

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง
เว็บเซิร์ฟเวอร์สำหรับการประมวลผลภาพ
Image Processing Server



เลขหมู่.....
เลขทะเบียน..... 72835
วัน,เดือน,ปี..... 23 ส.ย. 2550

b. 11972108
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เว็บเซิร์ฟเวอร์สำหรับการประมวลผลภาพ

Image Processing Server

นายกานต์ แก้วโมรา

นายอิทธิพล พาร์ตัน

อาจารย์ที่ปรึกษา

ดร. ปกรณ์ วัฒนจตุรพร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2549

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เว็บเซิร์ฟเวอร์สำหรับการประมวลผลภาพ

Image Processing Server

ผู้จัดทำ

1. นายกานต์ แก้วโมรา รหัสประจำตัว 47015671
2. นายอิทธิพล พาร์ตน์ รหัสประจำตัว 47015701

ปกรณ วัฒนจตุรพร

อาจารย์ที่ปรึกษา

(ดร. ปกรณ วัฒนจตุรพร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เว็บเซิร์ฟเวอร์สำหรับการประมวลผลภาพ

นายกานต์ แก้วโมรา

นายอิทธิพล พารัตน์

ดร.ปกรณ์ วัฒนจตุรพร อาจารย์ที่ปรึกษา

ปีการศึกษา 2549

บทคัดย่อ

การประมวลผลภาพในแบบต่างๆเป็นที่ต้องการอย่างมาก โดยเฉพาะอย่างยิ่งเมื่อกล้องภาพแบบดิจิตอลได้รับความนิยมอย่างแพร่หลายเช่นในปัจจุบันนี้ ในกรณีที่ผู้ใช้ต้องการตกแต่งภาพที่ถ่ายเพื่อให้ได้ภาพที่มีคุณภาพตามความต้องการ ผู้ใช้ต้องใช้ซอฟต์แวร์เพื่อการประมวลผลภาพนี้ ทั้งนี้ราคาของซอฟต์แวร์เหล่านี้มักจะมีราคาแพง ดังนั้นการสร้างเว็บเซิร์ฟเวอร์ขึ้นมาที่มีความสามารถในการประมวลผลภาพนั้นเป็นอีกทางเลือกหนึ่งของผู้ใช้ ที่สามารถทำให้ผู้ใช้ที่ไม่มีซอฟต์แวร์เพื่อการประมวลผลภาพเป็นของตนเองได้มีโอกาสได้ใช้งานโดยอัพโหลดไฟล์ภาพที่ต้องการตกแต่งขึ้นสู่เครื่องเซิร์ฟเวอร์ จากนั้นผู้ใช้จะเลือกใช้การตกแต่งในรูปแบบต่างๆ ตามที่ต้องการ เมื่อแล้วเสร็จผู้ใช้สามารถโหลดไฟล์ภาพที่ได้รับการปรับแต่งแล้วกลับสู่เครื่องของตนเอง

ทั้งนี้ผู้ใช้สามารถเข้าใช้งานได้ไม่ว่าผู้ใช้จะอยู่ที่ใดก็ตามเพียงแค่ทำการเชื่อมต่อมายังเครื่องเซิร์ฟเวอร์โดยผ่านเครือข่ายอินเทอร์เน็ต

Image Processing Server

Karn Kaewmora

Ittipol Pharat

Dr. Pakorn Watanachaturaporn Advisor

ABSTRACT

Processing of pictures files in various ways are much required especially, when a digital cameras are interested a lot of people like . When the users want to modify and decorate the picture files like as they want. Users have to use a software to process still pictures. Normally an image processing software is expensive, so a web server which can process pictures is a choice for users to process pictures. User upload as picture to the server and modify it image tools provided by the server. Users can download this pictures back to their own computers .

