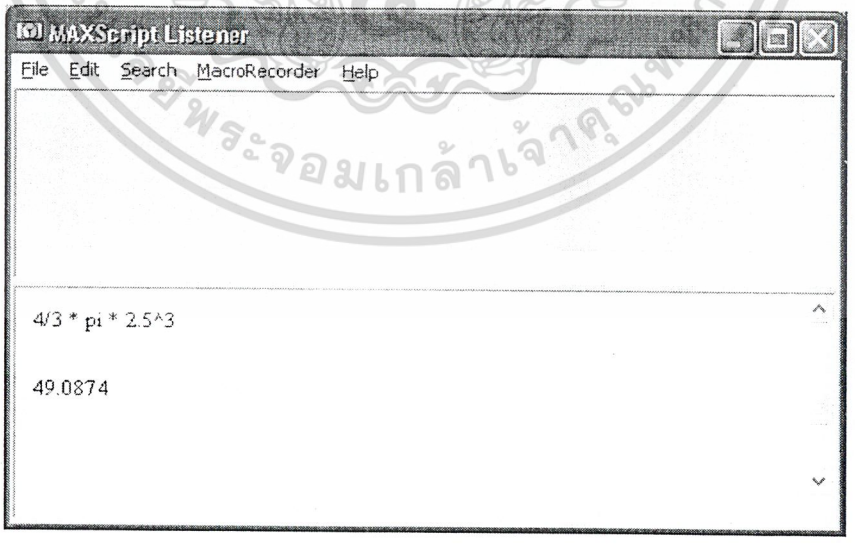
49907

สคลิปนั้นยอมรับค่าคงที่ทางคณิตศาสตร์ เช่นเมื่อเราป้อน pi




*รูปที่ 3.10 การป้อนค่าคงที่ทางคริตศาสตร์*

คุณสามารถใช้ค่าในการคำนวณด้วยการรวมหลายๆคำสั่ง สำหรับตัวอย่างถ้าคุณต้องการรู้ ค่าปริมาณของ ทรงกลม ซึ่งมีรัศมี 2.5 นิ้ว โดยการคูณค่าลูกบาศก์ ด้วย pi จากนั้นจึงคูณด้วย 4 แล้วหารด้วย 3 ดังนี้ 4/3 * pi * 2.5^3

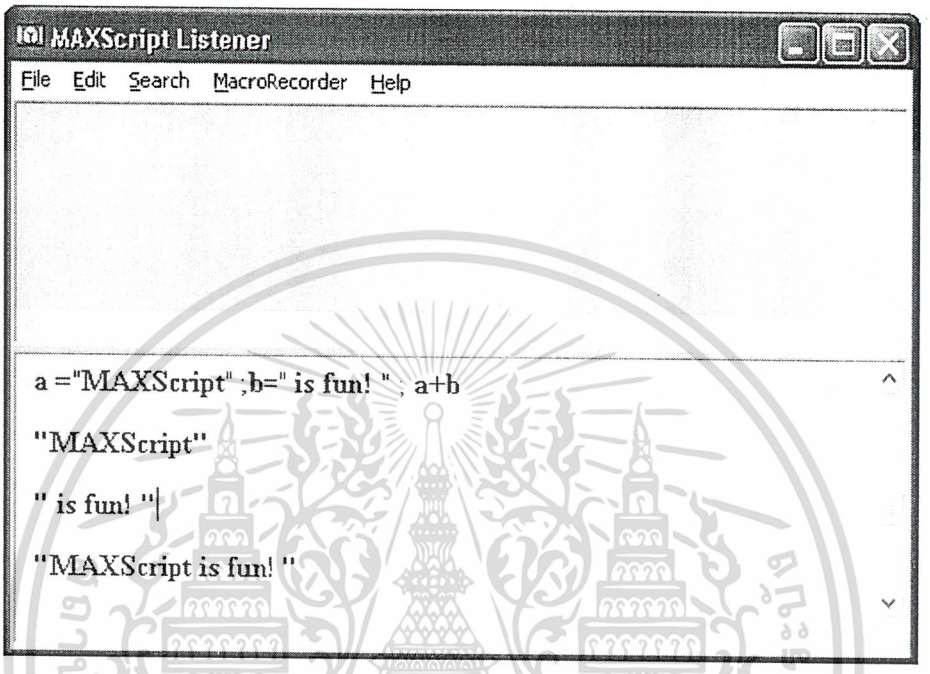



*รูปที่ 3.11 การคำนวรทางคณิตศาสตร์ด้วยหลายๆคำสั่ง*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 การทำงานกับสตริง

เราสามารถทำงานทางคณิตศาสตร์ง่ายๆกับ สตริง เช่นตัวอย่าง ถ้าคุณประกาศ a = "MAXScript " และ b = "is fun !" เมื่อเราป้อน a + b จะมีการคืนค่า "MAXScript is fun !"




*รูปที่ 3.12 การคำนวณทางคณิตศาสตร์กับ สตริง*

3.4.3 การสุ่มค่าตัวเลข

การสุ่มตัวเลขในแม็กส์สคลิป สามารถทำได้ดังนี้

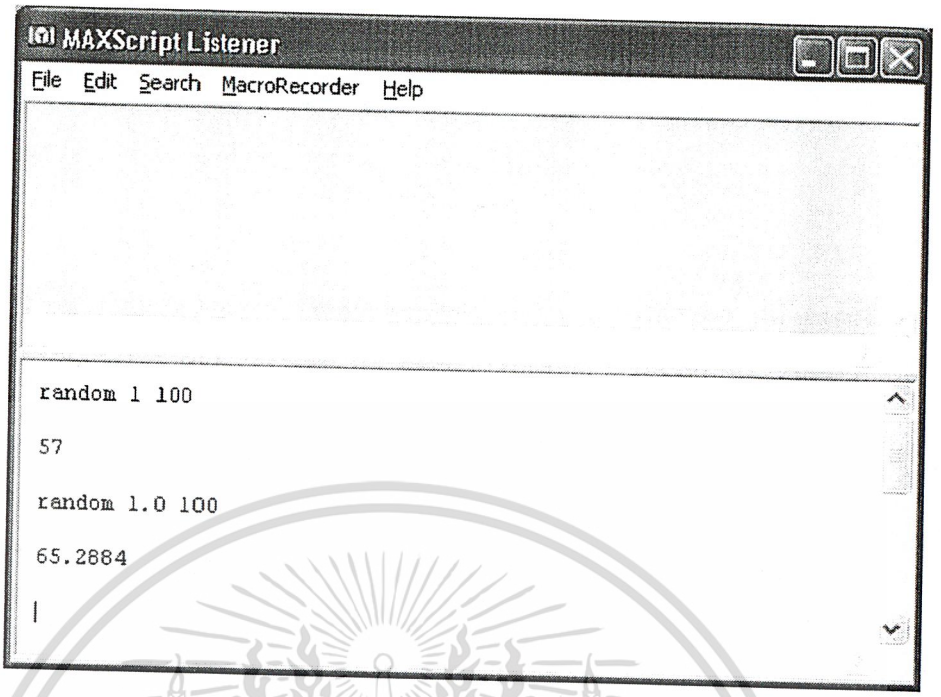
Random 1 100

ถ้าเราป้อนเป็นจุดทศนิยม

Ranmod 1.0 100

แม็กสคลิป จะมีการคืนค่าเป็นเลขทศนิยมระหว่าง 1 ถึง 100 แสดงได้ดังรูปที่ 3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




*รูปที่ 3.13 รูปแบบการสุ่มค่า*

3.4.4 การเพิ่มค่า

X = X + 1

การกำหนดการทำงานทางคณิตศาสตร์ ( + , - , * , / ) จะมีรูปแบบ ซินแท็กดังนี้

<destination> += <expr> -- บวก <expr> ไปปลายทาง

<destination> -= <expr> -ลบ <expr> จาก ปลายทาง

<destination> *= <expr> -- คูณ ปลายทางด้วย <expr>

<destination> /= <expr> -- หาร ปลายทาง ด้วย <expr>

**3.5 การเรียกใช้และทำงานกับ สคลิปไฟล์**

เราสามารถเปิดสคลิปไฟล์ขึ้นมาแก้ไขหรือประกาศสคลิปไฟล์อื่นๆให้ขึ้นมาทำงานโดยอัตโนมัติ เมื่อสคลิป นั้นเริ่มทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1 การทำงานกับสคลิปที่คุณได้ เก็บไว้แล้ว

1.รีเซ็ต สามดีสตูดิโอแม็ก

2.ไปที่แถบเครื่องมือเปิด แม็กส์สคลิปขึ้นมา ในแถบแม็กสคลิป ให้คลิกที่ ปุ่ม รันสคลิป เลือกที่ อิดิตเตอร์ จะปรากฏ ไดอะล็อกไฟล์ขึ้นมา

3.เลือก box_draw.ms ไฟล์ แล้วคลิกเปิด แม็กส์สคลิปจะทำงานตามคำสั่งที่บรรจุ อยู่ใน สคลิปไฟล์ แล้วก็จะ ทำการวาดกล่องขึ้นมาบน ฉาก

3.5.2 การเรียกใช้สคลิปอื่นๆ

ถ้าเราต้องการเขียนสคลิปแต่ละอัน ( จาตัวอย่างคือการวาดกล่องการแก้ไขกล่อง และการทำงานรูปแบบอื่นๆ ) คุณสามารถใช้ สามสคลิปอื่นที่ทำงานเกี่ยวกับกล่อง สมมุติว่าคุณมีการสร้างกล่องในสคลิป box_draw.ms , box_mod.ms , และ box_trans.ms, คุณสามารถประกาศ มันในสคลิปใหม่ ที่มีชื่อว่า box_all.ms จะมีการเขียนตามนี้ :

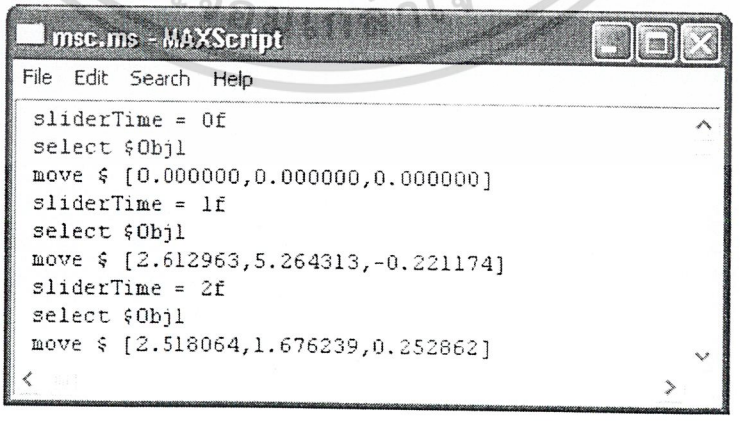
include "box_draw.ms"

include "box_mod.ms"

include "box_trans.ms"

**3.6 การทำแอนิเมชันด้วยแมกซ์สคลิป**

จากรูปที่ 3.14 แสดงถึงคำสั่งของแมกซ์สคลิปในการทำภาพแอนิเมชัน ซึ่งหลังจากรัน คำสั่งนี้แล้วจะได้ภาพแอนิเมชันด้วยกันทั้งหมด 1 ภาพจำนวน 3 เฟรม




*รูปที่ 3.14 แสดง Code MaxScript ที่ใช้ทำภาพแอนิเมชัน*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

sliderTime = 0f

คือคำสั่งที่กำหนดให้เป็นเฟรมที่ 0

select $Obj1

คือคำสั่งทีทำการเลือกวัตถุในตัวอย่างคือ Obj1

Move $ [0.000000,0.000000,0.000000]

คือคำสั่งที่ใช้ในการเลื่อนวัตถุที่เลือกในแกน X Y Z ตามลำดับในตัวอย่างยังไม่มีการเคลื่อนที่ของที่ของวัตถุ

sliderTime = 1f

คือคำสั่งที่กำหนดให้เป็นเฟรมที่ 1

select $Obj1

คือคำสั่งทีทำการเลือกวัตถุในตัวอย่างคือ Obj1

move $ [2.612963,5.264313,-0.221174]

คือคำสั่งที่ใช้ในการเลื่อนวัตถุที่เลือกไปในแนวแกน X = 2.612963, Y = 5.264313 และ Z = -0.221174

3.6.1 การวาดกล่องด้วย แม็กสคลิป

แม็กสคลิป เป็นเครื่องมือที่ช่วยในการทำงานเกี่ยวกับวัตถุ เช่น กล่อง หรือ ทรงกระบอกเราสามารถวาดกล่องใน แม็กสคลิป โดยใส่คำสั่ง BOX() ;

BOX ( )

นี่เป็นการสร้างกล่องโดยมีค่าพารามิเตอร์มาตรฐาน คือ ในการสร้างนั้นเราจะให้ค่า ออปเจกต์นั้นกับตัวแปร ซึ่งเราสามารถที่จะอ้างถึงได้มันในภายหลังโดยการใช้ชื่อนั้นและสามารถปรับปรุงมันโดยใช้ชื่อนั้น

Mybox = box ()

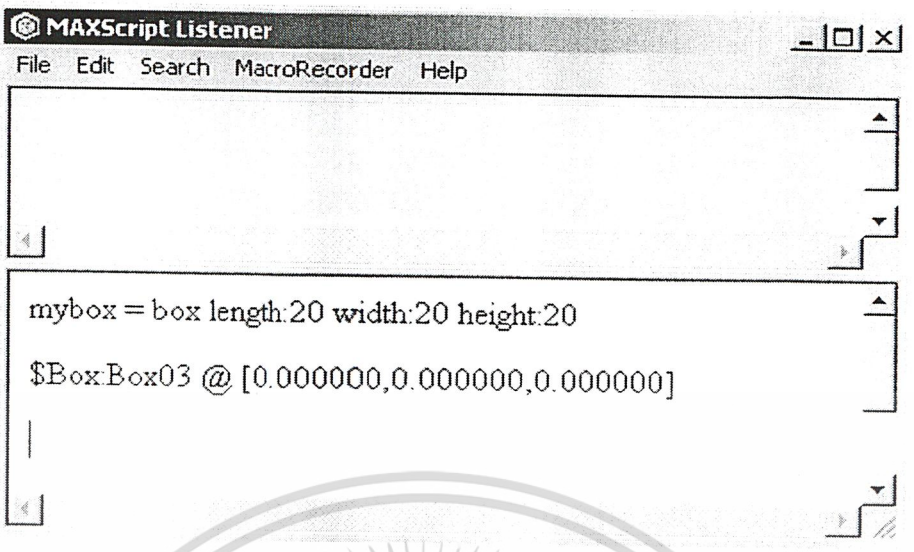
เมื่อเราได้สร้าง ออปเจกต์โดยไม่ได้กำหนดค่ารายละเอียดต่างๆนั้น เราจำเป็นที่จะต้องใช้ค่าพารามิเตอร์ว่าง " ( ) "

นี่เป็นการบอกว่าให้ใช้ค่ามาตรฐานกับวัตถุนั้น แต่ถ้าเราต้องการกำหนดรายละเอียดของวัตถุในระหว่างการสร้าง เช่น ขนาด หรือตำแหน่งที่ตั้งของวัตถุ เราก็ไม่จำเป็นที่จะต้องใส่ตัว พารามิเตอร์ว่างดังตัวอย่าง

Mybox = box length : 20 width : 20 height : 20

แม็กส์สคลิป จะมีการส่งค่า ชื่อของออปเจกต์ของกล่องและตำแน่งของกล่องกลับมาดังรูปที่ 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




*รูปที่ 3.15 การสร้างกล่องและกำหนดค่าพารามิเตอร์*

นี่เป็นตัวอย่างการสร้างกล่อง และการกำหนดค่า พารามิเตอร์ให้ 3 อย่าง คือ : ความยาว , ความกว้าง , และความสูงของกล่อง โดยที่ชื่อของพารามิเตอร์นั้นจะตามด้วย เครื่องหมาย โคล่อน และตามด้วยค่าของพารามิเตอร์นั้น เราสามารถที่จะใส่ค่าพารามิเตอร์โดยไม่จำเป็นที่จะต้องใส่เครื่องมาย เว้นวรรคระหว่างมันเลย แม็กส์สคลิป จะมีการคืนค่ากลับมาตามที่เรากำหนด:

$Box:Box01 @ [0.000000, 0.000000, 0.000000].

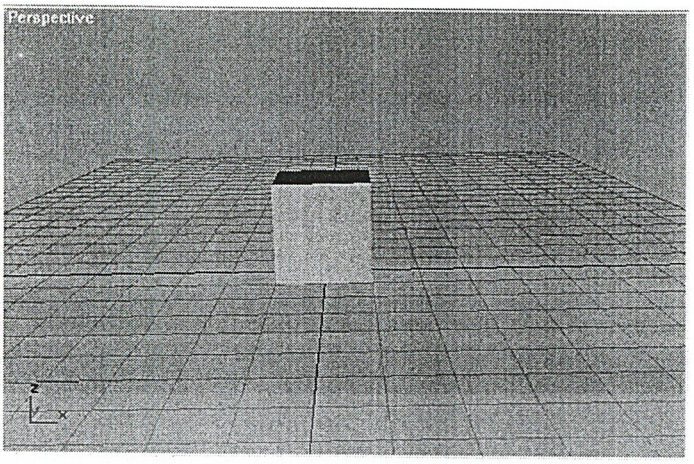
ในส่วนแรกนี้เป็นส่วนที่ถูกเรียกว่า ชื่อพาสต์ ชื่อพาสต์ใน แมกซ์สคลิป นั้นเหมือนกับชื่อพาสต์ใน วินโดวส์ ตัวอย่าง เช่น *c:\3dsmax\examples\file.max*, ซึ่งการแสดงพาสต์นั้นจะแสดงถึงโครงสร้างไดเร็กทรอรี่ของตำแหน่งที่ไฟล์อยู่อย่างไรก็ตามพาสต์ในแมกซ์สคลิป นั้น จะเป็นจุดที่อ้างไปถึงตัว ออปเจกต์ ไม่ใช่ ไฟล์ พาสต์เนม ใน แมกซ์สคลิป จะต้องขึ้นต้นด้วยเครื่องหมาย " $ " เสมอ

ส่วนที่สองที่กำหนดนั้นเป็น ชื่อของ ออปเจกต์ : Box01, ในกรณีนี้ชื่อนี้จะปรากฏในการเลือกจาก ออปเจกต์จากไดอะลอกต์ เมื่อเราใช้ เลือก โดยชื่อ ทูลบาร์ ปุ่มในสามดีเอสเม็กซ์จะไม่เป็นชื่อตัวแปรของกล่อง ซึ่งคือ mybox ที่เป็นเพียงชื่อของตัวแปรที่ชี้ไปยังตำแหน่งของออปเจกต์ Box01

ค่าข้างในนั้นเป็นค่าที่บอกถึงตำแหน่งบนแกน x , y และ z ที่เป็นจุดกึ่งกลางของกล่อง

แม็กส์สคลิป จะทำการวาดกล่องใน วิวพอรต์ , ซึ่งเป็นการแสดงใน perspective วิวพอรต์ดังรูปที่ 3.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




*รูปที่ 3.16 กล่องที่ถูกวาดโดยใช้แม็กส์สคลิป*

3.6.2 การแก้ไขค่าต่างๆของกล่อง

ในส่วนที่ผ่านมาเราได้พูดถึงการสร้างกล่องโดยการให้ค่าพามิเตอร์ต่างๆนั้นผ่านชื่อตัวแปร เพราะเราใช้ชื่อตัวแปรสำหรับกล่อง เราสามารถที่จะเข้าไปแก้ไขคุณสมบัติของกล่องได้โดยง่ายโดยทำผ่านตัวแปรชื่อ mybox ที่เป็น แฮนเดอนร์ที่กล่องของเราการเข้าถึงคุณสมบัติของ แฮนเดอนร์จะใช้เครื่องหมาย " . " แล้วตามด้วยชื่อของคุณสมบัตินั้น ตัวอย่างเช่น mybox.height ซึ่งทำให้สามารถอ่านค่า "ความสูงของกล่อง "

ออปเจกต์ใน สามดีคอสเม็กซ์ จะมีการสร้างค่าพามิเตอร์ เช่น กว้าง , ยาว , สูง ของกล่อง หรือค่ารัศมี ของวงกลม วัตถุนั้นจะมีการส่งค่า พารามิเตอร์ เช่น ขนาด การหมุน ,หรือตำแหน่ง และโดยทั่วไปคุณสมบัติจะเป็นเหมือนชื่อ ,สี , อื่นๆ การเปลี่ยนคุณสมบัติ เราสามารถตั้งค่าเป็นค่าใหม่ได้

3.6.3 การเปลี่ยนสีของกล่อง

เราสามารถเปลี่ยนค่าสีของกล่องโดยการใส่ค่า mybox.wireColor = blue และเราก็สามารถที่จะไปดูกล่องของเราจาก วิวพอรต์จะเห็นได้ว่ากล่องเป็นสีฟ้าแล้ว

