สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เมาส์ควบคุมด้วยดวงตา

Eye-Controlled Mouse

นายธีรพงศ์ ลีลานุภาพ

นายนฤภัทร กำเนิดรัตน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน 49982
วัน,เดือน,ปี 16 เม.ย. 2547

.b.....
.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมาส์ควบคุมด้วยดวงตา

Eye-Controlled Mouse

โดย

นายธีรพงศ์ ลีลานุภาพ

นายนฤภัทร กำเนิดรัตน์

อาจารย์ที่ปรึกษา

อ.เกียรติกูล เจียรนัยธนะกิจ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เมาส์ควบคุมด้วยดวงตา

Eye-Controlled Mouse

ผู้จัดทำ

1. นาย ธีรพงศ์ ลีลานุภาพ รหัสประจำตัว 42010148
2. นาย นฤภัทร กำเนิดรัตน์ รหัสประจำตัว 42010164

อาจารย์ที่ปรึกษา

(อ.เกียรติกูล เจียรนัยธนะกิจ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมาส์ควบคุมด้วยดวงตา

นายธีรพงศ์ ลีลานุภาพ 42010148
นายนฤภัทร กำเนิดรัตน์ 42010164
อ. เกียรติกูล เจียรนัยธนะกิจ อาจารย์ที่ปรึกษา

ปีการศึกษา 2545

บทคัดย่อ

ปัจจุบันนี้ คอมพิวเตอร์มีบทบาทและความสำคัญมากขึ้นเรื่อยๆ ต่อการดำเนินชีวิตประจำวัน ต่อการทำงาน ต่อธุรกิจ เป็นต้น ซึ่งในการรับข้อมูลของคอมพิวเตอร์นั้น เมาส์ (Mouse) เป็นอุปกรณ์ที่สำคัญอย่างยิ่งที่ใช้สำหรับรับข้อมูลจากผู้ใช้มาสู่คอมพิวเตอร์ซึ่ง การเมาส์ควบคุมด้วยดวงตานั้น เป็นทางเลือกหนึ่งสำหรับผู้พิการที่ไม่สามารถใช้มือและขาในการควบคุมเมาส์ ตัวอย่างเช่น ผู้ที่เป็นอัมพาตตั้งแต่กระดูกสันหลังลงมา เป็นต้น ซึ่งการทำงานเมาส์ควบคุมด้วยดวงตา นั้น ประกอบด้วย Web Camera โดยใช้กล้องวีดีโอจับภาพการเคลื่อนที่ของดวงตา 1 ข้าง ข้อมูลภาพที่ได้จะถูกนำไปประมวลผลทางรูปภาพ เพื่อนำไประบุและแสดงผลตำแหน่งการจ้องมองบนจอภาพ

ผู้จัดทำโครงงานจึงมีความคิดที่จะทำให้ผู้พิการสามารถใช้คอมพิวเตอร์ได้ทัดเทียมกับบุคคลปกติ เพราะผู้พิการเป็นบุคลากรที่สำคัญต่อประเทศชาติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Eye-Controlled Mouse

Mr. Teerapong Leelanupab 42010148

Mr. Narupat Kumnurtrath 42010164

Mr. Kietikul Jearanaitanakij Advisor

Year 2002

Abstract

Computer plays an important role over our daily life and business. In term of data receiving process, "Mouse" is one of the most important devices to transfer data from user to CPU.

Eye-Controlled Mouse, is an alternative for a computer user who cannot control mouse by hand or leg such as the case of ALS or motor neuron disease. The process of *Eye-Controlled Mouse* is started when web camera catches the movement of an eye. The image will be transformed to be a data by image processing to identify the mouse position on the monitor.

The inspiration of this project is to equalize the working ability between normal and disable by inventing a device to reduce the gap of difference.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้ สำเร็จลงด้วยดีได้ เพราะความกรุณาจากอาจารย์ต่างๆ ที่ให้การอบรมสั่งสอนวิชาความรู้ตั้งแต่ต้น จนถึงปัจจุบัน โดยเฉพาะอย่างยิ่ง อาจารย์ที่ปรึกษา อาจารย์เกียรติกูล เจียรนัยธนะกิจ ที่คอยให้คำปรึกษา พร้อมทั้งแนวคิด ข้อเสนอแนะ ตลอดจนการแก้ไขบกพร่องต่างๆ โดยตลอด ผู้จัดทำรู้สึกซาบซึ้ง และขอกราบขอบพระคุณอย่างสูง มา ณ โอกาสนี้

ขอขอบพระคุณ อาจารย์ อรฉัตร จิตต์โสภักตร์ ที่ให้คำปรึกษาทางด้าน Image processing เป็นอย่างดี รวมทั้งให้ยืม Web camera มาตลอดทั้งเทอม

ขอขอบพระคุณ บิดา มารดา ที่คอยอบรมสั่งสอน รับฟังปัญหาต่างๆ และทุ่มเทกำลังกายกำลังใจในการให้การศึกษาแก่ผู้จัดทำ และให้กำลังใจมาเป็นอย่างดีโดยตลอด ผู้จัดทำขอกราบขอบพระคุณมา ณ โอกาสนี้ด้วย

ขอขอบคุณ ตู่และเท็น ที่ให้ความช่วยเหลือด้าน Programming

นาย ธีรพงศ์ ลีลานุภาพ

นาย นฤภัทร กำเนิดรัตน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VIII
สารบัญตาราง	XI
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงงานวิจัย	1
1.3 ขอบเขตของโครงงาน	2
1.4 รายละเอียดโครงงาน	2
1.5 วิธีการดำเนินงาน	2
บทที่ 2 การเขียนโปรแกรมติดต่อระหว่างแอปพลิเคชั่น กับวินโดวส์โดยใช้ API	3
2.1 API คือ อะไร	3
2.2 ลักษณะของ Event-Driven Programming	4
2.3 ลักษณะโครงสร้างการทำงานของ Windows API	5
2.4 กลุ่มฟังก์ชันของ Windows API	8
2.5 การประกาศฟังก์ชันใน Windows API	9
บทที่ 3 โครงสร้างและการจัดการรูปภาพ	10
3.1 โครงสร้างของภาพชนิด Bitmap	10
3.2 การสร้างพื้นที่ในหน่วยความจำเพื่อเก็บภาพ	12
3.3 การโหลดหรือส่งรูปภาพมาเก็บไว้ในหน่วยความจำที่ได้สร้างไว้	14
3.4 การลดหรือขยายขนาดของรูปภาพในหน่วยความจำ	16
บทที่ 4 การประมวลผลทางรูปภาพ เรื่อง Similarity Segmentation : Thresholding	17
4.1 Segmentation	17
4.2 เทคนิค Segmentation	17
4.2.1 Discontinuity	17
4.2.1.1 First-Second Derivative (No noise)	17
4.2.1.2 First-Second Derivative (With noise)	18
4.2.1.3 First Derivative (Gradient)	19
4.2.1.4 Laplacian of Gaussian (LoG)	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้าที่

4.2.2 Simiailarity 20

4.2.2.1 *Thresholding* 20

4.2.2.2 Region Growing 26

4.2.2.3 Region Clustering 28

4.2.2.4 *Region Splitting and Merging* 29

บทที่ 5 การเขียนโปรแกรมติดต่อกับ Web Camera 30

5.1 การควบคุมและติดต่อ Web Camera ด้วย Component ของ ezVidCap 30

5.2 การดึงภาพจาก Component ของ ezVidCap เข้ามาเก็บไว้ในหน่วยความจำ 32

บทที่ 6 แนวคิดและการออกแบบ 33

6.1 Functional Requirements 33

6.2 Non-functional Requirements 33

6.3 Use Case 34

6.4 Sequence Diagram 35

6.5 Object Diagram 40

6.6 Deployment Diagram 40

บทที่ 7 การขยายภาพ 41

7.1 Geometric Transformation 41

7.2 Backward mapping 43

7.3 การขยายภาพ โดยใช้ Zero-order Interpolation 44

7.4 การขยายภาพ โดยใช้ฟังก์ชั่น StretchBlt 45

บทที่ 8 การหาจุดศูนย์การของดวงตา 47

8.1 ขั้นตอนการหาขอบบนขอบล่างของดวงตา 47

8.1.1 หาขอบล่างของดวงตา 48

8.1.2 หาขอบบนของดวงตา 49

8.2 ขั้นตอนการหาด้านซ้ายและด้านขวาของดวงตา 49

8.3 ปรับปรุง Algorithm ในการใช้ Sliding Window 54

8.4 ขั้นตอนการหาจุดศูนย์กลางของดวงตา 56

บทที่ 9 การกำจัดสิ่งรบกวน 57

9.1 หลักการสำคัญของการกำจัดสิ่งรบกวน 57

9.2 ขั้นตอนของการกำจัดสิ่งรบกวน 58

9.2.1 การหาขอบเขตในการค้นหาด้านซ้ายของภาพ 59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้าที่

9.2.2 การหาขอบเขตในการค้นหาด้านขวาของภาพ 59
9.2.3 ข้อจำกัดในการหาขอบเขตการค้นหาของภาพ 60
9.2.3.1 ขอบเขตค้นหาด้านซ้ายและด้านขวาของภาพจะต้องอยู่แนวเดียวกัน 60
9.2.3.2 ตาอยู่ขวาสุดของภาพ จะต้องปรับปรุงการใช้ Sliding Window 60
บทที่ 10 การแสดงผลตำแหน่งบนหน้าจอ 65
10.1 เคอร์เซอร์ (Cursor) 65
10.2 การแสดงผลตำแหน่งบนหน้าจอ 65
10.3 หลักการแปลงตำแหน่งจุดศูนย์กลางของตาเป็นตำแหน่งเมาส์ 66
10.4 Windows API ที่ใช้ในการกำหนดตำแหน่งของเคอร์เซอร์เมาส์ 67
บทที่ 11 การคลิก 69
11.1 หลักการคลิก 69
11.2 Windows API ที่ใช้ในการคลิก 69
11.2.1 การใช้ Windows API กับการคลิก 71
บทที่12 การทดสอบระบบ 72
12.1ทดสอบอัตราความเร็วของเมาส์ 72
12.1.1 การออกแบบการทดสอบอัตราความเร็วของเมาส์ 72
12.1.2 ผลลัพธ์การทดสอบอัตราความเร็วของเมาส์ 73
12.2 การทดสอบการลากและปล่อยของเมาส์ 73
12.2.1การออกแบบการทดสอบการลากและปล่อยของเมาส์ 73
12.2.2 ผลลัพธ์การทดสอบการลากและปล่อยของเมาส์ 74
12.3 การทดสอบการควบคุมเมาส์ 75
12.3.1 การออกแบบการทดสอบการควบคุมเมาส์ระดับ 1 75
12.3.2 การออกแบบการทดสอบการควบคุมเมาส์ระดับ 2 76
12.3.3 ผลลัพธ์การทดสอบการควบคุมเมาส์ 77
บทที่ 13 บทสรุปและวิจารณ์ 78
13.1 ปัญหา 78
13.2 แนวทางการแก้ไขปัญหาและการพัฒนาต่อ 78
13.3 ข้อเสนอแนะ 78
13.4 การประยุกต์ใช้ 79
13.5 สรุปผลการทดสอบโปรแกรม 79
ภาคผนวก ก โครงสร้างไฟล์ Bitmap 80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้าที่
ภาคผนวก ข Component ของ EzVidCap	86
บรรณานุกรม	110

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้าที่

รูปที่ 2-1 แสดงหลักการทำงานของ Windows API 4
รูปที่ 2-2 แสดงการทำงานของ API Programming โดยทำงานแบบ Event-Driven Programming 4
รูปที่ 2-3 แสดงการลำดับขั้นตอนการทำงานของ Windows API Programming 5
รูปที่ 2-4 แสดงขั้นตอนการทำงานของ Windows API Programming ตั้งแต่เริ่มต้นจนจบการทำงาน 6
รูปที่ 2-5 แสดงการทำงานของ Message ชนิดต่างๆ 7
รูปที่ 3-1 แสดงลักษณะการเก็บภาพที่อยู่ในไฟล์เปรียบเทียบกับภาพที่แสดงจริงโดย 10
รูปที่ 3-2 การโหลดภาพมาเก็บในหน่วยความจำ 12
รูปที่ 3-3 แสดงการส่งภาพของฟังก์ชั่น BitBlt 14
รูปที่ 3-4 แสดงขั้นตอนการส่งรูปภาพมาใส่ในหน่วยความจำที่จองไว้ 15
รูปที่ 3-5 แสดงการโหลดหรือส่งรูปภาพมาแสดงผลบนพื้นที่ที่สร้างไว้บนหน้าจอของแอปพลิเคชั่น 15
รูปที่ 3-6 แสดงการส่งภาพของฟังก์ชั่น StretchBlt 16
รูปที่ 4-1 การหาขอบของภาพด้วยวิธี First-Second Derivative (ไม่มี noise) 17
รูปที่ 4-2 การหาขอบของภาพด้วยวิธี First-Second Derivative (มี noise)(1) 18
รูปที่ 4-3 การหาขอบของภาพด้วยวิธี First-Second Derivative (มี noise)(2) 19
รูปที่ 4-4 การหาขอบของภาพด้วยวิธี First Derivative 19
รูปที่ 4-5 การหาขอบของภาพด้วยวิธี Laplacian of Gaussian (LoG) 20
รูปที่ 4-6 Histogram แสดงจุด Optimal 20
รูปที่ 4-7 Flow Chart แสดงการหา Optimum Threshold Estimation 21
รูปที่ 4-8 ตัวอย่างภาพขาวดำที่เลือกค่า Threshold=90 22
รูปที่ 4-9 Flow Chart การทำThresholding Gray Scale 23
รูปที่ 4-10 การแบ่งภาพขาวดำโดยใช้ค่า Threshold ต่างๆ 24
รูปที่ 4-11 การแบ่งภาพสีโดยใช้ค่า Threshold ของสีแดง น้ำเงิน เขียว 24
รูปที่ 4-12 ฟังก์ชัน ตรวจค่าในช่วงที่สนใจของภาพสี 25
รูปที่ 4-13 การจัดกลุ่ม pixel ที่สีเหมือนกันรวมกลุ่มด้วยกัน 25
รูปที่ 4-14 กระบวนการ Region Growing 26
รูปที่ 4-15 ขั้นตอนการทำ Region Growing 26
รูปที่ 4-16 Flow Chart ของ Region Growing 27
รูปที่ 4-17 Flow Chart ของ Region Clustering 28
รูปที่ 4-18 ภาพโครงสร้างของ Region Splitting and Merging 29
รูปที่ 4-19 ขั้นตอนการทำ Region Splitting and Merging 29
รูปที่ 5-1 แสดง ADD Components ให้โปรแกรมวิชวลเบสิก 30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้าที่

รูปที่ 5-2 แสดงการ ADD Component ที่ชื่อ ezVidCap Component 31
รูปที่ 5-3 แสดงการ ADD Control Component ลง Form 31
รูปที่ 6-1 Use Case Diagram 34
รูปที่ 6-2 Sequence Diagram ของการเคลื่อนที่ของเมาส์ 35
รูปที่ 6-3 Sequence Diagram ของการคลิกขวาของเมาส์ 36
รูปที่ 6-4 Sequence Diagram ของการคลิกซ้ายของเมาส์ 37
รูปที่ 6-5 Sequence Diagram ของการดับเบิลคลิกของเมาส์ 38
รูปที่ 6-6 Sequence Diagram ของการลากและปล่อยของเมาส์ 39
รูปที่ 6-7 Object Diagram ของเมาส์ 40
รูปที่ 6-8 Deployment Diagram 40
รูปที่ 7-1 Geometric Transformation 41
รูปที่ 7-2 ฟังก์ชัน Geometric Transformation 41
รูปที่ 7-3 Backward mapping 43
รูปที่ 7-4 Flow Chart Backward Mapping 43
รูปที่ 7-5 Flow Chart Zero-order Interpolation 44
รูปที่ 7-6 แสดงการขยายภาพของฟังก์ชั่น StretchBlt 45
รูปที่ 8-1 แสดงภาพดวงตาที่สมบูรณ์ไม่มีเงาของแสงสว่างมารบกวน 47
รูปที่ 8-2 ภาพดวงตาที่ผ่านการทำเซ็กเมนต์เตชั่น และกา ค้นหาขอบบนขอบล่างของดวงตา 48
รูปที่ 8-3 ภาพดวงตาที่มีเงาของแสงสว่างมารบกวนด้านขวาของตาดำ 50
รูปที่ 8-4 ทำปรับปรุงจุดเริ่มค้นหาและใช้ Sliding Windows เพื่อหาจำนวนจุดสีดำที่น่าจะเป็นตาดำ 50
รูปที่ 8-5 ภาพดวงตาที่มีเงาของแสงสว่างมารบกวนทั้งด้านซ้ายและขวาของตาดำ 51
รูปที่ 8-6 ทำการปรับปรุงจุดก่อนเริ่มการใช้ Sliding Window เพื่อความถูกต้อง 52
รูปที่ 8-7 ทำการปรับค่า Threshold เพิ่มมากขึ้น แต่ก็ทำให้มีสิ่งรบกวนเพิ่มมากขึ้นตามไปด้วย 53
รูปที่ 8-8 การค้นหาจุดสีดำของSliding Windowแบบเดิม ซึ่งมีส่วนซ้อนทับกัน ทำให้เกิดความล่าช้า 54
รูปที่ 8-9 การค้นหาจุดสีดำของ Sliding Window แบบใหม่ ซึ่งจะไม่ค้นหาส่วนที่เคยค้นหาแล้ว 55
รูปที่ 9-1 แสดงหลักการสำคัญของการกำจัดสิ่งรบกวน เพื่อหาช่วงการค้นหา 57
รูปที่ 9-2 ภาพดวงตาที่ผ่านการทำเซ็กเมนต์เตชั่น ของส่วนที่เป็นสีขาวของดวงตา 58
รูปที่ 9-3 ภาพดวงตามองไปทางขวาสุด(ซ้ายของภาพ) จะไม่เกิดปัญหา 60
รูปที่ 9-4 ภาพดวงตาที่ผ่านการทำเซ็กเมนต์เตชั่นในที่ช่วงเป็น “สีดำ” โดยใช้ค้นหาขอบเขตด้านซ้าย 61
รูปที่ 9-5 ภาพดวงตาที่ผ่านการทำเซ็กเมนต์เตชั่นในที่ช่วงเป็น “สีขาว” โดยใช้ค้นหาขอบเขตด้านขวา 62
รูปที่ 9-6 ตาอยู่ทางขวาสุดของภาพ จะเกิดปัญหาต้องแก้ไขการใช้ Sliding Window 63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้าที่

รูปที่ 9-7 ภาพที่ทำเซ็กเมนต์เตชั่นในที่ช่วงเป็น “สีขาว” โดยใช้ค้นหาขอบเขตด้านซ้ายและขวา 64
รูปที่ 10-1 การแสดงผลตำแหน่งบนหน้าจอ 65
รูปที่ 10-2 Flow Chart การแปลงตำแหน่งจุดศูนย์กลางของตาเป็นตำแหน่งเมาส์ 66
รูปที่ 10-3 แสดงการกำหนดตำแหน่งโคออร์ดิเนตของสี่เหลี่ยม 67
รูปที่ 11-1 แสดงตำแหน่งการมองจ้องกับการคลิก 69
รูปที่ 12-1 หน้าต่างการทดสอบอัตราความเร็วของเมาส์ 72
รูปที่ 12-2 ผลลัพธ์การทดสอบอัตราความเร็วของเมาส์ 73
รูปที่ 12-3 หน้าต่างการทดสอบการลากและปล่อยของเมาส์ 74
รูปที่ 12-4 ผลลัพธ์การทดสอบการลากและปล่อยของเมาส์ 74
รูปที่ 12-5 หน้าต่างการทดสอบการควบคุมเมาส์ระดับ 1 75
รูปที่ 12-6 ผลลัพธ์การทดสอบการควบคุมเมาส์ระดับ 1 75
รูปที่ 12-7 หน้าต่างการทดสอบการควบคุมเมาส์ระดับ 2 76
รูปที่ 12-8 ผลลัพธ์การทดสอบการควบคุมเมาส์ระดับ 2 76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้าที่

ตารางที่ 2-1 แสดงตารางเกี่ยวกับไฟล์ DLL ที่สำคัญ 8

ตารางที่ 7-1 *Geometric Transform Parameters (Backward Mapping)* 42

ตารางที่ 12-1 การเปรียบเทียบอัตราความเร็วของเมาส์ 73

ตารางที่ 12-2 การเปรียบเทียบเวลาการลากและปล่อยของเมาส์ 74

ตารางที่ 12-3 การเปรียบเทียบการทดสอบการควบคุมเมาส์ 77

ตารางที่ 13-1 สรุปผลการทดสอบโปรแกรมตาม Functional Requirements 79

ตารางที่ 13-2 สรุปผลการทดสอบโปรแกรมตาม Non-Functional Requirements 79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ปัจจุบันนี้ คอมพิวเตอร์มีบทบาทและความสำคัญมากขึ้นเรื่อยๆ ต่อการดำเนินชีวิตประจำวัน ต่อการทำงาน ต่อธุรกิจ เป็นต้น ซึ่งในการรับข้อมูลของคอมพิวเตอร์นั้น เมาส์ (Mouse) เป็นอุปกรณ์รับข้อมูล (Input device) ตัวหนึ่งของวินโดวส์ที่มีความสำคัญเป็นอย่างมาก เพราะในปัจจุบันได้ถูกการออกแบบให้มีการใช้เมาส์เป็นอุปกรณ์หลัก สำหรับการเลือกหรือตอบโต้การทำงานกับแอปพลิเคชั่น (ยกเว้นเมื่อต้องการให้มีการกรอกข้อความ ซึ่งต้องอาศัยคีย์บอร์ดเท่านั้น) แต่ถ้าหากคนเรามีความพิการ ทางด้านร่างกายที่ไม่สามารถใช้มือหรือขาควบคุมเมาส์ ผู้พิการก็จะไม่โอกาสใช้คอมพิวเตอร์

เมาส์ควบคุมด้วยดวงตา เป็นทางเลือกสำหรับผู้พิการไม่สามารถใช้มือและขาในการควบคุมเมาส์ ตัวอย่าง เช่น ผู้ที่เป็นอัมพาตตั้งแต่กระดูกสันหลังลงมา เป็นต้น ซึ่งการทำงานเมาส์ควบคุมด้วยดวงตานั้น ประกอบ ด้วย Web Camera โดยใช้กล้องวีดีโอจับภาพการเคลื่อนที่ของดวงตา 1 ข้าง ข้อมูลภาพที่ได้จะถูกนำไปประมวลผลทางรูปภาพ (Image processing) เพื่อนำไประบุและแสดงผลตำแหน่งการจ้องมองบนจอภาพ การคลิกนั้นสามารถทำได้โดย จ้องไปที่จุดที่เราต้องการคลิกเป็นเวลาประมาณ 3 วินาที โปรแกรมจะทำการเปลี่ยนรูปแบบการทำงานจากเคลื่อนที่เป็นการรอรับข้อมูลเพื่อทำการคลิก ดังต่อไปนี้

- เมื่อมองไปทางซ้ายจะทำการคลิกซ้าย (left click)
- เมื่อมองไปทางขวาจะทำการคลิกขวา(right click)
- เมื่อมองลงจะทำการดับเบิลคลิก (double click)
- *เมื่อมองขึ้นจะทำการลาก (drag) และเมื่อถึงจุดที่ต้องการจะปล่อยก็ให้จ้องมองจุดนั้น* เป็นเวลา 3 วินาที เพื่อที่จะทำการปล่อย (drop)

การตรวจจับดวงตา ใช้ตาสีดำ (ตาคนเอเชีย) เป็นหลัก หากจะใช้กับตาสีอื่นที่ไม่ใช่สีดำต้องมีการปรับค่าเริ่มต้นใหม่ให้เหมาะกับตาสีนั้นๆ การจับภาพจำเป็นต้องมีการจำกัดสิ่งแวดล้อม ตัวอย่างเช่น ปริมาณแสงต้องพอเพียงที่จะสามารถแยกสีตาดำกับตาขาวได้ ตำแหน่งของดวงตาผู้ใช้กับ Web Camera ต้องมีการกำหนดระยะห่างที่แน่นนอน เป็นต้น

1.2 วัตถุประสงค์ของโครงงานวิจัย

วัตถุประสงค์

1. เพื่อให้บุคคลที่พิการที่ไม่สามารถใช้มือหรือเท้าควบคุมเมาส์ได้ สามารถใช้สายตาควบคุมเมาส์แทนได้ เช่น ผู้ที่เป็นอัมพาตตั้งแต่กระดูกสั้นหลังลงมา เป็นต้น
2. แนวทางในการพัฒนาเทคโนโลยีคอมพิวเตอร์ ในอนาคต ไม่จำเป็นต้องใช้มือในการควบคุมคีย์บอร์ด(Key Board) และ เมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัตถุประสงค์ในการศึกษา

1. เพื่อศึกษาภาษาวิชลเบสิก (Visual Basic)
2. เพื่อศึกษาการใช้ฟังก์ชันWindows API ด้วยภาษาวิชลเบสิก

1.3 ขอบเขตของโครงงาน

1 เมาส์ควบคุมด้วยดวงตา สามารถ คลิกขวา, คลิกซ้าย, ดับเบิลคลิก และ ลากและปล่อย (drag and drop) ได้

2 เมาส์ควบคุมด้วยดวงตา สามารถจับการเคลื่อนไหวของดวงตาได้

1.4 รายละเอียดโครงงาน

1 ภาษาที่ใช้ในการเขียนโปรแกรม : Microsoft Visual Basic 6.0

(สาเหตุ: ที่เลือกใช้ Visual Basic 6.0 เนื่องจาก Visual Basic 6.0 มีความสามารถในการรองรับ API สูงและมีใช้อย่างแพร่หลาย)

2 รายละเอียดของกล้องวีดีโอ :Intel USB Video Camera III

ความละเอียด : 640 x 1024

ระบบสี : RGB

1.5 วิธีการดำเนินงาน

งานวิจัยในโครงงานนี้จะเริ่มด้วยการศึกษาทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องกับงานวิจัย ซึ่งก็มีเรื่องหลักๆ อยู่ 3 เรื่องด้วยกัน คือ ความรู้เรื่อง API Win32 ในบทที่2, ความรู้เรื่อง โครงสร้างและการจัดการรูปภาพในบทที่ 3, การประมวลผลทางรูปภาพ เรื่อง Similarity Segmentation : Thresholding ในบทที่ 4, ศึกษาการเขียนโปรแกรมเชื่อมต่อกับ Web Camera ด้วยโปรแกรมวิชวลเบสิก ในบทที่ 5

จากนั้นนำเอาความรู้ที่เรียบเรียงมา มาสู่ขั้นตอนพัฒนาโปรแกรม โดยบทที่ 6 จะกล่าวถึงการออกแบบโปรแกรม และจะกล่าวถึงการขยายภาพ (Zoom) , การระบุตำแหน่งจ้องมองของดวงตาบนหน้าจอ(Center) ,การกำจัดสิ่งรบกวน (Noise), การแสดงผลตำแหน่งบนหน้าจอ(Mapping) ,การคลิก ซึ่งจะมีรายละเอียดอยู่ในบทที่ 7, 8,9,10 และ 11 ตามลำดับ

สำหรับบทที่ 12 ก็จะเป็นการทดสอบระบบรวมทั้งหมด และบทที่ 13 ซึ่งเป็นบทสุดท้ายก็เป็นการสรุปการทำงาน ผลที่ได้รับจากงานวิจัยชิ้นนี้ แนวทางในการพัฒนาวิจัยเพิ่มเติม และแนวทางในการนำไปประยุกต์ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
การเขียนโปรแกรมติดต่อ
ระหว่างแอปพลิเคชั่น กับวินโดวส์โดยใช้ API

แม้ว่าภาษาวิชวลเบสิก จะเตรียมฟังก์ชั่น และความสามารถต่างๆ ไว้อย่างมากมาย แต่อาจไม่ตอบสนองความต้องการทั้งหมด ดังนั้นวิชวลเบสิกจึงยอมให้ผู้พัฒนาแอปพลิเคชั่น สามารถที่จะเรียกใช้ความสามารถ ของระบบปฏิบัติการทั้ง Windows 95/98 ,NT Workstation ได้อย่างเต็มที่ โดยการเรียกผ่านฟังก์ชั่น Windows API

2.1 API คือ

API ย่อมาจาก (Application Programming Interface) ซึ่งเป็นส่วนเชื่อมต่อระหว่างแอปพลิเคชั่น กับ วินโดวส์โดยจะแสดงลักษณะการติดต่อสื่อสารกันระหว่างแอปพลิเคชั่น กับวินโดวส์ ทำให้เข้าใจได้ว่าแอพพลิเคชั่น กับวินโดวส์มีการติดต่อสื่อสารกันอย่างไร โดยมีลักษณะการทำงานเป็นแบบ Event-Driven Programming ซึ่งมีลักษณะการทำงานที่ทำให้ผู้ใช้เป็นอิสระแตกต่างจากแบบเดิมที่เป็น Sequence Programming และเป็นแบบ Graphic Oriented โดยจะมีมีฟังก์ชั่นอยู่มากมายนับพันๆ ฟังก์ชั่น ถูกแบ่งออกเป็นหมวดหมู่ตามประเภทของงาน เช่น API ด้านกราฟฟิก , API ด้านเน็ตเวิร์ค เป็นต้นเรา

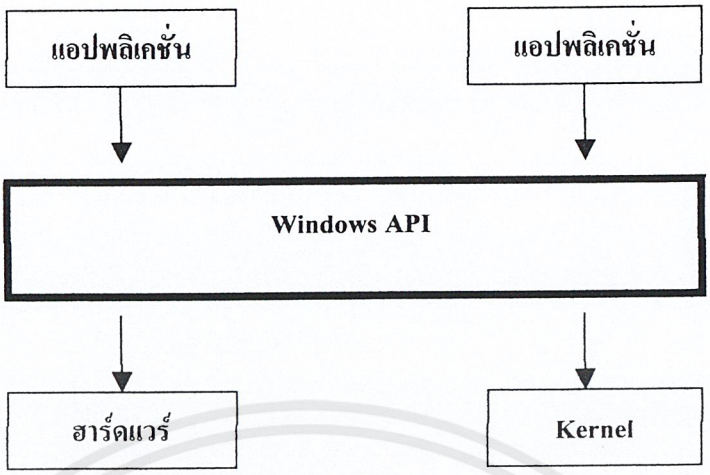
สำหรับสาเหตุที่ต้องมี Windows API ก็เพราะว่า เราจะได้มีวิธีการมาตรฐานในการใช้งานวินโดวส์แบบเดียว ซึ่งจะช่วยลดเวลาของเราที่ต้องไปศึกษาอุปกรณ์ และองค์ประกอบต่างๆ ทำให้เราไม่จำเป็นต้องรู้ว่าจอภาพทำงานอย่างไร, แป้นพิมพ์ทำงานอย่างไร เมาส์ทำงานอย่างไร แต่สามารถเรียกใช้งานผ่าน Windows API ได้ทันที

อีกหนึ่งข้อดีของ Windows API ที่น่าสนใจก็คือ ช่วยลดความผิดพลาดจากการเข้าถึงฮาร์ดแวร์โดยตรง เพราะเราจะสั่งงานฮาร์ดแวร์ผ่าน Windows API เท่านั้น ซึ่งช่วยลดความสับสนในการใช้งานฮาร์ดแวร์ที่ผิดวิธีไปได้มาก

นอกจากนี้ยังมีข้อดีของ Windows API ได้แก่ ความเป็นมาตรฐานที่ทุกคนรู้จัก จึงง่ายต่อการแก้ไขเปลี่ยนแปลง เช่น แม้ว่าเราจะเปลี่ยนไปใช้ฮาร์ดแวร์ใหม่ หรือแม้แต่ระบบปฏิบัติการเวอร์ชันใหม่ๆ แล้วก็ตาม ชื่อของฟังก์ชั่นเหล่านี้ ยังคงใช้ชื่อเดิมทำงานได้เหมือนเดิม ทำให้เราไม่ต้องกังวลกับความเปลี่ยนแปลงที่จะเกิดขึ้นจนบ่อยเกินไปนัก

ซึ่งมันก็คือ ชุดของฟังก์ชั่นที่พร้อมให้เราเรียกใช้งาน โดยที่ฟังก์ชั่นเหล่านั้นรับประกันได้ว่าทำงานอย่างมีประสิทธิภาพ เช่น เราไม่จำเป็นต้องเขียนโปรแกรมให้ยืดยาวเพื่อตรวจสอบว่าเนื้อที่ฮาร์ดดิสก์เหลือว่างอยู่เท่าไร แต่เรียกใช้ฟังก์ชั่น GetDiskFreeSpace ฟังก์ชั่นเดียวก็รู้แล้ว เป็นต้น ซึ่งในรูปที่ 2-1 จะแสดงหลักการทำงานของ Windows API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

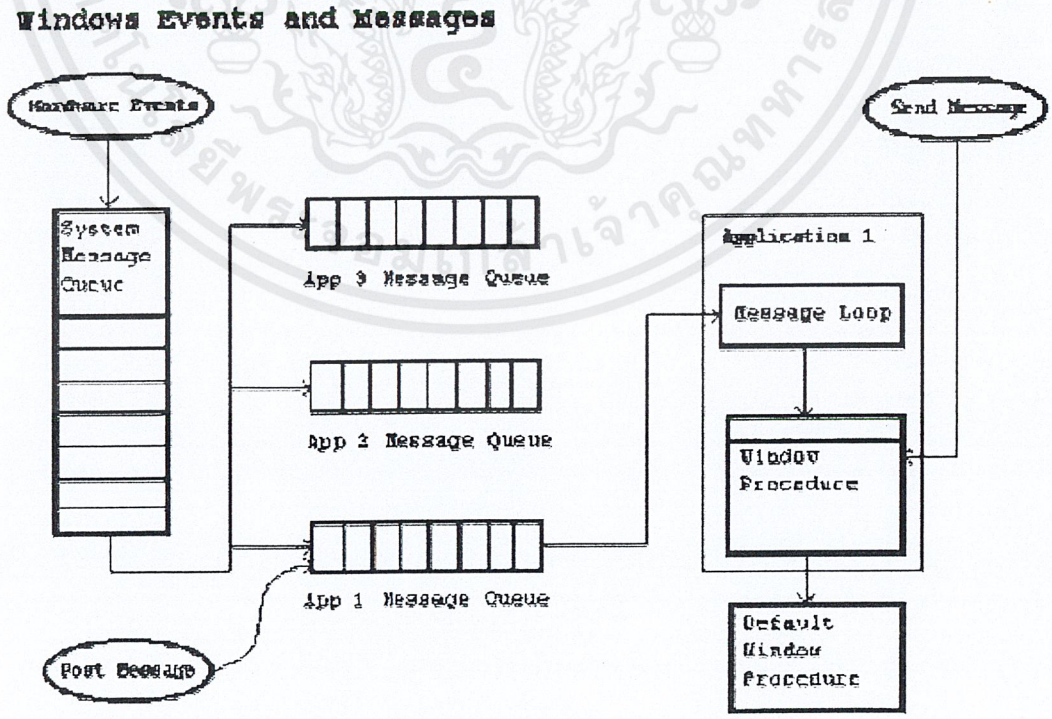


รูปที่ 2-1 แสดงหลักการทำงานของ Windows API

โดยในขั้นต้นเรามาทำความเข้าใจเกี่ยวกับการทำงานแบบ Event-Driven Programming กันก่อน

2.2 ลักษณะของ Event-Driven Programming

มีลักษณะการทำงานที่ไม่เป็นลูกโซ่ที่ต้องทำตามลำดับ โดยโปรแกรมจะทำงานตามเหตุการณ์ที่ผู้ใช้กระทำ โดยที่โปรแกรมจะไม่เป็นตัวชักจูงทำตามผู้ใช้ให้ทำตามที่โปรแกรมตั้งไว้ โดยที่ระบบปฏิบัติการจะเป็นผู้ตรวจสอบเหตุการณ์ที่เกิดขึ้นแล้วส่งข้อมูลของเหตุการณ์นั้น(Message) ไปให้ยังโปรแกรม โดยที่โปรแกรมจะทำงานตามข้อมูลของเหตุการณ์นั้น ดังจะแสดงให้เห็นในรูปที่ 2-2



รูปที่ 2-2 แสดงการทำงานของ API Programming โดยทำงานแบบ **Event-Driven Programming**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ลักษณะโครงสร้างการทำงานของ Windows API