Besides, users can use the software anywhere they are just connecting to the server through the internet.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายด้วยกันบุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ คร.ปกรณ์ วัฒนจตุรพร อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่แนะนำ และช่วยเหลือเสมอมานอกจากนั้นยังได้ให้ความรู้และแนวคิดในการทำงานรวมถึงการใช้ชีวิต อันมีค่ายิ่งให้แก่ผู้เขียน ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก ขอขอบคุณที่ ๆ เพื่อนๆ ห้อง 803 ที่ช่วยเป็นกำลังใจเป็นที่ปรึกษาและให้ประสบการณ์ในการใช้ชีวิตในมหาวิทยาลัยที่มีค่ายิ่งใช้กับผู้เขียน

สุดท้ายนี้ ต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักรยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจเอาใจใส่เสมอมา ในทุกๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอ กราบขอบพระคุณมา ณ ที่นี้

กานต์ แก้วโมรา
อิทธิพล พารัตน์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ขอบเขตของโครงการ	2
1.4 วิธีการดำเนินของโครงการ	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 คำนิยามและความหมายของการทำกระบวนการอิมเมจดิจิทัล	3
2.1 อิมเมจดิจิทัล	3
2.2 การทำกระบวนการอิมเมจดิจิทัล	4
2.3 วิธีการอ่านข้อมูลพิกเซลของอิมเมจ	5
2.4 โมเดลสี (Color Model)	5
2.5 การอ่านข้อมูลอิมเมจในหน่วยความจำ	7
2.6 การทำกระบวนการจุดพิกเซล (Point Processing)	8
2.6.1 การแปลงที่เกี่ยวกับความสว่าง (Luminance Transformation)	8
2.6.2 โปรซีเจอร์แปลงพิกเซลสีเป็นขาวดำ	9
2.6.3 การกลับสี (Invert Color)	9
2.6.4 การแก้ไขพิกเซลด้วย Gamma	10
2.6.5 โปรซีเจอร์ Gamma Correction Transformation	12
2.7 การทำกระบวนการรอบ ๆ จุดพิกเซล (Area processing)	12
2.7.1 โปรซีเจอร์ Convolution	14
2.7.2 ค่า Mask Coefficient	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

2.8 การทำกระบวนการรูปร่างลักษณะ (Morphology Processing)	15
2.8.1 การขยายพิกเซล (Dilation)	15
2.8.2 การลดขนาดพิกเซล (Erosion)	16
บทที่ 3 .NET Framework และ Microsoft Visual C#	17
3.1 .NET Framework	17
3.2 ที่มา .NET Framework	18
3.3 องค์ประกอบของ .NET	18
3.3.1 อุปกรณ์ลูกข่ายเสถียร (Smart Clients)	19
3.3.2 แม่ข่ายหรือเครื่องให้บริการ (Servers)	20
3.3.3 เครื่องมือของนักพัฒนา (Developer Tools)	21
3.4 Microsoft Visual C#.net	22
3.4.1 ไวยากรณ์ C#	22
3.4.2 ตัวแปร	23
3.4.3 การประกาศตัวแปร	25
3.4.5 นิพจน์	26
3.4.6 Operator ทางคณิตศาสตร์	26
3.4.7 ความสำคัญของ Operator	27
3.4.8 Namespace	28
บทที่ 4 ASP.NETเว็บแอปพลิเคชัน	30
4.1 ที่มาของ ASP	30
4.2 ที่มาของ ASP.NET	31
4.3 รู้จักกับ System.Web.UI Namespace	31
4.3.1 คลาส Control	31
4.3.1.1 คุณสมบัติของ Control	32
4.3.1.2 เมธอดในคลาส Control	32
4.3.2 คลาสเพจ	33

สารบัญ(ต่อ)

4.3.2.1 ตัวอย่างคุณสมบัติและเมธอดของคลาสเพจ	33
4.4 ไวยากรณ์สำหรับ Web Form	33
4.4.1 ไคเร็กทีฟ	33
4.4.1.1 @Page	34
4.4.1.2 @Control	34
4.4.1.3 @Import	34
4.4.1.4 @Register	34
4.4.1.5 @Assembly	34
4.4.1.6 @OutputCache	34
4.4.2 ขอบเขตการประกาศคำสั่ง	35
4.4.3 HTML-Control Syntax	35
4.4.4 Custom-Control Syntax	36
4.4.5 Data-Binding Expressions	36
4.5 การพัฒนาเว็บแอปพลิเคชันด้วย ASP.NET	37
4.5.1 ส่วนประกอบคอมโพเนนต์ของ Web Form	37
4.5.2 วงจรชีวิตของ Web Form	37
4.5.2.1 การกำหนดรูปแบบ	39
4.5.2.2 การถือครองอีเวนต์	40
4.5.2.3 การจบการทำงาน	40
4.6 การจัดการ State และ Scalability	40
4.6.1 ASP.NET Session-State Management	40
บทที่ 5 การทดลองและผลการทดลอง	42
5.1 การใช้งานเบื้องต้น	43
5.2 การเข้าใช้ระบบ ในส่วนของเว็บแอปพลิเคชัน	44

