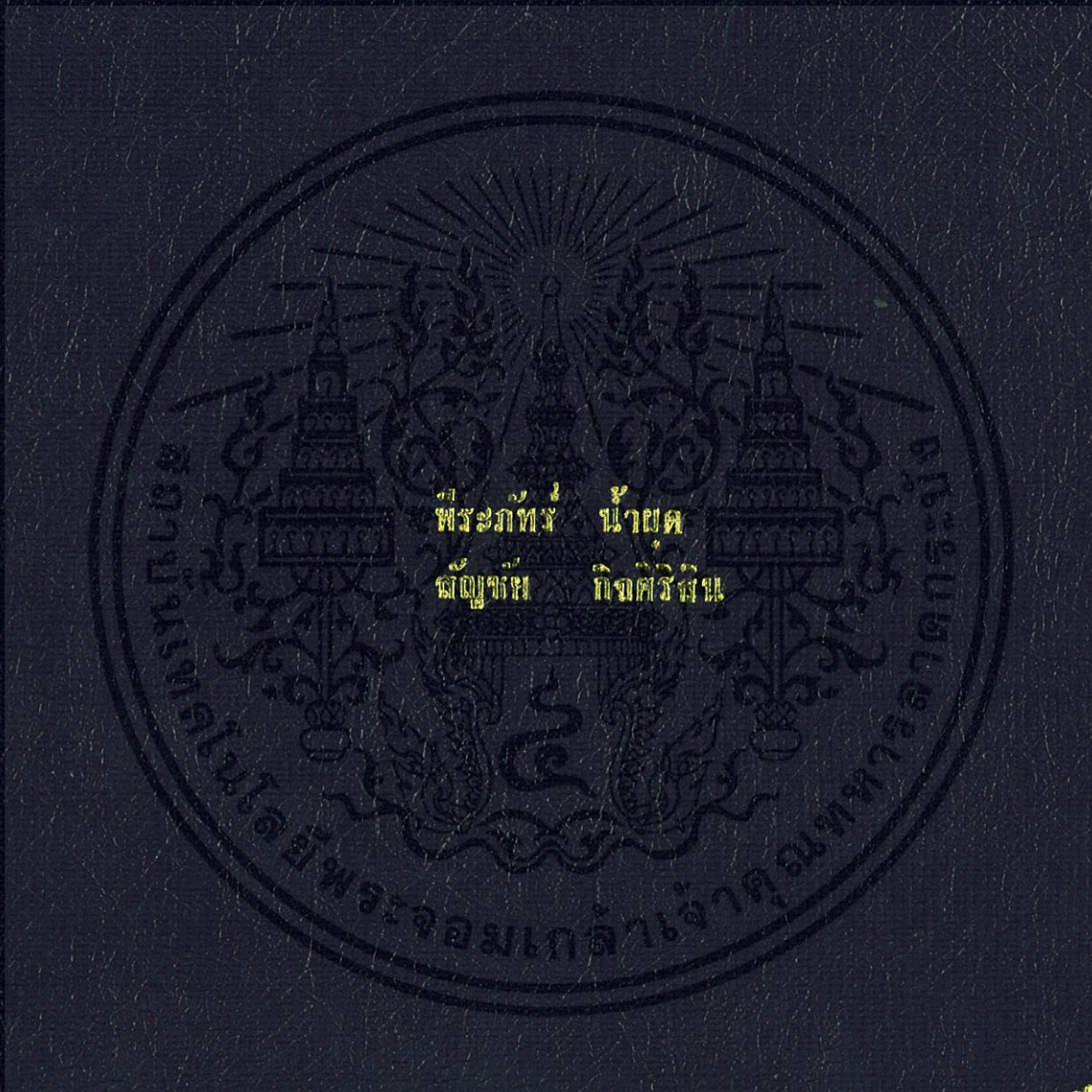


การพัฒนาแอปพลิเคชันชนิด AUGMENTED REALITY
AUGMENTED REALITY APPLICATION DEVELOPMENT



พระภิกษุ นายน
ตัญญาท กิจศรีสิน

วิทยานิพนธ์นี้จัดทำขึ้นเป็นต้นฉบับที่งานของภาควิชาเทคโนโลยีสารสนเทศ มหาวิทยาลัยราชภัฏบุรีรัมย์
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2555

การพัฒนาแอปพลิเคชันชนิด AUGMENTED REALITY
AUGMENTED REALITY APPLICATION DEVELOPMENT



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับนำไปใช้ในการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปีการศึกษา 2555

ปริญญาโทปีการศึกษา 2555

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาแอปพลิเคชันชนิด Augmented Reality

Augmented Reality Application Development

ผู้จัดทำ

1. นายพีระภัทร์ น้าผุด รหัสนักศึกษา 52010865

2. นายสัณชัย กิจศิริสิน รหัสนักศึกษา 52011259



..... อาจารย์ที่ปรึกษา
(ดร.วรวัดน์ ลิ้มโกศา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาแอปพลิเคชันชนิด Augmented Reality

นายพีระภัทร์	น้ำผุด	52010865
นายสฤษชัย	กิจศิริสิน	52011259
ดร.วรวัฒน์	ลี้มโสภา	อาจารย์ที่ปรึกษา
ปีการศึกษา 2555		

บทคัดย่อ

เทคโนโลยีเสมือนจริง (Augmented Reality : AR) เป็นประเภทหนึ่งของเทคโนโลยีความเสมือนจริง (Virtual Reality : VR) ที่มีการนำระบบความจริงมาผนวกกับเทคโนโลยีภาพ เพื่อสร้างสิ่งที่เสมือนจริงขึ้นมาแสดงผลผ่านทางหน้าจอคอมพิวเตอร์หรืออุปกรณ์แสดงผลอื่น ๆ

ในโครงการจะกล่าวถึงทฤษฎีต่าง ๆ และขั้นตอนในการสร้างเทคโนโลยีเสมือนจริงขึ้นมา ซึ่งกระบวนการของเทคโนโลยีนี้ก็คือ การวิเคราะห์ภาพ (Image Analysis) ซึ่งเป็นขั้นตอนการค้นหา Marker จากภาพที่ได้จากกล้อง แล้วสืบค้นจากฐานข้อมูล (Marker Database) ที่มีการเก็บข้อมูลขนาดและรูปแบบของ Marker เพื่อนำมาวิเคราะห์รูปแบบของ Marker การคำนวณค่าตำแหน่งเชิงสามมิติ (Pose Estimation) ของ Marker เทียบกับกล้อง และกระบวนการสร้างภาพสองมิติจากโมเดลสามมิติ (3D Rendering) ซึ่งเป็นการเพิ่มข้อมูลเข้าไปในภาพ โดยใช้ค่าตำแหน่งเชิงสามมิติที่คำนวณได้ จนได้ภาพเสมือนจริง โดยใช้ไลบรารีหลักในการช่วยพัฒนาแอปพลิเคชันคือ FLARToolKit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Augmented Reality Application Development

Mr. Peerapat Numpud 52010865

Mr. Sanshai Kitsirisin 51011259

Dr. Voravat Limpoka Advisor

Academic Year 2012

ABSTRACT

Augmented Reality (AR) technology is one type of Virtual Reality (VR) technology which combines reality system and image technology to create a virtual reality display via a computer screen or other display devices.

This study discusses the various theories of Augmented Reality and the process of creating augmented reality technology. The process of this technology is image analysis which implements the Marker detection process from camera images, and search Marker from the Marker database, which stores the size and format of the data to analyze the pattern of the marker. The analysis includes Pose Estimation of the camera comparative to the marker, and 3D Rendering, which creates two-dimensional images from three-dimensional models), adding the data into images by using the result of Pose Estimation so that augmented reality can be created. The main library used is FLARToolKit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการฉบับนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความช่วยเหลือและความร่วมมือจากบุคคลหลาย ๆ ฝ่ายเป็นอย่างดี จึงใคร่ขอขอบพระคุณบุคคลดังต่อไปนี้

ขอขอบพระคุณ ดร.วรวัฒน์ ลิ้มโกคา อาจารย์ที่ปรึกษาปริญญาโทที่กรุณาให้คำปรึกษา และชี้แนะแนวทางต่าง ๆ ในการดำเนินโครงการ การแก้ไขปัญหาและข้อผิดพลาดต่าง ๆ จนกระทั่งโครงการนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณ อาจารย์ทุกท่านที่ได้อบรมสั่งสอนและถ่ายทอดวิชาความรู้ให้แก่พวกเรา จนสามารถนำความรู้มาประยุกต์ใช้ในการดำเนินโครงการได้

ขอบคุณ เพื่อนทุกคนที่คอยให้คำปรึกษา คอยรับฟังปัญหา และให้กำลังใจกันตลอดมา

สุดท้ายนี้ ขอขอบคุณ ทุกคนในครอบครัวที่คอยสนับสนุน คอยห่วงใยและเป็นกำลังใจให้กันตลอดเวลา ซึ่งเป็นแรงใจสำคัญที่ทำให้โครงการนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี จึงขอกราบขอบพระคุณ มา ณ ที่นี้

นายพีระภัทร์ น้ำฝุด

นายสัญญาชัย กิจศิริสิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป	VIII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มาของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ	4
1.3 ขอบเขตของโครงการ	4
1.4 วิธีการดำเนินการ.....	5
1.5 ประโยชน์ที่คาดว่าจะได้รับ	5
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	6
2.1 หลักการของเทคโนโลยีเสมือนจริง (Augmented Reality).....	6
2.1.1 การวิเคราะห์ภาพ (Image Analysis).....	6
2.1.2 Pose Estimation.....	7
2.1.3 3D Rendering	10
2.2 ความรู้เบื้องต้นเกี่ยวกับการประมวลผลภาพดิจิทัล	10
2.2.1 ชนิดของไฟล์รูปภาพ	11
2.2.2 มาตรฐานของสี.....	13
2.2.2.1 ระบบสีอาร์จีบี (RGB).....	14
2.2.2.2 ระบบสีเอชเอสวี (HSV)	15
2.2.3 การแปลงภาพให้เป็นภาพเชิงดิจิทัล.....	16
2.2.3.1 การบันทึกภาพ (Image Acquisition).....	16

เอกสารนี้เป็นเอกสารที่สงวนไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ หากมีข้อสงสัยโปรดแจ้งผู้ดูแลระบบหรือแจ้งไปยังเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.2.3.2 การสุ่มเลือกจุดตำแหน่ง (Image Sampling)	16
2.2.3.3 จุด (Pixel).....	16
2.2.3.4 การประมาณค่าความเข้มแสง	18
2.2.4 การขีดแบ่ง (Threshold).....	19
2.2.5 การหาขอบภาพ (Edge Detection).....	19
2.2.6 Contour	21
2.2.7 การตรวจจับการเคลื่อนไหว (Motion Detection).....	21
2.2.8 วิธีการเข้าคู่รูปแบบ (Template Matching)	23
2.3 FLARManager	24
2.3.1 หลักการพื้นฐานของ FLARToolKit	25
2.3.2 หลักการพัฒนา.....	27
2.4 ActionScript 3.0	28
บทที่ 3 การออกแบบและพัฒนา.....	32
3.1 หลักการทำงานของเทคโนโลยีเสมือนจริง (Augmented Reality).....	32
3.2 Data Flow Diagram	33
3.2.1 Context Diagram	33
3.2.2 Diagram Level-0 หรือ Parent Diagram.....	34
3.2.3 Diagram Level 1	35
3.3 โปรแกรมและชุดเครื่องมือที่ใช้ในการพัฒนา	36
3.3.1 ซอฟต์แวร์.....	36
3.3.2 ฮาร์ดแวร์.....	36
บทที่ 4 การพัฒนาโปรแกรม.....	37
4.1 การสร้าง Marker และการนำไปใช้งาน.....	37
4.1.1 คำนิยามของ Marker	37

สารบัญ (ต่อ)

	หน้า
4.1.2 คำนียามของ Pattern	38
4.1.3 การสร้าง Marker ใหม่ (ทำตามค่าเริ่มต้น คือ ความหนาของขอบเป็น 25% ของ ความยาวของขอบของ Marker).....	38
4.2 การใช้งานโปรแกรม ARToolKit Marker Generator.....	39
4.3 การต่อวิดีโอด้วยเทคนิค Blue/Green Screen.....	41
4.3.1 การเตรียมวิดีโอสำหรับใช้ในการตัดต่อ	41
4.3.2 การตัดพื้นหลังหรือตัดฉากออกจากวิดีโอด้วยโปรแกรม Motion 5	42
4.4 การเขียนโปรแกรม	46
4.4.1 การสร้างงานใหม่และการนำไลบรารี FLARToolKit มาใช้งาน	46
4.4.2 การตั้งค่าสำหรับการติดตามและวิเคราะห์ Marker และการตั้งค่าเพื่อแสดง ผลลัพธ์บนจอภาพ	48
4.4.3 Basic Augmented Reality.....	49
4.5 ตัวอย่างแอปพลิเคชันชนิด Augmented Reality ที่ทำการพัฒนา.....	52
บทที่ 5 บทสรุปและข้อเสนอแนะ.....	54
5.1 บทสรุป.....	54
5.2 ปัญหาและอุปสรรค.....	54
5.3 แนวทางการแก้ไข.....	55
5.4 แนวทางในการพัฒนาต่อ.....	55
บรรณานุกรม หรือ เอกสารอ้างอิง	56
ภาคผนวก.....	57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 จำนวนสีที่ปรากฏต่อจำนวนบิตของรูปภาพ.....	13
2.2 แสดงระยะห่างสูงสุดที่สามารถทำการ Tracking ตามขนาดของ Marker ที่แตกต่างกัน.....	27



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่

หน้า

1.1 การใช้แอปพลิเคชัน Layar ในการแสดงรายละเอียดข้อมูลต่าง ๆ ของสถานที่	2
1.2 การโฆษณาารถเบนซ์ด้วยการแสดงโมเดลรถแบบ 3 มิติ	3
1.3 การสวมแว่นตาเพื่อเรียนรู้ขั้นตอนในการประกอบหรือซ่อมรถ	3
1.4 แสดงการประยุกต์ใช้เทคโนโลยีเสมือนจริงจำลองการผ่าตัด	4
2.1 แสดงความสัมพันธ์ Camera Coordinated Frame และ Marker Camera Coordinated Frame	8
2.2 แสดงความสัมพันธ์ระหว่าง Ideal Screen Coordinates และ Observe Screen Coordinates	9
2.3 กระบวนการคำนวณค่า 3D Poses	9
2.4 การแปลงรูปภาพจากสัญญาณอนาล็อกให้เป็นสัญญาณดิจิทัล	11
2.5 ภาพแบบ Binary หรือภาพขาว-ดำ	11
2.6 ภาพแบบ Grayscale หรือภาพระดับสีเทา	12
2.7 ภาพแบบ RGB	12
2.8 ภาพแบบ Indexed	13
2.9 ข้อมูลภาพในหนึ่งจุดภาพ	14
2.10 ภาพวงจรสีของแสงแบบแม่สีหลักและแม่สีรอง	14
2.11 แสดงระบบสีแบบ HSV	15
2.12 ค่าของสีแยกออกเป็นค่าของแต่ละพิกเซล	17
2.13 ขนาดของจุด (Pixel) ที่ไม่เท่ากัน	17
2.14 ค่าของสีจากภาพและช่วงของสี 0-255	18
2.15 ค่าของสีในรูปแบบของค่าดิจิทัล	18
2.16 ช่วงของค่าสีในแต่ละพิกเซลของภาพ	19
2.17 กราฟแสดงการหาขอบด้วยวิธี Gradient Method (รูปที่2) และ Laplacian Method (รูปที่ 3) โดยที่รูปที่ 1 แสดงถึงความแตกต่างของระดับความเข้มของสี (GIMP 2004)	20
2.18 ตัวอย่างการหา Contour	21
2.19 ภาพตัวอย่างการใช้ Template Matching ในการค้นหาภาพ	24

สารบัญรูป (ต่อ)

รูปที่

หน้า

2.20 แสดงขั้นตอนการทำงานของ FLARToolKit.....	26
2.21 โครงร่างของโปรแกรมประยุกต์ทางด้าน AR ที่ใช้ FLARToolKit.....	28
3.1 แผนภาพการทำงานของเทคโนโลยีเสมือนจริง	32
3.2 Context Diagram.....	33
3.3 Diagram Level-0 หรือ Parent Diagram.....	34
3.4 Diagram Level-1.....	35
4.1 ตัวอย่างของ Marker ที่ FLARToolKit ได้สร้างไว้ให้แล้ว.....	37
4.2 ไฟล์ถูกแบ่งออกเป็น 4 x 4 ช่องใหญ่ หรือขนาด 16 x 16 ช่องเล็ก.....	38
4.3 แสดงการใส่สีดำเพื่อสร้างเป็นขอบของ Marker.....	39
4.4 ตัวอย่างของ Marker ที่พร้อมจะนำไปใช้งาน.....	39
4.5 โปรแกรม ARToolKit Marker Generator หลังจากทำการอัปโหลดไฟล์ Marker.....	40
4.6 โปรแกรม ARToolKit Marker Generator แสดงไฟล์ Marker ที่อัปโหลด และ Preview Maker.....	41
4.7 การติดตั้งฉากสีเขียวสำหรับใช้ในการถ่ายวิดีโอ	42
4.8 การสร้างงานใหม่ใน Motion 5.....	42
4.9 การ import ไฟล์วิดีโอมาใช้งาน	43
4.10 การตัดพื้นหลังสีเขียวออกด้วยคำสั่ง Keying.....	43
4.11 วิดีโอที่ตัดพื้นหลังออกเรียบร้อยแล้ว	44
4.12 การ Save วิดีโอเพื่อนำไปใช้งาน.....	44
4.13 การแปลงนามสกุลไฟล์วิดีโอจาก .MOV เป็น .FLV.....	45
4.14 แสดงการสร้างงานใหม่ (New ActionScript Project)	46
4.15 การตั้งชื่องานใหม่และการเลือกรูปแบบการรันแอปพลิเคชัน	47
4.16 ไฟล์และโฟลเดอร์ที่ต้องทำการคัดลอกมาใส่ในโฟลเดอร์ src ของงานใหม่ที่สร้างขึ้น	48
4.17 ตัวอย่างแอปพลิเคชันที่ Render วัตถุเสมือนเป็นรูปภาพ	52
4.18 ตัวอย่างแอปพลิเคชันที่ Render วัตถุเสมือนเป็นวิดีโอ.....	53
4.19 ตัวอย่างแอปพลิเคชันที่ Render วัตถุเสมือนเป็นอนิเมชัน.....	53

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของโครงการ

เทคโนโลยีความจริงเสมือน (Virtual Reality : VR) คือการจำลองแบบของสภาพแวดล้อมจริงโดยใช้คอมพิวเตอร์ ซึ่งสภาพแวดล้อมในความเป็นจริงเสมือนนี้ส่วนมากจะเป็นเรื่องเกี่ยวกับสิ่งที่เรามองเห็นได้ แสดงผลบนจอคอมพิวเตอร์หรืออุปกรณ์แสดงผลสามมิติ และยังรวมถึงการจำลองข้อมูลสารสนเทศที่เกี่ยวกับประสาทสัมผัสด้วย เช่น เสียงจากลำโพงหรือหูฟัง นอกจากนี้แล้วยังรวมถึงการจำลองการสัมผัส เช่น การตอบสนองต่อแรงและป้อนกลับ โดยผู้ใช้สามารถโต้ตอบกับสิ่งแวดล้อมเสมือนจริงนี้ได้ ทั้งการใช้อุปกรณ์ เช่น แป้นพิมพ์หรือเมาส์ หรือใช้อุปกรณ์อื่น เช่น ถุงมือคอนโทรล แขนควบคุม หรือคั่นบังคับหลายทิศทาง เป็นต้น ถือเป็นเทคโนโลยีเชิงโต้ตอบที่ผลักดันให้ผู้ใช้เกิดความรู้สึกของการเข้าร่วมอยู่ภายในสิ่งแวดล้อมที่ไม่ได้มีอยู่จริง เป็นเพียงสภาพแวดล้อมจำลองที่ถูกสร้างขึ้นโดยคอมพิวเตอร์ เทคโนโลยีการจำลองสภาพแวดล้อมเสมือนจริงนี้ถูกนำมาใช้ในหลายวงการ เช่น การจำลองการฝึกนักบินหรือหน่วยรบ ระบบฝึกผ่าตัดสำหรับนักเรียนแพทย์ และการศัลยกรรมเสมือนจริง ซึ่งสถานที่ของศัลยแพทย์และผู้ป่วยอาจจะอยู่คนละซีกโลก โดยอาศัยอุปกรณ์ควบคุมการใช้งานระยะไกล เป็นต้น และอีกหนึ่งตัวอย่างที่เราคุ้นกันดี เช่น ภาพยนตร์ 3D และเกมคอมพิวเตอร์ ที่นับวันยิ่งมีความสมจริงมากขึ้นทุกวัน

ประเภทของระบบความจริงเสมือนสามารถแบ่งตามพื้นฐานวิธีที่ติดต่อกับผู้ใช้นี้

- 1) Desktop VR หรือ Window on World Systems (WoW) เป็นระบบความจริงเสมือนที่ใช้จอภาพของคอมพิวเตอร์ในการแสดงผล
- 2) Video Mapping เป็นการนำวิดีโอมาเป็นอุปกรณ์หรือเครื่องมือนำเข้าข้อมูลของผู้ใช้และใช้กราฟิกคอมพิวเตอร์นำเสนอการแสดงผลในโมเดลแบบสองมิติหรือสามมิติ โดยผู้ใช้จะเห็นตัวเอง และเปลี่ยนแปลงตัวเองจากจอภาพ
- 3) Immersive Systems เป็นระบบความจริงเสมือนสำหรับผู้ใช้ส่วนบุคคล โดยผู้ใช้จะใช้อุปกรณ์ประเภทจอภาพสวมศีรษะ (HMD) ได้แก่ หมวกเหล็กหรือหน้ากากมาใช้จำลองภาพและการได้ยิน
- 4) Telepresence เป็นระบบการนำเสนอแบบเสมือนจริงด้วยเทคนิคการสร้างภาพวัตถุต่าง ๆ จากระยะไกลด้วยเทคโนโลยีมัลติมีเดีย ทั้งด้านภาพ เสียงและการสัมผัส ซึ่งมีทั้งระบบ Realtime และระบบ VDO

- 5) Augmented / Mixed Reality Systems เป็นการผสมผสานระหว่าง Telepresence ระบบความจริงเสมือน และเทคโนโลยีภาพเพื่อสร้างสิ่งที่เสมือนจริงให้กับผู้ใช้

เทคโนโลยีเสมือนจริง (Augmented Reality) เป็นประเภทหนึ่งของเทคโนโลยีความจริงเสมือน (Virtual Reality) ที่มีการนำระบบความจริงเสมือนมาผนวกกับเทคโนโลยีภาพ เพื่อสร้างสิ่งเสมือนจริงให้กับผู้ใช้ จัดเป็นแขนงหนึ่งของงานวิจัยด้านวิทยาการคอมพิวเตอร์ว่าด้วยการเพิ่มภาพเสมือนของโมเดลสามมิติที่สร้างจากคอมพิวเตอร์ลงไปในภาพที่ถ่ายมาจากกล้องวิดีโอ เว็บแคม หรือกล้องในโทรศัพท์มือถือแบบเฟรมต่อเฟรมด้วยเทคนิคทางด้านคอมพิวเตอร์กราฟิก หลักการของเทคโนโลยีเสมือนจริงนั้นเรียบง่าย แต่สามารถนำมาพัฒนาและประยุกต์ใช้งานในด้านต่าง ๆ ได้มากมาย ดังเช่นตัวอย่างต่อไปนี้

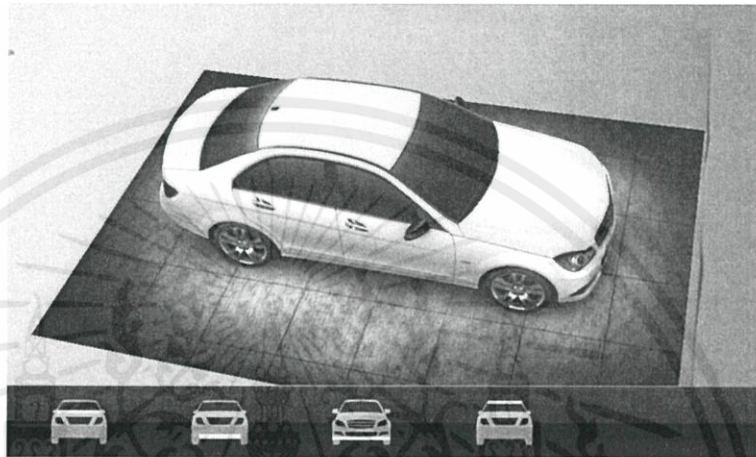
- 1) การประยุกต์ใช้ทางด้านระบบนำทาง เช่น ในโปรแกรม Layar สามารถซ้อนภาพสถานที่ปลายทางหรือจุดที่น่าสนใจลงในภาพจริง เพื่อให้ผู้ใช้โทรศัพท์มือถือเคลื่อนที่ตามทิศทางที่ระบุไปยังตำแหน่งนั้น ๆ ได้ นอกจากนั้นยังสามารถเลือกแสดงชั้นข้อมูลจาก Google Maps ซึ่งมีข้อมูลสถานที่ในประเทศไทยจำนวนมากพอสมควร สามารถใช้ค้นหาร้านอาหาร ศาลา ภัตตาคาร ในกรุงเทพฯ ได้ด้วยความแม่นยำระดับหนึ่ง