ลักษณะการทำงานของ Windows API จะมีขั้นตอนคราวๆดังนี้

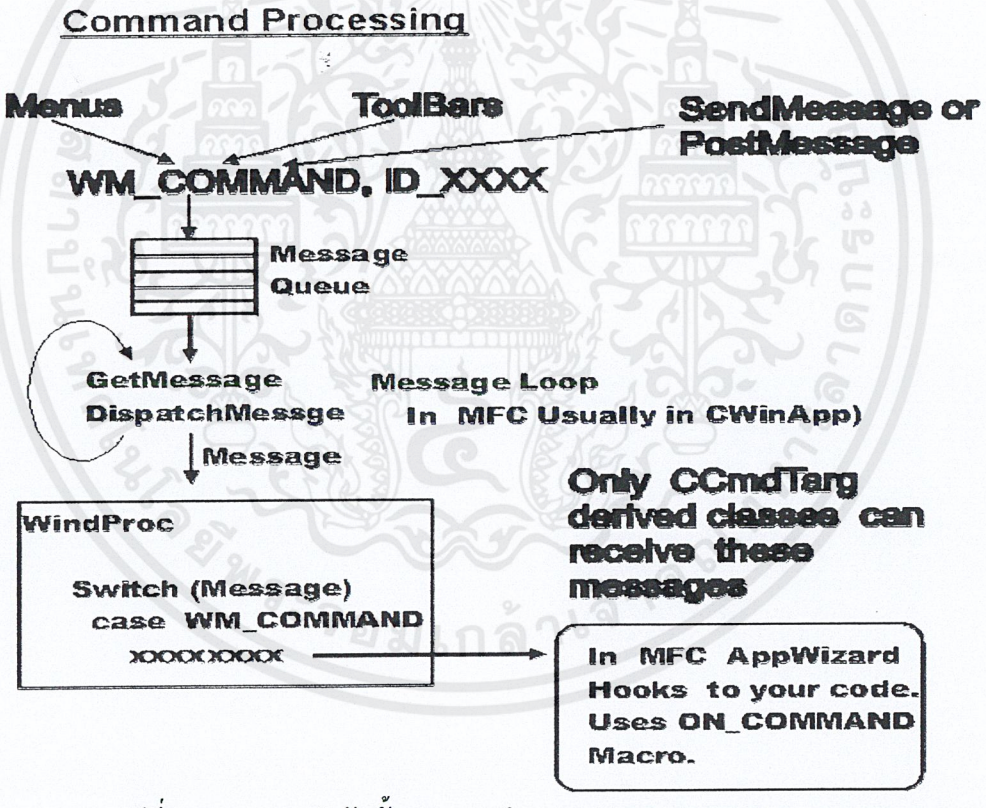
1. กำหนดค่าตัวแปรเริ่มต้นและเนื้อที่ของหน่วยความจำ (Initial Variable and Memory Space)
2. สร้างและแสดงหน้าต่างของโปรแกรม
3. จะเข้า Loop การทำงาน เพื่อรอรับ Message ของทำงาน

   - เพื่อรอรับ Message การทำงานที่สิ่งมาจากระบบปฏิบัติการมายัง แอปพลิเคชั่นของเรา

   - ถ้า Message คือ WM_QUIT จะจบการทำงานและคืนค่าการควบคุมกลับไปยังระบบปฏิบัติการ

   - ทำ Message เป็นคำสั่งที่มาจากผู้ใช้ ก็จะทำตามข้อมูลที่ส่งมานั้นๆ

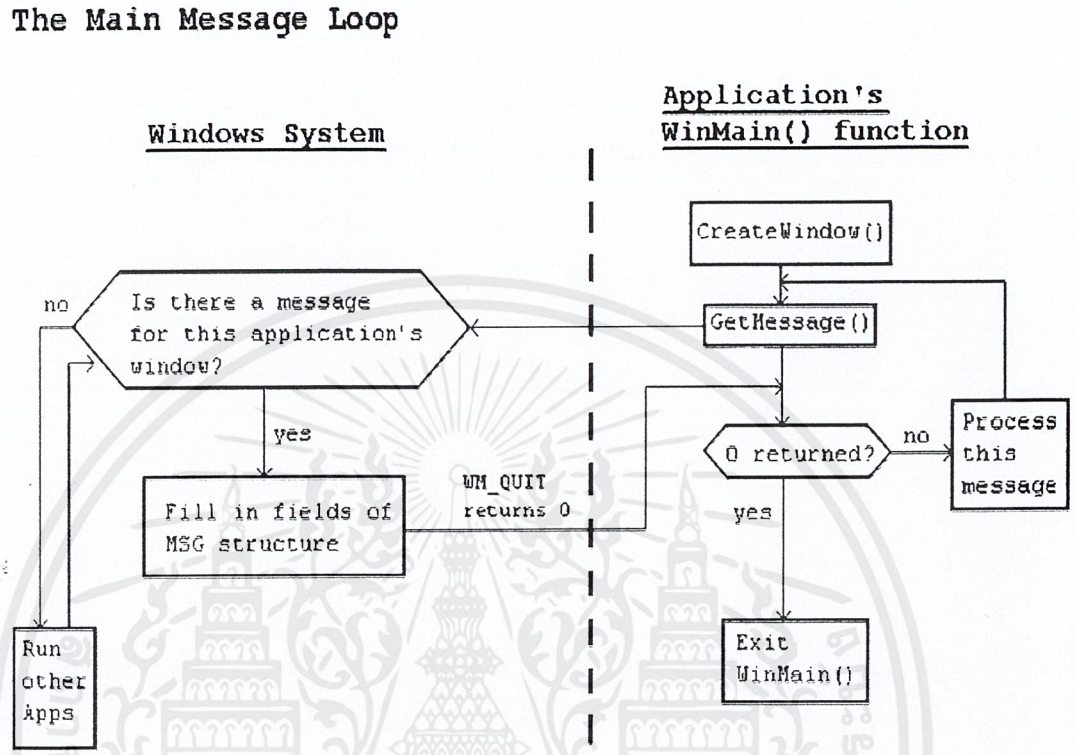
โดยขั้นตอนการทำงานจะแสดงในรูปที่ 2-3 และ 2-4



รูปที่ 2-3 แสดงการลำดับขั้นตอนการทำงานของ Windows API Programming

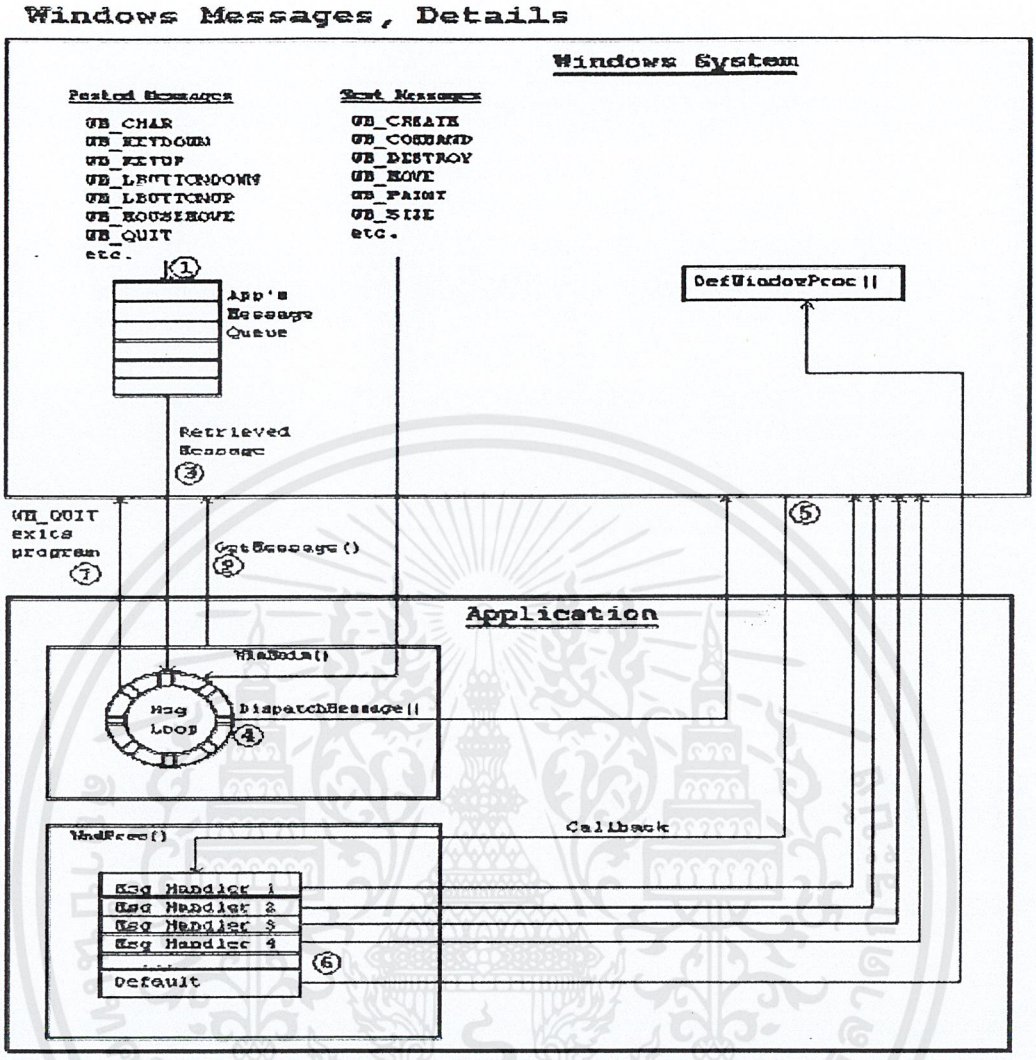
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยรูปที่ 2-4 จะแสดงขั้นตอนการทำงานของโปรแกรมจนจบการทำงาน โดยจะจบการทำงานเมื่อได้รับ Message “WM_QUIT”



รูปที่ 2-4 แสดงขั้นตอนการทำงานของ Windows API Programming ตั้งแต่เริ่มต้นจนจบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-5 แสดงการทำงานของ Message ชนิดต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในหัวข้อนี้เราจะเรียนรู้วิธีการเรียกใช้ฟังก์ชั่น API พร้อมทั้งบอกถึงฟังก์ชั่นที่ถูกนำมาใช้งาน

2.4 กลุ่มฟังก์ชั่นของ Windows API

Windows 95/98, NY Workstations จะมีฟังก์ชั่นในรูปของ Windows API ให้ใช้มากกว่า 1000 ฟังก์ชั่น ซึ่งฟังก์ชั่นต่างๆ จะถูกเก็บในไฟล์ .DLL ซึ่งอยู่ในโฟลเดอร์ C:\Windows\Systems ซึ่งมีไฟล์สำคัญๆ ได้แก่

ชื่อไฟล์	คำอธิบาย
ADVAPI32.DLL	มาจาก Advanced API ซึ่งจะเก็บฟังก์ชั่นเกี่ยวกับความปลอดภัย และ Registry
COMDLG.DLL	เก็บฟังก์ชั่นของ Common Dialog
GDI32.DLL	GDI ซึ่งจะเก็บฟังก์ชั่นเกี่ยวกับการแสดงผล และฟังก์ชั่นเกี่ยวกับกราฟฟิก
KERNEL32.DLL	เก็บฟังก์ชั่นที่ใช้จัดการทรัพยากรของระบบปฏิบัติการ
LZ32.DLL	เก็บฟังก์ชั่นที่ใช้เกี่ยวกับการบีบอัดข้อมูล
MPR.DLL	เก็บฟังก์ชั่นที่เกี่ยวกับ Multiple Provide Router
NETAPI32.DLL	เก็บฟังก์ชั่นที่เกี่ยวกับการจัดการเครือข่าย (Network)
SHELL32.DLL	เก็บฟังก์ชั่นที่เกี่ยวกับเชลล์ (Windows Explorer)
USER32.DLL	เก็บฟังก์ชั่นที่เกี่ยวกับการติดต่อกับผู้ใช้งาน (User Interface)
VERSION32.DLL	เก็บฟังก์ชั่นที่เกี่ยวกับการจัดการรุ่นของซอฟต์แวร์
WINMM.DLL	เก็บฟังก์ชั่นที่เกี่ยวกับมัลติมีเดีย
WINSPOLL.DRV	เก็บฟังก์ชั่นที่เกี่ยวกับการพิมพ์ผ่านเครื่องพิมพ์

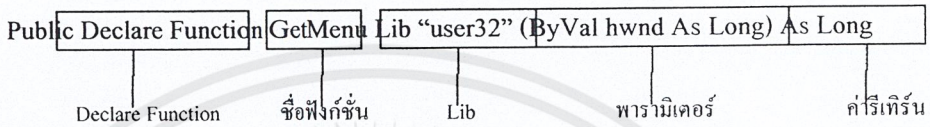
ตารางที่ 2-1 แสดงตารางเกี่ยวกับไฟล์ DLL ที่สำคัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การประกาศฟังก์ชั่นใน Windows API

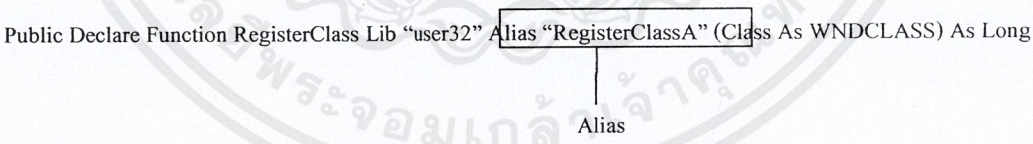
ในการเรียกใช้ฟังก์ชั่นต่างๆ ของ Windows API นั้น มีขั้นตอนที่แตกต่างจากการใช้เรียกฟังก์ชั่นที่วิชวลเบสิกเตรียมไว้ให้ หรือฟังก์ชั่นที่เราสร้างขึ้นมาเองนั่นคือ ต้องมีการประกาศค่าฟังก์ชั่นก่อนการใช้งาน

สำหรับการใช้งานฟังก์ชั่นนั้น เราจะคัดลอกมาจาก API Viewer ซึ่งฟังก์ชั่นที่คัดลอกมานั้น มีรูปแบบการประกาศดังนี้



จากตัวอย่าง เป็นการประกาศฟังก์ชั่น GetMenu และฟังก์ชั่น RegCloseKey จะเห็นว่าฟังก์ชั่นที่ประกาศนั้น ประกอบด้วยองค์ประกอบต่างๆ ดังนี้

- Declare Function เป็นคำสั่งแสดงว่านี้คือ การประกาศฟังก์ชั่น
- ชื่อฟังก์ชั่น เป็นชื่อของฟังก์ชั่น จากตัวอย่างนี้ ชื่อของฟังก์ชั่น คือ GetMenu และ RegClose
- Lib เป็นการบอกให้ทราบว่าให้ค้นหาจากไฟล์ .DLL ซึ่งจะต้องระบุชื่อไฟล์ .DLL นั้น ยกเว้นอยู่ในไฟล์ USER32.DLL, KERNEL32.DLL และ GDI32.DLL
- พารามิเตอร์ เป็นรายการของพารามิเตอร์ที่ต้องมีการผ่านค่าให้ฟังก์ชั่น
- ค่าที่รีเทิร์น เป็นชนิดข้อมูลที่จะคืนค่าให้กับผู้เรียกใช้ จากตัวอย่างมีชนิดข้อมูลเป็น Long ในบางฟังก์ชั่นนั้น อาจมีองค์ประกอบอื่นเพิ่มเติม เช่น



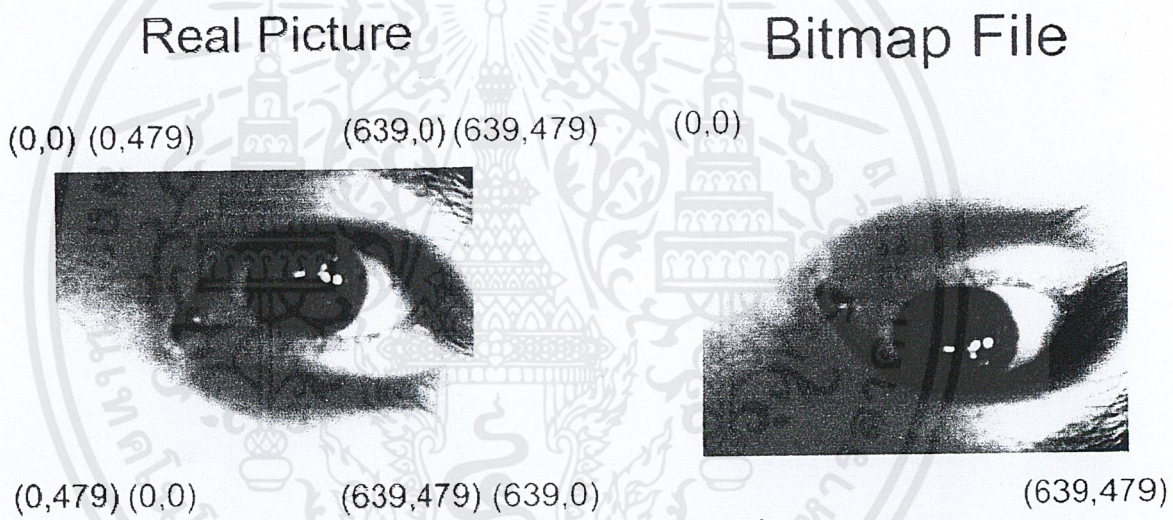
- Alias เป็นชื่อที่ระบบปฏิบัติการเรียกใช้ฟังก์ชั่นนี้ จากตัวอย่างเราจะเรียกใช้งานในแอปพลิเคชั่นของเราในชื่อฟังก์ชั่น RegisterClass แต่ Windows 95/98 จะรู้จักฟังก์ชั่นเดียวกันนี้ในชื่อ RegisterClassA เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
โครงสร้างและการจัดการรูปภาพ

3.1 โครงสร้างของภาพชนิด Bitmap

ภาพประเภท Bitmap Graphic นั้น เป็นรูปแบบหนึ่งของวิธีการเก็บภาพที่มีลักษณะเป็นพื้นฐานที่มีรูปแบบเรียบง่ายที่สุด โดยจะเก็บรูปเป็นลักษณะของสี่เหลี่ยม ซึ่งสามารถสร้างได้จากโปรแกรมวาดรูปทั่วๆไป โครงสร้างของภาพจะเก็บเป็นเมตริกซ์ของค่า pixel ของสีแต่ละจุดในหน่วยความจำ โดยสนับสนุนค่าpixel ที่เป็น 8 บิต และ 16 บิต (indirect) หรือ 24 บิต (direct) โดยสามารถเก็บอยู่ในไฟล์ชนิด .bmp โดยโครงสร้างของไฟล์ชนิด .bmp จะเก็บอยู่ในลักษณะที่ไม่เหมือนกับภาพที่แสดงผลออกมาโดยโปรแกรมวาดภาพต่างๆ ดังแสดงในรูปที่ 3-1 โดยที่ภาพต่างกันแกน Y คือกลับตรงกันข้ามกัน



รูปที่ 3-1 แสดงลักษณะการเก็บภาพที่อยู่ในไฟล์เปรียบเทียบกับภาพที่แสดงจริงโดยใช้โปรแกรมต่างๆ

ในการเขียนโปรแกรมวิชวลเบสิก 6.0 เพื่อที่จะดึงภาพชนิด Bitmap ขึ้นมาใช้งานที่จะต้องมีขั้นตอนทำดังต่อไปนี้

1. เปิดไฟล์ที่ต้องการมาพิจาณาชนิดที่เป็น Bitmap

```
OpenFile = Text1.Text
Open OpenFile For Binary As #1
```

โดยทำการเปิดภาพขึ้นมาเป็นชนิดไบนารี่ (binary) โดยดึงชื่อไฟล์มาจากเท็กซ์บ๊อกซ์ที่ชื่อว่า “Text1” มาใส่ตัวแปรชื่อ #1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ดึงส่วนหัว(header file)ของไฟล์ชนิด Bitmap มาใส่ในตัวแปลที่ h1,h2 ซึ่ง h2 จะการเก็บข้อมูลของขนาดของภาพเอาไว้

```
Dim h1 As BITMAPFILEHEADER
Dim h2 As BITMAPINFOHEADER
Dim XRes As Long 'Width pic
Dim YRes As Long 'Hight pic
Get #1, , h1
Get #1, , h2
XRes = h2.biWidth
YRes = h2.biHeight
```

นำขนาดของภาพมาใส่ที่ตัวแปรที่ชื่อว่า XRes ,YRes โดยที่ XRes คือ ขนาดความกว้างของภาพ และ YRes คือความสูงของภาพ

3. ดึงภาพมาใส่ตัวแปรชนิดอาร์เรย์ (Array) ซึ่งจะนำมาใช้ในการพิจารณาแต่ละจุดของภาพมีสีอะไร โดยตะละจุดจะประกอบด้วยค่า RGB ซึ่งคือค่าสี ประกอบด้วย แดง เขียว น้ำเงิน ตามลำดับ

```
Global RGBPalette() As Long 'Color Table
Global pic()  As Long    'pixel
ReDim pic(XRes, YRes, 2) As Long
ReDim RGBPalette(255, 3) As Long
For i = 0 To YRes - 1
  For j = 0 To XRes - 1
    Get #1, , RGBentry
    pic(j, i, 0) = Asc(Mid$(RGBentry, 3, 1))
    pic(j, i, 1) = Asc(Mid$(RGBentry, 2, 1))
    pic(j, i, 2) = Asc(Mid$(RGBentry, 1, 1))
  Next
Next
```

โดยเราจะนำค่าสีในแต่ละจุดมาใส่ในตัวแปรที่ชื่อว่า “pic” ซึ่งเป็นอาร์เรย์ขนาดเท่ากับขนาดความใหญ่ของภาพ

4. เราสามารถดึงค่าสีของแต่ละจุดมาพิจารณาเพื่อเป็นพื้นฐานขั้นต้นของการหาจุดศูนย์กลางของดวงตาก่อนที่จะนำไปใช้กับภาพเคลื่อนไหวโดยจะกล่าวขั้นตอนการหาจุดศูนย์กลางของดวงตาและกำจัดสิ่งรบกวนเพื่อให้ได้จุดศูนย์กลางที่ถูกต้องในบทที่ 8,9 ต่อไป ตามลำดับ

หากท่านต้องการทราบการเก็บข้อมูลชนิด Bitmap อย่างละเอียด ท่านสามารถศึกษาได้จากส่วนที่เป็นภาคผนวก ก ท้ายเล่ม

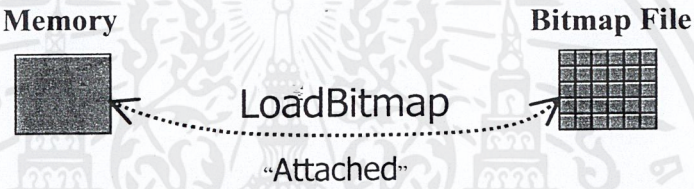
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การสร้างพื้นที่ในหน่วยความจำเพื่อเก็บภาพ

ในการดึงภาพขึ้นมานั้นเราอาจไม่จำเป็นต้องสร้างตัวแปรชนิดอาร์เรย์(Array) มาเก็บภาพเพื่อที่จะนำมาพิจารณา แต่เราสามารถสร้างเนื้อที่ในหน่วยความจำมาเก็บภาพ ซึ่งจะมีสะดวกและความรวดเร็วต่อการทำงานของโปรแกรมมากขึ้น

ซึ่งเนื้อในหน่วยความจำที่สร้างเพื่อเก็บรูปนั้นจะมีความสัมพันธ์กับการแสดงภาพบนจอ โดยการแสดงภาพบนจอนั้นปกติเราจะไม่สามารถเข้าถึงโดยตรงกับอุปกรณ์ได้(Hardware Device) เราต้องเขียนภาพผ่าน "Generic Drawing Function" ซึ่งเป็น API ตัวหนึ่ง คือ "Graphic Device Interface หรือ GDI" ซึ่ง GDI จะทำหน้าแปลงเป็น Hardware Command ให้

โดยการวาดรูปนั้นเราจะต้องวาดรูปลงบน Device Context (DC) ซึ่งจะเป็นตัวแทนของอุปกรณ์ โดยจะเปรียบเสมือนผืนผ้าที่เราไว้สำหรับวาดรูปลงไป



รูปที่ 3-2 การโหลดภาพมาเก็บในหน่วยความจำ

ขั้นตอนในการสร้างพื้นที่ในหน่วยความจำเพื่อเก็บภาพ

1. ประกาศคำสั่งที่ใช้ในการจัดการพื้นที่ในหน่วยคำจำที่ใช้สำหรับเก็บภาพ

ก่อนจะใช้ฟังก์ชั่นได้จะต้องมีการประกาศใช้ตอนต้นของโปรแกรมก่อน โดยจะอธิบายแต่ละฟังก์ชั่น ดังนี้

Private Declare Function GetDC Lib "user32" (ByVal hwnd As Long) As Long

เป็นฟังก์ชั่นสำหรับการสร้างตัวชี้(Pointer) ให้ชี้ไปยัง DC นั้น เพื่อใช้ในการจัดการกับ DC ของส่วนที่นำออกมาเช่น ของจอภาพ ของพื้นที่ที่ใช้ในการแสดงรูป ในที่นี้จะใช้ในการดึงภาพจากพื้นที่ของ Component ของ ezVidCap ซึ่งเป็นภาพจริงจากกล้อง Web Cam

Private Declare Function DeleteDC Lib "gdi32" (ByVal hdc As Long) As Long

เป็นฟังก์ชั่นสำหรับลบค่าที่ใช้ในการชี้ Device Context ตัวนั้นอยู่

Private Declare Function CreateCompatibleDC Lib "gdi32" (ByVal hdc As Long) As Long

เป็นฟังก์ชั่นสำหรับการสร้างพื้นที่ในหน่วยความจำที่ในที่นี้เป็นObject ที่สอดคล้องกับDCของจอภาพ หรือพื้นที่ของ Component ของ ezVidCap

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Private Declare Function CreateCompatibleBitmap Lib "gdi32" (ByVal hdc As Long, ByVal nWidth As Long, ByVal nHeight As Long) As Long

เป็นฟังก์ชั่นสำหรับการสร้างObject ที่สอดคล้องกับภาพชนิด Bitmap

Private Declare Function SelectObject Lib "gdi32" (ByVal hdc As Long, ByVal hObject As Long) As Long

เป็นฟังก์ชั่นสำหรับเลือก Object เข้าไปใน Object ของ DC ในที่นี้ คือ ให้จองพื้นที่ในหน่วยความจำเพื่อใช้ในการเก็บรูปชนิด Bitmap

Private Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long) As Long

เป็นฟังก์ชั่นสำหรับลบ Object นั้นทิ้งไป

Private Declare Function SetPixel Lib "gdi32" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal crColor As Long) As Long

เป็นฟังก์ชั่นสำหรับใส่(Set) ค่าสีลงไปในหน่วยความจำเฉพาะจุดที่เราต้องการแก้ไขเปลี่ยนแปลงสีของจุดนั้น

Private Declare Function GetPixel Lib "gdi32" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long) As Long

เป็นฟังก์ชั่นสำหรับดึง(Get) ค่าสีออกมาจากหน่วยความจำเฉพาะจุดที่เราต้องการทราบเพื่อพิจารณาค่าสีของจุดนั้น

2. ต้องมีการสร้างพื้นที่ในหน่วยความจำ(Memory DC) ซึ่งสอดคล้องกับจอภาพ (Screen DC) และใช้ในการเก็บภาพประเภท Bitmap

โดยเราจะต้องมีการสร้างพื้นที่ในหน่วยความจำเพื่อเก็บภาพ โดยจะต้องสอดคล้องกับพื้นที่ที่จะแสดงรูปละมีชนิดของรูปภาพเป็น Bitmap

```
Dim hDC, hBMP As Long
hDC = CreateCompatibleDC(PictureB.hdc)
hBMP = CreateCompatibleBitmap(PictureB.hdc, 256, 192)
SelectObject hDC, hBMP
```

ในที่นี้ต้องการแสดงภาพไปยัง Picture Box ที่ชื่อว่า “PictureB” โดยจะต้องสร้างพื้นที่ให้สอดคล้องด้วยโดยตัวแปรที่ใช้จัดการคือ hDC และ สร้างObject ชนิด Bitmap ซึ่งมีขนาด 256x192 แล้วเลือกให้พื้นที่ในหน่วยความจำ(hDC) เอาไว้เก็บภาพชนิดBitmapและมีขนาด 256x192

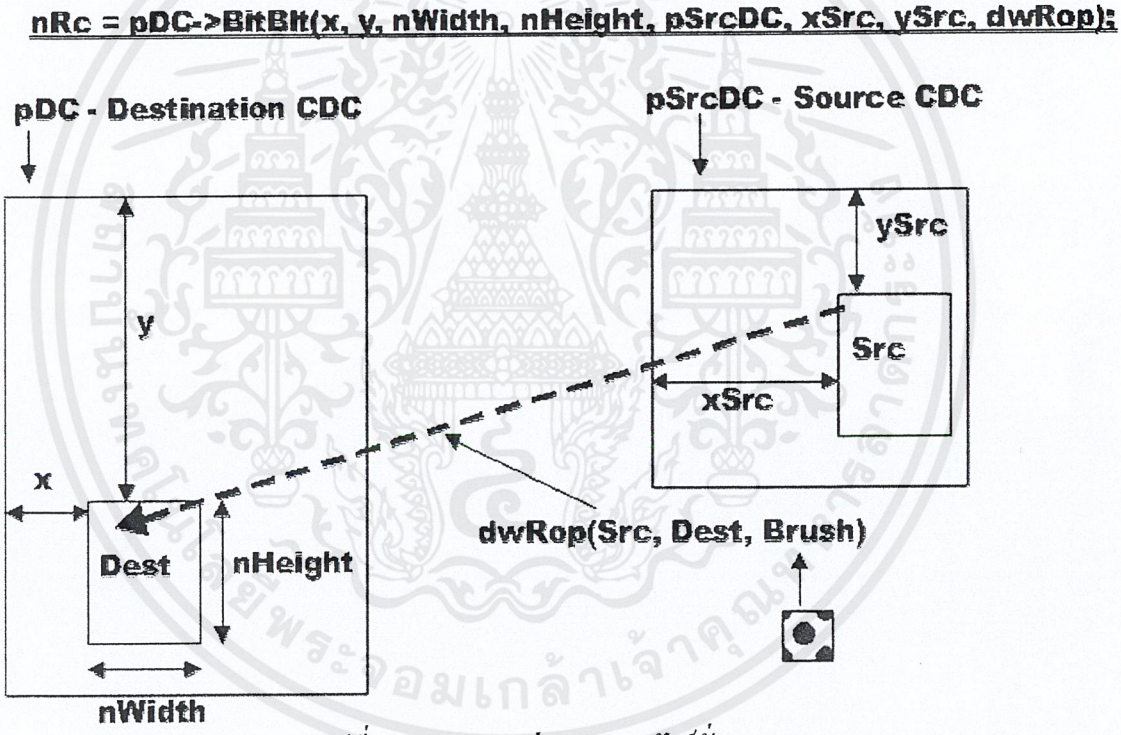
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การโหลดหรือส่งรูปภาพมาเก็บไว้ในหน่วยความจำที่ได้สร้างไว้

ในการส่งรูปลงในหน่วยความจำที่จองไว้นั้นเราจะใช้ฟังก์ชั่นที่ชื่อว่า BitBlt (Bitwise Block Transfer) ซึ่งจะทำการส่งค่าทั้งหมดไปยังเนื้อที่ในหน่วยความจำที่จองไว้ โดยที่ไม่ต้องใส่ลงไปทีละจุด (จะ Copy ระหว่าง Memory กับ Memory) ซึ่งจะมีความรวดเร็วอย่างมาก

โดยเราจะต้องประกาศในตอนต้นของโปรแกรมก่อนใช้งาน

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



รูปที่ 3-3 แสดงการส่งภาพของฟังก์ชั่น BitBlt

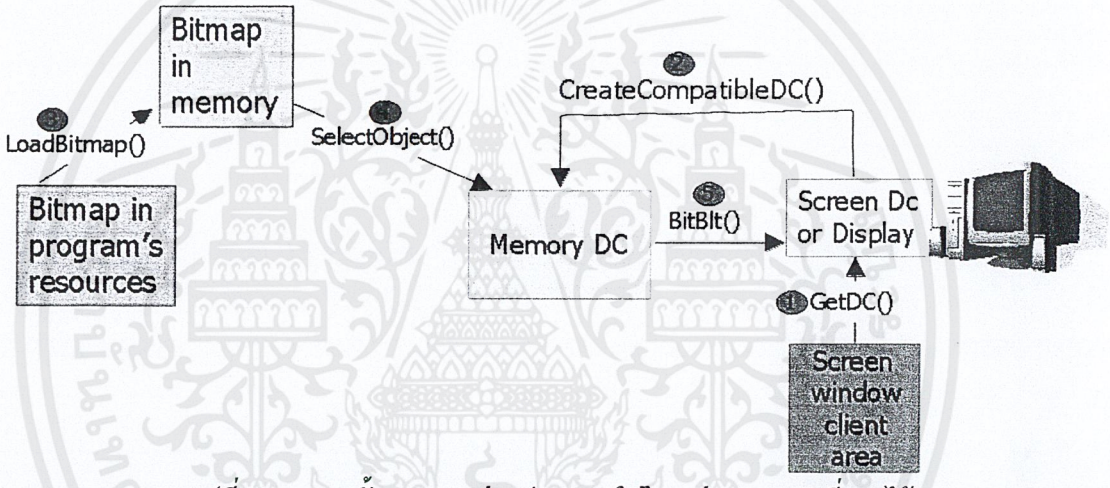
ในการทำงานของฟังก์ชั่น BitBlt จะต้องระบุตำแหน่งภาพของภาพต้นทางว่าจะเริ่มที่ไหน และตำแหน่งภาพของภาพปลายทางว่าเริ่มที่ไหน โดยที่ขนาดของภาพต้นทางและปลายทางจะมีขนาดเท่ากัน ดังจะแสดงในรูปที่ 3-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยตัวอย่างของการใช้ฟังก์ชั่น Bitblt คือ

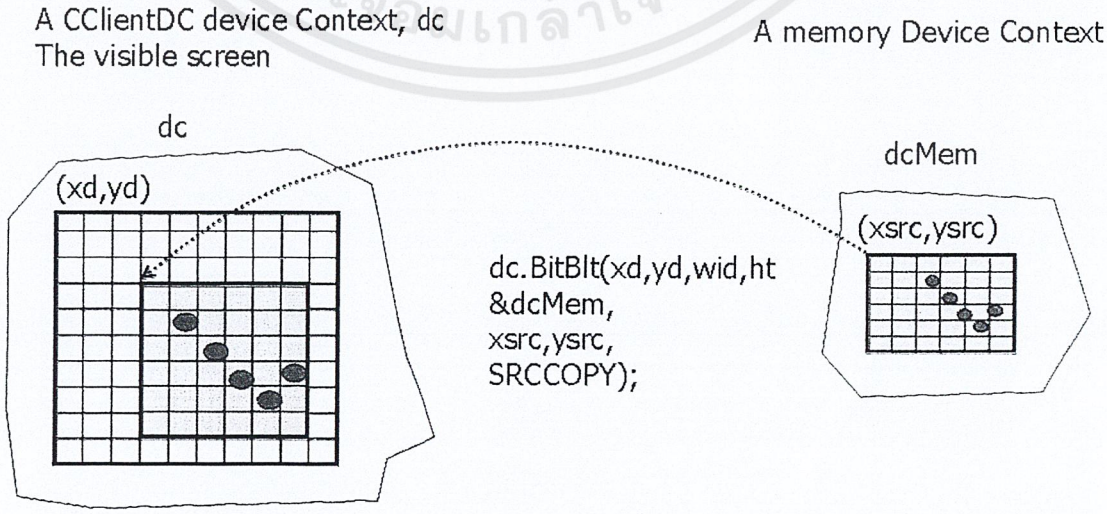
```
hDC = GetDC(ezVidCap1.hwnd)
Dim MemDC, MemBMP As Long
MemDC = CreateCompatibleDC(PictureB.hdc)
MemBMP = CreateCompatibleBitmap(PictureB.hdc, 256, 192)
SelectObject MemDC, MemBMP
BitBlt MemDC, 0, 0, Picture1.ScaleWidth, Picture1.ScaleHeight, hDC, 0, 0, vbSrcCopy
'No Zoom
```

โดยเราจะทำการสร้างตัวชี้ไปยังพื้นที่ของภาพจริงที่รับมาจากล้องจากนั้นนำมาใส่ในหน่วยคำจำที่ชื่อว่า "MemDC" ดังที่แสดงอยู่ในรูปที่ 3-4 แสดงขั้นตอนการส่งรูปภาพมาใส่ในหน่วยความจำที่จองไว้



รูปที่ 3-4 แสดงขั้นตอนการส่งรูปภาพมาใส่ในหน่วยความจำที่จองไว้

เราอาจจะใช้ฟังก์ชั่น Bitblt ในการโหลดค่าหรือส่งค่าจากหน่วยความจำมาแสดงผลบนพื้นที่ที่สร้างไว้บนหน้าจอของแอปพลิเคชั่นก็ได้ แสดงในรูปที่ 3-5



รูปที่ 3-5 แสดงการโหลดหรือส่งรูปภาพมาแสดงผลบนพื้นที่ที่สร้างไว้บนหน้าจอของแอปพลิเคชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