สารบัญ(ต่อ)

5.3 การใช้งานในรูปแบบต่างๆ	45
บทที่ 6 บทสรุป	55
6.1 บทสรุป	55
6.2 ปัญหาและอุปสรรค	55
6.3 แนวทางในการพัฒนา	55
บรรณานุกรม	56



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่	หน้า
2- 2.1 ระบบพิกัด Space	3
2- 2.2 การทำกระบวนการอิมเมจดิจิทัล	4
2- 2.3 โมเดลสี	6
2-2.4 อิมเมจต้นฉบับและอิมเมจส่วนประกอบสีต่าง ๆ	7
2-2.5 การแปลงแบบ Negative	10
2-2.6 Gamma Correction Transformation	11
2-2.7 วิธีการคำนวณ Convolution	13
3-3.1 องค์ประกอบของ .NET	19
3-3.2 รูปตัวอย่างอุปกรณ์ บางส่วนที่มีการใช้งานเพื่อเชื่อมต่อเข้ากับเครือข่ายในยุคของ .NET	20
3-3.4 เครื่องให้บริการ Servers	21
3-3.5 โปรแกรม Visual Studio .NET	21
5-5.1 แสดงรูปหน้าเว็บ	43
5-5.2 แสดงรูปหน้าต่างเมื่อเลือกรูปภาพที่ต้องการแล้วคลิกที่ Upload	44
5-5.3 แสดงการใช้งานรูปแบบ Blur	45
5-5.4 แสดงการใช้งานรูปแบบ Contrast	46
5-5.5 แสดงการใช้งานรูปแบบ Sharpen	46
5-5.6 แสดงการใช้งานรูปแบบ Brightness	47
5-5.7 แสดงการใช้งานรูปแบบ Negative	47
5-5.8 แสดงการใช้งานรูปแบบ Remove noise	48
5-5.9 แสดงการใช้งานรูปแบบ Grayscale	48
5-5.10 แสดงการใช้งานรูปแบบ Flip Horizon	49
5-5.11 แสดงการใช้งานรูปแบบ Flip Vertical	49
5-5.12 แสดงการใช้งานรูปแบบ Kodalith	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
5-5.13 แสดงการใช้งานรูปแบบ Watercolor	50
5-5.14 แสดงการใช้งานรูปแบบ Mosaic Effect	51
5-5.15 แสดงการใช้งานรูปแบบ Outline	51
5-5.16 แสดงการใช้งานรูปแบบ Frame Black	52
5-5.17 แสดงการใช้งานรูปแบบ Frame White	52
5-5.18 แสดงการใช้งานรูปแบบ Resize	53
5-5.19 แสดงการใช้งานรูปแบบ Text	53
5-5.20 แสดงการใช้งานรูปแบบ Zoom	54
5-5.21 แสดงการใช้งานรูปแบบ Undo	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของโครงการ

คอมพิวเตอร์นั้นมีความสำคัญในชีวิตประจำวันเป็นอย่างมากในปัจจุบัน ทำให้การใช้งานในด้านต่างๆ นั้นแล้วแต่เพียงพาดำเนินการของคอมพิวเตอร์ เช่น การประมวลผลรูปภาพ

