สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เมาส์สำหรับคนพิการ

ASSISTIVE MOUSE FOR THE DISABLE PERSON

ทนงเกียรติ พาดี
เศรษฐา ฉัตรไชยเดช

วท.
ท113 ส1
2547
เลขหมู่.....
เลขทะเบียน 61458
วัน,เดือน,ปี 17 ก.ค. 2549
b 11596117
i.....

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมาส์สำหรับคนพิการ

ASSISTIVE MOUSE FOR THE DISABLE PERSON

โดย

ทนงเกียรติ พาดี

เศรษฐา ฉัตรไชยเดช

อาจารย์ที่ปรึกษา

อ. เกียรติกูล เจียรนัยธนะกิจ

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2547

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เมาส์สำหรับคนพิการ

ASSISTIVE MOUSE FOR THE DISABLE PERSON

ผู้จัดทำ

1. นายทนงเกียรติ พาดี รหัสประจำตัว 45015366
2. นายเศรษฐา ฉัตรไชยเดช รหัสประจำตัว 45015384

อาจารย์ที่ปรึกษา

(อ. เกียรติกูล เจียรนัยธนะกิจ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมาส์สำหรับคนพิการ

นายทนงเกียรติ พาดี 45015366
นายเศรษฐา ฉัตรไชยเดช 45015384
อ. เกียรติกูล เจียรนัยธนะกิจ อาจารย์ที่ปรึกษา
ปีการศึกษา 2547

บทคัดย่อ

โครงงานเมาส์สำหรับคนพิการนี้มีความสามารถในการตรวจจับการเคลื่อนไหวของดวงตาได้และตอบสนองโดยการควบคุมการเคลื่อนที่ของเมาส์อย่างถูกต้อง โดยการทำงานจะต้องมีการประมวลผลภาพที่รวดเร็วและสามารถสั่งงานให้เมาส์ click,double,click, drag and drop โดยจะสั่งงานด้วยเสียงพูดผ่าน microphone โดยโปรแกรมสามารถที่จะกำหนดค่าตั้งต้นให้แก่โปรแกรมในกรณีที่มีผู้ใช้งานหลายคนซึ่งแต่ละคนจะมีขนาดและความเข้มของสีตาที่ไม่เท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Assistive Mouse For The Disable Person

Mr. Thanongkeat Padee
Mr. Saittha Chatchaiyadej
Kietikul Jearanaitanakij Advisor
Academic Year 2004

ABSTRACT

Assistive mouse for the disable person can be able to detect the eye movement and control the position of the mouse pointer. The software needs to perform an image processing within a rigid time period to determine a black eye position and then controls the mouse to do corresponding tasks such as Click , Double Click, Drag and Drop . In addition, a human voice is detected through a microphone to accommodate the mouse control. This program can be able to define the initial value for a specific user because of the variety of human eye characteristics.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ ก็คือ อาจารย์ เกียรติกูล เจียรนัยธนะกิจ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา ผู้ที่คอยให้กำลังใจเลี้ยงดู มอบโอกาสที่ดีทางการศึกษามาโดยตลอด ทำให้ข้าพเจ้าได้ทราบถึงองค์ความรู้ต่างๆ ซึ่งหลายคนอาจมิได้โอกาสเช่นนี้ และต้องขอขอบพระคุณอาจารย์ท่านอื่นที่ไม่ได้ออกนาม ที่อยู่ ณ. ภาควิศวกรรมคอมพิวเตอร์นี้ ทุกท่านไม่เคยที่จะบน คอยแต่ให้คำปรึกษาที่ดีเป็นประโยชน์ สุดท้ายนี้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณและขอกราบขอบพระคุณมา ณ ที่นี้

ทนงเกียรติ พาดี

เศรษฐา ฉัตรไชยเดช

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	IX
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ผลที่คาดว่าจะได้รับ	1
1.4 ขอบเขตงานวิจัย	2
1.5 คุณลักษณะของอุปกรณ์ที่ใช้กับเมาส์สำหรับคนพิการ	2
1.6 วิธีการดำเนินงาน	2
บทที่ 2 โครงสร้างรูปภาพและการประมวลผลทางรูปภาพ	3
2.1 โครงสร้างของภาพชนิด Bitmap	3
2.2 Segmentation	4
2.3 เทคนิค Segmentation	4
2.3.1 Discontinuity	4
2.3.1.1 First-Second Derivative (No noise)	4
2.3.1.2 First-Second Derivative (With noise)	5
2.3.1.3 First Derivative (Gradient)	6
2.3.1.4 Laplacian of Gaussian (LoG)	7
2.3.2 Simiailarity	7
2.3.2.1 Thresholding	7
บทที่ 3 การจำกัดสิ่งรบกวน	13
3.1 บทนำ	13
3.2 ขั้นตอนของการกำจัดสิ่งรบกวน	13
3.2.1 การหาขอบเขตในการค้นหาด้านซ้ายของภาพ	15
3.2.3 ข้อจำกัดในการหาขอบเขตการค้นหาของภาพ	15
บทที่ 4 การเขียนโปรแกรมติดต่อกับกล้อง Web Camera	20
4.1 ActiveX Control (VideoOCX และ VideoOCXTools) คือ	20
4.2 method ที่สำคัญ	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้าที่
บทที่ 5 การเขียนโปรแกรมติดต่อ ระหว่างแอปพลิเคชั่น กับวินโดวส์โดยใช้ API	24
5.1 API คือ	24
5.2 ลักษณะของ Event-Driven Programming	25
5.3 ลักษณะโครงสร้างการทำงานของ Windows API	26
5.4 กลุ่มฟังก์ชั่นของ Windows API	29
5.5 การประกาศฟังก์ชั่นใน Windows API	29
บทที่ 6 การเขียนโปรแกรมสั่งงานด้วยเสียงโดยใช้ Microsoft Speech SDK 5.1	31
6.1 Microsoft Speech SDK 5.1 คือ	31
6.2 โครงสร้าง Speech API	31
6.2.1 Text-to-Speech API	32
6.2.2 Recognition Engine	32
6.2.2.1 Shared Recognizer Engine	32
6.2.2.2 InProc Speech Recognition Engine	32
6.3 Speech Recognition interfaces	33
6.4 โครงสร้างที่สำคัญของโปรแกรมในส่วนการสั่งด้วยเสียง	36
บทที่ 7 แนวคิดการออกแบบ	38
7.1 Funtional requirements	38
7.2 Non-functional Requirements	38
7.3 Use Case	39
บทที่ 8 อัลกอริทึมและการแสดงผลตำแหน่งบนหน้าจอภาพ	43
8.1 บทนำ	43
8.2 อัลกอริทึม 1	43
8.3 อัลกอริทึม 2	44
8.4 อัลกอริทึม 3	45
8.5 อัลกอริทึม 4	46
8.6 การแสดงผลตำแหน่งบนหน้าจอภาพ	49
บทที่ 9 การใช้เสียงควบคุมโปรแกรม	50
9.1 บทนำ	50
9.2 คำสั่งควบคุมการทำงานโปรแกรม	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้าที่
บทที่ 10 การทดสอบระบบ	51
10.1 ทดสอบอัตราเร็วของเมาส์	51
10.2 การทดสอบการลากและปล่อยของเมาส์	53
10.3 การทดสอบการควบคุมเมาส์	54
10.4 การออกแบบการทดสอบการควบคุมเมาส์ระดับ 2	55
บทที่ 11 สรุปผลการทดลองและข้อเสนอแนะ	57
11.1 สรุปผลการทดลอง	57
11.2 ข้อเสนอแนะ	57
บรรณานุกรม	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่	หน้าที่
รูปที่ 2-1 แสดงลักษณะการเก็บภาพที่อยู่ในไฟล์ เปรียบเทียบกับภาพที่แสดงจริงโดยใช้โปรแกรมต่างๆ	3
รูปที่2-2 การหาขอบของภาพด้วยวิธี First-Second Derivative (ไม่มี noise)	4
รูปที่2-3 การหาขอบของภาพด้วยวิธี First-Second Derivative (มี noise)(1)	5
รูปที่ 2-4 การหาขอบของภาพด้วยวิธี First-Second Derivative (มี noise)(2)	6
รูปที่ 2-5 การหาขอบของภาพด้วยวิธี First Derivative	6
รูปที่ 2-6 การหาขอบของภาพด้วยวิธี Laplacian of Gaussian (LoG)	7
รูปที่2-7 Histogram แสดงจุด Optimal	7
รูปที่ 2-8 Flow Chart แสดงการหา Optimum Threshold Estimation	8
รูปที่ 2-9 ตัวอย่างภาพขาวดำที่เลือกค่า Threshold=90	9
รูปที่2-10 Flow Chart การทำThresholding Gray Scale	9
รูปที่ 2-11 การแบ่งภาพขาวดำโดยใช้ค่า Threshold ต่างๆ	10
รูปที่ 2-12 การแบ่งภาพสีโดยใช้ค่า Threshold ของสีแดง น้ำเงิน เขียว	10
รูปที่ 2-13 ฟังก์ชั่น ตรวจค่าในช่วงที่สนใจของภาพสี	11
รูปที่ 2-14 การจัดกลุ่ม pixel ที่สีเหมือนกันรวมกลุ่มด้วยกัน	11
รูปที่ 2-15 แสดงการทำ thresholding ภาพลูกตา	12
รูปที่ 3-1 แสดงภาพที่มีสิ่งรบกวน	13
รูปที่ 3-2 ภาพดวงตาที่ผ่านการทำเซ็กเมนต์เตชั่น ของส่วนที่เป็นสีขาวของดวงตา	14
รูปที่ 3-3 ภาพดวงตามองไปทางขวาสุด(ซ้ายของภาพ) จะไม่เกิดปัญหา	16
รูปที่ 3-4 ภาพดวงตาที่ผ่านการทำเซ็กเมนต์เตชั่นในที่ช่วงเป็น สีดำโดยใช้ค้นหาขอบเขตด้านซ้าย	17
รูปที่ 3-5 ภาพดวงตาที่ผ่านการทำเซ็กเมนต์เตชั่นในที่ช่วงเป็น สีขาว โดยใช้ค้นหาขอบเขตด้านขวา	17
รูปที่ 3-6 ภาพดวงตามองไปด้านซ้ายสุด(ขวาสุดของภาพ) จะเกิดปัญหาต้องแก้ไขการใช้ Sliding Window	18
รูปที่ 3-7 ภาพที่ผ่านการทำเซ็กเมนต์เตชั่นในที่ช่วงเป็น สีขาว โดยใช้ค้นหาขอบเขตด้านซ้ายและขวา	19
รูปที่ 4-1 การเก็บข้อมูลของ VideoOCX ใน VC++	23
รูปที่ 5-1 แสดงหลักการทำงานของ Windows API	25
รูปที่ 5-2 แสดงการทำงานของ API Programming ทำงานแบบEvent-Driven Programming	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

หน้าที่

รูปที่

รูปที่ 5-3 แสดงการลำดับขั้นตอนการทำงานของ Windows API Programming 26

รูปที่ 5-4 แสดงขั้นตอนการทำงานของ Windows API Programming ตั้งแต่เริ่มต้นจนจบการทำงาน 27

รูปที่ 5-5 แสดงการทำงานของ Message ชนิดต่างๆ 28

รูปที่ 6-1 แสดงโครงสร้างของ Speech API 31

รูปที่ 7-1 การออกแบบ High-Level 39

รูปที่ 7-1 Use case diagram 39

รูปที่ 7-3 แสดง Sequence Diagram ของการ click left 40

รูปที่ 7-4 แสดง Sequence Diagram ของการ click right 40

รูปที่ 7-5 แสดง Sequence Diagram ของการ double click 41

รูปที่ 7-6 แสดง Sequence Diagram ของการ drag and drop 41

รูปที่ 7-6 แสดง Sequence Diagram ของการ Mouse Move 42

รูปที่ 8-1 แสดงการทำงานตามอัลกอริทึม 1 43

รูปที่ 8-2 แสดงทำงานของ sliding window และการปรับปรุง 44

รูปที่ 8-3 แสดงการทำงานตามอัลกอริทึม 2 44

รูปที่ 8-4 แสดงการเคลื่อนที่ของลูกตาคน 45

รูปที่ 8-5 แสดงการทำงานของกรณี left ลบ right มากกว่า 1 45

รูปที่ 8-6 แสดงการทำงานของกรณี left ลบ right น้อยกว่า -1 46

รูปที่ 8-7 แสดงการทำงานของกรณี top ลบ down มากกว่า 1 46

รูปที่ 8-8 แสดงการทำงานของกรณี top ลบ down น้อยกว่า -1 47

รูปที่ 8-9 แสดงการทำงานตามอัลกอริทึม 4 48

รูปที่ 8-10 แสดงตำแหน่งพอยเตอร์ที่อยู่กึ่งกลางจอภาพ และดวงตาอยู่จุดอ้างอิง 49

รูปที่ 8-11 แสดงตำแหน่งพอยเตอร์ เมื่อมีการเคลื่อนที่ของดวงตาเปรียบกับจุดอ้างอิง 49

รูปที่ 10-1 หน้าต่างการทดสอบอัตราความเร็วของเมาส์ 51

รูปที่ 10-2 ผลลัพธ์การทดสอบอัตราความเร็วของเมาส์ 52

รูปที่ 10-3 หน้าต่างการทดสอบการลากและปล่อยเมาส์ 53

รูปที่ 10-4 หน้าต่างการทดสอบการควบคุมเมาส์ระดับ 1 54

รูปที่ 10-5 ผลลัพธ์การทดสอบการควบคุมเมาส์ระดับ 1 54

รูปที่ 10-6 หน้าต่างการทดสอบการควบคุมเมาส์ระดับ 2 55

รูปที่ 10-7 ผลลัพธ์การทดสอบการควบคุมเมาส์ระดับ 2 56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้าที่

ตารางที่

ตารางที่ 10-1 การเปรียบเทียบอัตราความเร็วของเมาส์ 52

ตารางที่ 10-2 การเปรียบเทียบอัตราการลากและปล่อยของเมาส์ 53

ตารางที่ 10-3 การเปรียบเทียบอัตราการควบคุมเมาส์ระดับ 1 55

ตารางที่ 10-4 การเปรียบเทียบอัตราการควบคุมเมาส์ระดับ 2 56

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

เทคโนโลยีในปัจจุบันมีความเกี่ยวเนื่องกับการดำเนินชีวิตของผู้คนแทบทุกคนเลยก็ว่าได้หนึ่งในนั้นก็มีผู้ใช้อยู่หลายประเภทซึ่งผู้ใช้อีกประเภทหนึ่งที่เราไม่อาจจะมองข้ามได้อันได้แก่ผู้ที่พิการซึ่งบุคคลเหล่านี้ได้รับความยากลำบากเป็นอย่างยิ่งในการใช้งานคอมพิวเตอร์

ดังนั้นแล้วด้วยเหตุนี้จึงได้เป็นแรงบันดาลใจในการพัฒนาโครงงานนี้ขึ้นมาซึ่งก็คือเมาส์สำหรับคนพิการ (Assistive Mouse for the disable person) ซึ่งโครงงานนี้จะช่วยผู้ที่พิการที่มีปัญหาเกี่ยวกับการใช้แขนในการควบคุมเมาส์ได้ยกตัวอย่างเช่นบุคคลที่แขนขาดหรือว่าผู้ที่เป็นอัมพาตท่อนบน คือ ส่วนตั้งแต่กระดูกสันหลังลงมา เป็นต้น ซึ่งการทำงานเมาส์ควบคุมด้วยดวงตา จะประกอบด้วย Web Camera โดยใช้กล้องจับภาพการเคลื่อนไหวของดวงตา แล้วจะนำภาพที่ได้จากกล้องนำมาประมวลผล ซึ่งเรียกว่าการทำ (Image Processing) เพื่อคำนวณหาจุดของดวงตาดำที่เคลื่อนที่ไปมา แล้วนำการเคลื่อนที่นี้ไประบุเป็นตำแหน่งต่างๆบนจอภาพ

1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่อสร้างเมาส์ควบคุมด้วยดวงตาและเสียง ให้คนที่พิการที่ไม่สามารถใช้มือควบคุมเมาส์สามารถใช้ได้โดยใช้สายตาควบคุมแทนมือ ยกตัวอย่างเช่น บุคคลที่แขนขาด หรือว่าผู้ที่เป็นอัมพาตท่อนบน
2. เพื่อศึกษาการเขียนโปรแกรมบน Visual Basic 6.0
3. เพื่อศึกษาการเขียนโปรแกรมเกี่ยวกับการประมวลผลภาพ Image Processing
4. เพื่อศึกษา API ที่ใช้ในการช่วยเขียนโปรแกรมในการติดต่อกับกล้อง (ในที่นี้เราใช้ VideoOCX)
5. เพื่อศึกษา API ที่ใช้ในการช่วยเขียนโปรแกรมรู้จำเสียง (ในที่นี้เราใช้ Microsoft® Speech SDK 5.1)

1.3 ผลที่คาดว่าจะได้รับ

1. ได้รับความรู้ เกี่ยวกับการประมวลผลภาพ Image Processing
2. ได้รับความรู้ ความสามารถในการเขียนโปรแกรมประยุกต์บนระบบปฏิบัติการ Microsoft Windows
3. ได้รับความรู้ ความสามารถในการเขียนโปรแกรมรู้จำเสียง Speech Recognition
4. โปรแกรมสามารถควบคุมเมาส์ สามารถใช้ได้โดยใช้สายตาควบคุม แทนมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขอบเขตของของงานวิจัย

1. โปรแกรมเมาส์สำหรับคนพิการ สามารถที่จะตรวจจับการเคลื่อนไหวของดวงตาได้
2. โปรแกรมเมาส์สำหรับคนพิการ สามารถที่จะบังคับตำแหน่งของเมาส์ให้สอดคล้องกับตำแหน่งที่ตามองไปยังจอภาพได้แม่นยำ
3. โปรแกรมมีการตอบสนองและการทำงานที่รวดเร็ว
4. สามารถสั่งงานให้เมาส์ click ,double click,drag and drop โดยจะสั่งงานด้วยเสียงพูดผ่านไมโครโฟน
5. สามารถกำหนดค่าตั้งต้นให้แก่โปรแกรมในกรณีที่มีผู้ใช้งานหลายคนซึ่งแต่ละคนจะมีขนาดและความเข้มของสีตาที่ไม่เท่ากัน

1.5 คุณลักษณะของอุปกรณ์ที่ใช้กับเมาส์สำหรับคนพิการ

1. กล้องเว็บแคม Web-Camera ที่มีอัตรา Frame rate : 30 frames/sec
2. ไมโครโฟน
3. การ์ดเสียง (soundcard)

1.6 วิธีการดำเนินงาน

งานวิจัยในโครงงานนี้จะเริ่มด้วยการศึกษาทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องกับงานวิจัย ซึ่งก็มีเรื่องหลักๆ ดังนี้ คือ บทที่ 2 โครงสร้างรูปภาพและการประมวลผลทางรูปภาพ , บทที่ 3 การกำจัดสิ่งรบกวน (Noise),บทที่ 4 การเขียนโปรแกรมติดต่อกับกล้อง Web Camera โดยใช้ด้วยโปรแกรมวิชวลเบสิกกับ VideoOCX, บทที่ 5 การเขียนโปรแกรมติดต่อระหว่างแอปพลิเคชั่น กับวินโดวส์โดยใช้ API , บทที่ 6 การเขียนโปรแกรมสั่งงานด้วยเสียงโดยใช้ Microsoft Speech SDK 5.1 โดยใช้ร่วมโปรแกรมวิชวลเบสิก

