



# หุ่นยนต์คัดแยกและเก็บขยะรีไซเคิลโดยใช้กล้อง CCD


โดย

วิระยุทธ ศรีธรรวานิช  
รัชทิน จันทรเจริญ

โครงการวิจัยเลขที่ 164-เครื่องกล-2543  
ทุนส่งเสริมการวิจัยคณะวิศวกรรมศาสตร์

คณะวิศวกรรมศาสตร์  
จุฬาลงกรณ์มหาวิทยาลัย  
กรุงเทพฯ

กันยายน 2544



สถาบันวิจัยและพัฒนาของ คณะวิศวกรรมศาสตร์ ไม่รับผิดชอบ  
ต่อผลเสียใด ๆ อันอาจเกิดจากการนำความคิดเห็นในเอกสาร  
ฉบับนี้ไปใช้ ความคิดเห็นที่ปรากฏในเอกสารเป็นความคิดเห็น  
ของผู้เขียนซึ่งไม่จำเป็นต้องเป็นความคิดเห็นของสถาบันฯ

จุฬาลงกรณ์มหาวิทยาลัย

# หุ่นยนต์คัดแยกและเก็บขยะรีไซเคิลโดยใช้กล้อง CCD

โดย



วีระยุทธ ศรีสุวรรณิช M.S.M.E.(Nagoya)

รัชทิน จันทร์เจริญ Ph.D.(Chula)

โครงการวิจัยเลขที่ 164-เครื่องกล-2543

ทุนส่งเสริมการวิจัยคณะวิศวกรรมศาสตร์

สถาบันวิทยบริการ

จุฬาลงกรณ์มหาวิทยาลัย

คณะวิศวกรรมศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย

กรุงเทพฯ

กันยายน 2544

14 ส.ค. 2547

I 20122688

## บทคัดย่อ

โครงการนี้มีวัตถุประสงค์ที่จะพัฒนาหุ่นยนต์ที่มีศักยภาพสูงสามารถทำงานในสภาพแวดล้อมต่างๆไป โดยพัฒนาหุ่นยนต์ที่สามารถตรวจหาวัตถุเป้าหมายในสภาพแวดล้อมและสามารถเดินไปหยิบวัตถุนั้น ซึ่งเป็นการเริ่มต้นการทำงานที่ซับซ้อนต่อไป ในรายงานฉบับนี้จะประกอบด้วย 2 ส่วนคือการพัฒนาตัวหุ่นยนต์และการพัฒนากระบวนการในการตรวจหาวัตถุในสิ่งแวดล้อมทั่วไป

ในการพัฒนาตัวหุ่นยนต์ได้พัฒนาหุ่นยนต์ที่สามารถหยิบจับขยะรีไซเคิลตามวัตถุประสงค์ของโครงการ โดยเน้นการพัฒนากลไกการจับวัตถุให้มีความยืดหยุ่นสามารถหยิบจับวัตถุที่มีรูปร่างต่างๆได้ พร้อมทั้งติดตั้งอุปกรณ์ตรวจรู้ (sensor) ต่างๆ เพื่อให้หุ่นยนต์สามารถทำงานต่างๆในสภาพแวดล้อมต่างๆไปได้

ในการพัฒนากระบวนการในการค้นหาวัตถุเป้าหมายในสิ่งแวดล้อมทั่วไปแบ่งขั้นตอนการทำงานเป็น 2 ขั้นตอนคือ ขั้นตอนการป้อนข้อมูลวัตถุเป้าหมาย โดยพัฒนาวิธีการป้อนข้อมูลวัตถุเป้าหมายอย่างอัตโนมัติ และขั้นตอนในการตรวจหาวัตถุเป้าหมายได้เสนอกระบวนการค้นหา 2 วิธีการ

วิธีแรกเป็นการค้นหาวัตถุด้วยข้อมูลสี โดยเลียนแบบกระบวนการค้นหาขั้นต้นของมนุษย์ กระบวนการนี้จะเริ่มจากการตรวจหากลุ่มสีหลักๆของวัตถุเป้าหมาย แล้วค้นหาบริเวณที่มีกลุ่มสีหลักเหล่านั้นที่อยู่ในบริเวณใกล้เคียงกันในสภาพแวดล้อม ซึ่งเป็นบริเวณที่มีโอกาสที่จะมีวัตถุเป้าหมายอยู่สูง

ส่วนวิธีที่สองเป็นการค้นหาโดยใช้ข้อมูลรูปร่างของวัตถุ เริ่มด้วยการตรวจหาวัตถุย่อยๆในสิ่งแวดล้อม แล้วทำการตรวจสอบความคล้ายของรูปร่างวัตถุทั้งหมดที่พบในสิ่งแวดล้อมเปรียบเทียบกับวัตถุเป้าหมายเพื่อค้นหาวัตถุที่มีรูปร่างคล้ายกับวัตถุเป้าหมายโดยใช้ตัวแปรโมเมนต์ไม่ผันแปร ในการวิจัยได้เสนอวิธีการในการแยกวัตถุย่อยๆนี้

เลขหมู่	กท
	๐๕ 15
เลขทะเบียน	๐1๐ 29๐
วัน,เดือน,ปี	25 ๕๓. 44



## Abstract

This research aims to develop high potential robot that is possible to execute tasks in unconstructed environment. The developed robot can recognise and go to grasp a target object in unconstructed environment that is the basic task of complex tasks. The report consists of 2 parts: robot development and target object detection algorithm development.

In the robot development part, a robot was developed according to the project purpose, especially the grasp mechanism that was developed to be flexible for grasping many types of objects. Moreover several sensors were installed in the robot for the purpose of increasing the robot ability in unconstructed environment.

In the target detection algorithm development part, there are 2 procedures: target data inputting procedure that is developed to input the data automatically and target detection procedure that we proposed 2 algorithms.

One is to detect the target object using color information that imitated the primary detection method of human, by seeking the main elemental colors of the target in the environment. The area that these main colors are located in, has high occasion to be the target object.

Second is to detect by the shape of target. First seek all sub-objects in the environment and compare these sub-objects with the target using Moment Invariant indice to find similar ones. In this part, we proposed a new method for the sub-object seeking.

## กิตติกรรมประกาศ

ปัจจัยที่ทำให้โครงการนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี ส่วนหนึ่งได้รับความร่วมมือจากเหล่านิสิตภาค วิชาวิศวกรรมเครื่องกลดังมีรายชื่อดังนี้

นาย	ชยุตม์	กฤษอร่ามเรือง	เลขประจำตัว	403 00911 21
นาย	ณรงค์ศักดิ์	โควินทเศรษฐู	เลขประจำตัว	403 01288 21
นาย	ต๋องพงศ์	อมรตระกูล	เลขประจำตัว	403 01614 21
นาย	ประเสริฐ	เพียรเลิศวรกุล	เลขประจำตัว	403 02840 21
นางสาว	ปราณูชลี	ปัญญะภาคย์	เลขประจำตัว	403 02862 21
นาย	พงษ์ศักดิ์	เล็กสกุลไชย	เลขประจำตัว	403 03141 21
นาย	ภัทรวุฒิ	ทรัพย์ไสวผล	เลขประจำตัว	403 03571 21
นางสาว	เมธาวี	สุคมภิรานนท์	เลขประจำตัว	403 03857 21
นางสาว	รัศมี	พูนรากุล	เลขประจำตัว	403 04033 21
นาย	ฤทธิชัย	เรียวิไลสุข	เลขประจำตัว	403 04062 21
นาย	สมบูรณ์	จิตโภคเกษม	เลขประจำตัว	403 05086 21
นาย	อนุชิต	บุตรขุนทอง	เลขประจำตัว	403 05922 21
นาย	มงคล	เทียนวิบูลย์	นิสิตปริญญาเอก (ผู้ช่วยนักวิจัย)	

โดยได้รับเงินทุนสนับสนุนจากทางคณะวิศวกรรมศาสตร์ นอกจากนี้ ความรู้ความสามารถเฉพาะทางต่าง ๆ ที่ เหล่าคณะผู้จัดทำไม่สามารถกระทำการได้เพียงลำพัง หากแต่ได้รับความกรุณา ความโอบอ้อมอารี ความเกื้อหนุน จากคณาจารย์ บุคลากร ตลอดจนนิสิตปริญญาโท ทางคณะผู้จัดทำมีความซาบซึ้งใจเป็นอย่างยิ่ง จึงขอแสดงความขอบพระคุณท่านเหล่านี้ไว้ ดังต่อไปนี้

รศ.ดร. วิบูลย์ แสงวีระพันธุ์ศิริ สำหรับความกรุณาที่ท่านอนุญาตให้สามารถใช้อห้องปฏิบัติการโคลัมโบได้ โดยสะดวก

นอกจากนี้ อาจารย์ ดร. วิทยา วัฒนสุโขประสิทธิ์ อาจารย์ ดร.วัชระ เลิศพิริยสุวัฒน์ และ อาจารย์ ดร. สมพงษ์ พุทธิวิสุทธิศักดิ์ สำหรับคำแนะนำ และความเมตตาที่มอบให้กับผู้จัดทำ

นิสิตปริญญาโทภาคเครื่องกล ที่ช่วยในส่วนการปฏิบัติการ กระบวนการผลิตมือ และโครงหุ่น การใช้เครื่อง CNC รวมทั้ง อุปกรณ์ วัสดุการทำงานมากมาย...

สุดท้ายนี้ ขอคุณพระศรีรัตนตรัยและสิ่งศักดิ์สิทธิ์ในสากลโลก โปรดช่วยคุ้มครองให้ทุกท่านประสบแต่ความสุข ความเจริญ สมหวังในสิ่งที่ปรารถนาทุกประการ เทอญ.

# สารบัญ

	หน้า
1. บทนำ	
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	2
1.4 ประโยชน์ที่ได้รับจากโครงการ	2
1.5 ภาพรวมของโครงการ	2
1.6 ส่วนประกอบของรายงาน	4
2. โครงสร้างหุ่นยนต์ (Hardware)	
2.1 ระบบกลและโครงหุ่น	5
2.1.1 ระบบมือกล	5
2.1.2 ระบบขับเคลื่อนแขนและข้อมือ	20
2.1.3 ระบบขับเคลื่อนล้อและโครงสร้างฐาน	25
2.2 ระบบไฟฟ้าและอุปกรณ์ตรวจรู้	29
2.2.1 เซลตีปมอเตอร์	29
2.2.2 วงจรขับเซลล์ปมอเตอร์	33
2.2.3 มอเตอร์กระแสตรงและวงจรขับ	42
2.2.4 แผงวงจรควบคุม (control board)	43
2.2.5 การเชื่อมต่อสายสัญญาณจาก microcontroller	46
2.2.6 Regulator	47
2.2.7 แบตเตอรี่	48
2.2.8 กาลังควบคุม	49
2.2.9 อุปกรณ์ที่ใช้ในส่วนระบบไฟฟ้า	51
2.2.10 อุปกรณ์ตรวจรู้ (sensor)	54
2.3 โปรแกรมควบคุมการทำงาน	58
2.3.1 โปรแกรมควบคุม stepping motor	58
2.3.2 โปรแกรมควบคุม DC motor ของ แขน ข้อมือ และนิ้วมือ	63
2.3.3 โปรแกรมการรับข้อมูลจากอุปกรณ์ตรวจรู้	65
2.4 บทสรุป	67

3.	กระบวนการตรวจหาวัตถุเป้าหมาย (Software)	
3.1	ขั้นตอนการป้อนข้อมูลวัตถุเป้าหมาย	70
3.1.1	การปรับแต่งภาพโดยการกรองแบบมัลติฐาน	71
3.1.2	ค่าความแตกต่างระหว่างพิกเซลและการกำหนดค่า Threshold	73
3.1.3	ขอบเขตวัตถุ Object Boundary	76
3.2	ขั้นตอนการตรวจหาวัตถุในสภาพแวดล้อมทั่วไป	80
3.2.1	กระบวนการค้นหาวัตถุโดยใช้กลุ่มสี	81
3.2.2	กระบวนการค้นหาวัตถุโดยใช้ข้อมูลรูปร่างของวัตถุ	85
3.3	การพัฒนาโปรแกรม	94
3.3.1	การพัฒนาโปรแกรมค้นหาวัตถุด้วยกล้อง CCD	94
3.3.2	โปรแกรมประมวลผลภาพพื้นฐาน	100
3.4	บทสรุป	121
4.	บทสรุปโครงการและโครงการวิจัยต่อเนื่อง	122
5.	เอกสารอ้างอิง	125
6.	ภาคผนวก	127



## 1. บทนำ

### 1.1 ความสำคัญและที่มา

ปัจจุบันแนวโน้มของการนำหุ่นยนต์มาประยุกต์ใช้ในภาคอุตสาหกรรมเพิ่มมากขึ้นเรื่อยๆ หุ่นยนต์ที่มีศักยภาพสูงย่อมเป็นที่ต้องการของตลาด เป็นเหตุให้งานวิจัยต่างๆ ได้พัฒนาไปอย่างต่อเนื่อง เพื่อพัฒนาหุ่นยนต์ที่มีความสามารถทำงานแทนมนุษย์ได้ แนวความคิดที่จะสร้างหุ่นยนต์เลียนแบบมนุษย์ให้ได้มากที่สุดเริ่มได้รับความสนใจมากขึ้น การเพิ่มประสาทรับรู้ให้กับหุ่นทำให้หุ่นสามารถทำงานได้หลากหลาย และสามารถตอบสนองกับความต้องการของมนุษย์ได้อย่างมีประสิทธิภาพ กล้องดิจิทัลเป็นทางเลือกหนึ่งที่จะทำให้หุ่นยนต์สามารถรับรู้ภาพและทำงานตามที่กำหนดไว้ในสภาวะแวดล้อมต่างๆ ได้ นอกจากนี้แล้วอุปกรณ์ตรวจจับ (sensor) ต่างๆ ยังทำหน้าที่เป็นประสาทรับรู้ให้กับหุ่นได้อีกด้วย ยกตัวอย่างเช่น อุปกรณ์ตรวจจับการสัมผัส จะช่วยให้หุ่นสามารถรับรู้ได้เมื่อเคลื่อนที่ไปชนสิ่งกีดขวาง หรืออุปกรณ์ตรวจจับทางเสียงที่สามารถทำให้หุ่นยนต์ปฏิบัติงานตอบสนองกับเสียงที่เกิดขึ้นได้

โครงการหุ่นยนต์คัดแยกขยะรีไซเคิลเป็นตัวอย่างหนึ่งของการใช้อุปกรณ์ตรวจจับเพื่อเพิ่มขีดความสามารถของหุ่น โดยใช้กล้องดิจิทัลบันทึกภาพ ข้อมูลภาพที่ได้นี้เทียบได้กับการมองเห็นของมนุษย์ ทำหน้าที่เป็นประสาทรับรู้ให้กับหุ่น วัตถุประสงค์หลักของโครงการคือ พัฒนารุ่นหุ่นที่สามารถตรวจหาวัตถุเป้าหมาย (ขยะรีไซเคิล) ในสภาวะแวดล้อมต่างๆ ไปอันเป็นจุดเริ่มต้นของการทำงานที่ซับซ้อนต่างๆ ภายใต้สภาวะแวดล้อมทั่วไป โครงการหุ่นยนต์คัดแยกขยะรีไซเคิล แบ่งออกเป็น 2 ส่วนคือ

- โครงสร้างหุ่นยนต์ (hardware) โครงการนี้จะประดิษฐ์หุ่นยนต์เพื่อเคลื่อนที่ไปเก็บขยะจากตำแหน่งที่ระบุไว้แล้วนำไปทิ้งในตำแหน่งที่กำหนดได้
- กระบวนการการตรวจหาวัตถุเป้าหมายด้วยข้อมูลภาพจากกล้อง CCD (software) ส่วนนี้จะคิดกระบวนการและพัฒนาโปรแกรมที่สามารถตรวจหาวัตถุที่ต้องการในสิ่งแวดล้อมที่หลากหลายได้

ในส่วนแรกของการศึกษาวิจัยได้พัฒนาและประดิษฐ์หุ่นยนต์ที่ใช้ในการตรวจหาและสามารถไปเก็บวัตถุเป้าหมายซึ่งประกอบไปด้วยระบบมือกล ระบบแขน ข้อมือ ฐานล่าง ระบบขับเคลื่อน นอกจากนี้ยังมีส่วนของวงจรไฟฟ้า อุปกรณ์ตรวจจับและการโปรแกรมในการควบคุมการทำงาน ในส่วนที่สองศึกษาและพัฒนากระบวนการในการตรวจหาวัตถุในสิ่งแวดล้อมต่างๆ ไปจากข้อมูลภาพจากกล้อง CCD

### 1.2 วัตถุประสงค์ของโครงการ

1. เพื่อเป็นการศึกษาวิจัยทางด้าน Robotics และ Artificial Intelligence
2. เพื่อนำความรู้ที่ได้จากการวิจัยมาประยุกต์ใช้ให้เกิดประโยชน์
3. เพื่อศึกษาและพัฒนาหุ่นยนต์ที่สามารถทำงานในสภาวะแวดล้อมที่หลากหลายได้

4. เพื่อพัฒนากระบวนการและโปรแกรมทางด้าน Image processing ที่สามารถประยุกต์ใช้ในงานอุตสาหกรรมได้

### 1.3 ขอบเขตของโครงการ

ในโครงการนี้มีวัตถุประสงค์ของโครงการโดยประดิษฐ์หุ่นยนต์เพื่อตรวจหาและไปเก็บวัตถุเป้าหมาย(ขยะรีไซเคิล) โดยมีขอบเขตดังนี้

1. ประดิษฐ์มีอกลให้สามารถหยิบจับวัตถุได้หลายชนิดที่มีความแตกต่างกันในด้านน้ำหนักและรูปร่าง
2. ประดิษฐ์มือให้มีประสิทธิภาพ เสถียรภาพและความแข็งแรงมากพอที่จะจับวัตถุได้โดยไม่ทำลายหรือสร้างความเสียหายให้แก่วัตถุ
3. ประดิษฐ์หุ่นยนต์ที่มีอุปกรณ์รับรู้สามารถทำงานบรรลุได้ตามวัตถุประสงค์ที่ตั้งไว้
4. สรรหากระบวนการประมวลผลภาพ พร้อมกับ พัฒนาโปรแกรมเพื่อการตรวจหาวัตถุ ที่มีประสิทธิภาพ และ มีความยืดหยุ่นใช้ได้ในสภาวะแวดล้อมที่หลากหลายได้
5. พัฒนาโปรแกรม สำหรับทดสอบกระบวนการประมวลผลภาพหลากหลายรูปแบบ ตามที่ได้ทำการศึกษาในขั้นรวบรวมข้อมูล โดยโปรแกรมต้องมีความยืดหยุ่น

### 1.4 ประโยชน์ที่ได้รับจากโครงการ

1. เสริมสร้างการค้นคว้าวิจัยในด้าน Robotics และ Artificial intelligent ในประเทศ
2. เป็นแนวความคิดและจุดริเริ่มของงานวิจัยใหม่ๆ ที่สามารถนำไปพัฒนาต่อให้เกิดประโยชน์ได้
3. สามารถพัฒนาหุ่นยนต์ศักยภาพสูงที่สามารถนำไปใช้ประโยชน์ในงานต่างๆได้
4. สามารถพัฒนาโปรแกรมในการประมวลผลภาพที่สามารถนำไปประยุกต์กับงานต่างๆได้
5. ช่วยส่งเสริมในการผลิตบุคลากรทางด้าน Robotics และ Artificial intelligent ในประเทศ

### 1.5 ภาพรวมโครงการ

โครงการคัดแยกและเก็บขยะรีไซเคิลโดยใช้กล้อง CCD มีจุดประสงค์ที่จะพัฒนาหุ่นยนต์ให้มีศักยภาพสูงขึ้นในการทำงานในสภาพแวดล้อมที่ไม่ได้จัดเตรียมไว้ โดยใช้กล้องดิจิทัลซึ่งเป็นอุปกรณ์ตรวจรู้อย่างหนึ่งมาประยุกต์ใช้กับหุ่นยนต์ เพื่อให้หุ่นยนต์สามารถค้นหาวัตถุที่ต้องการ และสามารถเดินไปหยิบวัตถุนั้นได้ โดยในรายงานฉบับนี้จะประกอบด้วย 2 ส่วนคือการสร้างและพัฒนาตัวหุ่นยนต์เก็บขยะและการพัฒนากระบวนการในการตรวจหาวัตถุในสิ่งแวดล้อมต่างๆไป ซึ่งถือเป็นการทำงานพื้นฐานของงานที่ซับซ้อนต่อไป

ส่วนของการสร้างหุ่นยนต์ได้ทำการพัฒนาหุ่นยนต์ซึ่งแบ่งออกได้เป็นส่วนหลัก 3 ส่วนคือ ส่วนของระบบกลที่ประกอบด้วย กลไกการจับ ระบบแขนและโครงสร้างฐานล่าง ส่วนของระบบไฟฟ้า และส่วนของโปรแกรมควบคุม

การทำงานของหุ่นยนต์ โดยในส่วนของระบบกลและไฟฟ้าได้ทำการคิด conceptual design พิจารณาความเหมาะสมของการนำมาใช้งานโดยเฉพาะกลไกการจับ (มือกล) ตลอดจนความเหมาะสมของวัสดุที่ใช้ เพื่อให้ทุกส่วนสามารถนำมาใช้งานได้อย่างมีประสิทธิภาพสูง นอกจากนี้การติดตั้งอุปกรณ์ตรวจรู้ (sensor) ที่สอดคล้องกับการใช้งาน เช่น กล้อง CCD และ limit switch ทั้งนี้การออกแบบในแต่ละชิ้นส่วน ได้พยายามคำนึงถึงความง่ายในการตรวจสอบและแก้ไขเป็นหลัก ทุกขั้นตอนจึงได้จัดให้มีการแสดงผลเพื่อช่วยในการตรวจสอบดังกล่าว อนึ่ง หุ่นยนต์ตัวนี้สามารถทำงานได้ตามวัตถุประสงค์ นอกจากนี้ได้นำหุ่นยนต์เก็บขยะนี้เข้าร่วมการแข่งขันหุ่นยนต์บริการสมองกลเพื่อการทดสอบการทำงานของหุ่นยนต์ ซึ่งประสบผลสำเร็จเป็นที่น่าพอใจในการแข่งขัน เมื่อรวมโครงงานนี้เข้ากับกระบวนการคิดแยกขยะด้วยกล้อง CCD หุ่นยนต์จะสามารถทำงานได้ตามเป้าหมาย และสามารถประยุกต์ใช้กับงานอื่นที่มีลักษณะคล้ายกันได้อีกด้วย

ส่วนที่สองคือการพัฒนากระบวนการในการค้นหาวัตถุเป้าหมายในสิ่งแวดล้อมทั่วไป โดยแบ่งขั้นตอนการทำงานเป็น 2 ขั้นตอนคือ ขั้นตอนการป้อนข้อมูลวัตถุเป้าหมายและขั้นตอนการตรวจหาวัตถุเป้าหมายในสภาพแวดล้อมที่หลากหลาย ทางคณะวิจัยได้เสนอขั้นตอนในการป้อนข้อมูลที่สามารถป้อนข้อมูลของวัตถุเป้าหมายได้โดยง่ายและเป็นอัตโนมัติเพียงนำวัตถุเป้าหมายมาวางในบริเวณจากหลังที่เป็นสีพื้น แล้วหุ่นยนต์ก็จะจำแนกว่าส่วนไหนเป็นส่วนที่เป็นวัตถุเป้าหมายและสามารถนำข้อมูลของวัตถุนั้นไปใช้ในขั้นตอนการค้นหาวัตถุในสภาพแวดล้อมทั่วไปต่อไปได้โดยง่าย ในขั้นตอนการตรวจหาวัตถุเป้าหมายในสภาพแวดล้อมต่างๆไป เริ่มจากการศึกษาพื้นฐาน เช่น หลักการระบบการมองเห็นของคอมพิวเตอร์ หลักการของกล้องรับภาพ และ ทฤษฎีการประมวลผลสัญญาณภาพเบื้องต้น ได้แก่ กระบวนการหาขอบของวัตถุในภาพแบบต่างๆ เช่น การตรวจหาขอบโดยใช้ ตัวปฏิบัติการเกรเดียนท์ ตัวปฏิบัติการคอมพาส ตัวปฏิบัติการลาปลาเซียน ตัวปฏิบัติการสโตคาสติกเกรเดียนท์ และ กรรมวิธีหาขอบของแคนนีย์ [15, 24] (ดูภาคผนวก C) รวมทั้งกระบวนการประมวลผลสำหรับรูปภาพไบนารีเบื้องต้น ซึ่งได้แก่ การคำนวณพื้นที่ ตำแหน่ง และ แนวการวางตัวของวัตถุในภาพ และ ค่าฟังก์ชันโมเมนต์ไม่ผันแปรของวัตถุ และ กรรมวิธีการแปลงรูปแบบวัตถุในภาพไบนารี เพื่อหาลักษณะเฉพาะ (Features) บางอย่างเช่น โครงร่างของวัตถุ เส้นรอบรูปของวัตถุ ทฤษฎีทั้งหมดดังกล่าวได้ถูกรวบรวมสร้างขึ้นเป็นโปรแกรม Image Processing Toolbox เพื่อใช้ในการศึกษา ทดสอบกระบวนการประมวลผลภาพต่อไป

หลังจากได้ทำการศึกษาและพัฒนาโปรแกรมการประมวลผลภาพแบบต่างๆ มาเป็นระยะเวลาหนึ่ง ผู้จัดทำก็ได้ทำการพัฒนากระบวนการในการค้นหาวัตถุเป้าหมายภายในสิ่งแวดล้อม พร้อมทั้งพัฒนาโปรแกรมทดสอบเปรียบเทียบไปพร้อมกันไปด้วย โดยในช่วงแรก ผู้จัดทำได้พยายามที่จะใช้วิธีการค้นหาโดยใช้ข้อมูลของเส้น ซึ่งพบว่าการค้นหาด้วยวิธีการดังกล่าวจะมีความยากมาก เนื่องจากข้อมูลของเส้นที่ได้มีเป็นจำนวนมาก รวมทั้งยังมีปัญหาเส้นขาดตอนอีกด้วยจึงไม่สามารถนำข้อมูลเส้นเหล่านี้มาใช้ในการค้นหาวัตถุในสภาพแวดล้อมต่างๆไปได้ ภายหลังจากจึงได้ทำการพัฒนากระบวนการค้นหาวัตถุขึ้นมาสองวิธี โดยอยู่บนพื้นฐานข้อมูลที่เป็นพื้นที่แทนข้อมูลเส้น

วิธีแรกเป็นวิธีการค้นหาแบบคร่าวๆ (Roughly Search) ด้วยข้อมูลสี (ความเข้มแสง) โดยกระบวนการนี้ได้แนวความคิดมาจากกระบวนการค้นหาวัตถุขั้นต้นของมนุษย์ เริ่มต้นด้วยการสำรวจสีประกอบหลักๆของวัตถุที่ต้องการค้นหา จากนั้นจะกวาดสายตาค้นหาอย่างรวดเร็วเพื่อมองหาบริเวณที่มีสีดังกล่าวในสิ่งแวดล้อม เราสามารถลดโอกาสของความผิดพลาดได้ด้วยการค้นหาบริเวณที่มีกลุ่มสีหลักเหล่านั้นอยู่ในบริเวณใกล้เคียงกันมากๆเท่านั้น ก็จะ

ทำให้สามารถตัดบริเวณที่บังเงามีสีคล้ายกับวัตถุต้นแบบเพียงบางสีออกไปได้ จุดเด่นของกระบวนการค้นหานี้คือสามารถค้นหาวัตถุได้รวดเร็วมาก

ส่วนวิธีที่สองเป็น เป็นการค้นหาวัตถุโดยใช้ข้อมูลรูปร่างของวัตถุ ซึ่งใช้การแยกวัตถุในภาพออกเป็นวัตถุย่อยๆ แล้วทำการตรวจสอบความคล้ายของรูปร่างวัตถุที่พบในสิ่งแวดล้อมและวัตถุเป้าหมาย โดยใช้ค่าโมเมนต์ไม่ผันแปร Moment Invariants<sup>(13, 20, 21, 22)</sup> ซึ่งพบว่า วิธีการนี้มีประสิทธิภาพในการแยกวัตถุได้ดีในระดับหนึ่ง จุดเด่นของกระบวนการนี้คือสามารถที่จะแยกวัตถุย่อยๆ (บริเวณที่มีสีใกล้เคียงกัน) ออกมาได้ ดังนั้นจึงสามารถประยุกต์ใช้ในการหารูปทรงวัตถุ รวมทั้ง ตัวอักษร ลวดลายต่างๆบนวัตถุได้ด้วยซึ่งเป็นการเพิ่มความแม่นยำในการค้นหาวัตถุจำเพาะ (specific object) ได้ๆ เช่น จำแนกกระป๋องโค้กและกระป๋องเป๊ปซี่ได้

## 1.6 ส่วนประกอบของรายงาน

รายงานฉบับนี้ได้อธิบายโดยแบ่งเนื้อออกเป็น 2 ส่วนคือส่วนโครงสร้างหุ่นยนต์ (Hardware) และส่วนของกระบวนการในการตรวจหาวัตถุเป้าหมาย (Software) ซึ่งประเด็นและการทบทวนผลงานที่ผ่านมาจะได้อธิบายไว้ในตอนเริ่มต้นของแต่ละส่วน โดยรายงานฉบับนี้แบ่งออกเป็น 6 บท ดังนี้คือ

### บทที่ 1 บทนำ

ประกอบด้วยความสำคัญและที่มา วัตถุประสงค์ ขอบเขตของโครงการ ประโยชน์ที่ได้รับ ภาพรวมโครงการ และส่วนประกอบรายงาน

### บทที่ 2 โครงสร้างหุ่นยนต์

ประกอบด้วยระบบกล, ระบบไฟฟ้า, โปรแกรมและการทดสอบการทำงานของหุ่นยนต์

### บทที่ 3 กระบวนการในการตรวจหาวัตถุเป้าหมาย

ประกอบด้วยขั้นตอนในการป้อนข้อมูลของวัตถุ, ขั้นตอนในการตรวจหาวัตถุในสภาพแวดล้อมทั่วไปที่ประกอบด้วยกระบวนการตรวจหาวัตถุคร่าวๆโดยอาศัยกลุ่มสี และกระบวนการตรวจหาวัตถุโดยอาศัยรูปทรงวัตถุ

### บทที่ 4 บทสรุปของโครงการ

บทสรุปภาพรวมของโครงการ ปัญหาและจุดบกพร่องของขั้นตอน และ กระบวนการที่เสนอรวมทั้งโครงการวิจัยต่อเนื่อง

### บทที่ 5 เอกสารอ้างอิง

รวบรวมรายชื่อเอกสารข้อมูลต่างๆที่ค้นคว้าทั้งจากหนังสือ วารสาร บทความทางวิชาและ web site ทางอินเทอร์เน็ต

### บทที่ 6 ภาคผนวก

เป็นส่วนของข้อมูลพื้นฐาน และ ทฤษฎีที่อ้างอิงมาเพื่อใช้ประกอบรายงานและการค้นคว้าเพิ่มเติม



## 2. โครงสร้างหุ่นยนต์

โครงสร้างหุ่นยนต์ประกอบด้วยส่วนประกอบหลัก 3 ส่วนคือระบบกล, ระบบไฟฟ้า, และโปรแกรมในการควบคุมการทำงาน โดยในบทนี้จะกล่าวถึงระบบต่างๆเหล่านี้ซึ่งเป็นส่วนประกอบของหุ่นยนต์

### 2.1 ระบบกลและโครงหุ่น

ระบบกลและโครงหุ่นถูกออกแบบโดยแบ่งเป็น 3 ระบบตามลักษณะการทำงาน

- ระบบมือกล ออกแบบให้จับวัตถุที่หลากหลายประเภทได้อย่างมีประสิทธิภาพ
- ระบบขับเคลื่อนแขนและข้อมือ ออกแบบให้มีแขนที่สามารถยกมือขึ้นลงได้ด้วยการหมุน (Revolute joint) และนอกจากนี้ยังมีข้อมือ (Wrist) ที่ทำให้มือมีการหมุนได้ (Rotation) ที่ปลายแขน นอกเหนือจากการเคลื่อนที่ขึ้น ลงเพียงอย่างเดียว
- ระบบขับเคลื่อน ประกอบไปด้วย ล้อ และ มอเตอร์ขับเคลื่อน และ ในที่นี้ยังรวมไปถึงการออกแบบโครงสร้างของตัวรถให้แข็งแรง และต้องมีขั้นตอนในการสร้างที่ทำให้ระบบขับเคลื่อนสามารถทำงานได้อย่างมีประสิทธิภาพ

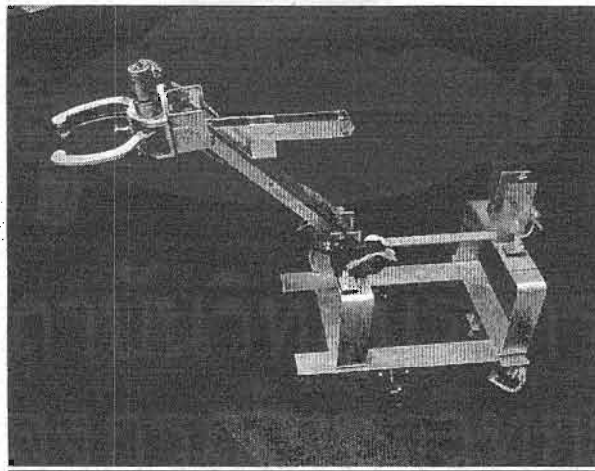


Fig 2.1-1 รูปโครงสร้างหุ่นยนต์

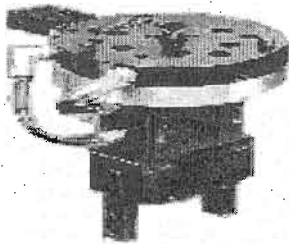
#### 2.1.1 ระบบมือกล

ในการออกแบบระบบมือกลนั้นมีเป้าหมายที่จะออกแบบมือกลให้สามารถจับวัตถุที่หลากหลายประเภทได้อย่างมีประสิทธิภาพโดยเริ่มจากการศึกษางานวิจัยที่ผ่านมาแล้ววิเคราะห์หรือออกแบบระบบมือกล จากนั้นจึงลงมือสร้างแล้วทำการทดสอบตามลำดับ

### 2.1.1.1 การศึกษางานวิจัยที่ผ่านมา ( Literature review )

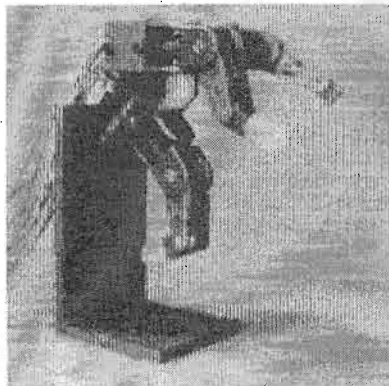
ในวงการวิจัยได้มีการประดิษฐ์มือหุ่นยนต์ขึ้นมาหลายรูปแบบ เพื่อทดลองใช้ในจุดประสงค์ต่างๆ ดังนั้นเพื่อให้ได้มือที่เหมาะสมกับหุ่นยนต์ที่ทำมากที่สุดจึงได้เริ่มศึกษาและเก็บข้อมูลเพื่อให้ทราบถึงข้อดี ข้อเสีย และความเหมาะสมที่จะนำมาประยุกต์ใช้ การประดิษฐ์มือ โดยได้ทำการหาข้อมูลจากสื่อต่างๆ ทั้งทางหนังสือ วารสารทางวิชาการและทาง internet และได้รวบรวมเนื้อหาที่น่าสนใจมานำเสนอไว้ ดังนี้

#### ตัวอย่างกลไกการจับ (Grasp Mechanics)



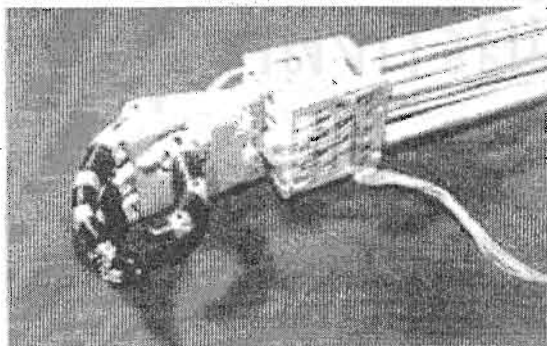
#### Traditional Robotic Gripper

เป็นมือหุ่นยนต์ที่เก่าแก่ที่สุดโดยมีลักษณะเป็นคีม ซึ่งแขนของคีมจะสามารถขยายออกได้ทั้งวิธีการหมุนรอบจุด หรือแบบใช้หลักการของชิ้นส่วนขนาน ซึ่งเป็นมือที่ประกอบด้วย 2 องศาอิสระ และทำงานได้ดีในระนาบเหมาะกับงานง่าย แต่การจับยังมีเสถียร



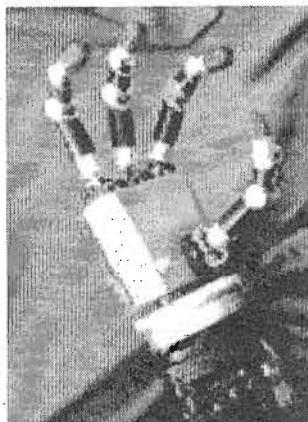
#### Stanford/JPL Hand

ถูกออกแบบโดย Kenneth J. Salisbury ในระหว่างที่ทำ doctoral work เป็นการออกแบบที่มีพื้นฐานอยู่กับนิ้วมือ 3 นิ้วที่แต่ละนิ้วมี 3 องศาอิสระ โดยที่ข้อต่อปลายของนิ้วมีความสามารถในการเคลื่อนที่ในขอบเขตที่มากกว่าปลายนิ้วของมนุษย์ ซึ่งเป็นการชดเชยส่วนที่ขาดของนิ้วมือมนุษย์



#### Utah/MIT Hand

ถูกออกแบบขึ้นในช่วงกลางของทศวรรษที่ 1980 เป็นมือที่ประกอบด้วยนิ้วมือ ซึ่งแต่ละนิ้วมือจะมี 4 องศาอิสระ และถูกขับโดยชุดของมอเตอร์และเส้นเอ็น การออกแบบนี้มีความซับซ้อนในการควบคุมมือสูง และมีข้อจำกัดในเรื่องความไว้วางใจได้ของระบบเนื่องจากความไม่สามารถคาดการณ์ความเสียหายของวัสดุที่ทำให้ทำเส้นเอ็นได้



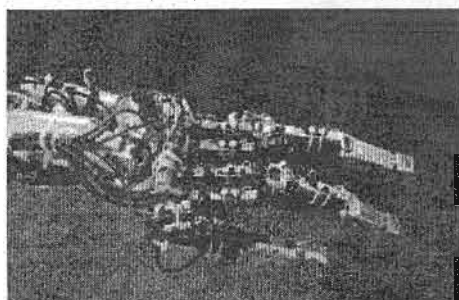
### Hirizinger Hand

เป็นมือที่ใหญ่กว่ามือของมนุษย์ประมาณ 2 เท่า มีความสามารถในการเคลื่อนที่เข้าถึงวัตถุได้ในทุกมุม ซึ่งเป็นสิ่งที่คล้ายคลึงกับมือของมนุษย์อย่างน่าทึ่ง มือชนิดนี้ประกอบด้วย sensor ที่ซับซ้อนจำนวนมากเรียงอยู่บนมือ คือ ใน 1 นิ้วประกอบด้วย sensor 28 ตัว และสามารถใช้กับระบบการมองเห็นภาพได้



### Robonaut Hand Conceptual Model

เป็นมือที่มีความคล้ายคลึงกับมือของมนุษย์ประมาณ 98 % ซึ่งประกอบด้วย 5 นิ้ว ที่มีองศาอิสระทั้งหมด 12 แห่ง มีการเรียงของ sensor ที่ไม่ซับซ้อนนัก โดยเน้นการติด sensor ไว้ที่ปลายนิ้วทั้ง 5 เพื่อป้องกันข้อมูลเกี่ยวกับการจับกลับเข้าสู่ระบบ



### Omni-Hand

เป็นมือหนึ่งในหลายๆมือของมือจับหุ่นยนต์แบบใหม่ เป็นมือที่คล้ายมือของมนุษย์มากที่สุดในปัจจุบัน ซึ่งเป็นการพัฒนาภายใต้สนธิสัญญาของนาซ่า มีการขับเคลื่อนเช่นเดียวกับ Robonaut hand และในรุ่นอนาคตจะทำการรวมเอาลักษณะของฝ่ามือเข้ามาไว้ที่มือหุ่นยนต์ด้วย โดยยังพยายามพัฒนาระบบ sensor และระบบขับเคลื่อนให้มีประสิทธิภาพสูงขึ้นต่อไป

โดยสรุปสามารถแบ่งระบบกลไกจับหรือมือกลได้ 2 ประเภทใหญ่ๆคือ กลไกแบบขับเคลื่อนด้วยมอเตอร์ (Gear Drive Motor) เช่น traditional robot gripper และ กลไกแบบขับเคลื่อนโดยอาศัยตัวส่งกำลังเช่น เส้นเอ็น ซึ่งกลไกการขับเคลื่อนมีจุดเด่นคือ สามารถสร้างปลายมือกล(นิ้ว) ให้มีขนาดเล็กและมีหลายข้อต่อได้โดยอาศัยการส่งกำลังผ่านเส้นเอ็นซึ่งจะได้อธิบายแนวความคิดของกลไกประเภทนี้โดยยกมือประเภท Soft gripper ที่เป็นกลไกการขับเคลื่อนที่อ่อนนุ่มมาอธิบาย

### Soft gripper

Soft gripper เป็นกลไกการจับทางอ้อมประเภทหนึ่งที่เป็นที่นิยมโดยเลียนแบบโครงสร้างที่พบในเส้นเอ็นในนิ้วของมนุษย์ จะเห็นได้ว่ากลไกแบบนี้สามารถสร้างแรงจับหรือบีบได้โดยการดึงเส้นเอ็นหรือเส้นลวดทำให้เกิดโมเมนต์

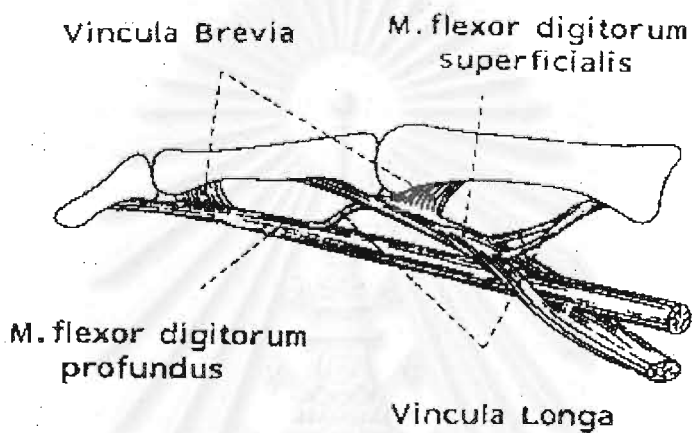


Fig 2.1-2 Bone and tendon system of the finger

(This is Composed of the Same Coupled Drive System as the CT Arm)

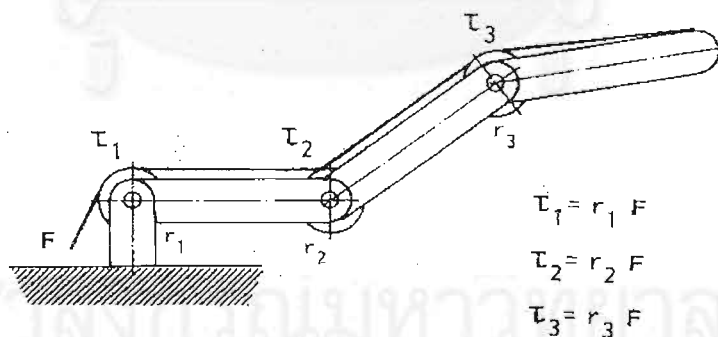


Fig 2.1-3 Soft gripper mechanism

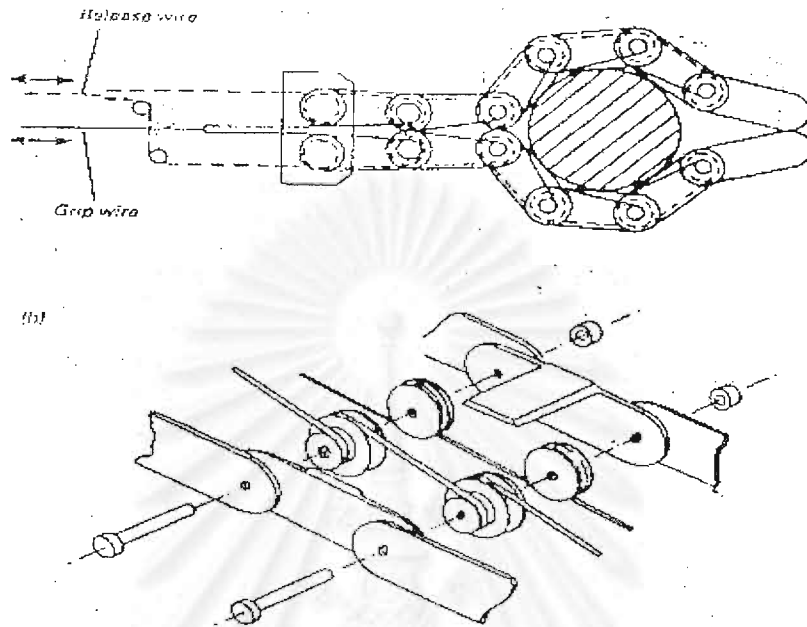


Fig 2.1-4 โครงสร้างของ Soft gripper

กลไกของ gripper ประกอบด้วยข้อต่อและรอกหลายชุด ที่ต่อกันแบบให้หมุนได้อย่างอิสระต่อกัน โดยที่ยึดข้อต่อแรกและพันลวดต่อไปยังรอกตัวอื่นๆ เมื่อดึงปลายลวด โมเมนต์บิดที่เกิดขึ้น จะแปรผันตามรัศมีของรอกของแต่ละข้อต่อ จึงทำการออกแบบให้รอกมีขนาดเรียงตามลำดับจากเล็กไปใหญ่ ส่วนกลไกการปล่อยของ gripper คือ การใส่ชุดของรอกที่มีขนาดเดียวกันทุกตัว แล้วทำการดึงในทิศทางตรงกันข้าม โดยอุปกรณ์ทั้งหมดจะถูกจัดให้อยู่ในรูปของขดลวด

### 2.1.1.2 การวิเคราะห์และออกแบบกลไกการจับ

เมื่อศึกษาและเก็บข้อมูลมือแบบต่างๆ เพื่อพิจารณาถึงข้อดี ข้อเสีย และความเหมาะสมกับการนำไปใช้งานในขอบเขตแล้ว จึงได้เลือกสร้างแบบจำลองของมือแบบ Soft gripper จากการสร้างแบบจำลองทำให้พบความไม่เหมาะสมและข้อเสียของแบบ Soft gripper มากขึ้น จึงได้ทดลองแบบใหม่คือแบบ Gear drive gripper เพื่อทำการศึกษาให้ละเอียดยิ่งขึ้น พบว่ามือกลแบบ Gear drive gripper เป็นแบบที่เหมาะสมกับโครงงานนี้เพราะมีกลไกที่ไม่ซับซ้อนและเหมาะสมกับงานในการหยิบหรือจับวัตถุ ซึ่งต่อจากนี้จะได้อธิบายถึงการพัฒนามือในโครงงานนี้ตามลำดับ

### Conceptual Design

กลไกการจับ (มือ) เป็นอุปกรณ์สำคัญของระบบทางกล (hardware) เนื่องจากเป็นส่วนที่ใช้หยิบจับสิ่งที่เป็นเป้าหมายของโปรแกรม (target) มีการกำหนดขอบเขตของสิ่งที่ต้องคำนึงถึงในการออกแบบมือที่จะใช้ ดังนี้

- ให้มีจุดสัมผัสระหว่างวัตถุกับมืออย่างน้อย 3 จุดเพื่อให้จับวัตถุได้อย่างมั่นคง
- ต้องแข็งแรงและมีแรงบีบมากเพียงพอที่จะยกวัตถุประมาณ 0.25 กิโลกรัมได้
- สามารถจับวัตถุได้หลากหลายรูปทรง โดยเน้นไปที่วัตถุทรงกระบอกที่วางตั้งไว้ (ในที่นี้ใช้กระป๋องในการปฏิบัติ)
- สามารถหยิบจับวัตถุได้โดยไม่ทำลายหรือสร้างความเสียหายแก่วัตถุนั้น
- ต้องมีขนาดเหมาะสมกับการนำไปใช้งาน
- มีน้ำหนักเบาพอสมควร เพื่อไม่สร้างภาระให้กับมอเตอร์ขับเคลื่อนมากเกินไป
- มีระบบขับเคลื่อนในตัว เพื่อความสะดวกในการติดตั้งกับอุปกรณ์แบบอื่นได้ ในที่นี้ต้องการนำไปติดกับส่วนปลายของหุ่นปฏิบัติการในห้องทดลอง (CRS robot)
- สามารถติดตั้งอุปกรณ์ตรวจรู้ (sensor) เพื่อนำไปพัฒนาความสามารถในการปฏิบัติการต่าง ๆ ได้เช่น ตัวตรวจวัดแรงที่ใช้จับวัตถุ, ตัวตรวจการลื่นไถลในแนวตั้ง (vertical slip sensor) เป็นต้น
- ความแข็งแรงของตัวมือต้องมีมากพอ เนื่องจากเป็นส่วนที่รับน้ำหนักจากวัตถุที่ยก (load) โดยตรง
- ต้องสามารถหยิบวัตถุที่ต้องการแยกออกจากสิ่งอื่น ๆ ที่วางรวมกันอยู่ได้
- มีความยืดหยุ่นในการทำงาน สามารถรับความคลาดเคลื่อนที่เกิดจากการเคลื่อนที่ของหุ่นได้

## ลักษณะมือต่างๆที่นำมาพิจารณา

แบบ	ลักษณะเด่น	ข้อดี	ข้อเสีย
<ul style="list-style-type: none"> <li>• ดึงขยะ กับที่โกย ผง</li> </ul> 	1.ประกอบไปด้วยที่กวาด วัตถุและดึงรองรับ	1.เมื่อหยิบวัตถุขึ้นมาได้ แล้ววัตถุจะไม่ตกลง ไปอีก 2.สามารถเผื่อความผิดพลาดได้ค่อนข้าง มาก	1.ไม่สามารถจัดวางวัตถุ ในลักษณะตำแหน่งที่ ต้องการได้ 2.ไม่สามารถเก็บวัตถุแยก เป็นชิ้นๆจากกองวัตถุ
<ul style="list-style-type: none"> <li>• แบบถุง</li> </ul>	1.หลักการทำงานคล้ายๆ กับตาข่ายจับแมลงคือ ใช้ครอบลงไปบนวัตถุ จากนั้นจึงปิดปากถุง เพื่อไม่ให้ของตกลงมา	1.น้ำหนักเบา 2.สามารถเผื่อความผิดพลาดได้ค่อนข้าง มาก	1.ไม่สามารถจัดวางวัตถุ ในลักษณะตำแหน่งที่ ต้องการได้ 2.ไม่สามารถเก็บวัตถุแยก เป็นชิ้นๆจากกองวัตถุ
<ul style="list-style-type: none"> <li>• Soft Fingers (Tendon driven)</li> </ul>	1.เลียนแบบการเคลื่อนไหวที่ เหมือนนิ้วคน คือมี ลักษณะเป็นข้อๆ 2.เป็นแบบใหม่ ที่กำลังมีการพัฒนาในวงการ วิจัยแบบนี้ได้เลือกมา ทดลองจึงจะอธิบาย ละเอียดอีกครั้ง	1.สามารถปรับเปลี่ยน รูปร่างของนิ้วได้ตาม ลักษณะของวัตถุที่จับ จึงสามารถจับวัตถุได้ หลายรูปทรง 2.มีจุดสัมผัสกับวัตถุ หลายจุดจึงสามารถ จับได้มั่นคง	1.แรงบีบน้อยเมื่อเทียบกับกำลังงานมอเตอร์ที่ ออกมาจากมอเตอร์ 2.ขั้นตอนการประกอบยุ่ง ยาก ซับซ้อน 3.เมื่อใช้ไปหลายๆครั้งจะทำให้ นิ้วกลับสู่สภาพ ปกติได้ยาก นั่นคือนิ้ว จะหงิกงอทั้งเวลาทาง ออก และบีบเข้า
<ul style="list-style-type: none"> <li>• แบบคีมจับชิ้น งาน (Gear drive gripper)</li> </ul>	1.ใช้ DC motor ขับ นิ้วทั้ง2ข้างให้เคลื่อนที่ เข้าหากันเพื่อเป็นการ จับวัตถุแบบนี้นำมาใช้ จริง จึงจะกล่าวละเอียด อีกครั้ง	1.ง่ายต่อการคำนวณ และขึ้นรูป 2.กางออกได้กว้างทำให้สามารถเผื่อความ ผิดพลาดได้ค่อนข้าง มาก 3.สามารถทำงานได้ ตามขอบเขตที่กำหนดไว้ข้างต้นได้	1.น้ำหนักมาก

จากขอบเขตความสามารถของมือที่กำหนดไว้ใน conceptual design จึงสามารถเลือกแบบมือจับมาทำแบบจำลอง คือ แบบ Soft gripper

#### การพัฒนา Soft Gripper

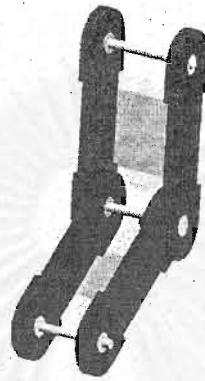


Fig 2.1-5 แบบข้อต่อ soft gripper

soft gripper เป็นกลไกการหยิบจับแบบใหม่ที่กำลังได้รับการสนับสนุน โดยเลียนแบบการทำงานของนิ้วคน กลไกประกอบด้วยข้อต่อจำนวนมากและชุดรอกที่คล้องเส้นเอ็น ข้อต่อจำนวนมากนี้ทำให้สามารถปรับตัวเองเข้าตามลักษณะของวัตถุที่จะจับจึงสามารถสัมผัสวัตถุได้หลายจุด และจากวิธีการคล้องเส้นเอ็น ทำให้แต่ละจุดสัมผัสออกแรงบีบได้เท่าๆกัน การเคลื่อนที่จะถูก บังคับโดยรอกและเอ็น 2 ชุด คือ ชุดบังคับให้บีบจับวัตถุ (บังคับนิ้วบีบเข้า) และ ชุดสำหรับคลายวัตถุ(บังคับนิ้วคลายตัวออก)

#### กระบวนการสร้างและทดสอบคุณสมบัติ ของ soft gripper ( tendon driven )

- กำหนดลักษณะ

มี 2 นิ้ว และมีนิ้วละ 5 ข้อ ข้อละ 5 cm.

ใช้เอ็นและรอกในการขับเคลื่อน

- การทดสอบความเป็นไปได้

สร้างแบบจำลองเพื่อดูลักษณะของมือ โดยใช้กระดาษแข็งสร้างนิ้ว มี 3 ข้อ ข้อละประมาณ 5 cm. และใช้เอ็นตกลาพันกับรอกเพื่อเป็นข้อนิ้ว

เมื่อได้แบบจำลองก็นำมาทดสอบโดยการใช้นิ้วมือดึงเพื่อดูลักษณะการเคลื่อนที่ของ soft gripper ซึ่งการเคลื่อนที่เป็นของแบบจำลอง ไปตามที่คาดการณ์เอาไว้ จึงทำการสร้างด้วยวัสดุที่ต้องการใช้จริงต่อไป

- ทำการสร้างด้วยวัสดุจริง

พิจารณาเลือกวัสดุที่ต้องการ คือ

- อะคริลิค : เพื่อให้มีน้ำหนักเบา ทำเป็นนิ้วแต่ละข้อ ตามจำนวนที่ต้องการ
- เอ็นตกลาขนาดรับน้ำหนักได้ 20-กิโลกรัม
- รอกพลาสติกในการขับเคลื่อนนิ้ว



- เหลาะอะลูมิเนียม 8 เหลี่ยมขนาด 8 mm.

แล้วจึงประกอบนิ้วตามลักษณะที่ต้องการ ดังรูปที่ 2.1-4 แล้วจึงนำมาทดลองใช้งานจริง ได้ผลการเคลื่อนที่ เป็นดังนี้

#### การเคลื่อนที่ของ soft gripper

- การบีบเข้า

เริ่มต้นจากการตั้งให้นิ้วอยู่ในตำแหน่งที่อ้าออกโดยมีวัตถุทรงกระบอกวางตั้งอยู่ตรงกลาง จากนั้นทำการดึงเอ็นในส่วนที่ควบคุมการบีบเข้าและปล่อยให้เอ็นในส่วนการควบคุมการคลายออกอิสระ พบว่า

- การเคลื่อนที่จะเริ่มจากนิ้วข้อต่อแรก (ส่วนที่ใกล้กับข้อมือมากที่สุด) เคลื่อนที่ก่อนจนกระทั่งชนกับสิ่งกีดขวาง(วัตถุที่ต้องการหยิบหรือ นิ้วส่วนแรกของอีกด้านหนึ่ง) ก็จะหยุด
- จากนั้นนิ้วข้อต่อข้อถัดมาก็จะเคลื่อนที่เข้าไปในลักษณะเดียวกันกับข้อต่อแรก และข้อต่อต่อไปก็จะเคลื่อนที่ได้ในลักษณะเดียวกัน
- จากการเคลื่อนที่ดังกล่าวทำให้นิ้วทั้งสองแนบกับรูปวัตถุที่ต้องการหยิบ
- ทำให้สามารถหยิบจับวัตถุรูปทรงใดก็ได้ โดยมีขนาดไม่ใหญ่จนเกินไป และพอเหมาะกับขนาดของข้อต่อของนิ้ว

- การคลายออก

หลังจากการบีบเข้าของนิ้วแล้วทำการบังคับให้อ้าออก โดยการปล่อยให้เอ็นควบคุมการบีบเข้าอิสระและดึงเอ็นควบคุมการคลายออก จะพบว่า

- ข้อต่อในส่วนปลายสุดจะเริ่มคลายออก จะคลายออกมาได้ระยะหนึ่งแล้วข้อต่อในส่วนถัดมาก็จะคลายออกตาม
- การเคลื่อนที่ดังกล่าวเกิดขึ้นเร็วจนเสมือนเกิดการเคลื่อนที่ออกพร้อมกันเกือบทุกข้อต่อ
- ลักษณะของนิ้วในหลังจากหยุดการบังคับการคลายจะมีลักษณะแตกต่างกันในแต่ละครั้ง ไม่สามารถควบคุมได้ และไม่อยู่ในตำแหน่งลักษณะเช่นเดียวกับตอนเริ่มต้น ทำให้ไม่สามารถควบคุมการบีบเข้าได้เป็นครั้งที่สองได้ติดกัน

จากการทดลอง จึงสามารถสรุป ข้อดี - ข้อเสีย โดยพิจารณาจากขอบเขตความสามารถของมือที่ต้องการดังที่ได้กล่าวไว้ใน conceptual design ของแบบ soft gripper ที่ทำจากอะคลิลิค ได้ดังนี้

- ข้อดี

- ทำจากวัสดุที่มีน้ำหนักน้อยและใช้เอ็นเป็นตัวถ่ายกำลังจากมอเตอร์ มือที่ได้จึงมีน้ำหนักน้อย ไม่สร้างภาระให้กับส่วนแขนมากเกินไป
- จากการที่มีข้อต่อจำนวนมาก ทำให้สามารถหยิบจับวัตถุรูปทรงใดก็ได้ โดยมีขนาดไม่ใหญ่จนเกินไป และพอเหมาะกับขนาดของข้อต่อของนิ้ว
- สามารถควบคุมการทำงานได้จากระยะไกล โดยการโยงสายเอ็นที่ใช้ในการควบคุมต่างๆ มาที่มอเตอร์ มอเตอร์จึงไม่จำเป็นต้องอยู่บนตัวมือ ทำให้สามารถลดน้ำหนักของตัวมือลงได้
- มีระบบขับเคลื่อนในตัว เพื่อความสะดวกในการติดตั้งกับอุปกรณ์แบบอื่นได้
- สามารถติดตั้งอุปกรณ์ตรวจจับ (sensor) เพื่อนำไปพัฒนาความสามารถในการปฏิบัติการต่าง ๆ ได้

- ข้อเสีย

- ไม่สามารถควบคุมลักษณะสุดท้ายหลังการคลายได้ ทำให้ไม่สามารถควบคุมการบีบเข้าได้เป็นครั้งที่สอง
- มีการสูญเสียแรงที่ข้อต่อต่าง ๆ เป็นจำนวนมากทำให้นิ้วไม่มีแรงบีบ อัตราส่วนของแรงที่บีบได้กับแรงที่ออก (ดึงเอ็น) มีค่าน้อยมาก ทำให้ต้องออกแรงมากหากต้องการหยิบยกวัตถุ ซึ่งอาจทำให้รอกเสียหายได้ นอกจากนี้การให้มอเตอร์ออกแรงดึงมากๆ ค้างอยู่เป็นเวลานาน อาจทำให้นิ้วมอเตอร์ไหม้ได้อีกด้วย
- การประกอบยุ่งยาก เนื่องจากใช้การพันเอ็นและอุปกรณ์ต่างๆ มีขนาดเล็ก ซึ่งยังทำให้การซ่อมแซมบำรุงรักษายุ่งยากตามไปด้วย หากมีจุดใดจุดหนึ่งชำรุด ต้องรื้อออกมาแก้ไขประกอบใหม่ทั้งหมด
- ระบบไม่ค่อยเสถียรมั่นคง โดยเอ็นที่พันสามารถคลายตัวหลุดออกจากรอกได้ทุกเมื่อหากปล่อยปละสายให้อยู่อย่างอิสระ
- นิ้วไม่สามารถรักษาตำแหน่งเดิมได้ เมื่อนำไปใช้งานแล้วจะไม่สามารถทำให้นิ้วกางออกให้อยู่ในสภาพเดิมได้ และการบีบตัวเข้าครั้งหลังๆ ก็จะไม่ได้อุปกรณ์ตามที่ต้องการ นิ้วจะเริ่มหงิกงอไม่ได้รูป

ข้อเสียต่าง ๆ ของ soft finger ที่พบนี้ไม่สามารถยอมรับได้ การแก้ไขจะทำให้เพิ่มความยุ่งยาก ซับซ้อนในการทำงานมาก อย่างเช่น ปัญหาที่เกี่ยวกับการรับภาระที่มือไม่สามารถรับได้เท่าที่ต้องการ แนวทางแก้ไขก็อาจจะเพิ่มกำลังของมอเตอร์และความแข็งแรงของรอกแต่จะต้องใช้มอเตอร์ที่ตัวใหญ่มากทำให้การสร้างมีความยุ่งยาก การเพิ่มความแข็งแรงของรอกก็อาจทำได้โดย

เพิ่มขนาดรอกให้มีพื้นที่รับแรงมากขึ้นหรือเปลี่ยนวัสดุ ซึ่งก็จะทำให้มือมีขนาดใหญ่และหนักขึ้น ซึ่งการปรับปรุงทั้งหมดนี้อาจยังไม่สามารถเพิ่มแรงบีบให้ได้เท่าที่ต้องการก็ได้

นอกจากนี้บางปัญหาเช่นปัญหานิ้วไม่สามารถรักษาท่าแหน่งได้ เป็นปัญหาเนื่องจากตัวแบบคือลักษณะการพันเอ็นไม่เหมาะสม การแก้ไข

เมื่อพิจารณาถึงข้อเสียและความไม่เหมาะสมแล้ว จึงเห็นว่าควรเปลี่ยนไปสร้าง gripper ชนิดอื่นที่มีคุณสมบัติเหมาะสมตามขอบเขตความสามารถมากกว่าแทน ดังนั้นจึงเริ่มต้นกระบวนการสร้างและทดสอบคุณสมบัติของ Gear drive gripper (Revolute joint gripper) ต่อไป

#### การพัฒนา Gear drive gripper

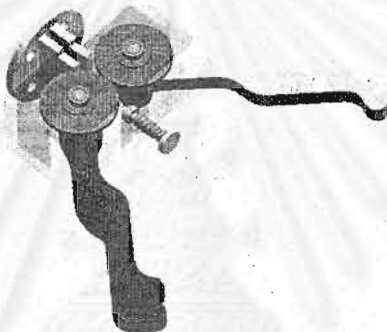


Fig 2.1-6 ลักษณะนิ้วและมือก่อนปรับปรุง (แบบ)

#### การออกแบบนิ้วและลักษณะการขั้วนิ้ว

เป็นการหยิบสิ่งของโดยมีจุดสัมผัส 3 จุด (Triple contacts) คือ ที่นิ้วแต่ละข้าง 2 จุด และที่สปริงตรงกลาง 1 จุด

#### Conceptual Design

- นิ้วที่ออกมาควรจะได้งัดเพื่อให้การหยิบวัตถุสะดวกขึ้น
- เมื่อกางนิ้วออกควรมีระยะจากปลายห่างกันมากพอสมควร
- สามารถประกอบเข้ากับหุ่น CRS ได้

นิ้วข้างหนึ่งจะถูกขับเคลื่อนด้วย DC motor และส่งผ่านกำลังไปยังนิ้วอีกข้างโดยเฟือง

ตรง ซึ่งการเคลื่อนที่ของนิ้วทั้งสองนั้นจะสมมาตรกันตลอด

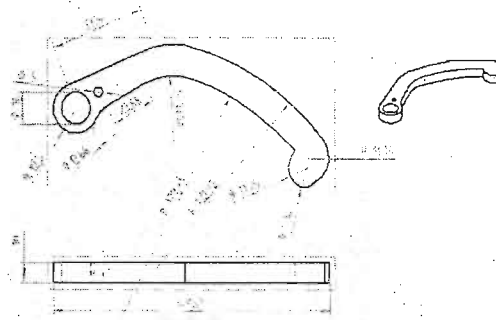


Fig 2.1-7 draft ของน๊ว

ทำการออกแบบด้วยชิ้นส่วนด้วย Solid edge version8 program ทำงานบนระบบ windows98 น๊วจะ ถูกออกแบบในสองลักษณะ คือ

