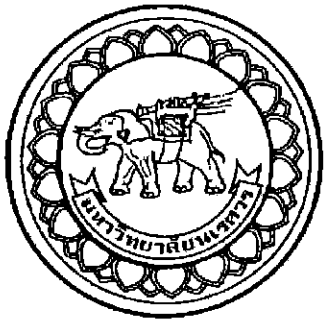
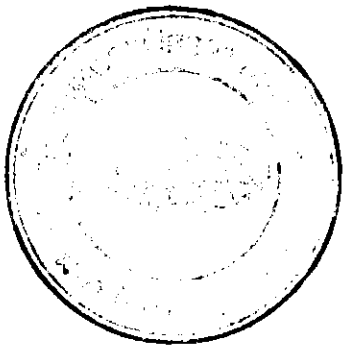




# การตรวจจับการเคลื่อนไหวของมนุษย์โดยใช้การประมวลผลภาพ และจำลองการเคลื่อนไหวด้วยภาพ 3 มิติ

**Human Motion Capture using Image Processing and Simulation in 3D Animation**


นายทรงกรด พิมใจ รหัส 48362278

นายธีรโรจน์ พ้นภัย รหัส 48362315



5 เม.ย. 2553

เลขทะเบียน i 499 3457

เลขเรียกหนังสือ ว/s.

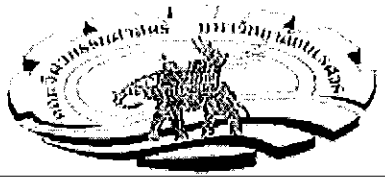
ว 124 ก 2551


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2551



# ใบรับรองโครงงานวิศวกรรม

| | |
|---|---|
| **หัวข้อโครงงาน** | การตรวจจับการเคลื่อนไหวของมนุษย์โดยใช้การประมวลผลภาพและจำลองการเคลื่อนไหวด้วยภาพ 3 มิติ |
| **ผู้ดำเนินโครงงาน** | นายทรงกรด พิมใจ รหัส 48362278 |
| | นายธีรโรจน์ พ้นภัย รหัส 48362315 |
| **อาจารย์ที่ปรึกษา** | นายเศรษฐา ตั้งค้าวานิช |
| **สาขาวิชา** | วิศวกรรมคอมพิวเตอร์ |
| **ภาควิชา** | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ |
| **ปีการศึกษา** | 2551 |

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ
(อาจารย์เศรษฐา ตั้งค้าวานิช)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุรเชษฐ์ กานต์ประชา)

.....กรรมการ
(ดร.อัครพันธ์ วงศ์กังแห)

| | |
|---|---|
| **หัวข้อโครงงาน** | การตรวจจับการเคลื่อนไหวของมนุษย์โดยใช้การประมวลผลภาพและจำลองการเคลื่อนไหวด้วยภาพ 3 มิติ |
| **ผู้ดำเนินโครงงาน** | นายทรงกรด พิมใจ รหัส 48362278 |
| | นายธีรโรจน์ พ้นภัย รหัส 48362315 |
| **อาจารย์ที่ปรึกษา** | อาจารย์เศรษฐา ตั้งค้าวานิช |
| **สาขาวิชา** | วิศวกรรมคอมพิวเตอร์ |
| **ภาควิชา** | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ |
| **ปีการศึกษา** | 2551 |

## บทคัดย่อ


เทคโนโลยีด้านภาพเคลื่อนไหว 3 มิติ ได้เข้ามามีบทบาทอย่างมากในปัจจุบัน ไม่ว่าจะเป็นวงการภาพยนตร์ การ์ตูนแอนนิเมชัน เกม 3 มิติ ฯลฯ เนื่องจากภาพ 3 มิติสามารถจำลองรูปร่างและการเคลื่อนไหวของสิ่งต่างๆ ได้อย่างสมจริง การกำหนดการเคลื่อนไหวให้กับโมเดล 3 มิติแบบเดิมจะต้องทำทีละเฟรม ซึ่งซับซ้อนมาก และใช้เวลานาน จึงต้องใช้เทคโนโลยีเพื่อเพิ่มความสะดวกในการกำหนดการเคลื่อนไหวให้กับโมเดล 3 มิติ วิธีที่ได้รับความนิยมมากในปัจจุบันวิธีหนึ่ง คือ การทำ โมชัน แคปเจอร์ (Motion Capture) ซึ่งเป็นการจับการเคลื่อนไหวจากวัตถุจริง ไม่ว่าจะเป็นมนุษย์ สัตว์ หรือสิ่งของ แล้วแปลงการเคลื่อนที่นั้นให้อยู่ในรูปของ3 มิติ

โครงงานนี้เป็นการพัฒนาโปรแกรมจับการเคลื่อนไหวของมนุษย์ โดยระบบจะใช้กล้องวิดีโอจำนวน 4 ตัว เพื่อ บันทึกภาพผู้แสดง โดยผู้แสดงจะสวมชุดซึ่งติดมาร์กเกอร์ไว้ ณ ส่วนต่างๆ ของร่างกายตามที่กำหนด จากนั้นจะใช้กล้องจับภาพการเคลื่อนไหว แล้วส่งภาพไปยังโปรแกรมประมวลผลภาพ เพื่อหาพิกัดและติดตามการเคลื่อนที่ของมาร์กเกอร์ แล้วนำพิกัดของมาร์กเกอร์ที่ได้จากแต่ละกล้อง มาคำนวณตำแหน่งใน 3 มิติ จากนั้นนำตำแหน่งใน 3 มิติ ที่ได้ มาแสดงผลเป็นภาพ 3 มิติ โดย OpenGL

| | |
|---|---|
| **Project Title** | **Human Motion Capture using Image Processing and Simulation in 3D Animation** |
| **Name** | Mr. Songkrod Pimjai ID. 48362278 |
| | Mr. Teerarot Ponpai ID. 48362315 |
| **Project Advisor** | Mr. Settha Tungkawanich |
| **Major** | Computer Engineering. |
| **Department** | Electrical and Computer Engineering. |
| **Academic Year** | 2008 |


## ABSTRACT

Nowadays, 3D animation technology has central role in film industry, animation movie, 3D games etc. 3D computer graphics can simulate shapes and motion of anything or create images that seem real to the viewer. Formerly, to animate 3D has to do frame by frame that cause loses much more times. For timesaving, the technology to give motion for animation automaticly is used and the most popular method used today is motion capture.

This project develops a motion capture system using video cameras and computer vision for process images. The system uses 4 video cameras to capture human motion. The human who does acting has to ware a specific markers suit. The cameras capture actor's movement and send digital images to the application. The application processes images to find 2D positions of markers, tracking markers movement in each view. Finally the application calculates these 2D positions data to 3D positions of each marker and display in 3D animation using OpenGL

# กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลืออย่างดียิ่งจาก อ.เศรษฐา ตั้งค้าวานิช อาจารย์ที่ปรึกษาโครงงาน ที่ได้ให้คำแนะนำและข้อคิดเห็นต่างๆของโครงงานมาโดยตลอด

ผู้ช่วยศาสตราจารย์ ดร.สุรเชษฐ์ กานต์ประชา และ ดร.อัครพันธ์ วงศ์กังแห ที่กรุณาสละเวลาเป็นอาจารย์กรรมการสอบโครงงาน พร้อมทั้งให้คำแนะนำที่เป็นประโยชน์

อาจารย์ทุกท่าน บิดา มารดา ญาติพี่น้อง และเพื่อนๆ ที่คอยให้คำปรึกษา แนะนำ และติชม ผู้ดำเนินโครงงาน

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ที่ได้อนุมัติเงินค่าใช้จ่ายในการทำโครงงานนี้

ผู้จัดทำโครงงานขอกราบขอบพระคุณเป็นอย่างสูง มา ณ โอกาสนี้

นายทรงกรด พิมใจ

นายธีรโรจน์ พันภัย

# สารบัญ


หน้า

บทคัดย่อภาษาไทย ..... ก

บทยัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ช

สารบัญรูป ..... ซ

บทที่ 1 บทนำ ..... 1

1.1 ที่มาและความสำคัญของโครงงาน ..... 1

1.2 วัตถุประสงค์ของโครงงาน ..... 1

1.3 ขอบข่ายของโครงงาน ..... 2

1.4 ขั้นตอนของการดำเนินงาน ..... 2

1.5 แผนผังการดำเนินโครงงาน ..... 3

1.6 ผลที่คาดว่าจะได้รับ ..... 4

1.7 งบประมาณของโครงงาน ..... 4

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง ..... 5

2.1 การตรวจจับการเคลื่อนไหว (Motion Capture) ..... 5

2.2 Digital Image Processing ..... 7

2.3 ระบบสเตอริโอวิชั่น (Stereo Vision System) ..... 15

2.4 ภาพเคลื่อนไหว 3 มิติ (3D Animation) ..... 18

# สารบัญ (ต่อ)


หน้า

บทที่ 3 ขั้นตอนการดำเนินงาน....21

3.1 ระบบโดยรวม....21

3.2 การออกแบบโปรแกรม....22

3.3 การออกแบบระบบจับภาพ....24

3.4 การประมวลผลภาพ....26

3.5 การหาพิกัดจุด....29

3.6 การกำหนดและติดตามมาร์กเกอร์....31

3.7 แสดงการเคลื่อนไหวให้เป็น 3 มิติ....31

บทที่ 4 ผลการทดลอง....32

4.1 การจับภาพเคลื่อนไหว....32

4.2 ผลการประมวลผลภาพ....35

4.3 ผลจากการแปลงพิกัดค่าพิกัด 2 มิติ เป็น 3 มิติ....38

บทที่ 5 บทสรุป....41

5.1 หน้าที่การทำงานของโปรแกรม....41

5.2 วิเคราะห์ผลการทดลอง....41

5.3 ปัญหาและแนวทางการแก้ไข....41

5.4 แนวทางการพัฒนาต่อ....42

# สารบัญ (ต่อ)


หน้า

เอกสารอ้างอิง ..... 43

ภาคผนวก ..... 44

ประวัติผู้เขียนโครงงาน ..... 53

# สารบัญตาราง


ตารางที่ หน้า

1.1 แผนผังการดำเนินโครงงาน ..... 3

4.1 แสดงการ Blob ที่ละเฟรม ..... 35


มหาวิทยาลัยนเรศวร

# สารบัญรูป


รูปที่ หน้า

2.1 ตัวอย่างการสวมชุดตรวจจับโดยใช้แม่เหล็ก และเครื่องส่งคลื่นสนามแม่เหล็ก ..... 6

2.2 ตัวอย่างการจับการเคลื่อนไหวโดยใช้กล้อง ..... 7

2.3 ตัวอย่างการจับการเคลื่อนไหวโดยใช้ชุดเชิงกล ..... 7

2.4 ตัวอย่างสีรูปแบบ RGB ..... 10

2.5 ตัวอย่างสีรูปแบบ HSV ..... 10

2.6 ตัวอย่างการแยกองค์ประกอบภาพ ..... 11

2.7 ตัวอย่างการทำ threshold เพื่อตัดสีฟ้าออก ..... 12

2.8 ภาพตัวอย่างการทำ Median Filter ..... 14

2.9 ภาพก่อนและหลังทำ Blob Analysis ..... 15

2.10 ตัวอย่างภาพของระบบรูปที่210สเตอริโอแบบถ่ายจากกล้อง 2 ตัว ..... 16

2.11 หลักการคำนวณพิกัดของระบบสเตอริโอ ..... 16

2.12 ลักษณะของเรขาคณิตอีพิรูปที่212โพลาร์ ..... 16

2.13 รังสีจากจุดของภาพที่ตรงกันไม่ตัดกันใน 3 มิติ ..... 17

2.14 เวคเตอร์ที่ใช้ในการหาค่าพิกัด ..... 17

2.15 ตัวอย่างภาพยนตร์การ์ตูน 3 มิติ เรื่องก้านกล้วย ..... 18

2.16 ตัวอย่างการนำเข้าไฟล์ BVH มาใช้งานในโปรแกรม Blender ..... 19

2.17 ตัวอย่างข้อมูลภายในไฟล์ BVH ส่วน Model hierarchy ..... 20

3.1 แสดงการทำงานโดยรวมของทั้งระบบ ..... 21

3.2 ผังการทำงานของระบบ ..... 23

3.3 ตัวอย่างกล้องวงจรปิด และ การ์ดจับภาพ ..... 24

3.4 การติดมาร์กเกอร์ตามจุดหมุนหรือข้อต่อกระดูกในร่างกาย ..... 24

3.5 แสดงการวางตำแหน่งของกล้อง ..... 25

3.6 แสดงส่วนสูงและมุมกล้อง ..... 26

# สารบัญรูป (ต่อ)


| รูปที่ | หน้า |
|---|---|
| 3.7 ขั้นตอนการทำงานส่วนประมวลผลภาพ | 26 |
| 3.8 ความสัมพันธ์ไรบรารี่ของ OpenCV | 27 |
| 3.9 ตัวอย่างการทำ Blob analysis | 28 |
| 3.10 ตัวอย่างการทำ Blob analysis | 28 |
| 4.1 แสดงด้านหน้าการติดมาร์กเกอร์ตามจุดหมุนหรือข้อต่อกระดูกในร่างกาย | 33 |
| 4.2 แสดงด้านหลังการติดมาร์กเกอร์ตามจุดหมุนหรือข้อต่อกระดูกในร่างกาย | 33 |
| 4.3 แสดงด้านข้างการติดมาร์กเกอร์ตามจุดหมุนหรือข้อต่อกระดูกในร่างกาย | 34 |
| 4.4 แสดงด้านหน้าส่วนบนการติดมาร์กเกอร์ตามจุดหมุนหรือข้อต่อกระดูกในร่างกาย | 34 |
| 4.5 แสดงภาพ3 มิติ ในท่ากางแขน และยกขา | 38 |
| 4.6 แสดงภาพ3 มิติ ในท่ากำลังยกแขน | 38 |
| 4.7 แสดงภาพ3 มิติ ในท่ากำลังก้มศรีษะ | 39 |
| 4.8 แสดงภาพ3 มิติ ในท่ากางแขนบิดลำตัว | 39 |
| 4.9 แสดงภาพ3 มิติ ในกรณีที่ระบุมาร์กเกอร์ผิดพลาด ทำให้การแสดงผลไม่ถูกต้อง | 40 |

# บทที่ 1
# บทนำ

## 1.1 ที่มาและความสำคัญของโครงงาน

เทคโนโลยีด้านภาพเคลื่อนไหว 3 มิติ ได้เข้ามามีบทบาทอย่างมากในโลกปัจจุบัน ไม่ว่าจะเป็นวงการภาพยนตร์ การ์ตูนแอนนิเมชัน เกม 3 มิติ ฯลฯ เนื่องจากภาพ 3 มิติสามารถจำลองรูปร่างและการเคลื่อนไหวของสิ่งต่างๆ ได้อย่างสมจริง การกำหนดการเคลื่อนไหวให้กับโมเดล 3 มิติแบบเดิมซึ่งจะต้องทำทีละเฟรมนั้น ค่อนข้างซับซ้อน และใช้เวลานาน จึงต้องใช้เทคโนโลยีเพื่อจับการเคลื่อนไหวจากวัตถุจริงแล้วประมวลผลข้อมูลเหล่านั้นมากำหนดการเคลื่อนไหวให้กับโมเดล 3 มิติ

เทคโนโลยีที่ใช้ในการตรวจจับการเคลื่อนไหวจริง แล้วนำข้อมูลการเคลื่อนไหวมาบันทึกให้อยู่ในรูปแบบ3 มิติ เรียกว่า Motion capture หรือ MoCap ซึ่ง มีอยู่หลายรูปแบบ เช่น การใช้เซ็นเซอร์วัดระยะขจัดเชิงมุม การใช้เซ็นเซอร์แบบแม่เหล็ก การใช้แสงอินฟลาเรด และการใช้กล้องวิดีโอ เป็นต้น

โครงงานนี้ได้มุ่งเน้นไปที่การจับการเคลื่อนไหวโดยใช้กล้องวิดีโอและใช้การประมวลผลภาพ มาวิเคราะห์ลักษณะการเคลื่อนไหว แล้วบันทึกข้อมูลการเคลื่อนไหวนั้นให้เป็นไฟล์ที่สามารถกำหนดการเคลื่อนไหวให้กับตัวละคร 3 มิติ โดยมีจุดมุ่งหมาย คือ ทำให้ตัวละคร 3 มิติ มีการเคลื่อนไหวตามที่กำหนดอย่างเป็นธรรมชาติ เช่น หากผู้แสดงทำการโบกมือ โมเดล 3 มิติก็จะโบกมือเช่นเดียวกัน เป็นต้น

## 1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อเขียนโปรแกรมที่สามารถตรวจจับการเคลื่อนไหวของมนุษย์โดยใช้กล้องวิดีโอและบันทึกข้อมูลการเคลื่อนไหวในรูปแบบของไฟล์ที่ใช้กับโปรแกรมสร้างงาน 3 มิติ ได้
2. เพื่อศึกษาการใช้ กล้องวิดีโอ และการประมวลผลภาพสำหรับตรวจจับการเคลื่อนไหวของมนุษย์
3. เพื่อศึกษาการสร้างภาพเคลื่อนไหว 3 มิติ
4. เพื่อศึกษารูปแบบของไฟล์ที่ใช้ในการบันทึกการเคลื่อนไหว สำหรับงานแอนนิเมชัน 3 มิติ
5. เพื่อศึกษาการกำหนดการเคลื่อนไหวให้กับภาพแอนนิเมชัน 3 มิติ

## 1.3 ขอบข่ายของโครงงาน

1. สามารถตรวจจับการเคลื่อนไหวของคน 1 คนโดยใช้กล้องวิดีโอได้
2. สามารถแปลงการเคลื่อนไหวที่จับได้ มาเป็นไฟล์บันทึกการเคลื่อนไหว ซึ่งสามารถกำหนดการเคลื่อนไหวให้กับตัวละคร 3 มิติ ได้ โดยใช้งานร่วมกับโปรแกรมสร้างงาน 3 มิติมาตรฐาน
3. ไม่สามารถตรวจจับการเคลื่อนไหวเล็กน้อยได้ เช่น การกระดิกนิ้วมือ
4. ผู้แสดงต้องสวมชุดที่ทำขึ้นมาเป็นพิเศษ
5. สภาพแวดล้อมในการแสดงต้องถูกจัดขึ้นอย่างเหมาะสม
6. ใช้กล้องวิดีโอจำนวน 4 ตัว ในมุมมองที่ถูกกำหนดไว้ตายตัว

## 1.4 ขั้นตอนของการดำเนินงาน

1. รวบรวมข้อมูลเกี่ยวกับการประมวลผลภาพ, ทฤษฎีคณิตศาสตร์สำหรับแปลงพิกัด 2 มิติ เป็นพิกัด 3 มิติ การสร้างไฟล์บันทึกการเคลื่อนไหวสำหรับโปรแกรม 3 มิติ และ การสร้างภาพเคลื่อนไหว 3 มิติ
2. ศึกษาทำความเข้าใจในเนื้อหาที่เลือกมาและเลือกเครื่องมือที่ใช้ในการทำงาน
3. ออกแบบและเขียนโปรแกรมในส่วนการประมวลผลภาพ
4. ออกแบบและเขียนโปรแกรมในส่วนการหาพิกัด 3 มิติ
5. ออกแบบและเขียนโปรแกรมในส่วนการบันทึกการเคลื่อนที่ในรูปแบบของไฟล์ที่สามารถใช้กับโปรแกรมสร้างงาน 3 มิติ มาตรฐาน
6. ออกแบบวิธีการทำให้โปรแกรมส่วนต่างๆ ทำงานร่วมกัน
7. ทำการทดลองโปรแกรมที่เขียนเพื่อใช้เป็นข้อมูลในการพัฒนา
8. แก้ไขข้อบกพร่องของโปรแกรม
9. สรุปและจัดทำรายงาน

## 1.5 แผนผังการดำเนินโครงงาน

**ตารางที่ 1.1** แผนผังการดำเนินโครงงาน

| กิจกรรม | ปี 2551 | | | | | | ปี 2552 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | ก.ค. | ส.ค. | ก.ย. | ต.ค. | พ.ย. | ธ.ค. | ม.ค. | ก.พ. | มี.ค. | เม.ย. | พ.ค. |
| 1. รวบรวมข้อมูล การประมวลผลภาพ, การหาพิกัด 3 มิติ จากภาพ 2 มิติ, | | | | | | | | | | | |
| 2. ศึกษาทำความเข้าใจกับเนื้อหาที่เลือกมา | | | | | | | | | | | |
| 3. ออกแบบและเขียนโปรแกรมในส่วนการรับภาพและประมวลผลภาพ | | | | | | | | | | | |
| 4. ออกแบบและเขียนโปรแกรมในส่วนการหาหาพิกัด 3 มิติ จากพิกัด 2 มิติ | | | | | | | | | | | |
| 5. ออกแบบและเขียนโปรแกรมส่วนแสดงผลภาพ 3 มิติ | | | | | | | | | | | |
| 6. ออกแบบวิธีการทำให้โปรแกรมส่วนต่างๆ ทำงานร่วมกัน | | | | | | | | | | | |
| 7. ทดลองใช้งาน และแก้ไขข้อบกพร่องของโปรแกรม | | | | | | | | | | | |
| 8. สรุปและจัดทำเอกสารรายงาน | | | | | | | | | | | |

## 1.6 ผลที่คาดว่าจะได้รับ

1. โปรแกรมติดต่อกล้องวิดีโอสำหรับจับการเคลื่อนไหวของมนุษย์และบันทึกเป็นไฟล์ที่สามารถใช้ร่วมกับโปรแกรมสร้างงาน 3 มิติมาตรฐาน
2. เข้าใจหลักการประมวลผลของภาพและนำไปประยุกต์ใช้งานจริง
3. เข้าใจทฤษฎีการแปลงพิกัดจาก 2 มิติ เป็นพิกัด 3 มิติ โดยใช้การคำนวณ
4. เข้าใจการสร้างการเคลื่อนไหวของภาพแอนนิเมชัน 3 มิติ
5. สามารถเป็นแนวทางพัฒนาเพื่อนำไปใช้งานจริง สำหรับการผลิตงานแอนนิเมชัน 3 มิติได้

