

การบอกตำแหน่งของตัวชี้
โดยใช้เวกเตอร์การเคลื่อนไหวของตา
Locating pointer using eye motion vector

นางสาวจิรนนท์ จิระตระการวงศ์ 5033660023
นางสาวลลิต เดชธำรงวัฒน์ 5033701123

โครงการนี้เป็นส่วนหนึ่งของการศึกษาระดับปริญญาตรี
สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2553

โครงการ	การบอกตำแหน่งของตัวชี้โดยใช้เวกเตอร์การเคลื่อนไหวของตา		
	Locating pointer using eye motion vector		
โดย	1. นางสาวจิรนนท์	จิระระการวงศ์	รหัสประจำตัว 5033660023
	2. นางสาวลลิต	เดชธำรงวัฒน์	รหัสประจำตัว 5033701123
อาจารย์ที่ปรึกษา	อาจารย์ ดร.ศุภกานต์ พิมลธเรศ		

ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับโครงการนี้เป็นส่วนหนึ่งของการศึกษาระดับปริญญาตรี ในรายวิชา 2301499 SENIOR PROJECT

(ศาสตราจารย์ ดร. กฤษณะ เนียมมณี)
หัวหน้าภาควิชาคณิตศาสตร์

(อาจารย์ ดร.ศุภกานต์ พิมลธเรศ)
อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร. วัชรินทร์ วิชิรมาลา)
กรรมการสอบ

(อาจารย์ ครรชิต จามรมาร)
กรรมการสอบ

บทคัดย่อ

ในปัจจุบันอุปกรณ์อิเล็กทรอนิกส์ได้มีการพัฒนาอย่างรวดเร็วตามเทคโนโลยีคอมพิวเตอร์ เมาส์จัดว่าเป็นอุปกรณ์หลักตัวหนึ่งที่สำคัญที่มีข้อจำกัดในการควบคุมด้วยมือเท่านั้น โครงการนี้มีจุดประสงค์เพื่อพัฒนาซอฟต์แวร์ที่สามารถควบคุมเมาส์ได้โดยการใช้กล้องเว็บแคมจับการเคลื่อนไหวของจุดกึ่งกลางระหว่างดวงตาทั้งสองข้างแทนการควบคุมด้วยมือดังเช่นเมาส์ในปัจจุบัน หลักการที่สำคัญคือการกำหนดให้ทิศทางการเคลื่อนที่ของเมาส์สัมพันธ์กับตำแหน่งในการมองผ่านกล้องเว็บแคม ส่วนการคลิกเมาส์นั้นทำได้โดยการหลับตาข้างที่สัมพันธ์กับการคลิกเมาส์ที่ต้องการค้างไว้ภายในระยะเวลาที่กำหนด ซึ่งเป็นการใช้เทคนิคการประมวลผลภาพดิจิทัลมาใช้ในการตรวจจับและติดตามดวงตา โดยวิธีดังกล่าวนี้เป็นวิธีที่สะดวกและง่ายต่อผู้ใช้งาน เนื่องจากอุปกรณ์ที่จำเป็นต้องใช้เพิ่มนอกเหนือจากเครื่องคอมพิวเตอร์และซอฟต์แวร์ มีเพียงกล้องเว็บแคมเท่านั้น ทำให้ใช้ต้นทุนในการผลิตต่ำ อีกทั้งยังเป็นประโยชน์สำหรับผู้พิการทางแขนและขาที่ไม่สามารถใช้งานเครื่องคอมพิวเตอร์ได้อย่างบุคคลทั่วไป

Abstract

Nowadays, electronic devices are rapidly developed according to computer technology. Mouse, a main computer device, has the limitation of only hand control. The objective of this project is to develop the software which allows user to locate mouse pointer position by tracking midpoint between two eyes via web camera instead of traditional device. Mouse pointer position movement will be related to the position where user is looking at whilst clicking action can be done by closing the eye with indicated period of time. The eye tracking part is built based on digital image processing in order to detect and track the eye movement. According to the above features of the software, this project method is designed to be convenient and ease for the user with lower production cost. The necessary additional equipment excluding computer and software is just only a web camera. With these reasons, this project will be helpful for the community especially the handicap person to operate the computer as normal person.

กิตติกรรมประกาศ

ขอขอบพระคุณอาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ คำชี้แนะ จนส่งผลให้ผู้จัดทำสามารถดำเนินโครงการนี้ให้สำเร็จลุล่วงไปได้ด้วยดี โดยเฉพาะอย่างยิ่ง ขอขอบพระคุณ อ. ดร. ศุภกานต์ พิมลธเรศ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการ และได้กรุณาถ่ายทอดความรู้ ให้การดูแล เอาใจใส่ ควบคุมให้การทำงานดำเนินไปอย่างเป็นขั้นเป็นตอน อีกทั้งยังให้คำแนะนำที่เป็นประโยชน์อย่างยิ่งแก่ผู้จัดทำมาตลอดตั้งแต่เริ่มต้นโครงการจนกระทั่งโครงการชิ้นนี้สำเร็จตามวัตถุประสงค์ที่ได้ตั้งไว้

นอกจากนี้ผู้จัดทำยังได้รับความเอื้อเฟื้อและความช่วยเหลือจากบุคคลต่างๆ ดังนี้

- อ. ครรชิต จามรมาร และผศ. ดร. วัชรินทร์ วิจิรมาลา คณะกรรมการสอบโครงการทั้งสองท่าน ที่ให้ข้อคิดเห็นต่างๆ ในการพัฒนา และปรับปรุงโครงการ
- คุณดรุณี สว่างดี ที่คอยอำนวยความสะดวกในการเข้าใช้เครื่องคอมพิวเตอร์ของภาควิชาคณิตศาสตร์
- คุณพ่อ และคุณแม่ สำหรับการสนับสนุน และอนุเคราะห์อุปกรณ์คอมพิวเตอร์ต่างๆ ซึ่งส่งผลให้โครงการนี้สามารถดำเนินต่อไปและสำเร็จได้ด้วยดี
- เพื่อนๆ ทุกคน สำหรับน้ำใจ ความห่วงใย และกำลังใจที่ให้กันเสมอมา

ผู้จัดทำรู้สึกซาบซึ้งและขอขอบพระคุณทุกท่านที่กล่าวมาในข้างต้นนี้เป็นอย่างยิ่ง ที่ทำให้โครงการนี้ประสบความสำเร็จ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ค
บทคัดย่อภาษาอังกฤษ	ง
กิตติกรรมประกาศ	จ
สารบัญ	ฉ
สารบัญรูป	ช
บทที่ 1 บทนำ	
1.1 หลักการและเหตุผล	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของโครงการ	2
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง	
2.1 OpenCV	4
2.2 การประมวลผลภาพดิจิทัล	5
2.3 Haar like-feature	7
2.4 การหาบริเวณที่สนใจ	8
2.5 การรู้จำ	8
2.6 การทำอ็ควอไลซ์ของฮิสโทแกรม	11
2.7 กระบวนการมอร์โฟโลยี	12
2.8 งานวิจัยที่เกี่ยวข้อง	15
บทที่ 3 การออกแบบโปรแกรมและขั้นตอนวิธี	
3.1 ขั้นตอนวิธีของโปรแกรมโดยรวม	16
3.2 ไลบรารีที่ใช้ในการพัฒนาโปรแกรม	17
3.3 การรับภาพเข้าโปรแกรม	18
3.4 การตรวจจับดวงตา	20
3.5 การติดตามดวงตา	21
3.6 การควบคุมเมาส์	22
3.7 การคลิกเมาส์	23
บทที่ 4 การพัฒนาและทดสอบระบบ	
4.1 การพัฒนาโปรแกรม	24
4.2 การทดสอบโปรแกรม	29
บทที่ 5 สรุปผลและข้อเสนอแนะ	
5.1 ผลที่ได้รับ	34
5.2 ปัญหาและอุปสรรค	34
5.3 การแก้ไขปัญหา	34

5.4 ข้อเสนอแนะ	35
บรรณานุกรม	36
ภาคผนวก ก แบบเสนอหัวข้อโครงการ Project Proposal	37
ภาคผนวก ข คู่มือการติดตั้งโปรแกรมที่ใช้ในการพัฒนา	41
ภาคผนวก ค วิธีการ compile OpenCV ด้วย CMake 2.8	53
ภาคผนวก ง วิธีการตั้งค่าให้กับ Microsoft Visual Studio 2008 และการสร้างโปรเจคใหม่	64
ภาคผนวก จ คู่มือการติดตั้งโปรแกรม Eye Tracking Mouse เพื่อใช้งานจริง	72

สารบัญรูป

	หน้า
รูปที่ 2.1 แสดงความแตกต่างระหว่างภาพแอนะล็อก (ซ้าย) กับภาพดิจิทัล (ขวา)	5
รูปที่ 2.2 ภาพดิจิทัลในระบบพิกัดระนาบ 2 มิติ	6
รูปที่ 2.3 รูปแบบของ Rectangle regions สำหรับการตรวจจับลักษณะแบบต่างๆ	7
รูปที่ 2.4 ตัวอย่างการกำหนดบริเวณที่สนใจ	8
รูปที่ 2.5 การทำการเทียบแผ่นแบบ	9
รูปที่ 2.6 Neural network อย่างง่าย	10
รูปที่ 2.7 เปรียบเทียบระหว่างภาพดั้งเดิมกับภาพที่ทำอควไลซ์ของฮิสโทแกรม	11
รูปที่ 2.8 ตัวอย่างส่วนย่อยโครงสร้างจุดภาพดำ S ขนาด 3X3 จุดภาพ	12
รูปที่ 2.9 การทำการพองภาพ $B \oplus S$	13
รูปที่ 2.10 การทำการกร่อนภาพ $B \ominus S$	13
รูปที่ 2.11 การทำการปิด $B \bullet S$	14
รูปที่ 2.12 การทำการเปิด $B \circ S$	14
รูปที่ 3.1 ผังการทำงานของโปรแกรมโดยรวม	16
รูปที่ 3.2 กล้องเว็บแคม Microsoft LifeCam VX-1000	18
รูปที่ 3.3 ภาพใบหน้าตรง	19
รูปที่ 3.4 ภาพการเลื่อนตำแหน่งของใบหน้า	19
รูปที่ 3.5 ภาพการตรวจจับดวงตา	20
รูปที่ 3.6 ภาพการติดตามดวงตาในบริเวณที่ใกล้เคียงตำแหน่งเดิมของดวงตา	21
รูปที่ 3.7 ภาพแสดงการหาจุดกึ่งกลางระหว่างดวงตาข้างซ้าย และข้างขวา	22
รูปที่ 4.1 ภาพหนึ่งเฟรม	24
รูปที่ 4.2 ภาพแสดงการสร้างกรอบบริเวณดวงตา	25
รูปที่ 4.3 ภาพแสดงจุดกึ่งกลางระหว่างดวงตา	25
รูปที่ 4.4 ภาพการสร้างกรอบใหญ่รอบกรอบบริเวณดวงตาที่ได้ถูกตรวจจับในตอนแรก	26
รูปที่ 4.5 ภาพแผ่นแบบของดวงตาที่ได้จากการตรวจจับดวงตาบนเฟรมภาพ	26
รูปที่ 4.6 ภาพแผ่นแบบของดวงตาหลังจากการแปลงเป็นภาพระดับสีเทา	27
รูปที่ 4.7 ภาพแผ่นแบบของดวงตาหลังจากที่ทำอควไลซ์ของฮิสโทแกรม	27
รูปที่ 4.8 ภาพแผ่นแบบของดวงตาหลังจากแปลงภาพด้วยค่าขีดแบ่ง	27
รูปที่ 4.9 ภาพแผ่นแบบของดวงตาหลังจากทำการกร่อนภาพ	28
รูปที่ 4.10 ภาพแผ่นแบบของดวงตาหลังจากทำการปิดครั้งที่หนึ่ง	28
รูปที่ 4.11 ภาพแผ่นแบบของดวงตาหลังจากทำการปิดครั้งที่สอง	28
รูปที่ 4.12 หน้าต่างของโปรแกรม	29
รูปที่ 4.13 การตรวจจับบริเวณดวงตาของโปรแกรม	29
รูปที่ 4.14 การเลื่อนเมาส์ไปทางซ้ายบนของโปรแกรม	30
รูปที่ 4.15 การเลื่อนเมาส์ไปทางขวาของโปรแกรม	30

รูปที่ 4.16 การเลื่อนเมาส์ไปทางซ้ายล่างของโปรแกรม	31
รูปที่ 4.17 การเลื่อนเมาส์ไปทางขวาล่างของโปรแกรม	31
รูปที่ 4.18 การคลิกเมาส์ซ้ายของโปรแกรม	32
รูปที่ 4.19 การคลิกเมาส์ขวาของโปรแกรม	32
รูปที่ 4.20 การดับเบิลคลิกของโปรแกรม	33
รูปที่ 4.21 การใช้งานโปรแกรมในการพิมพ์อักษรของโปรแกรม	33
รูปภาคผนวก ข-1 หน้าต่างเริ่มต้นการติดตั้งโปรแกรม Microsoft Visual Studio 2008	41
รูปภาคผนวก ข-2 หน้าต่างต้อนรับการติดตั้งโปรแกรม Microsoft Visual Studio 2008	42
รูปภาคผนวก ข-3 หน้าต่างแสดงข้อตกลงต่างๆของการติดตั้งโปรแกรม Microsoft Visual Studio 2008	42
รูปภาคผนวก ข-4 หน้าต่างแสดงรูปแบบการติดตั้งโปรแกรม Microsoft Visual Studio 2008	43
รูปภาคผนวก ข-5 หน้าต่างแสดงว่ากำลังติดตั้งโปรแกรม Microsoft Visual Studio 2008	43
รูปภาคผนวก ข-6 หน้าต่างแสดงว่าโปรแกรม Microsoft Visual Studio 2008 ติดตั้งสมบูรณ์	44
รูปภาคผนวก ข-7 ไอคอนตัวติดตั้งของ OpenCV	44
รูปภาคผนวก ข-8 หน้าต่างเริ่มต้นของการติดตั้ง OpenCV	45
รูปภาคผนวก ข-9 หน้าต่างแสดงข้อตกลงต่างๆของการติดตั้งโปรแกรม	45
รูปภาคผนวก ข-10 หน้าต่างตัวเลือกการติดตั้ง OpenCV	46
รูปภาคผนวก ข-11 หน้าต่างเลือกตำแหน่งการติดตั้ง OpenCV	46
รูปภาคผนวก ข-12 หน้าต่างเลือกรูปแบบการติดตั้ง OpenCV	47
รูปภาคผนวก ข-13 หน้าต่างแสดงว่ากำลังการติดตั้ง OpenCV	47
รูปภาคผนวก ข-14 หน้าต่างแสดงว่าการติดตั้ง OpenCV เสร็จสมบูรณ์	48
รูปภาคผนวก ข-15 ไอคอนตัวติดตั้งของ Cmake 2.8	48
รูปภาคผนวก ข-16 หน้าต่างเริ่มต้นของการติดตั้ง Cmake 2.8	49
รูปภาคผนวก ข-17 หน้าต่างแสดงข้อตกลงต่างๆของการติดตั้งโปรแกรม	49
รูปภาคผนวก ข-18 หน้าต่างตัวเลือกการติดตั้ง Cmake 2.8	50
รูปภาคผนวก ข-19 หน้าต่างเลือกตำแหน่งการติดตั้ง Cmake 2.8	50
รูปภาคผนวก ข-20 หน้าต่างการติดตั้ง Cmake 2.8	51
รูปภาคผนวก ข-21 หน้าต่างแสดงว่ากำลังการติดตั้ง Cmake 2.8	51
รูปภาคผนวก ข-22 หน้าต่างแสดงว่าการติดตั้ง Cmake 2.8 เสร็จสมบูรณ์	52
รูปภาคผนวก ค-1 ลักษณะโปรแกรม CMake 2.8	53
รูปภาคผนวก ค-2 การกำหนดตำแหน่ง source code และ build binaries ในโปรแกรม CMake 2.8	54
รูปภาคผนวก ค-3 การถามเพื่อยืนยันการสร้างที่อยู่ใหม่ในโปรแกรม CMake 2.8	54
รูปภาคผนวก ค-4 การเลือกตัวคอมไพเลอร์และรูปแบบการคอมไพล์ในโปรแกรม CMake 2.8	55
รูปภาคผนวก ค-5 การเลือกไฟล์ที่จะสร้างขึ้นมาโดยใช้โปรแกรม CMake 2.8	55
รูปภาคผนวก ค-6 การสร้างไฟล์ต่างๆโดยใช้โปรแกรม CMake 2.8	56
รูปภาคผนวก ค-7 การสร้างไฟล์ต่างๆโดยใช้โปรแกรม CMake 2.8	56
รูปภาคผนวก ค-8 ไฟล์ OpenCV ที่ได้จากการ build binaries	57
รูปภาคผนวก ค-9 การเปิดไฟล์ OpenCV ด้วยโปรแกรม Microsoft Visual studio 2008	57
รูปภาคผนวก ค-10 การไปหน้า Configuration Manager ในโปรแกรม Microsoft Visual studio 2008	58

รูปภาคผนวก ค-11 การตั้งค่า Configuration Manager สำหรับรันโหมด Debug	58
รูปภาคผนวก ค-12 การสร้างไฟล์ในการรันโหมด Debug	59
รูปภาคผนวก ค-13 แสดงการสร้างไฟล์ในโหมด Debug เสร็จเรียบร้อยแล้ว	59
รูปภาคผนวก ค-14 การตั้งค่า Configuration Manager สำหรับรันโหมด Release	60
รูปภาคผนวก ค-15 การสร้างไฟล์สำหรับการรันโหมด Release	60
รูปภาคผนวก ค-16 แสดงการสร้างไฟล์ในโหมด Release เสร็จเรียบร้อยแล้ว	61
รูปภาคผนวก ค-17 แสดงการเข้าสู่ Properties ของ My Computer	61
รูปภาคผนวก ค-18 แสดงการเข้าสู่หน้าต่าง Environment Variables ของ My Computer	62
รูปภาคผนวก ค-19 แสดงการเข้าสู่การแก้ไข Path ในหน้าต่าง Environment Variables	62
รูปภาคผนวก ค-20 แสดงหน้าต่าง Edit System Variables	63
รูปภาคผนวก ค-21 แสดงการเพิ่ม Path ในหน้าต่าง Edit System Variables	63
รูปภาคผนวก ง-1 แสดงการเข้าสู่หน้าต่าง Options ในโปรแกรม Microsoft Visual Studio 2008	64
รูปภาคผนวก ง-2 แสดงการเลือก VC++ Directories ในโปรแกรม Microsoft Visual Studio 2008	65
รูปภาคผนวก ง-3 แสดงการกำหนด Path ของไฟล์ Include จาก OpenCV	65
รูปภาคผนวก ง-4 แสดงการกำหนด Path ของไฟล์ Libraries สำหรับโหมด Debug จาก OpenCV	66
รูปภาคผนวก ง-5 แสดงการกำหนด Path ของไฟล์ Libraries สำหรับโหมด Release จาก OpenCV	66
รูปภาคผนวก ง-6 แสดงการสร้างโปรเจกต์ใหม่ในโปรแกรม Microsoft Visual Studio 2008	67
รูปภาคผนวก ง-7 แสดงการเลือกรูปแบบโปรเจกต์ในโปรแกรม Microsoft Visual Studio 2008	67
รูปภาคผนวก ง-8 หน้าต่าง Win32 Application Wizard ในโปรแกรม Microsoft Visual Studio 2008	68
รูปภาคผนวก ง-9 การเลือก Application type และ Additional options	68
รูปภาคผนวก ง-10 การเข้าสู่การกำหนดคุณสมบัติโปรเจกต์	69
รูปภาคผนวก ง-11 การกำหนด Additional Dependencies ไฟล์ สำหรับโหมด Debug	69
รูปภาคผนวก ง-12 การกำหนด Additional Dependencies ไฟล์ สำหรับโหมด Release	70
รูปภาคผนวก ง-13 การเพิ่มไฟล์ใหม่ในโปรแกรม Microsoft Visual Studio 2008	70
รูปภาคผนวก ง-14 การกำหนดประเภทไฟล์ของโปรเจกต์ในโปรแกรม Microsoft Visual Studio 2008	71
รูปภาคผนวก ง-15 หน้าต่างแสดงการเริ่มต้นเขียนโปรแกรม	71
รูปภาคผนวก จ-1 ไอคอนตัวติดตั้งของโปรแกรม Eye Tracking Mouse	72
รูปภาคผนวก จ-2 หน้าต่างแสดงการต้อนรับของโปรแกรม Eye Tracking Mouse	72
รูปภาคผนวก จ-3 หน้าต่างเลือกตำแหน่งการติดตั้ง Eye Tracking Mouse	73
รูปภาคผนวก จ-4 หน้าต่างยืนยันการติดตั้ง Eye Tracking Mouse	73
รูปภาคผนวก จ-5 หน้าต่างแสดงการติดตั้ง Eye Tracking Mouse สมบูรณ์	74
รูปภาคผนวก จ-6 ไอคอนโปรแกรม Eye Tracking Mouse	74

บทที่ 1

บทนำ

1.1 หลักการและเหตุผล

ในปัจจุบันนี้เครื่องคอมพิวเตอร์ได้กลายเป็นอุปกรณ์เทคโนโลยีที่มีการใช้งานกันอย่างแพร่หลาย และเข้ามามีบทบาทในชีวิตประจำวันเป็นอย่างมาก ไม่ว่าจะเป็นทั้งในองค์กรของรัฐบาล หรือเอกชน รวมไปถึงการใช้งานทั่วไปของส่วนบุคคล เนื่องจากเครื่องคอมพิวเตอร์เป็น เครื่องอิเล็กทรอนิกส์แบบอัตโนมัติ ทำหน้าที่เสมือนสมองกลใช้สำหรับแก้ปัญหาต่าง ๆ ทั้งที่ง่ายและซับซ้อนโดยวิธีทางคณิตศาสตร์ (จากพจนานุกรม ฉบับราชบัณฑิตยสถาน พ.ศ. 2542) [1] จึงเป็นเหตุให้เครื่องคอมพิวเตอร์สามารถให้ความช่วยเหลือ และอำนวยความสะดวกในการทำงานได้เป็นอย่างดี

เครื่องคอมพิวเตอร์ที่ใช้งานกันทั่วไปนั้น มักออกแบบมาให้ผู้ใช้สั่งงานโดยการควบคุมทิศทางของเมาส์ด้วยมือเป็นหลัก หากผู้ใช้งานมีความผิดปกติ ไม่สามารถใช้งานมือและแขนได้ก็จะไม่สามารถใช้งานเครื่องคอมพิวเตอร์ได้ ทำให้กลุ่มผู้พิการทั้งแขนและขาสูญเสียโอกาสในการเรียนรู้และใช้งานเครื่องคอมพิวเตอร์ ไม่สามารถใช้บริการเทคโนโลยีสารสนเทศ เช่น การทำธุรกรรมบนอินเทอร์เน็ต การลงทะเบียนเรียนบน-อินเทอร์เน็ต และการทำงานอื่นๆ ที่ต้องใช้เครื่องคอมพิวเตอร์เป็นอุปกรณ์ช่วย

จากที่กล่าวมาข้างต้นส่งผลให้เกิดข้อจำกัดในการใช้งานเครื่องคอมพิวเตอร์สำหรับผู้พิการ ทำให้ผู้ดำเนินการมีแรงบันดาลใจในการพัฒนาซอฟต์แวร์ที่ทำให้สามารถควบคุมทิศทางของเมาส์ได้ ด้วยการใช้กล้องเว็บแคมจับการเคลื่อนไหวของจุดกึ่งกลางระหว่างดวงตาทั้งสองข้าง โดยกำหนดให้ทิศทางการเคลื่อนที่ของเมาส์มีความสัมพันธ์กับตำแหน่งในการมองผ่านกล้องเว็บแคม และการคลิกเมาส์นั้นทำโดยการหลับตาข้างที่สัมพันธ์กับการคลิกเมาส์ที่ต้องการค้างไว้ภายในระยะเวลาที่กำหนด เช่น ใช้การหลับตาขวาในการคลิกเมาส์ขวา การคลิกเมาส์ซ้ายก็เช่นเดียวกัน ส่วนการดับเบิลคลิกจะใช้การหลับตาทั้งสองข้างพร้อมกัน ซึ่งเป็นวิธีที่สะดวกสบาย และง่ายต่อผู้ใช้งาน เนื่องจากอุปกรณ์ที่จำเป็นต้องใช้เพิ่มนอกเหนือจากเครื่องคอมพิวเตอร์และซอฟต์แวร์ มีเพียงกล้องเว็บแคมเท่านั้น ทำให้ใช้ต้นทุนในการผลิตต่ำอีกด้วย

1.2 วัตถุประสงค์

เพื่อพัฒนาซอฟต์แวร์ที่ใช้ในการควบคุมเมาส์โดยใช้กล้องเว็บแคมจับการเคลื่อนไหวของจุดกึ่งกลางระหว่างดวงตาทั้งสองข้าง

1.3 ขอบเขตของโครงการ

1. ตรวจสอบตำแหน่งของดวงตาทั้งสองข้าง และหาค่าเฉลี่ยให้ได้จุดกึ่งกลางระหว่างดวงตา
2. ติดตามทิศทางการเคลื่อนไหวของจุดกึ่งกลางระหว่างดวงตาทั้งสองข้างในขณะที่ใบหน้ามีการเคลื่อนไหวน้อยที่สุด
3. สามารถคลิกเมาส์ได้เพียงทีละครั้งเท่านั้นไม่สามารถคลิกเมาส์ค้างได้
4. ประสานการทำงานระหว่างทิศทางการเคลื่อนไหวของจุดกึ่งกลางระหว่างดวงตาทั้งสองข้างกับทิศทางของเมาส์
5. ตรวจจับดวงตาได้ครั้งละหนึ่งใบหน้าเท่านั้น
6. ใบหน้าผู้ใช้งานต้องไม่สวมแว่นตาทุกชนิด รวมถึงไม่มีสิ่งใดบดบังดวงตา
7. ใช้ Resolution ของภาพเพียงค่าเดียว
8. ผู้ใช้ต้องสามารถหลับตาข้างเดียวได้
9. เพื่อให้การควบคุมเมาส์มีประสิทธิภาพมากขึ้นควรตั้งค่าตัวอักษรหรือปุ่มต่างๆ บนหน้าจอ Desktop ให้มีขนาดใหญ่

1.4 ขั้นตอนการดำเนินการ

1. ระบุหัวข้อโครงการและศึกษาทฤษฎีที่เกี่ยวข้อง
2. ออกแบบโครงสร้างโปรแกรม และเลือกอุปกรณ์และเทคโนโลยีที่ใช้
3. ศึกษาลักษณะการเคลื่อนที่ของดวงตามนุษย์
4. พัฒนาโปรแกรมส่วนการรับอินพุตจากกล้องเว็บแคม
5. พัฒนาโปรแกรมตรวจจับดวงตามนุษย์ และหาค่าเฉลี่ยให้ได้จุดกึ่งกลางระหว่างดวงตา
6. พัฒนาโปรแกรมติดตามการเคลื่อนไหวของจุดกึ่งกลางระหว่างดวงตาทั้งสองข้าง
7. เชื่อมต่อการทำงานระหว่างทิศทางการเคลื่อนไหวของจุดกึ่งกลางระหว่างดวงตาทั้งสองข้างให้สอดคล้องกับทิศทางของเมาส์
8. พัฒนาโปรแกรมส่วนการคลิกเมาส์
9. ทดสอบและแก้ไขข้อผิดพลาด
10. สรุปผลการทดลอง
11. จัดทำเอกสารประกอบโครงการ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้เรียนรู้เทคนิคและขั้นตอนวิธีการตรวจจับดวงตามนุษย์
2. ได้เรียนรู้เทคนิคและขั้นตอนวิธีการติดตามจุดกึ่งกลางระหว่างดวงตาทั้งสองข้าง
3. ได้เรียนรู้เทคนิคการประยุกต์ใช้แหล่งรวมชุดคำสั่ง (Library) มาตรฐานต่าง ๆ ที่มีอยู่
4. ได้สร้างโอกาส (ทางเลือก) ให้แก่ผู้พิการทางแขน ที่มีความต้องการจะใช้งานเครื่องคอมพิวเตอร์

บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 OpenCV (Open source Computer Vision)

OpenCV คือ ไลบรารีที่เป็น open source ซึ่งพัฒนาโดยบริษัท Intel ใช้สำหรับการพัฒนาโปรแกรมภาษา C, C++ และ Python ที่เกี่ยวข้องกับการประมวลผลภาพ (Image Processing) และคอมพิวเตอร์วิทัศน์ (Computer Vision) สามารถประมวลผลภาพดิจิทัล (Digital) ได้ทั้งภาพนิ่ง และภาพเคลื่อนไหว มีฟังก์ชันสำเร็จรูปสำหรับการจัดการข้อมูลภาพ และการประมวลผลภาพพื้นฐาน เช่น การหาขอบภาพ การกรองข้อมูลภาพ การตรวจจับลักษณะเด่นบนใบหน้า โดยมีวัตถุประสงค์หลักเพื่อการทำงานได้แบบทันกาล (real time)

การใช้งาน OpenCV นั้นใช้งานร่วมกับ Microsoft Visual Studio และเนื่องจาก OpenCV เป็นเพียงชุดคำสั่ง ไม่ใช่เป็นตัวโปรแกรม จึงจำเป็นต้องมีการตั้งค่าให้กับโปรแกรม Microsoft Visual Studio ก่อนเพื่อระบุตำแหน่งไลบรารีของ OpenCV และตำแหน่งของไฟล์ที่ต้องใช้ในโปรแกรม เพื่อให้โปรแกรม Microsoft Visual Studio สามารถดึงไลบรารีของ OpenCV มาใช้งานได้ โดย OpenCV ประกอบด้วยไลบรารีทั้งหมด 4 ประเภทดังนี้

1. CV

ใช้ในการประมวลผลและการวิเคราะห์รูปภาพ ฟังก์ชันส่วนใหญ่จะทำงานกับจุดภาพที่เป็นอาร์เรย์สองมิติ หรือที่เรียกว่าภาพนั่นเอง เช่น การหาขอบหรือมุม การทำฮิสโตแกรม (Histogram) เป็นต้น

2. CXCORE

เป็นฟังก์ชันเบื้องต้นที่ใช้จัดการเกี่ยวกับจุด ขนาด อาร์เรย์ หน่วยความจำ คำสั่งในการวาดภาพ การประกาศตัวแปรภาพ เป็นต้น ตัวอย่างคำสั่งในการประกาศรูปภาพ คือ IplImage, CvMat, CvMatND

3. HighGUI

เป็นไลบรารีที่ใช้ในการดึงภาพ การบันทึกภาพ การติดต่อกับกล้องวิดีโอ (VDO) การสร้างหน้าต่างเพื่อแสดงภาพและทำลายภาพ การเปลี่ยนขนาดและเคลื่อนย้ายหน้าต่าง รวมไปถึงการตรวจสอบเมาส์ (Mouse) และแป้นพิมพ์

4. ML (Machine learning)

เป็นไลบรารีที่รวมคลาสและฟังก์ชันทางสถิติ (Statistics) การแยกคลาสและการแบ่งกลุ่มข้อมูล (Clustering)

2.2 การประมวลผลภาพเชิงดิจิทัล (Digital Image Processing)

การประมวลผลภาพเชิงดิจิทัลคือการปรับปรุงและนำข้อมูลภาพดิจิทัลที่ได้มาใช้ให้เป็นประโยชน์ในกระบวนการจัดเก็บภาพ และการส่งผ่านสัญญาณภาพ เป็นการประยุกต์การประมวลผลข้อมูลแบบสัญญาณ (Signal processing) และขั้นตอนวิธีต่างๆ มากระทำบนภาพดิจิทัล เพื่อให้ได้ข้อมูลที่เป็นประโยชน์สำหรับการประมวลผล หรือวิเคราะห์ต่อไป

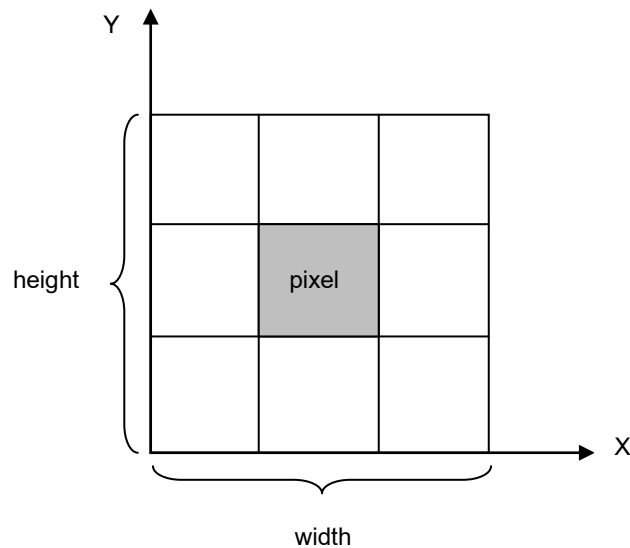
เนื่องจากภาพที่มองเห็นโดยปกติผ่านสายตามนุษย์นั้นจะมองเห็นภาพในลักษณะที่เรียกว่า ภาพแอนะล็อก (Analog) ซึ่งอธิบายในทางคณิตศาสตร์ได้ด้วยตัวแปรค่าต่อเนื่อง (Continuous variables) แต่เครื่องคอมพิวเตอร์นั้นจะใช้เลขฐาน 2 ซึ่งเป็นตัวแปรค่าไม่ต่อเนื่อง (Discrete variables) เป็นหลักในการคำนวณ โดยภาพนั้นจะถูกแทนด้วยตัวเลขในรูปเมตริกซ์ ทำให้ภาพที่นำมาแปลงเข้าสู่คอมพิวเตอร์กลายเป็นภาพดิจิทัลที่มีจำนวนสีและค่าความเปรียบต่างของภาพ (Contrast) มากกว่าภาพแอนะล็อก ภาพดิจิทัลจึงสามารถแสดงให้เห็นถึงรายละเอียดของภาพได้อย่างชัดเจน ทำให้สามารถนำไปประมวลผล หรือปรับปรุงข้อมูลภาพโดยใช้เทคนิคต่างๆ ผ่านคอมพิวเตอร์ได้ง่ายกว่าภาพแอนะล็อก



