



ปีการศึกษา 2533

ROBOT VISION



ภาควิชาระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
027923
18 ก.ค. 2534

ปีการศึกษา 2533

ภาควิชาระบบควบคุม

คณะวิศวกรรมศาสตร์

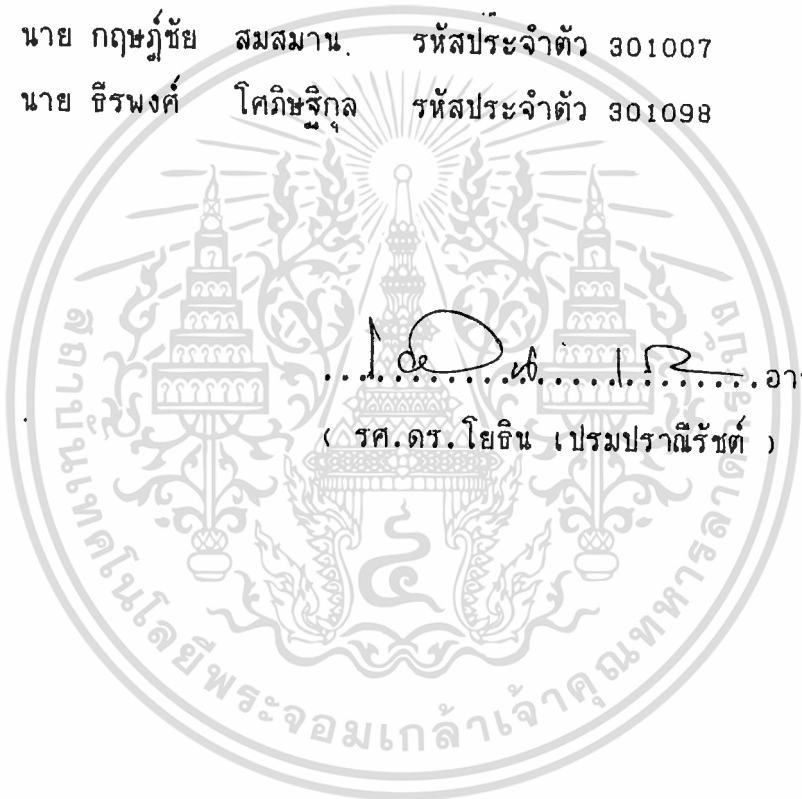
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

เรื่อง ROBOT VISION

ผู้จัดทำ นาย กฤษณ์ โกสวัสติ รหัสประจำตัว 301006

นาย กฤษณ์ชัย สมสมาน รหัสประจำตัว 301007

นาย ชีรพงศ์ ไคมิชจิกุล รหัสประจำตัว 301098



..... อาจารย์ที่ปรึกษา
(รศ.ดร. โยธิน เปรมปราณีรัตน์)

เลขหมุด T 3309074
เลขทะเบียน 027923
วัน, เดือน, ปี ๑๕-๑-๓๔

027923

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ROBOT VISION

นาย กฤษณ์ โกสวัตดี

นาย กฤษณ์ชัย สมสมาน

นาย ชีรพงศ์ ไชยชุกุล

รศ.ดร. โยธิน เปรมปราณีรัตน์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2539

บทคัดย่อ

ปฏิญานินพนธ์นี้เป็นการศึกษาและทดลองในเรื่องการรับรู้ การวิเคราะห์ข้อมูลภาพ และการบ่งชี้ถึงชนิดของวัตถุ เพื่อนำมาใช้ในการควบคุมการทำงานของอุปกรณ์ต่างๆ เช่น ควบคุมการทำงานของแขนกล ส่วนประกอบที่สำคัญของโครงงาน คือ แผงวงจร ติดต่อระหว่างกล้องถ่ายภาพวิดีโอ กับเครื่องคอมพิวเตอร์ ซึ่งทำหน้าที่รับสัญญาณภาพ จากกล้องถ่ายภาพวิดีโอ มาแปลงให้เป็นสัญญาณเชิงเลข เพื่อให้เครื่องคอมพิวเตอร์ สามารถประมวลผลได้ การบ่งชี้ถึงชนิดของวัตถุ ในปฏิญานินพนธ์ฉบับนี้ได้นำเสนอไว้ สองวิธีด้วยกัน วิธีแรก ใช้การหาส่วนประกอบของขอบภาพ และวิธีที่สอง ใช้การประมวลผลในโดเมนการแปลง โดยทั้งสองวิธีนั้น ก่อนอื่นจะต้องผ่านกระบวนการประมวลผลข้อมูลภาพเบื้องต้นเสียก่อน ซึ่งได้แก่ การปรับปรุงคุณภาพของภาพ การแปลงให้เป็นภาพที่มีระดับความเข้มสองระดับ และการหาขอบภาพ แล้วจึงค่อยใช้วิธีการทั้งสองวิธี ดังกล่าวข้างต้น ในการบ่งชี้และจำแนกประเภทของวัตถุ ออกเป็นวัตถุประเภทต่างๆ ตามที่กำหนดเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ROBOT VISION

KRIT KOSAWAT

KRITSACHAI SOMSAMAN

THEERAPONG SOPITHIKUL

ASSC.PROF.DR. YOTHIN PREMPRANEERACH ADVISOR

1990

ABSTRACT

Obtaining significant data from the interesting objects , by image analyzing and object recognition technique , is the main objective of this Robot Vision project. Digital data acquired by digitizing process of analog signal from video camera , will be treated in order to extract the required object parameter and supply them to the controller so that the necessary action can be planned. Interfacing card, or digitizer, plays an important role in making video camera and computer communicated each other , performing further computer processing. A few basic method such as filtering , binarizing and segmentation , will be introduced and two developed techniques in object recognition will be illustrated. Fourier descriptors technique , based on transform domain, and another technique called edge-component-analyzing method , carried out under spatial domain , can either make the desirable scheme succeeded.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

<u>เรื่อง</u>	<u>หน้า</u>
1. บทนำ	1 - 1
2. แฉ่นวงจรติดต่อข้อมูลภาพ	2 - 1
2.1 สัญญาณภาพ	2 - 1
2.2 ส่วนประกอบของแฉ่นวงจรและการทำงาน	2 - 9
3. การประมวลผลข้อมูลภาพ	3 - 1
3.1 ข้อพิจารณาเบื้องต้น	3 - 1
3.2 การปฏิบัติการประมวลผลข้อมูลภาพ	3 - 4
3.3 การปฏิบัติการทางด้านโครงสร้าง	3 - 8
4. การมองภาพโดยวิธีแยกส่วนประกอบของขอบภาพ	4 - 1
4.1 หลักการเบื้องต้น	4 - 1
4.2 ส่วนประกอบของโปรแกรม	4 - 2
4.3 การทำงานของโปรแกรม	4 - 3
5. การจำแนกวัตถุโดยวิธีฟูรีเยอร์ เดสคริปเตอร์	5 - 1
5.1 การแปลงฟูรีเยอร์	5 - 1
5.2 คุณสมบัติของการแปลง	5 - 4
5.3 การจำแนกวัตถุ	5 - 6
5.4 โปรแกรมการจำแนกวัตถุ	5 - 8
6. การทดลองและผลการทดลอง	6 - 1
6.1 การทดลองเก็บภาพด้วยแฉ่นวงจรติดต่อข้อมูลภาพ	6 - 1
6.2 การทดลองการแยกส่วนประกอบของขอบภาพ	6 - 5
6.3 การทดลองการแปลงฟูรีเยอร์	6 - 15
7. สรุปผลการทดลอง	7 - 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง

หน้า

8. ภาคผนวก	
8.1 ภาคผนวก (ก)	ก - 1
8.2 ภาคผนวก (ข)	ข - 1
8.3 ภาคผนวก (ค)	ค - 1
9. กิตติกรรมประกาศ	ง - 1
10. หนังสืออ้างอิง	จ - 1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีเอารนำไปได้

บทที่ 1

บทนำ

ในปัจจุบัน การใช้หุ่นยนต์ในงานด้านต่างๆได้เป็นที่แพร่หลายเป็นอย่างมาก ทั้งในด้านการศึกษาวิจัย และ การใช้งานทางอุตสาหกรรม หุ่นยนต์ ได้ถูกนำมาใช้แทนมนุษย์ในงานที่ต้องอาศัยความละเอียด งานที่จำเจ และในงานที่เป็นอันตราย

แต่ข้อจำกัดของ หุ่นยนต์ ส่วนใหญ่ในปัจจุบันก็คือ มันยังไม่สามารถคิดได้ด้วยตนเอง และความพยายามที่จะให้ถึงจุดนั้นได้ ก่อนอื่น หุ่นยนต์ จะต้องมีความสามารถในการรับรู้สภาพแวดล้อมในการทำงานของตัวเองเสียก่อน นั่นคือ จะต้องมิตัวตรวจจับ (sensor) ที่มีประสิทธิภาพ และมีการนำข้อมูลที่ได้มาประมวลผลด้วยวิธีการที่เหมาะสม

โครงการนี้ เป็นการนำเสนอแนวทางหนึ่งในการรับรู้สภาพแวดล้อมของหุ่นยนต์ ด้วยการรับรู้ภาพ โดยอาศัยการทำงานของกล้องวิดีโอ ทำหน้าที่เป็นตัวตรวจจับ แล้วส่งผ่านสัญญาณที่ได้ไปยังตัวแปลงสัญญาณเชิงอุปมานเป็นเชิงเส้น (Analog to Digital Converter) เพื่อแปลงข้อมูลให้เป็นแบบเชิงเลข ทำการเก็บไว้ในหน่วยความจำ ซึ่งในส่วนนี้ได้จัดสร้างเป็นแผงวงจรติดต่อ (Interface Card) สำหรับต่อกับเครื่องคอมพิวเตอร์ในตระกูลไอบีเอ็มพีซี (IBM PC) รายละเอียดในส่วนนี้ ได้ทำการอธิบายไว้ในบทที่ 2 จากนั้นคอมพิวเตอร์ก็จะทำการประมวลผลด้วยกรรมวิธีต่างๆโดยเริ่มจากการปรับปรุงคุณภาพของภาพที่ได้ การเปลี่ยนแปลงรูปแบบของข้อมูลให้อยู่ในรูปแบบที่จะเป็นประโยชน์ต่อการใช้งานต่อไป ซึ่งจะมีวิธีการอยู่หลายรูปแบบดังที่ได้อธิบายไว้ในบทที่ 3 ในเรื่องของการประมวลผลข้อมูลภาพ

การทำให้คอมพิวเตอร์สามารถเข้าใจภาพ และสามารถบ่งชี้วัตถุได้นั้น อาจทำได้ด้วยการคำนวณตำแหน่ง ขนาด ลักษณะการวาง และแยกแยะรูปทรงของวัตถุต่างๆออกจากกัน นอกจากนั้นก็ยังมีวิธีการอื่น ๆอีกหลายวิธี ซึ่งเรียกว่า การจดจำรูปแบบของวัตถุ (Pattern Recognition) ในปริภูมิตวินนธ์ฉบับนี้ ได้นำเสนอวิธีการไว้ 2 วิธี คือ ในบทที่ 4 เป็นการใช้วิธีการแยกส่วนประกอบของขอบภาพ และในบทที่ 5 เป็นการวิเคราะห์โดยใช้ฟูรีเยอร์ เดสคริปเตอร์ ซึ่งรายละเอียดของผลการทดลองทั้งหมดได้แสดงไว้ในบทที่ 6

บทที่ 2

แผงวงจรติดต่อข้อมูลภาพ

(Image Interface Card)

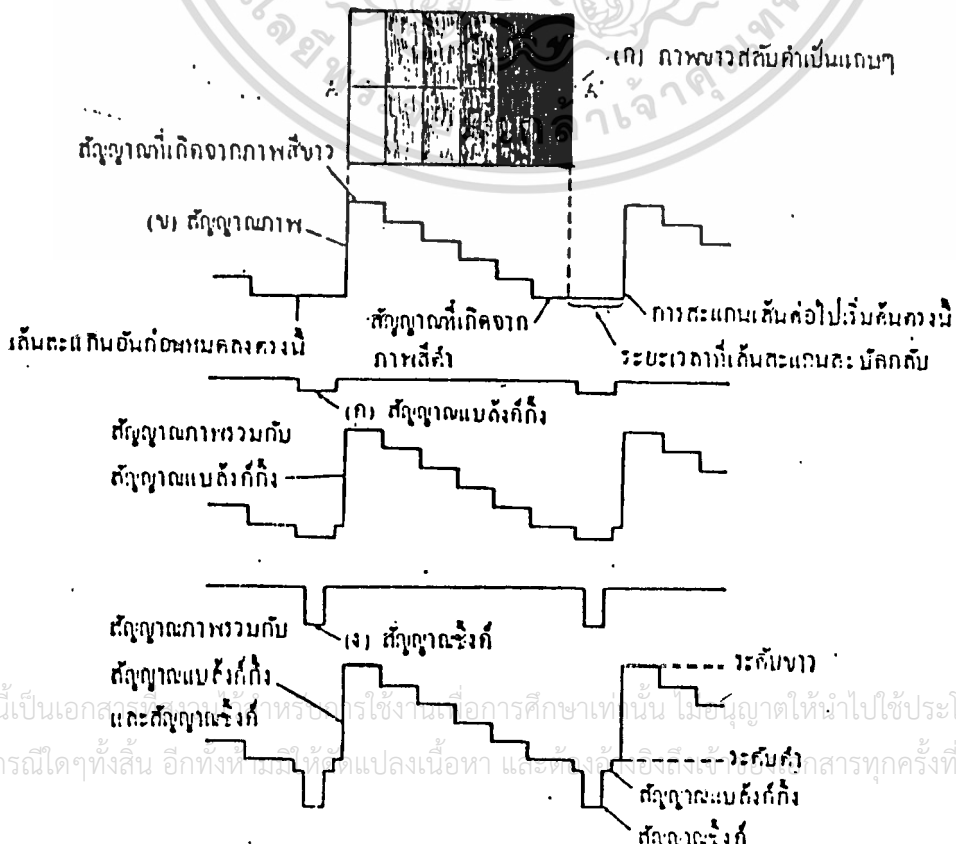
ก่อนอื่น จะขอกล่าวถึง ลักษณะของสัญญาณภาพในระบบโทรทัศน์อย่างคร่าวๆ เพื่อความเข้าใจในการอธิบายการทำงานในส่วนของวงจรต่อไป

2.1 สัญญาณภาพ

(Composite Signal)

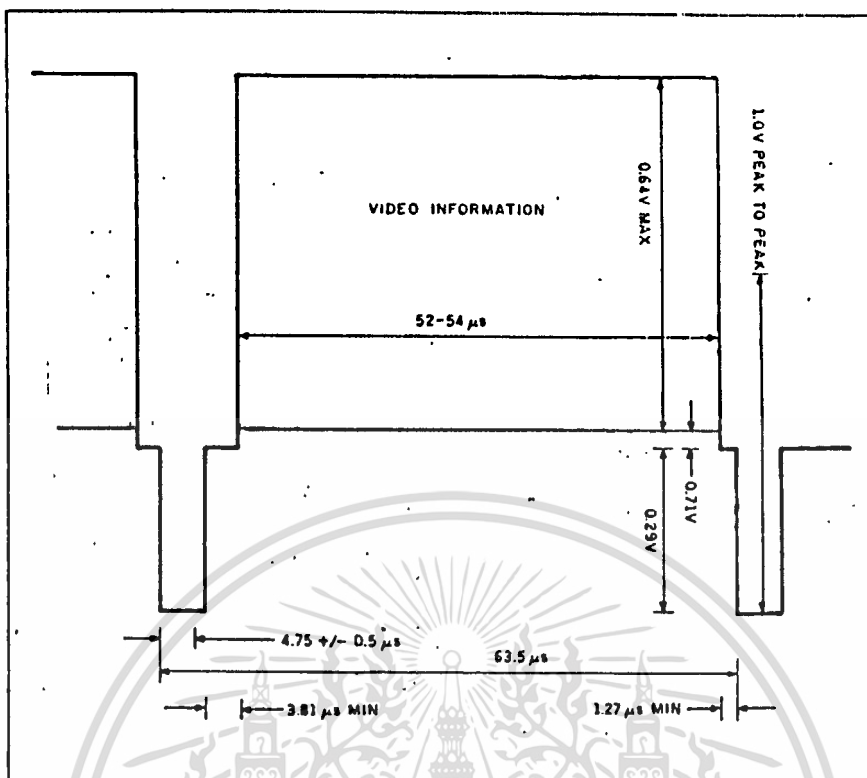
เป็นสัญญาณที่ได้จากกล้องโทรทัศน์ , วิดีโอ หรือสัญญาณจากเครื่องส่งโทรทัศน์ ที่ผ่านการแยกคลื่นพาหะออกจากสัญญาณภาพ และสัญญาณเสียงแล้ว

สัญญาณนี้จะประกอบด้วย สัญญาณความเข้มของแต่ละจุดภาพ ที่จะประกอบกัน เป็นภาพ , สัญญาณควบคุมการสแกนในแนวอนและแนวตั้ง ที่เรียกว่าสัญญาณซิงค์ และสัญญาณควบคุมการดับลำโวลิตรอนของหลอดภาพโทรทัศน์ ในขณะที่สลับกลับ เพื่อ ทำการสแกนเส้นใหม่ที่เรียกว่า สัญญาณแบล็งกิง ดังรูปที่ 2.1.1



เอกสารนี้เป็นเอกสารลิขสิทธิ์หรือการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

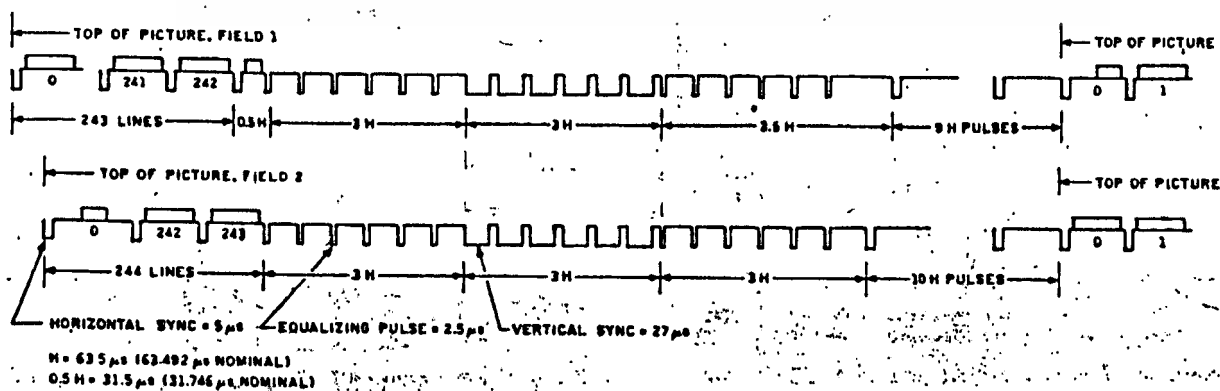
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามนำข้อมูลไปเผยแพร่หรือใช้ข้อมูลเอกสารทุกครั้งที่มีการนำไปใช้



A profile of the signal voltage sent to the display during a single scan line. Note the bracketing 63.5-microsecond horizontal sync pulses.

2.1.1 (b)

ระบบโทรทัศน์ในประเทศไทยนั้น จะเป็นระบบของยุโรปที่มีจำนวนเส้นในการสแกน 625 เส้น / ภาพ โดยจะทำการสแกน 25 ภาพ / วินาที ในการสแกนแต่ละภาพ (frame) นั้นจะประกอบด้วย 2 ส่วน (field) ส่วนละ 312.5 เส้น เป็น พิลด์ คี่ และคู่ ทำการสแกนแบบสลับเส้น (Interlaced Scanning) ดังรูปที่ 2.1.2



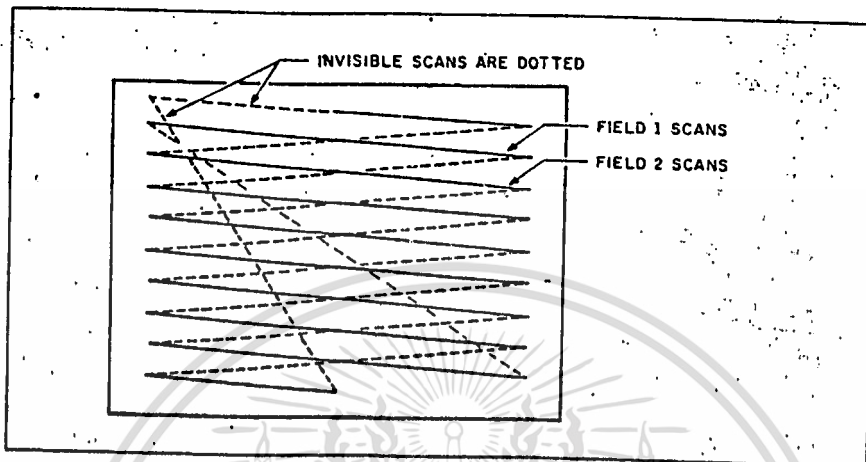
Signal voltage timing diagram for a complete field.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 (c)

รูปที่ 2.1.1 (a), (b), (c) ลักษณะภาพโทรทัศน์



The path of the electron beam for an interlaced video scan.

รูปที่ 2.1.2 การสแกนแบบสลับเส้น

2.2 ส่วนประกอบของแผ่นวงจร และการทำงาน

ในหัวข้อนี้จะกล่าวถึงส่วนฮาร์ดแวร์ของระบบ ซึ่งก็คือ แผ่นวงจรติดต่อข้อมูลภาพที่ใช้ในการแปลงสัญญาณโทรทัศน์ ซึ่งเป็นสัญญาณเชิงอุปมาน ให้เป็นสัญญาณเชิงเลขเก็บไว้ในหน่วยความจำสำรอง เพื่อให้ คอมพิวเตอร์สามารถดึงข้อมูล ไปประมวลผลได้ แผ่นวงจรนี้ได้ออกแบบเป็นการ์ด เพื่อเสียบลงในสล๊อต (slot) ของคอมพิวเตอร์ ในตระกูลไอบีเอ็ม พีซีและมีส่วนประกอบหลักๆดังในรูปที่ 2.2.1

วงจรการทำงานและอุปกรณ์ในแต่ละส่วนเป็นดังนี้

2.2.1 บัฟเฟอร์ (Buffer)

เป็นส่วนแรกของการ์ดที่ใช้ในการรับสัญญาณภาพ จากกล้องวิดีโอเข้ามา ดังแสดงในรูปที่ 2.2.2 เนื่องจากสัญญาณวิดีโอมักจะมีสัญญาณไฟดิซี (DC offset) ปะปนออกมาด้วย ทั้งนี้ขึ้นอยู่กับฉากที่กำลังถ่ายอยู่ ถ้าภาพที่ถ่ายค่อนข้างสว่าง ช่วงระดับสัญญาณจะต่ำกว่าภาพที่ค่อนข้างมืด เพื่อที่จะทำให้ช่วงระดับสัญญาณคงที่ จะต้องใช้วงจรแคลมป์ (Clamping Circuitry) เข้ามาเป็นส่วนประกอบหลักของบัฟเฟอร์ และจะได้สัญญาณที่ออกจากส่วนนี้มีค่าต่ำสุดของสัญญาณซิงค์ (sync pulse) อยู่ที่ระดับ 0 โวลต์

การดิจิติไลท์นี้เป็นการดิจิติไลท์แบบขาว-ดำ คือจะคำนึงถึงค่าความสว่างของแต่ละจุดเท่านั้นไม่คำนึงถึงสี แต่สัญญาณที่เข้ามาที่การ์दनนั้น อาจเป็นได้ทั้งสัญญาณภาพขาว-ดำหรือสัญญาณภาพสี ซึ่งภาพสีนั้นจะมีข้อมูลที่เกี่ยวกับสัญญาณสีอยู่ด้วย และอาจจะทำให้การดิจิติไลท์แบบขาว-ดำมีรอยซ้อนๆกัน ปัญหานี้สามารถแก้ไขได้ด้วยการใช้ตัวกรองสี (Color filter) ซึ่งจะไปลบแถบความถี่ที่ใช้ในการเข้ารหัสสีออกไป คงเหลือเพียงระดับแรงดันที่สำคัญ ที่บอกถึงค่าความสว่างของแต่ละจุดเท่านั้น แต่วิธีการนี้อาจทำให้สูญเสียรายละเอียดในเรื่องของความคมชัดลงไปบ้าง ดังนั้นจึงควรใส่ตัวเลือกไว้แล้วทดลองดูทั้งสองวิธี ว่าควรจะใช้ตัวกรองสีหรือไม่

ทรานซิสเตอร์ 2N 4401 เป็นตัวขยายกำลังเพื่อให้มั่นใจว่า วงจรในส่วนของเฮลิโอสจะได้รับสัญญาณที่สะอาด โดยไม่เป็นภาระทางด้านสัญญาณเข้ามากนัก

2.2.2 วงจรตรวจสอบสัญญาณซิงค์ (Sync Detector)

สัญญาณภาพในโทรทัศน์นั้นจะทำการสแกนทีละเส้น เมื่อสแกนจนสุดเส้นแล้วก็จะจะมีสัญญาณซิงค์ทางแนวนอน จากนั้นก็จะเริ่มสแกนเส้นใหม่ และเมื่อสแกนไปจนหมดฟิลด์แล้ว ก็จะมีสัญญาณซิงค์ทางแนวตั้ง จากนั้นก็จะเริ่มสแกนฟิลด์ใหม่

เพื่อให้การเก็บข้อมูล ลงในหน่วยความจำเป็นไปอย่างถูกต้อง เราจะต้องทำการตรวจสอบสัญญาณซิงค์ ทั้งทางแวนอนและแนวตั้ง เพื่อเอาไปใช้ในส่วนของวงจรนับอีกทีหนึ่ง

การตรวจสอบสัญญาณซิงค์ทางแวนอน ทำโดยการเปรียบเทียบระดับของสัญญาณ โดยใช้ไอซีเปรียบเทียบแรงดัน (Voltage Comparator) LM 311 ซึ่งจะให้ค่าเป็น 0 เมื่อสัญญาณต่ำกว่า 0.2 โวลต์ เนื่องจากเราเคลมปสัญญาณ ให้ระดับต่ำสุดของสัญญาณซิงค์เข้าใกล้ 0 ดังนั้น เอาท์พุทของ LM 311 จะมีค่า 0 เฉพาะช่วงเวลาที่เกิดสัญญาณซิงค์ทางแวนอนและแนวตั้งเท่านั้น ส่วนการตรวจสอบสัญญาณซิงค์ทางแนวตั้งทำได้โดยใช้วงจรกรองความถี่ต่ำผ่าน (low pass filter) กรองสัญญาณจากสัญญาณซิงค์ทางแวนอน

จากรูปที่ 2.2.3 เราใช้ โมโนสเตเบิล มัลติไวเบเรเตอร์ (Monostable Multivibrator) เพื่อหน่วงสัญญาณซิงค์ออกไปตามที่เรต้องการ เป็นการกำหนดจุดต้นของเส้นสแกนที่จะเริ่มเก็บข้อมูล

