การสร้างภาพสามมิติของอวัยวะจากกล้องเว็บแคม

3D Modeling Using Webcam

ปฏิภาณ เจ๊ะอื่อ

Patiparn Jejue

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2556

การสร้างภาพสามมิติของอวัยวะจากกล้องเว็บแคม

3D Modeling Using Webcam

ปฏิภาณ เจจือ

Patiparn Jejue

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างภาพสามมิติของอวัยวะจากกล้องเว็บแคม

3D Modeling Using Webcam

โดย

ปฏิภาณ เจจือ

Patiparn Jejue

อาจารย์ที่ปรึกษา

ผศ.ดร.ยุทธนา คิดใจเดียว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2556

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การสร้างภาพสามมิติของอวัยวะจากกล้องเว็บแคม

3D Modeling Using Webcam

ผู้จัดทำ นาย ปฏิภาณ เจจือ รหัสประจำตัว 53010901

ปริญญานิพนธ์นี้ผ่านการตรวจสอบจากอาจารย์ที่ปรึกษาแล้ว

.....

ผศ.ดร.ยุทธนา คิดใจเดียว

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis title	3D Modeling Using Web Cam
Student	Mr.Patiparn Jejue Student ID 53010901
Degree	Bachelor of Engineering
Program	Electronics Engineering
Year	2013
Thesis Advisor	Assistance Prof. Dr.Yutthana Kidjaidure

ABSTRACT

This thesis presents an instrument process image and reconstruct 3D model by Visual Hull method. We use web camera. Then MATLAB is used for processing images and reconstruction to 3D model that showed shape and resolution of the arm and showed on the monitor.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้จะไม่สามารถสำเร็จลุล่วงได้ ถ้าขาดการให้คำชี้แนะและปรึกษาจากท่านอาจารย์ที่ปรึกษาและเพื่อนทั้งหลายของข้าพเจ้า และขอขอบพระคุณการสนับสนุนทางด้านแนวคิดและเงินทุนในการจัดหาอุปกรณ์มาสร้างอุปกรณ์รวมทั้งชี้แนะแนวทางความคิดให้ข้าพเจ้า เมื่อข้าพเจ้าไม่สบายใจ ปริญญานิพนธ์ฉบับนี้ข้าพเจ้าทุ่มเทเวลา แรงกายและความคิดของข้าพเจ้า ถ้ามีข้อผิดพลาดประการใดในปริญญานิพนธ์ฉบับนี้ข้าพเจ้าขอน้อมรับคำติชมและกราบขออภัยมา ณ ที่แห่งนี้ด้วย

ปฏิภาณ เจจือ

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VI

สารบัญรูป.....VII

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์.....1

1.2 จุดมุ่งหมายและจุดประสงค์ของปริญญานิพนธ์.....1

1.3 ขอเขตของปริญญานิพนธ์.....1

1.4 วิธีการดำเนินงาน.....1

1.5 ประโยชน์ที่คาดว่าจะได้รับ.....1

1.6 ส่วนประกอบของปริญญานิพนธ์.....2

บทที่ 2 องค์ประกอบและหลักการทำงานของเครื่องตรวจหาเส้นเลือด.....3

2.1 กล้องเว็บแคม.....3

2.2 ทฤษฎีการประมวลผลภาพ.....3

2.2.1 การแปลงภาพสี 24 บิต เป็นภาพขาวดำ 8 บิต.....3

2.2.2 ทฤษฎีฮิสโตแกรมของสีและการปรับฮิสโตแกรมของสีให้เท่ากัน.....4

2.2.3 การใช้วิธีเทรชโชดิ้งเพื่อแปลงเป็นภาพไบนารี.....5

2.2.4 ระบบพิกัดพิเซลและพิกัดภาพ.....5

2.2.5 ระบบพิกัดและพิกัดโลก.....6

2.2.6 การปรับเทียบกล้อง.....7

2.2.7 ทฤษฎีการสร้างภาพสามมิติแบบวิชวลฮัล.....8

2.3 สเตปปิ้งมอเตอร์.....9

2.3.1 สเตปมอเตอร์แบบสองขั้ว.....9

2.3.2 สเตปมอเตอร์แบบหนึ่งขั้ว.....10

2.3.3 การไดร์ฟสเตปมอเตอร์.....10

2.3.3.1 แบบฟูลสเตปหนึ่งเฟส.....10

2.3.3.2 แบบฟูลสเตปสองเฟส.....10

2.3.3.3 แบบครึ่งสเตป.....11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

2.4 บอร์ดอาดูยโน่....12
บทที่ 3 การออกแบบอุปกรณ์และการเขียนโปรแกรมประมวลผลภาพ....13
3.1 ขั้นตอนการทำงานของอุปกรณ์....13
3.2 อุปกรณ์รับภาพ....14
3.3 อุปกรณ์ควบคุมการหมุน....14
3.3.1 รูปแบบการป้อนพลัส์ให้กับมอเตอร์....14
3.4 การประมวลภาพ....15
3.4.1 การปรับเทียบพิกัดของรูปภาพ....15
3.4.2 การบันทึกภาพ....16
3.4.3 แปลภาพเป็นภาพไบนารี....17
3.4.4 สร้างแบบจำลองสามมิติ....17
บทที่ 4 การทดลองและผลการทดลอง....18
4.1 การทดลองส่วนการควบคุมสเตปมอเตอร์....18
4.2 โปรแกรมส่วนการบันทึกภาพ....18
4.3 โปรแกรมคาลิเบรตกล้อง....19
4.4 ส่วนจำลองภาพสามมิติ....19
บทที่ 5 วิเคราะห์และวิจารณ์การทดลอง....21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

รูปที่ หน้า
2.1 รูปแบบการกระตุ้นสเตปมอเตอร์แบบฟูลสเตปหนึ่งเฟส....10
2.2 รูปแบบการกระตุ้นสเตปมอเตอร์แบบฟูลสเตปสองเฟส....11
2.3 รูปแบบการกระตุ้นสเตปมอเตอร์แบบครึ่งสเตป....11
3.1 ตารางแสดงการป้อนแรงดันให้กลับสเตปมอเตอร์....14
4.1 องศาการหมุนของสเตปมอเตอร์....18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 กล้องเว็บแคม....3

2.2 ลูกบาศก์สี RGB....4

2.3 ตัวอย่างกราฟฮิสโตแกรมของรูปภาพ....4

2.4 กราฟความสัมพันธ์ระหว่างความเข้มข้นอินพุทกับเอาท์พุท....5

2.5 ตัวอย่างภาพไบนารี....5

2.6 ความสัมพันธ์ระหว่างพิกัดรูปถ่ายและพิกัดของวัตถุที่ถ่ายจากจุด O....6

2.7 ภาพของตารางหมากรุกที่นิยมนำมาใช้ในการปรับเทียบกล้อง....8

2.8 เทคนิคการสร้างภาพสามมิติแบบวิชวลฮัล....9

2.9 หลักการทำงานของสเตปปิ้งมอเตอร์....9

2.10 รูปบอร์ดอาดูยโน่ รุ่น UNO R3....12

3.1 ลำดับการทำงานของโปรแกรมบันทึกภาพ....13

3.2 อุปกรณ์การรับภาพ....14

3.3 ลำดับการทำงานของโปรแกรมปรับเทียบกล้อง....15

3.4 ภาพกล่องที่ใช้ในการหาพิกัดกล้อง และพิกัดกล้องที่หาได้....16

3.5 แสดงรูปภาพที่บันทึก....16

3.6 ภาพไบนารี....17

3.7 ลำดับการทำงานของโปรแกรมสร้างวัตถุ 3 มิติ....18

4.1 GUI ของโปรแกรมบันทึกภาพ....18

4.2 ภาพแสดงการคาลิเบรตกล้องและผลลัพธ์ของโปรแกรม....19

4.3 ภาพแสดงการสร้างโมเดลสามมิติโดยใช้จำนวณมุมต่างๆกัน....20

4.4 ภาพแสดงผลลัพธ์ภาพถ่ายแขนของผู้เข้าร่วมการทดสอบ....20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

ปัจจุบันเทคโนโลยีทางด้านอิเล็กทรอนิกส์สามารถอำนวยความสะดวกในการทำงาน และการสร้างสรรค์สิ่งใหม่ๆ โดยการสร้างโมเดลสามมิติของอวัยวะ สามารถนำไปประยุกต์ใช้ในการช่วยระบุตำแหน่งของสิ่งแปลกปลอมที่อยู่บริเวณพื้นผิวของอวัยวะโดยการนำภาพถ่ายสองมิติของวัตถุจริงได้

1.2 จุดมุ่งหมายและจุดประสงค์ของปริญญานิพนธ์

1.ศึกษาการวิธีประมวลผลภาพแบบต่างๆ และสร้างโมเดลสามมิติโดยวิธีวิชวลฮัล

2.สร้างอุปกรณ์ช่วยเหลือทางการแพทย์ที่มีต้นทุนการผลิตในราคาถูก และมีประสิทธิภาพในการสร้างวัตถุสามมิติได้จริง

3.ศึกษาคำสั่งและการเขียนโปรแกรมการประมวลภาพโดยใช้โปรแกรม MATLAB

1.3 ขอบเขตของปริญญานิพนธ์

สามารถสร้างโมเดลสามมิติจำลองของอวัยวะโดยมีรูปทรงและรายละเอียดของวัตถุใกล้เคียงกับวัตถุจริง โดยใช้ภาพถ่ายสองมิติจากกล้อง Web Camera

1.4 วิธีการดำเนินงาน

1. ศึกษาวิธีการประมวลผลภาพในรูปแบบต่าง เพื่อการนำมาปรับใช้ให้เหมาะสมกับประเภทงานที่ต้องการใช้
2. ออกแบบการทำงานของอุปกรณ์ และลำดับการทำงานของโปรแกรม
3. สร้างอุปกรณ์จำลอง และเขียนชุดคำสั่งในการประมวลผลภาพเพื่อใช้ในการค้นหาสร้างวัตถุสามมิติ
4. วิเคราะห์ผลการทดลองและแก้ไขความผิดพลาดที่เกิดขึ้น

1.5 ประโยชน์ที่คาดว่าจะได้รับ

สามารถนำอุปกรณ์มาใช้สร้างวัตถุสามมิติโดยใช้ภาพถ่ายจาก Web Camera โดยมีรูปทรงใกล้เคียงกับวัตถุจริงที่เรานำมาเป็นแบบในการบันทึกภาพ

1.6 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้แบ่งออกเป็น 5 บทคือ

บทที่ 1 กล่าวถึงความเป็นมาและความสำคัญของโครงงาน จุดประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินงาน ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริญญานิพนธ์

บทที่ 2 กล่าวถึงองค์ประกอบและทฤษฎีหลักการทำงานของอุปกรณ์สร้างภาพสามมิติ

2.1 กล้องเว็บแคม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ทฤษฎีการประมวลผลภาพ

2.2.1 การแปลงภาพสี 24 บิต เป็นภาพขาวดำ 8 บิต

2.2.2 ทฤษฎีฮิสโตแกรมของสีและการปรับฮิสโตแกรมของสีให้เท่ากัน

2.2.3 การใช้วิธีเทรชโชดิ้ง (Thresholding) เพื่อแปลงภาพเป็นไบนารี

2.2.4 ระบบพิกัดพิกเซล และพิกัดภาพ (Pixel and Image Coordinate)