## 1.7 งบประมาณของโครงงาน

| | | | |
|---|---|---|---|
| 1. ค่ากล้องวีโอ | เป็นเงิน | 2,800 | บาท |
| 2. การ์ดติดต่อกล้องวิดีโอ | เป็นเงิน | 1,000 | บาท |
| 3. ค่าถ่ายเอกสารพร้อมเข้าเล่ม | เป็นเงิน | 1,000 | บาท |
| รวมเป็นเงินทั้งสิ้น | | 4,800 | บาท (สี่พันแปดร้อยบาทถ้วน) |

หมายเหตุ (ถ้ามีเฉลี่ยทุกรายการ)

# บทที่ 2
# ทฤษฎีพื้นฐาน

การพัฒนาระบบจับการเคลื่อนไหวของมนุษย์ เพื่อกำหนดการเคลื่อนไหวให้ภาพ 3 มิติ นั้น จำเป็นต้องใช้ความรู้ในหลายๆ ส่วนประกอบกันไม่ว่าจะเป็นด้านการประมวลผลภาพม การเขียนโปรแกรมด้วย Visual C++, การคำนวณเรขาคณิตเพื่อหาพิกัดใน 3 มิติ, การเขียนโปรแกรมเพื่อแสดงผลภาพ 3 มิติ ด้วย OpenGL รวมถึงความรู้ในด้านการสร้างงาน 3 มิติ

การพัฒนาระบบจับการเคลื่อนไหวของมนุษย์ เพื่อกำหนดการเคลื่อนไหวให้ภาพ 3 มิติ นั้น ต้องมีความรู้และทราบหลักการที่เกี่ยวข้องเพื่อนำมาใช้ประกอบในการสร้างโปรแกรม ซึ่ง หลักการและทฤษฎีที่สำคัญทั้งหมด ได้อธิบายไว้ดังต่อไปนี้

## 2.1 การตรวจจับการเคลื่อนไหว (Motion Capture) [1]

การตรวจจับการเคลื่อนไหวเป็นกระบวนการในการบันทึกการเคลื่อนไหวจริง และเปลี่ยนให้อยู่ในรูปทางคณิตศาสตร์ที่สามารถนำไปใช้ได้ โดยการติดตามจุดสำคัญ (Key Point) ที่เคลื่อนไหวในบริเวณ แล้วนำข้อมูลเหล่านั้นมารวมกันเพื่อสร้างการแสดงผลในรูปของ 3 มิติ หรือการตรวจจับการเคลื่อนไหว คือเทคโนโลยีที่ช่วยให้สามารถแปลงการแสดงในโลกแห่งความเป็นจริงเป็นการแสดงในรูปของ 3 มิติ นั่นเอง สิ่งที่จะนำมาตรวจจับนั้นจะเป็นอะไรก็ได้ในโลกแห่งความเป็นจริงที่มีการเคลื่อนไหว และจุดสำคัญก็คือบริเวณที่สามารถแสดงการเคลื่อนไหวในส่วนต่างๆ ของสิ่งๆ นั้นได้ดีที่สุด เช่น หากจะตรวจจับการเคลื่อนไหวของมนุษย์ จุดเหล่านั้นอาจเป็นส่วนของข้อต่อหรือจุดเชื่อมต่อของอวัยวะหลัก จุดเหล่านี้จะถูกตรวจจับโดยเซ็นเซอร์ (Sensor) มาร์กเกอร์ (Marker) หรือตัวต้านทานปรับค่าได้ (Potentiometer) ที่ติดไว้กับสิ่งที่จะตรวจจับ และจะเป็นตัวที่ให้ข้อมูลสำหรับการตรวจจับแก่อุปกรณ์เก็บข้อมูลต่อไป

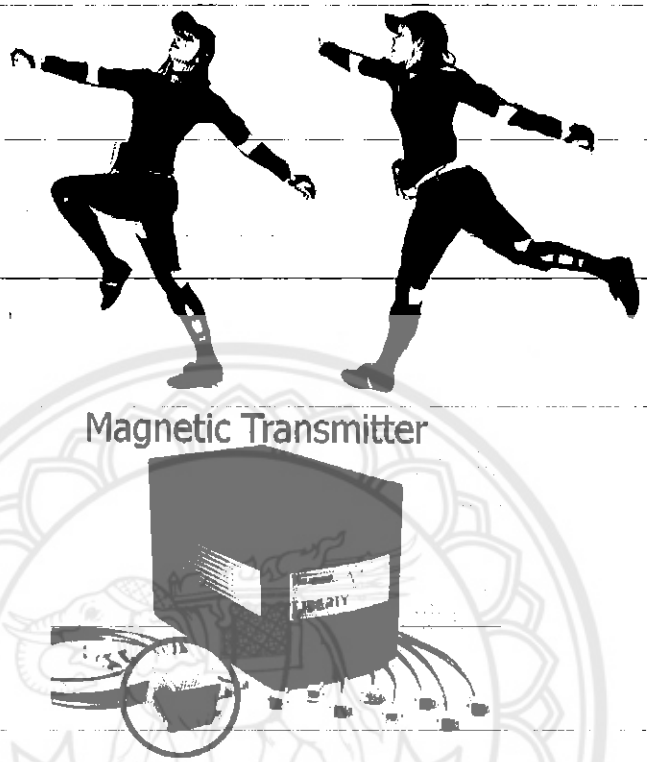
### 2.1.1 ชนิดของการตรวจจับการเคลื่อนไหว

การตรวจจับการเคลื่อนไหวในปัจจุบันมีด้วยกันหลายวิธีสามารถแบ่งตามเทคโนโลยีที่ใช้ได้ 3 ชนิดหลักๆ ได้แก่

1. การตรวจจับการเคลื่อนไหวโดยใช้แม่เหล็ก (Magnetic Motion Capture)

ซึ่งใช้เซ็นเซอร์ติดไว้ตามส่วนต่างๆ เพื่อวัดคลื่นแม่เหล็กไฟฟ้าความถี่ต่ำที่ผลิตออกมาจากแหล่งกำเนิดคลื่นที่ติดตั้งไว้ในบริเวณที่ทำการตรวจจับ เซ็นเซอร์แต่ละตัวจะถูกต่อสายมาเข้ายังวงจรวัดค่าเพื่อหาตำแหน่งของเซ็นเซอร์แต่ละตัวในสนามแม่เหล็ก วงจรวัดค่าจะส่งข้อมูลไปเข้า

เครื่องคอมพิวเตอร์เพื่อแสดงตำแหน่งและการหมุนใน 3 มิติ ตัวอย่างการจับการเคลื่อนไหวด้วยแม่เหล็กแสดงดังรูปที่ 2.1




**รูปที่** 2.1 ตัวอย่างการสวมชุดตรวจจับโดยใช้แม่เหล็ก และเครื่องส่งคลื่นสนามแม่เหล็ก [2]

2. การตรวจจับการเคลื่อนไหวโดยใช้กล้อง (Optical Motion Capture)

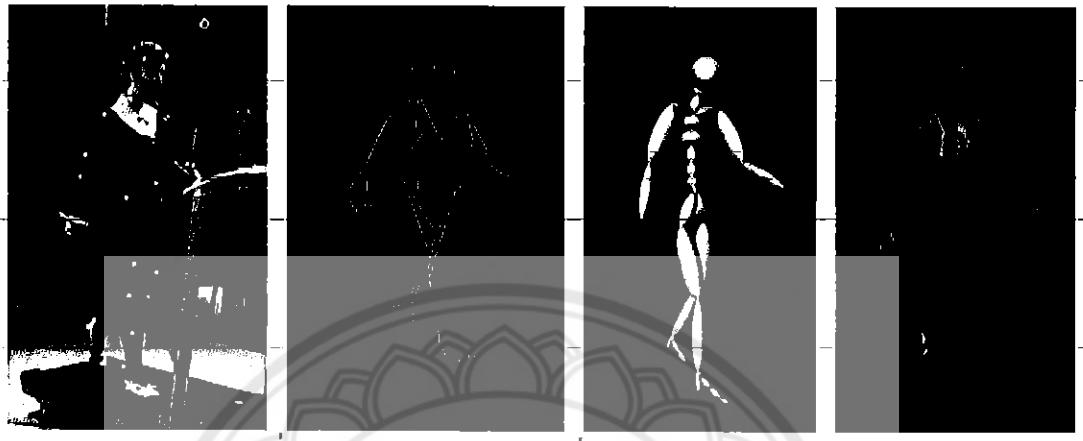
เป็นการใช้ถ่ายภาพวิดีโอชนิดพิเศษในการติดตามตำแหน่งของมาร์กเกอร์ที่ติดอยู่ตามส่วนต่างๆ โดยกล้องวิดีโอนี้จะใช้เลนส์แบบรับแสงอินฟราเรด เพื่อให้ได้ภาพที่แสดงตำแหน่งของจุดต่างๆ สำหรับวิธีการนี้ยังแบ่งออกไปได้อีก 2 แบบตามลักษณะมาร์กเกอร์ที่ใช้ ได้แก่

1) มาร์กเกอร์แบบสะท้อนแสงอินฟาเรด (Reflective Marker) ใช้แสงอินฟาเรดจากแหล่งกำเนิดแสงที่ติดไว้รอบๆ กล้องถ่ายภาพ แสงจะสะท้อนที่มาร์กเกอร์ทำให้เห็นเป็นจุดที่มีความเข้มแสงมากกว่าบริเวณอื่นๆ

2) มาร์กเกอร์แบบหลอด LED (Pulsed-LED) วิธีนี้แหล่งกำเนิดแสงจะอยู่ที่ตัวมาร์กเกอร์ การตรวจจับจะใช้วิธีวัดความเข้มแสงจากหลอด LED โดยตรง

จากนั้นจะตรวจจับตำแหน่งมาร์กเกอร์จากหลายๆ กล้องมารวมกัน เพื่อคำนวณโดยใช้ทฤษฎีสามเหลี่ยม แล้วได้ตำแหน่งใน3 มิติ วิธีนี้มักมีปัญหา เช่น การสลับมาร์กเกอร์ การรบกวน และการบดบังของมาร์กเกอร์เป็นต้น โดยในหลายๆ ระบบมักมีการนำเอาโครงกระดูกมาใช้ในการสร้างการเคลื่อนไหวด้วย โดยโครงกระดูกจะช่วยให้การเคลื่อนไหวมีความถูกต้องมากขึ้น เนื่องจาก

มองถึงความสัมพันธ์ระหว่างข้อต่อต่างๆ ด้วย ไม่เหมือนกับตำแหน่งที่ตรวจจับได้ที่เป็นตำแหน่งของจุดแต่ละจุดแยกกัน ซึ่งไม่มีความสัมพันธ์ใดๆ ต่อกันเลย ตัวอย่างการตรวจจับการเคลื่อนไหวโดยใช้กล้องแสดงดังรูปที่ 2.2



**รูปที่ 2.2** ตัวอย่างการจับการเคลื่อนไหวโดยใช้กล้อง [3]

3. การตรวจจับการเคลื่อนไหวโดยใช้ชุดเชิงกล (Mechanical Motion Capture)

การตรวจจับการเคลื่อนไหวโดยใช้ชุดเชิงกล มักใช้กับการจับการเคลื่อนไหวของมนุษย์โดยเฉพาะ โดยใช้ชุดที่ทำขึ้นพิเศษสำหรับให้มนุษย์สวมใส่สำหรับการตรวจจับการเคลื่อนไหว โดยชุดจะมีลักษณะเป็นโครงสร้างที่เชื่อมต่อกันโดยมีตัวต้านทานปรับค่าได้ในการวัดการหมุนของจุดต่างๆ ตามข้อต่อสำคัญของร่างกาย การรู้มุมการหมุนของจุดต่างๆ ทำให้เราสามารถรู้ท่าทางการเคลื่อนไหวของผู้สวมชุดได้ การตรวจจับการเคลื่อนไหวโดยใช้ชุดเชิงกลแสดงดังรูปที่ 2.3

**รูปที่ 2.3** ตัวอย่างการจับการเคลื่อนไหวโดยใช้ชุดเชิงกล [4]

## 2.2 Digital Image Processing [5]

Digital Image Processing คือ การเอาภาพในรูปแบบ 3 มิติ มาประมวลผลโดยการคำนวณทางคอมพิวเตอร์ การคิดคำนวณนั้นมีหลายวิธี ซึ่งแต่ละวิธีก็มีประโยชน์แตกต่างกันไป ไม่ว่าจะเป็นการนำเอาสี (Color) แต่ละจุด (Pixel) มาคิด การคิดคำนวณเป็นบริเวณหลายๆจุดรวมๆกัน (Area) เช่น การดูลวดลาย (Pattern, Texture) การวิเคราะห์หารูปร่าง (Shape) และการวิเคราะห์แบบอื่นๆ

แหล่งของรูปภาพนั้นอาจจะมาจากกล้อง 3 มิติ สแกน หรือจากสื่อ 3 มิติ ต่างๆ แล้วนำไปผ่านกระบวนการบางอย่างเพื่อให้เกิดเป็นภาพใหม่ เช่น การทำภาพเบลอ (Blued Image) การทำภาพนูน (Emboss-Image) การตรวจหาขอบภาพ (Edge Detector) ซึ่งศาสตร์ด้านนี้สามารถนำไปใช้ประโยชน์ได้มากมายหลายด้าน เช่น ทางด้านการแพทย์ การรักษาความปลอดภัย ตรวจนับจำนวนคน หรือตรวจสอบการเคลื่อนที่ของวัตถุต่าง ๆ ภายในภาพ

การประมวลผลภาพดิจิทัล (digital image processing) เป็นสาขาที่กล่าวถึงเทคนิคและอัลกอริทึมต่างๆ ที่ใช้การประมวลผลภาพที่อยู่ในรูปแบบดิจิทัล (ภาพดิจิทัล) ภาพในที่นี้ รวมความหมายถึงสัญญาณดิจิทัลใน 2 มิติอื่นๆ โดยทั่วไปคำนี้เมื่อใช้อย่างกว้างๆ จะครอบคลุมถึงสัญญาณวีดิโอ (video) หรือภาพเคลื่อนไหว ซึ่งจะเป็นชุดของภาพนิ่ง เรียกว่า เฟรม (frame) หลายๆภาพต่อกันไปตามเวลา ซึ่งก็คือสัญญาณ 3 มิติ เมื่อนับเวลาเป็นมิติที่ 3 หรือ อาจจะครอบคลุมถึงสัญญาณ 3 มิติอื่นๆ เช่น ภาพ 3 มิติทางการแพทย์ หรือ อาจจะมากกว่านั้น เช่น ภาพ 3 มิติ และ หลายชนิด (Multimodal image)

### 2.2.1 รูปร่างของภาพ (Image Shape)[6]

วัตถุที่มีอยู่ตามธรรมชาติและที่มนุษย์สร้างขึ้นมีรูปร่างที่แตกต่างกันไป ทั้งที่เป็นรูปทรงเรขาคณิตและไม่เป็นรูปทรงเรขาคณิต ในศาสตร์ของการประมวลผลภาพนั้น การกำหนดขอบเขตของภาพทุกภาพให้อยู่ในรูปสี่เหลี่ยม (Rectangular image model) เป็นวิธีที่นิยมใช้กันมากที่สุด เนื่องจากทำให้การอ่านภาพ การจัดเก็บข้อมูลภาพในหน่วยความจำ และการแสดงภาพออกทางอุปกรณ์ต่าง ๆ เป็นไปได้อย่างมีประสิทธิภาพ

การเก็บข้อมูลภาพลงหน่วยความจำของคอมพิวเตอร์สามารถทำได้โดยการจองหน่วยความจำของเครื่องไว้ในรูปของตัวแปรอะเรย์ (array) โดยค่าในแต่ละช่องของอะเรย์แสดงถึงคุณสมบัติของจุดภาพ (pixel) และตำแหน่งของช่องอะเรย์เป็นตัวกำหนดตำแหน่งของจุดภาพ

สมมุติให้ Image เป็นตัวแปรแบบอะเรย์ขนาด $M \times N$ ($M$ แถว และ $N$ คอลัมน์) ที่ใช้เก็บภาพขนาด $M \times N$ จุด ($M$ จุดในแนวนอน และ $N$ จุดในแนวตั้ง) ค่าสี (หรือความสว่าง ในกรณีที่เป็นภาพ grey level) ของจุดภาพในแถวที่ 5 คอลัมน์ที่ 4 จะตรงกับค่าของ Image(5, 4) จะเห็นว่าเราใช้ตำแหน่งของจุดภาพทั้งสองแกนเป็นตัวชี้ค่าข้อมูลในอะเรย์

จากการใช้หน่วยความจำเพื่อการเก็บภาพในลักษณะที่กล่าวมา เนื้อที่ในการเก็บภาพสามารถคำนวณได้จาก $M \times N \times g$ เมื่อ $g$ เป็นจำนวนเต็มที่แทนจำนวนบิตของข้อมูลในแต่ละจุดภาพ ตัวอย่างถ้า $g$ มีค่าเท่ากับ 8 บิต เราจะสามารถเก็บความแตกต่างของระดับสีที่เป็นไปสูงสุด 256 ระดับ ค่า M และ N จะเป็นตัวบอกถึงความละเอียดของภาพ สำหรับคอมพิวเตอร์ทั่วไปในระบบ VGA (Video Graphic Array) จะมีขนาด 640x480, 800x600 และ 1024x768 จุด เป็นต้น การกำหนดความละเอียดจะขึ้นอยู่กับงานที่จะใช้ ในงานบางอย่างใช้ความละเอียดแค่ 30 x 50 จุด ก็พอแล้วแต่ในงานบางชนิด ใช้ความละเอียดถึง 1000 x 1000 จุด ก็ยังไม่พอ

ปกติแล้วในการเก็บข้อมูลภาพโดยเครื่องมือต่าง ๆ จะเก็บตามมาตรฐานของโทรทัศน์ซึ่งมีอัตราส่วน x ต่อ y เท่ากับ 4:3 สำหรับเครื่องมือเก็บข้อมูลภาพที่ไม่เป็นไปตามอัตราส่วน 4:3 เมื่อนำภาพนี้ไปแสดงในจอภาพมาตรฐานจะทำให้ภาพที่แสดงนั้นมีขนาดของจุดภาพไม่ เป็นสีเหลี่ยมจัตุรัสเช่นในบางระบบอาจจะใช้ความละเอียดในการแสดงเท่ากับ 640 x 512 ซึ่งจะทำให้ขนาดของจุดภาพที่ได้มีขนาดของด้านกว้างมีความยาวมากกว่าด้านสูง ซึ่งลักษณะดังกล่าวนี้เป็นหัวข้อที่ต้องสนใจสำหรับการเขียนโปรแกรมทางด้าน กราฟฟิกและการจัดการข้อมูล

จำนวนสีสูงสุดที่เป็นไปได้ของแต่ละะจุดภาพขึ้นอยู่กับจำนวนบิตที่ใช้ เมื่อมีการกำหนดให้ขนาดของบิตต่อจุด มากขึ้นจะทำให้จำนวนของสีมากขึ้นด้วย ตัวอย่างเช่น

1 บิต = $2^1$ = 2 สี

2 บิต = $2^2$ = 4 สี

4 บิต = $2^4$ = 16 สี

8 บิต = $2^8$ = 256 สี

16 บิต = $2^{16}$ = 65,536 สี เป็นต้น

สำหรับการแสดงข้อมูลภาพที่มีขนาด 1 บิตและ 8 บิตนั้นจะมีการทำงานที่จะใกล้เคียงกัน เนื่องจากหน่วยประมวลผลจะไม่สามารถ จัดการกับข้อมูลที่เป็นบิตเดี่ยว ๆ ได้ดังนั้นในการแสดงข้อมูลออกทางจอภาพตัวโปรเซสเซอร์จะทำการก็อปปี้ข้อมูล ทั้ง 8 บิต(1 Byte) ส่งให้กับจอภาพซึ่งในกรณีที่ Pixel มีขนาด 1 บิต เมื่อโปรเซสเซอร์จะทำงานกับบิตแรกที่ต้องการแล้วก็จะทำการก็อปปี้ข้อมูลชุด ใหม่ทันที่โดยที่ไม่เกี่ยวกับข้อมูลอีก 7 บิตที่เหลือส่วนในกรณี Pixel ที่มีขนาด 8 บิต โปรเซสเซอร์จะทำการก็อปปี้ข้อมูลจุดใหม่ก็ต่อเมื่อโปรเซสเซอร์ทำงานกับทุก บิตแล้ว

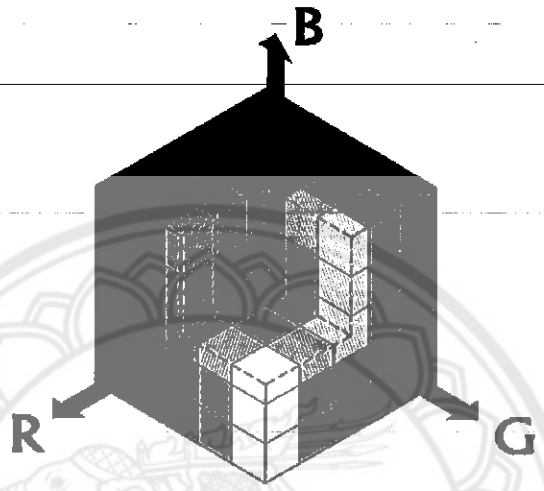
ตัวอย่างสำหรับระบบที่มีความละเอียดเท่ากับ 800x600 และมีขนาด 16 บิตต่อ Pixel จะสามารถแสดงสีได้ทั้งหมด 65536 ระดับและต้องใช้เนื้อที่ในการเก็บเท่ากับ 800x600x16 บิต

**2.2.2 รูปแบบสี (Color Model) [6]**

ก่อนที่จะทำการประมวลผลภาพสีนั้น เราจำเป็นที่จะต้องเข้าใจรูปแบบของสีที่ใช้ในคอมพิวเตอร์เสียก่อน ซึ่งจะอธิบายรูปแบบที่สำคัญ 2 แบบ ได้แก่