2.2.3 วงจรแปลงสัญญาณเชิงอุปมานเป็นสัญญาณเชิงเลข (Analogue to Digital Converter) Compare ธรรมดาได้

เนื่องจากเราต้องการเก็บสัญญาณภาพลงในหน่วยความจำ (ram) เราจึงต้องทำการแปลงสัญญาณภาพ ซึ่งเป็นสัญญาณเชิงอุปมาน ให้เป็นสัญญาณเชิงเลขเสียก่อน โดยอาศัยไอซีแปลงสัญญาณเชิงอุปมานเป็นเชิงเลข แต่เนื่องจากสัญญาณภาพมีความเร็วสูงมาก ประมาณ 6 MHz เราจึงต้องใช้ไอซีที่มีความเร็วในการแปลงสูงด้วย ในที่นี้เลือกใช้ แฟลชเอทีดี (Flash A/D) เบอร์ CA 3306 ซึ่งมีความเร็วในการแปลงสัญญาณ ประมาณ 12-18 MHz ได้สัญญาณเชิงเลขขนาด 6 บิต (bit) ซึ่งจะทำให้สามารถเก็บข้อมูลความสว่างของจุดภาพได้ 64 ระดับ

จากรูปที่ 2.2.4 แสดงให้เห็นถึงการใช้งานของ CA 3306 ซึ่งจะต้องมีสัญญาณเลือก (chip enable) เข้ามาจึงจะทำงาน และแรงดันอ้างอิงด้านบวกและด้านลบสามารถปรับได้ตั้งวงจรในรูป เราต้องวัดค่าของสัญญาณภาพที่เข้ามาว่า มีค่าแรงดันอยู่ในช่วงไหน แล้วปรับระดับแรงดันอ้างอิง ให้อยู่ในช่วงสัญญาณภาพที่ใช้งาน เนื่องจากเอทูดิตัวนี้เป็นแบบ 6 บิต แต่แรมที่ใช้เก็บข้อมูลเป็นขนาด 8 บิต เราจึงต้องเติมบิต 0 เข้าไปต่อท้ายอีก 2 บิต

2.2.4 วงจรนับ (Counter)

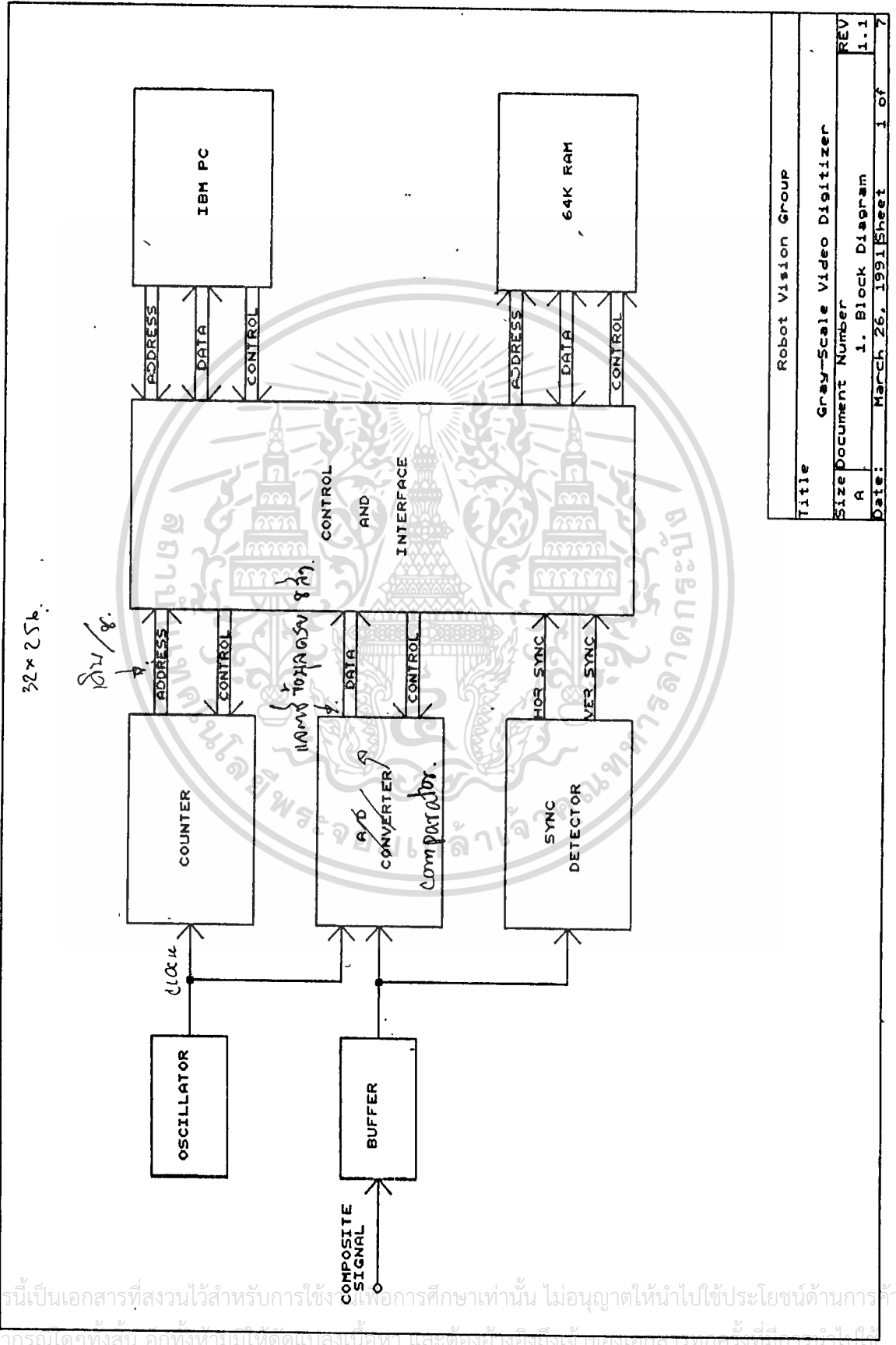
ทำหน้าที่ในการสร้างตำแหน่ง (address) สำหรับการเขียนข้อมูลลงในแรมขนาด 64 K จึงต้องมีแอดเดรสทั้งหมด 16 เส้น ดังรูปที่ 2.2.5 โดยใช้ไอซีเบอร์ 74LS191 (Synchronous up down Binary Counter) จำนวน 4 ตัว ทำงานประสานกัน ซึ่งจะทำให้ใน 1 เส้นสแกน แบ่งออกเป็นจุด 256 จุด และใน 1 ภาพจะมี 256 เส้น

2.2.5 หน่วยควบคุมและติดต่อ (Control and Interface)

ทำหน้าที่ควบคุมการทำงานของส่วนต่างๆ ให้สอดคล้องกัน โดยการสร้างสัญญาณนาฬิกาความถี่ 6 MHz ควบคุมการแปลงข้อมูลของเอทูดิ และยังควบคุมการติดต่อระหว่างเครื่องคอมพิวเตอร์ กับหน่วยความจำบนการ์ดด้วย ดังแสดงในรูปที่ 2.2.6

2.2.6 หน่วยความจำ (RAM)

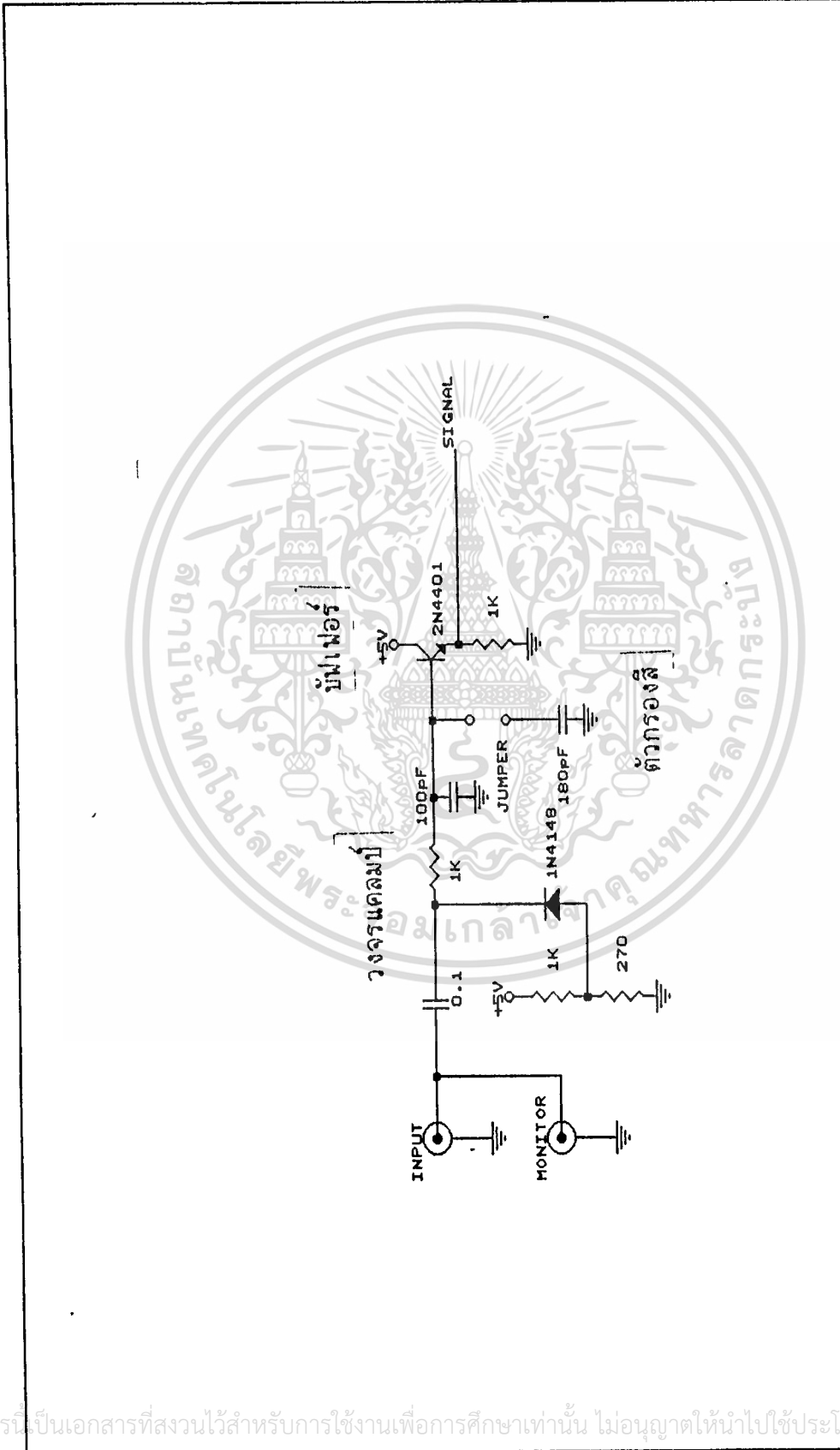
เป็นหน่วยความจำที่อยู่บนการ์ด ทำหน้าที่ในการเก็บข้อมูลภาพที่ได้แปลงมาเป็นสัญญาณเชิงเลขแล้ว จำนวน 64 กิโลไบต์ โดยใช้ไอซีเบอร์ 84256 ซึ่งเป็นแรม 32 K 8 บิต จำนวน 2 ตัว ดังแสดงในรูปที่ 2.2.7 และใช้สัญญาณควบคุมการอ่าน/เขียนจากหน่วยควบคุม



Robot Vision Group	
Title	Gray-Scale Video Digitizer
Size Document Number	A
REV	1.1
Date:	March 26, 1991 Sheet 1 of 7

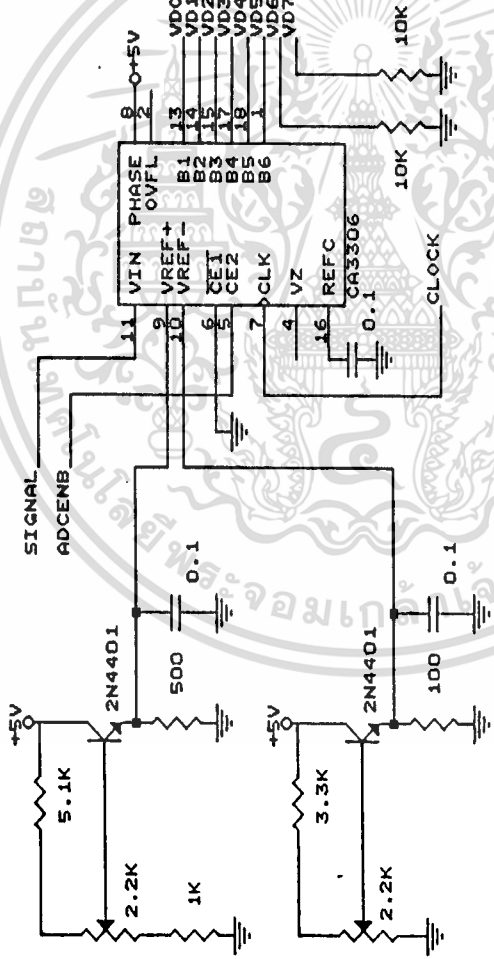
รูปที่ 2.2.1 ส่วนประกอบหลักของแผงวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อสาธารณะ และต้องยังต้องเก็บรักษาเอกสารทุกครั้งที่ได้รับมอบ



Robot Vision Group	
Title	Gray-Scale Video Digitizer
Size Document Number	2. Buffer Part
REV	1.1
Date:	March 23, 1991 Sheet 2 of 7

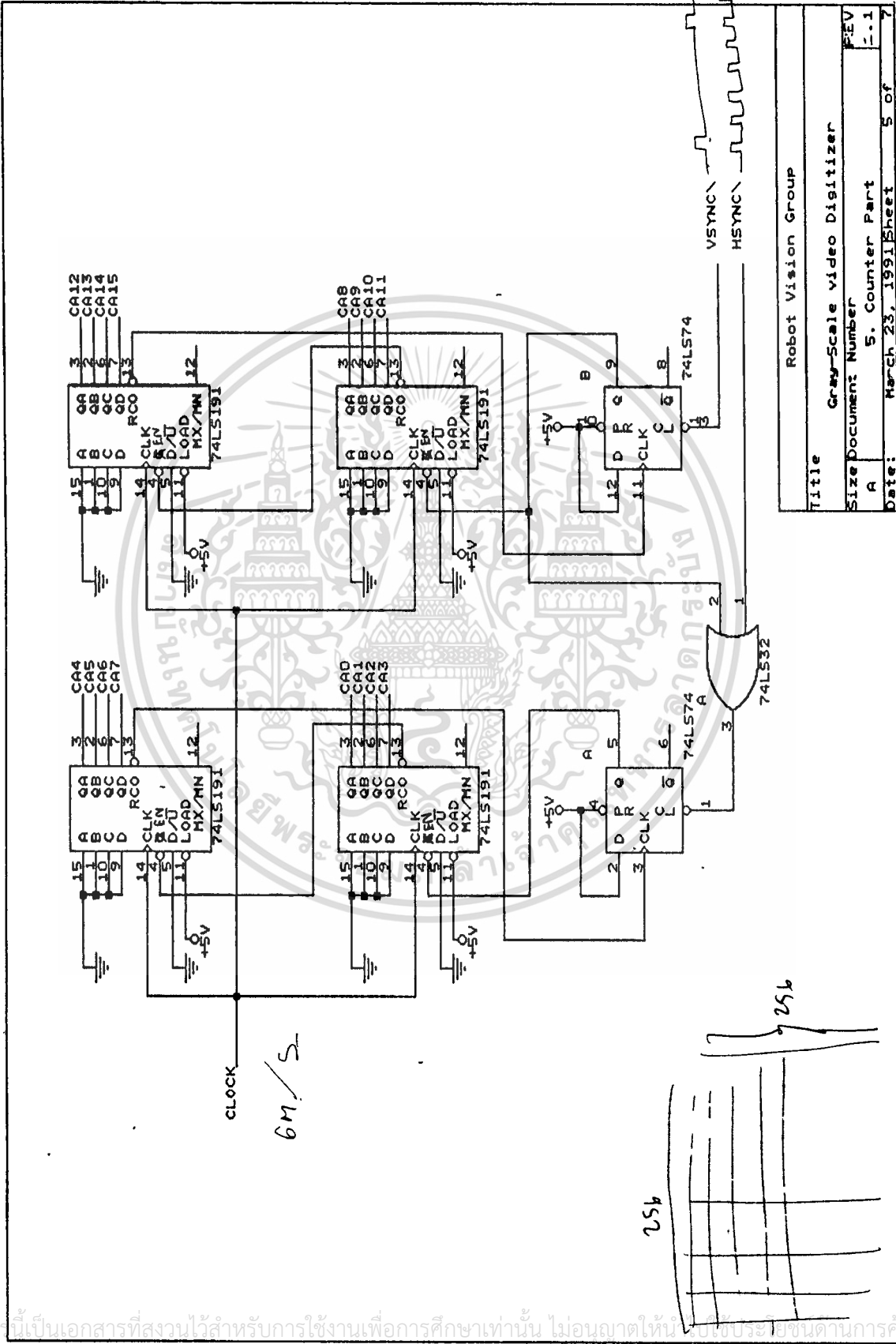
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Robot Vision Group	
Title	Gray-Scale Video Digitizer
Size Document Number	4. A/D Converter Part
REV	1-1
Date:	March 23, 1991
Sheet	4 of 7

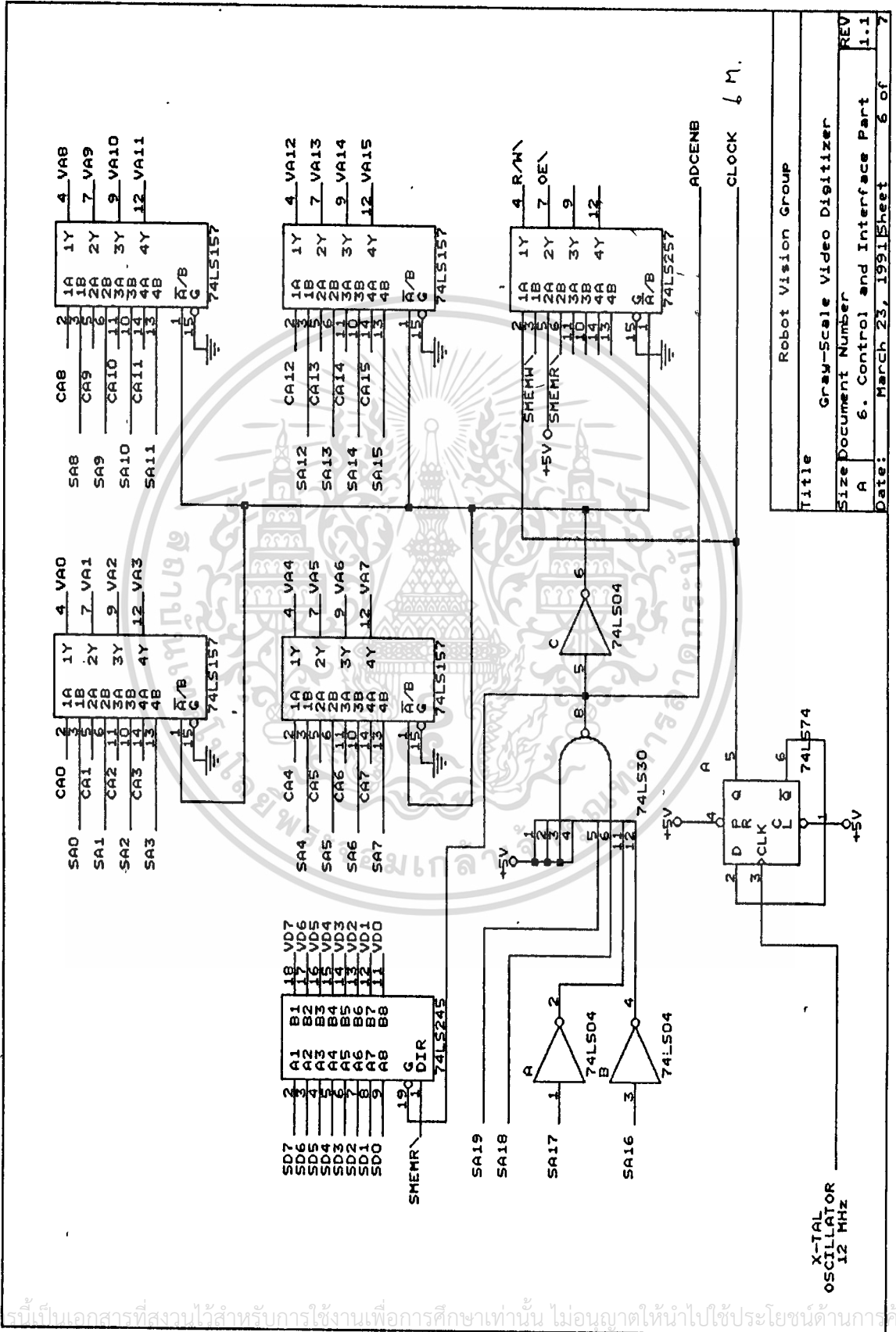
รูปที่ 2.2.4 วงจรแปลงสัญญาณวีดิโอมาเป็นเชิงเลข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรณี เบื้อง



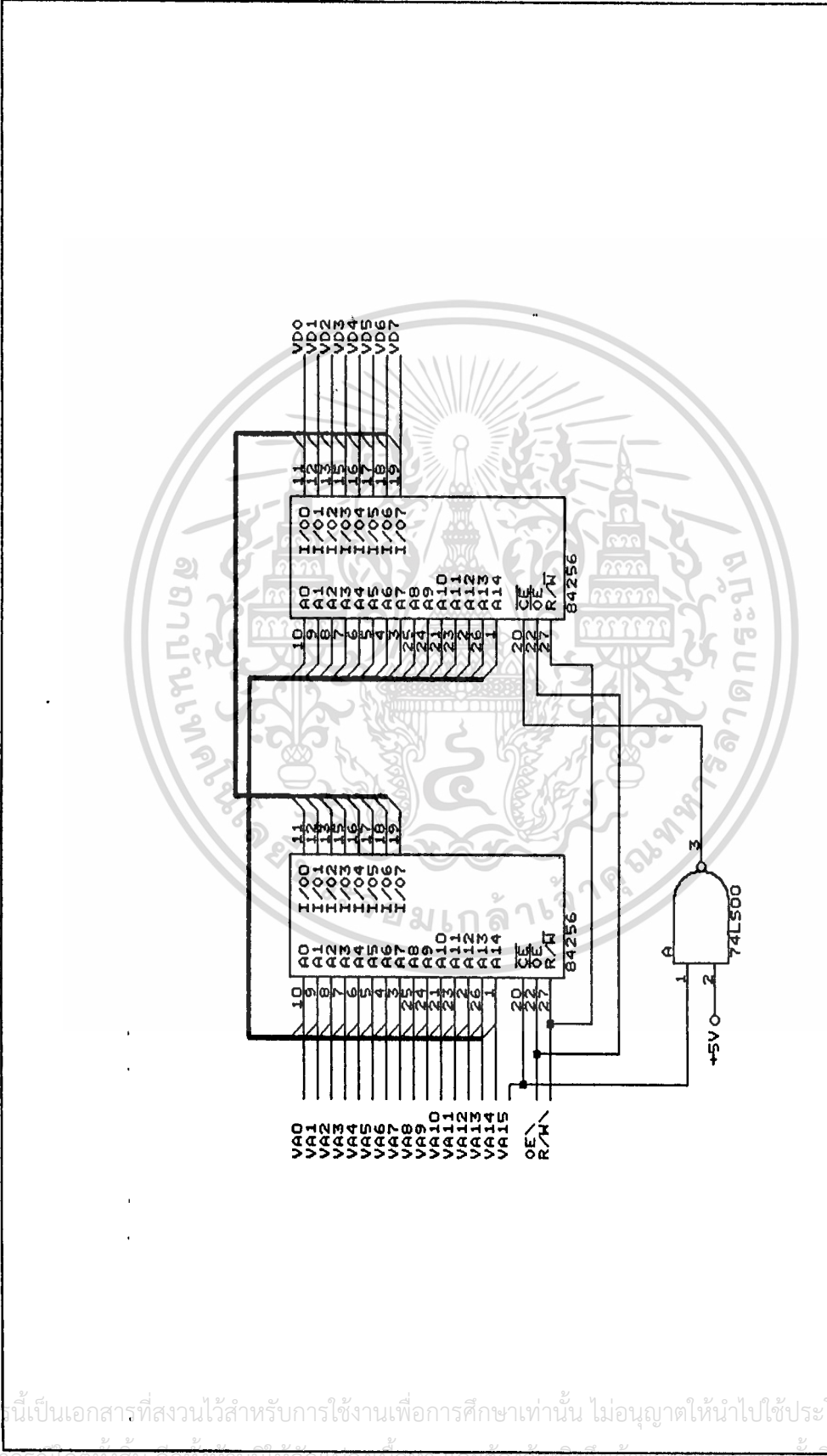
Robot Vision Group	
Title	Gray-Scale video Digitizer
Size	Document Number
A	5. Counter Part
Date:	March 23, 1991
Sheet	5 of 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้...
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Title		Robot Vision Group
Size Document Number		Gray-Scale Video Digitizer
Date:		March 23, 1991
REV		1.1
Sheet		6 of 7

รูปที่ 2.2.6 หน่วยควบคุมและรหัสตัว



Robot Vision Group	
Title	Gray-Scale Video Digitizer
Size Document Number	A
REV	1.1
Date:	March 23, 1991 Sheet 7 of 7

รูปที่ 2.2.7 หน่วยความจำ

บทที่ 3

การประมวลผลข้อมูลภาพ (IMAGE PROCESSING)

3.1 ข้อพิจารณาเบื้องต้น

3.1.1 ความหมายของอิมเมจ (IMAGE REPRESENTATION)

ถ้าเราจะพิจารณาจากจุดที่ง่ายที่สุด เราสามารถให้คำจำกัดความของ อิมเมจ (Image) ได้ว่าเป็นฟังก์ชัน 2 มิติ โดยที่ค่าของฟังก์ชัน $f(x,y)$ ที่โคออร์ดิเนต (x,y) ในระนาบ $x-y$ เป็นตัวกำหนดค่าความสว่าง (light intensity) ที่จุดนั้นๆ

แต่สำหรับกระบวนการที่จะต้องใช้คอมพิวเตอร์ประมวลผล เราจะสนใจ ข้อมูลภาพเชิงเลข (digital image) ซึ่งให้คำจำกัดความได้ว่า

ข้อมูลภาพเชิงเลข (digital image) คือ อิมเมจ ซึ่งถูกประมาณค่าใน 2 ทิศทาง

- การแบ่งจำนวนช่อง (Spatial digitization) บางครั้งจะหมายถึง การสุ่มตัวอย่างของค่าอิมเมจ (image sampling) ซึ่งก็คือ การแบ่งฟังก์ชันอิมเมจที่ต่อเนื่องหรือ ภาพใดๆ ออกเป็นเส้นวนเล็กๆ ทั้งหมด $M \times N$ ภาพ เรียกว่าส่วนย่อยของภาพ (picture element) หรือ พิกเซล(pixel)