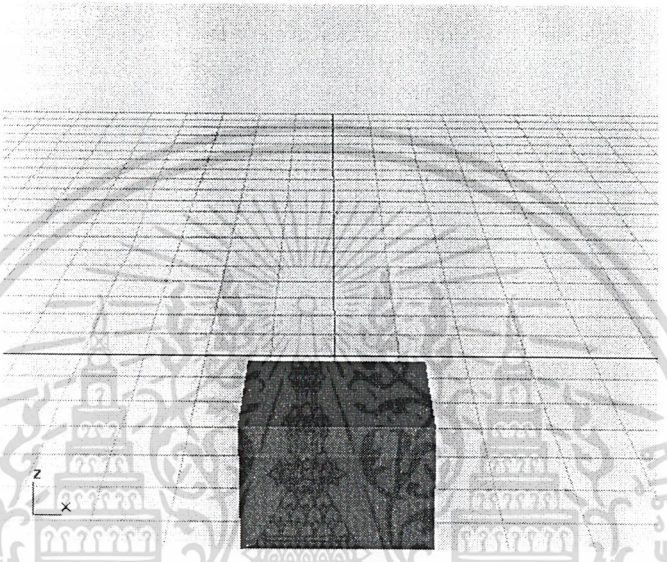
สีฟ้านั้นเป็นการอ้างถึงค่าคงที่ของสี ใน แม็กส์สคลิป และค่าของสีในค่าของ RGB คือ (0 , 0 , 255) การประกาศสีของตัวแปรอื่นๆ คือ red , green , white , black , orange , yellow และ brown แทนการใช้การอ้างถึงสีคุณสามารถใส่ค่าสีที่ต่างจากค่า RGB ดังเช่น:

```
Mybox.wrieColor = (color 255 0 255)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.4 การเปลี่ยนตำแหน่งของกล่อง

เราสามารถเปลี่ยนตำแหน่งของกล่อง โดยการเปลี่ยนที่คุณสมบัติตำแหน่งของกล่อง เราจะอ้างถึงคุณสมบัติ ของตำแหน่งจาก ตัวอย่าง กล่อง ด้วย mybox.pos ตำแหน่งถูกแสดงโดยเรียงตาม แกน X , Y , Z สำหรับตัวอย่าง [0 , -75 ,0] เป็นการนำกล่องไปข้างหน้าโดยการป้อน mybox.pos = [0 , -75 , 0] และเราสามารถดูการกระโดดไปข้างหน้าได้ใน เพอร์สเป็กทีฟ วิวส์:




*รูปที่ 3.17 การเปลี่ยนตำแหน่งของกล่อง*

3.6.5 การเปลี่ยนขนาดของกล่อง

การทำให้กล่องมีขนาดใหญ่ขึ้น เราสามารถใช้คุณสมบัติขนาดของกล่อง โดยการใช้ mybox.scale = [ 1.5 ,1.5 ,1.5] เราจะเห็นการเปลี่ยนขนาดเป็น 1.5 เท่าของขนาดเดิม คุณสมบัติขนาดนั้นต้องการค่าสำหรับแต่ละแกนคือแกน x , y ,และ z และในกรณีที่ความกว้างความยาวและความสูงของกล่อง ถ้าเราไม่ต้องการเปลี่ยนค่าขนาดของทุกแกนโดยจะเปลี่ยนเฉพาะขนาดความสูงความกว้างแต่ไม่เปลี่ยนขนาดความยาวจะได้ผลตามตัวอย่าง

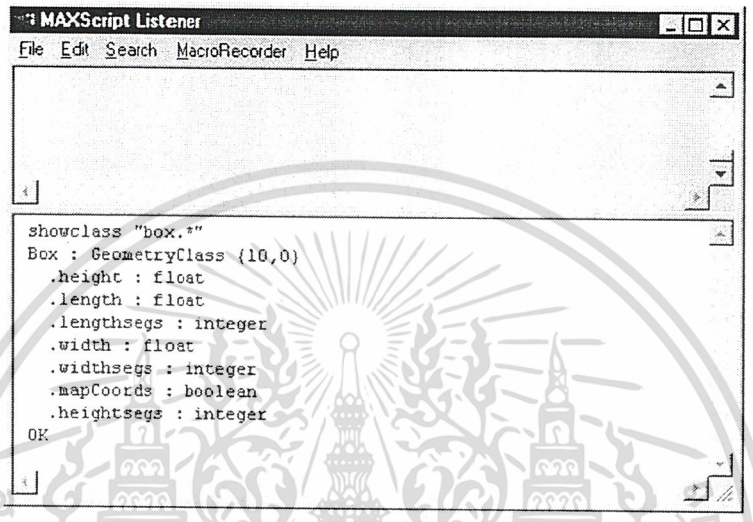
mybox.scale = [1,3,2]

เมื่อขนาดเดิมเป็น [20 , 20 , 20] ค่าความยาวจะไม่เปลี่ยนแปลงแต่ค่าความกว้างจะเปลี่ยนเป็น 60 และค่าความสูงจะเปลี่ยนเป็น 40 จากการคูณจำนวนเท่าของ ค่าที่เราใส่เข้าไป
การหาค่าพารามิเตอร์อื่นๆที่เราสามารถเปลี่ยนได้:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าพารามิเตอร์ต่างๆที่ใช้ในการสร้างกล่องทุกตัวจะถูกแสดงออกมา ถ้าเราต้องการที่จะเปลี่ยนค่าต่างๆ เช่น ค่าของ ความยาว, ความกว้าง ,ความสูงหรือการตั้งค่าการแมป จุดต่างๆใน แม็กส์สคลิป เราจำเป็นต้องรู้ซินแท็กของค่าต่างๆเหล่านั้น ในซินแท็กการตั้งค่าต่างๆนั้นในขั้นแรกเราจะใช้ showclass() โดยการป้อน showclass " box.* "

แม็กส์สคลิป จะแสดงค่าต่างๆทุกค่าและข้อมูลของพารามิเตอร์แต่ละตัว ตามที่ต้องการดังรูปที่ 3.18




*รูปที่ 3.18 คำสั่งแสดงค่าพารามิเตอร์ต่างๆของออปเจ็ก*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4

# Stereopsis

Stereopsis เป็นกระบวนการที่ทำให้เราสามารถอ้างถึงตำแหน่งของวัตถุในรูปแบบสามมิติได้โดยพิจารณาจากภาพอย่างน้อย 2 ภาพโดยภาพเหล่านั้นถ่ายในมุมมองที่ต่างกัน

ซึ่งในส่วนนี้จะกล่าวถึงปัญหาในการทำ การหาความลึกของภาพ และวิธีการที่จะทำให้ได้ระยะของวัตถุมาโดยใช้ภาพ 2 ภาพ

ดังรูป (ก) คือรูปด้านซ้ายซึ่งเป็นรูปอ้างอิงและรูป (ข) คือรูปด้านขวาซึ่งเป็นรูปที่ใช้ประกอบกับรูปด้านซ้ายเพื่อหาคาทิเซียนคอดิเนท(Cartesian coordinate) ของภาพอ้างอิงรูปที่ 4.1

(ก)

(ข)

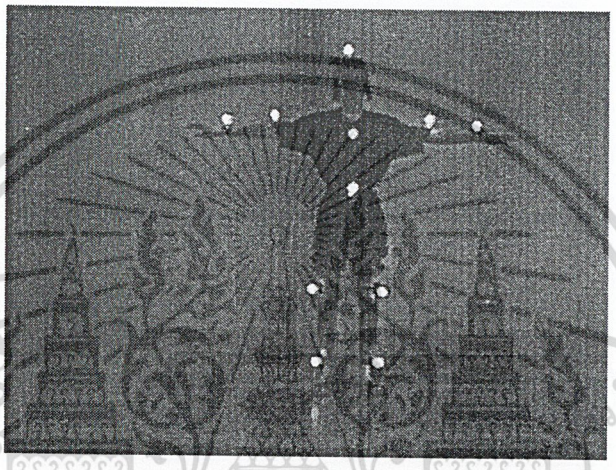
*รูปที่ 4.1 รูปภาพที่ใช้ในการหา ตำแหน่งของวัตถุ (Cartesian coordinate)*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

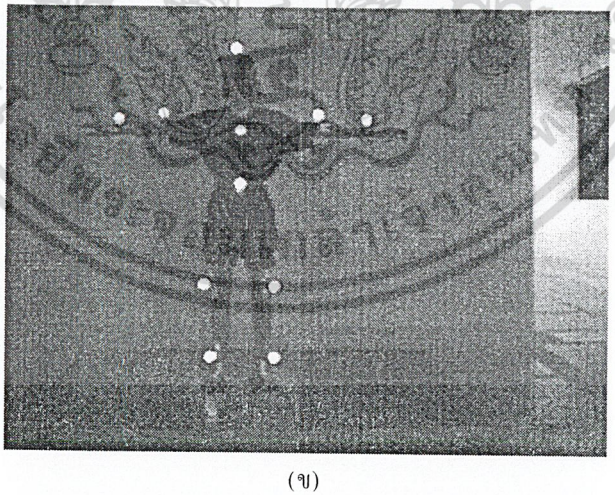
**4.1 ปัญหาที่เกิดขึ้นในการทำ Stereopsis**

ปัญหาที่ทำให้เกิดความยากในขั้นตอนการ Stereopsis มีอยู่ด้วยกัน 2 ปัญหาคือ

4.1.1 เราจะหาจุดของภาพที่ต่างกัน ที่แสดงถึงจุดของวัตถุเดียวกันในโลกจริงได้อย่างไร ตัวอย่างคือรูปที่4.2 ในความเป็นจริงเมื่อลองอ้างอิงกับตาของมนุษย์นั้นจะพบว่ามีบางพื้นที่ของภาพจากตาด้านซ้ายที่ไม่ปรากฏบนภาพของตาด้าน ขวา ในทาง กลับ กัน ก็เป็นเช่นเดียวกันคือ บางพื้นที่จากตาด้านขวาจะไม่ปรากฏบนภาพของตาด้าน ซ้าย ดังนั้นระบบ สเตอร์ริโอออปซิท ของเราต้องสามารถระบุพื้นที่เหล่านี้ให้ได้



(ก)



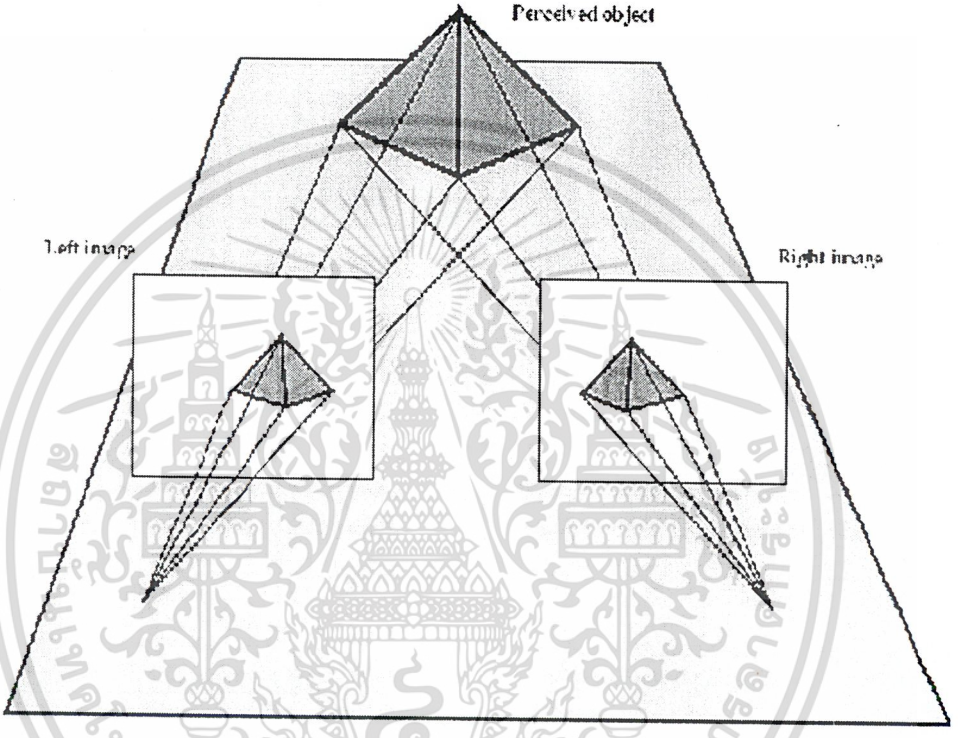
(ข)

*รูปที่ 4.2 จุดภาพบน (ก) (ข) ที่เป็นจุดเดียวกันในโลกจริง*

4.1.2 เมื่อเราทราบจุดของภาพที่แสดงถึงจุดของวัตถุเดียวกันแล้ว เราจะสามารถหา ค่าพิกัดของวัตถุเหล่านั้นได้อย่างไร วิธีการนั้นเราต้องทราบ เรขาคณิต ( geometry ) ของกล้อง คุณสมบัติของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

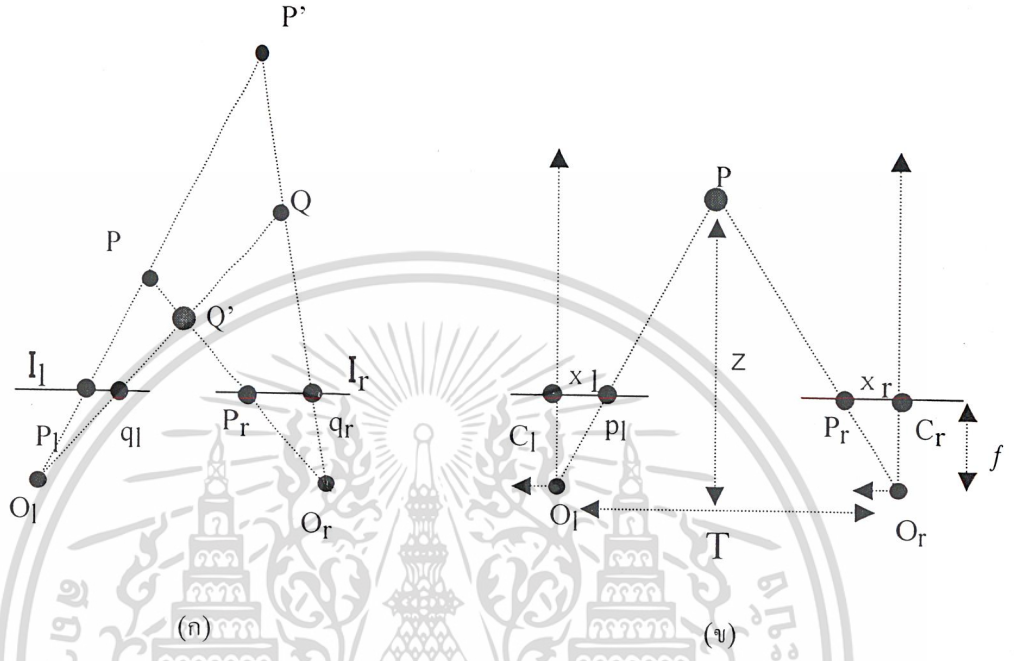
เลนส์เช่นระยะโฟกัส ( focal length ) ค่าผิดพลาดที่เกิดจากการรับภาพ เป็นต้น เมื่อเราทราบค่าเหล่านี้ก็จะทำให้การคำนวณค่าคาร์ทีเซียนโคออร์ดิเนท มีความแม่นยำยิ่งขึ้น แต่ในการใช้งานจริงค่าเหล่านี้ บางค่าอาจไม่ทราบ ดังนั้นต้องหาโดยการใช้การทดลองซึ่งจะกล่าวถึงในหัวข้อต่อไป รูปที่4.3 แสดงมุมมองที่ได้จากกล้อง 2 ตัว




*รูปที่ 4.3 แสดงมุมมองที่ได้จากกล้องสองตัว*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**4.2 ระบบ Stereopsis อย่างง่าย**




*รูปที่ 4.4 โมเดลของเรขาคณิตของจุด*

ในงานวิจัยนี้จะทำการหาค่า Stereopsis โดยมีโมเดลดังรูปที่ 4.2 โดยโมเดลดังรูป (ก),(ข) จะเป็นการมองดูระบบจากข้างบน (Top View) โดยที่ $I_l$ และ $I_r$ คือภาพที่ปรากฏทางด้านซ้ายและด้านขวาตามลำดับ $O_l$ และ $O_r$ คือจุดโฟกัสของเลนส์ทางด้านซ้ายและด้านขวาตามลำดับ $P_l$, $P_r$, $Q_l$, $Q_r$ คือ projection ของจุด P และ q บนภาพซ้ายและขวาตามลำดับ

หนทางที่จะให้ได้ตำแหน่งของ P และ Q คือการใช้ทฤษฎีสามเหลี่ยมคล้าย โดยการสร้างสามเหลี่ยมคล้ายขึ้นมานี้เราจะพบกับปัญหาข้อที่ 1 คือ เราจะเลือกจุดใดเพื่อแทนจุดทางโลกจริงที่เหมือนกัน สามารถแสดงตัวอย่างของปัญหาได้ดังนี้ กล่าวคือถ้าเราเลือก $P_l$, $P_r$ เป็นจุดของภาพซ้ายและขวาซึ่งแสดงถึงจุด P เดียวกัน เราสามารถรู้ได้ว่าจุด P อยู่ที่ใดโดยการลากรังสีที่เกิดจากจุดโฟกัสของทางด้านซ้ายพาดผ่าน $P_l$ ไปตัดกับรังสีที่เกิดจากจุดโฟกัสของทางด้านขวาลากผ่าน $P_r$ ซึ่งก็คือจุด P เมื่ออ้างอิงกับรูปที่ 5 (ก) ซึ่งเป็นจุดที่ถูกต้อง แต่ถ้าเราเลือกจุดที่แทนจุดเดียวกันในโลกจริงผิดคือเลือก $q_l$, $q_r$ จะพบว่าจุดที่ตัดกันของรังสีคือจุด Q ซึ่งจะทำให้เกิดการแปลค่ามาเป็นคาร์ทีเซียนโคออร์ดิเนท ที่ผิดพลาดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณารูปที่ 4 . 4 (ข) เราสามารถสร้างสมการเพื่อหาค่าระยะห่างของจุดของวัตถุโลกจริงในแนวตั้งฉากเมื่อสัมพันธ์กับผู้ดูได้ ดังสมการต่อไปนี้

$$\frac{T}{Z} = \frac{(T - (Xl + Xr))}{(Z - f)}$$

ดังนั้นเราสามารถหาค่า Z ได้ดังนี้

$$Z = f\frac{T}{d}$$

โดยที่

d = Xl+Xr

**4.3 การแก้ปัญหาของการหาจุดที่แสดงถึงจุดในโลกจริงเดียวกัน**

ภายใต้การแก้ปัญหาที่จะนำเสนอต่อไปนี้ ต้องอยู่บนข้อสมมุติที่ว่า

1. จุดของวัตถุนั้นต้องเห็นได้ทั้งมุมมองทั้งซ้ายและขวา
2. จุดของวัตถุจะต้องไปปรากฏที่ภาพด้านซ้ายและด้านขวาด้วยค่าความเข้มแสงที่เท่ากัน

สมมุติฐานข้างต้นจะเป็นจริงได้ก็ต่อเมื่อจุดตัดของวัตถุ ต้องอยู่ห่างจากภาพอ้างอิงมากกว่าระยะห่างของจุดโฟกัสของเลนส์ด้านซ้ายและเลนส์ด้านขวามาก ปัญหาการหาจุดที่ถูกต้องบนภาพทั้งสองภาพก็คือ ปัญหาที่เราจะตองเลือกจุดบนภาพด้านซ้าย แล้วทำการหาจุดบนภาพด้านขวา ที่แสดงถึงจุดของวัตถุเดียวกัน ดังนั้นเราสามารถสรุปปัญหาที่จะทำการแก้ออกเป็น 2 ข้อ ดังนี้

1. จุดใดในภาพด้านขวาที่ควรเลือก
2. สิ่งใดที่จะนำมาตัดสินใจว่าควรเลือกส่วนใด

ในที่นี้เราจะเสนอวิธีการแก้ปัญหาด้วยแนวคิด 2 แบบ วิธี correlation – based และ feature – based ซึ่งวิธี correlation – base จะพิจารณาโดยใช้จุดภาพทุกจุด แต่ function – based จะทำการเลือกเฉพาะบางจุดของภาพเท่านั้นเพื่อนำมาใช้ในการพิจารณา

4.3.1 วิธี Correlation – based

เราจะมีการใช้หน้าต่าง (windows) ที่มีขนาดตายตัวเพื่อทำหน้าที่ในการคัดเลือกจุดของภาพเพื่อทำการทดสอบ โดยจะมีตัววัดเพื่อบอกว่าหน้าต่างหนึ่งเหมาะสมกับอีกหน้าต่างหนึ่งหรือไม่ ซึ่งสิ่งนี้จะเป็นตัวบอกว่ามันคือพื้นที่เดียวกันในโลกจริงหรือไม่

สามารถแสดงวิธีการทำงานได้ดังนี้

รับภาพด้านซ้าย ( $I_l$ ) และขวาเข้ามา ( $I_r$ )

ให้ $P_l$ , $P_r$ เป็นพิกเซลของภาพด้านซ้ายและขวา ตามลำดับ ความกว้างของหน้าต่างคือ 2*w+1 พื้นที่ของภาพที่ต้องการหาคือ R( Pl ) ซึ่งเป็นพื้นที่ด้านขวาที่สัมพันธ์กับ Pl และ $\Psi(u,v)$ คือ ฟังก์ชั่น ( function ) ของค่าพิกเซล 2 ค่า สำหรับแต่ละ Pl = $[i,j]^T$ ของภาพด้านซ้าย :

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.สำหรับแต่ละค่า $d = [d1,d2]^T \in R(Pl)$ จะถูกคำนวณโดยใช้สมการ

$$c(d) = \sum_{k=-w}^{w} \sum_{l=-w}^{w} \psi \ (Il(i+k, j+l), Ir(i+k-d1, j+l-d2))$$

2.หน้าต่างของภาพซ้ายและขวา จะเป็นหน้าต่างที่แสดงถึงพื้นที่โลกจริงในจุดเดียวเดียวกันก็ต่อเมื่อได้ค่า c(d) มากที่สุดภายใน $R(P_l)$

ฟังค์ชั่น $\psi(u,v)$ สามารถมีตัวอย่างจองสมการดังนี้

$$\psi(u,v) = uv$$

เราจะทำการตัดสินใจเพื่อเลือกขนาดของ R ใน Algorithm โดยค่า R จะพิจารณาจากขอบเขตที่มากที่สุดที่ภาพด้านซ้ายจะไปปรากฏบนภาพด้านขวา

4.3.2 วิธี Feature – based

การหาจุดที่เหมาะสมของภาพด้านขวาจะถูกจำกัดบนลงจุดที่ถูกเลือกขึ้นมาเพื่อเป็น ลักษณะเด่นของภาพ เช่น ขอบ มุม หรือเส้นของภาพ ซึ่งในแต่ละลักษณะเด่นก็จะต้องมีการให้รายละเอียดกำกับไว้

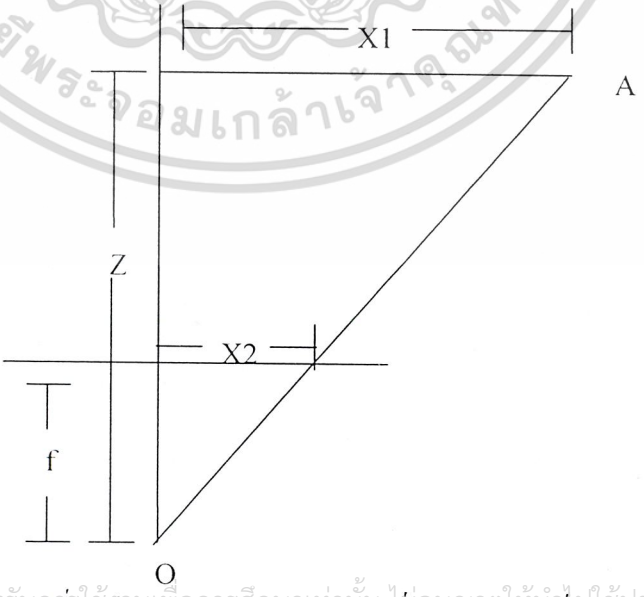
สามารถแสดง อัลกอลิทึม วิธีการทำได้ดังนี้

รับภาพด้านซ้าย ( Tl ) และขวาเข้ามา ( Tr ) พร้อมทั้งข้อมูลที่เป็นลักษณะเด่นของทั้ง 2 ภาพ

ให้ R(fl) เป็นพื้นที่ทำการหาในภาพด้านขวาซึ่งสัมพันธ์กับรายละเอียดของลักษณะเด่นในภาพด้านซ้าย fl และ d( fl , fr ) เพื่อนำมาคำนวณหา Cartesian coordinate ต่อไป

**4.4 การสร้าง Cartesian coordinate เมื่อทราบตำแหน่งที่ต้องการ**

เนื่องจากในงานวิจัยนี้เราไม่ทราบค่า focal length ของเลนส์ของกล้องใน Triclop StereoVision System ดังนั้นเราสามารถหาค่า คาทิเชียนคอดิเนท ด้วยสมการต่อไปนี้



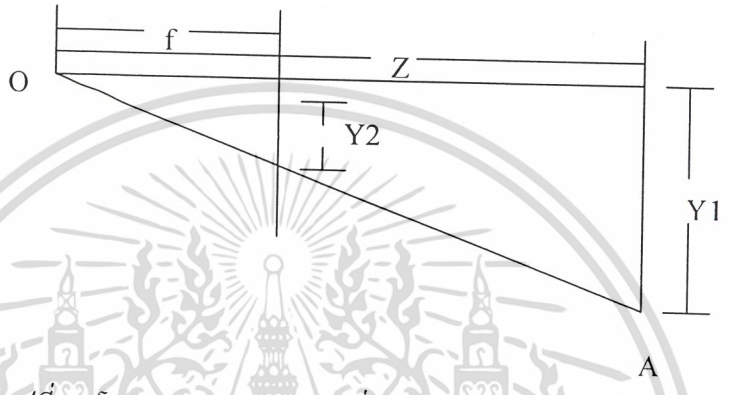

*รูปที่ 4.5 โมเดลของระบบ Sstereo เพื่อใช้ในสมการเพื่อหาค่า X*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการที่จะใช้ในการหาค่า Cartesian coordinate มีรูปแบบดังนี้

1. หาค่า X1 = $\frac{X2 * X3 * Z}{X4 * Z1}$ สมการที่ ( 1 )

โดยที่ X3 X4 Z1 เป็นค่าที่ได้จากการทดลองโดยวิธีการจะกล่าวถึงในหัวข้อ 4.3.1

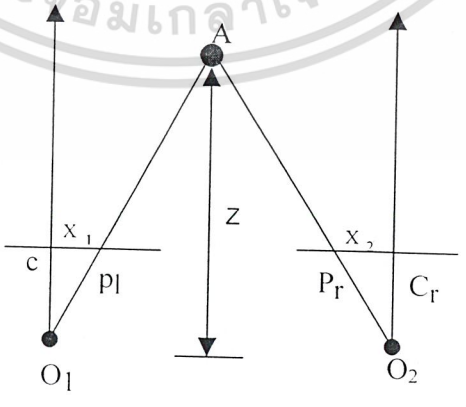



*รูปที่ 4.6 โมเดลของระบบ Stereo เพื่อใช้ในการสร้างสมการเพื่อหาค่า Y*

2. หาค่า Y1 ของจุด A เมื่อทราบ Y2 และ Z ( อ้างอิงกับรูป 4.6 ) โดยใช้ทฤษฎีสามเหลี่ยมคล้าย

$$Y1 = \frac{Y2 * Y3 * Z}{Y4 * Z1}$$ สมการที่ ( 2 )

โดยที่ Y3 Y4 เป็นค่าที่ได้จากการทดลองโดยวิธีการจะกล่าวถึงในหัวข้อ 4.3.1




*รูปที่ 4.7 โมเดลของระบบ Stereopsis เพื่อใช้อธิบายการหาค่า Z*

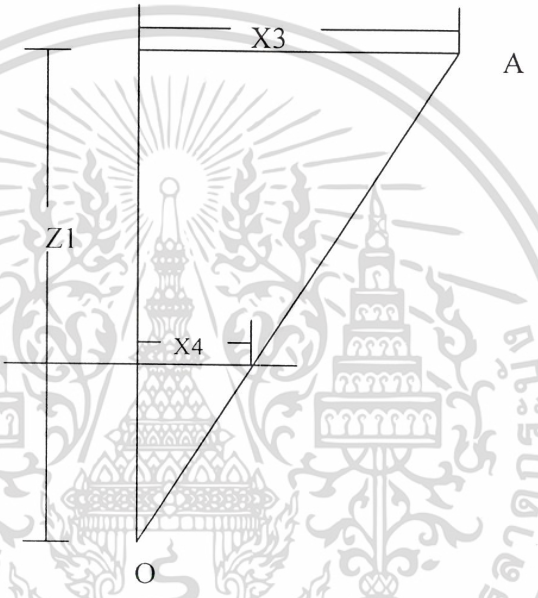
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. หาค่า Z ของจุด A เมื่อทราบ X1 และ X2 (อ้างอิงกับรูปที่ 4.7)

เราสามารถหาค่า Z ได้จากการทดลอง โดยการทดลองนี้จะเก็บผลรวมของ x1 x2 และ z ที่สัมพันธ์กันจากความสัมพันธ์สามารถระบุเป็นสมการได้ดังนี้

$$60 + 3(21-z) = x1 + x2 \qquad \text{สมการที่ (3)}$$

4.4.1 วิธีการหาค่า X , Y เพื่อใช้แทนในสมการที่ (1) และ (2) นั้นเราต้องทำการทดลองเพื่อหา X , Y โดยตั้งวัตถุแล้วทำการตั้งระยะที่สัมพันธ์กับภาพที่ใช้อ้างอิงในสมการที่ (1) และ (2) จริงๆ




*รูปที่ 4 .8 โมเดลของระบบ Stereopsis ที่ถูกสร้างขึ้นเพื่อหาค่า X3 X4 เพื่อใช้แทนในสมการที่ (1)*

จากรูปที่ 4.8 พบว่าเราสามารถทำการวัดระยะ X3 X4 และ Z1 แต่ผลที่ได้อาจมีความผิดพลาดที่สูงเนื่องจากความไม่แม่นยำของอุปกรณ์ที่ทำการวัด ซึ่งข้อผิดพลาดเหล่านี้สามารถแก้ไขได้โดยทำการสร้างระบบจำลองสภาพแวดล้อมที่มีการควบคุมมาตราวัดที่มีความละเอียดและและแม่นยำยิ่งขึ้นเช่นการใช้เวอร์เนีย แทน ไม้บรรทัด เป็นต้น

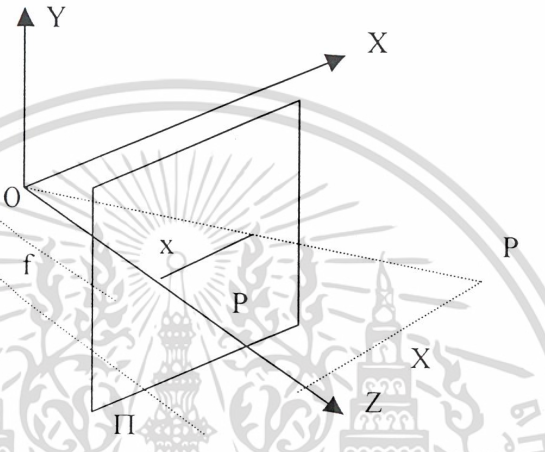
**4.5 เรขาคณิตของภาพ**

เพื่อต้องการแม็ช(match) จากจุดโลกจริงกับจุดบนภาพเพื่อใช้หาค่า Cartesian Coordinate ของจุดของโลกจริง ดังนั้นเราต้องมีความรู้ทางด้านเรขาคณิตของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.1 โมเดลพื้นฐานของเรขาคณิตของภาพ

โมเดลทางเรขาคณิตของกล้องที่รับภาพเป็นดังรูปที่ 4.9 โดยโมเดลนี้ประกอบไปด้วยระนาบของภาพ( $\Pi$ )จุดโฟกัสของภาพฉาย( **O** ) ระยะห่างระหว่างระนาบของภาพและจุดโฟกัสของภาพฉายเราจะเรียกว่าระยะโฟกัส เส้นที่ลากผ่านจุดโฟกัสของภาพฉายและตั้งฉากกับระนาบของภาพที่จุด P คือจุดของวัตถุที่พิจารณาซึ่งจะไป ปรากฏบนระนาบของภาพที่จุด P ดังนั้นเราสามารถใช้ทฤษฎีสามเหลี่ยมคล้ายเพื่อให้สามารถทราบตำแหน่งของจุดที่ไปปรากฏบนระนาบของภาพได้ โดยจากรูปที่ 4.9 เราจะได้สมการของตำแหน่งของจุดภาพดังสมการที่ (1)




*รูปที่ 4.9 เรขาคณิตภาพ*

$$x = f\frac{X}{y}$$

$$y = f\frac{Y}{Z}$$

สมการที่(1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5
# รูปแบบของไฟล์ เอวีไอ
# (Format file AVI)

รูปแบบของไฟล์ เอวีไอ นั้นจะประกอบด้วย 2 ส่วน คือ ส่วนที่เป็น ส่วนหัวของไฟล์ และส่วนที่เป็น เฟรม ของไฟล์ เอวีไอ

**5.1 ส่วนหัวของไฟล์ เอวีไอ (AVI Header Format)**

ไฟล์ เอวีไอ นั้นจะมีส่วนการอธิบายข้อมูลต่างๆของ ไฟล์อยู่ 56 ไบต์ ซึ่งจะมีการอธิบายส่วนต่างๆของไฟล์อยู่ทั้งหมด 14 ส่วนดังนี้

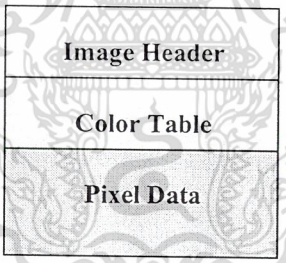
| ตำแหน่ง | ขนาด | คำอธิบาย |
|---|---|---|
| 0 | 4 | ระยะเวลาในการแสดงภาพในแต่ละเฟรม มีหน่วยเป็น ไมโครเซกก์ |
| 4 | 4 | อัตราข้อมูล ของข้อมูลแบบเอวีไอ |
| 8 | 4 | การเติมขนาดของข้อมูล ซึ่งอย่างน้อยมีค่าเท่ากับ 2048 |
| 12 | 4 | ค่า พารามิเตอร์ แฟลก์ต่างๆ |
| 16 | 4 | จำนวนของ วีดีโอเฟรม |
| 20 | 4 | จำนวนเฟรมที่นำมา แสดงเป็นตัวอย่าง |
| 24 | 4 | จำนวนของ ชุดข้อมูล (1 หรือ 2) |
| 28 | 4 | ข้อแนะนำว่าในเวลาแสดงผลควรใช้ขนาด บัฟเฟอร์ เท่าไหร่ มีหน่วยเป็น ไบต์ |
| 32 | 4 | ความกว้างของภาพ วีดีโอ มีหน่วยเป็น พิกเซล |
| 36 | 4 | ความสูงของภาพ วีดีโอ มีหน่วยเป็น พิกเซล |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ตำแหน่ง | ขนาด | คำอธิบาย |
|---|---|---|
| 40 | 4 | ช่วงเวลา อย่างน้อย คือ 30 |
| 44 | 4 | อัตราข้อมูล (อัตรา เฟรม = อัตราข้อมูล / ระยะเวลา) |
| 48 | 4 | เวลาเริ่ม, อย่างน้อย 0 |
| 52 | 4 | ขนาดของข้อมูล เอวีไอ ในหนึ่งหน่วยเวลา |

*ตารางที่ 5.1 รูปแบบส่วนหัวของไฟล์ เอวีไอ*

ในที่นี้จะขอพูดถึง รูปแบบ ของเฟรม เอวีไอ เท่านั้น เฟรมของไฟล์เอวีไอ จะถูกเก็บอยู่ในรูปของดีไอบี (Device Independent Bitmap (DIB)) โดยโครงสร้างมาตรฐานของดีไอบีประกอบไปด้วย Image Header, optional color table of RGBQUAD's และ Data แสดงได้ดังรูปที่ 5.1




*รูปที่ 5.1 แสดง รูปแบบ ของเฟรม เอวีไอ*

**5.2 ส่วนหัวของ เฟรม เอวีไอ (Frame AVI Header Format)**

เฟรม เอวีไอ นั้นจะมีส่วนการอธิบายข้อมูลต่างๆของ แต่ละเฟรม อยู่ทั้งหมด 54 ไบต์ ซึ่งจะมีการอธิบายส่วนต่างๆของเฟรมอยู่ทั้งหมด 14 ส่วนดังแสดงในตารางที่ 5.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ตำแหน่ง | ขนาด (ไบต์) | คำอธิบาย |
|---|---|---|
| 0 | 2 | เครื่องหมาย, ซึ่งต้องเป็น 4D42 hex |
| 2 | 4 | ขนาดของ ไฟล์รูปภาพ มีหน่วยเป็น ไบต์ (อาจเชื่อถือไม่ได้) |
| 6 | 2 | ถูกสงวนไว้, ต้องมีค่า เป็น 0 |
| 8 | 2 | ถูกสงวนไว้, ต้องมีค่า เป็น 0 |
| 10 | 4 | ตำแหน่งเริ่มต้นของข้อมูลภาพ มีหน่วย เป็น ไบต์ |
| 14 | 4 | ขนาดข้อมูลในส่วนหัวของไฟล์รูปภาพ มีหน่วยเป็น ไบต์, ต้องมี ขนาด 40 |
| 18 | 4 | ความกว้างของภาพ มีหน่วยเป็น พิกเซล |
| 22 | 4 | ความสูงของภาพ มีหน่วยเป็น พิกเซล |
| 26 | 2 | จำนวน ระนาบ ในภาพ, ต้องเท่ากับ 1 |
| 28 | 2 | จำนวนของ บิต ต่อ หนึ่ง พิกเซล (1, 4, 8, หรือ 24) |
| 30 | 4 | ประเภทของการบีบอัด (0=ไม่มีการบีบอัด, 1=อาร์แอลอี - 4(RLE-8), 2 = อาร์แอลอี - 4(RLE-4)) |
| 34 | 4 | ขนาดของข้อมูล ภาพ มีหน่วยเป็น ไบต์ (ทำการประกาศ เพื่อใช้ในการเติม) |
| 38 | 4 | การแสดงผลในแนวนอน มีหน่วยเป็น พิกเซล ต่อ เมตร (อาจไม่มีก็ได้) |
| 42 | 4 | การแสดงผลในแนว ตั้ง มีหน่วยเป็น พิกเซล ต่อ เมตร (อาจไม่มีก็ได้) |
| 46 | 4 | จำนวนแถวของตาราง สี , อาจมีค่าเป็น 0 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ตำแหน่ง | ขนาด (ไบต์) | คำอธิบาย |
|---|---|---|
| 50 | 4 | จำนวนสีที่จะนำมาแสดงถ้าฟิลด์นี้เป็น 0 แสดงว่าใช้สีทั้งหมด |

*ตารางที่ 5.2 รูปแบบส่วนหัวของไฟล์ เอวีไอ*

โครงสร้างคลาสส่วนหัวของดีไอบี :

