สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การตรวจจับวัตถุด้วยกระบวนการทางภาพ

Object Detection by using image processing

T055496

โดย

นายนันทภัส อัมพวานนท์

นางสาวนันทิยา ดีสุ่ย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เลขหมู่ 5549G
เลขทะเบียน
วัน,เดือน,ปี 10 พ.ค. 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจจับวัตถุด้วยกระบวนการทางภาพ

Object Detection by using image processing

โดย

นายนันทภัส อัมพวานนท์ เลขประจำตัวนักศึกษา 43010213

นางสาวนันทิยา ดีสุ่ย เลขประจำตัวนักศึกษา 43010215

อาจารย์ที่ปรึกษา

อาจารย์เทอดศักดิ์ ถิ่วหาทอง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2546

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การตรวจจับวัตถุด้วยกระบวนการทางภาพ

ผู้จัดทำ 1. นายนันทภัส อัมพวานนท์ เลขประจำตัวนักศึกษา 43010213

2. นางสาวนันทิยา ดีสุ่ย เลขประจำตัวนักศึกษา 43010215

(.....)

(อาจารย์เทอดศักดิ์ ลิ่วหาทอง)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจจับวัตถุด้วยกระบวนการทางภาพ

Object Detection by using image processing

1. นายนันทภัส อัมพวานนท์ เลขประจำตัวนักศึกษา 43010213

2. นางสาวนันทิยา ดีสุ่ย เลขประจำตัวนักศึกษา 43010215

โครงงานนี้ได้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

(.....)

(อาจารย์เทอดศักดิ์ ลิ่วหาทอง)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจจับวัตถุด้วยกระบวนการทางภาพ

นันทภัส อัมพวานนท์
นันทิยา ดีสุ่ย
อ.เทอดศักดิ์ ลิ่วหาทอง อาจารย์ที่ปรึกษา
ปีการศึกษา 2546

บทคัดย่อ

ปริญญานิพนธ์นี้เป็นการนำเสนอการตรวจจับและติดตามวัตถุจากสัญญาณภาพ โดยรับข้อมูลภาพจากกล้อง WEB Camera ที่ติดต่อกับคอมพิวเตอร์ส่วนบุคคลผ่าน USB Port และนำข้อมูลนั้นมาประมวลผลในการปรับปรุงคุณภาพของภาพ การแยกภาพวัตถุ จดจำภาพวัตถุที่ต้องการ และตรวจจับภาพวัตถุที่ได้จดจำไว้ โดยสามารถตรวจจับได้แม้ว่าวัตถุจะเปลี่ยนตำแหน่ง หมุนเปลี่ยนมุม หรือเปลี่ยนระยะทางที่ห่างจากกล้อง และแสดงค่าตำแหน่ง และมุมที่หมุนไปของวัตถุนั้นได้ ภาษาที่ใช้คือ Visual Basic 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Object Detection by using image processing

Mr.Nuntaput Ampawanon

Miss Nantiya Deesui

Mr. Thurdsak Leawhatong

2nd Semester, Educational Year 2003

Abstract

This thesis present Object Detection by using image processing .We can get the picture with WEB Camera that connects to computer through USB Port. And use this picture input in image processing contained image enhancement, segmentation and pattern recognition. Using this process we are able to detect rotated-object in variable distance from the camera. All of this developed by Visual Basic 6 programming language.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ

Abstract

สารบัญ

สารบัญรูปภาพ

บทที่ 1 บทนำ 1

1.1 ระบบการตรวจจับวัตถุด้วยกระบวนการทางภาพ 1

1.2 ขอบเขตของปริญญานิพนธ์ 2

1.3 โครงสร้างของปริญญานิพนธ์ 2

บทที่ 2 ทฤษฎีเบื้องต้น 3

2.1 การประมวลผลภาพเชิงตัวเลข (Digital image processing) 3

2.1.1 ลักษณะของแฟ้มการเก็บภาพให้อยู่ในระบบข้อมูลแบบ Digital 3

2.1.2 วิธีการสปาเชียลโดเมน 4

2.1.3 การแทนข้อมูลภาพด้วยข้อมูลแบบดิจิตอล 7

2.1.4 ลักษณะการจัดเก็บข้อมูลภาพแบบดิจิตอล 8

2.2 สัญญาณข้อมูลภาพจากดิจิตอลวิดีโอ 9

- Pixel data 9

2.3 การทำเทรชโชลด์ (Thersholding technique) 10

- การทำเทรชโฮลด์แบบครอบคลุม (Global Thresholding) 11

2.4 การสร้างภาพไบนารี 11

2.5 เทคนิคการติดตามรอยขอบภาพ (Contour Following) 13

2.5.2 การหาขอบภาพ (Edge Detection) 14

- การสร้างเกรเดียนท์มุมฉาก (Orthogonal Gradient Generation) 14

- โรเบิร์ตดิฟเฟอเรนซ์โอเปอเรเตอร์ 16

- พรีวิทท์โอเปอเรเตอร์ (Prewitt operator) 16

- โซเบลโอเปอเรเตอร์ (Sobel operator) 17

2.6 การติดต่อกับกล้อง 19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
บทที่ 3 การออกแบบการตรวจจับวัตถุด้วยกระบวนการทางภาพ	20
3.1 การออกแบบโปรแกรมควบคุมการตรวจจับวัตถุด้วย Visual Basic	20
- ขั้นตอนการออกแบบทดลอง	20
- Project Flowchart	21
- Smoothing	25
- Edge Detection	26
- Thresholding	26
- Find Contour	27
- Finding Contour Flowchart	28
- Pattern Recognition	30
- ขั้นตอนการทำงานของโปรแกรมในส่วนของการจดจำวัตถุ	31
- การหมุนภาพ (Rotation)	33
- การ FILTER ภาพที่ได้ เพื่อแก้ปัญหาการบิดเศษ	35
- การ Resize	38
บทที่ 4 ผลการทดลอง	39
- ผลการทดลองของการนำภาพที่พิจารณามากระทำการหาขอบ	39
- ผลการทดลองของการทำตามกระบวนการหา Contour	51
- ผลการทดลองจาการทำการจดจำวัตถุโดยขั้นตอนการ Rotation	56
- ผลการทดลองจาการทำการจดจำวัตถุโดยขั้นตอนการ Resize	61
- สรุปขั้นตอนการทำงานของการจดจำวัตถุ	64
บทที่ 5 สรุปและวิจารณ์ผลการทดลอง	71
กิตติกรรมประกาศ	73
หนังสืออ้างอิง	74
ภาคผนวก	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

		หน้า
รูปที่ 2.1	เมตริกซ์ของรูปขนาด 256 x 256	4
รูปที่ 2.2	แสดงลักษณะของฟังก์ชันการแปลค่าระดับสีเทาโดยกระทำกบจุด ๆ เดียวในหนึ่งครั้ง	5
รูปที่ 2.3	แสดงหน้ากากคอนโวลูชั่นขนาด 3 x 3	5
รูปที่ 2.4	แสดงจุดข้างเคียงต่าง ๆ ในบริเวณเมตริกขนาด 3 x 3 ของภาพต้นแบบ	5
รูปที่ 2.5	แสดงการเก็บข้อมูลของแต่ละพิกเซล	9
รูปที่ 2.6	ภาพไบนารีและข้อมูลของแต่ละพิกเซล	12
รูปที่ 2.7	แสดงการกระจายของทิศทางของภาพ (compass direction)	13
รูปที่ 2.8	แสดงลักษณะการเดินตามรอยขอบภาพ	14
รูปที่ 2.9	แสดงการหาเกรเดียนท์มุมฉาก	15
รูปที่ 2.10	ลักษณะของเทมเพลทที่มีขนาด 3 x 3	16
รูปที่ 2.11	แสดงอิมพัลส์เรสปอนส์ของเกรเดียน์โอเปอเรเตอร์ในแนวมุมฉาก	18
รูปที่ 2.12	รูปแสดงการติดต่อกับกล้อง	19
รูปที่ 3.1	ภาพสีจากกล้อง	22
รูปที่ 3.2	ภาพ Grey scale	22
รูปที่ 3.3	ภาพที่ผ่านการ Smooth แล้ว	23
รูปที่ 3.4	เป็นการแสดงขอบภาพของวัตถุ	23
รูปที่ 3.5	แสดงภาพ Grey scale ที่มีขอบคมชัดและมี Noise ลดลง	24
รูปที่ 3.6	แสดงภาพที่ผ่านการ Thresholdingแล้วภาพที่ได้จะเป็นภาพ Binary	24
รูปที่ 3.7	แสดงตำแหน่งของวัตถุที่ต้องการ	25
รูปที่ 3.8	แสดง mask ของ lowpass filter	25
รูปที่ 3.9	แสดง mask ของ Light Edge	26
รูปที่ 3.10	แสดงการกระจายของทิศทางของภาพ (compass direction)	27
รูปที่ 3.11	แสดงการทำ Find Contour	29
รูปที่ 3.12	แสดงการหาตำแหน่งของวัตถุ	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 3.13 แสดงผลค่า Chain Code 30
รูปที่ 3.14 แสดงการแบ่งวัตถุให้อยู่ในแกน x,y 34
รูปที่ 3.15 ภาพแสดงการเกิด noise เมื่อทำการ Rotation 35
รูปที่ 3.16 ภาพแสดงการทำ Filter ในกรณีที่จุดข้างเคียงเป็น 1 มากกว่า 3 จุด 36
รูปที่ 3.17 ภาพแสดงการทำ Filter ในกรณีที่จุดข้างเคียงเป็น 1 มากกว่า 4 จุด 36
รูปที่ 3.18 ภาพแสดง การResampling ภาพขนาด 11 X 11 -> 7 X 7 37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ระบบการตรวจจับวัตถุด้วยกระบวนการทางภาพ

ในปัจจุบันนี้เทคโนโลยีได้มีความก้าวหน้าไปอย่างรวดเร็ว และมนุษย์ซึ่งเป็นผู้นำเทคโนโลยีพวกนั้นมาใช้นั้น จึงจำเป็นต้องก้าวไกลให้ทันโลกสมัยปัจจุบันด้วย ในโครงงานนี้ได้นำเสนอการจดจำวัตถุด้วยกระบวนการทางภาพ ซึ่งเป็นการกระทำการจดจำภาพวัตถุตัวอย่างไว้ แล้วนำภาพวัตถุที่จับได้ใหม่มาทำการเปรียบเทียบกับภาพตัวอย่างที่ได้จดจำไว้ ว่ามีเปอร์เซ็นต์ความเหมือนกันอยู่ในระดับใด

ในการจดจำวัตถุนั้นมีส่วนที่เกี่ยวข้องอยู่ 2 ส่วนหลัก ๆ คือ กล้องดิจิตอลที่ใช้ในการจับภาพวัตถุและส่วนของโปรแกรมที่นำมาใช้ในการทำงาน ซึ่งเมื่อแยกส่วนของโปรแกรมออกเป็นส่วนย่อย ๆ ที่จำเป็นจะต้องศึกษาในโครงงานนี้นั้น จึงจำเป็นที่จะต้องศึกษาเกี่ยวกับการประมวลผลภาพทาง Image Processing ในเรื่องของการปรับปรุงภาพด้วยกระบวนการ Smoothing และแปลงภาพเป็นภาพไบนารีด้วยการเทรชโฮลด์ จากนั้นเข้าสู่กระบวนการ Find contour เพื่อหาตำแหน่งของวัตถุแล้วนำไปเข้าสู่การจำจดวัตถุด้วยการเก็บข้อมูลของวัตถุนั้นโดยการ Rotation และ Resize เพื่อใช้เป็นข้อมูลในการเปรียบเทียบกับภาพวัตถุที่จับมาได้ใหม่ว่าเป็นวัตถุชิ้นเดียวกันกับที่ได้จดจำไว้รึไม่ ซึ่งทั้งหมดนี้จำเป็นต้องทำการศึกษาโปรแกรมเพื่อนำมาใช้งาน ซึ่ง Visual Basic 6 ก็เป็นโปรแกรมที่ง่ายต่อการศึกษา เพื่อนำมาใช้ในการจดจำวัตถุในกระบวนการต่าง ๆ จนสามารถทำการจดจำวัตถุนั้นได้ เมื่อสามารถทำการจดจำวัตถุได้ก็บอกได้ว่าภาพวัตถุที่จับภาพมาได้นั้นเป็นวัตถุแบบเดียวกับที่ได้จดจำข้อมูลไว้ ผลที่ได้นั้นสามารถนำไปประยุกต์ใช้งานได้ต่อไป โดยอาจทำเป็นเครื่องตรวจอาวุธจากกระเป๋า และยังสามารถประยุกต์นำไปใช้งานได้ต่อไป

รายงานฉบับนี้ได้นำเสนอการศึกษาและการออกแบบการตรวจจับวัตถุดังกล่าว โดยอาศัยความรู้ทางด้าน Image Processing นำมาประยุกต์ใช้งานในการปรับปรุงภาพ ในด้านการจดจำวัตถุและใช้ในการตรวจสอบวัตถุด้วย รวมทั้งการศึกษาในเรื่องของการใช้งาน Visual Basic 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ขอบเขตของปริญญานิพนธ์

ศึกษาการออกแบบและโปรแกรมการตรวจจับวัตถุด้วยกระบวนการทางภาพ

1.การศึกษาการใช้งานโปรแกรม Visual Basic เพื่อนำมาใช้เขียนโปรแกรมทางด้าน Image Processing ซึ่งใช้ในกระบวนการปรับปรุงภาพ และ ในการคัดเลือกวัตถุจากภาพที่จับมาได้

2.การจับภาพวัตถุที่ต้องจากภาพต้นแบบแล้วนำมาประมวลผล รวมทั้งการศึกษาเรื่อง Pattern Recognition เพื่อนำมาใช้ในการจดจำลักษณะของวัตถุ

3.นำภาพวัตถุที่จดจำไว้มาทำการ rotation และ resize เพื่อเก็บเป็นข้อมูลในการเปรียบเทียบระหว่างภาพที่จับได้จากกล้อง และข้อมูลของภาพที่ได้จดจำไว้ ว่าเป็นวัตถุชิ้นเดียวกันรึไม่

1.3 โครงสร้างของปริญญานิพนธ์

ในรายงานฉบับนี้ได้จัดทำขึ้นเพื่อนำเสนอการตรวจจับวัตถุด้วยกระบวนการทางภาพ

บทที่ 1 กล่าวถึงความเป็นมาของความสำคัญของ โครงงานนี้ ขอบเขตของการศึกษาและการออกแบบชิ้นงาน

บทที่ 2 กล่าวถึงรายละเอียดของทฤษฎีที่ใช้ใน โครงงานนี้ และหลักการที่จำเป็นต้องรู้และเข้าใจ

บทที่ 3 กล่าวถึงขั้นตอนการออกแบบการตรวจจับวัตถุด้วยกระบวนการทางภาพ

บทที่ 4 กล่าวถึงผลการทดสอบของการทำงานของโปรแกรมการตรวจจับวัตถุ และการจดจำวัตถุ

บทที่ 5 แสดงการสรุปและวิจารณ์ผลที่ได้จากการทดสอบที่เกิดขึ้นตาม บทที่ 4 และกล่าวถึงปัญหาที่เกิดขึ้นรวมทั้งข้อบกพร่องในระหว่างการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีเบื้องต้น

2.1 การประมวลผลภาพเชิงตัวเลข (Digital image processing)

การประมวลผลภาพเชิงตัวเลข หมายถึง การนำภาพที่พบทั่วไปมาประมวลผลด้วยเครื่องคอมพิวเตอร์ โดยนำภาพที่นำมาประมวลผลด้วยเครื่องคอมพิวเตอร์นี้จะถูกแทนที่ด้วยตัวเลขให้อยู่ในรูปของเมตริกซ์ แต่ภาพที่ได้โดยส่วนมากแล้วจะเป็นภาพที่ได้จากตัวรับสัญญาณ ซึ่งอยู่ในรูปของฟังก์ชัน f(x,y) ที่ต่อเนื่องในระนาบ 2 มิติ โดยจะเป็นสัดส่วนกับความสว่างหรือความเข้มของภาพที่ตำแหน่ง (x,y) ซึ่งเรียกว่าระดับสีเทา (Gray level)

จากการที่นำภาพเชิงตัวเลขไปใช้ในการประมวลผลในรูปแบบต่าง ๆ มากมายนั้น สามารถที่จะแบ่งรูปแบบของการประมวลภาพเหล่านั้นออกเป็น 2 ระดับด้วยกัน คือ การประมวลผลภาพในระดับต่ำ (Low-level Image Processing) และการประมวลผลภาพในระดับสูง (High-level Image Processing)

การประมวลผลภาพในระดับต่ำนั้นเป็นการประมวลผลเชิงตัวเลขเกือบทั้งหมดเพื่อหาตัวแปรต่าง ๆ มาอธิบายข้อมูลภาพ และมีจุดประสงค์ที่จะนำตัวแปรเหล่านี้ไปใช้ในการประมวลผลภาพในระดับสูงต่อไป แต่การประมวลผลภาพในระดับสูง คือการทำให้คอมพิวเตอร์รู้จักและเข้าใจภาพได้ ในขณะที่การประมวลผลภาพในระดับต่ำโดยทั่วไปจะประกอบด้วย การกำจัดสัญญาณรบกวน, การทำให้ภาพคมชัด, การหาขอบเขตภาพ, การแบ่งแยกวัตถุภายในภาพ เป็นต้น

ความแตกต่างที่สำคัญอีกข้อหนึ่งของการประมวลผลภาพใน 2 ระดับ คือข้อมูลที่จะนำมาใช้ประมวลภาพ ซึ่งการประมวลผลในระดับต่ำจะใช้ค่าความสว่างหรือระดับความเข้มของจุดภาพโดยตรง ส่วนการประมวลผลภาพในระดับสูงข้อมูลที่จะมาประมวลจะถูกแสดงในรูปสัญลักษณ์ โดยสัญลักษณ์เหล่านั้นจะแสดงถึงสิ่งต่าง ๆ ที่อยู่ในภาพ และใช้ตัวแปรที่ได้จากการประมวลผลในระดับต่ำมาอธิบายถึงสัญลักษณ์เหล่านั้น ซึ่งจะเห็นได้ว่าการประมวลผลภาพในระดับต่ำนี้มีความสำคัญมากสำหรับที่จะทำให้คอมพิวเตอร์สามารถเข้าใจและรู้จักภาพได้

2.1.1 ลักษณะของแฟ้มการเก็บภาพให้อยู่ในระบบข้อมูลแบบ Digital

ลักษณะของแฟ้มข้อมูลภาพจะมีการกำหนดตำแหน่งเหมือนกับเมตริกซ์โดยที่จำนวนคอลัมน์ (Column) และโรว์ (row) จะแทนจำนวนจุดภาพ (Pixel) ของภาพเช่น ภาพขนาด 256 x 256 จะหมายถึงแฟ้มข้อมูลของภาพดังกล่าวมีขนาด 256 คอลัมน์และ 256 โรว์ จะสามารถแสดงได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$A = \begin{pmatrix} f(1,1) & f(1,1) & f(1,3) & \cdots\cdots & f(1,256) \\ f(2,1) & f(2,2) & f(2,3) & \cdots\cdots & f(2,256) \\ f(3,1) & f(3,2) & f(3,3) & \cdots\cdots & f(3,256) \\ - & - & - & & - \\ f(3,1) & f(256,1) & f(256,3) & \cdots\cdots & f(256,256) \end{pmatrix}$$

รูปที่ 2.1 เมตริกซ์ของรูปขนาด 256 x 256

จารูปที่ 2.1 จะเห็นว่าแฟ้มข้อมูลภาพจะประกอบไปด้วยสมาชิกขอเมตริกซ์ ทั้งหมด i x j ตัวและที่ตำแหน่งในเมตริกซ์ต่าง ๆ ก็จะมี f(I,j) ซึ่งเป็นค่าระดับสีเทา (Gray level) ของภาพที่จุดนั้นโดยที่ค่าระดับสีเทาอาจมีได้ตั้งแต่ 2 ระดับ จนถึง 256 ระดับโดยจะเพิ่มขึ้นในอัตราที่เป็นตัวเลข 2^n โดยที่ n = 1,2,3...ที่ 256 ระดับสี ซึ่งตัวอย่างของภาพสามารถแสดงได้ในรูปที่ 2.1 ซึ่งจะเห็นว่าภาพหนึ่งภาพนั้นเกิดจากการเรียงตัวกันของจุดภาพที่มีระดับค่า สีเทาต่างๆกันซึ่งการมองภาพใดๆ ให้อยู่ในรูปของเมตริกซ์เราจะเรียกการกระทำโดยตรงกับจุดภาพนี้ว่าวิธีการสปาเชียลโดเมน

2.1.2 วิธีการสปาเชียลโดเมน

จุดหลักของวิธีการนี้อยู่ที่เราจะปรับปรุงภาพหรือกระทำโดยตรงกับจุดภาพ โดยที่ฟังก์ชั่นในการปรับปรุงภาพในวิธีนี้สามารถเขียนได้เป็นสมการดังต่อไปนี้

$$g(i,j) = T\{f(i,j)\} \qquad \text{(สมการ 2.1)}$$

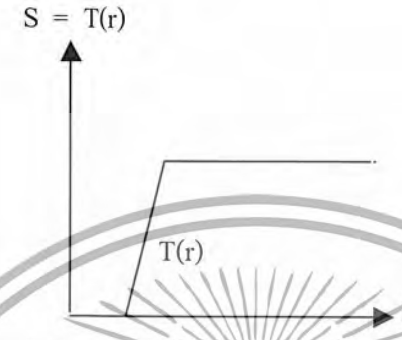
ซึ่ง g(i,j)ที่นำมาใช้ประมวลผล g(i,j) คือภาพที่ประมวลผลแล้วและ T คือ ตัวปฏิบัติการ (Operator)บน f ที่กำหนดในบริเวณตำแหน่ง (i,j) จะเห็นได้ว่าวิธีการสปาเชียลโดเมนนี้จะทำให้ง่ายในการออกแบบเขียนโปรแกรมหรือศึกษาอัลกอริทึมเนื่องจากหากเราทราบขนาดความกว้างและความยาวของภาพโดยกำหนดเป็นจำนวนจุดภาพออกมาแล้วทำการมองภาพนั้นให้เป็นเมตริกซ์ตัวหนึ่งเราก็สามารถจะใช้ตัวปฏิบัติการไปกระทำกับจุดภาพได้ตามหลักการทางคณิตศาสตร์สำหรับวิธีการสปาเชียลโดเมนนี้แบ่งเป็นได้สองกรณีดังนี้

2.1.1 ตัวปฏิบัติการกระทำต่อจุดภาพเพียงจุดเดียวต่อหนึ่งครั้ง หมายถึงว่า เมื่อตัวปฏิบัติการกระทำกับเมตริกซ์ของภาพแล้ว g(i,j) ที่ได้ออกมาจะขึ้นอยู่กับค่าของ f(i,j) ตัวเดียวเท่านั้นและตัวปฏิบัติการจะกลายเป็นฟังก์ชั่นการแปลง (Transfer Function) หรือฟังก์ชั่นการแมป (Mapping Function) ซึ่งอยู่ในรูป

$$S = T(r) \qquad \text{(สมการ 2.2)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ r และ s คือ ตัวแปรที่แทนค่าระดับสีเทาของ f(i,j) และ g(i,j) ที่จุด i,j เช่น ตัวอย่างในรูปที่ 2.3 เมื่อเราใช้ฟังก์ชั่นการแมปเป็นดังรูปจะทำให้ค่าระดับสีเทาในช่วงที่ค่าระดับสีเทาสูงเกินกว่าระดับ B และสูงกว่าถูกบีบให้อยู่ในช่วงแคบ ๆ ของ s และจะมีการกระจายค่าระดับสีเทาในช่วงที่ค่าระดับสีเทาสูงกว่าระดับ B วิธีการนี้เป็นการปรับปรุงระดับค่าสีเทาโดยรวมของภาพเพื่อให้สามารถมองเห็นภาพได้ดีขึ้น



รูปที่ 2.2 แสดงลักษณะของฟังก์ชันการแปลค่าระดับสีเทาโดยกระทำกับจุด ๆ เดียวในหนึ่งครั้ง

2.1.2 ตัวปฏิบัติการกระทำกับจุดภาพที่อยู่บริเวณรอบ ๆ กับจุดที่กำลังกระทำอยู่ หมายถึง เมื่อเราต้องการจะให้ตัวปฏิบัติการกระทำกับจุดใดๆ แล้วนั้นตัวปฏิบัติการจะต้องนำเอาค่าข้างเคียงของจุดนั้นมาคำนวณด้วยเสมอเราเรียกการทำงานแบบนี้ว่าการทำงานที่ต้องใช้จุดข้างเคียง (Neighborhood operation) ซึ่งการปฏิบัติการอย่างนี้เรามักจะใช้วิธีคอนโวลูชั่น (Convolution) โดยที่การทำคอนโวลูชั่นนี้จะเป็นการคำนวณผลบวกของผลคูณ (Sum of product) รูปที่ 2.3 แสดงหน้ากากของคอนโวลูชั่น

$$M = \begin{pmatrix} m(-1,-1) & m(0,-1) & m(1,-1) \\ m(-1,0) & m(0,0) & m(1,0) \\ m(-1,1) & m(0,1) & m(1,1) \end{pmatrix}$$

รูปที่ 2.3 แสดงหน้ากากคอนโวลูชั่นขนาด 3 x 3

$$F = \begin{pmatrix} F(I-1,J-1) & F(I,J-1) & F(I,J-1) \\ F(I-1,J) & F(I,J) & F(I+1,J) \\ F(I-1,J+1) & F(I,J+1) & F(I+1,J+1) \end{pmatrix}$$

รูปที่ 2.4 แสดงจุดข้างเคียงต่าง ๆ ในบริเวณเมตริกขนาด 3 x 3 ของภาพต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวชี้ในหน้ากากจะมีหน้าจุดเริ่มต้นที่จุดศูนย์กลางโดยเริ่มที่มุมบนด้านซ้ายของภาพ สามารถเขียนสมการการคอนโวลูชั่นได้ดังนี้

$$C(I,J) = \sum_{k=1}^{i+1}\sum_{l=1}^{j+1} F(k,1)M(I-K,K-1) \qquad \text{(สมการ 2.3)}$$

แต่ละจุดภาพจะเป็นการวนเอามาจากจุดรอบข้างและคูณด้วยค่าที่ตรงตำแหน่งเดียวกันกับหน้ากากนำผลคูณของแต่ละตัวนั้นรวมกันก็จะได้ค่าจุดภาพผลลัพธ์ซึ่งการคำนวณที่ทำนั้นสามารถกระจายแสดงออกมาได้ดังนี้

$$C(i,j)=F(i-1,j-1)M(1,1)+F(i,j-1)M(0,1)+F(i+1,j+1)M(-1,1)+F(i-1,j)M(1,0)+$$

$$F(i,j)M(0,0)+F(I+1,j)\,M(-1,0)+F(i-1,j+1)M(1,-1)+F(i,j+1)M(0,-1)F(I+1,j+1)M(-1,1) \qquad \text{(สมการ2.4)}$$

จากสมการเราพบว่ามีการจัดเรียงเทอมต่าง ๆ โดยลำดับในหน้ากากมีการจัดเรียงจากซ้ายลงไปยังล่างขวาและตำแหน่งที่ตรงกันในหน้ากากจะมีลำดับตรงข้ามกันคือเริ่มจากล่างขวาขึ้นไปถึงมุมบนซ้ายสำหรับขนาดของหน้ากากนั้นจะเป็นเท่าไรก็ได้และไม่จำเป็นต้องมีสมาชิกแนวตั้งเท่ากับแนวนอนเสมอไปแต่โดยทั่วไปแล้วการประยุกต์ที่ใช้กันอยู่จะใช้หน้ากากที่มีจำนวนสมาชิกแนวตั้งเท่ากับสมาชิกแนวนอนมีจำนวนคงที่และมักเป็นจำนวนเลขคี่เพื่อ ที่จะทำให้สามารถได้จุดศูนย์กลางของหน้ากากเป็นจุดศูนย์กลางได้

ปัญหาอีกอย่างหนึ่งที่เราพบในการทำคอนโวลูชั่นภาพคือการขาดจุดข้างเคียงในบริเวณของทำคอนโวลูชั่นในบริเวณขอบรอบนอกกรอบภาพหมายถึง การที่เราจะใช้หน้ากากคอนโวลูชั่นมาทำการคอนโวลูชั่นบริเวณขอบภาพซึ่งไม่สามารถทำได้เนื่องจากการคำนวณจะขาดจุดภาพบริเวณภายนอกอาณาบริเวณของภาพ โดยทั่วไปแล้ว วิธีการแก้ปัญหานี้ก็มักจะใช้วิธีการกำหนดจุดในบริเวณกรอบภาพทั้งสี่ด้านให้เป็นค่าเดิมหรือแล้วแต่กรณี ความกว้างของแถบด้านบนและด้านล่างของภาพมีค่าเท่ากับ (Ym-1)/2 และข้างซ้ายและขวาเท่ากับ (Xm-1)/2 เมื่อ Xm คือความกว้างของหน้ากากและ Ym คือความสูงของหน้ากาก ตัวอย่างเช่น หน้ากากขนาด 3 x 3 จะมี แถบกว้าง 1 จุด ภาพซึ่งไม่สามารถคำนวณได้รอบๆ ภาพ

ในการใช้งานจะต้องรู้ว่าจะนำภาพที่ได้จากการประมวลแล้วไปทำอะไรต่อไป เพราะอาจเป็นไปได้ที่ภาพผลลัพธ์อาจมีขนาดเล็กกว่าต้นแบบ บางระบบต้องการการคำนวณที่ถูกต้องสมบูรณ์แบบอย่างแท้จริงโดยนำข้อมูลในบริเวณอื่น ๆ มาใช้แทนบริเวณโรว์และคอลัมน์ของจุดภาพที่เลยออกจากต้นแบบไปวิธีการหนึ่งนั้นทำได้โดย การใช้โรว์ทางด้านล่างของภาพมาใช้เป็นข้อมูลที่ต้องการของการคำนวณ ทางด้านบนและในทำนองเดียวกันที่ใช้โรว์ทางด้านบนมาเป็นข้อมูลที่ต้องการคำนวณของการคำนวณด้านล่างในลักษณะที่คล้ายกัน โดยนำคอลัมน์ทางขวาของภาพมาใช้เป็นข้อมูลในการคำนวณของคอลัมน์ทางด้านซ้ายและคอลัมน์ทางด้านซ้ายของภาพนำมาเป็นข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ต้องการคำนวณทางด้านขวาเช่นกัน ถึงแม้ว่าวิธีนี้จะเป็นการไม่เหมาะสมนักก็ตามแต่ผลลัพธ์ที่ได้ก็ให้ภาพผลลัพธ์มีขนาดเท่ากับต้นแบบ เนื่องจากการทำคอนโวลูชั่นนี้ค่อนข้างจะใช้การคำนวณมากครั้ง เช่น หากถ้าเป็นหน้ากากขนาด 3x3 ในการที่จะได้ผลลัพธ์ 1 จุดภาพ ต้องมีการคูณ 9 ครั้ง การบวก 9 และ การหารอีก 9 ครั้ง และหากเราใช้ขนาดของหน้ากากที่ใหญ่ขึ้นก็ย่อมต้องใช้การคำนวณมากครั้งขึ้น เราจึงต้องพยายามใช้ขนาดของหน้ากากที่เล็กที่สุดเท่าที่จะทำได้ในการคอนโวลูชั่น

2.1.3 การแทนข้อมูลภาพด้วยข้อมูลแบบดิจิตอล

ภาพข้อมูลแบบดิจิตอล (Digital Image) เป็นภาพที่ถูกดัดแปลงมาจากอนาลอก อยู่ในรูปของตัวเลข โดยภาพอนาลอกถูกแบ่งเป็นพื้นที่สี่เหลี่ยมเล็ก ๆ ที่เรียกว่า พิกเซล (pixel) ในแต่ละพิกเซล จะถูกระบุตำแหน่งโดย (x,y) และค่าระดับสีเทาของพิกเซล โดยเราสามารถแปลงภาพเป็นข้อมูลแบบดิจิตอลได้โดยมีขั้นตอนและวิธีการดังนี้

เมื่อเรานำสัญญาณอนาลอกที่ต้องการประมวลผลมาผ่านส่วนที่รียกว่าดิจิไตเซอร์ (Digitizer) ซึ่งมีหน้าที่ในการเปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิตอล อุปกรณ์ส่วนนี้ได้แก่ กล้องโทรทัศน์ดิจิไตเซอร์ จากนั้นทำการควอนไตซ์ (Quantizing) เพื่อที่จะประมวลผลสัญญาณด้วยระบบคอมพิวเตอร์ฟังก์ชันของภาพ f(x,y) จะถูกทำให้เป็นสัญญาณที่ไม่ต่อเนื่องทั้งระนาบของภาพ ซึ่งเราเรียกว่า การสุ่มภาพ (Image Sampling) ของฟังก์ชันที่ได้เรียกว่า การควอนไตเซชันระดับสีเทา (Gray level quantization) ก็จะได้ข้อมูลที่เป็นดิจิตอล

สมมุติว่าสัญญาณภาพต่อเนื่อง f(x,y) ถูกดิจิไตซ์ในระนาบ x และ y เป็นช่วง ๆ เท่า ๆ กัน เราสามารถจัด f(x,y) ให้อยู่ในรูปเมตริกซ์ ขนาด NxN ได้ดังสมการที่ 2.5

$$f(x,y) = \begin{matrix} f(0,0) & f(0,1) & f(0,2)\ldots f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2)\ldots f(1,N-1) \\ . & . & . \quad . \\ . & . & . \quad . \\ f(N-1,0) & f(N-1,1) & f(N-1,2)\ldots f(N-1,N-1) \end{matrix} \qquad \text{(สมการ 2.5)}$$

ซึ่งทางขวาของสมการ จะเรียกได้ว่า ภาพดิจิตอล และทุก ๆสมาชิกของเมตริกซ์จะเรียกว่า พิกเซล จากขบวนการสร้างภาพดิจิตอลข้างต้นจะเห็นได้ว่าเราสามารถทราบขนาดของความละเอียดของภาพ NxN พิกเซล และจำนวนของเกรย์สเกล ในทางปฏิบัติการทำควอนไตเซชันในระบบภาพดิจิตอล จะมีค่าดังสมการที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B = NxNxM บิต (สมการ 2.6)

เมื่อ B = ขนาดของข้อมูลภาพที่เป็นดิจิตอล

G = จำนวนเกรย์สเกลที่ต้องใช้ในการเก็บภาพ

M = จำนวนบิตที่ใช้ในการแทนข้อมูลภาพ 1 พิกเซล

โดย M สามารถหาได้จาก $G = 2^M$

2.1.4 ลักษณะการจัดเก็บข้อมูลภาพแบบดิจิตอล

โดยทั่วไปแล้ว ข้อมูลภาพจะมีความเข้มตั้งแต่ 2 ระดับขึ้นไป แต่ที่ใช้กันมากจะใช้กันที่ค่าระดับความเข้มของจุดภาพเท่ากับ 256 ระดับ ซึ่งจะทำให้ค่าของจุดภาพอยู่ในช่วง (0-255) โดยใช้เนื้อที่เก็บข้อมูลภาพขนาด 1 ไบต์ หรือ 8 บิต สำหรับข้อมูล 1 จุดภาพ ($2^8 = 256$) ในกรณีที่ต้องการภาพที่มีความละเอียดของระดับความเข้มสูง ๆ อาจจะต้องการจำนวนบิตสำหรับเก็บข้อมูลมากกว่า 8 บิต อาจจะเป็น 16 หรือ 24 บิต โดยค่าความเข้มของจุดภาพเท่ากับ 2^{16} และ 2^{24} โดยจะแยกให้ชัดเจนดังนี้

1. ภาพ 2 ระดับ คือมีเพียงแค่จุดขาวกับจุดดำเท่านั้น โดยแต่ละจุดภาพเป็นข้อมูลขนาด 1 บิต

2. ภาพ 16 ระดับ คือ ในแต่ละจุดภาพจะมีขนาดของข้อมูล 4 บิต ซึ่งทำให้สามารถแสดงได้ 16 ระดับสี ขึ้นอยู่กับภาพนั้นเป็นภาพสีหรือขาวดำ

3. ภาพ 256 ระดับ คือ ในแต่ละจุดภาพจะมีขนาดของข้อมูล 8 บิต ซึ่งทำให้สามารถแสดงได้ 256 ระดับสี ขึ้นอยู่กับภาพนั้นเป็นภาพสีหรือขาวดำ

4. ภาพทรูคัลเลอร์ (True color) คือ ในแต่ละจุดภาพจะมีขนาดของข้อมูล 24 บิต ทำให้สามารถแสดงผลภาพได้เหมือนจริงที่สุด เพราะสามารถแสดงสีได้ถึง 16,777,216 ระดับสี โดยจะแสดงได้แต่ภาพสีเท่านั้น ไม่สามารถแสดงภาพขาวดำได้

การแสดงภาพนี้ใช้วิธีตั้งค่าของแม่สีในตารางสี โดยอาจเลือกสีเป็นแบบ 16 สี จาก 262,144 สี หรือ 256 สี จาก 262,144 สี ขึ้นอยู่กับโหมดการแสดงผลสำหรับทรูคัลเลอร์ จะไม่มีการเลือกสีแสดงผลโดยการส่งค่าสี RGB ผ่าน D/A สีละ 8 บิต ออกไป ความแตกต่างของการแสดงภาพสีและขาวดำ คือ ภาพขาวดำจะต้องตั้งให้แม่สีทั้งสามสีมีค่าเท่ากัน เนื่องจาก VGA กำหนดให้แม่สีแต่ละสีใช้ได้เพียง 64 ระดับสีเท่านั้น หากต้องการให้เห็นทั้ง 256 ระดับ ต้องแสดงในโหมดทรูคัลเลอร์ แล้วให้ RGB มีค่าเท่ากัน ซึ่งในโหมดนี้จะสามารถใช้รีจิสเตอร์ได้ 8 บิต สำหรับแต่ละแม่สี

โดยทั่วไปวิธีการประมวลผลภาพเชิงเลขที่ทำให้คอมพิวเตอร์สามารถรู้จักวัตถุภายในภาพได้นั้นแบ่งได้ 2 ระดับด้วยกันคือ การประมวลผลภาพในระดับต่ำ (Low level Image Processing) และการประมวลผลภาพในระดับสูง (High level Image Processing) การประมวลผลภาพระดับต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเป็นการประมวลผลเชิงตัวเลขเกือบทั้งหมด เพื่อหาตัวแปรต่าง ๆ มาอธิบายข้อมูลภาพ โดยมีจุดประสงค์เพื่อนำตัวแปรเหล่านั้นไปใช้ในการประมวลผลระดับสูงต่อไป

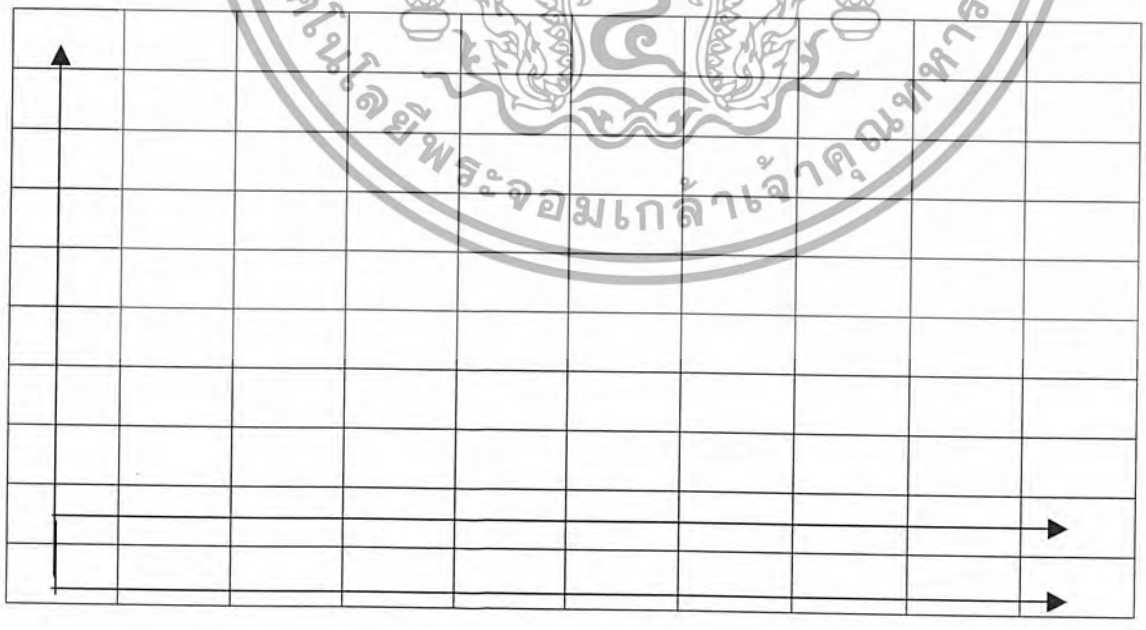
การประมวลผลภาพในระดับสูง เป็นการนำผลลัพธ์ที่ได้จากการประมวลผลระดับต่ำมาตีความ หรือเพื่อให้คอมพิวเตอร์สามารถรู้จักและเข้าใจภาพได้สำหรับความแตกต่างของการประมวลผลภาพระดับต่ำและสูง คือ ข้อมูลที่นำมาใช้ในการประมวลผล โดยที่การประมวลผลภาพระดับต่ำจะใช้ค่าความสว่างของจุดโดยตรง ส่วนการประมวลผลภาพระดับสูงนั้นข้อมูลภาพที่นำมาประมวลผลจะถูกแสดงในรูปของสัญลักษณ์ ซึ่งจะแสดงถึงสิ่งต่าง ๆ ที่อยู่ในภาพ เช่น ขนาด หรือ รูปร่างของวัตถุในภาพ

