

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การตอบสนองต่อวัตถุเสมือนจริงจากคอมพิวเตอร์

Interactive Visual Reality



T104230

โดย

นายเลชาติ

งามชูเกียรติ

นายสกลกร

กันฐักแก้ว

นายสุชาวัชร

อุดมมังกร

๒พ.
ค 8160
2551

เลขหมู่.....
เลขทะเบียน...104230
วัน,เดือน,ปี... 30 ต.ค. 2552

12110772

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตอบสนองต่อวัตถุเสมือนจริงจากคอมพิวเตอร์

Interactive Visual Reality

โดย

นายเอกชาติ งามชูเกียรติ รหัส 48010752

นายสกลกร กัญฐศแก้ว รหัส 48010923

นายสุชาวัชร อุดมมังกร รหัส 48010978

อาจารย์ที่ปรึกษา

ดร.สุรเดช ตรีไตรลักษณะ

ปริญญาานิพนธ์สำหรับปริญญาวิทยาศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อ โครงการภาษาไทย

การตอบสนองต่อวัตถุเสมือนจริงจากคอมพิวเตอร์

ชื่อ โครงการภาษาอังกฤษ

Interactive Visual Reality

จัดทำโดย

นายเลอชาติ งามชูเกียรติ รหัส 48010752

นายสกลกร กัญฐ์แก้ว รหัส 48010923

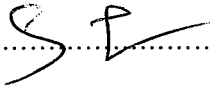
นายสุชาวัชร อุดมมังกร รหัส 48010978

อาจารย์ที่ปรึกษา

ดร.สุรเดช ตรีไตรลักษณ์



รายงานฉบับนี้ได้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ชื่อ..........อาจารย์ที่ปรึกษา

(.....)

วันที่ / /

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตอบสนองต่อวัตถุเสมือนจริงจากคอมพิวเตอร์

นายเลอชาติ งามชูเกียรติ รหัส 48010752

นายสกลกร กัญฐ์ศแก้ว รหัส 48010923

นายสุชาวัชร อุดมมังกร รหัส 48010978

ดร. สุรเดช ตรีไตรลักษณ์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2551

บทคัดย่อ

ปริญญานิพนธ์ที่ทำนี้เป็นโครงการเกี่ยวกับการสร้างวัตถุเสมือนจริงในคอมพิวเตอร์และทำการรับภาพการเคลื่อนไหวของวัตถุจากกล้องเว็บแคมและให้มีการแสดงผลออกมาทางหน้าจอคอมพิวเตอร์เมื่อภาพทั้ง 2 สัมผัสกัน จะเกิดเป็นผลลัพธ์ออกมาในรูปแบบต่างๆ อาทิ แสง สี เสียง และการเคลื่อนไหว โดยในการเขียนโปรแกรมจะใช้ C++ ในการเขียนและชุดคำสั่ง OpenCV

Interactive Visual Reality

Mr.Lerchard Ngamchukiet ID.48010752

Mr.Sakonkorn Kuntudgeow ID.48010923

Mr.Suchawat Udommangkorn ID.48010978

Dr.Suradej Tretriluxana Adviser

Educational 2008

Abstract

This thesis is about making visual objects in the computer and receiving the motion of objects from webcam . It will show the result appears on the computer and when the pictures on the computer touch each other . It will happen in many way such as light sound color and motion . It uses C++ for programming with OpenCV library.

กิตติกรรมประกาศ

โครงการนี้สามารถลุล่วงไปได้ด้วยดีเพราะได้รับความช่วยเหลือจากหลายบุคคลโดยเฉพาะอย่างยิ่ง ดร.สุรเดช ตรีไตรลักษณ์ (อาจารย์ที่ปรึกษา) ที่คอยให้คำปรึกษาและแนะนำเกี่ยวกับโครงการ Interactive Visual Reality อีกทั้งเพื่อนๆ พี่ๆ ทุกคนที่คอยช่วยเหลือในการปฏิบัติงานเป็นอย่างดีมาโดยตลอด จนทำให้โครงการนี้สำเร็จได้ถึงจุดนี้

จึงขอขอบคุณมา ณ ที่นี้

นายเลอชาติ งามชูเกียรติ รหัส 48010752

นายสกลกร กัญจัสแก้ว รหัส 48010923

นายสุชาวัชร อุดมมังกร รหัส 48010978

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีพื้นฐานที่ใช้ในโครงงาน	4
2.1 การประมวลผลภาพ (Image Processing)	4
2.2 ความรู้พื้นฐานสำหรับการประมวลผลภาพดิจิทัล	4
2.2.1 ความหมายของการประมวลผลภาพดิจิทัล	5
2.2.2 ระดับการประมวลผลภาพ	6
2.2.3 การแทนภาพด้วยข้อมูลแบบดิจิทัล	7
2.2.4 ภาพที่นำมาใช้ในการประมวลผล	9
2.3 มาตรฐานของสี	9
2.3.1 ระบบสี RGB	10
2.3.2 ระบบสี HSV	10
2.3.3 ระบบสี HLS	12
2.3.4 การแปลงระบบสี	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 Pixel (Picture element หรือ Pel)	13
2.5 ระดับเทา (Gray Level)	14
2.6 โอเพ่นซีวี (OpenCV)	15
2.6.1 ซีวีแคม (CVCam)	15
2.6.2 ไฮจียูไอ (HighGUI)	16
2.6.3 ซีเอ็กซ์คอร์ (CXCORE)	16
บทที่ 3 เว็บแคมและระบบการเชื่อมต่อกับคอมพิวเตอร์	17
3.1 ความรู้พื้นฐานเกี่ยวกับกล้องดิจิทัล	17
3.1.1 ประเภทของกล้องดิจิทัล	17
3.1.2 หลักการทำงานของกล้องดิจิทัล	18
3.1.3 CCD (Charge Coupled Device)	18
3.1.4 การจัดเก็บภาพของกล้องดิจิทัล	18
3.2 ระบบเชื่อมต่อเว็บแคมกับคอมพิวเตอร์	19
บทที่ 4 การออกแบบและพัฒนาโครงการ	24
4.1 บทนำ	24
4.2 โครงสร้างและการทำงานของระบบ	24
บทที่ 5 การทดลองและผลการทดลอง	28
บทที่ 6 วิเคราะห์และสรุปผลการทดลอง	34
ภาคผนวก	35
บรรณานุกรม	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่ 2.2.1 ภาพแบบต่อเนื่อง (Continuous)	5
รูปที่ 2.2.2 ภาพเชิงตัวเลขจากวิธีการ Digitization	5
รูปที่ 2.2.3.1 ลักษณะการประมวลผลภาพในกราฟิกของคอมพิวเตอร์	7
รูปที่ 2.2.3.2 ตัวอย่างในการแสดง ค่า RGB ใน 1 pixel	8
รูปที่ 2.3.1.1 แสดงความสัมพันธ์ ของ ระบบสี RGB	10
รูปที่ 2.3.2 แสดงความสัมพันธ์ ของ ระบบสี HSV	11
รูปที่ 2.3.3 แสดงความสัมพันธ์ ของ ระบบสี HLS	12
รูปที่ 2.5.1 รูปภาพที่จะทำ <i>Gray Level</i>	14
รูปที่ 2.5.2 ค่าของรูป 2.5.1 ในแต่ละ pixel	14
รูปที่ 3.2.1 Hub USB	19
รูปที่ 3.2.2 หัวต่อยูเอสบี	20
รูปที่ 3.2.3 ช่องสัญญาณของหัวยูเอสบี	21
รูปที่ 3.2.4 การใช้งาน USB ในลักษณะตัวควบคุมเสถียรภาพของแหล่งจ่ายไฟ	21
รูปที่ 4.2.1 แสดงลำดับ ส่วนประกอบของ โครงงานนี้	24
รูปที่ 4.2.2 Flow Chart	25
รูปที่ 4.2.3 แสดง ระยะเวลาของการทำงาน ที่วางแผนไว้ และที่ปฏิบัติได้จริง	26

รูปที่ 4.2.4 รูปจำลองการใช้งาน	27
รูปที่ 5.1 หน้าจอแสดงผลการรัน โปรแกรมที่เขียนขึ้น	28
รูปที่ 5.2 แสดงภาพที่ละเฟรมที่กล้องจับมาได้	29
รูปที่ 5.3 แสดงภาพที่ละเฟรมพร้อมทำ edge detection	30
รูปที่ 5.4 แสดงภาพจากกล้อง Grayscale Threshold และ Edge	31
รูปที่ 5.5 ทำการเปรียบเทียบการกรอง ค่า H อย่างเดียว กับ การกรองค่า H และ ค่า S	32
รูปที่ 5.6 ทำการแทนที่ภาพในขณะที่ภาพเคลื่อนไหว	33



บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของโครงการ

เว็บแคมหรือกล้องผ่านเครือข่ายในปัจจุบันนี้นั้นถือได้ว่าเป็นอุปกรณ์ที่โชกกันแพร่หลายในการรับส่งภาพของผู้ใช้งานแบบตามจริง (Real-Time) โครงการนี้จึงเป็นการนำเอาเว็บแคมมาใช้ร่วมกับคอมพิวเตอร์ในทางการแพทย์ โดยใช้ในเรื่องสมรรถภาพร่างกายของผู้ป่วย การเคลื่อนไหวต่างๆ การทำกายภาพบำบัด

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อช่วยในการทำกายภาพบำบัดของผู้ป่วย
- 1.2.2 เพื่อสร้างโปรแกรมในคอมพิวเตอร์เพื่อให้สามารถรับรูปจากเว็บแคมและแสดงผลออกมาได้
- 1.2.3 เพื่อศึกษาโปรแกรมเกี่ยวกับการประมวลผลภาพ (Image Processing)
- 1.2.4 เพื่อศึกษาการเขียนโปรแกรม Visual Studio C++
- 1.2.5 เพื่อศึกษาระบบการติดต่อกับเว็บแคม

1.3 ขอบเขตของโครงการ

- 1.3.1 โปรแกรมในคอมพิวเตอร์สามารถรับรูปจากเว็บแคมให้แสดงผลออกมาได้
- 1.3.2 โปรแกรมสามารถนำภาพมาประมวลผลได้
- 1.3.3 ภาพที่แสดงผลสามารถแสดงผลได้อย่างเป็นธรรมชาติ
- 1.3.4 โปรแกรมมีการตอบสนองและการทำงานอย่างมีประสิทธิภาพ
- 1.3.5 โปรแกรมมีฟังก์ชันที่ใช้งานได้อย่างสะดวกและเข้าใจวิธีการได้ไม่ยาก

1.4 วิธีการดำเนินงาน

- 1.4.1 ศึกษาเกี่ยวกับการเขียนโปรแกรม Visual Studio C++
- 1.4.2 ศึกษาเกี่ยวกับ OpenCV Library
- 1.4.3 ศึกษาเกี่ยวกับโครงสร้างของรูปภาพและการประมวลผลรูปภาพ
- 1.4.4 ศึกษาวิธีการเชื่อมต่อระหว่างเว็บแคมกับคอมพิวเตอร์
- 1.4.5 วิเคราะห์และออกแบบ

1.5 อุปกรณ์ที่ต้องใช้

- 1.5.1 คอมพิวเตอร์
- 1.5.2 เว็บแคม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ประโยชน์ที่คาดว่าจะได้รับ

- 1.6.1 ได้รับความรู้เกี่ยวกับการประมวลผลของภาพ
- 1.6.2 ได้รับความรู้เกี่ยวกับระบบการติดต่อกับเว็บแคม
- 1.6.3 ได้รับความรู้เกี่ยวกับการเขียนโปรแกรม Visual Studio C++
- 1.6.4 ได้รับความรู้เกี่ยวกับ OpenCV Library

1.7 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 5 บทด้วยกัน ซึ่งประกอบด้วย

บทที่ 1 บทนำ เป็นการกล่าวถึงความสำคัญและที่มาของโครงการ วัตถุประสงค์ของโครงการ ขอบเขตของโครงการ วิธีการดำเนินงาน อุปกรณ์ที่ต้องใช้ ประโยชน์ที่คาดว่าจะได้รับ

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง การประมวลผลภาพ (Image Processing) มาตรฐานของสี

บทที่ 3 เว็บแคมและระบบเชื่อมต่อกับคอมพิวเตอร์

บทที่ 4 การออกแบบและพัฒนาโครงการ เป็นการกล่าวถึงขั้นตอนที่ใช้ในการวิเคราะห์และการออกแบบเพื่อทำการสร้างระบบที่เกี่ยวข้องกับโครงการนี้

บทที่ 5 การทดลองและผลการทดลอง เป็นการกล่าวถึงขั้นตอนและกระบวนการทดลองที่ได้ทำการทดลองทั้งหมด โดยวิธีการนำหลักของการประมวลผลภาพที่ได้กล่าวไว้ในบทข้างต้นมาประยุกต์ใช้ในโปรแกรมรวมถึงผลการทดลองที่ได้ทั้งหมด

บทที่ 6 บทสรุป เป็นการกล่าวถึงผลสรุปผลการทดลอง โดยรวมที่ได้ทำการทดสอบขั้นตอนต่างๆ รวมทั้งผลที่ได้รับ ปัญหา และแนวทางการแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีพื้นฐานที่ใช้ในโครงการ

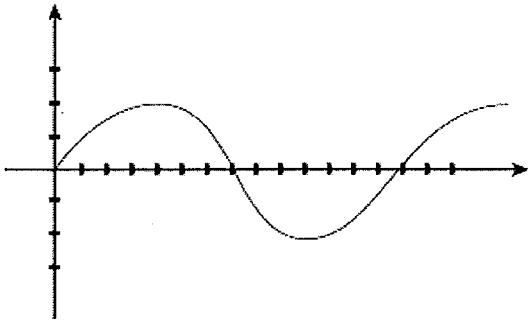
2.1 การประมวลผลภาพ (Image Processing)

การประมวลผลเชิงตัวเลข หมายถึง การนำภาพทั่วไปมาประมวลผลด้วยเครื่องคอมพิวเตอร์ โดยภาพที่นำมาประมวลผลด้วยเครื่องคอมพิวเตอร์นี้จะถูกแทนที่ด้วยตัวเลขให้อยู่ในรูปของเมตริกซ์ แต่ภาพที่ได้โค่นส่วนมากแล้วจะเป็นภาพที่ได้จากตัวรับสัญญาณซึ่งอยู่ในรูปของฟังก์ชัน $f(x,y)$ ที่ต่อเนื่องในระนาบสองมิติ (คือแกน X กับ Y) โดยจะเป็นสัดส่วนกับความสว่างหรือความเข้มของภาพที่ตำแหน่งซึ่งเรียกว่า ระดับสี (Gray Level)

