

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบหาพิกัด 3 มิติ

3D Coordinate Extraction System



ปฏิญานិพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ระบบหาพิกัด 3 มิติ

### 3D Coordinate Extraction System



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## ระบบหาพิกัด 3 มิติ

นายชารา	ศรีเมือง	รหัส 46015265
นายสรพงษ์	อู่ตะเภา	รหัส 46015285
ผศ.ดร. ชูชาติ	ปิณฑวิรุจน์	(อาจารย์ที่ปรึกษา)
ปีการศึกษา 2548		

### บทคัดย่อ

ในปัจจุบันการพัฒนาทางด้านเทคโนโลยีคอมพิวเตอร์นั้น ได้ก้าวหน้าไปอย่างรวดเร็ว โดยเฉพาะทางด้านกราฟฟิกแอนิเมชันซึ่งในปัจจุบันนี้แอนิเมชัน 3 มิติ ได้เข้าไปมีบทบาทในทางการแพทย์เพื่อใช้ในการวิเคราะห์การเคลื่อนไหวของมนุษย์โดยในการวิเคราะห์จะได้มาซึ่งภาพเคลื่อนไหวที่สมจริงจำเป็นจะต้องมีโปรแกรมในการวิเคราะห์ภาพที่เรียกว่า **ระบบหาพิกัด 3 มิติ**

ระบบหาพิกัด 3 มิติคือการบันทึกภาพการเคลื่อนไหวของมนุษย์ (หรือการเคลื่อนไหวของวัตถุอื่นๆ) ที่ภาพของวัตถุที่ถูกติดด้วยมาร์คเกอร์ ต้องถูกทำการวิเคราะห์ให้ได้เป็นพิกัด 3 มิติ ของมาร์คเกอร์ระบบพิกัด 3 มิติ ประกอบด้วยกล่องหลาย ๆ กล่องที่วางตำแหน่งใดก็ได้ กล่องจะทำการบันทึกภาพของวัตถุที่ติดมาร์คเกอร์ และทำการวิเคราะห์เพื่อให้ได้พิกัด 3 มิติของมาร์คเกอร์ระบบพิกัด 3 มิติจะต่างจากเทคนิคสเตอริโอสโคปโดยตรงที่ กล่องที่ใช้ไม่จำเป็นต้องมีคุณสมบัติเหมือนกัน ซึ่งจะลดความยุ่งยากทางฮาร์ดแวร์ลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3D Coordinate Extraction System

Mr. Tara Srimuang ID. 46015265

Mr. Sorapong Aootaphao ID. 46015285

Assistance.Prof.Dr.Chuchat Pentaverut (Adviser)

Educational Year 2005

### Abstract

Presently, the computer technology develops very fast, especially in 3D animation and 3D coordinate extraction. One of the application of 3D animation in medicine include gait analysis. This thesis concerns about 3D extraction system.

3D Coordinate Extraction System consists of three or more camera taking image of object in arbitrary position. Image of marker attached on the object will be analyzed to derive 3D coordinate of markers. Compared with stereoscopy technique, this technique does not require identical camera. Hence the complexity of the hardware is greatly reduced.

## กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้เสร็จสมบูรณ์ได้ด้วยความช่วยเหลือจากผู้ที่มีส่วนเกี่ยวข้องมากมายที่สำคัญคือ ผศ.ดร. ชูชาติ ปิณฑวิรุจน์ ที่ได้ให้ความเอื้อเฟื้อทางด้านข้อมูล อุปกรณ์ต่างๆ ที่ใช้ในโครงการนี้ รวมทั้งการให้คำแนะนำปรึกษาเมื่อประสบปัญหาเสมอมา รวมทั้งความช่วยเหลือของเพื่อนๆ รุ่นพี่นักศึกษาปริญญาโท และรุ่นน้อง ที่ช่วยเป็นแบบและคำแนะนำ

ขอขอบคุณทุกๆ คนที่คอยให้ความช่วยเหลืออย่างดีเสมอมา ถ้าหากปริญญาานิพนธ์ฉบับนี้มีข้อบกพร่องหรือความผิดพลาดประการใด ทางคณะผู้จัดทำขอน้อมรับความผิดทั้งหมดและจะทำการปรับปรุงแก้ไขให้ถูกต้องในภายหลัง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VII
สารบัญตาราง	X
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการเบื้องต้น	3
2.1 การแปลงเรขาคณิตของวัตถุใน 3 มิติ	3
2.1.1 การเคลื่อนย้าย	3
2.1.2 การย่อ/ขยาย	4
2.1.3 การหมุนภาพ	4
2.2 เรขาคณิตสำหรับการเห็นใน 3 มิติ	5
2.2.1 พื้นฐานของการโปรเจกชันแบบเพอสเปกทีฟของกล้อง	5
2.2.2 การแปลงจากหน่วยความยาวเป็นพิกเซล	8
2.3 กรณีที่พิกัดของโลกไม่ซ้อนทับกับพิกัดของกล้อง	9
2.4 การปรับเทียบกล้อง	13
2.5 การหาพิกัด 3 มิติของจุดในฉากโดยใช้กล้อง 2 ตัว	16
2.6 การหาพิกัด 3 มิติของจุดในฉากโดยกล้อง 3 ตัวหรือมากกว่า	19
บทที่ 3 การออกแบบโปรแกรมและอุปกรณ์มาร์คเกอร์	20
3.1 กระบวนการขั้นตอนการทำงานโปรแกรม MATLAB5.3	20
3.1.1 การทดสอบระบบหาพิกัด 3 มิติ	20
3.2 โปรแกรม 3D coordinate extraction	25
3.2.1 โครงสร้างโปรแกรม 3D coordinate extraction	26
3.3 โปรแกรมการปรับเทียบกล้อง (Calibration)	28
3.3.1 โครงสร้างโปรแกรมการปรับเทียบกล้อง	29
3.3.2 ขั้นตอนกระบวนการปรับเทียบกล้อง Calibration	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

	หน้า
3.4 โปรแกรมคำนวณหาพิกัด 3 มิติ(Computate_ 3D Extration)	36
3.4.1 โครงสร้างโปรแกรมคำนวณหาพิกัด 3 มิติ	37
3.5 โปรแกรมการสร้างภาพ 3 มิติโดย OpenGL	39
3.5.1 โครงสร้างโปรแกรมการสร้างภาพ 3 มิติ	40
3.6 อุปกรณ์มาร์คเกอร์	41
3.6.1 อุปกรณ์แสดงตำแหน่งจุดพิกัด 3 มิติ(มาร์คเกอร์)	41
3.6.2 ชุดจ่ายไฟอุปกรณ์มาร์คเกอร์	44
บทที่ 4 การทดลองและผลการทดลอง	46
4.1 การทดลองโปรแกรมการ Calibration กล้อง	46
4.1.1 ขั้นตอนการทดลองการ Calibration กล้อง	47
4.1.2 ผลการทดลองการ Calibration โดยกำหนดจุดพิกัดตามจุดที่กำหนด	50
4.1.3 ผลการทดลองการ Error Calibartion โดยกำหนดจุดพิกัด ไม่ตรงตามที่กำหนด	51
4.2 การทดลองหาพิกัด 3 มิติของวัตถุทรงเรขาคณิต	52
4.2.1 ขั้นตอนการทดลองการหาพิกัด 3 มิติของวัตถุทรงเรขาคณิต	53
4.2.2 ผลการทดลองประสิทธิภาพการหาปริมาตรของวัตถุแบบที่ 1	55
4.2.3 การผลการทดลองประสิทธิภาพหาปริมาตรของวัตถุแบบที่ 2	56
4.2.4 ผลการทดลองประสิทธิภาพการหาปริมาตรของวัตถุแบบที่ 3	58
4.3 การทดลองคำนวณหาพิกัด 3 มิติในการเคลื่อนไหวของมนุษย์	58
4.3.1 ขั้นตอนการทดลองคำนวณหาพิกัด 3 มิติในการเคลื่อนไหวของมนุษย์	59
4.3.2 ผลการทดลองการคำนวณหาพิกัด 3 มิติในการเคลื่อนไหวของมนุษย์	60
4.4 การทดลองสร้างภาพ 3 มิติโดยOpenGL	64
4.4.1 ผลการทดลองการสร้างภาพ 3 มิติโดยOpenGL	61
บทที่ 5 สรุปและวิจารณ์	66
5.1 วิเคราะห์และสรุปผลที่ได้จากการทดลอง	66
5.2 ปัญหาที่พบและข้อจำกัดของโปรแกรม	66
5.3 แนวทางในการพัฒนาและปรับปรุงเพื่อให้มีประสิทธิภาพมากขึ้น	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

เอกสารอ้างอิง

ภาคผนวก โปรแกรมที่ใช้ในโครงการ (Source Code)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 แขนพิกัด 3 มิติและการหมุนจุดรอบแกนพิกัด มุมของการหมุน วัดตามเข็มนาฬิกาเมื่อมองจากแกนสู่จุดกำเนิด	4
รูปที่ 2.2 แบบจำลองระบบสร้างภาพ	6
รูปที่ 2.3 แสดงพิกัดภาพ	8
รูปที่ 2.4 a) แบบจำลองการเกิดภาพจากกล้อง b) เรขาคณิตของการเกิดภาพ	10
รูปที่ 2.5 ภาพของตารางหมากรุกที่นำมาใช้ในการเปรียบเทียบกล้อง	15
รูปที่ 2.6 การหาพิกัด 3 มิติโดยใช้กล้อง 3 ตัว	19
รูปที่ 3.1 Flow Chat แสดงการทดสอบการหาพิกัด 3 มิติ	20
รูปที่ 3.2 แสดงการคลิกเพื่อกำหนดจุดพิกัดของวัตถุจะต้องคลิกตามจุดที่กำหนด	22
รูปที่ 3.3 แสดงภาพการ Calibration ของกล้องสี่เหลี่ยมในมุมมองที่1(m1)	23
รูปที่ 3.4 แสดงภาพการ Calibration ของกล้องสี่เหลี่ยมในมุมมองที่2(m2)	23
รูปที่ 3.5 แสดงภาพการ Calibration ของกล้องสี่เหลี่ยมในมุมมองที่3(m3)	24
รูปที่ 3.6 แสดงภาพการPlot ค่าพิกัดจุด(u, v)ของกล้องสี่เหลี่ยมในรูป 3 มิติ	24
รูปที่ 3.7 แสดงคอมโพเนนต์โปรแกรมปรับเทียบกล้องและการหาพิกัด 3 มิติ	25
รูปที่ 3.8 Flow Chart โครงสร้างการทำงานของระบบโดยรวม	26
รูปที่ 3.9 การ Calibration โดยใช้ model ซึ่งเป็น โครงกล้องสี่เหลี่ยมลูกบาศก์ 2 ชั้น	28
รูปที่ 3.10 Flow Chart แสดงโครงสร้างโปรแกรมการปรับเทียบกล้อง	29
รูปที่ 3.11 แสดงการเขียนโปรแกรมฟังก์ชัน LButton_load_uv	31
รูปที่ 3.12 แสดงการเขียน โปรแกรมเมทริกซ์ Calibration	32
รูปที่ 3.13 แสดงการเขียน โปรแกรมฟังก์ชัน RButton_Down	32
รูปที่ 3.14 แสดงการเขียน โปรแกรมฟังก์ชัน Compulate_3DExtraction	33
รูปที่ 3.15 แสดงการเขียน โปรแกรมฟังก์ชัน QRcmp	35
รูปที่ 3.16 การกำหนดจุดพิกัดของภาพ(u, v)โดยการสแกนด้วยการคลิกเมาส์	36
รูปที่ 3.17 Flow Chart แสดงโครงสร้างโปรแกรมคำนวณหาพิกัด 3 มิติ	37
รูปที่ 3.18 ภาพ 3 มิติจากการพล็อตจุดใน OpenGLให้เป็น โครงสร้างมนุษย์	39
รูปที่ 3.19 Flow Chart แสดงโครงสร้างโปรแกรมการสร้างภาพพิกัด 3 มิติ	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 3.20 แสดงอุปกรณ์มาร์คเกอร์ที่ใช้ติดตามส่วนต่างๆ ของร่างกาย	41
รูปที่ 3.21 รูปตำแหน่งการติดอุปกรณ์มาร์คเกอร์บนร่างกายมนุษย์	42
รูปที่ 3.22 ตำแหน่งการวางกล้องทั้ง 4 ตัวในการบันทึกภาพการเคลื่อนไหวของมนุษย์	43
รูปที่ 3.23 แสดงด้านหน้าของชุดจ่ายไฟอุปกรณ์มาร์คเกอร์	44
รูปที่ 3.24 แสดงด้านหลังของชุดจ่ายไฟอุปกรณ์มาร์คเกอร์	45
รูปที่ 4.1 แสดงภาพ model โครงกล่องสี่เหลี่ยมลูกบาศก์ที่ทำจากท่อ PVC	46
รูปที่ 4.2 แสดงโปรแกรม Perspective Calibration	47
รูปที่ 4.3 แสดงไฟล์ภาพ model ที่ทำการ Capture ของแต่ละมุมกล้องบน โปรแกรม	48
รูปที่ 4.4 แสดงผลการ Calibration โดยใช้ model ซึ่งเป็น โครงกล่องสี่เหลี่ยมลูกบาศก์ 2 ชั้น	49
รูปที่ 4.5 แสดงผลการ Error Calibration โดยใช้ model ซึ่งเป็น โครงกล่องสี่เหลี่ยมลูกบาศก์ 2 ชั้น	49
รูปที่ 4.6 แสดงตำแหน่งจุดในคลิกเมาส์เพื่อระบุพิกัดภาพของวัตถุแบบที่ 1	52
รูปที่ 4.7 แสดงรูปวัตถุที่ต้องการหาพิกัด 3 มิติทั้ง 4 มุมกล้อง	53
รูปที่ 4.8 แสดงภาพการคลิกตำแหน่งพิกัดภาพของวัตถุที่ 1 ด้วยเมาส์ทั้ง 4 มุมกล้อง	54
รูปที่ 4.9 แสดงค่าพิกัดจุด(X,Y,Z) ของจุดทั้ง 7 จุดของวัตถุแบบที่ 1	54
รูปที่ 4.10 แสดงภาพการคลิกตำแหน่งพิกัดภาพของวัตถุแบบที่ 2 ด้วยเมาส์ทั้ง 4 มุมกล้อง	55
รูปที่ 4.11 แสดงค่าพิกัดจุด(X,Y,Z) ของจุดทั้ง 7 จุดของวัตถุแบบที่ 2	56
รูปที่ 4.12 แสดงภาพการคลิกตำแหน่งพิกัดภาพของวัตถุแบบที่ 3 ด้วยเมาส์ทั้ง 4 มุมกล้อง	57
รูปที่ 4.13 แสดงค่าพิกัดจุด(X,Y,Z) ของจุดทั้ง 7 จุดของวัตถุแบบที่ 3	57
รูปที่ 4.14 แสดงภาพท่าทางการเคลื่อนไหวของมนุษย์ที่ทำการบันทึกไว้	58
รูปที่ 4.15 แสดงภาพการคลิกตำแหน่งพิกัดภาพมนุษย์ด้วยเมาส์ทั้ง 4 มุมกล้อง	59
รูปที่ 4.16 แสดงค่าพิกัดจุด(X,Y,Z) ของจุดทั้ง 13 จุดจากภาพการเคลื่อนไหวมนุษย์	60
รูปที่ 4.17 แสดงภาพจริงของวัตถุแบบที่ 1	61
รูปที่ 4.18 แสดงการสร้างภาพ 3 มิติของวัตถุแบบที่ 1 ในOpenGL	62
รูปที่ 4.19 แสดงภาพจริงของวัตถุแบบที่ 2	62
รูปที่ 4.20 แสดงการสร้างภาพ 3 มิติของวัตถุแบบที่ 2 ในOpenGL	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 4.21 แสดงภาพจริงของวัตถุแบบที่ 3	63
รูปที่ 4.22 แสดงการสร้างภาพ 3 มิติของวัตถุแบบที่3 ในOpenGL	64
รูปที่ 4.23 แสดงภาพทำทางการเคลื่อนไหวจริงของมนุษย์	64
รูปที่ 4.22 แสดงการสร้างภาพ 3 มิติของทำทางการเคลื่อนไหวของมนุษย์ ในOpenGL	65



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

	หน้า
ตารางที่ 4.1 ตารางแสดงค่าพารามิเตอร์ภายในและพารามิเตอร์ภายนอก ของกล้องทั้ง 4 ตัว กรณี Calibration ที่ถูกต้อง	50
ตารางที่ 4.2 ตารางแสดงค่าพารามิเตอร์ภายในและพารามิเตอร์ภายนอก ของกล้องทั้ง 4 ตัว กรณี Calibration ไม่ถูกต้อง	51
ตารางที่ 4.3 ตารางแสดงประสิทธิภาพการหาปริมาตรของวัตถุแบบที่ 1	55
ตารางที่ 4.4 ตารางแสดงประสิทธิภาพการหาปริมาตรของวัตถุแบบที่ 2	56
ตารางที่ 4.5 ตารางแสดงประสิทธิภาพการหาปริมาตรของวัตถุแบบที่ 3	58



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

ในปัจจุบันวิวัฒนาการทางด้านคอมพิวเตอร์ได้ก้าวหน้าไปอย่างรวดเร็วมาก โดยเฉพาะการวิเคราะห์ภาพโดยระบบพิกัด 3 มิติ ซึ่งได้นำไปใช้ในการวิเคราะห์การเคลื่อนไหว จุดประสงค์หลักของการวิเคราะห์การเคลื่อนไหว คือการพัฒนาและใช้โปรแกรม ในการศึกษาและวิจัยซึ่งวิเคราะห์การเคลื่อนไหวโดยระบบพิกัด 3 มิติได้ถูกใช้ประโยชน์ในหลาย ๆ ด้าน เช่น ทางด้านการศึกษาเป็นการวิเคราะห์การเคลื่อนไหวระดับสูงในวิทยาศาสตร์การกีฬา และการใช้อุปกรณ์กีฬาบางประเภท ทางด้านกลศาสตร์ชีวภาพจะเป็นการวิเคราะห์สำหรับงานวิจัยกลศาสตร์ชีวภาพด้านการกีฬาและการแพทย์ ทางด้านศัลยกรรมกระดูกจะเป็นการวิเคราะห์คุณภาพโดยการวิเคราะห์การเคลื่อนไหวแบบ 2D และ 3D ทางด้านการบำบัดผู้ป่วยโดยใช้การวิเคราะห์การเคลื่อนไหวจากข้อมูลภาพวีดีโอที่สามารถเปรียบเทียบ วัด และบันทึกข้อมูลการเคลื่อนไหวได้ในหลาย ๆ วิธี ทางด้านอุตสาหกรรมใช้สังเกตการณ์เคลื่อนไหวที่สมบูรณ์ของเครื่องจักรและมนุษย์ และด้านการบันเทิงคือการจับการเคลื่อนไหวแบบ 3 มิติ เช่น 3D แอนิเมชันในภาพยนตร์และเกมส์คอมพิวเตอร์

จึงได้มีแนวคิดที่จะพัฒนาภาพ 3 มิติ โดยการวิเคราะห์ระบบพิกัด 3 มิติ ซึ่งในโครงการนี้จะทำการออกแบบโปรแกรมการหาระบบพิกัด 3 มิติ ซึ่งจะนำพิกัดภาพที่ได้มาแปลให้อยู่ในรูปแบบพิกัด 3 มิติ โดยใช้ทฤษฎีเรขาคณิตสำหรับการเห็นใน 3 มิติ การปรับเทียบกล้อง การหาพิกัด 3 มิติของจุดในฉาก และการสร้างภาพ 3 มิติในการวิเคราะห์ท่าค้ำแหน่งของวัตถุ

### 1.1 รายละเอียดโดยย่อของโครงการ

ระบบพิกัด 3 มิติคือการแคปเจอร์ภาพการเคลื่อนไหวของมนุษย์ (หรือการเคลื่อนไหวของวัตถุอื่น ๆ) โดยจะทำการปรับแต่งกล้องเพื่อนำพารามิเตอร์ที่ได้ไปคำนวณหาพิกัด 3 มิติของไฟล์ภาพที่ได้ทำการบันทึกจากกล้อง ซึ่งภาพที่ได้จากกล้องเมื่อทำการแคปเจอร์ภาพจะได้ภาพที่มีพิกัด 2 มิติ ซึ่งพิกัดที่ 3 จะหายไปคือมีค่าเท่ากับศูนย์ซึ่งไม่สามารถสร้างภาพพิกัด 3 มิติได้ จึงต้องมีการทำโปรแกรมเพื่อหาพารามิเตอร์ที่ใช้ในการคำนวณพิกัดของภาพจากกล้อง โดยนำพารามิเตอร์ค่าพิกัดของโลก 3 มิติและพิกัดภาพ 2 มิติมาคำนวณทางคณิตศาสตร์เพื่อให้ได้ภาพพิกัด 3 มิติที่ถูกต้อง

## 1.2 วัตถุประสงค์ของโครงการ

- เพื่อศึกษาสร้างโปรแกรมใช้งาน (Application) โดยใช้โปรแกรม Microsoft Visual C++
- เพื่อศึกษาการใช้ฟังก์ชันคณิตศาสตร์ของโปรแกรม Match Lab 5.3 เพื่อวิเคราะห์สมการ
- เพื่อศึกษาหลักการประมวลผลภาพ

## 1.3 ขอบเขตของโครงการ

- ขอบเขตของการปรับเทียบกล้อง

เป็นการเขียนโปรแกรมในเชิงวิเคราะห์สมการทางคณิตศาสตร์ให้คำนวณพารามิเตอร์การปรับเทียบกล้องซึ่งจะได้ค่าโฟกัสและค่าความบิดเบือนของกล้องและนำไปหาพิกัด 3 มิติของภาพได้

- ขอบเขตการทำจุดมาร์คเกอร์

ทำการออกแบบจุดมาร์คเกอร์ที่ติดจุดข้อต่อของร่างกาย โดยจุดมาร์คเกอร์จะใช้ LED SUPERBITE ในการแสดงตำแหน่งจุดข้อต่อของร่างกาย

- ขอบเขตการหาพิกัด 3 มิติ

ทำการรับสัญญาณภาพเข้ามา โดยนำพิกัด  $(u, v)$  ของภาพและพิกัดโลก  $x, y, z$  มาใส่ matrix calibration และแก้สมการ โดยใช้วิธีการ SVD จะได้พารามิเตอร์หาพิกัด 3 มิติ

## 1.4 ประโยชน์และผลที่คาดว่าจะได้รับ

โครงการนี้ทำให้ได้โปรแกรมใช้งานที่สามารถสร้างภาพ 2 มิติ เป็น 3 มิติเพื่อนำไปประยุกต์ใช้งานในด้านต่างๆ เช่น ทางด้านการแพทย์คือ การวิเคราะห์การเคลื่อนไหวมนุษย์ การรักษาผู้ป่วยโดยวิธีกายภาพบำบัด ทางด้านด้านกีฬา คือ ศึกษาการเคลื่อนไหวในการออกกำลังกายที่ถูกต้อง เพื่อลดการบาดเจ็บขณะออกกำลังกาย

## บทที่ 2

### ทฤษฎี และ หลักการเบื้องต้น

#### การเห็นใน 3 มิติ ( 3D Vision )

ในบทนี้เรากล่าวถึงการเห็น 2 มิติ 3 มิติ มีหลายสาเหตุที่ยากลำบากในการนำข้อมูลการเห็น 2 มิติ เป็น 3 มิติ ได้แก่

1. ระบบภาพของกล้องและของตามนุษย์มีการโปรเจกชันแบบพอสเป็คทีฟ (Perspective Projection) ซึ่งทำให้มีการสูญเสียข้อมูลอย่างมาก
2. ความสัมพันธ์ระหว่างภาพใน 2 มิติและเรขาคณิตใน 3 มิติของจุดที่สอดคล้องกันในฉากมีความซับซ้อนมาก
3. การบังกันของวัตถุในฉากเป็นการเพิ่มความยุ่งยากของการเห็นภาพใน 3 มิติ
4. การปรากฏอยู่ของนอยส์ในภาพ

#### 2.1 การแปลงเรขาคณิตของวัตถุใน 3 มิติ (Geometric Transformation in 3D)

หมายถึงการแปลงเรขาคณิตสำหรับวัตถุใน 3 มิติ ซึ่งมีพิกัดอยู่ในรูป  $(x, y, z)$

##### 2.1.1 การเคลื่อนย้าย (Translation)

การย้ายจุดที่มีพิกัด  $(x, y, z)$  ไปยังจุดใหม่ที่มีระยะเคลื่อนที่  $(t_x, t_y, t_z)$  ดังสมการ

$$U = x + t_x$$

$$V = y + t_y$$

$$W = z + t_z$$

(2.1)

สามารถเขียนในระบบพิกัดโฮโมจีเนียสได้เป็น

$$\begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(2.2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.2 การย่อ/ ขยาย(Scaling)

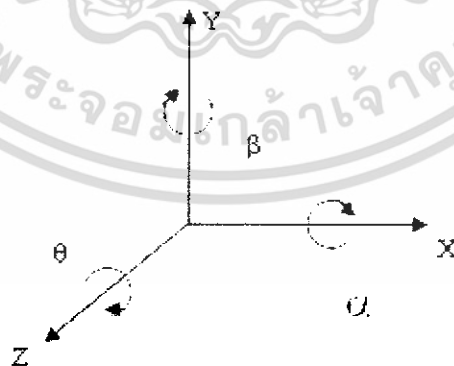
การย่อ/และการขยายใน 3 มิติ ที่มีพารามิเตอร์  $S_x, S_y$  และ  $S_z$  ตามแนวแกน  $x, y$  และ  $z$  เขียนในรูปเมทริกซ์ได้เป็น

$$\begin{pmatrix} U \\ V \\ W \\ I \end{pmatrix} = \begin{pmatrix} S_x & 0 & 0 & t_x \\ 0 & S_y & 0 & t_y \\ 0 & 0 & S_z & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(2.3)

### 2.1.3 การหมุนภาพ (Rotation)

การหมุนใน 3 มิติ สามารถแบ่งได้เป็น 3 ชนิดนั่นคือหมุนรอบแกน  $x$  แกน  $y$  และแกน  $z$  การหมุนวัตถุรอบจุดใดๆ เราต้องทำการย้ายจุดนั้นมาที่จุดกำเนิดจากนั้นทำการหมุนและย้ายกลับไปที่ได้เดิม



รูป 2.1 แกนพิกัด 3 มิติและการหมุนจุดรอบแกนพิกัด มุมของการหมุนวัดตามเข็มนาฬิกาเมื่อมองจากแกนสู่จุดกำเนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหมุนรอบแกน Z ด้วยมุม  $\theta$

$$R_{\theta} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(สมการ  $R_{\theta}$ ) (2.4)

การหมุนรอบแกน X ด้วยมุม  $\alpha$

$$R_{\alpha} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(สมการ  $R_{\alpha}$ ) (2.5)

การหมุนรอบแกน Y ด้วยมุม  $\beta$

$$R_{\beta} = \begin{pmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(สมการ  $R_{\beta}$ ) (2.6)

## 2.2 เรขาคณิตสำหรับการเห็นใน 3 มิติ (Geometry for 3D Vision)

### 2.2.1 พื้นฐานของการโปรเจกชันแบบเพอร์สเปกทีฟของกล้อง (Basis of Perspective Camera)

ภาพ 2 มิติส่วนใหญ่ได้มาจากการมองวัตถุ 3 มิติผ่านเลนส์ของตาหรือเลนส์ของกล้องภาพ สามารถถูกจำลองได้โดยการใช้ Perspective Transformation รูป 2.2 เรขาคณิตพิกัดโลก (World Coordinate System) ด้วยอักษรตัวใหญ่ (X, Y, Z) และแทนแกนของระบบพิกัดของกล้องด้วยตัวอักษรตัวเล็ก (x, y, z) เราสมมติว่าพิกัดของโลกซ้อนทับกับระบบพิกัดของกล้องแสงจากวัตถุใน 3 มิติ ส่องผ่านศูนย์กลางของการโปรเจกชันแล้วโปรเจกต์ลงบนภาพ ให้จุด  $(x_s, y_s, z_s)$  แทนจุด

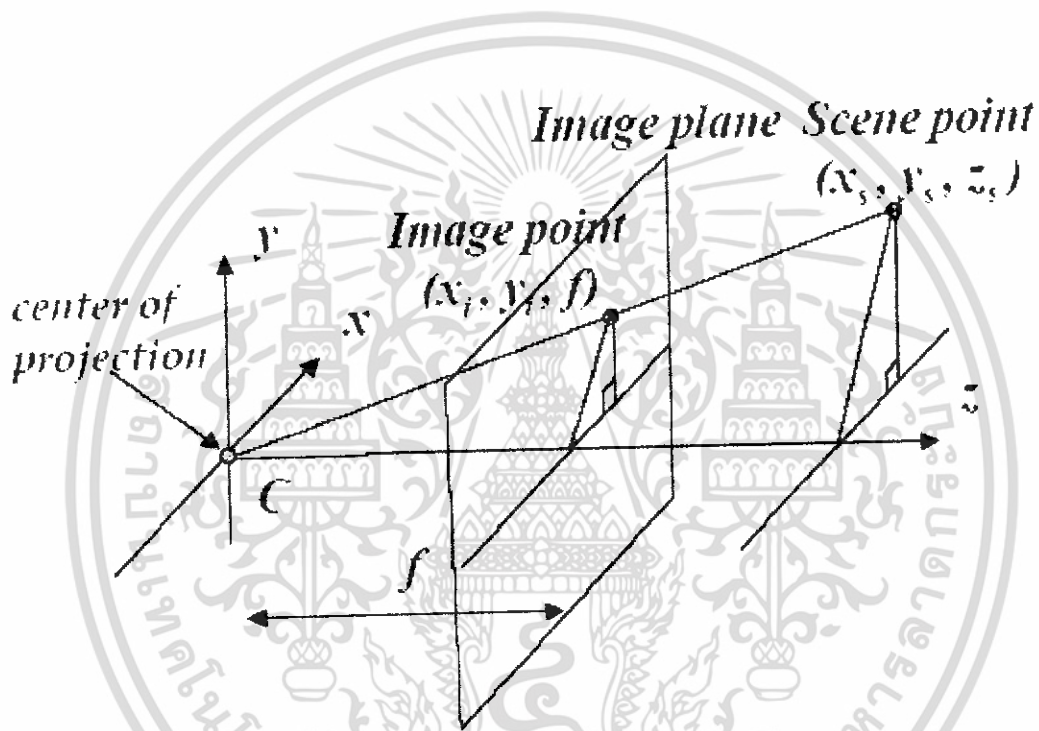
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิกัดของวัตถุ 3 มิติ ให้  $(x_s, y_s)$  ให้  $f$  คือโฟกัสของเลนส์ โดยใช้ตามเหลี่ยมคล้าย

$$x_i = \frac{fx_s}{z_s}$$

$$y_i = \frac{fy_s}{z_s}$$

(2.7)