การประมวลผลภาพในแบบต่างๆเป็นที่ต้องการอย่างมาก โดยเฉพาะอย่างยิ่งเมื่อกล้องภาพแบบดิจิทัลได้รับความนิยมอย่างแพร่หลายเช่นในปัจจุบันนี้ ในกรณีที่ผู้ใช้ต้องการตกแต่งภาพที่ถ่ายเพื่อให้ได้ภาพที่มีคุณภาพตามความต้องการ ผู้ใช้ต้องใช้ซอฟต์แวร์เพื่อการประมวลผลภาพนี้ ทั้งนี้ราคาของซอฟต์แวร์เหล่านี้มักจะมีราคาแพง ดังนั้นการสร้างเว็บเซิร์ฟเวอร์ขึ้นมาที่มีความสามารถในการประมวลผลภาพนั้นเป็นอีกทางเลือกหนึ่งของผู้ใช้ ที่สามารถทำให้ผู้ใช้ที่ไม่มีซอฟต์แวร์เพื่อการประมวลผลภาพเป็นของตัวเองได้มีโอกาสได้ใช้งานโดยอัตโนมัติภาพที่ต้องการตกแต่งขึ้นสู่เครื่องเซิร์ฟเวอร์ จากนั้นผู้ใช้จะเลือกใช้การตกแต่งในรูปแบบต่างๆ ตามที่ต้องการ เมื่อแล้วเสร็จผู้ใช้สามารถโหลดไฟล์ภาพที่ได้รับการปรับแต่งแล้วกลับสู่เครื่องของตัวเอง

ทั้งนี้ผู้ใช้สามารถเข้าใช้งานได้ไม่ว่าผู้ใช้จะอยู่ที่ใดก็ตามเพียงแค่ทำการเชื่อมต่อมายังเครื่องเซิร์ฟเวอร์โดยผ่านเครือข่ายอินเทอร์เน็ต

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1. เพื่อศึกษาวิธีการประมวลผลรูปภาพ
- 1.2.2. เพื่อศึกษาการสร้างเว็บเซิร์ฟเวอร์
- 1.2.3. เพื่อศึกษาและเปรียบเทียบการทำงานของภาษา Visual C# .NET กับภาษาอื่น
- 1.2.4. เพื่อสามารถนำความรู้ที่ได้ศึกษากับงานไปใช้ได้จริง

1.3 ขอบเขตของโครงการ

ในปฏิญานិพนธ์ฉบับนี้ได้นำเสนอวิธีการศึกษา และการสร้างเว็บไซต์เพื่อที่จะทำการประมวลผลภาพที่ ผู้ใช้งาน(User) สามารถที่จะเข้ามายังเว็บไซต์เพื่อที่จะอัปโหลดรูปภาพมายังเว็บไซต์เพื่อที่จะเลือกเครื่องมือในการตกแต่งรูปภาพโดยจะมีหลายรูปแบบ จนเมื่อที่ ผู้ใช้งาน (User) พอใจกับรูปภาพที่ทำการปรับแต่งแล้วก็สามารถที่จะ โหลดไฟล์รูปภาพนั้นกลับสู่เครื่องคอมพิวเตอร์ของผู้ใช้งาน(User)ได้ ทั้งนี้ผู้ใช้สามารถเข้าใช้งานได้ไม่ว่าผู้ใช้จะอยู่ที่ใดก็ตามเพียงแค่ทำการเชื่อมต่อมายังเครื่องเซิร์ฟเวอร์

1.4 วิธีการดำเนินการ

1. ศึกษาการทำงานของเว็บไซต์ว่ามีการทำงานอย่างไรบ้าง
2. ศึกษาภาษาที่ใช้ในการสร้างเว็บไซต์
3. ศึกษาการประมวลผลภาพในรูปแบบต่างๆ
4. พัฒนาส่วนประกอบต่างๆ และนำมารวมเข้าด้วยกัน
5. ทดสอบระบบต่างๆ และแก้ไขจุดบกพร่อง

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับความรู้ความเข้าใจเกี่ยวกับการประมวลผลรูปภาพ
2. สามารถสร้างเว็บไซต์เพื่อประมวลผลรูปภาพ
3. ได้รับความรู้ความเข้าใจเกี่ยวกับหลักการการประมวลผลรูปภาพ
4. ได้รับความรู้ความเข้าใจเกี่ยวกับการสร้างเว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