**2.2.2.1 รูปแบบ RGB**

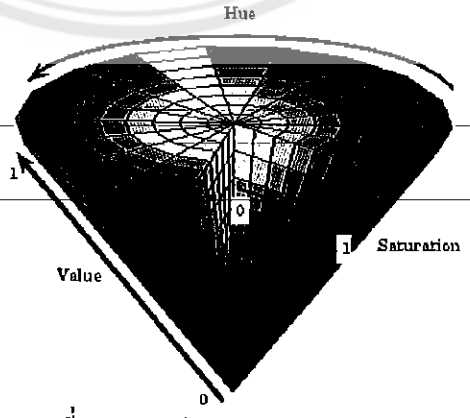
เป็นระบบสีพื้นฐานของคอมพิวเตอร์ที่ใช้ในการแสดงผล โดยจุดย่อยของภาพ (Pixel) จะประกอบด้วยค่าสี 3 ค่า คือ แดง (R) เขียว (G) และน้ำเงิน (B) การผสมสีทั้งสามนี้ด้วยค่าต่างๆ กัน จะก่อให้เกิดสีที่แตกต่างกัน โดยคอมพิวเตอร์จะเก็บค่าสีนี้แยกกัน โดยใช้ขนาดข้อมูล 1 ไบต์ต่อ 1 สี ทำให้ค่าของสีนั้นมีได้ 256 ระดับ และผสมได้สีทั้งหมด 16 ล้านสี




**รูปที่ 2.4** ตัวอย่างสีรูปแบบ RGB [7]

**2.2.2.2 รูปแบบ HSV**

เป็นระบบสีที่ประกอบไปด้วยค่า 3 ค่าคือ ค่าสี (Hue) บอกความเป็นสีใดๆ ค่าความอิ่มตัวของสี (Saturation) บอกความสีหรือขาว-ดำ และค่าความสว่าง (Intensity, Value) บอกความขาวหรือดำ รูปแบบสีนี้จะเหมาะกับการประมวลผลภาพที่ต้องแยกแยะสี เพราะสามารถใช้ค่า Hue เพียงค่าเดียวก็สามารถดูความแตกต่างของสีได้



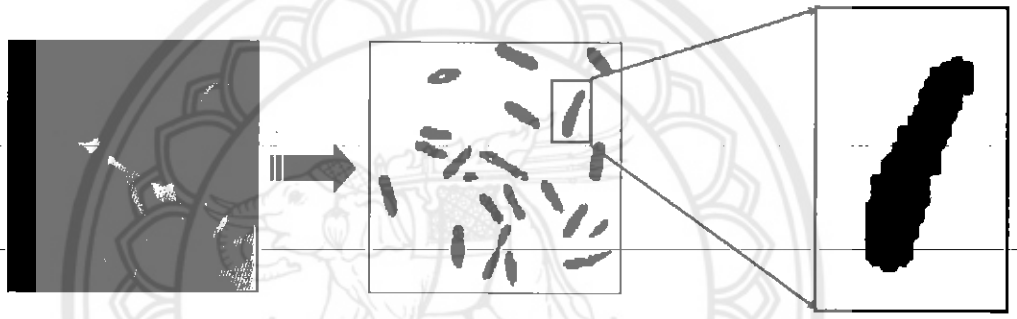

**รูที่ 2.5** ตัวอย่างสีรูปแบบ HSV [8]

**2.2.3 การแยกองค์ประกอบภาพ (Image Segmentation) [6]**

**Image segmentation** ใช้ในการแยกองค์ประกอบต่างๆของรูปภาพออกจากกันตามลักษณะสำคัญที่เราพิจารณา วิธีการพื้นฐานสำหรับการ Segmentation คือการพิจารณา Image amplitude (ได้แก่การพิจารณาความสว่างของภาพสำหรับภาพแบบ Gray scale และความแตกต่างของสีสำหรับภาพสี) นอกจากนี้ขอบของภาพ (edge) และลักษณะของพื้นผิว (Texture) ก็เป็นองค์ประกอบหนึ่งที่จะทำให้สามารถทำการ Segmentation ได้สะดวกยิ่งขึ้น

ประโยชน์ของการแยกองค์ประกอบภาพ

1. ลดจำนวนข้อมูลในรูปภาพที่ไม่จำเป็นในการวิเคราะห์ลง
2. จัดระเบียบข้อมูลในรูปภาพให้เป็นกลุ่มได้ดีขึ้น
3. แสดงข้อมูลในรูปที่เข้าใจง่ายขึ้น



**รูปที่ 2.6** ตัวอย่างการแยกองค์ประกอบภาพ [6]

หลักการในการแยกองค์ประกอบภาพ มี 2 หลักที่สำคัญ ได้แก่

1. การแยกองค์ประกอบตามความเหมือนกัน (similarity) ของคุณสมบัติของ พิกเซลของรูปภาพภายในพื้นที่เดียวกัน

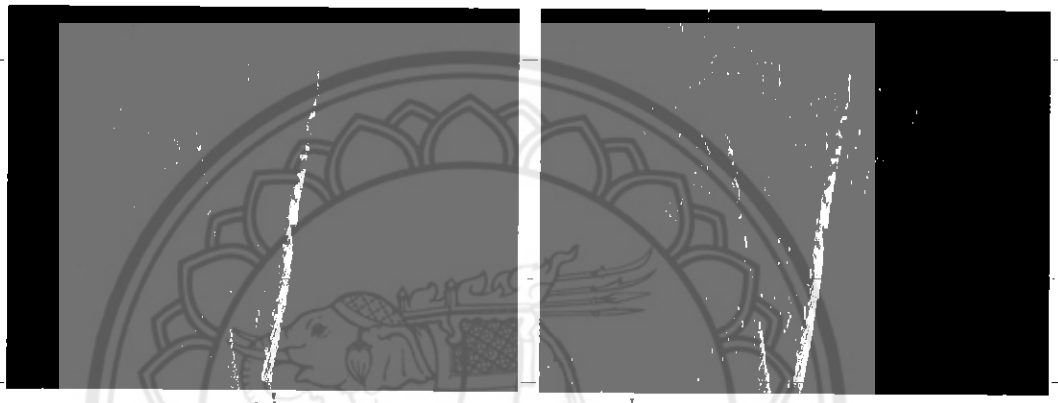
2. การแยกองค์ประกอบ โดยดูจากความไม่ต่อเนื่อง (Discontinuity) ของคุณสมบัติของพิกเซลบริเวณรอยต่อระหว่างวัตถุในภาพกับฉากหลัง

สำหรับในโครงงานนี้ ใช้หลักการแยกองค์ประกอบภาพ เพื่อแยกมาร์คเกอร์ในภาพที่ได้จากกล้อง ซึ่งมีลักษณะเป็นจุด จำนวนหลายจุด แล้วนำแต่ละจุดที่เอาไปคำนวนหาตำแหน่งในแกน3มิติ

**2.2.4 การแยกแยะสี (Threshold) [6]**

Threshold เป็นการเปลี่ยนสีของจุดย่อย (Pixel) ให้เป็นสีที่ต้องการหากค่าสีที่พิจารณาอยู่ในช่วงที่ต้องการ (Threshold Range) หากไม่อยู่ในช่วงก็จะเปลี่ยนเป็นอีกสีหนึ่ง ซึ่งจะเป็นการแยก

เอาเฉพาะส่วนที่เป็นมาร์กเกอร์ออกมาให้เห็นได้ชัด การทำ Threshold ในโครงงานนี้จะทำ Threshold ในระบบสี HSV โดยจะเลือกเอาส่วนใดๆของภาพที่มีลักษณะเป็นสีที่ต้องการ (คือส่วนที่เป็นมาร์กเกอร์) จึงต้องจัด สภาพแวดล้อม อื่นให้เหมาะสมด้วยจึงจะสามารถทำการ Threshold ได้ดี ในที่นี้จะใช้ค่าทั้ง 3 ค่าในระบบ HSV คือ H (Hue) เป็นค่าสีว่าเป็นสีใด ในการตรวจสอบ ค่านี้จะมีระดับตั้ง $0^{\circ} - 360^{\circ}$ ค่า ค่า S (Saturation) ค่าความเข้มสี มีค่าตั้งแต่ 0 – 100% และ V (Value ) ค่าความสว่าง มีค่าตั้งแต่ 0 – 100% ซึ่งถ้าเป็นส่วนที่สว่างมาก ค่า V ก็จะมากขึ้นตามเช่นเดียวกัน ที่ตำแหน่งไหนมีค่าสีอยู่ในช่วงที่ต้องการ และมีค่า V และ S มากเท่าที่เราต้องการ ก็แสดงว่าตำแหน่งนั้นคือส่วนที่เป็นมาร์กเกอร์ ก็ให้ทำให้พิกเซลนั้นเป็นสีที่เราสนใจ



รูปที่ 2.7 ตัวอย่างการทำ threshold เพื่อตัดสีฟ้าออก [6]

**2.2.5 การกรองข้อมูลภาพ (Filtering) [6]**

การกรองข้อมูลภาพ (Image Filtering) คือการนำภาพไปผ่านตัวกรองสัญญาณเพื่อให้ได้ภาพผลลัพธ์ออกมา ภาพผลลัพธ์ที่ได้จะมีคุณสมบัติแตกต่างจากภาพเริ่มต้น วัตถุประสงค์หลักของการกรองข้อมูลภาพคือการเน้น (enhance) หรือลดทอน (attenuate) คุณสมบัติบางประการของภาพเพื่อให้ได้ภาพที่มีคุณสมบัติตามต้องการ

การกรองข้อมูลภาพคือการประมวลผลภาพอย่างหนึ่งที่จำเป็นมาก เนื่องจากในการใช้งานจริง ภาพที่ได้มามักมีสัญญาณรบกวน หรือสัญญาณไม่พึงประสงค์อื่นๆ ปะปนอยู่ด้วย การกรองข้อมูลภาพสามารถปรับปรุงให้ภาพมีคุณสมบัติที่ดีขึ้น เหมาะแก่การประมวลผลในขั้นต่อไป

**2.2.5.1 การกรองข้อมูลภาพโดยใช้ค่าเฉลี่ยแบบคณิตศาสตร์ (Mean filtering)**

วิธีการนี้จะใช้ค่าเฉลี่ยแบบคณิตศาสตร์ของจุดทั้งหมด หากมีภาพขนาด N×M ทั้งหมด K ภาพ เราสามารถคำนวณหาภาพใหม่ได้ดังนี้

$$\hat{I}(x,y) = \frac{1}{K}\sum_{j-1}^{K} I_j(x,y)$$

$\hat{I}(x, y)$ คือ ความเข้มแสงของจุด ณ ตำแหน่ง (x, y) ในภาพผลลัพธ์

$I_j(x, y)$ คือ ความเข้มแสงของจุด ณ ตำแหน่ง (x, y) ในภาพที่ j

วิธีนี้เป็นการลดทอนสัญญาณรบกวน ภาพที่ได้จะมีสัญญาณรบกวนลดลง

**2.2.5.2 การกรองข้อมูลภาพโดยใช้ค่ามัธยฐาน (Median filtering)**

วิธีการนี้จะนำเอาความเข้มแสงของจุดที่ตรงกันในภาพต่างๆ มาเรียงลำดับ (sort) จากน้อยไปหามาก จากนั้นจะเลือกค่าที่อยู่ตรงกลางไปใช้ หากจำนวนภาพทั้งหมดเป็นจำนวนคู่ ค่าทั้งสองที่อยู่ตรงกลางจะนำมาหาค่าเฉลี่ย วิธีการนี้จะต้องใช้การเรียงลำดับซึ่งเป็นกระบวนการที่ใช้เวลาในการคำนวณสูง แต่ข้อดีคือไม่สูญเสียความคมชัด

ตัวอย่าง

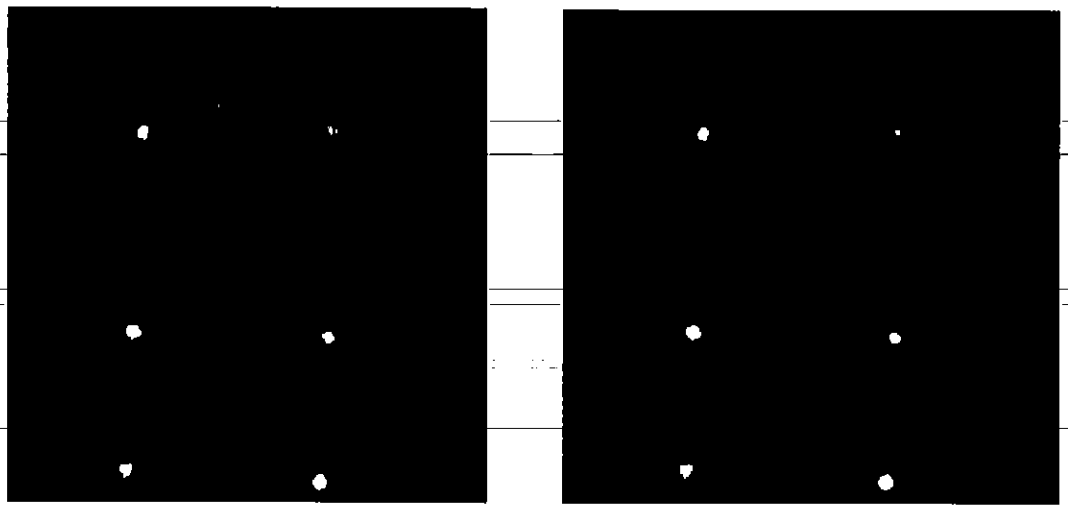
| ภาพที่ 1 | ภาพที่ 2 | ภาพที่ 3 | ผลลัพธ์ |
|---|---|---|---|
| 1 2 1 3 | 2 3 4 3 | 3 2 1 4 | 2 2 1 3 |
| 4 2 2 1 | 5 3 4 1 | 2 1 4 0 | 4 2 4 1 |
| 0 1 1 3 , | 3 2 4 2 , | 1 4 2 0 = | 1 2 2 2 |
| 2 2 1 1 | 1 3 1 2 | 2 4 0 2 | 2 3 1 2 |

**2.2.5.3 การกรองข้อมูลภาพโดยใช้ค่าฐานนิยม (Modal filtering)**

วิธีการนี้คล้ายกับวิธีใช้ค่ามัธยฐาน แต่ไม่ใช้การเรียงลำดับข้อมูล ระดับความเข้มแสงที่ใช้บ่อยที่สุดจะถูกเลือกไปใช้ วิธีนี้เสมือนการโหวตลงคะแนนเสียง ผู้ที่ได้คะแนนเสียงสูงที่สุดคือผู้ชนะ วิธีนี้เหมาะสำหรับการลดสัญญาณรบกวนที่เกิดขึ้นไม่บ่อย

ตัวอย่าง

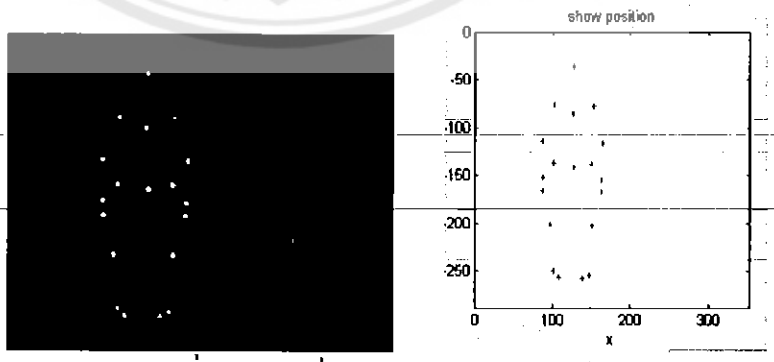
| ภาพที่ 1 | ภาพที่ 2 | ภาพที่ 3 | ผลลัพธ์ |
|---|---|---|---|
| 1 2 1 3 | 2 3 4 3 | 3 2 1 4 | 2 2 1 3 |
| 4 2 2 1 | 5 3 4 1 | 2 1 4 0 | 4 2 4 1 |
| 0 1 1 3 , | 3 2 4 2 , | 1 4 2 0 = | 1 2 2 2 |
| 2 2 1 1 | 1 3 1 2 | 2 4 0 2 | 2 3 1 2 |



รูปที่ 2.8 ภาพตัวอย่างการทำ Median Filter [1]

### 2.2.6 Blob Analysis [1]

Blob Anlysis คือการหาขนาดและตำแหน่งของวัตถุภายในภาพซึ่งมีความแตกต่างจากพื้นหลัง ในโครงงานนี้นำภาพจากการทำ Threshold และ Filtering มาแยกตัวมาร์กเกอร์ที่ตรวจจับเจอกับพื้นหลัง โดยจะทำการตรวจสอบภาพทีละพิเซล หากเป็นสีขาวจะทำการระบุ ลำดับ ของมาร์กเกอร์ (ID) หากเป็นสีดำหมายถึงพื้นหลังจะระบุ ID เป็น -1 ทำต่อในพิกเซลถัดไปถ้าหากพิกเซลที่ติดกันมีสีเดียวกันและถูกระบุ ID แล้ว จะระบุ ID ให้พิกเซลนี้มี ID เดียวกันนับพิกเซลที่ติดกันมีสีเดียวกันและถูกระบุ ID แล้ว ทำเช่นนี้ต่อไปกับทุกๆ พิกเซล จากนั้นเราก็นำมาทำการหาพิกัดของมาร์กเกอร์แต่ละตัวที่ตรวจเจอแล้วนำค่าตำแหน่งนั้นส่งต่อไปทำการประมวลผลหาตำแหน่งทั้ง 3 มิติ ต่อไป โดยวิธีการหาตำแหน่งมาร์กเกอร์แต่ละตัวนี้จะใช้วิธีเก็บค่าสูงสุดต่ำสุดของ x และ y ของทุก Blob แล้วนำมาหาค่าเฉลี่ย จะได้ตำแหน่ง x เฉลี่ย และ y เฉลี่ยเพื่อกำหนดเป็นพิกัดของมาร์เกอร์




รูปที่ 2.9 ภาพที่ใช้หา Blob และผลลัพธ์[1]

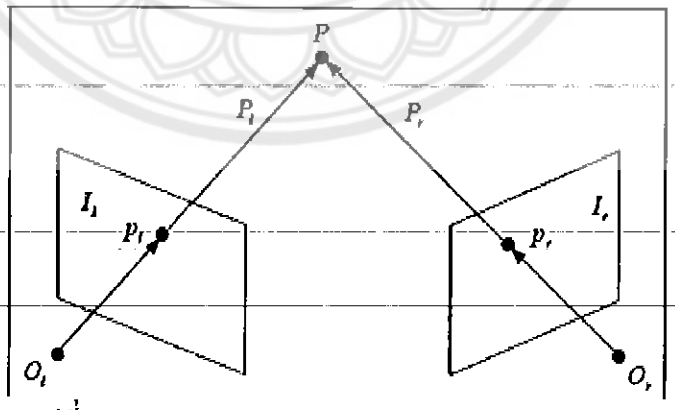
## 2.3 ระบบสเตอริโอวิชั่น (Stereo Vision System) [9]

### 2.3.1 หลักการพื้นฐาน

ระบบสเตอริโอวิชั่น (Stereo Vision System) หรือระบบสเตอริโอ (Stereo System) เป็นเทคนิคการวัดโดยการใช้กล้องตั้งแต่ 2 ตัวขึ้นไป ถ่ายภาพวัตถุจากมุมมองที่แตกต่างกัน ทำให้เกิดภาพซึ่งต่างกัน เมื่อนำภาพทั้งสองมารวมกันจะทำให้เกิดระยะลึกของภาพและสามารถนำมาคำนวณหาระยะลึกนั้นได้ ซึ่งเป็นลักษณะเดียวกับการมองเห็นภาพ 3 มิติของมนุษย์ ตัวอย่างของภาพถ่ายจากระบบสเตอริโอ แสดงใน รูปที่ 2.10

**รูปที่ 2.10** ตัวอย่างภาพของระบบสเตอริโอแบบถ่ายจากกล้อง 2 ตัว [9]

ระบบสเตอริโอคำนวณพิกัดของจุด โดยการนำภาพถ่ายจากกล้องแต่ละตัวมาค้นหาตำแหน่งของภาพที่เป็นจุดเดียวกันบนวัตถุ เพื่อนำ มาคำนวณหาค่าพิกัดของวัตถุที่ตำแหน่งนั้น หลักการในการหาค่าพิกัด แสดงในรูปที่ 2.11



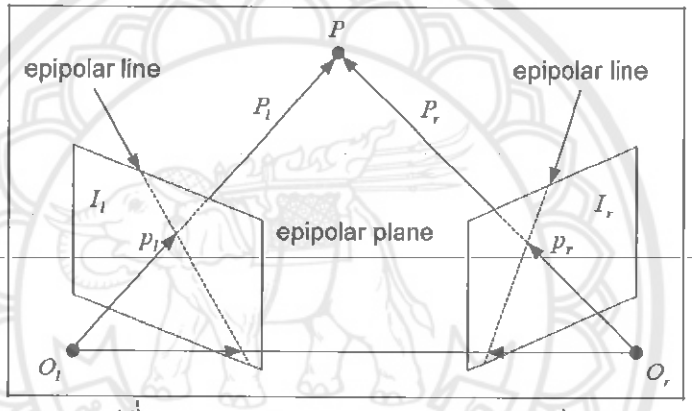

**รูปที่ 2.11** หลักการคำนวณพิกัดของระบบสเตอริโอ[9]

รูปที่ 2.11 เป็นแผนภาพของระบบสเตอริโอแบบ 2 กล้องที่แนวการมองของกล้องทั้งสองตัวไม่ขนานกัน โดยที่ $I_l$ และ $I_r$ เป็นระนาบที่เกิดภาพของกล้องซ้ายและขวาตามลำดับ กำหนดให้จุด $P_l$ และ $P_r$

เป็นจุดที่มาจากตำแหน่งเดียวกันบนวัตถุ เมื่อทำการสร้างรังสีจากจุดกึ่งกลางเลนส์ของกล้องทั้งสองหรือจุด $O_l$ และจุด $O_r$ ไปยังจุด $P_l$ และ $P_r$ ตามลำดับ จะได้รังสีสองเส้นตัดกันที่จุดๆ หนึ่งใน 3 มิติซึ่งจุดดังกล่าวคือจุด $P$ ซึ่งเป็นตำแหน่งของวัตถุ ณ ตำแหน่งนั้น