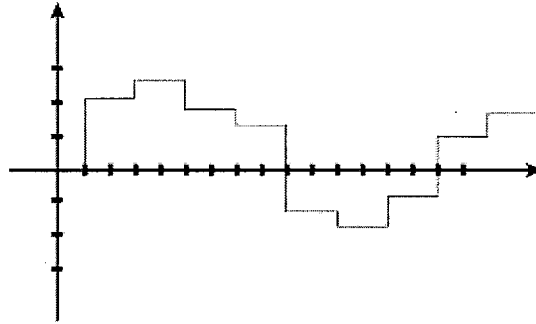
2.2 ความรู้พื้นฐานสำหรับการประมวลผลภาพดิจิทัล

Digital Image คือภาพที่เก็บอยู่ในรูปแบบของดิจิทัล ภาพที่เรามองเห็นด้วยสายตาทั่วไปนั้น เป็นภาพในลักษณะสามมิติ คือ มีมิติของความกว้าง ความยาว และความลึกหรือความสูง ส่วนภาพถ่ายที่เห็นกันอยู่ในโทรทัศน์หรือเครื่องคอมพิวเตอร์นั้น เป็นการแปลงภาพจากสามมิติ มาเป็นสองมิติ โดยการแปลงสัญญาณไฟฟ้าในรูปแบบอนาล็อก ยกตัวอย่างเช่น ในกล้องวิดีโอ เซนเซอร์ที่อยู่ในกล้องทำการสแกนหรือวัดผลรวมความเข้มแสงที่จุดต่าง ๆ ไปตามแนวสแกนที่เรียกว่า Raster Scan การสแกนแบบนี้จะมีทิศทางจากบนลงล่าง และจากซ้ายไปขวา ภาพที่ได้จากการสแกนนั้นจะเป็นภาพแบบต่อเนื่อง (Continuous) ด้วยความเร็วทั่วไปที่ 24 ภาพต่อวินาที เช่นเดียวกันนี้ในเครื่องรับภาพวิดีโอก็จะรับภาพที่ได้มาจากเครื่องถ่ายวิดีโอ และแสดงผลโดยเริ่มจากบนลงล่างและจากซ้ายไปขวาเช่นเดียวกัน

แต่ภาพที่ได้มาจากระบบอนาล็อกนั้นยังเป็นภาพแบบต่อเนื่อง ที่ยังไม่สามารถนำมาใช้ในการประมวลผลได้ ต้องมาทำการแปลงให้เป็นภาพเชิงตัวเลขเสียก่อนด้วยวิธีการ Digitization ซึ่งเป็นการแปลงฟังก์ชันต่อเนื่อง $f(x,y)$ ให้เป็นฟังก์ชันไม่ต่อเนื่อง $g(x,y)$ เพื่อนำมาประมวลผลด้วยคอมพิวเตอร์ได้



รูปที่ 2.2.1 ภาพแบบต่อเนื่อง (Continuous)



รูปที่ 2.2.2 ภาพเชิงตัวเลขจากวิธีการ

Digitization

2.2.1 ความหมายของการประมวลผลภาพดิจิทัล

การประมวลผลภาพ หมายถึง การเรียกใช้ขั้นตอนหรือกรรมวิธีใด ๆ มากระทำกับภาพ โดยมีวัตถุประสงค์เพื่อปรับปรุงคุณภาพของภาพ ให้ได้ภาพใหม่ที่มีคุณสมบัติตามต้องการ เช่น ความคมชัด หรือการประหยัพื้นที่ในการเก็บข้อมูล หรือใช้สำหรับการประมวลผลในระดับสูง เช่นการจดจำรูปร่างลักษณะได้อย่างแม่นยำ โดยทั่วไปแล้ววัตถุประสงค์ของการประมวลผลภาพก็คือ

- Image Processing : Image in \rightarrow Image out
วิธีนี้จะใช้กระบวนการประมวลผลภาพ เพื่อให้ได้ภาพออกมา เช่น การตกแต่งภาพด้วยโปรแกรม Photoshop เป็นต้น
- Image Analysis : Image in \rightarrow Measurements out
วิธีนี้จะใช้กระบวนการประมวลผลภาพ เพื่อให้ได้ค่าการวัดออกมา เช่น การวัดขนาดในงานอุตสาหกรรม เป็นต้น
- Image Understanding : Image in \rightarrow High-Level Description out
วิธีนี้จะใช้กระบวนการประมวลผลภาพ เพื่อให้ได้ผลลัพธ์ออกมาเป็นความหมาย ตัวอย่างของ High-Level Description เช่น การจดจำตัวอักษร (Optical Character Recognition : OCR) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประมวลผลภาพด้วยคอมพิวเตอร์ สามารถทำได้โดย นำภาพที่ได้มาจากกล้องหรือ Image Source ต่าง ๆ ซึ่งเป็นสัญญาณอนาล็อก แล้วนำมาแปลงเป็นสัญญาณดิจิทัลที่มีลักษณะเป็นรหัสเชิงตัวเลข 0,1 ที่สามารถใช้รูปแบบทางคณิตศาสตร์เข้ามาช่วยในการคำนวณและการประมวลผลข้อมูลภาพด้วยคอมพิวเตอร์ได้ต่อไป

2.2.2 ระดับการประมวลผลภาพ แบ่งได้เป็น 2 ระดับ คือ

1. การประมวลผลภาพระดับต่ำ (Low Level Image Processing)
2. การประมวลผลภาพระดับสูง (High Level Image Processing)

2.2.2.1 การประมวลผลภาพระดับต่ำ (Low Level Image Processing)

เป็นการประมวลผลขั้นแรกสุดก่อนที่จะนำไปสู่การประมวลผลภาพระดับสูงต่อไป นั่นคือหลังจากที่เราได้ภาพมา ภาพที่ได้ก็จะประกอบด้วยองค์ประกอบต่าง ๆ มากมาย รวมถึงสิ่งที่ไม่ต้องการด้วย ในที่นี้เราจะเรียกว่าสัญญาณรบกวน (Noise) ซึ่งทำให้ภาพที่ได้มีคุณภาพไม่ดี ยังไม่สามารถที่จะนำไปใช้ประมวลผลได้ ดังนั้น การประมวลผลภาพในระดับต่ำจึงประกอบไปด้วยการกำจัดสัญญาณรบกวน การทำภาพให้ชัด (High Pass Filter) การหาขอบภาพ (Edge Detection) การแปลง Binary Image การแบ่งแยกรูปร่างวัตถุ (Image Segmentation) เป็นต้น เพื่อหาค่าตัวแปรต่าง ๆ มาอธิบายข้อมูลภาพ และมีวัตถุประสงค์ที่จะนำตัวแปรเหล่านี้มาใช้ในการประมวลผลภาพในระดับสูงต่อไป

2.2.2.2 การประมวลผลภาพระดับสูง (High Level Image Processing)

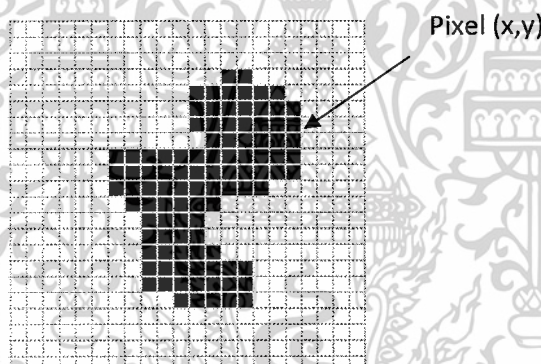
เป็นการทำให้คอมพิวเตอร์รู้จักและเข้าใจภาพนั้นได้ เช่น การจดจำใบหน้าคน หรืออาจจะเป็นการจดจำตัวอักษร เป็นต้น ความแตกต่างของการประมวลผลภาพระดับต่ำและระดับสูง คือ ข้อมูลที่นำมาใช้ในการประมวลผลระดับต่ำจะใช้ค่าความสว่าง หรือ ความเข้มของแสงโดยตรง ส่วนการประมวลผลภาพระดับสูง ข้อมูลที่นำมาใช้ในการประมวลผลจะถูกแสดงในรูปแบบของสัญลักษณ์ โดยสัญลักษณ์เหล่านี้จะแสดงถึงสิ่งต่าง ๆ ที่อยู่ในภาพ และการใช้ตัวแปรที่ได้จากการประมวลผลภาพระดับต่ำมาอธิบายถึงสัญลักษณ์เหล่านี้ การประมวลผลภาพระดับสูงนั้นส่วนใหญ่มักใช้ทฤษฎีต่าง ๆ เข้ามาใช้เป็นตัวช่วยในการทำงาน หรือเป็นหัวใจของโปรแกรม เช่น Fuzzy Logic , Neural Network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างที่กล่าวไปแล้วว่าการประมวลผลภาพระดับสูงจำเป็นต้องใช้ข้อมูลที่ได้มาจากการประมวลผลภาพระดับต่ำ ดังนั้นจะเห็นได้ว่าการประมวลผลภาพระดับต่ำมีความสำคัญมากสำหรับการทำคอมพิวเตอร์วิทัศน์และเข้าใจภาพได้

2.2.3 การแทนภาพด้วยข้อมูลแบบคิจิตอล

จากที่ได้กล่าวไปแล้วในตอนต้น ข้อมูลภาพแบบคิจิตอลเป็นภาพที่ถูกตัดแปลงมาจากภาพแบบต่อเนื่อง ให้อยู่ในรูปตัวเลข ด้วยวิธีการ Digitization โดยภาพ Analog Image จะถูกแบ่งให้เป็นพื้นที่สี่เหลี่ยมเล็ก ๆ ที่เรียกว่า พิกเซล (Pixel) โดยแต่ละพิกเซลจะใช้ (x, y) ในการระบุตำแหน่ง การแสดงข้อมูลภาพคิจิตอลสามารถอธิบายได้ด้วย เมทริกซ์ $(M \times N)$ และให้จุดต่าง ๆ ที่อยู่ในเมทริกซ์ เป็นจุดที่พิกัด (x, y) ใด ๆ เป็นส่วนประกอบของภาพ เมื่อเราเปรียบเทียบระหว่าง ภาพ และ Pixel Matrix ดังรูป จะเห็นว่าจุดกำเนิดของภาพจะอยู่ที่มุมล่างซ้าย แต่จุดกำเนิดของพิกเซลจะอยู่ที่มุมบนซ้าย ซึ่งจะเป็นลักษณะการประมวลผลภาพในกราฟิกของคอมพิวเตอร์



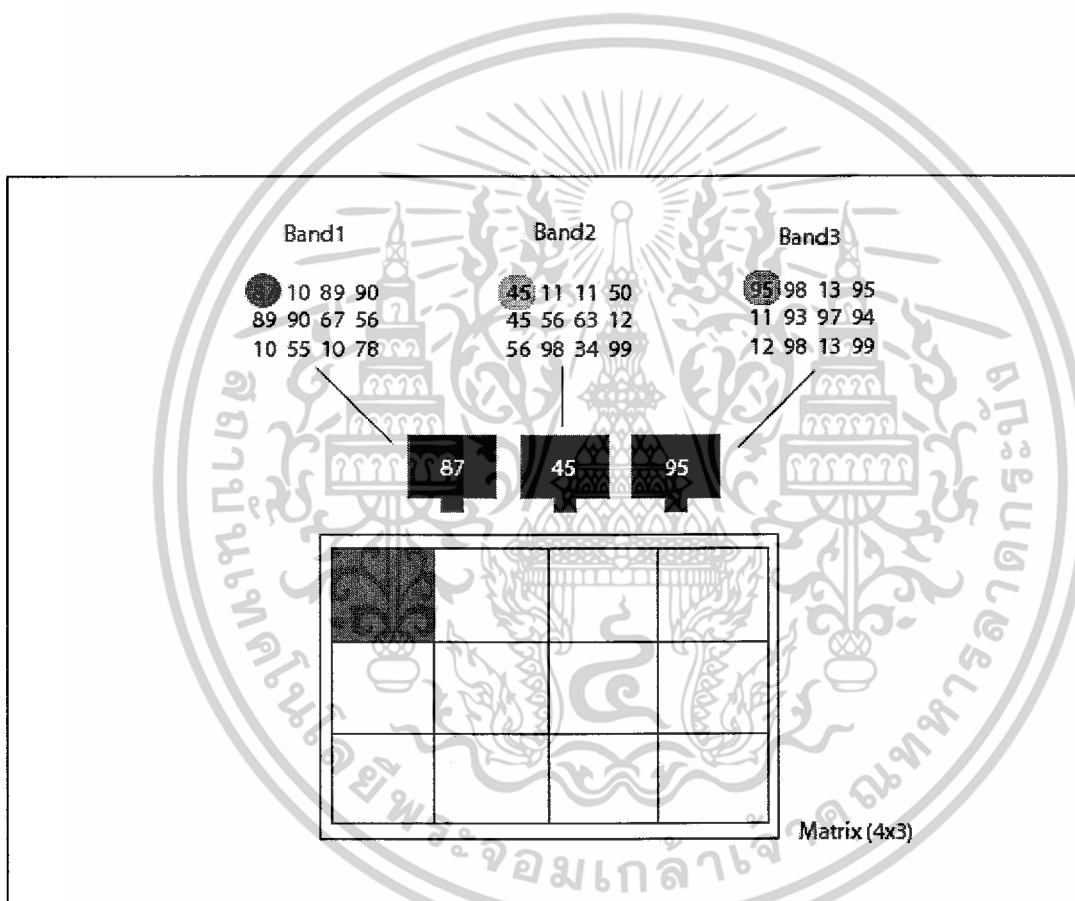
รูปที่ 2.2.3.1 ลักษณะการประมวลผลภาพในกราฟิกของคอมพิวเตอร์

ค่าของพิกเซล หรือ พังก์ชัน (x, y) ณ จุดใด ๆ จะแสดงได้ด้วยค่าของความเข้มแสงซึ่งอาจแบ่งได้หลายระดับ ถ้ามี 2 ระดับ ก็จะเป็นแค่ 0 กับ 1 จากรูป จุดต่าง ๆ ที่แสดงอยู่ในพิกัดนี้ก็คือ พิกเซล หรือ Picture Element ซึ่งก็คือ ความสว่างหรือค่า Luminance (L) ของภาพ ถ้าภาพนั้นเป็นภาพขาวดำ ขนาด 8 บิต จะมีค่า L เท่ากับ 28 หรือเท่ากับ 256 คือตั้งแต่ระดับ 0 จนถึง 255 บางครั้งค่าความสว่าง (L) อาจหมายถึงระดับความละเอียดของภาพ (Image Resolution) ถ้าพิกเซล เป็นภาพขาวดำจะอ่านค่าภาพคิจิตอลในรูปแบบเมทริกซ์ 2 มิติ ขนาด $(M \times N)$ ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,n-1) \\ f(1,0) & f(1,1) & \dots & f(1,n-1) \\ \dots & \dots & \dots & \dots \\ f(m-1,0) & f(m-1,1) & \dots & f(m-1,n-1) \end{bmatrix}_{(M \times N)}$$

