

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การมองแบบเวลาจริงสำหรับหุ่นยนต์เตะฟุตบอล

A REAL TIME VISUALIZATION FOR SOCCER ROBOTS



จตุรวิทย์ จันไพบูลย์
JATURAWIT JANPAIBOON

จพ.

จ 141 ก

๒๕๔๘

เลขหมู่.....
เลขทะเบียน..... 60508
ชั้น, เดือน, ปี - 3 ก.ค. 2549

b. 115 11880
i.

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2548

ISBN 974-15-1898-6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A REAL TIME VISUALIZATION FOR SOCCER ROBOTS



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN INFORMATION TECHNOLOGY ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2005

ISBN 974-15-1898-6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2005

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การมองแบบเวลาจริงสำหรับหุ่นยนต์เตะฟุตบอล
นักศึกษา	นายจตุรวิทย์ จันไพบูลย์
รหัสประจำตัว	44061816
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
พ.ศ.	2548
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ. ดร. ปิติเขต สุรักษา

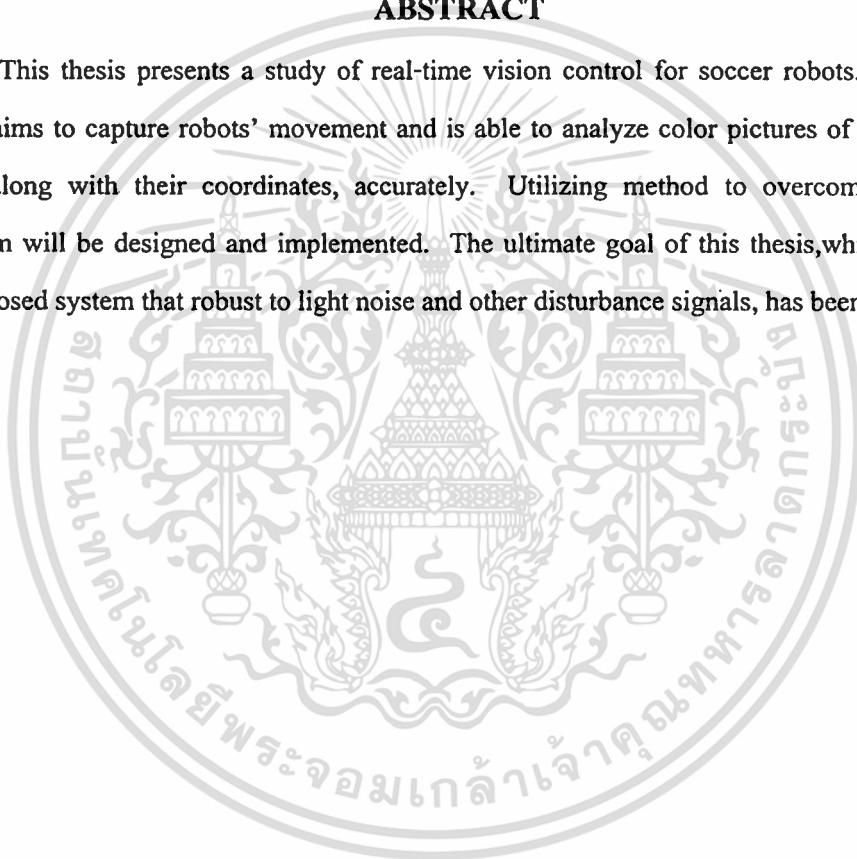
บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอการศึกษาเกี่ยวกับระบบการมองของหุ่นยนต์เตะฟุตบอล เพื่อที่จะจับภาพและหาตำแหน่งของวัตถุที่มีการเคลื่อนไหว ในการที่จะสามารถจับการเคลื่อนไหวของลูกกอล์ฟ 1 ลูก กับ หุ่นยนต์อีก 10 ตัวได้อย่างแม่นยำนั้น จะต้องมีการวิเคราะห์สีของภาพที่ดี มีความทนทานต่อสัญญาณรบกวนต่าง ๆ ที่เกิดจาก ไม่ว่าจะเป็นที่เกิดจากกล้อง สายนำสัญญาณ ตัวแปลงสัญญาณอนาล็อกเป็นดิจิทัล หรืออาจจะเกิดจากแสงก็ตาม ที่สำคัญจะต้องประมวลผลในเวลาจริง (Real Time) ซึ่งจะนำมาใช้ควบคุมและแก้ไขค่าความผิดพลาดของสี การหาพิกัด การหามุมของตัวหุ่นยนต์ที่ทำกับสนาม และตัดสินใจในการนำเฉพาะภาพบางส่วนที่สำคัญมาประมวลผล

Thesis Title	A REAL TIME VISUALIZATION FOR SOCCER ROBOTS
Student	Mr. Jaturawit Janpaiboon
Student ID.	44061816
Degree	Master of Engineering
Programme	Information Technology Engineering
Year	2005
Thesis Adviser	Asst. Prof. Dr. Pitikhate Sooraksa

ABSTRACT

This thesis presents a study of real-time vision control for soccer robots. The control system aims to capture robots' movement and is able to analyze color pictures of a ball and ten robots along with their coordinates, accurately. Utilizing method to overcome errors, this algorithm will be designed and implemented. The ultimate goal of this thesis, which is to build the proposed system that robust to light noise and other disturbance signals, has been achieved.



กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีเพราะได้รับช่วยเหลือจากบุคคลต่าง ๆ โดยเฉพาะอย่างยิ่งได้รับความเมตตาจาก รศ. ดร. ปิติเชต สุรักษา ที่กรุณาให้คำชี้แนะและเป็นที่ปรึกษาแก่ผู้วิจัย ด้วยดีตลอดมา อีกทั้งยังเอื้อเฟื้ออุปกรณ์และเครื่องมือต่าง ๆ ในการทำวิทยานิพนธ์ ผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้ด้วย

ขอขอบคุณ อาจารย์บุญยชนะ ภูระหงษ์ และทีมงาน ที่ได้ช่วยเหลือหาอุปกรณ์ต่าง ๆ ในการทำวิทยานิพนธ์ ซึ่งอุปกรณ์บางอย่างไม่สามารถหาได้ในประเทศไทย หากปราศจากความช่วยเหลือนี้ การทดสอบในงานวิจัย คงเป็นไปได้ยาก

ขอขอบคุณ คณาจารย์ภาควิชาวิศวกรรมสารสนเทศทุกท่าน ที่ได้ให้คำแนะนำอันเป็นประโยชน์ต่อการทำวิทยานิพนธ์ พร้อมกันนี้ขอขอบคุณเพื่อน ๆ พี่ ๆ และน้อง ๆ ที่คอยให้ความช่วยเหลือและเป็นกำลังใจมาตลอด

และสุดท้ายที่จะลืมเสียไม่ได้คือ กราบขอบพระคุณ คุณพ่อ คุณแม่ อันเป็นที่รักยิ่ง ที่คอยอบรมเลี้ยงดูตั้งแต่เล็กจนสำเร็จในสิ่งที่มุ่งหวังไว้ ฉะนั้นคุณงามความดีใด ๆ ก็ตามที่เกิดจากวิทยานิพนธ์ฉบับนี้ขอมอบแด่ทุก ๆ ท่านที่ได้กล่าวมาแล้วข้างต้น

จตุรวิทย์ จันไพบูลย์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของงานวิจัย.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	1
1.3 ขอบเขตของงานวิจัย.....	2
1.4 ขั้นตอนของงานวิจัย.....	2
1.5 สมมติฐานเบื้องต้น.....	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	4
2.1 การแยกสี (Color Segmentation).....	4
2.1.1 ค่าสีที่ได้จากกล้อง.....	4
2.1.2 การใช้มิติสี HSV โดยตั้ง Min-Max Threshold ในการเลือกช่วงสี.....	5
2.1.3 การใช้มิติสี HSV โดยตั้ง Min-Tolerance Threshold ในการเลือกช่วงสี.....	6
2.1.4 การใช้มิติสี HSV ใน LUT (RGB to HSV).....	8
2.1.5 การใช้มิติสี HSV ใน LUT (RGB to Color Code).....	10
2.2 การจ๊ากกลุ่ม (Blob).....	12
2.2.1 ทำไมต้องจ๊ากกลุ่ม.....	12
2.2.2 วิธีไล่วงขอบสี.....	12
2.2.3 วิธีสร้างกลุ่มข้อมูลแยกโดยพิกัดแกน x และ y	12
2.2.4 วิธีหาการติดกันของกลุ่มสี.....	13
2.3. ความบิดเบือนภาพ (Barrel distortion).....	16
2.3.1 การแก้ไขโดยใช้ตารางหมากรุก.....	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.3.2 การแก้ไขโดยใช้สมการความบิดเบือน.....	18
บทที่ 3 การออกแบบระบบการมอง.....	20
3.1 การจับภาพ (Image Capture).....	20
3.2 การแยกสี (Segmentation).....	20
3.3 การจัดกลุ่มสี (Blob).....	22
3.4 การระบุรหัสหุ่นยนต์ (Robot Identification).....	22
3.5 การหาพิกัดหุ่นยนต์ฝ่ายตรงข้าม.....	27
3.6 การหาลูกบอล (Ball Identification).....	27
3.7 การแก้ไขความบิดเบือนภาพ.....	30
3.8 การส่งข้อมูลระหว่างระบบปัญญาประดิษฐ์ (AI) กับระบบประมวลผลภาพ.....	30
บทที่ 4 ผลการทดลอง.....	32
4.1 อุปกรณ์ที่ใช้ในการทดลอง.....	32
4.2 ผลการทดลองประมวลผลภาพ.....	33
4.3 ทดสอบความถูกต้องของอัลกอริทึมจากการทดสอบจริงในภาคสนาม.....	37
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	38
5.1 สรุปผลการทดลอง.....	38
5.2 ข้อเสนอแนะในการวิจัย.....	38
เอกสารอ้างอิง.....	39
ภาคผนวก ก การทดสอบเบื้องต้น.....	42
ก.1 การทดสอบมิติสีแบบต่าง ๆ (Colors Space).....	42
ก.2 การทดสอบกล้อง (Camera) เพื่อเลือกใช้งาน.....	45
ก.3 การทดสอบเลนส์ (Lans) ของกล้อง.....	49
ก.4 การทดสอบสายสัญญาณของกล้อง.....	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
ก.5 ผลการทดสอบเบื้องต้น.....	54
ก.5.1 ทดสอบมิตีสี.....	54
ก.5.2 ทดสอบสมรรถนะของกล้อง.....	57
ก.5.3 ทดสอบเลนส์.....	60
ก.5.4 ทดสอบสายสัญญาณ.....	62
ภาคผนวก ข ผลงานทางวิชาการที่ได้รับการตีพิมพ์.....	65
ประวัติผู้เขียน.....	70



สารบัญตาราง

ตารางที่	หน้า
4.1 ค่าความผิดพลาดหลังการแก้ Distortion แล้ว.....	36
4.2 ค่าความเร็วในการประมวลผลภาพ.....	36
ก.1 การทดสอบจุดสีที่ผิดเพี้ยนจากสีพื้นหลังของกล้องทั้ง 3 รุ่นเฉลี่ยใน 100 ภาพ.....	60
ก.2 เปรียบเทียบจุดสีที่ผิดเพี้ยน ไปจากพื้นหลังของเส้นสีทั้ง 2 แบบเฉลี่ยใน 100 ภาพ.....	62
ก.3 จุดสีที่ผิดเพี้ยน ไปจากพื้นหลังของสายสัญญาณทั้ง 3 แบบเฉลี่ยใน 100 ภาพ.....	64



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 ส่วนประกอบของสี 24 บิต.....	4
2.2 ค่าที่อ่านจากจุดสี 24 บิตจากลูกบอลในสนามจริงที่สร้างโดยผู้วิจัย.....	4
2.3 ตัวอย่างขอบเขตของแต่ละสีที่ใช้จริง ที่แสงในสนามประมาณ 400 lux ซึ่งสอดคล้องกับค่าที่ได้ ในรูปที่ 2.2 โดยโปรแกรมการวิเคราะห์สีนี้ผู้วิจัยพัฒนาด้วย VB6.....	6
2.4 การตั้งค่าสี โดยการเลื่อนแถบค่าในกรอบ "Color Limit" และหากต้องการบันทึกให้กดปุ่ม Save.....	7
2.5 ตารางเทียบสีในระบบแสดงผล 256 สี (8 บิต).....	9
2.6 การแปลงสี RGB 24 บิตเป็นสี HSV	9
2.7 ตารางเทียบที่ใช้ในวิทยานิพนธ์.....	10
2.8 ขั้นตอนการสร้าง LUT.....	11
2.9 การจัดกลุ่ม.....	12
2.10 วิธีการแยกกลุ่มสีในที่นี้ <> คือสัญลักษณ์ \neq ในภาษา VB6.....	13
2.11 ปัญหาที่สามารถเกิดขึ้นกับวิธีแยกกลุ่มสีในรูปที่ 4.5 ทั้งที่วัตถุตั้งคั่นเป็นชั้นเดียวกัน.....	14
2.12 การรวมจุดศูนย์กลางกลุ่มสีเข้าด้วยกัน.....	16
2.13 การวางตารางลงบนสนาม.....	17
2.14 การหาจุดตัดของตารางเพื่อหาพิกัด	17
2.15 ความบิดเบือนแบบดึงน้ำมัน [3].....	18
2.16 ภาพจริงที่ได้จากกล้อง.....	19
2.17 ผลจากการแก้ความบิดเบือน.....	19
3.1 ส่วนประกอบของสี 24 บิต.....	20
3.2 บล็อกไดอะแกรมการแปลงค่าสี.....	21
3.3 ผลลัพธ์ที่ได้ในหน่วยความจำเมทริกซ์.....	22
3.4 การหามุมของหุ่นยนต์จากกลุ่มสีม่วง.....	23
3.5 การหามุมของหุ่นยนต์จากกลุ่มสีม่วงร่วมกับสีฟ้า.....	24
3.6 การหมุนมุมเพื่อง่ายต่อการหารหัสของหุ่นยนต์.....	26
3.7 โปรแกรมขณะพบปัญหาเกี่ยวกับสีของลูกบอลไปทับกับสีม่วง.....	29
3.8 โปรแกรมขณะทำงาน (ใช้อัลกอริทึมในการหาบอล).....	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.7 แบบจำลองของความบิดเบือนแบบถังน้ำมัน (Barrel Distortion).....	36
4.1 การติดตั้งอุปกรณ์ในการแข่งขันจริง.....	33
4.2 การทดลองโดยใช้โปรแกรมที่ผู้วิจัยสร้างขึ้น ตามหลักการในบทที่ 3 วางหุ่นยนต์ ณ พิกัดจริง เท่ากับ (159,200) ของสนาม.....	33
4.3 พิกัดแนวแกน x ของหุ่นยนต์ใน 100 ภาพ ณ พิกัดจริงเท่ากับ (159, 200) ของสนาม.....	34
4.4 พิกัดแนวแกน y ของหุ่นยนต์ใน 100 ภาพ ณ พิกัดจริงเท่ากับ (159, 200) ของสนาม.....	34
4.5 พิกัดมุม θ ของหุ่นยนต์ใน 100 ภาพ ณ พิกัดมุมจริงเท่ากับ 45.....	35
4.6 การทดลองจริงในภาคสนาม.....	37
ก.1 มิตีสีแบบ RGB เป็นมิตีสีที่ใช้แสดงผลบนหน้าจอคอมพิวเตอร์ (Monitor).....	42
ก.2 มิตีสีแบบ RGB ในรูปแบบของแกนสี.....	42
ก.3 มิตีสีแบบ HSV ซึ่ง A แทนค่า S (saturation), B แทนค่า H (hue), C แทนค่า V (value) และ D แทนค่า H (hue) เหมือนกับ B [38].....	44
ก.4 มิตีสีแบบ HSV ในรูปแบบของแกนสี [38].....	45
ก.5 กล้องวงจรเปิด ของ Sonican.....	46
ก.6 ภาพที่ได้จากกล้องวงจรเปิด ของ Sonican.....	46
ก.7 กล้องถ่ายวิดีโอ Sony รุ่น TRV35E.....	47
ก.8 ภาพที่ได้จากกล้องถ่ายวิดีโอ Sony.....	47
ก.9 กล้องถ่ายวิดีโอ Panasonic NV-GS200.....	48
ก.10 วิธีการแยกสีออกเป็นแดง เขียว น้ำเงินในกล้อง 3 CCD.....	48
ก.11 ภาพที่ได้จากกล้องถ่ายวิดีโอ Panasonic NV-GS200.....	49
ก.12 เลนส์ Sony wide conversion รุ่น VCI-0637H.....	49
ก.13 ภาพที่ได้จากเลนส์ Sony wide conversion รุ่น VCI-0637H.....	50
ก.14 เลนส์ Panasonic wide conversion รุ่น VW-LW3707M3.....	50
ก.15 ภาพที่ได้จากเลนส์ Panasonic.....	51
ก.16 สาย A/V ขั้วต่อ RCA.....	51
ก.17 สาย Y/C หรือเรียกอีกอย่างว่า S-Video ขั้วต่อแบบ Mini-DIM 4 Pin.....	52
ก.18 สาย IEEE1394 ขั้วต่อแบบ เล็กและแบบใหญ่.....	52
ก.19 มาตรฐานสายสัญญาณ IEEE1394.....	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
ก.20 โปรแกรมที่เขียนเพื่อดูระดับสีในมิติของ RGB, YUV และ HSV	54
ก.21 ผลของการทดสอบมิติของสี RGB.....	55
ก.22 ผลของการทดสอบมิติของสี YUV.....	56
ก.23 ผลของการทดสอบมิติของสี HSV.....	57
ก.24 กล้องวงจรปิด ของ Sonican รุ่น HV-2 ใน 100 ภาพ.....	58
ก.25 กล้องถ่ายวิดีโอ ของ Sony รุ่น TRV35E ใน 100 ภาพ.....	59
ก.26 กล้องถ่ายวิดีโอ ของ Panasonic รุ่น NV-GS200 ใน 100 ภาพ.....	59
ก.27 เลนส์อัตรากว้าง 0.6 เท่า ของ Sony รุ่น VCI-0637H ใน 100 ภาพ.....	61
ก.28 เลนส์อัตรากว้าง 0.7 เท่า ของ Panasonic รุ่น VW-LW3707M3 ใน 100 ภาพ.....	61
ก.29 สายสัญญาณเอวี (A/V) ขั้วต่อ RCA ใน 100 ภาพ.....	63
ก.30 เอชวิดีโอ (S-video) ใน 100 ภาพ.....	63
ก.31 สายสัญญาณไออีอีอี 1394 (IEEE1394 หรือเรียกอีกอย่างว่า FireWire) ใน 100 ภาพ.....	64

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของงานวิจัย

ตัวรับสัญญาณ (sensor) ที่นิยมใช้บอกพิกัดในหุ่นยนต์โดยมากนั้นคือ แสงอินฟราเรด (Infrared) และคลื่นความถี่เหนือเสียง (ultrasonic) ซึ่งเป็นอุปกรณ์ที่ค่อนข้างใช้ง่ายและมีความแม่นยำสูง แต่เนื่องจากความสามารถของตัวอุปกรณ์ครอบคลุมระยะเพียง 1 เมตรเท่านั้น ดังนั้นจึงได้นำระบบการมอง (vision system) มาใช้เพิ่มความสามารถของหุ่นยนต์ในการแบ่งแยก สี รูปร่าง และเพิ่มขีดความสามารถในการวัดระยะเป็น 2 มิติได้ การพัฒนาระบบตัวรับสัญญาณดังกล่าวนำมาใช้ในภาคอุตสาหกรรมและในภาคการศึกษา เพื่อกระตุ้นกระบวนการเรียนการสอนในภาคการศึกษา ดังนั้นจึงเกิดการแข่งขันหุ่นยนต์ประเภทหนึ่งเมื่อปี ค.ศ. 1997 ที่ประเทศญี่ปุ่นเป็นครั้งแรก ซึ่งบังคับให้ใช้ระบบการมองในการแข่งขัน การแข่งขันนี้คือ การแข่งขันหุ่นยนต์เตะฟุตบอล (RoboCup) [1] ต่อมาการแข่งขันนี้ได้รับความนิยมอย่างมากในต่างประเทศ และมีการพัฒนาอย่างต่อเนื่องของมหาอำนาจชั้นนำของโลกตั้งแต่ปี 1997 อย่างเช่น Cornell University, Carnegie Mellon University, Tokyo Institute of Technology ฯลฯ เป็นต้น

จากกระบวนการเรียนรู้ที่ควบคู่กับการบันเทิง (Plearn = Play & Learn) ในรูปแบบการแข่งขันจึงเป็นแรงบันดาลใจให้ศึกษาปัญหาดังปรากฏในวิทยานิพนธ์นี้ ดังนั้นเนื้อหาวิทยานิพนธ์นี้จึงได้อ้างอิงกฎกติกาการแข่งขัน ซึ่งในที่นี้เป็นหุ่นยนต์ประเภทขนาดเล็ก (small-size) อันเป็นข้อกำหนดขอบเขตของเนื้อหา ผลที่ได้จากงานวิจัยนี้จะยังประโยชน์แก่การพัฒนาการมองของหุ่นยนต์ทั้งในเชิงการศึกษาและเชิงอุตสาหกรรมของประเทศไทยต่อไป

1.2 วัตถุประสงค์ของงานวิจัย

- 1) เพื่อพัฒนาระบบการมองของหุ่นยนต์ให้สามารถใช้ในการแข่งขัน Robocup ได้
- 2) เพื่อเป็นระบบอ้างอิงในการประยุกต์ใช้ในทางอุตสาหกรรม เช่น การคัดพืชผลทางการเกษตร
- 3) เพื่อเป็นแนวทางในการใช้ระบบการมองในหุ่นยนต์และสร้างองค์ความรู้ในศาสตร์นี้ให้กับประเทศไทย
- 4) เพื่อให้ให้นักวิจัยที่วิจัยระบบปัญญาประดิษฐ์ (AI) สามารถพัฒนาหุ่นยนต์ที่ฉลาดขึ้นและมีความสามารถมากขึ้น

1.3 ขอบเขตของงานวิจัย

- 1) ทำการทดลองระบบ เปรียบเทียบกันทั้งพิกัดและมุมจริงของหุ่นยนต์ในสนาม
- 2) สามารถประมวลผลได้มากกว่า 24 ภาพต่อวินาที (เทียบเท่าตาของมนุษย์)
- 3) สามารถแยกวัตถุ (หุ่นยนต์และสิ่งกีดขวาง) และระบุหมายเลข (ID) ตามสัญลักษณ์สีที่ติดอยู่บนตัวหุ่นยนต์
- 4) สามารถแก้ไขความบิดเบือนของภาพ เพื่อให้ได้พิกัดหุ่นยนต์ที่ถูกต้อง
- 5) ทนทานต่อสภาพแวดล้อมที่เปลี่ยนแปลงได้ เช่น แสงจากการถ่ายรูป (flash light) และแสงที่ระบุในข้อบังคับของการแข่งขันหุ่นยนต์เตะฟุตบอล (RoboCup) [1, คู่มือผนวก ก]
- 6) นำระบบที่ออกแบบมาทดลองใช้จริงในการแข่งขัน Robocup Thailand Championship 2002 และ Robocup Thailand Championship 2003

1.4 ขั้นตอนของงานวิจัย

- 1) ค้นคว้าเอกสารและข้อมูลที่เกี่ยวข้อง
- 2) ศึกษาเอกสารและข้อมูลที่รวบรวมได้
- 3) ออกแบบระบบการมอง และเตรียมอุปกรณ์ต่าง ๆ ที่เกี่ยวข้อง
- 4) ทำการทดลองระบบที่ออกแบบ
- 5) นำระบบไปใช้จริงในสนามแข่งขัน Robocup Thailand Championship
- 6) สรุปผลและเรียบเรียงวิทยานิพนธ์

1.5 สมมุติฐานเบื้องต้น

- 1) ระบบที่ออกแบบจะต้องประมวลผลได้ใกล้เคียงหรือเท่ากับความเร็วกล้อง คือ 24 ภาพต่อวินาที (real time process) เนื่องจากถ้าประมวลผลช้ากว่านี้ทำให้ไม่สามารถประมวลผลภาพทั้งหมดที่กล้องส่งมาได้ และถ้าเร็วกว่านี้ไม่มีประโยชน์เพราะจะประมวลผลภาพซ้ำเดิม
- 2) วิธีที่ใช้เป็นวิธีที่ทำให้คอมพิวเตอร์ประมวลผลน้อย แต่ให้ผลลัพธ์ที่ดี
- 3) กล้องที่ใช้มีการตอบสนองต่อแสงได้ดี เพราะเป็นปัจจัยสำคัญของระบบประมวลผลภาพ
- 4) เลนส์ (lens) ที่ใช้มีการลดทอนแสงน้อยและให้ความบิดเบือนของภาพน้อย

สมมติฐานดังกล่าวเป็นข้อตกลงของงานวิจัยนี้ ในกรณีที่อุปกรณ์ที่ใช้ไม่เป็นไปตามข้อตกลงดังกล่าวข้างต้นนี้ ผู้วิจัยต้องทำการปรับพารามิเตอร์หรือจัดหาตั้งค่าอุปกรณ์ต่าง ๆ ให้เป็นไปตามสมมติฐานทั้งสี่ข้อ เพื่อที่จะสามารถทำการทดลองให้ได้ผลตรงกับที่วิธีที่ได้เสนอแนะไว้ในวิทยานิพนธ์นี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

บทนี้กล่าวถึงทฤษฎีที่เกี่ยวข้องที่ได้นำมาใช้ในงานวิจัยนี้ เริ่มจากการแยกสี การจัดกลุ่มจนถึงการแก้ความบิดเบือนของขอบภาพบริเวณขอบเลนส์ให้ได้พิกัดที่ถูกต้อง ดังมีรายละเอียดเป็นสังเขปดังนี้ สำหรับเนื้อหาโดยละเอียดสามารถอ่านเพิ่มเติมได้ที่ [2, 3, 8, 11, 19-25]

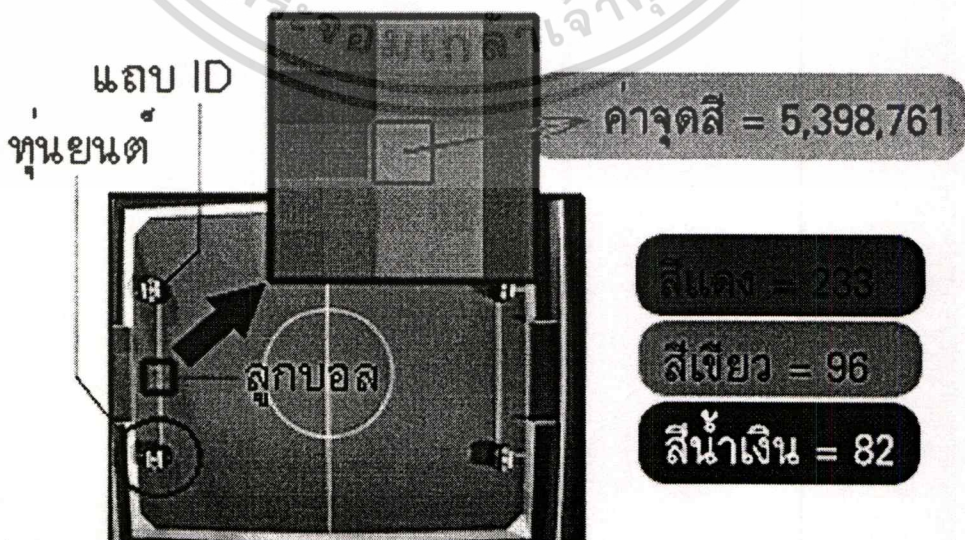
2.1 การแยกสี (Color Segmentation)

2.1.1 ค่าสีที่ได้จากกล้อง

ในการอ่านค่าสีใช้มาตรฐานบิตแมป (bitmap) แบบ 24 บิต ระดับสีต่อ 1 พิกเซล (pixel) โดยแบ่งเป็น สีแดง สีเขียว สีน้ำเงิน อย่างละ 8 บิต หรือ 256 ระดับสี ดังแสดงในรูปที่ 2.1 และ 2.2 ตามลำดับ



รูปที่ 2.1 ส่วนประกอบของสี 24 บิต



รูปที่ 2.2 ค่าที่อ่านจากจุดสี 24 บิตจากลูกบอลในสนามจริงที่สร้างโดยผู้วิจัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นาเบไซเซอร์เซชันด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยภาพที่ได้จากกล้องที่ใช้ในที่นี้ตั้งความละเอียดไว้ที่ 320x240 พิกเซล มีจุดสีทั้งหมด $320 \times 240 = 76,800$ จุดสี จุด 1 จุดมีขนาดข้อมูล 3 ไบต์ (byte) แสดงว่าจะต้องประมวลผลข้อมูลทั้งหมดเท่ากับ $320 \times 240 \times 3 = 230,400$ ไบต์

2.1.2 การใช้มิติสี HSV โดยตั้ง Min-Max Threshold ในการเลือกช่วงสี

ในการแปลงค่าสีในระบบ RGB มาเป็นระบบ HSV จะทำให้สามารถระบุสีได้ชัดเจนแม่นยำขึ้น การแปลงดังกล่าวอาศัยสมการที่กล่าวไว้ในภาคผนวก ก สมการที่ ก.3, ก.4 และ ก.5 เมื่อนำสมการเหล่านี้มาใช้จะต้องทำการกำหนดขอบเขตของสีที่ต้องการด้วย เนื่องจากสีหนึ่งสีที่กล้องจับภาพมาได้ นั้นมักจะมีสัญญาณรบกวนติดมาด้วย จึงทำให้สีที่เห็นนี้เป็นสีเดียวกัน คอมพิวเตอร์จะคำนวณค่า HSV ได้หลายแบบด้วยกัน เช่น ค่าสีเหลืองของแถบสีบนหุ่นยนต์อาจจะได้ค่า ณ จุดเดียวกันแต่ ณ เวลาต่างกัน ได้เป็น