รูปที่ 1.1 การใช้แอปพลิเคชัน Layar ในการแสดงรายละเอียดข้อมูลต่าง ๆ

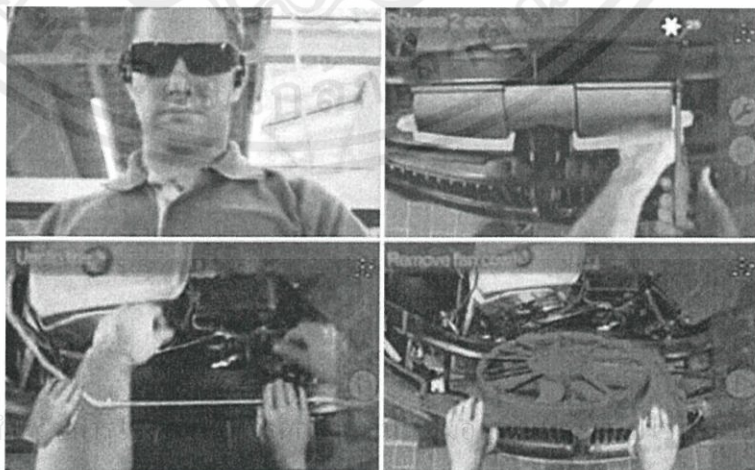
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าของสถานที่
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) การประยุกต์ใช้ทางด้านการโฆษณา เช่น บริษัทยานยนต์ Mercedes-Benz ได้นำเอาเทคโนโลยีเสมือนจริงมาช่วยในการประชาสัมพันธ์และทำแคมเปญการขายให้มีลูกเล่นและความน่าสนใจเพิ่มขึ้น ซึ่งลูกค้าสามารถเห็นแบบจำลองของรถเบนซ์ในรูปแบบของโมเดล 3 มิติ



รูปที่ 1.2 การโฆษณารถเบนซ์ด้วยการแสดงโมเดลรถแบบ 3 มิติ

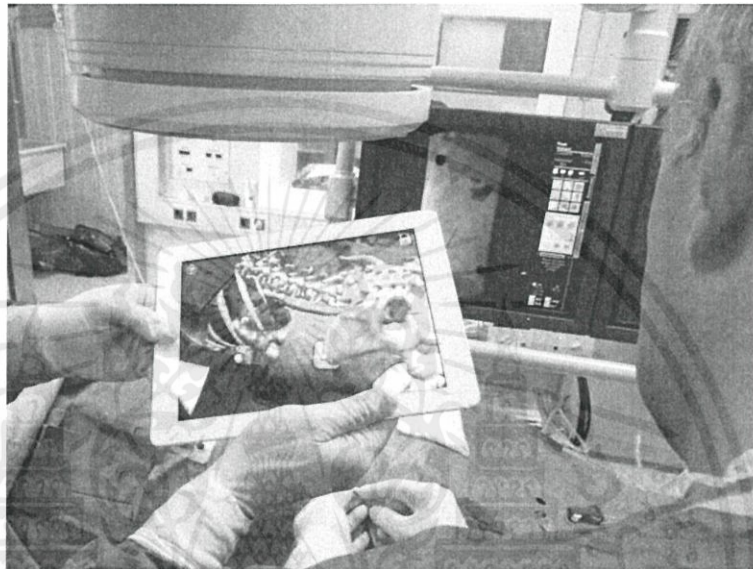
- 3) การประยุกต์ใช้ในอุตสาหกรรม เช่น อุตสาหกรรมสร้างเครื่องบิน อุตสาหกรรมการผลิตรถยนต์ โดยบริษัท BMW ได้ใช้เทคโนโลยีเสมือนจริงมาช่วยในขั้นตอนการผลิต โดยให้ผู้ใช้เรียนรู้การทำงาน ด้วยการสวมแว่นตาที่จะมีคำแนะนำและมีการจำลองการทำงาน แสดงให้เห็นแต่ละขั้นตอนก่อนปฏิบัติจริงในรูปแบบ 3 มิติ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ห้ามเผยแพร่หรือแจกจ่ายโดยไม่ได้รับอนุญาตจากฝ่ายที่เกี่ยวข้อง

รูปที่ 1.3 การสวมแว่นตาเพื่อเรียนรู้ขั้นตอนในการประกอบหรือซ่อมรถ

- 4) การประยุกต์ใช้ทางด้านการแพทย์ เช่น การเรียบเรียงหลักการประยุกต์ใช้ภาพเสมือนจริงทางการแพทย์ โดยเพิ่มตัวต่อประสานระบบสัมผัสภาพ 3 มิติเพื่อเพิ่มความสมจริงในการรักษา และให้นักศึกษาแพทย์ได้ใช้เครื่องมือแพทย์รักษาหรือผ่าตัดผู้ป่วยแบบไม่ต้องสัมผัสกับผู้ป่วยจริง



รูปที่ 1.4 แสดงการประยุกต์ใช้เทคโนโลยีเสมือนจริงจำลองการผ่าตัด

1.2 วัตถุประสงค์ของโครงการ

เพื่อศึกษาหลักการและทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับเทคโนโลยีเสมือนจริง เช่น การวิเคราะห์ภาพ (Image Analysis) การคำนวณค่าตำแหน่งเชิงสามมิติ (Pose Estimation) ของ Marker เทียบกับกล้อง และกระบวนการสร้างภาพสองมิติจากโมเดลสามมิติ (3D Rendering) เป็นต้น เพื่อนำไปใช้ในการพัฒนาแอปพลิเคชันชนิด Augmented Reality ประเภทต่าง ๆ ตามวัตถุประสงค์ที่ต้องการ

1.3 ขอบเขตของโครงการ

การพัฒนาแอปพลิเคชันชนิด Augmented Reality ด้วยภาษา ActionScript 3.0 โดยใช้ชุดเครื่องมือพัฒนาโปรแกรม (Software Development Kit) ที่มีชื่อว่า Adobe Flash Builder 4.6 และใช้ Library ที่ช่วยในการพัฒนาแอปพลิเคชัน คือ FLARToolKit ซึ่งกระบวนการภายในของเทคโนโลยีเสมือนจริง ประกอบด้วย 3 กระบวนการ ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1) การวิเคราะห์ภาพ (Image Analysis) เป็นขั้นตอนการค้นหา Marker จากภาพที่ได้จากกล้อง แล้วสืบค้นจากฐานข้อมูล (Marker Database) ที่มีการเก็บข้อมูลขนาดและรูปแบบของ Marker ไว้ เพื่อนำมาวิเคราะห์รูปแบบของ Marker
- 2) การคำนวณค่าตำแหน่งเชิง 3 มิติ (Pose Estimation) ของ Marker เทียบกับกล้อง
- 3) กระบวนการสร้างภาพสองมิติจากโมเดลสามมิติ (3D Rendering) เป็นการเพิ่มข้อมูลเข้าไปในภาพโดยใช้ค่าตำแหน่งเชิง 3 มิติที่คำนวณได้ จนได้ภาพเสมือนจริง

1.4 วิธีการดำเนินการ

- 1) ศึกษาเกี่ยวกับเทคโนโลยีเสมือนจริง (Augmented Reality)
- 2) ศึกษาหลักการและทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับเทคโนโลยีเสมือนจริง
- 3) ศึกษาโครงสร้างและรูปแบบการโปรแกรมเขียนภาษา ActionScript 3.0
- 4) ศึกษาเกี่ยวกับชุดเครื่องมือพัฒนาโปรแกรมและ Library ที่เหมาะสม
- 5) ศึกษาการติดตั้งและใช้โปรแกรม Adobe Flash Builder 4.6
- 6) ศึกษาการติดตั้ง FLARToolKit Library
- 7) วิเคราะห์และออกแบบระบบ
- 8) เขียนโปรแกรมและพัฒนาแอปพลิเคชันตามขั้นตอนที่ออกแบบ
- 9) วิเคราะห์ผลของแอปพลิเคชันที่พัฒนาขึ้น และแก้ไขส่วนที่ผิดพลาดเพื่อให้แอปพลิเคชันสามารถทำงานได้อย่างมีประสิทธิภาพมากที่สุด
- 10) สรุปผลและข้อเสนอแนะ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้รับความรู้ความเข้าใจเกี่ยวกับหลักการและทฤษฎีต่าง ๆ ของเทคโนโลยีเสมือนจริง
- 2) ได้รับความรู้ความเข้าใจเกี่ยวกับกระบวนการประมวลผลภาพต่าง ๆ
- 3) ได้รับความรู้ความเข้าใจเกี่ยวกับโครงสร้างภาษาที่ใช้ในการเขียนโปรแกรม
- 4) ได้ฝึกฝนการเขียนโปรแกรมด้วยภาษา ActionScript 3.0
- 5) ได้รับความรู้ความเข้าใจเกี่ยวกับการสร้างโมเดล 3 มิติ
- 6) ได้รับความรู้ความเข้าใจเกี่ยวกับการสร้างวิดีโอด้วยเทคนิค Green Screen
- 7) ได้ฝึกบริหารและจัดการเวลาในการทำงาน
- 8) ได้ฝึกทำงานร่วมกันเป็นทีม

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

Simple Augmented Reality นั้นประกอบไปด้วยกล้องวิดีโอหรือเว็บแคม (Webcam) ระบบประมวลผล และหน้าจอแสดงผล กล้องวิดีโอจะทำการจับภาพและระบบจะทำการนำวัตถุเสมือนมาแสดงบนภาพและแสดงผลออกทางจอภาพ ที่กล่าวมานี้เป็นหลักการง่าย ๆ ของเทคโนโลยีเสมือนจริง (Augmented Reality)

กระบวนการทำงานของเทคโนโลยีเสมือนจริง (Augmented Reality) นั้นจะประกอบด้วยขั้นตอนที่สำคัญดังต่อไปนี้คือ Image Analysis, Pose Estimation และ 3D Rendering เทคโนโลยีเสมือนจริงสามารถแบ่งประเภทตามส่วนวิเคราะห์ภาพ (Image Analysis) ได้ออกเป็น 2 ประเภทด้วยกัน ได้แก่ Marker based AR และ Marker-less Based AR โดยที่ Marker based AR นั้นเป็นการวิเคราะห์ภาพโดยอาศัย Marker (วัตถุสัญลักษณ์) เป็นหลักในการทำงาน ส่วน Marker-less Based AR เป็นการวิเคราะห์ภาพที่ใช้คุณลักษณะต่าง ๆ ที่อยู่ในภาพ (Natural Features) มาทำการวิเคราะห์เพื่อคำนวณหาค่าตำแหน่งเชิง 3 มิติ (3D Pose) เพื่อนำไปใช้งานต่อไป ในปริญญาโทฉบับนี้จะเน้นศึกษาเฉพาะ Marker based AR ก่อนอื่นเราจะขออธิบายกว้าง ๆ ถึงขั้นตอนของ Marker based AR ซึ่งสามารถแบ่งได้เป็น 3 ขั้นตอน ได้แก่ Image Analysis, Pose Estimation และ 3D Graphic Rendering

โดยรวมแล้วกระบวนการ Image Analysis และ Pose Estimation จะถูกเรียกรวม ๆ กันว่า Visual Tracking เนื่องจากในโครงการนี้เราได้เลือกศึกษาเครื่องมือที่เป็นที่รู้จักกันอย่างกว้างขวางในแวดวงของเทคโนโลยีเสมือนจริง โดยที่เครื่องมือพื้นฐานที่กล่าวถึงก็คือ FLARToolKit ดังนั้นเนื้อหาในส่วนของการบวนการ Image Analysis และ Pose Estimation เราจึงขออธิบายโดย ยึดหลักของ FLARToolKit Library เป็นพื้นฐาน

2.1 หลักการของเทคโนโลยีเสมือนจริง (Augmented Reality)

2.1.1 การวิเคราะห์ภาพ (Image Analysis)

- 1) แปลงภาพที่ได้เป็นภาพไบนารี (Binary Image) โดยในแต่ละพิกเซลของภาพ จะมีความเข้มของแสงได้ 2 แบบ คือ สว่างกับมืด เขียนแทนได้ด้วยตัวเลข 0 และ

เอกสารนี้เป็นเอกสารที่สงวน 1 โดยจะมีค่าเป็น 0 เมื่อค่าระดับความสว่าง (Intensity) ของพิกเซลนั้นมีค่าต่ำกว่าค่า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้ง ถ้า Threshold และมีค่าเป็น 1 เมื่อค่าระดับความสว่างของพิกเซลนั้นมีค่ามาก

กว่า Threshold ซึ่งความเข้มแสงที่สองระดับนั้นจะทำให้การประมวลผลทำงานได้อย่างมีประสิทธิภาพ

- 2) ทำการหาพื้นที่ติดต่อกัน (Connected Components) โดยใช้เทคนิคในการวิเคราะห์ภาพที่เรียกว่า Connected Component Labeling
- 3) ทำการหาเส้นรอบรูป (Contours) ของพื้นที่ที่ได้มา (ผลลัพธ์ในขั้นตอนที่แล้ว)
- 4) จากผลลัพธ์ของขั้นตอนที่แล้ว ระบบจะทำการประมาณค่าพารามิเตอร์ของสมการเส้นตรงที่แทนเส้นรอบรูปซึ่งเป็นรูปสี่เหลี่ยม หลังจากนั้นระบบจะหาจุดมุม (Corners) ทั้งสี่จุดของ Marker จากจุดตัดของเส้นตรงทั้งสี่เส้นที่หาได้ และสี่จุดที่หาได้นั้นจะถูกนำไปใช้ในขั้นตอนต่อไป

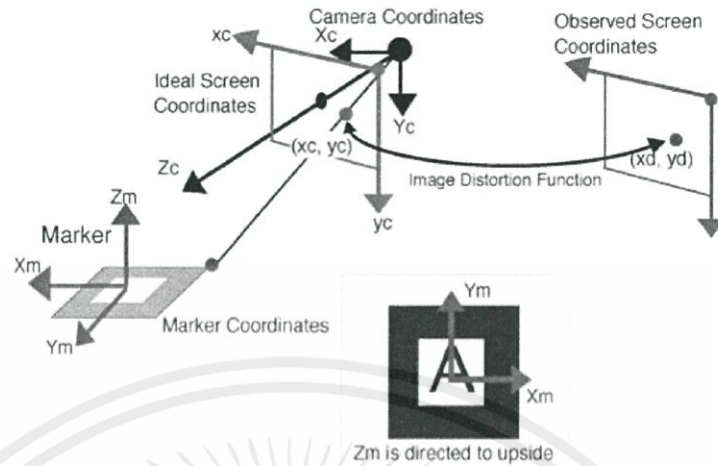
2.1.2 Pose Estimation

เป็นขั้นตอนของการคำนวณค่าตำแหน่งเชิง 3 มิติ (3D Pose) ของ Marker เมื่อเทียบกับกล้องวิดีโอ ค่านี้จะถูกแสดงในรูปเมตริกซ์ขนาด 4x4 (TCM) ที่ระบุความสัมพันธ์ระหว่าง Camera Coordinated Frame และ Marker Coordinated Frame ดังสมการที่ (1)

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

ซึ่ง Camera Coordinated Frame ก็คือ Coordinated Frame ที่ใช้อ้างอิงตำแหน่งใด ๆ ของกล้องวิดีโอ และ Marker Coordinated Frame ก็คือ Coordinated Frame ที่ใช้อ้างอิงตำแหน่งใด ๆ ของ Marker

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



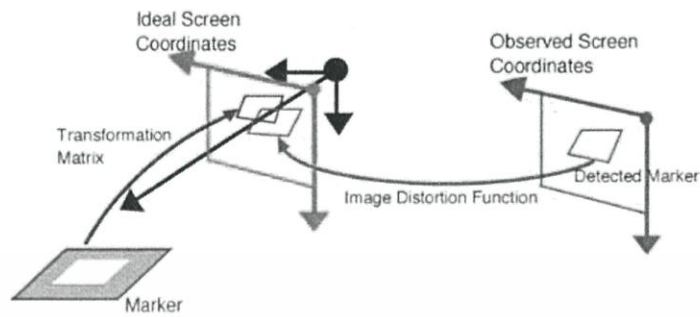
รูปที่ 2.1 แสดงความสัมพันธ์ Camera Coordinated Frame และ Marker Coordinated Frame

จากรูปที่ 2.1 ความสัมพันธ์ระหว่างจุดใด ๆ (X_c, Y_c, Z_c) บน Camera Coordinated Frame กับจุดที่ตรงกัน (x_i, y_i) ใน Ideal Screen Coordinated Frame เป็นไปตาม Perspective Projection ดังสมการที่ (2)

$$\begin{bmatrix} hx_i \\ hy_i \\ h \end{bmatrix} = \begin{bmatrix} sf_x & 0 & x_c & 0 \\ 0 & sf_y & y_c & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = C \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (2)$$

โดยที่ C เป็นเมตริกซ์ขนาด 3×4 ที่ประกอบไปด้วยค่า s, f_x, f_y, x_c, y_c โดยทั่วไปแล้วค่าเหล่านี้รวมกันเรียกว่า Camera Parameters ซึ่งจะคำนวณได้มาจากขั้นตอน Camera Calibration ส่วนค่าความสัมพันธ์ระหว่างจุดใด ๆ บน Ideal Screen Coordinated Frame (x_i, y_i) กับ Observe Screen Coordinated Frame (x_d, y_d) ซึ่งเป็นจุดที่เราเห็นจริง ๆ ในภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงความสัมพันธ์ระหว่าง Ideal Screen Coordinates และ Observe Screen Coordinates

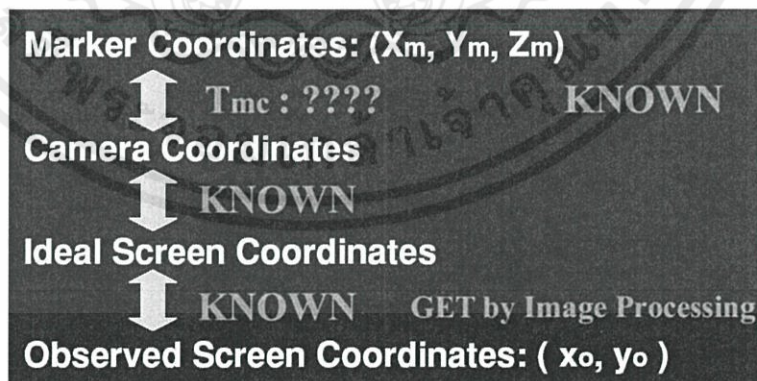
และสามารถอธิบายได้ดังสมการที่ (3)

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

$$p = \{1 - fd^2\}$$
(3)

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

โดยที่ x_0, y_0 คือจุดศูนย์กลางของการ Distortion (Center Coordinates of Distortion) และ f คือ Distortion Factor ซึ่งค่าทั้ง 2 จะได้มาจากกระบวนการ Camera Calibration



รูปที่ 2.3 กระบวนการคำนวณค่า 3D Poses

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.3 จะแสดงกระบวนการที่จะได้มาของค่า TCM เมื่อเรารู้ค่าตำแหน่งของ Marker ทั้ง 4 จุดบน Observed Screen Coordinates ในภาพที่ถ่ายจากกล้องวิดีโอ ซึ่งกล่าวโดยเฉพาะค่านี้ สามารถหาได้จากการคำนวณหาค่าตอบของฟังก์ชันค่าผิดพลาด (Error Function) ดังสมการที่ (4) โดยทั่วไปแล้วเราจะใช้เทคนิคทางด้าน การหาค่าที่เหมาะสม (Optimization) ซึ่งเป็นกระบวนการแบบ Iterative

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

โดยที่ \hat{x}_i \hat{y}_i แสดงได้โดย

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

2.1.3 3D Rendering

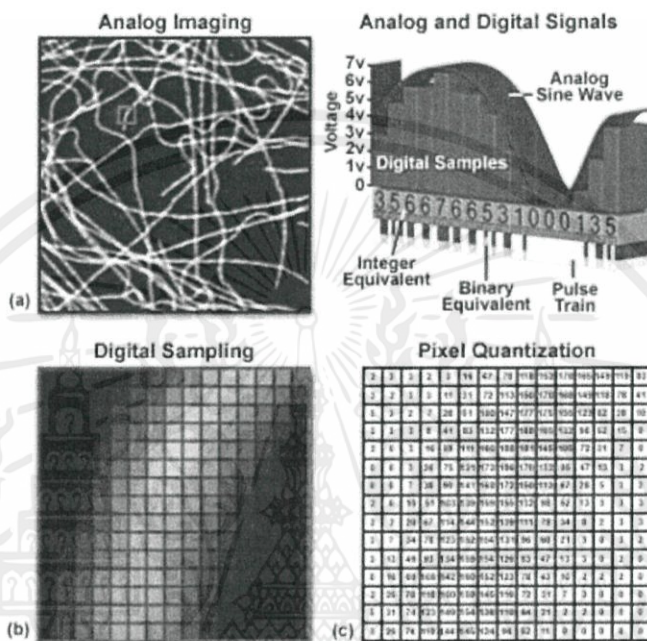
ส่วนนี้เป็นส่วนสุดท้ายที่จะทำให้กระบวนการ Augmented Reality ครบถ้วนสมบูรณ์ ซึ่งก็คือการเพิ่ม (Augment) ข้อมูลที่เราต้องการ โดยทั่วไปแล้วจะเป็นโมเดลสามมิติ (3D Model) ลงไปในภาพที่ได้จากกล้องวิดีโอ ณ ตำแหน่งของ Marker ที่ตรวจพบจากขั้นตอน Image Analysis โดยใช้ค่าตำแหน่งเชิงสามมิติที่คำนวณได้จากขั้นตอน Pose Estimation

โดยทั่วไปแล้ว 3D Rendering หมายถึง กระบวนการที่ทำการสร้างภาพสองมิติจากโมเดลสามมิติ ซึ่งโมเดลสามมิตินี้จะอธิบายวัตถุหรือสิ่งแวดล้อมหนึ่ง ๆ ที่เราต้องการสร้างภาพนั้น เนื่องจากเทคนิคทางด้าน 3D rendering นั้นมีหลากหลาย แต่ในโครงการนี้เราเลือกศึกษาเทคนิค 3D Rendering โดยใช้ Realtime 3D Engine คือ Papervision3D ช่วยในการพัฒนาแอปพลิเคชัน

2.2 ความรู้เบื้องต้นเกี่ยวกับการประมวลผลภาพดิจิทัล

การประมวลผลภาพดิจิทัลเกี่ยวข้องข้องกับการแปลงข้อมูลรูปภาพที่เป็นสัญญาณอะนาล็อก (Analogue Signal) ให้อยู่ในรูปของสัญญาณดิจิทัล (Digital Signal) เพื่อใช้ในการประมวลผลผ่านทางคอมพิวเตอร์ได้ และยังสามารถนำมาใช้ในการลดปัญหาของภาพ เช่น ลดสัญญาณรบกวนภายในภาพ เป็นต้น ในการแปลงภาพให้เป็นสัญญาณดิจิทัลนั้น ระบบจะนำรูปภาพที่รับเข้ามาไปคำนวณด้วย

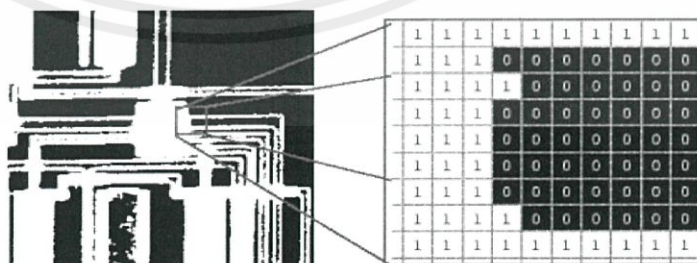
กระบวนการ Sampling และ Quantization และส่งข้อมูลออกมาในรูปแบบดิจิทัล จากนั้นคอมพิวเตอร์จะเก็บข้อมูลภาพลงหน่วยความจำ โดยการจองหน่วยความจำภายในเครื่องในรูปแบบของอาร์เรย์ ค่าในแต่ละช่องของอาร์เรย์แสดงถึงคุณสมบัติต่าง ๆ ของจุดภาพ (pixel) นั้น ๆ และตำแหน่งของช่องอาร์เรย์ก็เป็นตัวกำหนดตำแหน่งของจุดภาพภายในภาพนั้นด้วย