```
DWORD biSize;
LONG    biWidth;
LONG    biHeight;
WORD    biPlanes;
WORD    biBitCount
DWORD biCompression;
DWORD biSizeImage;
LONG    biXPelsPerMeter;
LONG    biYPelsPerMeter;
DWORD biClrUsed;
DWORD biClrImportant;
```

โดยแต่ละฟิลด์มีรายละเอียดดังนี้

biSize

เป็นฟิลด์ที่ใช้บอกขนาดของ BITMAPINFOHEADER มีหน่วยเป็นไบต์ ขนาด 40 ไบต์

biWidth

เป็นฟิลด์บอกขนาดความกว้างของภาพมีหน่วยเป็นพิกเซล

biHeight

เป็นฟิลด์บอกขนาดความสูงของภาพมีหน่วยเป็นพิกเซล

biPlanes

ฟิลด์นี้จะถูกเซ็ตเป็น 1 เสมอ

biBitCount

เป็นฟิลด์ที่ใช้บอกว่าภาพนี้ใช้กี่บิตต่อพิกเซล [ 1, 4, 8, 16, 24, 32 ]

biCompression

- uncompressed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BI_RGB

- compressed

BI_RLE8, BI_RLE4, BI_BITFIELDS

BiSizeImage

บอกขนาดจำนวนไบต์ที่ใช้เก็บภาพ

BiXPelsPerMeter

ไม่ใช้ ให้ = 0

biYPelsPerMeter

ไม่ใช้ ให้ = 0

biClrUsed

ฟิลด์บอกจำนวนแถวของตารางสี

biClrImportant

ฟิลด์บอกจำนวนสีที่จะนำมาแสดงถ้าฟิลด์นี้เป็น 0 แสดงว่าใช้สีทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

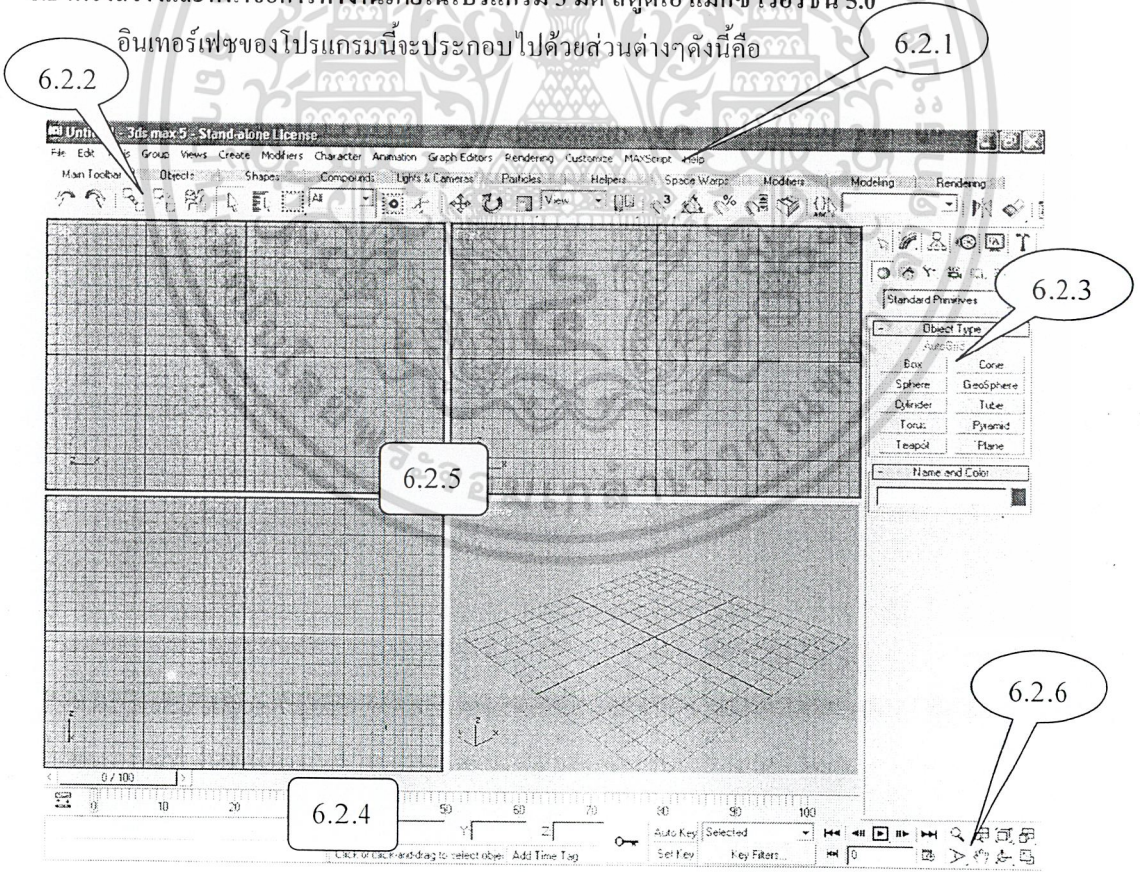
# บทที่ 6
# การสร้างโมเดล 3 มิติ

## 6.1 หลักการสร้างกราฟิก 3 มิติด้วยโปรแกรม 3 มิติ สตูดิโอ แมกซ์

ในการทำงานบน 3 มิติ สตูดิโอ แมกซ์ เวอร์ชัน 5.0 นั้นต้องทำความเข้าใจก่อนว่า กำลังติดต่อกับโลกเสมือนถูกสร้างขึ้นโดยคอมพิวเตอร์ ดังนั้นจึงต้องเข้าใจวิธีการแสดงและเก็บวัตถุต่างๆ ภายในโลกเสมือนนี้ด้วย คือ วัตถุทุกๆชิ้นมีโคออร์ดิเนตของตัวมันเอง และถูกสร้างขึ้นจากรูปทรงทางคณิตศาสตร์ หรือกล่าวได้ว่า วัตถุมีลักษณะเป็น 3 มิติ ( สร้างและอยู่เฉพาะบนซอฟท์แวร์เท่านั้น ) และทุกๆจุดภายในโลกจำลอง มี 3 โคออร์ดิเนต คือ การแสดงความสูง ความกว้างและความลึกของจุดแต่ละโคออร์ดิเนตจะเรียงตัวไปตามแกนเฉพาะ 1 แกน ( axis ) คือ ความกว้างถูกแสดงด้วยแกน X ความลึกถูกแสดงด้วยแกน Y และความสูงถูกแสดงด้วยแกน Z โดยจุดตัดของทั้ง 3 แกนนี้จะเป็นจุดเริ่มต้นของระบบแกนโคออร์ดิเนต จะมีพิกัด ( 0,0,0 ) เรานำมาระบุตำแหน่งต่างๆของวัตถุ 3 มิติใน โลกจำลอง ได้

## 6.2 โครงสร้างและหน้าจอการทำงานภายในโปรแกรม 3 มิติ สตูดิโอ แมกซ์ เวอร์ชัน 5.0

อินเทอร์เฟซของโปรแกรมนี้จะประกอบไปด้วยส่วนต่างๆดังนี้คือ




*รูปที่ 6.1 แสดงหน้าจอการทำงานภายในโปรแกรม 3 มิติ สตูดิโอ แมกซ์ เวอร์ชัน 5.0*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.1 แถบเมนู (Menu bar) คือ แถบแสดงรายการคำสั่ง มีลักษณะเป็น แถบลงมาด้านล่าง จะเก็บคำสั่งที่สำคัญต่างๆทั้งหมดตามหัวข้อที่ได้แยกไว้

6.2.2 แทปเครื่องมือ (Tab Panel) เป็นส่วนที่รวบรวมและจัดหมวดหมู่ของกลุ่มคำสั่งโดยแบ่งคำสั่งออกเป็นกลุ่มๆโดยปุ่มคำสั่งที่มีอยู่ในส่วนนี้ทั้งหมดจะมีอยู่ใน เครื่องมือควบคุม และ แถบเครื่องมือหลักด้วย

6.2.3 เครื่องมือควบคุม (Command Panel) เป็นส่วนหลักในการทำงานบน 3 มิติ สตูดิโอ แมกซ์ เวอร์ชัน 5.0 ซึ่งใน เครื่องมือควบคุม จะมีการแบ่งคำสั่งออกเป็น 6 กลุ่มดังนี้

1) Create Command Panel

2) Modifier

3) Hierachy Command Panel

4) Motion Command Panel

5) Display Command Panel

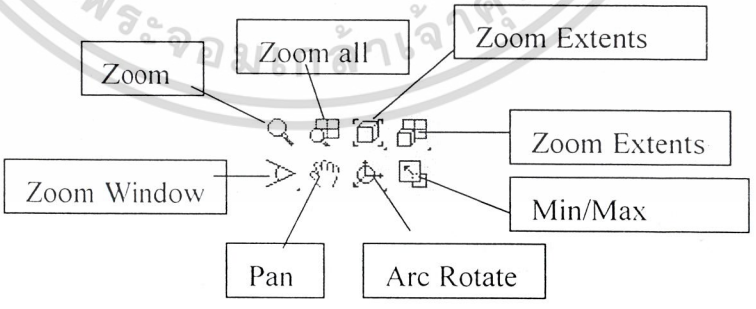
6) Utility Command Panel

6.2.4 หน้าต่างแสดงสถานะ ประกอบไปด้วยส่วนของการทำแอนิเมชัน ตำแหน่งของเคอร์เซอร์ ตำแหน่งของเฟรม องศาการหมุน สถานภาพการทำงานขณะนั้น และการ snap

6.2.5 หน้าจอย่อย (Viewport ) ในหน้าจอหลัก3 มิติ สตูดิโอ แมกซ์ เวอร์ชัน 5.0 สามารถเลือกที่จะแสดงหน้าจอย่อยได้พร้อมกันถึง 4 หน้าจอย่อยหรือแสดงเพียง 1 หน้าจอย่อยเท่านั้นก็ได้เช่นกัน หน้าจอย่อยใน 3 มิติ สตูดิโอ แมกซ์ เวอร์ชัน 5.0 ได้แก่ Top Font Right Perspective และ camera ( ถ้ามี ) และคลิกเปลี่ยนให้แสดงหน้าจอย่อยใดๆ ได้โดยกดปุ่มตัวอักษรบนคีย์บอร์ด เช่น Top กด T, Bottom กด B,, Right กด R, Front กด F, Back กด K, Perspective กด P, Camera กด C และ Spotlight กด S หรือคลิกเมาส์ขวาที่มุมบนซ้ายของหน้าจอย่อย ใดก็ได้เพื่อแสดงเมนู และสามารถเลือกหน้าจอย่อยได้

6.2.6 ส่วนควบคุมหน้าจอย่อย

ประกอบไปด้วยปุ่มคำสั่งควบคุมหน้าจอย่อยดังรูป




*รูปที่ 6.2 แสดงส่วนควบคุมหน้าจอย่อย*

Zoom เมื่อคำสั่งย่อ – ขยายภาพบนหน้าจอย่อยที่ทำงานอยู่ คลิกเมาส์ลากขึ้นเป็นการขยายภาพ ( Zoom – in ) ลากลงคือย่อภาพ ( Zoom – out )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| Zoom all | คำสั่งนี้ เหมือนคำสั่งที่แล้ว แต่มีผลกับทุกหน้าจอย่อย |
| Zoom extents | เป็นของคำสั่งย่อ – ขยายภาพ แต่จะทำจนกว่าจะมองเห็นวัตถุทั้งหมดในหน้าจอย่อย และมีคำสั่ง Zoom Extents Selected เป็น flyout ขึ้นมาให้เลือก ( ให้สังเกตว่าหากที่มุมล่าง – ขวาของปุ่มใดมีรูปสามเหลี่ยมเล็กๆ อยู่แสดงว่ามีปุ่ม flyout สำหรับเลือกคำสั่งเพิ่มเติม ) |
| Zoom extents all | คำสั่งนี้เหมือนกับคำสั่งที่แล้วแต่มีผลกับทุกหน้าจอย่อยในเวลาเดียวกัน และมีคำสั่ง Zoom Extents All Selected เป็น flyout สำหรับเลือกเฉพาะวัตถุที่ต้องการ |
| Zoom Window | คำสั่งนี้คล้ายกับคำสั่ง Zoom แต่จะเป็นการปรับค่าพื้นที่ในการมองเห็นของหน้าจอย่อยเท่านั้น ซึ่งถ้าใช้คำสั่งบน Planar Viewport เช่น Top Right หรือ Front จะถูกแทนที่ด้วย Region Zoom หรือถ้าใช้คำสั่งบน Perspective หน้าจอย่อย จะถูกแทนที่ด้วย Field of View |
| Pan | คำสั่งเลื่อน หน้าจอย่อย โดยไม่มีผลต่อการย่อ – ขยายภาพ |
| Arc Rotate | คำสั่งนี้ใช้หมุน หน้าจอย่อย ไปรอบๆ โดยใช้จุดศูนย์กลางของ หน้าจอย่อย เป็นแกนหมุนการใช้งานเมื่อคลิกเลือกคำสั่งนี้ จะปรากฏไอคอนสีเขียวบนหน้าจอย่อย คลิกเมาส์ค้างไว้แล้วลากเมาส์ไปรอบๆทำให้ หน้าจอย่อย เปลี่ยนไปตามเมาส์ และมี Arc Rotate Selected เป็น flyout สำหรับเลือกเฉพาะวัตถุที่ต้องการเป็นแกนหมุน และ Arc Rotate SubObject เป็น flyout สำหรับเลือก meterial ที่กำหนดลงบนวัตถุหรือวัตถุย่อย ( SubObject ) เป็นแกนหมุน |
| Min / Max Toggle | คำสั่งนี้ใช้ในการขยายภาพบนจอภาพย่อยที่ทำงานอยู่ให้แสดงเต็มจอภาพ เมื่อเราคลิกปุ่มนี้ซ้ำจอภาพที่ขยายเต็มจอภาพทำงานจะกลับสู่สภาพเดิม |

**6.3 การเชื่อมต่อ (Merge) ไฟล์**

การโหลดไฟล์และรวมเข้ากับไฟล์ที่แสดงอยู่ หรือเลือกรวมเฉพาะวัตถุได้เช่นกัน โดยเลือก file ( Menu Pull-Down ) > Merge และ Browse ไฟล์ที่ต้องการ แล้วเลือกวัตถุที่ต้องการจากรายการ

**6.4 การ อิมพอร์ต / เอ็กซ์พอร์ตไฟล์**

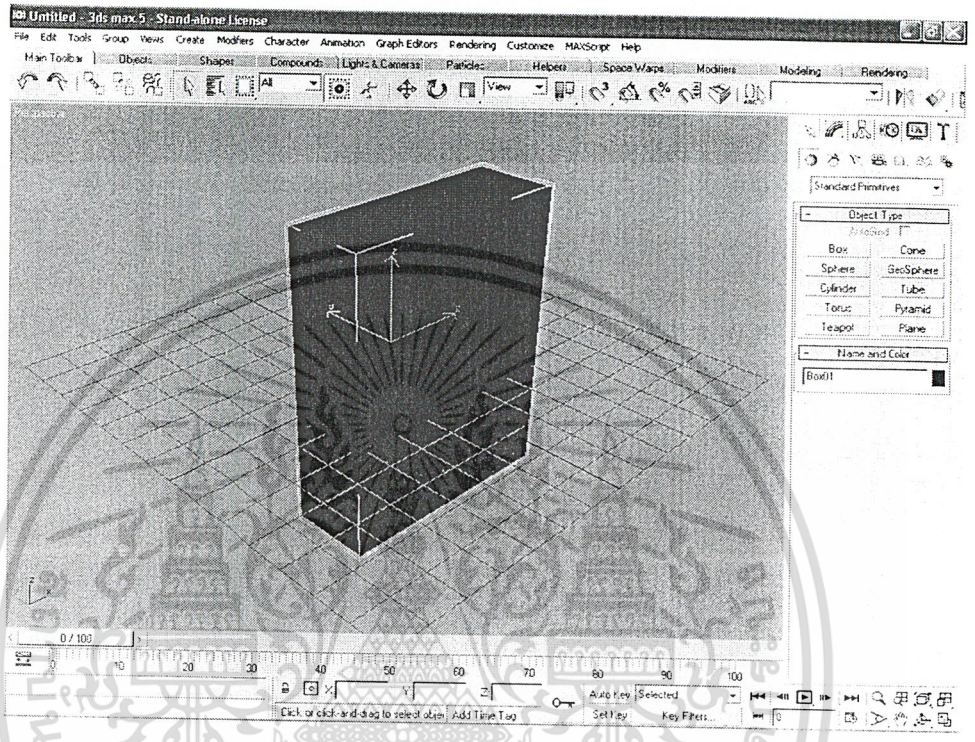
สามารถโหลดไฟล์ในรูปแบบอื่นได้ ซึ่งใน โปรแกรม3 มิติ สตูดิโอ แมกซ์ เวอร์ชัน 5.0 สามารถ import และรองรับไฟล์ประเภท 3 D Studio Mesh ( 3 มิติS ,PRJ ), ASE, DXF, STL, DWG, AI, IGE, SHP และ WRL และ เอ็กซ์พอร์ต เป็นไฟล์ประเภท เดียวกับที่สามารถ อิมพอร์ต เข้ามาได้ ซึ่งถ้าเราติดตั้ง ปลั๊กอินของมอร์ฟิทเข้าไปก็จะ เอ็กซ์พอร์ต เป็น WLD ได้เพิ่มอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**6.5 การสร้างวัตถุ 3 มิติรูปทรงพื้นฐาน**

การสร้างวัตถุ 3 มิติในโปรแกรม 3 มิติ สตูดิโอ แมกซ์ เวอร์ชัน 5.0 มีวิธีการสร้างได้ 2 วิธี โดยจะยกตัวอย่างประกอบเป็นการสร้างวัตถุรูปกล่องสี่เหลี่ยม

6.5.1 สร้างวัตถุโดยการใช้เมาส์ในจอภาพย่อย

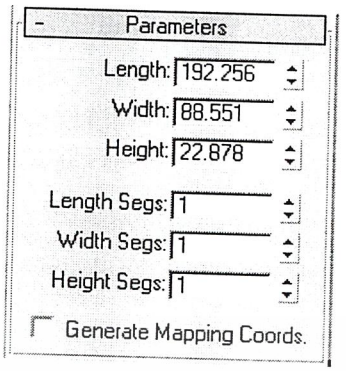



*รูปที่ 6.3 แสดงสร้างวัตถุโดยการใช้เมาส์ในจอภาพย่อย*

- คลิกเมาส์ที่ปุ่ม Box
- เลื่อนเคอร์เซอร์มาที่จอภาพย่อยมุมมอง Top แล้วคลิกให้หน้าจอย่อยนี้ทำงาน
- คลิกเมาส์ แล้วกดค้างไว้ ลากเคอร์เซอร์ไปยังตำแหน่งที่ต้องการ
- ปล่อยปุ่มเมาส์ แล้วเลื่อนเมาส์ขึ้นไปจนกล่องมีความสูงตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถสังเกตค่าต่างๆของกล่องได้ในแถบคำสั่ง Rollout หัวข้อ Parameters



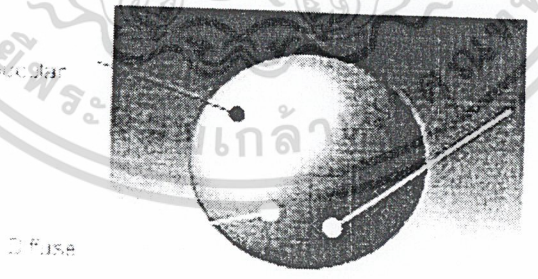

*รูปที่ 6.4 แสดงแถบคำสั่ง Rollout หัวข้อ Parameters*

6.5.2 สร้างกล่องโดยการกำหนดค่าจากคีย์บอร์ด

- กำหนดจุดวางกล่องโดยกำหนดตัวเลข X , Y และ Z ในแถบคำสั่ง Rollout หัวข้อ Keyboard Entry
- กำหนดค่าความกว้าง ความลึก และความสูงของกล่องที่ต้องการสร้างใน Keyboard Entry
- คลิกปุ่ม Create วัตถุรูปกล่องสี่เหลี่ยมจะปรากฏขึ้น ณ ตำแหน่งที่เราได้กำหนดไว้

**6.6 การกำหนดพื้นที่ผิววัตถุ**

โดยปกติพื้นผิวของวัตถุ 3 มิติ โดยพื้นฐาน จะแบ่งออกเป็น 3 ส่วนดังตามที่แสดงในรูป



*รูปที่ 6.5 แสดงคุณสมบัติเฉดสีของวัตถุ*

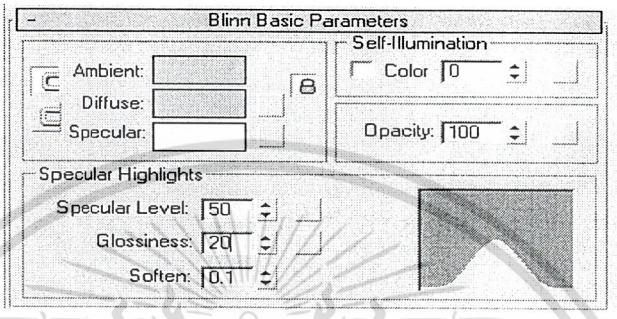
Specular คือ บริเวณพื้นผิววัตถุส่วนที่รับแสงในแนวตรงด้านหน้าสุดซึ่งบริเวณ specular นี้จะสะท้อนแสงจ้าที่สุดในพื้นผิวของวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Diffuse คือ พื้นผิวของสีพื้นฐานของวัตถุที่ถูกแสง และ ไล่เฉดของสีตามปริมาณแสงที่กระทบวัตถุ

Ambient คือ บริเวณด้านเงามืดของวัตถุที่ไม่ถูกแสง หรืออยู่ด้านตรงข้ามกับแหล่งกำเนิดแสง

นอกจากนั้นยังต้องกำหนดคุณสมบัติทางกายภาพที่มีต่อแสงของวัตถุ เช่น ความเงา ความมันวาว ความโปร่งใสของวัตถุ เป็นต้น




*รูปที่ 6.6 แสดงการกำหนดคุณสมบัติ*

เนื่องจากมีวัตถุ 3 มิติบางชิ้นในโครงงานนี้เป็นกระจก และ วัตถุที่สามารถสะท้อนแสง การกำหนดวัตถุจะช่วยให้วัตถุ 3 มิติที่สร้างขึ้นมีความเหมือนจริงมากยิ่งขึ้น ยกตัวอย่างเช่น กระจกเป็นวัตถุที่มีความโปร่งใส และ สามารถสะท้อนแสงได้

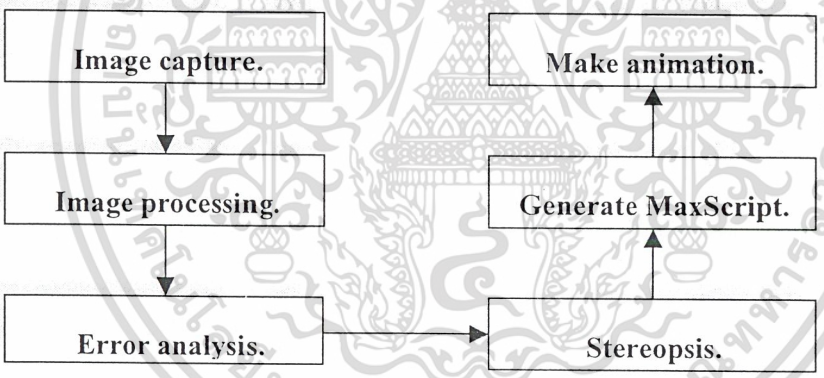
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 7
# การออกแบบโปรแกรมจับรูปแบบการเคลื่อนไหว

**7.1 ภาพรวมของโครงงาน**

โปรแกรมจับรูปแบบการเคลื่อนไหว (Motion capture) เป็นโปรแกรมที่ใช้ในการจับการเคลื่อนไหวของสิ่งมีชีวิตแล้วนำไปแมช์กับตัวการ์ตูนสามมิติให้ตัวการ์ตูนนั้นมีการเคลื่อนไหวเหมือนกับการเคลื่อนไหวของสิ่งมีชีวิตต้นแบบที่เราทำการบันทึกมา โดยผลลัพธ์จะอยู่ในรูปของกราฟิก 3 มิติ ซึ่งโปรแกรมจับรูปแบบการเคลื่อนไหวนี้สามารถนำไปใช้ประโยชน์ได้อย่างกว้างขวางเช่นนำไปใช้ช่วยในการทำภาพยนตร์ที่มีตัวละครถูกสร้างขึ้นจากโปรแกรม 3 มิติเพื่อทำให้ตัวละครนั้นมีการเคลื่อนไหวที่สมจริงหรือนำไปใช้ในวงการกีฬาเพื่อใช้ศึกษาทฤษฎีการเคลื่อนไหวของข้อต่อต่างๆเพื่อเพิ่มประสิทธิภาพการเคลื่อนไหวให้กับนักกีฬาหรือนำไปใช้ในวงการแพทย์หรือนำไปช่วยในการสร้างเกมให้ตัวละครในเกมมีการเคลื่อนไหวที่สมจริงมากขึ้นเพื่อให้ผู้เล่นมีความสนุกในการเล่นเกมมากขึ้น

ในการออกแบบและการเขียนโปรแกรมจับรูปแบบการเคลื่อนไหวนี้ จะมีรูปแบบขั้นตอนในการทำงานดังรูปที่ 7.1




*รูปที่7.1 แสดงภาพโดยรวมของระบบ*

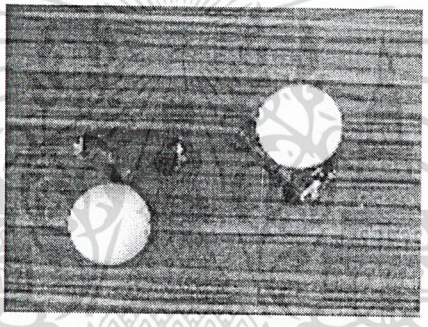
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**7.2 การออกแบบและการเขียนโปรแกรม**

การออกแบบโปรแกรมจับรูปแบบการเคลื่อนไหวนั้นเราได้แบ่งออกเป็นส่วนที่สำคัญๆทั้งหมด 6 ส่วนด้วยกันดังที่ได้แสดงไว้แล้วในรูปที่ 7.1 ซึ่งแต่ละส่วนจะมีการทำงานที่แตกต่างกันไป โดยเราจะทำการอธิบายแต่ละส่วนดังนี้

7.2.1 การบันทึกภาพเคลื่อนไหว (Image capture)

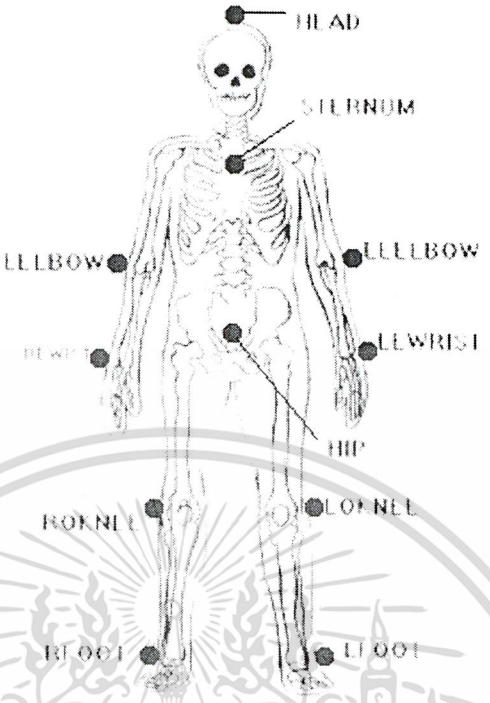
ส่วนของการบันทึกภาพเคลื่อนไหว นี้จะเป็นในส่วนของอุปกรณ์ คือการใช้กล้องในการจับภาพการเคลื่อนไหวของสิ่งมีชีวิตโดยในส่วนนี้เราได้ใช้กล้องเพียงแค่ 2 ตัวในการบันทึก ส่วนของอุปกรณ์มาคร์เกอร์นั้นจากการทดลองเพื่อหาอุปกรณ์สะท้อนแสงที่เหมาะสมที่สุดมาใช้ในโครงงานนี้ ในที่สุดเราได้เลือกใช้ลูกปิงปองที่ทำการติดหลอดไฟฉายขนาด 3 โวลต์ ไว้ภายใน ซึ่งแสดงไว้ดังรูปที่ 7.2



*รูปที่ 7.2 แสดงอุปกรณ์มาคร์เกอร์ที่ใช้ติดตามส่วนต่างๆของร่างกาย*

ส่วนของการติดอุปกรณ์มาคร์เกอร์ตามส่วนต่างๆของร่างกายนั้น การที่จะติดตรงส่วนไหนบ้างนั้นขึ้นอยู่กับว่าเราต้องการตรวจจับตรงส่วนไหนบ้าง จากโครงงานนี้เราจะทำการติดมาคร์เกอร์ตรงส่วนสำคัญหลักๆเท่านั้น ซึ่งในการจับการเคลื่อนไหวนี้จะมีขีดจำกัดการเคลื่อนไหวอยู่บ้างเพราะเราใช้กล้องเพียงแค่ 2 ตัวเท่านั้น ดังนั้นการเคลื่อนไหวในท่าทางที่ทำให้มาคร์เกอร์ถูกบังหรือซ้อนทับกันนั้นจะทำให้ไม่สามารถตรวจจับมาคร์เกอร์ตัวที่ถูกบังนั้นได้ ทำให้ไม่รู้ว่าขณะนั้นมาคร์เกอร์ที่ถูกบังนั้นอยู่ ณ ตำแหน่งไหนของภาพ และข้อจำกัดอีกข้อหนึ่งคือมันไม่สามารถตรวจจับการหมุนของรางกายส่วนต่างๆได้เพราะเนื่องมาจากการใช้กล้องเพียงแค่ 2 ตัว รูปที่ 7.3 แสดงการติดอุปกรณ์มาคร์เกอร์ตามส่วนต่างๆของร่างกายที่ได้ทำในการทดลองนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




*รูปที่ 7.3 แสดงตำแหน่งที่เราทำการติดอุปกรณ์มาคร์เกอร์*

ส่วนของไฟล์ที่ได้จากการบันทึกภาพเคลื่อนไหวนี้จะอยู่ในรูปของฟอร์แม็ตไฟล์วีดีโอเอวีไอ ซึ่งจะได้ไฟล์อยู่ 2 ไฟล์คือไฟล์จากกล้องทั้งสองตัวคือกล้องทางด้านซ้ายและกล้องทางด้านขวา ในโครงงานนี้เราจะใช้ไฟล์วีดีโอที่มีขนาด 320 x 240 พิกเซล และ ความละเอียดของสีจะใช้ขนาด 24 บิต ส่วนอัตราเฟรมเรทอยู่ที่ 25 เฟรมต่อวินาที ส่วนของอัตราเฟรมเรทนี้ถือว่าต่ำมากซึ่งขึ้นอยู่กับชนิดของกล้องที่เราใช้ด้วย ควรจะเลือกใช้กล้องที่มีอัตราเฟรมเรทที่สูงๆ ยิ่งอัตราเฟรมเรทยิ่งสูงยิ่งดี

ส่วนของสถานที่ที่จะใช้นั้นควรจะใช้สถานที่ ที่ไม่มีสิ่งที่จะมารบกวนในการบันทึกภาพ เช่นแสงไฟต่างๆ และสถานที่นั้นควรจะต้องมืดๆซะหน่อยหรือไม่สภาพแวดล้อมควรจะเป็นสีดำเพราะว่าโครงงานของเรานั้นใช้ตรวจจับแสงสีขาวที่ได้จากมาคร์เกอร์

7.2.2 อิมเมจโปร์เซ็สซิ่ง (Image processing)

ในส่วนนี้จะทำหน้าที่ในการนำไฟล์วีดีโอที่ได้มาประมวณผลเพื่อทำการหาตำแหน่งของมาคร์เกอร์ที่ปรากฏในภาพเพื่อนำตำแหน่งที่ได้ส่งไปให้ส่วนอื่นทำงานต่อไป โดยในส่วนของอิมเมจโปร์เซ็สซิ่งจะมีขั้นตอนการทำงานหลักๆอยู่ 4 ขั้นตอนดังนี้

- ส่วนของการจักการกับไฟล์วีดีโอเอวีไอ
- ส่วนของการหาตำแหน่งของมาคร์เกอร์
- ส่วนของการจัดกลุ่มตำแหน่งพิกเซลของมาคร์เกอร์

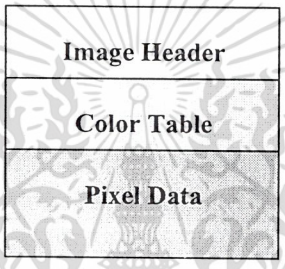
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนของการจัดลำดับมาคร์เกอร์

โดยแต่ละส่วนจะมีการทำงานเรียงตามลำดับกันไป และเราจะทำการอธิบายการทำงานของแต่ละส่วนโดยละเอียดในหัวข้อต่อไป

7.2.2.1 ส่วนของการจัดการกับไฟล์วีดีโอเอวีไอ

การทำงานในส่วนนี้คือการที่จะนำไฟล์วีดีโอ ที่ได้จากกล้องทั้งสองตัวมาทำให้อยู่ในรูปของดีไอบี(Device Independence bitmap (DIB) ) ซึ่งฟอร์แมตไฟล์วีดีโอจะมีส่วนของดีไอบี อยู่ด้วย โดย ดีไอบี นี้จะอยู่ในส่วนของเฟรมของไฟล์วีดีโอ ดีไอบีจะเป็นส่วนในการเก็บข้อมูลของภาพในแต่ละเฟรม โดยสามารถไปดูฟอร์แมตของดีไอบีแสดงได้ดังภาพที่ 7.4 เราจะทำการเขียนโปรแกรมเพื่อใช้ในการเข้าไปดึงข้อมูลในดีไอบีเพื่อที่ได้ข้อมูลของภาพในเฟรมนั้นๆออกมา เพื่อที่จะนำข้อมูลนั้นส่งไปให้ในส่วนต่อไปทำงาน




*รูปที่ 7.4 แสดง Format ของเฟรม AVI*

7.2.2.2 ส่วนของการหาตำแหน่งของมาคร์เกอร์

ในส่วนนี้จะทำการรับข้อมูลดีไอบีที่ถูกส่งมาเพื่อนำไปหาตำแหน่งของมาคร์เกอร์ โดยวิธีที่ใช้ในการหาตำแหน่งของมาคร์เกอร์นั้น เราจะเข้าไปสแกนหาทุกตำแหน่งของพิกเซลในภาพที่มีค่าความเข็มแสงตรงตามที่เราต้องการ และทำการเก็บตำแหน่งทุกพิกเซลที่สแกนหาได้ไว้และทำการส่งตำแหน่งที่หาได้นี้ไปให้ส่วนต่อไปทำงาน โดยมันจะมีการทำงานอย่างนี้ทุกครั้งไปเรื่อยๆในทุกๆเฟรม

7.2.2.3 ส่วนของการจัดกลุ่มตำแหน่งพิกเซลของมาคร์เกอร์

ส่วนนี้จะนำข้อมูลที่หาได้จากส่วนที่แล้วมาทำการหากลุ่มของพิกเซลที่คาดว่าจะเป็นกลุ่มของมาคร์เกอร์ตัวเดียวกันมาทำการจัดเป็นกลุ่มๆของมาคร์เกอร์แต่ละตัว แล้วเราจะทำการเช็คว่าพอแบ่งเป็นกลุ่มแล้ว ได้จำนวนของกลุ่มเท่ากับจำนวนของมาคร์เกอร์หรือไม่ถ้าไม่ก็จะกลับไปทำงานในขั้นตอนที่ 2 ใหม่มันก็จะทำการวนลูปทำงานแบบนี้ไปเรื่อยๆไปจนถึงค่าๆหนึ่งถ้ายังไม่ได้จำนวนกลุ่มเท่ากับจำนวนของมาคร์เกอร์ก็จะออกจากลูปแล้วแจ้งข้อผิดพลาดให้ผู้ใช้ทราบ เมื่อเราได้จำนวนกลุ่มพิกเซลเท่ากับจำนวนมาคร์เกอร์แล้ว เราก็จะส่งข้อมูลที่ได้นี้ไปให้ส่วนสุดท้ายทำงานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2.2.4 ส่วนของการจัดลำดับมาคร์เกอร์

ส่วนนี้จะนำข้อมูลที่ได้มาทำการหาจุดศูนย์กลางของมาคร์เกอร์แต่ละตัว แล้วทำการนำตำแหน่งศูนย์กลางที่ได้ ไปทำการเปรียบเทียบกับตำแหน่งมาคร์เกอร์ของเฟรมที่แล้วเพื่อทำการตรวจเช็คดูว่าตำแหน่งมาคร์เกอร์ในเฟรมปัจจุบันนั้นเป็นตำแหน่งมาคร์เกอร์จุดไหนของร่างกาย เพื่อที่จะได้นำไปประมวณผลได้ถูกต้อง

7.2.3 การวิเคราะห์ค่าความผิดพลาด (Error analysis)

การวิเคราะห์ค่าความผิดพลาด คือการที่เราได้ตำแหน่งพิกเซล X และ Y ของจำนวนเฟรมที่เรากำหนด ในโครงงานนี้เราใช้ 10 เฟรมต่อการปรับปรุงค่าที่ผิดพลาดในระหว่าง 10 เฟรมนี้เพื่อนำไปกำหนดเป็นคีย์เฟรม (Key frame) ซึ่งในกระบวนการวิเคราะห์และทำการปรับปรุงนี้เราจะทำการหาอัตราเฉลี่ยในการเคลื่อนที่ในแนวแกน X และ Y เพื่อเป็นการลดค่าผิดพลาดที่เกิดขึ้นจากการบันทึกภาพ ดังนั้นอย่างที่เคยกล่าวไว้แล้วว่ายิ่งอัตราเฟรมเรทของกล้องที่ใช้ยิ่งสูงเท่าไหร่ก็ยิ่งดีเพราะจะทำให้มีข้อมูลที่นำมาวิเคราะห์หาค่าผิดพลาดมีมากขึ้นในระยะเวลาที่สั้นลง ทำให้เราสามารถบันทึกภาพเคลื่อนไหวที่มีการเคลื่อนไหวที่รวดเร็วและนำมาทำภาพแอนิเมชันที่มีการเคลื่อนไหวที่รวดเร็วนั้นได้ ซึ่งการใช้กล้องที่มีอัตราเฟรมเรทที่ต่ำไม่สามารถทำภาพแอนิเมชันที่มีการเคลื่อนไหวรวดเร็วได้ ซึ่งจากโครงงานนี้กล้องที่เราใช้นั้นมีอัตราเฟรมเรทที่ต่ำดังนั้นเราไม่สามารถบันทึกภาพการเคลื่อนไหวที่รวดเร็วแล้วนำมาทำภาพแอนิเมชันที่มีการเคลื่อนไหวที่รวดเร็วนั้นได้ การวิเคราะห์และปรับปรุงค่าผิดพลาดนั้นสามารถดูได้ในบทที่ 8 เรื่องผลการทดลองและการทดสอบ

7.2.4 ซเทริโอออฟซิส (Stereopsis)

ซเทริโอออฟซิสเป็นกระบวนการที่ทำให้เราสามารถอ้างอิงตำแหน่งของวัตถุในรูปสามมิติได้โดยพิจารณาภาพอย่างน้อย 2 ภาพโดยภาพเหล่านั้นถ่ายในมุมมองที่แตกต่างกัน ซึ่งเราได้พูดถึงไปแล้วในบทที่ 4 และในหัวข้อนี้จะพูดถึงการนำทฤษฎีนี้มาใช้งาน

ในส่วนของการทำงานในซเทริโอออฟซิสนี้ จากรูปที่ 7.1 มันจะได้รับข้อมูลตำแหน่งการเคลื่อนที่ของมาคร์เกอร์ในรูปของตำแหน่งพิกเซลที่ผ่านขั้นตอนการวิเคราะห์และปรับปรุงค่าผิดพลาดแล้ว โดยข้อมูลนี้จะอยู่ในรูปของตำแหน่งพิกเซล X และ Y ของมาคร์เกอร์แต่ละตัว เราจะนำตำแหน่งที่ได้นี้ไปคำนวณหาตำแหน่งของมาคร์เกอร์ที่อยู่ในรูปของคาทิเซียนคอดิเนท (Caretesian coordination)

7.2.5 การสร้างไฟล์แมกซ์สคลิป (Generate MaxScript)

ไฟล์สคลิปคือภาษาสคลิปที่สร้างขึ้นมาสำหรับโปรแกรมสามดี สตูดิโอ แมกซ์ (3D Studio max) เพื่อนำมาใช้ในงานหลายๆอย่างในโปรแกรมสามดีสตูดิโอแมกซ์เอง ซึ่งอย่างหนึ่งที่ภาษาแมกซ์สคลิปสามารถทำได้คือการทำแอนิเมชัน เราจึงนำความสามารถตรงส่วนนี้ของแมกซ์สคลิปมาใช้ในโครงงานนี้

ในการสร้างไฟล์สคลิปนี้จะถูกสร้างขึ้นเองโดยโปรแกรมของเรา โดยมันจะทำการรับข้อมูลตำแหน่งที่ได้จากซเทริโอออฟซิสมาทำการเขียนเป็นโค้ดแมกซ์สคลิปโดยในกระบวนการทำนี้เราจะทำการคูณตำแหน่งที่ได้คือในแนวแกน X , Y และ Z ด้วยค่า 10 เพราะว่าในโปรแกรมสามดี สตูดิโอแมกซ์ การเคลื่อนที่ 1 หน่วยจะเท่ากับ 10 สมมุติเรากำหนดให้มีหน่วยเป็นนิ้ว ถ้าเราต้องการเคลื่อนที่เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระยะทาง 1 นิ้วในโปรแกรมสามดี สตูดิโอ แมกซ์ เราต้องทำการกำหนดให้มีค่าการเคลื่อนที่เท่ากับ 10 ดังนั้นนี่จึงเป็นเหตุผลที่ต้องคูณ 10 เข้าไป รูปที่ 7.5 แสดงตัวอย่างโค้ดแมกซ์สคลิปที่ได้จากการสร้างจากโปรแกรมที่เราเขียนขึ้น

msc6.ms - MAXScript

File Edit Search Help