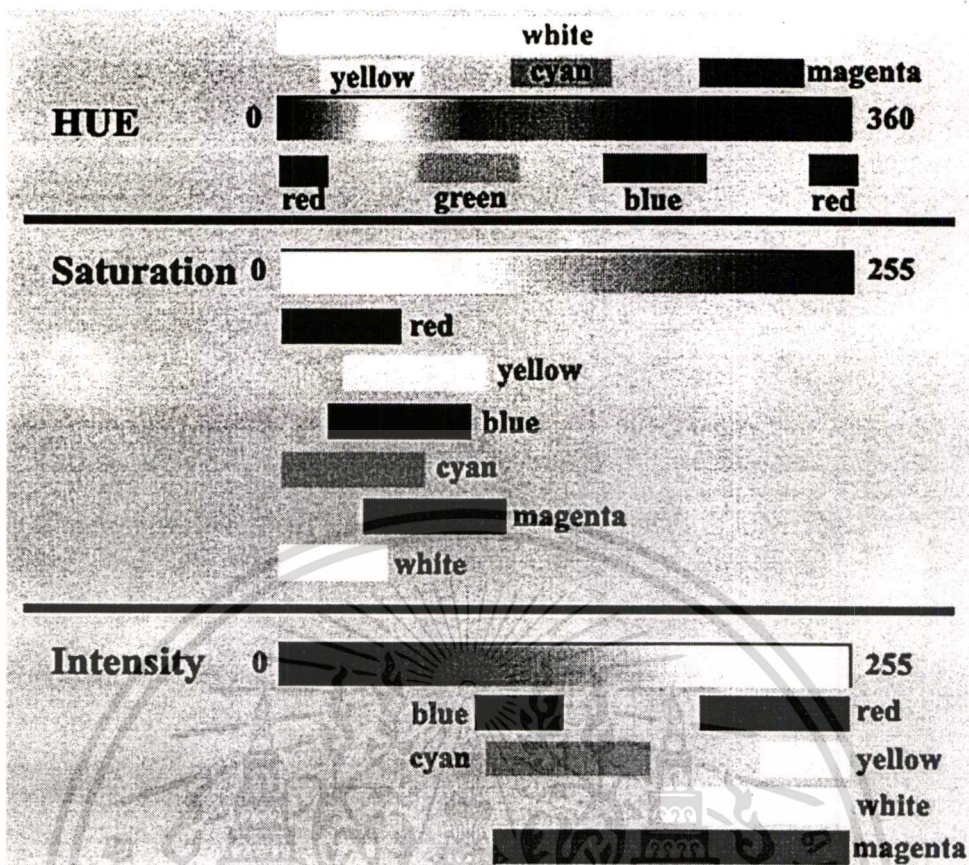
$$\text{Hue} = 30 - 80$$

$$\text{Saturation} = 0 - 50$$

$$\text{Intensity (Value)} = 70 - 150$$

จากตัวอย่างนี้จะเห็นได้ว่า ถ้าจะให้จับสีได้ต้องกำหนดขอบเขตของสี เพราะไม่อย่างนั้นก็ไม่สามารถจับสีที่ต้องการได้ทั้งหมด และประสิทธิภาพสูงสุดในการจับสีก็ควรกำหนดขอบเขตสีให้กว้างที่สุดโดยไม่รบกวนกับสีอื่น ๆ ซึ่งในการปรับแต่ง (Calibration) นั้น ต้องนำค่าของพิกเซลสีที่สนใจทั้งหมดมากำหนดค่าต่ำสุด (Min) และค่าสูงสุด (Max) เพื่อหาค่าเปรียบเทียบ (Threshold)

ข้อเสียของวิธีนี้คือ ในภาพขนาดกว้าง 320 พิกเซล ยาว 240 พิกเซล และ ความเร็ว 25 ภาพต่อวินาที ต้องใช้สูตรคำนวณ $320 \times 240 \times 25 = 1,920,000$ ครั้งต่อวินาที ซึ่งใช้จำนวนครั้งในการคำนวณอย่างมาก



รูปที่ 2.3 ตัวอย่างขอบเขตของแต่ละสีที่ใช้จริง ที่แสงในสนามประมาณ 400 lux ซึ่งสอดคล้องกับค่าที่ได้ในรูปที่ 2.2 โดยโปรแกรมการวิเคราะห์สีนี้ผู้วิจัยพัฒนาด้วย VB6

2.1.3 การใช้มิติสี HSV โดยตั้ง Mid-Tolerance Threshold ในการเลือกช่วงสี

ในหัวข้อนี้รายละเอียดวิธีการทำงานเหมือนกันกับหัวข้อที่แล้ว แต่ต่างกันตรงการหาค่าเปรียบเทียบ (Threshold) เนื่องจากในหัวข้อที่แล้วเป็นการยากที่จะนำพิกเซล (Pixel) ที่สนใจทั้งหมดมาคำนวณได้ จึงใช้วิธีใหม่ในการปรับแต่งโดยการกำหนดค่ากลางของสีที่สนใจขึ้นมา แล้วกำหนดช่วงบวกลบของข้อมูลนั้นทำให้ง่ายต่อการปรับแต่งขึ้นมาก วิธีนี้เป็นวิธีที่นักวิจัยแห่ง Cornell University ใช้ในปี 2000 [35]

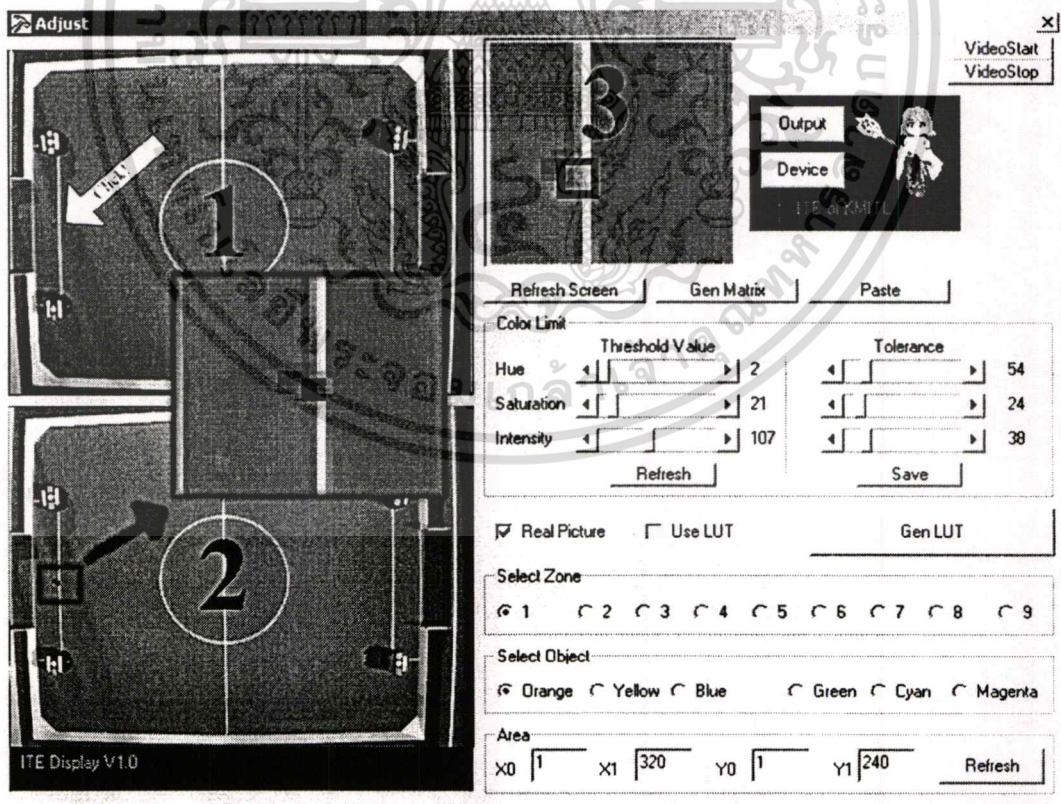
ในวิธีการตั้งค่าขอบเขตนี้ทำได้โดยการนำค่าตัวอย่างสีที่ได้จากกล้องจริงในขณะนั้นมา 1 จุดสี เพื่อนำค่าสีที่ได้มาตั้งขอบเขตแล้วนำกลับไปประมวลผลในภาพจริง และดูผลว่าขอบเขตสีที่ตั้งขึ้นนั้นเหมาะสมหรือไม่ เพราะต้องตั้งขอบเขตสีให้กว้างที่สุดแต่ต้องไม่กวนกับสีอื่น ๆ ในภาพดังตัวอย่าง

การปรับแต่งค่าสี (Color Calibration)

ขั้นตอนการปรับแต่งค่าสีผู้วิจัยได้เขียนซอฟต์แวร์เพื่อใช้ในการปรับแต่งค่าสี โดยในงานวิจัยนี้เริ่มจากเลือกสีที่จะตั้งค่าในกรอบ (Frame) "Select Object" ซึ่งสามารถตั้งได้ 6 สี คือ

1. สีส้ม (Orange) เป็นส่วนของสีลูกบอล
2. สีเหลือง (Yellow) เป็นส่วนของสีทิมฮุนยนต์ 1
3. สีน้ำเงิน (Blue) เป็นส่วนของสีทิมฮุนยนต์ 2
4. สีม่วงแดง (Magenta) เป็นส่วนของสีบอกมมฮุนยนต์ที่ทำกับสนาม 1
5. สีฟ้า (Cyan) เป็นส่วนของสีบอกมมฮุนยนต์ที่ทำกับสนาม 2
6. สีขาว (White) เป็นส่วนของสีบอกหมายเลขฮุนยนต์ (ID)

หลังจากเลือกสีที่จะตั้งค่าเรียบร้อยแล้ว ให้กด (Click) บริเวณที่ 1 ในรูปที่ 2.4 โลกกับลูกบอล จะเกิดภาพ ซึ่งสามารถดูภาพขยายของบริเวณที่ 1 ได้จากในบริเวณที่ 3 จากนั้นให้กด (Click) บนบริเวณที่ 3 เลือกพื้นที่สีที่ต้องการ (ดังรูปที่ 2.4 เลือกสีของลูกบอล) โปรแกรมจะประมวลผลค่าสีในพื้นที่ที่เลือกเพื่อหาค่ากลางเป็นค่า สีแดงเขียวน้ำเงิน (RGB) และใช้สมการในหัวข้อ ก.1 ในภาคผนวกให้ได้ผลลัพธ์เป็นมิตติสีแบบ HSV มาแสดงบนกรอบ "Color Limit" (ในส่วนของค่าสี Threshold Value)



รูปที่ 2.4 การตั้งค่าสีโดยการเลื่อนแถบค่าในกรอบ "Color Limit" และหากต้องการบันทึกให้กดปุ่ม Save

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับแต่งค่าสีทำได้จากแถบในกรอบ "Color Limit" ซึ่งมีความหมายและมีผลในการปรับดังนี้

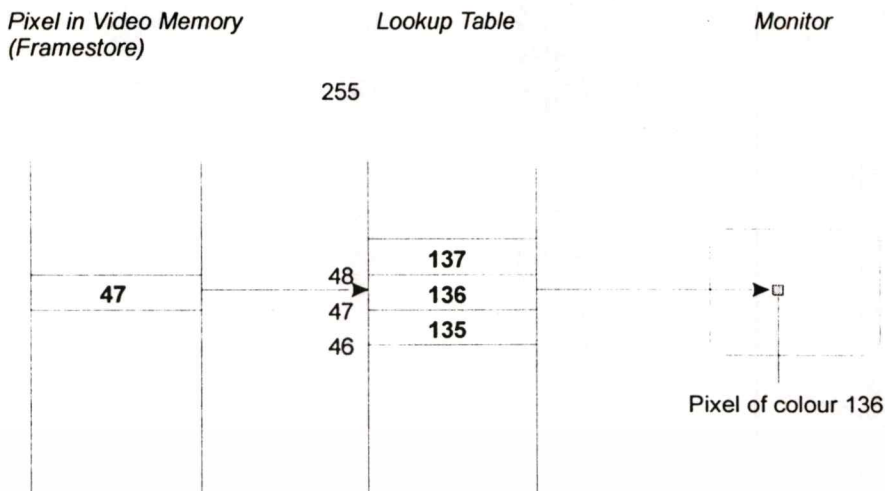
- Hue (Threshold Value) คือค่าสีมีค่าตั้งแต่ 0 ถึง 359 ปรับเพื่อเลือกค่าสี (สีแดง = 0)
- Saturation (Threshold Value) คือค่าความเข้มสีมีค่าตั้งแต่ 0 ถึง 255
- Intensity (Threshold Value) คือค่าความสว่างมีค่าตั้งแต่ 0 ถึง 255
- Tolerance คือค่าความกว้างของช่วงที่พิจารณา เช่น ในรูปที่ 2.4 ค่า Hue (Threshold Value) เท่ากับ 2 และค่า Tolerance เท่ากับ 54 หมายความว่า โปรแกรมจะพิจารณาในช่วงตั้งแต่ 308-359 กับ 0-56 (เนื่องจากค่า Hue พิจารณาแบบวนเป็นวงกลม) หรือ Saturation (Threshold Value) เท่ากับ 21 และค่า Tolerance เท่ากับ 24 หมายความว่า โปรแกรมจะพิจารณาในช่วงตั้งแต่ 0-45 (จริง ๆ ควรจะเป็น -3 ถึง 45 แต่ช่วงลบไม่ได้ใช้งาน) โดยปรกติจะตั้งค่าความกว้าง (Tolerance) ให้มากที่สุดเท่าที่จะเป็นไปได้แต่ต้องดูแลผลลัพธ์ที่บริเวณที่ 2 ของรูปที่ 2.4 ด้วย ว่ามีการทับกันของสีหรือไม่ เพื่อป้องกันการตีความผิดพลาดจากจุดที่เสมือนสีแดง

การตั้งค่าทุกครั้งต้องพิจารณาผลลัพธ์ที่บริเวณที่ 2 ของรูปที่ 2.4 ด้วยทุกครั้ง จะมีการแสดงผลโดยการทาสีแดงลงบนช่วงสีเป้าหมาย ยังมีปริมาณสีแดงมากยิ่งดี แต่ไม่ควรให้สีแดงไปปรากฏบนสีที่ไม่ใช่เป้าหมาย เพราะทำให้โปรแกรมคำนวณพิกัดผิดพลาดได้ง่าย ปรับแต่งแบบเดียวกันนี้จนครบทุกสี และจัดเก็บค่าเพื่อที่จะเข้าสู่กระบวนการสร้างตารางเทียบค่า (Lookup Table, LUT) ต่อไป

2.1.4 การใช้มิติสี HSV ใน LUT (RGB to HSV)

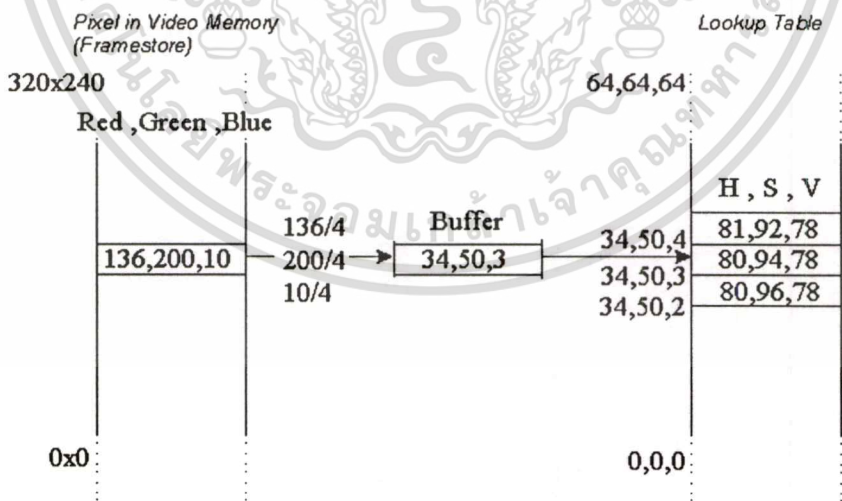
ในจุดสีแต่ละจุดสีจะมีค่าได้ตั้งแต่ 0 ถึง 16777216 (สี 24 บิต) จะต้องนำมาแปลงค่าให้โปรแกรมคอมพิวเตอร์สามารถตีความได้ โดยผ่านสูตรการคำนวณมิติสี (color space) แบบต่าง ๆ และยังคงเปรียบเทียบกับค่าสีที่ตั้งไว้อีก 1 ขั้นตอน ซึ่งการคำนวณที่กล่าวมานั้นใช้การประมวลผลโดยคอมพิวเตอร์จำนวนมาก สมมติว่าใช้ภาพขนาด 320x240 จุด จะประมวลผลสมการการแปลง RGB เป็น HSV ซ้ำกันจำนวน $320 \times 240 = 76800$ รอบต่อ 1 ภาพ เป็นอย่างน้อย จึงต้องหาวิธีใหม่เพื่อเพิ่มความเร็วและลดภาระการประมวลผลของเครื่องคอมพิวเตอร์ในการแปลงค่าสี ซึ่งอาจทำได้โดยการแปลงค่าสีได้อย่างรวดเร็วคือ ตารางเทียบค่า

ขอบเขตทั้งหมดของสีที่มีอยู่บนระบบแสดงผลนั้นเรียกว่า งานผสมสี (palette) แต่โดยมากงานผสมสีที่ถูกใช้งานเฉพาะบางส่วน ในระบบสีจริงอาจจะมีข้อจำกัดของขอบเขตสีที่มีความต่างของชนิดกับการใช้งาน และในระบบที่ต้องการความเร็วในการประมวลผลขอบเขตสีที่ใช้จะมีขนาดเล็กกว่างานสี ส่วนความสามารถเหล่านี้ใช้วิธีการเก็บ LUT ไว้ในหน่วยความจำ ซึ่งประกอบด้วยค่าจุดสีหลาย ๆ ค่าที่สอดคล้องกับสีบนจอภาพ (ดูรูปที่ 2.5)



รูปที่ 2.5 ตารางเทียบสีในระบบแสดงผล 256 สี (8 บิต)

ค่าจุดสี (Pixel) ที่อยู่ในรูปภาพเป็นเหมือนดัชนีบ่งชี้ใน LUT เพื่อว่าค่าจุดสี (Pixel) ที่เท่ากับ 47 จะเป็นสีในลำดับที่ 47 ของ LUT ในตัวอย่างแสดงในรูป 2.5 นี้จะเป็นค่าตรงกับ 136 โดยสามารถกำหนดค่าสีของอินพุตที่เข้า LUT จับคู่กับค่าสีของส่วนที่แสดงผลตามที่กำหนดไว้ โดยอาศัยหลักการนี้ภาพที่แสดงผลจะไม่ผิดเพี้ยนแม้ว่าบิตของข้อมูลขาเข้ากับขาออกจะมีขนาดต่างกัน



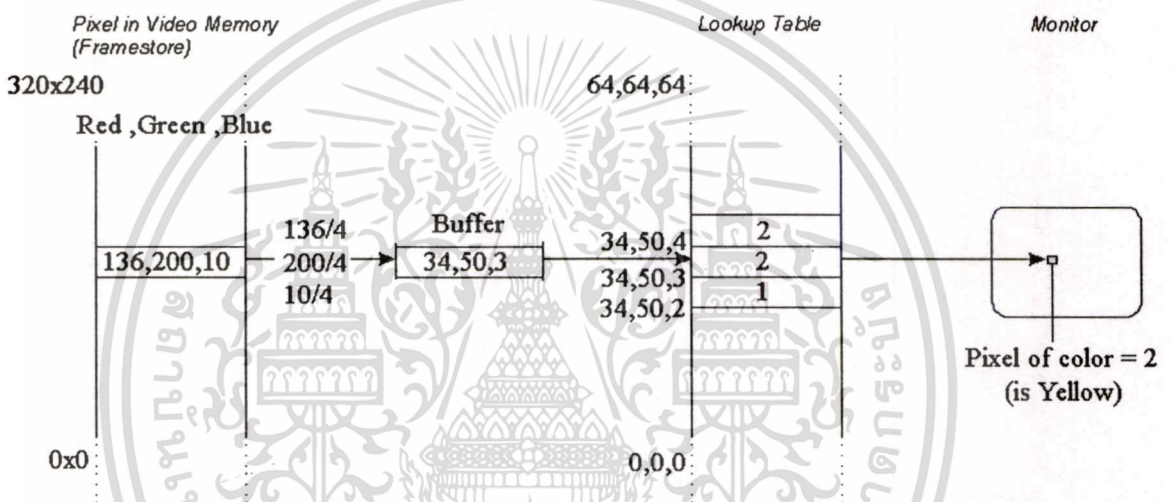
รูปที่ 2.6 การแปลงสี RGB 24 บิต เป็นสี HSV

การคำนวณค่า HSV รอไว้ใน LUT แล้วนำค่า RGB ที่ได้จากภาพมาชี้ค่า HSV ในตำแหน่งนั้น ๆ ออกมาจะทำให้การประมวลผลรวดเร็วขึ้น และนำค่า HSV ที่ได้เทียบค่า Threshold เหมือน

หัวข้อที่แล้ว ซึ่งเป็นเทคนิคเดียวกันกับที่ใช้โดยทีม RoboLoos QueenLand 2003 [36] รายละเอียดในรูปที่ 2.6 จะอธิบายในบทที่ 3 ต่อไป

2.1.5 การใช้มิติสี HSV ใน LUT (RGB to Color Code)

จากหลักการในหัวข้อ 2.1.4 ต้องดัดแปลงเพื่อที่จะนำมาใช้กับสีขนาด 24 บิต และต้องการผลลัพธ์ที่สามารถนำไปใช้ได้อย่างรวดเร็วที่สุด จึงได้ดัดแปลงให้เป็นการรับข้อมูล 24 บิตและส่งข้อมูลออกมาเป็นตัวเลข 1 ถึง 6 เป็นตัวเลขที่ระบุว่าเป็นสีอะไร เช่น เลข 1 คือสีแดง เลข 2 คือสีเหลือง เลข 3 คือสีน้ำเงิน เลข 4 คือสีฟ้า เลข 5 คือสีเขียว และเลข 6 คือสีม่วง เพื่อความสะดวกในการนำไปใช้ต่อไป ซึ่งแสดงดังรูปที่ 2.7

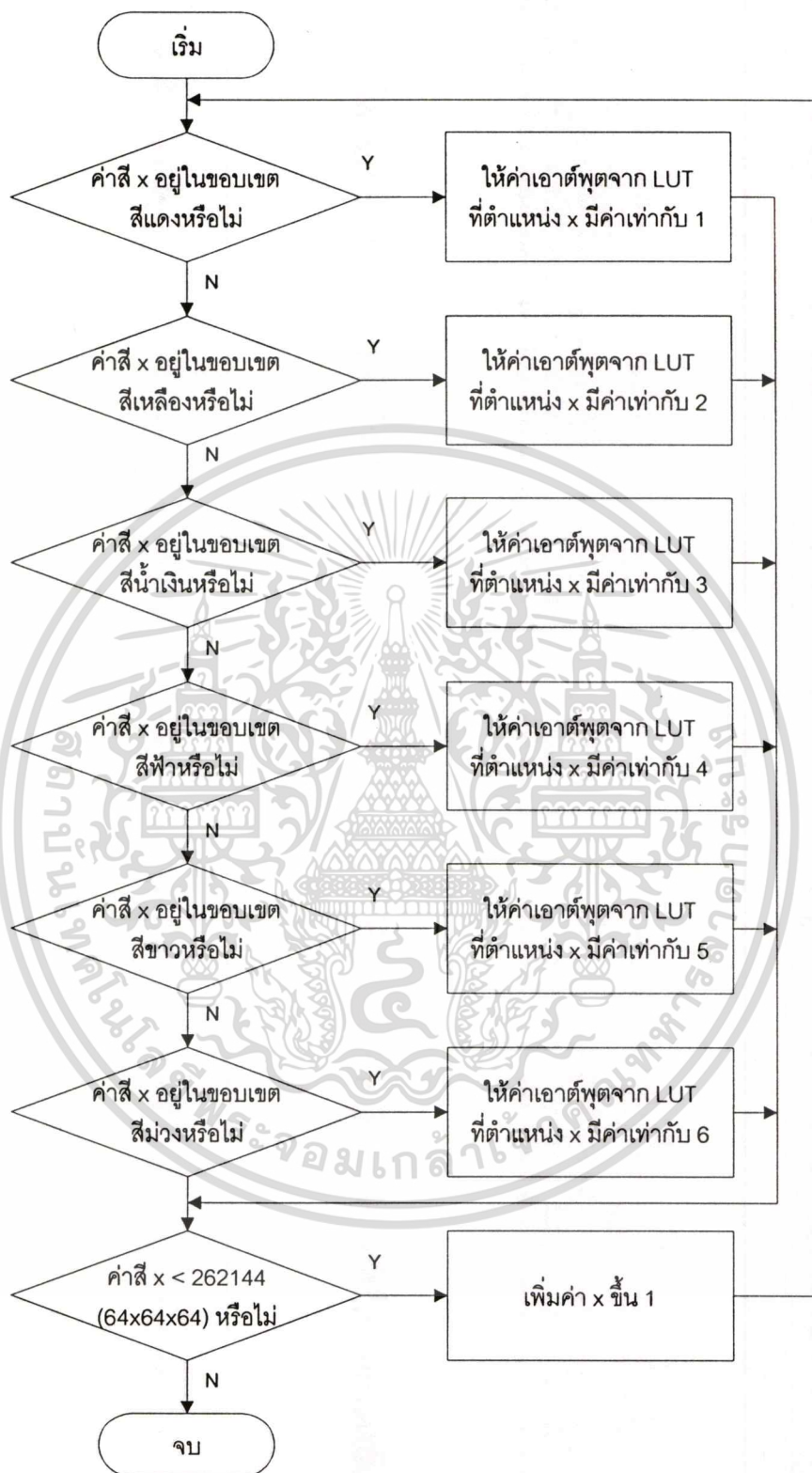


รูปที่ 2.7 ตารางเทียบค่าสีที่ใช้ในวิทยานิพนธ์

จากรูปที่ 2.7 จะเห็นได้ว่า ค่าที่รับมาจากกล้องนั้นเป็น 24 บิต ($256 \times 256 \times 256$ สี) ซึ่งเป็นค่าที่มากเกินไปสำหรับการทำตารางเทียบค่า (ถ้าทำแบบสี 24 บิตโดยตรงแล้ว จะต้องใช้ขนาดหน่วยความจำ เท่ากับ $256 \times 256 \times 256 = 16$ Mbyte) จึงได้ทำการแปลงค่าจาก 24 บิต ลงมาเหลือ 18 บิต เพื่อประหยัดทรัพยากรบนเครื่องคอมพิวเตอร์ (สี 18 บิต จะต้องใช้ขนาดหน่วยความจำ เท่ากับ $64 \times 64 \times 64 = 0.26$ Mbyte เท่านั้น) โดยการนำค่าสีแต่ละสีมาหารด้วย 4 ก่อนที่จะนำไปใช้ในตารางเทียบค่า

ดังนั้นจึงได้นำวิธีในหัวข้อ 2.1.4 มาดัดแปลงให้ดีขึ้น โดยการรวมภาคการตัดสินใจไว้ใน LUT ด้วย เพื่อให้ค่าที่ได้ออกมาเป็นค่าสีที่ต้องการ ส่วนการสร้าง LUT นั้นจะทำหลังจากการตั้งขอบเขตสี คือ เมื่อได้ขอบเขตสีทั้งหมดแล้ว (จากหัวข้อ 2.1.3) จะนำเข้าขั้นตอนการสร้าง LUT ดังรูปที่ 2.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



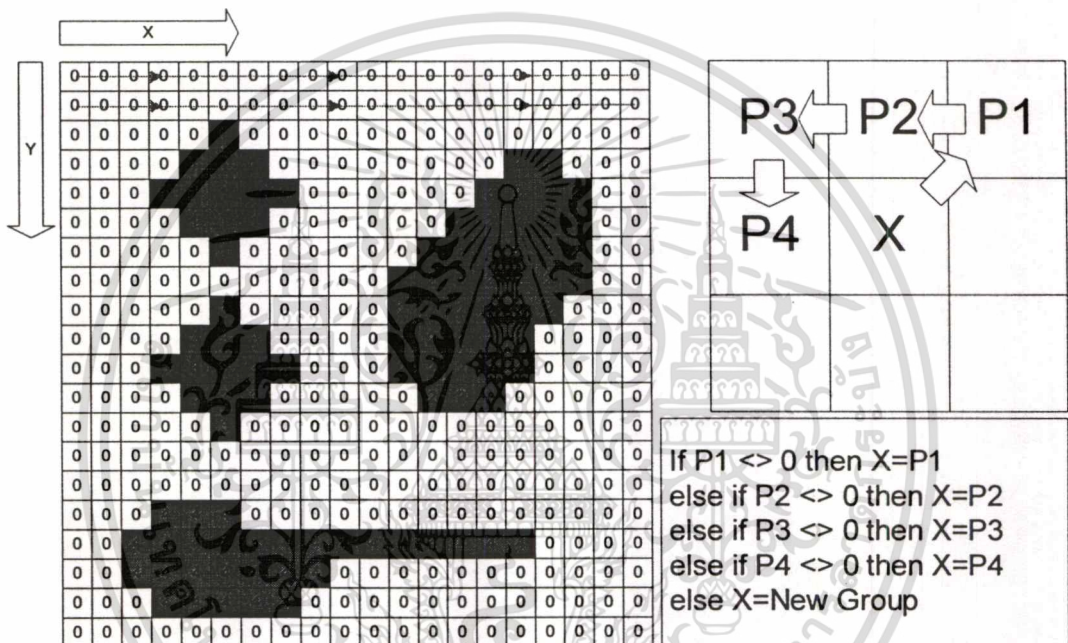
รูปที่ 2.8 ขั้นตอนการสร้าง LUT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ใกล้เคียงหรือไม่พิจารณาจากค่าความต่างของ x และ y ของจุดนั้น ๆ กับกลุ่มต่าง ๆ ซึ่งเป็นแนวคิดที่พัฒนาโดยทีม Carnegie-Mellon ปี 2002 [37]