- การแบ่งระดับ (Amplitude digitization) แต่ละพิกเซลใน อารีย์ จะต้องเข้ารหัส ค่าความสว่างเฉพาะจุด (local image intensity) และ ค่าระดับความสว่าง (image intensity level) จะถูกเรียกว่า ระดับสีเทา (gray level) และพิกัดสูงสุดที่เป็นไปได้ของ ระดับความสว่าง คือ อัตราสีเทา (gray scale) สำหรับพิกเซลใดๆ ในอารีย์ $M \times N$ จะมีระดับความสว่าง (g) ซึ่งกำหนดโดย

$$0 \leq f \leq 2^i - 1$$

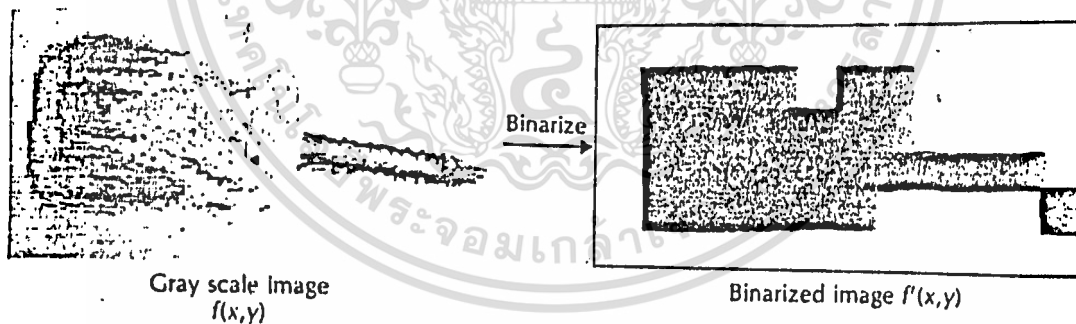
โดยที่ i คือ จำนวนบิตที่เราใช้งานในการแทนค่าอิมเมจ

3.1.2 ข้อมูลภาพแบบสองระดับ (BINARIZED IMAGES)

เป็นกรณีพิเศษของการแบ่งระดับ (amplitude digitization) ที่เกิดขึ้นเมื่อ อัตราสีเทา (gray scale) จะประกอบไปด้วย 2 ระดับที่เป็นไปได้เท่านั้น ($i = 1$) ภาพที่ออกมาจึงมีเพียง ดำ และ ขาวเท่านั้น ทำได้โดยการกำหนดค่า เทรดโวลต์ (thresholding value (T)) และพิจารณาค่าความสว่าง (intensity value) $f(x,y)$ ที่แต่ละจุดนั้น ดังรูปที่ 3.1.1

ถ้า $f(x,y) \geq T$ จะให้ $f'(x,y) = 1$

ถ้า $f(x,y) < T$ จะให้ $f'(x,y) = 0$



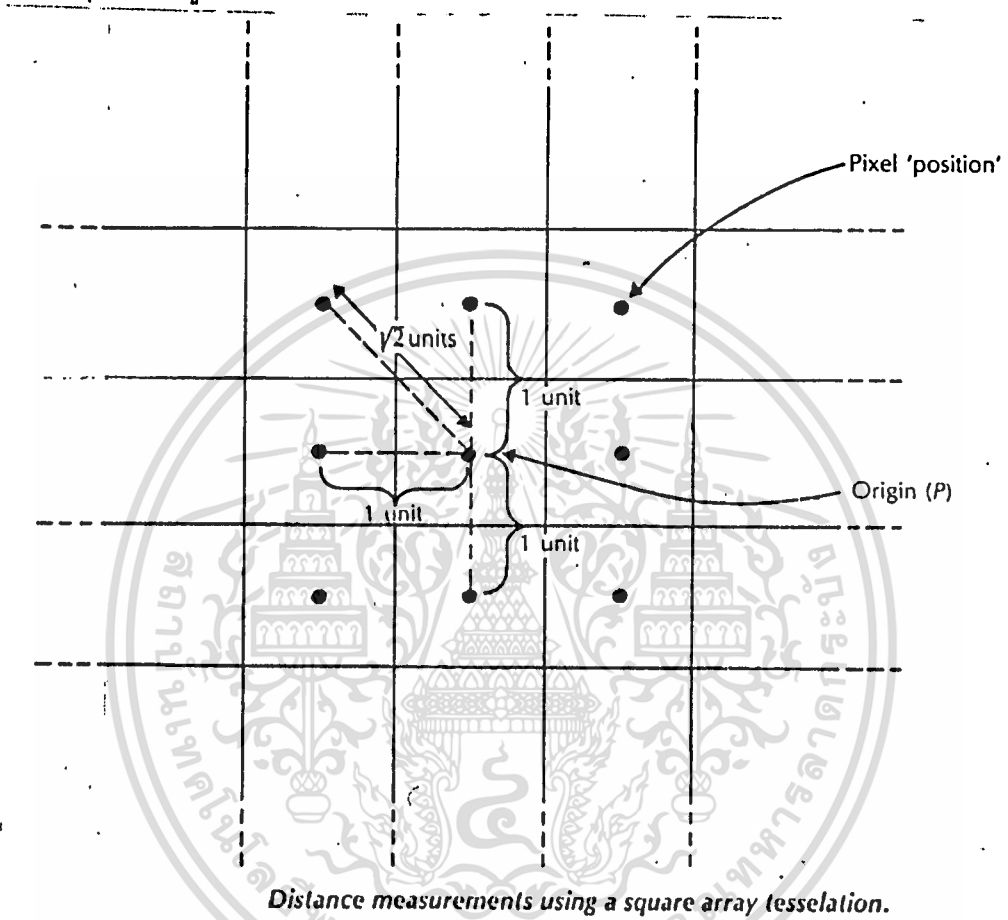
*Approximation of a gray scale image through binarization.
This also illustrates the effect of low-resolution spatial quantization.*

รูปที่ 3.1.1 ข้อมูลภาพแบบ 2 ระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 การคำนวณระยะทาง (DISTANCE MEASUREMENT)

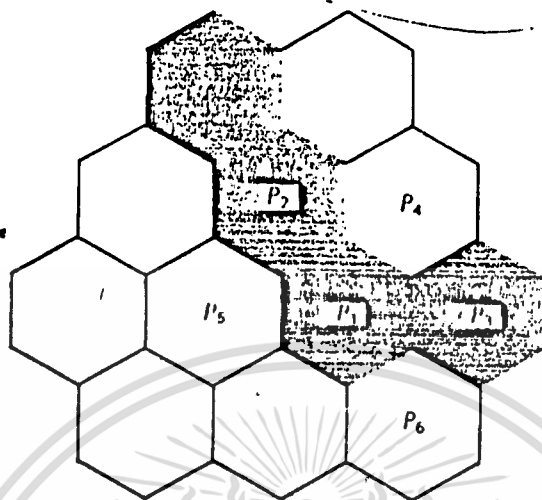
การวัดระยะห่างของแต่ละพิกเซลจะขึ้นอยู่กับลักษณะของพิกเซลนั้น ถ้าพิกเซลเป็นรูปสี่เหลี่ยมจัตุรัส ดังรูปที่ 3.1.2



รูปที่ 3.1.2 พิกเซลรูปสี่เหลี่ยมจัตุรัส

ถ้ากำหนดให้ระยะห่างระหว่าง 2 พิกเซล ที่อยู่ติดกัน เท่ากับหนึ่งหน่วย ตามทฤษฎีของ พิธากอรัส จะได้ว่าระยะห่างระหว่าง 2 พิกเซล ที่เฉียงกัน คือ $\sqrt{2}$ หน่วย

ถ้า พิกเซล เป็นรูปหกเหลี่ยม ระยะห่างของแต่ละ พิกเซล ที่อยู่ติดกัน จะเท่ากันทั้งหกด้าน คือ 1 หน่วย ดังรูป 3.1.3



รูปที่ 3.1.3 พิกเซลรูปหกเหลี่ยม

และ พิกเซล ที่ติดออกไป เช่น P_4 กับ P_5 เมื่อคำนวณตามกฎของ โคไซน์ แล้วจะมีระยะห่างเท่ากับ $\sqrt{3}$ หน่วย

3.2 การปฏิบัติการประมวลผลข้อมูลภาพ (IMAGE PROCESSING OPERATIONS)

ในงานส่วนใหญ่แล้ว ข้อมูลดิบที่ได้จากตัวตรวจจับ (sensor) มักจะมีคุณภาพต่ำ อันอาจจะเนื่องมาจาก อุปกรณ์ทำการสอบเทียบค่า (calibrate) มาไม่ดีพอ, มีสัญญาณรบกวน (noise) ในระบบหรืออยู่ในสภาพแสงสว่างที่ไม่ดี เพื่อที่จะปรับปรุงคุณภาพ จำเป็นจะต้องทำการแปลงข้อมูลดิบ (raw - image data) ด้วยวิธีทางที่เหมาะสม บางประการ

3.2.1 กระบวนการปรับปรุงโดยอิสระ (IMAGE ENHANCEMENT PROCESSING)

เป็นการแม็บ (map) โดยตรงระหว่างค่าของ พังก์ชันเดิม $f(x,y)$ มาเป็น พังก์ชันใหม่ $f'(x,y)$ จึงเป็นการเปลี่ยนแปลงข้อมูลเฉพาะของแต่ละพิกเซล โดยไม่ขึ้น กับพิกเซลอื่นๆที่อยู่รอบข้าง

-การเลื่อนระดับสีเทาอย่างเป็นเชิงเส้น (Linear graylevel shifting) เป็นการเลื่อนกราฟของฮิสโตแกรม (histogram) ทั้งหมด ออกไปที่ค่าระดับสีเทา อื่น

-การกระจายค่าฮิสโตแกรม (Enhancement through histogram flattening) เป็นการพยายามทำให้ กราฟของ ฮิสโตแกรม มีการแจกแจงความถี่ที่ สม่ำเสมอทั่วทุกระดับ

3.2.2 การปรับปรุงโดยขึ้นกับบริเวณโดยรอบ

(PROCESSING BASED ON LOCAL AREA COMPUTATION)

เป็นการเปลี่ยนแปลงข้อมูลใน พิกเซล โดยคำนึงถึงข้อมูลที่อยู่รอบๆ พิกเซล นั้น

$$f'(x_i, y_j) = \phi(P_{i,j}) \text{ สำหรับทุกค่า } i, j$$

วิธีนี้ จะทำให้ภาพที่ได้ มีความกลมกลืน ราบเรียบขึ้น (smooth) ซึ่งจะมี ประโยชน์ในการลดผลของความไม่ต่อเนื่อง และผลของสัญญาณรบกวนในอิมเมจ

3.2.3 การเปลี่ยนแปลงข้อมูลก่อนการประมวลผล

(PREPROCESSING FOR NORMALIZATION)

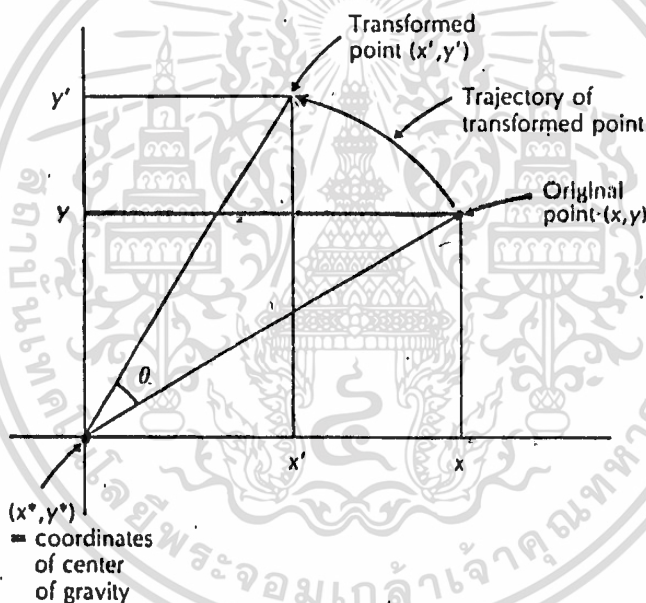
เป็นการปรับปรุงคุณภาพ โดยไม่ให้เกิดการเปลี่ยนแปลงคุณลักษณะของอิมเมจเดิม มากนัก

-การเลื่อนตำแหน่ง (Position normalization) เป็นการย้ายตำแหน่ง โคออร์ดิเนตโดยไม่ทำให้มีรูปร่างผิดไปจากเดิม โดยการใส่สูตร

$$x^* = \frac{\sum_{\text{all } y} \sum_{\text{all } x} x \cdot f(x,y)}{\sum_{\text{all } y} \sum_{\text{all } x} f(x,y)}$$

$$y^* = \frac{\sum_{\text{all } y} \sum_{\text{all } x} y \cdot f(x,y)}{\sum_{\text{all } y} \sum_{\text{all } x} f(x,y)}$$

- การหมุน (Rotation) เป็นการหมุนวัตถุเพื่อให้ได้ตำแหน่งที่เหมาะสม



Basic geometry of an image rotation algorithm.

รูปที่ 3.2.1 อัลกอริทึมของการหมุน

เป็นการแปลงตำแหน่ง (x, y) ไปเป็น (x', y') โดยใช้สูตร

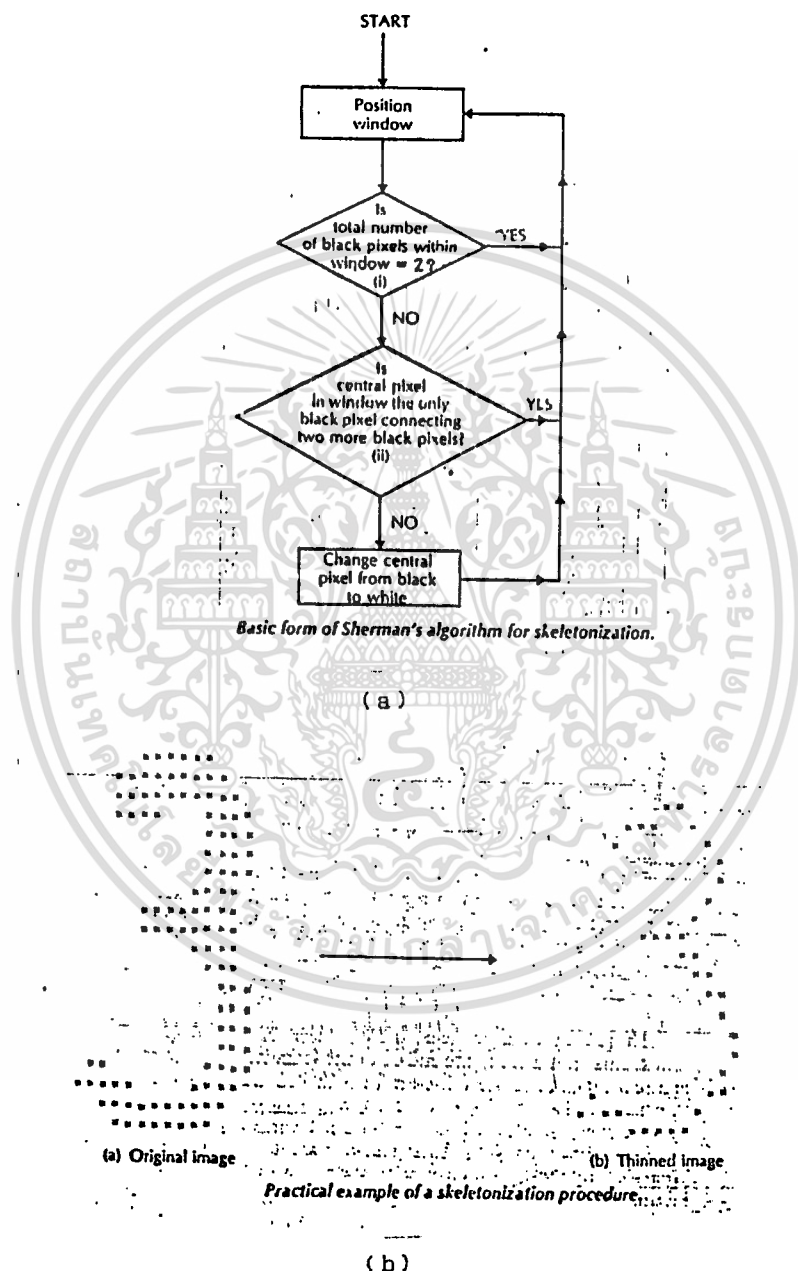
$$x' = x \cos\theta - y \sin\theta$$

$$y' = x \sin\theta + y \cos\theta$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 การลดขนาดเส้น (PATTERN THINNING)

เป็นการพยายามลดขนาดของเส้นที่หนาเกินความจำเป็น โดยใช้วิธีของ Sherman



รูปที่ 3.2.2 (a) อัลกอริทึมของการลดขนาดเส้น
(b) ผลจากการลดขนาดเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การปฏิบัติการทางด้านโครงสร้าง (STRUCTURAL OPERATIONS ON IMAGES)

การที่จะเข้าใจภาพได้นั้น ในบางครั้งไม่มีความจำเป็นที่จะต้องรู้ในรายละเอียด หรือ โครงสร้างภายในใดๆ ก็สามารถที่จะเข้าใจได้แล้ว ดังเช่น รูปที่มีเพียง โครงร่างอย่างรูปการ์ตูน (cartoon) หรือรูปที่มีระดับความสว่างเพียงแค่ 2 ระดับ (binarized object) บางรูปก็ไม่ยากแก่ความเข้าใจ ดังรูปที่ 3.3.1



รูปที่ 3.3.1 (a) รูปที่มีเพียงโครงร่าง
(b) รูปที่มีควมสว่าง 2 ระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นเพื่อให้สามารถประมวลผลได้ง่าย และ หุ่นยนต์สามารถเข้าใจได้ จึงต้องมีการวิเคราะห์ข้อมูลเหล่านั้น เพื่อแปลงให้อยู่ในรูปโครงร่าง หรือรูปที่มีความสว่าง 2 ระดับ ด้วยวิธีดังนี้

3.3.1 การหาขอบภาพ (EDGE DETECTION)

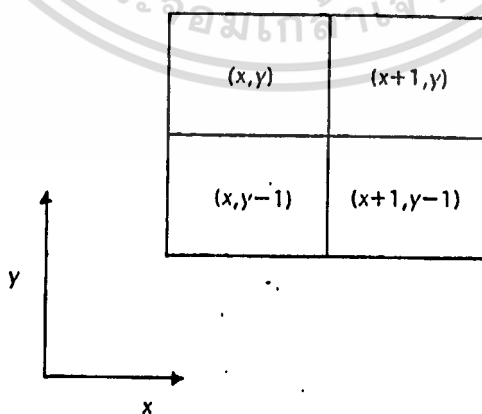
การหาขอบของภาพ เพื่อให้ได้โครงร่างคร่าวๆของภาพนั้นๆ พิจารณาได้จากจุดที่เป็นขอบ คือ จุดที่มีความแตกต่างของระดับความสว่างระหว่างวัตถุ(ซึ่งเป็นขอบ) กับสิ่งแวดล้อม(ซึ่งเป็นฉากหลัง)

3.3.1.1 ตัวปฏิบัติการหาขอบ (Edge detection operators)

วิธีการในการหาขอบนั้นจะอาศัยหน้าต่าง (window) ของพิกเซล ที่อยู่ในบริเวณเดียวกัน ดังเช่น

ตัวปฏิบัติการของโรเบิร์ต (Roberts Operator)

จะใช้หน้าต่างขนาด 2×2 ดังรูป 3.3.2



รูปที่ 3.3.2 หน้าต่างสำหรับตัวปฏิบัติการของโรเบิร์ต

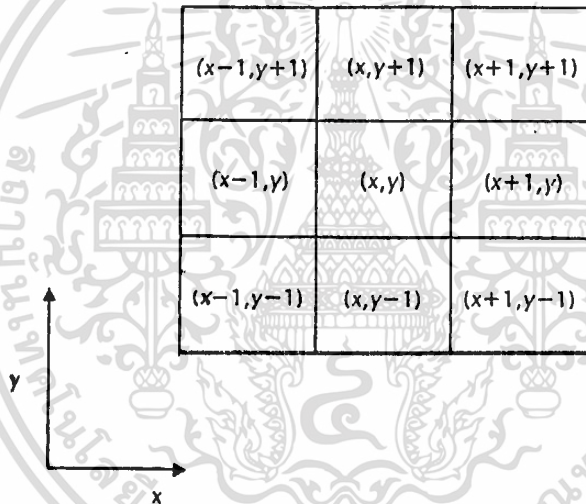
และมีสมการในการหาค่าเกรเดียนต์ ฟังก์ชัน (gradient function) ดังนี้

$$G(x,y) = \sqrt{ \{ f(x,y)-f(x+1,y-1) \}^2 + \{ f(x+1,y)-f(x,y-1) \}^2 }$$

วิธีนี้จะมีข้อเสียตรงที่ อาจจะได้รับการกระทบกระเทือนจากการเปลี่ยนแปลงความเข้ม อันเนื่องมาจากสัญญาณรบกวน เพื่อที่จะลดผลนี้ จำต้องใช้หน้าต่างที่มีขนาดใหญ่ขึ้น

ตัวปฏิบัติการของโซเบล (Sobel Operator)

จะใช้หน้าต่างขนาด 3*3 แสดงดังรูป 3.3.3



รูปที่ 3.3.3 หน้าต่างสำหรับตัวปฏิบัติการของโซเบล

และสามารถหาค่าเกรเดียนต์ ฟังก์ชัน (gradient function) ได้จาก

$$G(x,y) = \sqrt{ \{ (f(x+1,y+1)+2f(x+1,y)+f(x+1,y-1)) \\ - (f(x-1,y+1)+2f(x-1,y)+f(x-1,y-1)) \}^2 \\ + \{ (f(x-1,y-1)+2f(x,y-1)+f(x+1,y-1)) \\ - (f(x-1,y+1)+2f(x,y+1)+f(x+1,y+1)) \}^2 }$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.2 การพิสูจน์ว่าเป็นขอบ (Edge identification)

เมื่อเราได้ค่าเกรเดียนท์ ฟังก์ชัน มาแล้ว การพิจารณาว่า จุดนั้นสมควรจะเป็นขอบหรือไม่ โดยการกำหนดค่าเทรชโอล์ดของขอบ (edge threshold) แล้วทำการพิจารณาว่า

ถ้า $G(x,y) > \theta$ จุด (x,y) นั้นเป็นจุดบนขอบ

ถ้า $G(x,y) \leq \theta$ จุด (x,y) นั้นไม่ใช่จุดที่เป็นขอบ

3.3.2 การพิสูจน์ขอบเขต (REGION IDENTIFICATION)

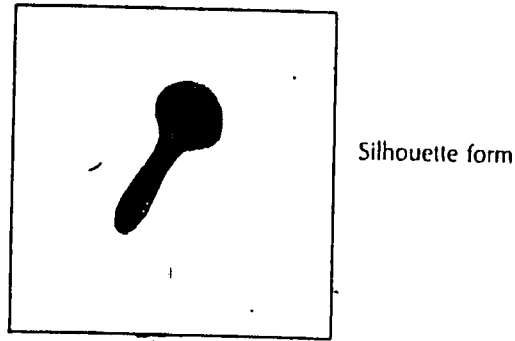
เป็นการทำให้ภาพนั้นมีความสว่างเพียงแค่ 2 ระดับ เรียกว่าภาพลักษณะเงาดำ (silhouette form) ทำได้โดยใช้ค่าเทรชโอล์ด เช่นเดียวกับข้อมูลภาพ 2 ระดับ

3.3.2.1 การวาดเส้นแสดงรูปร่าง (Regions and edges - contour tracing)

โดยการหาจุดใดๆจุดหนึ่งบนขอบ จากนั้นก็ทำการติดตาม (track) ตามจุดบนขอบไปเรื่อยๆ จนกระทั่งวนมาถึงจุดเริ่มต้นอีกครั้ง

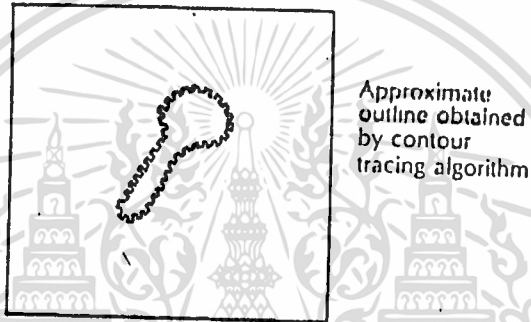
วิธีการนี้ อาจจะทำให้เกิดปัญหาเมื่อมีการเปลี่ยนแปลงทิศทางของขอบอย่างกะทันหัน หรือเป็นรูปที่มีความสลับซับซ้อน แต่สำหรับเหตุการณ์ที่เหมาะสมแล้ว วิธีการหาขอบวิธีนี้ใช้ได้ผลดีทีเดียว

ลักษณะของภาพเงาดำ และ การหาขอบโดยวิธีการวาดเส้นแสดงรูปร่าง แสดงไว้ในรูปที่ 3.3.4



(a)

Apply
contour
trace



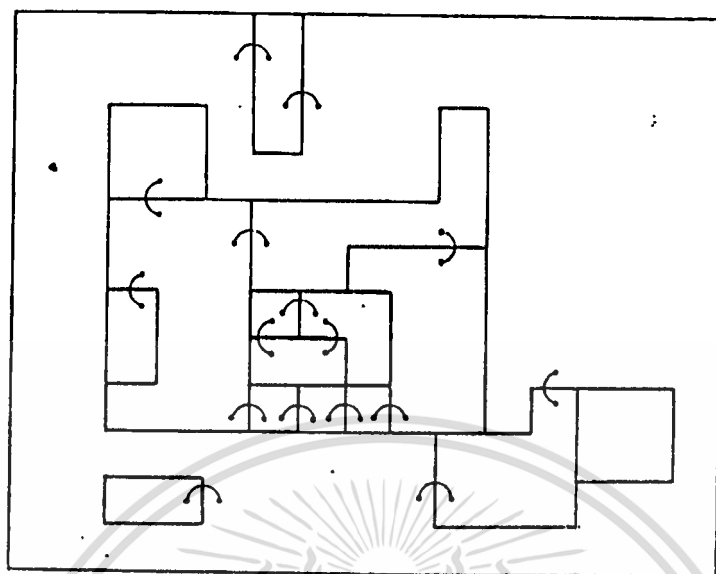
(b)

รูปที่ 3.3.4 (a) ลักษณะของภาพเงาดำ

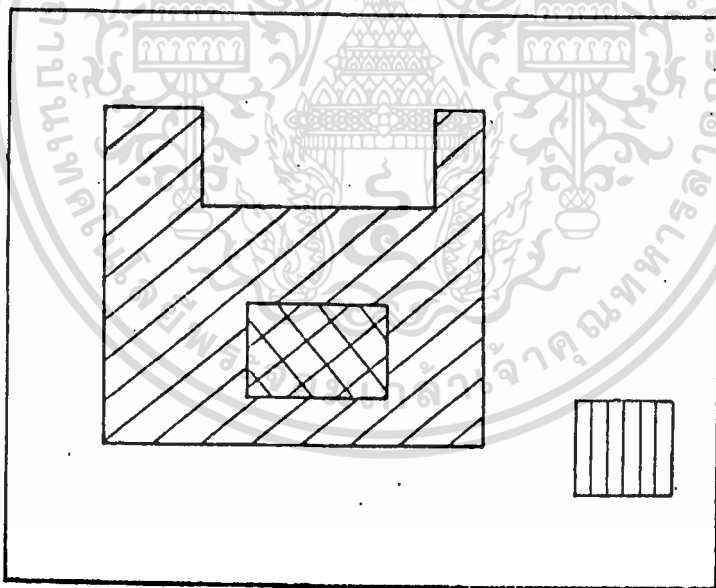
(b) การหาขอบภาพของภาพ (a)

3.3.3 การแบ่งขอบเขตอย่างกว้างๆ (GENERALIZED REGION SEGMENTATION)

สำหรับรูปที่มีความซับซ้อนมากๆ มีระดับสีเทา หลายระดับ การประมวลผลจะยุ่งยาก วิธีหนึ่งที่จะลดความซับซ้อนลง คือการกำหนดช่วงของระดับสีเทาให้ นิกเซล ที่อยู่ในช่วงระดับสีเทาเดียวกัน กำหนดให้มีความสว่างเท่ากัน วิธีนี้จะสามารถลดรายละเอียดของภาพลงได้ ดังรูป 3.3.5



(c)



(d)

Figure 4.10 Progression of steps in a region segmentation process.