2.2.5 ระบบพิกัดและพิกัดโลก (Image and World Coordinate System)

2.2.6 การปรับเทียบกล้อง (Camera Calibration)

2.2.7 ทฤษฎีการสร้างภาพ 3 มิติ แบบวิชวลฮัล (Visual Hull)

2.3 สเตปปิ้งมอเตอร์

2.3.1 สเตปมอเตอร์แบบสองขั้ว (Bipolar)

2.3.2 สเตปมอเตอร์แบบหนึ่งขั้ว (Unipolar)

2.3.3 การไดร์ฟสเตปปิ้งมอเตอร์

2.3.3.1. แบบฟูลสเตป 1 เฟส (Full Step 1 Phrase)

2.3.3.2. แบบฟูลสเตป 2 เฟส(Full Step 2 Phase)

2.3.3.3. แบบครึ่งสเต็ป (Half Step)

2.4 บอร์ดอาดูยโน่

บทที่ 3 กล่าวถึงรายละเอียดของอุปกรณ์และการลำดับการทำงานของโปรแกรมในการบันทึกรูปและสร้างโมเดลสามมิติ

บทที่ 4 กล่าวถึงการทดลองเพื่อสร้างโมเดลของแขน

บทที่ 5 บทวิจารณ์และสรุปโครงงาน และข้อเสนอแนะเพื่อพัฒนาปรับปรุงโครงงานในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
องค์ประกอบและหลักการทำงานของเครื่องสร้างภาพสามมิติ

การสร้างภาพสามมิติ เป็นการจำลองวัตถุขึ้นมาโดยวัตถุที่จำลองขึ้นมานั้นจำเป็นต้องมีรูปทรงและรายละเอียดใกล้เคียงกับวัตถุจริง โดยในปัจจุบันการสร้างวัตถุสามมิติสามารถนำไปประยุกต์ในการพิมพ์รูปเป็นโมเดลสามมิติเพื่อสร้างวัตถุเลียนแบบขึ้นมา หรือการนำไปประยุกต์ใช้กับการสร้างอนิเมชั่นเพื่อความสนุกสนานหรือจำลองสถานการณ์ขึ้นมาในคอมพิวเตอร์

2.1 กล้องเว็บแคม

รูปที่ 2.1 กล้องเว็บแคม

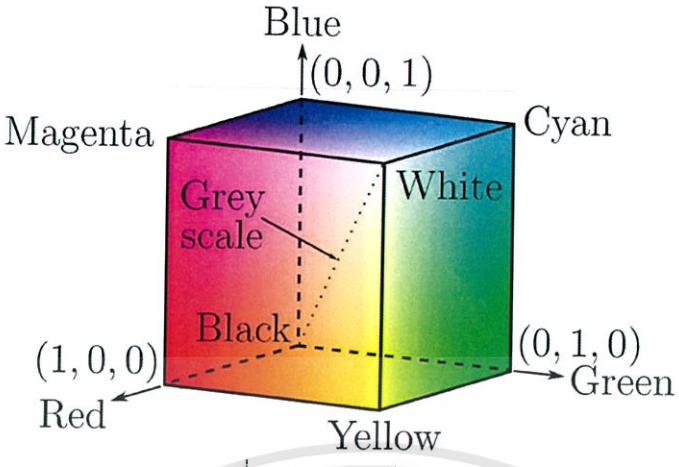
โดยทั่วไปแล้วกล้องเว็บแคมสามารถหาซื้อได้ง่ายตามท้องตลาดทั่วไป และในปัจจุบันกล้องเว็บแคมได้มีศักยภาพที่สูงขึ้น กล่าวคือสามารถบันทึกภาพที่มีความละเอียดสูงขึ้น

2.2 ทฤษฎีการประมวลผลภาพ

2.2.1 การแปลงภาพสี 24 บิต เป็นภาพขาวดำ 8 บิต

ภาพสีที่ที่เราเห็นกันทั่วไปนั้น ระบบคอมพิวเตอร์จะมองเป็นเมตริกคณิตศาสตร์ ซึ่งในภาพจะมีความลึกสี (Bit Depth) โดยภาพสี 24 บิต นั้น คือภาพ 8 บิต ที่แบ่งเป็น 3 ชาแนล คือ ชาแนลสีแดง ชาแนลสีเขียว ชาแนลสีน้ำเงิน การใช้พื้นที่ในการเก็บข้อมูล 8 บิตนั้น สามารถไล่ลำดับเฉดสีได้ถึง 256 เฉดสี ทำให้ภาพ 24 บิตนั้นสามารถผสมสีได้ถึง 256×256×256 = 16,777,216 เฉดสี โดยทั่วไปจะเรียกสี 24 บิตนี้ว่าเป็นสีเสมือนจริง เมื่อแปลงภาพสีเป็นขาวดำแล้ว แต่ละลำดับข้อมูลของชาแนลสีทั้ง 3 ชาแนลก็จะมีค่าเท่ากัน จึงเหลือแค่ลำดับเฉดสีเทาเพียง 256 เฉดสีเท่านั้น

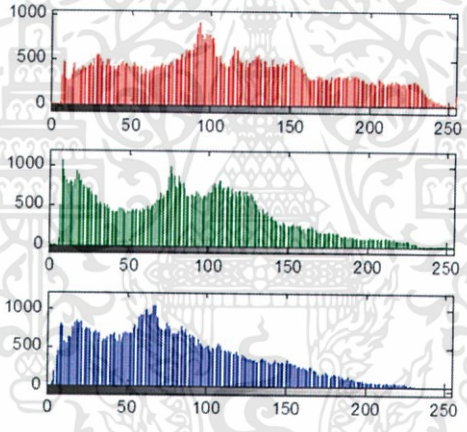
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ลูกบาศก์สี RGB

2.2.2 ทฤษฎีฮิสโตแกรมของสีและการปรับฮิสโตแกรมของสีให้เท่ากัน

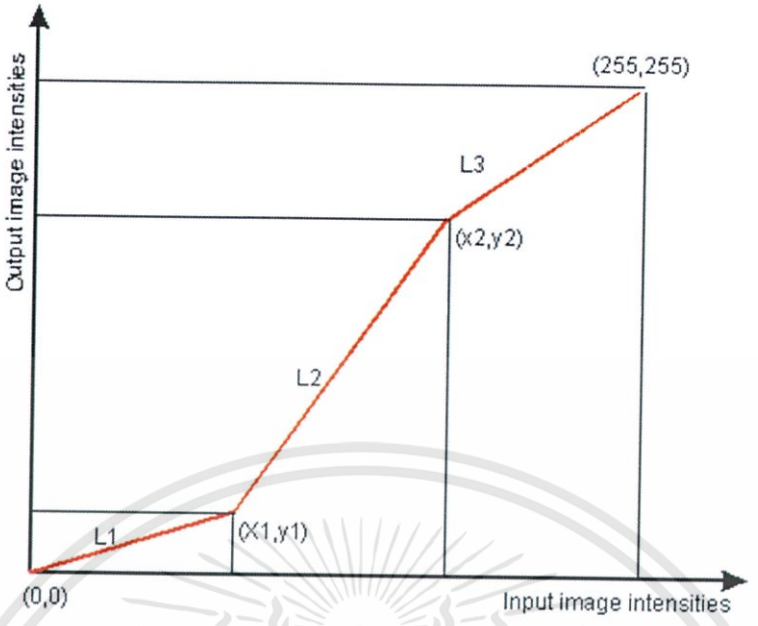
ฮิสโตแกรมของภาพเป็นตัวบ่งบอกความหนาแน่นของสีที่อยู่ในแต่ละลำดับข้อมูล โดยกราฟในแนวแกน Y คือปริมาณของข้อมูล ส่วนในแนวแกน X คือลำดับของข้อมูล



รูปที่ 2.3 ตัวอย่างกราฟฮิสโตแกรมของรูปภาพ

การปรับฮิสโตแกรมของรูปภาพให้เท่ากันนั้น ในมุมมองภาพสี 24 bit จะเป็นการปรับให้ลำดับของข้อมูลสีในภาพมีการกระจายตัวมากขึ้นอยู่ในค่าที่ต่างๆกัน ทำให้สีไม่กระจุกอยู่ในค่าๆเดียว ทำให้คอนทราสของภาพนั้นมีความแตกต่างกันชัดเจนมากขึ้น โดยความชันของกราฟหรือความแตกต่างของกราฟตรงกลางคือความเข้มข้นของคอนทราส

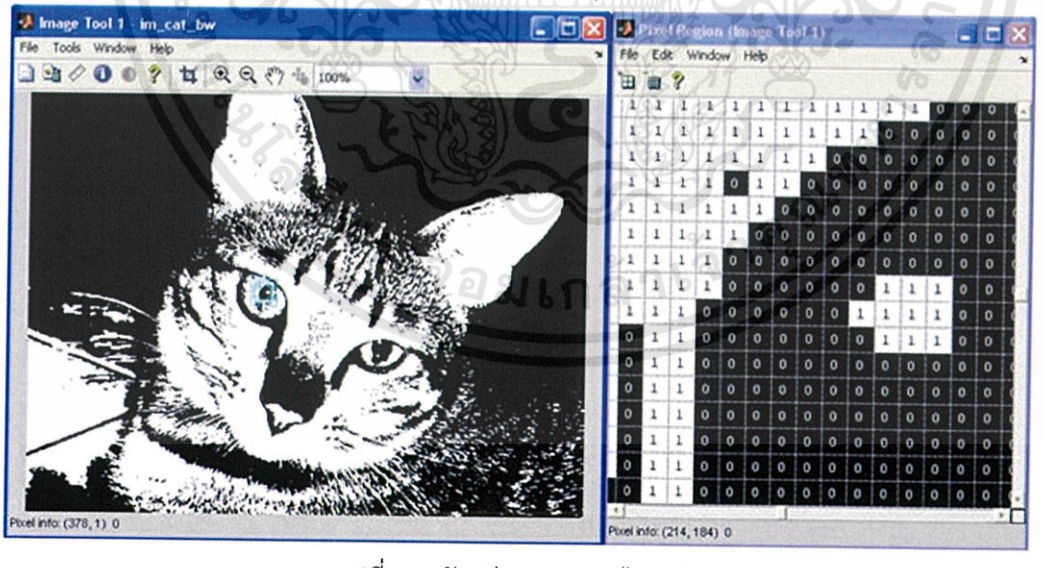
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 กราฟความสัมพันธ์ระหว่างความเข้มข้นอินพุทกับเอาท์พุท

2.2.3 การใช้วิธีเทรชโชดิ้ง (Thresholding) เพื่อแปลงเป็นภาพไบนารี

การ**เทรชโชดิ้ง** (Thresholding) คือการทำภาพ 8 บิต ให้เหลือเพียง 2 บิต โดยจะมีค่า คอนทราสสูงมาก มักใช้เพื่อการแบ่งแยกรูปภาพ (Image Segmentation) โดยทำให้วัตถุภาพในภาพที่มีรอยต่อของพิกเซลต่างกันนั้นสามารถเห็นได้ชัดเจนมากขึ้น โดยภาพจะมีข้อมูลแค่ 0 กับ 1 หรือสี ดำ กับ สีขาว เท่านั้นทำให้เห็นความแตกต่างระหว่างวัตถุชัดเจนขึ้น