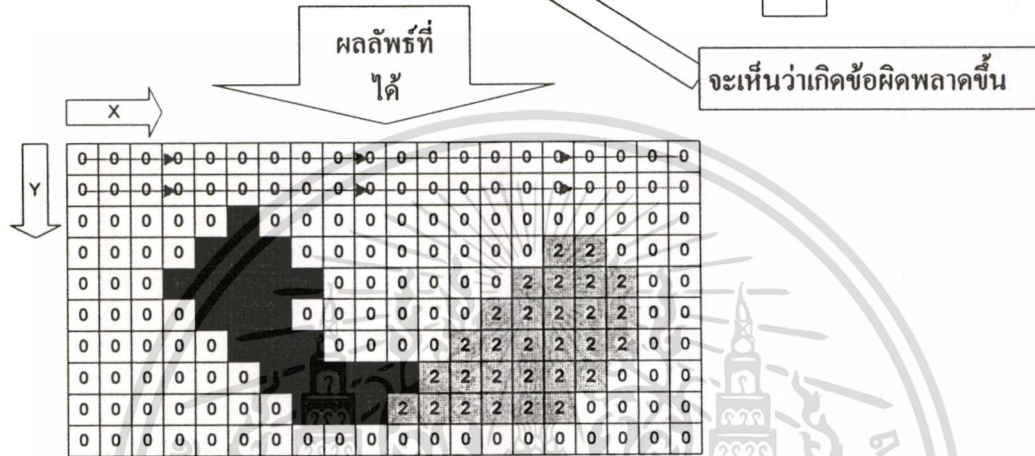
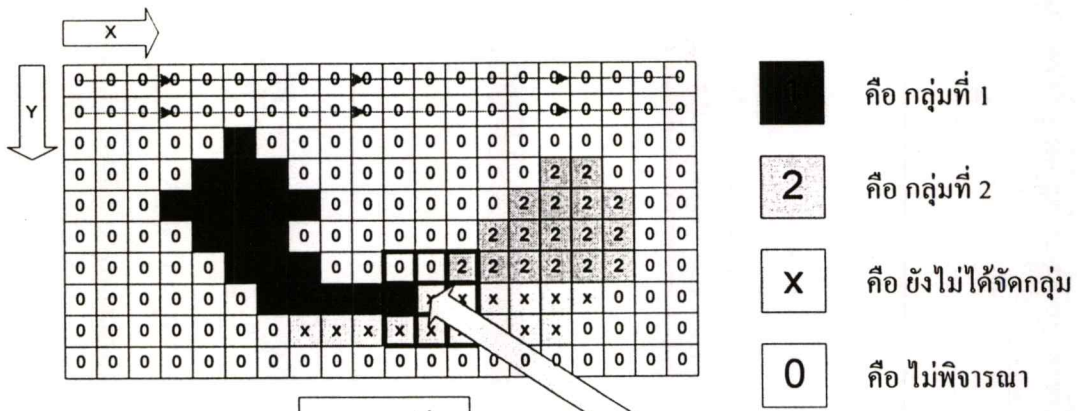
2.2.4 วิธีหาการติดกันของกลุ่มสี

วิธีนี้ทำได้โดยการรวมจุดสีให้เป็นกลุ่ม ๆ (เฉพาะสีแดงซึ่งแทนสีบอล สีเหลืองแทนสีทีม หุ่นยนต์ทีมที่ 1 และสีน้ำเงินแทนสีทีมหุ่นยนต์ทีมที่ 2) เพื่อหาลำดับและตำแหน่งจุดศูนย์กลางของแต่ละกลุ่ม โดยการหาว่ามีจุดสีเดียวกันอยู่ติดกันหรือไม่ ถ้าจุดรอบข้างมีสีเดียวกันก็จะกำหนดให้เป็นกลุ่มเดียวกัน แต่ถ้าไม่มีจุดสีเดียวกันอยู่รอบข้างก็จะตั้งเป็นกลุ่มใหม่



รูปที่ 2.10 วิธีการแยกกลุ่มสีในที่นี้ <> คือสัญลักษณ์ \neq ในภาษา VB6

จากรูปที่ 2.10 เป็นการแยกกลุ่มสีโดยการดูรอบข้างจุดสีที่สนใจ (จุด X) ว่ามีการกำหนดกลุ่มสี (จุด P1, P2, P3, P4) อยู่แล้วบ้างหรือไม่ ถ้ามีก็จะตั้งจุดสี (จุด X) ให้เลขที่กลุ่มเท่ากับจุดสีรอบข้าง (จุด P1 หรือ P2 หรือ P3 หรือ P4) แต่ไม่จำเป็นต้องดูทั้ง 8 จุดรอบข้างเนื่อง จากไม่ให้เกิดเวลาในการประมวลผลมากเกินไป จึงเลือกที่จะตรวจสอบเฉพาะ 4 จุดมุมขนาบเท่านั้น เพราะได้ผลลัพธ์เหมือนกัน และถ้าพบว่าใน 4 จุดดังกล่าวไม่มีกลุ่มสี ก็จะตั้งกลุ่มใหม่ที่จุด X



รูปที่ 2.11 ปัญหาที่สามารถเกิดขึ้นกับวิธีแยกกลุ่มสีในรูปที่ 4.5 ทั้งที่วัตถุประสงค์ตั้งต้นเป็นชั้นเดียวกัน

จากประสบการณ์การทดสอบวิธีแยกกลุ่มสีแบบในรูปที่ 2.10 พบปัญหาซึ่งเกิดในรูปแบบคล้าย ๆ กับในรูปที่ 2.11 คือแทนที่จะได้กลุ่มสีออกมา 1 กลุ่ม แต่กลับได้กลุ่มสีออกมาเป็น 2 กลุ่ม ทำให้การนับกลุ่มและการหาจุดศูนย์กลางกลุ่มเกิดความผิดพลาด จึงได้เพิ่มวิธีการแยกกลุ่มสีโดยการรวมจุดศูนย์กลางของ 2 กลุ่มเข้าด้วยกัน และลบจุดศูนย์กลางกลุ่มหนึ่งทิ้งไปโดยการให้ค่าเท่ากับ 0 ตามขั้นตอนดังนี้

1. ถ้าพบว่าในตำแหน่ง 3x3 จุดที่สนใจ มีกลุ่มสีที่อยู่ในกรอบน้อยกว่าหรือเท่ากับ 1 กลุ่ม ให้ใช้วิธีการแยกสีแบบในรูปที่ 2.10 แต่ถ้าไม่เป็นตามเงื่อนไขนี้ก็ให้ทำข้อ 2 ต่อไป
2. ทำการรวมจุดศูนย์กลางของทั้ง 2 กลุ่มเข้าด้วยกันดังสมการ

$$\sum x_2 = \sum x_2 + \sum x_1$$

$$\sum y_2 = \sum y_2 + \sum y_1$$

$$N_2 = N_2 + N_1$$

และให้

$$\sum x_1 = 0$$

$$\sum y_1 = 0$$

$$N_1 = 0$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ $\sum X_2, \sum X_1$ คือ ผลรวมพิกัด X ของกลุ่มที่ 1 และกลุ่มที่ 2
 $\sum Y_2, \sum Y_1$ คือ ผลรวมพิกัด Y ของกลุ่มที่ 1 และกลุ่มที่ 2
 N_2, N_1 คือ จำนวนจุดสีของกลุ่มที่ 1 และกลุ่มที่ 2

3. เปลี่ยนกลุ่มสีที่ 1 ในแกน X ให้เป็นกลุ่ม 2 ให้หมด (การไม่เปลี่ยนกลุ่ม X ทั้งหมดให้เป็นกลุ่ม 2 นั้น เนื่องจากการเสียเวลาในการประมวลผลค่อนข้างมาก เพราะมีการแก้ความผิดพลาดของจุดศูนย์กลางไปแล้วในข้อ 2) โดยใช้โปรแกรมดังนี้

I = 1

Do While Matrix(X - I, Y) \neq 0

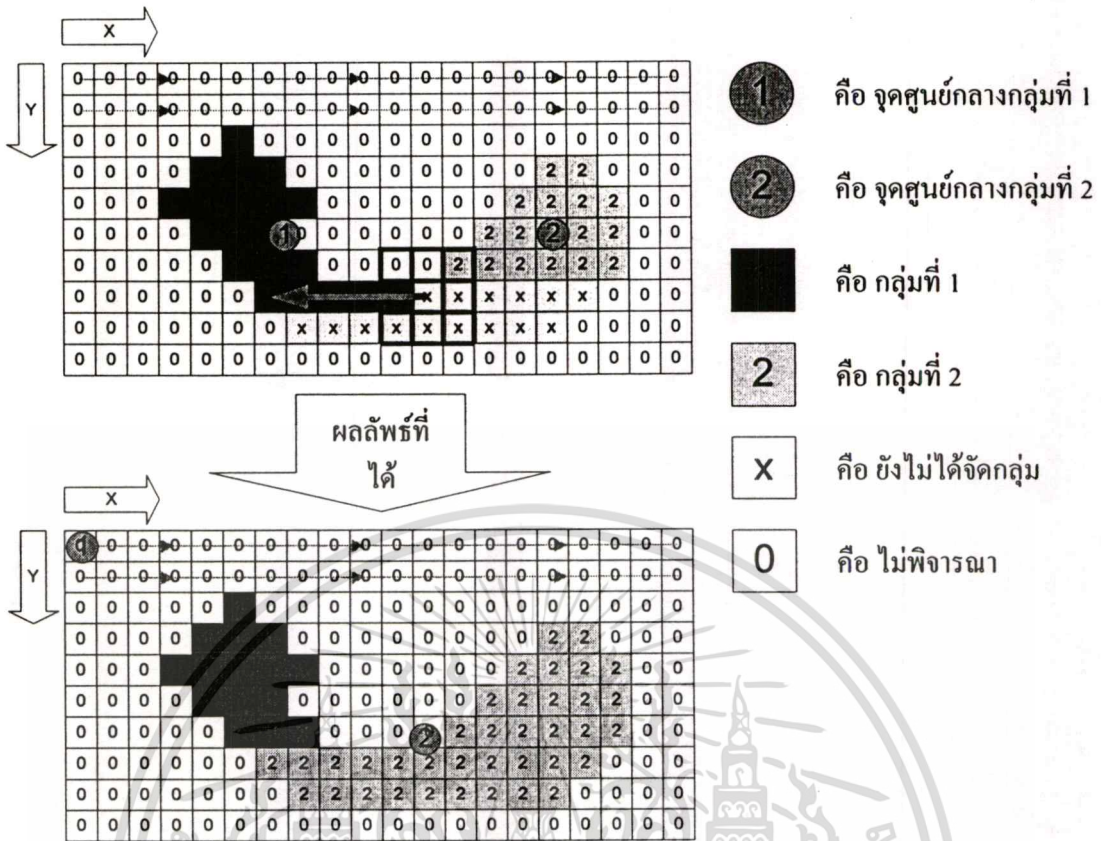
Matrix(X - I, Y) = Matrix(X, Y)

I = I + 1

Loop

จากโปรแกรมขั้นต้นนั้น จะทำงานโดยการตรวจสอบตำแหน่งก่อนจุดสนใจ 1 จุด (ตำแหน่ง X-1, Y) ว่าเป็นกลุ่มสีที่ 0 (กลุ่มสีพื้นหลัง) หรือไม่ ถ้าไม่เท่ากันก็จะให้กลุ่มสีตำแหน่งก่อนจุดสนใจเท่ากับกลุ่มสีของจุดสนใจ ต่อจากนั้นให้ขยับไปตรวจสอบตำแหน่งถัดไป (ตำแหน่ง X-2, Y) ตรวจสอบไปจนกว่าจะเจอกลุ่มสีที่ 0 (กลุ่มสีพื้นหลัง) สุดท้ายจะได้ผลลัพธ์ดังรูปที่ 2.12

อนึ่งในการคำนวณนี้ ประเด็นที่มุ่งเน้นก็คือการหาจุดศูนย์กลางมวลของแต่ละกลุ่มจึงไม่มีความจำเป็นต้องพิจารณาจุดสีในกลุ่มที่ 1 ดังรูปที่ 2.12 ทั้งหมดทุกจุด

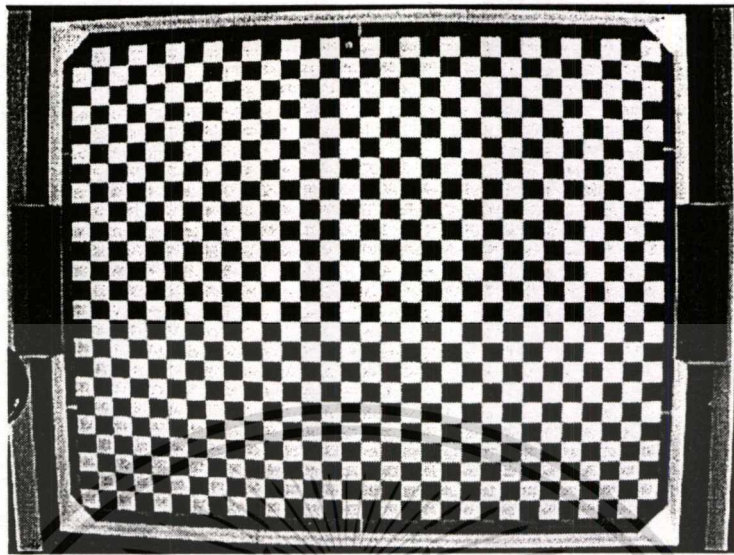


รูปที่ 2.12 การรวมจุดศูนย์กลางกลุ่มสีเข้าด้วยกัน

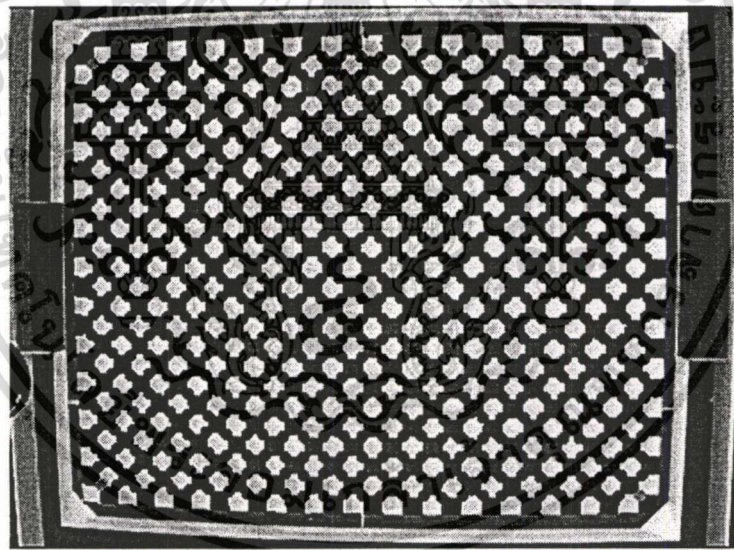
2.3 ความบิดเบือนภาพ (Barrel distortion)

2.3.1 การแก้ไขโดยใช้ตารางหมากรุก

วิธีนี้เป็นวิธีที่ใช้ง่าย แต่มีปัญหาในการวางตารางให้ทั่วสนามและใช้เวลาในการตั้งค่านาน เป็นการทำนายค่า (Interpolate) ระหว่างพิกัดจริง กับพิกัด บนหน้าจอคอมพิวเตอร์ ดังแสดงในรูปที่ 2.13 และ 2.14 เช่น วัฏระยะของมุมทั้ง 4 ด้านเทียบกับมุมสนาม ก็จะคำนวณจุดตัดทุกจุดได้เนื่องจาก ตารางมีขนาดเท่ากันทุกช่อง คือ 10 x 10 เซนติเมตร²



รูปที่ 2.13 การวางตารางลงบนสนาม



รูปที่ 2.14 การหาจุดตัดของตารางเพื่อหาพิกัด

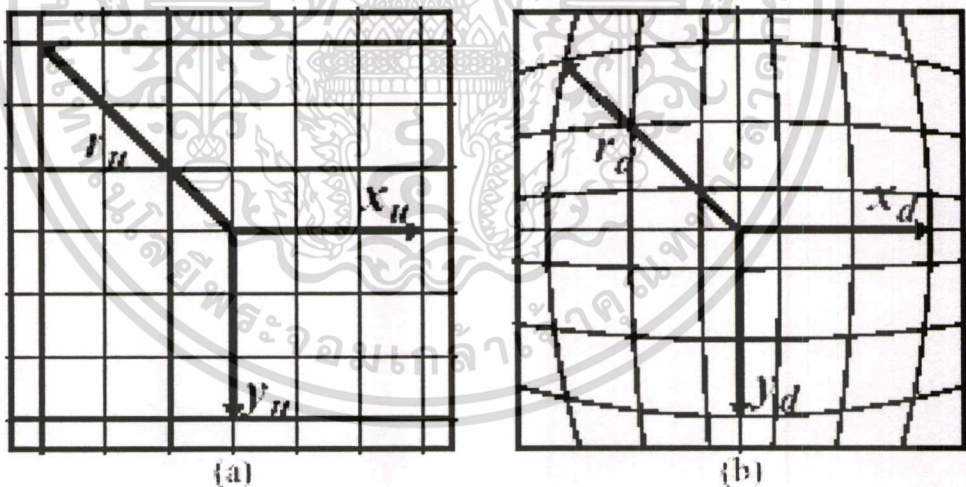
2.3.2 การแก้ไขโดยใช้สมการความบิดเบือน

ความบิดเบือนภาพเกิดขึ้นจากการลดลงของกำลังขยายของเลนส์ ด้วยเหตุที่ว่าค่าระยะห่างของแกนกับจุดบนภาพตามแนวรัศมีของจุดกึ่งกลางภาพเปลี่ยนไปจากเดิม ทำให้เกิดลักษณะของภาพที่เปลี่ยนแปลงไปในรูปแบบของถังน้ำมัน (Barrel) โดยสมการของความบิดเบือนแบบถังน้ำมัน [3] สามารถบรรยายได้ดังนี้

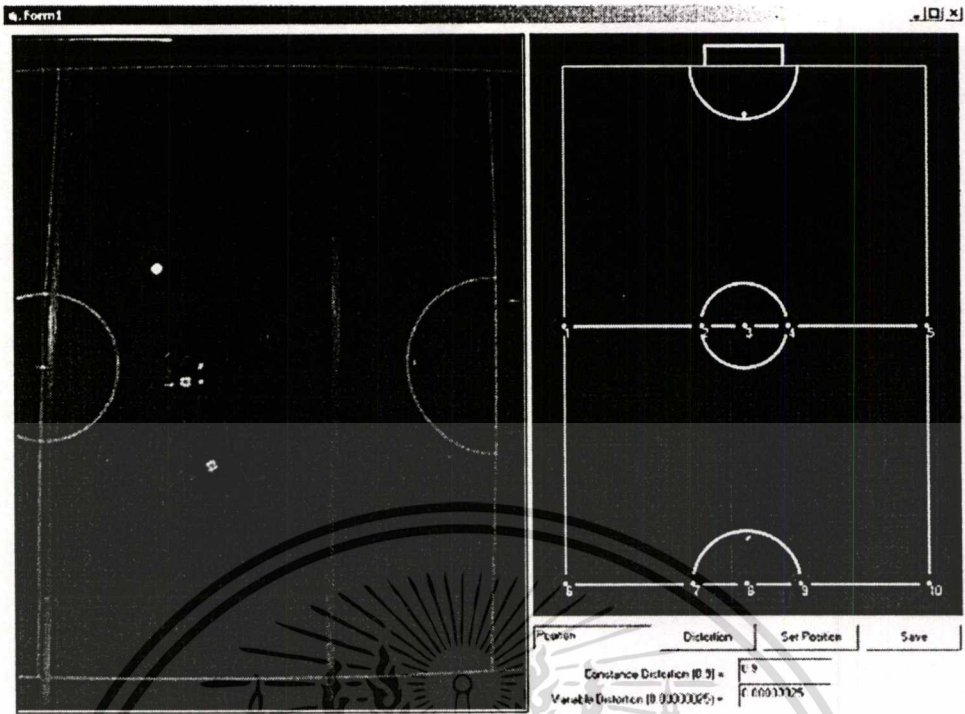
$$r_u = r_d(c + kr_d^2) \quad (2.1)$$

$$r_u = \sqrt{x_u^2 + y_u^2} \quad (2.2)$$

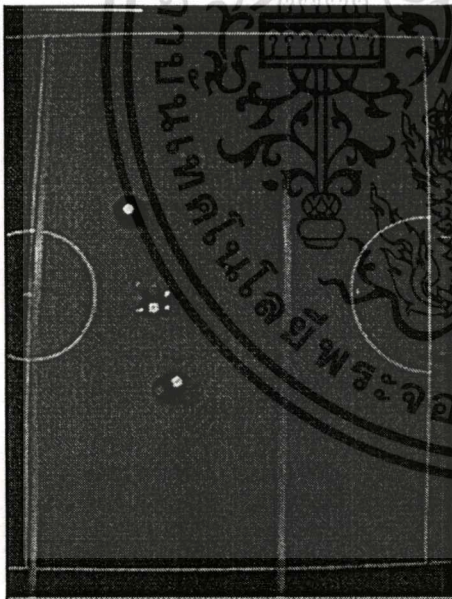
โดยที่ r_u และ r_d เป็นระยะห่างจุดที่สังเกตกับจุดศูนย์กลางของความบิดเบือน (ปรกติแล้วจะเป็นกึ่งกลางภาพ) ของภาพที่ไม่บิดเบือนและภาพที่บิดเบือนตามลำดับ ดังเช่นแสดงในรูปที่ 2.15 ค่า k คือตัวแปลของความบิดเบือน และค่า c คือตัวแปลของอัตราขยาย ซึ่งจะขึ้นอยู่กับชนิดของเลนส์ สมการ (2.1) และ (2.2) ใช้ในการปรับแก้ไขความบิดเบือนในวิทยานิพนธ์นี้ ซึ่งรูปที่ 2.16 และ 2.17 แสดงผลการแก้ไขโดยอาศัยสมการดังกล่าว จะสังเกตได้ว่าเส้นขอบสนามจากโค้งเปลี่ยนเป็นเส้นตรงโดยที่กำหนดให้ $c = 0.9$ และ $k = 0.00000025$



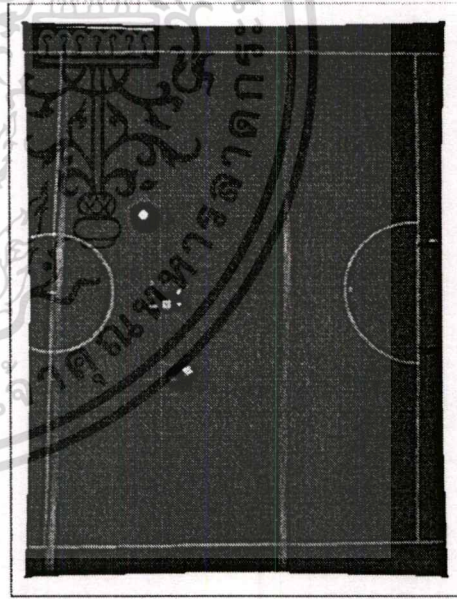
รูปที่ 2.15 ความบิดเบือนแบบถังน้ำมัน [3]



รูปที่ 2.16 ภาพจริงที่ได้จากกล้อง



$$r_u = r_d (c + kr_d^2)$$



$$r_u = r_d (0.9 + 0.00000025 r_d^2)$$

รูปที่ 2.17 ผลจากการแก้ความบิดเบือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

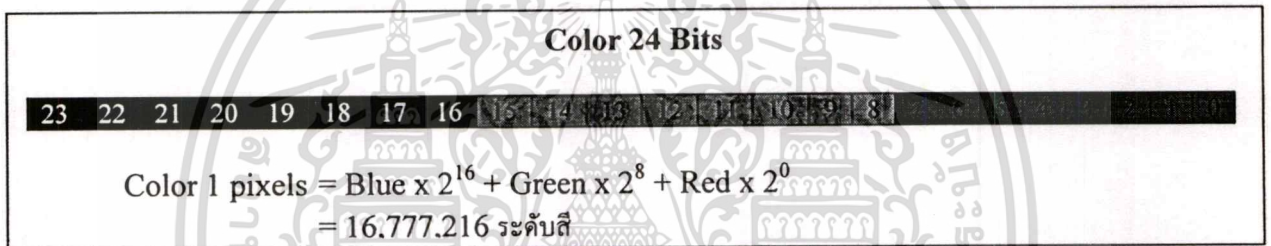
บทที่ 3

การออกแบบระบบการมอง

บทนี้นำทฤษฎีที่ได้กล่าวไว้ในบทที่ 2 มาใช้งานจริงโดยมุ่งเน้นวิธีการคิด (algorithm) เพื่อนำไปใช้ในบทที่ 4 สำหรับการทดลองและทดสอบประสิทธิภาพของวิธีการที่ได้ออกแบบ

3.1 การจับภาพ (Image Capture)

ภาพที่ได้จากคอมพิวเตอร์นั้น จะอยู่ในรูปตารางข้อมูล (matrix) ซึ่งขนาดของตารางข้อมูลขึ้นอยู่กับความละเอียดของภาพ ในที่นี้ใช้ภาพความละเอียด 320 x 240 จุด ก็จะได้ขนาดมิติของตารางข้อมูลเท่ากับ 320 x 240 ด้วยเช่นกัน และในสมาชิกของตารางข้อมูลแต่ละช่องมีค่าสีขนาด 24 บิต ซึ่งประกอบด้วยสีแดง 8 บิต สีเขียว 8 บิตและสีน้ำเงิน 8 บิต ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 ส่วนประกอบของสี 24 บิต

3.2 การแยกสี (Segmentation)

ในที่นี้ใช้มิติสี HSV ด้วยวิธี LUT(RGB to Color Code) โดยตั้ง Mid-Tolerance Threshold ไว้ใน LUT เนื่องจากเป็นวิธีที่เร็วที่สุด แต่ด้วยขนาดของสี RGB ที่ได้มีขนาด 24 บิต ทำให้ต้องสร้าง LUT ขนาด 16,777,216 ตำแหน่ง ซึ่งมากเกินไปจึงต้องลดความละเอียดของสีจาก 24 บิตเป็น 18 บิตคือ 262,144 ตำแหน่ง โดยในแต่ละสีจะใช้แค่ 6 บิตเท่านั้น

ในการแปลงค่าสีจาก 8 bits มาเป็น 6 bits เพื่อที่จะนำเข้าไปยัง LUT แบบ 3 มิติ ซึ่งขนาดของมิติเท่ากับ 6bits x 6bits x 6bits

Blue ก่อนเข้า Look Up Table = Blue ที่ได้จากจุดสี หารด้วย 4

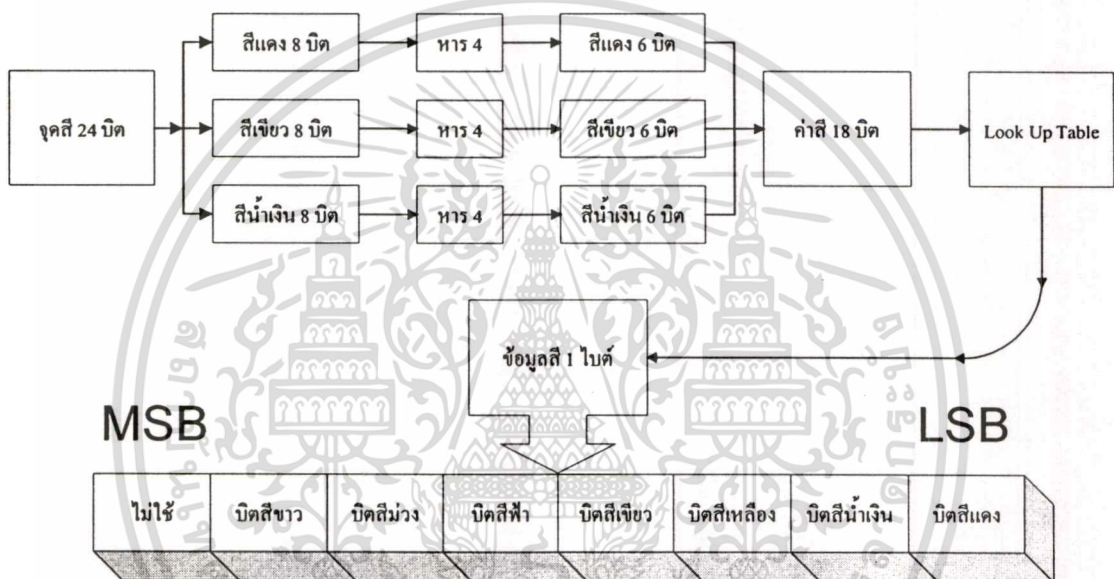
Green ก่อนเข้า Look Up Table = Green ที่ได้จากจุดสี หารด้วย 4

Red ก่อนเข้า Look Up Table = Red ที่ได้จากจุดสี หารด้วย 4

สาเหตุที่ต้องหารด้วย 4 เพราะเป็นการแปลงจากสี 24 บิตลดรูปเหลือ 18 บิต และเมื่อนำค่าสีไปเปรียบเทียบกับค่าสีใน LUT แล้วจะได้ว่าสีที่ได้คือสีของสิ่งใด ซึ่งค่าที่ได้นั้นมีขนาด 1 byte สามารถแจกแจงรายละเอียดได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บิตที่ 0 คือแสดงความเป็นสีแดงของลูกบอล
- บิตที่ 1 คือแสดงความเป็นสีน้ำเงิน
- บิตที่ 2 คือแสดงความเป็นสีเหลือง
- บิตที่ 3 คือแสดงความเป็นสีเขียว
- บิตที่ 4 คือแสดงความเป็นสีฟ้า
- บิตที่ 5 คือแสดงความเป็นสีม่วง
- บิตที่ 6 คือแสดงความเป็นสีขาว



รูปที่ 3.2 บล็อกไดอะแกรมการแปลงค่าสี

นำค่าที่ได้จาก LUT ของทุก Pixel ลงในหน่วยความจำเมทริกซ์ขนาดเท่ากับภาพที่ได้จากกล้อง โดยแบ่งค่าออกเป็น 6 ค่า คือ

- ค่า 0 แทนสีที่ไม่ใช่สีของหุ่นยนต์
- ค่า 1 แทนสีของลูกบอล (สีแดง-ส้ม)
- ค่า 2 แทนสีของหุ่นยนต์ทีมวิจัย (สีเหลือง หรือ สีน้ำเงิน)
- ค่า 3 แทนสีของหุ่นยนต์ทีมฝ่ายตรงข้าม (สีเหลือง หรือ สีน้ำเงิน)
- ค่า 4 แทนสีบอทิศทางสีที่ 1 (สีฟ้า)
- ค่า 5 แทนสีบอกรหัส (สีขาว)
- ค่า 6 แทนสีบอทิศทางสีที่ 2 (สีม่วง)

การหาจุดศูนย์กลางใช้สูตร

$$X_{center} = \frac{\sum X}{N}$$

$$Y_{center} = \frac{\sum Y}{N}$$

โดยที่ X คือ ตำแหน่งตามแนวแกน X, Y คือ ตำแหน่งตามแนวแกน Y, N คือ จำนวนจุดทั้งหมด

ถ้าพบว่ามีกลุ่มสี่มวงมากกว่า 2 จุดสี่ขึ้นไป ก็จะทำการหาจุดศูนย์กลางของกลุ่มสี่มวงนั้น เมื่อได้จุดศูนย์กลางของกลุ่มสี่มวงมาแล้วก็จะทำการหามุมระหว่างจุดสี่มวงกับจุดศูนย์กลางหุ่นยนต์ เพื่อจะหามุมของหุ่นยนต์เมื่อเทียบกับสนามดังรูปที่ 3.4 โดยใช้สูตร

$$Zeta = \tan^{-1} \left[\frac{(Y_2 - Y_1)}{(X_1 - X_2)} \right] \times \frac{180}{\pi} \quad \text{ใน quadrant ที่ 1}$$

$$Zeta = 180 + \tan^{-1} \left[\frac{(Y_2 - Y_1)}{(X_1 - X_2)} \right] \times \frac{180}{\pi} \quad \text{ใน quadrant ที่ 2}$$