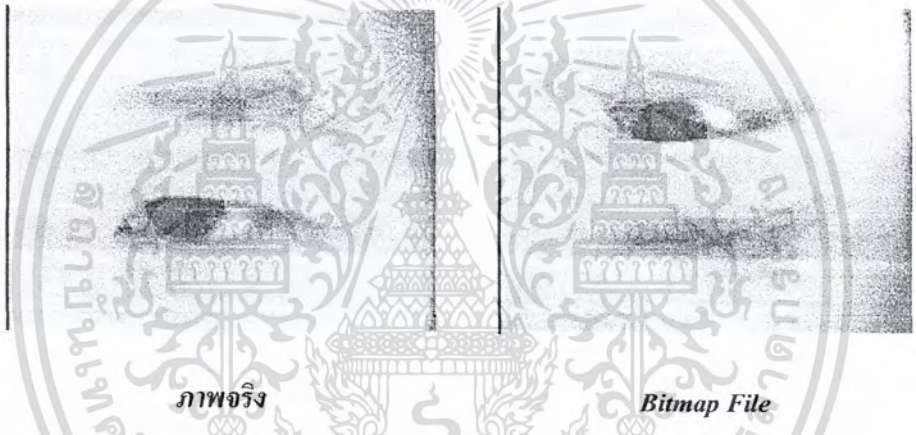
จากนั้นนำข้อมูลต่างที่ได้ศึกษามา มาทำการวางแผนการพัฒนาโครงงาน ดังจะเห็นในบทต่อไปดังนี้ เนื้อหาใน บทที่ 7 แนวคิดและการออกแบบ, บทที่ 8 อัลกอริทึมและการแสดงผลตำแหน่งบนหน้าจอภาพ, บทที่ 9 การใช้เสียงควบคุมการทำงานของโปรแกรม, และได้ทำครบทุกขั้นตอนแล้วก็จะเข้าสู่ขั้นตอน การทดสอบระบบ บทที่ 10 และลำดับสุดท้ายคือ บทที่ 11 สรุปผลการทดลองและข้อเสนอแนะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
โครงสร้างรูปภาพและการประมวลผลทางรูปภาพ

2.1 โครงสร้างของภาพชนิด Bitmap

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



รูปที่ 2-1 แสดงลักษณะการเก็บภาพที่อยู่ในไฟล์เปรียบเทียบกับภาพที่แสดงจริงโดยใช้โปรแกรมต่างๆ

ในการเขียนโปรแกรมวิชวลเบสิก 6.0 เพื่อที่จะดึงภาพชนิด Bitmap ขึ้นมาใช้งานที่จะต้องมีขั้นตอนทำดังต่อไปนี้

1. เปิดไฟล์ที่ต้องการมาพิจารณาชนิดที่เป็น Bitmap

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

โดยทำการเปิดภาพขึ้นมาเป็นชนิดไบนารี่ (binary) โดยดึงชื่อไฟล์มาจากเท็กซ์บ๊อกซ์ที่ชื่อว่า "Text1" มาใส่ตัวแปรชื่อ #1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Segmentation

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

2.3 เทคนิค Segmentation

2.3.1 Discontinuity : การหาขอบของภาพ

2.3.2 Similarity : การหาส่วนที่เหมือนกันของภาพ

2.3.1 Discontinuity

2.3.1.1 First-Second Derivative (No noise)

2.3.1.2 First-Second Derivative (With noise)

2.3.1.3 First Derivative (Gradient)

2.3.1.4 Laplacian of Gaussian (LoG)

2.3.1.1 First-Second Derivative (ไม่มี noise)

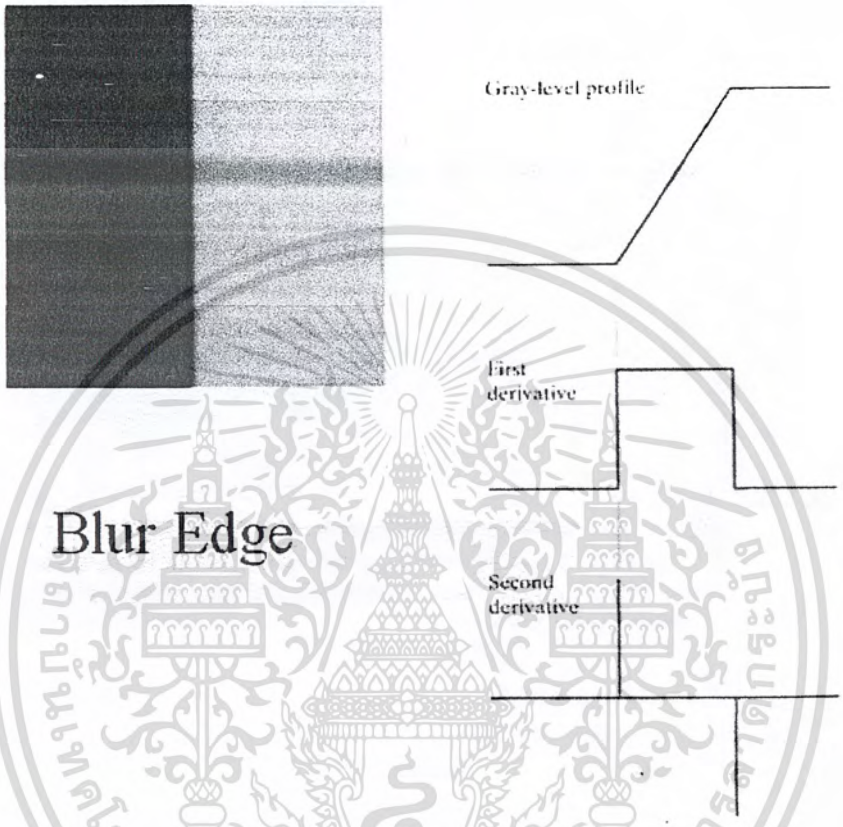
รูปที่2-2 การหาขอบของภาพด้วยวิธี First-Second Derivative (ไม่มี noise)

Sharp Edge เป็นกรณี idealไม่มี Noise มีแบ่งระหว่างความเข้มแสงมากกับความเข้มแสงน้อยอย่างชัดเจน ทำให้สามารถหาขอบได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

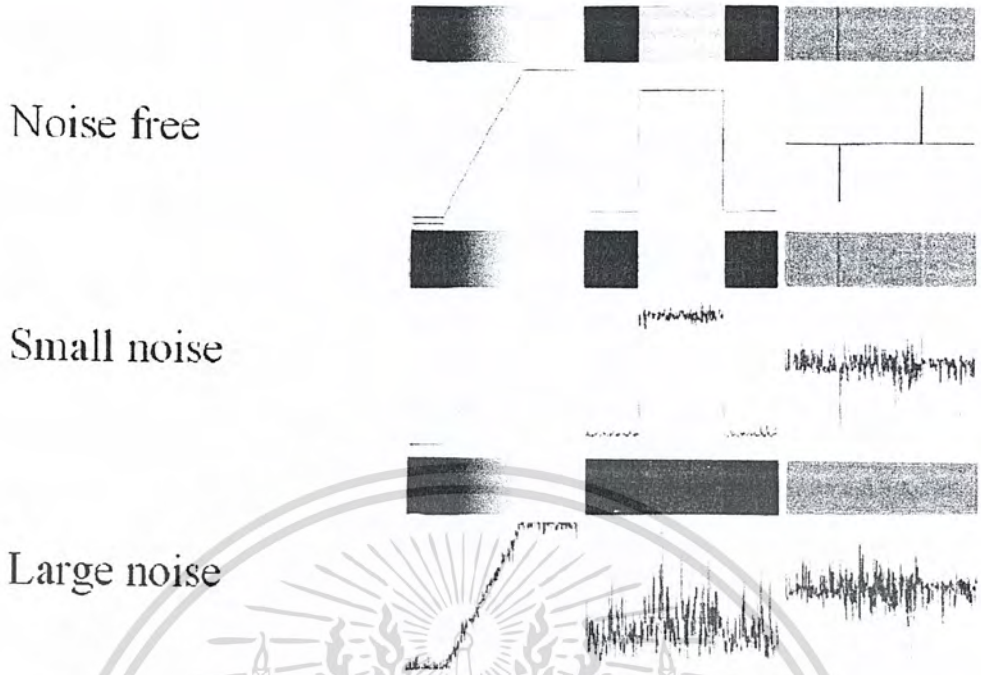
2.3.1.2 First-Second Derivative (มี noise)

ภาพที่เกิดจากการถ่ายภาพปกติ จะมี Noise คือไม่มีการแบ่งระหว่างความเข้มแสงมากกับน้อย อย่างชัดเจน แต่จะเป็นการเปลี่ยนแปลงความเข้มแสงจากมากค่อยๆลดเป็นน้อยหรือกรณีความเข้มแสงจากน้อยค่อยๆเพิ่มเป็นมาก เราสามารถกะขอบได้จากการเปลี่ยนแปลงนี้



รูปที่2-3 การหาขอบของภาพด้วยวิธี First-Second Derivative (มี noise)(1)

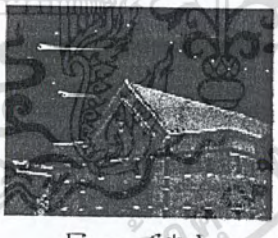
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

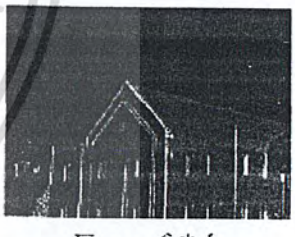


รูปที่ 2-4 การหาขอบของภาพด้วยวิธี First-Second Derivative (มี noise)(2)

2.3.1.3 First Derivative (Gradient)

Original image


$F_x = f * h_x$


$F_y = f * h_y$

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

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

$|F_x| + |F_y|$

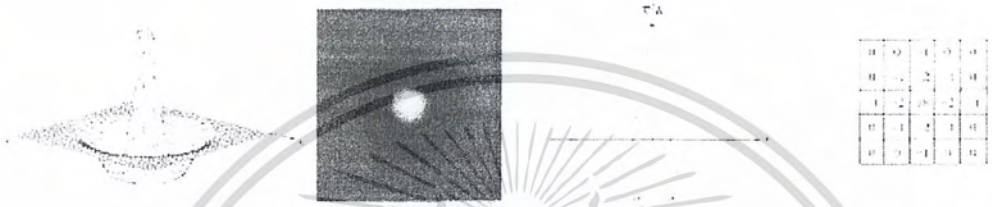
รูปที่ 2-5 การหาขอบของภาพด้วยวิธี First Derivative

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.4 Laplacian of Gaussian (LoG)

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

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



รูปที่ 2-6 การหาขอบของภาพด้วยวิธี Laplacian of Gaussian (LoG)

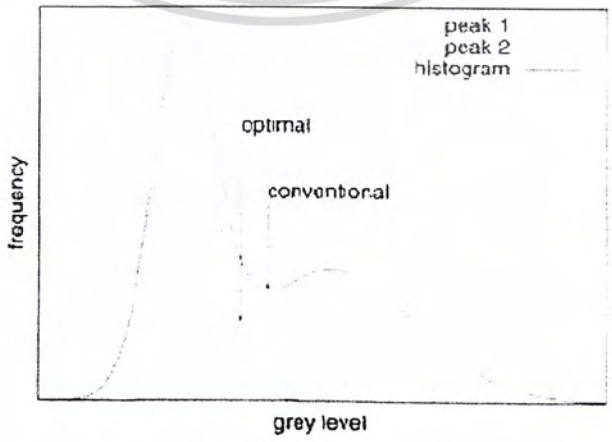
2.3.2 Simiailarity

2.3.2.1 Thresholding จะขอกล่าวถึงแค่ Thresholding เนื่องจากมีการนำมาใช้ในโครงงานวิจัยนี้ด้วย

2.3.2.1 Thresholding

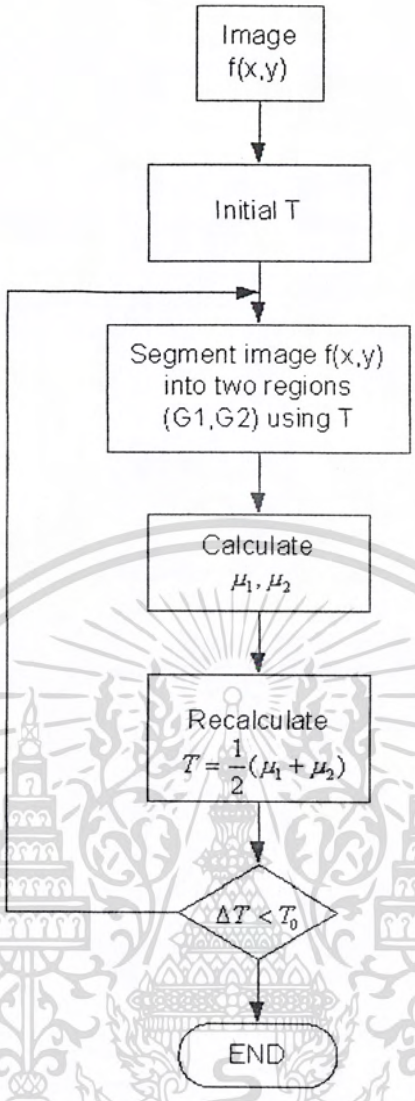
การแบ่งภาพออกเป็นส่วนๆ ส่วนที่สนใจจะถูกแบ่งโดยค่าสีการหาค่า Threshold ทำได้ 2 วิธี

1. มาจากการทดลอง
2. Optimum Threshold Estimation



รูปที่2-7 Histogram แสดงจุด Optimal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-8 Flow Chart แสดงการหา Optimum Threshold Estimation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

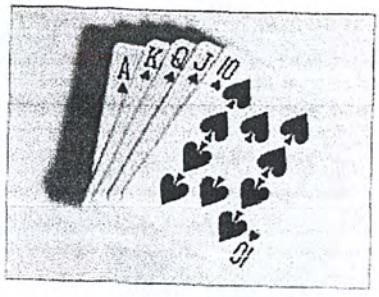
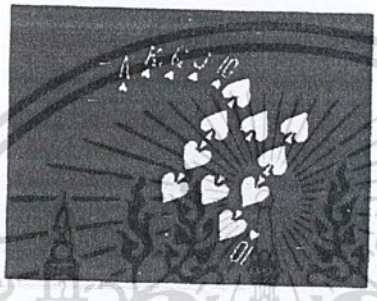
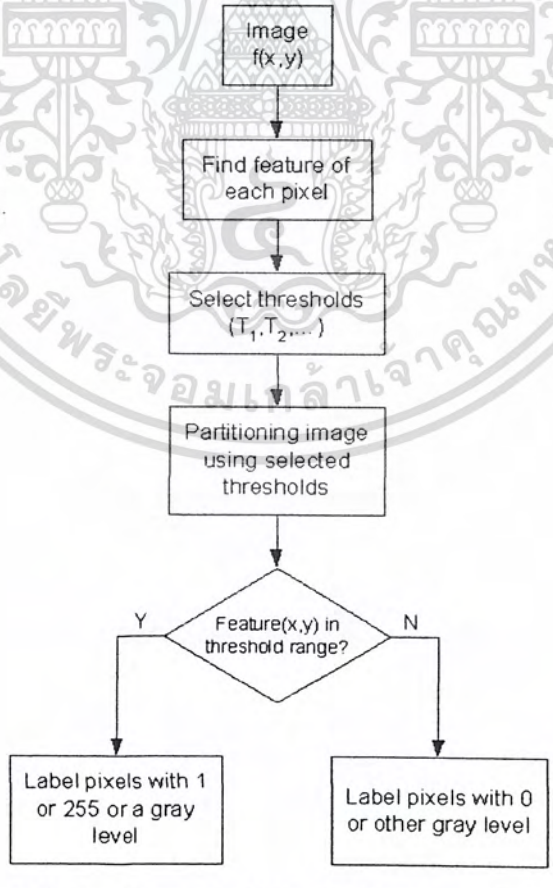


Image
f(x,y)

Thresholding
T = 90

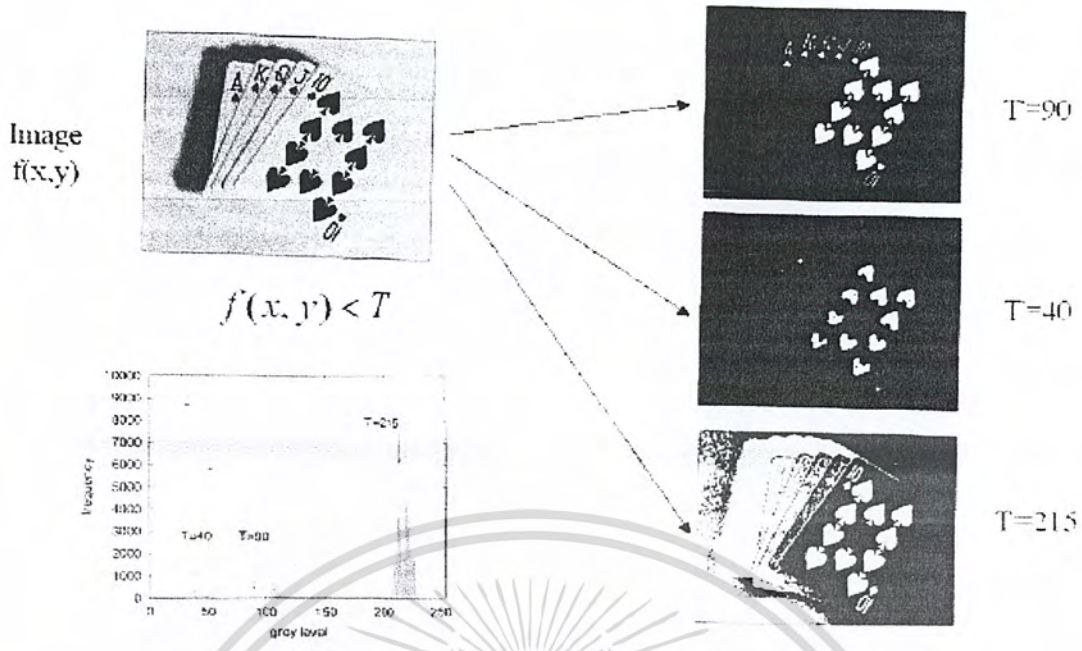


รูปที่ 2-9 ตัวอย่างภาพขาวดำที่เลือกค่า Threshold=90

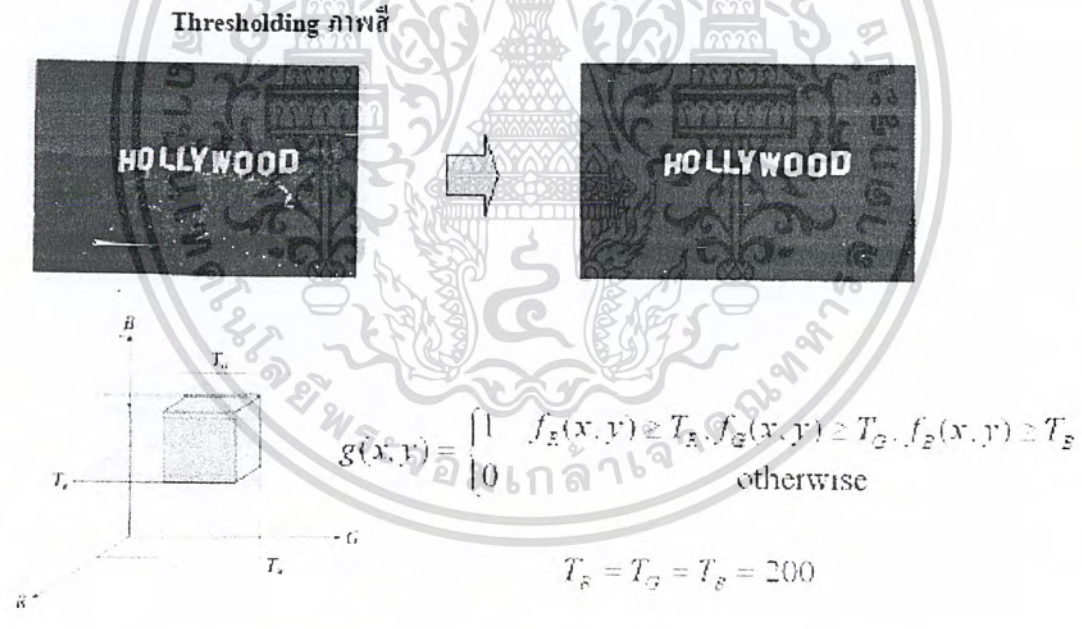


2-10 Flow Chart การทำThresholding Gray Scale

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

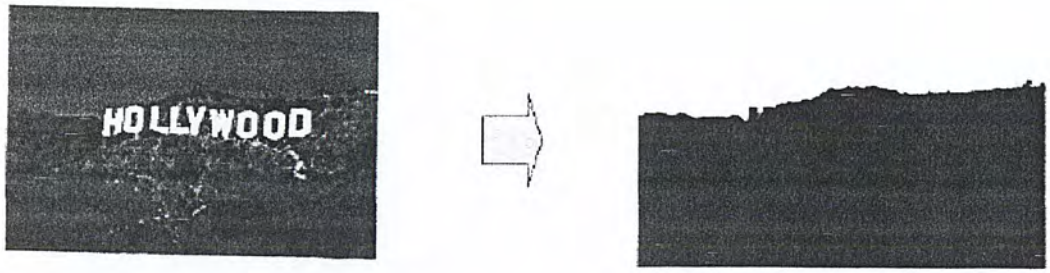


รูปที่ 2-11 การแบ่งภาพขาวดำโดยใช้ค่า Threshold ต่างๆ



รูปที่ 2-12 การแบ่งภาพสีโดยใช้ค่า Threshold ของสีแดง น้ำเงิน เขียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