2.2 สัญญาณข้อมูลภาพจากดิจิตอลวิดีโอ

การส่งสัญญาณจากข้อมูลภาพวีดิโอ จะมีลักษณะการส่งที่เป็นลำดับภาพเดี่ยวหรือเฟรม (frame) ที่ฉายต่อเนื่องกัน เช่นระบบวีดิโอ NTSC จะส่งด้วยอัตราเร็ว 30 เฟรมต่อวินาที โดยดิจิตอลวีดิโอแต่ละเฟรมจะเป็นข้อมูลภาพดิจิตอลในลักษณะของเมตริกซ์ ซึ่งแต่ละจุดเรียกว่า พิกเซล มีค่าของระดับความเข้มสี โดยทั่วไปจะใช้เกรย์สเกลที่มีค่าตั้งแต่ 0-255 โดย 0 แทนความมืดมากที่สุด และ 255 แทนความสว่างมากที่สุด

pixel data

เป็นส่วนเก็บข้อมูลสีของแต่ละพิกเซลของภาพ โดยข้อมูลแรกจะเป็นค่าสีของพิกเซลที่อยู่แถวล่างสุดที่ตำแหน่งซ้ายสุด ข้อมูลลำดับต่อไปจะเรียงทางขวาจากแถวล่างจนถึงแถวบนสุด



รูปที่ 2.5 แสดงการเก็บข้อมูลของแต่ละพิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การทำเทรชโฮลด์ (Thersholding technique)

การทำเทรชโฮลด์ถือว่าเป็นเทคนิคที่สำคัญในการประมวลผลภาพในส่วนของการทำเซกเมนต์ภาพ ซึ่งจุดประสงค์ของการทำเซกเมนต์ภาพ คือ การแยกองค์ประกอบของภาพออกไปเป็นส่วนประกอบย่อย ๆ ที่มีความสัมพันธ์กันทางกายภาพของภาพนั้น และส่วนประกอบที่ถูกแยกออกมานั้นอาจถูกนำไปประมวลภาพในส่วนอื่นได้ต่อไป ซึ่งการทำเซกเมนต์ภาพจะมีหลักการทำงานในแนวเดียวกันกับสายตาของคน คือ สามารถแยกลักษณะเด่นออกมาจากภาพที่มองเห็นได้ และเทคนิคการทำเทรชโฮลด์ถือว่าเป็นแยกองค์ประกอบของภาพที่ง่ายเทคนิคหนึ่ง มีหลักการว่าจุดภาพที่มีคุณสมบัติอยู่ในบางช่วงใด ๆ จะถูกจัดเป็นกลุ่มได้โดยที่ระดับความเข้มหนึ่งนั้นสามารถที่จะแบ่งแยกกลุ่มของจุดภาพออกเป็น 2 กลุ่มได้อย่างชัดเจน คือกลุ่มของวัตถุ (object) ซึ่งจะมีระดับความเข้มของภาพ g(x,y) ค่อนข้างต่ำ (มืด) กับกลุ่มของส่วนที่เป็นพื้นหลัง (background) ที่จะมีระดับความเข้มของภาพ g(x,y) ค่อนข้างสูง (สว่าง) ระดับความเข้มของภาพที่ถูกแบ่งออกเป็น 256 ระดับ จะเห็นได้ว่าการที่แยกกลุ่มข้อมูลออกเป็น 2 กลุ่มอย่างชัดเจนย่อมสามารถทำได้โดยการเลือกค่าเทรชโฮลด์ที่มีค่าความเข้มอยู่ระหว่างกลุ่มทั้งสองบนฮิสโตแกรมระดับความเข้มของภาพ แล้วทำการตรวจสอบแต่ละจุดภาพว่าถ้ามีค่า g(x,y) น้อยกว่าค่าเทรชโฮลด์ถือว่าเป็นจุดภาพของวัตถุที่แสดงได้ด้วยจุดดำ แต่หากว่าจุด g(x,y) นั้นมีค่ามากกว่าหรือเท่ากับค่าเทรชโฮลด์ก็ถือว่าเป็นจุดภาพในส่วนพื้นหลังที่แสดงได้ด้วยจุดขาว ดังนั้นข้อมูลภาพ $g_{thr}(x,y)$ ที่ผ่านการทำเทรชโฮลด์สามารถนิยามได้ดังนี้

$$G_{thr}(x,y) = \begin{cases} 0 \text{ if } g(x,y) < T \\ 1 \text{ if } g(x,y) \geq T \end{cases}$$

โดยที่ $g_{thr}(x,y)$ คือ ข้อมูลภาพผลลัพธ์เป็นไบนารี

g(x,y) คือ ข้อมูลภาพอินพุทที่มีระดับความเข้ม 0 ถึง L ระดับ

T คือ ค่าเทรชโฮลด์ เป็นค่าคงที่ที่มีค่าระหว่าง 0 ถึง L

0 คือ จุดดำ (ส่วนที่เป็นวัตถุ)

1 คือ จุดขาว (ส่วนที่เป็นพื้นหลัง)

จะเห็นได้ว่า การทำเซกเมนต์ภาพโดยใช้เทคนิคการทำเทรชโฮลด์เพื่อให้ได้ภาพผลลัพธ์ที่มีความถูกต้องเหมาะสมนั้นสิ่งที่สำคัญที่สุด คือ ค่าเทรชโฮลด์ที่ใช้นั่นเอง เนื่องจากถ้าเลือกค่าเทรชโฮลด์ที่ไม่เหมาะสมแล้วภาพผลลัพธ์ที่ได้อาจไม่ถูกต้อง ดังนั้นปัญหาของการทำเซกเมนต์ภาพโดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการทำเทรชโฮลด์นี้ก็คือทำอย่างไรจึงจะสามารถคำนวณหาค่าเทรชโฮลด์ที่เหมาะสมสำหรับภาพแต่ละภาพที่นำมาทำการเซกเมนต์ได้ ซึ่งได้มีผู้เสนอวิธีการในการคำนวณหาค่าเทรชโฮลด์ไว้หลายวิธี ซึ่งแต่ละวิธีย่อมมีความเหมาะสมกับภาพที่แตกต่างกันไป

การทำเทรชโฮลด์แบบครอบคลุม (Global Thresholding)

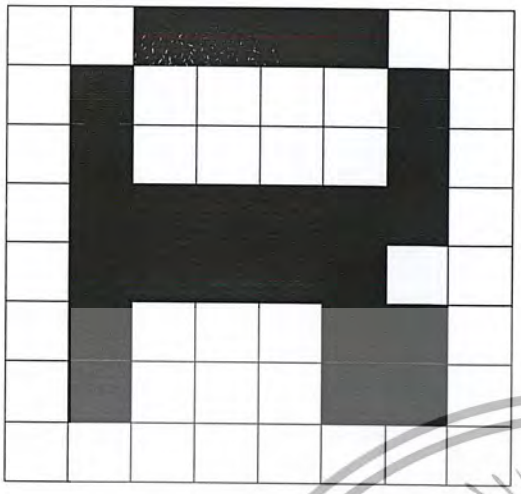
สำหรับขั้นตอนการหาค่าเทรชโฮลด์ที่ครอบคลุมตลอดทั้งภาพโดยอัตโนมัติ ปกติจะมีพื้นฐานของการดำเนินการบนฮิสโตแกรมของระดับความเข้มของจุดภาพ ซึ่งฮิสโตแกรมระดับความเข้มนี้สามารถสร้างได้จากการนับจำนวนของจุดภาพที่มีระดับความเข้มเท่ากับค่าความเข้มที่จุดนั้นทั้งหมดทั่วภาพนั่นเอง จากนั้นจึงทำการหาค่าเทรชโฮลด์ในรูปแบบต่าง ๆ ที่สามารถแบ่งฮิสโตแกรมนี้ออกเป็น 2 ส่วน (ส่วนที่เป็นระดับความเข้มของวัตถุ กับส่วนที่เป็นระดับความเข้มของพื้นหลัง) ได้อย่างถูกต้องตรงตามความต้องการ โดยเฉพาะอย่างยิ่ง ในกรณีของภาพที่มีอัตราความแตกต่างของระดับความเข้มระหว่างส่วนที่เป็นวัตถุกับส่วนที่เป็นพื้นหลังมีค่าสูง (แตกต่างกันมาก) และยังมีระดับความเข้มที่เกิดขึ้นในแต่ละส่วนมีความสม่ำเสมอ ย่อมเหมาะสมที่จะใช้ระดับความเข้มที่มีจำนวนของจุดภาพที่ต่ำสุดซึ่งอยู่ระหว่างกลุ่มระดับความเข้มที่มีค่าสูงสุด (peak) ทั้งสองบนฮิสโตแกรมเป็น " ค่าเทรชโฮลด์ "หรือในกรณีทั่ว ๆ ไป ค่าเทรชโฮลด์อาจจะพิจารณาจากค่าระดับความเข้มที่สามารถแบ่งฮิสโตแกรมออกเป็น 2 กลุ่ม แล้วทำให้ความแปรปรวนที่เกิดขึ้นระหว่างกลุ่มมีค่ามากที่สุด แต่ความแปรปรวนที่เกิดขึ้นระหว่างกลุ่มมีค่าต่ำที่สุด หลังจากนั้นนำค่าเทรชโฮลด์ที่คำนวณหาได้ไปทำเทรชโฮลด์กับแต่ละภาพทั่วทั้งภาพเพื่อให้ได้ภาพผลลัพธ์ที่เป็นไบนารีในที่สุด

2.4 การสร้างภาพไบนารี

การสร้างภาพไบนารี หมายถึงการแปลงข้อมูลที่มีความเข้มหลายระดับ ให้เป็นภาพที่มีระดับความเข้มเพียง 2 ระดับ คือ 1 จุดภาพ มีค่าได้แค่ 2 ค่าเท่านั้น โดยเป็น 0 กับ 1 ซึ่ง 1 จะหมายถึงจุดภาพที่มีสีดำ และ 0 จะหมายถึงจุดภาพที่มีสีขาว การแปลงข้อมูลภาพหลายระดับไปเป็นภาพไบนารีจึงมีความจำเป็นและมีประโยชน์มากในการแสดงผลภาพที่มีระดับความเข้มของภาพหลายระดับบนอุปกรณ์ที่มีความสามารถในการแสดงผลได้ 2 ระดับ ประโยชน์อีกประการคือ การลดเนื้อที่การเก็บข้อมูลภาพให้เหลือเพียง 8 บิต

ในการสร้างภาพไบนารี สามารถทำได้โดยใช้เทคนิคการทำเทรชโฮลด์ (Thresholding Technics) โดยพิจารณาว่าจุดใดควรจะเป็นจุดขาวหรือจุดดำ จะกระทำโดยการเปรียบเทียบระหว่างจุดภาพเริ่มต้นกับค่าคงที่ค่าหนึ่งเรียกว่า ค่าเทรชโฮลด์ เทคนิคนี้ใช้กันมากในกรณีที่ข้อมูลภาพมีลักษณะแตกต่างกันระหว่างวัตถุ และพื้นหลัง โดยค่าของจุดภาพใด ๆ ที่มีค่าน้อยกว่าค่าเทรชโฮลด์จะถูกกำหนดให้เป็นค่า 1 (จุดสีดำ) และถ้าค่าของจุดภาพมีค่ามากกว่าค่าเทรชโฮลด์จะถูกเปลี่ยนให้เป็นค่า 0 (จุดสีขาว) ซึ่งการทำงานสามารถแสดงได้ดังสมการ 2.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



0	0	1	1	1	1	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	0	0
0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	0
0	0	0	0	0	0	0	0

รูปที่ 2.6 ภาพไบนารีและข้อมูลของแต่ละพิกเซล

$$b(x,y) = \begin{cases} 1; & g(x,y) < Thr \\ 0; & g(x,y) \geq Thr \end{cases} \qquad \text{(สมการ 2.7)}$$

b(x,y) : ข้อมูลภาพผลลัพธ์เป็นภาพไบนารี

g(x,y) : ข้อมูลภาพอินพุทที่มีระดับความเข้ม 0 ถึง L ระดับ

Thr : ค่าเทรชโฮลด์เป็นค่าคงที่ระหว่าง 0 ถึง L ระดับ

1 : จุดดำ

0 : จุดขาว

โดยที่ L คือระดับความเข้มของจุดภาพสูงสุด

ในการสร้างภาพไบนารีโดยใช้เทคนิคเทรชโฮลด์เพื่อให้ได้ผลลัพธ์ที่คมชัดและเหมาะสม สิ่งที่สำคัญที่สุดคือ ค่าเทรชโฮลด์ เนื่องจากถ้าเลือกใช้ค่าเทรชโฮลด์ที่ไม่เหมาะสมภาพที่ได้จะไม่คมชัด และรายละเอียดบางส่วนจะขาดหายไป ดังนั้นปัญหาการสร้างภาพด้วยวิธีเทรชโฮลด์คือ ทำอย่างไรจึงสามารถคำนวณหาค่าเทรชโฮลด์ที่เหมาะสมได้

2.5 เทคนิคการติดตามรอยขอบภาพ (Contour Following)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคนิคการติดตามรอยขอบภาพถูกนำมาใช้ในการแยกและคัดลอกส่วนของรูปภาพใด ๆที่อยู่บนรูปใหญ่ ข้อมูลภาพที่จะนำมาประกอบด้วยผลด้วยเทคนิคนี้ต้องอยู่ในรูปของไบนารี นั่นคือจุดภาพจะแสดงด้วยตัวเลข 0 กับ 1 เท่านั้น

การทำงานของเทคนิคการติดตามรอยขอบภาพเป็นการเดินไต่ไปบนขอบระหว่างส่วนที่เป็นรูปภาพ กับส่วนที่เป็นพื้นหลัง โดยจะตรวจกวาดไปทุก ๆ พิกเซล โดยจะเริ่มจากจุดมุมซ้ายของภาพ ตรวจกวาดไปในทิศทางจากซ้ายไปขวา และเลื่อนจากบนลงล่าง เมื่อตรวจกวาดมาพบจุดใด ๆ ที่มีค่าของจุดภาพเป็น 1 ก็จะเปลี่ยนการเคลื่อนที่ไปยังจุดถัดไปใหม่โดยมีเงื่อนไขการเคลื่อนที่ ดังนี้

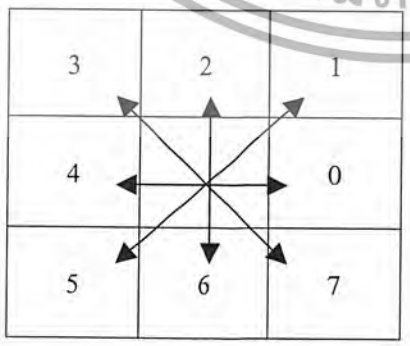
1. ถ้าจุดที่อยู่ปัจจุบันเป็นจุดของขอบภาพหรือมีค่าของจุดเป็น 1 ให้เลี้ยวซ้ายแล้วก้าวเดินตรงไปข้างหน้ายังจุดถัดไป

2. ถ้าจุดที่อยู่ในปัจจุบันเป็นพื้นเบื้องหลัง หรือมีค่าของจุดเป็น 0 ให้เลี้ยวขวา แล้วก้าวเดินตรงไปข้างหน้ายังจุดถัดไป

3. การเคลื่อนที่จะสิ้นสุดลงเมื่อจุดที่อยู่ปัจจุบันเป็นจุดเดียวกันกับจุดเริ่มต้น

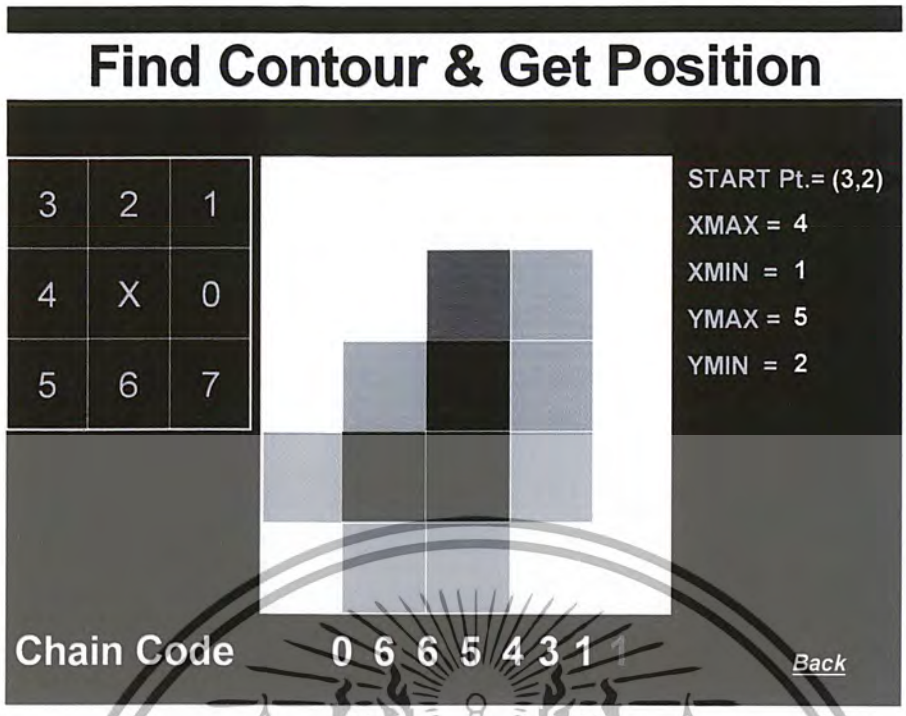
ในรูป 2.7 ได้แสดงลักษณะการทำงานของเทคนิคการติดตามรอยขอบภาพซึ่งจะแสดงการเคลื่อนที่ไปตามจุดต่าง ๆ ที่เป็นขอบภาพ เริ่มจากจุดที่ถูกแรเงาไว้ซึ่งเป็นจุดแรกที่ตรวจกวาดตามเงื่อนไขที่กำหนด เมื่อการเคลื่อนที่วนกลับมาถึงจุดเริ่มต้นก็จะทราบถึงจุดที่เป็นขอบของภาพทั้งหมด ขณะที่เดินไต่ขอบนั้นก็จดจำค่าพิกัดไปด้วย เมื่อเดินถึงจุดเดิมก็นำค่าพิกัดที่น้อยที่สุดและมากที่สุดมากำหนดขอบเขตของขนาดวัตถุเพื่อใช้เปรียบเทียบกันต่อไป

ขบวนการหรือขั้นตอนที่สำคัญอย่างหนึ่งของการประมวลผลภาพคือการหาขอบภาพเพื่อนำมาปรับปรุงภาพ ซึ่งคำว่าขอบในที่นี้จะหมายถึงจุดที่มีระดับความเข้ม (gray-level) เปลี่ยนแปลงอย่างเห็นได้ชัด และจุดที่มีการเปลี่ยนแปลงนี้จะเป็นกรอบของรายละเอียดในส่วนต่าง ๆ ของภาพซึ่งขอบที่กล่าวมานี้สามารถเกิดขึ้นได้ทุกทิศทางและอาจมีความเข้มไม่เท่ากันก็ได้ส่วนที่เป็นขอบนี้จะแสดงสเปกตรัมที่ความถี่สูงเท่านั้น



รูป 2.7 แสดงการกระจายของทิศทางของภาพ (compass direction)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แสดงลักษณะการเดินตามรอยขอบภาพ

2.5.1 การหาขอบภาพ (Edge Detection)

วิธีการเบื้องต้นที่ใช้สำหรับการหาขอบภาพในลำดับแรก (First order Derivative Edge Detection) มีอยู่ 2 วิธีด้วยกัน วิธีแรกคือการหาเกรเดียนท์ (gradient) ในทิศทางมุมฉาก คือตามแนวนอนและแนวตั้ง และอีกวิธีหนึ่งคือเกรเดียนท์ในหลายทิศทาง

การสร้างเกรเดียนท์มุมฉาก (Orthogonal Gradient Generation)

ถ้ากำหนดให้ G(x,y) คือเกรเดียนท์ใน 1 มิติ และ θ คือมุมที่พิจารณาโดยเทียบกันกับแนวระดับ ดังนั้นจะทำการคำนวณหาเกรเดียนท์สำหรับขอบภาพจาก

$$G(x,y) = \frac{\partial F(x,y)}{\partial x} \cos\theta + \frac{\partial F(x,y)}{\partial y} \sin\theta \qquad \text{(สมการ 2.8)}$$

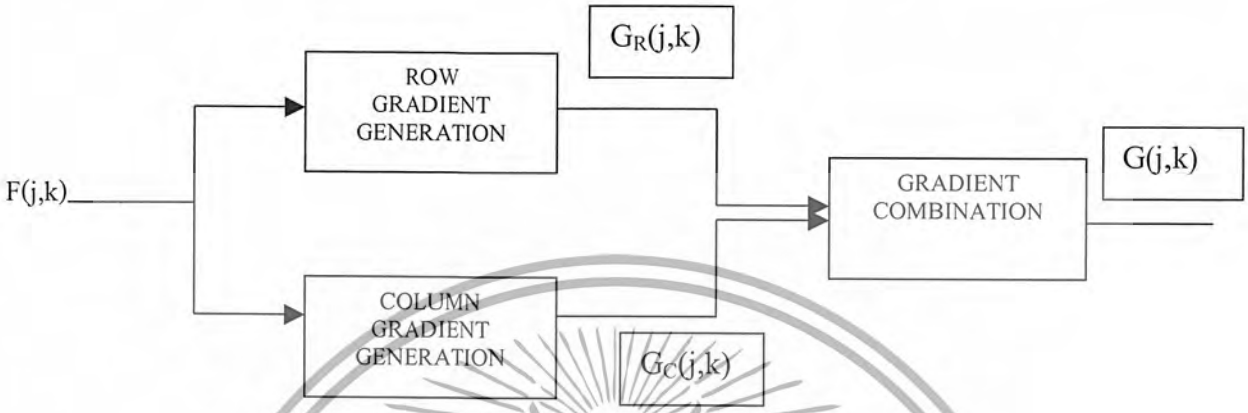
ในรูป 2.12 แสดงการสร้างเกรเดียนท์ของขอบภาพ[G(j,k)] ในขอบเขตที่ไม่ต่อเนื่อง โดยจะแยกเป็นเกรเดียนท์ของแถว (row edge gradient) [G(j,k)] และเกรเดียนท์ของหลัก (column edge gradient) [G(j,k)] และค่าแอมปลิจูดของเกรเดียนท์ซึ่งหมายความถึงความเข้มของภาพแสดงได้ดังนี้

$$[\,G(j,k)\,] = \{\,[\,Gr(j,k)\,] + [\,Gc(j,k)\,]\,\}^{1/2} \qquad \text{(สมการ 2.9)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณค่าเกรเดียนท์บางครั้งอาจประมาณค่าจาก

$$G(j,k) = \left| Gr(j,k) \right| + \left| Gc(j,k) \right| \qquad \text{(สมการ 2.10)}$$



รูป 2.9 แสดงการหาเกรเดียนท์มุมฉาก

และทิศทางของเกรเดียนท์ตามแนวนอนคือ

$$\theta(j,k) = \tan^{-1}\left\{\frac{Gc(j,k)}{Gr(j,k)}\right\} \qquad \text{(สมการ 2.11)}$$

ซึ่งมีวิธีการที่ง่ายที่สุดสำหรับการสร้างเกรเดียนท์คือ การหาค่าความแตกต่างของความเข้มในแต่ละจุดตามแนวแกนนอน (row) และตามแนวแกนตั้ง (column) ซึ่งเกรเดียน์ตามแนวแกนนอนหาได้จาก

$$Gr(j,k) = F(j,k) - F(j,k-1) \qquad \text{(สมการ2.12a)}$$

และเกรเดียนท์ในแนวแกนตั้งหาได้จาก

$$Gc(j,k) = F(j,k) - F(j+1,k) \qquad \text{(สมการ2.12b)}$$

โดยเกรเดียนท์นี้จะนำมาใช้หาขอบภาพก็ต่อเมื่อ Gc และ Gr มีค่าเป็นบวกจากซ้ายไปขวาหรือจากล่างขึ้นบนของภาพอิมเมจ

ตัวอย่างเช่นจะหาค่าผลต่างความเข้มโดยใช้เกรเดียนท์ตามแนวแกนนอนของรูปที่ 2.8 เป็นขอบของภาพแบบแรมป์ในแนวดิ่ง (vertical ramp edge model) ได้ดังนี้

0 0 0 0 h 0 0 0 0

โดยที่ h = b –a ซึ่งหมายถึง ความสูงของขั้นบันไดและเกรเดียนท์ในแนวแกนนอนของขอบภาพแบบแรมป์ในแนวดิ่งคือ

$$0\ 0\ 0\ 0\ \frac{h}{2}\ \frac{h}{2}\ 0\ 0\ 0$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากค่าข้างบนนี้จะมีค่าที่เท่ากันอยู่ 1 คู่

สำหรับเกรเดียนท์ของขอบในแนวทแยง (diagonal edge gradient) จะหาได้จากผลต่างของจุดแต่ละคู่ตามแนวทแยง ซึ่งในเรื่องนี้จะใช้วิธีการเรียกว่า โรเบิร์ตดิฟเฟอเรนซ์โอเปอเรเตอร์ (Robert Difference operator)

โรเบิร์ตดิฟเฟอเรนซ์โอเปอเรเตอร์

สำหรับโอเปอเรเตอร์ชนิดนี้หาได้จาก

$$G(j,k) = \left| G1(j,k) \right| + \left| G2(j,k) \right| \qquad \text{(สมการ2.13a)}$$

หรืออาจหาได้จาก

$$G(j,k) = \left\{ \left[G1(j,k) \right]^2 + \left[G2(j,k) \right]^2 \right\}^{1/2} \qquad \text{(สมการ2.13b)}$$

เมื่อ

$$G1(j,k) = F(j,k) - F(j+1,K+1) \qquad \text{(สมการ2.13c)}$$

$$G2(j,k) = F(j,k+1) - F(j+1,K) \qquad \text{(สมการ2.13d)}$$

และทิศทางของขอบภาพคือ

$$\theta(j,k) = \frac{\pi}{4} + \tan^{-1}\left\{ \frac{Gc(j,k)}{Gr(j,k)} \right\} \qquad \text{(สมการ 2.14)}$$

การหาขอบในทิศทางเดียวเช่นนี้จะ ไม่ค่อยสมบูรณ์นักสำหรับภาพอิมเมจที่มีความส่องสว่างเพียงเล็กน้อย ซึ่งการแก้ปัญหานำได้โดยการหาเกรเดียนท์ใน 2 มิติ คือทำการหาผลต่างใน 1 ทิศทางแต่จะแสดงค่าออกมาเป็นค่าเฉลี่ยทั้งหมดในทิศทางมุมฉาก

พรีวิทท์โอเปอเรเตอร์ (Prewitt operator)

A0	A1	A2
A7	F(j,k)	A3
A6	A5	A4

รูป 2.10 ลักษณะของเทมเพลทที่มีขนาด 3 x 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โอเปอร์เรเตอร์ชนิดนี้จะใช้เทมเพลท (template) ที่มีขนาด 3*3 ในการหาเกรเดียนท์โอเปอเรเตอร์ ให้แต่ละจุดมีการเรียงลำดับเกรเดียนท์ของขอบคือ

$$G(j,k) = \left\{ [Gr(j,k)]^2 + [Gc(j,k)]^2 \right\}^{1/2} \qquad \text{(สมการ2.15a)}$$

โดยที่

$$Gr(j,k) = \frac{1}{k+2}\left[(A2 + KA3 + A4) - (A0 + KA7 + A6)\right] \qquad \text{(สมการ 2.15b)}$$

$$Gc(j,k) = \frac{1}{k+2}\left[(A0 + KA1 + A2) - (A6 + KA5 + A4)\right] \qquad \text{(สมการ2.15c)}$$

การหาโอเปอเรเตอร์แบบพรีวิทท์ค่า K ที่นำมาใช้มีค่าเท่ากับ 1 เมื่อแทนค่าลงไปในสูตรข้างต้นของเกรเดียนท์ในแนวนอนและแนวตั้ง สุดท้ายจะได้ยูนิตเกน (unit gain) ที่มีค่าเป็นบวกและลบของค่าเฉลี่ยความเข้มเพื่อหาตำแหน่งของขอบภาพ

โซเบลโอเปอเรเตอร์ (Sobel operator)

การหาขอบภาพแบบโซเบลนี้จะแตกต่างจากแบบพรีวิทท์ตรงที่ค่าความเข้มของภาพในทิศเหนือ, ใต้, ตะวันออก, ตะวันตก จะมีค่าเพิ่มขึ้นอีกเท่าตัวในที่นี้คือค่า K ในสมการที่ (2.15b) และ (2.15c) มีค่าเท่ากับ 2 ดังนั้นค่าน้ำหนัก (Weight) ที่เปลี่ยนไปในแต่ละจุดคือตัวที่แสดงการกระจายของเกรเดียนท์

พรีวิทท์โอเปอเรเตอร์จะใช้สำหรับการหาขอบภาพในแนวตั้งและแนวนอนได้ดีกว่าโอเปอเรเตอร์แบบโซเบล แต่โซเบลโอเปอเรเตอร์จะใช้ในการหาขอบภาพในทิศทางแนวทแยงได้ดี การใช้โอเปอเรเตอร์แบบพรีวิทท์ และโซเบล จะทำให้ได้ขอบภาพของวัตถุที่มีความชัดเจนกว่าการใช้โรเบิร์ตโอเปอเรเตอร์ ทั้งนี้เพราะว่ามีขนาดของโอเปอเรเตอร์ที่ใหญ่กว่าทำให้การเฉลี่ยค่าความเข้มในส่วนที่มีความต่างสว่างน้อยได้ดีกว่าเกรเดียนท์ในแนวนอนและแนวตั้งของดีเทคเตอร์ (detector) ชนิดต่าง ๆ ที่กล่าวมาข้างต้นจะประกอบด้วยการนำความเข้มรอบ ๆ จุดที่ต้องการหาค่ามาร่วมกัน ดังนั้นเกรเดียนท์ในแนวนอนและแนวตั้งจะคำนวณจากความสัมพันธ์ดังนี้คือ

$$Gr(j,k) = F(j,k) * Hr(j,k) \qquad \text{(สมการ2.16a)}$$

$$Gc(j,k) = F(j,k) * Hc(j,k) \qquad \text{(สมการ2.16b)}$$

โดยที่ Hc(j,k) คืออิมพัลส์เรสปอนส์ในแนวแกนนอน

และ Hc(j,k) คืออิมพัลส์เรสปอนส์ในแนวแกนตั้ง

ในรูปต่อไปนี้จะแสดงเกรเดียนท์ในแนวแกนนอนและแนวแกนตั้งของโอเปอเรเตอร์ชนิดต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

55496

Operator	Row gradient	Column gradient
Pixel difference	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Robert	$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
prewitt	$1/3 \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$	$1/3 \begin{bmatrix} 1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Sobel	$1/4 \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	$1/4 \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

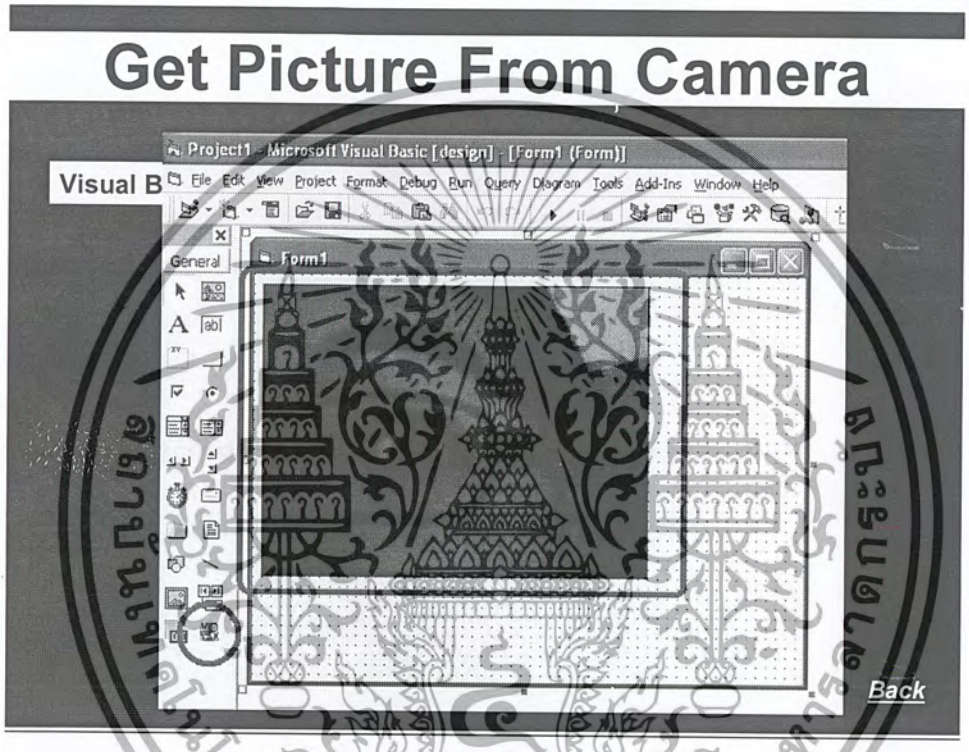
รูปที่ 2.11 แสดงอิมพัลส์เรสปอนส์ของเกรเดียนท์โอเปอเรเตอร์ในแนวมุมฉาก

โอเปอเรเตอร์ที่กล่าวมาข้างต้นที่กล่าวมาทั้งหมดนี้มีข้อจำกัดตรงที่จะไม่สามารถ หาขอบภาพได้ถูกต้องแม่นยำในสภาพที่มีการรบกวนสูง (noise) ซึ่งปัญหานี้อาจแก้ไขได้โดยการใช้โอเปอเรเตอร์มีขนาดใหญ่ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การติดต่อกับกล้อง

ในโครงงานนี้ได้ทำการศึกษาการใช้งาน Visual Basic เพื่อนำมาใช้งาน โดยการติดต่อกับกล้องนั้นจะเข้าไปที่ Visual basic Menu แล้วเข้าไปที่ Project component ซึ่งในโครงงานนี้ได้เลือกใช้ component ที่ชื่อว่า ezvidcab component by Ray Mercer (VB6) มาใช้งาน ซึ่งเมื่อนำกล้องมาต่อเข้ากับคอมพิวเตอร์ส่วนบุคคล (personal computer) ภาพที่กล้องดิจิตอลจับได้ในขณะนั้นก็จะทำการแสดงผลออกมาที่ From 1 ดังภาพ



รูปที่ 2.12 รูปแสดงการติดต่อกับกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบการตรวจจับวัตถุโดยกระบวนการทางภาพ

3.1 การออกแบบโปรแกรมควบคุมการตรวจจับวัตถุด้วย Visual Basic

ในส่วนของโปรแกรมที่ใช้ในการควบคุมการตรวจจับวัตถุนั้น ได้ทำการศึกษาเกี่ยวกับโปรแกรม Visual Basic 6 แล้วนำมาใช้งาน โดยได้ทำการแบ่งโปรแกรมออกเป็นหลายส่วนด้วยกัน คือ การจับภาพที่ต้องการ การลดสัญญาณรบกวนจากภาพที่ได้มา การเน้นขอบของวัตถุให้เด่นชัดขึ้นจากพื้นหลัง การหาค่าเทรชโฮลด์ เพื่อให้วัตถุเป็นภาพไบนารี และทำการ contour เพื่อหาทิศทางขอบของวัตถุ จากนั้นก็เข้าสู่กระบวนการจดจำวัตถุ

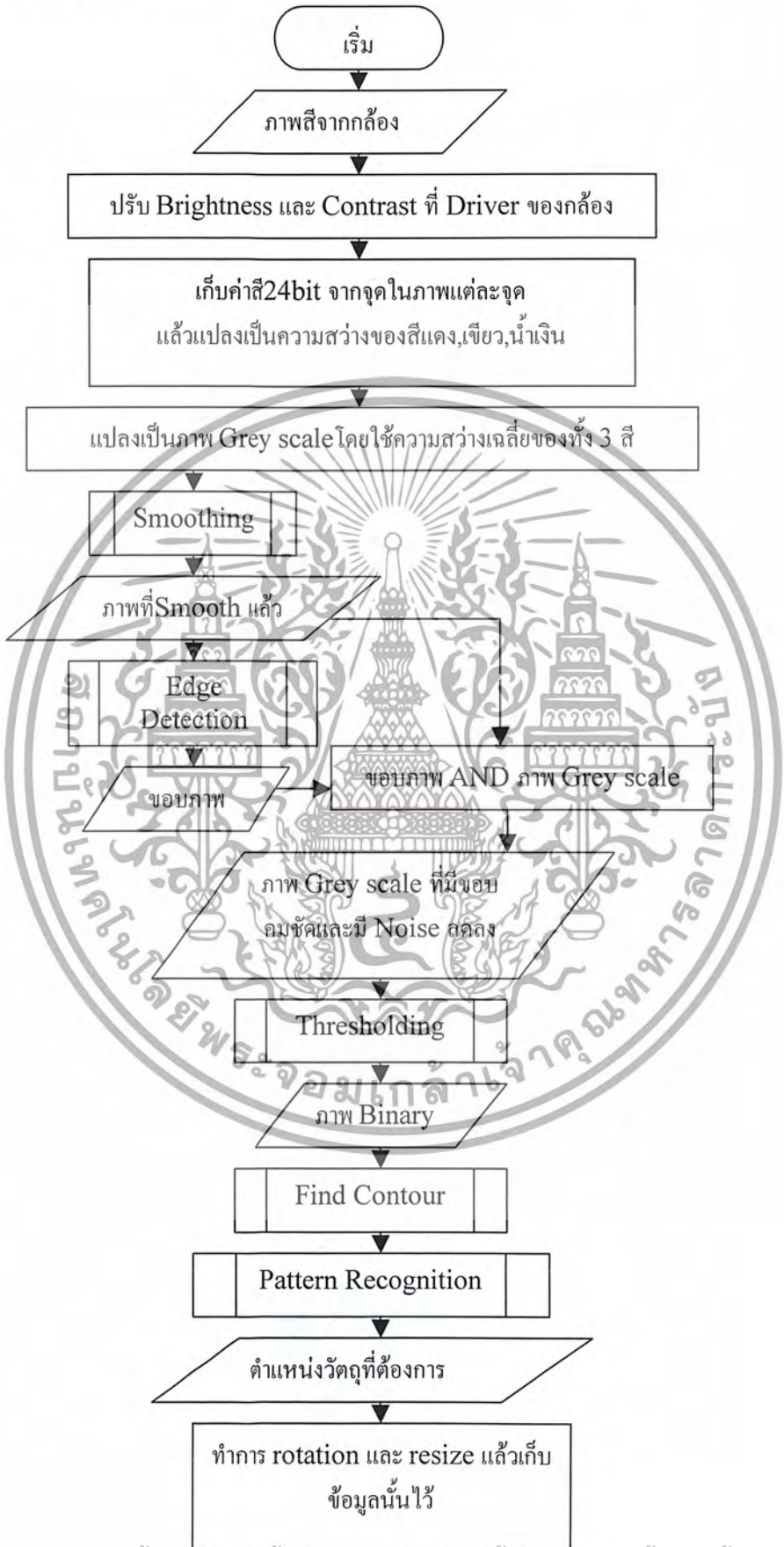
ขั้นตอนการออกแบบทดลอง

1. บันทึกภาพที่ต้องการด้วยกล้องดิจิตอลแล้วนำมาประมวลผลด้วยคอมพิวเตอร์
2. นำภาพสีที่ได้มาทำการปรับ Brightness และ Contrast ที่ Driver ของกล้อง แล้วทำการแปลงภาพที่ได้มาให้เป็นภาพขาว – ดำโดยเก็บค่าสี 24 bit จากจุดในภาพแต่ละจุดแล้วแปลงเป็นความสว่างของสีแดง,เขียว,น้ำเงิน จากนั้นแปลงเป็นภาพ Gray Scale โดยใช้ความสว่างเฉลี่ยของทั้ง 3 สี
3. ทำการลดสัญญาณรบกวนจากภาพที่ได้มาโดยกระบวนการ Smoothing เพื่อเป็นการลด noise ก่อนที่จะนำไป Boost edge
4. นำภาพที่ได้จากข้อ 3 มาทำการหาขอบภาพ ด้วยกระบวนการ Boost edge โดยการหาขอบภาพแล้วนำไป and กับภาพเดิม
5. ทำการแปลงภาพที่ได้ให้เป็นภาพ Binary โดยกระบวนการทำ Thresholding จะทำให้ภาพมีเฉพาะข้อมูล 0 กับ 1
6. ทำการ Contour เพื่อหา Chain code และในขณะเดียวกันเมื่อมีการ contour ไปยังจุดแต่ละจุดบนภาพก็จะทำให้เราได้ข้อมูลของค่าสูงสุดและค่าต่ำสุดของแกน X และ Y