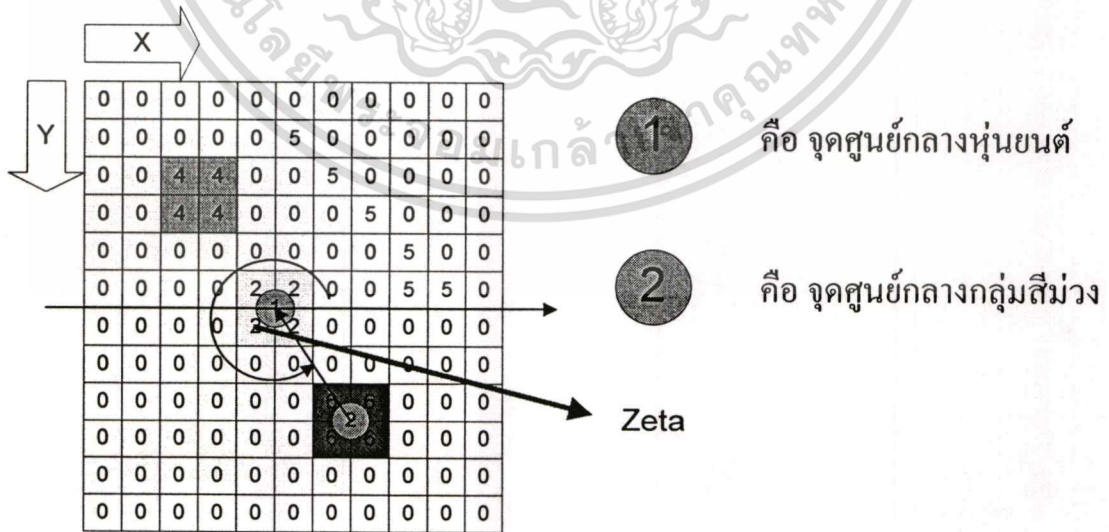
$$Zeta = 180 + \tan^{-1} \left[\frac{(Y_2 - Y_1)}{(X_1 - X_2)} \right] \times \frac{180}{\pi} \quad \text{ใน quadrant ที่ 3}$$

$$Zeta = 360 + \tan^{-1} \left[\frac{(Y_2 - Y_1)}{(X_1 - X_2)} \right] \times \frac{180}{\pi} \quad \text{ใน quadrant ที่ 4}$$

*Zeta แทน มุมที่ 1 ของหุ่นยนต์เปรียบเทียบกับสนาม

*X₁, Y₁ แทน พิกัด x, y ของหุ่นยนต์

*X₂, Y₂ แทน พิกัด x, y ของกลุ่มสี่มวง



Matrix memory

รูปที่ 3.4 การหามุมของหุ่นยนต์จากกลุ่มสี่มวง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากได้มุมของหุ่นยนต์มาแล้ว ก็จะทำการกวาด (Scan) รอบตำแหน่งของหุ่นยนต์อีก 1 รอบเพื่อหาจุดสีขาวยังซึ่งเป็นสีบอกรหัส (ID) ว่าเป็นหุ่นยนต์ตัวไหน โดยกวาดหาพิกัด 18x18 จุด รอบจุดศูนย์กลางหุ่นยนต์ ซึ่งเป็นขนาดเท่ากับหุ่นยนต์ในภาพที่กล้องถ่ายมาได้) โดยใช้หลักการเดียวกันกับการจัดกลุ่มสีทิมที่ได้กล่าวมาแล้วข้างต้น เมื่อได้กลุ่มสีขาวยังจะพิจารณาเฉพาะกลุ่มที่มีจำนวนจุดมากกว่า 10 จุดเท่านั้น (แถบสีขาวยังมีขนาดประมาณ 15-25 จุดขึ้นอยู่กับสัญญาณรบกวน) ซึ่งเป็นการตั้งค่าความไวต่อสัญญาณรบกวน จากนั้นจะพิจารณาว่ากลุ่มสีขาวยังที่ได้มานั้นอยู่บริเวณหน้าหรือหลังหุ่นยนต์ โดยใช้สมการดังนี้

If $N > 10$ Then

$$X_{center} = \frac{\sum X}{N} \quad (3.1)$$

$$Y_{center} = \frac{\sum Y}{N} \quad (3.2)$$

$$X'_{center} = (X_{center} - Robot_{width}) \times \left[-10 \cos\left(\frac{\theta \times \pi}{180^\circ}\right) \right] \quad (3.3)$$

$$Y'_{center} = (Y_{center} - Robot_{height}) \times \left[-10 \sin\left(\frac{\theta \times \pi}{180^\circ}\right) \right] \quad (3.4)$$

$$\text{If } |X'_{center} - Y'_{center}| < 100 \text{ Then} \quad (3.5)$$

$$\text{If } X'_{center} - Y'_{center} > 8 \text{ Then } ID_{front} = 1 \text{ กลุ่มสีขาวยังอยู่ด้านหน้า}$$

$$\text{If } X'_{center} - Y'_{center} < -8 \text{ Then } ID_{back} = 2 \text{ กลุ่มสีขาวยังอยู่ด้านหลัง}$$

End If

End If

หมายเหตุ พิกัดทั้งหมดนี้เทียบกับตำแหน่งของหุ่นยนต์ โดยที่

X คือ พิกัดของจุดสีขาวยังแกน X

Y คือ พิกัดของจุดสีขาวยังแกน Y

N คือ จำนวนจุดของกลุ่มสีขาวยัง

X_{center} คือ จุดศูนย์กลางของกลุ่มสีขาวยังในแนวแกน X

Y_{center} คือ จุดศูนย์กลางของกลุ่มสีขาวยังในแนวแกน Y

X'_{center} คือ จุดศูนย์กลางของกลุ่มสีขาวยังในแนวแกน X หลังจากแทนค่าในสมการหมุน (3.3)

เพื่อให้มุมของหุ่นยนต์เท่ากับ 0

Y'_{center} คือ จุดศูนย์กลางของกลุ่มสีขาวยังในแนวแกน Y หลังจากแทนค่าในสมการหมุน (3.4)

เพื่อให้มุมของหุ่นยนต์เท่ากับ 0

$Robot_{width}$ คือ ความกว้างของหุ่นยนต์เมื่อสังเกตจากกล้อง (ที่ใช้ใน โปรแกรมเท่ากับ 18 จุด)

$Robot_{height}$ คือ ความยาวของหุ่นยนต์เมื่อสังเกตจากกล้อง (ที่ใช้ใน โปรแกรมเท่ากับ 18 จุด)

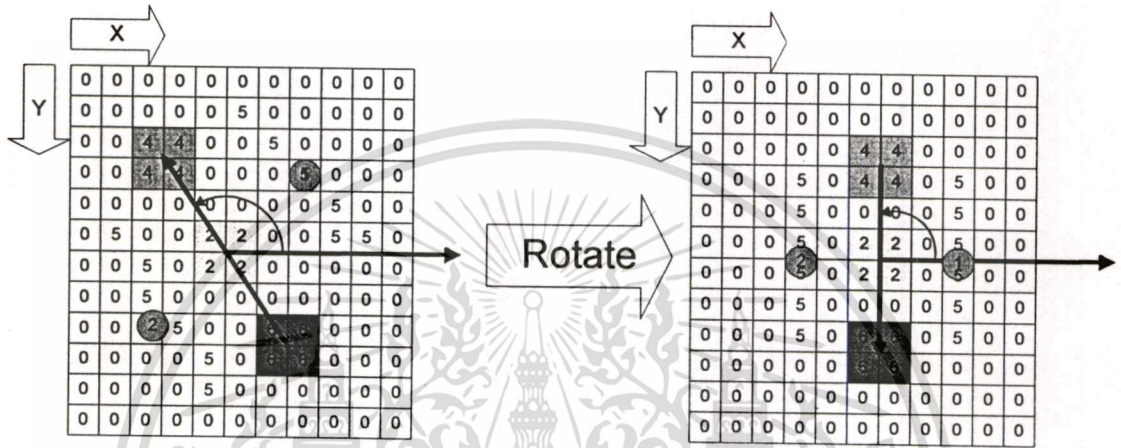
ID_{front} คือ รหัส (แถบสีขาวยัง) ของหุ่นยนต์ที่ติดด้านหน้าหุ่นยนต์

เอกสารนี้เป็นเอกสารทรัพย์สินทางปัญญาหรือการวิจัยเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ID_{back} คือ รหัส (แถบสีขาว) ของหุ่นยนต์ที่ติดด้านหลังหุ่นยนต์

เลข 10 ที่ได้ในสมการที่ (3.3) มาจากการขยายขนาดสเกลให้เห็นชัดขึ้น เครื่องหมายลบ แสดงการหมุนกลับทิศเพื่อให้แกนตั้งตรงเพื่อจะได้วิเคราะห์สะดวกขึ้นดังแสดงในรูปที่ 3.6 และ สำหรับสมการ (3.5) การตั้งค่าน้อยกว่า 100 มาจากการทดลองในสนามจริง โดยค่าคงที่ได้เป็นค่า หลังจากการขยายสเกล



Matrix memory

Matrix memory

1

คือ จุดศูนย์กลางแถบสีขาวด้านหน้า

2

คือ จุดศูนย์กลางแถบสีขาวด้านหลัง

รูปที่ 3.6 การหมุนมุมเพื่อง่ายต่อการหารหัสของหุ่นยนต์

เมื่อตรวจสอบกลุ่มสีขาวทุกกลุ่มที่อยู่บริเวณหุ่นยนต์เสร็จเรียบร้อยแล้ว ก็จะได้ค่า ID_{front} และ ID_{back} ออกมาเพื่อจะหารหัสหุ่นยนต์ โดยกำหนดให้

$$ID_{robot} = ID_{front} + ID_{back} + 1$$

สาเหตุที่บวกอีก 1 เพราะหุ่นยนต์หมายเลข 1 เป็นหุ่นยนต์ที่ไม่มีแถบ

เนื่องจากกลุ่มสีขาของหุ่นยนต์ตัวที่ 4 กับ 5 เหมือนกัน คือมีทั้งแถบด้านหน้าและด้านหลัง จึงต้องแยกกัน โดยใช้กลุ่มสีฟ้าช่วยในการช่วยแยกแยะ

If $ID_{robot} = 4$ Then

If ไม่มีกลุ่มสีฟ้าอยู่บนหุ่นยนต์ Then

$ID_{robot} = 5$

End If

End If

3.5 การหาพิกัดหุ่นยนต์ฝ่ายตรงข้าม

พิจารณาโดยเลือกกลุ่มสีของหุ่นยนต์ฝ่ายตรงข้าม แล้วทำการหาจุดศูนย์กลางของกลุ่มสี นั้น ๆ เท่านั้น ไม่ต้องหามุมและรหัสของตัวหุ่นเพราะไม่จำเป็นต้องทราบ โดยใช้สมการหาจุดศูนย์กลางมวลคือ

$$X_{center} = \frac{\sum X}{N}$$

$$Y_{center} = \frac{\sum Y}{N}$$

โดยที่ X คือ ตำแหน่งแกน X, Y คือ ตำแหน่งแกน Y, N คือ จำนวนจุดทั้งหมด

3.6 การหาลูกบอล (Ball Identification)

สุดท้ายคือการหาพิกัดของลูกบอล เนื่องจากว่าสีแดง-ส้ม มีค่าสีใกล้เคียงกับสีม่วง ทำให้สับสนในการหาลูกบอล บางครั้งคอมพิวเตอร์เข้าใจว่ามีลูกบอลอยู่บนตัวหุ่นยนต์และไม่หาลูกบอลที่จุดอื่นต่อไป จึงแก้ปัญหาโดยทำการหาพิกัดลูกบอลในอันดับท้ายสุด

บริเวณที่ติดกับลูกบอลต้องไม่มีสีม่วง ต้องไม่อยู่บริเวณทับกับหุ่นยนต์ และต้องมีจำนวนจุดไม่ต่ำกว่า 2 จุด เพราะป้องกันการตีความว่าเป็นสัญญาณรบกวนหากกำหนดค่าจุดสีเพียงจุดเดียวในการอ้างอิงโดยถ้าเกิดข้อแม้ขึ้นจะหากกลุ่มสีของลูกบอลต่อไปทันที

หลังจากที่จัดกลุ่มสีแดงส้มเรียบร้อยแล้ว (ใช้วิธีเดียวกับการจัดสีอื่น ๆ) ทำการตรวจสอบตามอัลกอริทึมข้างล่างนี้คือ

If Ball_N > 1 Then

X = Ball_X \ Ball_N *หาจุดศูนย์กลางของวงกลมสีแดง

Y = Ball_Y \ Ball_N *หาจุดศูนย์กลางของวงกลมสีส้ม

*และกำหนดค่าของพิกัดของศูนย์กลางสีแดง

If X > 3 And X < 315 And Y > 3 And Y < 236 Then

Magenta = 0 *จำนวนจุดสีม่วง

*ตรวจสอบแนวแกน X ของพิกัดลูกบอล 7 จุด ดูว่ามีสีม่วงหรือไม่

For Pointer = 0 To 6

*ถ้าพบจุดสีม่วงจะทำการนับจำนวนว่ามีกี่จุด

If MatrixColor(X - 3 + Pointer, Y) = 6 Then

Magenta = Magenta + 1 *จำนวนจุดสีม่วงบวก 1

End If

Next Pointer

*ตรวจสอบแนวแกน Y ของพิกัดลูกบอล 7 จุด ดูว่ามีสีม่วงหรือไม่

For Pointer = 0 To 6

If MatrixColor(X, Y - 3 + Pointer) = 6 Then *ถ้าพบจุดสีม่วง

Magenta = Magenta + 1 *จำนวนจุดสีม่วงบวก 1

End If

Next Pointer

End If

*ต้องตรวจสอบจุดสีม่วงน้อยกว่า 2 จุดเท่านั้นจึงจะพิจารณา

If Magenta < 2 Then

ตรวจสอบว่ามีการทับกันของพิกัดหุ่นยนต์ทั้งหมด 5 ตัวว่าอยู่ติดกับลูกบอลหรือไม่ ถ้ามากกว่า 7 ขึ้นไปแสดงว่า จุดสีแดงเป็นลูกบอลเพราะอยู่นอกตัวหุ่นยนต์

If (Abs(X - Robot_X (1)) > 7 Or Abs(Y - Robot_Y(1)) > 7) _

And (Abs(X - Robot_X (2)) > 7 Or Abs(Y - Robot_Y(2)) > 7) _

And (Abs(X - Robot_X (3)) > 7 Or Abs(Y - Robot_Y(3)) > 7) _

And (Abs(X - Robot_X (4)) > 7 Or Abs(Y - Robot_Y(4)) > 7) _

And (Abs(X - Robot_X (5)) > 7 Or Abs(Y - Robot_Y(5)) > 7) _

Then

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรวจสอบว่าเป็นกลุ่มสีเข้มแดงกลุ่มใหญ่ที่สุดหรือไม่ ถ้าใช่แสดงว่าเป็นลูกบอลเพราะว่าเป็นสีเดียวที่ใช้แทนลูกบอล

```
If Ball_N > Nums Then
```

```
  Nums = Ball_N
```

```
  Ball_X = X
```

```
  Ball_Y = Y
```

```
End If
```

```
End If
```

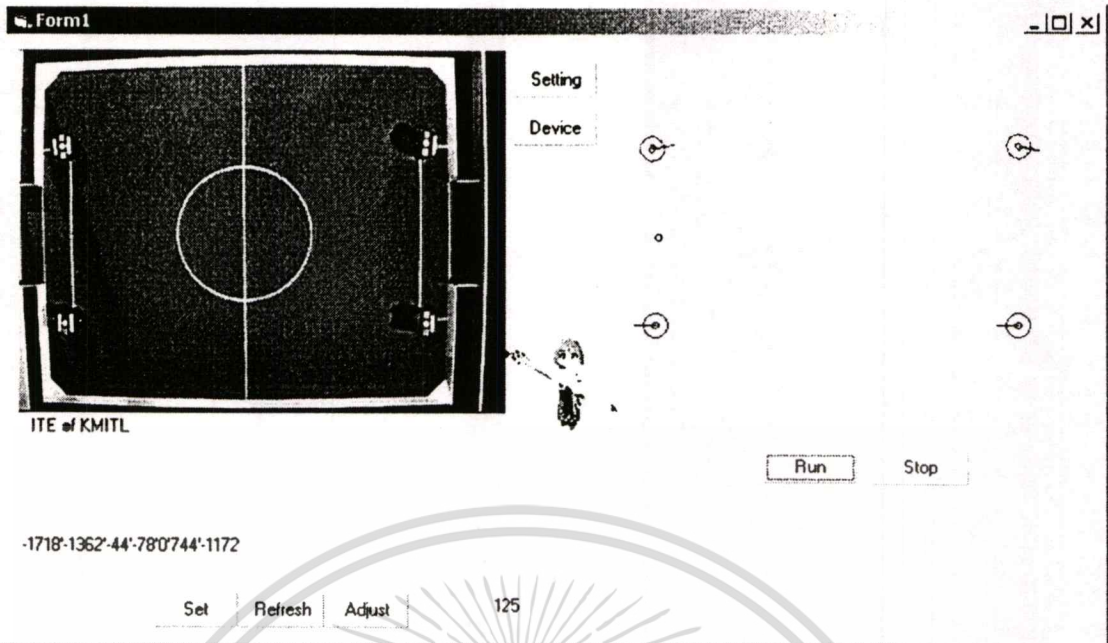
```
End If
```

```
End If
```

ทำอย่างเดิมวนจนกว่าจะครบจำนวนกลุ่มสีเข้มแดง จึงจะได้พิกัดลูกบอล (Ball_X, Ball_Y) อย่างแท้จริง รูปที่ 3.7 และ 3.8 แสดงผลของการไม่ใช้และใช้อัลกอริทึมที่บรรยายในหัวข้อนี้ตามลำดับ โดยจากรูปที่ 3.7 จะเห็นได้ว่าหุ่นยนต์ที่อยู่บนบด้านขวามีผลการตีความว่าแถบสีม่วงเป็นลูกบอล ซึ่งผิดจากสภาพที่เป็นจริง ในขณะที่รูปที่ 3.8 ลูกบอลสีแดงถูกระบุตำแหน่งได้ถูกต้อง



รูปที่ 3.7 โปรแกรมขณะพบปัญหาเกี่ยวกับสีของลูกบอลไปทับกับสีม่วง(ไม่ใช้อัลกอริทึมในการหาบอล)



รูปที่ 3.8 โปรแกรมขณะทำงาน(ใช้อัลกอริทึมในการหาบอล)

3.7 การแก้ไขความบิดเบือนภาพ

ทำได้โดยเลือกใช้วิธีในหัวข้อ 2.3.2 คือ การแก้ไขโดยใช้สมการความบิดเบือน (Barrel Distortion) เพราะง่ายต่อการตั้งค่าเมื่อสภาพแวดล้อมเปลี่ยนไป

3.8 การส่งข้อมูลระหว่างระบบปัญญาประดิษฐ์ (AI) กับระบบประมวลภาพ (IMAGE)