```
fileIn "D:\MotionCapture\ScripFile\AppScript\FnScript11.ms"
animationRange = Interval 0 10
set animate on
set time 0f
    Ba0 = #(0.000000,0.000000,0.000000)
    Ba1 = #(0.000000,0.000000,0.000000)
    Ba2 = #(0.000000,0.000000,0.000000)
    Ba3 = #(0.000000,0.000000,0.000000)
    Ba4 = #(0.000000,0.000000,0.000000)
    Ba5 = #(0.000000,0.000000,0.000000)
    Ba6 = #(0.000000,0.000000,0.000000)
    Ba7 = #(0.000000,0.000000,0.000000)
    Ba8 = #(0.000000,0.000000,0.000000)
    Ba9 = #(0.000000,0.000000,0.000000)
    Ba10 = #(0.000000,0.000000,0.000000)
    call_Move()
set time 10f
    Ba0 = #(0.000000,0.000000,0.000000)
    Ba1 = #(0.000000,0.000000,0.000000)
    Ba2 = #(0.000000,0.000000,0.000000)
    Ba3 = #(0.000000,0.000000,0.000000)
    Ba4 = #(0.000000,0.000000,-3.125000)
    Ba5 = #(0.000000,0.000000,0.000000)
    Ba6 = #(0.000000,0.000000,0.000000)
    Ba7 = #(0.000000,0.000000,3.076923)
    Ba8 = #(0.000000,0.000000,0.000000)
    Ba9 = #(0.000000,0.000000,0.000000)
    Ba10 = #(0.000000,0.000000,0.000000)
    call_Move()
set animate off
max time play
```

*รูปที่ 7.5 แสดงตัวอย่างโค้ดแมกซ์สคลิป*

จากรูปที่ 7.5 สามารถอธิบายแต่ละส่วนของโค้ดแมกซ์สคลิปได้ดังนี้ บรรทัดแรกคือการเรียกใช้งานไฟล์ FnScript11.ms โดยเราจะพูดถึงไฟล์ตัวนี้ภายหลัง บรรทัดที่สองคือการเซ็ตจำนวนเฟรมในการทำแอนิเมชัน จากตัวอย่างเป็นการเซ็ตให้มีจำนวนเฟรม 293 เฟรมคือตั้งแต่เฟรมที่ 0 ถึงเฟรมที่ 292 บรรทัดที่สามคือการเซ็ตเพื่อบอกว่าต่อไปนี้คือการทำแอนิเมชัน และบรรทัดที่สี่คือการเซ็ตคีย์เฟรมที่ 0 บรรทัดที่ห้าถึงบรรทัดที่ สิบห้าคือการเซ็ตระยะทางการเคลื่อนที่ในคีย์เฟรมที่ 0 ให้กับตัวแปร โดยตัวแปร Ba0 ถึง Ba10 หมายถึงตัวมาคร์เกอร์ทั้ง 11 จุด บรรทัดที่สิบหกคือการเรียกใช้ฟังค์ชัน call_Move() ซึ่งเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังค์ชันที่อยู่ในไฟล์ FnScript11.ms ที่เราเรียกใช้ ส่วนสองบรรทัดสุดท้ายคือการบอกว่าสิ้นสุดการทำแอนิเมชันและทำการเรียกให้โปรแกรมทำการเล่นภาพแอนิเมชัน 3 มิติที่ได้สร้างไว้

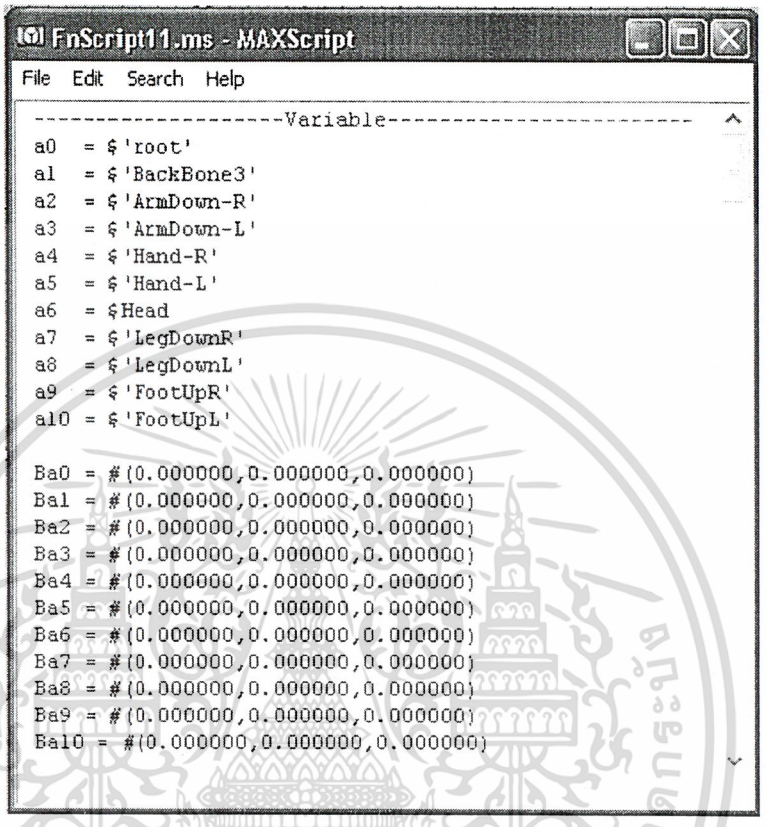
7.2.6 การทำแอนิเมชัน (Make animation)

ในส่วนของการทำแอนิเมชันนี้เป็นส่วนที่สำคัญอย่างหนึ่งในการทำโครงงานนี้ ใช่ว่าเราได้ตำแหน่งการเคลื่อนที่ของมาคร์เกอร์ทุกจุดมาแล้วจะสามารถนำมาทำเป็นภาพแอนิเมชันได้เลย เพราะว่าการเคลื่อนไหวของสิ่งมีชีวิตอย่างเช่นมนุษย์ทุกส่วนของร่างกายจะมีความสัมพันธ์ต่อกัน จากรูปที่ 7.6 แสดงโครงสร้างของร่างกายมนุษย์ เมื่อเราทำการขยับแขนท่อนบนแขนท่อนล่างก็จะขยับตามไปด้วย ดังนั้นเมื่อเราเอาตำแหน่งการเคลื่อนที่ของมาคร์เกอร์ที่ติดตรงแขนไปทำเป็นภาพแอนิเมชัน เราก็จะได้ตำแหน่งการเคลื่อนที่ของแขนท่อนบนและแขนท่อนล่างเมื่อเราเอาตำแหน่งการเคลื่อนที่ของแขนท่อนบนที่ได้ไปทำแอนิเมชันแขนท่อนล่างก็จะเคลื่อนที่ตามต่อจากนั้นเราก็ทำแอนิเมชันของแขนท่อนล่าง ซึ่งความจริงแขนท่อนล่างเคลื่อนที่ไปแล้วตามแขนท่อนบนพอเรามาเซ็ตให้มันเคลื่อนที่อีกก็จะทำให้ผิดไป ดังนั้นเราจึงสร้างไฟล์แมกซ์สคลิปขึ้นมาไฟล์หนึ่งเพื่อช่วยแก้ปัญหาข้อนี้

*รูปที่ 7.6 แสดงโครงสร้างร่างกายของมนุษย์*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ที่ใช้ในการแก้ปัญหาที่เกิดขึ้นข้อนี้มีชื่อไฟล์ว่า FnScript11.ms โดยเราสามารถอธิบายโค้ดของไฟล์นี้ได้ดังนี้




*รูปที่ 7.7 ไฟล์สคลิปส่วนที่ 1*

จากรูปที่ 7.7 แสดงถึงการประกาศตัวแปรเป็นแบบโกบอล (Global) โดย a0 – a10 เป็นการประกาศอ็อบเจ็ค ของส่วนต่างๆของร่างกายตัวการ์ตูน 3 มิติ และ Ba0 – Ba10 เป็นการประกาศตัวแปรแบบอะเรย์ (array) เพื่อเก็บระยะทางการเคลื่อนที่ในแนวแกน X , Y และ Z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FnScript11.ms - MAXScript

File Edit Search Help