รูปที่ 2.2 แบบจำลองระบบสร้างภาพ

ให้

$$V = \begin{pmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{pmatrix}$$

(2.8)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นเวกเตอร์ที่ประกอบจุดพิกัดของวัตถุเวกเตอร์ในระบบ โฮโมจีเนียส  $\tilde{V}$  ของ  $V$  คือ

$$\tilde{V} = \begin{pmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{pmatrix}$$

(2.9)

เป็นเวกเตอร์ในระบบพิกัด Cartesian  $V$  สามารถได้จากเวกเตอร์ในระบบ โฮโมจีเนียสได้โดยการหารสามสมาชิกแรกด้วยสมาชิกที่สี่

พิจารณาเมทริกซ์ของ Perspective Transformation

$$P = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

(2.10)

ทำการคูณเวกเตอร์

$$\tilde{W} = P\tilde{V}$$

(2.11)

ได้เป็น

$$\tilde{W} = \begin{pmatrix} fx_s \\ fy_s \\ z_s \end{pmatrix}$$

(2.12)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการนอร์มอลไลซ์สมาชิกที่สามให้เป็น 1 เพื่อแปลงเป็นเวกเตอร์ในระบบพิกัด Cartesian ดังนั้น

$$W = \begin{pmatrix} \frac{fx_s}{z_s} \\ 1 \\ \frac{fy_s}{z_s} \\ z_s \end{pmatrix}$$

(2.13)

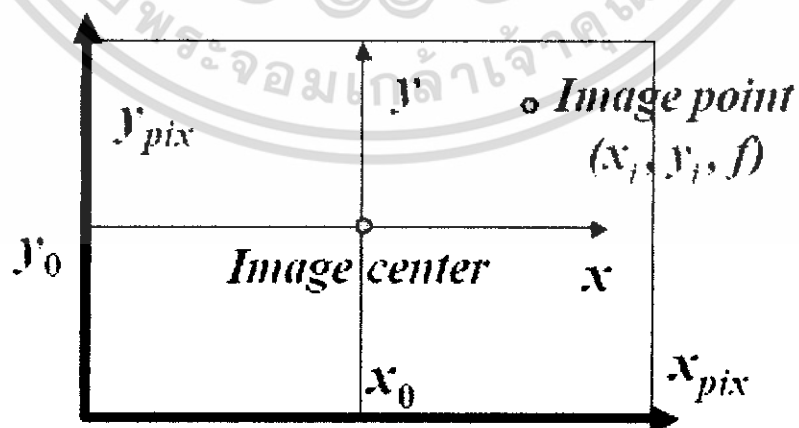
### 2.2.2 การแปลงจากหน่วยความยาวเป็นพิกเซล

การหาพิกัดของจุดที่ถูกโปรเจกชัน  $(x_i, y_i, 0)$  ในหน่วยของพิกเซล กำหนดให้พิกัดของภาพมีจุดกำเนิดอยู่ที่มุมด้านล่างซ้ายของภาพดังแสดงในรูป 2.3 การแปลงจากความยาวเป็นพิกเซล เราจำเป็นต้องรู้ค่าการปรับสเกล (Aspect Ratio) ของแต่ละแกนของระบบพิกัดภาพ

โดยที่  $k_x$  คือค่าการปรับสเกลในทิศทาง  $x$ ,  $k_y$  คือค่าการปรับสเกลในทิศทาง  $y$  และ  $s$  แทนพารามิเตอร์ของการเอียง เราเขียนในรูปเทียบบถกล้อง (Camera Calibration Procedure) เป็นจุดตัดระหว่างแกนออบติคกับระนาบภาพ

พิกัดของจุด  $(x_i, y_i, 0)$  หาได้จากความสัมพันธ์

$$\begin{aligned} x_{pix} &= k_x x_i + s y_i + x_0 \\ &= k_x f \frac{x_s}{z_s} + s f \frac{y_s}{z_s} + x_0 \end{aligned} \quad (2.14)$$



รูปที่ 2.3 แสดงพิกัดภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}
 y_{pix} &= k_y y_i + y_0 \\
 &= k_y f \frac{y_s}{z_s} + y_0
 \end{aligned} \tag{2.15}$$

เราสามารถอยู่ในรูปของเมทริกซ์ได้เป็น

$$\begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = \begin{pmatrix} \alpha_x & s & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_s \\ y_s \\ z_s \\ 1 \end{pmatrix} \tag{2.16}$$

โดยที่  $\alpha_x = k_x x_s$  ซึ่งเป็นค่าความยาวโฟกัสในทิศ  $x$  ในหน่วยของพิกเซลและ  $\alpha_y = k_y x_s$  ซึ่งเป็นค่าความยาวโฟกัสในทิศ  $y$  ในหน่วยของพิกเซล เราสามารถเขียน

$$\begin{pmatrix} \alpha_x & s & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha_x & s & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = K [I_3 | O_3] \tag{2.17}$$

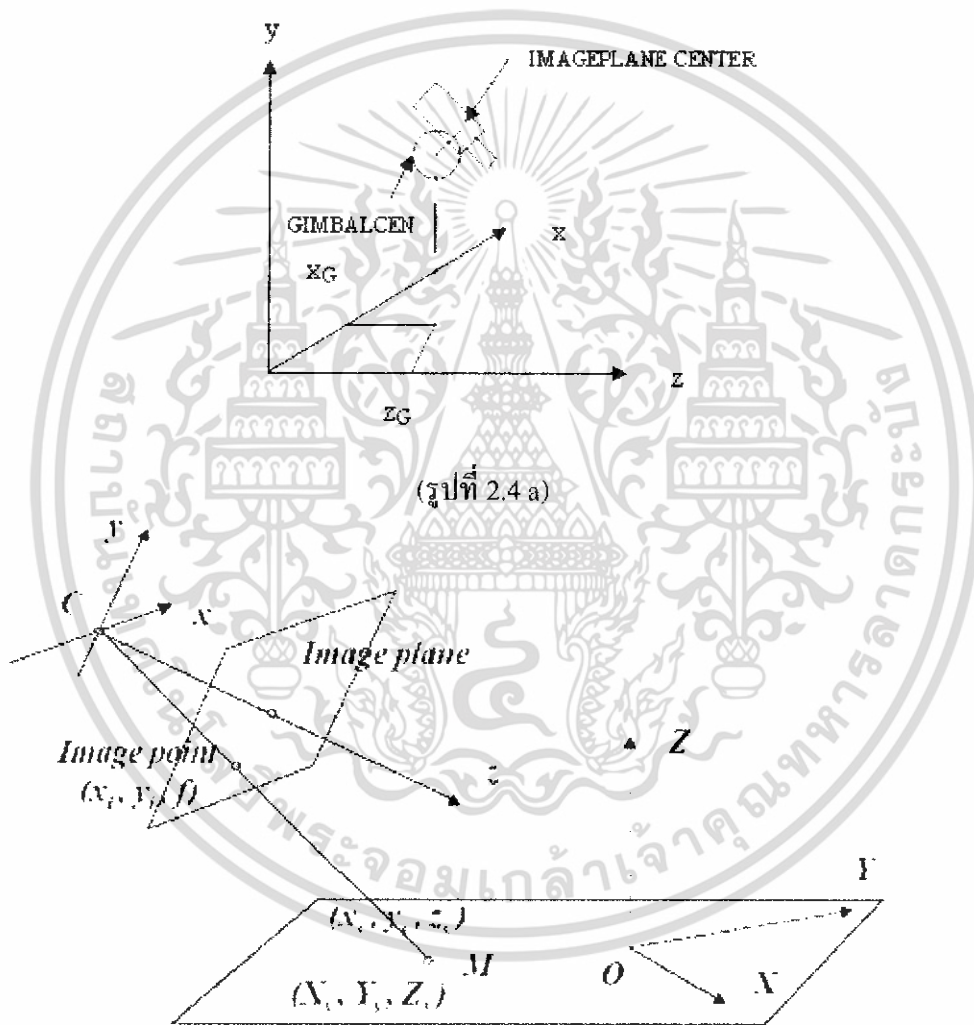
เมทริกซ์  $K$  มีลักษณะเป็นสามเหลี่ยมข้างบน (Upper Triangle) ขนาด  $3 \times 3$  มีชื่อว่า Calibration Matrix เป็นเมทริกซ์ที่ประกอบพารามิเตอร์ภายใน (Intrinsic Parameter) ที่สำคัญของกล้อง

### 2.3 กรณีที่พิกัดของโลกไม่ซ้อนทับกับพิกัดของกล้อง

แบบจำลองการเกิดภาพจากกล้องที่ไม่มีสมมติฐานรูปที่ 2.4a) แสดงกล้องอิเล็กทรอนิกส์ในช่วงว่างพิกัดของโลก (World Coordinate Space) กล้องนี้ถูกยึดโดยหัวยึดกล้อง (Gimbal) ที่สามารถหมุนทำส่าย (Pan) มุม  $\theta$  กับแนวนอน (Horizontal) และเอียง (Tilt) ทำมุม  $\phi$  กับแนวตั้ง จุดกึ่งกลางของหัวยึดกล้องอยู่ที่จุดพิกัด  $(X_G, Y_G, Z_G)$  เทียบกับระบบอ้างอิงของโลก จุดกึ่งกลางของหัวเอกซาร์นี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยัดกล้องและจุดกึ่งกลางของการโปรเจกชันของระนาบภาพอยู่ห่างกันด้วยระยะ  $(X_0, Y_0, Z_0)$  รูป 2.4 b) แสดงระบบพิกัด โลกและระบบพิกัดกล้องที่ไม่ซ้อนทับกัน

กรณีที่จุดกึ่งกลางของการโปรเจกชันของกล้องถูกวางไว้ที่จุดกึ่งกลางของระบบพิกัดอ้างอิง โลก ไม่มีการส่ายหรือเอียงทำมุมกับแกนอ้างอิง จุดกึ่งกลางของหัวยัดกล้องและของ



(รูปที่ 2.4 b)

รูปที่ 2.4 a) แบบจำลองการเกิดภาพจากกล้อง b) เรขาคณิตของการเกิดภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระนาบภาพอยู่ที่เดียวกัน แบบจำลองการเกิดภาพ ( Imaging Model ) ในระบบพิกัดโฮโมจีเนียสนั้นคือ

$$\tilde{W} = M\tilde{V} \quad (2.18)$$

โดยที่  $\tilde{V}$  คือเวกเตอร์ของจุดบนวัตถุในระบบพิกัดโฮโมจีเนียส  $\tilde{W}$  คือเวกเตอร์ของจุดบนระนาบภาพในระบบพิกัดโฮโมจีเนียสและ  $M$  คือเมทริกซ์ของ Perspective Transformation ในกรณีนี้แบบจำลองการเกิดภาพจากกล้องสามารถหาได้ง่ายโดยการเปลี่ยนแปลง (Modify) สมการ (2.17)

การที่ ( หัวชี้คกตั้ง ) กล้องอยู่ที่จุดพิกัด  $(X_G, Y_G, Z_G)$  เทียบกับระบบพิกัดอ้างอิงของโลก ดังนั้นแบบจำลองใหม่คือ

$$\tilde{W} = MT_G\tilde{V} \quad (2.19)$$

โดยที่

$$T_G = \begin{pmatrix} 1 & 0 & 0 & -X_G \\ 0 & 1 & 0 & -Y_G \\ 0 & 0 & 1 & -Z_G \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.20)$$

การส่ายหรือการเอียง ( Pan and Tilt ) โดยใช้เมทริกซ์การหมุนคูณเข้า

$$\tilde{W} = MRT_G\tilde{V} \quad (2.21)$$

โดยที่  $R = R_\phi R_\theta$

$$R_\theta = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(2.22)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ

$$R_\phi = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & \sin\phi & 0 \\ 0 & -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.23)$$

ท้ายสุดจุดกึ่งกลางของหัวขีตกล้องและจุดกึ่งกลางของการโปรเจกชันของระนาบภาพอยู่ห่างกันด้วยระยะ  $(X_0, Y_0, Z_0)$  ดังนั้น แบบจำลองของการเกิดภาพจากกล้องคือ

$$\tilde{W} = M T_C R T_G \tilde{V} \quad (2.24)$$

โดยที่

$$T_C = \begin{pmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.25)$$

สมการ (2.23) สามารถเขียนใหม่ได้เป็น

$$\begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = K \begin{bmatrix} I_3 & O_3 \end{bmatrix} \begin{pmatrix} R & -T \\ O_3^T & 1 \end{pmatrix} \begin{pmatrix} x_s \\ y_s \\ z_s \\ 1 \end{pmatrix} \quad (2.26)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่  $R$  แทนเมทริกซ์ที่เป็นผลรวมของการหมุนระบบพิกัด และ  $T$  แทนเวกเตอร์ที่เป็นผลรวมของการย้ายระบบพิกัด เรากล่าวว่าเมทริกซ์

$$\begin{pmatrix} R & -T \\ O_3^T & 1 \end{pmatrix}$$

(เมทริกซ์)

เป็นเมทริกซ์ที่รวมเอาพารามิเตอร์ภายนอก ( Extrinsic Parameter ) ของขบวนการปรับเทียบกล้อง เราสามารถเขียนสมการ ( 2.25 ) ในรูปแบบที่ง่ายได้เป็น

$$X = MX \quad (2.27)$$

โดยที่  $M = [KR \mid -KRT]$  (2.28)

#### 2.4 การปรับเทียบกล้อง ( Camera Calibration )

การปรับเทียบกล้องเป็นการหาประมาณเมทริกซ์  $M$  จากจุดในฉาก 3 มิติที่รู้ตำแหน่งและภาพที่เกิดขึ้นของจุด จากนั้นทำการหาพารามิเตอร์ภายในและพารามิเตอร์ภายนอก พิจารณาการปรับเทียบโดยใช้กล้องเดี่ยว

ในการหาเมทริกซ์  $M$  เราทำการหาจุดในฉากที่ทราบพิกัด  $X = [x, y, z]^T$  และจุดในภาพ 3 มิติที่สอดคล้องกัน  $[u, v]^T$  ซึ่งจะได้สมการที่อยู่ในรูปสมการ (2.26) ดังนี้

$$\begin{pmatrix} \alpha_u \\ \alpha_v \\ \alpha \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \begin{pmatrix} x_s \\ y_s \\ z_s \\ 1 \end{pmatrix}$$

(2.29)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{pmatrix} \alpha_U \\ \alpha_V \\ \alpha \end{pmatrix} = \begin{pmatrix} m_{11}x + m_{12}y + m_{13}z + m_{14} \\ m_{21}x + m_{22}y + m_{23}z + m_{24} \\ m_{31}x + m_{32}y + m_{33}z + m_{34} \end{pmatrix}$$

(2.30)

ทำการนอร์มอลไลซ์ให้สมาชิกที่ 3 เป็น 1 เราจะได้สมการ

$$\begin{aligned} U(m_{31}x + m_{32}y + m_{33}z + m_{34}) &= m_{11}x + m_{12}y + m_{13}z + m_{14} \\ V(m_{31}x + m_{32}y + m_{33}z + m_{34}) &= m_{21}x + m_{22}y + m_{23}z + m_{24} \end{aligned}$$

(2.31)

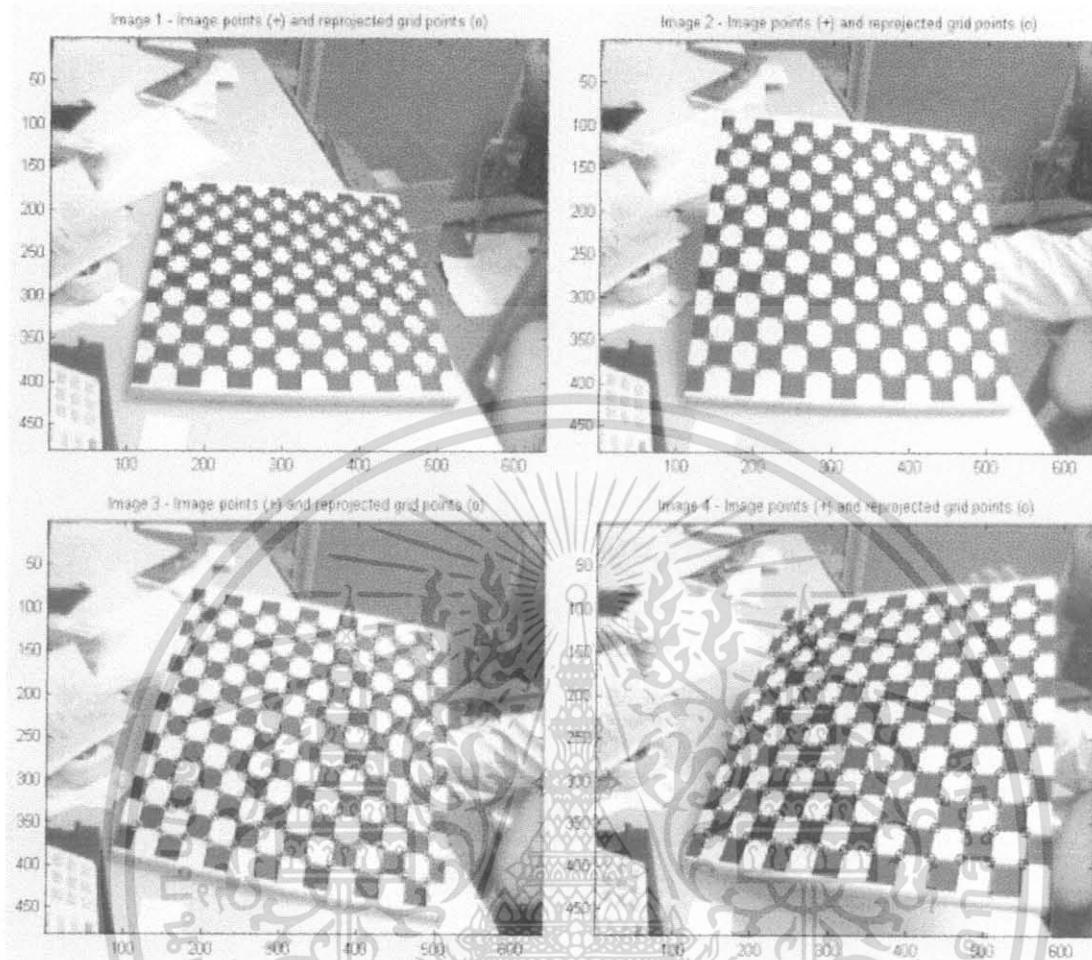
ดังนั้นเราจะได้สมการเชิงเส้น 2 สมการสำหรับจุดในฉาก 3 มิติ หนึ่งจุดและจุดในภาพ 2 มิติที่สอดคล้องกัน ทำการเขียนสมการ(2.30)ใหม่ในรูปของเมทริกซ์ได้เป็น

$$\begin{bmatrix} x & y & z & 1 & 0 & 0 & 0 & 0 & -ux & -uy & -uz & -u \\ 0 & 0 & 0 & 0 & x & y & z & 1 & -vx & -vy & -vz & -v \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ \vdots \\ m_{34} \end{bmatrix} = 0$$

(2.32)

จะเห็นว่าเรามีตัวแปรที่ไม่ทราบค่า 11 ตัวแปร แทนที่จะเป็น 12 ตัวแปรเนื่องจากแฟลคเตอร์การย่อ/ขยายที่ไม่สามารถทราบค่าได้ ในการแก้สมการโฮโมจีเนียสเราต้องใช้อย่างน้อย 6 จุด ถ้ามีมากกว่า 6 จุดเราจะได้สมการ Over-determined ซึ่งสามารถแก้ได้โดยใช้วิธี Least Square

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.5 ภาพของตารางหมากรุกที่นำมาใช้ในการเปรียบเทียบกล้อง

เมื่อได้เมทริกซ์  $M$  แล้วขั้นตอนต่อไปทำการแยกพารามิเตอร์ภายใน (Intrinsic Parameter) เนื่องจาก

$$M = [KR \mid -KRT] = [A \mid B] \quad (2.33)$$

ดังนั้นเมทริกซ์ย่อย  $3 \times 3$  แทนด้วย  $A$  เวกเตอร์คอลัมน์ทางขวามือสุดแทนด้วย  $b$  เวกเตอร์ที่มีพารามิเตอร์ระยะเคลื่อนที่สามารถแยกออกมาได้ง่ายโดย  $t = -A^{-1}b$  เวกเตอร์  $t$  ให้ข้อมูลที่บอกถึงตำแหน่งกึ่งกลางของ Image Plane

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นพิจารณา  $A = KR$  โดยที่  $K$  เป็นเมทริกซ์สามเหลี่ยมบน(Upper Triangle) และเมทริกซ์  $R$  เป็นเมทริกซ์ออร์โธกอนอล การแยกเมทริกซ์  $K$  และ  $R$  สามารถทำได้โดยใช้เทคนิค QR Decomposition สำหรับเมทริกซ์  $A$

## 2.5 การหาพิกัด 3 มิติของจุดในฉากโดยใช้กล้อง 2 ตัว

ในหัวข้อนี้เราพิจารณาการหาพิกัด 3 มิติของจุดในฉากโดยใช้กล้อง 2 ตัว โดยที่กล้องทั้งสองตัวผ่านขบวนการปรับเทียบกล้องที่กล่าวในหัวข้อ ในขบวนการปรับเทียบเราให้กล้องทั้งสองถ่ายภาพวัตถุในฉาก 3 มิติที่เราทราบพิกัด ซึ่งอาจเป็นตารางหมากรุกที่เราทราบขนาดของแต่ละตาราง (แสดงในรูป 2.5) จากนั้นทำการหาพิกัดของจุดตัดของตารางในกล้องทั้งสอง ใช้สมการ (9.15) ทำการคำนวณเมทริกซ์ของ Perspective Transformation  $M$  สำหรับแต่ละกล้องต่อไปทำการวางวัตถุที่เราต้องการหาพิกัด 3 มิติในฉาก ทำการหาจุดสอดคล้องของจุดในฉาก 3 มิติของกล้องทั้งสองตัว กำหนดให้เมทริกซ์ของ Perspective Transformation  $M$  ของกล้องที่ 1 และ 2 เป็น  $M$  และ  $M'$  ตามลำดับ เราแทนแต่ละแถวของเมทริกซ์  $M$  ด้วย  $m_1^T, m_2^T, m_3^T$  ทำนองเดียวกันสำหรับกล้องที่สอง เราแทนแต่ละแถวของเมทริกซ์  $M'$  ด้วย  $m'_1, m'_2, m'_3$  สำหรับกล้องที่ 1 เราได้

$$u = \begin{bmatrix} u \\ y \\ w \end{bmatrix} = MX = \begin{bmatrix} m_1^T \\ m_2^T \\ m_3^T \end{bmatrix} x \quad (2.34)$$

สำหรับกล้องที่ 2 เราได้

$$u = \begin{bmatrix} u \\ y \\ w \end{bmatrix} = M'X = \begin{bmatrix} m'^T_1 \\ m'^T_2 \\ m'^T_3 \end{bmatrix} x \quad (2.35)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โดยที่  $u$  และ  $v$  แทนพิกัดในภาพของวัตถุในกล้อง 1 และ 2 ตามลำดับ ในการกำจัดค่าสเกลที่ไม่ทราบค่า เราทำการหาสัดส่วนระหว่างสามแถวในเมทริกซ์ Perspective Transformation  $M$

$$u : v : w = m_1^T X : m_2^T X : m_3^T X \quad (2.36)$$

$$u' : v' : w' = m_1'^T X : m_2'^T X : m_3'^T X$$

ดังนั้นเราได้สมการสำหรับกล้อง 1 และกล้อง 2

$$\begin{aligned} um_2^T X &= vm_1^T X & u'm_2'^T X &= v'm_1'^T X \\ um_3^T X &= vm_1^T X & u'm_3'^T X &= v'm_1'^T X \\ um_3^T X &= vm_2^T X & u'm_3'^T X &= v'm_2'^T X \end{aligned} \quad (2.37)$$

สมการ (2.37) สามารถเขียนเป็น

$$\begin{bmatrix} um_2^T - vm_1^T \\ um_3^T - vm_1^T \\ um_3^T - vm_2^T \end{bmatrix} X = 0 \quad (2.38a)$$

$$\begin{bmatrix} u'm_2'^T - v'm_1'^T \\ u'm_3'^T - v'm_1'^T \\ u'm_3'^T - v'm_2'^T \end{bmatrix} X = 0 \quad (2.38b)$$

**73151**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการคูณแถวแรกของ (2.35.a) ด้วย  $w$  แถวที่สองด้วย  $-v$  แล้วทำการบวก เราได้

$$(uwm_2^T - vwm_1^T - uvm_3^T + vwm_1^T)X = (uwm_2^T - uvm_3^T)X = 0 \quad (2.39)$$

จะเห็นว่าสมการ (2.36) มีคุณสมบัติ Linearly Dependent กับแถวที่สามของสมการ (2.35a) ดังนั้นแต่ละกล้องเราใช้ได้เพียง 2 สมการดังนี้

$$\begin{bmatrix} um_3^T - vm_1^T \\ um_3^T - vm_2^T \end{bmatrix} X = 0 \quad \text{และ} \quad \begin{bmatrix} u'm_3^T - v'm_1^T \\ u'm_3^T - v'm_2^T \end{bmatrix} X = 0 \quad (2.40)$$

เขียน (2.37) ใหม่ได้ดังนี้

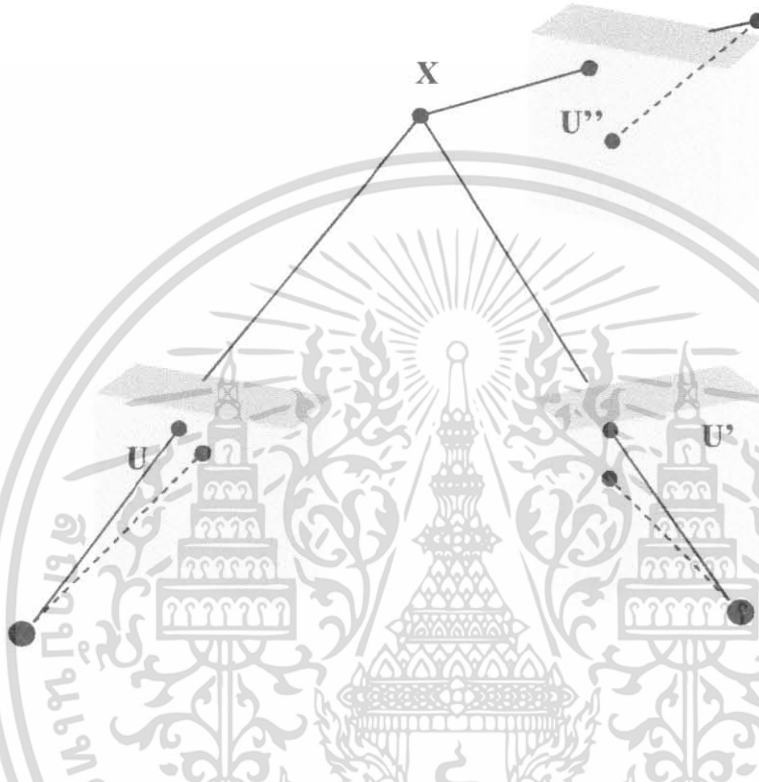
$$\begin{bmatrix} um_3^T - vm_1^T \\ vm_3^T - wm_2^T \\ um_3^T - vm_1^T \\ vm_3^T - wm_2^T \end{bmatrix} X = AX = 0 \quad (2.41)$$

ทำการแก้สมการโฮโมจีนีส (2.38) สำหรับพิกัด 3 มิติ  $X$  โดยเราสนใจเฉพาะคำตอบประเภท Non-Trivial หรือกรณี  $\det(A) = 0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6 การหาพิกัด 3 มิติของจุดในฉากโดยกล้อง 3 ตัวหรือมากกว่า

สำหรับกรณีของการใช้ 2 กล้องหรือมากกว่าเราได้สมการในลักษณะเดียวกัน เช่นถ้าใช้กล้อง 3 ตัวดังแสดงในรูป 2.6 เราได้สมการ



รูป 2.6 การหาพิกัด 3 มิติโดยใช้กล้อง 3 ตัว

$$\begin{bmatrix}
 um_3^T - wm_1^T \\
 v'm_3^T - w'm_2^T \\
 u''m_3^T - w''m_1^T \\
 v''m_3^T - w''m_2^T \\
 u'''m_3^T - w'''m_1^T \\
 v'''m_3^T - w'''m_2^T
 \end{bmatrix} X = AX = 0 \quad (2.42)$$