รูปที่ 2.4 การแปลงรูปภาพจากสัญญาณอนาล็อกให้เป็นสัญญาณดิจิทัล

2.2.1 ชนิดของไฟล์รูปภาพ

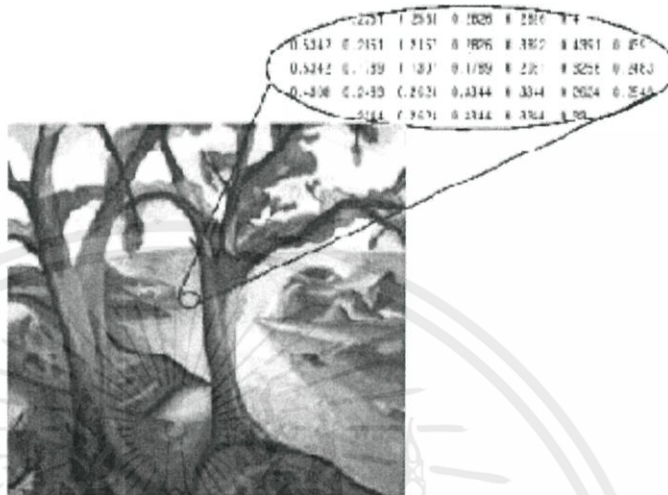
- 1) Binary Image หรือ ภาพขาว-ดำ เป็นรูปที่ใช้เนื้อที่เพียง 1 บิตต่อจุดภาพ โดยค่าสีจะมีแค่สองค่า คือค่า 0 หรือสีดำ และค่า 1 หรือสีขาว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

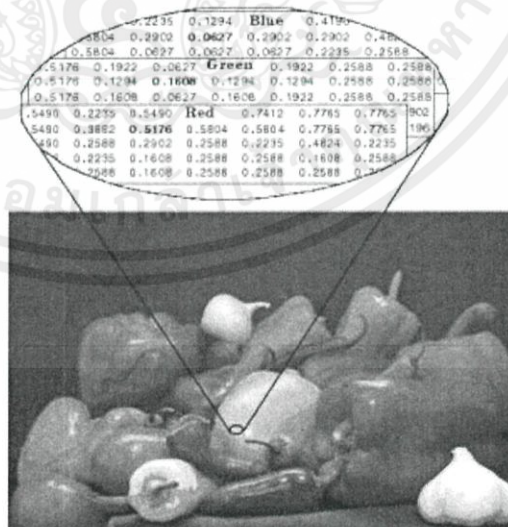
รูปที่ 2.5 ภาพแบบ Binary หรือ ภาพขาว-ดำ

- 2) Grayscale Image เป็นรูปที่เก็บโดยใช้รูปแบบของอาร์เรย์ 2 มิติ โดยค่าที่เก็บจะมีค่าอยู่ในช่วง ๆ หนึ่ง ซึ่งระดับของสีขึ้นอยู่กับขนาดของบิตที่ใช้เก็บค่าสี



รูปที่ 2.6 ภาพแบบ Grayscale หรือภาพระดับเทา

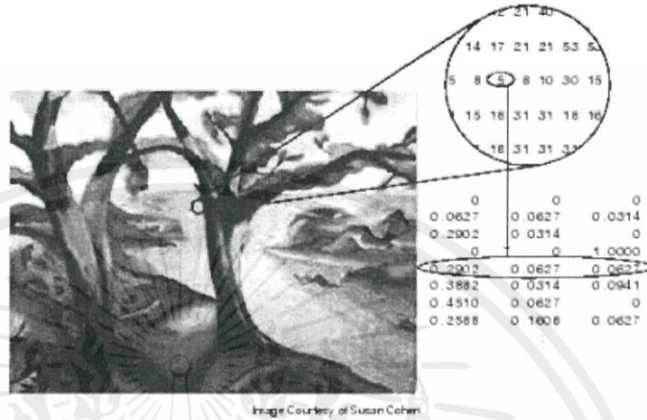
- 3) RGB Image หรือ True Color Image เป็นรูปที่เก็บโดยใช้อาร์เรย์ 3 มิติขนาด $m \times n \times g$ โดย m คือความยาว และ n คือความกว้างของภาพในหน่วยจุดภาพ ส่วน g นั้นเป็นจำนวนเต็มที่แทนจำนวนบิตของข้อมูลในแต่ละจุดภาพ ในแต่ละมิติ จะเก็บค่าสีแยกกันคือ สีแดง (Red) สีเขียว (Green) และสีน้ำเงิน (Blue)



รูปที่ 2.7 ภาพแบบ RGB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลเห็นว่ามีประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และข้อมูลเชิงพาณิชย์ในเอกสารทุกครั้งที่มีการนำไปใช้

- 4) Indexed Image เป็นรูปที่มีรูปแบบการเก็บแบบ Indexed คือ ภาพประเภทนี้จะเก็บค่าสีเป็น Indexed และในแต่ละช่องอาร์เรย์ จะเก็บตำแหน่งของสีใน Indexed นั้น ๆ ไว้



รูปที่ 2.8 ภาพแบบ Indexed

2.2.2 มาตรฐานของสี

จำนวนสีสูงสุดที่เป็นไปได้ของแต่ละจุดภาพจะขึ้นอยู่กับจำนวนบิตที่ใช้ โดยจำนวนบิตที่มากขึ้นจะให้จำนวนสีมากขึ้นไปด้วย ดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 จำนวนสีปรากฏต่อจำนวนบิตของรูปภาพ

จำนวนบิต	เลขฐานสอง	จำนวนสี
1	2^1	2
2	2^2	4
4	2^4	16
8	2^8	256
16	2^{16}	65536

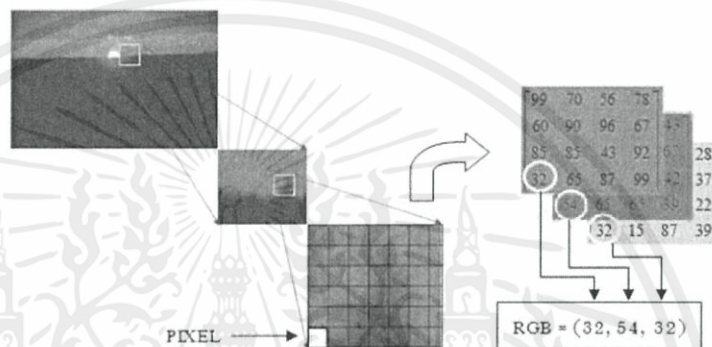
นอกจากนี้มาตรฐานของสีมีหลายแบบแต่ที่นิยมใช้กัน ได้แก่ ระบบอาร์จีบี (Red

Green Blue, RGB) และระบบ เอชเอสวี (Hue Saturation Value, HSV)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

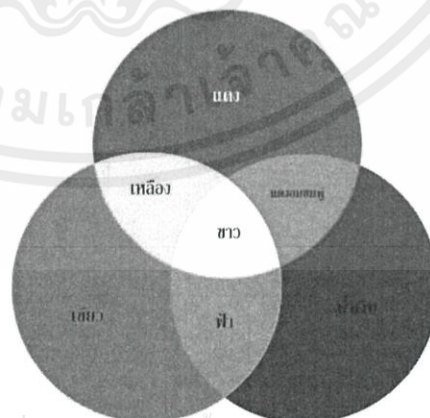
2.2.2.1 ระบบสีอาร์จีบี (RGB)

เป็นระบบของสีที่เกิดจากการรวมกันของแม่สี คือ สีแดง (R) สีเขียว (G) และสีน้ำเงิน (B) เข้าด้วยกันแบบเพิ่มเข้าไป อาทิเช่นรูปภาพ 24 บิต จะมีค่าสีแดงเท่ากับ 8 บิต (256 ค่า) สีเขียวมีค่าเท่ากับ 8 บิต (256 ค่า) และสีน้ำเงินมีค่าเท่ากับ 8 บิต (256 ค่า) โดยในแต่ละจุดของภาพจะเกิดจากการรวมสีต่าง ๆ กัน แสดงดังรูปที่ 2.9



รูปที่ 2.9 ข้อมูลภาพในหนึ่งจุดภาพ

เมื่อนำแม่สีของแสงทั้ง 3 มาผสมกัน ในปริมาณแสงสว่างเท่ากันก็จะได้เป็นแสงที่สีขาว แต่ถ้าผสมกันระหว่างแสงระดับความสว่างต่างกัน ก็จะได้ผลที่เป็นแสงสี ๆ มากมายเป็นล้านสี ส่วนใหญ่การใช้สีลักษณะนี้จะใช้ในอุปกรณ์ที่เกี่ยวกับแสง เช่น จอภาพ กล้องดิจิทัล สแกนเนอร์ เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกไปจำหน่ายและข้อมูลเชิงลึกเป็นของระบบขององค์กรนำไปใช้

รูปที่ 2.10 ภาพวงจรสีของแสงแบบแม่สีหลักและแม่สีรอง

2.2.2.2 ระบบเอชเอสวี (HSV)

เป็นระบบที่จะใช้การพิจารณาสีด้วย Hue (H) Saturation (S) และ Value (V) โดย H คือค่าของแม่สีหลัก (แดง เขียว และน้ำเงิน) ในทางปฏิบัติจะอยู่ที่ 0-255 รวมเป็น 256 ค่า ถ้าค่า H มีค่าเท่ากับศูนย์จะแสดงสีแดงและเมื่อค่าเพิ่มขึ้นจะแสดงค่าสีต่าง ๆ กันเปลี่ยนไปตามสเปกตรัม (Spectrum) ของสีจนค่าที่ 255 ก็จะมากลับมาเป็นสีแดงอีกครั้ง หรืออาจกล่าวได้ว่าค่า $H \in [0^\circ, 360^\circ]$ ซึ่งค่า H สามารถคำนวณจากระบบสี RGB ได้ดังนี้

$$R_H = R - \min(R, G, B)$$

$$G_H = G - \min(R, G, B)$$

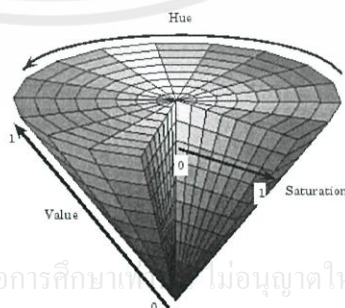
$$B_H = B - \min(R, G, B)$$

ค่า S คือค่าความบริสุทธิ์ของสี ซึ่งถ้าค่า S มีค่าเป็นศูนย์ ค่าสีที่ได้จะเป็นสีเทา (ไม่มีค่า H) หากค่า S เท่ากับ 255 จะได้ค่าสีเท่ากับ H อาจกล่าวได้ว่าค่า $S_{HSV} \in [0, 1]$ โดยสามารถคำนวณได้จาก

$$S = \frac{\max(R, G, B) - \min(R, G, B)}{\max(R, G, B)}$$

ค่า V คือค่าความสว่างของสี ณ จุด (Pixel) นั้น ๆ ซึ่งสามารถวัดโดยวัดค่าความเข้มของแสงสว่างของสีที่ประกอบกัน โดยค่า $V \in [0, 1]$ โดยสามารถคำนวณได้จาก

$$V = \max(R, G, B)$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.11 แสดงระบบสีแบบ HSV

2.2.3 การแปลงภาพให้เป็นภาพเชิงดิจิทัล

ภาพดิจิทัลนั้นมีลักษณะเป็นฟังก์ชัน 2 มิติ หรือ $f(x,y)$ ของค่าความเข้มของแสง โดยที่ x และ y คือ ค่าที่บอกถึงตำแหน่งในระบบพิกัดฉาก และค่าของฟังก์ชัน ณ ตำแหน่งใด ๆ จะเป็นสัดส่วนกับความสว่างของแสง ณ ตำแหน่งนั้น กระบวนการแปลงภาพให้เป็นภาพในเชิงดิจิทัลเรา เรียกว่า Image Digitization มีกระบวนการ 3 ขั้นตอน คือ การบันทึกภาพ (Image Acquisition) การสุ่มเลือกจุดตำแหน่ง (Image Sampling) และการประมาณค่าความเข้มของแสง (Image Quantization)

2.2.3.1 การบันทึกภาพ (Image Acquisition)

การถ่ายภาพเป็นการแปลงภาพเชิงต่อเนื่อง (Continuous Image) จากภาพแบบ 3 มิติที่มีองค์ประกอบทั้งความกว้าง ความสูง และความลึก ให้เป็นภาพเชิงต่อเนื่อง 2 มิติที่มีแต่ความกว้างและความสูงเท่านั้น โดยใช้อุปกรณ์เชิงแสง (Optical Device) เช่น กล้องถ่ายรูป เพื่อแปลงภาพให้มาเป็นภาพบนฟิล์ม รูปถ่ายบนกระดาษ หรือภาพบนจอคอมพิวเตอร์ โดยในส่วนของกระบวนการบันทึกภาพจะใช้กลไกทางแสงในอุปกรณ์การถ่ายภาพทำหน้าที่รับภาพเข้ามา

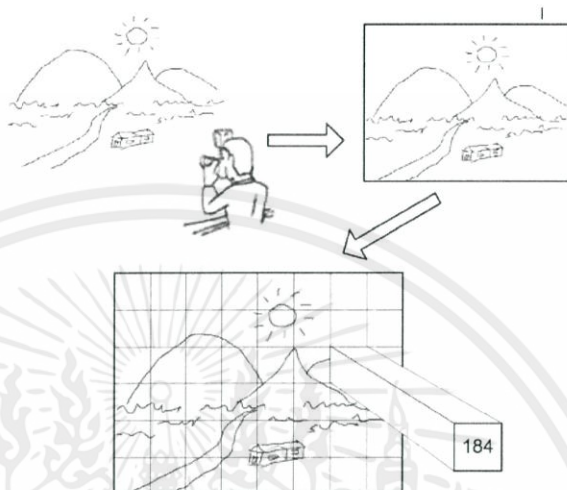
2.2.3.2 การสุ่มเลือกจุดตำแหน่ง (Image Sampling)

การทำภาพปกติให้กลายเป็นภาพดิจิทัลจะใช้วิธีการสุ่มเลือกจุดตำแหน่ง Spatially Sampling โดยสุ่มเลือกเฉพาะบางตำแหน่งในภาพดังแสดงในรูปที่ 2.12 ถ้าสุ่มเลือกมา รายละเอียดภาพที่ได้ก็จะมีรายละเอียดสูง โดยมีหน่วยของการสุ่มเลือกก็คือ จุดภาพ หรือ Pixel

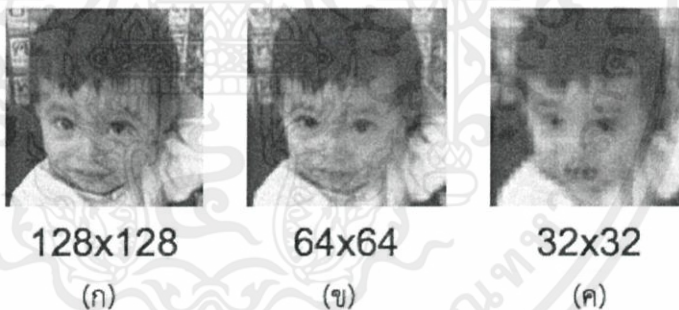
2.2.3.3 จุด (Pixel)

เป็นหน่วยที่แสดงความละเอียดของรูปภาพในการแสดงผลออกทางหน้าจอคอมพิวเตอร์ แต่รูปที่มองเห็นจริงด้วยตาไม่ใช่การเรียงกันของจุด แต่เป็นภาพเชิงต่อเนื่องคือไม่สามารถแยกลงไปเป็นที่ละจุด ๆ ได้ ดังนั้นเมื่อภาพถูกแสดงทางหน้าจอคอมพิวเตอร์ ภาพจะต้องถูกปรับให้อยู่ในหน้าจอที่ประกอบด้วยจุดสี จึงจำเป็นที่

จะต้องมีการสุ่มเลือกจุดตำแหน่ง ถ้ามีการสุ่มเลือกจุดตำแหน่งที่สูง คุณภาพของภาพที่ได้ก็จะดีขึ้น แสดงดังในรูปที่ 2.13



รูปที่ 2.12 ค่าของสีแยกออกเป็นค่าของแต่ละพิกเซล



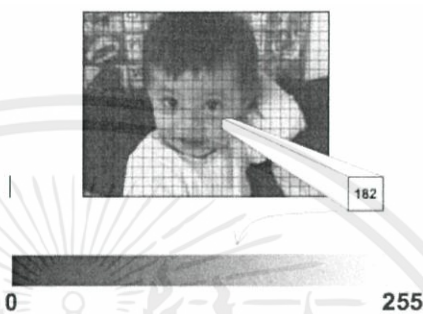
รูปที่ 2.13 ขนาดของจุด (Pixel) ที่ไม่เท่ากัน

จะเห็นได้ว่าภาพเดียวกัน แต่เมื่อทำการสุ่มเลือกจุดมาละเอียดไม่เท่ากัน เมื่อทำการขยายภาพจะให้คุณภาพของภาพแตกต่างกัน โดยจากรูป รูปที่ 2.13 (ก) มีความละเอียดสูงที่สุด ภาพที่มีความละเอียดสูงจะใช้นี้ในการเก็บข้อมูลขนาดของภาพสูงขึ้นตามไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.4 การประมาณค่าความเข้มแสง

ภาพที่ได้จากการสุ่มเลือก แต่ละจุดในภาพจะถูกแทนด้วยสี ซึ่งภาพในโทนสีเทา (Grayscale) ประกอบไปด้วยสีดำและไล่เฉดสีจางลงไปจนถึงสีขาว ดังแสดงในรูปที่ 2.14



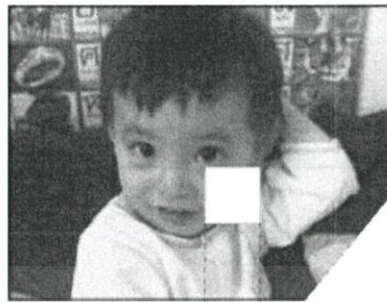
รูปที่ 2.14 ค่าของสีจากภาพและช่วงของสี 0-255

จากรูป สีดำจะแทนด้วยค่าตัวเลข 0 ในขณะที่สีขาวจะแทนด้วยค่าตัวเลข 255 รวมทั้งสิ้นเป็น 256 ระดับสี (0-255) โดยมีจำนวนบิตซึ่งเป็นหน่วยความจำที่ใช้ในการเก็บค่านี้นี่หนึ่งค่า จากรูปสามารถเขียนอยู่ในรูปเลขฐานสอง หรือ 2^8 โดยที่ 8 คือจำนวนบิต ดังนั้นสีดำในภาพจะถูกแทนด้วยรหัสในเลขฐานสองคือ 00000000 และสีขาวก็จะถูกแทนด้วยรหัส 11111111 และสีที่อยู่ตรงกลางระหว่างสีดำกับสีขาวก็จะไล่ไปตามลำดับการนับของบิต ในเลขฐานสองดังรูปที่ 2.15 และ 2.16

	0	0000 0000
	1	0000 0001
	2	0000 0010
	3	0000 0011
	4	0000 0100
		.
		.
	253	1111 1101
	254	1111 1110
	255	1111 1111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุยาคอให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.15 ค่าของสีในรูปแบบของค่าดิจิตอล



199	192	187	97	102	101	96	98
199	195	180	111	94	97	96	96
196	195	195	100	101	103	92	100
186	201	100	94	118	115	100	103
188	202	124	116	95	102	111	128
187	205	184	112	105	102	111	124
197	201	198	95	102	192	199	105
199	200	102	98	194	195	195	110
188	193	184	109	107	194	195	224
199	194	182	114	100	92	211	221
180	192	196	100	197	99	214	222
189	182	195	95	105	109	107	238

รูปที่ 2.16 ช่วงของค่าสีในแต่ละพิกเซลของภาพ

2.2.4 การขีดแบ่ง (Threshold)

ในการประมวลผลภาพที่ต้องการความเข้มเพียง 2 ระดับหรือ 1 บิต (0 กับ 1) โดยให้ค่า 0 แทนจุดที่เป็นสีดำและค่า 1 แทนจุดที่เป็นสีขาว เนื่องจากภาพที่รับเข้ามานั้นมีความเข้มหลายระดับ เช่น ภาพ 16 ล้านสี จึงต้องแปลงข้อมูลภาพให้เป็นรูปภาพขาว-ดำ

การทำรูปภาพขาว-ดำนั้น ทำได้โดยใช้เทคนิคเทรชโฮลด์ (Threshold) เพื่อเปลี่ยนทิศทางของภาพ โดยทำการแยกกลุ่มของภาพออกเป็นส่วน ๆ และทำการแปลงภาพให้เป็นภาพระบบระดับสีเทา (Grayscale) ซึ่งมีค่าความเข้มของสี (Intensity) อยู่ระหว่าง 0-255 ให้เป็นภาพที่มีค่าความเข้มสีเพียงสองระดับ (Binary Image) โดยมีเงื่อนไขว่า ถ้าความเข้มแสงของจุดภาพใดมีค่าต่ำกว่าหรือเท่ากับค่าเทรชโฮลด์ (Threshold) ให้จุดภาพนั้นมีค่าเป็น 0 คือเป็นสีดำ ในทางตรงกันข้ามหากมีค่าที่สูงกว่าจะให้ภาพในจุดนั้นเป็น 1 หรือเป็นสีขาว กระบวนการดังกล่าวอาจถูกนำไปใช้ในการสร้างภาพขาว-ดำ โดยอาจมีกระบวนการต่าง ๆ เพิ่มเข้ามาด้วยเพื่อให้คุณภาพของภาพนั้นดียิ่งขึ้น

2.2.5 การหาขอบภาพ (Edge Detection)

การหาขอบภาพ (Edge Detection) เป็นการหาเส้นรอบวัตถุที่อยู่ในภาพ เมื่อทราบเส้นรอบวัตถุเราจะสามารถคำนวณหาพื้นที่ (ขนาด) หรือรู้จำนวนของวัตถุนั้นได้ อย่างไรก็ตามการหาขอบภาพที่ถูกต้องสมบูรณ์ไม่ใช่เรื่องง่าย โดยเฉพาะอย่างยิ่งการหาขอบของภาพที่มีคุณภาพที่ต่ำ ซึ่งมีความแตกต่างระหว่างพื้นหน้ากับพื้นหลังน้อย หรือมีความสว่างไม่สม่ำเสมอทั่วภาพ

การหาขอบภาพจะตรวจสอบว่าเส้นขอบลากผ่านหรือใกล้เคียงกับจุดใด โดยวัดจากการเปลี่ยนแปลงของความเข้มในตำแหน่งที่ใกล้เคียงกับจุดดังกล่าว ซึ่งวิธีการหาขอบนั้นมี

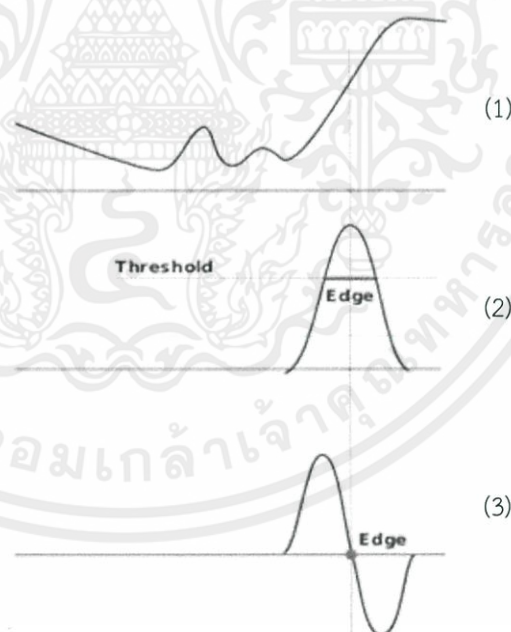
ด้วยกันหลายวิธี แต่อย่างไรก็ตามสามารถแบ่งได้เป็น 2 กลุ่มหลัก คือ Gradient method และ Laplacian method โดยในแต่ละวิธีมีรายละเอียดดังต่อไปนี้ (Castleman, Kenneth R. 1996 : 465-466 ; Fisher, Robert et al. 2004)

1) Gradient method

วิธีนี้จะหาขอบโดยการหาจุดต่ำสุดและจุดสูงสุดในรูปของอนุพันธ์อันดับ 1 โดยจุดที่เป็นขอบจะอยู่ในส่วนที่เหนือค่า Threshold จึงอาจทำให้เส้นขอบที่ได้มีลักษณะหนา ตัวอย่างวิธีการหาขอบของกลุ่มนี้ เช่น Roberts, Prewitt, Sobel และ Canny เป็นต้น