ซึ่งสามารถนำขั้นตอนการออกแบบการทดลองนี้มาเขียน Flow chart ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Flowchart

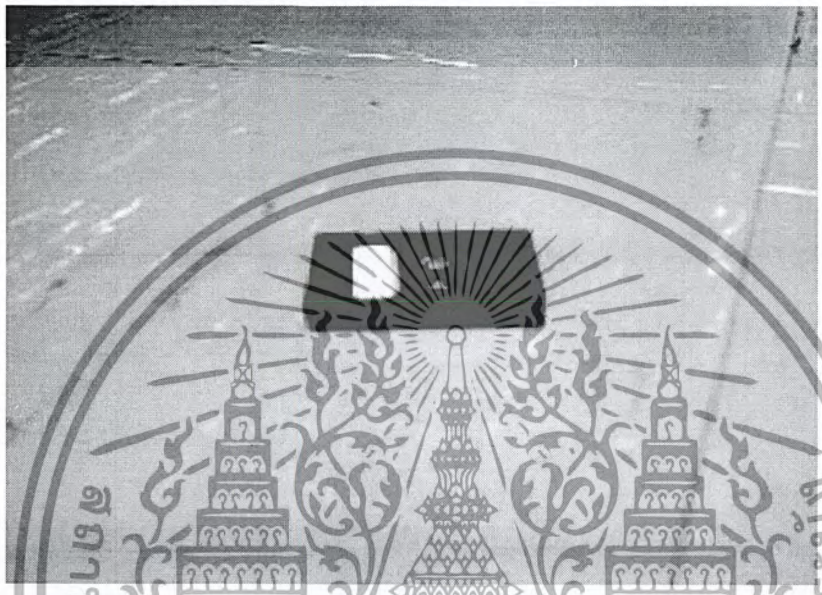


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพสีจากกล้อง

การจับภาพที่ต้องการ

ในส่วนของการจับภาพนี้ เป็นการนำภาพที่ได้จากกล้องดิจิตอลนำมาประมวลผล โดยจะได้ภาพจากกล้องเป็นดังรูปที่ 3.1แล้วนำภาพที่ได้ไปผ่านขั้นตอนต่อไป



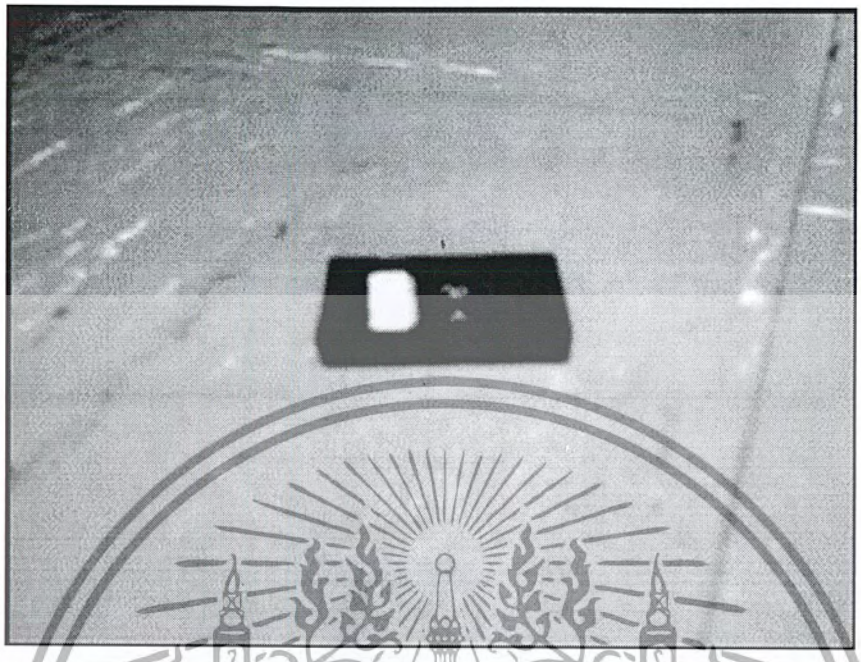
รูปที่ 3.1 ภาพสีจากกล้อง

ภาพ Grey scale

รูปที่ 3.2 ภาพ Grey scale

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่Smooth แล้ว



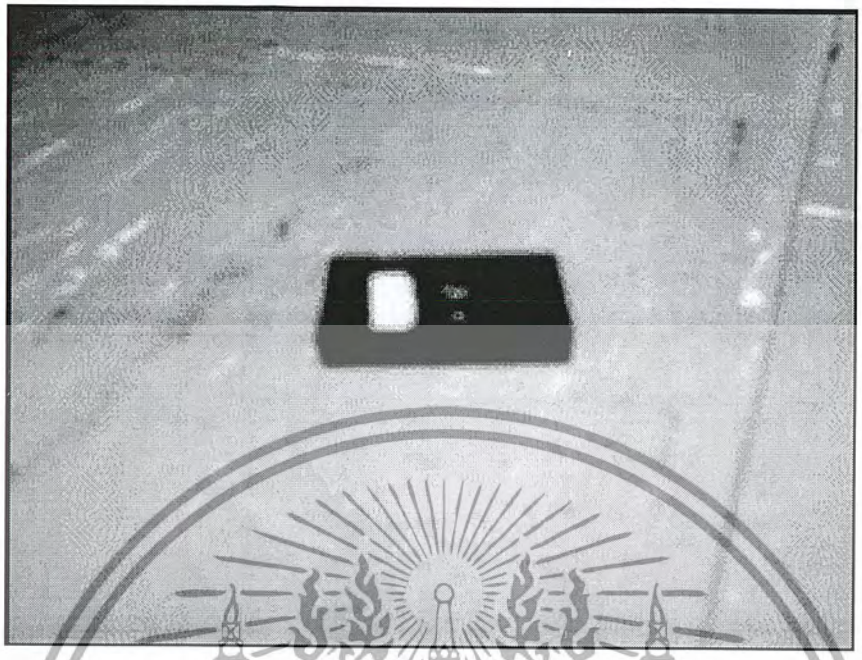
รูปที่ 3.3 ภาพที่ผ่านการ Smooth แล้ว

ขอบภาพ

รูปที่ 3.4 เป็นการแสดงขอบภาพของวัตถุ

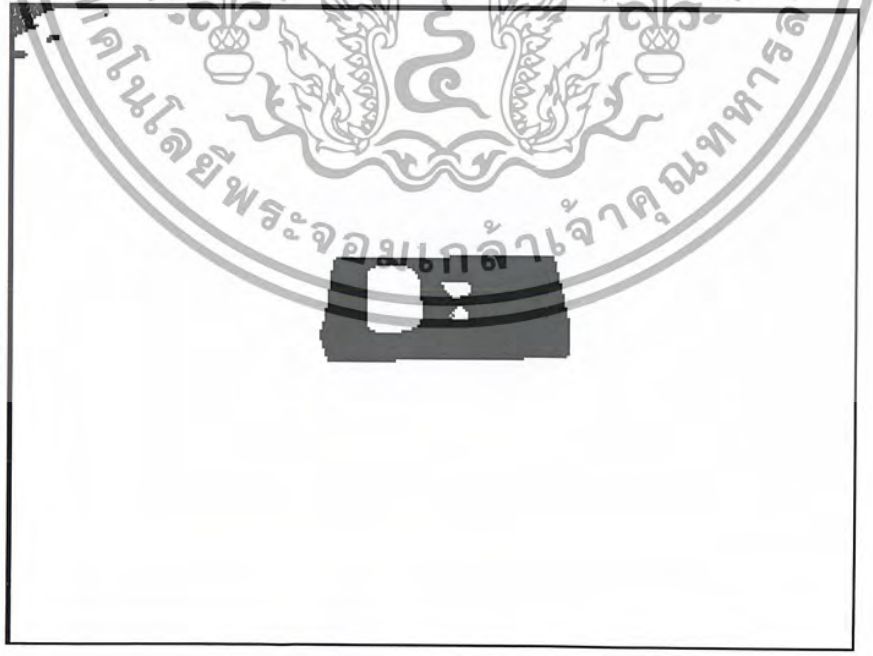
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพ Grey scale ที่มีขอบ
คมชัดและมี Noise ลดลง



รูปที่ 3.5 แสดงภาพ Grey scale ที่มีขอบคมชัดและมี Noise ลดลง

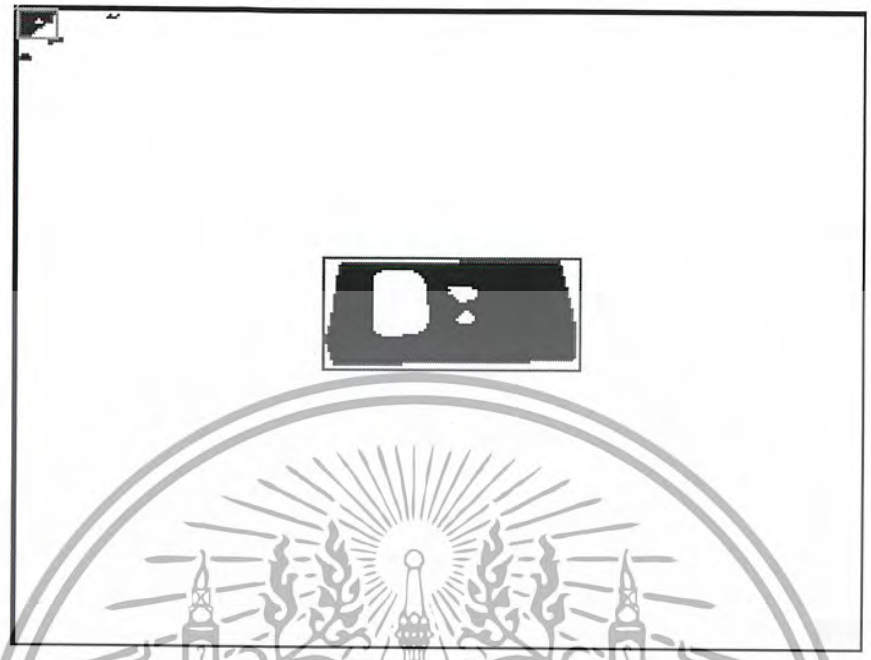
ภาพ Binary



รูปที่ 3.6 แสดงภาพที่ผ่านการ Thresholdingแล้วภาพที่ได้จะเป็นภาพ Binary

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งวัตถุที่ต้องการ



รูปที่ 3.7 แสดงตำแหน่งของวัตถุที่ต้องการ

Smoothing

Smoothing คือ การลดสัญญาณรบกวนของภาพโดยกระบวนการทำภาพให้ Smooth โดยใช้หลักการ low pass filter หลังจากที่เรานำภาพไป Convolution กับ mask ของ low pass filter ซึ่งในโครงงานนี้ได้เลือกใช้ mask

1/10	1/10	1/10
1/10	2/10	1/10
1/10	1/10	1/10

รูปที่ 3.8 แสดง mask ของ lowpass filter

ทำการ Convolution กับภาพต้นแบบ เมื่อภาพที่จับได้จากกล้องดิจิตอลนั้นผ่านกระบวนการ Smoothing แล้วภาพที่ได้ก็จะมีลักษณะเบลอ ไม่ชัด ซึ่งจะเป็นการทำให้สิ่งรบกวน (noise) หรือสิ่งที่เราไม่นำมาพิจารณาในภาพนั้นมีผลต่อภาพของวัตถุที่เราต้องการพิจารณาน้อยลง เนื่องจากถ้าเรานำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพไปประมวลผลในขั้นตอนของการเน้นขอบภาพ (Edge Detection) จะทำให้ในขั้นตอนนั้นทำการเน้น noise ที่เราไม่ต้องการไปด้วย ดังนั้นในขั้นตอนการทำ Smoothing นี้จึงเป็นการลดปัญหาในเรื่องของการที่เราไม่ต้องการเน้น noise ที่เป็นตัวรบกวนสัญญาณได้โดยการทำภาพให้เบลอ ซึ่งจะเป็นการทำให้ noise กลืนไปด้วยกันกับภาพที่เราพิจารณา จากนั้นก็จะสามารถทำการวิเคราะห์เฉพาะภาพของวัตถุได้ง่ายขึ้นซึ่งภาพที่ได้จะมีลักษณะดังรูปที่ 3.3

Edge Detection

Edge Detection คือ การเน้นขอบของวัตถุ ได้โดยการนำภาพที่ได้มาจากขั้นตอนการทำ Smoothing นั้นมาประมวลผลต่อ ซึ่งภาพที่ได้มานั้นจะมีลักษณะเบลอ ไม่ชัด เป็นการทำให้เราสามารถลดสัญญาณรบกวนได้เมื่อมาถึงขั้นตอนการเน้นขอบภาพ เนื่องจากภาพที่มีลักษณะเบลอนั้นจะทำให้ noise ไม่ชัดเจน กลืนไปด้วยกันกับภาพที่เราต้องการพิจารณา ทำให้ง่ายต่อการนำภาพวัตถุมาทำการเน้นขอบ เพื่อเป็นการแบ่งแยกวัตถุที่เราพิจารณากับพื้นหลัง โดยมีขอบของวัตถุเป็นตัวที่เน้นขึ้นมาทำการแบ่งแยก ในขั้นตอนแรกนั้นภาพที่เราได้จากการทำการเน้นขอบโดยกระบวนการ Edge Detection โดยทำการ Convolution กับ mask ของ Light Edge

0	-1	0
-1	4	-1
0	-1	0

รูปที่ 3.9 แสดง mask ของ Light Edge

เมื่อผ่านกระบวนการทำ Convolution แล้ว ภาพที่ได้จะมีคุณสมบัติไม่ได้ตามที่ต้องการดังรูปที่ 3.4 ดังนั้นจึงทำการนำเอาขอบภาพที่หาได้ไป and กับภาพเดิมก็จะได้ภาพดังรูปที่ 3.5 ซึ่งสามารถนำไปประมวลผลต่อไปได้

Thresholding

Thresholding คือ เป็นการทำให้ภาพวัตถุเป็นภาพไบนารีมีเฉพาะข้อมูล 0 กับ 1 โดยใช้ค่าเทรชโฮล์ดเป็นตัวควบคุมในการแปลงค่าของภาพวัตถุให้เป็นภาพไบนารี ซึ่งกำหนดได้โดยการใช้ระดับความเข้มของภาพวัตถุเข้ามาช่วย โดยมีหลักการว่าจุดภาพที่มีคุณสมบัติอยู่ในบางช่วงใด ๆ จะถูกจัด

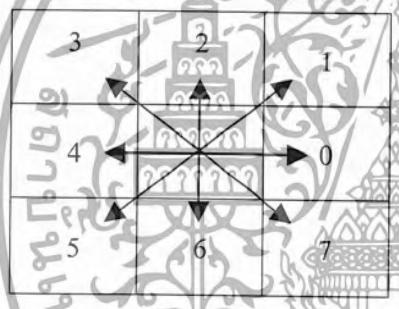
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นกลุ่มได้โดยที่ระดับความเข้มหนึ่งนั้นสามารถที่จะแบ่งแยกกลุ่มของจุดภาพออกเป็น 2 กลุ่มได้อย่างชัดเจน คือกลุ่มของวัตถุ (object) ซึ่งจะมีระดับความเข้มของภาพ ค่อนข้างต่ำ (มืด) กับกลุ่มของส่วนที่เป็นพื้นหลัง (background) ที่จะมีระดับความเข้มของภาพ ค่อนข้างสูง (สว่าง) ระดับความเข้มของภาพที่ถูกแบ่งออกเป็นมืดกับสว่างนี้จะทำให้ได้ภาพที่เป็นไบนารี คือ 0 กับ 1 ซึ่งง่ายต่อการนำไปทำกระบวนการ Contour ต่อไป ดังรูปที่ 3.6

	Find Contour	

Find Contour คือการหาทิศทางของขอบภาพวัตถุ

จากขั้นตอนการทำเทรชโฮลด์จะได้ภาพวัตถุที่เป็นภาพไบนารีมีเฉพาะข้อมูล 0 กับ 1 ดังนั้นจึงนำเอาข้อมูลนั้นมาทำการ Contour หาทิศทางของขอบภาพวัตถุ โดยมีหลักการดังนี้



รูปที่ 3.10 แสดงการกระจายของทิศทางของภาพ (compass direction)

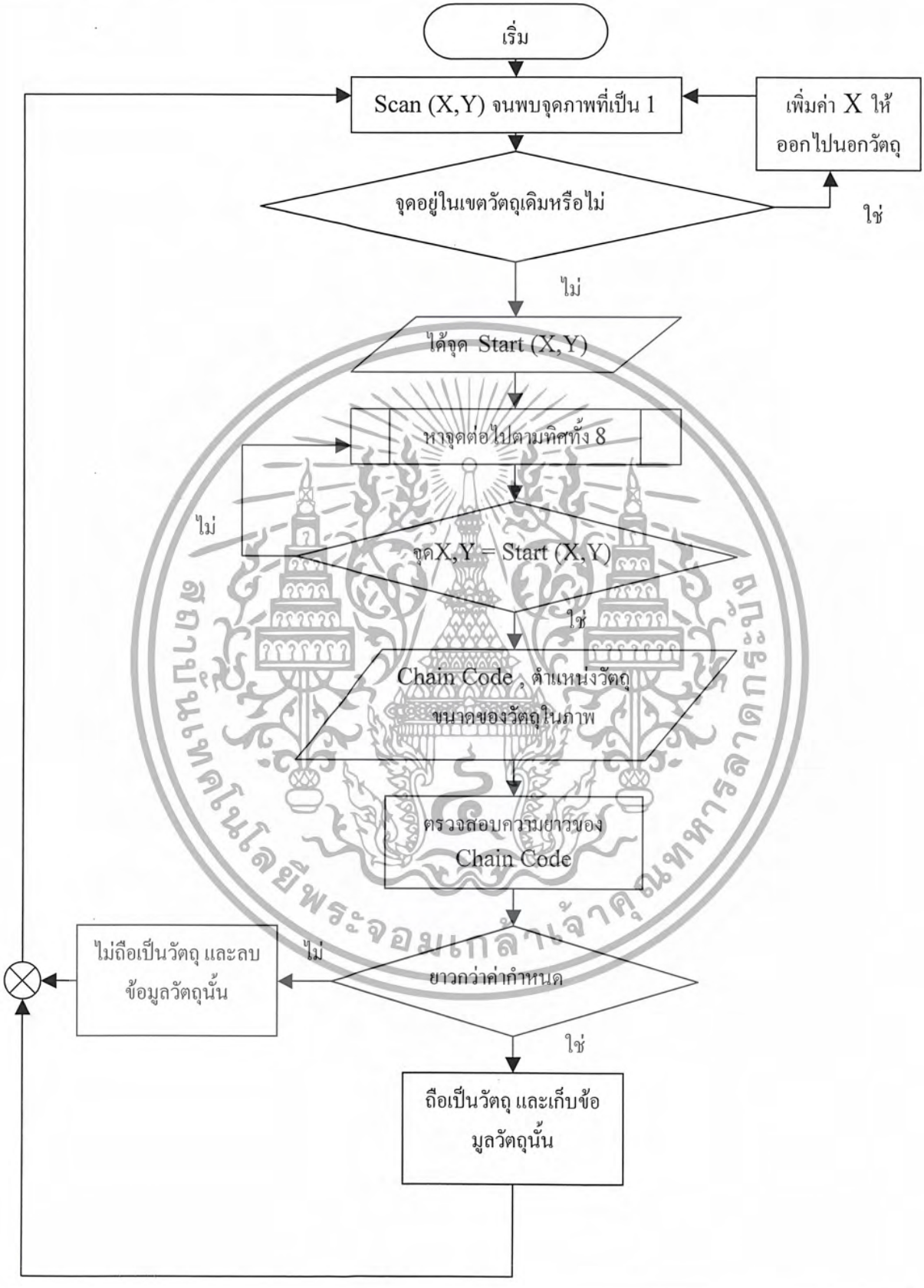
1. ให้จุดเริ่มต้นหรือจุดที่เราพิจารณาในการหาทิศทางในตอนแรกอยู่ที่จุดปัจจุบัน ถ้าจุดปัจจุบันเป็นจุดของขอบภาพหรือมีค่าของจุดเป็น 1 แล้วให้เลี้ยวซ้ายและก้าวเดินตรงไปข้างหน้าโดยให้มีทิศทางตามเข็มนาฬิกาไปยังจุดถัดไป ผลของ Code ที่ได้จะมีค่าตามรูปแสดงการกระจายของทิศทางของภาพ

2. ถ้าเจอจุดที่อยู่ในปัจจุบันเป็นพื้นเบื้องหลัง หรือมีค่าของจุดเป็น 0 ให้เลี้ยวขวา แล้วก้าวเดินตรงไปข้างหน้าโดยวนตามเข็มนาฬิกายังจุดถัดไป ผลของ Code ที่ได้จะมีค่าตามรูปแสดงการกระจายของทิศทางของภาพ

3. การเคลื่อนที่จะสิ้นสุดลงเมื่อจุดที่อยู่ปัจจุบันเป็นจุดเดียวกันกับจุดเริ่มต้น และทุกครั้งที่มีการ Contour นั้นจะมีการจดจำค่าตำแหน่ง (x,y) ไว้ด้วย ซึ่งสามารถเขียน Flow chart ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Finding Contour Flowchart

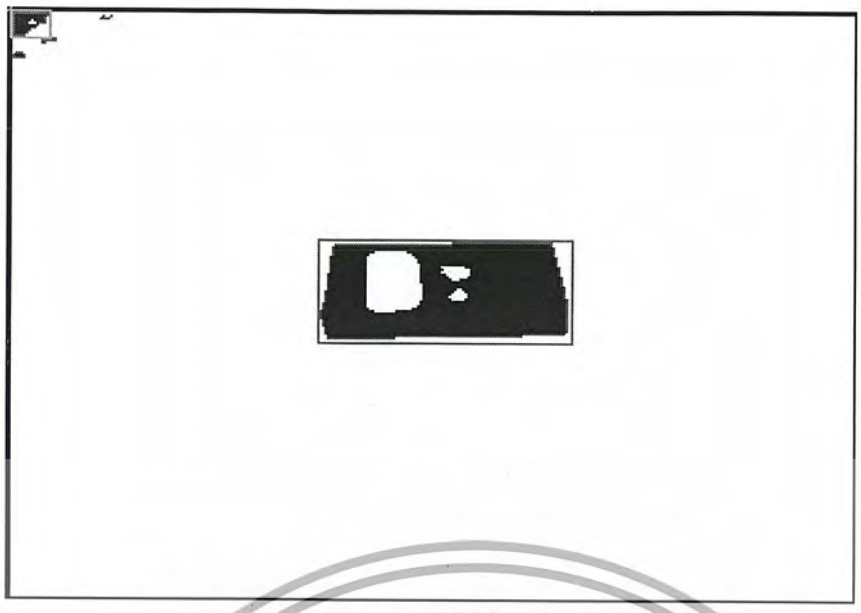


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทิศทางในการ Contour นั้นกล่าวอีกนัยหนึ่งก็คือจะมีทิศทางหมุนตามเข็มนาฬิกา จากทิศทางตรงข้ามกับทิศทางก่อนหน้านี้ โดยให้ทิศต่อจากทิศตรงข้ามเป็นทิศแรก เช่น ถ้าทิศที่มาจากครั้งที่แล้วเป็น 0 ทิศตรงข้ามคือ 4 เราจะให้ทิศที่ 3 เป็นทิศแรก จะได้เป็น 3,2,1,0,7,6,5,4 วิธีการหาแบบนี้จะได้ขอบภาพตามเข็มนาฬิกา และมีผลของค่า Chain Code ตามที่ปรากฏในผลการทดลองดังรูปที่ 3.13 ซึ่งผลนี้สามารถนำเอาไปประมวลผลได้ต่อไป โดยค่า Chain Code นี้จะมีค่าเปลี่ยนไปเมื่อเราเปลี่ยนค่าเทรชโฮลด์ใหม่ จะได้ภาพที่เป็นไบนารีที่มีข้อมูลใหม่ การวิ่งตามเส้นทางที่ Contour ก็จะเปลี่ยนไปตามข้อมูลที่เป็น 1 จึงทำให้ได้ค่า Chain Code ใหม่ด้วย เมื่อสิ้นสุดการทำการ Contour จะทำให้ได้ภาพดังรูปที่ 3.11 และเมื่อสิ้นสุดการ Contour นี้ทำให้รู้ค่าของจุดสูงสุดและจุดต่ำสุดของค่าในแกน Y กับในแกน X จึงทำให้เรารู้พิกัดวัตถุของเราอยู่ในขอบเขตไหนดังรูปที่ 3.12

รูปที่ 3.11 แสดงการทำ Find Contour

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 แสดงการหาตำแหน่งของวัตถุ

Chain Code

```
00000000767666766676667666676666
66675666666665444444444444444544
44444444444444444444444444444444
44444444543445444444444444444444
44444443223223222212221222212222
22122222212210000000000000000000
00000001000000070000010000000000
00000000000000000000001
```

รูปที่ 3.13 แสดงผลค่า Chain Code

	Pattern Recognition	

Pattern Recognition คือ การจดจำลักษณะของวัตถุจากภาพที่เราพิจารณา ซึ่งนำภาพที่ได้จากการ Contour เพื่อหาขอบเขตของวัตถุนั้นนำมาประมวลผลต่อ จากกระบวนการทำ Contour นั้นทำให้สามารถรู้ขอบเขตของตำแหน่งวัตถุได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทำงานของโปรแกรมในส่วนของการจดจำวัตถุ

1.ในขั้นตอนเริ่มต้นนั้น จะนำภาพที่จับได้จากกล้องดิจิตอลมาเข้าสู่กระบวนการปรับปรุงภาพ แล้วแปลงภาพเป็นภาพไบนารี

2.จากนั้นทำการจำแนกวัตถุออกเป็นชิ้น ๆ แล้วนำเข้าสู่กระบวนการจดจำภาพ โดยใช้การ Rotation และการ Resize โดยจะทำการหมุนภาพวัตถุที่ต้องการจำด้วยมุมต่างๆ และ บันทึกภาพนั้นแล้วทำการเก็บข้อมูล ความกว้าง ความยาว เพื่อนำมาหาค่าอัตราส่วนระหว่างความกว้างต่อความยาวของภาพนั้นไว้

3.ภาพตัวอย่างที่ถูกหมุนในมุมต่างๆ และ ข้อมูลต่างๆของภาพนั้นๆจะถูกส่งไปตรวจสอบขนาดความกว้างต่อความยาวของวัตถุ กับ ภาพตัวอย่างแต่ละภาพที่หมุนไว้ โดยจะทำการปรับขนาดวัตถุให้เท่ากับภาพตัวอย่าง จนทำให้ภาพวัตถุมีขนาดเท่ากับภาพตัวอย่าง เพื่อง่ายต่อการตรวจสอบ

4.นำภาพวัตถุมาทำการ XNOR กับภาพตัวอย่างที่ได้ทำการจดจำไว้ก่อนหน้านั้น และนับจำนวนจุดภาพหรือจุดพิกเซลที่เหมือนกัน แล้วนำมาคำนวณหาเปอร์เซ็นต์ความเหมือนกันของทั้ง 2 ภาพ ว่าได้ผลเป็นค่าเปอร์เซ็นต์ที่มากกว่าค่าเปอร์เซ็นต์ที่ยอมรับได้หรือไม่

5.ถ้าผลของค่าเปอร์เซ็นต์ที่ออกมานั้นอยู่ในค่าเปอร์เซ็นต์ที่ยอมรับได้ก็ถือว่าวัตถุนั้นเป็นวัตถุเดียวกับที่ได้จดจำไว้

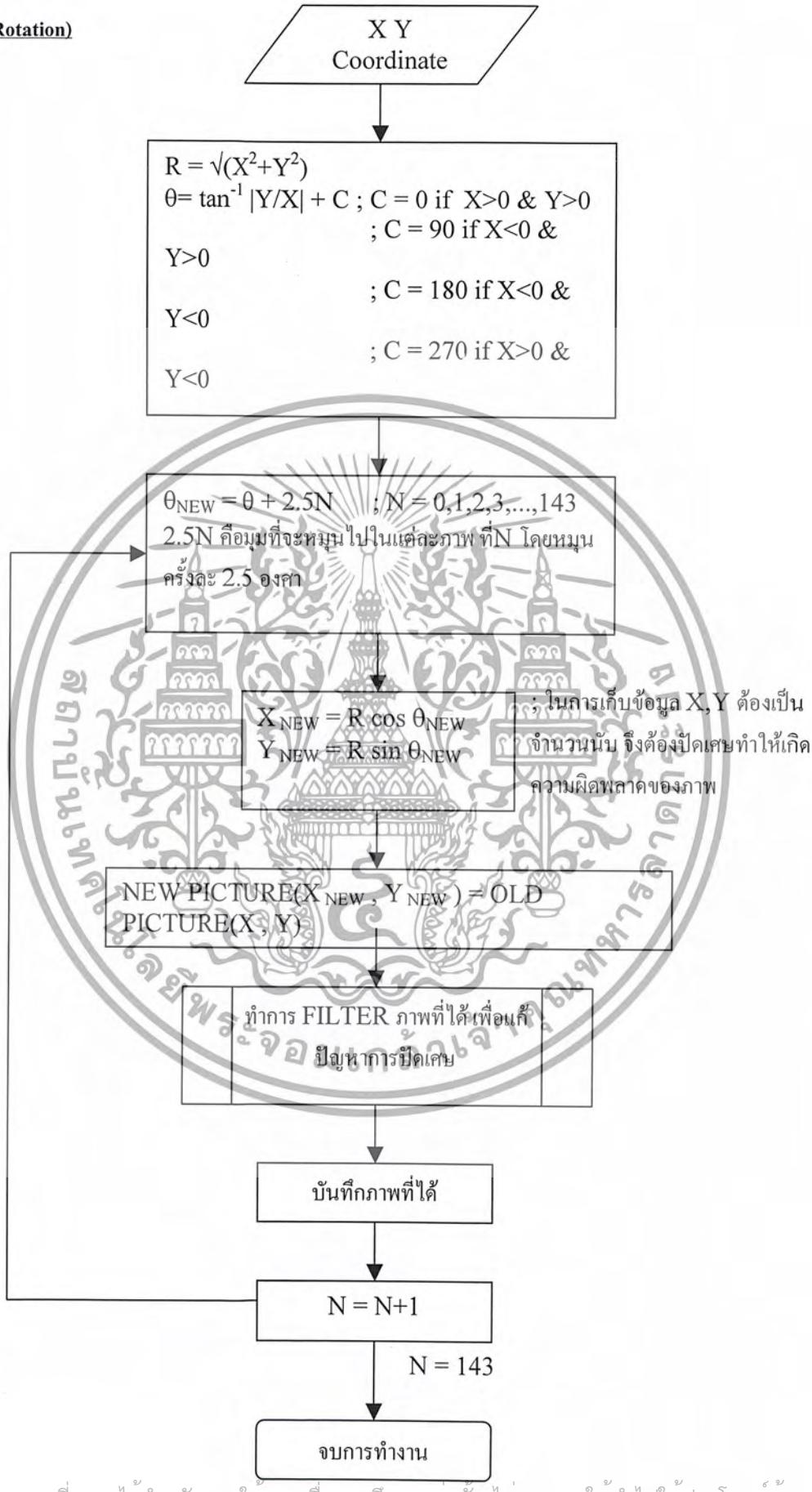
ซึ่งสามารถนำมาเขียนแสดงขั้นตอนการทำงานของโปรแกรมในส่วนของการจดจำวัตถุได้ดังโฟลชาร์ทข้างล่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

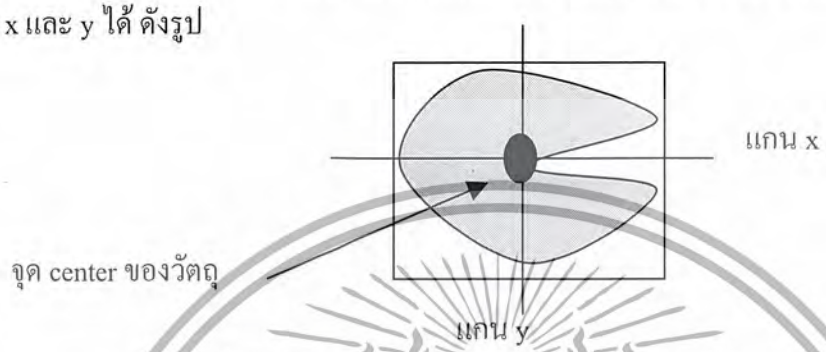
การหมุนภาพ (Rotation)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Rotation	

Rotation คือ การหมุนวัตถุ จากการทำ Pattern Recognition ของวัตถุทำให้รู้ขอบเขตของวัตถุว่าอยู่ในพิกัดจุด (x,y) ใด จากนั้นทำการ get picture ของภาพวัตถุนั้นมา จากภาพของวัตถุนั้นจะได้ค่า X_{max} กับ Y_{max} จากทั้งสองค่านี้ทำให้สามารถหาค่า center ของวัตถุได้ และจากจุด center สามารถสร้างแกน x และ y ได้ ดังรูป



รูปที่ 3.14 แสดงการแบ่งวัตถุให้อยู่ในแกน x,y

จากรูปเมื่อแบ่งวัตถุออกตามแนวแกน x และ y แล้ว จะเห็นได้ว่าวัตถุนั้นถูกแบ่งออกอยู่ทั้ง 4 quadrant และเมื่อทำการ rotate วัตถุนั้น จากเดิมที่วัตถุที่อยู่ในรูปของฟังก์ชั่น (x,y) ก็เปลี่ยนมาอยู่ในรูปของฟังก์ชั่น (r,θ) โดยที่ $r = \sqrt{x^2 + y^2}$ และ $\theta = \tan^{-1}(y/x)$ และเมื่อทำการหมุนวัตถุ(โดยในโครงงานนี้ใช้การหมุนวัตถุทีละ 2.5 ° จนครบ 1 รอบ นั่นคือ 360° ซึ่งจะมีข้อมูลภาพของวัตถุทั้งหมด 144 ค่า) โดยในการหมุนวัตถุนั้นค่า x กับค่า y จะมีค่าเปลี่ยนไปด้วยซึ่ง $x = x - (x_{max}/2)$ และ $y = y - (y_{max}/2)$ จากค่าของ x กับ y นี้ นำมาใช้ในการคำนวณหาค่า θ ใหม่ด้วย โดยจะได้ $\Theta_{new} = \theta + 2.5°N$ (N=0,1,2,...,143) และทำให้ได้ค่า (x,y) ใหม่ด้วย คือ $x = r \cos \Theta_{new}$ และ $y = r \sin \Theta_{new}$ ในการเก็บข้อมูล X,Y นี้มีความจำเป็นที่จะต้องเก็บข้อมูลเป็นจำนวนนับ จึงต้องทำการปัดเศษ ดังนั้นเมื่อทำการปัดเศษข้อมูลแล้ว ทำให้เกิดความผิดพลาดของภาพได้ จึงจำเป็นต้องมีการกรองข้อมูลอีกครั้งเพื่อช่วยให้ข้อมูลนั้นมีความผิดพลาดน้อยลง ซึ่งจะทำการอธิบายถึง Filter นี้ต่อไป

และเมื่อทำการคำนวณใน quadrant ที่ 2 นั้นก็ทำการบวกมุมเพิ่มเข้าไปอีก 90° ในกรณีของ quadrant ที่ 3 นั้นก็ทำการบวกมุมเพิ่มเข้าไปอีก 180° ในกรณีของ quadrant ที่ 4 นั้นก็ทำการบวกมุมเพิ่มเข้าไปอีก 270° ซึ่งแสดงได้ดังโฟลชาร์ทข้างบน

ในการ rotate วัตถุนั้นเป็นการจดจำลักษณะของวัตถุในมุมต่าง ๆ ไม่ว่าวัตถุนั้นจะเปลี่ยนมุมของวัตถุไปอย่างไรก็ทำให้ยังสามารถจดจำวัตถุนั้นได้อยู่ ทำให้ง่ายต่อการนำไปเปรียบเทียบกับภาพที่จับมาได้จากกล้องดิจิตอลด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Dialation	

Dialation คือการลด noise อีกแบบหนึ่ง โดยจะพิจารณาจุดที่มีข้อมูลเป็น 1 แล้วให้จุด บน,ล่าง,ซ้าย ,ขวา ของจุดที่พิจารณานั้นมีข้อมูลเป็น 1 ไปด้วย ซึ่งการทำเช่นนี้จะทำให้ขนาดของภาพมีขนาดใหญ่ขึ้น แต่สามารถลบจุดที่เป็นสัญญาณรบกวนออกไปได้

<u>การ FILTER ภาพที่ได้ เพื่อแก้ปัญหาการปัดเศษ</u>

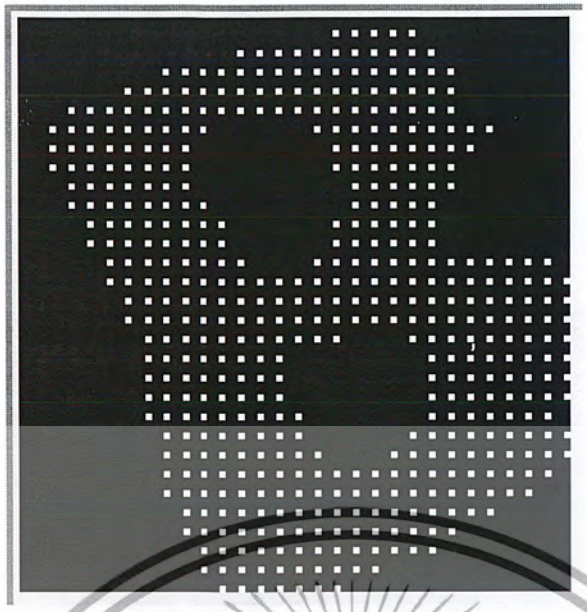
จากที่ได้กล่าวไว้ในเรื่อง Rotation เมื่อทำการหมุนภาพจะทำให้เกิด noise ขึ้น จึงจำเป็นต้องมีการแก้ปัญหา noise ที่เกิดขึ้น ซึ่งในโครงงานนี้ได้นำหลักการของ noise reduction เข้ามาช่วยแก้ปัญหาในเรื่องนี้ เมื่อทำการหมุนภาพวัตถุ จะทำให้ภาพวัตถุมีข้อมูลของภาพเดิมมาไม่ครบ ทำให้เกิดรูโหว่ของภาพวัตถุ ดังรูป

รูปที่ 3.15 ภาพแสดงการเกิด noise เมื่อทำการ Rotation

จากปัญหาที่เกิดขึ้นนี้ทำให้ข้อมูลของภาพเดิมมีไม่ครบ ดังนั้นจึงทำการลดปัญหานี้ด้วยหลักการ Noise reduction

Noise reduction มีหลักการที่ว่า จุดหรือ pixel ที่เป็น 1 อยู่แล้วก็ให้คงข้อมูลเดิมไว้ แต่ในกรณีที่จุดหรือ pixel นั้นเป็น 0 ให้ใช้หลักที่ว่า ให้ตรงจุดที่เป็น 0 นั้นเป็นจุดพิจารณา ถ้าจำนวนจุดที่อยู่ข้างเคียงจุดนี้ทั้ง 8 ด้านมีจุดที่เป็น 1 มากกว่า 3 จุด ก็ให้จุดที่พิจารณานี้เป็น 1 แต่ในทางตรงกันข้าม ถ้ามีน้อยกว่าหรือเท่ากับ 3 จุดก็ยังให้จุดนั้นเป็น 0 เหมือนเดิม ภาพที่ได้ก็จะทำให้มีข้อมูลภาพเดิมเพิ่มมากขึ้น ดังรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 ภาพแสดงการทำ Filter ในกรณีที่จุดข้างเคียงเป็น 1 มากกว่า 3 จุด