รูปที่ 3.3.5 (ต่อ)

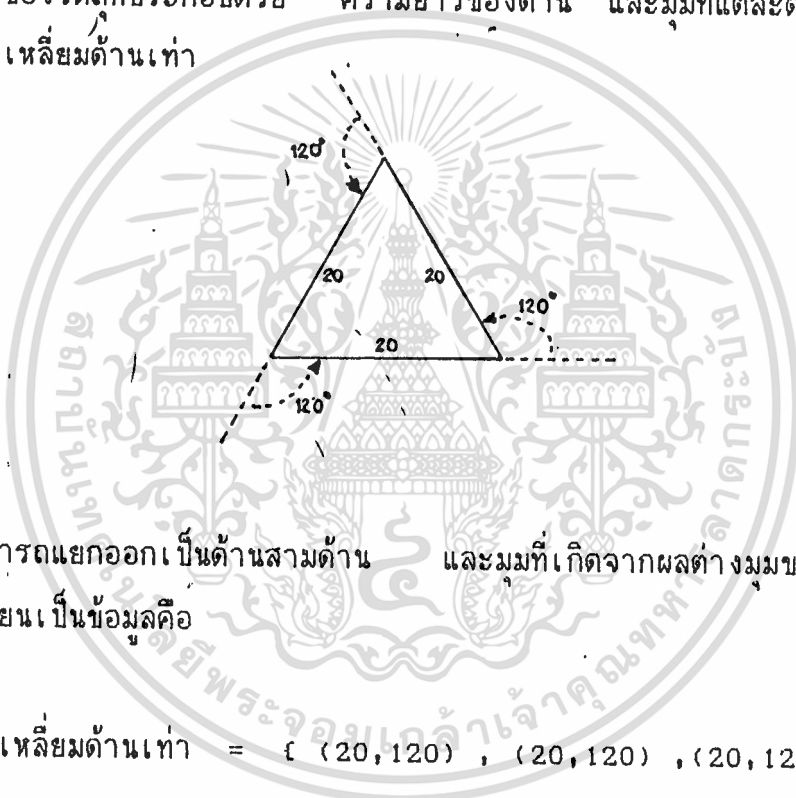
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การมองภาพโดยวิธีการแยกส่วนประกอบ ของขอบภาพ

4.1 หลักการเบื้องต้น

จุดหลักก็คือ ต้องทำการวิเคราะห์เส้นขอบของวัตถุที่ต้องการ โดยจะทำการแยกเป็นข้อมูลของวัตถุที่ประกอบด้วย ความยาวของด้าน และมุมที่แต่ละด้านสัมพันธ์กัน เช่น รูปสามเหลี่ยมด้านเท่า



จะสามารถแยกออกเป็นด้านสามด้าน และมุมที่เกิดจากผลต่างมุมของแต่ละด้าน ซึ่งสามารถเขียนเป็นข้อมูลคือ

$$\text{สามเหลี่ยมด้านเท่า} = \{ (20, 120), (20, 120), (20, 120) \}$$

ซึ่งจะเห็นได้ว่า ไม่ว่าจะรูปสามเหลี่ยมนั้นจะหันไปทางทิศใด ก็จะสามารถเขียนข้อมูลของวัตถุนั้นออกมาได้เหมือนเดิม

ดังนั้น ถ้าเราทำการเก็บข้อมูลของวัตถุต้นแบบไว้ แล้วมาทำการเปรียบเทียบกับวัตถุที่กำลังมอง เราก็จะสามารถทราบได้ว่า วัตถุที่กำลังมองนั้นเป็นวัตถุอะไร

และจากการที่เราไม่สามารถกำหนดได้ว่า ด้านไหนของวัตถุจะเป็นข้อมูลตัวแรก ดังนั้นข้อมูลที่ได้นี้ จะเป็นข้อมูลที่ไม่มีหัว-ท้าย คือเป็นข้อมูลชนิดวงแหวน (ring)

ซึ่งการเปรียบเทียบข้อมูลชนิดนี้ 2 ชุด จะต้องทำการหมุนข้อมูล (rotate) ด้วย

4.2 ส่วนประกอบของโปรแกรม

จากหลักการเบื้องต้น เราจะนำมาออกแบบโปรแกรม ให้สามารถทำงานตามที่ต้องการ โดยจะแยกเป็นส่วนต่างๆที่สำคัญ คือ

4.2.1 ส่วนหาขอบภาพ (edge_detect)

ทำการหาขอบภาพที่ได้จากการแปลงสัญญาณภาพ โดยใช้วิธี โซเบล (Sobel Operator) ขอบภาพที่ได้ี้จะมีลักษณะเป็นข้อมูลแบบต่อเนื่อง ที่มีค่าของข้อมูลบริเวณที่เป็นขอบภาพ (บริเวณที่มีความแตกต่างของระดับความเข้ม) สูงกว่าบริเวณอื่น

4.2.2 ส่วนตัดขอบภาพ (edge_track)

ข้อมูลที่ได้จากส่วนหาขอบภาพ จะเป็นข้อมูลแบบต่อเนื่อง ซึ่งจะยังไม่สามารถนำไปวิเคราะห์เส้นตรงได้ จึงต้องทำการแปลงให้เป็นเส้นขอบเพียงเส้นเดียว โดยใช้ในการติดตาม (track) ข้อมูลที่เป็นขอบภาพที่มีค่าสูงสุดในทิศทางที่เป็นไปได้ ซึ่งจะทำให้ได้เส้นที่เป็นขอบภาพเพียงเส้นเดียว

4.2.3 ส่วนวิเคราะห์เส้นตรง (line_analyze)

ส่วนนี้จะทำหน้าที่ในการตรวจสอบว่า เส้นที่เป็นขอบภาพนั้น จะแบ่งเป็นเส้นตรงได้กี่เส้น และแต่ละเส้นมีมุมเป็นเท่าไร

4.2.4 ส่วนกรองเส้นตรง (line_filter)

จากการทดลองปรากฏว่า เส้นตรงที่ได้จากส่วนตัดขอบภาพ ในส่วนที่เป็นมุมแหลม จะไม่ค่อยคมชัด คือมีลักษณะเป็นเส้นโค้ง จึงทำให้ส่วนวิเคราะห์เส้นตรง แยกออกมาเป็นเส้นเล็กๆอีก 1 เส้น จึงต้องทำการกรองเส้นเล็กๆนี้ออกไป

4.2.5 ส่วนสร้างข้อมูล (code_gen)

จะทำการนำเส้นตรงที่ผ่านการกรอง นำมาสร้างเป็นข้อมูลของเส้นที่เป็นส่วนประกอบของขอบภาพตามต้องการ

4.2.6 ส่วนเก็บข้อมูล (build_database)

จะทำการเก็บข้อมูลของวัตถุชนิดต่างๆ ไว้ในหน่วยความจำสำรอง

4.2.7 ส่วนเปรียบเทียบข้อมูล (look_object)

จะทำการเปรียบเทียบข้อมูลที่ได้จากการสร้างข้อมูล กับข้อมูลที่เก็บไว้กับหน่วยความจำสำรอง เพื่อดูว่าเป็นภาพของวัตถุอะไร

4.3 การทำงานของโปรแกรม

4.3.1 ส่วนหาขอบภาพ (edge_detect)

จะทำตามวิธีการของโซเบล คือ ข้อมูลที่ตำแหน่ง a จะเกิดจากข้อมูลรอบๆด้านของมัน ตามสูตร

$$a = \sqrt{[(c+2*b+i) - (e+2*f+g)]^2 + [(g+2*h+i) - (e+2*d+c)]^2}$$

โดยที่ค่าของ a คือค่าของข้อมูลที่ตำแหน่ง (x, y) และค่าของข้อมูลอื่นคือ

b = ค่าของข้อมูลที่ตำแหน่ง (x+1, y)

c = ค่าของข้อมูลที่ตำแหน่ง (x+1, -y-1)

d = ค่าของข้อมูลที่ตำแหน่ง (x, y-1)

e = ค่าของข้อมูลที่ตำแหน่ง (x-1, y-1)

f = ค่าของข้อมูลที่ตำแหน่ง (x-1, y)

g = ค่าของข้อมูลที่ตำแหน่ง $(x-1, y+1)$

h = ค่าของข้อมูลที่ตำแหน่ง $(x, y+1)$

i = ค่าของข้อมูลที่ตำแหน่ง $(x+1, y+1)$

ดังรูป

e	d	c
f	a	b
g	h	i

4.3.2 ส่วนตัดขอบภาพ (edge_track)

ขั้นแรกจะทำการสแกนภาพจากด้านบนลงสู่ด้านล่าง เพื่อหาจุดเริ่มต้นของขอบภาพ หลังจากนั้นจึงทำการติดตามทางเดินของเส้นขอบภาพ โดยจากจุดเริ่มต้น จะเริ่มการติดตามไปทางด้านซ้าย เพื่อให้เกิดการติดตามเส้นขอบภาพ ไปในทิศทางทวนเข็มนาฬิกา วิธีการที่ใช้ในการติดตาม จะใช้การมองจากตำแหน่งที่อยู่ในปีปัจจุบัน ของการติดตาม แล้วทำการเปรียบเทียบข้อมูลรอบๆตำแหน่งนั้น ยกเว้นด้านที่ผ่านการติดตามมาแล้ว โดยถือว่าตำแหน่งที่มีค่าของข้อมูลสูงสุด เป็นตำแหน่งที่จะทำการติดตามต่อไป เมื่อทำการติดตาม โดยวิธีการนี้ไปเรื่อยๆ จนกระทั่งถึงตำแหน่งที่เริ่มทำการติดตาม ก็จะได้เส้นขอบภาพที่เป็นเส้นเส้นเดียวรอบภาพตามต้องการ

4.3.3 ส่วนวิเคราะห์เส้นตรง (line_analyze)

การทำงานของส่วนนี้จะนำตำแหน่งที่ได้จากการตัดขอบภาพมาทำการวิเคราะห์คร่าวๆ 10 ตำแหน่ง มาทำการทดสอบว่าตำแหน่งต่างๆ นั้นอยู่บนเส้นตรงเดียวกันหรือไม่ โดยการคำนวณระยะห่างระหว่างตำแหน่งต่างๆ กับเส้นตรงที่ลากระหว่างตำแหน่งที่ 1

และ ทำหน้าที่ 10 ตามสมการ

$$d = \frac{|Ax_n + By_n + C|}{\sqrt{A^2 + B^2}}$$

(x_{10}, y_{10})

โดยจากรูปจะได้ว่า

$$m = \frac{y_{10} - y_1}{x_{10} - x_1}$$

$$A = -m$$

$$B = 1$$

$$C = -y_1 + (m * x_1)$$

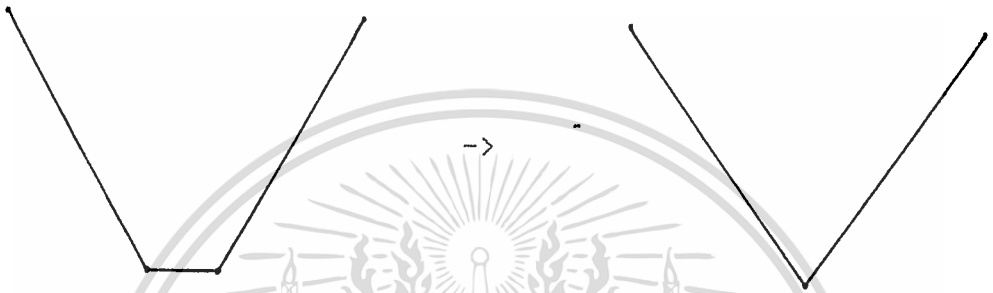
ในกรณีที่ไม่มีตำแหน่งหนึ่งตำแหน่งใดมีระยะห่างจากเส้นตรงนี้มากกว่าค่าที่กำหนด จะถือว่าตำแหน่งนั้นเป็นตำแหน่งเริ่มต้นของส่วนของเส้นตรงเส้นต่อไป แต่ถ้าทุกตำแหน่งอยู่บนเส้นตรงเดียวกัน ก็จะนำส่วนของเส้นตรงที่ได้นี้ ไปทำการเปรียบเทียบกับส่วนของเส้นตรงเส้นที่ผ่านมาว่าทำมุมต่อกันมากกว่าค่าที่กำหนดหรือไม่ ถ้ามากกว่าก็จะถือว่าเป็นเส้นตรงเส้นใหม่ต่อไป

เมื่อทำการทดสอบไปจนกระทั่งครบทุกตำแหน่งบนเส้นขอบภาพ ก็จะได้เส้นตรงและมุมที่เป็นส่วนประกอบของเส้นขอบภาพของวัตถุนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

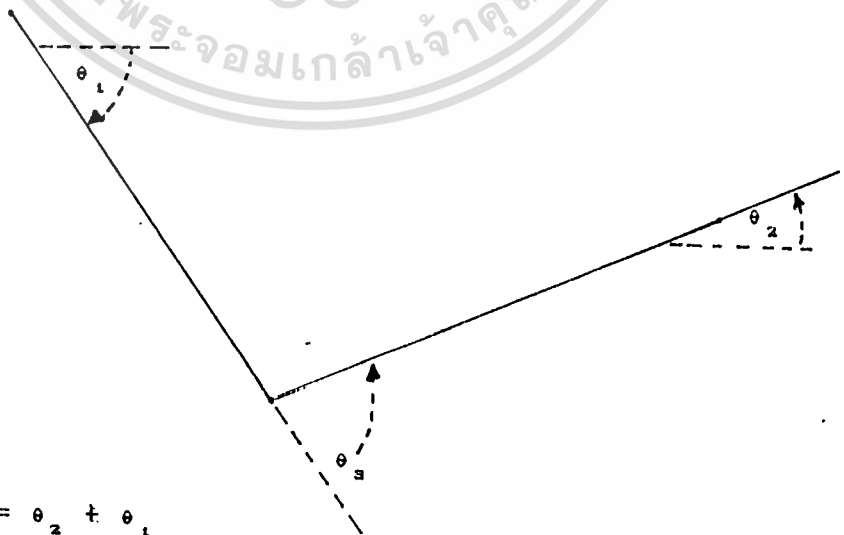
4.3.4 ส่วนกรองเส้นตรง (line_filter)

ส่วนนี้จะทำการแสดกขนาดของเส้นตรงที่เป็นส่วนประกอบของเส้นขอบภาพที่ได้ จากส่วนวิเคราะห์เส้นตรงว่ามีขนาดเล็กกว่าที่กำหนดหรือไม่ ถ้าเล็กกว่าก็จะทำการตัด ออกโดยการแบ่งเส้นตรงเส้นนั้นออกเป็นสองส่วนเท่าๆ กัน แล้วนำไปบวกเข้ากับ เส้นตรงที่อยู่กับเส้นตรงเส้นนี้ทั้งสองด้าน ดังรูป



4.3.5 ส่วนสร้างข้อมูล (code_gen)

จะนำความยาวและมุมของเส้นตรงที่ผ่านการกรองมาหาความสัมพันธ์ระหว่างเส้น โดยการนำมุมของเส้นที่ต่อเนื่องกันมาลบกัน ดังนั้นจะได้ข้อมูลที่เป็นความยาวของด้าน และมุมที่กระทำกับเส้นตรงเส้นถัดไปต่อเนื่องกันไปเรื่อยๆ จนครบทุกเส้น



$$\text{จะได้ว่า } \theta_3 = \theta_2 + \theta_1$$

4.3.6 ส่วนเก็บข้อมูล (build_database)

จะทำการเก็บข้อมูลของวัตถุที่ได้ไว้ในหน่วยความจำสำรอง โดยจะทำการเก็บชื่อของวัตถุ , มุมอ้างอิงของเส้นเริ่มต้น, จำนวนเส้นตรงที่เป็นส่วนประกอบ และตามด้วยข้อมูลของเส้นตรงแต่ละเส้นตามลำดับดังนี้

ลำดับที่	มุมอ้างอิง	ชื่อของวัตถุ	จำนวนเส้นตรง	{ ความยาว,มุม,ความยาว,มุม,...}
1	60	triangle	3	20, 60, 20, 60, 20, 60
2	90	square	4	50, 90, 50, 90, 50, 90,...
.
.
.

4.3.7 ส่วนเปรียบเทียบข้อมูล (look_object)

จะนำข้อมูลที่เก็บไว้ในหน่วยความจำสำรองมาไว้ในหน่วยความจำหลัก แล้วทำการเลือกว่าข้อมูลชุดใดบ้างที่มีจำนวนเส้นตรงเท่ากับข้อมูลของวัตถุที่ได้ แล้วจึงนำข้อมูลที่ได้ในแต่ละชุดมาทำการเปรียบเทียบเพื่อหาค่าความผิดพลาดของมุมและความยาวของแต่ละชุด

ในการหาค่าความผิดพลาดนั้น จะแยกเป็นค่าผิดพลาดของความยาวและค่าผิดพลาดของมุม โดยแต่ละส่วนจะเป็นการนำข้อมูลของวัตถุกับข้อมูลที่เก็บไว้ที่มีตำแหน่งตรงกันมาลบกันแล้วยกกำลังสอง แล้วนำค่าผิดพลาดของแต่ละตำแหน่งนี้มาบวกกันแล้วจึงถอดรากที่สอง ก็จะได้ค่าความผิดพลาดมาหนึ่งค่า ดังสมการ

$$error(n) = \sqrt{[r_2(n) - r_1(n)]^2 + [\theta_2(n) - \theta_1(n)]^2}$$

แต่จากการที่ข้อมูลที่ใช้ในการเปรียบเทียบนี้เป็นข้อมูลที่ไม่มีมิติ - ท้าย จึงต้องทำการหมุนข้อมูลที่ใช้ในการเปรียบเทียบ แล้วทำการเปรียบเทียบไปจนกระทั่ง

ครบทุกค่า แล้วจึงนำค่าผิดพลาดที่ได้มาหาค่าที่น้อยที่สุด ก็จะได้ค่าผิดพลาดของวัตถุกับข้อมูลที่เกิดขึ้นไว้

จากนั้นจึงทำการเปรียบเทียบหาค่าผิดพลาดที่น้อยที่สุด ก็จะถือว่าเป็นวัตถุชนิดเดียวกับข้อมูลนั้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การจำแนกวัตถุโดยวิธีฟูรีเยอร์เดสคริปเตอร์ (FOURIER DESCRIPTOR)

5.1 การแปลงฟูรีเยอร์

5.1.1 การแปลงฟูรีเยอร์ (Fourier transform)

ถ้าให้ $f(x)$ เป็นฟังก์ชันต่อเนื่องเมื่อ x เป็นเลขจำนวนจริง การแปลงฟูรีเยอร์ของ $f(x)$ กำหนดดังสมการ

$$F\{ f(x) \} = F(u) = \int_{-\alpha}^{\alpha} f(x) \exp[-j2\pi ux] dx \quad (5.1.1)$$

จากค่า $F(u)$ เราสามารถหา $f(x)$ ได้โดยการใช้ การแปลงฟูรีเยอร์ผกผัน ดังสมการ

$$\begin{aligned} F^{-1}\{ F(u) \} &= f(x) \\ &= \int_{-\alpha}^{\alpha} F(u) \exp[j2\pi ux] du \end{aligned} \quad (5.1.2)$$

สมการ (5.1.1) และ (5.1.2) เรียกว่า คู่การแปลงฟูรีเยอร์ (Fourier transform pair) เงื่อนไขเพียงพอสำหรับ $f(x)$ ที่จะมี $F(u)$ นั้น $f(x)$ จะต้องเป็นฟังก์ชันต่อเนื่อง และสามารถอินทิเกรตได้ (integrable) และ $F(u)$ ต้องสามารถอินทิเกรตได้ ซึ่งในทางปฏิบัติแล้ว เงื่อนไขเหล่านี้มักจะเป็นจริงเสมอ

ในที่นี้เรามักจะใช้ฟังก์ชัน $f(x)$ ที่เป็นค่าจริง ซึ่งการแปลงฟูรีเยอร์ของฟังก์ชันค่าจริง จะได้ค่าเป็นเชิงซ้อน ซึ่งแทนได้เป็น

$$F(u) = R(u) + jI(u)$$

เมื่อ $R(u)$ และ $I(u)$ เป็นส่วนจริงและส่วนจินตภาพ ของ $F(u)$ ตามลำดับ ซึ่งเราสามารถเขียนแทนในเทอมเอ็กโปเนนเชียลได้เป็น

$$F(u) = |F(u)| e^{j\phi(u)}$$

เมื่อ

$$|F(u)| = [R^2(u) + I^2(u)]^{1/2}$$

และ

$$\phi(u) = \tan^{-1} [I(u)/R(u)]$$

5.1.2 การแปลงฟูรีเยอร์เต็มหน่วย (Discrete Fourier transform)

ผลการแปลงฟูรีเยอร์ เป็นเครื่องมือทางคณิตศาสตร์ที่มีประโยชน์มาก โดยเฉพาะในการวิเคราะห์ ดิจิทัลระบบทางอิเล็กทรอนิกส์รวมทั้งการประมวลผลข้อมูลภาพการมีวิธีการคำนวณที่มีประสิทธิภาพ และเหมาะสมที่ใช้ในการคำนวณโดยใช้คอมพิวเตอร์จึงเป็นสิ่งจำเป็น แต่เนื่องจากในนิยามของการแปลงฟูรีเยอร์ที่กล่าวมา ค่าต่างๆ เป็นอินทิกรัลซึ่งเวลาในการคำนวณโดยใช้คอมพิวเตอร์ อาจต้องใช้เวลาาน หรืออาจมีปัญหาการลู่เข้าของคำตอบ หรืออาจหาคำตอบไม่ได้ ถ้าหากเราสามารถมีการแปลงที่เหมาะสม และข้อมูลหรือสัญญาณที่รับเข้ามามีความยาวจำกัด ใช้อุปกรณ์จำกัด ที่มีจำนวนพจน์มากพอ ค่าใกล้เคียงกับการแปลงเดิม และเวลาในการคำนวณเร็ว การแปลงใหม่นี้จะเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า

ไปว่ากรณิดองทั้งสี่ ลึกทั้งห้าขมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยชน์อย่างมาก การแปลงที่จะกล่าวถึงนี้มีชื่อเรียกว่า การแปลงฟูรีเยอร์เต็มหน่วย ซึ่งนิยามได้ดังนี้

ถ้าสมมติให้ ฟังก์ชันต่อเนื่องสามารถลุ่มออกมาเป็นลำดับได้เป็น $\{ f(x_0), f(x_0 + \Delta x), f(x_0 + 2\Delta x), \dots, f(x_0 + [N-1]\Delta x) \}$ โดยการลุ่มตัวอย่างเป็นจำนวน N ตัวอย่าง ด้วยช่วง Δx เพื่อความสะดวกเราจะเขียนแทนด้วยตัวแปร x ทั้งในตัวแปรแบบต่อเนื่อง และตัวแปรแบบเต็มหน่วย ซึ่งเราจะกำหนดได้ดังนี้

$$f(x) = f(x_0 + x\Delta x)$$

ซึ่งตัวแปร x ในที่นี้จะเป็นตัวแปรแบบเต็มหน่วย ซึ่งมีค่าตั้งแต่ 0 ถึง $N-1$ ซึ่งทำให้เราได้ลำดับเป็น $\{ f(0), f(1), f(2), \dots, f(N-1) \}$

จากข้อกำหนดเบื้องต้น เราสามารถกำหนดการแปลงฟูรีเยอร์เต็มหน่วย ของฟังก์ชันที่ถูกลุ่มตัวอย่างออกมาเป็น

$$F(u) = \sum_{x=0}^{N-1} f(x) \exp[-j2\pi ux/N] \quad (5.1.3)$$

เมื่อ $u = 0, 1, 2, \dots, N-1$

และกำหนดการแปลงผกผันฟูรีเยอร์เต็มหน่วย ได้ดังนี้

$$f(x) = \sum_{u=0}^{N-1} F(u) \exp[j2\pi ux/N] \quad (5.1.4)$$

เมื่อ $x = 0, 1, 2, \dots, N-1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 คุณสมบัติของการแปลง

5.2.1 การหมุน (rotation)

หากกำหนดตัวแปร ให้เป็นรูปเชิงขั้ว (polar coordinate) ดังนี้

$$x = r \cos \theta \qquad y = r \sin \theta$$

$$u = w \cos \phi \qquad v = w \sin \phi$$

เราจะได้ $f(x,y)$ และ $F(u,v)$ เป็น $f(r,\theta)$ และ $F(w,\phi)$ ซึ่งสามารถพิสูจน์ได้ว่า

$$f(r, \theta + \theta_0) = F(w, \phi + \phi_0)$$

และนอกจากนี้ ถ้า $f(x,y)$ ถูกทำให้หมุนไปด้วยมุม θ_0 เราจะได้ $F(u,v)$ ถูกหมุนไปด้วยมุมขนาดเดียวกัน ในทางกลับกัน ถ้าหมุน $F(u,v)$ ก็จะทำให้ $f(x,y)$ หมุนไปด้วยมุมขนาดเดียวกัน

5.2.2 การกระจายและอัตราส่วน (distributivity and scaling)

จากข้อกำหนดการแปลง พิสูจน์ได้ว่า

$$F\{ f_1(x,y) + f_2(x,y) \} = F\{ f_1(x,y) \} + F\{ f_2(x,y) \}$$

และโดยทั่วไป

$$F\{ f_1(x,y) \cdot f_2(x,y) \} = F\{ f_1(x,y) \} \cdot F\{ f_2(x,y) \}$$

สำหรับอัตราส่วน เมื่อ a และ b เป็นสเกลาร์ จะได้

$$af(x,y) = a F(u,v)$$

และ

$$f(ax,by) = (1/|ab|) F(u/a,v/b)$$

5.2.2. ประโยชน์ของการแปลงในการประมวลผลข้อมูลภาพ

ในการประมวลผลข้อมูลภาพ (image processing) เราได้นำการแปลงเข้ามาใช้ประโยชน์อย่างกว้างขวางในข้อมูลภาพ การเก็บข้อมูล มักจะมีการเก็บในรูปของอะเรย์ขนาด 2 มิติ (two-dimensional array) การแปลงที่นำมาใช้กับการประมวลผลข้อมูลภาพจึงมักจะเป็นการแปลง 2 มิติ (two-dimensional transform)