เนื่องจากคอมพิวเตอร์ระบบปัญญาประดิษฐ์ (AI) กับระบบประมวลภาพ (IMAGE) ไม่ใช่เครื่องเดียวกัน จึงต้องมีการส่งข้อมูลระหว่างคอมพิวเตอร์เกิดขึ้น และข้อมูลในรูปแบบภาษามนุษย์ที่สามารถเข้าใจได้ที่ส่งมีดังนี้

```

"#Robot1" , X , Y , Z , Identify,
"#Robot2" , X , Y , Z , Identify,
"#Robot3" , X , Y , Z , Identify,
"#Robot4" , X , Y , Z , Identify,
"#Robot5" , X , Y , Z , Identify,
"#OBSTACLE1" , X , Y , Identify,
"#OBSTACLE2" , X , Y , Identify,
"#OBSTACLE3" , X , Y , Identify,
"#OBSTACLE4" , X , Y , Identify,
"#OBSTACLE5" , X , Y , Identify,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“#BALL” , X , Y , Identify,

ทั้งหมดนี้ส่งแบบตัวอักษร ซึ่งใช้ “#” เป็นตัวเริ่มการส่งข้อมูลโดยที่

X คือ ค่าทางแกน X

Y คือ ค่าทางแกน Y

Z คือ ค่ามุม

Identify คือ ค่าที่ระบุว่าหาหุ่นยนต์หรือบอลพบหรือไม่ ถ้าเท่ากับ 1 แสดงว่าพบและเท่ากับ 0 แสดงว่าไม่พบ

วิธีการที่นำเสนอในบทนี้ เป็นวิธีการที่ได้จากการปฏิบัติและปรับแต่งค่าพารามิเตอร์จากภาคสนาม ซึ่งจะนำวิธีการเหล่านี้ไปใช้ในบทที่ 4 ต่อไป



บทที่ 4

ผลการทดลอง

ผลการทดลองในบทที่ 4 นี้อาศัยองค์ความรู้ที่ได้กล่าวไว้ในบทที่ผ่านมา บทนี้เริ่มจากการพิจารณาอุปกรณ์ที่ใช้ในการทดลอง ทดสอบการประมวลผลภาพและทดสอบความถูกต้องของอัลกอริทึมที่ได้จากการทดสอบจริงในภาคสนามตามลำดับ

4.1 อุปกรณ์ที่ใช้ในการทดลอง

อุปกรณ์ที่ใช้ในการทดลองมีดังนี้

1. เครื่องคอมพิวเตอร์ 1 เครื่อง

- หน่วยประมวลผลกลาง (CPU) ของ Intel Pentium4 2.8 GHz Bus 800 MHz Hyper-Threading
- หน่วยความจำ (RAM) ขนาด 512 Mbyte
- แผ่นแสดงผล (VGA-Card) ของ Power Color รุ่น ATi Radeon 9600 128Mb RAM
- ฮาร์ดดิสก์ (Hard Disk) ของ Maxtor ขนาด 80 Gbyte
- จอภาพ (Monitor) ของ Samsung รุ่น SyncMaster152s TFT ขนาด 15 นิ้ว

2. แผ่นรับสัญญาณภาพ (Capture Card) ของ Winfast รุ่น TV2000XP Deluxe

3. สายสัญญาณแบบ S-Video ความยาว 5 เมตร

4. กล้องวิดีโอ ของ Panasonic รุ่น NV-GS200

5. เลนส์อัตรากาขยาย 0.7 เท่า ของ Panasonic รุ่น VW-LW3707M3

6. สนามมาตรฐานการแข่งขันหุ่นยนต์เตะฟุตบอล (Robocup) ปี ค.ศ. 2003-2004

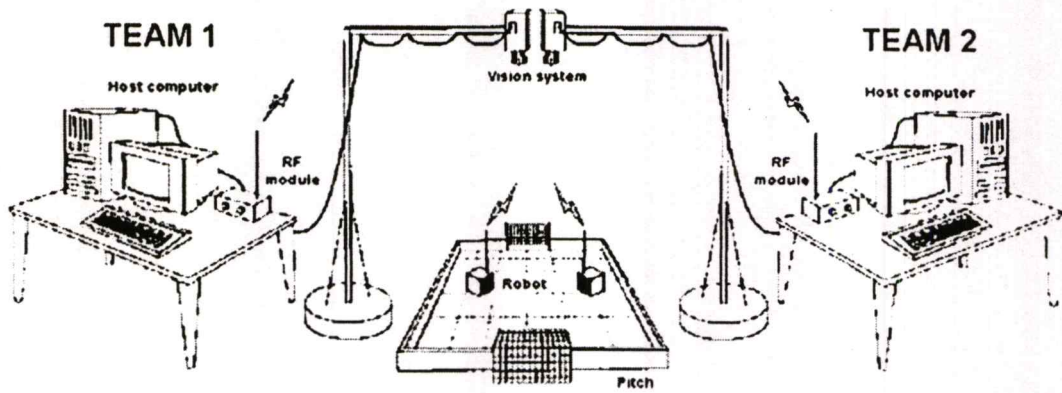
7. ตัวหุ่นยนต์จากทีม ITE-Gold1 ปี ค.ศ. 2004 ของ ภาควิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

8. ระบบปฏิบัติการ ของ Microsoft รุ่น Microsoft Windows XP Professional

9. โปรแกรมเขียนด้วย Microsoft Visual Basic version 6.0

10. โปรแกรมช่วยปรับแต่งตั้งค่าเริ่มต้น (Calibration) ของ Adobe รุ่น Adobe PhotoShop version 6.0

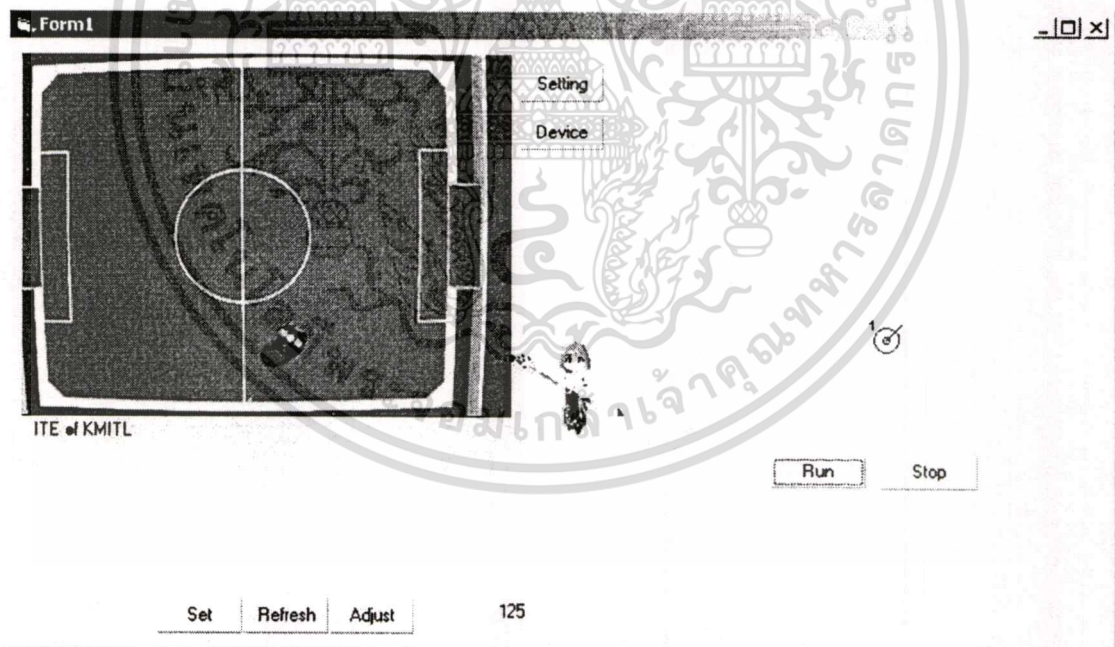
โดยการติดตั้งอุปกรณ์เหล่านี้แสดงในรูปที่ 4.1



รูปที่ 4.1 การติดตั้งอุปกรณ์ในการแข่งขันจริง

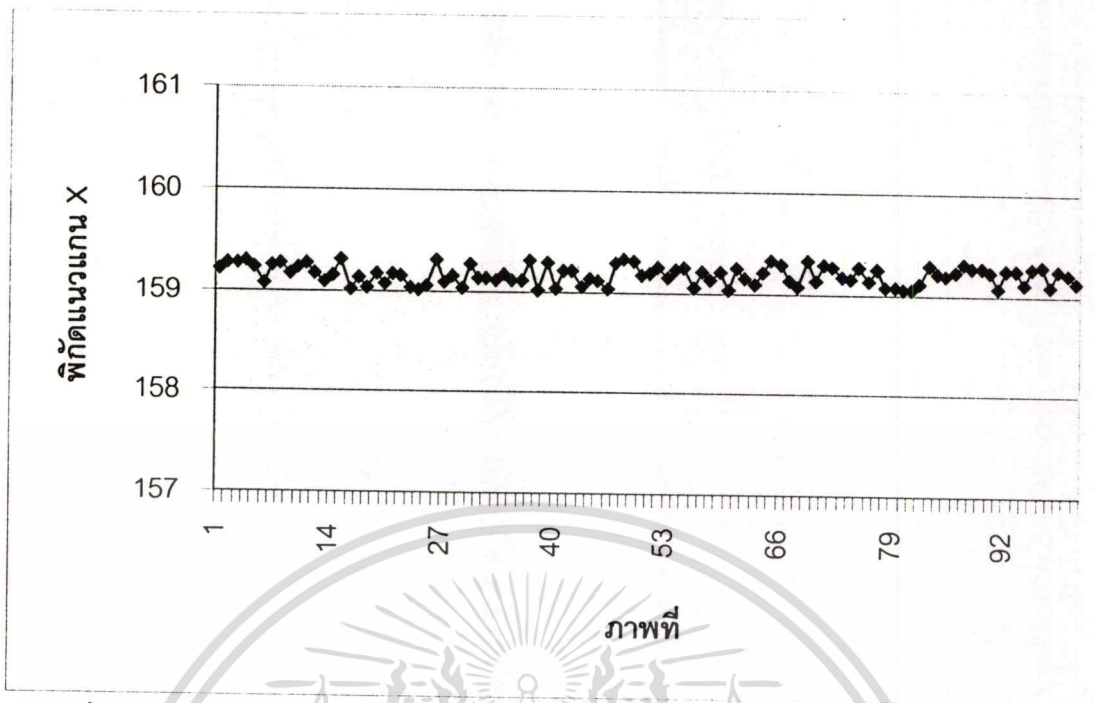
4.2 ผลการทดลองประมวลผลภาพ

ผลการทดลองในรูปที่ 4.2, 4.3, 4.4 และ 4.5 เป็นการทดลองนำหุ่นยนต์เบอร์ 1 ตั้งไว้ที่ตำแหน่งพิกัด แกน x เท่ากับ 159 แกน y เท่ากับ 200 และทำมุม θ กับสนาม เท่ากับ 45° วัดค่าพิกัดด้วยสายวัดในสนามจริง เทียบกับพิกัดที่ได้จากระบบการมอง หน่วยเป็นเซนติเมตร



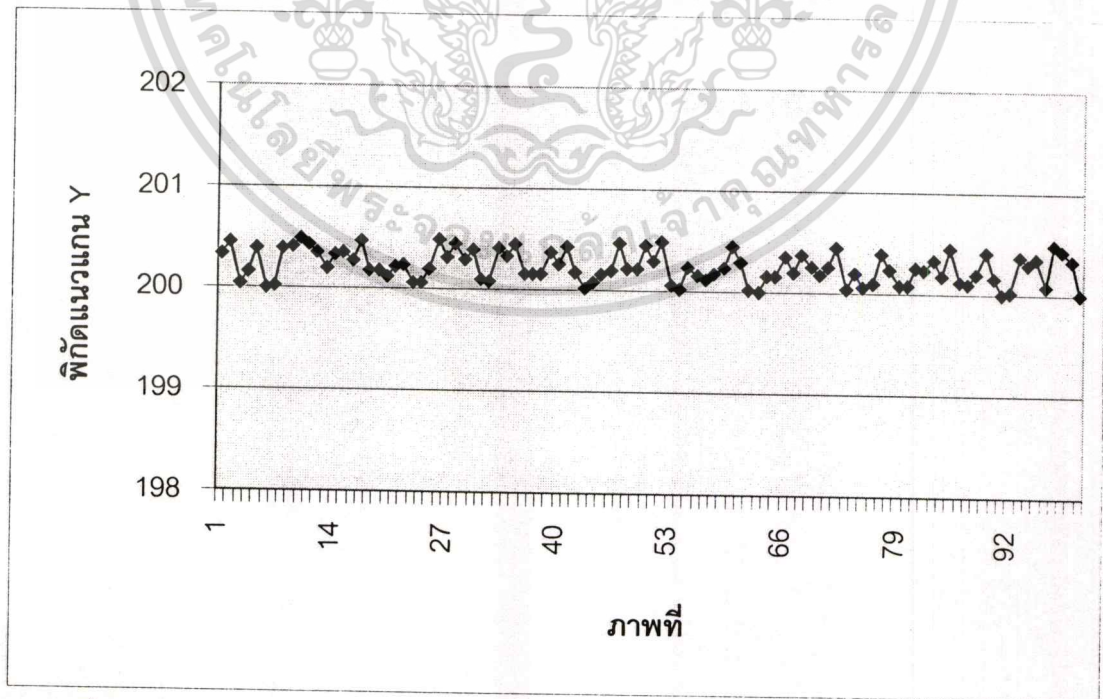
รูปที่ 4.2 การทดลองโดยใช้โปรแกรมที่ผู้วิจัยสร้างขึ้น ตามหลักการในบทที่ 3 วางหุ่นยนต์ ณ พิกัดจริงเท่ากับ (159,200) ของสนาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 พิกัดแนวแกน x ของหุ่นยนต์ใน 100 ภาพ ณ พิกัดจริงเท่ากับ (159,200) ของสนาม

จากรูป 4.2 จะเห็นว่าพิกัดในแนวแกน x ผิดพลาดไปประมาณ 0.2 เซนติเมตร ในสภาวะใช้งานจริง ซึ่งถือว่าใช้งานได้เพราะหน่วยที่ใช้ในการส่งข้อมูลเป็นหน่วยเซนติเมตร ดังนั้น 159.2 จึงปัดเศษเหลือ 159 เซนติเมตร ซึ่งอยู่ในพิสัยที่ยอมรับได้ เนื่องจากเมื่อเปรียบเทียบกับขนาดของบอลซึ่งมีเส้นผ่านศูนย์กลางขนาด 4 เซนติเมตร จึงถือได้อย่างจากหลักพิจารณา

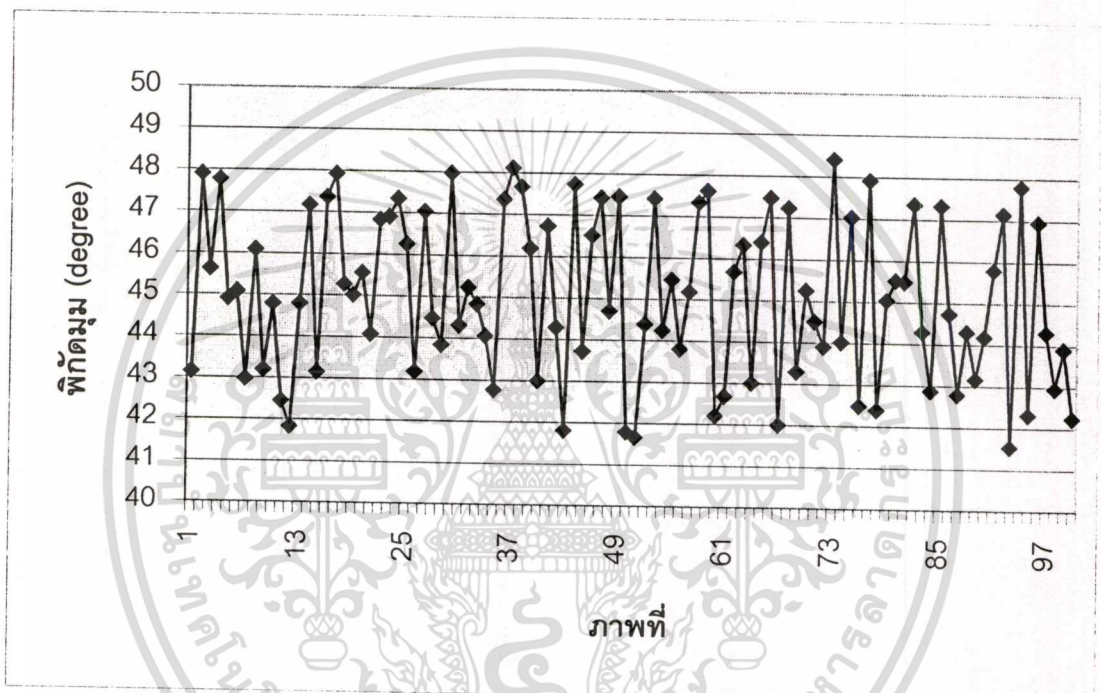


รูปที่ 4.4 พิกัดแนวแกน y ของหุ่นยนต์ใน 100 ภาพ ณ พิกัดจริงเท่ากับ (159,200) ของสนาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 4.3 จะเห็นได้ว่าพิกัดในแนวแกน y มีความผิดพลาดมากกว่าแนวแกน x เล็กน้อย มีค่าประมาณ 0.4 เซนติเมตร ในสภาวะใช้งานจริง แต่ยังคงถือว่าใช้งานได้เพราะหน่วยที่ใช้ในการส่งข้อมูลเป็นหน่วยเซนติเมตร ซึ่ง 200.4 บัดเศษเหลือ 200 เซนติเมตร

แต่ผลลัพธ์พิกัดมุม θ ในรูป 4.4 กลับไม่ค่อยดีนัก จะเห็นได้ว่าค่าที่ได้แกว่งพอสมควร คือ $\pm 3^\circ$ ซึ่งอาจจะเป็นเพราะความละเอียดของภาพน้อย จึงทำให้ระยะห่างระหว่างสีฟ้ากับสีม่วงน้อยเกินไป การประมวลผลจากหัวข้อ 3.2 จึงมีความผิดพลาดเกิดขึ้น (สีฟ้ากับสีม่วงเป็นสีที่ใช้คำนวณมุมของหุ่นยนต์)



รูปที่ 4.5 พิกัดมุม θ ของหุ่นยนต์ใน 100 ภาพ ณ พิกัดมุมจริงเท่ากับ 45

ตารางที่ 4.1 คือผลลัพธ์ที่ได้จากการแก้ความบิดเบือน (จากหัวข้อ 3.7) ของภาพแล้ว จะเห็นได้ว่ามีความผิดพลาดจากพิกัดเดิมเล็กน้อยที่ขอบภาพ ประมาณ 1-2 เซนติเมตรในแกน x และ y ซึ่งยังยอมรับได้ (2 ใน 320 เซนติเมตร ถือเป็น 0.625%)

ตารางที่ 4.1 ค่าความผิดพลาดหลังการแก้ Distortion แล้ว

ค่าความผิดพลาด (x, y) ของพิกัดภาพกับพิกัดจริงในสนาม

		ที่พิกัดแนวแกน x (cm)								
		-160	-120	-80	-40	0	40	80	120	160
ที่พิกัดแนวแกน y (cm)	-120	(1,1)	(1,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(1,0)	(1,1)
	-80	(1,0)	(1,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(1,0)	(1,0)
	-40	(1,0)	(1,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(1,0)	(1,0)
	0	(1,0)	(1,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(1,0)	(1,0)
	40	(1,0)	(1,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(1,0)	(1,0)
	80	(1,0)	(1,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(1,0)	(1,0)
	120	(1,2)	(1,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(1,0)	(1,2)

การทดลองต่อไปจะดูถึงความเร็วในการประมวลผล จะเปรียบเทียบการใช้ LUT กับการใช้
ไม่ใช่ แต่ในตาราง 4.2 จะเห็นว่ามีการส่งข้อมูลทาง RS232 มาเกี่ยวข้องด้วย ซึ่ง RS232 ใช้ความเร็ว
การส่ง 115200 บิตต่อวินาที และส่งข้อมูลขนาด 120 ไบต์ต่อภาพ ทั้งนี้เพราะในการใช้งานจริง
คอมพิวเตอร์ต้องมีภาระหน้าที่ในการส่งข้อมูลไปยังคอมพิวเตอร์เครื่องอื่นด้วย

ตารางที่ 4.2 ค่าความเร็วในการประมวลผลภาพ

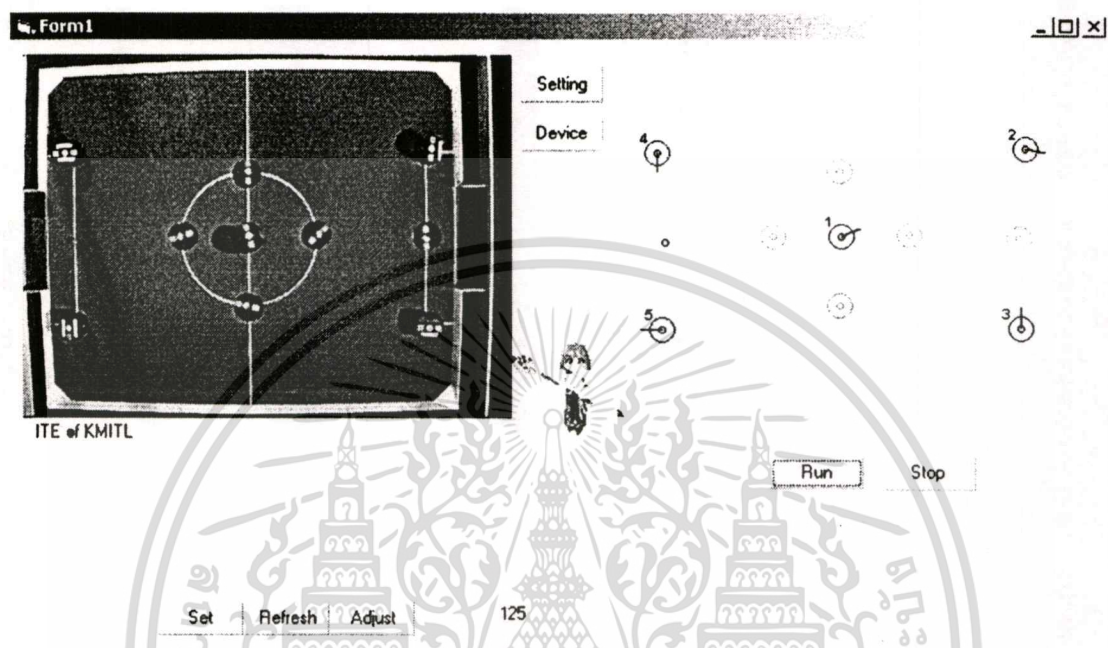
ความเร็วประมวลผลภาพต่อวินาที

	ใช้ LUT	ไม่ใช่ LUT
ความเร็วในการประมวลผลภาพ + การส่งข้อมูลทาง RS232	25.6 fps	12.5 fps
ความเร็วในการประมวลผลภาพ	33.1 fps	15 fps

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ทดสอบความถูกต้องของอัลกอริทึมจากการทดสอบจริงในภาคสนาม

การทดลองสุดท้ายในรูปแบบที่ 4.6 นี้ทำโดยการวางหุ่นยนต์และลูกบอลตามจุดต่าง ๆ ในสนามตามจำนวนที่กติกากำหนดการแข่งขันหุ่นยนต์เตะฟุตบอลกำหนดในสนามจริง การทดลองนี้เป็นการทดสอบภาพรวมของระบบและอัลกอริทึมทั้งหมดที่ได้ออกแบบและใช้ในงานวิจัยนี้



รูปที่ 4.6 การทดลองจริงในภาคสนามเพื่อตรวจสอบอัลกอริทึมที่ใช้โดยองค์รวม

ผลที่ได้จากข้อมูลที่ประมวลผลได้ในแบบซึ่งวางข้อกำหนดเพื่อส่งผลการประมวลผลภาพไปให้กับระบบ AI ดังแสดงตามหัวข้อ 3.8 มีดังนี้