รูป 2.1 แสดงความแตกต่างระหว่างภาพแอนะล็อก (ซ้าย) กับภาพดิจิทัล (ขวา)

ที่มา : <http://www.ncgicc.com/>

การแสดงผลภาพดิจิทัลในคอมพิวเตอร์นั้นใช้ระบบพิกัดเชิงพื้นที่ (Spatial coordinate) บนระนาบ 2 มิติ โดยใช้ แกน X และแกน Y บ่งบอกถึงความกว้างและความยาวของภาพ ส่วนจุดใดๆ ที่อยู่บนระนาบ XY จะเรียกแทนพิกัดนั้นๆ ว่าจุดภาพ (pixel)



รูป 2.2 ภาพดิจิทัลในระบบพิกัดระนาบ 2 มิติ

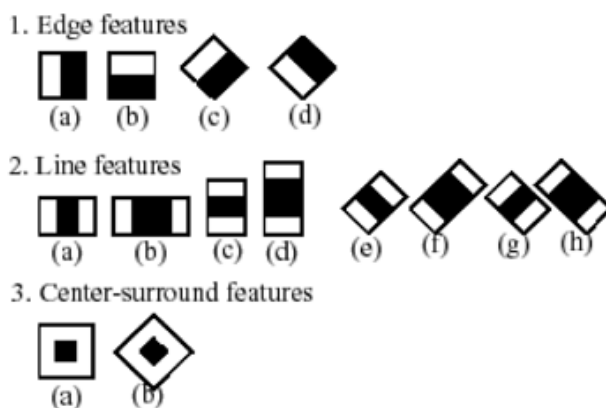
ในทางคณิตศาสตร์อาจอธิบายนิยามภาพดิจิทัลในรูปของฟังก์ชัน 2 มิติ $f(x, y)$ โดยที่ x และ y เป็นพิกัดเชิงพื้นที่ของภาพและขนาดของ f ที่พิกัด (x, y) ใดๆ ภายในภาพคือค่าความเข้มแสงของภาพ (intensity) ที่ตำแหน่งนั้นๆ เมื่อ x, y และขนาดของ f เป็นค่าจำกัด (finite value) ถ้ากำหนดให้ภาพ $f(x, y)$ มีขนาด M แถว และ N คอลัมน์ และพิกัดของจุดกำเนิด (origin) ของภาพคือที่ตำแหน่ง $(x, y) = (0, 0)$ แล้ว จะสามารถเขียนสมการให้อยู่ในรูปเมตริกซ์ได้ดังนี้

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}$$

การประมวลผลภาพเชิงดิจิทัลนั้นได้นำมาใช้ในงานต่างๆ ได้แก่ การหาจุดเด่นของภาพ (Feature extraction) เพื่อใช้เป็นข้อมูลในการวิเคราะห์และบ่งชี้ลักษณะเฉพาะของวัตถุในภาพ นอกจากนี้แล้วยังใช้ในการจัดแบ่งประเภทหมวดหมู่ (Classification) ของข้อมูลภาพว่าภาพนั้นจัดอยู่ในกลุ่มหรือคลาสใด ซึ่งสามารถทำได้โดยใช้ตัวแยกประเภท (Classifier) มาใช้งาน โดยตัวแยกประเภทนั้นสามารถสร้างได้จากหลายเทคนิค เช่น โครงข่ายประสาท (Neural network) และ ตัวจำแนกของฮาร์ (Haar Classifier)

2.3 Haar-like feature

Haar-like feature คือ ลักษณะบ่งชี้ที่นำไปใช้จำแนกและตรวจจับวัตถุภายในภาพ โดยการใช้การสร้างพื้นที่รูปเหลี่ยม (Rectangle regions) ที่แสดงถึงผลต่างระหว่างพื้นที่ส่วนสีขาว และส่วนที่เป็นสีดำ ซึ่งรูปเหลี่ยมที่สร้างขึ้นสามารถเปลี่ยนแปลงขนาด และตำแหน่งได้ โดยลักษณะการทำงานคือ พื้นที่รูปเหลี่ยมจะเคลื่อนที่ไปบริเวณส่วนต่างๆ ของภาพ เพื่อคำนวณหาค่าที่อยู่ในกรอบบริเวณนั้นออกมาและนำค่าลักษณะบ่งชี้ที่นั้นไปใช้คำนวณต่อไป

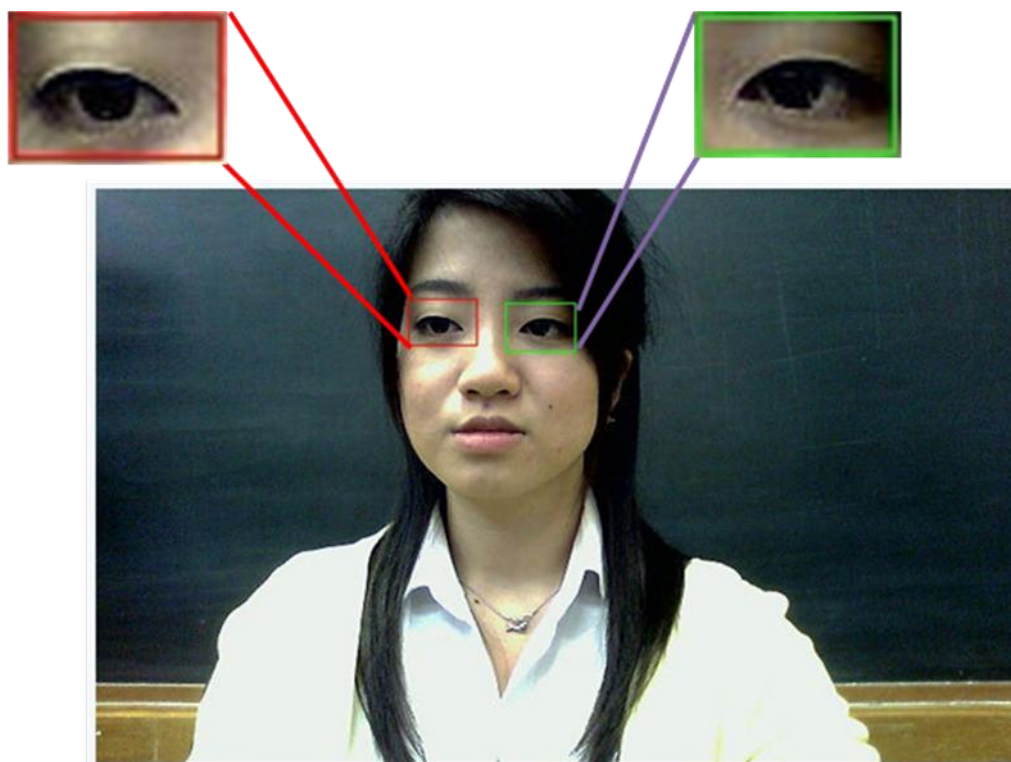


รูปที่ 2.3 รูปแบบของ *Rectangle regions* สำหรับการตรวจจับลักษณะแบบต่างๆ
 (1) ลักษณะเด่นแบบเส้นขอบ (2) ลักษณะเด่นแบบเส้น (3) ลักษณะเด่นแบบศูนย์กลางที่ล้อมรอบ

ที่มา : *Facial Feature Detection Using Haar Classifiers*

2.4 การหาบริเวณที่สนใจ (Region of interest)

Region of interest คือบริเวณที่สนใจ อาจจะเป็นบริเวณใดภายในภาพก็ได้ โดยการตีกรอบล้อมรอบบริเวณที่สนใจ ด้วยวงกลม กรอบสี่เหลี่ยม หรือกรอบรูปเหลี่ยมใดๆ เพื่อนำภาพเฉพาะส่วนดังกล่าวมาประมวลผล หรือเปลี่ยนแปลงภาพตามต้องการ โดยไม่มีผลกระทบกับส่วนอื่นๆ ซึ่งภายในหนึ่งภาพ สามารถกำหนดได้หลายๆ บริเวณที่สนใจ



รูปที่ 2.4 ตัวอย่างการกำหนดบริเวณที่สนใจ

2.5 การรู้จำ (Recognition)

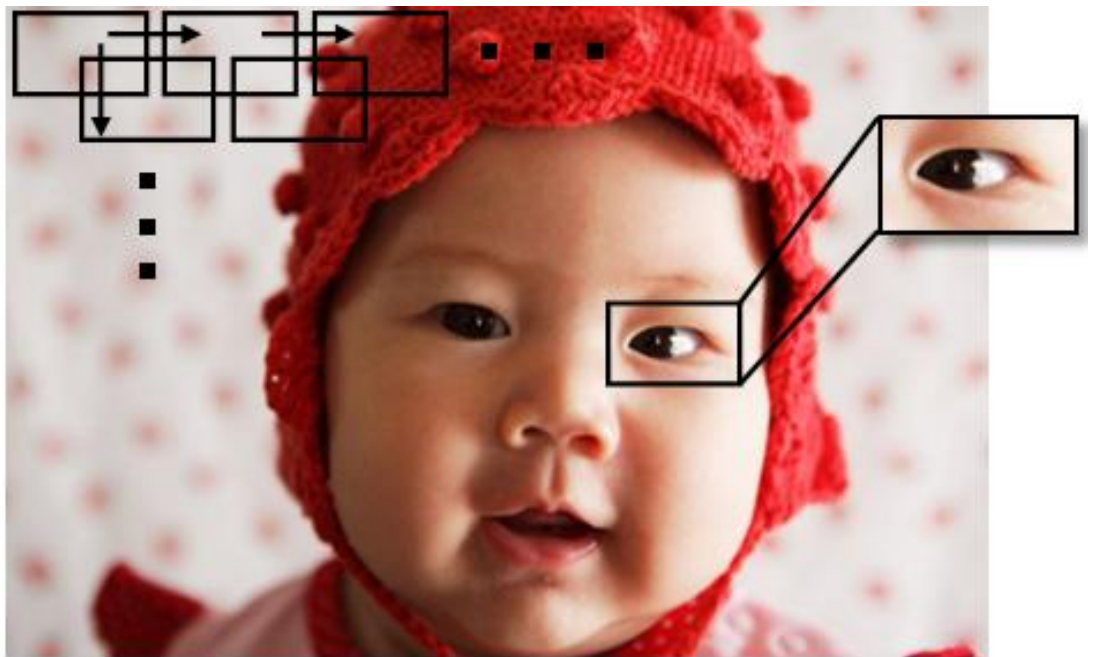
การรู้จำ คือการนำข้อมูลชุดใดชุดหนึ่งที่เรานำไปเปรียบเทียบกับข้อมูลอีกมากมายหลายชุด แล้วใช้กระบวนการเทคนิควิธีต่างๆ ในการค้นหาว่าข้อมูลชุดที่สนใจนั้น มีความคล้ายคลึงกับข้อมูลชุดใดในจำนวนชุดข้อมูลที่มากมายมหาศาลนั้น เพื่อนำไปประยุกต์ใช้ให้เกิดประโยชน์ต่างๆ ทั้งด้านความปลอดภัย หรือในด้านความสะดวกสบาย เช่น การรู้จำใบหน้า ลายมือ เพื่อบ่งชี้ความเป็นตัวบุคคล การรู้จำภาพภาษามือของผู้ที่เป็นใบ้ แล้วนำมาตีความเป็นภาษาพูด หรือ การรู้จำบางช่วงของเสียงเพลงที่ไม่รู้จัก เพื่อนำไปค้นหาชื่อเพลงนั้นได้อย่างง่ายดาย

เทคนิคของการรู้จำนั้นมีหลายวิธี ยกตัวอย่างดังนี้

1. การเทียบแผ่นแบบ (template matching)

เป็นวิธีที่ใช้ค้นหาบางส่วนเล็กๆ ของภาพที่มีความคล้ายคลึงกับ ภาพแผ่นแบบ (template) ที่เรานำมาเปรียบเทียบ โดยเลื่อนภาพแผ่นแบบบนภาพที่เราต้องการตรวจสอบจากด้านซ้ายบนไปยังด้านขวาล่างของภาพ และเปรียบเทียบหาบริเวณที่มีความคล้ายคลึงมากที่สุดกับภาพแผ่นแบบ โดยภาพแผ่นแบบควรจะเล็กกว่าหรือเท่ากับภาพที่เราค้นหา จึงเป็นวิธีเหมาะกับภาพที่มีขนาดไม่ใหญ่มาก เพราะใช้เวลาดำเนินการมาก

การเทียบแผ่นแบบนี้เป็นวิธีที่นิยมใช้ในการทำงานหลายประเภท เช่น การรู้จำใบหน้า และการติดตามวัตถุใดๆ บนภาพ



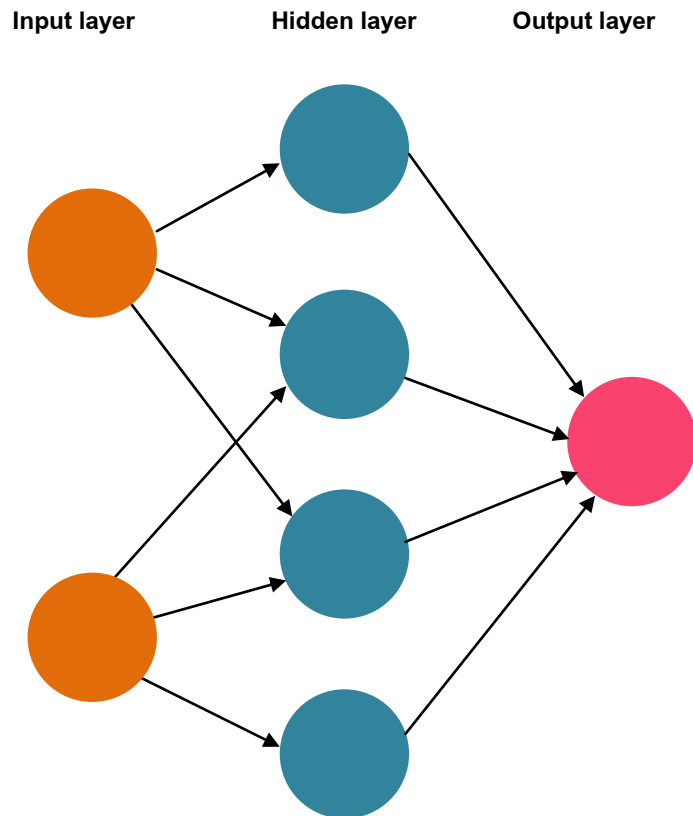
รูปที่ 2.5 การทำการเทียบแผ่นแบบ

2. วิธีการทางสถิติ (Statistical Approach)

เป็นวิธีที่ใช้หลักการทางสถิติ โดยนำค่าความน่าจะเป็นมาใช้ในการตัดสินใจ โดยใช้ลักษณะเด่นของภาพที่ต้องการนำมาค้นหาจากฐานข้อมูลภาพที่ได้มีการรู้จำลักษณะเด่นเอาไว้ทั้งหมดแล้ว และให้ภาพที่นำมาค้นหานั้นเปรียบเทียบกับทุกภาพในฐานข้อมูลเพื่อให้ได้ค่าความน่าจะเป็นของแต่ละภาพออกมา และเลือกภาพที่ได้ค่าความน่าจะเป็นมากที่สุด

3. โครงข่ายประสาท (Neural Network)

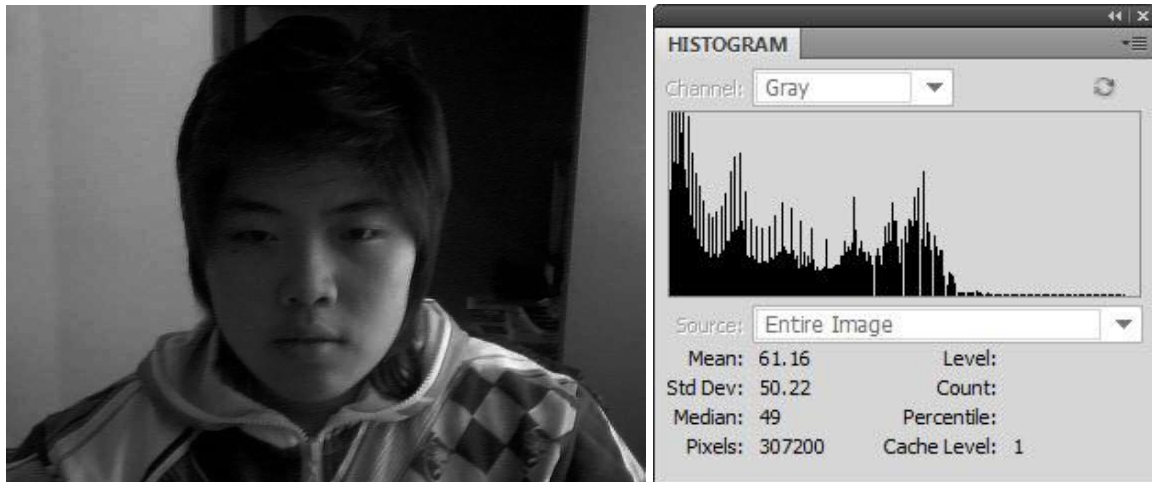
เป็นวิธีที่เลียนแบบการทำงานของสมองมนุษย์ที่มีลักษณะเป็นโครงข่ายเชื่อมต่อกันคล้ายหน่วยความจำย่อยๆ ที่สะสมความรู้เอาไว้ โดยความรู้เหล่านั้นจะได้รับการฝึกสอนไว้ก่อน ใช้ในการทำนายโดยการนำข้อมูลมาเรียนรู้ในจำนวนที่เหมาะสมเพื่อให้ได้โครงข่ายประสาทที่มีคุณภาพและทำนายได้แม่นยำ



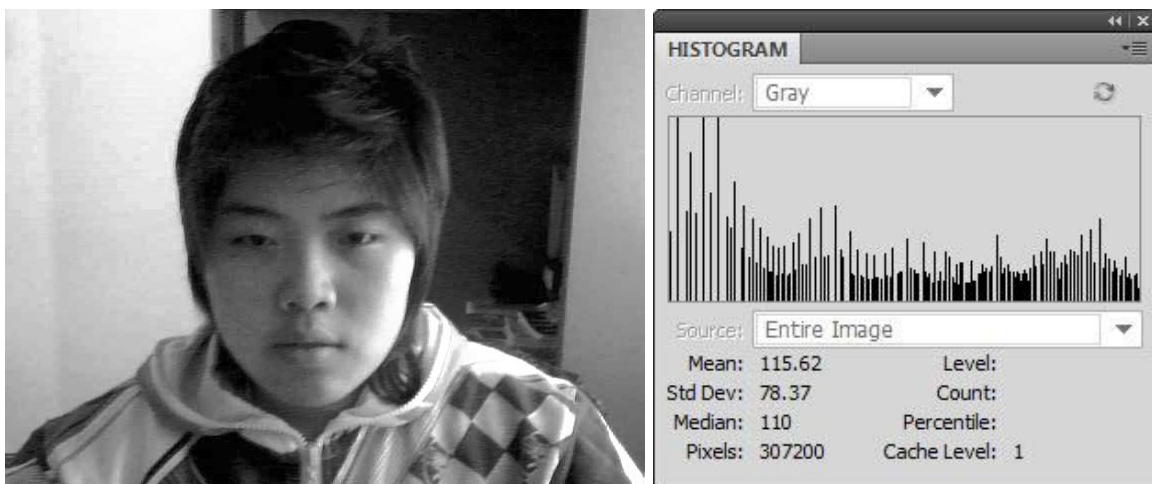
รูปที่ 2.6 Neural network อย่างง่าย

2.6 การทำอิกวไลซ์ของฮิสโทแกรม (Histogram Equalization)

เป็นวิธีการปรับปรุงภาพให้มีความเหมาะสมในแต่ละจุดภาพ ทำให้ค่าความเข้มของสีมีการกระจายอย่างสม่ำเสมอ เช่น หากภาพที่รับเข้ามามีมืดหรือสว่างมากเกินไป ก็จะทำการปรับภาพให้มีความสว่างที่เหมาะสม



(ก)



(ข)

รูปที่ 2.7 เปรียบเทียบระหว่างภาพดั้งเดิมกับภาพที่ทำอิกวไลซ์ของฮิสโทแกรม

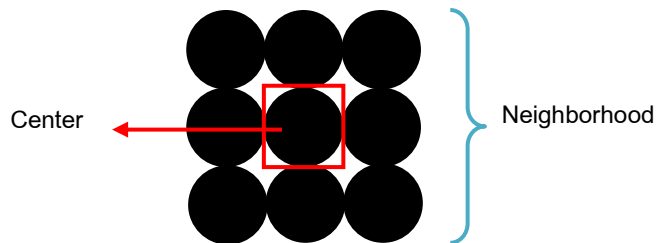
(ก) ภาพดั้งเดิม

(ข) ภาพที่ทำอิกวไลซ์ของฮิสโทแกรม

2.7 กระบวนการมอร์โฟโลยี (Morphological image processing)

ภาพของวัตถุที่นำมาประมวลผลด้วยเครื่องคอมพิวเตอร์นั้นมาจากหลากหลายวิธี อาจได้จากเครื่องสแกน หรือจากกล้องดิจิทัล ทำให้ภาพที่ได้มานั้นอาจมีสัญญาณรบกวน (noise) เกิดขึ้นซึ่งเป็นสิ่งที่ไม่น่าพึงประสงค์ จึงต้องมีวิธีจัดการกับสัญญาณรบกวนเหล่านั้น นั่นก็คือกระบวนการมอร์โฟโลยี

หลักการของกระบวนการมอร์โฟโลยีนั้นจะพิจารณาเฉพาะกลุ่มสีที่เป็นวัตถุในภาพตามแต่กำหนดไว้ ซึ่งอาจพิจารณาวัตถุสีดำบนพื้นหลังภาพสีขาว หรือวัตถุสีขาวบนพื้นหลังภาพสีดำ ซึ่งแต่ละกลุ่มสีของวัตถุนั้นอาจมีสัญญาณรบกวนเกิดขึ้น จึงต้องกำจัดสัญญาณรบกวนเหล่านั้นออกก่อน เพื่อนำภาพไปประมวลผลต่อไปได้อย่างสะดวก ในการพิจารณาการกำจัดสัญญาณรบกวนนั้น ต้องมีการกำหนดส่วนย่อยโครงสร้าง (Structure Elements) มาช่วยในการพิจารณา เช่น ส่วนย่อยโครงสร้างจุดภาพดำ S ที่มีขนาด 3×3 จุดภาพ จะนำจุดกลาง (Center) และจุดข้างเคียง (Neighborhood) ของส่วนย่อยโครงสร้างนั้น มาพิจารณาร่วมกับวัตถุที่เป็นเซตของจุดภาพดำ



รูปที่ 2.8 ตัวอย่างส่วนย่อยโครงสร้างจุดภาพดำ S ขนาด 3×3 จุดภาพ

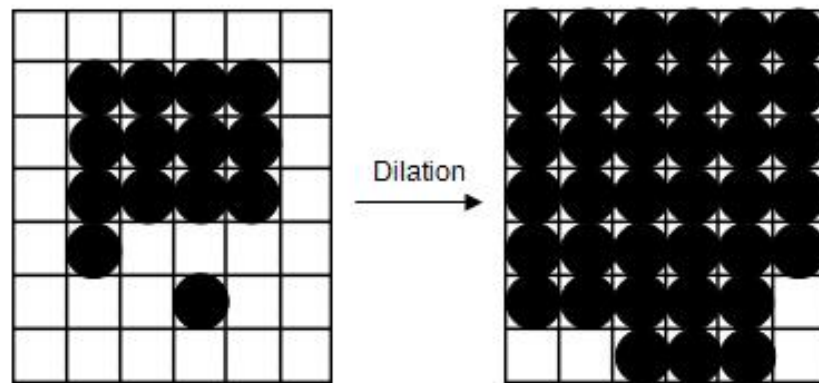
การทำงานพื้นฐานโดยทั่วไปของกระบวนการมอร์โฟโลยีมีดังนี้

1. การพองภาพ (Dilation)

เป็นการเพิ่มจำนวนจุดภาพดำเพื่อขยายขนาดกลุ่มของจุดภาพดำบนภาพ ทำให้เป็นการลดช่องว่างให้แคบลง หรือเชื่อมต่อกันระหว่างแต่ละกลุ่มจุดภาพดำที่อยู่ใกล้กัน โดยเมื่อนำส่วนย่อยโครงสร้างเลื่อนไปในบริเวณใดของภาพ แล้วพบว่าจุดกลางทับกับบริเวณที่เป็นจุดภาพดำบนภาพพอดี ก็จะทำให้การสำเนาส่วนย่อยโครงสร้างลงไปบริเวณนั้นของภาพผลลัพธ์

ยกตัวอย่างเช่น ให้เซต B แทนภาพที่เราต้องการนำมาทำการพองภาพโดยใช้ส่วนย่อยโครงสร้างจุดภาพดำ S ที่มีขนาด 3×3 จุดภาพข้างต้นมาพิจารณา สามารถเขียนแทนได้ด้วยสมการ $B \oplus S$ โดยมีนิยามดังนี้

$$B \oplus S = \{x \mid (S_x) \cap B \neq \emptyset\}$$



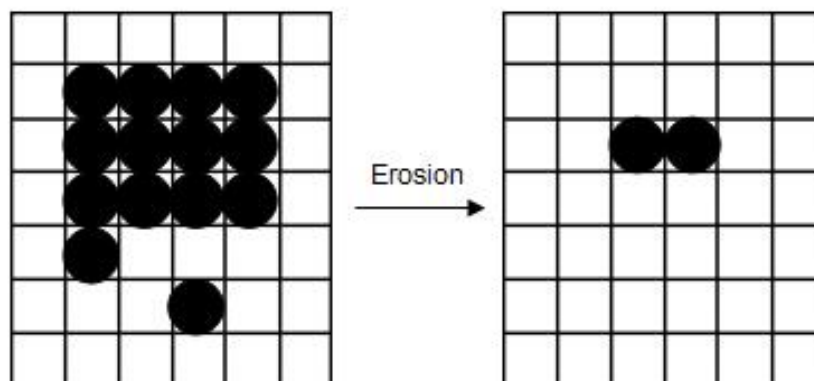
รูปที่ 2.9 การทำการพองภาพ $B \oplus S$

2. การกร่อนภาพ (Erosion)

เป็นการทำงานที่ตรงข้ามกับการพองภาพคือ ลดจำนวนของจุดภาพดำในภาพที่เราเป็นส่วนย่อยของโครงสร้างนั้นไปใช้พิจารณา หรือเป็นการกำจัดจุดภาพดำที่มีขนาดเล็กมากออกไปจากภาพ โดยเมื่อนำส่วนย่อยโครงสร้างเลื่อนไปในบริเวณใดของภาพ แล้วพบว่าส่วนย่อยโครงสร้างทับกับบริเวณของภาพพอดี ก็จะทำให้การใส่จุดภาพดำที่ภาพผลลัพธ์ ในตำแหน่งที่จุดภาพของภาพดั้งเดิมทับพอดีกับจุดกลางของส่วนย่อยโครงสร้าง

ยกตัวอย่างเช่น ให้เซต B แทนภาพที่เราต้องการนำมาทำการพองภาพโดยใช้ส่วนย่อยโครงสร้างจุดภาพดำ S ที่มีขนาด 3×3 จุดภาพข้างต้นมาพิจารณา สามารถเขียนแทนได้ด้วยสมการ $B \ominus S$ โดยมีนิยามดังนี้

$$B \ominus S = \{x \mid (S_x) \subseteq A\}$$



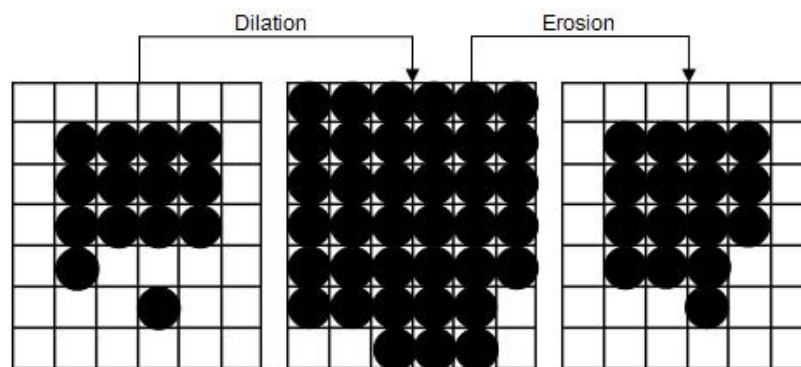
รูปที่ 2.10 การทำการกร่อนภาพ $B \ominus S$

3. การปิด (Closing)

เป็นการทำงานที่นำเอาการพองภาพ และการกร่อนภาพมาทำร่วมกัน โดยการปิดนั้นจะทำการพองภาพก่อนโดยเพิ่มจำนวนจุดภาพดำ จากนั้นค่อยทำการกร่อนภาพอีกครั้งหนึ่ง โดยลดจำนวนจุดภาพดำให้เท่ากับจำนวนรอบของการทำการพองภาพ

ยกตัวอย่างเช่น ให้เซต B แทนภาพที่เราต้องการนำมาทำการปิดโดยใช้ส่วนย่อยโครงสร้างจุดภาพดำ S ที่มีขนาด 3X3 จุดภาพข้างต้นมาพิจารณา สามารถเขียนแทนได้ด้วยสมการ $B \bullet S$ โดยมีนิยามดังนี้

$$B \bullet S = (B \oplus S) \ominus S$$



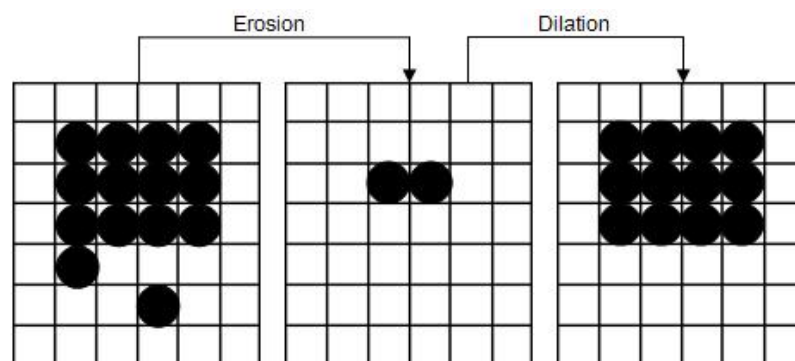
รูปที่ 2.11 การทำการปิด $B \bullet S$

4. การเปิด (Opening)

เป็นการทำงานที่นำเอาการพองภาพ และการกร่อนภาพมาทำร่วมกันเช่นเดียวกับการปิด แต่มีลำดับการทำงานที่ตรงข้ามกัน โดยการเปิดนั้นจะทำการกร่อนภาพก่อนโดยลดจำนวนจุดภาพดำ จากนั้นค่อยทำการพองภาพอีกครั้งหนึ่ง โดยเพิ่มจำนวนจุดภาพดำให้เท่ากับจำนวนรอบของการทำการกร่อนภาพ

ยกตัวอย่างเช่น ให้เซต B แทนภาพที่เราต้องการนำมาทำการเปิดโดยใช้ส่วนย่อยโครงสร้างจุดภาพดำ S ที่มีขนาด 3X3 จุดภาพข้างต้นมาพิจารณา สามารถเขียนแทนได้ด้วยสมการ $B \diamond S$ โดยมีนิยามดังนี้

$$B \diamond S = (B \ominus S) \oplus S$$



รูปที่ 2.12 การทำการเปิด $B \diamond S$

2.8 งานวิจัยที่เกี่ยวข้อง

โครงการนี้มีความเกี่ยวข้องกับงานวิจัยในหลายๆ ด้าน โดยเฉพาะเรื่องการปฏิสัมพันธ์ระหว่างมนุษย์กับเครื่องคอมพิวเตอร์ (Human computer interface: HCI) ซึ่งเป็นการศึกษาและวิเคราะห์เกี่ยวกับความสัมพันธ์ระหว่างมนุษย์กับเครื่องคอมพิวเตอร์ซึ่งมีจุดประสงค์หลักคือ เพื่อสร้างความสุข ความสะดวกสบายให้กับผู้ใช้ ณ เวลาที่ใช้งาน