```
--------------------Method-----------------------
fn Keep_Old_Pos =
(
    global ta0  = #(a0.pos.x,a0.pos.y,a0.pos.z)
    global ta1  = #(a1.pos.x,a1.pos.y,a1.pos.z)
    global ta2  = #(a2.pos.x,a2.pos.y,a2.pos.z)
    global ta3  = #(a3.pos.x,a3.pos.y,a3.pos.z)
    global ta4  = #(a4.pos.x,a4.pos.y,a4.pos.z)
    global ta5  = #(a5.pos.x,a5.pos.y,a5.pos.z)
    global ta6  = #(a6.pos.x,a6.pos.y,a6.pos.z)
    global ta7  = #(a7.pos.x,a7.pos.y,a7.pos.z)
    global ta8  = #(a8.pos.x,a8.pos.y,a8.pos.z)
    global ta9  = #(a9.pos.x,a9.pos.y,a9.pos.z)
    global ta10 = #(a10.pos.x,a10.pos.y,a10.pos.z)
    return true
)

fn ActPro r_real r_model axis =
(
    if axis > r_real Then
      axis = r_real
    else if axis < r_real*-1 Then
      axis = r_real*-1
    Angle = asin(axis/r_real)
    return r_model*sin(Angle)
)
```

*รูปที่ 7.8 ไฟล์สคลิปส่วนที่ 2*

จากรูปที่ 7.8 นี้คือการประกาศเมทธอด(Method) จากรูปมีอยู่สองฟังค์ชันคือ Keep_Old_Pos() เป็นฟังค์ชันที่จะถูกเรียกเป็นอันดับแรก ฟังค์ชันนี้จะทำการเก็บตำแหน่งปัจจุบันของข้อต่อทุกจุด ก่อนที่จะทำการเคลื่อนไหวส่วนต่างๆของร่างกายของตัวการ์ตูนสามมิติไว้ในตัวแปรที่เป็นอะเรย์ส่วนฟังค์ชันต่อมาคือ ActPro() โดยจะมีการรับพารามิเตอร์อยู่ 3 ตัว ฟังค์ชันนี้จะทำหน้าที่ในการปรับระยะการเคลื่อนที่ให้เหมาะสมกับตัวการ์ตูนเมื่อคนที่เราทำการบันทึกการเคลื่อนไหวมา มีขนาดไม่เท่ากับตัวละคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FnScript11.ms - MAXScript

File Edit Search Help

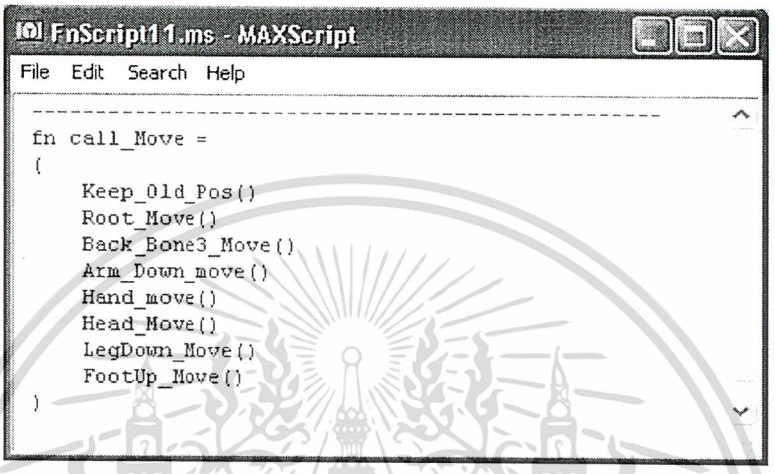
```
fn Root_Move =
(
    move a0[Ba0[1],Ba0[2],Ba0[3]]
)
fn Back_Bone3_Move =
(
    rr = 100.0
    rm = 30.0
    dz = ta1[3] - a1.pos.z
    dy = ta1[2] - a1.pos.y
    dx = ta1[1] - a1.pos.x

    dx += Ba1[1]
    dy += Ba1[2]
    dz += Ba1[3]
    dx = ActPro 100.0 60.0 dx
    dy = (rm*dy)/rr
    dz = ActPro rr rm dz
    move a1[dx,dy,dz]

)
fn Arm_Down_move =
(
    dz = ta2[3] - a2.pos.z
    dy = ta2[2] - a2.pos.y
    dx = ta2[1] - a2.pos.x
    dx += Ba2[1]
    dy += Ba2[2]
    dz += Ba2[3]
    move a2[dx,dy,dz]

