ก้อนดนตรีเสมือน

AR MUSIC CUBE JOCKER

นายคัมภีร์ อินทพงษ์ รหัส 51361933

นายไกรทอง พันสนิท รหัส 51364668

คณะวิศวกรรมศาสตร์
วันที่รับ 25 มิ.ย. 2556
เลขทะเบียน 1627092X
เลขเรียกหนังสือ ปร.
มหาวิทยาลัยนเรศวร 2554

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2554

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน	ก้อนดนตรีเสมือน
ผู้ดำเนินโครงงาน	นายคัมภีร์ อินทพงษ์ รหัส 51361933
	นายไกรทอง พันสนิท รหัส 51364668
ที่ปรึกษาโครงงาน	อาจารย์รัฐภูมิ วรานุสาสน์
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2554

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็น**ส่วนหนึ่งของ**การศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. พนมขวัญ ริยะมงคล)

.....กรรมการ

(ดร. วรลักษณ์ คงเด่นฟ้า)

.....กรรมการ

(อาจารย์สิรภพ คชรัตน์)

.....กรรมการ

(อาจารย์รัฐภูมิ วรานุสาสน์)

ชื่อหัวข้อโครงงาน	ก้อนดนตรีเสมือน	
ผู้ดำเนินโครงงาน	คัมภีร์ อินทพงษ์	รหัส 51361933
	ไกรทอง พันสนิท	รหัส 51364668
ที่ปรึกษาโครงงาน	อ.รัฐภูมิ วรานุสาสน์	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์	
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์	
ปีการศึกษา	2554	

บทคัดย่อ

โครงงานนี้นำเทคโนโลยีความจริงเสริม (Augmented Reality) มาประยุกต์เพื่อสร้างเครื่องดนตรีอิเล็กทรอนิกส์ที่สามารถผสมเสียงดนตรีและปรับจังหวะรวมทั้งความดังของเสียงดนตรีแต่ละเสียงได้ผ่านการจัดการกล่องทรงลูกบาศก์ที่เรียกว่า "ก้อนดนตรีเสมือน" โดยการทำงานหลักแบ่งได้เป็น 3 ส่วน ได้แก่ ส่วนแรก คือ การตรวจจับและประมาณตำแหน่งและทิศทางตัวของสัญลักษณ์ (Marker) ซึ่งติดอยู่บนด้านทั้งหกของก้อนดนตรีเสมือนผ่านทางกล้องเว็บแคม ส่วนที่สอง คือ การเล่นดนตรีโดยอาศัยข้อมูลตำแหน่งและทิศทางของสัญลักษณ์ในการเพิ่มหรือลดจังหวะและความดังของเสียงดนตรี และส่วนที่สาม คือ การแสดงภาพกราฟฟิกให้สอดคล้องกับตำแหน่งและทิศทางของสัญลักษณ์ที่ตรวจจับได้ โครงงานนี้แสดงให้เห็นถึงความเป็นไปได้การประยุกต์ใช้งานเทคโนโลยีความจริงเสริมด้านความบันเทิงและนันทนาการซึ่งมีความแปลกใหม่น่าสนใจแต่ใช้เพียงแค่อุปกรณ์คอมพิวเตอร์พื้นฐานเท่านั้น

Project title	AR Music Cube Jocker
Name	Mr. Kampee Intapong ID. 51361933
	Mr. Gaitong Punsanit ID. 51364668
Project advision	Mr. Rattapoom Waranusast
Major	Computer Engineering
Department	Electrical and Computer Engineering
Academic year	2011

Abstract

This project applies augmented reality technology to music in order to build an electronic music instrument which can mix different sounds and adjust tempo and volume of them through manipulations of several cubes, called "AR music cubes". The program consists of 3 major parts; first, the detection of positions and orientations of markers which are attached on each faces of each cube, second, playing music sounds with tempos and volumes corresponding to markers' poses, and third, rendering graphic elements according to markers' positions and orientations. This project shows possibilities of applications of augmented reality technology on entertainment and recreation, which are novel and interesting with only basic computer devices.

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปได้ด้วยความกรุณาเป็นอย่างยิ่งจากอาจารย์ที่ปรึกษา อาจารย์รัฐภูมิ วรานุสาสน์ ที่คอยให้คำปรึกษาและคอยแนะนำวิธีการแก้ปัญหามากมาย และให้ความกรุณาในการตรวจทานปริญญานิพนธ์ คณะผู้จัดทำโครงงานขอกราบขอบพระคุณเป็นอย่างสูงและขอระลึกถึงความกรุณาของท่านไว้ตลอดไป

ขอขอบพระคุณคณะกรรมการโครงงานทั้ง 3 ท่าน ผู้ช่วยศาสตราจารย์ ดร. พนมขวัญ ริยะมงคล ดร. วรลักษณ์ คงเด่นฟ้า และ อาจารย์สิรภพ คชรัตน์ ที่ช่วยแนะนำสิ่งที่ควรปรับปรุงโครงงาน

ขอขอบคุณคณาจารย์ทุกท่านที่ประสิทธ์ประสาทวิชาความรู้ให้กับคณะผู้จัดทำโครงงาน

ขอขอบคุณภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ที่ช่วยอนุเคราะห์สถานที่ในการจัดทำโครงงาน

และสุดท้ายนี้ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ที่คอยช่วยเหลือและคอยให้การสนับสนุนในทุกด้านจนทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

นายคัมภีร์ อินทพงษ์

นายไกรทอง พันสนิท

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ก

บทคัดย่อภาษาอังกฤษ..... ข

กิตติกรรมประกาศ..... ค

สารบัญ..... ง

สารบัญตาราง..... ช

สารบัญรูป..... ซ

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญ..... 1

1.2 วัตถุประสงค์..... 2

1.3 ขอบเขตของโครงงาน..... 2

1.4 ขั้นตอนการดำเนินงาน..... 2

1.5 แผนการดำเนินงาน..... 3

1.6 ผลที่คาดว่าจะได้รับ..... 3

1.7 งบประมาณของโครงงาน..... 4

บทที่ 2 หลักการและทฤษฎี

2.1 ความรู้เบื้องต้นเกี่ยวกับเทคโนโลยีความจริงเสริม..... 5

2.2 ความรู้เบื้องต้นเกี่ยวกับไลบรารี ARToolKit..... 10

2.3 ความรู้เบื้องต้นเกี่ยวกับไลบรารี OpenGL 11

2.4 การหาค่าขีดแบ่ง..... 14

2.5 การติดป้ายให้ส่วนประกอบที่ติดกัน..... 15

2.6 การตรวจจับจุดตัดผ่านศูนย์..... 16

2.7 การประมาณตำแหน่ง..... 17

2.8 แมทริกซ์การแปลง (Tranformation Matrix)..... 19

2.9 การเรนเดอร์ภาพสามมิติ..... 21
2.10 ความรู้เกี่ยวกับไลบรารี OpenAL..... 22
2.11 ไฟล์เสียงรูปแบบต่าง ๆ..... 23

บทที่ 3 วิธีดำเนินโครงงาน

3.1 ศึกษารายละเอียดและหลักการ ความรู้ ทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับโครงงาน..... 26
3.2 ออกแบบการทำงานทั้งหมด..... 27
3.3 ออกแบบการทำงานของโปรแกรม..... 28
3.4 ออกแบบสัญลักษณ์และสร้างก้อนดนตรีเสมือน..... 29
3.5 ออกแบบการอ่านและค้นหาสัญลักษณ์..... 31
3.6 การออกแบบการแสดงผล..... 33
3.7 ออกแบบการส่งภาพพื้นผิว..... 35
3.8 ออกแบบโปรแกรมเพื่อปรับระดับเสียงและจังหวะความเร็วของเสียงดนตรี..... 36

บทที่ 4 ผลการทดลอง

4.1 ผลการทดลองการค้นหากรอบสัญลักษณ์ (Marker)..... 42
4.2 ผลการทดลองการคำนวณหาพิกัด 3 มิติ บนสัญลักษณ์..... 45
4.3 ผลการทดลองการค้นหาสัญลักษณ์หลายสัญลักษณ์ (Multi Marker)..... 46
4.4 ผลการทดลองเกี่ยวกับแสง..... 47
4.5 ผลการทดลองการทำภาพพื้นผิว..... 48
4.6 ผลการทดลองการปรับความดังเสียงดนตรี..... 49
4.7 ผลการทดลองการปรับความเร็วเสียงดนตรี..... 51
4.8 ผลการทดลองการปรับเอฟเฟกต์เสียงดนตรี..... 53
4.9 สรุปผลการทดลอง..... 54

บทที่ 5 บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการทดลอง..... 56
5.2 ปัญหาที่พบ..... 56
5.3 ข้อเสนอแนะและวิธีการแก้ไขปัญหา..... 57

ภาคผนวก..... 59
ประวัติผู้เขียนโครงงาน..... 67

สารบัญตาราง

ตารางที่ หน้า

1.1 ตารางแสดงระยะเวลาแผนการดำเนินงาน..... 1

3.1 ตารางแสดงค่าความดังเสียง..... 38

3.2 ตารางแสดงค่าความเร็วจังหวะเสียง..... 40

มหาวิทยาลัยนเรศวร

สารบัญรูป

รูปที่ หน้า

2.1 แสดงการทำงานของเทคโนโลยีความจริงเสริม..... 6

2.2 แสดงหลักการของเทคโนโลยีความจริงเสริม..... 7

2.3 เกมยิงรูปแบบใหม่..... 7

2.4 หนังสือ 3D 8

2.5 โปรแกรมจำลองการขับเครื่องบินรบ..... 9

2.6 แสดงการจำลองสมองก่อนการผ่าตัด..... 9

2.7 จำลองการใส่นาฬิกา..... 10

2.8 แสดงตัวอย่างการสร้างภาพความจริงเสริม..... 11

2.9 แสดงโครงสร้างของ OpenGL..... 12

2.10 แบบแม่สีหลักและแม่สีรอง..... 14

2.11 แสดงภาพการแยกภาพค่าขีดแบ่ง..... 15

2.12 แสดงการทำงานของกระบวนการการติดป้ายให้ส่วนประกอบที่ติดกัน..... 15

2.13 แสดงภาพตั้งต้น..... 16

2.14 แสดงอนุพันธ์อันดับสองของภาพตั้งต้น..... 16

2.15 ความสัมพันธ์ระหว่างกรอบพิกัดอ้างอิงของกล้อง และกรอบพิกัดอ้างอิงของสัญลักษณ์..... 17

2.16 แสดงความสัมพันธ์ระหว่างกรอบพิกัดอ้างอิงจอภาพในอุดมคติ (Ideal Screen Coordinates) และกรอบพิกัดอ้างอิงจอภาพที่สังเกตได้ (Observe Screen Coordinates)..... 18

2.17 กระบวนการคำนวณค่าประมาณค่าตำแหน่ง..... 19

2.18 แสดงความสัมพันธ์แมทริกซ์การแปลง..... 20

2.19 แสดงความสัมพันธ์ของมุมระหว่างกล้องกับสัญลักษณ์..... 21

2.20 แสดงการคำนวณหาโมเดล 3 มิติจากตำแหน่งวัตถุ..... 22

2.21 แสดงโลโก้ OpenAL..... 22

3.1 แสดงเครื่องมือที่ใช้..... 26

3.2 แสดงขั้นตอนการทำงานของ “ก้อนคนตรีเสมือน”..... 27

3.3 แสดงลำดับการทำงานของโปรแกรม..... 28

3.4 แสดงตัวอย่างสัญลักษณ์..... 29

3.5 แสดงสัญลักษณ์ถูกเก็บเป็นพิกเซล..... 30
3.6 แสดง "ก้อนดนตรีเสมือน"..... 30
3.7 แสดงหลักการอ่านและค้นหาสัญลักษณ์ (Marker)..... 31
3.8 แสดงการคำนวณหาโมเดล 3 มิติจากตำแหน่งวัตถุ..... 32
3.9 แสดงภาพก้อนดนตรีเสมือน..... 33
3.10 แสดงสถานะเสียงดนตรี..... 34
3.11 แสดงส่วนประกอบการทำ texture 35
3.12 แสดงการออกแบบการปรับระดับความเบา-ดังของเสียง..... 36
3.13 แสดงการแปลงค่าสัญลักษณ์ให้อยู่ในรูปแมทริกซ์..... 37
3.14 แสดงการออกแบบการปรับระดับความเร็ว-ช้าของจังหวะเสียงดนตรี..... 38
3.15 แสดงความสัมพันธ์องศาของกล้องกับสัญลักษณ์..... 39
3.16 แสดงการออกแบบการปรับเอฟเฟกต์เสียงดนตรี..... 41
4.1 แสดงการค้นหาวัตถุสัญลักษณ์กรอบด้านบน..... 42
4.2 แสดงการค้นหาวัตถุสัญลักษณ์กรอบด้านข้าง..... 43
4.3 แสดงการค้นหาวัตถุสัญลักษณ์กรอบด้านล่าง..... 43
4.4 แสดงการค้นหาวัตถุสัญลักษณ์ไม่เจอเมื่อมีวัตถุมาบังกรอบสี่เหลี่ยมสีดำ..... 44
4.5 แสดงการค้นหาวัตถุสัญลักษณ์ไม่เจอเมื่อมีแสงตกกระทบวัตถุมากเกินไป..... 44
4.6 แสดงการทำเรนเดอร์ภาพ..... 45
4.7 แสดงการสร้างโมเดล 3 มิติ..... 45
4.8 แสดงการค้นหาสัญลักษณ์หลายสัญลักษณ์..... 46
4.9 แสดงการจำลองความจริงเสมือนและความเป็นจริง..... 46
4.10 แสดงการใช้คำสั่งเปิด-ปิดแสง..... 47
4.11 แสดงภาพที่ต้องการทำภาพพื้นผิว ขนาดภาพ 521x256 พิกเซล..... 48
4.12 แสดงวัตถุที่ยังไม่ได้ทำการส่งภาพพื้นผิว..... 48
4.13 แสดงผลลัพธ์ที่ได้จากการทำการส่งภาพพื้นผิว..... 49
4.14 แสดงวิธีการปรับความดังเสียง..... 49
4.15 แสดงความดังเสียงดนตรีสูงสุด..... 50
4.16 แสดงความดังเสียงดนตรีเริ่มต้น..... 50
4.17 แสดงความดังเสียงดนตรีต่ำสุด..... 51
4.18 แสดงวิธีการปรับความเร็วเสียง..... 51

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.19 แสดงความเร็วสูงสุด..... 52

4.20 แสดงความเร็วเริ่มต้น..... 52

4.21 แสดงความเร็วต่ำสุด..... 53

4.22 แสดงเอฟเฟกต์เสียงปกติจะแสดงเป็นสีเขียว..... 53

4.23 แสดงเอฟเฟกต์เสียงเพี้ยนจะแสดงเป็นสีแดง..... 54

4.24 การแสดงผลทั้งหมดของโปรแกรม..... 54

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันสามารถแบ่งเครื่องดนตรีแบบ Hornbostel-Sachs ตามวิธีกำเนิดเสียงของเครื่องดนตรีเป็นหลัก ได้ 5 กลุ่ม ได้แก่ กลุ่มเครื่องกระทบ (Idiophones) กลุ่มเครื่องหนัง (Membranophones) กลุ่มเครื่องสาย (Chordophones) กลุ่มเครื่องลม (Aerophones) และกลุ่มเครื่องอิเล็กโทรนิค (Electrophones)[1] เครื่องดนตรีแต่ละชนิดมีเสียงที่เป็นเอกลักษณ์ที่แตกต่างกันออกไป และมีวิธีการใช้งานหรือการติดต่อกับนักดนตรี (User interface) แตกต่างกันไปเช่นกัน คณะผู้วิจัยจึงมีแนวคิดที่จะสร้างเครื่องดนตรีอิเล็กโทรนิคที่มีการใช้งานที่ง่ายและแปลกใหม่ขึ้นมา โดยรวมเข้ากับเทคโนโลยีความจริงเสริม (Augmented Reality)

ความจริงเสริม หรือ AR เป็นเทคโนโลยีทางด้านการโต้ตอบระหว่างมนุษย์และคอมพิวเตอร์ (Human-Computer interaction) กำลังเป็นที่น่าสนใจในวงการต่างๆ เป็นการผสานเอาโลกแห่งความจริง (reality) เข้ากับโลกเสมือน (virtual) ผ่านทางกล้อง และอุปกรณ์แสดงผล เช่น เครื่องฉายภาพหรือแว่นตาแสดงผลรวมกับการใช้งานซอฟต์แวร์ต่าง ๆ คณะผู้วิจัยมีแนวคิดจะสร้าง "ก้อนดนตรีเสมือน" ขึ้นมา ซึ่งก้อนดนตรีเสมือนนี้ถูกสร้างขึ้นมาจากการนำเอาลายพิมพ์ต่าง ๆ (AR markers) มาประกอบเป็นทรงสี่เหลี่ยมลูกบาศก์ ก้อนดนตรีเสมือนมีวิธีการสร้างและผสมเสียงด้วยการนำก้อนลูกบาศก์ส่องผ่านกล้อง เพื่อหาตำแหน่งวัตถุสัญลักษณ์ จากนั้นโปรแกรมจะทำการอ่านวัตถุสัญลักษณ์ และประมวลออกมาเป็นรูปแบบเสียงตามที่กำหนดไว้ และได้มีการนำเอาเทคโนโลยีการ สร้างภาพจำลอง 3 มิติบนตำแหน่งวัตถุสัญลักษณ์ทำงานร่วมกับการเล่นเสียงดนตรีไปพร้อมกัน ซึ่งก้อนดนตรีเสมือนหนึ่งก้อนนั้น สามารถเล่นได้ถึง 6 เสียง เสียงในก้อนดนตรีแต่ละก้อนก็จะให้เสียงที่แตกต่างกันตามจังหวะ เช่น เบส กลอง กีตาร์ และเอฟเฟคอื่น ๆ เป็นต้น อีกทั้งยังเพิ่มลูกเล่นให้สามารถเล่นเสียงพร้อมกันหลาย ๆ เสียงหรือมิกซ์เสียงได้นั่นเอง ในขณะที่เล่นเพลงอยู่นั่นยังสามารถใช้ฟังก์ชั่นการปรับเพิ่มหรือลดระดับของเสียงด้วยการหมุนปรับองศาก้อนดนตรีเสมือนไปซ้าย-ขวาตามองศาที่เปลี่ยนไป และอีกฟังก์ชั่นสามารถปรับเพิ่มหรือลดระดับความเร็วของจังหวะดนตรีได้

การพัฒนาก้อนดนตรีเสมือนนี้จุดประสงค์หลักคือเพื่อสร้างความด้านความบันเทิง นอกจากนั้นยังสามารถทำการตลาดช่วยในด้านการส่งเสริมการขายสินค้าได้อีกด้วย

1.2 วัตถุประสงค์

1. เพื่อสร้าง "ก้อนดนตรีเสมือน" (AR Music Cube Jocker) ที่สามารถเล่นเสียงและให้ความบันเทิงได้

2. เพื่อให้สามารถนำโปรแกรมไปประยุกต์ใช้ทำการตลาดในการส่งเสริมการขายสินค้าได้

1.3 ขอบเขตของโครงงาน

1. สร้าง "ก้อนดนตรีเสมือน" ที่สามารถเล่นดนตรีได้เมื่อส่องผ่านกล้อง

2. สร้าง "ก้อนดนตรีเสมือน" จำนวน 6 ก้อนที่สามารถผสมเสียงและจังหวะได้พร้อมกัน โดยจะมี 4 ก้อนที่ใช้เล่นเสียง และอีก 2 ก้อน ก้อนแรกใช้เป็นฟังก์ชั่นปรับระดับความเร็วจังหวะ และก้อนที่สองใช้เป็นฟังก์ชั่นปรับระดับความดังเสียง

3. สามารถปรับระดับความเร็ว-ช้าของจังหวะได้ผ่านทาง "ก้อนดนตรีเสมือน"

4. สามารถปรับระดับความเบา-ดังของเสียงได้ผ่านทาง "ก้อนดนตรีเสมือน"

5. ติดตั้งกล้องไว้ในแนวระดับสูงจากพื้นประมาณ 1 เมตร

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาและค้นคว้าข้อมูลเกี่ยวกับไลบรารี่ Open Framework

2. ศึกษาและค้นคว้าข้อมูลเกี่ยวกับไลบรารี่ OpenAL

3. ศึกษาและค้นคว้าข้อมูลเกี่ยวกับไลบรารี่ ARToolkit

4. ศึกษาและค้นคว้าข้อมูลเกี่ยวกับไลบรารี่ irrKlang

5. ออกแบบโปรแกรมที่สามารถเล่นเสียงดนตรีได้

6. ออกแบบโปรแกรมที่สามารถเล่นเสียงดนตรีหลายเสียงพร้อมกัน

7. เขียนโปรแกรมส่วนการรับข้อมูลจากวัตถุสัญลักษณ์ผ่านกล้อง

8. เขียนโปรแกรมส่วนการอ่านวัตถุสัญลักษณ์แล้วเล่นเสียงดนตรี

9. เขียนโปรแกรมสร้างกราฟฟิกแสดงผลที่ตำแหน่งวัตถุสัญลักษณ์

10. ทดสอบความสมบูรณ์และใช้งาน

11. สรุปผลและจัดทำรูปเล่มโครงงาน

1.5 แผนการดำเนินงาน

ตารางที่ 1.1 ตารางแสดงระยะเวลาแผนการดำเนินงาน

กิจกรรม	ปี 2554							ปี 2555		
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
ศึกษาและค้นคว้าข้อมูลเกี่ยวกับไลบรารี่ Open Framework	←	→								
ศึกษาและค้นคว้าข้อมูลเกี่ยวกับไลบรารี่ OpenAL		←	→							
ศึกษาและค้นคว้าข้อมูลเกี่ยวกับไลบรารี่ ARToolkit			←	→						
ศึกษาและค้นคว้าข้อมูลเกี่ยวกับไลบรารี่ irrKlang			←		→					
ออกแบบโปรแกรมที่สามารถเล่นเสียงดนตรีได้				←	→					
ออกแบบโปรแกรมที่สามารถเล่นเสียงดนตรีหลายเสียงพร้อมกัน					←	→				
เขียนโปรแกรมส่วนการรับข้อมูลจากวัตถุสัญลักษณ์ผ่านกล้อง					←		→			
เขียนโปรแกรมส่วนการอ่านวัตถุสัญลักษณ์แล้วเล่นเสียงดนตรี					←			→		
เขียนโปรแกรมสร้างกราฟฟิกแสดงผลที่ตำแหน่งวัตถุสัญลักษณ์							←	→		
ทดสอบความสมบูรณ์และใช้งาน									←	→
สรุปผลและจัดทำรูปเล่มโครงงาน									←	→

1.6 ผลที่คาดว่าจะได้รับ

1. "ก้อนดนตรีเสมือน" ที่สามารถเล่นดนตรีได้
5. สามารถนำโปรแกรมไปประยุกต์ใช้ทำการตลาดในการส่งเสริมการขายสินค้าได้

1.7 งบประมาณของโครงงาน

ค่าอุปกรณ์	เป็นเงิน 1,000 บาท
ค่าเอกสารทำรูปเล่ม	เป็นเงิน 1,000 บาท
รวม	เป็นเงิน 2,000 บาท
	(สองพันบาทถ้วน)

หมายเหตุ ถัวเฉลี่ยทุกรายการ

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ความรู้เบื้องต้นเกี่ยวกับเทคโนโลยีความจริงเสริม