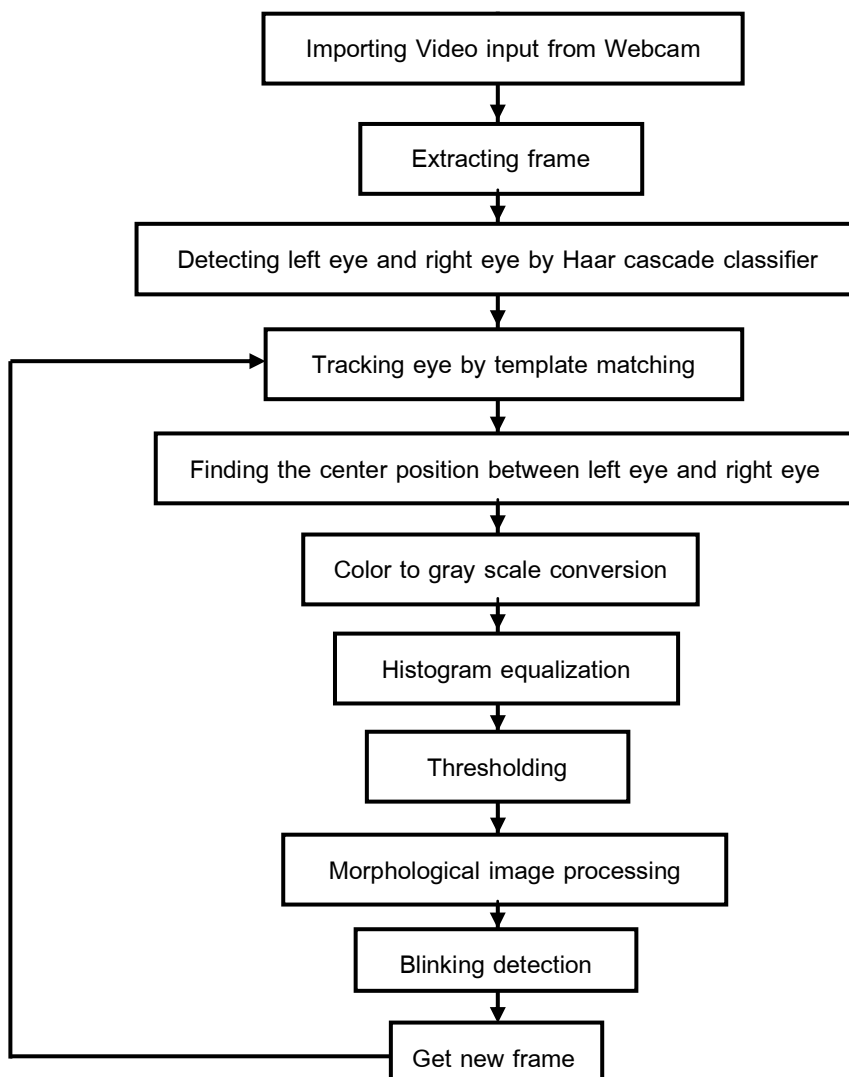
จากงานวิจัยในหัวข้อ Robust real time eye tracking for computer interface for disabled people โดย *Alberto De Santis, Daniela Iacoviello* จาก *Sapienza University of Rome* พบว่าการปฏิสัมพันธ์ระหว่างมนุษย์กับเครื่องคอมพิวเตอร์สำหรับผู้พิการนั้นเป็นระบบที่ซับซ้อน การใช้งานต้องประกอบไปด้วยอุปกรณ์มากมาย ได้แก่ อุปกรณ์ทางกายภาพ เช่น กล้องวิดีโอ อุปกรณ์ตรวจจับความเคลื่อนไหว เลนส์ชนิดพิเศษ และขั้นตอนวิธีที่ใช้วิเคราะห์สัญญาณภาพ โดยจุดประสงค์หลักของการทำงานนั้นคือ ระบุจุดที่จ้องมองของวัตถุที่ต้องการวิเคราะห์บนหน้าจอเครื่องคอมพิวเตอร์อย่างแม่นยำ เพื่อจัดหาอุปกรณ์ที่มีประสิทธิภาพมาใช้ในการพัฒนา ทั้งในเรื่องของการติดต่อสื่อสาร และสภาพแวดล้อม กล่าวคือหากอุปกรณ์ที่นำมาใช้นั้นมีประสิทธิภาพมาก ก็จะส่งผลให้ทำงานแบบทันกาลทำได้อย่างรวดเร็ว แต่ก็ทำให้ค่าใช้จ่ายในการพัฒนางานก็จะสูงตามไปด้วย ดังนั้นในงานวิจัยดังกล่าวจึงพัฒนาโปรแกรมโดยใช้ต้นทุนต่ำ แต่ใช้ขั้นตอนวิธีที่ซับซ้อนมาช่วยในการทำงานแบบทันกาลให้มีประสิทธิภาพแทน ซึ่งเป็นงานวิจัยที่ได้ส่งผลต่อแนวความคิด และตรงตามวัตถุประสงค์ของโครงการนี้เป็นอย่างมาก

บทที่ 3

การออกแบบโปรแกรมและขั้นตอนวิธี

3.1 ขั้นตอนวิธีของโปรแกรมโดยรวม

การทำงานของโปรแกรมเริ่มจากรับอินพุตที่เป็นภาพจากกล้องเว็บแคม จากนั้นใช้เทคนิคการประมวลผลภาพดิจิทัลมาระบุตำแหน่งของดวงตาทั้ง 2 ข้าง แล้วคำนวณหาตำแหน่งเฉลี่ยให้ได้จุดกึ่งกลางระหว่างดวงตาทั้ง 2 ข้าง เพื่อนำจุดนั้นมาใช้เป็นจุดอ้างอิงในการควบคุมทิศทางของเมาส์ และการคลิกเมาส์นั้นทำโดยการหลับตาข้างที่สัมพันธ์กับการคลิกเมาส์ที่ต้องการค้างไว้ภายในระยะเวลาที่กำหนด เช่น ใช้การหลับตาขวาในการคลิกเมาส์ขวา การคลิกเมาส์ซ้ายก็เช่นเดียวกัน ส่วนการดับเบิลคลิกจะใช้การหลับตาทั้งสองข้างพร้อมกัน



รูปที่ 3.1 ผังการทำงานของโปรแกรมโดยรวม

1. รับอินพุตจากกล้องเว็บแคม
2. ดึงเฟรมภาพจากวิดีโอมาประมวลผล
3. หาบริเวณของดวงตาทั้ง 2 ข้าง โดยใช้ตัวจำแนกของฮาร์
4. ติดตามดวงตาทั้ง 2 ข้างโดยใช้วิธีการเทียบแผ่นแบบ
5. หาจุดกึ่งกลางระหว่างดวงตาทั้ง 2 ข้าง
6. แปลงภาพส่วนบริเวณดวงตาทั้งสองข้างเป็นภาพระดับสีเทา
7. ทำฮิสโทแกรม อีควอไลเซชันกับภาพดวงตาทั้ง 2 ข้าง
8. แปลงภาพดวงตาทั้ง 2 ข้างให้เป็นภาพขาว-ดำ โดยใช้ค่าขีดแบ่ง (threshold)
9. ใช้เทคนิคการบวกรวมมอร์โฟโลยีเพื่อทำการจำแนกการเปิดปิดของตา
10. ประมวลผลคำสั่งการคลิกเมาส์ โดยที่การหลับตาซ้ายคือการคลิกเมาส์ซ้าย หลับตาขวาคือการคลิกเมาส์ขวา ส่วนการหลับตาทั้งสองข้างพร้อมกันคือดับเบิลคลิก
11. รับเฟรมภาพถัดไป

3.2 ไลบรารีที่ใช้ในการพัฒนาโปรแกรม

3.2.1 OpenCV

การใช้งาน OpenCV นั้นไม่สามารถเรียกใช้งานได้อย่างโปรแกรมทั่วไป ต้องใช้งานร่วมกับ Microsoft Visual Studio เนื่องจาก OpenCV เป็นเพียงชุดคำสั่งไม่ใช่เป็นตัวโปรแกรม จึงจำเป็นต้องมีการตั้งค่าให้กับโปรแกรม Microsoft Visual Studio ก่อนเพื่อระบุตำแหน่งไลบรารีของ OpenCV และตำแหน่งของไฟล์ที่ต้องใช้ในโปรแกรม เพื่อให้โปรแกรม Microsoft Visual Studio สามารถดึงไลบรารีของ OpenCV มาใช้งานได้

ประโยชน์ของ OpenCV คือ เป็นไลบรารีที่ช่วยให้นักพัฒนาโปรแกรมที่เกี่ยวข้องกับการประมวลผลภาพ และคอมพิวเตอร์วิทัศน์ทำงานได้อย่างสะดวก เนื่องจากมีฟังก์ชันสำเร็จรูปสำหรับการจัดการข้อมูลภาพ และการประมวลผลภาพพื้นฐาน ได้แก่

- การรับภาพจากกล้องเว็บแคมโดยใช้ชุดคำสั่งจาก HighGUI
- การทำฮิสโทแกรม อีควอไลเซชันเพื่อปรับปรุงค่าความเข้มสีของภาพให้มีความเหมาะสมโดยใช้ฟังก์ชัน `equalizeHist()`
- การแปลงภาพระดับสีเทาให้เป็นภาพขาว-ดำ ซึ่งประกอบด้วยสองสีเท่านั้นคือขาวกับดำ โดยใช้ฟังก์ชัน `threshold()`
- การทำเทคนิคของกระบวนการมอร์โฟโลยี โดยใช้การพองภาพ และการกร่อนภาพ เพื่อปรับปรุงคุณภาพของภาพ กำจัดสิ่งรบกวนที่ไม่พึงประสงค์ โดยใช้ฟังก์ชัน `dilate()` ในการพองภาพ และฟังก์ชัน `erode()` ในการกร่อนภาพ
- การทำการเทียบแผ่นแบบเพื่อติดตามดวงตาเมื่อมองไปยังตำแหน่งต่างๆ โดยใช้ฟังก์ชัน `matchTemplate()`
- การใช้ตัวจำแนกของฮาร์เพื่อตรวจจับดวงตาโดยใช้ฟังก์ชัน `HaarClassifierCascade` ดึงไฟล์ `.xml` จากไลบรารีใน OpenCV ที่มีภาพเทรนนิ่งดวงตาเอาไว้แล้ว เพื่อนำมาเทียบ และตรวจสอบว่าภาพที่เราป้อนเข้ามานั้นมีดวงตาอยู่บนภาพหรือไม่
- การหาบริเวณที่สนใจของภาพ เพื่อหาคำอธิบายของบริเวณที่เป็นดวงตาเอาไว้

3.3 การรับภาพเข้าโปรแกรม

3.3.1 คุณสมบัติกล้อง



รูปที่ 3.2 กล้องเว็บแคม Microsoft LifeCam VX-1000

กล้องเว็บแคม Microsoft LifeCam VX-1000 ใช้งานได้กับเครื่องคอมพิวเตอร์ Intel Pentium® III 550 MHz ขึ้นไป โดยเฉพาะ Intel Pentium 4 1.4 GHz สามารถถ่ายภาพนิ่งและวิดีโอขนาด 640X480 จุดภาพ (30 เฟรม/วินาที)

3.3.1 คุณสมบัติของภาพอินพุต

ภาพที่ใช้เป็นอินพุตนั้นต้องเป็นเฟรมภาพที่ดึงจากกล้องเว็บแคมจากสถานที่ที่มีแสงพอเหมาะ โดยที่มีขนาด 640X480 จุดภาพ ใบหน้าของผู้ใช้งานไม่ควรสวมแว่นตาทุกชนิด รวมถึงไม่มีสิ่งใดบดบังดวงตา นอกจากนั้นแล้วฉากหลังของภาพควรเป็นฉากเรียบ ไม่มีวัตถุใดๆ วางอยู่ โดยเริ่มจากใบหน้าตรง ลืมตาปกติ ทั้งสองข้างเพื่อทำการตรวจจับดวงตา หลังจากนั้นจึงเลื่อนตำแหน่งของใบหน้า เพื่อเป็นการควบคุมเมาส์ ส่วนการคลิกเมาส์ให้ทำโดยการหลับตา



รูปที่ 3.3 ภาพใบหน้าตรง



(ก)



(ข)



(ค)



(ง)

รูปที่ 3.4 ภาพการเลื่อนตำแหน่งของใบหน้า

(ก) การเลื่อนหน้าไปทางซ้าย

(ข) การเลื่อนหน้าไปทางขวา

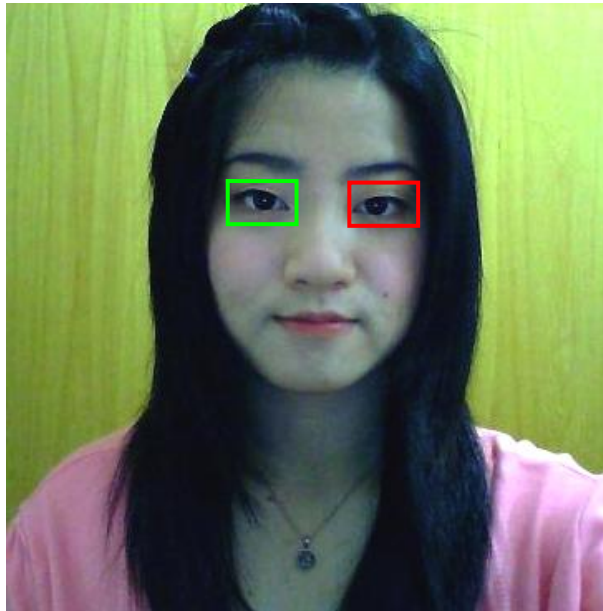
(ค) การเงยหน้าขึ้นด้านบน

(ง) การก้มหน้าลงด้านล่าง

3.4 การตรวจจับดวงตา

การตรวจจับดวงตานี้ทำได้โดยใช้ Haar cascade classifier ซึ่งเป็นฟังก์ชันใน OpenCV มาช่วยในการทำงาน เป็นการนำเอาไฟล์ประเภท XML ที่อยู่ในไลบรารีมาใช้ ซึ่งเป็นไฟล์ที่รวบรวมข้อมูลพิกัดตำแหน่งดวงตาบนใบหน้ามนุษย์ที่ได้รับเรียนรู้จากรูปภาพเป็นจำนวนมาก โดยแต่ละภาพก็จะบอกพิกัดของตำแหน่งที่มีดวงตาสถิตบนใบหน้าเอาไว้

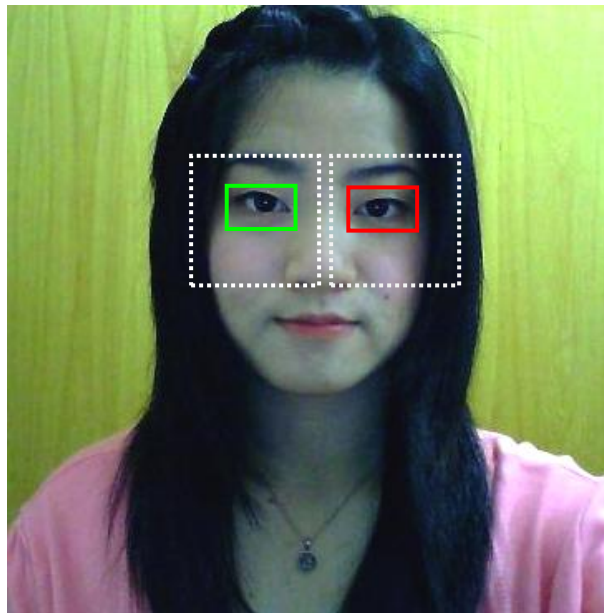
เมื่อได้รับเฟรมภาพจากกล้องเว็บแคมก็จะทำการประมวลผลเพื่อหาตำแหน่งของดวงตา แล้ววาดกรอบครอบบริเวณดวงตาโดยใช้หลักการหาบริเวณที่สนใจ



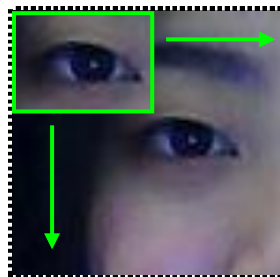
รูปที่ 3.5 ภาพการตรวจจับดวงตา

3.5 การติดตามดวงตา

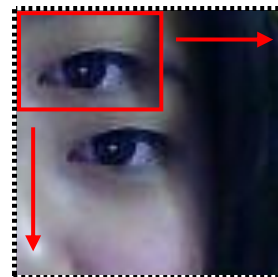
การติดตามดวงตานี้ทำได้โดยใช้เทคนิคการเทียบแผ่นแบบ ซึ่งเป็นวิธีที่ใช้ค้นหาบางส่วนของภาพที่มีความคล้ายคลึงกับภาพแผ่นแบบที่นำมาเปรียบเทียบโดยเลื่อนภาพแผ่นแบบบนภาพที่เราต้องการตรวจสอบจากบนซ้ายไปยังขวาล่าง ในกรณีนี้เราใช้การเทียบแผ่นแบบช่วยในการติดตามดวงตา โดยหลังจากการตรวจจับดวงตาได้จากภาพเฟรมแรกของกล้องเว็บแคมที่นำเข้ามา กำหนดให้ภาพดวงตานั้นเป็นภาพแผ่นแบบ จากนั้นตีกรอบรอบดวงตาอีกครั้งให้มีขนาดใหญ่ขึ้นกว่าเดิม เพื่อกำหนดให้เป็นบริเวณที่ใช้ทำการเทียบแผ่นแบบ แล้วนำภาพแผ่นแบบของดวงตาที่ได้มาเปรียบเทียบในบริเวณที่ใกล้เคียงตำแหน่งเดิมของดวงตา ที่อยู่ในกรอบนั้นเพื่อหาตำแหน่งถัดไปที่ดวงตามอง



(ก)



(ข)



(ค)

รูปที่ 3.6 ภาพการติดตามดวงตาในบริเวณที่ใกล้เคียงตำแหน่งเดิมของดวงตา
 (ก) การตีกรอบรอบดวงตาให้กว้างขึ้นเพื่อกำหนดบริเวณที่ใช้ทำการเทียบแผ่นแบบ
 (ข) การเทียบแผ่นแบบของตาซ้าย
 (ค) การเทียบแผ่นแบบของตาขวา

3.6 การควบคุมเมาส์

การควบคุมเมาส์นั้นในเบื้องต้นจะต้องหาจุดกึ่งกลางระหว่างดวงตาข้างซ้าย และข้างขวา เป็นจุดเริ่มต้นก่อนที่จะควบคุมการเลื่อนเมาส์ ซึ่งคำนวณได้จากสมการดังนี้

$$(x_c, y_c) = \left(\frac{x_l + x_r}{2}, \frac{y_l + y_r}{2} \right)$$

โดยกำหนดให้ (x_c, y_c) แทนพิกัดจุดกึ่งกลางระหว่างดวงตา
 (x_l, y_l) แทนพิกัดของตาซ้าย
 และ (x_r, y_r) แทนพิกัดของตาขวา



รูปที่ 3.7 ภาพแสดงการหาจุดกึ่งกลางระหว่างดวงตาข้างซ้าย และข้างขวา

หลังจากหาจุดกึ่งกลางระหว่างดวงตาทั้ง 2 ข้างซึ่งกำหนดให้เป็นจุดเริ่มต้นได้แล้วนั้น เมื่อผู้ใช้งานเปลี่ยนตำแหน่งของจุดนี้ไปยังที่ใดๆ ก็จะส่งผลให้เกิดระยะทางระหว่างจุด 2 จุด ซึ่งระยะห่างนี้มีผลต่อระยะการเคลื่อนที่ของเมาส์ ถ้าระยะห่างมากเมาส์ก็จะเคลื่อนที่ได้ไกล โดยมีวิธีคำนวณดังสมการนี้

$$distance = \sqrt{(x_c - x'_c)^2 + (y_c - y'_c)^2}$$

$$\theta = \arctan \left(\frac{y_c - y'_c}{x_c - x'_c} \right)$$

โดยกำหนดให้ $distance$ แทนระยะการเคลื่อนที่ของเมาส์
 และ (x'_c, y'_c) แทนพิกัดจุดกึ่งกลางระหว่างดวงตาตำแหน่งปัจจุบัน

ในกรณีที่ถ้าระยะห่างน้อยกว่าหรือเท่ากับ 8 จุดภาพ กำหนดให้เมาส์อยู่นิ่ง แต่ถ้าระยะห่างมากกว่า 8 จุดภาพ ขึ้นไปจะคำนวณหาพิกัดใหม่ที่เมาส์เลื่อนไป ตามสมการดังนี้

$$x_{c_update} = x_c + \frac{distance}{3} \cos\theta$$

$$y_{c_update} = y_c + \frac{distance}{3} \sin\theta$$

โดยกำหนดให้ $(x_{c_update}, y_{c_update})$ แทนพิกัดของตัวชี้ที่เลื่อนไป

หลังจากที่ได้ค่า (x_c, y_c) ในแต่ละครั้ง จะมีการตรวจสอบว่าพิกัดนั้นอยู่ภายในจอคอมพิวเตอร์หรือไม่ ถ้าหากเกินขอบจอก็จะกำหนดให้เมาส์หยุดอยู่แค่เพียงขอบจอเสมอ

3.7 การคลิกเมาส์

การคลิกเม้าสนั้นในเบื้องต้นจะควบคุมโดยการหลับตาข้างที่สัมพันธ์กับการคลิกเมาส์ที่ต้องการค้างไว้ภายในระยะเวลาที่กำหนด เช่น ใช้การหลับตาขวาในการคลิกเมาส์ขวา การคลิกเมาส์ซ้ายก็เช่นเดียวกัน ส่วนการดับเบิ้ลคลิก จะใช้การหลับตาทั้งสองข้างพร้อมกัน

โดยการลื้มตาหรือหลับตานั้นเราได้อาศัยเทคนิคพื้นฐานที่เกี่ยวข้องกับการประมวลผลภาพเชิงดิจิทัลมาใช้ในการปรับปรุงเฟรมภาพที่ได้รับมาจากกล้องเว็บแคม เพื่อให้นำมาพิจารณาจำแนกระหว่างการลื้มตาและหลับตาได้อย่างมีประสิทธิภาพยิ่งขึ้นโดยมีขั้นตอนดังนี้

1. เริ่มจากแปลงภาพส่วนของดวงตาที่ได้รับการตรวจจับจากภาพ RGB เป็นภาพระดับสีเทา
2. จากนั้นทำการปรับปรุงภาพระดับสีเทาที่ได้ให้มีสภาพแสงที่เหมาะสมโดยการทำฮิสโทแกรมอีควอไลเซชัน
3. จากนั้นแปลงภาพเป็นภาพขาว-ดำ
4. ทำ กระบวนการมอร์โฟโลยีเพื่อกำจัดสิ่งรบกวนที่ไม่พึงประสงค์บนภาพ

บทที่ 4

การพัฒนาและทดสอบระบบ

ในบทนี้จะกล่าวถึงการพัฒนาโปรแกรม และการทดสอบการทำงานของโปรแกรมการบอกตำแหน่งของตัวชี้ โดยใช้वेक्टरการเคลื่อนไหวของตา

4.1 การพัฒนาโปรแกรม

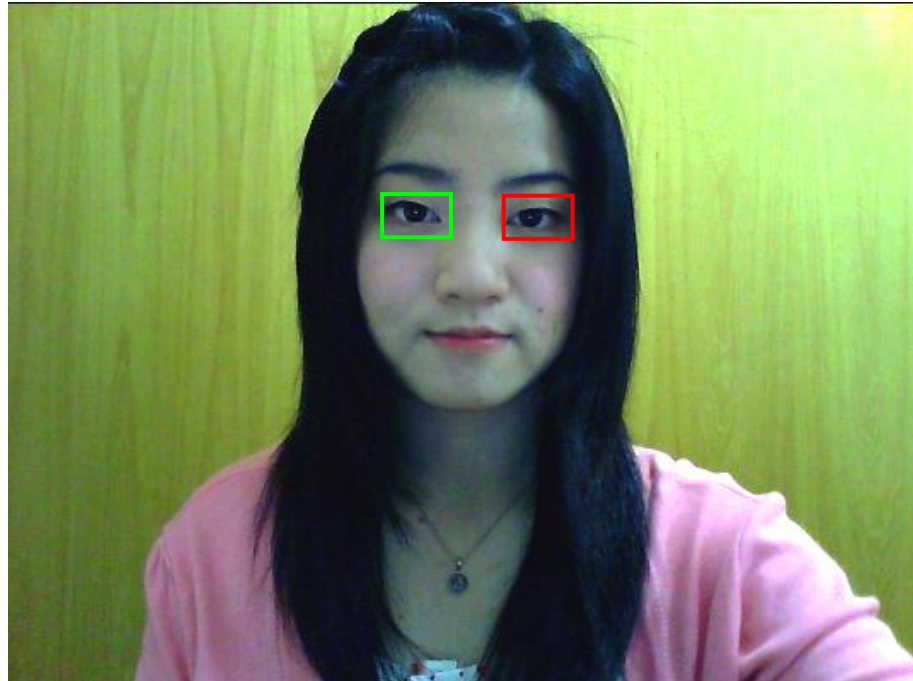
การพัฒนาโปรแกรมการบอกตำแหน่งของตัวชี้โดยใช้वेक्टरการเคลื่อนไหวของตา ถูกพัฒนาโปรแกรมบน Microsoft Visual Studio 2008 ด้วยภาษา C++ โดยมี OpenCV เป็นไลบรารีหลักที่ช่วยในการพัฒนาโปรแกรม ซึ่งมีการพัฒนาดังนี้

1. รับเฟรมภาพจากกล้องเว็บแคมโดยมีขนาด 640X480 จุดภาพ



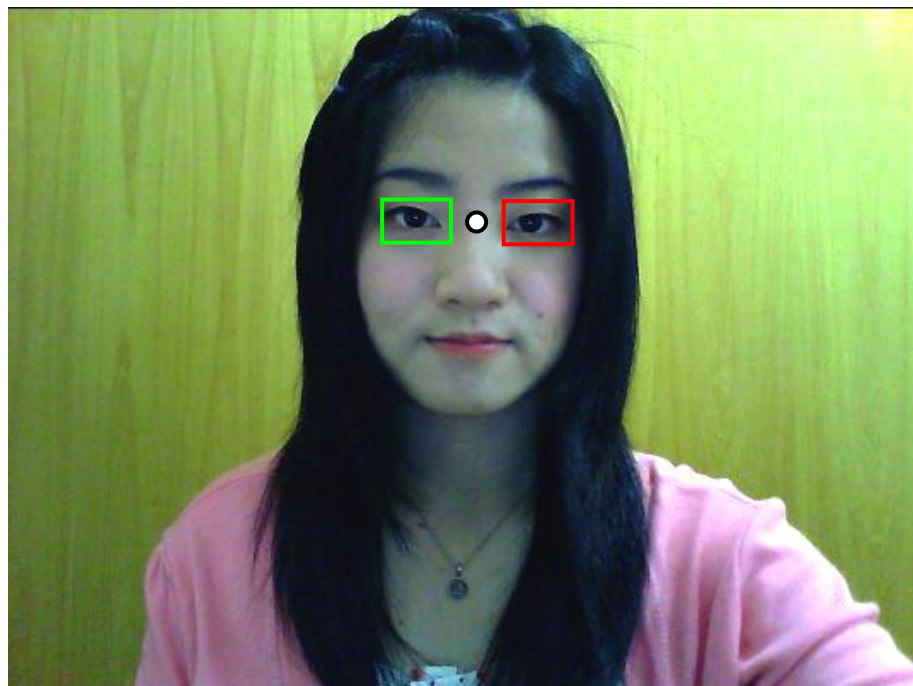
รูปที่ 4.1 ภาพหนึ่งเฟรม

2. ตรวจจับบริเวณดวงตาโดยใช้ Haar cascade classifier แล้วสร้างกรอบรอบ



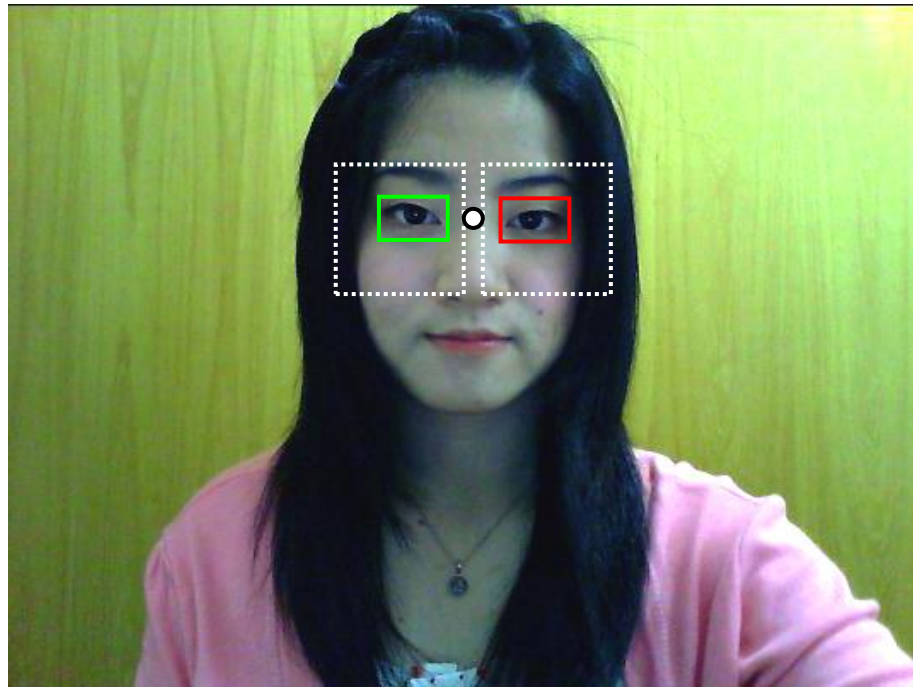
รูปที่ 4.2 ภาพแสดงการสร้างกรอบบริเวณดวงตา

3. หาดำแหน่งจุดกึ่งกลางระหว่างดวงตาเพื่อใช้เป็นจุดในการควบคุมเมาส์



รูปที่ 4.3 ภาพแสดงจุดกึ่งกลางระหว่างดวงตา

4. สร้างกรอบใหญ่ครอบกรอบที่ได้จากการตรวจจับบริเวณดวงตาทั้งสองข้าง เพื่อเป็นการกำหนดบริเวณในการค้นหาตำแหน่งถัดไปที่ดวงตามอง



รูปที่ 4.4 ภาพการสร้างกรอบใหญ่ครอบกรอบบริเวณดวงตาที่ได้ถูกตรวจจับในตอนแรก

5. นำภาพบริเวณดวงตาที่ตรวจจับได้จากเฟรมแรกนั้นพิจารณาแยกออกมาเป็นภาพแผ่นแบบของแต่ละด้าน



(ก)



(ข)

รูปที่ 4.5 ภาพแผ่นแบบของดวงตาที่ได้จากการตรวจจับดวงตาบนเฟรมภาพ
(ก) ภาพแผ่นแบบของดวงตาด้านซ้าย (ข) ภาพแผ่นแบบของดวงตาด้านขวา

6. นำภาพแผ่นแบบที่ได้มาทำการแปลงให้เป็นภาพระดับสีเทา



(ก)



(ข)

รูปที่ 4.6 ภาพแผ่นแบบของดวงตาหลังจากการแปลงเป็นภาพระดับสีเทา
(ก) ภาพแผ่นแบบของดวงตาด้านซ้าย (ข) ภาพแผ่นแบบของดวงตาด้านขวา

7. ทำฮิสโตแกรมของฮิสโทแกรมกับภาพแผ่นแบบระดับสีเทาที่ได้ เพื่อให้ได้รายละเอียดของดวงตาที่ชัดเจนขึ้น



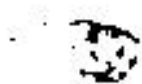
(ก)



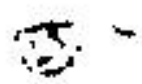
(ข)

รูปที่ 4.7 ภาพแผ่นแบบของดวงตาหลังจากที่ทำฮิสโตแกรมของฮิสโทแกรม
(ก) ภาพแผ่นแบบของดวงตาด้านซ้าย (ข) ภาพแผ่นแบบของดวงตาด้านขวา

8. แปลงภาพให้เป็นภาพขาว-ดำโดยใช้ค่าขีดแบ่ง



(ก)



(ข)

รูปที่ 4.8 ภาพแผ่นแบบของดวงตาหลังจากแปลงภาพด้วยค่าขีดแบ่ง
(ก) ภาพแผ่นแบบของดวงตาด้านซ้าย (ข) ภาพแผ่นแบบของดวงตาด้านขวา

9. ใช้เทคนิคกระบวนการมอร์โฟโลยีให้เห็นส่วนที่เป็นตาดำชัดขึ้น โดยทำการกร่อนภาพก่อนหนึ่งครั้ง ซึ่งในที่นี้ทำกับจุดภาพขาว



(ก)



(ข)

รูปที่ 4.9 ภาพแผ่นแบบของดวงตาหลังจากทำการกร่อนภาพ

(ก) ภาพแผ่นแบบของดวงตาด้านซ้าย (ข) ภาพแผ่นแบบของดวงตาด้านขวา

10. จากนั้นใช้เทคนิคกระบวนการมอร์โฟโลยีโดยทำการปิดครั้งที่หนึ่ง ในที่นี้ทำการพองภาพแล้วกร่อนภาพกับจุดภาพขาวเพื่อให้ได้บริเวณส่วนตาดำชัดยิ่งขึ้น



(ก)



(ข)

รูปที่ 4.10 ภาพแผ่นแบบของดวงตาหลังจากทำการปิดครั้งที่หนึ่ง

(ก) ภาพแผ่นแบบของดวงตาด้านซ้าย (ข) ภาพแผ่นแบบของดวงตาด้านขวา

11. และสุดท้ายใช้เทคนิคกระบวนการมอร์โฟโลยีโดยทำการปิดครั้งที่สอง จะได้บริเวณตาดำที่มีความชัดเจนมาก เพื่อให้ง่ายต่อการแยกแยะการเปิดและปิดของดวงตาสำหรับการคลิกเมาส์