**2.3.2 เรขาคณิตอีพิโพลาร์ (Epipolar Geometry)**

เรขาคณิตอีพิโพลาร์ (Epipolar Geometry) เป็นหลักการที่ช่วยให้การค้นหาตำแหน่งของภาพที่มาจากจุดเดียวกันบนวัตถุอยู่ในแนวของเส้นตรงเส้นหนึ่งที่เรียกว่าเส้นอีพิโพลาร์ (Epipolar Line)



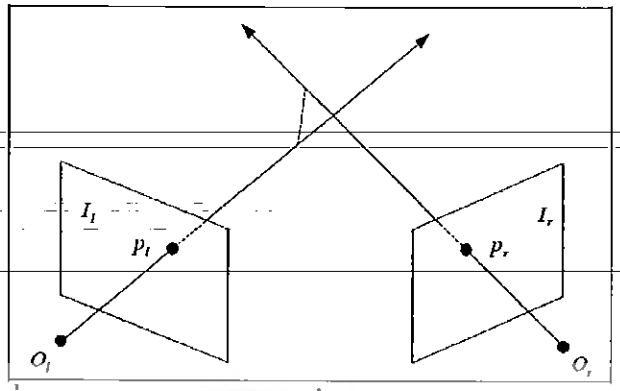

**รูปที่ 2.12** ลักษณะของเรขาคณิตอีพิโพลาร์[9]

พิจารณาแผนภาพของระบบสเตอริโอในรูปที่ 2.12 กำหนดให้ ระนาบที่เกิดจากจุดศูนย์กลางของกล้องทั้งสองและจุดบนวัตถุเรียกว่า ระนาบอีพิโพลาร์ (Epipolar Plane) และตำแหน่งที่ระนาบอีพิโพลาร์ ตัดกับระนาบของภาพเกิดเป็นเส้นตรงมีชื่อว่า เส้นอีพิโพลาร์ (Epipolar Line) สมมติให้จุด P คือตำแหน่งของวัตถุใน 3 มิติ, เวคเตอร์ $p_l = [x_l, y_l, z_l]^T$ และ $p_r = [x_r, y_r, z_r]^T$ เป็นเวคเตอร์ที่ชี้ไปยังจุดที่ได้จากการฉายจุด $P$ บนระนาบภาพของกล้องซ้ายและขวา ตามลำดับ เมื่อกำหนด $p_l$ ในภาพทางซ้ายเพื่อเริ่มทำการค้นหาจุดที่ตรงกันในภาพทางขวาจะพบว่าจุดที่ตรงกันหรือจุด $p_r$ จะอยู่บนเส้นอีพิโพลาร์เสมอ

**2.3.4 การคำนวณพิกัด 3 มิติ**

ในทางปฏิบัติ การคำนวณพิกัด 3 มิติโดยใช้หลักการในรูปที่ 2.11 ไม่สามารถทำได้ เนื่องจากความผิดพลาดในการหาจุดของภาพที่ตรงกัน และความผิดพลาดของค่าพารามิเตอร์ของ

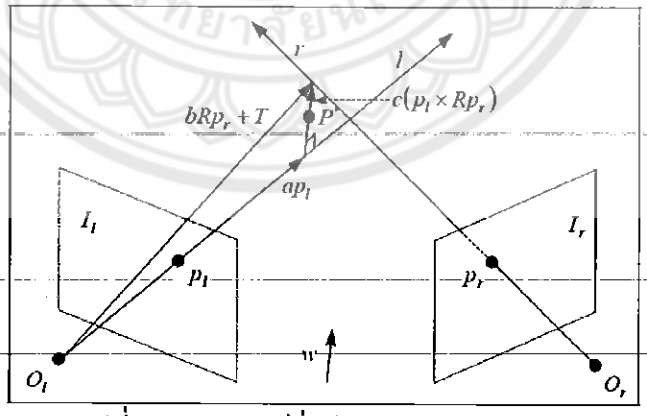
กล้องที่ได้จากการสอบเทียบกล้องซึ่งใช้ในการแปลงพิกัดระหว่างระบบพิกัดต่างๆ ทำให้รังสีที่เกิดขึ้นไม่ตัดกันจริงดังรูปที่ 2.13




**รูปที่ 2.13** รังสีจากจุดของภาพที่ตรงกันไม่ตัดกันใน 3 มิติ [9]

ดังนั้นการหาค่าพิกัด 3 มิติโดยใช้ระบบสเตอริโอจึงเป็นการหาตำแหน่งพิกัดใน 3 มิติของจุดที่มีระยะห่างจากรังสีทั้งสองน้อยที่สุด

พิจารณารูปที่ 2.14 กำหนดให้ $p_l$ และ $p_r$ เป็นคู่จุดของภาพที่ตรงกันที่อยู่ในระบบพิกัดกล้องขของกล้องตัวซ้ายและขวาตามลำดับ, $ap_l$ โดย $a \in R$ เป็นรังสี *l* ที่ผ่านจุด $O_l$ *(a=0)* และ $p_l$ *(a=1)* และให้ $T+bRp_r$ โดย $b \in R$ เป็นรังสี *r* ที่ผ่านจุด $O_r$ *(b=0)* และ $p_r$*(b=1)* อยู่ในระบบพิกัดกล้องขของกล้องตัวซ้าย, กำหนดให้ $w = p_l \times Rp_r$ เป็นเวคเตอร์ที่ตั้งฉากกับทั้งรังสี *l* และ *r* ส่วน *c* เป็นความยาวของเวคเตอร์ *w* ที่สั้นที่สุดโดย $c \in R$




**รูปที่ 2.14** เวคเตอร์ที่ใช้ในการหาค่าพิกัด [9]

จากเวคเตอร์ในรูปที่ 2.8 จะได้ สมการ

$$ap_l + c(p_l \times Rp_r) = cRp_r + T \qquad (2.4)$$

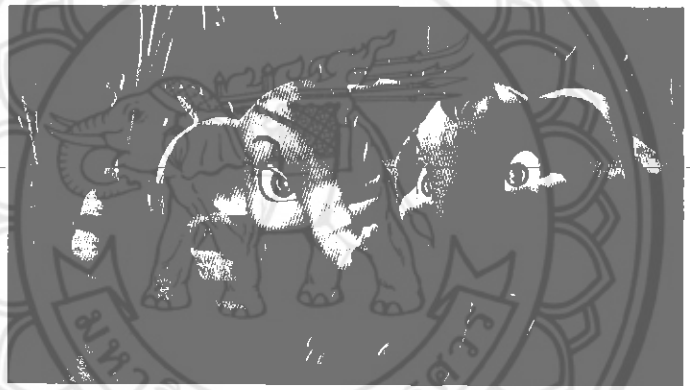
$$ap_l - bRp_r + c(p_l \times Rp_r) = T \qquad (2.5)$$

เมื่อทำการแก้สมการ (2.5) เพื่อหาค่าคงที่ a, b และ c จะสามารถหาค่าพิกัดของจุด $P'$ ได้จาก

$$P' = ap_l + \frac{c}{2}(p_l \times Rp_r) \quad (2.6)$$

## 2.4 ภาพเคลื่อนไหว 3 มิติ (3D Animation)

ภาพเคลื่อนไหว 3 มิติ (3D Animation) คือ ภาพเคลื่อนไหวที่สร้างจากคอมพิวเตอร์ โดยมีภาพเป็นลักษณะ 3 มิติ คือมีความกว้าง ความสูง และความลึก โดยสามารถใช้โปรแกรมเฉพาะสำหรับสร้างภาพ 3 มิติ ดังกล่าวได้ รวมถึงสามารถกำหนดให้เกิดการเคลื่อนไหวได้อีกด้วย ภาพเคลื่อนไหว 3 มิติ ถูกใช้อย่างมากในอุตสาหกรรมภาพยนตร์ การ์ตูน และเกม เนื่องจากสามารถสร้างภาพที่มีความสมจริงได้ และสามารถสร้างภาพที่เป็นไปไม่ได้ในโลกแห่งความเป็นจริงให้เกิดขึ้นมาได้



รูปที่ 2.15 ตัวอย่างภาพยนตร์การ์ตูน 3 มิติ เรื่องก้านกล้วย [10]

### 2.4.1 Character Animation

Character Animation นั่นเป็นงาน 3D Animation อย่างหนึ่งเหมือนกับงาน 3D Animation ทั่วๆ ไป เพียงแต่งาน Character Animation นั้นเราใช้เรียกระบุเฉพาะกับงานที่จะต้องมีการทำงานเกี่ยวข้องกับวัตถุ หรือฉาก ที่มีรูปแบบการเคลื่อนไหว ในลักษณะของมนุษย์หรือสัตว์ ซึ่งการทำงานในลักษณะนี้จะมีความยุ่งยากกว่าการทำ Animation อย่างเห็นได้ชัด เริ่มตั้งแต่การกระทำหุ่นตัวละคร ไปจนจนถึง Character Setup และ Animation

#### 2.4.1.1 Character Setup

Character Setup เป็นขั้นตอนหนึ่งที่สำคัญมากในการสร้างงาน Character Animation โดย Character Setup มีบทบาทเกี่ยวกับการนำเอาหุ่นตัวละครมาทำการสร้างส่วนของกระดูก (Bone) เพื่อกำหนดข้อต่อในการเคลื่อนไหว การกำหนดการยึดเกาะระหว่างพื้นผิวและกระดูกในส่วนต่างๆ

การกำหนด Ik และ Limit (จุดสุดของมุมข้อพับ) ให้หับกระดูก และการกำหนดจุดควบคุมหุ่นตัวละคร (Control Point) ซึ่งการเตรียมงานเหล่านี้ที่ดี จะมีผลโดยตรงให้งานในการสร้างภาพเคลื่อนไหว (Animation) เป็นไปด้วยความราบรื่นยิ่งขึ้น

**2.1.1.2 รูปแบบไฟล์ .BVH**

รูปแบบไฟล์ BVH (Biovision Hierarchy) คือไฟล์สำหรับบันทึกการเคลื่อนไหว ถูกพัฒนาขึ้นโดยบริษัทที่ให้บริการการทำ Motion Capture บริษัทไบโอวิชัน (Biovision) ซึ่งพัฒนาขึ้นมาเพื่อเก็บข้อมูล การเคลื่อนไหวจากการทำ Motion Capture สามารถนำเข้า (Import) เพื่อใช้ในการกำหนดการเคลื่อนไหวสำหรับงาน Character Animation ในโปรแกรมสร้างงาน 3 มิติ ได้หลายโปรแกรม เช่น 3Ds Max, Blender, Maya, Poser, Second Life เป็นต้น

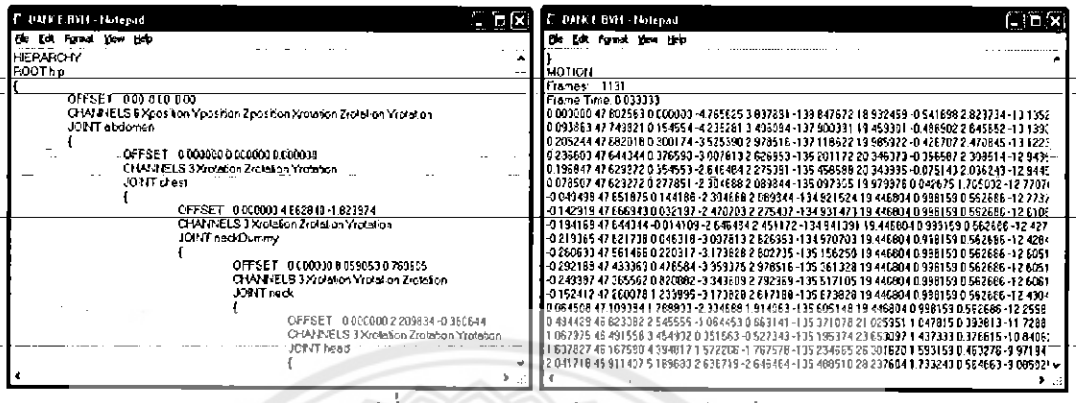
**รูปที่ 2.16** ตัวอย่างการนำเข้าไฟล์ BVH มาใช้งานในโปรแกรม Blender

รูปแบบของไฟล์ BVH มีส่วนประกอบ 2 ส่วนได้แก่ ส่วนความสัมพันธ์ระหว่างอวัยวะ (Model Hierarchy) และส่วนข้อมูลการเคลื่อนไหว (Motion Data)

ส่วนแรก Model hierarchy ส่วนนี้เป็นส่วนกำหนดความสัมพันธ์ของจุดต่างๆ ของโมเดลในลักษณะของโครงสร้างต้นไม้ (tree structure) มีความสัมพันธ์แบบ Parent-Child คือ การเคลื่อนที่ของวัตถุแม่ (parent object) จะมีผลต่อการเคลื่อนที่ของวัตถุลูก (Child object) รวมถึงค่าความต่างเริ่มต้น (initial offset) ของจุดนั้นๆ เทียบกับ วัตถุแม่ (parent object) และกำหนดชนิดพิกัด (Channel) ว่าเป็นแบบระยะห่าง (translation) หรือแบบระยะการหมุน (Rotation) ใน Channel หนึ่งๆ จะประกอบด้วยค่าทิศทาง x, y และ z หรือระยะการหมุนตามแนวแกน x, y และ z

ส่วนที่สอง Motion data จะประกอบด้วย จำนวนเฟรม (Frame: จำนวนเฟรม) ระยะเวลาแสดงของเฟรมหนึ่งๆ (Frame Time: ระยะเวลาของ 1 เฟรมในหน่วยวินาทีต่อเฟรม) และท้ายสุดจะเป็นข้อมูลตัวเลขของ channel ต่างๆ ในรูปแบบของอนุกรม ซึ่ง 1 บันทัด จะแทน 1 เฟรม ส่วนชุด

ของตัวเลข หากเป็น Translation Channel ค่าตัวเลขแต่ละตัว คือ ระยะจาก parent object ตามแนวแกน ไม่มีหน่วย ถ้าหากเป็น Rotation Channel ตัวเลขแต่ละตัวหมายถึง มุมที่อ้างอิงจาก parent object ตามแนวแกน มีหน่วยเป็นองศา

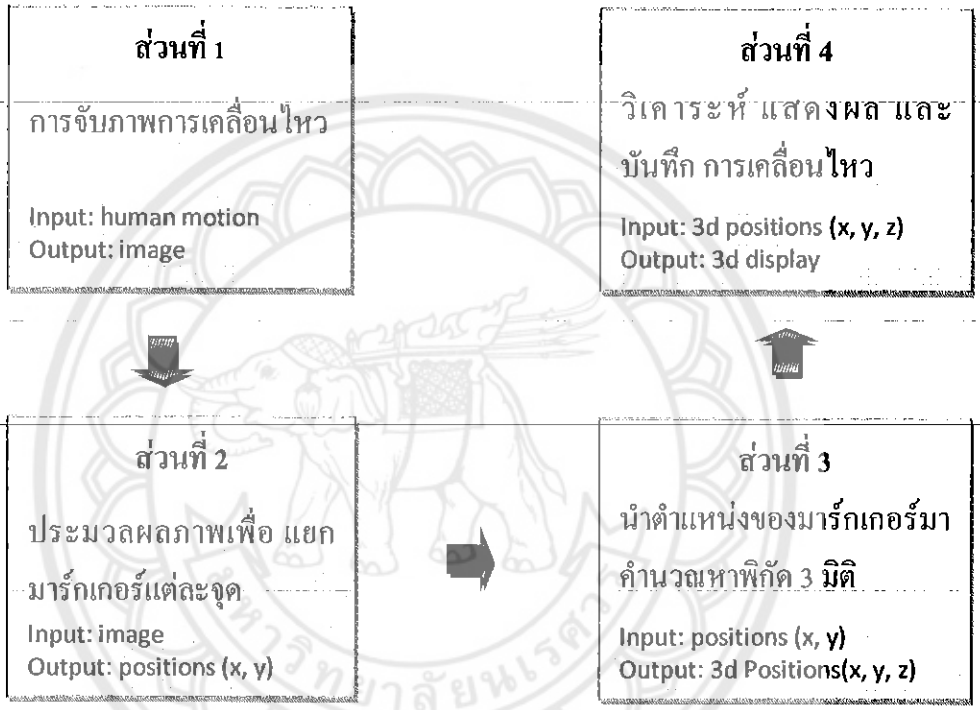



**รูปที่ 2.17** ตัวอย่างข้อมูลภายในไฟล์ BVH

# บทที่ 3
# ขั้นตอนการดำเนินงาน

ในบทนี้จะเป็นการกล่าวถึงขั้นตอนในการออกแบบและดำเนินงานต่างๆ โดยใช้ทฤษฎีจากที่กล่าวมาแล้ว มีขั้นตอนดังต่อไปนี้

## 3.1 ระบบโดยรวม




**รูปที่ 3.1** แสดงการทำงานโดยรวมของทั้งระบบ

### 3.1.1 การจับภาพการเคลื่อนไหว

คือ ส่วนที่ใช้กล้องเพื่อจับการเคลื่อนไหว โดยจะต้องมีการเตรียมการ ของตำแหน่งการวางกล้อง สภาพแวดล้อม ชุดของนักแสดง และ มาร์กเกอร์ ภาพที่ได้จะถูกส่งไปยังส่วนประมวลผลภาพต่อไป

### 3.1.2 การประมวลผลภาพเพื่อแยกมาร์กเกอร์

ส่วนนี้จะนำภาพที่ได้จากส่วนแรก มาทำการประมวลผลเพื่อตัดส่วนที่ไม่ต้องการออก ให้เหลือเพียงมาร์กเกอร์เท่านั้น รวมถึงการแยกแยะมาร์กเกอร์จุดต่างๆ และกำหนดพิกัดใน 2 มิติ

### 3.1.3 การคำนวณหาพิกัด 3 มิติ

เมื่อได้พิกัด 2 มิติ จากกล้องแต่ละตัวแล้ว ในส่วนนี้จะทำการนำพิกัด และค่าตัวแปรต่างๆ มาคำนวณ โดยใช้หลักการ Stereo Vision จึงได้พิกัดของมาร์กเกอร์แต่ละจุดเป็น 3 มิติ

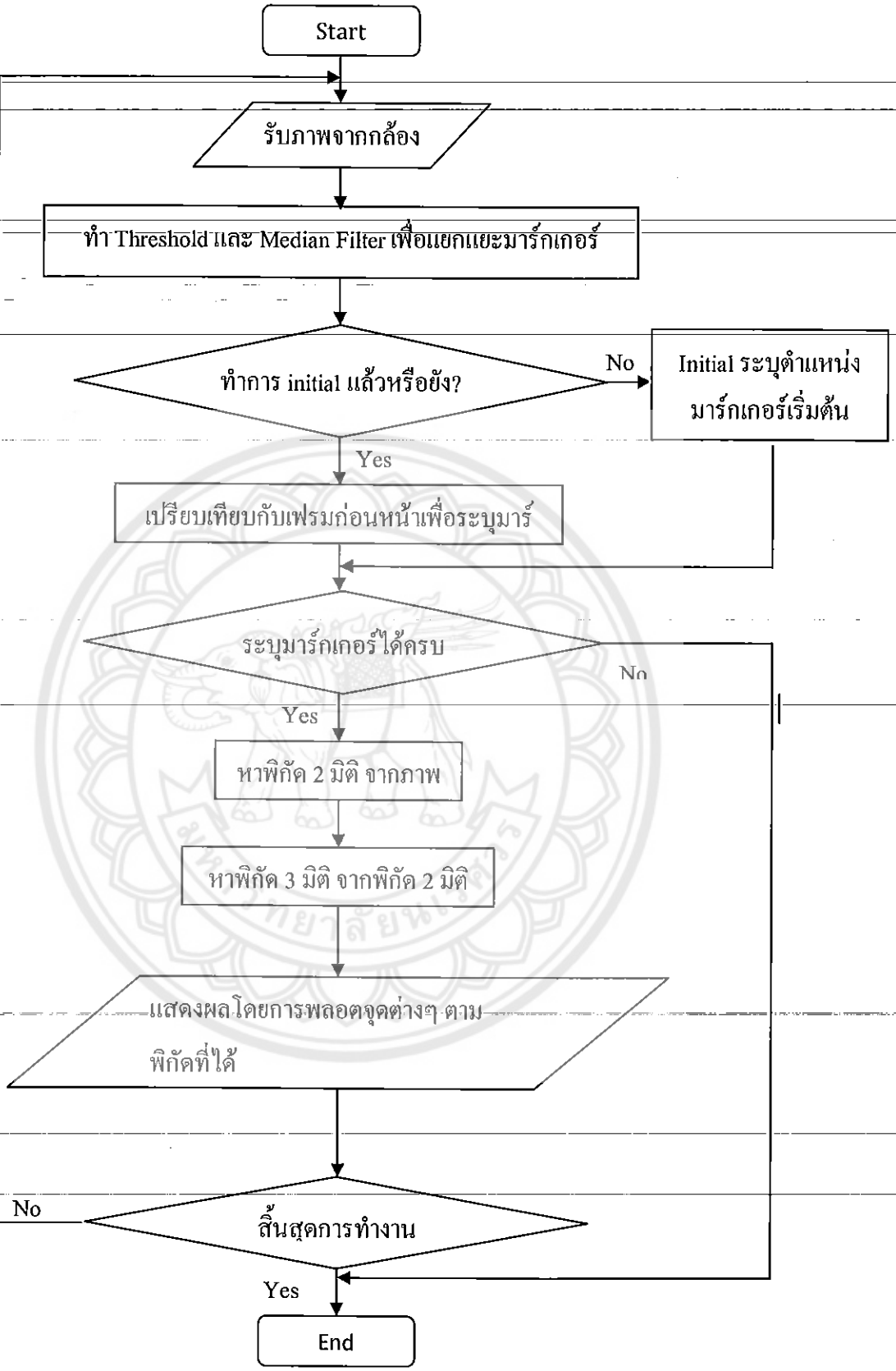
### 3.1.4 การวิเคราะห์การเคลื่อนไหว แสดงผล และ บันทึกการเคลื่อนไหว