โดยที่ค่า $f(x,y)$ จะอยู่ในช่วง 0 ถึง 255 สมมติว่าอ่านค่าพิกเซล จากภาพหนึ่งได้ $f(x,y)$ เท่ากับ 10 แสดงว่าจุดพิกเซลนั้นมีความสว่างน้อยหรือค่อนข้างดำ ถ้าอ่านได้เป็น 255 แสดงว่าจุดพิกเซลนั้นมีความสว่างมาก หรือเป็นสีขาว



รูปที่ 2.2.3.2 ตัวอย่างในการแสดง ค่า RGB ใน 1 pixel

จากรูปด้านบนจะช่วยให้เข้าใจการแสดงค่า พิกเซลในเมทริกซ์มากขึ้น เริ่มต้นด้วยพิกัด

$f(x,y) = f(0,0)$ ค่าของพิกเซลที่ได้จะเป็นการผสมสีกันระหว่างค่าของแม่สีทั้งสาม ซึ่งได้แก่ แดง เขียว น้ำเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 ภาพที่นำมาใช้ในการประมวลผล

โดยทั่วไปแล้วภาพที่นำมาใช้ในการประมวลผลภาพดิจิทัลนั้นแบ่งออกเป็นสองประเภทหลัก ๆ คือ

1. ภาพเคลื่อนไหว
2. ภาพนิ่ง

2.2.4.1 ภาพเคลื่อนไหว

ความจริงแล้วภาพเคลื่อนไหวก็คือภาพนิ่งที่นำมาแสดงต่อ ๆ กันแบบต่อเนื่อง จะต้องใช้รูปภาพอย่างน้อย 24 รูปต่อหนึ่งวินาที เนื่องจากสายตาของคนเราเมื่อนำภาพนิ่งมาฉายติดต่อกันมากกว่า 24 รูปต่อหนึ่งวินาทีแล้วก็จะมองว่าภาพนั้นเป็นภาพเคลื่อนไหว เพราะสายตารเราแยกไม่ออกเนื่องจากมีความเร็วมากเกินไป แต่หากนำภาพนิ่งมาฉายติดต่อกันน้อยกว่า 24 รูปต่อหนึ่งวินาทีแล้ว เราจะมองเห็นว่าภาพนั้นไม่ต่อเนื่อง

2.2.4.2 ภาพนิ่ง

ภาพที่นำเข้ามาในประมวลผลในคอมพิวเตอร์นั้น ถ้าในระบบ RGB ก็จะใช้ความเข้มแสงสีแดง เขียว และน้ำเงิน ความหมายของภาพนิ่งก็คือมีอยู่เพียงภาพเดียว (ถ้าเป็นภาพเคลื่อนไหวก็จะประกอบไปด้วยภาพนิ่งหลาย ๆ ภาพ) ภาพนิ่งที่นำมาใช้ก็มีอยู่หลาย Format ไม่ว่าจะเป็น .bmp หรือ .jpg เป็นต้น ส่วนการเลือกใช้นั้นก็แล้วแต่ความเหมาะสม ส่วนใหญ่จะเป็น .bmp เพราะไม่ต้องมาลดอัตราที่สก่อน เนื่องจากภาพที่เป็น .jpg นั้นมีการบีบอัดภาพให้มีขนาดเล็ก ดังนั้นหากจะนำมาใช้ก็ต้องคลายข้อมูลออกมาก่อนที่จะนำภาพไปประมวลผลต่อไป

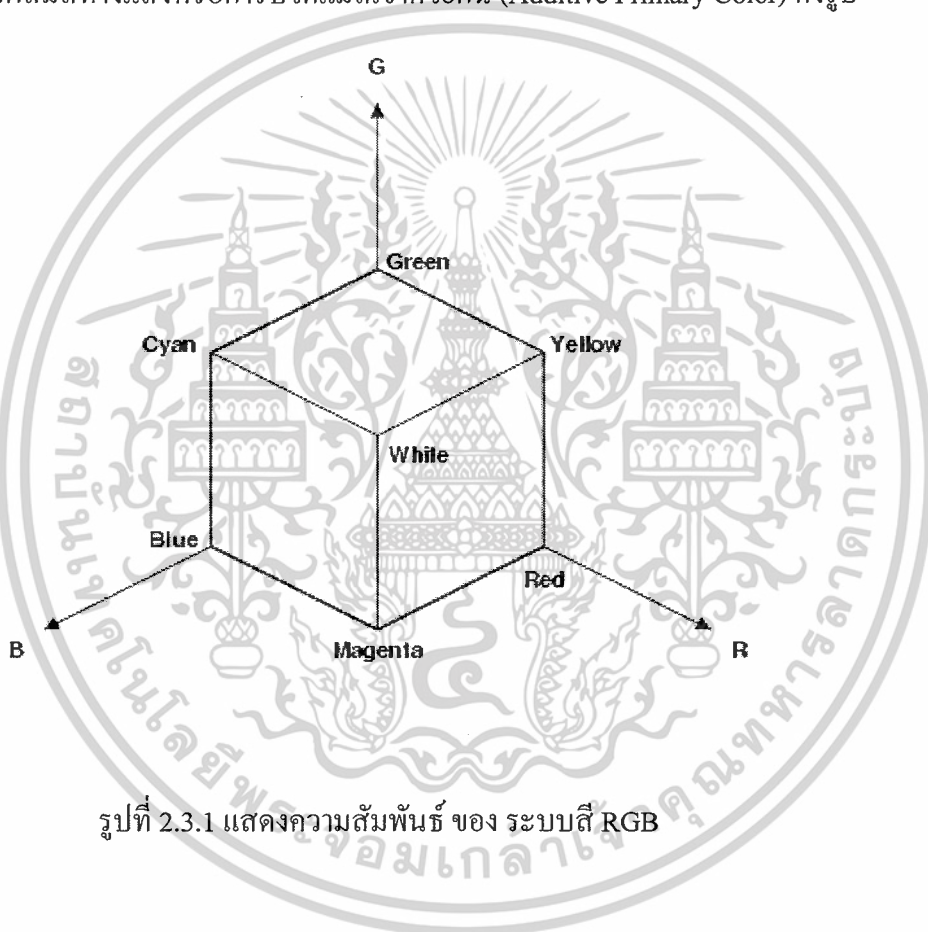
2.3 มาตรฐานของสี

มาตรฐานของสีที่ใช้อยู่ในปัจจุบันมีหลายระบบด้วยกัน แต่โดยทั่วไปแล้วทุกมาตรฐานจะมีแนวคิดเดียวกันก็คือ การแทนจุดสีด้วยจุดที่อยู่ภายในสเปซ 3 มิติ โดยจะมีแกนอ้างอิงสำหรับจุดสีนั้นในสเปซซึ่งแต่ละแกนจะมีอิสระต่อกัน ตัวอย่างเช่นในระบบ RGB จะมีแกนสีคือ แกนสีแดง สีเขียว และสีน้ำเงิน

2.3.1 ระบบสี RGB

ระบบสี RGB เป็นระบบสีที่เกิดจากการรวมกันของแสงสีแดง เขียวและน้ำเงิน โดยปกติจะใช้ในจอภาพแบบ CRT และเนื่องจากระบบสี RGB เป็นระบบสีของแสง จึงทำให้ภาพที่ได้ออกมานั้นมีความสมจริงและยังดูสวยงาม

โมเดลสี หรือ Color Space ประกอบด้วย 3 แม่สีหลัก ได้แก่ สีแดง สีเขียว และสีน้ำเงิน ถ้านำแต่ละแม่สีมาพล็อตกราฟในระดับพิกัด Color Space โดยแต่ละสีมีค่า 0 ถึง 1 (0 แสดงถึงค่าความมืด และ 1 แสดงถึงความสว่าง) จะได้ภาพการผสมสีทางแสงหรือการบวกแม่สีเข้าด้วยกัน (Additive Primary Color) ดังรูป

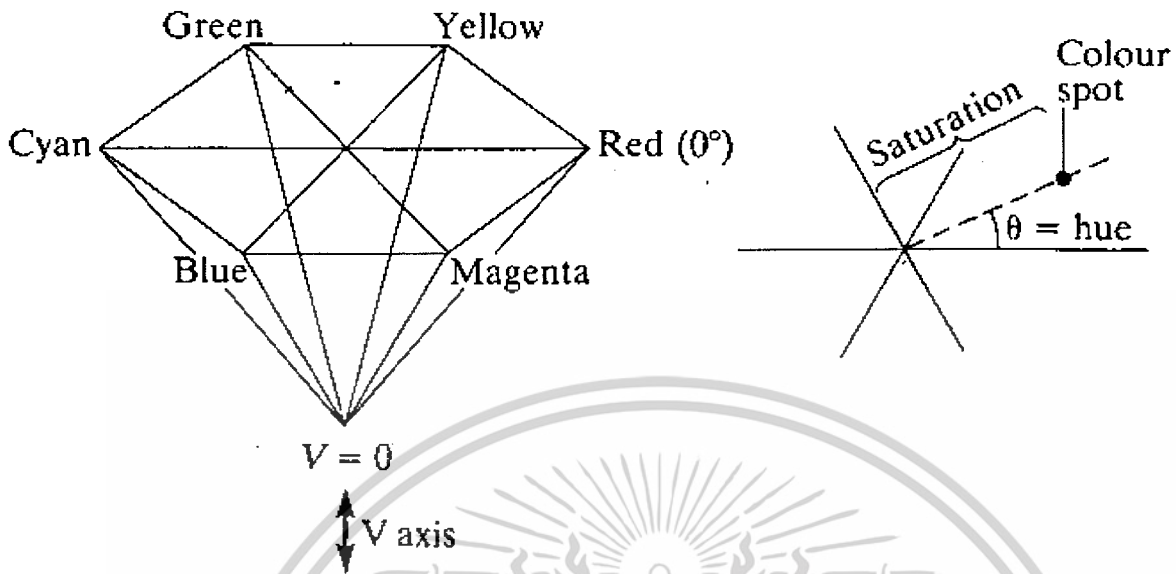


รูปที่ 2.3.1 แสดงความสัมพันธ์ของระบบสี RGB

2.3.2 ระบบสี HSV

ระบบสี HSV (Hue Saturation Value) เป็นการพิจารณาสีโดยใช้ Hue Saturation และ Value ซึ่ง Hue คือค่าสีของสีหลัก (แดง เขียวและน้ำเงิน) ในทางปฏิบัติจะอยู่ระหว่าง 0 และ 255 ซึ่งถ้า Hue มีค่าเท่ากับ 0 จะแทนสีแดงและเมื่อ Hue มีค่าเพิ่มขึ้นเรื่อยๆ สีก็จะเปลี่ยนแปลงไปตามสเปกตรัมของสีจนถึง 256 จึงจะกลับมาเป็นสีแดงอีกครั้ง ซึ่งสามารถแทนให้อยู่ในรูปขององศาได้ ดังนี้คือ สีแดง = 0 องศา สีเขียวเท่ากับ 120 องศา สีน้ำเงินเท่ากับ 240 องศา

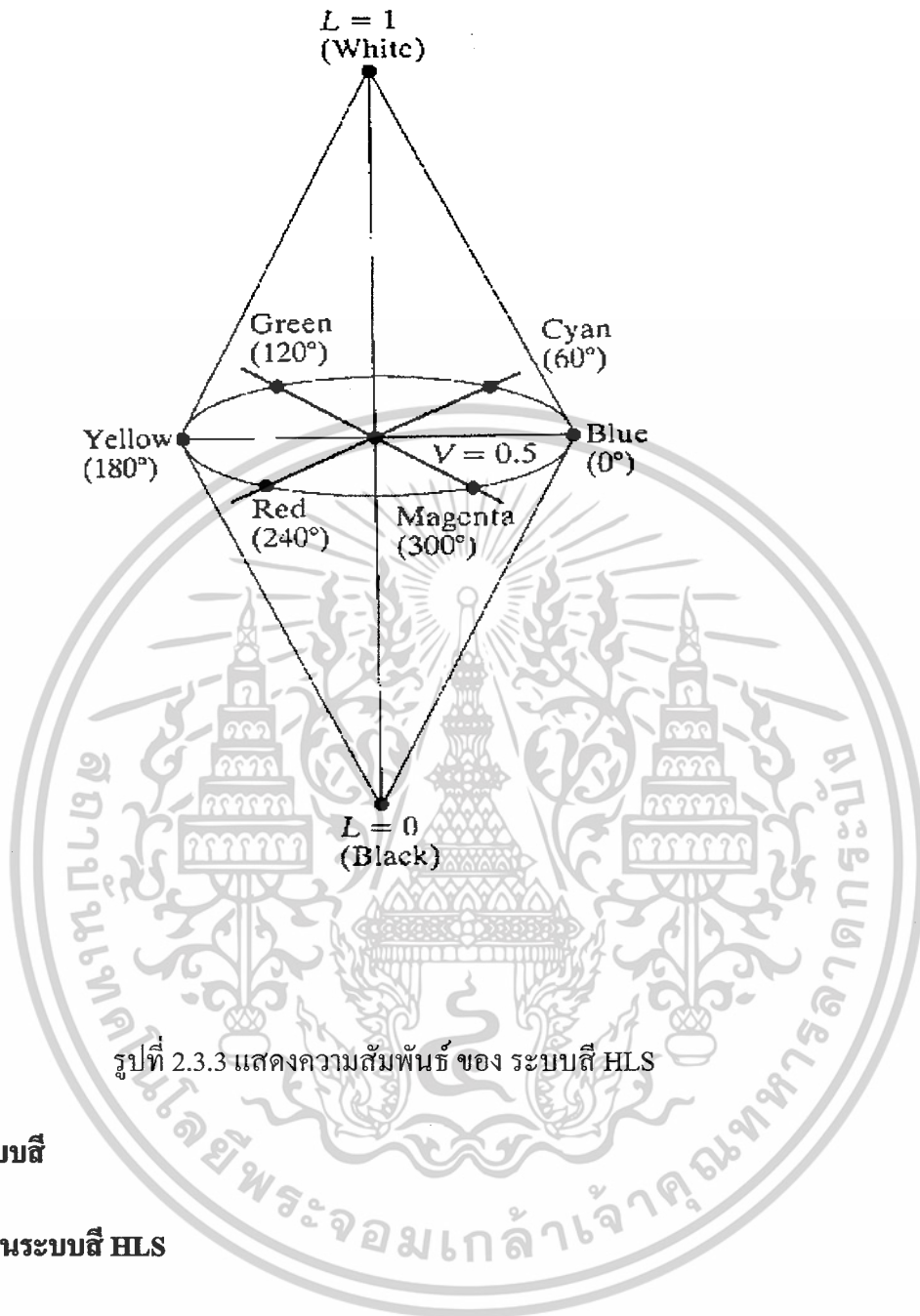
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3.2 แสดงความสัมพันธ์ของระบบสี HSV