(ก)



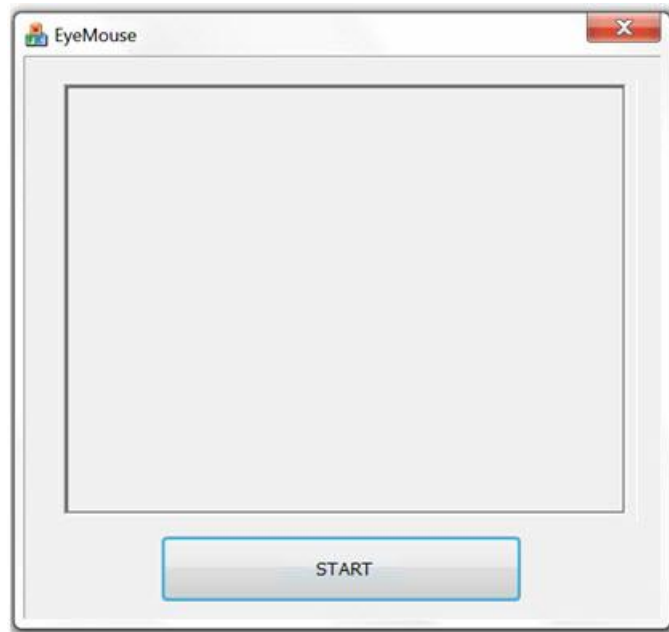
(ข)

รูปที่ 4.11 ภาพแผ่นแบบของดวงตาหลังจากทำการปิดครั้งที่สอง

(ก) ภาพแผ่นแบบของดวงตาด้านซ้าย (ข) ภาพแผ่นแบบของดวงตาด้านขวา

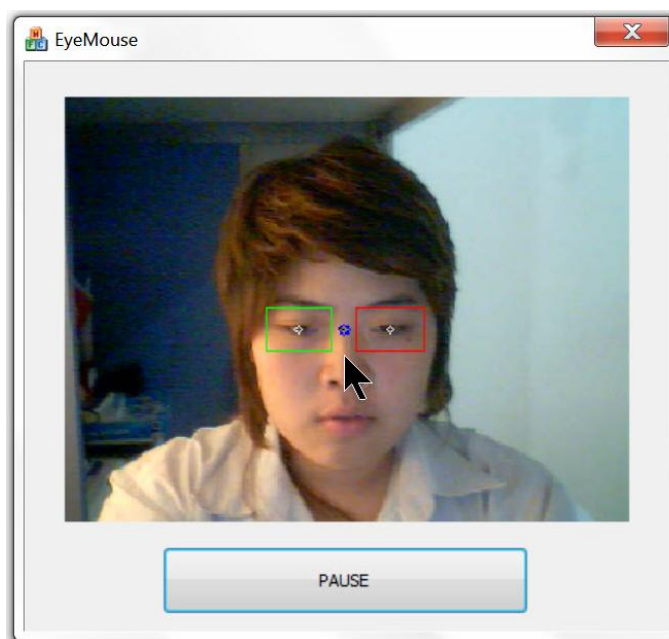
4.1 การทดสอบโปรแกรม

1. เปิดหน้าต่างของโปรแกรมขึ้นมา



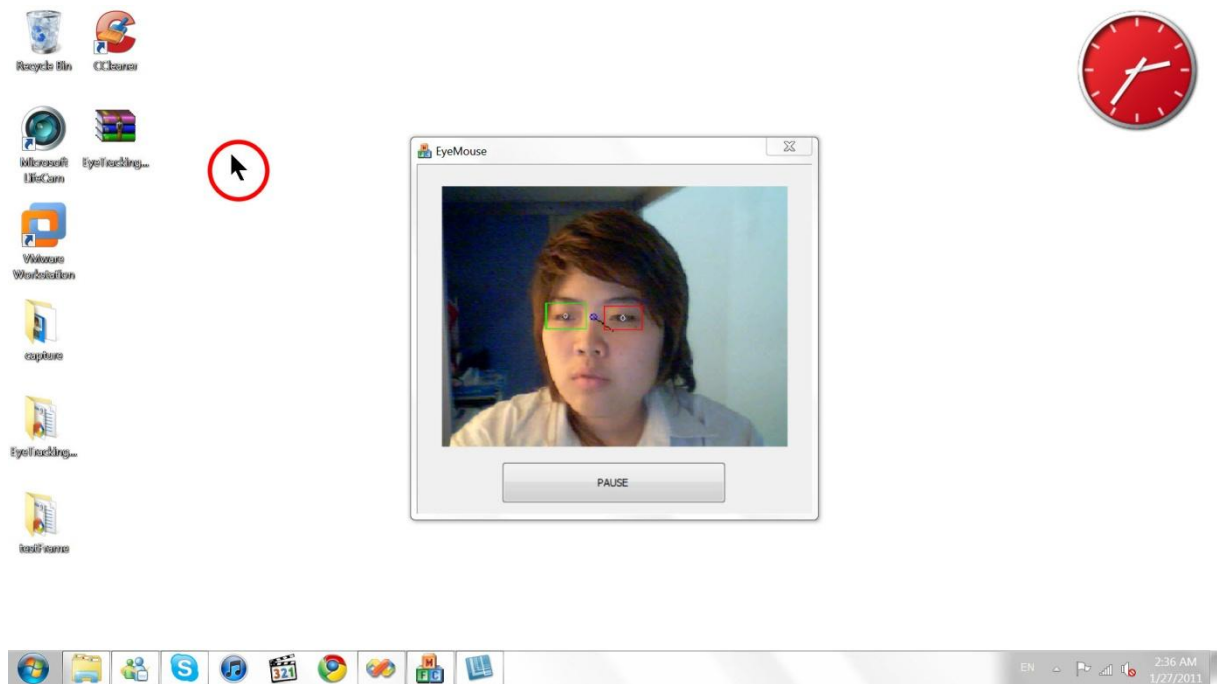
รูปที่ 4.12 หน้าต่างของโปรแกรม

2. หลังจากคลิกปุ่ม START จะเป็นการเริ่มตรวจจับบริเวณดวงตา และหาจุดกึ่งกลางระหว่างดวงตา



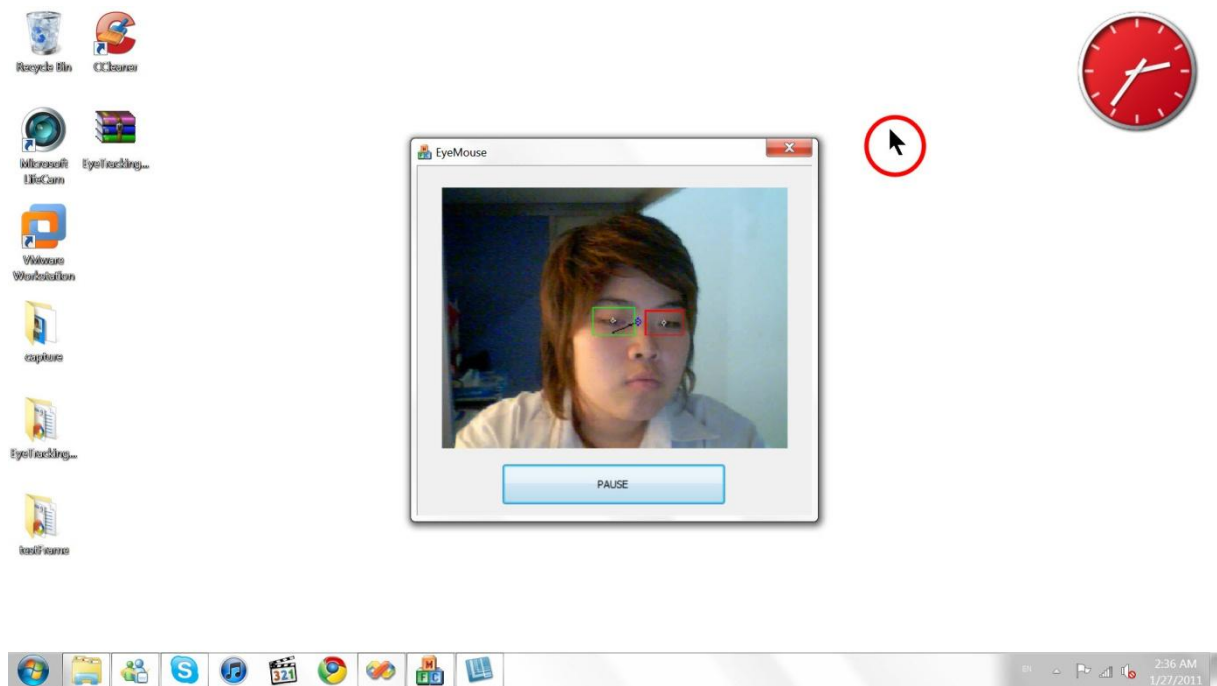
รูปที่ 4.13 การตรวจจับบริเวณดวงตาของโปรแกรม

3. การเลื่อนเมาส์ไปทางซ้ายบน



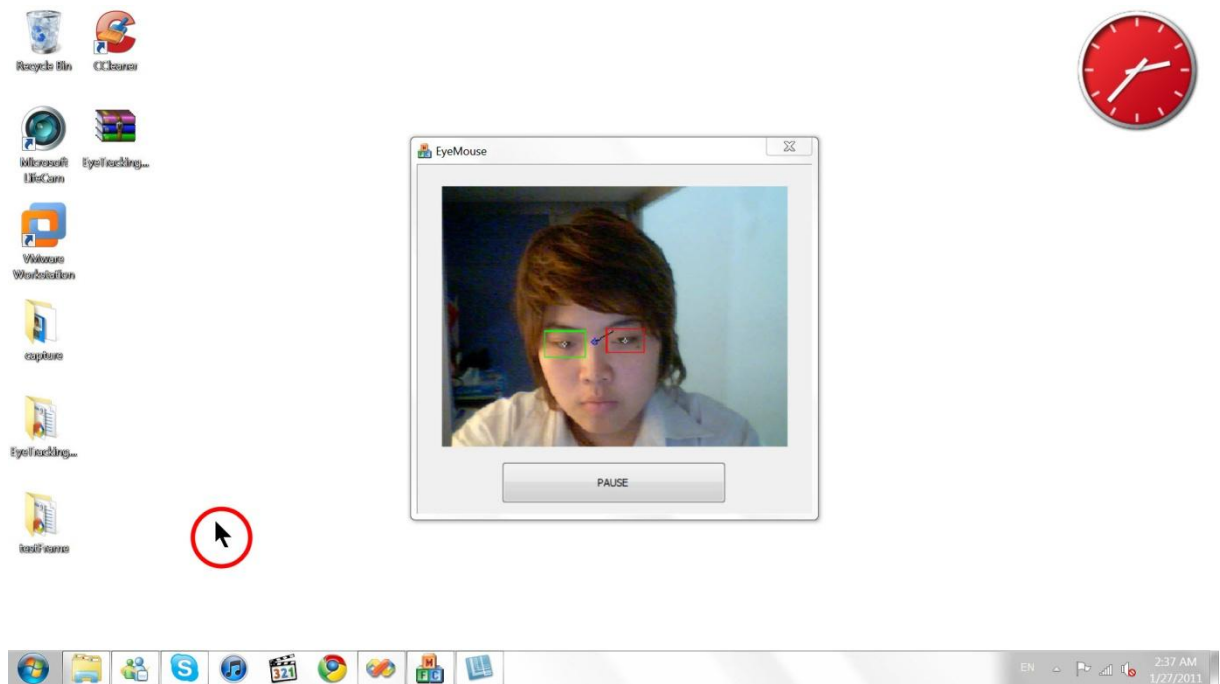
รูปที่ 4.14 การเลื่อนเมาส์ไปทางซ้ายบนของโปรแกรม

4. การเลื่อนเมาส์ไปทางขวาบน



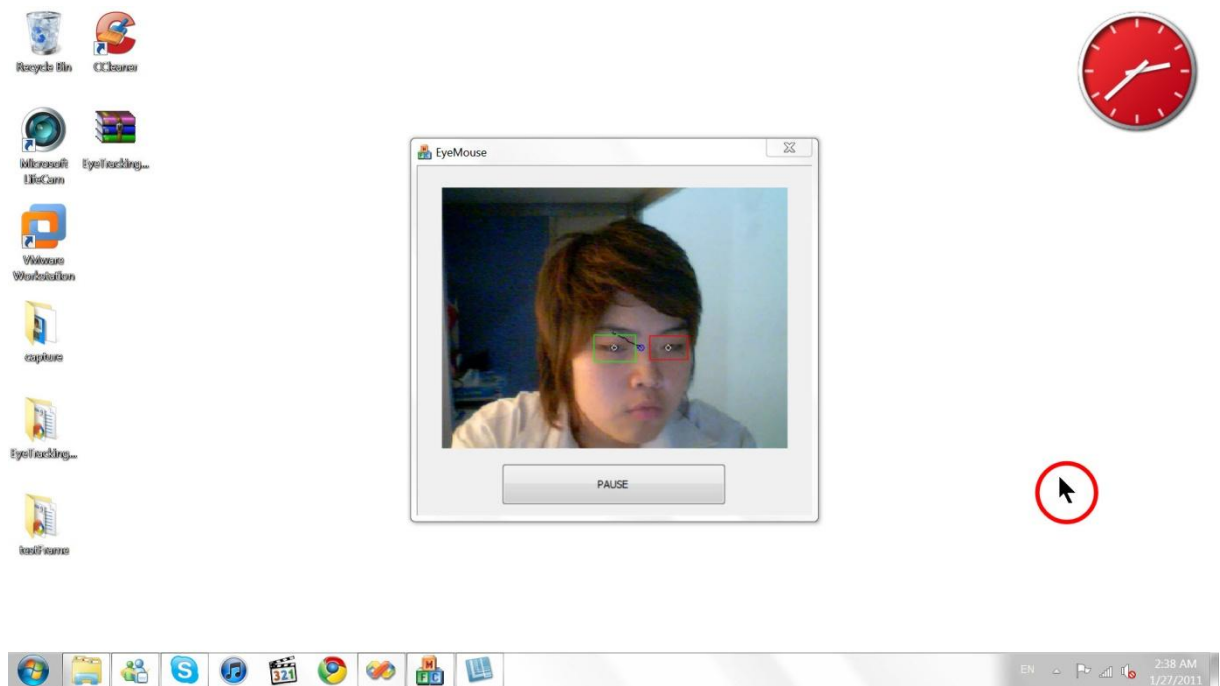
รูปที่ 4.15 การเลื่อนเมาส์ไปทางขวาของโปรแกรม

5. การเลื่อนเมาส์ไปทางซ้ายล่าง



รูปที่ 4.16 การเลื่อนเมาส์ไปทางซ้ายล่างของโปรแกรม

6. การเลื่อนเมาส์ไปทางขวาล่าง



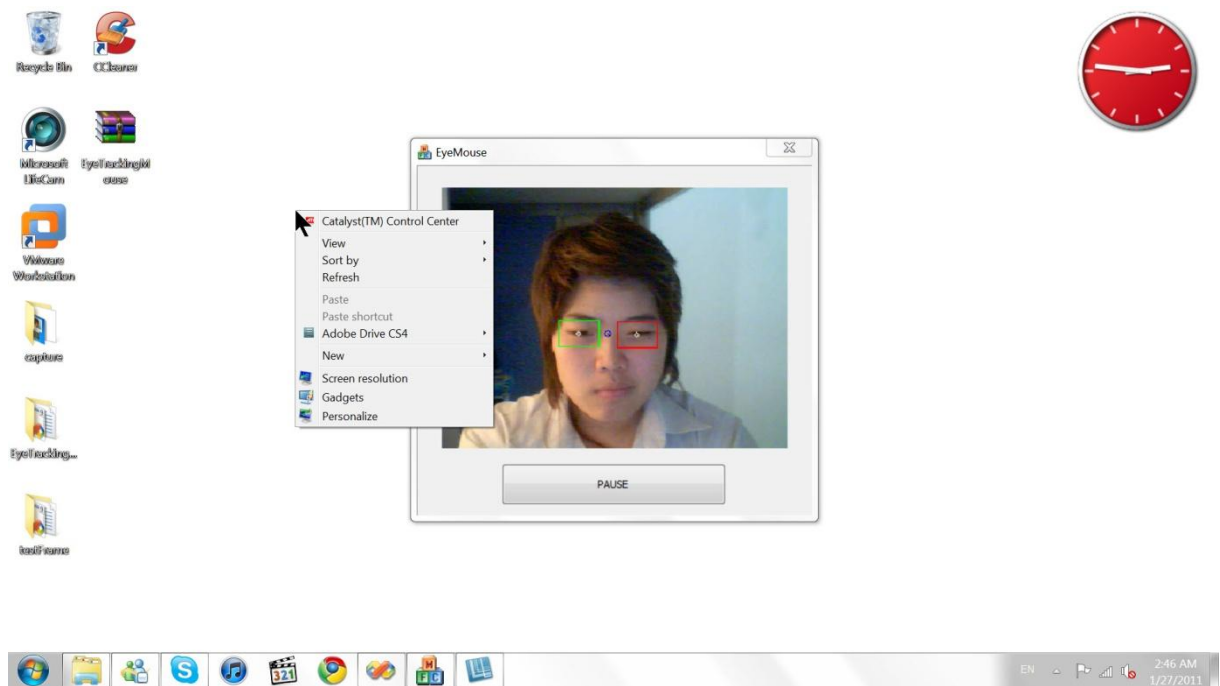
รูปที่ 4.17 การเลื่อนเมาส์ไปทางขวาล่างของโปรแกรม

7. การคลิกเมาส์ซ้ายด้วยการหลับตาซ้ายค้างไว้ภายในระยะเวลาหนึ่ง (12 เฟรมติดต่อกัน)



รูปที่ 4.18 การคลิกเมาส์ซ้ายของโปรแกรม

8. การคลิกเมาส์ขวาด้วยการหลับตาขวาค้างไว้ภายในระยะเวลาหนึ่ง (12 เฟรมติดต่อกัน)



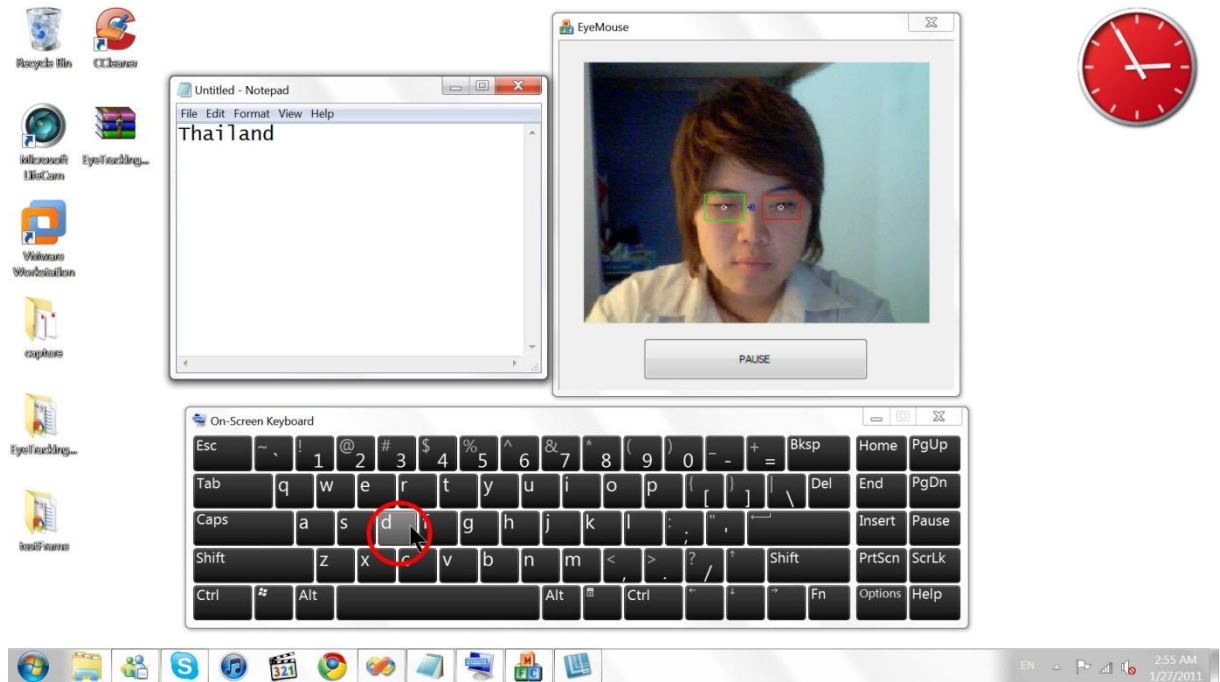
รูปที่ 4.19 การคลิกเมาส์ขวาของโปรแกรม

9. การดับเบิลคลิกด้วยการหันตาทั้งสองข้างค้างไว้ภายในระยะเวลาหนึ่ง (12 เฟรมติดต่อกัน)



รูปที่ 4.20 การดับเบิลคลิกของโปรแกรม

10. การใช้งานโปรแกรมในการพิมพ์อักษรโดยเรียกใช้แป้นพิมพ์บนหน้าจอคอมพิวเตอร์



รูปที่ 4.21 การใช้งานโปรแกรมในการพิมพ์อักษรของโปรแกรม

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 ผลที่ได้รับ

จากการทดลองโปรแกรมการบอกตำแหน่งของตัวชี้โดยใช้वेक्टरการเคลื่อนไหวของตา พบว่าโปรแกรมสามารถทำงานได้ตรงตามที่ตั้งเป้าหมายและอยู่ภายในขอบเขตของโครงการ โดยโปรแกรมสามารถควบคุมเมาส์ผ่านกล้องเว็บแคมได้ใกล้เคียงกับการใช้เมาส์จริง สามารถคลิกเมาส์ซ้ายได้ด้วยการหลับตาซ้าย คลิกเมาส์ขวาด้วยการหลับตาขวา และดับเบิลคลิกได้ด้วยการหลับตาพร้อมกันทั้งสองข้าง สามารถพิมพ์ตัวอักษรผ่านแป้นพิมพ์บนหน้าจอคอมพิวเตอร์ (On screen keyboard) ได้ และที่สำคัญงบประมาณในการพัฒนาโปรแกรมนี้นั้นมีราคาค่อนข้างต่ำ เพราะอุปกรณ์ที่ต้องใช้เพิ่มเติมมีเพียงกล้องเว็บแคมเท่านั้น ซึ่งหาได้ง่าย และราคาถูก

5.2 ปัญหาและอุปสรรค

1. ผู้พัฒนาโครงการมีความรู้พื้นฐานของภาษา C++ ไม่เพียงพอ ไม่เคยใช้งานไลบรารีที่มีความจำเป็นในการทำงานที่เกี่ยวข้องกับการประมวลผลภาพเชิงดิจิทัล หรือคอมพิวเตอร์วิทัศน์มาก่อน จึงทำให้ต้องใช้ระยะเวลาจำนวนมากในการศึกษา หาข้อมูล เพื่อนำมาใช้ในการพัฒนาโครงการ
2. ปัญหาเกี่ยวกับสภาพแวดล้อมของบริเวณที่ใช้งาน โดยมีบางอย่างที่อาจทำให้เกิดข้อผิดพลาดได้ เช่น สว่างมากเกินไป มีดเกินไป ทำให้ไม่สามารถแยกแยะการเปิดปิดของตาได้
3. ปัญหาด้านความละเอียดของภาพที่นำเข้า หรือความละเอียดของกล้องนั่นเอง หากมีความละเอียดที่ไม่เพียงพอจะก่อให้เกิดผลการทำงานที่มีข้อผิดพลาดและคลาดเคลื่อนได้
4. การเลื่อนตัวชี้ที่มีความคลาดเคลื่อนสูงมาก เนื่องจากตำแหน่งของตัวชี้จะแปรผันตามตำแหน่งของตา ณ เวลานั้นๆ โดยตรง

5.3 การแก้ปัญหา

จากปัญหาที่พบสามารถแก้ได้ ดังนี้

1. มีการศึกษาไปพร้อมกับการพัฒนาจริง ค้นหาข้อมูลเพิ่มเติมจากหนังสือและอินเทอร์เน็ต รวมไปถึงการศึกษาตัวอย่างการทำงานที่มีลักษณะคล้ายกัน
2. เพิ่มขอบเขตของการใช้งาน บริเวณที่ใช้งานจะต้องมีแสงสว่างที่อยู่ในระดับพอดี ไม่สว่างเกินไป หรือมืดเกินไป นอกจากนี้ยังมีการแก้ปัญหาโดยการปรับปรุงภาพให้สามารถทำงานได้อย่างมีประสิทธิภาพมากที่สุด
3. เพิ่มขอบเขตของอุปกรณ์ที่ใช้ โดยหากภาพที่นำเข้ามีความละเอียดมากเท่าไร การทำงานก็จะยิ่งมีประสิทธิภาพมากขึ้น แต่ในกรณีที่ภาพนำเข้ามีความละเอียดน้อย ก็จะต้องสามารถทำงานได้เช่นกัน โดยมีการใช้เทคนิคการประมวลผลภาพเชิงดิจิทัลให้มากขึ้น
4. เปลี่ยนวิธีการคำนวณหาตำแหน่งของตัวชี้ โดยใช้ตรีโกณมิติ

5.4 ข้อเสนอแนะ

1. ผู้ใช้งานจริงต้องมีการฝึกฝนการใช้งาน เพื่อให้โปรแกรมทำงานได้อย่างมีประสิทธิภาพและตรงตามความต้องการของผู้ใช้
2. ควรมีการพัฒนาระบบให้สามารถใช้งานได้กับสภาพแวดล้อมที่หลากหลายมากขึ้น
3. ควรมีการพัฒนาระบบให้สามารถทำงานอย่างมีประสิทธิภาพมากขึ้นเมื่อใช้กล้องเว็บแคมที่มีความละเอียดต่ำ

บรรณานุกรม

- [1] ศัพท์บัญญัติราชบัณฑิตยสถาน. [ออนไลน์]. เข้าถึงได้จาก <http://rirs3.royin.go.th/> สืบค้น 28 ธันวาคม 2553.
- [2] Bradski, Gary., & Kaehler, Adrian. (2008). **Learning OpenCV**. California : O'reilly.
- [3] Duchowski, Andrew T. (2007). **Eye Tracking Methodology**. (2nd ed.). London : Springer.
- [4] Gonzalez, Rafael C., & Woods, Richard E. (2002). **Digital Image Processing**. (2nd ed.). New Jersey : Prentice-Hall.
- [5] Santis, Alberto D., & Iacoviello, D. (2009). **Robust real time eye tracking for computer interface for disabled people**. ELSEVIER, Rome.
- [6] Shapiro, Linda G., & Stockman, George C. (2001). **Computer Vision**. New Jersey : Prentice-Hall.
- [7] Rafael C. Gonzalez and Richard E. Woods. **Digital Image Processing**. Addison-Wesley Publishing Company, 1992, chapter 4.
- [8] Robyn Owens. "Histogram equalization". [ออนไลน์] เข้าถึงได้จาก <http://homepages.inf.ed.ac.uk/rbf/CVonline> สืบค้น 3 มกราคม 2554.
- [9] Naotoshi Seo. "Tutorial: OpenCV haartraining". [ออนไลน์]. เข้าถึงได้จาก <http://note.sonots.com/SciSoftware/haartraining.html> สืบค้น 3 มกราคม 2554
- [10] "OpenCV Wiki". [ออนไลน์]. เข้าถึงได้จาก <http://opencv.willowgarage.com/wiki/Welcome> สืบค้น 3 มกราคม 2554.

ภาคผนวก ก

แบบเสนอหัวข้อโครงการ รายวิชา 2301399 Project Proposal

ชื่อโครงการ (ภาษาไทย)	การบอกตำแหน่งของตัวชี้โดยใช้เวกเตอร์การเคลื่อนไหวของตา		
ชื่อโครงการ (ภาษาอังกฤษ)	Locating pointer using eye motion vector		
อาจารย์ที่ปรึกษา	อาจารย์ ดร.ศุภกานต์ พิมลธเรศ		
ผู้ดำเนินการ	1. นางสาวจิรนนท์	จิระตระกูลวงศ์	รหัสประจำตัว 5033660023
	2. นางสาวลลิต	เดชธำรงวัฒน์	รหัสประจำตัว 5033701123

หลักการและเหตุผล

ในปัจจุบันนี้เครื่องคอมพิวเตอร์ได้กลายเป็นอุปกรณ์เทคโนโลยีที่มีการใช้งานกันอย่างแพร่หลาย และเข้ามามีบทบาทในชีวิตประจำวันเป็นอย่างมาก ไม่ว่าจะเป็นทั้งในองค์กรของรัฐบาล หรือเอกชน รวมไปถึงการใช้งานทั่วไปของส่วนบุคคล เนื่องจากเครื่องคอมพิวเตอร์เป็น เครื่องอิเล็กทรอนิกส์แบบอัตโนมัติ ทำหน้าที่เสมือนสมองกลใช้สำหรับแก้ปัญหาต่างๆ ทั้งที่ง่ายและซับซ้อนโดยวิธีทางคณิตศาสตร์ (จากพจนานุกรม ฉบับราชบัณฑิตยสถาน พ.ศ. 2542) [1] จึงเป็นเหตุให้เครื่องคอมพิวเตอร์สามารถให้ความช่วยเหลือ และอำนวยความสะดวกในการทำงานได้เป็นอย่างดี

เครื่องคอมพิวเตอร์ที่ใช้งานกันทั่วไปนั้น เป็นการออกแบบมาให้ผู้ใช้สั่งงานโดยการควบคุมทิศทางของเมาส์ด้วยมือเป็นหลัก หากผู้ใช้งานมีความผิดปกติ ไม่สามารถใช้งานมือและแขนได้ก็จะไม่สามารถใช้งานเครื่องคอมพิวเตอร์ได้ ทำให้กลุ่มผู้พิการมือและแขนนั้นสูญเสียโอกาสในการเรียนรู้และใช้งานเครื่องคอมพิวเตอร์ ไม่สามารถใช้บริการเทคโนโลยีสารสนเทศ เช่น การทำธุรกรรมบนอินเทอร์เน็ต การลงทะเบียนเรียนบน-อินเทอร์เน็ต และการทำงานอื่นๆ ที่ต้องใช้เครื่องคอมพิวเตอร์เป็นอุปกรณ์ช่วย

จากที่กล่าวมาข้างต้นส่งผลให้เกิดข้อจำกัดในการใช้งานเครื่องคอมพิวเตอร์สำหรับผู้พิการ ทำให้ผู้ดำเนินการมีแรงบันดาลใจในการพัฒนาซอฟต์แวร์ที่ทำให้สามารถควบคุมทิศทางของเมาส์ได้ ด้วยการใช้กล้องเว็บแคมจับการเคลื่อนไหวของเมาส์ โดยให้ทิศทางการเคลื่อนที่ของเมาส์มีความสัมพันธ์กับตำแหน่งในการมองผ่านกล้องเว็บแคม ซึ่งเป็นวิธีที่สะดวกสบายและง่ายต่อผู้ใช้งาน เนื่องจากอุปกรณ์ที่จำเป็นต้องใช้เพิ่มนอกเหนือจากเครื่องคอมพิวเตอร์และซอฟต์แวร์ มีเพียงกล้องเว็บแคมเท่านั้น อีกทั้งยังใช้ต้นทุนในการผลิตต่ำอีกด้วย

วัตถุประสงค์

1. เพื่อพัฒนาซอฟต์แวร์ที่ใช้ในการควบคุมเมาส์โดยใช้กล้องเว็บแคมจับการเคลื่อนไหวของม่านตา

ขอบเขตของโครงการ

1. ตรวจสอบตำแหน่งของม่านตาเป็นหลัก
2. ติดตามทิศทางการเคลื่อนไหวของม่านตาในขณะที่ใบหน้ามีการเคลื่อนไหวน้อยที่สุด
3. ประสานการทำงานระหว่างทิศทางการเคลื่อนไหวของม่านตา กับทิศทางของเมาส์ในจอมอนิเตอร์
4. ตรวจจับได้ครั้งละ 1 ใบหน้า

ขั้นตอนการดำเนินการ

1. ระบุหัวข้อโครงการและศึกษาทฤษฎีที่เกี่ยวข้อง
2. ออกแบบโครงสร้างโปรแกรม และเลือกอุปกรณ์และเทคโนโลยีที่ใช้
3. ศึกษาลักษณะการเคลื่อนที่ของลูกตาดำของมนุษย์
4. พัฒนาโปรแกรมส่วนการประมวลผลภาพจากกล้องเว็บแคม
5. พัฒนาโปรแกรมตรวจจับลูกตาดำของมนุษย์
6. พัฒนาโปรแกรมติดตามการเคลื่อนไหวลูกตาดำของมนุษย์
7. เชื่อมต่อการทำงานระหว่างทิศทางการเคลื่อนไหวของลูกตาดำให้สอดคล้องกับทิศทางของเมาส์
8. ทดสอบและแก้ไขข้อผิดพลาด
9. สรุปผลการทดลอง
10. จัดทำเอกสารประกอบโครงการ

ตารางเวลาการดำเนินการ

ขั้นตอนการดำเนินการ	เดือน								
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.
1. ระบุหัวข้อโครงการและศึกษา ทฤษฎีที่เกี่ยวข้อง									
2. ออกแบบโครงสร้างโปรแกรม และเลือกอุปกรณ์และเทคโนโลยีที่ใช้									
3. ศึกษาลักษณะการเคลื่อนที่ของ ลูกตาของผู้ใช้									
4. พัฒนาโปรแกรมส่วนการ ประมวลผลภาพจากกล้องเว็บแคม									
5. พัฒนาโปรแกรมตรวจจับลูกตา ของผู้ใช้									
6. พัฒนาโปรแกรมติดตามการ เคลื่อนไหวลูกตาของผู้ใช้									
7. เชื่อมต่อการทำงานระหว่างทิศ ทางการเคลื่อนไหวของลูกตาให้ สอดคล้องกับทิศทางของเมาส์									
8. ทดสอบและแก้ไขข้อผิดพลาด									
9. สรุปผลการทดลอง									
10. จัดทำเอกสารประกอบโครงการ									

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้เรียนรู้เทคนิคและขั้นตอนวิธีการตรวจจับลูกตา
2. ได้เรียนรู้เทคนิคและขั้นตอนวิธีการติดตามลูกตา
3. ได้เรียนรู้เทคนิคการประยุกต์ใช้แหล่งรวมชุดคำสั่ง (Library) มาตรฐานต่างๆที่มีอยู่
4. ได้สร้างโอกาส (ทางเลือก) ให้แก่ผู้พิการทางแขน ที่มีความต้องการจะใช้งานเครื่องคอมพิวเตอร์

อุปกรณ์และเครื่องมือที่ใช้

1. Hardware

1.1 Computer

- Processor : Intel[®] Core™2 Duo CPU T6600 2.2.GHz
- Memory : 4.00 GB 800 MHz DDR2 SDRAM
- Hard Drive : 320 GB (SATA) with shock absorbers
- Graphic Card : NVIDIA GEFORCE[®] with CUDA™ GT210
- Screen : 13.3" WXGA Clear Super View TFT display

2. Software

2.1 Microsoft Windows 7 Home Premium

2.2 Microsoft Visual Studio 2008 Professional Edition

2.3 Intel Open Source Computer Vision Library (OpenCV

บรรณานุกรม

[1] พจนานุกรม ฉบับราชบัณฑิตยสถาน พ.ศ. 2542

<http://rirs3.royin.go.th/>

[2] Santis, Alberto D., & Iacoviello, D. (2009). **Robust real time eye tracking for computer interface for disabled people.** ELSEVIER, Rome.