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

รูปที่ 2-13 ฟังก์ชั่น ตรวจค่าในช่วงที่สนใจของภาพสี

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

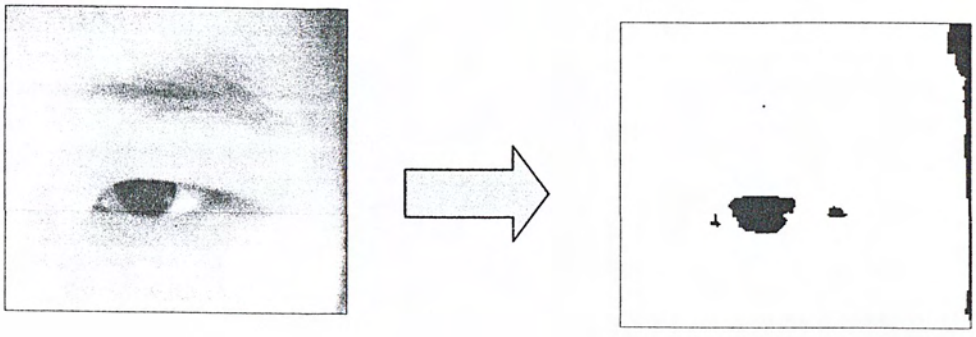
$$d(x,y)=\sqrt{(f_R(x,y)-T_R)^2+(f_G(x,y)-T_G)^2+(f_B(x,y)-T_B)^2}$$

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

รูปที่ 2-14 การจัดกลุ่ม pixel ที่สีเหมือนกันรวมกลุ่มด้วยกัน

ค่า d(x,y) คือค่าที่ควบคุมระยะห่างของช่วงสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-15 แสดงการทำ thresholding ภาพลูกตา

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

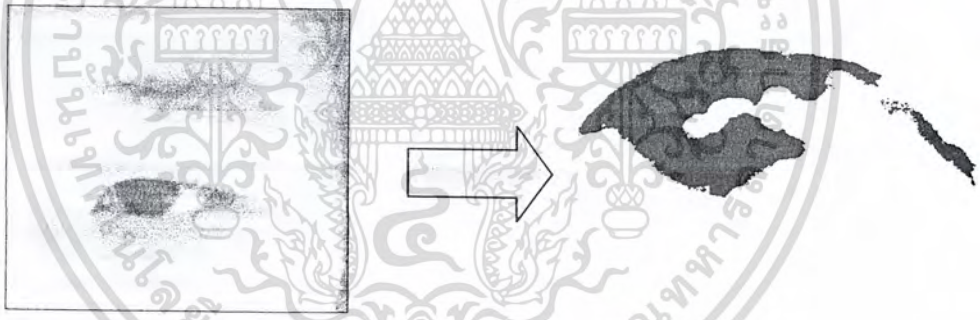
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การจำกัดสิ่งรบกวน

3.1 บทนำ

การหาจุดศูนย์กลางของดวงตานั้นมีสิ่งที่สำคัญในการที่ทำให้เรานั้นสามารถค้นหาจุดศูนย์กลางได้ถูกต้องแม่นยำ คือ การกำจัดสิ่งรบกวน ซึ่งเป็นการตัดส่วนที่เป็นอุปสรรคในการหาจุดศูนย์กลางของดวงตาทำให้ไม่สามารถหาตำแหน่งของดวงตาที่ถูกต้องได้ และสิ่งที่เราจำเป็นต้องกำจัดซึ่งเป็นสิ่งสำคัญที่สุดก็ คือ เส้นผมของผู้ใช้ ซึ่งเส้นผมของคนเอเชียนั้นมีสีเดียวกับตาดำ ทำให้เป็นสิ่งที่ยากยิ่งในการที่จะกำจัดส่วนนั้นออกไป

โดยขั้นตอนการกำจัดสิ่งรบกวนนั้นเราจะทำไปพร้อมๆกับการปรับปรุงการหาจุดศูนย์กลางของตาแบบเดิมที่กล่าวในบทที่ 8 ให้มีประสิทธิภาพสูงขึ้นกว่าเดิม โดยเราเริ่มทดลองจากภาพนิ่งนามสกุล Bitmap ก่อนที่จะนำไปใช้กับภาพเคลื่อนไหวจริง โดยเราสามารถแบ่งส่วนสำคัญของการกำจัดสิ่งรบกวนได้ดังนี้ได้ดังนี้



รูปที่ 3-1 แสดงภาพที่มีสิ่งรบกวน

3.2 ขั้นตอนของการกำจัดสิ่งรบกวน

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 3-2 ภาพดวงตาที่ผ่านการทำเซ็กเมนต์เตชั่น ของส่วนที่เป็นสีขาวของดวงตา

จากรูปภาพที่ 3-2 แสดงให้เห็นที่ภาพผ่านการทำเซ็กเมนต์เตชั่น ของส่วนที่เป็นสีขาวของดวงตา โดยการเปลี่ยนช่วงสีที่เป็นสีขาวให้เป็นสีดำ และสีอื่นให้เป็นสีขาวเพื่อให้ง่ายต่อการพิจารณาโดยขั้นตอนของการกำจัดสิ่งรบกวนนั้นการใช้ตาซ้ายหรือตาขวาจะมีผลช่วยในการกำจัด สิ่งรบกวนได้ง่ายยิ่งขึ้นโดยภาพที่แสดงทั้งหมดต่อไปนี้จะเป็นการกำจัดสิ่งรบกวนของ “ตาซ้าย” โดยหาก ใช้ “ตาขวา” ก็ต้องใช้ขั้นตอนในการกำจัดสิ่งกวนในทิศทางตรงกันข้ามกัน ขั้นตอนการกำจัดสิ่งรบกวนของ “ตาซ้าย” แบ่งออกเป็นขั้นตอนดังต่อไปนี้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 การหาขอบเขตในการค้นหาด้านซ้ายของภาพ

ในการกำจัดสิ่งรบกวนเราจะเริ่มหาขอบเขตในการค้นหาทั้งด้านซ้ายและด้านขวาไปพร้อมๆกัน แต่ในที่นี้จะพูดถึงเฉพาะการค้นหาขอบเขตในการค้นหาเฉพาะด้านซ้ายเท่านั้น

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

การค้นหาด้านซ้ายของภาพนั้นเราจะพิจารณาทั้งจุดที่เป็นสีขาวและสีดำเนื่องจากเราใช้ตาซ้ายเราจะถือว่าด้านซ้ายของภาพจะต้องไม่มีผมอยู่ ดังนั้นจึงสรุปได้ว่าหาเจอสีขาว จะหมายถึง ตาขาว หรือหาเจอสีดำ จะหมายถึง ตาดำหรือคิ้วหรือเงาของหนังตาก็ได้ เนื่องต้องการแค่ช่วงในการค้นหาตาต่อไป คิ้วจึงไม่มีผล (เพราะการค้นหาขอบล่างนั้นเราเริ่มจากล่างขึ้นบนจึงไม่มีปัญหาเรื่องคิ้วหรือเงาของหนังตา) โดยก่อนที่เราจะสรุปว่าเป็นตาขาวจริงนั้นเมื่อเราเจอจุดสีขาวหรือดำจะมีการใช้ Window เล็กที่มี ขนาด 5x5 Pixel เพื่อนับจำนวนจุดว่าจะต้องมีจุดสีขาวมากกว่าที่ตั้งไว้จึงถือว่าเป็นตาขาว เนื่องจากช่วงของสีขาวนั้นกว้างมากอาจมี จุดสีขาวที่เกิดจากเงาของแสงดังนั้นวิธีนี้จึงเป็นวิธีลดข้อบกพร่องลงได้บางส่วน ดังจะแสดงให้เห็นในรูปที่ 3-2 เส้นตามแนวแกน Y เส้นซ้ายมือของภาพ

3.2.2 การหาขอบเขตในการค้นหาด้านขวาของภาพ

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

โดยการค้นหาด้านขวาของภาพนั้นเราจะพิจารณาเฉพาะจุดที่ เป็นสีขาวเท่านั้น เนื่องจากเราใช้ตาซ้ายเพราะฉะนั้นด้านขวาของภาพจะมีเส้นผมที่เราต้องการกำจัดอยู่อยู่ ดังนั้นจะสรุปได้เฉพาะเมื่อเจอสีขาว จะหมายถึง ตาขาว เท่านั้น แต่ถ้าหากเจอสีดำ เราจะไม่รู้สิ่งที่เจอว่าเป็นเส้นผม หรือ ตาดำ โดยในลักษณะที่ตามองด้านซ้ายสุดไม่มีสีขาวด้านซ้ายเลย(จากภาพที่ถ่ายมาจะเป็นด้านขวา) จะกล่าวถึงการแก้ในภายหลัง ในหัวข้อที่ 3.2.3.2

3.2.3 ข้อจำกัดในการหาขอบเขตการค้นหาของภาพ

ข้อจำกัดในการกำหนดขอบเขตของภาพนั้นมีหลายลักษณะเพื่อช่วยในการลดข้อผิดพลาดในการหาขอบเขตการค้นหา จะมีลักษณะดังต่อไปนี้

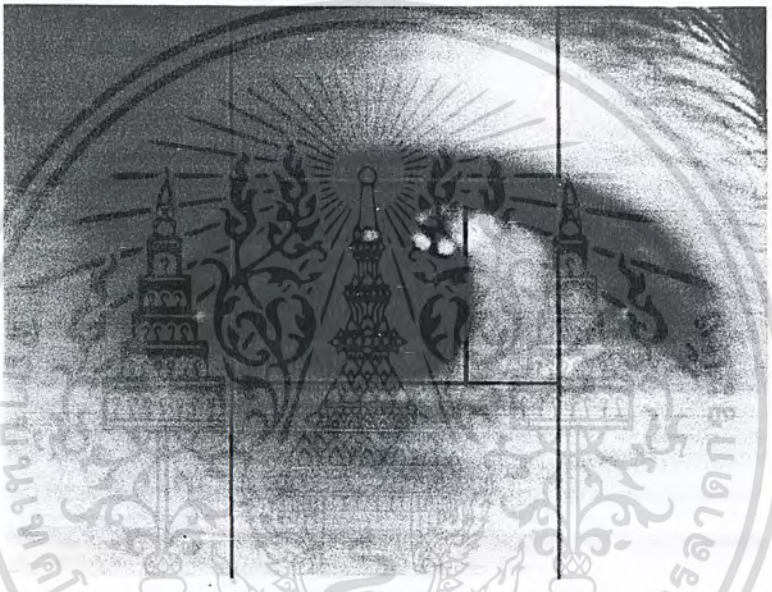
3.2.3.1 ขอบเขตในการค้นหาด้านซ้ายและด้านขวาของภาพจะต้องอยู่แนวเดียวกัน เราสามารถดูได้จากรูปที่ 9-2 จะมีเส้นในแนวนอน 2 เส้น นั้นคือ ขอบเขตในการค้นหาด้านซ้ายและด้านขวาของภาพจะต้องอยู่แนวเดียวกัน ไม่เกิน +/-10 Pixel ของจุดที่ใช้เป็นขอบเขตในการค้นหาด้านซ้ายและด้านขวาของภาพ เพราะว่า จะพิจารณาได้จากส่วนบนขวาของรูปที่ 3-2 จะมีส่วนของเงาที่นำมาพิจารณาเป็นสีขาวด้วย เราจึงแก้โดยใช้วิธีนี้ โดยจะพิจาณากับแนวจุดของแนวที่คาดว่าน่าจะเป็นขอบเขตในการค้นหาที่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยหากด้านไหนเจอก่อนยังไม่สรุปจะรอของอีกด้านหนึ่งแล้วนำมาเปรียบเทียบว่าอยู่แนวเดียวกันหรือไม่ หากไม่ด้านขวาก็จะทำการค้นหาต่อโดยจะไม่ค้นหาแนวที่เคยพบแล้ว แต่จะค้นหาในแนวของด้านซ้ายแทน เนื่องจากเงาส่วนมากจะเกินจากด้านซ้ายของใบหน้าเราเนื่องจากเป็นส่วนโค้งเว้าอยู่ หากด้านซ้ายค้นหาผิดก็ไม่มีปัญหาเนื่องจากด้านซ้ายไม่มีปัญหาเรื่องของเส้นผมอยู่

3.2.3.2 ตามองไปด้านซ้ายสุด(ขวาสุดของภาพ) จะต้องปรับปรุงการกำหนดขอบเขตค้นหาเล็กน้อยและการใช้ Sliding Window

จากรูปที่ 3-3 ภาพควงตามองไปทางขวาสุด(ซ้ายของภาพ)จนไม่มีสีขาวเลยที่ด้านซ้ายเลยของตาดำ จะไม่เกิดปัญหาเนื่องจากการหาขอบเขตในการค้นหาด้านซ้ายของภาพ จะค้นหาจุดได้ทั้งสีขาวและสีดำเนื่องจากไม่มีเส้นผมอยู่จึงไม่เกิดปัญหาขึ้น

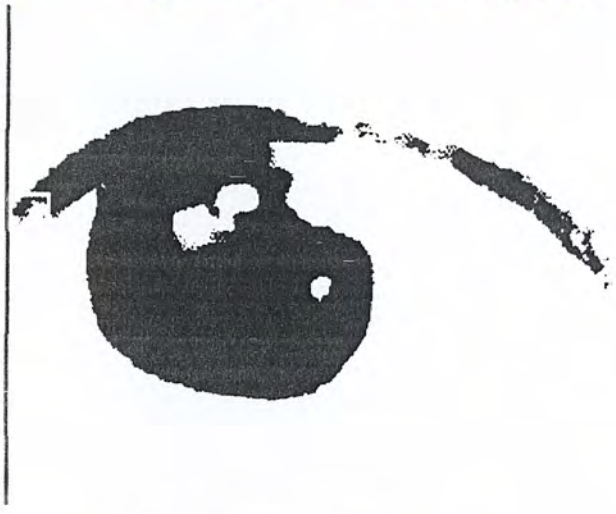


รูปที่ 3-3 ภาพควงตามองไปทางขวาสุด(ซ้ายของภาพ) จะไม่เกิดปัญหา

การกำหนดขอบเขตในการค้นหาด้านซ้ายจะใช้ช่วงของสีดำเป็นตัวกำหนดขอบเขตได้ เนื่องจากถือไม่มีเส้นผมอยู่ ดังจะเห็นได้จากรูปที่ 3-4 จะเป็นการแสดงภาพควงตาที่ผ่านการทำเซ็กเมนต์เตชั่น โดยพิจารณาในที่ช่วงเป็นสีดำ โดยส่วนที่เป็นช่วงสีดำจะแสดงเป็นสีดำ หากเป็นสีอื่นจะเป็นสีขาว เพื่อให้ง่ายต่อการพิจารณา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 3-4 ภาพดวงตาที่ผ่านการทำเซ็กเมนต์เตชั่นในที่ช่วงเป็น "สีดำ" โดยใช้ค้นหาขอบเขตด้านซ้าย

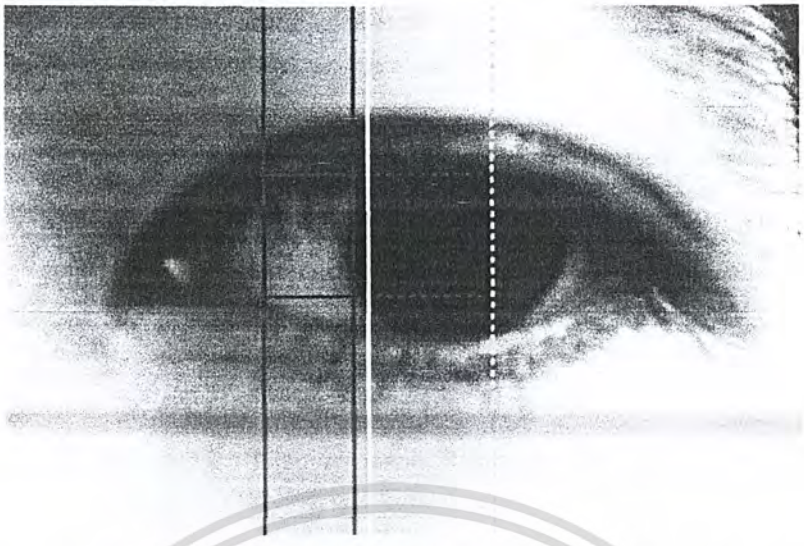
โดยก่อนที่เราจะสรุปว่าเป็นตาดำจริงนั้นเมื่อเราเจอจุดสีดำจะมีการใช้ Window เล็กที่มีขนาด 5x5 Pixel เพื่อนับจำนวนจุดว่าจะต้องมีจุดสีดำมากกว่าที่ตั้งไว้จึงถือว่าเป็นตาดำจริง วิธีนี้จึงเป็นวิธีลดข้อบกพร่องลงได้บางส่วน ดังจะแสดงให้เห็นในรูปที่ 3-4

การกำหนดขอบเขตในการค้นหาด้านขวาจะใช้ช่วงของสีขาวเป็นตัวกำหนดขอบเขตเพียงอย่างเดียว เนื่องจากถือมีเส้นผมอยู่ ดังจะเห็นได้จากรูปที่ 3-5 จะเป็นการแสดงภาพดวงตาที่ผ่านการทำเซ็กเมนต์เตชั่นโดยพิจารณาในที่ช่วงเป็นสีขาว โดยส่วนที่เป็นช่วงสีขาวจะแสดงเป็นสีดำ หากเป็นสีอื่นจะเป็นสีขาว

รูปที่ 3-5 ภาพดวงตาที่ผ่านการทำเซ็กเมนต์เตชั่นในที่ช่วงเป็น "สีขาว" โดยใช้ค้นหาขอบเขตด้านขวา