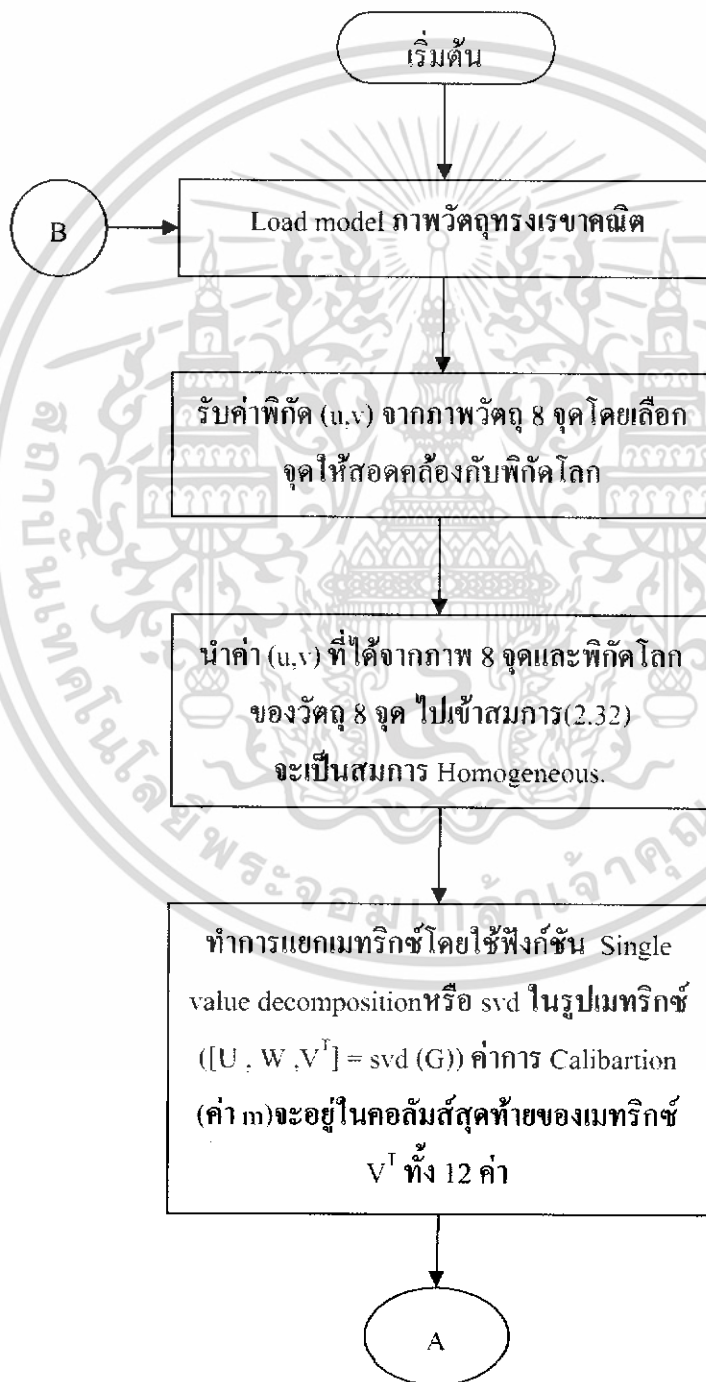
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3

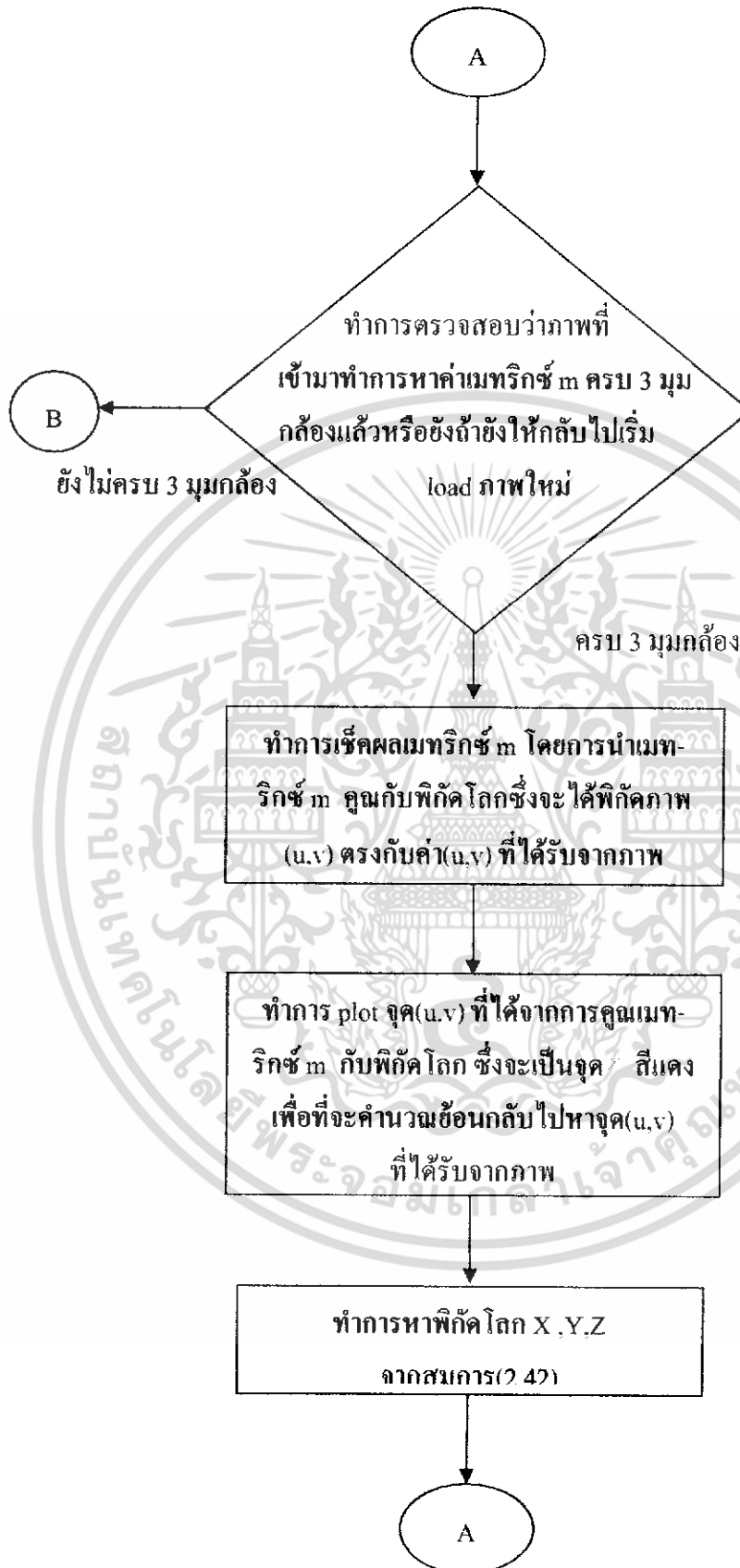
#### การออกแบบโปรแกรมและอุปกรณ์มาร์คเกอร์

##### 3.1 กระบวนการขั้นตอนการทำงานโปรแกรม MATLAB5.3

###### 3.1.1 การทดสอบระบบหาพิกัด 3 มิติ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

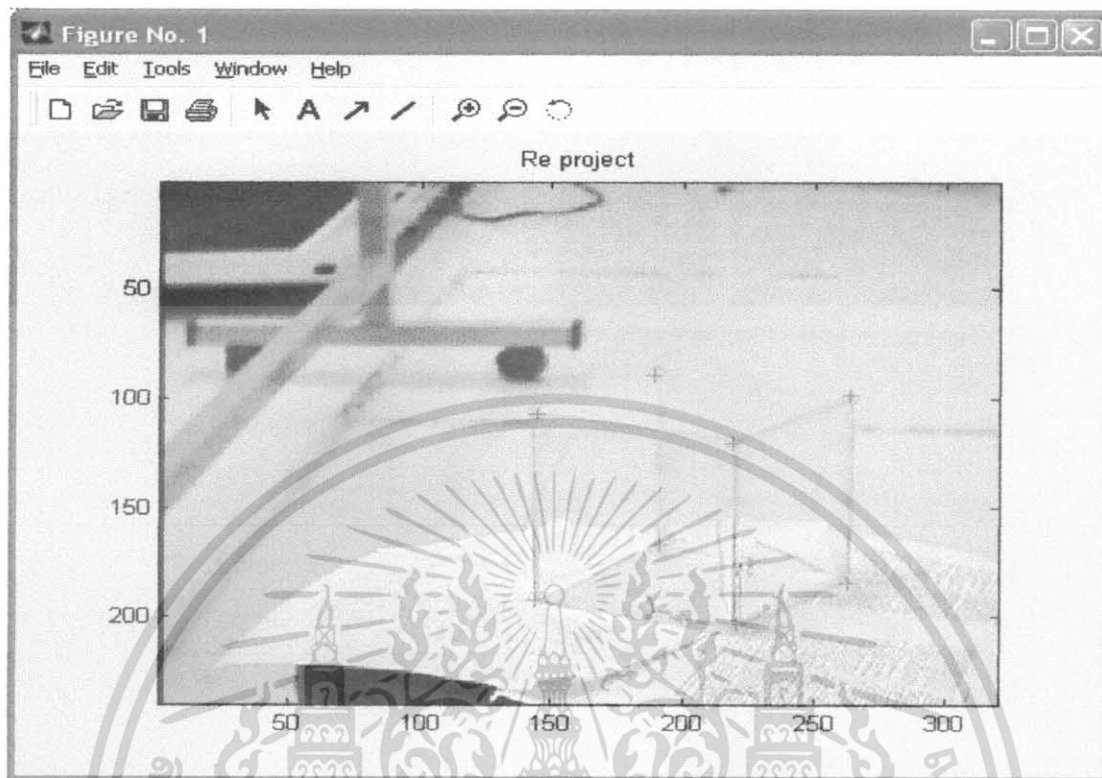


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

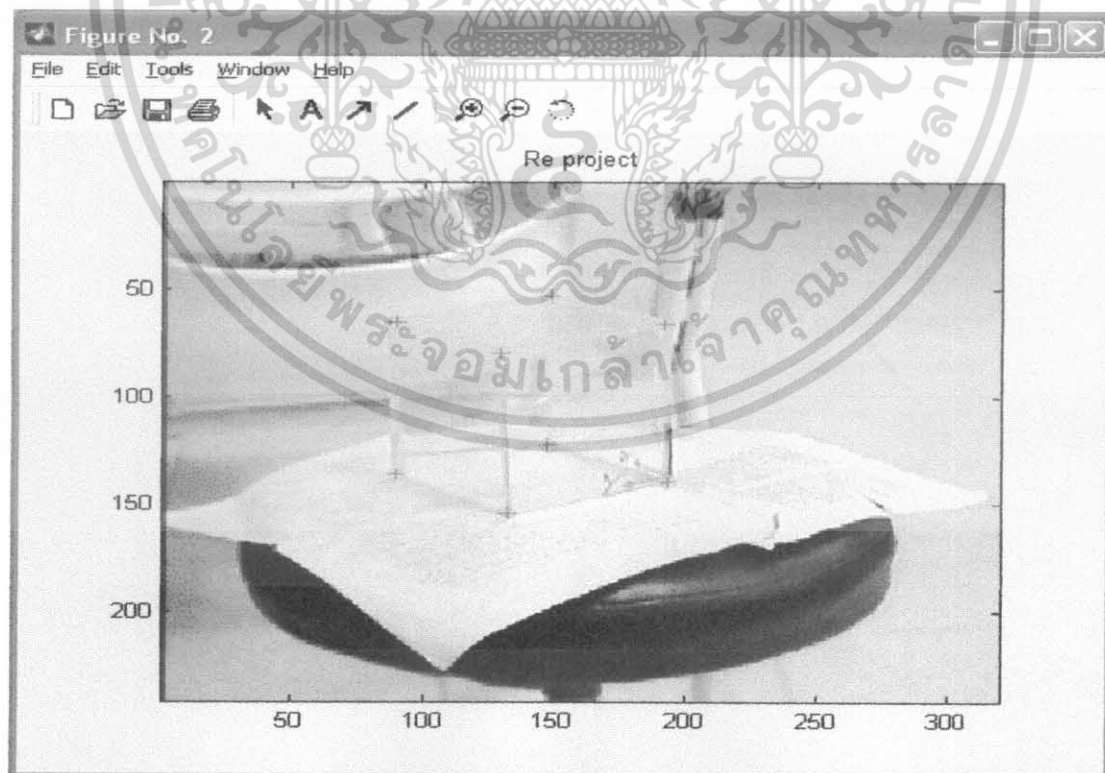


รูปที่ 3.2 แสดงการคลิกเพื่อกำหนดจุดพิกัดของวัตถุจะต้องคลิกตามจุดที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

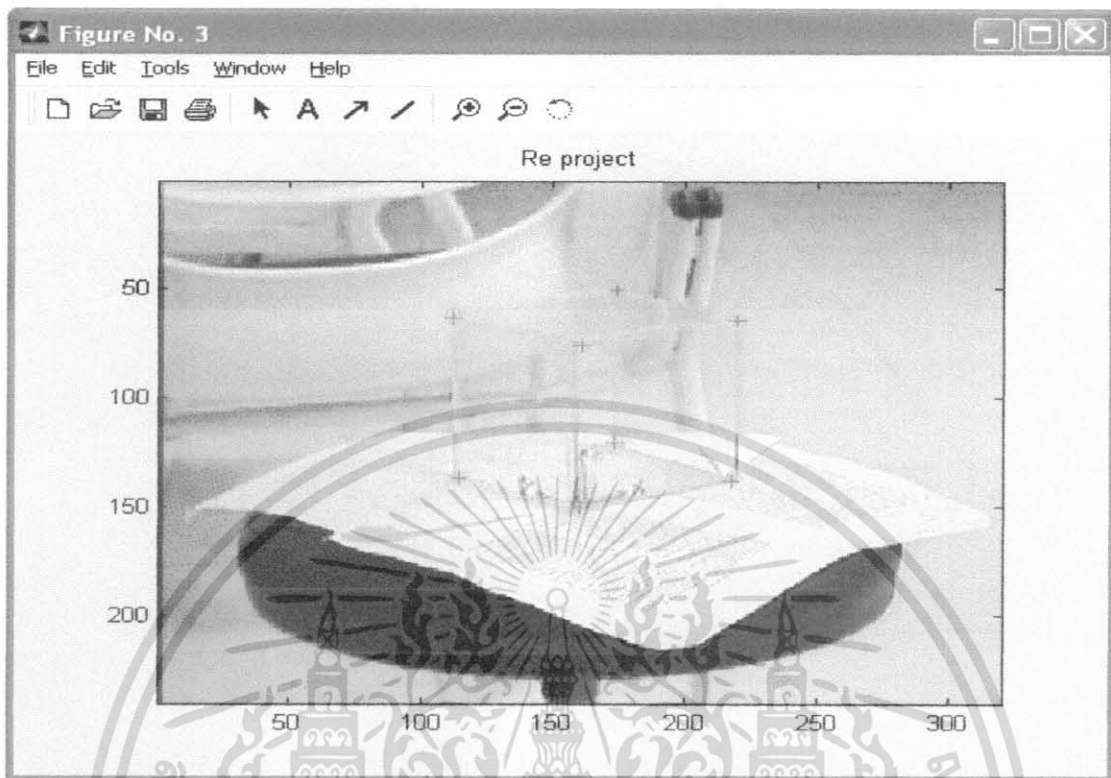


รูปที่ 3.3 แสดงภาพการ Calibration ของกล้องดีเลย์มในมุมกล้องที่1(m1)

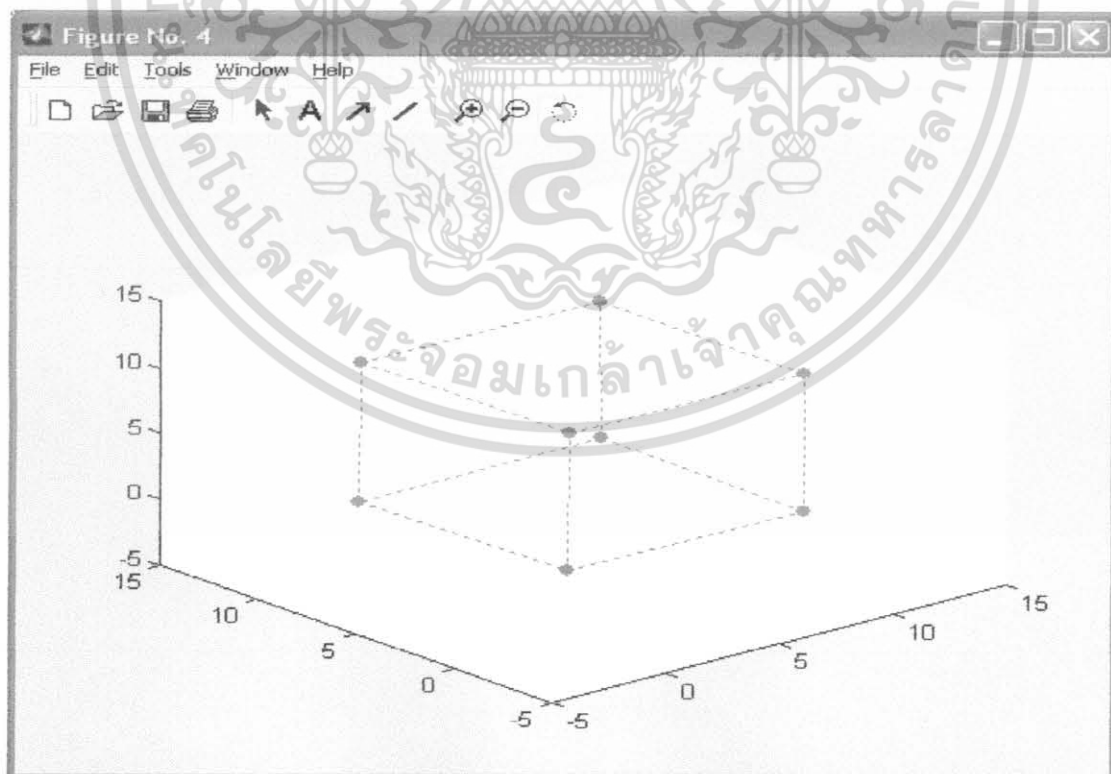


รูปที่ 3.4 แสดงภาพการ Calibration ของกล้องดีเลย์มในมุมกล้องที่2(m2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดงภาพการ Calibration ของกล้องสี่เหลี่ยมในมุมมองที่ 3(m3)



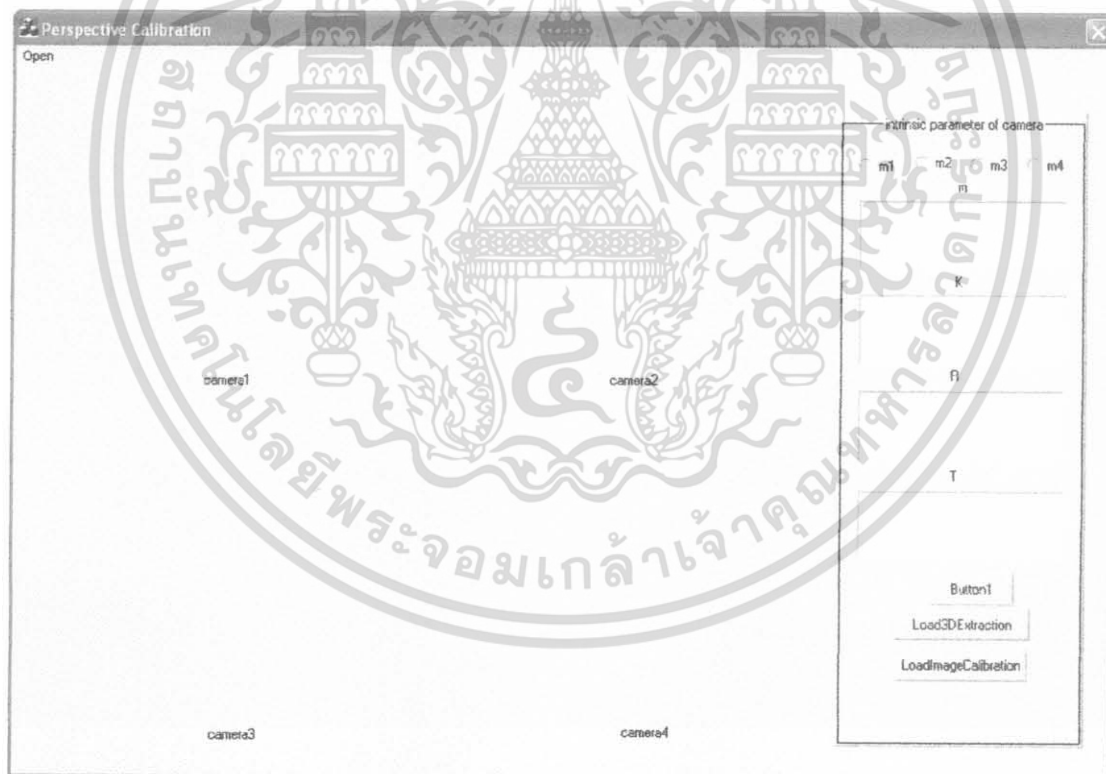
รูปที่ 3.6 แสดงภาพการPlot ค่าพิกัดจุด(u,v)ของกล้องสี่เหลี่ยมในรูป 3 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 โปรแกรม 3D coordinate extraction

ซึ่งใช้ในการวิเคราะห์และประมวลภาพ 3 มิติของวัตถุในส่วนของกรวิเคราะห์ภาพวัตถุที่ทำการเคลื่อนไหวจะทำการบันทึกภาพ โดยใช้กล้องทั้ง 4 กล้องวางในตำแหน่งที่เห็นทุกจุดพิกัดของวัตถุ สำหรับแต่ละกล้องจะใช้สมการ Calibration (2.32) ในการวิเคราะห์ภาพถ่ายของวัตถุ ซึ่งเราจะได้ค่าพารามิเตอร์ภายในภายนอกเพื่อนำมาพิจารณาการปรับเทียบโดยใช้กล้องหลายตัว

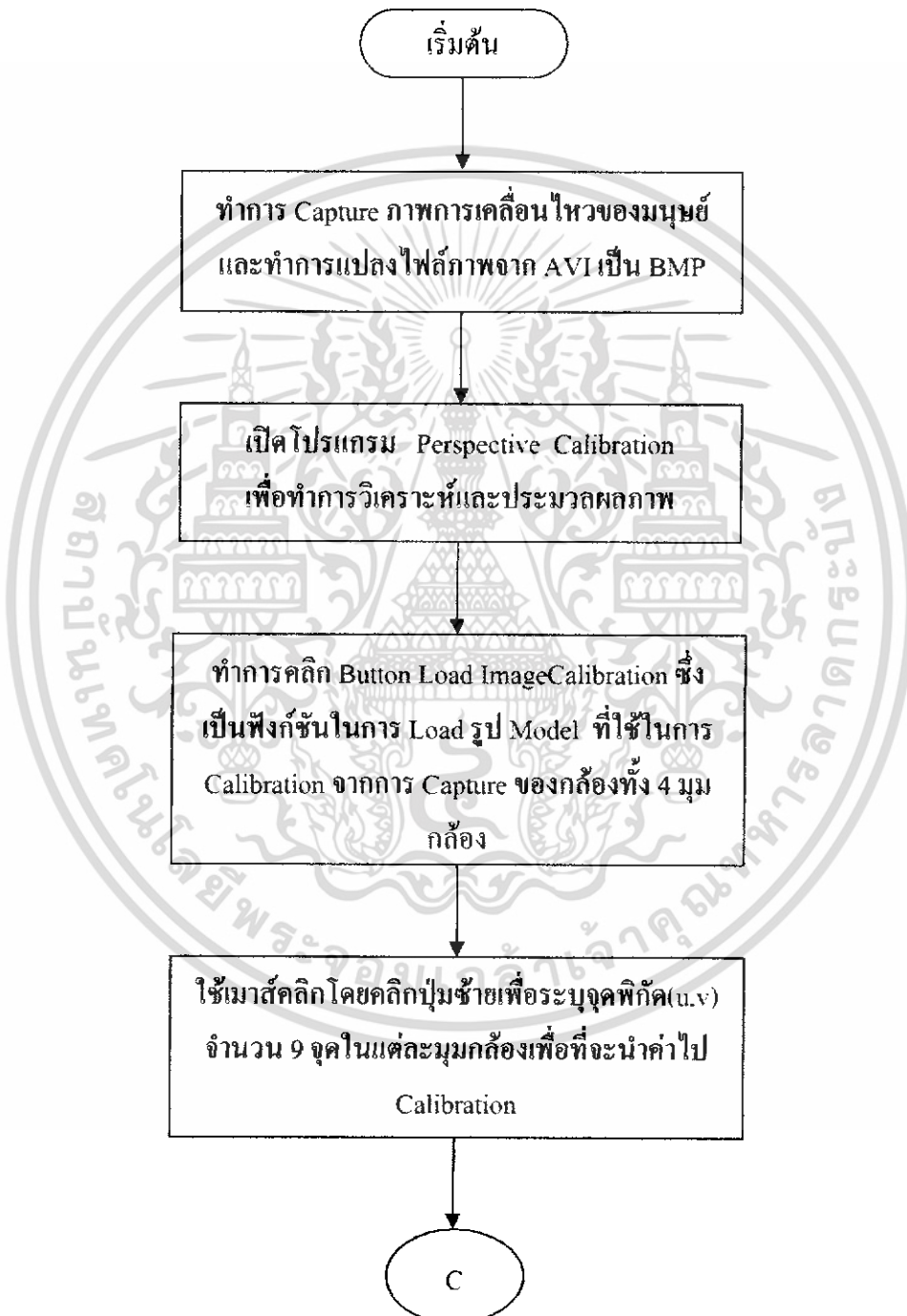
โดย  $m_1$ ,  $m_2$ ,  $m_3$  และ  $m_4$  คือการคำนวณ projection matrix สำหรับกล้องแต่ละกล้อง ในการบอกพิกัดตำแหน่งของร่างกายมนุษย์ในการเคลื่อนไหว จะใช้อุปกรณ์มาร์คเกอร์ติดที่ข้อต่อของร่างกายทั้งหมด 13 จุด ทำการแสดงผลภาพมนุษย์ที่ทำการ Capture ไว้แล้วทั้ง 4 มุมกล้องจากนั้นทำการหาตำแหน่งมาร์คเกอร์ 13 จุด จากนั้นนำพิกัดภาพมาเข้าสมการ(2.42) เพื่อไปหาพิกัดโลกในระบบ 3 มิติ จากนั้นนำผลพิกัดจุดที่ได้ไปสร้างภาพ 3 มิติซึ่งจะได้พิกัด(X,Y,Z)ที่สอดคล้องกับพิกัดจุดของแต่ละมาร์คเกอร์ในแต่ละกล้อง



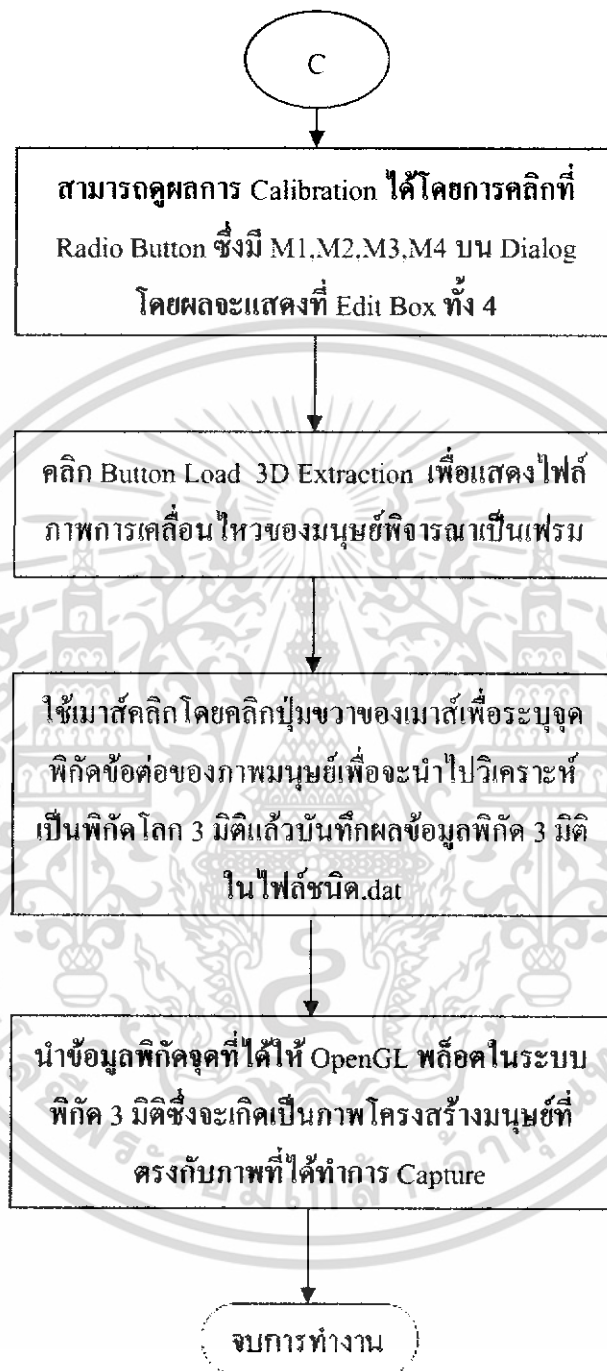
รูปที่ 3.7 แสดงคอมโพเนนต์โปรแกรมการปรับเทียบกล้องและการหาพิกัด 3 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.1 โครงสร้างโปรแกรม 3D coordinate extraction



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

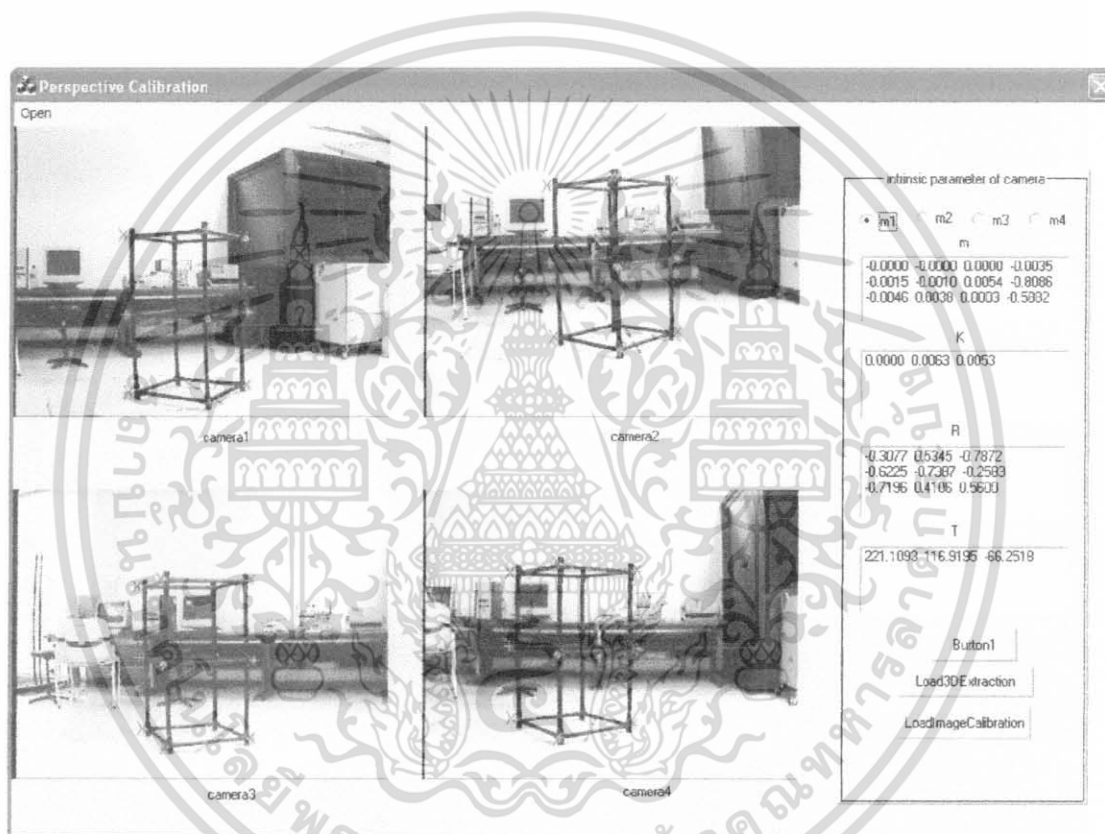


รูปที่ 3.8 Flow Chart โครงสร้างการทำงานของระบบโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 โปรแกรมการปรับเทียบกล้อง (Calibration)