- ในขั้นแรกให้มีลักษณะโค้งและเป็นเส้นตรง (รูปที่ 2.1-7) เพื่อให้สามารถจับวัตถุที่มีลักษณะเหลี่ยมแบนราบ และวัตถุทรงโค้งได้ แต่จะสามารถจับวัตถุได้มั่นคงเมื่อวัตถุอยู่ในตำแหน่งที่กำหนดเท่านั้น (วัตถุจะเคลื่อนที่ออกแม้จะอยู่ในพื้นที่การทำงานของน๊วแต่ออกห่างจากสปริงมากเกินไป)

- ในการออกแบบขั้นที่สองเน้นไปในการจับวัตถุทรงระบอก(กระป๋อง) จึงออกแบบให้มีส่วนโค้งทั้งหมด โดยมีโค้งสองลักษณะ คือในส่วนโค้งแรก(ใกล้จุดหมุน) จะเป็นส่วนในการจับยึดวัตถุแม้มีขนาดเล็กก็สามารถจับยึดได้ และส่วนโค้งอีกส่วนหนึ่งจะช่วยในการจับวัตถุที่ใหญ่ขึ้น และช่วยนำวัตถุให้เคลื่อนที่มาในตำแหน่งศูนย์กลางของน๊ว ทำให้สามารถจับวัตถุทรงระบอกได้ทุกตำแหน่งของวัตถุ(ที่อยู่ในพื้นที่การทำงานของน๊ว)

น๊วจะถูกแบ่งเป็น 2 ข้าง ข้างซ้ายมี 1 น๊ว และข้างขวาจะมี 2 น๊ว เพื่อให้เกิดสมดุลย์โมเมนต์เมื่อจับวัตถุ โดยน๊วด้านซ้ายจะสอดเข้าไประหว่างน๊วด้านขวาได้พอดี เมื่อน๊วมือเคลื่อนที่เข้าหากันตรงกลางทำการติดตั้งสปริง นอกจากเพื่อช่วยประคองวัตถุแล้ว สปริงยังเป็นตำแหน่งที่สามารถติดตั้งอุปกรณ์ตรวจวัดแรงบีบเพื่อพัฒนาความสามารถในการรับรู้ให้แก่หุ่นได้ในอนาคต

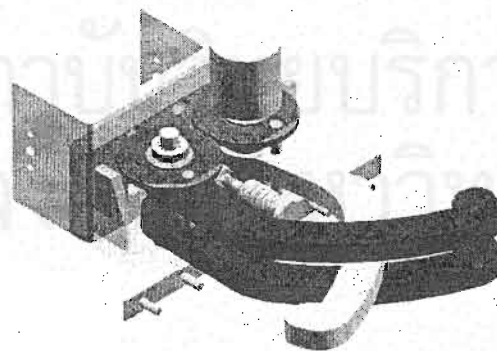


Fig 2.1-8 แบบมือที่ประกอบสมบูรณ์

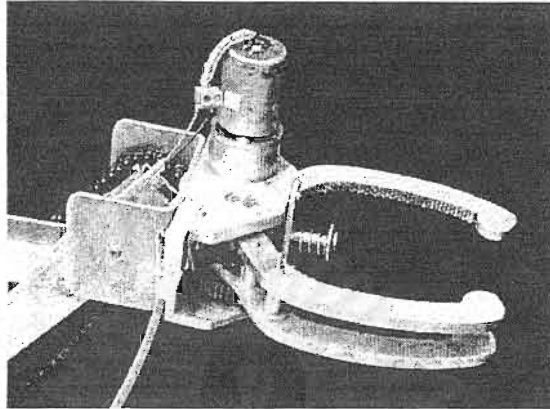


Fig 2.1-9 มือที่ประกอบสมบูรณ์ ( ภาพจากของจริง )

การออกแบบการจับยึดมือกับแขน



Fig 2.1-10 ข้อมือและนอตยึดแขน

ข้อมือเป็นส่วนเสริมเพื่อเชื่อมมือกับแขนเข้าด้วยกัน ข้อมือจะมีขนาดค่อนข้างสั้นเพื่อลดความยาวโดยรวมของแขนทำให้ลดภาระของมอเตอร์ที่ใช้ในการขับเคลื่อนแขน(ลดน้ำหนักและโมเมนต์) และสามารถประกอบเข้าใช้งานกับ CRS Robot ได้

## ขั้นตอนการสร้าง ( Manufacturing )

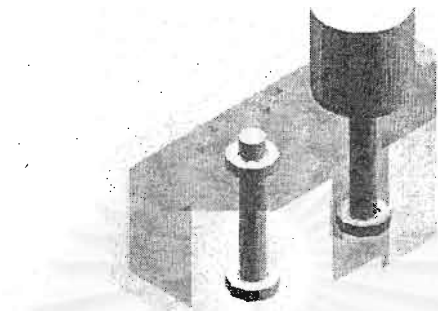


Fig 2.1-11 Housing ( ออกแบบ )

- สร้างแบบ โดยใช้ส่วนประกอบของแบบจำลอง เป็นไม้ทุกชิ้น ยกเว้น เฟลาที่ใช้หมุนไปกับเฟืองทำจากเหล็ก พบว่า แบบไม้ไม่สามารถรองรับน้ำหนักของวัตถุที่มากได้และผลงานไม่สวยงามเท่าที่ควร มีความแข็งแรงไม่เพียงพอที่จะรองรับการแก้ไข ปรับปรุงบ่อยๆ ได้ จึงได้เปลี่ยนแบบมาเป็นอะลูมิเนียม
- การสร้างนี้ ทำโดยการใช้เครื่อง CNC ขึ้นรูปแผ่นอะลูมิเนียมความหนา 5 mm. โดยใช้รูปนี้จากการออกแบบ โดยใช้โปรแกรม Solid edge version 8(draft) นำมาสร้าง G-code (ดูภาคผนวก) ในโปรแกรมน้อย 2D-contour ของโปรแกรม AutoCAD นำนี้ที่ขึ้นรูปเรียบร้อยแล้ว ประกอบเข้ากับเฟลากลมขนาด 8 mm. เพื่อทำการประกอบกับเฟืองและมอเตอร์ต่อไป รวมทั้งทำการติดแผ่นยางที่ผิวสัมผัสด้านในของนี้ทั้งสอง เพื่อป้องกันการ ลื่นไถลลงของวัตถุที่จับกับนี้
- การสร้างและประกอบกรอบ (Housing)
 

กรอบมีไว้เพื่อติดตั้งเฟลา มอเตอร์และเชื่อมต่อกับแขน รวมทั้งเพิ่มความเป็นระเบียบเรียบร้อย ประกอบขึ้นจากอะลูมิเนียม จำนวน 3 ชั้น คือ แผ่นหลังใช้ยึดติดกับข้อมือกับ แผ่นบนและ แผ่นล่างที่เจาะรูขนาด 16 mm. ด้านละ 2 รูเพื่อใส่แบริ่งรองรับเฟลา ทำการยึดแบริ่งเข้ากับแผ่นอะลูมิเนียมด้วยการตอกอัด (interference fit) ยึดเหล็ก การขึ้นรูปกรอบ ทำจากอะลูมิเนียมแผ่นหนา 5 mm.โดยการใช้เครื่องเจาะ (Drilling Machine)

การประกอบแผ่นอะลูมิเนียมทั้ง 3 ชั้นเข้าด้วยกัน ทำได้โดยการใช้เหล็กฉากมายึดประกอบมุมทั้ง 2 มุม แล้วเจาะรูยึดเพื่อเพิ่มความแข็งแรงของโครงสร้าง

ข้อควรระวังในการเจาะรูเพื่อใส่แปรงรับเพลลา คือ จะต้องให้รูทั้งสองของทั้งด้านบนและด้านล่างตรงกัน มิฉะนั้น จะทำให้เพลลาเอียงและเกิดปัญหาในการประกอบเฟือง ซึ่งทำให้เกิดความคลาดเคลื่อนในการทำงานของมือได้

วิธีการเจาะรู: จะต้องเจาะรูทั้งแผ่นบนและแผ่นล่างไปพร้อมๆกัน และหากใช้ดอกสว่านขนาดใหญ่ เจาะลงไปเพียงครั้งเดียว จะทำให้เกิดความคลาดเคลื่อนได้ง่าย จึงควรใช้ดอกสว่านดอกเล็ก ประมาณ 5 mm. เจาะนำลงไปก่อน แล้วจึงขยายรูด้วยดอกสว่านที่มีขนาดใหญ่ตามลำดับ



Fig 2.1-12 การประกอบชุดเฟืองและเพลลาเข้ากับกรอบ (แบบ)

- เฟืองและเพลลา เป็นกลไกหลักในการขับเคลื่อนนิ้วมือให้เคลื่อนที่ การยึดเฟือง เพลลาและมอเตอร์เข้าด้วยกันอย่างมีประสิทธิภาพ จะช่วยลดความคลาดเคลื่อนในการเข้าถึงเป้าหมายได้มาก ซึ่งความคลาดเคลื่อนนี้ ไม่สามารถคำนวณหาค่าเพื่อชดเชยให้ถูกต้องได้ แกนมอเตอร์และแกนเพลลาถูกยึดเข้ากับหน้าแปลนชิ้นเดียวกัน ด้วยน็อตขนาด 3 mm. เพื่อให้แกนทั้งสองหมุนไปพร้อมกันขณะทำงาน และเพลลา กับเฟือง ยึดเข้าด้วยกัน โดยใช้สลัก แต่มีข้อควรระวัง คือ ต้องใช้สลักที่มีขนาดพอดีกับรูที่เจาะเอาไว้ เพื่อป้องกันการ

## 2.1.2 ระบบขับเคลื่อนแขนและข้อมือ

ระบบขับเคลื่อนแขนและข้อมือเป็นส่วนในการเคลื่อนย้ายมือให้สามารถขึ้นลงในแนวตั้งได้ ออกแบบให้มีแขนที่สามารถยกมือขึ้นลงได้ด้วยการหมุน (Revolute joint) และนอกจากนี้ยังมีข้อมือ (Wrist) ที่ทำให้มือมีการหมุนได้ (Rotation) ที่ปลายแขน นอกเหนือจากการเคลื่อนที่ขึ้นลง เพียงอย่างเดียว

### Conceptual Design

- แขนต้องมีโครงสร้างที่ไม่ซับซ้อน สามารถถอด - ประกอบและแก้ไขได้ง่ายเมื่อเกิดปัญหาผิดพลาด

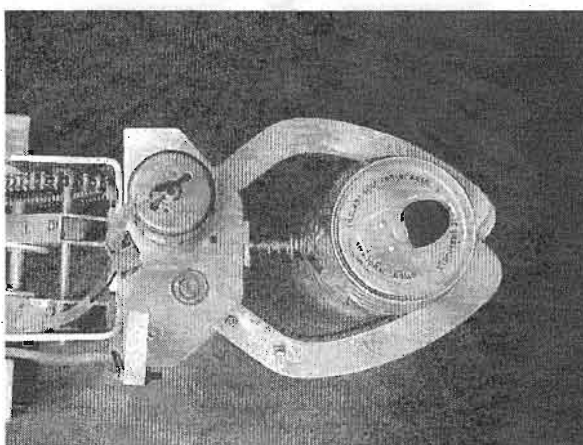


Fig 2.1-13 เปรียบเทียบขนาดที่เหมาะสม ระหว่างมือและกระป๋องที่

- ข้อมือควรมีความแข็งแรง และสามารถเคลื่อนที่ในทิศทางที่ต้องการได้อย่างสะดวก นอกจากนี้ข้อมือต้องสามารถรักษาระดับของตนเองให้ขนานกับพื้นได้
- พยายามให้เป็นโครงสร้างที่มีน้ำหนักเบาที่สุดเท่าที่เป็นไปได้ โดยต้องคำนึงตำแหน่งของจุดศูนย์ถ่วงและคงความแข็งแรงเท่าเดิม
- มีขนาดเหมาะสมกับงานที่ใช้ (ในที่นี้ คือใช้เคลื่อนที่ขึ้นลงเพื่อยีบสิ่งของ เช่น แก้วพลาสติก กระป๋อง เปลา เป็นต้น)
- ต้องประกอบด้วยชิ้นส่วนที่สามารถนำมือที่ได้จากการออกแบบ เคลื่อนที่ในแนวตั้งได้ สามารถยีบของจากพื้นและยกขึ้น - ลงได้
- มีความสวยงามและเป็นระเบียบเรียบร้อย

### ขั้นตอนการสร้าง

แขนเป็น four bar linkage วางตำแหน่งมอเตอร์ไว้ที่จุดหมุนซึ่งอยู่ปลายแขน

- ใช้อะลูมิเนียมกล่องขนาดเท่ากัน 2 อัน วางขนานกันโดยมีเฟลาเหล็กร้อยผ่านอะลูมิเนียมทั้งสอง



- ลักษณะแขนเป็น four bar linkage โดยมีมอเตอร์ควบคุม 2 ตัว ตัวที่ 1 ขับที่ปลายแขนทำหน้าที่ยก แขนขึ้น-ลง ตัวที่ 2 ทำหน้าที่ในการควบคุมการหมุนของมือ ในการเปลี่ยนรูปทรงของ four bar linkage ด้านสั้น ทำให้มือสามารถหมุนได้ที่ความสูงเท่าเดิม
- ลักษณะของ four bar linkage จะทำให้สามารถรักษาระดับของมือให้ขนานกับพื้นได้ตลอด โดยเมื่อ 2 link ใดๆขนานกันอยู่ก็จะทำให้อีก 2 link ที่เหลือขนานกันไปด้วย

จากการประกอบและทดลองใช้งาน พบข้อดีและข้อเสียของแขนแบบนี้ คือ

- ข้อดี
  - โครงสร้างมีความแข็งแรง
  - รูปทรงมีความสมมาตร ทำให้กระจายแรงได้ดี
- ข้อเสีย
  - เนื่องจากต้องใช้อะลูมิเนียม ในสร้าง four bar linkage ที่มีความยาวมาก ซึ่งประกอบมือไว้ทาง ปลายแขน น้ำหนักของมือก็มากเพราะทำจากอะลูมิเนียม จึงทำให้เกิดโมเมนต์ด้านการยกของ มอเตอร์สูง มอเตอร์ตัวที่ใช้ขับขึ้น-ลงไม่สามารถขับได้
  - การหมุนของเครื่องมือทำได้ในวงแคบ เพราะข้อจำกัดทางรูปทรง หากต้องการให้หมุนได้กว้างขึ้น ต้องให้ ชั้นที่เป็นด้านสั้นมีความยาวมากขึ้น มิฉะนั้นจะชนกับ link ที่อยู่ข้างๆ
  - การติดตั้งชุดแขนทำได้ยาก รวมทั้งการติดตั้ง มอเตอร์ก็มีความยุ่งยากด้วย เนื่องจากมอเตอร์ทั้ง 2 ตัวจะใช้แกนเดียวกันเป็นแกนหมุนเพื่อรักษาความเป็น four bar linkage

หลังจากพิจารณาถึงจุดด้อยและปัญหาที่เกิดขึ้นกับแขนที่ออกแบบมาแล้ว จึงมีการทำการปรับปรุงแบบ และระบบการทำงานของแขน ดังต่อไปนี้

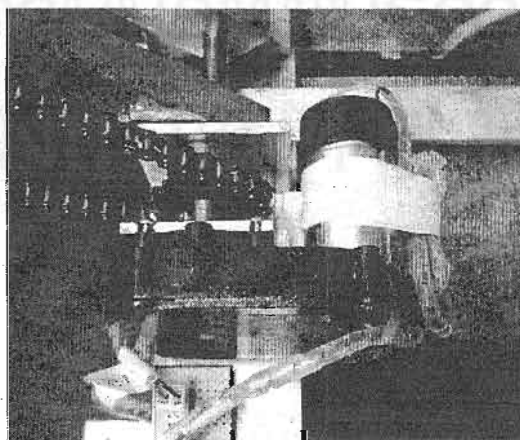


Fig 2.1-14 การใช้ชุดโซ่แทน four bar-linkage

การปรับปรุงครั้งที่ 1 : ใช้ชุดโซ่แทน four bar linkage วางตำแหน่งมอเตอร์ที่จุดหมุนซึ่งอยู่ปลายแขน

แก้ไขข้อเสียจากแบบแรก คือ

- ลดความยาวของแขนและลดน้ำหนักมือลง
- ใช้โซ่แทน four bar linkage ทำให้แก้ไขปัญหาที่ไม่สามารถหมุนข้อมือเป็นมุมกว้างๆได้

อย่างไรก็ตามโมเมนต์ที่ต้านการหมุนของมอเตอร์ ยังสูงมากอยู่ มอเตอร์ไม่สามารถขับได้ จึงต้องเปลี่ยนระบบการยกแขนใหม่

การปรับปรุงครั้งที่ 2 : ใช้มอเตอร์เป็นก้านในการยกแขน

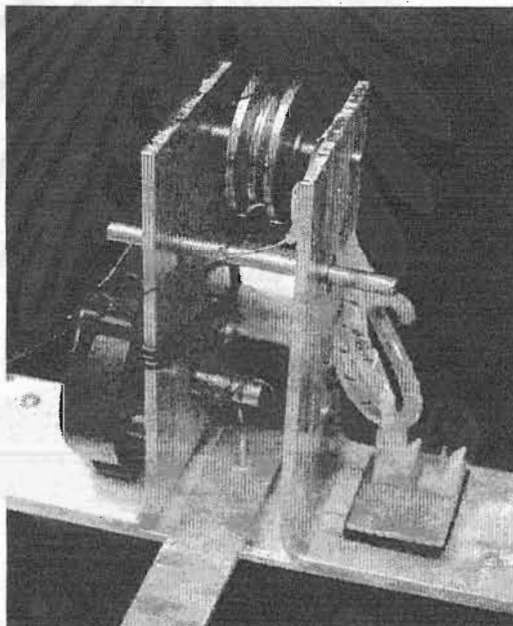


Fig 2.1-15 การใช้มอเตอร์เป็นก้าน

มีการออกแบบวิธีการยกแขนขึ้น-ลงใหม่ เป็นการใช้เส้นเอ็นร้อยที่เพลลาเหล็ก ที่อยู่ใกล้ข้อมือระหว่างแท่งอะลูมิเนียม แล้วใช้มอเตอร์เป็นก้านดึงแขนขึ้นลง

วิธีการใช้ก้านดึงนั้น สามารถยกแขนเมื่อติดมือขึ้นได้ แต่จำเป็นต้องหาตำแหน่งที่ร้อยเอ็นที่เหมาะสม เมื่อแขนเคลื่อนที่จนเส้นเอ็นวางตัวในแนวเดียวกันหรือใกล้เคียงกับแขน จะทำให้มอเตอร์มีกำลังไม่พอที่จะยกแขน แขนจึงไม่สามารถเคลื่อนที่ต่ำไปกว่านี้ได้

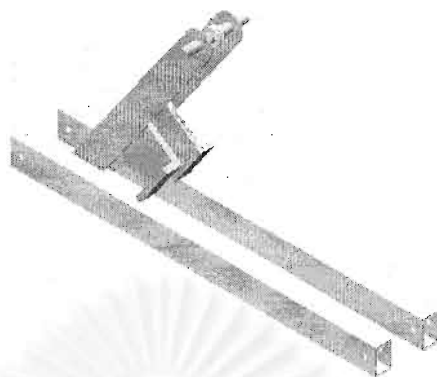


Fig 2.1-16 โครงสร้างป้องกันการวางตัวในแนวเดียวกันของเอ็นกับแขน ( ออกแบบ )

**การปรับปรุงครั้งที่ 3 :** ต่อโครงสร้างที่แขนเพื่อป้องกันการวางตัวในแนวเดียวกันของเอ็นกับแขน

เนื่องจากการที่เอ็นวางตัวในแนวเดียวกันหรือใกล้เคียงกับแขน จะทำให้มีแรงบิดไม่พอที่จะยกแขน จึงใช้การต่อโครงสร้างของแขนเพิ่มเติมเพื่อเปลี่ยนแนวการวางตัวของเส้นเอ็นมิให้อยู่ในแนวเดียวกันหรือใกล้เคียงกับแขน นอกจากนี้ยังได้เพิ่มรอกเข้าไปเพื่อให้มีแรงดึงในการยกมากขึ้นด้วย

เมื่อทำการเพิ่มโครงสร้างลงบนแขนพบว่าเส้นเอ็นไม่วางตัวในแนวเดียวกันหรือใกล้เคียงกับแขนตลอดการเคลื่อนที่ ทำให้แขนสามารถเคลื่อนที่ลงได้มากขึ้น โดยโครงสร้างที่เพิ่มขึ้นจะวางตัวเป็นเส้นตรงในแนวเดียวกับเอ็นที่ยึดแขนในตำแหน่งของแขนที่ยกตั้งขึ้นในระดับหนึ่ง และจะมีลักษณะเป็นวัตถุแข็งเกร็ง (rigid body) ยึดติดกับแขนเมื่อแขนก้มลงในมุมต่ำ ทำให้สามารถหยิบของที่วางบนพื้นได้โดยที่รอกไม่ล้มและไม่มีจุดตาย (singularity) เกิดขึ้นใน workspace

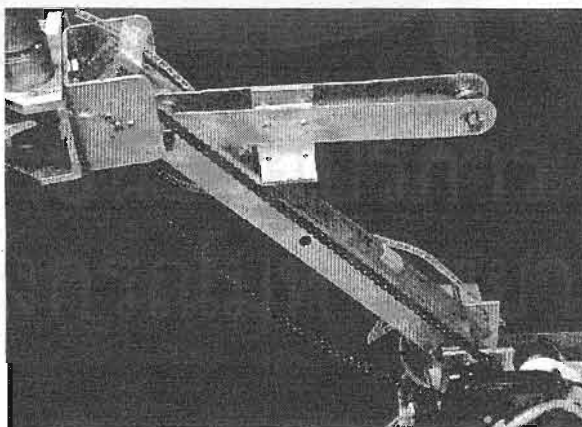


Fig 2.1-17 โครงสร้างป้องกันการวางตัวในแนวเดียวกันของเอ็นกับแขน (ภาพจากของจริง)

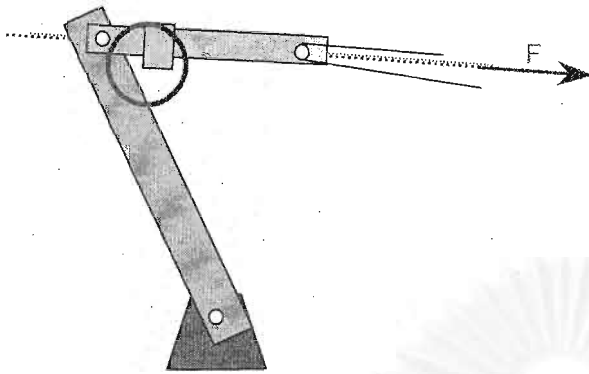


Fig 2.1-18 การดึงแขนในตำแหน่งที่ปลายแขนยก

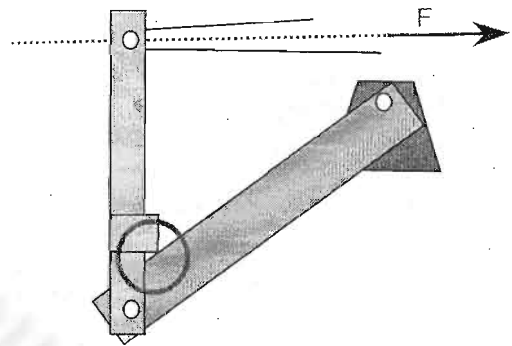


Fig 2.1-19 การดึงแขนในตำแหน่งที่ปลายแขนยกต่ำลง

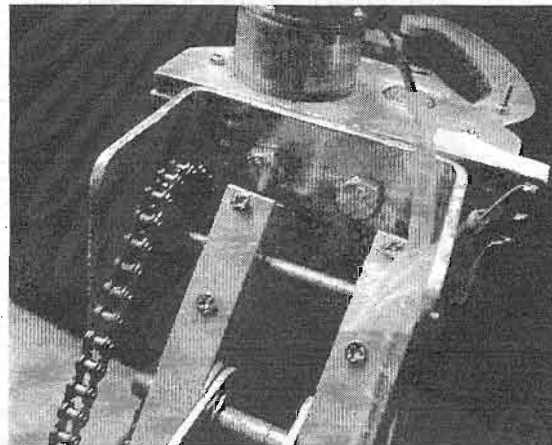


Fig 2.1-20 ระบบข้อมือ

ส่วนระบบข้อมือนั้นยังเหมือนเดิมคือ ใช้อะลูมิเนียมรูปตัว C 1 ชิ้นเป็นส่วนที่ใช้ประกบกับมือ โดยด้านในของอะลูมิเนียมจะมีเฟืองยึดติดอยู่กับอะลูมิเนียม และใช้มอเตอร์ที่มีเฟืองประกบที่เพลาลูกตัวหนึ่งขับผ่านโซ่ ไปยังเฟืองที่ข้อมือเพื่อให้ข้อมือสามารถหมุนได้ทั้งในทิศทางตามเข็มนาฬิกาและทวนเข็มนาฬิกา นอกจากนี้ยังให้โซ่ทำตัวเป็น four bar linkage จึงสามารถรักษาระดับของข้อมือให้สามารถขนานกับพื้นได้ตามต้องการ

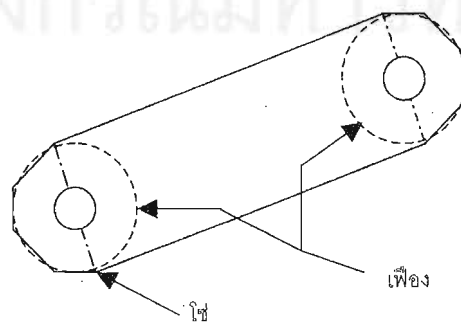


Fig 2.1-21 การติดตั้งระบบโซ่และเฟือง

เราทำการรักษาข้อมือให้อยู่ในแนวระดับ โดยการปรับข้อมือให้อยู่ในแนวระดับเสียก่อน จากนั้นทำการ ล็อคเฟืองที่จับข้อมือ เมื่อเราทำการยกแขนขึ้นเฟืองที่ข้อมือจะเคลื่อนที่หมุนไปยังโซ่ข้อถัดไปทำให้ข้อมือยังคงอยู่ใน แนวระดับตลอดเวลา ซึ่งเราสามารถเทียบกับรูปของ four bar linkage ได้ดังนี้

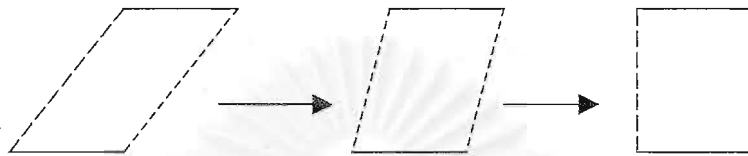


Fig 2.1-22 แนวการยกแขน แบบ four bar linkage

### 2.1.3 ระบบขับเคลื่อนล้อ และ โครงสร้างฐาน ( base )

ระบบขับเคลื่อนล้อและโครงสร้างฐาน ประกอบไปด้วย ล้อ และ มอเตอร์ขับเคลื่อน และ ในที่นี้ยังรวมไป ถึงการออกแบบโครงสร้างของตัวรถให้แข็งแรง และต้องมีขั้นตอนในการสร้างที่ทำให้ระบบขับเคลื่อนสามารถทำงาน ได้อย่างมีประสิทธิภาพ

#### Conceptual Design

- ตัวโครงรถต้องมีความแข็งแรงพอที่จะรองรับระบบทุกระบบของหุ่นยนต์ แต่ไม่ควรมีน้ำหนักมากจนเกินไป จนมอเตอร์ที่ใช้ขับเคลื่อนไม่สามารถทำงานได้
- โครงสร้างมีขนาดที่เหมาะสม กะทัดรัด แต่ต้อง มีที่ว่างเพียงพอสำหรับวางกล่องควบคุม มอเตอร์ และแบตเตอรี่ได้
- มีการประกอบที่ประณีต เข้ามุมและได้ฉาก เนื่องจากจะส่งผลต่อระบบขับเคลื่อน
- ความสูงพอเหมาะ เพราะต้องหยิบของจากพื้น และต้องคำนึงถึงจุดศูนย์ถ่วง
- ตัวรถควรมีความสวยงาม เป็นระเบียบและคงทน รวมทั้งสามารถถอด - ประกอบได้สะดวก

#### ขั้นตอนการสร้าง

ในการสร้างโครงฐานล้อ เริ่มจากการตัดอะลูมิเนียมฉากขนาด 5 cm. หนา 4 mm. ออกเป็นขนาดดังต่อไปนี้

นี้

- ขนาดยาว 40 cm. จำนวน 2 ชิ้น พร้อมเจาะรูขนาด 8 mm. ที่ระยะห่างจากขอบด้านหน้า 17 cm. และ สูงจากขอบล่างในแนวตั้ง 1.5 cm.
- ขนาดยาว 30 cm. จำนวน 1 ชิ้น
- ขนาดยาว 26.2 cm. จำนวน 1 ชิ้น

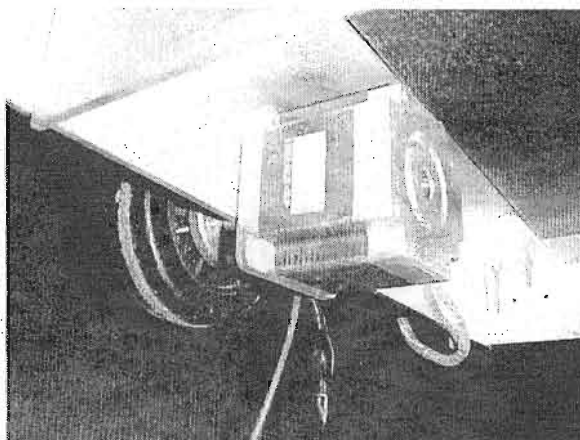


Fig 2.1-23 การติดตั้ง stepping motor เพื่อเป็นต้นกำลังในการขับเคลื่อน

จากนั้นจึงนำมาประกอบกันเป็นฐานรูปสี่เหลี่ยม โดยให้อะลูมิเนียมขนาด 40 cm. 2 ชั้นวางขนานกันเพื่อเป็นด้านข้าง แล้วนำอะลูมิเนียมขนาด 26.2 cm. มาสอดเข้าด้านหน้าห่างจากขอบหน้า 6.2 cm. เพื่อเป็นตัวกำหนดระยะห่างที่เป็นระยะขนานของอะลูมิเนียมขึ้นด้านข้างทั้งสอง สุดท้ายจึงนำอะลูมิเนียมยาว 30 cm. มาวางทาบด้านบนของอะลูมิเนียมด้านข้างทั้งสองอัน จากนั้นทำการเจาะรูยึดน็อต อะลูมิเนียมทั้งหมดเข้าด้วยกัน และทำการเจาะรูเพื่อประกอบล้อ เพลาและ stepping motor ทั้ง 2 ด้าน



Fig 2.1-24 ฐานรถประกอบล้อ (แบบ)

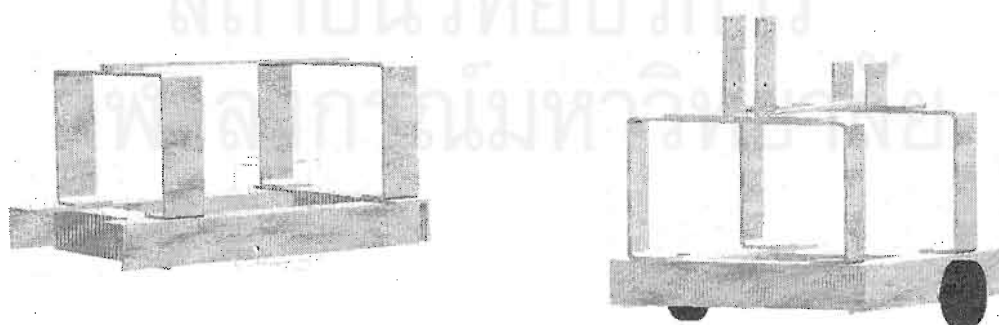


Fig 2.1-25 ฐานรถยึดโครงรูปตัว C (แบบ) Fig 2.1-26 ฐานรถประกอบล้อ ยึดโครง รูปตัว C ที่ยึดแขนและก้าน (แบบ)

จากการประกอบฐาน พบว่า ฐานที่ประกอบเสร็จแล้วมีความสูงจากพื้นไม่มากนัก ทำให้มีความยากลำบากในการหยิบจับวัตถุที่อยู่สูง จึงทำการแก้ไขโดยการเจาะรูขั้วน็อตยึดอะลูมิเนียมที่มีลักษณะคล้ายตัว C ที่ตำแหน่งตรงกับอะลูมิเนียมด้านหน้า (ขนาดยาว 26.2 cm.) และอะลูมิเนียมด้านหลัง (ขนาดยาว 30 cm.) จากการติดโครงสร้างเพิ่ม จะทำให้ตำแหน่งยึดจุดหมุนของแขนอยู่สูงจากพื้น 16 cm. แต่ปรากฏว่าโครงสร้างที่ติดเพิ่มเติมจะรับแรงกดหรือแรงดึงในแนวหน้า-หลังได้ไม่ดี จึงนำอะลูมิเนียมแผ่นขนาดกว้าง 2 cm. ยาว 33.8 cm. หนา 4 mm. มายึดระหว่างโครงสร้างที่อยู่ด้านหน้าและหลังที่ทำการเพิ่มเติม เพื่อให้โครงสร้างมีความแข็งแรงมากขึ้น เมื่อทำการประกอบล้อ โครงรูปตัว C ที่ยึดแขนและที่ยึดคว้านเข้ากับ ฐานรถ จะได้โครงรถดังภาพ

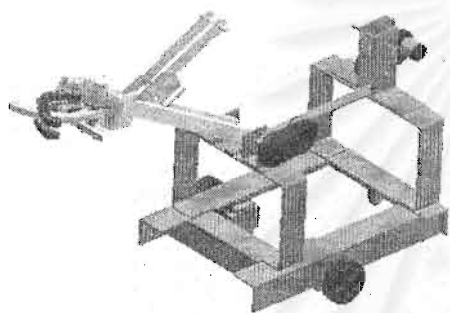


Fig 2.1-27 โครงรถสมบูร์น (แบบ)

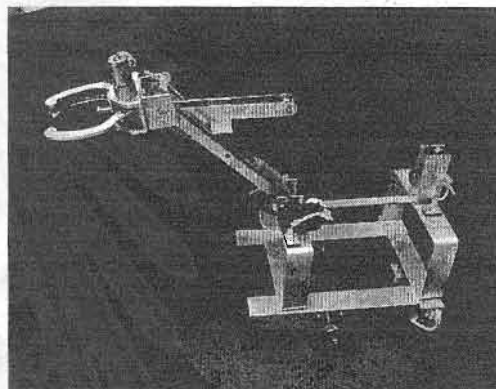


Fig 2.1-28 โครงรถสมบูร์น (ภาพจากของจริง)

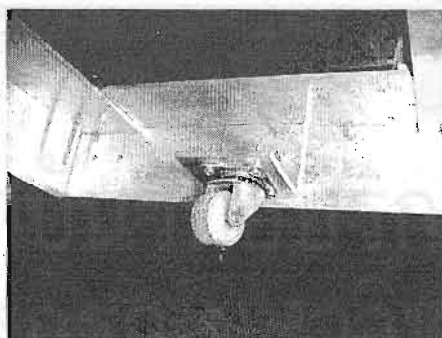


Fig 2.1-29 แสดงล้อหลังขนาดเล็กเพื่อรักษา

หลังจากประกอบโครงรถเรียบร้อยแล้ว จึงนำแขนและมือมาประกอบรวมกัน จะได้โครงรถสมบูร์น ดังภาพ เพื่อเพิ่มความสมดุลย์ของโครงหุ่นขณะเคลื่อนที่ จึงติดตั้งล้อหลังขนาดเล็กที่ขอบหลังของโครงสร้างฐาน โดยต้องวางตำแหน่งให้มีระนาบสัมผัสพื้นเดียวกับล้อรถทั้งสองพอดี ซึ่งเป็นสิ่งที่ทำได้ยาก จึงอาจอนุโลมให้ติดตั้งล้อหลังที่มีระนาบสัมผัส สูงกว่าระนาบของล้อคู่หน้าเล็กน้อย เพราะถ้าระนาบสัมผัสของล้อหลังต่ำกว่าแล้ว จะทำให้เกิด ปัญหาการขับเคลื่อนได้ไม่เต็มที่ เนื่องจากล้อคู่หน้าสัมผัสพื้นได้ไม่เต็มที่

**ข้อควรระวัง :** ในการประกอบล้อจะต้องวางตำแหน่งล้อให้ตรงกัน และทำให้ล้อยึดอย่างมั่นคงบนฐานเพราะความคลาดเคลื่อนระหว่างล้อกับแกนเพล่าไม่สามารถตรวจสอบด้วยโปรแกรม จึงไม่สามารถประมาณความผิดพลาดเนื่องจากความไม่เป็นวัตถุแข็งแข็งเกร็งขึ้นเดียวกันได้เลย

การทำให้ตำแหน่งล้อตรงกันจะทำให้สะดวกในการโปรแกรมมากขึ้นเพราะเมื่อโปรแกรมให้หุ่นเลี้ยวหรือหมุน หุ่นจะเคลื่อนที่รอบจุดศูนย์กลางการเลี้ยว นั่นคือจุดตรงกลางระหว่างล้อทั้งสอง



จุฬาลงกรณ์มหาวิทยาลัย



## 2.2 ระบบไฟฟ้าและอุปกรณ์รับรู้

เนื่องจากระบบหุ่นยนต์ มีระบบขับเคลื่อนด้วย stepping motor ในส่วนการเคลื่อนที่ และใช้ DC motor ในการขับเคลื่อนกลไก แขน ข้อมือ และมือจับ ดังนั้นในส่วนของระบบไฟฟ้าจึงมี วงจรขับ stepping motor วงจรขับ DC motor และรวมถึงในส่วนของ controller ซึ่งในหุ่นดังกล่าวเลือกใช้ microcontroller ตระกูล AVR เบอร์ 90S8535 ของ Atmel

### 2.2.1 Stepping Motor

การทำงานของ stepping motor เริ่มจากเมื่อป้อนไฟฟ้าให้กับ motor จะทำให้หมุนเพียงเล็กน้อยตามเส้นรอบวงและหยุด ซึ่งต่าง จาก motor ทั่วไปที่จะหมุนตลอดเวลาเมื่อป้อนแรงดันไฟฟ้า

ข้อดีของ stepping motor คือ สามารถกำหนด ตำแหน่งของการหมุนตาม องศาหรือระยะทาง ได้อย่างละเอียดโดยใช้คอมพิวเตอร์หรือ microcontroller เป็น เครื่องกำหนด

โครงสร้างของขั้วแม่เหล็กบนสเตเตอร์ ทำมาจากแผ่นเหล็กวงแหวนที่มีซี่ยื่นออกมาประกอบกันเป็นชั้นๆ โดยที่แต่ละซี่นั้นจะมีคอยล์(ขดลวด)พันสวมอยู่ เมื่อมีการป้อนกระแสผ่านคอยล์ ทำให้เกิดสนามแม่เหล็กไฟฟ้า (Electromagnetic) ดังรูปด้านล่างนี้ จะแสดงถึงองค์ประกอบที่กล่าวมา

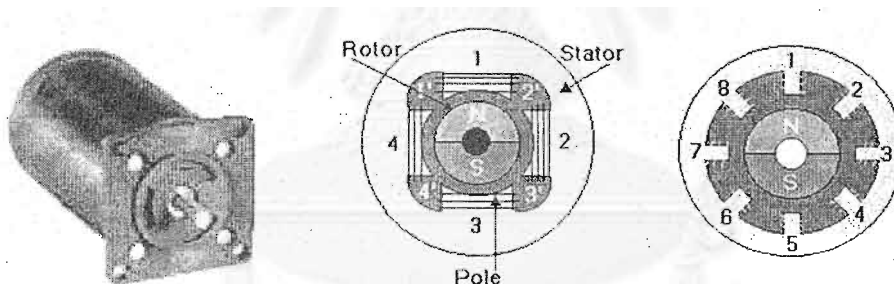


Fig 2.2-1 แสดงโครงสร้างภายในstepping motor

ในที่นี้ซึ่งถ้าเราเพิ่มจำนวนของขั้วแม่เหล็กมากขึ้นจะเพิ่มจำนวนของสเต็ปต่อวงจรรอบมากขึ้นตามด้วย ดูตามรูปด้านบน ลักษณะการนำไปใช้งาน stepping motor จะใช้งานลักษณะ Open Loop คือ stepping motor สามารถทำงานได้โดยไม่ต้องมีการ ป้อนค่าพารามิเตอร์กลับมา (Feed back)

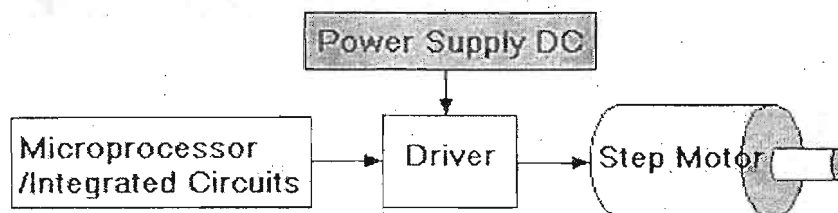


Fig 2.2-2 ลักษณะระบบการขับ และควบคุม stepping motor

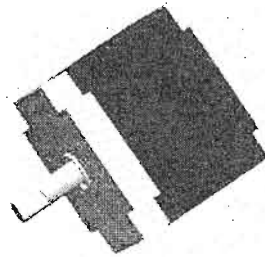


Fig 2.2-3 stepping motor

stepping motor เป็นอุปกรณ์จำพวกเชิงกลทางไฟฟ้า โดยการหมุนที่ละ สติ็ปซึ่งอยู่ระหว่าง 0.1 - 30 องศา ขึ้นอยู่กับโครงสร้างของ stepping motor โดยสัญญาณพัลส์ที่จ่ายให้กับขดลวดสเตเตอร์ทำให้เกิดแรงผลักแก่ โรเตอร์หมุนไป stepping motor มีขดลวดหลายชุดในที่นี้เราเรียกว่า เฟส (Phase) ดังนั้นสัญญาณที่ต่อเนื่องเป็นลำดับ ทำให้ motor หมุนวนไปในทิศทางตามที่ต้องการได้

stepping motor แบ่งเป็น 2 ชนิด ตามการพันขดลวดบนสเตเตอร์ Stepping motor คือ แบบ Bipolar(ไบโพลาร์) กับ แบบ Unipolar(ยูนิโพลาร์)

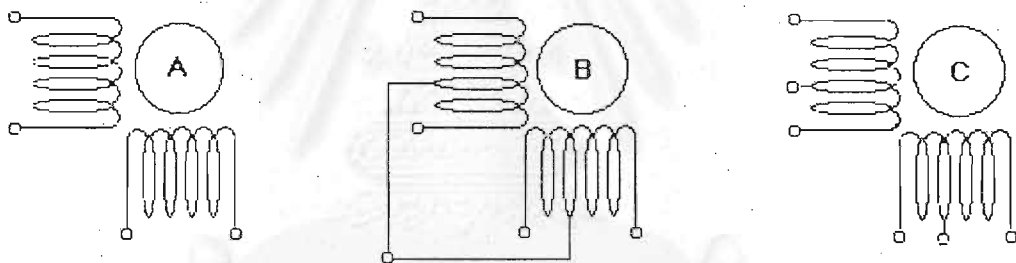


Fig 2.2-4 stepping motor แบบ bipolar      stepping motor แบบ unipolar 5 ขั้ว      stepping motor แบบ unipolar 6 ขั้ว

#### แบบ Bipolar

จะมีการพันขดลวดหนึ่งขด ในแต่ละขั้วแม่เหล็กของสเตเตอร์ โดยขั้วแม่เหล็กที่เกิดขึ้น ที่สเตเตอร์จะถูกกำหนดโดยทิศทางของการไหลของกระแสไฟฟ้า ซึ่งสามารถทำให้เกิดขั้วแม่เหล็กในทิศทางตรงกันข้ามได้เพียงทางเดียว การกลับทิศทางของการไหลในกระแสไฟฟ้า ทำได้โดยการควบคุมของวงจรสวิทชิงให้กลับขั้วไฟฟ้า

#### แบบ Unipolar

แบบนี้มี 2 ขด บนแต่ละขั้วแม่เหล็กของสเตเตอร์ ทำให้แต่ละขดลวดเกิดขั้วแม่เหล็กในทิศทางตรงข้าม การควบคุมและสั่งงานให้ stepping motor ทำงาน ไปที่ละสติ็ปสามารถทำได้โดยการจ่ายกำลังไฟไปยัง ขดลวด ในแต่ละขดบนสเตเตอร์ โดยการป้อนจะทำในลักษณะเป็นลำดับที่ถูกต้อง ซึ่งจะแบบ ได้เป็น 3 รูปแบบ คือ แบบ 1 เฟส แบบ 2 เฟส และแบบ half step ทั้ง 3 แบบนี้ก็จะมีข้อดีและข้อเสียต่างกันออกไป

### แบบ 1 เฟส

จะเป็นการกระตุ้นแบบง่ายที่สุด ซึ่งจะทำให้การกระตุ้นขดลวดทีละขดในเวลาหนึ่งๆเรียงกันไป ตัวอย่างเช่น ขดที่ 1, 2, 3, 4, 1, 2, 3, 4 เป็นลำดับแบบนี้ หรือ ขด 1, 4, 3, 2, 1, 4, 3, 2 เป็นลำดับกันไป ทั้งนี้ขึ้นอยู่กับทิศทางที่เราต้องให้มอเตอร์หมุนไป วงจรที่นำมากระตุ้นนั้นจะมีราคาค่อนข้างจะถูกกว่า และง่ายกว่า ดังในรูปของวงจรการจ่ายไฟ ที่อยู่ด้านบนนั้น

เราสามารถเขียนขั้นตอนการทำงานเป็นตารางออกมาได้ดังนี้

ตารางที่ 2.1 ขั้นตอนการทำงานของ stepping motor แบบ 1 เฟส

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

### แบบ 2 เฟส

แบบนี้ก็จะคล้ายกับการกระตุ้นในแบบ 1 เฟส แต่จะต่างกันตรงที่ แบบ 2 เฟส จะกระตุ้นทีละ 2 ขด ที่อยู่ใกล้กันใน เวลาเดียวกัน และจะเรียงลำดับกันไป ดังเช่นแบบเดียวกับแบบ 1 เฟส จะยกตัวอย่าง การกระตุ้นขดลวดในลักษณะ เป็นลำดับดังนี้ 12,23,34,41,12,23,34,41 เรียงลำดับกันไปเรื่อยๆ หรือจะเป็น 14,43,32,21,14,43,32,21 เรียงกันไปเรื่อยๆเช่นกัน จะกล่าวถึงข้อดี - ข้อเสียของแบบ 2 เฟส แล้วมีดังนี้

- **ข้อดี** : การที่เราจะเพิ่มจำนวนขดลวดที่ถูกกระตุ้นจะทำให้แรงบิดได้มากกว่า แบบ 1 เฟส ซึ่งมอเตอร์จะหมุนด้วยแรง ดึงแบบเต็มๆแรงจาก ทั้ง 2 ขดลวดที่กระตุ้นพร้อมกัน
- **ข้อเสีย** : แบบ 2 เฟส จะกระตุ้นขดลวดนั้นต้องใช้กำลังไฟมากขึ้นเป็น 2 เท่าของแบบ 1 เฟส

เราสามารถเขียนลำดับการกระตุ้นของขดลวดแบบ 2 เฟส ได้ดังในตาราง ต่อไปนี้

ตารางที่ 2.2 ขั้นตอนการทำงานของ stepping motor แบบ 2 เฟส

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

#### แบบ half step

แบบนี้แบบรูปแบบผสมผสานของการกระตุ้นระหว่าง แบบ 1 เฟส กับ แบบ 2 เฟส เพื่อให้จำนวนรอบของ สเต็ปให้ มากขึ้นเป็น 2 เท่า ซึ่งในระบบนี้จะทำการกระตุ้นขดลวดเรียงกันไปเรื่อยๆเป็นลำดับ ดังจะยกตัวอย่างต่อไปนี้ 1,12,2,23,3,34,4,41,1,12,2,23,3,34,4,41,1 เป็นลำดับอยู่อย่างนี้เรื่อยไป ถ้าจะกลับทิศทางการหมุนก็จะขับดังนี้ 1,41,4,43,3,32,2,21,1,41,4,43,3,32,2,21,1 เป็นลำดับ กันไป ข้อดีและข้อเสียของการกระตุ้นแบบhalf stepเป็นดังนี้

- ข้อดี : การกระตุ้นแบบนี้จะให้แรงบิดที่เพิ่มมากขึ้น เนื่องจากช่วงสเต็ปที่มีระยะสั้นลง อีกประการหนึ่งคือ แต่ละสเต็ปเกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกันเป็นผลให้ค่าตำแหน่งความถูกต้องมากขึ้นไปด้วย
- ข้อเสีย : ก็คงจะเช่นเดียวกับแบบ 2 เฟส ที่ต้องจ่ายกำลังไฟเป็น 2 เท่าของแบบ 1 เฟส หรือจะใช้เท่า กับแบบ 2 เฟส นั้นเอง

ดังนั้นเราสามารถนำลำดับการทำงานของ แบบ half step ในรูปของตารางได้ดังนี้

ตารางที่ 2.3 ขั้นตอนการทำงานของ stepping motor แบบ half step

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

## 2.2.2 Stepping Motor Driver Circuit

### 2.2.2.1 แบบใช้ SM2220 เป็นตัวสลับสัญญาณ

ในส่วนของ วงจรขับ stepping motor จะใช้วงจรขับของ robo basic ซึ่งเป็นหุ่นประกอบสำเร็จรูปของบริษัท inex เป็นต้นแบบ แต่เนื่องจาก robo basic เป็นหุ่นขนาดเล็กที่ใช้การขับเคลื่อนด้วย stepping motor เช่นเดียวกันแต่จะเป็น ขนาดที่เล็กกว่ามาก คือ ขนาด 12 โวลท์ ขนาดกระแส 300 mA ดังนั้นจึงไม่สามารถนำ ส่วนกำลังมาใช้ได้ แต่จะสามารถใช้ได้เพียงส่วนของการสร้างสัญญาณขับ stepping motor โดยใช้ IC เบอร์ SM2220 ในความเป็นจริงเราสามารถ ขับ motor แบบขับตรงได้โดยใช้ ส่วน controller สร้างสัญญาณ ที่เป็นลำดับการขับได้ แต่เมื่อจะขับ motor 2 ตัว จะต้องใช้ จำนวนบิตในการขับเป็น 8 บิต ในขณะที่การใช้ IC เบอร์ดังกล่าวจะใช้เพียง 4 บิต

วงจรขับ stepping motor ของ robo basic จะใช้สัญญาณจาก microcontroller โดย motor 1 ตัวจะต้อง ส่งสัญญาณออกมาเป็นพัลส์ 1 ช่องและ logic ควบคุมทิศ 1ช่อง เข้าสู่ขาของ IC SM2220 จากนั้น IC จะทำการ สร้างลำดับสัญญาณเพื่อขับ stepping motor สัญญาณที่ได้นี้จะเป็น TTL logic เข้าภาคกำลัง และให้ IC ULN2003 ซึ่งเป็น IC ที่ทำงานแบบ Not Logic Darlington switch 7 channels การขับ stepping motor จึงทำงานโดยจ่ายไฟบวก +Vcc เข้าขาร่วมของ motor และลำดับสัญญาณด้วยขั้วลบ แต่เนื่องจาก ULN2003 มีคุณสมบัติที่สามารถจ่ายกระแสได้เพียง 500 mA ต่อช่อง ไม่เพียงพอต่อความต้องการในการขับ motor ขนาดใหญ่ที่จะ ใช้ และต้องสามารถขับกระแสได้ถึงประมาณ 2A ต่อช่อง และไม่สามารถขับ stepping motor ที่ทำงานที่ระดับแรง

ดันต่ำกว่า 5 โวลท์ ในขณะที่ stepping motor จำนวนไม่น้อย มักทำงานที่แรงดันต่ำกว่า 5 โวลท์ ส่วนวงจรที่ทำการ  
ดัดแปลง เพื่อให้สามารถทำงานได้กับ stepping motor ขนาดที่ต้องการใช้

จากวงจรขับ robo basic เราสามารถใช้ข้อดี ของ IC SM2220 ที่สามารถใช้งานได้สะดวกและจะทำการ  
เปลี่ยนแปลงเฉพาะภาคกำลังและส่วนแสดงผลจาก

วงจรและหลักการทำงานของวงจรขับ stepping motor ของ robo basic

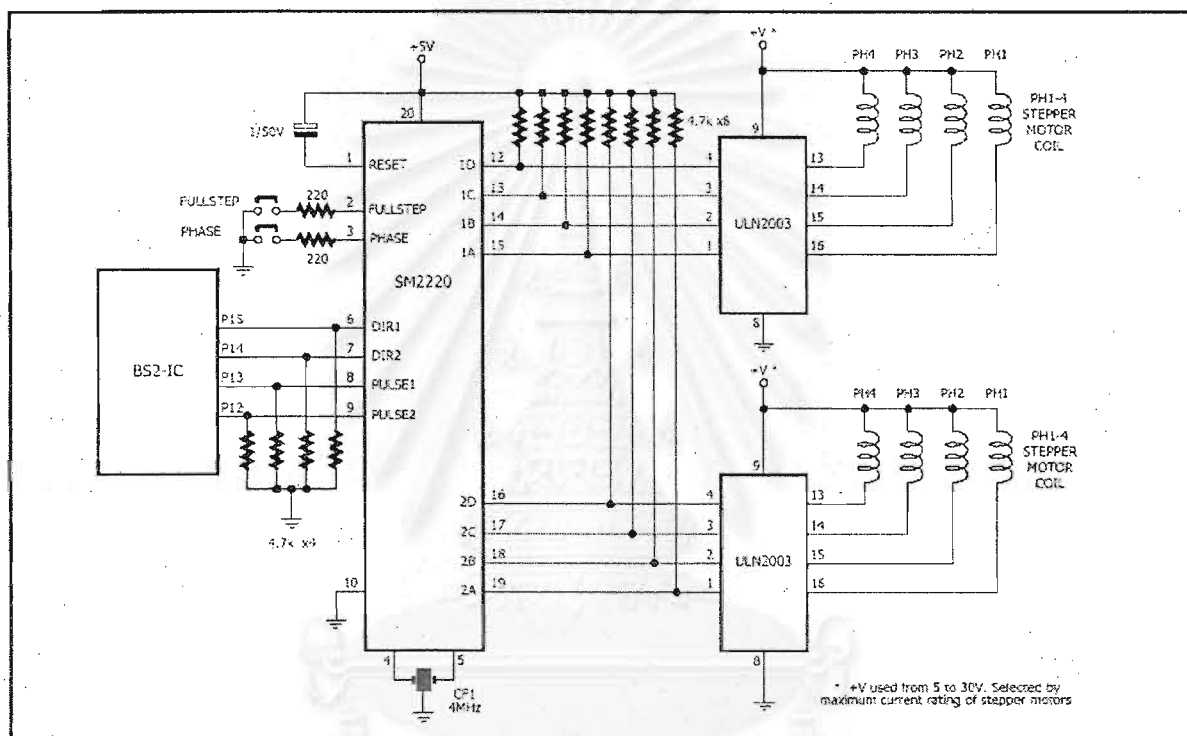


Fig 2.2-5 วงจรขับ stepping motor โดยใช้ IC Motor เป็นตัวขับ

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## STEPPING MOTOR CIRCUIT USING SM2220



Fig 2.2-6 วงจรขับ stepping motor โดยใช้ BJT

รูปร่างวงจรขับ stepping motor โดยใช้ IC SM2220 สามารถแบ่งการทำงานออกเป็น 2 ส่วนได้คือ ภาคสัญญาณ และภาคกำลัง

### 1. ภาคสัญญาณ

- SM2220 (ดู datasheet ประกอบ)

เป็น IC ที่สามารถขับ unipolar stepper motor ได้ 2 ตัว ซึ่งสามารถเลือกแบบการขับได้ 3 แบบ คือ full step (1 phase และ 2 phase) หรือ half step โดยจะสร้างสัญญาณ 8 บิต เพื่อจ่ายให้ขดลวดมอเตอร์ได้ 8 ขด (4 บิต ต่อมอเตอร์ 1 ตัว) ลักษณะการทำงานเป็นแบบ TTL logic สามารถทำงานได้ที่ 2.7 - 6.0 โวลต์และใช้งานร่วมกับ crystal 4 MHz

การเขียนโปรแกรมควบคุมการทำงานของ SM2220 ทำได้โดยดูตารางข้างล่างประกอบ แล้วส่งไปยัง microcontroller (AVR) การเขียนโปรแกรมนี้อธิบายละเอียดอีกทีในหัวข้อเรื่อง การเขียนโปรแกรมขับเคลื่อน stepping motor ด้วย SM2220

ในวงจรข้างต้นนี้ ให้ SM2220 รับสัญญาณจาก AVR ทั้งหมด 7 บิต แต่ในวงจรใช้ port 8 บิต

ในการเชื่อมต่อ ดังนั้นจะมี 1 ขาของ port นี้เป็นขาลอยเสมอ

Pin	Description	Logic '0'	Logic '1'	Comment
1	RESET	Reset	Run	When reset, all outputs are logic '1'
2	FULLSTEP	Full step	Half step	
3	PHASE	2 phase	1 phase	Don't care if FULLSTEP on as '1'
4	X1			Connect XTAL or ceramic resonator pin 1
5	X2			Connect XTAL or ceramic resonator pin 2
6	DIR1	Backward	Forward	
7	DIR2	Backward	Forward	
8	PULSE1	Step run	Stop	Active low
9	PULSE2	Step run	Stop	Active low
10	GND			Ground system
11	NC			No connect
12	1A			Phase 1 of motor #1
13	1B			Phase 2 of motor #1
14	1C			Phase 3 of motor #1
15	1D			Phase 4 of motor #1
16	2D			Phase 4 of motor #2
17	2C			Phase 3 of motor #2
18	2B			Phase 2 of motor #2
19	2A			Phase 1 of motor #2
20	+Vcc			Power supply

Fig 2.2-7 รายละเอียดของขา

ตารางที่ 2.4 แสดงสัญญาณที่ SM2220 ได้รับมาในแต่ละขา

บิต	7	6	5	4	3	2	1	0
ความหมายของสัญญาณที่ได้รับมา	DIR2	PULSE2	DIR1	PULSE1	NON CONNECT	FULL/HALF STEP	PHASE	RESET
ส่งสัญญาณไปที่ SM2220 ที่ขาเบอร์	7	9	6	8	-	2	3	1



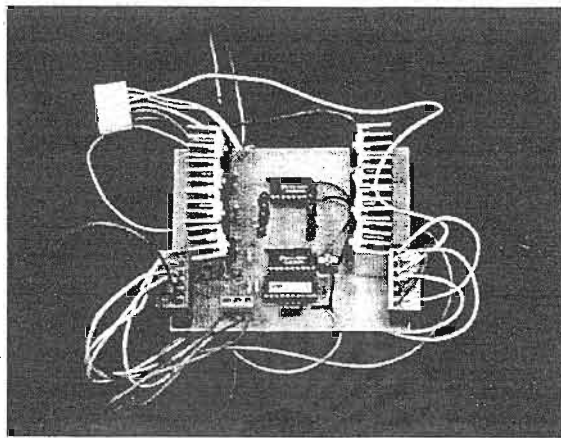
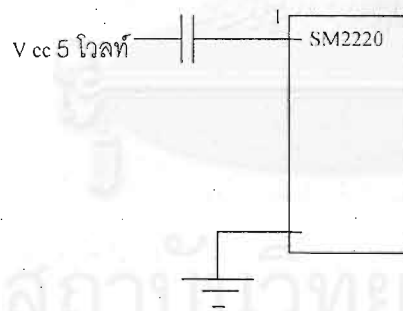


Fig 2.2-8 แผ่นวงจร stepping motor โดยใช้ SM2220 เป็นตัวสร้างสัญญาณ

สัญญาณที่ออกจาก SM2220 จะแยกเพื่อขับมอเตอร์ 2 ตัว นั่นคือ ขาที่ 12 - 15 จะสำหรับมอเตอร์ตัวที่ 1 และขาที่ 16 - 19 สำหรับมอเตอร์ตัวที่ 2 เมื่อได้รับสัญญาณ reset จาก microcontroller สัญญาณที่ออกจาก SM2220 จะเป็น 1 ทั้งหมด และจะเริ่มเข้าสู่การรับ pulse เพื่อขับมอเตอร์โดยมีลักษณะและทิศทางการขับตามโปรแกรมที่เขียนไว้

หมายเหตุ : การรับคำสั่ง RESET อาจไม่รับจาก microcontroller ก็ได้ โดยการเปลี่ยนวงจรส่วนที่ต่อเข้ากับขา 1 ของ SM2220 เป็นคิงรูปข้างล่างนี้ ผลก็คือจะ RESET ทุกครั้งที่จ่ายไฟครบวงจร



- แบบใช้ ULN2803 เป็นตัวกลับสัญญาณให้

เป็น IC ที่ทำงานแบบ Not Logic Darlington switch 8 channels นั่นคือมีช่องทำงานเพิ่มขึ้นจาก ULN2003 ขึ้นมา 1 ช่อง แต่การทำงานเหมือนกัน จึงสามารถใช้ IC ULN2803 เพียงตัวเดียวแทนที่ ULN2003 จำนวน 2 ตัวได้ และในที่นี้ไม่ได้ใช้ IC ตัวนี้ในการขับมอเตอร์โดยตรง ใช้เพียงส่งสัญญาณให้ทรานซิสเตอร์ จะมีกระแสไหลผ่านไม่มากนัก ดังนั้นจึงไม่ต้องระวังเรื่องความร้อนที่จะเกิดขึ้นถ้าใช้ทำงานแทน ULN2003 จำนวน 2 ตัว

การต่อ IC ตัวนี้เพื่อกลับสัญญาณให้มีลักษณะตรงกันข้ามกับสัญญาณที่ออกเพื่อประโยชน์ในการแสดงผลที่ LED โดยกำลังไม่ลดน้อยลง สาเหตุที่ไม่ใช้เกท NOT (74LS04) เพื่อกลับสัญญาณแทน ULN2803 คือ

- ULN2803 สามารถจ่ายกระแสได้ถึง 500 mA ต่อช่องสัญญาณ โดยจะขยายกระแสทาง input ให้เพิ่มขึ้น
- output voltage สามารถสูงได้ถึง 50 โวลท์ แต่ในที่นี้ใช้ 5 โวลท์ ตามลักษณะสัญญาณ TTL ที่ออกมาจาก SM2220
- มี diode ป้องกัน back electromagnetic field (EMF) ในตัวเองแล้ว

หลังจากนั้นใช้ ULN2803 อีก 1 ตัว เพื่อกลับสัญญาณให้เป็นดังเดิม เพื่อให้ทรานซิสเตอร์ on และมอเตอร์ทำงานตามต้องการ

- ตัวอย่างสัญญาณจาก SM2220

สมมติกำหนดการขับเป็นแบบ full step 1 phase (วิธีการเขียนคำสั่งจะอธิบายละเอียดอีกทีในหัวข้อเรื่องการเขียนโปรแกรมขับเคลื่อน stepping motor ด้วย SM2220

สัญญาณที่ออกจาก SM2220 ครั้งแรก คือ 1111 1111 นั่นคือได้รับคำสั่ง RESET จาก AVR หลังจากนั้น SM 2220 จะสร้างสัญญาณให้เป็น 1000 0001 เมื่อได้รับคำสั่ง pulse จาก AVR ก็จะสร้างสัญญาณ 0100 0010 เมื่อได้รับคำสั่ง pulse จาก AVR อีกครั้ง ก็จะได้สัญญาณ 0010 0100 ตามด้วย 0001 1000 แล้ววนซ้ำเช่นนี้ไปเรื่อยๆ

ตารางที่ 2.5 แสดงตัวอย่างสถานะบิตของ IC และ diode ตามลำดับการทำงานในวงจรขับ stepping motor

ลักษณะสัญญาณ เมื่อผ่าน IC ในวงจร	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SM2220	1	0	0	0	0	0	0	1
ULN 2803 ตัวที่ 1	0	1	1	1	1	1	1	0
ลักษณะ DIODE	ติด	ไม่ติด	ไม่ติด	ไม่ติด	ไม่ติด	ไม่ติด	ไม่ติด	ติด
ULN 2803 ตัวที่ 2	1	0	0	0	0	0	0	1

#### VOLTAGE PULLING UP

ในภาคทฤษฎี voltage ที่ออกจาก SM2220 จะมีแค่ 5 โวลท์ (logic 1) และ 0 โวลท์ (logic 0) เท่านั้น แต่ในทางปฏิบัติเมื่อกระแสไหลผ่านอุปกรณ์ต่างๆ ในวงจรจะเกิดการสูญเสียแรงดันขึ้น (voltage drop) ทำให้แรงดันไม่ถึง 5 โวลท์ และอาจน้อยมากจนไม่สามารถ on ทรานซิสเตอร์ ให้ทำงานได้ ดังนั้นจึงต้องมีการเพิ่มแรงดันชดเชยเข้าไป โดยการต่อตัวต้านทานเข้ากับขาสัญญาณ แล้วใช้แหล่งจ่ายไฟ 5 โวลท์ จ่ายสำรองไว้ตลอดเวลา เมื่อใดที่เกิดการสูญเสียแรงดันที่ขาสัญญาณขึ้น กระแสก็จะไหลจากแหล่งจ่ายไฟผ่านตัวต้านทานเข้าเพิ่มแรงดันที่วงจรจนกว่าแรงดันที่เพิ่มจะถึง 5 โวลท์ กระแสจึงจะหยุดไหล

ในวงจรนี้ใช้ชุดต้านทานสำเร็จรูป (R package) เบอร์ 9A472J ซึ่งข้างในจะต่อเป็นวงจรตามที่ต้องการให้ แล้วเพื่อเป็นการลดขนาดและความยุ่งยากของวงจร และทำการชดเชยแรงดันทั้งหมด 3 ครั้ง คือหลังออกจาก SM2220 และ หลังออกจาก ULN2803 ทั้ง 2 ตัว