ประโยชน์ของการแปลงในงานการประมวลผลข้อมูลภาพ

1. การลดขนาดข้อมูล (image compression) โดยการใช้การแปลงกับข้อมูลภาพ แล้วเก็บค่าสัมประสิทธิ์ที่สำคัญบางส่วนไว้ ส่วนที่เหลือสามารถที่จะตัดทิ้งได้ ทำให้ขนาดของข้อมูลที่จะทำการเก็บมีขนาดเล็กลง สิ้นเปลืองเนื้อที่ในหน่วยความจำน้อยลง และสามารถส่งข้อมูลไปโดยใช้เวลาอันน้อยลง เมื่อต้องการนำข้อมูลภาพกลับมาใช้งานใหม่ก็สามารถทำได้โดยใช้ การแปลงผกผัน จะได้ภาพที่มีข้อมูลผิดเพี้ยนไปบ้างแต่เพียงเล็กน้อยเท่านั้น

2. ใช้ในการคำนวณผลการประสาน (convolution) ซึ่งใช้ในการกรอง (filter) การนำการแปลง เช่น การแปลงฟาสต์ฟูรีเยอร์มาใช้งาน ทำให้สามารถคำนวณผลการประสานได้รวดเร็วยิ่งขึ้น

3. ใช้ในการจำแนกวัตถุ (description) โดยวิธีการที่เรียกกันว่า ฟูรีเยอร์เดสคริปเตอร์ (fourier descriptor) ซึ่งจะใช้วิธีนี้ในโครงการ robot vision) ซึ่งจะนำมาอธิบายในตอนถัดไป

5.3 การจำแนกวัตถุ

5.3.1 การจำแนกวัตถุโดยวิธีการฟูรีเยอร์ เกลสคริปเตอร์

การแปลงฟูรีเยอร์เต็มหน่วย สามารถที่จะนำมาใช้เป็นเครื่องมือในการจำแนกรูปทรงของวัตถุได้อย่างมีประสิทธิภาพ โดยไม่มีข้อจำกัดในเรื่องของขนาดของวัตถุ , ตำแหน่งการวางตัว และลักษณะทิศทางในการวางตัว (orientation)

เราจะใช้การแปลงฟูรีเยอร์เต็มหน่วย กับข้อมูลภาพที่ผ่านกระบวนการหาขอบภาพมาแล้ว โดยที่ข้อมูลของขอบภาพจะวางตัวอยู่ในระนาบ 2 มิติ x และ y สำหรับวิธีการที่เราจะใช้ที่นี่ เราจะนำเอาการแปลงฟูรีเยอร์เต็มหน่วยแบบมิติเดียวมาใช้งาน เราจึงเปลี่ยนจากระนาบ 2 มิติ x และ y มาเป็นระนาบเชิงซ้อน โดยให้แกน x เป็นแกนจริง และแกน y เป็นแกนจินตภาพ ทำให้เราได้ข้อมูลของขอบภาพแต่ละจุด เป็นจุดในระนาบเชิงซ้อน ซึ่งแทนได้ด้วย $x+iy$

ถ้าหากว่าเราใช้วิธีการติดตาม (tracking) ไปตามแนวขอบภาพจนครบรอบ เราก็จะได้เลขจำนวนเชิงซ้อนขึ้นมา 1 ชุด ซึ่งจะเป็นลำดับ 1 ลำดับ จากนั้นจึงนำเอาลำดับที่ได้มานี้เข้ามาผ่านกระบวนการแปลงฟูรีเยอร์เต็มหน่วย เราก็จะได้ลำดับที่ผ่านการแปลงแล้ว 1 ชุด ซึ่งลำดับนี้เราจะเรียกว่า ฟูรีเยอร์ เกลสคริปเตอร์ หรือ FD ของเส้นขอบภาพ

จากความสามารถในการผันกลับได้ของการแปลงฟูรีเยอร์เต็มหน่วย ข้อมูลของภาพหลังจากผ่านการแปลง และการแปลงผกผันจะยังอยู่ครบถ้วน เราสามารถนำลำดับ FD มาเป็นเครื่องมือในการจำแนกวัตถุได้ โดยนำมาผ่านกระบวนการบางอย่าง เพื่อให้อยู่ในรูปแบบมาตรฐาน (normalized fourier descriptor) หรือ nFD มีค่าไม่ขึ้นกับขนาด , ตำแหน่งและทิศทางในการวางตัว

5.3.2 การหา nFD

จากคุณสมบัติของการแปลงฟูรีเยอร์เต็มหน่วย ตามที่ได้กล่าวมาแล้วข้างต้น การเปลี่ยนแปลงขนาดของวัตถุ ทำได้โดยการคูณค่าคงที่เข้าไปกับสัมประสิทธิ์การแปลง ซึ่งจากคุณสมบัติการเป็นเชิงเส้น เมื่อเราทำการแปลงผกผัน ก็เท่ากับว่า เราคูณค่าคงที่เข้าไปในค่าขอบภาพในระนาบ 2 มิติ

การหมุนภาพเราสามารถทำได้โดย คูณแต่ละจุดด้วย $\exp(j\theta)$ เมื่อ θ คือมุมในการหมุนจากคุณสมบัติความเป็นเชิงเส้น ก็จะมีผลเช่นเดียวกันในโดเมนความถี่

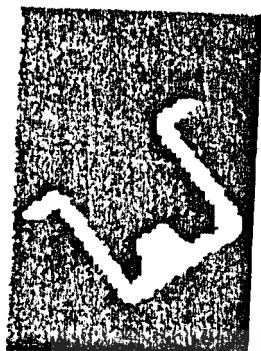
สำหรับผลกระทบของการเปลี่ยนจุดเริ่มต้นของการติดตามเส้น เราสามารถดูได้จากคุณสมบัติการเป็นคาบ (periodicity property) ของการแปลงฟูรีเยอร์เติมหน่วย ลำดับของตัวเลขที่มีจำนวนจำกัดใน spatial domain เราถือว่าเป็นการแทน 1 ไซเคิลของฟังก์ชันเป็นคาบ ค่าสัมประสิทธิ์การแปลงฟูรีเยอร์เติมหน่วยในที่นี้ก็คือสัมประสิทธิ์ของอนุกรมฟูรีเยอร์ของฟังก์ชันเป็นคาบ จากลักษณะนี้เมื่อเราเลื่อนจุดเริ่มต้นของเส้นรอบรูปใน spatial domain เราจะพบว่า ในโดเมนความถี่ คือการคูณสัมประสิทธิ์ด้วย k ด้วย $\exp(jkT)$ เมื่อ T เป็นช่วงซึ่งจุดเริ่มต้นถูกเลื่อนไป (เมื่อ T เป็นการเปลี่ยนจาก 0 ไปถึง 2π จุดเริ่มต้นจะถูกเลื่อนไปครบ 1 รอบ)

จากค่า FD ที่เราได้ ขั้นตอนการทำ nFD ในที่นี้เราทำการ normalize เพียงแค่ขนาดก็เพียงพอ เพราะเราจะใช้เฉพาะค่าขนาด (magnitude) ของ FD เท่านั้น ไม่ใช่มุม เราจึงสามารถหา nFD ได้โดยการเลือกค่าขนาดที่มากที่สุดให้มีขนาดเป็น 1 แล้วปรับตัวอื่นตาม ก็จะได้ nFD ตามต้องการ

5.3.3 การเปรียบเทียบข้อมูลโดยใช้ nFD

การเปรียบเทียบข้อมูลที่ต้องการ กับข้อมูลในหน่วยความจำโดยใช้ nFD เราจะใช้เฉพาะข้อมูลในลำดับแรกของ nFD ก็เพียงพอ เพราะเพียงแค่เฉพาะข้อมูลในลำดับแรกๆ ก็สามารถแทนข้อมูลสำคัญของวัตถุไว้ได้ ดังเช่นภาพที่ 5.3.1 เราสามารถแยกความแตกต่างได้โดยใช้ข้อมูลเพียงแค่ 10 เปรอร์เซ็นต์ของทั้งหมด

ในโครงการนี้เราจะใช้ nFD จำนวน 32 ข้อมูล นำมาเปรียบเทียบกับข้อมูลต้นแบบที่เก็บไว้ในหน่วยความจำ แล้วหาผลรวมของความแตกต่างทั้งหมด โดยวัตถุที่มีลักษณะใกล้เคียงกับวัตถุที่เป็นต้นแบบ จะมีค่าความแตกต่างน้อยที่สุด



รูปที่ 5.3.1 วัตถุ 2 แบบที่สามารถแยกแยะได้โดยใช้ FD

5.4 โปรแกรมการจำแนกวัตถุ

5.4.1 การประมวลผลเบื้องต้น

ข้อมูลภาพที่เข้ามา จะถูกทำให้กลายเป็นภาพที่มีระดับความเข้ม 2 ระดับ โดยการใช้ค่าระดับความเข้มที่จุดแตกต่างระหว่างตัววัตถุกับพื้น เป็นค่าหลักที่ใช้แบ่งระดับความเข้ม เรียกว่า threshold value

จากนั้นจะทำการหาค่าขอบภาพโดยวิธีติดตามขอบภาพ (tracking) ไปตามเข็มนาฬิกาจนครบรอบ จะได้เป็นลำดับ (x, y) ซึ่งเราจะนำมาเปลี่ยนให้เป็นค่าเลขเชิงซ้อน $(x + jy)$ ก่อนที่จะนำไปผ่านการแปลงฟูรีเยอร์เต็มหน่วย

5.4.2 การแปลงข้อมูลขอบภาพ

ข้อมูลขอบภาพที่ได้ จะถูกนำมาแปลงฟูรีเยอร์เต็มหน่วย ซึ่งในส่วนนี้จะใช้เวลา นาน ขึ้นกับความยาวของลำดับที่นำมาแปลง ถ้าจำนวนของลำดับขยาวมากก็ยิ่งจะใช้เวลา นานมากขึ้น ข้อมูลที่ผ่านโปรแกรมส่วนนี้แล้วจะได้เป็น FD

จาก FD ที่ได้เราจะนำมาทำการ normalize โดยทำการหาค่าขนาด (magnitude) ของ FD เสียก่อน แล้วหาค่าขนาดที่มากที่สุด (maximum magnitude) เป็นขนาดมาตรฐาน คือ นำมาหาร FD ทั้งหมด จะได้เป็น nFD

5.4.3 การเปรียบเทียบโดยใช้ nFD

ข้อมูล nFD ที่ได้จะถูกตัด (truncate) ให้เหลือเพียง 32 ตัว เพื่อจะนำมาเปรียบเทียบกับข้อมูลต้นแบบในหน่วยความจำ เพื่อหาค่าความแตกต่าง โดยใช้วิธีลบกัน แล้วนำมายกกำลังสองแล้วนำมาลบกัน จากนั้นจึงนำค่าความแตกต่างกำลังสองทั้งหมดมารวมกัน ได้เป็นผลรวมความแตกต่างกำลังสอง

ผลรวมความแตกต่างของวัตถุ ที่มีคล้ายคลึงกันจะมีค่าน้อยมาก ทำให้เราสามารถจำแนกวัตถุออกจากกันได้

5.4.4 ข้อดีและข้อเสียของการใช้การจำแนกวัตถุโดยใช้ nFD

การใช้ nFD ในการจำแนกวัตถุ มีข้อดีในการที่สามารถจำแนกวัตถุที่มีสัญญาณรบกวนมาเกี่ยวข้อง สามารถจำแนกวัตถุที่มีข้อมูลขอบผิดเพี้ยนไป ดังจะนำมาแสดงให้เห็นในการทดลอง

ข้อเสียของการใช้ nFD ในการจำแนกวัตถุ คือในขั้นตอนการแปลงฟูรีเยอร์เต็มหน่วยต้องใช้เวลาในการกระทำการพอสมควร

บทที่ 6

การทดลองและผลการทดลอง

6.1 การทดลองเก็บภาพด้วยแผ่นวงจรติดต่อข้อมูลภาพ

6.1.1 การปรับตั้ง

ในส่วนนัฟเฟอ์ของการ์ด มีตัวเลือกเพื่อจะใช้ตัวกรองสีสำหรับสัญญาณภาพสี แต่จากการทดลองด้วยกล่องวิดีโอที่ให้สัญญาณภาพสี ถ้าใช้ตัวกรองจะทำให้ภาพขาดความคมชัดลงไป และเมื่อไม่ใช้ก็ไม่ได้ให้ผลเสียใดๆปรากฏขึ้นมา ดังนั้นเราจึงตกลงที่จะไม่ใช้ตัวกรองสีในการใช้งาน

การปรับค่าระดับแรงดันอ้างอิงของตัวแปลงสัญญาณเชิงอุปมานเป็นเชิงเลข เพื่อให้ได้รายละเอียดของภาพสูงสุด จะต้องระวังไม่ให้ระดับสีดำ (V_{black}) สูงเกินไป เพราะจะทำให้บริเวณที่สีดำ มีรายละเอียดน้อย และภาพโดยรวมค่อนข้างดำ ในทางตรงกันข้าม ถ้าระดับสีขาว (V_{white}) น้อยเกินไป รายละเอียดบริเวณสีขาวก็จะขาด และภาพโดยรวมจะขาวเกินไป ในขณะที่เดียวกัน ก็ไม่ควรให้ระดับสีดำต่ำเกินไป หรือระดับสีขาวสูงเกินไป เพราะมันจะไปลดจำนวนระดับข้อมูลเชิงเลข (digital levels) ลง เช่น ถ้าค่าแรงดันสูงสุดของกล่อง คือ 1.4 โวลต์ การตั้งระดับขาวไว้ที่ 2 โวลต์ จะไม่เป็นการดี เพราะจะมีค่าแรงดันถึง 0.6 โวลต์ ซึ่งมีค่าเชิงเลขอยู่แต่ไม่มีทางได้ใช้งาน

จากการทดลองวัดระดับสัญญาณเชิงอุปมานที่เข้าตัวแปลงสัญญาณ จะมีค่าระดับต่ำสุดของสัญญาณเชิงคี่ที่ประมาณ 0 โวลต์ ระดับที่ให้ภาพดำที่ประมาณ 0.6 โวลต์ และระดับสูงสุดของสัญญาณภาพที่ประมาณ 2 โวลต์ ดังนั้นจึงทำการปรับตั้งโดยให้ระดับสีดำ (V_{black}) ที่ขา 10 ของเอาต์พุตมีค่า 0.75 โวลต์ และระดับสีขาว (V_{white}) ที่ขา 9 มีค่า 1.9 โวลต์ ผลปรากฏว่าได้ภาพที่มีรายละเอียดค่อนข้างสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.2 ผลจากการเก็บภาพ

เมื่อปรับตั้งค่าต่างๆของการ์ดเสริ์จเรียบร้อยแล้ว ใช้โปรแกรมดั่งแสดงในภาคผนวก*ก. ทำการเก็บภาพจากกล้องวิดีโอ ได้ผลดั่งแสดงในรูปที่ 6.1.1 และ 6.1.2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.1.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 การทดลองการแยกส่วนประกอบของ ขอบภาพ

6.2.1 การหาขอบภาพโดยวิธีของ โซเบล

ภาพที่ได้จากกล้องวิดีโอ เมื่อผ่านการแปลงโดยวงจรมัลติเพล็กซ์ข้อมูลภาพแล้ว จะได้ภาพที่มีความเข้ม 64 ระดับ ดังรูปที่ 6.2.1 และเมื่อผ่านกระบวนการหาขอบภาพแล้วจะได้ข้อมูลที่มีความเข้มที่บริเวณขอบภาพมากกว่าบริเวณอื่นอย่างชัดเจน ดังรูป 6.2.2

6.2.2 การตัดขอบภาพ

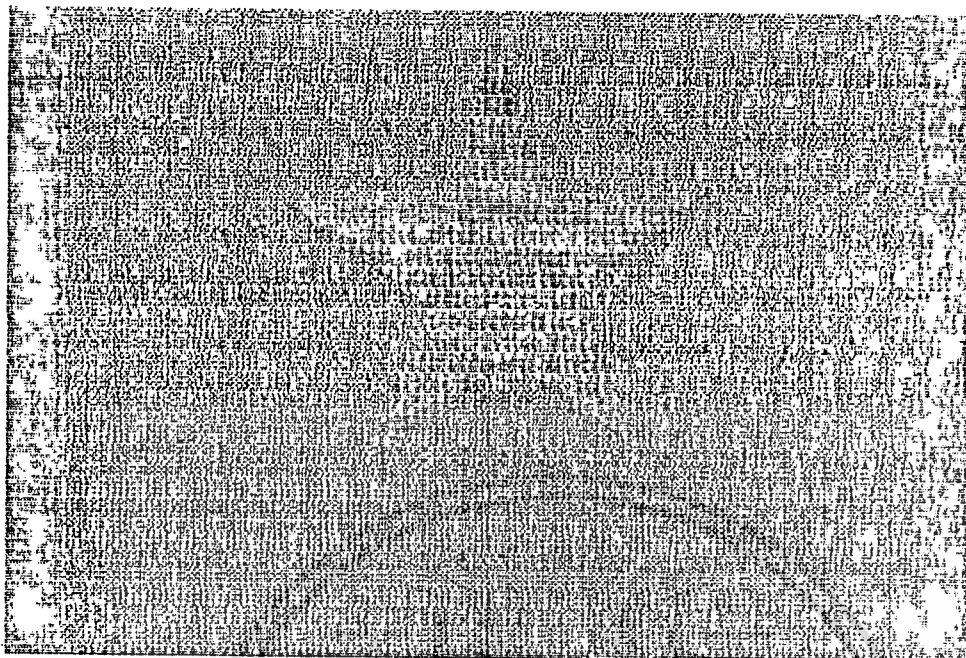
ข้อมูลที่ได้จากการหาขอบภาพ จะเป็นข้อมูลที่มีระดับความเข้มจึงมีลักษณะเป็นขอบที่มีความหนา เมื่อผ่านการตัดขอบภาพแล้วจะได้ข้อมูลที่เป็นเส้นขอบภาพดังรูป 6.2.3

6.2.3 การสร้างข้อมูลของเส้นขอบภาพ

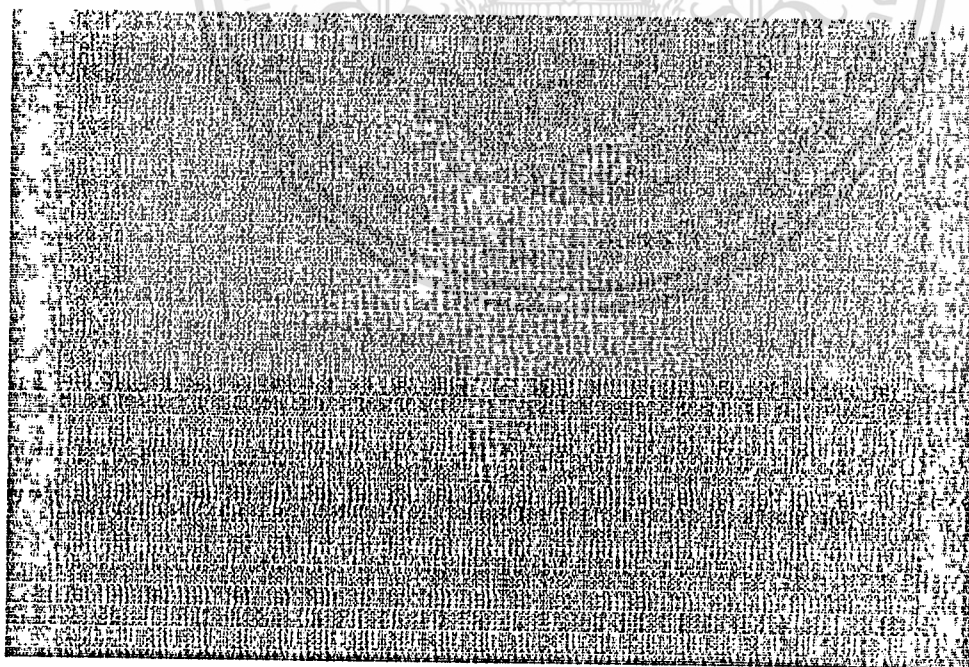
เส้นขอบภาพที่ได้จะผ่านส่วนที่ทำการวิเคราะห์เส้นตรง แล้วจึงสร้างเป็นข้อมูลของเส้นขอบภาพดังตาราง 6.2.1

6.2.4 การเปรียบเทียบข้อมูล

จากข้อมูลของเส้นขอบภาพ จะสามารถหาค่าผิดพลาดได้ดังตาราง 6.2.2



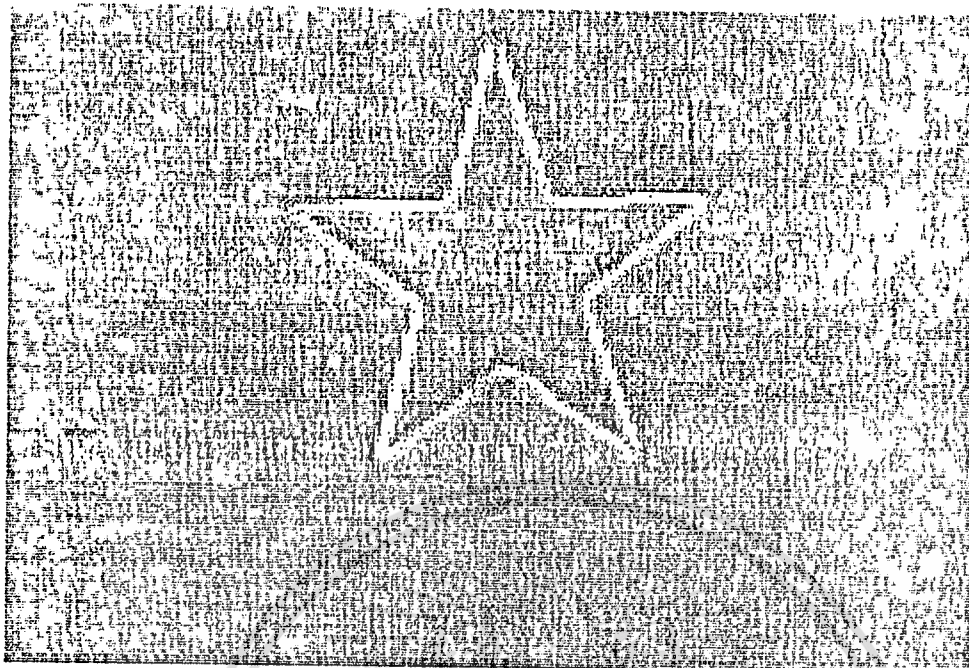
(ก) รูป star 1



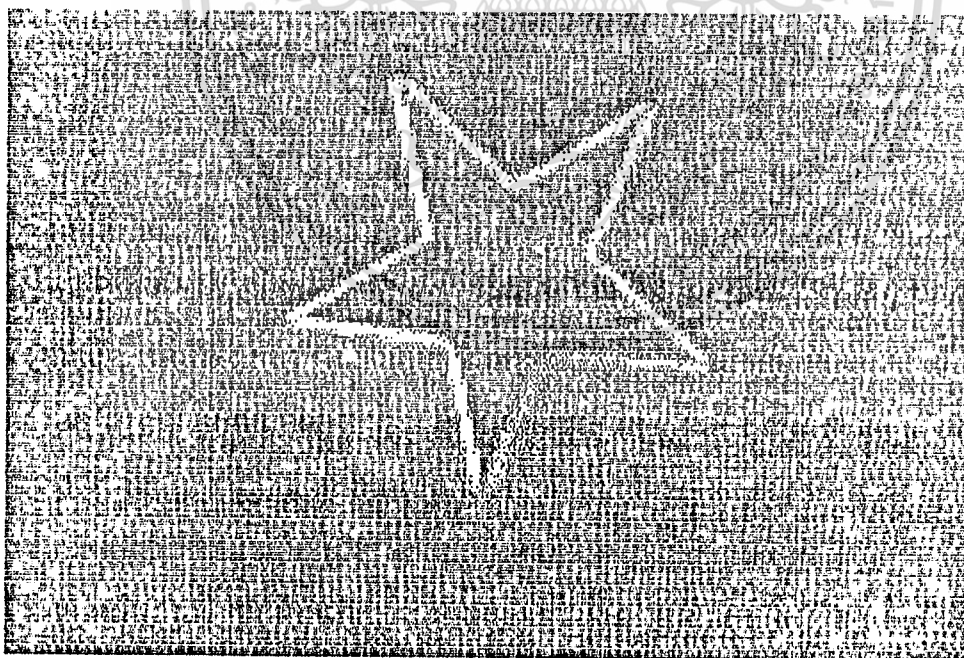
(ข) รูป star 2

รูปที่ 6.2.1 แสดงข้อมูลภาพที่ได้จากกล้องวิดีโอ

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของสำนักงานส่งเสริมการค้าในต่างประเทศ ณ นครโฮจิมินห์ ประเทศเวียดนาม ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