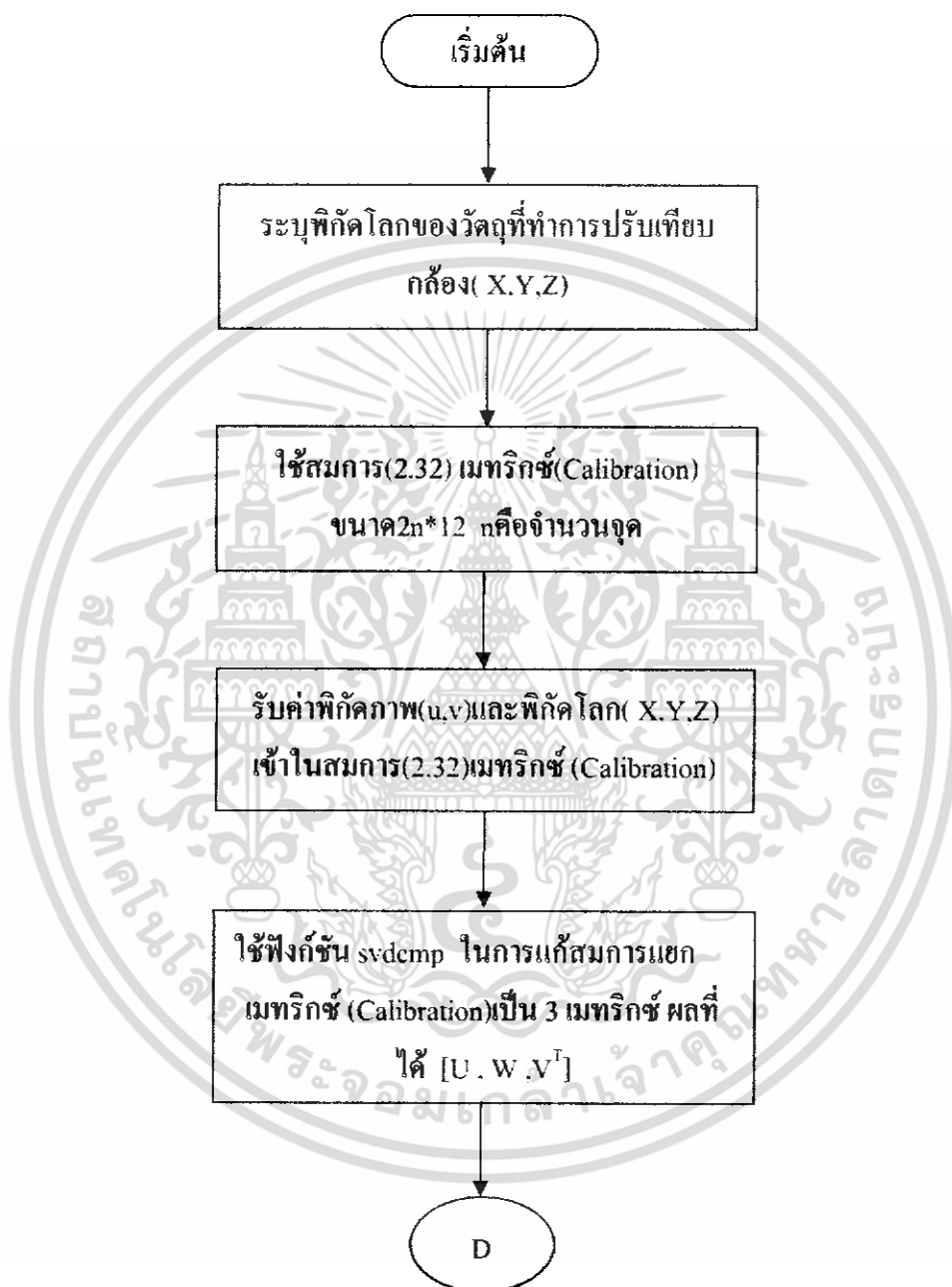
การปรับเทียบกล้องเป็นกระบวนการในการใช้กล้อง 4 ตัวในการ Capture ภาพนิ่งโดยอ้างอิงวัตถุในพิกัดโลกจุดที่อ้างอิงจะอ้างอิงเป็นกรณีพิเศษคือให้มุมใดมุมหนึ่ง เป็นจุดกำเนิด จากนั้นทำการตรวจสอบโดยการคำนวณ projection matrix  $M$  และทำการ reprojection (การคำนวณย้อนกลับไปหาภาพเดิม) สมการ projection matrix  $M$  (2.32) เพื่อทำการคำนวณจุดอ้างอิงของภาพและแสดงผลพิกัดซ้อนทับของภาพที่ทำการ Capture ดังรูปที่ 3.9



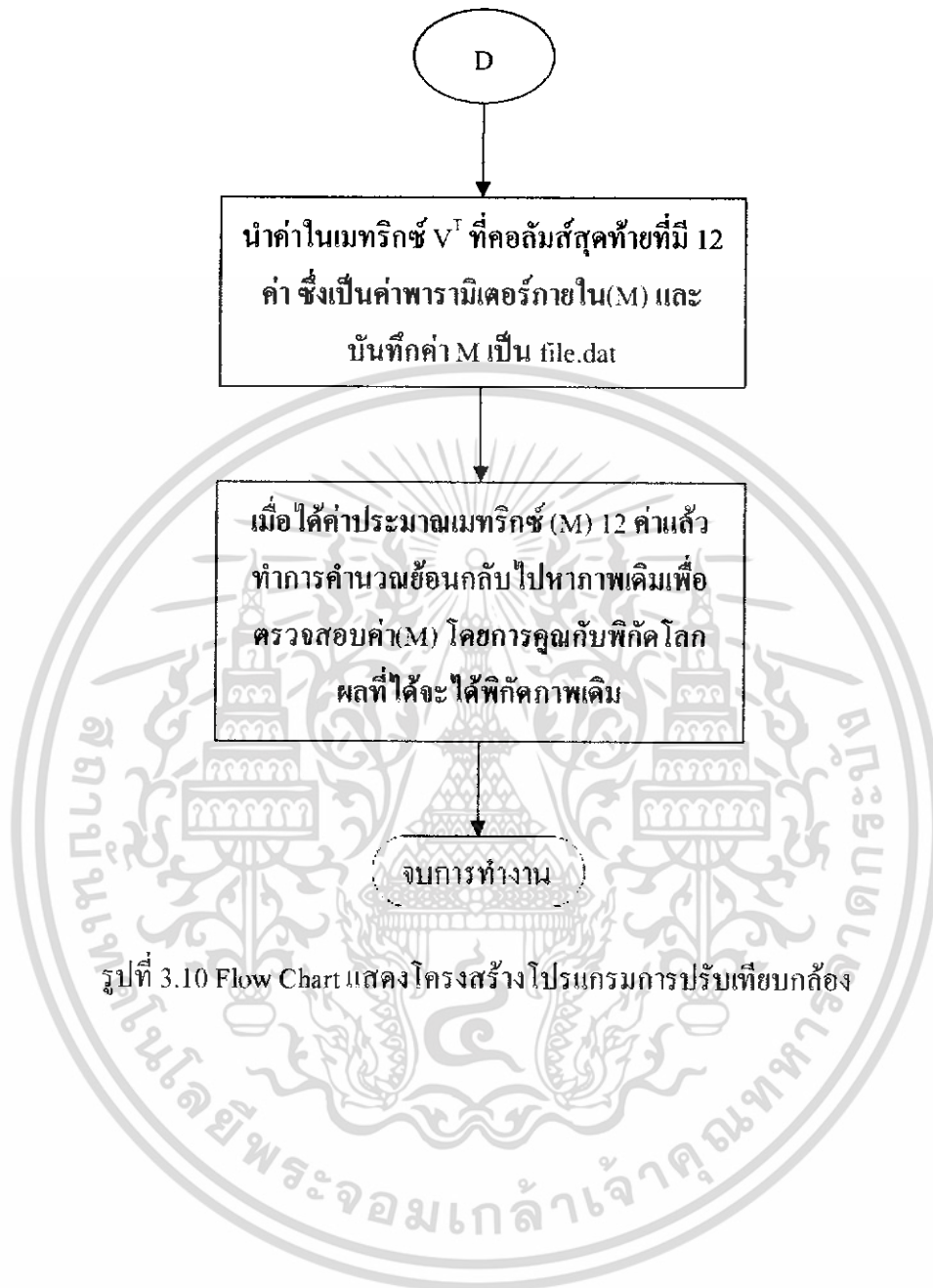
รูปที่ 3.9 การ Calibration โดยใช้ model ซึ่งเป็น โครงกล่องสี่เหลี่ยมลูกบาศก์ 2 ชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.1 โครงสร้างโปรแกรมการปรับเทียบกล้อง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.2 ขั้นตอนกระบวนการปรับเทียบกล้อง Calibration

หลังจากการทำ Capture ภาพแล้วจะนำพิกัดภาพใช้ model ซึ่งเป็นโครงกลองสี่เหลี่ยมลูกบาศก์ 2 ชั้น ที่ได้ไปทำการ Calibration ซึ่งเป็นกระบวนการนำพิกัดของโลกและพิกัดของภาพเข้ามาในเมทริกซ์ Calibration ขนาด  $2n \times 12$  หลังจากนั้นทำการแยกเมทริกซ์ออกโดยวิธีการ(SDV Singular Value decomposition) ผลลัพธ์ของเมทริกซ์ SDV ที่ถูกกระจายจะได้เป็น 3 เมทริกซ์คือ  $U$ ,  $W$ ,  $V^T$  ค่าการ Calibration (ค่า  $M$ ) จะอยู่ที่คอลัมส์สุดท้าย  $V^T$  ซึ่งจะได้ค่าเมทริกซ์  $M$  ขนาด  $3 \times 4$  จากนั้นนำเมทริกซ์  $M$  ทางด้านซ้ายมือซึ่งเป็นเมทริกซ์ขนาด  $3 \times 3$  มากระจายออกโดยใช้วิธีการ(QR decomposition)เพื่อแยกค่า Calibration ซึ่งจะได้ค่าเมทริกซ์  $Q$  และ  $R$  โดยค่าการ Calibration จะอยู่ที่เมทริกซ์  $R$  โดยค่าในเมทริกซ์  $R$  จะมีค่าพารามิเตอร์ที่จะนำไปใช้ในการหาพิกัด 3 มิติ

#### 1. การนำพิกัดของโลกและพิกัดของภาพเข้ามาในเมทริกซ์ Calibration ขนาด $2n \times 12$ ดังสมการ(2.32)

```

3DExtraction - Microsoft Visual C++ [design] - 3DExtractionDlg.cpp
File Edit View Project Build Debug Tools Window Help
Debug
3DExtractionDlg.cpp
void CMy3DExtractionDlg::OnLButtonDown(UINT nFlags, CPoint point)
{
    CDC *pdc=this->GetDC();
    CString str;
    CPen pen;
    pen.CreatePen(PS_SOLID,1,RGB(0,255,0));
    pdc->SelectObject(pen);
    int A=9; //rows of point
    n++;
    a[n]=point.x;
    b[n]=point.y;

    pdc->MoveTo(point.x-4,point.y);
    pdc->LineTo(point.x+5,point.y);
    pdc->MoveTo(point.x,point.y-4);
    pdc->LineTo(point.x,point.y+5);

    if (n==A){
        n=0;
        U=matrix(1,A,1,2);
        for(int j=1;j<=A;j++){
            U[j][1]=a[j]; U[j][2]=b[j]; //local point from x,y to u[j]
            Compute_m(U);
        }
    }

    int joy=0;
    FILE *pfile=NULL;
    float m1[5][6],m2[5][6],m3[5][6],m4[5][6],u1[15][4],u2[15][4],u3[15][4],u4[15][4];
    float** CMy3DExtractionDlg::Compute_m(float **u)
    {
        CDC *pdc=this->GetDC();
    }
}

```

รูปที่ 3.11 แสดงการเขียน โปรแกรมฟังก์ชัน LButton\_load\_uv

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Float** CMy3DExtrationDlg::Compute_m(Float**u)
{
    CDC *pdc=this->GetDC();
    CString str;
    CPen pen;
    pen.CreatePen(PS_SOLID,1,RGB(255,0,0)); //red pen
    pdc->SelectObject((CObject*)pen);
    float **xx,**yy,**z,**u,**v,**m,**p,**q,**r,**s,**t,**u;
    int i,k,l,j,nm=0,m=0;
    int A=9; //define size of 3D point
    xx=matrix(1,A,1,4);
    yy=matrix(1,9,1,4);
    z=matrix(1,2*A,1,12); //matrix of 3D point
    v=vector(1,12);
    Vmatrix(1,12,1,12);
    Wmatrix(1,3,1,A);
    m=matrix(1,3,1,9); //3x3 matrix
    u=matrix(1,4,1,A); //3x4 matrix
    Q=matrix(1,1,1,3); //3x1 matrix
    R=matrix(1,1,1,3); //3x1 matrix

    xx[0][4]=0;      xx[1][4]=0;      xx[2][4]=0;
    xx[0][5]=55;     xx[1][5]=0;      xx[2][5]=0;
    xx[0][6]=0;     xx[1][6]=0;      xx[2][6]=0;
    xx[0][7]=0;     xx[1][7]=55;     xx[2][7]=0;
    xx[0][8]=0;     xx[1][8]=0;      xx[2][8]=0;
    xx[0][9]=0;     xx[1][9]=0;      xx[2][9]=0;
    xx[0][10]=0;    xx[1][10]=0;     xx[2][10]=0;
    xx[0][11]=0;   xx[1][11]=0;     xx[2][11]=0;
    xx[0][12]=0;   xx[1][12]=0;     xx[2][12]=0;
    xx[0][13]=0;  xx[1][13]=0;     xx[2][13]=0;
    xx[0][14]=0;  xx[1][14]=0;     xx[2][14]=0;
    xx[0][15]=0;  xx[1][15]=0;     xx[2][15]=0;
    xx[0][16]=0;  xx[1][16]=0;     xx[2][16]=0;
    xx[0][17]=0;  xx[1][17]=0;     xx[2][17]=0;
    xx[0][18]=0;  xx[1][18]=0;     xx[2][18]=0;
    xx[0][19]=0;  xx[1][19]=0;     xx[2][19]=0;
    xx[0][20]=0;  xx[1][20]=0;     xx[2][20]=0;
    xx[0][21]=0;  xx[1][21]=0;     xx[2][21]=0;
    xx[0][22]=0;  xx[1][22]=0;     xx[2][22]=0;
    xx[0][23]=0;  xx[1][23]=0;     xx[2][23]=0;
    xx[0][24]=0;  xx[1][24]=0;     xx[2][24]=0;
    xx[0][25]=0;  xx[1][25]=0;     xx[2][25]=0;
    xx[0][26]=0;  xx[1][26]=0;     xx[2][26]=0;
    xx[0][27]=0;  xx[1][27]=0;     xx[2][27]=0;
    xx[0][28]=0;  xx[1][28]=0;     xx[2][28]=0;
    xx[0][29]=0;  xx[1][29]=0;     xx[2][29]=0;
    xx[0][30]=0;  xx[1][30]=0;     xx[2][30]=0;
    xx[0][31]=0;  xx[1][31]=0;     xx[2][31]=0;
    xx[0][32]=0;  xx[1][32]=0;     xx[2][32]=0;
    xx[0][33]=0;  xx[1][33]=0;     xx[2][33]=0;
    xx[0][34]=0;  xx[1][34]=0;     xx[2][34]=0;
    xx[0][35]=0;  xx[1][35]=0;     xx[2][35]=0;
    xx[0][36]=0;  xx[1][36]=0;     xx[2][36]=0;
    xx[0][37]=0;  xx[1][37]=0;     xx[2][37]=0;
    xx[0][38]=0;  xx[1][38]=0;     xx[2][38]=0;
    xx[0][39]=0;  xx[1][39]=0;     xx[2][39]=0;
    xx[0][40]=0;  xx[1][40]=0;     xx[2][40]=0;
    xx[0][41]=0;  xx[1][41]=0;     xx[2][41]=0;
    xx[0][42]=0;  xx[1][42]=0;     xx[2][42]=0;
    xx[0][43]=0;  xx[1][43]=0;     xx[2][43]=0;
    xx[0][44]=0;  xx[1][44]=0;     xx[2][44]=0;
    xx[0][45]=0;  xx[1][45]=0;     xx[2][45]=0;
    xx[0][46]=0;  xx[1][46]=0;     xx[2][46]=0;
    xx[0][47]=0;  xx[1][47]=0;     xx[2][47]=0;
    xx[0][48]=0;  xx[1][48]=0;     xx[2][48]=0;
    xx[0][49]=0;  xx[1][49]=0;     xx[2][49]=0;
    xx[0][50]=0;  xx[1][50]=0;     xx[2][50]=0;
    xx[0][51]=0;  xx[1][51]=0;     xx[2][51]=0;
    xx[0][52]=0;  xx[1][52]=0;     xx[2][52]=0;
    xx[0][53]=0;  xx[1][53]=0;     xx[2][53]=0;
    xx[0][54]=0;  xx[1][54]=0;     xx[2][54]=0;
    xx[0][55]=0;  xx[1][55]=0;     xx[2][55]=0;
    xx[0][56]=0;  xx[1][56]=0;     xx[2][56]=0;
    xx[0][57]=0;  xx[1][57]=0;     xx[2][57]=0;
    xx[0][58]=0;  xx[1][58]=0;     xx[2][58]=0;
    xx[0][59]=0;  xx[1][59]=0;     xx[2][59]=0;
    xx[0][60]=0;  xx[1][60]=0;     xx[2][60]=0;
    xx[0][61]=0;  xx[1][61]=0;     xx[2][61]=0;
    xx[0][62]=0;  xx[1][62]=0;     xx[2][62]=0;
    xx[0][63]=0;  xx[1][63]=0;     xx[2][63]=0;
    xx[0][64]=0;  xx[1][64]=0;     xx[2][64]=0;
    xx[0][65]=0;  xx[1][65]=0;     xx[2][65]=0;
    xx[0][66]=0;  xx[1][66]=0;     xx[2][66]=0;
    xx[0][67]=0;  xx[1][67]=0;     xx[2][67]=0;
    xx[0][68]=0;  xx[1][68]=0;     xx[2][68]=0;
    xx[0][69]=0;  xx[1][69]=0;     xx[2][69]=0;
    xx[0][70]=0;  xx[1][70]=0;     xx[2][70]=0;
    xx[0][71]=0;  xx[1][71]=0;     xx[2][71]=0;
    xx[0][72]=0;  xx[1][72]=0;     xx[2][72]=0;
    xx[0][73]=0;  xx[1][73]=0;     xx[2][73]=0;
    xx[0][74]=0;  xx[1][74]=0;     xx[2][74]=0;
    xx[0][75]=0;  xx[1][75]=0;     xx[2][75]=0;
    xx[0][76]=0;  xx[1][76]=0;     xx[2][76]=0;
    xx[0][77]=0;  xx[1][77]=0;     xx[2][77]=0;
    xx[0][78]=0;  xx[1][78]=0;     xx[2][78]=0;
    xx[0][79]=0;  xx[1][79]=0;     xx[2][79]=0;
    xx[0][80]=0;  xx[1][80]=0;     xx[2][80]=0;
    xx[0][81]=0;  xx[1][81]=0;     xx[2][81]=0;
    xx[0][82]=0;  xx[1][82]=0;     xx[2][82]=0;
    xx[0][83]=0;  xx[1][83]=0;     xx[2][83]=0;
    xx[0][84]=0;  xx[1][84]=0;     xx[2][84]=0;
    xx[0][85]=0;  xx[1][85]=0;     xx[2][85]=0;
    xx[0][86]=0;  xx[1][86]=0;     xx[2][86]=0;
    xx[0][87]=0;  xx[1][87]=0;     xx[2][86]=0;
    xx[0][88]=0;  xx[1][88]=0;     xx[2][86]=0;
    xx[0][89]=0;  xx[1][89]=0;     xx[2][86]=0;
    xx[0][90]=0;  xx[1][90]=0;     xx[2][86]=0;
    xx[0][91]=0;  xx[1][91]=0;     xx[2][86]=0;
    xx[0][92]=0;  xx[1][92]=0;     xx[2][86]=0;
    xx[0][93]=0;  xx[1][93]=0;     xx[2][86]=0;
    xx[0][94]=0;  xx[1][94]=0;     xx[2][86]=0;
    xx[0][95]=0;  xx[1][95]=0;     xx[2][86]=0;
    xx[0][96]=0;  xx[1][96]=0;     xx[2][86]=0;
    xx[0][97]=0;  xx[1][97]=0;     xx[2][86]=0;
    xx[0][98]=0;  xx[1][98]=0;     xx[2][86]=0;
    xx[0][99]=0;  xx[1][99]=0;     xx[2][86]=0;
    xx[0][100]=0; xx[1][100]=0;     xx[2][86]=0;
}

```

รูปที่ 3.12 แสดงการเขียนโปรแกรมเมทริกซ์ Calibration  
2. นำพิกัดภาพที่ได้จากการสแกนด้วยเมตริกซ์(คลิกขวา)มาสร้างเป็นพิกัด 3 มิติโลก

```

void CMy3DExtrationDlg::OnRButtonDown(UINT nFlags, CPoint point)
{
    CDC *pdc=this->GetDC();
    CString str;
    CPen pen;
    int j,A=13; //define size of 3D point
    m++;
    a[m]=point.x;
    b[m]=point.y;

    pen.CreatePen(PS_SOLID,1,RGB(0,0,255));
    pdc->SelectObject((CObject*)pen);

    pdc->MoveTo(point.x-4,point.y);
    pdc->LineTo(point.x+5,point.y);
    pdc->MoveTo(point.x,point.y-4);
    pdc->LineTo(point.x,point.y+5);

    if(m==A) {
        m++;
        switch(m) {
            case 1: { m=0;
                    U1=matrix(1,A,1,2);
                    for(j=1;j<=A;j++){ U1[j][1]=a[j]; U1[j][2]=b[j]; //load points from a,b to u1[j]
                    }break;
                }
            case 2: { m=0;
                    U2=matrix(1,A,1,2);
                }
        }
    }
}

```

รูปที่ 3.13 แสดงการเขียนโปรแกรมฟังก์ชัน RButton\_Down

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. การนำพิกัดภาพของวัตถุจากกล้องทั้ง 4 มุมกล้องไปคำนวณหาพิกัด 3 มิติโดยใช้สมการ(2.42)

```

void CMy3DExtractionDlg::Compute_3DExtraction(float **u1, float **u2, float **u3, float **u4)
{
    int min=0, mx=0, l, j, k, l, A=13; //define row column
    float m1[5][5], m2[5][5], m3[5][5], u4[5][5], **xx, **G, **v, **V, X;

    xx=matrix(1,4,1,A);
    G=matrix(1,8,1,4); //matrix G v v use dvd
    u=vector(1,12);
    v=matrix(1,4,1,4);

    pfile=fopen("C:\\file_m1.dat", "r");
    for(int b=1;b<=5;b++)
    for(int k=1;k<=4;k++){fscanf(pfile, "%f", &m1[b][k]); printf("%f ", m1[b][k]);
        printf("\n");
    }
    fclose(pfile);

    pfile=fopen("C:\\file_m2.dat", "r");
    for(b=1;b<=5;b++){
    for(k=1;k<=4;k++){fscanf(pfile, "%f", &m2[b][k]); printf("%f ", m2[b][k]);
        printf("\n");
    }
    fclose(pfile);

    pfile=fopen("C:\\file_m3.dat", "r");
    for(b=1;b<=5;b++){
    for(k=1;k<=4;k++){fscanf(pfile, "%f", &m3[b][k]); printf("%f ", m3[b][k]);
        printf("\n");
    }
}
    
```

รูปที่ 3.14 แสดงการเขียนโปรแกรมฟังก์ชัน Compute\_3DExtraction

### 4. นำค่าการ Calibration (ค่า M) จะอยู่ที่คอลัมสุดท้าย $V^T$ ซึ่งจะได้ค่าเมทริกซ์

M ขนาด  $3 \times 4$

$$M = \begin{pmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \end{pmatrix} \quad (4.2)$$

### 5. เลือกเมทริกซ์ M ทางด้านซ้ายมือซึ่งเป็นเมทริกซ์ขนาด $3 \times 3$ ออกมา

$$M = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix} \quad (4.3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. กระจายเมทริกซ์  $M$  ออกโดยใช้วิธีการ (QR decomposition) เพื่อแยกค่า Calibration ซึ่งค่าการ Calibration จะอยู่ที่เมทริกซ์  $R$  และ เมทริกซ์  $Q$  ก็คือ  $Q * Q^T = I$  ซึ่งเป็นเมทริกซ์แบบ Orthonormal โดยเมทริกซ์  $R$  จะมีค่าพารามิเตอร์ที่จะนำไปใช้ในการหาพิกัด 3 มิติดังนี้

$$R = \begin{pmatrix} fc(1) & \alpha_c * fc(1) & cc(1) \\ 0 & fc(2) & cc(2) \\ 0 & 0 & 1 \end{pmatrix} \quad (4.4)$$

#### ตัวแปรภายในกล้อง (Intrinsic camera parameters)

1. Focal Length :  $fc(1)$  และ  $fc(2)$  คือ ค่าความยาวโฟกัสในหน่วยพิกเซล
2. Principal point :  $cc(1)$  และ  $cc(2)$  คือ ค่าพิกัดจุดกึ่งกลางของภาพ
3. Distortion :  $\alpha_c$  คือ ค่าสัมประสิทธิ์ความความโค้งของมุมระหว่างแกนพิกเซล  $x$  และ  $y$  ที่เก็บไว้ในรูปขนาด  $\alpha_c$
4. Distirtion:  $kc$  คือ ใช้แบบจำลองทางความเพี้ยนในอัตราส่วนของกล้อง

#### ตัวแปรภายนอกกล้อง (Extrinsic camera parameters)

อธิบายเกี่ยวกับตำแหน่งและทิศทางของระบบกล้องในพิกัด 3 มิติของโลก รวมไปถึง

การเลื่อน (Translation)

$$t = [t_x, t_y, t_z]^T$$

การหมุน (Rotation)

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

float **QrC(float **ake) //QR
{
    int i,j,k,l,m,n,sing;
    float con,*c,*d,**q,**r,**x;

    c=vector(1.3);
    d=vector(1.3);
    q=matrix(1.3,1.3);
    qt=matrix(1.3,1.3);
    r=matrix(1.3,1.3);
    x=matrix(1.3,1.3);

    n=3;
    qrdcmp(ake,n,c,d,&sing);
    for(k=1;k<=n;k++){
        for(l=1;l<=n;l++){
            if(l>k){
                r[k][l]=ake[k][l];
                q[k][l]=0.0;
            }else if(l<k){
                r[k][l]=q[k][l]*0.0;
            }else {
                r[k][l]=d[k];
                q[k][l]=1.0;
            }
        }
    }
    for(i=n-1;i>=1;i--){
        for(con=0.0;k=i;k<=n;k++){
            con+=ake[k][i]*ake[k][i];
            con/=2.0;
            for(k=i;k<=n;k++){
                for(l=i;l<=n;l++){
                    qt[k][l]=0.0;
                    for(j=i;j<=n;j++){
                        qt[k][l]+=q[j][l]*ake[k][i]*ake[j][i]/con;
                    }
                }
            }
            for(k=i;k<=n;k++)
                for(l=i;l<=n;l++)q[k][l]=qt[k][l];
        }
    }
}

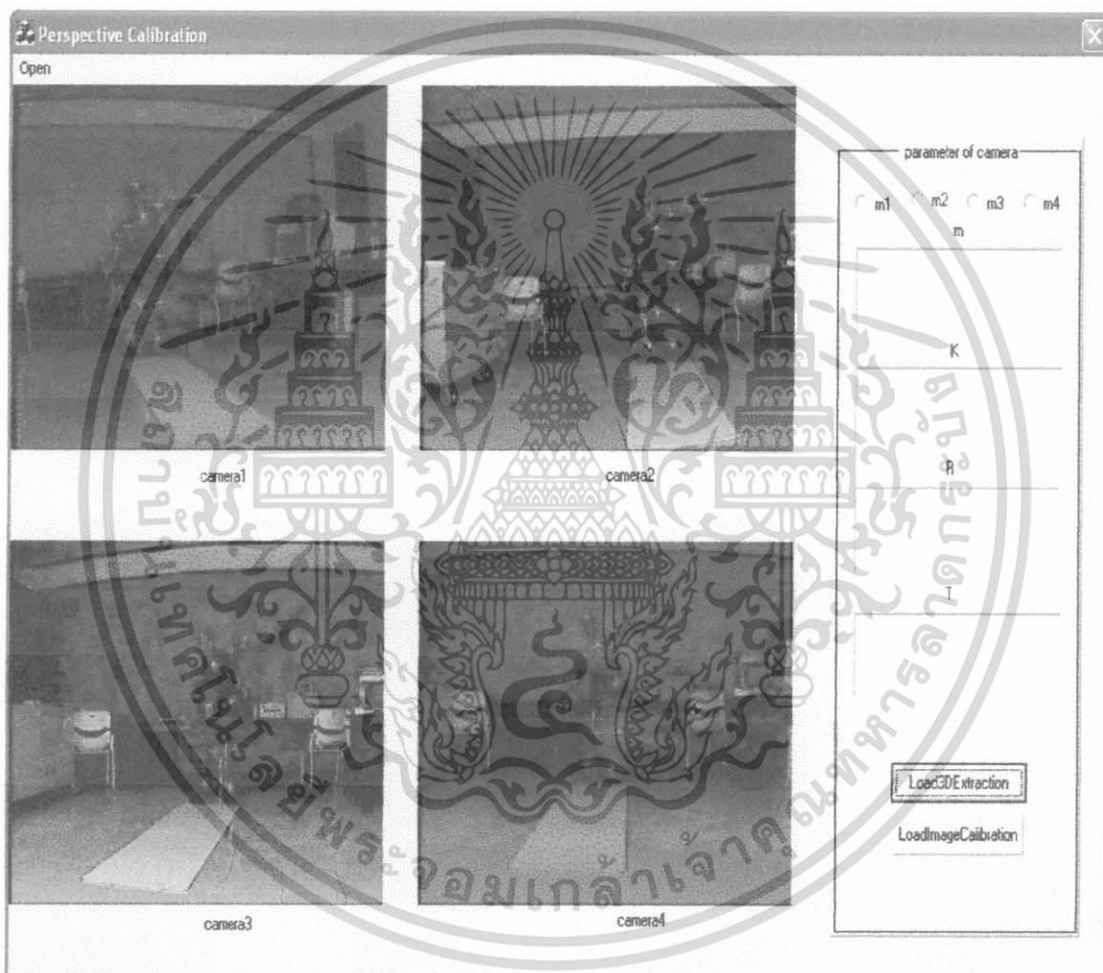
```