2.3.3 ระบบสี HLS

ระบบสีแบบ HLS (Hue lightness saturation) พัฒนาโดย Teletromix Incorporated จะมีลักษณะคล้ายกับ HSV ดังนั้นคือสีของระบบจะขึ้นอยู่กับ Hue Lightness และ Saturation



รูปที่ 2.3.3 แสดงความสัมพันธ์ของระบบสี HLS

2.3.4 การแปลงระบบสี

2.3.4.1 RGB ไปเป็นระบบสี HLS

$$h = \begin{cases} 0 & \text{if max} = \text{min} \\ (60^\circ \times \frac{g-b}{\text{max} - \text{min}} + 360^\circ) \bmod 360^\circ, & \text{if max} = r \\ 60^\circ \times \frac{b-r}{\text{max} - \text{min}} + 120^\circ, & \text{if max} = g \\ 60^\circ \times \frac{r-g}{\text{max} - \text{min}} + 240^\circ, & \text{if max} = b \end{cases}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$l = \frac{1}{2}(\max + \min)$$

$$s = \begin{cases} 0 & \text{if } \max = \min \\ \frac{\max - \min}{\max + \min} = \frac{\max - \min}{2l}, & \text{if } l \leq \frac{1}{2} \\ \frac{\max - \min}{2 - (\max + \min)} = \frac{\max - \min}{2 - 2l}, & \text{if } l > \frac{1}{2} \end{cases}$$

2.3.4.2 RGB ไปเป็นระบบสี HSV

$$h = \begin{cases} 0 & \text{if } \max = \min \\ (60^\circ \times \frac{g-b}{\max - \min} + 360^\circ) \bmod 360^\circ, & \text{if } \max = r \\ 60^\circ \times \frac{b-r}{\max - \min} + 120^\circ, & \text{if } \max = g \\ 60^\circ \times \frac{r-g}{\max - \min} + 240^\circ, & \text{if } \max = b \end{cases}$$

$$s = \begin{cases} 0, & \text{if } \max = 0 \\ \frac{\max - \min}{\max} = 1 - \frac{\min}{\max}, & \text{otherwise} \end{cases}$$

$$v = \max$$

2.4 Pixel (Picture element หรือ Pel) คือ พื้นที่เล็กๆจุดหนึ่งในภาพ โดยในแต่ละจุดนั้นจะมีค่าตัวเลขกำกับ ซึ่งตัวเลขเหล่านี้จะมาจากค่าของแม่สีสามสี R (สีแดง) G (สีเขียว) B (สีฟ้า) ใช้บอกระดับความเข้มของแต่ละเฉดสี หากมี Pixel หลายๆจุดมาต่อกันจะกลายเป็นภาพซึ่งมีขนาด จำนวน Pixel ด้านกว้าง X จำนวน Pixel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้านยาว ยกตัวอย่าง เช่น รูปภาพขนาด 800 x 600 pixels หมายความว่า รูปถ่ายนี้มีควมกว้าง 800 pixels และมีความยาว 600 pixels เป็นต้น

2.5 ระดับเทา (Gray Level) เป็นค่าซึ่งระบุความสว่างหรือความเข้ม ซึ่งมีค่าตั้งแต่ 0-255 (0 คือระดับเข้ม 255 คือระดับสว่าง) รวมทั้งพิกัดแนวนอนและแนวตั้ง ซึ่งใช้ระบุตำแหน่งในแถวลำดับภาพ (Image Array) เช่นจากรูปตัวอย่างที่ 2.5.1 และ 2.5.2 จุดภาพแถวอนที่ 3 และแนวตั้งที่ 2 ในรูป 2.5.2 ซึ่งมีค่าระดับเทา 40

วิธีการหาค่าระดับเทา (Gray Level)

$$\text{Gray Level} = \frac{R + G + B}{3}$$



40	40	40	40	40	40	40	40	40	40	40
40	40	40	40	200	200	40	40	40	40	40
40	40	40	40	200	200	40	40	40	40	40
40	40	200	200	200	200	200	200	40	40	40
40	40	40	40	200	200	40	40	40	40	40
40	40	40	40	200	200	40	40	40	40	40
40	40	40	40	40	40	40	40	40	40	40

รูปที่ 2.5.1 รูปภาพที่จะทำ Gray Level

รูปที่ 2.5.2 ค่าของรูป 2.5.1 ในแต่ละ pixel

วิธีการหาค่าระดับเทาที่กล่าวมาข้างต้นเป็นการเฉลี่ยค่าของแม่สีทั้งสามซึ่งเป็นวิธีการที่ง่ายที่สุดแต่ก็อาจจะมีความเพี้ยนของสีได้จึงมีวิธีอีกอย่างหนึ่งซึ่งจะติดตามความสว่างของแต่ละแม่สีมีรูปแบบดังสมการ

$$R_R = \frac{(R_S + G_S + B_S)}{3} \quad \text{หรือ} \quad R_R = ((0.299 \times R_S) + (0.587 \times G_S) + (0.114 \times B_S))$$

$$G_R = \frac{(R_S + G_S + B_S)}{3} \quad \text{หรือ} \quad G_R = ((0.299 \times R_S) + (0.587 \times G_S) + (0.114 \times B_S))$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$B_R = \frac{(R_S + G_S + B_S)}{3} \quad \text{หรือ} \quad B_R = ((0.299 \times R_S) + (0.587 \times G_S) + (0.114 \times B_S))$$

โดยที่ R_R หมายถึง ค่าเอาต์พุต Pixels สีแดง

G_R หมายถึง ค่าเอาต์พุต Pixels สีเขียว

B_R หมายถึง ค่าเอาต์พุต Pixels สีน้ำเงิน

R_S หมายถึง ค่าอินพุต Pixels สีแดง G_S หมายถึง ค่าอินพุต Pixels สีเขียว

B_S หมายถึง ค่าอินพุต Pixels สีน้ำเงิน

2.6 โอเพ่นซีวี (OpenCV)

โอเพ่นซีวีเป็นไลบรารี (Library) ที่สร้างขึ้นมาโดยอ้างอิงจากสถาปัตยกรรมของบริษัทอินเทล (Intel) เป็นหลัก เพื่อประมวลผลภาพในคอมพิวเตอร์แบบเรียลไทม์ (Real Time) ซึ่งมีคุณสมบัติมากมายเช่น การอินเตอร์แอคทีฟระหว่างมนุษย์และคอมพิวเตอร์ (Human-Computer Interactive: HCI) การระบุชนิดของวัตถุ การแยกส่วนของภาพ และมีส่วนของการจดจำวิเคราะห้ เช่น การรู้จำใบหน้าของมนุษย์ การรู้จำปฏิกิริยาของมนุษย์ การติดตามการเคลื่อนไหว การรู้จำความรู้สึกของมนุษย์ การเข้าใจการเคลื่อนไหว และการประยุกต์ใช้สำหรับหุ่นยนต์

โอเพ่นซีวีเป็นไลบรารีที่มีขนาดของซอร์ซโคด (Source Code) เพียงเล็กน้อยแต่สามารถประมวลผลภาพได้อย่างมีประสิทธิภาพ

2.6.1 ซีวีแคม (CVCam)

ซีวีแคมเป็น โมดูล (Module) ที่สามารถพัฒนาได้หลายแพลตฟอร์ม (Platform) ถูกสร้างขึ้นมาเพื่อประมวลผลสัญญาณวิดีโอจากกล้องวิดีโอดิจิทัล (Digital Camera) ซึ่งได้รับการพัฒนาให้อยู่ในรูปแบบไดนามิกลิงค์ไลบรารี (Dynamic link library : DLL) สำหรับระบบปฏิบัติการวินโดวส์และเป็นแชร์อ็อบเจกต์ไลบรารี (Shared Object Library : SO) สำหรับระบบปฏิบัติการลินุกซ์ (Linux) โดยเรียกใช้งานผ่านเอพีไอ (Application Programming Interface : API) เพื่ออ่านและควบคุมสัญญาณวิดีโอ ประมวลผลภาพในแต่ละเฟรม (frames) และแสดงผล ซึ่งซีวีแคมเป็นส่วนประกอบส่วนหนึ่งของไลบรารีโอเพ่นซีวีและอ้างอิงฟังก์ชันบางฟังก์ชันจากโอเพ่นซีวี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 ไฮจียูไอ (HighGUI)

ไฮจียูไอเป็นส่วนที่ทำหน้าที่จัดการข้อมูลสัญญาณวิดีโออินพุทเข้าพุทเช่นการรับสัญญาณวิดีโอจากกล้องมาเก็บไว้ในหน่วยความจำและดึงสัญญาณวิดีโอจากหน่วยความจำไปแสดงผลออกทางหน้าจอ และทำการสร้างหน้าต่างสำหรับการแสดงผลบนหน้าจอคอมพิวเตอร์

2.6.3 ซีเอ็กซ์คอร์ (CXCore)

ซีเอ็กซ์คอร์เป็นส่วนที่ทำหน้าที่ประมวลผลกับข้อมูลภาพแต่ละภาพเช่นการกระทำกระบวนการทางคณิตศาสตร์กับข้อมูลภาพ การวาดเส้นตรง เส้น โค้ง เขียนตัวหนังสือ และวาดจุดบนภาพ การกระทำกระบวนการทางสถิติ และการเข้าถึงข้อมูลภายในภาพโดยใช้เมตริกซ์ (Matrix)



เว็บแคมและระบบการเชื่อมต่อกับคอมพิวเตอร์

3.1 ความรู้พื้นฐานเกี่ยวกับกล้องดิจิทัล

กล้องดิจิทัลมีลักษณะเหมือนกล้องถ่ายรูปธรรมดาทั่วไปคือใช้ในการถ่ายภาพ แต่ กล้องดิจิทัลมีคุณลักษณะพิเศษคือไม่ต้องใช้ฟิล์ม ดังนั้นจึงไม่ต้องล้างและอัดภาพเพียงแค่นั้นที่เก็บข้อมูลภาพที่ถ่ายในหน่วยความจำ แล้วจึงนำตัวกล้องนำมาพ่วงต่อกับคอมพิวเตอร์ก็สามารถจัดเก็บรูปภาพเหล่านั้นเป็นไฟล์รูปภาพได้ นอกจากนี้ยังสามารถนำรูปภาพที่ได้จากกล้องดิจิทัลไปตกแต่งให้สวยงามได้ ด้วยโปรแกรมตกแต่งภาพต่าง ๆ ไม่ว่าจะเป็น PhotoShop, CorelDraw ตลอดจน โปรแกรมต่างๆ ที่แถมมากับกล้อง

3.1.1 ประเภทของกล้องดิจิทัล หากนำมาเปรียบเทียบตามลักษณะการใช้งาน

จะแบ่งได้ออกมา 3 ประเภท ดังนี้

ก) กล้องถ่ายภาพดิจิทัลระดับต่ำ หรือ กล้องถ่ายภาพดิจิทัลสำหรับบุคคลทั่วไป

เป็นกล้องที่ใช้งานง่าย ไม่จำเป็นต้องปรับแต่งค่าใดๆ กล้องถ่ายภาพดิจิทัลประเภทนี้มักมีขนาดเล็ก ราคาไม่แพง มีระบบการทำงานอัตโนมัติและตายตัว แต่ความละเอียดของภาพที่ได้น้อยมาก สามารถแสดงเป็นภาพถ่ายขนาดเพียงประมาณ 4 x 6 นิ้ว ซึ่งความละเอียดของภาพเหมาะที่จะใช้ใน เว็บเพจ (Webpage) หรือ จดหมายอิเล็กทรอนิกส์เท่านั้น

ข) กล้องถ่ายภาพดิจิทัลระดับปานกลาง หรือ กล้องดิจิทัลสำหรับช่างภาพสมัครเล่น

ลักษณะตัวกล้องยังคงมีลักษณะของกล้องถ่ายภาพประเภทแรก แต่มีระบบการทำงานบางอย่างที่สามารถเลือกปรับได้ และที่สำคัญคือมีความละเอียด สูงขึ้น คืออยู่ในระดับมากกว่า 1,000,000 Pixels ซึ่งสามารถแสดงเป็นภาพถ่ายขนาด 8 x 10 นิ้วได้ พอเพียงต่อการนำภาพถ่ายไปใช้ในงานต่าง ๆ

ก) กล้องถ่ายภาพดิจิทัลระดับสูง หรือกล้องถ่ายภาพสำหรับมืออาชีพ

ลักษณะกล้องถ่ายภาพ SLR มีการทำงานเช่นเดียวกับกล้องถ่ายภาพธรรมดาประเภท SLR ทั่วไปที่สามารถเลือกปรับตั้งการทำงานได้เองทุกอย่าง รวมทั้งสามารถถอดเปลี่ยนเลนส์ได้ ที่เป็นเช่นนี้เพราะกล้องประเภทนี้สร้างขึ้นจากการนำกล้องถ่ายภาพธรรมดาแบบ SLR มาดัดแปลงเพิ่มเติมโดยประกอบฝาหลังดิจิทัลเข้าไป กล้องประเภทนี้จะมีความละเอียดสูงมาก คือ มากกว่า 2,000,000 Pixels ขึ้นไป แต่ราคาก็สูงมากเช่นเดียวกัน

3.1.2 หลักการทำงานของกล้องดิจิทัล เมื่อทำการถ่ายภาพที่ถ่ายจะตกบน CCD (Charge Coupled Device) ซึ่งเป็นอุปกรณ์รับภาพที่ประกอบด้วยเซลล์ไวแสงจำนวนมาก และเซลล์เหล่านี้จะทำหน้าที่แปลงภาพซึ่งเป็นสัญญาณ Analog ไปเป็นสัญญาณ Digital โดยอุปกรณ์ที่เรียกว่า ATD (Analog to Digital converter) จากนั้นจึงเข้าสู่กระบวนการตรวจสอบความคมชัดของภาพและทำการบีบอัดเพื่อลดพื้นที่ในการจัดเก็บภาพ ก่อนจะส่งไปเก็บที่หน่วยความจำของกล้อง ดังนั้นยังกล้องมีเซลล์ใน CCD (Charge Coupled Device) มากเท่าไร ภาพที่ได้ก็จะยิ่งมีความละเอียดมากขึ้นเท่านั้น

3.1.3 CCD (Charge Coupled Device) เป็นอุปกรณ์อิเล็กทรอนิกส์ที่มีความไวต่อแสงหรืออาจเรียกอีกอย่างหนึ่งว่าอิมเมจเซ็นเซอร์ และมีความสำคัญเกี่ยวกับความละเอียดของภาพที่ได้จากการถ่ายรูปด้วยกล้องดิจิทัล เนื่องจากยิ่ง CCD มีพื้นที่ในการรับภาพมากเท่าไรความละเอียดของภาพที่ได้ก็จะสูงมากขึ้นเท่านั้น แต่ในขณะเดียวกันราคาของกล้องก็จะสูงขึ้นเช่นเดียวกัน กล้องดิจิทัลส่วนใหญ่จะมี CCD เพียง 1 ตัวและใช้ฟิลเตอร์เพื่อกรองแสง ซึ่งแตกต่างจากกล้องวิดีโอดิจิทัลที่มี CCD ถึง 3 ตัวเพื่อทำหน้าที่รับแสงให้ครบทั้ง 3 สี คือ สีแดง สีเขียว และสีน้ำเงิน (RGB)