    dz = ta3[3] - a3.pos.z
    dy = ta3[2] - a3.pos.y
    dx = ta3[1] - a3.pos.x
    dx += Ba3[1]
    dy += Ba3[2]
    dz += Ba3[3]
    move a3[dx,dy,dz]
)
```

*รูปที่ 7.9 ไฟล์สคลิปส่วนที่ 3*

จากรูปที่ 7.9 แสดงฟังค์ชันที่ใช้ในการขยับส่วนต่างๆของร่างกาย จากรูป ฟังค์ชัน Root_Move() จะทำการขยับส่วนที่เป็นรูทซึ่งเป็นส่วนที่สำคัญที่สุด ต่อมาจะมาทำการขยับส่วนของกระดูกส่วนหลังคือฟังค์ชัน Back_Bone3_Move() ซึ่งจะมีฟังค์ชันที่ใช้ในการขยับส่วนต่างๆของร่างกาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของตัวการ์ตูน 3 มิติครบตามทุกส่วนที่เราได้ทำการติดมาคร์เกอร์ไว้ ซึ่งจะไม่ได้เอามาแสดงไว้ทั้งหมด ท่านสามารถ ดูโค้ดทั้งหมดของไฟล์ตัวนี้ได้ทั้งหมดที่ภาคผนวก ก และฟังค์ชันสุดท้ายที่จะขอกล่าวถึงคล้ายๆกับเป็นเมนฟังค์ชันเพราะเป็นฟังค์ชันที่จะทำการเรียกฟังค์ชันที่กล่าวถึงมาแล้วให้ทำงานดังแสดงไว้ดังรูปที่ 7.10

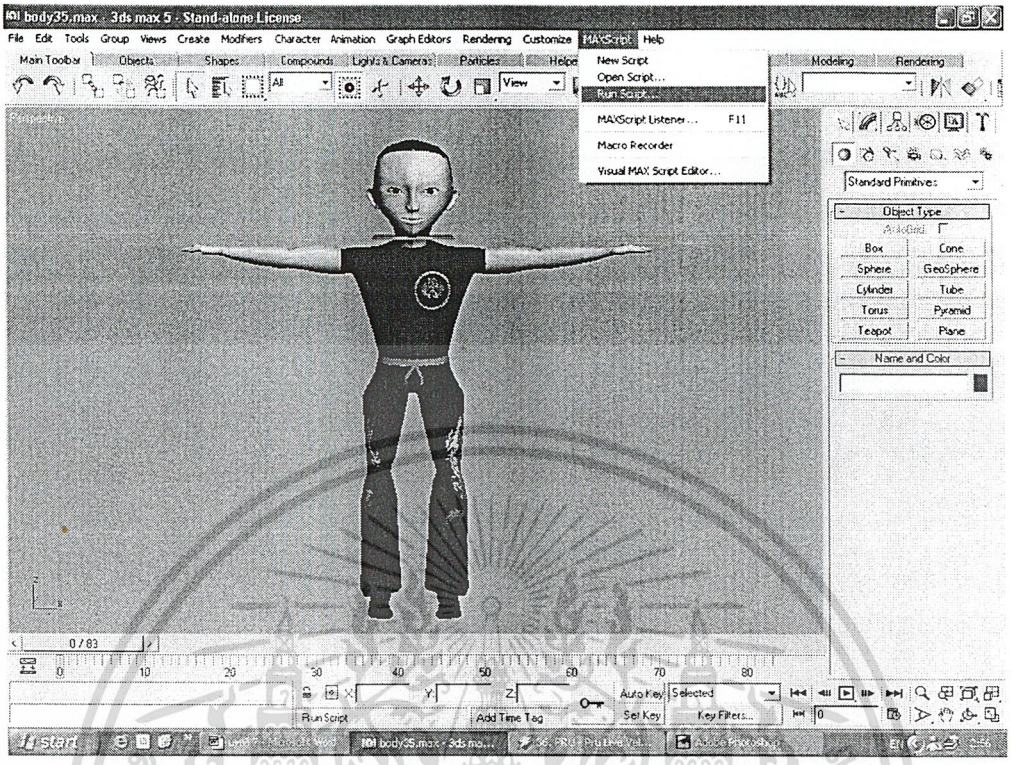



*รูปที่7.10 ไฟล์สคลิปส่วนที่ 4*

จากรูปที่ 7.10 จะมีฟังค์ชัน call_Move() โดยจะถูกเรียกใช้โดยโค้ดแมกซ์สคลิปที่โปรแกรมได้สร้างไว้ ที่พูดไว้แล้วในหัวข้อที่ 7.2.5 การสร้างไฟล์แมกซ์สคลิป โดยในฟังค์ชันนี้จะทำการเรียกฟังค์ชันอื่นๆให้ทำงานโดยลำดับการทำงานไม่สามารถสลับได้ ฟังค์ชันนี้จะถูกเรียกใช้ทุกๆครั้งที่ทำการเซ็ตคีย์เฟรม

ส่วนวิธีที่ที่จะทำการรันไฟล์สคลิปที่โปรแกรมได้สร้างขึ้นมานั้นสามารถทำได้โดยเราต้องทำการเปิดโปรแกรมสามดี สตูดิโอ แมกซ์ แล้วทำการเปิดตัวการ์ตูน 3 มิติ ที่เราได้สร้างขึ้นมา หลังจากนั้นให้เราไปที่คลิกที่เมนูแม็กซ์สคลิป แล้วทำการเลือกคำสั่ง รันสคลิป (Run Script) ดังแสดงไว้ดังรูปที่ 7.11 แล้วเราก็ทำการเลือกไฟล์สคลิปที่จะนำมาทำการรันเพื่อทำเป็นภาพแอนิเมชันสามมิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




*รูปที่ 7.11 แสดงวิธีการรันไฟล์แม็กซ์สคลิป*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**7.3 แผนภาพต่างๆ ของการออกแบบระบบ (Diagram)**

ในที่นี้เราจะอธิบายระบบโดยใช้ไดอะแกรม ของ ยูเอ็มเอ็ล (UML) 3 ชนิดคือ

- ยูสเคสไดอะแกรม ซึ่งใช้เพื่ออธิบายถึงผู้ใช้สามารถใช้งานใดกับระบบได้บ้าง
- ซีเควนไดอะแกรมถูกใช้อธิบายลำดับของการติดต่อกันของอ็อบเจ็คท
- คลาสไดอะแกรมซึ่งจะอธิบายถึงความสัมพันธ์ระหว่างคลาสที่ถูกออกแบบขึ้นมาเพื่อให้บรรลุถึงวัตถุประสงค์ของระบบ

7.3.1 ยูสเคสไดอะแกรม (Use Case Diagram)

*รูปที่ 7.12 แสดงยูสเคสไดอะแกรมของระบบ*

จากรูปที่ 7.12 แสดงยูสเคสไดอะแกรมของระบบโดยผู้ใช้สามารถใช้งานระบบได้ดังนี้

- Image Capture

  เป็นยูสเคสที่ติดต่อโดยผู้ใช้ ใช้สำหรับบันทึกการเคลื่อนไหวของสิ่งมีชีวิตที่เราสนใจ

- MotionProcess

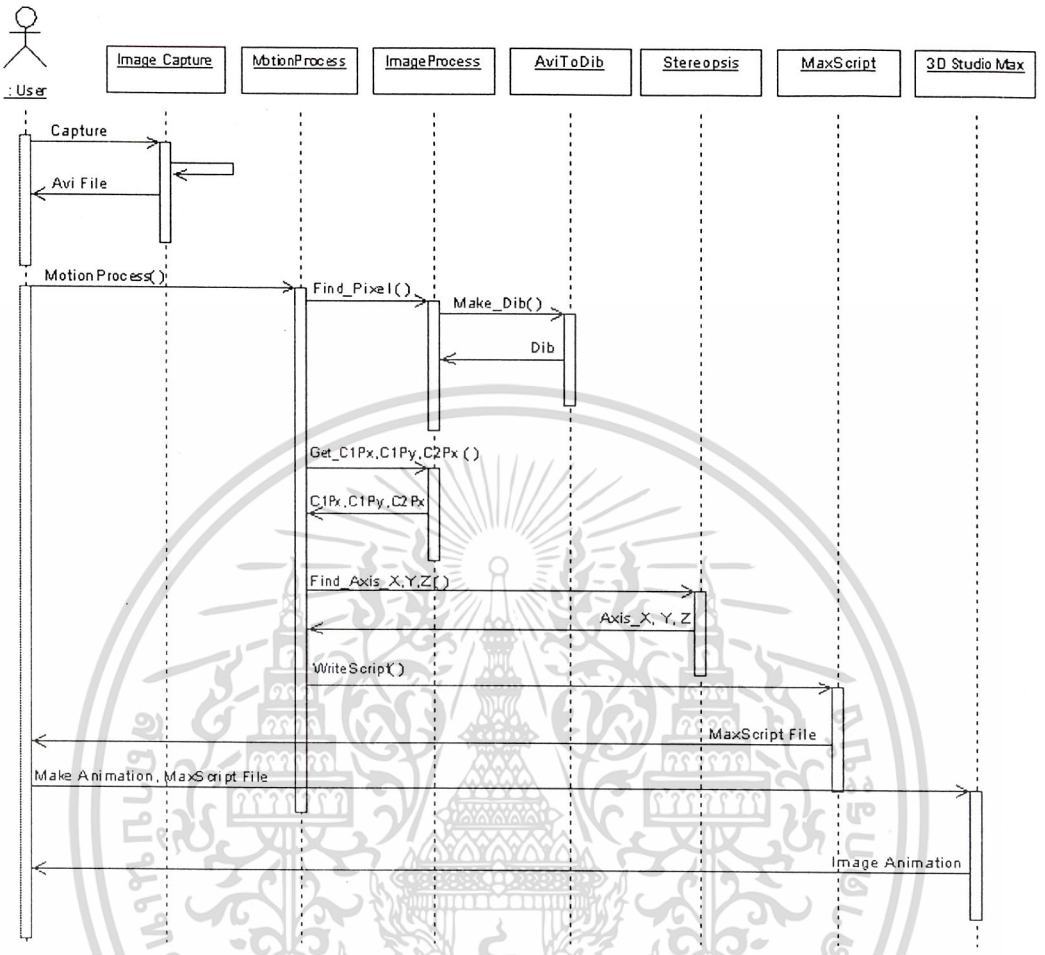
  เป็นยูสเคสที่จะทำการวิเคราะห์เพื่อหาตำแหน่งในโลกจริงแล้วแปลงไปสู่โลก 3 มิติ

- Make_Animation

  เป็นยูสเคสที่ผู้ใช้ ใช้ในการสร้างภาพแอนิเมชัน ที่มีการเคลื่อนไหวได้เหมือนจริงกับการเคลื่อนไหวของสิ่งมีชีวิตต้นแบบที่เราทำการบันทึกภาพเคลื่อนไหวนั้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3.2 ซีเควนไดอะแกรม (Sequence Diagram)




*รูปที่ 7.13 แสดงซีเควนไดอะแกรมของระบบ*

จากรูปที่ 7.13 สามารถอธิบายลำดับการใช้งานของวัตถุเป็นขั้นๆ ได้ดังนี้

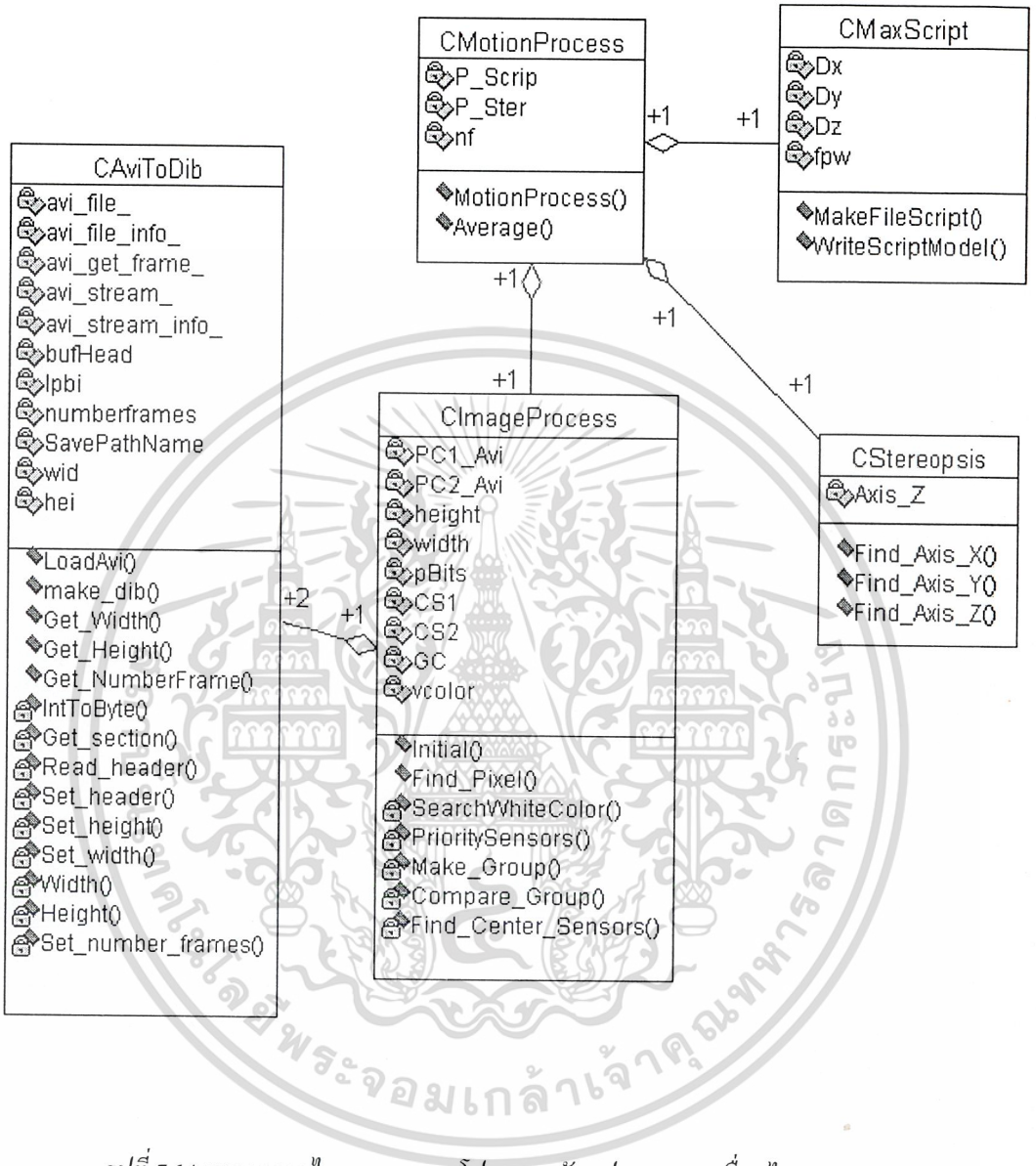
1. ผู้ใช้ต้องการบันทึกภาพเคลื่อนไหวของวัตถุที่สนใจจึงส่งข้อความ Capture() ไปยังวัตถุ Capture Image เพื่อทำการบันทึกภาพวัตถุที่กำลังเคลื่อนไหวแล้วส่งค่าคืนกลับมาเป็นไฟล์วีดีโอ
2. ผู้ใช้ต้องการทำ Motion Capture จึ่งส่งข้อความ MotionProcess() เพื่อแจ้งให้วัตถุ MotionProcess ทำงาน วัตถุนี้จึงส่งข้อความ Find_Pixel ไปยังวัตถุ ImageProcess เพื่อทำการหาตำแหน่งพิกเซลที่สนใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. วัตถุ ImageProcess จะทำการประมวนผลต้องได้ข้อมูลของเฟรมที่ได้จากกล้องทั้งสองตัวก่อนจึงส่งข้อความ Make_Dib() ไปยังวัตถุ CAviToDib ให้ทำการส่งข้อมูลของ แต่ละเฟรมกลับมายังวัตถุ ImageProcess เพื่อทำการหาตำแหน่งพิกเซลต่อไป
4. วัตถุ MotionProcess ส่งข้อความ Get_C1Px(),Get_C1Py(),Get_C2Px() ไปยังวัตถุ ImageProcess เพื่อให้ส่งค่าตำแหน่งพิกเซลที่คำนวณได้กลับมา
5. วัตถุ MotionProcess ส่งข้อความ Get_Axis_X(),Get_Axis_Y(),Get_Axis_Z() ไปยังวัตถุ Stereopsis เพื่อทำการคำนวณหาตำแหน่งที่แท้จริงของวัตถุที่เราบันทึกมาแล้วทำการแปลงไปสู่ตำแหน่งของโลก 3 มิติ แล้วมันจะทำการส่งตำแหน่งในโลก 3 มิติที่หาได้กลับคืนมาให้
6. วัตถุ MotionProcess เมื่อได้รับข้อมูลที่ส่งกลับมาให้แล้วก็จะส่งข้อความ WriteScript()ไปให้วัตถุ แม็กส์สคลิป เพื่อทำการเขียนไฟล์แมกซ์สคลิป
7. User ได้รับไฟล์แมกซ์สคลิปแล้วก็จะทำการสร้างภาพแอนิเมชันในโลก 3 มิติโดยการส่งข้อความพร้อมกับไฟล์แมกซ์สคลิปไปยังวัตถุ 3D Studio Max เพื่อทำการสร้างภาพแอนิเมชันที่เหมือนกับการเคลื่อนที่ของวัตถุต้นแบบ เมื่อทำเสร็จ User ก็จะได้ภาพแอนิเมชันที่ตัวเองต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3.3 คลาสไดอะแกรม(Class Diagram)




*รูปที่ 7.14 แสดงคลาสไดอะแกรมของโปรแกรมจับรูปแบบการเคลื่อนไหว (Motion Capture)*

เราสามารถอธิบายแต่ละคลาสภายในไดอะแกรมได้ดังนี้

- คลาส CMotionProcess

มีหน้าที่ควบคุมการทำงานของคลาสต่างๆ โดยมี

1. MotionProcess() เป็นฟังค์ชันหลักในการทำงาน
2. Average() เป็นฟังค์ชันที่ใช้ในการคำนวณหาค่าผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คลาส CImageProcess

หน้าที่ในการคำนวณหาตำแหน่งพิกเซลที่เราสนใจ โดยมีเมทธอดดังนี้

1. Initial() มีหน้าที่ในการส่งชื่อไฟล์วีดีโอ ที่ได้จากการจับภาพจากกล้องทั้งสองตัว ไปให้กับคลาส CaviToDib
2. Find_Pixel() ทำหน้าที่ดึงเอาข้อมูลของเฟรมที่ได้จากคลาส CAviToDib และทำการเรียกใช้ ฟังก์ชัน SearchWhiteColor()
3. SearchWhiteColor()ทำหน้าที่ในการหาตำแหน่งพิกเซลที่เป็นสีขาว ซึ่งเป็นสีที่เราสนใจ
4. PrioritySensors() ทำหน้าที่ในการจัดลำดับความสำคัญของมาคร์เกอร์
5. Make_Group() ทำหน้าที่ในการแบ่งกลุ่มพิกเซลที่เป็นจุดมาคร์เกอร์เดียวกัน
6. Compare_Group() ทำหน้าที่ในการเปรียบเทียบว่ามาคร์เกอร์ที่หาได้นั้นเป็นมาคร์เกอร์ตรงจุดใดของร่างกาย
7. Fine_Center_Sensors() ทำหน้าที่ในการคำนวณหาจุดศูนย์กลางของมาคร์เกอร์ที่เราตรวจเช็คมาได้

- คลาส CaviToDib

ทำหน้าที่ในการอ่านไฟล์วีดีโอและทำการดึงเอาข้อมูลของแต่ละเฟรมออกมาโดยเฟรมพวกนี้จะถูกเก็บอยู่ในรูปของฟอร์แมตดีไอบี

1. LoadAvi( ) ทำหน้าที่โหลดไฟล์วีดีโอ
2. Make_dib() ทำหน้าที่ในดึงข้อมูลจากเฟรม และส่งค่ากลับไปให้ยังคลาส CImageProcess()
3. Get_Width( ) ทำหน้าที่ดึงเอาค่าความกว้างของภาพ
4. Get_Height( ) ทำหน้าที่ดึงเอาค่าความสูงของภาพ
5. Get_NumberFrame() ทำหน้าที่ดึงเอาค่าจำนวนเฟรมทั้งหมดของไฟล์วีดีโอ จาก Header ของไฟล์

- คลาส Cstereopsis

ทำหน้าที่คำนวณหาค่าคาทิเซียนคอดิเนท

1. Find_Axis_X( ) คำนวณหาตำแหน่งแกน 'X' ของวัตถุในโลกจริง
2. Find_Axis_Y( ) คำนวณหาตำแหน่งแกน 'Y' ของวัตถุในโลกจริง
3. Find_Axis_Z( ) คำนวณหาตำแหน่งแกน 'Z' ของวัตถุในโลกจริง

- คลาส CmaxScript

1. MakeFileScript( ) ทำหน้าที่สร้างไฟล์ MaxScript (*.ms)