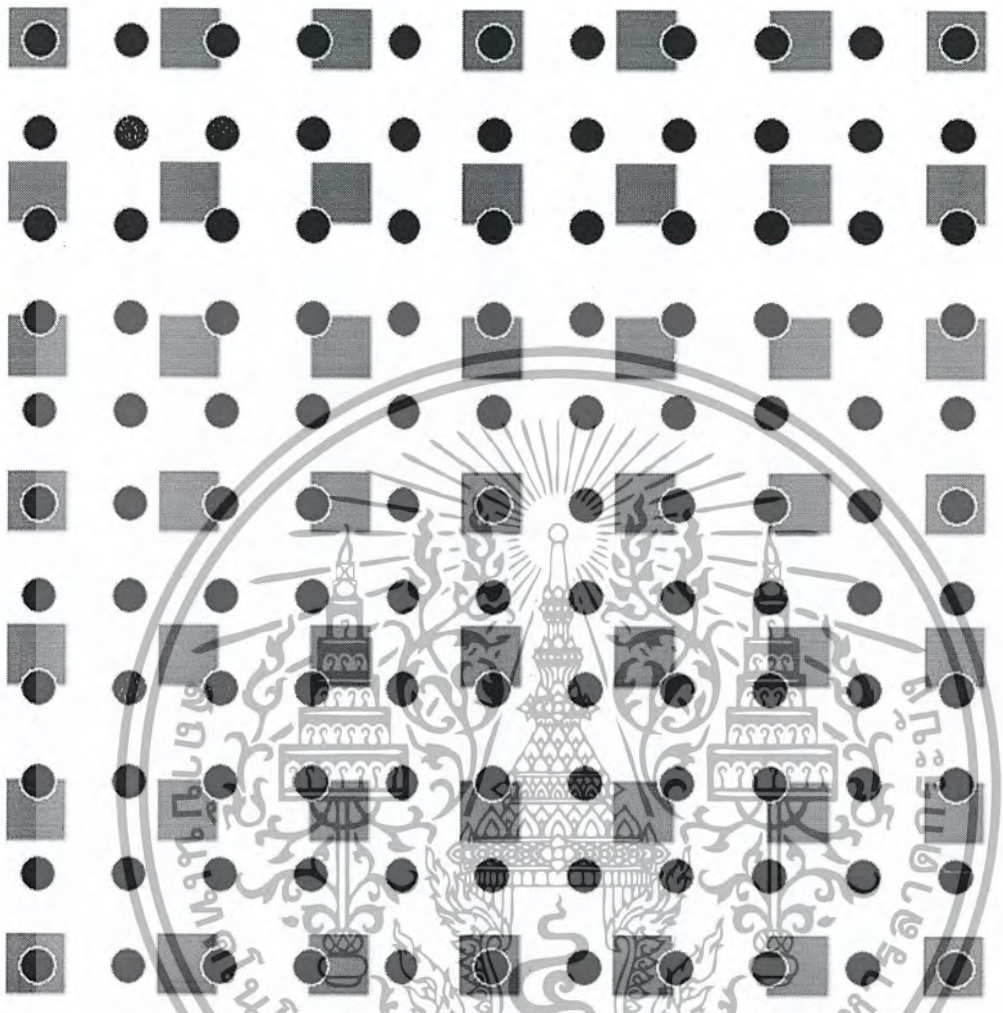
อีกกรณีหนึ่งคิดในกรณีที่จุดข้างเคียงของจุดพิจารณามีข้อมูลเป็น 1 มากกว่า 4 จุด ก็ให้ข้อมูลตรงจุดนั้นเป็น 1 แต่ถ้ามีข้อมูลที่เป็น 1 น้อยกว่าหรือเท่ากับ 4 จุด ก็ยังคงให้จุดนั้นเป็นข้อมูลเดิม หรือก็คือ 0 นั่นเอง ดังภาพ

รูปที่ 3.17 ภาพแสดงการทำ Filter ในกรณีที่จุดข้างเคียงเป็น 1 มากกว่า 4 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับขนาดวัตถุ เราจะใช้การ RESAMPLING ภาพ

ตัวอย่างการ Resampling



รูปที่ 3.18 ภาพแสดง การResampling ภาพขนาด 11 X 11 -> 7 X 7

โดยจุด ดำคือภาพต้นแบบ จุดส้มคือ ภาพที่ได้จากการResampling

อัตราส่วน ของการ Resampling คือ อัตราส่วนระหว่างขนาดของภาพที่จับได้กับขนาดของภาพที่ต้องการให้เท่ากับภาพตัวอย่างที่ได้จดจำไว้ ซึ่งขนาดของภาพกล่าวอีกนัยหนึ่งก็คือจำนวนพิกเซลของภาพนั่นเอง ซึ่งในโครงงานนี้กำหนดให้ค่านี้เป็นค่า ratio จากรูปภาพแสดงการ Resampling ถ้าต้องการขยายวัตถุจาก 7 X 7 เป็น 11 X 11 จะทำได้โดยการประมาณค่าจากพื้นที่สีส้มให้เป็นจุดสีดำ หรือในทางกลับกันถ้าต้องการลดขนาดของวัตถุให้เล็กลงก็ทำได้โดย การประมาณค่าของจุดสีดำให้เป็นค่าของพื้นที่สีส้มที่อยู่ใกล้ที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Resize	

Resize คือ ลักษณะของการลดหรือว่าขยายขนาดของวัตถุให้มีขนาดเท่ากับที่จะนำไปใช้งานได้ เช่น ในกรณีที่ภาพของวัตถุที่จับมาได้มีขนาดใหญ่เกินกว่าข้อมูลที่มีหรือที่ได้จดจำไว้ ฉะนั้นต้องทำการ resize ขนาดของวัตถุให้มีขนาดเล็กลงเพื่อที่จะสามารถนำวัตถุนั้นไปเปรียบเทียบกับวัตถุที่ได้จดจำไว้นั้นได้ หรือในกรณีที่ขนาดของวัตถุมีขนาดใหญ่เกินกว่าข้อมูลที่เรามี ก็ต้อง resize ขนาดของวัตถุให้มีขนาดใหญ่ขึ้นให้มีขนาดเท่ากันกับข้อมูลที่มีอยู่

ภาพของวัตถุที่ได้จับได้จากกล้องเมื่อผ่านการ rotate และ resize วัตถุแล้วก็ทำการเก็บข้อมูลนั้นไว้ เมื่อทำการจับข้อมูลภาพวัตถุมาใหม่ ก็จะสามารถทำการตรวจสอบได้ว่าภาพที่จับมาได้นั้นเป็นวัตถุเดียวกันกับที่เราได้เก็บข้อมูลไว้หรือไม่ โดยการนำภาพที่เราจดจำไว้นั้นมาทำการเปรียบเทียบกับภาพที่จับได้จากกล้องดิจิตอลที่ได้มาใหม่นั้นด้วยการใช้หลักการ XNOR ภาพทั้ง 2 ภาพ โดยเปรียบเทียบภาพในลักษณะของ pixel ต่อ pixel ถ้าข้อมูลของทั้ง 2 ภาพนั้นเหมือนกันผลที่ได้ก็จะเป็น 0 แต่ถ้าทั้ง 2 ภาพใน pixel นั้นไม่เหมือนกันผลลัพธ์ที่ได้ก็จะเป็น 1 ถ้าภาพทั้ง 2 นั้นมีข้อมูลเหมือนกันหรือว่าใกล้เคียงกันมากเปอร์เซ็นต์ของการเป็นวัตถุชิ้นเดียวกันก็จะมีมากด้วย ทำให้สามารถรู้ได้ว่าวัตถุที่กล้องดิจิตอลจับได้นั้นเป็นชิ้นเดียวกันกับที่ได้จดจำไว้ก่อนหน้านี้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

ผลการทดลองของการนำภาพที่พิจารณามากระทำการหาขอบ

จากการออกแบบการทดลองในบทที่ 3 จึงได้ทำการทดลองการปรับปรุงภาพด้วยกระบวนการต่าง ๆ เป็นการทดลองโดยการนำภาพที่จับได้จากกล้องดิจิตอลมาผ่านกระบวนการ Smoothing เพื่อลดสัญญาณรบกวนเปรียบเทียบผลการทดลองกับภาพที่ยังไม่ผ่านกระบวนการ Smoothing แล้วนำภาพทั้ง 2 แบบนั้นมาหาขอบด้วย mask ต่าง ๆ เช่น Sensitive Edge,Sobel 1 ทิศทาง ,Sobel 2 ทิศทาง และ Light Edge แล้วนำภาพที่ผ่านขั้นตอนต่าง ๆ นั้นมาเข้าสู่การหาขอบภาพ โดยทำการ And กับภาพเดิม หรือว่าไม่ And กับภาพเดิม ซึ่งผลจากการทดลองที่ได้ก็จะมีค่าดังตารางข้างล่างนี้ และดังรูปที่ 4.1 ถึงรูปที่ 4.19

ลำดับที่	การ Smoothing	Mask ของ Edge Detection	ลักษณะของภาพ	ผลของภาพที่ได้
1	ไม่ผ่านการ Smoothing	Sensitive Edge	Find Edge	ดังรูปที่ 4.1
2	ไม่ผ่านการ Smoothing	Sensitive Edge	Find Edge & ภาพเดิม	ดังรูปที่ 4.2
3	ไม่ผ่านการ Smoothing	Sensitive Edge	Find Edge & ภาพเดิม และ Thresholding ค่า Threshold = 30	ดังรูปที่ 4.3
4	ผ่านการ Smoothing	Sensitive Edge	Find Edge	ดังรูปที่ 4.4
5	ผ่านการ Smoothing	Sensitive Edge	Find Edge & ภาพเดิม	ดังรูปที่ 4.5
6	ผ่านการ Smoothing	Sensitive Edge	Find Edge & ภาพเดิม และ Thresholding ค่า Threshold = 30	ดังรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7	ไม่ผ่านการ Smoothing	Sobel 1 ทิศทาง	Find Edge	ดังรูปที่ 4.7
8	ไม่ผ่านการ Smoothing	Sobel 2 ทิศทาง	Find Edge	ดังรูปที่ 4.8
9	ไม่ผ่านการ Smoothing	Sobel 2 ทิศทาง	Find Edge & ภาพเดิม	ดังรูปที่ 4.9
10	ไม่ผ่านการ Smoothing	Sobel 2 ทิศทาง	Find Edge & ภาพเดิม และ Thresholding ค่า Threshold = 30	ดังรูปที่ 4.10
11	ผ่านการ Smoothing	Sobel 2 ทิศทาง	Find Edge	ดังรูปที่ 4.11
12	ผ่านการ Smoothing	Sobel 2 ทิศทาง	Find Edge & ภาพเดิม	ดังรูปที่ 4.12
13	ผ่านการ Smoothing	Sobel 2 ทิศทาง	Find Edge & ภาพเดิม และ Thresholding ค่า Threshold = 30	ดังรูปที่ 4.13
14	ไม่ผ่านการ Smoothing	Light Edge	Find Edge	ดังรูปที่ 4.14
15	ไม่ผ่านการ Smoothing	Light Edge	Find Edge & ภาพเดิม	ดังรูปที่ 4.15
16	ไม่ผ่านการ Smoothing	Light Edge	Find Edge & ภาพเดิม และ Thresholding ค่า Threshold = 30	ดังรูปที่ 4.16
17	ผ่านการ Smoothing	Light Edge	Find Edge	ดังรูปที่ 4.17
18	ผ่านการ Smoothing	Light Edge	Find Edge & ภาพเดิม	ดังรูปที่ 4.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

19	ผ่านการ Smoothing	Light Edge	Find Edge & ภาพเดิม และ Thresholding ค่า Threshold = 30	ดังรูปที่ 4.19