3.1.4 การจัดเก็บภาพของกล้องดิจิทัล ภายในกล้องดิจิทัลจะประกอบด้วยส่วนที่เรียกว่า CCD Array ซึ่งประกอบด้วยตัวเซ็นเซอร์นับล้านตัวในส่วนที่ทำหน้าที่เก็บความสว่างของสี (แดง,เขียว,น้ำเงิน) ข้อมูลเกี่ยวกับความสว่างจะถูกอ่านเข้ามาประมวลผลครั้งละ 1 แถวโดยไอซีที่ทำหน้าที่แปลงสัญญาณแอนาล็อกเป็นดิจิทัล และจะทำการเปลี่ยนสัญญาณข้อมูลความสว่างเป็นระดับแรงดันไฟฟ้าที่ แตกต่างกันไปตามระดับความสว่างของแสง กล้องดิจิทัลแบบเก่าจะทำการเก็บข้อมูลรูปภาพ ลงในหน่วยความจำ หรือ Memory Chip ที่บรรจุอยู่ในตัวกล้อง ซึ่งหากหน่วยความจำเต็มจะทำให้ไม่สามารถที่จะบันทึกภาพได้อีก ดังนั้นจึงต้องลบภาพที่ไม่จำเป็นออกหรือทำการถ่ายโอนภาพเข้าสู่เครื่องคอมพิวเตอร์เพื่อล้างหน่วยความจำ แต่สำหรับกล้องดิจิทัลรุ่นใหม่สามารถเก็บรูปภาพไว้ในหน่วยความจำที่เรียกว่า Flash Memory ซึ่งเป็น RAM หรือหน่วยความจำชนิดหนึ่งซึ่งมีลักษณะคล้ายกับ RAM ที่ใช้ในเครื่องคอมพิวเตอร์ส่วนบุคคลทั่วไป และสามารถจัดเก็บรูปภาพได้หลายรูปมากกว่าด้วยหน่วยความจำของตัวกล้อง นอกจากนั้นยังสามารถเปลี่ยนหน่วยความจำแบบ Flash นี้ได้เมื่อเก็บภาพจนเต็มแล้ว เพราะหน่วยความจำชนิดนี้สามารถถอดออกจากร่างกล้องได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

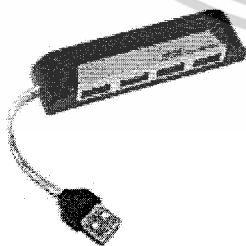
3.2 ระบบเชื่อมต่อเว็บแคมกับคอมพิวเตอร์

USB หรือ Universal Serial Bus

USB ย่อมาจาก Universal Serial Bus หมายถึง ช่องทางที่ใช้ในการสื่อสารระหว่างคอมพิวเตอร์กับอุปกรณ์อื่นๆ หรือเรียกว่า พอร์ต ซึ่ง พอร์ตนี้มีลักษณะสี่เหลี่ยมเล็กๆ ที่ติดมากับเครื่องคอมพิวเตอร์ หรือ notebook อาจกล่าวได้ว่าถ้าใครซื้อเครื่องในยุคปัจจุบันนี้ก็จะไม่มี พอร์ต USB นี้ติดมากับเครื่องเลย แล้วแต่ spec ของเครื่องว่าจะมีมาให้ที่พอร์ต

ระบบยูเอสบีเป็นการออกแบบโดยประกอบด้วย โฮสต์คอนโทรลเลอร์ และอุปกรณ์หลาย ๆ อุปกรณ์ที่ต่อเชื่อมในรูปแบบต้นไม้โดยใช้อุปกรณ์พิเศษเรียกว่า "ฮับ (hub)" โดยมีข้อจำกัดของการต่อเชื่อมฮับได้ไม่เกิน 5 ระดับต่อ 1 คอนโทรลเลอร์ และสามารถต่อเชื่อมได้กับอุปกรณ์ 127 อุปกรณ์ต่อ 1 โฮสต์คอนโทรลเลอร์ โดยนับรวมฮับเป็นอุปกรณ์ด้วย ในคอมพิวเตอร์รุ่นใหม่ ๆ จะมีโฮสต์คอนโทรลเลอร์อยู่หลายช่อง ซึ่งพอเพียงสำหรับการต่อเชื่อมอุปกรณ์จำนวนมาก ๆ การต่อเชื่อมแบบยูเอสบีไม่จำเป็นต้องมีจุดสิ้นสุด (terminator) เหมือนการต่อเชื่อมแบบ SCSI

การออกแบบของยูเอสบีมีจุดมุ่งหมายที่จะขจัดความจำเป็นในการเพิ่มการ์ดขยาย (expansion card) ในช่องการเชื่อมต่อแบบบัส ISA หรือ PCI และเพิ่มความสามารถของรูปแบบ plug-and-play โดยยอมให้อุปกรณ์ต่าง ๆ สามารถถอด สับเปลี่ยน หรือเพิ่มจากระบบโดยไม่ต้องปิดคอมพิวเตอร์หรือบูตระบบใหม่ เมื่ออุปกรณ์ใหม่ถูกต่อเชื่อมเข้าสู่บัสเป็นครั้งแรก โฮสต์จะทำการระบุอุปกรณ์ และติดตั้งตัวขับอุปกรณ์ (device driver) ที่จำเป็นในการใช้งานอุปกรณ์นั้น



รูปที่ 3.2.1 Hub USB

ยูเอสบีสามารถเชื่อมต่อกับอุปกรณ์ต่อพ่วง (peripherals) เช่น เมาส์ แป้นพิมพ์ แพนดเกม จอยสติ๊ก สแกนเนอร์ กล้องถ่ายภาพดิจิทัล เครื่องพิมพ์ ฮาร์ดดิสก์ และ อุปกรณ์เครือข่าย เป็นต้น ยูเอสบีได้กลายเป็นเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบการเชื่อมต่อมาตรฐานสำหรับอุปกรณ์มีลติมีเดีย เช่น สแกนเนอร์ หรือกล้องถ่ายภาพดิจิทัล และนิยมนำไปทดแทนการเชื่อมต่อแบบเดิม เช่น การเชื่อมต่อแบบขนาน (parallel) สำหรับเครื่องพิมพ์ การเชื่อมต่อแบบอนุกรม (serial) สำหรับโมเด็ม ทั้งนี้เนื่องจากยูเอสบีช่วยลดยอดข้อจำกัดหลาย ๆ ด้านของการเชื่อมต่อแบบเดิม เช่น การเชื่อมต่อเครื่องพิมพ์หลาย ๆ เครื่องเข้ากับคอมพิวเตอร์เครื่องเดียว ในปี 2547 มีอุปกรณ์ยูเอสบีประมาณ 1 พันล้านชิ้นถูกผลิตขึ้น และอุปกรณ์ต่อพ่วงใหม่ๆ ที่ถูกผลิตออกมาก็จะใช้รูปแบบการต่อเชื่อมแบบยูเอสบี มีเพียงอุปกรณ์ที่ต้องการความสามารถในการส่งผ่านข้อมูลมาก ๆ เท่านั้นที่ไม่สามารถใช้ยูเอสบี เช่น จอภาพแสดงผล หรือ มอนิเตอร์ และอุปกรณ์ดิจิทัลวิดีโอคุณภาพสูง เป็นต้น

สำหรับรุ่นหรือ Version ของ USB จนถึงปัจจุบันมี 2 รุ่นด้วยกันคือ USB 1.1 หรือ USB 2.0 มีรายละเอียดคือ

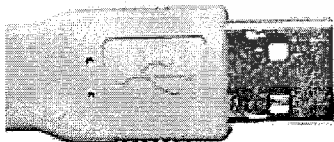
- USB เวอร์ชัน 1.1 มีความเร็วในการรับส่งข้อมูล อยู่ที่ 12 Mbits ต่อวินาที
- USB เวอร์ชัน 2.0 มีความเร็วในการรับส่งข้อมูล อยู่ที่ 480 Mbits ต่อวินาที

สำหรับตอนนี้แล้วส่วนมากจะเป็น USB version 2.0 ซึ่งมีรายละเอียดที่ต้องคำนึงถึงคือ

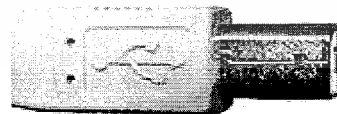
- USB 2.0 แบบ Full Speed มีความเร็วในการรับส่งข้อมูล 12 Mbits ต่อวินาที
- USB 2.0 แบบ Hi-Speed มีความเร็วในการรับส่งข้อมูล 480 Mbits ต่อวินาที

หัวต่อยูเอสบี (USB Connector)

หัวต่อของยูเอสบี แบ่งออกเป็น 2 ประเภทหลักๆคือ แบบ A และ B ดังแสดงในรูปด้านล่าง



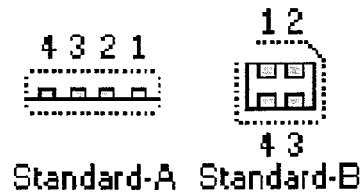
หัวต่อแบบ A



หัวต่อแบบ B

รูปที่ 3.2.2 หัวต่อยูเอสบี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



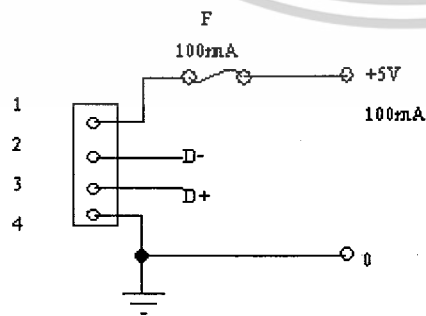
รูปที่ 3.2.3 ช่องสัญญาณของหัวยูเอสบี

เมื่อดูเทียบกับตารางด้านล่างก็จะเห็นว่าสายต่างๆ ที่อยู่ภายในนั้นมีสายอะไรบ้างดังตารางที่ 2.1

ตารางที่ 3.2.1 ช่องสัญญาณยูเอสบี

Pin	ฟังก์ชัน (โฮส)	ฟังก์ชัน (อุปกรณ์)
1	V_{BUS} (4.75-5.25 V)	V_{BUS} (4.4-5.25 V)
2	D-	D-
3	D+	D+
4	Ground	Ground

สัญญาณยูเอสบีถูกส่งผ่านโดยสายส่งข้อมูลคู่แบบบิดเกลียว (twisted pair) แทนโดยสัญญาณ D+ และ D- ในการส่งสัญญาณ สัญญาณจะถูกส่งในลักษณะความต่าง คือ ในกรณีการส่งสัญญาณ “0” สัญญาณ D- จะมีแรงดันที่สูงกว่า D+ ส่วนในกรณีการส่งสัญญาณ “1” สัญญาณ D+ จะมีแรงดันที่สูงกว่า D- (Bus USB) บัสยูเอสบีจะต้องใช้แหล่งจ่ายไฟขนาดแรงดัน 5 โวลต์ และสามารถรองรับโหลดได้ถึง 100 มิลลิแอมป์ สิ่งหนึ่งที่จะขาดไม่ได้ก็คือการป้องกันการลัดวงจร ซึ่งทำหน้าที่เสมือนกับฟิวส์ทั่วไป ขาของข้อมูล D+ และ D- อาจถูกใช้งานรวมกันได้โดยใช้อุปกรณ์ยูเอสบีชนิดพิเศษ เช่น ไมโครคอนโทรลเลอร์แบบพิเศษที่ต้องใช้แหล่งจ่ายไฟที่ส่งผ่านมากับบัสยูเอสบีทำให้อุปกรณ์ต่อพ่วงนั้นสามารถรับกระแสไฟฟ้าในการใช้งานได้สูงถึง 500 มิลลิแอมป์ ดังแสดงในรูปที่ 3.2.4



รูปที่ 3.2.4 การใช้งาน USB ในลักษณะตัวควบคุมเสถียรภาพของแหล่งจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงดันไฟฟ้าที่ยูเอสบีซียอมรับได้จะอยู่ในช่วง 4.2 V ถึง 5.25 V ถ้าหากว่าอุปกรณ์มีการดึงกระแสไฟฟ้าเกินกว่า 100 มิลลิแอมป์ ก็จะมีการแจ้งสู่ระบบ ซึ่งก็จะยินยอมให้ส่งผ่านข้อมูลได้ในกรณีที่มิกระแสไฟฟ้าใช้งานเพียงพอเท่านั้น ตามมาตรฐานของยูเอสบีซีแล้วค่าการสิ้นเปลืองกระแสไฟฟ้าของบีสจะถูกลำกวดไว้โดยอัตโนมัติ ดังนั้นหากมีการดึงกระแสไฟเกินกว่าที่ยินยอม แรงดันไฟเลี้ยงก็จะถูกตัดออกไป

การทำงานของ ยูเอสบี

เมื่อเปิดเครื่องคอมพิวเตอร์ขึ้นมา ระบบจะทำการตรวจสอบอุปกรณ์ต่อพ่วงทาง พอร์ทยูเอสบี (Port USB) แล้วจะกำหนด Address ให้แต่ละอุปกรณ์เรียกว่ากระบวนการ “Enumeration” หรือเมื่อเราทำการ Plug อุปกรณ์เข้าไปยัง พอร์ทยูเอสบี ระบบก็จะทำการตรวจสอบด้วยกระบวนการ “Enumeration” ทันที เพื่อตรวจสอบชนิดของข้อมูลที่จะทำการรับ หรือ จัดส่งให้กับอุปกรณ์ที่ต่อพ่วงเข้ามา ซึ่งแบ่งชนิดของข้อมูลได้เป็น 3 แบบ คือ

1. Interrupt เสมือนเป็นการตอบรับ โดยส่งข้อมูลที่ละน้อยๆ เช่น อุปกรณ์จำพวก Mouse หรือ Keyboard หรือ GamePad ต่างๆ จะทำการส่งข้อมูลให้กับเครื่องคอมพิวเตอร์คราวละเล็กน้อยเท่านั้น และจะส่งแบบไม่ต่อเนื่อง ตามแต่ลักษณะการใช้งาน
2. Bulk หรือการส่งข้อมูลคราวละมากๆ เป็นก้อนๆ เช่น การพิมพ์งานซึ่งเครื่องคอมพิวเตอร์ จะส่งข้อมูลให้กับเครื่องพิมพ์คราวละมากๆ เป็นต้น ซึ่งระบบจะทำการตรวจสอบข้อมูลทั้งความถูกต้อง และความครบถ้วนด้วย
3. Isochronous หรือการส่งข้อมูลแบบต่อเนื่อง เป็น Stream เช่น พวกร Speaker หรือ เว็บแคม ที่จะมาการส่งข้อมูลอย่างต่อเนื่องแบบ เรียลไทม์ ระหว่างเครื่องคอมพิวเตอร์ หรือ โฮสต์ (Host) กับ อุปกรณ์ต่อพ่วง ซึ่งใน Mode นี้ จะไม่มีการตรวจสอบความถูกต้องของข้อมูล ว่าได้รับครบถ้วนถูกต้องหรือไม่