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
61458



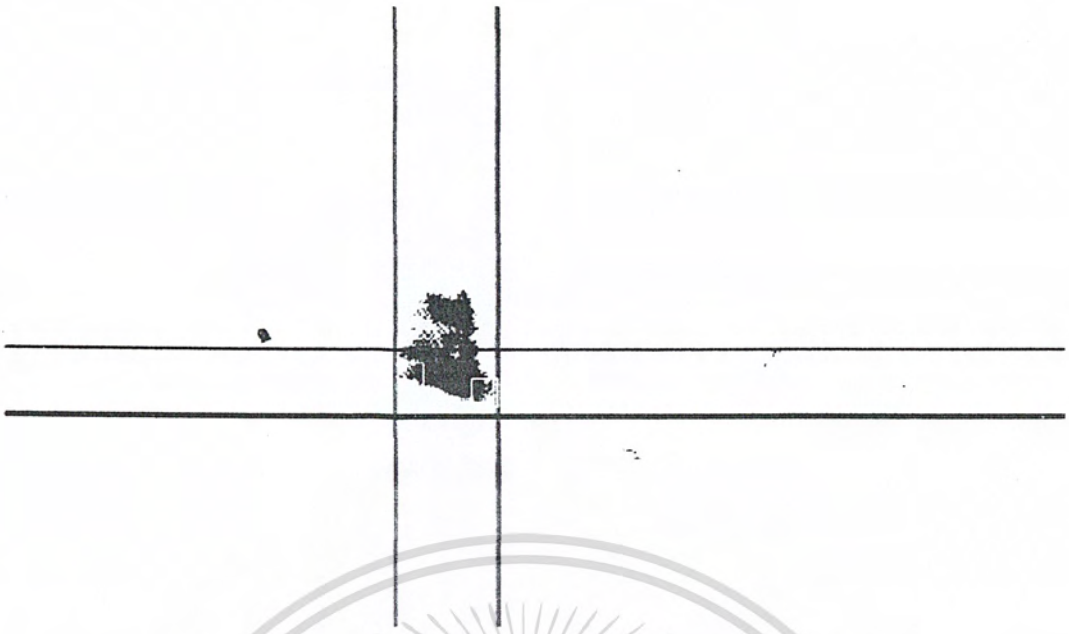
รูปที่ 3-6 ภาพดวงตามองไปด้านซ้ายสุด(ขวาสุดของภาพ) จะเกิดปัญหาต้องแก้ไขการใช้ Sliding Window

ขั้นต้นปรับปรุงขอบเขตในการค้นหาด้านขวาโดยเพิ่มขอบเขตอีก 5 Pixel เพื่อเพิ่มขอบเขตในการหาขอบเขตบนและล่าง เพราะหากไม่เพิ่มอาจจะไม่สามรถค้นหาขอบเขตบนและล่างได้เลย

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 ที่จะสามารถครอบคลุมตาที่ขอบเขตการค้นหาไม่ได้ครอบคลุมไว

โดยภาพที่ 3-7 แสดงให้เห็นที่การค้นหาขอบเขตด้านซ้ายและด้านขวาเพื่อตัวสิ่งรบกวนโดยผ่านการทำเซ็กเมนต์เตชั่น ของส่วนที่เป็นสีขาวของดวงตาโดยการเปลี่ยนช่วงสีที่เป็นสีขาวให้เป็นสีดำและสีอื่นให้เป็นสีขาวเพื่อให้ง่ายต่อการพิจารณา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-7 ภาพที่ผ่านการทำเซ็กเมนต์เตชั่นในที่ช่วงเป็น "สีขาว" โดยใช้ค้นหาขอบเขตด้านซ้ายและขวา

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การเขียนโปรแกรมติดต่อกับกล้อง Web Camera

ปัจจุบันกล้อง เว็บแคม ถือว่าเป็นอุปกรณ์หนึ่งที่ได้รับความนิยมใช้งานเป็นจำนวนมาก ดังนั้นแล้วเพื่อตอบสนองถึงความนิยมนี้ ถึงมีผู้พัฒนาเครื่องมือต่างๆที่ช่วยในการเขียนโปรแกรมที่ใช้ในการติดต่อกับกล้องออกมามาก ซึ่งในโครงงานนี้ได้เลือกใช้ ตัวเครื่องมือที่เรียกว่า VideoOCX

4.1 ActiveX Control (VideoOCX และ VideoOCXTools) คือ

VideoOCX เป็นคอนโทรลที่ช่วยให้การเขียนโปรแกรม Visual C++ และ Visual Basic ในการติดต่อกับกล้อง Webcam สามารถทำได้โดยง่ายเพียงกำหนดค่าตัวแปรให้กับ VideoOCX และ เรียกใช้งานผ่าน method โดยโหมดการทำงานของ VideoOCX จะมีอยู่ 2 โหมด คือ โหมด 0 (Video) และ โหมด 1 (AVI) ตัวอย่างของ method ที่สำคัญได้แก่

4.2 method ที่สำคัญ

Syntax : long *object*.**Capture**(long *imagehandle*)

Description : capture the current video image

Parameter : current video image is copied into the handle

Return Type : address to image data array

Syntax : BOOL *object*.**ToGray**(long *colorimagehandle*, long *grayimagehandle*)

Description : convert a color image to rayscale

Parameter : color image handle and grayscale image handal

Return Type : true is successful

Syntax : VARIANT *object*.**GetMatrix**(long *imagehandle*)

Description : convert a color image to rayscale

Parameter : handle to the image

Return Type : 2 or 3 dimensional matrix cotaining to image pixel

Syntax : VOID *object*.**Show**(long *imagehandle*)

Description : display any image in the control

Parameter : handle to the image

Return Type : none

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Syntax : BOOL *object*.**ShowDriverDlg()**

Description : open dialog that enables the user to choose a video driver

Parameter : none

Return Type : true is successful

Syntax : VOID *object*.**SetErrorMessages**(BOOL *flag*)

Description : use will display every error that occurs

Parameter : true to turn error message on

Return Type : none

Syntax : BOOL *object*.**Init**()

Description : initialize VideoOCX and connect to the specified image source

Parameter : none

Return Type : true is successful

Syntax : LPCTSTR *object*.**GetLastErrorString()**

Description : return the message of the last occurred error

Parameter : none

Return Type : error message

Syntax : long *object*.**GetColorImageHandle**()

Description : recive an empty color image

Parameter : none

Return Type : handle to the new allocated image

Syntax : long *object*.**GetGrayImageHandle**()

Description : recive an empty grayscale image

Parameter : none

Return Type : handle to the new allocated image

Syntax : BOOL *object*.**Start**()

Description : initiate the background capture process

Parameter : none

Return Type : true is successful

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Syntax : BOOL *object*.**Stop**()

Description : end the internal capturing process

Parameter : none

Return Type : true is successful

Syntax : VOID *object*.**ReleaseImageHandle**(long *imagehandle*)

Description : release the image memory

Parameter : handle to an image

Return Type : none

Syntax : BOOL *object*.**Close**()

Description : close the current connection

Parameter : none

Return Type : true is successful

Syntax : VOID *object*.**ReleaseMatrixToImageHandle**(long *imagehandle*)

Description : release memory for the last matrix

Parameter : handle to an image

Return Type : none

Syntax : long *object*.**GetImageDataSize**(long *imagehandle*)

Description : retrieve the number of pixels

Parameter : handle to an image

Return Type : number of bytes

Syntax : long *object*.**GetDataPointer**(long *imagehandle*)

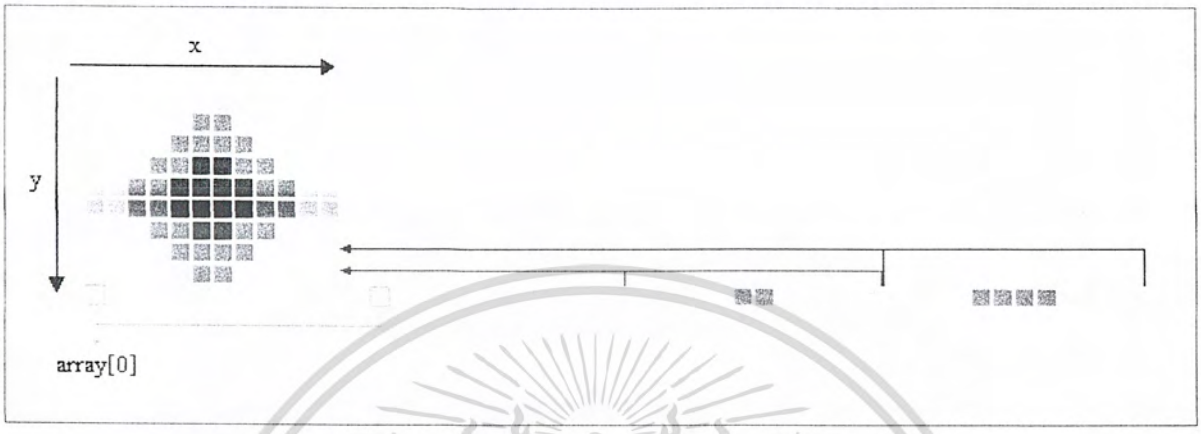
Description : get a pointer to the image data

Parameter : image handle

Return Type : address of image data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วน VideoOCXTools จะเป็นคอนโทรลที่ช่วยในการจัดการ image processing ร่วมกับ VideoOCX ซึ่งจะต้องใช้การ capture ภาพเข้ามาทีละเฟรมซึ่งในขั้นตอนนี้หากเราไม่ต้องการใช้งาน VideoOCXTools เราก็สามารถเขียนโปรแกรมเพื่อทำงานเกี่ยวกับ image processing เองก็สามารถทำได้โดยเรียกผ่าน GetDataPointer Method ของ VideoOCX ซึ่งรูปแบบการเก็บข้อมูลจะมีลักษณะดังนี้



รูปที่ 4-1 การเก็บข้อมูลของ VideoOCX ใน VC++

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การเขียนโปรแกรมติดต่อ
ระหว่างแอปพลิเคชั่น กับวินโดวส์โดยใช้ API

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

5.1 API คือ

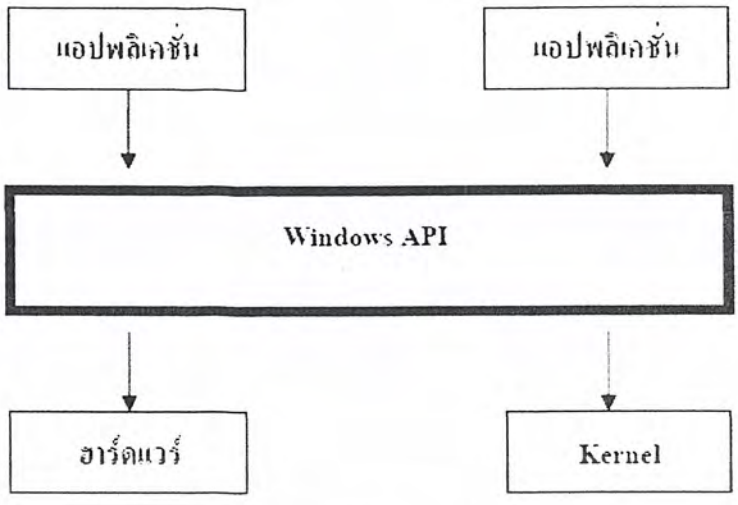
API ย่อมาจาก (Application Programming Interface) ซึ่งเป็นส่วนเชื่อมต่อ ระหว่างแอปพลิเคชั่น กับ วินโดวส์โดยจะแสดงลักษณะการติดต่อสื่อสารกันระหว่างแอปพลิเคชั่น กับวินโดวส์ ทำให้เข้าใจได้ว่าแอพพลิเคชั่น กับวินโดวส์มีการติดต่อสื่อสารกันอย่างไร โดยมีลักษณะการทำงานเป็นแบบ Event-Driven Programming ซึ่งมีลักษณะการทำงานที่ทำให้ผู้ใช้เป็นอิสระแตกต่างจากแบบเดิมที่เป็น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

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 ได้ แก่ ความเป็นมาตรฐานที่ทุกคนรู้จักจึงง่ายต่อการแก้ไขเปลี่ยนแปลง เช่น แม้ว่าเราจะเปลี่ยนไปใช้ฮาร์ดแวร์ใหม่หรือแม้แต่ระบบปฏิบัติ การเวอร์ชันใหม่ๆแล้วก็ตาม ชื่อของฟังก์ชั่ นเหล่านี้ ยังคงใช้ชื่อเดิมทำงานได้เหมือนเดิมทำให้เราไม่ต้องกังวลกับความเปลี่ยนแปลงที่จะเกิดขึ้นจนบ่อยเกินไปนักซึ่งมันก็คือ ชุดของฟังก์ชั่นที่พร้อมให้เราเรียกใช้งาน โดยที่ฟังก์ชั่นเหล่านั้นรับประกันได้ว่าทำงานอย่างมีประสิทธิภาพ เช่น เราไม่จำเป็นต้องเขียนโปรแกรมให้ยืดยาวเพื่อตรวจสอบว่าเนื้อที่ฮาร์ดดิสก์เหลือว่างอยู่เท่าไร แต่เรียกใช้ฟังก์ชั่น GetDiskFreeSpace ฟังก์ชั่นเดียวก็รู้แล้ว เป็นต้นซึ่งในรูปที่ 5-1 จะแสดงหลักการทำงานของ Windows API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

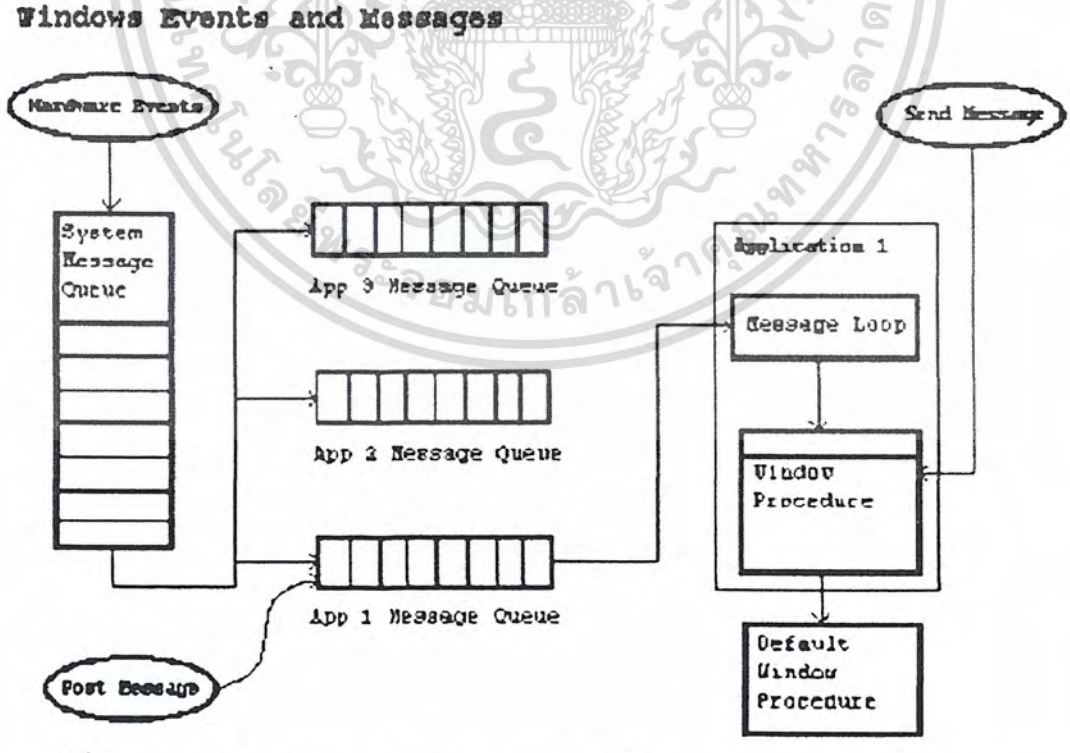


รูปที่ 5-1 แสดงหลักการทำงานของ Windows API

โดยในขั้นต้นเรามาทำความเข้าใจเกี่ยวกับการทำงานแบบ Event-Driven Programming กันก่อน

5.2 ลักษณะของ Event-Driven Programming

มีลักษณะการทำงานที่ไม่เป็นลูกโซ่ที่ต้องทำตามลำดับ โดยโปรแกรมจะทำงานตามเหตุการณ์ที่ผู้ใช้กระทำ โดยที่โปรแกรมจะไม่เป็นตัวชักจูงทำตามผู้ใช้ให้ทำตามที่โปรแกรมตั้งไว้ โดยที่ระบบปฏิบัติการจะเป็นผู้ตรวจสอบเหตุการณ์ที่เกิดขึ้นแล้วส่งข้อมูลของเหตุการณ์นั้น(Message) ไปให้ยังโปรแกรม โดยที่โปรแกรมจะทำงานตามข้อมูลของเหตุการณ์นั้น ดังจะแสดงให้เห็นในรูปที่ 5-2



รูปที่ 5-2 แสดงการทำงานของ API Programming โดยทำงานแบบ Event-Driven Programming

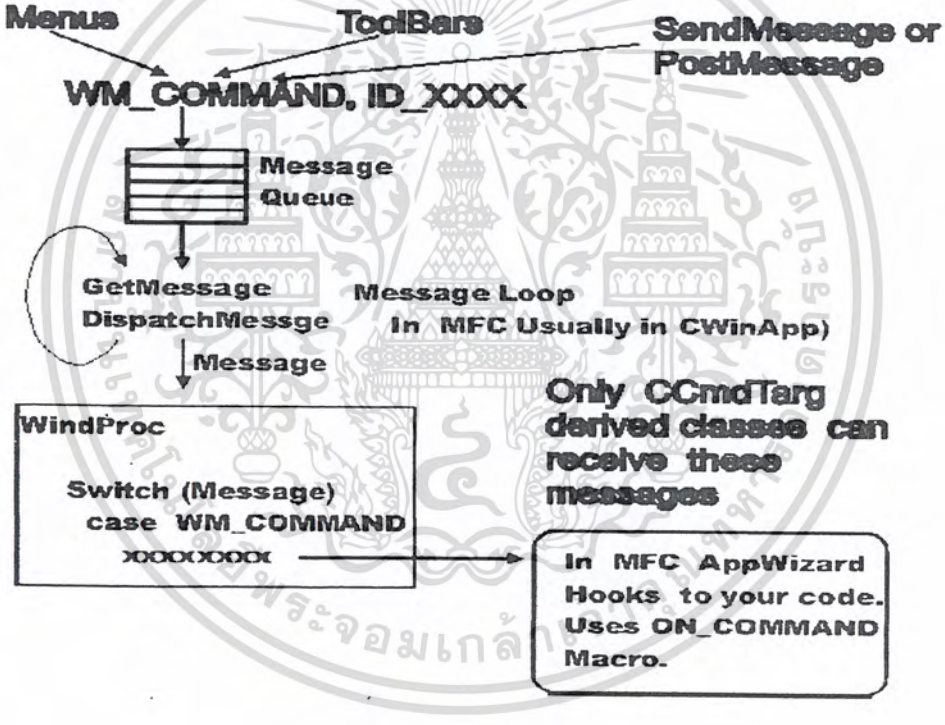
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 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 จะจบการทำงานและคืนค่าการควบคุมกลับไปยัง ระบบปฏิบัติการ
   - ทำ Message เป็นคำสั่งที่มาจากผู้ใช้ ก็จะทำตามข้อมูลที่ส่งมานั้นๆ