ในส่วนนี้ จะเป็นการติดตามจุดมาร์กเกอร์ในภาพแต่ละเฟรม เพื่อวิเคราะห์การเคลื่อนที่ รวมถึงการสร้างจุดมาร์กเกอร์เสมือนขึ้นมา หากในส่วนของการจับภาพหรือการกำหนดพิกัดไม่สามารถหาจุดมาร์กเกอร์บางจุดได้ โดยจะวิเคราะห์จากพิกัดเดิมในเฟรมก่อนหน้า และมาร์กเกอร์ที่มีส่วนเชื่อมต่อกับมาร์กเกอร์ที่หายไป เมื่อได้พิกัดที่สมบูรณ์ของมาร์กเกอร์ทุกจุดแล้ว จะนำพิกัดแต่ละจุดนั้นมาเป็นข้อมูลเพื่อเขียนภาพ 3 มิติ โดยใช้ OpenGL

## 3.2 การออกแบบโปรแกรม

### 3.2.1 ขั้นตอนการทำงานของโปรแกรม

1. ใช้กล้องจับการเคลื่อนไหว
2. นำภาพที่ได้มาประมวลผล เพื่อตัดส่วนที่ไม่ต้องการออก เหลือไว้เฉพาะตำแหน่งของจุดมาร์กเกอร์
3. หาพิกัด 2 มิติ ของมาร์กเกอร์ทุกๆ จุด ในแต่ละมุมมอง
4. นำจุดมาร์กเกอร์แต่ละจุดที่ได้มาคำนวณหาพิกัดใน 3 มิติ
5. ติดตามการเคลื่อนที่ของมาร์กเกอร์ หากมาร์กเกอร์ใดหายไป ให้ทำการวิเคราะห์ตำแหน่งที่ควรจะเป็น โดยจะวิเคราะห์จากพิกัดเดิมในเฟรมก่อนหน้า และมาร์กเกอร์ที่มีส่วนเชื่อมต่อกับมาร์กเกอร์ที่หายไป
6. ส่งตำแหน่งพิกัดของมาร์กเกอร์แต่ละส่วนว่าอยู่ตรงไหนแล้วแสดงผลภาพ 3 มิติ
7. บันทึกข้อมูลพิกัดในเฟรมนั้นลงในไฟล์
8. กลับไปขั้นตอนการรับภาพ เพื่อนำมาคำนวณใหม่
9. เมื่อสิ้นสุดการทำงานจะได้ไฟล์บันทึกข้อมูลการเคลื่อนไหว ที่สามารถนำเข้าไปใช้ในโปรแกรมสร้างงาน 3 มิติ ได้ทันที




**รูปที่ 3.2** ผังการทำงานของระบบ

## 3.3 ออกแบบระบบจับภาพ

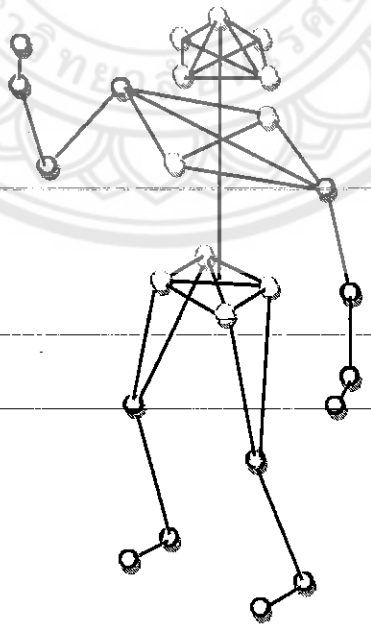
### 3.3.1 ประเภทของกล้องที่ใช้

ในโครงงานนี้ เลือกใช้กล้องประเภท กล้องวงจรปิด CCTV และมีเซ็นเซอร์แบบ CCD จำนวน 4 ตัว ซึ่งทำงานร่วมกับการ์ดจับภาพ (DVR card) ซึ่งติดตั้งในคอมพิวเตอร์ โดยการ์ดจับภาพมีหน้าที่ในการแปลงสัญญาณภาพ Analog จาก กล้องวิดีโอ ให้กลายเป็นภาพ Digital

รูปที่ 3.3 ตัวอย่างกล้องวงจรปิด และ การ์ดจับภาพ

### 3.3.2 การติดมาร์เกอร์ให้กับผู้แสดง

การติดมาร์กเกอร์ จะทำโดยติดมาร์กเกอร์ไว้ตามส่วนสำคัญของร่างกาย โดยเฉพาะจุดหมุน หรือ ข้อต่อกระดูกต่างๆ ซึ่งมาร์กเกอร์ทำขึ้นโดยใช้หลอดไฟกลมที่มีสีต่างๆ ซึ่งการใช้หลอดไฟสีต่างๆ กัน ทำให้ช่วยแยกแยะ มาร์กเกอร์ได้ง่ายขึ้น



**รูปที่ 3.4** การติดมาร์กเกอร์ตามจุดหมุนหรือข้อต่อกระดูกในร่างกาย

1. มาร์กเกอร์ที่แทนการเคลื่อนไหวของศีรษะ จะถูกติดไว้ที่หมวกด้านหน้า และด้านหลัง

2. มาร์กเกอร์สำหรับจับการเคลื่อที่ของลำตัวส่วนบนจะถูกติดไว้ที่ หัวไหล่ทั้งสองข้าง และด้านหลัง

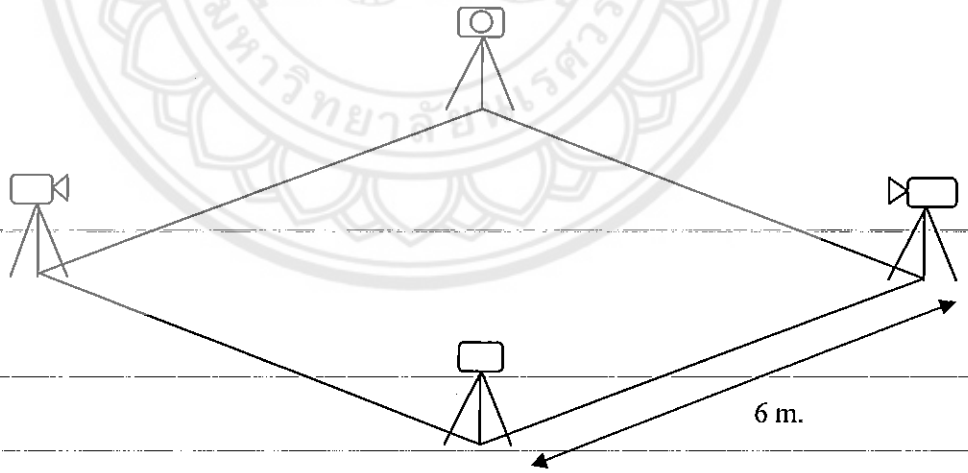
3. มาร์กเกอร์สำหรับจับการเคลื่อนไหวของแขนและมือทั้งสองข้าง จะถูกติดไว้ที่ ข้อศอก ข้อมือ 2 อัน และ ที่ปลายมือ

4. มาร์กเกอร์สำหรับจับการเคลื่อนที่ของส่วนสะโพก จะถูกติดที่เอว ด้านหน้า 2 อัน และด้านหลัง 1 อัน

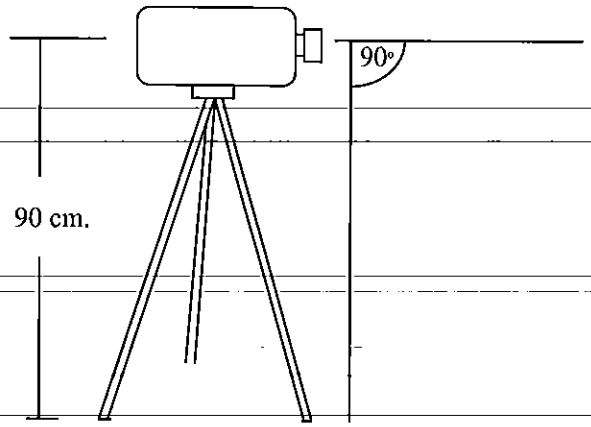
5. มาร์กเกอร์สำหรับจับการเคลื่อนที่ของส่วนขาและเท้าทั้งสองข้าง จะถูกติดที่ หัวเข่า ส้นเท้า และที่ปลายเท้า 2 อัน

**3.3.3 ตำแหน่งการวางกล้อง**

การติดตั้งกล้องนั้นควรจะเลือกใช้กล้องที่มีชนิดเดียวกัน รุ่นเดียวกัน เพื่อให้มีความสมดุลของค่าพารามิเตอร์ในกล้องให้ใกล้เคียงกันที่สุด และการจัดตำแหน่งของกล้องนั้นวางโดยให้ฉากรับภาพของกล้องสามารถรับรายละเอียดภาพได้มากที่สุด เพื่อนำภาพที่ได้ไปคำนวณหาตำแหน่ง 3 มิติ ในขั้นตอนของ Stereo Vision นั้นเอง ซึ่งในโครงงานนี้ใช้รูปแบบการวางกล้องแบบหันหน้าชนกันทั้งสี่มุม ได้แก่ ด้านหน้า ด้านหลัง ด้านซ้ายและด้านขวา โดยทั้งสี่ด้านวางให้ตำแหน่งห่างกัน 6 เมตร ความสูงของกล้อง 90 เซนติเมตร โดยมุมมองของกล้องขนานกับพื้น ดังรูปที่ 3.5 และ 3.6



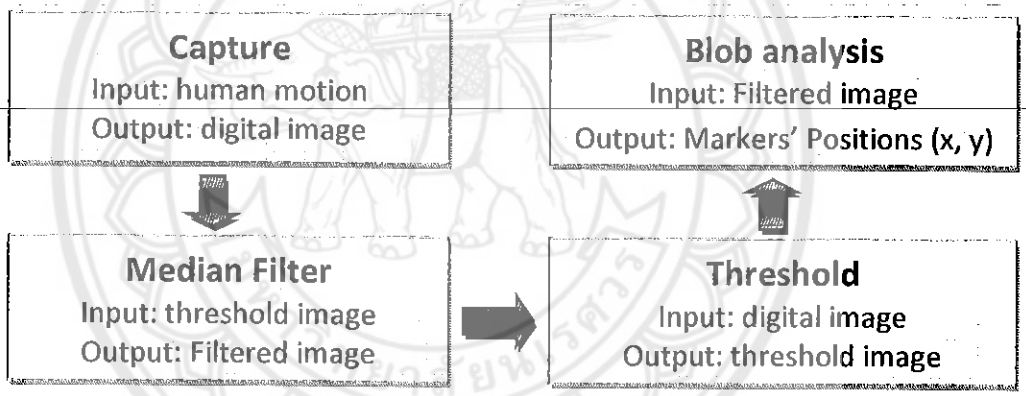

**รูปที่ 3.5** แสดงการวางตำแหน่งของกล้อง




**รูปที่ 3.6** แสดงส่วนสูงและมุมกล้อง

## 3.4 การประมวลผลภาพ

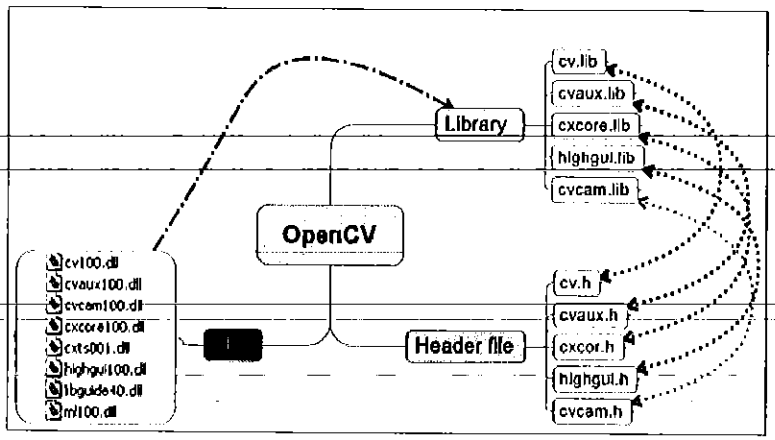
ส่วนนี้จะนำภาพที่ได้จากส่วนแรก มาทำการประมวลผลเพื่อตัดส่วนที่ไม่ต้องการออก ให้เหลือเพียงมาร์กเกอร์เท่านั้น รวมถึงการแยกแยะมาร์กเกอร์จุดต่างๆ และกำหนดพิกัดใน 2 มิติ




**รูปที่ 3.7** ขั้นตอนการทำงานส่วนประมวลผลภาพ

### 3.4.1 การรับค่าจากกล้องและการประมวลผล

โปรแกรมส่วนติดต่อกับฮาร์ดแวร์ในโครงงานนี้จะใช้โปรแกรม OpenCV ใน Microsoft Visual C++ โดย OpenCV เป็นไลบรารี่สำหรับใช้ในการประมวลผลภาพ (Image Processing) สามารถประมวลผลได้ทั้งภาพนิ่งและภาพเคลื่อนไหวจากทั้งกล้องวิดีโอ และไฟล์วิดีโอ โดยฟังก์ชั่นต่างๆ ของ OpenCV จะสามารถเรียกใช้งานได้จะต้องมีการเรียกไฟล์ส่วนหัว (Head file) และลิงค์ (Link) ไลบรารี่ต่างๆ รวมถึง DLL (Dynamic Link Library) ดังภาพ




**รูปที่ 3.8** ความสัมพันธ์ไรบรารี่ของ OpenCV

### 3.4.2 การทำ Threshold

การทำ Threshold ในโครงงานนี้จะทำ Threshold โดยจะเลือกเอาส่วนใดๆ ของภาพที่มีสีขาว (คือส่วนที่เป็นมาร์กเกอร์) ในโครงงานนี้จะมีการทำให้ฉากหรือสภาพแวดล้อมมีสีอื่นมาปะปนน้อยที่สุด คือ พยายามทำให้ฉากหลังเป็นสีดำ เนื่องจากจะทำให้การแยกแยะมาร์กเกอร์เป็นไปได้ง่ายขึ้น และมาร์กเกอร์ที่ใช้ติดบนตัวนักแสดงจะทำมาจากหลอดไฟ ดังนั้น มาร์กเกอร์ที่ได้ในภาพจะเป็นลักษณะของกลุ่มสีขาว ซึ่งการทำ Threshold จะทำการ ปัดค่าของกลุ่มสีเหล่านั้นให้ชัดเจน เช่น ภาพมาร์กเกอร์ จะได้ภาพกลุ่มสีที่มีสีขาว เทาหลายๆ เฉด บริเวณที่เป็นสีเทาเหล่านี้ จะมีค่าสีตั้งแต่ 0 ถึง 255 ถ้าสว่างมาก จะได้ค่ามาก ดังนั่นส่วนที่เป็นมาร์เกอร์จะมีค่ามาก เนื่องจากมีแสงสว่างมาก ซึ่งจะกำหนดค่าๆ หนึ่งสำหรับการทำ Theshold คือ ถ้าพิกเซลใดต่ำกว่าค่านี้จะถูกเปลี่ยนเป็นสีดำ หากมากกว่าจะเป็นสีขาว

### 3.4.3 การทำ Median Filter

หลังจากทำ Threshold แล้วจะได้จุดสีต่างๆ ที่เป็นมาร์กเกอร์หลายๆ อัน ซึ่งในบางตำแหน่งอาจจะมีจุดสีที่ไม่ต้องการเนื่องมาจากเป็นจุดที่ไม่ได้เกิดจากมาร์กเกอร์ จึงต้องกำจัดจุดเหล่านี้ก่อนโดยการทำ Median Filter ภาพที่เกิดจากการทำ Threshold และ Median Filter จะเป็นภาพที่แยกมาร์กเกอร์ออกจากพื้นหลังอย่างชัดเจน

### 3.4.4 Blob analysis

เป็นวิธีการนำมาร์กเกอร์ที่หาได้จากการทำ Threshold และ Median Filter มาระบุตำแหน่งพิกัดจุด x และ y ให้เหลือเพียงจุดเดียว โดยหลังจากการทำ Threshold และ Median Filter แล้วจะได้มาร์กเกอร์ในลักษณะของกลุ่มสี ซึ่งจะหาค่าพิกัดกลางของกลุ่มสีนี้โดย เฉลี่ยค่าพิกัด x, y ของทุกพิกเซลภายในกลุ่มสี แล้วนำค่ากลางที่ได้มาเป็นพิกัดจุดบนภาพของมาร์กเกอร์จุดนั้น

**ขั้นตอนการทำ Blob analysis**

1. ระบุหมายเลขให้ทุกพิกเซลให้เป็น -1 ทุกพิกเซล

2. เริ่มทำทีละพิกเซลหากพิกเซลใดแป็นสีขาว ให้ดูพิกเซลอื่นที่อยู่ติดกัน ว่ามีสีขาวหรือไม่ ถ้ามีพิกเซลที่ติดกันเป็นสีขาว จะระบุหมายเลข ให้เหมือนกับพิกเซลที่อยู่ติดกันนั้น แต่หากไม่มีจะถูกระบุเป็นหมายเลขใหม่ โดยหากมีการเพิ่มกลุ่มสีขาวหมายเลขนี้จะเพิ่มไปเรื่อยๆ

3. ในกรณีที่พิกเซลที่ติดกันมีมากกว่าสองพิกเซล และมีหมายเลขต่างกัน จะทำการเก็บข้อมูลการรวม เข้าสู่สแต็ก (Stack) เพื่อจะรวมกลุ่มพิกเซลสองกลุ่มนี้ทีหลัง แล้วเลือกระบุหมายเลขพิกเซล ตามพิกเซลที่มีหมายเลขมากกว่า

4. ทำเช่นนี้ต่อไป จนครบทุกพิกเซล จากนั้นดึงข้อมูลจาก สแต็กเพื่อทำการรวมกลุ่มเลขที่ต่างกันแต่อยู่ติดกัน ให้เป็นเลขเดียวกัน

| | | | | | |
|---|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 | -1 |
| -1 | 0 | -1 | 1 | -1 | 2 |
| 0 | 0 | -1 | 1 | 2 | 2 |
| -1 | -1 | -1 | -1 | -1 | -1 |

รูปที่ 3.9 ตัวอย่างการทำ Blob analysis

**ตัวอย่างการทำ Blob analysis** จากรูปที่ 3.9

1. ระบุหมายเลขให้ทุกพิกเซลให้เป็น -1 ทุกพิกเซล

2. ในแถวแรกไม่มีพิกเซลสีขาวจึงไม่มีการเปลี่ยนแปลง

3. ในแถวที่ 2 พิกเซลที่ 2 เป็นสีขาวและพิกเซลก่อนหน้า ไม่มีสีขาว พิกเซลนี้จึงถูกระบุหมายเลขเป็นเลข 0

4. ในแถวที่ 2 พิกเซลที่ 4 เป็นสีขาวและพิกเซลก่อนหน้าที่อยู่ติดกันไม่มีสีขาว จึงถูกระบุเลขใหม่ คือเลข 1 เนื่องจากเลข 0 ถูกใช้ไปแล้ว

5. ในแถวที่ 2 พิกเซลที่ 6 เป็นสีขาวและพิกเซลก่อนหน้าที่อยู่ติดกันไม่มีสีขาว จึงถูกระบุเลขใหม่ คือเลข 2

6. ในแถวที่ 3 พิกเซลที่ 1 มีพิกเซลก่อนหน้าที่อยู่ติดกัน เป็นสีขาว คือ พิกเซล 2 แถวที่ 2 ถูกระบุเป็นเลข 0 ดังนั้นพิกเซลนี้จะถูกระบุเลข 0 เช่นเดียวกัน

7. ในแถวที่ 3 พิกเซลที่ 2 มีพิกเซลก่อนหน้าที่อยู่ติดกัน เป็นสีขาว คือ พิกเซล 2 แถวที่ 2 และพิกเซลที่ 1 แถวที่ 3 ถูกระบุเป็นเลข 0 ดังนั้นพิกเซลนี้จะถูกระบุเลข 0 เช่นเดียวกัน

8. ในแถวที่ 3 พิกเซลที่ 4 มีพิกเซลก่อนหน้าที่อยู่ติดกัน เป็นสีขาว คือ พิกเซล 4 แถวที่ 2 ถูกระบุเป็นเลข 1 ดังนั้นพิกเซลนี้จะถูกระบุเลข 1 เช่นเดียวกัน

9. ในแถวที่ 3 พิกเซลที่ 5 มีพิกเซลก่อนหน้าที่อยู่ติดกัน เป็นสีขาว คือ พิกเซล 4 แถวที่ 2 ถูกระบุเป็นเลข 0 และพิกเซล 6 แถวที่ 2 ถูกระบุเลข 2 ดังนั้นพิกเซลนี้จะถูกระบุเลขที่มากกว่าคือ เลข 2 พร้อมเก็บข้อมูลการรวมเลข ของเลข 2 และ 1 ลงสแต็ก

10. ในแถวที่ 3 พิกเซลที่ 6 มีพิกเซลก่อนหน้าที่อยู่ติดกัน เป็นสีขาว คือ พิกเซล 6 แถวที่ 2 และพิกเซล 4 แถวที่ 3 ถูกระบุเป็นเลข 2 ดังนั้นพิกเซลนี้จะถูกระบุเลข 2 เช่นเดียวกัน

11. เมื่อทำจนครบทุกพิกเซลจนหมด ให้รวมเลขที่เป็บไว้ในแสต็คเป็นกลุ่มเดียวกัน ในที่นี้ คือ กลุ่มเลข 2 และ 1

12. จะได้ Blob ที่ระบุเลขโดยสมบูรณ์ ดังรูปที่ 3.10

| | | | | | |
|---|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 | -1 |
| -1 | 0 | -1 | 1 | -1 | 1 |
| 0 | 0 | -1 | 1 | 1 | 1 |
| -1 | -1 | -1 | -1 | -1 | -1 |

รูปที่ 3.10 ตัวอย่างการทำ Blob analysis

## 3.5 การคำนวณหาจุดพิกัด

เมื่อได้พิกัด 2 มิติ จากกล้องแต่ละตัวแล้ว ในส่วนนี้จะทำการนำพิกัด และค่าตัวแปรต่างๆ มาคำนวณ โดยใช้หลักการ Stereo Vision จึงได้พิกัดของมาร์กเกอร์แต่ละจุดเป็น 3 มิติ โดยใช้อัลกอริทึม ดังนี้