```

"#Robot1", 145, 120, 22, 1, "#Robot2", 267, 60, 354, 1,
"#Robot3", 267, 180, 89, 1, "#Robot4", 23, 60, 271, 1,
"#Robot5", 23, 180, 182, 1,
"#OBSTACLE1", 145, 75, 1, "#OBSTACLE2", 100, 120, 1,
"#OBSTACLE3", 190, 120, 1, "#OBSTACLE4", 267, 120, 1,
"#OBSTACLE5", 145, 165, 1,
"#BALL", 23, 120, 1,

```

ซึ่งผลที่ได้ตรงกับตำแหน่งจริงในสนามทั้งหมด ซึ่งดูจากจอแสดงภาพด้านขวาในรูปแบบที่ 4.6

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

วิทยานิพนธ์นี้นำเสนอวิธีการประมวลผลภาพโดยการกำหนดพิกัดของหุ่นยนต์ โดยพบว่าการมองในเวลาจริงผลลัพธ์ที่ติดตามแนวแกน x และแกน y สามารถระบุได้ดี มีความคลาดเคลื่อนน้อย แม้ว่าพิกัดมุมจะให้ผลที่ยังไม่สมบูรณ์ นอกจากนี้การแก้ไขความบิดเบือนของภาพโดยใช้สมการความบิดเบือนให้ผลการแก้ไขได้ดี อีกทั้งระบบที่ใช้อัลกอริทึมที่นำเสนอในงานวิจัยนี้ทำให้ความเร็วในการประมวลผลเร็วขึ้นกล่าวคือ มีความเร็วมากกว่าภาพที่ได้จากกล้องจึงถือว่าสามารถนำภาพทุกภาพมาประมวลผลได้ครบถ้วนในเวลาจริง

5.2 ข้อเสนอแนะในการวิจัย

- หากต้องการจะเพิ่มความเร็วในการประมวลผลให้เร็วกว่าผลที่ได้ในงานวิจัยนี้ ต้องเปลี่ยนตัวแปลภาษาใหม่ (Compiler) เป็น Microsoft Visual C++ เพราะ Microsoft Visual C++ ไม่ต้องมีโปรแกรมช่วยทำงาน (run time) จึงทำงานได้รวดเร็วกว่า และมีผู้ร่วมพัฒนาเครื่องมือ (tools) มากกว่า
- การเปลี่ยนความละเอียดในการประมวลผลจาก 320×240 มาเป็น 640×480 จะทำให้ความผิดพลาดในการหาพิกัดมุมน้อยลง แต่ความเร็วในการประมวลผลจะตกลงด้วยเช่นกัน
- การเพิ่มจำนวนกล้องจาก 1 ตัวให้เป็น 3 ตัว จะช่วยแก้ปัญหาขณะที่หุ่นยนต์เลี้ยงลูกบอลอยู่ริมสนาม ทำให้มองเห็นตำแหน่งของลูกบอลหายไป เนื่องจากจุดบอดของการใช้กล้องเพียงตัวเดียว

เอกสารอ้างอิง

- [1] <http://www.robocup.org>
- [2] A.C. Hurlbert, "The computation of Color", **MIT Artificial Intelligence Laboratory Technical**, Report 1154, <ftp://publications.ai.mit.edu/ai-publications/1000-1499/AITR-1154.ps> .
- [3] D.G. Bailey, "A new approach to lens distortion correction", **Proceedings Image and Vision Computing New Zealand 2002**, pp 59-64.
- [4] <http://www.fifa.com/>
- [5] <http://www.fira.net/>
- [6] <http://www.robocup2002.org/>
- [7] A. Vijitanasarn, B. Purahong, P. Sooraksa, and G. Chen, "A Fuzzy Control for Soccer Robots", **FIRA Robot World Congress 2002**, pp 127-130.
- [8] C. F. Marques and P. U. Lima, "Vision-based self-localization for soccer robots", **IEEE/RSJ International Conference 2000**, pp 1193-1198.
- [9] <http://www.robocup.org/regulations/4.html>
- [10] R.W.G. Hunt, "The Specification of Colour Appearance: I. Concepts and Terms", **Color Research and Application 1977**, pp 55-68.
- [11] R.W.G. Hunt, "The Specification of Colour Appearance: II. Effects of Changes in Viewing Conditions", **Color Research and Application 1977**, pp 109-120.
- [12] M. Simon, S. Behnke, R. Rojas, "Robust Real Time Color Tracking", **RoboCup-2000: Robot Soccer World Cup IV**, pp 239-248.
- [13] K.J. Yoon and I.S. Kweon, "Landmark design and real-time landmark tracking for mobile robot localization", **SPIE --The International Society for Optical Engineering 2002**, pp 219-226.
- [14] P. Kornprobst and G. Medioni, "Tracking segmented objects using tensor voting", **In Proc. IEEE Conf. on Computer Vision and Pattern Recognition 2000**, pp 118-125.
- [15] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking", **In Proc. European Conference on Computer Vision 2002**, pp 661-675.
- [16] H.D.Cheng, X.H. Jiang, Y.Sun and J`Wang, "Color image segmentation: advance and prospects", **Pattern Recognition 2001**, pp 2259-2281.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [17] J. Bruce, T. Balch and M. Veloso, "Fast and Inexpensive Color Image Segmentation for Interactive Robots", **Proc. 2000 IEE/RSJ Int. Conf. On Intelligent Robots and Systems**, pp 2061-2066.
- [18] M. Simon, S. Behnke and R. Rojas, "Robust Real Time Color Tracking", **RoboCup 2000 Book**, pp 239-248.
- [19] S. Hibino, Y. Kodama and Y. Nagasaka, "Fast Image Processing and Flexible Path Generation System for RoboCup Small Size League", **RoboCup 2002 Book**, pp 53-64.
- [20] T.J. Rubio, A. Bandera, C. Urdiales and F. Sandoval, "A hierarchical context-based textured image segmentation algorithm for aerial images", **Texture2002**, <http://www.cce.hw.ac.uk/~texture2002/>.
- [21] J. Bruce, T. Balch and M. Veloso, "Fast and inexpensive color image segmentation for interactive robots", **In Proc. of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '00)**, pp 2061-2066.
- [22] P. Doubek, T. Svoboda and L. Van Gool, "Monkeys - a Software Architecture for ViRoom - Low-Cost Multicamera System", **ICVS 2003**, pp 386-395.
- [23] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale and S. Shafer, "Multi-Camera Multi-Person Tracking for EasyLiving", **International Workshop on Visual Surveillance**, pp 3-10.
- [24] K. Nummiaro, E. Koller-Meier and L. Van Gool, "An Adaptive Color-Based Particle Filter", **Journal of Image and Vision Computing 2003**, pp 99-110.
- [25] P. Pérez, C. Hue, J. Vermaak and M. Gangnet, "Color-Based Probabilistic Tracking", **ECCV 2002**, pp 661-675.
- [26] T. Svoboda, H. Hug and L. Van Gool, "ViRoom - Low Cost Synchronised Multi-camera System and its Self-Calibration", **DAGM 2002**, pp 515-522.
- [27] N.O. Khessal, Y.N. Maung, E.N.B. Hwee, S.O. Pyi and L.H.S. Antony, "Vision-based autonomous soccer robots", **TENCON 2000**, pp 207 -212.
- [28] G.F. Tong, W.D. Xiao and X.H. Xu, "Design of RoboCup robot soccer global vision system", **Proceedings of the 3rd World Congress on Intelligent Control and Automation 2000**, pp 212-214.

- [29] C.F. Marques and P.U. Lima, "Vision-based self-localization for soccer robots", **Proceedings of 2000 IEEE/RSJ International Conference on Intelligent Robotics and Systems**, pp 1193-1198.
- [30] C.S. Hong, S.M. Chun, J.S. Lee and K.S. Hong, "A vision-guided object tracking and prediction algorithm for soccer robots", **Proceedings of 1997 IEEE International Conference on Robotics and Automation**, pp 346-351.
- [31] A. Tremeau and N. Borel, "A region growing and merging algorithm to color segmentation", **Pattern Recognition 1997**, pp 1191-1203.
- [32] H.D. Cheng, X.H. Jiang and J. Wang, "Color image segmentation based on homogram thresholding and region merging", **Pattern Recognition 2002**, pp 373-393.
- [33] J. Fan, D. Yau, A. Elmagarmid and W. Aref, "Automatic image segmentation by integrating color-edge extraction and seeded region growing", **IEEE Trans. on Image Processing 2001**, pp 1454-1466.
- [34] N. Ikonomakis, K. Plataniotis, and A. Venetsanopoulos, "Unsupervised seed determination for a region based color image segmentation scheme", In **IEEE Int. Conf. Image Processing 2000**, pp 537-540.
- [35] <http://robocup.mae.cornell.edu/>
- [36] <http://www.araa.asn.au/acra/acra2004/papers/ball.pdf>
- [37] <http://www.cs.cmu.edu/~robosoccer/small/>
- [38] <http://www.cs.fit.edu/wds/classes/cse5255/cse5255/davis/text.html#colormodels>

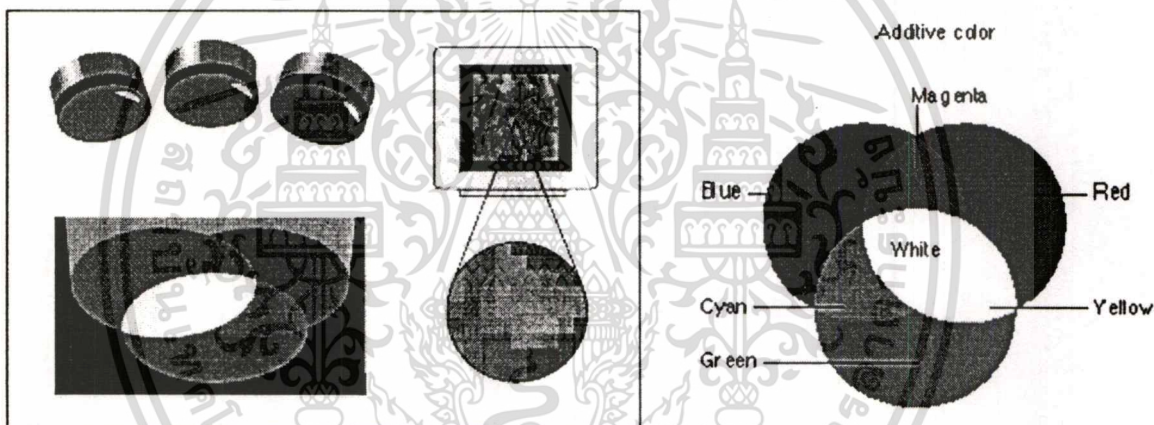
ภาคผนวก ก

การทดสอบเบื้องต้น

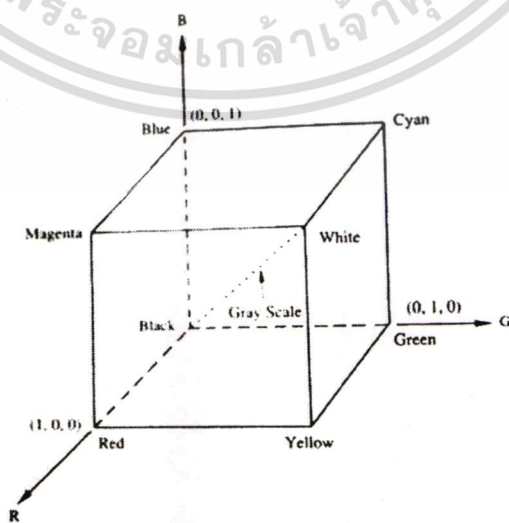
ก.1 การทดสอบมิติสีแบบต่าง ๆ (Colors Space)

ในการทดสอบสีที่นิยมใช้กันทั่วไปในระบบการมองแบบเวลาจริง (real-time vision) นั้นมีอยู่หลายแบบ เช่น RGB, CMY, Lab, YIQ, YUV, HSI และ HSV เป็นต้น แต่ในวิทยานิพนธ์ฉบับนี้จะกล่าวถึงมิติสี 3 แบบเท่านั้น คือ

1. RGB เป็นสีมาตรฐานที่ได้จากกล้องซีซีดี (CCD Camera) แบ่งเป็น ค่าสีแดง (R-red) ค่าสีเขียว (G-green) และค่าสีน้ำเงิน (B-blue) ดังแสดงในรูปที่ ก.1 และ ก.2



รูปที่ ก.1 มิติสีแบบ RGB เป็นมิติสีที่ใช้แสดงผลบนหน้าจอคอมพิวเตอร์ (Monitor)



รูปที่ ก.2 มิติสีแบบ RGB ในรูปแบบของแกนสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ระบบสี YUV เป็นมิตติที่ตัดแปลงมาจาก RGB มีสมการการแปลงค่าดังนี้

$$Y = 0.299R + 0.587G + 0.114B$$

$$U = 0.492(B-Y)$$

$$V = 0.877(R-Y)$$

และเขียนแบบเมทริกซ์ได้ดังนี้

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (ก.1)$$

ซึ่งมิตติแบบ YUV นั้นเกิดขึ้นจากการสร้างสัญญาณโทรทัศน์ยุคแรก ๆ คือ ในยุคแรกที่โทรทัศน์เป็นขาวดำ สัญญาณที่นำมาใช้จะเป็นสัญญาณความสว่าง (Luminance แทนด้วย Y) อย่างเดียว ในขณะที่โทรทัศน์ที่ต้องการสีสามสีคือ แดง เขียว และน้ำเงิน (แทนด้วย R, G และ B ตามลำดับ) เป็นองค์ประกอบหลัก ในการออกแบบจอสีบางระบบ เช่นจอคอมพิวเตอร์ จำเป็นจะต้องใช้สัญญาณซิงค์ (Sync) แยกต่างหาก ดังนั้นจึงต้องใช้สัญญาณ 4 แบบรวมกัน คือ RGBS บางระบบก็จะนำสัญญาณซิงค์นี้ไปผสมไว้ในสัญญาณสีทั้งสาม แต่ที่ใช้กันมากก็คือนำไปผสมกับสัญญาณสีเขียว จึงเป็นที่มาของตัวย่อ RGSB บางครั้งให้เขียนเป็นมิตติ GBR เพื่อให้ทราบว่าเป็นสัญญาณสีประเภทที่มีซิงค์อยู่ในสัญญาณสีเขียวนี้ด้วย เนื่องจาก RGB กับ Y นี้เข้ากันไม่ได้เลย เมื่อจะมีการนำโทรทัศน์สีมาใช้ในครั้งแรก จำเป็นอย่างยิ่งที่จะต้องให้โทรทัศน์ขาวดำเดิมที่มีอยู่มากมายในขณะนั้นรับสัญญาณสีได้ด้วย (Compatible) ขณะเดียวกันเครื่องรับโทรทัศน์สีอันใหม่ก็จะต้องรับสัญญาณจากสถานีโทรทัศน์ขาวดำเดิมได้เช่นกัน (Recompatible) แม้ว่าการสลับการรับสัญญาณต่างกันนี้จะได้ภาพออกมาเป็นขาวดำเท่านั้น แต่ยังคงว่าที่รับสัญญาณไม่ได้เลย อันจะเป็นสาเหตุของการชลดตัวในการนำโทรทัศน์ระบบใหม่ไปใช้งาน ปัญหาก็คือ ทำอย่างไรที่จะส่งสัญญาณไปให้น้อยที่สุดเพื่อความประหยัดและยังคงรักษาความ Compatible กับ Recompatible ดังกล่าวไว้ได้อีกทั้งซึ่งสามารถหาความสัมพันธ์ระหว่างสีกับความสว่างได้ โดยการนำแหล่งกำเนิดแสง ของสีทั้งสามและแสงสีขาวที่มีกำลังส่องสว่าง (แรงเทียน) เท่ากันมาเปรียบเทียบกันบนพื้นขาวพบว่า ด้วยสายตาของผู้ทดสอบโดยเฉลี่ยจะมองเห็นแสง RGB มีความสว่างเทียบกับแสงสีขาวเป็น 0.30, 0.59, 0.11 เท่าของ R, G และ B ตามลำดับ จากตัวเลขจะทราบว่าสายตาของคนเรามีความไวต่อแสงสีเขียวมากที่สุด กล่าวคือเมื่อส่องสว่างด้วยกำลังเท่ากันแต่มองเห็นสีเขียวสว่างกว่า เขียนเป็นสมการแสดงความสัมพันธ์ได้ดังนี้

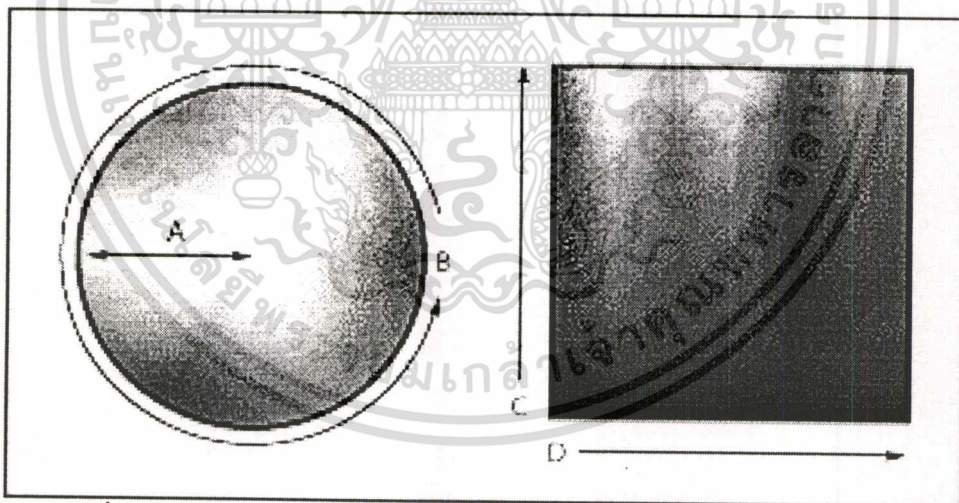
$$Y = 0.30R + 0.59G + 0.11B$$

(ก.2)

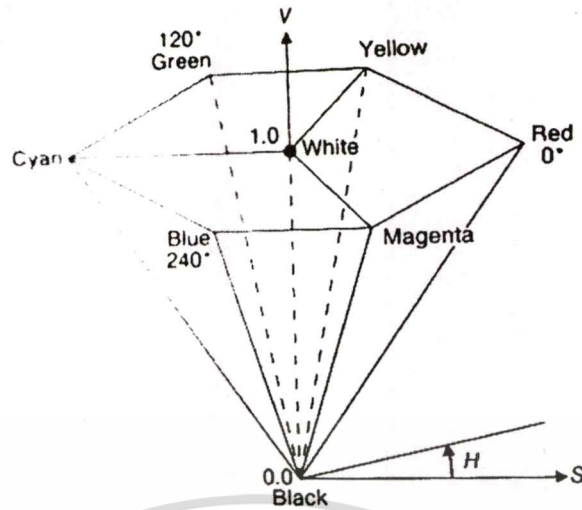
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการข้างต้น ทำให้มีทางออกที่จะไม่ต้องส่งสัญญาณไปได้พร้อมกันทั้งหมด 4 สัญญาณ เนื่องจากสามารถสร้างสัญญาณใดสัญญาณหนึ่งคืนกลับมาใหม่จากความสัมพันธ์ของสมการดังกล่าว แต่หนึ่งในสัญญาณที่จะต้องเลือกก็คือสัญญาณ Y เพื่อให้โทรทัศน์ขาวดำนำไปใช้ได้ทันที อีกสองสัญญาณให้เลือกลงจากสีทั้งสามสี ในทางปฏิบัติจะนำสัญญาณความต่างสี (Color differences) สองสัญญาณคือ R-Y และ B-Y มาใช้งาน แทนที่จะใช้สัญญาณ R และ B เนื่องจากการคืนตัวของสัญญาณ R, G และ B ในทางอิเล็กทรอนิกส์นั้นทำได้สะดวกกว่า อย่างไรก็ตาม สัญญาณความต่างสีนี้เมื่อผ่านขั้นตอนการมอดูเลชันจะทำให้มีความสูงเกินกว่ากำหนด (Over Modulation) จึงต้องลดสัดส่วนลงทำให้ได้สัญญาณใหม่ที่เรียกว่าสัญญาณสี (Chrominance) U และ V นั่นเอง โดยที่ $U = 0.877 (R-Y)$ และ $V = 0.492 (B-Y)$

3. HSV เป็นมิติสีที่ให้ความหมายออกมาได้ใกล้เคียงกับความรู้สีของคนมากที่สุด คือมีตัว H (Hue) เป็นตัวบ่งบอกว่าเป็นสีอะไร (เช่นสีแดง สีส้ม หรือ สีแดงอมม่วงเป็นต้น) ซึ่งมีค่าตั้งแต่ 0-359 หรือจะเรียกว่ามุมของสีก็ได้ เพราะค่า H จะมองสีเป็นวงกลม มีตัว S (Saturation) เป็นตัวบอกความเข้มของสีว่ามีความสดแค่ไหน และตัวสุดท้ายคือ V (Value) เป็นตัวบอกความเข้มของแสงว่ามีความมืดหรือสว่างเท่าใด ดังแสดงในรูป ก.3 และ ก.4 ตามลำดับ



รูปที่ ก.3 มิติสีแบบ HSV ซึ่ง A แทนค่า S(saturation), B แทนค่า H(hue), C แทนค่า V(value) และ D แทนค่า H(hue) เหมือนกับ B [38]



รูปที่ ก.4 มิติสีแบบ HSV ในรูปแบบของแกนสี [38]

จากรูปที่ ก.3 และ รูปที่ ก.4 จะได้สมการการคำนวณค่า HSV ดังนี้

$$H = \cos^{-1} \left\{ \frac{\frac{1}{2}[(\text{Red} - \text{Green}) + (\text{Red} - \text{Blue})]}{[(\text{Red} - \text{Green})^2 + (\text{Red} - \text{Blue})(\text{Green} - \text{Blue})]^{1/2}} \right\} \quad (\text{ก.3})$$

$$S = \left[\frac{\text{MAX}(\text{Red}, \text{Green}, \text{Blue}) - \text{MIN}(\text{Red}, \text{Green}, \text{Blue})}{\text{MAX}(\text{Red}, \text{Green}, \text{Blue})} \right] \times 100 \quad (\text{ก.4})$$

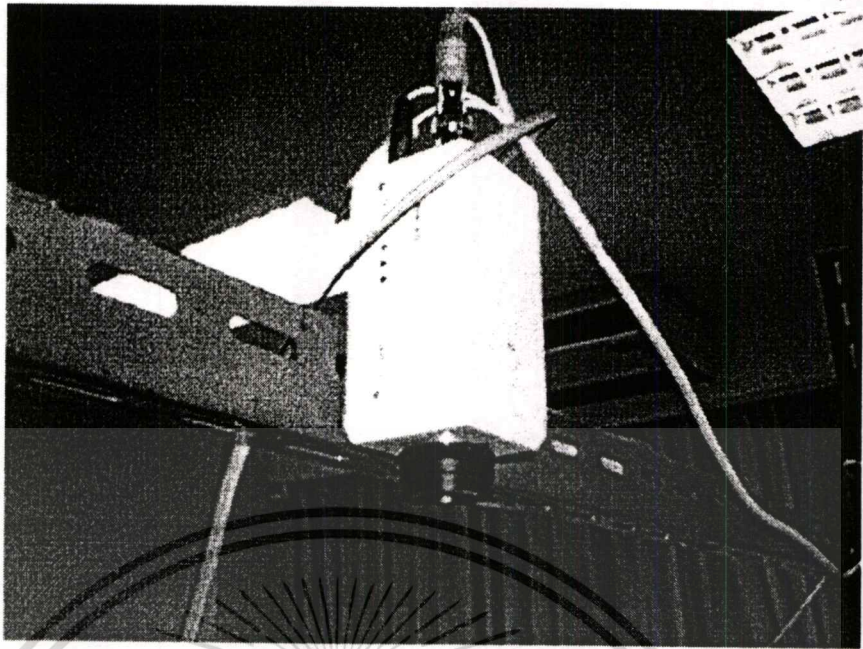
$$V = \text{MAX}(\text{Red}, \text{Green}, \text{Blue}) \quad (\text{ก.5})$$

ก.2 การทดสอบกล้อง (Camera) เพื่อเลือกใช้งาน

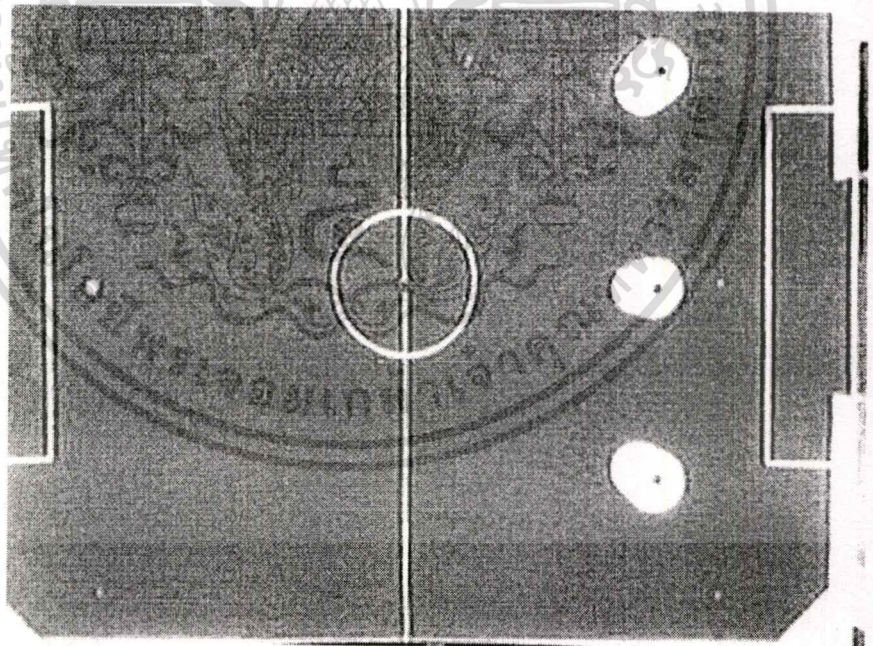
อุปกรณ์ที่นำมาใช้ทดสอบมี 3 แบบ คือ

1. กล้องวงจรปิด ของ Sonican รุ่น HV-2 (รูปที่ ก.5)

เป็นกล้องที่หาได้ง่ายอาจหาซื้อได้โดยทั่วไปโดยมี ราคาไม่แพง ใช้ในชุดระบบโทรทัศน์วงจรปิดทั่วไป แต่เนื่องจากคุณภาพของภาพที่ออกมาไม่ค่อยดี สีจึงผิดเพี้ยน และมีสัญญาณรบกวนมาก และจับภาพได้ไม่เต็มสนาม ดังรูปที่ ก.6 จึงไม่มีการทดสอบต่อ



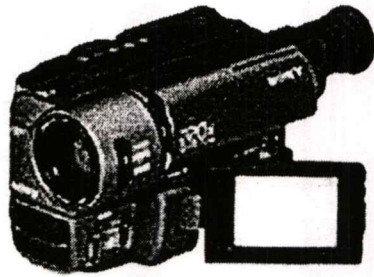