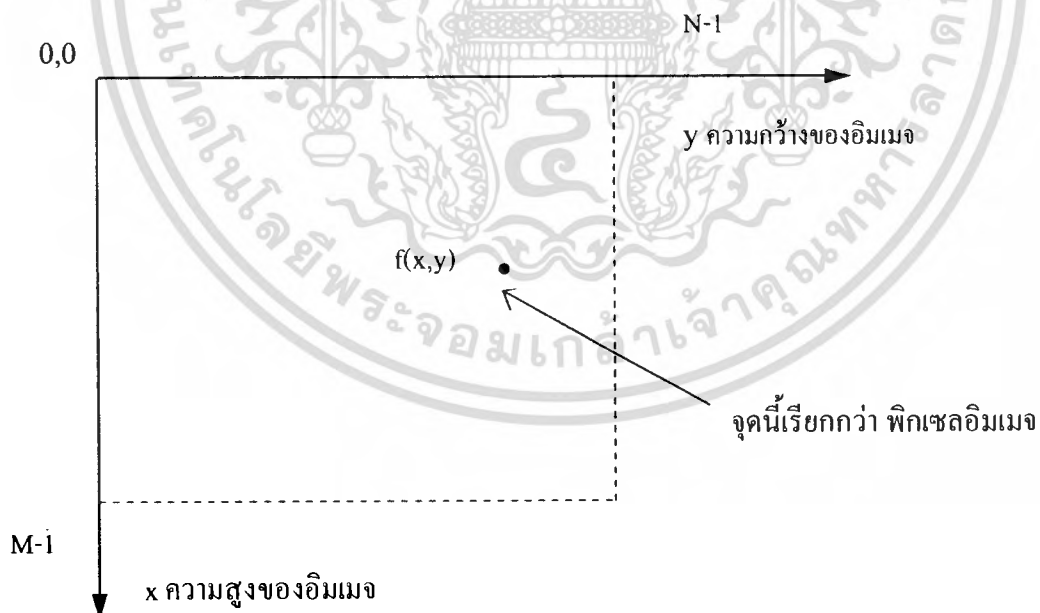
บทที่ 2

คำนิยามและความหมายของการทำกระบวนการอิมเมจดิจิทัล

ก่อนที่จะเข้าสู่การทำกระบวนการอิมเมจดิจิทัล เราจะเริ่มต้นที่อิมเมจก่อน โดยปกติแล้วสายตาของบุคคลทั่วไปจะมองเห็นภาพทิวทัศน์วิวต่าง ๆ เป็นลักษณะแบบอนาล็อก ซึ่งสามารถอธิบายได้ด้วยคณิตศาสตร์ที่มีตัวแปรแบบนับได้อย่างต่อเนื่อง แต่เครื่องคอมพิวเตอร์จะใช้เลขฐานสองเป็นหลักในการคำนวณ เมื่อนำภาพอิมเมจมาแปลงเข้าสู่เครื่องคอมพิวเตอร์ อิมเมจนั้นก็จะเป็น อิมเมจดิจิทัล (Digital Image)

2.1 อิมเมจดิจิทัล

อิมเมจดิจิทัลเป็นผลมาจากการสุ่มค่าในระบบพิกัด Space หรือ (Spatial Coordinate) ดังรูปที่ 2.1 และการทำ Quantization ของค่าระดับความสว่าง (Brightness Value) หรือความเข้ม (Intensity) ระบบพิกัด Space นี้จะใช้กับการแสดงอิมเมจดิจิทัล ซึ่งจะมีขนาดความกว้างและความสูงของอิมเมจแสดงในแกน Y และ X ตามลำดับ ส่วนจุดใด ๆ ที่วางบนระนาบ XY จะเป็นฟังก์ชัน $f(x,y)$ และเรียกว่า พิกเซล (Pixel) ที่แสดงถึงค่าระดับความเข้ม ซึ่งจะเป็นจำนวนที่นับได้จำกัด (Finite Number) แบบไม่ต่อเนื่อง หรือเรียกว่า Discrete Quantity ค่า Discrete Quantity เป็นผลมาจากการทำ Quantization โดยจะทำการแปลงจากอนาล็อก (Analog) เป็นดิจิทัล (Digital)