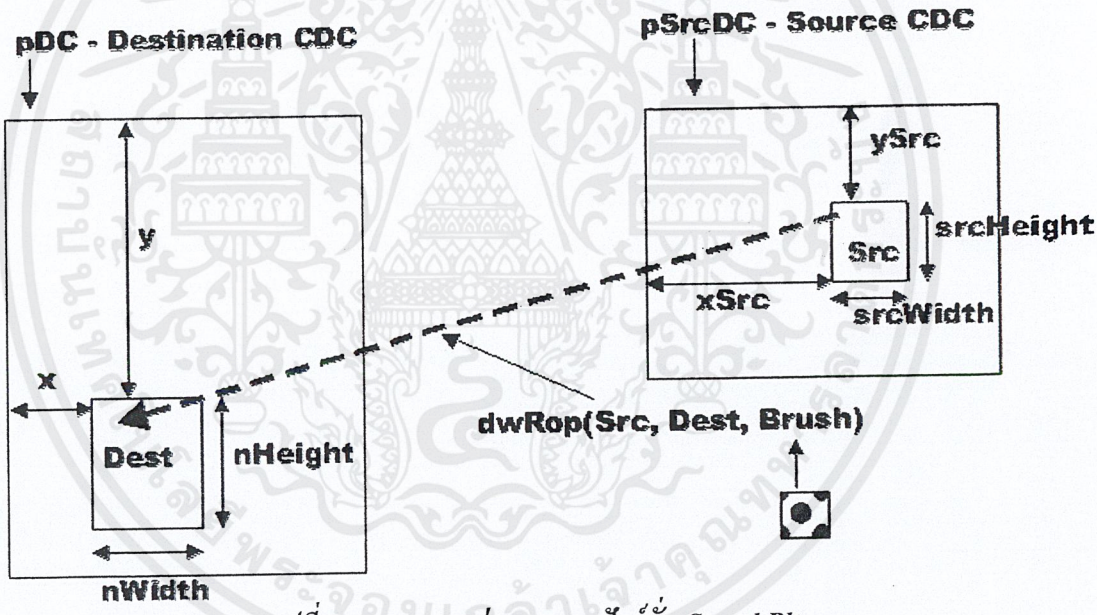
3.4 การลดหรือขยายขนาดของรูปภาพในหน่วยความจำ

เราสามารถลดหรือขยายขนาดของภาพได้โดยใช้ฟังก์ชั่น StretchBlt โดยที่ไม่ต้องใช้ทฤษฎีการขยายภาพแบบ First Order โดยการทำงานของฟังก์ชั่น StretchBlt นั้นจะเหมือนกับ BitBlt เพียงแต่จะสามารถขยายหรือลดขนาดของภาพได้

โดยเราจะต้องประกาศในตอนต้นของโปรแกรมก่อนใช้งาน

Declare Function StretchBlt Lib "gdi32" (ByVal hdc 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 nSrcWidth As Long, ByVal nSrcHeight As Long, ByVal dwRop As Long) As Long

nRc = pDC->StretchBlt(x, y, nWidth, nHeight, pSrcDC, xSrc, ySrc, srcWidth, srcHeight, dwRop);



รูปที่ 3-6 แสดงการส่งภาพของฟังก์ชั่น StretchBlt

ในการทำงานของฟังก์ชั่น StretchBlt จะต้องระบุตำแหน่งภาพของภาพต้นทางว่าจะเริ่มที่ไหน และตำแหน่งภาพของภาพปลายทางว่าเริ่มที่ไหนเช่นกัน เพียงแต่โดยที่ขนาดของภาพต้นทางแลปลายทางนั้นจะมีขนาดไม่เท่ากัน โดยเราสามารถระบุได้ว่าภาพต้นทางมีขนาดเท่านี้ ภาพปลายทางจะให้มีขนาดเท่าไหร่ได้ โดยจะแสดงในรูปที่ 3-6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

บทที่ 4

การประมวลผลทางรูปภาพ

เรื่อง Similarity Segmentation : Thresholding

4.1 Segmentation

Segmentation หมายถึง การแบ่งภาพออกเป็นส่วนๆ โดยการแบ่ง เราจะใช้คุณสมบัติพิเศษ (Feature) ของแต่ละภาพ เช่น ค่าความเข้มแสง,สี ,การเปลี่ยนแปลงความเข้มแสง(Gradient) , ลักษณะของพื้นผิว (Texture)

4.2 เทคนิค Segmentation

4.2.1 Discontinuity : การหาขอบของภาพ

4.2.2 Similarity : การหาส่วนที่เหมือนกันของภาพ

4.2.1 Discontinuity

4.2.1.1 First-Second Derivative (No noise)

4.2.1.2 First-Second Derivative (With noise)

4.2.1.3 First Derivative (Gradient)

4.2.1.4 Laplacian of Gaussian (LoG)

4.2.1.1 First-Second Derivative (ไม่มี noise)

รูปที่4-1 การหาขอบของภาพด้วยวิธี First-Second Derivative (ไม่มี noise)

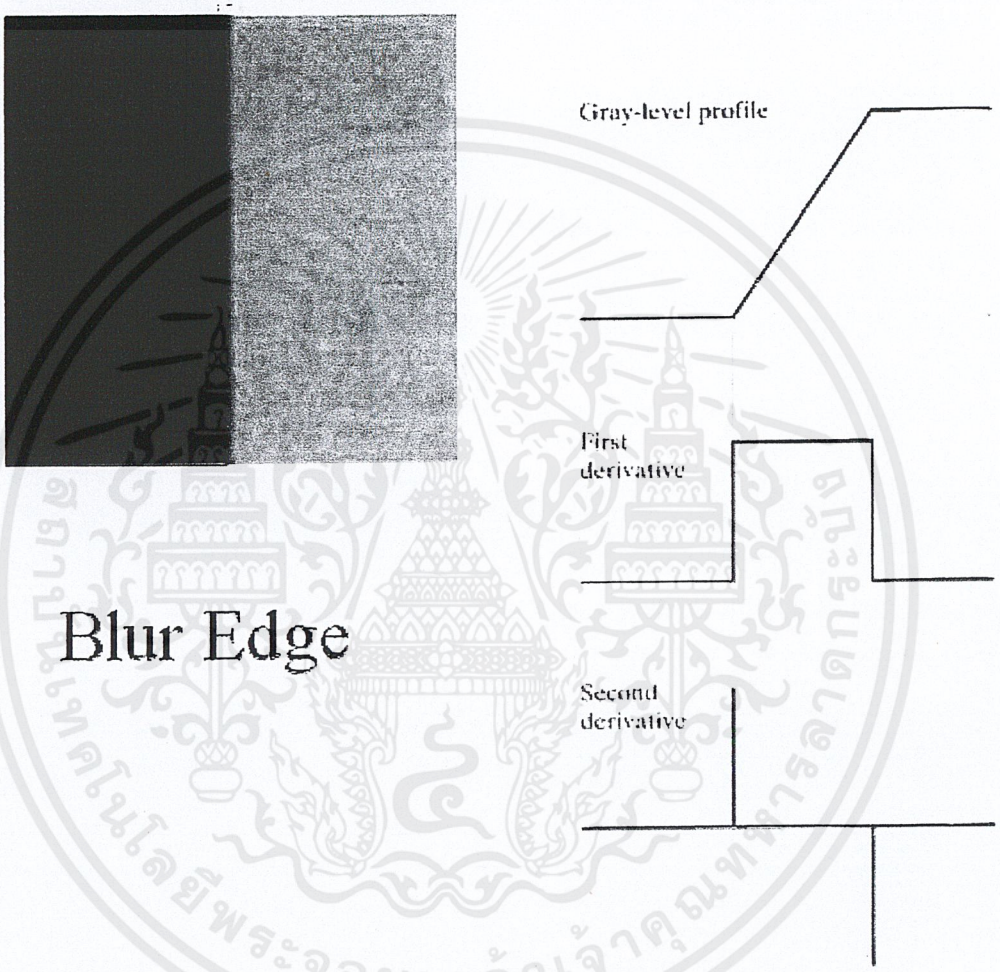
Sharp Edge เป็นกรณี ideal ไม่มี Noise มีแบ่งระหว่างความเข้มแสงมากกับความเข้มแสงน้อยอย่างชัดเจน ทำให้สามารถหาขอบได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

49982

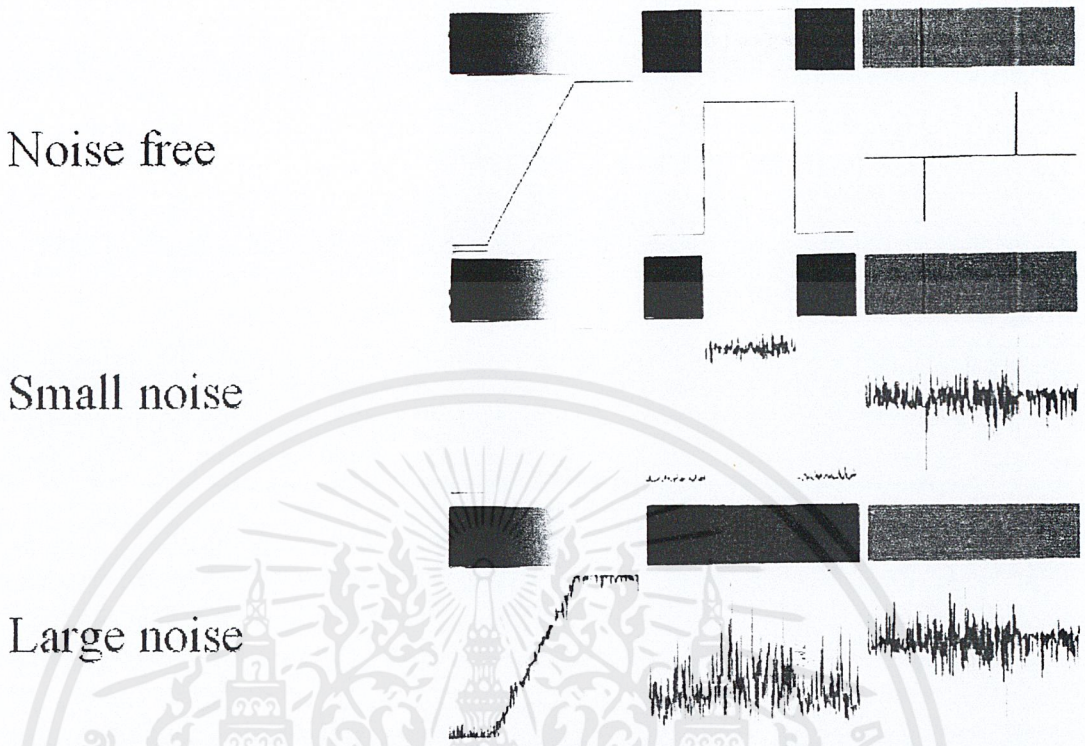
4.2.1.2 First-Second Derivative (มี noise)

ภาพที่เกิดจากการถ่ายภาพปกติ จะมี Noise คือไม่มีการแบ่งระหว่างความเข้มแสงมากกับน้อยอย่างชัดเจน แต่จะเป็นการเปลี่ยนแปลงความเข้มแสงจากมากค่อยๆลดเป็นน้อยหรือกรณีความเข้มแสงจากน้อยค่อยๆเพิ่มเป็นมาก เราสามารถกาขอบได้จากการเปลี่ยนแปลงนี้



รูปที่4-2 การหาขอบของภาพด้วยวิธี First-Second Derivative (มี noise)(1)

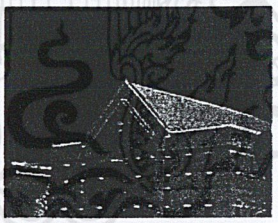
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



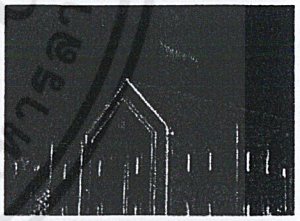
รูปที่4-3 การหาขอบของภาพด้วยวิธี First-Second Derivative (มี noise)(2)

4.2.1.3 First Derivative (Gradient)

Original image



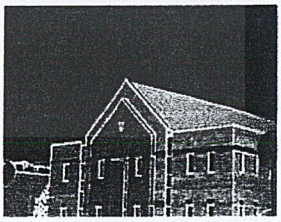
$F_x = f * h_x$



$F_y = f * h_y$

$$h_x(x,y) = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} or \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$h_y(x,y) = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} or \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$



$|F_x| + |F_y|$

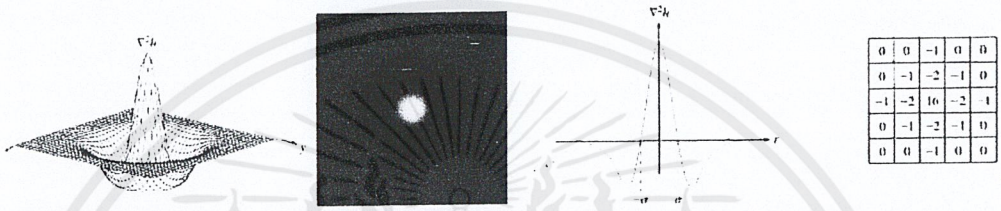
รูปที่4-4 การหาขอบของภาพด้วยวิธี First Derivative

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.4 Laplacian of Gaussian (LoG)

Laplacian	GLPF	LoG
$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$	$* \quad h(r) = -e^{-r^2/2\sigma^2}$	$= \quad \nabla^2 h(r) = -\left[\frac{r^2 - \sigma^2}{\sigma^4}\right] e^{-r^2/2\sigma^2}$

$$r^2 = (x - M/2)^2 + (y - N/2)^2$$



รูปที่4-5 การหาขอบของภาพด้วยวิธี Laplacian of Gaussian (LoG)

4.2.2 Simiailarity

4.2.2.1 Thresholding

4.2.2.2 Region Growing

4.2.2.3 Region Clustering

4.2.2.4 Region Splitting and Merging

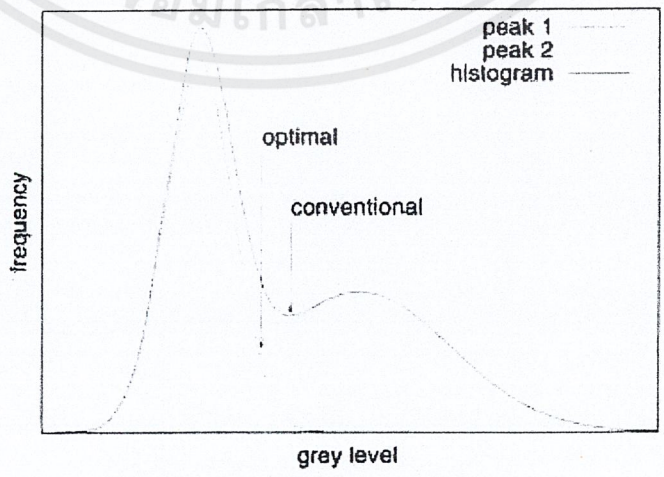
4.2.2.1 Thresholding

การแบ่งภาพออกเป็นส่วนๆ ส่วนที่สนใจจะถูกแบ่งโดยค่าสี

การหาค่า Threshold ทำได้ 2 วิธี

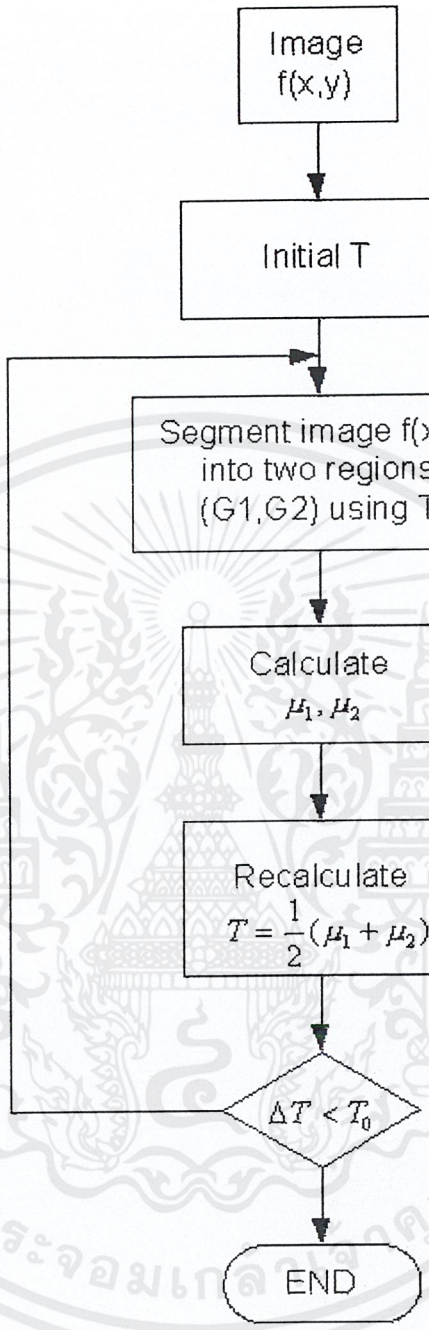
1.มาจากการทดลอง

2. Optimum Threshold Estimation



รูปที่4-6 Histogram แสดงจุด Optimal

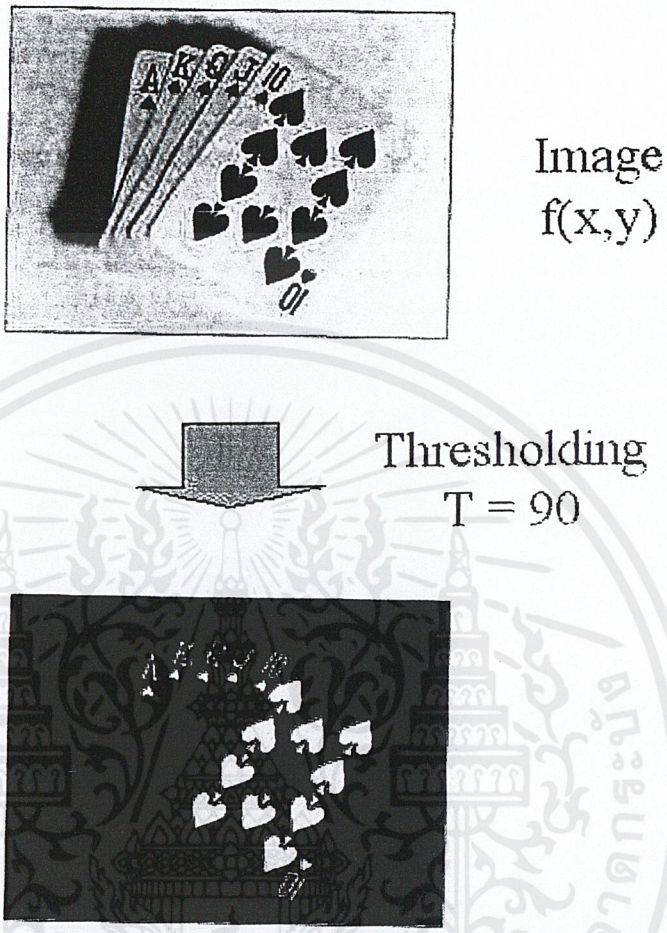
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4-7 Flow Chart แสดงการหา Optimum Threshold Estimation

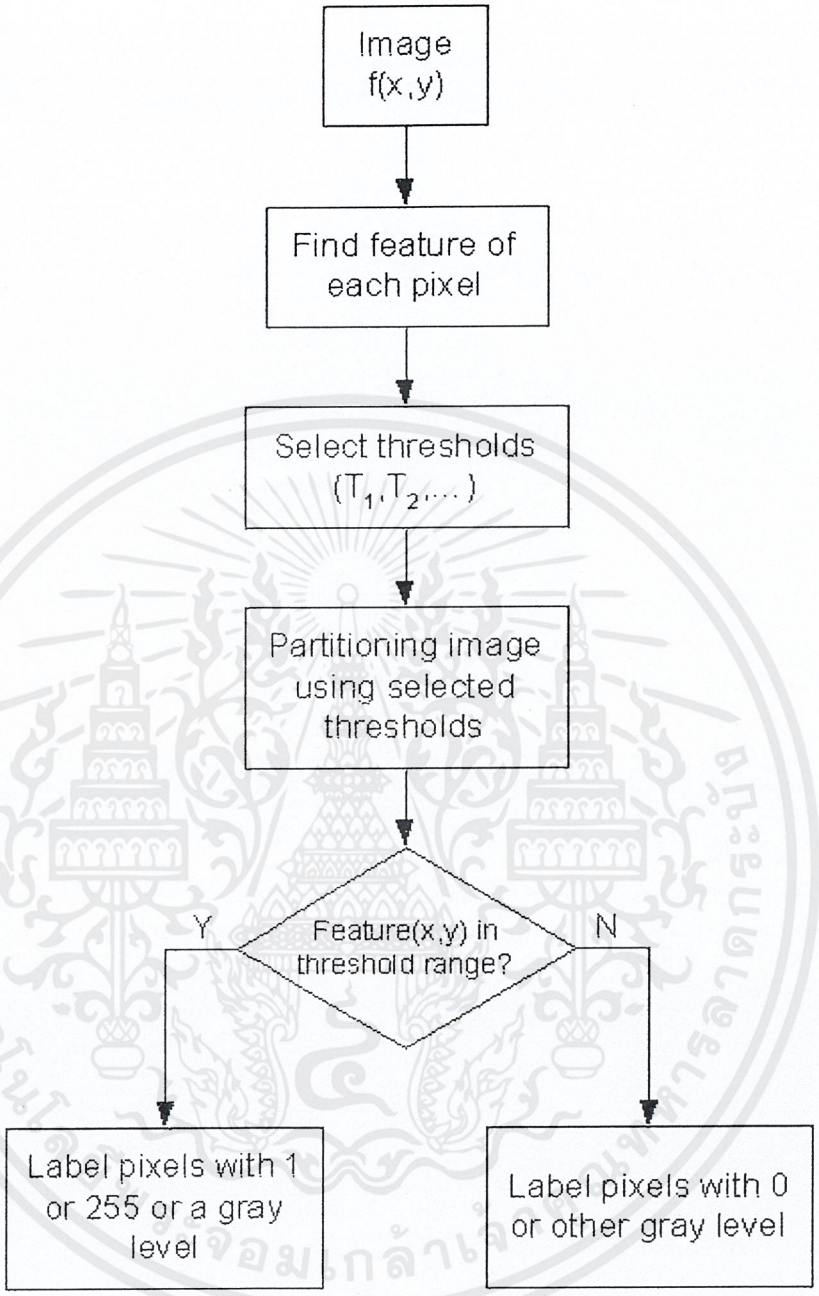
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thresholding ภาพ Gray Scale



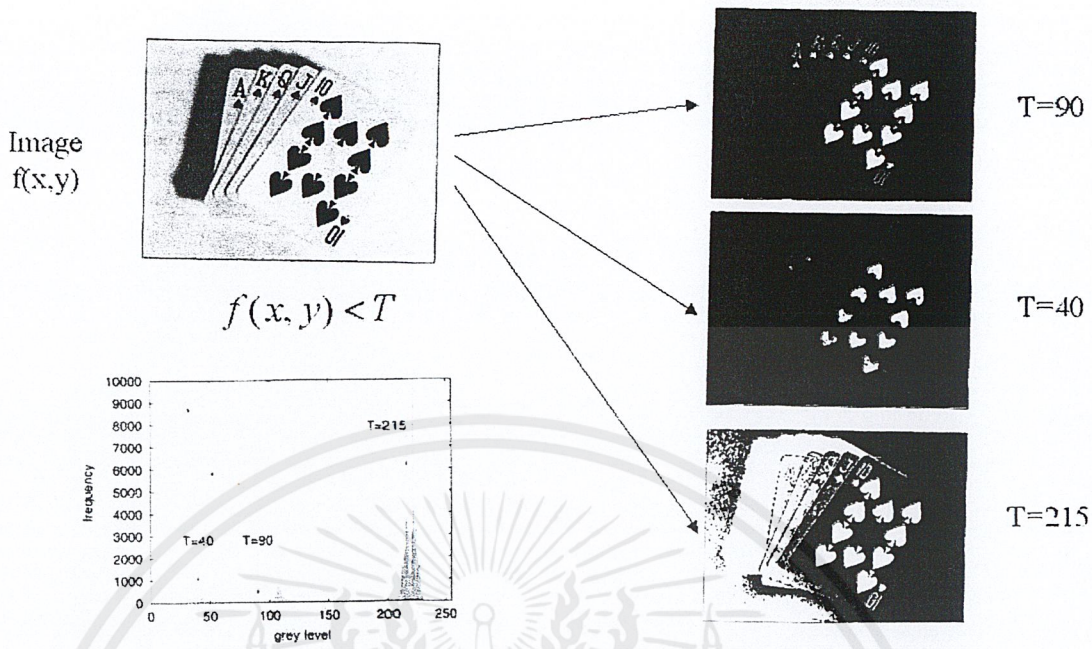
รูปที่ 4-8 ตัวอย่างภาพขาวดำที่เลือกค่า Threshold=90

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

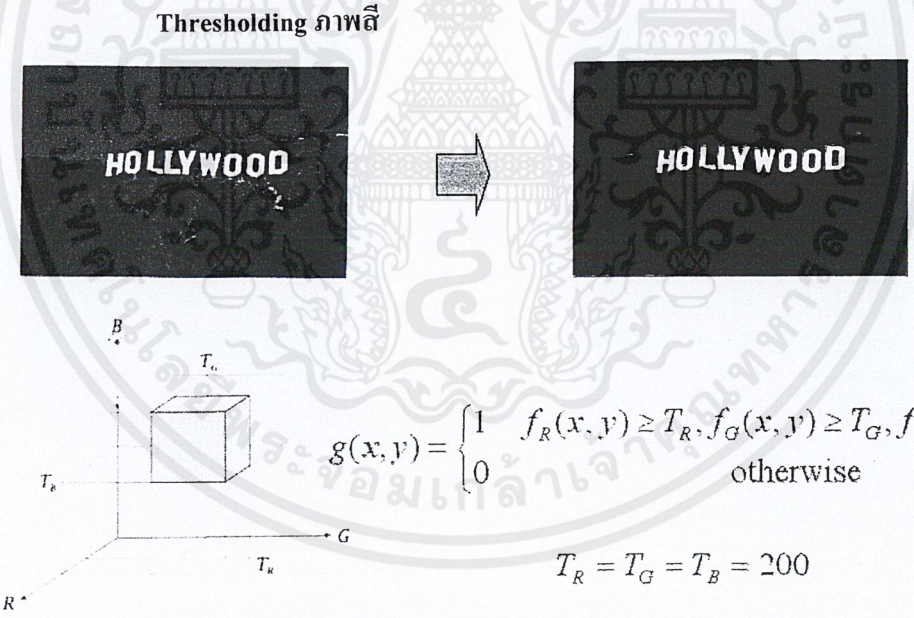


4-9 Flow Chart การทำThresholding Gray Scale

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

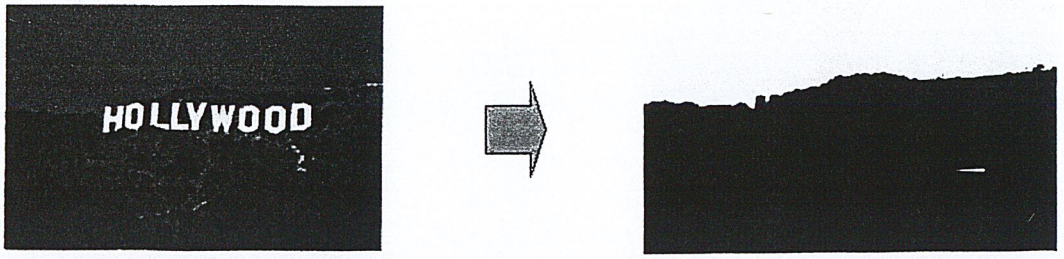


รูปที่4-10 การแบ่งภาพขาวดำโดยใช้ค่า Threshold ต่างๆ



รูปที่4-11 การแบ่งภาพสีโดยใช้ค่า Threshold ของสีแดง น้ำเงิน เขียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



$$g(x,y)=\begin{cases}1 & \begin{aligned}&T_{R1}\le f_R(x,y)\le T_{R2}, && [T_{R1},T_{R2}]=[50,100]\\ &T_{G1}\le f_G(x,y)\le T_{G2}, && [T_{G1},T_{G2}]=[100,150]\\ &T_{B2}\le f_B(x,y)\le T_{B2} && [T_{B1},T_{B2}]=[150,200]\end{aligned}\\ 0 & \text{otherwise}\end{cases}$$

รูปที่4-12 ฟังก์ชั่น ตรวจค่าในช่วงที่สนใจของภาพสี

$$g(x,y)=\begin{cases}1 & d(x,y)\le d_{MAX}\\ 0 & \text{otherwise}\end{cases}$$

$$d(x,y)=\sqrt{(f_R(x,y)-T_R)^2+(f_G(x,y)-T_G)^2+(f_B(x,y)-T_B)^2}$$

$$(T_R,T_G,T_B)=(80,100,50) \qquad d_{MAX}=50$$

รูปที่4-13 การจัดกลุ่ม pixel ที่สีเหมือนกันรวมกลุ่มด้วยกัน

ค่า d(x,y) คือค่าที่ควบคุมระยะห่างของช่วงสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.2 Region Growing

ขั้นตอน

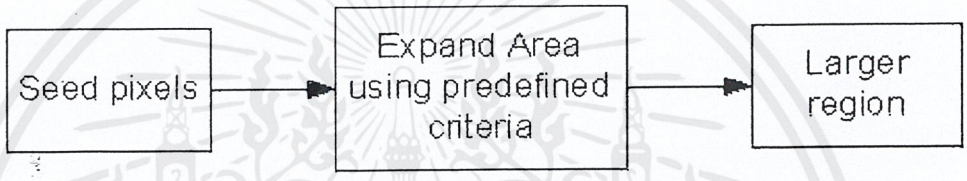
1.) ตั้งค่าเริ่มต้น Seed pixel

2.) กำหนดกฎ (Criteria)

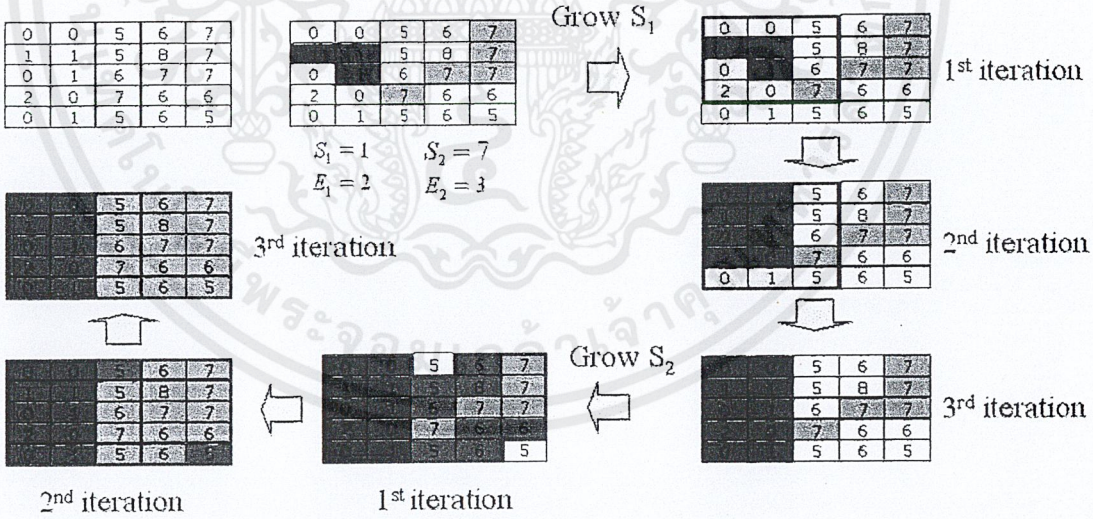
3.) รวมส่วนที่สีเหมือนกัน (Growing)

4.) กำหนดกฎในการหยุด เช่น

- กำหนดเวลา
- กำหนดรอบในการ ทำซ้ำ
- ใช้คุณสมบัติพิเศษอื่นๆในการตัดสินใจ

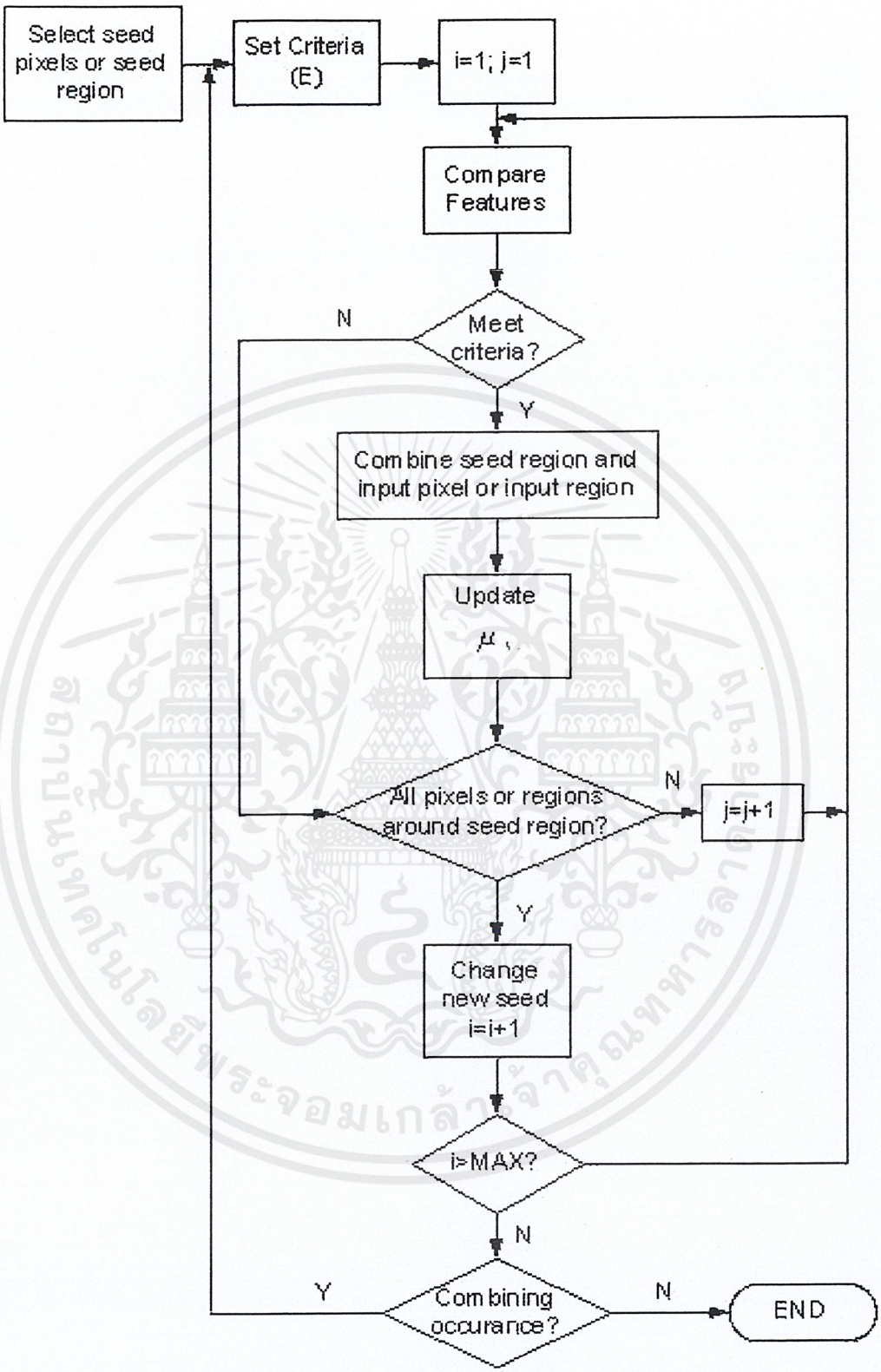


รูปที่4-14 กระบวนการ Region Growing



รูปที่4-15 ขั้นตอนการทำ Region Growing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4-16 Flow Chart ของ Region Growing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.3 Region Clustering