2) Laplacian method

วิธีนี้จะหาขอบโดยใช้อนุพันธ์อันดับ 2 โดยใช้จุดที่ค่า y เป็น 0 (Zero crossing) ซึ่งวิธีนี้จะใช้เวลาในการคำนวณมากกว่า Gradient Method ตัวอย่างวิธีการหา ขอบของกลุ่มนี้ เช่น Laplacian of Gaussian และ Marrs-Hildreth เป็นต้น

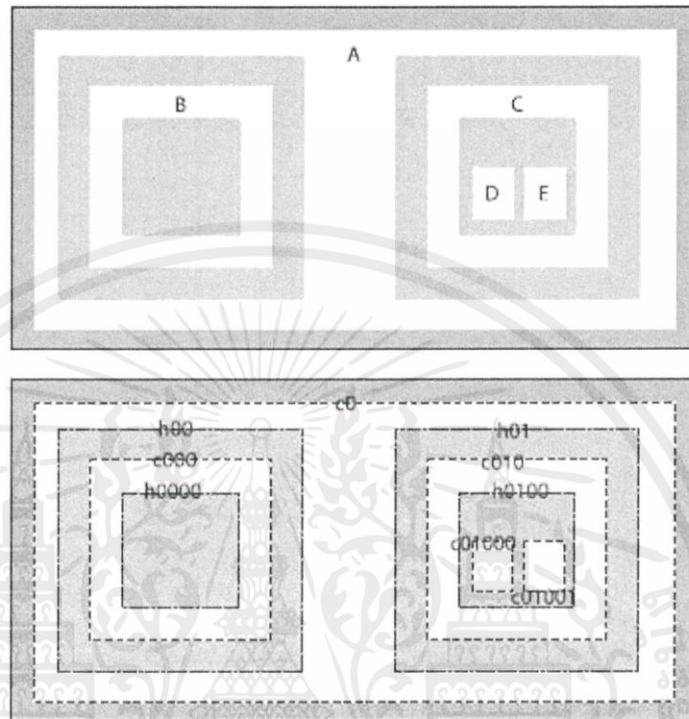


รูปที่ 2.17 กราฟแสดงการหาขอบด้วยวิธี Gradient Method (รูปที่ 2)

และ Laplacian Method (รูปที่ 3) โดยรูปที่ 1 แสดงถึงความแตกต่าง
ของระดับความเข้มของสี (GIMP 2004)

2.2.6 Contour

Contour คือ List ของจุดที่แทน Curve ในรูปภาพ



รูปที่ 2.18 ตัวอย่างการหา Contour

จากภาพเป็นภาพที่เราต้องการหา Contour ซึ่งประกอบด้วยส่วน A-E ส่วนด้านล่างคือ Contour โดย c หมายถึง Contour, h หมายถึง Hole ตัวที่เป็นเส้นประเป็น Exterior Boundaries ของพื้นที่สีขาว (ที่ไม่ใช่ 0 ก็คือกรอบของพื้นที่ขาว) ตัวเส้นจุดนั้นเป็น Interior Boundaries ของพื้นที่สีขาว หรือ Exterior Boundary ของพื้นที่ดำ (คืออยู่ในพื้นที่ขาวหรือว่าเป็นกรอบของพื้นที่สีดำ) ในความเป็นจริงระหว่างภาพ Binary และ Edge ที่ Detect ได้ จะมีความแตกต่างกันเล็กน้อยในผลลัพธ์ ดังนั้นผลลัพธ์ที่ได้ก็คือ ใน Edge วงรอบปิดหนึ่ง ๆ จะได้ Exterior Contour และ Interior Contour มาสองอันที่ขนาดใกล้เคียงกันมาก

2.2.7 การตรวจจับการเคลื่อนไหว (Motion Detection)

การตรวจจับการเคลื่อนไหวเป็นสิ่งจำเป็นที่สุดในขบวนการเรียนรู้วัตถุ (Video Understanding) ซึ่งนำเสนอวิธีการตรวจจับการเคลื่อนที่ของบุคคล โดยการตรวจจับการเคลื่อนที่เป็นส่วน ๆ (Window-based Object Detection) ซึ่งแตกต่างไปจากการตรวจจับการเคลื่อนที่แบบทั้งเฟรม (Conventional Object Detection) การตรวจจับการเคลื่อนที่

เป็นส่วน ๆ ช่วยลดความผิดพลาดการตรวจจับในสภาพแวดล้อมที่เปลี่ยนแปลงไป เช่น การเปลี่ยนความเข้มของแสง การสะท้อนของกระจก เป็นต้น ทำให้ง่ายต่อการควบคุมค่าในการตัดสินใจ (Thresholding) ว่าเป็นวัตถุที่เคลื่อนที่หรือไม่ นอกจากนี้ยังช่วยลดการประมวลผลทำให้หน่วยประมวลผลทำงานน้อยลง โดยผู้ใช้เป็นผู้กำหนดว่าจะให้ตรวจจับในบริเวณใดบ้าง ซึ่งในอนาคตอาจพัฒนาให้ระบบเรียนรู้สภาพแวดล้อมที่จะตรวจจับเองได้ รวมถึงการตรวจจับเฉพาะสิ่งที่ระบบกำหนดไว้หรือผู้บุกรุกเท่านั้น โดยสอนให้ระบบมีการรู้จำโดยใช้ทฤษฎีปัญญาประดิษฐ์ (Artificial Intellingence)

การตรวจจับการเคลื่อนไหวมีวัตถุประสงค์เพื่อพัฒนาระบบตรวจจับการเคลื่อนไหวแบบทันทีทันใด (Real Time) โดยใช้การตรวจจับการเคลื่อนไหวแบบเป็นส่วน ๆ การตรวจจับวัตถุที่เคลื่อนที่นั้นเริ่มจากการตรวจจับวัตถุที่เคลื่อนที่ (Moving Object Detection) ตามด้วยการติดตามวัตถุที่เคลื่อนที่ (Object Tracking)

การตรวจจับวัตถุที่เคลื่อนที่ (Object Tracking) นั้นสามารถทำได้หลายวิธี ซึ่งสามารถแบ่งเป็น 2 แนวทางใหญ่ ๆ คือ การสังเกตค่าการเปลี่ยนแปลงของเฟรมภาพที่อยู่ใกล้กัน (Temporal Differencing) และการค้นหาพื้นหลังและลบพื้นหลังทิ้ง (Background Modeling and Subtraction) โดยวิธีการแรกเป็นวิธีการขั้นพื้นฐานที่มีความสามารถในการตรวจจับโดยอาศัยการคำนวณของส่วนประมวลผลไม่มากนัก แต่อย่างไรก็ตามวิธีการนี้จะให้ประสิทธิภาพที่ต่ำในการนำมาใช้จริง ส่วนวิธีการหลังได้มีผู้วิจัยนำเสนอความเป็นจำนวนมาก แต่การทำงานของวิธีการหลังนี้ต้องการภาพพื้นหลังก่อนเริ่มทำงานจริง

การติดตามวัตถุที่เคลื่อนที่ (Object Tracking) เป็นสิ่งจำเป็นมากเพราะทำให้ทราบถึงทิศทางการเคลื่อนที่และการกระทำซึ่งจะนำไปสู่การรู้จำวัตถุนั้น ๆ นักวิจัยหลายท่านได้นำเสนอการติดตามวัตถุที่เคลื่อนที่ ซึ่งโดยทั่วไปแล้วสามารถจำแนกออกได้เป็น 4 ประเภทดังต่อไปนี้

- 1) การติดตามวัตถุที่เคลื่อนที่บนพื้นฐานแบบจำลอง (Model-based) เป็นความพยายามที่จะติดตามวัตถุที่มีรูปแบบเข้ากับแบบจำลองที่เตรียมไว้ โดยที่แบบจำลองเหล่านี้มักจะถูกเตรียมไว้ก่อนและต้องรู้ล่วงหน้า ดังนั้นจึงมีข้อด้อยที่ไม่สามารถทำงานกับวัตถุที่ไม่ทราบล่วงหน้าได้
- 2) การติดตามวัตถุที่เคลื่อนที่บนพื้นฐานบริเวณนั้น (Region-based) เป็นการดึงข้อมูลสำคัญ เช่น สี ลวดลาย จากนั้นทำการติดตามพื้นที่เหล่านั้น โดยหาข้อมูลของสิ่งที่ดึงออกมาใช้
- 3) การติดตามวัตถุที่เคลื่อนที่บนพื้นฐานขอบวัตถุ (Contour-based) จะขึ้นกับขอบของวัตถุที่เคลื่อนที่ไม่ได้ขึ้นกับตัววัตถุทั้งชิ้น ขอบวัตถุจะถูกสกัดออกมา

ปรับแต่งในแต่ละเฟรมที่เกิดขึ้น ดังนั้นการติดตามวัตถุที่เคลื่อนที่จะได้ผลดีหรือไม่ดีนั้นขึ้นกับขอบวัตถุเริ่มแรกที่ถูกสกัดออกมาได้ ซึ่งวิธีนี้ยากต่อการนำมาใช้ในระบบตรวจจับเพื่อรักษาความปลอดภัย

- 4) การติดตามวัตถุที่เคลื่อนที่บนพื้นฐานลักษณะเฉพาะ (Feature-based) มีจุดประสงค์ที่จะค้นหาและติดตามการติดตามวัตถุที่เคลื่อนที่โดยใช้คุณลักษณะเฉพาะบางอย่าง เช่น เส้นรอบวง พื้นที่ของวัตถุ ขอบ มุม เป็นต้น ซึ่งวิธีการนี้จะให้ผลไม่ดี หากวัตถุถูกบดบังบางส่วน เมื่อนำวิธีการนี้มาพัฒนาตัวโปรแกรมพบว่าสามารถตรวจจับการเคลื่อนที่ของวัตถุได้มากกว่าหนึ่งวัตถุ และสามารถแก้ไขปัญหาของการตั้งค่าขีดแบ่งเพียงค่าเดียว เฟรมที่นำเข้ามาในระบบไม่จำเป็นต้องมีการเรียนรู้พื้นหลังมาก่อน ทำให้ไม่ต้องการความรู้ก่อนประสบการณ์ ผลจากการแบ่งพื้นที่ตรวจจับเป็นบล็อกทำให้ระบบสามารถทำงานได้รวดเร็วขึ้น

2.2.8 วิธีการเข้าคู่รูปแบบ (Template Matching)

การทำงานของวิธีการเข้าคู่รูปแบบ (Template Matching) คือ การนำรูปภาพที่ต้องการหาค่าไปเปรียบเทียบกับรูปแบบตัวอย่าง (Template) ซึ่งรูปแบบตัวอย่างนี้จะเก็บค่าและกำหนดลักษณะสำคัญต่าง ๆ ที่สามารถแยกความแตกต่างของตัวอักษรต่าง ๆ ได้ และเมื่อนำภาพตั้งต้นที่มีมาเปรียบเทียบกับรูปแบบที่มี เพื่อตรวจสอบความคล้ายคลึงกันของภาพทั้งสอง ซึ่งอาศัยการระดับที่ตั้งขึ้นมาเพื่อตรวจสอบหรือตัดสินใจ แต่ข้อเสียของวิธีการนี้มีค่อนข้างมากด้วย เช่น มีความอ่อนไหวต่อข้อมูลที่มีการแทรกซ้อน ขนาด และความเอียง ซึ่งปัญหาเหล่านี้สามารถแก้ไขได้ โดยต้องอาศัยการทำงานในขั้นตอนการประมวลผลขั้นต้นที่ดี Template Matching เป็นเทคนิคที่นิยมใช้กันอย่างกว้างขวาง เพราะให้ผลลัพธ์ที่ค่อนข้างมีประสิทธิภาพแต่ข้อเสียคือช้า เทคนิคนี้เหมาะสำหรับการหาลักษณะเฉพาะที่มีขนาดไม่ใหญ่บนภาพ เทคนิค Template Matching จะมีความคล้ายคลึงกับเทคนิคที่เรียกว่า Convolution สำหรับเทคนิค Template Matching เราจะทำการเคลื่อน Template ไปยังจุดต่าง ๆ จากซ้ายไปขวา บนลงล่าง เพื่อคำนวณหาค่าความเหมือนของ Template กับบริเวณต่าง ๆ บนภาพ ซึ่งขบวนการที่ทำเช่นนี้เรียกว่าการ Correlation และค่าความเหมือนนี้มีชื่ออีกอย่างว่า Correlation Coefficient สามารถคำนวณได้จากสมการที่ (5)

$$c = \frac{\sum_{i=1}^{W \times H} (f_i - \langle f \rangle) \times \sum_{j=1}^{N \times M} (w_j - \langle w \rangle)}{\sqrt{\sum_{i=1}^{W \times H} (f_i - \langle f \rangle)^2 \times \sum_{j=1}^{N \times M} (w_j - \langle w \rangle)^2}} \quad (5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในโครงการวิจัยเท่านั้น ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้ c คือ ค่า Correlation Coefficient

f คือ ค่าความเข้มสีแต่ละจุดสีบนภาพ

w คือ ค่าความเข้มสีแต่ละจุดบน Template

$\langle f \rangle$ และ $\langle w \rangle$ คือ ค่าเฉลี่ยสีของภาพ (ขนาดเท่า Template) และ Template ตามลำดับ

ดังนั้นถ้าบริเวณใดของภาพเหมือนกับ Template พอดี ค่าที่ได้ก็จะมีค่าเป็น 1 แต่หากไม่เหมือนกันค่าก็จะมีค่าลดลง



รูปที่ 2.19 ภาพตัวอย่างการใช้ Template Matching ในการค้นหาภาพ

2.3 FLARManager

FLARManager เป็น Framework ชนิดหนึ่งที่ช่วยให้เราสามารถใช้งานไลบรารี FLARToolKit ในการสร้างแอปพลิเคชันได้ง่ายขึ้น FLARManager สามารถใช้งานได้กับ 3D Engine ที่หลากหลาย และนอกเหนือจากนั้นยังสามารถจัดการเกี่ยวกับการเพิ่ม การอัปเดตและการลบ Marker ได้อย่างสะดวกสบาย และยิ่งไปกว่านั้น FLARManager ยังสนับสนุนการตรวจจับและจัดการรูปแบบของ Marker แบบ Multi-Marker ได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับไลบรารีใน FLARManager ที่ใช้สำหรับ Tracking มีดังต่อไปนี้

- 1) FLARToolKit
- 2) flare*tracker
- 3) flare*NFT

และสำหรับ 3D Framework ใน FLARManager ที่ใช้สำหรับการ Render มีดังต่อไปนี้

- 1) Alternativa3D
- 2) Away3D
- 3) Away3D Lite
- 4) Papervision3D
- 5) Sandy3D

หมายเหตุ FLARManager ยังไม่สามารถใช้ flare* ร่วมกับ Away3D, Sandy3D, Alternativa3D ได้

FLARToolKit เป็น Flash Actionscript Version ของ ARToolKit ซึ่งสามารถถูกนำมาพัฒนาใน web-base ได้โดยง่าย FLARToolKit ถูกเผยแพร่โดยได้รับการสนับสนุนจากนักพัฒนาเป็นอย่างมาก และในเว็บไซด์บางเว็บไซด์ก็ได้นำมาประยุกต์ใช้ทำเป็นแอปพลิเคชันอย่างมากมาย

FLARToolKit เป็นเครื่องมือที่ประกอบด้วยฟังก์ชันต่าง ๆ ที่จำเป็นต้องใช้ในการพัฒนาโปรแกรมในด้านเทคโนโลยี AR นอกจากนี้แล้วยังได้ประกอบด้วยไลบรารีที่สามารถรองรับได้หลากหลาย Platform และพยายามลดข้อแตกต่างของไลบรารี เพื่อให้ง่ายต่อการนำไปประยุกต์ใช้ใน Platform ที่แตกต่างกันและมีประสิทธิภาพ FLARToolKit ใช้ Papervision3D ช่วยในการแสดงผล พร้อมทั้งจัดการกับอุปกรณ์และไลบรารีของกล้องวิดีโอด้วย

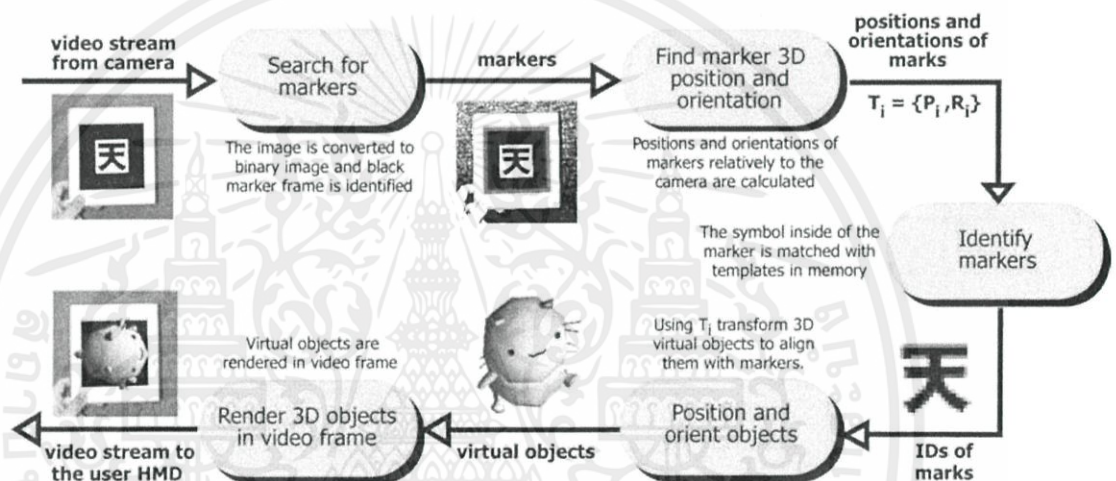
2.3.1 หลักการพื้นฐานของ FLARToolKit

FLARToolKit อนุญาตให้วัตถุเสมือนซ้อนทับภาพวิดีโอของโลกแห่งความจริง สิ่งที่เกิดขึ้นนั้นไม่ใช่เวทมนต์ ความลับอยู่ในสี่เหลี่ยมสีดำที่ใช้เป็น Tracking Marker โดยมีขั้นตอนในการ Tracking ดังนี้

- 1) กล้องวิดีโอทำการจับภาพจากโลกแห่งความเป็นจริงและส่งไปยังคอมพิวเตอร์
- 2) ซอฟต์แวร์คอมพิวเตอร์ทำการค้นหากรอบรูปทรงสี่เหลี่ยมใด ๆ ในแต่ละวิดีโอเฟรม

- 3) ถ้าตรวจจับเจอกรอบรูปทรงสี่เหลี่ยมใด ๆ ซอฟต์แวร์จะทำการคำนวณทางคณิตศาสตร์เกี่ยวกับตำแหน่งที่สัมพันธ์กันกับกล้องวิดีโอ

- 4) เมื่อรู้ตำแหน่งของกล้องวิดีโอ คอมพิวเตอร์กราฟิกจะถูกวาดจากตำแหน่งเดียวกัน จากนั้นจะทำการเปรียบเทียบค่า Marker ที่ตรวจจับได้จากโลกเสมือนจริงว่ามีค่าตรงกับค่า Marker ที่รับมาหรือไม่
- 5) ถ้าค่าที่รับมาตรงกับค่า Marker ที่มีอยู่จะทำการวาดโมเดลลงไปบนวิดีโอในโลกแห่งความจริงบน Marker สีเหลือง
- 6) ขั้นตอนสุดท้าย เมื่อผู้ใช้ดูผ่านทางจอแสดงผล ผู้ใช้จะเห็นภาพสามมิติซ้อนทับกันบนโลกแห่งความจริง



รูปที่ 2.20 แสดงขั้นตอนการทำงานของ FLARToolKit

มีข้อจำกัดสำหรับระบบคอมพิวเตอร์เกี่ยวกับการมองเห็นของระบบเสมือนจริงในความเป็นจริงวัตถุเสมือนจะปรากฏให้เห็นก็ต่อเมื่อ Marker อยู่ในส่วนที่กล้องสามารถจะตรวจจับได้หรือถูก Tracking ซึ่งอาจถูกจำกัดด้วยขนาดและการเคลื่อนที่ของวัตถุเสมือน นอกจากนั้นหาก Marker ถูกปิดหรือโดนบังด้วยมือหรืออื่น ๆ จะทำให้วัตถุเสมือนนั้นหายไปหรือไม่ปรากฏออกมา สำหรับข้อจำกัดอีกข้อหนึ่งว่าด้วยเรื่องปัญหาขนาดทางกายภาพของ Pattern ซึ่งระยะห่างระหว่างกล้องกับ Marker ที่สามารถจะทำการตรวจจับได้ แสดงให้เห็นดังตารางที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 แสดงระยะห่างสูงสุดที่สามารถทำการ Tracking ตามขนาดของ Marker ที่แตกต่างกัน

Pattern Size (inches)	Usable Range (inches)
2.75	16
3.50	25
4.25	35
7.37	50


ผลที่ได้จากการทดลองสร้าง Pattern ที่มีขนาดที่ต่างกันวางตั้งฉากกับกล้อง และเคลื่อนกล้องออกมาจนกระทั่งวัตถุเสมือนนั้นหายไป จะเห็นได้ว่า Pattern ที่มีขนาดใหญ่ขึ้นมีประสิทธิภาพมากกว่า ซึ่งเห็นได้จากตารางที่ 2.2 และในกรณีที่ Pattern มีขนาดเดียวกันแต่มีความซับซ้อนของ Pattern แตกต่างกัน Pattern ที่มีขนาดซับซ้อนมากกว่า จะมีระยะทางในการแสดงวัตถุเสมือนจริงที่ลดลงไปอีก ดังนั้นนอกจากเรื่องขนาดแล้ว ข้อจำกัดในการมองเห็นวัตถุเสมือนยังขึ้นอยู่กับความซับซ้อนของ Marker อีกด้วย

ผลของการ Tracking นั้นมีปัจจัยเรื่องของแสงเข้ามาเกี่ยวข้อง ในกรณีที่แสงมากเกินไปนั้นอาจจะทำให้เกิดการสะท้อนของแสงบน Marker ทำให้เป็นผลกระทบต่อการ Tracking Marker นั้นเอง ดังนั้นควรเลือกว่าวัสดุที่จะใช้มาเป็น Marker ให้มีความมันวาวน้อยที่สุดจึงจะมีผลกระทบน้อยต่อการ Tracking Marker

2.3.2 หลักการพัฒนา

การพัฒนาแอปพลิเคชัน AR โดยใช้ FLARToolKit นั้นประกอบด้วย 2 ส่วนหลัก ๆ ด้วยกัน คือ การเขียนแอปพลิเคชันและส่วนของการทำให้โปรแกรมรู้จัก Marker ที่จะใช้กับโปรแกรม (Marker Training) การเขียนโปรแกรมเพื่อเรียกใช้ FLARToolKit เป็นเรื่องที่ทำได้ง่ายมาก เนื่องจากพื้นฐานของ FLARToolKit มีฟังก์ชันพื้นฐานที่ไม่เยอะเยอะมาก ซึ่งจะทำให้โครงร่างของโปรแกรมดูไม่ซับซ้อน นอกจากนี้ยังมีตัวอย่างโปรแกรมซึ่งเราสามารถนำมาใช้เป็นต้นแบบในการเขียนโปรแกรมของเรา สำหรับในส่วนของการ Marker Training นั้น ทาง FLARToolKit ได้จัดเตรียมเครื่องมือที่ง่ายต่อการใช้งานไว้ให้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Initialization	1. Initialize the video capture and read in the marker pattern files and camera parameters.
	2. Grab a video input frame.
	3. Detect the markers and recognized patterns in the video input frame.
	4. Calculate the camera transformation relative to the detected patterns.
	5. Draw the virtual objects on the detected patterns.
Shutdown	6. Close the video capture down.