ปัจจุบันเทคโนโลยีความจริงเสริม (Augmented Reality : AR) ถูกนำมาใช้กับธุรกิจต่าง ๆ เช่น อุตสาหกรรม การแพทย์ การตลาด การบันเทิง การสื่อสาร โดยใช้เทคโนโลยีความจริงเสมือนมาผนวกเข้ากับเทคโนโลยีภาพผ่านซอฟต์แวร์และอุปกรณ์เชื่อมต่อต่างๆ และแสดงผลผ่านหน้าจอคอมพิวเตอร์หรือบนหน้าจอโทรศัพท์มือถือ ทำให้ผู้ใช้สามารถนำเทคโนโลยีเสมือนจริงมาใช้กับการทำงานแบบออนไลน์ที่สามารถโต้ตอบได้ทันทีระหว่างผู้ใช้กับสินค้าหรืออุปกรณ์ต่อเชื่อมแบบเสมือนจริงของโมเดลแบบสามมิติ โดยผู้ใช้ไม่จำเป็นต้องไปสถานที่จริง[12]

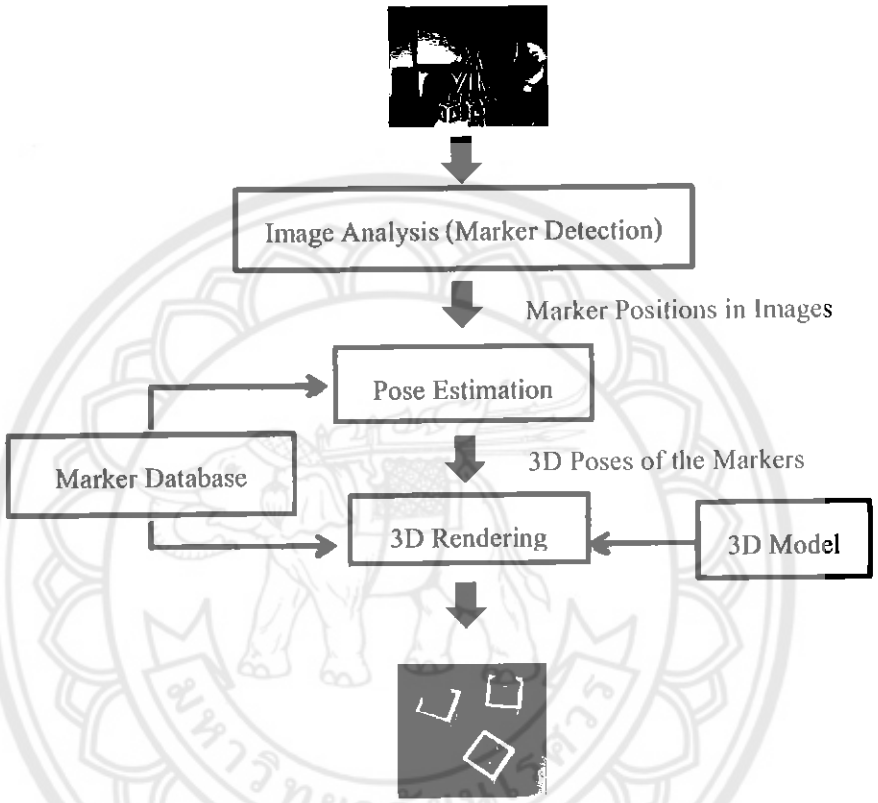
2.1.1 แนวคิดหลักของเทคโนโลยีความจริงเสริม

คือการพัฒนาเทคโนโลยีที่ผสานเอาโลกแห่งความเป็นจริง (Reality) และความเสมือนจริง (Virtual) เข้าด้วยกัน ผ่านซอฟต์แวร์และอุปกรณ์เชื่อมต่อต่างๆ เช่น เว็บแคมคอมพิวเตอร์ หรืออุปกรณ์อื่นที่เกี่ยวข้อง ซึ่งภาพเสมือนจริงนั้นจะแสดงผลผ่านหน้าจอคอมพิวเตอร์บนเครื่องฉายภาพหรือบนอุปกรณ์แสดงผลอื่นๆ โดยภาพเสมือนจริงที่ปรากฏขึ้นจะมีปฏิสัมพันธ์ (Interactive) กับผู้ชมได้ทันที อาจมีลักษณะทั้งที่เป็นภาพนิ่งสามมิติ ภาพเคลื่อนไหว หรืออาจจะเป็นสื่อที่มีเสียงประกอบทั้งนี้ขึ้นอยู่กับ การออกแบบสื่อแต่ละรูปแบบว่าให้ออกมาแบบใด พื้นฐานหลักของเทคโนโลยีความจริงเสริมที่ดีจำเป็นที่ต้องรวบรวมหลักการของการตรวจจับความเคลื่อนไหว (Motion Detection) การตรวจจับบีต (Beat Detection) และการรู้จำเสียงพูดและการประมวลผลภาพ (Voice Recognize and Image Processing) นอกจากการตรวจจับการเคลื่อนไหวแล้ว การตอบสนองบางอย่างของระบบผ่านสื่อนั้น ต้องมีการตรวจจับเสียงของผู้ใช้บริการและประมวลผลด้วยหลักการตรวจจับบีต เพื่อเกิดจังหวะการสร้างทางเลือกให้แก่ระบบ เช่น เสียงในการสั่งให้สื่อเชิงโต้ตอบ (Interactive Media) ทำอะไรต่อไป โดยกระบวนการภายในของเทคโนโลยีความจริงเสริมประกอบด้วย 3 กระบวนการ ได้แก่

การวิเคราะห์ภาพ (Image Analysis) เป็นขั้นตอนการค้นหาสัญลักษณ์ (Marker) จากภาพที่ได้จากกล้องแล้วสืบค้นจากฐานข้อมูล (Marker Database) ที่มีการเก็บข้อมูลขนาดและรูปแบบของสัญลักษณ์ไว้

การคำนวณค่าตำแหน่งเชิง 3 มิติ (Pose Estimation) ของสัญลักษณ์เพื่อนำมาวิเคราะห์รูปแบบของสัญลักษณ์เทียบกับกล้อง

กระบวนการสร้างภาพ 2 มิติ จากโมเดล 3 มิติ (3D Rendering) เป็นการเพิ่มข้อมูลเข้าไปในภาพ โดยใช้ค่าตำแหน่งเชิง 3 มิติ ที่คำนวณได้จนได้ภาพเสมือนจริง[12] ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 แสดงการทำงานของเทคโนโลยีความจริงเสริม

ที่มา: http://msmisthammasat.blogspot.com/2011/01/augmented-reality.html

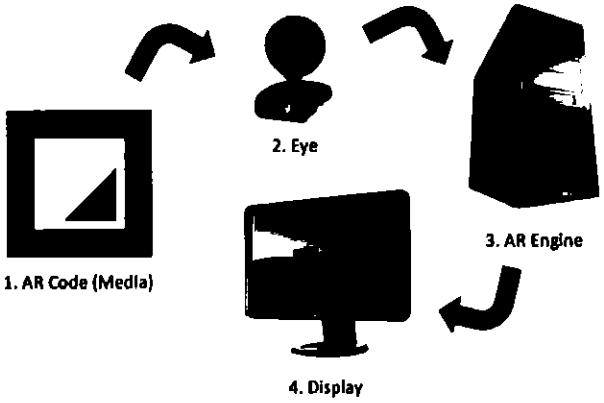
2.1.2 หลักการของเทคโนโลยีความจริงเสริม ประกอบด้วย 4 ส่วน ดังรูปที่ 2.2

1. สัญลักษณ์ (Marker หรือ AR code) ใช้ในการกำหนดตำแหน่งของวัตถุ

2. อุปกรณ์รับภาพ (Image input device) หรือกล้องเว็บแคม ใช้มองตำแหน่งสัญลักษณ์แล้วส่งเข้าส่วนประมวลผลความจริงเสริม (AR Engine)

3. ส่วนประมวลผลความจริงเสริม (AR Engine) ทำหน้าที่อ่านข้อมูลจากกล้องเข้ามาประมวลผลเพื่อคำนวณตำแหน่งของสัญลักษณ์

4. อุปกรณ์แสดงผล (Display device) จอแสดงผลข้อมูลที่ผ่านการประมวลผลแล้วส่งมาให้ในรูปแบบของวีดีโอ



รูปที่ 2.2 แสดงหลักการของเทคโนโลยีความจริงเสริม

ที่มา: http://learngears.com/2011/08/10/เทคโนโลยี-augmented-reality-**คืออะไร**

2.1.3 เทคโนโลยีความจริงเสริมในปัจจุบันได้เข้ามีบทบาทให้หลายด้าน ดังนี้

1. ด้านการเกมและความบันเทิง โลกเสมือนผสานโลกจริงสามารถนำมาใช้เพื่อเสริมสร้างการเล่นเกมและความบันเทิง โดยเฉพาะเกมที่มีรูปแบบเล่นตามบทบาท หรือ เกมอาร์พีจี (Role-playing game: RPG) ซึ่งในอนาคตสามารถนำไปรวมกับระบบโลกเสมือนผสานโลกจริง เพื่อให้ผู้เล่นมีความรู้เสมือนอยู่ในสภาพแวดล้อมจริง ผู้เล่นเกิดความรู้สึกเป็นส่วนหนึ่งในเกมและความบันเทิงรูปแบบต่างๆ ได้อย่างเสมือนจริง รับรู้ได้ว่าเป็นส่วนหนึ่งหรือเป็นตัวละครในเกม สำหรับด้านความบันเทิง ระบบความจริงเสมือนผสานกับโลกจริงสนับสนุนการนำเสนอสินค้า การแสดงละคร การโต้ตอบ ขององค์กรธุรกิจ[13] ตัวอย่างดังรูปที่ 2.3

รูปที่ 2.3 เกมยิงรูปแบบใหม่

ที่มา: http://www.apptoyz.com/products

2. ด้านการศึกษา โลกเสมือนผสานโลกจริง สามารถนำมาประยุกต์ใช้ร่วมกับเทคโนโลยีอื่น ๆ ที่เกี่ยวข้องในการศึกษา ให้ข้อมูลสาระที่ด้านการศึกษากับผู้เรียนได้ทันที ผู้เรียนได้สัมผัสประสบการณ์ใหม่ในมิติที่เสมือนจริง ผู้เรียนเกิดกระบวนการร่วมกันเรียนรู้ ครูผู้สอนเสริมสร้างความรู้ของผู้เรียนผ่านการสาธิต การสนทนา รูปแบบการเรียนรู้จะปรับเปลี่ยนเป็นโลกเสมือนผสานโลกจริงมากขึ้นสงเสริมให้ผู้เรียนเข้าใจลึกซึ้งในสิ่งที่ต้องการเรียนรู้

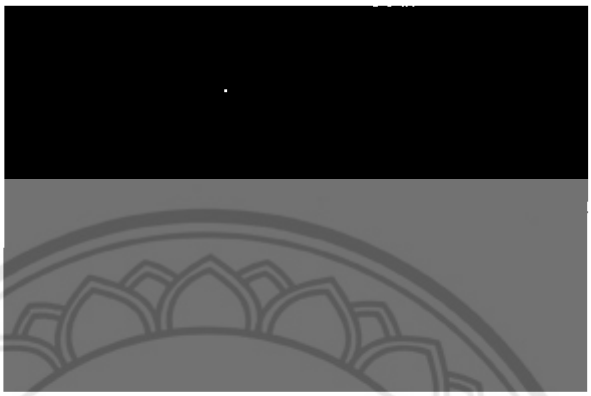
สถานศึกษา นักการศึกษา ผู้สอนจะเป็นจุดเริ่มต้นสำคัญในการนำเทคโนโลยีเสมือนผสานโลกจริงมาใช้เพื่อให้ผู้เรียนได้รับมีประสบการณ์มีความหมายลึกมากขึ้นโดยการเชื่อมโยงเนื้อหาที่ได้เรียนรู้กับสถานที่หรือวัตถุที่เฉพาะเจาะจงเหมาะสมกับเนื้อหาที่เรียนรู้ด้วยภาพสามมิติโดยการผนวกเข้ากับการเรียนรู้แบบสำรวจด้วยเทคโนโลยีมือถือและอุปกรณ์สมัยใหม่ ที่ทำให้การเรียนสามารถจะขยายออกหรือย้ายการเรียนรู้สู่นอกห้องเรียนมากขึ้น ส่งเสริมการเรียนรู้จากรูปแบบเดิม และในบางกรณีเทคโนโลยีเสมือนผสานโลกจริงสามารถผนวกเข้ากับรูปแบบการเรียนรู้อื่นๆ เข้าไป เช่นการนำมาใช้กับเกมการศึกษา นำมาใช้กับกิจกรรมสงเสริมการทำงานเป็นทีม และนำมาใช้การเรียนรู้แบบท้าทาย เป็นต้น[13] ตัวอย่างดังรูปที่ 2.4

รูปที่ 2.4 หนังสือ 3D

ที่มา: http://www.amazon.com/Wonderful-Wizard-Oz-Commemorative-Pop-up/dp/0689817517

3. การรักษาความปลอดภัยและการป้องกันประเทศ การนำเทคโนโลยีโลกเสมือนผสานโลกจริงมาใช้งานด้านการทหาร โดยเฉพาะอย่างยิ่ง Office of Naval Research and Defense Advanced Research Projects Agency หรือ DARPA ประเทศสหรัฐอเมริกา เป็นกลุ่มผู้บุกเบิกระบบโลกเสมือนผสานโลกจริง นำมาใช้ในการฝึกให้กับทหาร ให้เกิดการเรียนรู้เกี่ยวกับสภาพแวดล้อมของพื้นที่ ข้อมูลต่างๆ ในพื้นที่รบ สามารถนำมาใช้ฝึกการเคลื่อนไหวของกองกำลังและวาง

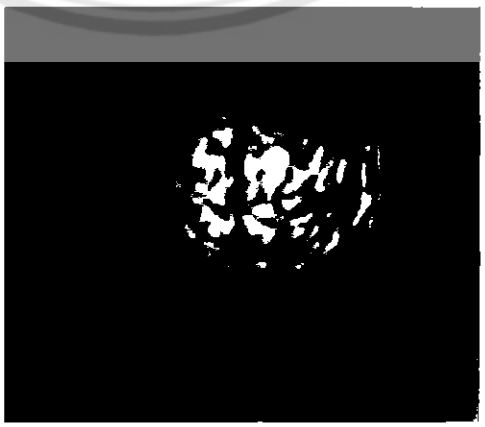
แผนการเคลื่อนกำลังของทหารในฝ่ายเดียวกันและศัตรูในพื้นที่สงครามเสมือนจริง และโลกเสมือนผสานโลกจริง ยังมีบทบาทสำคัญในการบังคับใช้กฎหมายและหน่วยงานข่าวกรอง ระบบจะช่วยให้เจ้าหน้าที่ตำรวจ สามารถสร้างมุมมองที่สมบูรณ์ในรายละเอียดเกี่ยวกับข้อมูลพื้นที่ลาดตระเวน[13] ตัวอย่างดังรูปที่ 2.5



รูปที่ 2.5 โปรแกรมจำลองการขับเครื่องบินรบ

ที่มา: http://flightgear.warbricktech.com/Gallery-v1.0

4. ทางการแพทย์ ระบบโลกเสมือนผสานโลกจริงสามารถนำมาใช้ทางด้านศัลยแพทย์ทางระบบประสาทสัมผัสการรับรู้ ส่งผลให้การดำเนินการการผ่าตัดที่มีความเสี่ยงเกิดขึ้นน้อยลงได้ และมีประสิทธิภาพมากขึ้น ระบบโลกเสมือนผสานโลกจริงยังสามารถใช้ร่วมกับอุปกรณ์ทางการแพทย์อื่น ๆ เช่น เครื่อง X - ray หรือ MRI เพื่อให้แพทย์ได้วินิจฉัยทางการแพทย์หรือการตัดสินใจสมบูรณ์มากขึ้น[13] ตัวอย่างดังรูปที่ 2.6



รูปที่ 2.6 แสดงการจำลองสมองก่อนการผ่าตัด

ที่มา: http://www.atre.t.u-tokyo.ac.jp/en/content/view/57/86

5. ทางด้านธุรกิจ สามารถนำเทคโนโลยีโลกเสมือนผสานโลกจริงมาใช้ในงานที่เกี่ยวกับการส่งเสริมการขายสินค้า ปัจจุบันมีสินค้าหลาย ๆ ชนิดได้นำมาประยุกต์ใช้ เช่น ธุรกิจด้านเครื่องประดับ ได้นำเทคโนโลยีนี้มาช่วยให้ลูกค้าสามารถทดลองใส่สินค้าได้[13] ตัวอย่างดังรูปที่ 2.7

รูปที่ 2.7 จำลองการใส่นาฬิกา

ที่มา: http://www.tlcthai.com/technology/new-product/1637.html

2.2 ความรู้เบื้องต้นเกี่ยวกับไลบรารี ARToolKit

ARToolKit เป็นไลบรารี (library) ที่ช่วยในการสร้างเทคโนโลยีความจริงเสริมและเครื่องมือช่วยวาดวัตถุสามมิติ ประกอบฉากภาพที่ได้เกิดจากการรวมภาพที่ถ่ายจากกล้องเข้ากับโมเดลสามมิติ ที่ต้องการโดยมีการจัดทิศทางของวัตถุให้สอดคล้องกับมุมของกล้องทำให้ดูเสมือนราวกับว่ามีวัตถุอยู่ในบริเวณนั้นจริงๆ อุปกรณ์ที่จำเป็นคือ 1. กล้อง 2. โปรแกรม 3.เครื่องพิมพ์เอาไว้พิมพ์ลวดลายลงบนกระดาษให้กล้องตรวจจับหลักการทำงานของโปรแกรมคือเมื่อภาพลวดลายเกิดการขยับโปรแกรมจะคำนวนหามุมที่เปลี่ยนไปแล้วนำมาใช้อ้างอิงซ้อนวัตถุสามมิติลงในภาพอีกที[5]

AR-ToolKit เป็นชุดพัฒนาซอฟแวร์ (SDK) ที่พัฒนาขึ้นโดยภาษา C และ C++ เพื่อใช้เป็นแก่นในการพัฒนาโปรแกรมประยุกต์ทางด้านเทคโนโลยีความจริงเสริม โดยอ้างอิงเนื้อหาจากหัวข้อ 2.1.1 (หลักการของเทคโนโลยีความจริงเสริม) เครื่องมือ ARToolKit จะมีฟังก์ชั่นสำหรับส่วนของการวิเคราะห์ภาพ (Image Analysis) และ ส่วนคำนวณตำแหน่งเชิง 3 มิติ (Pose Estimation)

ที่ค่อนข้างสมบูรณ์ อย่างไรก็ตาม ARToolKit ยังมีฟังก์ชั่นที่ใช้สำหรับการสร้างภาพจากโมเดล 3 มิติ (3D Rendering) ในระดับพื้นฐานซึ่งอิงอยู่กับไลบรารี OpenGL [11]

ส่วนที่ยากของการพัฒนาโปรแกรมประยุกต์เทคโนโลยีความจริงเสริม คือความแม่นยำในการคำนวณตำแหน่งของวัตถุในสิ่งแวดล้อมจริง (Image Analysis และ Pose Estimation) เพื่อเพิ่มภาพที่สร้างจากคอมพิวเตอร์กราฟิกลงไปในตำแหน่งเดียวกันตามมุมมองของผู้ใช้ในแบบเวลาจริง (Real time) ซึ่งใน ARToolKit นี้ได้ใช้เทคนิคทางทัศนศาสตร์คอมพิวเตอร์ (Computer Vistion) ในการคำนวณตำแหน่งของกล้องเทียบกับตำแหน่งของสัญลักษณ์ เพื่อช่วยให้โปรแกรมเมอร์ง่ายต่อการนำภาพที่สร้างจากคอมพิวเตอร์กราฟิกใส่ลงไปยังตำแหน่งที่คำนวณได้ นอกจากนี้ ARToolKit มีส่วนของ การติดตามสัญลักษณ์ (Marker Tracking) ที่รวดเร็วและแม่นยำทำให้เกิดการพัฒนาโปรแกรมประยุกต์เทคโนโลยีความจริงเสริมใหม่ๆที่น่าสนใจอย่างแพร่หลาย[11] ตัวอย่างดังรูปที่ 2.8

รูปที่ 2.8 แสดงตัวอย่างการสร้างภาพความจริงเสริม

ที่มา: http://www.hitl.washington.edu/artoolkit

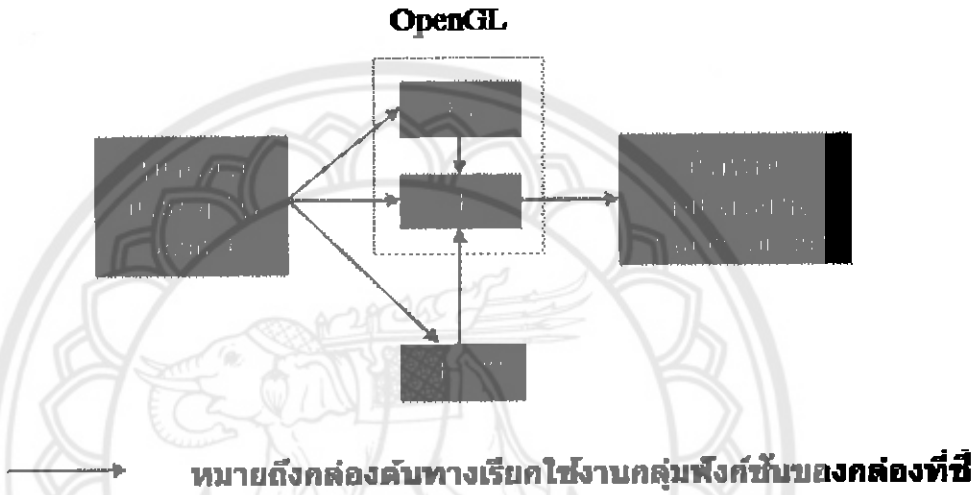
2.3 ความรู้เบื้องต้นเกี่ยวกับไลบรารี OpenGL

OpenGL เป็นโปรแกรมมาตรฐานที่เกี่ยวข้องกับการสร้างงานกราฟฟิกโดยเฉพาะอย่างยิ่งกราฟฟิกในรูปแบบของ 3 มิติ ในปัจจุบันนี้ OpenGL ได้รับการยอมรับให้เป็นมาตรฐานของโปรแกรมทางกราฟฟิกที่ใช้กันทั่วไป

สำหรับรูปแบบการใช้งานของ OpenGL จะคล้ายคลึงกับการใช้งาน DirectX3D ของไมโครซอฟต์ สำหรับผู้ที่สนใจสามารถประยุกต์การใช้งานจาก OpenGL ไปเป็น DirectX โดยใช้เวลาในการศึกษาไม่มากนัก[6]

2.3.1 องค์ประกอบต่างๆของ OpenGL

องค์ประกอบโดยรวมของ OpenGL จะประกอบด้วยไลบรารีกลุ่มใหญ่ๆอยู่ 3 กลุ่ม สำหรับการเรียกใช้งานอันได้แก่ GL GLU และ GLUT (สร้างขึ้นภายหลังโดย Mark J. Gilard เพื่อให้การเรียกใช้งานคล่องตัวขึ้น) ส่วนองค์ประกอบอื่นๆที่ได้เพิ่มขึ้นมาเพื่อให้เหมาะสมกับภาษาที่ใช้เขียนเช่น FGL ใช้กับภาษา FORTRAN หรือ WGL ใช้กับวินโดวส์เป็นต้น[6] โดยรูปแบบการทำงานแสดงได้ดังรูปที่ 2.9