รูปที่ 3.15 แสดงการเขียน โปรแกรมฟังก์ชัน QRcmp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 โปรแกรมคำนวณหาพิกัด 3 มิติ(Compute\_3D Extration)

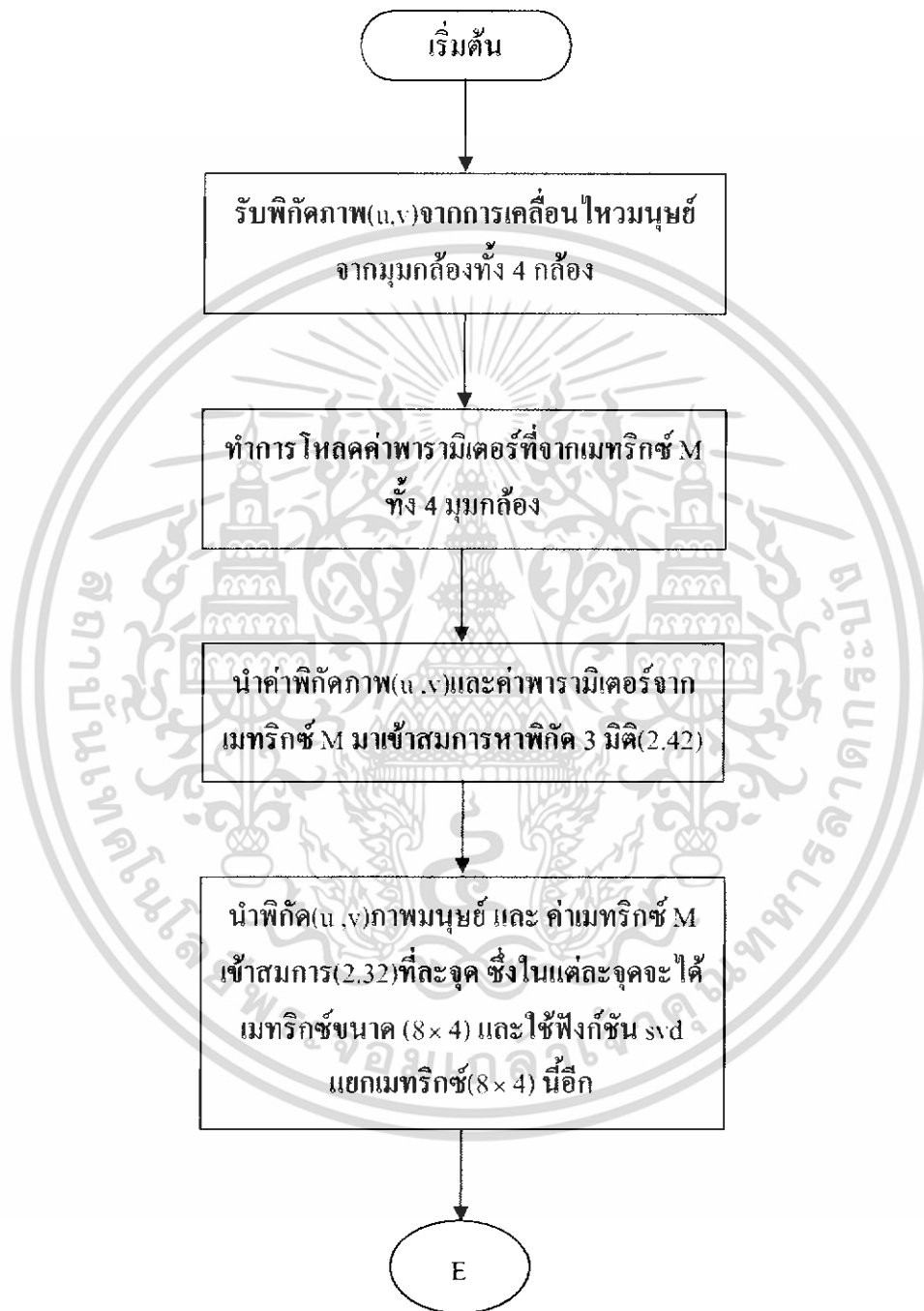
หลังจากทำการ Calibration โดยใช้ model จะได้ค่าเมทริกซ์  $M$  เพื่อนำไปใช้ในการแปลงพิกัดจาก 2 มิติเป็น 3 มิติโดยใช้สมการ(2.42) จากนั้นนำพิกัดข้อต่อมนุษย์(พิกัดภาพ)ที่ได้จากการสแกนด้วยเมตาส์ 13 จุด เพื่อนำข้อมูลจุดพิกัดภาพจากกล้องทั้ง 4 มุมกล้องไปสร้างภาพ 3 มิติ



รูปที่ 3.16 การกำหนดจุดพิกัดของภาพ(u, v) โดยการสแกนด้วยการคลิกเมตาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.1 โครงสร้างโปรแกรมคำนวณหาพิกัด 3 มิติ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5 โปรแกรมการสร้างภาพ 3 มิติโดย OpenGL

จากพิกัดที่ได้ทั้งหมด 13 จุดจากสมการ(2.42) เป็นพิกัดโลก 3 มิตินั้นจะส่งข้อมูลจุดพิกัดที่ได้ไปให้ OpenGL พล็อตในระบบพิกัด 3 มิติ จากนั้นจะทำการลากเส้นเชื่อมจุดพิกัดให้เป็นรูปโครงสร้างมนุษย์ทั้ง 13 จุด ซึ่งจะเกิดเป็นภาพโครงสร้างมนุษย์ที่ตรงกับการเคลื่อนไหวของมนุษย์ ซึ่งจะสอดคล้องกับภาพที่ทำการ Capture จากตัวมนุษย์ในแต่ละเฟรมนั้น



รูปที่ 3.18 ภาพ 3 มิติจากการพล็อตจุดใน OpenGL ให้เป็นโครงสร้างมนุษย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5.1 โครงสร้างโปรแกรมการสร้างภาพ 3 มิติ



รูปที่ 3.19 Flow Chart แสดงโครงสร้างโปรแกรมการสร้างภาพพิกัด 3 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.6 อุปกรณ์มาร์คเกอร์

#### 3.6.1 อุปกรณ์แสดงตำแหน่งจุดพิกัด 3 มิติ(มาร์คเกอร์)

ส่วนของการบันทึกภาพเคลื่อนไหวคือ การใช้กล้องจำนวน 3 ตัวหรือมากกว่า ซึ่งจะต้องวางในตำแหน่งที่เห็นมาร์คเกอร์ครบทุกจุดในฉาก กล้องจะทำการบันทึกภาพของวัตถุที่ติดมาร์คเกอร์ และทำการวิเคราะห์เพื่อให้ได้พิกัด 3 มิติของมาร์คเกอร์ ส่วนอุปกรณ์มาร์คเกอร์นั้นจากการทดลองเพื่อหาอุปกรณ์สะท้อนแสงที่เหมาะสมที่สุดมาใช้ โครงการนี้เราได้เลือกใช้ LED SUPERBITE ขนาด 10 มม. ต่อกับสายนำสัญญาณโดยปลายสายจะต่อกับแฉีกเสียบ RCA ซึ่งแสดงไว้ดังรูปที่ 3.20



รูปที่ 3.20 แสดงอุปกรณ์มาร์คเกอร์ที่ใช้ติดตามส่วนต่างๆ ของร่างกาย

ส่วนของการติดตั้งอุปกรณ์มาร์คเกอร์ตามส่วนต่างๆ ของร่างกายนั้นการติดตั้งไหนขึ้นอยู่กับว่าเราต้องการตรวจจับตรงไหนบ้าง จากโครงการนี้เราจะทำการติดมาร์คเกอร์ตรงส่วนสำคัญหลักๆ เท่านั้น ซึ่งในการจับภาพจะสามารถจับในท่าทางของวัตถุที่หยุดนิ่งและ เคลื่อนไหวได้ โดยใช้กล้องทั้งหมด 4 ตัว ดังนั้นไม่ว่าการเคลื่อนไหวในท่าทางการเดิน การกระโดดก็ จะสามารถตรวจจับจุดพิกัด 3 มิติ ของมาร์คเกอร์ได้เนื่องมาจากการใช้กล้องหลายๆ ตัวในการจับภาพ รูปที่ 4.21 แสดงรูปตำแหน่งการติดอุปกรณ์มาร์คเกอร์บนร่างกายมนุษย์ในการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 รูปตำแหน่งการติดอุปกรณ์มาร์คเกอร์บนร่างกายมนุษย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

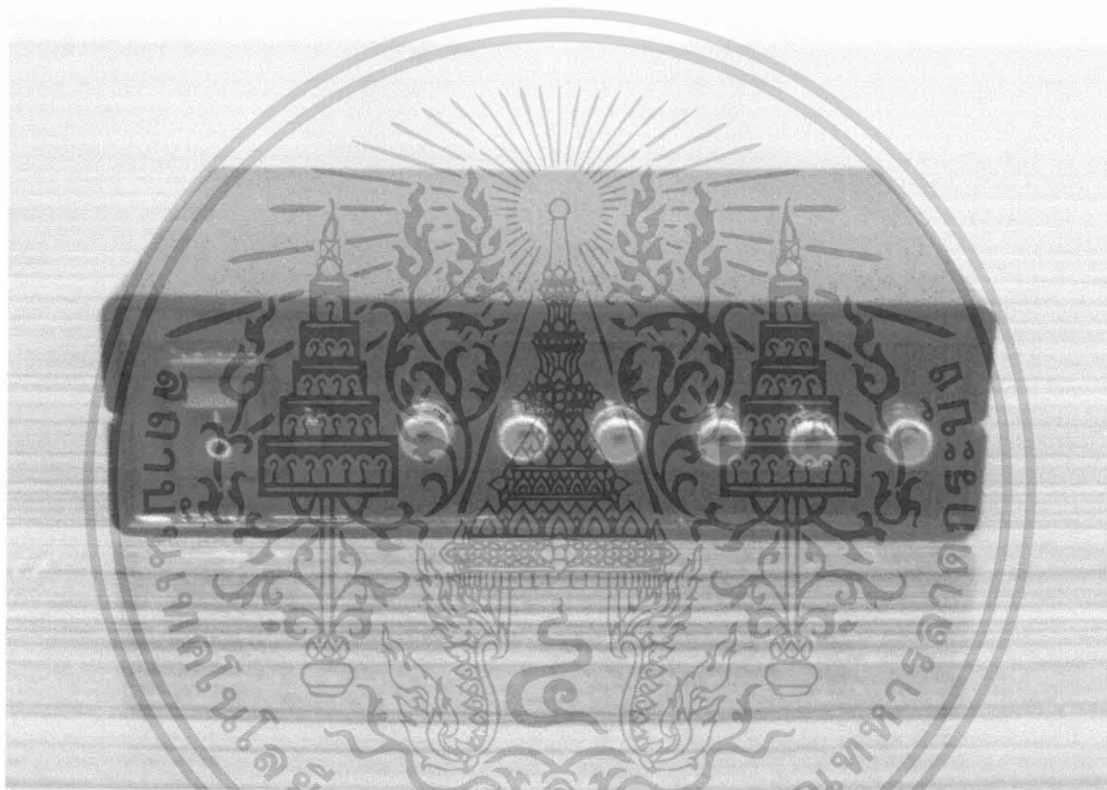


รูปที่ 3.22 ตำแหน่งการวางกล้องทั้ง 4 ตัวในการบันทึกภาพการเคลื่อนไหวของมนุษย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

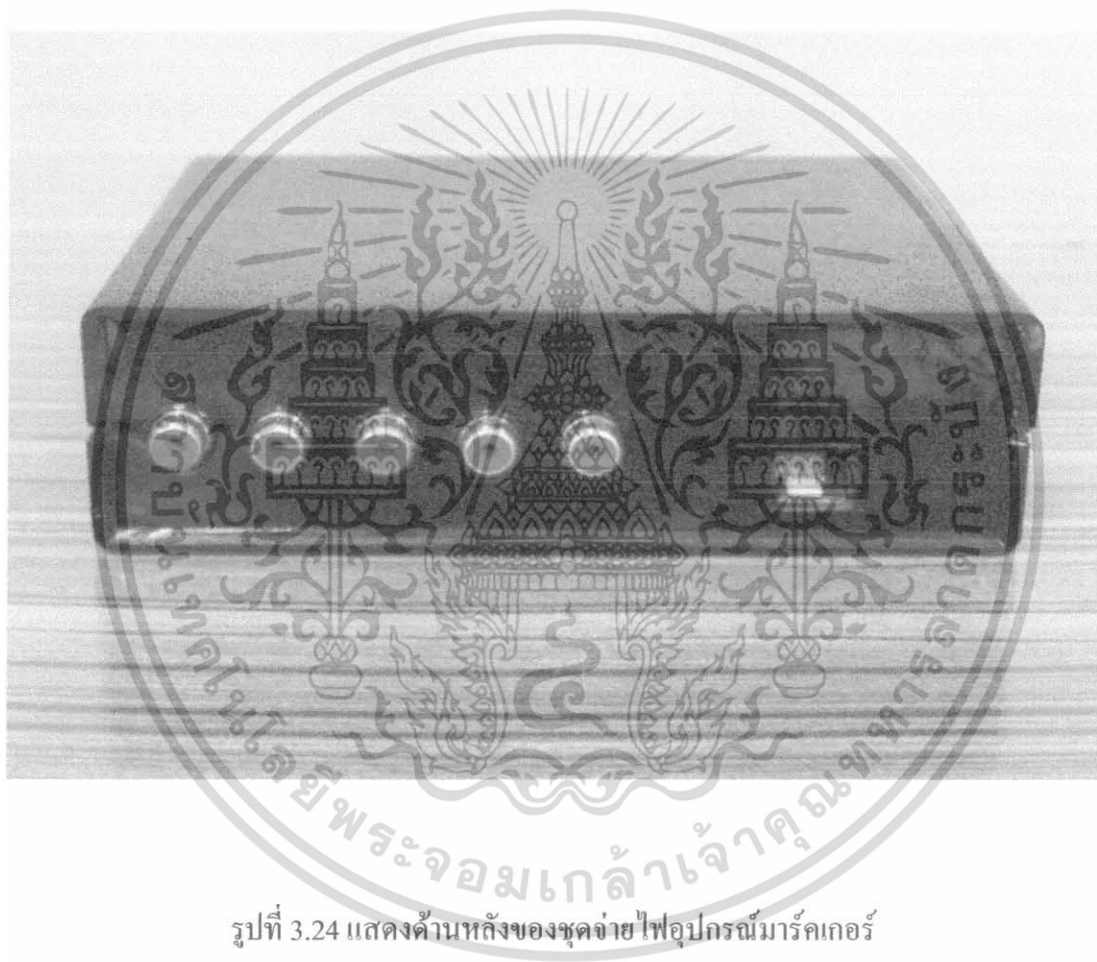
### 3.6.2 ชุดจ่ายไฟอุปกรณ์มาร์คเกอร์

ชุดจ่ายไฟอุปกรณ์มาร์คเกอร์จะทำหน้าที่ในการจ่ายไฟให้กับมาร์คเกอร์เพื่อให้เกิดการเปล่งแสงของLED SUPERBITE ขนาด 10 มม. โดยต่อกับสายนำสัญญาณ โดยปลายสายจะต่อกับแจ็คเสียบ RCA ที่จะนำมาต่อกับกล่องชุดจ่ายไฟเมื่อทำการเปิดสวิทช์ มาร์คเกอร์จะเปล่งแสงทั้งหมด 13 จุดพร้อมกันเพื่อแสดงตำแหน่งส่วนต่างๆของร่างกายมนุษย์



รูปที่ 3.23 แสดงด้านหน้าของชุดจ่ายไฟอุปกรณ์มาร์คเกอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 แสดงด้านหลังของชุดจ่ายไฟอุปกรณ์มาร์คเกอร์

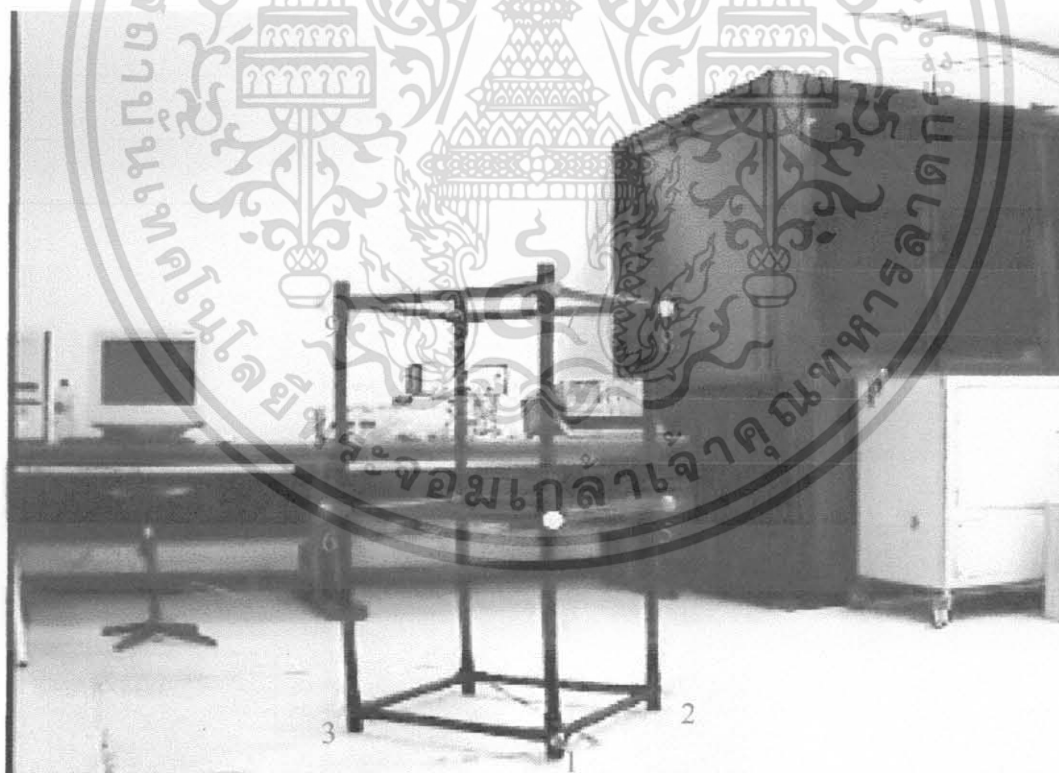
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การทดลองและผลการทดลอง

#### 4.1 การทดลองโปรแกรมการ Calibration กล้อง

การทดลองนี้เราจะใช้โปรแกรมที่ทำการเขียนขึ้นมาทำการ Calibration กล้องทั้งหมด 4 ตัว โดยทำการใช้ model ซึ่งเป็นโครงกล่องสี่เหลี่ยมลูกบาศก์ 2 ชั้น ที่มีความกว้างความยาวและความสูงขนาด 50 cm ซึ่งการทดลองโปรแกรมการ Calibration กล้องนี้คือ การปรับเทียบกล้องเป็นการหาประมาณเมทริกซ์ของ Perspective Transformation  $M$  สำหรับกล้องแต่ละตัวจากจุดในฉาก 3 มิติที่รู้ตำแหน่งและภาพที่เกิดขึ้นของจุดโดยใช้สมการ(2.32) จากนั้นทำการหาพารามิเตอร์ภายในและพารามิเตอร์ภายนอก



รูปที่ 4.1 แสดงภาพ model โครงกล่องสี่เหลี่ยมลูกบาศก์ที่ทำจากท่อ PVC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.1 ขั้นตอนการทดลองมีดังต่อไปนี้

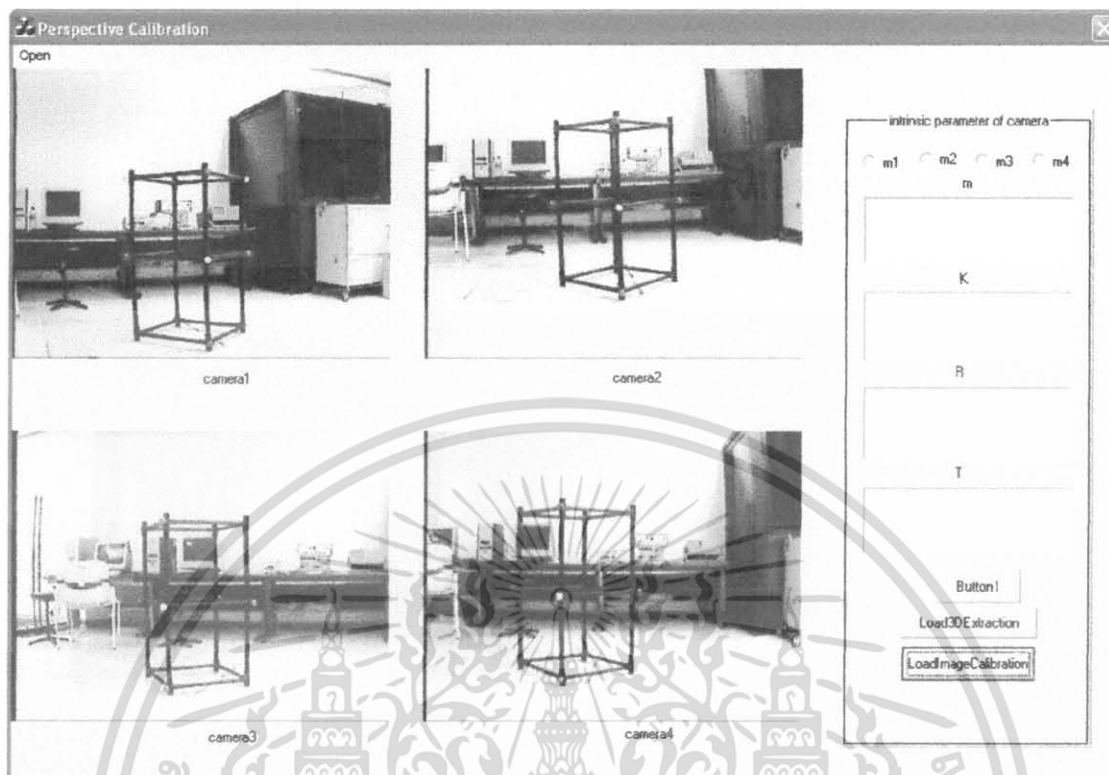
1. ทำการ Capture ภาพ model จากกล้อง CCD ทั้ง 4 ตัวพร้อมกันอย่าให้กล้องขยับ
2. แปลงไฟล์ภาพจาก AVI เป็น BMP และ save ไฟล์ลงที่ไดร์ฟที่ต้องการ
3. จากนั้นทำการเปิดโปรแกรม Perspective Calibration จากโปรแกรม application ที่ได้ออกแบบไว้ใน Visual C++ ดังรูปที่ 5.2



รูปที่ 4.2 แสดงโปรแกรม Perspective Calibration

4. ทำการคลิกที่ปุ่ม Loadimage Calibration โปรแกรมจะแสดงไฟล์ภาพ model ที่ทำการ Capture ของแต่ละมุมกล้องบนโปรแกรมดังรูปที่ 5.3

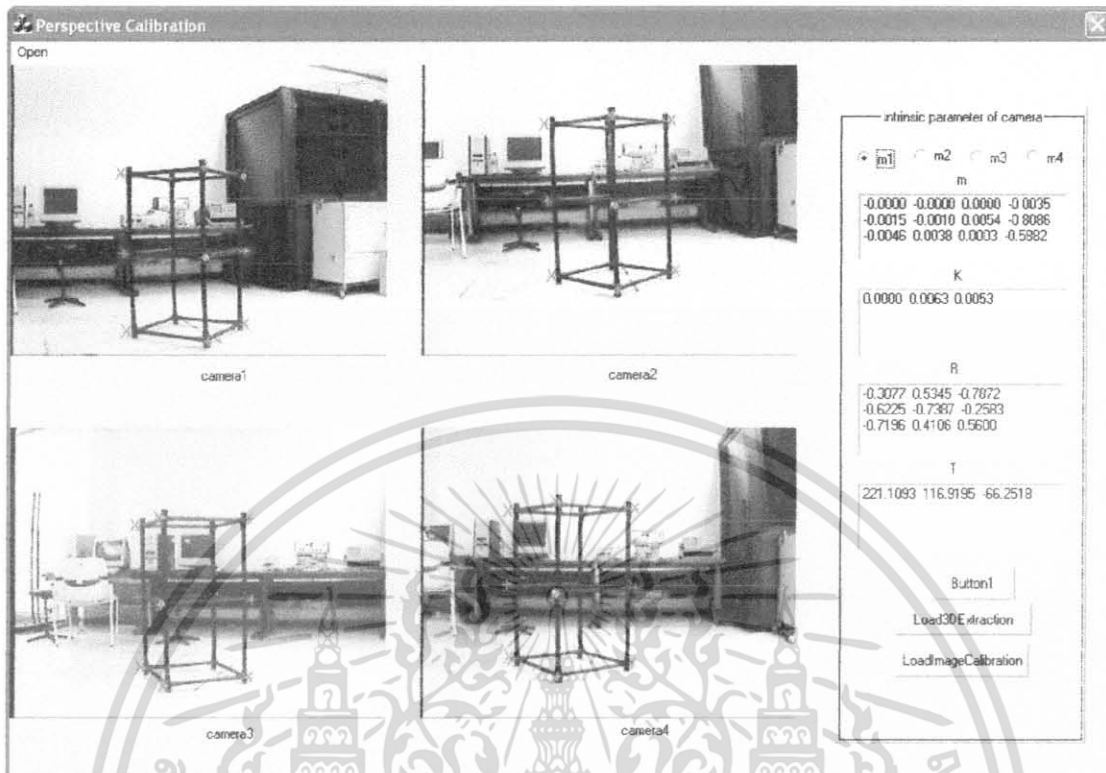
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



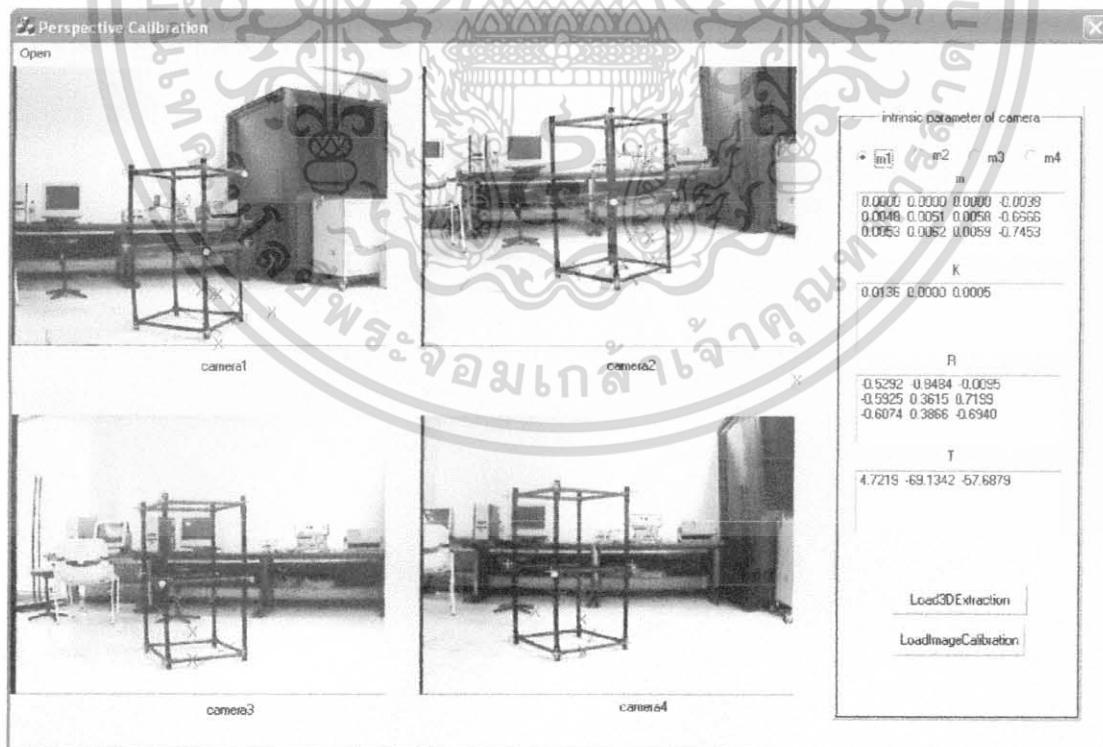
รูปที่ 4.3 แสดง ไฟล์ภาพ model ที่ทำการCapture ของแต่ละมุมกล้องบนโปรแกรม