รูปที่ 2.21 โครงร่างของโปรแกรมประยุกต์ทางด้าน AR ที่ใช้ FLARToolKit

ในขั้นตอนที่ 2-5 จะวนทำงานไปเรื่อย ๆ จนกว่าเราจะออกจากแอปพลิเคชัน ซึ่งกระบวนการเหล่านี้จำเป็นต้องมีการรับค่าจากผู้ใช้หรือแอปพลิเคชันอื่น ๆ ที่เกี่ยวข้อง ในขณะที่ขั้นตอนที่ 1 จะเป็นส่วนของการเตรียมความพร้อมก่อนการทำงานและขั้นตอนที่ 6 จะเป็นส่วนของการสิ้นสุดการทำงานของแอปพลิเคชัน

2.4 ActionScript 3.0

ActionScript เป็น Object-Oriented Language ซึ่งถูกพัฒนาขึ้นโดย Macromedia Inc (ปัจจุบัน Adobe Systems เป็นเจ้าของ) ซึ่งเป็นภาษาพื้นฐานของ ECMAScript (หมายความว่ามันเป็น Superset ของ syntax และถูกรู้จักกันมากขึ้นอย่างกว้างขวางว่าเป็น JavaScript) และใช้เป็นหลักสำหรับการพัฒนาเว็บไซต์และซอฟต์แวร์ที่เป็นการกำหนดเป้าหมายแพลตฟอร์ม Adobe Flash Player ซึ่งใช้บนหน้าเว็บในรูปแบบของการฝังตัวของไฟล์ SWF เป็นภาษาที่เป็น Open-source (หมายถึงโปรแกรมที่ไม่มีการปิดบังโค้ด เพื่อให้คนอื่นสามารถร่วมพัฒนาต่อยอดได้ และที่สำคัญคือไม่เสียเงิน)

ActionScript ถูกออกแบบเบื้องต้นสำหรับการควบคุมภาพเคลื่อนไหวเวกเตอร์ที่ง่ายในรูปแบบ 2 มิติ ซึ่งทำใน Adobe Flash (ชื่อเดิมคือ Macromedia Flash) ในช่วงแรก ๆ จะมุ่งเน้นใน animation และนำเสนอเนื้อหาของ flash ที่มีคุณสมบัติในการตอบโต้ได้เล็กน้อยจึงค่อนข้างที่จะทำการเขียนสคริปต์ที่ค่อนข้างจำกัด ในรุ่นถัดมาได้มีการเพิ่มฟังก์ชันการทำงานที่ช่วยให้สำหรับการสร้างเกมที่อยู่บน Web-based และการใช้แอปพลิเคชันทางด้านสื่ออย่างมากมาย (ตัวอย่างเช่น วิดีโอ) ในปัจจุบัน ActionScript เป็นที่เหมาะสมสำหรับการใช้งานในบางแอปพลิเคชันที่มีฐานข้อมูลสำหรับ Flash ในเวอร์ชันยุคแรกเริ่มนั้น เวอร์ชันที่ใช้คือ 1.0 ซึ่งการทำงานนั้นค่อนข้างจะครบถ้วนสมบูรณ์ระดับหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับความรู้เท่านั้น เพื่อการศึกษานี้เท่านั้น ไม่ควรนำเอกสารนี้ไปใช้เพื่อการค้า
ไม่ว่ากรณีใดๆก็ตาม ผู้อ่านมีหน้าที่ในการแจ้งการเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash MX 2004 ทำให้รู้จัก ActionScript 2.0 ซึ่งการเขียนโปรแกรมภาษาสคริปต์ที่เหมาะสมกับการพัฒนาของ Flash Application ทำให้ประหยัดเวลาในการเขียนสคริปต์ ซึ่งจะทำให้เกิดความยืดหยุ่นมากขึ้นในการแก้ไข

Flash Player 9 alpha (ในปี 2006) เป็นรุ่นที่ใหม่ของ ActionScript ซึ่งได้รับการเผยแพร่เรียกว่า ActionScript 3.0 เป็นภาษาโปรแกรมเชิงวัตถุที่ช่วยให้การควบคุมเพิ่มมากขึ้นและสามารถนำกลับมาใช้ใหม่ได้เมื่อมีการสร้างแอปพลิเคชันแฟลชที่ซับซ้อนขึ้น ในเวอร์ชันนี้มีวัตถุประสงค์คือการคอมไพล์และรันของ ActionScript Virtual Machine ซึ่งมีการเขียนใหม่จากพื้นฐานเดิม ด้วยเหตุนี้โค้ดที่เขียนใน ActionScript 3.0 ได้กำหนดเป้าหมายโดยทั่วไปสำหรับ Flash Player 9 ซึ่งในรุ่นที่ใหม่ขึ้นไปจะไม่สามารถทำงานบนรุ่นเก่า ๆ ได้ นอกจากนั้น ActionScript 3.0 ยังเป็น รูปแบบพื้นฐานของ Flex 2 API

แต่ทั้งนี้กระแสนการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming เป็นการเขียนโปรแกรมโดยมองสิ่งต่าง ๆ เป็นวัตถุที่มีคุณสมบัติ, การดำเนินงานต่าง ๆ ฯลฯ) นั้นเป็นสิ่งที่ไม่ควรจะมองข้าม ถึงแม้ว่า ActionScript 1.0 นั้นจะเขียน OOP ได้ก็ตาม แต่ไม่ใช่รูปแบบการเขียน OOP ที่เป็นมาตรฐานทั่วไปสักเท่าไรนัก ดังนั้น ActionScript 2.0 จึงได้ถูกนำมาใช้โดยเราสามารถเขียน ActionScript เป็นไฟล์สำหรับคลาสแยกออกมาต่างหาก โดยมีการจัดวางรูปแบบการเปิดและปิดการประกาศตัวแปรต่าง ๆ ให้ตรงตามลักษณะของ OOP มากขึ้น

ถึงแม้ว่า ActionScript 3.0 จะเปลี่ยนแปลงรูปแบบการเขียนไปหลายอย่างก็ตาม แต่พื้นฐานความเข้าใจเบื้องต้นของ ActionScript 2.0 ก็ถือว่ายังใช้ได้อยู่ หรือหาก Developer ที่มีพื้นฐานการเขียนโปรแกรมแบบ OOP อยู่แล้วบางคนนั้นเริ่มต้นกับ ActionScript 3.0 เลยก็สามารถทำได้

Flash libraries สามารถใช้กับ XML ของเบราว์เซอร์เพื่อแสดงเนื้อหาของเบราว์เซอร์ได้ เทคโนโลยีนี้ถูกรู้จักในฐานะ XML เหมือน AJAX

ในทางการเขียนโปรแกรมไม่ว่าจะเป็นภาษาไหนก็ตาม เรามักจะพบกฎทางด้านไวยากรณ์การเขียนโปรแกรมแทบทั้งนั้น และกฎทางไวยากรณ์นี้เราจะต้องปฏิบัติตามอย่างเคร่งครัดห้ามละเว้นมิเช่นนั้น จะทำให้ไม่สามารถคอมไพล์หรือรันโปรแกรมได้ กฎทางด้านไวยากรณ์ของ ActionScript มีดังต่อไปนี้

1) Dot Syntax (.)

เราใช้เครื่องหมายจุด (.) เพื่อแสดงถึง TargetPath ของ Timeline เช่น

root.mc_1.mc_2 เนื่องจาก ActionScript เป็น Object-Oriented Language เราจึง

ใช้เครื่องหมายจุดสำหรับระบุถึง Properties หรือ Method ที่มีใน Movie Clip การใช้

งานนั้นเราจะพิมพ์จุดขึ้นกลางระหว่าง Instance Name กับ Properties เช่น

mc_1.play()

2) Case Sensitivity

ภาษา ActionScript เป็นภาษาที่มีความเข้มงวดเรื่องตัวพิมพ์ใหญ่-เล็กของตัวอักษร จึงทำให้เมื่อเราเขียน ActionScript และมีการประกาศตัวแปรชื่อว่า movieClip จะแตกต่างกับตัวแปรที่ชื่อว่า MovieClip

3) Semicolons (;)

ในภาษา ActionScript เมื่อจบคำสั่งจะต้องปิดท้ายด้วยเครื่องหมาย Semicolons (;) ทุกครั้ง

```
var x:number = 20;
break;
```

4) Comment

การเขียนคอมเมนต์ในโปรแกรมเป็นการอธิบายว่าโปรแกรมนั้นมีรายละเอียดอย่างไร โดยข้อความที่ถูกคอมเมนต์นั้นจะไม่มีผลต่อการแปลคำสั่งของ ActionScript โดยรูปแบบของการเขียนคอมเมนต์ใน ActionScript มีดังนี้

```
// this is first comment
/* this is second comment */
```

จากตัวอย่างจะเห็นได้ว่าการคอมเมนต์โดยใช้เครื่องหมาย (//) จะเป็นการคอมเมนต์หนึ่งบรรทัดและการใช้เครื่องหมาย (/*.....*/) เป็นการคอมเมนต์หลาย ๆ บรรทัด

การเขียนคอมเมนต์ไม่จำเป็นต้องอยู่ที่ตอนต้นของบรรทัดเสมอไป สามารถเขียนคอมเมนต์ไว้ในบรรทัดเดียวกับประโยคคำสั่งต่าง ๆ ของ ActionScript ได้

```
my_txt.text = "hello world"; // assigns text to text fields
```

5) Quotation Marks (“”)

ในการแสดงข้อมูลที่เป็นตัวอักษรในภาษา ActionScript นั้นจะต้องใช้ Quotation Marks ในการแบ่งแยกตัวอักษรจริงที่จะใช้แสดงผลหรือว่าชื่อตัวแปร เพราะ

ภาษา ActionScript นั้นจะใช้ตัวอักษรเป็นตัวสร้างสคริปต์ ฉะนั้นจึงจำเป็นต้องแยก
อย่างชัดเจนถึง ชุดตัวอักษรที่ใช้แสดงในโปรแกรม ตัวอย่างเช่น

```
trace("x");
```

```
trace(x);
```

โดยในคำสั่งแรกจะมีการแสดงผลตัวอักษร x บนหน้าจอ Output แต่ในคำสั่ง
บรรทัดที่สองจะแสดงค่าที่อยู่ในตัวแปร x บนหน้าจอ Output



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและพัฒนา

หลังจากได้เรียนรู้หลักการและทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับเทคโนโลยี Augmented Reality ในบทที่ 2 แล้ว บทนี้จะกล่าวถึงขั้นตอนการออกแบบและพัฒนาแอปพลิเคชัน โดยการนำหลักการและทฤษฎีต่าง ๆ เหล่านั้นมาประยุกต์ใช้

3.1 หลักการทำงานของเทคโนโลยีเสมือนจริง (Augmented Reality)



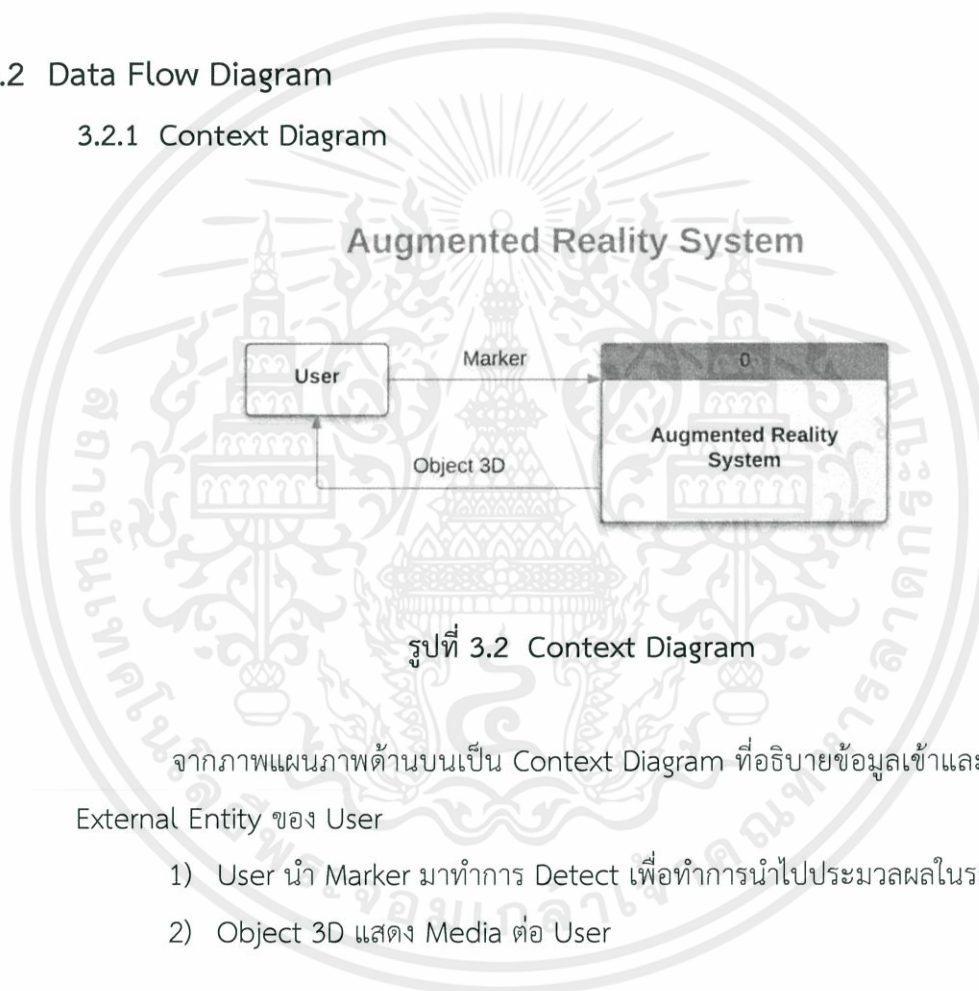
รูปที่ 3.1 แผนภาพการทำงานของเทคโนโลยีเสมือนจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการแข่งขันเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หลักการทำงานของแอปพลิเคชันชนิด Augmented Reality มี 3 กระบวนการ ดังต่อไปนี้
- 1) การวิเคราะห์ภาพ (Image Analysis) เป็นขั้นตอนการค้นหา Marker ที่มีการเก็บข้อมูลขนาดและรูปแบบของ Marker เพื่อนำมาวิเคราะห์รูปแบบของ Marker
 - 2) การคำนวณค่าตำแหน่งเชิง 3 มิติ (Pose Estimation) ของ Marker เทียบกับกล้อง
 - 3) กระบวนการสร้างภาพสองมิติจากโมเดลสามมิติ (3D Rendering) เป็นการเพิ่มข้อมูลเข้าไปในภาพโดยใช้ค่าตำแหน่งเชิง 3 มิติที่คำนวณได้จนได้ภาพเสมือนจริง

3.2 Data Flow Diagram

3.2.1 Context Diagram

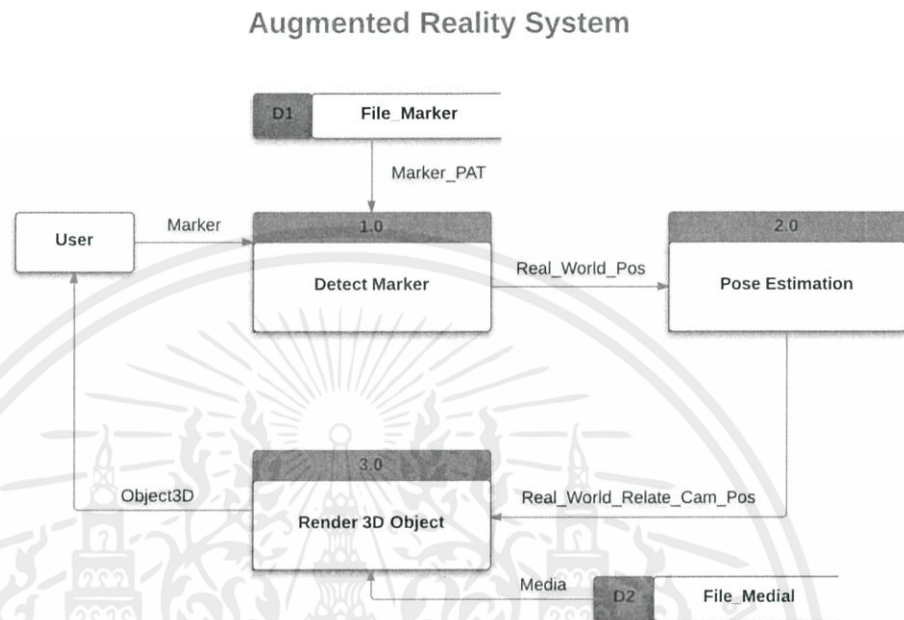


จากภาพแผนภาพด้านบนเป็น Context Diagram ที่อธิบายข้อมูลเข้าและออกของ External Entity ของ User

- 1) User นำ Marker มาทำการ Detect เพื่อทำการนำไปประมวลผลในระบบ
- 2) Object 3D แสดง Media ต่อ User

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 Diagram Level-0 หรือ Parent Diagram



รูปที่ 3.3 Diagram 0 หรือ Parent Diagram

จากภาพเป็น Diagram Level 0 หรือ Parent Diagram เป็น Diagram ที่เป็นระดับถัดจาก Context Diagram ซึ่งจะเห็นได้ว่าในระดับนี้มีกระบวนการต่าง ๆ ที่สำคัญหลัก ๆ คือ

- 1) Detect Marker เป็นกระบวนการที่ทำการ Detect Marker จาก User
- 2) Pose Estimation เป็นกระบวนการคำนวณค่าตำแหน่งเชิง 3 มิติจากโลกความจริงให้สอดคล้องกับโลกเสมือน
- 3) Render 3D Object เป็นกระบวนการ Render วัตถุแสดงให้ User เห็นเป็นโลกเสมือนจริง

นอกจากนั้นยังมีแหล่งเก็บข้อมูล (Data store Symbol) ที่เก็บข้อมูลสำคัญ ได้แก่

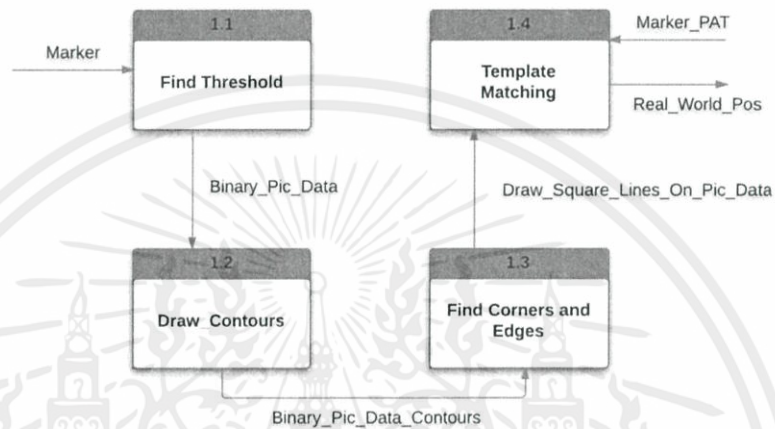
- 1) File_Marker เป็นแหล่งเก็บข้อมูลที่เก็บไฟล์ Marker ที่เป็นชนิด PAT File
- 2) File_Media เป็นแหล่งเก็บข้อมูลที่เก็บไฟล์ Media ต่าง ๆ เช่น ภาพ วิดีโอ

อนิเมชัน เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 Diagram Level-1

Augmented Reality System



รูปที่ 3.4 Diagram Level-1

จากภาพเป็น Diagram Level 1 หรือ Child Diagram ของ Process ที่จะอธิบายดังต่อไปนี้

- 1) Find Threshold เป็นการเปลี่ยนทิศทางของภาพ โดยทำการแยกกลุ่มของภาพออกเป็นส่วน ๆ และทำการแปลงภาพให้เป็นภาพระดับสีเทา (Grayscale) ซึ่งมีค่าความเข้มของสี (Intensity) อยู่ระหว่าง 0-255 ให้เป็นภาพที่มีค่าความเข้มสีเพียงสองระดับ (Binary Image) โดยมีเงื่อนไขว่า ถ้าความเข้มแสงของจุดภาพใดมีค่าต่ำกว่าหรือเท่ากับค่าเทรชโฮลด์ (Thresholds) ให้จุดภาพนั้นมีค่า เป็น 0 คือเป็นสีดำ ในทางตรงกันข้ามหากมีค่าที่สูงกว่าจะให้ภาพในจุดนั้นเป็น 1 หรือเป็นสีขาว
- 2) Draw_Contours เป็นการหาเส้นรอบรูปต่าง ๆ ที่ Webcam ตรวจจับได้
- 3) Find Corners and edges เป็นการหามุมและขอบต่าง ๆ ที่เป็นสี่เหลี่ยม จากนั้นทำการวาดกรอบล้อมรอบมุมและขอบสี่เหลี่ยมนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 โปรแกรมและชุดเครื่องมือที่ใช้ในการพัฒนาแอปพลิเคชัน

3.3.1 ซอฟต์แวร์

- 1) Adobe Flash Builder 4.6
- 2) ระบบปฏิบัติการ Windows 7 และ Mac OS
- 3) Library ที่ช่วยในการพัฒนา คือ ARToolKit

3.3.2 ฮาร์ดแวร์

- 1) Notebook TOSHIBA Qosmio F750-1016XT
- 2) CPU : Intel Core i5-2450M 2.5 GHz, RAM 8 GB, HDD 640 GB
- 3) Graphic Chip : nVidia GeForce GT 540M (2 GB GDDR3)
- 4) Logitech Webcam C600



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การพัฒนาโปรแกรม

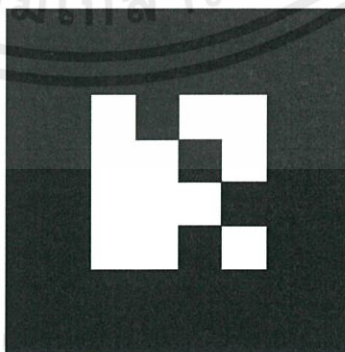
ในบทนี้จะเป็นการนำเอาทฤษฎีและหลักการที่เกี่ยวข้องกับเทคโนโลยีเสมือนจริงมาใช้ในการพัฒนาแอปพลิเคชัน โดยจะกล่าวถึงขั้นตอนต่าง ๆ ตั้งแต่การสร้าง Marker ไปจนถึงการเขียนโปรแกรม

4.1 การสร้าง Marker และการนำไปใช้งาน

4.1.1 คำนิยามของ Marker

Marker คือรูปสี่เหลี่ยมที่ FLARToolKit สามารถเรียนรู้และติดตามในวิดีโอสตรีม ซึ่ง Marker เป็นรูปแบบทางกายภาพที่เราสามารถสร้างขึ้นหรือพิมพ์ออกมาได้ โดย FLARToolKit มาพร้อมกับไฟล์ PDF สำหรับบาง Marker ที่สร้างไว้ล่วงหน้าแล้ว และในการสร้าง Marker นั้นมีข้อจำกัดเพียงเล็กน้อยดังต่อไปนี้

- 1) Marker จะต้องเป็นรูปทรงสี่เหลี่ยม
- 2) Marker จะต้องมีส่วนขอบที่ต่อเนื่องกัน (สีดำหรือสีขาวล้วน) และมันจะต้องอยู่บนพื้นหลังที่มีสีตัดกัน (โดยทั่วไปจะตรงข้ามกับสีของเส้นขอบ) โดยค่าเริ่มต้นจะมีความหนาของขอบเป็น 25% ของความยาวของขอบของ Marker
- 3) ข้อจำกัดสุดท้าย คือ พื้นที่ภายในเส้นขอบ หรือที่เรียกว่า Marker Image จะต้องไม่เป็นรูปสมมาตร พื้นที่ภายในสามารถเป็นสีดำ สีขาว หรือสีอื่นต่าง ๆ โดย FLARToolKit จะติดตามด้วยความแม่นยำที่มากขึ้นเมื่อ Marker มีสีสัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน ให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.1 ตัวอย่างของ Marker ที่ FLARToolKit ได้สร้างไว้ให้แล้ว

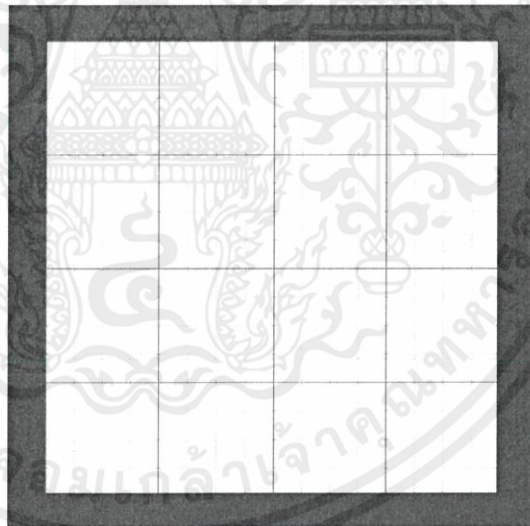
4.1.2 คำนิยามของ Pattern

ไฟล์ Pattern เป็นไฟล์ที่เก็บข้อมูลของรูปภาพในจุดศูนย์กลางของ Marker เมื่อเปิดโปรแกรม FLARToolKit จะทำการโหลดไฟล์ Pattern ขึ้นมาหนึ่งไฟล์หรือมากกว่านั้น เพื่อที่จะนำไปเปรียบเทียบและสามารถรู้ได้ว่า Marker ใดอยู่บนวัตถุโฮโลแกรม

4.1.3 การสร้าง Marker ใหม่ (ทำตามค่าเริ่มต้น คือ ความหนาของขอบเป็น 25% ของความยาวของขอบของ Marker)