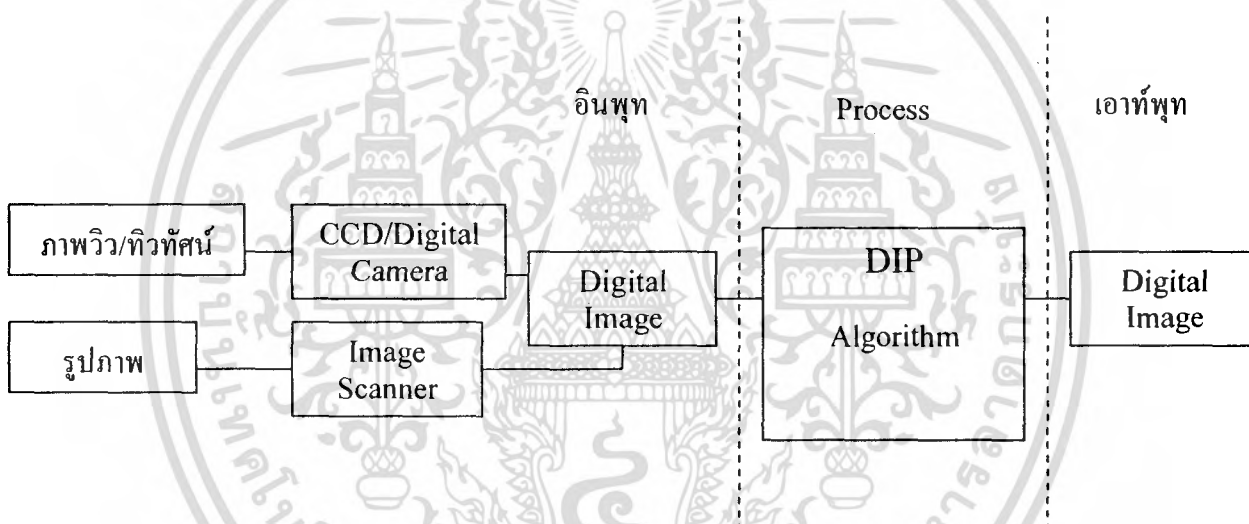
รูปที่ 2.1 ระบบพิกัด Space

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การทำกระบวนการอิมเมจดิจิทัล

มีสองเหตุผลใหญ่ ๆ ที่ต้องการทำกระบวนการอิมเมจดิจิทัล คือเพื่อปรับปรุงอิมเมจดิจิทัลให้มองเห็นได้ง่ายขึ้น และเพื่อปรับปรุงอิมเมจให้หุ่นยนต์ตีความหมาย หรือเข้าใจจดจำรูปร่างลักษณะได้อย่างแม่นยำ ตัวอย่างเช่น การจดจำอักษรหรือ Optical Character Recognition (OCR) ที่สามารถจดจำตัวอักษรได้ถึง 99.9 % การปรับปรุงอิมเมจให้ใช้พื้นที่เก็บน้อยลง การตรวจสอบลายพิมพ์มือของแต่ละบุคคล เป็นต้น

การทำงานกระบวนการอิมเมจดิจิทัล คือการนำอิมเมจดิจิทัลเข้ามาทำกระบวนการ โดยการใส่ฟังก์ชันอัลกอริทึมต่าง ๆ เข้าไป ก็จะได้เอาต์พุตเป็นอิมเมจดิจิทัล ที่ตรงตามแนวความคิดของการทำกระบวนการอิมเมจดิจิทัล เราสามารถแสดงภาพตั้งแต่การนำภาพวิว ทิวทัศน์ จนถึงเอาต์พุต ดังรูปที่ 2.2



รูปที่ 2.2 การทำกระบวนการอิมเมจดิจิทัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 วิธีการอ่านข้อมูลพิกเซลของอิมเมจ

จากรูปที่ 1 จุดที่วางอยู่ในพิกัด Space นี้ก็คือ พิกเซล (Pixel) หรือ Picture Element ซึ่งก็คือความสว่างหรือค่า Luminance (L) ของอิมเมจ ถ้าอิมเมจนั้นเป็นภาพขาวดำ ขนาด 8 บิต จะมีค่า L เท่ากับ 2^8 หรือเท่ากับ 256 ระดับ คือตั้งแต่ ระดับ 0 (พิกเซลเป็นสีดำ) จนถึง 255 (พิกเซลจะเป็นสีขาว) ($0 < L < 255$) บางครั้งค่าความสว่าง (L) อาจมีความหมายถึงระดับความละเอียดของอิมเมจ (Image Resolution)