ภาคผนวก ข

คู่มือการติดตั้งโปรแกรมที่ใช้ในการพัฒนา

ในภาคผนวก ข นี้จะแสดงขั้นตอนวิธีการลงโปรแกรมที่ใช้ในการพัฒนาโปรแกรมการบอกตำแหน่งของตัวชี้โดยใช้เวกเตอร์การเคลื่อนไหวของตา ซึ่งโปรแกรมหลักที่ใช้คือ Microsoft Visual Studio 2008, OpenCV และ Cmake 2.8 ที่ช่วยในการสร้าง Build script ของ OpenCV เพื่อให้ใช้งานร่วมกับ Microsoft Visual Studio 2008 ได้ง่ายและมีประสิทธิภาพ

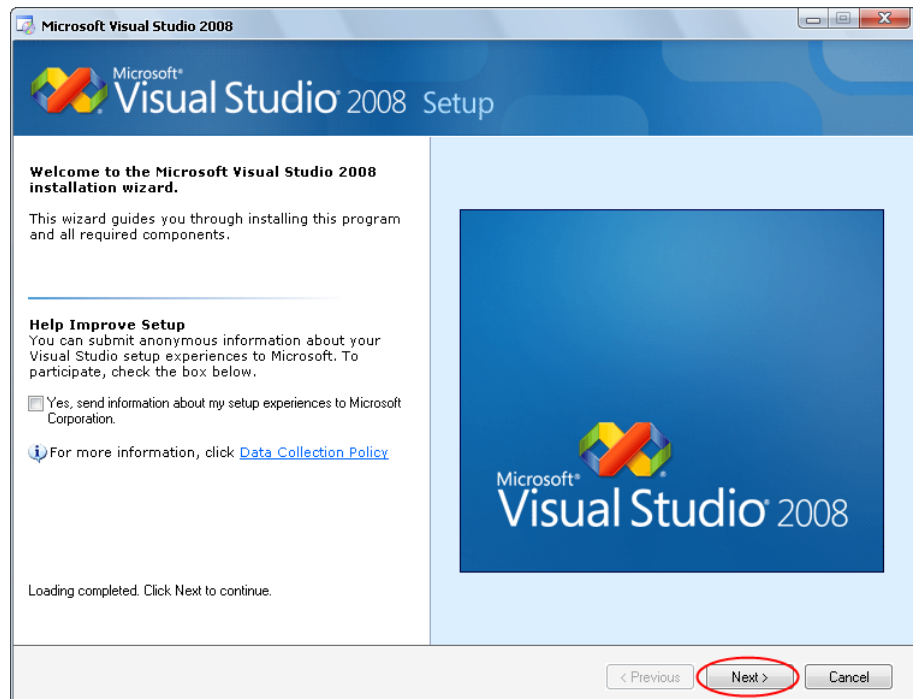
ขั้นตอนวิธีการติดตั้งโปรแกรม Microsoft Visual Studio 2008

1. เมื่อใส่แผ่นโปรแกรมเข้าไปในเครื่องคอมพิวเตอร์แผ่นโปรแกรมจะทำงานอัตโนมัติซึ่งจะปรากฏหน้าต่างให้คลิกที่ Install Visual Studio 2008 ดังภาพ



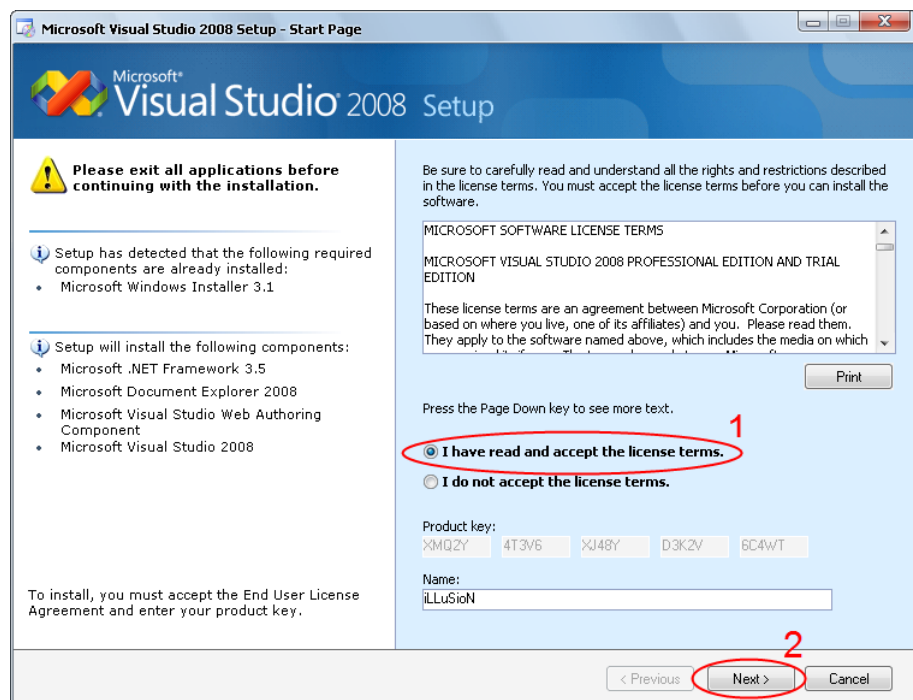
รูปภาคผนวก ข-1 หน้าต่างเริ่มต้นการติดตั้งโปรแกรม Microsoft Visual Studio 2008

2. หลังจากนั้นเลือก Next เพื่อดำเนินการต่อไป



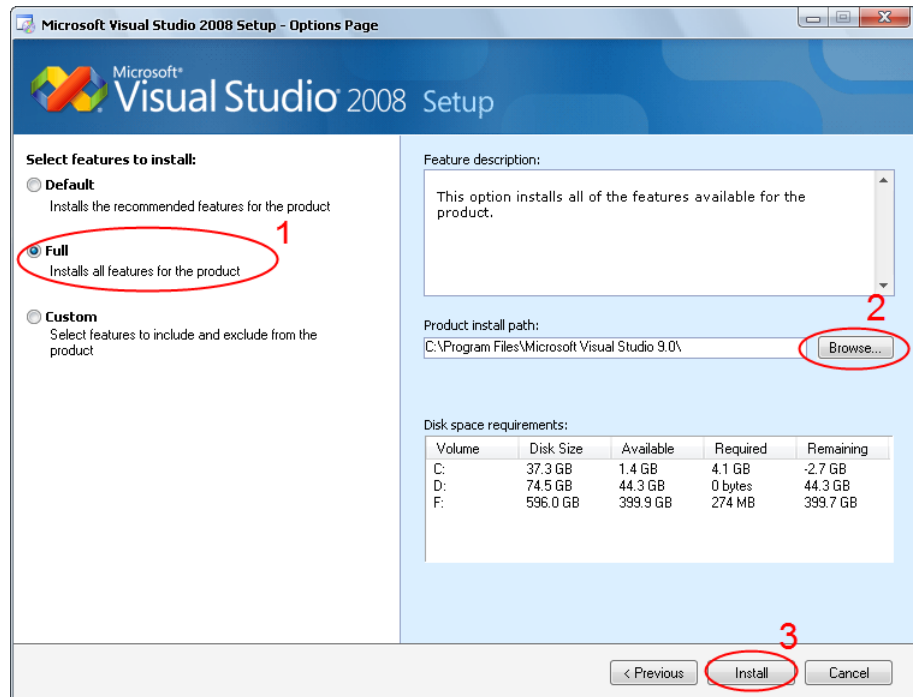
รูปภาพผนวก ข-2 หน้าต่างต้อนรับการติดตั้งโปรแกรม Microsoft Visual Studio 2008

3. เมื่อปรากฏหน้าต่างถัดมา จะแสดงถึงข้อตกลงต่างๆของการติดตั้งโปรแกรม ให้เลือก I have read and accept the license terms. จากนั้นเลือก Next



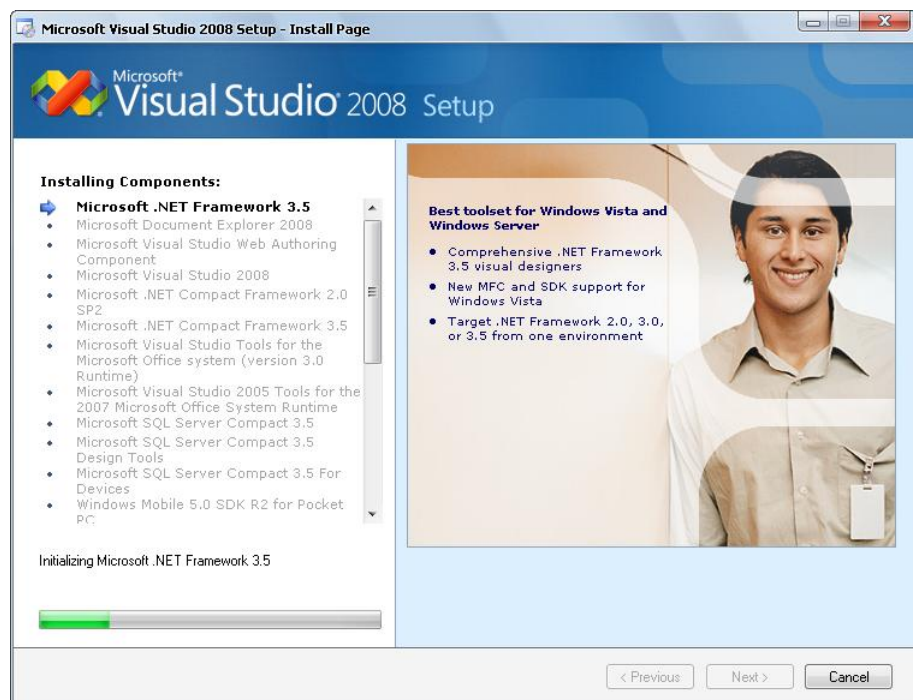
รูปภาพผนวก ข-3 หน้าต่างแสดงข้อตกลงต่างๆของการติดตั้งโปรแกรม Microsoft Visual Studio 2008

4. เมื่อปรากฏหน้าต่างถัดมา ให้เลือกติดตั้งโปรแกรมแบบ Full เพื่อติดตั้งโปรแกรมแบบสมบูรณ์ และเลือกจุดหมาย (Path) ที่จะติดตั้งโปรแกรม จากนั้นคลิก Install



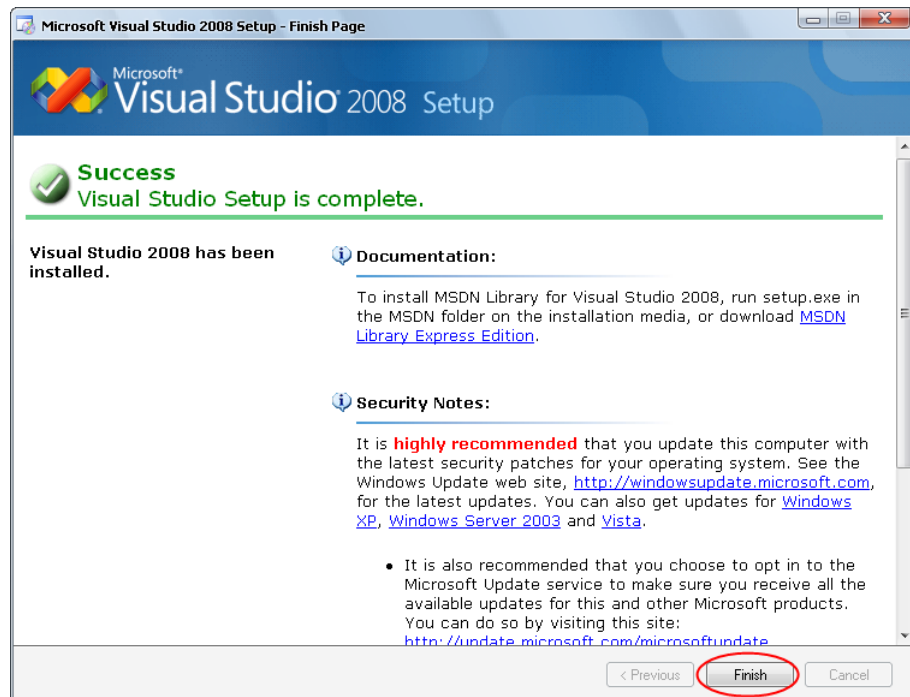
รูปภาคผนวก ข-4 หน้าต่างแสดงรูปแบบการติดตั้งโปรแกรม Microsoft Visual Studio 2008

5. จากนั้น รอจนกว่าโปรแกรมจะติดตั้งสมบูรณ์



รูปภาคผนวก ข-5 หน้าต่างแสดงว่ากำลังติดตั้งโปรแกรม Microsoft Visual Studio 2008

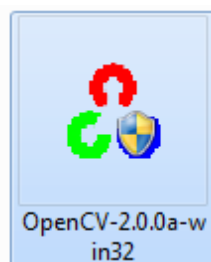
6. เมื่อโปรแกรมติดตั้งเสร็จสมบูรณ์ จะปรากฏหน้าต่างว่าติดตั้งเสร็จเรียบร้อยแล้ว ให้คลิก Finish



รูปภาพผนวก ข-6 หน้าต่างแสดงว่าโปรแกรม Microsoft Visual Studio 2008 ติดตั้งสมบูรณ์

ขั้นตอนวิธีการติดตั้ง Open CV

1. ดับเบิลคลิกที่ไอคอนตัวติดตั้งของ OpenCV ดังภาพ



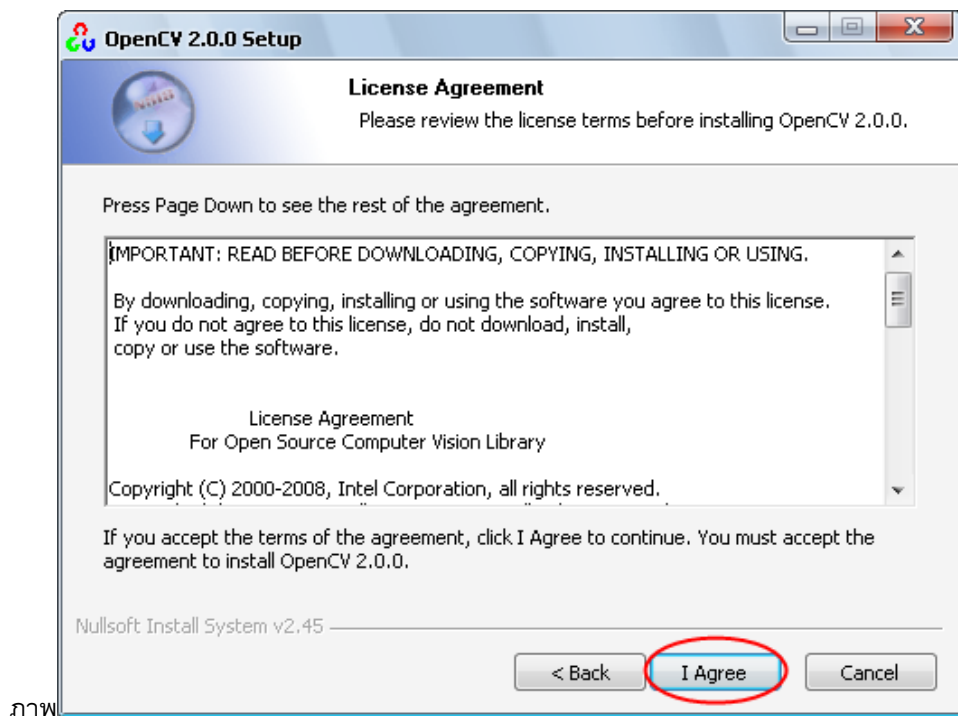
รูปภาพผนวก ข-7 ไอคอนตัวติดตั้งของ OpenCV

2. หลังจากที่ได้ดับเบิลคลิกไอคอนตัวติดตั้งของ OpenCV จะมีหน้าต่างปรากฏขึ้นมาดังภาพ



รูปภาพผนวก ข-8 หน้าต่างเริ่มต้นของการติดตั้ง OpenCV

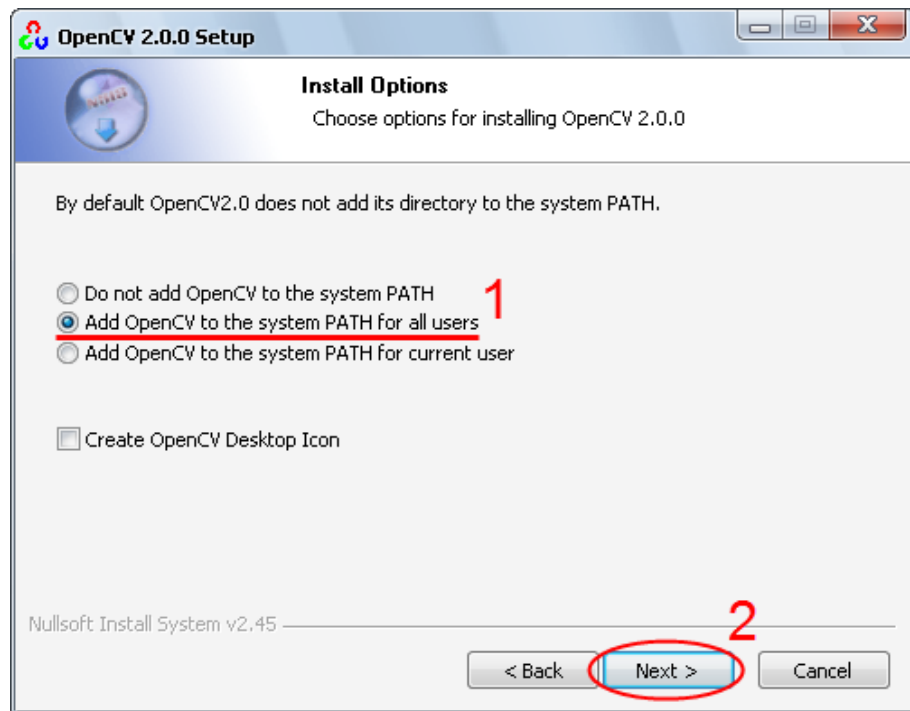
3. หลังจากคลิก Next แล้วจะปรากฏหน้าต่างแสดงข้อตกลงต่างๆ ของการติดตั้งโปรแกรมให้คลิก Agree ดัง



ภาพ

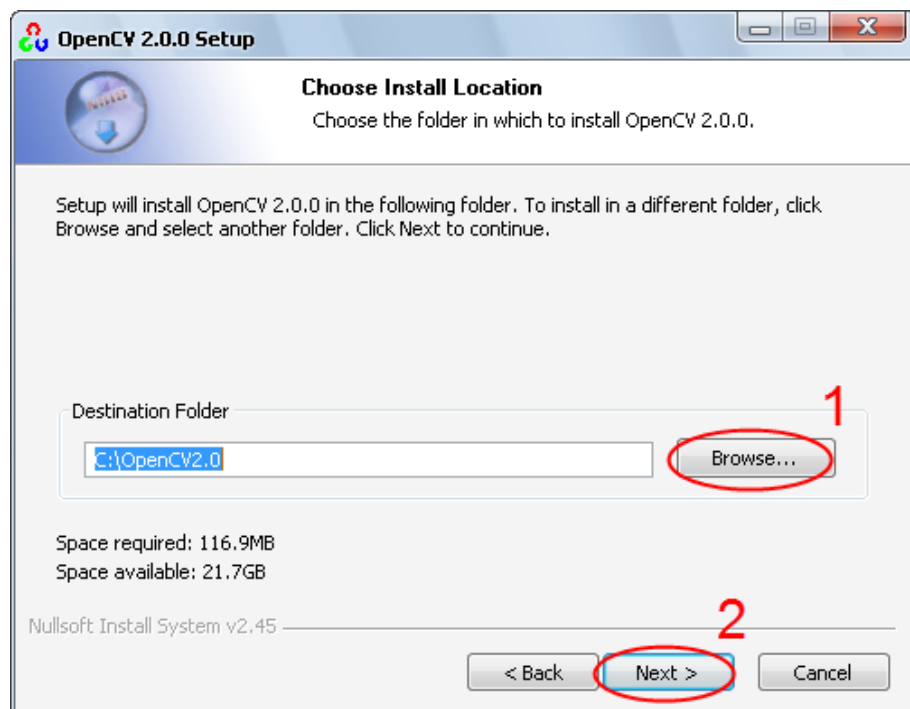
รูปภาพผนวก ข-9 หน้าต่างแสดงข้อตกลงต่างๆของการติดตั้งโปรแกรม

4. จากนั้นจะปรากฏหน้าต่างตัวเลือกการติดตั้ง ให้เลือก **Add OpenCV to the system PATH for all user** และคลิก Next เพื่อดำเนินการต่อดังภาพ



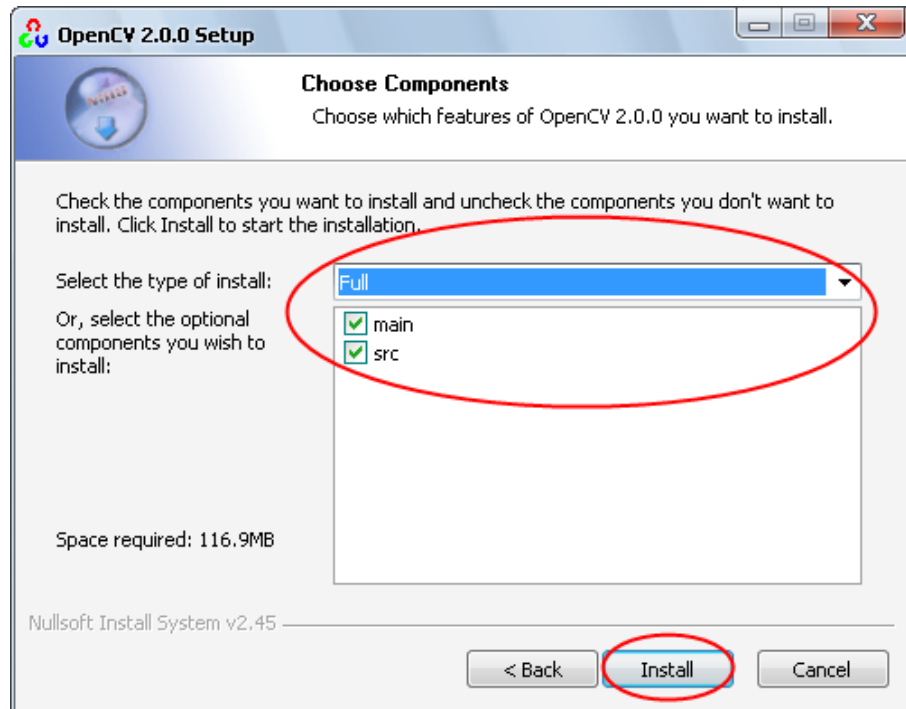
รูปภาพผนวก ข-10 หน้าต่างตัวเลือกการติดตั้ง OpenCV

5. หลังจากนั้นจะแสดงหน้าต่างให้เลือกตำแหน่งที่จะติดตั้ง OpenCV แล้วคลิก Next เพื่อดำเนินการต่อดังภาพ



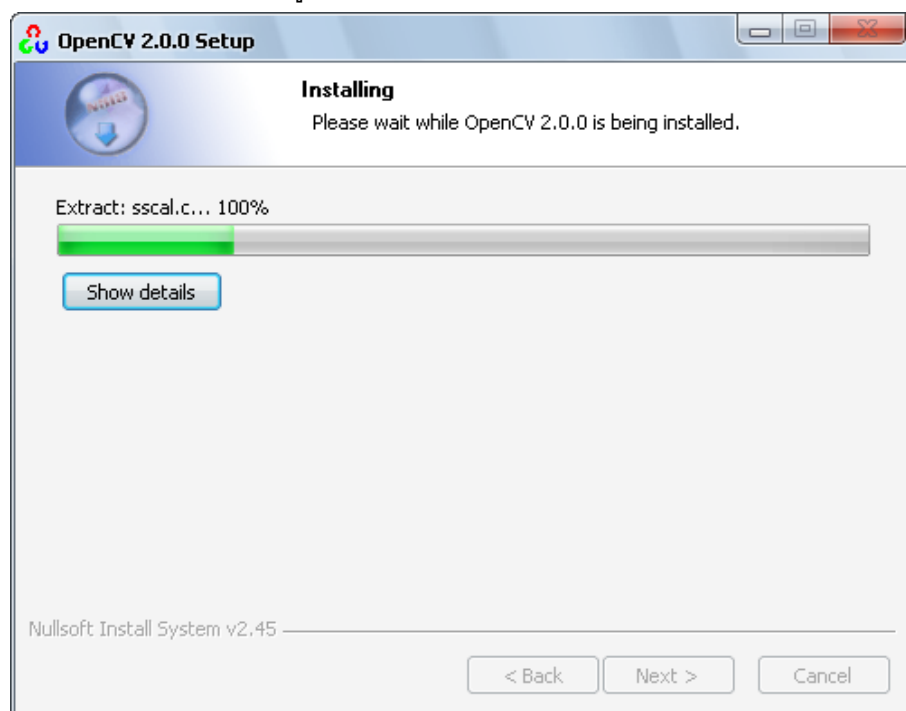
รูปภาคผนวก ข-11 หน้าต่างเลือกตำแหน่งการติดตั้ง OpenCV

6. หลังจากคลิก Next จนเรียบร้อยแล้วจะปรากฏหน้าต่างให้เลือกรูปแบบการติดตั้ง ให้เลือกรูปแบบ Full และเลือกทั้ง main และ src จากนั้นเริ่มดำเนินการติดตั้งโดยคลิก install ดังภาพ



รูปภาคผนวก ข-12 หน้าต่างเลือกรูปแบบการติดตั้ง OpenCV

7. จากนั้นรอกจนกว่าการติดตั้งจะเสร็จสมบูรณ์



รูปภาพผนวก ข-13 หน้าต่างแสดงว่ากำลังการติดตั้ง OpenCV

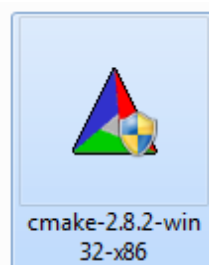
- เมื่อ OpenCV ติดตั้งเสร็จสมบูรณ์แล้วให้คลิก finish



รูปภาพผนวก ข-14 หน้าต่างแสดงว่าการติดตั้ง OpenCV เสร็จสมบูรณ์

ขั้นตอนวิธีการติดตั้งโปรแกรม Cmake 2.8

- ดับเบิลคลิกที่ไอคอนตัวติดตั้งของ Cmake 2.8 ดังภาพ



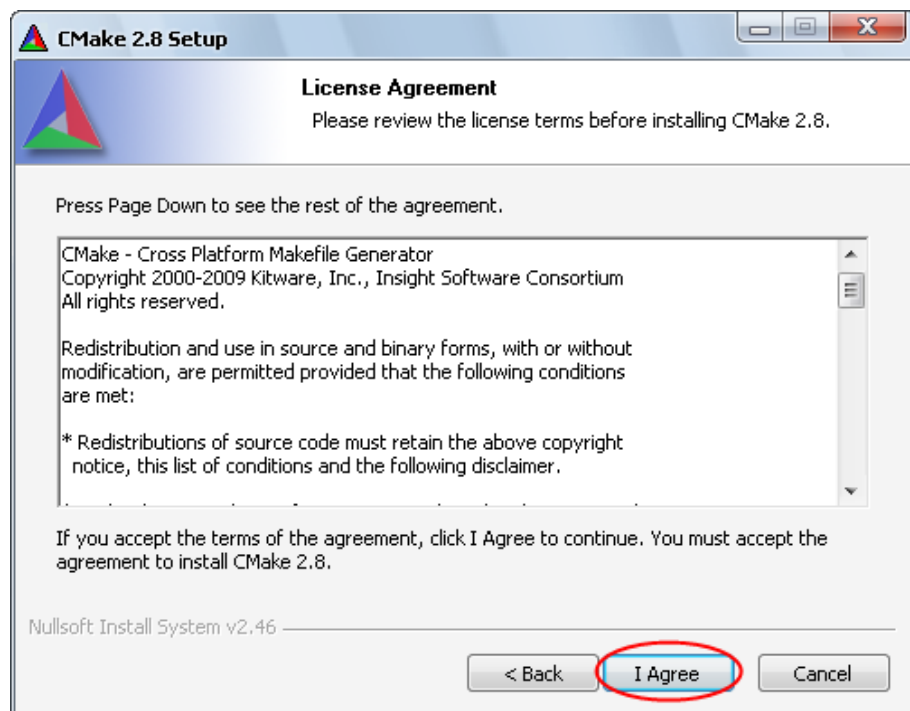
รูปภาพผนวก ข-15 ไอคอนตัวติดตั้งของ Cmake 2.8

2. หลังจากที่ได้ดับเบิลคลิกไอคอนตัวติดตั้งของ Cmake 2.8 จะมีหน้าต่างปรากฏขึ้นมาดังภาพ



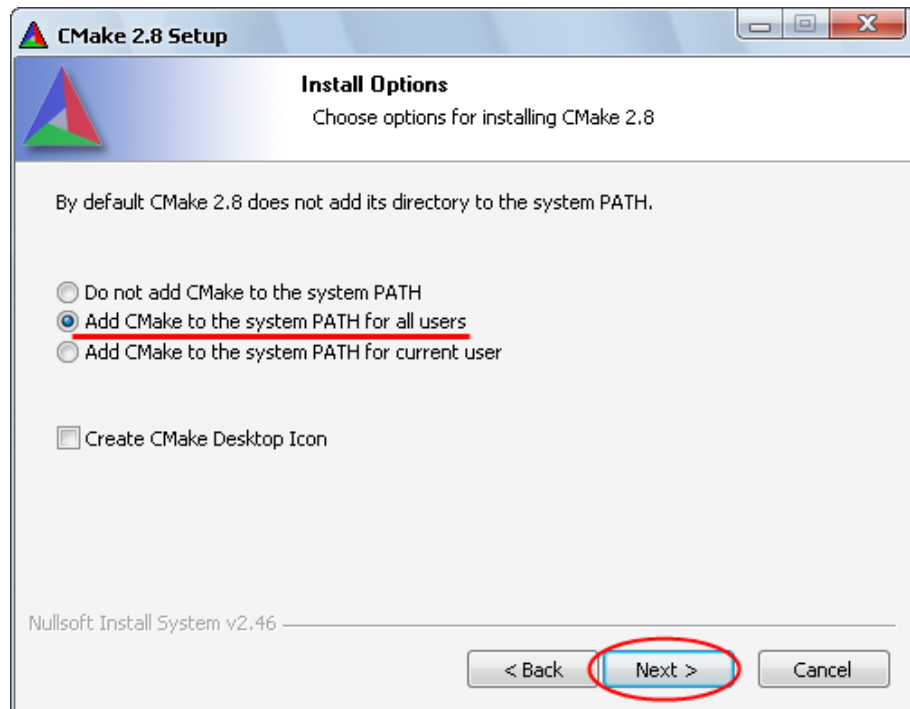
รูปภาคผนวก ข-16 หน้าต่างเริ่มต้นของการติดตั้ง Cmake 2.8