ให้ p1, p2,p3, p4, pa และ pb เป็นจุดใน 3 มิติ

```
p13.x = p1.x - p3.x;
p13.y = p1.y - p3.y;
p13.z = p1.z - p3.z;
p43.x = p4.x - p3.x;
p43.y = p4.y - p3.y;
p43.z = p4.z - p3.z;
if (ABS(p43.x) < EPS && ABS(p43.y) < EPS && ABS(p43.z) < EPS)
        return(FALSE);
```

```
p21.x = p2.x - p1.x;
p21.y = p2.y - p1.y;
p21.z = p2.z - p1.z;
if (ABS(p21.x) < EPS && ABS(p21.y) < EPS && ABS(p21.z) < EPS)
        return(FALSE);

d1343 = p13.x * p43.x + p13.y * p43.y + p13.z * p43.z;
d4321 = p43.x * p21.x + p43.y * p21.y + p43.z * p21.z;
d1321 = p13.x * p21.x + p13.y * p21.y + p13.z * p21.z;
d4343 = p43.x * p43.x + p43.y * p43.y + p43.z * p43.z;
d2121 = p21.x * p21.x + p21.y * p21.y + p21.z * p21.z;

denom = d2121 * d4343 - d4321 * d4321;
if (ABS(denom) < EPS)
return(FALSE);
numer = d1343 * d4321 - d1321 * d4343;

*mua = numer / denom;
*mub = (d1343 + d4321 * (*mua)) / d4343;

pa->x = p1.x + *mua * p21.x;
pa->y = p1.y + *mua * p21.y;
pa->z = p1.z + *mua * p21.z;
pb->x = p3.x + *mub * p43.x;
pb->y = p3.y + *mub * p43.y;
pb->z = p3.z + *mub * p43.z;
        return(TRUE);
```

จะได้จุด Pa และ Pb ซึ่งจะนำจุดสองจุดนี้มาหาระยะกลาง เพื่อใช้เป็นพิกัดของมาร์กเกอร์

## 3.6 การกำหนดและติดตามมาร์กเกอร์

### 3.6.1 การกำหนดค่าเริ่มต้นให้มาร์กเกอร์

การกำหนดตำแหน่งเริ่มต้นให้กำมาร์กเกอร์แต่ละอัน ทำโดย ให้ผู้แสดงสวมชุดมาร์กเกอร์ที่กำหนด แล้วทำท่าเริ่มต้น คือ ท่ายืนกางแขน จากนั้น บันทึกภาพเริ่มต้นนี้เก็บไว้ ทั้ง 4 มุมมอง จากนั้น นำภาพที่ได้มาผ่านกระบวนการหาตำแหน่งมาร์กเกอร์ในรูป 2 มิติ แล้วแปลงเป็น 3 มิติ ด้วยขั้นตอนที่กล่าวมาแล้ว ซึ่งจะได้ผลลัพธ์เป็น ชุดของมาร์กเกอร์ที่ถูกกำหนดลำดับและตำแหน่งไว้แล้วทุกจุด หลังจากนั้น ในขั้นตอนการจับการเคลื่อนไว้ เมื่อเริ่มต้นโปรแกรม ให้ผู้แสดงเริ่มทำการเคลื่อนไหว โดยเริ่มต้นจากท่ากางแขนค้างไว้ โปรแกรมจะทำการหาตำแหน่งมาร์กเกอร์ที่อยู่ใกล้ที่สุด โดยเทียบจากกลุ่มตำแหน่งมาร์กเกอร์เริ่มต้นที่สร้างไว้แล้ว ก็จะสามารถระบุได้ว่ามาร์กเกอร์ลำดับที่เท่าไร อยู่ในตำแหน่งใด

### 3.6.2 การติดตามมาร์กเกอร์

การติดตามมาร์กเกอร์แต่ละตัว จะทำในขั้นตอนหลังจากได้ตำแหน่งใน 2 มิติ โดยใช้วิธีการเทียบจากตำแน่งของมาร์กเกอร์แต่ละตัวในเฟรมก่อนหน้า คือ ในแต่ละเฟรมมีระยะเวลาซึ่งห่างกันน้อยมาก ดังนั้น ตำแหน่งของมาร์กเกอร์ที่เคลื่อนที่ไปในแต่ละเฟรมนั้น จะมีระยะทางไม่ไกลกันมาก ซึ่งสามารถกล่าวได้ว่า มาร์กเกอร์ในเฟรมก่อนหน้าซึ่งมีตำแหน่งใกล้ที่สุด และมีสีเดียวกันกับมาร์กเกอร์ในเฟรมปัจจุบัน เป็นมาร์กเกอร์เดียวกัน ก็จะสามารถระบุ ลำดับของมาร์กเกอร์ได้ แต่วิธีนี้ก็มีโอกาสผิดพลาดค่อนข้างมาก คือ หากมีมาร์กเกอร์สีเดียวกัน เคลื่อนที่มาในตำแหน่งใกล้กัน จะทำให้โปรแกรมระบุลำดับผิดได้ ดังนั้น จึงต้องติดตั้งมาร์กเกอร์ที่มีสีเดียวกันไว้ห่างกัน หรือทำให้มีโอกาสเคลื่อนที่มาใกล้กันได้น้อยที่สุด

## 3.7 แสดงการเคลื่อนไหวให้เป็น 3 มิติ

### 3.7.1 เขียนภาพ 3 มิติจากตำแหน่งของมาร์กเกอร์

การแสดงตำแหน่งของมาร์กเกอร์ ใช้ OpenGL ในการแสดงผล โดยการ สร้างภาพทรงกลมมีสี และตำแหน่งตาม ลักษณะของมาร์กเกอร์ทุกๆตัว และให้เปลี่ยนตำแหน่งไปตามมาร์กเกอร์ในแต่ละเฟรม ก็จะได้ภาพ 3 มิติ ของการเคลื่อนที่ของมาร์กเกอร์

# บทที่ 4

# ผลการทดลอง

ในบทนี้จะกล่าวถึงรายละเอียดต่างๆ ของการติดตั้งมาร์กเกอร์ไว้ตามจุดต่างๆ ของร่างกาย การจับภาพเคลื่อนไหวตัวละคร และผลการทำงานของโปรแกรม ซึ่งสามารถแยกการทำงานหลักๆ ของแต่ละขั้นตอนออกเป็นดังนี้

-การจับภาพเคลื่อนไหวตัวละคร ใช้กล้องทั้งหมด 4 ตัววางหันหน้าเข้าหากันโดยการกำหนดพื้นที่แสดงตามความกว้างของระยะโฟกัสของกล้อง ใช้หลอดไฟเป็นมาร์กเกอร์เพื่อบอกตำแหน่งของจุดต่างๆ ของร่างกาย

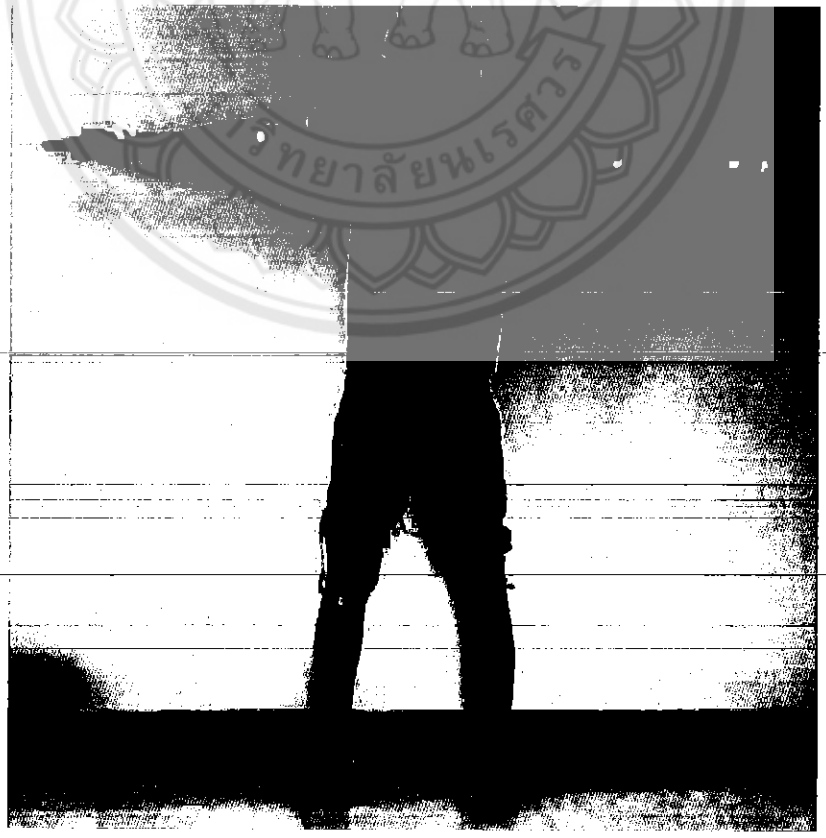
- การทำงานของโปรแกรม ในแต่ละการบันทึกวีดิโอ 1 ครั้ง จะได้ไฟล์วีดิโอทั้งหมด 4 ไฟล์ จะนำไฟล์แต่ละไฟล์เข้าสู่กระบวนการประมวลผลภาพเริ่มจากการจับภาพของแต่ละเฟรมการเคลื่อนไหว แล้วนำภาพที่ได้ไปทำ Threshold และ Median Filter เพื่อแยกมาร์กเกอร์ออกจากพื้นหลัง การ blob เพื่อหาพิกัดจุดแกน x และแกน y เมื่อได้จุดแกน x, y ของแต่ละไฟล์แล้ว จะนำไฟล์วีดิโอทั้งหมดมาคำนวนหาพิกัด 3 มิติ โดยใช้หลักการ Stereo Vision จะได้พิกัดเป็น x, y, z บันทึกพิกัดแล้ววนกลับไปเริ่มการทำงานที่เฟรมถัดไป

## 4.1 การจับภาพเคลื่อนไหว

การจับภาพจะใช้กล้องวิดีโอจำนวน 4 ตัว โดยวางตำแหน่งไว้ แบบหันมุมกล้องเข้าหาพื้นที่ตรงกลางทั้ง 4 ตัว แล้วต่อกล้องเข้ากับการ์ดจับภาพซึ่งติดตั้งไว้กับคอมพิวเตอร์

ในการทดสอบ จะให้ผู้แสดง เคลื่อนไหวในท่าทางต่างๆ โดยโครงงานนี้แสดงทั้งหมด 4 แบบ คือ แกว่งขา กางแขนบิดลำตัว ยกแขนขึ้นลง และหมุนคอ โดยแต่ละแบบจะทำซ้ำกัน 3 ครั้ง ชุดที่ใช้เป็นชุดที่ทำขึ้นเอง ใช้หลอดไฟสีแดงและเหลืองเป็นมาร์กเกอร์ ติดตามจุดต่างๆ ตามรูปที่ 4.1, 4.2, 4.3 และ 4.4

**รูปที่ 4.1** แสดงด้านหน้าการติดมาร์กเกอร์ตามจุดหมุนหรือข้อต่อกระดูกในร่างกาย



**รูปที่ 4.2** แสดงด้านหลังการติดมาร์กเกอร์ตามจุดหมุนหรือข้อต่อกระดูกในร่างกาย

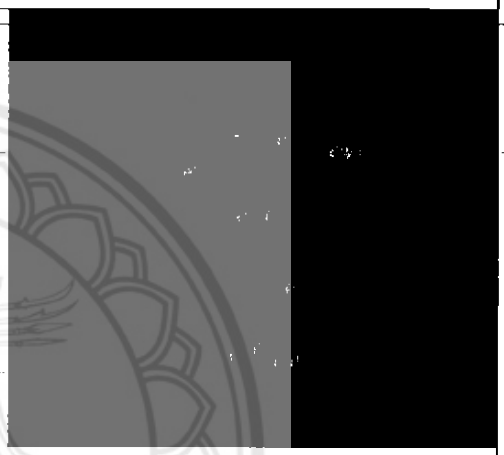
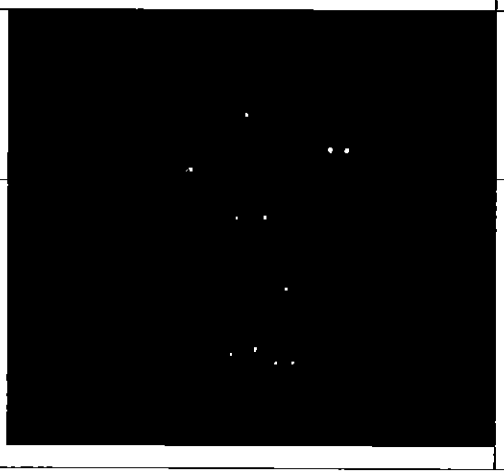
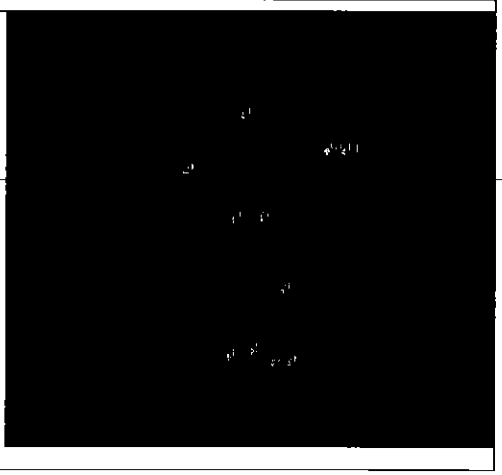
**รูปที่ 4.3** แสดงด้านข้างการติดมาร์กเกอร์ตามจุดหมุนหรือข้อต่อกระดูกในร่างกาย

**รูปที่ 4.4** แสดงด้านหน้าส่วนบนการติดมาร์กเกอร์ตามจุดหมุนหรือข้อต่อกระดูกในร่างกาย

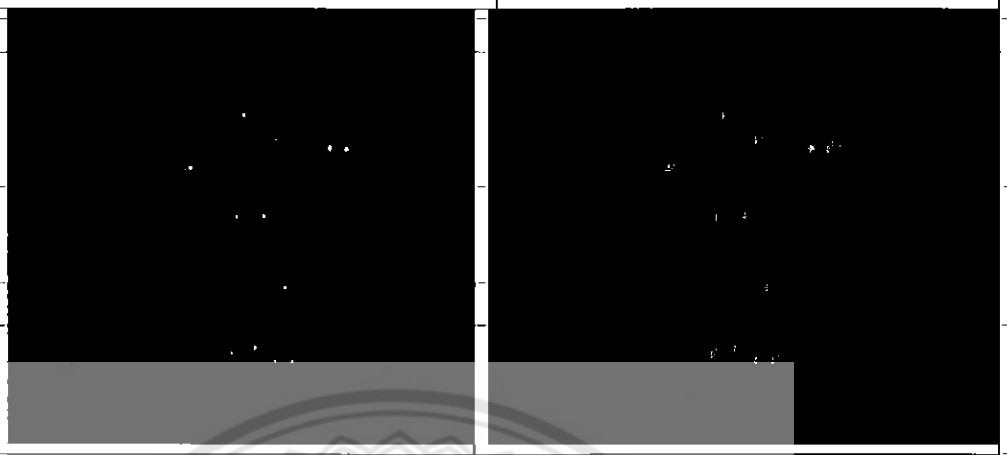
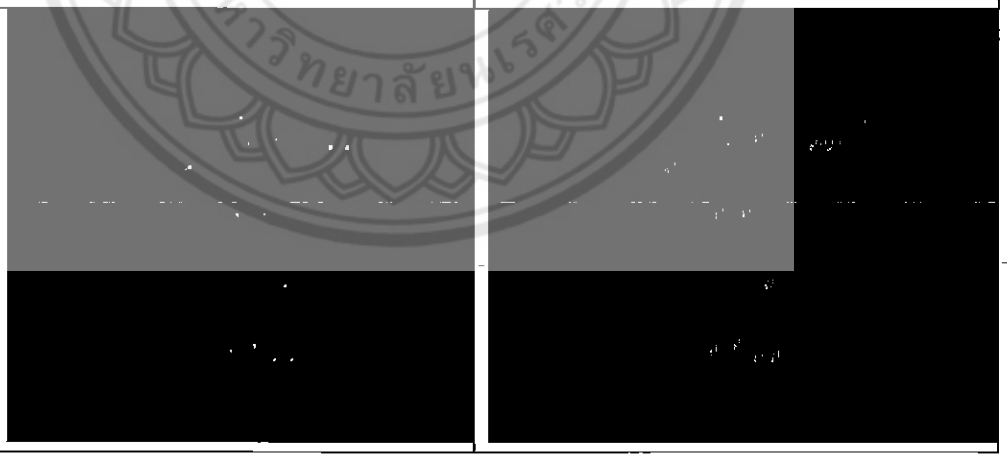
## 4.2 ผลการประมวลผลภาพ

หลังจากได้ภาพจากกล้องวิดีโอ จะเป็นส่วนของการประมาวลผลภาพ ซึ่งเป็นส่วนของการทำ Thresholding และหา Blob การทดลองทำโดยนำไฟล์วิดีโอที่ได้จากการจับภาพ มาเข้าสู่กระบวนการประมวลผลภาพ ซึ่งได้ผลดังนี้

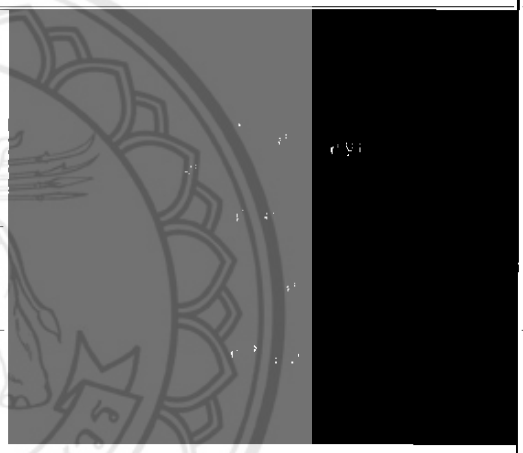
**ตารางที่ 4.1** แสดงการ Blob ที่ละเฟรม

| เฟรมที่ | ภาพที่รับจากกล้อง | ภาพหลังจากหา Blob |
|---|---|---|
| 1 | |  |
| 2 | | |
| 3 |  |  |

**ตารางที่ 4.1** แสดงการ Blob ทีละเฟรม (ต่อ)

| เฟรมที่ | ภาพที่รับจากกล้อง | ภาพหลังจากหา Blob |
|---|---|---|
| 4 |  | |
| 5 | | |
| 6 |  | |

**ตารางที่ 4.1** แสดงการ Blob ทีละเฟรม (ต่อ)

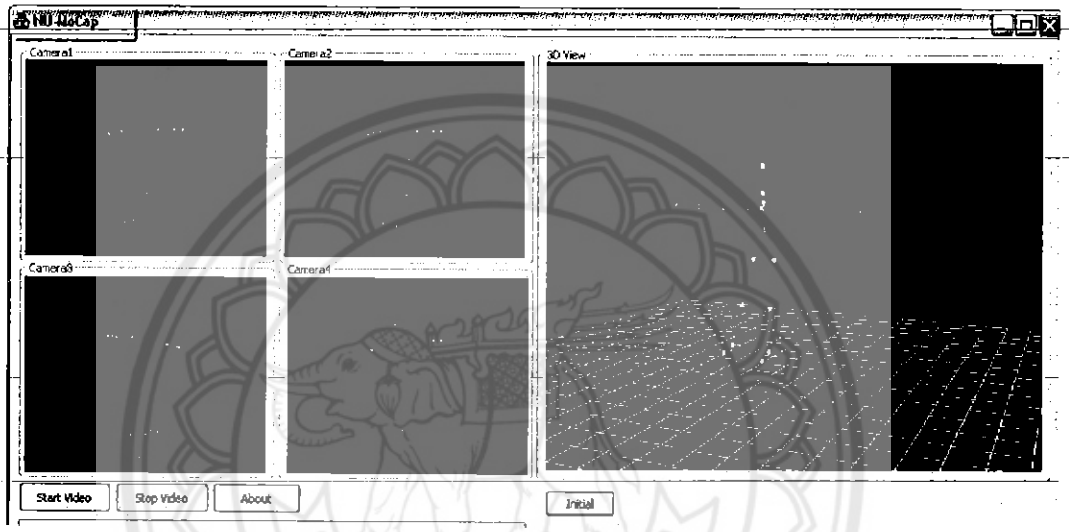
| เฟรมที่ | ภาพที่รับจากกล้อง | ภาพหลังจากหา Blob |
|---|---|---|
| 7 | | |
| 8 | |  |
| 9 | | |

จากผลการทดลองพบว่าหากแสดงท่าที่ไม่ซับซ้อน ซึ่งทำให้โอกาสที่มาร์กเกอร์มาทับกันมีน้อยลง จะทำให้การติดตามการเคลื่อนที่เป็นไปอย่างถูกต้อง แต่ยังมีบางเฟรมซึ่งมีปัญหาการตรวจไม่พบมาร์กเกอร์ หรือระบุมาร์กเกอร์ผิดลำดับ