5. ต่อจากนั้นให้ใช้เมาส์คลิกจุดอะตอมที่อยู่ตามมุมของ model โดยคลิกให้ตรงจุดตรงกลางของอะตอมให้ได้มากที่สุด ซึ่งสามารถมองเห็นได้จากภาพที่ทำการCapture โดยจะเป็นรูป สีเขียว 9 จุดเพื่อจะเอาพิกัดภาพ  $(u, v)$  ไปคำนวณในสมการCalibration
6. ในการคลิกเพื่อกำหนดจุดพิกัดจะต้องคลิกตามจุดที่ทำการกำหนดไว้ในการออกแบบโปรแกรมตามรูปที่ 5.1 เพื่อให้ตรงกับกรคำนวณย้อนกลับไปหาภาพเดิม(reproject)ถ้าทำการคลิกไม่ตรงตามจุดจะเกิดการ Error Calibration ขึ้น
7. เมื่อคลิกจุดอะตอมที่อยู่ตามมุมของ model ครบ 9 จุดแล้ว โปรแกรมจะทำการCalibration โดยอัตโนมัติผลที่ได้จะมีภาพ  $\times$  สีแดงเกิดขึ้นเป็นค่า  $(u, v)$  จากการคำนวณย้อนกลับทับบนจุด  $\times$  สีเขียวครบ 9 จุดพอดีซึ่งแสดงว่าการ Calibration ถูกต้องตามรูปที่ 5.4
8. กรณีที่ทำการคลิกไม่ตรงตามจุดพิกัดที่กำหนดไว้ในการออกแบบโปรแกรมจะทำให้เกิดการ Error Calibration ดังรูปที่ 5.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงผลการ Calibration โดยใช้ model ซึ่งเป็นโครงกลองสี่เหลี่ยมลูกบาศก์ 2 ชั้น



รูปที่ 4.5 แสดงผลการ Error Calibration โดยใช้ model ซึ่งเป็นโครงกลองสี่เหลี่ยมลูกบาศก์ 2 ชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.2 ผลการทดลองการ Calibration โดยกำหนดจุดพิกัดตามจุดที่กำหนด

กล้อง ที่	M	K	R	T
1	-0.0000 -0.0000 -0.0000 -0.0035 -0.0015 -0.0010 0.0054 -0.8086 -0.0046 0.0038 0.0003 -0.5882	0.0000 0.0063 0.0053	-0.3077 0.5345 -0.7872 -0.6225 -0.7387 -0.2583 -0.7196 0.4106 0.5600	221.1093 116.9195 -66.2518
2	-0.0000 -0.0000 0.0000 -0.0018 -0.0004 -0.0004 0.0027 -0.3376 -0.0048 -0.0003 0.0004 -0.9413	0.0049 0.0027 0.0000	0.0803 0.1402 0.9869 -0.1886 -0.9700 0.1532 0.9788 -0.1985 -0.0515	-20.9535 401.6131 62.3444
3	-0.0000 -0.0000 0.0000 -0.0019 -0.0008 -0.0027 0.0030 -0.9690 -0.0031 -0.0000 0.0002 -0.2470	0.0043 0.0029 0.0000	0.4238 0.9052 0.0326 0.6030 0.3088 0.7356 0.6759 -0.2920 -0.6767	-13.5704 306.0577 281.5449
4	-0.0000 -0.0000 0.0000 -0.0015 -0.0015 -0.0021 0.0024 -0.7347 -0.0037 -0.0011 0.0003 -0.6784	0.0047 0.0022 0.0000	-0.3883 0.7234 0.5708 0.4676 0.3791 0.7985 0.7941 -0.5770 -0.1911	-104.5443 436.8962 312.3036

ตารางที่ 4.1 ตารางแสดงค่าพารามิเตอร์ภายในและพารามิเตอร์ภายนอกของกล้องทั้ง 4 ตัว  
กรณี Calibration ที่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.1.3 ผลการทดลองการ Error Calibration โดยกำหนดจุดพิกัดไม่ตรงตามที่กำหนด

กล้อง ที่	M	K	R	T
1	0.0000 0.0000 0.0000 -0.0038 0.0048 0.0051 0.0058 -0.6666 0.0053 0.0062 0.0059 -0.7453	0.0136 0.0000 0.0005	-0.5292 -0.5345 -0.0095 -0.5925 -0.7387 -0.7199 -0.6074 0.3866 0.6940	4.7219 -69.1342 -57.6879
2	0.0000 0.0000 0.0000 -0.0018 0.0011 0.0025 0.0031 -0.2652 0.0055 0.0068 0.0109 -0.9641	0.0145 0.0007 0.0000	-0.3820 0.6803 -0.6255 -0.4957 -0.7221 -0.4825 -0.7799 0.1258 0.6131	-1493.3582 -1151.9739 1463.6940
3	-0.0000 -0.0000 -0.0000 0.0020 -0.0014 -0.0015 -0.0030 0.3046 -0.0039 -0.0059 -0.0088 0.9524	0.0119 0.0000 0.0004	0.3820 -0.8848 0.3094 0.5115 -0.0971 -0.8538 0.7854 0.4558 0.4187	1236.7062 135.7542 -637.1705
4	0.0000 0.0000 0.0000 -0.0015 0.0031 0.0043 0.0072 -0.7172 0.0032 0.0047 0.0066 -0.6967	0.0125 0.0000 0.0005	-0.3575 -0.9159 -0.1822 -0.5097 0.3549 -0.7837 -0.7825 0.1874 0.5938	2638.7583 -1022.3400 -539.7660

ตารางที่ 4.2 ตารางแสดงค่าพารามิเตอร์ภายในและพารามิเตอร์ภายนอกของกล้องทั้ง 4 ตัว  
กรณี Calibration ไม่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2 การทดลองหาพิกัด 3 มิติของวัตถุทรงเรขาคณิต

ในการทดลองนี้เป็นการทดสอบโปรแกรมการหาพิกัด 3 มิติในการหาค่าพิกัดแกน (X, Y, Z) ของวัตถุทรงเรขาคณิตและมาคำนวณหาปริมาตรของวัตถุว่ามีความถูกต้องมากน้อยเพียงใดซึ่งเป็นการหาพิกัดจุด 3 มิติของวัตถุที่ไม่มีการเคลื่อนไหว

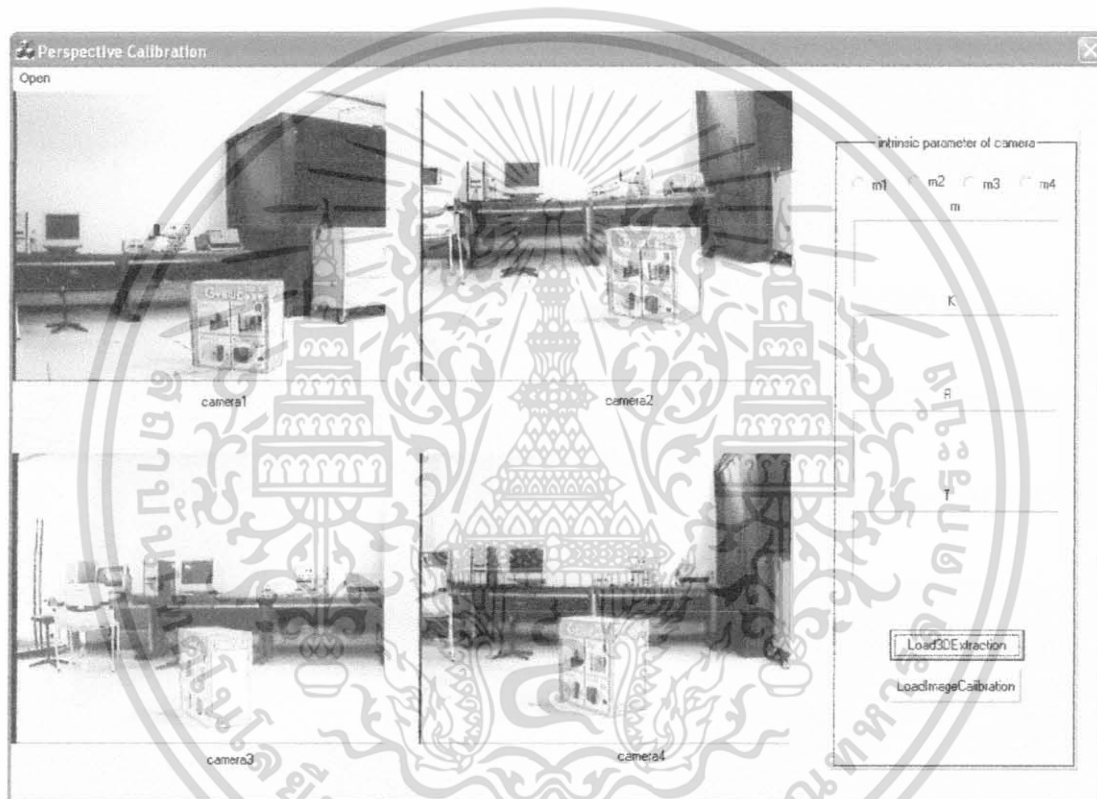


รูปที่ 4.6 แสดงตำแหน่งจุดในคลิกเมาส์เพื่อระบุพิกัดภาพของวัตถุแบบที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2.1 ขั้นตอนการทดลองการหาพิกัด 3 มิติของวัตถุทรงเรขาคณิตมีดังต่อไปนี้

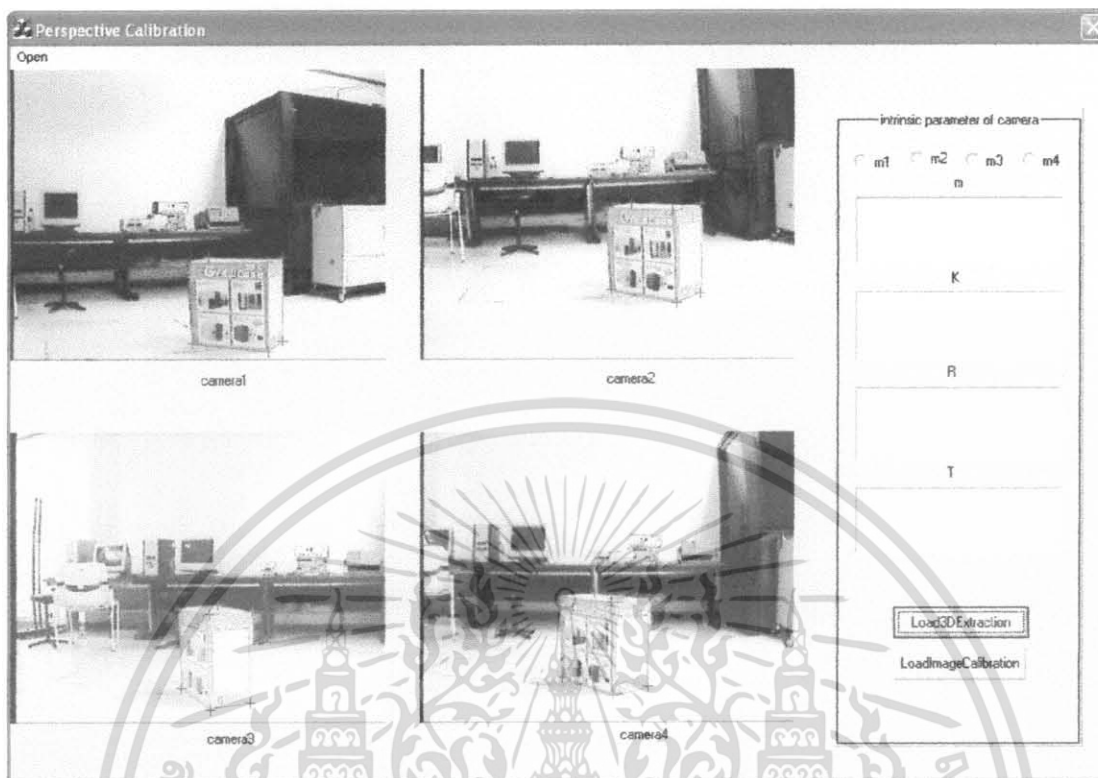
1. หลังจากทำการ Capture ภาพ model แล้วอย่าเพิ่งขยับกล้อง จากนั้นให้เปลี่ยนวัตถุจาก model เป็นวัตถุที่ต้องการหาพิกัด 3 มิติและทำการ Capture อีกครั้ง
2. ทำการแปลงไฟล์ที่ทำการ Capture ภาพจาก AVI เป็น BMP และ save ไฟล์ลงที่ไดรฟ์ที่ต้องการ
3. ทำการคลิกปุ่ม Load3DExtraction จะปรากฏรูปวัตถุที่ทำการ Capture ไว้ดังรูปที่ 4.7



รูปที่ 4.7 แสดงรูปวัตถุที่ต้องการหาพิกัด 3 มิติทั้ง 4 มุมกล้อง

4. จากนั้นทำการใช้เมาส์คลิก (คลิกขวา) ตรงมุมกล้องให้ครบทุกมุมตามที่มองเห็นจากภาพ จากกล้องที่ 1 โดยในการคลิกจะคลิกที่มุมใดก่อนก็ได้ให้ครบ 7 จุด
5. ทำการคลิกลำดับพิกัดจุดในภาพของกล้องที่ 2, 3 และ 4 ให้เหมือนกับกล้องที่ 1 ดังรูปที่ 4.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 แสดงภาพการคลิกตำแหน่งพิกัดภาพของวัตถุที่ 1 ด้วยเมาส์ทั้ง 4 มุมกล้อง  
 6. หลังจากทำการคลิกครบ 7 จุดทั้ง 4 มุมกล้องแล้วค่าพิกัด (X,Y, Z) จะถูกบันทึกไว้ที่ file.dat ดังรูปที่ 4.9



รูปที่ 4.9 แสดงค่าพิกัดจุด(X,Y,Z) ของจุดทั้ง 7 จุดของวัตถุแบบที่ 1

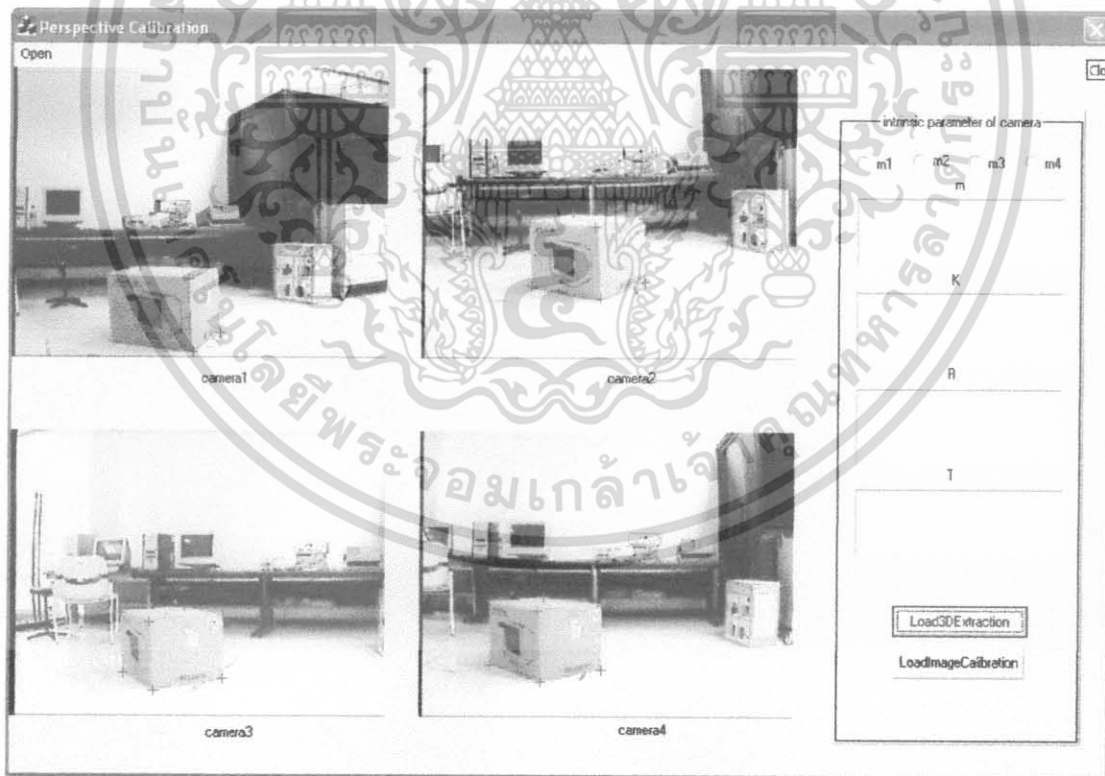
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2.2 ผลการทดลองประสิทธิภาพการหาปริมาตรของวัตถุแบบที่ 1

	ค่าจริง	ค่าจากการทดลอง	%ความถูกต้อง
ความกว้าง (X)	24 cm	23.28 cm	3
ความยาว (Y)	47 cm	46.83 cm	0.3
ความสูง (Z)	51.5 cm	50.9 cm	1.16
ปริมาตร(V)	58,092 cm <sup>3</sup>	55,557 cm <sup>3</sup>	4.36

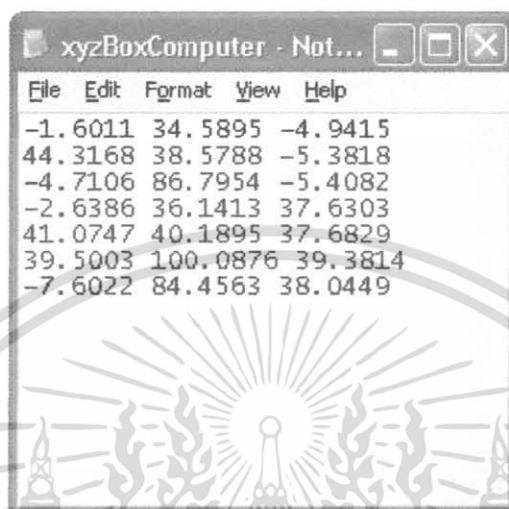
ตารางที่ 4.3 ตารางแสดงประสิทธิภาพการหาปริมาตรของวัตถุแบบที่ 1

#### วัตถุทรงเรขาคณิตแบบที่ 2



รูปที่ 4.10 แสดงภาพการคลิกตำแหน่งพิกัดภาพของวัตถุแบบที่ 2 ด้วยเมาส์ทั้ง 4 มุมกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



X	Y	Z
-1.6011	34.5895	-4.9415
44.3168	38.5788	-5.3818
-4.7106	86.7954	-5.4082
-2.6386	36.1413	37.6303
41.0747	40.1895	37.6829
39.5003	100.0876	39.3814
-7.6022	84.4563	38.0449

รูปที่ 4.11 แสดงค่าพิกัดจุด(X,Y,Z) ของจุดทั้ง 7 จุดของวัตถุแบบที่ 2

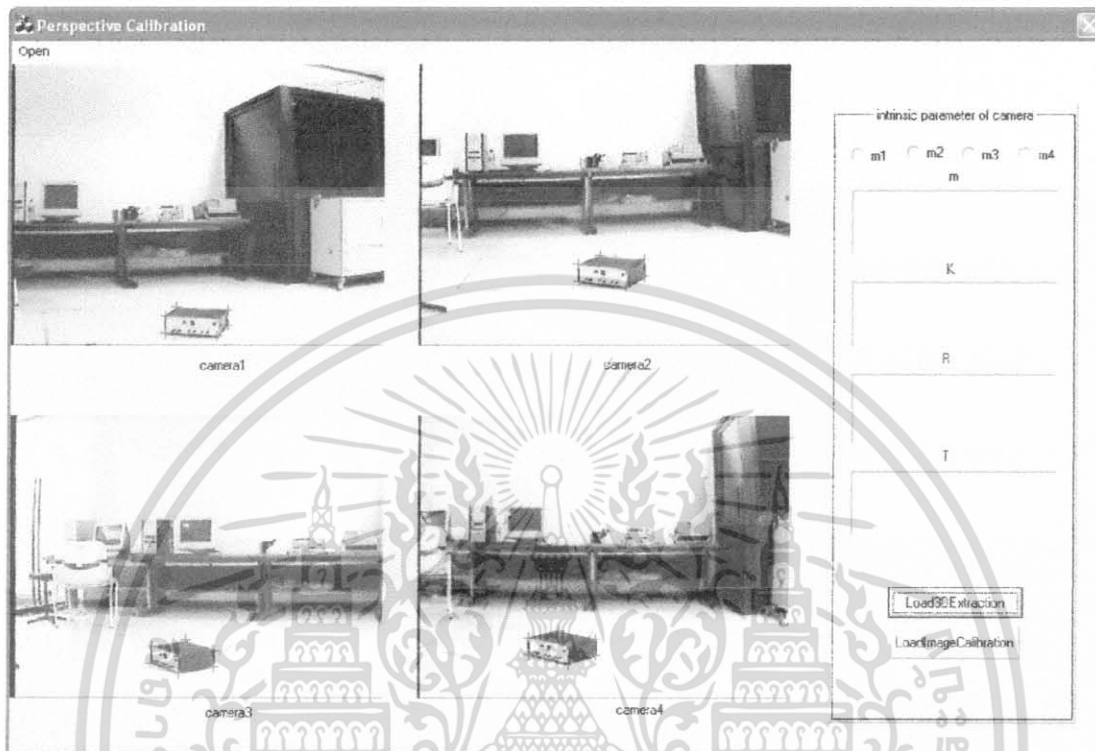
#### 4.2.3 การผลการทดลองประสิทธิภาพการหาปริมาตรของวัตถุแบบที่ 2

	ค่าจริง	ค่าจากการทดลอง	%ความถูกต้อง
ความกว้าง (X)	45.5 cm	46 cm	1.11
ความยาว (Y)	50.5 cm	50.3 cm	0.39
ความสูง (Z)	43 cm	42.8 cm	0.46
ปริมาตร(V)	98,803 cm <sup>3</sup>	99030.64 cm <sup>3</sup>	0.23

ตารางที่ 4.4 ตารางแสดงประสิทธิภาพการหาปริมาตรของวัตถุแบบที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### วัตถุทรงเรขาคณิตแบบที่ 3



รูปที่ 4.12 แสดงภาพการคลิกตำแหน่งพิกัดภาพของวัตถุแบบที่ 3 ด้วยเมาส์ทั้ง 4 มุมกล้อง



รูปที่ 4.13 แสดงค่าพิกัดจุด(X,Y,Z) ของจุดทั้ง 7 จุดของวัตถุแบบที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

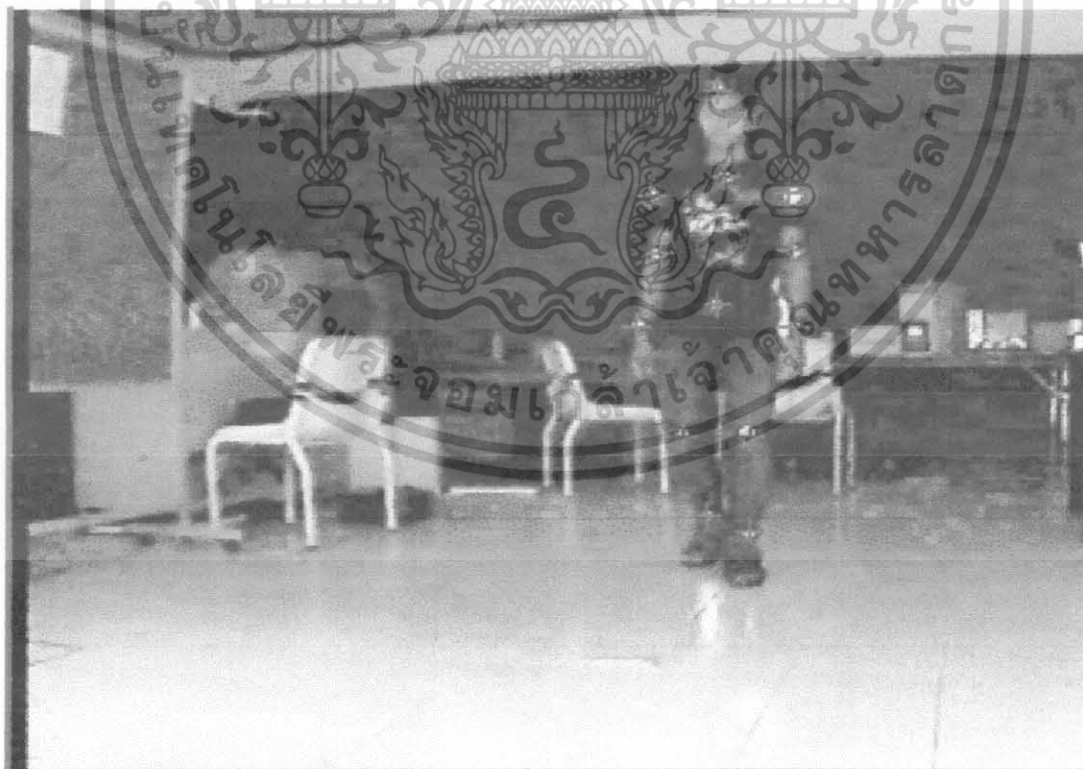
#### 4.2.4 ผลการทดลองประสิทธิภาพการหาปริมาตรของวัตถุแบบที่ 3

	ค่าจริง	ค่าจากการทดลอง	%ความถูกต้อง
ความกว้าง (X)	24.7 cm	25.1 cm	1.61
ความยาว (Y)	30.8 cm	30.4 cm	1.29
ความสูง (Z)	11.9 cm	11.7 cm	1.68
ปริมาตร(V)	9,053 cm <sup>3</sup>	8927.5 cm <sup>3</sup>	1.38

#### ตารางที่ 4.5 ตารางแสดงประสิทธิภาพการหาปริมาตรของวัตถุแบบที่ 3

#### 4.3 การทดลองคำนวณหาพิกัด 3 มิติในการเคลื่อนไหวของมนุษย์

หลังจากสามารถคำนวณหาพิกัด 3 มิติของวัตถุได้แล้ว เราจะทำการทดลองคำนวณหาพิกัด 3 มิติในการเคลื่อนไหวของมนุษย์ที่ทำการติดอุปกรณ์มาร์คเกอร์ ตามส่วนต่างๆ บนร่างกายของมนุษย์ทั้งหมด 13 จุดเพื่อที่จะนำข้อมูลภาพไปใช้ในการวิเคราะห์ท่าทางการเดินของมนุษย์

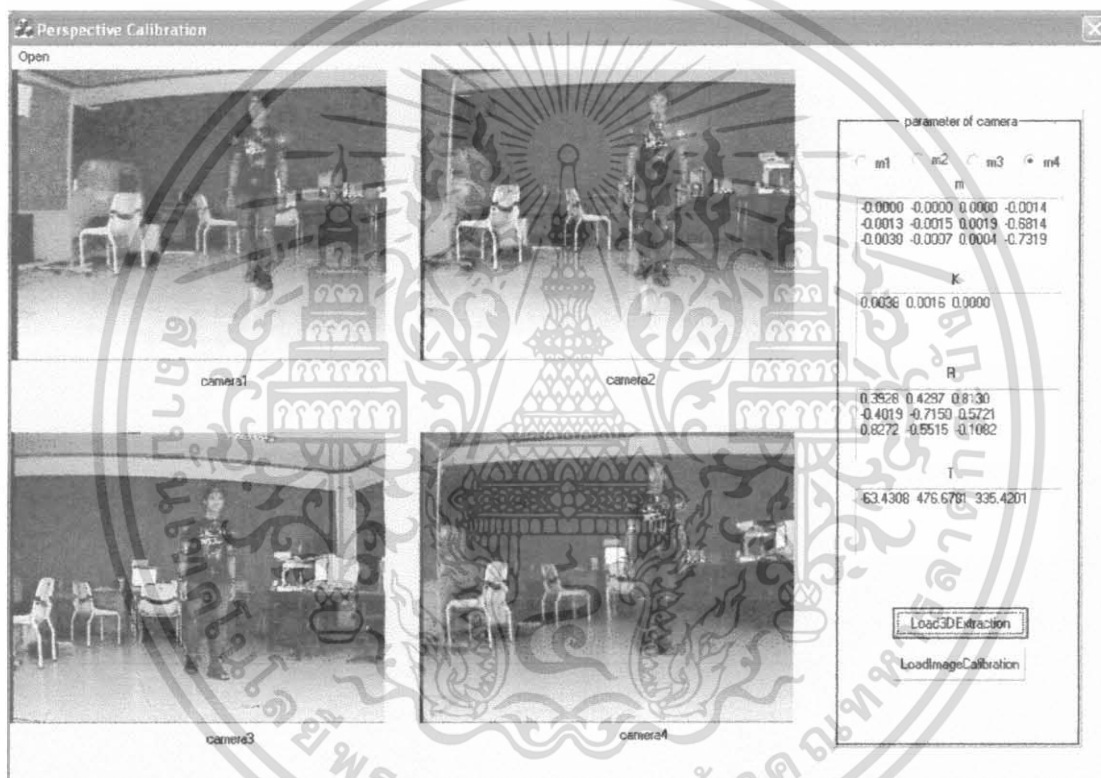


#### รูปที่ 4.14 แสดงภาพท่าทางการเคลื่อนไหวของมนุษย์ที่ทำการบันทึกไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.3.1 ขั้นตอนการทดลองการคำนวณหาพิกัด 3 มิติในการเคลื่อนไหวของมนุษย์มีดังต่อไปนี้

1. หลังจากทำการ Capture ภาพ model แล้วอย่าเพิ่งขยับกล้อง จากนั้นให้ทำการ Capture การเคลื่อนไหวของมนุษย์ที่ทำการติดอุปกรณ์มาร์คเกอร์
2. ทำการแปลงไฟล์ภาพที่ทำการ Capture จาก AVI เป็น BMP และ save ไฟล์ลงที่ไดรฟ์ที่ต้องการ
3. ทำการคลิกปุ่ม Load3DExtraction จะปรากฏการเคลื่อนไหวของมนุษย์ที่ทำการ Capture ไว้ดังรูปที่ 4.15