รูปที่ ก.5 กล้องวงจรเปิด ของ Sonican



รูปที่ ก.6 ภาพที่ได้จากกล้องวงจรเปิด ของ Sonican

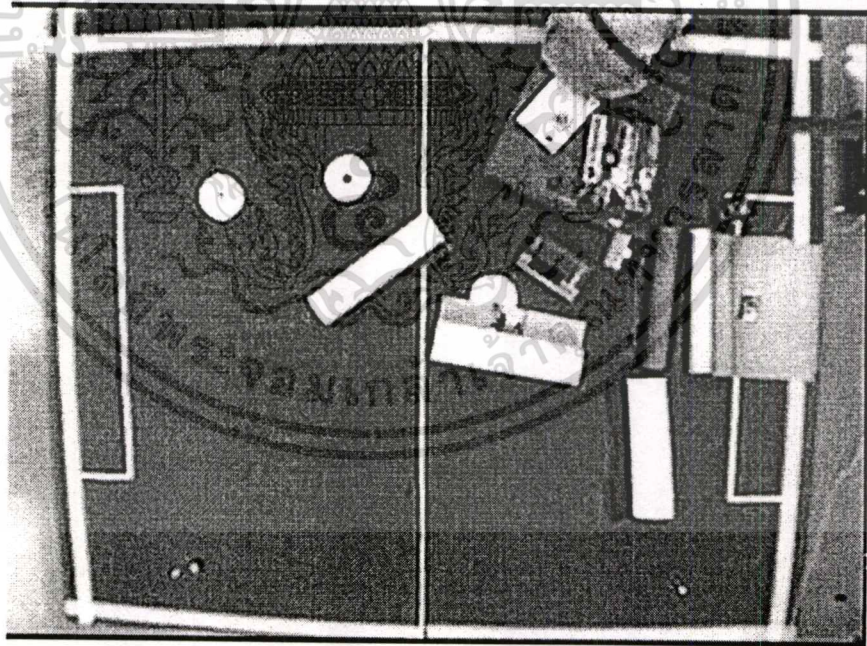
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กล้องถ่ายวิดีโอ ของ Sony รุ่น TRV35E (รูปที่ ก.7)



รูปที่ ก.7 กล้องถ่ายวิดีโอ Sony รุ่น TRV35E

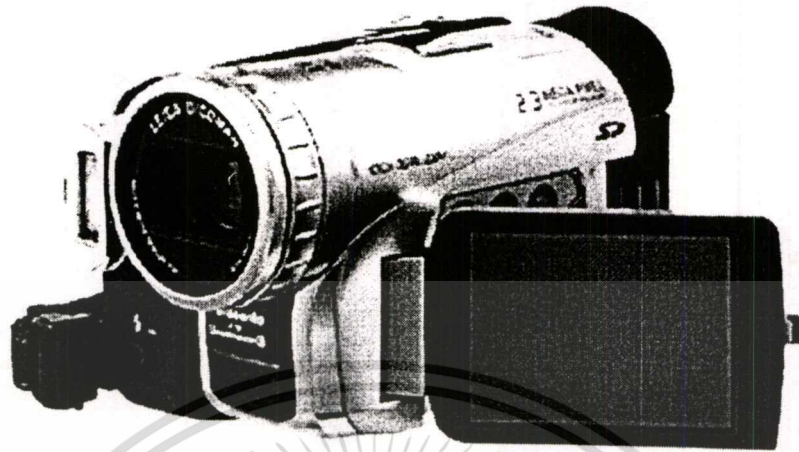
กล้องถ่ายวิดีอรุ่นนี้เป็นรุ่นที่ได้รับความนิยมอย่างมาก จึงได้นำมาทดสอบ แต่เนื่องจากว่ากล้องตัวนี้ไม่สามารถจับภาพได้ทั่วทั้งสนาม จึงได้นำเอาเลนส์มุมกว้างอัตราขยาย 0.6 เท่ามาใช้ร่วมด้วย จากรูป ก.8 จะเห็นว่าได้ภาพทั้งสนาม และสีสันทันก่อนข้างดีน่าจะใช้กับระบบการมองของหุ่นยนต์ได้ดีพอสมควร (ทีม ITE-Gold ของพระจอมเกล้าฯลาดกระบังได้ใช้ในการแข่งขัน Robocup ปี 2003)



รูปที่ ก.8 ภาพที่ได้จากกล้องถ่ายวิดีโอ Sony

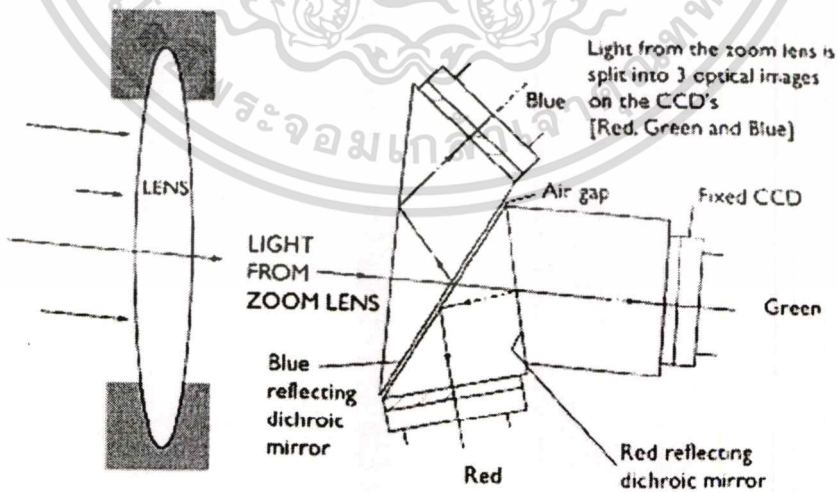
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. กล้องถ่ายวิดีโอ ของ Panasonic รุ่น NV-GS200 (รูปที่ ก.9)



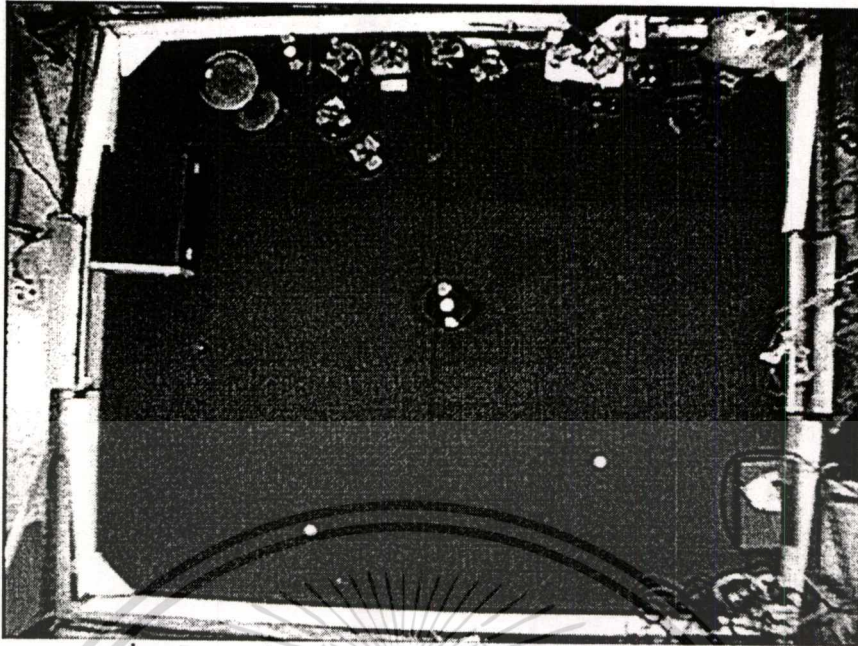
รูปที่ ก.9 กล้องถ่ายวิดีโอ Panasonic NV-GS200

กล้องถ่ายวิดีโอรุ่นนี้เป็นกล้องถ่ายวิดีโอขนาดเล็กรุ่นแรก ๆ ที่ใช้เทคนิคการรับภาพแบบแยกตัวรับภาพ (CCD Sensor) เป็น 3 ตัว เพื่อที่จะแยกรับเฉพาะสีดังรูปที่ ก.10 คือ สีแดง 1 CCD, สีเขียว 1 CCD และสีน้ำเงิน 1 CCD เพื่อที่จะให้สีของภาพถูกต้องชัดเจนยิ่งขึ้น และด้วยปัญหาเดียวกันกับกล้องถ่ายวิดีโอทั่วไปคือ ไม่สามารถถ่ายภาพได้ทั่วสนาม จึงต้องใช้เลนส์มุมกว้างอัตราขยาย 0.6 เท่ามาช่วย จะได้ผลลัพธ์ดังรูปที่ ก.11



รูปที่ ก.10 วิธีการแยกสีออกเป็นแดง เขียว น้ำเงินในกล้อง 3 CCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.11 ภาพที่ได้จากกล้องถ่ายวิดีโอ Panasonic NV-GS200

ก.3 การทดสอบเลนส์ (Lens) ของกล้อง

อุปกรณ์ที่นำมาใช้ทดสอบมี 2 แบบ คือ

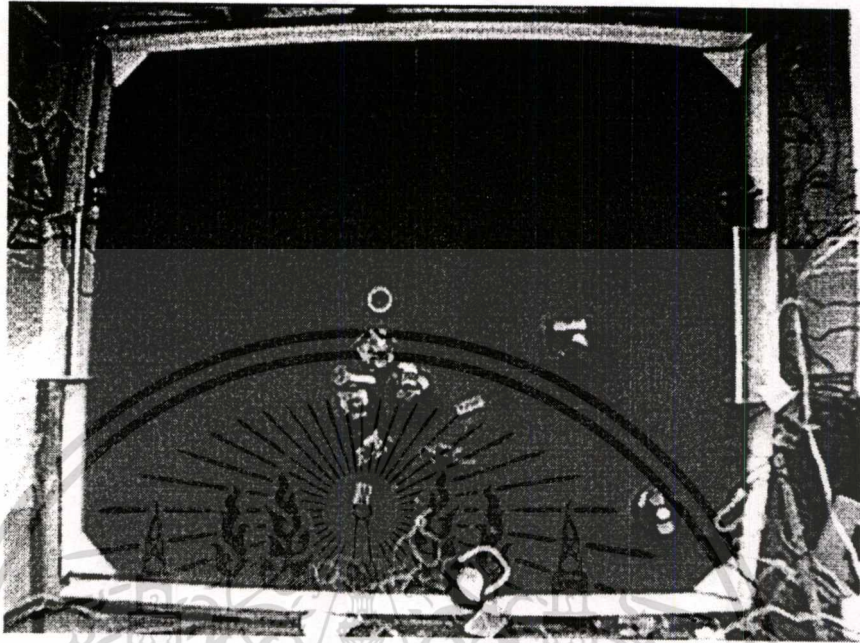
1. เลนส์อัตรการขยาย 0.6 เท่า ของ Sony รุ่น VCI-0637H (รูปที่ ก.12)



รูปที่ ก.12 เลนส์ Sony wide conversion รุ่น VCI-0637H

จากการทดสอบ อัตราการขยาย 0.6 เท่า สามารถเก็บภาพได้ทั้งสนามและเหลือขอบอีกพอสมควร จึงได้ขยายภาพจากกล้อง (Zoom Function) อีกประมาณ 1.1 เท่าเพื่อให้จับภาพได้เต็มสนาม เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอดี ดังรูปที่ ก.13 ซึ่งเห็นความโค้งงอของขอบสนาม (Distortion Error) และขาดความคมชัดของภาพ บริเวณมุมของภาพ แต่อยู่ในเกณฑ์ที่ใช้งานได้



รูปที่ ก.13 ภาพที่ได้จากเลนส์ Sony wide conversion รุ่น VCI-0637H

2. เลนส์อัตรากว้าง 0.7 เท่า ของ Panasonic รุ่น VW-LW3707M3 (รูปที่ ก.14)

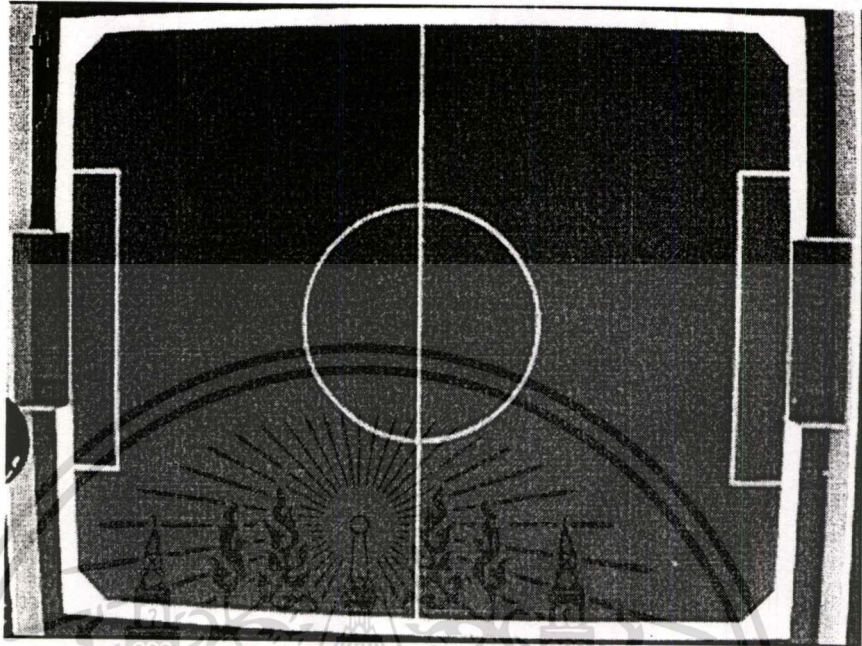


รูปที่ ก.14 เลนส์ Panasonic wide conversion รุ่น VW-LW3707M3

ในรูปที่ ก.15 จะเห็นว่าภาพที่ได้เต็มสนามพอดีโดยไม่ต้องขยายภาพจากกล้อง (Zoom Function) แต่มีปัญหาในการติดตั้ง ก็คือต้องติดตั้งตรงกลางสนามจริง ๆ ติดตั้งเอียงไม่ได้เพราะจะทำให้เก็บภาพได้ไม่ทั่วสนาม คุณภาพของภาพที่ได้มีคุณภาพค่อนข้างดี ความโค้งงอของขอบสนาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Distortion Error) น้อยกว่าเลนส์ของ Sony เนื่องจากทาง Panasonic ใช้วัสดุและการออกแบบที่ดีกว่า และมีราคาสูงกว่า



รูปที่ ก.15 ภาพที่ได้จากเลนส์ Panasonic

ก.4 การทดสอบสายสัญญาณของกล้อง

สายที่นำมาทดสอบแบบออกเป็น 3 ประเภทหลัก ๆ คือ

1. สายสัญญาณเอวี (A/V) ขั้วต่อ RCA



รูปที่ ก.16 สาย A/V ขั้วต่อ RCA

สาย AV (Audio/Video) เป็นสายนำสัญญาณวิดีโอแบบพื้นฐานที่สุดใช้สำหรับส่งสัญญาณภาพจากอุปกรณ์หนึ่งไปยังอีกอุปกรณ์หนึ่ง ใช้กับวิดีโอตามบ้านทั่วไป สาย AV ส่วนใหญ่เป็นสายทองแดงธรรมดา อาจจะมีชีลด์ (หุ้มด้วยลวดเส้นเล็ก ๆ ที่ถักเป็นไหมต่ออยู่กับสายดิน) โดยมีขั้วต่อเป็นแบบ RCA หรือที่เรียกกันว่า หัวฝักบัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สายแบบนี้จะรวมสัญญาณทุกอย่างเข้าเป็นสัญญาณเดี่ยวจึงใช้สายเพียงเส้นเดียว เป็นสัญญาณวิดีโอพื้นฐานที่มีใช้ในอุปกรณ์วิดีโอทุก การรวมสัญญาณทุกอย่างเป็นสัญญาณเดี่ยวทำให้มีการผิเคเพี้ยนของสัญญาณได้ง่ายที่สุดในบรรดาสายสัญญาณทั้งหมดที่นำมาทดสอบ

2. สายสัญญาณวายซี (Y/C) หรือเอชวิดีโอ (S-video) ขั้วต่อ Mini-DIN 4 Pin



รูปที่ ก.17 สาย Y/C หรือเรียกอีกอย่างว่า S-Video ขั้วต่อแบบ Mini-DIN 4 Pin

เป็นสายสัญญาณประเภทหนึ่ง ส่งสัญญาณคล้ายกับมิตีสี YUV แต่ในส่วนของ U กับ V จะถูกจับรวมกันเป็น สัญญาณสี (Chrominance) สัญญาณเดี่ยว ดังนั้นจึงใช้สายสัญญาณเพียงสองเส้น มีคุณภาพรองลงมาจากสัญญาณ YUV แต่ยังคงดีกว่าสายสัญญาณแบบ A/V ที่กล่าวมาข้างต้น

3. สายสัญญาณไออีอี 1394 (IEEE1394 หรือเรียกอีกอย่างว่า FireWire)

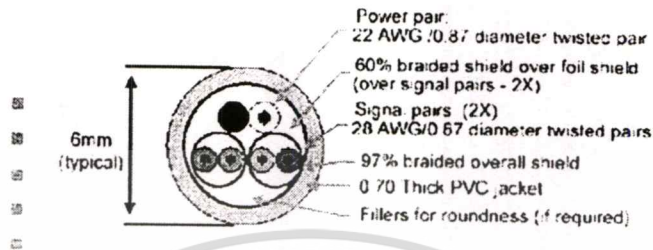


รูปที่ ก.18 สาย IEEE1394 ขั้วต่อแบบ เล็กและแบบใหญ่

จากรูปที่ ก.19 สาย IEEE1394 เป็นสายชนิดแบบมีไส้ 6 เส้น ยาวไม่เกิน 4.5 เมตร ภายในแยกออกเป็นคู่ 3 คู่ สองคู่แรกเป็นสายบิดเกลียวขนาด 40 รอบต่อเมตร ทำจากสายชนิดทองแดงเบอร์ 28 AWG ปลายสองข้างจะสลับกันเพื่อจับคู่การต่อรับ-ส่ง อีกคู่หนึ่งเป็นสายไฟเลี้ยงอุปกรณ์ปลายทาง ทำจากสายทองแดงเช่นกันแต่มีขนาด 22 AWG สำหรับแรงดัน 8-40 Volt และกระแสสูงสุด 1.5 Ampere หัวต่อของสาย 1394 มีสองแบบ แบบแรกเป็นขั้วต่อ 6 ขาเท่ากับจำนวนสาย ขั้วต่อนี้ นำมาจากหัวต่อของสายเกม Nintendo ที่พิสูจน์ความทนทานและใช้งานง่ายมาแล้ว อย่างไรก็ตาม Sony

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้สายที่มีขนาดเล็กที่มีเพียง 4 ขาสำหรับกล่อง DV ของตนเอง โดยออกแบบขั้วต่อใหม่ให้มีขนาดเล็กลงด้วย และสามารถใช้ตัวแปลงสายจากแบบหนึ่งไปยังอีกแบบหนึ่งได้ กรณีที่ต้องการขยายความยาวของสายสามารถทำได้โดยการเพิ่มความหนาของสายหรือลดความเร็วของการส่งลงได้ ทั้งนี้เนื่องจากกล่อง Sony ทั่วไปจะใช้ความเร็วเพียง 100 Mbps เท่านั้น



Capable of operation at 400 Mbit/sec for 4.5 m
- Slightly thicker wire allows 10 meter operation

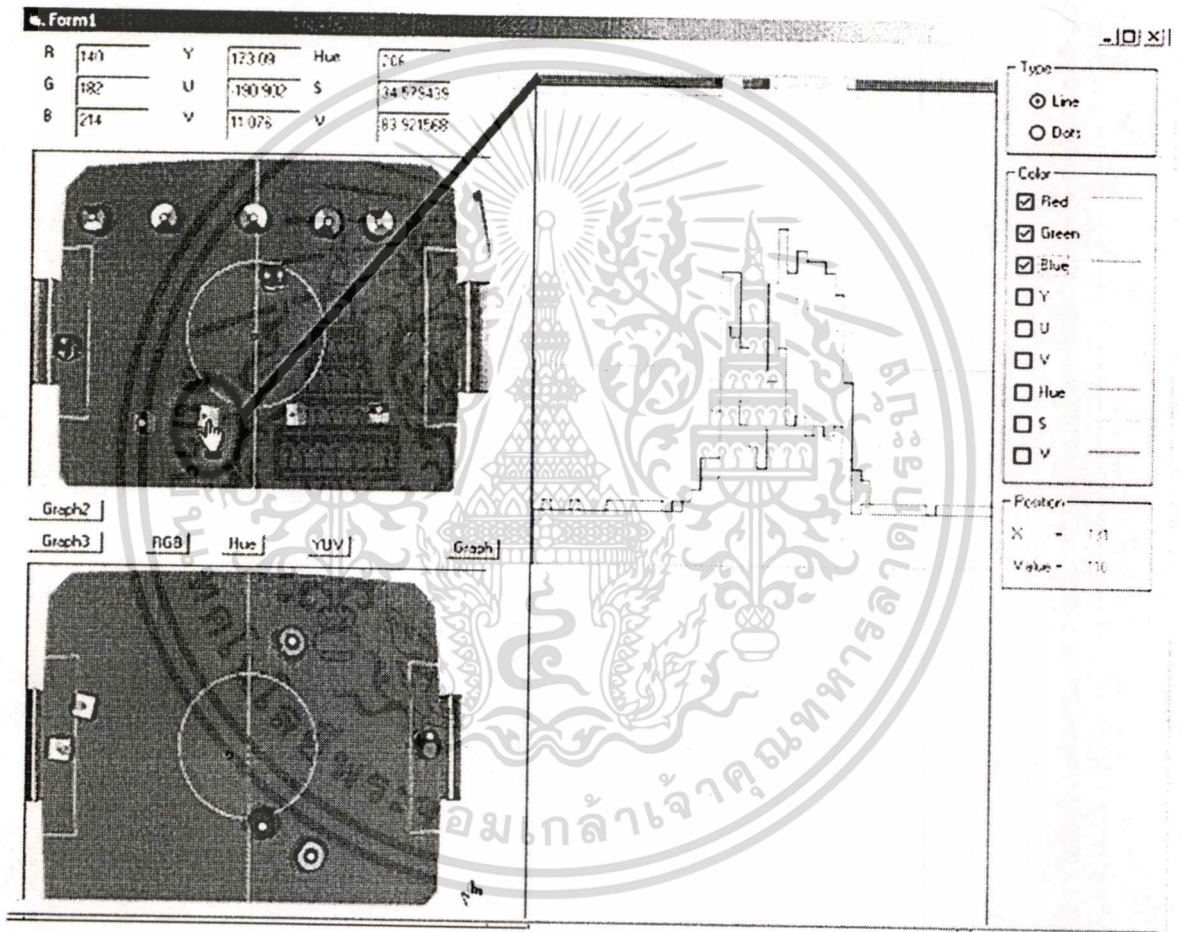
รูปที่ ก.19 มาตรฐานสายสัญญาณ IEEE1394

หลังจากที่ได้ทดสอบพบว่าสายสัญญาณ IEEE1394 พบว่าให้ภาพที่คมชัดมากและความละเอียดสูง แต่ติดปัญหาการแปลงสัญญาณข้อมูลทำให้ภาพที่ได้นั้นช้ากว่าความเป็นจริงประมาณ 1 - 1.5 วินาที (ปัญหานี้เกิดขึ้นจากแผ่นจับสัญญาณที่นำมาใช้คุณภาพต่ำ ไม่สามารถแปลงสัญญาณโดยใช้ตัวเองได้ต้องพึ่งคอมพิวเตอร์ตลอดเวลา) จึงไม่สามารถใช้กับระบบการมองแบบเวลาจริงได้

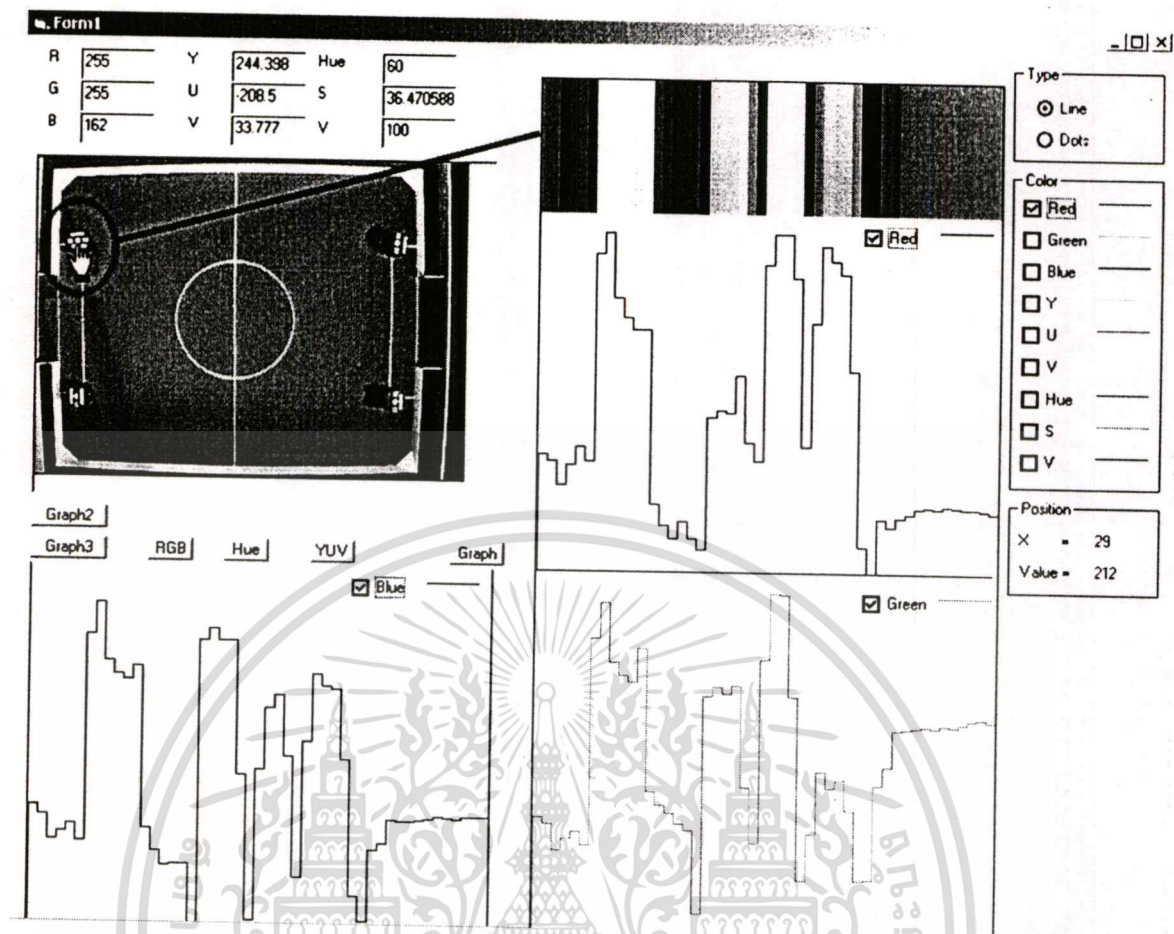
ก.5 ผลการทดสอบเบื้องต้น

ก.5.1 ทดสอบมิติสี

การทดสอบนี้จะใช้โปรแกรมที่เขียนขึ้นเองตามสมการมิติสีต่าง ๆ ชั้นต้นที่กล่าวไปแล้ว (สมการที่ ก.1, ก.3, ก.4 และ ก.5) บนกล้องถ่ายวิดีโอ ของ Panasonic รุ่น NV-GS200 สายเอชวีดีโอ (S-video) เลนส์อัตรากว้าง 0.7 เท่า ของ Panasonic รุ่น VW-LW3707M3 ที่ความละเอียด 320 X 240 จุดสี (pixel) ต่อภาพ ซึ่งโปรแกรมที่ใช้ทดสอบ ดังรูปที่ ก.20, ก.21, ก.22 และ ก.23 นั้นพัฒนาโดยผู้วิจัย จะพิจารณาผลลัพธ์จากกราฟความต่างกันของสีในมิติต่าง ๆ

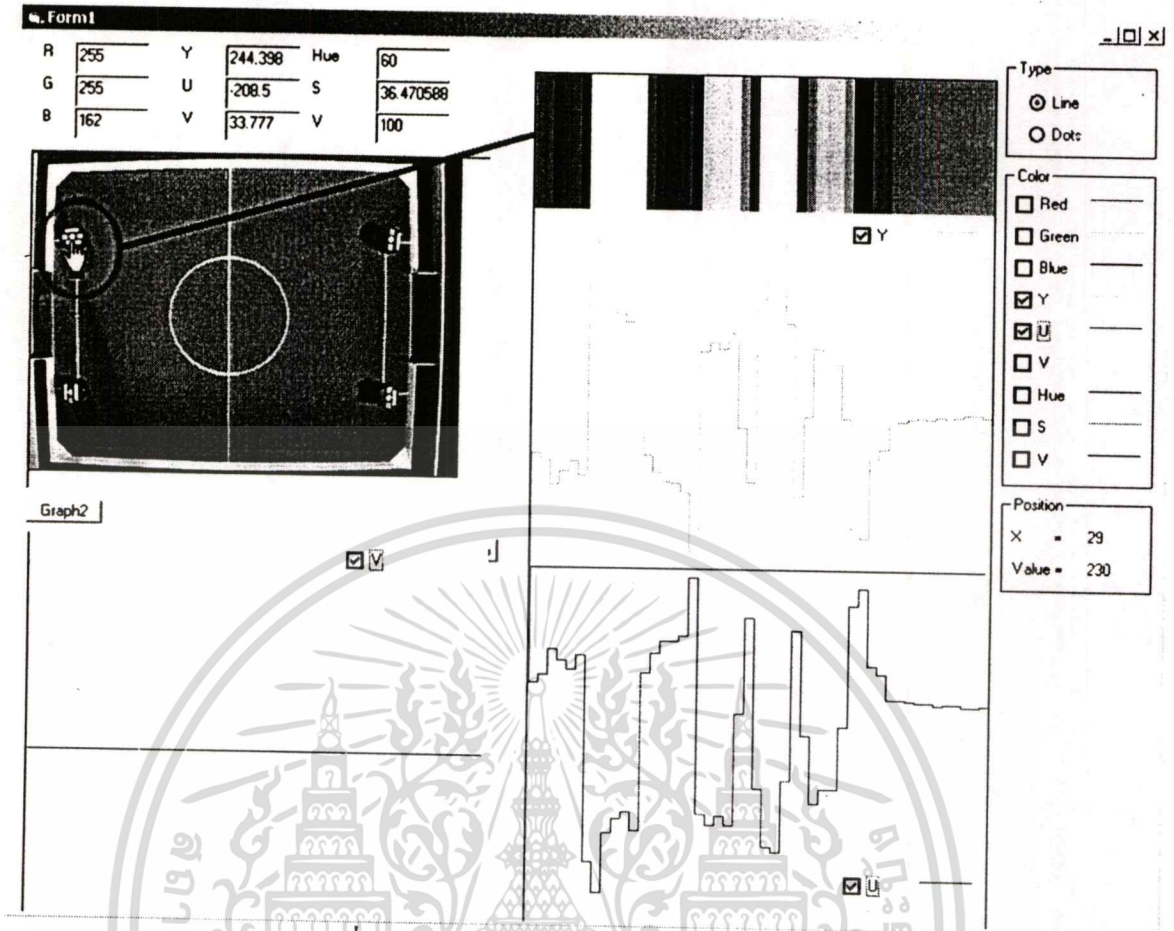


รูปที่ ก.20 โปรแกรมที่เขียนเพื่อระดับสีในมิติของ RGB, YUV และ HSV



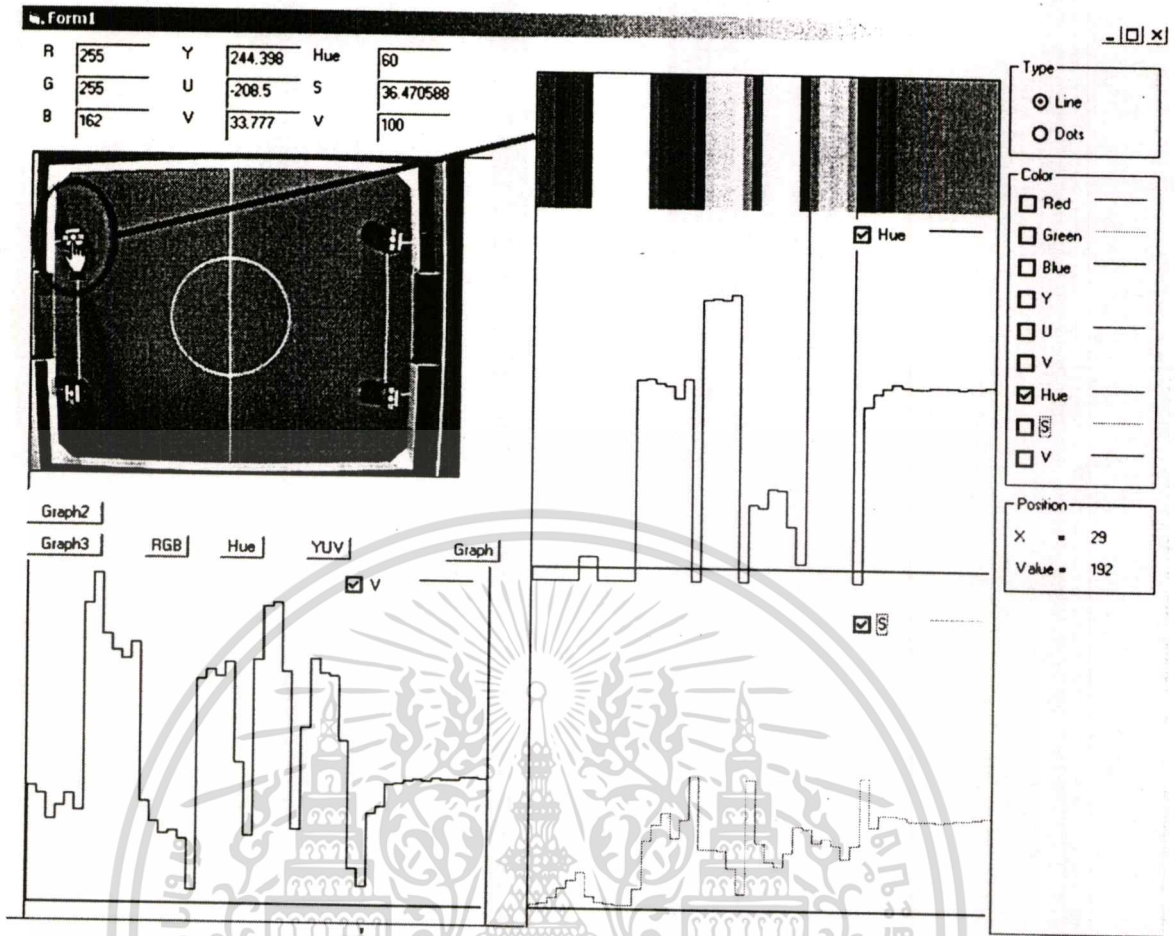
รูปที่ ก.21 ผลของการทดสอบมิติของสี RGB

ผลลัพธ์ในมิติสี RGB ดังรูป ก.21 แสดงให้เห็นว่าความแตกต่างระหว่างสีไม่ค่อยแตกต่างกันมาก ใน Red เห็นว่าสีขาว สีเหลือง และสีม่วงแดงนั้นมีระดับสีใกล้เคียงกันมาก ใน Green มีสีขาวกับสีฟ้าระดับสีใกล้เคียงกัน และใน Blue ทุกสีบนหุ่นยนต์มีระดับใกล้เคียงกัน จึงยากที่จะแบ่งสีต่างๆ ออกจากกัน



รูปที่ ก.22 ผลของการทดสอบมิติของสี YUV

ผลลัพธ์ในมิติสี YUV ดังรูป ก.22 แสดงให้เห็นว่าความแตกต่างระหว่างสีไม่ค่อยแตกต่างกัน แต่แตกต่างมากกว่าในรูปที่ ก.21 คือมิติสี RGB ใน Y เห็นว่าสีขาวกกับสีเหลือง และสีม่วงแดง กับสีฟ้านั้นมีระดับสีใกล้เคียงกันมาก ใน U มีลักษณะคล้ายคลึงกับ Y แต่กลับด้านกัน และใน V แบ่งสีต่างๆ ออกจากกันได้ แต่แบ่งสีขาวกกับสีเหลืองไม่ค่อยดีนัก



รูปที่ ก.23 ผลของการทดสอบมิติของสี HSV

ผลลัพธ์ในมิติสี HSV ดังรูป ก.23 แสดงให้เห็นว่าความแตกต่างระหว่างสีแตกต่างกันได้ดีกว่าสีรูปที่ ก.21 และ ก.22 ใน Hue เห็นว่าสีต่าง ๆ แตกต่างกันอย่างชัดเจน ใน S ความแตกต่างค่าสีไม่ค่อยดีนัก และใน V แบ่งความสว่างของสีได้อย่างชัดเจนแต่แบ่งสีที่ต้องการบนตัวหุ่นยนต์ไม่ค่อยดีนัก

เมื่อพิจารณาผลลัพธ์ของทั้ง 3 มิติสี สรุปได้ว่าค่าที่ได้จากมิติสี HSV ให้ค่าที่ดีที่สุด เพราะสามารถแบ่งแยกสีต่าง ๆ ได้อย่างชัดเจนที่สุด

ก.5.2 ทดสอบสมรรถนะของกล้อง

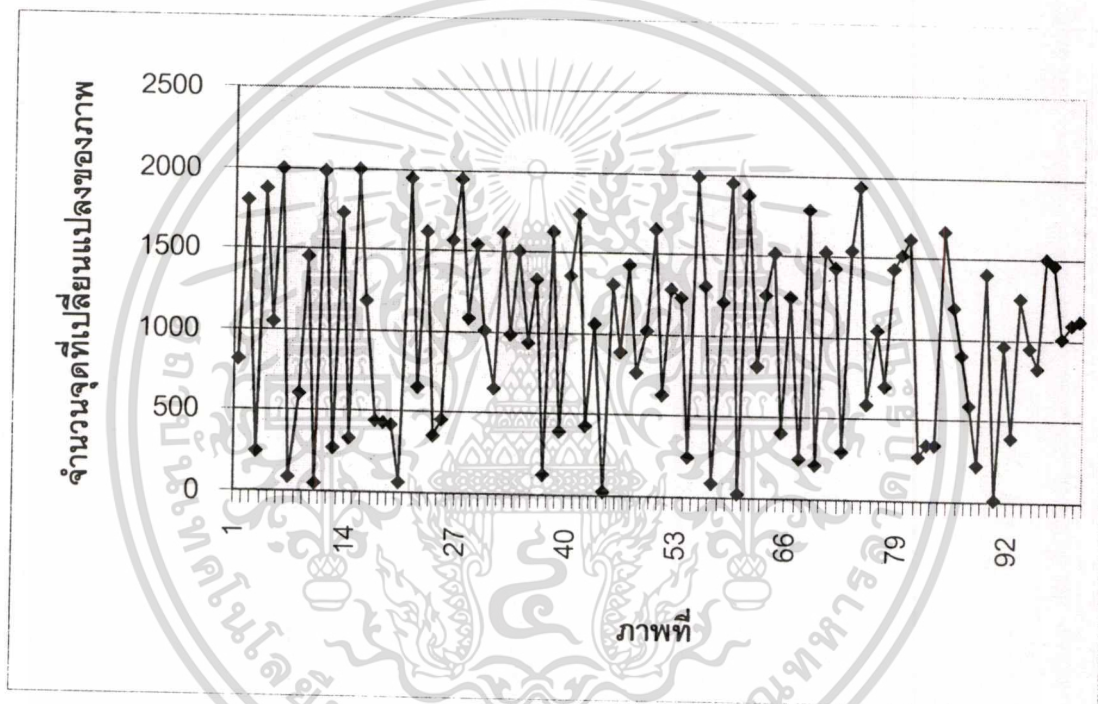
การทดสอบสมรรถนะกล้องจะใช้ทดสอบบนสายเอชวีดีโอ (S-video) เเลนส์อัตราขยาย 0.7 เท่า ของ Panasonic รุ่น VW-LW3707M3 ที่ความละเอียด $320 \times 240 = 76800$ จุดสีต่อภาพ โดยมีขั้นตอนการทดสอบดังนี้

1. นำกล้องยึดติดกับขาตั้งกล้องให้แน่น ห้ามกล้องสั่นในขณะทดลอง
2. จัดพื้นหลังหรือพื้นสนามให้อยู่ในสถานะคงที่ ห้ามมีสิ่งใดขยับ
3. ถ่ายภาพจำนวน 100 ภาพ

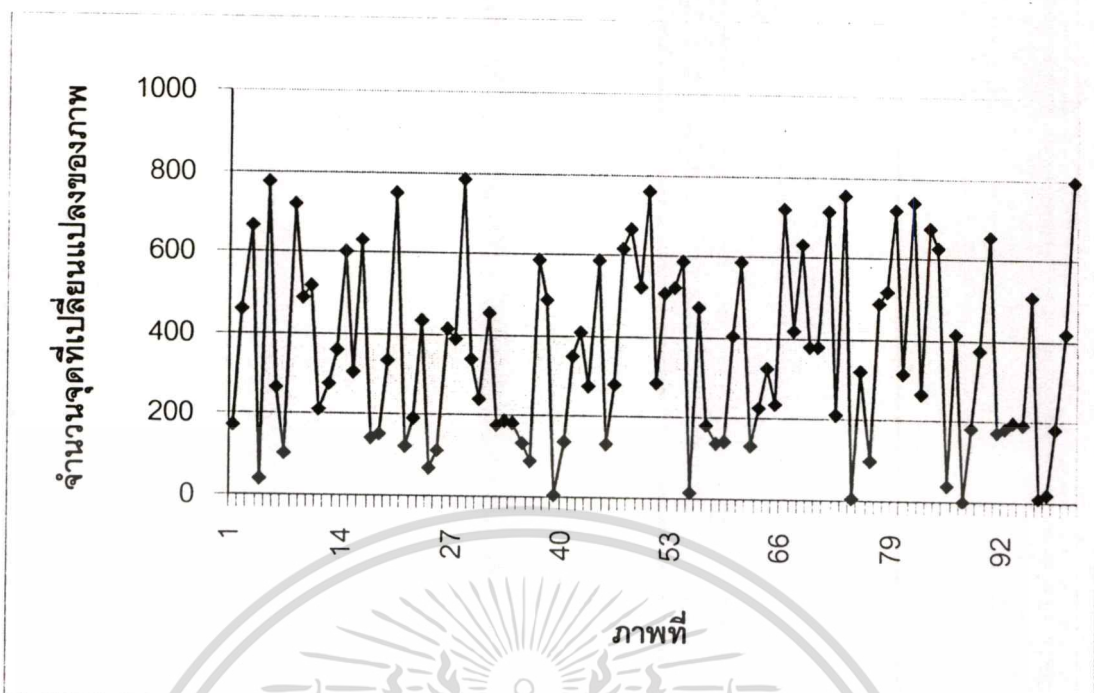
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. นำภาพทั้งหมดมาเปรียบเทียบกับภาพแรกในชุดแบบจุดต่อจุด เพื่อหาจุดที่เปลี่ยนแปลง
5. นับจำนวนจุดที่เปลี่ยนแปลงของภาพนั้น ๆ และบันทึกผล
6. เปลี่ยนกล้องและกลับไปทดสอบข้อที่ 1

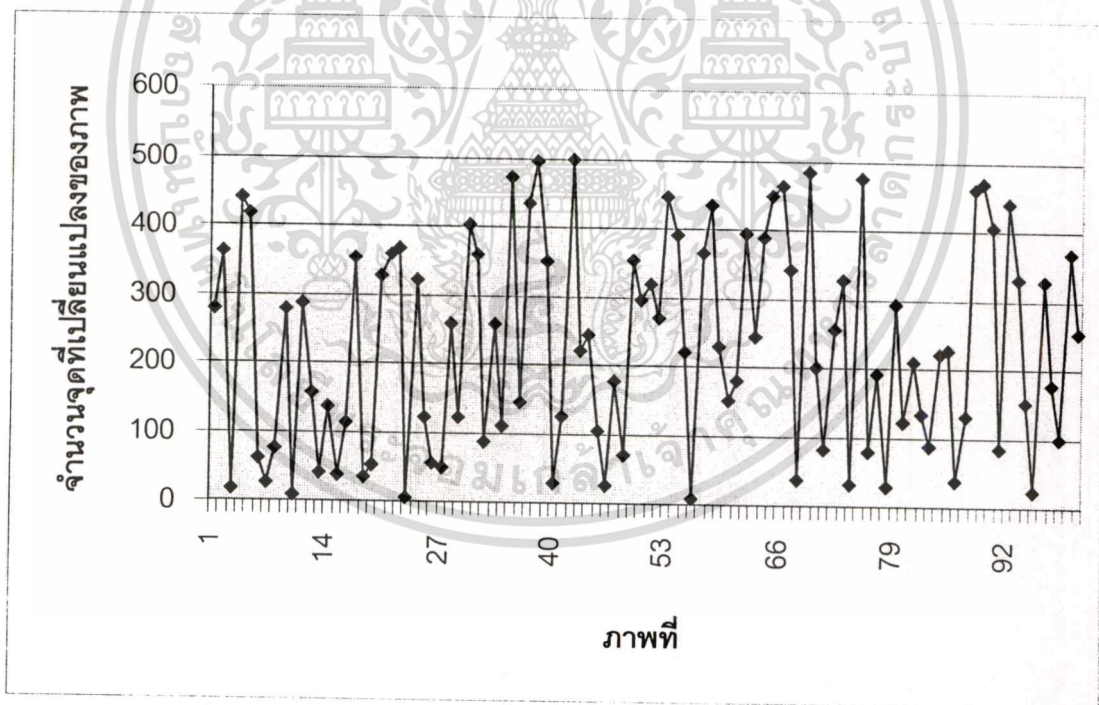
ผลการทดสอบค่าจุดสีที่เปลี่ยนแปลงใน 1 ช่วงเวลา (100 ภาพ) โดยหากมีจุดเปลี่ยนแปลงของจุดสีน้อยจะบอกระดับสมรรถนะที่ทนทานต่อสัญญาณรบกวนที่ดี แสดงดังรูป ก.24 – ก.26 เมื่อเทียบผลการทดสอบสามารถสรุปได้ดังตารางที่ ก.1 ซึ่งแสดงผลการทดลองทั้งหมด 6 ครั้ง ที่สภาวะเดียวกัน



รูปที่ ก.24 กล้องวงจรปิด ของ Sonican รุ่น HV-2 ใน 100 ภาพ



รูปที่ ก.25 กล้องถ่ายวิดีโอ ของ Sony รุ่น TRV35E ใน 100 ภาพ



รูปที่ ก.26 กล้องถ่ายวิดีโอ ของ Panasonic รุ่น NV-GS200 ใน 100 ภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.1 การทดสอบจุดสีที่ค่าเปลี่ยนแปลงจากสีพื้นหลังของกล้องทั้ง 3 รุ่นเฉลี่ยใน 100 ภาพ

	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	ครั้งที่ 6	เฉลี่ย
Sonican รุ่น HV-2	976.78	1027.2	917.53	898.83	826.98	950.87	933.03
Sony รุ่น TRV35E	357.03	316.78	310.67	307.21	397.2	396.4	347.55
Panasonic รุ่น NV-GS200	246.08	281.09	245.48	248.42	220.52	247.58	248.11

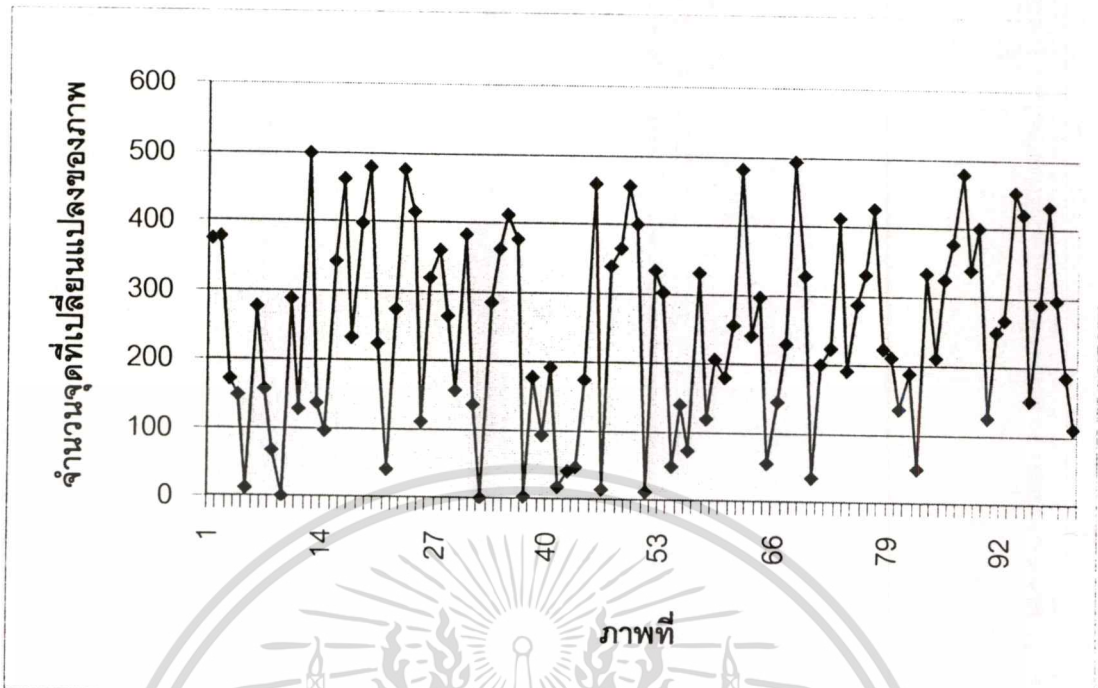
จากผลลัพธ์จากตารางที่ ก.1 สรุปว่าควรเลือกใช้กล้อง Panasonic รุ่น NV-GS200 เพราะ ให้ค่าเปลี่ยนแปลงของจุดสีในสภาวะหยุดนิ่งน้อยที่สุด หรือเรียกได้ว่ามีสัญญาณรบกวนออกมาน้อยที่สุด

ก.5.3 ทดสอบเลนส์

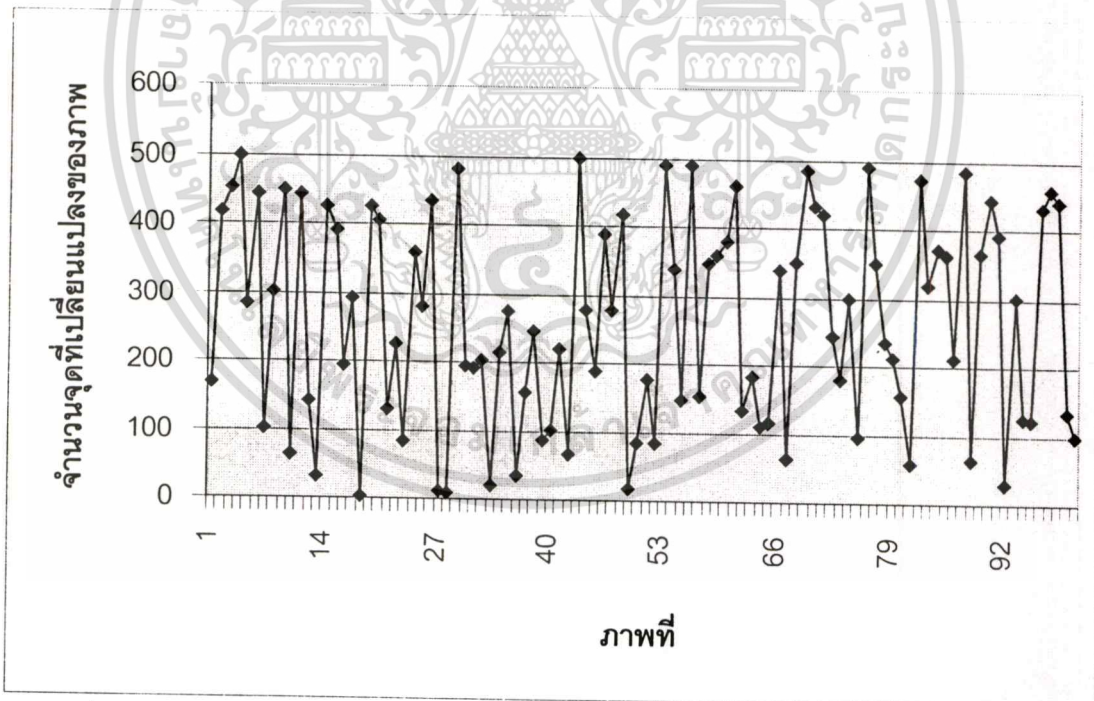
การทดสอบเลนส์ (Lens) จะใช้ทดสอบบนกล้องถ่ายวิดีโอ ของ Panasonic รุ่น NV-GS200 สายเอชวีดีโอ (S-video) ที่ความละเอียด $320 \times 240 = 76800$ จุดสีต่อภาพ โดยมีขั้นตอนการทดสอบดังนี้

1. นำกล้องยึดติดกับขาตั้งกล้องให้แน่น ห้ามกล้องสั่นในขณะที่ทดลอง
2. จัดพื้นหลังหรือพื้นสนามให้อยู่ในสภาวะคงที่ ห้ามมีสิ่งใดขยับ
3. ถ่ายภาพจำนวน 100 ภาพ
4. นำภาพทั้งหมดมาเปรียบเทียบกับภาพแรกในชุดแบบจุดต่อจุด เพื่อหาจุดที่เปลี่ยนแปลง
5. นับจำนวนจุดที่เปลี่ยนแปลงของภาพนั้น ๆ และบันทึกผล
6. เปลี่ยนเลนส์ และกลับไปทดสอบข้อที่ 1

ผลการทดสอบค่าจุดสีที่เปลี่ยนแปลงใน 1 ช่วงเวลา (100 ภาพ) โดยหากมีจุดเปลี่ยนแปลงของจุดสีน้อยจะบอกถึงสมรรถนะที่ทนทานต่อสัญญาณรบกวนที่ดี แสดงดังรูป ก.27 - ก.28 เมื่อเทียบผลการทดสอบสามารถสรุปได้ดังตารางที่ ก.2 ซึ่งแสดงผลการทดลองทั้งหมด 6 ครั้ง ที่สภาวะเดียวกัน



รูปที่ ก.27 เกลนส์อัตรการขยาย 0.6 เท่า ของ Sony รุ่น VCI-0637H ใน 100 ภาพ



รูปที่ ก.28 เกลนส์อัตรการขยาย 0.7 เท่า ของ Panasonic รุ่น VW-LW3707M3 ใน 100 ภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.2 เปรียบเทียบจุดสีที่เปลี่ยนแปลงไปจากพื้นหลังของเลนส์ทั้ง 2 แบบเฉลี่ยใน 100 ภาพ

	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	ครั้งที่ 6	เฉลี่ย
Sony รุ่น VCI-0637H	246.084	275.556	239.867	222.829	288.767	226.387	249.925
Panasonic รุ่น VW-LW3707M3	239.235	217.278	223.743	266.57	213.25	267.536	240.935

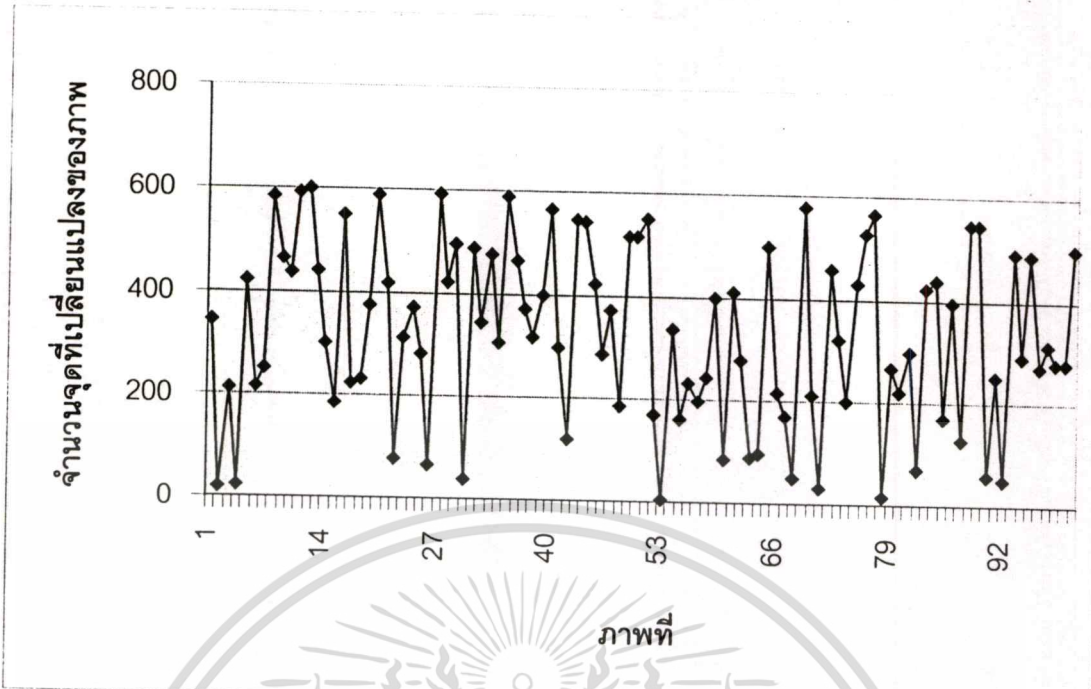
จากผลลัพธ์จากตารางที่ ก.2 สรุปว่าควรเลือกใช้เลนส์ Panasonic รุ่น VW-LW3707M3 เพราะ ให้ค่าเปลี่ยนแปลงของจุดสีในสภาวะหยุดนิ่งน้อยที่สุดโดยเฉลี่ย หรือเรียกได้ว่ามีสัญญาณรบกวนออกมาน้อยที่สุด

ก.5.4 ทดสอบสายสัญญาณ

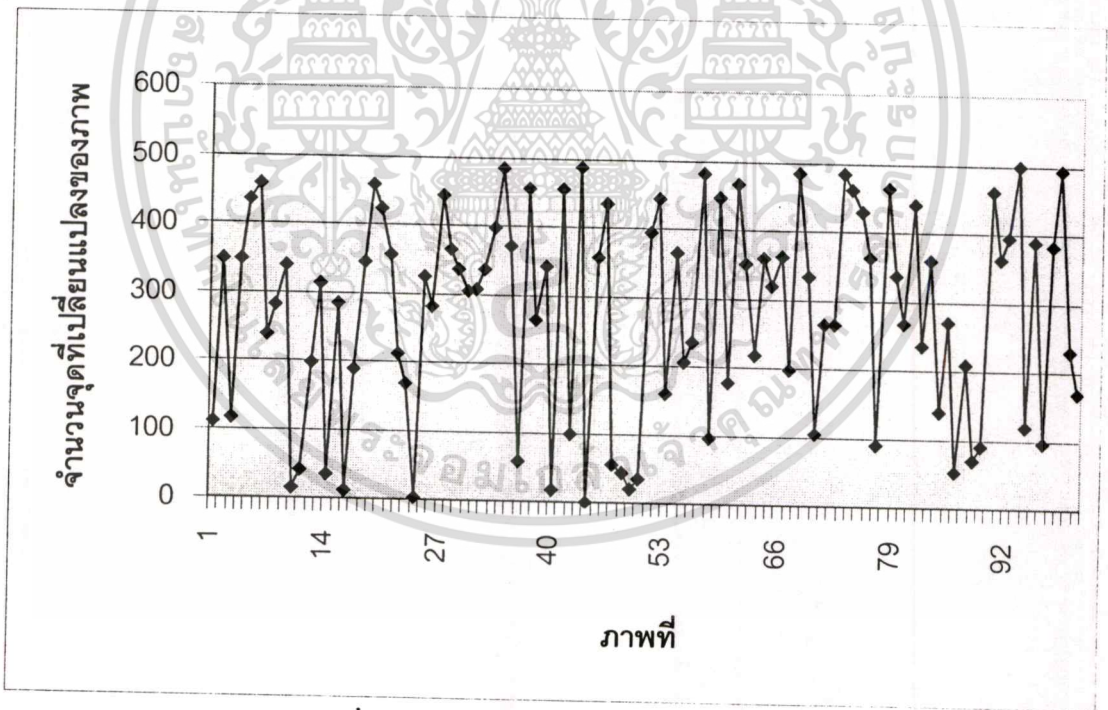
การทดสอบสายสัญญาณจะทดสอบบนกล้องถ่ายภาพวิดีโอ ของ Panasonic รุ่น NV-GS200 เลนส์อัตราขยาย 0.7 เท่า ของ Panasonic รุ่น VW-LW3707M3 ที่ความละเอียด 320 x 240 = 76800 จุดสีต่อภาพ โดยมีขั้นตอนการทดสอบดังนี้

1. นำกล้องยึดติดกับขาตั้งกล้องให้แน่น ห้ามกล้องสั่นในขณะทดลอง
2. จัดพื้นหลังหรือพื้นสนามให้อยู่ในสภาวะคงที่ ห้ามมีสิ่งใดขยับ
3. ถ่ายภาพจำนวน 100 ภาพ
4. นำภาพทั้งหมดมาเปรียบเทียบกับภาพแรกในชุดแบบจุดต่อจุด เพื่อหาจุดที่เปลี่ยนแปลง
5. นับจำนวนจุดที่เปลี่ยนแปลงของภาพนั้น ๆ และบันทึกผล
6. เปลี่ยนสายสัญญาณ และกลับไปทดสอบข้อที่ 1

ผลการทดสอบค่าจุดสีที่เปลี่ยนแปลงใน 1 ช่วงเวลา (100 ภาพ) โดยหากมีจุดเปลี่ยนแปลงของจุดสีน้อยจะบอกถึงสมรรถนะที่ทนทานต่อสัญญาณรบกวนที่ดี แสดงดังรูป ก.29 – ก.31 เมื่อเทียบผลการทดสอบสามารถสรุปได้ดังตารางที่ ก.3 ซึ่งแสดงผลการทดลองทั้งหมด 6 ครั้ง ที่สภาวะเดียวกัน

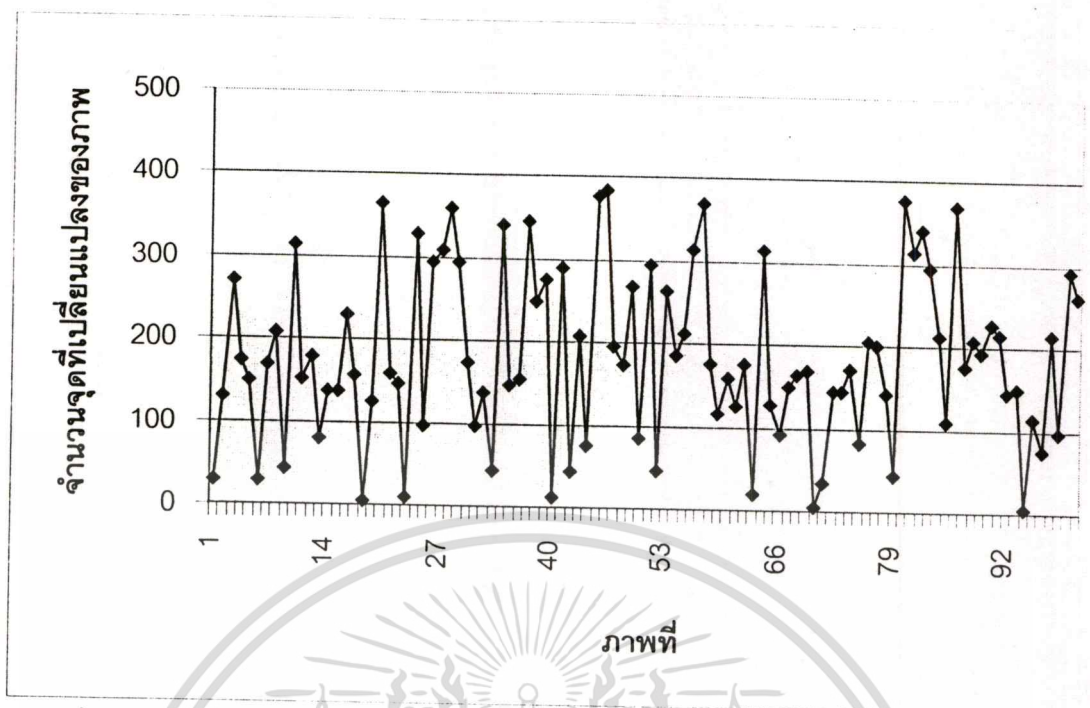


รูปที่ ก.29 สายสัญญาณเอวี (A/V) ขั้วต่อ RCA ใน 100 ภาพ



รูปที่ ก.30 เอชวีดีโอ (S-video) ใน 100 ภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.31 สายสัญญาณไออีอีอี 1394 (IEEE1394 หรือเรียกอีกอย่างว่า FireWire) ใน 100 ภาพ

ตารางที่ ก.3 จุดสีที่ผิดเพี้ยนไปจากพื้นหลังของสายสัญญาณทั้ง 3 แบบเฉลี่ยใน 100 ภาพ

	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	ครั้งที่ 6	เฉลี่ย