หมายเหตุ

1. Mask ของ Smooth

1/10	1/10	1/10
1/10	2/10	1/10
1/10	1/10	1/10

2. Mask ของ Light Edge

0	-1	0-
-1	4	-1
0	-1	0

3. Mask ของSensitive Edge

-1	-1	-1
-1	8	-1
-1	-1	-1

4. Mask ของ Sobel คือ

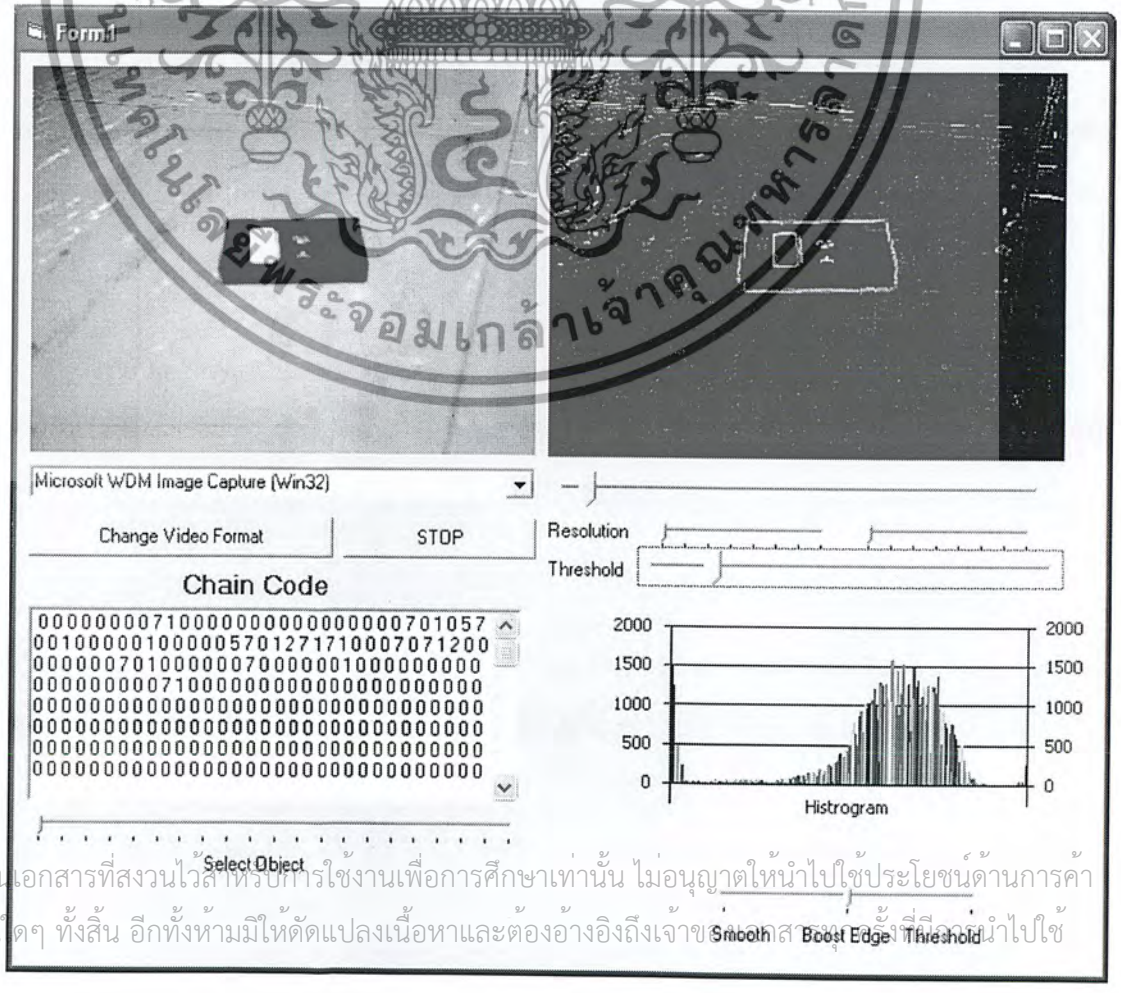
1	0	-1
2	0	-2
1	0	-1

5. Mask ของ Sobel 2 ทิศทาง คือ

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

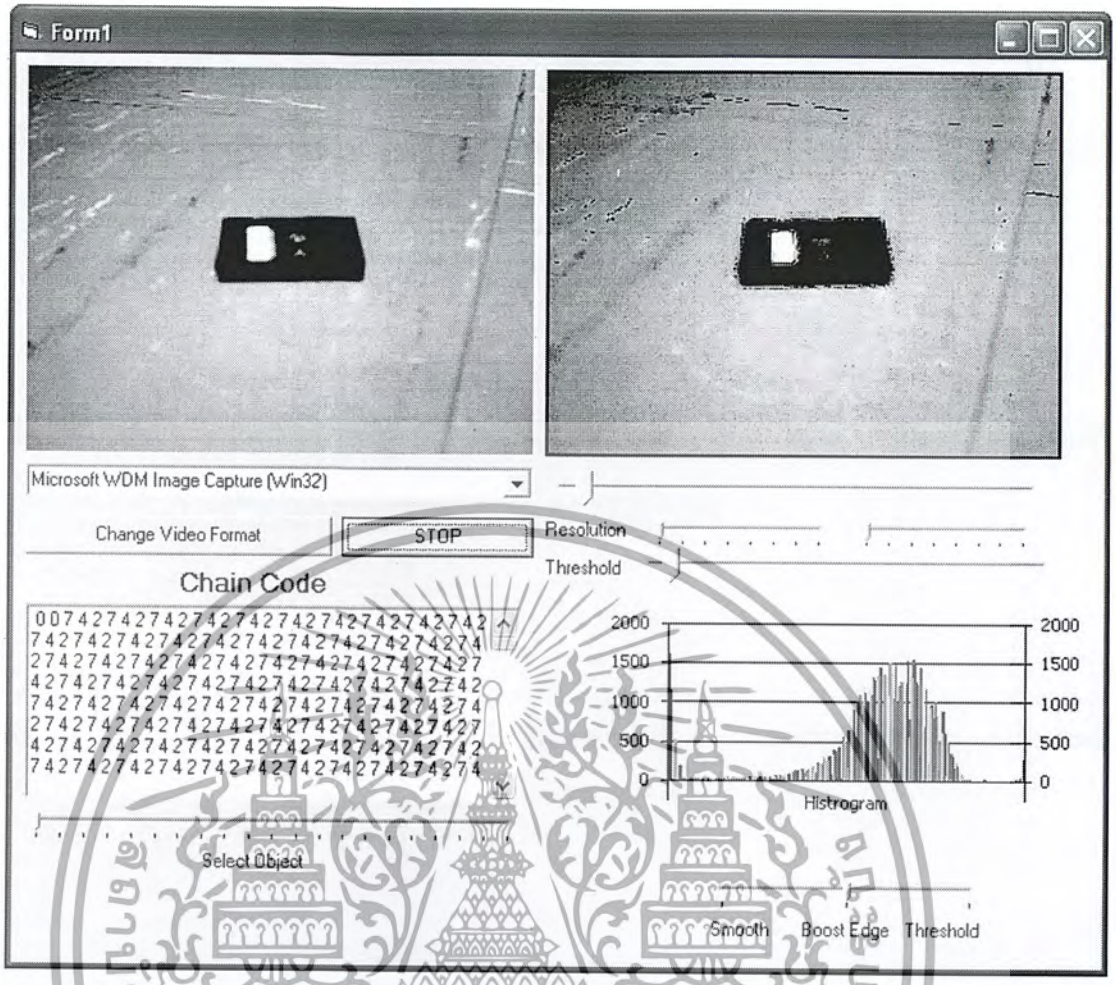
&

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

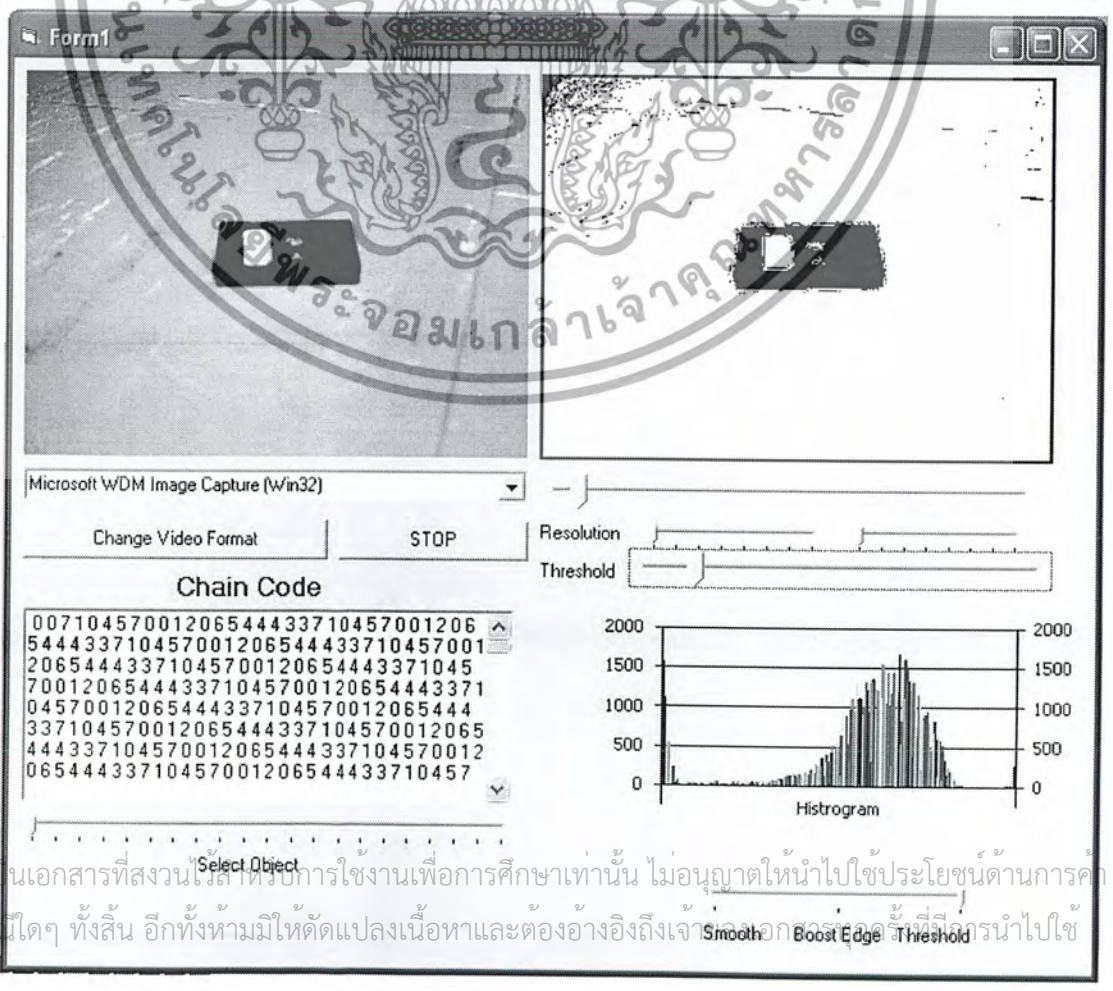


รูปที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

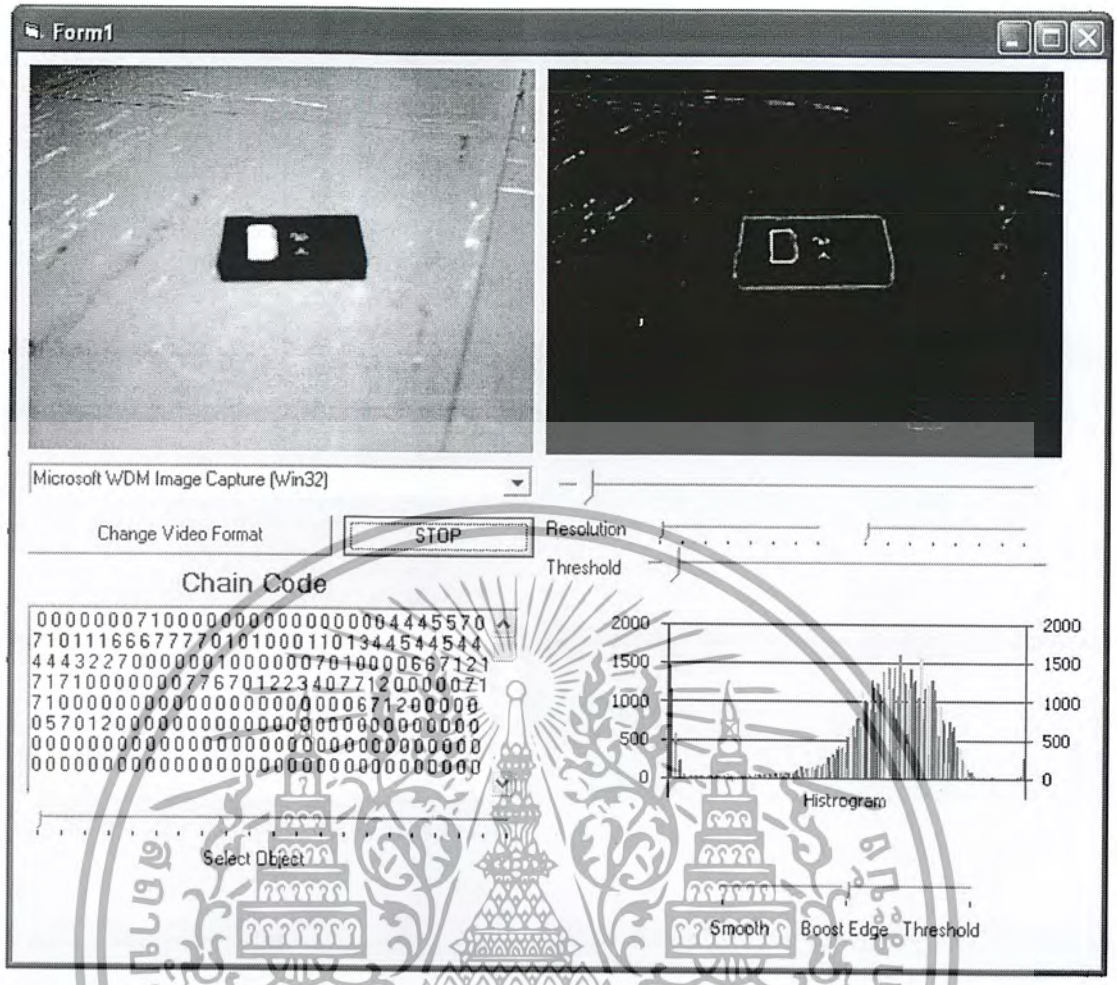


รูปที่ 4.2

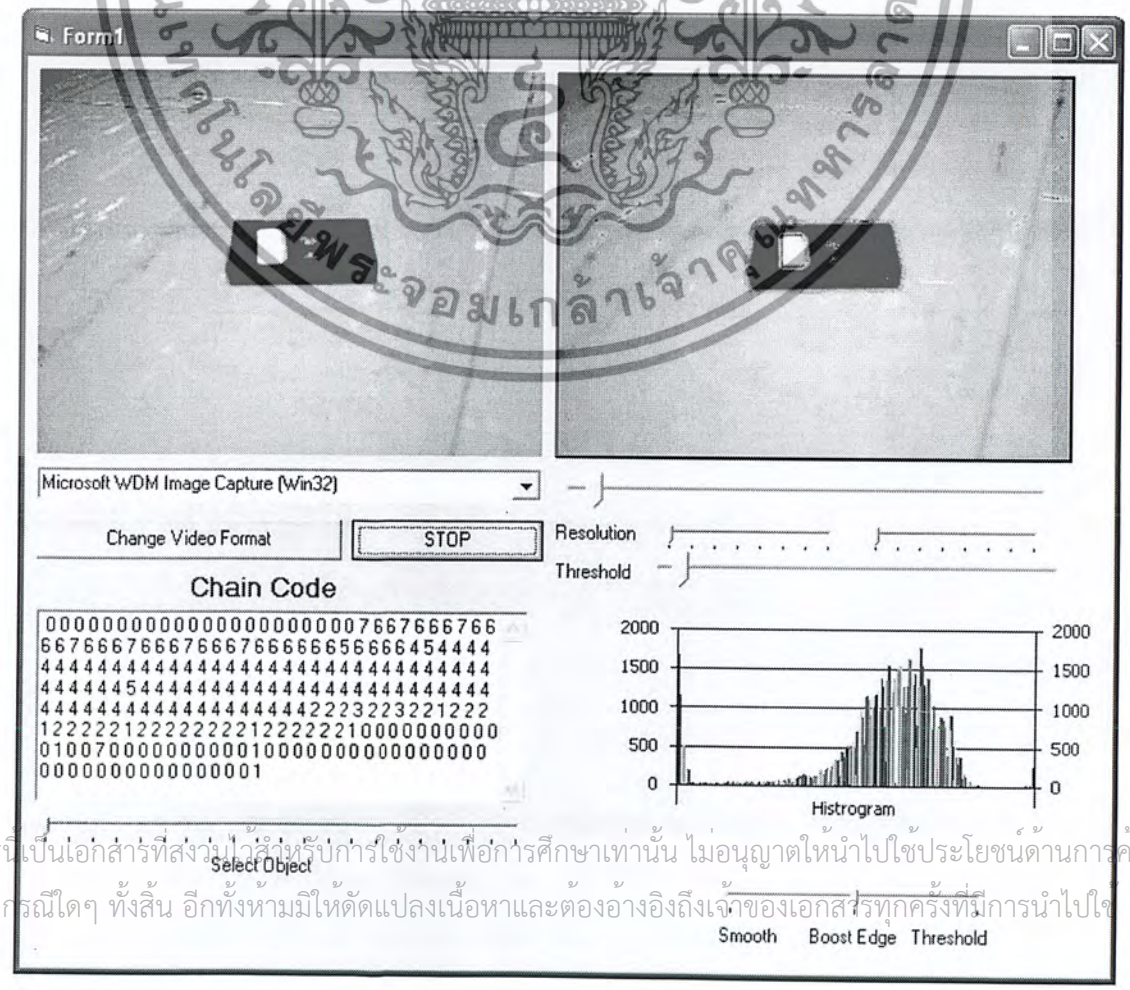


รูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

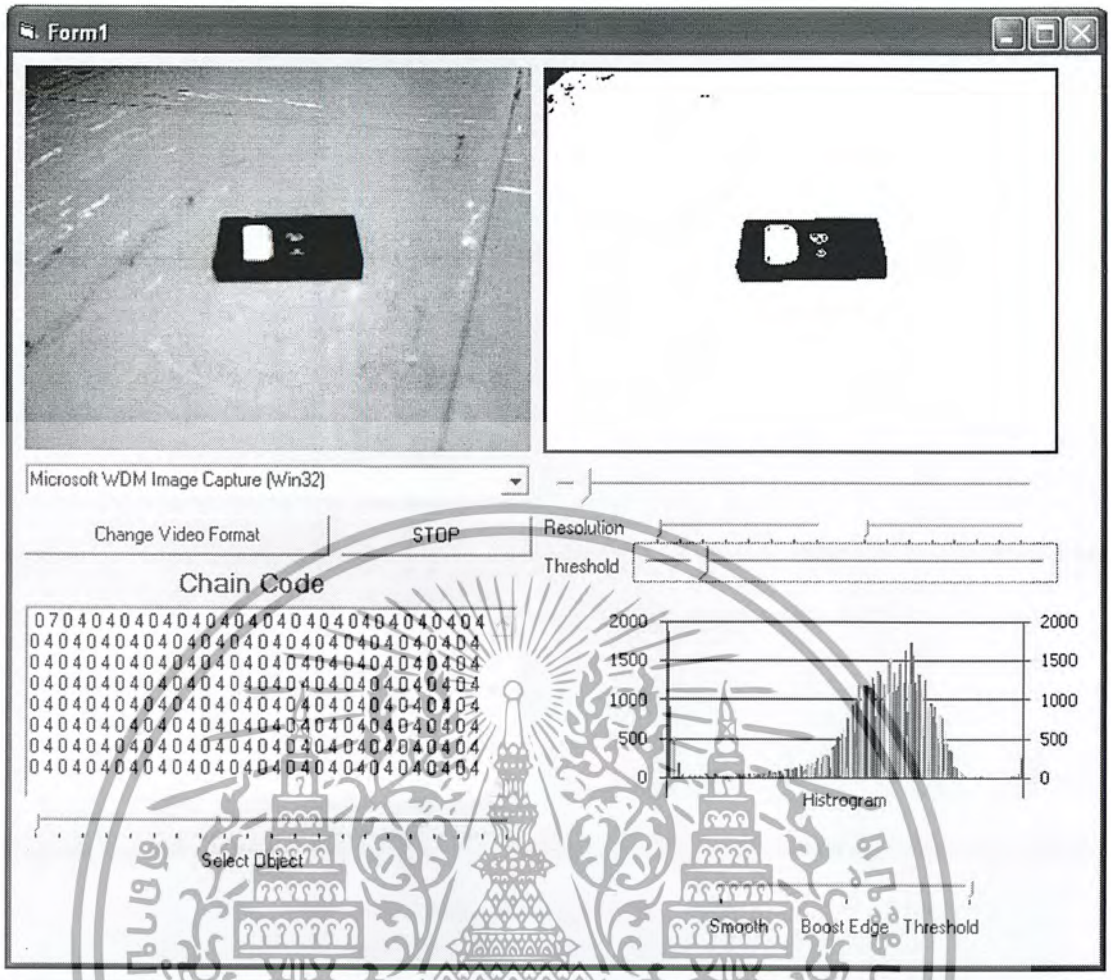


รูปที่ 4.4

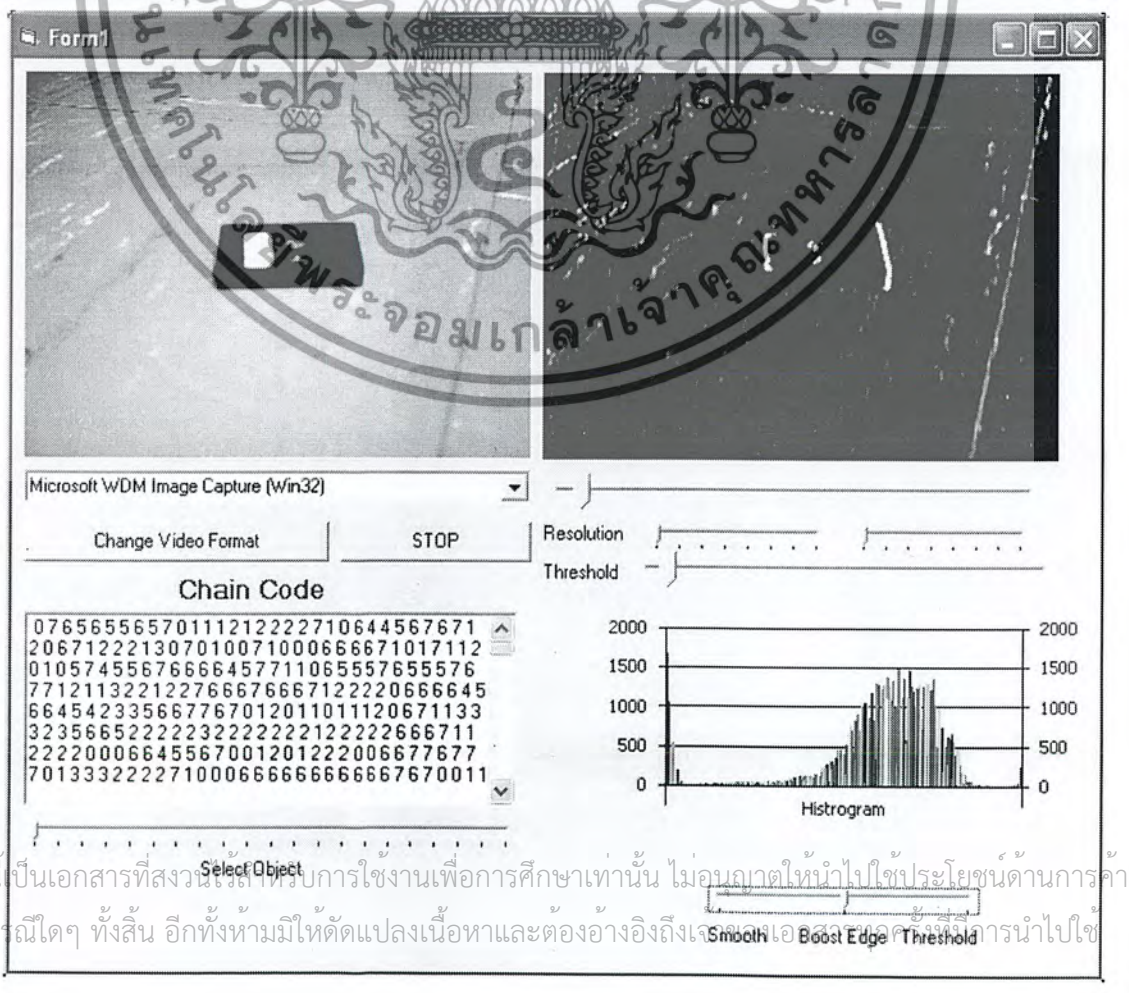


รูปที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

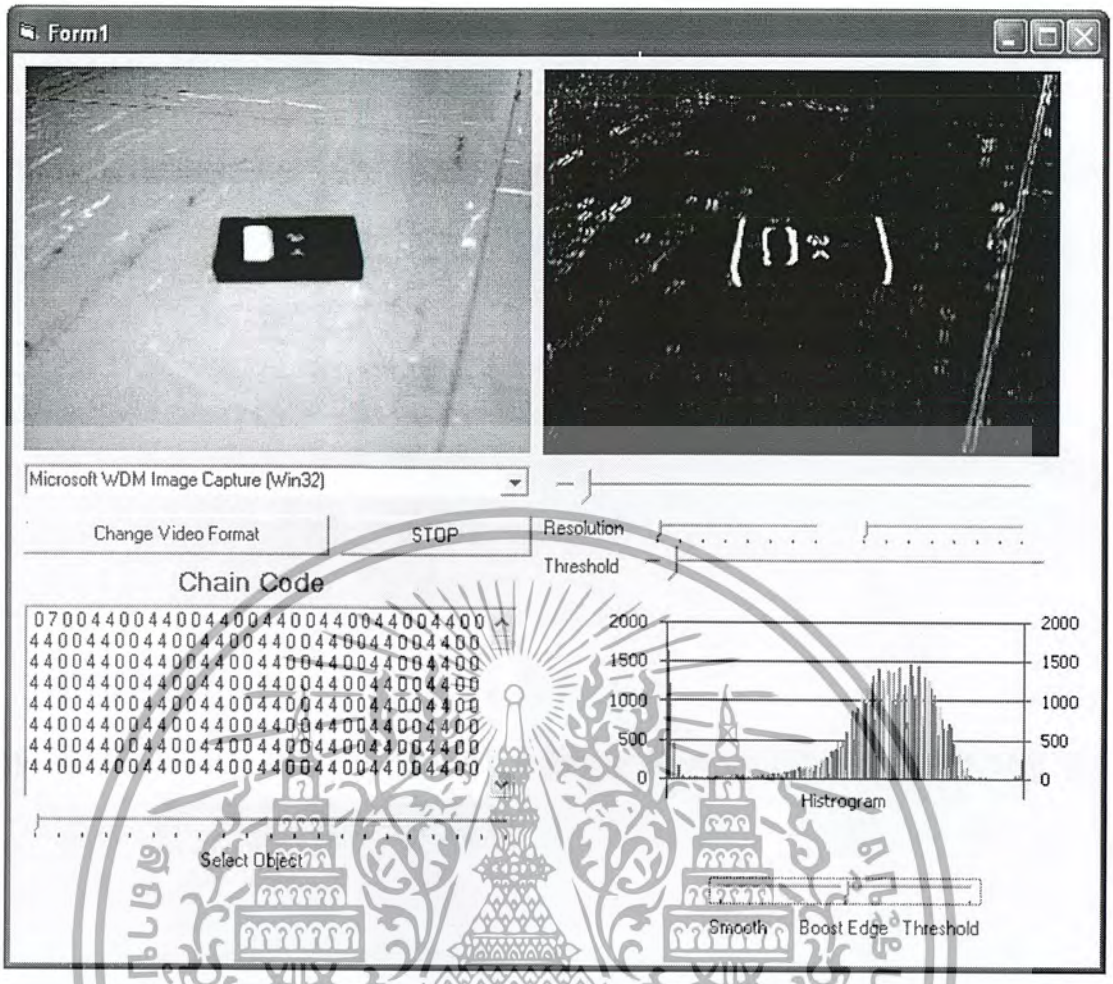


รูปที่ 4.6

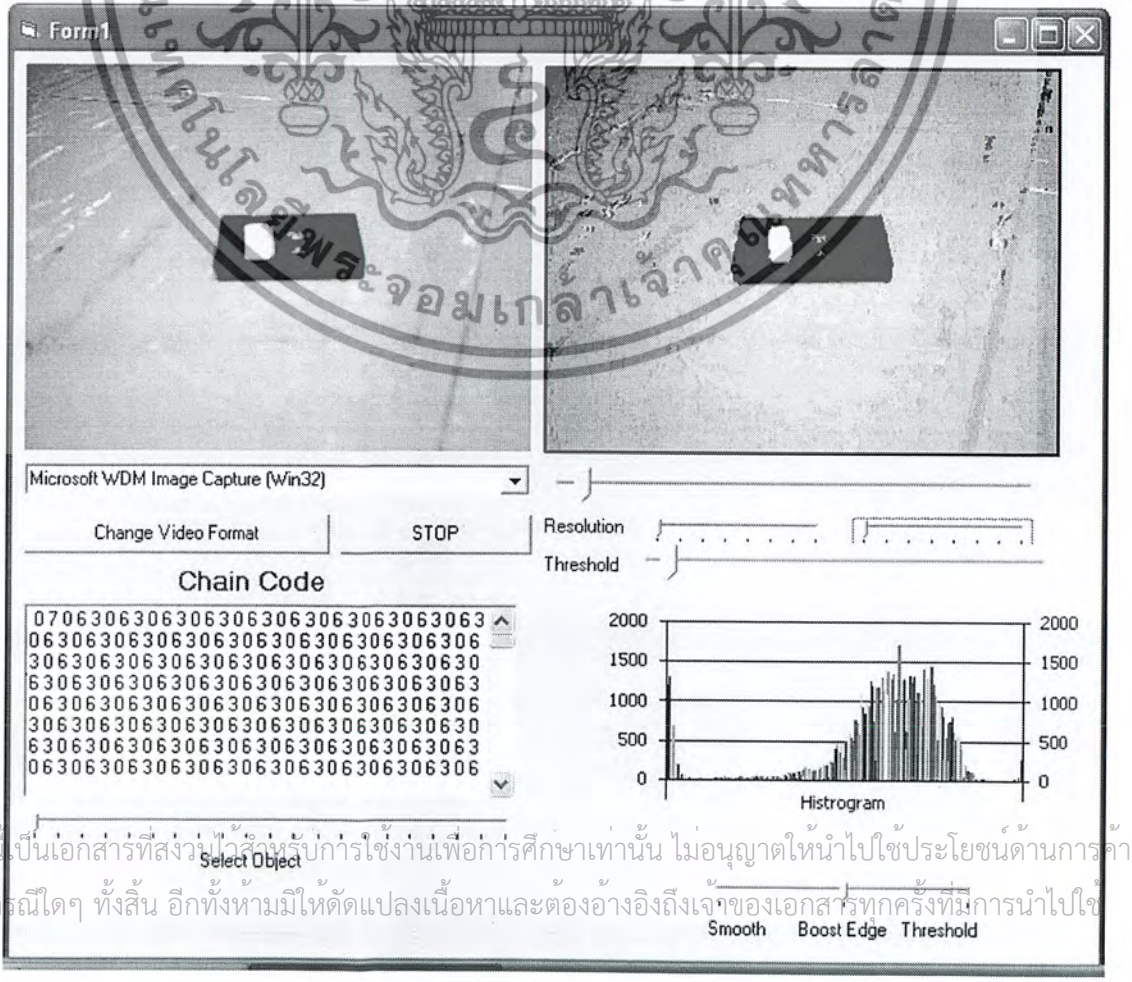


รูปที่ 4.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

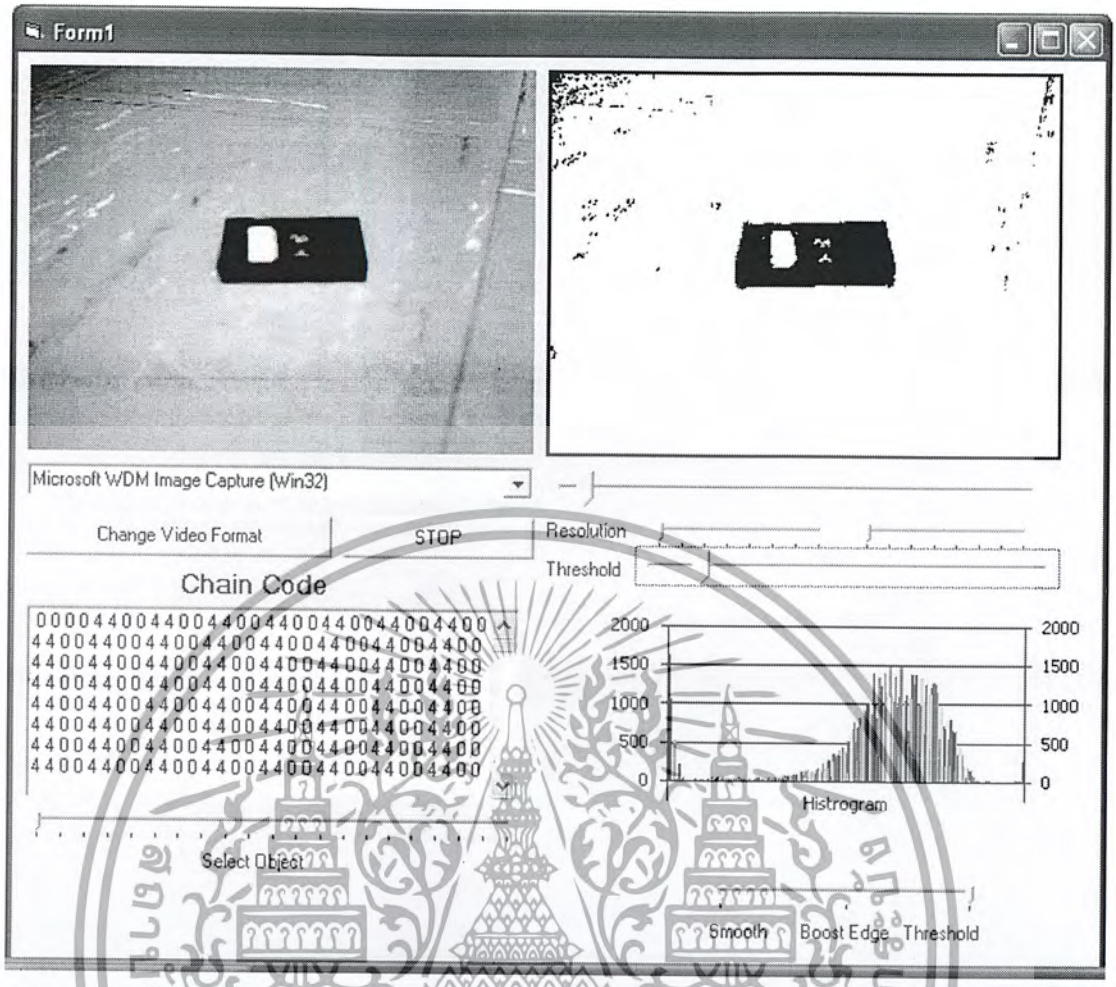


รูปที่ 4.8

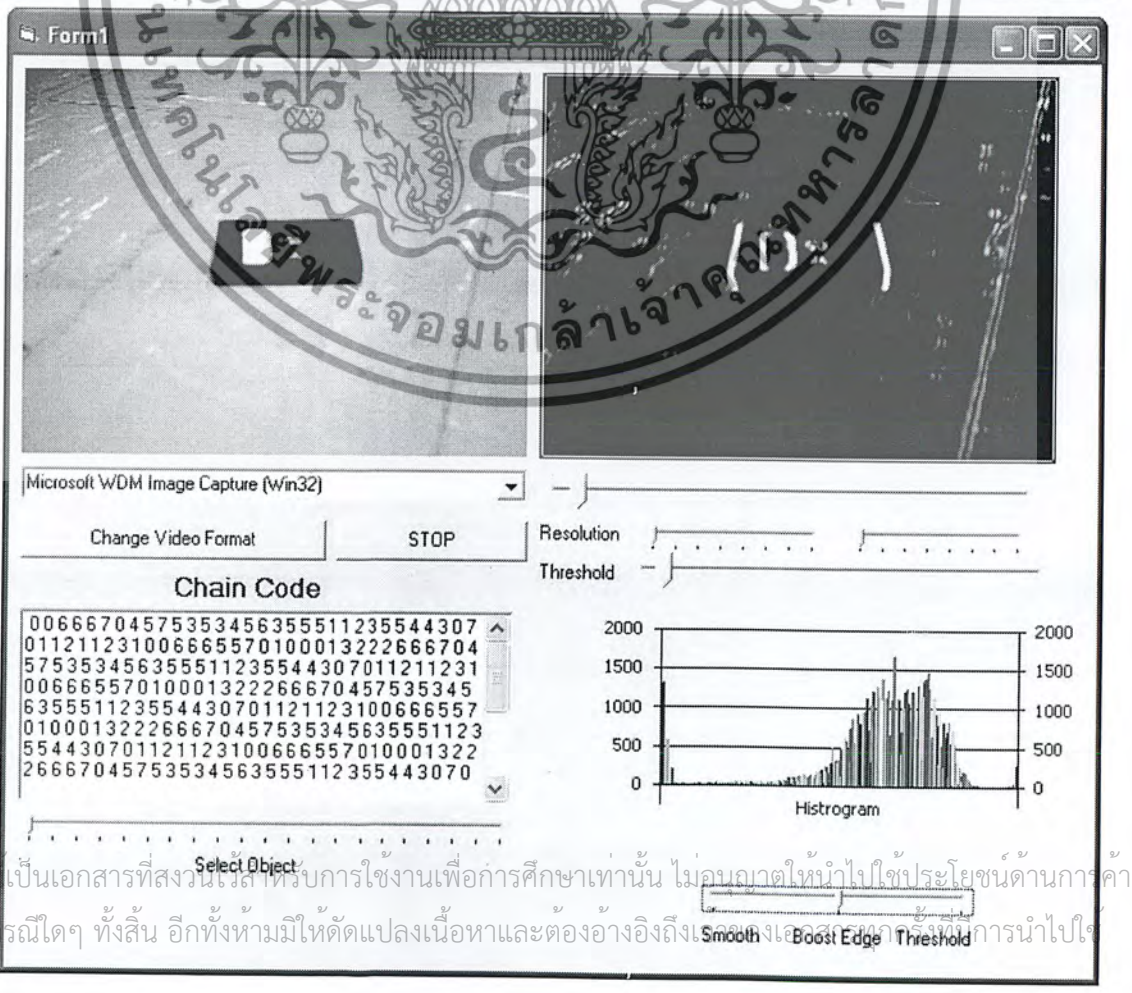


รูปที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

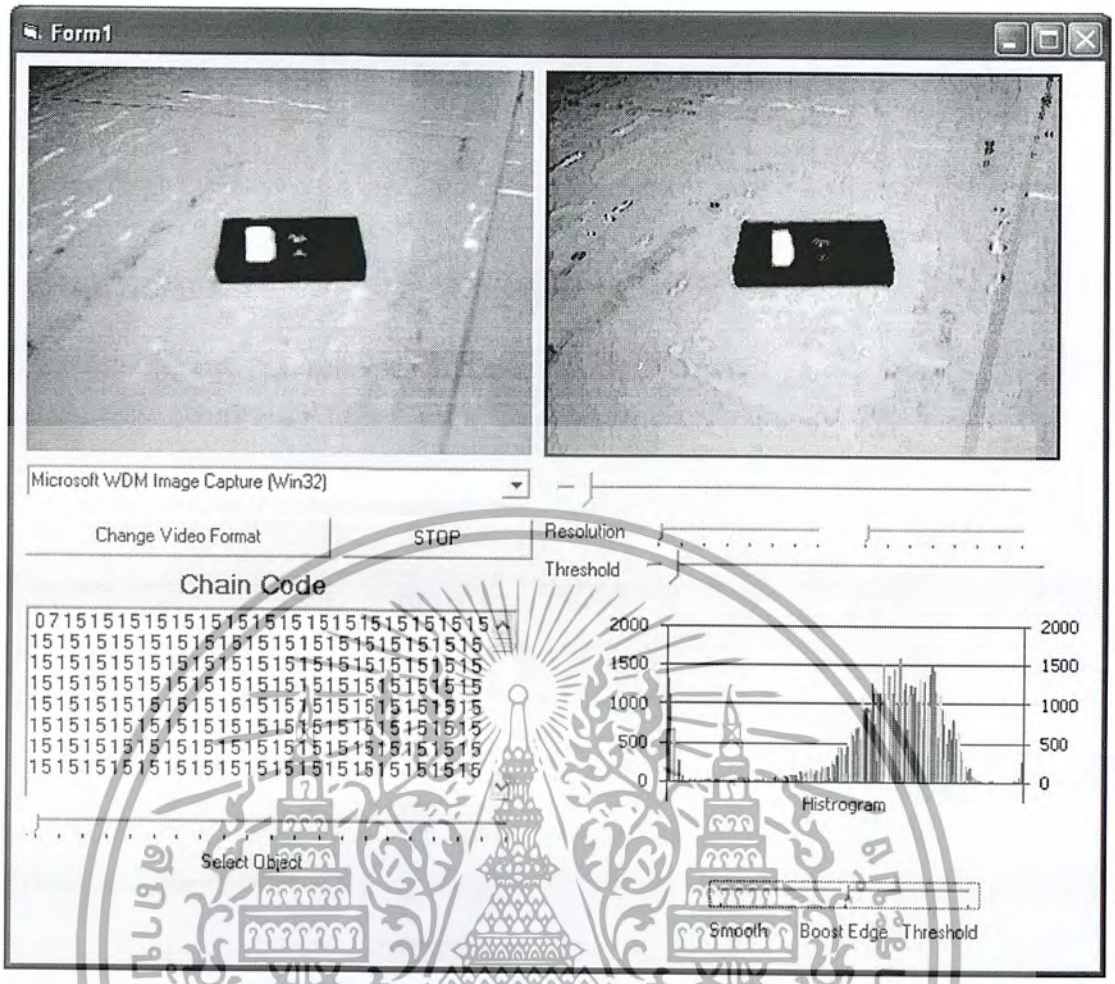


รูปที่ 4.10

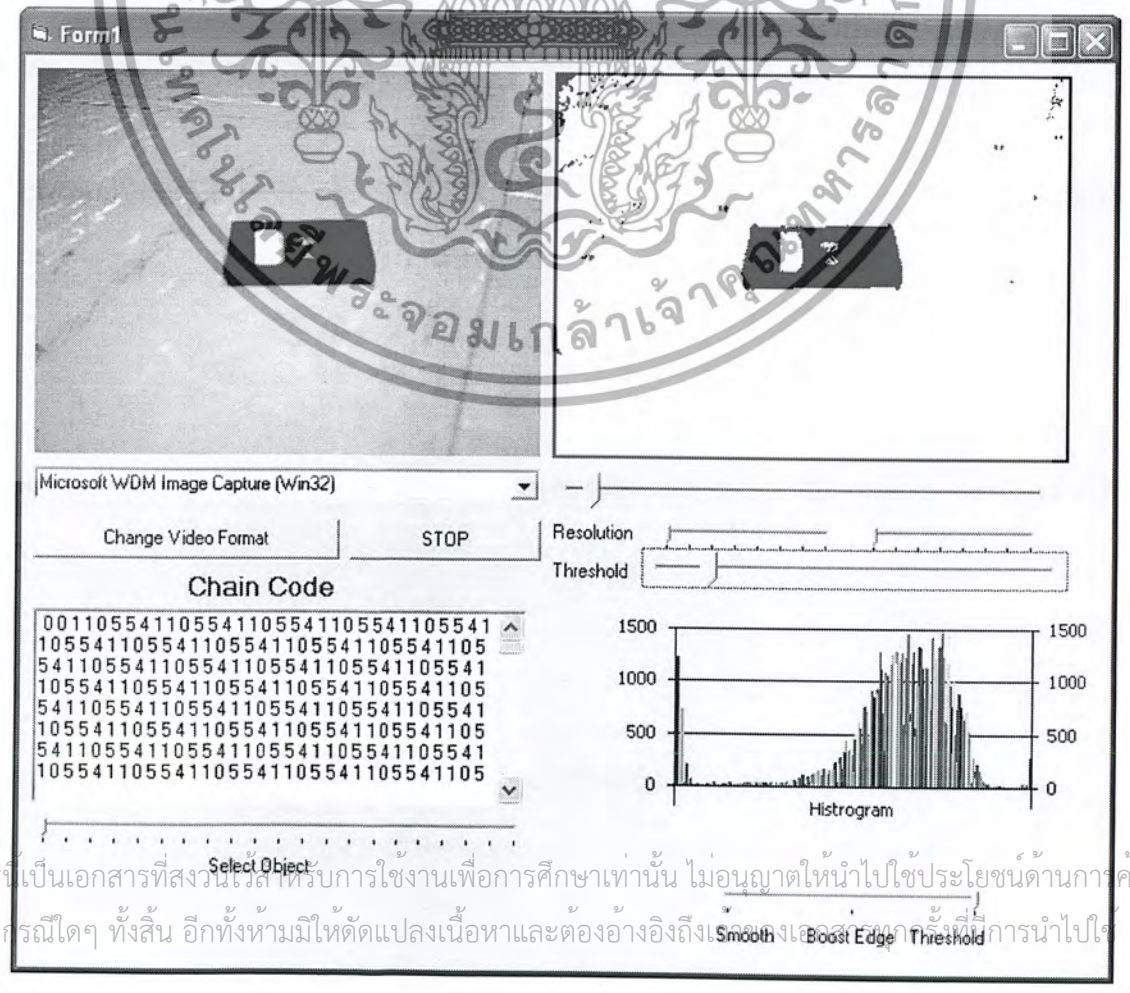


รูปที่ 4.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

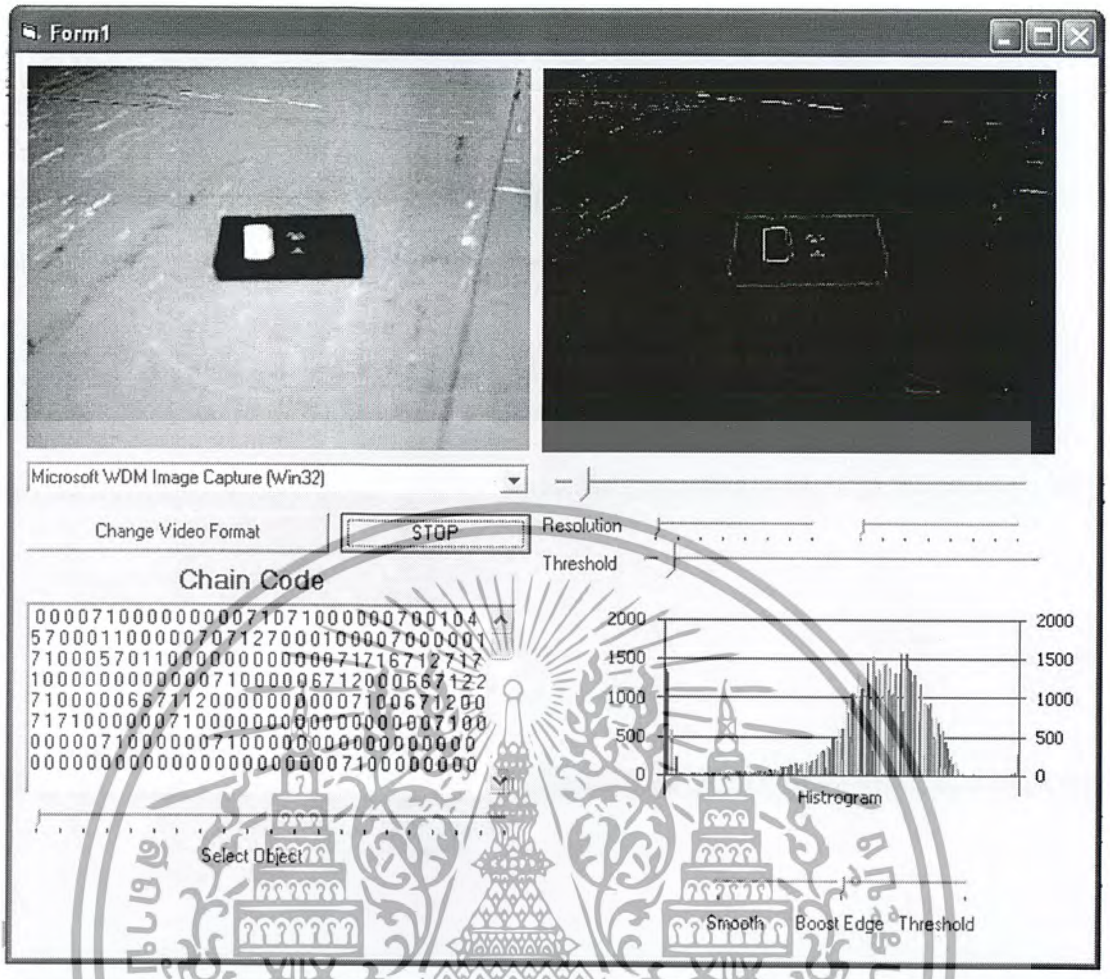


รูปที่ 4.12

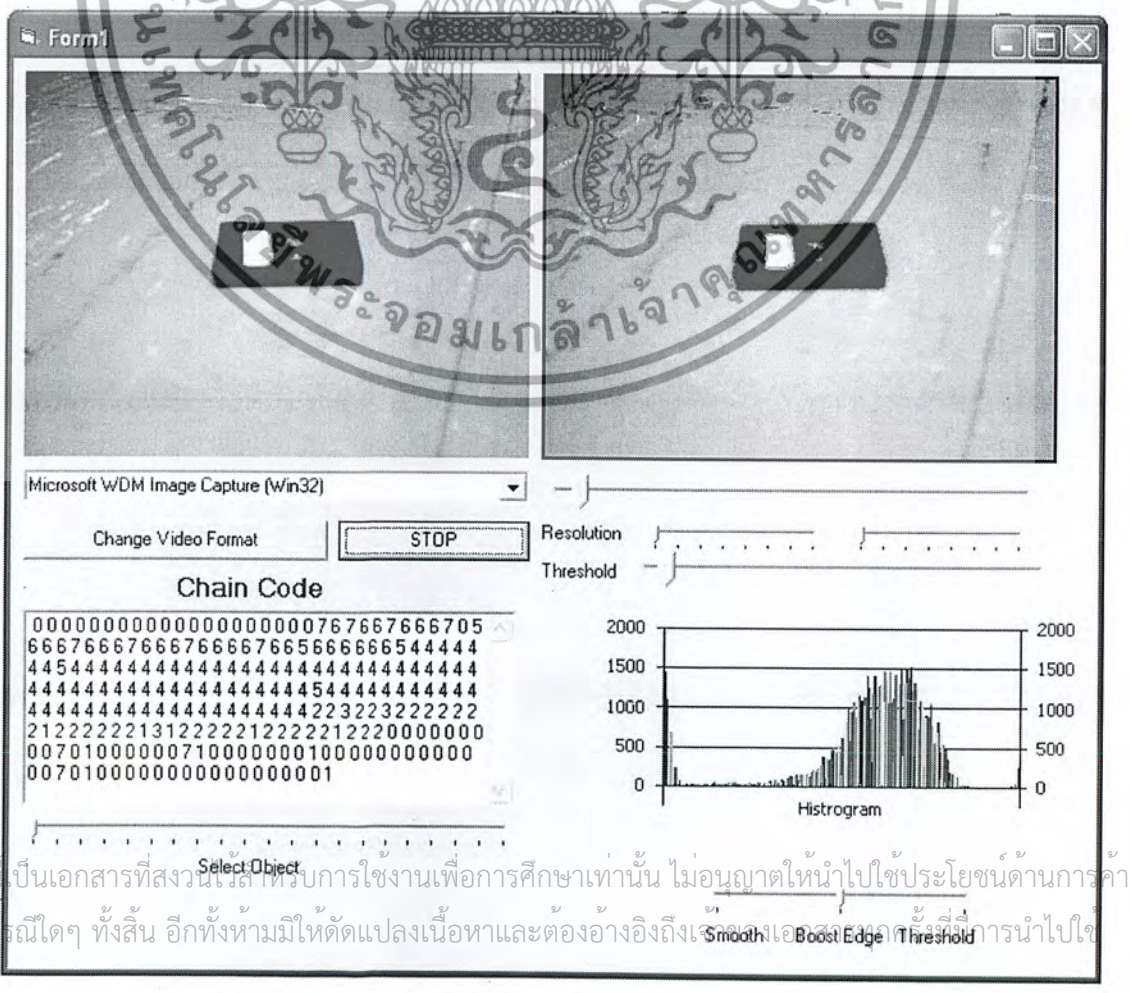


รูปที่ 4.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

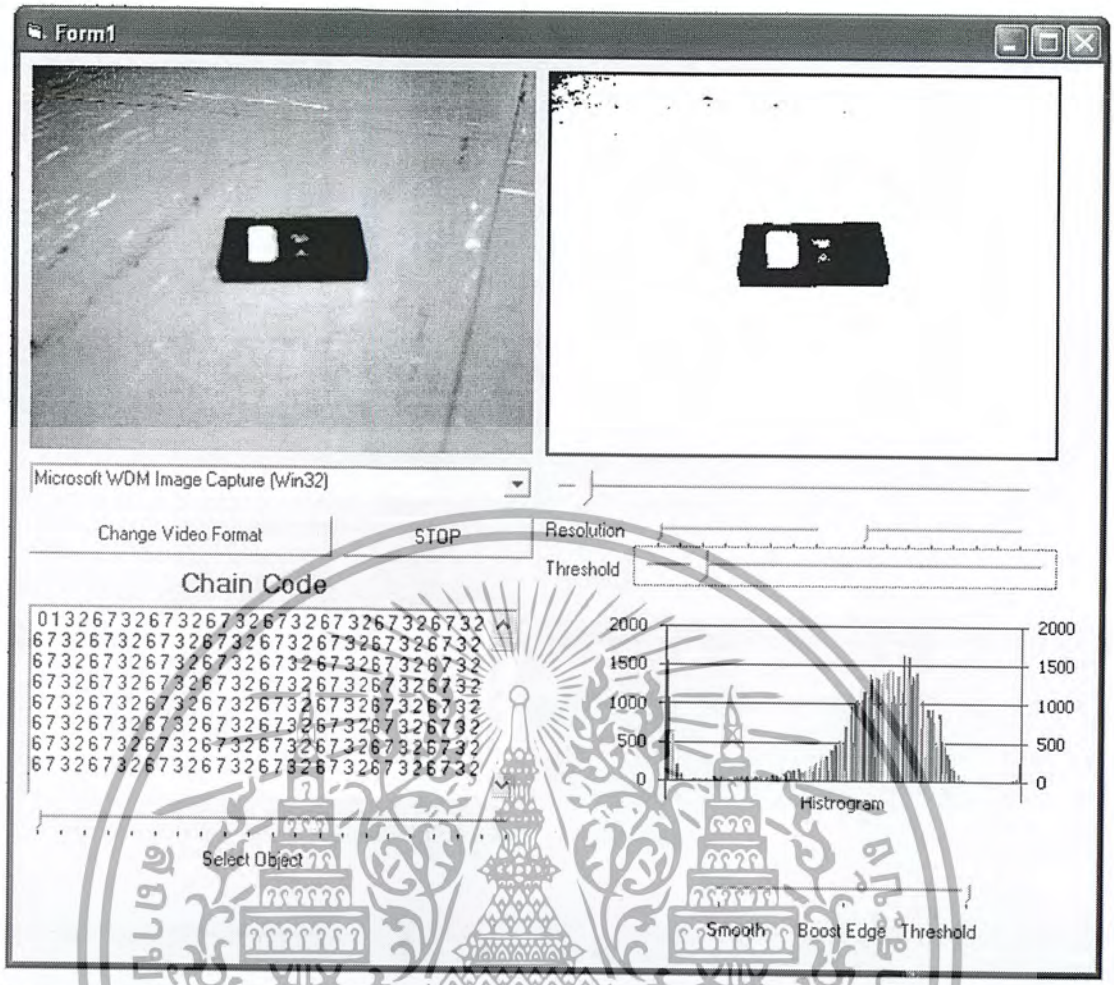


รูปที่ 4.14

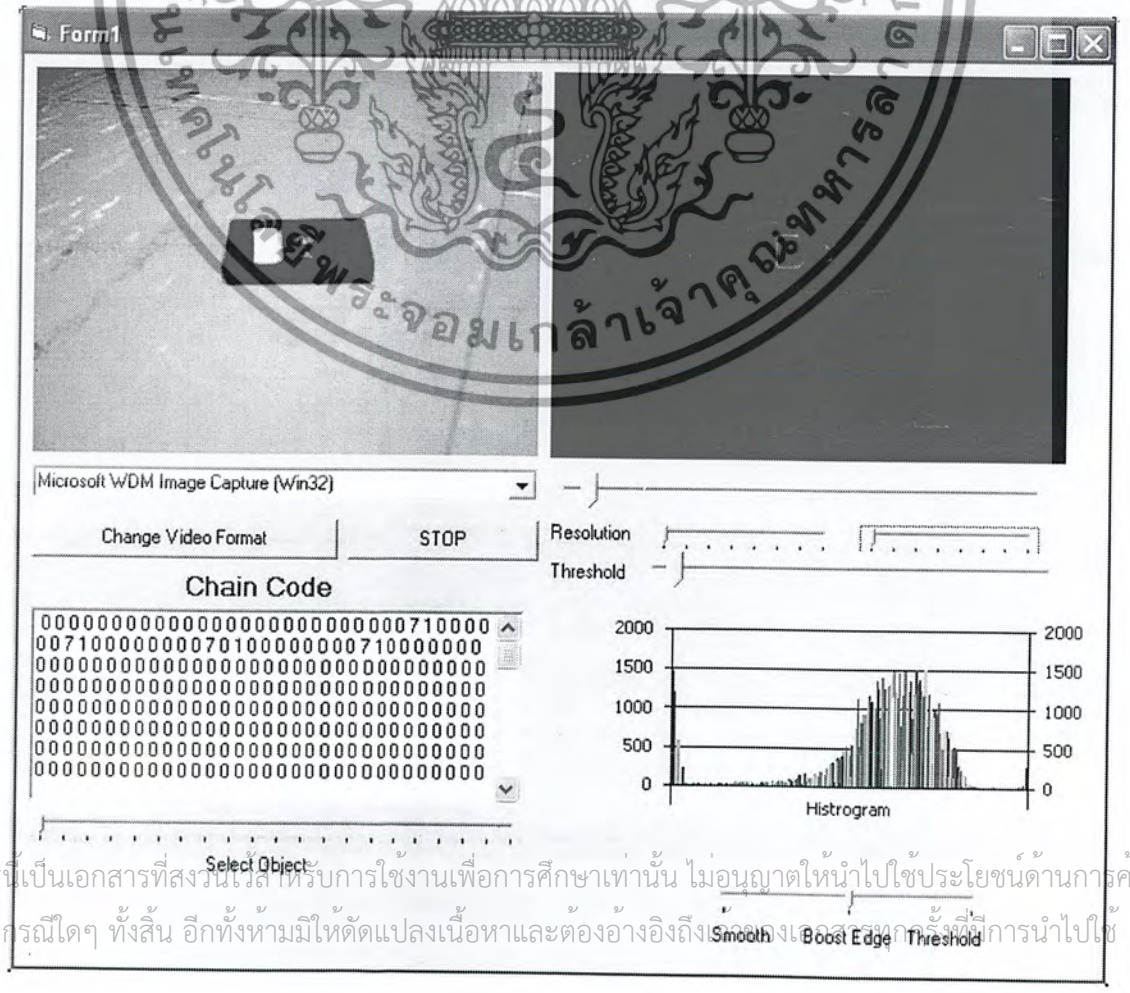


รูปที่ 4.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

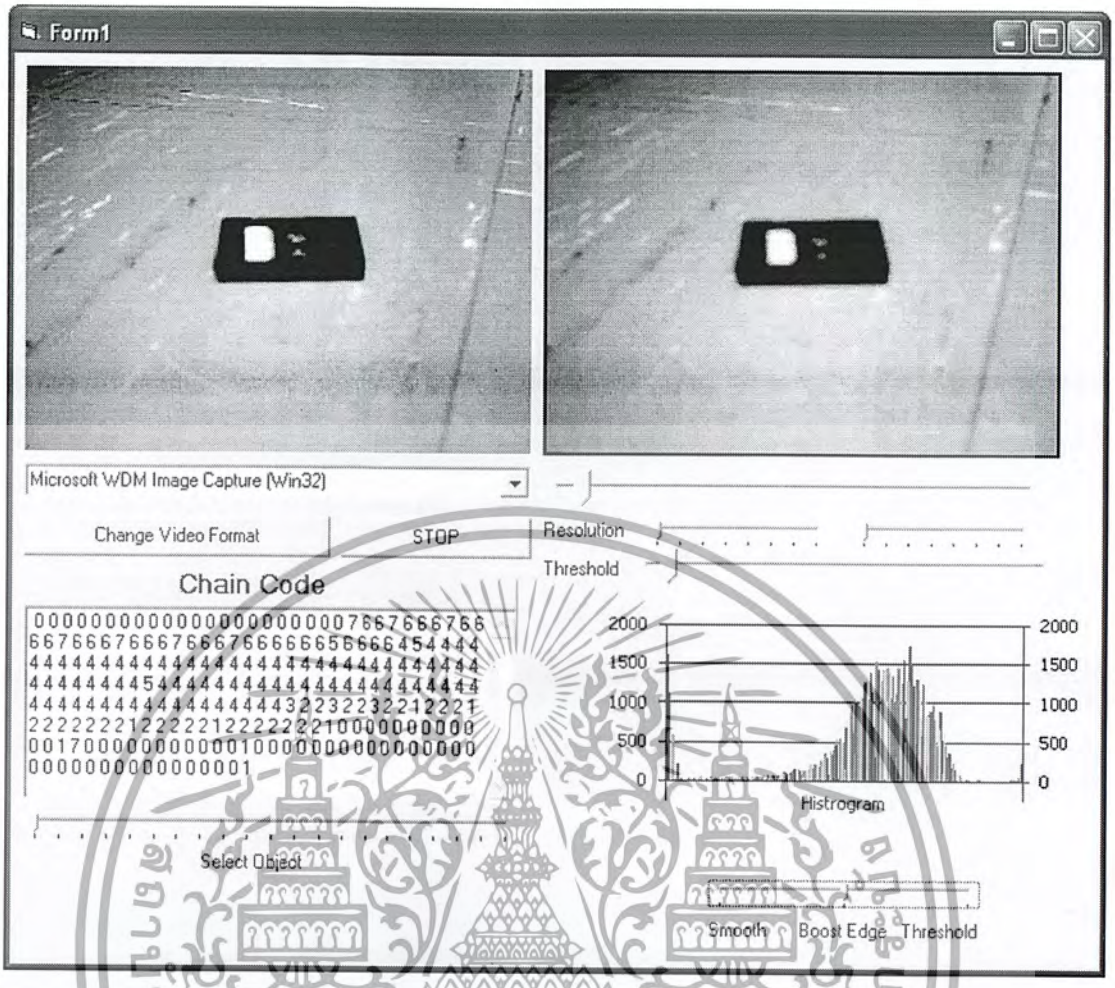


รูปที่ 4.16

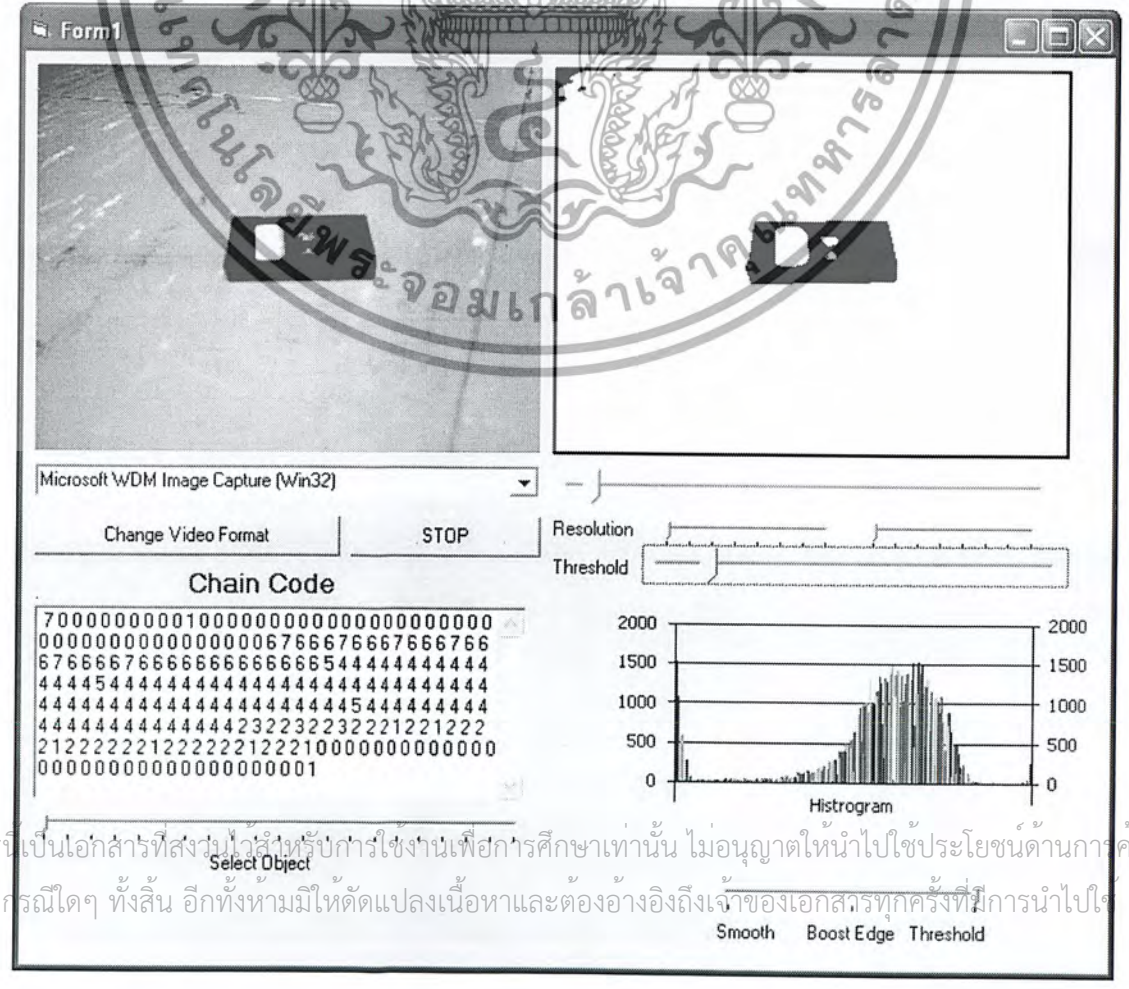


รูปที่ 4.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18



รูปที่ 4.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองของการทำตามกระบวนการหา Contour

ซึ่งการหา Contour นั้นเป็นการติดตามรอยขอบภาพ ซึ่งทำให้รู้ว่าภาพวัตถุนั้นอยู่ในตำแหน่งไหน หรือว่าอยู่ในพิกัดใด ซึ่งผลการทดลองที่ได้เป็นดังภาพดังต่อไปนี้

ในรูปที่ 4.19 เป็นภาพที่ผ่านขั้นตอนการ Smoothing และนำไป Convolution กับ Mask ของ Laplacian แบบ Light Edge จนได้ภาพที่เป็นขอบ แล้วนำไป AND กับภาพ Grey Scale เดิม จากนั้นนำไปผ่านการ Thresholding

ในรูปที่ 4.20 เป็นการนำภาพที่ได้มาหา Contour ที่วิ่งไปตามขอบของภาพ โดยเส้นทางที่วิ่งไปจะถูกทำให้เป็นสีแดง เพื่อสังเกตผล และการหา Contour จะทำให้ได้ Chain Code ออกมาด้วย

ในรูปที่ 4.21 เป็นการแสดงภาพที่หาค่า Chain Code ได้แล้ว ทำให้รู้ค่ามากสุดกับค่าน้อยที่สุดในแกน X และ Y ซึ่งสามารถนำเอาข้อมูลนั้นมาตีกรอบบริเวณของวัตถุได้

ในรูปที่ 4.22 เป็นการนำเสนอภาพที่มีวัตถุ 2 ชิ้น ภายในภาพที่เราพิจารณา จะทำการตรวจจับวัตถุทั้งหมดที่อยู่ในภาพ