รูปที่ 2.9 แสดงโครงสร้างของ OpenGL

ที่มา: http://202.44.14.219/~isurnich/ftp/CG/html/OpenGL.htm

2.3.2 โครงสร้างโปรแกรมกราฟฟิกโดยใช้ OpenGL

ในการเขียนโปรแกรมกราฟฟิกจะมีลักษณะที่คล้ายๆกันสำหรับ OpenGL ในที่นี้จะใช้วิธีการเขียนโปรแกรมด้วยภาษา C/C++ รูปแบบของโปรแกรมที่ง่ายต่อการเขียนใช้งานจะเป็นในลักษณะของโปรแกรมแบบคอนโซลหรือโปรแกรมที่ยังอิงอยู่กับระบบปฏิบัติการของดอส (DOS) โดยโปรแกรมจะแบ่งออกเป็นชนิดของฟังก์ชันต่างๆดังนี้

ฟังก์ชันทำงาน

เป็นฟังก์ชันที่ผู้เขียนสร้างขึ้นเพื่อให้ทำงานตามที่ผู้เขียนกำหนด บางฟังก์ชันที่สร้างขึ้นในลักษณะนี้จะเกี่ยวข้องกับการเขียนกราฟฟิก ผู้เขียนจะต้องนำฟังก์ชันดังกล่าวไปขึ้นทะเบียนกับ OpenGL เพื่อให้มันทำงานในทุกๆรอบการทำงานในรูปแบบของฟังก์ชันที่ถูกเรียกกลับ (callback function) ฟังก์ชันการทำงานที่สำคัญมีได้ดังนี้

main – เป็นฟังก์ชันหลักซึ่งเขียนโดยผู้เขียนโปรแกรมมักใช้กำหนดค่าเริ่มต้นต่างๆ และเป็นที่อยู่ของฟังก์ชันสำหรับขึ้นทะเบียนฟังก์ชันเรียกกลับ (callback function)และท้ายที่สุดจะต้อง

ตามด้วยฟังก์ชัน glutMainloop(); เพื่อที่จะให้โปรแกรมกราฟฟิกมีการตรวจสอบการทำงานของฟังก์ชันเรียกกลับที่ขึ้นทะเบียนตลอดเวลา

display – เป็นฟังก์ชันที่ใช้สำหรับการนิยามวัตถุ การแปลงวัตถุและมุมมองของกราฟฟิก สำหรับการใช้ฟังก์ชัน display จะต้องขึ้นทะเบียนเป็นฟังก์ชันเรียกกลับในฟังก์ชัน main โดยเรียกใช้ ฟังก์ชัน glutDisplayFunc(); เช่น glutDisplayFunc(display);

reshape – ฟังก์ชันที่เกี่ยวข้องกับการปรับปรุงหรือวินโดวส์มีการเปลี่ยนแปลงขนาดอันเนื่องมาจากผู้ใช้ขยายหรือหดหน้าต่างของ OpenGL และการฉายภาพจากสามมิติเป็นสองมิติ ในการเรียกใช้ฟังก์ชันจะต้องขึ้นทะเบียนด้วยฟังก์ชัน glutReshapeFunc(); ตัวอย่างเช่นฟังก์ชันที่จัดการเกี่ยวกับงานดังกล่าวผู้ใช้ตั้งชื่อว่า reshape การขึ้นทะเบียนจะเป็นดังนี้ glutReshapeFunc(reshape);

นอกจากนี้แล้วยังมีฟังก์ชันเรียกกลับสำหรับการโต้ตอบกับผู้ใช้เช่น glutMouseFunc(), glutKeyboardFunc(), glutIdleFunc() เป็นต้น ซึ่งสามารถหารดูรายละเอียดได้ที่คู่มือของ OpenGL

ฟังก์ชันเรียกกลับ (callback function)

เป็นฟังก์ชันที่ใช้สำหรับการขึ้นทะเบียนของฟังก์ชันทำงาน ฟังก์ชันเรียกกลับเป็นฟังก์ชันของ GLUT และมักจะลงท้ายด้วยคำว่า "Func" เช่น glutReshapeFunc, glutDisplayFunc โดยที่พารามิเตอร์ของมันจะเป็นฟังก์ชันการทำงานที่สอดคล้องกับการทำงานของมันที่ควรเป็นฟังก์ชันเรียกกลับนี้ใช้สำหรับตรวจจับเหตุการณ์ที่กำหนดใน OpenGL ถ้าเกิดเหตุการณ์ดังกล่าวมันจะมุ่งตรงไปยังฟังก์ชันที่สอดคล้องกับเหตุการณ์นั้นๆ เช่น glutMouseFunc(mouse) จะเป็นฟังก์ชันเรียกกลับสำหรับเหตุการณ์การเปลี่ยนแปลงของเมาส์ เช่นถ้าเมาส์มีการกดปุ่ม การควบคุมของโปรแกรมจะตรงไปยังฟังก์ชันmouse ซึ่งเป็นชื่อของพารามิเตอร์ของมัน

ฟังก์ชันกำหนดการวนรอบ

ได้แก่ฟังก์ชัน glutMainLoop() สำหรับกราฟฟิกจะเป็นการทำงานที่จะต้องทำการเขียนซ้ำอยู่ตลอดเวลาเพื่อที่จะให้วัตถุปรากฏที่หน้าจอ ดังนั้นฟังก์ชัน glutMainLoope จะทำหน้าที่ในการสร้างรอบวนดังกล่าว เสมือนกับว่าฟังก์ชันต่างๆถูกกำหนดอยู่ภายใต้รอบวนของ while(1) ซึ่งมันจะทำงานภายในรอบวนตลอดจนกว่าจะปิดโปรแกรม[6]

2.3.3 สี RGB

RGB ย่อมาจาก red, green และ blue คือระบบสีของแสง เกิดจากการหักเหของแสงกลายเป็นสีรุ้ง ด้วยกัน 7 สี ซึ่งเป็นช่วงแสงที่ตาของคนเราสามารถมองเห็นได้ แสงสีม่วงจะมีความถี่สูงสุดเรียกว่า อุลตร้าไวโอแรต และแสงสีแดงจะมีความถี่ต่ำสุด เรียกว่าอินฟาเรต คลื่นแสงที่มีความถี่สูงกว่าสีม่วง และต่ำกว่าสีแดงนั้น สายตาของมนุษย์ไม่สามารถรับได้ แสงสีทั้งหมดเกิดจากแสงสี 3 สี คือ สีแดง (Red) สีน้ำเงิน (Blue) และสีเขียว (Green) ทั้งสามสีถือเป็นแม่สีของแสง

แม่สีของแสงมีด้วยกัน 3 สี คือ สีแดง (Red) ,สีเขียว(G),สีน้ำเงิน(B) และแต่ละแม่สีเมื่อรวมกันก็จะได้สีดังนี้

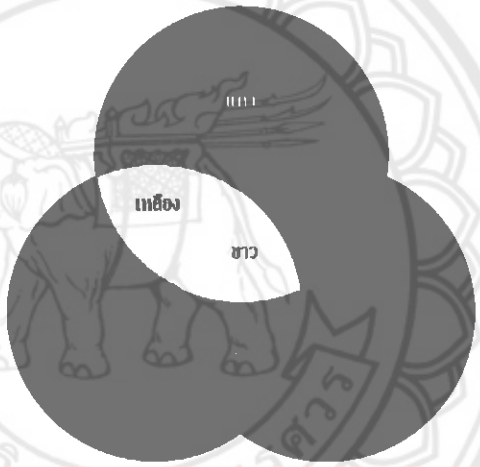
สีแดง+สีเขียว ได้ สีเหลือง (Yellow)

สีเขียว+น้ำเงิน ได้ สีฟ้า (Cyan)

สีแดง+สีน้ำเงิน ได้ สีแดงอมชมพู่ (Magenta)

เมื่อนำแม่สีของแสงทั้ง 3 มาผสมกัน ในปริมาณแสงสว่างเท่ากันก็จะได้เป็นแสงที่สีขาว แต่ถ้าผสมกันระหว่างแสงระดับความสว่างต่างกัน ก็จะได้ผลทีเป็นแสงสีๆ มากมายเป็นล้านสี ทีเดียว ส่วนใหญ่การใช้สีลักษณะนี้จะใช้ในอุปกรณ์ที่เกี่ยวกับแสง เช่น จอภาพ กล้องดิจิตอล สแกนเนอร์ เป็นต้น

ระบบสี RGB จะการแสดงผลออกมา เป็นรูปแบบการรับแสงแสดงผลด้วยแสงทีเป็นแม่สีได้แก่ สีแดง สีเขียว สีน้ำเงิน ซึ่งอุปกรณ์อิเล็กทรอนิกส์ต่างๆไม่ว่าจะเป็น จอภาพ,สแกนเนอร์, กล้องดิจิตอลหรือดวงตาคนเราล้วนแต่รับและแปลผลเป็นสีต่างๆ ด้วยแสงเหล่านี้[14] ดังรูปที่ 2.10



รูปที่ 2.10 แบบแม่สีหลักและแม่สีรอง

ที่มา: http://www.klongdigital.com/photoshop/photoshop12

2.4 การหาค่าขีดแบ่ง

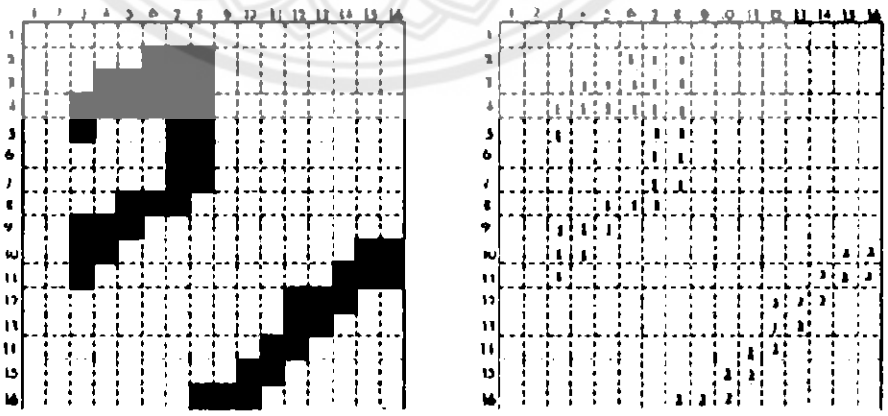
การแยกบริเวณโดยการใช้ค่าขีดแบ่ง (Threshold) ค่าขีดแบ่งเป็นค่าที่เป็นจำนวนเต็มที่มีค่าระหว่าง 0 ถึง 255 เช่นเดียวกับค่าความเข้มแสงของพิกเซลที่อยู่ในภาพระดับเทา (Gray scale) สำหรับการแยกบริเวณโดยการใช้ค่าขีดแบ่งนั้น จะเป็นการแปลงภาพระดับเทาให้เปลี่ยนเป็นภาพที่มีเพียงสองระดับ (binary image) โดยการใช้เงื่อนไขว่าถ้าค่าความเข้มแสงที่พิกเซลตำแหน่งใดมีค่าต่ำกว่าหรือเท่ากับค่าขีดแบ่งให้ค่าพิกเซลในตำแหน่งนั้นมีค่าเป็น 0 หรือเปลี่ยนเป็นด้านมืดไป และถ้าพิกเซลใดมีค่าสูงกว่าค่าขีดแบ่งแล้วให้พิกเซลนั้นมีค่าเป็นค่า 255 หรือเปลี่ยนเป็นด้านสว่างไป [15] ตัวอย่างดังรูปที่ 2.11

รูปที่ 2.11 แสดงภาพการแยกภาพค่าขีดแบ่ง

ที่มา: http://matlab.exponenta.ru/imageprocess/book2/80.php

2.5 การติดป้ายให้ส่วนประกอบที่ติดกัน

การพิจารณาว่าพิกเซลใดบ้างที่มีการ "เชื่อมต่อ (connect)" กัน เพื่อที่จะจัดให้พิกเซลเหล่านั้นให้อยู่ในบริเวณหรือวัตถุเดียวกัน กระบวนการย่อยนี้เรียกว่าการติดป้ายให้ส่วนประกอบที่ติดกัน (Connected components labeling) ซึ่งผลที่ได้จากการทำงานของกระบวนการย่อยนี้ คือ จะทำให้รู้ว่าพิกเซลในแต่ละตำแหน่งนั้น จัดเป็นของวัตถุชิ้นใด ดังแสดงไว้ในรูปที่ ซึ่งแสดงหมายเลขของชิ้นวัตถุที่พิกเซลตำแหน่งนั้น ๆ เป็น "สมาชิก" อยู่[15] ตัวอย่างดังรูปที่ 2.12



รูปที่ 2.12 แสดงการทำงานของกระบวนการการติดป้ายให้ส่วนประกอบที่ติดกัน

ที่มา: cdn.researchers.in.th/assets/media/files/000/.../original_chapter01.pdf

2.6 การตรวจจับจุดตัดผ่านศูนย์

การตรวจจับจุดผ่านศูนย์ (Zero crossing Detection) คือ ค่าที่ลาปาเซียนผ่านศูนย์ เป็นการหาอนุพันธ์อันดับสองของภาพเพื่อให้ได้ขอบของภาพ โดยภาพที่ผ่านการหาอนุพันธ์อันดับสอง (Laplacian Operator($\nabla^2 P$)) บริเวณที่เป็น ส่วนขอบจะเด่นชัดขึ้น การตรวจจับจุดตัดผ่านศูนย์ทำได้โดยแบ่งด้วยค่าขีดแบ่ง (thresholding) กับ ของ $\nabla^2 P$ (ตำแหน่งที่พิกเซลของ $\nabla^2 P$ เปลี่ยนจาก + เป็น – หรือจาก – เป็น +)

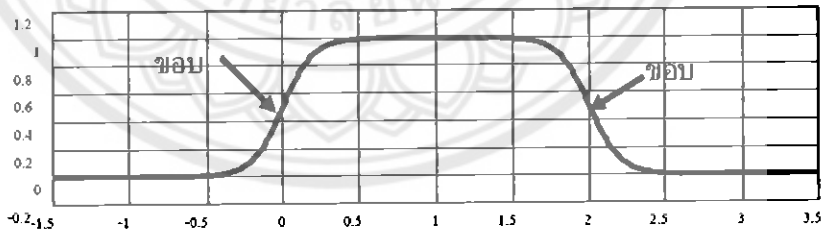
$$Mask(E_x) = \begin{bmatrix} Z_{x1} & Z_{x2} & Z_{x3} \\ Z_{x4} & Z_{x5} & Z_{x6} \\ Z_{x7} & Z_{x8} & Z_{x9} \end{bmatrix} = E_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$Mask(E_y) = \begin{bmatrix} Z_{y1} & Z_{y2} & Z_{y3} \\ Z_{y4} & Z_{y5} & Z_{y6} \\ Z_{y7} & Z_{y8} & Z_{y9} \end{bmatrix} = E_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

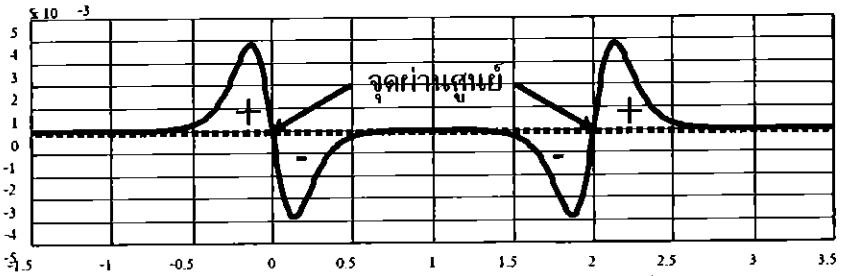
ทำให้ตำแหน่งของบริเวณขอบของวัตถุในภาพคือค่าจุดผ่านศูนย์ (Zero Crossing) ของ $\nabla^2 P$ การหาอนุพันธ์อันดับสอง สามารถหาได้จากสมการ

$$\nabla^2 P = \frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2}$$

การค้นหาขอบภาพอนุพันธ์อันดับสอง เมื่อทำ $\nabla^2 P$ บริเวณขอบจะมีตำแหน่งเดียวกับค่าจุดผ่านศูนย์ของค่าจาก $\nabla^2 P$ พิจารณาได้จากตำแหน่งที่พิกเซลเปลี่ยนแปลงจากค่าที่เป็นบวกเป็นค่าที่เป็นลบหรือจากค่าที่เป็นลบไปเป็นค่าที่เป็นบวก ดังรูปที่ 2.13 และ 2.14



รูปที่ 2.13 แสดงภาพตั้งต้น



รูปที่ 2.14 แสดงอนุพันธ์อันดับสองของภาพตั้งต้น

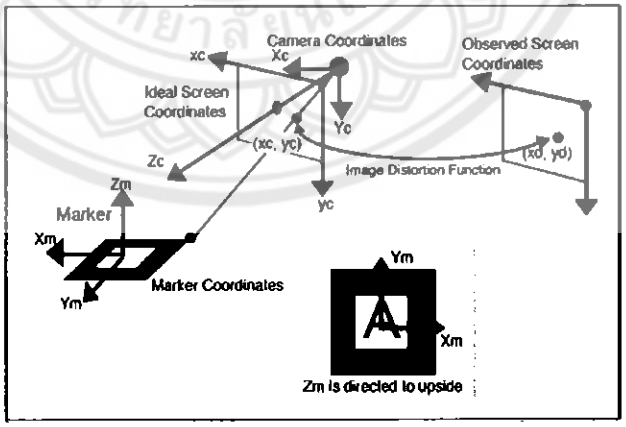
การหาขอบโดยใช้อนุพันธ์อันดับสอง ไม่สนใจทิศทางของภาพในแนวแกน x และ แกน y กำหนดจุดที่ค่า y เป็นจุดผ่านศูนย์ วิธีนี้ใช้เวลาในการคำนวณมากกว่าการค้นหาขอบโดยใช้อนุพันธ์อันดับหนึ่ง กล่าวคือเราสามารถตรวจจับความไม่ต่อเนื่องของพิกเซลในโดเมนรูปภาพได้โดยใช้อนุพันธ์ของภาพนั้นเอง

2.7 การประมาณตำแหน่ง

การประมาณตำแหน่ง (Pose Estimation) เป็นขั้นตอนของการคำนวณค่าตำแหน่งเชิง 3 มิติ (3D Pose) ของสัญลักษณ์เมื่อเทียบกับกล้องวีดีโอ ค่านี้จะถูกแสดงในรูปเมตริกซ์ขนาด 4x4 (TCM) ที่ระบุความสัมพันธ์ระหว่างกรอบพิกัดอ้างอิงของกล้อง (Camera Coordinated Frame) และกรอบพิกัดอ้างอิงของสัญลักษณ์ ดังสมการ

$$\begin{bmatrix} X_C \\ Y_C \\ Z_X \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_M \\ Y_M \\ Z_M \\ 1 \end{bmatrix} = T_{CM} \begin{bmatrix} X_M \\ Y_M \\ Z_M \\ 1 \end{bmatrix}$$

ซึ่งกรอบพิกัดอ้างอิงของกล้อง คือกรอบพิกัดอ้างอิงที่ใช้อ้างอิงตำแหน่งใด ๆ ของกล้องวีดีโอ และกรอบพิกัดอ้างอิงของสัญลักษณ์ คือกรอบพิกัดอ้างอิงที่ใช้อ้างอิงตำแหน่งใดๆของสัญลักษณ์ซึ่งสามารถแสดงดังรูปที่ 2.15



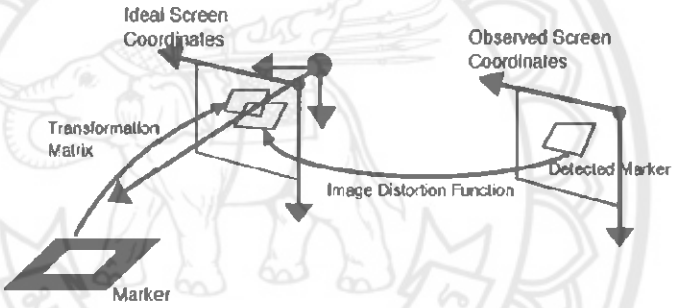
รูปที่ 2.15 ความสัมพันธ์ระหว่างกรอบพิกัดอ้างอิงของกล้อง และกรอบพิกัดอ้างอิงของสัญลักษณ์

ที่มา: http://www.hitl.washington.edu/artoolkit/documentation/cs.htm

จากรูปที่ 2.16 ความสัมพันธ์ระหว่างจุดใดๆ (X_c Y_c Z_c) บนกรอบพิกัดอ้างอิงของกล้องกับจุดที่ตรงกัน (x_I y_I) ในกรอบพิกัดอ้างอิงจอภาพในอุดมคติ (Ideal Screen Coordinated Frame) เป็นไปตามการฉายภาพแบบทัศนมิติ (Perspective Projection) ดังสมการ

$$\begin{bmatrix} hx_I \\ hy_I \\ h \end{bmatrix} = \begin{bmatrix} sf_x & 0 & x_c & 0 \\ 0 & sf_y & y_c & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = C \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

โดยที่ C ซึ่งเป็นเมตริกซ์ขนาด 3x4 ซึ่งประกอบไปด้วยค่า s, f_x, f_y, x_c, y_c โดยทั่วไปค่าเหล่านี้รวมกันเรียกว่าพารามิเตอร์ (Camera Parameters) ซึ่งจะคำนวณได้มาจากขั้นตอนการเทียบมาตรฐานกล้อง (Camera Calibration) ส่วนค่าความสัมพันธ์ระหว่างจุดใด ๆ บนกรอบพิกัดอ้างอิงจอภาพในอุดมคติ (x_I y_I) กับ กรอบพิกัดอ้างอิงจอภาพที่สังเกตได้ (Observe Screen Coordinated Frame) (x_O y_O) ซึ่งเป็นจุดที่เราเห็นจริงๆในภาพ สามารถแสดงดังรูปที่ 2.16



รูปที่ 2.16 แสดงความสัมพันธ์ระหว่างกรอบพิกัดอ้างอิงจอภาพในอุดมคติ (Ideal Screen Coordinates) และกรอบพิกัดอ้างอิงจอภาพที่สังเกตได้ (Observe Screen Coordinates)

ที่มา: http://www.hitl.washington.edu/artoolkit/documentation/cs.htm

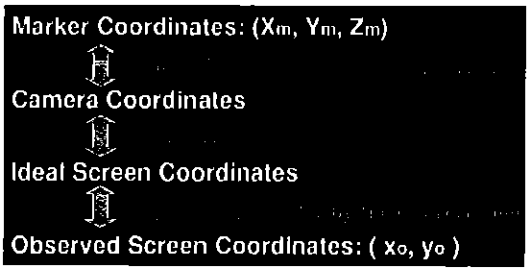
สามารถอธิบายได้ดังสมการ

$$d^2 = (x_1 - x_0)^2 + (y_1 - y_0)^2$$

$$p = \{1 - fd^2\}$$

$$x_0 = p(x_1 - x_0) + x_0, y_0 = p(y_1 - y_0) + y_0$$

โดยที่ x_0 y_0 คือจุดพิกัดศูนย์กลางของการความเพี้ยน (Center Coordinates of Distortion) และ f คือ ค่าปัจจัยความเพี้ยน (Distortion Factor) ซึ่งค่าทั้ง 2 จะได้มาจากกระบวนการการเทียบมาตรฐานกล้อง (Camera Calibration)