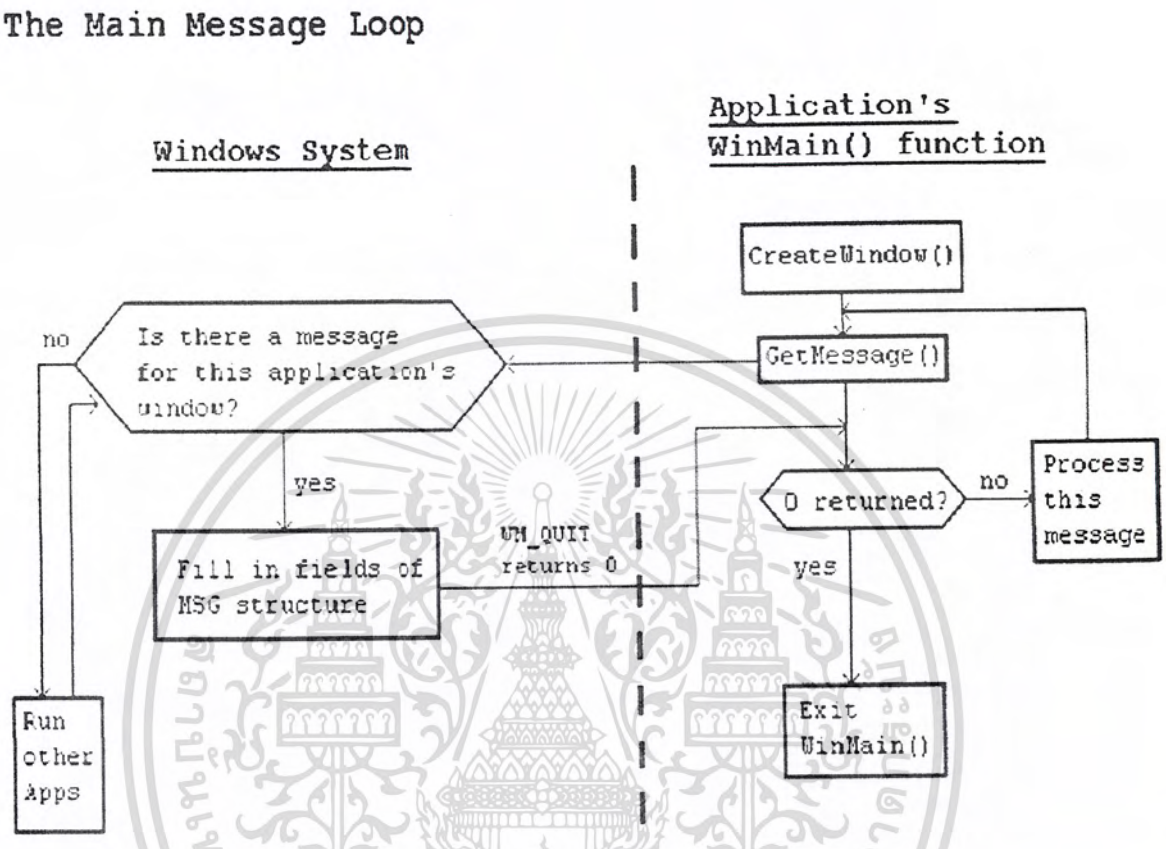
   โดยขั้นตอนการทำงานจะแสดงในรูปที่ 5-3 และ 5-4



รูปที่ 5-3 แสดงการลำดับขั้นตอนการทำงานของ Windows API Programming

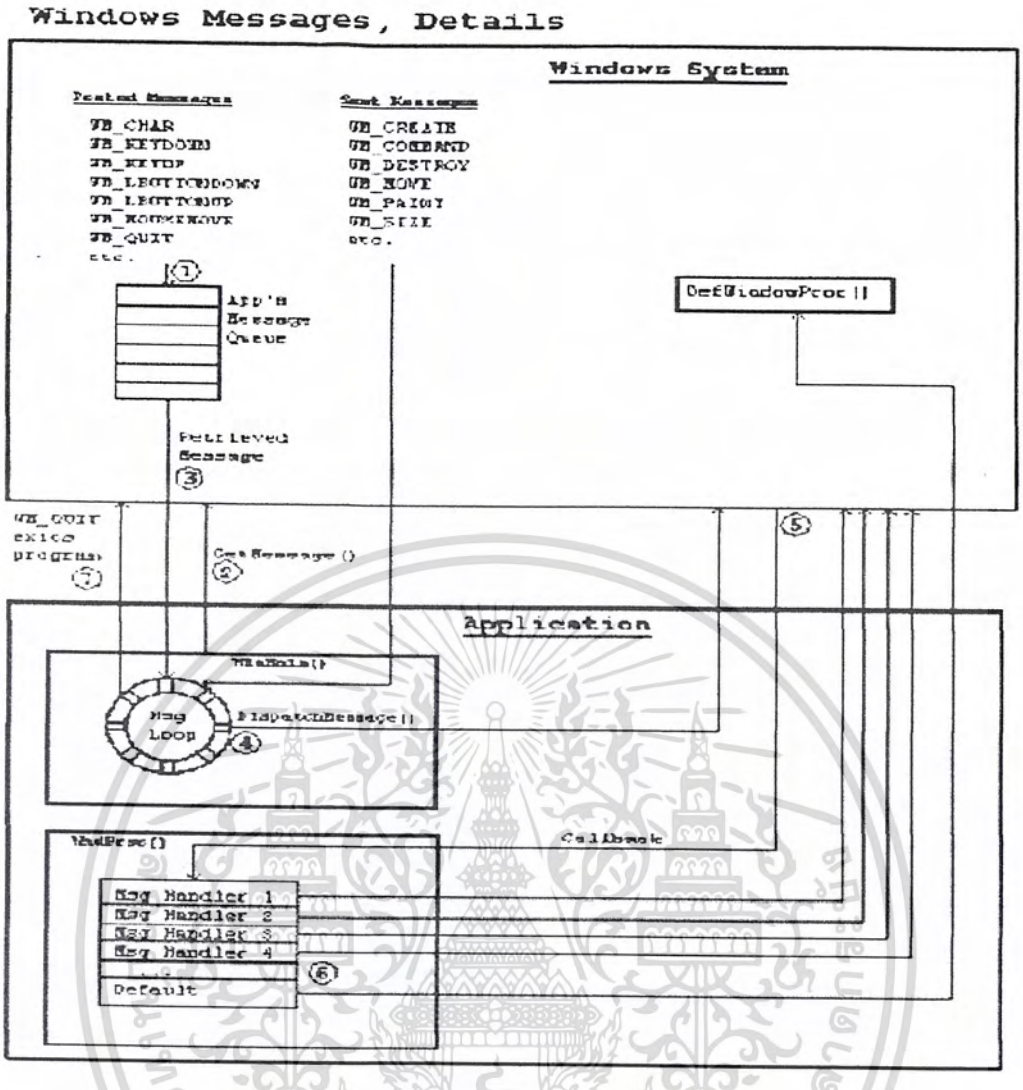
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยรูปที่ 5-4 จะแสดงขั้นตอนการทำงานของโปรแกรมจนจบการทำงาน โดยจะจบการทำงานเมื่อได้รับ Message "WM_QUIT"



รูปที่ 5-4 แสดงขั้นตอนการทำงานของ Windows API Programming ตั้งแต่เริ่มต้นจนจบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-5 แสดงการทำงานของ Message ชนิดต่างๆ

ในหัวข้อนี้เราจะเรียนรู้วิธีการเรียกใช้ฟังก์ชั่น API พร้อมทั้งบอกถึงฟังก์ชั่นที่ถูกนำมาใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 กลุ่มฟังก์ชั่นของ Windows API

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

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 เก็บฟังก์ชั่นที่เกี่ยากับ Multiple Provide Router

NETAPI32.DLL เก็บฟังก์ชั่นที่เกี่ยวกับการจัดการเครือข่าย (Network)

SHELL32.DLL เก็บฟังก์ชั่นที่เกี่ยวกับเชลล์ (Windows Explorer)

USER32.DLL เก็บฟังก์ชั่นที่เกี่ยวกับการติดต่อกับผู้ใช้งาน (User Interface)

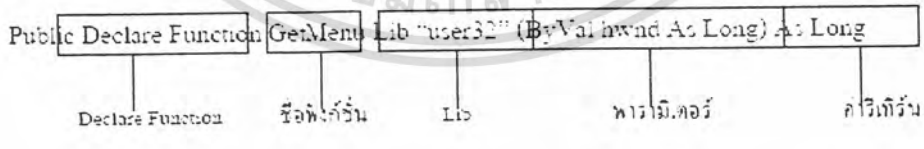
VERSION32.DLL เก็บฟังก์ชั่นที่เกี่ยวกับการจัดการรุ่นของซอฟต์แวร์

WINMM.DLL เก็บฟังก์ชั่นที่เกี่ยวกับมัลติมีเดีย

WINSPOLL.DRV เก็บฟังก์ชั่นที่เกี่ยวกับการพิมพ์ผ่านเครื่องพิมพ์

5.5 การประกาศฟังก์ชั่นใน Windows API

ในการเรียกใช้ฟังก์ชั่นต่างๆ ของ Windows API นั้น มีขั้นตอนที่ แตกต่างจากการใช้เรียกฟังก์ชั่นที่วิชวลเบสิกเตรียมไว้ให้ หรือฟังก์ชั่นที่เราสร้างขึ้นมาเองนั่นคือ ต้องมีการประกาศค่าฟังก์ชั่นก่อนการใช้งานสำหรั บการใช้งานฟังก์ชั่ นนั้น เราจะคัดลอกมาจาก API Viewer ซึ่งฟังก์ชั่นที่คัดลอกมานั้น มีรูปแบบการประกาศดังนี้



จากตัวอย่าง เป็นการประกาศฟังก์ชั่น GetMenu และฟังก์ชั่น RegCloseKey จะเห็นว่าฟังก์ชั่นที่ประกาศนั้น ประกอบด้วยองค์ประกอบต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

• 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 Longในบางฟังก์ชั่นนั้น อาจมีองค์ประกอบอื่นเพิ่มเติม เช่น

```
Public Declare Function RegisterClass Lib "user32" Alias "RegisterClassA" (Class As WNDCLASS) As Long
                                                    Alias
```

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

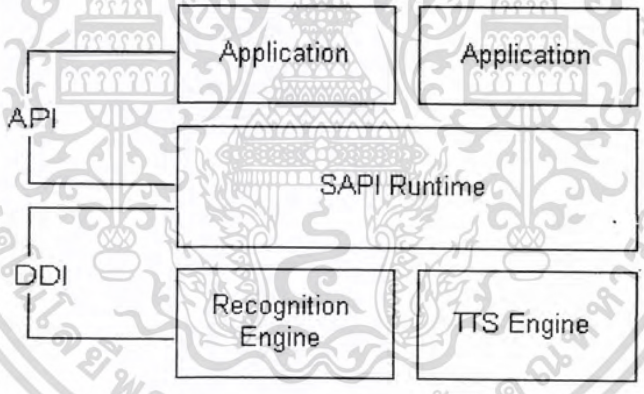
บทที่ 6
การเขียนโปรแกรมสั่งงานด้วยเสียงโดยใช้ Microsoft Speech SDK 5.1

ปัจจุบันเทคโนโลยีเกี่ยวกับคอมพิวเตอร์ ได้พัฒนาไปอย่างกว้างขวาง จึงมีผู้ที่เป็นนักพัฒนาได้ออกแบบและสร้างเครื่องมือหนึ่งที่สามารถ พูดแล้วสามารถให้เครื่องคอมพิวเตอร์รับรู้ว่าเราได้พูดข้อความอะไรไป ซึ่งเครื่องมือประเภทนี้ก็มีหลายกลุ่มที่ทำการพัฒนา แต่ในโครงงานนี้เราเลือกใช้ Microsoft Speech SDK 5.1

6.1 Microsoft Speech SDK 5.1 คือ

Microsoft. Speech SDK 5.1 เป็น Speech API application programming interface (SAPI) เป็นตัวช่วยในการพัฒนาโปรแกรมให้ง่ายขึ้นลดจำนวนคำสั่งในโปรแกรมให้น้อยลง และ เป็นหนึ่งในหลายๆ เครื่องมือที่ช่วยในการพัฒนาโปรแกรม

6.2 โครงสร้าง Speech API



รูปที่ 6-1 แสดงโครงสร้างของ Speech API

จากรูปข้างต้น SAPI ถ้ามองในระดับ high-level จะเป็นตัวติดต่อสื่อสารระหว่าง Application กับ Speech Engine ถ้ามองในระดับ low-level SAPI จะเป็นการควบคุมและจัดการ Speech Engine ในรูปแบบ real time ใน SAPI จะประกอบไปด้วย SAPI engines สองชนิดด้วยกันคือ Text to speech conversion (TTS Engines) เป็นโปรแกรมที่จะเปล่งเสียงคล้ายเสียงมนุษย์ออกมาทางลำโพง ตามข้อความ หรือ ไฟล์ ตามที่เรากำหนดไว้ Speech Recognition เป็นโปรแกรมที่จะแปลงจากเสียงของมนุษย์ที่รับเข้ามาทาง ไมโครโฟน ออกมาอยู่ในรูปแบบข้อความ ตามที่เราได้พูดออกไป ในส่วนของโครงงานนี้จะเน้นไปในส่วนของการพัฒนาโปรแกรม Speech Recognition

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.1 Text-to-Speech API

โปรแกรมสามารถใช้งาน Text to speech API โดยใช้ ISpVoice ซึ่งเป็น Component Object Model (COM) โปรแกรมจะต้องสร้าง IspVoice object โปรแกรมจะต้องเรียกใช้ ISpVoice::Speak ซึ่งมันจะสร้างเสียง ออกมาจากข้อความที่เป็นตัวอักษร และยังมีอีกหลายฟังก์ชั่นที่ไว้สำหรับเปลี่ยนเสียง และเปลี่ยนคุณสมบัติต่างๆ เช่น อัตราการเปล่งเสียง ISpVoice::SetRate ระดับเสียงที่ออกเอาท์พุต (speaker) ISpVoice::SetVolume Speech Recognition API

6.2.2 Recognition Engine

โปรแกรมสามารถใช้งาน Speech Recognition API โดยอาศัย IspRecoContext เป็นหลัก โปรแกรมในการรู้จำเสียง (Speech Recognition Program) ที่ใช้ Speech Recognition API Speech Recognition APIจะแบ่งหมวดการทำงานออกเป็น 2 ประเภท ซึ่งหมวดการทำงานนี้ จะเรียกว่า speech recognition engines (IspRecognizer)

6.2.2.1 Shared Recognizer Engine

Shared Recognizer Engine เป็นรูปแบบที่นิยมกันโดยทั่วไปในการเขียนโปรแกรมในการรู้จำเสียง (Speech Recognition Program) ซึ่งโปรแกรมจะต้องสร้าง IspRecoContext เอาไว้สำหรับที่จะเข้ามาเรียกใช้ IspRecognizerและโปรแกรมจะต้องเรียกใช้ CLSID_SpSharedRecoContext ซึ่งเป็น Component Object Model (COM) ในกรณีนี้ Speech API จะทำการจัดเตรียม Audio Input Stream และถูกกำหนดให้เป็นค่ามาตรฐาน ซึ่งในโครงงานนี้เราเลือกใช้การทำงานในหมวดนี้ เนื่องจากว่าโครงงานนี้มีการจดจำเสียงเพียงแค่ไม่กี่คำเลย จึงเป็นวิธีที่เหมาะสมที่สุด

6.2.2.2 InProc Speech Recognition Engine

เหมาะสำหรับโปรแกรมขนาดใหญ่ที่ทำงานอยู่บน เครื่องให้บริการ (large server applications)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 Speech Recognition interfaces

ที่เกี่ยวกับการพัฒนาโปรแกรมในการจดจำเสียง (Speech Recognition Program) มีดังนี้

6.3.1 ISpRecoContext

6.3.2 ISpRecoGrammar

6.3.3 ISpRecoResult

6.3.4 ISpRecognizer

6.3.5 ISpPhrase

6.3.6 ISpPhraseAlt

6.3.7 IspProperties

6.3.1 ISpRecoContext

ในหนึ่งโปรแกรมจะต้องมี ISpRecoContext instances เพื่อที่จะมารับค่า recognitions และ ในหนึ่งโปรแกรมสามารถมีได้หลายๆ ISpRecoContext instances และแต่ ISpRecoContext instances ก็สามารถที่จะมี รูปแบบการ recognition ที่แตกต่างกัน การสร้างISpRecoContext สามารถสร้างได้โดยใช้ CreateRecoContext ซึ่งมีตัวอย่างการสร้าง ISpRecoContext ใน Visual Basic ดังนี้

```
Private Sub Form_Load()
    Dim SharedRecognizer As SpSharedRecognizer
    Set SharedRecognizer = CreateObject("SAPI.SpSharedRecognizer")
    Dim myContext As Object
    Set myContext = SharedRecognizer.CreateRecoContext
End Sub
```

ตัวอย่างข้างบน เป็นการสร้าง ISpRecoContext แบบ SpSharedRecognizer ซึ่งเหมาะกับการพัฒนาโปรแกรมที่ขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.2 ISpRecoGrammar

เป็นส่วนที่ช่วยในการจัดการคำต่างๆ ที่เราต้องการจะให้มีการจดจำ ซึ่งหนึ่ง SpRecognizer object สามารถมีได้หลาย SpRecoContext objects ซึ่ง SpRecoContext objects คือ ตัวบอกถึงคุณสมบัติของ Grammar และในทำนองเดียวกัน หนึ่ง SpRecoContext object ก็สามารถมี SpRecoGrammar ได้หลาย SpRecoGrammar เรียกความสัมพันธ์นี้ว่า one-to-many ระหว่าง SpRecoContext objects กับ SpRecoGrammar objects เมธทอดตัวอย่างที่สำคัญดังนี้

```
Grammar.CmdSetRuleState "MenuCommands", SGDSActive
```

ตัวอย่างข้างบนเป็นการ กำหนดให้ IspeechRecoGrammar มีการทำงาน Active

6.3.3 ISpRecoResult

เป็นส่วนที่จะใช้ในการรับค่าผลลัพธ์ที่ส่งออกมาจาก Speech recognitions engin ซึ่งจะได้ค่าที่ส่งกลับออกมา 3 ชนิดดังนี้

hypotheses คือคำที่มีความเป็นไปได้จากข้อสมมุติฐานต่างๆ

recognitions คือค่าที่รู้จำคำได้

false recognitions คือไม่สามารถรู้จำเสียงได้

โปรแกรมตัวอย่างในส่วน ISpRecoResult มีดังนี้

```
Private Sub RecoContext_Recognition(ByVal StreamNumber As Long, _
                ByVal StreamPosition As Variant, _
                ByVal RecognitionType As SpeechRecognitionType, _
                ByVal Result As ISpeechRecoResult )
Dim strText As String
                strText = Result.PhraseInfo.GetText
                Debug.Print "Recognition: " & strText & ", " & _
                StreamNumber & ", " & StreamPosition
                txtSpeech.SelStart = m_cChars
                txtSpeech.SelText = strText & " "
                m_cChars = m_cChars + 1 + Len(strText)
End Sub
```

ตัวอย่างข้างบน เป็นฟังก์ชั่นที่ใช้ในการรับค่าการ recognitions ซึ่งค่าที่ได้จะสามารถที่จะนำออกมาแสดงโดยอาศัย IspeechRecoResult และทำการเรียกใช้ ฟังก์ชั่น ISpPhrase::GetText

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.4 ISpRecognizer

เป็นส่วนที่ใช้ควบคุมรูปแบบ speech recognition (SR) engine โปรแกรมสามารถติดต่อกับแต่ละหนึ่ง recognizer object กับหลายๆ recognition contexts ซึ่งอาจจะเป็นการสั่งให้โปรแกรมเริ่มต้นการ recognition หรือว่าให้หยุดการ recognition หรือว่าจะเป็นการรับเหตุการณ์ Event และการรับค่าผลลัพธ์จากการ recognition

6.3.5 IspPhrase

เป็นส่วนที่ใช้ในการเข้าถึงค่าผลลัพธ์คำ ที่ได้จากการทำ recognition เรียบร้อยแล้ว ฟังก์ชั่นที่ใช้ยกตัวอย่างเช่น ฟังก์ชั่น GetText