3. หลังจากคลิก Next แล้วจะปรากฏหน้าต่างแสดงข้อตกลงต่างๆ ของการติดตั้งโปรแกรมให้คลิก Agree ดังภาพ



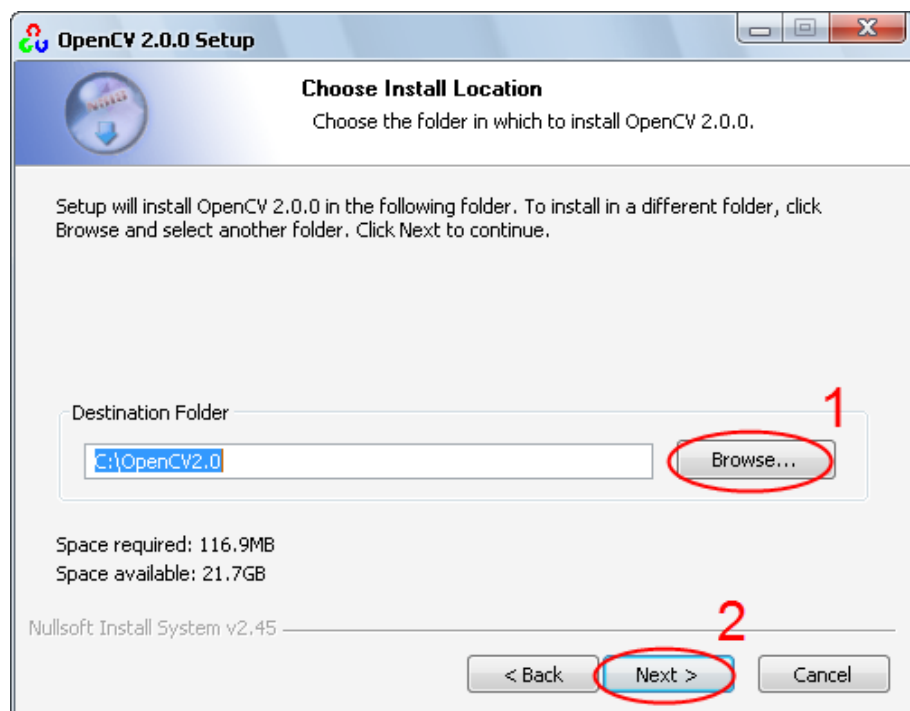
รูปภาคผนวก ข-17 หน้าต่างแสดงข้อตกลงต่างๆของการติดตั้งโปรแกรม

4. จากนั้นจะปรากฏหน้าต่างตัวเลือกการติดตั้ง ให้เลือก **Add CMake to the system PATH for all user** และคลิก Next เพื่อดำเนินการต่อดังภาพ



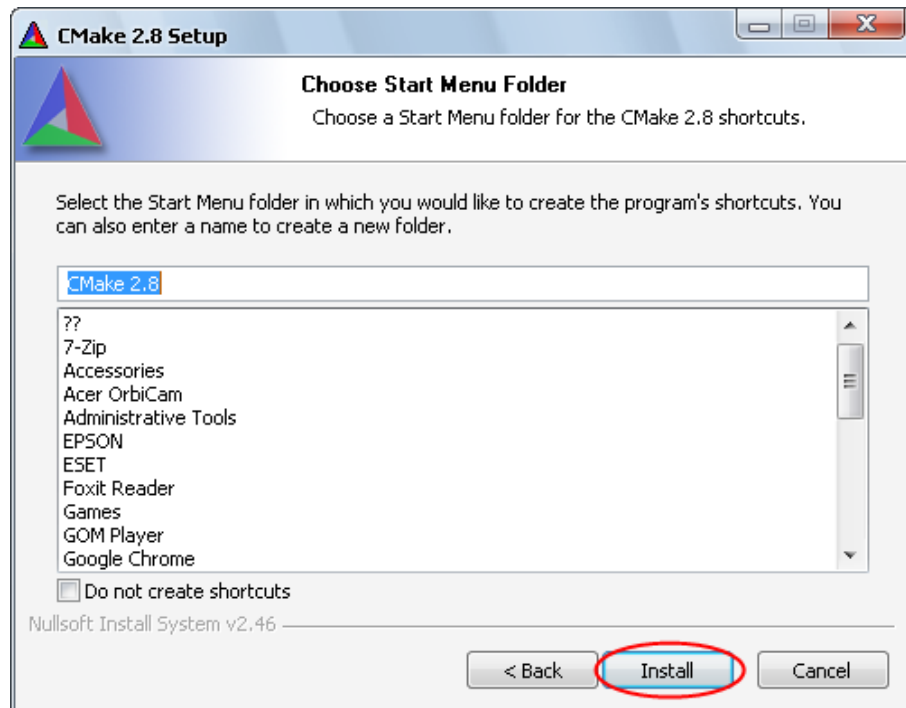
รูปภาพผนวก ข-18 หน้าต่างตัวเลือกการติดตั้ง Cmake 2.8

5. หลังจากนั้นจะแสดงหน้าต่างให้เลือกตำแหน่งที่จะติดตั้ง Cmake 2.8 แล้วคลิก Next เพื่อดำเนินการต่อดังภาพ



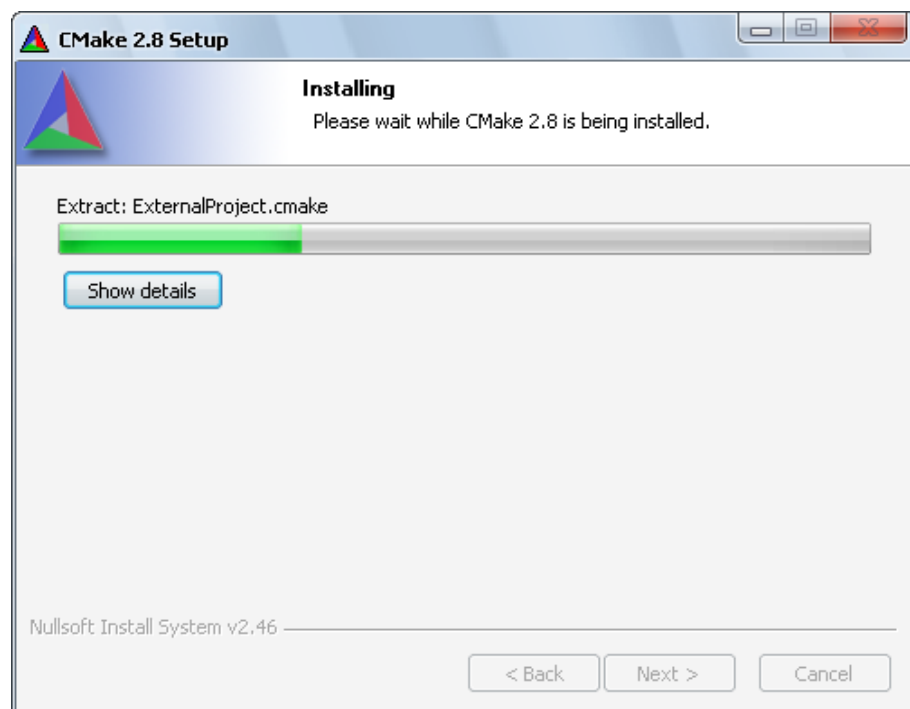
รูปภาคผนวก ข-19 หน้าต่างเลือกตำแหน่งการติดตั้ง Cmake 2.8

6. หลังจากนั้นทำการติดตั้งโปรแกรมโดยคลิก install ดังภาพ



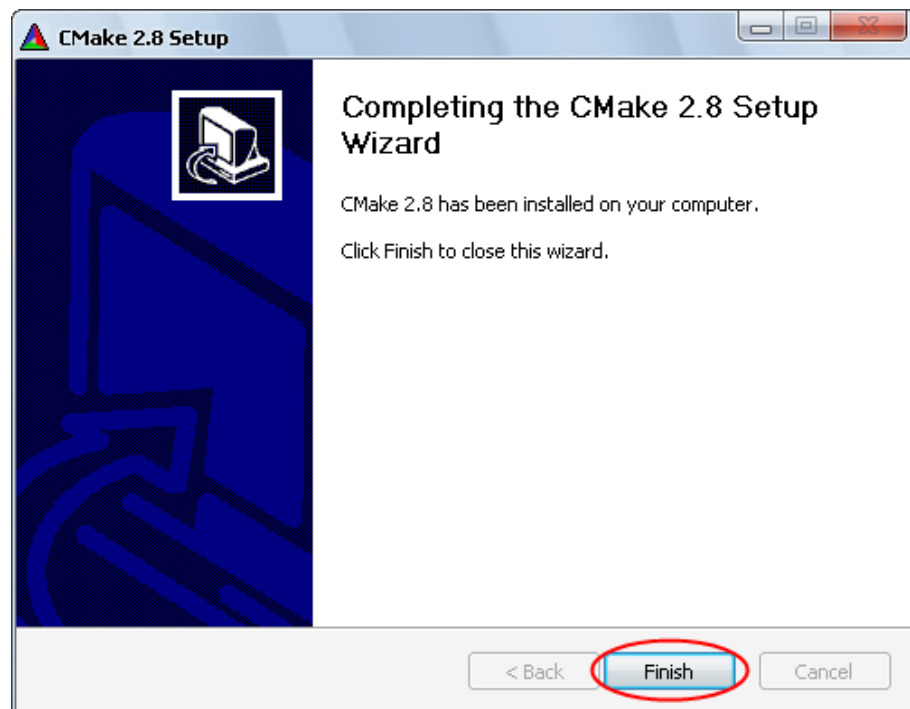
รูปภาคผนวก ข-20 หน้าต่างการติดตั้ง Cmake 2.8

7. จากนั้นรอนจนกว่าการติดตั้งจะเสร็จสมบูรณ์



รูปภาคผนวก ข-21 หน้าต่างแสดงว่ากำลังการติดตั้ง Cmake 2.8

8. เมื่อ Cmake 2.8 ติดตั้งเสร็จสมบูรณ์แล้วให้คลิก finish



รูปภาคผนวก ข-22 หน้าต่างแสดงว่าการติดตั้ง Cmake 2.8 เสร็จสมบูรณ์

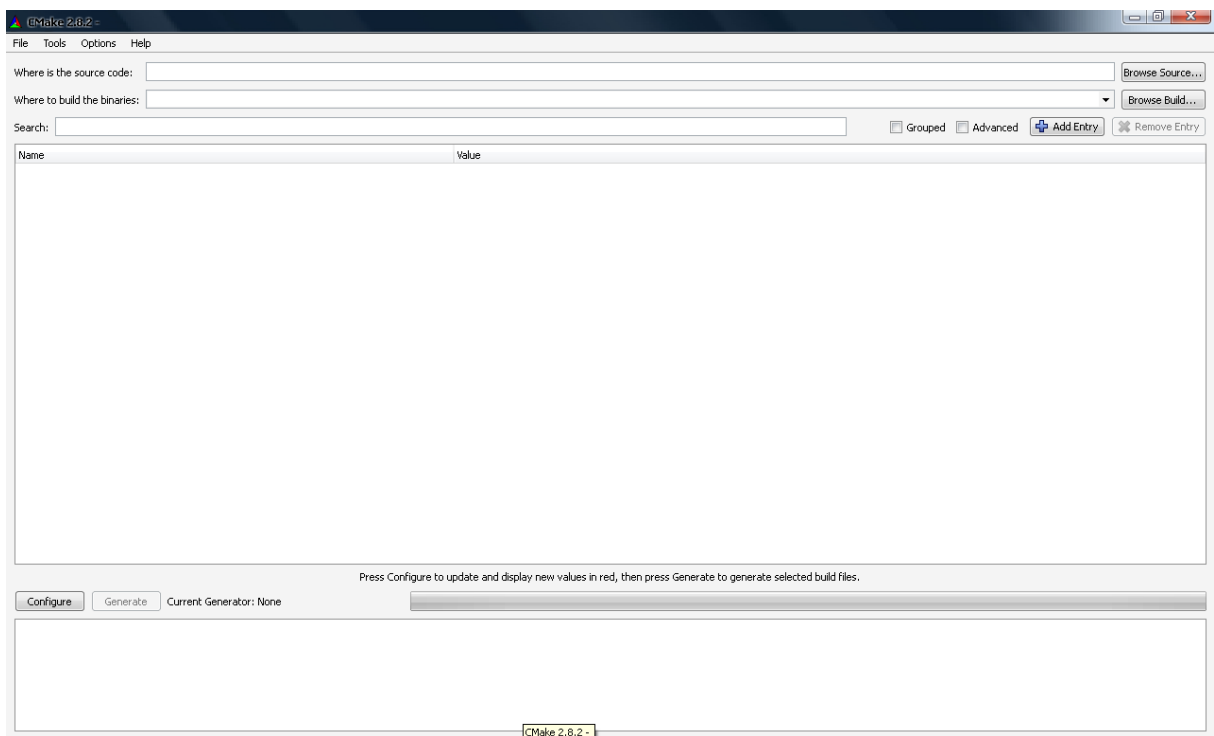
ภาคผนวก ค

วิธีการ compile OpenCV ด้วย CMake 2.8

เนื่องจากการติดตั้ง OpenCV เพียงอย่างเดียวนั้น ไม่สามารถทำให้นำมาใช้งานร่วมกับ Microsoft Visual Studio 2008 ได้ทันทีเราจึงใช้โปรแกรม CMake 2.8 มาช่วยในการ compile OpenCV เพื่อให้ใช้งานร่วมกันได้

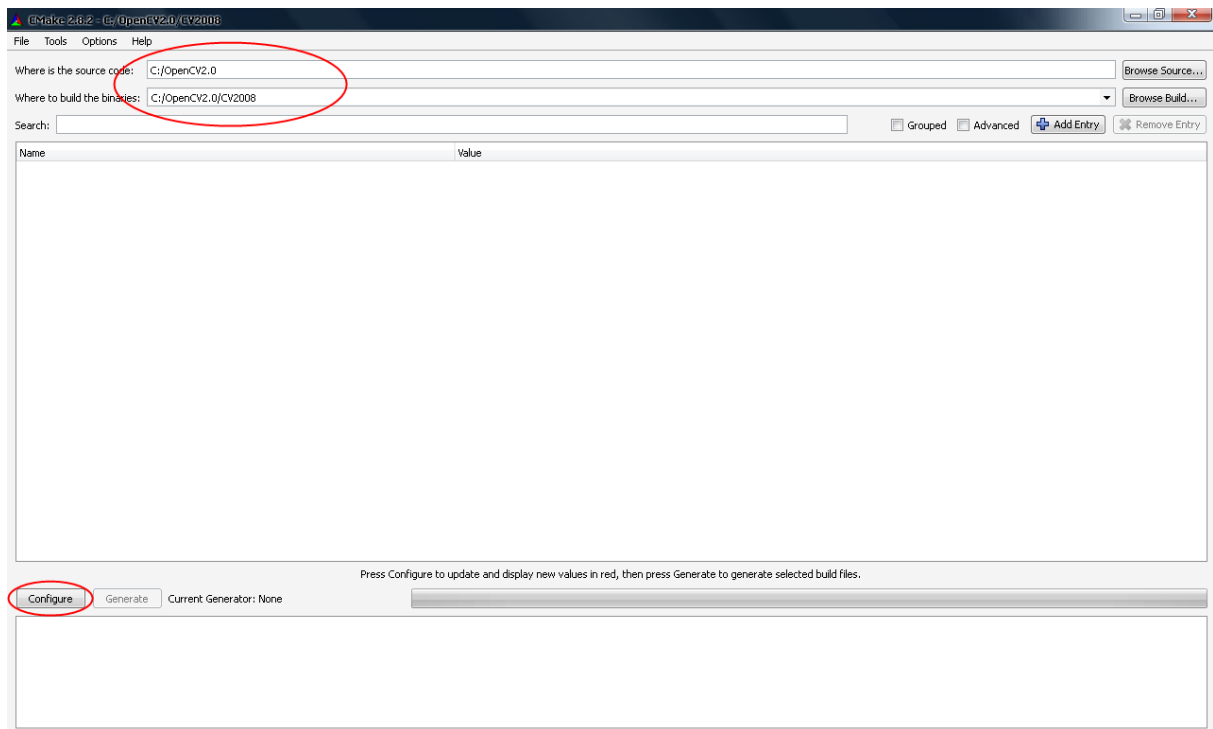
ขั้นตอนการ compile OpenCV ด้วย CMake 2.8

1. เปิดโปรแกรม CMake 2.8 ขึ้นมาจะพบลักษณะโปรแกรมดังภาพ



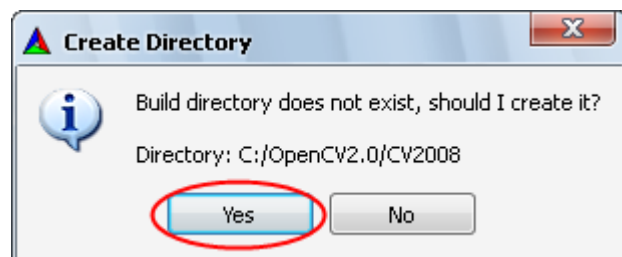
รูปภาคผนวก ค-1 ลักษณะโปรแกรม CMake 2.8

2. ทำการกำหนดตำแหน่งของ source code ของ OpenCV ที่ได้ติดตั้งไว้แล้วในที่นี้คือ "C:/OpenCV2.0" และตำแหน่งที่จะ build binaries ในที่นี้คือ "C:/OpenCV2.0/CV2008" ดังภาพ จากนั้นให้คลิก configure



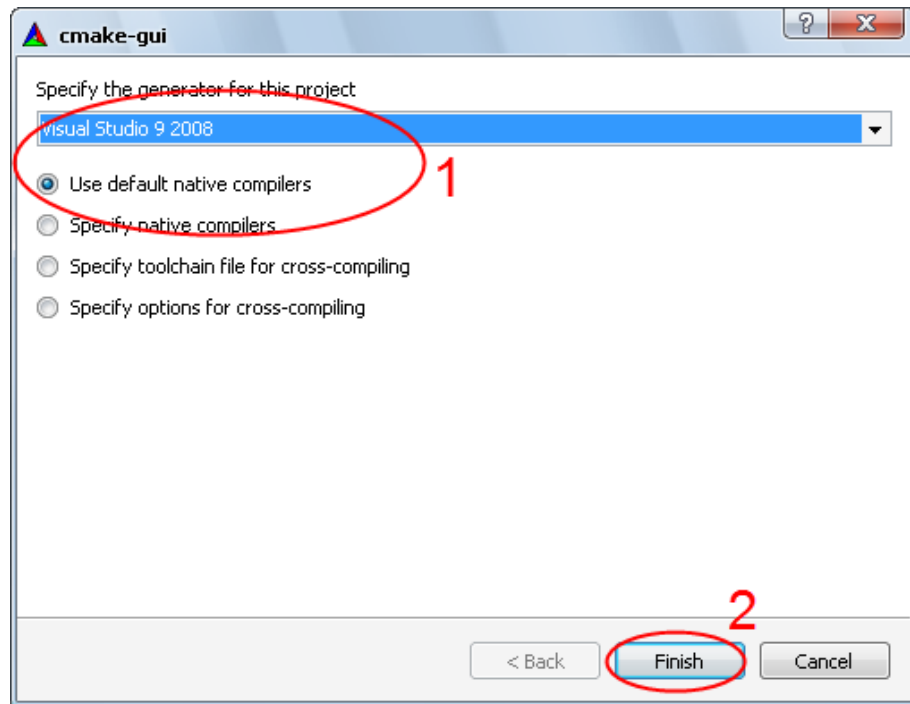
รูปภาพผนวก ค-2 การกำหนดตำแหน่ง source code และ build binaries ในโปรแกรม CMake 2.8

3. ถ้าตำแหน่งที่เราจะ build binaries ไม่มีอยู่จริง โปรแกรมจะถามเพื่อให้ยืนยันการสร้างที่อยู่ใหม่ ให้คลิก yes ดังภาพ



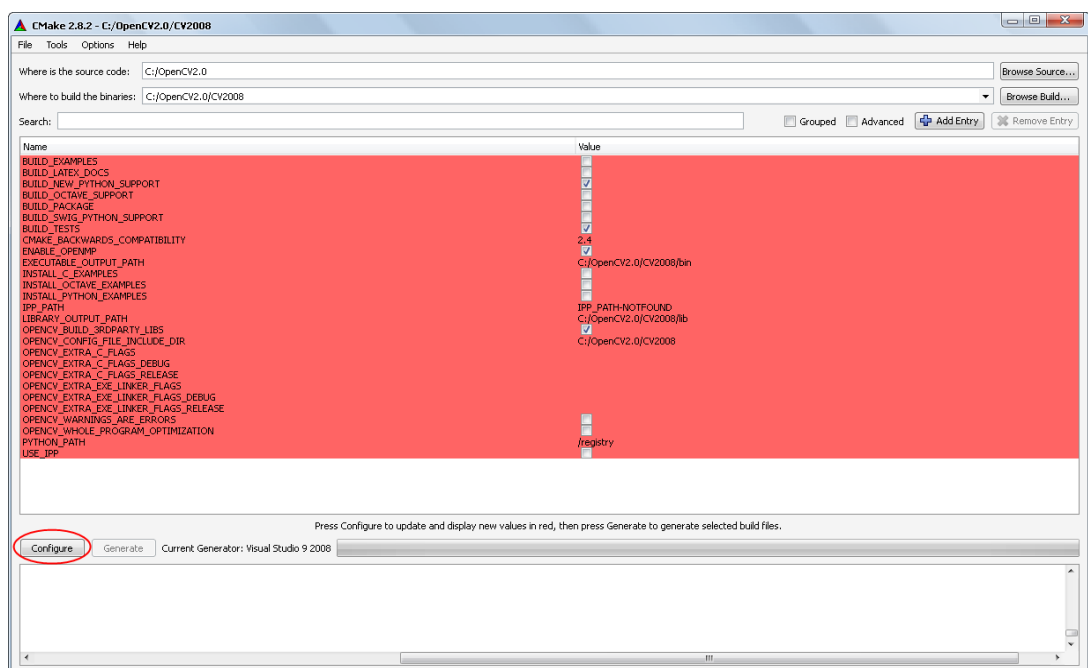
รูปภาพผนวก ค-3 การถามเพื่อยืนยันการสร้างที่อยู่ใหม่ในโปรแกรม CMake 2.8

4. หลังจากนั้นโปรแกรมจะให้เลือกตัวคอมไพเลอร์ที่กำลังใช้อยู่ ซึ่งในที่นี้คือ Visual Studio 9 2008 และเลือกรูปแบบการคอมไพล์เป็น Use default native compilers เสร็จแล้วคลิก Finish เพื่อดำเนินการต่อ



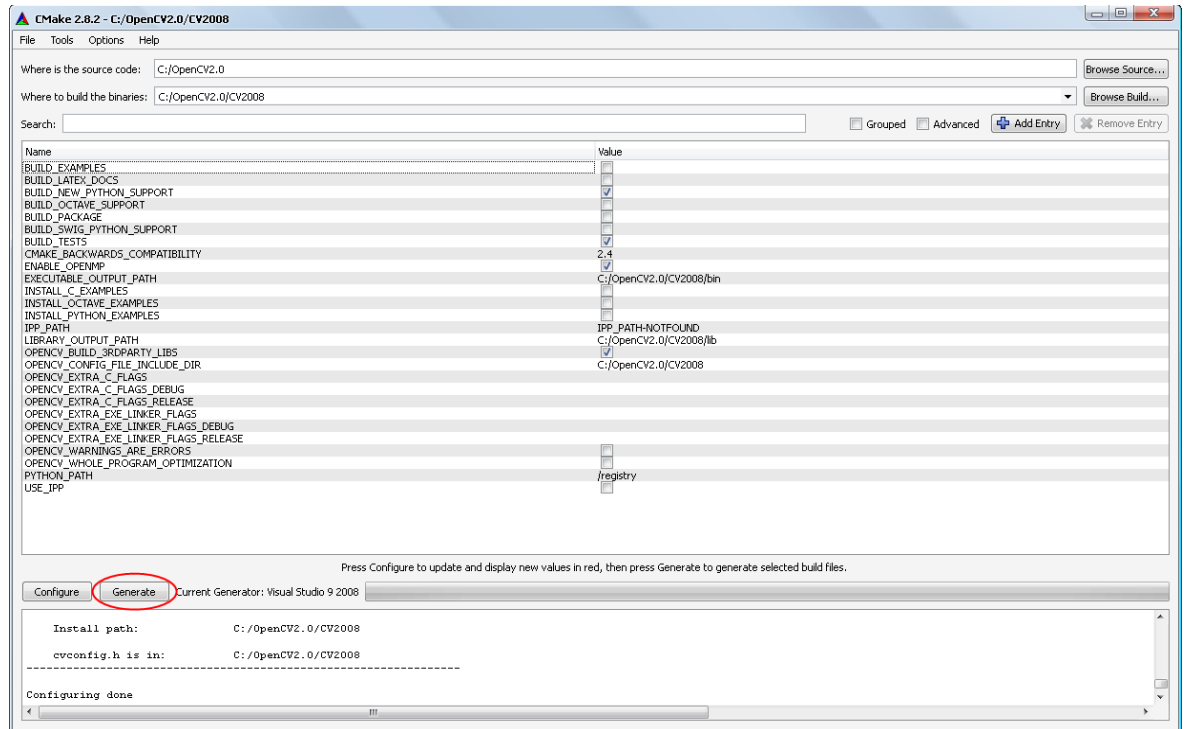
รูปภาพผนวก ค-4 การเลือกตัวคอมไพเลอร์และรูปแบบการคอมไพล์ในโปรแกรม CMake 2.8

5. โปรแกรมจะแสดงตัวเลือกที่ CMake 2.8 จะทำการสร้างไฟล์ขึ้นมา สามารถยกเลิกหรือเลือกเพิ่มเติมได้ตามความต้องการ แต่ในที่นี้จะไม่เปลี่ยนแปลงใดๆ ให้คลิก configure เพื่อดำเนินการต่อ



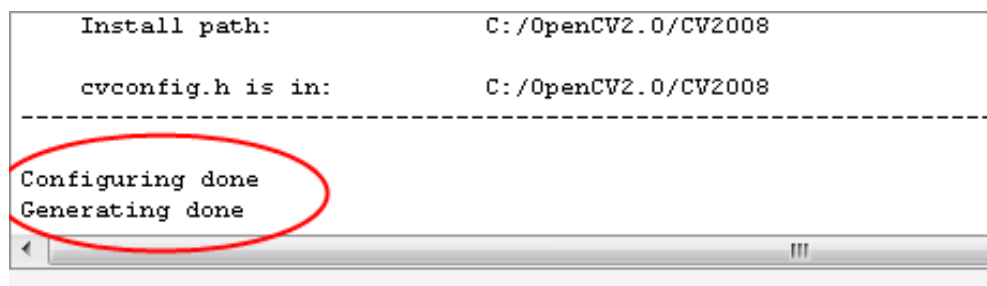
รูปภาพผนวก ค-5 การเลือกไฟล์ที่จะสร้างขึ้นมาโดยใช้โปรแกรม CMake 2.8

6. จากนั้นเริ่มสร้างไฟล์ต่างๆ โดยคลิก Generate ดังภาพ



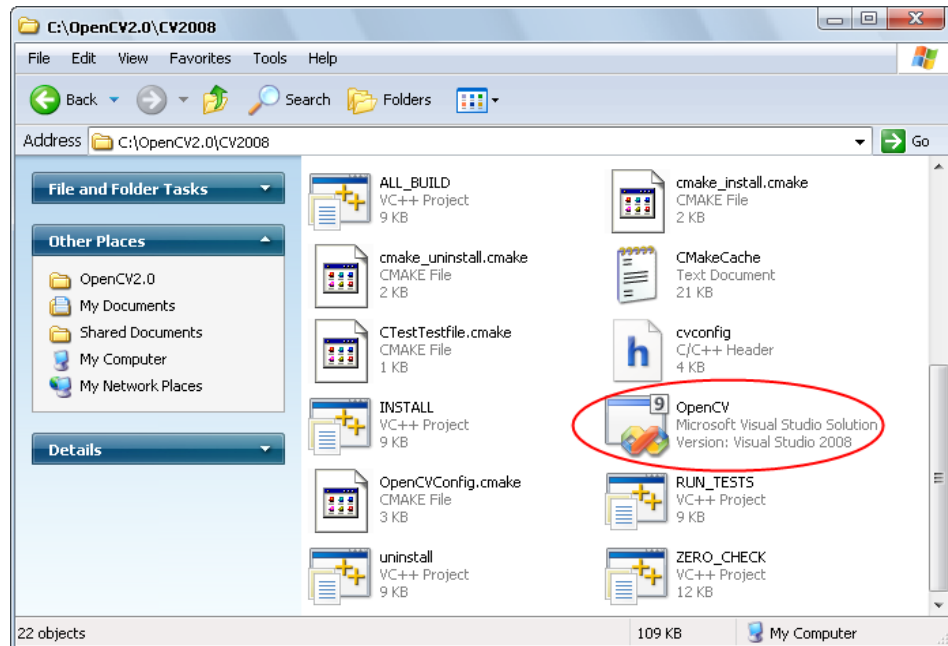
รูปภาพผนวก ค-6 การสร้างไฟล์ต่างๆโดยใช้โปรแกรม CMake 2.8

7. เมื่อไฟล์ต่างๆ ถูกสร้างเสร็จสมบูรณ์จะปรากฏข้อความ Generating done ดังภาพ จากนั้นสามารถปิดโปรแกรมได้เลย



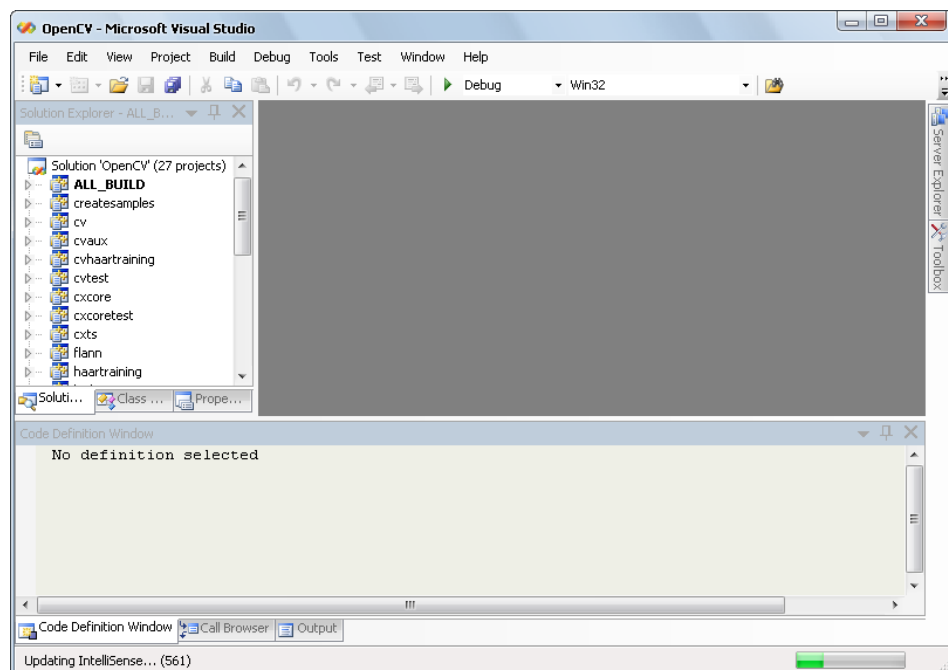
รูปภาพผนวก ค-7 การสร้างไฟล์ต่างๆโดยใช้โปรแกรม CMake 2.8

8. เริ่มทำการคอมไพล์ โดยให้เข้าไปยังโฟลเดอร์ที่เรา build binaries ไว้ในขั้นตอนที่ 2 ในที่นี้คือ "C:/OpenCV2.0/CV2008" จากนั้นให้เปิดไฟล์ OpenCV ด้วยโปรแกรม Microsoft Visual Studio 2008 ดังภาพ



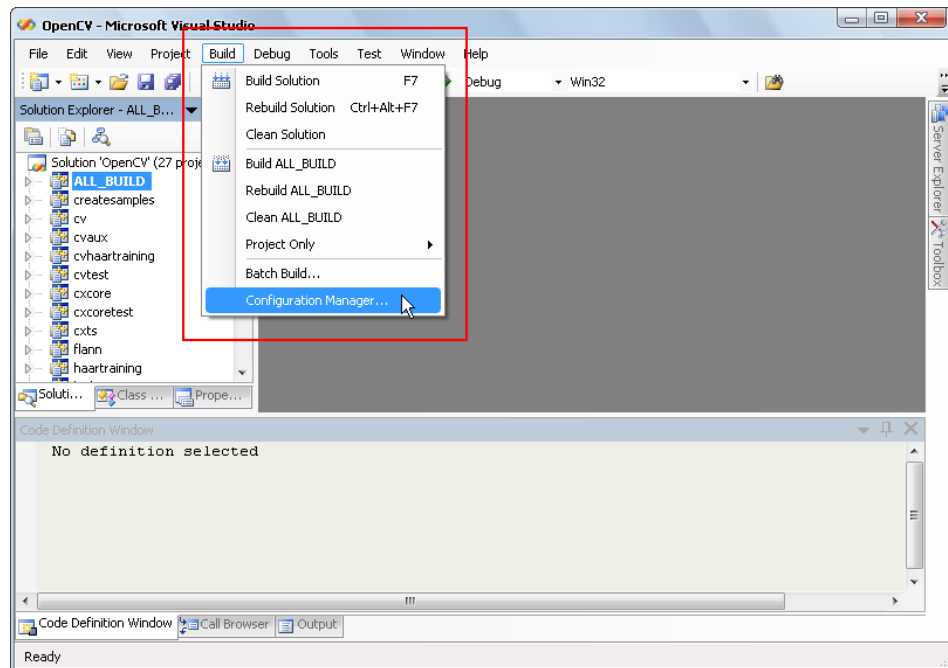
รูปภาพผนวก ค-8 ไฟล์ OpenCV ที่ได้จากการ build binaries