เป็นการจับกลุ่มของ pixel ที่มีค่าคุณสมบัติพิเศษใกล้เคียงกันเข้าด้วยกัน

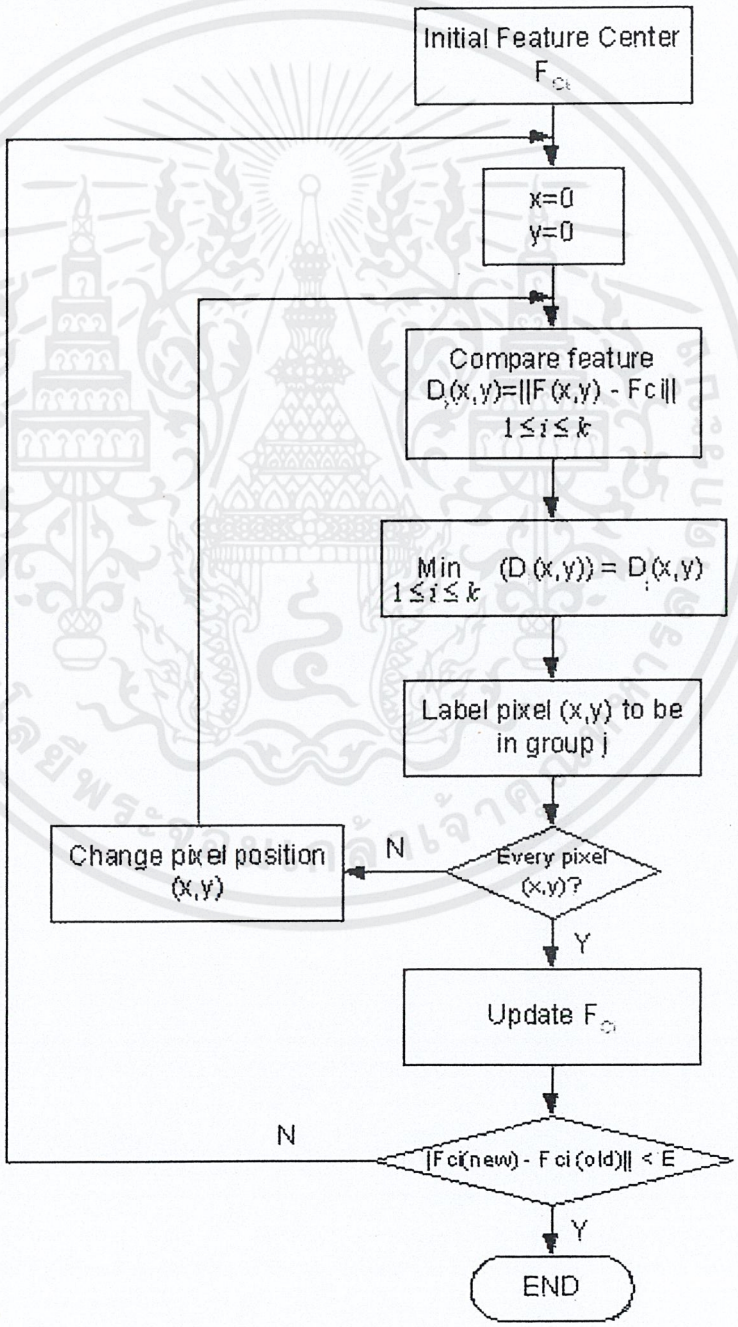
K= จำนวนกลุ่ม

μ_k = ค่า mean ของแต่ละกลุ่ม

เอาจุดสีเทียบกับค่า U ของแต่ละกลุ่ม หา $d_1,d_2,d_3,...,d_k$

ค่า d (ค่าระยะทาง) d มีค่าน้อยที่สุด จุดนั้นจะเข้าไปไหนกลุ่ม

ปรับค่า μ ไปจนกว่า delta μ จะมีค่าน้อยมากๆ

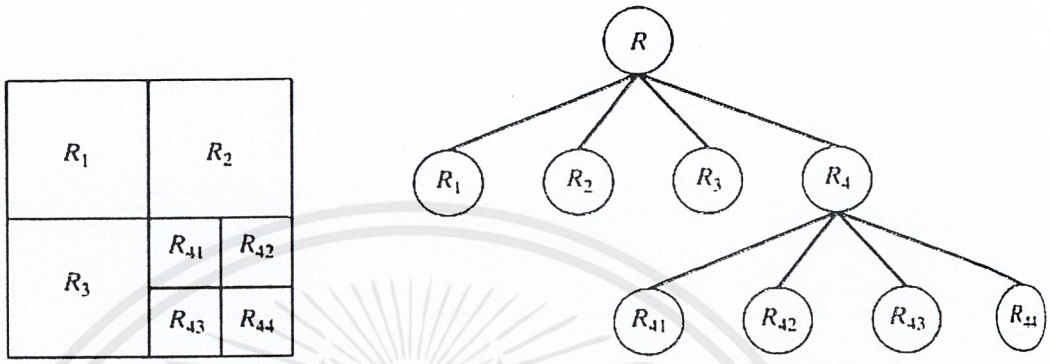


รูปที่4-17 Flow Chart ของ Region Clustering

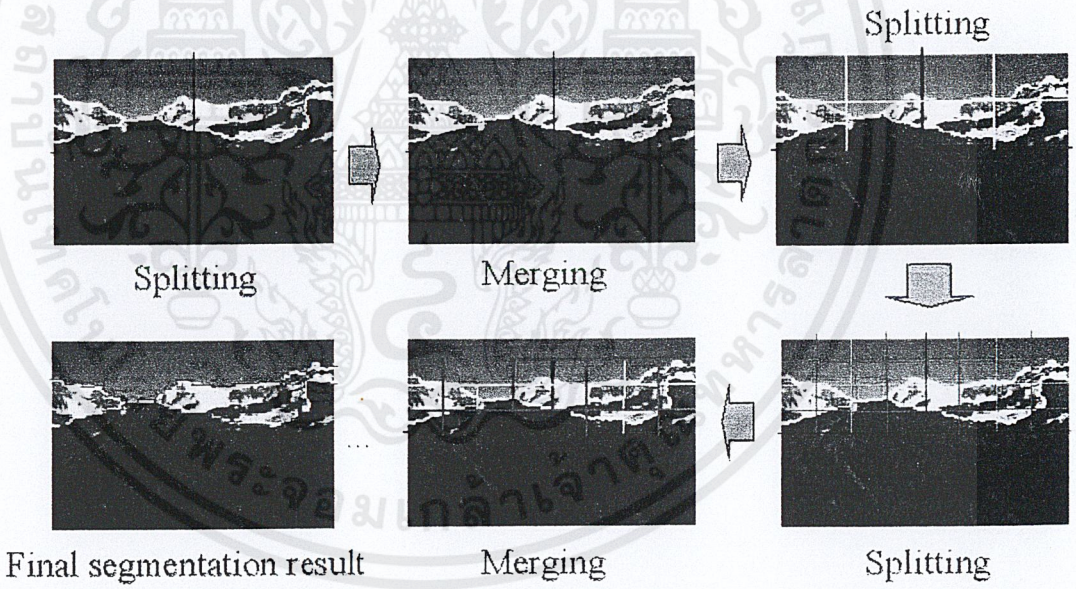
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.4 Region Splitting and Merging

เป็นการแบ่งภาพออกเป็นส่วนๆ โดยส่วนที่ต่างกันจะถูกแบ่งออกไป ส่วนที่เหมือนกันจะถูกรวมเข้าด้วยกัน



รูปที่4-18 ภาพโครงสร้างของ Region Splitting and Merging



รูปที่4-19 ขั้นตอนการทำ Region Splitting and Merging

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

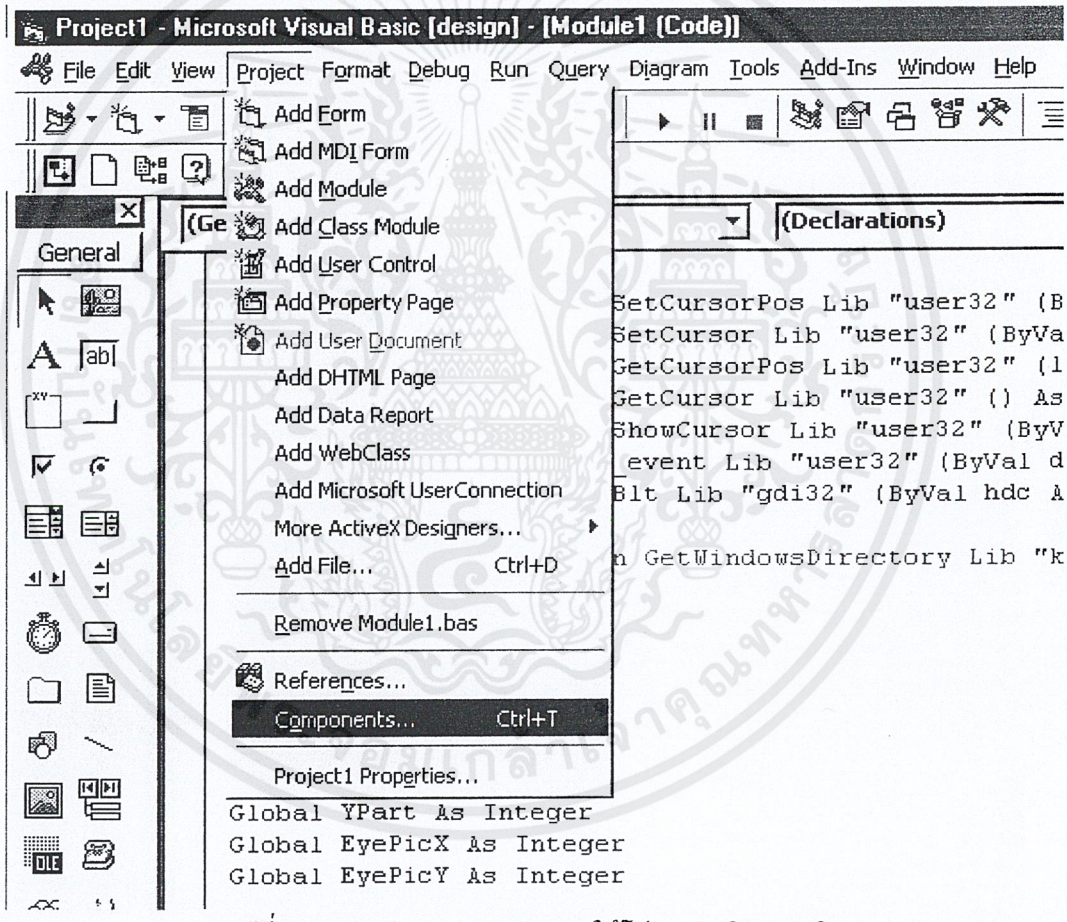
การเขียนโปรแกรมติดต่อกับกล้อง Web Camera

5.1 การควบคุมและติดต่อกล้อง Web Camera ผ่านทางพอร์ต USB ด้วย Component ของ ezVidCap

สามารถทำได้โดยใช้ VB Control Add Component ของ ezVidCap ซึ่งเป็น Component ที่ใช้ติดผ่านกล้อง Web Camera โดยมีขั้นตอนการทำดังต่อไปนี้

1. **ขั้นที่ตอนแรก การกำหนด Custom Control เข้าไปที่ เมนู**

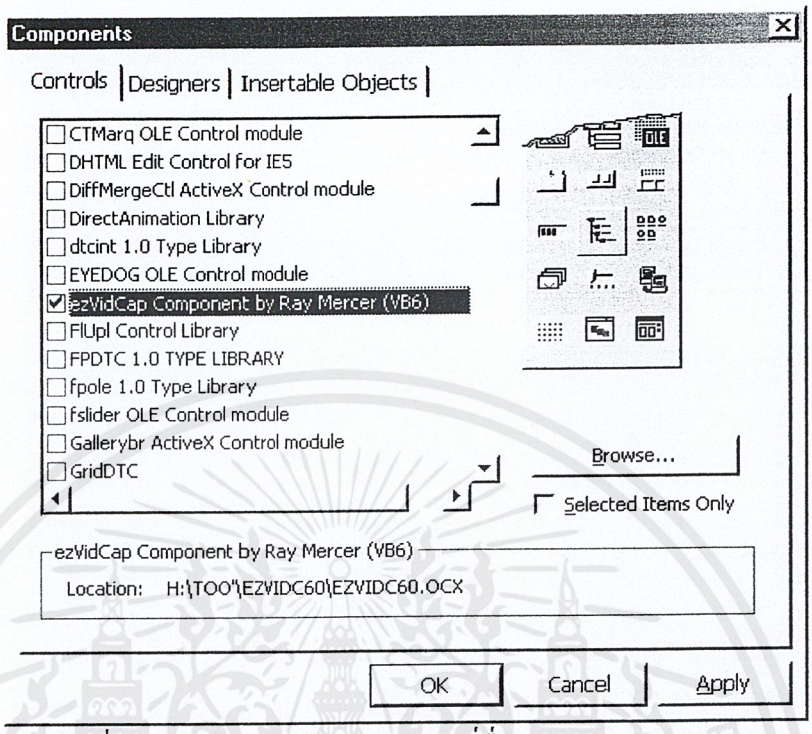
เลือกที่เมนูบาร์ด้านบนของโปรแกรมวิชวลเบสิก โดยเลือก Project--->Companents ดังรูปที่ 5-1



รูปที่ 5-1 แสดง ADD Components ให้โปรแกรมวิชวลเบสิก

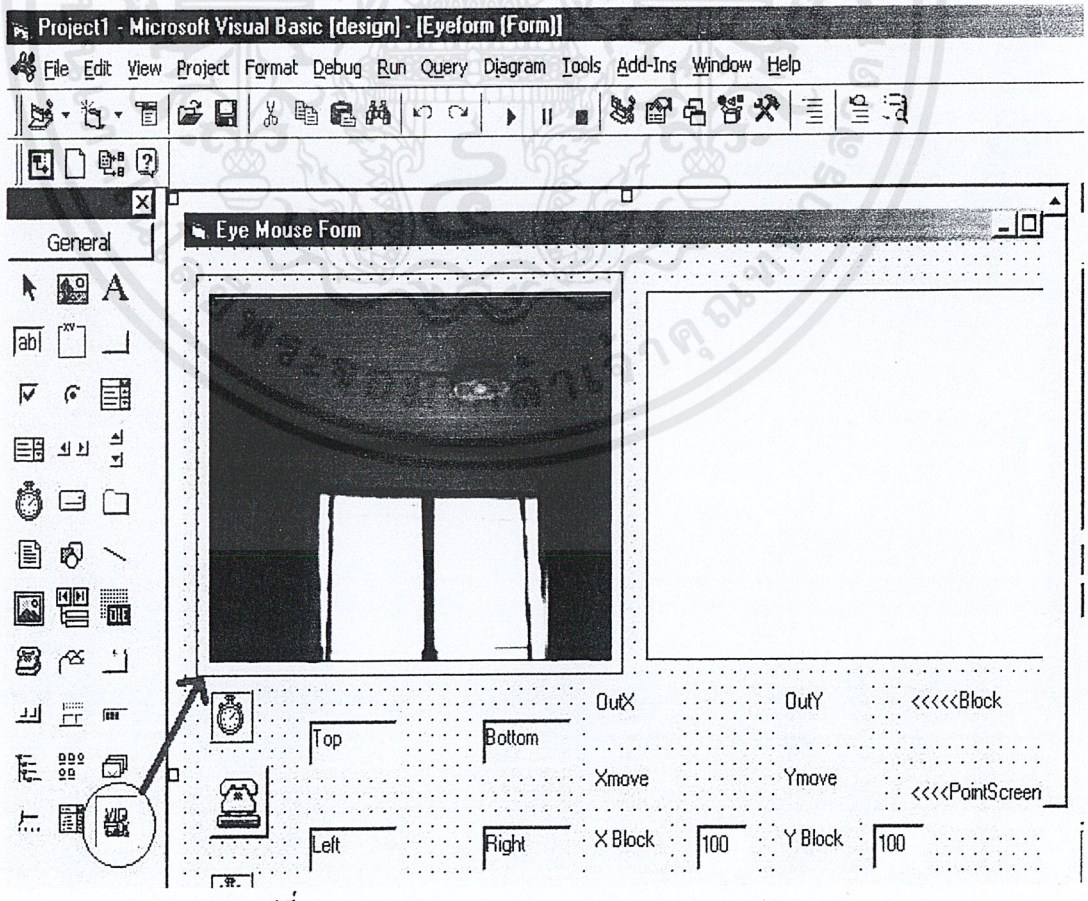
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ขั้นที่สอง เลือกชื่อ Control ชื่อ ezVidCap Component ดังรูปที่5-2



รูปที่ 5-2 แสดงการ ADD Component ที่ชื่อ ezVidCap Component

3. ลาก Control ชื่อ ezVidCap มาไว้บนForm ดังรูปที่ 5-3



รูปที่ 5-3 แสดงการ ADD Control Component ลง Form

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การดึงภาพจาก Component ของ ezVidCap เข้ามาเก็บไว้ในหน่วยความจำ

ขั้นแรกเราจะต้องตั้งชื่อ Component ของ ezVidCap ในที่นี้ตั้งชื่อว่า "ezVidCap1" หลังจากนั้นต้องสร้างเนื้อที่ในหน่วยความจำ เพื่อเก็บภาพที่ดึงขึ้นมา โดยแสดงตาม Code ข้างล่างนี้

```
Dim hDC, hBMP As Long
hDC = CreateCompatibleDC(PictureB.hdc)
hBMP = CreateCompatibleBitmap(PictureB.hdc, 256, 192)
SelectObject hDC, hBMP
Dim MemDC, MemBMP As Long
MemDC = CreateCompatibleDC(PictureB.hdc)
MemBMP = CreateCompatibleBitmap(PictureB.hdc, 256, 192)
SelectObject MemDC, MemBMP
```

หลังจากสร้างพื้นที่ในหน่วยความจำเรียบร้อยแล้วจึงใช้ฟังก์ชั่น GetDC เพื่อให้ HDC เป็นตัวชี้ (Pointer) ไปยังบริเวณภาพของ ezVidCap1

```
hDC = GetDC(ezVidCap1.hwnd)
```

จากนั้นจึงใช้ฟังก์ชั่น Bitblt ในการส่งค่ามาเก็บในหน่วยความจำที่จองไว้ชื่อ "MemDC"

```
BitBlt MemDC, 0, 0, Picture1.ScaleWidth, Picture1.ScaleHeight, hDC, 0, 0, vbSrcCopy
'No Zoom
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

แนวคิดและการออกแบบ

6.1 Functional Requirements

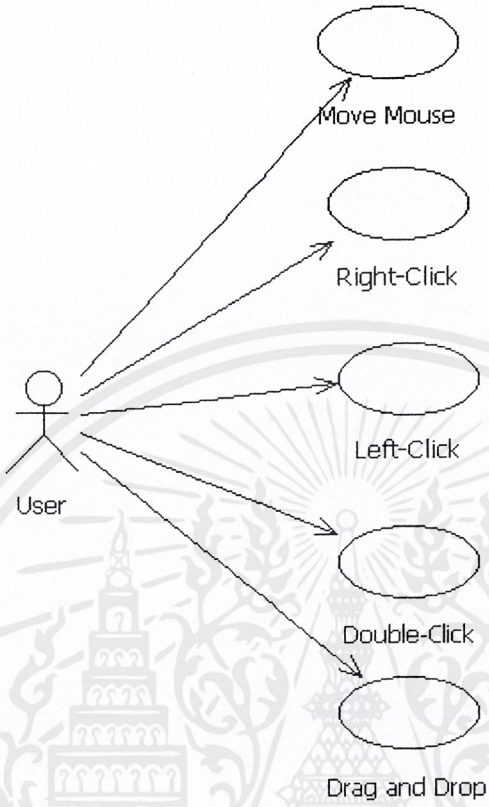
- การเคลื่อนที่ของเคอร์เซอร์เมาส์ (Cursor Mouse)
- คลิกซ้าย
- คลิกขวา
- ดับเบิลคลิก
- ลากและปล่อย

6.2 Non-functional Requirements

- ความเร็ว
- ความแม่นยำ
- ใช้ง่าย
- รองรับผู้ใช้ให้ได้มากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

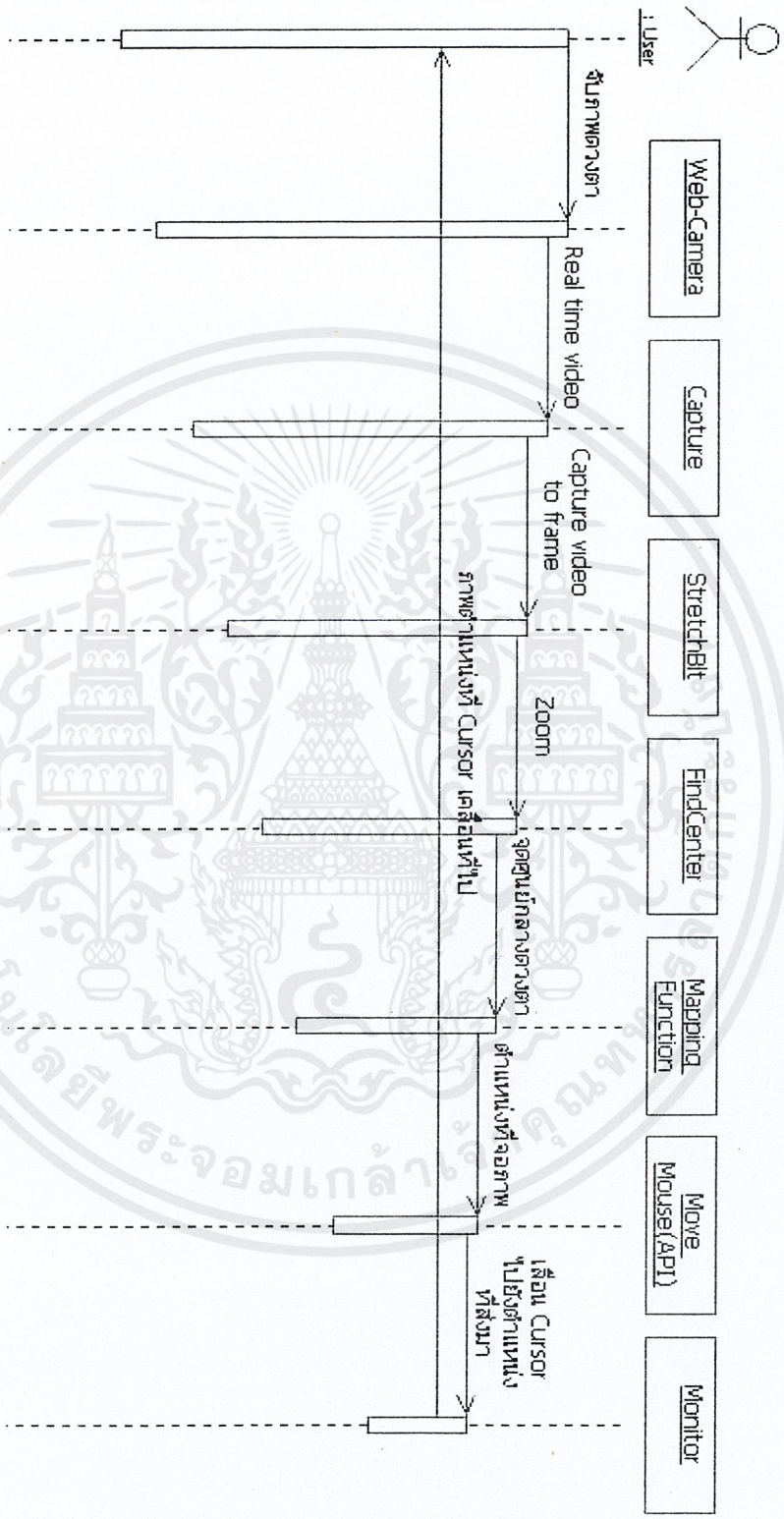
6.3 Use Case



รูปที่6-1 Use Case Diagram

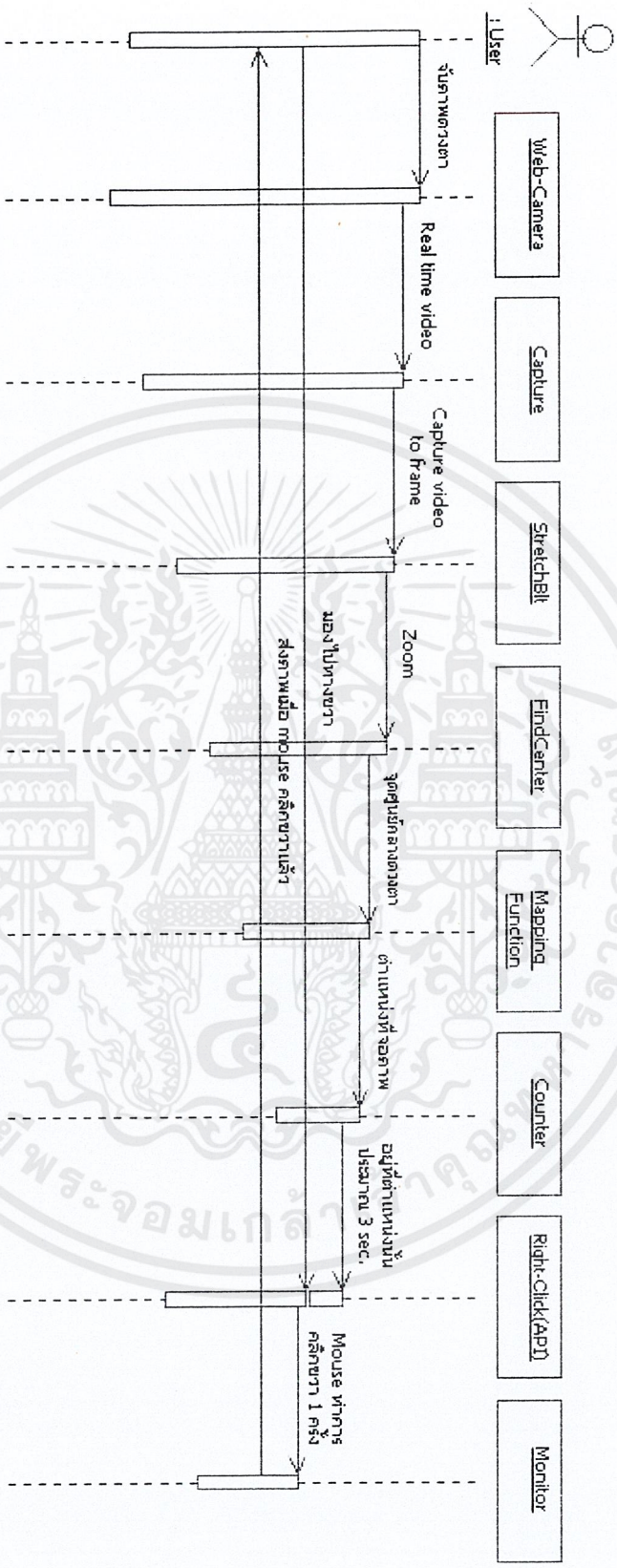
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 Sequence Diagram



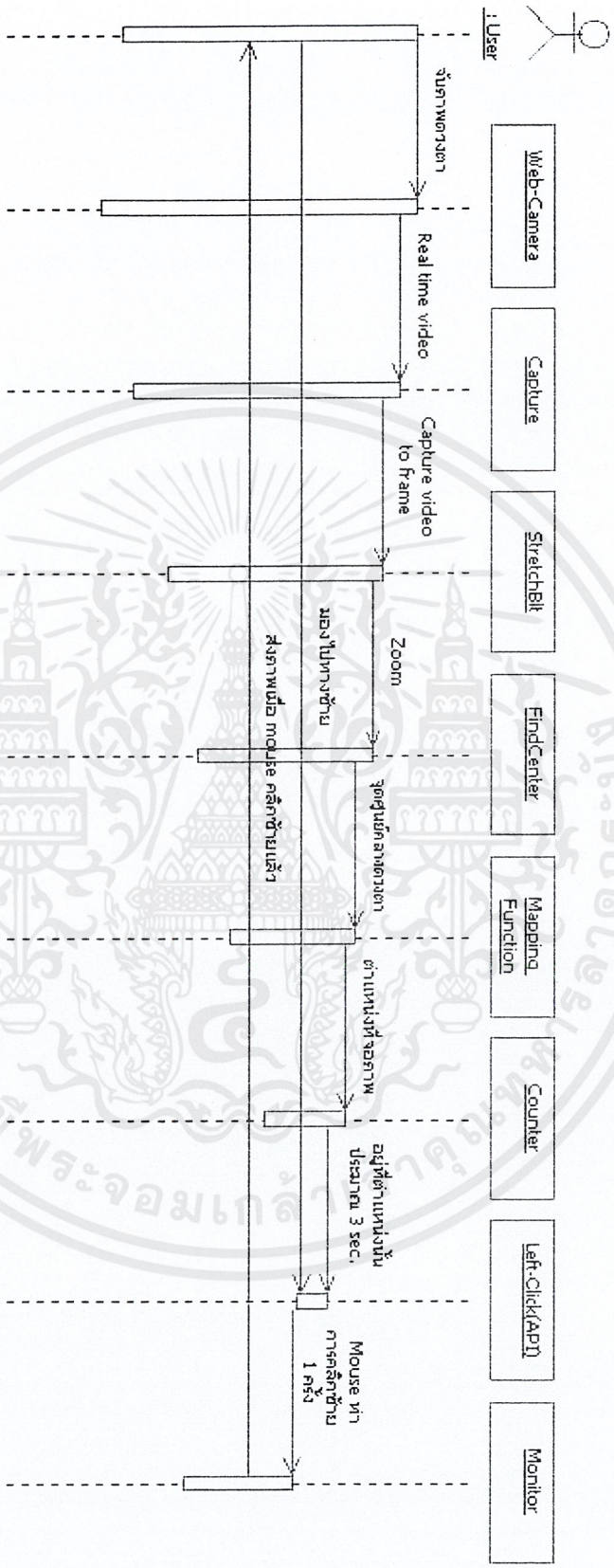
รูปที่6-2 Sequence Diagram ของการเคลื่อนที่ของเมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



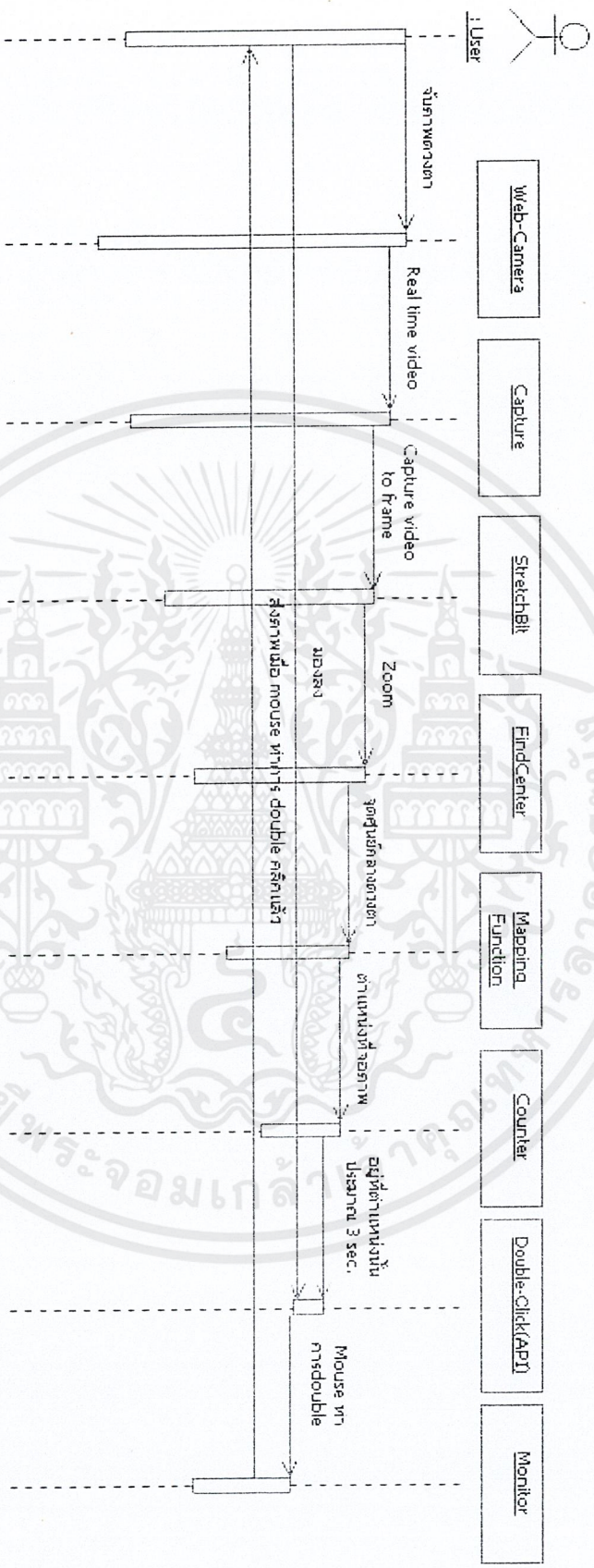
รูปที่6-3 Sequence Diagram ของการคลิกขวาของเมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



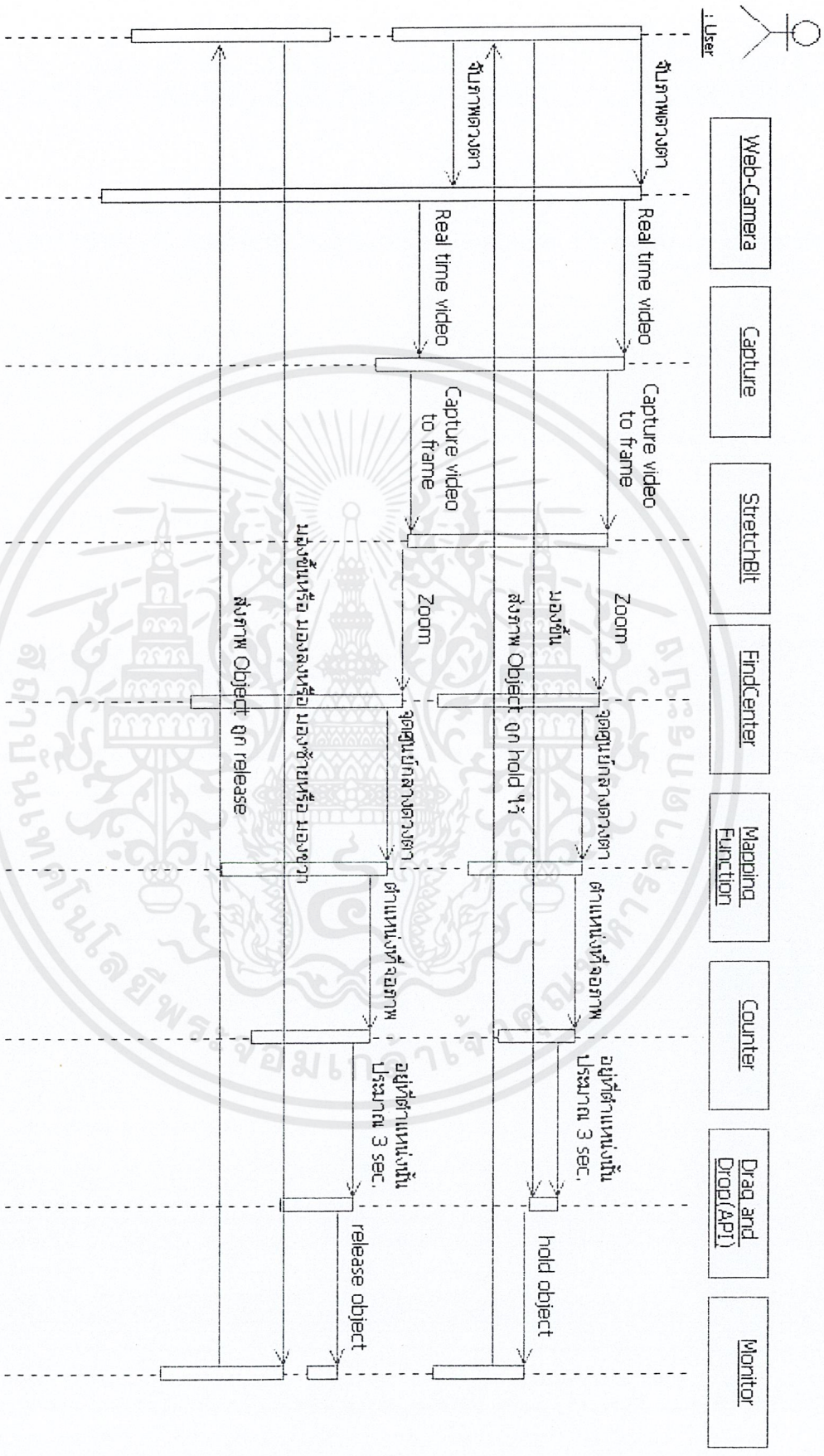
รูปที่6-4 Sequence Diagram ของการคลิกซ้ายของเมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่6-5 Sequence Diagram ของการดับเบิลคลิกของเมาส์

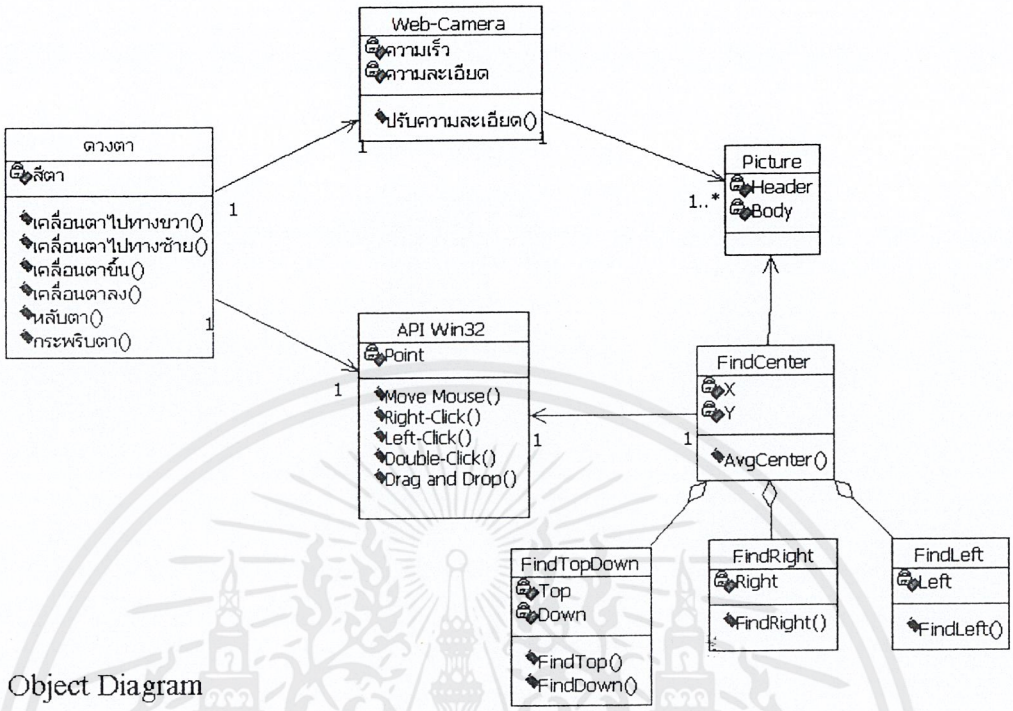
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่6-6 Sequence Diagram ของการลากและปล่อยของเมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

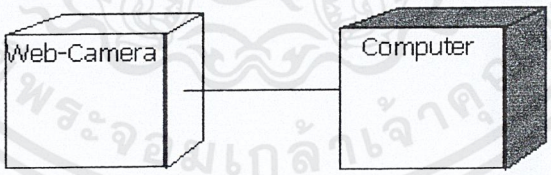
6.5 Object Diagram



Object Diagram

รูปที่6-7 Object Diagram ของเมาส์

6.6 Deployment Diagram



Deployment Diagram

รูปที่6-8 Deployment Diagram

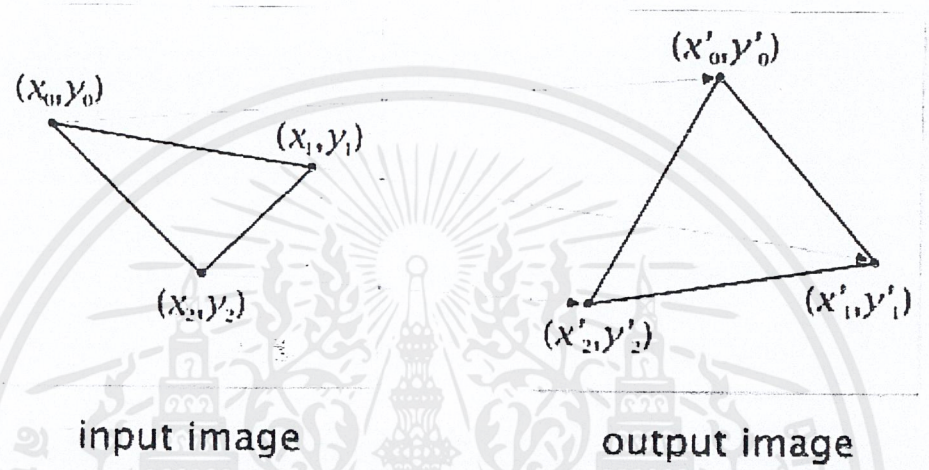
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

การขยายภาพ

7.1 Geometric Transformation

คือการเปลี่ยนแปลงภาพ เช่น หมุนภาพ, เลื่อนภาพ, ขยายหรือย่อภาพ, บิดภาพ เป็นต้น



รูปที่7-1 Geometric Transformation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_0 & a_1 \\ b_0 & b_1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_2 \\ b_2 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

รูปที่7-2 ฟังก์ชั่น Geometric Transformation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

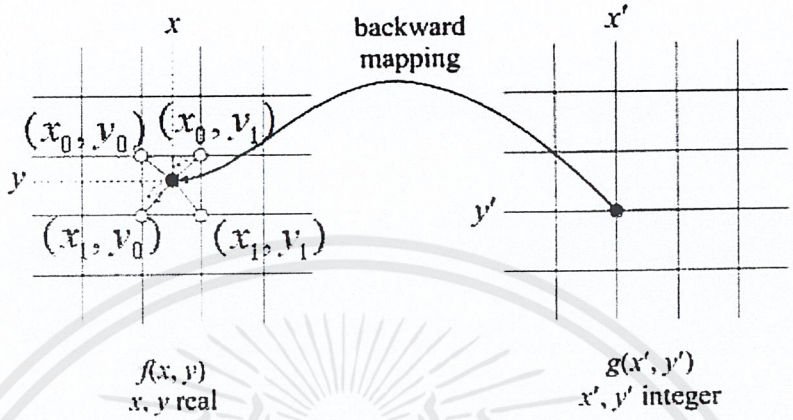
1) Rotation	$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$
2) Translation (Shift)	$x' = x - \Delta x$ $y' = y - \Delta y$
3) Enlarge or reduce by scaling factor (S)	$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \frac{1}{S} & 0 \\ 0 & \frac{1}{S} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$
4) Shear (บิด)	$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_0 & a_1 \\ b_0 & b_1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \end{bmatrix}$

ตารางที่ 7-1 Geometric Transform Parameters (Backward Mapping)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 Backward mapping

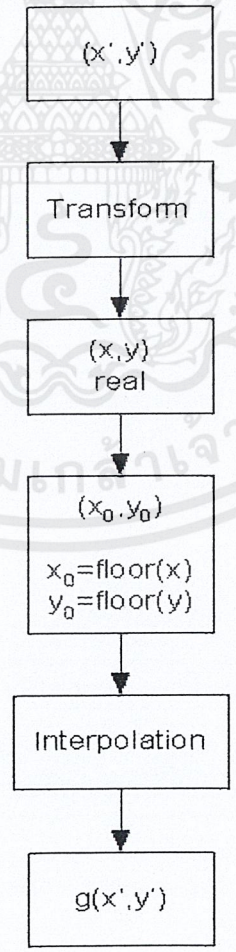
เป็นการมองจากผลลัพธ์ไปหาค่าเริ่มต้นจะทำให้ไม่เกิดปัญหา กรณี 2 จุดเมื่อผ่านฟังก์ชั่นแล้วได้จุดเดียวกัน หรือได้ค่าไม่ลงตัว



รูปที่7-3 Backward mapping

ฟังก์ชั่น Transformation

$g(x',y') = interpolation[f(x_0,y_0)]$



รูปที่7-4 Flow Chart Backward Mapping

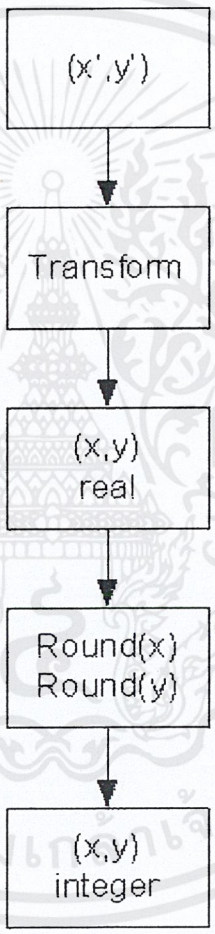
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3 การขยายภาพ โดยใช้ Zero-order Interpolation

Zero-order Interpolation เป็นกระบวนการประมวลผลทางรูปภาพ เรื่อง Geometric Transformation ใช้ในการเพิ่มหรือลดขนาดของภาพ ขึ้นกับค่า Scaling factor (S) > 1

Zero-order Interpolation เป็นวิธีที่เร็วกว่าการใช้ First-order Interpolation และ High-order Interpolation เพราะการคำนวณ มีความซับซ้อนน้อยกว่ามากทำให้ใช้เวลาน้อยในการคำนวณฟังก์ชั่น Transformation

g(x',y') = f[round(x),round(y)]



รูปที่ 7-5 Flow Chart Zero-order Interpolation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

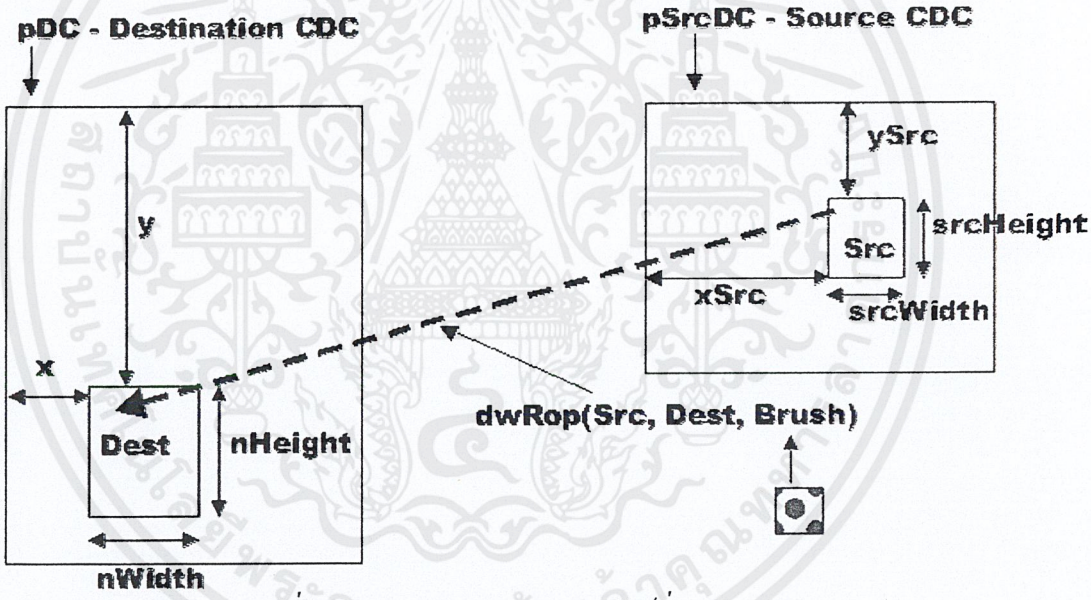
7.4 การขยายภาพ โดยใช้ฟังก์ชั่น StretchBlt

เราสามรถขยายขนาดของภาพโดยใช้ฟังก์ชั่น StretchBlt แทนซึ่งมีความรวดเร็วการที่เราเขียนฟังก์ชั่นการขยายภาพแบบทฤษฎี First Order เพราะเป็นการทำงานระหว่างหน่วยความจำ โดยมีขั้นตอนในการใช้งานดังต่อไปนี้

1. จะต้องประกาศส่วน Declaration ในตอนต้นของโปรแกรมก่อนใช้งาน

Declare Function StretchBlt Lib "gdi32" (ByVal hdc 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 nSrcWidth As Long, ByVal nSrcHeight As Long, ByVal dwRop As Long) As Long

nRc = pDC->StretchBlt(x, y, nWidth, nHeight, pSrcDC, xSrc, ySrc, srcWidth, srcHeight, dwRop);



รูปที่ 7-6 แสดงการขยายภาพของฟังก์ชั่น StretchBlt

2. จะต้องระบุตำแหน่งเริ่มต้นของภาพต้นทางและปลายทาง

ในการทำงานของฟังก์ชั่น StretchBlt จะต้องระบุตำแหน่งภาพของภาพต้นทางว่าจะเริ่มที่ไหน และตำแหน่งภาพของภาพปลายทางว่าเริ่มที่ไหน เพื่อที่จะเป็นจุดเริ่มต้นของการเริ่มคัดลอกภาพ(copy) จากภาพต้นทางไปปลายทาง

3. ระบุขนาดของภาพต้นทางและปลายทาง

โดยที่เราจะต้องกำหนดขนาดของภาพต้นทางแลปลายทางที่ต้องการ โดยเราสามารถระบุได้ว่าภาพต้นทางมีขนาดเท่านี้ ภาพปลายทางจะให้มีขนาดเท่าไหร่ได้ โดยจะแสดงในรูปที่ 7-6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของการใช้ฟังก์ชั่น StretchBlt