รูปที่ 2.5 ตัวอย่างภาพแบบไบนารี

2.2.4 ระบบพิกัดพิกเซล และพิกัดภาพ (Pixel and Image Coordinate)

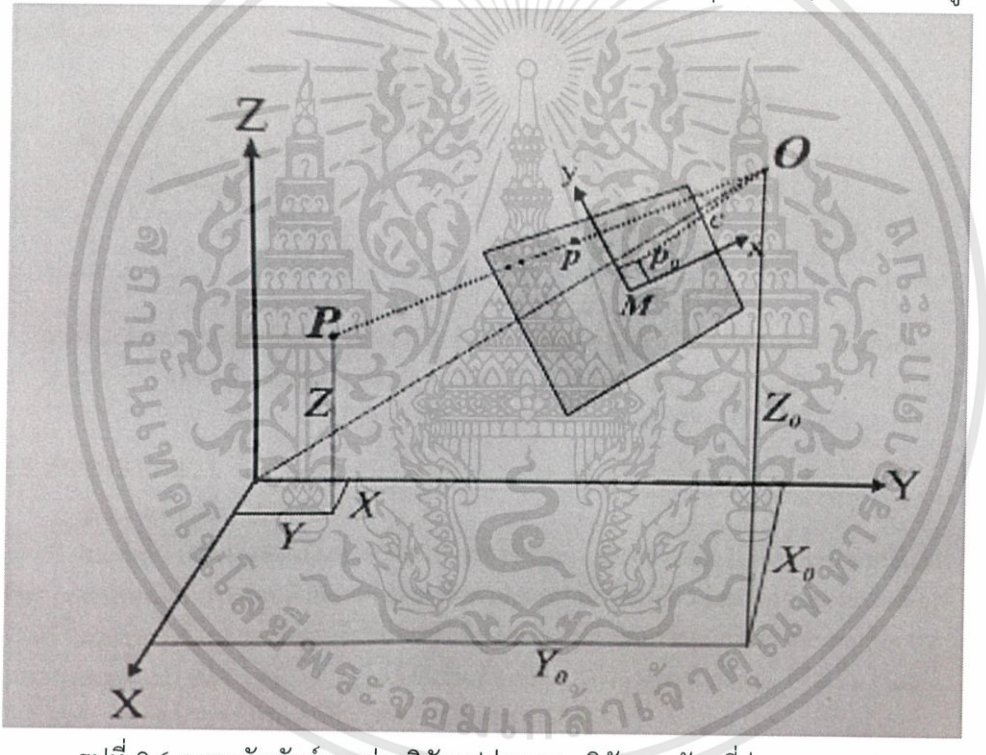
ระบบพิกัดพิกเซล ต่างไปจากพิกัดภาพ คือเป็นชุดเลขจำนวนเต็มแบบอาเรย์ 2 มิติ โดยทั่วไปแล้ว ภาพดิจิตอลนั้นจะเป็นค่าตัวแปรลำดับแบบ 2 มิติ ซึ่งจะมีค่าสีเก็บไว้ ความกว้างและสูงของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะแทนด้วยระบบแกน XY โดยมีจุดเริ่มที่จุดกำเนิด (0,0) ที่จุดซ้ายบนส่วนพิกัดของภาพจะเริ่มจากกึ่งกลางของภาพนั้นๆ และมีจุดหลักสำคัญด้วย ในการขึ้นรูปใหม่จะใช้จุดกลางภาพ หรือจุด M และมีจุดสำคัญ คือจุด p_0 ซึ่งได้มาจากการปรับค่ากล้อง

2.2.5 ระบบพิกัดและพิกัดโลก (Image and World Coordinate System)

การที่จะสร้างรูปทรงใหม่ขึ้นมาจากรูปภาพนั้น จำเป็นต้องมีความรู้ในการกำหนดทิศทางให้กล้อง และต้องรู้องค์ประกอบภายใน(Interior) หรือคุณสมบัติของกล้องที่จะใช้ เช่น ความยาวโฟกัส ส่วนใหญ่มักเป็นค่าคงที่ และองค์ประกอบภายนอก(Exterior) หรือตำแหน่งของกล้องตอนที่ถ่ายภาพ ขณะที่รูปถ่าย 1 รูปมักมีความหลากหลายของตัวแปรต่างๆใน 1 ช่วงเวลาที่ถ่ายเท่านั้น ซึ่งจะประกอบไปด้วยตำแหน่งและทิศทางในการถ่ายภาพ ถ้าไม่สนใจเรื่องการปรับแต่งเลนส์แล้ว ความสัมพันธุ์ระหว่างจุด P บนระบบพิกัดภาพ(แกน x,y) และพิกัด 3 มิติของวัตถุ (แกน x,y,z) แสดงดังรูป



รูปที่ 2.6 ความสัมพันธ์ระหว่างพิกัดรูปถ่าย และพิกัดของวัตถุที่ถ่ายจากจุด O

ทุกๆจุดบนวัตถุที่ถูกฉายลงไปจะเทียบได้กับจุดบนภาพเพียง 1 จุด แบบจุดต่อจุด การแปลงจุดบนภาพไปเป็นจุดบนวัตถุเป็นจำนวนมหาศาลตามแนวการฉายภาพ แต่สามารถรู้ได้ว่าจุดใดบนภาพเป็นพื้นที่ว่างเปล่าในวัตถุจริงได้จากการใช้รูปอื่นๆเพิ่ม หรือกล่าวอีกแบบหนึ่ง คือ ต้องการข้อมูลบนแกน Z ของวัตถุนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 การปรับเทียบกล้อง (Camera Calibration)

การปรับเทียบกล้องเป็นการหาประมาณเมทริกซ์ M จากจุดในฉาก 3 มิติที่รู้ตำแหน่งและภาพที่เกิดขึ้นจากจุด จากนั้นทำการหาพารามิเตอร์ภายในและพารามิเตอร์ภายนอกพิจารณาการปรับเทียบโดยใช้กล้องตัวเดียว

ในการหาเมทริกซ์ M เราทำการหาจุดในฉากที่เราทราบพิกัด X = [x, y, z]^{T} และจุดในภาพ 2 มิติที่สอดคล้องกัน [u, v]^{T} ซึ่งจะได้สมการที่อยู่ในรูปสมการ (13.60) ดังนี้

$$\begin{bmatrix} \alpha u \\ \alpha v \\ \alpha \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \alpha u \\ \alpha v \\ \alpha \end{bmatrix} = \begin{bmatrix} m_{11}x + & m_{12}x + & m_{13}x + & m_{14} \\ m_{21}x + & m_{22x+} & m_{23}x + & m_{24} \\ m_{31}x + & m_{32}x + & m_{33}x + & m_{34} \end{bmatrix}$$

ทำการนอร์มอลไลซ์ให้สมาชิกที่ 3 เป็น 1 เราจะได้สมการ

$$u(m_{31}x + m_{32}y + m_{33}z + m_{34}) = m_{11}x + m_{12}y + m_{13}z + m_{14}$$

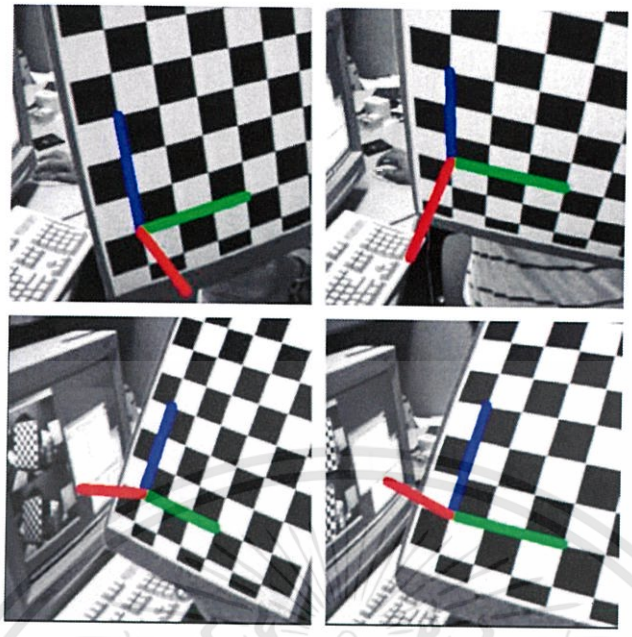
$$v(m_{31}x + m_{32}y + m_{33}z + m_{34}) = m_{11}x + m_{12}y + m_{13}z + m_{14}$$

ดังนั้นเราจะได้สมการเชิงเส้น 2 สมการสำหรับจุดในฉาก 3 มิติหนึ่งจุดและจุดในภาพ 2 มิติที่สอดคล้องกัน ทำการเขียนสมการ ใหม่ในรูปของเมทริกซ์ได้เป็น

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

$$\mathbf{AM = 0}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 ภาพของตารางหมากรุกที่นิยมนำมาใช้ในการปรับเทียบกล้อง

จะเห็นว่าเรามีตัวแปรที่ไม่ทราบค่า 11 ตัวแปร แทนที่จะเป็น 12 ตัวแปรเนื่องจากแฟคเตอร์การสเกลที่ไม่สามารถทราบค่าได้ ในการแก้สมการโฮโมจีเนียสเราต้องใช้อย่างน้อย 6 จุด ถ้ามีมากกว่า 6 จุด จะได้สมการ Over-determined ซึ่งสามารถแก้ได้โดยใช้วิธี Least Square หรือใช้การหา Singular Value Decomposition (SVD) ของ A เมทริกซ์ M ได้จากแถวสุดท้ายของ V เมื่อได้เมทริกซ์ M แล้วขั้นตอนต่อไปทำการแยกพารามิเตอร์ภายใน (Intrinsic Parameter) เนื่องจาก

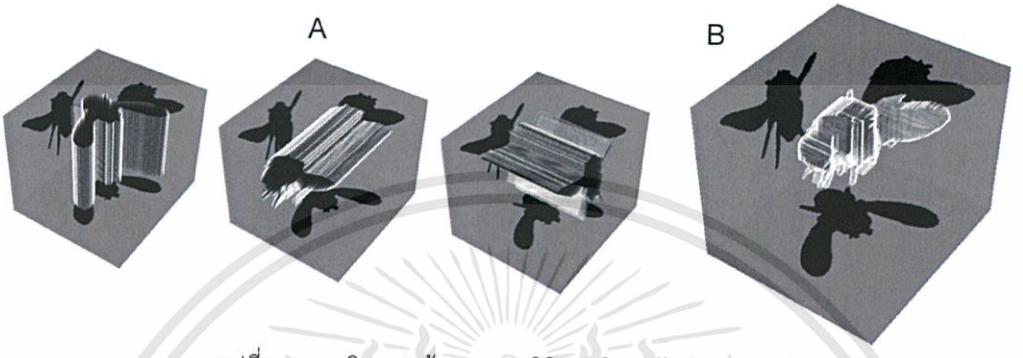
$$M = [KR \mid -KRT] = [A \mid b]$$

ดังนั้นเมทริกซ์ย่อย 3x3 แทนด้วย A เวคเตอร์คอลัมน์ขวามือสุดแทนด้วย b เวคเตอร์ที่มีพารามิเตอร์ระยะเคลื่อนที่สามารถแยกออกมาได้ง่าย โดย $t = -A^{-1}b$ เวคเตอร์ t ให้ข้อมูลที่บอกถึงตำแหน่งกึ่งกลางของระนาบภาพ (Image Plane) จากนั้นพิจารณา $A = KR$ โดยที่ K เป็นเมทริกซ์สามเหลี่ยมบน และเมทริกซ์ R เป็นเมทริกซ์ออร์โธกอนอล การแยกเมทริกซ์ K และ R สามารถทำได้โดยใช้เทคนิค QR Decomposition สำหรับเมทริกซ์ A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.7 ทฤษฎีการสร้างภาพ 3 มิติ แบบวิชวล ฮัล (Visual Hull)