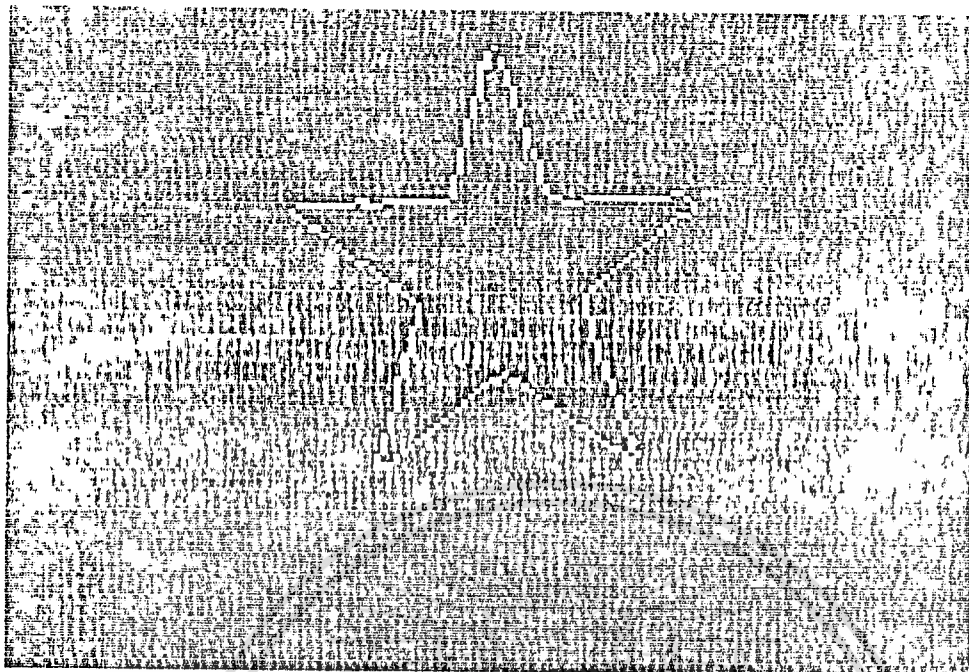
(ก) รูป star 1



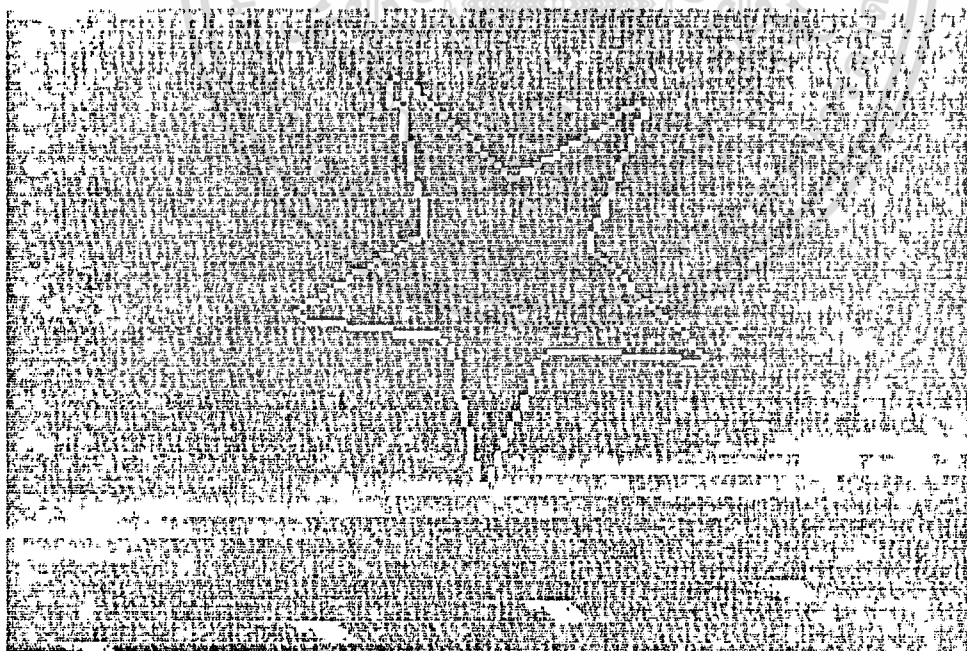
(ข) รูป star 2

รูปที่ 6.2.2 แสดงข้อมูลภาพที่ได้จากการหาขอบภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) รูป star 1



(ข) รูป star 2

เอกสารนี้เป็นเอกสารที่ 6.3.3 แสดงข้อมูลภาพที่ได้จากการตัดขอบภาพให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความยาว	มม
18.62	-130.23
21.85	66.89
22.14	-140.61
18.18	65.89
20.15	-133.86
21.98	56.53
19.96	-132.63
18.99	71.83
20.85	-139.57
18.87	55.80

ตารางที่ 6.2.1 (ก) แสดงข้อมูลส่วนประกอบของภาพ Star_1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความยาว	มุม
16.84	-135.31
20.37	72.57
20.58	-136.42
17.22	67.50
17.84	-148.06
20.92	67.97
19.54	-128.42
16.57	59.13
19.36	-140.44
19.27	61.48

ตารางที่ 6.2.1 (ข) แสดงข้อมูลส่วนประกอบของภาพ star_1

ครั้งที่	ค่าผิดพลาด
1	25.44
2	636.63
3	26.50
4	636.59
5	27.74
6	636.27
7	25.74
8	636.74
9	28.27
10	636.25

ตารางที่ 6.2.2 แสดงค่าผิดพลาดจากการเปรียบเทียบข้อมูล Star_1 กับ Star_2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 การทดลองการแปลงฟูรีเยอร์

6.3.1 การทำภาพที่มีความเข้ม 2 ระดับ

ภาพที่ได้จากกล้องวิดีโอ เมื่อทำการแปลงเป็นข้อมูลเชิงเลขแล้ว จะเป็นภาพที่มีระดับความเข้มถึง 64 ระดับ ซึ่งในการแยกความแตกต่างระหว่างวัตถุกับฉาก จะทำให้เป็นภาพที่มีความเข้ม 2 ระดับก่อน

รูปที่ 6.3.1 แสดงถึงภาพที่ได้จากกล้องถ่ายภาพวิดีโอระดับความเข้มของภาพแสดงได้ดังรูปที่ 6.3.2 โดยนำมาแสดงเป็นกราฟแท่ง จะเห็นจุดแตกต่างอยู่ตรงกลาง ซึ่งเราจะใช้เป็นจุดแบ่งระดับความเข้ม

รูปที่ 6.3.3 เป็นภาพหลังจากที่ได้ทำให้เป็นภาพที่มีความเข้ม 2 ระดับแล้วจะเห็นความแตกต่างระหว่างวัตถุกับฉากอย่างชัดเจน

6.3.2 การหาขอบภาพ

การหาขอบภาพจะเริ่มจากการหาดำแหน่งเริ่มต้นของภาพ แล้วทำการติดตามขอบภาพตามทิศทางตามเข็มนาฬิกา ซึ่งจะได้ขอบภาพเป็นดังรูปที่ 6.3.4

6.3.3 การหา nFD

จากขอบภาพที่ได้เราจะนำมาทำเป็นค่า complex แล้วผ่านการแปลงฟูรีเยอร์เต็มหน่วย ซึ่งเปรียบเทียบค่าข้อมูลที่เข้าไป และข้อมูลที่แปลงแล้วเฉพาะข้อมูลลำดับต้นๆ ได้ดังตาราง 6.3.1

6.3.4 การเปรียบเทียบกับค่าต้นแบบ

สำหรับค่าต้นแบบเราจะใช้ชุดของ nFD ต้นแบบ ตามจำนวนที่เรากำหนด ซึ่งได้ผ่านการแปลงและทำให้เป็นมาตรฐาน แล้วตัดทิ้งจนเหลือขนาด 32 ข้อมูลต่อต้นแบบ 1 ตัว จากนั้นจึงนำชุดของตัวต้นแบบทั้งหมดที่เลือกไว้ มาหาค่าผลรวมความแตกต่างกำลังสองจากวัตถุที่เราสนใจ

รูปที่ 6.3.5 แสดงภาพของตัวต้นแบบที่นำมาทำ nFD เก็บไว้ในหน่วยความจำแล้ว

ตาราง 6.3.2 แสดงค่า nFD ของตัวต้นแบบ และตัววัตถุที่สนใจ

เมื่อ prot.1	แทนข้อมูลของวัตถุต้นแบบตัวที่ 1
เมื่อ prot.2	แทนข้อมูลของวัตถุต้นแบบตัวที่ 2
เมื่อ prot.3	แทนข้อมูลของวัตถุต้นแบบตัวที่ 3
เมื่อ prot.4	แทนข้อมูลของวัตถุต้นแบบตัวที่ 4
เมื่อ prot.5	แทนข้อมูลของวัตถุต้นแบบตัวที่ 5
เมื่อ int.Obj	แทนข้อมูลของวัตถุที่สนใจ

ตาราง 6.3.3 แสดงค่าความแตกต่างของวัตถุที่สนใจกับตัวต้นแบบ

จะเห็นว่าวัตถุที่ใกล้เคียงกับตัวต้นแบบจะมีค่าผลรวมความแตกต่างกำลังสองน้อยที่สุด

สำหรับข้อดีของวิธีการใช้วิธีนี้ คือการใช้จำแนกวัตถุที่มีข้อมูลที่สัญลักษณ์รบกวน ดังเช่น ข้อมูลในรูปที่ 6.3.6 รูป (ก) เป็นข้อมูลเข้ารูปวงกลม ส่วนรูปที่ 6.3.6 (ข) เป็นรูป หลังจากผ่านการทำให้เป็นข้อมูลที่มีระดับความเข้ม 2 ระดับและหาขอบภาพแล้ว พบว่า ผลของสัญลักษณ์รบกวนจะทำให้ขอบภาพผิดเพี้ยนไปพอสมควร

เมื่อทำการหา nFD ของขอบภาพนี้แล้ว นำมาหาค่าผลรวมความแตกต่างกำลังสอง จาก nFD ของวงกลมต้นแบบ จะได้ค่าผลรวมความแตกต่างกำลังสองเพียงแค่ 0.180324 ทำให้สามารถบอกได้ว่าเป็นภาพของวัตถุทรงกลม

ตาราง 6.3.1 แสดงค่าเปรียบเทียบระหว่าง ข้อมูลที่ได้จากรูป (input) , ข้อมูลที่ผ่านการแปลงฟูรีเยอร์ และ ข้อมูลมาตรฐาน (nFD)

n	input	DFT data	nFD
0	.0+j00	5958.000000-j13049.000000	1.000000
1	1-j01	-2356.613525+j02598.886719	0.244566
2	1-j02	-865.768433+j00246.900604	0.062760
3	1-j03	-412.443787+j00102.199486	0.029622
4	2-j04	211.343185+j00702.419189	0.051135
5	2-j05	429.909668-j00303.186249	0.040341
6	2-j06	-95.311729+j00029.692289	0.006959
7	3-j07	35.544437+j00227.734329	0.016068
8	3-j8	163.540024+j00147.080460	0.015333
9	3-j09	49.498413+j00080.858147	0.006609

ตาราง 6.3.1 (ต่อ)

n	input	DFT data	nFD
10	3-j10	2.637232+j00029.849260	0.002089
11	3-j11	8.386927-j00015.411332	0.001223
12	3-j12	-78.141525-j00017.536829	0.005583
13	3-j13	-15.172418+j00034.348404	0.002618
14	3-j14	-32.260864+j00005.859158	0.002286
15	2-j15	38.792236+j00000.252191	0.002704
16	2-j16	-34.467216-j00005.008080	0.002428
17	3-j17	1.642878+j00034.518009	0.002409
18	4-j17	-7.737175+j00023.917517	0.001752
19	5-j18	8.703117-j00026.341364	0.001934

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 6.3.2 แสดงการเปรียบเทียบค่า nFD ระหว่างตัวต้นแบบทั้ง 5 และ nFD ของวัตถุที่เราสนใจ

n	prot.1	prot.2	prot.3	prot.4
0	1.000000	1.000000	1.000000	1.000000
1	0.252754	0.087311	0.080357	0.082245
2	0.081913	0.020301	0.154230	0.004560
3	0.037746	0.028254	0.003816	0.082686
4	0.058804	0.155902	0.009351	0.002895
5	0.037234	0.028824	0.204581	0.007895
6	0.012125	0.001166	0.002857	0.000984
7	0.020932	0.011208	0.004664	0.015158
8	0.011201	0.015012	0.009691	0.001361
9	0.012241	0.002408	0.002017	0.005695

ตาราง 6.3.2 (ต่อ)

n	prot.5	int.Obj.
0	1.000000	1.000000
1	0.070420	0.242185
2	0.003048	0.065595
3	0.012629	0.031897
4	0.001946	0.052638
5	0.004339	0.034267
6	0.000541	0.010310
7	0.001605	0.018912
8	0.001106	0.012299
9	0.001226	0.008117

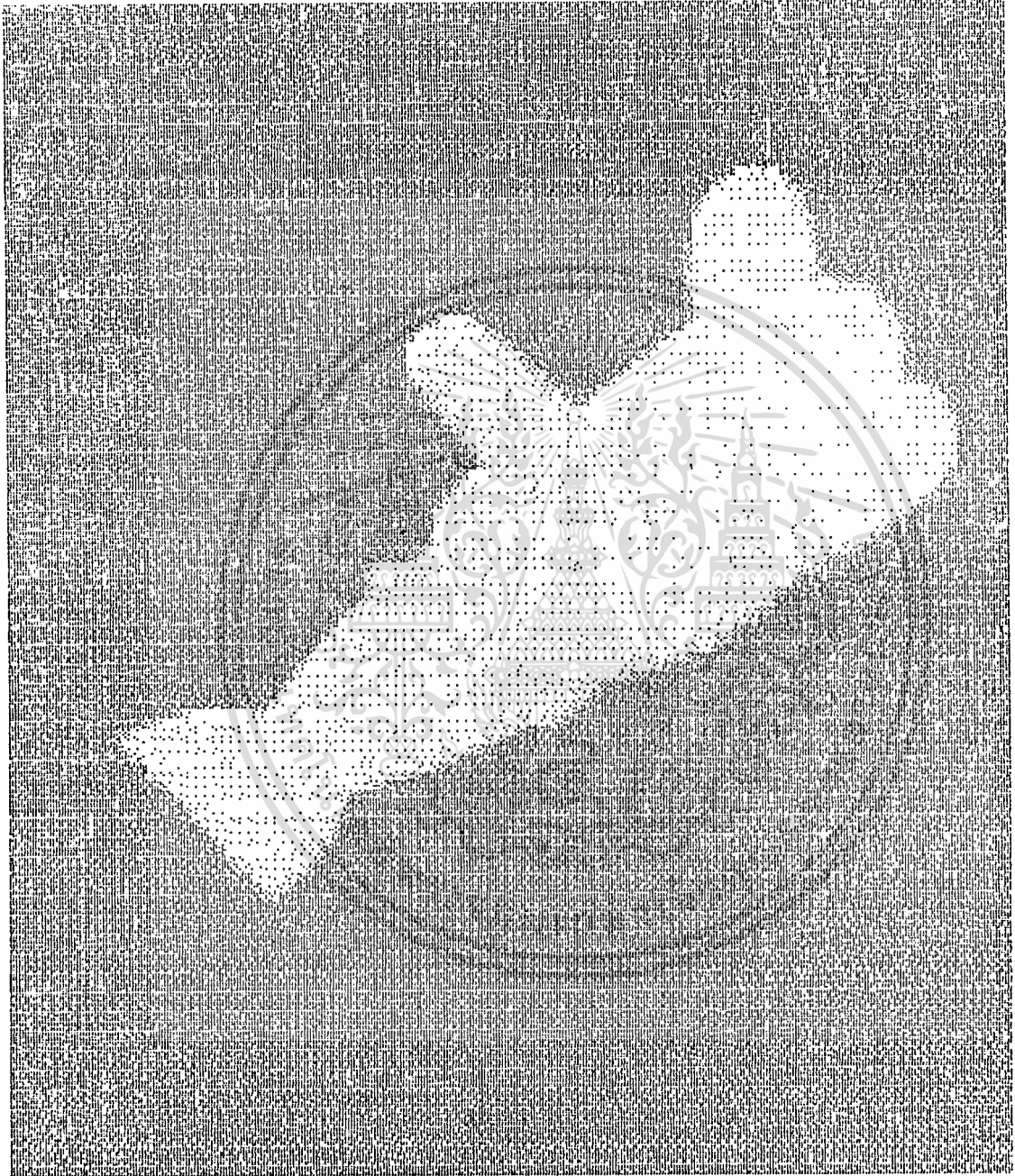
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 6.3.3 แสดงค่าผลรวมความแตกต่างระหว่างวัตถุที่สนใจ กับวัตถุต้นแบบ

object	prot.1	prot.2	prot.3
error	0.055259	3.702423	3.186807

object	prot.4	prot.5
error	3.514805	3.808082

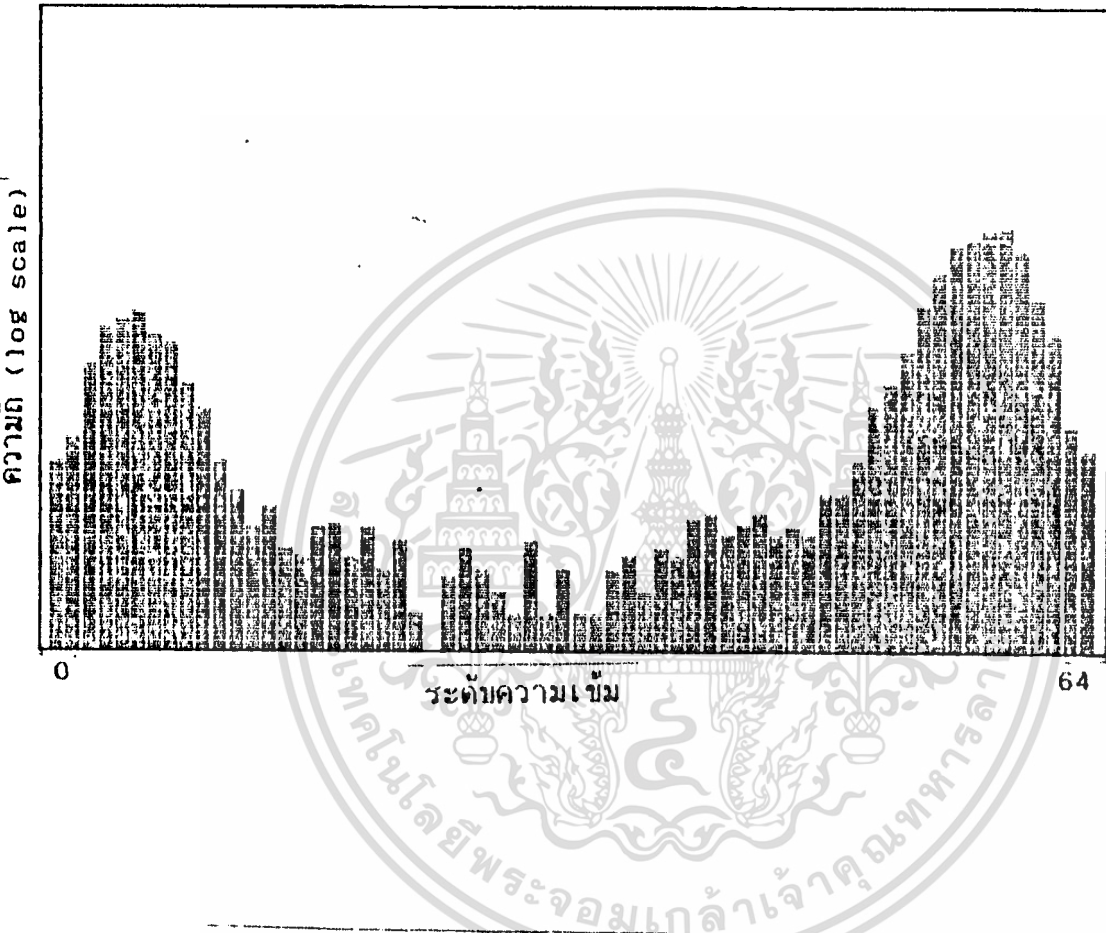
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3.1 แสดงภาพวัตถุที่สนใจจากกล้องถ่ายภาพวีดีโอ

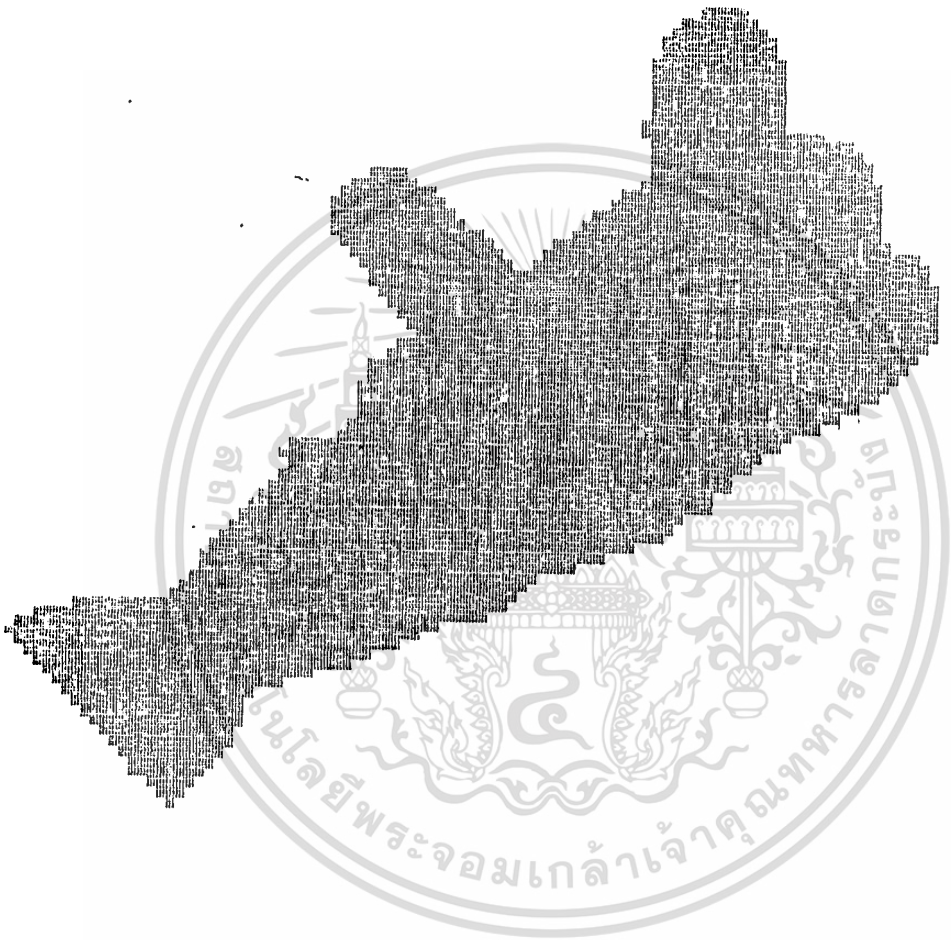
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HISTOGRAM



รูปที่ 6.3.2 กราฟแสดงระดับความเข้มของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3.3 ภาพวัตถุหลังจากทำให้เป็นภาพที่มีความเข้ม 2 ระดับแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3.4 ขอบภาพที่หาได้จากกระบวนการติดตามขอบภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.3.5 แสดงภาพของตัวต้นแบบที่นำมาทำ nFD เก็บไว้ในหน่วยความจำ

- (ก) ภาพต้นแบบชื่อ flipper 1
- (ข) ภาพต้นแบบชื่อ star 1
- (ค) ภาพต้นแบบชื่อ tri 2
- (ง) ภาพต้นแบบชื่อ square 2
- (จ) ภาพต้นแบบชื่อ circle
- (ฉ) ภาพวัตถุที่สนใจชื่อ flipper 2

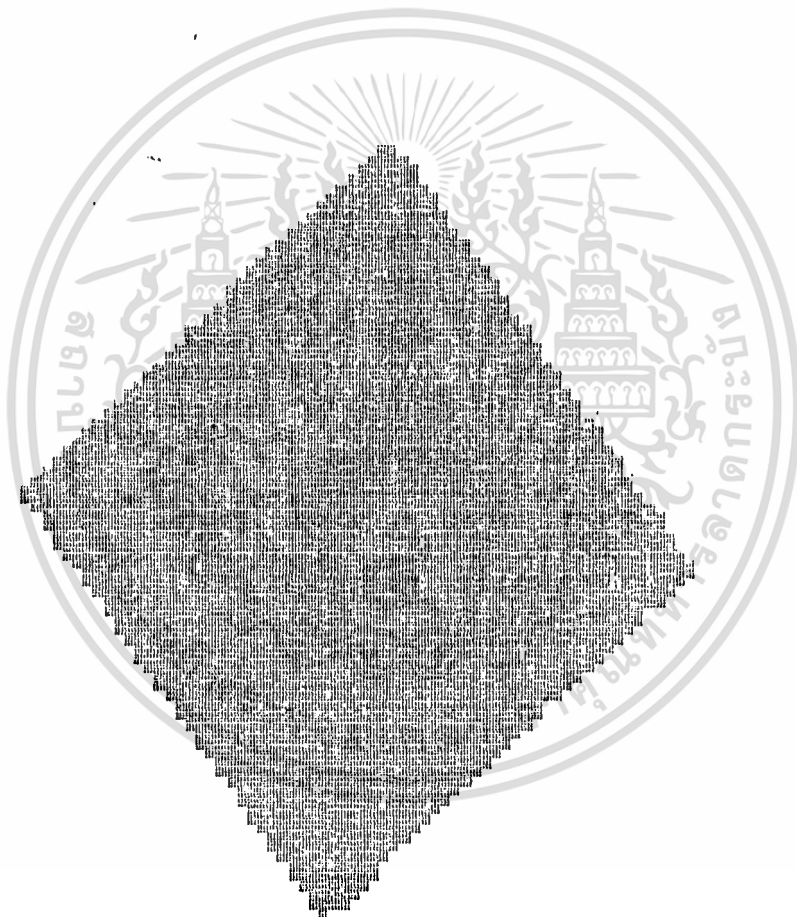


(ก) ภาพต้นแบบชื่อ flipper 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอก (ค) ภาพต้นแบบชื่อ **star 1** นี้ของเอกสารทุกครั้งที่มีการนำไปใช้