6.3.6 IspPhraseAlt

เป็นส่วนที่ใช้สนับสนุน ในส่วนของการประมวลผล คำ หรือ วลี ในโปรแกรมโดยการเรียกใช้

ISpRecoResult::GetAlternates

ซึ่งมี ISpPhraseAlt object เป็นตัวคอยจัดการ การรับข้อมูลจาก Speech recognition engine และทำการ Update Speech recognition engine's language model เมื่อมีการเปลี่ยนแปลง

6.3.7 IspProperties

ใช้สำหรับปรับค่าคุณสมบัติของ Speech recognition engines คือ IspRecognizer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 โครงสร้างที่สำคัญของโปรแกรมในส่วนการสั่งด้วยเสียง

โปรแกรมการสั่งงานด้วยเสียงในโครงงานนี้จะเป็นการสั่งงานผ่านไมโครโฟน ไปควบคุมการทำงานของเมาส์ Mouse ซึ่งคำสั่งที่ใช้งานในการควบคุมเมาส์ Mouse ในโครงงานนี้มีดังนี้ Click, Drag, Drop, DoubleClick, RightClick, free, lock, reset, exit ซึ่งจะเห็นว่าโปรแกรมจะจดจำคำเพียงไม่กี่คำเท่านั้น ดังนั้นจะเห็นว่าเราควรจะเลือกใช้ Speech Recognition Engines ที่เป็น Shared Recognizer Engine

```
Dim WithEvents RecoContext As SpSharedRecoContext
Set RecoContext = New  SpSharedRecoContext
```

เป็นการสร้าง IspRecoContext เอาไว้สำหรับที่จะเข้ามาเรียกใช้ IspRecognizer ในหมวด Shared Recognizer เมื่อเราทราบว่าจะใช้ หมวด Engine ชนิดใดแล้ว จากนั้นก็มีอีกคำหนึ่งที่สำคัญคือ Grammar ซึ่ง Grammar จะเป็นการเรียกแทนกลุ่มของคำ ที่เราจะทำการจำเสียง Recognition ซึ่งหลังจากที่ได้สร้าง Shared Recognizer Engine แล้วทำการสร้าง Grammar

```
Dim Grammar As SpeechLib.IspeechRecoGrammar
Set Grammar = RecoContext.CreateGrammar(0)
```

และเมื่อสร้าง Grammar แล้วก็ต้องทำการกำหนดคุณสมบัติของ Grammar โดยคำสั่งต่อไปนี้

```
Dim MenuCommands As SpeechLib.IspeechGrammarRule
Set MenuCommands = Grammar.Rules.Add("MenuCommands",
SRATopLevel Or SRADynamic, 1)
```

โปรแกรมข้างต้นกำหนดคุณสมบัติของ Grammar เรียกว่า Grammar Rules เป็นการปรับตั้งค่าของ Grammar ซึ่งในที่นี้เป็นการสร้าง Rules ขึ้นมาใช้เอง ขั้นตอนต่อไปจะเป็นการเพิ่มคำที่ต้องการเข้าไปใน Grammar Rules

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MenuCommands.InitialState.AddWordTransition Nothing, "click", "", SGLexical, "Click", 1, 0, 1
MenuCommands.InitialState.AddWordTransition Nothing, "drag", "", SGLexical, "Drag", 1, 0, 1
MenuCommands.InitialState.AddWordTransition Nothing, "drop", "", SGLexical, "Drop", 1, 0, 1
```

เมื่อเพิ่มคำเข้าไปใน Grammar Rules แล้วจากนั้นก็ทำการ Commit เพื่อเป็นการเช็คว่า เราได้ทำถูกต้องตามโครงสร้าง Grammar Rule และทำการบันทึกค่าเก็บไว้

```
Grammar.Rules.Commit
```

จากนั้นก็กำหนด Grammar Rules มีการทำงาน Active ตามคำสั่งต่อไปนี้

```
Grammar.CmdSetRuleState "MenuCommands", SGDSActive
```

และสุดท้ายเป็นส่วนที่จะใช้ในการรับค่าผลลัพธ์ที่ส่งออกมาจาก Speech recognitions engin ซึ่งจะใช้ ISpRecoResult