รูปที่ 2.17 กระบวนการคำนวณค่าประมาณค่าตำแหน่ง

ที่มา: facstaff.swu.ac.th/praditm/CP499_2552_AR.pdf

จากรูปที่ 2.17 จะแสดงกระบวนการที่จะได้มาของค่า T_{CM} เมื่อเรารู้ค่าตำแหน่งของสัญลักษณ์ทั้ง 4 จุดบนจุดกรอบพิกัดอ้างอิงจอภาพที่สังเกตได้ในภาพที่ถ่ายจากกล้องวีดีโอ ซึ่งกล่าวโดยเฉพาะค่านี้สามารถหาได้จากการคำนวณหาคำตอบของฟังก์ชั่นค่าผิดพลาด (Error Function) ดังสมการที่ 4 ซึ่งโดยทั่วไปแล้วเราจะใช้เทคนิคทางด้านการหาค่าที่เหมาะสม (Optimization) ซึ่งเป็นกระบวนแบบวนซ้ำ (Iterative) [11]

$$err = \frac{1}{4}\sum_{i=1,2,3,4}\{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\}$$

โดยที่ $\hat{x}_i$ $\hat{y}_i$ แสดงได้โดย

$$\begin{bmatrix} h\hat{x}_i \\ h\hat{y}_i \\ h \end{bmatrix} = \mathbf{C}.\mathbf{T_{CM}} \begin{bmatrix} X_{Mi} \\ Y_{Mi} \\ Z_{Mi} \\ 1 \end{bmatrix}, i = 1,2,3,4$$

2.8 แมทริกซ์การแปลง (Tranformation Matrix)

จากการประมาณตำแหน่งจะทำให้ได้ความสัมพันธ์ของแมทริกซ์ดังนี้ คือ แมทริกซ์ขนาด 3x4 โดยที่มีความสัมพันธ์เกี่ยวกับการปรับหมุน (rotation) เป็นขนาด 3x3 และ ความสัมพันธ์เกี่ยวกับทิศทาง (vector) เป็นขนาด 3x1 ดังรูปที่ 2.18

ความสัมพันธ์เกี่ยวกับทิศทาง (vector)

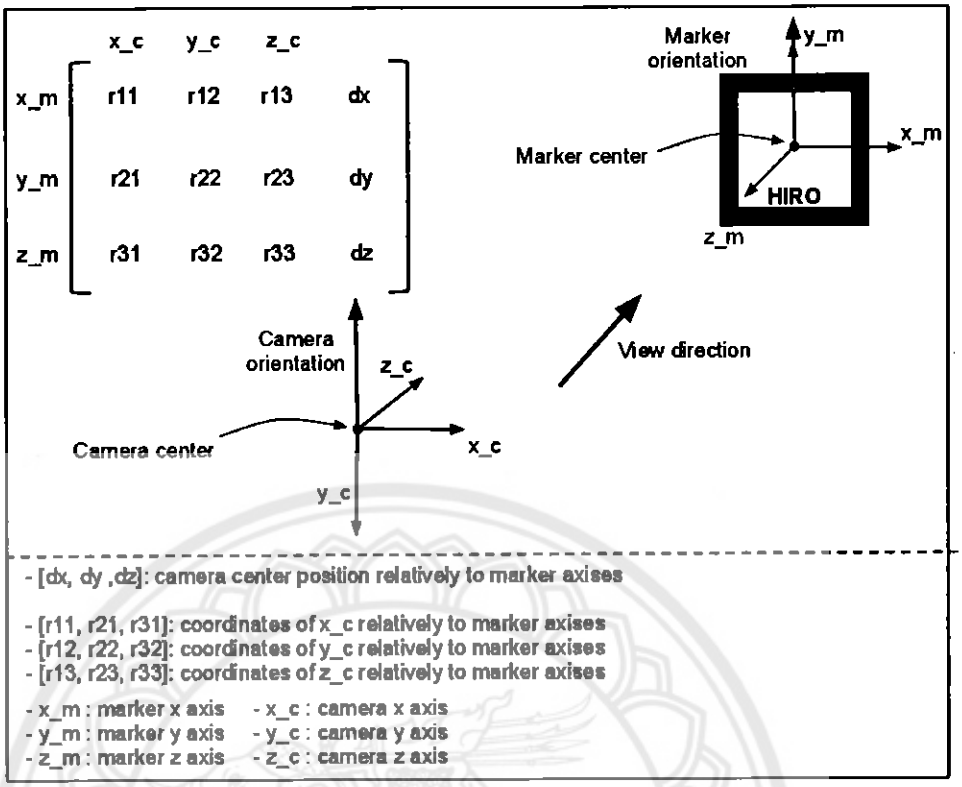
[dx ,dy, dz] คือ จุดศูนย์กลางของความสัมพันธ์ระหว่างกล้องกับสัญลักษณ์

ความสัมพันธ์เกี่ยวกับการปรับหมุน (rotation)

[r11, r12, r13] คือ กรอบพิกัดอ้างอิงแกน x ของกล้องกับสัญลักษณ์

[r21, r22, r23] คือ กรอบพิกัดอ้างอิงแกน y ของกล้องกับสัญลักษณ์

[r31, r32, r33] คือ กรอบพิกัดอ้างอิงแกน z ของกล้องกับสัญลักษณ์



รูปที่ 2.18 แสดงความสัมพันธ์แมทริกซ์การแปลง

ที่มา : http://www.omf.mytestbed.net/projects/robot/wiki/RobotControlSoftware?version=51

กรอบพิกัดอ้างอิงของสัญลักษณ์และกล้อง

x_m คือ แกน x ของสัญลักษณ์

y_m คือ แกน y ของสัญลักษณ์

z_m คือ แกน z ของสัญลักษณ์

x_c คือ แกน x ของกล้อง

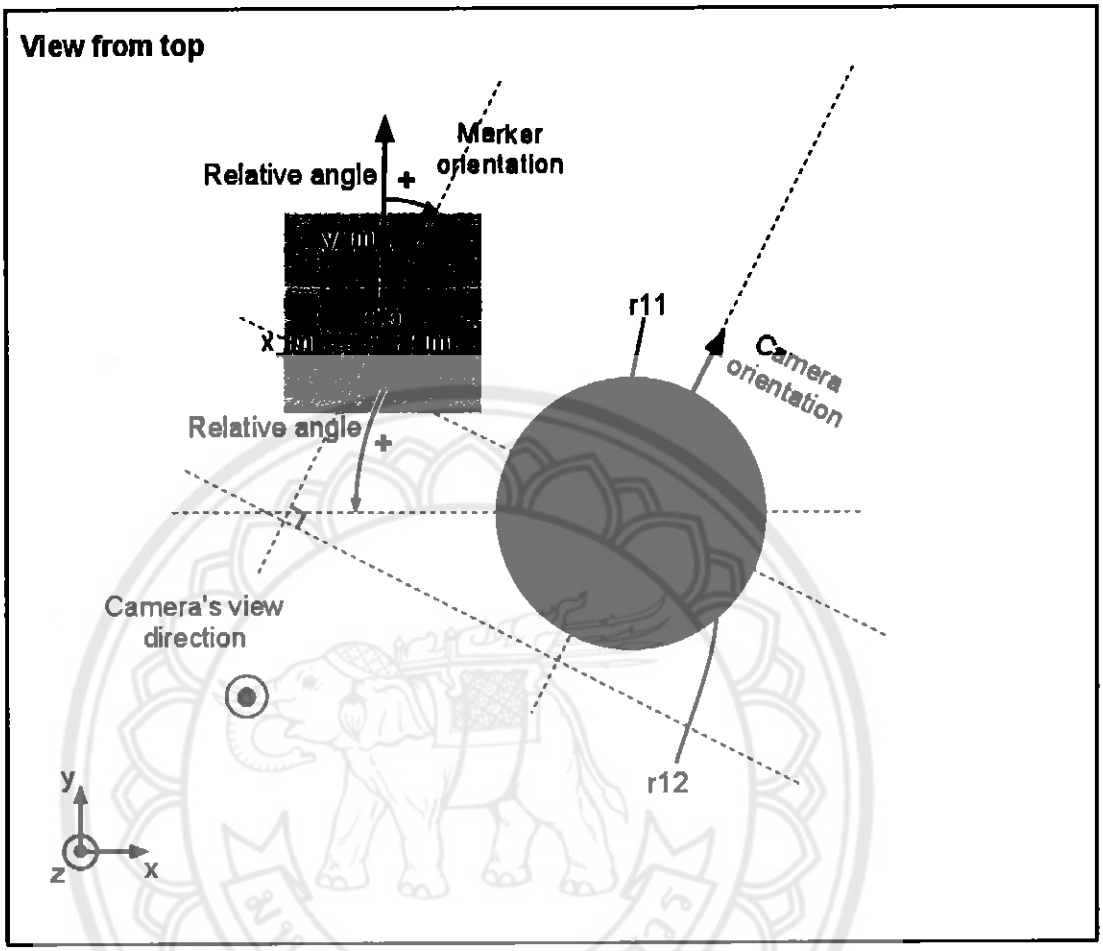
y_c คือ แกน y ของกล้อง

z_c คือ แกน z ของกล้อง

สามารถหาระยะทางระหว่างกล้องกับสัญลักษณ์ได้จากสมการนี้

ระยะทางระหว่างกล้องกับสัญลักษณ์ = $\sqrt{dx^2 + dy^2 + dz^2}$

ความสัมพันธ์มุม (Relative angle) เกิดจากการเปลี่ยนองศาระหว่างกล้องกับสัญลักษณ์ซึ่งเกิดจากความสัมพันธ์บางส่วน ดังรูปที่ 2.19



รูปที่ 2.19 แสดงความสัมพันธ์ของมุมระหว่างกล้องกับสัญลักษณ์

ที่มา: http://www.omf.mytestbed.net/projects/robot/wiki/RobotControlSoftware?version=51

ซึ่งสามารถคำนวณหาขนาดของมุมได้จากสมการนี้

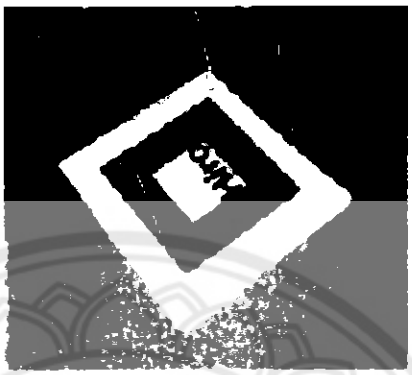
$$\text{relative angle} = 2\tan^{-1}\frac{r_{12}}{r_{11}}$$

มุมที่ส่งกลับจะอยู่ในรูปมุมเรเดียนอยู่ระหว่าง -pi และ pi โดยไม่นับมุม –pi

2.9 การเรนเดอร์ภาพสามมิติ

ส่วนนี้เป็นส่วนสุดท้ายที่จะทำให้กระบวนการสร้างความจริงเสริม ครบถ้วนสมบูรณ์ ซึ่งก็คือ การเพิ่ม (Augment) ข้อมูลที่เราต้องการซึ่งโดยทั่วไปแล้วจะเป็นโมเดล 3 มิติ (3D Model) ลงไปในภาพที่ได้จากกล้องวีดีโอ ณ ตำแหน่งของสัญลักษณ์ ที่ตรวจพบจากขั้นตอนการวิเคราะภาพ โดยใช้ค่าตำแหน่งเชิง 3 มิติที่คำนวณได้จากขั้นตอนประมาณตำแหน่ง กล่าวโดยทั่วไปแล้วการ

เรนเดอร์ภาพสามมิติ (3D Rendering) หมายถึง กระบวนการที่ทำการสร้างภาพ 2 มิติ จากโมเดล 3 มิติ ซึ่งโมเดล 3 มิตินี้ จะอธิบายวัตถุหรือสิ่งแวดล้อมหนึ่ง ๆ ที่ต้องการสร้างภาพนั้น[11] ดังตัวอย่างภาพที่ 2.20



รูปที่ 2.20 แสดงการคำนวณหาโมเดล 3 มิติจากตำแหน่งวัตถุ

2.10 ความรู้เกี่ยวกับไลบรารี OpenAL

รูปที่ 2.21 แสดงโลโก้ OpenAL

ที่มา: http://connect.creativelabs.com/openal/default.aspx

เนื่องจาก OpenGL เป็นการจัดการที่เกี่ยวกับภาพเพียงอย่างเดียว ดังนั้น OpenAL จึงได้เกิดขึ้นมาเพื่อลดช่องว่างดังกล่าว และตัวชุดคำสั่งมีลักษณะรูปแบบเดียวกับ OpenGL ชุดคำสั่งใน OpenAL เป็นมาตรฐานเดียวกันที่สามารถใช้ได้ทั้งบนเครื่องคอมพิวเตอร์ส่วนบุคคล และมุ่งเป้าหมายไปยังอุปกรณ์ (cross-platform) โดยไม่จำเป็นต้องแก้ไขชุดคำสั่งอีก

ถ้าต้องการใช้ OpenGL และระบบเสียง อาจเลือกใช้ DirectSound ซึ่งเป็นส่วนหนึ่งของ DirectX เป็นส่วนประกอบภายในโปรแกรมได้

เสียงและเพลงในเกมกำลังกลายเป็นสิ่งที่สำคัญต่อเกมมากขึ้นเรื่อย ๆ เนื่องมาจากความก้าวหน้าทางด้านเทคโนโลยี หลายปีที่ผ่านมาบริษัทผู้ผลิตการ์ดเสียงได้พัฒนาระบบ EAX Sound ขึ้นมาเพื่อขยายขีดความสามารถให้กับ DirectX และตอนนี้พวกเขาก็ได้สร้างเทคโนโลยี OpenAL(Open Audio Library)ขึ้นมาใหม่มันคือ API ของระบบเสียงที่มีแนวทางการทำงานไปในทางเดียวกันกับ OpenGL ซึ่งเป็น API ของกราฟฟิก OpenAL ถูกออกแบบเพื่อสนับสนุนคุณลักษณะพิเศษของการ์ดเสียง และมีการจัดเตรียมซอฟแวร์เอาไว้ให้เพื่อเป็นทางเลือกในกรณีที่ฮาร์ดแวร์ไม่มีคุณลักษณะพิเศษดังกล่าว อาจกล่าวได้ว่า OpenAL เป็นซอฟแวร์ที่ใช้ในการติดต่อกับฮาร์ดแวร์ออดิโอ สร้างขึ้นเพื่อให้โปรแกรมเมอร์ใช้ทักษะในการผลิตช่องสัญญาณเสียงที่ซับซ้อนได้ในระดับคุณภาพสูง OpenAL สามารถทำการจำลองสัญญาณเสียงในรูปแบบสามมิติได้และสามารถทำงานได้บนอุปกรณ์ทุกตัว ปัจจุบันเทคโนโลยีนี้กำลังถูกใช้กับเกมส์มากขึ้นเรื่อย ๆ[7]

2.11 ไฟล์เสียงรูปแบบต่าง ๆ

ไฟล์เสียงจัดเป็นข้อมลทางเสียงแบบดิจิตอลที่ใช้กับคอมพิวเตอร์ ไฟล์เสียงมีอยู่หลายประเภทด้วยกันและแต่ละประเภทก็มีคุณสมบัติและวัตถุประสงค์ในการใช้งานแตกต่างกันไป ดังนี้

wave

ไฟล์เสียง wave เป็นไฟล์เสียงที่เราคุ้นเคยกันมากที่สุด ไฟล์ประเภทนี้มีนามสกุล .wav จัดเป็นไฟล์เสียงมารฐานที่ใช้กับ Windows คุณสมบัติที่สำคัญคือครอบคลุมความถี่เสียงได้ทั้งหมด ทำให้คุณภาพเสียงดีมาก และยังให้เสียงในรูปแบบสเตอริโอได้อีกด้วย ข้อเสียคือไฟล์ .wav มีขนาดใหญ่ทำให้สิ้นเปลืองพื้นที่ในการเก็บข้อมูลมาก

CD Audio

ไฟล์ CDA เป็นไฟล์เพลงบนแผ่น CD ที่ใช้กับเครื่องเล่น CD ทั่วไปไฟล์ประเภทนี้เมื่อนำมาใช้กับเครื่องคอมพิวเตอร์ โปรแกรมสำหรับเล่น CD จะมองเห็นข้อมูลเสียงในรูปของแทร็กเสียง (Audio Tack) แต่ถ้าดูด้วย Windows Explorer จะเห็นเป็นไฟล์มีนามสกุล .cda ไฟล์ CDA มีคุณสมบัติทางเสียงเหมือนกับไฟล์ wave คือให้คุณภาพเสียงที่ดีเป็นธรรมชาติ จึงนิยมใช้บันทึกลงบนแผ่น CDเป็นสื่อดนตรี เรียกทั่วไปว่า CD เพลง ถ้าต้องการ copy หรือนำไฟล์ประเภทนี้มาใช้งานกับโปรแกรมประยุกต์ต่างๆ จะต้องแปลงให้เป็นไฟล์ wave หรือไฟล์ที่โปรแกรมประยุกต์นั้นๆ รู้จักเสียก่อนจึงจะใช้ได้ หรืออาจจะใช้โปรแกรมที่สามารถ Extract ไฟล์ Audio CD ออกมาเป็นไฟล์ wave ซึ่งก็มีใช้หลายโปรแกรม เช่น Sound Forge, Ware Lab ฯลฯ

MP3

ไฟล์เสียง MP3 เป็นไฟล์เสียงยอดนิยมในปัจจุบัน มีนามสกุล .mp3 ไฟล์ mp3 เป็นไฟล์ที่ถูกบีบอัดข้อมูลทำให้ไฟลีประเภทนี้มีขนาดเล็กลงมาก ลดลงประมาณ 10 เท่าเมื่อเทียบกับไฟล์ wave คุณภาพเสียง mp3 ค่อนข้างดีจึงนิยมใช้ไฟล์ประเภทนี้บันทึกข้อมูลเพลงลงบนสื่อคอมพิวเตอร์หรือแผ่น CD การเล่นไฟล์ mp3 บนเครื่องคอมพิวเตอร์ต้องใช้โปรแกรมโดยเฉพาะซึ่งในปัจจุบันมีให้เลือกมากมาย เช่น Winamp, MP3 Player, Windows Media Player, Musicmaatch Jukebox ฯลฯ นอกจากนี้ยังสามารถเล่นได้กับเครื่องเล่น VCD ,DVD, CD ติดรถยนต์, เครื่องเล่น MP3 แบบพกพา (ใช้หน่วยความจำเฉพาะ เช่น Flash Memory หรือ memory Stick) รวมทั้งโทรศัพท์มือถือรุ่นใหม่ๆ อีกด้วยปัจจุบันมีการพัฒนาไฟล์ MP3 เป็นไฟล์ MP4 ให้มีขนาดเล็กลงและมีคุณภาพเสียงดีขึ้น แต่ยังไม่เป็นที่นิยมใช้มากนัก

Ogg

เป็นรูปแบบของไฟล์เสียงใหม่ล่าสุด มีชื่อเต็มคือ Ogg มีนามสกุล .ogg ไฟล์ Ogg Vorbis ใช้เทคโนโลยีการบีบอัดไฟล์แบบใหม่ ทำให้ไฟล์ที่ได้มีขนาดเล็กกว่า MP3 เสียอีก แต่ให้คุณภาพเสียงที่ดีกว่าและที่สำคัญคืออยู่ในกลุ่มของ Open Source Project ทำให้กลายเป็นฟรีแวร์ อีกทั้งยังมีความสามารถด้าน Streaming ด้วย ทำให้ได้รับความนิยมมากในหมู่ผู้เล่นอินเตอร์เน็ต ไฟล์ Ogg Vorbis สามารถเล่นได้โดยใช้โปรแกรมสำหรับเล่นไฟล์ MP3 โดยมีข้อแม้ว่าโปรแกรมนั้นจะต้องมี Plug-in สำหรับ Ogg ด้วย Ogg Vorbis นับเป็นคลื่นลูกใหม่ของวงการไฟล์เสียง เพราะมีขนาดที่เล็กมาก สามารถเข้ารหัสเสียงได้หลายแบบทั้ง mono,stereo จนถึงระบบ 5.1 Surround Sound

MIDI

ไฟล์เสียง MIDI ไฟล์ข้อมูลเสียงดนตรี โดยมีนามสกุล .midi ไฟล์ MIDI จะบรรจุข้อมูลของอุปกรณ์ต่างๆ ที่ให้เสียงดนตรี เมื่อเล่นไฟล์ MIDI ก็จะเป็นการสั่งให้อุปกรณ์นั้นๆให้มีเสียงดนตรีออกมา เมื่อนำมาเรียงกันก็จะกลายเป็นท่วงทำนองดนตรีซึ่งก็คือเสียงเพลงนั่นเอง MIDI มีขนาดของไฟล์ที่เล็กมากทำให้นิยมใช้ในการเก็บข้อมูลที่เป็นเสียงดนตรี ดังจะเห็นได้จากวงดนตรีประเภทเล่นคนเดียวจะใช้ข้อมูลเพลงจากแผ่นดิสก์ขนาด 3.5 นิ้วซึ่งสามารถเก็บข้อมูลดนตรีได้เป็น 10 เพลง ใส่เข้าไปในเครื่องสร้างเสียงดนตรี (Sequencer) เพื่อให้สร้างเสียงเพลงตามข้อมูลดนตรีที่อ่านจากแผ่น สำหรับเครื่องคอมพิวเตอร์ก็สามารถเล่นไฟล์ MIDI ได้โดยใช้โปรแกรมประเภท MIDI Player ซึ่งมีให้เลือกใช้มากมาย เสียงเพลงที่ได้จากโปรแกรมคาราโอเกะก็เป็นเสียงที่ได้จาก

ไฟล์ MIDI เช่นเดียวกัน ดังนั้นเราจึงสามารถนำเสียงเพลงจากโปรแกรมคาราโอเกะมาใช้ได้ แต่ถ้าจะให้สะดวกก็ควรแปลงให้เป็น wave เสียก่อนจะทำให้สามารถนำไปตัดต่อและใช้งานได้ง่ายขึ้น

WMA

ไฟล์ WMA เป็นรูปแบบไฟล์แบบหนึ่งของบริษัทไมโครซอฟต์ ชื่อเต็มคือ Windows Media Audio เป็นไฟล์ที่มีนามสกุลเป็น .wma ไฟล์ WMA จัดได้ว่าเป็นคู่แข่งของ mp3 และ Real Audio เพราะมีคุณสมบัติด้านการ Streaming เช่นเดียวกัน แต่ให้คุณภาพเสียงที่ดีกว่าในขณะที่ขนาดของไฟล์เล็กกว่าประมาณครึ่งหนึ่ง ทำให้ใช้เวลาน้อยกว่าในการดาวน์โหลดผ่านอินเตอร์เน็ต เมื่อก่อนการเล่นไฟล์ประเภทนี้ต้องเล่นผ่านโปรแกรม Windows Media Player เท่านั้น แต่ปัจจุบันมีโปรแกรมหลายโปรแกรมที่สามารถเล่นไฟล์นี้

Real Audio

Real Audio ไฟล์ Real Audio เป็นไฟล์ที่มีนามสกุล .ra เป็นไฟล์ประเภท Streaming ที่ใช้สำหรับการชมภาพและเสียงบนอินเตอร์เน็ต ไฟล์ประเภท Streaming จะใช้เทคโนโลยีในการบีบอัดข้อมูลที่ปรับเปลี่ยนได้เพื่อให้ได้ไฟล์ขนาดเล็กที่มีขนาดเหมาะสมสำหรับการส่งผ่านระบบอินเตอร์เน็ตในระดับความเร็วที่แตกต่างกัน ไฟล์ประเภทนี้ทำให้เราสามารถชมภาพและเสียงผ่านระบบอินเตอร์เน็ตได้อย่างต่อเนื่องผ่านโปรแกรมประเภท Real ต่างๆ ได้โดยไม่ต้องรอให้การดาวน์โหลดข้อมูลเสร็จสิ้น โปรมแกรม Real Player ก็เป็นโปรแกรมประเภทหนึ่งที่ใช้เล่นไฟล์ Real Audio ได้