9. เมื่อเปิดไฟล์เรียบร้อยแล้ว จะปรากฏหน้าต่างโปรแกรมดังภาพ



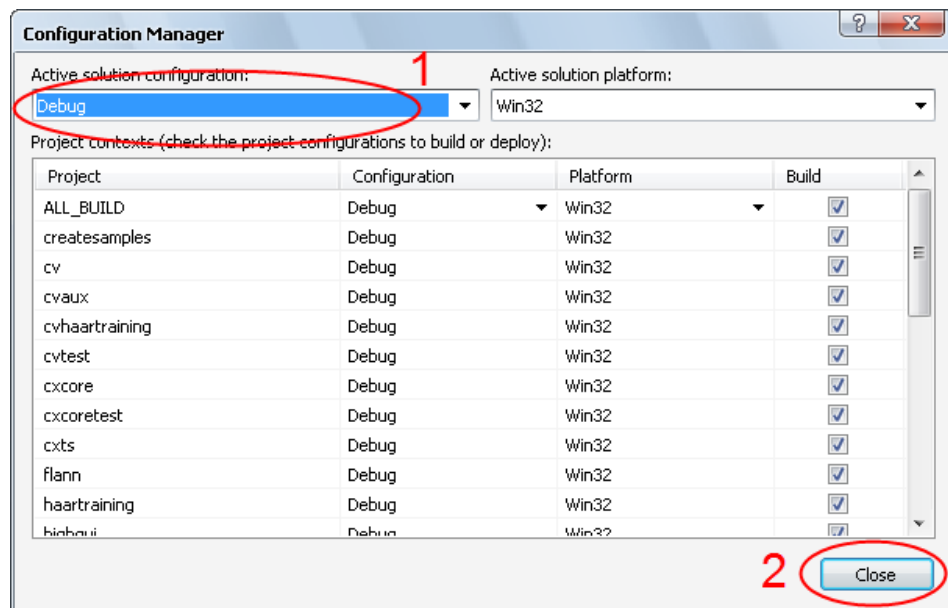
รูปภาพผนวก ค-9 การเปิดไฟล์ OpenCV ด้วยโปรแกรม Microsoft Visual studio 2008

10. เริ่มทำการกำหนดรูปแบบการสร้างไฟล์สำหรับการรันโปรแกรมในโหมด debug โดยการคลิกที่ Build ตามด้วย Configuration Manager



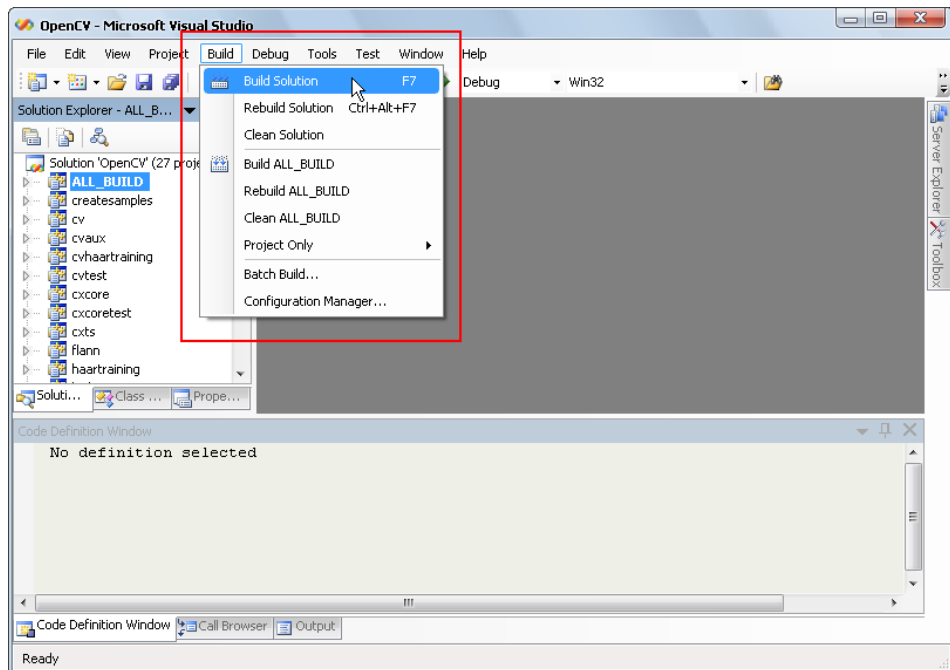
รูปภาพผนวก ค-10 การไปหน้า Configuration Manager ในโปรแกรม Microsoft Visual studio 2008

11. หลังจากเข้าสู่หน้า Configuration Manager แล้วให้ทำการปรับค่าที่ Active solution configuration ให้เป็น Debug เพื่อทำการสร้างไฟล์ที่ใช้ในกระบวนการรันในโหมด Debug จากนั้นคลิกปุ่ม Close



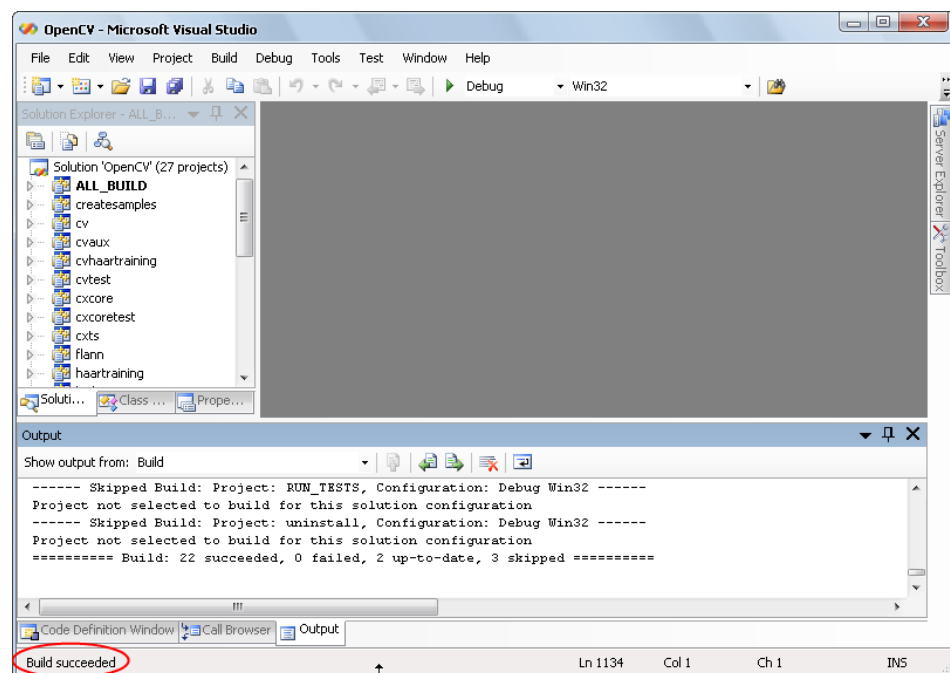
รูปภาพผนวก ค-11 การตั้งค่า Configuration Manager สำหรับรันโหมด Debug ในโปรแกรม Microsoft Visual studio 2008

12. ทำการสร้างไฟล์สำหรับการรันโหมด Debug โดยการคลิกที่ Build ตามด้วย Build Solution ดังภาพ



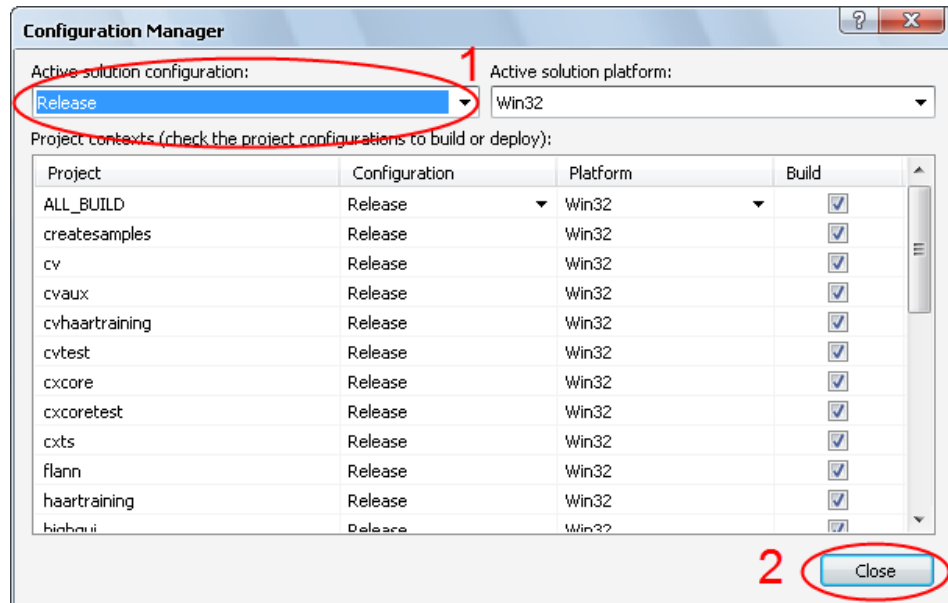
รูปภาพผนวก ค-12 การสร้างไฟล์ในการรันโหมด Debug

13. เมื่อทำการสร้างไฟล์ในโหมด Debug เสร็จเรียบร้อยแล้วจะปรากฏข้อความ Build succeeded ดังภาพ



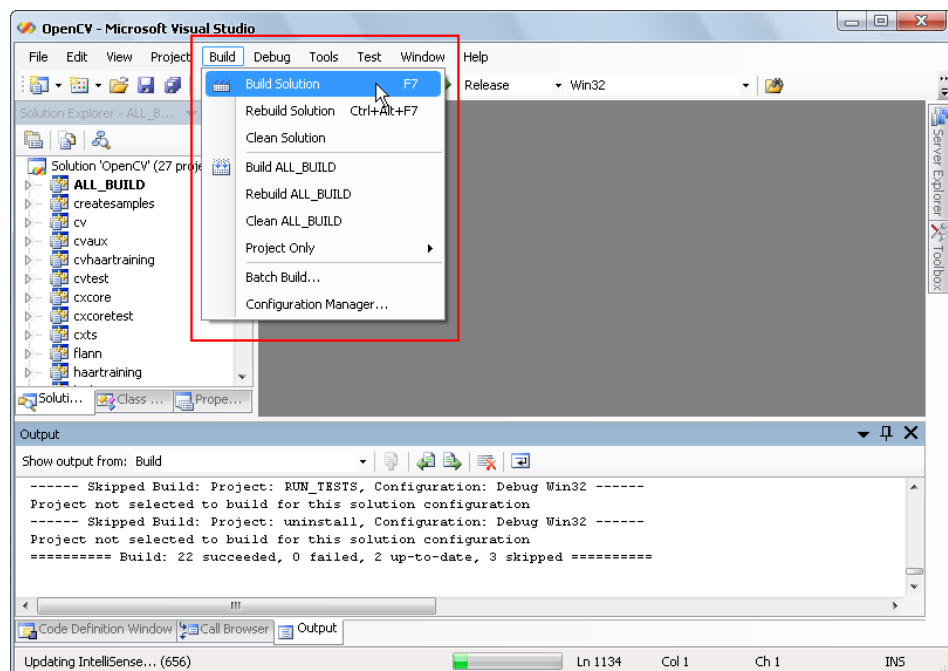
รูปภาพผนวก ค-13 แสดงการสร้างไฟล์ในโหมด Debug เสร็จเรียบร้อยแล้ว

14. จากนั้นให้ทำการกำหนดค่าในการสร้างไฟล์สำหรับรันโหมด Release ต่อไปโดยเข้าสู่หน้า Configuration Manager อีกครั้งเช่นเดียวกับในขั้นตอนที่ 10 แต่ปรับค่าที่ Active solution configuration ให้เป็น Release เพื่อทำการสร้างไฟล์ที่ใช้ในกระบวนการรันโหมด Release จากนั้นคลิกปุ่ม Close ดังภาพ



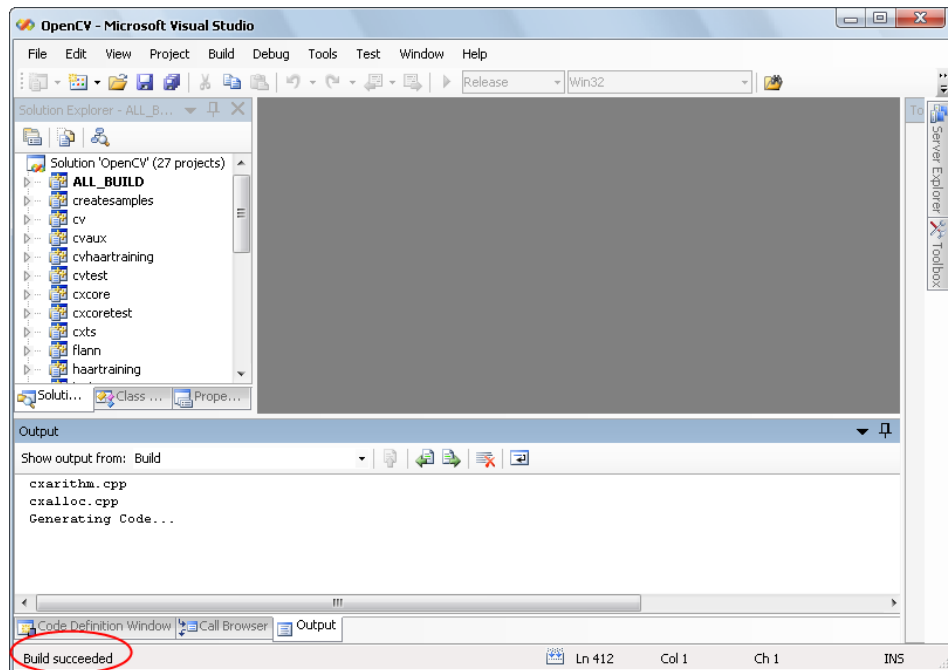
รูปภาพผนวก ค-14 การตั้งค่า Configuration Manager สำหรับรันโหมด Release

15. ทำการสร้างไฟล์สำหรับการรันโหมด Release โดยการคลิกที่ Build ตามด้วย Build Solution ดังภาพ



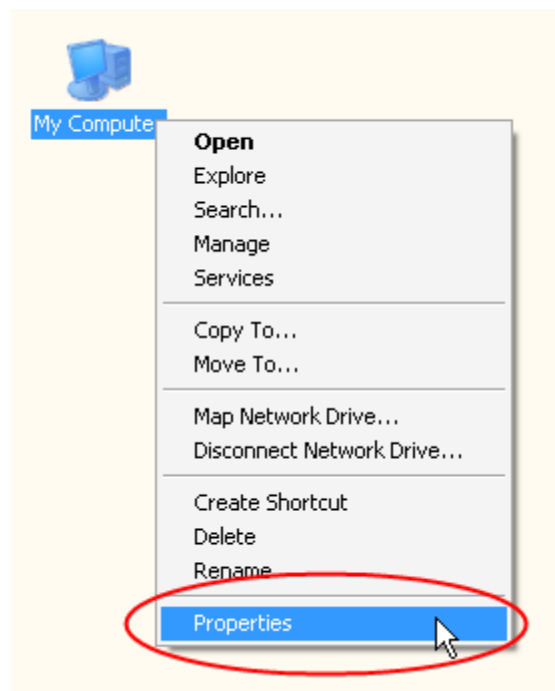
รูปภาพผนวก ค-15 การสร้างไฟล์ในการรันโหมด Release

16. เมื่อทำการสร้างไฟล์ในโหมด Debug เสร็จเรียบร้อยแล้วจะปรากฏข้อความ Build succeeded ดังภาพ



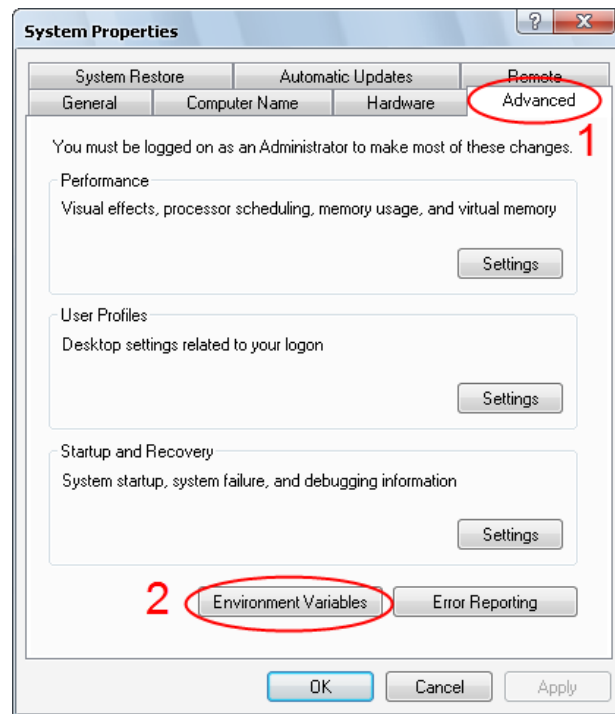
รูปภาพผนวก ค-16 แสดงการสร้างไฟล์ในโหมด Release เสร็จเรียบร้อยแล้ว

17. หลังจากการสร้างไฟล์ในโหมด Debug และ Release เสร็จเรียบร้อยแล้ว ต้องทำการกำหนด Path ของระบบเพื่อให้การทำงานเป็นไปอย่างถูกต้องโดยคลิกขวาที่ My Computer แล้วเลือก Properties ดังภาพ



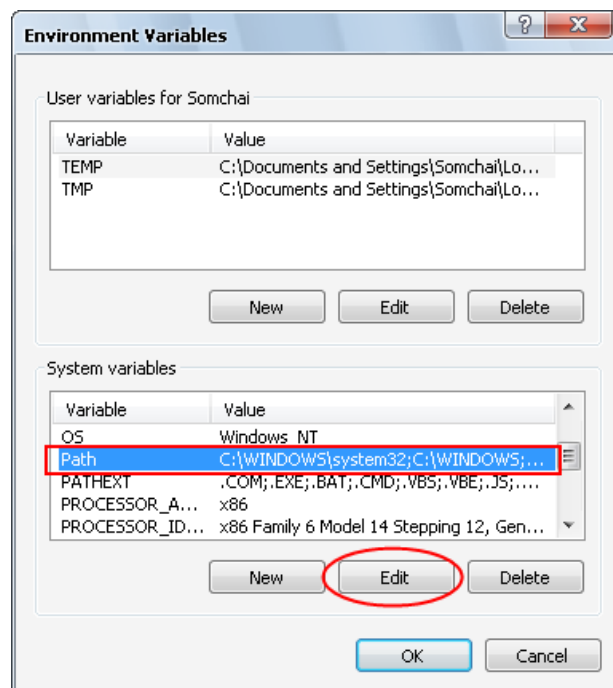
รูปภาพผนวก ค-17 แสดงการเข้าสู่ Properties ของ My Computer

18. เมื่อเข้าสู่หน้า Properties แล้วให้เลือก Advanced และคลิกที่ Environment Variables ดังภาพ



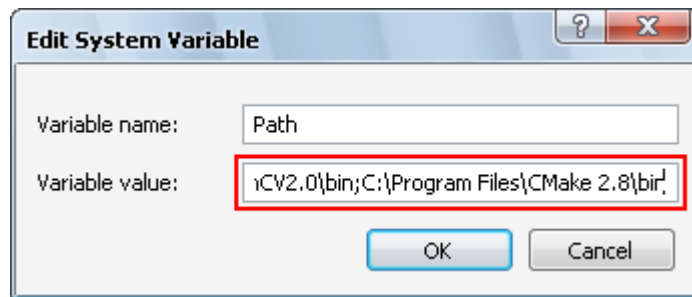
รูปภาพผนวก ค-18 แสดงการเข้าสู่หน้าต่าง Environment Variables ของ My Computer

19. จากนั้นจะปรากฏหน้าต่างของ Environment Variables ให้คลิกเลือก Path แล้วคลิก Edit ดังภาพ



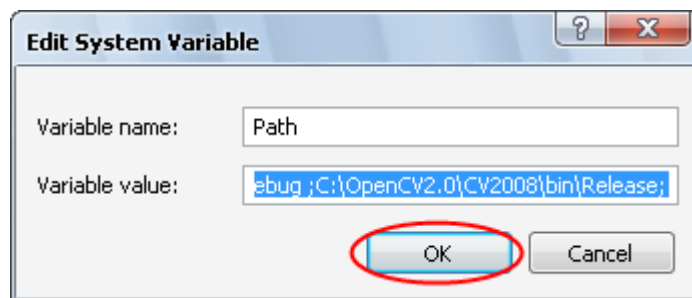
รูปภาพผนวก ค-19 แสดงการเข้าสู่การแก้ไข Path ในหน้าต่าง Environment Variables

20. หลังจากนั้นจะปรากฏหน้าต่าง Edit System Variables ให้เลื่อนเคอร์เซอร์ไปทางขวามือสุด เพื่อเตรียมการเพิ่ม Path ดังภาพ



รูปภาพผนวก ค-20 แสดงหน้าต่าง Edit System Variables

21. จากนั้นทำการเพิ่ม Path ต่อไปนี้ลงไป แล้วกด Ok ดังภาพ
;C:\OpenCV2.0\CV2008\bin\Debug ;C:\OpenCV2.0\CV2008\bin\Release; แล้วคลิก Ok ดังภาพ



รูปภาพผนวก ค-21 แสดงการเพิ่ม Path ในหน้าต่าง Edit System Variables

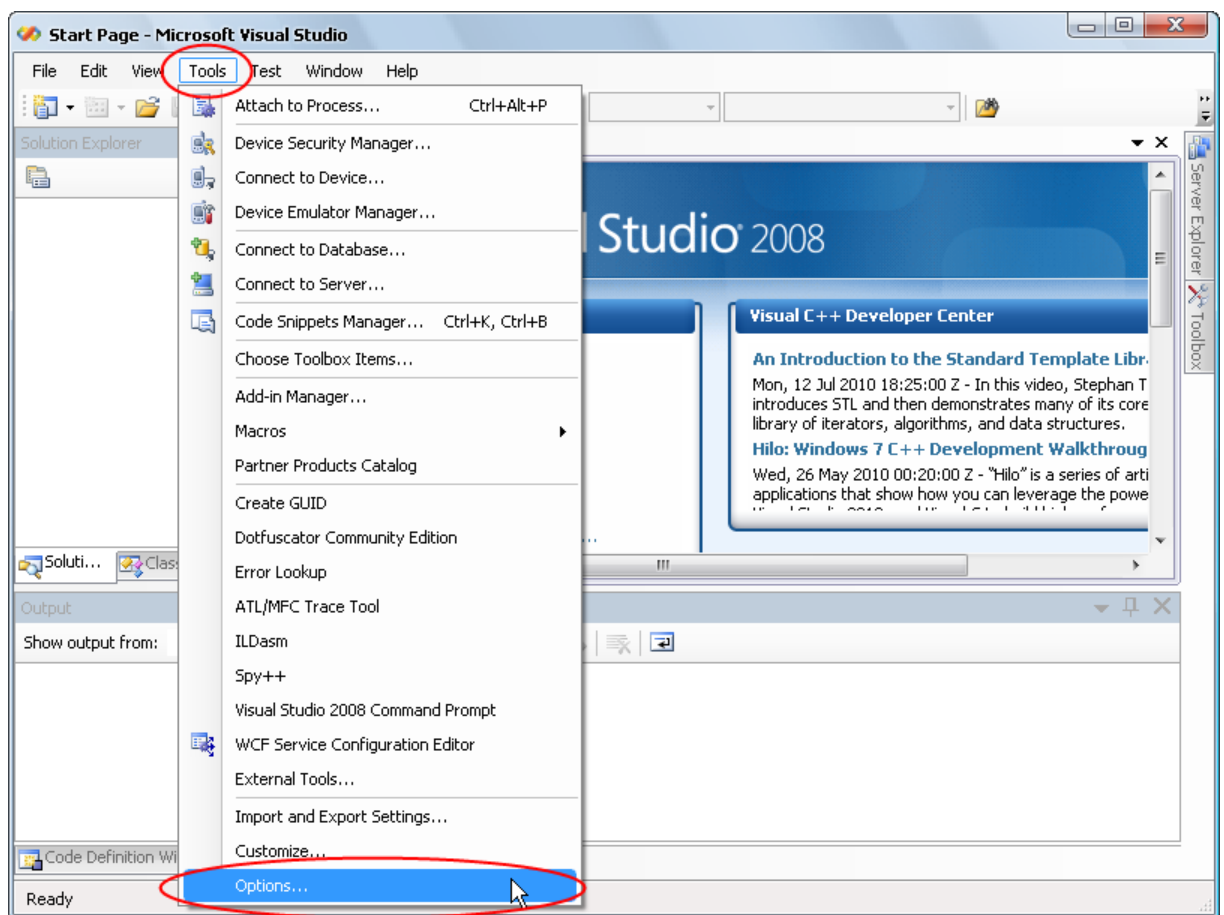
ภาคผนวก ง

วิธีการตั้งค่าให้กับ Microsoft Visual Studio 2008

และการสร้างโปรเจคใหม่

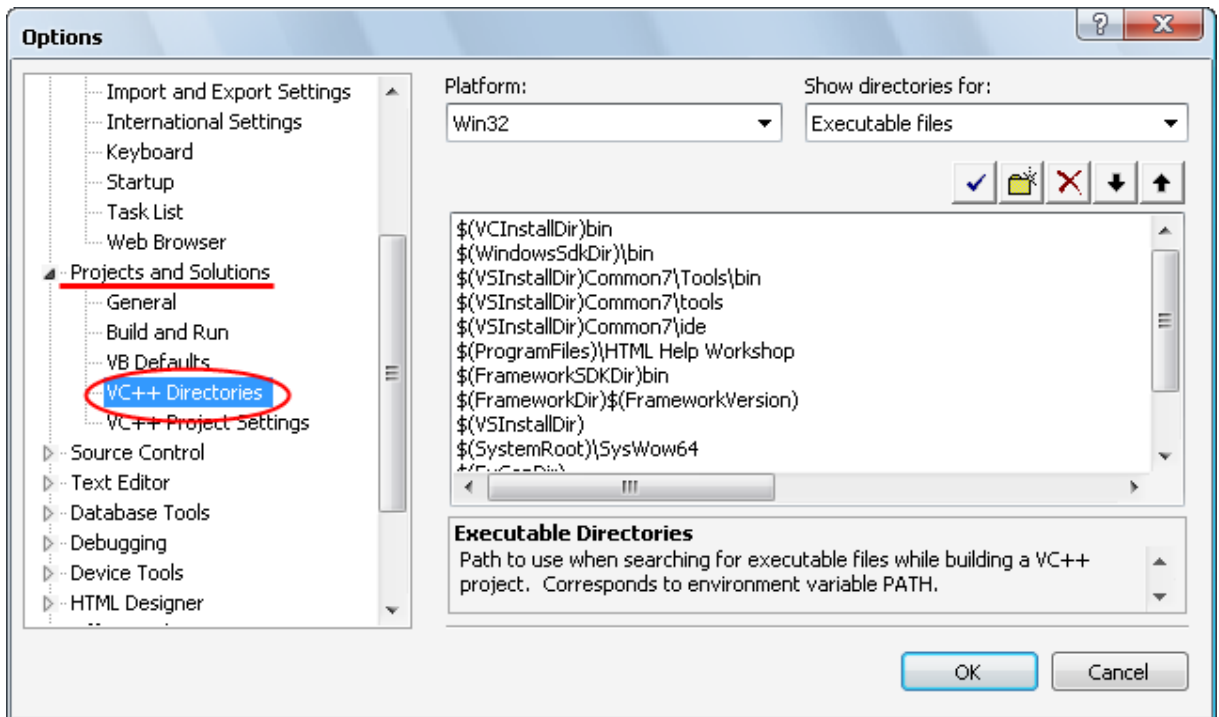
ขั้นตอนวิธีการตั้งค่าให้กับ Microsoft Visual Studio 2008

1. เปิดโปรแกรม Microsoft Visual Studio 2008 ขึ้นมา จากนั้นไปที่ Tools แล้วคลิก Options ดังภาพ



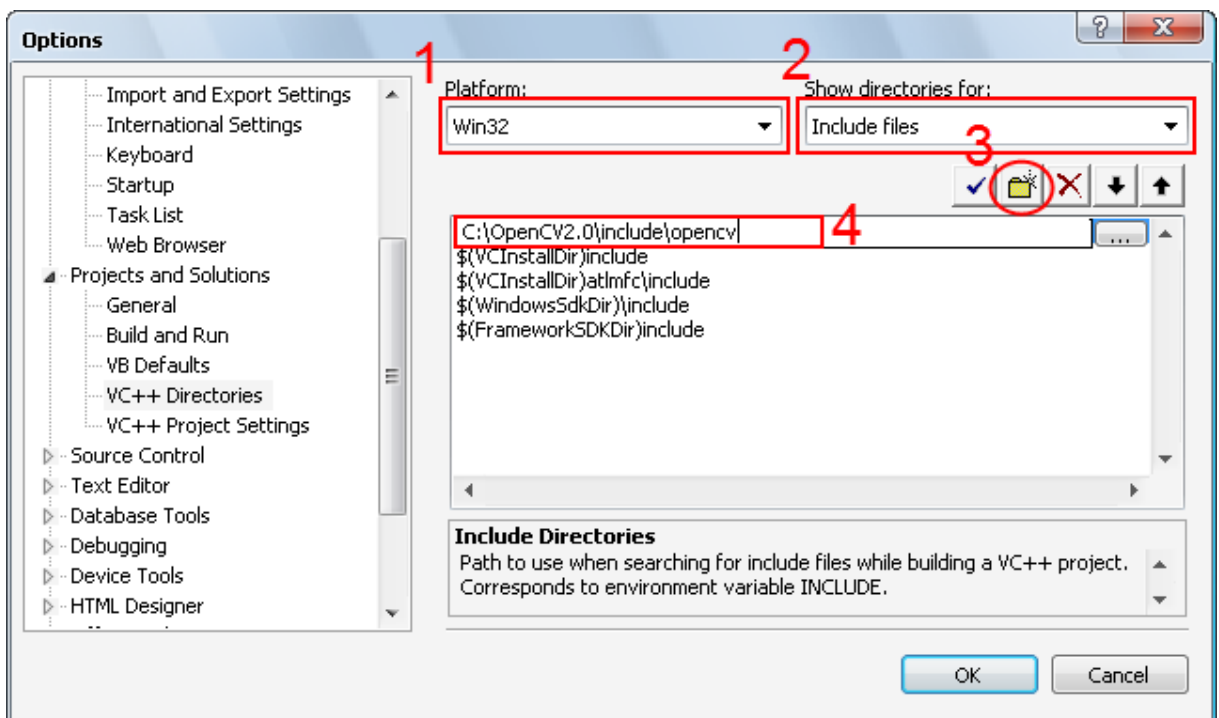
รูปภาคผนวก ง-1 แสดงการเข้าสู่หน้าต่าง Options ในโปรแกรม Microsoft Visual Studio 2008

2. จากนั้นคลิก Project and Solutions แล้วเลือก VC++ Directories ดังภาพ



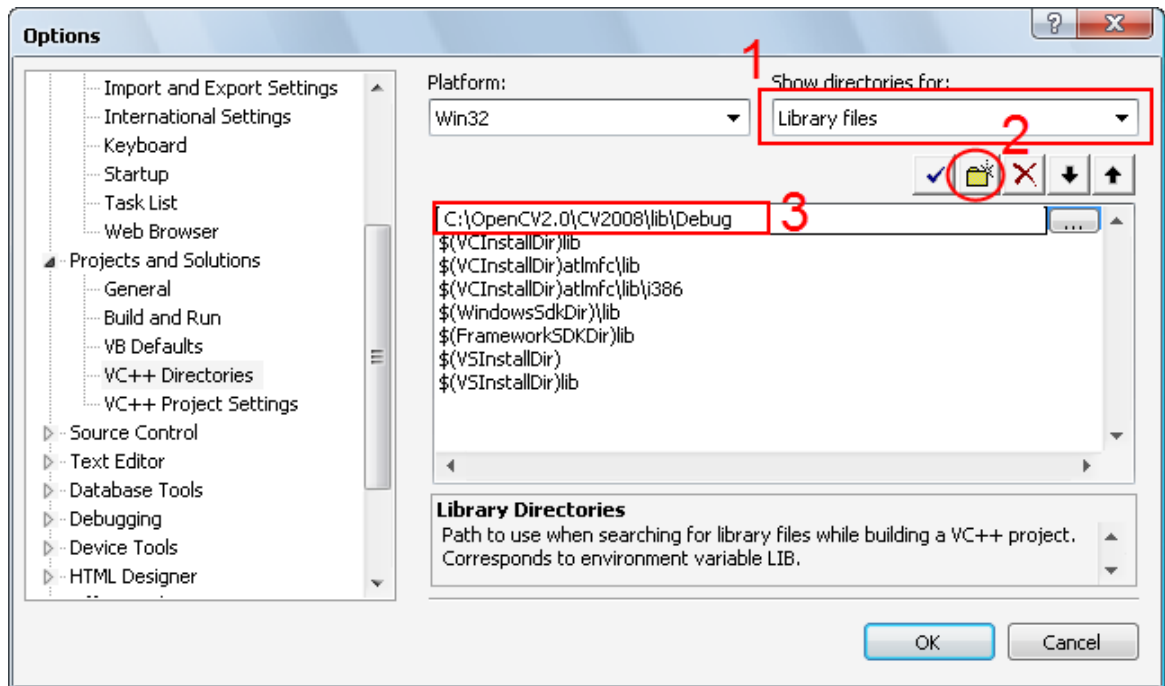
รูปภาคผนวก ง-2 แสดงการเลือก VC++ Directories ในโปรแกรม Microsoft Visual Studio 2008