```
Dim hDC, hBMP As Long
hDC = CreateCompatibleDC(PictureB.hdc)
hBMP = CreateCompatibleBitmap(PictureB.hdc, 256, 192)
SelectObject hDC, hBMP

Dim MemDC, MemBMP As Long
MemDC = CreateCompatibleDC(PictureB.hdc)
MemBMP = CreateCompatibleBitmap(PictureB.hdc, 256, 192)
SelectObject MemDC, MemBMP

hDC = GetDC(ezVidCap1.hwnd)
StretchBlt MemDC, 0, 0, 256, 192, hDC, 0, 0, 64, 48, vbSrcCopy
```

จากตัวอย่างคือการขยายภาพเป็น 4 เท่าโดยที่จุดเริ่มต้นของภาพต้นทาง คือ (0,0) โดยขนาดของภาพต้นทางคือ กว้าง 64 x สูง 48

ในขณะที่ภาพปลายทางจะเริ่มส่งรูปมาที่ตำแหน่ง (0,0) โดยที่ขนาดของภาพปลายทางมีขนาด คือ กว้าง 256 x สูง 192 ซึ่งมีขนาดเป็น 4 เท่าของภาพต้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

การหาจุดศูนย์การของดวงตา

ขั้นตอนการหาจุดศูนย์กลางเป็นสิ่งที่สำคัญที่สุดในการทำโปรแกรมควบคุมเมาส์ด้วยดวงตาเนื่องจากหากเราไม่สามารถหาตำแหน่งของดวงตาที่ถูกต้องแม่นยำแล้วเราก็จะไม่สามารถหาตำแหน่งที่ถูกต้องเพื่อที่จะควบคุมเมาส์ไปยังตำแหน่งที่ต้องการได้

โดยขั้นตอนการหาจุดศูนย์กลางของดวงตานั้นเราจะเริ่มทดลองจากภาพนิ่งนามสกุล Bitmap ก่อนที่จะนำไปใช้กับภาพเคลื่อนไหวจริง โดยเราสามารถแบ่งขั้นตอนได้ดังนี้

8.1 ขั้นตอนการหาขอบบนขอบล่างของดวงตา

การที่จะหาจุดศูนย์กลางของดวงตานั้น ขั้นแรกจะต้องหาขอบบนขอบล่างของดวงตาก่อน สาเหตุที่ต้องหาขอบบนขอบล่างของดวงตาก่อนนั้นก็ เพราะ หากเราหาบริเวณด้านซ้ายและขวาก่อนนั้นจะมีสิ่งรบกวนมากกว่า ตัวอย่างเช่น ผม เงาของหนังตา นั้นเป็นสิ่งสำคัญอย่างมากที่จะให้การหาจุดศูนย์กลางของตานั้นยากขึ้น (ขั้นต้นเรายังไม่มีการกำจัดสิ่งรบกวน เช่น ผม โดยการกำจัดสิ่งรบกวนนั้นจะกล่าวในบทถัดไป)

รูปที่ 8-1 แสดงภาพดวงตาที่สมบูรณ์ไม่มีเงาของแสงสว่างมารบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนตอนการหาขอบบนขอบล่างของดวงตานั้น เรานำภาพมาพิจารณาเฉพาะช่วงที่เราสนใจ(Threshold) โดยช่วงที่เราสนใจนั้นขั้นพื้นฐานคือบริเวณที่เป็นสีดำ โดยมีค่า RGB คือ

1. ค่า R อยู่ในช่วง 0-30
2. ค่า G อยู่ในช่วง 0-30
3. ค่า B อยู่ในช่วง 0-30

โดยวิธีการนั้นเรียกว่า Similarity Segmentation หรือการทำเซ็กเมนต์เตชั่น จะตัดภาพเหลือแต่ช่วงสีที่เราสนใจนั้นดังจะเห็นได้จากรูปที่ 8-2

รูปที่ 8-2 ภาพดวงตาที่ผ่านการทำเซ็กเมนต์เตชั่น และการค้นหาขอบบนขอบล่างของดวงตา

ขั้นตอนการหาขอบบนขอบล่างจะแบ่งออกเป็น 2 ขั้นตอนย่อยคือ

8.1.1 หาขอบล่างของดวงตา

การที่เราจะเริ่มจากของล่างของดวงตานั้นก่อนเนื่องจากหากเราหาขอบบนของดวงตาก่อนเราจะเจอคิ้วซึ่งเป็นสิ่งรบกวนที่เราไม่ต้องการ

โดยเราจะทำการค้นหาตามแนวแกน X จากด้านล่างของภาพ จากล่างขึ้นบนเพื่อหาจุดแรกที่เป็นสีดำ โดยเราจะถือจุดแรกที่พบจะเป็นขอบล่างของดวงตา (สมมุติว่าด้านล่างของตาไม่มีสิ่งรบกวน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.1.2 หาขอบบนของดวงตา

หลังจากเราสามารถค้นหาขอบล่างของดวงตาได้แล้วนั้น เราจะทำการหาขอบบนของดวงตาโดยการค้นหาต่อจากขอบล่างของตาที่พบไปตามแนวแกน X จากล่างขึ้นบน โดยค้นหาแนวแกน X ที่ปราศจากสีดำทั้งแนวเราจะถือว่าแนวนั้นจะเป็นขอบบนของตาโดยเราอาจจะลดตำแหน่งของจากจุดที่สันนิษฐานว่าเป็นขอบบนลงมาได้อีก เนื่องจากสีดำที่เราพบเป็นจุดสุดท้ายนั้นอาจเกิดจากเงาของหนังตาก็ได้ ตามเส้นที่โยงจากขอบบนลงมาในรูปที่ 8-2

โดยขั้นตอนการหาขอบบนนั้นเราสามรถลดขั้นตอนการค้นหาได้คือ เราจะลดแนวการค้นหาจากล่างขึ้นบนโดยจะสันนิฐานว่าตาดำนั้นจะต้องไม่มีขนาดเล็กกว่าที่สันนิษฐานไว้ ซึ่งจะทำให้เราลดการค้นหาได้ "โดยจะลดการค้นหาตามแนวแกน Y โดยข้ามไป 60 Pixel (โดยจำนวนจุดที่ข้ามไปนั้นต้องสำพันธ์กับความสูงของภาพที่นำมาพิจารณาด้วย)" ตามเส้นที่โยงขึ้นไปจากขอบล่างในรูปที่ 8-2 โดยจะไม่มีการค้นหาส่วนที่ข้ามนั้นเพื่อลดความเร็ว

8.2 ขั้นตอนการหาด้านซ้ายและด้านขวาของดวงตา

ในการหาด้านซ้ายและด้านขวาของดวงตานั้นเราจะใช้ Sliding Window โดยขนาดของ Sliding Window นั้นมีขนาดเป็นเป็นรูปทรงสี่เหลี่ยมจัตุรัส โดยที่เราสันนิฐานว่าตาดำของเรานั้นมีขนาดสมมาตรเท่ากันทั้งความกว้างและความสูง

โดยเมื่อเราสามารถหาขอบบนและขอบล่างของดวงตาได้แล้วนั้นทำให้เรารู้ถึงขนาดของดวงตาที่สันนิษฐานไว้และช่วงขอบบนและขอบล่างที่ดวงตาอยู่ โดยเราจะใช้ Sliding Window ในการค้นหาตาดำที่อยู่ในช่วงนั้น โดยการค้นหานั้นเราจะปรับปรุงวิธีการค้นหาเพื่อลดจำนวนการค้นหาให้น้อยลงที่สุดเท่าที่จะทำได้โดยมีรูปแบบการค้นหาตามการทดลองต่อไปนี้

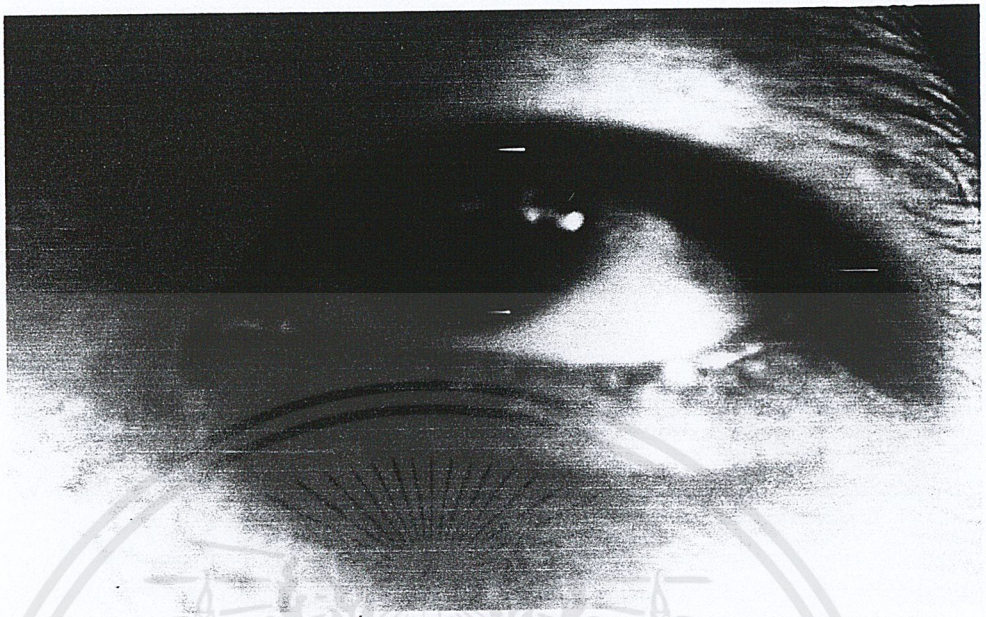
วิธีแรกเริ่ม

เราจะเริ่มใช้ Sliding Window ตั้งแต่จุกแรกของแนวที่เราได้จากการหาขอบบนขอบล่าง โดย Sliding Windowเริ่มค้นหาไปตามแนวแกนขอบบนขอบล่าง จากซ้ายไปขวา โดยภายSliding Windowเราจะค้นหาจุดทุกจุดที่เป็นสีดำที่อยู่ภายใน Window เอาไว้ โดยรวมค่านั้นเอาไว้เพื่อนำมาเปรียบแต่ละ Window ที่ค้นหาตามแนวแกน X ว่า Window ไหนหากมีสีดำมากที่สุดจะถือว่าWindowนั้นเป็นตาดำโดยเราจะได้ค่าด้านซ้ายและด้านขวาของตามาคำนวณกับขอบบนขอบล่างของตาเพื่อหาจุดศูนย์กลางของดวงตา

จะเห็นได้ว่าเราต้องทำการค้นหาทุกๆจุดตามแนวขอบบนขอบล่าง ซึ่งจะใช้ทรัพยากรในการประมวลผลของคอมพิวเตอร์จำนวนมากทำให้ช้าเราจึงต้องทำการปรับปรุงวิธีการค้นหา

จากการทดลองหากใช้วิธีนี้ 1 ภาพ ใช้เวลาประมาณ 30-60 วินาที ซึ่งช้ามาก

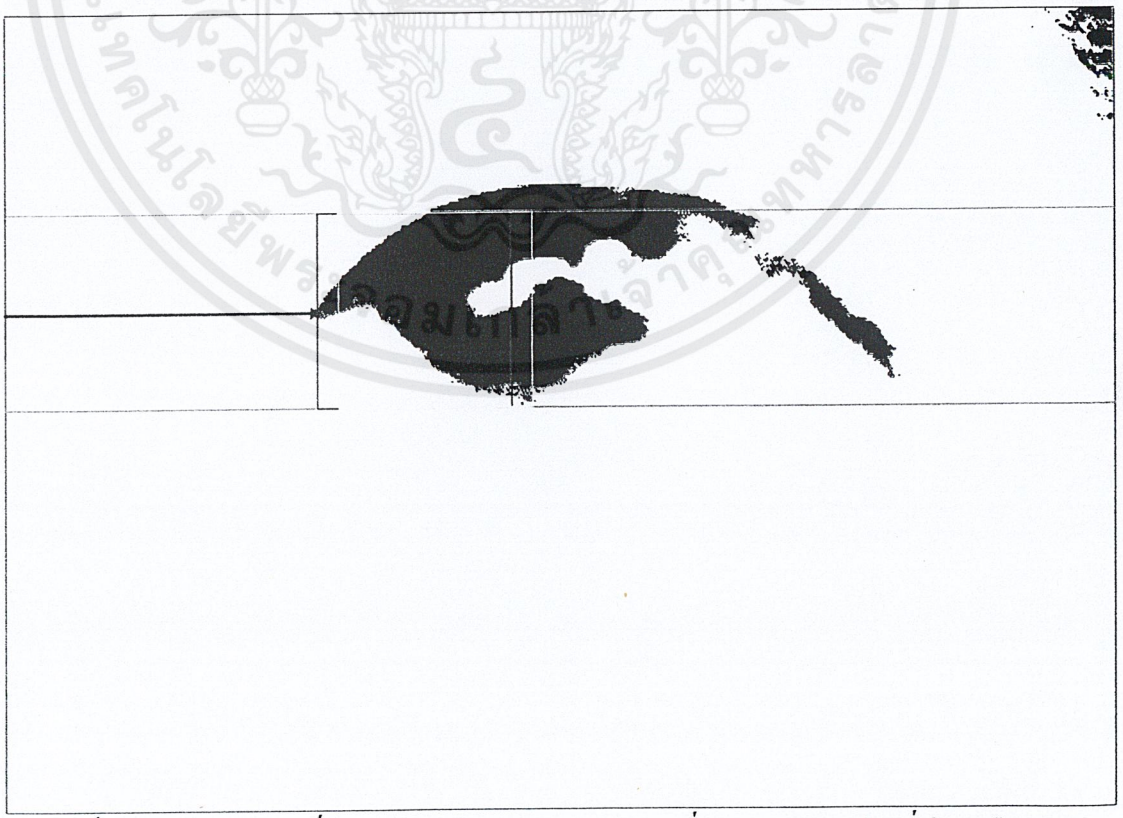
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-3ภาพดวงตาที่มีเงาของแสงสว่างมารบกวนด้านขวาของตาดำ

ปรับปรุงจุดเริ่มค้นหา

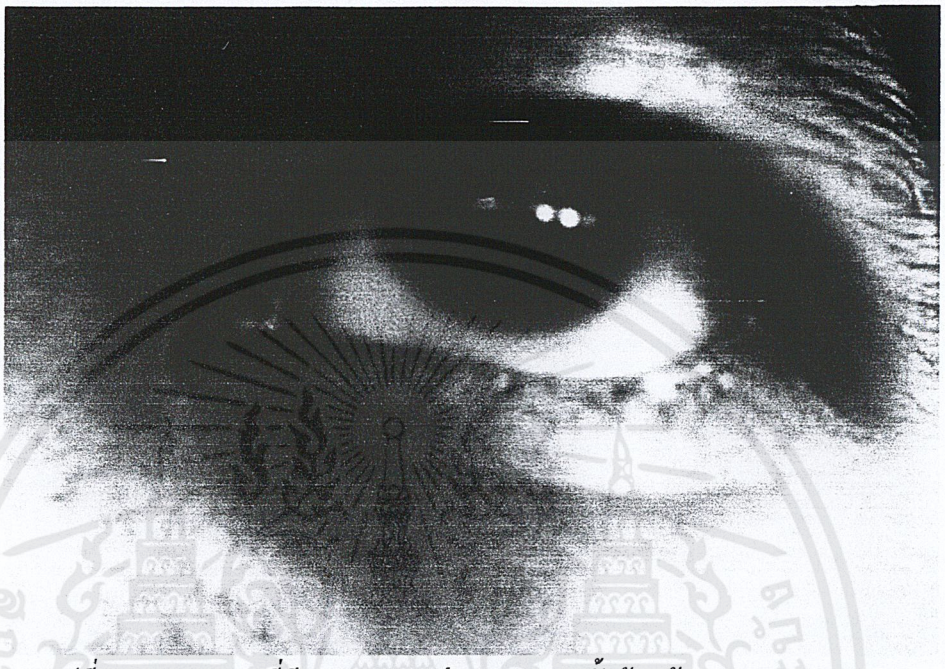
จากภาพที่ 8-4 แสดงให้เห็นว่าเราไม่จำเป็นต้องใช้ Sliding Window ตั้งแต่จุดเริ่มแรกของแนวขอบบนขอบล่าง ดังนั้นเราจะใช้เพียงการค้นหาเพียงแนวใดเพียงแนวหนึ่งบนแกนYเพื่อหาว่าจุดใดควรจะเป็นจุดเริ่มที่จะใช้ Sliding Window



รูปที่ 8-4 ทำปรับปรุงจุดเริ่มค้นหาและใช้ Sliding Windows เพื่อหาจำนวนจุดสีดำที่น่าจะเป็นตาดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองหลังจากปรับปรุงจุดเริ่มต้นของการค้นหา 1 ภาพ ใช้เวลาประมาณ 10-20 วินาที ซึ่งเร็วขึ้นมากทีเดียว



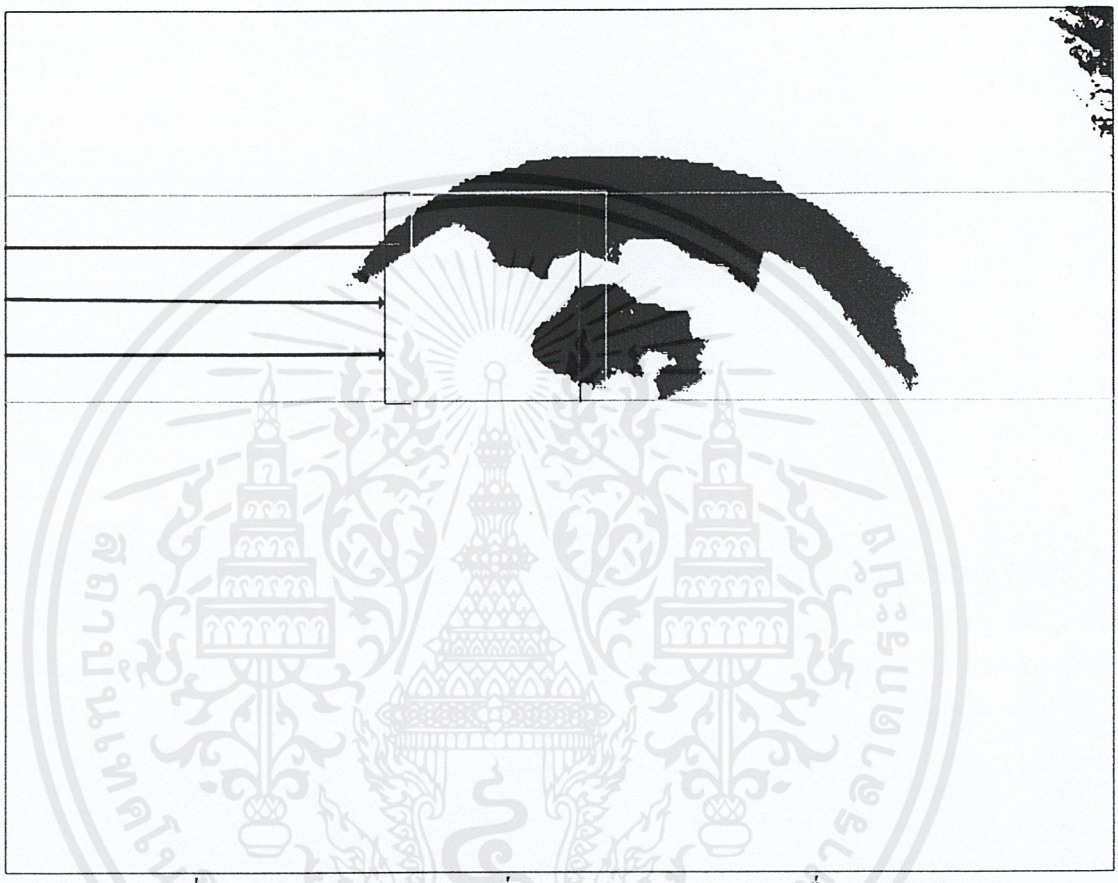
รูปที่ 8-5 ภาพดวงตาที่มีเงาของแสงสว่างมารบกวนทั้งด้านซ้ายและขวาของตาดำ

หลังจากเราปรับปรุงเรื่องความเร็วเราก็ต้องคำนึงถึงความถูกต้องด้วยโดยตามรูปที่ 8-5 จะสังเกตเห็นได้ว่า ภาพนี้มีสิ่งรบกวนมากโดยมาจากแสงสะท้อนต่างๆ ซึ่งหลังจากการทำเซ็กเมนต์เตชั่นเราจะทำให้ได้ภาพดังรูปที่ 8-6

ซึ่งหากเราใช้เพียงแถว(Row) เดียวที่ใช้สำหรับการค้นหาจุดเริ่มต้นการใช้ Sliding Window อาจจะทำให้เกิดข้อผิดพลาดในการจุดศูนย์กลางของดวงตาได้ เราจึงทำการปรับปรุงการค้นหาโดยใช้ 3 แถวในการค้นหาจุดเริ่มต้น ก่อนการใช้ Sliding Window ในการค้นหาตามรูปที่ 8-6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 8-6 แสดงให้เห็นว่าหากที่แสงสะท้านบนตาเรามากทำให้ภาพที่เรารับนั้นผิดไป คือ ตาดำเรานั้นจะไม่เป็นสีดำ ซึ่งจะมีผลอย่างมาในการค้นหาจุดศูนย์กลางของตา



รูปที่ 8-6 ทำการปรับปรุงจุดก่อนเริ่มการใช้ Sliding Window เพื่อความถูกต้อง

นอกจากการปรับปรุงจุดเริ่มต้นค้นหาแล้วนั้นจริงๆแล้ว เราสามาปรับปรุงส่วนของช่วงที่เราสนใจ(Threshold) ให้มีช่วงที่กว้างมากขึ้นก็ได้ แต่การที่เราเพิ่มช่วงมากขึ้นนั่นก็เป็นการเพิ่มสิ่งรบกวนเข้ามามากเพิ่มตามไปด้วย ฉะนั้นจะต้องคำนึงถึงส่วนนี้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามรูปที่ 8-7 แสดงให้เห็นว่าหากเพิ่มช่วงที่เราสนใจมากขึ้นก็จะมีสิ่งรบกวนเพิ่มมากขึ้นตามไปด้วยนั้นก็คือเงาของหนังตานั้นเอง

รูปที่ 8-7 ทำการปรับค่า Threshold เพิ่มมากขึ้น แต่ก็ทำให้มีสิ่งรบกวนเพิ่มมากขึ้นตามไปด้วย

ปรับปรุงจุดสิ้นสุดการค้นหา

จากภาพที่ 8-6 แสดงให้เห็นว่าเราไม่จำเป็นต้องใช้ Sliding Window ค้นหาไปจนสุดของแนวขอบบนขอบล่าง หาก Sliding Window ของเรานั้นค้นหาไปจนจุดสีดำน้อยกว่าที่เราสมมุติฐานว่าจะไม่มีทางเป็นตาดำแล้ว ก็จะหยุดการค้นหา

จากการทดลองหลังจากปรับปรุงจุดสิ้นสุดของการค้นหา 1 ภาพ ใช้เวลาประมาณ 5-10 วินาที ซึ่งอาจจะลดลงมาไม่มาก แต่ก็ช่วยได้มากทีเดียว

ปรับปรุงโดยการหยุดการค้นหาโดยหากเจอ Window ใดมีจำนวนจุดสีดำเกินที่ตั้งไว้

เราอาจสามารถหยุดการค้นหาโดยหากมี Window ใดก็ตามที่มีจำนวนจุดสีดำเกินที่ตั้งไว้ให้สมมุติฐานได้เลยว่าเป็นตาดำ ซึ่งวิธีการนี้อาจไม่ใช่วิธีที่ดีเพราะอาจทำให้ความถูกต้องลดลงได้ แต่หากต้องการลดเวลาก็อาจเป็นทางเลือกหนึ่งที่นำมาใช้ เพราะว่าในแนวขอบบนขอบล่างอาจมีจุดที่มีสีดำสูงกว่าเกณฑ์ที่ตั้งไว้มากว่าหนึ่งจุดดังนั้นจุดที่มีสีดำมากสุดอาจไม่ใช่จุดนั้น

จากการทดลองหลังจากปรับปรุง 1 ภาพ ใช้เวลาประมาณ 3-8 วินาที ซึ่งลดลงมาไม่มาก เมื่อนำไปเทียบกับความถูกต้องที่ลดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

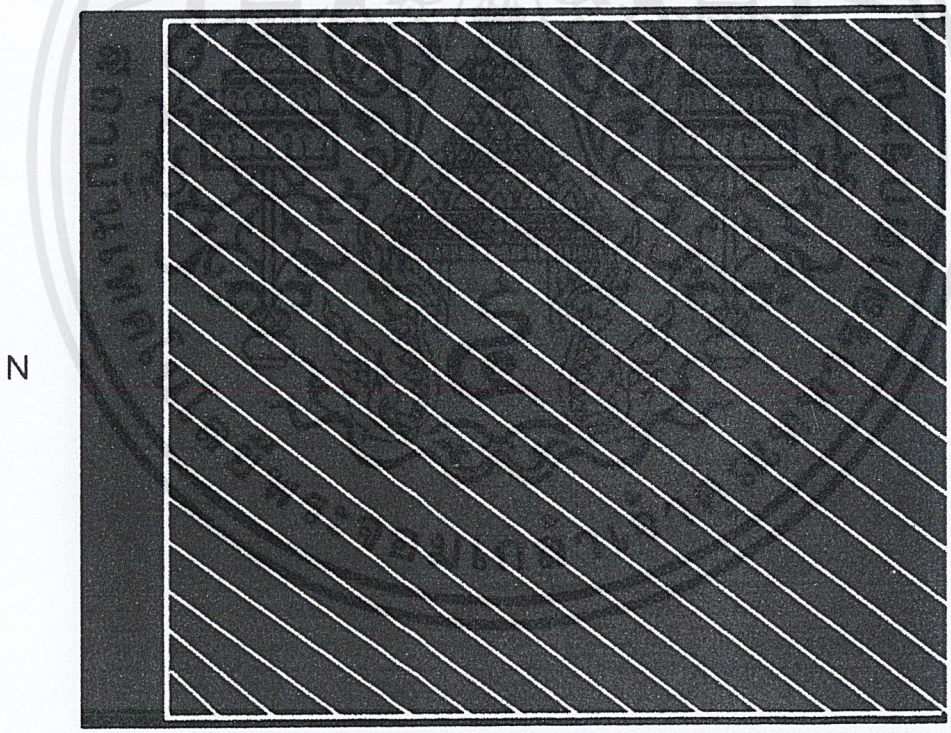
8.3 ปรับปรุง Algorithm ในการใช้ Sliding Window

จากการเขียนโปรแกรมในลักษณะเดิม เราได้มีการพิจารณาข้อผิดพลาดในการเขียนโปรแกรมว่าสาเหตุใด เป็นสาเหตุของความล้าช้าในการใช้ Sliding Window ซึ่งหากเราพิจารณาจากรูปที่ 8-8 จะพบว่าเรามีการค้นหาจุดที่เป็นสีดำซ้ำกันในแต่ละ Window โดยจะดูได้จากพื้นที่ส่วนที่ซ้อนทับกันของ Window หากเรามาลองคำนวณจุดที่ที่ใช้ในการค้นหาจะได้ว่า

Sliding Window มีขนาด = N

จำนวนจุดการค้นหาตามแนวขอบบนขอบล่างในภาพเดียว = X

จำนวนจุดที่ต้องค้นหา คือ $\sum_{i=1}^{X} (N^2)$

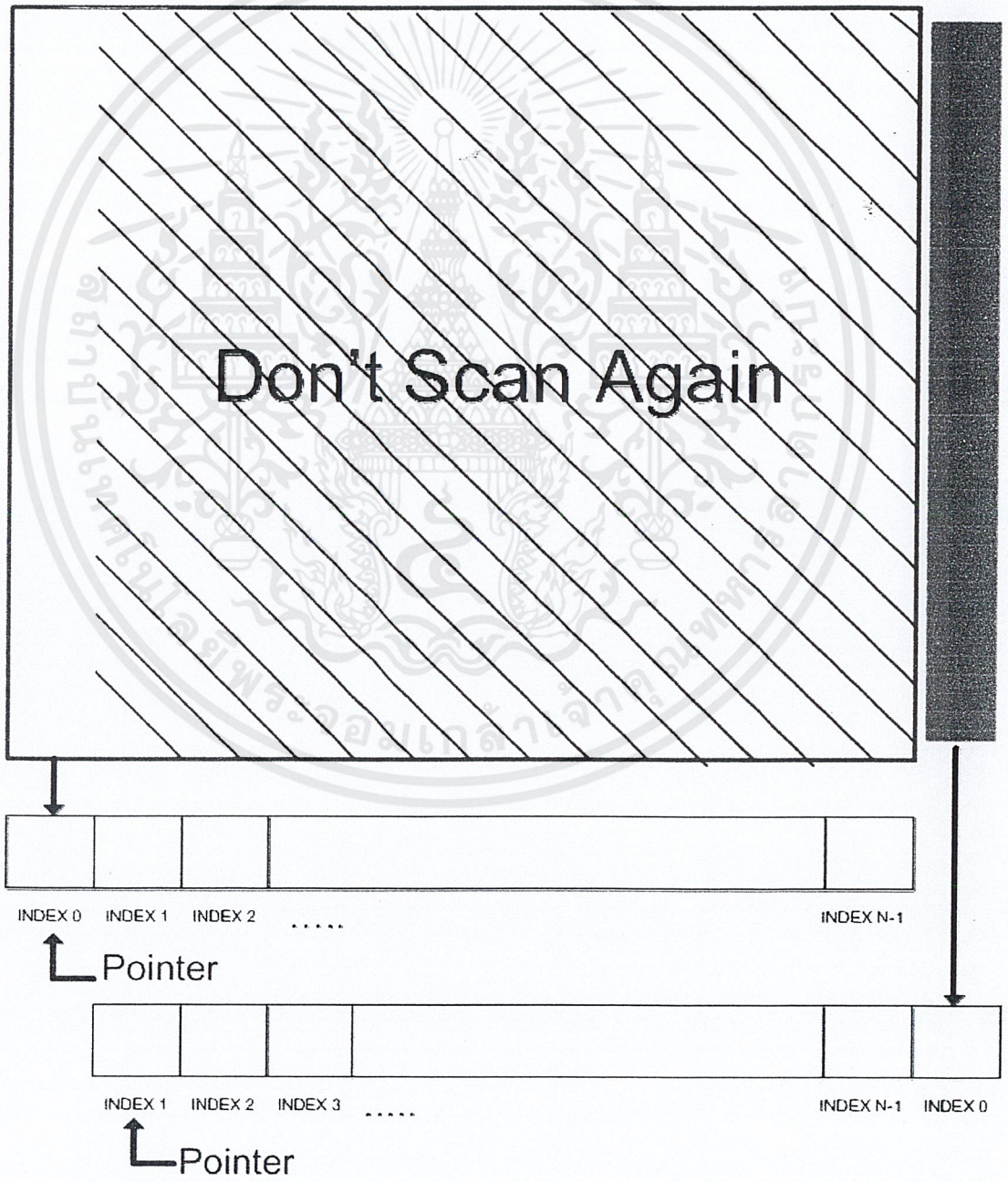


รูปที่ 8-8 การค้นหาจุดสีดำของSliding Windowแบบเดิม ซึ่งมีส่วนซ้อนทับกัน ทำให้เกิดความล่าช้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราทำการปรับปรุงโดยการสร้างบัฟเฟอร์ ตัวหนึ่งขึ้นซึ่งเป็น อาร์เรย์ 1 มิติ มีขนาดเท่ากับความกว้างของ Sliding Window เพื่อทำการเก็บผลรวมของจุดสีดำในแต่ละคอลัมน์ ซึ่งบัฟเฟอร์นี้เราจะมองในลักษณะของวงแหวน คือ แบบคิว (Queue) โดยจะมีตัวชี้ (Pointer) ชี้ไปยังคอลัมน์แรกของแต่ละ Window ดังเห็นได้จากรูปที่ 8-9

การค้นหาแบบนี้จะไม่ค้นหาจุดที่เคยค้นแล้วโดยเราจะเก็บผลรวมของจุดสีดำในแต่ละคอลัมน์ไว้ในบัฟเฟอร์ หากต้องการหาผลรวมของจุดสีดำใน Window ก็จะทำการค้นหาเพียง 1 คอลัมน์เท่านั้น แล้วนำใส่ลงไปในบัฟเฟอร์ในคอลัมน์ที่ไม่อยู่ใน Window จากนั้นจะมารวมค่าที่อยู่บัฟเฟอร์ทั้งหมด ก็จะได้จุดสีดำทั้งหมดใน Window นั้น



รูปที่ 8-9 การค้นหาจุดสีดำของ Sliding Window แบบใหม่ ซึ่งจะไม่ค้นหาส่วนที่เคยค้นหาแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากเรามาลองคำนวณจุดที่ที่ใช้ในการค้นหาจะได้ว่า

Sliding Window มีขนาด = N

จำนวนจุดการค้นหาตามแนวขอบบนขอบล่างในภาพเดียว = X

จำนวนจุดที่ต้องค้นหา คือ N x X

จะสังเกตได้ว่าจะลดจำนวนจุดที่ใช้ในการค้นหาลงเป็นจำนวนมาก ซึ่งช่วยลดเวลาในการค้าหาลงไปมาก

จากการทดลองหลังการปรับปรุงการใช้ Sliding Window 1 ภาพ ใช้เวลาประมาณไม่ถึง 1 วินาที ซึ่งเร็วขึ้นอย่างมาก

8.4 ขั้นตอนการหาจุดศูนย์กลางของดวงตา

หลังจากเราใช้ Sliding Window หาบริเวณที่เป็นของตาดำได้แล้วนั้น เราจะได้ค่าที่เป็นช่วงของด้านซ้ายและด้านขวาของตาดำ เราจะนำมาคำนวณพร้อมกับขอบล่างขอบบนของตาดำเพื่อทำการหาจุดศูนย์กลางได้ดังนี้

จุดตามแนวแกน X = (ด้านซ้าย+ด้านขวา) / 2

จุดตามแนวแกน Y = (ขอบล่าง + ขอบบน) / 2

จุดศูนย์กลางของดวงตา = (จุดตามแนวแกน X, จุดตามแนวแกน Y)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9
การกำจัดสิ่งรบกวน

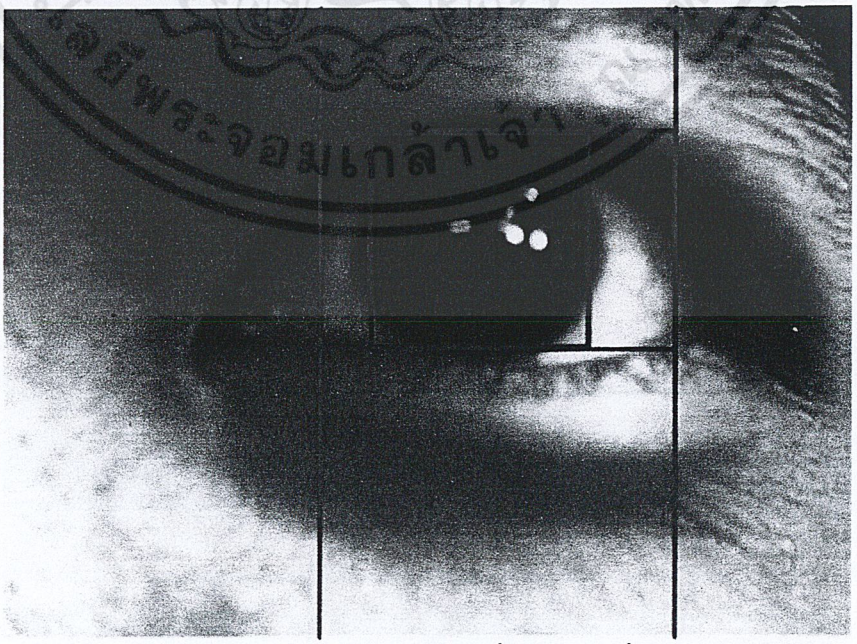
การหาจุดศูนย์กลางของดวงตานั้นมีสิ่งที่สำคัญในการที่ทำให้เรานั้นสามารถค้นหาจุดศูนย์กลางได้ถูกต้องแม่นยำยิ่ง คือ การกำจัดสิ่งรบกวน ซึ่งเป็นการตัดส่วนที่เป็นอุปสรรคในการหาจุดศูนย์กลางของดวงตาทำให้ไม่สามารถหาตำแหน่งของดวงตาที่ถูกต้องได้ และสิ่งที่เราจำเป็นต้องกำจัดซึ่งเป็นสิ่งสำคัญที่สุดก็ คือ เส้นผมของผู้ใช้ ซึ่งเส้นผมของคนเอเชียนั้นมีสีเดียวกับตาดำ ทำให้เป็นสิ่งที่ยากยิ่งในการที่จะกำจัดส่วนนั้นออกไป

โดยขั้นตอนการกำจัดสิ่งรบกวนนั้นเราจะทำไปพร้อมๆกับการปรับปรุงการหาจุดศูนย์กลางของตาแบบเดิมที่กล่าวในบทที่ 8 ให้มีประสิทธิภาพสูงขึ้นกว่าเดิม โดยเราเริ่มทดลองจากภาพนิ่งนามสกุล Bitmap ก่อนที่จะนำไปใช้กับภาพเคลื่อนไหวจริง โดยเราสามารถแบ่งส่วนสำคัญของการกำจัดสิ่งรบกวนได้ดังนี้ได้ดังนี้

9.1 หลักการสำคัญของการกำจัดสิ่งรบกวน

สิ่งสำคัญที่สุดของการกำจัดสิ่งรบกวน คือ ทำอย่างจึงจะกำจัดส่วนที่เป็นผมออกไปให้ได้ โดยปัญหาหลักของมีส่งรบกวนคือ หากมีเส้นผมเข้ามาในภาพจะทำให้เราค้นหาส่วนของขอบล่างของภาพผิด ซึ่งส่วนของการหาขอบล่างของตานั้นเป็นส่วนที่บอบบางที่สุดเนื่องจากเป็นส่วนแรกที่เราจะใช้หาจุดศูนย์กลางของดวงตา อีกยังเป็นส่วนที่มีการตรงสอบน้อยที่สุดเพราะเช็คเพียงจุดๆเดียว หากเส้นผมอยู่ต่ำกว่าขอบล่างของดวงตา จะทำให้โปรแกรมต้นแบบค้นหาผิดเข้าใจว่าจุดล่างสุดของดวงตาคือเส้นผม

ดังนั้นเราจะต้องกำหนดขอบเขตของการค้นหาขอบบนและขอบล่างของดวงตาโดยพยายามตัดส่วนที่เป็นผมออกไปให้ได้มากที่สุด ดังรูปที่ 9-1



รูปที่ 9-1 แสดงหลักการสำคัญของการกำจัดสิ่งรบกวน เพื่อหาช่วงการค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.2 ขั้นตอนของการกำจัดสิ่งรบกวน

ขั้นตอนของการกำจัดสิ่งรบกวนนั้นเราจะใช้เพียงค่าช่วงสีที่เราสนใจ(Threshold) เฉพาะ “สีดำ” เพียงอย่างเดียวไม่ได้ เราจะต้องนำเอาช่วงที่เป็นค่าสีของ “สีขาว” มาช่วยประกอบในการค้นหา แต่ช่วงสีขาวของบริเวณตาขาวของคนเรานั้นจะไม่เป็นสีขาวที่แท้จริงจะมีสีอื่นๆปะปนเข้ามาอยู่มากทำให้ต้องมีช่วงในการพิจาณาสูงมากทำให้อาจมีของผิดพลาดสูงเราจึงนำเอาค่าของ “สีผิว” มาช่วยในการประกอบอีกเพื่อตัดสภาพแวดล้อมด้านหลังออกไปอีกด้วย

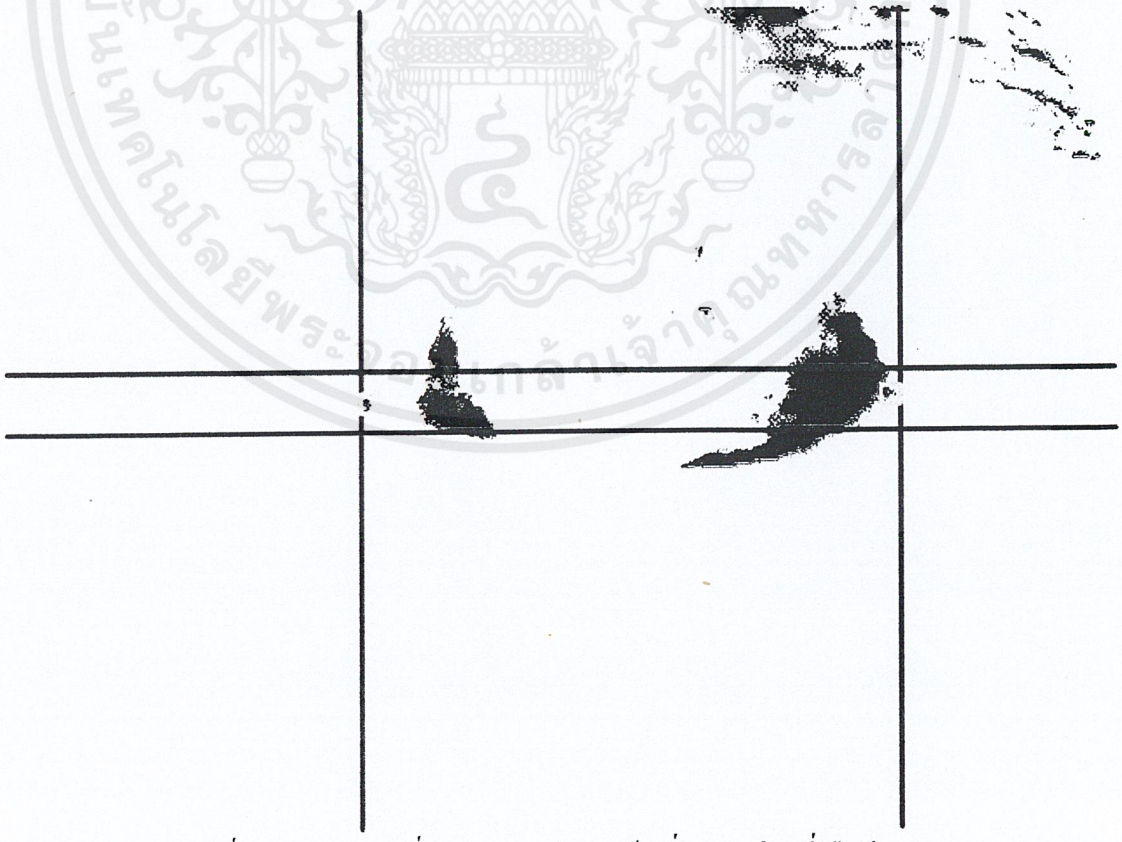
โดยค่าของช่วงที่เราสนใจของ “สีขาว” นั้น จะต้องมีค่า RGB คือ

1. ค่า RGB จะต้องมีค่ามากกว่า 120
2. ผลต่างระหว่างค่า R กับ G จะต้องมีค่าน้อยกว่า 20
3. ผลต่างระหว่างค่า R กับ B จะต้องมีค่าน้อยกว่า 20
4. ผลต่างระหว่างค่า G กับ B จะต้องมีค่าน้อยกว่า 20

โดยค่าของช่วงที่เราสนใจของ “สีผิว” นั้น จะต้องมีค่า RGB คือ

1. ค่า R หรือ G มีค่ามากกว่า B 38