2. WriteScriptModel( ) ทำหน้าที่เขียนไฟล์ แม็กส์สคลิป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 8

# ผลการทดลองและการทดสอบ

## 8.1 วัตถุประสงค์ในการทดสอบ

ในส่วนของการทดสอบนี้จะเป็นส่วนของการทดสอบความสามารถของโปรแกรมจับรูปแบบการเคลื่อนไหว เพื่อเป็นการทดสอบว่าเราสามารถทำภาพแอนิเมชันที่มีการเคลื่อนไหวเหมือนกับการเคลื่อนไหวจริงที่เราได้ทำการบันทึกมาว่ามากน้อยเพียงใด

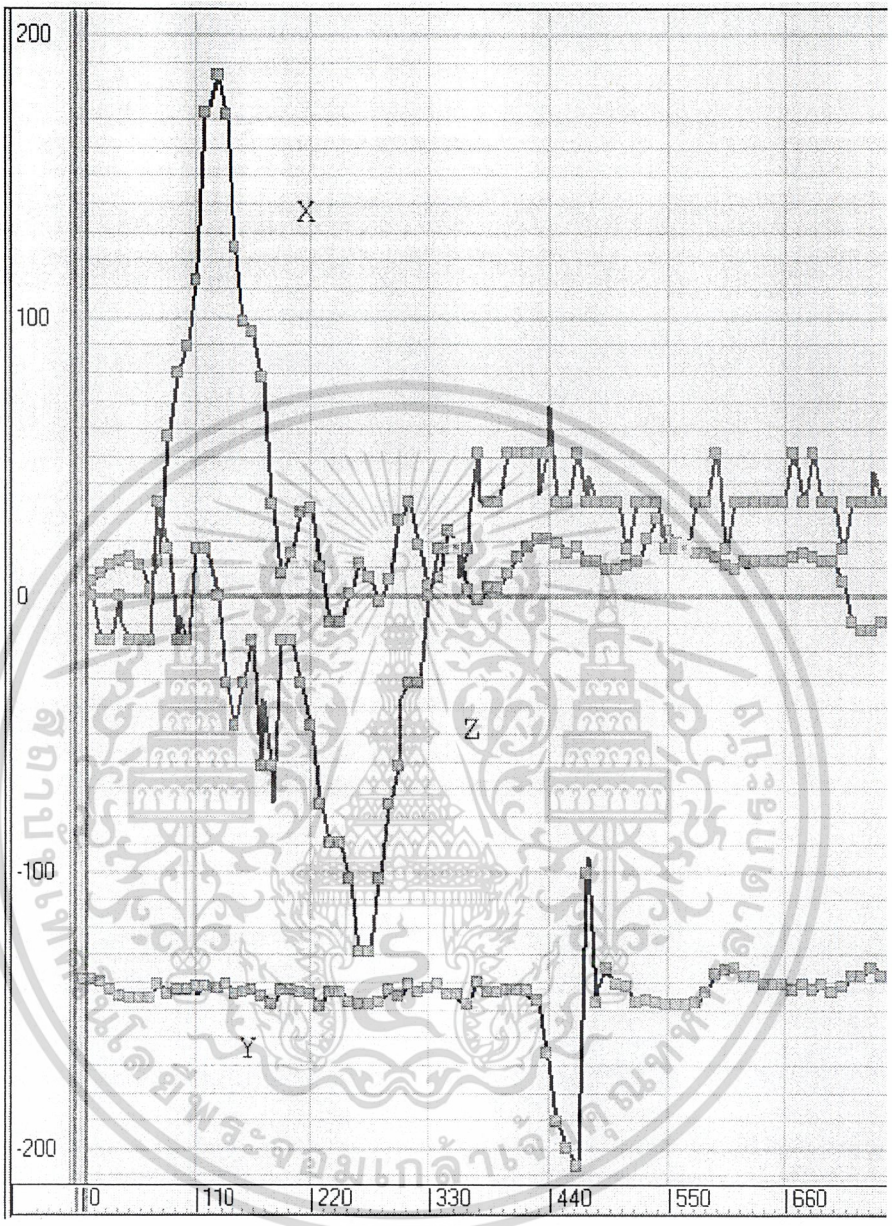
## 8.2 การทดสอบโปรแกรมจับรูปแบบการเคลื่อนไหว

การทดสอบนี้เราจะทำการบันทึกการเคลื่อนไหวของมนุษย์แล้วเราจะนำมาพิจารณาผลลัพธ์ที่ออกมาในที่นี้เราจะแสดงด้วยกราฟ โดยแกวตั้งคือตำแหน่งการเคลื่อนที่และแกวนอนคือตำแหน่งเฟรม และเราจะให้อัตราเฟรมเรทของกล้องที่ใช้ในการบันทึกอยู่ที่ 25 เฟรมต่อวินาที เพราะกล้องที่เราใช้สามารถบันทึกภาพได้อยู่ที่อัตราเฟรมเรทนี้ โดยเราจะทำการเปลี่ยนช่วงในการเซ็ทคีย์เฟรมในการทำภาพแอนิเมชันและทำการดูผลลัพธ์การเคลื่อนที่ที่ได้นั้น ซึ่งในการทดสอบนี้เราจะทำการทดสอบตรงจุดมาคร์เกอร์ตำแหน่งรูท(root) ค่าการเซ็ทคีย์เฟรมคือค่าที่กำหนดว่ากี่เฟรมถึงจะทำการเซ็ทตำแหน่งการเคลื่อนที่สักครั้งหนึ่ง

จากรูปกราฟ 8.1 ถึง รูปที่ 8.4 ที่แสดงในหน้าถัดไป เราจะเห็นว่า เมื่อเราทำการเซ็ทให้ค่าคีย์เฟรมน้อย ข้อมูลการเคลื่อนที่ของเซ็นเซอร์จะอยู่ครบแต่ก็มีข้อเสียตรงที่ค่าผิดพลาดก็มีเยอะเช่นเดียวกัน เพราะว่าเราไม่มีข้อมูลที่จะนำมาทำการคำนวณหาค่าผิดพลาด เพื่อทำการปรับให้ค่าความผิดพลาดนั้นน้อยลง แต่เมื่อเราทำการเซ็ทค่าคีย์เฟรมให้มากขึ้นเราจะสังเกตเห็นว่าค่าผิดพลาดนั้นน้อยลงแต่ข้อมูลการเคลื่อนที่บางส่วนก็หายไปด้วย สามารถสังเกตเห็นได้จากการเคลื่อนที่ในแนวแกน Y ตรงตำแหน่งประมาณเฟรมที่ 440 จากกราฟทำให้เรารู้ว่าช่วงเวลานั้นมาคร์เกอร์มีการเคลื่อนที่ที่รวดเร็วเพราะว่าจากกราฟตรงตำแหน่งนั้นมีระยะการเคลื่อนที่ที่มากแต่ใช้เวลาในการเคลื่อนที่น้อย เมื่อเราให้ค่าคีย์เฟรมมีค่า 1 การเคลื่อนที่ในแนวแกน Y นั้นจะมีค่าประมาณ -100 แต่เมื่อเราให้ค่าคีย์เฟรมเพิ่มมากขึ้น ค่าการเคลื่อนที่ในแนวแกน Y นั้นจะลดต่ำลง แสดงให้เห็นว่า ถ้าเราใช้กล้องที่อัตราเฟรมเรทที่ต่ำมาทำการเคปเจอร์การเคลื่อนไหวที่รวดเร็วจะให้ผลที่แย่มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

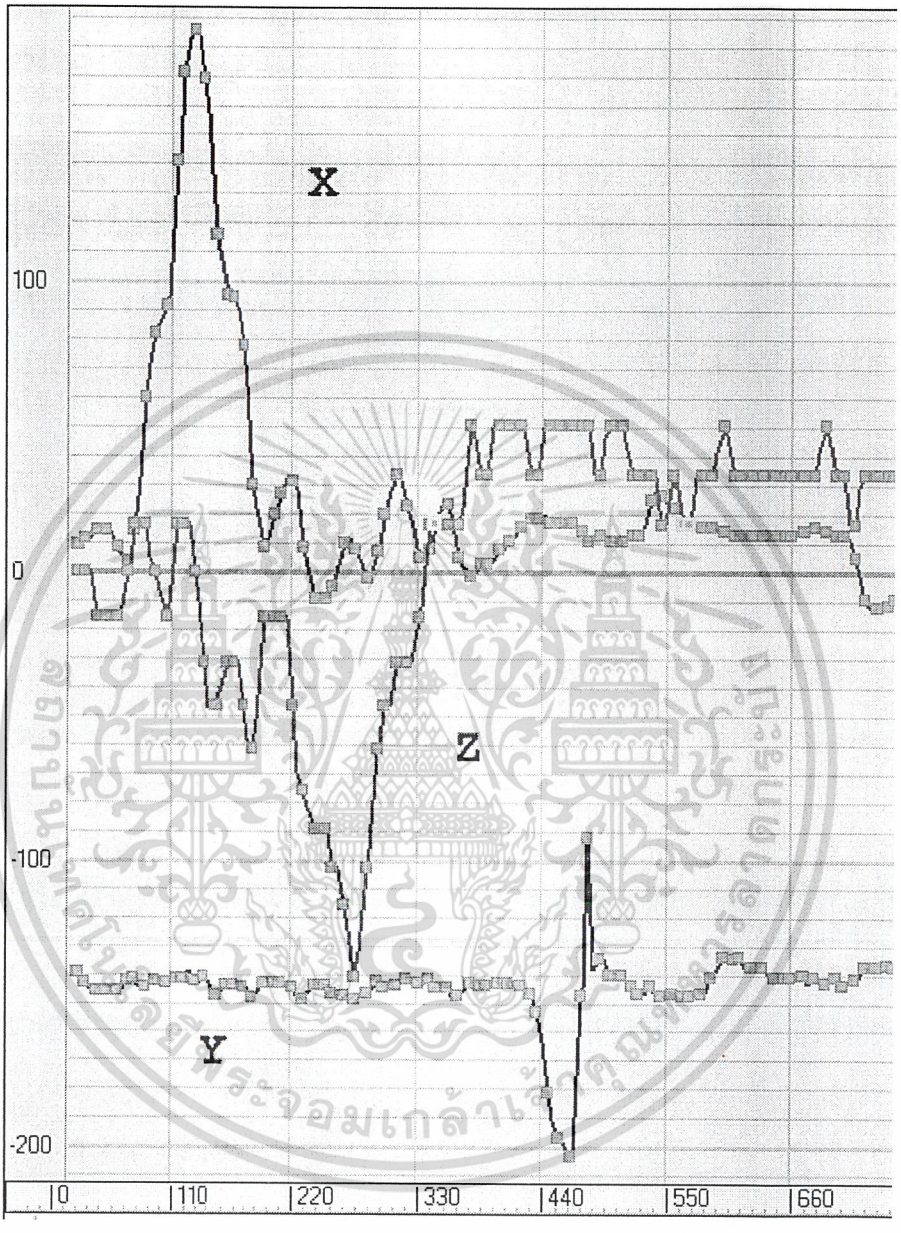
- เมื่อเรากำหนดให้ระยะของคีย์เฟรมเป็น 1




*รูปที่ 8.1 กราฟของภาพแอนิเมชันเมื่อคีย์เฟรมเป็น 1*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

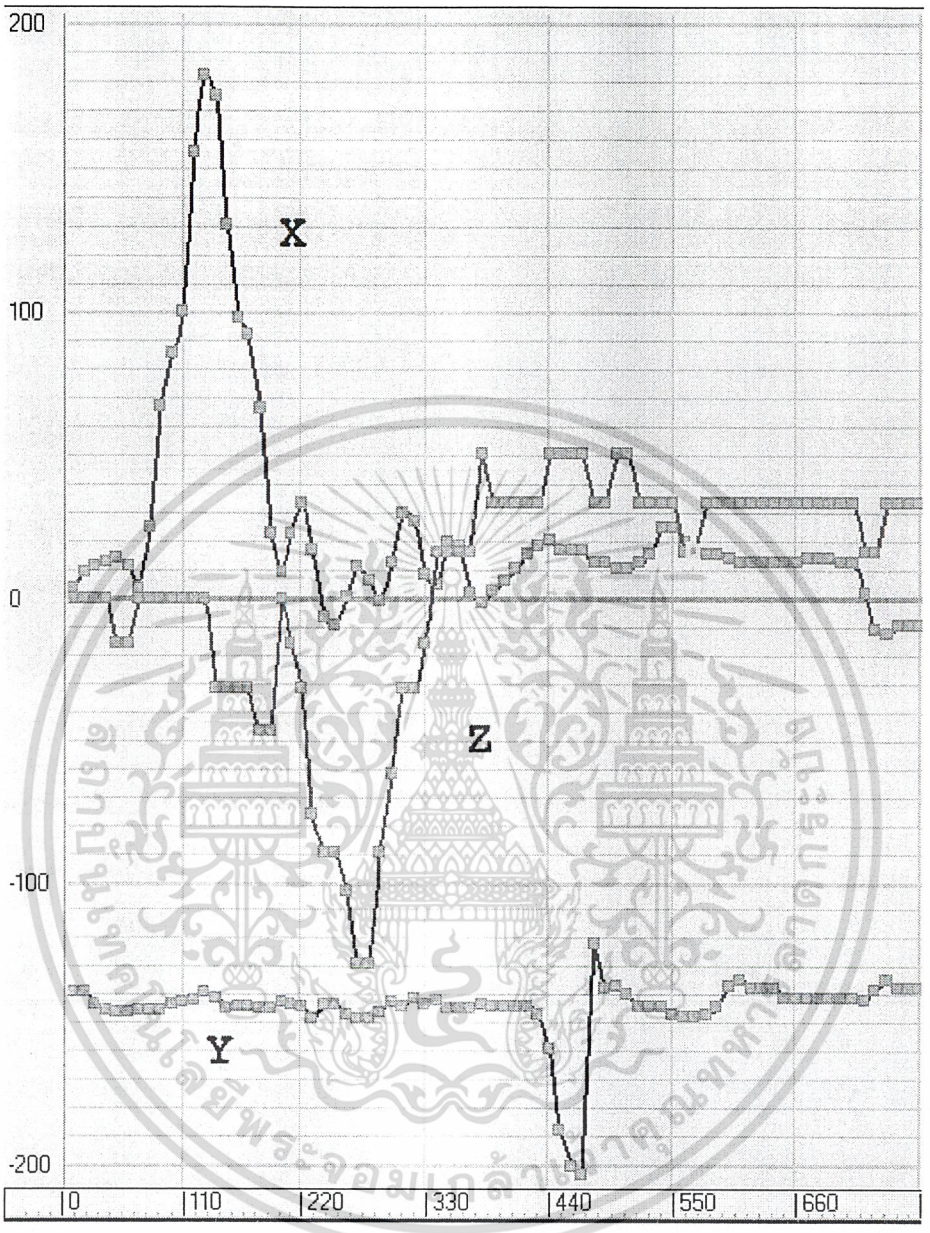
- เมื่อเรากำหนดให้ระยะของคีย์เฟรมเป็น 5




*รูปที่ 8.2 กราฟของภาพแอนิเมชันเมื่อคีย์เฟรมเป็น 5*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

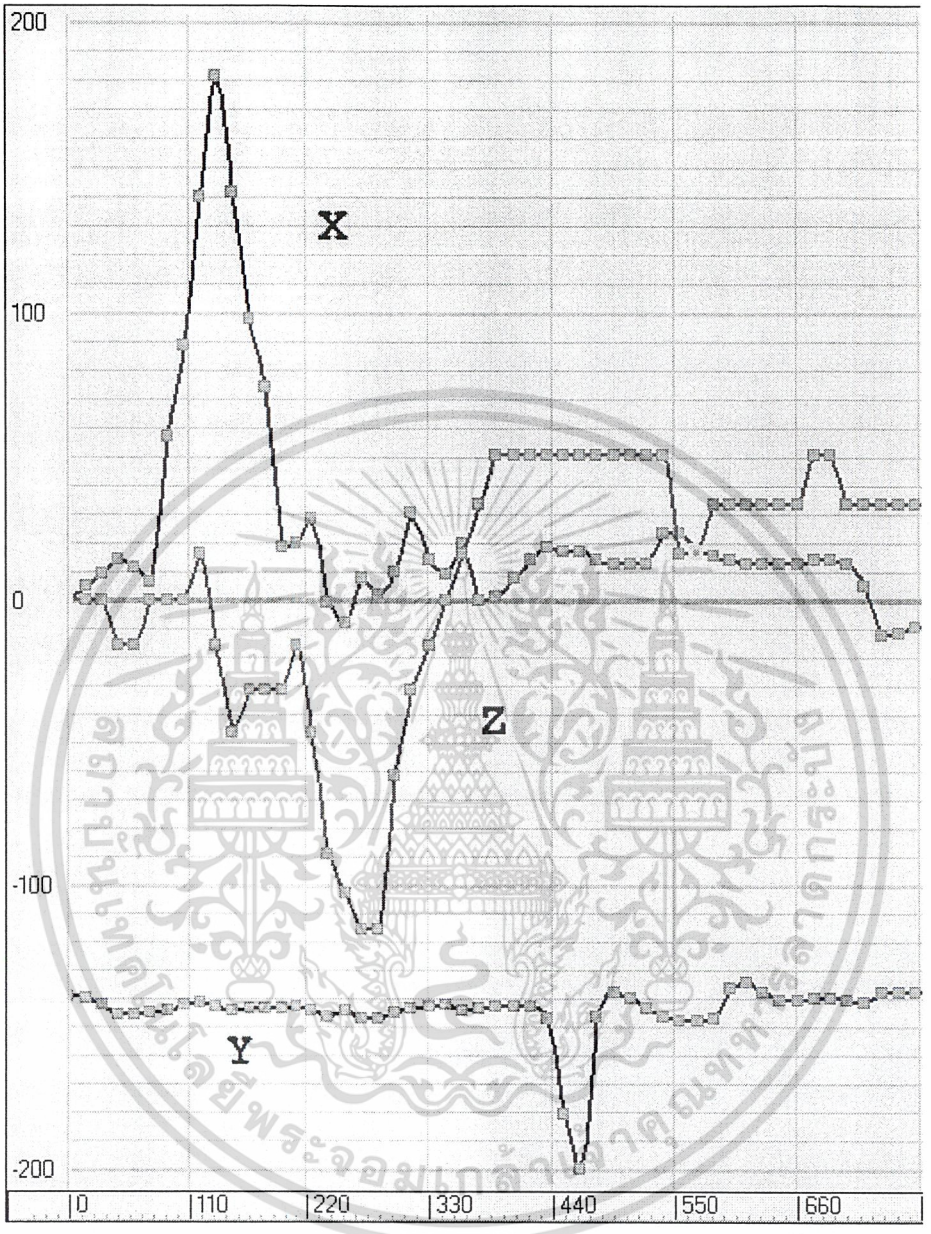
- เมื่อเรากำหนดให้ระยะของคีย์เฟรมเป็น 10




*รูปที่ 8.3 กราฟของภาพแอนิเมชันเมื่อคีย์เฟรมเป็น 10*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อเรากำหนดให้ระยะของคีย์เฟรมเป็น 15

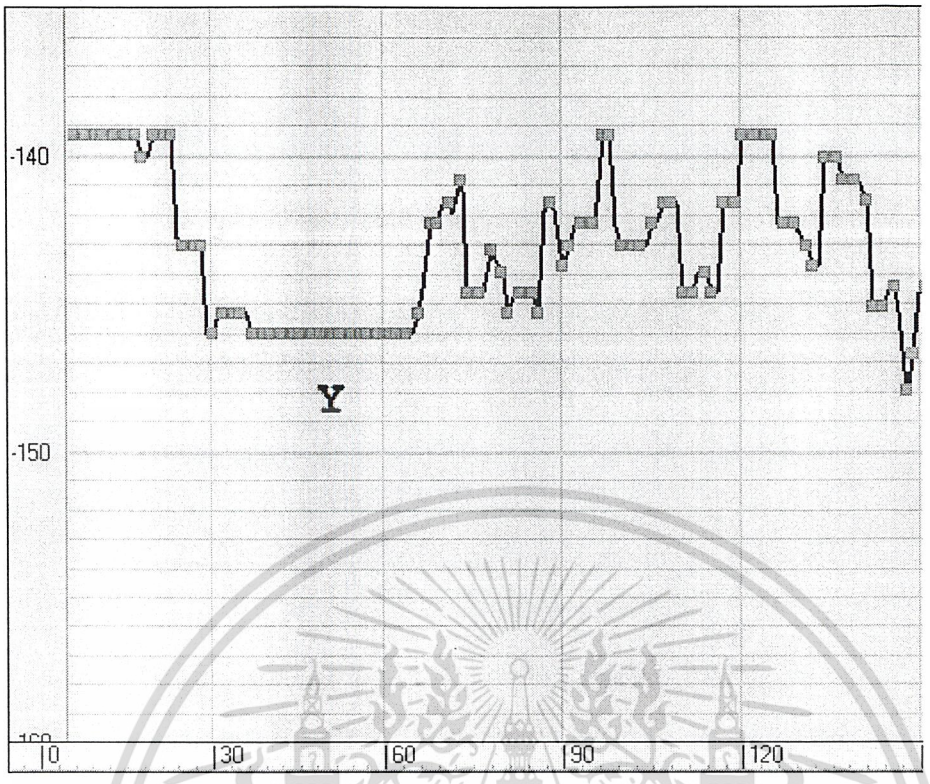



*รูปที่ 8.4 กราฟของภาพแอนิเมชันเมื่อคีย์เฟรมเป็น 15*

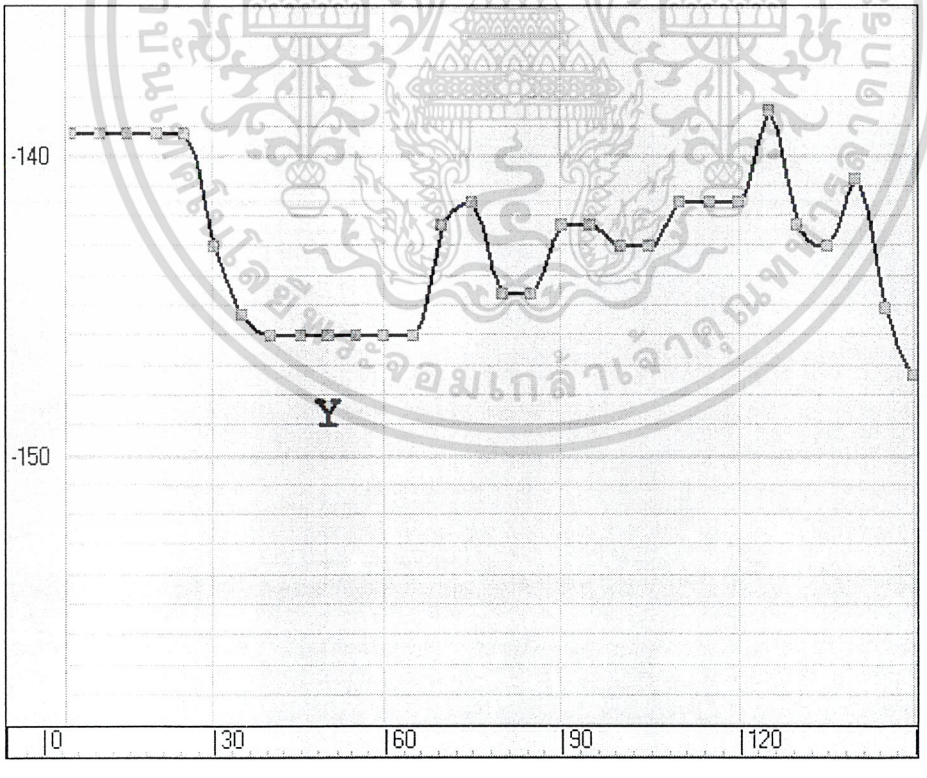
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อให้เห็นภาพชัดขึ้นเราได้ทำการขยายภาพกราฟให้ละเอียดมากขึ้นดังแสดงได้ดังรูปที่ 8.5 ถึง รูปที่ 8.8 โดยจะทำการขยายในส่วนของการเคลื่อนที่ในแนวแกน Y จากกราฟทำให้เราเห็นชัดเจนมากขึ้น ว่าเมื่อเซ็ทให้ค่าคีย์เฟรมน้อยคือ มีค่าข้อมูลที่จะนำมาวิเคราะห์หาค่าผิดพลาดน้อย เราจะสังเกตเห็นว่า กราฟที่ได้นั้นจะอยู่ในลักษณะการแกว่ง ซึ่งก็คือการเคลื่อนในแนวแกน Y ของตำแหน่งจุดนั้นจะมี ลักษณะของการเคลื่อนที่ไปยังตำแหน่งต่างๆในช่วงเวลาสั้นๆ หลายๆช่วงเวลาซึ่งผิดไปจากการเคลื่อนที่ที่ เราได้ทำการบันทึกมา แต่เมื่อเราทำการเซ็ทให้ค่าคีย์เฟรมเพิ่มมากขึ้น ทำให้กราฟที่ได้มีลักษณะการแกว่ง ที่ลดน้อยลงเป็นผลให้การเคลื่อนที่มีลักษณะที่ใกล้เคียงกับการเคลื่อนที่ที่เราทำการบันทึกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

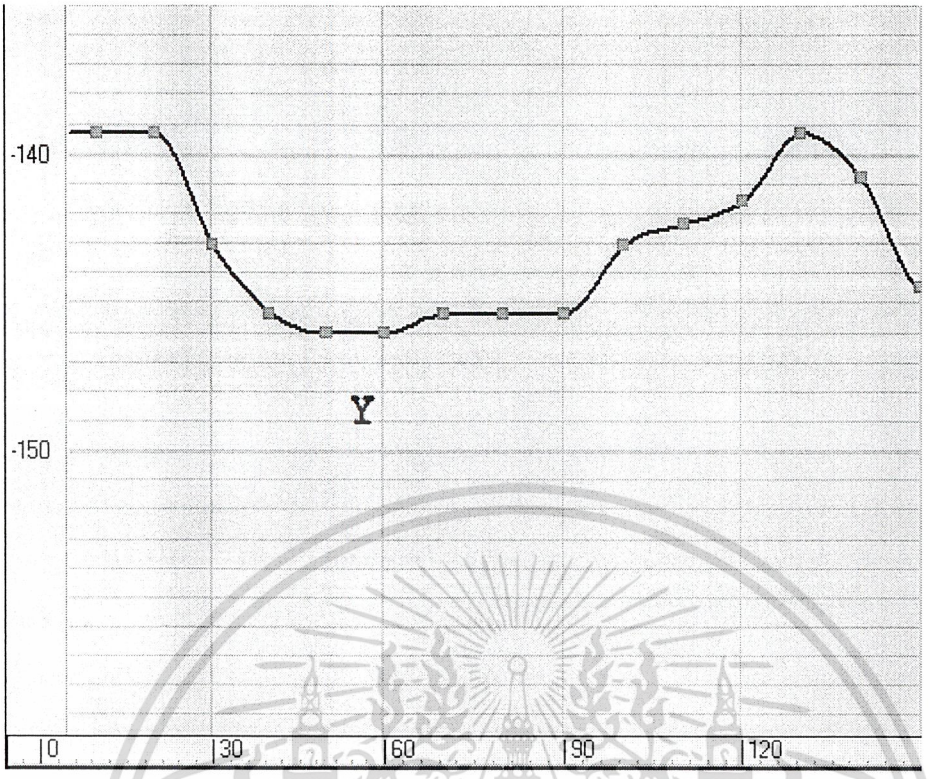



*รูปที่ 8.5 คีย์เฟรมเป็น 1*

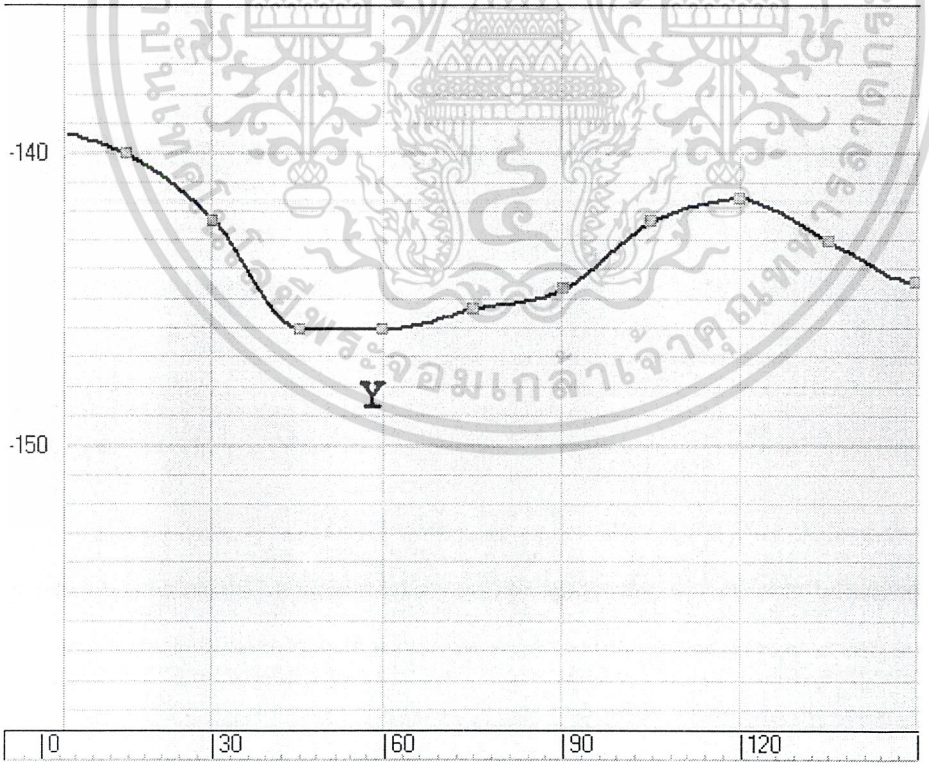



*รูปที่ 8.6 คีย์เฟรมเป็น 5*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



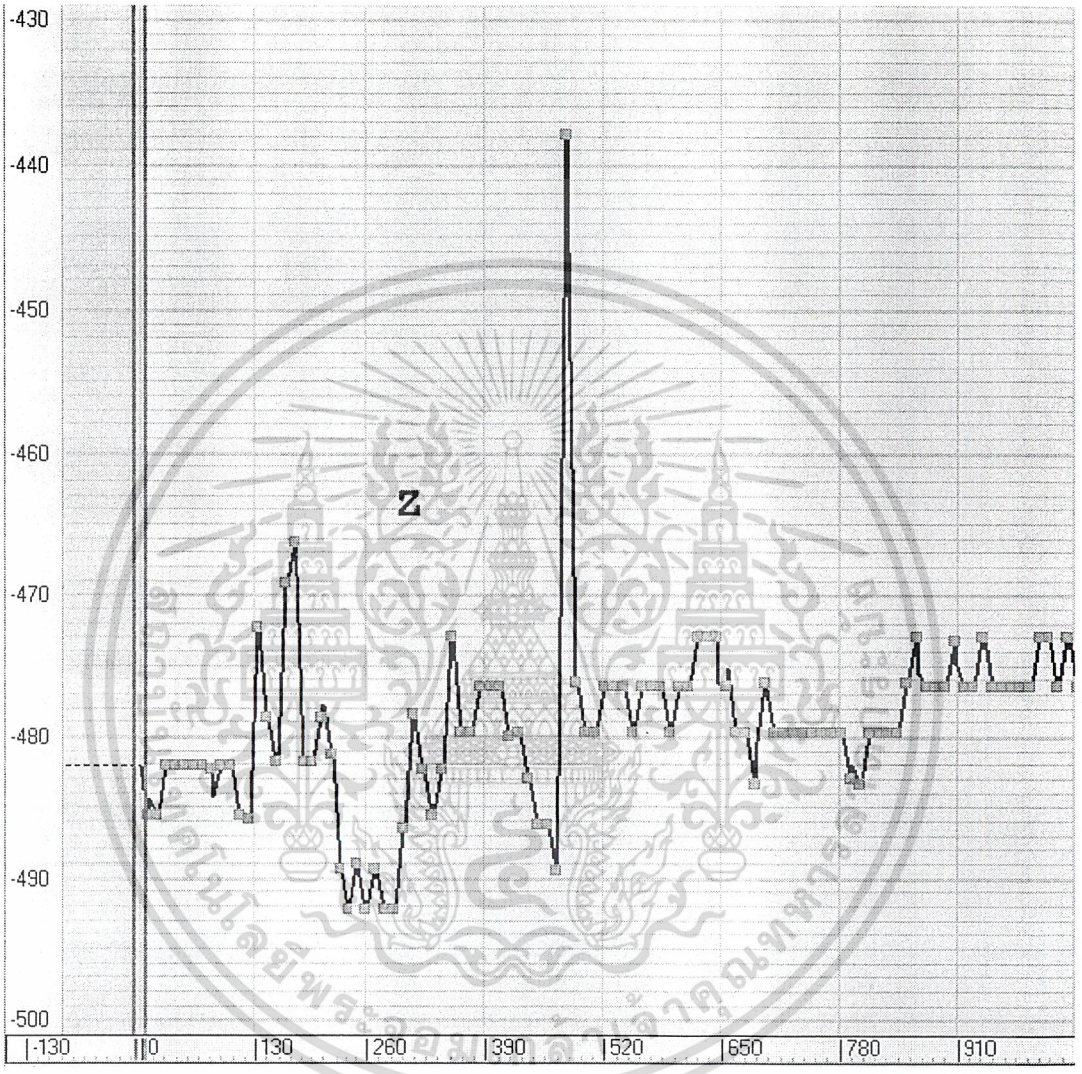

*รูปที่ 8.7 คีย์เฟรมเป็น 10*




*รูปที่ 8.8 คีย์เฟรมเป็น 15*

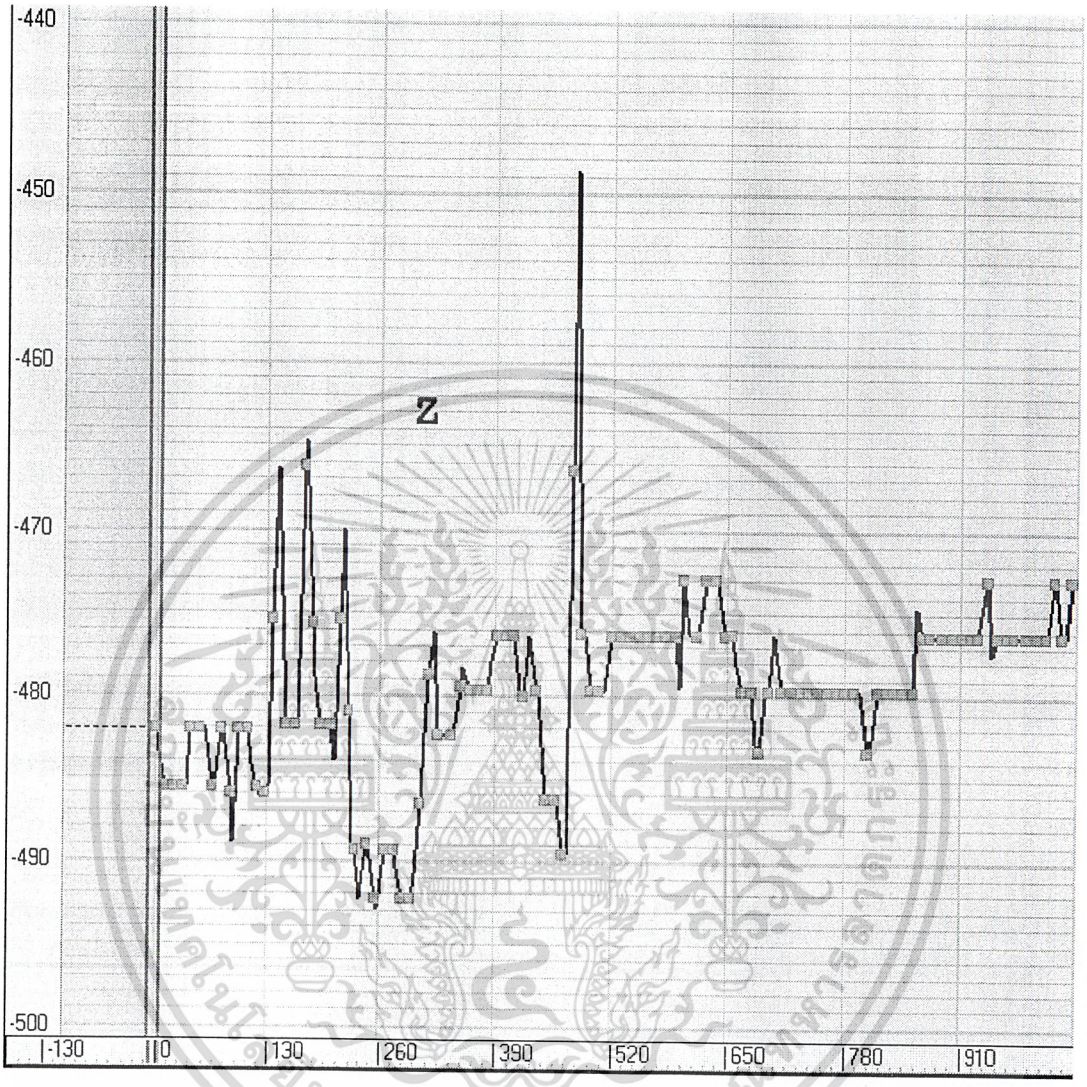
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพกราฟเพิ่มเติมที่ตำแหน่งการเคลื่อนที่ตรงตำแหน่งอื่น เราขอยกตัวอย่างการเคลื่อนที่ตรงส่วนของข้อเท้าด้านซ้ายมาให้พิจารณาเพิ่ม ดังรูปที่ 8.9 ถึง รูปที่ 8.12



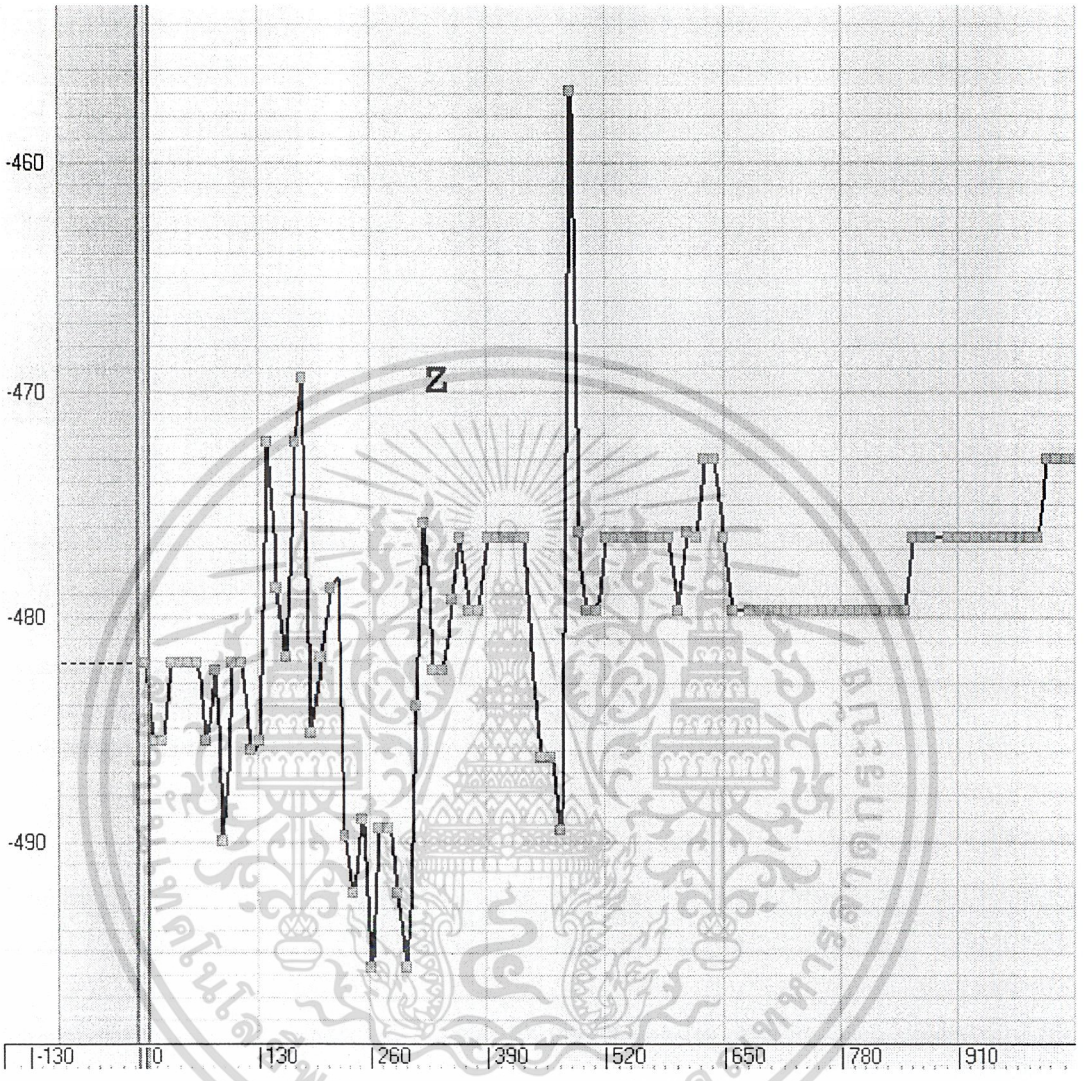

*รูปที่ 8.9 คีย์เฟรมเป็น 1*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



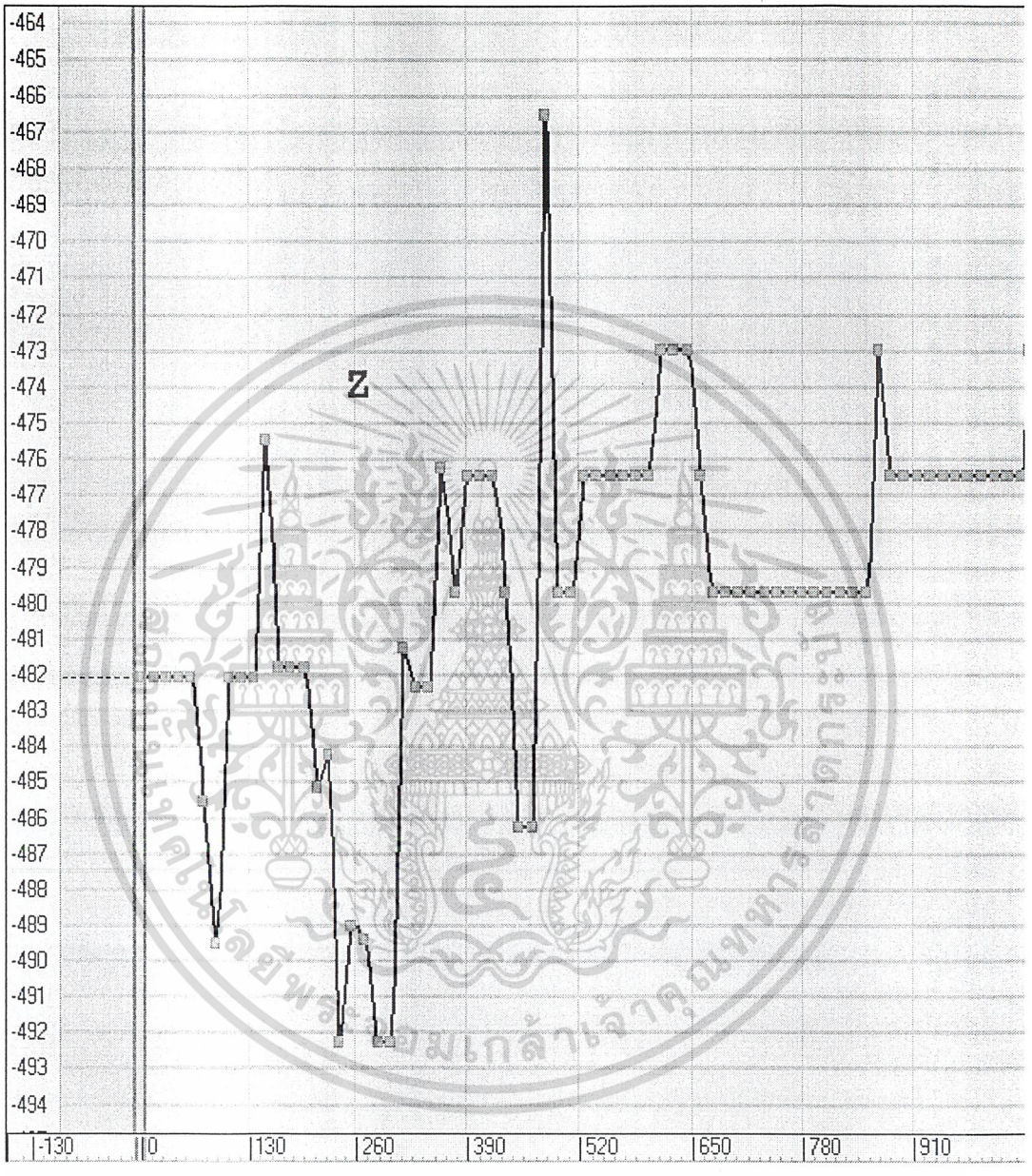

*รูปที่ 8.10 คีย์เฟรมเป็น 5*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




*รูปที่ 8.11 คีย์เฟรมเป็น 10*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




*รูปที่ 8.12 คีย์เฟรมเป็น 15*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 9
# บทวิจารณ์และสรุป

## 9.1 ประเมินผล

ในโครงงานนี้ เราได้พัฒนาโปรแกรมโดยใช้ภาษาภาษาซีพลัสพลัสและภาษาแมกซ์สคลิปร่วมกัน ในการคำนวณหาระยะทางการเคลื่อนที่ของสิ่งมีชีวิตที่เราสนใจโดยใช้กล้อง 2 ตัวในการบันทึก เพื่อนำมาทำเป็นภาพแอนิเมชัน 3 มิติ ที่มีการเคลื่อนไหวที่เหมือนกับการเคลื่อนไหวของสิ่งที่เราทำการบันทึกมา ผลที่ได้จากการทำนั้น ตัวหุ่นโมเดลมีการเคลื่อนไหวได้ใกล้เคียงกับการเคลื่อนที่จริงที่เราได้ทำการบันทึกมา แต่ไม่สามารถเคลื่อนที่ในท่าทางที่ทำให้มาคร์เกอร์หลุดออกจากกล้อง หรือถูกบังจากส่วนต่างๆ ของร่างกายได้ เพราะการที่จะคำนวณหาค่าพิกัดในรูปของคาทิเซียนคอดิเนท (Caretesian coordination) ได้นั้นภาพมาคร์เกอร์ต้องปรากฏให้เห็นทั้งกล้องทั้งสองตัว

## 9.2 แนวทางการพัฒนาต่อ

ในปัจจุบันการพัฒนาทางทางด้านเทคโนโลยีคอมพิวเตอร์นั้น ได้ก้าวหน้าไปอย่างรวดเร็ว โดยเฉพาะในด้านกราฟิกแอนิเมชัน ซึ่งในปัจจุบันนี้เราจะเห็นว่าภาพแอนิเมชันได้เข้าไปมีบทบาททางด้านสื่อต่างๆไม่ว่าจะเป็นทางโทรทัศน์หรือทางเว็บไซท์ต่างๆ หรือการนำภาพแอนิเมชัน 3 มิติไปใช้ในการทำภาพยนตร์หรือวงการแพทย์หรือนำไปสร้างเป็นเกม และโดยส่วนมากจะเป็นการเคลื่อนไหวของสิ่งมีชีวิตเช่น มนุษย์ หรือ สัตว์ แนวทางในการพัฒนาจึงน่าจะเป็นไปในทิศทางที่จะทำให้ภาพแอนิเมชันสมจริงมากขึ้น โดยใช้กล้องที่มีประสิทธิภาพมากขึ้นมีจำนวนกล้องมากขึ้น ทำให้แสดงท่าทางได้หลากหลาย มีการแคปเจอร์ที่ละเอียดมากขึ้นสามารถแคปเจอร์การแสดงสีหน้าของบุคคลในลักษณะต่างๆได้ เพื่อให้ภาพแอนิเมชันที่ออกมาสวยงามและสมจริงมากขึ้น และสามารถพัฒนาไปสู่การทำเป็นเรียวไทม (real time) ได้เป็นต้น

## 9.3 สรุป

จากผลลัพธ์ของโครงงานที่ออกมานั้น ได้บรรลุถึงจุดประสงค์ของโครงงานที่ตั้งไว้เป็นอย่างดี โดยทางผู้พัฒนาได้ใช้ความรู้ที่ได้ศึกษามาอย่างเต็มที่ จึงได้ผลงานออกมาอยู่ในระดับที่น่าพอใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ก

## โค้ดไฟล์ FnScript11.ms