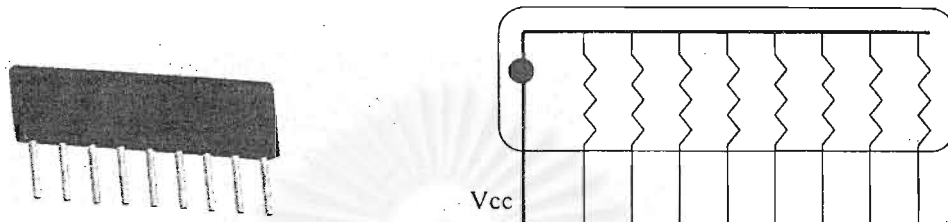


Fig 2.2-9 ตัวต้านทานแบบแถวเรียง Fig 2.2-10 แสดงลักษณะภายในของตัวต้านทานแบบแถวเรียง

อย่างไรก็ตามการชดเชยแรงดันนี้ก็ยังไม่สามารถทำได้ดี 100 % นั่นคือแรงดันที่วัดได้หลังจากผ่านการชดเชยนี้แล้วยังไม่ถึง 5 โวลต์ จากการวัดพบว่าจะช่วยเพิ่มขึ้นได้ประมาณ 0.5 - 1 โวลต์

- ส่วนแสดงผล

เป็นส่วนที่เพิ่มขึ้นมาเพื่อให้สามารถตรวจสอบความถูกต้องในการทำงานของวงจรได้ง่ายขึ้น จะใช้ LED เป็นตัวแสดงผล LED ที่ใช้เป็นแบบมาตรฐาน 3 โวลต์ จึงต้องมีส่วนตัวต้านทานลดแรงดันลงก่อนเข้า LED ภาคแสดงผลจะมีผลช่วยในการ pull up voltage ด้วย

## II. ภาคกำลัง

จะใช้ Darlington Power Transistor เบอร์ TIP122 (NPN type) ซึ่งมีคุณสมบัติที่ ตรงตามต้องการได้แก่

- มีหลักการทำงานคือ เมื่อมีกระแสไหลเข้าที่ขาเบสเพียงเล็กน้อย กระแสจะสามารถไหลจาก collector ไปสู่ emitter ได้ ซึ่งหมายถึงจะผ่านขดลวดมอเตอร์ 1 ขดได้
- สามารถขับกระแสได้สูงสุดถึง 5A ซึ่งกระแสที่ไหลผ่านมอเตอร์นี้จะแปรผันตามกระแสที่เข้าขาเบส ดังนั้นเมื่อ ULN2803 สามารถจ่ายกระแสได้สูง ก็จะทำให้กระแสที่ไหลผ่านมอเตอร์เพิ่มขึ้นด้วย ทำให้มอเตอร์มีแรงบิดมากขึ้น
- ทรานซิสเตอร์จะ on 1.5 โวลต์ คือจะแบ่ง logic 0 และ 1 ที่ 1.5 โวลต์
- แรงดันที่ใช้ในการขับ stepping motor ไม่ถูกจำกัดที่ 5 โวลต์ แต่จะสามารถทำงานได้ตั้งแต่ที่แรงดัน 2 โวลต์
- ต้องวางโหนดไว้ที่ขา connector ของทรานซิสเตอร์ เพื่อให้ได้กำลังขับสูงสุดและไม่เกิดความผิดพลาดในการขับ

ปัญหาที่เกิดขึ้นของภาคกำลังในการออกแบบวงจรได้แก่

- การออกแบบวงจรโดยวางตำแหน่งของภาวะ ผิดพลาดเช่นวางที่ขา emitter ทำให้มีผลกระทบต่อกระแสที่ขา base (ซึ่งเป็นขาควมคุม ควรจะมีกระแสไหลคงที่) การวางโหนดไว้ที่ emitter นี้ ถ้าไหลดมากจะทำให้กระแสที่ไหลเข้าขาเบสน้อยเกินไป และเนื่องจากทรานซิสเตอร์ทำหน้าที่เหมือนตัวขยายกระแส ดังนั้นกระแสของภาคกำลังที่จ่ายให้กับมอเตอร์จะน้อยลงไปด้วย จึงทำให้แรงบิด (torque) ที่ได้จากมอเตอร์น้อยลงไปด้วย หรือบางครั้งกระแสที่ให้กับขาเบสอาจจะน้อยมากจนทรานซิสเตอร์ไม่สามารถทำงานได้ นั่นคือ มอเตอร์ก็จะไม่หมุนที่จังหวะพัลสนั้น
- ปัญหาในเรื่องการระบายความร้อน เมื่อมีกระแสขับผ่านสูง 2 - 3 A จะเกิดความร้อนขึ้นสูงมาก จึงต้องการ heat sink ที่มีขนาดใหญ่ หรือพัดลมช่วยในการระบายความร้อนที่เกิดขึ้น เพื่อป้องกันความเสียหาย
- เกิดการสูญเสียแรงดันที่ IC SM2220 มาก (ประมาณ 3 โวลท์) จึงต้องชดเชยแรงดันเข้าไปด้วย

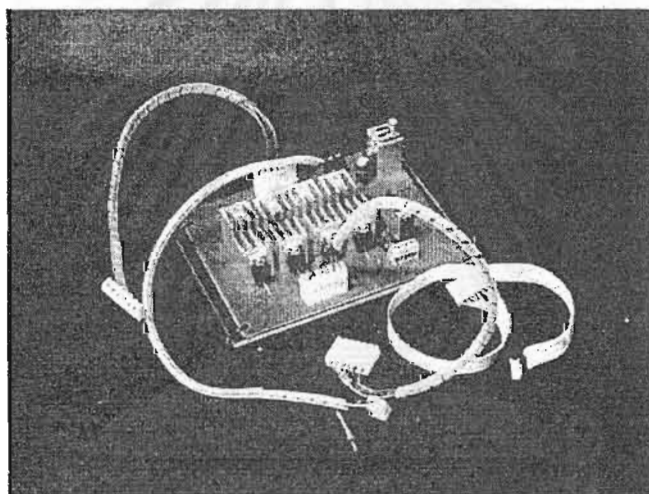


Fig 2.2-11 แผงวงจรขับ stepping motor

ข้อเสนอแนะ : สำหรับในส่วนภาคกำลัง คือ การดัดแปลงใช้ Power MOSFET เข้าทำงานแทน Power transistor เนื่องจาก Power MOSFET มีคุณสมบัติที่ดีกว่าคือ ทั้งในส่วนการขับกระแสที่สูงกว่า สูญเสียกำลังน้อยกว่าทำให้ความร้อนที่เกิดขึ้นน้อยกว่า รวมทั้งยังใช้กระแสในการ trig สัญญาณที่ขา gate (เหมือนขา base ของ ทรานซิสเตอร์) น้อยมากในระดับ  $\mu\text{A}$  จึงไม่มีผลกระทบต่อวงจร แต่ปัญหาที่ต้องแก้คือ Power MOSFET จะมีอัตราขับกระแสที่ขึ้นกับแรงดันขา gate เป็นหลัก ดังนั้นแรงดันระดับ TTL Logic จึงไม่เพียงพอ ต้องหา buffer ที่สามารถเพิ่มแรงดันสัญญาณ TTL ก่อนจึงจะใช้งานได้ นอกจากนี้ Power MOSFET ยังมีราคาสูงกว่าทรานซิสเตอร์ ถึง 2.5 - 3.0 เท่าตัว

#### 2.2.2.2 แบบขับตรง ใช้ microcontroller เป็นตัวสลับสัญญาณ

การขับ stepping มอเตอร์ สามารถทำได้อีกวิธีหนึ่งคือ เขียนโปรแกรมส่งสัญญาณให้ทรานซิสเตอร์ ON โดยตรง ขดลวดมอเตอร์ก็จะได้รับการจ่ายไฟเหมือนเดิม นั่นคือจะต่างกันในส่วนของภาคสัญญาณและการโปรแกรม

แต่ภาคกำลังยังใช้หลักการเดิมในการทำงาน ตัวอย่างเช่น คำสั่ง 1000 0001 จะหมายถึงจ่ายไฟให้ขดลวดมอเตอร์ทั้ง 2 ตัวที่ขดแรก คำสั่ง 0100 0010 ก็จะมีหมายถึงจ่ายไฟให้ขดลวดถัดมาทั้ง 2 มอเตอร์ วงจรนี้ไม่มี IC มากมาย จึงไม่ต้องการการชดเชยแรงดัน และการแสดงผลก็ใช้ส่วนแสดงผลบน AVR ซึ่งคือ LED ได้ วงจรนี้จะซับซ้อนน้อยกว่าวงจรขับ stepping motor ที่ใช้ SM2220 เป็นตัวสลับสัญญาณมาก และจะได้แรงขับมากกว่า เนื่องจากกระแสและแรงดันที่ไหลเข้าขาเบส ไม่สูญเสียที่ IC ต่างๆ

วงจรมีได้ดัดแปลงมาใช้ Power MOS FET แทนทรานซิสเตอร์เพื่อให้ได้แรงบิดสูงสุดดังที่ได้กล่าวมาแล้ว

### STEPPING MOTOR CIRCUIT (NOT USING SM2220)

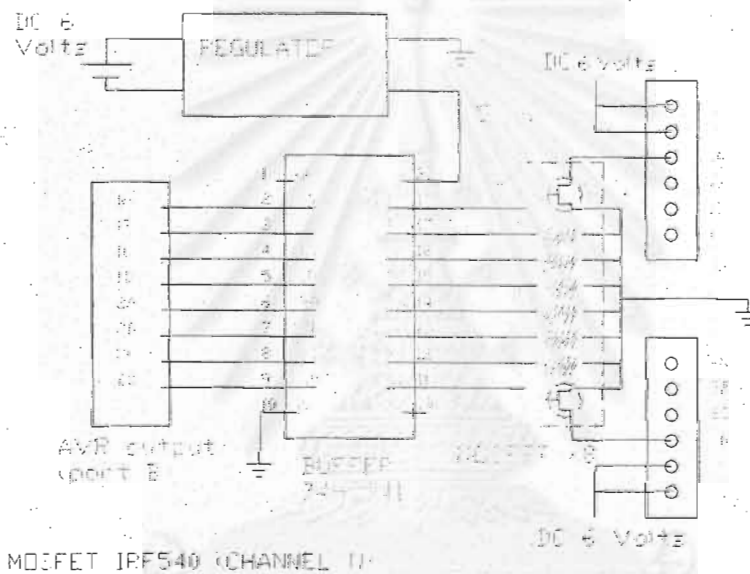


Fig 2.2-12 วงจรขับ stepping motor โดยใช้ power

จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 2.6 แสดงการเปรียบเทียบข้อดีและข้อเสียของการขับ stepping motor ด้วยวิธีต่างๆ

วิธีการขับ	ข้อดี	ข้อเสีย
1) ขับด้วย IC ULN2003 เช่นเดียวกับ ROBO BASIC ใช้ IC SM2220 ในการเรียงสัญญาณ	1) สร้างวงจรและนำมาใช้งานได้ง่าย 2) โปรแกรมง่าย	1) ได้แรงบิดต่ำ เพราะกระแสในการขับถูกจำกัดด้วยขีดสูงสุดที่ IC สามารถให้ได้ 2) ไม่สามารถขับ stepping motor ที่ใช้แรงดันต่ำกว่า 5 volts ได้
2) ขับด้วย transistor (TIP122) ใช้ IC SM 2220 ในการเรียงสัญญาณ	1) โปรแกรมง่าย 2) ได้แรงบิดสูง	1) วงจรซับซ้อนเนื่องจากต้องแสดงผลในวงจรด้วย 2) ต้องมีการชดเชยแรงดันที่ตกลง 3) เกิดความร้อนสูงที่ transistor ได้ง่ายเนื่องจากเกิดการสูญเสียกระแสมาก
3) ขับด้วย power MOSFET (IRF 540) ไม่ใช้ IC SM 2220 ในการเรียงสัญญาณ	1) วงจรเข้าใจง่ายเนื่องจากไม่ต้องมีส่วนแสดงผล ใช้ส่วนแสดงผลของ microcontroller ได้เลย 2) power MOSFET ให้แรงบิดได้สูง 3) สามารถทนกระแสได้สูงมาก 4) ไม่รบกวนสัญญาณ logic เนื่องจากใช้กระแสที่ขาคอนน้อยมาก	1) Power MOSFET ราคาแพงกว่า transistor ถึง 3 เท่า

### 2.2.3 DC motor

ลักษณะการใช้งานของ DC motor จะต้องการขับ motor ที่แรงดันใช้งานในช่วง 12 - 24 โวลท์ ที่กระแส ประมาณ 0.5 A เป็น DC gear motor ที่มีอัตราทดรอบสูง 25:1 เนื่องจากต้องการแรงที่สูงมาก ในขณะที่ไม่ต้องการความเร็ว และวงจรขับต้องสามารถขับให้ motor หมุนไปกลับได้

ส่วนของวงจรขับ สามารถสร้างจาก Power transistor ได้ โดย IC buffer เพิ่มกระแส logic ที่ออกจาก controller board ก่อนเข้าทรานซิสเตอร์ 2 ตัวที่ต่อในลักษณะกลับทิศทาง แต่ที่ใช้งานจริง จะมี Power IC ที่สามารถทำงานที่ตามที่ต้องการทุกประการ คือ รับสัญญาณเป็น TTL Logic และสามารถควบคุม motor ให้สามารถหมุนไปกลับได้ ได้แก่ IC เบอร์ L298 และ

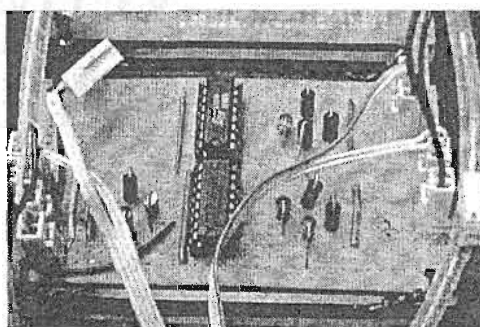


Fig 2.2-13 แผงวงจรขับ DC

L293 ซึ่ง IC ทั้งสองจะมามีการทำงานเหมือนกันแต่ L298 สามารถขับกระแสได้ 2 A 2 channels ขณะที่ L293 จะสามารถขับ กระแสได้ 1A 2 channels โดย L293 จะเป็นตัวจ่ายกระแสเพื่อให้ขับมอเตอร์ได้ 2 ทาง โดยใช้ ไดโอด ขนาด 2A ชนิด fast recovery 4 ตัวในการต่อวงจรเพื่อขับ DC มอเตอร์ 1ตัว และ L293 1 ตัวจะจ่ายกระแสเพื่อขับ มอเตอร์ 2 ทางได้ 2 ตัว

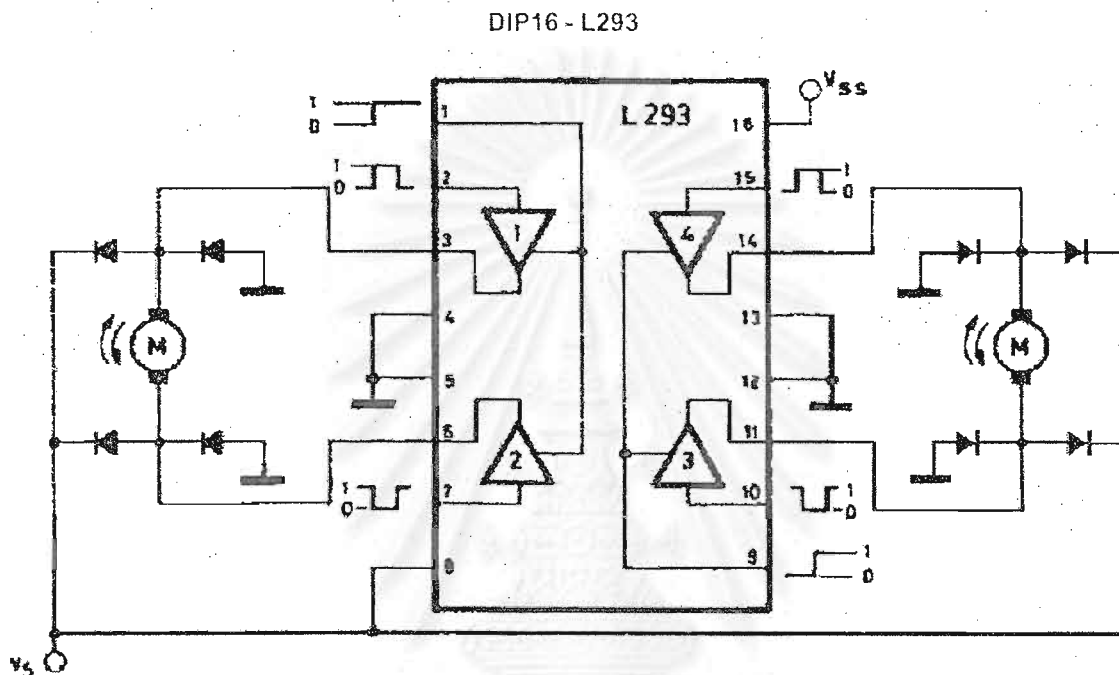


Fig 2.2-14 วงจรขับ DC Motor

การใช้งานในหุ่นยนต์ตัวนี้ได้เลือกใช้วงจรจาก L293 เนื่องจากวงจรง่าย สามารถต่อวงจรเองได้โดยใช้แผ่น วงจรเนกประเภทสังค์ แต่การ L298 วงจรซับซ้อน ต้องสั่งทำแผ่นวงจรสำเร็จซึ่งทำให้เมื่อเกิดปัญหาไม่สามารถแก้ไขได้ เอง

#### 2.2.4 Controller Board

Controller board จะใช้ microcontroller ตระกูล AVR ของ Atmel เบอร์ 90S8535 บนบอร์ดควบคุมที่มีคุณสมบัติดังนี้

- หน่วยความจำแบบ FLASH สำหรับบันทึก program memory ขนาด 8 Kbytes (AT90S/LS8535)
- หน่วยความจำแบบ EEPROM สำหรับบันทึก data memory ขนาด 512 Bytes (AT90S/LS8535)
- หน่วยความจำแบบ RAM ขนาด 512 Bytes (AT90S/LS8535)
- ระบบการเปลี่ยนสัญญาณ Analog To Digital ขนาด 10 บิต จำนวน 8 channels
- กลุ่ม Register ใช้งานทั่วไป ขนาด 8 บิต จำนวน 32 ตัว
- input/output port ขนาด 8 บิต จำนวน 4 ports

- input/output line ขนาด 32 pins จำนวน 4 ports
- ระบบการสื่อสารข้อมูล digital แบบอะซิงโครนัส ( UART ) 1 channel
- ระบบการสื่อสารข้อมูล digital แบบซิงโครนัส ( SPI ) 1 channel
- ความถี่สัญญาณนาฬิกา 0 - 8 MHz ( AT90S8535/90S4434 )
- ระบบการรีเซ็ตแบบอัตโนมัติเมื่อเริ่มจ่ายกระแสไฟฟ้าเข้า microcontroller ( power on reset )
- ระบบการกำเนิดความถี่สัญญาณ แบบ PWM จำนวน 3 channels
- ระบบการตรวจจັบระดับสัญญาณ analog ( analog comparator )
- 3 sleep mode : idle , power save และ power down
- ระบบการป้องกันการ copy ข้อมูลภายในหน่วยความจำ ( lock for software security )
- ระบบตรวจจັบการทำงานผิดพลาดของ CPU ( watchdog timer with on-chip oscillator )
- ระบบการอินเตอร์รัพท์จากภายนอก ( external interrupt )
- timer / counter ขนาด 16 บิต 1 channels
- timer / counter ขนาด 8 บิต 2 channels
- Vcc : 4.0 – 6.0 โวลท์ AT90S4434/AT90S8535

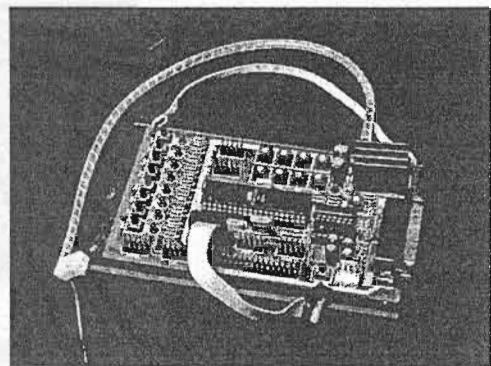


Fig 2.2-15 Controller board ET-AVR

#### 2.2.4.1 รายละเอียดของขาสัญญาณ

Vcc : ขาจ่ายไฟให้กับ CPU

GND : กราวด์

Port A ( PA 7 .. PA 0 ) : เป็น port 2 ทิศทาง ขนาด 8 บิต โดยสามารถกำหนดให้แต่ละขาของ port สามารถ pull up ภายในแยกจากกัน ซึ่งสามารถรับกระแส sink 20 mA โดย port A ยังใช้เป็นขา input เพื่อรับสัญญาณ analog ในส่วนของการแปลงสัญญาณ Analog To Digital

Port B ( PB 7 .. PB 0 ) : เป็น port 2 ทิศทาง ขนาด 8 บิต โดยสามารถกำหนดให้แต่ละขาของ port สามารถ pull up ภายในอิสระแยกจากกัน ซึ่งแต่ละขาสสามารถรับกระแส sink ได้ 20 mA โดยในแต่ละขาสัญญาณจะถูกใช้งานในฟังก์ชันอื่นๆอีก

Port C ( PC 7 .. PC 0 ) : เป็น port 2 ทิศทาง ขนาด 8 บิต โดยสามารถกำหนดให้แต่ละขาของ port สามารถ pull up ภายในอิสระแยกจากกัน ซึ่งแต่ละขาสสามารถรับกระแส sink ได้ 20 mA และถูกนำไปใช้งานอื่นๆอีก

Port D ( PD 7 .. PD 0 ) : เป็น port 2 ทิศทาง ขนาด 8 บิต โดยสามารถกำหนดให้แต่ละขาของ port สามารถ pull up ภายในอิสระแยกจากกัน ซึ่งแต่ละขาสสามารถรับกระแส sink ได้ 20 mA และถูกนำไปใช้งานอื่นๆอีก

RESET : ขารีเซ็ต

XTAL1 : เป็นขา input ของ OSC

XTAL2 : เป็นขา output ของ OSC



AVcc : ใช้จ่ายไฟให้กับวงจร Analog To Digital ที่อยู่ภายใน MCU

AREF : เป็นขาแรงดันอ้างอิงที่ใช้ในงานในส่วนของวงจร Analog To Digital

AGND : ขากราวด์ของวงจร Analog To Digital

#### 2.2.4.2 โครงสร้างภายใน

โครงสร้างภายในจะประกอบด้วย Register ใช้งานทั่วไปได้ขนาด 8 บิต จำนวน 32 ตัว ที่สามารถเข้าถึงข้อมูลได้ใน 1 clock ซึ่งหมายความว่า MCU สามารถจัดการข้อมูลภายใน Register ใช้งานทั่วไปได้เสร็จภายใน 1 clock ของสัญญาณนาฬิกา

โดย Register R26 – R31 เป็น Register ขนาด 8 บิต จำนวน 6 ตัวสามารถจับคู่เพื่อใช้เป็น Register ขนาด 16 บิต 3 ตัว โดยใช้ชื่อว่า Register X, Y และ Z

ALU จะสนับสนุนการกระทำทางคณิตศาสตร์และ logic ระหว่าง Register กับ Register หรือระหว่าง Register กับค่าคงที่ ซึ่งการเรียกใช้ Register ใช้งานทั่วไป สามารถกระทำได้โดยการอ้างหน่วยความจำภายในที่ตำแหน่ง \$00 - \$1F จำนวน 32 ตำแหน่ง และใน MCU ได้จัดแบ่งให้มี Register ที่ใช้การควบคุมการทำงานของหน่วย input / output ต่างๆ อีก 64 ตำแหน่ง และสามารถเรียกใช้งานได้โดยการอ้างตำแหน่งหน่วยความจำที่ตำแหน่ง \$20 - \$5F

ระบบการทำงานของ microcontroller ใช้หลักการออกแบบของ HAVARD ด้วยการแยกระบบ bus ของ program และ data ออกจากกัน โดยโปรแกรมจะมีการประมวลผลด้วย single level pipelining ซึ่งทำให้ CPU สามารถ fetch และ execute คำสั่งได้ภายใน 1 คาบเวลา

ด้วยคำสั่ง jump และ call แบบ relative ที่สามารถ กระโดดข้ามการทำงานได้ไกลถึง 2K/4K ซึ่งใน 1 คำสั่ง จะใช้รหัสการทำงาน 16 บิตหรือ 1 word โดยทุกครั้งที่มีการ interrupt หรือการข้ามไปทำงานในโปรแกรมย่อย ค่าของ program counter (PC) จะถูกเก็บลง stack ซึ่งจะใช้พื้นที่หน่วยความจำใน SRAM บางส่วนเพื่อทำเป็นพื้นที่ของ stack

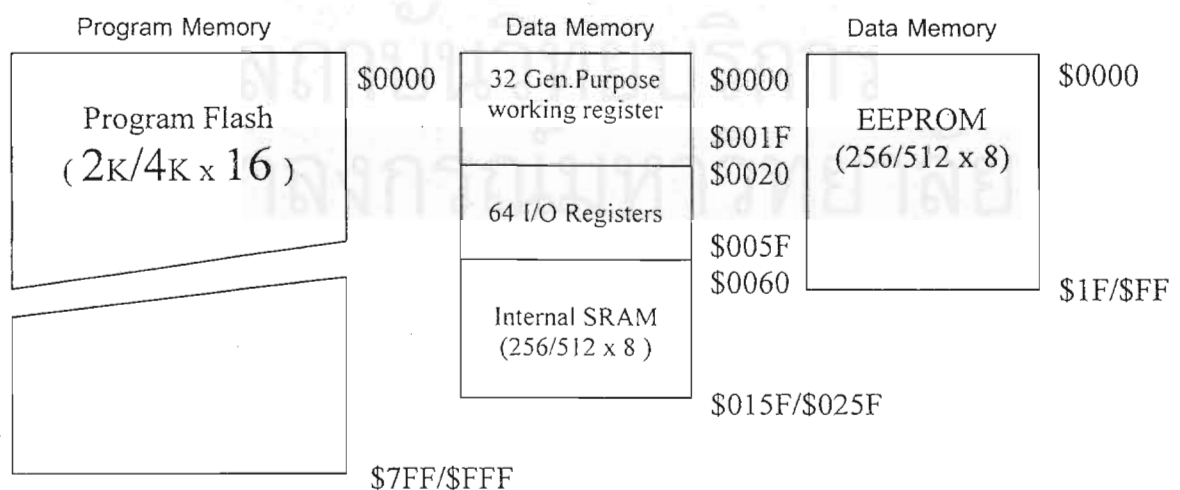


Fig 2.2-16 การจัดโครงสร้างของหน่วย

2.2.4.3 Register ใช้งานทั่วไป

รีจิสเตอร์ทั้งหมดสามารถใช้ชุดคำสั่งเพื่อเข้าถึงได้โดยตรงและจะใช้ช่วงเวลาการเข้าถึงเพียง 1 clock โดย คำสั่ง SBCI , SUBI , CPI , ANDI และ ORI ซึ่งกระทำระหว่าง Register กับค่าคงที่และ Register กับ Register และ คำสั่ง LDI ที่ใช้โหลดค่าคงที่เข้าไปใน Register จะต้องใช้งานกับ Register R16 – R31 ส่วนคำสั่ง SBC , SUB , CP , AND และ OR และคำสั่งใช้งานอื่นๆ สามารถใช้งานได้ใน Register ทั่วไป

2.2.5 การเชื่อมต่อสายสัญญาณจาก microcontroller

สามารถอธิบายได้ด้วยแผนผังข้างล่างนี้

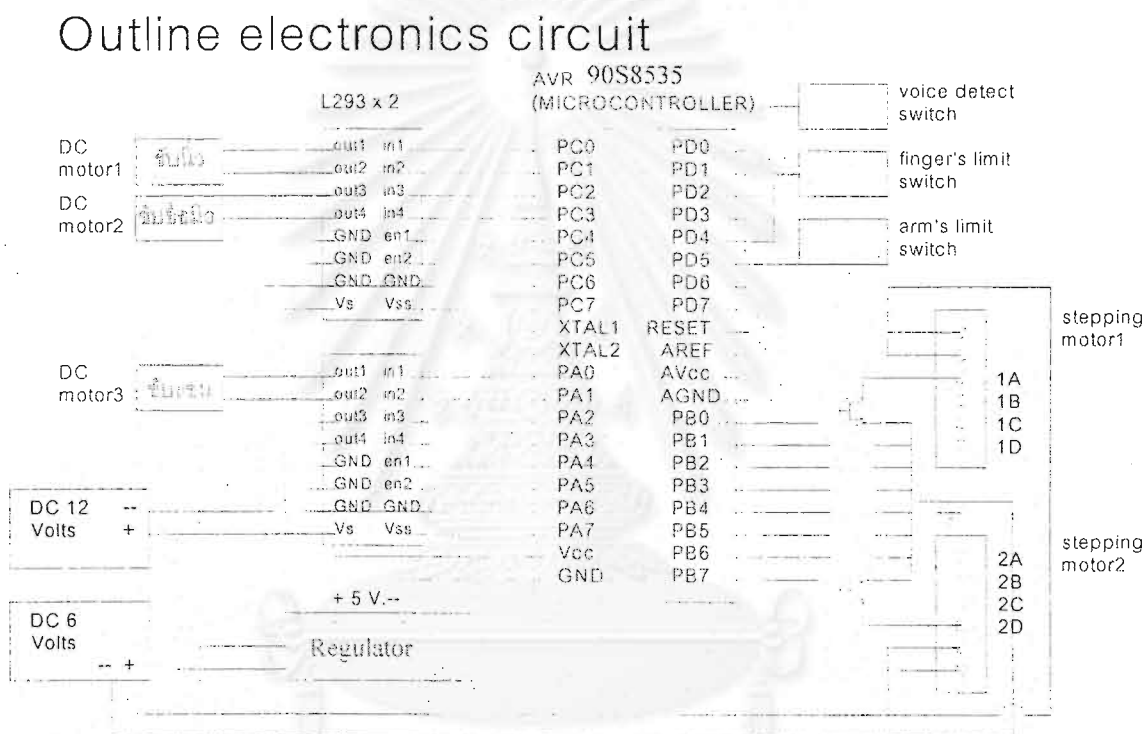


Fig 2.2-17 การเชื่อมสัญญาณจาก microcontroller

สัญญาณขาออกจาก AVR แบ่งออกเป็น 4 ports คือ

- port A รับ-ส่งสัญญาณ analog ซึ่งไม่ได้ใช้ในหุ่นตัวนี้
- port B ส่งสัญญาณดิจิทัล เข้ากับขาเกตของ power MOSFET เพื่อขับ stepping motor โดยใช้สัญญาณ 4 บิตต่อการขับ stepping motor 1 ตัว หุ่นยนต์ตัวนี้ใช้ stepping motor 2 ตัว ดังนั้นจึงใช้ 8 บิต คือ PB0-PB7
- port C ส่งสัญญาณดิจิทัล เข้ากับขา input ของ L293 โดยใช้สัญญาณ 2 บิตต่อการขับ DC motor 1 ตัว หุ่นยนต์ตัวนี้ใช้ DC motor 3 ตัว ดังนั้นจึงใช้เพียง 6 บิต คือ PC0-PC5 โดย PC0-PC1 ส่งสัญญาณให้ DC motor ขับนิ้ว, PC2-PC3 ส่งสัญญาณให้ DC motor ขับข้อมือ, PC4-PC5 ส่งสัญญาณให้ DC motor ขับแขน

- port D รับสัญญาณดิจิทัล จากอุปกรณ์ตรวจรู้ต่างๆโดยใช้ 1 บิต ต่ออุปกรณ์ตรวจรู้ 1 ตัว หุ่นยนต์ตัวนี้ใช้อุปกรณ์ตรวจรู้เพียง 1 ตัว ดังนั้นจึงใช้ 3 บิต คือ PD2, PD4, PD5

หมายเหตุ : Regulator ที่เห็นในรูปหมายถึงรวมถึง Regulator ที่ทำขึ้นมาเอง และ Regulator ที่มีอยู่แล้วใน บอร์ด AVR version 1 ดังนั้นการใช้บอร์ด AVR version 1 นี้สามารถจ่ายไฟ 12 โวลต์ เข้าที่บอร์ดได้เลย Regulator ในบอร์ดจะทำหน้าที่แปลงไฟให้เหลือเพียง 5 โวลต์เพื่อเลี้ยงวงจรในบอร์ดนั่นเอง ส่วน Regulator ที่ทำขึ้นมาเองนั้นทำหน้าที่แปลงไฟจาก 12 โวลต์ เหลือ 5 โวลต์เช่นเดียวกัน แต่เพื่อเลี้ยงวงจรขับ DC motor

## 2.2.6 Regulator

หุ่นยนต์ตัวนี้ใช้ต้นกำลังเป็นแบตเตอรี่ขนาด 6 โวลต์ และ 12 โวลต์ เท่านั้น เนื่องจากวงจรขับ DC Motor ใช้ IC ที่เป็นแบบ TTL ต้องการกำลังไฟที่ 5 โวลต์ ในการเลี้ยงแผงวงจร จึงใช้ regulator เพื่อใช้ในการแปลงความดันไฟฟ้าจาก 12 โวลต์ เป็น 5 โวลต์ โดยใช้ IC7805 เป็นตัวแปลงสัญญาณ ดังรูป.

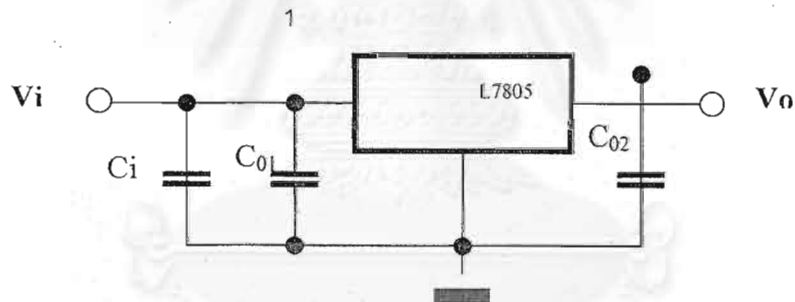


Fig 2.2-18 วงจร Regulator

โดยที่  $C_i = 470 \mu\text{F}$  เป็นแบบ Electrolyte Capacitor

$C_{01} = 470 \mu\text{F}$  เป็นแบบ Electrolyte Capacitor

$C_{02} = 0.1 \mu\text{F}$  เป็นแบบ Ceramic Capacitor

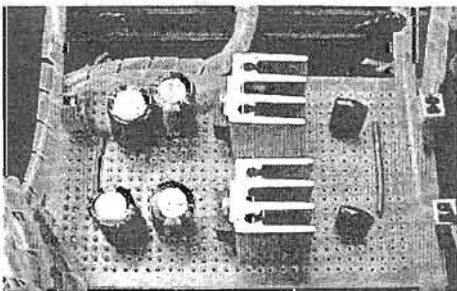


Fig 2.2-19 แผงวงจรปรับแรงดัน

คุณสมบัติของวงจร Regulator ที่ใช้ IC7805 มีดังนี้

แรงดันขาเข้า (Input voltage, $V_i$ )	=	9-12	โวลต์
แรงดันขาออก (Output voltage, $V_o$ )	=	5	โวลต์
กระแสขาออก (Output current, $I_o$ )	=	1	A.

2.2.7 แบตเตอรี่

หุ่นยนต์ตัวนี้ใช้ไฟจากแหล่งจ่ายไฟ (แบตเตอรี่) 2 ขนาด คือ 6 โวลท์ และขนาด 12 โวลท์ โดยมีรายละเอียดการจ่ายไฟให้กับอุปกรณ์ต่างๆดังนี้

BATTERY USAGE

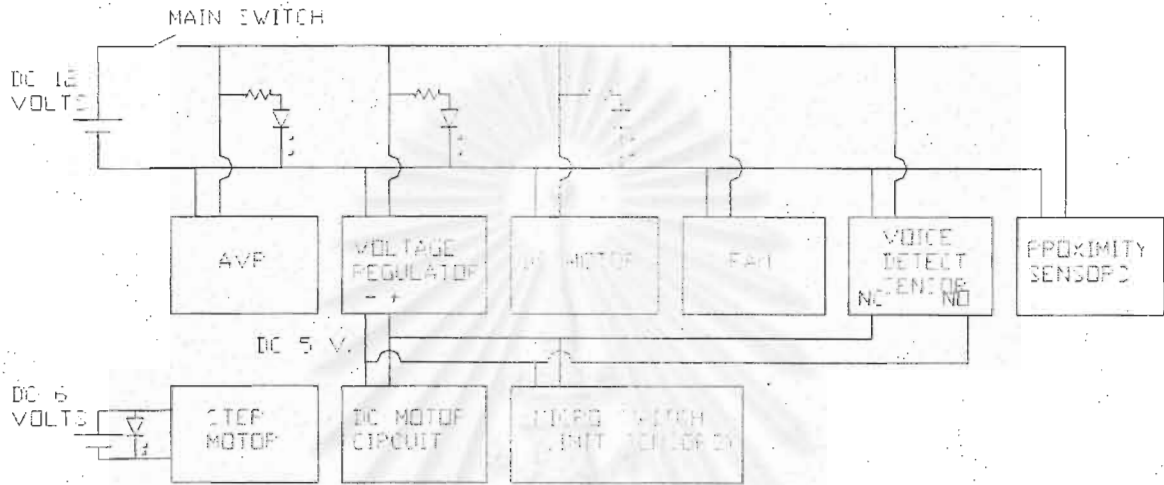


Fig 2.2-20 แสดงการแจกแจงการใช้แบตเตอรี่

แบตเตอรี่ 6 โวลท์ เป็นปัญหาที่สำคัญอีกปัญหาหนึ่ง ในการขับ stepping motor ที่มีไหลดมาก เพราะต้องการแรงบิดสูง เมื่อแรงเคลื่อนไฟฟ้าของแบตเตอรี่ลดลงต่ำกว่าถึงค่าหนึ่งจะไม่สามารถจ่ายกระแสที่เพียงพอกับการขับได้ ต้องนำไปอัดกระแสมาให้เต็ม แล้วจึงจะทำงานได้อีก

จากการทดลองด้วยโปรแกรมที่ใช้ในการแข่งขันหุ่นยนต์บริการให้วิ่งไป 4 รอบการแข่งขัน แบตเตอรี่ 6 โวลท์ จะลดลงเหลือ 5.75 โวลท์ และการทำงานในรอบที่ 5 จะเกิดปัญหาไม่สามารถทำงานได้ดีนัก

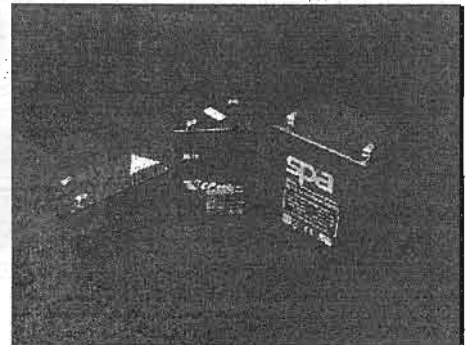


Fig 2.2-21 แบตเตอรี่ที่ใช้ในระบบ

ตารางที่ 2.8 แสดงผลการทดลองเพื่อหาแรงดันเฉลี่ย (โวลท์) ต่อเวลาที่ใช้ (นาที)

ปริมาณ \ รอบที่	1	2	3	4	เฉลี่ยปริมาณที่ใช้ต่อรอบ
เวลาที่ใช้ (นาที)	5.50	5.75	5.75	5.65	5.6625
แรงดันที่วัดได้หลังจากวิ่งครบรอบ (โวลท์)	5.94	5.87	5.81	5.75	0.0625

จากการทดลองพบว่า แบตเตอรี่ 6 โวลต์ เมื่อใช้งานไปจนถึง 5.75 โวลต์ หรือใช้งานไป 22.65 นาที จะไม่สามารถขับ stepping motor ที่มีโหลดขนาดมากกว่าหรือเท่านี้ได้แล้ว

หมายเหตุ : จากการทดลองให้รถวิ่งไป 4 รอบไว้ แบตเตอรี่ 12 โวลต์จะลดลงเหลือ 11.75 โวลต์ แต่แบตเตอรี่ 6 โวลต์ จะเกิดปัญหาไม่สามารถจ่ายไฟเพียงพอได้ก่อนแบตเตอรี่ 12 โวลต์ ดังนั้นเมื่อทำการอัดประจุแบตเตอรี่ 6 โวลต์ ก็อัดประจุแบตเตอรี่ 12 โวลต์ ไปพร้อมกัน ด้วยวิธีนี้จะทำให้สามารถป้องกันปัญหาที่เกิดจากแบตเตอรี่ 12 โวลต์ ต่ำเกินไปได้แน่นอน

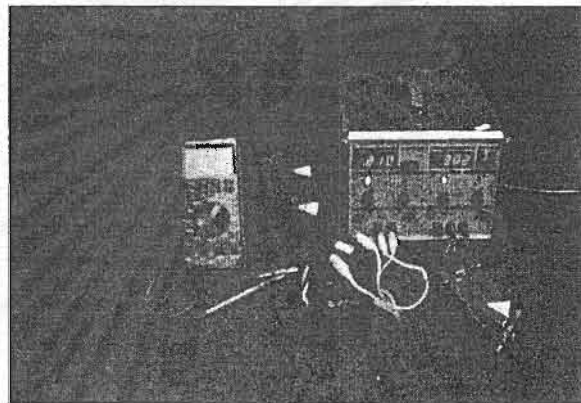


Fig 2.2-22 การอัดแบตเตอรี่

หุ่นยนต์ตัวนี้จะประกอบด้วย แบตเตอรี่ 3 ก้อน คือ 12 โวลต์ และ 6 โวลต์ จำนวน 2 ก้อน (เนื่องจากแบตเตอรี่ 6 โวลต์ ใช้เป็นแหล่งจ่ายไฟขับ stepping motor 2 ตัว ซึ่งต้องการแรงบิดมาก และจากปัญหาที่กล่าวมาแล้ว จึงต้องมีการสำรองแบตเตอรี่ 6 โวลต์ ไว้ 1 ก้อน และยังมีผลพลอยได้เป็นการถ่วงน้ำหนักไว้ข้างหลัง เพื่อมิให้ล้มเนื่องจากน้ำหนักที่มากของมือด้วย)

### 2.2.8 กล่องควบคุม

สิ่งสำคัญอีกประการหนึ่งของการสร้างหุ่นยนต์ให้ทำงานได้ผิดพลาดน้อยที่สุด คือการคำนึงถึงรูปแบบการจัดเก็บวงจร ความเป็นระเบียบเรียบร้อยของวงจรและสายไฟ การแสดงสภาวะการทำงานของวงจรไฟฟ้าในระบบรวมทั้งหมดเรียกว่า *กล่องควบคุม (Control Box)*

วัตถุประสงค์ของการสร้างกล่องควบคุม คือ

- เพื่อให้แผงวงจรและสายไฟถูกจัดวางอย่างเป็นระเบียบทำให้สะดวกในการใช้งานไม่สับสนและเมื่อเกิดข้อผิดพลาด จะสามารถค้นหาสาเหตุได้ง่ายขึ้น
- ช่วยบำรุงรักษาวงจรให้ทำงานได้อย่างเป็นปกติโดยป้องกันการเคลื่อนที่ของแผงวงจร กันความเสียหายเนื่องจากการกระแทก หรือความเสียหายอันเกิดจากความบังเอิญต่างๆ และยังมีกระบายความร้อนของระบบโดยรวม

- การจัดวางระบบควบคุมการทำงานทั้งหมดบนตัวหุ่นทำได้สะดวกมีการกำหนดพื้นที่ใช้งานอย่างชัดเจน
- สามารถตรวจสอบการทำงานของแผงวงจรต่างๆในระบบได้ง่ายเพราะมีการจัดระเบียบแสดงสัญญาณการทำงานจากระบบ

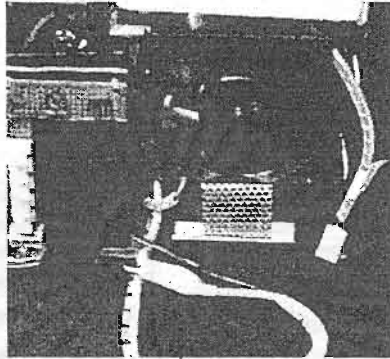


Fig 2.2-23 การติดตั้งพัดลมช่วยระบายความร้อน

ลักษณะของกล่องควบคุม เป็นกล่องสี่เหลี่ยมคล้าย Computer case โดยด้านปลายทั้งสองจะเปิด เพื่อให้มีการระบายความร้อนได้สะดวก รวมทั้งมีการติดตั้งพัดลมเพื่อช่วยในการระบายความร้อนแก่ แผงวงจรให้รวดเร็วขึ้นด้วย ภายในของ กล่องควบคุม จะมี AVR-Card (เปรียบเสมือน Mother board) วางในแนวตั้ง ขนานกับข้าง board และมีการ์ดอื่นๆ วางในแนวขวางประกอบไปด้วย

- วงจรขับ stepping motor 1 วงจร – ใช้ในการขับล้อ
- วงจรขับ DC motor 2 วงจร – ใช้ในการขับนิ้ว ข้อมือ และแขน
- วงจรปรับค่าแรงดันไฟฟ้า (regulator) 2 วงจร
- วงจรวัดค่ากระแสไฟฟ้า 1 วงจร
- แผงเชื่อมโยงไฟฟ้าจากแบตเตอรี่ 1 วงจร
- แผงรับค่าอุปกรณ์ตรวจจับ 1 วงจร –รับค่าจากอุปกรณ์ตรวจจับต่างๆมาเป็น input ให้กับ AVR
- วงจรรับสัญญาณตรวจวัดการทำงานด้วยเสียง
- วงจร proximity sensor

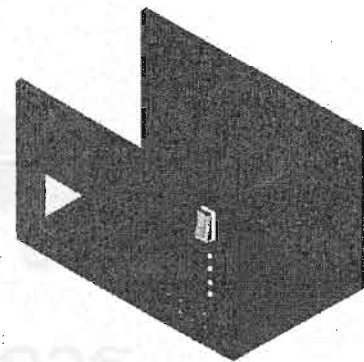


Fig 2.2-24 แบบกล่องควบคุม

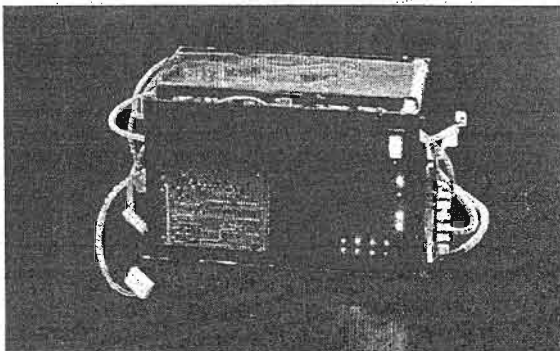


Fig 2.2-25 ไฟแสดงสถานะการทำงานของ board ต่างๆ

ที่ผนังด้านหนึ่งของ กล่องควบคุม จะทำการเจาะรูตามตำแหน่งที่กำหนด เพื่อแสดงสถานะการทำงานของแต่ละ board (วงจร AVR, วงจรควบคุม stepping motor , วงจรควบคุม DC motor, วงจรควบคุมแรงขับเคลื่อนไฟฟ้า และวงจรสวิตช์ควบคุมด้วยเสียง ด้วย LED สีเขียว และ LED สีแดง แสดงสถานะการทำงานของ stepping

motor นอกจากนี้ ยังมีการติดตั้งมีการติดตั้ง main switch เพื่อใช้ควบคุมการเริ่มต้นการทำงานของวงจร ควบคุมของทุกระบบ

จากจุดประสงค์หลักของการสร้าง กล้องควบคุม ซึ่งคือ การพยายามทำทุกอย่างให้อยู่ในสภาวะที่สะดวก ต่อการใช้งานให้มากที่สุด ดังนั้น จึงได้มีการทำบอร์ดเพิ่มเติมขึ้น เพื่อทำหน้าที่ในการเชื่อมต่อสายไฟสำหรับการ เชื่อมต่อส่วนของแผงวงจร กับอุปกรณ์ภายนอก ในที่นี้จะเรียกบอร์ดนี้ว่า บอร์ดเชื่อมต่อสาย

ซึ่งมีวัตถุประสงค์หลักในการใช้งาน คือ

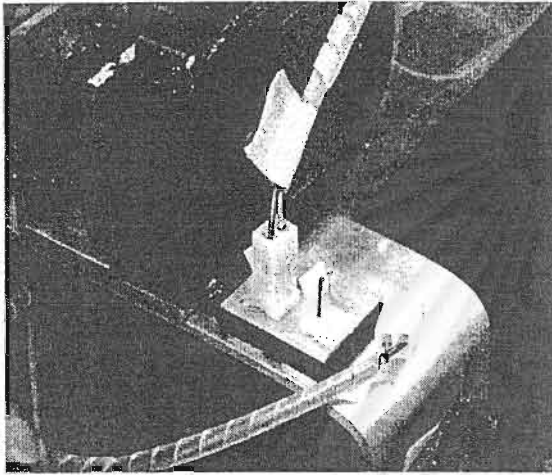


Fig 2.2-26 บอร์ดเชื่อมต่อสายของ DC Motor

- เพื่อให้สายไฟเป็นระเบียบเรียบร้อย สามารถ ตรวจสอบความผิดพลาดได้ง่ายถ้าหากเกิดปัญหาขึ้น
- เพื่อให้สามารถถอด - ประกอบ กล้องควบคุม เข้ากับตัวรถได้ง่าย โดยไม่ต้องเกี่ยวข้องกับบอร์ดแผงวงจร เพราะการถอด - ประกอบ connector บ่อยๆ อาจทำให้ connector หลวมได้ ดังนั้นถ้ามีการใช้บอร์ดเชื่อมต่อสาย การหลวมที่เกิดขึ้น จะไปเกิดที่บอร์ดเชื่อมต่อสาย แทน ซึ่งสามารถแก้ไขหรือถอดเปลี่ยนได้ง่าย

- เพื่อป้องกันไม่ให้เกิดการผิดพลาดในการประกอบสายไฟ เพราะที่สายของบอร์ดเชื่อมต่อ จะมีระบุไว้ว่าต้อง ประกอบสายคู่ใดเข้าด้วยกัน และสายใดต้องเสียบ เข้ากับบอร์ดใด จะถูกยึดด้วยเทปกาว ให้อยู่ใกล้ๆกัน นอกจากนี้ยังสามารถป้องกัน การลืมนเสียบสายไฟบาง เส้นได้

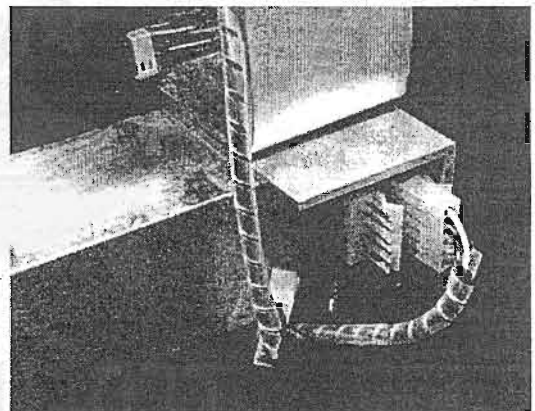


Fig 2.2-27 บอร์ดเชื่อมต่อสายของ stepping motor

### 2.2.9 สรุปอุปกรณ์ที่ใช้ในส่วนระบบไฟฟ้า

ตารางที่ 2.7 อุปกรณ์ที่สำคัญที่ใช้ในส่วนระบบไฟฟ้า

อุปกรณ์	คุณลักษณะ (Specifications)	หน้าที่
แบตเตอรี่ 6 โวลท์ จำนวน 2 ตัว	- 6 แอมป์/ชั่วโมง	เป็นแหล่งจ่ายไฟให้แก่ - stepping motor 2 ตัว

	<ul style="list-style-type: none"> <li>- จ่ายกระแสเริ่มต้นได้ 2.4 แอมป์</li> <li>- หลังจากนั้นจ่ายกระแสได้ไม่จำกัด</li> </ul>	
แบตเตอรี่ 12 โวลต์ จำนวน 1 ตัว	<ul style="list-style-type: none"> <li>- 1.3 แอมป์/ชั่วโมง</li> <li>- จ่ายกระแสได้ไม่จำกัด</li> </ul>	<p>เป็นแหล่งจ่ายไฟให้แก่</p> <ul style="list-style-type: none"> <li>- microcontroller (AVR)</li> <li>- DC motor 3 ตัว</li> <li>- วงจร voltage regulator</li> <li>- พัดลม</li> <li>- วงจร proximity sensor</li> </ul>
AVR board	<ul style="list-style-type: none"> <li>- AVR version 1 ของบริษัท ETT.co.,ltd</li> <li>- ใช้ AVR เบอร์ 90S8535 ของ ATMEL</li> <li>- มี output 4 ports port ละ 10 บิต (port A รับ-ส่ง สัญญาณ analog port B,C และ D รับ-ส่ง สัญญาณ digital)</li> </ul>	<ul style="list-style-type: none"> <li>- ทำหน้าที่เป็นหน่วยควบคุมและประมวลผล(microcontroller)</li> <li>- port A ไม่ได้ใช้ในหุ่นยนต์ตัวนี้</li> <li>- port B ใช้ 8 บิต เพื่อส่งสัญญาณขับ stepping motor 2 ตัว</li> <li>- port C ใช้ 6 บิต เพื่อส่งสัญญาณขับ DC motor 3 ตัว</li> <li>- port D ใช้ 6 บิต เพื่อรับสัญญาณที่มาจากอุปกรณ์ตรวจจับ</li> </ul>
DC motor จำนวน 3 ตัว	<ul style="list-style-type: none"> <li>- ทำงานที่ 12 โวลต์ ทุกตัว</li> </ul>	<ul style="list-style-type: none"> <li>- ขับแขนให้ยกขึ้น-ลง</li> <li>- หมุนข้อมือ</li> <li>- ขับนิวให้บีบ-กางออก</li> </ul>
DC motor driving circuit	<ul style="list-style-type: none"> <li>- ใช้ L293 2 ตัวเป็นตัวขับ</li> </ul>	<ul style="list-style-type: none"> <li>- ขับ DC motor 3 สามารถขับกลับทิศทางได้</li> </ul>
Voltage regulator board	<ul style="list-style-type: none"> <li>- ใช้ IC7805 เป็นตัวแปลงแรงดัน</li> <li>- แรงดันขาเข้า 9 -12 โวลต์</li> <li>- แรงดันขาออก 5 โวลต์</li> </ul>	<ul style="list-style-type: none"> <li>- แปลงแรงดัน 12 โวลต์ ให้เป็น 5 โวลต์ เพื่อเลี้ยง IC ที่วงจรขับ DC motor และวงจร micro switch (limit sensor)</li> </ul>
พัดลม	<ul style="list-style-type: none"> <li>- ทำงานที่ 12 โวลต์</li> </ul>	<ul style="list-style-type: none"> <li>- ระบายความร้อน ติดไว้ที่ด้านหน้าของ กล่องควบคุม</li> </ul>
Stepping motor 2 ตัว	<ul style="list-style-type: none"> <li>- ทำงานที่ 4.5 โวลต์ แต่การใช้งานจริงทำที่ 6 โวลต์</li> <li>- เป็น unipolar stepping motor</li> </ul>	<ul style="list-style-type: none"> <li>- ใช้สำหรับขับล้อทั้ง 2 ข้าง</li> </ul>



<p>วงจรขับ stepping motor</p>	<ul style="list-style-type: none"> <li>- เลี้ยงด้วยไฟ 6 โวลต์ เช่นเดียวกับ stepping motor แต่มีวงจร regulator ในตัวเพื่อแปลงเป็น 5 โวลต์ สำหรับเลี้ยง buffer ในวงจร</li> <li>- ใช้แบบควบคุมการจ่ายกระแสจาก AVR (ไม่ใช้ SM2220 ในการเรียงสัญญาณควบคุม)</li> <li>- ใช้ power MOSFET IRF540</li> </ul>	<ul style="list-style-type: none"> <li>- จ่ายกระแสให้ stepping ที่ละขดลวด หรือ 2 ขดลวด ขึ้นอยู่โปรแกรมที่ตั้ง ให้การขับเป็นแบบ half/full step (1 phase/2 phase)</li> </ul>
<p>วงจรสวิตช์ควบคุม ด้วยเสียง</p>	<ul style="list-style-type: none"> <li>- ใช้ relay 12 โวลต์</li> <li>- จ่ายสัญญาณ 5 โวลต์ และ 0 โวลต์ เข้าที่ขา normally close และ normally open ตามลำดับ</li> <li>- ซ็อกเก็ตกิ่งสำเร็จมาประกอบเป็น วงจร</li> </ul>	<ul style="list-style-type: none"> <li>- ส่งสัญญาณ 0 โวลต์ ให้ AVR เมื่อมีเสียงมากกระตุ้น โปรแกรมจะเขียนให้ เมื่อได้รับสัญญาณ 0 จึงจะเริ่มทำงาน</li> </ul>

## 2.2.10 อุปกรณ์ตรวจรู้ (sensor)

หุ่นยนต์ตัวนี้ใช้อุปกรณ์ตรวจรู้ดังต่อไปนี้

- A. กล้อง CCD
- B. limit switch
- C. สวิตช์ควบคุมด้วยเสียง (voice detect switch)

### A. CCD Camera

โครงการนี้ทำการติดต่อบริษัทรับข้อมูลภาพจากกล้อง CCD เข้ามาประมวลผลเพื่อหาวัตถุเป้าหมายโดยตรง และแสดงตำแหน่งวัตถุที่ตรวจหาพบในสิ่งแวดล้อม กล้อง CCD ที่ใช้ในโครงการนี้ เป็นกล้อง CCD จากบริษัท Camstar รุ่น CIC 741 แบบปรับชัดเตอร์หน้ากล้องด้วยมือ ซึ่งมีความละเอียด 640 x 480 พิกเซล และให้ภาพขาว-ดำ ขนาด 8 บิต หรือ 256 ระดับความเข้มสี ในแต่ละพิกเซล

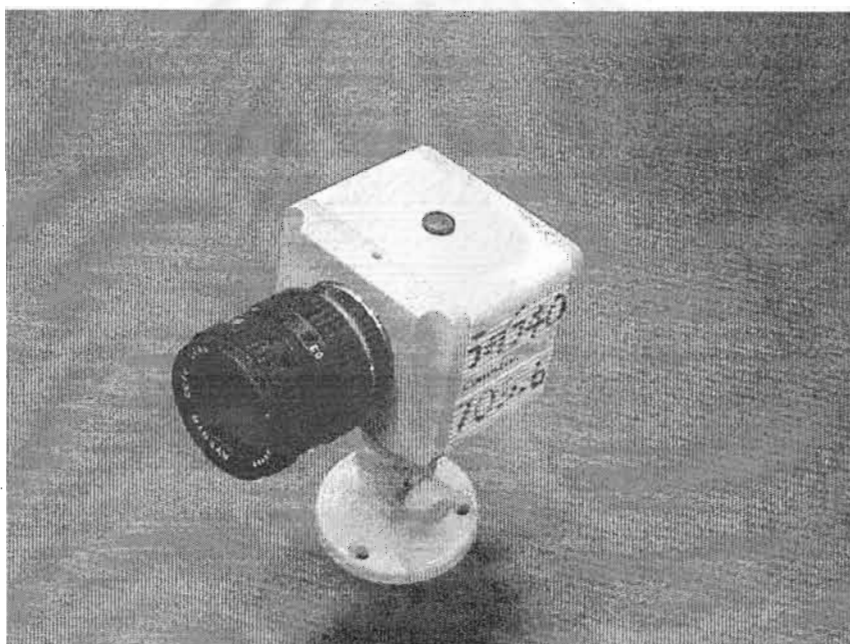


Fig 2.2-28 กล้อง CCD

ในการติดต่อกับกล้องนั้น เราสามารถใช้ โปรแกรม HL-Image จากบริษัท Western Vision Software ได้ ซึ่งภายในโปรแกรม จะบรรจุกระบวนการประมวลผลภาพจำนวนหนึ่งไว้ด้วย และ สามารถสั่งถ่ายภาพจากกล้อง และ บันทึกภาพลงไฟล์ได้

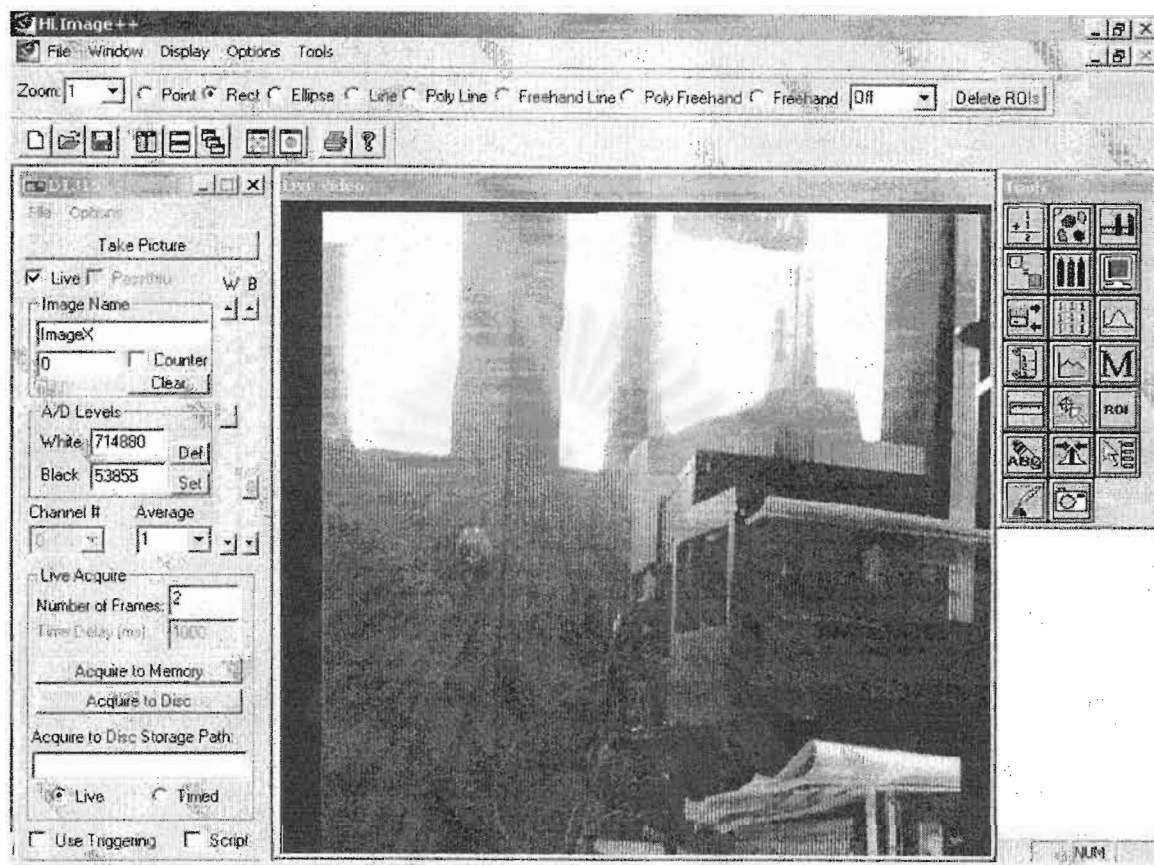


Fig 2.2-29 โปรแกรม HL-Image

## B. Limit switch

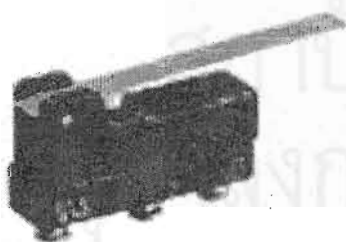


Fig 2.2-30 Limit switch

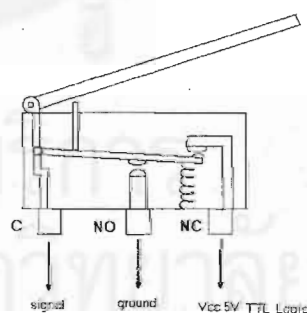


Fig 2.2-31 โครงสร้างของ limit switch และการต่อ logic

อุปกรณ์ตรวจรู้ อีกชนิดหนึ่งที่ใช้ คือ limit switch ติดไว้ 2 ตำแหน่ง คือ ที่ด้านนอกสุดของมือเพื่อตรวจรู้เมื่อนิ้วมาชนขอบมือ และ ที่ขอบของโครงรถด้านหน้า ได้จุดหมุนของแขน เพื่อตรวจรู้เมื่อแขนมาชนขอบรถ limit switch จะเป็นตัวส่งสัญญาณไปที่ AVR โดยสัญญาณดังกล่าวจะไปสั่งการให้มอเตอร์หยุดทำงาน ป้องกันการเสียหายที่จะเกิดขึ้นกับหุ่นได้

ลักษณะการใช้งานของ limit switch เพื่อใช้ในการป้อน logic เข้า controller board ซึ่งจะใช้การป้อนไฟเลี้ยงที่แรงดัน 5 โวลท์ เป็น logic ที่เข้าบอร์ดได้โดยตรง

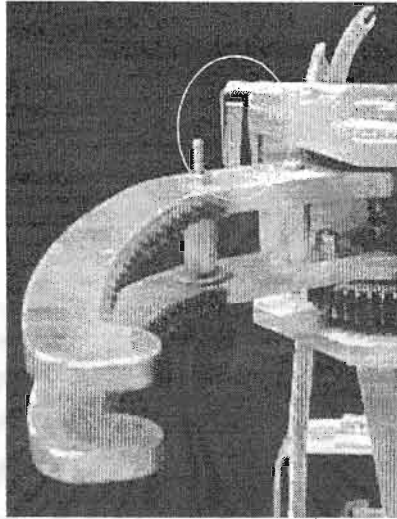


Fig 2.2-32 ตำแหน่งการติดตั้ง limit switch ที่ขอบด้านนอกของมือ

หลังจากการใช้งาน สามารถสรุปข้อดี - ข้อเสียของ limit switch ได้ดังนี้

- ข้อดี
  - limit switch สามารถนำไปใช้งานได้ง่าย โดยสามารถติดตั้งได้โดยตรง ทุกบริเวณที่ต้องการวัด
  - มีราคาถูก หลายชนิดหลายขนาด หลายระดับความแข็งแรงสัมผัส และความไว
- ข้อเสีย
  - การตรวจจู้ของ limit switch จะต้องใช้การกดสัมผัส เข้าไปจึงต้องมีการแตะเกิดขึ้นจริง จึงอาจมีการเคลื่อนที่เนื่องจากการสัมผัส รวมทั้งไม่สามารถตรวจจู้ล่วงหน้าก่อนการสัมผัสจึงอาจเข้าเกินความต้องการ รวมทั้งไม่สามารถตั้งความไว การตรวจจู้ดังเช่น ตัวตรวจจู้ ชนิดอื่นๆ การเปลี่ยนความไวทำได้โดยเปลี่ยนตัว limit switch

### C. สวิตช์ควบคุมด้วยเสียง

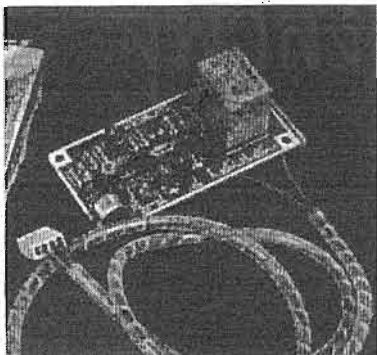
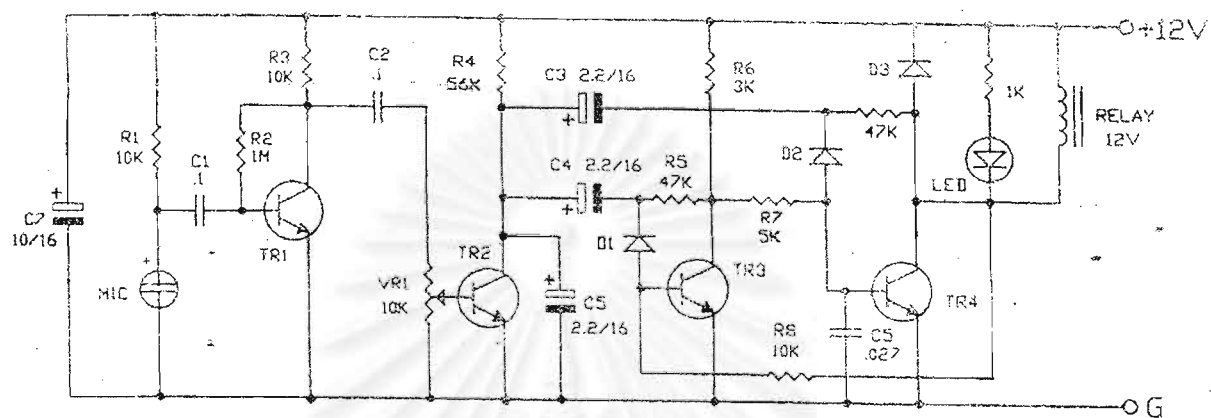


Fig 2.2-33 สวิตช์ควบคุมด้วยเสียง

อุปกรณ์ตรวจจู้ที่ใช้อีกชนิด คือ อุปกรณ์ที่ใช้เป็นที่ทำงานด้วยการตรวจจับเสียง ที่ผ่านเข้ามา โดยเราจะสามารถกำหนดความดัง และความไวของเสียงได้ โดยมีหลักการทำงานคือ จะเริ่มจากบนแผงวงจรมี ไมโครโฟน ซึ่งเป็นลักษณะของคอนเดนเซอร์ ไมโครโฟนมีข้อดี คือในเรื่องของความไว และขนาดที่เล็ก เมื่อมีสัญญาณเสียงเข้ามา ไมโครโฟนจะตรวจจับส่งสัญญาณเสียงออกเข้าสู่ทรานซิสเตอร์เพื่อขยายสัญญาณในขั้นแรก จากนั้นสัญญาณจะถูกส่งผ่าน เข้าสู่ตัวต้านทานปรับค่าได้ เพื่อปรับลดสัญญาณเป็นการควบคุมความไวในการรับเสียงที่เกิดขึ้น สัญญาณจากทรานซิสเตอร์

นี้จะถูกส่งผ่านเข้าหาทรานซิสเตอร์อีกตัว เพื่อไปขับรีเลย์เปิด ปิดไฟ ซึ่งรีเลย์นี้เป็นแบบที่มีทั้ง normally open และ normally close จึงสามารถต่อแรงดัน 5 โวลท์ เข้ากับรีเลย์ เพื่อเป็นสัญญาณ TTL Logic ได้เหมือน limit switch



# D1-D3 = 1N4148

# TR1-TR4 = C458, C828, C1815, C1959, C945

Fig 2.2-34 แสดงลักษณะวงจร สวิตช์ควบคุมด้วยเสียง

หลังจากการใช้งาน สามารถสรุปข้อดี - ข้อเสีย ของสวิตช์ควบคุมด้วยเสียง ได้ดังนี้

- ข้อดี
  - เป็นตัวตรวจรู้เสียง ดังนั้น การสั่งการของตัวตรวจรู้ไม่ต้องสัมผัสเข้าโดยตรงกับสวิตช์
  - สามารถสั่งการได้ในระยะไกลได้ เมื่อดังความไวที่เหมาะสม
- ข้อเสีย
  - สวิตช์ควบคุมด้วยเสียงเป็นตัวตรวจรู้ที่จะถูกรบกวนได้ง่ายจากสิ่งแวดล้อมภายนอก เพียงมีระดับเสียงที่ดังเพียงพอก็จะทำงานทันที
  - ไม่สามารถจำแนกความถี่เสียงได้ ซึ่งถ้าหากแก้ไขจุดนี้ได้ก็จะสามารถแยกว่าเป็นคำสั่งที่ต้องการหรือไม่



## 2.3 โปรแกรมควบคุมและการทดสอบการทำงาน

การโปรแกรมใช้ Instruction ของ AVR ซึ่งสามารถค้นคว้าข้อมูลการใช้คำสั่งต่าง ๆ ได้จากเว็บไซต์ของ Atmel ลักษณะของโปรแกรม ในส่วนของอุปกรณ์ต่าง ๆ คือ

1. โปรแกรมควบคุมการทำงานของ stepping motor ที่ใช้ในการขับเคลื่อนล้อ
2. โปรแกรมควบคุมการทำงานของ DC motor ที่ใช้ในการขับเคลื่อน แขน ข้อมือ และมีมือ
3. โปรแกรมการรับข้อมูลจาก sensor ต่าง ๆ และดำเนินการตามคำสั่งหลังจากที่ได้รับข้อมูล เช่น โปรแกรมการเริ่มต้นการทำงานด้วยเสียง

สามารถสรุปการใช้งานได้ดังนี้

### 2.3.1 โปรแกรมควบคุม stepping motor

การทำงานของ stepping motor จะสามารถหมุนได้โดยการจ่ายไฟหมุนเวียนไปตามขั้วต่าง ๆ รอบๆ ตัวมอเตอร์ และจากการทำงานของวงจรควบคุม stepping motor ซึ่งอาศัยสัญญาณไฟฟ้าจากบอร์ด AVR (โดยสัญญาณออกจาก Port B) มาทริกสัญญาณจ่ายกระแสไฟเวียนไปตามขั้วต่างๆ ดังที่กล่าวข้างต้น สำหรับ stepping motor 1 ตัว ใช้ 4 บิต เนื่องจากมี motor สองตัวคำสั่งการเคลื่อนที่แต่ละครั้งจึงต้องใช้ทั้งหมด 8 บิต โดยบิตคำสั่งของ stepping motor เป็นดังนี้ คือ

บิต	7	6	5	4	3	2	1	0
ขั้วมอเตอร์	1	2	3	4	1	2	3	4
สัญญาณ 1	1	0	0	0	1	0	0	0
สัญญาณ 2	0	1	0	0	0	1	0	0
สัญญาณ 3	0	0	1	0	0	0	1	0
สัญญาณ 4	0	0	0	1	0	0	0	1

ตัวที่ 1

ตัวที่ 2

หมายเหตุ : - ค่า 1 คือ การจ่ายแรงเคลื่อนไฟฟ้า 6 โวลต์ ไปที่แต่ละขั้วของ stepping motor

- การสั่งการเคลื่อนที่ของมอเตอร์แต่ละตัวสามารถทำได้ 2 ทิศทาง คือ

การสั่งสัญญาณ 1, 2, 3, 4, 1, 2, ... เรียงลำดับกันไป และ การสั่งสัญญาณ 4, 3, 2, 1, 4, 3, ...