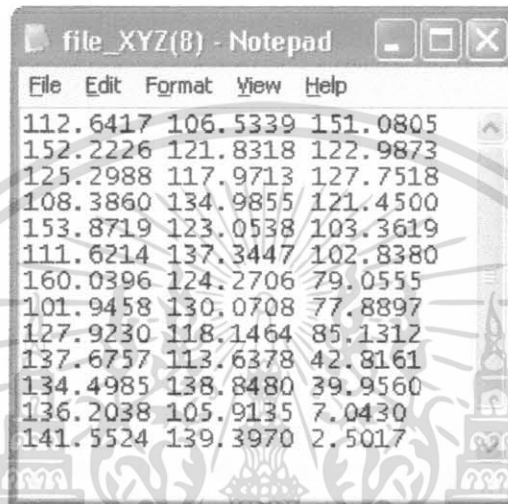
รูปที่ 4.15 แสดงภาพการคลิกตำแหน่งพิกัดภาพมนุษย์ด้วยเมาส์ทั้ง 4 มุมกล้อง

4. จากนั้นทำการใช้เมาส์คลิก (คลิกขวา) ตรงจุดมาร์คเกอร์โดยกำหนดจุดพิกัดของภาพจากกล้องที่ 1 โดยในการคลิกจะคลิกที่จุดใดก่อนก็ได้ให้ครบ 13 จุด
5. ทำการคลิกลำดับพิกัดจุดในภาพของกล้องที่ 2, 3 และ 4 ให้เหมือนกับกล้องที่ 1 ดังรูปที่ 4.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. หลังจากทำการคลิกครบ 13 จุดทั้ง 4 มุมก็ต้องแก้ค่าพิกัด (X,Y, Z) จะถูกบันทึกไว้ที่ file.dat ดังรูปที่ 4.16

#### 4.3.2 ผลการทดลองการคำนวณหาพิกัด 3 มิติในการเคลื่อนไหวของมนุษย์



File	Edit	Format	View	Help
112.6417	106.5339	151.0805		
152.2226	121.8318	122.9873		
125.2988	117.9713	127.7518		
108.3860	134.9855	121.4500		
153.8719	123.0538	103.3619		
111.6214	137.3447	102.8380		
160.0396	124.2706	79.0555		
101.9458	130.0708	77.8897		
127.9230	118.1464	85.1312		
137.6757	113.6378	42.8161		
134.4985	138.8480	39.9560		
136.2038	105.9135	7.0430		
141.5524	139.3970	2.5017		

รูปที่ 4.16 แสดงค่าพิกัดจุด(X,Y,Z) ของจุดทั้ง 13 จุดจากภาพการเคลื่อนไหวมนุษย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4 การทดลองสร้างภาพ 3 มิติโดยOpenGL

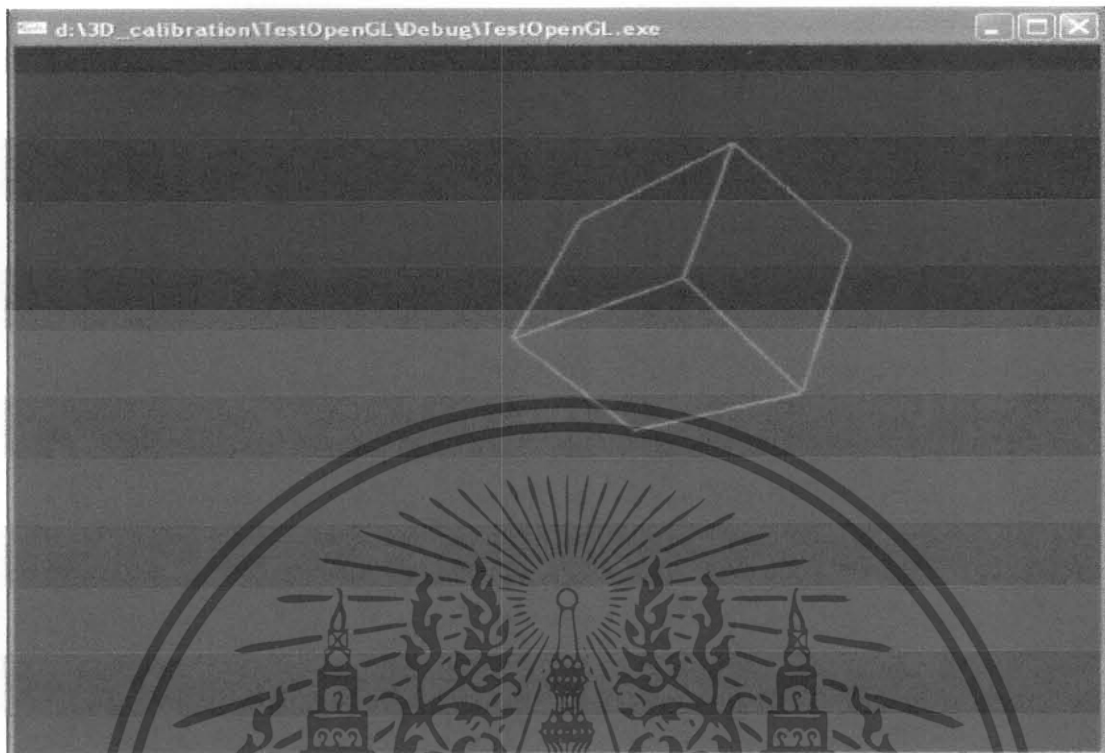
เมื่อได้จุดพิกัดโลก(X,Y,Z) ที่ได้จากพิกัดภาพของวัตถุ(u,v)ที่ทำการบันทึกเป็น file.dat OpenGL จะอ่านข้อมูลพิกัดโลก(X,Y,Z)ที่ทำการบันทึกเป็น file.dat มาพล็อตจุดพิกัดและทำการลากเส้นให้เป็น โครงสร้างรูประบบ 3 มิติ

##### 4.4.1 ผลการทดลองการสร้างภาพ 3 มิติโดยOpenGL



รูปที่ 4.17 แสดงภาพจริงของวัตถุแบบที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 แสดงการสร้างภาพ 3 มิติของวัตถุแบบที่ 1 ใน OpenGL

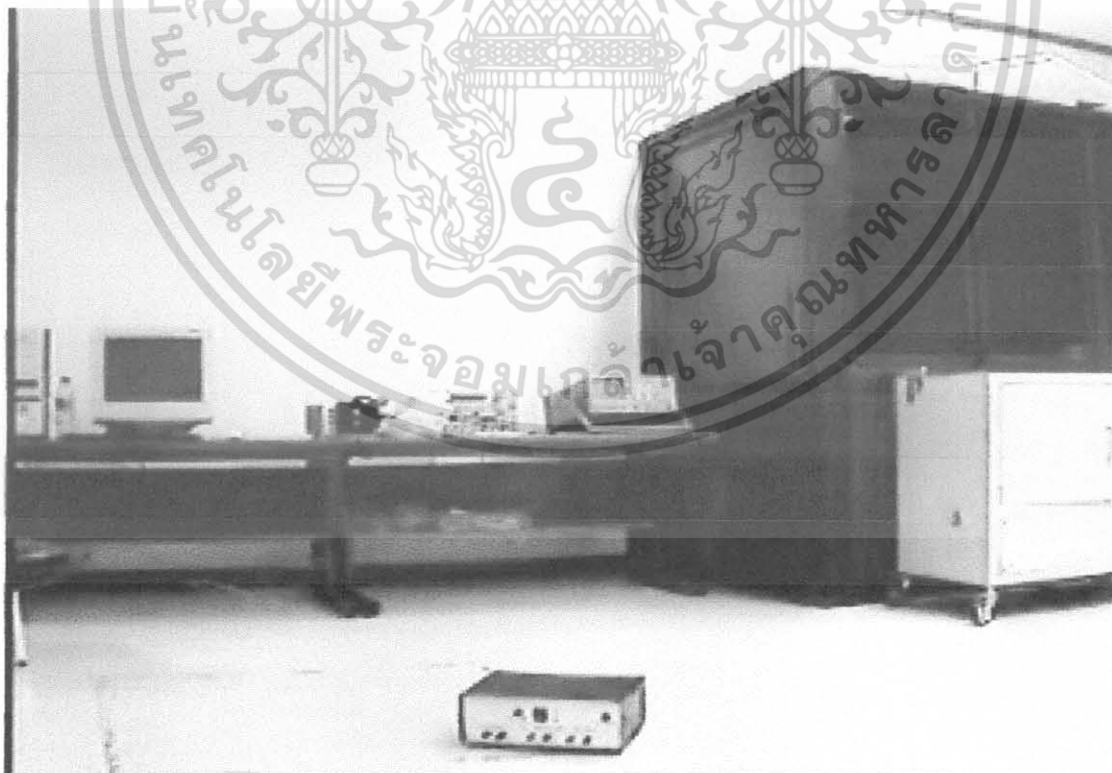


รูปที่ 4.19 แสดงภาพจริงของวัตถุแบบที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

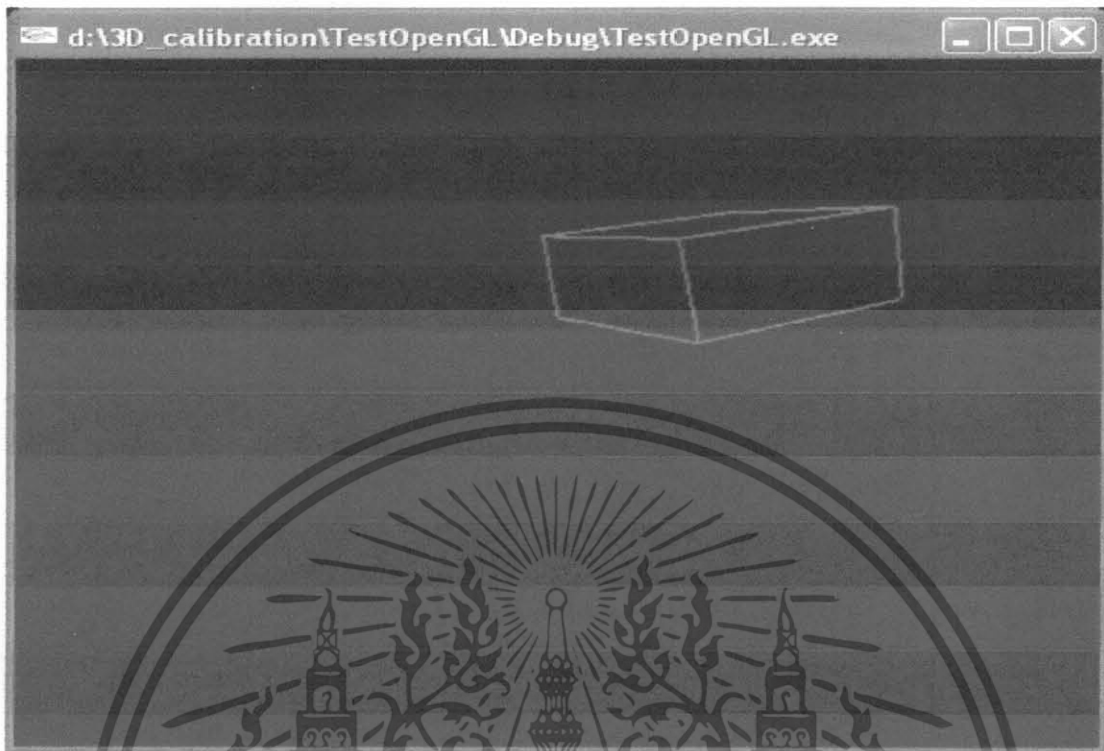


รูปที่ 4.20 แสดงการสร้างภาพ 3 มิติของวัตถุแบบที่ 2 ใน OpenGL



รูปที่ 4.21 แสดงภาพจริงของวัตถุแบบที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

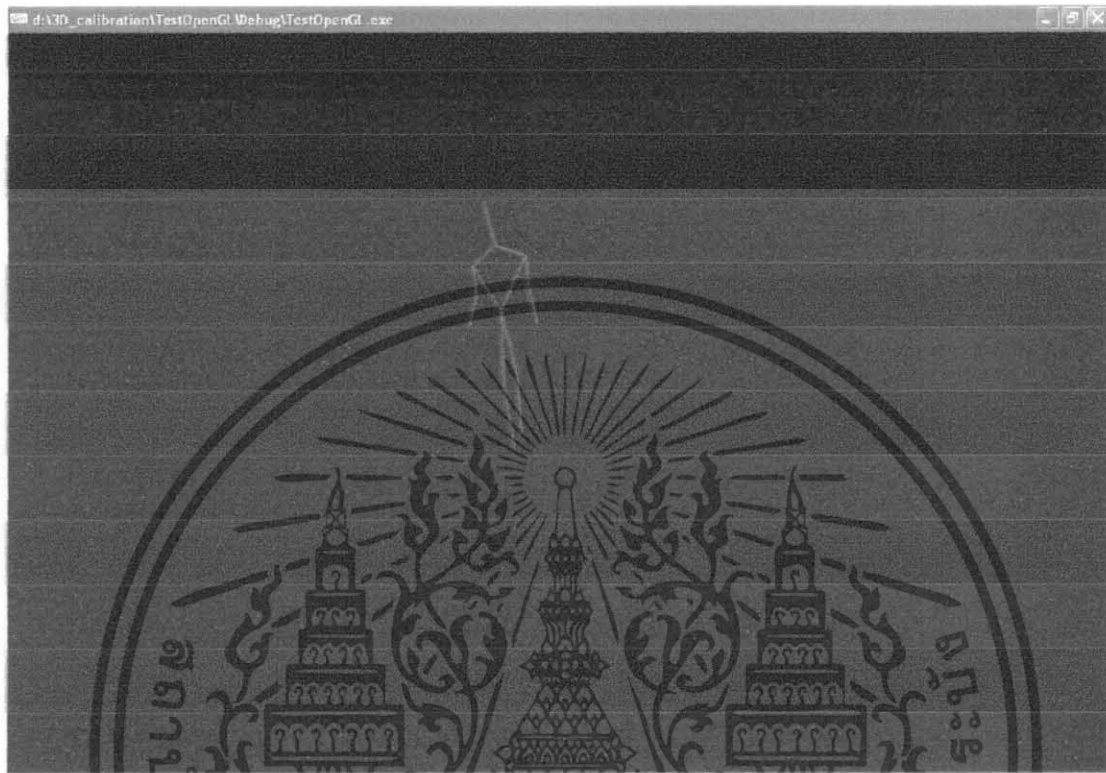


รูปที่ 4.22 แสดงการสร้างภาพ 3 มิติของวัตถุแบบที่3 ในOpenGL



รูปที่ 4.23 แสดงภาพท่าทางการเคลื่อนไหวจริงของมนุษย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 แสดงการสร้างภาพ 3 มิติของท่าทางการเคลื่อนไหวของมนุษย์ ในOpenGL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปและวิจารณ์

#### 5.1 วิเคราะห์และสรุปผลที่ได้จากการทดลอง

##### 1. ผลการทดลอง calibration

จากการใช้ฟังก์ชัน SVD แยก Matrix Perspective Transformation  $M$  นั้นจะได้ค่า  $(M_{11}-M_{34})$  ทั้งหมด 12 ค่า สามารถตรวจสอบผลค่า  $M$  12 ค่าได้โดยคำนวณย้อนกลับ ให้เมทริกซ์พิกัดโลกคูณเมทริกซ์  $M$  ผลที่ได้จะได้พิกัดภาพ ซึ่งทับซ้อนกับพิกัดภาพที่นำมาใส่ Matrix Perspective Transformation  $M$  จากการทดลอง ค่าพิกัดภาพที่ได้จากการคูณทับซ้อนกับพิกัดภาพเดิมไม่สนิทแต่มีความใกล้เคียงมากจึงยอมรับค่านี้ได้

##### 2. ผลการทดลองคำนวณหาปริมาตร

ผลจากการนำพิกัดที่สนใจในภาพทั้ง 4 มุมกลิ้งนำมาหาพิกัด 3 มิติในระบบพิกัดโลก โดยใช้สมการหาพิกัด 3 มิติ (2.42) นั้น จุดที่ได้ในระบบพิกัดโลก 3 มิติจะมีความสอดคล้องกับจุดพิกัดวัตถุจริงบนโลกเรียงตามลำดับกัน ซึ่งสามารถคำนวณระยะกว้าง ยาว และสูงได้จากจุดพิกัดโลกที่ได้จากสมการ และนำผลที่ได้ไปเทียบระยะของวัตถุจริง โดยคิดให้อยู่ในรูปเชิงปริมาตร จากการทดลองนั้นผลปริมาตรจากการทำ 3D Extraction นั้นได้ค่าปริมาตรใกล้เคียงกับปริมาตรวัตถุจริงที่ทำการ Extraction วัตถุนั้น ค่าผิดพลาดที่เกิดขึ้นจากการหาปริมาตรจริงของรูปทรงเรขาคณิตนั้นเกิดจากผลของค่าเมทริกซ์  $M$  และการสแกนจุดที่สนใจ ซึ่งการสแกนจุดนั้นจะใช้เมาส์คลิกที่ตำแหน่งพิกัดจึงทำให้เกิดความผิดพลาดของตำแหน่งได้

##### 3. ผลการทดลองสร้างภาพ 3 มิติจากวัตถุและมนุษย์โดย OpenGL

ค่าพิกัดโลกของวัตถุที่ทำการ Extraction นั้นจะถูกบันทึกไว้ในรูปไฟล์ข้อมูล ซึ่งจะเขียนโปรแกรมเพื่ออ่านพิกัดโลกมาเก็บไว้ในตัวแปร และนำตัวแปรนั้นมาสร้างพิกัด 3 มิติใน OpenGL จากนั้นลากเส้นเชื่อมต่อเพื่อง่ายต่อการสังเกต ภาพผลการทดลองที่ได้มีความคล้ายกับพิกัดที่พิจารณาจริงแต่มีปัญหาเรื่องมุมกลิ้งเนื่องจากสามารถเห็นตำแหน่งของมาร์คเกอร์ได้เฉพาะด้านหน้าเท่านั้น

#### 5.2 ปัญหาที่พบและข้อจำกัดของโปรแกรม

ในโปรแกรมนี้การคำนวณทางคณิตศาสตร์ที่ค่อนข้างยาก ซึ่งมีอุปสรรคในส่วนของ การปรับเทียบกล้อง (calibration camera) ซึ่งพบในส่วนของ การคำนวณย้อนกลับไม่ทับกันสนิท ห่างค่อนข้างมาก ซึ่งเป็นผลมาจากวัตถุที่ทำการ calibration (ตารางหมากรุก) โดยส่วนใหญ่ตารางหมากรุก

จะใช้ในการ calibration ได้อย่างถูกต้อง แต่ในการกำหนดพิกัดโลกในการออกแบบโปรแกรมมีปัญหา  
ค่อนข้างยากจึงทำให้การ calibration (ตารางหมากรุก) ได้ผลออกมาไม่สมบูรณ์ ดังนั้นจึงทำการเปลี่ยน  
วัตถุในการ calibration แทนโดยใช้เป็นวัตถุโครงสร้าง โมเดลสี่เหลี่ยมแทน

### 5.3 แนวทางในการพัฒนาและปรับปรุงเพื่อให้มีประสิทธิภาพมากขึ้น

จะต้องทำการพัฒนาการ calibration ด้วยตารางหมากรุกที่มีปัญหาดังเดิมนั้นของการทำ  
โครงงานและการปรับปรุงการสร้างภาพ 3 มิติให้ดูสมจริงและรูปร่างที่สวยงามมากขึ้น ในส่วนการหา  
ตำแหน่งพิกัดภาพของวัตถุซึ่งในการทดลองจะใช้เมาส์สแกน โดยการคลิกบอกตำแหน่งดังนั้นควรเพิ่ม  
ประสิทธิภาพให้สามารถสแกนบอกตำแหน่งพิกัดได้จากโปรแกรม โดยอัตโนมัติเพื่อให้เกิดความ  
ถูกต้องแม่นยำมากขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง

1. ผศ.ดร.ชูชาติ ปิณฑวิรุจน์, “เอกสารประกอบการเรียนการสอน วิชา Digital Image Processing”
2. นิรุช อำนวยศิลป์, “คู่มือการเขียนโปรแกรม Visual C++ version 6.0”, ซักเซส มีเดีย 1989



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

%%%%ฟังก์ชันรับพิกัดจากเมาส์คลิกซ้ายเพื่อนำพิกัดไปปรับเทียบกล้อง%%%%

```
void CMy3DExtrationDlg::OnLButtonDown(UINT nFlags, CPoint point)
```

```
{
```

```
    CDC *pdc=this->GetDC();
```

```
    CString str;
```

```
    CPen pen;
```

```
    pen.CreatePen(PS_SOLID,1,RGB(0,255,0));
```

```
    pdc->SelectObject(&pen);
```

```
    int A=9;//num of point
```

```
    n++;
```

```
    a[n]=point.x;
```

```
    b[n]=point.y;
```

```
    pdc->MoveTo(point.x-4,point.y);
```

```
    pdc->LineTo(point.x+5,point.y);
```

```
    pdc->MoveTo(point.x,point.y-4);
```

```
    pdc->LineTo(point.x,point.y+5);
```

```
    if(n==A){ n=0;
```

```
        U=matrix(1,A,1,2);
```

```
        for(int j=1;j<=A;j++){ U[j][1]=a[j]; U[j][2]=b[j];} //load point from
```

```
a,b to u[][]
```

```
        Computate_m(U);
```

```
    }
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

%%%%ฟังก์ชันคำนวณพารามิเตอร์ปรับเทียบกล้อง%%%%

```
float** CMy3DExtrationDlg::Compulate_m(float **u)
```

```
{
```

```
    CDC *pdc=this->GetDC();
```

```
    CString str;
```

```
    CPen pen;
```

```
    pen.CreatePen(PS_SOLID,1,RGB(255,0,0));//create pen reproject red color
```

```
    pdc->SelectObject(&pen);//keep pen draw
```

```
    float **xx,**xxt,**G,*w,**v,**vv,**m,pX,pY,X,**zz,**QR;
```

```
    int i,k,l,j,nn=0,min=0;
```

```
    int A=9;//define num of point
```

```
    xx=matrix(1,A,1,4);
```

```
    xxt=matrix(1,4,1,A);
```

```
    G=matrix(1,2*A,1,12);//matrix G w v use svd
```

```
    w=vector(1,12);
```

```
    v=matrix(1,12,1,12);
```

```
    vv=matrix(1,3,1,A);
```

```
    m=matrix(1,3,1,4);//m 12 value
```

```
    zz=matrix(1,4,1,A);//fine coordinate 3 dimension
```

```
    QR=matrix(1,3,1,3);//fine calibration
```

```
    xx[1][1]=0;    xx[1][2]=0;    xx[1][3]=0;
```

```
    xx[2][1]=53;  xx[2][2]=0;    xx[2][3]=0;
```

```
//    xx[3][1]=53;    xx[3][2]=53;    xx[3][3]=0;
```

```
    xx[3][1]=0;    xx[3][2]=53;    xx[3][3]=0;
```

```
    xx[4][1]=0;    xx[4][2]=0;    xx[4][3]=48.5;
```

```
    xx[5][1]=53;  xx[5][2]=0;    xx[5][3]=48.5;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//      xx[7][1]=53;      xx[7][2]=53;      xx[7][3]=48.5;
      xx[6][1]=0;      xx[6][2]=53;      xx[6][3]=48.5;

      xx[7][1]=0;      xx[7][2]=0;      xx[7][3]=97;
      xx[8][1]=53;      xx[8][2]=0;      xx[8][3]=97;
//      xx[11][1]=53;      xx[11][2]=53;      xx[11][3]=97;
      xx[9][1]=0;      xx[9][2]=53;      xx[9][3]=97;

      for(i=1;i<=A;i++){
      xxt[1][i]=xx[i][1];
      xxt[2][i]=xx[i][2];
      xxt[3][i]=xx[i][3];
      xxt[4][i]=xx[i][4]=1;
      }

      for(i=1;i<=A;i++){ //matrix calibration
      G[2*i][1]=0;
      G[2*i][2]=0;
      G[2*i][3]=0;
      G[2*i][4]=0;
      G[2*i][5]=xx[i][1];
      G[2*i][6]=xx[i][2];
      G[2*i][7]=xx[i][3];
      G[2*i][8]=1;
      G[2*i][9]=-u[i][2]*xx[i][1];
      G[2*i][10]=-u[i][2]*xx[i][2];
      G[2*i][11]=-u[i][2]*xx[i][3];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

G[2*i][12]=-u[i][2];

G[(2*i)-1][1]=xx[i][1];
G[(2*i)-1][2]=xx[i][2];
G[(2*i)-1][3]=xx[i][3];
G[(2*i)-1][4]=1;
G[(2*i)-1][5]=0;
G[(2*i)-1][6]=0;
G[(2*i)-1][7]=0;
G[(2*i)-1][8]=0;
G[(2*i)-1][9]=-u[i][1]*xx[i][1];
G[(2*i)-1][10]=-u[i][1]*xx[i][2];
G[(2*i)-1][11]=-u[i][1]*xx[i][3];
G[(2*i)-1][12]=-u[i][1];
}

svdcmp(G,2*A,12,w,v);//(u,m,n,w,v)

for(k=1;k<=12;k++){ //select column minimum
    X=w[k];
    for(l=1;l<=12;l++){
        if(X-w[l]<0){
            nn++;
        }
        if(nn==3)min=k;
    }
    nn=0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

m[1][1]=v[1][min];    m[1][2]=v[2][min];    m[1][3]=v[3][min];    m[1][4]=v[4][min];
m[2][1]=v[5][min];    m[2][2]=v[6][min];    m[2][3]=v[7][min];    m[2][4]=v[8][min];
m[3][1]=v[9][min];    m[3][2]=v[10][min];    m[3][3]=v[11][min];    m[3][4]=v[12][min];
/*
QR[1][1]=m[1][1];QR[1][2]=m[1][2];QR[1][3]=m[1][3];
QR[2][1]=m[2][1];QR[2][2]=m[2][2];QR[2][3]=m[2][3];
QR[3][1]=m[3][1];QR[3][2]=m[3][2];QR[3][3]=m[3][3];
joy++;
*/
//Compute_calibration(QR);
joy++;
switch(joy){
case 1:{
m1[1][1]=m[1][1];    m1[1][2]=m[1][2];    m1[1][3]=m[1][3];    m1[1][4]=m[1][4];
m1[2][1]=m[2][1];    m1[2][2]=m[2][2];    m1[2][3]=m[2][3];    m1[2][4]=m[2][4];
m1[3][1]=m[3][1];    m1[3][2]=m[3][2];    m1[3][3]=m[3][3];    m1[3][4]=m[3][4];

pfile=fopen("C:\\file_m1.dat","w");
for(j=1;j<=3;j++){
for(i=1;i<=4;i++){ fprintf(pfile,"%f",m1[j][i]);}
fprintf(pfile,"\n");
}
fclose(pfile);

u1[1][1]=u[1][1];u1[1][2]=u[1][2];
u1[2][1]=u[2][1];u1[2][2]=u[2][2];
u1[3][1]=u[3][1];u1[3][2]=u[3][2];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

u1[4][1]=u[4][1];u1[4][2]=u[4][2];
u1[5][1]=u[5][1];u1[5][2]=u[5][2];
u1[6][1]=u[6][1];u1[6][2]=u[6][2];
u1[7][1]=u[7][1];u1[7][2]=u[7][2];
u1[8][1]=u[8][1];u1[8][2]=u[8][2];
u1[9][1]=u[9][1];u1[9][2]=u[9][2];
//u1[10][1]=u[10][1];u1[10][2]=u[10][2];
//u1[11][1]=u[11][1];u1[11][2]=u[11][2];
//u1[12][1]=u[12][1];u1[12][2]=u[12][2];
    }
    break;
case 2: {
m2[1][1]=m[1][1];    m2[1][2]=m[1][2];    m2[1][3]=m[1][3];    m2[1][4]=m[1][4];
m2[2][1]=m[2][1];    m2[2][2]=m[2][2];    m2[2][3]=m[2][3];    m2[2][4]=m[2][4];
m2[3][1]=m[3][1];    m2[3][2]=m[3][2];    m2[3][3]=m[3][3];    m2[3][4]=m[3][4];
pfile=fopen("C:\\file_m2.dat","w");
for(j=1;j<=3;j++){
for(i=1;i<=4;i++){ fprintf(pfile,"%f ",m2[j][i]);}
fprintf(pfile,"\n");
}
fclose(pfile);

u2[1][1]=u[1][1];u2[1][2]=u[1][2];
u2[2][1]=u[2][1];u2[2][2]=u[2][2];
u2[3][1]=u[3][1];u2[3][2]=u[3][2];
u2[4][1]=u[4][1];u2[4][2]=u[4][2];
u2[5][1]=u[5][1];u2[5][2]=u[5][2];
u2[6][1]=u[6][1];u2[6][2]=u[6][2];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

u2[7][1]=u[7][1];u2[7][2]=u[7][2];
u2[9][1]=u[9][1];u2[9][2]=u[9][2];
//u2[10][1]=u[10][1];u2[10][2]=u[10][2];
//u2[11][1]=u[11][1];u2[11][2]=u[11][2];
//u2[12][1]=u[12][1];u2[12][2]=u[12][2];