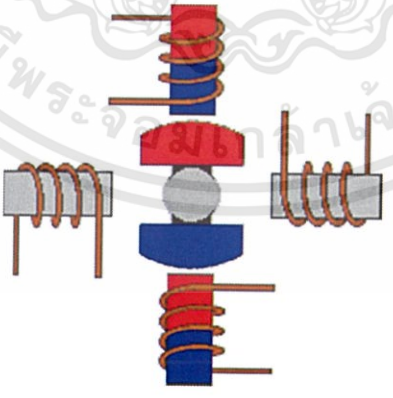
Visual Hull เป็นเอกลักษณ์ทางเรขาคณิตที่สร้างโดยเทคนิคการสร้างภาพจากเงา เทคนิคนี้จะทำการแยกภาพเงาของวัตถุออกจากพื้นหลัง ภายใต้สมมติฐานนี้ ภาพต้นฉบับสามารถเข้าสู่พื้นหลัง/พื้นหน้าของภาพไบนารี่หรือภาพเงาได้ โดยภาพเงานั้นเป็นภาพฉาย 2 มิติของวัตถุ 3 มิติเบื้องหน้าที่สอดคล้องกัน



รูปที่ 2.8 เทคนิคการสร้างภาพ 3 มิติแบบวิชวลฮัล (Visual Hull)

2.3 สเตปปิ้งมอเตอร์

สเตปปิ้งมอเตอร์ หรือสเตปเปอร์มอเตอร์ คือมอเตอร์ที่สามารถควบคุมองศาของการหมุนได้อย่างแม่นยำ โดยบังคับให้แม่เหล็กถาวรบนแกนโรเตอร์หมุนไปตามทิศการบังคับของขดลวดที่ติดตั้งบนสเตเตอร์ของมอเตอร์ ยิ่งมอเตอร์มีซี่แม่เหล็กถี่มากเท่าใด ก็จะสามารถควบคุมการหมุนของมอเตอร์ได้ละเอียดยิ่งขึ้น วิธีการควบคุมคือการป้อนแรงดันไฟฟ้าเข้าไปที่ขดลวดบนสเตเตอร์ของมอเตอร์ โดยจะทำให้เกิดสนามแม่เหล็กเหนี่ยวนำให้แกนแม่เหล็กถาวรหันทิศที่ตรงข้ามกับขั้วแม่เหล็กไฟฟ้าที่เกิดขึ้น



รูปที่ 2.9 หลักการทำงานของสเตปปิ้งมอเตอร์

ประเภทของสเตปปิ้งมอเตอร์ แบ่งได้เป็น 2 ประเภท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 สเตปมอเตอร์แบบสองขั้ว (Bipolar)

จะมีการพันขดลวดหนึ่งขดในขั้วแม่เหล็กของสเตเตอร์ โดยขั้วแม่เหล็กที่เกิดขึ้นที่สเตเตอร์จะถูกกำหนดโดยทิศทางของการไหลของกระแสไฟฟ้า ซึ่งสามารถทำให้เกิดขั้วแม่เหล็กในทิศทางตรงกันข้ามได้เพียง การกลับทิศทางของการไหลในกระแสไฟฟ้า โดยมาจากการควบคุมของวงจรสวิทชิ่งให้กลับขั้วไฟฟ้า

2.3.2 สเตปมอเตอร์แบบหนึ่งขั้ว (Unipolar)

มีขดลวด 2 ขด บนแต่ละขั้วแม่เหล็กของสเตเตอร์ ทำให้แต่ละขดลวดเกิดขั้วแม่เหล็กในทิศทางตรงกันข้าม การกลับทิศทางขั้วแม่เหล็กทำได้โดยใช้วงจรสวิทชิ่งให้สลับหนึ่งไปยังอีกขั้วหนึ่งแทน

2.3.3การไดร์ฟสเตปปิ้งมอเตอร์

ทำได้โดยการป้อนแรงดันเข้าไปที่ขั้วต่างๆของสเตปปิ้งมอเตอร์ มีหลายชนิด

2.3.3.1แบบฟูลสเตป 1 เฟส (Full Step 1 Phrase)

จะเป็นการกระตุ้นแบบที่ง่ายที่สุด ซึ่งจะทำการกระตุ้นขดลวดทีละขดในเวลาหนึ่งๆเรียงกันไป ตัวอย่างเช่น ขดที่ 1 , 2 , 3 , 4 ,1, 2 , 3 , 4 เป็นลำดับ หรือ ขด 1 , 4 , 3 , 2 , 1 , 4 , 3 , 2 เป็นลำดับกันไป ทั้งนี้ขึ้นอยู่กับทิศทางที่เราต้องให้มอเตอร์หมุนไป วงจรที่นำมากระตุ้นนั้นจะมีราคาค่อนข้างจะถูกกว่าและง่ายกว่า

เราสามารถเขียนขั้นตอนการทำงานเป็นตารางออกมาได้ดังนี้

Step No.	Phase 1	Phase 2	Phase 3	Phase 4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	0	0	0
6	0	1	0	0

ตารางที่ 2.1 รูปแบบการสเตปมอเตอร์แบบฟูลสเตปหนึ่งเฟส

2.3.3.2 แบบฟูลสเตป 2 เฟส (Full Step 2 Phase)

แบบนี้ก็จะคล้ายกับการกระตุ้นในแบบฟูลสเตป 1 เฟส แต่จะต่างกันตรงที่ แบบ 2 เฟส จะกระตุ้นทีละ 2 ขด ที่อยู่ใกล้กันใน เวลาเดียวกัน และจะเรียงลำดับกันไป ดังเช่นแบบเดียวกับแบบฟูลสเตป 1 เฟส เช่น 12,23,34.41,12,23,34,41 เรียงลำดับกันไปเรื่อยๆ หรือจะเป็น 14,43,32,21,14,43,32,21 เรียงกันไปเช่นกัน ข้อดีข้อเสียของแบบ 2 เฟส มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดี การที่เราจะเพิ่มจำนวนขดลวดที่ถูกกระตุ้นจะทำให้แรงบิดได้มากกว่า แบบฟูลสเตป 1 เฟส ซึ่งโรเตอร์จะหมุนด้วยแรง ดึงแบเต็มๆแรงจาก ทั้ง 2 ขดลวดที่กระตุ้นพร้อมกัน

ข้อเสีย แบบ 2 เฟส จะกระตุ้นขดลวดนั้นต้องใช้กำลังไฟมากขึ้นเป็น 2 เท่าของแบบฟูลสเตป 1 เฟส เราสามารถเขียนลำดับการกระตุ้นของขดลวดแบบ 2 เฟส ได้ดังในตารางต่อไปนี้

Step No.	Phase 1	Phase 2	Phase 3	Phase 4
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1
5	1	1	0	0
6	0	1	1	0

ตารางที่ 2.2 รูปแบบการสเตปมอเตอร์แบบฟูลสเตปสองเฟส

2.3.3.3 แบบครึ่งสเต็ป (Half Step)

เป็นรูปแบบผสมผสานของการกระตุ้นระหว่าง แบบเวฟ กับ แบบ 2 เฟส เพื่อให้จำนวนรอบของสเต็ปให้ มากขึ้นเป็น 2 เท่า ซึ่งในระบบนี้จะทำการกระตุ้นขดลวดเรียงกันไปเรื่อยๆเป็นลำดับ ดังจะยกตัวอย่างต่อไปนี้ 1,12,2,23,3,34,4,41,1,12,2,23,3,34,4,41,1 เป็นลำดับ ถ้าจะกลับทิศทางการหมุนจะเป็นดังนี้ 1,41,4,43,3,32,2,21,1,41,4,43,3,32,2,21,1 เป็นลำดับ ข้อดีและข้อเสียของการกระตุ้นแบบครึ่งสเต็ปมีดังนี้

ข้อดี การกระตุ้นแบบนี้จะให้แรงบิดที่เพิ่มมากขึ้น เนื่องจากช่วงสเต็ปที่มีระยะสั้นลงอีก แต่ละ สเต็ปเกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกันเป็นผลให้ค่าตำแหน่งความถูกต้องมากขึ้นไปด้วย

ข้อเสีย ต้องจ่ายกำลังไฟเป็น 2 เท่าของแบบเวฟหรือจะใช้เท่ากับแบบ 2เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นเราสามารถนำลำดับการทำงานของ แบบครึ่งเฟส ในรูปของตารางได้ดังนี้

Step No.	Phase 1	Phase 2	Phase 3	Phase 4
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1
9	1	0	0	0
10	1	1	0	0

ตารางที่ 2.3 รูปแบบการกระตุ้นสเตปมอเตอร์แบบครึ่งสเตป

2.4 บอร์ดอาดูยโน่ (Arduino board)

Arduino เป็นระบบที่ใช้ในการพัฒนาอุปกรณ์อิเล็กทรอนิกส์ต้นแบบ ซึ่งออกแบบให้ใช้งานง่ายทั้งฮาร์ดแวร์ และซอฟแวร์ Arduino มีผู้ริเริ่มเป็นชาวอิตาเลียน เริ่มต้นในปี 2005 ผู้ริเริ่ม ชื่อว่า Massimo Banzi และ David Cuartielles ซึ่งตั้งใจสร้างอุปกรณ์ประเภทไมโครคอนโทรลเลอร์ราคาถูกที่นักเรียน นักศึกษาสามารถเข้าถึง และซื้อหามาเป็นเจ้าของได้

นอกจากจะตั้งใจให้ราคาของอุปกรณ์นั้นถูกเมื่อเทียบกับไมโครคอนโทรลเลอร์ตระกูลอื่นๆ ในท้องตลาดแล้ว พวกเขายังตั้งใช้ให้ Arduino สามารถพัฒนาโดยโปรแกรมที่ "แจกฟรี" ภายใต้เงื่อนไขในการใช้งานลักษณะ Open Souce ดังนั้นจึงเลือกใช้การพัฒนาบนพื้นฐานของระบบ Wiring