```
--=======================================
-- Project :    Motion Capture
-- Modifier:        Kridsada Sawasdee
-- Complite:        20 February 2003
--=======================================
----------Variable----------
a0  = $'root'
a1  = $'BackBone3'
a2  = $'ArmDown-R'
a3  = $'ArmDown-L'
a4  = $'Hand-R'
a5  = $'Hand-L'
a6  = $Head
a7  = $'LegDownR'
a8  = $'LegDownL'
a9  = $'FootUpR'
a10 = $'FootUpL'

Ba0 = #(0.000000,0.000000,0.000000)
Ba1 = #(0.000000,0.000000,0.000000)
Ba2 = #(0.000000,0.000000,0.000000)
Ba3 = #(0.000000,0.000000,0.000000)
Ba4 = #(0.000000,0.000000,0.000000)
Ba5 = #(0.000000,0.000000,0.000000)
Ba6 = #(0.000000,0.000000,0.000000)
Ba7 = #(0.000000,0.000000,0.000000)
Ba8 = #(0.000000,0.000000,0.000000)
Ba9 = #(0.000000,0.000000,0.000000)
Ba10 = #(0.000000,0.000000,0.000000)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
----------Method----------
fn Keep_Old_Pos =
(
        global ta0  = #(a0.pos.x,a0.pos.y,a0.pos.z)
        global ta1  = #(a1.pos.x,a1.pos.y,a1.pos.z)
        global ta2  = #(a2.pos.x,a2.pos.y,a2.pos.z)
        global ta3  = #(a3.pos.x,a3.pos.y,a3.pos.z)
        global ta4  = #(a4.pos.x,a4.pos.y,a4.pos.z)
        global ta5  = #(a5.pos.x,a5.pos.y,a5.pos.z)
        global ta6  = #(a6.pos.x,a6.pos.y,a6.pos.z)
        global ta7  = #(a7.pos.x,a7.pos.y,a7.pos.z)
        global ta8  = #(a8.pos.x,a8.pos.y,a8.pos.z)
        global ta9  = #(a9.pos.x,a9.pos.y,a9.pos.z)
        global ta10 = #(a10.pos.x,a10.pos.y,a10.pos.z)
        return true
)
fn ActPro r_real r_model axis =
(
        if axis > r_real Then
          axis = r_real
        else if axis < r_real*-1 Then
          axis = r_real*-1
        Angle = asin(axis/r_real)
        return r_model*sin(Angle)
)
fn Root_Move =
(
        move a0[Ba0[1],Ba0[2],Ba0[3]]
)

fn Back_Bone3_Move =
(
        rr = 100.0
        rm = 30.0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        dz = ta1[3] - a1.pos.z
        dy = ta1[2] - a1.pos.y
        dx = ta1[1] - a1.pos.x
        dx += Ba1[1]
        dy += Ba1[2]
        dz += Ba1[3]
        dx = ActPro 100.0 60.0 dx
        dy = (rm*dy)/rr
        dz = ActPro rr rm dz
        move a1[dx,dy,dz]

)
fn Arm_Down_move =
(
        dz = ta2[3] - a2.pos.z
        dy = ta2[2] - a2.pos.y
        dx = ta2[1] - a2.pos.x
        dx += Ba2[1]
        dy += Ba2[2]
        dz += Ba2[3]
        move a2[dx,dy,dz]
        dz = ta3[3] - a3.pos.z
        dy = ta3[2] - a3.pos.y
        dx = ta3[1] - a3.pos.x
        dx += Ba3[1]
        dy += Ba3[2]
        dz += Ba3[3]
        move a3[dx,dy,dz]
)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fn Hand_move =
(
        dz = ta4[3] - a4.pos.z
        dy = ta4[2] - a4.pos.y
        dx = ta4[1] - a4.pos.x
        dx += Ba4[1]
        dy += Ba4[2]
        dz += Ba4[3]
        move a4[dx,dy,dz]
        dz = ta5[3] - a5.pos.z
        dy = ta5[2] - a5.pos.y
        dx = ta5[1] - a5.pos.x
        dx += Ba5[1]
        dy += Ba5[2]
        dz += Ba5[3]
        move a5[dx,dy,dz]
)
fn Head_Move =
(
        rr = 100.0
        rm = 50.0
        dz = ta6[3] - a6.pos.z
        dy = ta6[2] - a6.pos.y
        dx = ta6[1] - a6.pos.x
        dx += Ba6[1]
        dy += Ba6[2]
        dz += Ba6[3]
        dx = ActPro rr rm dx
        dy = (rm*dy)/rr
        dz = ActPro rr rm dz
        move a6[dx,dy,dz]
)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fn LegDown_Move =
(
	rr = 100.0
	rm = 140.0
	dz = ta7[3] - a7.pos.z
	dy = ta7[2] - a7.pos.y
	dx = ta7[1] - a7.pos.x
	dx += Ba7[1]
	dy += Ba7[2]
	dz += Ba7[3]
	move a7[dx,dy,dz]
	dz = ta8[3] - a8.pos.z
	dy = ta8[2] - a8.pos.y
	dx = ta8[1] - a8.pos.x
	dx += Ba8[1]
	dy = ActPro rr rm dy
	dz = ActPro rr rm dz
	move a8[dx,dy,dz]
)
fn FootUp_Move =
(
	dz = ta9[3] - a9.pos.z
	dy = ta9[2] - a9.pos.y
	dx = ta9[1] - a9.pos.x
	dx += Ba9[1]
	dy += Ba9[2]
	dz += Ba9[3]
	move a9[dx,dy,dz]
	dz = ta10[3] - a10.pos.z
	dy = ta10[2] - a10.pos.y
	dx = ta10[1] - a10.pos.x
	dx += Ba10[1]
	dy += Ba10[2]
	dz += Ba10[3]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        move a10[dx,dy,dz]
)
fn call_Move =
(
        Keep_Old_Pos()
        Root_Move()
        Back_Bone3_Move()
        Arm_Down_move()
        Hand_move()
        Head_Move()
        LegDown_Move()
        FootUp_Move()
)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บรรณานุกรม


[1] Menache,Alberto : "Understanding motion capture for computer animation and video game"

[2] R.C. Gonzalez and R. E. Woods : "Digital Image Processing" Prentice Hall, New Jersey

[3] J.R.Parker : "ALGORITHMS FOR IMAGE PROCESSING AND COMPUTER VISION" John Wiley & Sons,Inc ISBN 0-471-14056-2,1997

[4] นิรุธ อำนวยศิลป์ : "คู่มือการเขียนโปรแกรม Microsoft Visual C++ Version 6.0"

[5] ศุภพงศ์ เลิศสินธวานนท์ : "ก้าวสู่โลก 3 มิติ 3 D Studio MAX"

[6] พูนศักดิ์ ธนพันธ์พาณิช : "เทคนิคการสร้าง character modeling บน 3ds max 4.2"


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้