เครื่องคอมพิวเตอร์ หรือ โฮสต์นั้น จะส่งคำสั่ง หรือ query ไปยังอุปกรณ์ ผ่านทาง “Control Packet” โดยเครื่องคอมพิวเตอร์นั้น จะทำการกันเนื้อที่ 90% ของ Bandwidth ทั้งหมด (USB 1.1 ที่ 12 Mbps และ USB 2.0 ที่ 480 Mbps) สำหรับใช้งานการส่งข้อมูลแบบ Isochronous หากมีการใช้งานถึง 90% เมื่อไรระบบก็จะทำการปฏิเสธการร้องขอในแบบ Interrupt และ Isochronous ที่เข้ามาใหม่ทันที โดย 10% ที่กันไว้ นั้นจะไว้ใช้สำหรับการส่งข้อมูลแบบ Bulk และสำหรับ Control Packet ของ โฮสต์นั่นเอง

หัวต่อของ ยูเอสบี ทั้ง 2 แบบคือ หัวต่อแบบ “A” และ “B” ซึ่งจะทำงานคล้ายๆ กัน แต่ว่ามีทิศทางในการส่งข้อมูลที่สลับกัน โดยที่หัวต่อแบบ “A” จะมีทิศทางของข้อมูลจากอุปกรณ์ ไปเข้าเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์ เรียกว่า “upstream” เช่น MouseKeyboard จะส่งข้อมูลเข้าหาเครื่องคอมพิวเตอร์ เพื่อประมวลผล ส่วนหัวต่อแบบ “B” จะมีทิศทางกลับกัน คือข้อมูลจะวิ่งเข้าหาอุปกรณ์ เรียกว่า “downstream” เช่นคอมพิวเตอร์ส่งข้อมูลไปหา Printer เพื่อทำการพิมพ์ภาพ เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การออกแบบและพัฒนาโครงการ

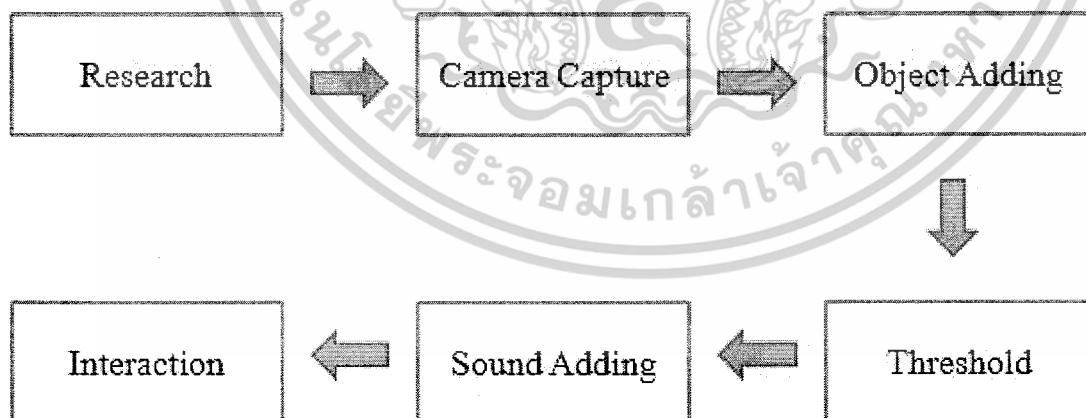
4.1 บทนำ

ในโครงการเขียนโปรแกรมนี้ใช้โปรแกรมของ C++ เป็นหลักในการออกแบบ โดยนำมาประยุกต์ใช้กับ Webcam เพื่อใช้ในการเปิดกล้อง การจับภาพ และ อื่นๆอีกมากมาย

4.2 โครงสร้างและการทำงานของระบบ

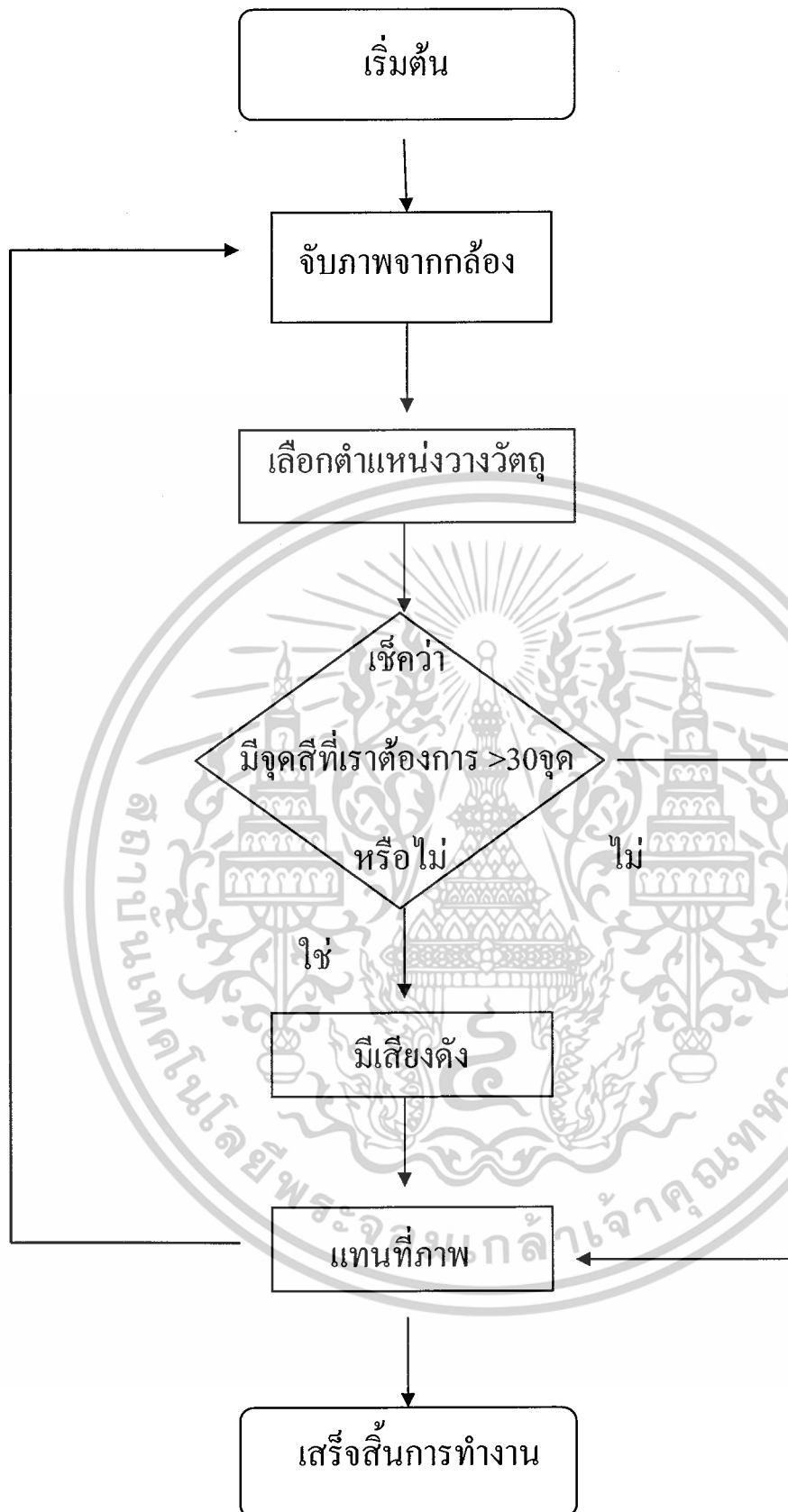
โครงสร้างของระบบมีดังต่อไปนี้

เปิด Webcam - รับภาพ Webcam - ทำ Threshold ของภาพ - แปลงภาพเป็นระดับสีเทา - หาขอบรูปจากภาพระดับสีเทา - แสดงภาพที่ได้ - สร้าง object + ใส่ effect เมื่อมีการสัมผัส



รูปที่ 4.2.1 แสดงลำดับ ส่วนประกอบของ โครงการนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2.2 Flow Chart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการดำเนินงาน	มี.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.
	1 2 3 4	1 2 3 4	1 2 3 4	1 2 3 4	1 2 3 4	1 2 3 4	1 2 3 4	1 2 3 4	1 2 3 4
แบ่งหน้าที่ศึกษาข้อมูล									
เขียนcode แสดงผลบนหน้าจจ จาก Webcam									
เขียนcode แยก frame ภาพ									
เขียน cod ทำthreshold และ detect edge									
สร้าง object และใส่ effect เมื่อมีการสัมผัส									

วางแผนไว้
ปฏิบัติได้จริง

รูปที่ 4.2.3 แสดง ระยะเวลาของการทำงาน ที่วางแผนไว้ และที่ปฏิบัติได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2.4 รูปจำลองการใช้งาน

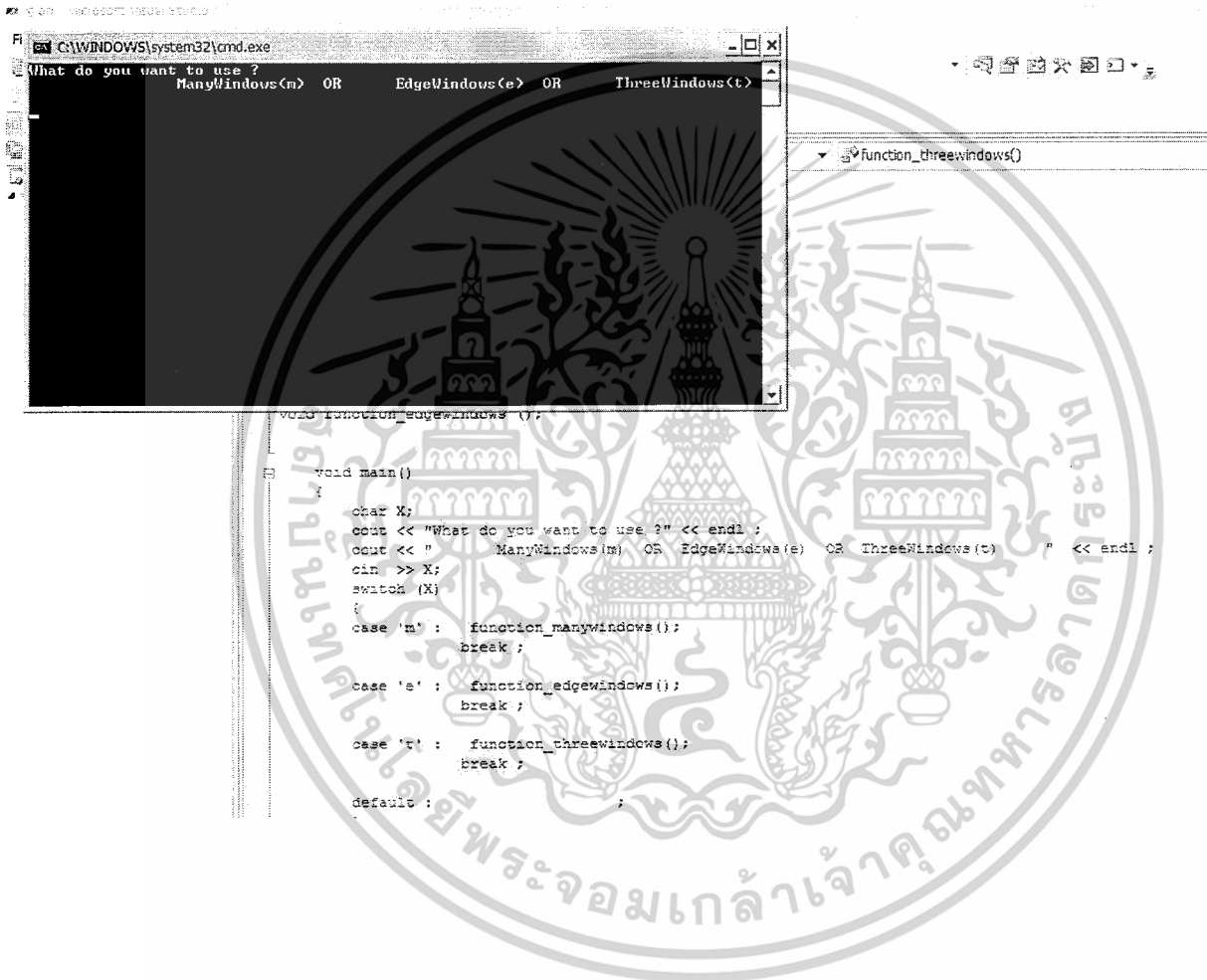
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การทดลองและผลการทดลอง

จากการได้ทดลองทำการ RUN CODE ที่เขียนขึ้นมาเพื่อทำการแสดงภาพที่ได้จาก WebCam

บนจอ Monitor เป็นไปดังนี้



```

C:\WINDOWS\system32\cmd.exe
What do you want to use ?
ManyWindows(m) OR      EdgeWindows(e) OR      ThreeWindows(t)

void function_manywindows ();
void function_edgewindows ();
void main()
{
    char X;
    cout << "What do you want to use ?" << endl ;
    cout << "      ManyWindows(m) OR      EdgeWindows(e) OR      ThreeWindows(t) " << endl ;
    cin  >> X;
    switch (X)
    {
        case 'm' :    function_manywindows();
                    break ;
        case 'e' :    function_edgewindows();
                    break ;
        case 't' :    function_threewindows();
                    break ;
        default :    ;
    }
}