```
Private Sub RecoContext_Recognition(ByVal StreamNumber As Long, _
        ByVal StreamPosition As Variant, _
        ByVal RecognitionType As SpeechRecognitionType, _
        ByVal Result As ISpeechRecoResult )
    Dim strText As String
        strText = Result.PhraseInfo.GetText
        Debug.Print "Recognition: " & strText & "," & _
        StreamNumber & "," & StreamPosition
        txtSpeech.SelStart = m_cChars
        txtSpeech.SelText = strText & " "
        m_cChars = m_cChars + 1 + Len(strText)
End Sub
```

Speech SDK สามารถใช้ได้กับ C#, C++, VB,Win32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

แนวคิดการออกแบบ

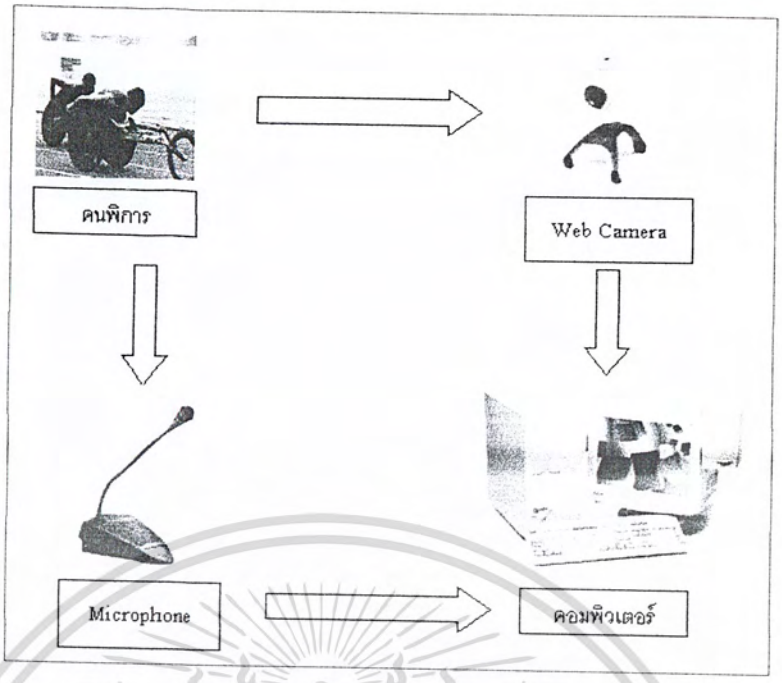
7.1 Funtional requirements

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

7.2 Non-functional Requirements

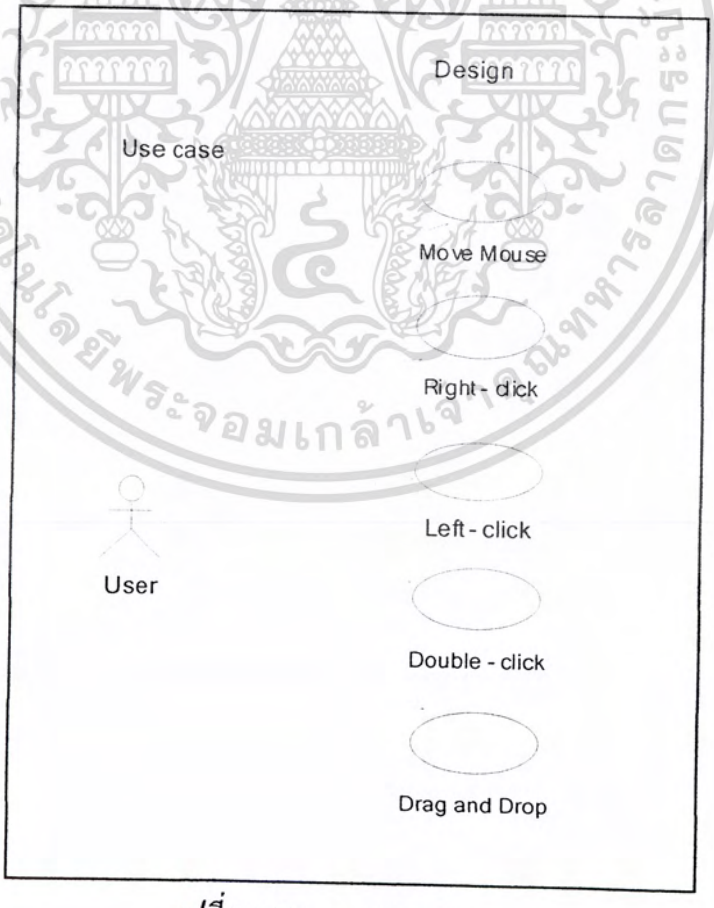
- ความเร็ว
- ความแม่นยำ
- ใช้ง่าย
- รองรับผู้ใช้ให้ได้มากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



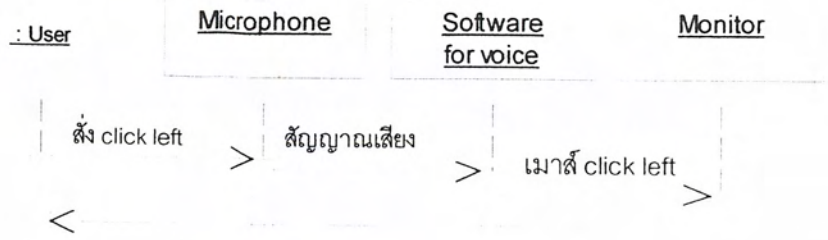
รูปที่ 7-1 การออกแบบ High-Level

7.3 Use Case



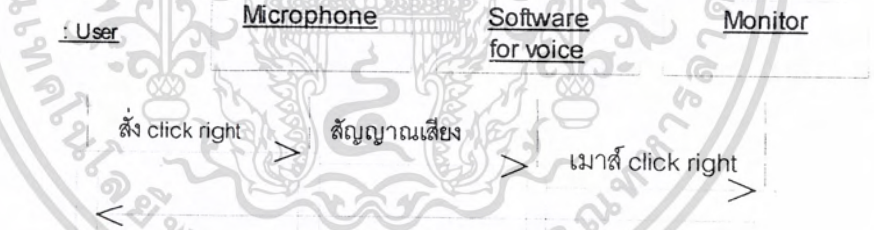
รูปที่ 7-2 Use case diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Click Left

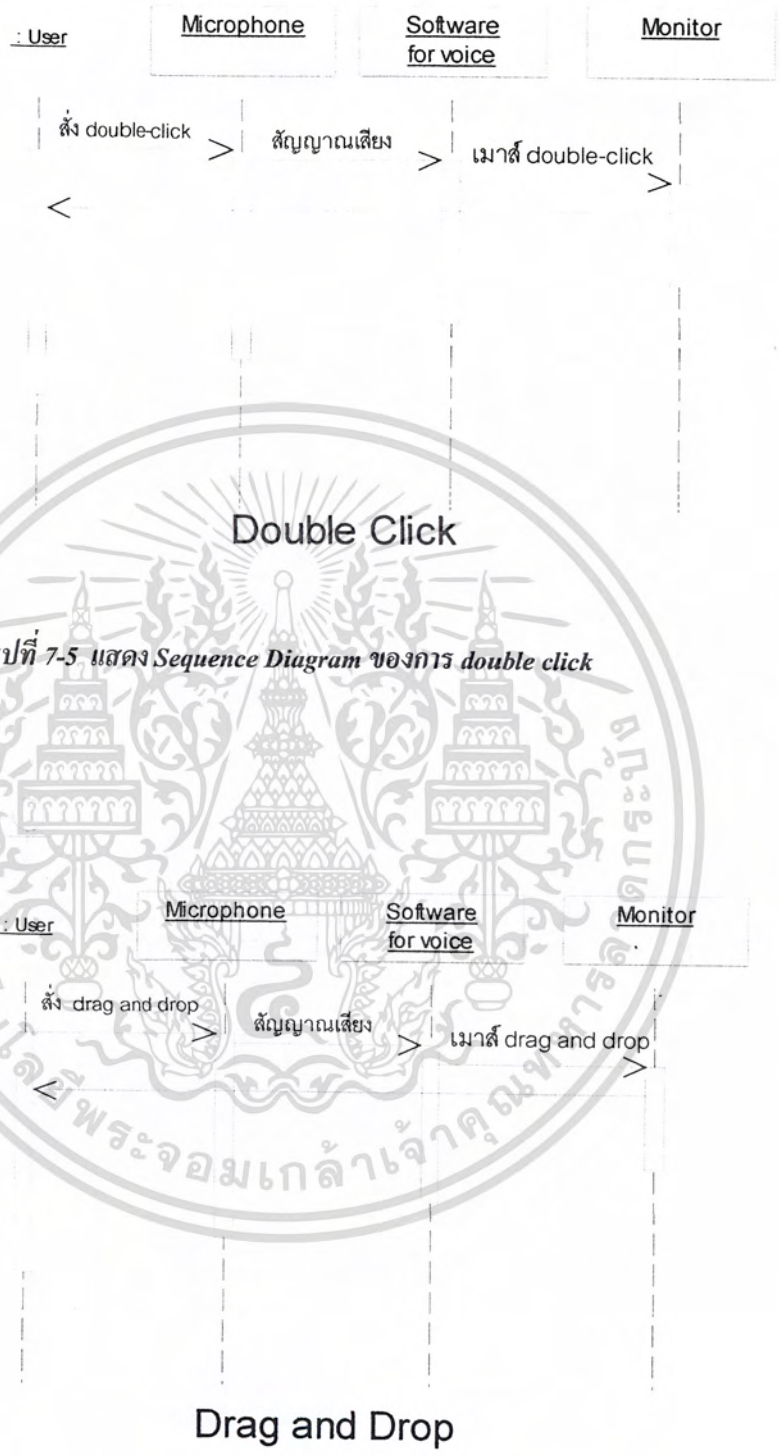
รูปที่ 7-3 แสดง Sequence Diagram ของการ click left



Click Right

รูปที่ 7-4 แสดง Sequence Diagram ของการ click right

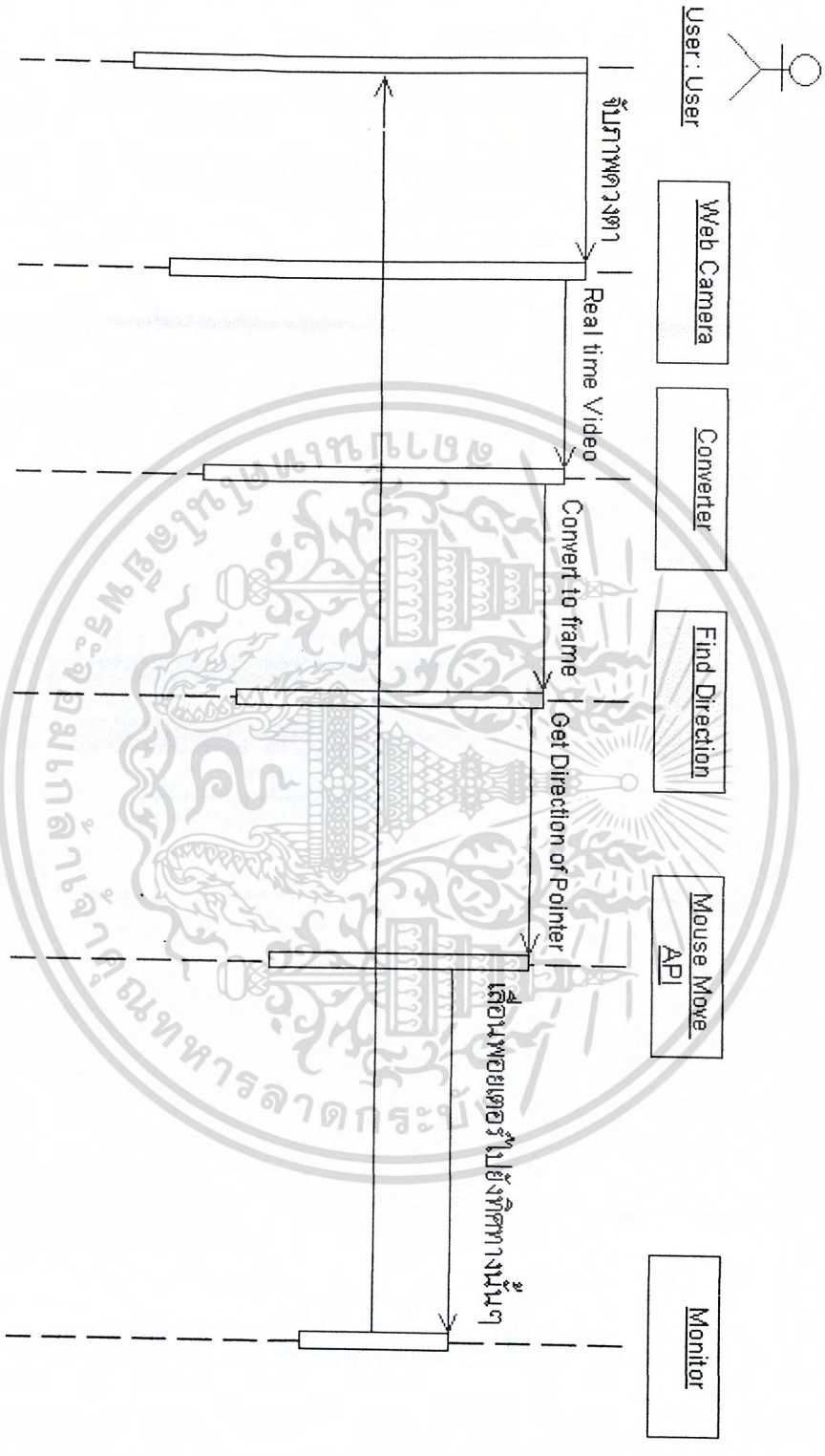
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-5 แสดง Sequence Diagram ของการ double click

รูปที่ 7-6 แสดง Sequence Diagram ของการ drag and drop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-6 แสดง Sequence Diagram ของการ Mouse Move

บทที่ 8
อัลกอริทึมและการแสดงผลตำแหน่งบนหน้าจอภาพ

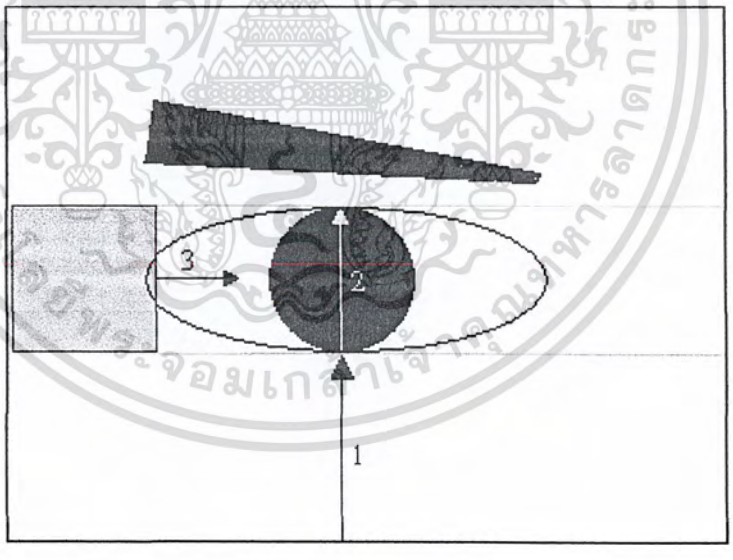
8.1 บทนำ

เนื่องจากโครงงานวิจัย นี้เป็นการนำภาพดวงตามาวิเคราะห์ประมวลผลการเคลื่อนที่ เพื่อที่จะนำไปควบคุมการทำงานของเมาส์ ดังนั้นเราจะมีวิธีการใดที่จะนำภาพดวงตาที่เราได้มา มาวิเคราะห์หาอัตราการเคลื่อนที่ แล้วนำไประบุลงตำแหน่งในจอภาพ ได้อย่างถูกต้อง ซึ่งอาจจะต้องมีการวิเคราะห์หากึ่งกลางของดวงตา หรืออาจจะเป็นการตีกรอบล้อมรอบลูกตา เป็นต้น

8.2 อัลกอริทึม 1

อัลกอริทึมนี้จะใช้ในการค้นหาลูกตาดำจากภาพที่รับเข้ามาโดยขั้นตอนในการทำงานอย่างคร่าวๆ จะแบ่งเป็น 3 ขั้นตอนดังต่อไปนี้

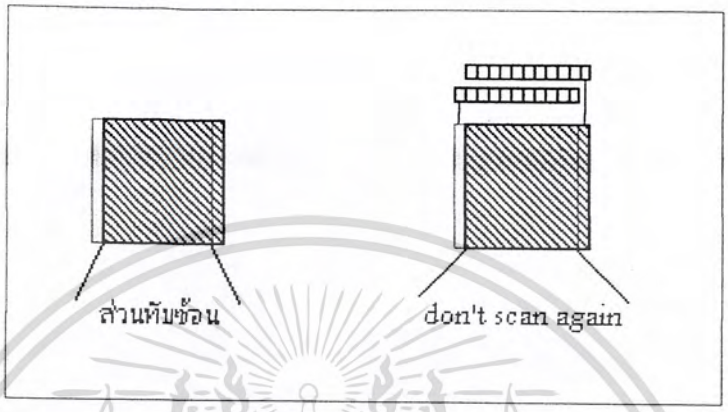
1) หาขอบตาล่างก่อน เนื่องจากหากหาขอบตาบนก่อน ก็จะพบกับคิ้วซึ่งในที่นี้ถือว่าเป็น noise
2) หาขอบตาบนโดยค้นหาต่อจากขอบตาล่าง
3) หาด้านซ้ายของดวงตาโดยการใช้ sliding window



รูป 8-1 แสดงการทำงานตามอัลกอริทึม 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

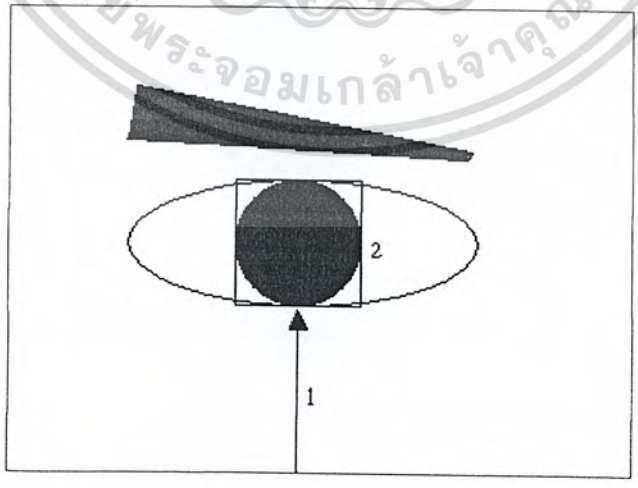
เนื่องจากการค้นหาข้อมูลโดยการใช้ sliding window จะมีการค้นหาข้อมูลจุดสีดำซ้ำกันในแต่ละ window โดยจะดูได้จากส่วนที่ซ้อนทับกันของ window โดยเราสามารถปรับปรุงโดยการใช้ buffer มาเก็บข้อมูลผลรวมของจุดสีดำในแต่ละคอลัมน์ ซึ่ง buffer ที่ใช้จะเป็นลักษณะของ circle queue การค้นหาแบบนี้จะไม่ค้นหาจุดที่เคยค้นหาแล้ว



รูปที่ 8-2 แสดงทำงานของ sliding window และการปรับปรุง

8.3 อัลกอริทึม 2

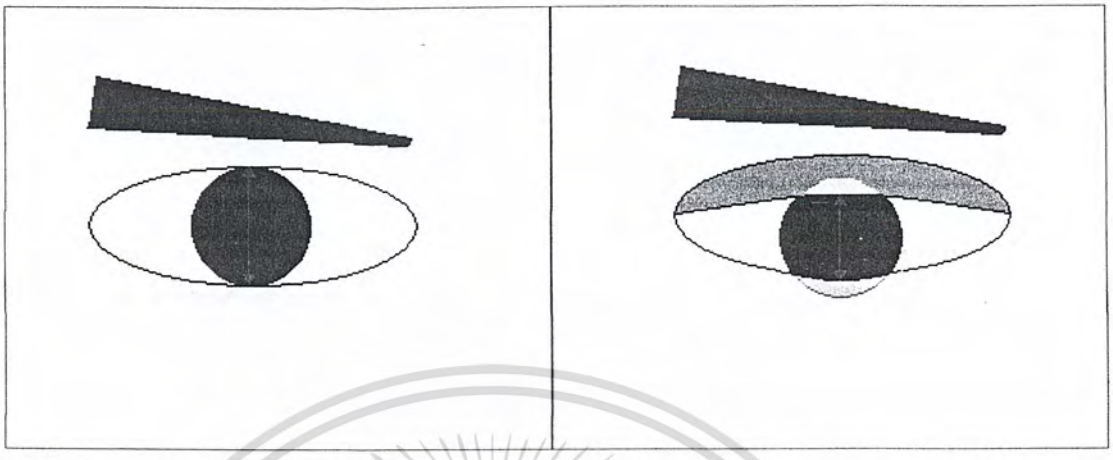
อัลกอริทึมนี้จะต้องมีการกำหนดค่าเริ่มต้นก่อนโดยค่าที่ต้องกำหนดคือขนาด window ที่จะนำมาครอบลูกตาดำหลังจากนั้นก็จะเริ่มกระบวนการหาลูกตาดำโดยทำการหาขอบตาล่างก่อนเมื่อพบก็จะนำ window ไปใส่เลย



รูปที่ 8-3 แสดงการทำงานตามอัลกอริทึม 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

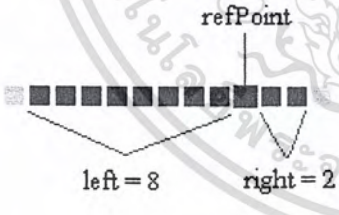
อาศัยการสังเกตพฤติกรรมของคนจะพบว่าในการมองลงด้านล่างจะทำให้หนังตาปิดลงมาเล็กน้อยทำให้ขนาดของลูกตาดำไม่สมมาตรจึงไม่สามารถใช้ window ที่มีขนาดคงที่ได้

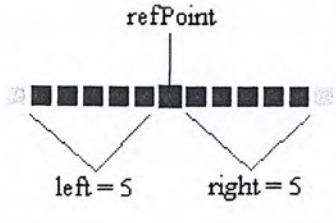


รูปที่ 8-4 แสดงการเคลื่อนที่ของลูกตาคน

8.4 อัลกอริทึม 3

อัลกอริทึมวิธีนี้จะเริ่มจากการกำหนดจุดอ้างอิงก่อนโดยจุดที่ใช้นี้จะต้องอยู่ในบริเวณพื้นที่สีดำหลังจากนั้นจึงจะเริ่มทำการ scan ทั้งหมด 4 ทิศทางคือ ซ้าย, ขวา, บน, ล่าง โดยจะทำการนับจุดสีดำและหยุดเมื่อเจอสีขาวเพื่อหาค่าความแตกต่างของจำนวนจุดสีทิศทาง ซ้าย – ขวา, บน - ล่าง หลังจากนั้นก็จะทำการย้ายจุดอ้างอิงเพื่อให้ค่าความแตกต่างของจำนวนจุดสีทิศทาง ซ้าย - ขวา, บน - ล่าง เท่ากันแบ่งได้ 4 กรณีดังนี้

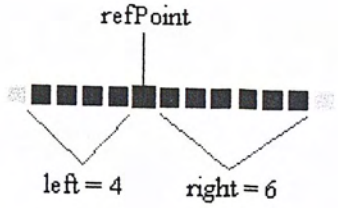




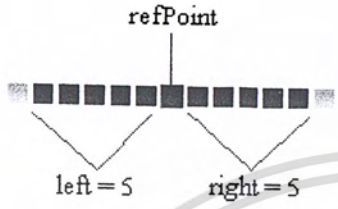
$$refPoint_{(new)} = refPoint_{(old)} - \frac{(left - right)}{2}$$

รูปที่ 8-5 แสดงการทำงานของกรณี left ลบ right มากกว่า 1

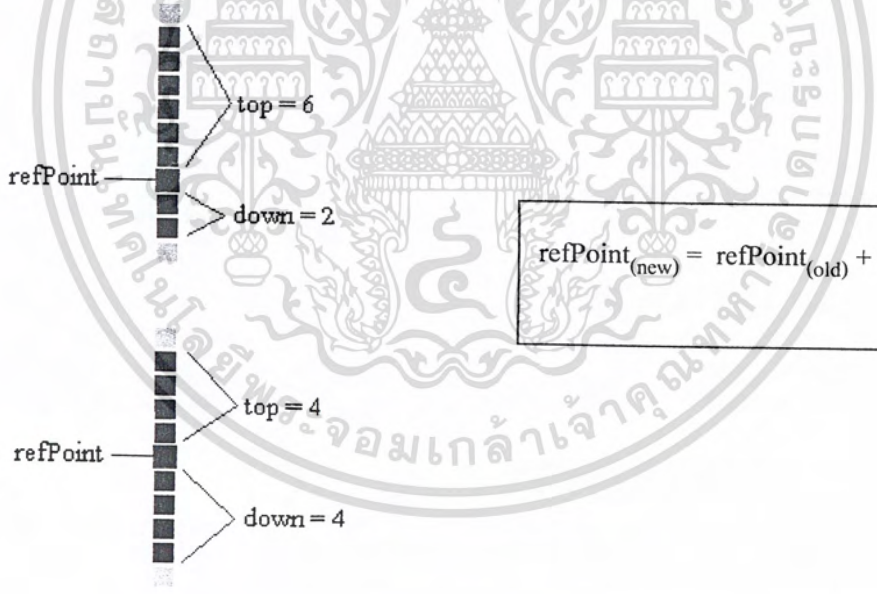
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



$$refPoint_{(new)} = refPoint_{(old)} + \frac{(right - left)}{2}$$



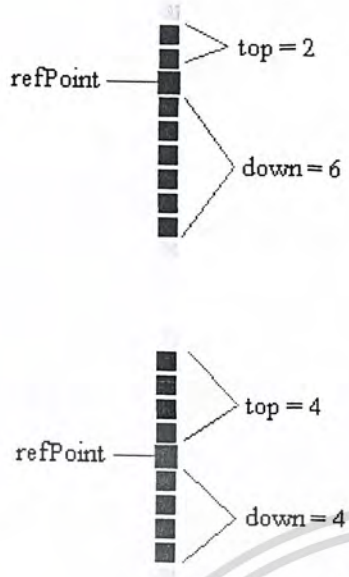
รูปที่ 8-6 แสดงการทำงานของกรณี left ลบ right น้อยกว่า -1



$$refPoint_{(new)} = refPoint_{(old)} + \frac{(top - down)}{2}$$

รูปที่ 8-7 แสดงการทำงานของกรณี top ลบ down มากกว่า 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



$$refPoint_{(new)} = refPoint_{(old)} - \frac{(down - top)}{2}$$

รูปที่ 8-8 แสดงการทำงานของกรณี top ลบ down น้อยกว่า -1

ซึ่งผลที่ได้จะมีผลให้ refPoint(new) ไปอยู่ที่ตำแหน่งกึ่งกลางส่วนกรณีที่ผลจากการลบกันของ top/down และ left/right เท่ากับ 0,1,-1 จะไม่มีการเปลี่ยนแปลงค่าของ refPoint

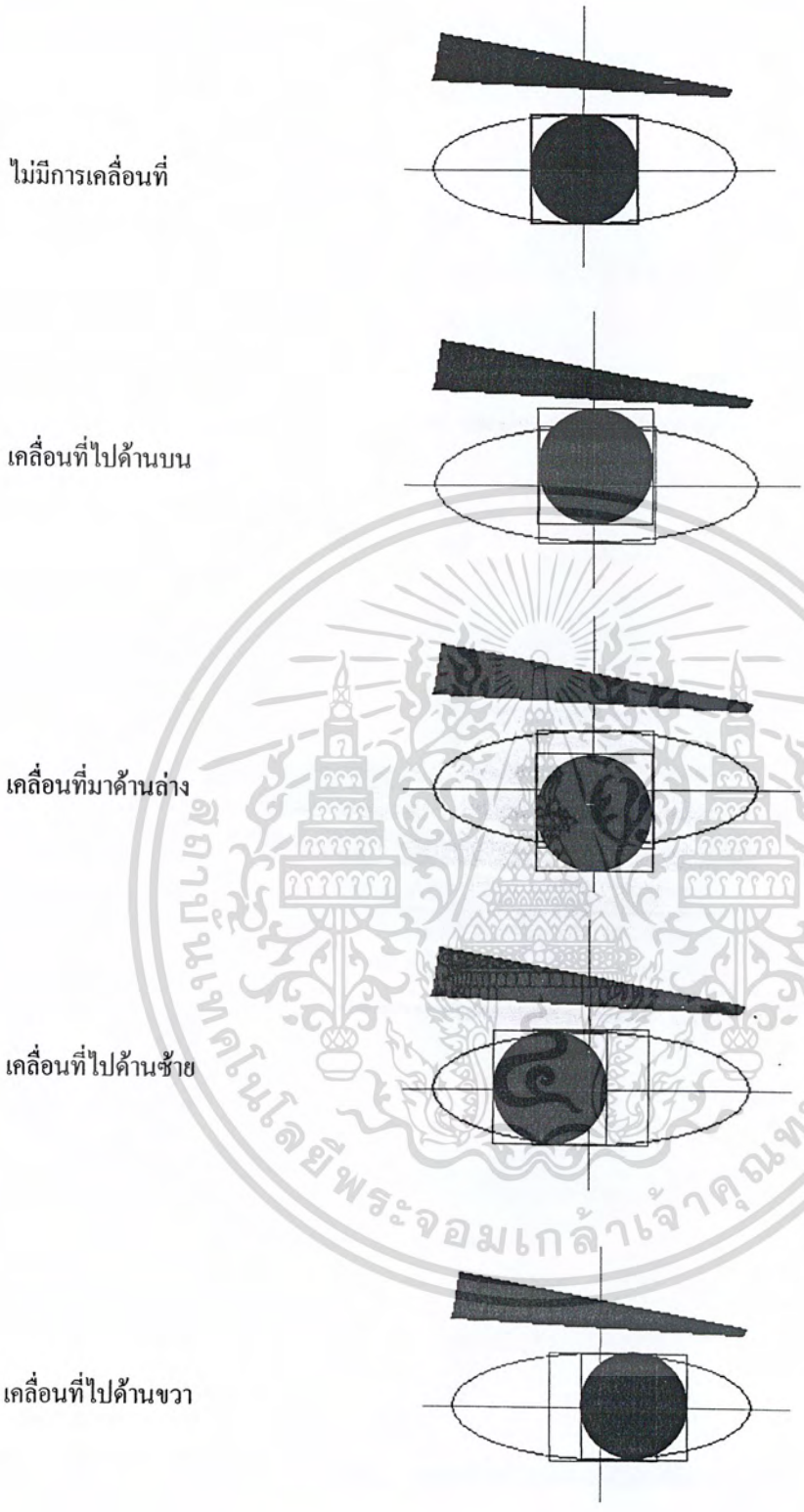
8.5 อัลกอริทึม 4

เนื่องจากที่ได้ลองศึกษาและทดลองเขียนโปรแกรมจากอัลกอริทึมข้างต้นที่ผ่านมา ทำให้หาข้อสรุปได้ข้อหนึ่งว่า ในกรณีที่เราอัลกอริทึมเหล่านั้นไปควบคุมเมาส์ จะทำให้เกิดการเคลื่อนที่ของเมาส์ ที่ไม่มีความเรียบ เมาส์จะมีการเคลื่อนที่กระโดดไปมาเนื่องจากดวงตาของเราไม่นิ่งพอ

ดังนั้นจึงเป็นที่มาของอัลกอริทึมนี้ คือเราจะสมมุติการเคลื่อนที่ของเมาส์คลายๆกับกำลังควบคุมจอยสติกอยู่ ซึ่งก็คือถ้าดวงตาเรามองไปทางไหนเมื่อเปรียบเทียบกับจุดอ้างอิงแล้ว แสดงว่าเราต้องการที่จะให้เมาส์เคลื่อนที่ไปจุดนั้น

เมื่อได้ทิศทางแล้วเมาส์ก็จะค่อยๆเคลื่อนที่ไปยังจุดนั้น ซึ่งค่าอัตราความเร็วของการเคลื่อนที่ของเมาส์ จะขึ้นอยู่กับการสั่งงานด้วย ซึ่งถ้าเราสั่งเพิ่มความเร็วโดยคำสั่ง สปีด (speed) ความเร็งของการเคลื่อนที่จะเร็วขึ้นอีกระดับหนึ่ง ถ้าพูดสปีดอีกก็จะเร็วขึ้นอีก จนกว่าเมื่อถึงตำแหน่งที่ต้องการแล้วเราต้องทำการสั่งเมาส์ให้หยุดเคลื่อนที่ ซึ่งจากแนวคิดนี้จะทำให้เราสามารถที่จะควบคุมเมาส์ให้ สามารถเลือกจุดที่จะคลิกได้ละเอียดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-9 แสดงการทำงานตามอัลกอริทึม 4

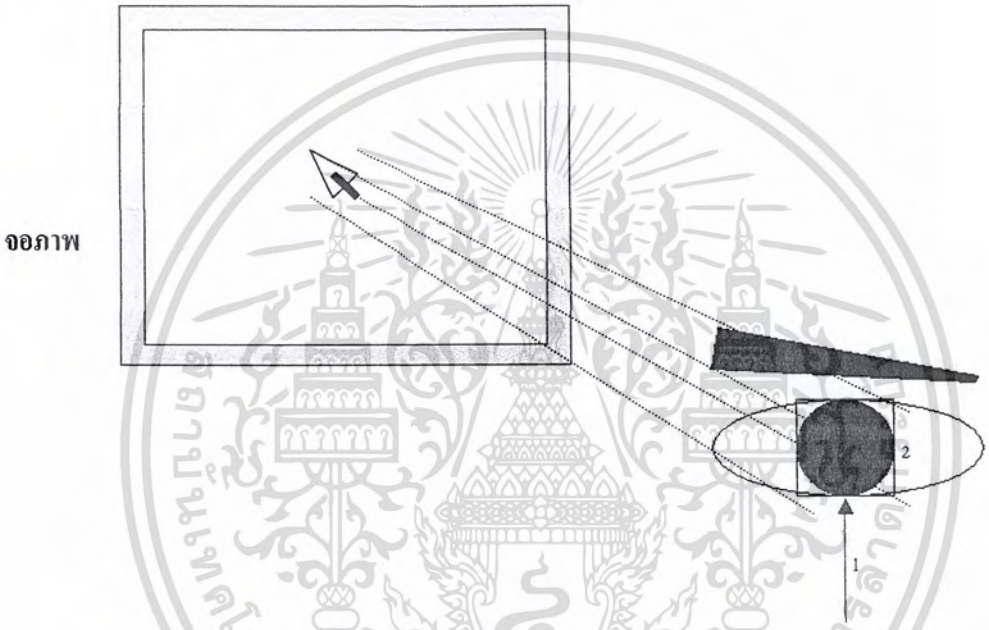
เราจะทำการเลือกกรอบให้กับดวงตา แล้วทำการล็อกไว้โดยแทนด้วยสีแดงตามรูป ซึ่งจะให้เป็นจุดอ้างอิงของเมาส์หรือว่าจุดกึ่งกลางนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

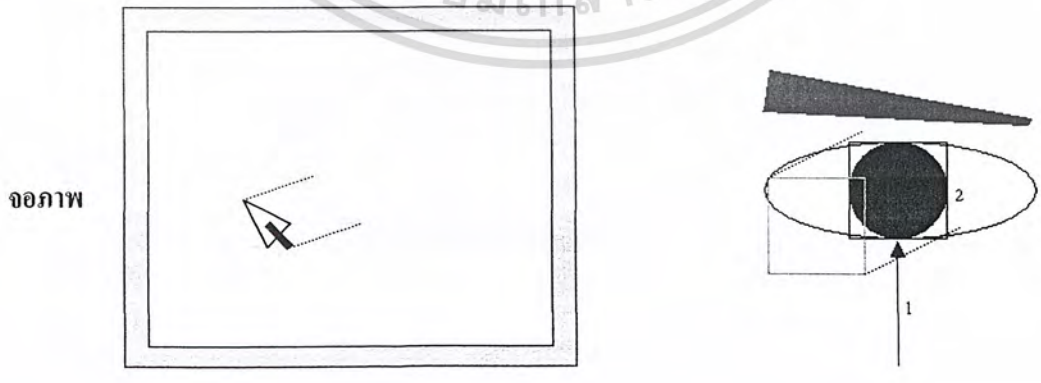
แล้วจากนั้นจะมีกรอบสีน้ำเงินจะเป็นตัวที่เคลื่อนที่ตามการเคลื่อนที่ของดวงตา ถ้าดวงตามีการเคลื่อนที่เราก็จะนำเอาตำแหน่งของการเคลื่อนที่ของกรอบสีน้ำเงินมาเปรียบเทียบกับกรอบสีแดงซึ่งเป็นจุดอ้างอิง จะทำให้เราทราบว่าขณะนั้นเมาส์ มีการเคลื่อนที่ไปในทิศทางใด

และถ้ากรอบสีน้ำเงิน เกิดเคลื่อนที่หลุดออกนอกจากกรอบสีแดงแสดงว่าตาเรามีการมองหลุดกรอบของจอภาพ เนื่องจากการกวาดสายตาของคนเราจากขวาไปซ้ายจะมีระยะทางที่จำกัด ซึ่งไม่น่าจะเกินจากกรอบอ้างอิง