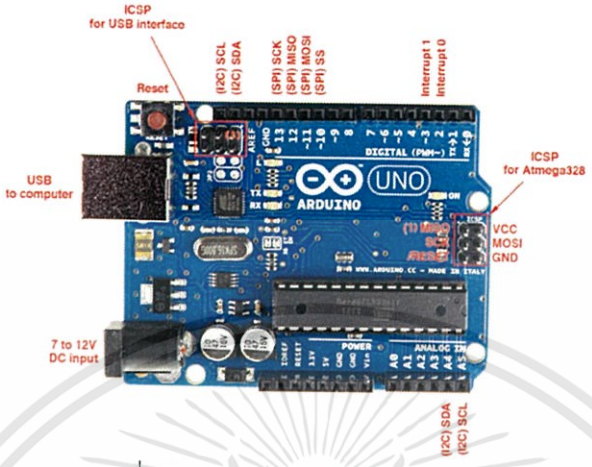
ไมโครคอนโทรลเลอร์คืออะไร ไมโครคอนโทรลเลอร์เปรียบเหมือนกับสมองของมนุษย์ คือมีหน้าที่คิดคำนวณทางคณิตศาสตร์ คำนวณทางลอจิก สั่งการ มีส่วนความจำ เพื่อใช้เป็นข้อมูลในการคำนวณ หรือประมวลผลต่างๆ "แต่จะไม่สามารถทำงานได้เอง" โดยไม่มีมือ เท้า แขน ขา หรือ ตา หู จมูก ซึ่งเปรียบได้กับอุปกรณ์ส่วนควบ (Accessories) อื่น เช่น เซนเซอร์ มอเตอร์ ระบบสื่อสารผ่านอินเตอร์เน็ต ระบบแสดงผลผ่านจอภาพ เป็นต้น ดังนั้นโดยสรุปคือ ไมโครคอนโทรลเลอร์จะทำหน้าที่ในการคิดคำนวณ รับค่าจากระบบวัดผลภายนอก เข้ามาประมวลผล เพื่อสั่งการตอบสนองออกไปที่อุปกรณ์ต่อเชื่อมอื่นๆ ตัวมันเองเดียวๆ จะทำอะไรไม่ได้มากไปกว่าการคิด

ไมโครคอนโทรเลอร์ในปัจจุบันก็มีอยู่หลายยี่ห้อ เช่น PIC ของบริษัทไมโครชิพ Z80 MCS-51 ARM-Cortex AVR และ อื่นๆอีกมากครับ Arduino ก็เป็นไมโครคอนโทรเลอร์แบบหนึ่งที่มีเอกลักษณ์เฉพาะตัวที่ต่างจากยี่ห้ออื่นๆ คือ การเป็น Open Source ซึ่งทำให้ได้เปรียบเรื่องราคา และจำนวนผู้ใช้งานทั่วโลก สาเหตุที่ราคาถูกก็สามารถสรุปได้ดังนี้

- ระบบเป็นแบบ Open Source ไม่มีลิขสิทธิ์ในการนำไปใช้งานต่อเชิงพาณิชย์ แถมแจกไฟล์ที่ใช้ในการสร้างต้นแบบให้ฟรีๆ ทำให้ประเทศผู้ผลิตอย่างจีนสามารถนำไปผลิตได้ในราคาสบายๆ โดยไม่ต้องกังวลเรื่องค่าลิขสิทธิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ซอฟแวร์ หรือ Arduino IDE ที่ใช้ในการพัฒนายังแจกให้ฟรี ดาว์นโหลดกันได้อย่างถูกกฎหมาย เอาไปใช้งานต่อ สร้างผลิตภัณฑ์แล้วขายต่อ ก็ไม่ต้องเสียเงินให้แบบไมโครคอนโทรลเลอร์เจ้าอื่นๆ
- มีซอฟแวร์ แบบฟรีๆ โดยผู้ที่สนใจในพัฒนาเช่นกัน



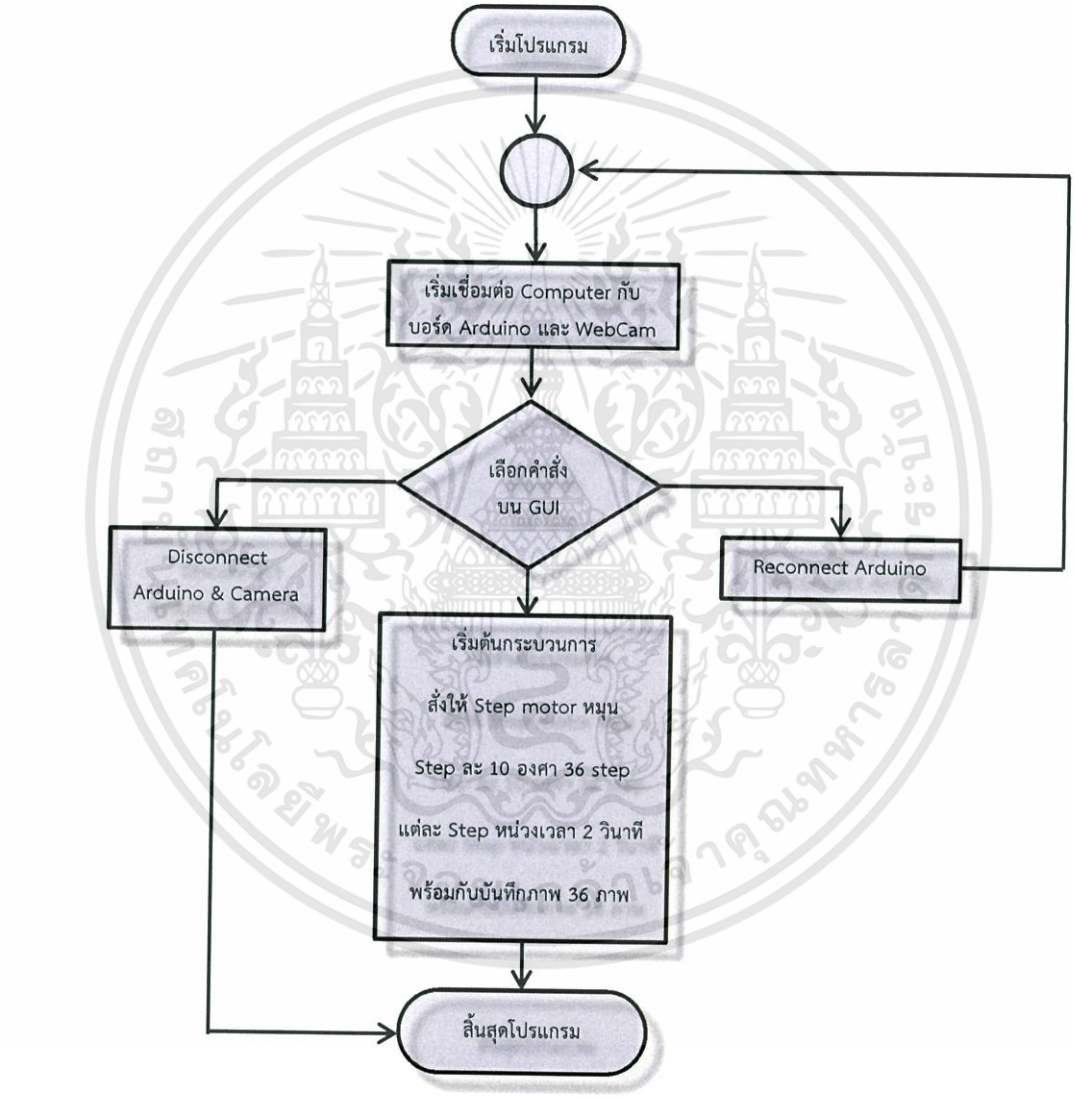
รูปที่ 2.10 รูปบอร์ดอาดูยโน่ รุ่น UNO R3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบอุปกรณ์และการเขียนโปรแกรมประมวลผลภาพ

3.1 ขั้นตอนการทำงานของอุปกรณ์

อุปกรณ์นี้จะมีลำดับการทำงานคือ รับภาพโดยใช้กล้องเว็บแคม โดยจะเปลี่ยนมุมการถ่ายภาพโดยใช้ Stepping motor ในการหมุนอุปกรณ์ที่ติดกล้องไว้ โดยใช้การควบคุมผ่านโปรแกรมที่เขียนบน MATLAB ควบคุมผ่าน USB port ไปยังอุปกรณ์บอร์ด Arduino และป้อนลำดับการกระตุ้นไปยังมอเตอร์ตามลำดับ



รูปที่ 3.1 ลำดับการทำงานของโปรแกรมบันทึกภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 อุปกรณ์รับภาพ

อุปกรณ์ที่ใช้คือกล้องเว็บแคมยี่ห้อ Signo รุ่น 207Y ความละเอียด 16 ล้านพิเซล โดยตัวกล้องสามารถรับช่วงคลื่นอินฟราเรดได้ และมีแหล่งกำเนิดแสงอินฟราเรดในตัวด้วย

รูปที่ 3.2 อุปกรณ์การรับภาพ

3.3 อุปกรณ์ควบคุมการหมุน

ในส่วนอุปกรณ์ควบคุมการหมุนนั้น ใช้สเตปมอเตอร์ในการควบคุม เนื่องจากสเตปมอเตอร์สามารถองศาการหมุนได้อย่างแม่นยำมากกว่ามอเตอร์ไฟฟ้ากระแสตรง และเซอร์โวมอเตอร์ โดยใช้บอร์ดอาดูยโน่เป็นตัวประมวลคำสั่งจากคอมพิวเตอร์และป้อนสัญญาณเข้าไปยังสเตปมอเตอร์ โดยโปรแกรมควบคุมบอร์ดอาดูยโน่นั้นเขียนขึ้นโดยโปรแกรม MATLAB และทำการเชื่อมต่อผ่าน USB port ไปยังบอร์ด

3.3.1 รูปแบบการป้อนพัลส์ให้กับมอเตอร์

ในการทดลองนี้จะใช้วิธีการป้อนแบบ ฮาล์ฟสเตป คือ ป้อนทีละครึ่งลำดับให้กับมอเตอร์ ดังตาราง

Step No.	Phase 1	Phase 2	Phase 3	Phase 4
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1
9	1	0	0	0
10	1	1	0	0

ตารางที่ 3.1 ตารางแสดงการป้อนแรงดันให้กลับสเตปมอเตอร์

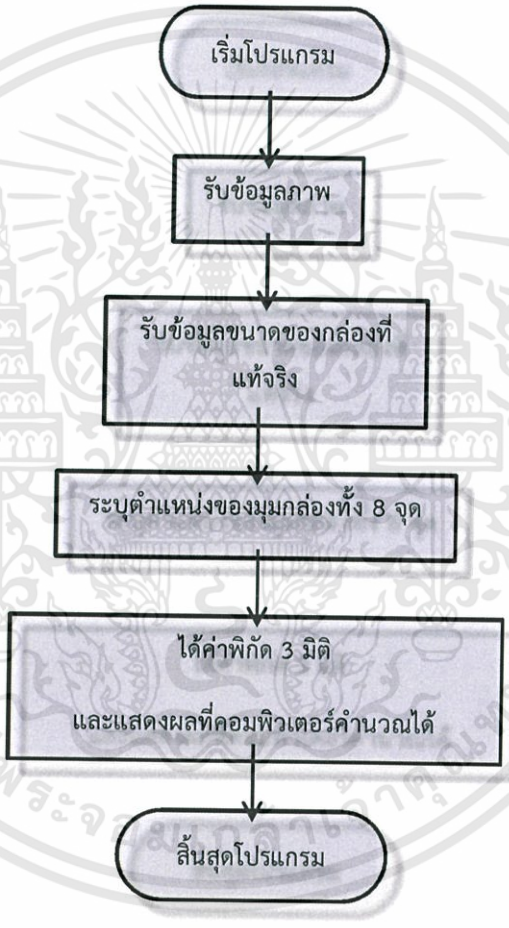
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจะทำให้มอเตอร์มีแรงบิดสูงกว่าการขับดันแบบอื่นๆ แต่ข้อเสียคือใช้กำลังไฟมากขึ้น