เราสามารถสร้าง Marker ขนาดใด ๆ ก็ได้ตามความต้องการ และสามารถผสม Marker ที่มีขนาดแตกต่างกันได้ ซึ่งขั้นตอนในการสร้าง Marker ใหม่ โดยใช้โปรแกรม Adobe Photoshop สามารถทำได้ดังนี้

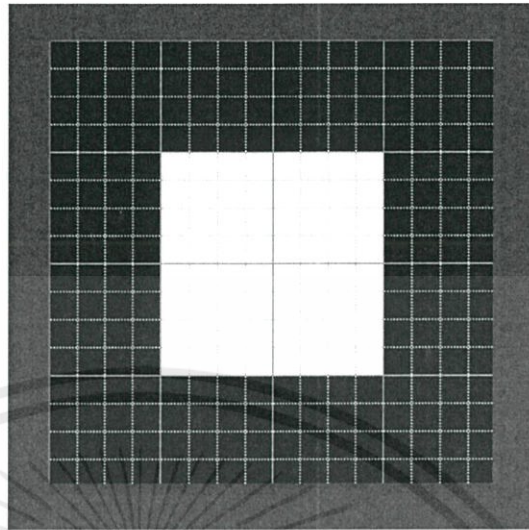
- 1) สร้างไฟล์ใหม่ขึ้นมาขนาด 285 x 285 pixel
- 2) เข้าไปที่เมนู View เลือกเมนู Show แล้วเลือก Grid จากนั้นเราจะเห็นตารางขนาด 4 x 4 ช่องใหญ่ หรือขนาด 16 x 16 ช่องเล็ก



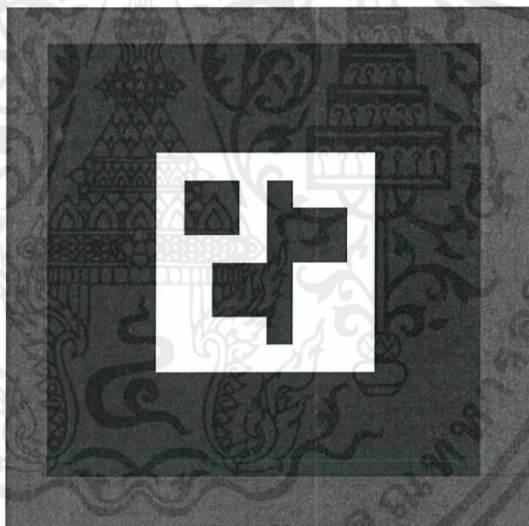
รูปที่ 4.2 ไฟล์ถูกแบ่งออกเป็น 4 x4 ช่องใหญ่ หรือขนาด 16 x 16 ช่องเล็ก

- 3) ทำการใส่สีดำในช่องที่เป็นเส้นขอบ (ช่องใหญ่รอบนอกทั้งหมด) ส่วนพื้นที่ภายในเส้นขอบนั้นสามารถจะใส่อะไรลงไปก็ได้ อาจจะเป็นรูปภาพหรือทำการใส่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



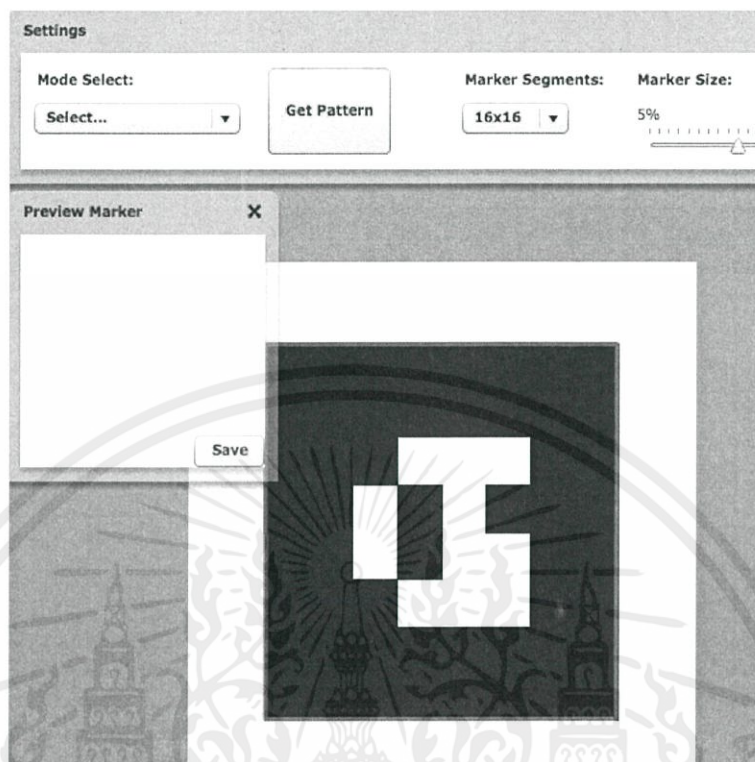
รูปที่ 4.3 แสดงการใส่สีดำเพื่อสร้างเป็นขอบของ Marker



รูปที่ 4.4 ตัวอย่างของ Marker ที่พร้อมจะนำไปใช้งาน

4.2 การใช้งานโปรแกรม ARToolKit Marker Generator

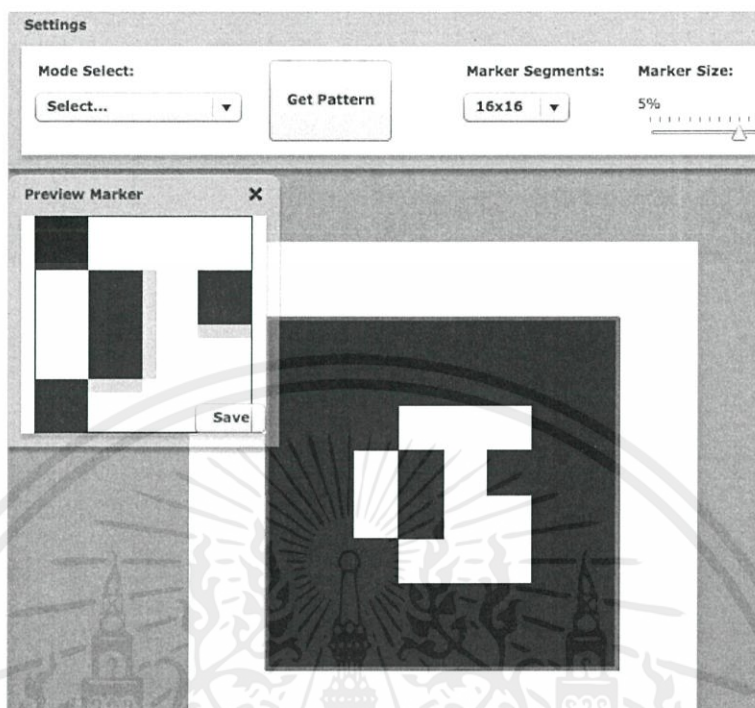
โปรแกรม ARToolKit Marker Generator เป็นโปรแกรมที่ใช้ในการสร้างไฟล์ .pat จากไฟล์ Marker ที่เราสร้างขึ้น โดยเราสามารถไปใช้โปรแกรมนี้ได้ที่เว็บไซต์ <http://flash.tarotaro.org/blog/2008/12/14/artoolkit-marker-generator-online-released/> เริ่มต้นด้วยการอัปโหลดไฟล์รูป Marker เข้าไปในโปรแกรม โดยวิธีการอัปโหลดไฟล์รูปนั้นจะทำได้ 2 รูปแบบ คือ เลือกไฟล์ ไม่ว่าจะทำได้ทางอื่น อีกหนึ่งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ จากไต่เรกทอรีในคอมพิวเตอร์ หรือทำการใช้กล้องเว็บแคมในการจับภาพแล้วทำการอัปโหลดไฟล์



รูปที่ 4.5 โปรแกรม ARToolKit Marker Generator หลังจากทำการอัปโหลดไฟล์ Marker

เมื่อโปรแกรมสามารถตรวจจับได้แล้วว่าไฟล์ที่เราอัปโหลดไปเป็นไฟล์ Marker มันจะทำการตีกรอบรูป Marker ด้วยเส้นสีแดง นอกจากนี้เรายังสามารถกำหนดค่า Marker Segments เพื่อให้มีความละเอียด และ Pattern Size เพื่อเพิ่มหรือลดขนาดพื้นที่ของ Marker ที่จะนำไปทำเป็นไฟล์ Pattern เมื่อกำหนดทุกอย่างเรียบร้อยแล้ว ทำการกดปุ่ม Get Pattern จากนั้นโปรแกรมจะแสดงพื้นที่ของ Marker ที่นำไปทำ Pattern เมื่อตรวจสอบว่าเป็นไปตามที่ต้องการแล้ว จึงทำการกดปุ่ม Save เพื่อบันทึกไฟล์ .pat ไปใช้งานใน ขั้นตอนต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 โปรแกรม ARToolKit Marker Generator แสดงไฟล์ Marker ที่อัฟโหลดและ Preview Marker

4.3 การตัดต่อวิดีโอด้วยเทคนิค Blue/Green Screen

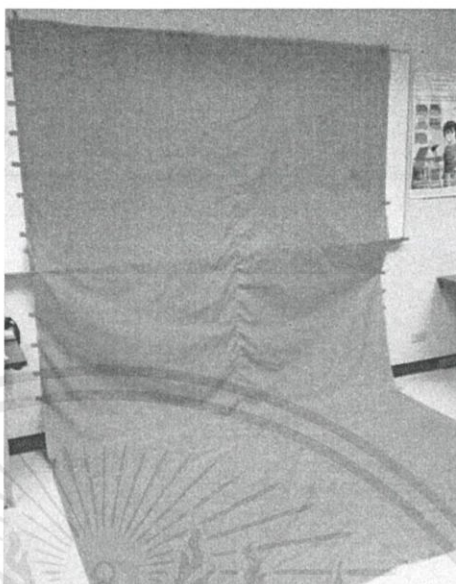
เทคนิค Blue/Green Screen นั้นใช้ในการตัดภาพวัตถุออกจากพื้นหลังหรือฉากหลัง เพื่อนำภาพวัตถุที่ได้ไปใส่เข้ากับพื้นหลังอื่น ๆ โดยในโครงงานนี้เราได้เลือกใช้โปรแกรม Motion 5 ในการทำวิดีโอด้วยเทคนิค Green Screen เพื่อนำวิดีโอที่ได้ไปใช้ในการ Render โดยมีขั้นตอนการทำดังนี้

4.3.1 การเตรียมวิดีโอสำหรับการตัดต่อ

วิดีโอที่จะใช้ในการตัดต่อสามารถหาได้จากเว็บไซต์ต่าง ๆ หรือสามารถถ่ายวิดีโอด้วยตนเองก็ได้ สำหรับการถ่ายทำวิดีโอด้วยตนเองนั้นสามารถทำได้ดังนี้

- 1) เตรียมฉากหลังสีเขียว โดยใช้ผ้าสีเขียวหรือกระดาษสีเขียว มาทำการติดตั้งบนผนังห้องและพื้นห้อง ดังรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

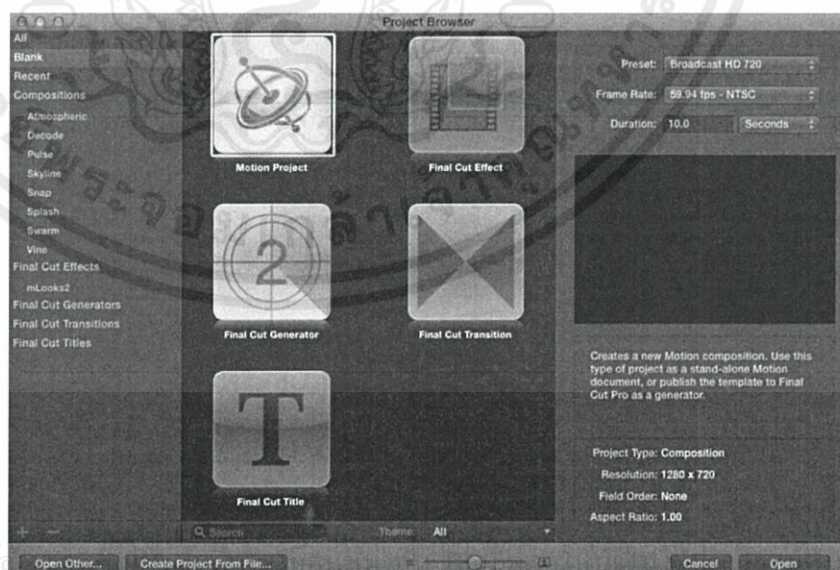


รูปที่ 4.7 การติดตั้งฉากสีเขียวสำหรับการถ่ายวิดีโอ

2) ทำการถ่ายวิดีโอตามที่ต้องการ

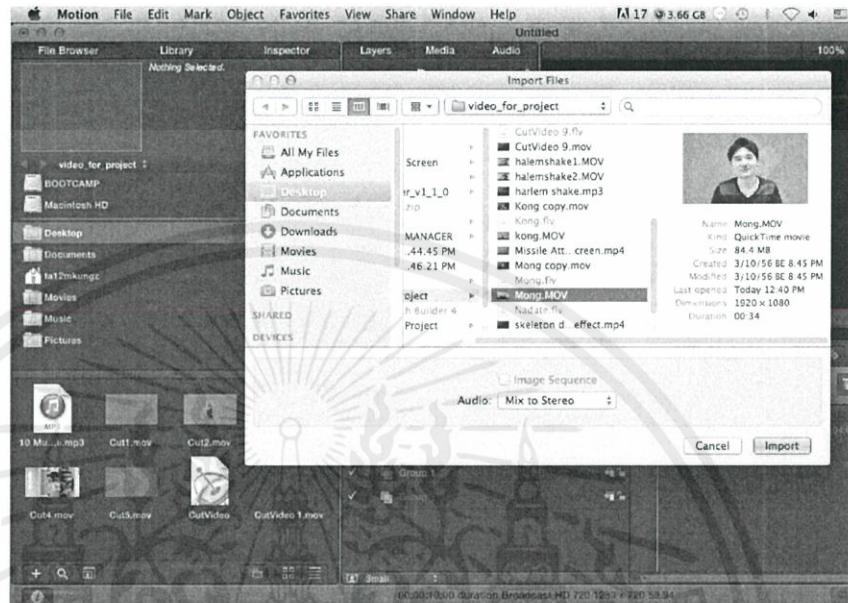
4.3.2 การตัดพื้นหลังหรือตัดฉากออกจากวัตถุด้วยโปรแกรม Motion 5

1) ทำการเปิดโปรแกรม Motion 5 ขึ้นมา เลือก Motion Project จากนั้นคลิกปุ่ม Open เพื่อสร้างงานใหม่



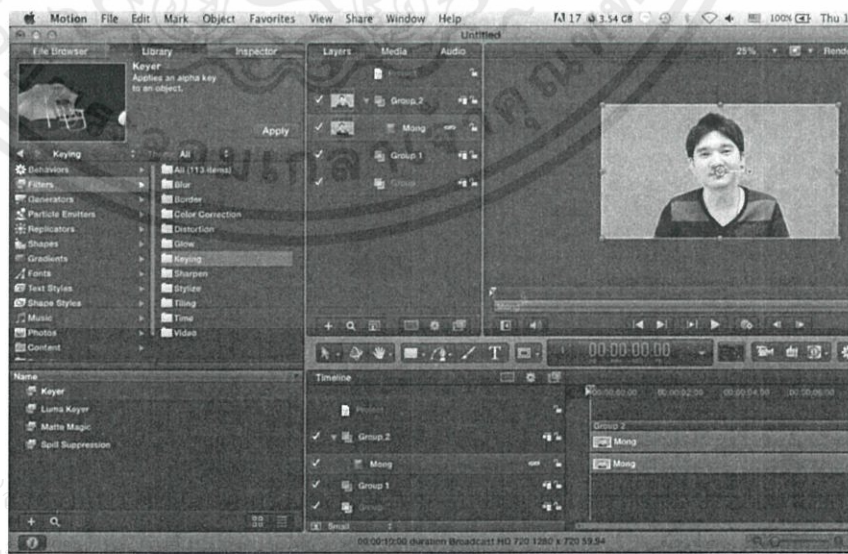
รูปที่ 4.8 การสร้างงานใหม่ในโปรแกรม Motion 5

- 2) เลือกไฟล์วิดีโอที่จะนำมาตัดพื้นหลังออก โดยเลือกแถบเมนู File -> Import จากนั้นทำการเลือกไฟล์วิดีโอ กดปุ่ม Import เพื่อนำวิดีโอชิ้นนั้นมาใช้งาน



รูปที่ 4.9 การ import ไฟล์วิดีโอมาใช้งาน

- 3) ทำการตัดพื้นหลังสีเขียวออก เริ่มต้นจากการคลิกเลือกวิดีโอที่ต้องการตัดพื้นหลังออก จากนั้นเลือกที่ Library -> Filters -> Keying -> Keyer กดปุ่ม Apply เพื่อยืนยันการตัดพื้นหลังสีเขียวออก



รูปที่ 4.10 การตัดพื้นหลังสีเขียวออกด้วยคำสั่ง Keying

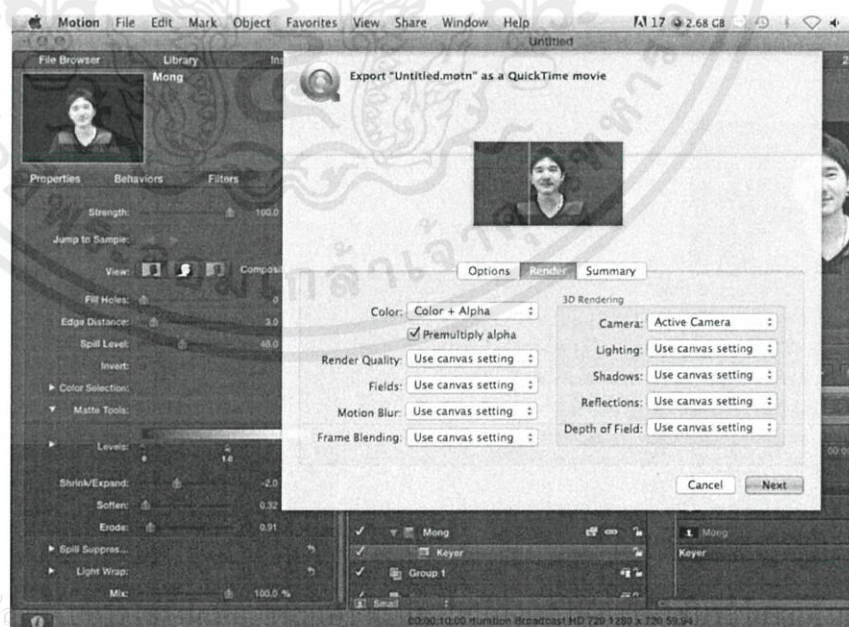
เอกสารนี้เป็นเอกสารที่สงวนไว้
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้าม

การคัด



รูปที่ 4.11 วิดีโอที่ตัดพื้นหลังออกเรียบร้อยแล้ว

- 4) บันทึกไฟล์วิดีโอ โดยเลือกแถบเมนู Share -> Export Movie... จากนั้นคลิก Render เลือกที่ Color เป็น Color + Alpha เพื่อให้วิดีโอสามารถแสดงพื้นหลังเป็นแบบโปร่งใสได้ กดปุ่ม Next เพื่อทำการบันทึกไฟล์วิดีโอ



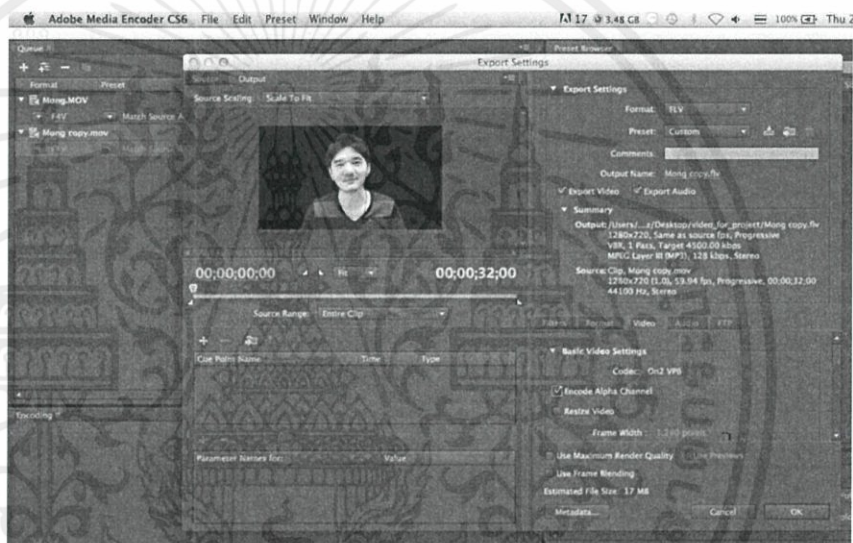
เอกสารนี้เป็นเอกสารที่สงวนไว้

การค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงหน่วยงานของกรมการรังที่มีการนำไปใช้

รูปที่ 4.12 การ Save วิดีโอเพื่อนำไฟล์ไปใช้งาน

เนื่องจากไฟล์วิดีโอที่สามารถนำไปทำการ Render ได้นั้นต้องมีไฟล์นามสกุล .FLV ดังนั้นเราจะใช้โปรแกรม Adobe Media Encoder ในการแปลงไฟล์ โดยเริ่มจากการเปิดโปรแกรม Adobe Media Encoder ขึ้นมา จากนั้นทำการ Import ไฟล์วิดีโอ โดยคลิกที่เครื่องหมาย + ที่มุมซ้ายบนของโปรแกรม เพื่อเลือกไฟล์วิดีโอที่จะนำมาทำการแปลง กดปุ่ม Open เพื่อเปิดไฟล์วิดีโอขึ้นมา ทำการตั้งค่าไฟล์วิดีโอที่ต้องการจะแปลงออกมา โดยเลือก Format เป็น FLV เพื่อแปลงไฟล์วิดีโอออกมาเป็นนามสกุล FLV และคลิกเลือก Encode Alpha Channel เพื่อให้วิดีโอสามารถแสดงพื้นหลังเป็นแบบโปร่งใสได้



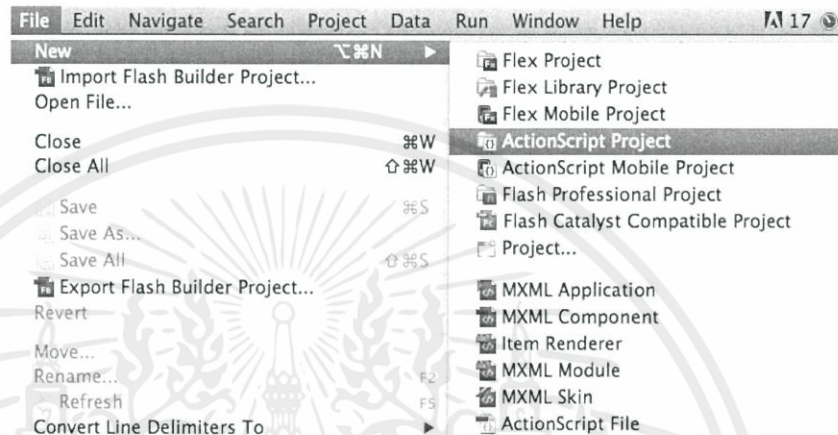
รูปที่ 4.13 การแปลงนามสกุลไฟล์วิดีโอจาก .MOV เป็น .FLV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การเขียนโปรแกรม

4.4.1 การสร้างงานใหม่และการนำไลบรารี FLARToolKit มาใช้งาน

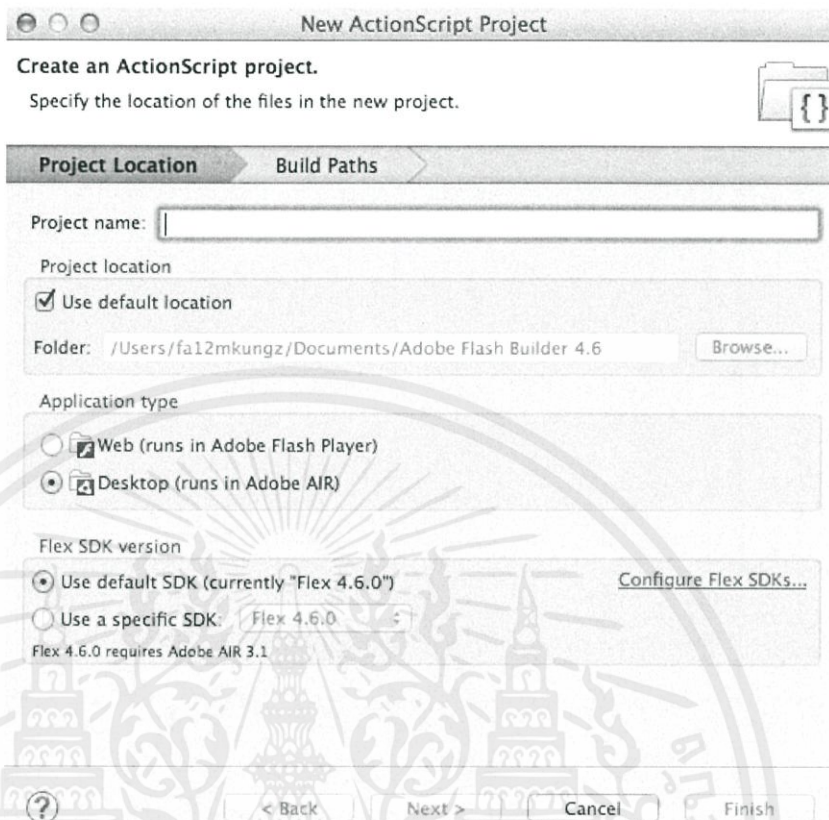
- 1) เปิดโปรแกรม Adobe Flash Builder ขึ้นมา หลังจากนั้นทำการสร้างงานใหม่ โดยเลือกที่ File -> New -> ActionScript Project