การโปรแกรมการเคลื่อนที่ต่าง ๆ

- การเดินหน้า (Subroutine 'walkf') โดยจำนวนครั้งของการจ่ายไฟเท่ากับค่าใน r23+1

```

walkf: ldi    r26,0b10001000
        out    PORTb,r26
        rcall  delay1
rotf:   ror    r26
        out    PORTb,r26
        rcall  delay1
        clz
        dec    r23
        brne  rotf
        ret

```

- การเดินถอยหลัง (Subroutine 'walkb') โดยจำนวนครั้งของการจ่ายไฟเท่ากับค่าใน r23+1

```

walkb: ldi    r26,0b00010001
        out    PORTb,r26
        rcall  delay1
rotb:   rol    r26
        out    PORTb,r26
        rcall  delay1
        clz
        dec    r23
        brne  rotb
        ret

```

- การหมุนตามเข็มนาฬิกา (Subroutine 'turnr') เป็นการเรียงลำดับสัญญาณที่ให้แก่มอเตอร์ทั้งสอง เป็นการจ่ายไฟ r23 รอบ (1 รอบต่อ 4 ชุดสัญญาณ : สัญญาณ 1234,สัญญาณ 4321)

```

turnr: ldi    r26,0b10000001
        out    PORTb,r26
        rcall  delay1
        ldi    r26,0b01000010
        out    PORTb,r26
        rcall  delay1
        ldi    r26,0b00100100

```

```

out    PORTb,r26
rcall  delay1
ldi    r26,0b00011000
out    PORTb,r26
rcall  delay1
clz
dec    r23
brne   turnr
ret

```

- การหมุนทวนเข็มนาฬิกา (Subroutine 'turnl') เป็นการเรียงลำดับสัญญาณที่ให้แก่มอเตอร์ทั้งสอง เป็นการจ่ายไฟ r23 รอบ (1 รอบต่อ 4 ชุดสัญญาณ : สัญญาณ 4321,สัญญาณ 1234)

```

turnl: ldi    r26,0b00011000
        out    PORTb,r26
        rcall  delay1
        ldi    r26,0b00100100
        out    PORTb,r26
        rcall  delay1
        ldi    r26,0b01000010
        out    PORTb,r26
        rcall  delay1
        ldi    r26,0b10000001
        out    PORTb,r26
        rcall  delay1
        clz
        dec    r23
        brne   turnl
        ret

```

#### โปรแกรมขับเคลื่อน stepping motor ด้วย SM2220

SM2220 เป็น IC ที่ช่วยในการขับเคลื่อน stepping motor ให้ง่ายกว่าวิธีข้างต้น โดยตัว SM2220 จะทำการเวียนการส่งแรงเคลื่อนไฟฟ้าให้เอง ดังนั้นสัญญาณขับ stepping motor จะใช้สัญญาณ 2 บิตในการสั่งมอเตอร์แต่



ละตัว คือ สัญญาณพัลส์ และ ทิศทาง แต่ในที่นี้จะใช้สัญญาณไฟฟ้าจากบอร์ด AVR (โดยสัญญาณออกจาก Port B) ทั้ง 8 บิตในการควบคุม stepping motor ทั้ง 2 ตัว โดยบิต คำสั่งของ stepping motor เป็นดังนี้ คือ

บิต	7	6	5	4	3	2	1	0
คำสั่ง	DIR2	PULSE2	DIR1	PULSE1	NC	HALF/ FULL STEP	PHASE	RESET

การควบคุมการทำงานของมอเตอร์ทำได้โดยการจ่ายแรงเคลื่อนไฟฟ้า 5 โวลท์ ไปยังบิต คำสั่งที่ต้องการจะให้ทำงาน สำหรับคำสั่ง Pulse จะเป็นการจ่ายแรงดันไฟฟ้า 6 โวลท์ให้แก่ขั้วของมอเตอร์ขั้วใดขั้วหนึ่งก็ได้ แล้วจากนั้น SM2220 ก็จะทำการหมุนเวียนการจ่ายไฟไปแต่ละขั้วตามทิศทางที่ได้มีการสั่งใน DIR (คำสั่ง 0,1 จะมีทิศทางต่างกัน) โดยก่อนจะเริ่มการทำงานทุกครั้งจะต้องสั่งรีเซ็ตก่อนเสมอ หลังจากนั้นจึงจะสามารถทำงานอื่น ๆ ต่อไปได้

วิธีการสั่งให้ stepping motor หมุนทำได้ดังนี้ คือ ขั้นตอนแรกต้องกำหนดทิศทางการหมุนของมอเตอร์ก่อน โดยการให้บิต 5 และบิต 7 เป็น 0 หรือ 1 ตามทิศทางที่ต้องการ หลังจากนั้นจึงให้สัญญาณพัลส์ (คำสั่ง 0 สลับ 1) แก่บิต 4 และบิต 6 เป็นการสั่งให้มอเตอร์หมุนโดยต้องคงค่าของทิศทางเอาไว้

อนึ่ง การสั่งให้ stepping motor หมุน โดย AVR ซึ่งมีความถี่ของการทำงานสูง ทำให้มอเตอร์ ไม่สามารถทำตามคำสั่งได้ทัน จึงต้องมีการหน่วงเวลา (Delay) ระหว่างการพัลส์สัญญาณ โดยค่าเวลาที่เหมาะสมของ stepping motor แต่ละตัวจะมีค่าที่ต่างกัน

การโปรแกรมการเคลื่อนที่ต่าง ๆ

- การเดินหน้า (Subroutine 'walkf) โดยจำนวนครั้งของการจ่ายไฟเท่ากับค่าใน r23+1

```

walkf: ldi    r26,0b00000010
        out   PORTb,r26
        rcall delay1
        ldi   r26,0b01010010
        out   PORTb,r26
        rcall delay1
        clz
        dec   r23
        brne walkf
        ret

```

- การเดินถอยหลัง (Subroutine 'walkb') โดยจำนวนครั้งของการจ่ายไฟเท่ากับค่าใน r23+1

```

walkb: ldi    r26,0b10100010
        out   PORTb,r26
        rcall delay1
        ldi   r26,0b11110010
        out   PORTb,r26
        rcall delay1
        clz
        dec  r23
        brne walkb
        ret

```

- การหมุนตามเข็มนาฬิกา (Subroutine 'turnr') เป็นการเรียงลำดับสัญญาณที่ให้แก่มอเตอร์ทั้งสอง เป็นการจ่ายไฟ r23+1 ครั้ง (SM2220 จะทำการหมุนเวียนการจ่ายกระแส โดยทิศทางของมอเตอร์ต่างกัน)

```

turnr: ldi    r26,0b10000010
        out   PORTb,r26
        rcall delay1
        ldi   r26,0b11010010
        out   PORTb,r26
        rcall delay1
        clz
        dec  r23
        brne turnr
        ret

```

- การหมุนทวนเข็มนาฬิกา (Subroutine 'turnl') เป็นการเรียงลำดับสัญญาณที่ให้แก่มอเตอร์ทั้งสอง เป็นการจ่ายไฟ r23+1 ครั้ง (SM2220 จะทำการหมุนเวียนการจ่ายกระแส โดยทิศทางของมอเตอร์ต่างกัน)

```

turnl: ldi    r26,0b00100010
        out   PORTb,r26
        rcall delay1
        ldi   r26,0b01110010

```

```

out    PORTb,r26
rcall  delay1
clz
dec    r23
brne   turnl
ret

```

### 2.3.2 โปรแกรมควบคุม DC motor ของ แขน ข้อมือ และนิ้วมือ

การควบคุมการขับเคลื่อนของ DC motor กระทำโดยผ่านบอร์ด AVR ซึ่งได้ทำการเขียน instruction ของ ATMEL เอาไว้

เนื่องจากใช้ DC motor 3 ตัว ถ้าต้องการให้สามารถเคลื่อนที่หมุนทั้งไปและกลับได้โดยใช้ IC 293D จึงจำเป็นต้องใช้สัญญาณทริก 2 บิต ต่อมอเตอร์ 1 ตัวทำให้ต้องใช้ทั้งหมดถึง 6 บิต เพื่อขับมอเตอร์ทั้งสามตัวให้เป็นไปตามต้องการ

สัญญาณที่แต่ละขาที่ใช้ขับมอเตอร์ สามารถแสดงเป็นแต่ละบิต ได้ ดังนี้

	มอเตอร์ 3			มอเตอร์ 2			มอเตอร์ 1	
บิต	7	6	5	4	3	2	1	0
สัญญาณ	X	X	0	0	0	0	0	1
การทำงาน	-	-	-	-	-	-	กาง	

บิต	7	6	5	4	3	2	1	0
สัญญาณ	X	X	0	0	0	0	1	0
การทำงาน	-	-	-	-	-	-	บีบ	

บิต	7	6	5	4	3	2	1	0
สัญญาณ	X	X	0	0	0	1	0	0
การทำงาน	-	-	-	-	งย		-	-

บิต	7	6	5	4	3	2	1	0
สัญญาณ	X	X	0	0	1	0	0	0
การทำงาน	-	-	-	-	ก้ม		-	-

บิต	7	6	5	4	3	2	1	0
สัญญาณ	X	X	0	1	0	0	0	0
การทำงาน	-	-	ขึ้น		-	-	-	-

บิต	7	6	5	4	3	2	1	0
สัญญาณ	X	X	1	0	0	0	0	0
การทำงาน	-	-	ลง		-	-	-	-

หมายเหตุ :

- โดย X แทนบิตใช้ในการควบคุม เมื่อใช้ I/O port 8 บิต ของ AVR
- สัญญาณ 1 ของมอเตอร์ 1, มอเตอร์ 2 และ มอเตอร์ 3 แทนการจ่ายแรงเคลื่อนไฟฟ้า 12 โวลท์ให้กับมอเตอร์ 1, มอเตอร์ 2 และ มอเตอร์ 3 ตามลำดับ

ตัวอย่างของโปรแกรมที่ใช้

```
ldi    r26,0b00100000
out    portc,r26
rcall  delay
rcall  delay
rcall  delay
rcall  delay
rcall  delay
```

โปรแกรมข้างบนนี้เป็นการยกแขนลงเพื่อทำการหยิบแก้ว

```
ldi    r26,0b00000010
out    portc,r26
rcall  delay
```

โปรแกรมข้างบนนี้เป็นการบีบนิ้วมือออก

```
ldi    r26,0b00010000
out    portc,r26
rcall  delay
rcall  delay
```

```
rcall delay
rcall delay
rcall delay
ldi r26,0b00000000
out portc,r26
```

โปรแกรมข้างบนนี้เป็นการยกแขนขึ้น จากนั้นจึงหยุดมอเตอร์ทุกตัวเอาไว้

```
ldi r26,0b00100000
out portc,r26
rcall delay
rcall delay
rcall delay
rcall delay
rcall delay
```

โปรแกรมข้างบนนี้เป็นการยกแขนลงเพื่อที่จะวางแก้ว

```
ldi r26,0b00000001
out portc,r26
rcall delay
```

โปรแกรมข้างบนนี้เป็นการกางนิ้วมือเพื่อปล่อยแก้วลง

### 2.3.3 โปรแกรมการรับข้อมูลจากอุปกรณ์ตรวจรู้

อุปกรณ์ตรวจรู้ ที่ติดอยู่ที่มือเพื่อใช้ในการรับส่งข้อมูลมีอยู่ 2 ชนิด คือ

- สวิตช์ควบคุมด้วยเสียง ใช้ในการสั่งให้ระบบเริ่มต้นการทำงาน
- limit switch ใช้ตรวจสอบการกางของนิ้วเพื่อป้องกันการเสียหาย

#### สวิตช์ควบคุมด้วยเสียง

หลักการทำงานของสวิตช์นี้ คือ ในขณะที่จ่ายแรงเคลื่อนไฟฟ้าให้กับวงจรแล้วแต่ยังไม่เกิดเสียง หลอด LED จะดับและสายสัญญาณจะมีแรงเคลื่อนไฟฟ้าเท่ากับ 5 โวลต์ โดยสายสัญญาณนี้ต่อเข้ากับ บิต 2 ของบอร์ด AVR ซึ่งจะทำให้มี logic เป็น 1 แต่ถ้าเกิดเสียงเช่นเสียงปรบมือ หรือ เสียงวัตถุกระทบกันจะทำให้วงจรทำงาน หลอด LED ติด และ สายสัญญาณจะมีแรงเคลื่อนไฟฟ้าเท่ากับ 0 โวลต์ ซึ่งจะทำให้มี logic เป็น 0 ดังนั้นจึงทำให้สามารถสั่งให้บอร์ด AVR ทำงานได้จาก logic ที่กล่าวมาแล้ว

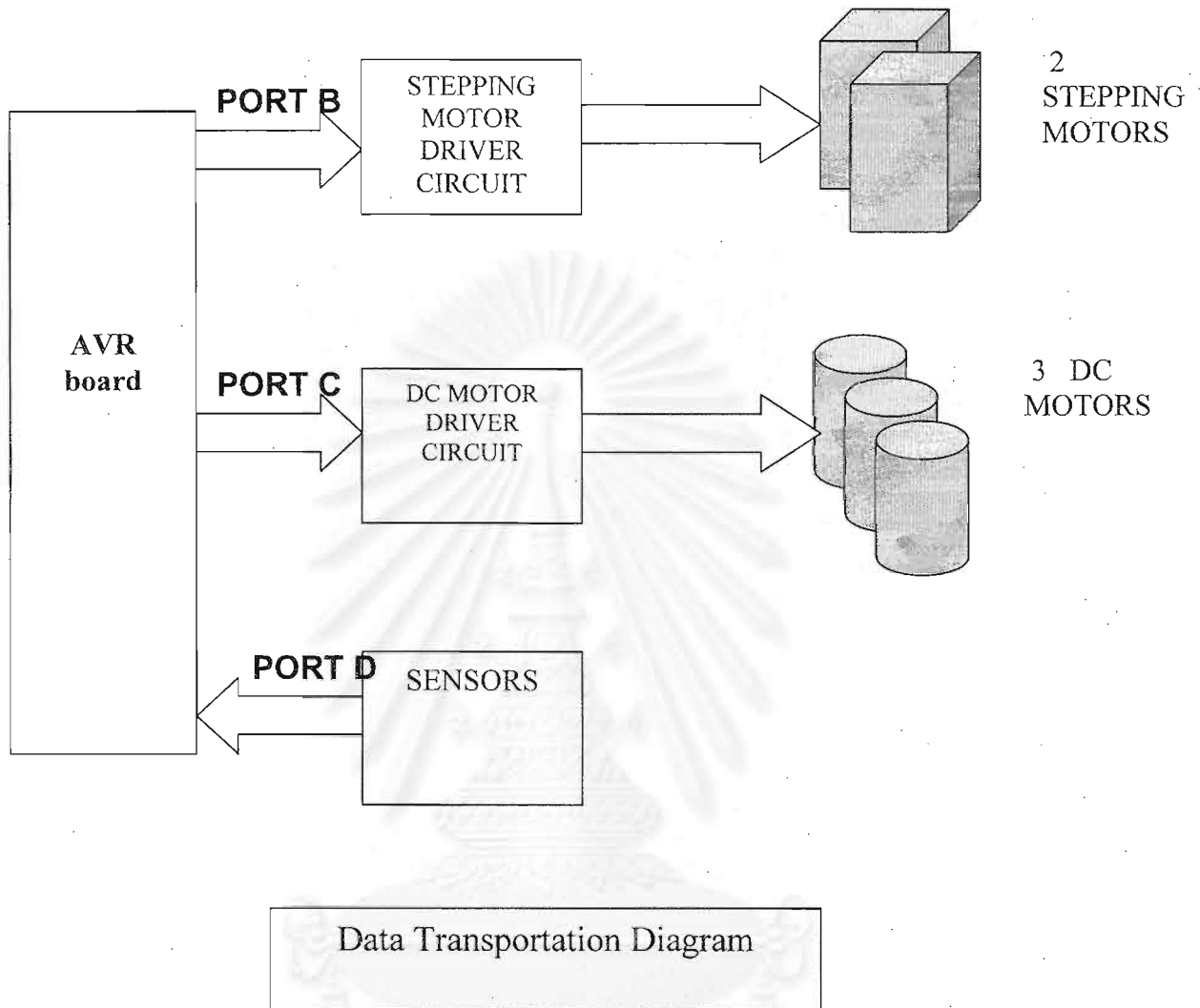


Fig 2.3-1 แสดงการรับ-ส่งข้อมูลของ AVR

ทางคณะวิจัยได้นำหุ่นยนต์คัตแยกขยะรีไซเคิลนี้เข้าร่วมการแข่งขันหุ่นยนต์บริการสมองกล (Partner Robot Contest) เพื่อทดสอบการทำงานของหุ่นยนต์ ในการแข่งขันหุ่นยนต์บริการสมองกล (Partner Robot Contest) แบ่งกิจกรรมเป็น 2 ส่วน คือ ส่วนแรกเป็นส่วนของกิจกรรมบังคับ โดยหุ่นยนต์จะต้องเคลื่อนที่เข้าไปหยิบแก้วน้ำในบริเวณที่กำหนดไว้ แล้วจึงทำการเคลื่อนที่หลบสิ่งกีดขวางไปวางแก้วยังบริเวณที่กำหนด ในขณะที่ส่วนที่ 2 เป็นกิจกรรมพิเศษที่เน้นความสัมพันธ์และประโยชน์ต่อมนุษย์ โดยทางคณะได้ดัดแปลงหุ่นยนต์คัตแยกขยะเป็นหุ่นยนต์สุนัข ซึ่งในการแข่งขันหุ่นยนต์คัตแยกขยะสามารถปฏิบัติกิจกรรมดังกล่าวได้เป็นอย่างดี

## 2.4 บทสรุป

หน้าที่หลักของหุ่นยนต์ คือสามารถเคลื่อนที่ค้นหาและเก็บขยะเพื่อนำไปรีไซเคิล โครงการนี้จะนำเสนอเฉพาะตัวหุ่นและการโปรแกรมเส้นทาง การขยับมือเพื่อยับสิ่งของ การเคลื่อนที่ของแขนและการติดตั้งตัวตรวจจับ เพื่อให้หุ่นตัวนี้ทำหน้าที่ได้โดยสมบูรณ์จะต้องรวมโครงการนี้กับโครงการการใช้โปรแกรมแยกแยะวัตถุและใช้กล้อง CCD (charge-coupled device)

หุ่นยนต์ตัวนี้สามารถแบ่งส่วนประกอบทางกลได้เป็น 2 ส่วน คือ มือ และ โครงสร้างหุ่นยนต์ ซึ่งประกอบด้วยระบบแขน และระบบขับเคลื่อน โดยใช้แบตเตอรี่เป็นต้นกำลัง ทั้งนี้ ทุกระบบถูกควบคุมด้วยโปรแกรมบน micro-controller AVR (ET-AVR V1.0) ซึ่งมีลักษณะการทำงาน ที่ถูกออกแบบให้เป็นบอร์ดเพื่อการศึกษา แต่จะมีคุณสมบัติเทียบเท่ากับบอร์ดควบคุมขนาดใหญ่

มือ มีลักษณะเป็น Gear drive gripper ที่มี 1 องศาอิสระ โดยใช้ DC motor ขนาด 12 โวลท์ขับเคลื่อนข้างหนึ่ง เพื่อส่งกำลังไปที่นิ้วอีกข้างหนึ่งโดยเฟืองตรง ลักษณะการหยิบวัตถุจะคล้ายกับการหนีบของก้ามปู

ส่วนข้อมือเคลื่อนที่ได้เพียง 1 องศาอิสระ คือ pitch ใช้โซ่คล้องกับเฟืองที่ตัวฐานและเฟืองที่กรอมนิ้วโดยใช้ DC motor ขนาด 12 โวลท์ในการขับเคลื่อนโซ่ทำให้ข้อมือเคลื่อนที่ตามไปด้วย มอเตอร์ที่ใช้ในการขับเคลื่อนทรอบด้วยเฟืองนอน จะมีความสามารถในการรักษาตำแหน่งของระบบได้เมื่อไม่มีการจ่ายไฟ (Self lock) ทำให้ไม่มีปัญหาที่เกิดจาก ข้อมือเคลื่อนที่ตกลงมาเมื่อหยุดจ่ายไฟ

รถ มีฐานเป็นรูปสี่เหลี่ยม มีล้อขับเคลื่อนด้วย stepping motor 2 ตัว ด้านข้างขับเคลื่อนแต่ละล้อโดยตรง และล้อหลัง(ช่วยในการทรงตัวของหุ่น) ด้านหลังหนึ่งล้อ ด้านบนมีการติดตั้งกล่องควบคุม ไว้เพื่อบรรจุแผงวงจรควบคุมระบบ และหน่วยแสดงผลการทำงานต่างๆ ด้วยหลอด LED นอกจากนี้ยังมีการเพิ่มพื้นที่การทำงานของแขนโดยการยกระดับของจุดหมุนของแขนและมอเตอร์ขับเคลื่อนให้สูงขึ้น โดยใช้เหล็กโค้งรูปตัว C สองชิ้นเพื่อยกมอเตอร์ขับเคลื่อนและมือ

กล่องควบคุม และ แผงวงจรควบคุมต่างๆ - กล่องควบคุม เป็นที่ติดตั้งแผงวงจรต่างๆ โดยมีหน่วยแสดงผลสถานะการทำงานของ stepping motor ที่ใช้ขับเคลื่อนล้อ ด้วยหลอด LED สีแดง 8 หลอด และ หลอด LED แสดงผลสีเขียว 5 ดวงแสดงว่าแผงวงจรต่างๆ ได้มีการทำงานหรือไม่ ได้แก่ แผงวงจร AVR , แผงวงจรแปลงความดันไฟฟ้า , แผงวงจรสวิทช์ควบคุมด้วยเสียง , แผงวงจรควบคุม DC motor , แผงวงจรควบคุม stepping motor รวมทั้งยังติดตั้งพัดลมระบายความร้อนเพื่อช่วยระบายความร้อนจากวงจรกำลังต่างๆ

ระบบไฟฟ้าและการโปรแกรมบน AVR ภาษาที่ใช้โปรแกรมคือภาษา Assembly เพื่อสั่งให้ AVR ควบคุมแผงวงจรต่างๆ ได้แก่ แผงวงจรควบคุม stepping motor เพื่อควบคุมล้อ ควบคุมการเคลื่อนที่ไปในทิศทางต่างๆ และความเร็วในการเคลื่อนที่ , แผงวงจรควบคุม DC motor เพื่อควบคุมการเคลื่อนที่ของข้อมือ นิ้วมือ และแขนของหุ่นยนต์ ตลอดจนรับค่าจากแผงวงจรสวิทช์ควบคุมด้วยเสียง เพื่อเป็นตัวเริ่มสั่งให้โปรแกรมต่างๆทำงาน

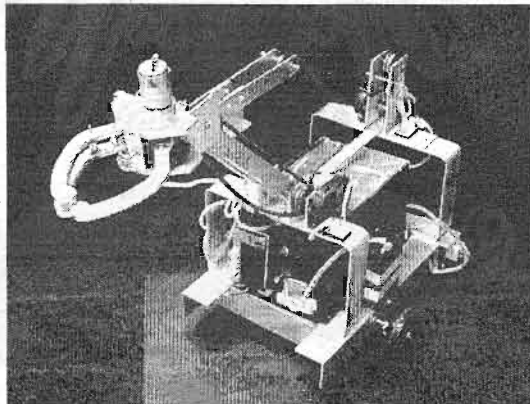
แผงวงจรขับ stepping motor เริ่มจากการใช้ SM2220 เป็นตัวช่วยโปรแกรมในการขับเคลื่อน stepping motor และใช้ทรานซิสเตอร์เป็นตัวขยายกระแส โดยใช้ช่องสัญญาณจาก AVR จำนวน 6 บิตสามารถควบคุม stepping motor ได้ 2 ตัว จากนั้นก็พัฒนามาเป็นการใช้การขับเคลื่อนโดยตรงผ่านสัญญาณจาก AVR สามารถควบคุมความเร็วและทิศทางได้ด้วยอัตราการใช้ pulse และลำดับของ Pulse จาก AVR และมีทรานซิสเตอร์ ชนิด

มอสเฟตเป็นตัวขยายกระแสที่เข้าไปยัง pole ทั้ง 4 ของ motor แต่ละตัวจึงต้องใช้ช่องสัญญาณจำนวน 6 บิตจาก AVR ข้อดีของการขับตรงโดยใช้ ทรานซิสเตอร์ชนิด มอสเฟต คือ ใช้กระแสเกต (เทียบเท่ากับกระแสเบสใน ทรานซิสเตอร์) น้อยมากทำให้มีกระแสเหลือเพียงพอที่จะจ่ายให้กับ LED เพื่อแสดงผลการทำงานของแผงวงจร อีกทั้งยังสามารถจ่ายกระแสให้มอเตอร์ได้มากกว่าด้วย และวงจรมิใช่ IC ชนิดอื่นๆอีก ทำให้ไม่เสียกระแสในวงจร

แผงวงจรขับ DC motor ใช้ L293 เป็นตัวจ่ายกระแสให้กับมอเตอร์ โดย L293 หนึ่งตัวสามารถจ่ายกระแสไฟให้กับ DC motor ได้จำนวน 2 ตัว เราจึงต้องใช้ L293 2 ตัว ในการขับ DC motor 3 ตัวเพื่อใช้ควบคุมนิ้วมือ ข้อมือ และแขนของหุ่นยนต์ การควบคุมลักษณะนี้จะสามารถควบคุมความเร็วของมอเตอร์ได้แต่ในที่นี้ต้องการให้ความเร็วของมอเตอร์มีขนาดคงที่ จึงใช้ควบคุมเพียงทิศทางและหยุดการหมุนของมอเตอร์เท่านั้น

แผงวงจรสวิตช์ควบคุมด้วยเสียง ใช้แทนสวิตช์เริ่มต้นการทำงานของระบบทั้งหมด โดยการใช้เสียง มี ไมโครโฟนเป็นตัวคอนเดนเซอร์ และขยายสัญญาณโดยทรานซิสเตอร์เพื่อไปขับรีเลย์เพื่อเปิดปิดไฟ และสามารถปรับความไวเสียงได้โดยตัวต้านทานปรับค่าได้

ต้นกำลัง ใช้เป็นแบตเตอรี่ขนาด 12 โวลท์ ขับ DC motor ใช้เป็นไฟเลี้ยงให้กับแผงวงจร AVR และพัดลมระบายความร้อนในกล่องควบคุม และ 6 โวลท์ ขับ stepping motor อีกส่วนหนึ่งนำมาต่อ Regulator-7805 เพื่อใช้เป็นทั้งไฟเลี้ยงขนาด 5 โวลท์ L293 ของแผงวงจรขับ DC motor และแผงวงจรตรวจวัดการทำงานด้วยเสียง

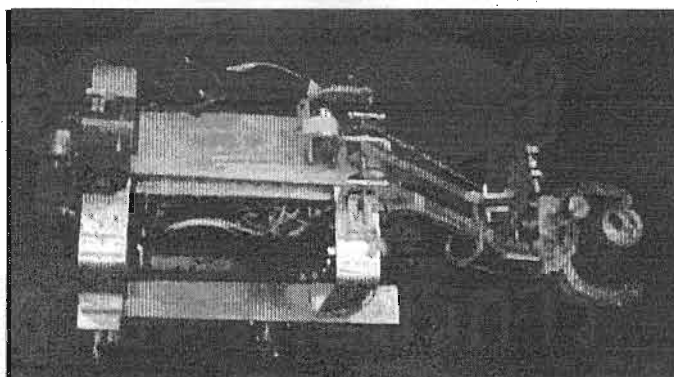


### ปัญหาและข้อผิดพลาดที่เกิดขึ้น

1. โครงสร้างของหุ่นและมือมีน้ำหนักมาก เนื่องจากวัสดุที่ใช้เป็นอะลูมิเนียม เป็นภาระในการขับเคลื่อนด้วย stepping motor ซึ่งโดยปกติ ให้แรงบิดน้อยอยู่แล้ว จึงทำให้เมื่อแรงเคลื่อนไฟฟ้าลดต่ำลงจากมาตรฐานเดิมเพียงเล็กน้อย หุ่นก็จะไม่สามารถเคลื่อนที่ได้
2. ปัญหาในเรื่องกำลังของแรงบิด และแรงดันที่จ่ายให้กับ stepping motor ที่ใช้แรงดันที่จ่ายให้สูงเกินไป จะทำให้ไม่สามารถเคลื่อนที่ได้ เนื่องจากที่แรงดันสูง motor จะขยับในลักษณะเกิดการสั่นที่รุนแรงแต่จะเป็นกำลังที่หักล้างแรงบิดที่ควรจะได้



3. การออกแบบระบบแขน ซึ่งยื่นออกไปข้างหน้ามาก ประกอบกับน้ำหนักของมือ ทำให้จุดศูนย์ถ่วงตกอยู่ใกล้ขอบด้านหน้ามาก จึงต้องสร้างความสมดุลให้กับหุ่น โดยการวางแบตเตอรี่ไว้บริเวณที่เหมาะสม มิฉะนั้นหุ่นจะล้มคว่ำไปด้านหน้า
4. ระบบการยกแขนขึ้นซึ่งจะใช้สายเอ็น ในการส่งกำลังจึงทำให้เกิดปัญหาในเรื่องของการสั่นเพราะสายเอ็นไม่มีความแข็งแรงรวมทั้งหากยกแขนอยู่ในตำแหน่งสูงเกินไป แขนก็จะพับลงมาด้านหลังได้
5. ปัญหาในเรื่องการทำโครงสร้างแขนและมือที่ไม่แม่นยำ ไม่เรียบร้อยจึงทำให้ส่วนของโซ่ควบคุมข้อมือนวม และข้อมือสามารถขยับขึ้นได้ แต่ปัญหาดังกล่าวไม่ค่อยปรากฏ เนื่องจากน้ำหนักของมือที่มากและในขณะที่จับวัตถุ ก็จะถูกกดลงเพียงอย่างเดียว
6. การประกอบล้อหลังไม่ดี ทำให้แนวของล้อหลัง ซึ่งเป็นล้อช่วยพวง บิดไปหลังจากการเลี้ยวแต่ละครั้ง ซึ่งแรงบิดของมอเตอร์ที่น้อยไม่สามารถเอาชนะแรงต้านที่เกิดจากแนวของล้อบิดไปได้ ทำให้หุ่นไม่สามารถเคลื่อนที่ได้ดีหลังจากทดลองทำการเลี้ยว จึงแก้ไขโดยพยายามเลือกที่วางแบตเตอรี่ให้น้ำหนักของหุ่นอยู่ที่ล้อขับเคลื่อนให้ได้มากที่สุด ให้น้ำหนักตกไปที่ล้อช่วยพวงน้อยที่สุด
7. เกิดการไถล ระหว่าง ล้อ กับ พื้น ซึ่งไม่สามารถทำการวัดได้ แต่สำหรับกรณี การเกิด การสั่นไถลระหว่าง มือกับวัตถุ นั้น มีการออกแบบให้มีการรับความคลาดเคลื่อนได้เล็กน้อย จึงสามารถชดเชยได้
8. การส่งข้อมูลจากคอมพิวเตอร์ไปยัง controller จะต้องมีการถอด controller board ออกจากกล่องควบคุม ซึ่งไม่สะดวกเท่าที่ควร อาจทำให้เกิดผลกระทบต่อการเชื่อมต่อ connector ในบริเวณต่างๆได้



### 3. กระบวนการตรวจหาวัตถุเป้าหมาย

ในบทที่ 3 นี้อธิบายถึงกระบวนการในการตรวจหาวัตถุเป้าหมายในสภาวะแวดล้อมต่างๆไปโดยใช้ข้อมูลภาพที่ได้จากกล้อง CCD เราสามารถแบ่งขั้นตอนในการทำงานได้เป็น 2 ขั้นตอนคือ ขั้นตอนในการป้อนข้อมูลวัตถุ และขั้นตอนในการตรวจหาวัตถุในสิ่งแวดล้อมทั่วไป ในขั้นตอนการป้อนข้อมูลของวัตถุที่ต้องการตรวจหาให้แก่นุ่นยนต์นั้น ทำโดยการถ่ายภาพวัตถุเป้าหมายโดยวางบนฉากที่มีสี(ความเข้มแสง)พื้นเดียวกันทั้งแผ่นและมีสีแตกต่างจากวัตถุเป้าหมาย เมื่อเราได้ข้อมูลของวัตถุเป้าหมายแล้วจากนั้นเราจะทำการตรวจหาวัตถุเป้าหมายในสิ่งแวดล้อมทั่วไปโดยคณะวิจัยได้เสนอกระบวนการในการตรวจหาวัตถุเป้าหมาย 2 วิธีซึ่งจะได้อธิบายอย่างละเอียดต่อไป

#### 3.1 ขั้นตอนการป้อนข้อมูลวัตถุ

ขั้นตอนนี้มีวัตถุประสงค์เพื่อที่จะป้อนข้อมูลวัตถุที่ต้องการตรวจหา ซึ่งผลลัพธ์ที่ต้องการคือข้อมูลเฉพาะบริเวณส่วนที่เป็นวัตถุโดยไม่ต้องกรข้อมูลส่วนที่เป็นฉาก โดยมีขั้นตอนคร่าวๆดังต่อไปนี้ เริ่มด้วยการถ่ายภาพวัตถุเป้าหมายบนฉาก (พื้น) ที่มีสีเดียวกันทั้งหมดและแตกต่างจากสีของวัตถุเป้าหมาย จากนั้นทำการปรับแต่งภาพต้นแบบ (Preprocessing) เพื่อลดสิ่งรบกวน (noise) โดยใช้การกรองแบบมัธยฐาน เนื่องจากบริเวณรอยต่อของวัตถุและฉากนั้นจะมีค่าความแตกต่างของสี (ความเข้มแสง) ระหว่างพิกเซลสูงดังนั้นเราจึงใช้ค่าดังกล่าวในการระบุว่าตรงไหนที่เป็นขอบของวัตถุ โดยหาค่าความแตกต่างของทุกพิกเซลบนภาพต้นแบบ จากนั้นกำหนดค่าตัด (Threshold) เพื่อระบุว่าส่วนใดที่เป็นขอบของวัตถุเป้าหมาย เมื่อเราสามารถระบุได้ว่าในภาพต้นแบบส่วนไหนคือวัตถุเป้าหมายแล้วเราสามารถนำข้อมูลนี้มาใช้ในขั้นตอนการตรวจหาวัตถุในสิ่งแวดล้อมต่างๆต่อไป

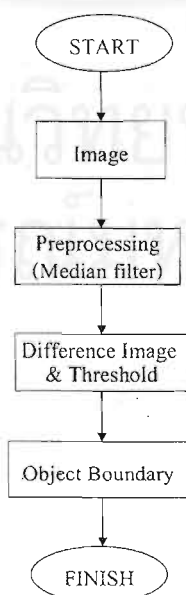


Fig 3.1-1 Flowchart ของขั้นตอนการป้อนข้อมูลวัตถุ

### 3.1.1 การปรับแต่งภาพโดยการกรองแบบมัธยฐาน Preprocessing

ในขั้นตอนแรกเราจะทำการปรับแต่งภาพขั้นต้นโดยการกรองเพื่อลดสิ่งรบกวน (noise) การกรองประเภทมัธยฐาน (Median filter) ซึ่งในบริเวณพื้นที่ที่ค่ามัธยฐาน (Median) หรือค่ากลางของกลุ่มข้อมูลสามารถแทนข้อมูลในบริเวณนั้นได้ โดยเราได้นำหลักการนี้มาใช้ โดยการมองพื้นที่สี่เหลี่ยมจัตุรัสบนภาพที่ละ 9 พิกเซลดังรูป แล้วเลือกเอาค่ากลางของกลุ่มข้อมูลของพื้นที่นี้มาแทนค่ากลางที่ตำแหน่งตรงกลางของพื้นที่สี่เหลี่ยมจัตุรัสนั้นดังแสดงในรูป ซึ่งวิธีการนี้จะทำให้สามารถลดสิ่งรบกวน (noise) ที่เป็นจุดกระจายบนภาพที่เกิดในขั้นตอนการถ่ายได้

$X_{i-1,j-1}$	$X_{i-1,j}$	$X_{i-1,j+1}$
$X_{i,j-1}$	$X_{i,j}$	$X_{i,j+1}$
$X_{i+1,j-1}$	$X_{i+1,j}$	$X_{i+1,j+1}$

( แทนค่าที่ตำแหน่ง  $(i, j)$  ด้วยค่า median ของทั้ง 9 ช่อง )

$$\text{New } X_{i,j} = \text{Median} \{ X_{m,n} \}, \quad (m = i-1, i, i+1; n = j-1, j, j+1) \quad (3.1)$$

เมื่อทำกระบวนการดังกล่าว จนครบทั้งภาพ จะทำให้สิ่งรบกวน(noise) ที่เป็นจุดกระจายบนภาพหายไป และยังทำให้ภาพเนียนขึ้นโดยเฉพาะบริเวณที่มีสี(ความเข้ม)เดียวกัน

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



Fig 3.1-1 รูปจากภาพถ่าย ก่อนการทำการกรองด้วย median filter



Fig 3.1-2 รูปหลังการทำการกรองด้วย median filter

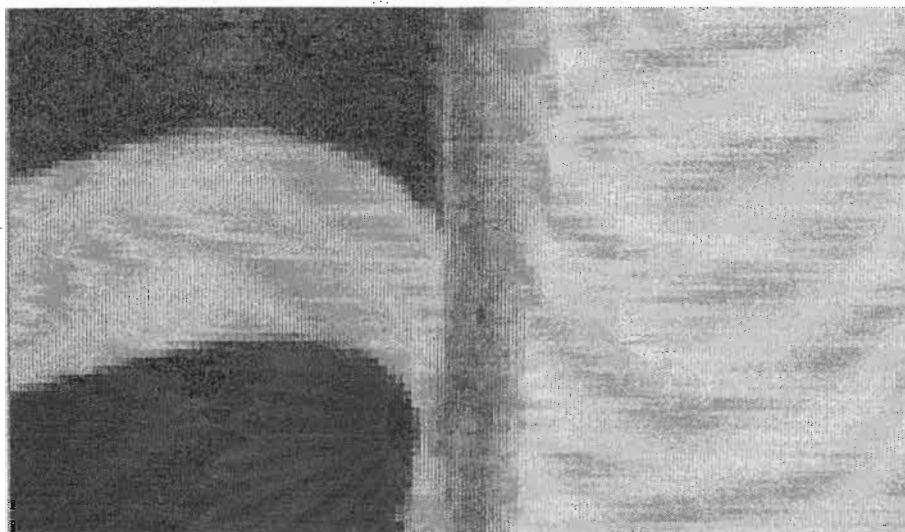


Fig 3.1-3 ภาพขยาย รูปก่อนทำการกรองด้วย median filter



Fig 3.1-4 ภาพขยาย รูปหลังผ่านการกรองด้วย median filter

จะสังเกตได้ว่า การกระจายของสีหลังจากการกรองด้วย median filter แล้วจะมีการกระจายที่สม่ำเสมอมากขึ้น ซึ่งจะทำให้ การแยก Object ในขั้นต่อไปได้ผลดีขึ้นด้วย

### 3.1.2 ค่าความแตกต่างระหว่างพิกเซลและการกำหนดค่า Threshold

หลังจากทำการปรับแต่งพื้นฐานด้วยการกรองแบบมัธยฐานแล้ว เพื่อที่จะระบุว่าส่วนใดในภาพต้นแบบเป็นวัตถุเป้าหมาย เราจะดูค่าความแตกต่างของสี (ความเข้มแสง) ระหว่างพิกเซล เนื่องจากจากมีสีเดียวกันดังนั้นค่าความแตกต่างนี้จึงมีค่าต่ำ ส่วนบริเวณรอยต่อระหว่างวัตถุกับฉากจะมีค่าสูงและเป็นบริเวณที่อยู่นอกสุด

การหาค่าความแตกต่างของสี (ความเข้มแสง) ระหว่างพิกเซลนั้นทำได้ดังนี้

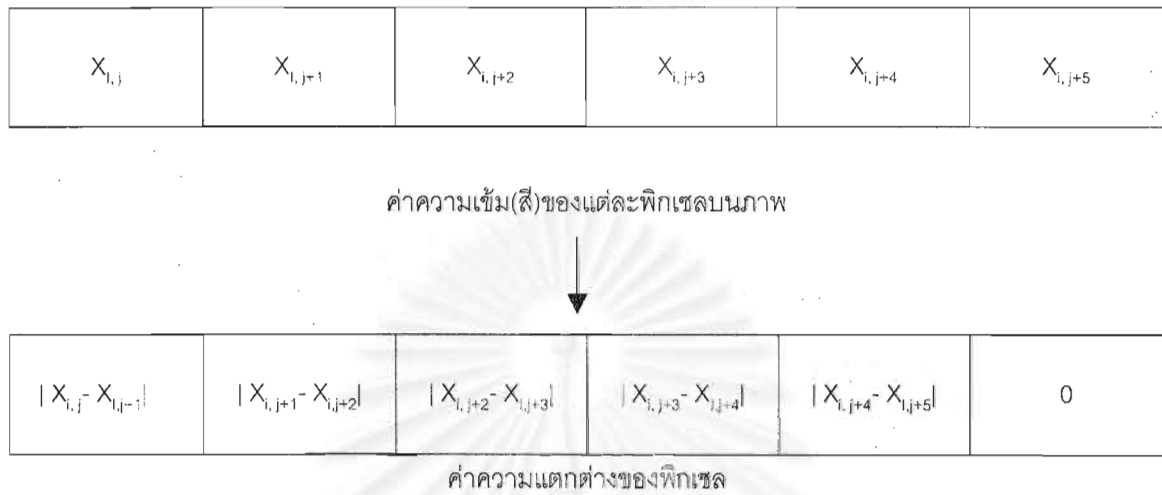
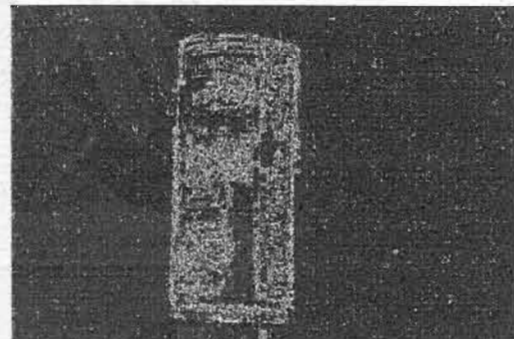


Fig 3.1-5 การคำนวณค่าความแตกต่างของพิกเซล



ภาพต้นแบบ



ภาพของค่าความแตกต่างระหว่างพิกเซล

Fig 3.1-6

ภาพแสดงค่าความแตกต่างของพิกเซลแสดงดังในรูป 3.1-6 (ขวา) ส่วนที่เป็นคือบริเวณที่มีค่าความแตกต่างระหว่างพิกเซลสูง เมื่อเราได้ค่าความแตกต่างของพิกเซลแล้ว สิ่งต่อไปที่เราต้องทำคือกำหนดค่าตัด (Threshold) ที่ใช้ในการระบุว่าตำแหน่งไหนเป็นขอบของวัตถุ



Fig 3.1-7 ภาพแสดงบริเวณที่เรานำค่าของพิกเซลมาพิจารณาเป็นภาพพื้นหลัง ในการคำนวณค่าทางสถิติ

ในการกำหนดค่าตัดนี้ เราได้ตั้งสมมติฐานว่าวัตถุที่เราสนใจอยู่บริเวณกลางภาพดังรูป เราจึงสามารถนำข้อมูลเหล่านี้มาใช้ในการจำแนกว่าส่วนใดคือวัตถุและส่วนใดคือฉากโดยใช้วิธีการทางสถิติดังต่อไปนี้

ในภาพ 3.1-7 บริเวณที่ถูกครีซึ้นนั้นเป็นบริเวณที่เป็นฉากดังนั้น เราจะนำข้อมูลค่าความแตกต่างของบริเวณนี้มาหาค่าทางสถิติ เช่น ค่าเฉลี่ย และ ค่าความเบี่ยงเบนมาตรฐาน (Standard Deviation, SD.) จากความรู้พื้นฐานทางสถิติการกระจายของข้อมูลจะมีลักษณะเป็นแบบโค้งมาตรฐาน (Normal Curve) ดังนั้นในการกำหนดค่าตัด threshold ที่ใช้จำแนกบริเวณที่เป็นวัตถุเราสามารถนำค่าเฉลี่ยกับค่าความเบี่ยงเบนมาตรฐาน (Standard Deviation) ของค่าความแตกต่างที่ได้จาบริเวณฉากนั้นมาประยุกต์ใช้ได้ เช่น กำหนด  $\bar{x} + 5SD$ . เป็นค่า threshold ดังนั้นถ้าค่าความแตกต่างของพิกเซลสูงกว่าค่านี้ให้ถือว่าเป็นขอบของวัตถุ ซึ่งโอกาสที่จะผิดพลาดอันเนื่องจากการเอาบริเวณฉากเป็นขอบวัตถุน้อยมากดังแสดงในภาพที่ 3.1-7 (บริเวณส่วนขาวคือโอกาสผิดพลาด)

การกำหนดค่าตัด threshold นี้ ถ้ากำหนดต่ำไปอาจจะได้วัตถุใหญ่กว่าจริง (มีส่วนที่เป็นฉากรวมเข้าไปด้วย) ถ้ากำหนดสูงไปอาจจะทำให้วัตถุที่ได้เล็กกว่าจริง (บางส่วนของวัตถุถูกเข้าใจว่าเป็นฉาก) ซึ่งค่าที่เหมาะสมอยู่ประมาณที่ ค่าเฉลี่ยบวกกับ 5 - 6 SD. ในการถ่ายภาพเพื่อป้อนข้อมูลนั้นเราสามารถควบคุมเงื่อนไขในการถ่ายภาพได้ง่ายซึ่งทำให้วัตถุมีสีแตกต่างจากฉากค่อนข้างชัดเจนดังนั้นในขั้นตอนการป้อนข้อมูลนั้นจึงสามารถใช้ค่าตัดดังกล่าวในการจำแนกขอบเขตวัตถุซึ่งจากการทดสอบปรากฏว่าได้ผลดีมาก

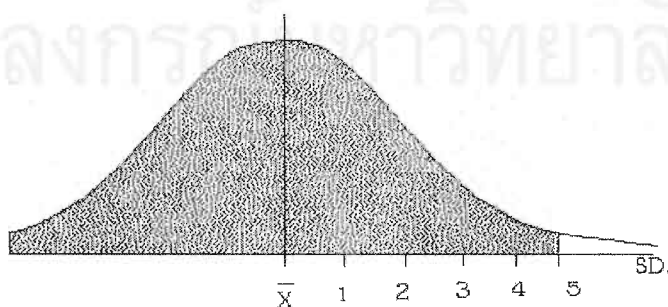


Fig 3.1-8 ภาพแผนภูมิทางสถิติ

### 3.1.3 ขอบเขตวัตถุ Object Boundary

เมื่อเราได้ค่าตัด threshold แล้วเราจะนำมาเปรียบเทียบกับค่าความแตกต่างของพิกเซล (รูป 3.1-6) ซึ่งค่าความแตกต่างที่สูงกว่าค่าตัด threshold จะถือว่าเป็นขอบของวัตถุโดยเราต้องการตำแหน่งเริ่มต้นและตำแหน่งปลายในบรรทัดนั้นเพื่อแสดงขอบเขตของวัตถุดังในภาพ 3.1-9 ทำเช่นนี้ที่ละบรรทัดจากบนลงล่าง แล้วแสดงขอบเขตวัตถุด้วยสีดำจะเห็นได้ว่ายังไม่สมบูรณ์เท่าที่ควร ดังนั้นเรากระทำเช่นเดียวกันอีกตามแนวตั้งจากซ้ายไปขวา(รูป 3.1-10) แล้วนำภาพที่ได้ทั้ง 2 มาทำการ intersection กันจะได้ภาพที่สมบูรณ์ขึ้น

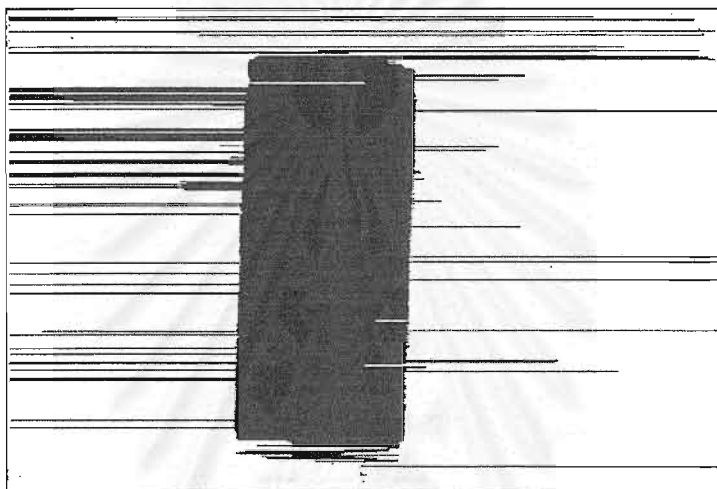


Fig 3.1-9 ภาพหลังการรันโปรแกรมตามแนวแกน x เท่านั้น

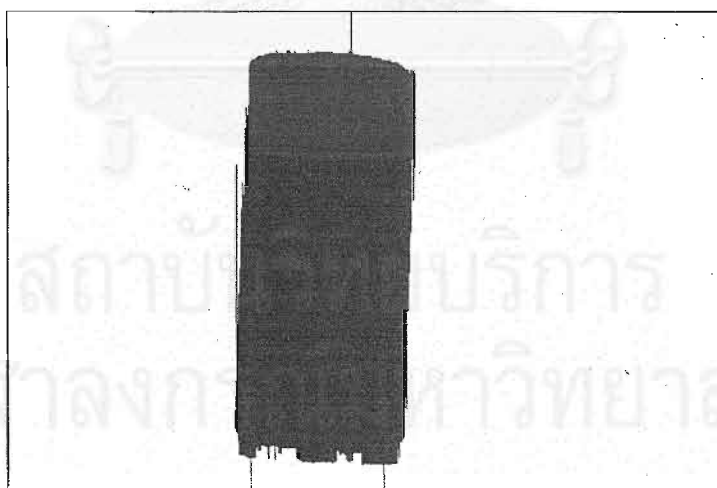


Fig 3.1-10 ภาพหลังการรันโปรแกรมตามแนวแกน y เท่านั้น



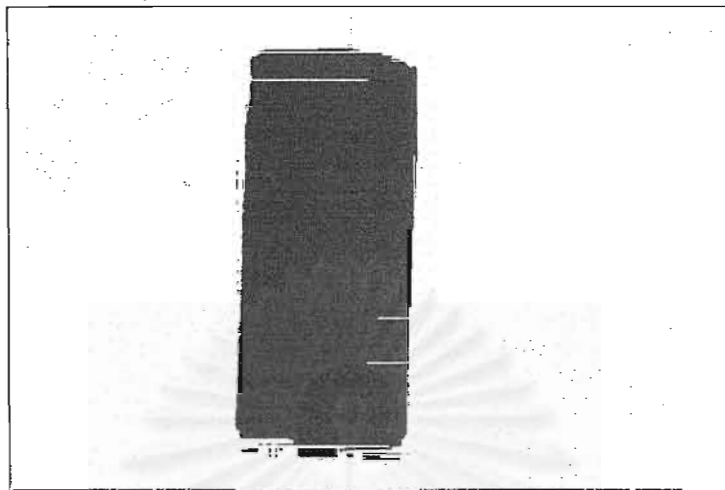


Fig 3.1-11 ภาพหลังจากนำภาพทั้งสองภาพข้างต้นมาอินเตอร์เซกกัน

อย่างไรก็ตามข้อจำกัดของวิธีนี้คือ ค่าของสีที่พื้นหลัง กับ วัตถุ ควรจะมีค่าแตกต่างกันพอสมควร เนื่องจากหากว่าสีของพื้นหลังและสีของวัตถุ ใกล้เคียงกันมากจะทำการแยกวัตถุ ได้ไม่ดี คือจะไม่สามารถกำหนดบริเวณของวัตถุที่ถูกต้องได้อย่างสมบูรณ์ ซึ่งข้อจำกัดดังกล่าวก็ไม่ใช่ปัญหาแต่อย่างไรเนื่องจากเราสามารถกำหนดสภาวะในการเก็บภาพต้นแบบได้ ว่าจะต้องทำการถ่ายภาพวัตถุต้นแบบบนฉากที่มีสีแตกต่างจากวัตถุมากๆ ดังกล่าวไว้แล้ว

สำหรับค่าอ้างอิงนั้น จากการทดลองพบว่า ควรมีความระหว่าง  $\bar{x} + 5SD$ . ถึง  $\bar{x} + 6SD$ . ซึ่งให้ผลที่ต่างกันเพียงเล็กน้อยดังภาพ



Fig 3.1-12 ภาพต้นแบบ

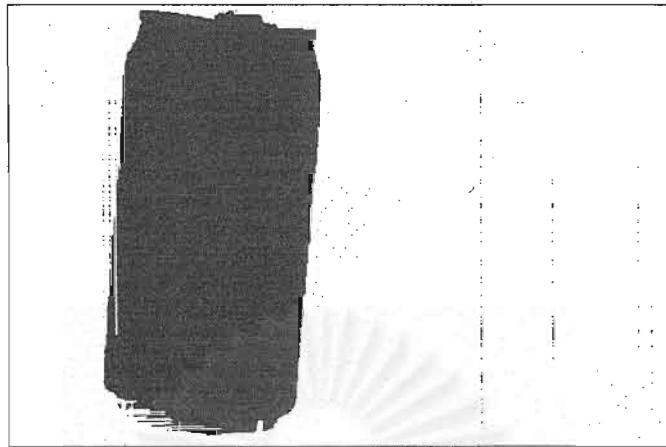


Fig 3.1-13 ภาพหลังจากการทำกรแยก วัตถุต้นแบบออกจากพื้นหลังแล้ว (เลือกค่าอ้างอิง ที่ 5 SD.)

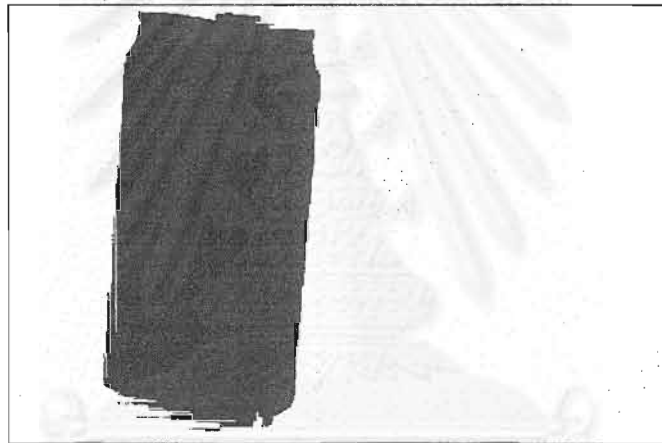


Fig 3.1-14 ภาพหลังจากการทำกรแยก วัตถุต้นแบบออกจากพื้นหลังแล้ว (เลือกค่าอ้างอิง ที่ 6 SD.)