เอวี (A/V) ขั้วต่อ RCA	294.59	313.03	297.88	305.38	328.85	285.65	304.23
เอชวีดีโอ (S-video)	236.41	254.26	272.76	222.9	198.54	277.47	243.72
ไออีอีอี 1394 (IEEE1394)	192.04	179.1	181.33	229.48	165.72	186.51	189.03

จากผลลัพธ์จากตารางที่ ก.3 สรุปว่าควรเลือกใช้สายสัญญาณแบบไออีอีอี 1394 เพราะให้ค่าเปลี่ยนแปลงของจุดสีในสภาวะหุคหนึ่งน้อยที่สุด หรือเรียกได้ว่ามีสัญญาณรบกวนออกมาน้อยที่สุด แต่จากการทดลองพบปัญหากับไออีอีอี 1394 คือภาพที่ได้ช้ากว่าความเป็นจริงประมาณ 1 - 1.5 วินาที ซึ่งไม่สามารถใช้งานได้แบบเวลาจริง จึงเปลี่ยนมาเลือกใช้แบบเอชวีดีโอ เพราะให้ผลลัพธ์ลดลงมาจาก ไออีอีอี 1394

ภาคผนวก ข
ผลงานทางวิชาการที่ได้รับการตีพิมพ์

[1] J. Jaturawit, B. Purahong, P. Sooraksa, and G. Chen, " A Low-Cost Soccer Robot Term for Education ", 2002 FIRA Robot World Congress, pp.255-257, 2002.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KRSA

Proceedings

2002

IFIRA



0110111010211010101011101010011011
 101010101011101010101000101011011
 P101001010101010101010100110101011
 01010110011010101010000101011010100
 1010101010101010100110101011010001
 1001101010110100101011010100101110

0101100110101
 1000101011010
 0011010101101
 0101101010010
 0101101000101
 1010010111011

World Congress

May 26-29, 2002 / COEX, Seoul, KOREA



ORGANIZED BY Korea Robot Soccer Association (KRSA)

SUPPORTED BY - Robotics Society of Japan (RSJ)

- Chinese Association of Artificial Intelligence (CAAI)

- Korean Society of Precision Engineering (KSPE)

- The Institute of Control, Automation and Systems Engineers (ICASE)

- Ministry of Science and Technology (MOST)

- Ministry of Commerce, Industry and Energy (MOCIE)

- Korea Science and Engineering Foundation (KOSEF)

- Korea Science Foundation (KSF)

- Micro-Robot Design Education Center (MRDEC)

- Yujin Robotics Co., Ltd.

- Robotech Co., Ltd.

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารไว้ทุกแห่ง

A Low-cost Soccer Robot Team for Education

Janrawit Jantapboon*, Boonchana Parahong*, Pitkhate Sooraksa*, Guanrong Chen**

* Department of Information Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Chalokkrung Rd., Ladkrabang, 10520 Thailand

(Tel: 66-0-2326-9084; Fax: 66-0-2326-9084; E-mail: kspitkh@kmit.ac.th)

** Department of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Avenue Kowloon, Hong Kong, P. R. China

(E-mail: gchen@ee.cityu.edu.hk)

Abstract: Implementation of a soccer robot team requires both interdisciplinary knowledge and integrated techniques in many engineering fields such as communication, computer, electronics, mechanics, materials and control systems. This sets a good example for cooperative studies in sciences and engineering. For education purpose, it is desirable to downscale the state-of-the-art design and implementation of a soccer robot to be one with low-cost for high school teaching and demo. This paper reports that this indeed is possible, and we have completed a design and implementation of a soccer robot team of four players with a total budget under US\$300. The system can be built by high school students within a reasonable time duration with minimum effort. This paper also describes hardware description and components functionality of the system.

1. Introduction

The emergence of robot soccer games offers creativity and entertainment to engineering education. This attracts and motivates students, researchers, and many others around the world to the field of robotics. Robots not only perform a service to humans in various dirty, dangerous, or repetitious and redundant jobs, but also act like entertainers on many occasions such as the FIFA [1] and FIRA games [2], as well as the soccer robot games [3].

Although many institutions would like to participate in this kind of technological competitions and activities, some of them have difficulty in funding the project due to insufficient budget. One example is the high school education where students have great interest but have no funds to participate in such soccer robot games. This paper is to look for an option providing a low-cost solution to build a soccer robot, along with its design and implementation. The total cost can be under US\$300, excluding the PC price. Although this simplified technology cannot compete with the cutting-edge model, it is a good start for better learning and changing the students' attitude from a watcher to a player in the games. At the technical level, this paper describes major components needed for building a soccer robot player, which is given in Section 2, and discusses how to mechanically assemble the player in Section 3. Section 4 provides a procedure of electronically assembling the soccer robot, and Section 5 concludes the paper.

2. Major Components

An example of choosing components and estimating price for building a robot soccer team is given in this section. The suggestion is based on our experience of building soccer robot teams. Similar to those reported in [4], [5], major components of a player uses wireless communication via IR, which are:

1. Micro Controller: There are many choices for a microcontroller for each robot. However, for economic reason and easy to use, 4 of 8-bit type AT89C2051 are used. MCS51 microcontroller is used, which approximately costs around US\$2.5 each.
2. IC Driver Motor: It is recommended to obtain an IC that has competency to drive two motors under low voltages. If we cannot find this type of IC, alternatively, we can

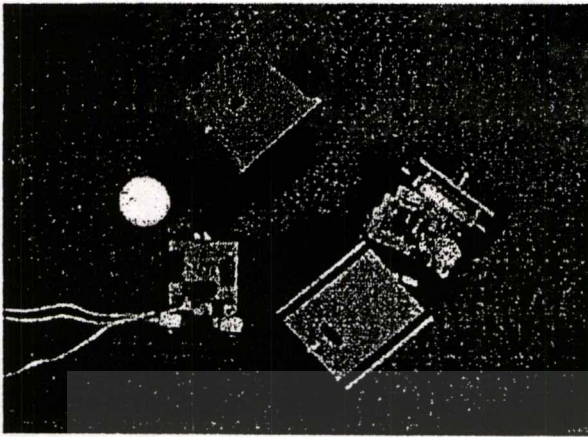
use an H-Bridge drive. Here, we use 4 of IC numbered L293D (each IC costs about US\$2.5).

3. Transmitter and Receiver Units: Infrared transmitter and receiver unit is cheaper than the RF ones. In our project, receivers with 9600 bits per sec with 3-6 volt power supply are used. Here, 5 of IR module Infrared, TFDS4500, are installed (each costs about US\$5.6).
4. Motor and Gear Box: It is convenient to use motor and gear box of TAMIYA because its availability in toy stores. We need 4 sets for a team of four players, and each set costs about US\$8.
5. Encoder: This part is optional. However, it is easy to control and to improve accuracy of the robotic system if each player in the system is equipped with this part. Here, we use 4 encoder obtained from a PC mouse (each costs about US\$5).
6. Battery: We use 4 sets of Ni-MH rechargeable battery (1.2V, 700mAh), each set costs about US\$8.
7. Housing: Body of a robot used here is made of PVC plastic, which costs about US\$50.
8. CCD Camera: A CCD camera is need. The price is about \$70 with acceptable performance.
9. Computer with TV-input: This computer requires a speed of at least 500 Mhz to process signals and carry out calculation—the faster, the better.
10. Other minor components, such as wires, cost about US\$20-\$30.

As can be seen from the parts and components listed above for a soccer robot team (a set of four players), the total price is about US\$282, excluding the computer. This estimated price is obtained from our recent survey from sources and electronic stores in Thailand. The price could be even cheaper in some countries where semi-conductor manufacturing industries are located.

Figure 1 shows a top-view of the major components assembled as modules for a robot player.

A LOW-COST ROBOT TEAM FOR



(a) Compared with a golf ball.

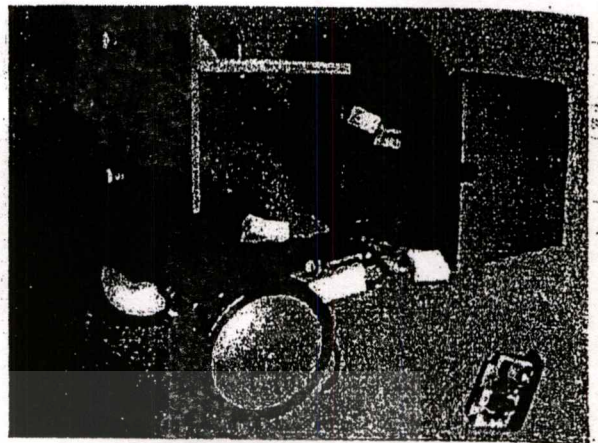


Fig. 3 Procedure for assembling mechanical modules of the robot body



(b) The whole module as component LEGO-like blocks.

Fig. 1 Major components for a soccer robot player (top-view)

3. Assembling Mechanical Parts and Components

For mechanical parts, a mobile robot as a soccer player is first equipped with a set of TAMIYA gearbox. The procedure of installation is guided in the user manual that comes with the gear set. It is optional to install an encoder obtained from a PC mouse hooked with that gearbox. Note that an old PC mouse can be re-used for this purpose. Figures 2-4 form a series of installation guide, looking from various angles of viewpoints.

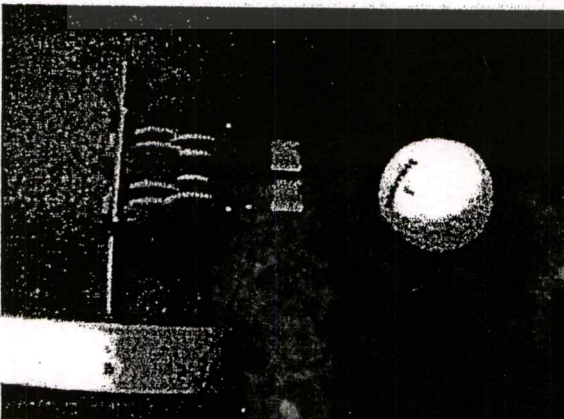


Fig. 2 Motors and a set of gearbox after assembling

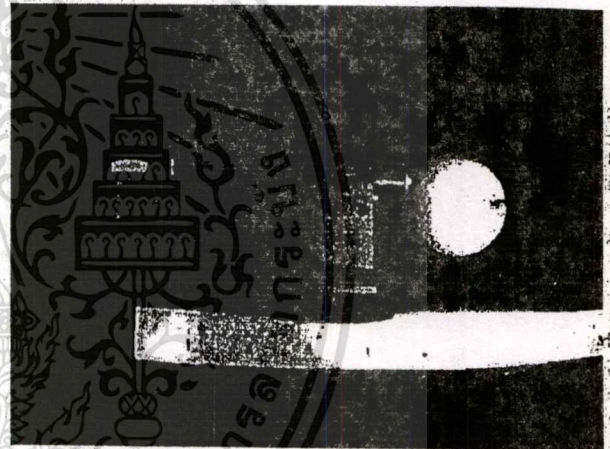


Fig. 4 Top view of a robot soccer player after complete assembling

4. Assembling Electronic Parts and Components

A major electronic part of the control unit is an AT89C2051 IC functioned as a processor. This IC receives data or signals from the encoder and from TFDS4500 IC that obtained data by means of infrared from the PC. Once an appropriate control action is fired, the control signal fetches the command to L293D IC to drive motors. Figure 5 provides a block diagram for easier understanding of this description.

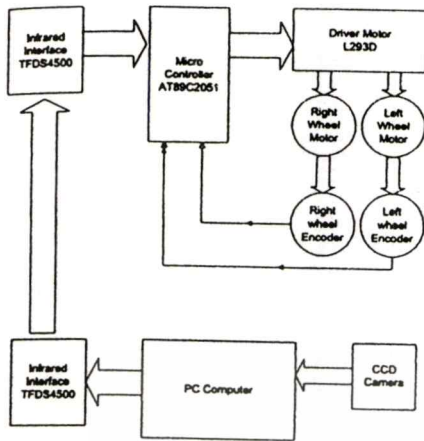


Fig. 5 Block diagram for the electronic part of the robot player

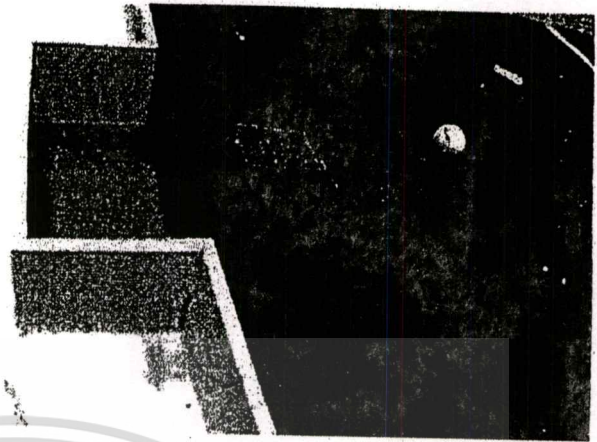


Fig. 7 A low-cost soccer robot team built by the authors

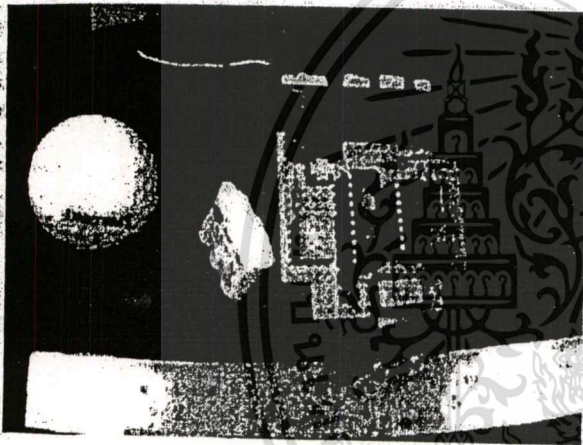


Fig. 6 A final look of the electronic module after assembling

After assembling both mechanical and electronic parts, a mobile robot as a soccer player is ready for the mission. Like the installation of a robot soccer team, the control and transmitting units are connected to the PC and the next task is to assign an address to each robot. The team is then ready for a rehearsal! Figure 7 is a real soccer robot team built by the authors.

5. Conclusions

We have described how to implement and build a soccer robot team for beginners with a very low cost. Components and parts along with estimating costs for building such a soccer robot team are presented. Assembling processes for both mechanical and electronic modules have also been illustrated. It is the authors' believe that this is excellent for education such as high school learning.

Acknowledgment

This works is supported in part by Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand.

REFERENCES

- [1] <http://www.fifa.com/>
- [2] <http://www.fira.net/>
- [3] <http://www.robocup2002.org/>
- [4] D. S. Kwon, J. H. Oh, Y. K. Kwak, J. J. Lee, J. H. Hwang, H. R. Kim, J. W. Lee, S. K. Hong and S. G. Hong, "A novel interactive robot soccer system," *Advanced Intelligent Mechatronics, 2001. Proceedings. 2001 IEEE/ASME International Conference on*, vol. 2, pp. 1082-1086, 2001
- [5] C. F. Marques and P. U. Lima, "Vision-based self-localization for soccer robots," *Intelligent Robots and Systems 2000 (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, vol. 2, pp. 1193-1198, 2000

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายจตุรวิทย์ จันไพบูลย์
วัน เดือน ปีเกิด	12 กุมภาพันธ์ 2523 ที่อำเภอเมือง จังหวัดนครศรีธรรมราช
ที่อยู่	59/12 หมู่ 6 ซอยชินเขต 2/37 ถนนงามวงศ์วาน แขวงทุ่งสองห้อง เขตหลักสี่ กรุงเทพฯ 10210 โทร.0-2591-6964
ประวัติการศึกษา	2537 มัธยมศึกษาตอนต้น โรงเรียนเบญจมราชูทิศ จ.นครศรีธรรมราช 2539 มัธยมศึกษาตอนปลายสายวิทยาศาสตร์ โรงเรียนสารวิทยา กทม. 2543 วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมระบบควบคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ความชำนาญเฉพาะด้าน	1.) ระบบควบคุม 2.) ระบบหุ่นยนต์ 3.) ระบบการมองเห็น (Vision Systems)
ประสบการณ์การทำงานและผลงานวิจัย	
พ.ศ. 2544-2546	ตำแหน่งวิศวกรบริษัทเทคโนโลยีคอมเอเชีย จำกัด - ผลงานการออกแบบตัวควบคุมเปิด-ปิดไฟในตู้โทรศัพท์ - ผลงานการออกแบบต้นแบบเครื่องโทรสารสาธารณะ
พ.ศ. 2547-ปัจจุบัน	ตำแหน่งนักวิจัยสังกัดหน่วยปฏิบัติการวิจัยเทคโนโลยีอิเล็กทรอนิกส์ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ - ผลงานการออกแบบเครื่องวัดกำลังรถยนต์ - ผลงานการออกแบบตัวควบคุมอิเล็กทรอนิกส์ของเครื่องยนต์ก๊าซธรรมชาติอัด
รางวัลเกียรติคุณ	
พ.ศ. 2544	รางวัลความคิดสร้างสรรค์ Nectec Programming Contest 2001, type Software Game
พ.ศ. 2545	รางวัลชนะเลิศ Honda Econo Power Contest 5 th 2002, type 125 cc
พ.ศ. 2546	รางวัลรองชนะเลิศ The Second International Robot Contest 2003, National University of Laos
พ.ศ. 2547	รางวัลเทคนิคยอดเยี่ยม Robocup Thailand Championship 2004
พ.ศ. 2548	รางวัลรองชนะเลิศ Robocup Thailand Championship 2005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้