ในรูปที่ 4.23 เป็นการนำเสนอภาพที่มีวัตถุ 3 ชิ้น ภายในภาพที่เราพิจารณา

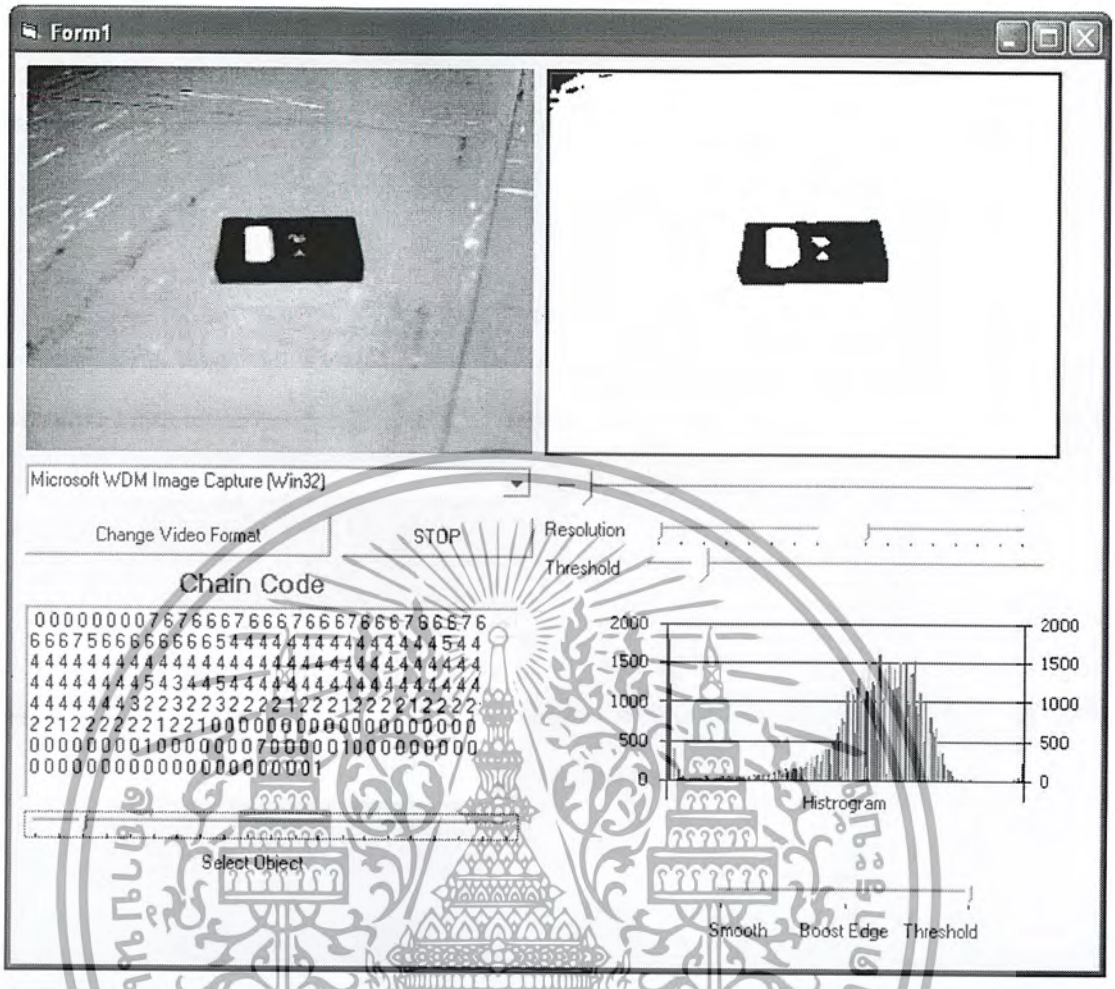
ในรูปที่ 4.24 เป็นการแสดงภาพที่วัตถุวางซ้อนกัน ผลที่ได้ก็คือวัตถุที่วางซ้อนทับกันนั้นจะถือเสมือนว่าเป็นวัตถุเพียงชิ้นเดียว

ในรูปที่ 4.25 เป็นการแสดงภาพที่หลุดขอบ

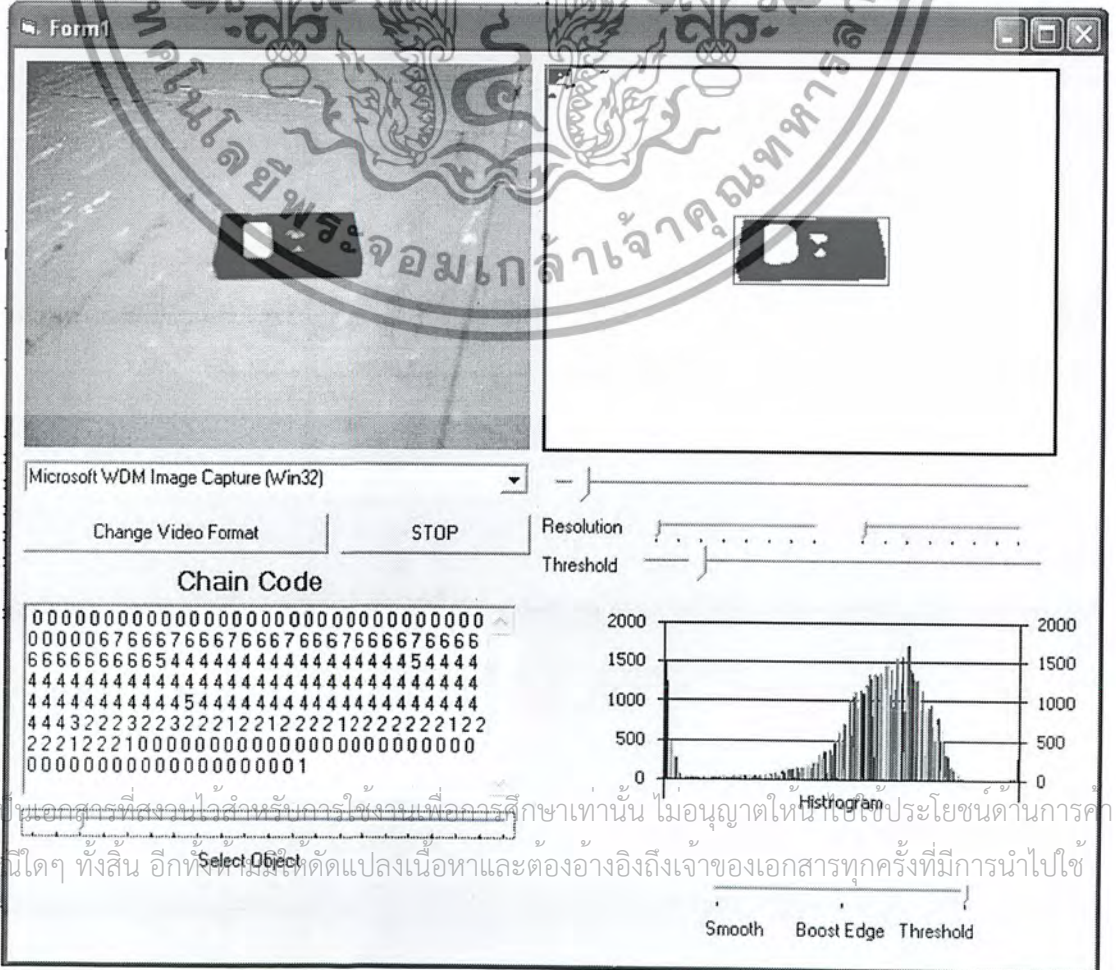
ในรูปที่ 4.26 เป็นภาพที่ใช้ Resolution ต่ำ เพื่อง่ายต่อการตรวจสอบความถูกต้องของ Chain Code ที่หาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.20

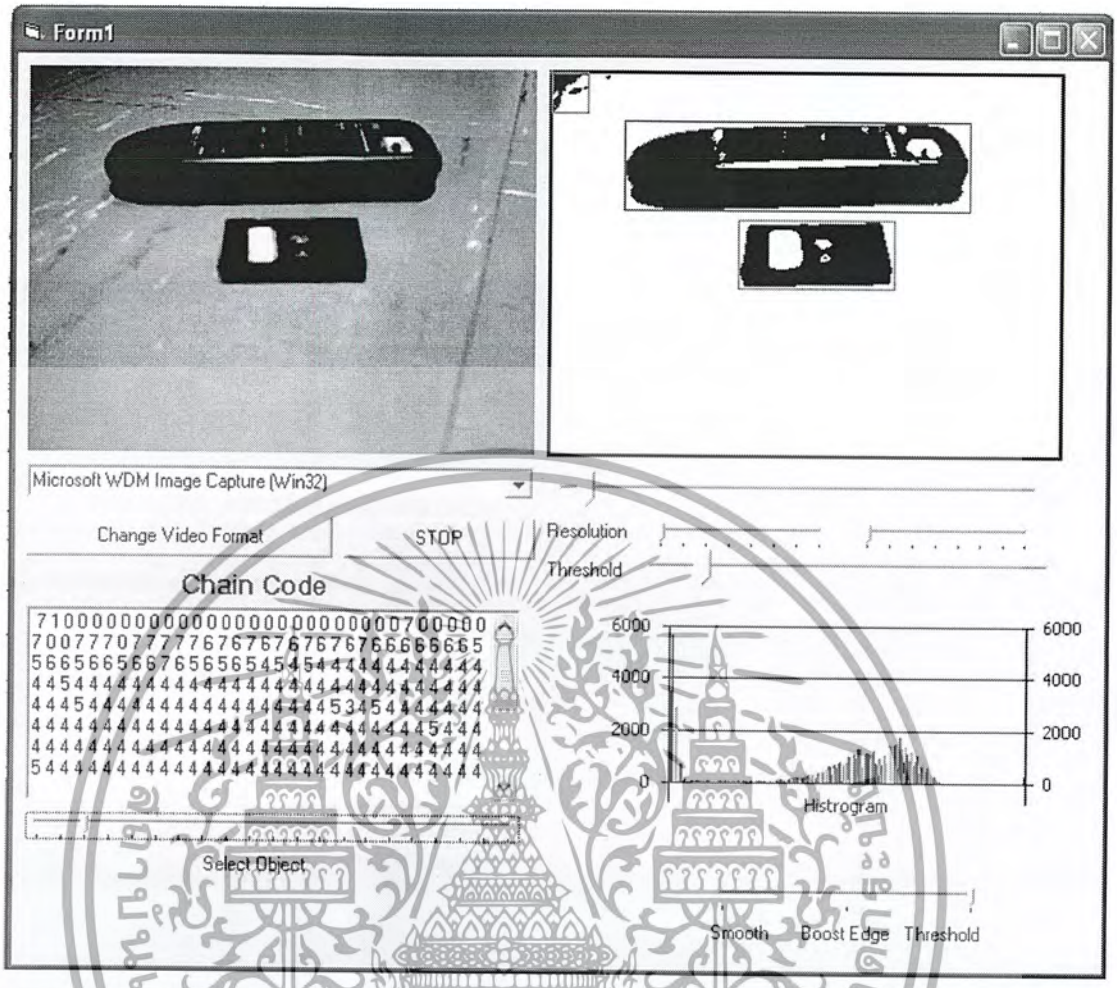


รูปที่ 4.21

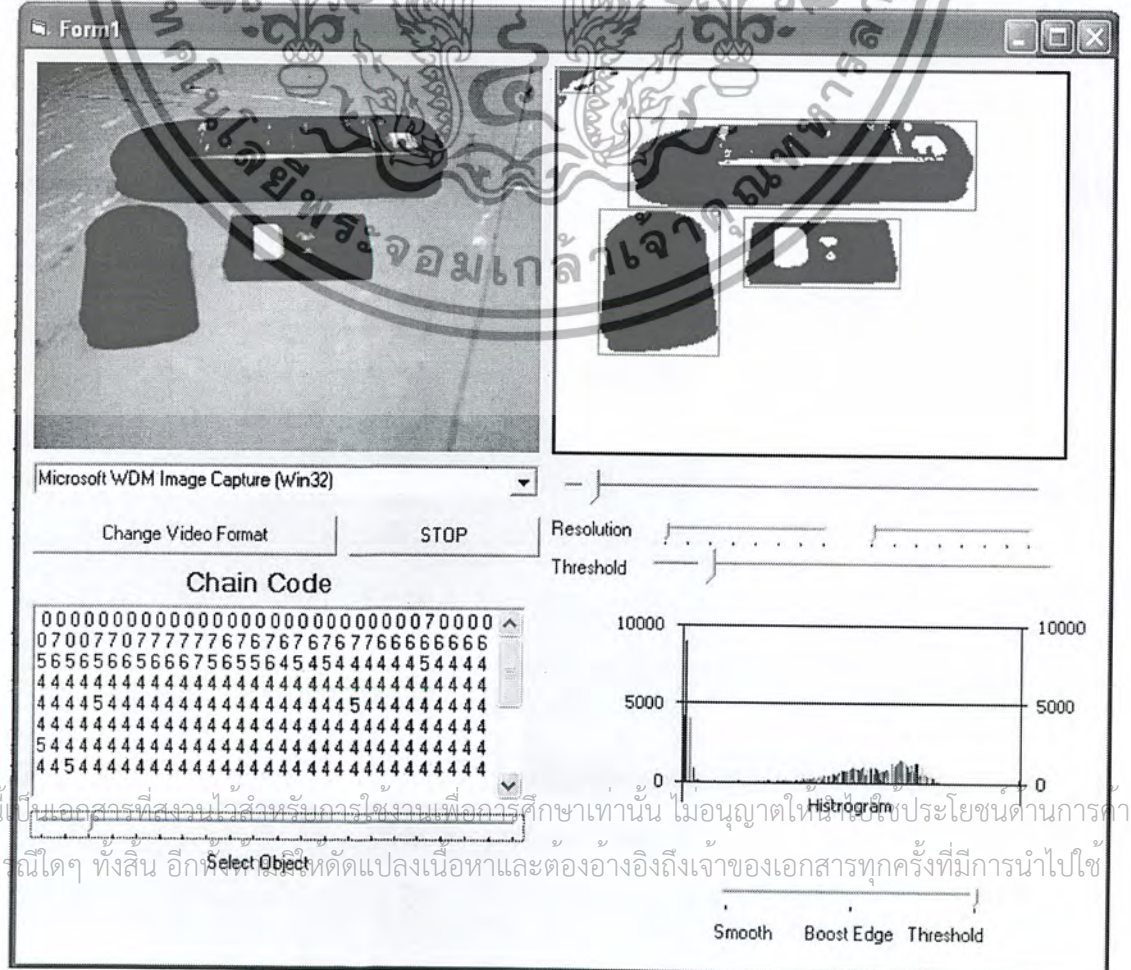


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.22

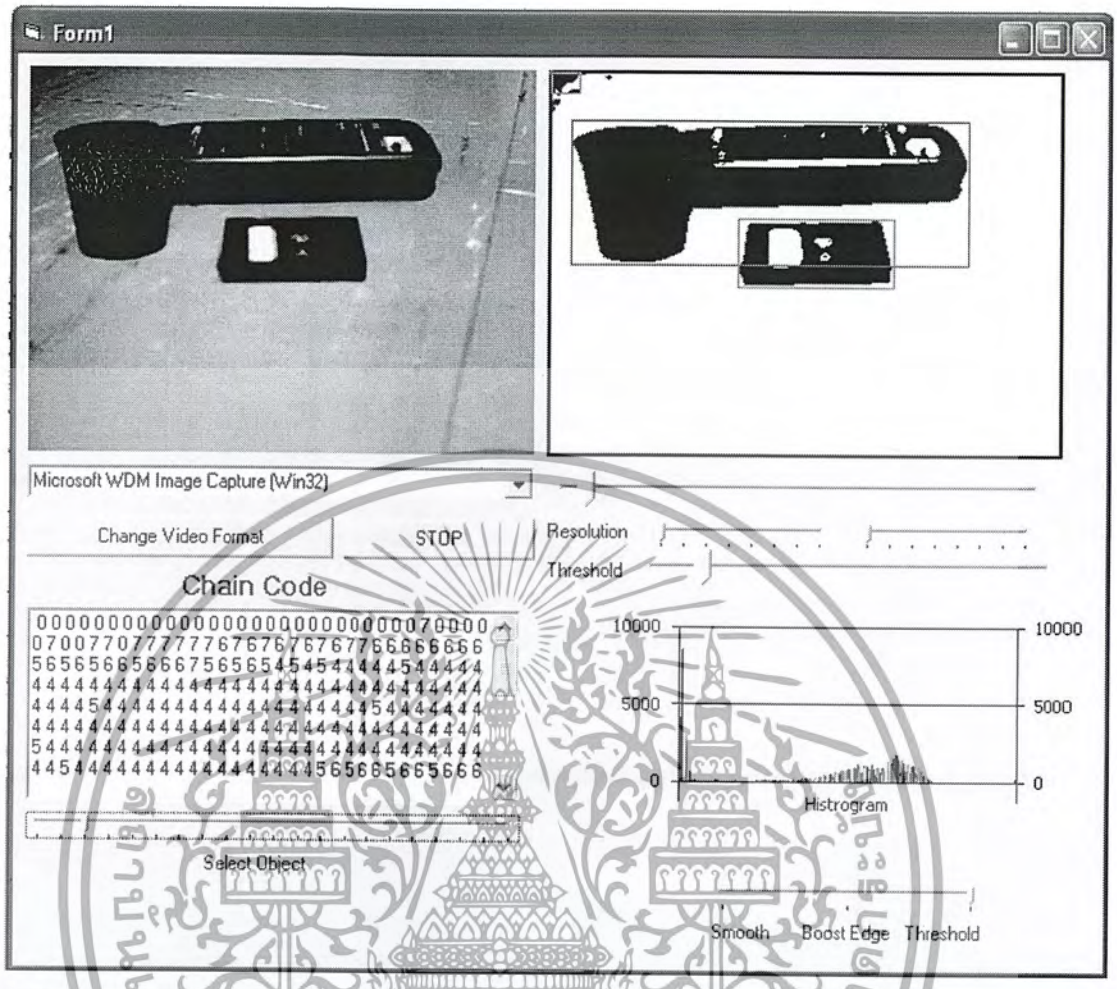


รูปที่ 4.23

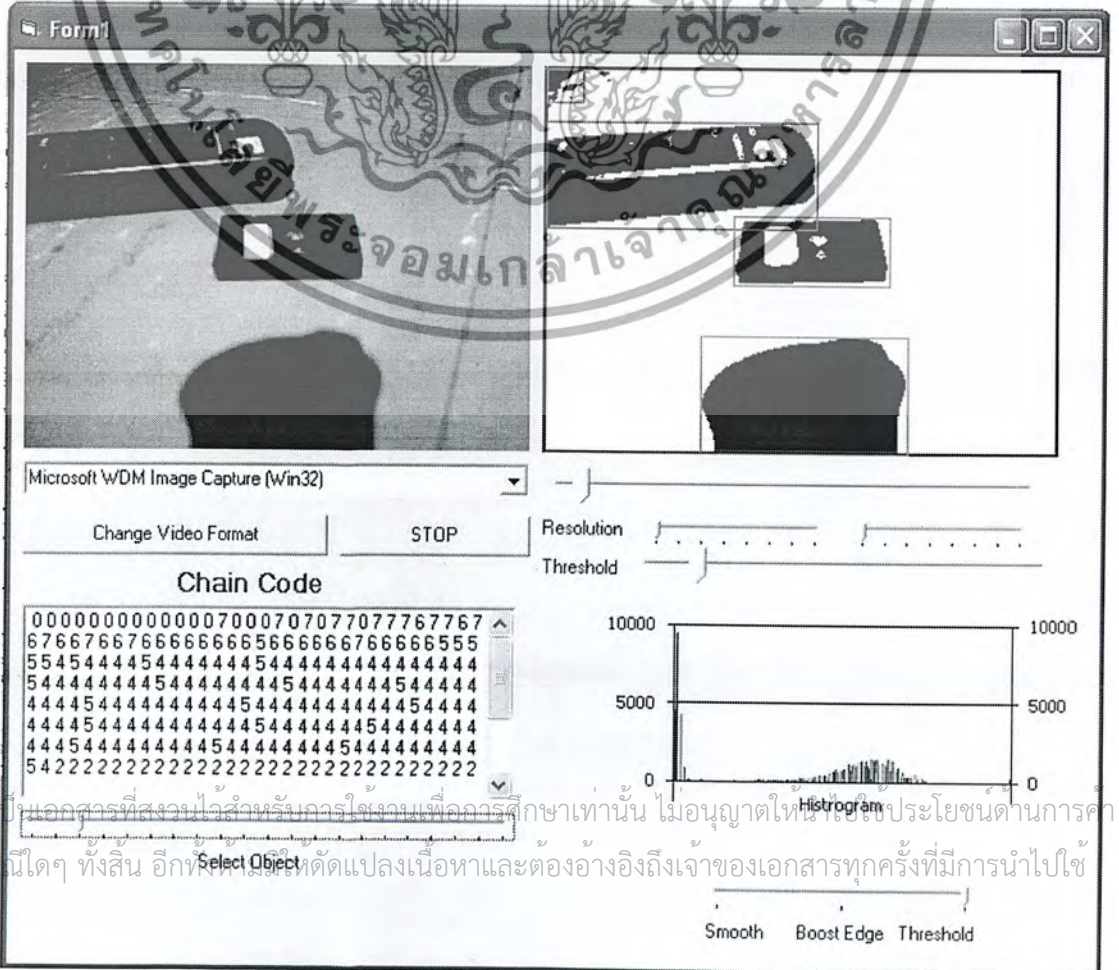


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.24

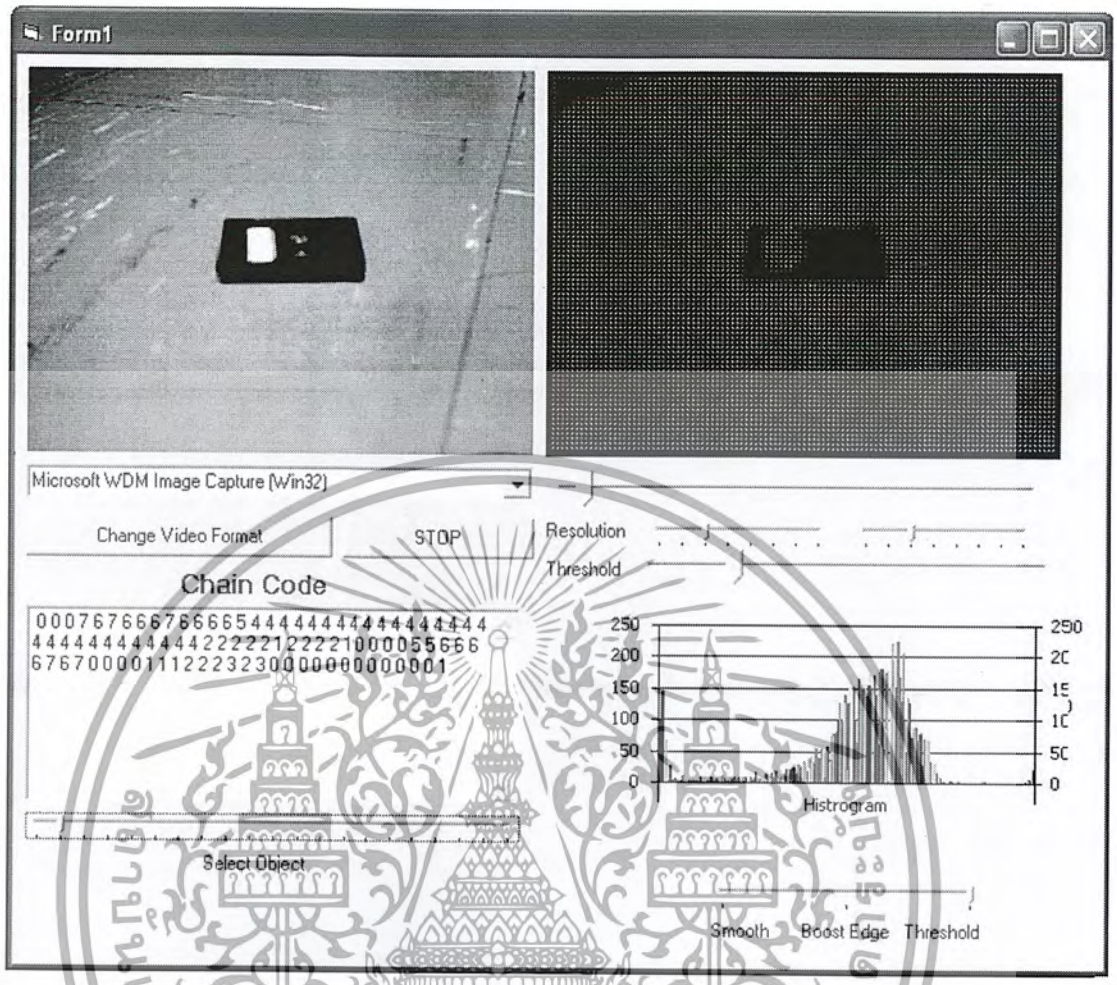


รูปที่ 4.25



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.26



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองจากการทำการจดจำวัตถุโดยขั้นตอนการ Rotation

เป็นการนำภาพวัตถุที่จับได้จากกล้องดิจิตอลมาทำการหมุนเพื่อเก็บไว้เป็นข้อมูลสำหรับเปรียบเทียบภาพที่จับมาได้ใหม่ ว่าเป็นภาพที่เหมือนกับภาพตัวอย่างที่ได้ทำการจดจำไว้ โดยได้อธิบายขั้นตอนการ Rotation ไว้ในบทที่ 3

รูปที่ 4.27 เป็นรูปจากกล้องดิจิตอลที่จับได้จริงจากวัตถุ

รูปที่ 4.28 เป็นรูปแสดงการ Rotation ภาพวัตถุที่จับได้จากกล้องดิจิตอล

รูปที่ 4.29 เป็นรูปแสดงการ Dilation ภาพวัตถุที่จับได้จากกล้องดิจิตอล

รูปที่ 4.30 เป็นรูปแสดงจุดที่เพิ่มจากภาพวัตถุจริงหลังจากการทำ Dilation

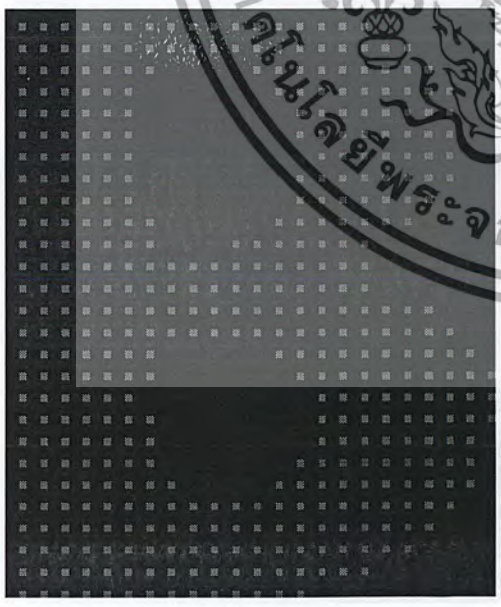
รูปที่ 4.31 เป็นรูปแสดงการทำ Filter ในกรณีที่จุดข้างเคียงเป็น 1 มากกว่า 3 จุด

รูปที่ 4.32 เป็นรูปแสดงจุดที่เพิ่มจากการทำ Filter ในกรณีที่จุดข้างเคียงเป็น 1 มากกว่า 3 จุด

รูปที่ 4.33 เป็นรูปแสดงการทำ Filter ในกรณีที่จุดข้างเคียงเป็น 1 มากกว่า 4 จุด

รูปที่ 4.34 เป็นรูปแสดงจุดที่เพิ่มจากการทำ Filter ในกรณีที่จุดข้างเคียงเป็น 1 มากกว่า 4 จุด

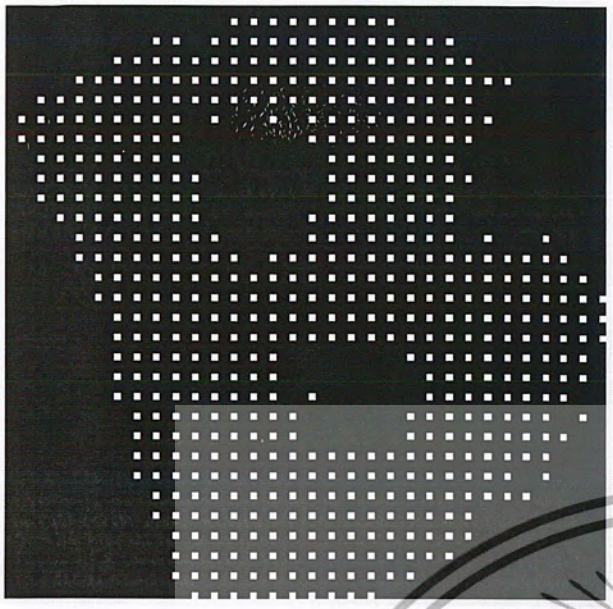
ส่วนรูปที่ 4.35 ถึงรูปที่ 4.42 เป็นรูปที่แสดงผลการทดลองเช่นเดียวกับรูปที่ 4.27 ถึง 4.34 เรียงตามลำดับ แต่เป็นวัตถุคนละชิ้นกัน



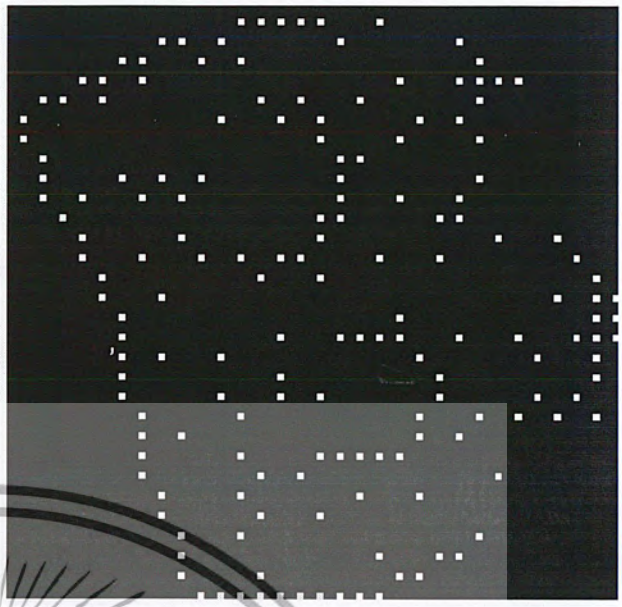
รูปที่ 4.27

รูปที่ 4.28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.29

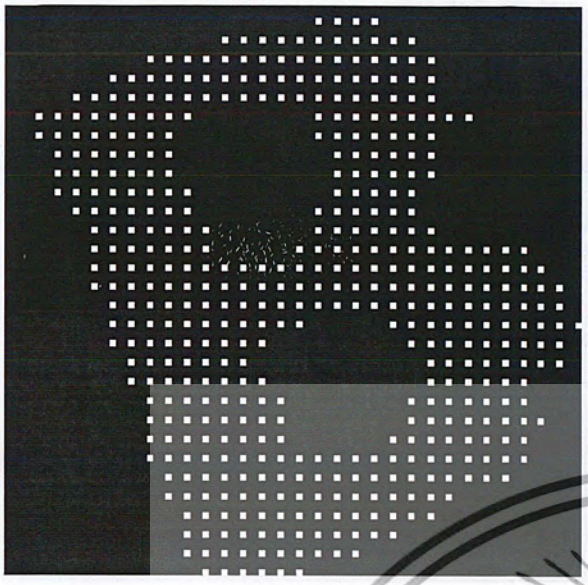


รูปที่ 4.30

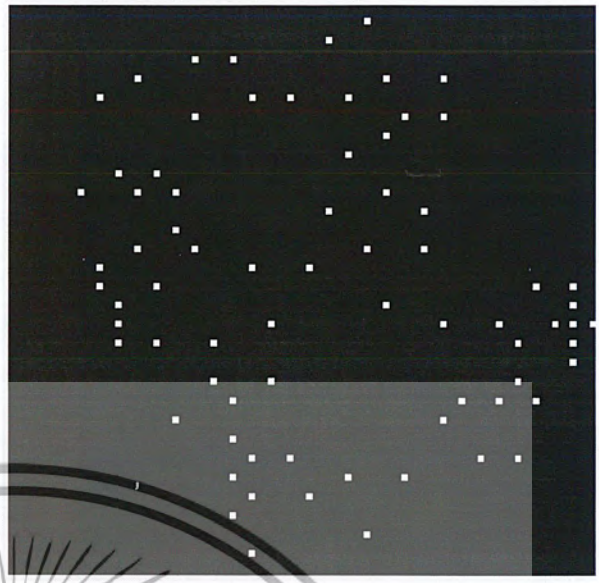
รูปที่ 4.31

รูปที่ 4.32

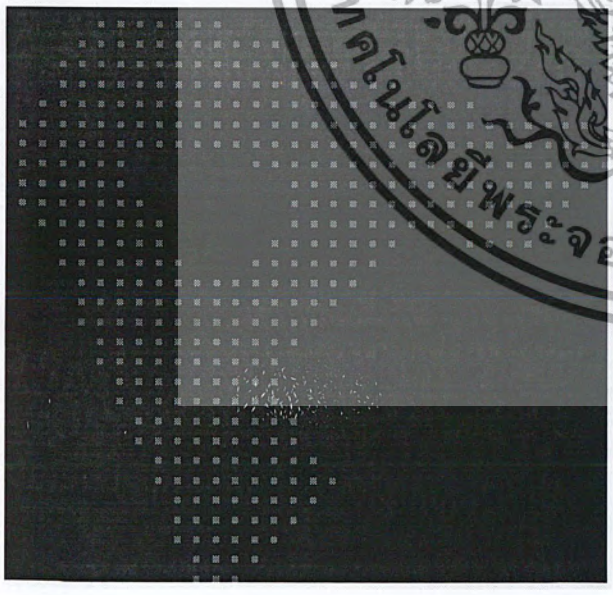
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



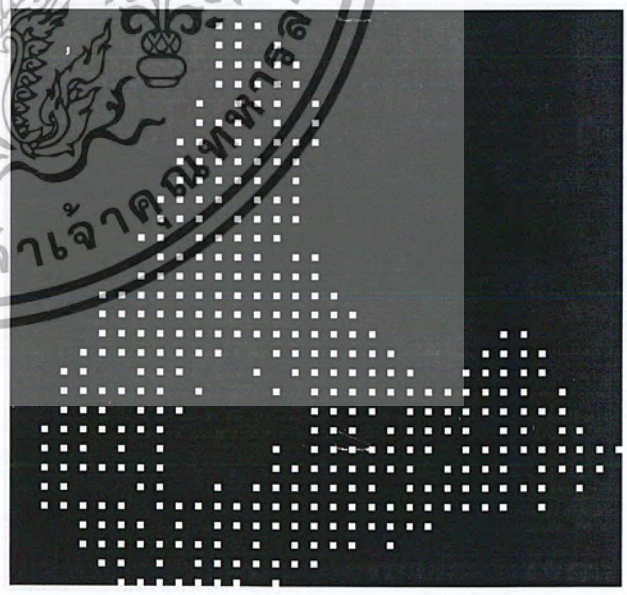
รูปที่ 4.33



รูปที่ 4.34

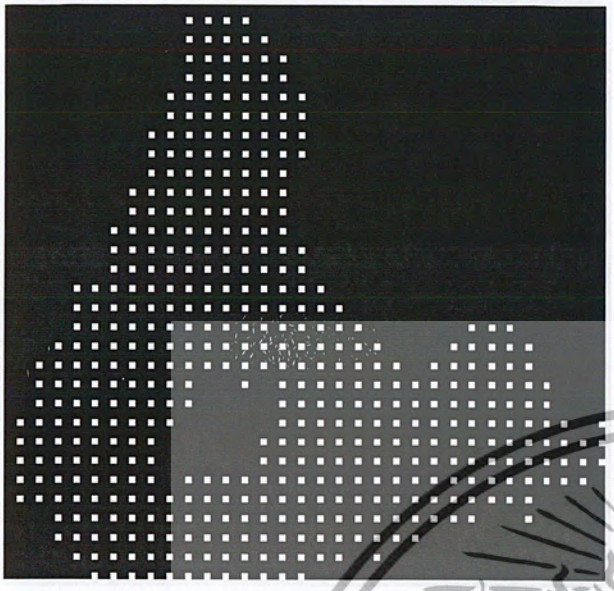


รูปที่ 4.35

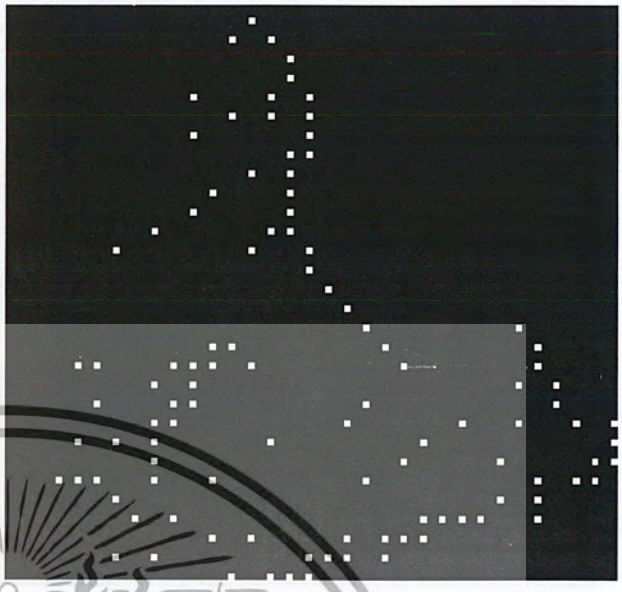


รูปที่ 4.36

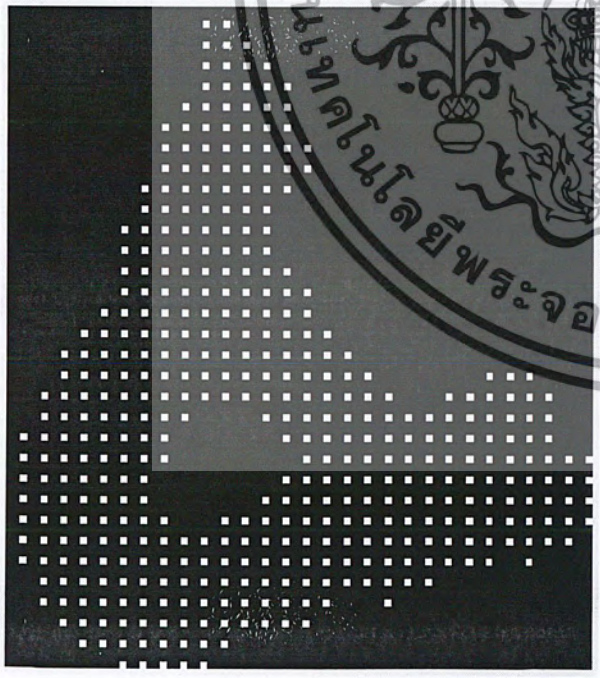
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