รูปที่ 4.14 แสดงการสร้างงานใหม่ (New ActionScript Project)

- 2) ทำการตั้งชื่อไฟล์งานใหม่ลงในช่อง Project Name การพัฒนาแอปพลิเคชันด้วยโปรแกรม Adobe Flash Builder นั้น เราสามารถเลือกได้ว่าจะพัฒนาแบบไหน โดยที่ Application Type จะมีให้เลือกด้วยกัน 2 แบบคือ Web (runs in Adobe Flash Player) และ Desktop (runs in Adobe Air) ซึ่งในโครงการนี้เราจะใช้การพัฒนาแบบ Desktop (runs in Adobe Air) เนื่องจากการพัฒนาแบบ Web (runs in Adobe Flash Player) นั้นยังมี ปัญหา คือสามารถรันได้แค่บางเบราว์เซอร์เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

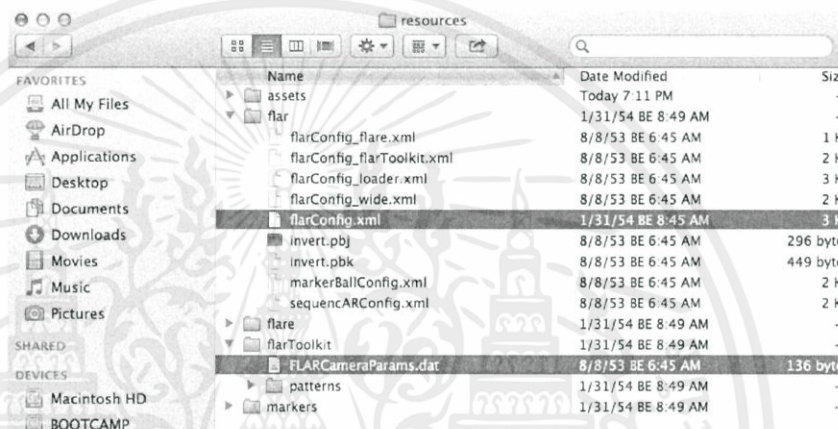
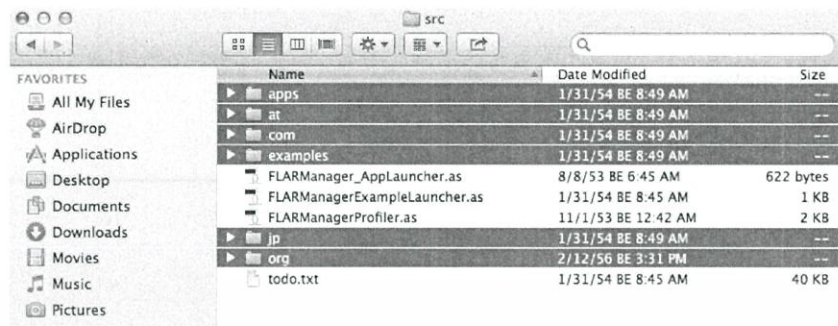


รูปที่ 4.15 การตั้งชื่องานใหม่และการเลือกรูปแบบการรันแอปพลิเคชัน

3) การนำไลบรารี FLARToolKit มาใช้งานนั้น เราจำเป็นต้องทำการคัดลอกไฟล์และโฟลเดอร์ที่จำเป็นมาใส่ในโฟลเดอร์ src ของงานที่เราสร้างขึ้น โดยมีไฟล์และโฟลเดอร์ที่สำคัญมีดังต่อไปนี้

- โฟลเดอร์ทั้งหมดใน FLARManager_v1_1_0/src
- ไฟล์เกี่ยวกับการตั้งค่าเพื่อรับภาพจากกล้องและการจัดการ Marker คือไฟล์ flarConfig.xml (อยู่ใน FLARManager_v1_1_0/resources/flar) และไฟล์ FLARCameraParams.dat (อยู่ใน FLARManager_v1_1_0/resources/flarToolkit)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 ไฟล์และโฟลเดอร์ที่ต้องทำการคัดลอกมาใส่ในโฟลเดอร์ src ของงานใหม่ที่สร้างขึ้น

4.4.2 การตั้งค่าสำหรับการติดตามและวิเคราะห์ Marker และการตั้งค่าเพื่อแสดงผลเฟรมบนจอภาพ

ในการกำหนดค่าต่าง ๆ เกี่ยวกับการติดตามและวิเคราะห์ Marker และการตั้งค่าเพื่อแสดงผลเฟรมบนหน้าจอ นั้น สามารถไปกำหนดค่าพารามิเตอร์ต่าง ๆ ในไฟล์ flarConfig.xml โดยจะมีค่า พารามิเตอร์หลัก ๆ ที่ต้องกำหนดดังนี้

1. sourceWidth และ sourceHeight เป็นค่าความกว้างและความสูงที่ใช้ในการจับภาพวิดีโอ โดยค่านี้อาจส่งผ่านไปยัง Camera.setMode () ใน FLARCameraSource.as
2. displayWidth และ displayHeight เป็นค่าความกว้างและความสูงที่ใช้ในการแสดงวิดีโอ
3. framerate คือค่าอัตราที่ภาพเคลื่อนไหวในเวลา 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังห้ามเผยแพร่ข้อมูลอันเป็นที่ยอมรับของสาธารณะแก่บุคคลอื่นที่มีการนำไปใช้

4. trackerToSourceRatio ภาพหรือวิดีโอที่ใช้สำหรับการติดตามจะถูกปรับขนาด โดยค่า trackerToSourceRatio ก่อนที่จะถูกส่งไปวิเคราะห์การติดตาม โดยไลบรารีสำหรับการติดตามนั้นจะทำงานได้เร็วขึ้น ถ้า downsampling มากขึ้น แต่จะทำให้การเรียนรู้จดจำ Pattern ยากยิ่งขึ้น โดยค่า 1.0 จะไม่มีการทำ downsampling ถ้ามีค่าเป็น 0.5 (ค่าเริ่มต้น) จะมีการทำ downsampling กับภาพหรือวิดีโอครึ่งหนึ่ง
5. activityThreshold เมื่อ camera.activityLevel น้อยกว่าค่า activityThreshold ภาพหรือวิดีโอที่รับเข้ามาจะไม่ถูกส่งไปยัง FLARToolkit และ Marker จะถูกตรึงไว้เมื่อมีการ เคลื่อนไหวน้อยมากและจะหยุดการติดตาม เพื่อเพิ่ม frame rates ค่าเริ่มต้นของ activityThreshold คือ 16 โดยค่าที่สามารถใช้ได้เริ่มตั้งแต่ 0-100 การตั้งค่าน้อย ๆ จะทำให้ Marker สามารถเคลื่อนไหวได้อิสระมากขึ้น
6. mirrorDisplay ถ้าค่านี้เป็น true วิดีโอและการตรวจจับ Marker จะถูกพลิกตามแนวนอน
7. smoothing ถ้าค่า smoothing มีค่ามาก จะทำให้วัตถุเสมือนที่ทำการ Render ดูราบเรียบขึ้น แต่จะตอบสนองช้าลงต่อการเปลี่ยนตำแหน่งของ Marker
8. pattern resolution เป็นค่าความละเอียดของ Pattern ที่นำมาใช้ในการเปรียบเทียบ
9. pattern path คือตำแหน่งของไฟล์ Pattern (ไฟล์นามสกุล .pat) ที่นำมาใช้ในการเปรียบเทียบ

4.4.3 Basic Augmented Reality

1. การรองรับคลาสที่ถูกเพิ่มเข้ามาใน stage

```
this.addEventListener(Event.ADDED_TO_STAGE, this.onAdded);
```

2. การใช้ไลบรารี FLARToolkit สำหรับการติดตาม Marker

```
var flarManager:FLARManager = new FLARManager  
("flarConfig.xml" , new FLARToolkitManager(), this.stage);
```

3. การเปิดกล้องเว็บแคมเพื่อใช้ในการจับภาพวิดีโอ

```
this.addChild(Sprite(this.flarManager.flarSource));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้นำข้อมูลไปเผยแพร่อย่างอื่นนอกเหนือจากเอกสารทุกครั้งที่มีการนำไปใช้

4. การตรวจจับความเปลี่ยนแปลงของ Marker เช่น การตรวจพบ Marker หรือการที่ Marker ถูกนำออกจากเฟรม

```
this.flarManager.addEventListener(FLARMarkerEvent.  
MARKER_ADDED, this.onMarkerAdded);  
  
this.flarManager.addEventListener(FLARMarkerEvent.  
MARKER_UPDATED, this.onMarkerUpdated);  
  
this.flarManager.addEventListener(FLARMarkerEvent.  
MARKER_REMOVED, this.onMarkerRemoved);
```

5. ขณะที่รอ FLARManager เริ่มต้น ก่อนการติดตั้งสภาพแวดล้อม Papervision3D เราจะเขียนตัวจัดการเหตุการณ์เล็กน้อย

```
this.flarManager.addEventListener(Event.INIT,  
this.onFlarManagerInited);
```

6. การติดตั้ง Papervision3D

```
private function onFlarManagerInited (evt:Event) :void {  
    this.flarManager.removeEventListener(Event.INIT,  
    this.onFlarManagerInited);  
  
    this.scene3D = new Scene3D();  
    this.viewport3D = new Viewport3D(this.stage.stageWidth,  
    this.stage.stageHeight);  
    this.addChild(this.viewport3D);
```

```
this.camera3D = new FLARCamera_PV3D(this.flarManager,  
    new Rectangle(0, 0, this.stage.stageWidth,  
    this.stage.stageHeight));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
this.pointLight3D = new PointLight3D();
```

```

this.pointLight3D.x = 1000;
this.pointLight3D.y = 1000;
this.pointLight3D.z = -1000;

this.addEventListener(Event.ENTER_FRAME,
this.onEnterFrame);

```

เนื่องจาก Papervision มีการ Render Scene ซ้ำในทุก ๆ เฟรม ดังนั้นเราต้อง
 เพิ่มตัวจัดการเหตุการณ์ ENTER_FRAME

7. การสร้างวัตถุเสมือนเพื่อใช้ในการ Render (ตัวอย่างจะเป็นการสร้างลูกบาศก์)

```

var cubeMaterial:FlatShadeMaterial = new FlatShadeMaterial
(this.pointLight3D, 0xFF1919, 0x730000);
var materialsList:MaterialsList = new MaterialsList
({all: cubeMaterial});
var cube:Cube = new Cube(materialsList, 40, 40, 40);
cube.z += 20;

// create a container for the cube, that will accept matrix
transformations.
this.cubeContainer = new DisplayObject3D();
this.cubeContainer.addChild(cube);
this.scene3D.addChild(this.cubeContainer);

```

8. การตอบสนองต่อ FLARMarkerEvents

```

private function onMarkerAdded (evt:FLARMarkerEvent) :void {
    this.cubeContainer.visible = true;
    this.activeMarker = evt.marker;
}
private function onMarkerRemoved (evt:FLARMarkerEvent)
:void {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

this.cubeContainer.visible = false;
this.activeMarker = null;
}

```

9. การอัปเดตลูกบาศก์ เนื่องจาก Papervision มีการ Render Scene ซ้ำในทุก ๆ เฟรม เราจะใช้ประโยชน์จากโอกาสนี้เพื่อแปลงเมทริกซ์ให้ใหม่ล่าสุด เพื่อจะทำให้ลูกบาศก์ดูสมจริงมากที่สุด

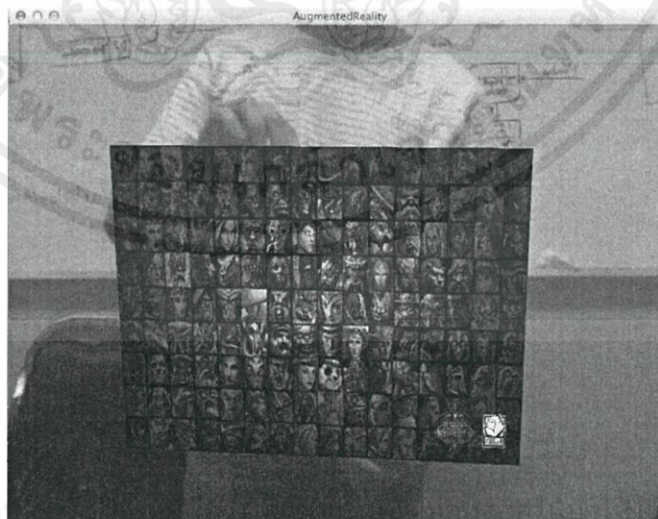
```

private function onEnterFrame (evt:Event) :void {
    this.cube.transform =
    PVGeomUtils.convertMatrixToPVMMatrix(this.activeMarker.
    transformMatrix, this.flarManager.flarSource.mirrored);
    this.renderEngine.render();
}

```

4.5 ตัวอย่างแอปพลิเคชันชนิด Augmented Reality ที่ทำการพัฒนา

- 1) แอปพลิเคชันที่ Render วัตถุเสมือนเป็นรูปภาพ
- 2) แอปพลิเคชันที่ Render วัตถุเสมือนเป็นวิดีโอ
- 3) แอปพลิเคชันที่ Render วัตถุเสมือนเป็นอนิเมชัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ

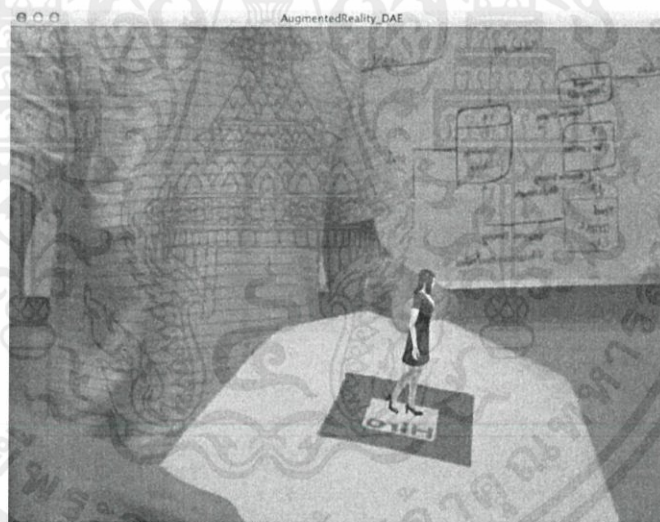
ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.17 ตัวอย่างแอปพลิเคชันที่ Render วัตถุเสมือนเป็นรูปภาพ



รูปที่ 4.18 ตัวอย่างแอปพลิเคชันที่ Render วัตถุเสมือนเป็นวิดีโอ



รูปที่ 4.19 ตัวอย่างแอปพลิเคชันที่ Render วัตถุเสมือนเป็นอนิเมชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

แอปพลิเคชันชนิด Augmented Reality ที่พัฒนาขึ้นมานั้น พัฒนาด้วยไลบรารีที่สำคัญคือ FLARToolKit และมีเฟรมเวิร์คคือ FLARManager ซึ่งในการเขียนโปรแกรมนั้น ทางคณะผู้จัดทำได้ใช้ภาษา ActionScript 3.0 ซึ่งเป็นภาษาที่ใช้งานกับไฟล์ Flash เป็นหลัก โดยผลที่ได้นั้นเป็นที่น่าพอใจเป็นอันมากเพราะโปรแกรมทำงานได้ถูกต้องและแม่นยำในการตรวจจับ Marker ทำการซ้อนทับภาพวัตถุเสมือน 3 มิติได้อย่างถูกต้องตามวัตถุประสงค์ของการพัฒนาที่ได้วางไว้ จากการพัฒนาโปรแกรมสามารถสรุปปัจจัยที่มีผลต่อการดำเนินโครงการได้ดังต่อไปนี้

- 1) กล้องเว็บแคม (Webcam) ที่นำมาใช้งานควรมีคุณสมบัติที่ดี เช่น ความละเอียดของภาพ อัตราการถ่ายภาพ (Frame Rate) และสามารถปรับความสว่างของแสงและสีได้
- 2) อัตราความสว่างของแสงในห้องที่ใช้ในการถ่ายภาพ โดยห้องที่ใช้ในการถ่ายภาพควรมีการควบคุมแสงให้คงที่ และมีความสว่างเพียงพอ ถ้าแสงสว่างมากเกินไปกล้องจะทำการ Detect Marker ได้ไม่ชัดเจนซึ่งอาจทำให้โปรแกรมได้ผลที่ไม่ดีพอ

5.2 ปัญหาและอุปสรรค

- 1) ปัญหาที่ผู้พัฒนาโครงการไม่มีความรู้พื้นฐานในการใช้ฟังก์ชัน (Function) ต่าง ๆ ของไลบรารี FLARToolKit จึงต้องใช้เวลาในการศึกษาและวิธีการใช้ซึ่งทำให้การทำงานนี้ใช้เวลาค่อนข้างนาน
- 2) ปัญหาเกี่ยวกับกล้องที่นำมาใช้ประมวลผล เนื่องจากการประมวลผลจากภาพที่ได้มาจากกล้องเว็บแคม (Webcam) นั้นมีหลายปัจจัยที่ก่อให้เกิดความผิดพลาดขึ้นได้ เช่น แสงหรือสิ่งรบกวนต่าง ๆ ที่ทำให้เกิดปัญหาในการประมวลผลภาพ เป็นต้น
- 3) ปัญหาเกี่ยวกับซอฟต์แวร์ไลบรารีที่เลือกใช้ในการพัฒนาโครงการนี้ เนื่องจากโครงการนี้จะต้องทำการติดตั้งไลบรารีที่ใช้ จึงจำเป็นที่จะต้องทำการติดตั้งให้สมบูรณ์และครบถ้วนถูกต้องทุกขั้นตอน ซึ่งหากผิดพลาดบางขั้นตอนแล้วอาจจะทำให้เสียเวลาในการทำการแก้ไขข้อผิดพลาดต่าง ๆ ให้ใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดูเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มาเป็นใช้
ARToolKit นั้นมีปัญหาในการนำซอฟต์แวร์ไลบรารีทั้งคู่มารวมกัน ซึ่งปัญหาดังกล่าว

ทางคณะผู้จัดทำได้ใช้เวลาในการค้นคว้าข้อมูลอยู่นานจนทำให้เสียเวลาไปค่อนข้างมาก ทำให้คณะผู้จัดทำต้องทำการหาซอฟต์แวร์ไลบรารีใหม่เพื่อให้ได้งานที่มีคุณภาพมากที่สุด ในระยะเวลาที่มีอยู่อย่างจำกัด

5.3 แนวทางการแก้ไข

- 1) เนื่องจากมีระยะเวลาในการพัฒนาโปรแกรมจำกัด ทำให้ต้องศึกษาไปพร้อม ๆ กับการพัฒนาจริง
- 2) พยายามปรับสภาพแวดล้อมให้เหมาะสมที่สุดในการใช้งานตัวโปรแกรมเทคโนโลยีเสมือนจริง (Augmented Reality)
- 3) ควรใช้เครื่องคอมพิวเตอร์ที่มีประสิทธิภาพเพียงพอในการทำโครงการ ซึ่งสามารถรองรับการทำงานด้านต่าง ๆ ได้อย่างเหมาะสม
- 4) เนื่องจากมีปัญหาในการใช้ไลบรารีของ OpenCV และ ARToolKit ในการใช้งานร่วมกัน จึงทำให้คณะผู้จัดทำต้องทำการเปลี่ยนไลบรารีใหม่โดยใช้เป็น FLARToolKit เพื่อให้โครงการเสร็จทันตามเวลาที่กำหนด
- 5) ศึกษาการใช้งานไลบรารี FLARToolKit ในระยะเวลาที่จำกัดให้เร็วที่สุดเพื่อพัฒนาโปรแกรมเพื่อชดเชยกับเวลาที่สูญเสียไปของการใช้ไลบรารี OpenCV และ ARToolKit

5.4 แนวทางในการพัฒนาต่อ

จากโครงการเรื่องเทคโนโลยีเสมือนจริง (Augmented Reality) สามารถนำไปประยุกต์ใช้กับด้านต่าง ๆ ได้มากมาย เช่น สื่อโฆษณา เกมส์ สถาปัตยกรรม และอื่น ๆ เป็นต้น ซึ่งในปัจจุบันได้มีการนำเทคโนโลยีเหล่านี้มาใช้กันอย่างแพร่หลายมากขึ้น และในอนาคตอันใกล้นี้อาจจะเห็นการใช้งานเทคโนโลยีนี้กันอย่างแพร่หลายอย่างแน่นอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] ผศ.ดร. อรรถจักร จิตตโสภักดิ์, **Digital Image Processing**. กรุงเทพฯ : บริษัท สงวนกิจ พรินท์ แอนด์ มีเดีย, มีนาคม 2552
- [2] Gary Bradski, Adrian Kaehler. **Learning OpenCV**. the United States of America : O'Reilly Media, Inc., 2008
- [3] Robert Laganier. **OpenCV2 Computer Vision Application Programming Cookbook**. Birmingham, B27 6PA, UK : Packt Publishing Ltd., May 2011
- [4] ณัฐวิทย์ ชูกิจรุ่งโรจน์, ตรีภัทร โภคทรัพย์, เอกภาพ ศศิบุตร. “Augmented Reality.” **ปริญญา นิตยสารวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมอิเล็กทรอนิกส์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**. 2553.
- [5] สิ้นจัย นุ่มทิม, สุชาติ เป็รื่องคำ, อาภาภรณ์ หมัดสะและ. “สื่อสามมิติเสมือนจริง” **ปริญญา นิตยสารวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**. 2553.
- [6] MUKMIK. “เทคโนโลยี Augmented Reality.” [Online]. Available : <http://mukmik.wordpress.com/2011/02/09/%E0%B9%80%E0%B8%97%E0%B8%84%E0%B9%82%E0%B8%99%E0%B9%82%E0%B8%A5%E0%B8%A2%E0%B8%B5-augmented-reality/>. 2011
- [7] ARTToolKit [Online]. Available : <http://www.hitl.washington.edu/ARTToolKit>, February 2010
- [8] OpenCV [Online]. Available : <http://docs.opencv.org/>. 2012
- [9] FLARTToolKit [Online]. Available : <http://www.libspark.org/wiki/saqoosha/FLARTToolKit/en>
- [10] FLARManager [Online]. Available : <http://words.transmote.com/wp/flarmanager/>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การดาวน์โหลดโปรแกรม Adobe Flash Builder

ดาวน์โหลดโปรแกรม Adobe Flash Builder ได้ที่

https://www.adobe.com/cfusion/tdrc/index.cfm?product=flash_builder

2. การดาวน์โหลด 3D Engine Papervision3D

- ดาวน์โหลดไฟล์ Papervision3D ได้ที่

<http://code.google.com/p/papervision3d/downloads/list>

- ทำการคัดลอกโฟลเดอร์ ascollada และ papervision3d ในโฟลเดอร์ org ที่ดาวน์โหลดมาไปใส่ไว้ใน src/org ของงานที่สร้างขึ้น

3. การดาวน์โหลดไลบรารี FLARManager

- ดาวน์โหลดไลบรารี FLARManager ได้ที่

<http://words.transmote.com/wp/flarmanager/>

- ทำการ Extract ไฟล์และนำไปใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง Source Code ของแอปพลิเคชันที่ Render วัตถุเสมือนเป็นรูปภาพ แบบ Multi-Marker

```

package
{
    import com.transmote.flar.FLARManager;
    import com.transmote.flar.camera.FLARCamera_Flash3D;
    import com.transmote.flar.marker.FLARMarker;
    import com.transmote.flar.marker.FLARMarkerEvent;
    import com.transmote.flar.tracker.FLARToolkitManager;
    import com.transmote.utils.time.FramerateDisplay;

    import examples.support.MarkerPlane;

    import flash.display.Scene;
    import flash.display.Shape;
    import flash.display.Sprite;
    import flash.events.ErrorEvent;
    import flash.events.Event;
    import flash.geom.Rectangle;

    [SWF(width = '800',height = '600', backgroundColor = '#000000', frameRate = '30')]

    public class AugmentedReality extends Sprite
    {
        private var flarManager:FLARManager;
        private var markerPlane:MarkerPlane;

        private var scene:Sprite;
        private var camera:FLARCamera_Flash3D;

        public function AugmentedReality()
        {
            this.addEventListener(Event.ADDED_TO_STAGE, this.onAdded);
        }
    }

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

this.addEventListener(Event.ADDED_TO_STAGE, this.onAdded);

```

```

}
private function onAdded (evt:Event):void
{
    this.removeListener(Event.ADDED_TO_STAGE, this.onAdded);

    // pass the path to the FLARManager xml config file into the
    FLARManager constructor.