3.4 การประมวลภาพ

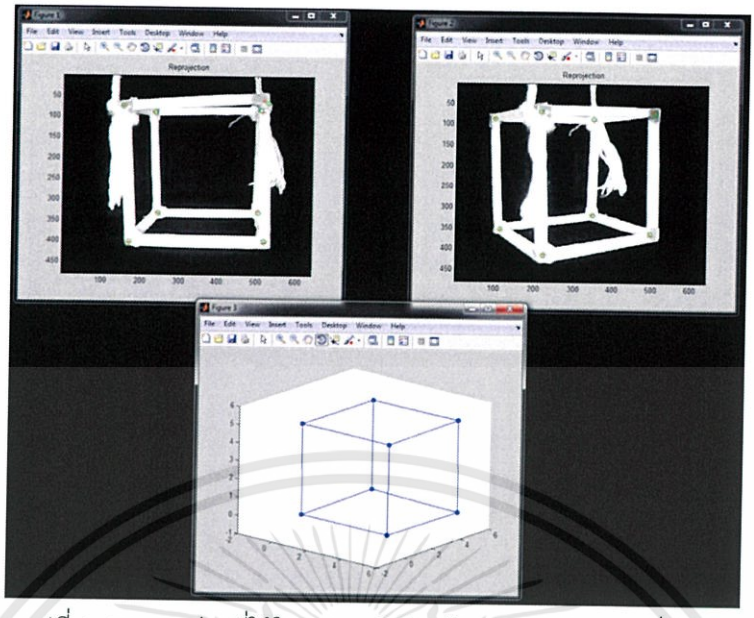
3.4.1 การปรับเทียบพิกัดของรูปภาพ (Camera Calibration)

การหาพิกัดของรูปภาพ เราจะใช้การหามุมของกล่อง โดยจะป้อนค่าพิกัดที่แท้จริงของกล่องไว้กับโปรแกรม และทำการระบุตำแหน่งของกล่องด้วยมืออีกครั้งจากรูปถ่ายที่ได้มาการทดลองจะเริ่มจากการหาพิกัด3 มิติของกล้อง โดยการใช้กล่องที่เราทราบขนาด และทำการ mark จุด ของกล้องในโปรแกรม เพื่อให้คอมพิวเตอร์สามารถคำนวณตำแหน่งของพิกัดที่แท้จริงของในรูปได้ โดยมีบล็อกไดอะแกรมการทำงานดังต่อไปนี้



รูปที่ 3.3 ลำดับการทำงานของโปรแกรมปรับเทียบกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



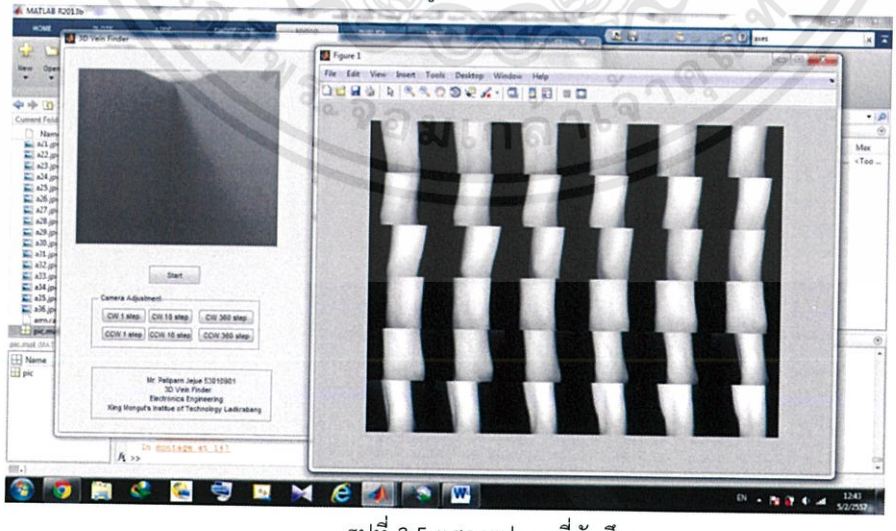
รูปที่ 3.4 ภาพกล่องที่ใช้ในการหาพิกัดกล้อง และพิกัดกล้องที่หาได้

โดยนอกจากรู้พิกัดเชิงลึก หรือสามมิติภายในรูปแล้ว ยังสามารถหาโปรเจคชั่นดาต้า (Projection Data) หรือข้อมูลภาพฉายของกล้องได้อีกด้วย

แล้วเริ่มทำการบันทึกรูปภาพวัตถุ โดยใช้โปรแกรม MATLAB ที่เขียนขึ้นเอง โดยใช้บอร์ดอาดูยโน่ (Arduino Board) ในการรับคำสั่งจากคอมพิวเตอร์และแปลงเป็นลอจิก 0 1 เพื่อป้อนแรงดันให้กับตัวสเตปมอเตอร์เพื่อควบคุมการหมุน

3.4.2 การบันทึกภาพ (Image Acquisition)

การบันทึกภาพ จะทำการบันทึกโดยใช้โปรแกรม MATLAB ซึ่งจะกำหนดระยะเวลาการบันทึกแต่ละเฟรมไว้เป็นจำนวนทั้งหมด 36 รูป

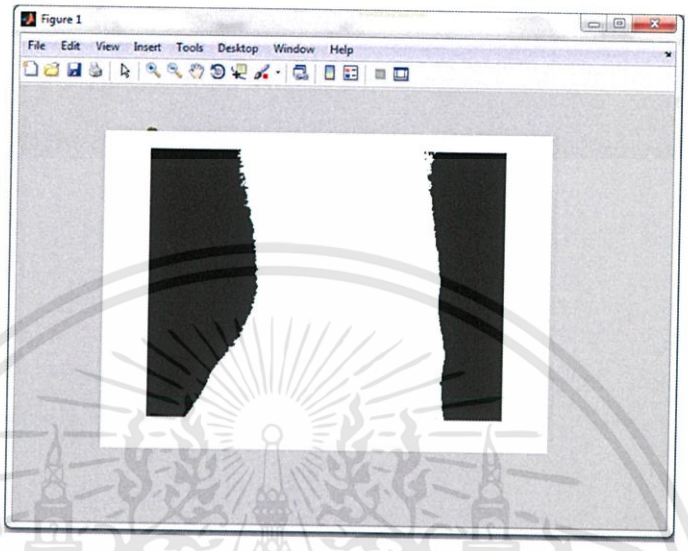


รูปที่ 3.5 แสดงรูปภาพที่บันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 แปลงภาพเป็นไบนารี

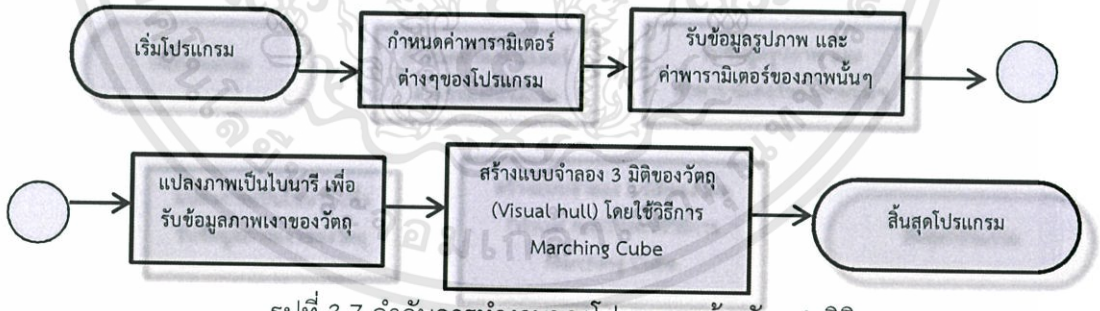
เป็นการแปลงรูปให้อยู่รูป 1 บิต คือ ขาว และ ดำ โดยจะช่วยทำให้กำหนดขอบเขตการสร้างโมเดลของรูปได้ง่ายยิ่งขึ้น



รูปที่ 3.6 ภาพไบนารี

3.4.4 การสร้างแบบจำลอง 3 มิติ (3D Modeling)

คือกระบวนการสุดท้ายของโปรแกรม โดยสร้างโมเดลวัตถุ 3 มิติ โดยใช้ภาพเงาของวัตถุที่เราได้ทำการบันทึกไว้ และใช้ค่าพารามิเตอร์ของภาพฉายนั้นมาคำนวณในการสร้างแบบจำลอง 3 มิติ



รูปที่ 3.7 ลำดับการทำงานของโปรแกรมสร้างวัตถุ 3 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

4.1 การทดลองส่วนการควบคุมสเตปปิ้งมอเตอร์

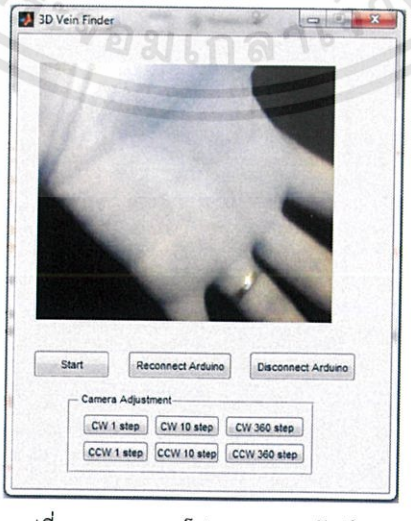
การควบคุมการหมุนของสเตปปิ้งมอเตอร์ โดยใช้บอร์ดอาดูยโน่(Arduino Board) ในการควบคุมการป้อน แรงดันให้กับปิ้งมอเตอร์ ผ่านโปรแกรมแมทแลป (MATLAB)

มุมที่ต้องการ	มุมที่มอเตอร์หมุน	มุมที่ต้องการ	มุมที่มอเตอร์หมุน
10	10.2	190	190
20	20.1	200	200.2
30	30.8	210	211
40	40.3	220	220.1
50	49.9	230	230.2
60	60.1	240	240.5
70	70.3	250	250.4
80	80.2	260	260.1
90	90.5	270	270.7
100	101.2	280	280.3
110	110.7	290	290.5
120	120.4	300	300.1
130	130.2	310	310.6
140	140.1	320	320.4
150	150.3	330	330.8
160	160.5	340	340.1
170	170.1	350	350
180	179.8	360	361

ตารางที่ 4.1 องศาการหมุนของสเตปมอเตอร์

4.2 โปรแกรมส่วนการบันทึกภาพ

เขียนคำสั่งการบันทึกโดยใช้โปรแกรม MATLAB โดยใช้กล้องเว็บแคมในการบันทึกข้อมูล โดยมีปุ่มการทำงานดังนี้



รูปที่ 4.1 GUI ของโปรแกรมการบันทึกภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>การทำงานของปุ่มบนอุปกรณ์</u>