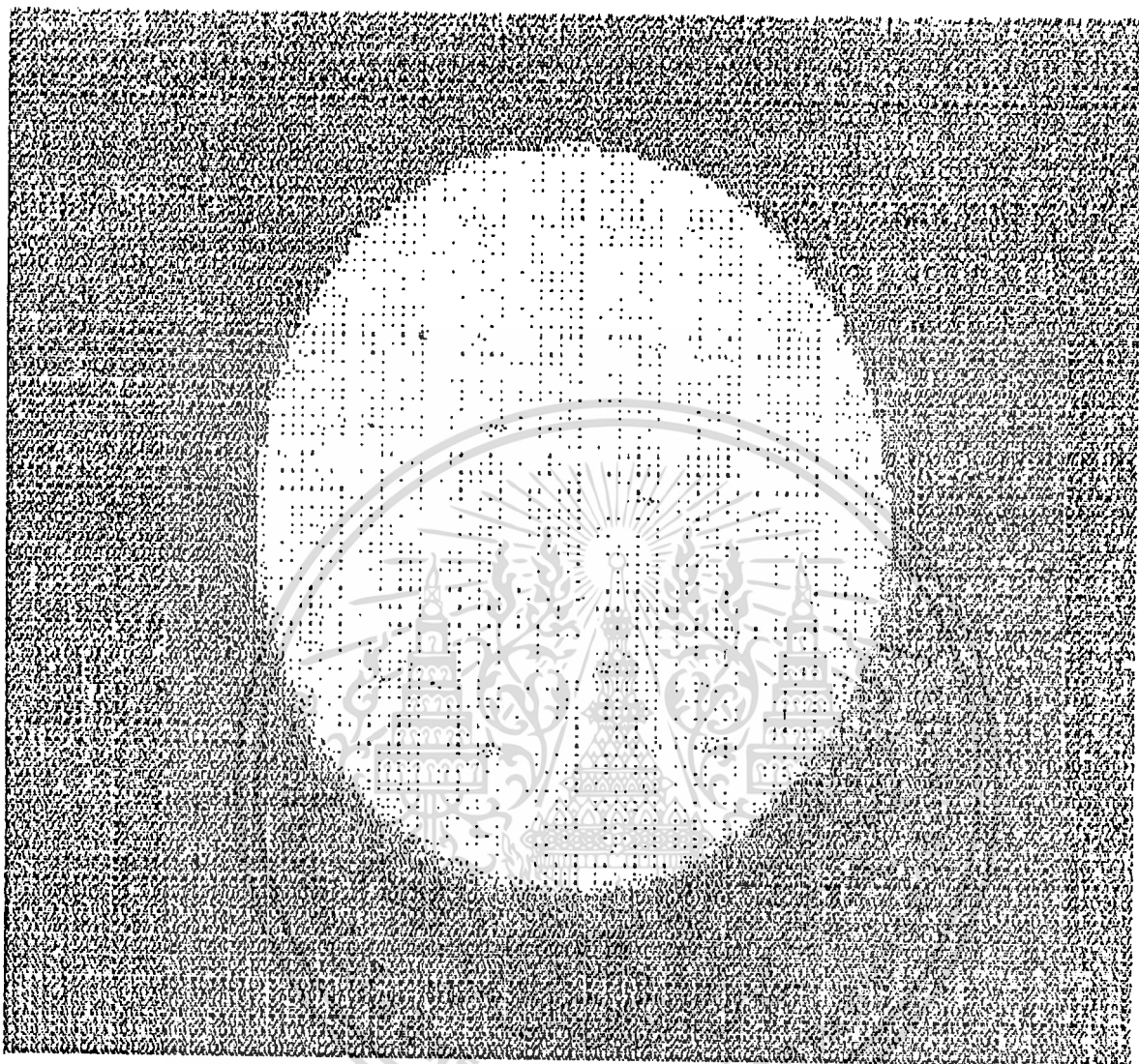
(ง) ภาพต้นแบบชื่อ square 2



(จ) ภาพต้นแบบชื่อ circle

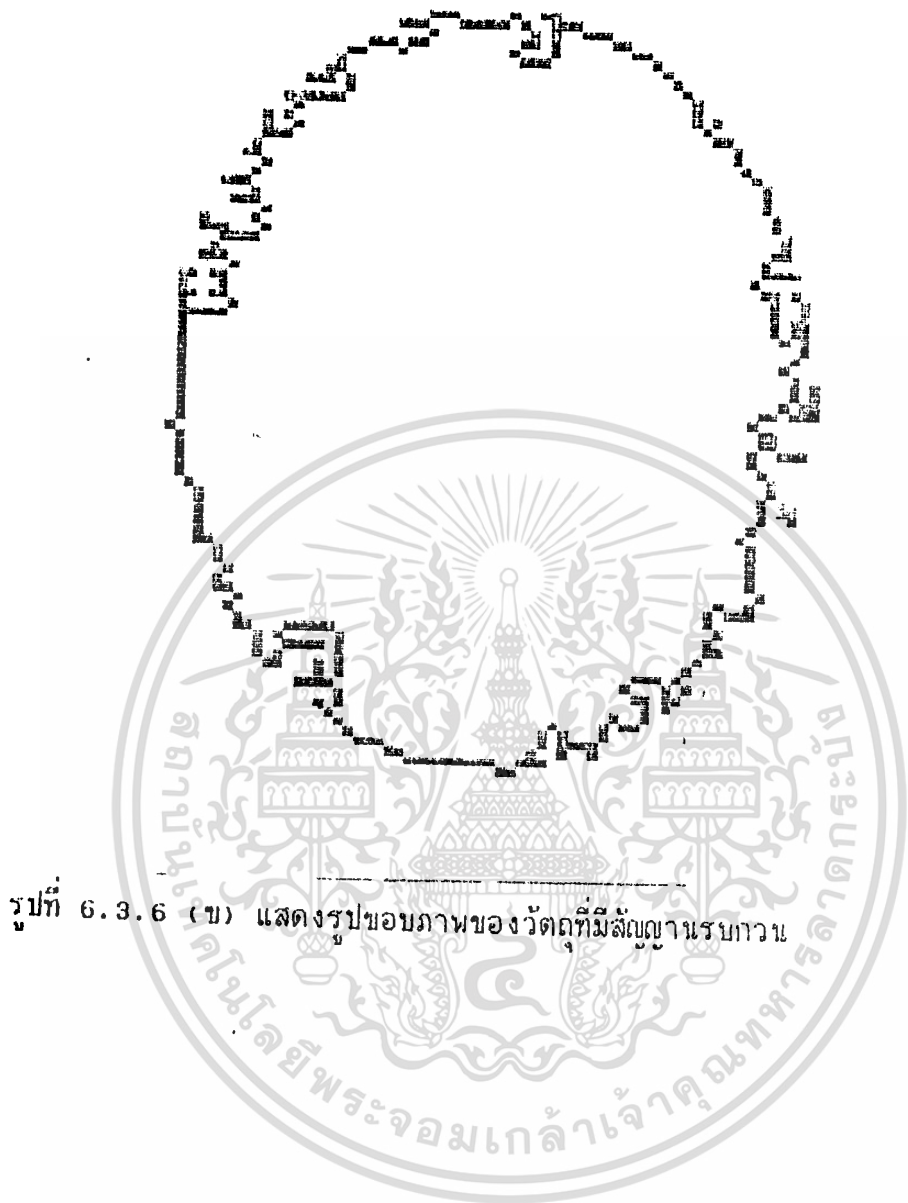


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น (จ) ภาพวัตถุที่สนใจชื่อ flipper 2



รูปที่ 6.3.6 (ก) แสดงวัตถุรูปวงกลมต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3.6 (ข) แสดงรูปขอบภาพของวัตถุที่มีสัญญาณรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

สรุปผลการทดลอง

จากกรทดลองเก็บภาพด้วยแผ่นวงจรติดต่อข้อมูลภาพ ขนาด 265*265 จุด แต่การแสดงผลได้แสดงเพียง 128*128 จุด เนื่องจากจอมอนิเตอร์ที่ใช้เป็นแบบ โมโนโครม (Monochrome) ซึ่งแต่ละจุดแสดงผลได้เพียงแค่ 2 ระดับ คือ สว่างและมืด ดังนั้น การที่จะทำให้ภาพมีน้ำหนักอ่อนแก่ จึงต้องใช้วิธีสร้างเมตริกของจุดขึ้นมา (half tone) ในวิธีนี้ใช้เมตริกขนาด 5*3 มี 15 จุด ทำให้แต่ละเมตริกสามารถแสดงระดับสว่างได้ทั้งหมด 16 ระดับ แต่การทำวิธีนี้ จำเป็นจะต้องสูญเสียความละเอียดของภาพไปบ้าง เพราะต้องขยายขนาดของจุดให้ใหญ่ขึ้น ในส่วนของภาพที่พิมพ์ออกทางเครื่องพิมพ์ก็ประสบกับปัญหานี้เช่นเดียวกัน เพราะเครื่องพิมพ์แบบดอทเมตริก (Dot Matrix) ก็สามารถแสดงผลได้เพียง 2 ระดับ จึงจำเป็นต้องใช้วิธีฮาล์ฟโทนเช่นเดียวกัน ภาพที่ได้แม้จะมีน้ำหนักแต่ก็ขาดความคมชัด จะต้องมองจากระยะห่างพอสมควร จึงจะสามารถเห็นรายละเอียดของภาพได้

การจำแนกวัตถุด้วยวิธีการแปลงใช้ฟูรีเยอร์ เดสคริปเตอร์ ที่ใช้ในปริญาณิพนธ์นี้ ใช้ได้ผลเป็นดังที่คาดหมาย โดยสามารถจำแนกวัตถุ ทั้งแบบเรขาคณิต และไม่ใช้เรขาคณิตได้ดี โดยมีข้อดีอยู่ที่สามารถจำแนกวัตถุ ที่มีผลกระทบของสัญญาณรบกวน ทำให้การหาขอบภาพผิดเพี้ยนไปจากวัตถุจริง ข้อดีของวิธีนี้อยู่ที่ขนาดของข้อมูลขอบภาพ โดยถ้าข้อมูลขอบภาพมีมาก จะทำให้เวลาในการจำแนกมากขึ้นอีก

สำหรับวิธีการจำแนกวัตถุทั้งสองวิธี ยังคงมีข้อจำกัดอยู่ที่ ข้อมูลต้องเป็นภาพในระนาบเดียว หรือต้องให้กล้องตั้งฉากกับวัตถุ เนื่องจากน้ำหนักของกล้องถ่ายภาพวิดีโอ ที่มีน้ำหนักมากเคลื่อนที่กล้องไม่สะดวก จึงจำเป็นที่จะต้องยึดติดอยู่กับที่ ทำให้ได้ภาพในระนาบเดียว วิธีการจำแนกวัตถุก็คงเป็นวิธีการจำแนกวัตถุในระนาบเดียวเช่นกันด้วย ทำให้ต้องใช้ภาพวัตถุที่เป็น 2 มิติเท่านั้น

ข้อจำกัดอีกประการในการจำแนกวัตถุ คือเวลาในการประมวลผล ซึ่งเครื่องคอมพิวเตอร์ที่ใช้งาน ไม่สามารถทำงานได้เร็วตามที่ต้องการ ทำให้ไม่สามารถจำแนกวัตถุได้ทันที ต้องใช้เวลาการคำนวณระยะหนึ่ง

ภาคผนวก

- (ก) โปรแกรมการแสดงภาพ
- (ข) โปรแกรมการจำแนกวัตถุโดยวิธีการหาส่วนประกอบของขอบภาพ
- (ค) โปรแกรมการจำแนกวัตถุโดยวิธีการฟูรีเยอร์ เดสคริปเตอร์



ภาคผนวก ก

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <dos.h>
#include <string.h>
#include <math.h>
#include <graphics.h>
#include <stdarg.h>

/* Constants */
#define GRAYLEV 0x3F
#define MAX(a,b) (((a) > (b)) ? (a) : (b))
#define MIN(a,b) (((a) < (b)) ? (a) : (b))

/* Global Variables */
char data1[128_1128_],
data2[128_1128_],
strbuff[31_],
path[31_];
int sumgray[GRAYLEV_],length,
check;

/* Function Prototypes */
void opengraph(void);
void plotscreen(char data[128_1128_]);
void histogram();
void loadfile(void);
void savefile(void);
void sequence(void);
void histonum(void);
void histopict(void);
void binary();
void transferdata();
void edge_detect();

void main(void)
{

```

```

strbuff[0] = 30;
opengraph();
sequence();
closegraph();
clrscr();
}

```

```

void opengraph(void)

```

```

{
    int    GraphDriver,
          GraphMode,
          ErrorCode;

    GraphDriver = DETECT;
    initgraph(&GraphDriver, &GraphMode, "");
    if((ErrorCode = graphresult()) != grOk)
    {
        printf("\n\nGraphic System Error! %d\n", grapherrormsg(ErrorCode));
        abort();
    }
}

```

```

void sequence(void)

```

```

{
    int    key;

    do
    {
        restorecrtmode();
        clrscr();
        printf("MAIN MENU");
        printf("\n\n    0 -> Exit");
        printf("\n\n    1 -> Transfer data");
        printf("\n\n    2 -> Save data to file");
        printf("\n\n    3 -> Load Original-Data");
        printf("\n\n    4 -> Histogram,numeric");
        printf("\n\n    5 -> Histogram,picture");
        printf("\n\n    6 -> Plotscreen");
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cprintf("\r\n 7 -> Plot binary picture");
cprintf("\r\n 8 -> Binary by user threshold");
cprintf("\r\n 9 -> Edge Detection");
cprintf("\r\n\nSelect : ");

```

```

switch(key = getche())
{
    case '1': transferdata();
                break;
    case '2': savefile();
                break;
    case '3': loadfile();
                break;
    case '4': histogram();
                histonam();
                break;
    case '5': histogram();
                histopict();
                break;
    case '6': plotscreen(data1);
                break;
    case '7': plotscreen(data2);
                break;
    case '8': binary();
                break;
    case '9': edge_detect();
                break;
}

```

```

} while(key!='0');
}

```

```

void loadfile(void)

```

```

{
    FILE *inFile;

    cprintf("\n\n");
    do
    {
        cprintf("\r\n");
    }
}

```

```

    fprintf("\rFile's Name : ");
    strcpy(path, "PIC\");
    strcat(path, cgets(strbuff));
    InFile = fopen(path, "rb");
} while((InFile==NULL)&&(strbuff[1]!=0));
if(InFile!=NULL)
{
    fprintf("\r\n\nLoading Original Data from %s", path);
    fread(data1, 128, 128, InFile);
    fclose(InFile);
}
}
void savefile(void)
{
    FILE #OutFile;

    fprintf("\n\n");
    do
    {
        fprintf("\r
        fprintf("\rFile's Name : ");
        strcpy(path, "PIC\");
        strcat(path, cgets(strbuff));
        OutFile = fopen(path, "wb");
    } while((OutFile==NULL)&&(strbuff[1]!=0));
    if(OutFile!=NULL)
    {
        fprintf("\r\n\nSaving Original Data to %s", path);
        fwrite(data1, 128, 128, OutFile);
        fclose(OutFile);
    }
}
}

void plotscreen(char data[128][128])
{
    int i, j, m, n, gray,
        x, y, pi;

    setygraphmode(&setygraphmode());

```

```

for(i=0; i<=127; i++)
{
    x = i*5;
    for(j=0; j<=114; j++)
    {
        y = j*3;
        gray = data[i][j]/4;
        switch(gray)
        {
            case 0:    p = 0x0000; break;
            case 1:    p = 0x0080; break;
            case 2:    p = 0x2020; break;
            case 3:    p = 0x0604; break;
            case 4:    p = 0x1209; break;
            case 5:    p = 0x4889; break;
            case 6:    p = 0x28b4; break;
            case 7:    p = 0x2aaa; break;
            case 8:    p = 0x5555; break;
            case 9:    p = 0x574b; break;
            case 10:   p = 0x9776; break;
            case 11:   p = 0x6df6; break;
            case 12:   p = 0x79fb; break;
            case 13:   p = 0x5ff4; break;
            case 14:   p = 0x7ff7; break;
            default:   p = 0xffff;
        }
        for(m=0; m<=2; m++)
            for(n=0; n<=4; n++)
            {
                if((p & 0x0001)!=0)
                    putpixel(x+n*40, y+m, 5);
                p >>= 1;
            }
    }
}

getch();
}

void edge_detect(void)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(
int    x, y;

printf("\n\n##### edge processing #####\n\n");
for(y=1; y<=126; y++){
    for(x=1; x<=126; x++){
        data2[x][y] = sqrt( pow( (data1[x+1][y-1]+2*data1[x+1][y]+data1[x+1][y+1])
                                -(data1[x-1][y-1]+2*data1[x-1][y]+data1[x-1][y+1]), 2)
                            +pow( (data1[x-1][y+1]+2*data1[x][y+1]+data1[x+1][y+1])
                                -(data1[x-1][y-1]+2*data1[x][y-1]+data1[x+1][y-1]), 2) );
    }
    printf("%s");
}
}

void    histogram()
{
int    n,j,i,m,k;

for(n=0; n<=63; n++)
    sumgray[n]=0;

for(i=0; i<=127; i++)
    for(j=0; j<=127; j++){
        m=data1[i][j];
        sumgray[m]++;
    }
check=0;
for(k=0; k<=GRAYLKV;k++)
    check+=sumgray[k-1]
}

void    histopict(void)
{
int    n,
        xleft=104,
        ybotb=274,
        result;

setgraphmode(getgraphmode());

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cleardevice();
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
outtextxy(300,20,"HISTOGRAM");
rectangle(99,69,621,274);
outtextxy(350,284,"GRAY LEVEL");
for (n=0x00; n<=GRAYLEV; n++){
    if(sumgray[n_>1){
        result = 40*log10(sumgray[n_);
        bar(xleft+8*n,ybott-result,(xleft+8*n)+6,ybott);
    }
}
getch();
}

void histonum(void)
{
    int i,j;
    printf("\n\n");
    for (i=0;i<=GRAYLEV;i+=4)
        for (j=0;j<=4 && j!=64; j++)
            printf("data[%2d_%2d] = %5d ",i+j,sumgray[i+j]);
    printf("\n");
    printf("All data = %d ",chock);
    getch();
}

void binary()
{
    int x, y,
        thres;
    cprintf("\r\n Threshold Value : ");
    scanf("%d",&thres);
    for(y=0; y<=127; y++)
        for(x=0; x<=127; x++)
            if(data1[x_>thres)
                data2[x_ = 0x00;
            else
                data2[x_ = 0x3f;
}

```

```

void transferdata(void)
{
    char far *ptr;
    int i, j;

    ptr = MK_FP(0xC000, 0);

    for(j=0; j<=254; j+=2)
        for(i=0; i<=254; i+=2)
            data1[i/2][j/2] = *(ptr+(i+i)+(j+1)*256)&0x3f;
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <dos.h>
#include <string.h>
#include <math.h>
#include <graphics.h>

/* Constants */
#define PI 3.141592654
#define MAX_OBJ 20
#define MAX_DATA 50
#define LINE_SECTION 10
#define Y_RATIO 0.7

/* Global Variables */
unsigned char
    image1[128][128],
    image2[128][128],
    strbuf[30],
    path[30];

struct RECTAN
{
    float x1;
    float y1;
} track_data[LINE_SECTION + 1];

struct POLAR
{
    float r;
    float th;
} data1[MAX_DATA];

struct POLAR data2[MAX_DATA];

struct OBJECT

```

```

{
    int    select;
    float  obj_error,
          head_ref,
          heading;
    char   name[16];
    int    line_count;
    struct POLAR data[MAX_DATA];
} obj[MAX_OBJ];

char   section = 1;
float  ref_angle,
       last_angle;
int    line_count,
       obj_count;

/* Function Prototypes */
void  opengraph(void);
void  mainmenu(void);
void  look_object(void);
int   select_obj(void);
int   compare_data(void);
void  build_database(void);
void  copydata(char #dest, char #src);
void  transferdata(void);
void  loadfile(void);
void  plotscreen(unsigned char data[128][128]);
void  edge_detect(void);
void  line_filter(void);
void  edge_track(void);
void  code_gen(void);
float add_angle(float, float);
int   findbres(void);
int   findframe(int #x1, int #y1, int #x2, int #y2);
void  line_analyze(int);

void  main(void)
{
    opengraph();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mainmenu();
closegraph();
}

void opengraph(void)
{
    int GraphDriver,
        GraphMode,
        ErrorCode;

    GraphDriver = DETECT;
    clrscr();
    cprintf("\rPath of BGI : ");
    strbuff[0] = 13;
    strcpy(path, cgets(strbuff));
    initgraph(&GraphDriver, &GraphMode, path );
    if((ErrorCode = graphresult()) != grOk)
    {
        cprintf("\r\nGraphics System Error: %s\r\n", grapherrormsg(ErrorCode));
        abort();
    }
}

void mainmenu(void)
{
    int key;

    do
    {
        restorecrtmode();
        clrscr();
        cprintf("MAIN MENU");
        cprintf("\r\n\n 0 -> Exit");
        cprintf("\r\n\n 1 -> Build Database");
        cprintf("\r\n\n 2 -> Look Object");
        cprintf("\r\n\nSelect : ");

        switch(key = getch())
        {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        case '1': build_database();
                break;

        case '2': look_object();
    }
} while(key!='0');
}

void look_object(void)
{
    int key,
        select;
    int count;
    FILE *InFile;

    restorecrmode();
    clrscr();
    printf("\rLook Object ..\n\n");

    do
    {
        printf("\rDatabase's Name : ");
        strcpy(path, "P(C\\");
        strbuf[0] = '\0';
        strcat(path, cgets(strbuf));
        InFile = fopen(path, "rb");
    } while((InFile==NULL)&&(strbuf[0]!='\0'));
    if(InFile!=NULL)
    {
        obj_count = 0;

        while(fread(obj[obj_count].name, sizeof(char), 15, InFile) != 0)
        {
            fread(&obj[obj_count].line_count, sizeof(int), 1, InFile);
            fread(&obj[obj_count].head_ref, sizeof(float), 1, InFile);
            for(count = 0; count < obj[obj_count].line_count; ++count)
                fread(&obj[obj_count].data[count].r, sizeof(float), 2, InFile);
            ++obj_count;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
fclose(InFile);

do
{
    restorecrtmode();
    clrscr();
    cprintf("Look Object ..");
    cprintf("\r\n\n    0 -> End");
    cprintf("\r\n\n    1 -> Look From Camera");
    cprintf("\r\n\n    2 -> Look From Image File");
    cprintf("\r\n\nSelect : ");

    switch(key = getch())
    {
        case '1':  transferdata();
                   plotscreen(image1);
                   edge_detect();
                   plotscreen(image2);
                   edge_track();
                   plotscreen(image1);
                   line_filter();
                   code_gen();

                   restorecrtmode();
                   clrscr();

                   compare_data();
                   break;

        case '2':  loadfile();
                   plotscreen(image1);

                   edge_detect();
                   plotscreen(image2);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

edge_track();
plotscreen(image!);

line_filter();
code_gen();

restorecrtmode();
clrscr();

compare_data();
break;
}

} while(key != '0');
}
}

int select_obj(void)
{
    int index,
        count;

    count = 0;

    for(index = 0; index < obj_count; ++index)
    {
        if(obj[index].line_count == line_count)
        {
            obj[index].select = 1;
            ++count;
        }
        else
            obj[index].select = 0;
    }

    return count;
}
}

```

```
int compare_data(void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int    obj_index, min_index,
       index1, index2)
float  data_error[MAX_DATA];
float  heading;

restorecrtmode();
clrscr();

if(select_obj() == 0)
{
    cprintf("\n\nUnable to recognize ..");
    return 1;
}

for(obj_index = 0; obj_index < obj_count; ++obj_index)
{
    if(obj[obj_index].select == 1)
    {
        for(index1 = 0; index1 < obj[obj_index].line_count; ++index1)
        {
            data_error[index1] = 0;
            for(index2 = 0; index2 < obj[obj_index].line_count; ++index2)
            {
                data_error[index1] += pow(obj[obj_index].data[index2].r
                    - data1[(index1 + index2) % obj[obj_index].line_count_r, 2)
                    + pow(obj[obj_index].data[index2].th
                    - data1[(index1 + index2) % obj[obj_index].line_count_th, 2);
            }
            data_error[index1] = sqrt(data_error[index1]);
        }

        min_index = 0;

        for(index1 = 0; index1 < obj[obj_index].line_count; ++index1)
            if(data_error[index1] < data_error[min_index])
                min_index = index1;

        obj[obj_index].obj_error = data_error[min_index];
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

obj[obj_index].heading = add_angle(data2[min_index].th , -(obj[obj_index].head_ref));
}
}

min_index = 0;
while(obj[min_index].select != 1)
    ++min_index;

for(obj_index = 0; obj_index < obj_count; ++obj_index)
    if(obj[obj_index].select == 1)
        if(obj[obj_index].obj_error < obj[min_index].obj_error)
            min_index = obj_index;

printf("\r\n\nIt's a %s", obj[min_index].name);
printf("\r\n\nError = %f6.2", obj[min_index].obj_error);
printf("\r\n\nHeading = %f6.2 degree", obj[min_index].heading);
getch();

return 0;
}

void build_database(void)
{
    int    key, count;
    FILE  *OutFile;
    char   obj_name[16];

    restorecrtmode();
    clrscr();
    printf("\r\nBuild Database ..\n\n");

    do
    {
        printf("\r\nDatabase's Name : ");
        strcpy(path, "PIC\\");
        strbuff[0] = 13;
        strcat(path, cgets(strbuff));
        OutFile = fopen(path, "wb");
    } while((OutFile==NULL)&&(strbuff[1]!=0));
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(OutFile!=NULL)
{
do
{
restorecrmode();
clrscr();
cprintf("Build Database ..");
cprintf("\r\n\n 0 -> End");
cprintf("\r\n\n 1 -> Get Image From Camera");
cprintf("\r\n\n 2 -> Get Image From File");
cprintf("\r\n\n 3 -> Save Image To Database");
cprintf("\r\n\nSelect : ");

switch(key = getch())
{
case '1': transferdata();
plotscreen(image1);
edge_detect();
plotscreen(image2);
edge_track();
plotscreen(image1);
line_filter();
code_gen();
break;

case '2': loadfile();
plotscreen(image1);

edge_detect();
plotscreen(image2);

edge_track();
plotscreen(image1);

line_filter();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

code_gen();
break;

case '3': fprintf("\r\n\nObject's Name : ");
strbuff[0] = 15;
strcpy(obj_name, cgets(strbuff));

fwrite(obj_name, sizeof(char), 15, OutFile);
fwrite(&line_count, sizeof(int), 1, OutFile);
fwrite(&data2[0].th, sizeof(float), 1, OutFile);
for(count = 0; count < line_count; ++count)
    fwrite(&data1[count].r, sizeof(float), 2, OutFile);
}

} while(key != '0');

fclose(OutFile);
}
}

void copydata(char #dest, char #src)
{
int i;

for(i=0; i < (128*128); ++i)
    *(dest+i) = *(src+i);
}

void transferdata(void)
{
char far #ptr;
int i, j;

ptr = MK_FP(0xC000, 0);

for(j=0; j<=254; j+=2)
    for(i = 0; i <= 254; i += 2)
        image1[i/2][j/2] = *(ptr+(i+1)+(j+1)*256)*0x3f;
}

```

```

void loadfile(void)
{
    FILE *InFile;

    cprintf("\n\n");
    do
    {
        cprintf("\r
        ");
        cprintf("\rFile's Name : ");
        strcpy(path, "PIC\\");
        strbuff[0] = 13;
        strcat(path, cgets(strbuff));
        InFile = fopen(path, "rb");
    } while((InFile==NULL)&&(strbuff[1]!=0));

    if(InFile!=NULL)
    {
        cprintf("\r\n\nLoading Original Data from %s", path);
        fread(image1, 128, 128, InFile);
        fclose(InFile);
    }
}

void plotscreen(unsigned char data[128][128])
{
    int i, j, m, n, gray,
        x, y, p;

    setgraphmode(getgraphmode());
    for(i=0; i<=127; i++)
    {
        x = i*5;
        for(j=0; j<=114; j++)
        {
            y = j*3;
            gray = data[i][j]/4;
            switch(gray)
            {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```
-(imagel[x-1][y-1]+2*imagel[x][y-1]+imagel[x+1][y-1], 2) );
```

```
}
```

```
void code_gen(void)
```

```
{
```

```
int index;
```

```
data1[0].r = data2[0].r;
```

```
data1[0].th = add_angle(data2[0].th, -(data2[line_count-1].th));
```

```
restorecrtmode();
```

```
clrscr();
```

```
cprintf("\rLine = %d", line_count);
```

```
cprintf("\r\n\nData =\n");
```

```
for(index = 1; index < line_count; ++index)
```

```
{
```

```
data1[index].r = data2[index].r;
```

```
data1[index].th = add_angle(data2[index].th, -(data2[index-1].th));
```

```
cprintf("\r\n %f6.2 %f6.2", data1[index].r, data1[index].th);
```

```
}
```

```
getch();
```

```
}
```

```
void line_filter(void)
```

```
{
```

```
int index1, index2, count1, count2;
```

```
struct RECTAN temp1, temp2;
```

```
count1 = line_count;
```

```
for(index1 = 0; index1 < count1; ++index1)
```

```
{
```

```
if(data1[index1].r < 10)
```

```
{
```

```
data1[index1].r = data1[index1].r / 2;
```

```
temp1.x = data1[index1].r * cos(PI/180 * data1[index1].th);
```

```
temp1.y = data1[index1].r * sin(PI/180 * data1[index1].th);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(count2 = 1; count2 <= 2; ++count2)
{
    index2 = index1;

    if(count2 == 1)
    {
        do
        {
            --index2;
            if(index2 < 0)
                index2 = count1 - 1;
        } while(data1[index2].r == 0);
    }
    else
    {
        do
        {
            ++index2;
            if(index2 > count1-1)
                index2 = 0;
        } while(data1[index2].r == 0);
    }

    temp2.x = data1[index2].r * cos(Pi/180 * data1[index2].th);
    temp2.y = data1[index2].r * sin(Pi/180 * data1[index2].th);

    temp2.x += temp1.x;
    temp2.y += temp1.y;

    data1[index2].r = sqrt(pow(temp2.x,2) + pow(temp2.y,2));
    data1[index2].th = 180/Pi * atan2(temp2.y, temp2.x);
}

data1[index1].r = 0;
data1[index1].th = 0;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(index1 = 0, index2 = 0; index1 < count1; ++index1)
{
    if(data1[index1...r] != 0)
    {
        data2[index2...r] = data1[index1...r]
        data2[index2...th] = data1[index1...th];
        ++index2;
    }
    else
    {
        --line_count;
    }
}
}

```