    // FLARManager creates and uses a FLARCameraSource by default.
    // the image from the first detected camera will be used for marker
    detection.

    // also pass an IFLARTrackerManager instance to communicate with a
    tracking library,
    // and a reference to the Stage (required by some trackers).
    this.flarManager = new FLARManager("flarConfig.xml", new
    FLARToolkitManager(), this.stage);

    this.flarManager.addEventListener(ErrorEvent.ERROR,
    this.onFlarManagerError);

    // add FLARManager.flarSource to the display list to display the video
    capture.
    this.addChild(Sprite(this.flarManager.flarSource));

    // begin listening for FLARMarkerEvents.
    this.flarManager.addEventListener(FLARMarkerEvent.MARKER_ADDED,
    this.onMarkerAdded);
    this.flarManager.addEventListener(FLARMarkerEvent.MARKER_REMOVED,
    this.onMarkerRemoved);
    this.flarManager.addEventListener(FLARMarkerEvent.MARKER_UPDATED,
    this.onMarkerUpdated);

    // framerate display helps to keep an eye on performance.
    var framerateDisplay:FramerateDisplay = new FramerateDisplay();
    this.addChild(framerateDisplay);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาร่วมกัน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่สู่สาธารณะซึ่งจะผิดกฏลิขสิทธิ์เจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        this.flarManager.addEventListener(Event.INIT, this.onFlarManagerInited);
    }
    private function onFlarManagerError (evt:ErrorEvent) :void
    {
        this.flarManager.removeEventListener(ErrorEvent.ERROR,
this.onFlarManagerError);
        this.flarManager.removeEventListener(Event.INIT,
this.onFlarManagerInited);

        trace(evt.text);
    }
    private function onFlarManagerInited (evt:Event) :void
    {
        this.scene = new Sprite();
        this.addChild(this.scene);

        var axes:Sprite = new Sprite();
        this.addChild(axes);

        var zAxis:Shape = new Shape();
        zAxis.rotationX = -90;
        axes.addChild(zAxis);
    }
    private function onMarkerAdded (evt:FLARMarkerEvent):void
    {
        trace("[ "+evt.marker.patternId+" ] added");

        switch(evt.marker.patternId){
        case 0:
        {
            // load an image to display on top of the marker.
            // note, this could be any content loaded with a Loader, still or
            animated.
            this.markerPlane = new MarkerPlane("rockman.jpg");
            this.markerPlane.visible = false;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงหรือเผยแพร่ข้อมูลอันเป็นความลับของทางบริษัทฯ ไปใช้

```

// rotate an image in z-axis
this.markerPlane.rotationY = 180;
this.markerPlane.rotationZ = 90;
this.scene.addChild(this.markerPlane);

this.camera = new FLARCamera_Flash3D(this.flarManager, new
Rectangle(0, 0, this.stage.stageWidth, this.stage.stageHeight));
this.camera.scene = this.scene;
}

case 1:
{
// load an image to display on top of the marker.
// note, this could be any content loaded with a Loader, still or
animated.
this.markerPlane2 = new MarkerPlane("HoN.jpg");
this.markerPlane2.visible = false;

// rotate an image in z-axis
this.markerPlane2.rotationY = 180;
this.markerPlane2.rotationZ = 90;
this.scene.addChild(this.markerPlane2);

this.camera = new FLARCamera_Flash3D(this.flarManager, new
Rectangle(0, 0, this.stage.stageWidth, this.stage.stageHeight));
this.camera.scene = this.scene;
}
}

this.onMarkerUpdated(evt);
}

private function onMarkerRemoved (evt:FLARMarkerEvent):void
{
trace("[ "+evt.marker.patternId+" ] removed");
if(this.markerPlane)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่เอกสารนี้เป็นอันขาดเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        this.markerPlane.visible = false;
    }
}
private function onMarkerUpdated (evt:FLARMarkerEvent):void
{
    trace("[ "+evt.marker.patternId+" ] updated");
    switch(evt.marker.patternId){
        case 0:
        {
            this.markerPlane.visible = true;
            this.camera.applyTransform(evt.marker.transformMatrix);
            break;
        }
        case 1:
        {
            this.markerPlane2.visible = true;
            this.camera.applyTransform(evt.marker.transformMatrix);
            break;
        }
    }
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง Source Code ของแอปพลิเคชันที่ Render วัตถุเสมือนเป็นวิดีโอ

```

package
{
    import com.transmote.flar.FLARManager;
    import com.transmote.flar.camera.FLARCamera_PV3D;
    import com.transmote.flar.marker.FLARMarker;
    import com.transmote.flar.marker.FLARMarkerEvent;
    import com.transmote.flar.tracker.FLARToolkitManager;
    import com.transmote.flar.utils.geom.PVGeomUtils;
    import com.transmote.utils.time.FramerateDisplay;

    import flash.display.*;
    import flash.events.*;
    import flash.geom.Rectangle;
    import flash.media.Video;
    import flash.net.NetConnection;
    import flash.net.NetStream;
    import flash.text.*;
    import flash.utils.Timer;

    import org.papervision3d.materials.MovieMaterial;
    import org.papervision3d.materials.VideoStreamMaterial;
    import org.papervision3d.objects.DisplayObject3D;
    import org.papervision3d.objects.primitives.Plane;
    import org.papervision3d.render.LazyRenderEngine;
    import org.papervision3d.scenes.Scene3D;
    import org.papervision3d.view.Viewport3D;

```

```
[SWF(width = '800',height = '600', backgroundColor = '#000000', frameRate = '30')]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น

```
public class NewsPaper extends Sprite
{
    องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

```

private var flarManager:FLARManager;
private var framerateDisplay:FramerateDisplay;
private var scene3D:Scene3D;
private var viewport3D:Viewport3D;
private var camera3D:FLARCamera_PV3D;
private var lr:LazyRenderEngine;
private var plane3D:Plane;
private var container:DisplayObject3D;
private var marker:FLARMarker;
private var material:MovieMaterial;
protected var video:Video;
protected var nc:NetConnection;
protected var ns:NetStream;

public function NewsPaper()
{
    this.addEventListener(Event.ADDED_TO_STAGE, this.onAdded);
}

private function onAdded (evt:Event):void
{
    this.removeEventListener(Event.ADDED_TO_STAGE, this.onAdded);

    // Initial FLARManager
    this.flarManager = new FLARManager("flarConfig.xml", new
FLARToolkitManager(), this.stage);
    this.addChild(Sprite(this.flarManager.flarSource));

    this.flarManager.addEventListener(ErrorEvent.ERROR,
this.onFlarManagerError);

    // begin listening for FLARMarkerEvents.
    this.flarManager.addEventListener(FLARMarkerEvent.MARKER_ADDED,
this.onMarkerAdded);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไปอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ท่านใดนำเอกสารฉบับนี้ไปเผยแพร่หรือใช้

```

        this.flarManager.addEventListener(FLARMarkerEvent.MARKER_REMOVED,
this.onMarkerRemoved);

        //this.flarManager.addEventListener(FLARMarkerEvent.MARKER_UPDATED,
this.onMarkerUpdated);

        // framerate display helps to keep an eye on performance.
        this.framerateDisplay = new FramerateDisplay();
        this.addChild(framerateDisplay);

        this.flarManager.addEventListener(Event.INIT, this.onFlarManagerInited);
    }

    private function onFlarManagerInited (evt:Event) :void
    {
        this.removeEventListener(Event.INIT, this.onFlarManagerInited);

        this.scene3D = new Scene3D();
        this.camera3D = new FLARCamera_PV3D(this.flarManager, new
Rectangle(0, 0, this.stage.stageWidth, this.stage.stageHeight));
        this.camera3D.z = -30;
        this.viewport3D = new Viewport3D(800, 600, true);

        lr = new LazyRenderEngine(this.scene3D, this.camera3D,
this.viewport3D);

        /*-----
        ----- Video Render -----
        -----*/

        this.nc = new NetConnection();
        this.nc.connect(null);

        this.ns = new NetStream(this.nc);
        this.ns.client = {};

        this.ns.play("Nadate.flv");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

this.ns.pause();
this.video = new Video(1080, 608);
this.video.smoothing = true;
this.video.attachNetStream(this.ns);

this.material = new VideoStreamMaterial(this.video, this.ns, true, true);

this.material.smooth = true;
this.material.doubleSided = true;
this.plane3D = new Plane(this.material, 340, 255, 5, 5);
this.plane3D.scaleY = -1;
this.plane3D.rotationZ = -90;
//this.plane3D.rotationX = 90;
this.plane3D.visible = false;

this.container = new DisplayObject3D();
this.container.addChild(this.plane3D);
//-----
//---- (END) Video Render ----
//-----

this.scene3D.addChild(this.container);
this.addChild(this.viewport3D);

this.addEventListener(Event.ENTER_FRAME, this.onEnterFrame);
}

private function onEnterFrame (evt:Event):void
{
    if(this.marker != null)
    {
        this.container.transform =
        PVGeomUtils.convertMatrixToPVMMatrix(marker.transformMatrix);
    }

    this.lre.render();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    private function onFlarManagerError (evt:ErrorEvent):void
    {
        this.flarManager.removeEventListener(ErrorEvent.ERROR,
this.onFlarManagerError);
        this.flarManager.removeEventListener(Event.INIT,
this.onFlarManagerInited);

        trace(evt.text);
    }

    private function onMarkerAdded (evt:FLARMarkerEvent):void
    {
        trace("[ "+evt.marker.patternId+" ] added");
        this.marker = evt.marker;
        stage.addEventListener( KeyboardEvent.KEY_DOWN, keyDownHandler);
        //this.ns.resume();
        this.plane3D.visible = true;
        //this.ns.play();
    }

    private function onMarkerRemoved (evt:FLARMarkerEvent):void
    {
        trace("[ "+evt.marker.patternId+" ] removed");
        this.marker = null;
        //stage.addEventListener( KeyboardEvent.KEY_DOWN, keyDownHandler);
        this.ns.pause();
        this.plane3D.visible = false;
        //this.video.vid.stop();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการเรียนเท่านั้น ไปขอเทคโนโลยีไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//trace("Key Value" + evt.keyCode);
```

```
if(evt.keyCode == 13)
{
    this.ns.togglePause();
}
}
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง Source Code ของแอปพลิเคชันที่ Render วัตถุเสมือนเป็นอนิเมชัน แบบ Multi-Marker

```

package
{
    import com.transmote.flar.FLARManager;
    import com.transmote.flar.camera.FLARCAMERA_PV3D;
    import com.transmote.flar.marker.FLARMarker;
    import com.transmote.flar.marker.FLARMarkerEvent;
    import com.transmote.flar.tracker.FLARToolkitManager;
    import com.transmote.flar.utils.geom.PVGeomUtils;

    import flash.display.Sprite;
    import flash.events.Event;
    import flash.geom.Rectangle;

    import org.papervision3d.cameras.Camera3D;
    import org.papervision3d.lights.PointLight3D;
    import org.papervision3d.materials.BitmapFileMaterial;
    import org.papervision3d.materials.BitmapMaterial;
    import org.papervision3d.materials.utils.MaterialsList;
    import org.papervision3d.objects.DisplayObject3D;
    import org.papervision3d.objects.parsers.DAE;
    import org.papervision3d.render.LazyRenderEngine;
    import org.papervision3d.scenes.Scene3D;
    import org.papervision3d.view.Viewport3D;

    [SWF(width = '800',height = '600', backgroundColor = '#000000', frameRate = '30')]

    public class AugmentedReality_GUI extends Sprite
    {
        private var flarManager:FLARManager;
        private var scene3D:Scene3D;
    }

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private var camera3D:Camera3D;
private var viewport3D:Viewport3D;
private var renderEngine:LazyRenderEngine;
private var pointLight3D:PointLight3D;

private var model0:DAE;
private var activeMarker0:FLARMarker;
private var modelContainer0:DisplayObject3D;
private var bitmapFileMaterial01:BitmapFileMaterial;
private var bitmapFileMaterial02:BitmapFileMaterial;
private var bitmapFileMaterial03:BitmapFileMaterial;
private var materialsList0:MaterialsList;

private var model1:DAE;
private var activeMarker1:FLARMarker;
private var modelContainer1:DisplayObject3D;
private var bitmapFileMaterial11:BitmapFileMaterial;
private var bitmapFileMaterial12:BitmapFileMaterial;
private var bitmapFileMaterial13:BitmapFileMaterial;
private var bitmapFileMaterial14:BitmapFileMaterial;
private var bitmapFileMaterial15:BitmapFileMaterial;
private var bitmapFileMaterial16:BitmapFileMaterial;
private var bitmapFileMaterial17:BitmapFileMaterial;
private var bitmapFileMaterial18:BitmapFileMaterial;
private var materialsList1:MaterialsList;

private var model2:DAE;
private var activeMarker2:FLARMarker;
private var modelContainer2:DisplayObject3D;

private var markerId:int;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษามาก่อน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

this.addEventListener(Event.ADDED_TO_STAGE, this.onAdded);

```

```

}

private function onAdded (evt:Event) :void {
    this.removeListener(Event.ADDED_TO_STAGE, this.onAdded);

    // pass the path to the FLARManager xml config file into the
    FLARManager constructor.

    // FLARManager creates and uses a FLARCameraSource by default.
    // the image from the first detected camera will be used for marker
    detection.

    // also pass an IFLARTrackerManager instance to communicate with a
    tracking library,
    // and a reference to the Stage (required by some trackers).
    this.flarManager = new FLARManager("flarConfig.xml", new
    FLARToolkitManager(), this.stage);

    // to switch tracking engines, pass a different IFLARTrackerManager into
    FLARManager.
    // refer to this page for information on using different tracking engines:
    // http://words.transmote.com/wp/inside-flarmanager-tracking-engines/
    //
    //
    this.flarManager = new
    FLARManager("../resources/flar/flarConfig.xml", new FlareManager(), this.stage);
    //
    this.flarManager = new
    FLARManager("../resources/flar/flarConfig.xml", new FlareNFTManager(), this.stage);

    // add FLARManager.flarSource to the display list to display the video
    capture.

    this.addChild(Sprite(this.flarManager.flarSource));

    // begin listening for FLARMarkerEvents.
    this.flarManager.addEventListener(FLARMarkerEvent.MARKER_ADDED,
    this.onMarkerAdded);
    this.flarManager.addEventListener(FLARMarkerEvent.MARKER_UPDATED,
    this.onMarkerUpdated);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆก็ตาม กรุณาแจ้งให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        this.flarManager.addEventListener(FLARMarkerEvent.MARKER_REMOVED,
this.onMarkerRemoved);

        // wait for FLARManager to initialize before setting up Papervision3D
environment.

        this.flarManager.addEventListener(Event.INIT, this.onFlarManagerInited);
    }

```

```

        private function onFlarManagerInited (evt:Event) :void {
            this.flarManager.removeEventListener(Event.INIT,
this.onFlarManagerInited);

            this.scene3D = new Scene3D();
            this.viewport3D = new Viewport3D(this.stage.stageWidth,
this.stage.stageHeight);
            this.addChild(this.viewport3D);

            this.camera3D = new FLARCamera_PV3D(this.flarManager, new
Rectangle(0, 0, this.stage.stageWidth, this.stage.stageHeight));

            this.renderEngine = new LazyRenderEngine(this.scene3D, this.camera3D,
this.viewport3D);

            this.pointLight3D = new PointLight3D();
            this.pointLight3D.x = 1000;
            this.pointLight3D.y = 1000;
            this.pointLight3D.z = -1000;

            //----- Setting Model 0 -----
            this.bitmapFileMaterial01 = new BitmapFileMaterial("_01_-
_Default1noCulling.JPG");
            this.bitmapFileMaterial02 = new BitmapFileMaterial("_02_-
_Default1noCulling.JPG");
            this.bitmapFileMaterial03 = new BitmapFileMaterial("_03_-
_Default1noCulling.JPG");

```

```

this.materialsList0 = new MaterialsList();
this.materialsList0.addMaterial(this.bitmapFileMaterial01);
this.materialsList0.addMaterial(this.bitmapFileMaterial02);
this.materialsList0.addMaterial(this.bitmapFileMaterial03);

// load the model 0. (this model has to be scaled and rotated to fit the
marker; every model is different.)
this.model0 = new DAE(true, "model0", true);
model0.load("Citizen.dae", this.materialsList0);

model0.rotationX = 90;
model0.rotationZ = -90;
model0.scale = 0.5;

//---- Setting Model 2 ----
this.bitmapFileMaterial11 = new
BitmapFileMaterial("__Paper_noCulling.jpg");
this.bitmapFileMaterial12 = new
BitmapFileMaterial("__Plant_Cover_Kudzu_noCulling.jpg");
this.bitmapFileMaterial13 = new BitmapFileMaterial("Burj-TennisH.png");
this.bitmapFileMaterial14 = new
BitmapFileMaterial("Burj3_1noCulling.png");
this.bitmapFileMaterial15 = new BitmapFileMaterial("Burj3.png");
this.bitmapFileMaterial16 = new
BitmapFileMaterial("fenster1noCulling.png");
this.bitmapFileMaterial17 = new BitmapFileMaterial("fenster2_1.png");
this.bitmapFileMaterial18 = new
BitmapFileMaterial("fenster2noCulling.png");

this.materialsList1 = new MaterialsList();
this.materialsList1.addMaterial(this.bitmapFileMaterial11);
this.materialsList1.addMaterial(this.bitmapFileMaterial12);
this.materialsList1.addMaterial(this.bitmapFileMaterial13);

```

```

this.materialsList1.addMaterial(this.bitmapFileMaterial14);
this.materialsList1.addMaterial(this.bitmapFileMaterial15);
this.materialsList1.addMaterial(this.bitmapFileMaterial16);
this.materialsList1.addMaterial(this.bitmapFileMaterial17);
this.materialsList1.addMaterial(this.bitmapFileMaterial18);

// load the model 1. (this model has to be scaled and rotated to fit the
marker; every model is different.)
this.model1 = new DAE(true, "model1", true);
model1.load("Hotel.dae", this.materialsList1);
model1.rotationX = 90;
model1.rotationZ = -90;
model1.scale = 0.06;

// load the model 2. (this model has to be scaled and rotated to fit the
marker; every model is different.)
this.model2 = new DAE(true, "model2", true);
model2.load("EBVEYRON.dae");
model2.rotationX = 90;
model2.rotationZ = -90;
model2.scale = 8;

// create a container for the model, that will accept matrix
transformations.
this.modelContainer0 = new DisplayObject3D();
this.modelContainer0.addChild(model0);
this.modelContainer0.visible = false;
this.scene3D.addChild(this.modelContainer0);

this.modelContainer1 = new DisplayObject3D();
this.modelContainer1.addChild(model1);
this.modelContainer1.visible = false;
this.scene3D.addChild(this.modelContainer1);

this.modelContainer2 = new DisplayObject3D();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาร่วมกัน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

this.modelContainer2.addChild(model2);
this.modelContainer2.visible = false;
this.scene3D.addChild(this.modelContainer2);

this.addEventListener(Event.ENTER_FRAME, this.onEnterFrame);
}

```

```
private function onMarkerAdded (evt:FLARMarkerEvent) :void
```

```

{
    if(evt.marker.patternId == 0)
    {
        trace("[ "+evt.marker.patternId+" ] added");
        markerAdded(0);
        this.activeMarker0 = evt.marker;
    }
    if(evt.marker.patternId == 1)
    {
        trace("[ "+evt.marker.patternId+" ] added");
        markerAdded(1);
        this.activeMarker1 = evt.marker;
    }
    if(evt.marker.patternId == 2)
    {
        trace("[ "+evt.marker.patternId+" ] added");
        markerAdded(2);
        this.activeMarker2 = evt.marker;
    }
}
}

```

```
private function onMarkerUpdated (evt:FLARMarkerEvent) :void
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้อัปโหลดเอกสารฉบับนี้ขึ้นสู่เว็บไซต์ของเจ้าคุณธนพาทรรลาดกระบัง

```

    if(evt.marker.patternId == 0)
    {

```

```

        trace("[ "+evt.marker.patternId+" ] updated");
        markerAdded(0);
        this.activeMarker0 = evt.marker;
    }

```

```

    if(evt.marker.patternId == 1)
    {
        trace("[ "+evt.marker.patternId+" ] updated");
        markerAdded(1);
        this.activeMarker1 = evt.marker;
    }

```

```

    if(evt.marker.patternId == 2)
    {
        trace("[ "+evt.marker.patternId+" ] updated");
        markerAdded(2);
        this.activeMarker2 = evt.marker;
    }
}

```

```

private function onMarkerRemoved (evt:FLARMarkerEvent) :void
{
    if(evt.marker.patternId == 0)
    {
        trace("[ "+evt.marker.patternId+" ] removed");
        markerRemoved(0);
        this.activeMarker0 = evt.marker;
    }
}

```

```

    if(evt.marker.patternId == 1)
    {
        trace("[ "+evt.marker.patternId+" ] removed");
        markerRemoved(1);
        this.activeMarker1 = evt.marker;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือเผยแพร่ข้อมูลข้างต้นไปยังผู้อื่น เอกสารทุกครั้งที่มีการนำไปใช้

```

if(evt.marker.patternId == 2)
{
    trace("[ "+evt.marker.patternId+" ] removed");
    markerRemoved(2);
    this.activeMarker2 = evt.marker;
}

this.activeMarker0 = null;
this.activeMarker1 = null;
this.activeMarker2 = null;
}

private function onEnterFrame (evt:Event) :void {
    // apply the FLARToolkit transformation matrix to the Cube.
    if (this.activeMarker0)
    {
        this.modelContainer0.transform =
PVGeomUtils.convertMatrixToPVMatrix(this.activeMarker0.transformMatrix);
    }

    if (this.activeMarker1)
    {
        this.modelContainer1.transform =
PVGeomUtils.convertMatrixToPVMatrix(this.activeMarker1.transformMatrix);
    }

    if (this.activeMarker2)
    {
        this.modelContainer2.transform =
PVGeomUtils.convertMatrixToPVMatrix(this.activeMarker2.transformMatrix);
    }
    // update the Papervision3D view.
    this.renderEngine.render();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือเผยแพร่ข้อมูลข้างต้นไปยังผู้อื่นโดยไม่ได้รับอนุญาต

```

private function markerAdded (markerId:int) :void
{
    var x:int = markerId;
    switch(x)
    {
        case 0:
        {
            //trace("Matching Case 0");
            if(this.modelContainer0.visible == false)
            {
                //trace("Set Visible -> True");
                this.modelContainer0.visible = true;
                break;
            }
            else
                break;
        }
        case 1:
        {
            //trace("Matching Case 1");
            if(this.modelContainer1.visible == false)
            {
                //trace("Set Visible -> True");
                this.modelContainer1.visible = true;
                break;
            }
        }
        else
            break;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//trace("Matching Case 1");
```

```

if(this.modelContainer2.visible == false)
{
    //trace("Set Visible -> True");
    this.modelContainer2.visible = true;
    break;
}
else
    break;
}
}
}

private function markerRemoved (markerId:int) :void
{
    var x:int = markerId;
    switch(x)
    {
        case 0:
        {
            //trace("Matching Case 0");
            if(this.modelContainer0.visible == true)
            {
                //trace("Set Visible -> false");
                this.modelContainer0.visible = false;
                break;
            }
        }
        else
            break;
    }

    case 1:
    {
        trace("Matching Case 1");
        if(this.modelContainer1.visible == true)
        {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติไปอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และผู้ถือลิขสิทธิ์ของเอกสารฉบับนี้ที่มีการนำไปใช้

```

        //trace("Set Visible -> false");
        this.modelContainer1.visible = false;
        break;
    }
    else
        break;
}

case 2:
{
    trace("Matching Case 2");
    if(this.modelContainer2.visible == true)
    {
        //trace("Set Visible -> false");
        this.modelContainer2.visible = false;
        break;
    }
    else
        break;
}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้