    }

    break;

case 3:{
m3[1][1]=m[1][1];    m3[1][2]=m[1][2];    m3[1][3]=m[1][3];    m3[1][4]=m[1][4];
m3[2][1]=m[2][1];    m3[2][2]=m[2][2];    m3[2][3]=m[2][3];    m3[2][4]=m[2][4];
m3[3][1]=m[3][1];    m3[3][2]=m[3][2];    m3[3][3]=m[3][3];    m3[3][4]=m[3][4];

pfile=fopen("C:\\file_m3.dat","w");
for(j=1;j<=3;j++){
for(i=1;i<=4;i++){ fprintf(pfile,"%f",m3[j][i]);}
fprintf(pfile,"\n");
}
fclose(pfile);
u3[1][1]=u[1][1]; u3[1][2]=u[1][2];
u3[2][1]=u[2][1]; u3[2][2]=u[2][2];
u3[3][1]=u[3][1]; u3[3][2]=u[3][2];
u3[4][1]=u[4][1]; u3[4][2]=u[4][2];
u3[5][1]=u[5][1]; u3[5][2]=u[5][2];
u3[6][1]=u[6][1]; u3[6][2]=u[6][2];
u3[7][1]=u[7][1]; u3[7][2]=u[7][2];
u3[9][1]=u[9][1];u3[9][2]=u[9][2];
//u3[10][1]=u[10][1];u3[10][2]=u[10][2];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//u3[11][1]=u[11][1];u3[11][2]=u[11][2];
//u3[12][1]=u[12][1];u3[12][2]=u[12][2];
    }
    break;
case 4:{
m4[1][1]=m[1][1];    m4[1][2]=m[1][2];    m4[1][3]=m[1][3];    m4[1][4]=m[1][4];
m4[2][1]=m[2][1];    m4[2][2]=m[2][2];    m4[2][3]=m[2][3];    m4[2][4]=m[2][4];
m4[3][1]=m[3][1];    m4[3][2]=m[3][2];    m4[3][3]=m[3][3];    m4[3][4]=m[3][4];

pfile=fopen("C:\\file_m4.dat","w");
for(j=1;j<=3;j++){
for(i=1;i<=4;i++){fprintf(pfile,"%f",m4[j][i]);}
fprintf(pfile,"\n");
}
fclose(pfile);

u4[1][1]=u[1][1];u4[1][2]=u[1][2];
u4[2][1]=u[2][1];u4[2][2]=u[2][2];
u4[3][1]=u[3][1];u4[3][2]=u[3][2];
u4[4][1]=u[4][1];u4[4][2]=u[4][2];
u4[5][1]=u[5][1];u4[5][2]=u[5][2];
u4[6][1]=u[6][1];u4[6][2]=u[6][2];
u4[7][1]=u[7][1];u4[7][2]=u[7][2];
u4[9][1]=u[9][1];u4[9][2]=u[9][2];
//u4[10][1]=u[10][1];u4[10][2]=u[10][2];
//u4[11][1]=u[11][1];u4[11][2]=u[11][2];
//u4[12][1]=u[12][1];u4[12][2]=u[12][2];

joy=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    break;
}
for(k=1;k<=3;k++){ //reproject
    for(l=1;l<=A;l++){ //multiplie
m*xx(Transpos)
                                vv[k][l]=0.0; //equa coordidte Image
        for(j=1;j<=4;j++){
                                vv[k][l]+=m[k][j]*xxt[j][l]; //
        }
    }
}
//
for(i=1;i<=A;i++){
    str.Format("%.2f %.2f",pX=(vv[1][i]/vv[3][i]),pY=(vv[2][i]/vv[3][i]));
    pdc->MoveTo((pX*1)-4,(pY*1)-5);
    pdc->LineTo((pX*1)+4,(pY*1)+5);
    pdc->MoveTo((pX*1)-4,(pY*1)+5);
    pdc->LineTo((pX*1)+4,(pY*1)-5);
}
free_matrix(xxt,1,4,1,A);
free_matrix(vv,1,3,1,A);
free_matrix(m,1,3,1,4);
free_matrix(xx,1,A,1,4);
free_matrix(G,1,2*A,1,12);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
free_matrix(v,1,12,1,12);
```

```
free_vector(w,1,12);
```

```
return 0;
```

```
}
```

```
%%%%ฟังก์ชันรับพิกัดจากเมาส์คลิกขวาเพื่อนำไปสร้างพิกัด3มิติ%%%%
```

```
void CMY3DExtrationDlg::OnRButtonDown(UINT nFlags, CPoint point)
```

```
{
```

```
    CDC *pdc=this->GetDC();
```

```
    CString str;
```

```
    CPen pen;
```

```
    int j,A=13;//define num of point
```

```
    m++;
```

```
    a[m]=point.x;
```

```
    b[m]=point.y;
```

```
    pen.CreatePen(PS_SOLID,1,RGB(0,0,255));
```

```
    pdc->SelectObject(&pen);
```

```
    pdc->MoveTo(point.x-4,point.y);
```

```
    pdc->LineTo(point.x+5,point.y);
```

```
    pdc->MoveTo(point.x,point.y-4);
```

```
    pdc->LineTo(point.x,point.y+5);
```

```
if(m==A){
```

```
    ake++;
```

```
    switch(ake){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 1:{ m=0;

U1=matrix(1,A,1,2);
for(j=1;j<=A;j++){ U1[j][1]=a[j]; U1[j][2]=b[j];}

//load point from a,b to u[][]

}break;

case 2:{ m=0;

U2=matrix(1,A,1,2);
for(j=1;j<=A;j++){ U2[j][1]=a[j]; U2[j][2]=b[j];}

//load point from a,b to u[][]

}break;

case 3:{ m=0;

U3=matrix(1,A,1,2);
for(j=1;j<=A;j++){ U3[j][1]=a[j]; U3[j][2]=b[j];}

//load point from a,b to u[][]

}break;

case 4:{ m=0;

U4=matrix(1,A,1,2);
for(j=1;j<=A;j++){ U4[j][1]=a[j]; U4[j][2]=b[j];}

//load point from a,b to u[][]

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ake=0;
Compute_3DExtraction(U1,U2,U3,U4);
}break;
}
}
/*
pfile=fopen("C:\\file_zz.dat","r");
for(b=1;b<=8;b++){
for(k=1;k<=3;k++){fscanf(pfile,"%f",&X[b][k]);printf("%f ",X[b][k]);}
fprintf(pfile,"\n");
printf("\n");
}
fclose(pfile);
*/
}
}
%%%%ฟังก์ชันคำนวณหาพิกัด3มิติ%%
void CMy3DExtractionDlg::Compute_3DExtraction(float **u1, float **u2, float **u3, float **u4)
{
int min=0,nn=0,i,j,k,l,A=13;//define num ofpoint
float m1[5][5],m2[5][5],m3[5][5],m4[5][5]**zz,**G,*w,**v,X;

zz=matrix(1,4,1,A);
G=matrix(1,8,1,4); //matrix G w v use svd
w=vector(1,12);
v=matrix(1,4,1,4);

pfile=fopen("C:\\file_m1.dat","r");
for(int b=1;b<=3;b++){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        for(int k=1;k<=4;k++){fscanf(pfile,"%f",&m1[b][k]);printf("%f ",m1[b][k]);}
fprintf(pfile,"\n");
        printf("\n");
    }
fclose(pfile);

pfile=fopen("C:\\file_m2.dat","r");
for(b=1;b<=3;b++){
    for(k=1;k<=4;k++){fscanf(pfile,"%f",&m2[b][k]);printf("%f ",m2[b][k]);}
fprintf(pfile,"\n");
    printf("\n");
}
fclose(pfile);

pfile=fopen("C:\\file_m3.dat","r");
for(b=1;b<=3;b++){
    for(k=1;k<=4;k++){fscanf(pfile,"%f",&m3[b][k]);printf("%f ",m3[b][k]);}
fprintf(pfile,"\n");
    printf("\n");
}
fclose(pfile);

pfile=fopen("C:\\file_m4.dat","r");
for(b=1;b<=3;b++){
    for(k=1;k<=4;k++){fscanf(pfile,"%f",&m4[b][k]);printf("%f ",m4[b][k]);}
fprintf(pfile,"\n");
    printf("\n");
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

fclose(pfile);

for(i=1;i<=A;i++){
    for(k=1;k<=4;k++){
        G[1][k]=(u1[i][1]*m1[3][k])-m1[1][k]; //matrix fine 8x4
        G[2][k]=(u1[i][2]*m1[3][k])-m1[2][k];
        G[3][k]=(u2[i][1]*m2[3][k])-m2[1][k];
        G[4][k]=(u2[i][2]*m2[3][k])-m2[2][k];
        G[5][k]=(u3[i][1]*m3[3][k])-m3[1][k];
        G[6][k]=(u3[i][2]*m3[3][k])-m3[2][k];
        G[7][k]=(u4[i][1]*m4[3][k])-m4[1][k];
        G[8][k]=(u4[i][2]*m4[3][k])-m4[2][k];
    }
    svdcmp(G,8,4,w,v); // (u,m,n,w,v)
    for(k=1;k<=4;k++){ //select column minimum
        X=w[k];
        for(l=1;l<=4;l++){
            if(X-w[l]<0){
                nn++;
            }
            if(nn==3)min=k;
        }
        nn=0;
    }

    zz[1][i]=v[1][min];
    zz[2][i]=v[2][min];
    zz[3][i]=v[3][min];
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        zz[4][i]=v[4][min];
    }

char a[40];
count++;
sprintf(a,"C:\\Ake\\file_XYZ(%d).dat",count);
pfile=fopen(a,"w");

for(j=1;j<=A;j++){
for(i=1;i<=3;i++){ fprintf(pfile,"%0.4f",zz[i][j]/zz[4][j]);}
fprintf(pfile,"\n");
}

fclose(pfile);

pfile=fopen("C:\\file_u1.dat","w");
for(j=1;j<=A;j++){
for(i=1;i<=2;i++){ fprintf(pfile,"%f",u1[j][i]);}
fprintf(pfile,"\n");
}
fclose(pfile);

pfile=fopen("C:\\file_u2.dat","w");
for(j=1;j<=A;j++){
for(i=1;i<=2;i++){ fprintf(pfile,"%f",u2[j][i]);}
fprintf(pfile,"\n");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    fclose(pfile);

    pfile=fopen("C:\\file_u3.dat","w");
    for(j=1;j<=A;j++){
    for(i=1;i<=2;i++){ fprintf(pfile,"%f ",u3[j][i]);}
    fprintf(pfile,"\n");
    }
    fclose(pfile);

    pfile=fopen("C:\\file_u4.dat","w");
    for(j=1;j<=A;j++){
    for(i=1;i<=2;i++){ fprintf(pfile,"%f ",u4[j][i]);}
    fprintf(pfile,"\n");
    }
    fclose(pfile);
}

}

%%%%%%ฟังก์ชันแยกเมทริกด้วยQR%%%%%%%%
float** CMy3DExtractionDlg::Compuatate_calibration(float **qr)
{
int i,j,k,l,n,sing;
float con,*c,*d,**q,**qt,**r,**x;

c=vector(1,3);
d=vector(1,3);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

q=matrix(1,3,1,3);
qt=matrix(1,3,1,3);
r=matrix(1,3,1,3);
x=matrix(1,3,1,3);

n=3;
qrncmp(qr,n,c,d,&sing);
for(k=1;k<=n;k++){
    for(l=1;l<=n;l++){
        if(l>k){
            r[k][l]=qr[k][l];
            q[k][l]=0.0;
        }else if(l<k){
            r[k][l]=q[k][l]=0.0;
        }else {
            r[k][l]=d[k];
            q[k][l]=1.0;
        }
    }
}
for(i=n-1;i>=1;i--){
    for(con=0.0,k=i;k<=n;k++) con+=qr[k][i]*qr[k][i];
    con/=2.0;
    for(k=i;k<=n;k++){
        for(l=i;l<=n;l++){
            qt[k][l]=0.0;
            for(j=i;j<=n;j++){
                qt[k][l]+=q[j][l]*qr[k][i]*qr[j][i]/con;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}
}
for(k=i;k<=n;k++)
    for(l=i;l<=n;l++)q[k][l]=qt[k][l];
}

```

```

//for(k=1;k<=n;k++){
// for(l=1;l<=n;l++) printf("%12.6f",q[k][l]);
// printf("\n");
//}

```

```

//printf("\n");
//for(k=1;k<=n;k++){
// for(l=1;l<=n;l++) printf("%12.6f",r[k][l]);
// printf("\n");
//}

```

```
char ake[20];
```

```
switch(joy){
```

```
case 1:{
```

```
printf(ake,"f1=%f",r[1][1]/r[3][3]);
```

```
m_list1.AddString(ake);
```

```
printf(ake,"f2=%f",r[2][2]/r[3][3]);
```

```
m_list1.AddString(ake);
```

```
printf(ake,"alpha=%f",r[1][2]/r[3][3]);
```

```
m_list1.AddString(ake);
```

```
printf(ake,"cc1=%f",r[1][3]/r[3][3]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

m_list1.AddString(ake);
sprintf(ake,"cc2=%f",r[2][3]/r[3][3]);
m_list1.AddString(ake);
    }break;

case 2:{
sprintf(ake,"f1=%f",r[1][1]/r[3][3]);
m_list2.AddString(ake);
sprintf(ake,"f2=%f",r[2][2]/r[3][3]);
m_list2.AddString(ake);
sprintf(ake,"alpha=%f",r[1][2]/r[3][3]);
m_list2.AddString(ake);
sprintf(ake,"cc1=%f",r[1][3]/r[3][3]);
m_list2.AddString(ake);
sprintf(ake,"cc2=%f",r[2][3]/r[3][3]);
m_list2.AddString(ake);
    }break;

case 3:{
sprintf(ake,"f1=%f",r[1][1]/r[3][3]);
m_list3.AddString(ake);
sprintf(ake,"f2=%f",r[2][2]/r[3][3]);
m_list3.AddString(ake);
sprintf(ake,"alpha=%f",r[1][2]/r[3][3]);
m_list3.AddString(ake);
sprintf(ake,"cc1=%f",r[1][3]/r[3][3]);
m_list3.AddString(ake);
sprintf(ake,"cc2=%f",r[2][3]/r[3][3]);
m_list3.AddString(ake);
    }break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 4:{
printf(ake,"f1=%f",r[1][1]/r[3][3]);
m_list4.AddString(ake);
printf(ake,"f2=%f",r[2][2]/r[3][3]);
m_list4.AddString(ake);
printf(ake,"alpha=%f",r[1][2]/r[3][3]);
m_list4.AddString(ake);
printf(ake,"cc1=%f",r[1][3]/r[3][3]);
m_list4.AddString(ake);
printf(ake,"cc2=%f",r[2][3]/r[3][3]);
m_list4.AddString(ake);
    }break;
}
free_matrix(x,1,3,1,3);
//free_matrix(r,1,3,1,3);
free_matrix(qt,1,3,1,3);
//free_matrix(q,1,3,1,3);
free_vector(d,1,3);
free_vector(c,1,3);

/*
pfile=fopen("C:\\file_qr.dat","w");
for(int j=1;j<=3;j++){
for(int i=1;i<=3;i++){ fprintf(pfile,"%0.4f ",qr[j][i]);}
fprintf(pfile,"\n");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fclose(pfile);
```

```
*/
```

```
return 0;
```

```
}
```

```
%%%%ฟังก์ชันโหลดภาพ%%%%
```

```
void CMy3DExtractionDlg::Functionloadimage3D()
```

```
{
```

```
char a[100];
```

```
int i;
```

```
digi2++;
```

```
for(i=1;i<=4;i++){
```

```
if(digi2==10){ digi1++;digi2=0;}
```

```
sprintf(a,"D:\\3D_calibration\\avitobmp\\motion_%d\\2549-02-05_00-13-14 2549-02-05_00-13-1800%d%d.bmp",i,digi1,digi2);
```

```
//sprintf(a,"D:\\3D_calibration\\avitobmp\\supplie%d.bmp\\2006-01-29_23-26-02 2006-01-29_23-26-0600%d%d.bmp",i,digi1,digi2);
```

```
HBITMAP hBitmap = (HBITMAP) ::LoadImage(AfxGetInstanceHandle(),a, IMAGE_BITMAP, 0, 0, LR_LOADFROMFILE | LR_CREATEDIBSECTION);
```

```
//Attach the HBITMAP to a CBitmap
```

```
CBitmap m_bmpBitmap;
```

```
m_bmpBitmap.Attach(hBitmap);
```

```
//Get the loaded bitmap
```

```
BITMAP bm;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

m_bmpBitmap.GetBitmap(&bm);

//prepare drawing
CDC dcMem;
CDC* pdc = GetDC();
dcMem.CreateCompatibleDC(pdc);
CBitmap* pOldBitmap = (CBitmap*)dcMem.SelectObject(m_bmpBitmap);
switch(i){
case 1:{pdc->BitBlt (0,0, bm.bmWidth, bm.bmHeight, &dcMem, 0, 0, SRCCOPY);}break;
case 2:{pdc->BitBlt (350,0, bm.bmWidth, bm.bmHeight, &dcMem, 0, 0, SRCCOPY);}break;
case 3:{pdc->BitBlt (350,300, bm.bmWidth, bm.bmHeight, &dcMem, 0, 0, SRCCOPY);}break;
case 4:{pdc->BitBlt (0,300, bm.bmWidth, bm.bmHeight, &dcMem, 0, 0, SRCCOPY);}break;
}
ReleaseDC(pdc);
}
}

%%%%ฟังก์ชันเหตุการณ์เพื่อดูพารามิเตอร์ที่1%%%%
void CMY3DExtrationDlg::OnRadio_m1()
{

m_list1.DeleteString(3);
m_list1.DeleteString(2);
m_list1.DeleteString(1);
m_list1.DeleteString(0);

m_list2.DeleteString(0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
m_list3.DeleteString(2);
m_list3.DeleteString(1);
m_list3.DeleteString(0);
```

```
m_list4.DeleteString(0);
```

```
char ake[50];
float X[5][5]**A,**v,*w;
double **A_inv,B[4]**d,T[4];
int k,b;

d=dmatrix(1,3,1,3);
A_inv=dmatrix(1,3,1,3);
A=matrix(1,3,1,3);
v=matrix(1,3,1,3);
w=vector(1,3);

d[1][1]=1;d[1][2]=0;d[1][3]=0;
d[2][1]=0;d[2][2]=1;d[2][3]=0;
d[3][1]=0;d[3][2]=0;d[3][3]=1;

pfile=fopen("C:\\file_m1.dat","r");
for(b=1;b<=3;b++){
    for(k=1;k<=4;k++){fscanf(pfile,"%f",&X[b][k]);}
    fprintf(pfile,"\n");
    sprintf(ake,"%0.4f %0.4f %0.4f %0.4f ",X[b][1],X[b][2],X[b][3],X[b][4]);
    m_list1.AddString(ake);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

fclose(pfile);

for(b=1;b<=3;b++){ for(k=1;k<=3;k++){A_inv[b][k]=A[b][k]=X[b][k];} }

for(k=1;k<=3;k++)B[k]=X[k][4];

svdcmp(A,3,3,w,v); //(u,m,n,w,v)

/*
for(i=1;i<=3;i++){
    X=w[i];
    for(k=1;k<=3;k++)
        if((X-w[k])<0)n++;
    n=0;
    if(n==2){joy[3]=X;ff=i;n=0;}
}
*/

sprintf(ake,"% .4f % .4f % .4f ",w[1],w[2],w[3]);
m_list2.AddString(ake);

for(b=1;b<=3;b++){
    sprintf(ake,"% .4f % .4f % .4f ",v[b][1],v[b][2],v[b][3]);
    m_list3.AddString(ake);
}

gaussj(A_inv,3,d,3);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(k=1;k<=3;k++){
    T[k]=0;
    for(b=1;b<=3;b++){T[k]=T[k]+A_inv[k][b]*B[b];}
}

sprintf(ake,"%0.4f %0.4f %0.4f ",T[1],T[2],T[3]);
m_list4.AddString(ake);

free_dmatrix(d, 1, 3, 1, 3);
free_matrix(A,1,3,1,3);
free_matrix(v,1,3,1,3);
free_vector(w,1,3);
}

```

%% %% ฟังก์ชันเหตุการณ์เพื่อดูพารามิเตอร์ที่ 2 %% %%

```

void CMy3DExtrationDlg::OnRadio_m2()
{
    m_list1.DeleteString(3);
    m_list1.DeleteString(2);
    m_list1.DeleteString(1);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
m_list1.DeleteString(0);
```

```
m_list2.DeleteString(0);
```

```
m_list3.DeleteString(2);
```

```
m_list3.DeleteString(1);
```

```
m_list3.DeleteString(0);
```

```
m_list4.DeleteString(0);
```

```
char ake[50];
```

```
float X[5][5]**A,**v,*w;
```

```
double **A_inv,B[4]**d,T[4];
```

```
int k,b;
```

```
d=dmatrix(1,3,1,3);
```

```
A_inv=dmatrix(1,3,1,3);
```

```
A=matrix(1,3,1,3);
```

```
v=matrix(1,3,1,3);
```

```
w=vector(1,3);
```

```
d[1][1]=1;
```

```
d[1][2]=0;
```

```
d[1][3]=0;
```

```
d[2][1]=0;
```

```
d[2][2]=1;
```

```
d[2][3]=0;
```

```
d[3][1]=0;
```

```
d[3][2]=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
d[3][3]=1;
```

```
pfile=fopen("C:\\file_m2.dat","r");
```

```
for(b=1;b<=3;b++){
```

```
    for(k=1;k<=4;k++){fscanf(pfile,"%f",&X[b][k]);}
```

```
    fprintf(pfile,"\n");
```

```
    sprintf(ake,"%0.4f %0.4f %0.4f %0.4f ",X[b][1],X[b][2],X[b][3],X[b][4]);
```

```
    m_list1.AddString(ake);
```

```
    }
```

```
fclose(pfile);
```

```
for(b=1;b<=3;b++){ for(k=1;k<=3;k++){A_inv[b][k]=A[b][k]=X[b][k];} }
```

```
for(k=1;k<=3;k++)B[k]=X[k][4];
```

```
svdcmp(A,3,3,w,v); //(u,m,n,w,v)
```

```
/*
```

```
for(i=1;i<=3;i++){
```

```
    X=w[i];
```

```
    for(k=1;k<=3;k++)
```

```
        if((X-w[k])<0)n--;
```

```
        n=0;
```

```
        if(n==2){joy[3]=X;ff=i;n=0;}
```

```
    }
```

```
*/
```

```
    sprintf(ake,"%0.4f %0.4f %0.4f ",w[1],w[2],w[3]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

m_list2.AddString(ake);

for(b=1;b<=3;b++){
    sprintf(ake,"%0.4f %0.4f %0.4f ",v[b][1],v[b][2],v[b][3]);
    m_list3.AddString(ake);
}

gaussj(A_inv,3,d,3);

for(k=1;k<=3;k++){
    T[k]=0.0;
    for(b=1;b<=3;b++){T[k]=+A_inv[k][b]*B[b];}
}

    sprintf(ake,"%0.4f %0.4f %0.4f ",T[1],T[2],T[3]);
    m_list4.AddString(ake);

free_dmatrix(d, 1, 3, 1, 3);
free_matrix(A,1,3,1,3);
free_matrix(v,1,3,1,3);
free_vector(w,1,3);

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

%%%%ฟังก์ชันเหตุการณ์เพื่อดูพารามิเตอร์ที่3%%%%

```
void CMy3DExtractionDlg::OnRadio_m3()
```

```
{
```

```
m_list1.DeleteString(3);
```

```
m_list1.DeleteString(2);
```

```
m_list1.DeleteString(1);
```

```
m_list1.DeleteString(0);
```

```
m_list2.DeleteString(0);
```

```
m_list3.DeleteString(2);
```

```
m_list3.DeleteString(1);
```

```
m_list3.DeleteString(0);
```

```
m_list4.DeleteString(0);
```

```
char ake[50];
```

```
float X[5][5]**A,**v,*w;
```

```
double **A_inv,B[4]**d,T[4];
```

```
int k,b;
```

```
d=dmatrix(1,3,1,3);
```

```
A_inv=dmatrix(1,3,1,3);
```

```
A=matrix(1,3,1,3);
```

```
v=matrix(1,3,1,3);
```

```
w=vector(1,3);
```

```
d[1][1]=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
d[1][2]=0;
d[1][3]=0;
d[2][1]=0;
d[2][2]=1;
d[2][3]=0;
d[3][1]=0;
d[3][2]=0;
d[3][3]=1;
```

```
pfile=fopen("C:\\file_m3.dat","r");
for(b=1;b<=3;b++){
    for(k=1;k<=4;k++){fscanf(pfile,"%f",&X[b][k]);}
    fprintf(pfile,"\n");
    sprintf(ake,"%0.4f %0.4f %0.4f %0.4f ",X[b][1],X[b][2],X[b][3],X[b][4]);
    m_list1.AddString(ake);
}
fclose(pfile);

for(b=1;b<=3;b++){ for(k=1;k<=3;k++){A_inv[b][k]=A[b][k]=X[b][k];} }

for(k=1;k<=3;k++)B[k]=X[k][4];
```

```
svdcmp(A,3,3,w,v); //(u,m,n,w,v)
```

```
/*
```

```
for(i=1;i<=3;i++){
```

```
    X=w[i];
```

```
    for(k=1;k<=3;k++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if((X-w[k])<0)n++;
        n=0;
        if(n==2){joy[3]=X;ff=i;n=0;}
    }
*/

printf(ake,"%%.4f %.4f %.4f ",w[1],w[2],w[3]);
m_list2.AddString(ake);

for(b=1;b<=3;b++){
    printf(ake,"%%.4f %.4f %.4f ",v[b][1],v[b][2],v[b][3]);
    m_list3.AddString(ake);
}

gaussj(A_inv,3,d,3);

for(k=1;k<=3;k++){
    T[k]=0.0;
    for(b=1;b<=3;b++){T[k]=+A_inv[k][b]*B[b];}
}

printf(ake,"%%.4f %.4f %.4f ",T[1],T[2],T[3]);
m_list4.AddString(ake);

free_dmatrix(d, 1, 3, 1, 3);
free_matrix(A,1,3,1,3);
free_matrix(v,1,3,1,3);
free_vector(w,1,3);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
```

```
void CMy3DExtrationDlg::OnRadio_m4()
```

```
{
```

```
m_list1.DeleteString(3);
```

```
m_list1.DeleteString(2);
```

```
m_list1.DeleteString(1);
```

```
m_list1.DeleteString(0);
```

```
m_list2.DeleteString(0);
```

```
m_list3.DeleteString(2);
```

```
m_list3.DeleteString(1);
```

```
m_list3.DeleteString(0);
```

```
m_list4.DeleteString(0);
```

```
char ake[50];
```

```
float X[5][5]**A,**v,*w;
```

```
double **A_inv,B[4]**d,T[4];
```

```
int k,b;
```

```
d=dmatrix(1,3,1,3);
```

```
A_inv=dmatrix(1,3,1,3);
```

```
A=matrix(1,3,1,3);
```

```
v=matrix(1,3,1,3);
```

```
w=vector(1,3);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

d[1][1]=1;
d[1][2]=0;
d[1][3]=0;
d[2][1]=0;
d[2][2]=1;
d[2][3]=0;
d[3][1]=0;
d[3][2]=0;
d[3][3]=1;

```

```

pfile=fopen("C:\\file_m4.dat","r");
for(b=1;b<=3;b++){
    for(k=1;k<=4;k++){fscanf(pfile,"%f",&X[b][k]);}
    fprintf(pfile,"\n");
    sprintf(ake,"%0.4f %0.4f %0.4f %0.4f ",X[b][1],X[b][2],X[b][3],X[b][4]);
    m_list1.AddString(ake);
}
fclose(pfile);

for(b=1;b<=3;b++){ for(k=1;k<=3;k++){A_inv[b][k]=A[b][k]=X[b][k];} }

for(k=1;k<=3;k++)B[k]=X[k][4];

svdcmp(A,3,3,w,v); //(u,m,n,w,v)

/*

for(i=1;i<=3;i++){

                                X=w[i];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        for(k=1;k<=3;k++)
            if((X-w[k])<0)n++;
            n=0;
            if(n==2){joy[3]=X;ff=i;n=0;}
    }
*/

printf(ake,"%4f %4f %4f ",w[1],w[2],w[3]);
m_list2.AddString(ake);

for(b=1;b<=3;b++){
    printf(ake,"%4f %4f %4f ",v[b][1],v[b][2],v[b][3]);
    m_list3.AddString(ake);
}

gaussj(A_inv,3,d,3);

for(k=1;k<=3;k++){
    T[k]=0.0;
    for(b=1;b<=3;b++){T[k]=+A_inv[k][b]*B[b];}
}

printf(ake,"%4f %4f %4f ",T[1],T[2],T[3]);
m_list4.AddString(ake);

free_dmatrix(d, 1, 3, 1, 3);
free_matrix(A,1,3,1,3);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

free_matrix(v,1,3,1,3);
free_vector(w,1,3);

}

```

%%%%ฟังก์ชันเหตุการณ์เพื่อดูพารามิเตอร์ที่4%%%%

```

void CMy3DExtractionDlg::OnRadio_m4()

```

```

{
m_list1.DeleteString(3);
m_list1.DeleteString(2);
m_list1.DeleteString(1);
m_list1.DeleteString(0);

m_list2.DeleteString(0);

m_list3.DeleteString(2);
m_list3.DeleteString(1);
m_list3.DeleteString(0);

m_list4.DeleteString(0);

```

```

char ake[50];

```

```

float X[5][5],**A,**v,*w;

```

```

double **A_inv,B[4],**d,T[4];

```

```

int k,b;

```

```

d=dmatrix(1,3,1,3);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

A_inv=dmatrix(1,3,1,3);
A=matrix(1,3,1,3);
v=matrix(1,3,1,3);
w=vector(1,3);

d[1][1]=1;
d[1][2]=0;
d[1][3]=0;
d[2][1]=0;
d[2][2]=1;
d[2][3]=0;
d[3][1]=0;
d[3][2]=0;
d[3][3]=1;

pfile=fopen("C:\\file_m4.dat","r");
for(b=1;b<=3;b++){
    for(k=1;k<=4;k++){fscanf(pfile,"%f",&X[b][k]);}
    fprintf(pfile,"\n");
    sprintf(ake,"%0.4f %0.4f %0.4f %0.4f ",X[b][1],X[b][2],X[b][3],X[b][4]);
    m_list1.AddString(ake);
}
fclose(pfile);

for(b=1;b<=3;b++){ for(k=1;k<=3;k++){A_inv[b][k]=A[b][k]=X[b][k];} }

for(k=1;k<=3;k++)B[k]=X[k][4];

svdcmp(A,3,3,w,v); //(u,m,n,w,v)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*
    for(i=1;i<=3;i++){

        X=w[i];

        for(k=1;k<=3;k++)

            if((X-w[k])<0)n--;

            n=0;

            if(n==2){joy[3]=X;ff=i;n=0;}

    }

*/

printf(ake,"%4f %4f %4f ",w[1],w[2],w[3]);
m_list2.AddString(ake);

for(b=1;b<=3;b++){

    printf(ake,"%4f %4f %4f ",v[b][1],v[b][2],v[b][3]);
    m_list3.AddString(ake);

}

gaussj(A_inv,3,d,3);

for(k=1;k<=3;k++){

    T[k]=0.0;

    for(b=1;b<=3;b++){T[k]=+A_inv[k][b]*B[b];}

}

printf(ake,"%4f %4f %4f ",T[1],T[2],T[3]);
m_list4.AddString(ake);

free_dmatrix(d, 1, 3, 1, 3);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
free_matrix(A,1,3,1,3);  
free_matrix(v,1,3,1,3);  
free_vector(w,1,3);  
  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้