Audio Streaming Format

Audio Streaming Format (.asf) เป็นไฟล์เสียงหนึ่งที่มีรูปแบบ Streaming ที่เน้นส่งข้อมูลเสียงแบบ real time ใช้กันมากในการฟังวิทยุออนไลน์บนอินเตอร์เน็ต

Audio Interchange File Format

Audio Interchange File Format (.aif , .aiff) เป็นไฟล์ลักษณะคล้ายไฟล์ Wave แต่ใช้สำหรับเครื่อง

MacintoshACC

MacintoshACC (.acc) เป็นไฟล์เสียงที่มีคุณภาพสูงมาก สุ่มความถี่ได้ถึง 96 kHz รองรับอัตราการเล่นไฟล์สูงถึง 576 Kbps สามารถแยกเสียงได้ถึงระบบ 5.1ช่อง เทียบเท่า Dolby Digital หรือ AC-3[8]

บทที่ 3

วิธีการดำเนินโครงการ

จุดประสงค์ของโครงงานก็เพื่อที่จะสร้างโปรแกรมที่สามารถรับข้อมูลทางกล้องแล้วนำไปประมวลผลให้เกิดเป็นเสียงดนตรีและแสดงภาพจำลอง 3 มิติ ดังนั้นจึงต้องมีการทำงานหลาย ๆ ส่วนร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์ โดยมีขั้นตอนคร่าว ๆ ดังนี้

- ศึกษารายละเอียดและหลักการ ความรู้ ทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับโครงงาน
- ออกแบบการทำงานทั้งหมด
- ออกแบบการทำงานของโปรแกรม
- ออกแบบสัญลักษณ์ (Marker) และสร้าง "ก้อนดนตรีเสมือน"
- ออกแบบการอ่านและค้นหาสัญลักษณ์ (Marker)
- ออกแบบโปรแกรมแสดงผล
- ออกแบบโปรแกรมเพื่อปรับระดับเสียงและจังหวะความเร็วของเสียงดนตรี

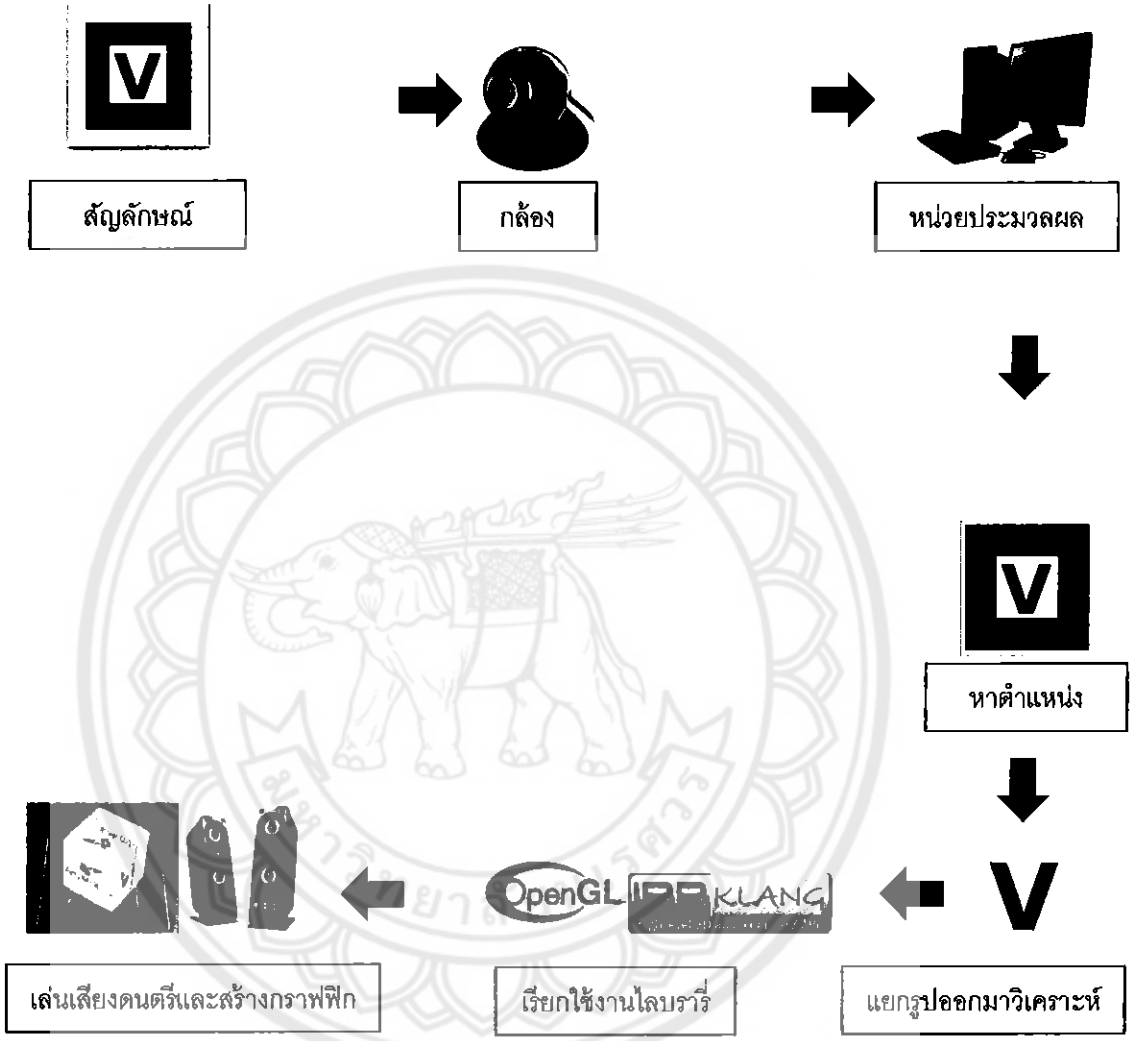
3.1 ศึกษารายละเอียดและหลักการ ความรู้ ทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับโครงงาน

โครงงาน "ก้อนดนตรีเสมือน" แบ่งการทำงานหลักได้เป็น 3 ส่วน ได้แก่ ส่วนแรก คือ การตรวจจับและประมาณตำแหน่งและทิศทางตัวของสัญลักษณ์ (Marker) ซึ่งติดอยู่บนด้านทั้งหกของก้อนดนตรีเสมือนผ่านทางกล้องเว็บแคม ซึ่งเครื่องมือที่ใช้ในการทำงานส่วนนี้คือไลบรารี่ ARToolKit ส่วนที่สอง คือ การเล่นดนตรีโดยอาศัยข้อมูลตำแหน่งและทิศทางของสัญลักษณ์ในการเพิ่มหรือลดจังหวะและความดังของเสียงดนตรี ซึ่งเครื่องมือที่ใช้ในการทำงานส่วนนี้คือไลบรารี่ irrKlang และส่วนที่สาม คือ การแสดงภาพกราฟฟิกให้สอดคล้องกับตำแหน่งและทิศทางของสัญลักษณ์ที่ตรวจจับได้ เครื่องมือที่ใช้ในการทำงานส่วนนี้ก็คือไลบรารี่ OpenGL ซึ่งเครื่องมือทั้ง 3 อย่างนี้จะใช้หลักการเขียนโปรแกรมภาษา C++ เครื่องต่าง ๆ ที่ใช้แสดงดังรูปที่ 3.1

รูปที่ 3.1 แสดงเครื่องมือที่ใช้

3.2 ออกแบบการทำงานทั้งหมด

หลักการทำงานของ "ก้อนดนตรีเสมือน" มีขั้นตอนต่าง ๆ ดังรูปที่ 3.2



รูปที่ 3.2 แสดงขั้นตอนการทำงานของ "ก้อนดนตรีเสมือน"

3.2.1 ขั้นตอนการทำงานของก้อนดนตรีเสมือน มีดังนี้

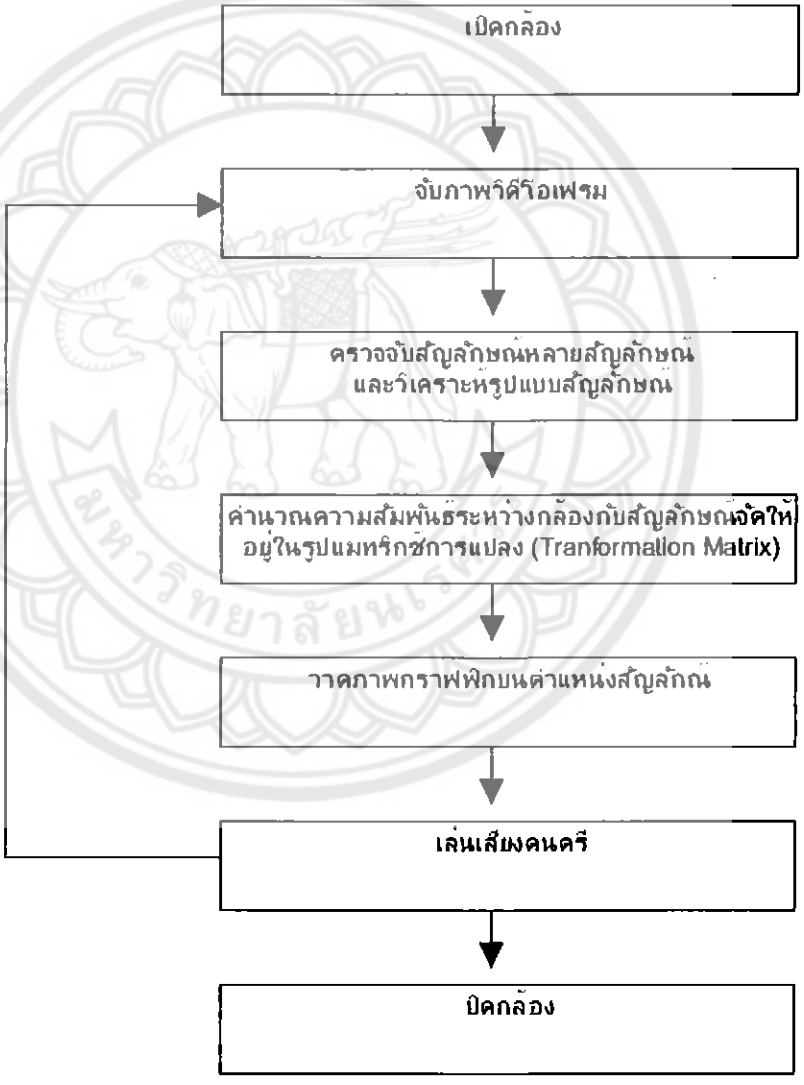
1. อ่านสัญลักษณ์ผ่านเข้าทางกล้อง
2. กล้องจะส่งภาพสัญลักษณ์ที่เป็นกรอบสี่เหลี่ยมไปยังคอมพิวเตอร์เพื่อประมวลผล
3. ซอฟต์แวร์ทำการค้นหากรอบสี่เหลี่ยมซึ่งเรียกว่า "สัญลักษณ์" เพื่อหารูปร่างที่โปรแกรม รู้จัก
4. จากนั้นซอฟต์แวร์จะทำการคำนวณเพื่อแยกแยะรูปร่างของ สัญลักษณ์ เพื่อเปรียบเทียบว่าตรงกันกับที่กำหนดไว้หรือไม่

5. หากถูกต้อง ก็จะสร้างรูปภาพวัตถุเสมือน (virtual object) ที่จะเป็นรูปแบบสองหรือสามมิติก็ได้ตามที่กำหนด

6. สุดท้ายเป็นขั้นตอนของการแสดงผลออกมาทางหน้าจอคอมพิวเตอร์เป็นภาพกราฟฟิกพร้อมทั้งเล่นเสียงดนตรี

3.3 ออกแบบการทำงานของโปรแกรม

กระบวนการทำงานทั้งหมดของโปรแกรมมีดังรูปที่ 3.3



รูปที่ 3.3 แสดงลำดับการทำงานของโปรแกรม

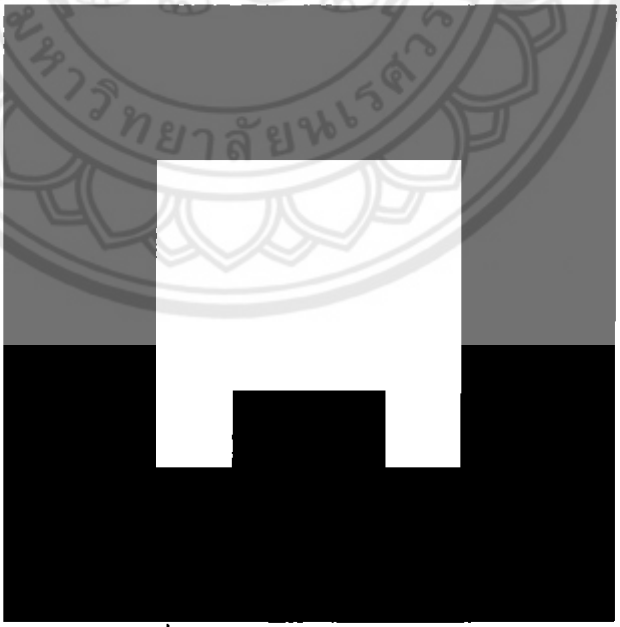
3.3.1 รายละเอียดส่วนของโปรแกรมที่สำคัญ

- main	ฟังก์ชั่นการทำงานหลักของโปรแกรม
- init	ฟังก์ชั่นการเรียกใช้งานกล้อง
- mainLoop	ฟังก์ชั่นการทำงานหลักสำหรับ AR function
- draw	ฟังก์ชั่นวาดภาพกราฟฟิก
- playSound	ฟังก์ชั่นเล่นเสียงดนตรี
- cleanup	ฟังก์ชั่นหยุดการทำงาน

3.4 ออกแบบสัญลักษณ์และสร้างก้อนดนตรีเสมือน

3.4.1 การออกแบบสัญลักษณ์

ในขั้นตอนนี้เป็นการค้นหาสัญลักษณ์จากภาพที่ได้จากกล้องวีดีโอ โดยก่อนอื่นต้องทำการดึงข้อมูลที่จำเป็นของสัญลักษณ์ที่จะใช้ มาสร้างเป็นฐานข้อมูลเก็บไว้ก่อน ซึ่งข้อมูลที่จำเป็น เช่น ขนาดของสัญลักษณ์ (เซนติเมตร) และรูปแบบของสัญลักษณ์เป็นต้น โดยทั่วไปแล้ว ARToolKit รูปแบบของสัญลักษณ์จะต้องเป็นรูปสี่เหลี่ยมจัตุรัสที่มีขอบสีดำ พื้นหลังด้านในสีขาว และรูปแบบของสัญลักษณ์เป็นสีดำดังรูปที่ 3.4 จะถูกจัดเก็บค่าพิกเซล (pixel) ให้อยู่ในรูปแบบของเมทริกซ์ (Matrix) และถูกจัดเก็บให้อยู่ให้ไฟล์นามสกุล ".patt" ตัวอย่างสัญลักษณ์แสดงดังรูปที่ 3.5



รูปที่ 3.4 แสดงตัวอย่างสัญลักษณ์

255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255

255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255

255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255

255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255

255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255

255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255

255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255

255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255

255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255

255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255

255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255

255 255 255 255 0 0 0 0 0 0 0 0 255 255 255 255

255 255 255 255 0 0 0 0 0 0 0 0 255 255 255 255

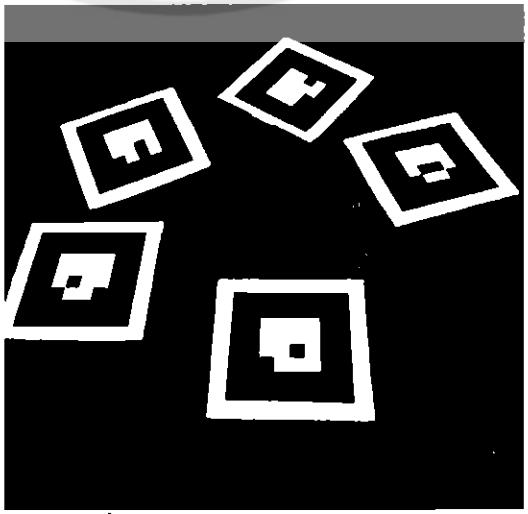
255 255 255 255 0 0 0 0 0 0 0 0 255 255 255 255

255 255 255 255 0 0 0 0 0 0 0 0 255 255 255 255

รูปที่ 3.5 แสดงสัญลักษณ์ถูกเก็บเป็นพิกเซล

3.4.2 ออกแบบและสร้าง “ก้อนดนตรีเสมือน”

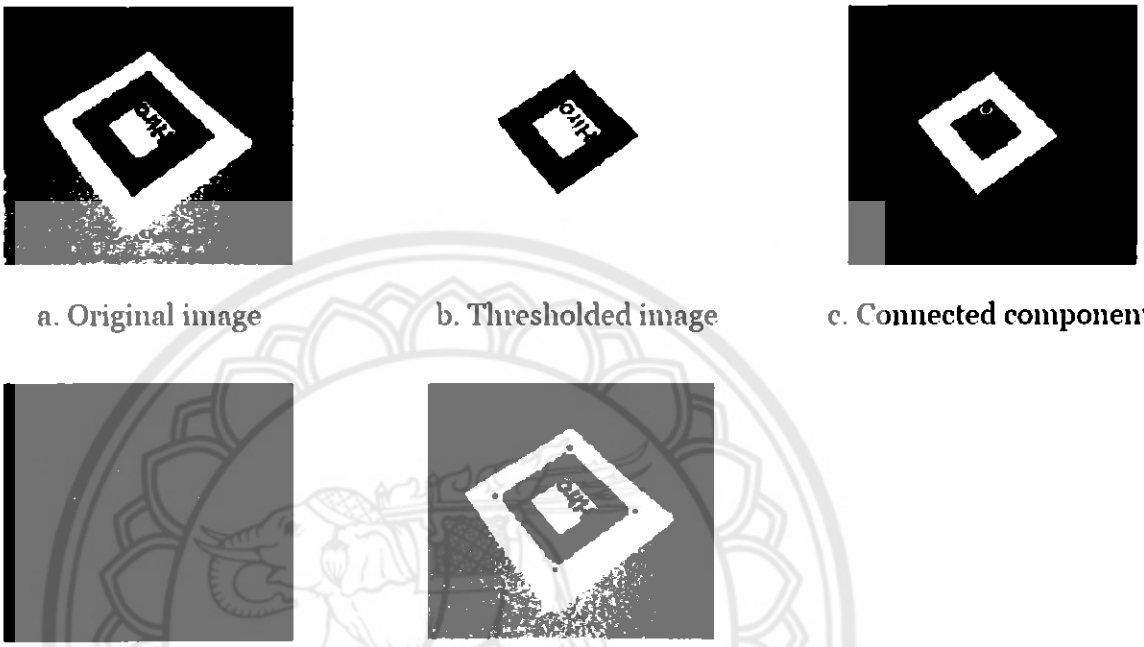
โดยการนำสัญลักษณ์มาประกอบเป็นทรงสี่เหลี่ยมลูกบาศก์ที่มีขนาด 9x9x9 เซนติเมตร แต่ละด้านใช้สัญลักษณ์ที่แตกต่างกันออกไป ดังรูปที่ 3.6



รูปที่ 3.6 แสดง “ก้อนดนตรีเสมือน”

3.5 ออกแบบการอ่านและค้นหาสัญลักษณ์

การค้นหากรอบสี่เหลี่ยมจะใช้หลักการประมาณตำแหน่ง (pose estimation algorithm) โดยมีหลักการดังรูปที่ 3.7



a. Original image b. Thresholded image c. Connected components

d. Contours e. Edges and corners

รูปที่ 3.7 แสดงหลักการอ่านและค้นหาสัญลักษณ์ (Marker)

a. รับภาพจากกล้อง

b. ทำการแปลงภาพให้กลายเป็นระดับไบนารี โดยการกำหนดให้แต่ละพิกเซลในภาพมีค่าเป็น 2 ระดับ คือ 0 หรือ 1 โดยที่จะมีค่าเป็น 0 เมื่อค่าระดับความสว่าง (Intensity) ของพิกเซล นั้นมีต่ำกว่าค่าขีดแบ่ง (Threshold) มิฉะนั้นแล้วจะมีค่าเป็น 1

c. ทำการหาพื้นที่ติดต่อกันโดยใช้เทคนิคทางการวิเคราะห์ภาพที่เรียกว่าติดป้ายองค์ประกอบที่ติดกัน (Connected component labeling)

d. เมื่อทำขั้นตอน c. เสร็จแล้วก็จะได้ผลลัพธ์เป็นเส้นรอบรูป (Contours)

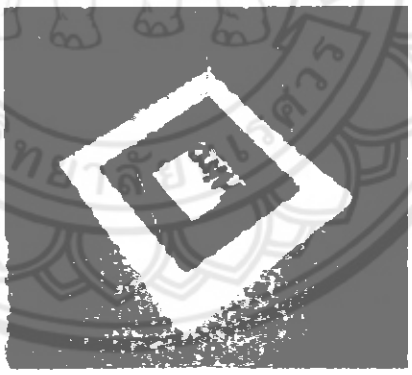
e. เมื่อหาเส้นรอบรูปได้แล้ว ระบบจะทำการประมาณหาค่าพารามิเตอร์ของสมการเส้นตรงที่แทนเส้นรอบรูปซึ่งเป็นรูปสี่เหลี่ยมทั้ง 4 เส้น หลังจากนั้นระบบจะหาจุดมุม (Corners) ทั้ง 4 จุดของสัญลักษณ์จากจุดตัดของเส้นตรงทั้งสี่ที่หาได้ ซึ่งจุดมุม 4 จุดนี้จะถูกนำไปใช้ในกระบวนการประมาณตำแหน่ง

3.5.1 การประมาณตำแหน่ง

เป็นขั้นตอนของการคำนวณค่าตำแหน่งเชิง 3 มิติ (3D Pose) ของสัญลักษณ์เมื่อเทียบกับกล้องวีดีโอ ค่านี้จะถูกแสดงในรูปเมตริกซ์ขนาด 4x4 ที่ระบุความสัมพันธ์ระหว่างกรอบพิกัดอ้างอิงของกล้องและกรอบพิกัดอ้างอิงของสัญลักษณ์ ซึ่งกรอบพิกัดอ้างอิงของกล้องก็คือกรอบพิกัดอ้างอิงที่ใช้อ้างอิงตำแหน่งใดๆของกล้องวีดีโอ และกรอบพิกัดอ้างอิงสัญลักษณ์ก็คือกรอบพิกัดอ้างอิงที่ใช้อ้างอิงตำแหน่งใดๆของสัญลักษณ์

3.5.2 3D rendering

ส่วนนี้เป็นส่วนสุดท้ายที่จะทำให้กระบวนการ AR ครบถ้วนสมบูรณ์ ซึ่งก็คือ การเพิ่ม (Augment) ข้อมูลที่เราต้องการซึ่งโดยทั่วไปแล้วจะเป็นโมเดล 3 มิติ (3D Model) ลงไปในภาพที่ได้จากกล้องวีดีโอ ณ ตำแหน่งของ Marker ที่ตรวจพบจากขั้นตอน Image Analysis โดยใช้ค่าตำแหน่งเชิง 3 มิติที่คำนวณได้จากขั้นตอน Pose Estimation กล่าวโดยทั่วไปแล้ว 3D Rendering หมายถึงกระบวนการที่ทำการสร้างภาพ 2 มิติ จากโมเดล 3 มิติ ซึ่งโมเดล 3 มิตินี้ จะอธิบายวัตถุหรือสิ่งแวดล้อมหนึ่ง ๆ ที่ต้องการสร้างภาพนั้น ตัวอย่างดังรูปที่ 3.8