1.) ปุ่มเริ่มการทำงานของโปรแกรม

2.) ปุ่มเชื่อมต่อกับบอร์ดอาดูยโน่ ใช้ในการเชื่อมต่อกับบอร์ดอีกครั้งในกรณีที่ทำการตัดการเชื่อมต่อไปแล้ว (ในกรณีที่โปรแกรมมีปัญหา หรือค้าง)

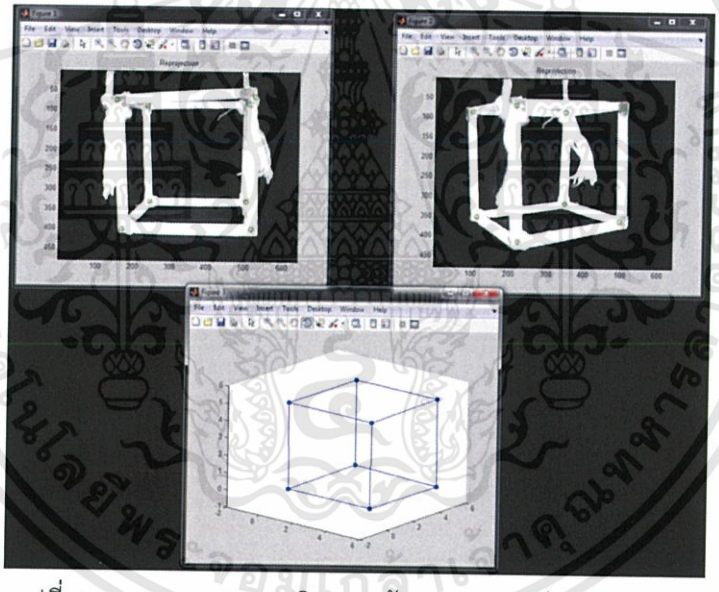
3.) โปรแกรมตัดการเชื่อมต่อกับบอร์ดอาดูยโน่ ใช้ในกรณีที่กำลังจะปิดโปรแกรม เพื่อเป็นการเคลียร์การเชื่อมต่อระหว่างคอมพิวเตอร์และบอร์ด เพื่อป้องกันการไม่สามารถเชื่อมต่อกับบอร์ดได้เนื่องจากพอร์ตไม่ว่าง ในกรณีที่ต้องการเปิดโปรแกรมขึ้นมาใหม่

4.) กลุ่มคำสั่งปรับการหมุนของกล้อง ใช้ในกรณีที่อุปกรณ์หมุนไปติดอยู่ในตำแหน่งที่ไม่ถูกต้อง เพื่อปรับให้มุมกล้องกลับมาอยู่ตำแหน่งที่ต้องการ

4.3 โปรแกรมคาลิเบรตกล้อง

ขั้นตอนการทำงาน

1. โหลดภาพที่ต้องการเก็บพิกัดจุดคาลิเบรต โดยต้องร์พิกัดที่แท้จริง
2. มาร์คมุมกล่องทั้ง 8 จุดเพื่อหาพารามิเตอร์ของกล้อง

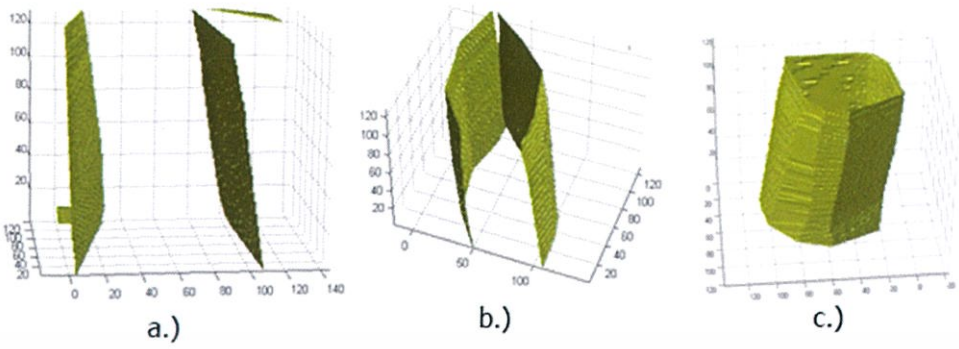


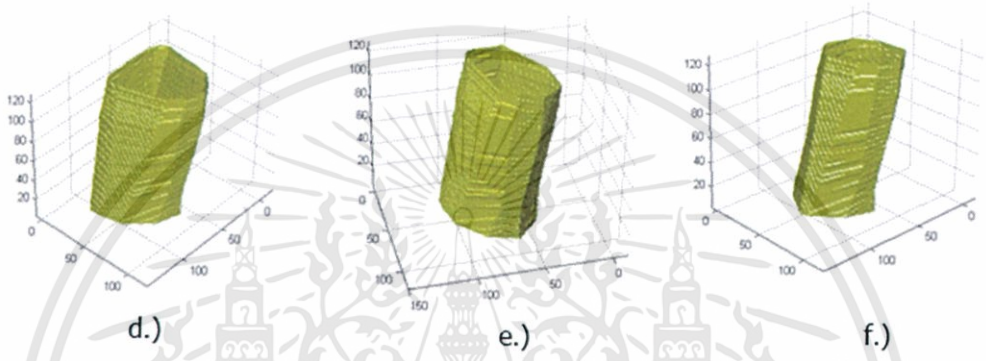
รูปที่ 4.2 ภาพแสดงการคาลิเบรตกล้องและผลลัพธ์ของโปรแกรม

4.4 ส่วนจำลองภาพ 3 มิติ

ในส่วนนี้จำเป็นต้องใช้การแปลงภาพสีเป็นภาพไบนารี เพื่อหาขอบเขตของวัตถุที่เราต้องการจะสร้างเป็นวัตถุ 3 มิติและทำการรับค่าโปรเจคชั่นของกล้องเพื่อทำการสร้างวัตถุเป็น 3 มิติ

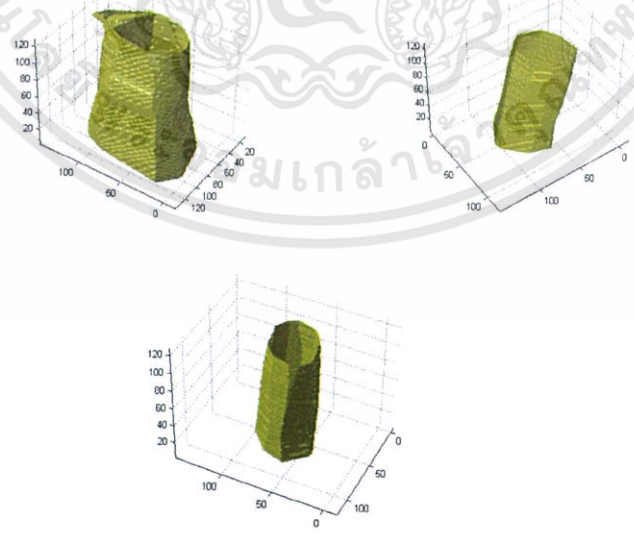
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 4.3 ภาพแสดงการสร้างโมเดล 3 มิติ โดยใช้จำนวนมุมต่างๆกัน
a.) 1 มุม b.) 6 มุม c.) 12 มุม d.) 18 มุม e.) 24 มุม f.) 36 มุม ตามลำดับ

โดยทดลองถ่ายแขนผู้เข้าร่วมทดสอบจำนวน 3 คนเพื่อนำมาสร้างเป็นโมเดล 3 มิติ โดยใช้จำนวนมุมทั้งหมด 36 มุม ทำให้ได้ผลลัพธ์ ดังรูปต่อไปนี้



รูปที่ 4.4 ภาพแสดงผลลัพธ์ภาพถ่ายแขนของผู้เข้าร่วมการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
วิเคราะห์และวิจารณ์ผลการทดลอง

5.1 สรุปผลการทดลอง

จากการทดลองทำให้พบว่า

1. สามารถสร้างรูปจำลอง 3 มิติ ของแขนผู้ป่วยได้ใกล้เคียงกับรูปทรงจริง
2. โปรแกรมไม่สามารถแสดงผลได้ทันที เหมาะสมกับการใช้งานประเภทเก็บข้อมูลผู้ป่วยมากกว่าจะใช้งานแบบเรียลไทม์

5.2 ปัญหาในการดำเนินงาน

1. .ตัวโปรแกรมมีการกินทรัพยากรเครื่องคอมพิวเตอร์มาก จึงไม่สามารถสร้างโมเดลที่มีความละเอียดสูงได้
2. การบันทึกรูปแขนผู้ป่วยสามารถกระทำการได้ยาก เนื่องจากอุปกรณ์อยู่ในแนวขนานกับพื้นโลก จึงทำให้ผู้ป่วยต้องตั้งแขนในมุมตั้งฉากกับพื้นโลก จึงมีโอกาสที่จะเกิดการเคลื่อนไหวของตำแหน่งแขน ทำให้ภาพที่บันทึกมาไม่สมบูรณ์แบบได้ แต่เนื่องจากการทำอุปกรณ์ที่มีลักษณะการหมุนตั้งฉากกับพื้นโลกนั้น จำเป็นต้องใช้มอเตอร์ที่กำลังบิดที่สูง เนื่องจากต้องรับน้ำหนักตัวกล้อง ฉาก และฐานตั้งกล้องของอุปกรณ์ และยังต้องใช้วัสดุที่มีความแข็งแรงสูงในการทำฐานตั้งกล้อง เพื่อป้องกันไม่ให้เกิดการแตกหักเนื่องจากน้ำหนักของตัวอุปกรณ์เอง
3. รูปร่างอุปกรณ์ยังไม่เหมาะสมกับผู้ป่วยที่เป็นเด็ก เนื่องจากไม่สามารถควบคุมให้เด็กอยู่นิ่งๆตลอดการบันทึกรูปได้

5.3 แนวการพัฒนาต่อ

สามารถปรับปรุงเทคนิคการแสดงผลภาพให้ออกมาเป็นโมเดลที่มีพื้นผิวจริงจากภาพถ่ายได้ เพื่อเพิ่มประสิทธิภาพในการตรวจสอบและบันทึกข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1.] http://www.openu.ac.il/home/hassner/projects/poses/

[2.] http://www.ycshao.com/?p=73

[3.] http://en.wikipedia.org/wiki/Visual_hull

[4.] http://cohengroup.lassp.cornell.edu/facilities.php?device=11

[5.] http://www.mathworks.com/hardware-support/arduino-matlab.html

[6.] http://www.arduitronics.com/article/arduino-and-motor-control-part-3-stepper-motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