(ค่าของ “สีผิว” นั้นเราเอามาใช้เพียงแค่ประกอบในการตัดสิ่งรบกวนจึงไม่ต้องมีค่าความละเอียดมากนัก)



รูปที่ 9-2 ภาพดวงตาที่ผ่านการทำเซ็กเมนต์เตชั่น ของส่วนที่เป็นสีขาวของดวงตา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปภาพที่ 9-2 แสดงให้เห็นที่ภาพผ่านการทำเซ็กเมนต์เตชั่น ของส่วนที่เป็นสีขาวของดวงตา โดยการเปลี่ยนช่วงสีที่เป็นสีขาวให้เป็นสีดำ และสีอื่นให้เป็นสีขาวเพื่อให้ง่ายต่อการพิจารณา

โดยขั้นตอนของการกำจัดสิ่งรบกวนนั้นการใช้ตาซ้ายหรือตาขวาจะมีผลช่วยในการกำจัดสิ่งรบกวนได้ง่ายยิ่งขึ้นโดย ภาพที่แสดงทั้งหมดต่อไปนี้จะเป็นการกำจัดสิ่งรบกวนของ **"ตาซ้าย"** โดยหากใช้ **"ตาขวา"** ก็ต้องใช้ขั้นตอนในการกำจัดสิ่งกวนในทิศทางตรงกันข้ามกัน

ขั้นตอนการกำจัดสิ่งรบกวนของ "ตาซ้าย" แบ่งออกเป็นขั้นตอนดังต่อไปนี้ คือ

9.2.1 การหาขอบเขตในการค้นหาด้านซ้ายของภาพ

ในการกำจัดสิ่งรบกวนเราจะเริ่มหาขอบเขตในการค้นหาทั้งด้านซ้ายและด้านขวาไปพร้อมๆกัน แต่ในที่นี้จะพูดถึงเฉพาะการค้นหาขอบเขตในการค้นหาเฉพาะด้านซ้ายเท่านั้น

โดยการค้นหาจะเริ่มค้นหาทีละคอลัมน์จากด้านซ้ายมาด้านขวา กล่าวคือ การค้นหาทีละคอลัมน์จะค้นหาตามแนวแกน Y จากล่างขึ้นข้างบน เช่น ภาพมีขนาด 256x192 (X = 0 -255,Y = 0-191) เริ่มที่คอลัมน์ที่ 0 (แกน X = 0) จะค้นหาจากจุด (0,191) จากล่างขึ้นบนไปยังจุด (0,0) แล้วเลื่อนจากซ้ายไปขวาจากคอลัมน์ที่ 0,1,2,3,…,255

การค้นหาด้านซ้ายของภาพนั้นเราจะพิจารณาทั้งจุดที่เป็นสีขาวและสีดำเนื่องจากเราใช้ตาซ้ายเราจะถือว่าด้านซ้ายของภาพจะต้องไม่มีผมอยู่ ดังนั้นจึงสรุปได้ว่าหาเจอสีขาว จะหมายถึง ตาขาว หรือหาเจอสีดำ จะหมายถึง ตาดำหรือคิ้วหรือเงาของหนังตาก็ได้ เนื่องเราต้องแค่ช่วงในการค้นหาตาต่อไป คิ้วจึงไม่มีผล (เพราะการค้นหาขอบล่างนั้นเราเริ่มจากล่างขึ้นบนจึงไม่มีปัญหาเรื่องคิ้วหรือเงาของหนังตา)

โดยก่อนที่เราจะสรุปว่าเป็นตาขาวจริงนั้นเมื่อเราเจอจุดสีขาวหรือดำจะมีการใช้ Window เล็กที่มีขนาด 5x5 Pixel เพื่อนับจำนวนจุดว่าจะต้องมีจุดสีขาวมากกว่าที่ตั้งไว้จึงถือว่าเป็นตาขาว เนื่องจากช่วงของสีขาวนั้นกว้างมากอาจมีจุดสีขาวที่เกิดจากเงาของแสงดังนั้นวิธีนี้จึงเป็นวิธีลดข้อบกพร่องลงได้บางส่วน ดังจะแสดงให้เห็นในรูปที่ 9-2 เส้นตามแนวแกน Y เส้นซ้ายมือของภาพ

9.2.2 การหาขอบเขตในการค้นหาด้านขวาของภาพ

ในการหาขอบเขตในการค้นหาด้านขวาของภาพจะทำในทำนองเดียวกับการหาขอบเขตด้านซ้ายและทำไปพร้อมๆกัน คือจะเริ่มค้นหาทีละคอลัมน์จากด้านขวามาด้านซ้าย โดยเริ่มที่คอลัมน์ที่ 255 (แกน X = 255) แล้วเลื่อนจากขวาไปซ้าย จากคอลัมน์ที่ 255,254,253,…,0

โดยการค้นหาด้านขวาของภาพนั้นเราจะพิจารณาเฉพาะจุดที่เป็นสีขาวเท่านั้น เนื่องจากเราใช้ตาซ้ายเพราะฉะนั้นด้านขวาของภาพจะมีเส้นผมที่เราต้องการกำจัดอยู่อยู่ ดังนั้นจะสรุปได้เฉพาะเมื่อเจอสีขาว จะหมายถึง ตาขาว เท่านั้น แต่ถ้าหากเจอสีดำ เราจะไม่รู้สิ่งที่เจอว่าเป็นเส้นผม หรือ ตาดำ โดยในลักษณะที่ตามองด้านซ้ายสุดไม่มีสีขาวด้านซ้ายเลย(จากภาพที่ถ่ายมาจะเป็นด้านขวา) จะกล่าวถึงการแก้ในภายหลัง ในหัวข้อที่ 9.2.3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.2.3 ข้อจำกัดในการหาขอบเขตการค้นหาของภาพ

ข้อจำกัดในการกำหนดขอบเขตของภาพนั้นมีหลายลักษณะเพื่อช่วยในการลดข้อผิดพลาดในการหาขอบเขตการค้นหา จะมีลักษณะดังต่อไปนี้

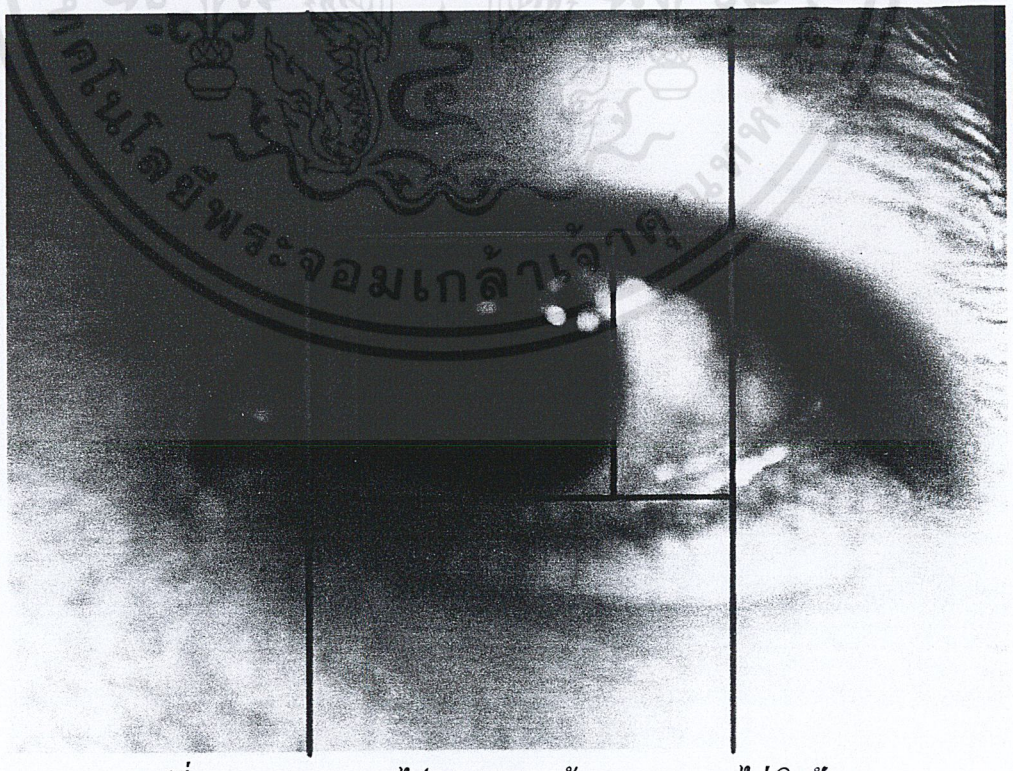
9.2.3.1 ขอบเขตในการค้นหาด้านซ้ายและด้านขวาของภาพจะต้องอยู่แนวเดียวกัน

เราสามารถดูได้จากรูปที่ 9-2 จะมีเส้นในแนวนอน 2 เส้น นั้นคือ ขอบเขตในการค้นหาด้านซ้ายและด้านขวาของภาพจะต้องอยู่แนวเดียวกัน ไม่เกิน +/-10 Pixel ของจุดที่ใช้เป็นขอบเขตในการค้นหาด้านซ้ายและด้านขวาของภาพ เพราะว่า จะพิจารณาได้จากส่วนบนขวาของรูปที่ 9-2 จะมีส่วนของเงาที่นำมาพิจารณาเป็นสีขาวด้วย เราจึงแก้โดยใช้วิธีนี้ โดยจะพิจาณากับแนวจุดของแนวที่คาดว่าน่าจะเป็นขอบเขตในการค้นหาที่ถูกต้อง

โดยหากด้านไหนเจอก่อนยังไม่สรุปจะรอของอีกด้านหนึ่งแล้วนำมาเปรียบเทียบว่าอยู่แนวเดียวกันหรือไม่ หากไม่ด้านขวาก็จะทำการค้นหาต่อโดยจะไม่ค้นหาแนวที่เคยพบแล้ว แต่จะค้นหาในแนวของด้านซ้ายแทน เนื่องจากเงาส่วนมากจะเกินจากด้านซ้ายของใบหน้าเราเนื่องจากเป็นส่วนโค้งเว้าอยู่ หากด้านซ้ายค้นหาผิดก็ไม่มีปัญหาเนื่องจากด้านซ้ายไม่มีปัญหาเรื่องของเส้นผมอยู่

9.2.3.2 ตามองไปด้านซ้ายสุด(ขวาสุดของภาพ) จะต้องปรับปรุงการกำหนดขอบเขตค้นหาเล็กน้อยและการใช้ Sliding Window

จากรูปที่ 9-3 ภาพดวงตามองไปทางขวาสุด(ซ้ายของภาพ)จนไม่มีสีขาวเลยที่ด้านซ้ายเลยของตาดำ จะไม่เกิดปัญหาเนื่องจากการหาขอบเขตในการค้นหาด้านซ้ายของภาพ จะค้นหาจุดได้ทั้งสีขาวและสีดำเนื่องจากไม่มีเส้นผมอยู่จึงไม่เกิดปัญหาขึ้น



รูปที่ 9-3 ภาพดวงตามองไปทางขวาสุด(ซ้ายของภาพ) จะไม่เกิดปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดขอบเขตในการค้นหาด้านซ้ายจะใช้ช่วงของสีดำเป็นตัวกำหนดขอบเขตได้ เนื่องจากถือไม่มีเส้นผมอยู่ ดังจะเห็นได้จากรูปที่ 9-4 จะเป็นการแสดงภาพดวงตาที่ผ่านการทำเซ็กเมนต์เตชั่น โดยพิจารณาในที่ช่วงเป็นสีดำ โดยส่วนที่เป็นช่วงสีดำจะแสดงเป็นสีดำ หากเป็นสีอื่นจะเป็นสีขาวเพื่อให้ง่ายต่อการพิจารณา

รูปที่ 9-4 ภาพดวงตาที่ผ่านการทำเซ็กเมนต์เตชั่นในที่ช่วงเป็น "สีดำ" โดยใช้ค้นหาขอบเขตด้านซ้าย

โดยก่อนที่เราจะสรุปว่าเป็นตาดำจริงนั้นเมื่อเราเจอจุดสีดำจะมีการใช้ Window เล็กที่มีขนาด 5x5 Pixel เพื่อนับจำนวนจุดว่าจะต้องมีจุดสีดำมากกว่าที่ตั้งไว้จึงถือว่าเป็นตาดำจริง วิธีนี้จึงเป็นวิธีลดข้อบกพร่องลงได้บางส่วน ดังจะแสดงให้เห็นในรูปที่ 9-4

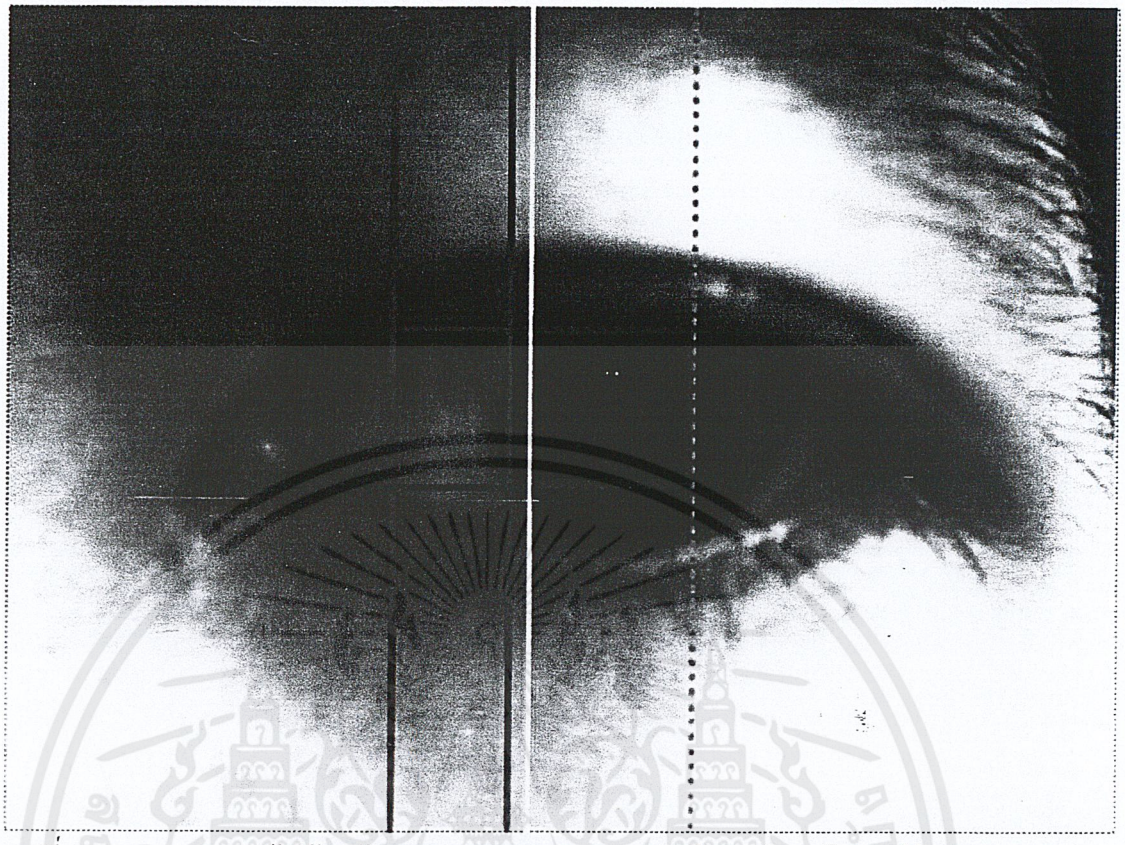
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดขอบเขตในการค้นหาด้านขวาจะใช้ช่วงของสีขาวเป็นตัวกำหนดขอบเขตเพียงอย่างเดียว เนื่องจากถือมีเส้นผมอยู่ ดังจะเห็นได้จากรูปที่ 9-5 จะเป็นการแสดงภาพดวงตาที่ผ่านการทำเซ็กเมนต์เตชั่นโดยพิจารณาในที่ช่วงเป็นสีขาว โดยส่วนที่เป็นช่วงสีขาวจะแสดงเป็นสีดำ หากเป็นสีอื่นจะเป็นสีขาว

รูปที่ 9-5 ภาพดวงตาที่ผ่านการทำเซ็กเมนต์เตชั่นในที่ช่วงเป็น "สีขาว" โดยใช้ค้นหาขอบเขตด้านขวา

จากรูปที่ 9-3, 9-4, 9-5 แสดงว่าหากตามองไปทางขวาสุด (ซ้ายของภาพ) จนไม่มีสีขาวเลยที่ด้านซ้ายเลยของตาดำ จะไม่เกิดปัญหา แต่ถ้าหากตามองไปด้านซ้ายสุด (ขวาสุดของภาพ) จะเกิดปัญหาเนื่องจากขอบเขตในหารค้นหาด้านขวาไม่ได้ครอบคลุมตาดำไปด้วยเนื่องจากเราต้องตัดสีดำทั้งหมดที่อยู่ด้านขวาทิ้งไปเนื่องจากมีผมอยู่ ดังจะเห็นได้จากรูปที่ 9-6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



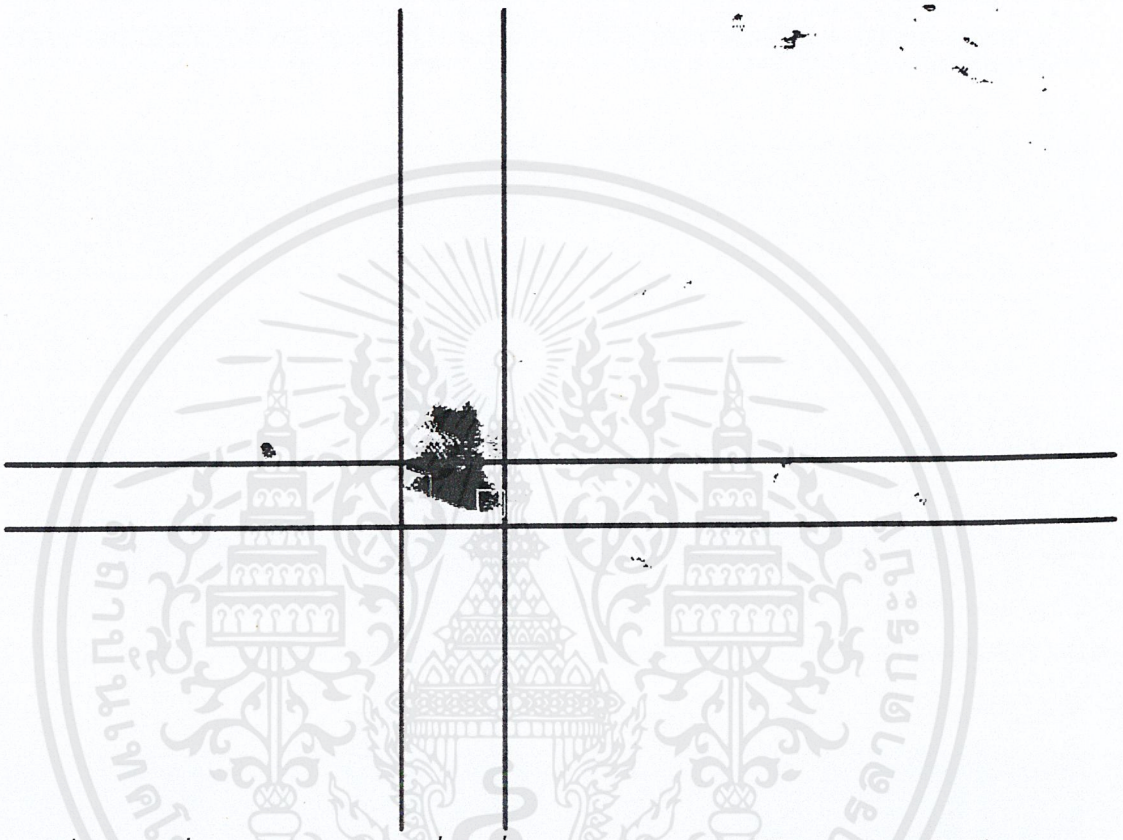
รูปที่ 9-6 ภาพดวงตามองไปด้านซ้ายสุด(ขวาสุดของภาพ) จะเกิดปัญหาต้องแก้ไขการใช้ Sliding Window

ขั้นต้นปรับปรุงขอบเขตในการค้นหาด้านขวาโดยเพิ่มขอบเขตอีก 5 Pixel เพื่อเพิ่มขอบเขตในการหาขอบเขตบนและล่าง เพราะหากไม่เพิ่มอาจจะไม่สามารถค้นหาขอบเขตบนและล่างได้เลย

เราจะแก้โดยการปรับปรุงการใช้ Sliding Window โดยขั้นตอนการหาขอบล่างขอบบนยังเหมือนเดิมโดยจะค้นหาเฉพาะในขอบเขตที่หามาได้เพื่อกำจัดสิ่งรบกวน แต่ส่วนของการหาด้านซ้ายและด้านขวาของตาที่ใช้ Sliding Window จะต้องปรับปรุง โดยตอนเริ่มใช้Sliding Window ยังเหมือนเดิม คือ ค้นหาสีดำก่อน 3 แถว จึงจะเริ่มใช้ ส่วนที่ต้องแก้ไข คือ จะขยายส่วนการใช้Sliding Window ให้กว้างเพิ่มขึ้นกว่าขอบเขตการค้นหาด้านขวาเป็นขนาด 1 เท่าของขนาด Sliding Window ที่หาได้จากการหาขอบเขตบนขอบเขตล่าง เพื่อขยายส่วนของใช้Sliding Window ที่จะสามารถครอบคลุมตาที่ขอบเขตการค้นหาไม่ได้ครอบคลุมไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยภาพที่ 9-7 แสดงให้เห็นที่การค้นหาขอบเขตด้านซ้ายและด้านขวาเพื่อตัวสิ่งรบกวน โดยผ่านการทำเซ็กเมนต์เตชั่น ของส่วนที่เป็นสีขาวของควงตาโดยการเปลี่ยนช่วงสีที่เป็นสีขาวให้เป็นสีดำ และสีอื่นให้เป็นสีขาวเพื่อให้ง่ายต่อการพิจารณา



รูปที่ 9-7 ภาพที่ผ่านการทำเซ็กเมนต์เตชั่นในที่ช่วงเป็น "สีขาว" โดยใช้ค้นหาขอบเขตด้านซ้ายและขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 10
การแสดงผลตำแหน่งบนหน้าจอ

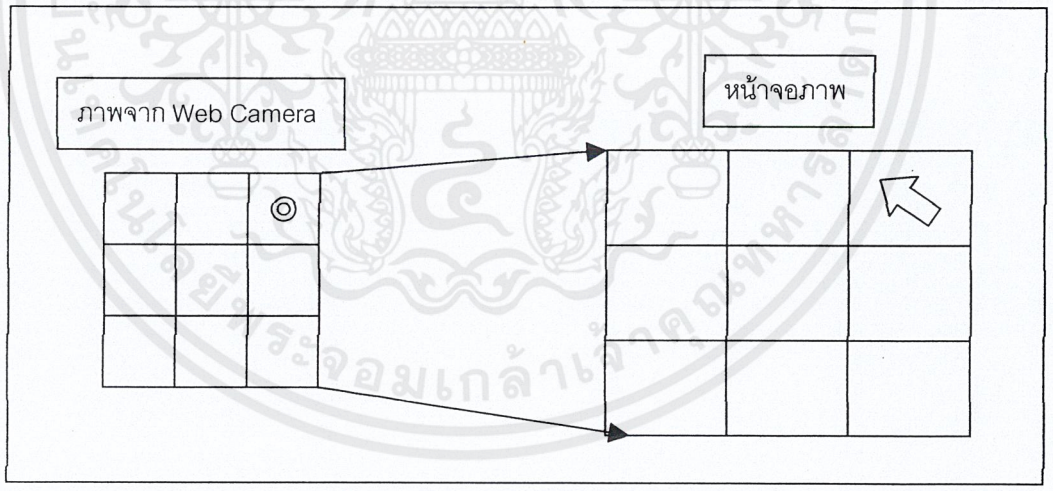
10.1 เคอร์เซอร์ (Cursor)

เคอร์เซอร์ หมายถึง ไอคอนที่ใช้แสดงตำแหน่งปัจจุบันของเมาส์สำหรับแต่ละแอปพลิเคชัน แต่เนื่องจากวินโดวส์ 32 บิต สนับสนุนการทำงานของแอปพลิเคชันในระดับเธรด (Thread) ดังนั้น เคอร์เซอร์สำหรับแต่ละแอปพลิเคชันที่ทำงานภายใต้วินโดวส์ 32 บิต จึงหมายถึงไอคอนที่ใช้แสดงตำแหน่งปัจจุบันของเมาส์สำหรับ เธรด ที่อยู่ในสถานะรับโฟกัสนั่นเอง

การกำหนดไอคอนสำหรับเคอร์เซอร์สามารถทำได้ผ่านทางคุณสมบัติ MousePointer ของคอนโทรล ActiveX

10.2 การแสดงผลตำแหน่งบนหน้าจอ

การแปลงตำแหน่งจุดที่มองจ้องเป็นตำแหน่ง Pointer ของเมาส์ นั้นใช้หลักการแบบ ปากกา คือ เมื่อตามองไปจุดไหนให้เมาส์เคลื่อนไปจุดนั้นทันที ในขณะที่เมาส์ปกติที่ใช้อยู่ใช้หลักการแบบ Joy Stick คือเมื่อเคลื่อนเมาส์ไปทางซ้าย ตำแหน่ง Pointer ของเมาส์ก็จะเคลื่อนไปทางซ้ายไม่ได้ไปจุดที่เราต้องการได้ทันที



รูปที่ 10-1 การแสดงผลตำแหน่งบนหน้าจอ

จากรูปที่ 10-1

◎ คือตำแหน่งจุดศูนย์กลางของตา

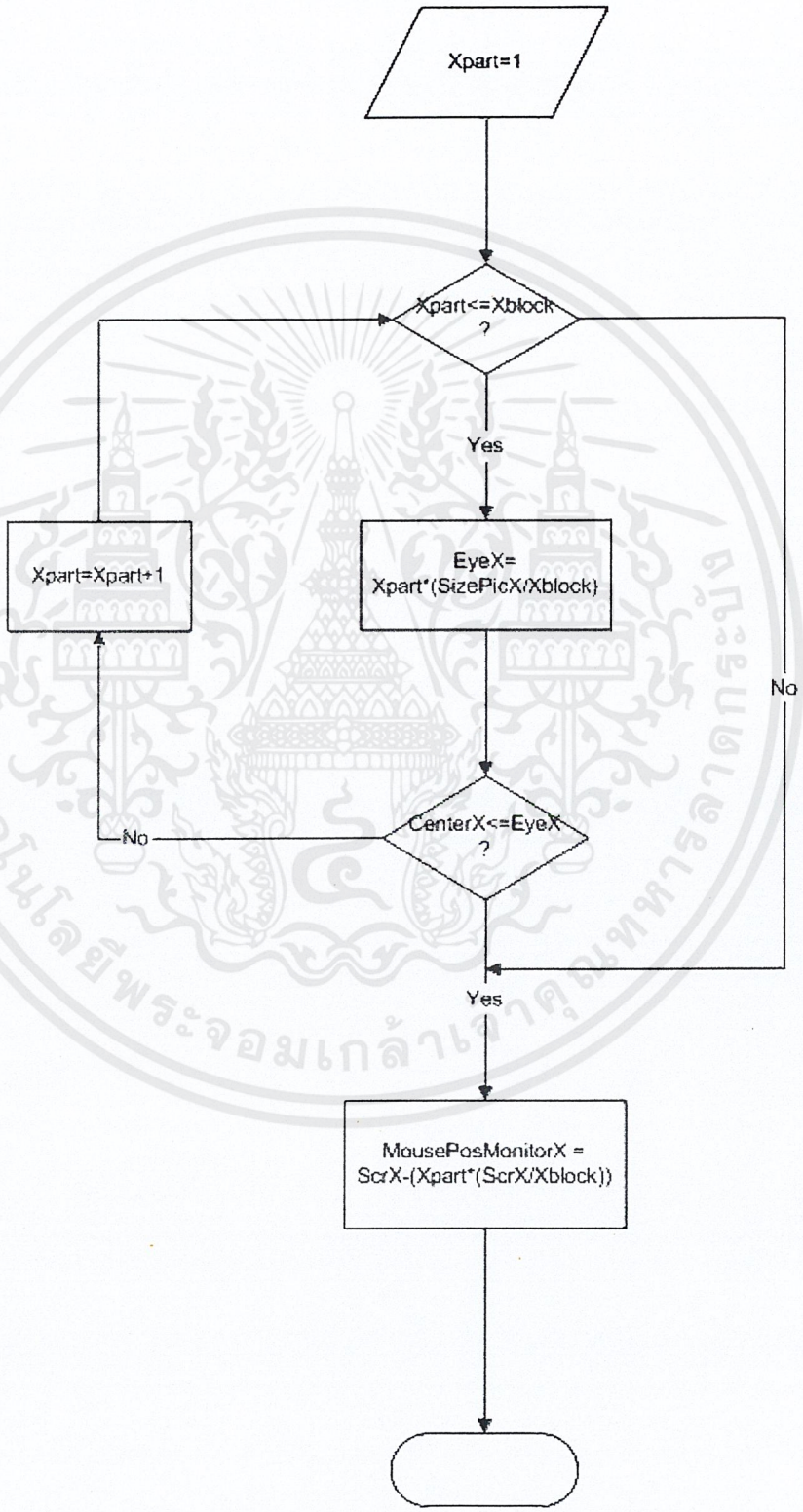
↖ คือตำแหน่งเมาส์บนจอภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.3 หลักการแปลงตำแหน่งจุดศูนย์กลางของตาเป็นตำแหน่งเมาส์

ขนาดภาพของ Web Camera = 256x192

ขนาดของจอภาพ (Monitor) = 1024x768



รูปที่10-2 Flow Chart การแปลงตำแหน่งจุดศูนย์กลางของตาเป็นตำแหน่งเมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่10-2

Xpart : เป็นค่าที่ไว้บอกว่า CenterX อยู่ ช่องที่เท่าไหรในแนวแกน X

Xblock: จำนวนช่องทั้งหมดในแนวแกน X

SizePicX: ขนาดภาพของ Web Camera ตามแนวแกน X (ในที่นี้มีค่าเป็น 256)

EyeX: ผลลัพธ์จากการคำนวณ Xpart*(SizePicX/Xblock)

CenterX: จุดศูนย์กลางดวงตา

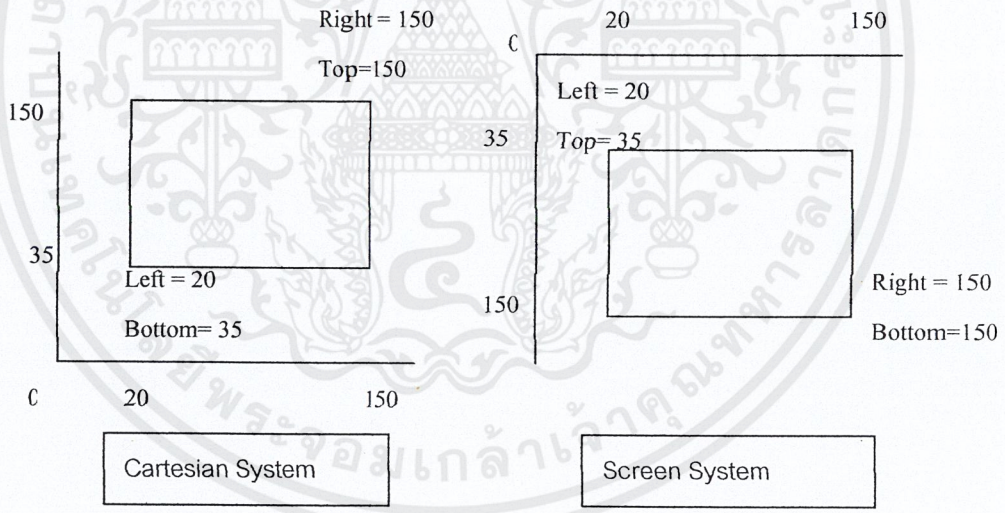
MousePosMonitorX : ตำแหน่งของเมาส์บนหน้าจอ

MousePosMonitorX = ScrX-(Xpart*(ScrX/Xblock)) ที่ทำการลบ เพื่อทำการกลับค่าซ้ายเป็นขวา และ จากขวาเป็นซ้าย (Mirror)

ScrX: ขนาดของจอภาพตามแนวแกน X (ในที่นี้มีค่าเป็น 1024)

ในการหาตำแหน่งในแนวแกน Y ทำเช่นเดียวกับการหาในแนวแกน X

10.4 Window API ที่ใช้ในการกำหนดตำแหน่งของเคอร์เซอร์เมาส์



รูปที่ 10-3 แสดงการกำหนดตำแหน่งโคออร์ดิเนตของสี่เหลี่ยม

Type POINTAPI

x as Long

y as Long

End Type

สำหรับการกำหนดตำแหน่งโคออร์ดิเนตของจุด(x,y) ดังนั้น

x หมายถึง โคออร์ดิเนต x

y หมายถึง โคออร์ดิเนต y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GetCursorPos

การประกาศ	*Public Declare Function GetCursorPos Lib "User32" _*
	(lpPoint As POINTAPI) As Long
หน้าที่	อ่านค่าตำแหน่งโคออร์ดิเนตปัจจุบันของเคอร์เซอร์ของเมาส์
พารามิเตอร์	lpPoint ข้อมูลชนิดโครงสร้าง POINTAPI เพื่อใช้จัดเก็บตำแหน่งโคออร์ดิเนตของเคอร์เซอร์ ซึ่งตัวเลขที่จะวัดในหน่วยของความละเอียดของจอภาพ (Pixel)
ตัวอย่างการใช้	Dim lpPoint AS POINTAPI
	Return& = GetCursorPos(lpPoint)
ค่าที่ส่งกลับ	= 0 หมายถึง ฟังก์ชันทำงานไม่สำเร็จ
	<> 0 หมายถึง ฟังก์ชันทำงานสำเร็จ

SetCursorPos

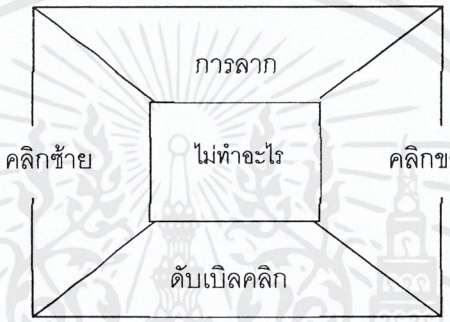
การประกาศ	*Public Declare Function SetCursorPos Lib "User32" _*
	(ByVal x As Long , ByVal y As Long) As Long
หน้าที่	กำหนดตำแหน่งปัจจุบันของเคอร์เซอร์
พารามิเตอร์	x ข้อมูลชนิดจำนวนเต็ม long ที่กำหนดตำแหน่งโคออร์ดิเนตของเคอร์เซอร์ในแกนนอน
	y ข้อมูลชนิดจำนวนเต็ม long ที่กำหนดตำแหน่งโคออร์ดิเนตของเคอร์เซอร์ในแกนตั้ง
ตัวอย่างการใช้	ReVal& = SetCursorPos (hCur, OCR_NORMAL)
ค่าที่ส่งกลับ	= 0 หมายถึง ฟังก์ชันทำงานไม่สำเร็จ
	<> 0 หมายถึง ฟังก์ชันทำงานสำเร็จ
หมายเหตุ	สำหรับตำแหน่งโคออร์ดิเนตของเคอร์เซอร์ที่กำหนดให้กับพารามิเตอร์ x และ y จะถูกวัดในหน่วยของจุด(Pixel) ซึ่งเป็นหน่วยวัดตำแหน่งโคออร์ดิเนตกายภาพของจอภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 11
การคลิก

11.1 หลักการคลิก

การคลิกในโครงงาน *เมาส์ควบคุมด้วยดงตา* นั้นสามารถทำได้โดยการมองจ้องจุดที่เราต้องการคลิกเป็นเวลา 3 วินาที โปรแกรมจะเปลี่ยนรูปแบบการทำงาน จากการเคลื่อนที่ของเมาส์เป็นการคลิก โดยที่ โปรแกรมจะรอรับข้อมูลในเวลา 2 วินาที ข้อมูลที่รอรับนั้นคือตำแหน่งการมองจ้องในตำแหน่งต่างๆ บนหน้าจอตามรูปที่ 11-1



รูปที่11-1 แสดงตำแหน่งการมองจ้องกับการคลิก

ขณะที่โปรแกรมรอรับข้อมูล

เคลื่อนตาไปทางซ้าย ⟶ คลิกซ้าย
เคลื่อนตาไปทางขวา ⟶ คลิกขวา
เคลื่อนตาไปด้านบน ⟶ การลาก
เคลื่อนตาไปด้านล่าง ⟶ ดับเบิลคลิก

หมายเหตุ การปล่อย จะเกิดขึ้นเมื่อทำการลากแล้ว หากต้องการปล่อยสามารถทำได้โดยจ้องมองจุดที่ต้องการปล่อยเท่านั้น

11.2 Window API ที่ใช้ในการคลิก

mouse_event

การประกาศ Public Declare Sub mouse_event Lib "User32" _
(ByVal dwFlags As Long, ByVal dx As Long, ByVal dy As Long, _
, ByVal cButtons As Long, ByVal dwExtraInfo As Long)

หน้าที่ จำลองเหตุการณ์เกี่ยวกับเมาส์

พารามิเตอร์ dwFlags ข้อมูลชนิดเลขจำนวนเต็ม long ที่กำหนด flag เพื่อจำลองเหตุการณ์เกี่ยวกับเมาส์ ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOUSEEVENTF_ABSOLUTE	ค่าพารามิเตอร์ dx และ dy จะเป็นค่าตัวเลขบอกตำแหน่งแบบสมบรูณ์ โดยที่แกนตั้งและแกนนอนของจอภาพจะถูกแบ่งออกเป็น 65,535 หน่วยเท่าๆกัน
MOUSEEVENTF_MOVE	จำลองการเคลื่อนที่
MOUSEEVENTF_LEFTDOWN	จำลองการกดปุ่มมือซ้าย
MOUSEEVENTF_LEFTUP	จำลองการปล่อยปุ่มมือซ้าย
MOUSEEVENTF_RIGHTDOWN	จำลองการกดปุ่มมือขวา
MOUSEEVENTF_RIGHTUP	จำลองการกดปล่อยมือขวา
MOUSEEVENTF_MIDDLEDOWN	จำลองการกดปุ่มกลาง
MOUSEEVENTF_MIDDLEUP	จำลองการปล่อยปุ่มกลาง
dx	ข้อมูลชนิดเลขจำนวนเต็ม long ที่กำหนดตำแหน่งแกนนอน (x)ของเมาส์ซึ่งสามารถเป็นได้ทั้งตำแหน่งสมบรูณ์ หรือตำแหน่งแบบสัมพัทธ์ก็ได้
dy	ข้อมูลชนิดเลขจำนวนเต็ม long ที่กำหนดตำแหน่งแกนนอน (y)ของเมาส์ซึ่งสามารถเป็นได้ทั้งตำแหน่งสมบรูณ์ หรือตำแหน่งแบบสัมพัทธ์ก็ได้
cButton	ข้อมูลชนิดเลขจำนวนเต็ม long ที่ไม่มีการใช้งานในเวอร์ชันปัจจุบันของวินโดวส์API
dwExtraInfo	ข้อมูลชนิดเลขจำนวนเต็ม long ที่ไม่ค่อยมีการใช้งาน แต่ก็ขึ้นกับชนิดของไดเวอร์ของเมาส์ที่ใช้ด้วย

ตัวอย่างการใช้