รูปที่ 3.8 แสดงการคำนวณหาโมเดล 3 มิติจากตำแหน่งวัตถุ

3.6 การออกแบบการแสดงผล

รูปที่ 3.9 แสดงภาพก้อนดนตรีเสมือน

3.6.1 การแสดงผลด้านกราฟฟิกแบ่งออกเป็น 2 ส่วนที่สำคัญ ดังนี้

1. ส่วนของการวาดภาพกราฟฟิก 3 มิติ จะเป็นการวาดวัตถุรูปทรงกลมที่ลอยอยู่บนตำแหน่งของสัญลักษณ์ที่ใส่ลวดลายและสีสันต่าง ๆ สวยงาม ดังหมายเลข 1 ในรูปที่ 3.9

2. ส่วนของการแสดงสถานะของเสียงดนตรี จะเป็นการวาดรูปวงกลมลักษณะเป็นวงแหวนรอบวัตถุทรงกลมเพื่อแสดงสถานะของเสียงดนตรีด้วยสีต่างที่แตกต่างกัน ดังหมายเลข 2 ในรูปที่ 3.9

อธิบายการแสดงผลส่วนที่ 2 ส่วนของการแสดงสถานะของเสียงดนตรี

รูปที่ 3.10 แสดงสถานะเสียงดนตรี

วงกลมสีเขียวนอกสุด แสดงสถานะของเอฟเฟกต์เสียง เช่น เสียงปกติจะแสดงเป็นสีเขียว และเสียงเอฟเฟกต์เสียงเพี้ยน (distortion) จะแสดงเป็นสีแดง ดังรูปที่ 3.10

วงกลมสีน้ำเงินตรงกลาง แสดงสถานะความช้า-เร็วของจังหวะเสียงดนตรี แถบสีน้ำเงินมีทั้งหมด 20 แถบ สามารถปรับความเร็วได้ 10 ระดับ ดังนั้นสีน้ำเงิน 2 แถบจะแทนค่าความเร็วเท่ากับ 1 ระดับ เมื่อความเร็วเพิ่มขึ้น 1 ระดับ แถบสีน้ำเงินก็จะเพิ่มขึ้นทีละ 2 แถบ และในทางกลับกันเมื่อความเร็วลดลง 1 ระดับ แถบสีน้ำเงินก็จะลดลงทีละ 2 แถบ ดังรูปที่ 3.10

วงกลมสีแดง แสดงสถานะความเบา-ดังของเสียงดนตรี แถบสีแดงมีทั้งหมด 20 แถบ สามารถปรับความดังได้ 10 ระดับ ดังนั้นสีแดง 2 แถบจะแทนค่าความดังเท่ากับ 1 ระดับ เมื่อความดังเพิ่มขึ้น 1 ระดับ แถบสีแดงก็จะเพิ่มขึ้นทีละ 2 แถบ และในทางกลับกันเมื่อความดังลดลง 1 ระดับ แถบสีแดงก็จะลดลงทีละ 2 แถบ ดังรูปที่ 3.10

3.7 ออกแบบการส่งภาพพื้นผิว

รูปที่ 3.11 แสดงส่วนประกอบการทำ texture

การส่งภาพพื้นผิว (Texture Mapping) เป็นเทคนิคในงานกราฟิกส์ที่นิยมใช้ในการทำให้วัตถุดูสมจริง อีกทั้งยังแสดงผลได้อย่างรวดเร็วอีกด้วยโดยทั่วไปแล้ว รูปถ่ายหรือภาพที่สร้างขึ้นจะถูกใช้เพื่อแม็พลงบนวัตถุหรือพื้นผิวที่อยู่ในปริภูมิ 3 มิติ จุด (vertex) บนวัตถุแต่ละจุดจะมีการกำหนดตำแหน่งพิกัดบนรูปภาพหรือที่เรียกว่าพิกัดภาพพื้นผิว (texture coordinate) เพื่อให้รู้ว่าบริเวณไหนบนรูปจะถูกนำไปแสดง ณ บริเวณใดบนวัตถุ ดังตัวอย่างรูปที่ 3.11

3.7.1 ขั้นตอนการส่งภาพพื้นผิว

1. สร้างวัตถุภาพพื้นผิว (Texture object) ขึ้นมา (เช่น โหลดเข้ามาจากไฟล์รูปภาพ) แล้วโหลดภาพพื้นผิวเข้าไปไว้ในหน่วยความจำที่วัตถุภาพพื้นผิวอ้างอิงไปถึง
2. กำหนดลักษณะการส่งภาพพื้นผิวลงไปบนแต่ละพิกเซล
3. กำหนดให้สถานะของการส่งภาพพื้นผิวถูกเปิดขึ้นเพื่อใช้งาน
4. วาดวัตถุ โดยการกำหนดพิกัดภาพพื้นผิว (Texture coordinate) กับตำแหน่งของจุด (vertex)

3.7.2 การสร้างวัตถุภาพพื้นผิว

1. ส่วนใหญ่ภาพพื้นผิว (Texture) ที่ใช้โดยทั่วไปจะอยู่ในรูปของข้อมูล 2 มิติอย่างเช่นข้อมูลรูปภาพ แต่ภาพพื้นผิวสามารถอยู่ในรูปของภาพพื้นผิว 1 มิติและ 3 มิติได้
2. ข้อมูลในภาพพื้นผิวอาจอยู่ในรูปของข้อมูล 1, 2, 3 หรือ 4 องค์ประกอบ (component) ซึ่งสามารถเป็นได้ตั้งแต่ค่าสเกลาร์จนถึงเวกเตอร์ของสี <R,G,B,A>
3. ข้อมูลภาพพื้นผิวจะถูกเก็บในหน่วยความจำที่วัตถุภาพพื้นผิวอ้างอิง
4. การเรียกใช้ฟังก์ชัน glGenTextures(1,&tex_id) โดยที่ tex_id เป็นตัวแปรชนิด int จะสร้างวัตถุภาพพื้นผิวจำนวน 1 อัน โดยที่ tex_id จะใช้อ้างอิงไปยังวัตถุภาพพื้นผิวที่ถูกสร้างขึ้นมา
5. วัตถุภาพพื้นผิวที่ถูกสร้างขึ้นมาสามารถเก็บข้อมูลของภาพพื้นผิวตั้งแต่ 1 มิติจนถึง 3 มิติ

6. ฟังก์ชัน glBindTexture(GL_TEXTURE_2D,tex_id) ใช้ระบุประเภทภาพพื้นผิวที่จะถูกเก็บในวัตถุภาพพื้นผิวซึ่งในตัวอย่างนี้จะเก็บภาพพื้นผิว 2 มิติ เช่น รูปภาพ เป็นต้น

7. การใช้งานภาพพื้นผิวมีลักษณะเป็นเครื่องสถานะ (state machine) เช่นกัน กล่าวคือ คำสั่งที่มีผลต่อภาพพื้นผิวจะส่งผลไปยังภาพพื้นผิวที่เป็นวัตถุ tex_id

8. ฟังก์ชัน glTexImage2D(GL_TEXTURE_2D,0,GL_RGB,w,h,0, GL_RGB,GL_UNSIGNED_BYTE,image) ใช้โหลดข้อมูลภาพพื้นผิวที่อยู่ในตัวแปร image ไปเก็บในวัตถุภาพพื้นผิว tex_id

3.8 ออกแบบโปรแกรมเพื่อปรับระดับเสียงและจังหวะความเร็วของเสียงดนตรี

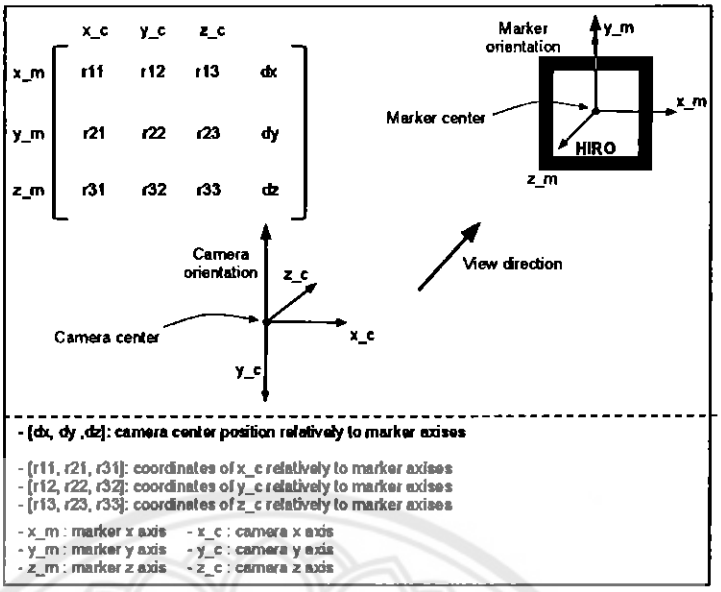
3.8.1 ออกแบบการปรับระดับความเบา-ดังของเสียงดนตรี

การปรับความดังเสียงดนตรีใช้หลักการวัดระยะทางระหว่างก้อนดนตรีเสมือน 2 ก้อน คือ ก้อนที่เล่นเสียงดนตรี (music marker) กับ ก้อนปรับเสียง (volume marker) ถ้าก้อนดนตรีอยู่ใกล้ก้อนปรับเสียงเสียงจะเบาลง และถ้าก้อนดนตรีเสมือนอยู่ไกลก้อนปรับเสียงเสียงจะดังขึ้น ดังรูปที่ 3.12

รูปที่ 3.12 แสดงการออกแบบการปรับระดับความเบา-ดังของเสียง

3.8.1.1 หลักการปรับเสียง

เมื่อค้นหาสัญลักษณ์ได้แล้วจะถูกจัดให้อยู่ในรูปแมทริกซ์การแปลง (Tranformation Matrix) ดังรูปที่ 3.13



รูปที่ 3.13 แสดงการแปลงค่าสัญลักษณ์ให้อยู่ในรูปแมทริกซ์

ที่มา: http://omf.mytestbed.net/projects/robot/wiki/RobotControlSoftware?version=51

แมทริกซ์ที่ได้จะมีขนาด [3][4] ซึ่งจะสามารถหาค่าจุดศูนย์กลางของสัญลักษณ์ได้จากสมการนี้

$$\text{ระยะทาง} = \sqrt{dx^2 + dy^2 + dz^2}$$

ดังนั้นจะหาระยะห่างระหว่างสัญลักษณ์ 2 อัน ได้ ดังนี้

$$\text{ระยะทาง} = \sqrt{(Vx - Mx)^2 + (Vy - My)^2 + (Vz - Mz)^2}$$

ตัวอย่างเช่น

แมทริกซ์ M เป็นของ ก้อนดนตรีเสมือน

$$\begin{bmatrix} 0 & -0.5 & 8 & 7 \\ 0.5 & 0 & 6 & 4 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

แมทริกซ์ V เป็นของ ก้อนปรับเสียง

$$\begin{bmatrix} 0 & -0.5 & 8 & 2 \\ 0.5 & 0 & 6 & 2 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

ดังนั้นจึงหาระยะห่างระหว่างจุดศูนย์กลางก้อนดนตรีเสมือนกับก้อนปรับเสียงได้ดังนี้

$$\text{ระยะทางV-M} = \sqrt{(2-7)^2 + (2-4)^2 + (1-1)^2}$$

$$= \sqrt{(-5)^2 + (-2)^2 + 0}$$

$$= 5.385 \text{ เซนติเมตร}$$

เมื่อนำค่าระยะทางไปเปรียบเทียบดังตารางที่ 3.1 จะได้ค่าความเร็วจังหวะเสียงเท่ากับ 0

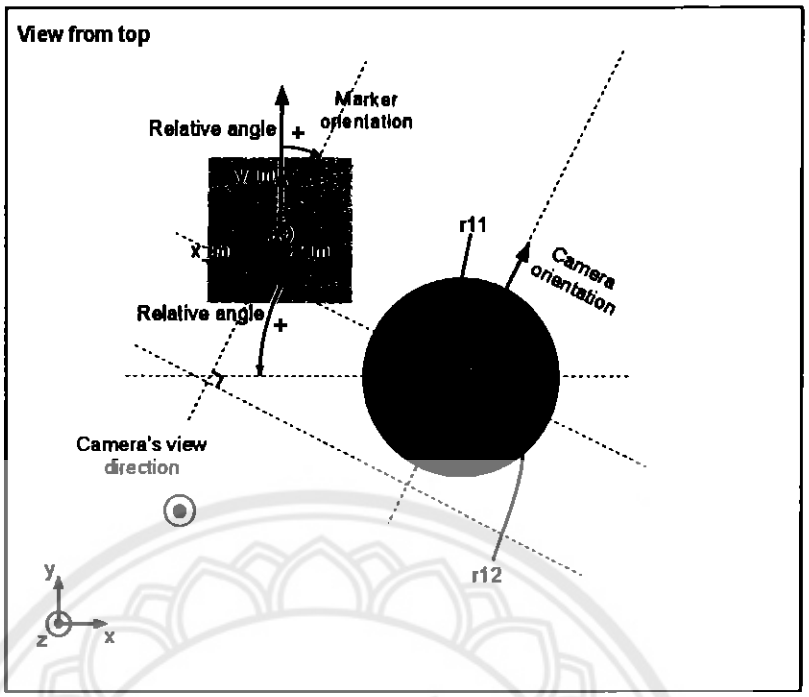
ตารางที่ 3.1 ตารางแสดงค่าความดังเสียง

ระยะทาง (cm)	ระดับความเร็วจังหวะเสียง
น้อยกว่าเท่ากับ 9.99	0
10.00 – 10.99	1
11.00 – 11.99	2
12.00 – 12.99	3
13.00 – 13.99	4
14.00 – 14.99	5
15.00 – 15.99	6
16.00 – 16.99	7
17.00 – 17.99	8
18.00 – 18.99	9
มากกว่าเท่ากับ 19.00	10

3.8.2 ออกแบบการปรับระดับความเร็ว-ช้าของจังหวะเสียงดนตรี

การปรับความเร็วของจังหวะเสียงดนตรีใช้หลักการวัดองศาที่เปลี่ยนไป คือ เมื่อนำก้อนปรับความเร็ว (speed marker) มาวางใกล้ ๆ กับก้อนดนตรีเสมือนที่กำลังเล่นเสียงดนตรีอยู่แล้วหมุนตัวก้อนปรับความเร็วโดยกำหนดให้หมุนตามเข็มนาฬิกา คือ เพิ่มความเร็ว และหมุนทวนเข็มนาฬิกา คือ ลดความเร็ว ดังรูปที่ 3.14

รูปที่ 3.14 แสดงการออกแบบการปรับระดับความเร็ว-ช้าของจังหวะเสียงดนตรี



รูปที่ 3.15 แสดงความสัมพันธ์องศาของกล้องกับสัญลักษณ์

ที่มา: http://omf.mytestbed.net/projects/robot/wiki/RobotControlSoftware?version=51

3.8.2.1 หลักการปรับความเร็ว

โดยมีหลักการนำแมทริกซ์ที่ได้จะมีขนาด [3x4] ไปคำนวณหาค่าองศาที่เปลี่ยนไปของสัญลักษณ์ได้จากสมการนี้

$$\text{relative angle} = 2\tan^{-1}\frac{r_{12}}{r_{11}}$$

$$2\tan^{-1}\frac{r_{12}}{r_{11}} = 2\tan^{-1}\left(\frac{r_{12}}{\sqrt{r_{11}^2+r_{12}^2+r_{11}}}\right)$$

$$\text{speed} = \left((2\tan^{-1}\left(\frac{r_{12}}{\sqrt{r_{11}^2+r_{12}^2+r_{11}}}\right)) * 180/\text{pi}\right) + 180$$

ตัวอย่างการคำนวณ

$$\begin{bmatrix} 0 & 2 & -12 & 7 \\ -2 & 0 & 16 & 7 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

speed = $(2\tan^{-1}(2,0))$

= $((2\tan^{-1}\left(\frac{r_{12}}{\sqrt{r_{11}^2+r_{12}^2+r_{11}}}\right)) * 180/\text{pi}) + 180$

= 180 องศา

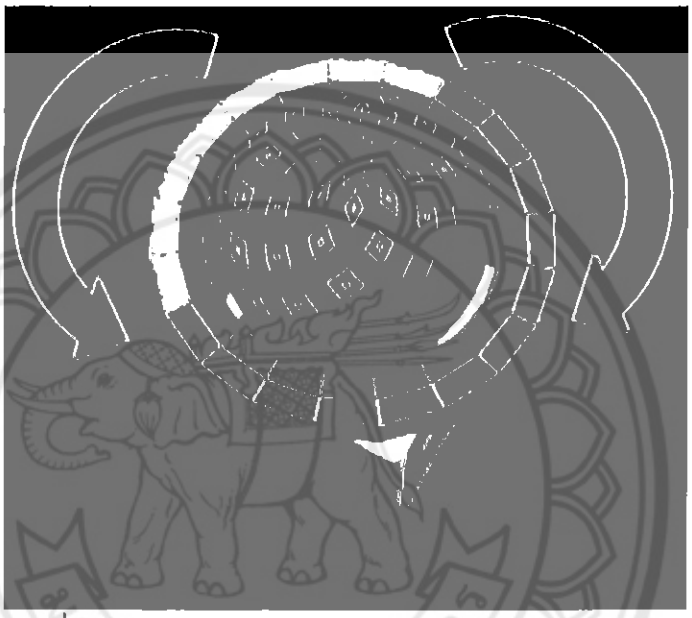
เมื่อนำองศาไปเปรียบเทียบดังตารางที่ 3.2 จะมีค่าความเร็วเท่ากับ 3

ตารางที่ 3.2 ตารางแสดงค่าความเร็วจังหวะเสียง

องศาที่เปลี่ยน (องศา)	ระดับความเร็วจังหวะเสียง
0 – 17	0
18 - 35	1
36-71	2
72 - 107	3
108 - 143	4
144 - 179	5
180 -215	6
216 - 251	7
252 - 287	8
288 - 323	9
324 - 360	10

3.8.3 ออกแบบการปรับเอฟต์เฟกเสียงดนตรี

การปรับเอฟต์เฟกเสียงใช้หลักการวัดองศาที่หมุนคล้ายกับการปรับความเร็ว แต่ส่วนนี้จะวัดว่าเมื่อก้อนดนตรีเสมือนถูกหมุนไป 180 องศาโดยไม่จำกัดว่าหมุนไปในทิศทางตามเข็มนาฬิกาหรือทวนเข็มนาฬิกาเสียงเอฟต์เฟก็จะเปลี่ยนทันที ซึ่งหลักการหาองศาที่เปลี่ยนไปจะใช้การคำนวณเช่นเดียวกับการคำนวณความเร็วเสียง ตัวอย่างดังรูปที่ 2.16



รูปที่ 3.16 แสดงการออกแบบการปรับเอฟเฟกต์เสียงดนตรี

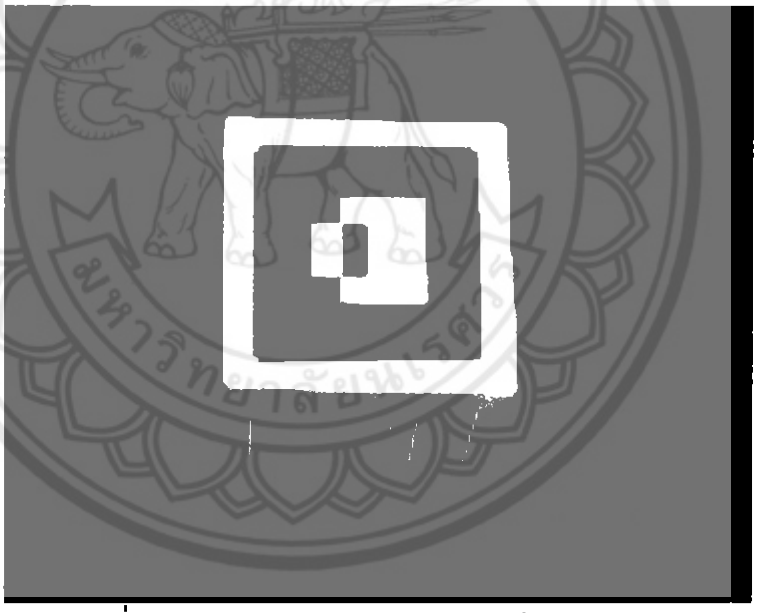
บทที่ 4

ผลการทดลอง

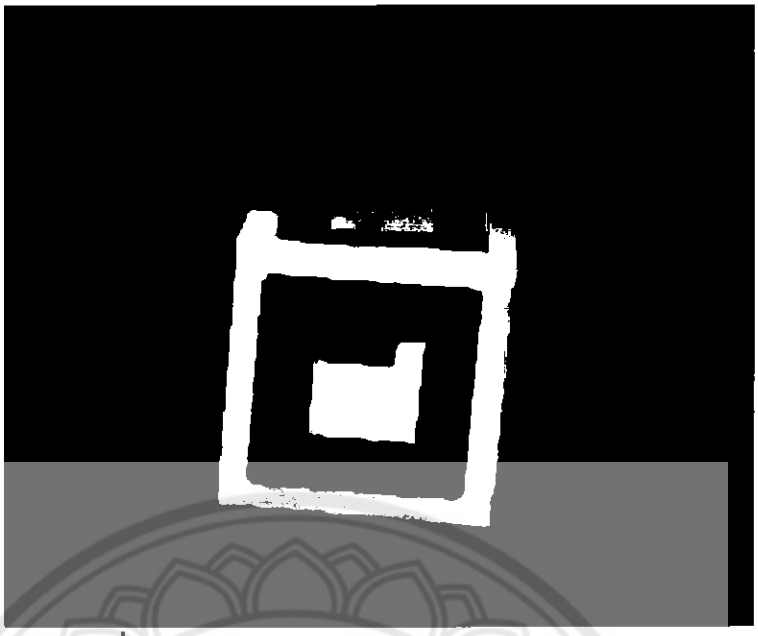
โครงงาน "ก้อนดนตรีเสมือน" เป็นการสร้างเครื่องดนตรีอิเล็กทรอนิกส์ขึ้นมาใหม่ จึงแบ่งการทดลองออกเป็น 2 ส่วน คือ การสร้างกราฟฟิก และการเล่นเสียงดนตรี

4.1 ผลการทดลองการค้นหากรอบสัญลักษณ์ (Marker)

โปรแกรมจะคำนวณหากรอบของพื้นที่สีดำ เมื่อค้นหาเจอสัญลักษณ์จะตีกรอบสี่เหลี่ยมสีเขียวรอบ ๆ รูป ซึ่งโปรแกรมมีความสามารถในการประมวลผลได้ไว ถึงแม้สัญลักษณ์ไม่ได้ตั้งอยู่ในแนวระนาบที่ตั้งฉากกลับกล้องก็สามารถค้นหาสัญลักษณ์ได้ ดังแสดงตัวอย่างการค้นหาสัญลักษณ์องศาต่าง ๆ ดังรูปที่ 4.1 – รูปที่ 4.3