8.6 การแสดงผลตำแหน่งบนหน้าจอภาพ



รูปที่ 8-10 แสดงตำแหน่งพอยเตอร์ที่อยู่กึ่งกลางจอภาพ และดวงตาอยู่จุดอ้างอิง



รูปที่ 8-11 แสดงตำแหน่งพอยเตอร์ เมื่อมีการเคลื่อนที่ของดวงตาเปรียบกับจุดอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9
การใช้เสียงควบคุมโปรแกรม

9.1 บทนำ

การสั่งงานโปรแกรมจะเป็นการสั่งงานด้วยเสียง โดยใช้ตัว Speech API เป็นตัวจัดการเรื่องของการ ตรวจสอบคำที่พูดออกมา Speech Recognition และเมื่อรู้ว่าคำที่พูดเข้ามาเป็นคำอะไร แล้วก็นำมาเรียกใช้ Windows API เพื่อควบคุมการทำงานของเมาส์ และไปควบคุมโปรแกรม

9.2 คำสั่งควบคุมการทำงานโปรแกรม

9.2.1 การคลิก click

Call mouse_event(MOUSEEVENTF_LEFTDOWN, 0, 0, 0, 0)

Call mouse_event(MOUSEEVENTF_LEFTUP, 0, 0, 0, 0)

9.2.2 การดับเบิลคลิก doubleclick

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)

9.2.3 การคลิกขวา rightclick

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

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

9.2.4 การแดรก drag

Call mouse_event(MOUSEEVENTF_LEFTDOWN, 0, 0, 0, 0)

9.2.5 การดร็อป drop

Call mouse_event(MOUSEEVENTF_LEFTUP, 0, 0, 0, 0)

9.2.5 การสั่งให้เปลี่ยน โหมดการแสดงภาพ switch

9.2.6 การสั่งให้ล็อกกรอบของดวงตา lock

9.2.7 การสั่งให้ยกเลิกการล็อกดวงตา free

9.2.8 การสั่งให้โปรแกรมซ่อน (hidden)ลงมาอยู่ที่ system tray

9.2.9 การสั่งให้การเคลื่อนที่ของเมาส์ช้าลง show

9.2.10 การสั่งให้การเคลื่อนที่ของเมาส์เร็วขึ้น speed

9.2.11 การสั่งให้ออกจากโปรแกรม exit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 10

การทดสอบระบบ

10.1 ทดสอบอัตราความเร็วของเมาส์

- สูตรคำนวณหาอัตราความเร็วของเมาส์

  อัตราความเร็วของเมาส์ = ระยะทาง / เวลาในการคลิก หน่วย pixel/Sec.

- สูตรคำนวณหาระยะทาง

  ระยะทาง = $\text{sqt}\ (x2-x1)^2+(y2-y1)^2$

การออกแบบการทดสอบอัตราความเร็วของเมาส์

ตามรูปที่ 10-1

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 10 -1 หน้าต่างการทดสอบอัตราความเร็วของเมาส์

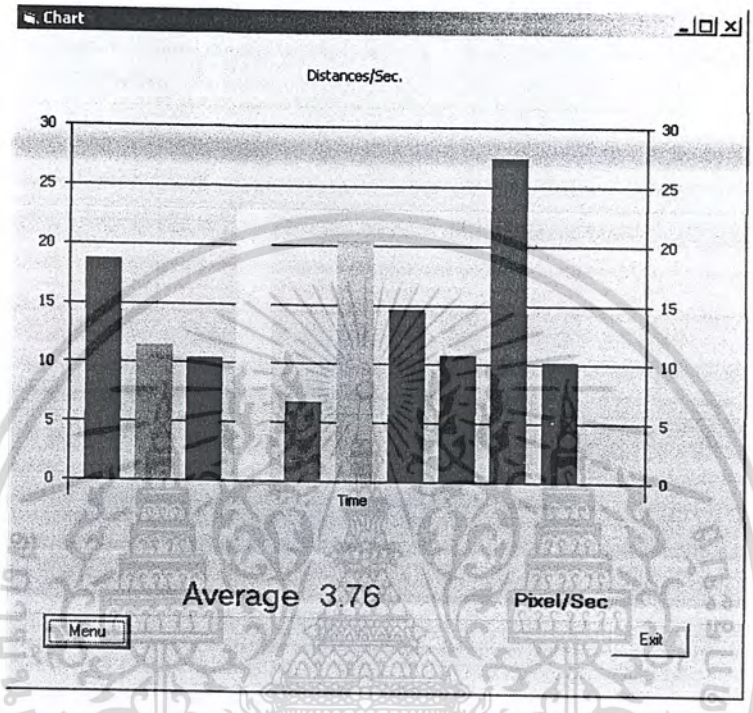
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์การทดสอบอัตราความเร็วของเมาส์ตามรูปที่10-0 จะแสดงด้วยกราฟแท่ง เป็นอัตราความเร็วในแต่ละครั้งของการคลิก และจะหาค่าเฉลี่ยตามสูตรด้านล่าง

o สูตรคำนวณหาค่าเฉลี่ยอัตราความเร็วของเมาส์

ค่าเฉลี่ยอัตราความเร็วของเมาส์ = ผลรวมอัตราความเร็วของเมาส์ N ครั้ง/ N

N = จำนวนครั้งที่คลิก



รูปที่ 10-2 ผลลัพธ์การทดสอบอัตราความเร็วของเมาส์

ผลลัพธ์จากการทดสอบอัตราความเร็วเมาส์

	ค่าเฉลี่ยอัตราความเร็วของเมาส์ (Pixel/Sec)
เมาส์ปกติ	52.82
ค่าจากโครงเดิม	10.6
จุดสี	2.92
ดวงตา	3.76

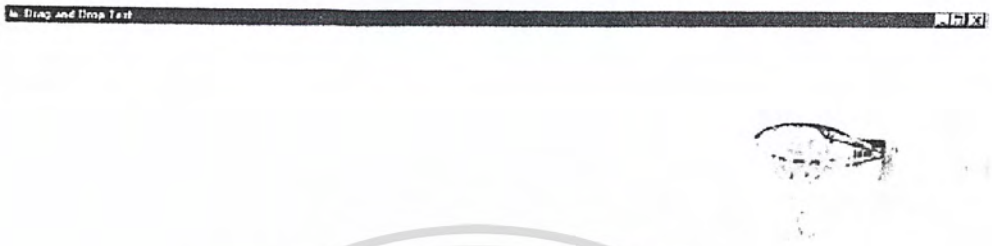
ตารางที่ 10-1การเปรียบเทียบอัตราความเร็วของเมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.2 การทดสอบการลากและปล่อยของเมาส์

การออกแบบการทดสอบการลากและปล่อยของเมาส์ตามรูปที่10 - 3

1. เริ่มโดยการกดปุ่ม Start
2. จากนั้นให้ใช้เมาส์ลากลูกบาสเกตบอลลงห่วง



รูปที่ 10 – 3 หน้าต่างการทดสอบการลากและปล่อยเมาส์

ผลลัพธ์จากการทดสอบการลากและปล่อยของเมาส์

	เวลาที่ใช้ในการลากลูกบาสเกตบอลลงห่วง (Sec)
เมาส์ปกติ	1
ค่าจากโครงเดิม	57
จุดสี	38
ดวงตา	37

ตารางที่ 10-2การเปรียบเทียบอัตราการลากและปล่อยของเมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

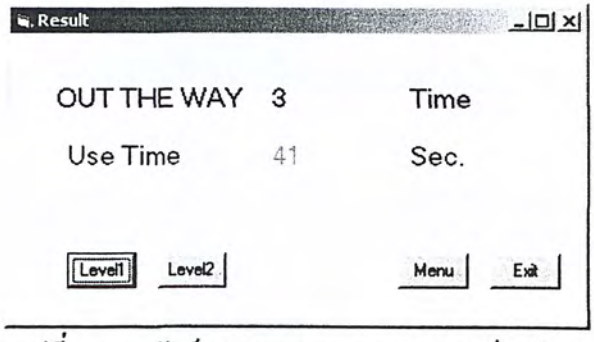
10.3 การทดสอบการควบคุมเมาส์

การออกแบบการทดสอบการควบคุมเมาส์ระดับ 1 ตามรูปที่ 10 - 4

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

รูปที่10-4 หน้าต่างการทดสอบการควบคุมเมาส์ระดับ 1

ผลลัพธ์เป็นไปตามรูปที่ 10 - 5 แอปพลิเคชันจะทำการบันทึกเวลาที่ใช้ในการเลื่อนเมาส์จากตำแหน่ง Start ถึง END และจำนวนครั้งที่เมาส์ออกนอกเส้นทาง



รูปที่10-5 ผลลัพธ์การทดสอบการควบคุมเมาส์ระดับ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์จากการทดสอบการควบคุมเมาส์ระดับที่ 1

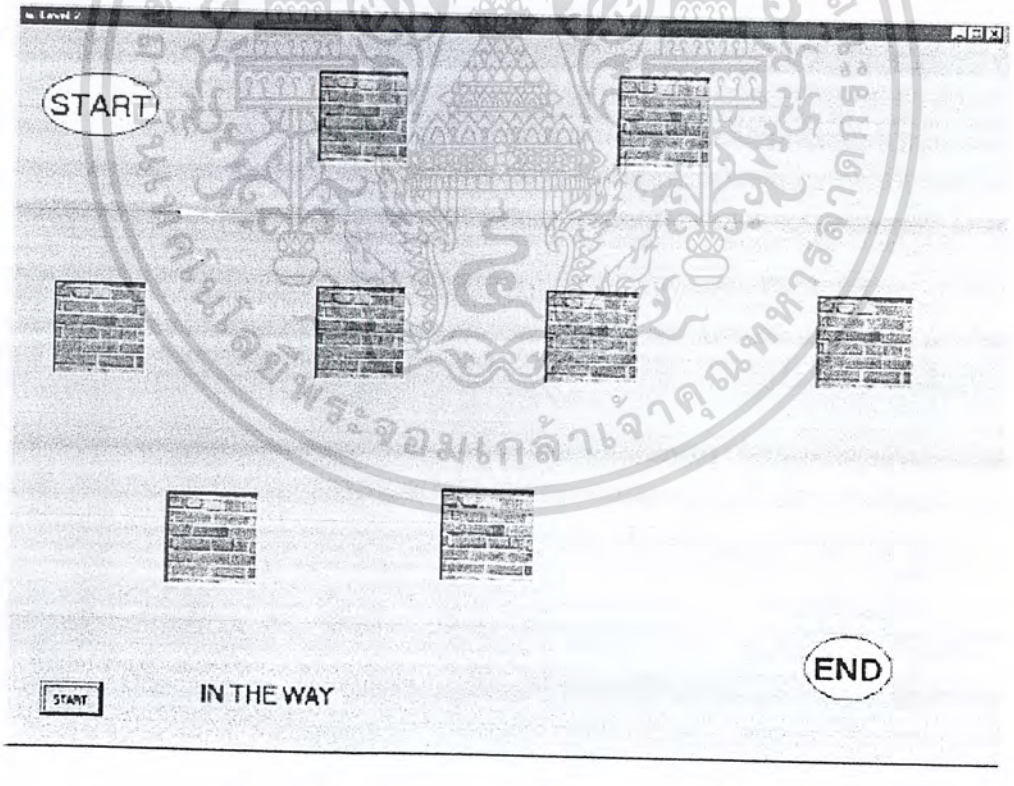
	การหลุดออกนอกเส้นทาง(ครั้ง)/เวลาที่ใช้ (Sec)
เมาส์ปกติ	1/4
ค่าจากโครงเดิม	52/67
จุดสี	20/29
ดวงตา	3/41

ตารางที่ 10-3การเปรียบเทียบอัตราการควบคุมเมาส์ระดับ 1

10.4 การออกแบบการทดสอบการควบคุมเมาส์ระดับ 2

ตามรูปที่ 10 - 6

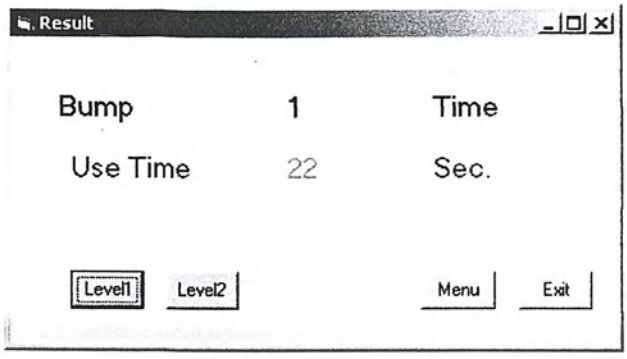
1. เริ่มโดนการกดปุ่ม Start
2. จากนั้นเมาส์จะไปอยู่ที่ตำแหน่ง Start (บริเวณพื้นที่สีขาว)
3. ให้เลื่อนเมาส์ไปตำแหน่ง END (บริเวณพื้นที่สีเหลือง) โดยเลื่อนให้อยู่ตามแนวที่กำหนด ไว้เมื่อเมาส์อยู่ในแนวที่กำหนด ด้านล่างของหน้าต่างจะขึ้นว่า "IN THE WAY" เมื่อเมาส์ชนผนังอิฐ ด้านล่างของหน้าต่างจะขึ้นว่า "BUMP"



รูปที่ 10-6 หน้าต่างการทดสอบการควบคุมเมาส์ระดับ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์เป็นไปตามรูปที่ 10 - 7 แอปพลิเคชันจะทำการบันทึกเวลาที่ใช้ในการเลื่อนเมาส์จาก ตำแหน่ง Start ถึง END และจำนวนครั้งที่เมาส์ชนผนังอิฐ



รูปที่ 10 - 7 ผลลัพธ์การทดสอบการควบคุมเมาส์ระดับ 2

ผลลัพธ์จากการทดสอบการควบคุมเมาส์ระดับที่ 1

	ชนกำแพง(Bump)/เวลาที่ใช้ (Sec)
เมาส์ปกติ	0/3
ค่าจากโครงเดิม	0/40
จุดสี	0/25
ดวงตา	1/22

ตารางที่ 10-4การเปรียบเทียบอัตราการควบคุมเมาส์ระดับ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 11
สรุปผลการทดลองและข้อเสนอแนะ

11.1 สรุปผลการทดลอง

จากการทดลองอัลกอริทึมทั้ง 4 อัลกอริทึมด้านบนจะเห็นว่าการที่ให้โปรแกรมทำงานเองแบบ auto ของอัลกอริทึม 1 และ 2 นั้นจะต้องใช้การ scan หาขอบตาล่างก่อนซึ่งมีผลให้การทำงานใช้เวลามากกว่าอัลกอริทึม 3 ที่มีการกำหนดจุดอ้างอิงนอกจากนั้นในอัลกอริทึม 2 ยังมีปัญหาในเรื่องของขนาดของลูกตาที่อาจจะเปลี่ยนแปลงได้เนื่องจากเปลือกตามาปิดส่วนที่เป็นลูกตาดำ

แต่เนื่องจากหลังจากที่มีการทดลองโปรแกรมตาม อัลกอริทึม 1,2,3 ตามลำดับแล้วพบว่า ในส่วนของหลักการ การเคลื่อนที่ของเมาส์เกิดการกระโดดไปมาไม่เรียบ Smooth ดังนั้นจึงมีแนวคิดใหม่เพื่อให้การเคลื่อนที่ของ เมาส์ เป็นไปด้วยความ smooth

จึงเกิดแนวคิด การเคลื่อนที่ของเมาส์แบบจอยสติก ตามการทำงานใน อัลกอริทึม 4 เมื่อได้มีการทดลองเขียนโปรแกรมแล้ว ทำให้เห็นการเคลื่อนที่ของเมาส์มีการ smooth ขึ้น และสามารถที่จะใช้งานในการเลือกบริเวณที่มีความละเอียดมากขึ้นกว่าเดิม เนื่องจากการเคลื่อนที่ของเมาส์จะเคลื่อนที่ไปทีละ pixel จึงสามารถที่จะเลือก ไอคอน (icon) ที่มีขนาดเล็กได้

11.2 ข้อเสนอแนะ

เนื่องจากการเคลื่อนที่ของดวงตาคนเรา เป็นการเคลื่อนที่ ไม่นิ่งคงที่อยู่ตลอดเวลา ดังนั้นแล้วตรงจุดนี้ถือเป็นปัญหาสำคัญอันหนึ่งในการพัฒนาโครงงานนี้ ดังนั้นควรจะมีการจัดสภาพแวดล้อม ให้เหมาะสมกับการใช้โปรแกรมก่อน เช่น หาตำแหน่งติดตั้งกล้อง และผู้ใช้งานต้องเป็นผู้ที่มีความอดทนต่อการเพิ่งเล็งอะไรนานๆ ทนต่อความเมื่อยล้าของดวงตา ถ้ามีการฝึกใช้งานบ่อยๆ ก็จะเกิดความชำนาญขึ้นเอง

และในส่วนของการสั่งงานด้วยเสียง อาจมีปัญหาตรงที่ว่า เมื่อสั่งแล้วมันไม่ตอบสนองทันที ทำให้เกิดการทำงานพลาด ดังนั้น ควรจะหาเครื่องมือในการเขียนโปรแกรมสั่งงานด้วยเสียงตัวใหม่ ที่มีการตอบสนอง เร็วกว่าเดิม

ในส่วนของไมโครโฟน ที่มีความไวสูงจะรับเอาเสียงที่เป็นสัญญาณรบกวนเข้าไป ซึ่งจะส่งผลให้โปรแกรมมีการทำงานที่ผิดพลาด ดังนั้นในการทดลองควรที่จะทดลองในที่ที่มีความเงียบพอสมควร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[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

[4] Rafael C. Gonzalez, Richard E. Woods: *"Digital Image Processing"*, Second Edition , Prentice Hall 2002

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้