```

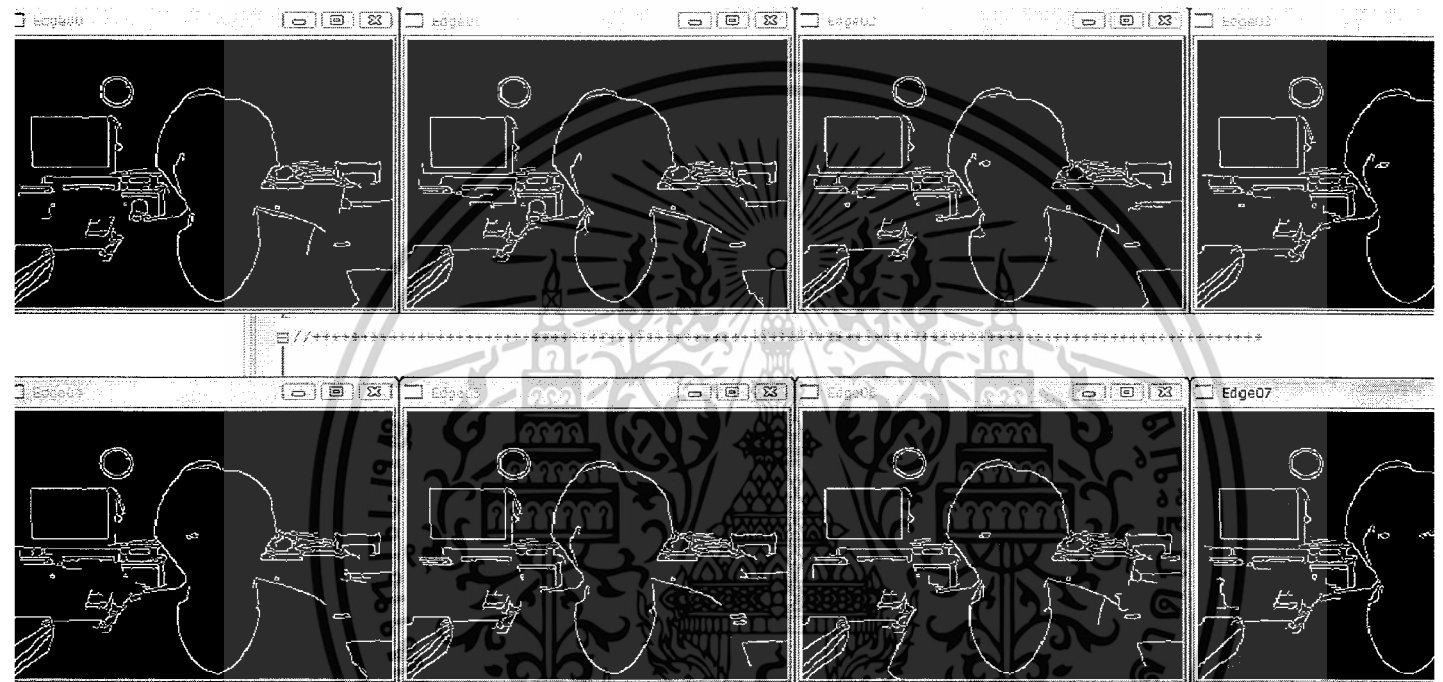
รูปที่ 5.1 หน้าจอแสดงผลการรันโปรแกรมที่เขียนขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



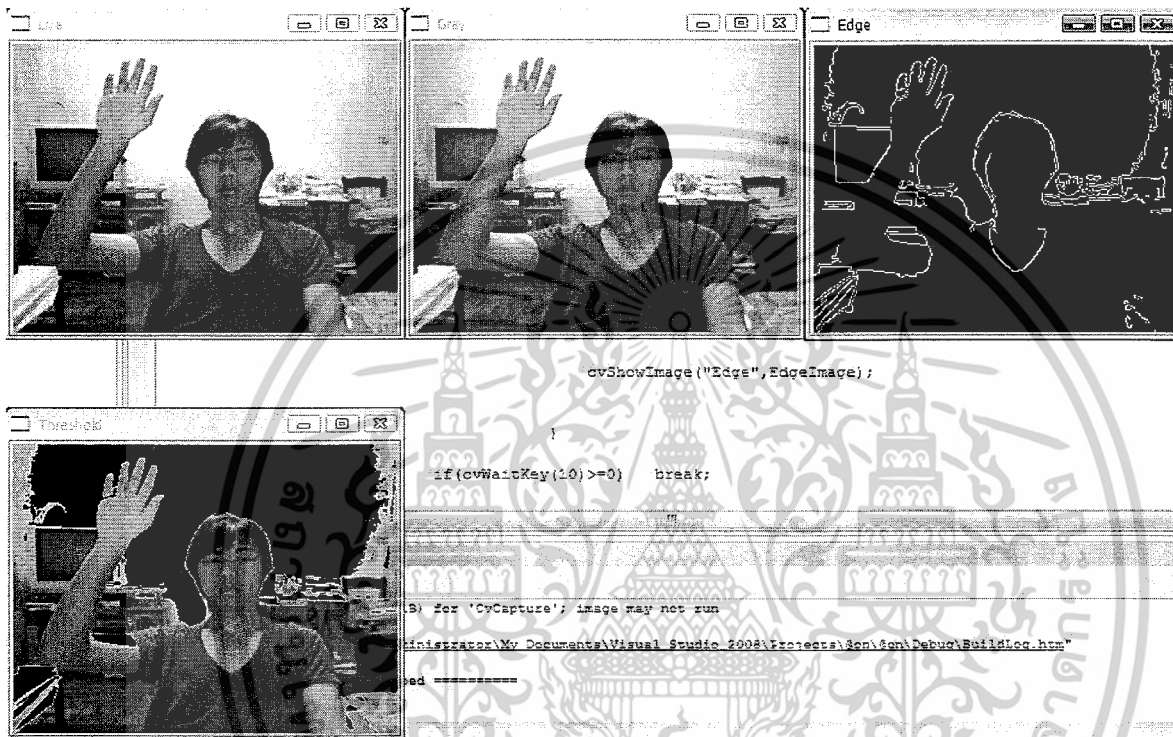
รูปที่ 5.2 แสดงภาพที่สะท้อนที่กล้องจับมาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 แสดงภาพที่สะเฟรมพร้อมทำ edge detection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



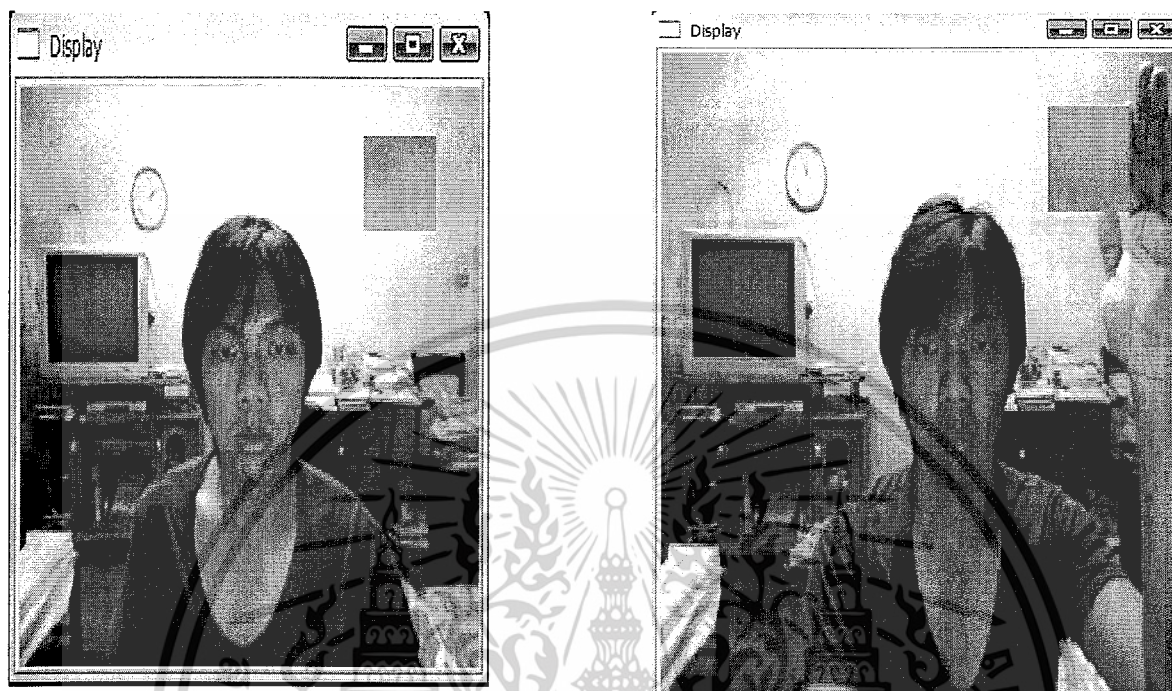
รูปที่ 5.4 แสดงภาพจากกล้อง Grayscale Threshold และ Edge

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 ทำการเปรียบเทียบการกรอง ค่า H อย่างเดียว กับ การกรองค่า H และ ค่า S

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 ทำการแทนที่ภาพในขณะที่ภาพเคลื่อนไหว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

วิเคราะห์และสรุปผลการทดลอง

จากการที่ได้ทำการทดลอง เขียนโปรแกรมดึงภาพจากกล้อง ทำให้สรุปได้ว่า

1. การที่ คอมพิวเตอร์ จะรู้จักกับกล้องนั้น มันจำเป็นต้องทำการลง driver ของกล้องเสียก่อน เพราะถ้า ไม่ลง driver คอมพิวเตอร์ก็จะไม่รู้ว่าไอที่ต่อกับมันนั้นมันคืออะไร
2. กรณีที่มีกล้อง มากกว่า 1 ตัวติดอยู่กับ คอมพิวเตอร์ ตอนทำการ รัน โปรแกรมมันมันจะให้ทำการเลือกกล้องที่ต้องการจะใช้ก่อนแสดงผล
3. กรณีที่ต้องการแสดงออกมา ทีละ Frame นั้น เราต้องทำการเขียน โปรแกรมให้มีวนลูป และเลือก หน้าต่างที่ต้องการแสดงเอง
4. ในการแสดงผล หลายหน้าต่างนั้น ทางที่ดีควรทำการจัดตำแหน่งของหน้าต่างแต่ละ หน้าต่างด้วย เพื่อเวลา แสดงผลจะได้ไม่ซ้อนทับกัน
5. ในการเขียนโปรแกรมนั้น เวลาเราสร้างที่สำหรับเก็บรูป หรือหน้าต่าง นั้น เวลาใช้งาน เสร็จเราควรจะทำลายมันทิ้งด้วย เพราะจะได้เป็นการคืนค่าให้กับคอมพิวเตอร์
6. ขอบเขตของระยะเวลา ใช้งาน ไม่เกิน 2 เมตร

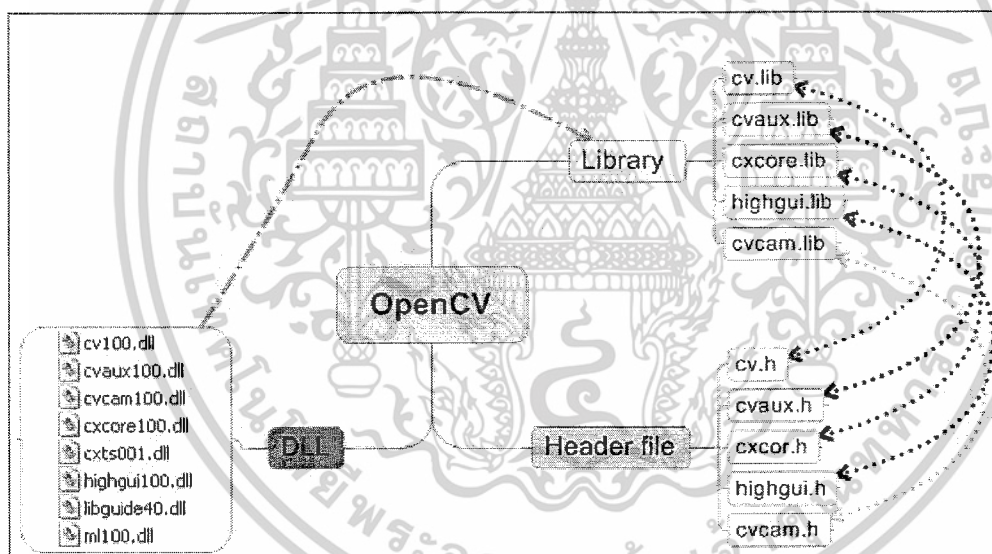
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีลง OpenCV กับ Visual Studio C++

OpenCV เป็นไลบรารีสำหรับใช้ในการประมวลผลภาพ (Image Processing) ซึ่งเป็นไลบรารีโอเพนซอร์ส (Open Source) สามารถดาวน์โหลดใช้งานได้ฟรี ไลบรารีต่างๆของ OpenCV ได้พัฒนาขึ้นด้วยบริษัทอินเทล (Intel) จุดเด่นในด้านความสามารถของไลบรารี OpenCV คือสามารถประมวลผลภาพดิจิทัลได้ทั้งภาพนิ่ง และภาพเคลื่อนไหวเช่น ภาพจากกล้องวิดีโอ หรือไฟล์วิดีโอ เป็นต้น โดยไม่ยึดติดทางด้านฮาร์ดแวร์ทำให้ OpenCV สามารถพัฒนาโปรแกรมได้หลากหลายภาษา รวมถึงมีฟังก์ชันสำเร็จรูปสำหรับการจัดการข้อมูลภาพ และการประมวลผลภาพพื้นฐานเช่น การหาขอบภาพ การกรองข้อมูลภาพ โดยฟังก์ชันต่างๆของ OpenCV จะสามารถเรียกใช้งานได้จะต้องมีการเรียก ไฟล์ส่วนหัว (Header file) และลิงค์ (Link) ไลบรารีต่างๆ รวมถึง DLL (Dynamic Link Library) โดยมีความสัมพันธ์ดังภาพที่ 1



ภาพที่ 1 ความสัมพันธ์ไลบรารีของ OpenCV

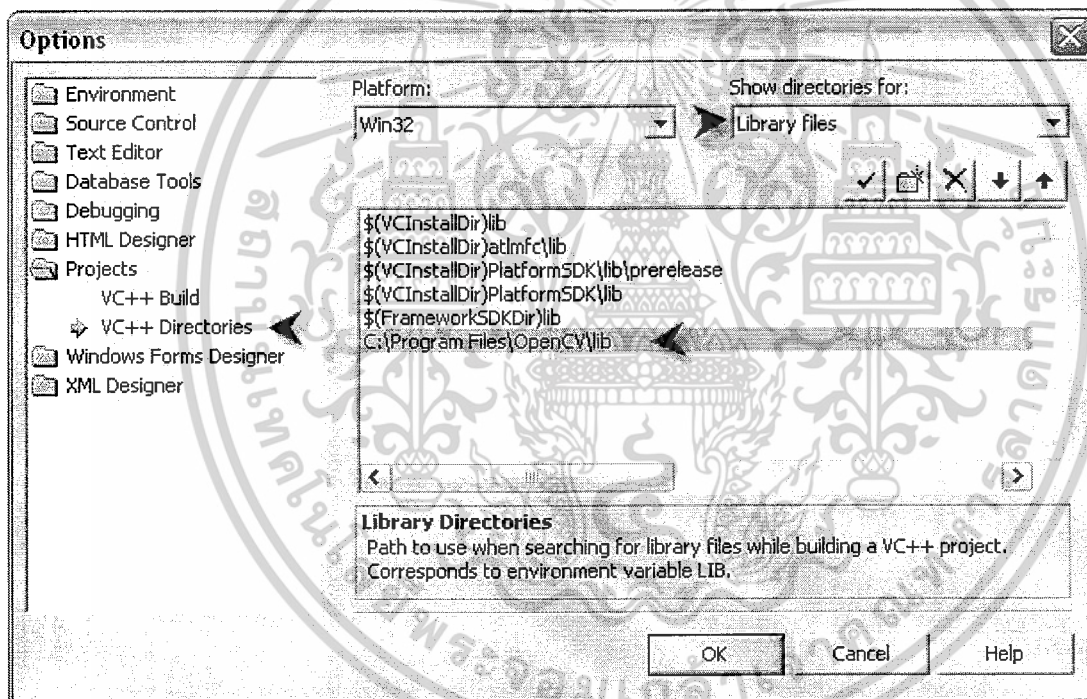
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมโยงไลบรารี OpenCV สำหรับ Visual C++.NET 2003