Fig 3.1-15 ภาพวัตถุที่ได้หลังผ่านกระบวนการปิด (Close)

เมื่อพิจารณา ทั้ง 2 ภาพแล้วจะเห็นว่า การเลือกค่าอ้างอิงเท่ากับ 6 SD. จะมี noise ออกมาน้อยกว่า แบบที่เลือกค่าอ้างอิง เท่ากับ 5 SD. แต่ภาพที่ได้ออกมาจะถูกตัดหายไปเป็นบางส่วน ซึ่งเป็นส่วนที่มีความแตกต่างของสีระหว่างวัตถุ กับ พื้นหลัง น้อยเกินไปนั่นเอง

จากภาพที่ได้ จะเห็นว่ามียังได้บริเวณของวัตถุที่ไม่สมบูรณ์ คือ มีรอยขาดแหว่งอยู่ ขั้นตอนต่อไปเป็นการตกแต่งขอบ โดยใช้กระบวนการปิด (Close) ตามที่ได้อธิบายไว้ในภาคผนวก เพื่อปิดรอยขาดแหว่ง ซึ่งจะทำให้ได้ภาพบริเวณของวัตถุต้นแบบที่ค่อนข้างสมบูรณ์มากขึ้น ดังรูป 3.1-15



จุฬาลงกรณ์มหาวิทยาลัย

### 3.2 ขั้นตอนการตรวจหาวัตถุในสภาพแวดล้อมทั่วไป

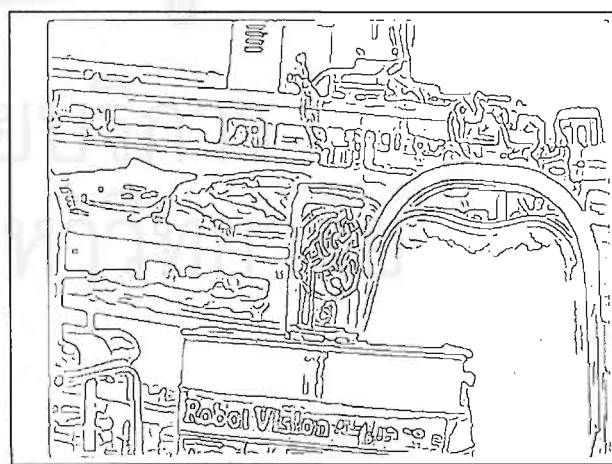
การค้นหวัตถุเป้าหมายภายในสิ่งแวดล้อมด้วยการประมวลผลภาพนั้น โดยทั่วไปเราอาจทำได้ในสองทาง ทางแรกเป็นการค้นหาด้วยข้อมูลของเส้น (Line Information Approach) ทางที่สองคือการค้นหาด้วยข้อมูลของพื้นที่ (Area Information Approach) ข้อมูลของเส้น ได้แก่ ความยาวของเส้น แนวการวางตัวของเส้นตรง รัศมีความโค้งของเส้นโค้ง ขอบมุม จุดตัด รวมทั้ง รูปปิดต่างๆ เช่น วงกลมสี่เหลี่ยม เป็นต้น ข้อมูลของพื้นที่ ได้แก่ ขนาดพื้นที่ ค่าของฟังก์ชันโมเมนต์ สี หรือ ระดับความเข้มแสง เป็นต้น

ในกระบวนการค้นหาด้วยเส้นนั้น เราอาจจะเริ่มต้นด้วยกระบวนการหาขอบของวัตถุในภาพ (Edge Detection) แล้วทำการแปลงข้อมูลขอบของรูปเป็นรหัสเวกเตอร์ จากนั้นทำการตรวจสอบความคล้ายกับรหัสเวกเตอร์ของข้อมูลขอบของวัตถุต้นแบบ เพื่อหาตำแหน่งของบริเวณที่มีความคล้ายมากที่สุดในการค้นหา เป็นต้น

อย่างไรก็ตามพบว่า กระบวนการค้นหาด้วยเส้นนี้ ไม่เหมาะในการใช้ค้นหวัตถุ ในกรณีที่ภาพสิ่งแวดล้อมมีขนาดใหญ่ และมีรายละเอียดมาก เนื่องจากหลังจากการค้นหาขอบของวัตถุในภาพแล้ว จะได้ข้อมูลเส้นเป็นจำนวนมากซึ่งจะต้องใช้เวลาในการคำนวณเปรียบเทียบนานมากจนแทบจะเป็นไปไม่ได้ ถึงแม้ว่าจะทำการลดรายละเอียดด้วยการลบเส้นที่มีขนาดเล็กกว่าค่าหนึ่งออกไปแล้วก็ตาม ดังตัวอย่างในรูป นอกจากนี้การเปรียบเทียบข้อมูลเส้นที่ได้มากก็เป็นเรื่องยากเนื่องจากเส้นที่ได้ส่วนมากไม่เป็นเส้นปิด (เส้นที่แสดงถึงขอบเขตวัตถุ) ดังนั้นการนำข้อมูลเส้นมาใช้ จึงเป็นเรื่องที่ลำบาก ทางคณะจึงได้พัฒนากระบวนการค้นหวัตถุบนพื้นฐานของพื้นที่ (Area Based Approach) โดยได้เสนอกระบวนการในการตรวจหาวัตถุเป้าหมาย 2 วิธีคือกระบวนการค้นหาโดยใช้กลุ่มสี และ กระบวนการค้นหาด้วยรูปร่างวัตถุ ซึ่งจะได้อธิบายรายละเอียดต่อไป



ภาพวัตถุในสิ่งแวดล้อมทั่วไป



ภาพข้อมูลขอบซึ่งประกอบไปด้วยเส้นจำนวนมาก

Fig 3.2-1

### 3.2.1 กระบวนการค้นหาวัตถุโดยใช้กลุ่มสี

กระบวนการนี้ได้แนวความคิดมาจาก กระบวนการค้นหาวัตถุขั้นแรกของมนุษย์ ซึ่งเริ่มต้นด้วยการนึกว่า วัตถุที่ต้องการค้นหา มีสีอะไรบ้าง จากนั้นจะกวาดสายตาค้นหาอย่างรวดเร็วเพื่อมองหาจุดที่มีสีดังกล่าว จากนั้นค่อยเพ่งมองรายละเอียดเพื่อดูว่าเป็นวัตถุเป้าหมายที่ถูกต้องหรือไม่ ซึ่งกระบวนการดังกล่าวในระบบการมองของมนุษย์นั้น เป็นไปโดยอัตโนมัติ และ รวดเร็วมาก

กระบวนการในโปรแกรมจึงถูกพัฒนาขึ้นในทำนองเดียวกันนั้น โดยอาศัยหลักการว่า ภายในวัตถุหนึ่งๆ จะมีสีที่เป็นจุดเด่น กล่าวคือเป็นสีหลักๆ อยู่จำนวนหนึ่ง ถ้าเราทำการตรวจหาสีดังกล่าวจากวัตถุต้นแบบได้ว่าเป็นสีใดบ้าง แล้วทำการค้นหาในสิ่งแวดล้อมก็จะได้บริเวณที่มีโอกาสมีวัตถุอยู่ โดยเราสามารถลดโอกาสของความผิดพลาดได้ด้วยการค้นหาบริเวณที่มี กลุ่มสีหลักทั้งหมดของวัตถุต้นแบบอยู่ในบริเวณใกล้เคียงกันมากๆ เท่านั้น ก็จะทำให้สามารถตัดบริเวณที่บังเอิญมีสีคล้ายกับวัตถุต้นแบบเพียงบางสีออกไปได้ ซึ่งกระบวนการนี้เหมาะอย่างยิ่งสำหรับการค้นหาวัตถุที่มีลวดลาย และ สีสิ้นมาก เช่น ระเบิดน้ำอัดลม เนื่องจากโอกาสที่บริเวณสิ่งแวดล้อมจะมีสีสิ้นคล้ายกับวัตถุทั้งหมดมีน้อยมาก อย่างไรก็ตาม การที่กล้องรับภาพที่ใช้เป็นกล้องขาว-ดำ 8 บิต จะทำให้ใช้คุณสมบัติในการค้นหาด้วยวิธีนี้มีประสิทธิภาพน้อยลงไปบ้าง เนื่องจากข้อมูลที่ได้ไม่ใช่สีแท้จริงของวัตถุ



Fig 3.2-2 ภาพแสดงบริเวณที่เป็นวัตถุในภาพต้นแบบจากขั้นตอนการป้อนข้อมูลวัตถุ

เมื่อได้บริเวณของวัตถุต้นแบบที่สมบูรณ์จากขั้นตอนการป้อนข้อมูลแล้ว เราสามารถนำข้อมูลนี้ไปใช้ในการค้นหาวัตถุต่อไป ในกระบวนการนี้เราจะเก็บข้อมูลของวัตถุเป้าหมายเพื่อหาค่าช่วงความเข้มสีที่มากที่สุดเรียงลำดับเก็บไว้เป็นข้อมูล แล้วในการหา ก็ทำการหาบริเวณที่มีกลุ่มสีเหล่านั้นในภาพที่ได้จากสิ่งแวดล้อมโดยใช้ช่วงสีเดียว อาจจะมีได้หลายบริเวณดังนั้นจึงวิเคราะห์โดยใช้ช่วงถัดไปที่มีขนาดรองลงมา

มาถึงขั้นนี้โปรแกรมก็พร้อมที่จะทำการค้นหาวัตถุอย่างคร่าวๆ (Roughly Search) ในสิ่งแวดล้อมแล้ว เมื่อโปรแกรมรับภาพสิ่งแวดล้อมที่มีวัตถุต้นแบบอยู่เข้ามา โปรแกรมก็จะนำข้อมูลสีที่มากที่สุดที่เก็บไว้ตอนแรก มาทำการค้นหาในภาพที่ได้มาจากสิ่งแวดล้อม บริเวณที่สอดคล้องกับช่วงสีดังกล่าวจะถูกเปลี่ยนเป็นสีดำเพื่อให้สามารถประมวลผลได้สะดวก ถ้าสมมติว่าเรากำหนดช่วงสีที่มีความถี่สูงสุดที่เก็บมาจากภาพต้นแบบ ทั้งหมด 5 ช่วง เราจะได้ภาพไปนารี 5 ภาพ ดังตัวอย่างในรูป

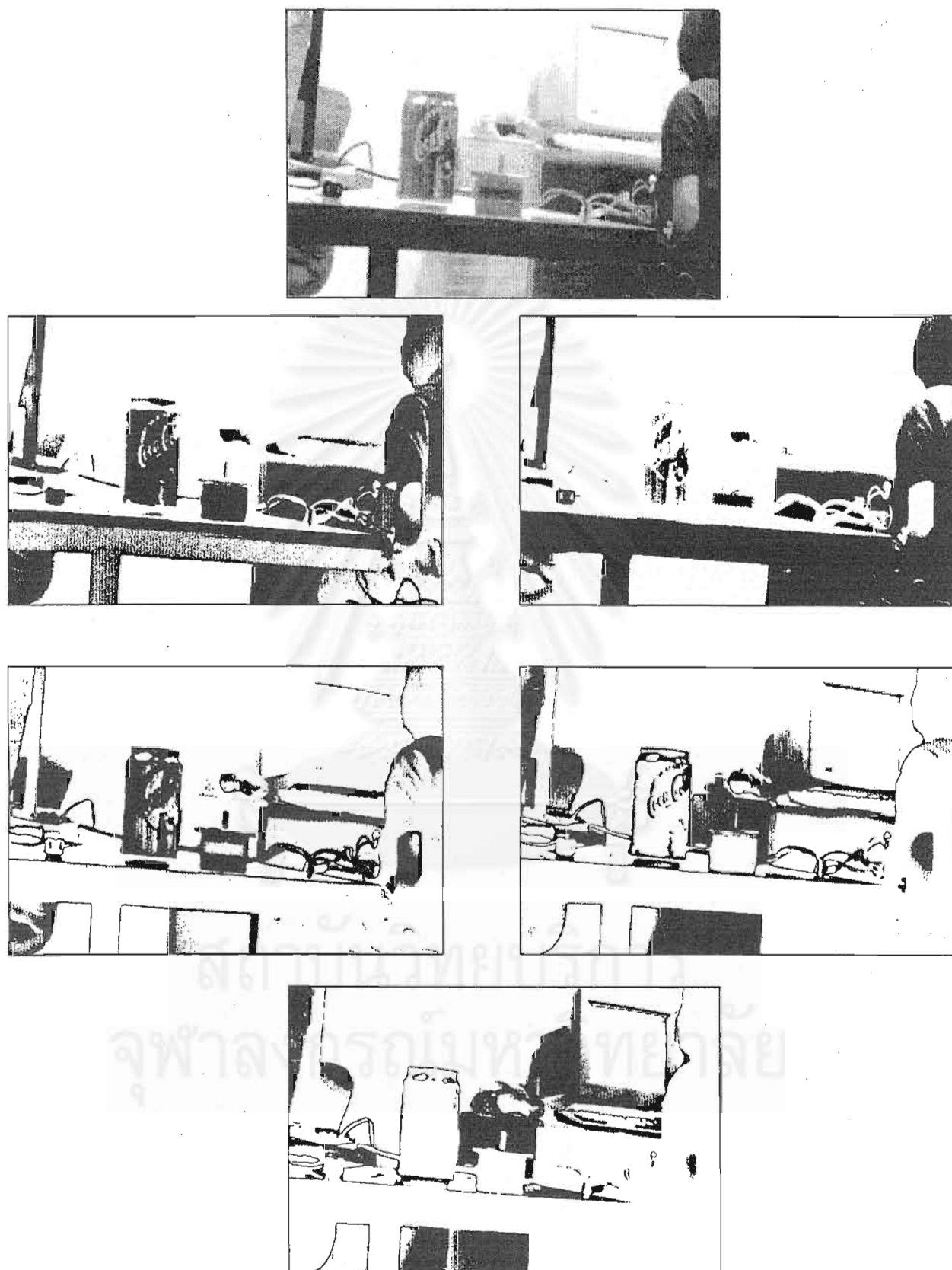


Fig 3.2-3 ภาพโมโนกรัฟที่ได้จากการค้นหาลีแต่ละช่วงสี

ในขั้นตอนนี้เราจะสามารถทำการหาบริเวณซึ่งมีช่วงสีทั้ง 5 อยู่ใกล้กันพร้อมกันทั้งหมดได้โดยง่าย โดยการประมวลผลภาพไบนารีทั้ง 5 ภาพดังกล่าว โดยการไล่หาบริเวณที่มีจุดสีดำในภาพไบนารีทั้ง 5 พร้อมกันในบริเวณใกล้เคียงกัน ผลลัพธ์ที่ได้จะเป็นภาพไบนารีอีกภาพหนึ่งซึ่งแสดงจุดที่มีโอกาสมีวัตถุอยู่ด้วยสีดำ ซึ่งในขั้นนี้เราจะได้ภาพไบนารีซึ่งมีสีดำมากที่สุดที่บริเวณวัตถุเป้าหมาย

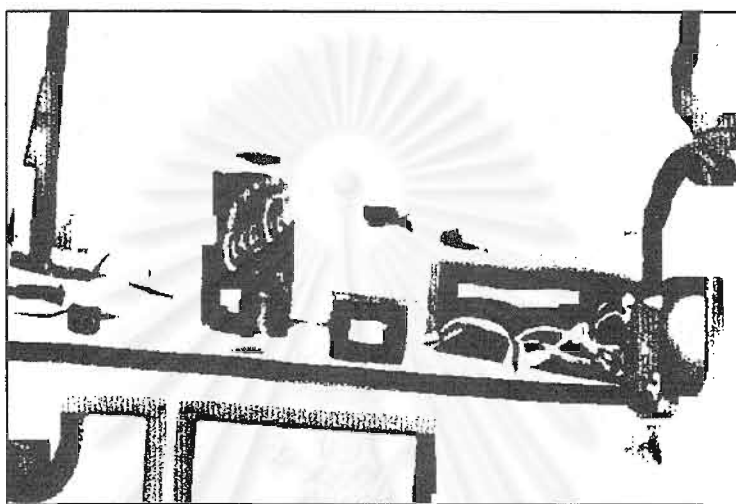


Fig 3.2-4 ภาพไบนารีที่ได้จากการค้นหาบริเวณที่มีช่วงสีทั้ง 5 ใกล้ๆกัน

ขั้นต่อไปเราจะทำการหาบริเวณที่มีจุดซึ่งมีโอกาสมีวัตถุอยู่มากที่สุด โดยการใช้กรอบหน้าต่างสี่เหลี่ยมจตุรัสวิ่งไปภายในรูป เพื่อค้นหาบริเวณที่มีจุดสีดำมากที่สุด โดยเคลื่อนกรอบนี้ไปแล้วนับจุดดำในกรอบนั้น เมื่อเจอบริเวณนั้นเราจะถือว่าจุดกึ่งกลางของกรอบหน้าต่างนั้น เป็นบริเวณวัตถุ



Fig 3.2-5 ภาพแสดงกรอบหน้าต่างที่ได้จากการค้นหาบริเวณที่มีจุดสีดำมากที่สุด

จากนั้นเราจะทำการขยายบริเวณจากจุดกึ่งกลางนั้นออกไปเพื่อให้ครอบคลุมตัววัตถุเพื่อใช้เป็นบริเวณในการค้นหาอย่างละเอียดต่อไป กระบวนการในการขยายกรอบจะใช้หลักการ โดยเพิ่มขนาดกรอบทีละหนึ่งหน่วยไปในทิศทางที่ทำให้ สัดส่วนพื้นที่สีดำต่อพื้นที่สีขาวสูงสุด และ หยุดในกรณีที่ไม่ว่าจะขยายกรอบไปในทิศทางใดๆ จะทำให้สัดส่วนพื้นที่สีดำต่อพื้นที่สีขาวน้อยกว่าค่าๆ หนึ่ง มาถึงขั้นนี้เราก็จะได้บริเวณที่คาดว่ามียัตถุอยู่ภายในดังแสดงในรูป



Fig 3.2-6 ภาพผลลัพธ์สุดท้ายในการค้นหาวัตถุ

กระบวนการในการค้นหาวัตถุโดยใช้ข้อมูลกลุ่มสีนี้สามารถค้นหาวัตถุในสภาพแวดล้อมที่มีสภาพใกล้เคียงกับสภาพของภาพต้นแบบ (ในขั้นตอนการป้อนข้อมูลวัตถุ) ได้อย่างรวดเร็วและถูกต้อง เราสามารถเพิ่มความแม่นยำถูกต้องโดยนำข้อมูลภาพที่เป็นภาพสี (Colour Image) ก็จะสามารถเพิ่มศักยภาพของวิธีการนี้ได้เป็นอันมาก แต่วิธีการนี้ยังคงขึ้นอยู่กับสภาพแวดล้อมอยู่ดีและยังไม่สามารถระบุขนาดของวัตถุได้อย่างถูกต้อง วิธีการนี้จึงเหมาะในการค้นหาว่ามีวัตถุที่ต้องการค้นหาอยู่ในสิ่งแวดล้อมหรือไม่ หรือ หาตำแหน่งคร่าวๆของวัตถุ

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



### 3.2.2 กระบวนการตรวจหาวัตถุโดยใช้ข้อมูลรูปร่างวัตถุ

ในกระบวนการตรวจหาวัตถุโดยอาศัยข้อมูลรูปร่างของวัตถุ เราจำเป็นที่จะต้องค้นหาบริเวณที่มีโอกาสที่จะเป็นวัตถุก่อน ซึ่งอาจจะเป็นวัตถุอะไรก็ได้ในสิ่งแวดล้อมนั้น จากนั้นจึงนำวัตถุที่พบในสิ่งแวดล้อมไปเปรียบเทียบกับวัตถุต้นแบบในขั้นตอนการป้อนข้อมูล ถ้าเราพบว่าวัตถุใดๆในสิ่งแวดล้อมที่คล้ายกับวัตถุเป้าหมายเราสามารถถือว่ วัตถุ นั้นคือวัตถุเป้าหมายได้

ในการค้นหาวัตถุใดๆในสิ่งแวดล้อมนั้น เราจำเป็นจะต้องได้เส้นขอบของวัตถุอย่างชัดเจน วัตถุในที่นี้หมายถึง บริเวณซึ่งถูกล้อมรอบด้วยเส้นปิด โดยปกติเรามักเริ่มต้นด้วยการหาเส้นขอบของวัตถุต้นแบบ และ วัตถุในสิ่งแวดล้อม ด้วยการใช้ Mask (ดูรายละเอียดในภาคผนวก) ในการคำนวณหาค่าความแตกต่างระหว่างพิกเซล (เนื่องจากบริเวณรอยต่อของวัตถุกับฉาก หรือระหว่างวัตถุกับวัตถุ ค่าความแตกต่างระหว่างพิกเซลจะมีค่าสูง) แล้วตัดด้วยค่า threshold ค่าหนึ่งนั้น จะทำให้เกิดปัญหากรณีได้เส้นขอบที่ไม่ครบ นอกจากนี้ยังเกิดจุดดำเล็กๆ (noise) ขึ้นมามากมาย วิธีการของแคนนี่ที่นิยมใช้ในการหาเส้นบนภาพก็ได้พัฒนามาจากหลักการดังกล่าวเช่นกันแต่ได้พัฒนาให้มีความยืดหยุ่นมากยิ่งขึ้นโดยใช้ค่า threshold 2 ค่าโดยค่าหนึ่งจะทำให้ได้เส้นหลักๆในภาพนั้นและอีกค่าหนึ่งจะได้เส้นออกมามากกว่าแต่จะคงไว้เฉพาะเส้นที่ต่อกับเส้นหลักที่ได้ก่อนหน้านั้น (ดูรายละเอียดในภาคผนวก) ดังนั้นวิธีการนี้จึงมีข้อผิดพลาดเช่นกันและใช้เวลาประมวลผลนาน

#### วิธีการ Bubble

ดังนั้นทางคณะจึงได้ทำการพัฒนาวิธีใหม่ขึ้น โดยเรียกว่าวิธีการ Bubble ซึ่งมีจุดประสงค์ที่จะหาเส้นปิด (เส้นล้อมรอบพื้นที่) จากพื้นที่ซึ่งจะได้เส้นปิดที่สมบูรณ์กว่าเมื่อเทียบกับการหาเส้นจากความแตกต่างของพิกเซล กระบวนการที่เสนอมีขั้นตอนดังต่อไปนี้ เริ่มจากการมองระนาบ  $x-y$  เป็นตำแหน่งบนภาพ และแกน  $z$  เป็นค่าสี (0-255) ที่ตำแหน่ง (1) สมมติมีค่าสีเป็น  $z_1$  ที่ตำแหน่ง (2) สมมติมีค่าสีเป็น  $z_2$  โดยอาศัยหลักการว่า วัตถุเดียวกัน

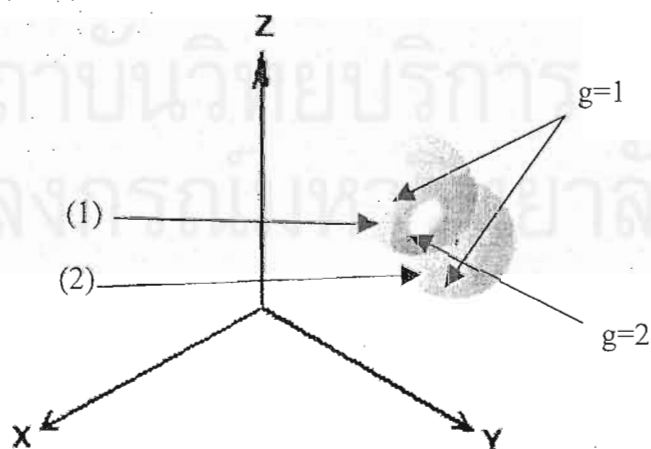


Fig 3.2-7 ภาพแสดงการคำนวณค่า  $g$  ของกระบวนการ Bubble

ยอมมีสีในบริเวณเดียวกันใกล้เคียงกันในระดับหนึ่ง ที่ตำแหน่งใดๆ เรากำหนดให้ฟังก์ชันนั้นมีชื่อว่า  $g(x,y,z)$  ค่าฟังก์ชันนี้จะเก็บค่าความถี่ของความใกล้เคียงของสี (เริ่มต้นทุกๆจุดมีค่า  $g=0$ ) กล่าวคือ เมื่อจุด (1) และ (2) ขยายออกเป็นทรงกลมรัศมีเท่าๆกัน เหตุผลที่กำหนดให้รัศมีเท่ากันเนื่องจากมีข้อสมมติฐานว่าค่าสี ตั้งแต่ 0 ถึง 255 มีความสำคัญเท่ากัน โดยที่ค่าสี 255 (ขาว) กำหนดให้เป็นสีพื้น บริเวณที่ขยายออกไปจะทำให้ค่า  $g$  บวกหนึ่ง และบริเวณที่เกิดการทับกันค่า  $g$  ก็จะบวกทับกัน ดังรูปที่ 3.2-7



Fig 3.2-8 ภาพต้นแบบในกระบวนการ Bubble.

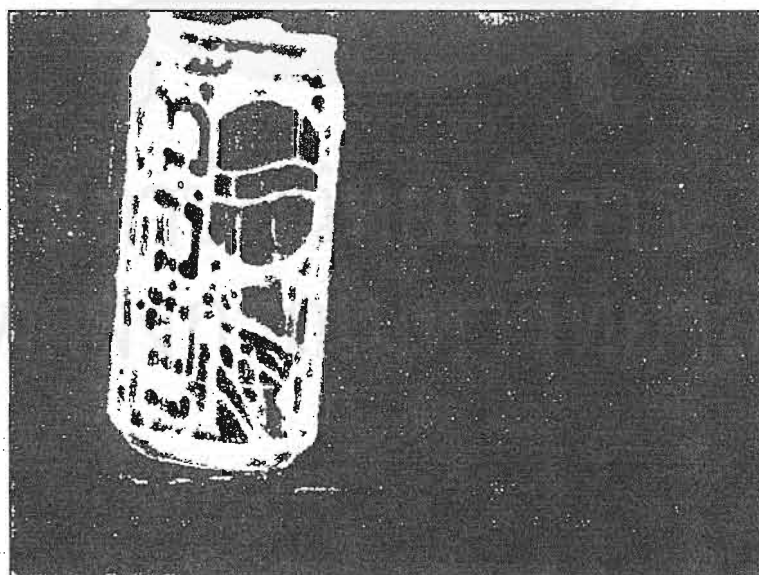


Fig 3.2-9 ภาพแสดงผลลัพธ์จากกระบวนการ Bubble

เมื่อเรากำหนดค่าฟังก์ชัน  $g$  ได้ตลอดทั้งภาพแล้ว เราก็จะทำการตัดค่า threshold โดยตำแหน่งใดมีค่า  $g$  มากกว่าเท่ากับ ค่า threshold จะแสดงเป็นสีดำ (ค่าสี=0) ตำแหน่งที่ ค่า  $g$  น้อยกว่าค่า threshold จะแสดงเป็นสีขาว (ค่าสี=255) ซึ่งการทำเช่นนี้จะทำให้เราได้บริเวณที่มีสีใกล้เคียงกันทั้งหมดเป็นวัตถุสีดำในภาพดังตัวอย่างในรูปที่ 3.2-8 กระบวนการดังกล่าวให้ผลเป็นที่น่าพอใจ ซึ่งในขั้นตอนนี้เราจะได้เส้นขอบวัตถุที่มีความต่อเนื่อง

#### ขั้นตอนในกระบวนการตรวจหาวัตถุโดยใช้ข้อมูลรูปร่างวัตถุ

เราได้กล่าวถึงหลักการของ bubble ในการหาวัตถุในภาพไปแล้ว เราจะอธิบายขั้นตอนในกระบวนการตรวจหาวัตถุโดยใช้ข้อมูลรูปร่างวัตถุ เริ่มด้วยการหาวัตถุในภาพต้นแบบและภาพที่ได้จากสิ่งแวดล้อมด้วยวิธีการ Bubble จากนั้นเราจะตัดส่วนที่เป็นวัตถุเล็กๆซึ่งอาจจะถือเป็น noise ทิ้งไปขั้นตอนต่อไปเราจะดึงวัตถุมาทีละรูป แล้วหาค่า Moment Invariants ในการนิยามลักษณะของวัตถุย่อยๆนั้นเปรียบเทียบกับของวัตถุต้นแบบ ถ้าพบวัตถุที่มีค่า Moment Invariants ใกล้เคียงกับของวัตถุต้นแบบก็ให้แยกออกมาจากภาพสิ่งแวดล้อมต่อไป



Fig 3.2-10 ภาพต้นแบบ

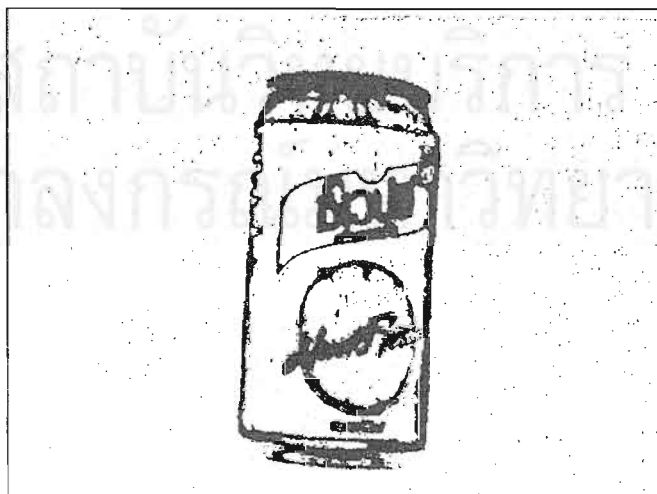


Fig 3.2-11 ภาพที่ผ่านขั้นตอนการเตรียมภาพต้นแบบ

ในการนิยามวัตถุต้นแบบ ให้นำวัตถุต้นแบบวางไว้โดยมีฉากหลังเป็นสีพื้นสีเดียวและต้องเป็นสีที่มีความเข้มต่างจากวัตถุต้นแบบพอสมควร ดังแสดงในรูป 3.2-10

จากนั้นจะทำการเบลอ เพื่อให้ภาพเรียบขึ้น นั่นคือ จะทำให้ noise ลดลงและกลมกลืนไปกับภาพ จากนั้นจึงใช้กระบวนการ bubble เพื่อแยกชิ้นส่วนย่อยๆ ของวัตถุขึ้นมา ผลของกระบวนการ bubble จะทำให้บริเวณที่มีความเข้มเดียวกันหรือใกล้เคียงกันเป็นสีดำ ซึ่งขอบของวัตถุ (เป็นบริเวณที่มีการเปลี่ยนแปลงความเข้มมาก) จะกลายเป็นสีขาว จากนั้นเราจะใช้การ invert เพื่อสลับขาวดำ ซึ่งจะได้ขอบวัตถุเป็นสีดำ และพื้นฉากเป็นสีขาว ดังแสดงในรูป 3.2-11

จากนั้นเข้ามาสู่ขั้นตอนการลด noise ซึ่งเราสามารถนำหลักการของ bubble มาใช้ในการลด noise ได้ โดยบริเวณที่เป็น noise นั้นจะเป็นจุดกระจุกกระจายอยู่เมื่อคำนวณด้วย วิธีการ Bubble แล้วจะมีค่า  $g$  ต่ำจึงใช้ค่าตัด threshold ตัดค่าเหล่านั้นแล้วคงไว้แต่บริเวณที่มีค่า  $g$  สูงก็จะทำให้ noise ลดลงได้ จากการทดลองพบว่าการลด noise โดยใช้กระบวนการ bubble กับภาพที่ผ่านขั้นตอนการเตรียมภาพ จะให้ผลดีและเร็วขึ้นมาก เพราะในขั้นแรกที่ใช้การลบวัตถุ (ถือว่า noise เป็นวัตถุหนึ่งในภาพ) ที่มีพื้นที่ เล็กกว่าค่าหนึ่งออกไปนั้น ใช้เวลาดค่อนข้างมาก กล่าวคือประมาณ 1 นาทีขึ้นไป ซึ่งขั้นตอนนี้เราใช้กระบวนการ bubble 3 ครั้ง เพราะทดลองแล้วว่าสามารถลด noise ลงได้ดี ใช้เวลาไม่นานและยังไม่ทำให้รายละเอียดของภาพเสียไปด้วย จากภาพข้างต้นจะเห็นว่า noise มีพื้นที่น้อยมาก จึงถูกตัดหายไป ดังรูป 3.2-12



Fig 3.2-12 ภาพที่ผ่านการลด noise ด้วยวิธีการ Bubble

ขั้นตอนต่อไปเป็นการนิยามวัตถุในภาพ ซึ่งเรามีสมมติฐานที่ว่าในภาพต้นแบบนั้น จะมีวัตถุเพียงชิ้นเดียว นั่นคือ วัตถุต้นแบบนั่นเอง และวัตถุที่เจอตงต้องมีขนาดของภาพใหญ่ที่สุดในภาพต้นแบบด้วย ดังนั้นวัตถุอื่นที่ไม่เกี่ยวข้อง เช่น จุดดำของฉากหลัง ตัวอักษรหรือรูปภายในตัววัตถุต้นแบบเอง และ noise ต่างๆที่ยังหลงเหลือจากการลด noise ในขั้นตอนที่ผ่านมา จะไม่ถูกสนใจและจะตัดทิ้งไปเลย ทำให้เหลือวัตถุที่มีขนาดใหญ่ที่สุดไว้เพียงชิ้นเดียว ซึ่ง

ต้องมีสมมติฐานอีกว่าวัตถุดังกล่าวที่คาดว่าเป็นวัตถุตันแบบต้องมีเส้นขอบปิดล้อมโดยรอบด้วย เพราะจะมีผลต่อไป  
ในขั้นตอนต่อไป ตัวอย่างของภาพที่ผ่านกระบวนการดังกล่าวแสดงในรูปที่ 3.2-13

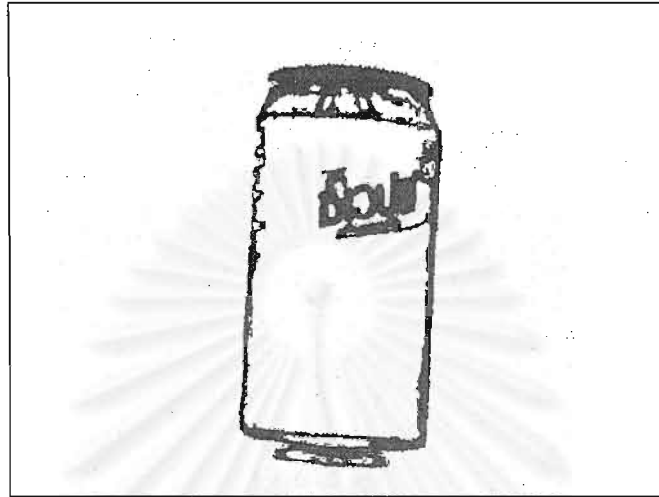


Fig 3.2-13 ภาพที่ผ่านการหาวัตถุตันแบบ

หลังจากนั้นเราจะทำให้พื้นที่ที่ปิดล้อมด้วยขอบของวัตถุตันแบบนี้กลายเป็นพื้นสีดำ อยู่บนพื้นจากสีขาว  
กรรมวิธีดังกล่าวเริ่มด้วยการทำ invert ภาพแล้วทำการแยกวัตถุออกมา ซึ่งจะทำให้ได้ภาพที่มีพื้นสีดำและตรงกลาง  
เป็นพื้นที่สีขาวของวัตถุตันแบบอยู่ จากนั้นจึงทำการ invert อีกครั้ง ที่นี้เราก็จะได้ภาพเงาของวัตถุตันแบบอยู่บนพื้นสี  
ขาว และสามารถนำมาคำนวณหาค่า moment เพื่อนิยามวัตถุตันแบบต่อไป ดังแสดงในรูปที่ 3.2-14



Fig 3.2-14 ภาพที่ผ่านการลงสีดำภายในวัตถุตันแบบ

มาถึงขั้นนี้เราจะสามารถเริ่มทำการค้นหาวัตถุเป้าหมายในสิ่งแวดล้อมได้แล้ว



Fig 3.2-15 ภาพจากสิ่งแวดล้อม

ขั้นตอนในการเตรียมภาพจากสิ่งแวดล้อมนั้น เหมือนกับการเตรียมภาพจากภาพต้นแบบทุกประการ คือ มี การทำเบลอภาพ การทำ bubble และการทำ invert ต่อมาจึงเป็นการลด noise ซึ่งเป็นการใช้กระบวนการ bubble 3 ครั้งติดต่อกัน ผลเป็นดังภาพในหน้าถัดไป

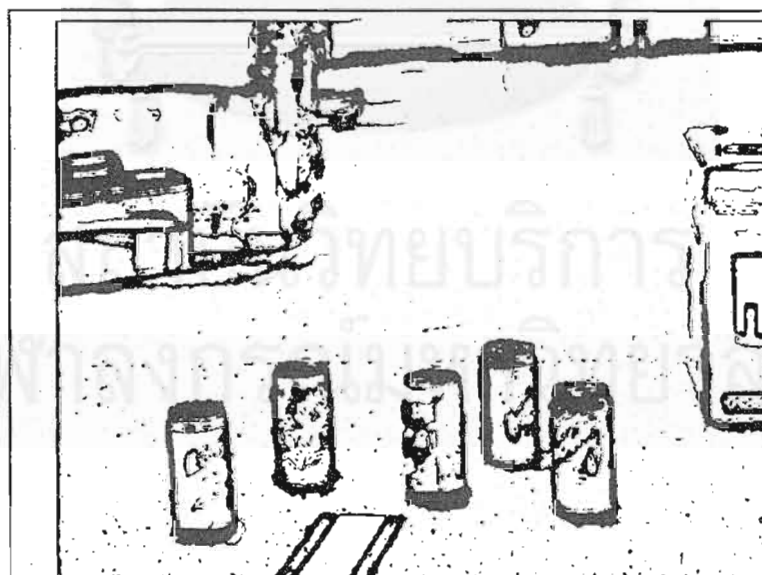


Fig 3.2-16 ภาพจากสิ่งแวดล้อม ผ่านการกระบวนการเริ่มต้น

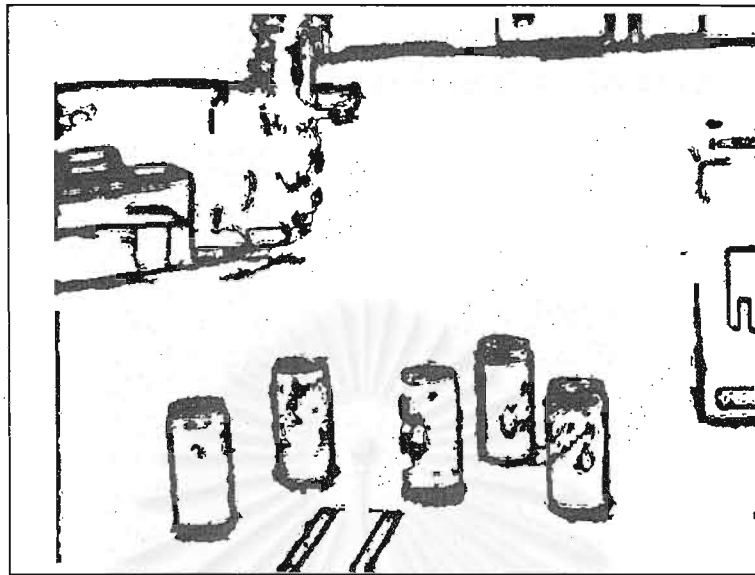


Fig 3.2-17 ภาพหลังจากทำการลด noise

ต่อมาเป็นขั้นตอนการระบายสีดำลงในวัตถุในภาพ ซึ่งจะเหมือนกับการทำภาพต้นแบบ เพียงแต่จะเพิ่มส่วนอื่นๆ เข้ามา คือ การกำจัด noise อีกครั้ง ซึ่งในการทำภาพต้นแบบนั้นเรามีสมมติฐานที่ว่าในภาพต้นแบบมีวัตถุเพียงชิ้นเดียว แต่ภาพจากสิ่งแวดล้อมอาจจะมีวัตถุมากกว่าหนึ่งชิ้นก็เป็นได้ ฉะนั้นเราจึงต้องกำจัดบริเวณที่คาดว่ามีวัตถุ

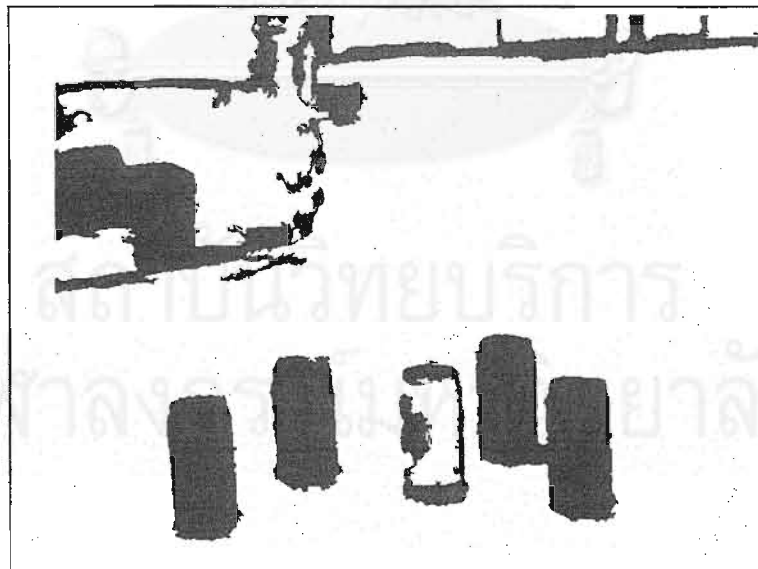


Fig 3.2-18 ภาพจากการทำให้เป็นสีดำ และ ตัดบริเวณที่ไม่เกี่ยวข้องออก

ซึ่งออกไป บริเวณดังกล่าวคือ มุมทั้งสี่ของภาพ ซึ่งเรามีสมมติฐานที่ว่าภาพวัตถุที่เราต้องการต้องอยู่บริเวณกลางภาพเท่านั้น ฉะนั้นบริเวณที่อยู่ขอบหรือมุมของภาพเราจะตัดทิ้ง นอกจากนั้น เราจะกำจัด noise ออกด้วย ซึ่งเราถือว่าภาพวัตถุที่มีขนาดเล็กขนาดหนึ่ง ไม่ถือเป็นวัตถุ หรือให้ถือเป็น noise ซึ่งในขั้นตอนนี้จะใช้เวลาค่อนข้างมากเมื่อเทียบกับขั้นตอนอื่น เพราะเราต้องพิจารณาทุกๆ ภาพวัตถุ ซึ่งถ้าภาพไหนมี noise เหลือจากกระบวนการข้างต้นมามากยิ่งจะทำให้โปรแกรมประมวลผลช้าลงไปอีก จากกระบวนการดังกล่าวเราจะได้ภาพดังนี้ ซึ่งจะเห็นว่าภาพไม่เปลี่ยนแปลงเลย นอกจากการระบายสีลงในวัตถุและลบ noise เท่านั้น เพราะที่มุมของภาพไม่มีวัตถุอยู่เลย

จากภาพ จะเห็นว่าจะมีกระป๋อง 4 กระป๋อง (ซึ่งเป็นวัตถุต้นแบบด้วย) ที่ถูกระบายสีดำ และอีกกระป๋องที่ไม่ถูกระบายสี เพราะเนื่องจากไม่ถูกปิดล้อมรอบด้วยเส้นขอบหรือเส้นขอบไม่ครบรอบ ซึ่งจะขัดกับสมมติฐานที่ตั้งไว้ว่าวัตถุต้องมีเส้นขอบที่ครบรอบ และเรายังพบว่ามีการปอง 2 กระป๋องที่เชื่อมกันอยู่ ซึ่งตรงเส้นที่เชื่อมกันพบว่าเป็นเงาของกระป๋องใบซ้าย

จากนั้นก็เข้ามาสู่ขั้นตอนการตรวจสอบว่าวัตถุที่เหลือเป็นวัตถุเดียวกับวัตถุต้นแบบหรือไม่ โดยอาศัยการคำนวณการค่า moment นำมาเปรียบเทียบและต้องไม่เกินค่าความแตกต่างที่กำหนดไว้ตอนแรก ถ้าพบว่าในภาพมีวัตถุต้นแบบอยู่มากกว่า 1 ชิ้น เราจะถือให้วัตถุชิ้นสุดท้ายที่พบเป็นวัตถุที่เราจะเก็บ และผลการวิเคราะห์เป็นดังรูปที่ 3.2-19



Fig 3.2-19 ภาพผลลัพธ์สุดท้ายจากการเปรียบเทียบค่าโมเมนต์

จากภาพข้างต้น จะเห็นว่า เราพบสามารถค้นหาวัตถุต้นแบบได้แล้ว แต่จาก รูปที่ 3.2-19 เราพบภาพวัตถุต้นแบบถึง 3 ชิ้น ซึ่งในความเป็นจริงควรจะมีแค่ 2 (เพราะกระป๋อง 2 ใบทางขวาสุดเชื่อมติดกันเป็นชิ้นเดียวซึ่งเป็นผลจากการทำ bubble) เหตุที่พบวัตถุต้นแบบ 3 ชิ้น เพราะค่า moment ของวัตถุทั้งสามใกล้เคียงกับค่า moment ของวัตถุต้นแบบ ซึ่งในความเป็นจริงค่า moment ของกระป๋อง 2 ใบที่ติดกันไม่น่าจะใกล้เคียงกับของต้นแบบได้ แต่จาก



กรณีนี้ทำให้พิสูจน์ได้ว่า ข้อจำกัดอีกอย่างหนึ่งของการใช้ค่า moment ในการนิยามวัตถุต้นแบบ คือ ค่า moment ของวัตถุชิ้นอื่นอาจมาตรงหรือใกล้เคียงกับของวัตถุต้นแบบได้ ซึ่งทำให้การค้นหาวัตถุต้นแบบไขว้เขวได้

อย่างไรก็ดี จากผลการทดลองใช้โปรแกรมดังกล่าว ทำให้เราทราบถึงข้อดีและข้อเสียบางอย่างของวิธีการนี้ ข้อดีคือค่อนข้างแม่นยำในการบอกตำแหน่งของวัตถุ และแม่นยำในการหาวัตถุต้นแบบจากภาพต้นแบบ (ได้วัตถุต้นแบบจากภาพต้นแบบออกมาชัดเจนมากและใกล้เคียงมาก) แต่ข้อเสียคือ ประมวลผลช้ามาก (เพราะผ่านกระบวนการมากมาย) และ วัตถุภายในสิ่งแวดล้อมต้องไม่ใกล้กันมากหรือติดกัน นั่นหมายความว่าไม่สามารถค้นหาวัตถุต้นแบบออกจากกองวัตถุได้ อีกทั้งยังเสี่ยงในกรณีที่วัตถุอื่นที่ไม่ใช่วัตถุต้นแบบมีค่า moment ใกล้เคียงกับของวัตถุต้นแบบ หรือ มีรูปทรงใกล้เคียงกัน และเมื่อเปรียบเทียบแล้วค่าความแตกต่างน้อยกว่าที่กำหนด ก็จะทำให้ผลลัพธ์ผิดพลาดได้



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

### 3.3 การพัฒนาโปรแกรม

การพัฒนาโปรแกรมในการค้นหาวัตถุเป้าหมายในสิ่งแวดล้อมด้วยกล้อง CCD เป็นเป้าหมายหลักของโครงการนี้ โดยโปรแกรมนี้จะทำการติดต่อบริษัทข้อมูลภาพจากกล้อง CCD ที่ใช้งาน เข้ามาประมวลผลเพื่อหาวัตถุเป้าหมายโดยตรง และ แสดงผลตำแหน่งวัตถุที่ค้นหาพบด้วยเครื่องหมายกากบาท พร้อมทั้ง ทำการตีกรอบบริเวณที่คาดว่าคือวัตถุเป้าหมายอยู่โดยใช้กระบวนการค้นหาวัตถุด้วยกลุ่มสีและ กระบวนการค้นหาวัตถุด้วยรูปร่างวัตถุ ตามที่ได้อธิบายในหัวข้อที่ผ่านมาแล้ว นอกจากนี้ทางคณะยังได้พัฒนาโปรแกรมในการประมวลผลพื้นฐานที่ใช้ทดสอบในโครงการซึ่งจะได้กล่าวถึงในส่วนท้ายของบท

#### 3.3.1 การพัฒนาโปรแกรมค้นหาวัตถุด้วยกล้อง CCD

กล้อง CCD ที่ใช้ในโครงการนี้ เป็นกล้อง CCD จากบริษัท Camstar รุ่น CIC 741 แบบปรับชัดเตอร์หน้า กล้องด้วยมือ ซึ่งมีความละเอียด 640 x 480 พิกเซล และให้ภาพขาว-ดำ ขนาด 8 บิต หรือ 256 ระดับความเข้มสี ในแต่ละพิกเซล

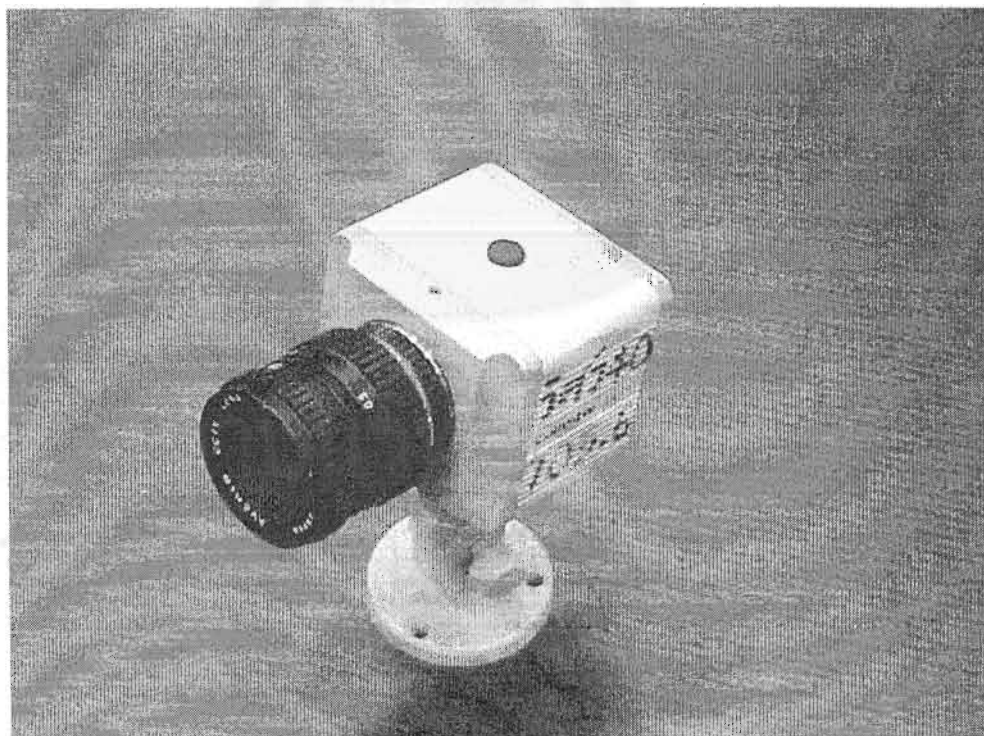
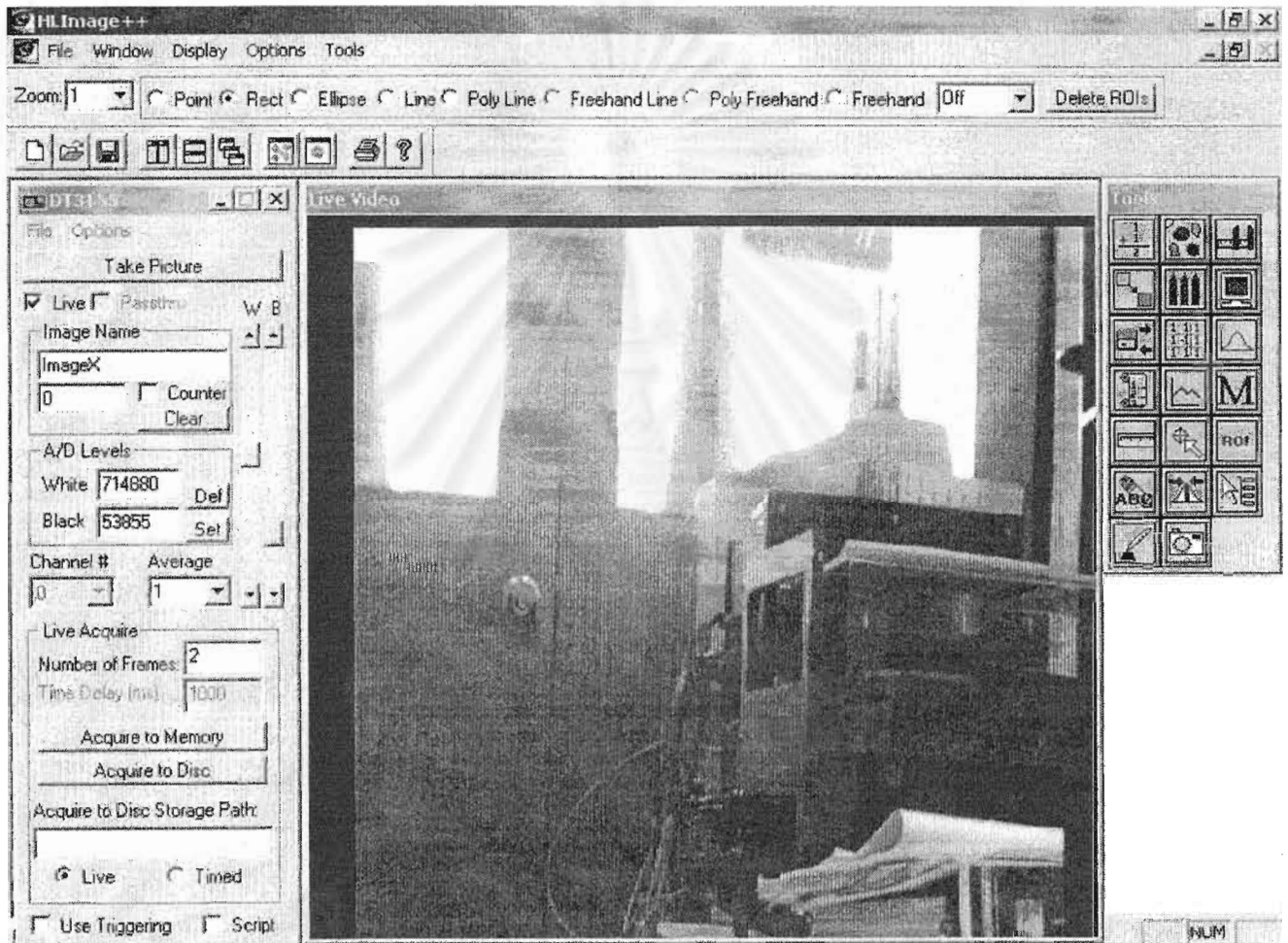


Fig 3.3-1 กล้อง CCD

ในการติดต่อกับกล้องนั้น เราสามารถใช้ โปรแกรม HL-Image จากบริษัท Western Vision Software ได้ ซึ่งภายในโปรแกรม จะบรรจุกระบวนการประมวลผลภาพจำนวนหนึ่งไว้ด้วย และสามารถสั่งถ่ายภาพจากกล้อง และบันทึกภาพลงไฟล์ได้



จุฬาลงกรณ์มหาวิทยาลัย

Fig 3.3-2 โปรแกรม HL-Image

### การเขียนโปรแกรมเพื่อติดต่อกับกล้อง CCD โดยตรง ผ่านไลบรารีของ HL - Image

การติดต่อกับกล้องโดยใช้ โปรแกรม HL-Image ดังกล่าวในหัวข้อที่ผ่านมา นั้น เราไม่สามารถใช้งานกล้องได้อย่างอิสระ เนื่องจากต้องบันทึกภาพลงไฟล์ด้วยโปรแกรม HL - Image เสียก่อน จึงจะสามารถทำการประมวลผลจากไฟล์ภาพได้ วิธีการที่ดีกว่าคือการสร้างโปรแกรมติดต่อกับกล้อง CCD โดยตรง โดยนำเอาฟังก์ชันพื้นฐาน และ ไฟล์ Preprocessor, Dynamic Link Library ภายในโปรแกรม HL - Image ซึ่งรองรับภาษาซี มารวมในโปรแกรมที่สร้างขึ้น ด้วยวิธีนี้เราสามารถนำข้อมูลจากกล้องเข้ามายังหน่วยความจำของคอมพิวเตอร์ได้โดยตรง โดยไม่ต้องผ่านการบันทึกไฟล์ลงฮาร์ดดิสก์

ในการนำข้อมูลจากกล้อง มาประมวลผลมีสิ่งหนึ่งที่เราต้องระวัง คือ ข้อมูลภาพบีตแมป ขนาด 640 x 480 พิกเซล ที่ได้จากกล้อง CCD มีลักษณะขอบทางด้านซ้าย และ ด้านบนของรูปเป็นสีดำ ซึ่งเป็นข้อมูลของรูปภาพที่ไม่ถูกต้องดังแสดงในรูป

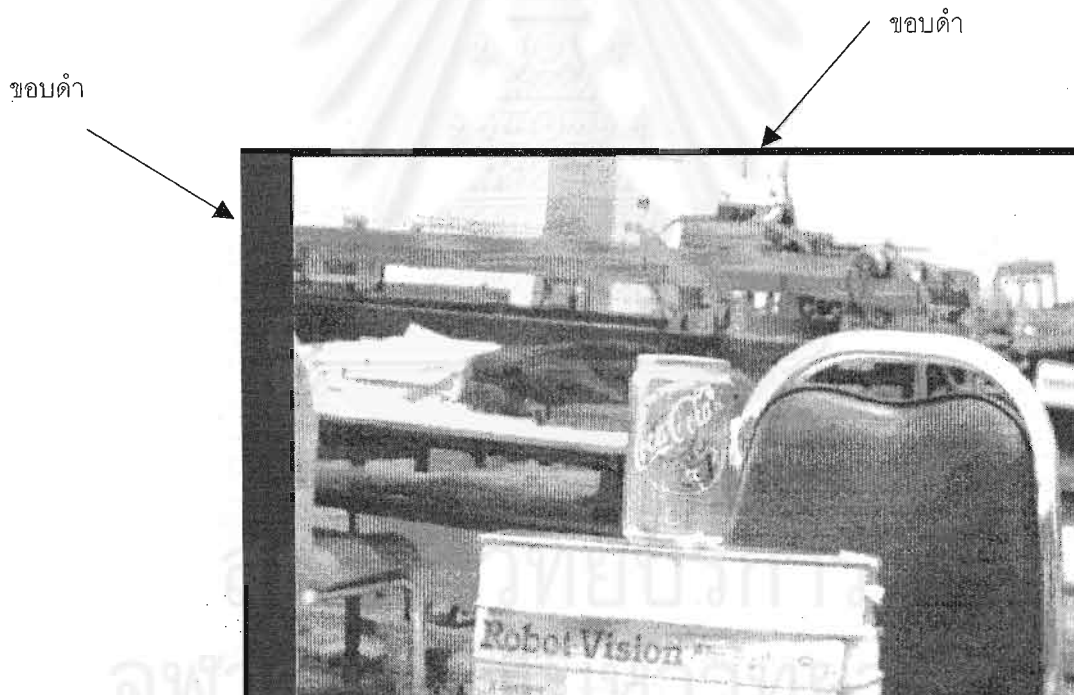


Fig 3.3-3 ภาพแสดงบริเวณข้อมูลที่ผิดพลาดที่ได้จากกล้อง CCD

ดังนั้น เพื่อความถูกต้องของข้อมูล ในการนำข้อมูลจากกล้องไปประมวลผล จำเป็นจะต้องละทิ้งบริเวณดังกล่าวออกไป

## การทดสอบโปรแกรม

ลักษณะของโปรแกรม เป็นดังนี้

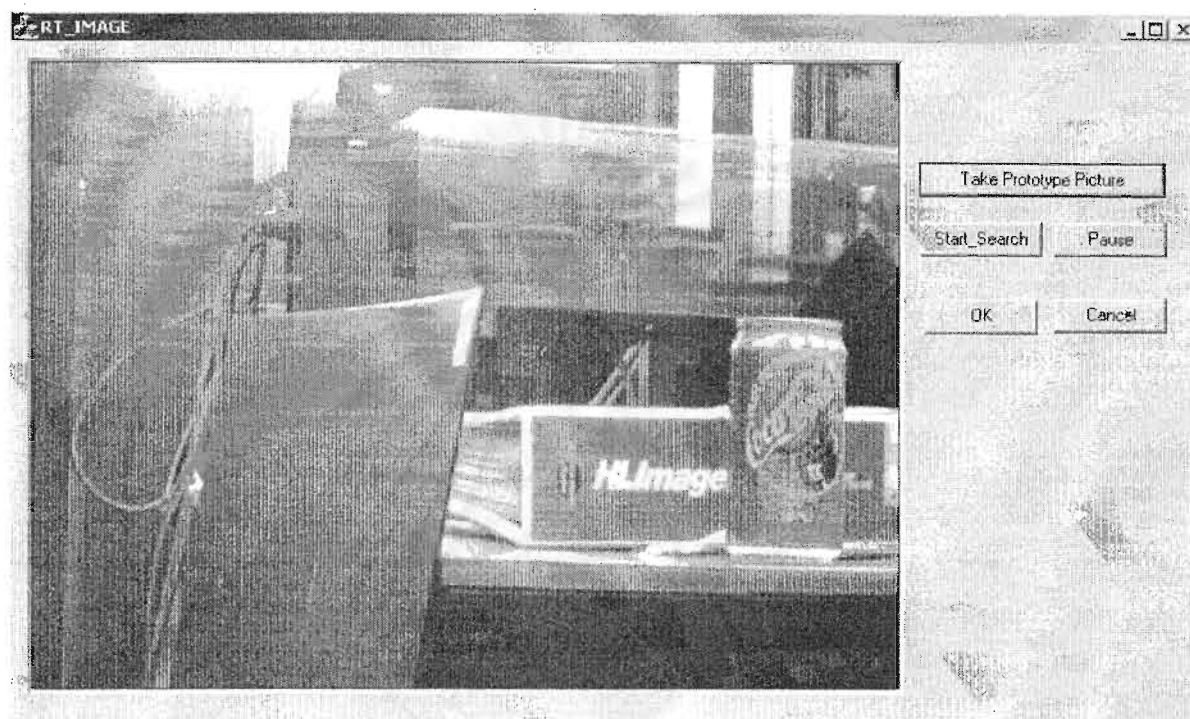


Fig 3.2-4 ภาพแสดงโปรแกรม ค้นหาวัตถุ ซึ่งติดต่อกับกล้อง CCD โดยตรง

การเริ่มต้นโปรแกรมทำได้โดยการนำวัตถุเป้าหมายที่ต้องการทำการค้นหาวางบนฉากที่มีสีค่อนข้างแตกต่างจากวัตถุ แล้ว คลิกเมาส์ที่ปุ่ม Take prototype picture ภาพต้นแบบที่ถ่ายได้จะปรากฏบนจอคอมพิวเตอร์ หากได้ภาพที่ไม่สมบูรณ์ สามารถกดปุ่ม Take prototype picture ใหม่จนกว่าจะได้ภาพที่สมบูรณ์ได้ เมื่อได้ภาพที่สมบูรณ์แล้ว โปรแกรมก็จะทำการค้นหาวัตถุในสิ่งแวดล้อมแล้ว จากนั้นให้คลิกเมาส์ที่ปุ่ม Start โปรแกรมก็จะเริ่มทำการค้นหาวัตถุเป้าหมายจากภาพสิ่งแวดล้อมใดๆ ที่กล้องกำลังจับภาพอยู่ในขณะนั้น แล้วแสดงผลตำแหน่งวัตถุ ด้วยเครื่องหมายกากบาท พร้อมแสดงบริเวณวัตถุเป้าหมายด้วยกรอบสี่เหลี่ยม บนจอคอมพิวเตอร์



Fig 3.3-5 ภาพแสดงการเก็บภาพวัตถุต้นแบบ

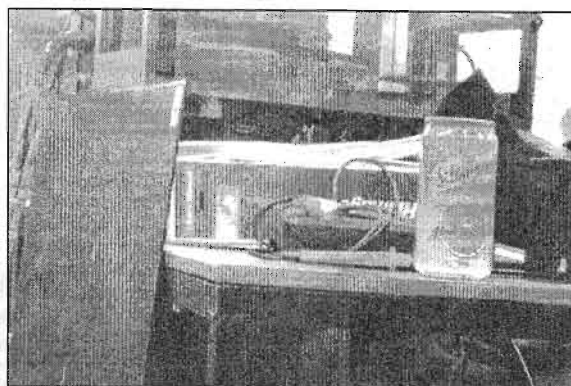


Fig 3.3-6 ภาพวัตถุในสิ่งแวดล้อมที่ต้องการค้นหา



Fig 3.3-7 ภาพแสดงบริเวณวัตถุที่โปรแกรมค้นหาพบ

จากการทดลองใช้โปรแกรมดังกล่าวพบว่า สามารถใช้ค้นหาวัตถุได้ค่อนข้างแม่นยำ แต่ใช้เวลาในการประมวลผลค่อนข้างมาก และ เนื่องจากใช้กระบวนการวนลูปเงื่อนไข ทำให้ ในภาพที่ต่างกันจะใช้ระยะเวลาประมวลผลไม่เท่ากัน ทำให้ระยะเวลาเก็บภาพแต่ละเฟรม (Sampling time) ไม่เท่ากัน

นอกจากนี้ยังพบข้อจำกัดบางอย่างคือ การที่กล้อง CCD ที่ใช้เป็นแบบปรับชัดเตอร์หน้ากล้องด้วยมือ ทำให้ จำเป็นต้องทำการรักษาตำแหน่งชัดเตอร์ขณะทำการค้นหาภาพในสิ่งแวดล้อม ให้เป็นตำแหน่งเดียวกับเมื่อทำการ

ถ่ายภาพต้นแบบ มีฉะนั้นข้อมูลความเข้มแสงที่เก็บจากภาพต้นแบบจะผิดทันที รวมทั้ง กล้อง CCD ที่ใช้ยังมีปัญหาตรงที่มันมีความไวต่อปริมาณแสงที่เข้าอย่างมาก โดยกล้องจะปรับระดับความเข้มสีของทั้งภาพโดยอัตโนมัติ ทำให้วัตถุเดิม มีระดับความเข้มแสงเปลี่ยนไปได้อย่างมาก เมื่อถ่ายในสิ่งแวดล้อมที่มีความเข้มแตกต่างกันเพียงเล็กน้อยซึ่งก่อให้เกิดปัญหาได้เมื่อสภาวะแสงของสิ่งแวดล้อมเปลี่ยนแปลงไป



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



### 3.3.2 โปรแกรมประมวลผลภาพพื้นฐาน

เนื่องจากในการศึกษาวิธีการประมวลผลภาพ จำเป็นจะต้องทำการทดลองใช้กระบวนการประมวลผลภาพแบบต่างๆ เป็นลำดับต่อเนื่องกัน โดยแต่ละกระบวนการจะมีพารามิเตอร์ที่เราต้องทำการลองสุ่มค่าไปเรื่อยๆ (Trial & Error) เพื่อหาค่าที่ดีที่สุด เช่น Threshold value, Standard Deviation เป็นต้น ดังนั้น จึงจำเป็นต้องมีโปรแกรมที่รวบรวมกระบวนการประมวลผลภาพแบบต่างๆ เข้าด้วยกัน ซึ่งสามารถกำหนดเลือกภาพที่จะประมวลผลด้วยกระบวนการแบบต่างๆได้ และสามารถปรับเปลี่ยนพารามิเตอร์ของกระบวนการต่างๆ ได้สะดวก ด้วยเหตุผลดังกล่าว จึงได้ทำการสร้างโปรแกรม Image Processing Toolbox ขึ้น

รายละเอียด และ รูปแบบของโปรแกรม Image Processing Toolbox มีดังนี้

#### A. รูปแบบไฟล์ภาพบิตแมปที่ใช้ในการประมวลผลในโปรแกรม

เนื่องจากข้อมูลภาพที่จะได้จากกล้อง CCD ที่ใช้ เป็นภาพขาว - ดำระดับความเข้มแสงขนาด 8 บิต หรือ 256 สี ดังนั้นโปรแกรมสำหรับศึกษาวิธีการประมวลผลภาพนี้จะใช้สำหรับประมวลผลภาพรูปแบบดังกล่าวเท่านั้น โดยถ้าไฟล์ภาพที่จะนำมาประมวลผล ถูกสร้างขึ้นเองด้วยโปรแกรมสำหรับวาดรูปภาพใดๆ เช่น ไมโครซอฟท์ เพนท็บริช จะต้องทำการบันทึกไฟล์เป็นรูปแบบ 256 Color Bitmap ภาพจึงจะถูกประมวลผลอย่างถูกต้อง