```

void edge_track(void)
{
    int x, y, thres, found;
    int max, count, data1[8];
    int start_x, start_y, loop;
    int index;

    thres = findthres();
    for(y = 0; y < 128; ++y)
        for(x = 0; x < 128; ++x)
            image1[x...y] = 0;

    for(found = 0, y = 1; (y < 127) && !found; ++y)
        for(x = 1; x < 127; ++x)
            if(image2[x...y] >= thres)
            {
                found = 1;
                break;
            }

    start_x = x;
    start_y = y;
    loop = 0;
    index = 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

track_data[0].x = x;
track_data[0].y = y;
ref_angle = 0;
last_angle = 0;
line_count = 0;

do
{
    image1[x][y] = 0x3f;

    if(index < LINE_SECTION)
    {
        ++index;
        track_data[index].x = x;
        track_data[index].y = y;
    }
    else
    {
        line_analyze(index);
        index = 0;
    }

    data[0] = image2[x + 1][y - 1];
    data[1] = image2[x + 1][y + 1];
    data[2] = image2[x - 1][y + 1];
    data[3] = image2[x - 1][y - 1];
    data[4] = image2[x - 1][y];
    data[5] = image2[x - 1][y - 1];
    data[6] = image2[x - 1][y + 1];
    data[7] = image2[x + 1][y - 1];

    if((image1[x + 1][y - 1] != 0) || (loop < 5))
    {
        data[7] = -1;
        data[0] = -1;
        data[1] = -1;
    }

    if(image1[x + 1][y + 1] != 0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    data[0_] = -1;
    data[1_] = -1;
    data[2_] = -1;
}

if(image1[x + 1, y + 1] != 0)
{
    data[1_] = -1;
    data[2_] = -1;
    data[3_] = -1;
}

if(image1[x - 1, y + 1] != 0)
{
    data[2_] = -1;
    data[3_] = -1;
    data[4_] = -1;
}

if(image1[x - 1, y - 1] != 0)
{
    data[3_] = -1;
    data[4_] = -1;
    data[5_] = -1;
}

if(image1[x - 1, y - 1] != 0)
{
    data[4_] = -1;
    data[5_] = -1;
    data[6_] = -1;
}

if(image1[x - 1, y - 1] != 0)
{
    data[5_] = -1;
    data[6_] = -1;
    data[7_] = -1;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

if(imageI[x + 1][y - 1] != 0)
{
    data[6_] = -1;
    data[7_] = -1;
    data[0_] = -1;
}

for(max = 0, count = 1; count < 8; ++count)
    if(data[count_] > data[max_])
        max = count;

switch(max)
{
case 0: ++x;
        break;

case 1: ++x; ++y;
        break;

case 2: ++y;
        break;

case 3: --x; ++y;
        break;

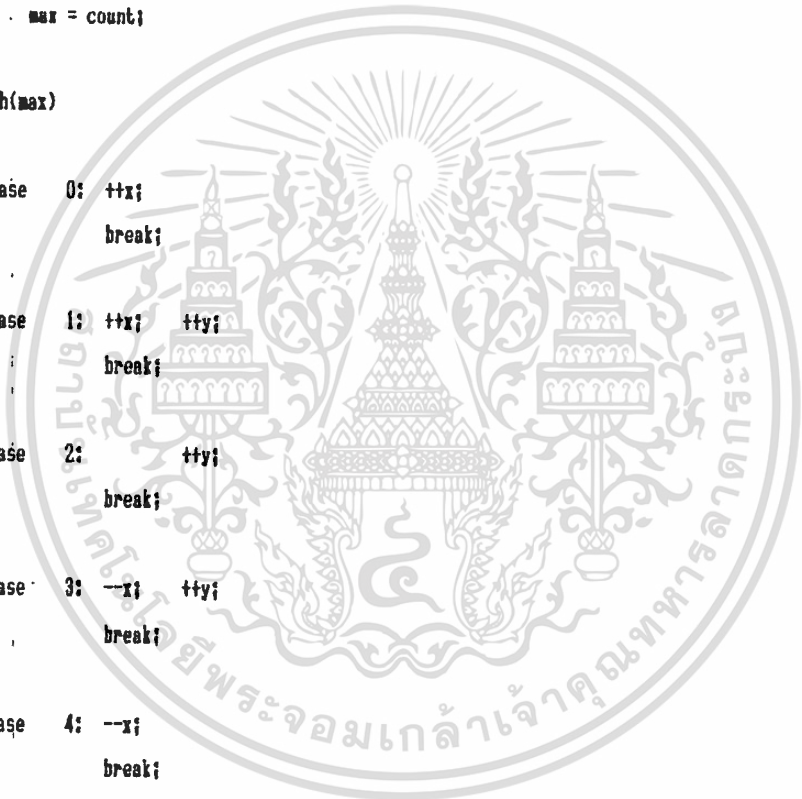
case 4: --x;
        break;

case 5: --x; --y;
        break;

case 6: --y;
        break;

case 7: ++x; --y;
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    ++loop;
} while((loop < 5) || (abs(start_x - x) > 1) || (abs(start_y - y) > 1));

track_data[1..x] = start_x;
track_data[1..y] = start_y;
line_analyze(0);
}

float add_angle(float th1, float th2)
{
    float temp;

    temp = th1 + th2;

    while(temp > 180.0)
        temp -= 360.0;

    while(temp < -180.0)
        temp += 360.0;

    return temp;
}

void line_analyze(int count)
{
    int index1, index2;
    float slope, distance;
    struct RECTAN delta;

    index2 = line_count;
    index1 = 0;

    do
    {
        if(track_data[count..x] != track_data[1..x])
            slope = ((float)track_data[count..y] - (float)track_data[1..y]) /
                    ((float)track_data[count..x] - (float)track_data[1..x]);
        else
            if(track_data[count..y] > track_data[1..y])

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        slope = 3.4E+38;
    else
        slope = -3.4E+38;

do
{
    ++index1;
    distance = fabs((-slope)*(float)track_data[index1].x + (float)track_data[index1].y
        - (float)track_data[index1].y + slope*(float)track_data[index1].x) /
        sqrt(pow(slope,2) + 1.0);

} while((distance <= 1.0) && (index1 < count));

if((distance > 1.0) || (count == 0))
{
    delta.x = track_data[index1].x - track_data[0].x;
    delta.y = track_data[index1].y - track_data[0].y;

    data[index2].r = sqrt(pow(delta.x, 2) + pow(delta.y*Y_RATIO, 2));
    data[index2].th = 180.0/PI * atan2(delta.y*Y_RATIO, delta.x);
    track_data[0].x = track_data[index1].x;
    track_data[0].y = track_data[index1].y;
    ++index2;
    section = 1;
}

} while(index1 < count);

if(line_count == index2)
{
    delta.x = track_data[count].x - track_data[1].x;
    delta.y = track_data[count].y - track_data[1].y;

    if(section == 1)
    {
        ref_angle = 180.0/PI * atan2(delta.y*Y_RATIO, delta.x);
        section = 0;
    }
}
else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

last_angle = 180.0/PI * atan2(delta.y*Y_RATIO, delta.x);
if(fabs(add_angle(ref_angle, -(last_angle))) > 20.0)
{
    delta.x = track_data[l_.x] - track_data[0_.x];
    delta.y = track_data[l_.y] - track_data[0_.y];

    data[index2_.r] = sqrt(pow(delta.x, 2) + pow(delta.y*Y_RATIO, 2));
    data[index2_.th] = ref_angle;

    ref_angle = last_angle;

    track_data[0_.x] = track_data[l_.x];
    track_data[0_.y] = track_data[l_.y];

    ++index2;
}
}
}
line_count = index2;
}
int findthres(void)
{
    int i, j,
        max = 0,
        min = 64;

    for(j=0; j<=127; ++j)
        for(i=0; i<=127; ++i)
        {
            if(image2[i][j] > max)
                max = image2[i][j];
            if(image2[i][j] < min)
                min = image2[i][j];
        }

    return (max + min)/2;
}

```

ภาคผนวก ค

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <dos.h>
#include <string.h>
#include <math.h>
#include <graphics.h>
#include <stdarg.h>

/***** Constants *****/
#define GRAYLEV 0x3F
#define MAX(a,b) (((a) > (b)) ? (a) : (b))
#define MIN(a,b) (((a) < (b)) ? (a) : (b))
#define MAX_OBJ 20
#define UNKNOWN 200

/***** Global Variables *****/
char data1[128][128],
data2[128][128],
strbuff[31],
path[31];

int length, obj_count;

float nfd_data[32];

struct OBJECT
{
    char name[16];
    float nfd[32];
}obj[MAX_OBJ];

/***** Complex structure *****/
typedef struct {
    float real , imag;
} COMPLEX;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/##### Function Prototypes #####/
void  opengraph(void);
void  plotscreen(char data[128][128]);
void  build_database(void);
void  look_object(void);
int   select_obj(void);
void  loadfile(void);
void  savefile(void);
void  sequence(void);
void  binary();
void  transferdata();
void  track( char data[128][128_]);
void  dft(COMPLEX #Datain , COMPLEX #Dataout , int n);
void  magnphase(COMPLEX #Datain , COMPLEX #Dataout , int n);
float msqerr(int k);
void  putscreen(COMPLEX #Datain , float minex , int n);

void  main(void)
{
    strbuff[0] = 30;
    opengraph();
    sequence();
    closegraph();
    clrscr();
}

void  opengraph(void)
{
    int  GraphDriver,
        GraphMode,
        ErrorCode;

    GraphDriver = DETECT;
    initgraph(&GraphDriver, &GraphMode, "" );
    if((ErrorCode = graphresult()) != grOk)
    {
        printf("\n\nGraphics System Error: %s\n", grapherrormsg(ErrorCode));
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        abort();
    }
}

void sequence(void)
{
    int key;

    do
    {
        restorecrtmode();
        clrscr();
        cprintf("MAIN MENU");
        cprintf("\r\n\n 0 -> Exit");
        cprintf("\r\n\n 1 -> ##MAKING DATABASE ROUTINE##");
        cprintf("\r\n\n 2 -> ##FINDING OBJECT ROUTINE##");
        cprintf("\r\n\nSelect : ");

        switch(key = getch())
        {
            case '1': build_database();
                    break;
            case '2': look_object();
                    break;
        }
    } while(key!='0');
}

void build_database(void)
{
    int key, count;
    FILE #Out;File;
    char obj;name[16];

    restorecrtmode();
    clrscr();
    cprintf("\r\nBuild Database ..\n\n");

    do
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cprintf("\rDatabase's Name : ");
strcpy(path, "PIC\\");
strbuff[0] = 13;
strcat(path, cgets(strbuff));
OutFile = fopen(path, "wb");
} while((OutFile==NULL)&&(strbuff[1]!=0));

if(OutFile!=NULL)
{
do
{
restorecrtmode();
clrscr();
cprintf("Build Database ..");
cprintf("\r\n\n 0 -> End");
cprintf("\r\n\n 1 -> Get Image From Camera");
cprintf("\r\n\n 2 -> Out Image From File");
cprintf("\r\n\n 3 -> Save Image To Database");
cprintf("\r\n\nSelect : ");

switch(key = getch())
{
case '1': transferdata();
binary();
track(data2);
break;

case '2': loadfile();
binary();
track(data2);
break;

case '3': cprintf("\r\n\nObject's Name : ");
strbuff[0] = 15;
strcpy(obj_name, cgets(strbuff));

fwrite(obj_name, sizeof(char), 15, OutFile);
fwrite(nfd_dat1, sizeof(float), 32, OutFile);
}
}

```

```

    } while(key != '0');

    fclose(OutFile);
}
}

void look_object(void)
{
    int key;
        select;
    int count;
    FILE *InFile;

    restorecrmode();
    clrscr();
    cprintf("\rLook Object ..\n\n");

do
    (
        cprintf("\rDatabase's Name : ");
        strcpy(path, "PIC\");
        strbuff[0] = 13;
        strcut(path, cgets(strbuff));
        InFile = fopen(path, "rb");
    } while((InFile==NULL)&&(strbuff[1]!=0));
    if(InFile!=NULL)
    {
        obj_count = 0;

        while(fread(obj[obj_count].name, sizeof(char), 15, InFile) != 0)
        {
            fread(obj[obj_count].nfd, sizeof(float), 32, InFile);
            ++obj_count;
        }
        fclose(InFile);
    }
do

```

```

restorecrtmode();
clrscr();
printf("Look Object ..");
printf("\r\n\n 0 -> End");
printf("\r\n\n 1 -> Look From Camera");
printf("\r\n\n 2 -> Look From Image File");
printf("\r\n\nSelect : ");

switch(key = getch())
{
case '1': transferdata();
binary();
track(data2);
restorecrtmode();
clrscr();
if((select = select_obj()) != UNKNOWN)
{
clrscr();
printf("\n\n\n");
printf("\r\n\n\n *****
HEY BABY!
IT'S A %s",obj[select...n
*****
}
else
{
clrscr();
printf("\n\n\n");
printf("\r\n\n\n ?????????????????????????????
SORRY BABY!
I CAN'T KNOW WHAT IT IS
????????????????????????????
}
getch();
break;

```

```

case '2': loadfile();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

binary();
track(data2);
restorecrtmode();
clrscr();

if((select = select_obj()) != UNKNOWN)
{
    clrscr();
    cprintf("\n\n\n");
    cprintf("\r\n\n\n          #####\n");
    cprintf("\r\n\n\n          HEY BABY!\n");
    cprintf("\r\n\n\n          IT'S A %s",obj[select_..n\n");
    cprintf("\r\n\n\n          #####\n");
}
else
{
    clrscr();
    cprintf("\n\n\n");
    cprintf("\r\n\n\n          ?????????????????????\n");
    cprintf("\r\n\n\n          SORRY BABY!\n");
    cprintf("\r\n\n\n          I CAN'T KNOW WHAT IT IS\n");
    cprintf("\r\n\n\n          ?????????????????????\n");
}
getch();
break;
}

} while(key != '0');
}
}

```

```

int select_obj(void)
{
    int i, d_out = 0;
    float mi_err = UNKNOWN ,t_error[20];

    for( i = 0 ; i < obj_count ; i++){
        t_error[i_] = msqerr(i);
        mi_err = MIX(mi_err,t_error[i_]);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if(mi_err == t_error[i]){
        d_out = i ;
    }
}
getch();
if(mi_err>0.21)
    return(UNKNOWN);
else
    return(d_out);
}

void loadfile(void)
{
    FILE #InFile;

    cprintf("\n\n");
    do
    {
        cprintf("\r
        cprintf("\rFile's Name : ");
        strcpy(path, "PIC\");
        strcat(path, cgets(strbuff));
        InFile = fopen(path, "rb");
    } while((InFile==NULL)&&(strbuff[1]!=0));
    if(InFile!=NULL)
    {
        cprintf("\r\n\nLoading Original Data from %s", path);
        fread(data1, 128, 128, InFile);
        fclose(InFile);
    }
}

void savefile(void)
{
    FILE #OutFile;

    cprintf("\n\n");
    do

```

```

{
    cprintf("\r                                     ");
    cprintf("\rFile's Name : ");
    strcpy(path, "PIC\\");
    strcat(path, cgets(strbuff));
    OutFile = fopen(path, "wb");
} while((OutFile==NULL)&&(strbuff[1]!=0));
if(OutFile!=NULL)
{
    cprintf("\r\n\nSaving Original Data to %s", path);
    fwrite(data1, 128, 128, OutFile);
    fclose(OutFile);
}
}

void plotucrun(char data[128][128])
{
    int i, j, m, n, gray,
        x, y, p;

    setgraphmode(getgraphmode());
    for(i=0; i<=127; i++)
    {
        x = 185;
        for(j=0; j<=114; j++)
        {
            y = j*3;
            gray = data[i][j]/4;
            switch(gray)
            {
                case 0: p = 0x0000; break;
                case 1: p = 0x0000; break;
                case 2: p = 0x2020; break;
                case 3: p = 0x0604; break;
                case 4: p = 0x1209; break;
                case 5: p = 0x4889; break;
                case 6: p = 0x28b4; break;
                case 7: p = 0x2aaa; break;
                case 8: p = 0x5555; break;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        case 9:    p = 0x574b; break;
        case 10:   p = 0x3776; break;
        case 11:   p = 0x6df6; break;
        case 12:   p = 0x79fb; break;
        case 13:   p = 0x5ffd; break;
        case 14:   p = 0x77ff; break;
        default:   p = 0x7fff;
    }

    for(m=0; m<=2; m++)
        for(n=0; n<=4; n++)
            [
                if((p & 0x0001)!=0)
                    putpixel(x+n*40, y+m, 5);
                p >>= 1;
            ]
        ]
    ]

void binary()
{
    int    x, y,
           thres;
    printf("\n\n Threshold Value : ");
    scanf("%d",&thres);
    for(y=0; y<=127; y++)
        for(x=0; x<=127; x++)
            if(data1[x][y]>=thres)
                data2[x][y] = 0x00;
            else
                data2[x][y] = 0x3f;
}

void transferdata(void)
{
    char far  *ptr;
    int       i, j;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ptr = MK_FP(0xC000, 0);

for(j=0; j<254; j+=2)
    for(i=0; i<254; i+=2)
        data1[i/2_1j/2_ = *(ptr+(i+1)+(j+1)*256)&0x3f;
}

void dft(COMPLEX #Datain,COMPLEX #Dataout,int N)
{
    int i,k,n,pi;
    static int nstore = 0;    /* store N for future use */
    static COMPLEX #cf;      /* coefficient storage */
    COMPLEX #cfptr,#Dinptr;
    double arg;

/* Create the coefficients if N has changed */

if(N != nstore) {
    if(nstore != 0) free((char #) cf); /* free previous */
    cf = (COMPLEX #) calloc(N, sizeof(COMPLEX));
    if (!cf) {
        printf("\nUnable to allocate memory for coefficients.\n");
        exit(1);
    }

    arg = 0.0#atan(1.0)/N;
    for (i=0 ; i<N ; i++) {
        cf[i]_.real = (float)cos(arg#i);
        cf[i]_.imag = -(float)sin(arg#i);
    }
}

/* Perform the DFT calculation */

printf("\n");
for (k=0 ; k<N ; k++) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dinptr = Datain;
Dataout->real = Dinptr->real;
Dataout->imag = Dinptr->imag;
Dinptr++;
for (n=1; n<N; n++) {

    p = (int)((long)n&k % N);
    cfptr = cf + p;      /* pointer to cf modulo N */

    Dataout->real += Dinptr->real * cfptr->real
                  - Dinptr->imag * cfptr->imag;

    Dataout->imag += Dinptr->real * cfptr->imag
                  + Dinptr->imag * cfptr->real;
    Dinptr++;
}
if (k % 32 == 31) printf("%s");
Dataout++;      /* next output */
}
printf("\n");
}

void magniphase(COMPLEX #Datain,COMPLEX #Dataout,int n)
{
    int i,j;
    COMPLEX #xi,#yi;
    float dvide = 0.0;
    xi = Datain;
    yi = Dataout;
    for(i = 0; i <= n-1; i++){
        xi->real *= xi->real;
        xi->real += xi->imag * xi->imag;
        Dataout->real =sqrt((double)xi->real);
        dvide = MAX(dvide,Dataout->real);
        xi++;
        Dataout++;
    }
    Dataout = yi ;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

COMPLEX  #cpxin,#tempcpx,#cpxout,
        #outtemp,#magnit;

float  minex = 0.0 ;

cpxin = (COMPLEX #) calloc(512,sizeof(COMPLEX));
if(!cpxin) {
    printf("\nError allocating data memory \n");
    exit(1);
}
tempcpx = cpxin ;

for( y=0;y<=127;y++ )
for( x=0;x<=127;x++ )
    data[x][y] = 0x00;

cprintf("\n\rSCANNING PROCESS AT (");
for( y=0 ; y<=127 ; y++ ){
    printf("%3d_\\b\\b\\b",y);
    delay(0);
    for( x=0 ; x<=127 ; x++ )
        if(data[x][y]==0x3f)
            break;
        if(data[x][y]==0x3f)
            break;
}

if( x==128 && y==128 ){
    cprintf("\r\nCAN'T FIND ANY DOT ON THIS IMAGE");
    getch();
}

else{
    printf("\n\rFINDING AN IMAGE AT Y= %d AND X = %d\n ",y,x);
    delay(0);

/##### tracking process #####/
    start_x = x;
    start_y = y;
    length = 0;

```

```

vector = 1;
cpxin->real = 0.0;
cpxin->imag = 0.0;
cpxin++;
do
  do
    if(vector==1)
      vector--;
    else vector = 7;

    switch(vector){
      case 0 : x_search = x+1 ; y_search = y;
              break ;
      case 7 : x_search = x+1 ; y_search = y+1;
              break ;
      case 6 : x_search = x   ; y_search = y+1;
              break ;
      case 5 : x_search = x-1 ; y_search = y+1;
              break ;
      case 4 : x_search = x-1 ; y_search = y;
              break ;
      case 3 : x_search = x-1 ; y_search = y-1;
              break ;
      case 2 : x_search = x   ; y_search = y-1;
              break ;
      case 1 : x_search = x+1 ; y_search = y-1;
              break ;
    }
  } while( data[x_search][y_search] != 0x3f );
x = x_search;
y = y_search;
cpxin->real = (float)(x-start_x);
cpxin->imag = ((float)(y-start_y) * (-1.0));
minex = MIN(minex,cpxin->real);
data[x][y]=0x3f;
cpxin++;
length++;

if(vector <=4)

```

```

        vector +=3 ;
    else if (vector == 6 )
        vector = 1 ;
    else if (vector == 7 )
        vector = 2 ;
    else vector = 0 ;
} while(x != start_x || y != start_y);
length++;
cpxin = tempcpx ;
plotscreen(data1);
putscreen(cpxin,minex,length);
/* for( b=0 ; b (<= 24 ; b++){
    printf("\r\datain[%3d_ = %f %f ",b,cpxin->real,cpxin->imag);
    delay(0);
    cpxin++;
}
getch();
cpxin = tempcpx ;
printf("\r\nlength = %d\n",length);
##### MEMORY ALLOC #####

cpxout = (COMPLEX #) calloc(length+1,sizeof(COMPLEX));
if(!cpxout){
    printf("\nNot enough memory for output data ");
    exit(1);
}
outtemp = cpxout ;

magnit = (COMPLEX #) calloc(length+1,sizeof(COMPLEX));
if(!magnit){
    printf("\nNot enough memory for magnitude data ");
    exit(1);
}
##### RUNNING DFT #####

cprintf("\n\rRUNING DFT ");
dft(cpxin,cpxout,length);
/* for(b=0 ; b (<= 24 ; b++){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf("\n dft data[%d]= %f %f ",b,cpxout->real,cpxout->imag);
cpxout++;
delay(0);
}
getch(); /*
cpxout = outtemp;
magniphase(cpxout,magnit,length);
/* for(b=0 ; b <= 24 ; b++){
printf("\n magnitude of data[%3d]= %f ",b,magnit->real);
magnit++;
delay(100);
}
getch(); /*
/***** RETAINING 32 VALUE *****/
for(b=0 ; b <= 31 ; b++){
nfd_dat1[b] = magnit->real ;
magnit++;
}
cpxin = tempcpx;
cpxout = outtemp;
free((char *)cpxin);
free((char *)cpxout);
free((char *)magnit);
}
}

```



กิติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ เสร็จสมบูรณ์ลงได้ ก็ด้วยความเอาใจใส่ดูแลและการประสิทธิ
 ประสาทวิชาความรู้ ของอาจารย์โยธิน เปรมปราณีรัตน์ , อาจารย์ประจำภาควิชา
 ระบบควบคุมทุกท่าน นอกจากนี้ยังได้รับคำแนะนำจากอาจารย์ พุศัคดี ชิวะสุวิทย์ และ
 อาจารย์ วัลลภ สุระกำพลธร ซึ่งต้องขอกราบขอพระคุณอาจารย์ทุกท่านมา ณ ที่นี้

ขอขอบคุณ บริษัท เวฟเกต จำกัด สำหรับการเอื้อเฟื้ออุปกรณ์การทดลองบางส่วน
 ขอขอบคุณในความสนใจใยดีของเพื่อนๆในภาคทุกท่าน ขอขอบคุณน้องดาที่มาช่วยพิมพ์รายงาน
 และมาคอยให้กำลังใจอย่างดี

สุดท้ายนี้ต้องกราบขออภัย หากมีสิ่งใดขาดตกบกพร่องไปบ้าง ซึ่งจะได้พยายามทำ
 การแก้ไขให้ดีขึ้น ในภายภาคหน้า



หนังสืออ้างอิง

- จรูญศักดิ์ ว่างวิวัฒน์. " MACHINE INTELLIGENCE. " ปรินญาณินพนธ์ปรินญาณินทิต
สาขาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์สถาบันเทคโนโลยีพระจอมเกล้า
เจ้าคุณทหาร ลาดกระบัง , 2531.
- สุนทรชัย โชติพิทักษ์กุล, สุนันต์ เอี่ยมวิวัฒน์ และ อนุวัฒน์ พงษ์นิจารณ์. " แผงวงจรควบคุม
คุณภาพ." ปรินญาณินพนธ์ปรินญาณินทิต สาขาวิศวกรรมโทรคมนาคม คณะ
วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง ,
2531.
- Ciarcia, Steve. " Build a Gray-Scale Video Digitizer , Part 1 :
Display / Receiver . " *BYTE* (May , 1987) : pp.95 - 106 .
- Ciarcia, Steve . " Build a Gray-Scale Video Digitizer , Part 2 :
Digitizer/Transmitter. " *BYTE* (June, 1987) : pp.129 - 138 .
- Persoon, Eric and Fu, King - Sun. " Shape Discrimination Using
Fourier Descriptors. " *IEEE Transactions on Pattern
Analysis and Machine Intelligence* , Vol. PAMI-8 , No. 3
(May , 1986) : pp.388 - 397.
- Embree, Paul M. and Kimble, Bruce . *C Language Algorithms
for Digital Signal Processing*. Englewood Cliffs,
New Jersey : Prentice-Hall , Inc. , 1991.
- Fairhurst, Michael C. *COMPUTER VISION FOR ROBOTIC SYSTEMS An
introduction*. London, Prentice-Hall International(UK) Ltd.,
1988.
- Gonzalez, Rafael C. *DIGITAL IMAGE PROCESSING*. 2nd ed.
Menlo Park , California : Addison - Wesley Publishing
Company , Inc. , 1987.