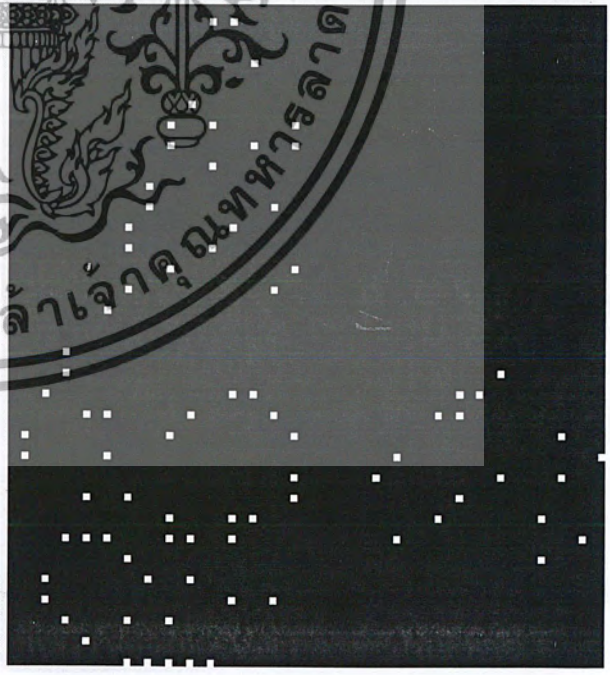
รูปที่ 4.37



รูปที่ 4.38

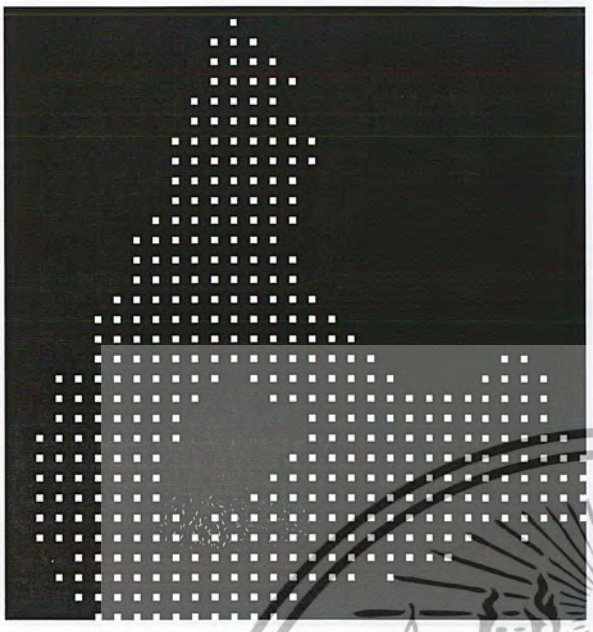


รูปที่ 4.39

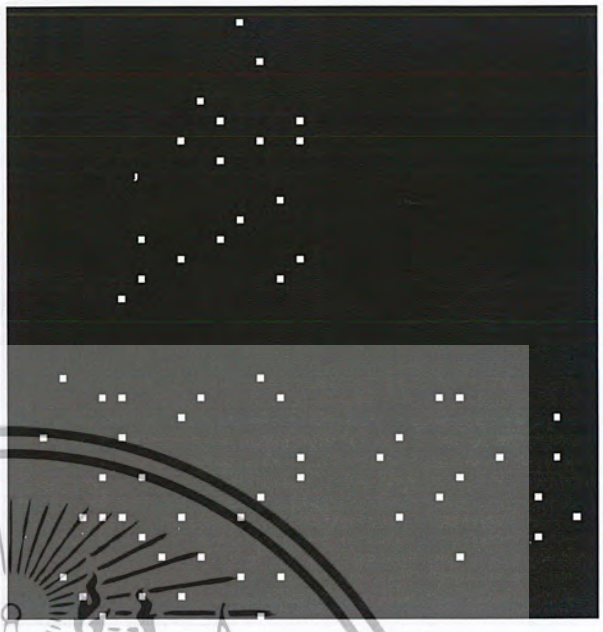


รูปที่ 4.40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.41



รูปที่ 4.42

ผลการทดลองจากการ Rotation วัตถุต่าง ๆ ชนิดกัน

ผลของการ Rotation วัตถุ B ดังรูปที่4.27 ถึง 4.34 เมื่อนำมาทำการคิดเป็น % ความผิดพลาดจากการแก้ไขเรื่องของการหมุนวัตถุแล้วเกิดข้อบกพร่อง ด้วยกระบวนการ Dilation , Filter 3 จุด และ Filter 4 จุด ผลที่ได้เป็นดังตารางข้างล่างนี้

ลักษณะของภาพวัตถุ	ค่า % ความผิดพลาดที่เปลี่ยนไปเมื่อเทียบกับภาพปกติ
ภาพปกติเมื่อทำการ Rotation (หมุนวัตถุ)	8.803
ภาพที่หมุนแล้วผ่านการ Dilation	27.534
ภาพที่หมุนแล้วผ่านการทำ Filter 3 จุด	11.061
ภาพที่หมุนแล้วผ่านการทำ Filter 4 จุด	8.126

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทดลองกับวัตถุ A ผลที่ได้ดังรูปที่ 4.35 ถึง 4.42 แล้วนำมาคิดเป็น % ความผิดพลาดจากการแก้ไขข้อบกพร่องจากการหมุนวัตถุนั้น ผลที่ได้เป็นไปตามตารางข้างล่างนี้

ลักษณะของภาพวัตถุ	ค่า % ความผิดพลาดที่เปลี่ยนไปเมื่อเทียบกับภาพปกติ
ภาพปกติเมื่อทำการ Rotation (หมุนวัตถุ)	9.204
ภาพที่หมุนแล้วผ่านการ Dilation	16.667
ภาพที่หมุนแล้วผ่านการทำ Filter 3 จุด	7.462
ภาพที่หมุนแล้วผ่านการทำ Filter 4 จุด	2.487

ผลการทดลองจากการทำการจดจำวัตถุโดยขั้นตอนการ Resize

เป็นการลดหรือขยายขนาดของวัตถุให้มีขนาดเท่ากับขนาดของวัตถุที่ได้จดจำไว้ เพื่อสามารถนำไปใช้ในการเปรียบเทียบวัตถุได้ ซึ่งในโครงงานนี้ได้ทำการ Resize วัตถุหลาย ๆ ขนาดด้วยกัน เพื่อทำให้มีข้อมูลเพียงพอที่จะสามารถนำมาใช้งานได้

รูปที่ 4.43 เป็นรูปภาพวัตถุปกติ

รูปที่ 4.44 เป็นรูปแสดงขนาดของภาพวัตถุ 0.4 เท่าของภาพปกติ

รูปที่ 4.45 เป็นรูปแสดงขนาดของภาพวัตถุ 0.6 เท่าของภาพปกติ

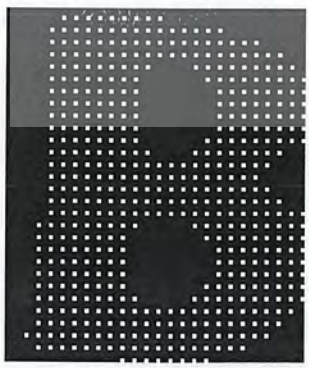
รูปที่ 4.46 เป็นรูปแสดงขนาดของภาพวัตถุ 0.8 เท่าของภาพปกติ

รูปที่ 4.47 เป็นรูปแสดงขนาดของภาพวัตถุ 1.4 เท่าของภาพปกติ

รูปที่ 4.48 เป็นรูปแสดงขนาดของภาพวัตถุ 1.8 เท่าของภาพปกติ

รูปที่ 4.49 เป็นรูปแสดงขนาดของภาพวัตถุ 2.2 เท่าของภาพปกติ

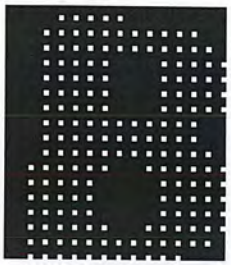
รูปที่ 4.50 เป็นรูปแสดงขนาดของภาพวัตถุ 2.6 เท่าของภาพปกติ



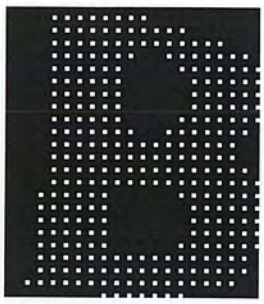
รูปที่ 4.43

รูปที่ 4.44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



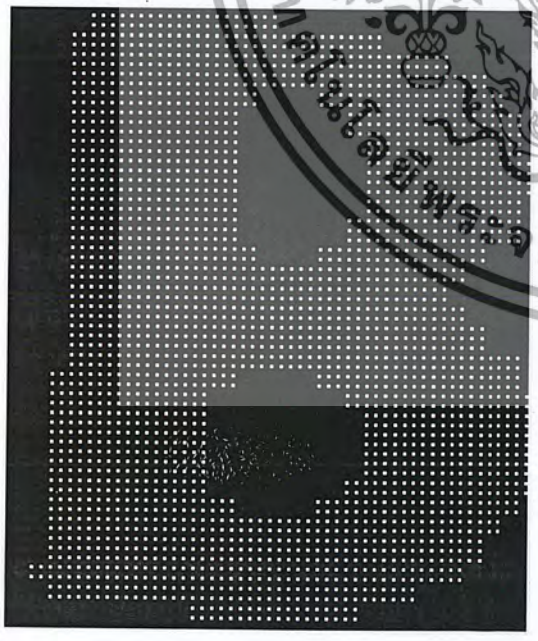
รูปที่ 4.45



รูปที่ 4.46

รูปที่ 4.47

รูปที่ 4.48



รูปที่ 4.49

รูปที่ 4.50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองจากการ Resize วัตถุต่าง ๆ ชนิดกัน

ผลของการ Resize วัตถุ B ดังรูปที่4.43 ถึง 4.50 เมื่อนำมาคิดเป็น % ความผิดพลาดที่เปลี่ยนไปจากภาพวัตถุที่จับได้จริงจากกล้อง ผลที่ได้เป็นไปตามตารางข้างล่างนี้

จำนวนเท่าของขนาดวัตถุเทียบกับขนาดของภาพวัตถุที่จับได้จริงจากกล้อง	ค่า % ความผิดพลาดที่เปลี่ยนไปจากภาพวัตถุที่จับได้จริงจากกล้อง
0.4	8.052
0.6	0.590
0.8	4.945
1.4	0.799
1.8	0.753
2.2	0.884
2.6	0.286

เมื่อทดลองกับวัตถุลักษณะต่าง ๆ ก็สามารถสังเกตได้ว่าความคลาดเคลื่อนของการ Resize วัตถุนั้น จะมีผลมากเมื่อวัตถุถูกลดขนาดให้มีขนาดเล็กมาก ๆ และกับภาพวัตถุที่นำมาทำการ Resize ที่มีขนาดเล็กด้วย เช่นการทำการ Resize วัตถุ A ในรูปที่ 4.35 จะได้ผลการทดลองดังนี้

จำนวนเท่าของขนาดวัตถุเทียบกับขนาดของภาพวัตถุที่จับได้จริงจากกล้อง	ค่า % ความผิดพลาดที่เปลี่ยนไปจากภาพวัตถุที่จับได้จริงจากกล้อง
0.4	11.504
0.6	0.7866
0.8	1.825
1.4	2.020
1.8	1.333
2.2	2.545
2.6	0.377
3.6	0.511
4.4	1.082

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปขั้นตอนการทำงานของการจดจำวัตถุ

เป็นการสรุปว่าขั้นตอนในการทำงานของการจดจำวัตถุนั้นได้ผ่านกระบวนการใดบ้าง

รูปที่ 4.51 รูปแสดงส่วนประกอบของโครงงานนี้ซึ่งประกอบไปด้วยการทำงานร่วมกันระหว่างกล้องดิจิตอลกับคอมพิวเตอร์

รูปที่ 4.52 รูปแสดงการจับภาพของกล้องเมื่อนำมาทำงานร่วมกันกับส่วนของโปรแกรมที่ใช้งานในการจดจำวัตถุ

รูปที่ 4.53 รูปแสดงการทำเทรชโฮลด์เพื่อให้สามารถจับภาพได้ใกล้เคียงกับภาพจริง

รูปที่ 4.54 รูปแสดงการเลือกวัตถุที่จะทำการจดจำ

รูปที่ 4.55 รูปแสดงการ get วัตถุจากภาพที่จับมาได้

รูปที่ 4.56 รูปแสดงการทำ Recognization เพื่อใช้ในการจดจำวัตถุ

รูปที่ 4.57 รูปแสดงการ Rotation วัตถุ เพื่อทำให้มีข้อมูลในการเปรียบเทียบระหว่างภาพที่จับมาได้ใหม่กับภาพวัตถุที่ได้ทำการจดจำไว้

รูปที่ 4.58 รูปแสดงการ Resize เพื่อช่วยให้วัตถุที่จับมาได้มีขนาดเท่ากับภาพที่จดจำไว้เพื่อง่ายต่อการเปรียบเทียบ

รูปที่ 4.59 รูปแสดงการจดจำวัตถุที่ถูกต้อง วัตถุที่ล้อมรอบด้วยกรอบสี่เหลี่ยมสีเขียวคือการแสดงผลว่าวัตถุชิ้นนั้นคือวัตถุที่ได้ทำการจดจำไว้

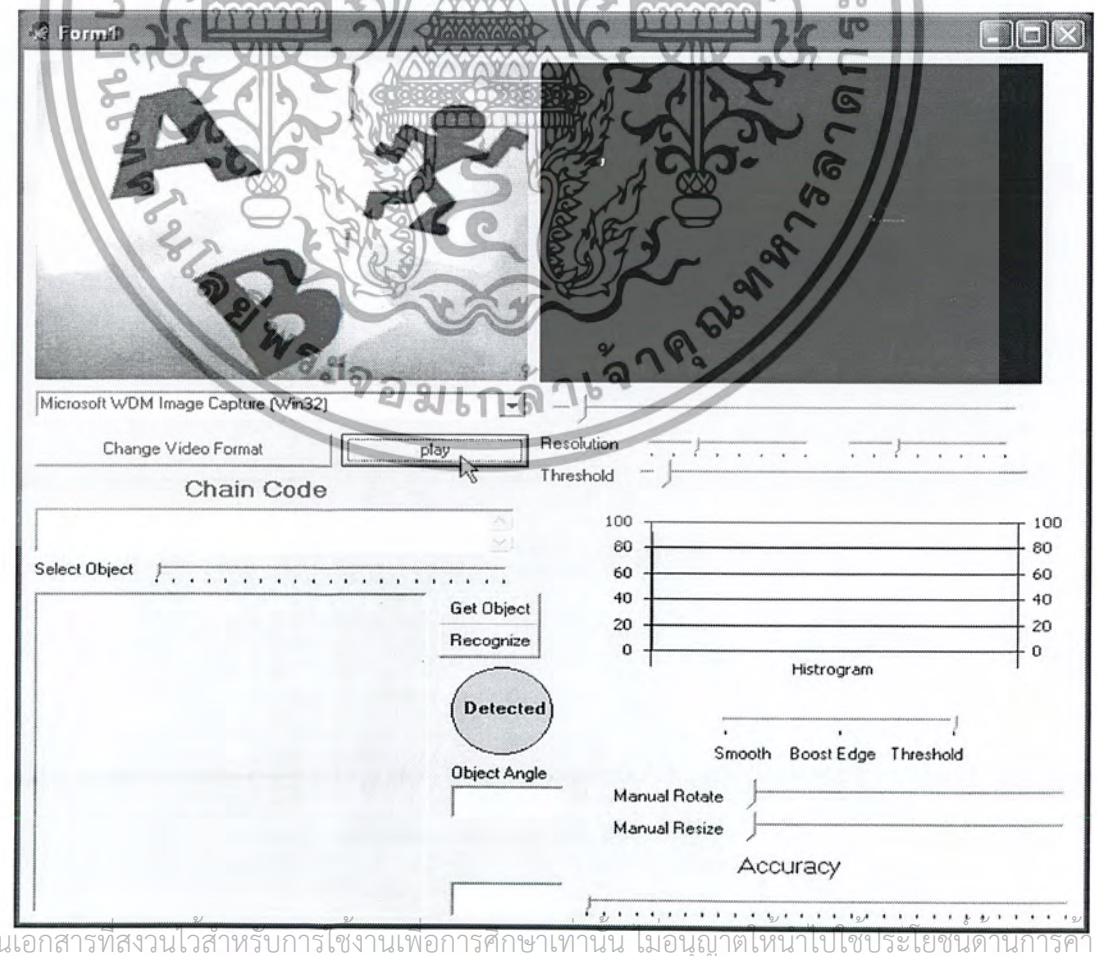
รูปที่ 4.60 รูปแสดงการจดจำวัตถุที่เกิดการคลาดเคลื่อน ทำให้มีการแสดงผลของวัตถุที่จดจำไว้ที่มีรูปร่างไม่เหมือนกัน

รูปที่ 4.61 รูปแสดงการจดจำวัตถุที่เกิดการคลาดเคลื่อน ในกรณีที่มีวัตถุวางซ้อนทับกันอยู่ ทำให้ไม่สามารถจดจำวัตถุได้

รูปที่ 4.62 รูปแสดงการสิ้นสุดการทำงานของโปรแกรม

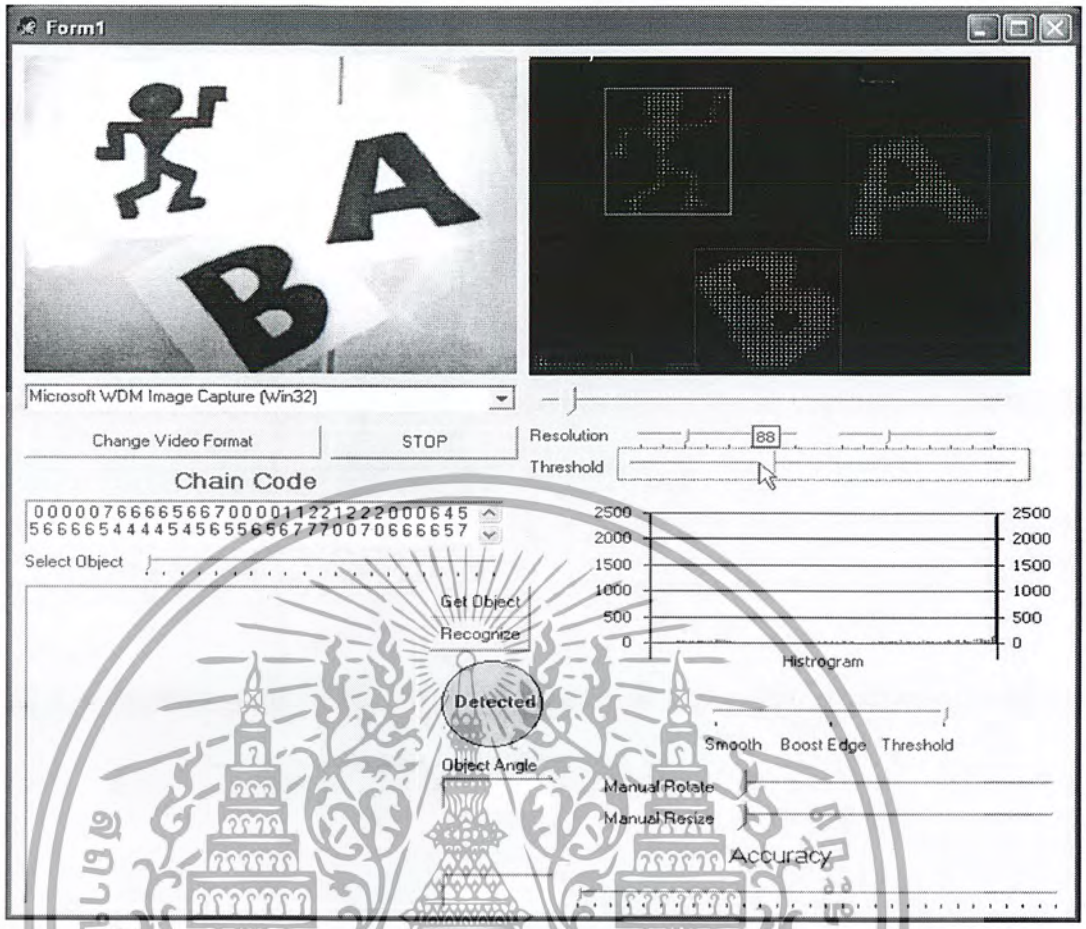
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.51