ไฟล์ภาพชนิดบิตแมปนี้จะเก็บข้อมูลเป็นสองส่วน ส่วนแรกเรียกว่า ส่วนหัวของไฟล์ ซึ่งเก็บข้อมูลรูปแบบของภาพไว้ ส่วนที่สองเป็นข้อมูลของภาพในแต่ละพิกเซล ดังนั้นในการนำข้อมูลรูปภาพ เข้ามาประมวลผลในโปรแกรมจะต้องทำการบันทึกส่วนหัวของไฟล์เก็บไว้ก่อน จากนั้นจึงทำการดึงข้อมูลของพิกเซลมาเก็บไว้ในตัวแปรอาร์เรย์ชนิด Unsigned char ซึ่งเป็นข้อมูลที่มีค่าได้ตั้งแต่ 0 - 255 หรือ 8 บิต เพื่อใช้ในการประมวลผล เมื่อทำการประมวลผลภาพเสร็จเรียบร้อยแล้ว จึงทำการบันทึกส่วนหัวของไฟล์ที่เก็บไว้ในไฟล์ผลลัพธ์ ตามด้วยข้อมูลของพิกเซลที่ทำการประมวลผลเรียบร้อยแล้ว จึงจะได้ไฟล์ผลลัพธ์ ในรูปแบบเดียวกับไฟล์รูปภาพก่อนทำการประมวลผล

#### B. การสร้างแอปพลิเคชันแบบไดอะล็อก และ คอนโทรลแบบต่างๆ

โปรแกรมที่ทำการพัฒนาขึ้นทั้งหมด สร้างขึ้นเป็นแอปพลิเคชันแบบไดอะล็อก ด้วยโปรแกรม วิซวล C++ ซึ่งทำให้สามารถใช้ คลาสพื้นฐานของไมโครซอฟท์ (MFC) ได้ เนื่องจากโปรแกรมไมโครซอฟท์วินโดวส์เองก็ถูกสร้างขึ้นด้วยภาษาซี และ MFC เช่นกัน ทำให้ข้อมูลทุกชนิดที่ใช้ในโปรแกรมวิซวล C++ เป็นข้อมูลที่วินโดวส์รู้จักทั้งสิ้น ดังนั้นโปรแกรมที่ได้สร้างขึ้นจึงสามารถนำไปใช้ได้ไม่ว่าในคอมพิวเตอร์ได้ทุกเครื่องที่ใช้ระบบปฏิบัติการวินโดวส์ตั้งแต่เวอร์ชัน 95 ขึ้นไป

การสร้างแอปพลิเคชันแบบไดอะล็อก มีจุดเด่นอยู่ที่การใช้คอนโทรล สำหรับติดต่อสื่อสาร กับ ผู้ใช้ ซึ่งในโปรแกรมที่สร้างขึ้นนี้ใช้คอนโทรลพื้นฐาน 5 อย่างดังต่อไปนี้



1. ปุ่มกด (Push Button) เป็นคอนโทรลที่ใช้สะดวกมากในการติดต่อกับผู้ใช้ ในการสั่งงานต่างๆ
2. กล่องข้อความ (Text box) เป็นคอนโทรลสำหรับรับค่าซึ่งอาจเป็นตัวเลข หรือ ข้อความสตริงก็ได้
3. กล่องคอมโบ (Combo box) เป็นคอนโทรลสำหรับติดต่อกับผู้ใช้ในการเลือก ทางเลือกต่างๆ
4. ปุ่มเรดิโอ (Radio button) เป็นคอนโทรลสำหรับติดต่อกับผู้ใช้ในการเลือก ทางเลือกต่างๆ เช่นกัน
5. สเตติก (Static) เป็นคอนโทรลสำหรับแสดงผล ทั้งในรูปแบบข้อความ และ รูปภาพ

เมื่อต้องการรับส่งข้อมูลจากคอนโทรลใด จะต้องทำการสร้างตัวแปรสมาชิก (member variable) ให้แก่คอนโทรลนั้นก่อนโดยใช้ คลาสวิซาร์ด

เนื่องจากในโปรแกรมวิซวล C++ ไม่มีคลาสสำหรับนำภาพจากไฟล์บิตแมปมาแสดงผล จึงต้องสร้างคลาสควบคุมบิตแมปขึ้น (BitmapControl Class) ซึ่งคลาสนี้จะทำให้สามารถนำข้อมูลภาพจากไฟล์บิตแมป ไปแสดงผลบนคอนโทรลชนิดสเตติกได้ ซึ่ง Source Code ของคลาสนี้ และ วิธีการใช้งาน สามารถดาวน์โหลดได้จากเว็บไซต์ [www.codeguru.com](http://www.codeguru.com)

### C. ส่วนโปรแกรมหลัก และ ส่วนแสดงผล

โปรแกรม Image Processing Toolbox ประกอบไปด้วย 8 ส่วนหลัก คือ

1. ส่วนแสดงภาพ ก่อนผ่านกระบวนการประมวลผล
2. ส่วนแสดงภาพ หลังผ่านกระบวนการประมวลผล
3. ส่วนเลือกกระบวนการประมวลผลภาพ เป็น Combo box สำหรับเลือกกระบวนการ
4. ส่วนกำหนดชนิดของภาพ สำหรับระบุว่ากระบวนการประมวลผลภาพ เป็นของภาพชนิด Gray-level หรือ ชนิด Binary
5. ส่วนกำหนดขนาดของภาพ สำหรับระบุว่าภาพที่ใช้ในการประมวลผลนั้น มีขนาดเป็น Pixel เท่าไร โดยจะต้องระบุให้ถูกต้องมิฉะนั้นการประมวลผลจะผิดพลาดได้
6. ส่วนเลือกภาพก่อนการประมวลผล สำหรับรับข้อมูลจากไฟล์ภาพ เข้ามาในโปรแกรม ซึ่งอาจพิมพ์ชื่อไฟล์ ลงใน Text box หรือ อาจเลือกไฟล์โดยใช้ปุ่ม Browse ก็ได้
7. ส่วนกำหนดชื่อไฟล์ภาพหลังการประมวลผล ซึ่งอาจพิมพ์ชื่อไฟล์ ลงใน Text box หรือ อาจเลือกไฟล์โดยใช้ปุ่ม Browse ก็ได้
8. ปุ่มรันโปรแกรม สำหรับสั่งให้โปรแกรมเริ่มทำการประมวลผล

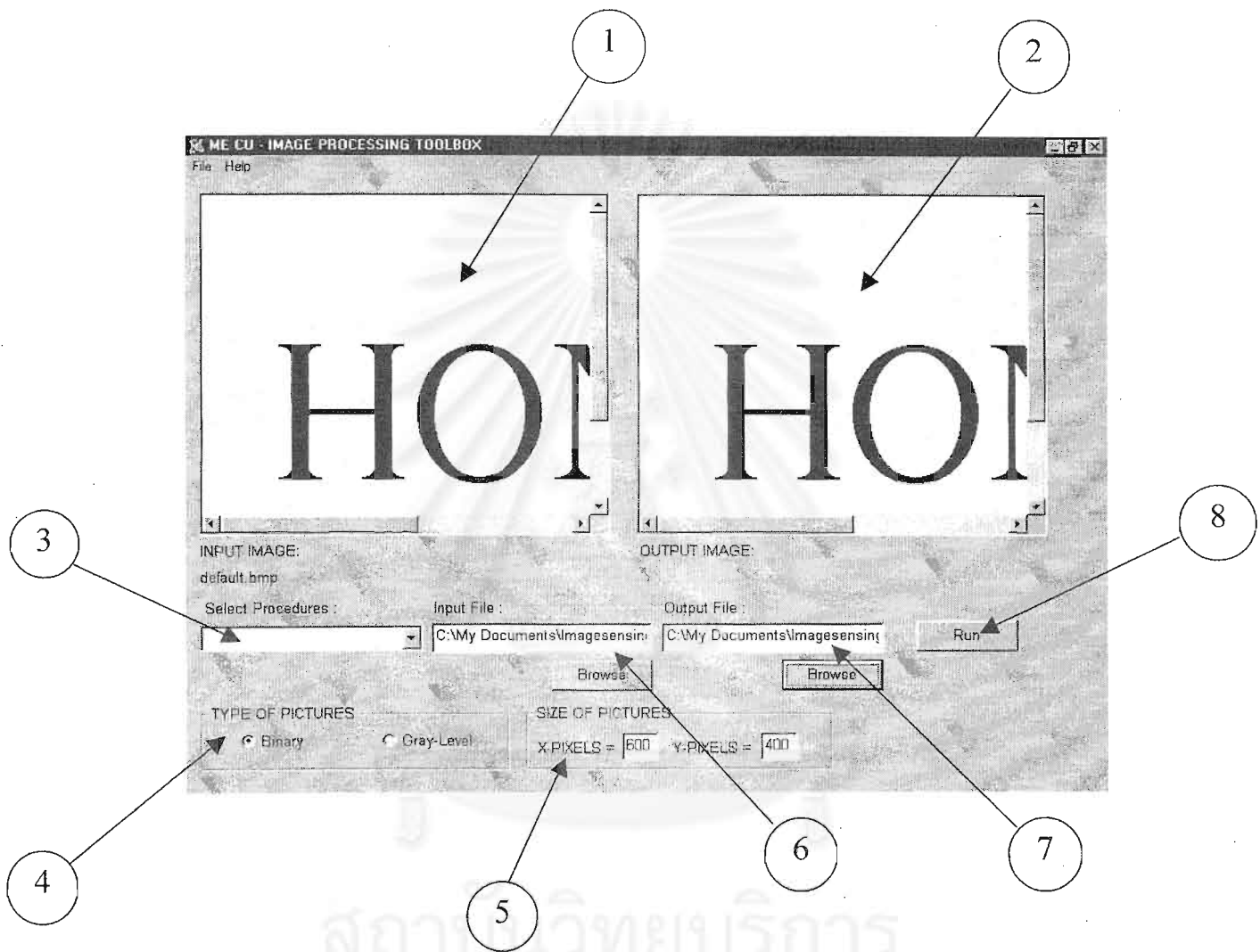


Fig 3.3-8 ภาพแสดงส่วนประกอบหลักของโปรแกรม Image Processing Toolbox

#### D. กระบวนการค้นหาขอบของวัตถุในภาพ

1. Edge Detection เป็นกระบวนการในการหาขอบของภาพ Gray - Level อย่างง่าย โดยอาศัยการคำนวณค่าด้วย Mask แบบอยู่นิ่ง (Static) แล้วทำการ Threshold โดยภายในโปรแกรมได้บรรจุ Mask มาตรฐานแบบต่างๆ ดังที่ได้อธิบายในบทที่ 2 ไว้ด้วย ได้แก่ Smoothed, Sobel, Isotropic, Discrete Laplace และ Stochastic นอกจากนี้ยังสามารถทดลองออกแบบ Mask ขนาดใหญ่ได้ถึง 7x7 รวมทั้งสามารถกำหนดค่า Threshold ได้

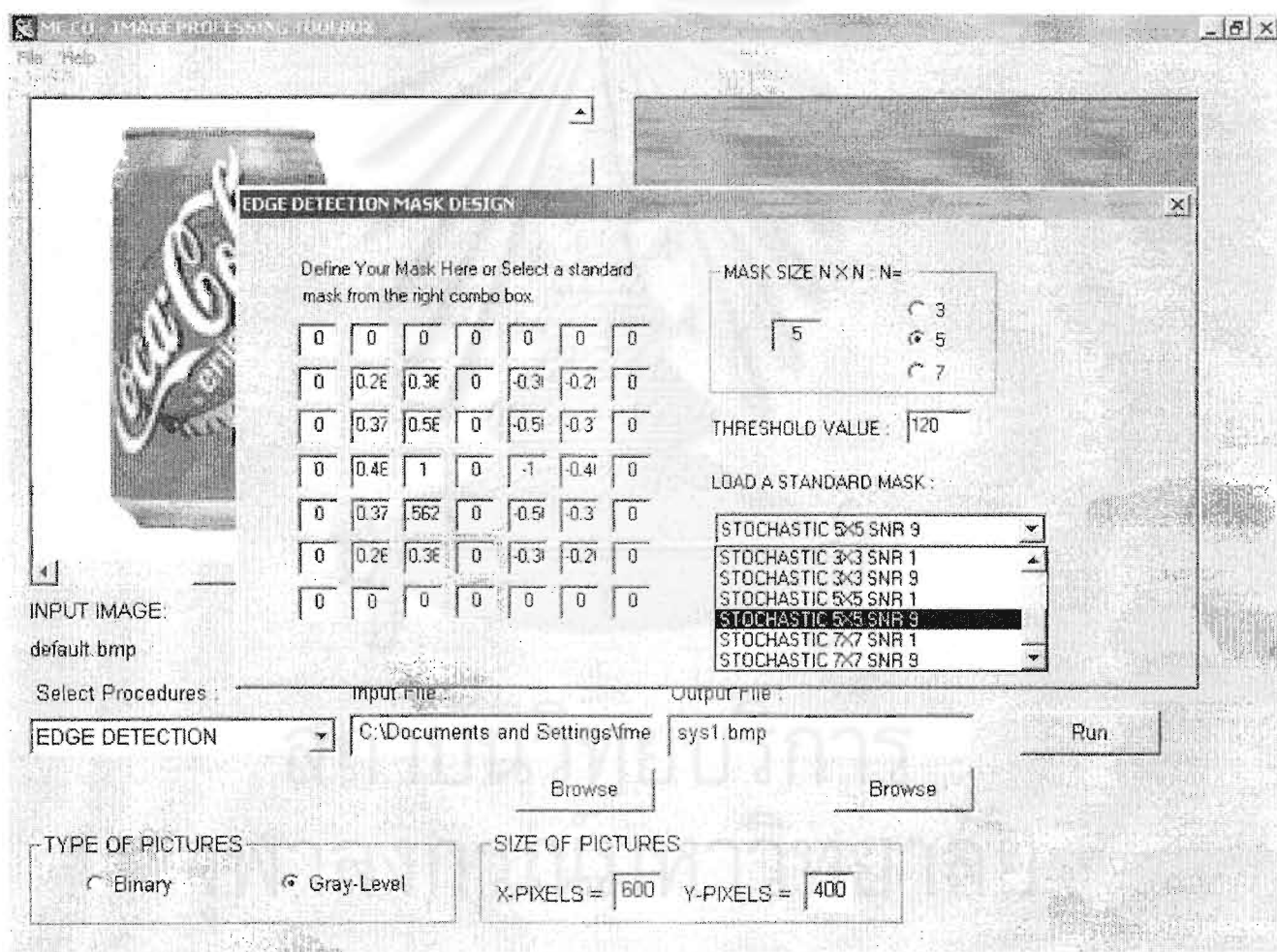
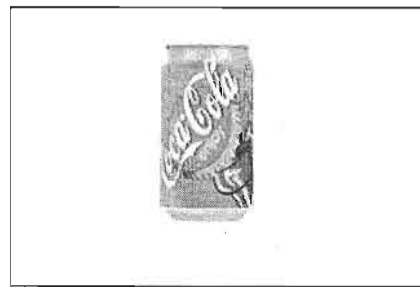
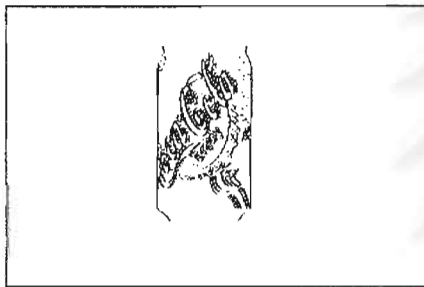


Fig 3.3-9 ภาพแสดงส่วนโปรแกรม EDGE DETECTION



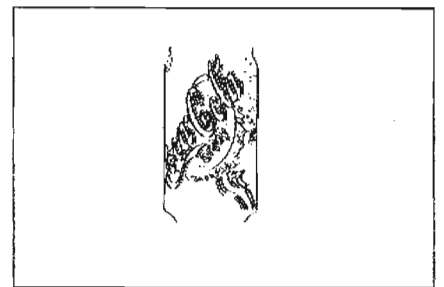
ภาพต้นแบบ



Smoothed



Sobel



Isotropic



Discrete Laplace



Stochastic

Fig 3.3-10 ภาพแสดงผลลัพธ์ที่ได้จากโปรแกรม EDGE DETECTION

จะสังเกตเห็นได้ว่าภาพขอบที่ได้จาก Mask แบบ อยู่นิ่ง (Static) อาจทำให้ได้ขอบเพียงด้านเดียว โดยภาพขอบของกระป๋องด้านล่าง และ ด้านบนจะหายไป ในบางภาพ ซึ่งสามารถแก้ได้ด้วยการใช้ Mask แบบหมุน (Rotating/ Compass Operator) ดังจะกล่าวถึงต่อไป

2. Compass เป็นกระบวนการหาขอบโดย Compass Operator ซึ่งเป็น Mask ขนาด 3x3 หมุนตามเข็มนาฬิกาครั้งละ 45 องศา ในโปรแกรมสามารถทดลองออกแบบ Mask และ ปรับเปลี่ยนค่า Threshold ได้

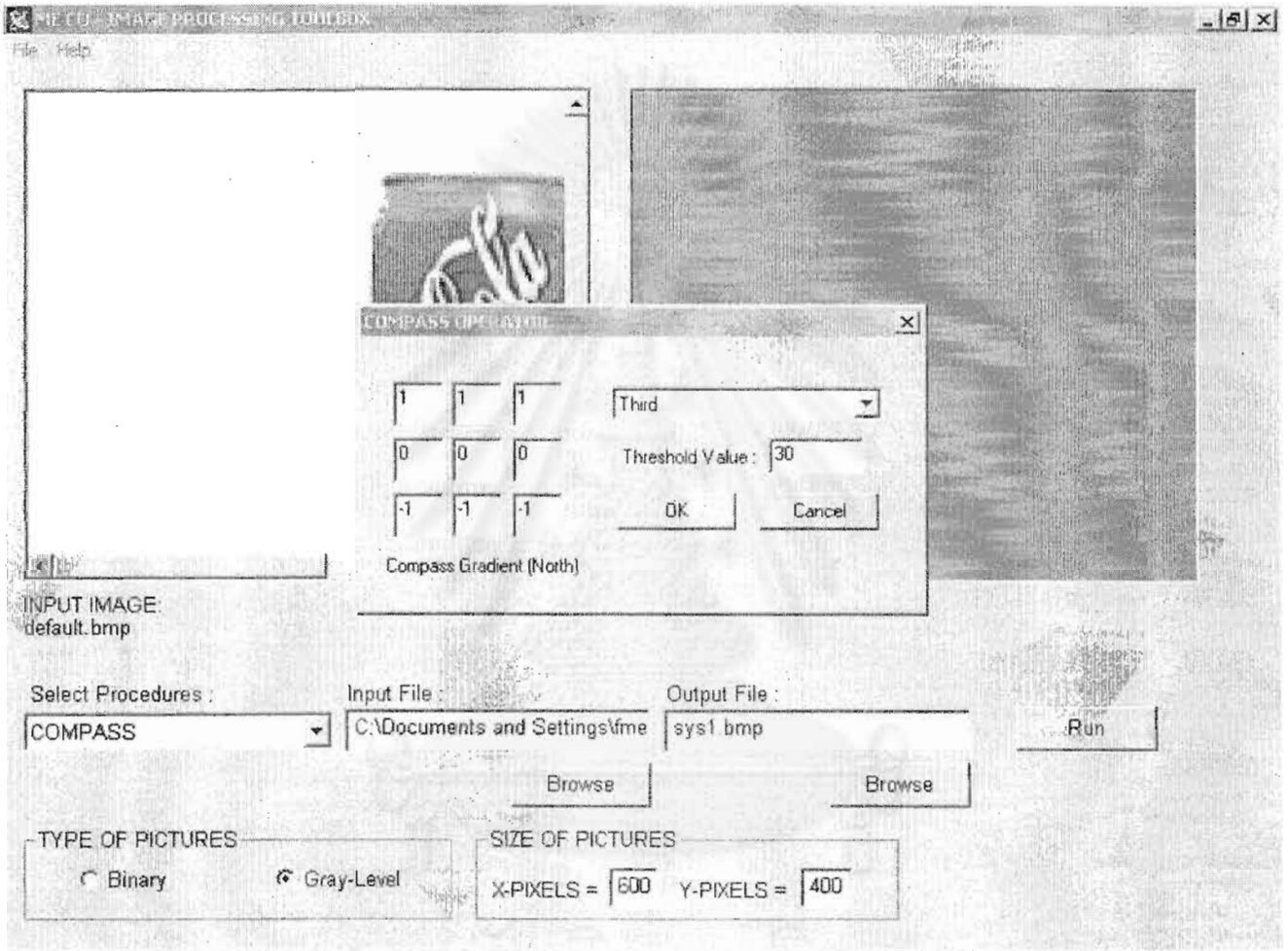


Fig 3.3-11 ภาพแสดงโปรแกรม COMPASS

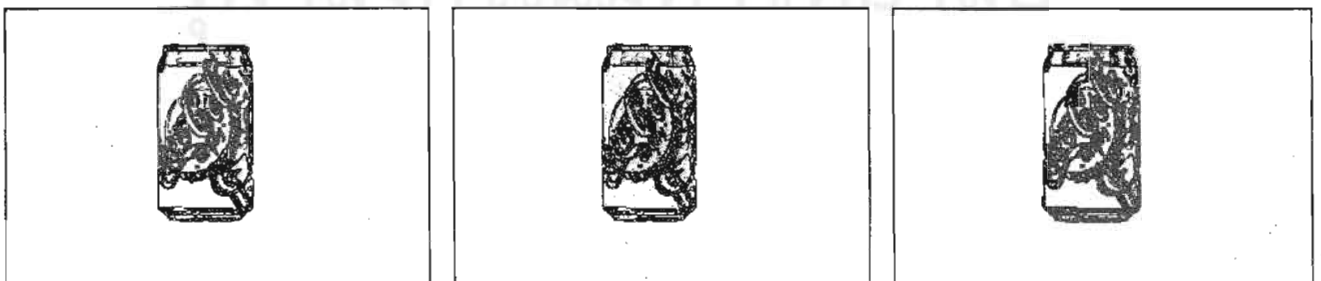


Fig 3.3-12 ภาพแสดงผลลัพธ์ที่ได้จากโปรแกรม COMPASS

3. Canny เป็นกระบวนการในการหาขอบด้วยวิธีของแคนนี่ ซึ่งที่มีประสิทธิภาพสูง แต่ใช้เวลาในการประมวลผลค่อนข้างนาน สามารถปรับเปลี่ยนค่า Standard Deviation, Lower Threshold และ Higher Threshold ได้ และสามารถกำหนดให้โปรแกรมทำการตัดค่าขอบเป็นภาพขาวดำได้

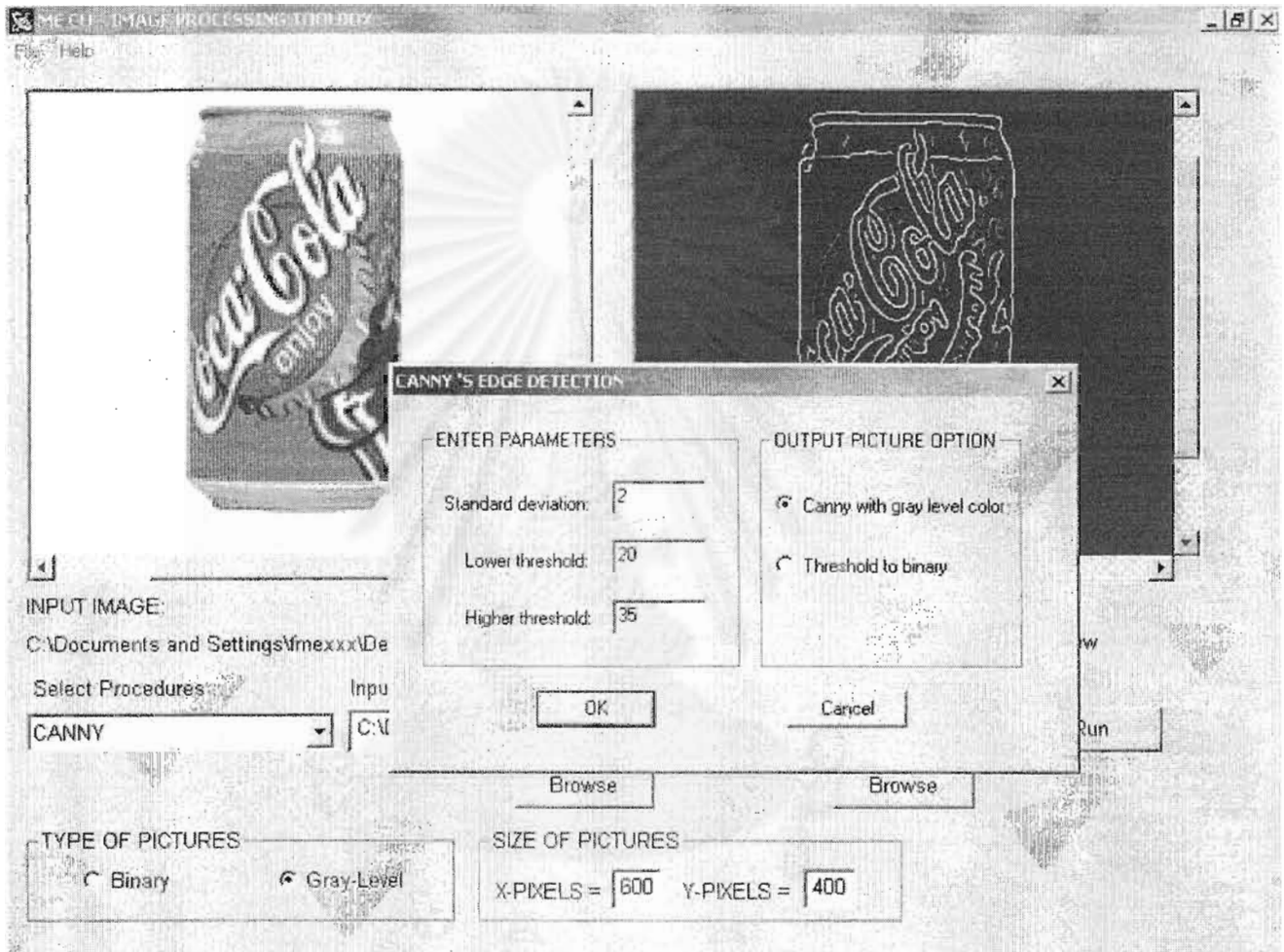


Fig 3.3-13 ภาพแสดงโปรแกรม CANNY



Fig 3.3-14 ภาพแสดงผลลัพธ์ที่ได้จากโปรแกรม CANNY

4. Blur คือ กระบวนการที่ทำให้ภาพมีความเรียบ และ มีความต่อเนื่องมากขึ้น

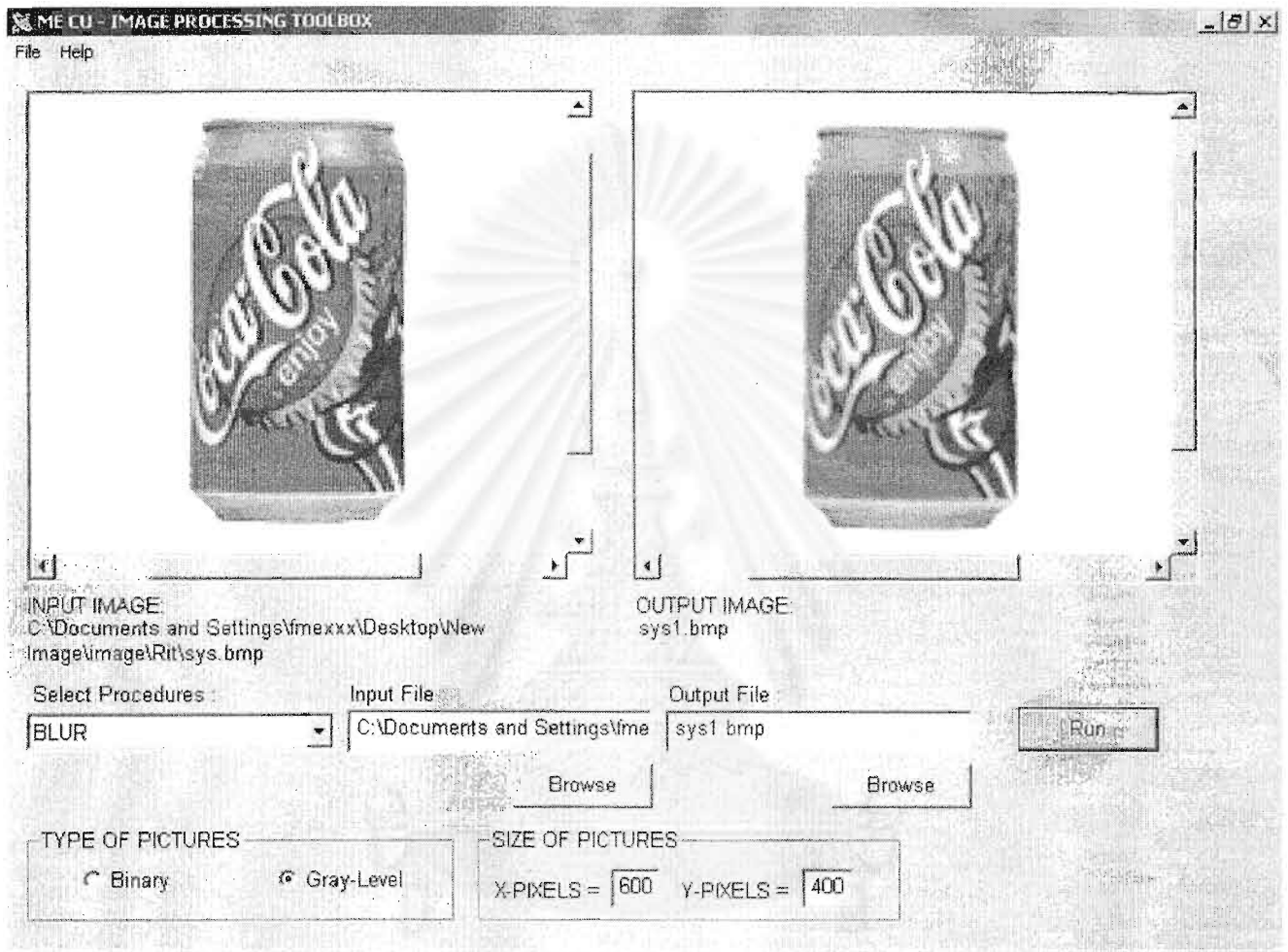


Fig 3.3-15 ภาพแสดงโปรแกรม BLUR



## E. กระบวนการประมวลผลภาพไบนารี

1. Boundary เป็นกระบวนการในการหาขอบของวัตถุในภาพไบนารี ดังรูป

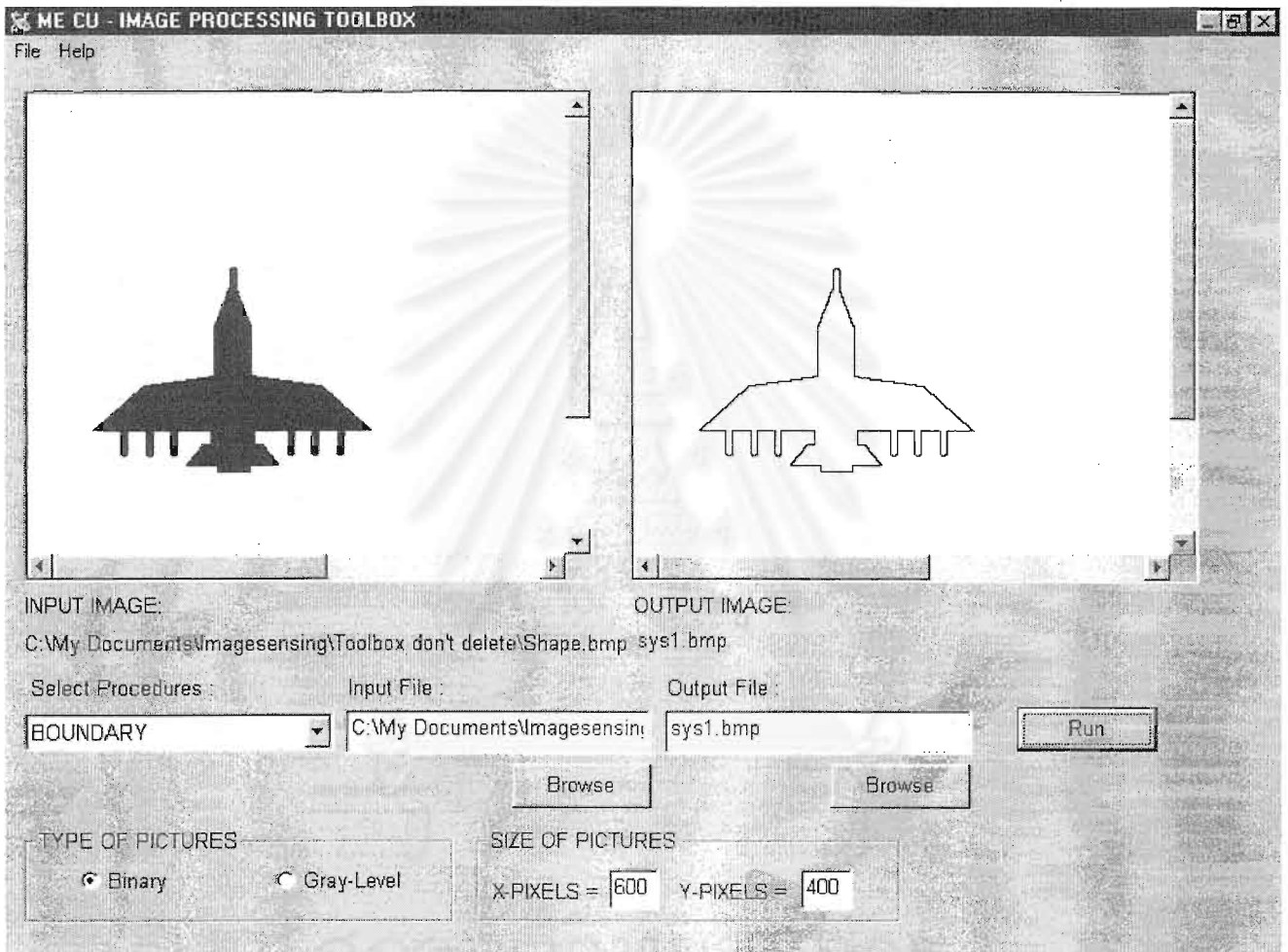


Fig 3.3-16 ภาพแสดงโปรแกรม BOUDARY



2. Close เป็นการปิดช่องว่างเล็กๆ ภายในวัตถุ

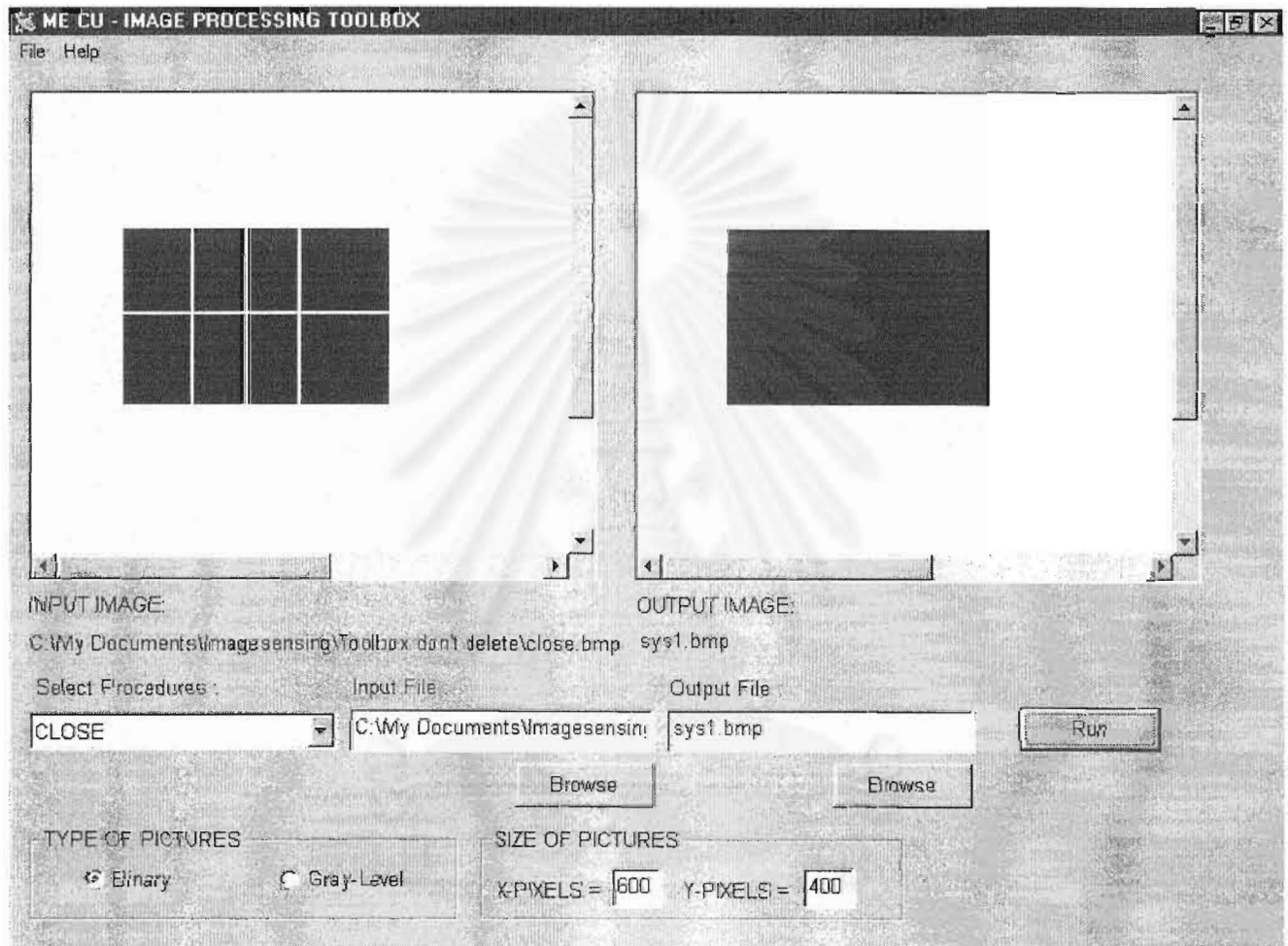


Fig 3.3-17 ภาพแสดงโปรแกรม CLOSE

3. Open เป็นการตัดส่วนกิ่งก้านเล็กๆ ที่ติดกับวัตถุ

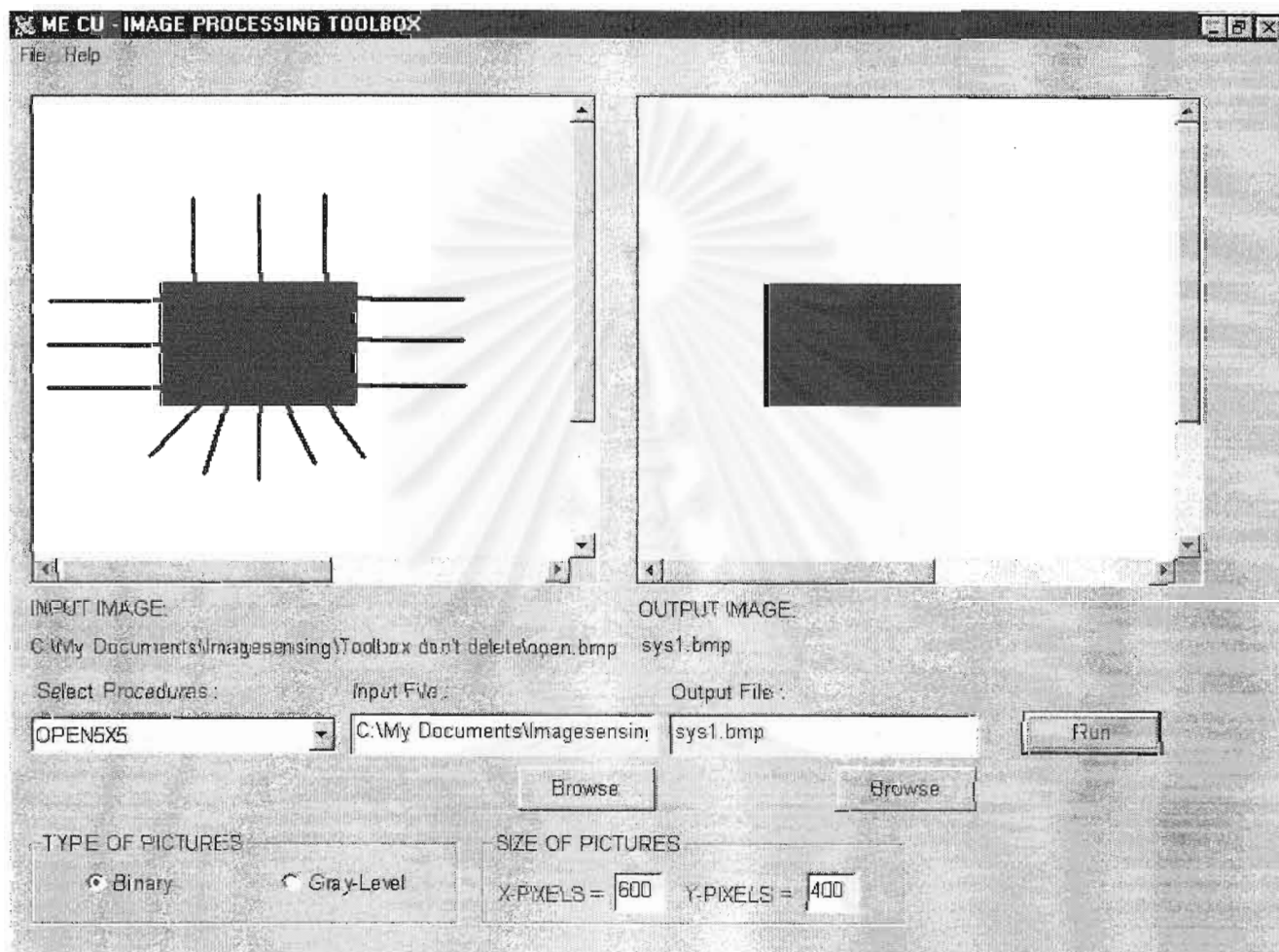
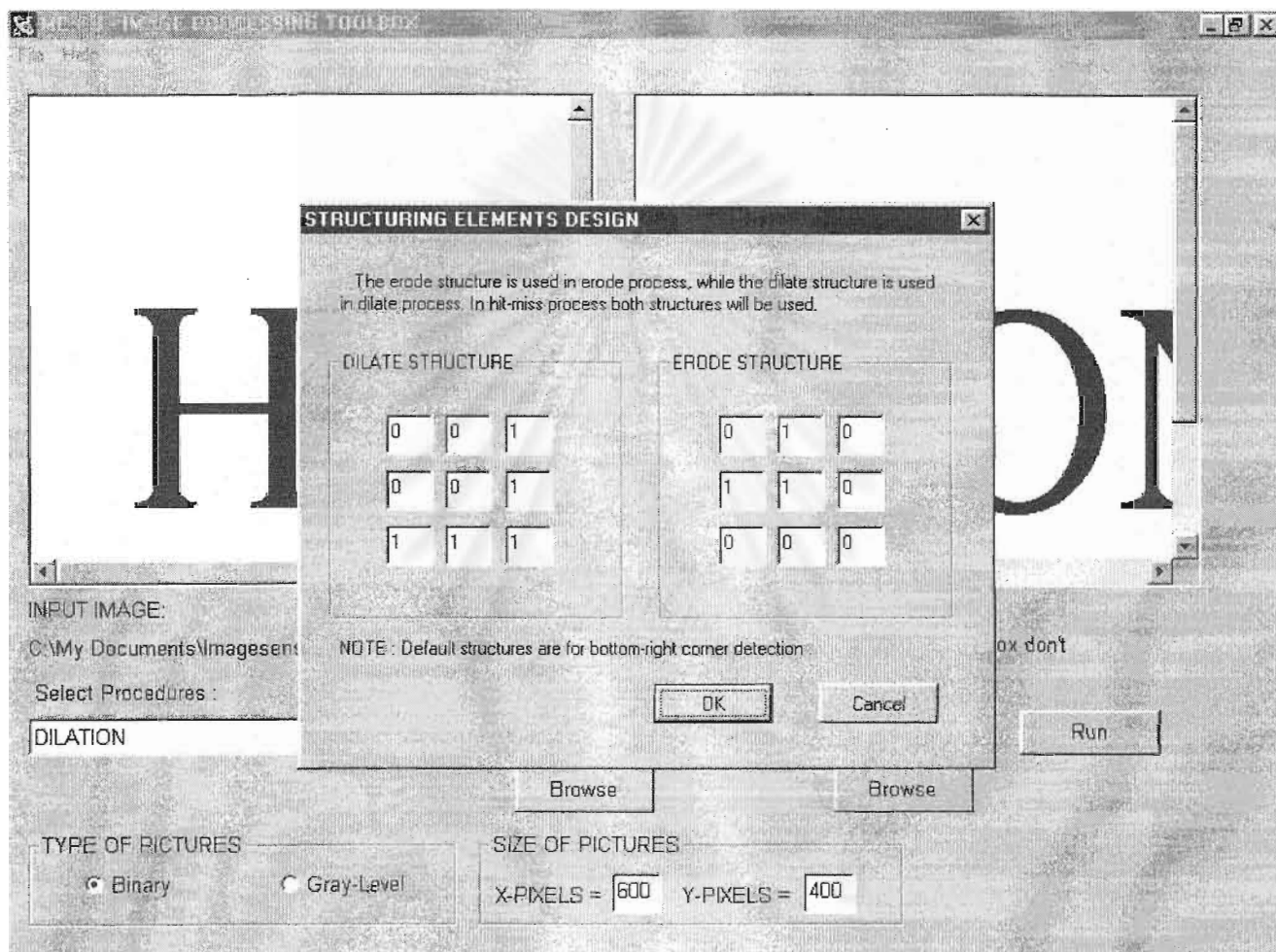


Fig 3.3-18 ภาพแสดงโปรแกรม OPEN

4. Erode, Dilate และ Hit-miss เป็นกระบวนการทาง Logic พื้นฐานในการประมวลผลภาพ โดยใช้ Structuring elements ที่สามารถกำหนดได้ภายในโปรแกรม



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

Fig 3.3-19 ภาพแสดงโปรแกรม ERODE, DILATE และ HIT-MISS

5. Convex-Hull กระบวนการนี้ใช้ในการหาสี่เหลี่ยมในแนวทแยงมุมที่เล็กที่สุดที่สามารถบรรจุวัตถุทั้งชิ้นได้ ดังรูป

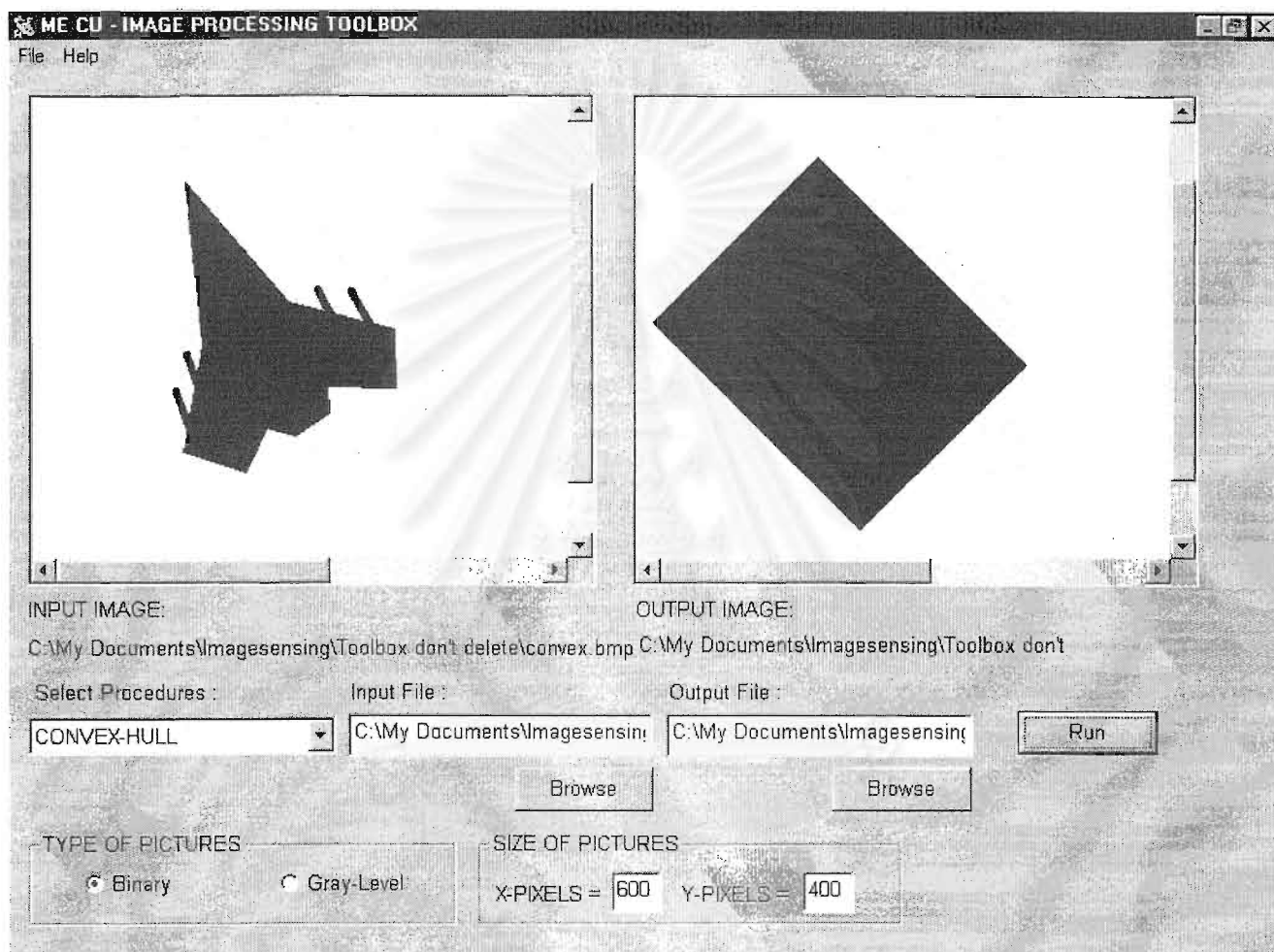


Fig 3.3-20 ภาพแสดงโปรแกรม CONVEX-HULL

## 6. Skeleton เป็นกระบวนการในการหาโครงร่างของวัตถุ ดังรูป

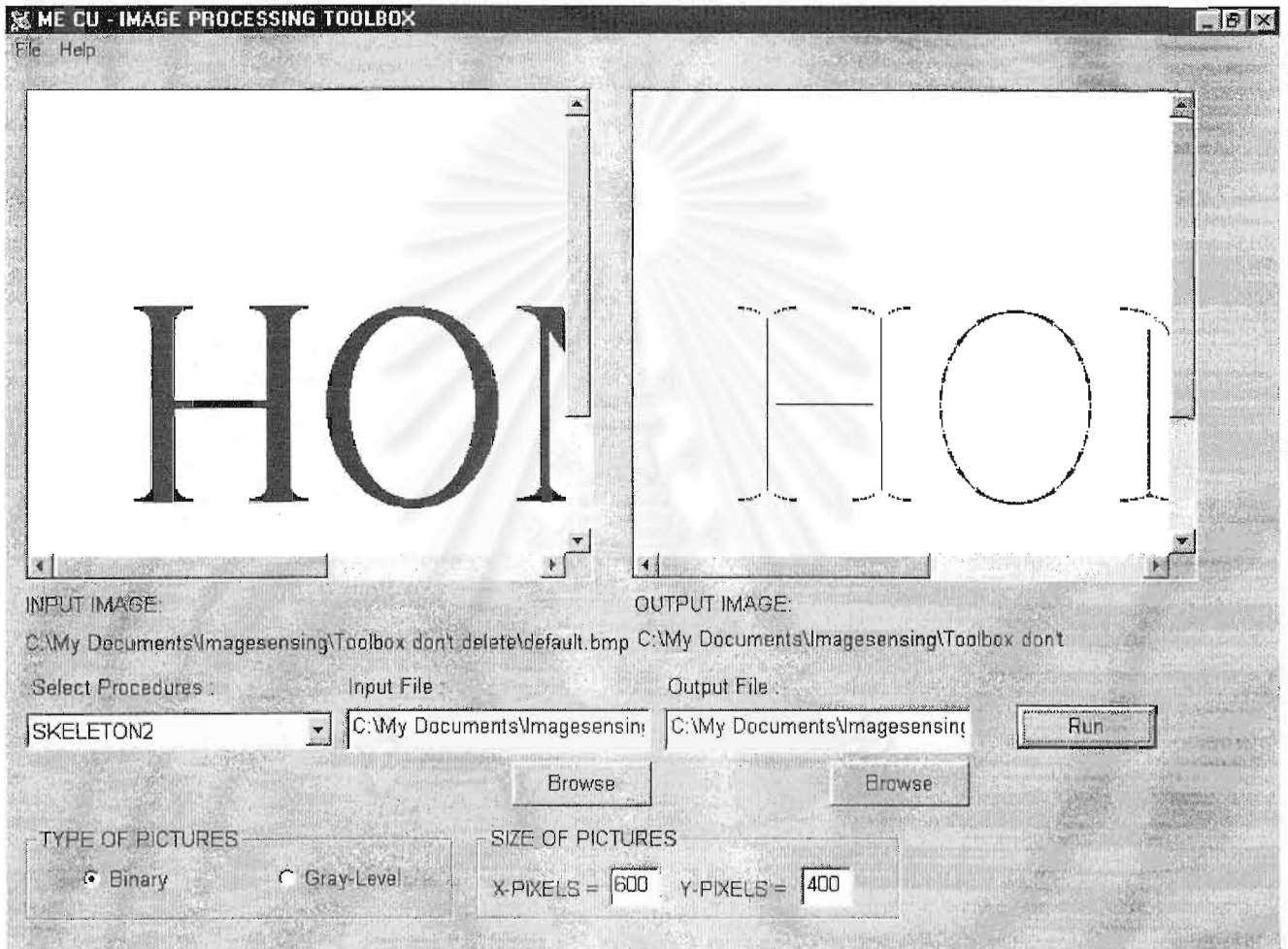


Fig 3.3-21 ภาพแสดงโปรแกรม SKELETON

7. Thinning เป็นกระบวนการที่ทำให้วัตถุบางลง ดังรูป

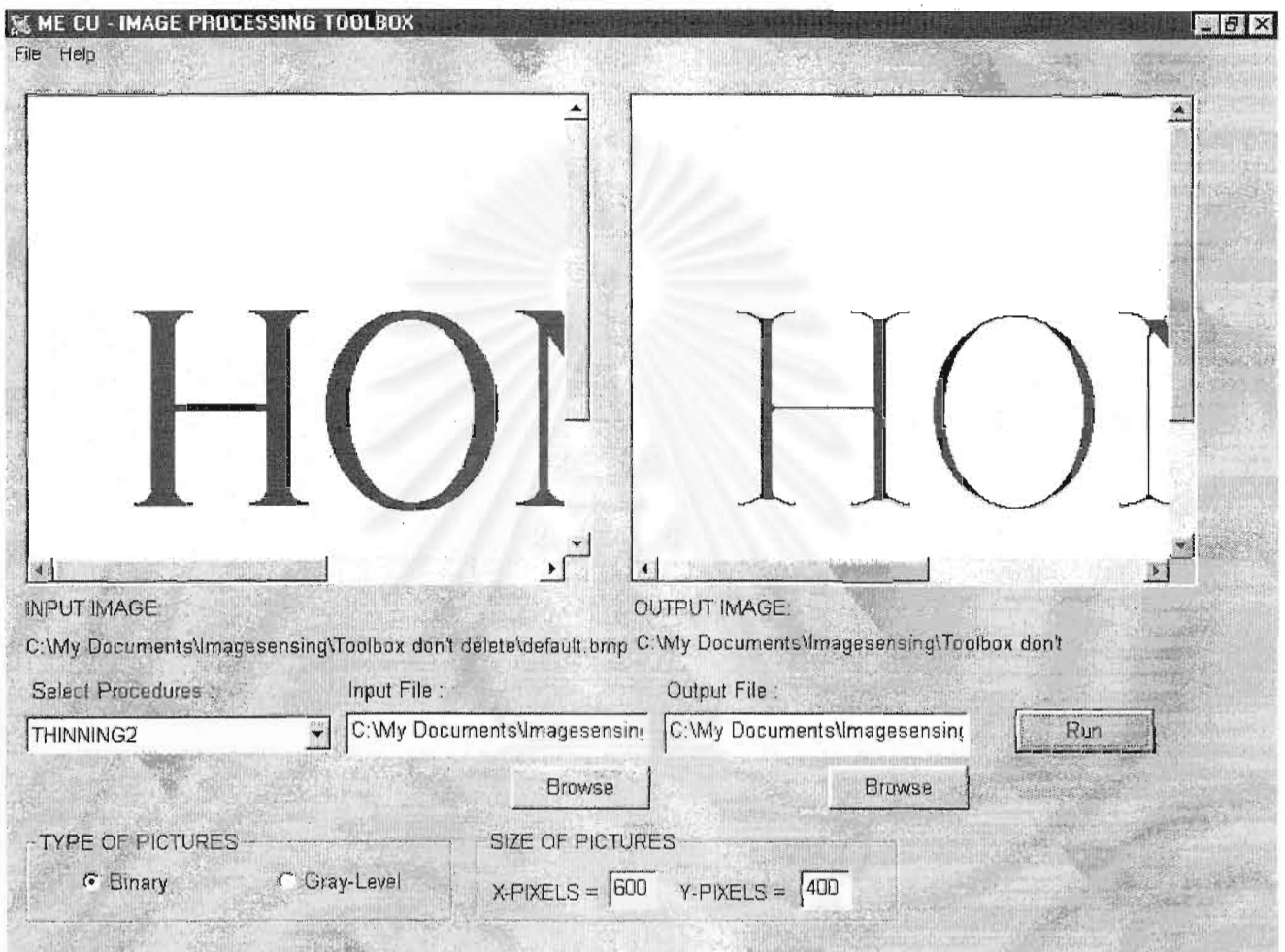


Fig 3.3-22 ภาพแสดงโปรแกรม THINNING

จุฬาลงกรณ์มหาวิทยาลัย



8. Thickening เป็นกระบวนการที่ทำให้วัตถุหนาขึ้น ดังรูป

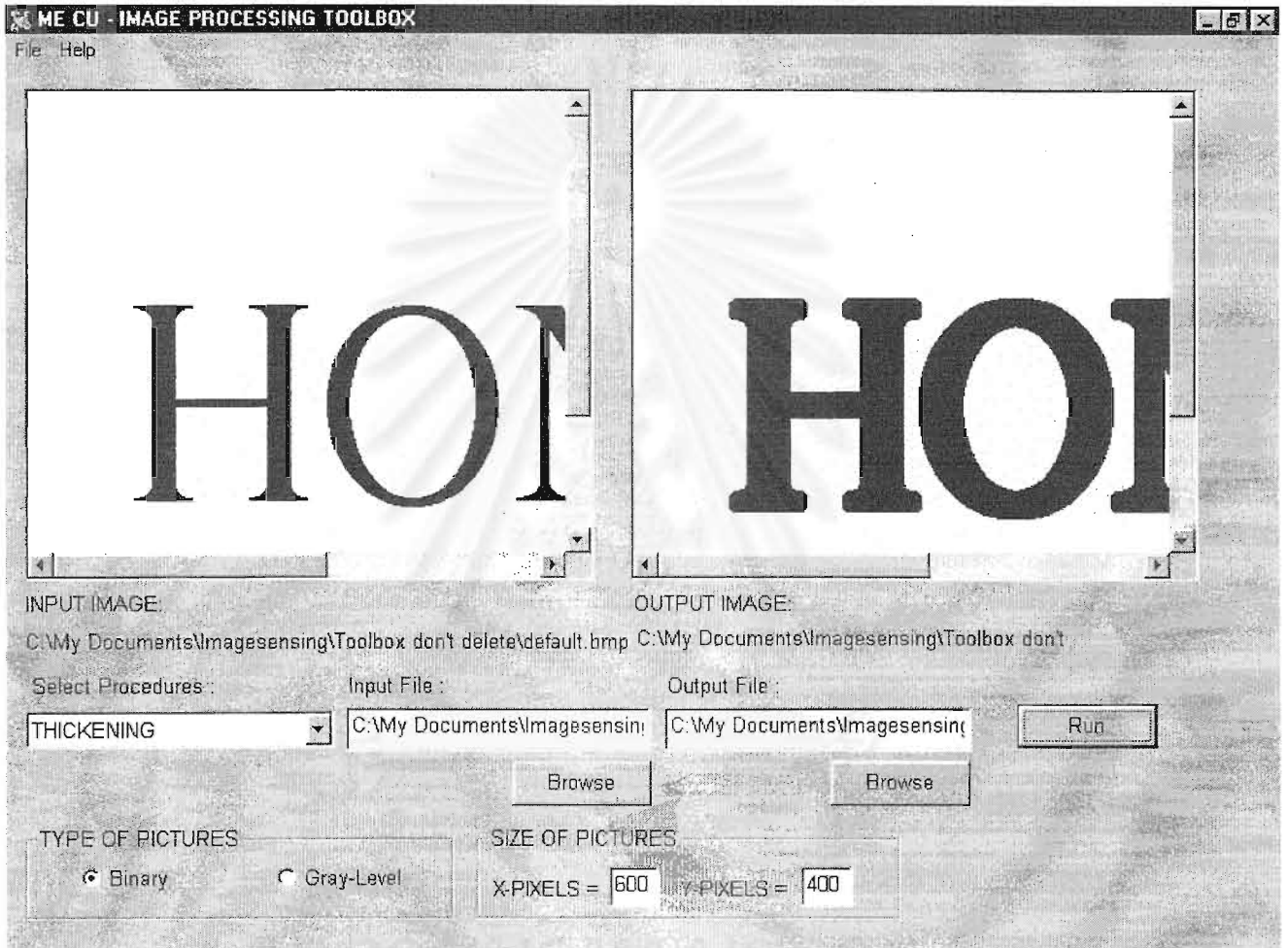
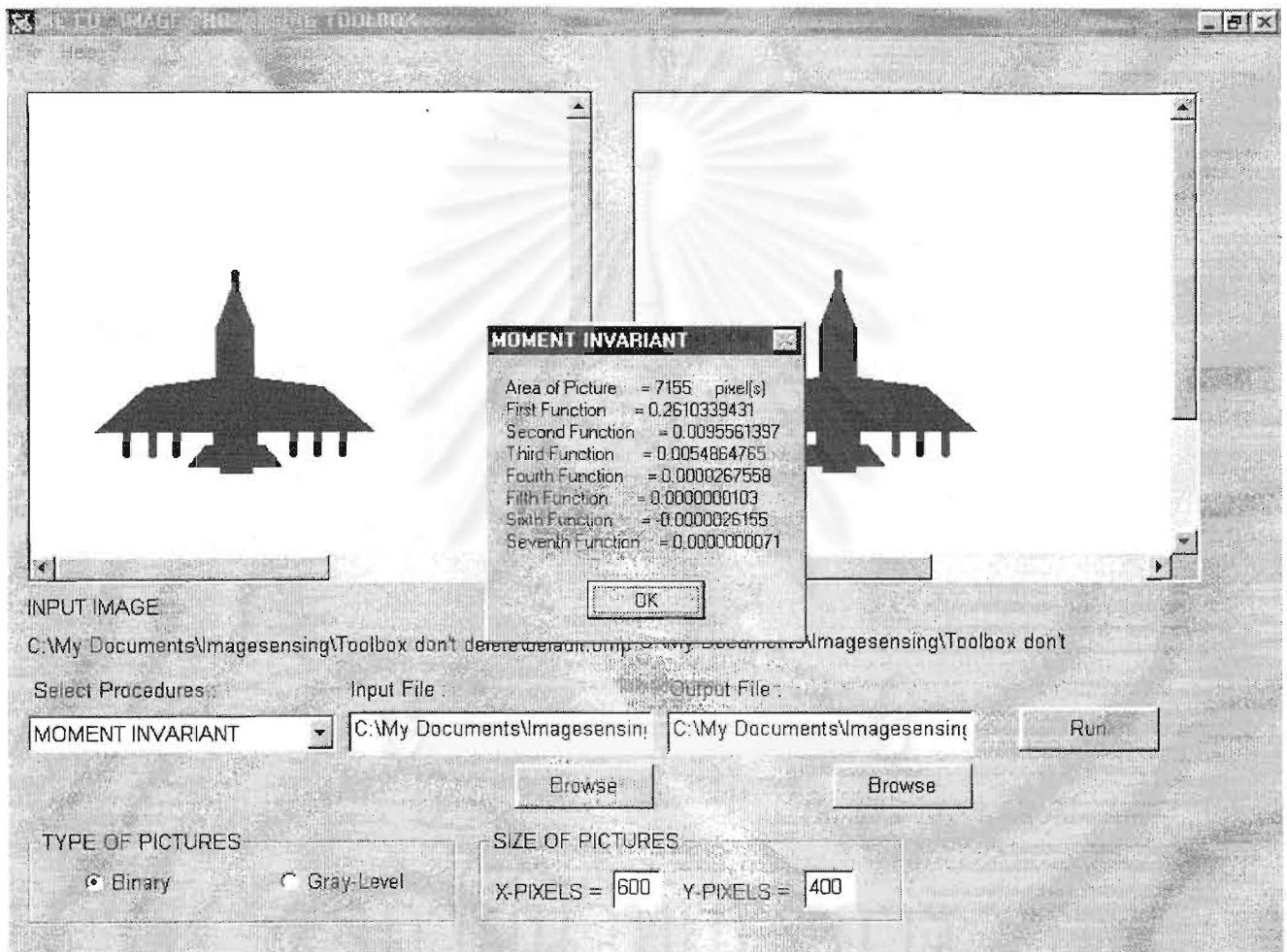