3. จากนั้นให้ทำการกำหนด Path ของไฟล์ Include จาก OpenCV ดังภาพ



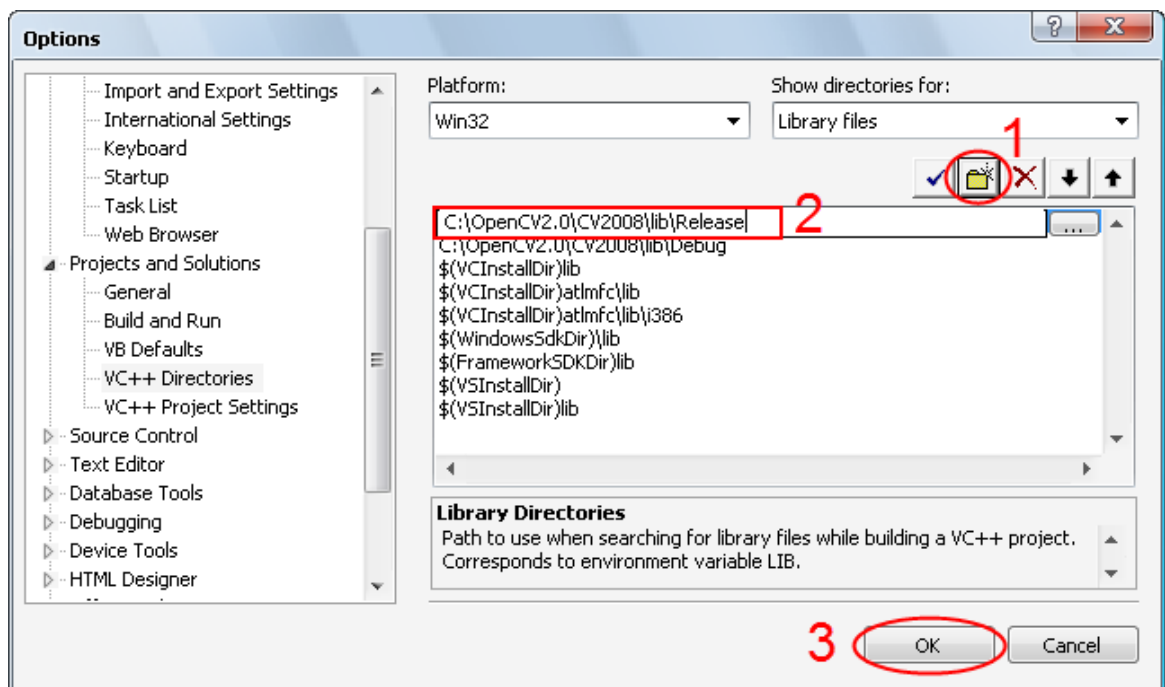
รูปภาคผนวก ง-3 แสดงการกำหนด Path ของไฟล์ Include จาก OpenCV

4. จากนั้นให้ทำการกำหนด Path ของไฟล์ Libraries สำหรับโหมด Debug จาก OpenCV ดังภาพ



รูปภาพผนวก ง-4 แสดงการกำหนด Path ของไฟล์ Libraries สำหรับโหมด Debug จาก OpenCV

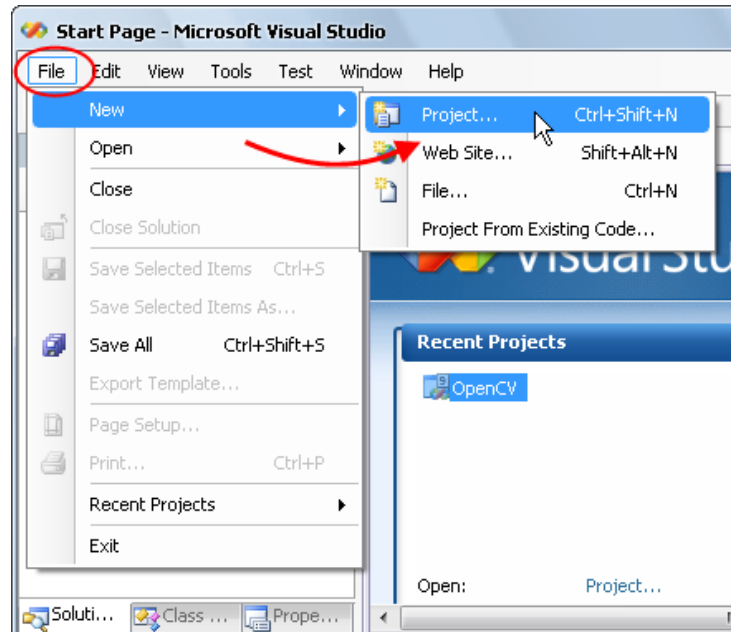
5. จากนั้นให้ทำการกำหนด Path ของไฟล์ Libraries สำหรับโหมด Release จาก OpenCV แล้วคลิก OK เพื่อสิ้นสุดการตั้งค่าให้กับโปรแกรม Microsoft Visual Studio 2008 ดังภาพ



รูปภาพผนวก ง-5 แสดงการกำหนด Path ของไฟล์ Libraries สำหรับโหมด Release จาก OpenCV

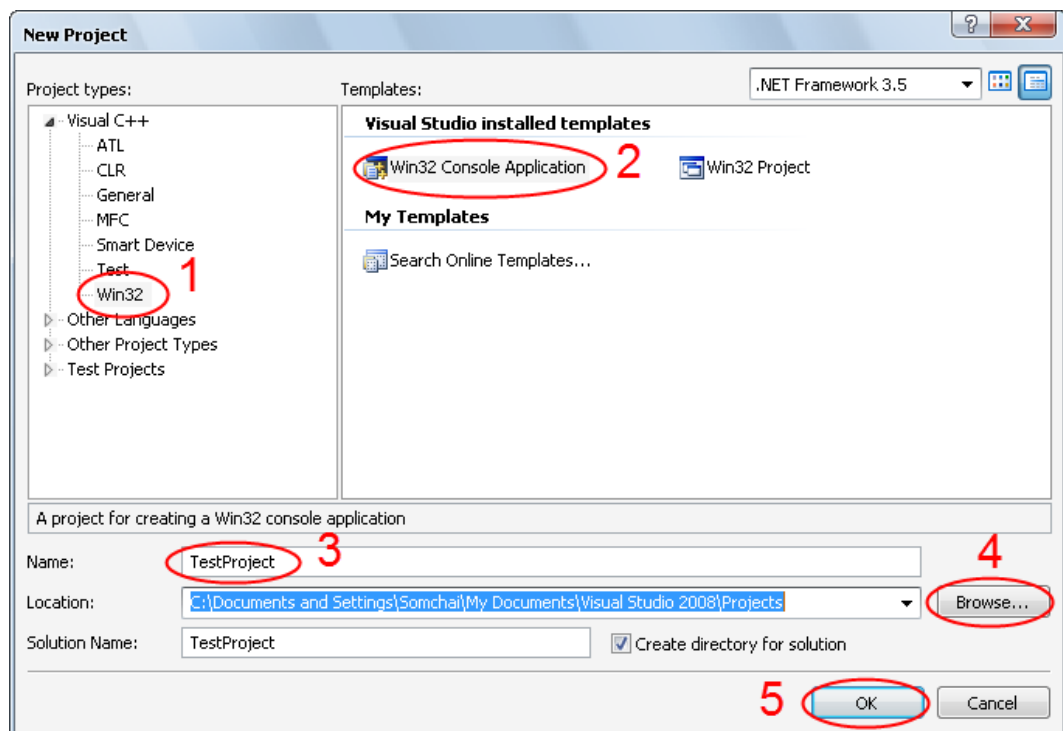
ขั้นตอนวิธีการสร้างโปรเจกใหม่ใน Microsoft Visual Studio 2008

1. เปิดโปรแกรม Microsoft Visual Studio 2008 จากนั้นคลิก File -> New -> Project ดังภาพ



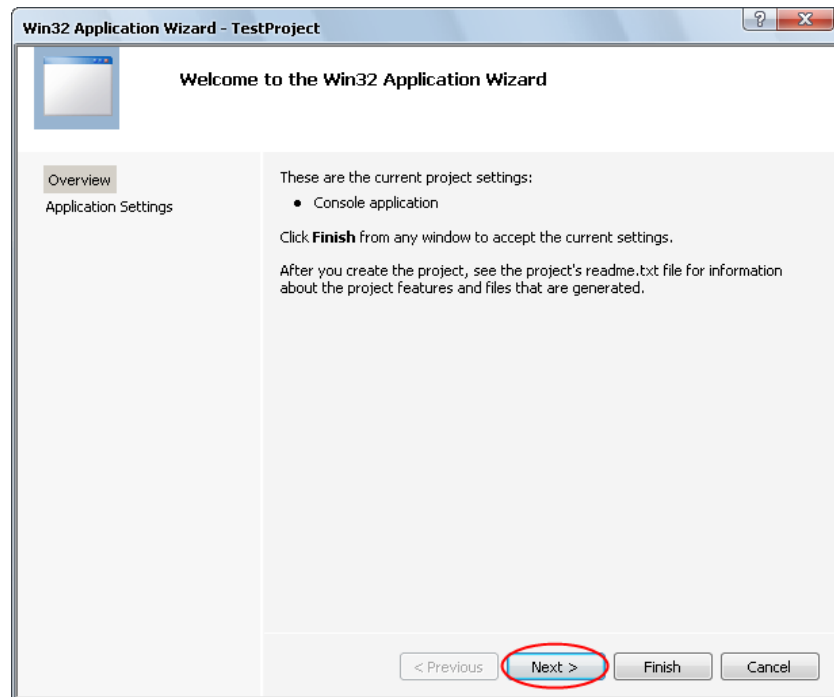
รูปภาพผนวก ง-6 แสดงการสร้างโปรเจกใหม่ในโปรแกรม Microsoft Visual Studio 2008

2. หลังจากนั้นจะปรากฏหน้าต่างให้เราเลือกรูปแบบของโปรเจก เลือก Win32 แล้วคลิก Win32 Console Application จากนั้นตั้งชื่อ และเลือกตำแหน่งของโปรเจก แล้วคลิก OK ดังภาพ



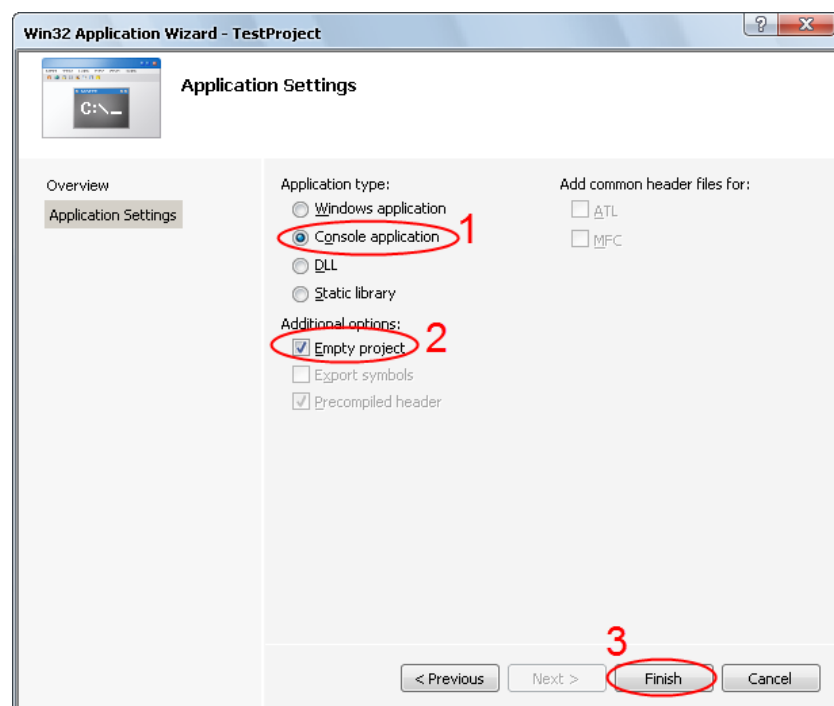
รูปภาพผนวก ง-7 แสดงการเลือกรูปแบบโปรเจกในโปรแกรม Microsoft Visual Studio 2008

3. โปรแกรมจะปรากฏหน้าต่าง Win32 Application Wizard ขึ้นมาให้คลิก Next ดังภาพ



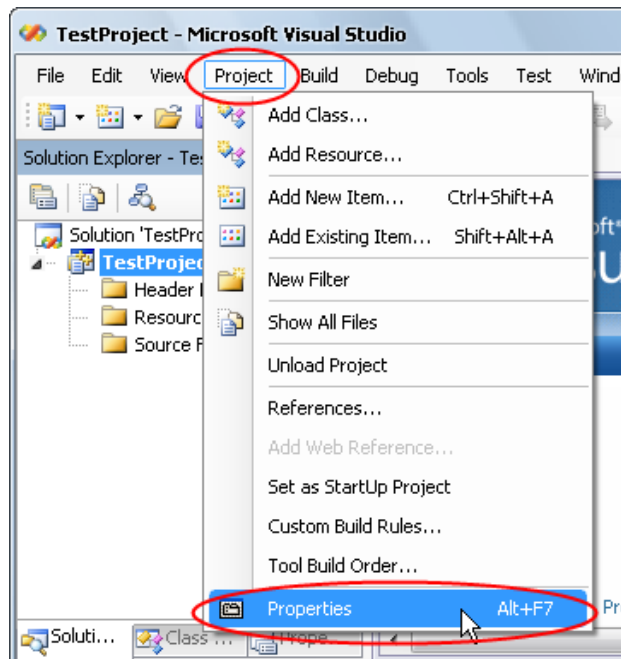
รูปภาพผนวก ง-8 หน้าต่าง Win32 Application Wizard ในโปรแกรม Microsoft Visual Studio 2008

4. จากนั้นให้เลือก Application type เป็น Console application และ Additional options เป็น Empty project แล้วคลิก Finish ดังภาพ



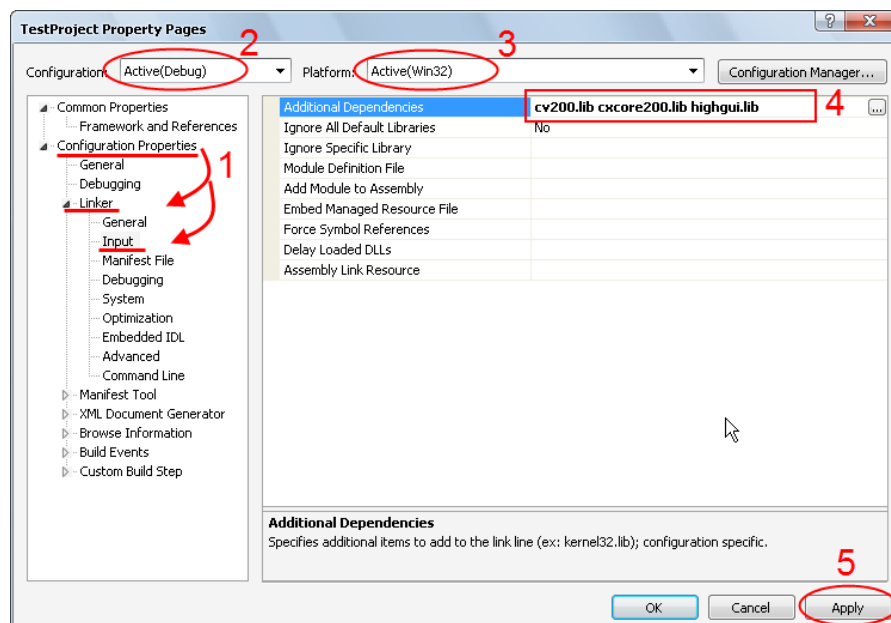
รูปภาพผนวก ง-9 การเลือก Application type และ Additional options

5. ทำการกำหนดคุณสมบัติของโปรเจค โดยไปที่ Project แล้วเลือก Properties ดังภาพ



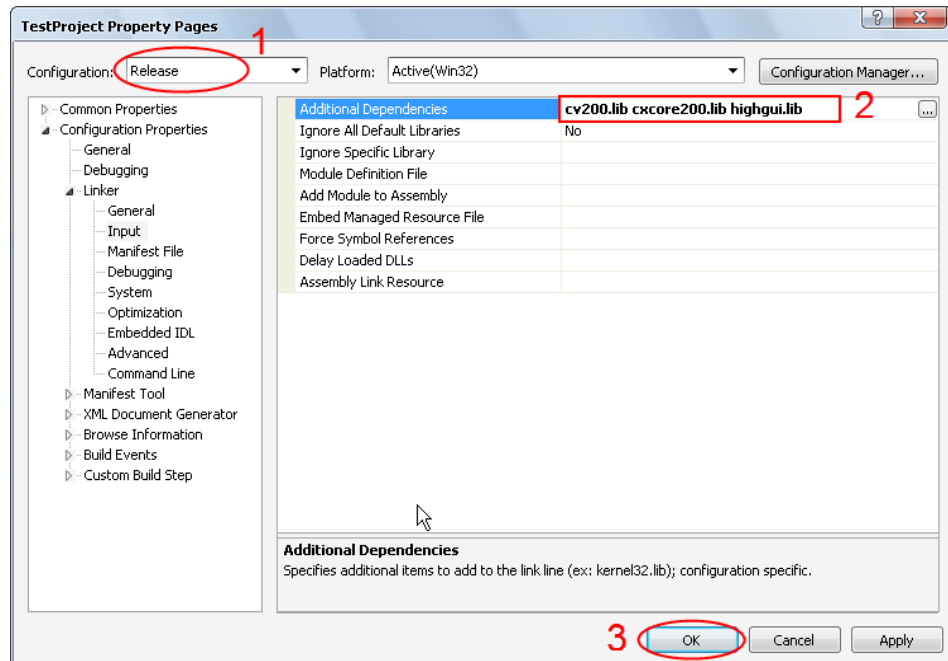
รูปภาพผนวก ง-10 การเข้าสู่การกำหนดคุณสมบัติโปรเจค

6. ให้กำหนด Additional Dependencies ไฟล์ สำหรับโหมด Debug โดยการคลิกเลือกในส่วนต่างๆดังภาพ และกำหนดค่าในช่อง Additional Dependencies เป็น cv200.lib cxcore200.lib highgui.lib จากนั้นคลิกปุ่ม Apply ดังภาพ



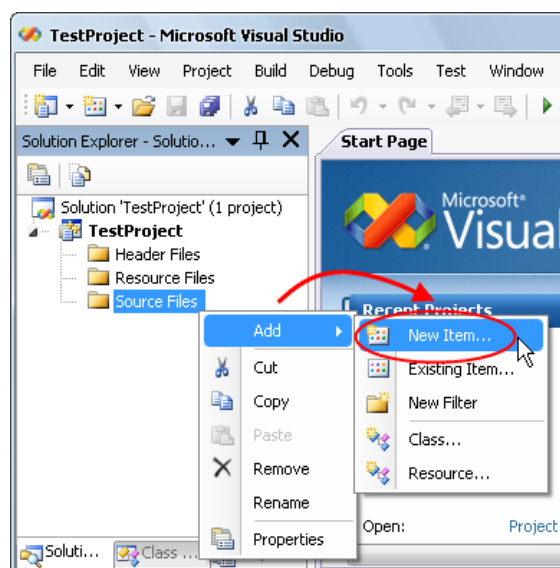
รูปภาพผนวก ง-11 การกำหนด Additional Dependencies ไฟล์ สำหรับโหมด Debug

7. จากนั้นให้กำหนด Additional Dependencies ไฟล์ สำหรับโหมด Release โดยการคลิกเลือกในส่วนต่างๆ ดังภาพ และกำหนดค่าในช่อง Additional Dependencies เป็น cv200.lib cxcore200.lib highgui.lib จากนั้นคลิก OK ดังภาพ



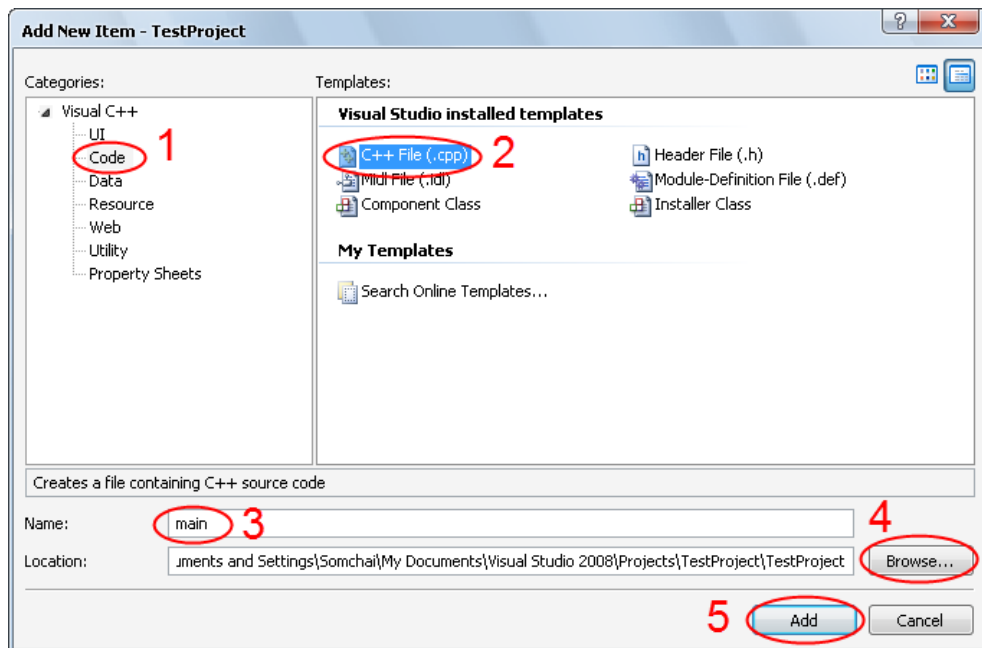
รูปภาพผนวก ง-12 การกำหนด Additional Dependencies ไฟล์ สำหรับโหมด Release

8. ทำการเพิ่มไฟล์ใหม่เพื่อใช้ในการเขียนโปรแกรม โดยการคลิกขวาที่ Source Files เลือก Add จากนั้นเลือก New Item ดังภาพ



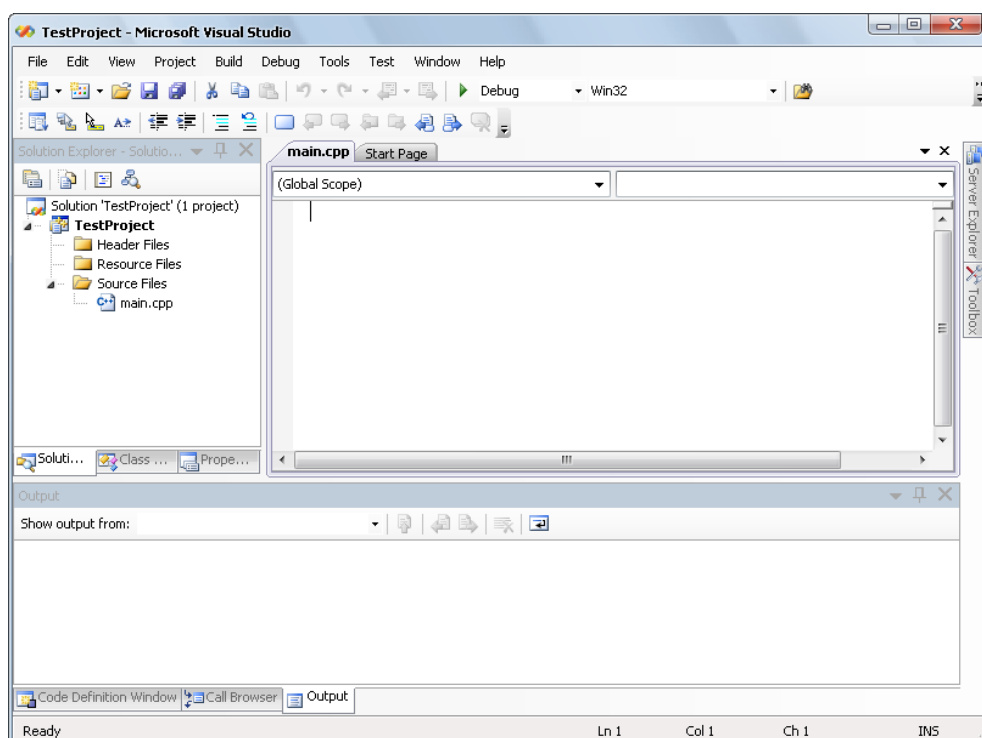
รูปภาพผนวก ง-13 การเพิ่มไฟล์ใหม่ในโปรแกรม Microsoft Visual Studio 2008

9. จากนั้นเลือกที่ Code และ C++ File (.cpp) จากนั้นให้กำหนดชื่อไฟล์ หากต้องการเปลี่ยนที่บันทึกไฟล์ให้คลิกที่ปุ่ม Browse แล้วคลิก Add ดังภาพ



รูปภาคผนวก ง-14 การกำหนดประเภทไฟล์ของโปรเจกต์ในโปรแกรม Microsoft Visual Studio 2008

10. พบว่ามีไฟล์ .cpp ของเรา ปรากฏอยู่ในโฟลเดอร์ Source Files และสังเกตเห็นเคอร์เซอร์กระพริบอยู่ พร้อมให้เราทำการเขียนโปรแกรมแล้วดังภาพ



รูปภาคผนวก ง-15 หน้าต่างแสดงการเริ่มต้นเขียนโปรแกรม

ภาคผนวก จ

คู่มือการติดตั้งโปรแกรม Eye Tracking Mouse เพื่อใช้งานจริง

ในภาคผนวก ข นี้จะแสดงขั้นตอนวิธีการลงโปรแกรม Eye Tracking Mouse ที่สมบูรณ์พร้อมใช้งานจริงบนเครื่องคอมพิวเตอร์ดังนี้

ขั้นตอนวิธีการติดตั้งโปรแกรม Eye Tracking Mouse

1. ดับเบิลคลิกที่ไอคอนตัวติดตั้งของ Eye Tracking Mouse ดังภาพ



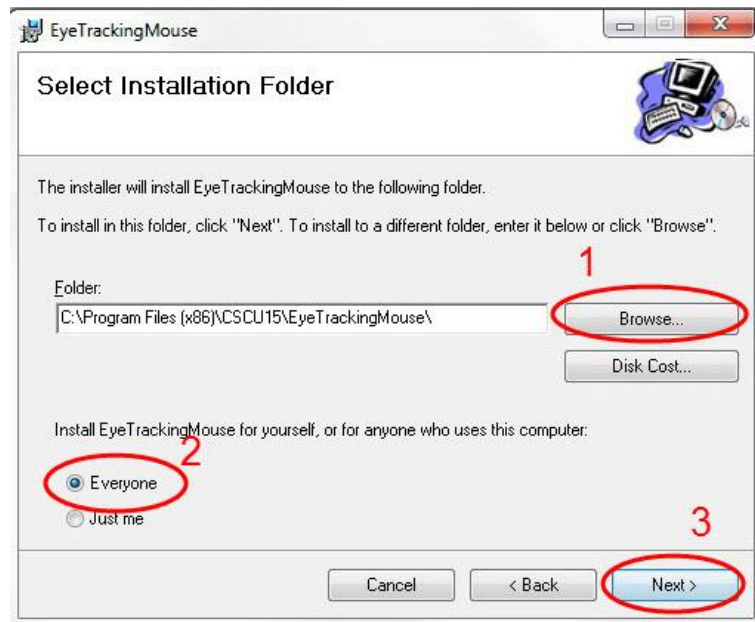
รูปภาคผนวก จ-1 ไอคอนตัวติดตั้งของโปรแกรม Eye Tracking Mouse

2. จะปรากฏหน้าต่างแสดงการต้อนรับเข้าสู่การติดตั้งโปรแกรมให้คลิก Next เพื่อดำเนินการต่อ



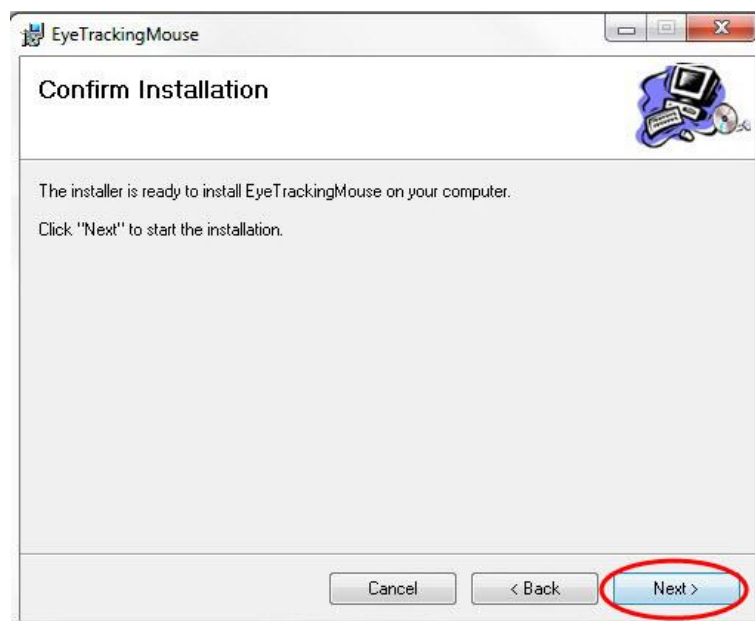
รูปภาคผนวก จ-2 หน้าต่างแสดงการต้อนรับของโปรแกรม Eye Tracking Mouse

3. หลังจากนั้นจะแสดงหน้าต่างให้เลือกตำแหน่งที่จะติดตั้ง Eye Tracking Mouse และเลือกการใช้งานสำหรับทุกคนโดยเลือก Everyone และคลิก Next เพื่อดำเนินการต่อติดตั้งภาพ



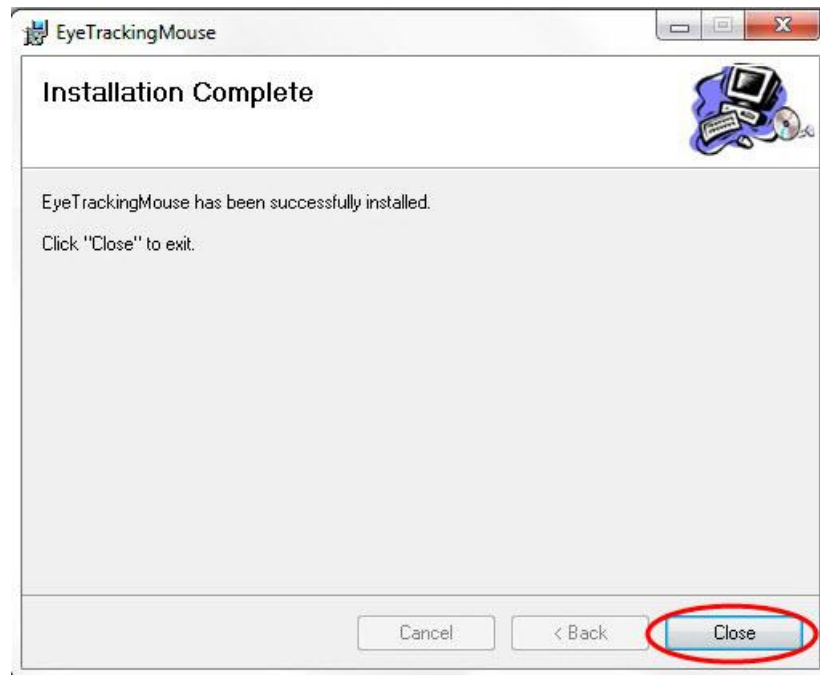
รูปภาคผนวก จ-3 หน้าต่างเลือกตำแหน่งการติดตั้ง Eye Tracking Mouse

4. จากนั้นจะปรากฏหน้าต่างยืนยันการติดตั้งโปรแกรมอีกครั้ง ให้คลิก Next เพื่อเริ่มทำการติดตั้งโปรแกรม



รูปภาคผนวก จ-4 หน้าต่างยืนยันการติดตั้ง Eye Tracking Mouse

5. หลังจากโปรแกรมติดตั้งสมบูรณ์จะปรากฏหน้าต่างดังภาพ ให้คลิก Close



รูปภาคผนวก จ-5 หน้าต่างแสดงการติดตั้ง Eye Tracking Mouse สมบูรณ์

6. เริ่มการใช้งานโปรแกรม Eye Tracking Mouse ได้ทันทีโดยดับเบิลคลิกที่ไอคอนบนหน้าเดสก์ทอปดังภาพ



รูปภาคผนวก จ-6 ไอคอนโปรแกรม Eye Tracking Mouse