```
scrnX& =Screen.Width
scrnY& =Screen.Heigth
' Convert to mouse unit
curX& = pt.x * &HFFFF& / scrnX&
curY& = pt.y * &HFFFF& / scrnY&
mouse_event MOUSEEVENTF_ABSOLUTE Or MOUSEEVENTF_LEFTDOWN,_
    curX&, curY&, 0, 0
mouse_event MOUSEEVENTF_ABSOLUTE Or MOUSEEVENTF_LEFTUP,_
    curX&, curY&, 0, 0
```

ค่าที่ส่งกลับ ไม่มี

หมายเหตุ โค้ดตัวอย่างข้างต้น การเรียกใช้ฟังก์ชันวินโดวส์ mouse_event ติดต่อกัน 2 ครั้ง โดยการกำหนดให้พารามิเตอร์ dwFlags มีค่าเท่ากับค่าคงที่ MOUSEEVENTF_LEFTDOWN และ MOUSEEVENTF_LEFTUP เป็นการจำลองการคลิกซ้ายของเมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11.2.1 การใช้ Window API กับการคลิก

การคลิกขวา

```
Call mouse_event(MOUSEEVENTF_RIGHTDOWN, 0, 0, 0, 0)
Call mouse_event(MOUSEEVENTF_RIGHTUP, 0, 0, 0, 0)
```

การคลิกซ้าย

```
Call mouse_event(MOUSEEVENTF_LEFTDOWN, 0, 0, 0, 0)
Call mouse_event(MOUSEEVENTF_LEFTUP, 0, 0, 0, 0)
```

การดับเบิลคลิก

```
Call mouse_event(MOUSEEVENTF_LEFTDOWN, 0, 0, 0, 0)
Call mouse_event(MOUSEEVENTF_LEFTUP, 0, 0, 0, 0)
Call mouse_event(MOUSEEVENTF_LEFTDOWN, 0, 0, 0, 0)
Call mouse_event(MOUSEEVENTF_LEFTUP, 0, 0, 0, 0)
```

การลาก (การจับวัตถุ)

```
Call mouse_event(MOUSEEVENTF_RIGHTDOWN, 0, 0, 0, 0)
```

การปล่อย

```
Call mouse_event(MOUSEEVENTF_RIGHTUP, 0, 0, 0, 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่12

การทดสอบระบบ

12.1 ทดสอบอัตราความเร็วของเมาส์

- สูตรคำนวณหาอัตราความเร็วของเมาส์

  อัตราความเร็วของเมาส์ = ระยะทาง / เวลาในการคลิก หน่วย pixel/Sec.

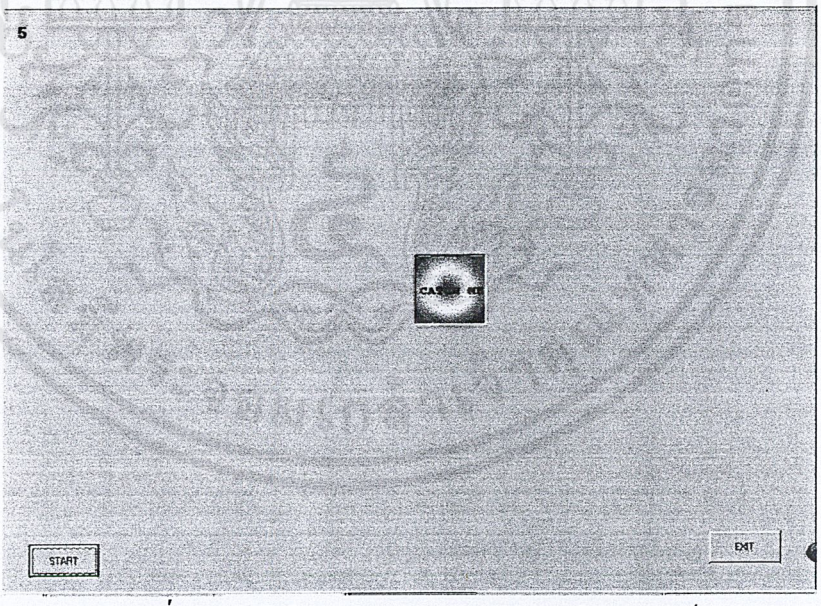
- สูตรคำนวณหาระยะทาง

  ระยะทาง = sqt $(x2-x1)^2+(y2-y1)^2$)

12.1.1 การออกแบบการทดสอบอัตราความเร็วของเมาส์

ตามรูปที่ 12-1

1. เริ่มโดยการกดปุ่ม Start
2. จากนั้นภาพ “Catch me” จะปรากฏขึ้น
3. ให้ใช้เมาส์คลิกซ้ายที่ภาพ “Catch me”
4. ภาพ “Catch me”จะสุ่มตำแหน่งขึ้นมาใหม่เรื่อยๆ
5. เมื่อคลิกครบ 10 ครั้ง หรือ กดปุ่ม Exit จะเป็นการออกจากโปรแกรม



รูปที่ 12-1 หน้าต่างการทดสอบอัตราความเร็วของเมาส์

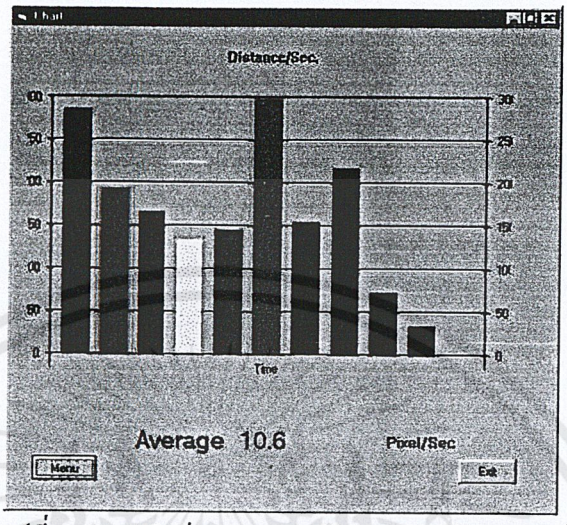
ผลลัพธ์การทดสอบอัตราความเร็วของเมาส์ตามรูปที่12-2 จะแสดงด้วยกราฟแท่ง เป็นอัตราความเร็วในแต่ละครั้งของการคลิก และจะหาค่าเฉลี่ยตามสูตรด้านล่าง

- สูตรคำนวณหาค่าเฉลี่ยอัตราความเร็วของเมาส์

  ค่าเฉลี่ยอัตราความเร็วของเมาส์ = ผลรวมอัตราความเร็วของเมาส์ N ครั้ง/ N

  N = จำนวนครั้งที่คลิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12-2 ผลลัพธ์การทดสอบอัตราความเร็วของเมาส์

12.1.2 ผลลัพธ์การทดสอบอัตราความเร็วของเมาส์

	ค่าเฉลี่ยอัตราความเร็วของเมาส์ (Pixel/Sec)
เมาส์ปกติ	93.98
เมาส์ควบคุมด้วยดวงตา	10.72

ตารางที่12-1การเปรียบเทียบอัตราความเร็วของเมาส์

12.2 การทดสอบการลากและปล่อยของเมาส์

12.2.1 การออกแบบการทดสอบการลากและปล่อยของเมาส์

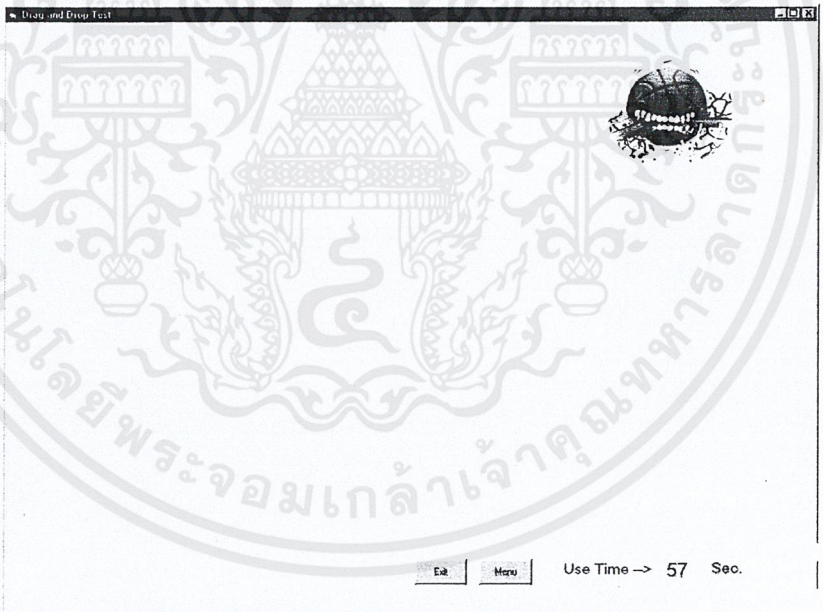
ตามรูปที่12-3

1. เริ่มโดยการกดปุ่ม Start
2. จากนั้นให้ใช้เมาส์ลากลูกบาสเกตบอลลงห่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 12-3 หน้าต่างการทดสอบการลากและปล่อยของเมาส์

ผลลัพธ์เป็นไปตามรูปที่ 12-4 แอปพลิเคชันจะทำการบันทึกเวลาที่ใช้ในการลากลูกบาสเกตบอลลงห่วงสำเร็จ



รูปที่ 12-4 ผลลัพธ์การทดสอบการลากและปล่อยของเมาส์

12.2.2 ผลลัพธ์การทดสอบการลากและปล่อยของเมาส์

	เวลาที่ใช้ในการลากลูกบาสเกตบอลลงห่วง (Sec)
เมาส์ปกติ	2
เมาส์ควบคุมด้วยดวงตา	128

ตารางที่12-2 การเปรียบเทียบเวลาการลากและปล่อยของเมาส์

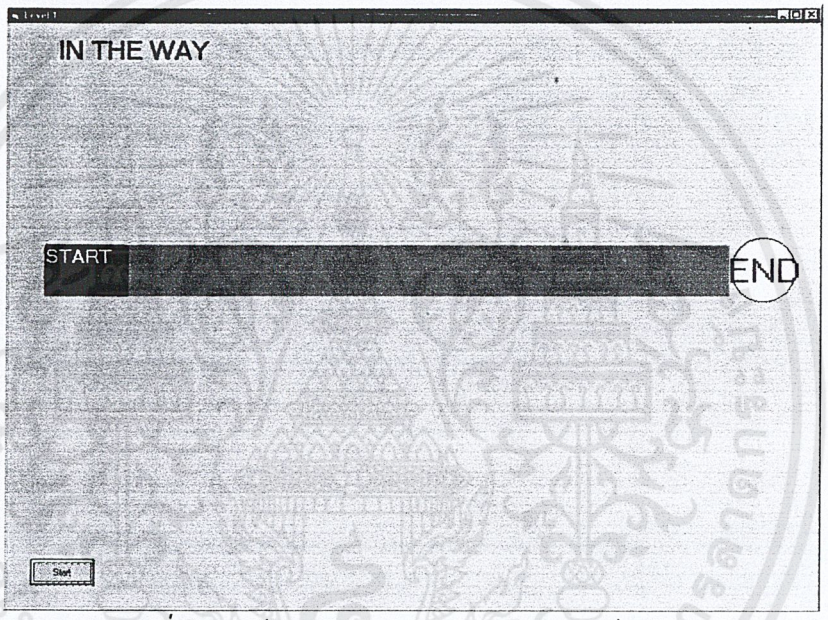
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12.3 การทดสอบการควบคุมเมาส์

12.3.1 การออกแบบการทดสอบการควบคุมเมาส์ระดับ 1

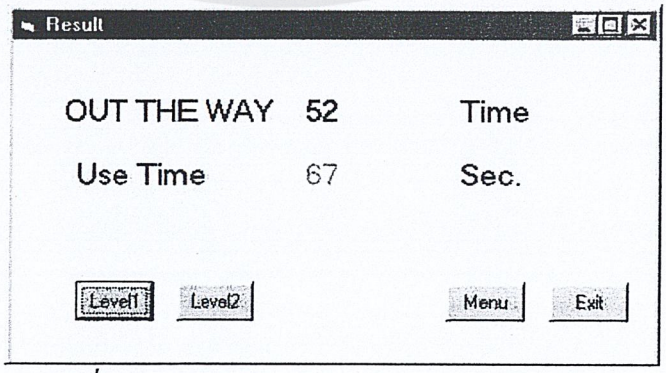
ตามรูปที่ 12-5

1. เริ่มโดนการกดปุ่ม Start
2. จากนั้นเมาส์จะไปอยู่ที่ตำแหน่ง Start (บริเวณพื้นที่สีแดง)
3. ให้เลื่อนเมาส์ไปตำแหน่ง END (บริเวณพื้นที่สีขาว) โดยเลื่อนให้อยู่ตามแนวที่กำหนดไว้ เมื่อเมาส์อยู่ในแนวที่กำหนด ด้านบนซ้ายของหน้าต่างจะขึ้นว่า "IN THE WAY" เมื่อเมาส์ออกจากแนวที่กำหนด ด้านบนซ้ายของหน้าต่างจะขึ้นว่า "OUT THE WAY"



รูปที่12-5 หน้าต่างการทดสอบการควบคุมเมาส์ระดับ 1

ผลลัพธ์เป็นไปตามรูปที่ 12-6 แอปพลิเคชันจะทำการบันทึกเวลาที่ใช้ในการเลื่อนเมาส์จากตำแหน่ง Start ถึง END และจำนวนครั้งที่เมาส์ออกนอกเส้นทาง



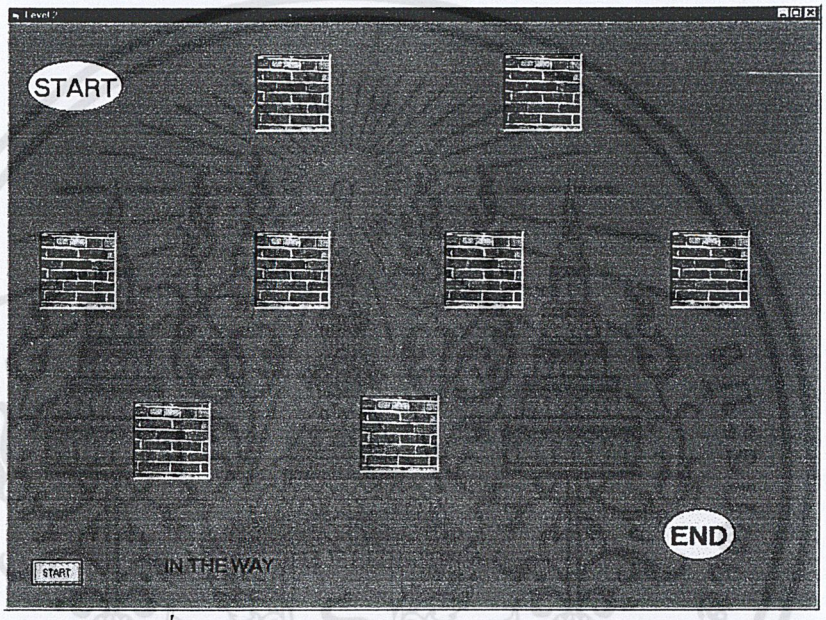
รูปที่12-6 ผลลัพธ์การทดสอบการควบคุมเมาส์ระดับ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12.3.2 การออกแบบการทดสอบการควบคุมเมาส์ระดับ 2

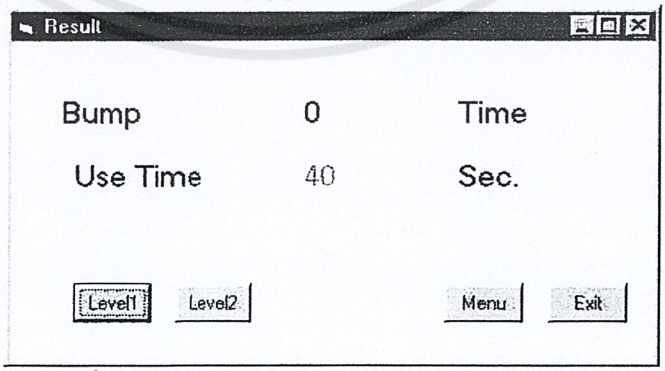
ตามรูปที่ 12-7

1. เริ่มโดนการกดปุ่ม Start
2. จากนั้นเมาส์จะไปอยู่ที่ตำแหน่ง Start (บริเวณพื้นที่สีขาว)
3. ให้เลื่อนเมาส์ไปตำแหน่ง END (บริเวณพื้นที่สีเหลือง) โดยเลื่อนให้อยู่ตามแนวที่กำหนด ไว้เมื่อเมาส์อยู่ในแนวที่กำหนด ด้านล่างของหน้าต่างจะขึ้นว่า "IN THE WAY" เมื่อเมาส์ชนผนังอิฐ ด้านล่างของหน้าต่างจะขึ้นว่า "BUMP"