Fig 3.3-23 ภาพแสดงโปรแกรม THICKENING

9. Moment Invariant ใช้สำหรับคำนวณค่า พื้นที่ และ ฟังก์ชันโมเมนต์ไม่ผันแปรทั้ง 7 ค่าของวัตถุในภาพ



จุฬาลงกรณ์มหาวิทยาลัย

Fig 3.3-24 ภาพแสดงโปรแกรม MOMENT INVARIANT



10. Object Separation ใช้สำหรับแยกวัตถุในภาพออกจากกัน ดังรูป

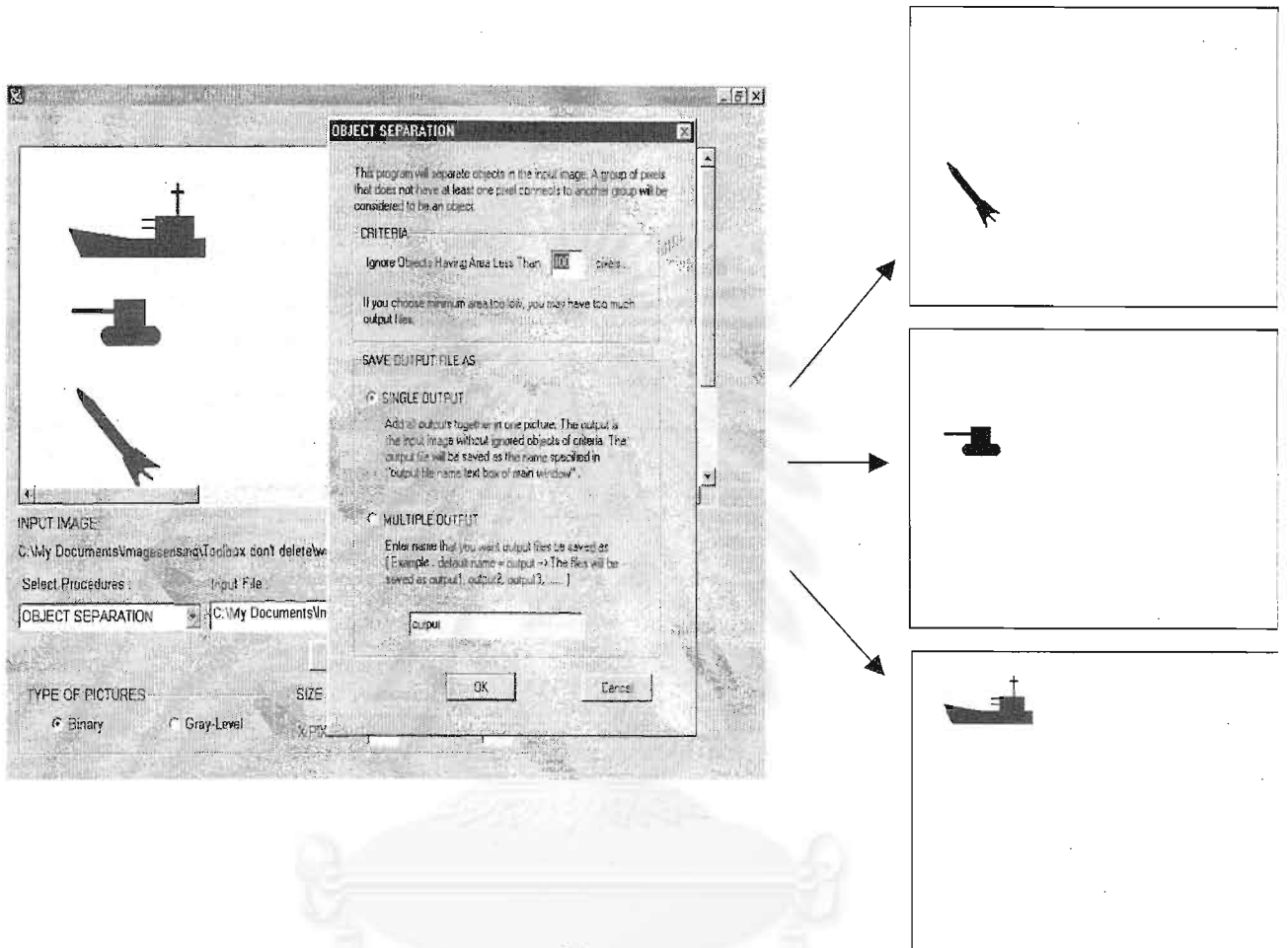
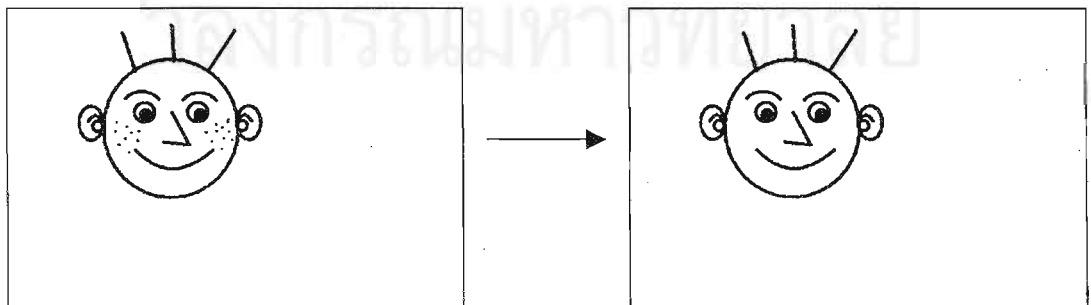


Fig 3.3-25 ภาพแสดงโปรแกรม OBJECT SEPARATION

นอกจากนี้ ยังสามารถใช้ Object Separation ในการลดรายละเอียดของภาพได้ โดยให้เลือกบันทึกภาพเป็น Single output พร้อมกำหนดขนาดวัตถุเล็กสุดเป็น Pixel ที่จะทำการลบทิ้งเพื่อลดรายละเอียด ตัวอย่างเช่น



ภาพก่อนทำการลดรายละเอียด

ภาพหลังทำการลดรายละเอียด

Fig 3.3-26

11. Object Extraction ใช้ในการค้นหาวัตถุที่มีรูปร่างเหมือนภาพต้นแบบ (Prototype image) ภายในภาพ Input โดยอาศัยหลักการของ Moment Invariant

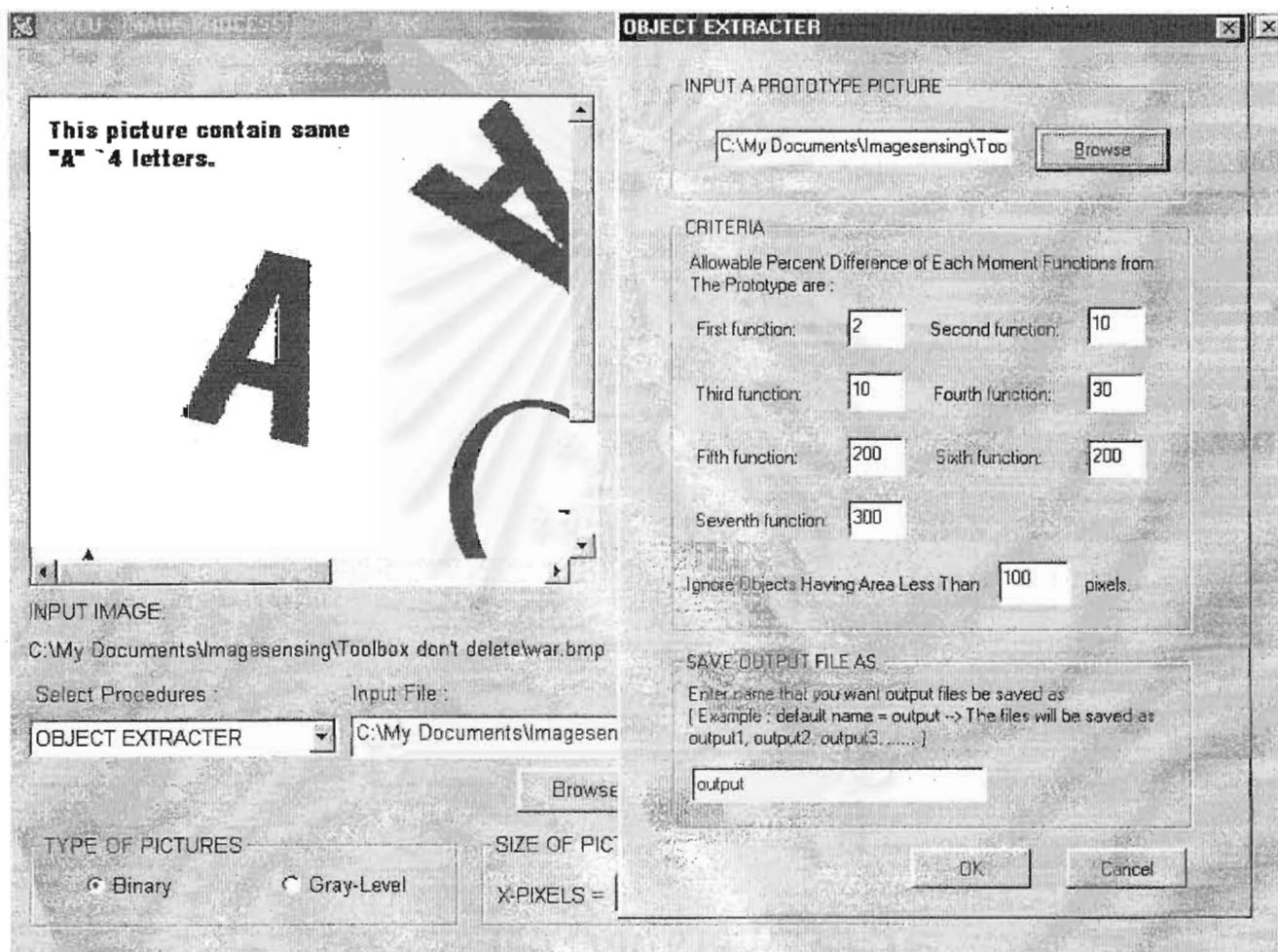
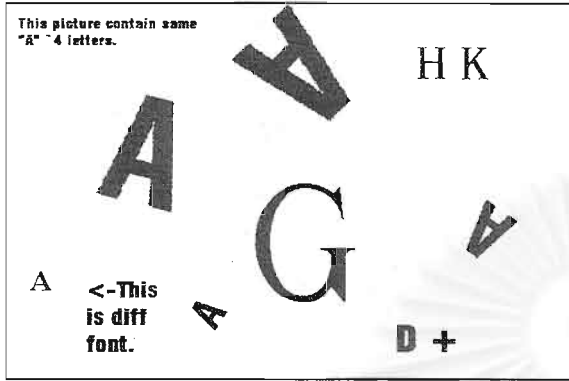
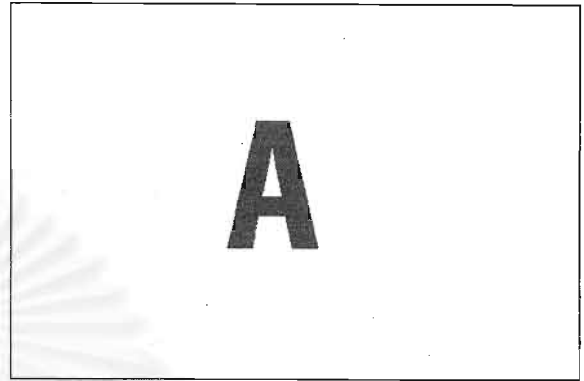


Fig 3.3-27 ภาพแสดงโปรแกรม OBJECT EXTRACTOR



ภาพ INPUT



ภาพวัตถุต้นแบบ

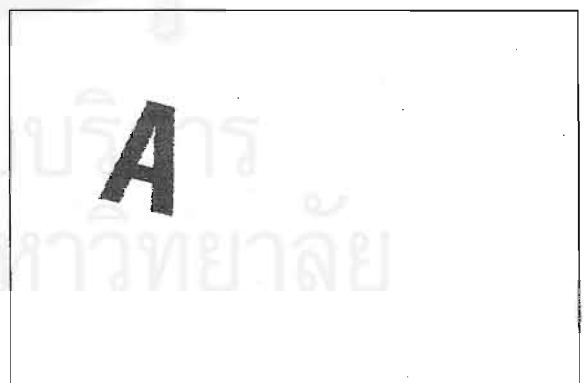
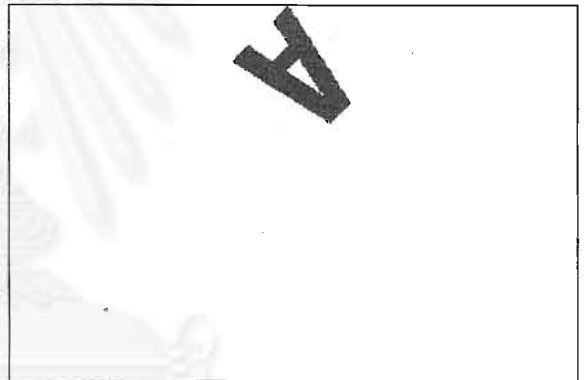


Fig 3.3-28 ภาพผลลัพธ์จากการแยกวัตถุต้นแบบออกจากภาพ Input

## 12. Orientation ใช้ในการหาแนวการวางตัวของวัตถุในภาพ

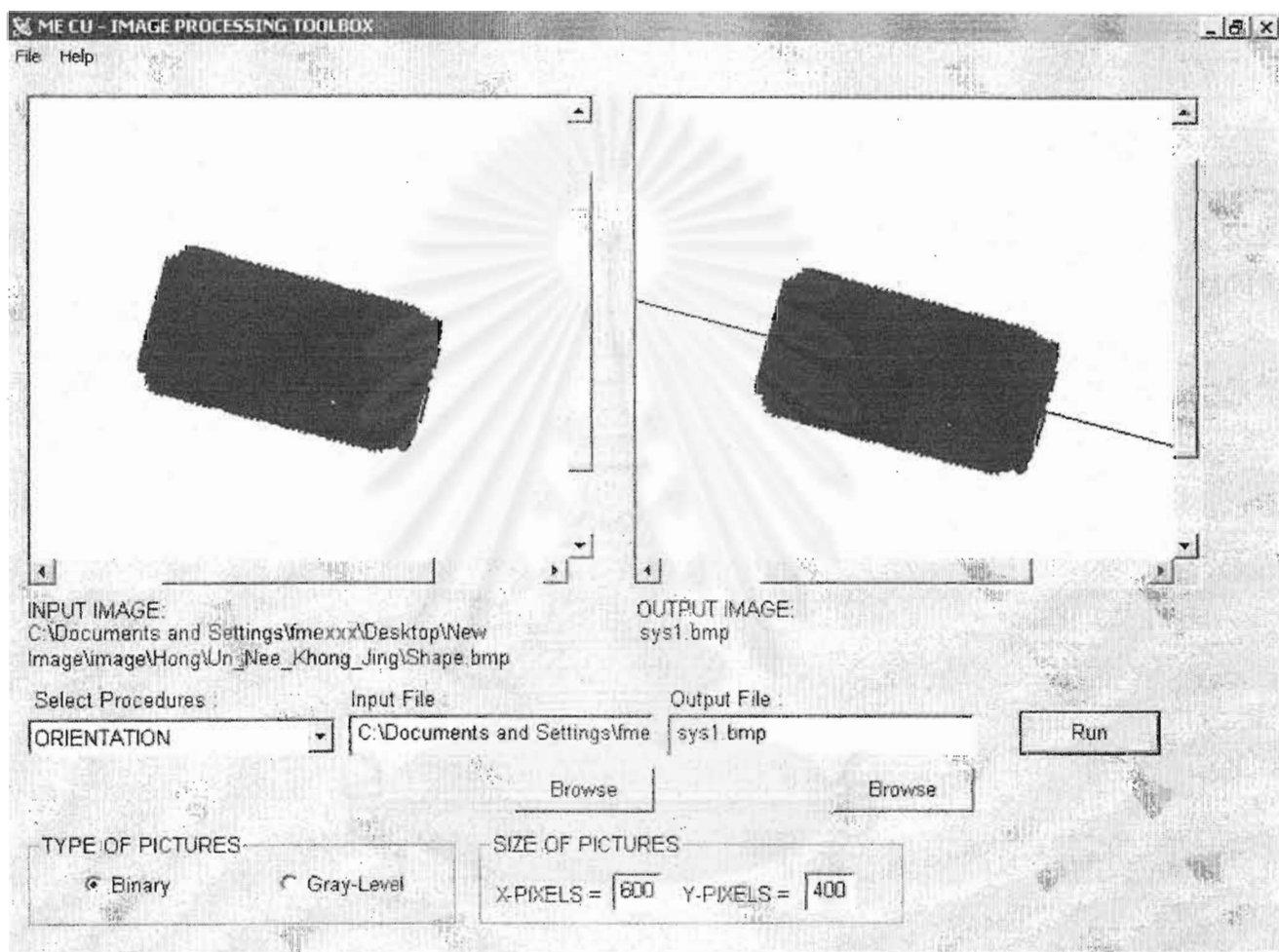


Fig 3.3-29 ภาพแสดงโปรแกรม ORIENTATION

### 3.4 บทสรุป

งานวิจัยเกี่ยวกับการพัฒนาโปรแกรมเพื่อค้นหาวัตถุเป้าหมาย ภายในสิ่งแวดล้อมที่ไม่ได้จัดเตรียมล่วงหน้า นับว่าเป็นงานวิจัยที่กำลังเป็นที่สนใจอย่างมากในปัจจุบัน เนื่องจากโปรแกรมหาดังกล่าวสามารถนำไปพัฒนาเป็นระบบการมองเห็นที่สมบูรณ์สำหรับควบคุม หุ่นยนต์ หรือ เครื่องจักรกล ต่างๆ ภายใต้สภาวะแวดล้อมทั่วไปได้ โครงการนี้กล่าวได้ว่าเป็นจุดเริ่มต้นในการรวบรวมข้อมูล และ พัฒนาโปรแกรมสำเร็จรูปสำหรับการประมวลผลภาพเบื้องต้น เพื่ออำนวยความสะดวกในการศึกษาทดสอบกรรมวิธีประมวลผลสัญญาณภาพ ต่อไป นอกจากนี้ยังได้ทำการพัฒนาระบบการค้นหาวัตถุแบบคร่าวๆ (Roughly Search) ด้วยกลุ่มสีและข้อมูลรูปร่างวัตถุเพื่อค้นหาบริเวณในรูปภาพที่มีโอกาสมีวัตถุอยู่ โดยใช้หลักการค้นหาด้วยข้อมูลระดับความเข้มสี ซึ่งให้ผลลัพธ์ที่แม่นยำในระดับหนึ่ง อย่างไรก็ตามการดังกล่าวยังมีข้อจำกัดที่ต้องปรับปรุงในด้าน เวลาในการประมวลผล รวมทั้งต้องหาทางแก้ปัญหาความไวแสงของกล้อง CCD อีกด้วย

#### งานวิจัยต่อเนื่อง

จากกระบวนการค้นหาวัตถุอย่างคร่าวๆ (Roughly Search) เราจะได้บริเวณซึ่งมีโอกาสมีวัตถุอยู่ การพัฒนาโปรแกรมในขั้นต่อไป ควรเป็นการพัฒนาระบบการค้นหาอย่างละเอียดในบริเวณที่ได้จากกระบวนการ Roughly Search เป็นการเช็คความถูกต้องของผลลัพธ์ที่ได้ หากบริเวณดังกล่าวไม่มีวัตถุอยู่จริง ก็ให้กล้อง CCD เปลี่ยนตำแหน่งถ่ายภาพ เพื่อทำการค้นหา Roughly Search ใหม่อีกครั้ง จนกว่าเราจะหาบริเวณตำแหน่งที่ตรวจสอบได้อย่างแน่ชัดแล้วว่ามีวัตถุเป้าหมายอยู่ในบริเวณนั้นจริง โดยอาศัยข้อมูลของขนาดวัตถุ และ แนวการวางตัวของวัตถุ เราก็จะสามารถคำนวณตำแหน่งวัตถุเป้าหมายนั้นเทียบกับตำแหน่งของกล้อง CCD ได้ เมื่อถึงจุดนั้น หากเราทำการพัฒนาระบบการมองเห็นดังกล่าวเข้ากับส่วนของหุ่นยนต์ เราก็จะสามารถสั่งการให้หุ่นยนต์เคลื่อนที่ไปยังตำแหน่งนั้นเพื่อเก็บวัตถุเป้าหมายได้อย่างสมบูรณ์

จุฬาลงกรณ์มหาวิทยาลัย



## 4. บทสรุปโครงการ

โครงการคัดแยกและเก็บขยะรีไซเคิลโดยใช้กล้อง CCD มีจุดประสงค์ที่จะพัฒนาหุ่นยนต์ให้มีศักยภาพสูงขึ้นในการทำงานในสภาพแวดล้อมที่ไม่ได้จัดเตรียมไว้ โดยใช้กล้องดิจิทัลซึ่งเป็นอุปกรณ์ตรวจรู้อย่างหนึ่งมาประยุกต์ใช้กับหุ่นยนต์ เพื่อให้หุ่นยนต์สามารถค้นหาวัตถุที่ต้องการ และสามารถเดินไปหยิบวัตถุนั้นได้ โดยในรายงานฉบับนี้จะประกอบด้วย 2 ส่วนคือการสร้างและพัฒนาตัวหุ่นยนต์เก็บขยะและการพัฒนากระบวนการในการตรวจหาวัตถุในสิ่งแวดล้อมทั่วไป ซึ่งถือเป็นการทำงานพื้นฐานของงานที่ซับซ้อนต่อไป

ส่วนของการสร้างหุ่นยนต์ได้ทำการพัฒนาหุ่นยนต์ซึ่งแบ่งออกได้เป็นส่วนหลัก 3 ส่วนคือ ส่วนของระบบกลที่ประกอบด้วย กลไกการจับ ระบบแขนและโครงสร้างฐานล่าง ส่วนของระบบไฟฟ้า และส่วนของโปรแกรมควบคุมการทำงานของหุ่นยนต์ โดยในส่วนของระบบกลและไฟฟ้าได้ทำการคิด conceptual design พิจารณาความเหมาะสมของการนำมาใช้งานโดยเฉพาะกลไกการจับ (มือกล) ตลอดจนความเหมาะสมของวัสดุที่ใช้ เพื่อให้ทุกๆ ส่วนสามารถนำมาใช้งานได้มีประสิทธิภาพสูง นอกจากนี้การติดตั้งอุปกรณ์ตรวจรู (sensor) ที่สอดคล้องกับการใช้งาน เช่น กล้อง CCD และ limit switch ทั้งนี้การออกแบบในแต่ละชั้นส่วน ได้พยายามคำนึงถึงความง่ายในการตรวจสอบและแก้ไขเป็นหลัก ทุกชั้นตอนจึงได้จัดให้มีการแสดงผลเพื่อช่วยในการตรวจสอบดังกล่าว อนึ่ง หุ่นยนต์ตัวนี้สามารถทำงานได้ตามวัตถุประสงค์ นอกจากนี้ได้นำหุ่นยนต์เก็บขยะนี้เข้าร่วมการแข่งขันหุ่นยนต์บริการสมองกลเพื่อทดสอบการทำงานของหุ่นยนต์ ซึ่งประสบผลสำเร็จเป็นที่น่าพอใจในการแข่งขัน เมื่อรวมโครงการนี้เข้ากับกระบวนการคัดแยกขยะด้วยกล้อง CCD หุ่นยนต์จะสามารถทำงานได้ตามเป้าหมาย และสามารถประยุกต์ใช้กับงานอื่นที่มีลักษณะคล้ายกันได้อีกด้วย

ส่วนที่สองคือการพัฒนาระบบการในการค้นหาวัตถุเป้าหมายในสิ่งแวดล้อมทั่วไป โดยแบ่งขั้นตอนการทำงานเป็น 2 ขั้นตอนคือ ขั้นตอนการป้อนข้อมูลวัตถุเป้าหมายและขั้นตอนการตรวจหาวัตถุเป้าหมายในสภาพแวดล้อมที่หลากหลาย ทางคณะวิจัยได้เสนอขั้นตอนในการป้อนข้อมูลที่สามารถป้อนข้อมูลของวัตถุเป้าหมายได้โดยง่ายและเป็นอัตโนมัติเพียงนำวัตถุเป้าหมายมาวางในบริเวณฉากหลังที่เป็นสีพื้น แล้วหุ่นยนต์ก็จะจำแนกว่าส่วนไหนเป็นส่วนที่เป็นวัตถุเป้าหมายและสามารถนำข้อมูลของวัตถุนั้นไปใช้ในขั้นตอนการค้นหาวัตถุในสภาพแวดล้อมทั่วไปต่อไปได้โดยง่าย ในขั้นตอนการตรวจหาวัตถุเป้าหมายในสภาพแวดล้อมทั่วไป ทางคณะวิจัยได้เสนอกระบวนการในการค้นหาวัตถุเป้าหมาย 2 วิธีการและพัฒนาโปรแกรมเพื่อค้นหาวัตถุเป้าหมายในภาพด้วยกล้อง CCD ตามกระบวนการที่เสนอ

โดยเริ่มจากการศึกษาพื้นฐาน เช่น หลักการระบบการมองเห็นของคอมพิวเตอร์ หลักการของกล้องรับภาพ และ ทฤษฎีการประมวลผลสัญญาณภาพเบื้องต้น ได้แก่ กระบวนการหาขอบของวัตถุในภาพแบบต่างๆ เช่น การตรวจหาขอบโดยใช้ ตัวปฏิบัติการเกรเดียนท์ ตัวปฏิบัติการคอมพาส ตัวปฏิบัติการลาปลาเซียน ตัวปฏิบัติการลดคาสติคเกรเดียนท์ และ กรรมวิธีหาขอบของแคนนี่<sup>[15, 24]</sup> (ดูภาคผนวก C) รวมทั้งกระบวนการประมวลผลสำหรับรูปภาพไบนารีเบื้องต้น ซึ่งได้แก่ การคำนวณพื้นที่ ตำแหน่ง และ แนวการวางตัวของวัตถุในภาพ และ ค่าฟังก์ชันโมเมนต์ไม่ผันแปรของวัตถุ และ กรรมวิธีการแปลงรูปแบบวัตถุในภาพไบนารี เพื่อหาลักษณะเฉพาะ (Features) บางอย่าง

เช่น โครงร่างของวัตถุ เส้นรอบรูปของวัตถุ ทฤษฎีทั้งหมดดังกล่าวได้ถูกรวบรวมสร้างขึ้นเป็นโปรแกรม Image Processing Toolbox เพื่อใช้ในการศึกษา ทดสอบกระบวนการประมวลผลภาพต่อไป

หลังจากได้ทำการศึกษาและพัฒนาโปรแกรมการประมวลผลภาพแบบต่างๆ มาเป็นระยะเวลาหนึ่ง ผู้จัดทำก็ได้ทำการพัฒนากระบวนการในการค้นหาวัตถุเป้าหมายภายในสิ่งแวดล้อม พร้อมทั้งพัฒนาโปรแกรมทดสอบเปรียบเทียบไปพร้อมกันไปด้วย โดยในช่วงแรก ผู้จัดทำได้พยายามที่จะใช้วิธีการค้นหาโดยใช้ข้อมูลของ ซึ่งพบว่าการค้นหาด้วยวิธีการดังกล่าวจะมีความยากมาก เนื่องจากข้อมูลของเส้นที่ได้มีเป็นจำนวนมาก รวมทั้งยังมีปัญหาเส้นขาดตอนอีกด้วยจึงไม่สามารถนำข้อมูลเส้นเหล่านี้มาใช้ในการค้นหาวัตถุในสภาพแวดล้อมต่างๆไปได้ ภายหลังจึงได้ทำการพัฒนากระบวนการค้นหาวัตถุขึ้นมาสองวิธี โดยอยู่บนพื้นฐานข้อมูลที่เป็นพื้นที่แทนข้อมูลเส้น

วิธีแรกเป็นวิธีการค้นหาแบบคร่าวๆ (Roughly Search) ด้วยข้อมูลสี (ความเข้มแสง) โดยกระบวนการนี้ได้แนวความคิดมาจากกระบวนการค้นหาวัตถุขั้นต้นของมนุษย์ เริ่มต้นด้วยการสำรวจสีประกอบหลักๆของวัตถุที่ต้องการค้นหา จากนั้นจะกวาดสายตาค้นหาอย่างรวดเร็วเพื่อมองหาบริเวณที่มีสีดังกล่าวในสิ่งแวดล้อม เราสามารถลดโอกาสของความผิดพลาดได้ด้วยการค้นหาบริเวณที่มีกลุ่มสีหลักเหล่านั้นอยู่ในบริเวณใกล้เคียงกันมากขึ้น ก็จะทำให้สามารถตัดบริเวณที่บังเอิญมีสีคล้ายกับวัตถุต้นแบบเพียงบางสีออกไปได้ จุดเด่นของกระบวนการค้นหานี้คือสามารถค้นหาวัตถุได้รวดเร็วมาก ในการวิจัยนี้ได้ใช้ข้อมูลภาพประเภท Gray scale หรือข้อมูลภาพความเข้มแสงซึ่งมีเพียง  $2^8$  (256) ระดับนั้น ทำให้อิทธิพลของแสงสว่างมีค่อนข้างมากและทำให้เกิดการผิดพลาดได้ ซึ่งเราสามารถเพิ่มศักยภาพของกระบวนการนี้โดยใช้ข้อมูลประเภทสี (RGB) ที่มีถึง  $2^{24}$  ระดับจะทำให้วิธีการนี้มีศักยภาพขึ้นอย่างมาก

ส่วนวิธีที่สองเป็น เป็นการค้นหาวัตถุโดยใช้ข้อมูลรูปร่างของวัตถุ ซึ่งใช้การแยกวัตถุในภาพออกเป็นวัตถุย่อยๆ แล้วทำการตรวจสอบความคล้ายของรูปร่างวัตถุที่พบในสิ่งแวดล้อมและวัตถุเป้าหมาย โดยใช้ค่าโมเมนต์ไม่ผันแปร Moment Invariants [ 13, 20, 21, 22 ] ซึ่งพบว่า วิธีการนี้มีประสิทธิภาพในการแยกวัตถุได้ดีในระดับหนึ่ง จุดเด่นของกระบวนการนี้คือสามารถที่จะแยกวัตถุย่อยๆ (บริเวณที่มีสีใกล้เคียงกัน) ออกมาได้ ดังนั้นจึงสามารถประยุกต์ใช้ในการหารูปทรงวัตถุ รวมทั้ง ตัวอักษร ลวดลายต่างๆบนวัตถุได้ด้วยการเพิ่มความแม่นยำในการค้นหาวัตถุจำเพาะ (specific object) ใดๆได้ เช่น จำแนกกระป๋องโค้กและกระป๋องเป๊ปซี่ได้ แต่ในการใช้งานกับข้อมูลภาพประเภท Gray scale ยังมีข้อจำกัดในเรื่องระดับ (ความละเอียด) ของข้อมูลซึ่งก็สามารถเพิ่มศักยภาพของกระบวนการนี้ได้โดยการใช้อข้อมูลภาพประเภทสี RGB จะทำให้ประสิทธิภาพของกระบวนการนี้ดีขึ้นมาก กระบวนการนี้ยังมีข้อจำกัดอีกอย่างหนึ่งคือกรณีที่วัตถุย่อยทับกันซึ่งจากวิธีการนี้จะถือว่าวัตถุที่ทับกันเป็นวัตถุเดียวกัน เราสามารถปรับปรุงกระบวนการนี้ได้ด้วยการนำภาพข้อมูลระยะทางจากกล้อง (Depth Image) [ 25 ] ที่ได้จากการใช้กล้อง 2 ตัวที่เรียกว่า Stereo Vision มาวิเคราะห์เพิ่ม เนื่องจากวัตถุที่ซ้อนกันในภาพนั้นในความเป็นจริงต้องมีอันหนึ่งอยู่หน้าและอีกอันอยู่หลัง ดังนั้นถ้าเรานำภาพข้อมูลระยะทางจากกล้องมาวิเคราะห์ด้วยจะทำให้สามารถจำแนกได้

### โครงการวิจัยต่อเนื่อง

ในการวิจัยต่อเนื่องนั้นตามที่ได้กล่าวไปข้างต้นแล้วบ้าง ในเรื่องการปรับปรุงโปรแกรมสำหรับข้อมูลภาพสี RGB รวมทั้งการนำภาพข้อมูลระยะทาง (Depth Image) มาวิเคราะห์ร่วมด้วยซึ่งจะเป็นการเพิ่มความความแม่นยำ

ของกระบวนการที่เสนอให้สูงยิ่งขึ้น นอกจากนี้ยังอาจปรับปรุงให้กระบวนการ รวมทั้งโปรแกรมมีความเร็วสูงขึ้น นอกจากนี้เมื่อหุ่นยนต์ตรวจพบวัตถุในสภาพแวดล้อมแล้วในการไปหยิบ จับวัตถุอาจนำวิธีการ Visual Servo มาประยุกต์ ในการกำหนดเส้นทางเพื่อไปให้ถึงวัตถุด้วย และยังอาจจะสามารถพัฒนาระบบติดต่อสื่อสารไร้สายระหว่างตัวหุ่นยนต์กับคอมพิวเตอร์ที่ตั้งไว้กับที่ นอกจากนี้ยังมีประเด็นในเรื่องการพัฒนาหุ่นยนต์ให้สามารถทำงานที่ซับซ้อนยิ่งขึ้น เช่น การประกอบ เป็นต้น



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



## 5. เอกสารอ้างอิง

### Hardware

1. คู่มือ ET-AVR V1.0 & ET-AVR V2.0 AVR RISC MICROCONTROLLER BARD DOWNLOADABLE FLASH , บริษัท อีทีที จำกัด
2. คู่มือการทำงาน ROBO-BASIC ด้วยบอร์ดไมโครคอนโทรลเลอร์ PB-2SX , Innovative Experiment Co.,Ltd.
3. รายงานโครงการทางวิศวกรรมเครื่องกล เรื่อง " X-Y Table ," ภาคปลายปีการศึกษา 2542
4. B.M. Hove and J.J.E. Slotine. "Experiments in Robotic Catching," *Proceedings of the 1991 American Control Conference*, Vol. 1, Boston, MA, pp. 380-385, June 1991.
5. W. Hong. "Robotic Catching and Manipulation Using Active Vision," *M.S. Thesis, Department of Mechanical Engineering, MIT, Cambridge, MA, August 1995.*
6. [http://www.robotics.utexas.edu/rrg/education/high\\_education/grasp\\_planning/index.html](http://www.robotics.utexas.edu/rrg/education/high_education/grasp_planning/index.html) (For Grasp Mechanic)
7. <http://mozu.mes.titech.ac.jp/research/snake/sg/sg.html> ( For Soft Gripper)
8. <http://www.st.com> (For IC datasheet)
9. <http://www.questlink.com> (For Electronic Component)
10. <http://www.atmel.com> (For AVR datasheet and instruction)
11. <http://www.etteam.com> (For the AVR instruction Assembler and Emulator)

### Software

12. Berthold Klaus Paul Horn. "Robot Vision," The MIT Press, McGrawHill 1997.
13. Anil K.jain. "Fundamentals of Image Processing," Prentice Hall.
14. Wolfram Stadler. "Analytical Robotics and Mechatronics," McGrawHill.
15. J.R. Parker. "Algorithms for Image Processing and Computer Vision," John Wiley & Sons, Inc.
16. T. S. Huang and G. Y. Tang. "A Fast Two Dimensional Median Filtering Algorithm," *IEEE Trans. Accoustic Speech, Signal Processing*, ASSP-27, p.13-18 1979.
17. G. S. Robinson, " Edge Detection by Compass Gradient Masks," *Computer Graphics Image Proc.* 6, p.492-501, 1977.
18. D. Marr and E. C. Hildreth. " Theory of Edge Detection," *Proc. R. Soc. Lond. B.* 270, p.187-217, 1980.

19. R. M. Haralick. "Zero Crossing of Second Directional Derivative Edge Detector," *Robot Vision*, SPIE 336, p.91-96, 1982.
20. M.K. Hu. "Visual Pattern Recognition by Moment invariants," *Computer methods in image Analysis*, Los Angeles: IEEE Computer Society, 1977.
21. M. R. Teague. "Image Analysis via the General Theory of Moments," *J. of Optical Society of America* 70, No. 8 , p.920-930, August, 1980.
22. R. Wong and E. Hall. "Scene Matching with Moment Invariants," *Computer Graphics and Image Processing* 8, p.16-24, 1978.
23. E.R. Davies and A. P. Plummer. " Thinning Algorithm: A Critique and a New Methodology," *Pattern Recognition* 14, p.53-63, 1981.
24. Canny, J. "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.PAMI-8,p.679-698, 1986.
25. Olivier Faugeras et al. "Real Time Correlation-based Stereo: Algorithm, Implementations and Applications," *Research Report, 2013 INRIA*, 1993.

## 6. ภาคผนวก

รายการภาคผนวก

Appendix A : บทนำสู่ระบบการมองเห็นของคอมพิวเตอร์

Appendix B : หลักการทำงานของกล้องรับภาพ

Appendix C : ทฤษฎีการประมวลผลภาพ

Appendix D : แบบหุ่นยนต์



จุฬาลงกรณ์มหาวิทยาลัย

## ภาคผนวก

### A. บทนำสู่ระบบการมองเห็นของคอมพิวเตอร์

การมองเห็น เป็นประสาทสัมผัสที่มีประสิทธิภาพมากที่สุด เนื่องจากมันให้ข้อมูลเกี่ยวกับภาวะแวดล้อมแก่เราเป็นจำนวนมาก ทำให้เราสามารถติดต่อประสานกับสิ่งแวดล้อมได้อย่างถูกต้อง โดยปราศจากการสัมผัสทางกายภาพใดๆ

ระบบการมองเห็น ยังเป็นประสาทสัมผัสที่ซับซ้อนที่สุด โดยมันจะต้องทำงานประสานกับความจำและประสบการณ์ ที่สะสมอยู่ในสมองเพื่อทำความเข้าใจ ภาวะแวดล้อมที่มองเห็น จึงไม่น่าแปลกใจเลยที่หลายครั้งความพยายามในการจำลองระบบการมองเห็นทางชีวภาพ มาเป็นระบบการมองเห็นของหุ่นยนต์ ด้วยคอมพิวเตอร์ จึงจบลงด้วยความล้มเหลว อย่างไรก็ตาม ระบบการมองเห็นของหุ่นยนต์ก็ถูกพัฒนาอย่างต่อเนื่อง โดยในปัจจุบัน เราสามารถพบเห็น เครื่องจักรกลหลายอย่างที่มีระบบการมองเห็นที่สามารถทำงานได้ใน สภาวะแวดล้อมที่หลากหลายได้ การใช้งานระบบการมองเห็นส่วนใหญ่มักอยู่ในอุตสาหกรรม ซึ่งเราสามารถทราบและจัดเตรียมสภาวะแวดล้อมในการมองเห็นล่วงหน้าได้ และ มักใช้เป็นระบบควบคุมแบบป้อนกลับ ตัวอย่างเช่น การใช้หุ่นยนต์หยิบจับสิ่งของที่เคลื่อนที่ไปตามสายพานลำเลียง เป็นต้น ในบางครั้งระบบการมองเห็นถูกใช้ในการหาจุดสำคัญบางจุดในภาพ ซึ่งมนุษย์ยากที่จะหาพบ โดยมากมักเป็นภาพที่ได้จากกรรมวิธีอื่นที่ไม่ใช่การมองเห็นทั่วไป ตัวอย่างเช่น การตรวจสอบภาพที่ได้จาก การเอ็กซ์เรย์ปอด หรือ การตรวจสอบจุดขาดของแผงวงจรอิเล็กทรอนิกส์ เป็นต้น

การประยุกต์ใช้งาน	ปัญหา
1. การคัดแยกจดหมาย การอ่านฉลาก การอ่านตัวอักษร	Character recognition
2. การตรวจหาเนื้องอก การวัดขนาด และ รูปร่าง อวัยวะภายใน	Medical image analysis
3. การจำแนกชิ้นส่วนบนสายการผลิต การตรวจสอบจุดบกพร่อง	Industrial automation
4. การตรวจรู้ และ ตีความวัตถุในภาพ การควบคุมการเคลื่อนที่ - ด้วยการป้อนกลับด้วยการมองเห็น (Visual feedback)	Robotics
5. การสร้างแผนที่จากภาพถ่าย การสังเคราะห์แผนที่ภูมิอากาศ	Cartography
6. การตรวจสอบลายนิ้วมือ ในระบบรักษาความปลอดภัยอัตโนมัติ	Forensics
7. การตรวจเป้าหมาย ระบบนำทางเรดาร์ และ เครื่องบิน - ในการลงจอด การนำร่องขีปนาวุธ	Radar imaging
8. การวิเคราะห์ภาพถ่ายดาวเทียม เพื่อการทำนายอากาศ และ เพื่อ ระบุแยกพื้นที่ชนบท พื้นที่เกษตรกรรม	Remote sensing

ตารางที่ A.1 แสดงตัวอย่างการใช้งานระบบการมองเห็นในงานสาขาต่างๆ

ปัจจุบันยังไม่มีระบบการมองเห็นสำหรับหุ่นยนต์ที่สมบูรณ์แบบ ที่สามารถใช้ได้กับทุกสภาวะแวดล้อม แต่มีระบบการมองเห็นที่ใช้กับงานเฉพาะอย่างเกิดขึ้นมากมาย ความยากในการพัฒนาระบบการมองเห็นสำหรับหุ่นยนต์ในแต่ละครั้งนั้น ยังอยู่ที่การพิจารณาความสมดุลระหว่าง ความถูกต้องแม่นยำของการทำความเข้าใจภาพ และ ความเร็วในการประมวลผล

การสร้างระบบการมองเห็น มีความสัมพันธ์โดยตรงกับความรู้สามสาขาวิชา ได้แก่

1. การประมวลผลภาพ (Image Processing) คือ กระบวนการที่เกี่ยวข้องกับการสร้างภาพใหม่ จาก ภาพที่มีอยู่ โดยอาจเป็นการกำจัดสัญญาณรบกวน (Noise) หรือ การทำให้ภาพ เบลอ (Blur) หรือ การเน้นขอบ (Edge accentuated) เป็นต้น

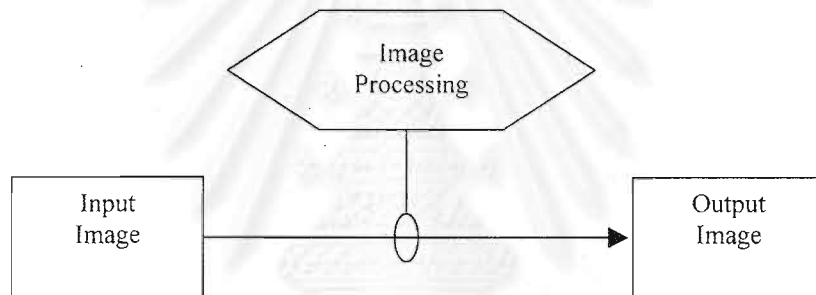


Fig A.1

2. การจำแนกรูปแบบ (Pattern Classification) โดยปกติ ข้อมูลเข้ากระบวนการนี้มักไม่ใช่รูปภาพ แต่เป็นกลุ่มข้อมูลตัวเลขที่ได้จากรูปภาพ ที่ใช้สำหรับแยกกลุ่มของวัตถุ ผลลัพธ์จากกระบวนการจำแนกรูปแบบ คือ การแบ่งวัตถุในภาพออกเป็นกลุ่ม

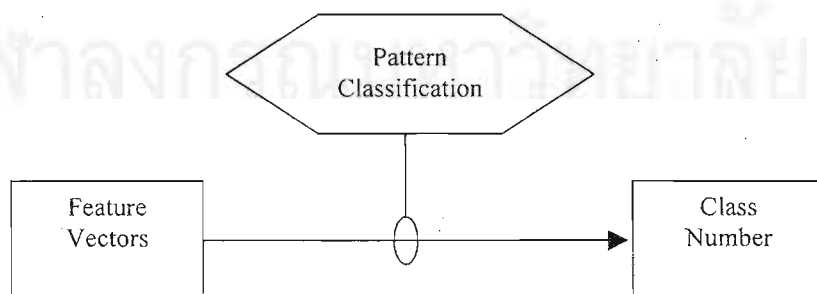


Fig A.2

3. การวิเคราะห์ภาพโดยรวม (Scene Analysis) คือ การแปลงการอธิบายอย่างง่ายที่ได้โดยตรงจากภาพ เป็นการอธิบายที่ซับซ้อนมากขึ้น เช่น การแปลงข้อมูลเส้น และ จุดตัดในภาพเป็นคำอธิบายว่าแต่ละเส้นเป็นขอบของวัตถุใด และ วัตถุจัดเรียงซ้อนทับกันอย่างไร เป็นต้น

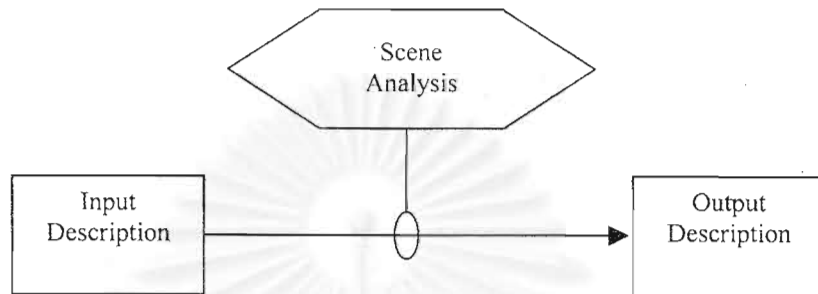


Fig A.3

ระบบการมองเห็นโดยทั่วไปมักมี ลักษณะการทำงานดังแผนภาพด้านล่าง

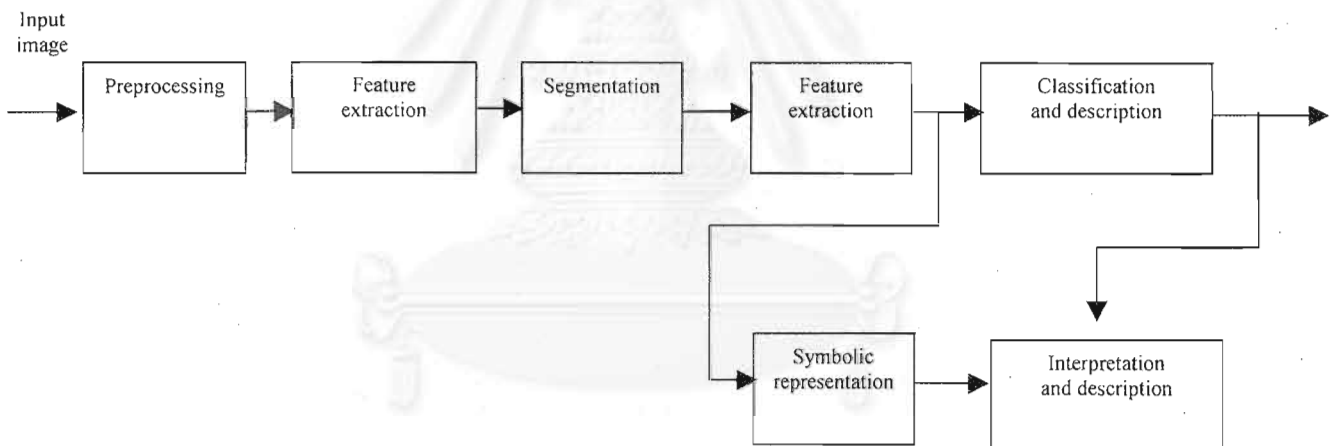


Fig A.4

เริ่มต้นจาก ภาพ Input จะถูก preprocess ก่อนทำการหา features สำหรับทำการแยกส่วนวัตถุ เช่น ทำการเน้นขอบวัตถุเพื่อทำการแยกวัตถุด้วยขอบของวัตถุ จากนั้นภาพวัตถุที่แยกแล้วจะถูกส่งไปยัง ตัวจำแนกวัตถุ และ ระบบทำความเข้าใจภาพ (Image understanding system) การจำแนกวัตถุจะทำการรวมชิ้นส่วนเข้าเป็นวัตถุเดียวกันหลายชิ้น ซึ่งถูกระบุด้วยเลเบล จากนั้น ระบบทำความเข้าใจภาพ จะทำการตีความว่าวัตถุแต่ละชิ้นในภาพมีความสัมพันธ์กันอย่างไร

## B. หลักการทำงานของกล้องรับภาพ

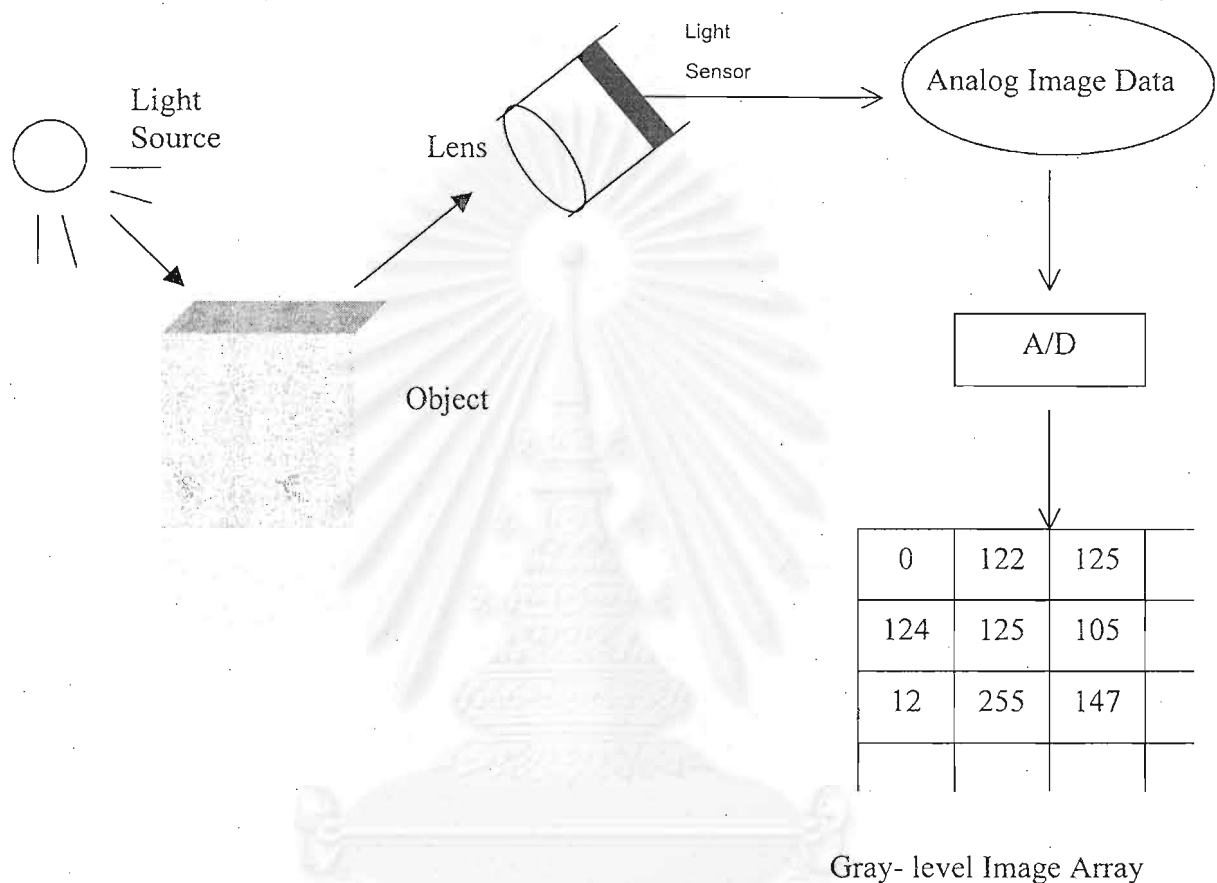


Fig B.1

จากรูปอธิบายหลักการทำงานของกล้องรับภาพได้ดังนี้ แสงที่สะท้อนจากวัตถุมาเข้า อุปกรณ์ตรวจจับแสง ถูกเปลี่ยนเป็นสัญญาณทางไฟฟ้าขนาดออก และ ถูกแปลงอีกที่เป็นข้อมูลตัวเลขอาเรย์ในรูปแบบพิกเซล

อุปกรณ์ตรวจจับแสงแบบ CCD มีโครงสร้างเป็นซิลิกอน ซึ่งจัดวางเป็นแถวอาร์เรย์สองมิติแทนแต่ละพิกเซล อาศัยหลักการเกิดคู่อิเล็กตรอน - โฮล (electron - hole pair) เมื่อมีโฟตอนตกกระทบ อย่างไรก็ตามไม่ใช่โฟตอนทุกตัวที่ตกกระทบจะก่อให้เกิดคู่อิเล็กตรอน - โฮลได้ เพราะมีบางส่วนทะลุผ่านอุปกรณ์ตรวจจับออกไป และ บางส่วนสะท้อนกลับออกไป สัดส่วนของพิกเซลอิเล็กตรอน ต่อ จำนวนพิกเซลของโฟตอน ซึ่งเป็นตัววัดความไวของตัวตรวจจับแสง เรียกว่า ประสิทธิภาพควอนตัม (Quantum efficiency) ซึ่งประสิทธิภาพควอนตัมนี้ ขึ้นอยู่กับพลังงานของโฟตอน หรือก็คือความยาวคลื่นของแสงนั่นเอง ประจุไฟฟ้าจากการเกิดคู่อิเล็กตรอน - โฮลสามารถถ่ายเทจากตำแหน่งหนึ่งไปยังอีกตำแหน่งได้ด้วยความต่างศักย์ทางไฟฟ้าสถิตซึ่งควบคุมด้วยสัญญาณนาฬิกา จึงเกิดสัญญาณทางไฟฟ้าขนาดออกในแต่ละ พิกเซล

อุปกรณ์ตรวจจับแสงแบบหลอดสุญญากาศ หรือ Vidicon tube อาศัยการตกกระทบของโฟตอนบนพื้นผิวที่ไวแสงในหลอดสุญญากาศที่เป็นขั้วแคโทดเพื่อให้เกิดการปลดปล่อยอิเล็กตรอนวิ่งไปยังขั้วแอโนด และ เกิดสัญญาณไฟฟ้าขึ้น เทคโนโลยี CCD นี้มาแทนเทคโนโลยีหลอดสุญญากาศ เนื่องจากมันมีอายุใช้งานดีกว่า และ ใช้กำลังน้อยกว่า รวมทั้ง CCD ยังมีประสิทธิภาพควอนตัมสูงเกือบเป็นอุดมคติในทุกช่วงความยาวคลื่น ขณะที่ Vidicon tube มีประสิทธิภาพควอนตัมต่ำมาก

อุปกรณ์ตรวจจับแสงซึ่งเปลี่ยนความเข้มแสงเป็นสัญญาณไฟฟ้าอีกอย่างหนึ่งนอกจาก CCD และ Vidicon tube เรียกว่า CID (Charge Injection Device) ซึ่งเป็นตัวตรวจจับแสงแบบซิลิคอนแต่แตกต่างกับ CCD อุปกรณ์แบบนี้ในแต่ละพิกเซล จะมี MOS capacitor สองตัวติดกัน ซึ่งถูกระบุแอดเดรสแยกจากกัน สามารถถูกใช้พร้อมกันหรือ อีสระต่อกันก็ได้ การกระตุ้นแยกเช่นนี้ทำให้ประจุสามารถเลื่อนระหว่างกันได้โดยไม่ทำให้ คุณภาพสัญญาณด้อยลง และสามารถตรวจวัดสัญญาณไฟฟ้าที่ได้ออกมาในรูปกระแสไฟฟ้า





C. ทฤษฎีการประมวลผลภาพ

C.1 การตรวจหาขอบของวัตถุภายในภาพ (EDGE DETECTION)

ปัญหาพื้นฐานที่สำคัญ ในกระบวนการประมวลผลภาพ คือ การตรวจหาขอบ เนื่องจากขอบเป็นตัวแบ่งแยกอาณาบริเวณของวัตถุ ดังนั้นมันจึงมีความสำคัญมากในการใช้แยกส่วนวัตถุ จุดขอบของภาพไบนารี ขาว - ดำ คือ จุดสีดำซึ่งติดกับจุดสีขาวรอบด้าน อย่างน้อย 1 จุด ซึ่งสามารถหาได้ด้วย อัลกอริทึมดังนี้

$$g(m, n) = [ u(m, n) \oplus u(m \pm 1, n) ] \text{OR} [ u(m, n) \oplus u(m, n \pm 1) ]$$

ส่วนกรณีขอบของภาพ Gray - level อาจนิยามได้ว่าเป็นจุดในภาพที่มีลักษณะการเปลี่ยนระดับความเข้มอย่างไม่ต่อเนื่อง เทคนิคหนึ่งที่ใช้กัน คือการหา gradient ของ  $f$  ตามแกน  $r$  ในทิศทางของ  $\theta$  ดังรูป

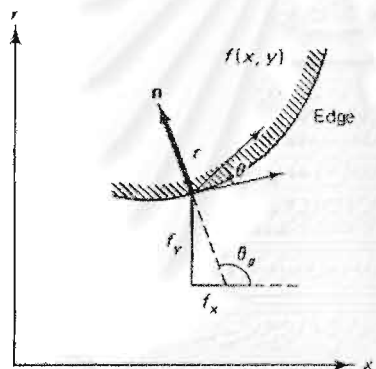


Figure 9.6 Gradient of  $f(x, y)$  along  $r$  direction.

Fig C.1

โดยทางคณิตศาสตร์เราสามารถหาทิศทางที่มีอัตราการเปลี่ยนแปลงปริมาณใดๆ เทียบกับตำแหน่งมากที่สุดได้จากเกรเดียนท์ ตามสมการด้านล่าง

$$\frac{\partial f}{\partial r} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial r} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial r} = f_x \cos\theta + f_y \sin\theta$$

ค่า maximum ของ  $\frac{\partial f}{\partial r}$  เมื่อ  $(\partial/\partial\theta)(\partial f/\partial r) = 0$  ซึ่งทำให้

$$-f_x \sin\theta_g + f_y \cos\theta_g = 0 \Rightarrow \theta_g = \tan^{-1}\left(\frac{f_y}{f_x}\right)$$

$$\left(\frac{\partial f}{\partial r}\right)_{\text{MAX}} = \sqrt{f_x^2 + f_y^2} = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

เมื่อ  $\theta_g$  เป็นทิศของขอบ (edge) ซึ่งจากแนวความคิดนี้ นำมาประยุกต์ใช้กับภาพดิจิทัล (digital image) เรียกว่า *masks*

ในการจะระบุว่าส่วนไหนเป็น วัตถุ (Object) หรือส่วนไหนเป็น พื้นหลัง (Background) เริ่มแรกเราต้องนิยามก่อนว่าสิ่งที่คั่น ระหว่าง วัตถุ และ พื้นหลัง คือ ขอบ (Edge)

ขอบในทางอุดมคติ (Ideal edge) คือ จุดที่เกิดการเปลี่ยนระดับของสีแบบฉับพลันทันทีทันใด โดยการเปลี่ยนแปลงมีความแตกต่างอย่างเห็นได้ชัด ดังรูป (a) แต่ในความเป็นจริงการเปลี่ยนระดับสีของ Edge ไม่ได้มีการเปลี่ยนแปลงอย่างเห็นได้ชัดขนาดนั้น แต่จะเป็นการไล่ระดับการเปลี่ยนแปลงดังรูป (b), (c) และ (d)

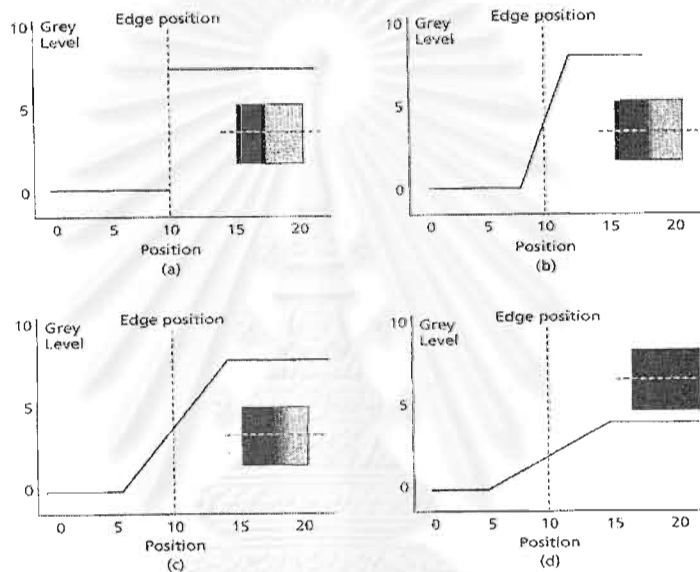


Figure 1.3 Step edges. (a) The change in level occurs exactly at pixel 10. (b) The same level change as before, but over four pixels centered at pixel 10. This is a *ramp* edge. (c) Same level change but over 10 pixels, centered at 10. (d) A smaller change over 10 pixels. The insert shows the way the image would appear, and the dotted line shows where the image was sliced to give the illustrated cross-section.

Fig C.2

ด้วยเหตุผลดังกล่าวจึงได้มีการพัฒนาทฤษฎีกระบวนการตรวจหาขอบขึ้นมามากมาย ดังเช่น

C.1.1 ตัวตรวจหาขอบแบบเกรเดียนท์ (Gradient Operators)

วิธีนี้แสดงโดยใช้ masks 2 คู่ที่วัด gradient ของภาพ  $u(m,n)$  ใน 2 orthogonal ดังรูปแผนภาพด้านล่าง

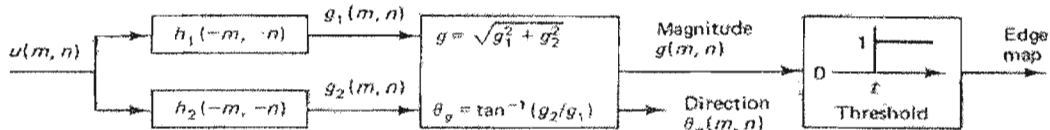


Figure 9.7 Edge detection via gradient operators.

Fig C.3

ซึ่งจะได้ ขนาด และ ทิศทางของ gradient ดังนี้

$$g(m,n) = \sqrt{g_1^2(m,n) + g_2^2(m,n)} \text{ และ } \theta_g(m,n) = \tan^{-1} \frac{g_2(m,n)}{g_1(m,n)}$$

หรืออาจจะหาขนาดของ gradient โดย

$$g(m,n) = |g_1(m,n)| + |g_2(m,n)|$$

จุด (pixel) ที่ตำแหน่ง  $(m,n)$  จะถูกตัดค่าที่ค่าหนึ่ง (threshold)  $t$  ดังนี้

$$\varepsilon(m,n) = \begin{cases} 1, & (m,n) \in I_g \\ 0, & \text{otherwise} \end{cases}$$

เมื่อ  $I_g = \{(m,n); g(m,n) > t\}$

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ C.2 แสดง masks gradient operator ต่างๆ

	$H_1$	$H_2$
Roberts	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Smoothed ( Prewitt )	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Sobel	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Isotropic	$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$

C.1.2 ตัวตรวจหาขอบแบบคอมพาส (Compass Operators)

Compass Operator เป็นการวัด gradient รอบทิศทางของจุดที่เรากำลังสังเกต กระบวนการทำงานดังเช่นแผนภาพต่อไปนี้

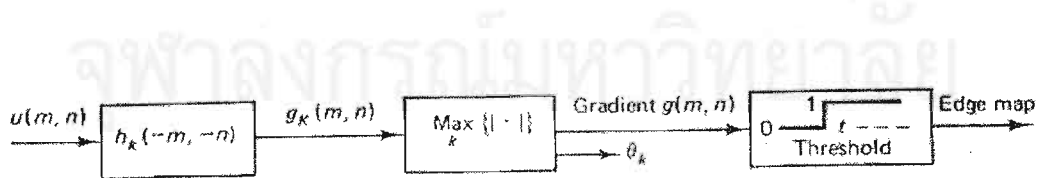
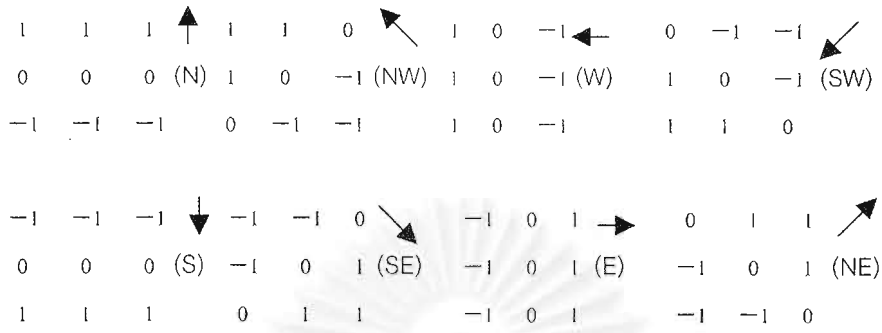


Figure 9.9 Edge detection via compass operators.