Code โปรแกรมถ่ายภาพจาก Webcamera

```
function varargout = GUI(varargin)
% GUI MATLAB code for GUI.fig
%      GUI, by itself, creates a new GUI or raises the existing
%      singleton*.
%
%      H = GUI returns the handle to a new GUI or the handle to
%      the existing singleton*.
%
%      GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in GUI.M with the given input arguments.
%
%      GUI('Property','Value',...) creates a new GUI or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before GUI_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to GUI_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI

% Last Modified by GUIDE v2.5 05-Feb-2014 12:28:23

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GUI is made visible.
function GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI (see VARARGIN)

% Choose default command line output for GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
varargout{1} = handles.output;

imaqreset;
clear all;

global a;

a = arduino('com39');

a.pinMode(8,'OUTPUT');
a.pinMode(9,'OUTPUT');
a.pinMode(10,'OUTPUT');
a.pinMode(11,'OUTPUT');

%ส่วนคุมกล้อง
global src;
global vid;

vid = videoinput('winvideo', 2, 'RGB24_640x480');
src = getselectedsource(vid);
vid.TriggerRepeat = Inf;
vid.FramesPerTrigger = 1;
vid.ReturnedColorspace = 'rgb';
triggerconfig(vid, 'manual');

subplot(111);
h1 = image;
axis ij
preview(vid,h1);
start(vid);

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global vid;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
global a;
global pic;

clear pic;

pause(2);
trigger(vid);
pause(0.5);

for i=1:37;

for j=1:14;
    a.digitalWrite(8,1);
    a.digitalWrite(9,0);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);

    a.digitalWrite(8,1);
    a.digitalWrite(9,1);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);

    a.digitalWrite(8,0);
    a.digitalWrite(9,1);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);

    a.digitalWrite(8,0);
    a.digitalWrite(9,1);
    a.digitalWrite(10,1);
    a.digitalWrite(11,1);

    a.digitalWrite(8,0);
    a.digitalWrite(9,0);
    a.digitalWrite(10,1);
    a.digitalWrite(11,0);

    a.digitalWrite(8,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    a.digitalWrite(9,0);
    a.digitalWrite(10,1);
    a.digitalWrite(11,1);

    a.digitalWrite(8,0);
    a.digitalWrite(9,0);
    a.digitalWrite(10,0);
    a.digitalWrite(11,1);
end

%pause(0.1);
trigger(vid);
%pause(0.2);

end

pic = getdata(vid,36);
save('pic.mat', 'pic');
clear pic;

for k=1:37;
   for l=1:14;
   a.digitalWrite(8,0);
   a.digitalWrite(9,0);
   a.digitalWrite(10,0);
   a.digitalWrite(11,1);

   a.digitalWrite(8,0);
   a.digitalWrite(9,0);
   a.digitalWrite(10,1);
   a.digitalWrite(11,1);

   a.digitalWrite(8,0);
   a.digitalWrite(9,0);
   a.digitalWrite(10,1);
   a.digitalWrite(11,0);

   a.digitalWrite(8,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    a.digitalWrite(9,1);
    a.digitalWrite(10,1);
    a.digitalWrite(11,0);

    a.digitalWrite(8,0);
    a.digitalWrite(9,1);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);

    a.digitalWrite(8,1);
    a.digitalWrite(9,1);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);

    a.digitalWrite(8,1);
    a.digitalWrite(9,0);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);
    end
end
clear pic;


% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

imaqreset;
clear all;
delete(instrfind({'Port'},{'COM39'}))


% --- Executes on button press in pushbutton4.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
% handles    structure with handles and user data (see GUIDATA)

global a;

a = arduino('com39');


% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


global a;

 a.digitalWrite(8,1);
    a.digitalWrite(9,0);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);

    a.digitalWrite(8,1);
    a.digitalWrite(9,1);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);

    a.digitalWrite(8,0);
    a.digitalWrite(9,1);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);

    a.digitalWrite(8,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    a.digitalWrite(9,1);
    a.digitalWrite(10,1);
    a.digitalWrite(11,1);

    a.digitalWrite(8,0);
    a.digitalWrite(9,0);
    a.digitalWrite(10,1);
    a.digitalWrite(11,0);

    a.digitalWrite(8,0);
    a.digitalWrite(9,0);
    a.digitalWrite(10,1);
    a.digitalWrite(11,1);

    a.digitalWrite(8,0);
    a.digitalWrite(9,0);
    a.digitalWrite(10,0);
    a.digitalWrite(11,1);


% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


global a;

   a.digitalWrite(8,0);
   a.digitalWrite(9,0);
   a.digitalWrite(10,0);
   a.digitalWrite(11,1);

   a.digitalWrite(8,0);
   a.digitalWrite(9,0);
   a.digitalWrite(10,1);
   a.digitalWrite(11,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    a.digitalWrite(8,0);
    a.digitalWrite(9,0);
    a.digitalWrite(10,1);
    a.digitalWrite(11,0);

    a.digitalWrite(8,0);
    a.digitalWrite(9,1);
    a.digitalWrite(10,1);
    a.digitalWrite(11,0);

    a.digitalWrite(8,0);
    a.digitalWrite(9,1);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);

    a.digitalWrite(8,1);
    a.digitalWrite(9,1);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);

    a.digitalWrite(8,1);
    a.digitalWrite(9,0);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);


% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


global a;

for j=1:14;
   a.digitalWrite(8,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    a.digitalWrite(9,0);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);

    a.digitalWrite(8,1);
    a.digitalWrite(9,1);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);

    a.digitalWrite(8,0);
    a.digitalWrite(9,1);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);

    a.digitalWrite(8,0);
    a.digitalWrite(9,1);
    a.digitalWrite(10,1);
    a.digitalWrite(11,1);

    a.digitalWrite(8,0);
    a.digitalWrite(9,0);
    a.digitalWrite(10,1);
    a.digitalWrite(11,0);

    a.digitalWrite(8,0);
    a.digitalWrite(9,0);
    a.digitalWrite(10,1);
    a.digitalWrite(11,1);

    a.digitalWrite(8,0);
    a.digitalWrite(9,0);
    a.digitalWrite(10,0);
    a.digitalWrite(11,1);
end


% --- Executes on button press in pushbutton11.
function pushbutton11_Callback(hObject, eventdata, handles)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
% hObject    handle to pushbutton11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


global a;

for l=1:14;
    a.digitalWrite(8,0);
    a.digitalWrite(9,0);
    a.digitalWrite(10,0);
    a.digitalWrite(11,1);

    a.digitalWrite(8,0);
    a.digitalWrite(9,0);
    a.digitalWrite(10,1);
    a.digitalWrite(11,1);

    a.digitalWrite(8,0);
    a.digitalWrite(9,0);
    a.digitalWrite(10,1);
    a.digitalWrite(11,0);

    a.digitalWrite(8,0);
    a.digitalWrite(9,1);
    a.digitalWrite(10,1);
    a.digitalWrite(11,0);

    a.digitalWrite(8,0);
    a.digitalWrite(9,1);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);

    a.digitalWrite(8,1);
    a.digitalWrite(9,1);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    a.digitalWrite(8,1);
    a.digitalWrite(9,0);
    a.digitalWrite(10,0);
    a.digitalWritc(11,0);
end

% --- Executes on button press in pushbutton12.
function pushbutton12_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global a;

for i=1:37;

   for j=1:14;
   a.digitalWrite(8,1);
   a.digitalWrite(9,0);
   a.digitalWrite(10,0);
   a.digitalWrite(11,0);

   a.digitalWrite(8,1);
   a.digitalWrite(9,1);
   a.digitalWrite(10,0);
   a.digitalWrite(11,0);

   a.digitalWrite(8,0);
   a.digitalWrite(9,1);
   a.digitalWrite(10,0);
   a.digitalWrite(11,0);

   a.digitalWrite(8,0);
   a.digitalWrite(9,1);
   a.digitalWrite(10,1);
   a.digitalWrite(11,1);

   a.digitalWrite(8,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    a.digitalWrite(9,0);
    a.digitalWrite(10,1);
    a.digitalWrite(11,0);

    a.digitalWrite(8,0);
    a.digitalWrite(9,0);
    a.digitalWrite(10,1);
    a.digitalWrite(11,1);

    a.digitalWrite(8,0);
    a.digitalWrite(9,0);
    a.digitalWrite(10,0);
    a.digitalWrite(11,1);
    end
end
% --- Executes on button press in pushbutton13.
function pushbutton13_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global a ;

for k=1:37;
   for l=1:14;
   a.digitalWrite(8,0);
   a.digitalWrite(9,0);
   a.digitalWrite(10,0);
   a.digitalWrite(11,1);

   a.digitalWrite(8,0);
   a.digitalWrite(9,0);
   a.digitalWrite(10,1);
   a.digitalWrite(11,1);

   a.digitalWrite(8,0);
   a.digitalWrite(9,0);
   a.digitalWrite(10,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    a.digitalWrite(11,0);

    a.digitalWrite(8,0);
    a.digitalWrite(9,1);
    a.digitalWrite(10,1);
    a.digitalWrite(11,0);

    a.digitalWrite(8,0);
    a.digitalWrite(9,1);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);

    a.digitalWrite(8,1);
    a.digitalWrite(9,1);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);

    a.digitalWrite(8,1);
    a.digitalWrite(9,0);
    a.digitalWrite(10,0);
    a.digitalWrite(11,0);
    end
end


% --- Executes on button press in pushbutton14.
function pushbutton14_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Code 3D Reconstruction Visual Hull

clear, close all;

numCamera = 36;
volumeX = 128;
volumeY = 128;
volumeZ = 128;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
dX = volumeX/10;
dY = volumeY/10;
dZ = volumeZ/10;
visualHullSamplingRate = 20;
silhouetteThreshold = 180;

% load silhouette images and projection matrices
for i=1:numCamera

    if(i<10)
        PMatx{i} = load(['projectiondata\b_0' int2str(i) '.PA']);
        pic{i} = adapthisteq(rgb2gray(imread(['OIL\a_' int2str(i) '.jpg'])));
        Ivpic{i} = pic{i};
    else
        PMatx{i} = load(['projectiondata\b_' int2str(i) '.PA']);
        pic{i} = adapthisteq(rgb2gray(imread(['OIL\a_' int2str(i) '.jpg'])));
        Ivpic{i} = pic{i};
    end;
end;

% visual hull computation
thresh = 30;
scale = 1/19;
volume = ones(volumeX,volumeY,volumeZ);

for i = 1:numCamera
    img = Ivpic{i} >= thresh;
    img = bwareaopen(img,500,8);
    P = PMatx{i};

    x1 = P(1,1)*scale*((1:volumeX)-dX);
    y1 = P(2,1)*scale*((1:volumeX)-dX);
    z1 = P(3,1)*scale*((1:volumeX)-dX);
    x2 = P(1,2)*scale*((1:volumeY)-dY);
    y2 = P(2,2)*scale*((1:volumeY)-dY);
    z2 = P(3,2)*scale*((1:volumeY)-dY);

    for Z = 1:volumeZ
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        p = P(:,3:4)*[scale*(Z-dZ) 1]';
        z = z1'*ones(1,volumeY) +  ones(volumeX,1)*z2 + p(3);
        x = (x1'*ones(1,volumeY) +  ones(volumeX,1)*x2 + p(1)) ./ z;
        y = (y1'*ones(1,volumeY) +  ones(volumeX,1)*y2 + p(2)) ./ z;

        volume(:,:,Z) = volume(:,:,Z)& interp2(img, x, y,'nearest', 1);
    end
end

% display result
figure;
set(gca,'projection','perspective');
axis equal;
grid on;
hold on;
[xMesh, yMesh, zMesh] = meshgrid(1:volumeY,1:volumeX,1:volumeZ);
pt = patch(isosurface(xMesh, yMesh, zMesh, volume, 0.9));
isonormals(xMesh, yMesh, zMesh, volume, pt);
set(pt,'FaceColor','yellow','EdgeColor','none');
daspect([1 1 1]);
camlight;
lighting gouraud;

view(-90,90);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้