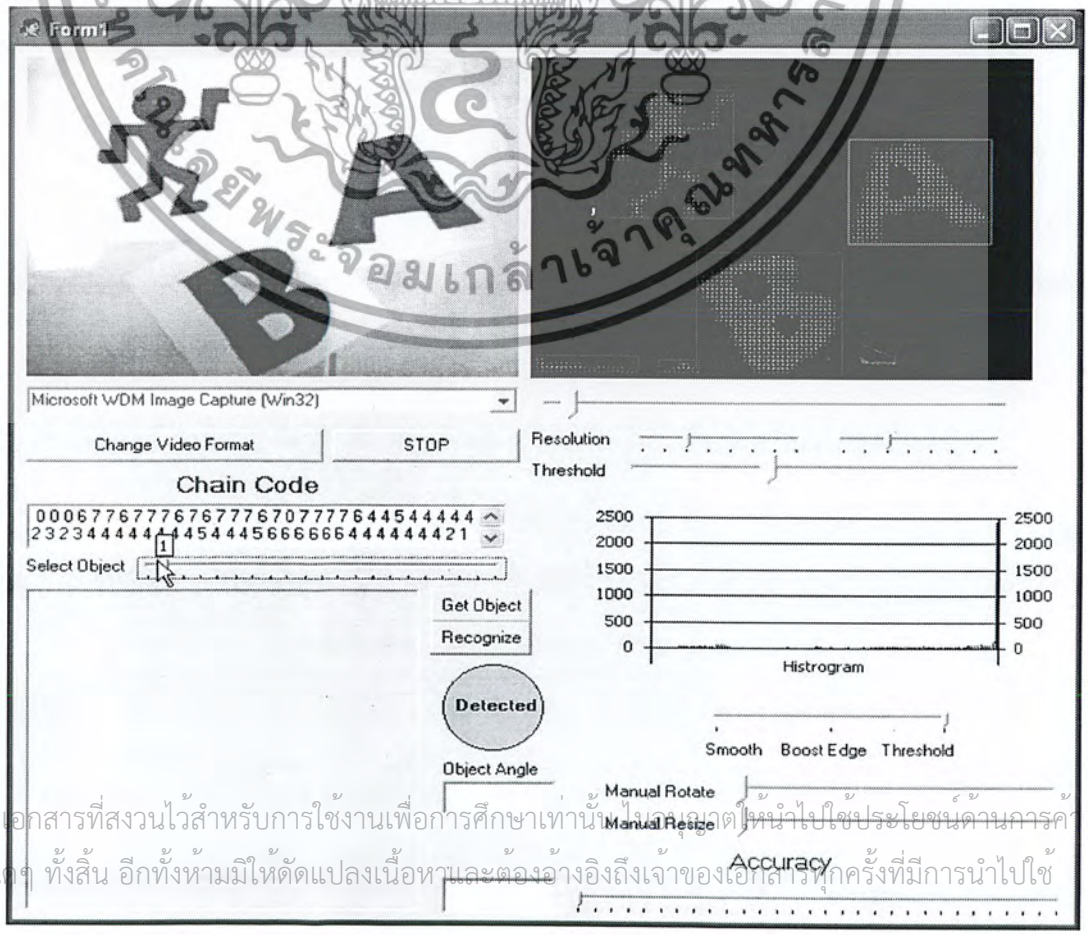


รูปที่ 4.52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

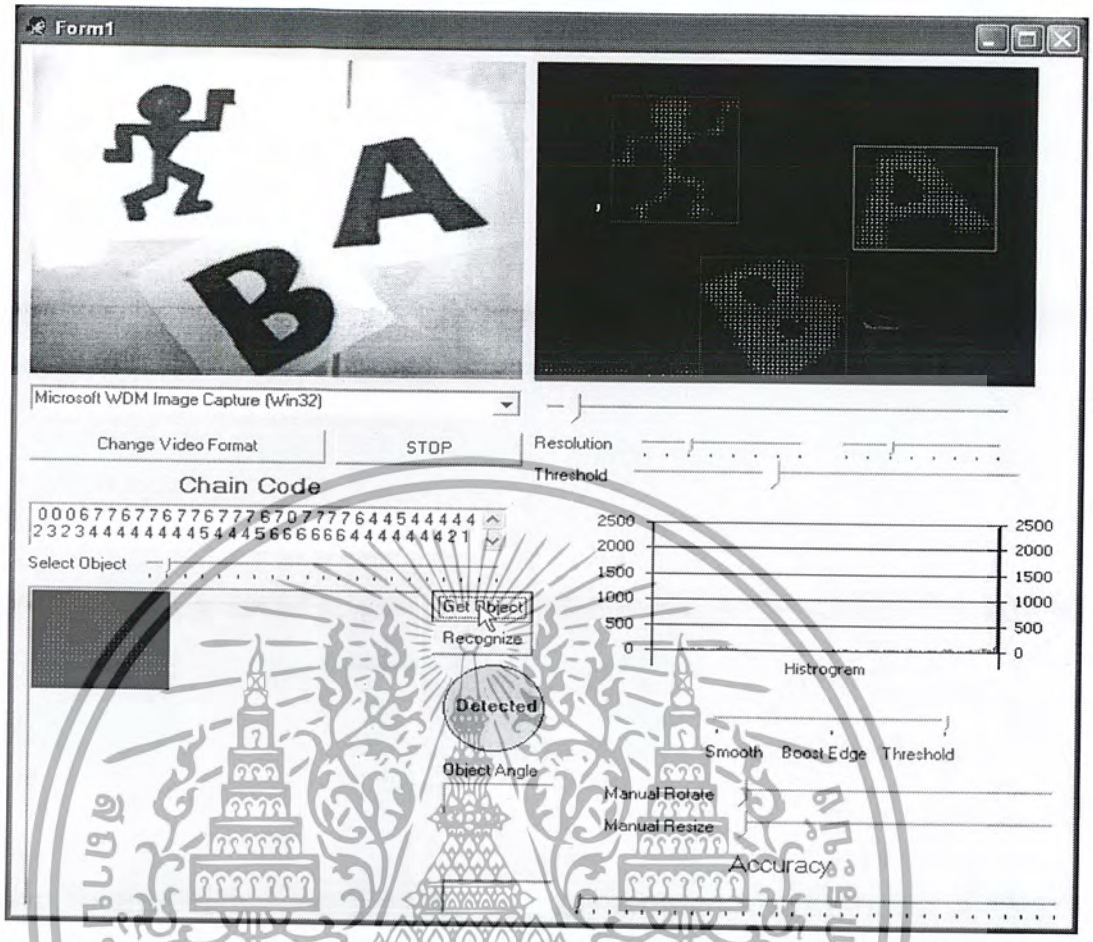


รูปที่ 4.53

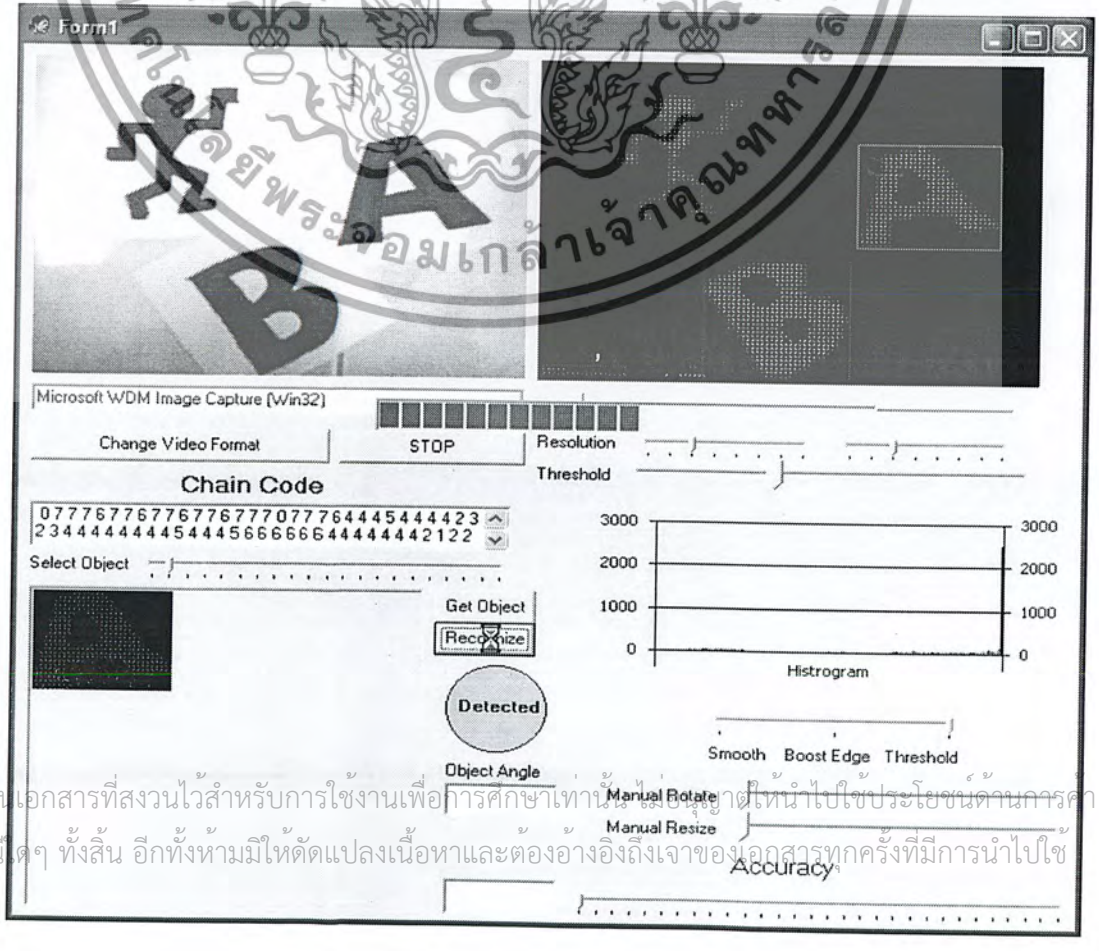


รูปที่ 4.54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

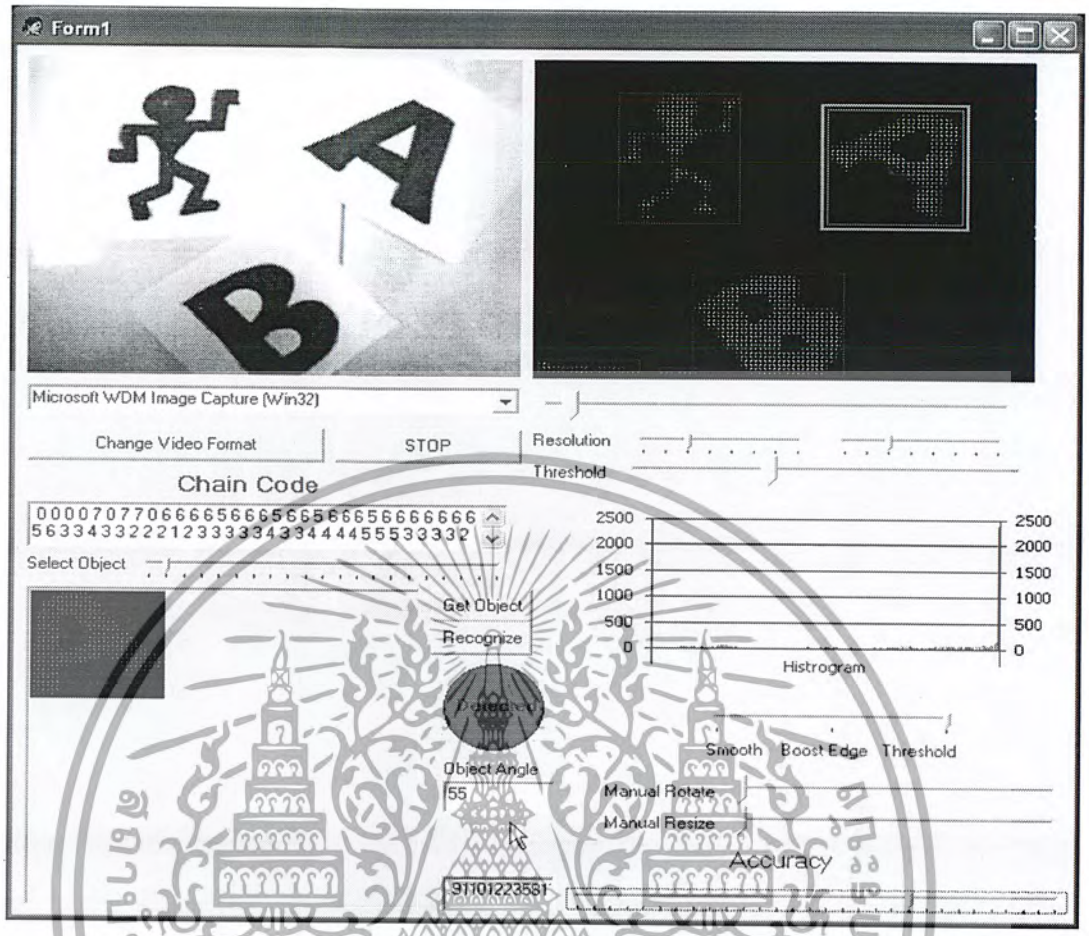


รูปที่ 4.55

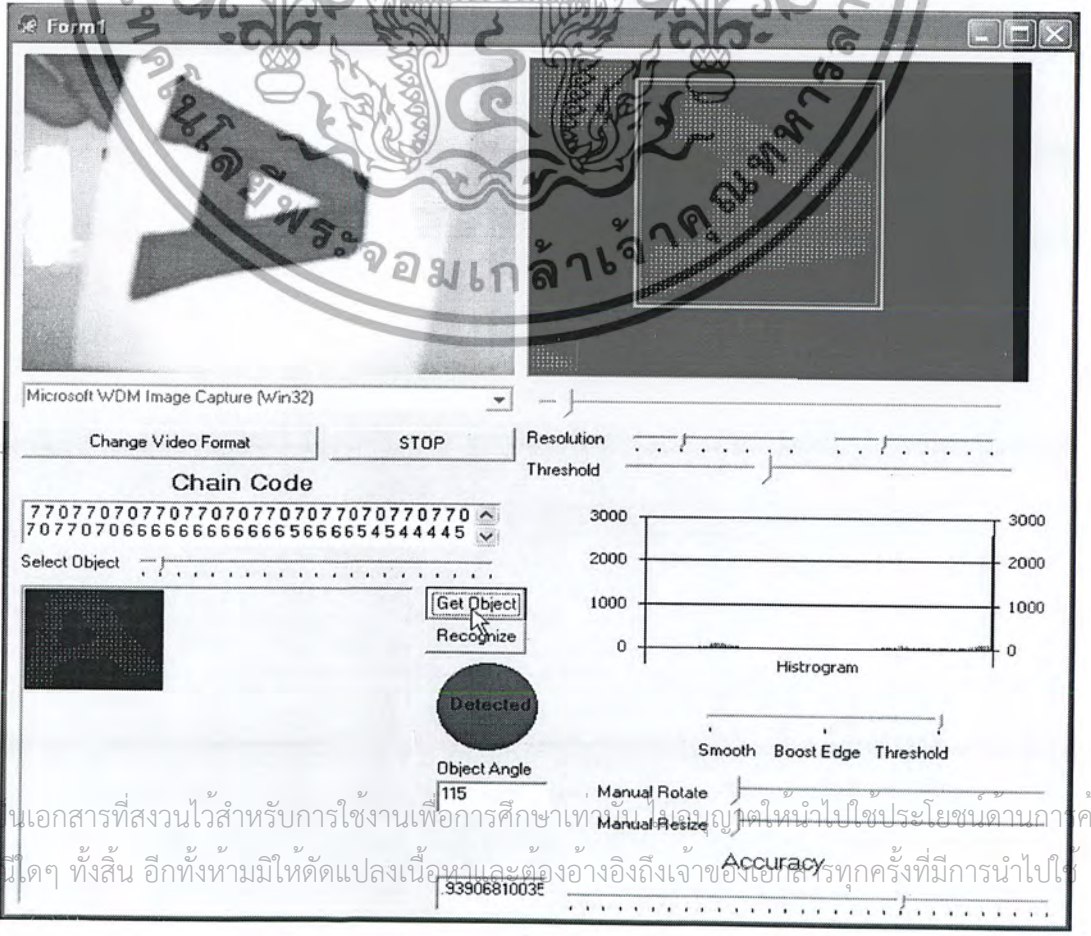


รูปที่ 4.56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

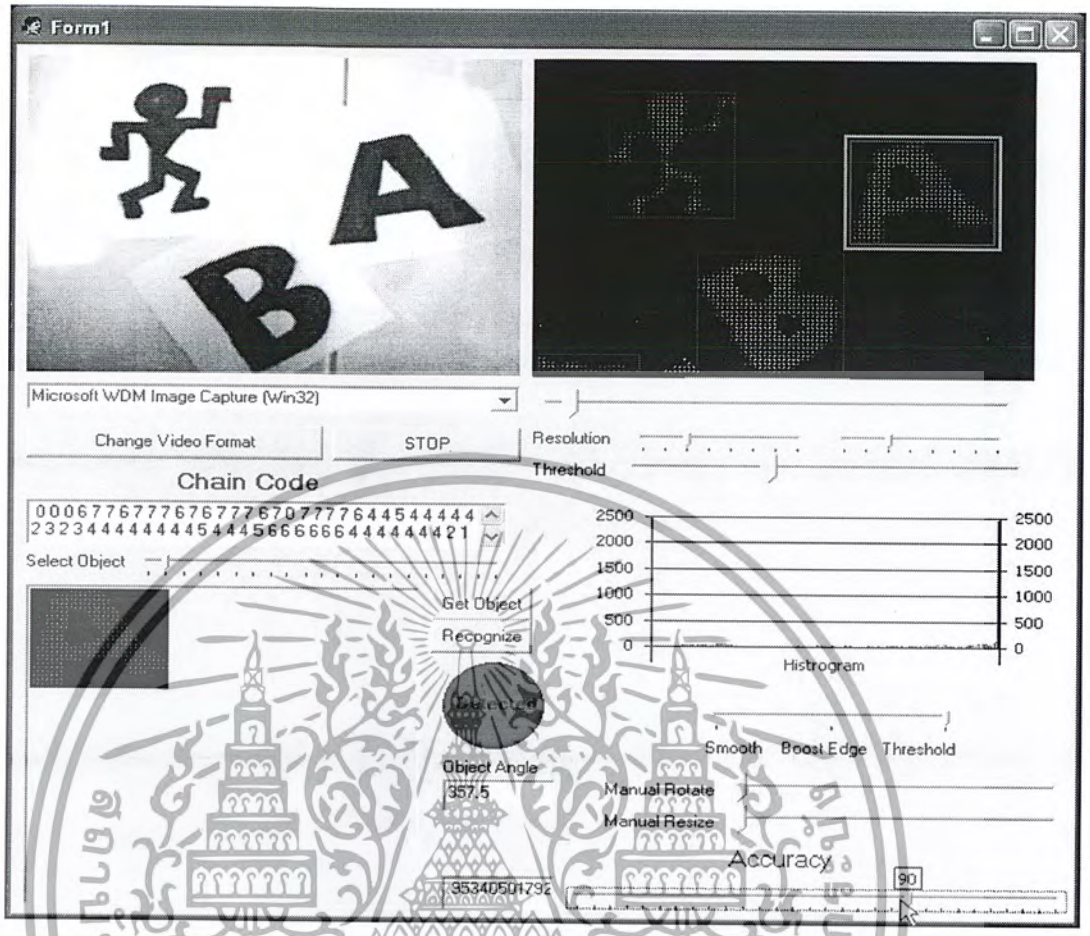


รูปที่ 4.57

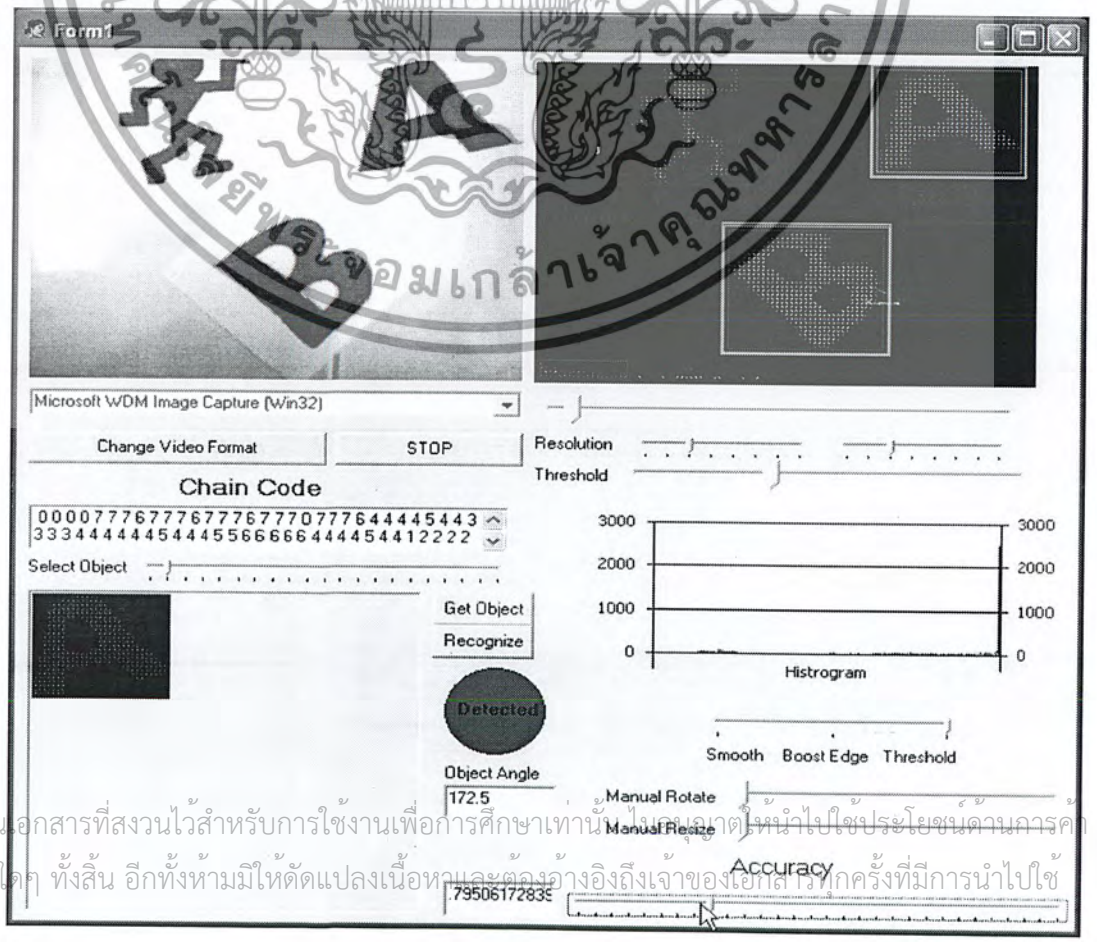


รูปที่ 4.58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

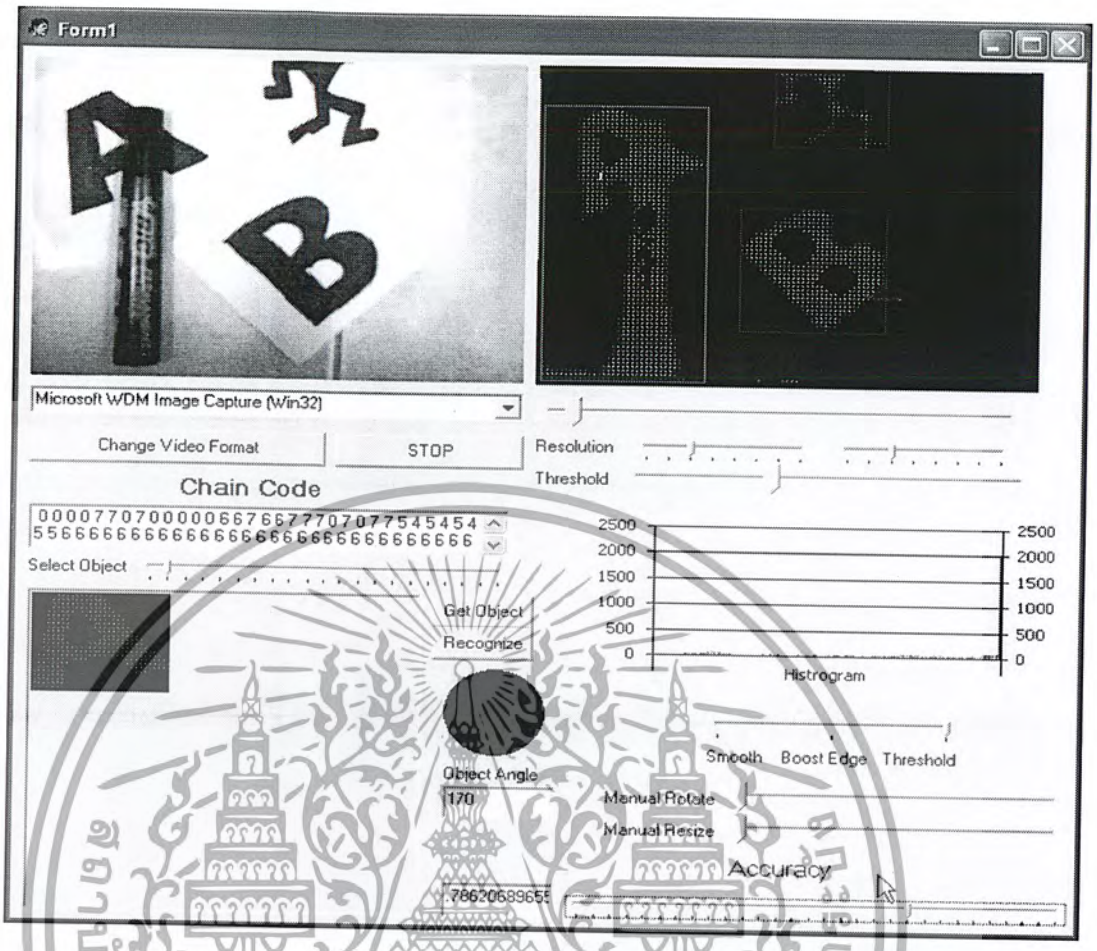


รูปที่ 4.59

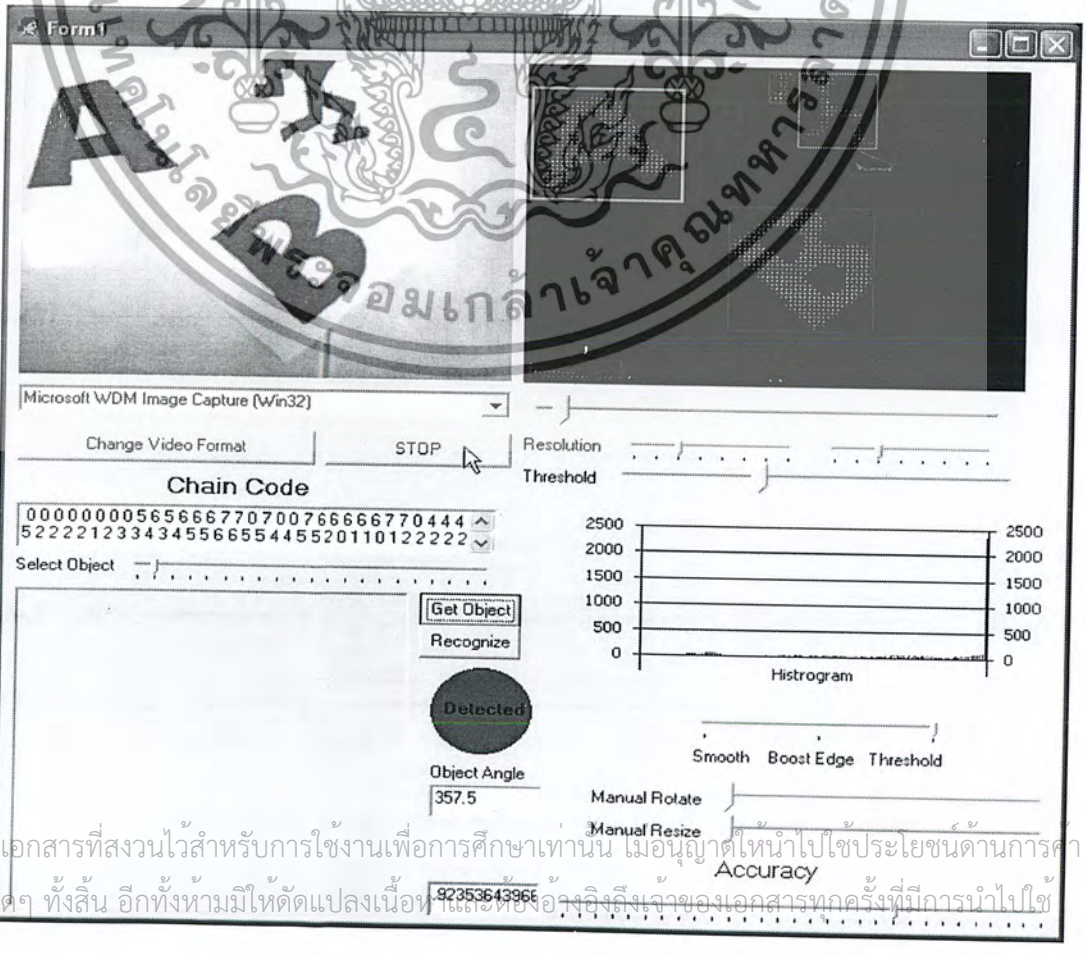


รูปที่ 4.60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.61



รูปที่ 4.62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์ผลการทดลอง

1. การ Smoothing สามารถลดการเกิด Noise จากการหา Edge ได้มาก ถึงแม้ภาพที่ได้จะมีความชัดน้อยลง แต่จะถูกชดเชยได้โดยการเพิ่มความเข้มของขอบภาพ

2. การหาขอบภาพโดยใช้ Filter ต่างๆจะได้คุณสมบัติต่างกัน

แบบ Laplacian 2^{nd} order แบบ Sensitive Edge สามารถหาขอบได้ความสว่างสูง แต่ก็มีข้อเสียคือเพิ่ม Noise มากขึ้นด้วย ถึงแม้จะทำการ Smooth ก่อนก็ยังมี Noise มากกว่าแบบ Light Edge

-1	-1	-1
-1	8	-1
-1	-1	-1

แบบ Sobel ขอบที่ได้จะหนา และหาได้เพียงทิศทางเดียว ถ้าต้องการหาทั้ง 4 ทิศทาง ต้องใช้ Filter 4 อัน ภาพที่ได้จะมี Noise น้อย เนื่องจากตัว Filter มีการ Smooth อยู่ในตัวอยู่แล้ว

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

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

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

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

แบบ Laplacian 2^{nd} order แบบ Light Edge สามารถหาขอบได้ความสว่างไม่มากนัก Noise ไม่สูงมากนัก เมื่อผ่านการ Smooth ก่อน จะมีเหลือ Noise น้อยมาก

0	-1	0
-1	4	-1
0	-1	0

3. การThreshold ใช้แบบ Manual โดยผู้ปรับสามารถสังเกต จาก Histrogram เป็นวิธีที่ง่ายที่สุด แต่ไม่เหมาะกับระบบที่เป็นอัตโนมัติ ต้องปรับปรุงให้เลือกค่า Threshold เองโดยโปรแกรม

4. การหาContour สามารถใช้แยกวัตถุออกจากกันเป็นชิ้นๆได้ เพื่อนำไปประมวลผลในส่วนการจดจำรูปร่างวัตถุต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. การจดจำรูปร่างวัตถุด้วยการ Rotation นั้น เมื่อทำการหมุนวัตถุแล้วทำให้ภาพที่ถูกหมุนไปมีข้อมูลไม่ครบ แก้ไขด้วยการใช้ Filter แบบ 4 จุด จะให้ผลใกล้เคียงกับภาพวัตถุจริงมากที่สุด ซึ่งผลของ Filter แบบ 3 จุดและการทำ Dilation ทำให้ภาพวัตถุนั้นมีขนาดใหญ่เกินความเป็นจริง

6. การ Resize ภาพวัตถุเพื่อทำให้มีขนาดใกล้เคียงกับภาพที่จดจำไว้นั้นในกรณีที่ถูกย่อขนาดให้มีขนาดเล็กมาก ๆ จะทำให้ภาพเกิดการคลาดเคลื่อนของข้อมูลมากกว่าการขยายภาพวัตถุให้มีขนาดใหญ่ ๆ

7. การจดจำรูปภาพนั้นเราสามารถนำภาพตัวอย่างมาทำการหมุนและเก็บภาพนั้นไว้ก่อน เพื่อความไวในการประมวลผล

8. เราสามารถวิเคราะห์วัตถุจากคุณสมบัติง่ายๆเช่นอัตราส่วนความกว้างต่อความยาวก่อน เพื่อลดจำนวนครั้งในการเปรียบเทียบกับภาพต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องมาจากความร่วมมือและการสนับสนุนจากหลายฝ่าย ซึ่งทางคณะผู้จัดทำมีความซาบซึ้งเป็นอย่างยิ่ง ใคร่ขอขอบพระคุณ อาจารย์เทอดศักดิ์ ลิ่วหาทอง อาจารย์ที่ปรึกษาที่คอยให้คำแนะนำ และดูแลการทำโครงงานในครั้งนี้เป็นอย่างดี ขอบคุณเพื่อนๆ พระนครเหนือที่มานั่งคุยเป็นเพื่อนในวันที่มีปัญหา และเพื่อน ๆ ทุกคนที่คอยเป็นกำลังใจและให้คำปรึกษาที่ดีมาโดยตลอด หากมีข้อผิดพลาดประการใดผู้จัดทำก็ขออภัยมา ณ โอกาสนี้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. มนิศร ศิริวงศ์วัฒนา – ยุทธศักดิ์ เภาสมบัติ, เครื่องคัดขนาดวัตถุด้วย Image processing , ปริญญานิพนธ์ สาขาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 2542 .
2. ศิริศักดิ์ วรชื่น – เรืองศักดิ์ โตอร่ามรัตน์ , การประมวลผลภาพ Image processing , ปริญญานิพนธ์ สาขาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 2539 .
3. Wayne Niblack, Digital Image Processing , Prentice – Hall , 1986 .
4. Abhijit S.Pandya – Robert B. Macy , Pattern Recognition with Neural Networks in C++ , CRC Press , 1996 .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Declare Function GetDC Lib "user32" (ByVal hwnd As Long) As Long
Private Declare Function ReleaseDC Lib "user32" (ByVal hwnd As Long, ByVal hdc As Long)
As Long
Private Declare Function GetPixel Lib "gdi32" (ByVal hdc As Long, ByVal x As Long, ByVal y
As Long) As Long
Private Declare Function SetPixel Lib "gdi32" (ByVal hdc As Long, ByVal x As Long, ByVal y
As Long, ByVal crColor As Long) As Long
Private Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal x As Long, ByVal
y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal
xSrc As Long, ByVal ySrc As Long, ByVal dwRop As Long) As Long
Private Declare Function CreateCompatibleBitmap Lib "gdi32" (ByVal hdc As Long, ByVal
nWidth As Long, ByVal nHeight As Long) As Long
Private Declare Function CreateCompatibleDC Lib "gdi32" (ByVal hdc As Long) As Long
Private Declare Function DeleteDC Lib "gdi32" (ByVal hdc As Long) As Long
Private Declare Function SelectObject Lib "gdi32" (ByVal hdc As Long, ByVal hObject As
Long) As Long
Private Declare Function SetPixelV Lib "gdi32" (ByVal hdc As Long, ByVal x As Long, ByVal
y As Long, ByVal crColor As Long) As Long
Private Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long) As Long
Private Declare Function CreateBitmap Lib "gdi32" (ByVal nWidth As Long, ByVal nHeight As
Long, ByVal nPlanes As Long, ByVal nBitCount As Long, lpBits As Any) As Long
Private Type Chaincode
    startx As Integer
    starty As Integer
    datacode As String
    minx As Integer
    miny As Integer
    maxx As Integer
    maxy As Integer
End Type
Private Type mempic
    xmax As Integer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ymax As Integer
    pic(0 To 120, 0 To 120) As Boolean
    xbyy As Single
End Type
Private Type mempicb
    xmin As Integer
    ymin As Integer
    xmax As Integer
    ymax As Integer
    xbyy As Single
    pic(-60 To 60, -60 To 60) As Boolean
End Type
Dim dapic(0 To 143) As mempic
Dim respic As mempic
Dim bufpica As mempic
Dim angee, rox, roy, bufam As Integer
Dim pi, bufan As Single
Dim obj(20) As Chaincode
Dim AMbmp, AMdestdc, Miibmp, Miidestdc As Long
Dim Red, Gre, Blu, g, h, i, j, k, l, a, b, c, tr, mx, mn, ri, rj, objnum As Integer
Dim histro(255) As Long
Dim avv(-10 To 330, -10 To 330) As Integer
Dim edg(-10 To 330, -10 To 330), edg2(-10 To 330, -10 To 330) As Integer
Dim mii(-10 To 330, -10 To 330), objchk As Boolean
Dim miii(-10 To 330, -10 To 330) As Boolean
Dim camerah, gcolor As Long
Dim obnum As Integer
Dim getob As Boolean
Dim cenx, ceny As Integer
Dim comp, totl As Integer
Dim rati As Single
Dim recchk As Boolean
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Function colorR(amman As Long) As Integer
colorR = (amman Mod 256)
End Function

Private Function colorG(amman As Long) As Integer
colorG = (amman \ 65536)
End Function

Private Function colorB(amman As Long) As Integer
colorB = ((amman \ 256) Mod 256)
End Function

Private Sub cbdriver_Change()
Dim olddriver As Long
olddriver = ezVidCap1.DriverIndex
On Error Resume Next
ezVidCap1.DriverIndex = cbdriver.ListIndex
If Err Then
  ezVidCap1.DriverIndex = olddriver
  cbdriver.ListIndex = olddriver
  End If
End Sub

Private Sub Command1_Click()
If Timer1.Enabled = True Then
Timer1.Enabled = False
command1.Caption = "PLAY"
Else: Timer1.Enabled = True
command1.Caption = "STOP"
End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Command2_Click()
ezVidCap1.ShowDlgVideoFormat
End Sub

Private Sub Form_Load()
pi = 3.141592653
Dim i As Integer
Timer1.Interval = Slider1.Value
Frame1.Visible = False
camerah = GetDC(ezVidCap1.hwnd)
 If 0 < ezVidCap1.NumCapDevs Then
  For i = 0 To ezVidCap1.NumCapDevs - 1
   cbdriver.AddItem (ezVidCap1.GetDriverName(i))
   Next i
   cbdriver.ListIndex = ezVidCap1.DriverIndex
   Else
   cbdriver.AddItem ("<none>")
   cbdriver.ListIndex = 0
   MsgBox "no device", vbInformation, App.Title
   End If
   AMbmp = CreateCompatibleBitmap(Picture1(0).hdc, Picture1(0).ScaleWidth,
Picture1(0).ScaleHeight)
   AMdestdc = CreateCompatibleDC(Picture1(0).hdc)
   SelectObject AMdestdc, AMbmp

   Miibmp = CreateCompatibleBitmap(Picture2.hdc, Picture1(0).ScaleWidth,
Picture1(0).ScaleHeight)
   Miidestdc = CreateCompatibleDC(Picture2.hdc)
   SelectObject Miidestdc, Miibmp

   tr = Slider4.Value
   ri = Slider2.Value
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  rj = Slider3.Value

Picture1(2).Enabled = True
Picture1(2).Visible = True

End Sub

Private Sub Form_Unload(Cancel As Integer)
ReleaseDC ezVidCap1.hwnd, camerah
DeleteDC (AMdestdc)
DeleteObject (AMbmp)
End Sub

Private Sub Slider1_Click()
Timer1.Interval = Slider1.Value
End Sub

Private Sub Slider2_Click()
Picture1(Slider7.Value).Refresh
ri = Slider2.Value
Call BitBlt(AMdestdc, 1, 1, 320, 240, Picture1(Slider7.Value).hdc, 1, 1, &HCC0020)
For i = -10 To 325
For j = -10 To 245
   edg(i, j) = 255
   mii(i, j) = False
Next j
Next i
End Sub

Private Sub Slider3_Click()
Picture1(Slider7.Value).Refresh
rj = Slider3.Value
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Call BitBlt(AMdestdc, 1, 1, 320, 240, Picture1(Slider7.Value).hdc, 1, 1, &HCC0020)
For i = -10 To 325
For j = -10 To 245
   edg(i, j) = 255
   mii(i, j) = False
Next j
Next i
End Sub

Private Sub Slider4_Click()
tr = Slider4.Value
End Sub

Private Sub Slider5_Click()
Text2.Text = obj(Slider5.Value).datacode

End Sub

Private Sub Slider6_Click()
Picture2.Cls
For i = 0 To dapic(Slider6.Value).xmax
For j = 0 To dapic(Slider6.Value).ymax
    If dapic(Slider6.Value).pic(i, j) = True Then
    SetPixelV Miidestdc, (i + 1) * ri, (1 + j) * rj, RGB(255, 255, 255)
    Else
    SetPixelV Miidestdc, (1 + i) * ri, (j + 1) * rj, RGB(0, 0, 0)
    End If
Next j
Next i
Call BitBlt(Picture2.hdc, 1, 1, dapic(Slider6.Value).xmax * ri, dapic(Slider6.Value).ymax * rj,
Miidestdc, 1, 1, &HCC0020)
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Slider7_Click()
For i = 0 To 2
If i = Slider7.Value Then
Picture1(i).Enabled = True
Picture1(i).Visible = True
Else
Picture1(i).Enabled = False
Picture1(i).Visible = False
End If
Next i
End Sub

Private Sub Timer1_Timer()
Dim bufpicc As mempic

For i = -10 To 325
For j = -10 To 245
   avv(i, j) = 255
   edg(i, j) = 255
   mii(i, j) = False
Next j
Next i
 '---------------Get Picture to Array of variable--------
   For i = 1 To Picture1(0).ScaleWidth Step ri
      For j = 1 To Picture1(0).ScaleHeight Step rj
         gcolor = GetPixel(camerah, i, j)
         Red = colorR(gcolor)
         Gre = colorG(gcolor)
         Blu = colorB(gcolor)
         avv(i, j) = Int((Red + Gre + Blu) / 3)
         If avv(i, j) > 255 Then avv(i, j) = 255
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        If avv(i, j) < 0 Then avv(i, j) = 0
        histro(avv(i, j)) = histro(avv(i, j)) + 1
      Next j
    Next i
    For i = 0 To 255
      histro1.Column = i + 1
      histro1.Data = histro(i)
      histro(i) = 0
    Next i
  '----------Smooth ----------
    For i = ri + 1 To Picture1(0).ScaleWidth - (ri + 1) Step ri
      For j = rj + 1 To Picture1(0).ScaleHeight - (rj + 1) Step rj
        edg(i, j) = (Int(0.1 * ((2 * avv(i, j)) + avv(i - ri, j - rj) + avv(i, j - rj) + avv(i + ri, j - rj) +
avv(i + ri, j) + avv(i + ri, j + rj) + avv(i, j + rj) + avv(i - ri, j + rj) + avv(i - ri, j))))
        If edg(i, j) > 255 Then edg(i, j) = 255
        If edg(i, j) < 0 Then edg(i, j) = 0
      Next j
    Next i
    For i = ri + 1 To Picture1(0).ScaleWidth - (ri + 1) Step ri
      For j = rj + 1 To Picture1(0).ScaleHeight - (rj + 1) Step rj
        avv(i, j) = edg(i, j)
        SetPixelV AMdestdc, i, j, RGB(edg(i, j), edg(i, j), edg(i, j))
      Next j
    Next i
    Call BitBlt(Picture1(0).hdc, 1, 1, 320, 240, AMdestdc, 1, 1, &HCC0020)
  '----------Edge Boost by Laplacian 2----------
    For i = ri + 1 To Picture1(0).ScaleWidth - ((ri) + 1) Step ri
      For j = rj + 1 To Picture1(0).ScaleHeight - ((rj) + 1) Step rj
        edg(i, j) = (Int(((4 * avv(i, j)) - avv(i, j - rj) - avv(i + ri, j) - avv(i, j + rj) - avv(i - ri, j))))
        If edg(i, j) > 255 Then edg(i, j) = 255
        If edg(i, j) < 0 Then edg(i, j) = 0
        edg(i, j) = 255 - edg(i, j)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Next j
  Next i


  For i = ri + 1 To Picture1(0).ScaleWidth - ((ri) + 1) Step ri
    For j = rj + 1 To Picture1(0).ScaleHeight - ((rj) + 1) Step rj
      edg(i, j) = edg(i, j) And avv(i, j)
      SetPixelV AMdestdc, i, j, RGB(edg(i, j), edg(i, j), edg(i, j))
    Next j
  Next i
  Call BitBlt(Picture1(1).hdc, 1, 1, 320, 240, AMdestdc, 1, 1, &HCC0020)
'----------Threshold picture to binary----------
  For i = ri + 1 To Picture1(0).ScaleWidth - (ri + 1) Step ri
    For j = rj + 1 To Picture1(0).ScaleHeight - (rj + 1) Step rj
      If edg(i, j) < tr Then
      mii(i, j) = True
      SetPixelV AMdestdc, i, j, RGB(255, 255, 255)
      Else
      SetPixelV AMdestdc, i, j, RGB(0, 0, 0)
      mii(i, j) = False
      End If
    Next j
  Next i


'----------Find Contour----------
  Dim ccc As Long
  objchk = True
  i = (ri) + 1
  j = (rj) + 1
  objnum = 0
  Do
    Do
      objchk = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
For k = 0 To objnum
If ((i <= (obj(k).maxx + ri)) And (i >= (obj(k).minx - ri)) And (j <= (obj(k).maxy + rj))
And (j >= (obj(k).miny - rj))) Then
objchk = False
i = obj(k).maxx + ri
Else
objchk = (True And objchk)
End If
Next k

If (mii(i, j) = True) And (objchk = True) Then
obj(objnum).startx = i
obj(objnum).starty = j
obj(objnum).datacode = ""
a = i
b = j
c = 0
ccc = 0
obj(objnum).minx = i
obj(objnum).miny = j
obj(objnum).maxx = i
obj(objnum).maxy = j

Do
ccc = ccc + 1

If c = 0 Then
  If mii(a - ri, b - rj) = True Then
  c = 3
  ElseIf mii(a, b - rj) = True Then
  c = 2
  ElseIf mii(a + ri, b - rj) = True Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
c = 1
ElseIf mii(a + ri, b) = True Then
c = 0
ElseIf mii(a + ri, b + rj) = True Then
c = 7
ElseIf mii(a, b + rj) = True Then
c = 6
ElseIf mii(a - ri, b + rj) = True Then
c = 5
ElseIf mii(a - ri, b) = True Then
c = 4
End If

ElseIf c = 1 Then
If mii(a - ri, b) = True Then
c = 4
ElseIf mii(a - ri, b - rj) = True Then
c = 3
ElseIf mii(a, b - rj) = True Then
c = 2
ElseIf mii(a + ri, b - rj) = True Then
c = 1
ElseIf mii(a + ri, b) = True Then
c = 0
ElseIf mii(a + ri, b + rj) = True Then
c = 7
ElseIf mii(a, b + rj) = True Then
c = 6
ElseIf mii(a - ri, b + rj) = True Then
c = 5
End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ElseIf c = 2 Then
    If mii(a - ri, b + rj) = True Then
    c = 5
    ElseIf mii(a - ri, b) = True Then
    c = 4
    ElseIf mii(a - ri, b - rj) = True Then
    c = 3
    ElseIf mii(a, b - rj) = True Then
    c = 2
    ElseIf mii(a + ri, b - rj) = True Then
    c = 1
    ElseIf mii(a + ri, b) = True Then
    c = 0
    ElseIf mii(a + ri, b + rj) = True Then
    c = 7
    ElseIf mii(a, b + rj) = True Then
    c = 6
    End If
ElseIf c = 3 Then
    If mii(a, b + rj) = True Then
    c = 6
    ElseIf mii(a - ri, b + rj) = True Then
    c = 5
    ElseIf mii(a - ri, b) = True Then
    c = 4
    ElseIf mii(a - ri, b - rj) = True Then
    c = 3
    ElseIf mii(a, b - rj) = True Then
    c = 2
    ElseIf mii(a + ri, b - rj) = True Then
    c = 1
    ElseIf mii(a + ri, b) = True Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    c = 0
    ElseIf mii(a + ri, b + rj) = True Then
    c = 7
    End If
ElseIf c = 4 Then
    If mii(a + ri, b + rj) = True Then
    c = 7
    ElseIf mii(a, b + rj) = True Then
    c = 6
    ElseIf mii(a - ri, b + rj) = True Then
    c = 5
    ElseIf mii(a - ri, b) = True Then
    c = 4
    ElseIf mii(a - ri, b - rj) = True Then
    c = 3
    ElseIf mii(a, b - rj) = True Then
    c = 2
    ElseIf mii(a + ri, b - rj) = True Then
    c = 1
    ElseIf mii(a + ri, b) = True Then
    c = 0
    End If
ElseIf c = 5 Then
    If mii(a + ri, b) = True Then
    c = 0
    ElseIf mii(a + ri, b + rj) = True Then
    c = 7
    ElseIf mii(a, b + rj) = True Then
    c = 6
    ElseIf mii(a - ri, b + rj) = True Then
    c = 5
    ElseIf mii(a - ri, b) = True Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    c = 4
    ElseIf mii(a - ri, b - rj) = True Then
    c = 3
    ElseIf mii(a, b - rj) = True Then
    c = 2
    ElseIf mii(a + ri, b - rj) = True Then
    c = 1
    End If
ElseIf c = 6 Then
    If mii(a + ri, b - rj) = True Then
    c = 1
    ElseIf mii(a + ri, b) = True Then
    c = 0
    ElseIf mii(a + ri, b + rj) = True Then
    c = 7
    ElseIf mii(a, b + rj) = True Then
    c = 6
    ElseIf mii(a - ri, b + rj) = True Then
    c = 5
    ElseIf mii(a - ri, b) = True Then
    c = 4
    ElseIf mii(a - ri, b - rj) = True Then
    c = 3
    ElseIf mii(a, b - rj) = True Then
    c = 2
    End If
ElseIf c = 7 Then
    If mii(a, b - rj) = True Then
    c = 2
    ElseIf mii(a + ri, b - rj) = True Then
    c = 1
    ElseIf mii(a + ri, b) = True Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    c = 0
    ElseIf mii(a + ri, b + rj) = True Then
    c = 7
    ElseIf mii(a, b + rj) = True Then
    c = 6
    ElseIf mii(a - ri, b + rj) = True Then
    c = 5
    ElseIf mii(a - ri, b) = True Then
    c = 4
    ElseIf mii(a - ri, b - rj) = True Then
    c = 3
    End If
End If

If c = o Then
a = a + ri

ElseIf c = 1 Then
a = a + ri
b = b - rj

ElseIf c = 2 Then
b = b - rj
ElseIf c = 3 Then
a = a - ri
b = b - rj

ElseIf c = 4 Then
a = a - ri

ElseIf c = 5 Then
a = a - ri
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    b = b + rj

    ElseIf c = 6 Then
    b = b + rj

    ElseIf c = 7 Then
    a = a + ri
    b = b + rj

    End If

If a < obj(objnum).minx Then obj(objnum).minx = a
If a > obj(objnum).maxx Then obj(objnum).maxx = a
If b < obj(objnum).miny Then obj(objnum).miny = b
If b > obj(objnum).maxy Then obj(objnum).maxy = b

obj(objnum).datacode = obj(objnum).datacode & Str(c)
SetPixelV AMdestdc, a, b, RGB(255, 0, 0)

Loop Until ((a = obj(objnum).startx) And (b = obj(objnum).starty) Or ccc > 5000)
  If Len(obj(objnum).datacode) > (50 / (ri * rj)) Then
  objnum = objnum + 1
  If objnum = 20 Then
    i = 320
    j = 240
  End If

Else
  obj(objnum).datacode = ""
  obj(numobj).maxx = 0
  obj(numobj).maxy = 0
  obj(numobj).minx = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    obj(numobj).miny = 0
    obj(numobj).startx = 0
    obj(numobj).starty = 0

  End If

End If

  i = i + ri
  Loop Until (i >= Picture1(0).ScaleWidth - (2 * ri + 1))
  j = j + rj
  i = (ri) + 1
  Loop Until (j >= Picture1(0).ScaleHeight - (2 * rj + 1))

Call BitBlt(Picture1(2).hdc, 1, 1, 320, 240, AMdestdc, 1, 1, &HCC0020)

For i = 0 To objnum - 1
If i = Slider5.Value Then Picture1(2).ForeColor = &HFF00FF Else Picture1(2).ForeColor =
&HFF0000
Picture1(2).Line (obj(i).minx - 1, obj(i).miny - 1)-(obj(i).maxx + 1, obj(i).miny - 1)
Picture1(2).Line (obj(i).maxx + 1, obj(i).miny - 1)-(obj(i).maxx + 1, obj(i).maxy + 1)
Picture1(2).Line (obj(i).maxx + 1, obj(i).maxy + 1)-(obj(i).minx - 1, obj(i).maxy + 1)
Picture1(2).Line (obj(i).minx - 1, obj(i).maxy + 1)-(obj(i).minx - 1, obj(i).miny - 1)
Next i

Text2.Text = obj(Slider5.Value).datacode

'=================================

If recchk = True Then
'=================================================================================

'-----------------------compare-------------------------------

Shape1.FillColor = vbBlack
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
For k = 0 To objnum - 1
  For l = 0 To 143
    If Abs(((obj(k).maxx - obj(k).minx) / ((obj(k).maxy - obj(k).miny) + 1)) - dapic(l).xbyy) <
0.1 Then
    comp = 0
    '-----------------object to bufpicc----------
    For i = obj(k).minx To obj(k).maxx Step ri
     For j = obj(k).miny To obj(k).maxy Step rj
      g = Round((i - obj(k).minx) / ri)
      h = Round((j - obj(k).miny) / rj)
       bufpicc.pic(g, h) = mii(i, j)
     Next j
    Next i
    bufpicc.xmax = (obj(k).maxx - obj(k).minx) / ri
    bufpicc.ymax = (obj(k).maxy - obj(k).miny) / rj
    '-----------------------------------------


    '-----------------resize bufpicc to dapic(l)---------
    rati = dapic(l).xmax / (bufpicc.xmax)
    For i = 0 To Round(bufpicc.xmax * rati)
      For j = 0 To Round(bufpicc.ymax * rati)
       respic.pic(i, j) = bufpicc.pic(Round(i / rati), Round(j / rati))


       If respic.pic(i, j) = True Then
       Else
       End If
      Next j
     Next i
    '-------------------------------------------------


    For i = 0 To dapic(l).xmax
      For j = 0 To dapic(l).ymax
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    If (dapic(l).pic(i, j) Xor respic.pic(i, j)) = False Then
    comp = comp + 1
    End If
   Next j
  Next i

  If (comp / (dapic(l).xmax * dapic(l).ymax)) > (Slider8.Value / 100) Then
  Text1.Text = (comp / (dapic(l).xmax * dapic(l).ymax))
  Text3.Text = l * 2.5
  Picture1(2).ForeColor = vbGreen
  Picture1(2).Line (obj(k).maxx + 5, obj(k).maxy + 5)-(obj(k).minx - 5, obj(k).maxy + 5)
  Picture1(2).Line (obj(k).minx - 5, obj(k).maxy + 5)-(obj(k).minx - 5, obj(k).miny - 5)
  Picture1(2).Line (obj(k).minx - 5, obj(k).miny - 5)-(obj(k).maxx + 5, obj(k).miny - 5)
  Picture1(2).Line (obj(k).maxx + 5, obj(k).miny - 5)-(obj(k).maxx + 5, obj(k).maxy + 5)
  Picture1(2).Line (obj(k).maxx + 4, obj(k).maxy + 4)-(obj(k).minx - 4, obj(k).maxy + 4)
  Picture1(2).Line (obj(k).minx - 4, obj(k).maxy + 4)-(obj(k).minx - 4, obj(k).miny - 4)
  Picture1(2).Line (obj(k).minx - 4, obj(k).miny - 4)-(obj(k).maxx + 4, obj(k).miny - 4)
  Picture1(2).Line (obj(k).maxx + 4, obj(k).miny - 4)-(obj(k).maxx + 4, obj(k).maxy + 4)

  Shape1.FillColor = vbRed
  End If

  End If
 Next l
Next k
'=============================================================================
==========
End If
'================================
'-----------------------get object----------------------------
If getob = True Then
For i = obj(Slider5.Value).minx To obj(Slider5.Value).maxx Step ri
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    For j = obj(Slider5.Value).miny To obj(Slider5.Value).maxy Step rj
     g = Int((i - obj(Slider5.Value).minx) / ri)
     h = Int((j - obj(Slider5.Value).miny) / rj)
      bufpica.pic(g, h) = mii(i, j)
    If mii(i, j) = True Then
    SetPixelV Miidestdc, g * ri, h * rj, RGB(255, 0, 0)
    End If
    If mii(i, j) = False Then
    SetPixelV Miidestdc, g * ri, h * rj, RGB(0, 0, 0)
    End If
    Next j
Next i
bufpica.xmax = ((obj(Slider5.Value).maxx) - (obj(Slider5.Value).minx)) / ri
bufpica.ymax = ((obj(Slider5.Value).maxy) - (obj(Slider5.Value).miny)) / rj
bufpica.xbyy = bufpica.xmax / (bufpica.ymax + 1)
Call BitBlt(Picture2.hdc, 1, 1, bufpica.xmax * ri, bufpica.ymax * rj, Miidestdc, 1, 1, &HCC0020)

getob = False
End If

For i = 0 To objnum - 1
   obj(i).datacode = ""
    obj(i).startx = 0
    obj(i).starty = 0
    obj(i).minx = 0
    obj(i).miny = 0
    obj(i).maxx = 0
    obj(i).maxy = 0

 Next i

End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Slider9_Click()
Picture2.Cls
rati = Slider9.Value / 10
For i = 0 To Round(dapic(0).xmax * rati)
   For j = 0 To Round(dapic(0).ymax * rati)
      respic.pic(i, j) = dapic(0).pic(Round(i / rati), Round(j / rati))

      If respic.pic(i, j) = True Then
      SetPixelV Miidestdc, (i + 1) * ri, (1 + j) * rj, RGB(255, 255, 255)
      Else
      SetPixelV Miidestdc, (1 + i) * ri, (j + 1) * rj, RGB(0, 0, 0)
      End If
   Next j
Next i
Call BitBlt(Picture2.hdc, 1, 1, Round(dapic(0).xmax * rati) * ri, Round(dapic(0).ymax * rati) * rj,
Miidestdc, 1, 1, &HCC0020)
End Sub

Private Sub getobj_Click()
Picture2.Cls
getob = True
End Sub

Private Sub recognize_Click()
Dim bufpicb As mempicb
recchk = True
For k = 0 To 143
Frame1.Visible = True
Form1.MousePointer = 11
repercent.Value = k
bufpicb.xmin = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
bufpicb.ymin = 0
bufpicb.xmax = 0
bufpicb.ymax = 0
cenx = Round(bufpica.xmax / 2)
ceny = Round(bufpica.ymax / 2)
For i = 0 To bufpica.xmax Step 1
    For j = 0 To bufpica.ymax Step 1

        bufam = Sqr(((i - cenx) * (i - cenx)) + ((j - ceny) * (j - ceny)))
        If (i - cenx) > 0 And (j - ceny) >= 0 Then bufan = Atn((j - ceny) / (i - cenx)) * (180 / pi)
        If (i - cenx) < 0 And (j - ceny) >= 0 Then bufan = 180 - (Atn(Abs((j - ceny) / (i - cenx)))
* (180 / pi))
        If (i - cenx) < 0 And (j - ceny) < 0 Then bufan = 180 + (Atn(Abs((j - ceny) / (i - cenx)))
* (180 / pi))
        If (i - cenx) > 0 And (j - ceny) < 0 Then bufan = 360 - Atn(Abs((j - ceny) / (i - cenx))) *
(180 / pi)
        If (i - cenx) = 0 And (j - ceny) < 0 Then bufan = 270
        If (i - cenx) = 0 And (j - ceny) > 0 Then bufan = 90
        bufan = bufan + (k * 2.5)
        rox = (Round(bufam * Cos(bufan * pi / 180)))
        roy = (Round(bufam * Sin(bufan * pi / 180)))

            If bufpica.pic(i, j) = True Then
            bufpicb.pic(rox, roy) = True

            'bufpicb.pic(rox + 1, roy) = True     'dilation
            'bufpicb.pic(rox - 1, roy) = True
            'bufpicb.pic(rox, roy + 1) = True
            'bufpicb.pic(rox, roy - 1) = True

            If bufpicb.xmin > rox Then bufpicb.xmin = rox - 1
            If bufpicb.ymin > roy Then bufpicb.ymin = roy - 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        If bufpicb.xmax < rox Then bufpicb.xmax = rox + 1
        If bufpicb.ymax < roy Then bufpicb.ymax = roy + 1



        Else
        bufpicb.pic(rox, roy) = False
        End If

Next j
Next i


For i = bufpicb.xmin To bufpicb.xmax
  For j = bufpicb.ymin To bufpicb.ymax
    a = 0
    If bufpicb.pic(i - 1, j - 1) = True Then a = a + 1
    If bufpicb.pic(i, j - 1) = True Then a = a + 1
    If bufpicb.pic(i + 1, j - 1) = True Then a = a + 1
    If bufpicb.pic(i - 1, j) = True Then a = a + 1
    If bufpicb.pic(i + 1, j) = True Then a = a + 1
    If bufpicb.pic(i - 1, j + 1) = True Then a = a + 1
    If bufpicb.pic(i, j + 1) = True Then a = a + 1
    If bufpicb.pic(i + 1, j + 1) = True Then a = a + 1
    If (bufpicb.pic(i, j) = True) Or (a > 4) Then
    dapic(k).pic(i - bufpicb.xmin, j - bufpicb.ymin) = True
    Else
    dapic(k).pic(i - bufpicb.xmin, j - bufpicb.ymin) = False
    End If
  Next j
Next i


For i = bufpicb.xmin To bufpicb.xmax
    For j = bufpicb.ymin To bufpicb.ymax
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            bufpicb.pic(i, j) = False
        Next j
    Next i
    dapic(k).xmax = bufpicb.xmax - bufpicb.xmin
    dapic(k).ymax = bufpicb.ymax - bufpicb.ymin
    dapic(k).xbyy = dapic(k).xmax / (dapic(k).ymax + 1)
Next k

Frame1.Visible = False
Form1.MousePointer = 0
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้