เมื่อทำการติดตั้งไลบรารี OpenCV ลงเครื่องคอมพิวเตอร์เรียบร้อยแล้ว ก่อนที่โปรแกรมจะสามารถเรียกใช้คำสั่ง OpenCV ได้ จะต้องกำหนดการเชื่อมโยงไฟล์ไลบรารีต่างๆ ให้กับ Visual C++.NET 2003 ดังนี้

1.เปิดโปรแกรม Visual C++.NET 2003 ที่เมนูบาร์เลือก Tool> Options จะปรากฏไดอะล็อก ดังภาพที่ 2 จากนั้นเลือกที่ "Project -> VC++ Directories" ค้นหาไดเรกทอรีที่ติดตั้ง OpenCV แล้วทำการเพิ่มตำแหน่งพาร์ท (Path) สำหรับการเชื่อมโยง Library ของ OpenCV กับ Visual C++.NET ดังนี้

"C:\Program Files\OpenCV\lib"



ภาพที่ 2 การกำหนด Library files

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เลือกที่ List box เปลี่ยนจาก Library files เป็น Include files แล้วทำการเพิ่มตำแหน่งไดเรกทอรี สำหรับการเรียกใช้ Header file ดังภาพที่ 3

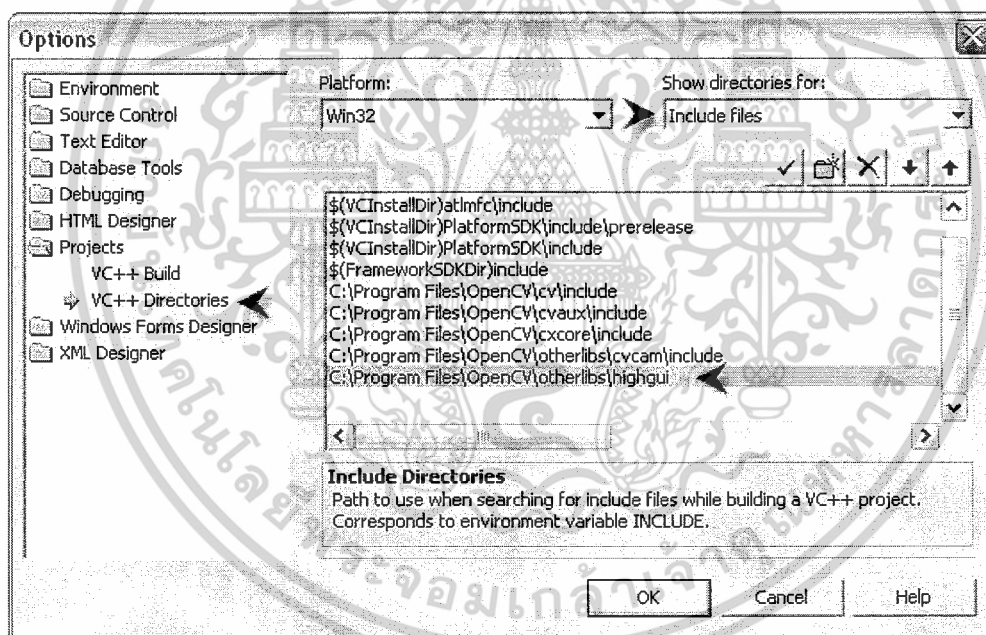
"C:\Program Files\OpenCV\cv\include"

"C:\Program Files\OpenCV\cxcore\include"

"C:\Program Files\OpenCV\otherlibs\highgui"

"C:\Program Files\OpenCV\cvaux\include"

"C:\Program Files\OpenCV\otherlibs\cvcam\include"



ภาพที่ 3 การกำหนด Include files

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลือกที่ List box เปลี่ยนจาก Include files เป็น Source files แล้วทำการเพิ่มตำแหน่งไดเรกทอรี สำหรับการเรียกใช้ Source file ดังภาพที่ 4 จากนั้นเลือกที่ Finish

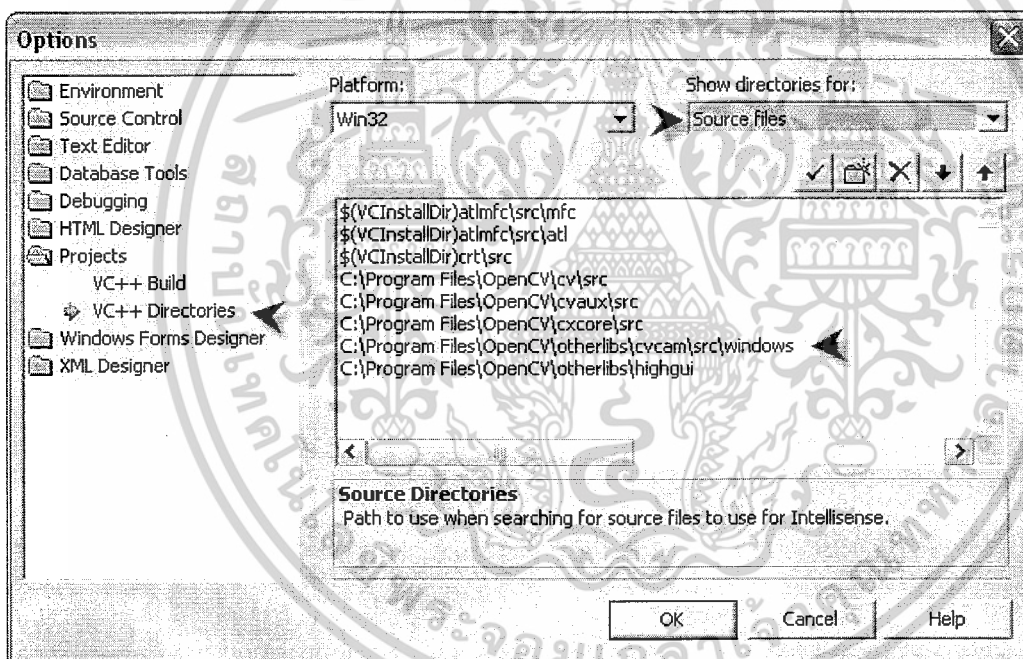
"C:\Program Files\OpenCV\cv\src"

"C:\Program Files\OpenCV\cxcore\src"

"C:\Program Files\OpenCV\cvaux\src"

"C:\Program Files\OpenCV\otherlibs\highgui"

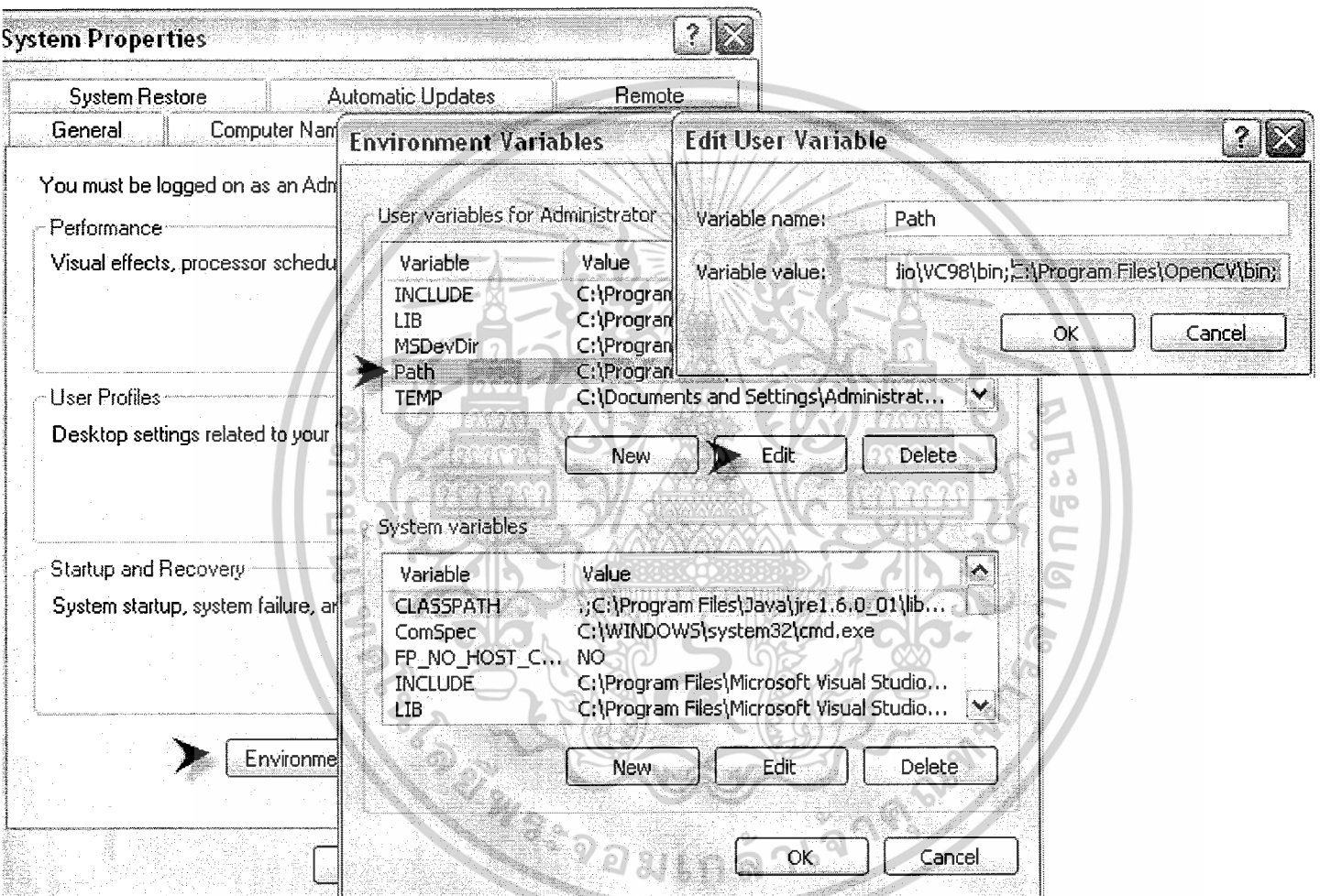
"C:\Program Files\OpenCV\otherlibs\cvcam\src\windows"



ภาพที่ 4 การกำหนด Source files

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. กำหนดค่าตำแหน่งไดเรกทอรีของไฟล์ DLL (Dynamic link library) ให้สามารถค้นหาไฟล์ DLL อัตโนมัติ โดยคลิกขวาที่ My Computer จะปรากฏไอคอน System Properties กำหนดค่าดังภาพที่ 5



ภาพที่ 5 กำหนดตำแหน่งไดเรกทอรีสำหรับค้นหาไฟล์ DLL อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code และ คำอธิบายเบื้องต้น

```

#include <cv.h>
#include <highgui.h>
#include <stdio.h>
#include <string>
#include <iostream>
#include <cvcam.h>
#include <irrKlang.h>
using namespace irrklang;
//using namespace cimg_library;
#pragma comment(lib, "irrKlang.lib") // link with irrKlang.dll
int main(){
ISoundEngine* engine = createIrrKlangDevice();
cvNamedWindow("Display", 2);

//set ขนาดของภาพที่จะแสดง
CvCapture* capture = cvCaptureFromCAM( 0 );
cvSetCaptureProperty( capture,CV_CAP_PROP_FRAME_WIDTH,640 );
cvSetCaptureProperty( capture,CV_CAP_PROP_FRAME_HEIGHT, 480 );

//ประกาศ และ สร้างตัวแปร
IplImage *frame;
IplImage *image,*hsv,*hue;
frame = cvQueryFrame( capture );
image = cvCreateImage(cvGetSize(frame), IPL_DEPTH_8U, 3);
hsv = cvCreateImage(cvGetSize(frame), IPL_DEPTH_8U, 3);
int count=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(;;){
frame = cvQueryFrame( capture );           //เริ่มทำการจับภาพ ที่ชื่อ Frame
cvCvtColor( frame,hsv, CV_BGR2HSV );      //BRG2HSV

//กำหนดตำแหน่งที่จะแทนที่ภาพ
for(int y=100;y<150;y++)
    for(int x=530;x<580;x++){

//การเข้าถึง pixel ของแต่ละสี
((uchar*)(frame->imageData + frame->widthStep*y))[x*3] = 255; // B
((uchar*)(frame->imageData + frame->widthStep*y))[x*3+1] = 0; // G
((uchar*)(frame->imageData + frame->widthStep*y))[x*3+2] = 0; // R

//กำหนดช่วง ค่า H กับ S ที่ต้องการ
if((((uchar*)(hsv->imageData + hsv->widthStep*y))[x*3]<10 || ((uchar*)(hsv->imageData + hsv->
widthStep*y))[x*3] > 170)&& (((uchar*)(hsv->imageData + hsv->widthStep*y))[x*3+1]>40) &&
(((uchar*)(hsv->imageData + hsv->widthStep*y))[x*3+1]<80) ))

count++;
}

//ทำให้มีเสียงและแสดงรูป
if(count>30)
    engine->play2D("bell.wav");
count=0;
cvShowImage( "Display", frame );
if( (cvWaitKey(10) & 255) == ' ' ){    engine->play2D("bell.wav");    break;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

//ทำการทำลายตัวแปรที่สร้างไว้เมื่อ ถึงจุดโปรแกรม

```
cvReleaseImage(&loadimg1);
cvReleaseImage(&frame);
cvReleaseImage(&image);
cvReleaseImage(&hsv);
cvReleaseImage(&hue);
cvReleaseCapture( &capture );
cvDestroyWindow( "Display" );
return 0;
```

}



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. หนังสือเริ่มต้นเขียน โปรแกรมด้วยภาษา C++ ยุทธนา ธิลาศวัฒน์กุล
2. การประมวลผลภาพด้วย Opencv รศ.ดร. ชูชาติ ปิณฑวิรุจน์
3. <http://opencv.willowgarage.com/wiki/>
4. <http://en.wikipedia.org/wiki/OpenCV>
5. <http://public.cranfield.ac.uk/c5354/teaching/dip/opencv/manual/>
6. <http://www.ambiera.com/irrklang/>