รูปที่ 4.1 แสดงการค้นหาวัตถุสัญลักษณ์กรอบด้านบน



รูปที่ 4.2 แสดงการค้นหาวัตถุสัญลักษณ์กรอบด้านข้าง

รูปที่ 4.3 แสดงการค้นหาวัตถุสัญลักษณ์กรอบด้านล่าง

4.1.1 ผลการทดลองที่ทำให้ไม่สามารถค้นหาสัญลักษณ์ได้

1. เมื่อมีวัตถุใด ๆ มาปิดหรือบังกรอบสี่เหลี่ยมสีดำ ดังรูปที่ 4.4

รูปที่ 4.4 แสดงการค้นหาสัญลักษณ์ไม่เจอเมื่อมีวัตถุมาบังกรอบสี่เหลี่ยมสีดำ

2. เมื่อมีแสงมากเกินไปหรือน้อยเกินไป ดังรูปที่ 4.5

รูปที่ 4.5 แสดงการค้นหาสัญลักษณ์ไม่เจอเมื่อมีแสงตกกระทบวัตถุมากเกินไป

4.2 ผลการทดลองการคำนวณหาพิกัด 3 มิติ บนสัญลักษณ์

เมื่อโปรแกรมหาเส้นรอบรูปได้แล้ว ระบบจะทำการประมาณหาค่าพารามิเตอร์ของสมการเส้นตรงที่แทนเส้นรอบรูปซึ่งเป็นรูปสี่เหลี่ยมทั้ง 4 เส้น หลังจากนั้นระบบจะหาจุดมุม (Corners) ทั้ง 4 จุดของสัญลักษณ์จากจุดตัดของเส้นตรงทั้ง 4 ที่หาได้ ซึ่งจุดมุม 4 จุดนี้จะถูกนำไปใช้ในกระบวนการประมาณตำแหน่ง เป็นขั้นตอนของการคำนวณค่าตำแหน่งเชิง 3 มิติ และสุดท้ายระบบจะทำการเรนเดอร์ภาพสามมิติเป็นส่วนสุดท้ายที่จะทำให้กระบวนการสร้างความจริงเสริมครบถ้วนสมบูรณ์ ซึ่งก็คือ การเพิ่ม (Augment) ข้อมูลที่เราต้องการซึ่งโดยทั่วไปแล้วจะเป็นโมเดล 3 มิติ (3D Model) ลงไปในภาพที่ได้จากกล้องวีดีโอ ณ ตำแหน่งของสัญลักษณ์ที่ตรวจพบจากขั้นตอนการวิเคราะห์ภาพโดยใช้ค่าตำแหน่งเชิง 3 มิติที่คำนวณได้จากขั้นตอนประมาณตำแหน่ง ดังรูปที่ 4.6 – 4.7

รูปที่ 4.6 แสดงการทำเรนเดอร์ภาพ

รูปที่ 4.7 แสดงการสร้างโมเดล 3 มิติ

4.3 ผลการทดลองการค้นหาสัญลักษณ์หลายสัญลักษณ์(Multi Marker)

"ก้อนดนตรีเสมือน" สามารถเล่นเสียงดนตรีได้หลายก้อนพร้อมกัน ดังนั้นจึงต้องออกแบบโปรแกรมเพื่อให้สามารถค้นหาวัตถุหลายสัญลักษณ์ได้

รูปที่ 4.8 แสดงการค้นหาสัญลักษณ์หลายสัญลักษณ์

ก้อนดนตรีเสมือนแต่ละก้อนจะถูกกำหนดให้แสดงกราฟฟิกคนละรูปแบบ ดังในรูปที่ 4.9 เป็นการแสดงผลของก้อนดนตรีเสมือนหลายก้อนที่เปรียบเทียบให้เห็นความแตกต่างระหว่างโลกแห่งความจริง (reality) กับ โลกเสมือน (virtual)

รูปที่ 4.9 แสดงการจำลองความจริงเสมือนและความเป็นจริง

4.4 ผลการทดลองเกี่ยวกับแสง

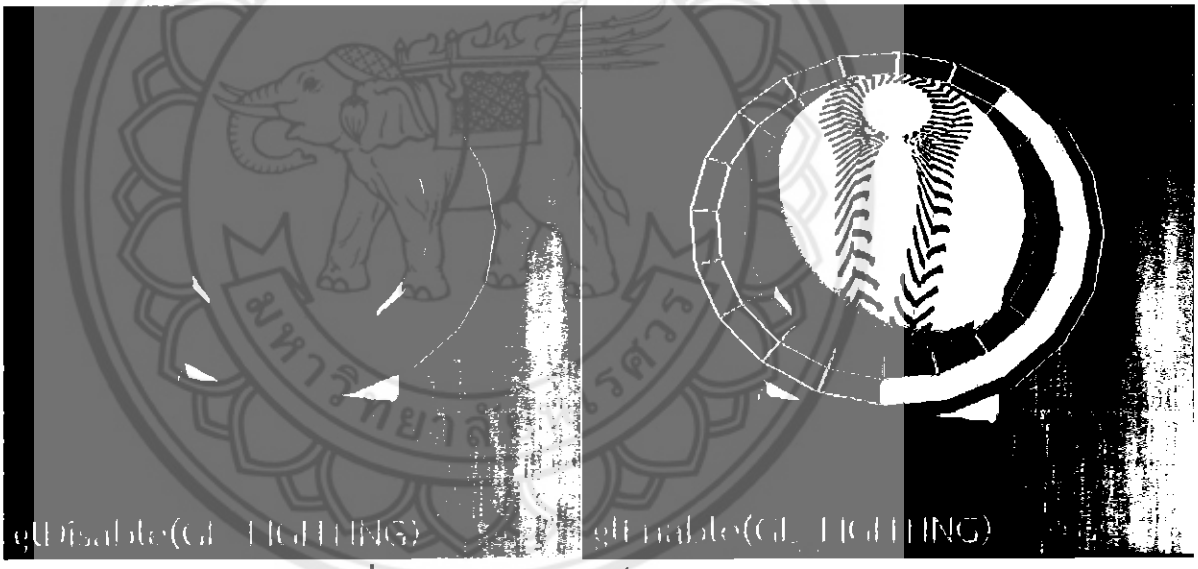
แสงใน OpenGL จำลองแหล่งกำเนิดแสง โดยจำลองจากหลักการธรรมชาติดังนี้

1. แสงจากตัววัตถุเองเช่น ถ่านไฟ ซึ่งจะมีลักษณะเปล่งแสงได้เอง แต่ไม่สว่างมาก (Emission)

2. แสงจากแหล่งกำเนิดแสงอื่นที่เข้มมากเช่น ดวงอาทิตย์ สปอร์ตไลท์ที่ส่องตรงเป็นลำไปที่วัตถุ และเกิดการสะท้อนเข้าตาอย่างเข้มข้น ทำให้มองเห็นเป็นจุดสว่างจ้าในวัตถุ (Specular) ดังจุดสีขาวในรูปข้างบนซ้ายมือ

3. แสงจากแหล่งกำเนิดแสงอื่นที่ส่องตรงไปที่วัตถุ และเกิดการสะท้อนแบบกระจายไม่เข้มข้น (Diffuse)

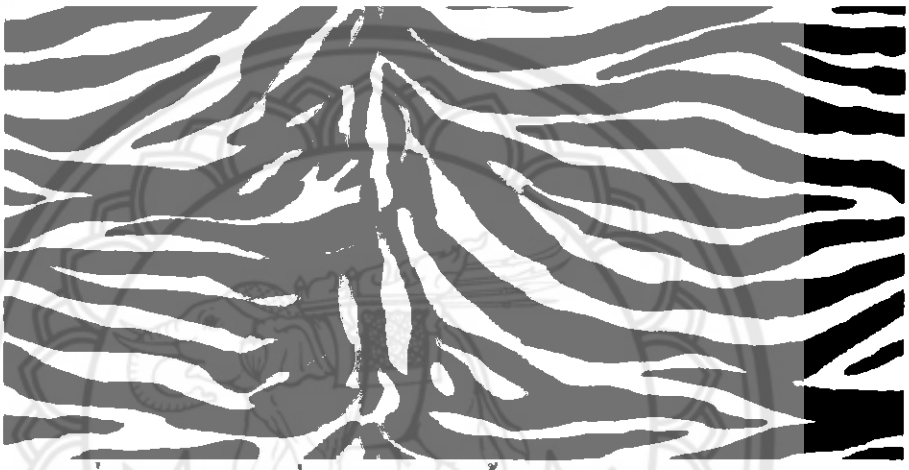
4. แสงที่กระจัดกระจายในอากาศมาจากหลายทิศทาง กระทบวัตถุ แล้วสะท้อนเข้าตา จะเป็นลักษณะแสงจางๆ แต่กระจายทั่วไปในวัตถุ (Ambient)



รูปที่ 4.10 แสดงการใช้คำสั่งเปิด-ปิดแสง

4.5 ผลการทดลองการทำภาพพื้นผิว

การทำภาพพื้นผิว (Texture) คือ การสร้างพื้นผิวของวัตถุโดยการวาดหรือการนำภาพมาห่อหุ้มวัตถุ ซึ่งภาพที่นำมาต้องเป็นภาพบิตแมพแบบ 24 บิต (24-bit bitmap file (.bmp)) และมีขนาดความกว้าง-ยาวต้องเป็นเลขยกกำลังสอง โดยอย่างน้อยต้องมีขนาด 64x64 พิกเซล จากนั้นจะนำภาพมาแปลงเป็นบิตแมพแล้วเอาผลลัพธ์มาปรับให้เข้ากับพื้นผิววัตถุด้วยพิกัดภาพพื้นผิว (texture coordinate)



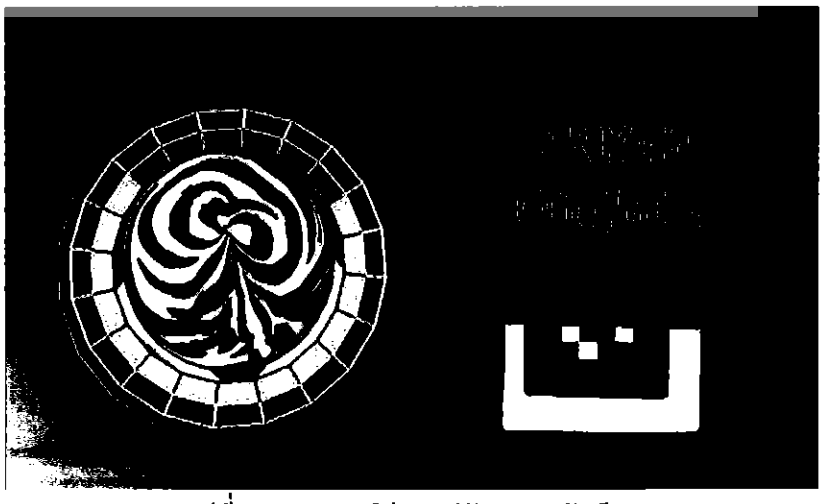
รูปที่ 4.11 แสดงภาพที่ต้องการทำภาพพื้นผิว ขนาดภาพ 521x256 พิกเซล

รูปที่ 4.12 แสดงวัตถุที่ยังไม่ได้ทำการส่งภาพพื้นผิว

รูปที่ 4.13 แสดงผลลัพธ์ที่ได้จากการทำการส่งภาพพื้นผิว

4.6 ผลการทดลองการปรับความดังเสียงดนตรี

การปรับความดังเสียงดนตรีนั้นอาศัยหลักการวัดระยะทางระหว่างก้อนดนตรีเสมือนกับก้อนปรับเสียง (Volume Marker) ซึ่งได้กำหนดให้ก้อนดนตรีเสมือนก้อนหนึ่งมีคุณสมบัติสามารถปรับความดังเสียงดนตรีได้ ดังแสดงในรูปที่ 4.14 จะเห็นว่ามีก้อนดนตรีเสมือนก้อนหนึ่งแสดงภาพกราฟฟิกเป็นรูปสี่เหลี่ยมผืนผ้าที่แสดงข้อความว่า "AR Music Cube Jocker" ซึ่งก้อนนั้นก็คือ ก้อนปรับเสียงในการปรับความดังเสียง ดนตรีนั้นต้องนำก้อนปรับเสียงเลื่อนไป-มา ตามระดับเสียงที่ต้องการ ซึ่งโปรแกรมจะทำการวัดระยะห่างระหว่างก้อนดนตรีเสมือนกับก้อนปรับเสียง หลักการปรับเสียงก็คือ ถ้าก้อนดนตรีเสมือนอยู่ใกล้ก้อนปรับเสียงเสียงจะเบาลง และถ้าก้อนดนตรีเสมือนอยู่ไกลก้อนปรับเสียงเสียงจะดังขึ้น พร้อมทั้งแสดงแถบวงกลมสีแดงวงด้านในแทนระดับความดังเสียงดนตรีที่มีทั้งหมด 20 แถบ



รูปที่ 4.14 แสดงวิธีการปรับความดังเสียง

เมื่อเลื่อนตัวก้อนปรับเสียงห่างออกไปมากกว่า 19.00 เซนติเมตร จะทำให้ได้ความดังสูงสุด ซึ่งเห็นได้จากแถบวงกลมสีแดงที่แสดงทั้งหมด 20 แถบ ดังรูปที่ 4.15

รูปที่ 4.15 แสดงความดังเสียงดนตรีสูงสุด

เมื่อนำก้อนดนตรีเสมือนส่องผ่านกล้องจะแสดงความดังเสียงเริ่มต้นเป็นครึ่งหนึ่งซึ่งจะเห็นได้จากแถบวงกลมสีแดงที่แสดงจำนวน 10 แถบ ดังรูปที่ 4.16

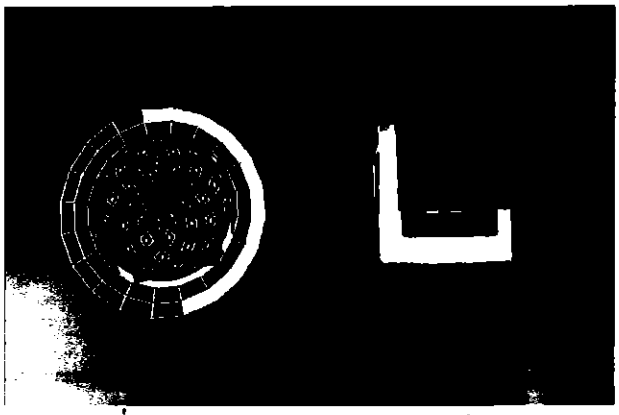
รูปที่ 4.16 แสดงความดังเสียงดนตรีเริ่มต้น

เมื่อเลื่อนตัวก้อนปรับเสียงเข้าไปใกล้ให้มีความห่างกับก้อนดนตรีเสมือนต่ำกว่า 9.99 เซนติเมตร จะทำให้ได้ความดังต่ำสุด ซึ่งเห็นได้จากแถบวงกลมสีแดงที่ไม่มีการแสดงเลยสักแถบ ดังรูปที่ 4.17

รูปที่ 4.17 แสดงความดังเสียงดนตรีต่ำสุด

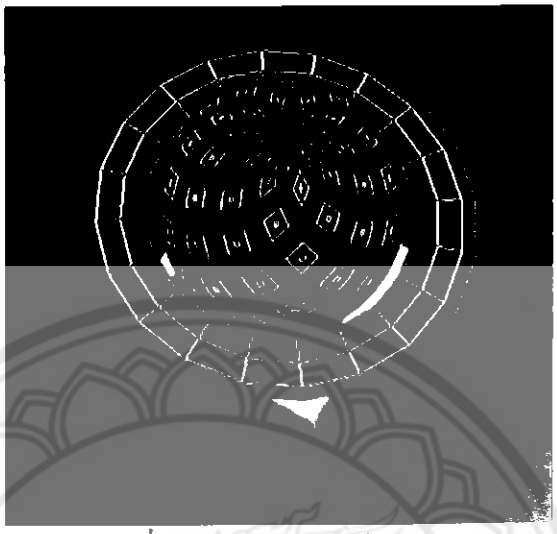
4.7 ผลการทดลองการปรับความเร็วเสียงดนตรี

การปรับความเร็วเสียงนั้นอาศัยหลักการหมุนองศาที่เปลี่ยนไปของก้อนปรับความเร็ว (Speed marker) ซึ่งได้กำหนดให้ก้อนดนตรีเสมือนก้อนหนึ่งมีคุณสมบัติสามารถปรับความเร็วเสียงได้ ดังแสดงในรูปที่ 4.18 จะเห็นว่ามีก้อนดนตรีเสมือนก้อนหนึ่งแสดงภาพกราฟฟิกเป็นรูปเกียร์ ซึ่งก้อนนั้นก็คือก้อนปรับความเร็ว ในการปรับนั้นความเร็วต้องนำก้อนปรับความเร็วมาวางใกล้ ๆ กับก้อนดนตรีเสมือนที่กำลังเล่นเสียงดนตรีอยู่ แล้วหมุนตัวก้อนปรับความเร็วโดยกำหนดให้หมุนตามเข็มนาฬิกา คือ เพิ่มความเร็ว และหมุนทวนเข็มนาฬิกา คือ ลดความเร็ว พร้อมทั้งจะแสดงระดับความเร็วเสียงด้วยแถบวงกลมสีน้ำเงิน



รูปที่ 4.18 แสดงวิธีการปรับความเร็วเสียง

เมื่อหมุนก้อนปรับความเร็วตามเข็มนาฬิกาให้เปลี่ยนไป 324 - 360 องศา จะทำให้ได้ความเร็วเสียงเป็นสูงสุด ซึ่งเห็นได้จากแถบวงกลมสีน้ำเงินที่แสดงทั้งหมด 20 แถบ ดังรูปที่ 4.19



รูปที่ 4.19 แสดงความเร็วสูงสุด

เมื่อนำก้อนดนตรีเสมือนส่องผ่านกล้องจะแสดงความเร็วเสียงเริ่มต้นจะถูกกำหนดให้เป็นครึ่งหนึ่งซึ่งเห็นได้จากแถบวงกลมสีน้ำเงินที่แสดงจำนวน 10 แถบ ดังรูปที่ 4.20

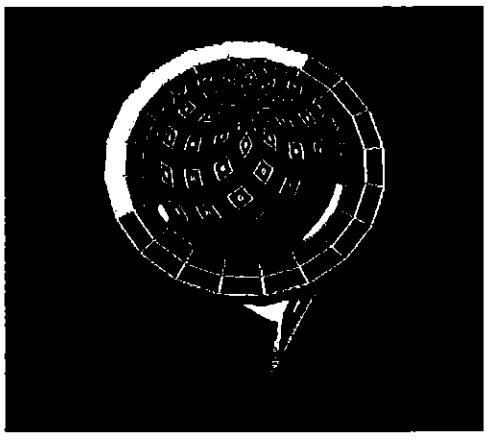
รูปที่ 4.20 แสดงความเร็วเริ่มต้น

เมื่อหมุนก้อนปรับความเร็วทวนเข็มนาฬิกาให้เปลี่ยนไป 324 - 360 องศา จะทำให้ได้ความเร็วเสียงเป็นต่ำสุด ซึ่งเห็นได้จากแถบวงกลมสีน้ำเงินที่ไม่มีการแสดงเลยทั้งหมด 20 แถบ ดังรูปที่ 4.21

รูปที่ 4.21 แสดงความเร็วต่ำสุด

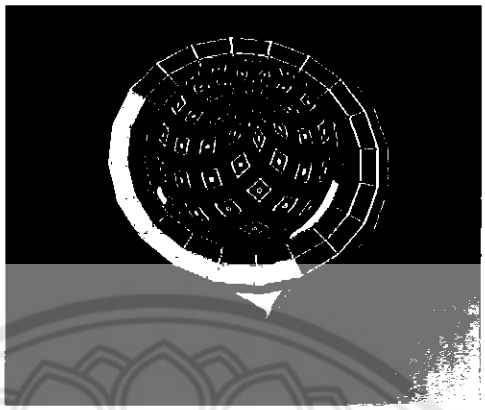
4.8 ผลการทดลองการปรับเอฟเฟกต์เสียงดนตรี

ก้อนดนตรีแต่ละก้อนสามารถปรับเอฟเฟกซ์เสียงได้โดยการหมุนก้อนนั้น ๆ ตามหลักการคล้าย ๆ การหมุนปรับความเร็วเสียงดนตรี ซึ่งจะมีวงกลมวงนอกสุดแสดงว่าขณะนั้นคือเอฟเฟกต์เสียงแบบใด โดยที่จะแสดงเป็นสีเขียวหากก้อนดนตรีเสมือนไม่มีการหมุนก็จะเล่นเสียงปกติ แต่ถ้าก้อนดนตรีเสมือนถูกหมุนจะเล่นเสียงเอฟเฟกต์เสียงเพี้ยน (distortion) พร้อมกับแสดงวงกลมเป็นสีแดง เมื่อนำก้อนดนตรีเสมือนไปส่องกล้องจะเล่นเสียงปกติ ซึ่งจะแสดงเป็นวงกลมสีเขียว ดังในรูปที่ 4.22



รูปที่ 4.22 แสดงเอฟเฟกต์เสียงปกติจะแสดงเป็นสีเขียว

เมื่อก้อนดนตรีเสมือนถูกหมุนในทิศทางใดก็ตามที่ทำให้องศาเปลี่ยนไป 180 องศา จะทำให้เปลี่ยนเอฟเฟกต์เสียงเพี้ยนและวงกลมวงนอกสุดจะเปลี่ยนเป็นสีแดง ดังในรูปที่ 4.23



รูปที่ 4.23 แสดงเอฟเฟกต์เสียงเพี้ยนจะแสดงเป็นสีแดง

4.9 สรุปผลการทดลอง

จากผลการทดลอง "ก้อนดนตรีเสมือน" สามารถเล่นเสียงดนตรีและแสดงผลกราฟฟิกได้หลายก้อนพร้อมกัน และสามารถควบคุมระดับความดังของเสียง, ความเร็วของเสียง, และการปรับเอฟเฟกต์ (Effect) ของเสียง ผ่านทางสัญลักษณ์ได้ ดังแสดงในรูปที่ 2.24

รูปที่ 4.24 การแสดงผลทั้งหมดของโปรแกรม

บทที่ 5

สรุปผลและข้อเสนอแนะ

บทนี้กล่าวถึงบทสรุป ปัญหาที่พบและข้อเสนอแนะของโครงงาน "ก้อนดนตรีเสมือน" (AR Music Cube Jocker) เพื่อให้เกิดความเข้าใจในโครงงานและนำไปพัฒนาโครงงานนี้ต่อไป

5.1 สรุปผลการทดลอง

โครงงานนี้จัดทำขึ้นเพื่อนำเทคโนโลยีความจริงเสริม (Augmented Reality) มารวมกับเสียงดนตรีเพื่อให้ความบันเทิง และให้ผู้ใช้ได้รับประสบการณ์ใหม่ ๆ เกี่ยวกับเสียงดนตรี ผู้ใช้สามารถควบคุมระดับความดังของเสียง ความเร็วของเสียง และการปรับเอฟเฟกต์ (Effect) ของเสียงผ่านทางสัญลักษณ์ (Marker) และดูภาพเสมือนจริงที่ผสมผสานภาพในโลกแบบเวลาจริง (Real time) ทำให้ผู้ใช้ได้มีปฏิสัมพันธ์โดยตรงกับภาพเสมือนจริงและผลรับของเสียง ทำให้ผู้ใช้เกิดประสบการณ์ใหม่ ๆ และมีความสนใจในเทคโนโลยีความจริงเสริมมากขึ้น

5.2 ปัญหาที่พบ

จากการทำลองโครงงานนี้ทำให้พบปัญหาใหญ่ๆ 2 ส่วนได้แก่การศึกษาและใช้งานไลบรารี่ต่างๆที่ใช้ในโครงงานนี้ และขั้นตอนการทดลอง

1. ปัญหาการใช้งานไลบรารี่ ARToolKit, OpenGL และ irrKlang ที่ต้องใช้เวลาศึกษาพอสมควร เนื่องจากไม่เคยใช้ไลบรารี่เหล่านี้มาก่อน

2. อุปกรณ์ที่ใช้ในการทดลองของโครงงานไม่ค่อยอำนวยความสะดวก จึงทำให้ขั้นตอนการทดลองเกิดการผิดพลาดบ้าง

3. สภาพแวดล้อมในการทดลองไม่เอื้ออำนวย เช่น เรื่องแสง เป็นต้น ทำให้เกิดการติดขัดในการทดลองของโครงงาน

5.3 ข้อเสนอแนะและวิธีการแก้ไขปัญหา

จากปัญหาที่พบในโครงงานนี้ ทางผู้จัดทำขอเสนอข้อเสนอแนะ และวิธีการแก้ไขปัญหาดังนี้

1. วางแผนการศึกษาไลบรารี่แต่ละตัวให้เข้าใจ และศึกษาตามเว็บไซต์ต่างประเทศเพราะมีข้อมูลและวิธีการใช้งานไลบรารี่มากกว่าเว็บไซต์ในประเทศไทย
2. จัดหาอุปกรณ์ต่างๆ ดังนี้
   2.1 กล้องเว็บแคม
   2.2 ขาตั้งกล้องเว็บแคม
   2.3 ฟิวเจอร์บอร์ดเพื่อนำมาเป็นพื้นในการทดลอง เพื่อลดปัญหาการค้นหาสัญลักษณ์
   2.4 สัญลักษณ์ ให้ออกแบบเองให้มีความแตกต่างกันพอสมควร
3. หาสถานที่เพื่อทำการทดลองที่มีแสงไม่มากและไม่น้อยเกินไป เพื่อลดปัญหาการค้นหาสัญลักษณ์ (Marker)

เอกสารอ้างอิง

[1] ดุริยางคศิลป์และนาฏศิลป์ปริทัศน์.สืบค้นเมื่อ 14 สิงหาคม 2554, จาก http://cyberclass.msu.ac.th/cyberclass/cyberclass-uploads/libs/document/allunit01_4903.swf.