รูปที่12-7 หน้าต่างการทดสอบการควบคุมเมาส์ระดับ 2

ผลลัพธ์เป็นไปตามรูปที่ 12-8 แอปพลิเคชันจะทำการบันทึกเวลาที่ใช้ในการเลื่อนเมาส์จากตำแหน่ง Start ถึง END และจำนวนครั้งที่เมาส์ชนผนังอิฐ



รูปที่12-8 ผลลัพธ์การทดสอบการควบคุมเมาส์ระดับ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12.3.3 ผลลัพธ์การทดสอบการควบคุมเมาส์

	การทดสอบการควบคุมเมาส์ระดับ 1		การทดสอบการควบคุมเมาส์ระดับ 2	
	OUT THE WAY (ครั้ง)	เวลาที่ใช้ (Sec.)	BUMP (ครั้ง)	เวลาที่ใช้ (Sec.)
เมาส์ปกติ	0	2	0	4
เมาส์ควบคุมด้วยดวงตา	13	84	47	136

ตารางที่12-3 การเปรียบเทียบการทดสอบการควบคุมเมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 13
บทสรุปและวิจารณ์

13.1 ปัญหา

สิ่งแวดล้อม

สิ่งแวดล้อมเป็นปัญหาหลักของการประมวลผลทางรูปภาพ

ปริมาณแสงที่เข้ามาบริเวณดวงตา หากเข้ามามากเกินไปทำให้บริเวณตาดำมีแสงสะท้อนมากและบริเวณผิวหนังรอบดวงตามีแสงตกกระทบมากทำให้ภาพผิวหนังมีความสว่างมาก ดังนั้นภาพผิวหนังที่ได้จึงเป็นสีขาวซึ่งใกล้เคียงกับตาขาว

หากปริมาณแสงที่เข้ามาบริเวณดวงตา หากเข้ามาน้อยเกินไป ภาพจะมืดทำให้ไม่สามารถแบ่งระหว่างตาสีดำและตาสีขาวได้อย่างชัดเจน

อุปกรณ์ในการจับภาพ

ในโครงงานนี้อุปกรณ์ในการจับภาพ นั้นใช้ Web Camera ซึ่ง Web Camera ที่ใช้ไม่สามารถทำการขยายภาพได้ จำเป็นต้องเขียนแอปพลิเคชั่นขึ้นมารองรับการขยายภาพ ทำให้เพิ่มเวลาในการประมวลผลรวมทั้งภาพที่ได้ยังไม่ละเอียดเท่าที่ควร จึงสามารถจับภาพตาดำได้ แต่ไม่สามารถจับภาพละเอียดได้ถึงภาพรูม่านตา ทำให้ขอบเขตการเคลื่อนที่ของตามีข้อจำกัดสูง

ระยะ Web Camera กับดวงตาต้องไม่ห่างจนเกินไปเนื่องจากภาพที่ได้จาก Web Camera มีความละเอียดไม่เพียงพอ

13.2 แนวทางการแก้ไขปัญหาและการพัฒนาต่อ

สิ่งแวดล้อม

เราสามารถเพิ่มอุปกรณ์เสริม คือ อุปกรณ์ยิงแสงอินฟราเรต (Infrared) และ ตัวกรองแสงอินฟราเรตติดไว้บริเวณกล้องวีดีโอหรือ Web Camera เพื่อเป็นการตัดแสงภายนอก

อุปกรณ์ในการจับภาพ

เราสามารถเปลี่ยนจาก Web Camera เป็นกล้องวีดีโอที่มีความสามารถในการขยายภาพจะทำให้ความแม่นยำของเมาส์ควบคุมด้วยดวงตาสูงขึ้น

13.3 ข้อเสนอแนะ

ปัจจุบันสีตาที่ใช้ในการจับภาพเป็นสีดำ หากต้องการใช้กับตาสีอื่นให้เปลี่ยนค่า Threshold ให้ตรงกับตาสีนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13.4 การประยุกต์ใช้

หากเราสามารถเคลื่อนศีรษะได้จะทำให้ขอบเขตการเคลื่อนที่ของเมาส์มีเพิ่มมากขึ้น หรือนำวัตถุอื่นที่ไม่ใช้สีดำ (แนะนำ: วัตถุสีฟ้า หรือ สีเขียว เนื่องจากผิวหนังของมนุษย์มีเม็ดสีสีฟ้าและเขียวน้อย) มาแปะบริเวณหน้าผาก ทำการปรับ Threshold ให้เหมาะสมจะทำให้แอปพลิเคชั่นมีประสิทธิภาพสูงขึ้น

13.5 สรุปผลการทดสอบโปรแกรม

Functional Requirements	ผลการทดสอบ
การเคลื่อนที่ของเคอร์เซอร์เมาส์	สามารถทำได้
คลิกซ้าย	สามารถทำได้
คลิกขวา	สามารถทำได้
ดับเบิลคลิก	สามารถทำได้
ลากและปล่อย	สามารถทำได้

ตารางที่13-1 สรุปผลการทดสอบโปรแกรมตาม *Functional Requirements*

Non-Functional Requirements	ผลการทดสอบ
ความเร็ว	ความเร็วช้ากว่าเมาส์ปกติเนื่องจากมีความแม่นยำไม่สูงมากและใช้เวลาในการคลิกอย่างน้อย 5 วินาที
ความแม่นยำ	ไม่สูงมากเนื่องจากใช้ Web Camera ต้องมีการปรับค่าจำนวนช่องตามแนวแกน x และ y ของจอภาพให้เหมาะสมเพื่อเพิ่มความแม่นยำ
การใช้ง่าย	ใช้ยากกว่าเมาส์ปกติเมื่อเงื่อนไขการใช้มากกว่าและต้องมีการจำกัดระยะระหว่างตากับ Web Camera
การรองรับผู้ใช้	รองรับผู้ใช้ได้มากเนื่องจากใช้การจับตาเพียงข้างเดียวและสามารถปรับค่า Threshold ให้เหมาะกับสีตาต่างๆได้

ตารางที่13-2 สรุปผลการทดสอบโปรแกรมตาม *Non-Functional Requirements*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก โครงสร้างไฟล์ Bitmap

BitMap File

Introduction

BMP files are an historic (but still commonly used) file format for the historic (but still commonly used) operating system called "Windows". BMP images can range from black and white (1 byte per pixel) up to 24 bit colour (16.7 million colours). While the images can be compressed, this is rarely used in practice and won't be discussed in detail here.

Structure

A BMP file consists of either 3 or 4 parts as shown in the diagram on the right. The first part is a header, this is followed by a information section, if the image is indexed colour then the palette follows, and last of all is the pixel data. The position of the image data with respect to the sart of the file is contained in the header. Information such as the image width and height, the type of compression, the number of colours is contained in the information header.

1. File Header

The **BITMAPFILEHEADER** structure contains information about the type, size, and layout of a file that contains a DIB.

5 field 14 bytes

Field Name	Size(Byte)	Description
1.bfYype	2	Contains the characters "BM" that identify the file type
2.bfSize	4	Specifies the size, in bytes, of the bitmap file.
3.bfReservaed1	2	Reserved; must be zero
4. bfReservaed2	2	Reserved; must be zero

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.bfOffbits	4	Specifies the offset, in bytes, from the beginning of the **BITMAPFILEHEADER** structure to the bitmap bits

2. Image Header

The **BITMAPINFOHEADER** structure contains information about the dimensions and color format of a DIB.

11 field 40 bytes

Field Name	Size(Byte)	Description
1.biSize	4	Specifies the number of bytes required by the structure.
2.biWidth	4	Specifies the width of the bitmap, in pixels.
3. biHeight	4	Specifies the height of the bitmap, in pixels. If **biHeight** is positive, the bitmap is a bottom-up DIB and its origin is the lower-left corner. If **biHeight** is negative, the bitmap is a top-down DIB and its origin is the upper-left corner.
4. biplanes	2	Specifies the number of planes for the target device. This value must be set to 1. **biBitCount**
5. biBitCount	2	Specifies the number of bits-per-pixel. The **biBitCount** member of the **BITMAPINFOHEADER** structure determines the number of bits that define each pixel and the maximum number of colors in the bitmap.
6. biCompression	4	Specifies the type of compression for a compressed bottom-up bitmap (top-down DIBs cannot be compressed).
7. biSizeImage	4	Specifies the size, in bytes, of the image. This may be set to zero for BI_RGB bitmaps.
8. biXPelsPerMeter	4	Specifies the horizontal resolution, in pixels-per-meter, of the target device for the bitmap. An application can use this

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		value to select a bitmap from a resource group that best matches the characteristics of the current device.
9. biYPelsPerMeter	4	Specifies the vertical resolution, in pixels-per-meter, of the target device for the bitmap.
10. biClrUsed	4	Specifies the number of color indexes in the color table that are actually used by the bitmap. If this value is zero, the bitmap uses the maximum number of colors corresponding to the value of the **biBitCount** member for the compression mode specified by **biCompression**.
11. biClrImportant	4	Specifies the number of color indexes that are required for displaying the bitmap. If this value is zero, all colors are required.

biBitCount

Specifies the number of bits-per-pixel. The **biBitCount** member of the **BITMAPINFOHEADER** structure determines the number of bits that define each pixel and the maximum number of colors in the bitmap. This member must be one of the following values.

Value	Meaning
0	**Windows 98/Me, Windows 2000/XP:** The number of bits-per-pixel is specified or is implied by the JPEG or PNG format.
1	The bitmap is monochrome, and the **bmiColors** member of BITMAPINFO contains two entries. Each bit in the bitmap array represents a pixel. If the bit is clear, the pixel is displayed with the color of the first entry in the **bmiColors** table; if the bit is set, the pixel has the color of the second entry in the table.
4	The bitmap has a maximum of 16 colors, and the **bmiColors** member of **BITMAPINFO** contains up to 16 entries. Each pixel in the bitmap is represented by a 4-bit index into the color table. For example, if the first byte in the bitmap is 0x1F, the byte represents two pixels. The first pixel contains the color in the second table entry, and the second pixel contains the color in the sixteenth table entry.
8	The bitmap has a maximum of 256 colors, and the **bmiColors** member of

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BITMAPINFO contains up to 256 entries. In this case, each byte in the array represents a single pixel.

16 The bitmap has a maximum of 2^16 colors. If the **biCompression** member of the **BITMAPINFOHEADER** is BI_RGB, the **bmiColors** member of **BITMAPINFO** is NULL. Each **WORD** in the bitmap array represents a single pixel. The relative intensities of red, green, and blue are represented with five bits for each color component. The value for blue is in the least significant five bits, followed by five bits each for green and red. The most significant bit is not used. The **bmiColors** color table is used for optimizing colors used on palette-based devices, and must contain the number of entries specified by the **biClrUsed** member of the **BITMAPINFOHEADER**.

If the **biCompression** member of the **BITMAPINFOHEADER** is BI_BITFIELDS, the **bmiColors** member contains three **DWORD** color masks that specify the red, green, and blue components, respectively, of each pixel. Each **WORD** in the bitmap array represents a single pixel.

Windows NT/Windows 2000/XP: When the **biCompression** member is BI_BITFIELDS, bits set in each **DWORD** mask must be contiguous and should not overlap the bits of another mask. All the bits in the pixel do not have to be used.

Windows 95/98/Me: When the **biCompression** member is BI_BITFIELDS, the system supports only the following 16bpp color masks: A 5-5-5 16-bit image, where the blue mask is 0x001F, the green mask is 0x03E0, and the red mask is 0x7C00; and a 5-6-5 16-bit image, where the blue mask is 0x001F, the green mask is 0x07E0, and the red mask is 0xF800.

24 The bitmap has a maximum of 2^24 colors, and the **bmiColors** member of **BITMAPINFO** is NULL. Each 3-byte triplet in the bitmap array represents the relative intensities of blue, green, and red, respectively, for a pixel. The **bmiColors** color table is used for optimizing colors used on palette-based devices, and must contain the number of entries specified by the **biClrUsed** member of the **BITMAPINFOHEADER**.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

32	The bitmap has a maximum of 2^32 colors. If the **biCompression** member of the **BITMAPINFOHEADER** is BI_RGB, the **bmiColors** member of **BITMAPINFO** is NULL. Each **DWORD** in the bitmap array represents the relative intensities of blue, green, and red, respectively, for a pixel. The high byte in each **DWORD** is not used. The **bmiColors** color table is used for optimizing colors used on palette-based devices, and must contain the number of entries specified by the **biClrUsed** member of the **BITMAPINFOHEADER**. If the **biCompression** member of the **BITMAPINFOHEADER** is BI_BITFIELDS, the **bmiColors** member contains three **DWORD** color masks that specify the red, green, and blue components, respectively, of each pixel. Each **DWORD** in the bitmap array represents a single pixel. **Windows NT/ 2000:** When the **biCompression** member is BI_BITFIELDS, bits set in each **DWORD** mask must be contiguous and should not overlap the bits of another mask. All the bits in the pixel do not need to be used. **Windows 95/98/Me:** When the **biCompression** member is BI_BITFIELDS, the system supports only the following 32-bpp color mask: The blue mask is 0x000000FF, the green mask is 0x0000FF00, and the red mask is 0x00FF0000.

biCompression

Specifies the type of compression for a compressed bottom-up bitmap (top-down DIBs cannot be compressed). This member can be one of the following values.

Value	Description
BI_RGB	An uncompressed format.
BI_RLE8	A run-length encoded (RLE) format for bitmaps with 8 bpp. The compression format is a 2-byte format consisting of a count byte followed by a byte containing a color index. For more information, see Bitmap Compression.
BI_RLE4	An RLE format for bitmaps with 4 bpp. The compression format is a 2-byte format consisting of a count byte followed by two word-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	length color indexes. For more information, see Bitmap Compression.
BI_BITFIELDS	Specifies that the bitmap is not compressed and that the color table consists of three **DWORD** color masks that specify the red, green, and blue components, respectively, of each pixel. This is valid when used with 16- and 32-bpp bitmaps.
BI_JPEG	**Windows 98/Me, Windows 2000/XP:** Indicates that the image is a JPEG image.
BI_PNG	**Windows 98/Me, Windows 2000/XP:** Indicates that the image is a PNG image.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข Component ของ EzVidCap

EzVidCap.ocx

Properties, Methods and Events

Author: *Ray Mercer*

Last Modified: *01/04/98*

Copyright © 1998 – 1999 by Ray Mercer, all rights reserved

Microsoft, Inc copyrights VFWDK and VFW API for Windows.

Properties

AbortLeftMouse Boolean

Set to *True* to allow left mouse button to cancel video capture session

Default value: *True*

AbortRightMouse Boolean

Set to *True* to allow right mouse button to cancel video capture session

Default value: *True*

AutoSize Boolean

Set to *True* to force control to size itself to the video format size

Default value: *True*

BackColor OLE_COLOR

Sets the backcolor of the control

Default value: *&H80000005* (Windows default)

BorderStyle VBCAPBORDERSTYLES

Sets the border style of the control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Default value: *SingleLine* (1)

CancelKey CANCEL_KEY

Sets a keyboard hotkey which will allow user to cancel a capture session from the keyboard

Default value: *KEY_NONE* (0)

CaptureAudio Boolean

Set to *True* to capture audio along with video

Default value: *False*

CaptureFile String

Allows you to set the location and name of the video capture cache file.

*Note** setting this property does not create a file or allocate space. The capture file can be pre-allocated for better performance using the *AllocCapFile* method. Otherwise this file is created, if necessary, when the capture session begins

Default value: *"C:\Capture.avi"*

CaptureRate Long

Set this property to a value between 1 and 100 to request a video capture frame rate from the capture device.

Default value: *15*

Capturing Boolean (Read only)

This property indicates whether a capture session is in progress (*True*)

Default value (when not capturing video): *False*

CaptureViaBackgroundThread Boolean

If this property is *True*, the capture window spawns a separate background thread to perform capture. For example, when you call the *CaptureVideo* function it will return immediately and your code will continue to the next line with no pause.

*Note** Applications that set this flag must handle potential reentry issues because the capture will occur in a separate thread and the application will continue running in the current thread.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Default value: *False*

CenterVideo Boolean

Set to *True* to center the video preview window in the control. This property is useful when *AutoSize* is set to False. Otherwise, it has no effect.

Default value: *True*

Container Object

Standard property

DragIcon StdPicture

Standard property

DragMode Integer

Standard property

DriverIndex DRIVERINDEXES

Set this to the VFW driver index of the capture device you want to connect to. The NumCapDevs method should be used to determine the number of installed devices on a system. The GetDriverName and GetDriverVersion methods should be used to return info about devices at each index before connecting. *Note** the control will automatically attempt to connect to the VFW default index (0) even if this property is not set.

Default value: *VFW_DEFAULT* (0)

FrameEventEnabled Boolean

Set to *True* to tell the control to call the *FrameCallback* event before displaying preview frames. This allows an application to directly access each video frame if desired. The *FrameCallback* event is not raised during streaming video capture.

Default value: *False*

HasAudio Boolean (Read Only)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A value of *True* indicates that there is a WaveIn audio capture device on this system

Default value: *<depends on installed hardware>*

HasDlgDisplay Boolean (Read Only)

A value of *True* indicates that the currently connected VFW capture device will display a standard Display Dialog Box to the user when you call the *ShowDlgVideoDisplay* method. The user can then access and change device specific parameters for this capture device.

Default value: *<depends on installed hardware>*

HasDlgFormat Boolean (Read Only)

A value of *True* indicates that the currently connected VFW capture device will display a standard Format Dialog Box to the user when you call the *ShowDlgVideoFormat* method. The user can then access and change device specific parameters for this capture device.

Default value: *<depends on installed hardware>*

HasDlgSource Boolean (Read Only)

A value of *True* indicates that the currently connected VFW capture device will display a standard Source Dialog Box to the user when you call the *ShowDlgVideoSource* method. The user can then access and change device specific parameters for this capture device.

Default value: *<depends on installed hardware>*

HasOverlay Boolean (Read Only)

A value of *True* indicates that the currently connected VFW capture device supports hardware overlay (allows direct access to the graphics display for improved display speed / lower CPU usage).

Default value: *<depends on installed hardware>*

Height Single

Returns or sets the height of the control

*Note** this property cannot be changed when *AutoSize* is set to *True.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Default value: *<depends on video format size>*

HelpContextID Long

Standard property

hWnd Long (Read Only)

This property is the hWnd of the control's internal CapCaptureWnd. This value may be used with the functions and macros in the Win32 VFW API in place of a CapWnd hWnd.

Default value: *0* until control is initialized, then *hwnd of internal CapWnd*

Index Integer (Read Only)

Standard property

IndexSize Long

Maximum number of index entries in an AVI file. Values range from 1800 to 324,000. If set to 0, a default value of 34,952 (32K frames plus a proportional number of audio buffers) is used.

*Note**Each video frame or buffer of waveform-audio data uses one index entry. The value of this entry establishes a limit for the number of frames or audio buffers that can be captured.

Default value: *27000* (30<fps> * 60<sec> * 15<min>)

Left Single

Standard property

MakeUserConfirmCapture Boolean

If this member is *True*, VFW displays a dialog box prompting the user to initiate capture.

Default value: *True*

Name String

Standard property

Object Object (Read Only)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Standard property

Overlay Boolean

Setting this property to *True* causes VFW to use the hardware overlay features in the installed capture hardware when displaying video in the capture window. The *HasOverlay* property should be checked prior to setting this property to confirm that the hardware actually supports overlay mode.

Default value: *False*

Parent Form (Read Only)

Standard property

PercentDropForError Long

Maximum allowable percentage of dropped frames during capture. Values range from 0 to 100. If the percentage of dropped frames exceeds this value the *Error* event will be raised.

Default value: *10*

Preview Boolean

Enables or disables preview mode.

*Note** Enabling preview mode automatically disables overlay mode.

Default value: *True*

PreviewRate Long

Sets/Returns the frame display rate in preview mode using fps. Valid rates are 1 - 100

*Note** The preview mode uses substantial CPU resources. Applications can disable preview or lower the preview rate when another application has the focus. During streaming video capture, the previewing task is lower priority than writing frames to disk, and preview frames are displayed only if no other buffers are available for writing.

Default value: *15*

StreamMaster CAPSTREAMMASTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Indicates whether the audio stream controls the clock when writing an AVI file. If this member is set to AVSTREAMMASTER_AUDIO, the audio stream is considered the master stream and the video stream duration is forced to match the audio duration. If this member is set to AVSTREAMMASTER_NONE, the durations of audio and video streams can differ.

Default value: *AVSTREAMMASTER_AUDIO* (0)

StretchPreview Boolean

Enables or disables scaling of the preview video images. If scaling is enabled, the captured video frame is stretched to the dimensions of the capture window.

*Note** Scaling preview images controls the immediate presentation of captured frames within the capture window. It has no effect on the size of the frames saved to file. Scaling has no effect when using overlay to display video in the frame buffer.

Default value: *False*

TabIndex Integer

Standard property

TabStop Boolean

Standard property

Tag String

Standard property

TimeLimit Long

Time limit for capture, in seconds. This parameter is ignored if TimeLimitEnabled is *False.*

Default value: *30*

TimeLimitEnabled Boolean

If this member is *True*, streaming capture stops after the number of seconds in the *TimeLimit* property has elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Default value: *False*

UsePreciseCaptureControls Boolean

Setting this property to *True* gives the user precise control over the capture start and stop times. After setting this property, when the user calls the *CaptureStart* method the control will do all necessary preparations before capture, such as allocating memory buffers. After doing this the control will raise the *PreRollComplete* event and wait for the user to call either the *PreciseCaptureStart* method , to start capturing, or the *PreciseCaptureCancel* method. to cancel the capture session. This method is more complicated than simply calling CaptureStart, but it gives the user precise control over the instant that the capture session starts, an it also makes it possible to show a custom user dialog rather than the VFW defined one if desired (see sample application).*Note**

Default value: *False*

VideoBorder VBCAPBORDERSTYLES

Sets the border around the video window to *NONE* or *SINGLE_LINE.*

*Note** This border is separate from the control border

Default value: *SINGLE_LINE*

VideoStreamEventEnabled Boolean

Set to *True* to tell the control to call the *VideoStreamCallback* event before before writing the captured frame to disk. This allows an application to directly access each video frame if desired.

Note* If this property cannot be changed while video is streaming. It can be disabled after capture ends.

Default value: *False*

Visible Boolean

Standard property

WaveStreamEventEnabled Boolean

Set to *True* to tell the control to call the *WaveStreamCallback* event before before writing the audio buffer to disk. This allows an application to directly access each audio buffer if desired.

Note* If this property cannot be changed while video is streaming. It can be disabled after capture ends.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Default value: *False*

WhatsThisHelpID Long

Standard property

Width Single

Returns or sets the width of the control

*Note** this property cannot be changed when *AutoSize* is set to *True*.

Default value: *<depends on video format size>*

YieldEventEnabled Boolean

Set to *True* to tell the control to call the *CaptureYield* event during video capture. This event is guaranteed to be called at least once per video frame captured. If this property is set the *CaptureYield* event will be called regardless of the state of the *CaptureViaBackgroundThread* property.

Note* An application can add a DoEvents or process other tasks in this event during video capture. However extreme care must be taken to handle reentrancy issues which might be caused by accessing the control or related processes during streaming.

Default value: *False*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.Methods

About returns:<none>

Displays an about box with ezVidCap version and copyright information. Takes no parameters.

AllocCapFile returns: Boolean

Creates (preallocates) a capture file of a specified size

Parameters:		
Type	**Name**	**Description**
Long	bytes	Number of bytes to allocate
Returns:		
Type	**Description**	
Boolean	Returns *True* if successful or *False* otherwise. Note* If an error occurs, the *ErrorMessage* event will be raised	

CapSingleFrame returns: Boolean

<takes no parameters>

Retrieves and displays a single frame from the capture driver. After capture, overlay and preview are disabled.

Returns:	
Type	**Description**
Boolean	Returns *True* if successful or *False* otherwise.

CaptureAbort returns: Boolean

<takes no parameters>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Abandons the current capture operation. The capture operation must yield to use this method (set *CaptureViaBackgroundThread* to *True*).

Returns:	
Type	**Description**
Boolean	Returns *True* if successful or *False* otherwise.

CaptureEnd returns: Boolean

<takes no parameters>

Stops the current capture operation. The capture operation must yield to use this method (set *CaptureViaBackgroundThread* to *True*).

Returns:	
Type	**Description**
Boolean	Returns *True* if successful or *False* otherwise.

CaptureVideo returns: Boolean

<takes no parameters>

Initiates streaming video and audio capture to a file.

By default, the capture control does not allow other applications to continue running during capture. To override this, either set the *Yield* property to *True*, or add code to the *Yield Event* in the client application.

During streaming capture, the capture window will issue notifications to your application of specific types of conditions. The *CaptureYield* event will be raised at least once per frame captured if the *YieldEventEnabled* property is set to *True*. The following events may also be raised during capture if the control is either yielding CPU time or running in a background thread:

ErrorMessage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

StatusMessage

Returns:	
Type	**Description**
Boolean	Returns *True* if successful or *False* otherwise.

DiskSpace returns: Long

<takes no parameters>

Returns free space on drive which contains capture file in bytes.

Allocated capture file size is not counted as used diskspace since it will be overwritten by the control during capture. (for example if there is 10000 bytes free and the capture file size is 5000 bytes, this function will return 15000 bytes).

Returns:	
Type	**Description**
Long	Amount of free disk space in bytes + size of capture file

Drag returns: <none>

Standard method

EditCopy returns: Boolean

<takes no parameters>

Copies the contents of the video frame buffer and associated palette to the clipboard.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Returns:	
Type	**Description**
Boolean	Returns *True* if successful or *False* otherwise.

GetDriverName returns: String

Returns the driver description string for the currently selected VFW driver. Or if optional Index parameter is passed returns information for device located at the VFW index specified.

Parameters:		
Type	**Name**	**Description**
Optional Long	Index	VFW index to return info on (if omitted the current device is queried)
Returns:		
Type	**Description**	
String	Driver description string for VFW driver. If there is no driver at the VFW index specified, returns "" (Empty string)	

GetDriverVersion returns: String

Returns version string for the currently selected VFW driver. Or if optional Index parameter is passed returns version string for device located at the VFW index specified.

Parameters:		
Type	**Name**	**Description**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Optional Long	Index	VFW index to return info on (if omitted the current device is queried)
Returns:		
Type	**Description**	
String	Version information string for VFW driver. If there is no driver at the VFW index specified, returns "" (Empty string) Note* Some devices do not return version information. In this case the function will return "" (Empty String).	

LoadPalette returns: Boolean

Loads a new palette from a palette file and passes it to a capture driver. Palette files typically use the filename extension .PAL. A capture driver uses a palette when required by the specified digitized image format

Parameters:		
Type	**Name**	**Description**
String	palFile	Palette file to load from disk.
Returns:		
Type	**Description**	
Boolean	Returns *True* if successful or *False* otherwise. Note* If an error occurs, the *ErrorMessage* event will be raised.	

Move returns: <none>

Standard method

NumDevCaps returns: Long

<takes no parameters>

Returns the number of installed VFW capture devices in the system (VFW spec allows for up to 9 devices).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Returns:	
Type	**Description**
Long	Returns number of video capture devices

PreciseCaptureCancel returns: <none>

<takes no parameters>

Aborts capture session after PreRoll when *UsePreciseCaptureControls* property is set to *True.*

PreciseCaptureEnd returns: <none>

<takes no parameters>

Ends capture session when *UsePreciseCaptureControls* property is set to *True.*

PreciseCaptureStart returns: <none>

<takes no parameters>

Capture session will start at the precise moment this method is called when *UsePreciseCaptureControls* property is set to *True.* Note* This method will have no effect until the *PreRollComplete* event is raised.

SaveAs returns: Boolean

Copies the contents of the capture file to another file.

This message does not change the name or contents of the current capture file.

If the copy operation is unsuccessful due to a disk full error, the destination file is automatically deleted.

Typically, a capture file is preallocated for the largest capture segment anticipated and only a portion of it might be used to capture data. This message copies only the portion of the file containing the capture data.

Parameters:		
Type	**Name**	**Description**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

String	FileName	Destination filename to to copy data to.
Returns:		
Type	**Description**	
Boolean	Returns *True* if successful or *False* otherwise. Note* If an error occurs, the *ErrorMessage* event will be raised.	

SaveDIB returns: Boolean

Copies the current frame to a DIB file.

If the capture driver supplies frames in a compressed format, this call attempts to decompress the frame before writing the file..

Parameters:		
Type	**Name**	**Description**
String	FileName	Destination filename to to copy data to.
Returns:		
Type	**Description**	
Boolean	Returns *True* if successful or *False* otherwise. Note* If an error occurs, the *ErrorMessage* event will be raised.	

SavePalette returns: Boolean

saves the current palette to a palette file.

Parameters:		
Type	**Name**	**Description**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

String	palFile	Destination filename to to copy data to.
Returns:		
Type	**Description**	
Boolean	Returns *True* if successful or *False* otherwise. Note* If an error occurs, the *ErrorMessage* event will be raised.	

SetFocus returns: <none>

Standard method

ShowDlgAudioFormat returns: Boolean

<takes no parameters>

Displays a dialog which allows the user to select an ACM audio format to use during the capture process.

Returns:	
Type	**Description**
Boolean	Returns *False* if no WaveIn audio device is installed on the system (usually means there is no sound card).

ShowDlgCompressionOptions returns: Boolean

<takes no parameters>

Displays a dialog box in which the user can select a compressor to use during the capture process. The list of available compressors can vary with the video format selected in the capture driver's Video Format dialog box.

Software compression is only used with capture drivers that provide frames in the BI_RGB format. This message is most useful in the step capture operation to combine capture and compression in a single

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

operation. Compressing frames with a software compressor as part of a real-time capture operation is most likely too time-consuming to perform.

Compression does not affect the frames copied to the clipboard.

Returns:	
Type	**Description**
Boolean	Returns *True* if successful or *False* otherwise.

ShowDlgVideoDisplay returns: Boolean

<takes no parameters>

Displays a dialog box in which the user can set or adjust the video output. This dialog box might contain controls that affect the hue, contrast, and brightness of the displayed image, as well as key color alignment.

The controls in this dialog box do not affect digitized video data; they affect only the output or redisplay of the video signal.

The Video Display dialog box is unique for each capture driver. Some capture drivers might not support a Video Display dialog box. Applications can determine if the capture driver supports this method by checking the *HasDlgDisplay* property.

Returns:	
Type	**Description**
Boolean	Returns *True* if successful or *False* otherwise.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ShowDlgVideoFormat returns: Boolean

<takes no parameters>

Displays a dialog box in which the user can select the video format. The Video Format dialog box might be used to select image dimensions, bit depth, and hardware compression options.

The Video Format dialog box is unique for each capture driver. Some capture drivers might not support a Video Format dialog box. Applications can determine if the capture driver supports this method by checking the *HasDlgFormat* property.

Returns:	
Type	**Description**
Boolean	Returns *True* if successful or *False* otherwise.

ShowDlgVideoFormat returns: Boolean

<takes no parameters>

Displays a dialog box in which the user can control the video source. The Video Source dialog box might contain controls that select input sources; alter the hue, contrast, brightness of the image; and modify the video quality before digitizing the images into the frame buffer.

The Video Source dialog box is unique for each capture driver. Some capture drivers might not support a Video Source dialog box. Applications can determine if the capture driver supports this method by checking the *HasDlgSource* property.

Returns:	
Type	**Description**
Boolean	Returns *True* if successful or *False* otherwise.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

StreamNoFile returns: Boolean

<takes no parameters>

Initiates streaming video capture without writing data to a file.

This function is useful if you want to access video or audio data in the *VideoStreamCallback* event or *WaveStreamCallback* event without writing data to a file.

By default, the capture control does not allow other applications to continue running during capture. To override this, either set the *Yield* property to *True*, or add code to the *Yield Event* in the client application.

During streaming capture, the capture window will issue notifications to your application of specific types of conditions. The *CaptureYield* event will be raised at least once per frame captured if the *YieldEventEnabled* property is set to *True*. The following events may also be raised during capture if the control is either yielding CPU time or running in a background thread:

ErrorMessage

StatusMessage

Returns:	
Type	**Description**
Boolean	Returns *True* if successful or *False* otherwise.

ZOrder returns: <none>

Standard method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Events

CaptureYield

This event is raised when the capture window yields during streaming capture.

Parameters:	
Type	**Description**
<none>	
Remarks:	
See documentation for *YieldEventEnabled* property	

DragDrop

Standard event.

DragOver

Standard event.

ErrorMessage

This event is raised when the capture window yields during streaming capture.

Parameters:		
Type	**Name**	**Description**
Long	ErrCode	An IDS_CAP* error code (see VFW.bas by Ray Mercer for VB translations of the Microsoft VFW.H)
String	ErrString	A String describing the ErrorMessage
Remarks:		
The ErrString parameter is defined by the VFW system, therefore the strings will vary depending upon the O.S. language version. Use the ErrCode parameter if you need consistency across international Windows platforms.		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FrameCallback

This event is raised when the capture window captures preview frames.

Parameters:		
Type	**Name**	**Description**
Long	lpVHdr	A pointer to a VIDEOHDR type which contains information about the captured frame. (see VFW.bas by Ray Mercer for VB translations of the Microsoft VFW.H)
Remarks:		
This version of the ezVidCap.ocx does not provide guaranteed access to this data before the frame is shown in real time. Development is currently underway for this feature to be implemented in a future version.		

GotFocus

Standard event.

LostFocus

Standard event.

PreRollComplete

If the *UsePreciseCaptureControls* property is set to *True* this event will be raised when the *CaptureVideo()* method is called after all buffers are allocated and preroll activity is complete. After this event is raised, you must either call *PreciseCaptureStart()* or *PreciseCaptureCancel()* to start or cancel a capture session, respectively.

Parameters:		
Type	**Name**	**Description**
<none>		
Remarks:		
This event is not raised unless *UsePreciseCaptureControls* is set to *True* before *CaptureVideo()* is called. See the documentation for *UsePreciseCaptureControls* for more information.		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

StatusMessage

This event is raised when VFW sends a status message to the control. Some status messages include both a numeric code and a native VB string, others do not return a description string.

Parameters:		
Type	**Name**	**Description**
Long	StatCode	An integer which indicates a status change in the control
String	StatString	A String description of the current status (not all status changes fill this parameter – if there is no description string for the code being sent this parameter will be "" – empty string)
Remarks:		
A list of VFW status messages is included in the file VFW.bas by Ray Mercer. This file is freely downloadable from http://i.am/shrinkwrapvb		

VideoStreamCallback

This event is raised during streaming capture when a video buffer is filled.

Parameters:		
Type	**Name**	**Description**
Long	lpVHdr	A pointer to a VIDEOHDR type which contains information about the captured frame. (see VFW.bas by Ray Mercer for VB translations of the Microsoft VFW.H)
Remarks:		
This version of the ezVidCap.ocx does not provide guaranteed access to this data before the frame is written to disk. Development is currently underway for this feature to be implemented in a future version.		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WaveStreamCallback

This event is raised during streaming capture when a new audio buffer becomes available.

Parameters:		
Type	**Name**	**Description**
Long	lpWHdr	Pointer to a WAVEHDR structure containing information about the captured audio data
Remarks:		
This version of the ezVidCap.ocx does not provide guaranteed access to this data before the data is written to disk. Development is currently underway for this feature to be implemented in a future version.		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

หนังสือภาษาไทย

[1] กิตติ ภักดีวัฒนะกุล – จำลอง ครูอุตสาหะ : "Visual Basic 6 ฉบับโปรแกรมเมอร์" , บ.เคทีพี คอมพ์แอนด์ คอนซัลท์ จำกัด ,2521

[2] สัจจะ จรัสรุ่งรวีวร : "คู่มือการเขียนโปรแกรมและใช้งาน Visual Basic 6" , สำนักพิมพ์ อินโฟเพรส ,2544

[3] สุทธิศักดิ์ พงศ์ธนาพานิช : "การเขียนโปรแกรมด้วย Visual Basic 6.0 ระดับสูง (การใช้งานฟังก์ชั่นวินโดวส์ API-32 บิต) " , บ.ไพรเมด จำกัด ,2542

หนังสือภาษาอังกฤษ

[1] Rafael C. Gonzalez and Richard E. Woods : Digital Image Processing , Prentice-Hall, International Edition

แหล่งค้นคว้าทางอินเทอร์เน็ต

[1] http://www.shrinkwrapvb.com/ezvidcap.htm

[2] http://www.shrinkwrapvb.com/avihelp/avihlp_3.htm

[3] http://www.experts-exchange.com/Programming/Programming_Languages/ Visual_Basic

[4] http://www-rcf.usc.edu/~wdutton/comm533/EYEM-WU.htm

[5] http://www.tracksys.co.uk/Pages/EyeTrack/eyemain.htm

[6] http://main.cs.qub.ac.uk/~fmurtagh/eyemouse/Eye_Tracking_Manual.pdf

[7] http://www.videoforwindows.com/articles/docs/art1005.asp

[8] http://www.geocities.com/lev_lv/

[9] http://www.research.linst.ac.uk/eyemouse/eyeintro.htm

[10] http://www.eyegaze.com/doc/ecs.htm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้