ถ้าพิกเซลเป็นภาพขาวดำ จะอ่านค่าอิมเมจดิจิทัลในรูปแบบเมตริก 2 มิติ ขนาด $M \times N$ ได้ดังนี้

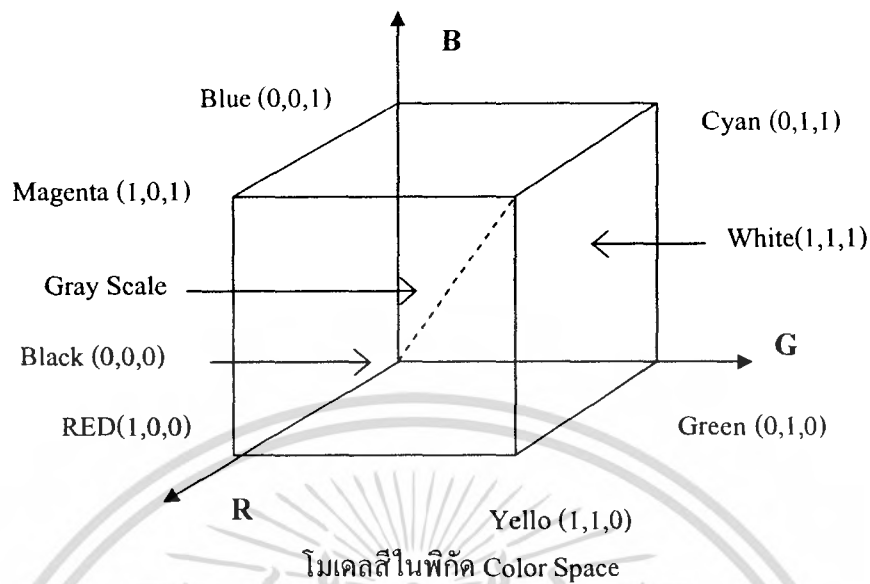
$$f(x,y) = \begin{pmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \dots & \dots & \dots & \dots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{pmatrix}_{M \times N} \quad \dots(1.)$$

โดยที่ ค่า $f(x,y)$ จะอยู่ในช่วง 0 ถึง 255 ($0 < f(x,y) < 255$)

สมมติว่าอ่านค่าพิกเซลจากอิมเมจหนึ่งได้ $f(x,y)$ เท่ากับ 10 แสดงว่า จุดพิกเซลนั้นมีความสว่างน้อยมากหรือค่อนข้างจะดำ ถ้าค่าที่อ่านได้ 250 แสดงว่า จุดพิกเซลนั้นมีความสว่างมาก

2.4 โมเดลสี (Color Model)

โมเดลสีหรือ Color Space ประกอบด้วย 3 แม่สีหลัก ได้แก่ สีแดง เขียว และน้ำเงิน ถ้านำแต่ละแม่สีมาพล็อตกราฟในระบบพิกัด Color Space โดยแต่ละสีมีค่า 0 ถึง 1 (0 แสดงถึงความมืด และ 1 แสดงถึงความสว่าง) จะได้ภาพการผสมสีทางแสงหรือการบวกแม่เข้าด้วยกัน (Additive Primary Color) ดังรูปที่ 2.3



รูปที่ 2.3 โมเดลสี

ถ้าแต่ละแม่สีเป็นขนาด 8 บิต รวมทั้งหมด เท่ากับ 24 บิต ซึ่งสามารถสร้างสีใหม่ได้ถึง $256 \times 256 \times 256$ เท่ากับ 16,777,216 สี ในที่นี่จะใช้พิกเซลอิมเมจที่มีแต่ละแม่สีเท่ากับ 8 บิต หรือเรียกว่า มีความลึกเท่ากับ 24 บิต เป็นหลัก