[2] AUGMENTED REALITY.สืบค้นเมื่อ 14 สิงหาคม 2554, จาก http://msmisthammasat.blogspot.com/2011/01/augmented-reality.html.

[3] เทคโนโลยี Augmented Reality คืออะไร?.สืบค้นเมื่อ 14 สิงหาคม 2554, จาก http://learngears.com/2011/08/10.

[4] Augmented Reality โลกเสมือนผสานโลกจริง.สืบค้นเมื่อ 14 สิงหาคม 2554, จาก http://www.edu.nu.ac.th/wiwatm.

[5] เทคโนโลยี่ :ARToolKitเครื่องมือช่วยวาดวัตถุ 3D ประกอบฉาก.สืบค้นเมื่อ 14 สิงหาคม 2554, จาก http://board.palungjit.com/f6/เทคโนโลยี่-artoolkit-เครื่องมือช่วยวาดวัตถุ-3d-ประกอบฉาก-118596.html.

[6] OpenGL (Open source of Graphic Library).Retrieved August 15, 2011, fromhttp://202.44.14.219/~isurnich/ftp/CG/html/OpenGL.htm.

[7] OpenAL. สืบค้นเมื่อ 15 สิงหาคม 2554, จาก http://th.wikipedia.org/wiki/OpenAL.

[8] ไฟล์เสียงต่าง ๆ ที่ควรรู้จัก. สืบค้นเมื่อ 15 สิงหาคม 2554, จาก http://www.vecthai.com/forums/index.php?topic=324.0.

[9] ARToolKitDevelopment Principles.Retrieved August 15,2011, จาก http://www.hitl.washington.edu/artoolkit/documentation/devprinciple.htm.

[10] Augmented Reality Model for Fungi Simulation การจำลองตัวแบบภาพเสมือนสำหรับการจำลองอาณาจักรเห็ดรา.สืบค้นเมื่อ 9กันยายน2554, จาก http://202.28.94.53/e-Project/ProjectWeb.php?id=74.

[11] การศึกษาเทคโนโลยีออคเมนต์เตดเรียลริตี้ กรณีศึกษาพัฒนาเกมส์ "เมมการ์ด".ผู้จัดทำ วสันต์เกียรติแสงทอง,พรรษพลพรหมมาศและอนุวัตรเฉลิมสกุลกิจ.สืบค้นเมื่อ 3 ธันวาคม 2554, จาก facstaff.swu.ac.th/praditm/CP499_2552_AR.pdf.

[12] พนิดาตันศิริ. "โลกเสมือนผสานโลกจริง Augmented Reality." สืบค้นเมื่อ 5 มีนาคม 2555, จาก www.bu.ac.th/knowledgecenter/executive_journal/30_2/.../aw28.pdf.

[13] ดร.วิวัฒน์. "Augmented Reality." สืบค้นเมื่อ 5 มีนาคม 2555, จาก http://www.edu.nu.ac.th/wiwatm.

[14] "RGB คืออะไร อาร์จีบี คือระบบสีของแสง". สืบค้นเมื่อ 5 มีนาคม 2555, จาก http://www.mindphp.com/คู่มือ/73-คืออะไร/2172-rgb-คืออะไร.html.

[15] "Introduction to Machine Vision". Retrieved march 5, 2012, fromcdn.researchers.in.th/assets/media/files/000/.../original_chapter01.pdf.

[16] SalimBensiali."Control Software for the OMF-enabled Robot".Retrieved march 5, 2012, from http://omf.mytestbed.net/projects/robot/wiki/RobotControlSoftware?version=51.

ภาคผนวก

1. คู่มือการใช้งาน “ก้อนดนตรีเสมือน”

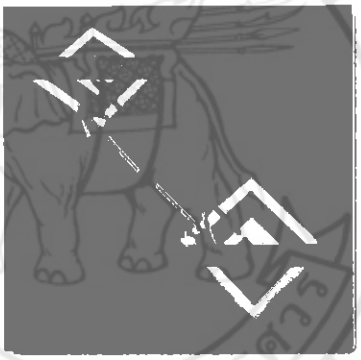
1. รันไฟล์ ARCubeMusicJocker.exe

2. ปรับเลือกการตั้งค่ากล้องวีดีโอ

3. พร้อมใช้งาน สามารถนำก้อนดนตรีเสมือนมาส่องกล้องได้เลย

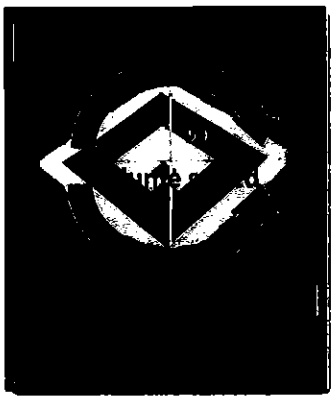
1.1 ฟังก์ชั่นในการใช้งาน

การปรับความดังเสียง



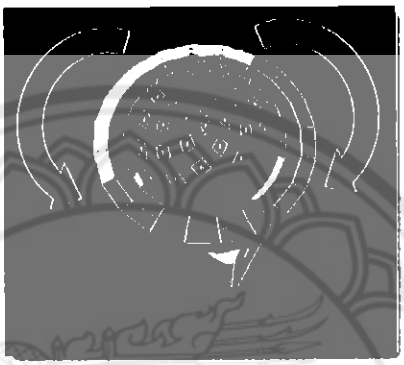
วิธีการ : ใช้วัตถุสัญลักษณ์ก้อน volume control เลื่อนไปมาก็สามารถปรับเสียงได้เลย

การปรับความเร็วเสียง



วิธีการ : ใช้วัตถุสัญลักษณ์ก้อน volume speed ไปวางไว้ใกล้ ๆ กับก้อนที่ต้องการปรับความเร็วเสียง แล้วทำการหมุนก้อน volume speed ตามเข็มนาฬิกาเพื่อเพิ่มความเร็ว หมุนทวนเข็มนาฬิกาเพื่อลดความเร็ว

การปรับเอฟต์เฟก



วิธีการ : หมุนก้อนดนตรีเสมือนที่ต้องการไปตามเข็มนาฬิกาหรือทวนเข็มนาฬิกา**เอฟต์เฟกเสียงจะเปลี่ยนทันที**

1.2 อุปกรณ์ที่ใช้

1. ก้อนดนตรีเสมือนหรือวัตถุสัญลักษณ์
2. กล้อง
3. คอมพิวเตอร์
4. โปรแกรม

2. การติดตั้ง ARToolKit และการ Test กล้อง

download Microsoft Visual Studio 2010 หรือ version ไหนก็ได้ และ installให้เรียบร้อย

download DSVL-0.0.8b.zip จาก http://sourceforge.net/projects/artoolkit/files/artoolkit/

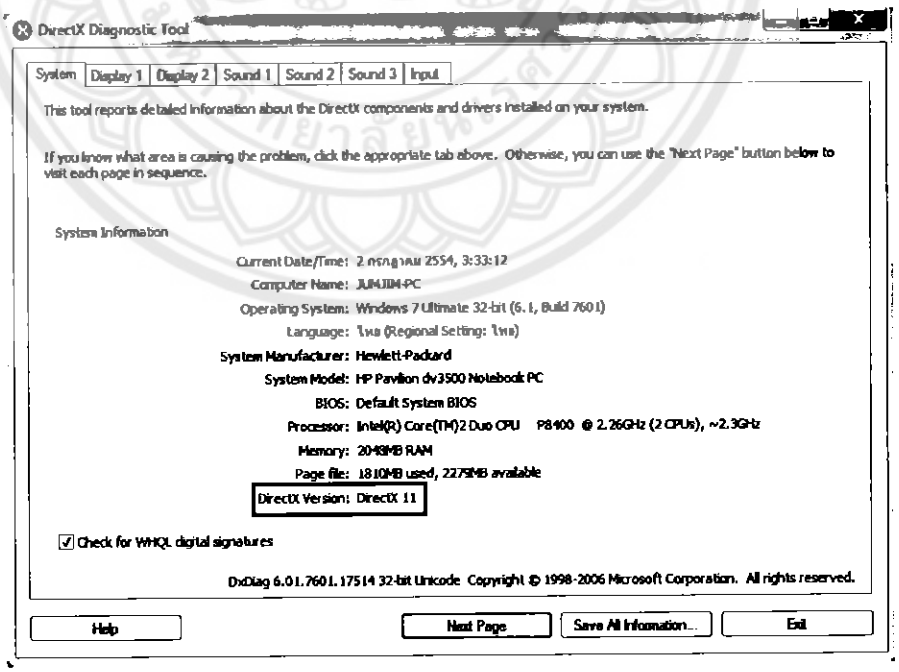
(download มาเตรียมไว้ก่อน)

download GLUT จาก http://www.xmission.com/~nate/glut.html เลือก glut-3.7.6-bin.zip (117 KB) กด save แล้วแตกไฟล์ จากนั้น

3.1 เอาไฟล์ glut32.dll ไปไว้ใน c:\windows\system32

3.2 สร้างโฟลเดอร์ gl , เอาไฟล์ glut.h ไปไว้ในโฟลเดอร์ gl แล้ว เอาโฟลเดอร์ gl ไปไว้ใน C:\Program Files\Microsoft Visual Studio 10.0\VC\include

3.3 เอาไฟล์ glut32.lib ไปไว้ใน C:\Program Files\Microsoft Visual Studio 10.0\VC\lib เอาไฟล์ MSVCP71D.dll และ MSVCR71D.dll ที่ส่งไปให้ ไปไว้ที่ c:\windows\system32 ตรวจสอบเวอร์ชั่น ของ directX ที่มีในเครื่องของเราโดย start >> run >>dxdiag >>OK directX version จะต้องมากกว่า 9.0b ขึ้นไป



ต่อสายกล้อง ทดสอบกล้องว่าใช้งานได้ป่าว

download ARToolKit แตกไฟล์ ไว้ที่สำหรับใช้งาน เช่น C:\Program Files\ARToolKit

แตกไฟล์ DSVL-0.0.8b.zip ที่โหลดมา ใส่ไว้ใน C:\Program Files\ARToolKit ที่เราเพิ่งแตกเมื่อกี้ (ข้อ6) (Make sure that the directory is named "DSVL".)

Copy the files DSVL.dll and DSVLd.dll จาก C:\Program Files\ARToolKit\DSVL\bin ไปไว้ที่ C:\Program Files\ARToolKit\bin

รันสคริปต์ C:\Program Files\ARToolKit\Configure.win32.bat

เปิด microsoft visual studio เลือก open project เลือก ARToolKit (type= Microsoft Visual Studio Solution Object) จาก C:\Program Files\ARToolKit

Build >> Build Solution

เสร็จต้อง build succeeded ,0 failed

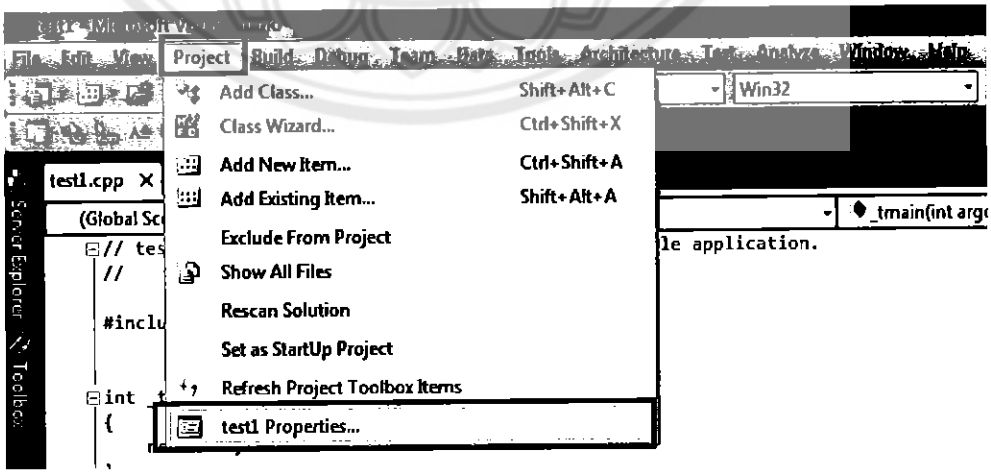
***หมายเหตุ ถ้า Build ไม่สำเร็จ ให้ทำดังนี้

เปิด microsoft visual studio ขึ้นมา เลือก new project ขึ้นมา

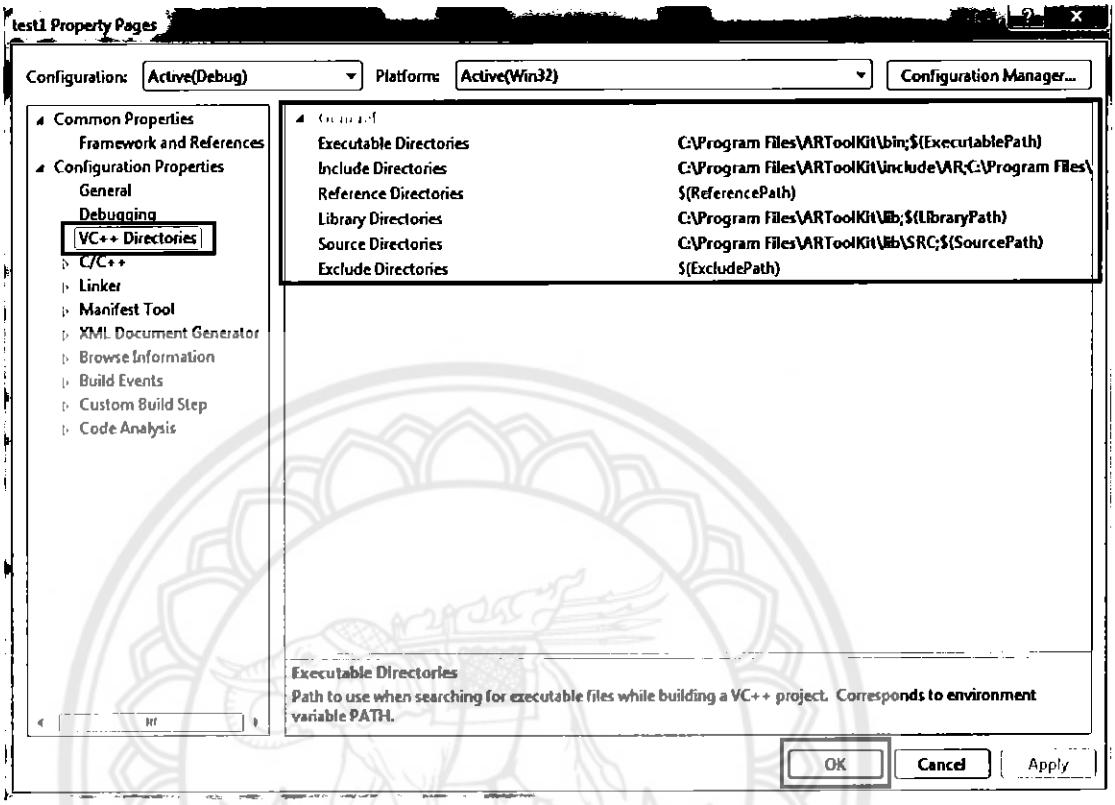
Copy Code จากไฟล์ simpleLite.c ใน ARToolKit (type= Microsoft Visual Studio Solution Object) มาวางที่ project ของเรา

Set ค่าใน microsoft visual studio ดังนี้

ไปที่ Project >> ชื่อไฟล์ Properties…



เลือก VC++ Directories และทำการเพิ่มข้อมูลในหน้าต่างด้านขวาดังนี้



Executable Directories : C:\Program Files\ARToolKit\bin

Include Directories : C:\Program Files\ARToolKit\include\AR และ C:\Program Files\ARToolKit\include

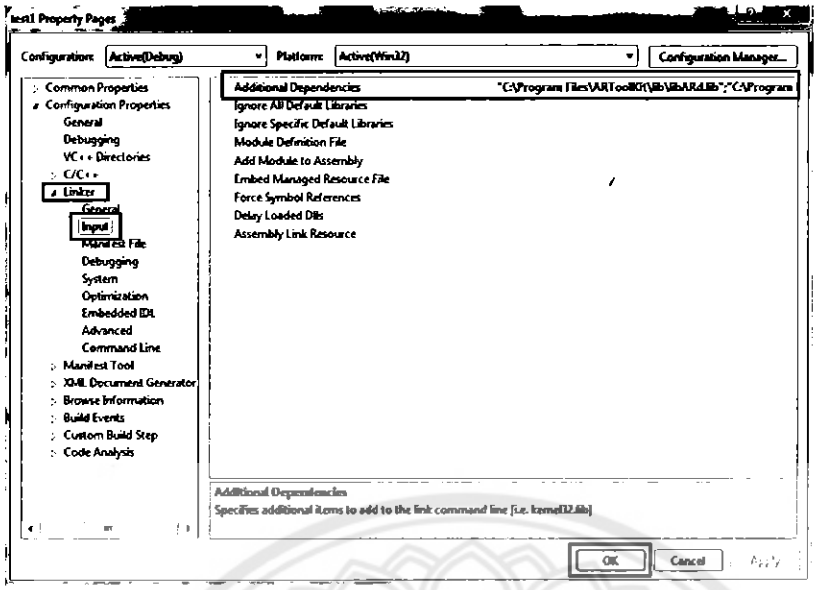
Reference Directories : เหมือนเดิม

Library Directories : C:\Program Files\ARToolKit\lib

Source Directories : C:\Program Files\ARToolKit\lib\SRC

Exclude Directories : เหมือนเดิม

จากนั้นไปเลือกที่ Linker >> Input



แล้วใส่ข้อมูลที่ Additional Dependencies ดังนี้

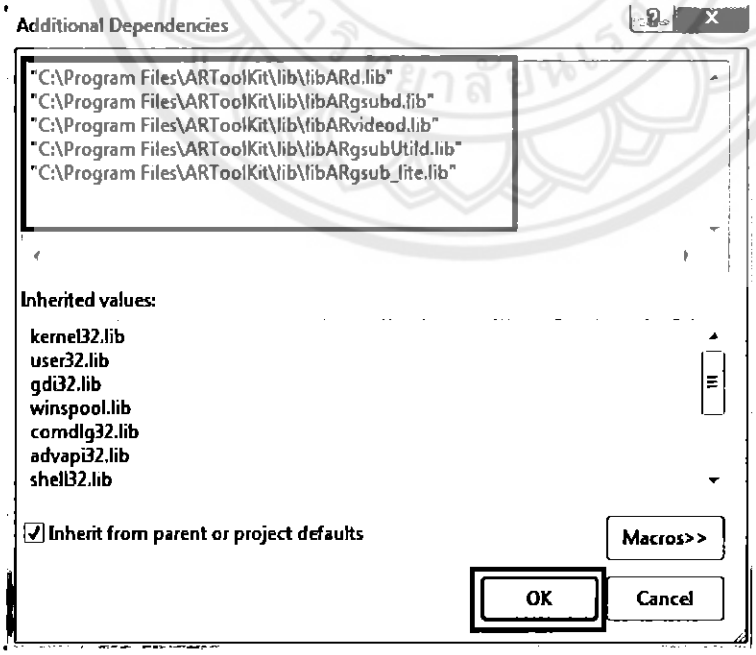
C:\Program Files\ARToolKit\lib\libARd.lib

C:\Program Files\ARToolKit\lib\libARgsubd.lib

C:\Program Files\ARToolKit\lib\libARvideod.lib

C:\Program Files\ARToolKit\lib\libARgsubUtild.lib

C:\Program Files\ARToolKit\lib\libARgsub_lite.lib



และสุดท้ายให้ copy เอา folder data , DSVL.dll, DSVLd.dll , js32.dll , libARvideo.dll และ libARvideod.dll จาก C:\Program Files\ARToolKit\bin ไปไว้ที่ไฟล์ .exe ของโปรเจ็คที่เราจะรัน เช่น C:\Users\jumjim\Documents\Visual Studio 2010\Projects\test1\Debug

เท่านี้ก็เรียบร้อย!!

จากนั้นก็ Build >> Build Solution โปรเจ็คของเรา แล้วก็รัน ทดลองเล่น AR ได้ตามสบาย

** สามารถปริ้นตัว marker ได้ที่ C:\Program Files\ARToolKit\patterns

3. การติดตั้ง irrKlang

1. เข้าไปที่ C/C++ -> General -> Additional Iclude Directories แล้ว add

C:\irrKlang-1.3.0b\irrKlang-1.3.0\include

C:\irrKlang-1.3.0b\irrKlang-1.3.0\plugins\ikpMP3

C:\irrKlang-1.3.0b\irrKlang-1.3.0\plugins\ikpMP3\decoder

2. เข้าไปที่ Linker -> General -> Additional Library Directories แล้ว add

C:\irrKlang-1.3.0b\irrKlang-1.3.0\lib\Win32-visualStudio

4. การติดตั้ง OpenGL

Install Visual C++ 2010 Express

- Go to http://www.microsoft.com/express/

  and download the VC++ 2010 Express
- Install and Register the product

Install GLUT Library

- Go to http://www.xmission.com/~nate/glut.html

  and download glut-3.7.6-bin.zip (117 KB)
- Copy glut32.dll to "C:\Windows\System32"
- Copy glut32.lib to "C:\Program Files\Microsoft Visual Studio 10.0\VC\lib"
- Copy glut.h to "C:\Program Files\Microsoft Visual Studio 10.0\VC\Include\gl"

  Note that you may have to create the gl directory.

- Note that for newer and more active version of GLUT, try *freeglut* at http://freeglut.sourceforge.net and http://www.transmissionzero.co.uk/software/freeglut-devel/

เรียบร้อย!!