Fig C.4

แสดงการทำ Compass gradient หมุนรอบทวนเข็มนาฬิกาทีละ 45 องศา

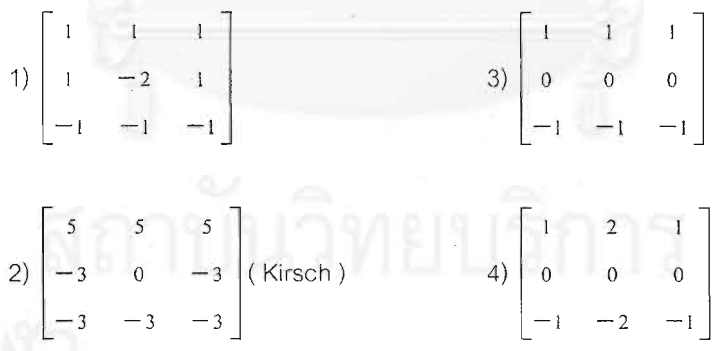


ให้  $g_k(m,n)$  เป็น compass gradient ในทิศทาง  $\theta_k = \pi/2 + k \pi/4, k = 0, \dots, 7, \dots$  จะได้ gradient ที่  $(m,n)$  คือ

$$g(m,n) = \max_k \{g_k(m,n)\}$$

ซึ่งสามารถตัดค่า Threshold ที่ค่าหนึ่งก่อนได้ เพื่อได้ edge map การจะเพิ่ม Higher angular resolution ของ compass gradient ทำได้โดยการเพิ่มขนาดของ mask

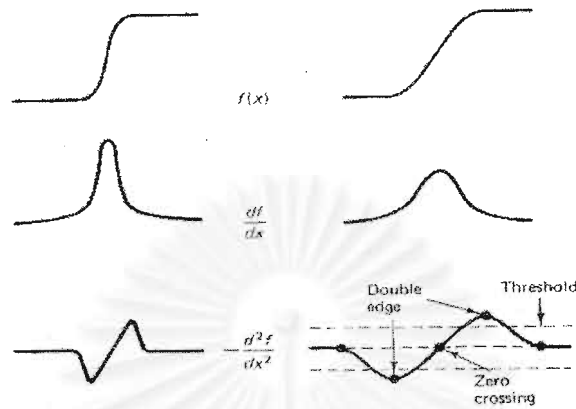
Compass Gradient ( North ) หมุนตามเข็มนาฬิกาทีละ 45 องศา ด้วย mask ต่างๆ



C.1.3 ตัวตรวจหาขอบแบบลาปลาเซียน (Laplace Operators)

วิธีที่กล่าวถึงไปก่อนหน้านี้เหมาะกับการเปลี่ยนแปลงของเมตริกซ์ที่เปลี่ยนแปลงอย่างกะทันหันแบบ Step function ซึ่งถ้าภาพมีการไล่สีอย่างเรียบจะให้ผลไม่ค่อยดีนัก จึงได้มีการพยายามคิดวิธีใหม่โดยใช้ Second-Order Derivatives หรือเรียกว่าวิธี Laplacian Operator นิยามโดย

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



(a) First and second derivatives for edge detection

Fig C.5

แสดง masks ต่างๆที่ใช้กับ Laplacian Operators

$$1) \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad 2) \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad 3) \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

เนื่องจากวิธีนี้เป็นการทำงาน Second-Order Derivatives ทำให้วิธีนี้มีความไวต่อสัญญาณรบกวนมาก ทำให้ผลที่ได้ไม่ดีเท่าที่คาดไว้ ดังนั้นจึงได้มีการปรับปรุงโดยเรียกว่า Zero-Crossings หรือ Laplacian of Gaussian functions นิยามโดย

$$h(m,n) \triangleq c \left[ 1 - \frac{(m^2 + n^2)}{\sigma^2} \right] \exp\left( -\frac{(m^2 + n^2)}{2\sigma^2} \right)$$

การทำ Zero-Crossings นี้ก็เหมือนกับวงจร Low pass filter ที่จะกำจัดสัญญาณรบกวนให้น้อยลงได้แต่จะไม่ได้กล่าวถึงในที่นี้ เนื่องจากมีวิธีที่ดีกว่าในหัวข้อต่อไป

#### C.1.4 ตัวตรวจหาขอบ แบบสโตคาสติกเกรเดียนท์ (Stochastic Gradient Operators)

วิธีการที่ผ่านมามีข้อเสียคือ จะใช้ไม่ได้ผลดีกับภาพที่มี noise มากๆ วิธีการของ Stochastic นี้มีคุณสมบัติที่ดีคือ จะให้ผลที่ดีต่อภาพที่มี noise ที่ที่ค่า SNR สูงมากๆได้ คล้ายๆกับการทำ Low pass filtering หรือเรียกอีกอย่างว่า Least square edge fitting

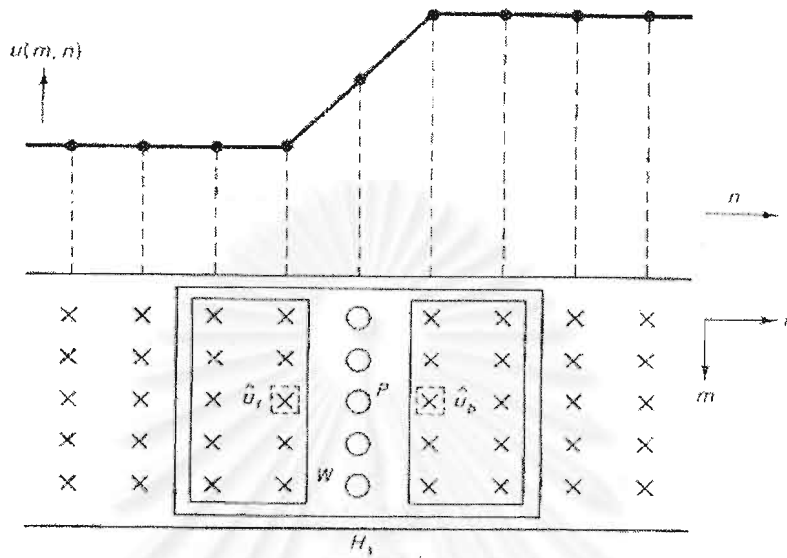


Figure 9.11 Edge model with transition region one pixel wide.

Fig C.6

อย่างเช่น ในการหา Edge ของ Object ที่จุด P จะทำการหา gradient ตามแนวอนดั่งนี้คือ

$$g_1(m, n) = \hat{u}_f(m, n-1) - \hat{u}_b(m, n+1)$$

ในที่นี้  $\hat{u}_f(m, n)$  และ  $\hat{u}_b(m, n)$  เป็นการประมาณไปข้างหน้า และย้อนหลัง อย่างเหมาะสมของ

$u(m, n)$  โดยถ้าทำการประมาณแบบ linear ของ  $\hat{u}_f(m, n)$  ได้ดังนี้

$$\hat{u}_f(m, n) = \sum_{(k,l) \in \varepsilon W} \sum a(k, l) v(m-k, n-l), \quad W = \{(k, l) : |k| \leq p, 0 \leq l \leq q\}$$

ตารางแสดงค่า Stochastic gradients  $H_1$

(โดย  $H_2 = H_1^T$ ),  $r_0(k, l) = 0.99^{\sqrt{k^2+l^2}}$  และ  $SNR = \sigma^2 / \sigma_n^2$

3 x 3 SNR=1

$$\begin{bmatrix} 0.97 & 0 & -0.97 \\ 1.00 & 0 & -1.00 \\ 0.97 & 0 & -0.97 \end{bmatrix}$$

3 x 3 SNR=9

$$\begin{bmatrix} 0.776 & 0 & -0.776 \\ 1.00 & 0 & -1.00 \\ 0.776 & 0 & -0.776 \end{bmatrix}$$

$$5 \times 5 \quad \text{SNR}=1 \quad \begin{bmatrix} 0.802 & 0.836 & 0 & -0.836 & -0.802 \\ 0.845 & 0.897 & 0 & -0.897 & -0.845 \\ 0.870 & 1.00 & 0 & -1.00 & -0.870 \\ 0.845 & 0.897 & 0 & -0.897 & -0.845 \\ 0.802 & 0.836 & 0 & -0.836 & -0.802 \end{bmatrix}$$

$$5 \times 5 \quad \text{SNR}=9 \quad \begin{bmatrix} 0.267 & 0.364 & 0 & -0.364 & -0.267 \\ 0.373 & 0.562 & 0 & -0.562 & -0.373 \\ 0.463 & 1.00 & 0 & -1.00 & -0.463 \\ 0.373 & 0.562 & 0 & -0.562 & -0.373 \\ 0.267 & 0.364 & 0 & -0.364 & -0.267 \end{bmatrix}$$

$$7 \times 7 \quad \text{SNR}=1 \quad \begin{bmatrix} 0.641 & 0.672 & 0.719 & 0 & -0.719 & -0.672 & -0.641 \\ 0.656 & 0.719 & 0.781 & 0 & -0.781 & -0.719 & -0.656 \\ 0.688 & 0.781 & 0.875 & 0 & -0.875 & -0.781 & -0.688 \\ 0.703 & 0.813 & 1.00 & 0 & -1.00 & -0.813 & -0.703 \\ 0.688 & 0.781 & 0.875 & 0 & -0.875 & -0.781 & -0.688 \\ 0.656 & 0.719 & 0.781 & 0 & -0.781 & -0.719 & -0.656 \\ 0.641 & 0.672 & 0.719 & 0 & -0.719 & -0.672 & -0.641 \end{bmatrix}$$

$$7 \times 7 \quad \text{SNR}=9 \quad \begin{bmatrix} 0.073 & 0.140 & 0.283 & 0 & -0.283 & -0.140 & -0.073 \\ 0.104 & 0.213 & 0.348 & 0 & -0.348 & -0.213 & -0.104 \\ 0.165 & 0.354 & 0.579 & 0 & -0.579 & -0.354 & -0.165 \\ 0.195 & 0.463 & 1.00 & 0 & -1.00 & -0.463 & -0.195 \\ 0.165 & 0.354 & 0.579 & 0 & -0.579 & -0.354 & -0.165 \\ 0.104 & 0.213 & 0.348 & 0 & -0.348 & -0.213 & -0.104 \\ 0.073 & 0.140 & 0.283 & 0 & -0.283 & -0.140 & -0.073 \end{bmatrix}$$



### C.1.5 การเปรียบเทียบประสิทธิภาพของ ตัวตรวจหาขอบ (Edge detection operator )

ได้มีการทดลองวัดประสิทธิภาพของ Operator แบบต่างๆ พบว่าในภาพที่ไม่มี Noise แต่ละ Operator จะให้ภาพที่เกือบเหมือนกันหมด แต่สำหรับภาพที่มี Noise พบว่า Stochastic มีประสิทธิภาพดีกว่า โดยค่าที่ใช้ในการวัดประสิทธิภาพในแง่ของ Noise ซึ่งปริมาณดังกล่าว คือ edge detection error  $p_e$  โดย  $p_e = \frac{n_1}{n_0}$  เมื่อ  $n_0$  คือ จำนวนจุด pixel ของ edge ในกรณีที่ไม่มี noise และ  $n_1$  คือ จำนวนจุด pixel ของ edge ที่เพิ่มขึ้นหรือหายไป ในกรณีที่เพิ่ม noise แล้ว

### C.1.6 กระบวนการหาขอบของแคนนี่ (Canny 's Edge Detection)

เป็นกระบวนการหาขอบที่มีประสิทธิภาพสูงมาก ถูกพัฒนาโดย John Canny ในปี 1986 ซึ่ง แคนนี่ได้นิยามเป้าหมายสามอย่างที่ การหาขอบจะต้องมี คือ

1. การตรวจหาขอบจะต้องตอบสนองกับขอบเท่านั้น และ ต้องหาขอบได้ทั้งหมด
2. ระยะทางระหว่างจุดขอบที่ได้จากการตรวจหาขอบ กับ จุดขอบแท้จริงของวัตถุจะต้องเล็กที่สุด
3. การตรวจหาขอบต้องไม่มีการตรวจพบขอบแบบหลายพิกเซล กล่าวคือ จุดขอบจะต้องเป็นเส้นขนาด 1 พิกเซลเท่านั้น

แคนนี่ได้สมมุติ ขอบของรูปที่มี White Gaussian Noise ซึ่ง Edge detector ถูกสมมุติว่าเป็นฟิลเตอร์แบบคอนโวลูต (f) เพื่อให้ Noise เรียบขึ้นและหาขอบได้

ในทาง 1 มิติ ผลการตอบสนองของ ฟิลเตอร์ f ต่อขอบ G เขียนในรูปสมการได้ดังนี้

$$H = \int_{-w}^w G(-x)f(x)dx$$

โดยสมมุติว่าฟิลเตอร์เป็นศูนย์ นอกขอบเขต [-w,w] มีเกณฑ์ criteria 3 เกณฑ์ คือ

1.  $SNR = \frac{A \left| \int_{-w}^0 f(x)dx \right|}{n_0 \sqrt{\int_{-w}^w f^2(x)dx}}$
2.  $Localization = \frac{A|f(0)|}{n_0 \sqrt{\int_{-w}^w f^2(x)dx}}$

$$3. \quad x_{zc} = \pi \left( \frac{\int_{-\infty}^{\infty} f^2(x) dx}{\int_{-\infty}^{\infty} f'^2(x) dx} \right)^{\frac{1}{2}}$$

ค่าของ SNR ต้องการค่าที่มากที่สุดเท่าที่จะทำได้ เพื่อให้ได้ค่า signal มากและ noise น้อยๆ โดยค่า localization แสดงการวนกลับของระยะขอบที่ตรวจพบเทียบกับขอบจริง จึงควรจะมีค่ามากที่สุดซึ่งหมายถึงระยะความห่างจะน้อยที่สุดนั่นเอง

ค่าของ  $X_{zc}$  เป็นค่าที่จำกัด ซึ่งแสดงถึงระยะเฉลี่ยระหว่างค่า Zero crossing ของ  $f'$  เป็นสัจเจเมนต์ที่แสดงว่า Edge detector  $f'$  จะไม่ทำการหาขอบวนไปวนมากับที่ในบริเวณเล็กๆ

แดนนี้พยายามที่จะหาค่าฟิลเตอร์ที่ทำให้ค่าที่จะทำให้เกิด SNR x localization มากที่สุด ซึ่งในหารแก้ปัญหาที่มีการคำนวณที่ยุ่งยากและซับซ้อนมาก วิธีการประมาณที่ให้ผลดีอีกตัวหนึ่งที่น่ามาใช้คือ Gaussian function มีรูปสมการคือ

$$G(x) = e^{-\frac{x^2}{2\sigma^2}}$$

และอนุพันธ์ของ  $x$  มีค่า

$$G'(x) = \left( -\frac{x}{\sigma^2} \right) e^{-\left( \frac{x^2}{2\sigma^2} \right)}$$

ค่า Gaussian 2 มิติมีค่าดังนี้คือ

$$G(x, y) = \sigma^2 e^{-\left( \frac{x^2 + y^2}{2\sigma^2} \right)}$$

โดยที่ค่า  $G$  มีอนุพันธ์ทั้งทิศทาง  $x$  และ  $y$  ค่าการประมาณของฟิลเตอร์ของแดนนี้คือค่า  $G'$

ในการหาค่า Edge detection โดย 2 ทางพร้อมกันนั้น การคำนวณจะซับซ้อน ทำให้ใช้เวลาในการคำนวณมาก ซึ่งสามารถแก้ปัญหาได้โดยการทำ Gaussian ที่ละ 1 มิติและทำการหาอนุพันธ์ จากนั้นจะได้ภาพออกมา 1 ภาพ หลังจากนั้นทำการหาอีก 1 ทางแล้วนำภาพทั้ง 2 มาประมวลผลรวมกันซึ่งมี algorithm ดังนี้

1. รับภาพเข้ามา กำหนดให้เรียกว่า  $I$
2. สร้าง Mask ของ Gaussian 1 มิติ เรียกว่า  $G$  เพื่อนำไปประมวลผลกับภาพ  $I$
3. สร้าง Mask ของ อนุพันธ์อันดับที่ 1 ในทิศทาง  $x$  และ  $y$  เรียกว่า  $G_x$  และ  $G_y$
4. นำภาพ  $I$  มาประมวลผลกับ Mask  $G$  ตามแนว  $x$  และตามคอลัมน์  $y$  ได้  $I_x$  และ  $I_y$  ตามลำดับ
5. นำภาพ  $I_x$  และ  $I_y$  มาคอนโวลูตกับ  $G_x$  และ  $G_y$  ตามลำดับ ได้  $I'_x$  และ  $I'_y$
6. นำผลการคำนวณมารวมกันจะได้ magnitude ดังนี้

$$M(x, y) = \sqrt{I'_x(x, y)^2 + I'_y(x, y)^2}$$

ค่า  $M$  จะมีค่ามากสำหรับจุดที่เป็นขอบ และ จะมีค่าน้อยสำหรับจุดที่ไม่ใช่ขอบ สิ่งที่จะต้องทำต่อไปคือการ Threshold ตามทิศทางเกรเดียนท์ของแต่ละพิกเซล ขั้นตอนสุดท้ายคือการกำจัดจุดที่ไม่ใช่จุดที่มีค่ามากที่สุดในบริเวณหนึ่งๆ ตามทิศทางของเกรเดียนท์ (Nonmaximum suppression) ซึ่งทำโดยการตรวจสอบค่าขนาดเกรเดียนท์ โดยการวิ่งตามทิศทางเกรเดียนท์

ในกรณีที่ทิศทางเกรเดียนท์ ตั้งฉากกับขอบ เมื่อทำการวิ่งตามทิศทางเกรเดียนท์จะสามารถกำจัดจุดที่มีค่าเกรเดียนท์ไม่สูงสุด และ แยกขอบเขตได้อย่างชัดเจน แต่กรณีเช่นนี้เกิดขึ้นน้อยมาก ปกติการวิ่งตามทิศทางเกรเดียนท์มักจะไปบรรจบที่บริเวณระหว่างจุดพิกเซลสองจุด ในกรณีเช่นนี้เราจะสมมุติว่าเกรเดียนท์เป็นฟังก์ชันต่อเนื่องตามตำแหน่ง จากนั้นจะทำการประมาณค่าในช่วงแบบเชิงเส้น (Linear interpolation) กับค่าเกรเดียนท์ระหว่างจุดสองจุดนั้นในการตรวจสอบค่าขนาดเกรเดียนท์ตามทิศทางเกรเดียนท์

ภาพสุดท้ายที่ได้ในขั้นนี้ยังคงเป็นภาพเกรย์อยู่ และ จำเป็นจะต้องทำการตัดค่า (Threshold) เพื่อให้เหลือแต่ขอบของภาพจริงๆ ในขั้นนี้เราได้แนะนำวิธีการพิเศษในการ Threshold ที่ดีกว่าการตัดค่า Threshold ง่ายๆธรรมดา นั่นคือการตัดค่าแบบฮิสเทอรีซิส ซึ่งอาศัยค่า Threshold สองค่าคือ ค่า Higher Threshold  $TH$  และ ค่า Lower Threshold  $TL$  และมีการบันทึกคะแนนให้แก่แต่ละจุด โดยเมื่อจุดใดมีค่ามากกว่าค่า  $TH$  จะถูกกำหนดว่ามีโอกาสเป็นขอบ จุดนั้นก็ถูกบันทึกคะแนนหนึ่งครั้ง และ จุดรอบๆ จุดนั้นทั้งหมดที่มีค่าสูงกว่าค่า  $TL$  จะถูกกำหนดว่าเป็นขอบด้วยเช่นกัน และ จะได้หนึ่งคะแนน การให้คะแนนจะถูกทำซ้ำแก่ทุกจุดทั้งภาพ สุดท้ายจุดที่มีคะแนนสูงกว่าค่าหนึ่งจึงจะถูกกำหนดว่าเป็นขอบ

## C.2 กระบวนการประมวลผลสำหรับภาพไบนารี (Binary Image Processing)

ภาพไบนารี คือ ภาพที่แต่ละพิกเซลเป็นได้เพียงสองสถานะคือ 1 หรือ 0 เท่านั้น โดย 1 แทนบริเวณวัตถุ และ 0 แทนพื้นหลัง ภาพไบนารีอาจได้มาจากการ Threshold ภาพ Gray - level ด้วย ค่าความเข้มสีแต่ละจุดด้วยค่า Threshold หนึ่งๆ

เนื่องจากแต่ละจุดพิกเซลของภาพไบนารีมีสถานะได้เพียง 1 หรือ 0 ทำให้ภาพไบนารีมีความง่ายในการประมวลผล และสามารถใช้กระบวนการทางลอจิก เช่น การ AND OR หรือ XOR ๗ กับภาพได้ กระบวนการสำหรับภาพไบนารีที่น่าสนใจได้แก่

### C.2.1 การหาคุณสมบัติทางเรขาคณิตอย่างง่าย

#### 1. การคำนวณพื้นที่ และ ตำแหน่งของวัตถุ (Area and Position)

นิยามฟังก์ชันสถานะ  $b(x,y) = 1$ , เมื่อ  $(x,y)$  อยู่ในบริเวณวัตถุ  
 $0$ , เมื่อ  $(x,y)$  อยู่ในบริเวณพื้นหลัง

สามารถหาพื้นที่ของวัตถุได้จากสมการ

$$A = \sum \sum_{Image} b(x, y),$$

โดยปกติมักใช้จุดเซนทรอยด์ แทนตำแหน่งของวัตถุ ตำแหน่งเซนทรอยด์ หาได้จาก

$$\bar{x} \sum \sum_{Image} b(x, y) = \sum \sum_{Image} x * b(x, y)$$

$$\bar{y} \sum \sum_{Image} b(x, y) = \sum \sum_{Image} y * b(x, y)$$

#### 2. การหาทิศทางของแนวการวางตัวของวัตถุ (Orientation)

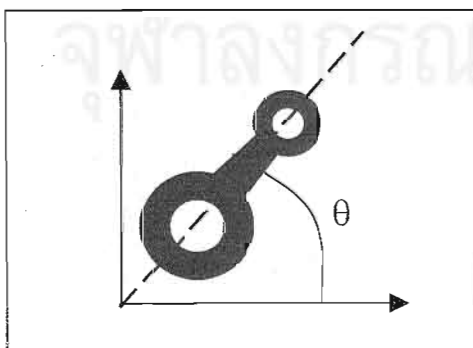


Fig C.7

แนวการวางตัวของวัตถุ นิยามโดยใช้แกนที่ผ่านจุดเซนทรอยด์ที่มีโมเมนต์อันดับสองของพื้นที่น้อยที่สุดตามสมการ

$$E = \sum \sum_{Image} r^2 * b(x, y),$$

โดยที่  $r$  คือระยะทางจากจุดเซนทรอยด์ถึงจุด  $(x, y)$

โดยทางคณิตศาสตร์สามารถพิสูจน์ได้ว่า แกนดังกล่าวเป็นเส้นตรงผ่านจุดเซนทรอยด์ ที่ทำมุม  $\theta$  กับแกน  $x$  โดยที่

$$\tan 2\theta = \frac{b}{a - c}$$

เมื่อ 
$$a = \sum_{\text{Image}} \sum x'^2 b(x, y), \quad b = \sum_{\text{Image}} \sum x' y' b(x, y), \quad c = \sum_{\text{Image}} \sum y'^2 b(x, y)$$

$$X' = X - \bar{X}, \quad Y' = Y - \bar{Y} \quad \text{โดยที่}$$

สมการ ดังกล่าวจะให้คำตอบเป็นค่ามุมสองค่า ค่าหนึ่งจะเป็นมุมของแนวแกนที่มีค่าโมเมนต์น้อยที่สุด อีกมุมหนึ่งเป็นของแนวแกนที่มีค่าโมเมนต์มากที่สุด ซึ่งแกนทั้งสองตั้งฉากกัน

### C.2.2 ค่าโมเมนต์ที่ไม่ผันแปรของวัตถุ (Moment Invariants)

โมเมนต์อันดับต่างๆของวัตถุในภาพไบนารีนิยามโดย

$$\mu_{p,q} = \sum_{\text{Image}} \sum b(x, y) (x - \bar{x})^p (y - \bar{y})^q$$

โดยการพิสูจน์ทางคณิตศาสตร์ เราสามารถเขียนฟังก์ชันโพลีโนเมียล ซึ่งขึ้นกับรูปร่างของวัตถุเท่านั้น แต่มีค่าไม่ผันแปรต่อ การเลื่อนตำแหน่ง การย่อขยาย และการหมุนเอียงของวัตถุซึ่งเรียกว่า ฟังก์ชันโมเมนต์ที่ไม่ผันแปรของวัตถุ

นิยามนอร์มัลไลซ์โมเมนต์

$$\eta_{p,q} = \frac{\mu_{p,q}}{\mu_{0,0}^\gamma}, \quad \gamma = (p + q + 2) / 2$$

เมื่อพิจารณาเฉพาะโมเมนต์อันดับสอง และ สาม จะได้ฟังก์ชัน โมเมนต์ไม่ผันแปรของวัตถุทั้งหมด 7 ฟังก์ชันคือ

$$\text{Second-order} \quad \phi_1 = \eta_{2,0} + \eta_{0,2}$$

$$\phi_2 = (\eta_{2,0} - \eta_{0,2})^2 + 4\eta_{1,1}^2$$

Third-order

$$\phi_3 = (\eta_{3,0} - 3\eta_{1,2})^2 + (\eta_{0,3} - 3\eta_{2,1})^2$$

$$\phi_4 = (\eta_{3,0} + \eta_{1,2})^2 + (\eta_{0,3} + \eta_{2,1})^2$$

$$\phi_5 = (\eta_{3,0} - 3\eta_{1,2})(\eta_{3,0} + \eta_{1,2})[(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{2,1} + \eta_{0,3})^2] \\ + (\eta_{0,3} - 3\eta_{2,1})(\eta_{0,3} + \eta_{2,1})[(\eta_{0,3} + \eta_{2,1})^2 - 3(\eta_{1,2} + \eta_{3,0})^2]$$

$$\phi_6 = (\eta_{2,0} - \eta_{0,2})[(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2] \\ + 4\eta_{1,1}(\eta_{3,0} + \eta_{1,2})(\eta_{0,3} + \eta_{2,1})$$

$$\phi_7 = (3\eta_{2,1} - \eta_{0,3})(\eta_{3,0} + \eta_{1,2})[(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{2,1} + \eta_{0,3})^2] \\ + (\eta_{3,0} - 3\eta_{2,1})(\eta_{2,1} + \eta_{0,3})[(\eta_{0,3} + \eta_{2,1})^2 - 3(\eta_{1,2} + \eta_{3,0})^2]$$

ตัวอย่าง การคำนวณค่าฟังก์ชันโมเมนต์ไม่ผันแปร

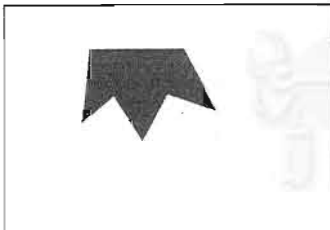


Fig C.8

จากรูปด้านซ้ายมือ สามารถคำนวณค่าฟังก์ชันทั้งเจ็ด ได้ดังต่อไปนี้

$$\phi_1=0.208159, \phi_2=0.009676, \phi_3=0.000186, \phi_4=0.0000598, \phi_5=- \\ 0.000000028, \phi_6=0.0000010332, \phi_7=0.0000000052$$

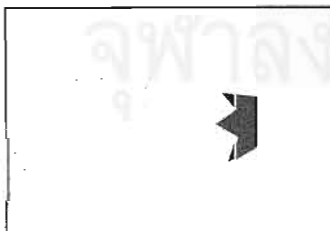


Fig C.9

เมื่อทำการหมุน-ย่อ-เลื่อนตำแหน่งภาพดังรูป ฟังก์ชันทั้งเจ็ดจะมีค่าดังนี้

$$\phi_1=0.208770, \phi_2=0.009896, \phi_3=0.000187, \phi_4=0.0000604, \\ \phi_5=-0.000000028, \phi_6=0.0000010612, \phi_7=0.0000000053$$

จะเห็นว่าทั้งสองกรณี แต่ละฟังก์ชันมีค่าใกล้เคียงกันมาก ความคลาดเคลื่อนเล็กน้อยนั้นเกิดจากการประมาณค่าอินทิกรัล ด้วยผลรวมพีชคณิต (Summation) ฟังก์ชันโมเมนต์ไม่ผันแปรนี้มีประสิทธิภาพอย่างมาก เมื่อใช้ในการตรวจหาวัตถุ สองมิติ

C.2.3 โครงสร้างของวัตถุ และ การแปลงวัตถุ (Structure and Morphological Transform)

ในบางครั้งรายละเอียดสำคัญของวัตถุไม่ขึ้นอยู่กับความหนาเส้นของวัตถุ เช่น ภาพลายนิ้วมือ หรือ แผงวงจร กรณีเช่นนี้เราอาจแทนวัตถุด้วยโครงสร้างของมันได้ โครงสร้างชนิดหนึ่งคือ แกนกลางของวัตถุ (medial axis) ซึ่งเราสามารถหาได้ 2 วิธีคือ

1). การหาโครงร่างของวัตถุ (Skeleton algorithms) โครงร่างของวัตถุ คือ จุดที่อยู่ไกลจากขอบที่ใกล้ที่สุด เป็นระยะทางมากที่สุด เมื่อเทียบกับจุดอื่นๆ ในบริเวณนั้น ซึ่งสามารถหาได้โดยอัลกอริทึมดังนี้

$$u_k(m,n) = u_0(m,n) + \min_{\Delta(m,n;i,j)} \{u_{k-1}(i,j); ((i,j): \Delta(m,n;i,j) \leq 1)\}$$

$$u_0(m,n) = u(m,n) \quad k = 1, 2, \dots$$

ซึ่ง  $\Delta(m,n;i,j)$  เป็นระยะทางระหว่าง  $(m,n)$  และ  $(i,j)$

และ Skeleton เป็นเซตของจุด  $\{(m,n): u_k(m,n) \geq u_k(i,j), \Delta(m,n;i,j) \leq 1\}$

การหาโครงร่างของวัตถุด้วยอัลกอริทึมดังกล่าวสามารถเข้าใจได้ง่าย โดยดูจากรูปตัวอย่างต่อไปนี้

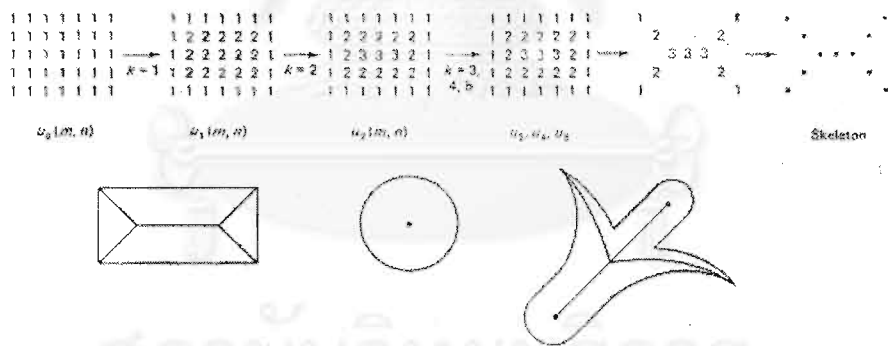


Figure 9.31 Skeleton examples.

Fig C.10

2) การทำให้วัตถุบางลง (Thinning algorithm) เป็นการทำให้วัตถุในภาพมีความหนาน้อยลง ซึ่งจะให้ผลลัพธ์ใกล้เคียงกับการหาโครงร่างของวัตถุ การพิจารณาลบจุดเพื่อให้วัตถุบางลง โดยมีหลักการดังนี้

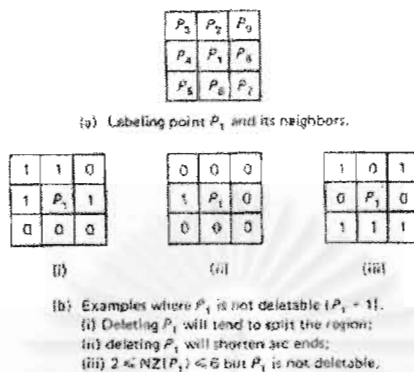


Fig C.11

จากรูป กำหนดให้  $Z0(P_1)$  เป็นจำนวนครั้งการเปลี่ยนค่าจาก 0 เป็นค่าที่ไม่ใช่ 0 ในลำดับเซต  $P_2, P_3, \dots, P_9, P_2$   $NZ(P_1)$  เป็นจำนวนของ จุดรอบๆ  $P_1$  ที่ไม่ใช่ 0 จุด  $P_1$  จะถูกลบในกรณีที่

$$2 \leq NZ(P_1) \leq 6$$

และ  $Z0(P_1) = 1$

และ  $P_2 \cdot P_4 \cdot P_8 = 0$  or  $Z0(P_2) \neq 1$

และ  $P_2 \cdot P_4 \cdot P_6 = 0$  or  $Z0(P_4) \neq 1$

ผลจากการลบจุดจะทำให้วัตถุบางลงดังแสดงในรูปตัวอย่างต่อไปนี้



(c) Example of thinning.  
 (i) Original.  
 (ii) thinned.

Fig C.12

3) กระบวนการทางตรรกพื้นฐาน (Basic Operation) ในการแปลงรูปภาพไบนารี จะมีกระบวนการทางตรรกพื้นฐานเพิ่มเติมอีกสองกระบวนการ คือ

(1). Erosion  $\ominus$  : นิยามด้วย  $X \ominus B = \{x : B_x \subset x\}$



ซึ่งเป็นการลบจุดใน Object ที่เมื่อนำ Structure element ไปทาบบแล้ว Structure element ไม่ได้อยู่ใน Object ทั้งหมด

$$(2). \text{Dilation } \oplus : \text{นิยามด้วย } X \oplus B = \{x : B_x \cap X \neq \Phi\}$$

ซึ่งเป็นการหาจุดที่เมื่อนำ Structure element ไปทาบบแล้ว มีอย่างน้อยหนึ่งจุดของ Structure element อยู่ใน Object ทั้งนี้ถ้าจุดเดิมใน Object ที่ถูกทาบบด้วย Structure element แล้วไม่มีจุดใดของ Structure element ใน Object เลย จุดนั้นก็จะถูกลบไปด้วย

กระบวนการทั้งสองมีคุณสมบัติ และ ข้อกำหนดต่างๆ ดังนี้

1. การเลื่อนตำแหน่งของวัตถุ ไม่มีผลต่อการทำ Erosion และ Dilation
2. Erosion และ Dilation ไม่ใช่ อินเวอร์ส ของกันและกัน
3. คุณสมบัติ Distributivity :  $X \oplus (B \cup B') = (X \oplus B) \cup (X \oplus B')$   
 $X \ominus (B \cup B') = (X \ominus B) \cap (X \ominus B')$
4. คุณสมบัติ Local knowledge :  $(X \cap Z) \ominus B = (X \ominus B) \cap (Z \ominus B)$
5. Iteration :  $(X \ominus B) \ominus B' = X \ominus (B \oplus B')$   
 $(X \oplus B) \oplus B' = X \oplus (B \ominus B')$
6. Increasing :  $\left. \begin{array}{l} \text{ถ้า } X \subset X' \Rightarrow X \ominus B \subset X' \ominus B \quad \forall B \\ X \oplus B \subset X' \oplus B \quad \forall B \end{array} \right\}$   
 $\left. \begin{array}{l} \text{ถ้า } B \subset B' \Rightarrow X \ominus B \subset X \ominus B' \quad \forall X \end{array} \right\}$
7. Duality : ให้  $X^c$  คือ ส่วนเติมเต็มของ  $X$  จะได้  $X^c \oplus B = (X \ominus B)^c$

4) กระบวนการแปลงวัตถุด้วยกรรมวิธีทางตรรก (Morphological Transforms) นอกจากกระบวนการหาโครงร่าง และ การทำให้วัตถุบางแล้ว ยังมีกระบวนการแปลง Object แบบต่างๆโดยใช้ กรรมวิธีทางตรรก (คือ  $\cup, \cap, /$  (set difference),  $\oplus, \ominus$ ) กับรูปภาพไบนารี ซึ่งทั้งกระบวนการ, คำนิยาม, คุณสมบัติและ การใช้งาน รวมทั้ง Structure element สามารถดูได้จากตารางที่ 3.3

จุฬาลง

TABLE 9.10 Some Useful Morphological Transforms

Operation	Definition	Properties & Usage
Hit-Miss	$X \odot B = (X \ominus B_{ob}) / (X \oplus B_{bk})$	Searching for a match or a specific configuration. $B_{ob}$ : set formed from pixels in $B$ that should belong to the object. $B_{bk}$ : ... background.
Open	$X_B = (X \ominus B) \oplus B$	Smooths contours, suppress small islands and sharp caps of $X$ . Ideal for object size distribution study.
Close	$X^B = (X \oplus B) \ominus B$	Blocks up narrow channels and thin lakes. Ideal for the study of inter object distance.
Boundary	$\partial X = X / X \ominus G$	Gives the set of boundary points.
Convex Hull	$X_1 = X$ $X_{i+1} = (X_i \oplus B^i)$ $X_{CH} = \bigcup_{j=1}^n X_j$	$B^1, B^2, \dots$ are rotated versions of the structuring element $B$ . $C$ is an appropriate structuring element choice for $B$ .
Skeleton	$S(X) = \bigcup_{n=0}^{n_{max}} s_n(x)$ $= \bigcup_{n=0}^{n_{max}} [(X \ominus nG) / (X \ominus nG)_G]$ $X = \bigcup_{n=0}^{n_{max}} [s_n(x) \oplus nG]$	$n_{max}$ : max size after which $X$ erodes down to an empty set. The skeleton is a regenerative representation of the object.
Thin	$X \circ B = X / X \odot B$ $X \circ \{B\} = ((\dots((X \circ B^1) \circ B^2) \dots) \circ B^n)$	To symmetrically thin $X$ a sequence of structuring elements, $\{B\} = \{B^i, 1 \leq i \leq n\}$ , is used in cascade, where $B^i$ is a rotated version of $B^{i-1}$ .
Thick	$X \circ B = X \cup X \odot B$	A widely used element is $L$ .
Prune	$X_1 = X \circ \{B\}$ $X_2 = \bigcup_{j=1}^n (X_1 \odot P^j)$ $X_{pn} = X_1 \cup [(X_2 \oplus \{G\}) \cap X]$	Dual of thinning. $E$ is a suitable structuring element. $X_2$ : end points. $X_{pn}$ : pruned object with Parasite branches suppressed.

The symbols “/” and “ $\cup$ ” represent the set difference and the set union operations respectively. Examples of structuring elements are

$$G = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & d & d \\ 1 & 0 & d \\ 1 & d & d \end{bmatrix}, \quad L = \begin{bmatrix} 0 & 0 & 0 \\ d & 1 & d \\ 1 & 1 & 1 \end{bmatrix}, \quad E = \begin{bmatrix} d & d & d \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

where 1, 0, and  $d$  signify the object, background and ‘don’t care’ states, respectively.

ตัวอย่างผลลัพธ์ที่ได้จากการแปลงแบบต่างๆ เป็นดังนี้

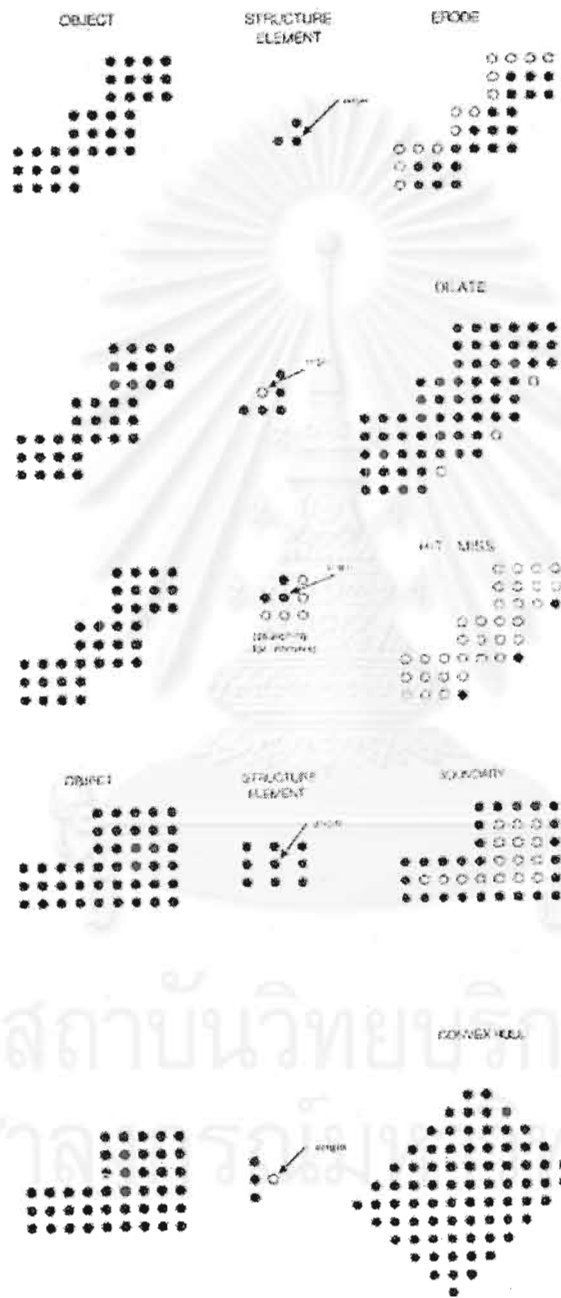


Figure 9.33 Examples of some morphological operations.

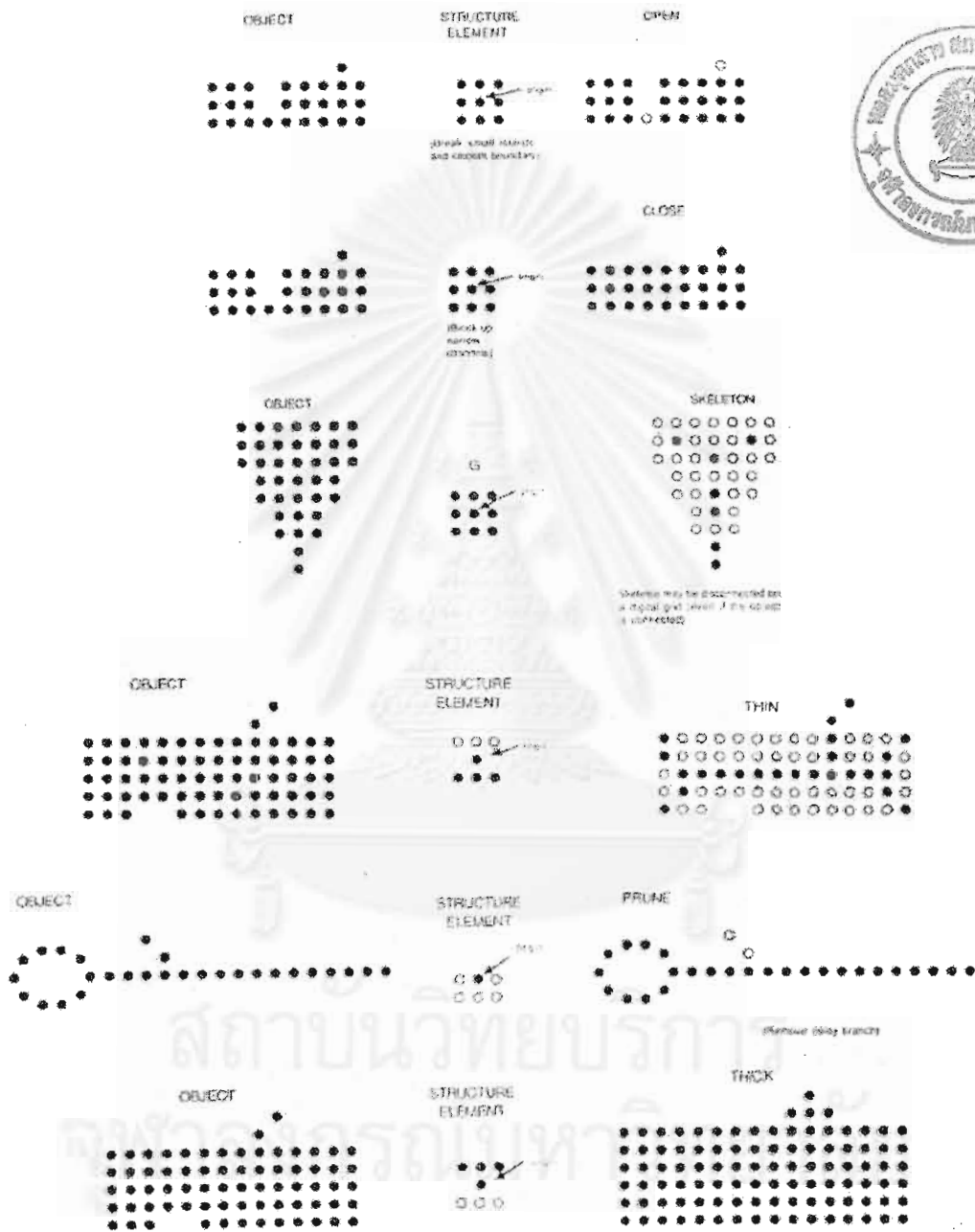
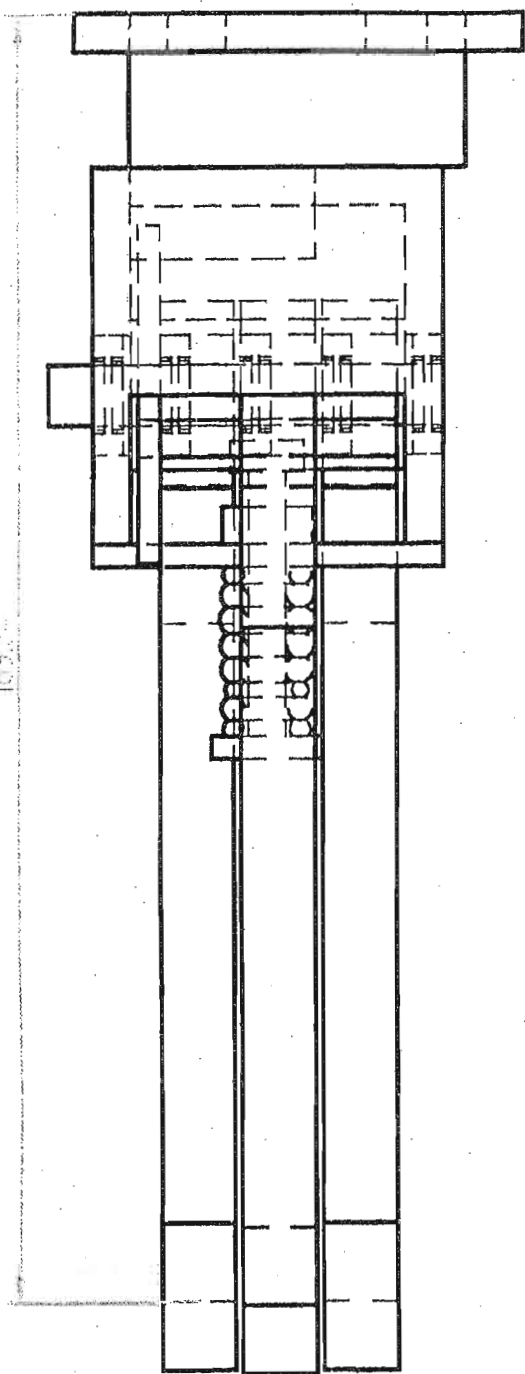
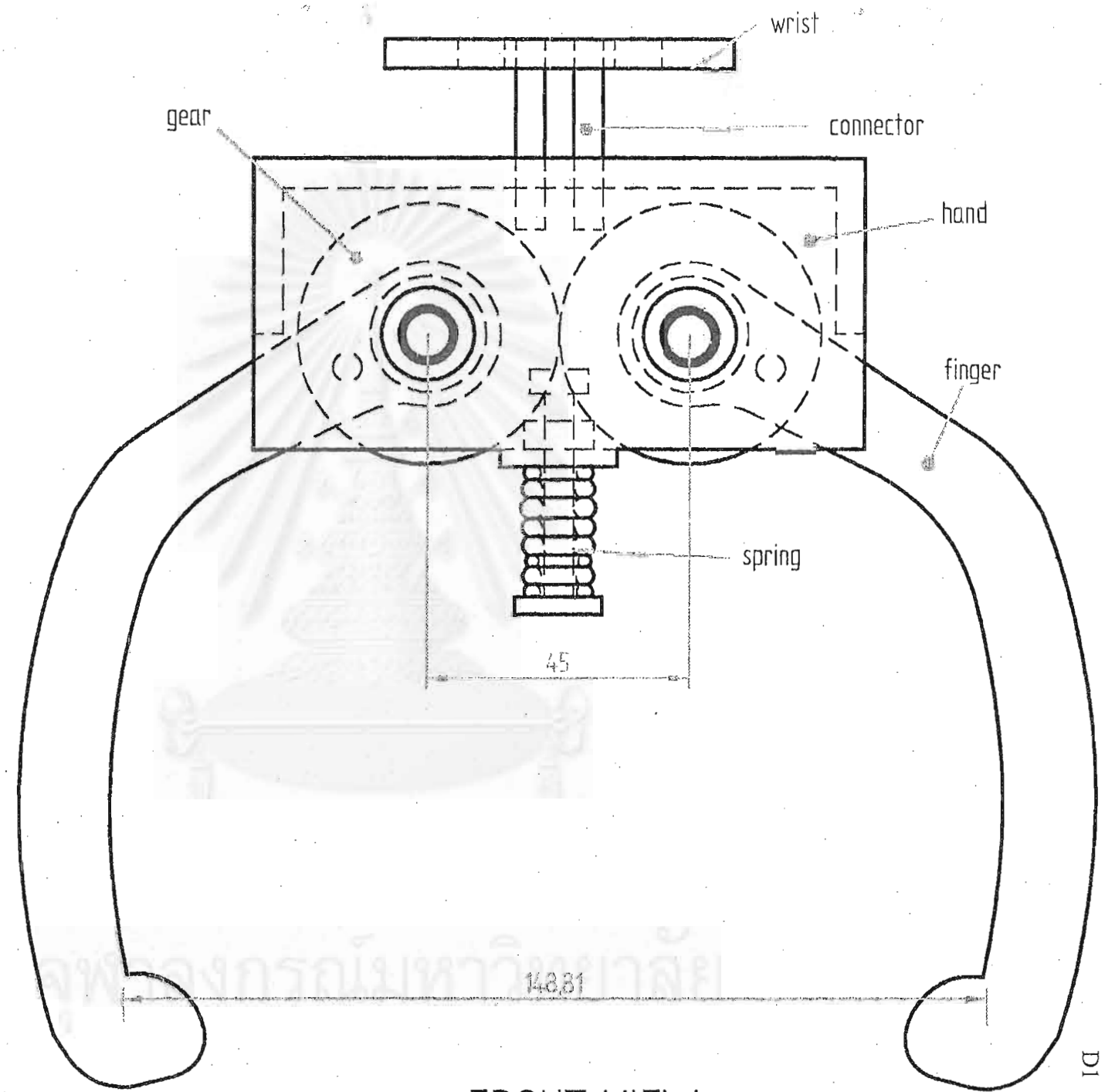


Figure 9.33 Cont'd.

Fig C.13 (ต่อ)



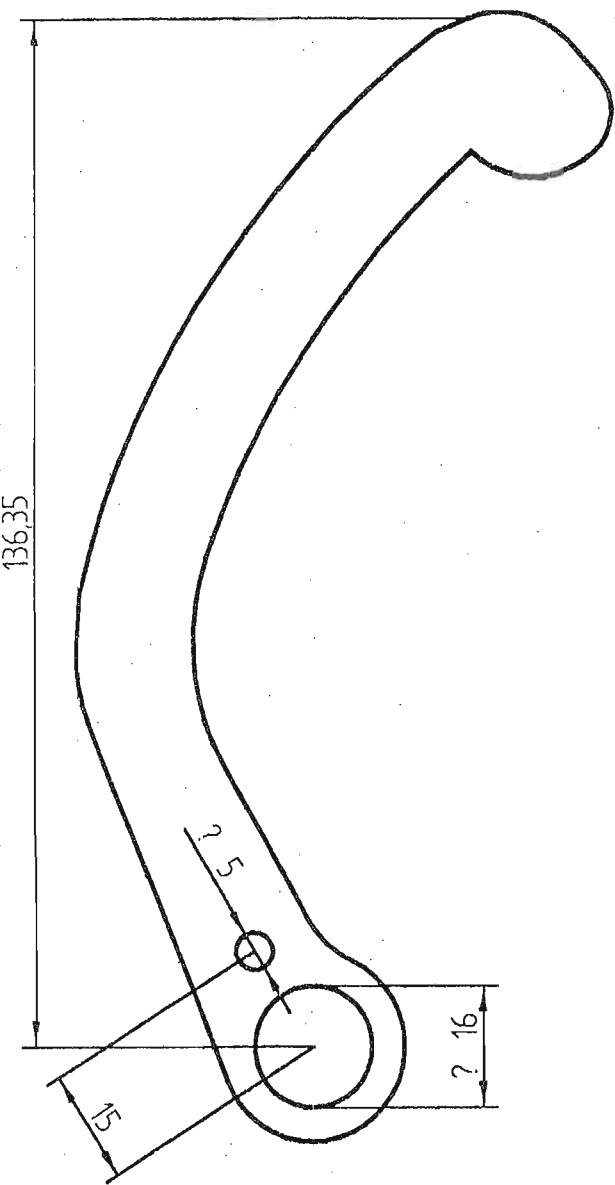
SIDE VIEW



FRONT VIEW

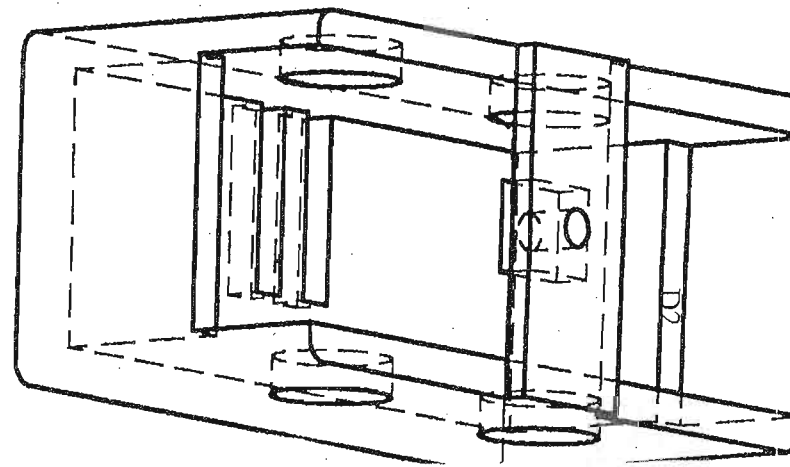
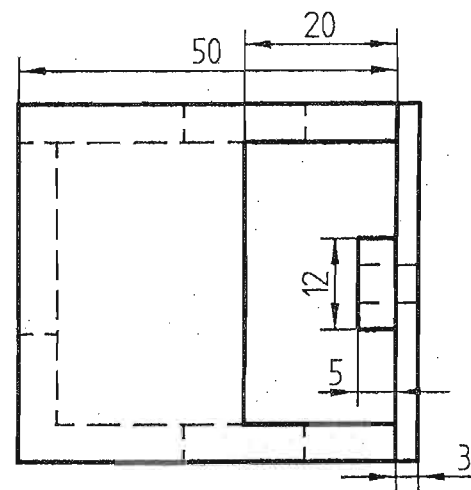
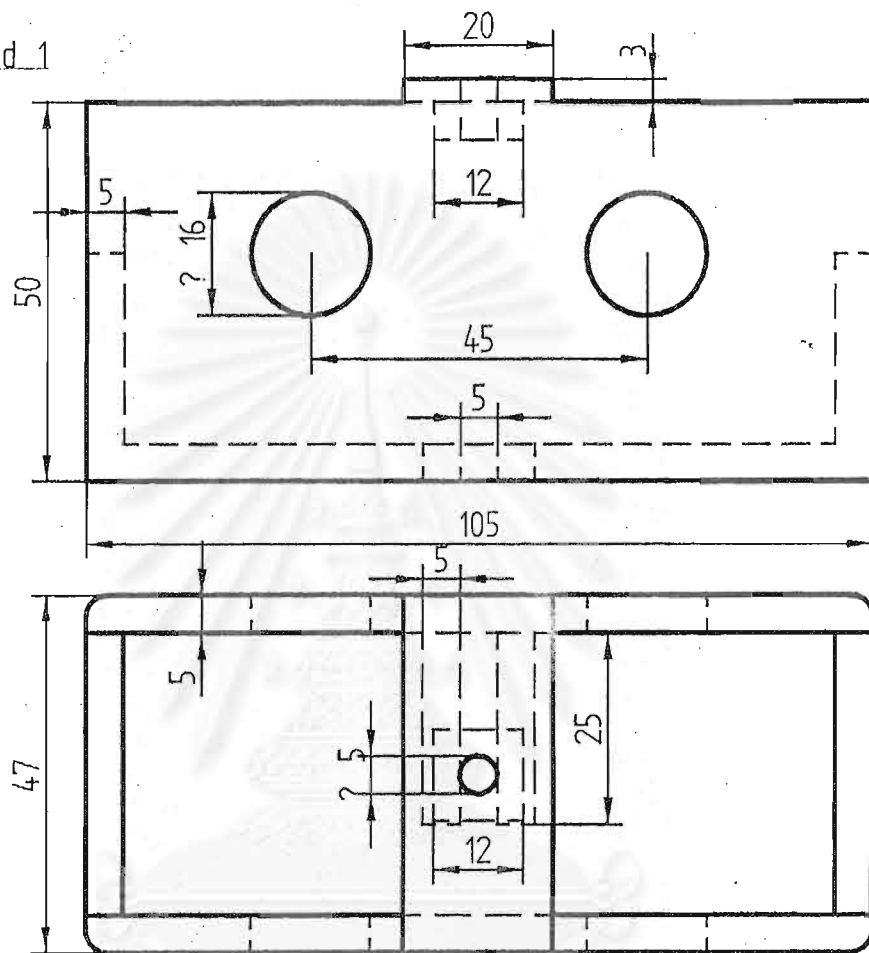
Scale 1:1

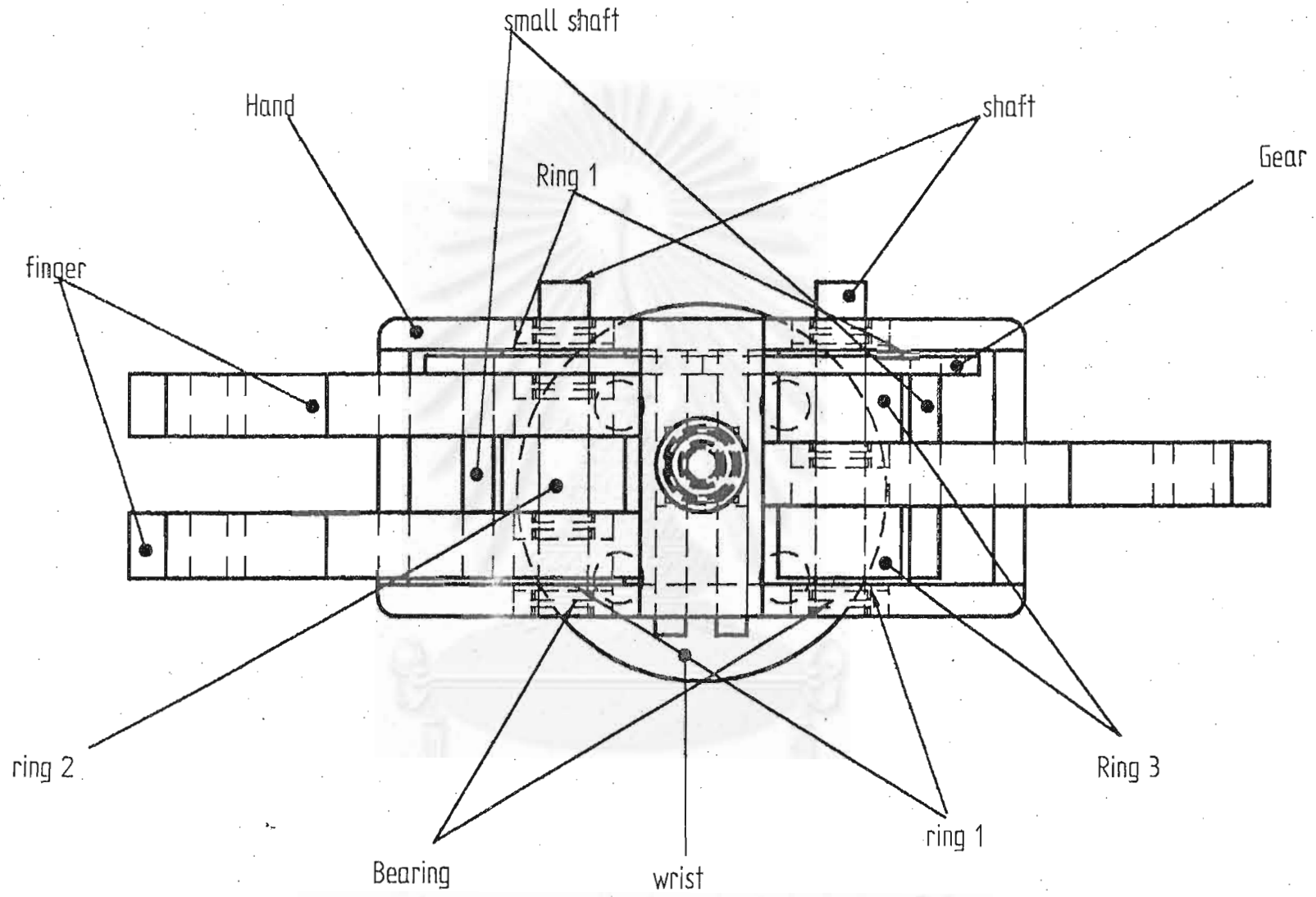
finger\_3



Scale 1:1

hand\_1

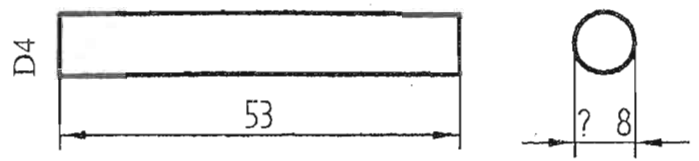




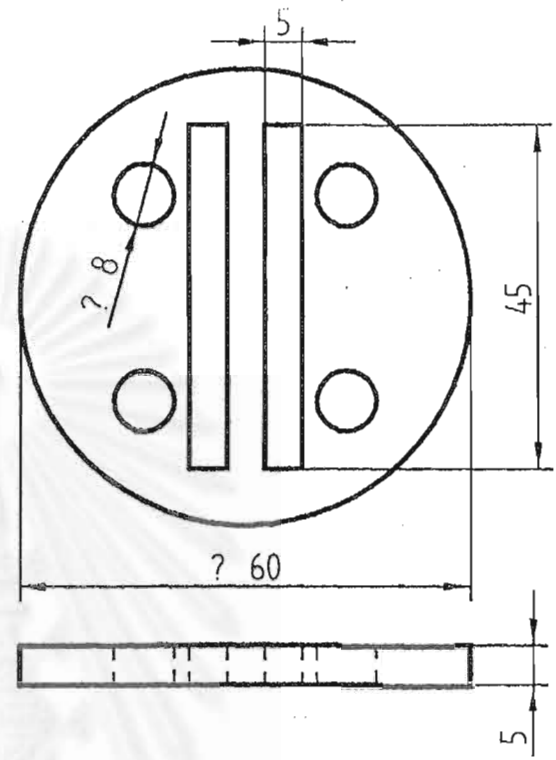
จุฬาลงกรณ์มหาวิทยาลัย

Scale 1:1

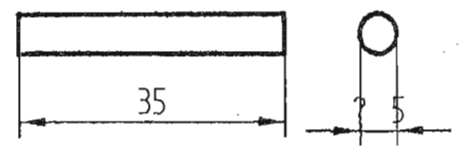
shaft 2



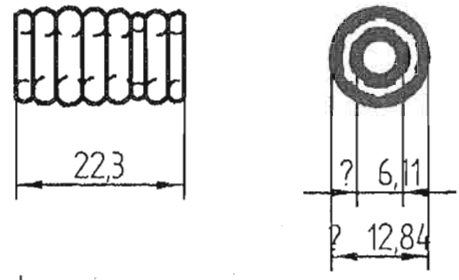
wrist 1



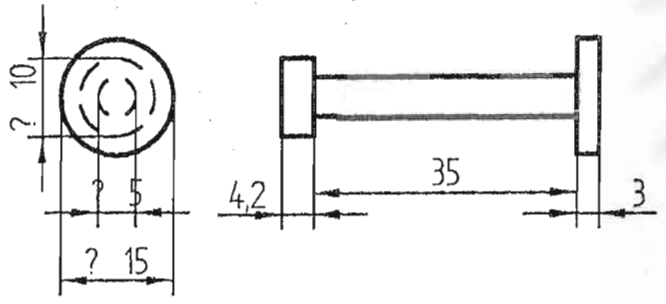
small shaft 2



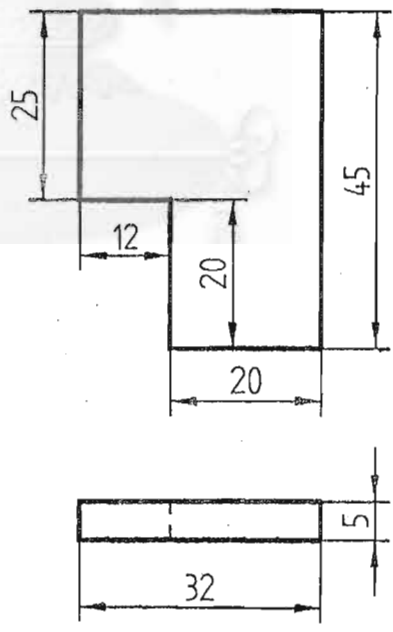
Spring 1



spring shaft 1

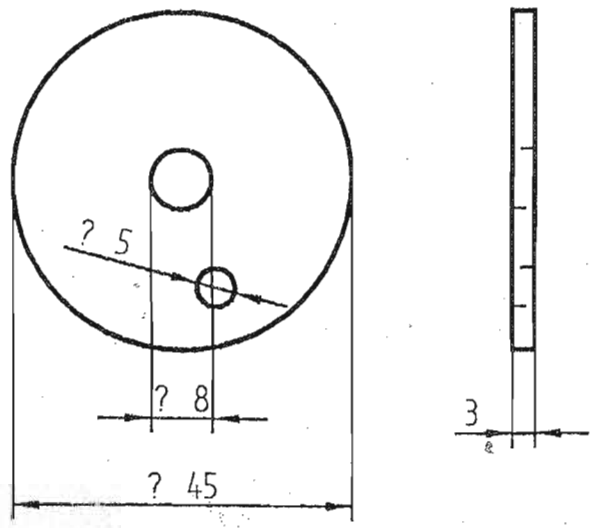


connector 2



- ring 1 dia(out) 20, dia(in) 8, thick 1, 4
- ring 2 dia(out) 20, dia(in) 8, thick 12, 1
- ring 3 dia(out) 20, dia(in) 8, thick 11, 2
- Bearing dia(out) 16, dia(in) 8, thick 4, 7

Gear 2



Scale 1:1