## 4.3 ผลจากการแปลงค่าพิกัด 2 มิติ เป็น 3 มิติ

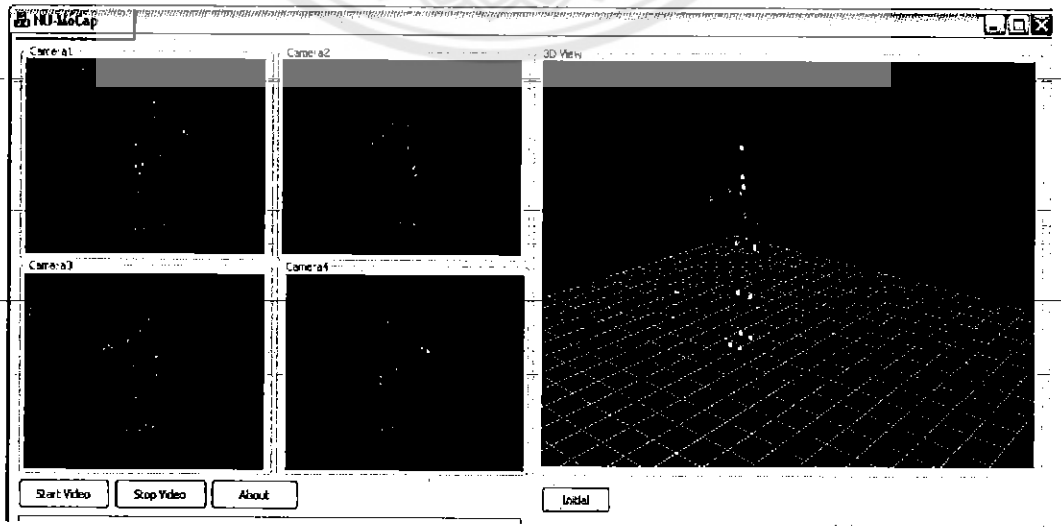
ส่วนนี้จะแสดงผลที่ได้ เมื่อทำการแปลงค่าพิกัด 2 มิติ จากทั้งสี่มุมมมอง ให้เป็นภาพ 3 มิติ โดยใช้การแสดงผลด้วย OpenGL

1. ทำการทดสอบแสดงท่ากางแขนแล้วยกขาซ้าย ขวา สลับกัน ได้ผลดังรูปที่ 4.5



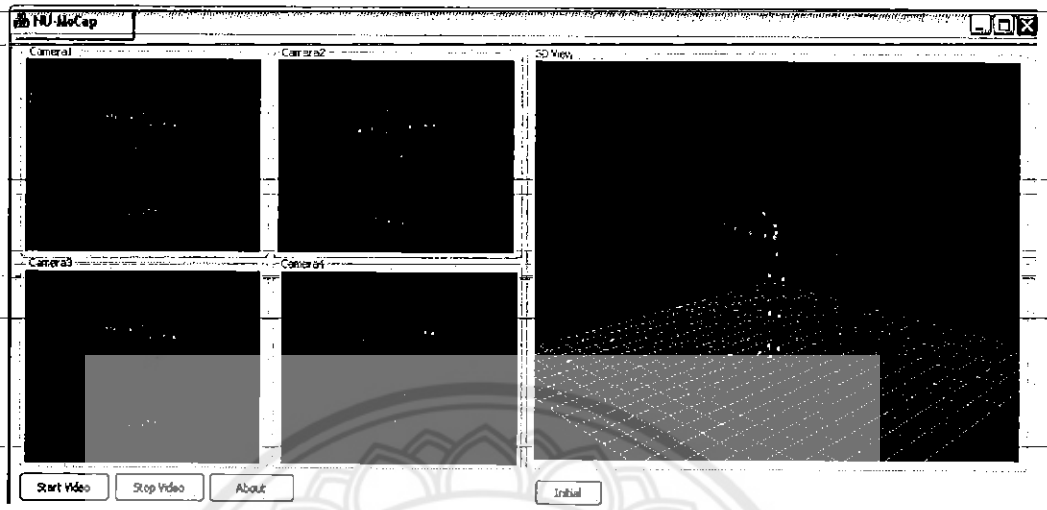

**รูปที่ 4.5** แสดงภาพ 3 มิติ ในท่ากางแขน และยกขา

2. ทำการทดสอบแสดงท่ายกแขนซ้าย ขวา สลับกัน ได้ผลดังรูปที่ 4.6



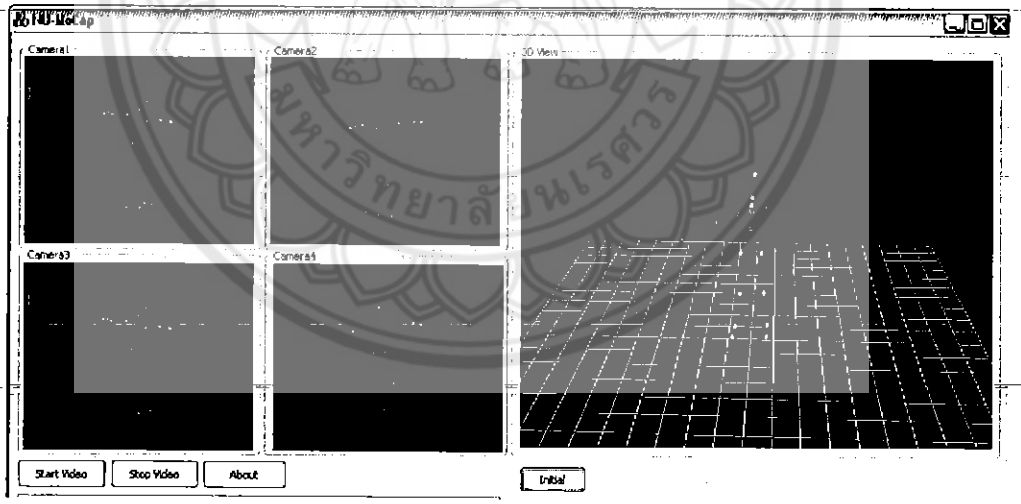

**รูปที่ 4.6** แสดงภาพ 3 มิติ ในท่ากำลังยกแขน

3. ทำการทดสอบแสดงท่ากางแขนแล้วก้ม เงย ศรีษะสลับกัน ได้ผลดังรูปที่ 4.7



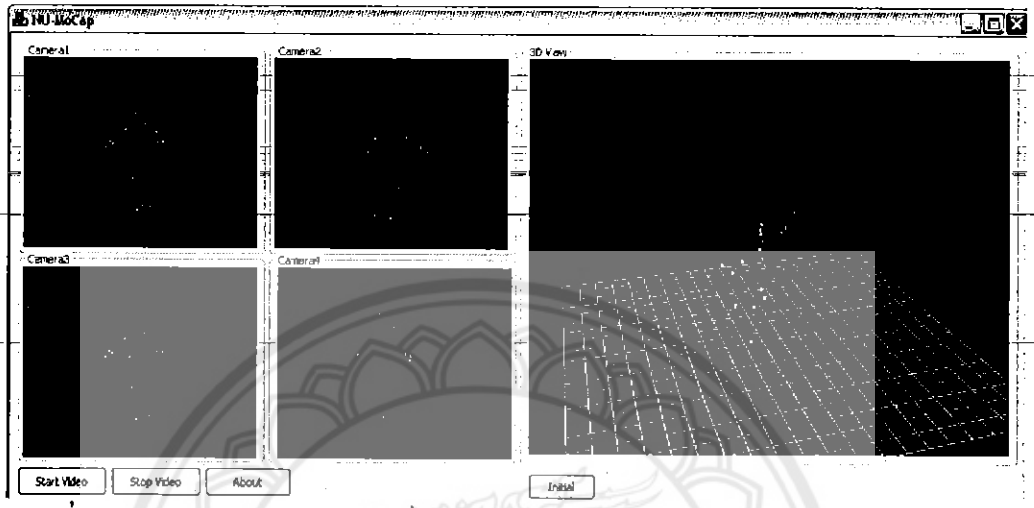

**รูปที่ 4.7** แสดงภาพ 3 มิติ ในท่ากำลังก้มศรีษะ

4. การทดสอบแสดงท่ากางแขนแล้วแล้วบิดลำตัวทางซาย-ขวา ได้ผลดังรูปที่ 4.8




**รูปที่ 4.8** แสดงภาพ 3 มิติ ในท่ากางแขนบิดลำตัว

ในการทดสอบในแต่ละท่ามีหลายครั้ง ที่แสดงผลผิดพลาด เนื่องจากสาเหตุต่างๆ เช่น การไม่พบมาร์กเกอร์ หรือ การระบุมาร์กเกอร์ผิดตำแหน่ง เป็นต้น ตัวอย่างกรณีที่แสดงผลผิดพลาด ดังรูปที่ 4.9




**รูปที่ 4.9** แสดงภาพ 3 มิติ ในกรณีที่ระบุมาร์กเกอร์ผิดพลาด ทำให้การแสดงผลไม่ถูกต้อง

จากภาพการทดลองแสดงให้เห็นว่า การแสดงผลในรูปแบบ 3 มิติ มีความใกล้เคียงกับท่าทางจริงของผู้แสดง แต่ในท่าที่มีความซับซ้อนหรือการเคลื่อนที่รวดเร็วจะทำให้การระบุมาร์กเกอร์ผิดพลาดซึ่งทำให้ไม่สามารถแสดงผลใน 3 มิติ ได้

# บทที่ 5

# บทสรุป

## 5.1 หน้าที่การทำงานของโปรแกรม

โปรแกรมที่พัฒนาในโครงงานนี้มีหน้าที่ในการรับค่าไฟล์วิดิโอทั้งสี่ไฟล์แล้วคำนวนหาพิกัดจุด x, y, z ของแต่ละเฟรมแล้วบันทึกออกมาในรูปแบบของไฟล์ .BVH เพื่อนำไปใช้ในการเรนเดอร์กับโปรแกรมสร้างภาพ3 มิติ เช่น 3Ds Max, Blender, Maya, Poser, Second Life เป็นต้น

## 5.2 วิเคราะห์ผลการทดลอง

จากผลการทดลอง ระบบการตรวจจับการเคลื่อนไหวของมนุษย์ โดยใช้การประมวลผลภาพ และจำลองการเคลื่อนไหวด้วยภาพ 3 มิติ เมื่อทำการทดสอบระบบ โดยให้ผู้แสดง ทำท่าทางในลักษณะต่าง 4 ท่า พบว่า ในแต่ละการทดสอบ มีปัญหาการไม่พบมาร์กเกอร์และการระบุมาร์กเกอร์มีการสลับตำแหน่ง ซึ่งทำให้ได้ผลลัพธ์ออกมา มีมาร์กเกอร์บางส่วนเกิดการคลาดเคลื่อน แต่ภาพการเคลื่อนที่ได้ มีความใกล้เคียงที่ผู้แสดงทำการแสดง โดยมี 1 ครั้งที่โปรแกรมไม่สามารถทำงานต่อไปได้ เนื่องจากจำนวนการไม่พบมาร์กเกอร์มากเกินไป

## 5.3 ปัญหาและแนวทางการแก้ไข

1. การตรวจจับมาร์กเกอร์บางครั้ง ไม่มีกล้องตัวใด หรือมีกล้องน้อยกว่าสองตัว ที่สามารถตรวจจับมาร์กเกอร์ได้ ทำให้ไม่สามารถหาพิกัด 3 มิติได้ แนวทางการแก้ปัญหา คือ เพิ่มจำนวนกล้อง เพื่อทำให้โอกาสที่ตรวจจับมาร์กเกอร์ไม่พบลดลง

2. เฟรมเรทของการ์ดจับภาพต่ำ ทำให้การเคลื่อนที่ไม่ต่อเนื่อง มีลักษณะภาพกระตุก โดยเฉพาะลักษณะการเคลื่อนที่ที่มีความเร็วสูง ทำให้การจับการเคลื่อนที่ของมาร์กเกอร์คลาดเคลื่อน สามารถแก้ไขโดยใช้กล้องและการ์ดจับภาพ ที่มีเฟรมเรทสูง เพราะจะทำให้จับภาพที่มีความเคลื่อนไหวรวดเร็วได้ดีขึ้น

3. มีความคลาดเคลื่อนในการระบุตำแหน่งสูง เนื่องจากความละเอียดของภาพค่อนข้างต่ำ แก้ไขได้โดย ใช้กล้อง ที่มีความละเอียดของภาพสูง เพราะจะทำให้ภาพมีรายละเอียดที่มากขึ้น และควรมี

การหาคุณสมบัติกล้อง (Camera Calibration) เพื่อให้ได้ตำแหน่งของกล้อง และมุมกล้องที่ถูกต้องแม่นยำ ซึ่งจะทำให้ได้พิกัดที่ถูกต้องแม่นยำมากขึ้น

4. มีปัญหาในการระบุมาร์กเกอร์บ่อยครั้ง เนื่องจากใช้การประมวลผลภาพแบบขาว-ดำ ทำให้แยกแยะมาร์กเกอร์ได้ยาก แนวทางการแก้ไข คือ ใช้มาร์กเกอร์ที่มีสีต่างกัน และใช้กล้องที่จับสีได้คมชัด เพราะจะทำให้แยกแยะมาร์กเกอร์ได้ถูกต้องและง่ายขึ้น

## 5.4 แนวทางการพัฒนาต่อ

1. พัฒนาอัลกอริทึม ในการหา Blob และการติดตามมาร์กเกอร์ ให้มีประสิทธิภาพสูงกว่าเดิม โดยอาจจะใช้ระบบตรวจสอบย้อนกลับ (Inverse Kinemetic) เพื่อช่วยในการจำลองตำแหน่งมาร์กเกอร์ ให้ใกล้เคียงความเป็นจริงมากขึ้น

2. ใช้กล้องที่มีคุณภาพสูง และจำนวนมากขึ้น เพื่อเพิ่มประสิทธิภาพในการหามาร์กเกอร์

3. พัฒนาระบบการประมวลผลให้มีประสิทธิภาพสูงขึ้น เช่น การใช้การประมวลผลแบบหลายเธรด (Muti threading) เพื่อให้สามารถประมวลผลแบบเวลาจริง (Realtime) ได้

# เอกสารอ้างอิง


[1] นายสกล จารุเสรีนนท์ และ นายสุรพันธ์ จักรมณ. "ระบบสร้างตัวละคร 3 มิติโดยการตรวจจับการเคลื่อนไหวของร่างกาย". ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์. สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2549.

[2] Inition. "Animazoo IGS-190." [Online]. เข้าถึงได้จาก : http://www.inition.co.uk/inition/product.php?URL_product_mocaptrack_animazoo_gypsygyro_18&SubCatID_=20&Tab= theblurb. 2552

[3] "Motion Data Acquisition." [Online]. เข้าถึงได้จาก : http://vrlab.epfl.ch/research/LO_locomotin_engine.html. 2552

[4] Inition. "Gyspy6 Torso." [Online]. เข้าถึงได้จาก : http://www.inition.co.uk/inition/product.php?URL_=product_mocaptrack_animazoo_gypsy&SubCatID_=20&Tab=theblurb. 2552

[5] kho_hi (นามแฝง). "อิมเมจโปรเซสซิ่งเบื้องต้น." [Online]. เข้าถึงได้จาก : http://www.bloggang.com/viewdiary.php?id=it-mixs&group=1. 2551

[6] ดร.มนตรี กาญจนะเดชะ. "Course Meterials for 240-373: Image Processing." [Online] เข้าถึงได้จาก : http://fivedots.coe.psu.ac.th/~montri/Teaching/image/image.htm. 2551

[7] 4-FLYING SRL. "RGB Workflow." [Online]. เข้าถึงได้จาก : http://www.4-flying.com/flyingcolours/en/content/rgb-workflow . 2552

[8] "Histogram-Based Color Image Retrieval." [Online]. เข้าถึงได้จาก : http://scien.stanford.edu/class/psych221/projects/02/sojeong. 2552

[9] วิบูลย์ แสงวีระพันธุ์ศิริ และ พงศกร เพชรพันธ์ศรี. "เครื่องวัดพิกัด 3 มิติโดยใช้ภาพสเตอริโอ." [Online] เข้าถึงได้จาก : http://www.me.psu.ac.th/ME_NETT/paper/DRC/DRC011.pdf. 2551

[10] กันตนา แอนนิเมชัน. "ก้านกล้วย." [Online]. เข้าถึงได้จาก : http://www.khankluay-themovie.com. 2552

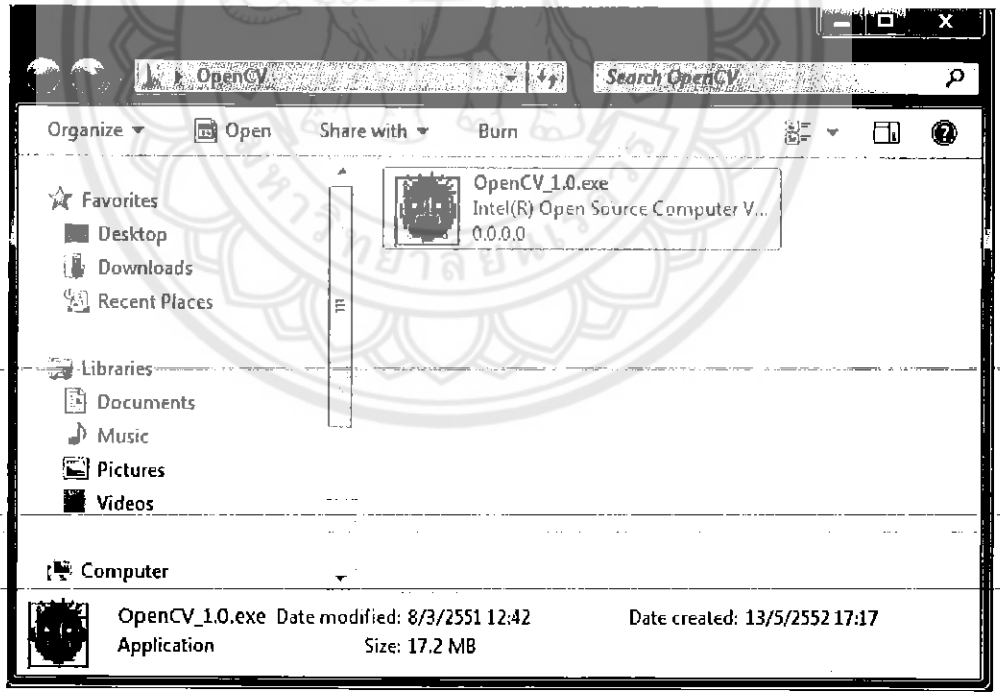
# ภาคผนวก

# OpenCV

**OpenCV** เป็นไลบรารี่สำหรับใช้ในการประมวลผลภาพ (Image Processing) ซึ่งเป็นไลบรารี่โอเพนซอร์ส (Open Source) สามารถดาวน์โหลดใช้งานได้ฟรี ไลบรารี่ต่างๆของ OpenCV ได้พัฒนาขึ้นด้วยบริษัทอินเทล (Intel) จุดเด่นในด้านความสามารถของไลบรารี่ OpenCV คือ สามารถประมวลผลภาพ3 มิติ ได้ทั้งภาพนิ่ง และภาพเคลื่อนไหวเช่น ภาพจากกล้องวีดีโอ หรือไฟล์วีดีโอเป็นต้น โดยไม่ยึดติดทางด้านฮาร์ดแวร์ทำให้ OpenCV สามารถพัฒนาโปรแกรมได้หลากหลายภาษา รวมถึงมีฟังก์ชันสำเร็จรูปสำหรับจัดการข้อมูลภาพ และการประมวลผลภาพพื้นฐานเช่น การหาขอบภาพ การกรองข้อมูลภาพ โดยฟังก์ชันต่างๆของ OpenCV จะสามารถเรียกใช้งานได้จะต้องมีการเรียก ไฟล์ส่วนหัว (Header file) และลิงค์ (Link) ไลบรารี่ต่างๆ รวมถึง DLL (Dynamic Link Library)

ขั้นตอนการติดตั้ง OpenCV

1. เปิดไฟล์ OpenCV_1.0.exe เพื่อเริ่ม Setup



2. คลิก Next

**Welcome to the Intel(R) Open Source Computer Vision Library Setup Wizard**

This will install Intel(R) Open Source Computer Vision Library 1.0 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Next > Cancel

3. เลือก I accept the agreement แล้วกด Next

**License Agreement**

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

IMPORTANT: READ BEFORE DOWNLOADING, COPYING, INSTALLING OR USING.

By downloading, copying, installing or using the software you agree to this license. If you do not agree to this license, do not download, install, copy or use the software.

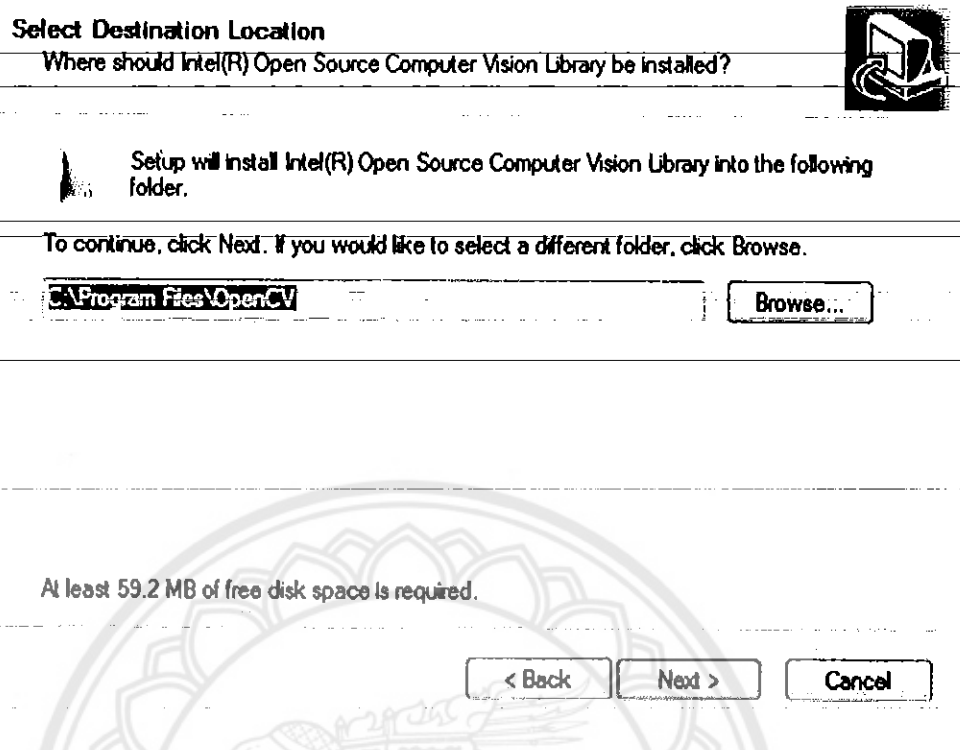
Intel License Agreement
For Open Source Computer Vision Library

(•) I accept the agreement

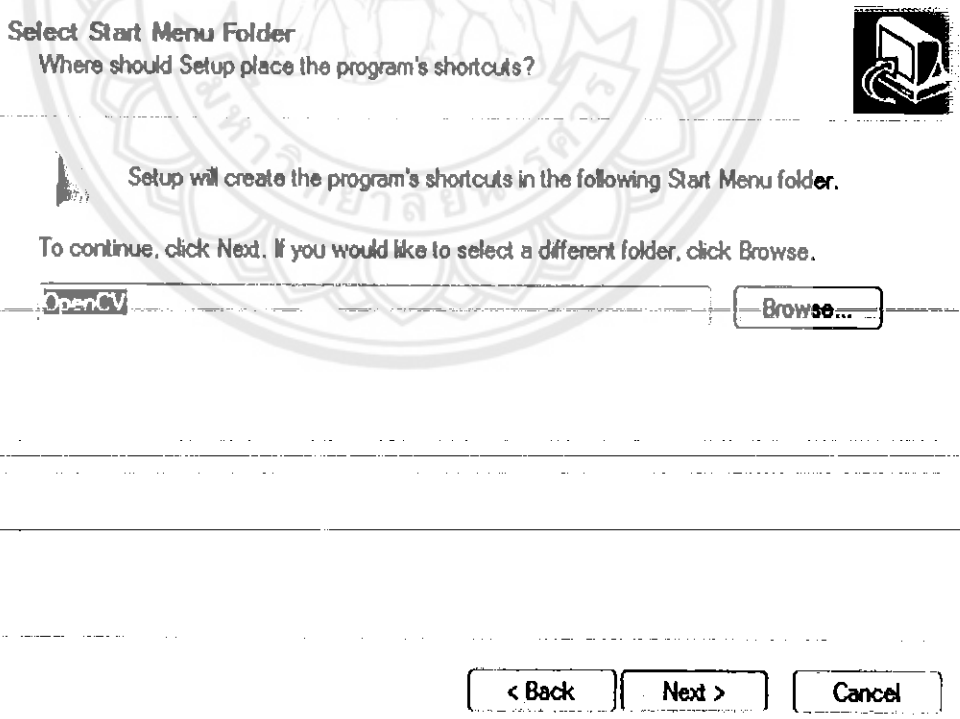
( ) I do not accept the agreement

< Back Next > Cancel

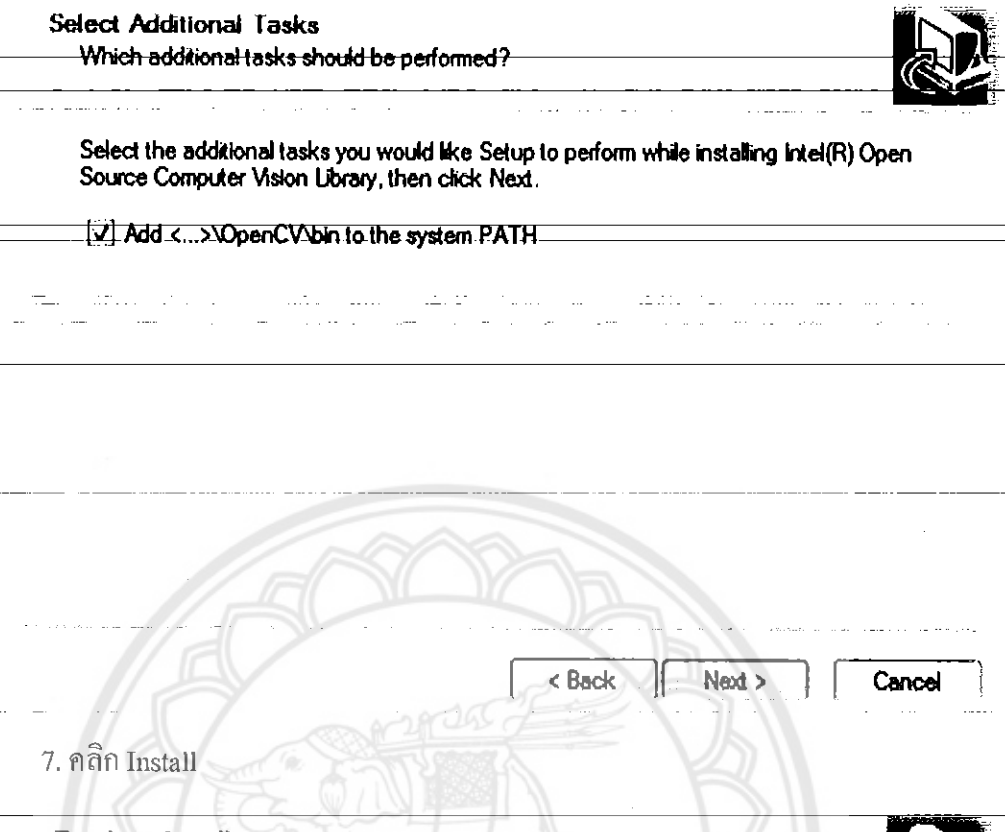
4. กด Next



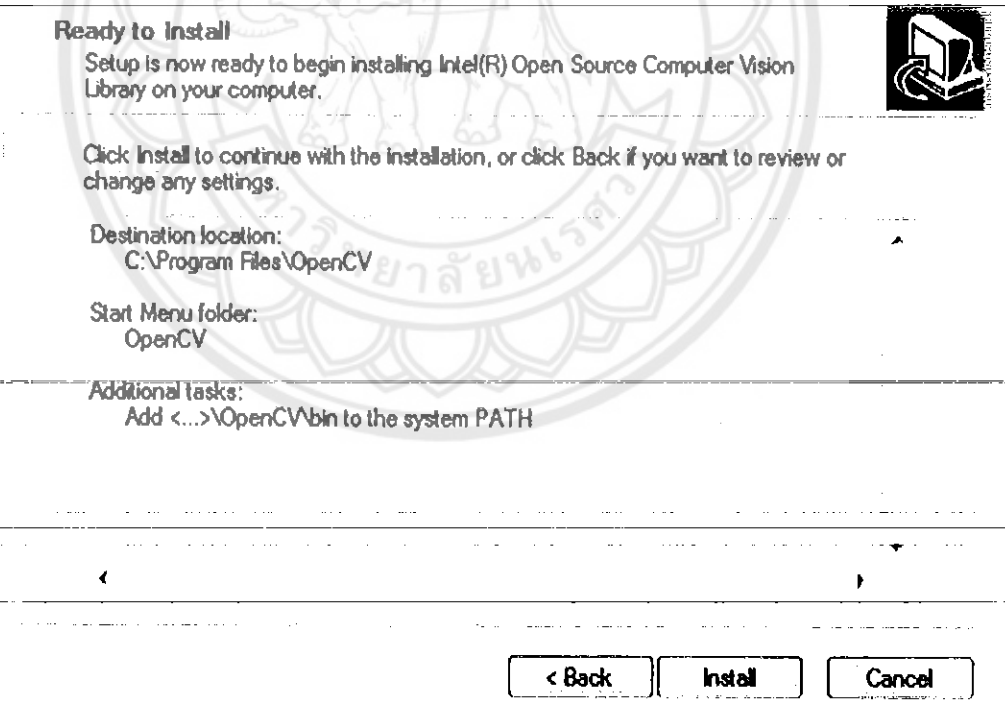

5. กด Next



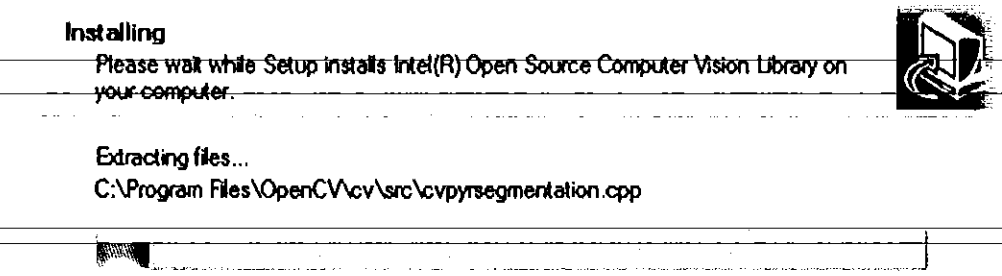
6. ติ๊กถูกแล้วกด Next



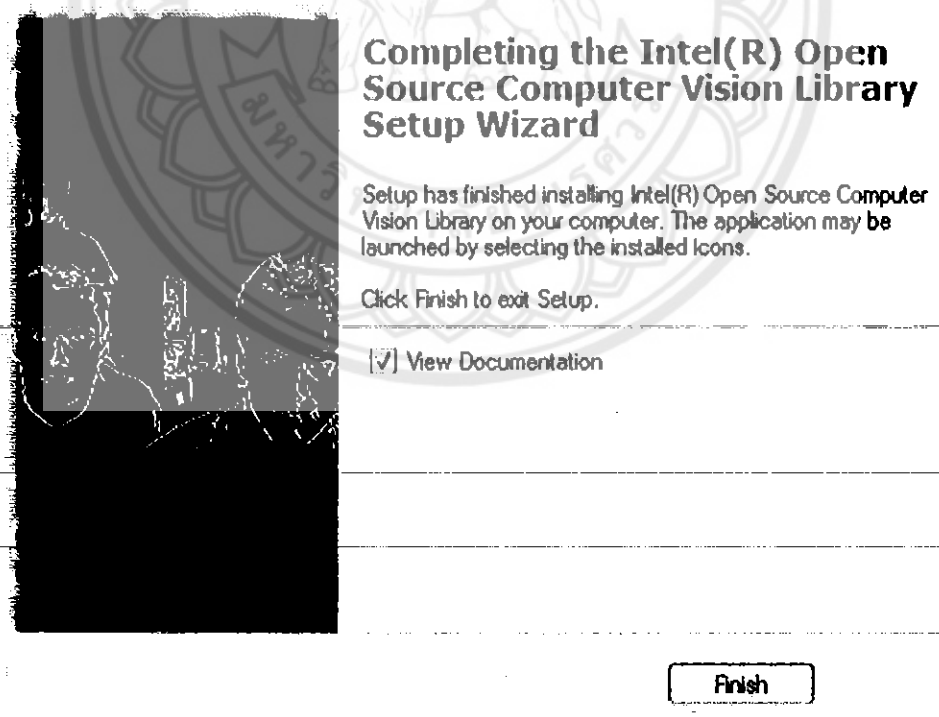

7. คลิก Install



8. รอ...

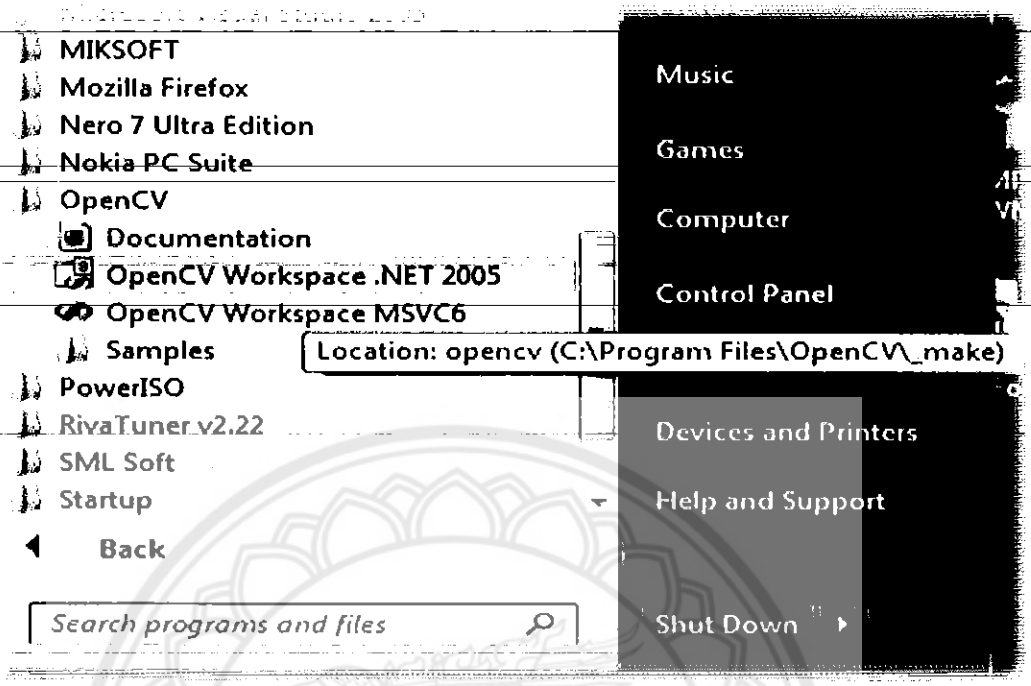



9. คลิก Finish

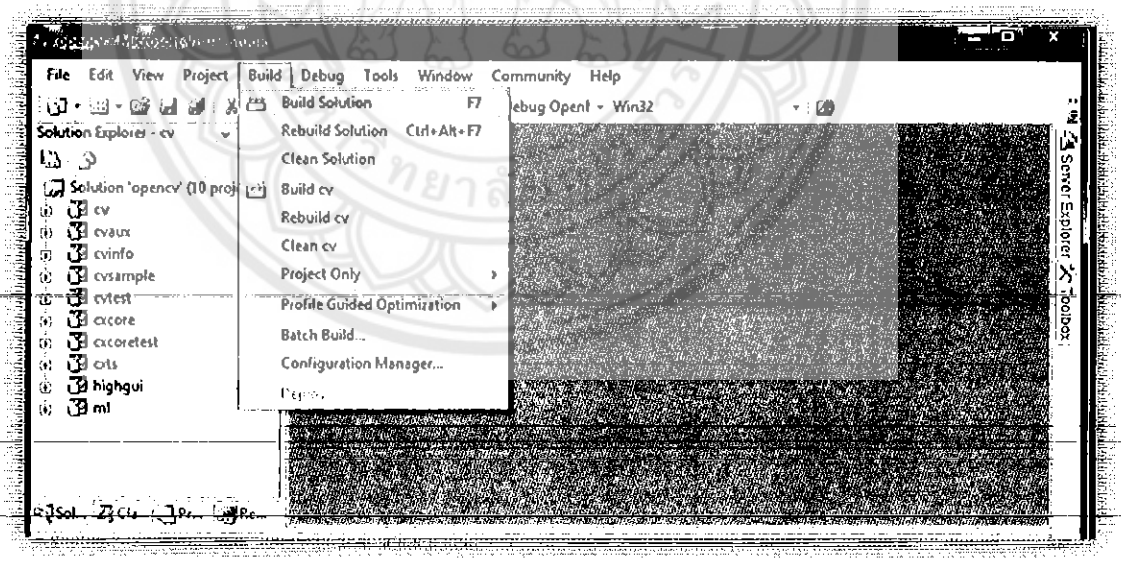


การติดตั้ง OpenCV เสร็จแล้ว ต่อไปเป็นการ built

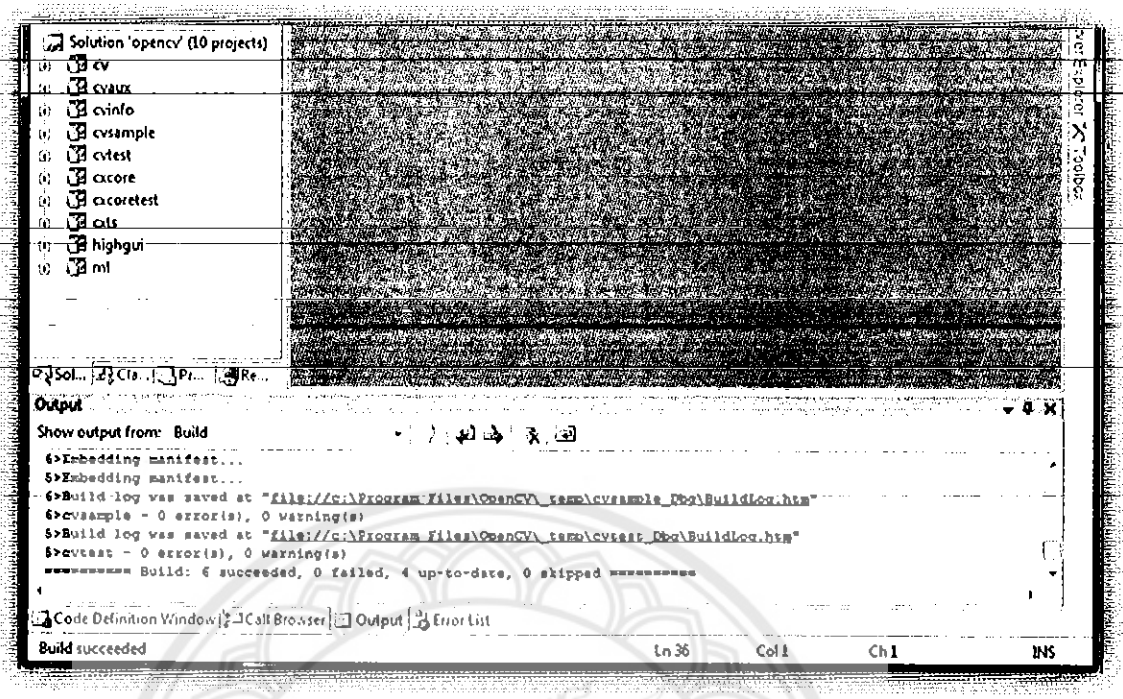
1. เปิด OpenCV Workspace.Net 2005




2. กด Built Solution หรือ กด F7

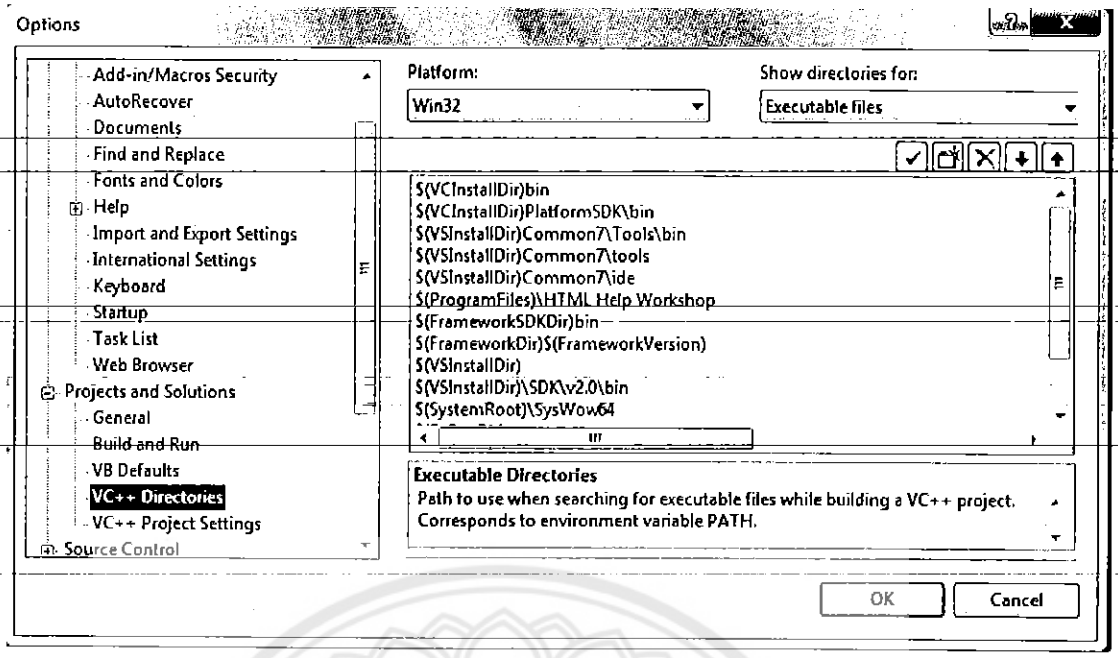


3. รอจน built เสร็จ




4. เมื่อทำการติดตั้งไลบรารี่ OpenCV ลงเครื่องคอมพิวเตอร์เรียบร้อยแล้ว ก่อนที่โปรแกรมจะสามารถเรียกใช้คำสั่ง OpenCV ได้ จะต้องกำหนดการเชื่อมโยงไฟล์ไลบรารี่ต่างๆ ให้กับ Visual C++.NET 2005 ดังนี้

4.1 เปิดโปรแกรม Visual C++.NET 2005 ที่ เมนูบาร์เลือก Tool> Options จากนั้นเลือกที่ “Project -> VC++ Directories” เข้าไปที่ Tool->Option->Project and Solutions -> VC++ directories




4.2 เลือกไดเร็กทอรี่แล้วทำการเพิ่มตำแหน่งพาร์ท (Path) สำหรับการเชื่อมโยง Library ของ OpenCV กับ Visual C++.NET ดังนี้

Executable files

C:\Program Files\OpenCV\bin

Include files

C:\Program Files\OpenCV\cv\include

C:\Program Files\OpenCV\cvaux\include

C:\Program Files\OpenCV\otherlibs\cvcam\include

C:\Program Files\OpenCV\otherlibs\highgui

C:\Program Files\OpenCV\cxcore\include

Libraly file

C:\Program Files\OpenCV\lib

C:\Program Files\OpenCV\otherlibs

C:\Program Files\OpenCV\otherlibs\cvcam

C:\Program Files\OpenCV\otherlibs\highgui

C:\Program Files\OpenCV\cxcore\includ

จบการ Install Open CV

# ประวัติผู้เขียนโครงงาน

ชื่อ นายทรงกรด พิมใจ

ภูมิลำเนา 429/3 ม.2 ต.ลาดแค อ.ชนแดน จ.เพชรบูรณ์ 67150

ประวัติการศึกษา

- จบการศึกษาระดับมัธยมศึกษาจากโรงเรียนดงขุยวิทยาคม
- ปัจจุบันกำลังศึกษาอยู่ในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : **krod-cpe@hotmail.com**

ชื่อ นายธีรโรจน์ พันภัย

ภูมิลำเนา 342 ม.11 ต.บ้านตึก อ.ศรีสัชนาลัย จ.สุโขทัย 64130

ประวัติการศึกษา

- จบการศึกษาระดับมัธยมศึกษาจากโรงเรียนเมืองเชลียง
- ปัจจุบันกำลังศึกษาอยู่ในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : **teerarod@hotmail.com**