ถ้าพิกเซลเป็นภาพสีขนาด 24 บิต จะอ่านอิมเมจดิจิทัลในรูปแบบเมตริก 2 มิติ ขนาด $M \times N$ เหมือนกับในสมการ (1.) แต่ค่า $f(x,y)$ จะอยู่ในช่วงที่ประกอบด้วย

$$R \text{ ระดับ } 0 \text{ จนถึง } 255 \quad (0 < R < 255)$$

$$G \text{ ระดับ } 0 \text{ จนถึง } 255 \quad (0 < G < 255)$$

$$\text{และ } B \text{ ระดับ } 0 \text{ จนถึง } 255 \quad (0 < B < 255)$$

ในบางครั้งถ้าต้องการแปลงโมเดลสี ให้เป็นขาวดำ ซึ่งก็คือ Gray Scale จะใช้

$$\text{สมการ Gray Scale} = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \quad \dots (2.)$$

แต่เราสามารถใช้อีกสมการ โดยการหาค่าเฉลี่ยทั้งสามสีดังนี้

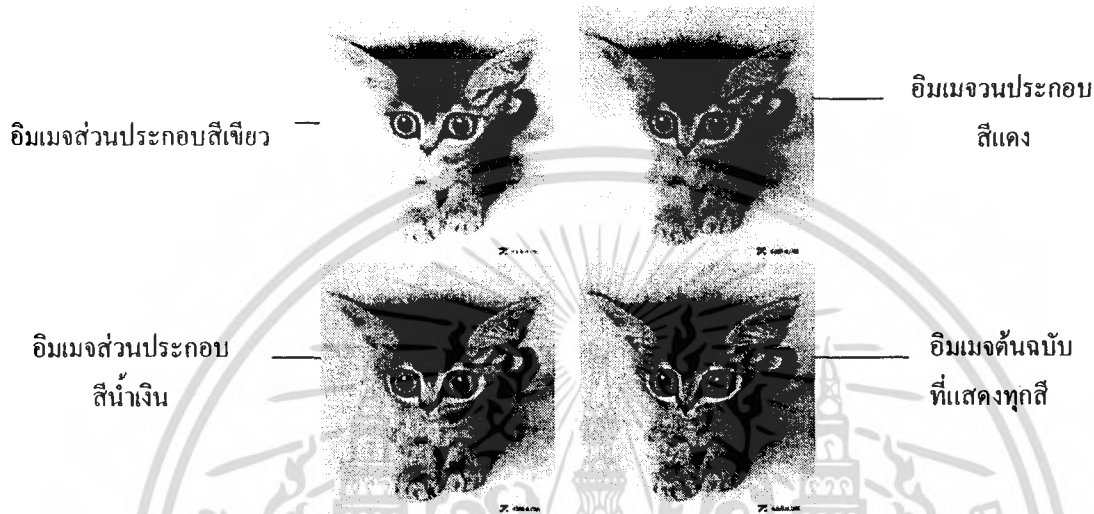
$$\text{Gray Scale} = \frac{R+G+B}{3} \quad \dots (3.)$$

3

จากรูปที่ 2.3 ค่า Gray Scale ก็คือค่าที่อยู่ในช่วง (0,0,0) จนถึง (1,1,1)

เพื่อที่จะดูเฉพาะอิมเมจส่วนประกอบสีแดง (Red Component Image หรือ Red Channel) อิมเมจส่วนประกอบสีเขียว อิมเมจส่วนประกอบสีน้ำเงิน และอิมเมจที่แสดงครบทุกสี เราจะใช้

โปรแกรม ColorImage ซึ่งได้อธิบายการสร้างไว้แล้ว ในที่นี้กำหนดชื่อไฟล์อิมเมจเป็น Tak.Bmp จากนั้นคอมไพล์โปรแกรมแล้วเรียกใช้งานด้วยการกดคีย์ Ctrl + Fs เราก็จะเห็น โปรแกรมโหลดภาพอิมเมจที่แสดงครบทุกสีในวินโดว์แรกต่อไปคลิกเมนู View > Red Image, View > Green Image และ View > Blue Image ก็จะแสดงภาพอิมเมจส่วนประกอบสีแดง เขียว และน้ำเงินตามลำดับ สุดท้ายคลิกเมนู Window > Tile เพื่อจัดวางแต่ละวินโดว์ให้แสดงอยู่ในเฟรมหลัก ดังรูปที่ 2.4



รูปที่ 2.4 อิมเมจต้นฉบับและอิมเมจส่วนประกอบสีต่าง ๆ

จากรูปที่ 2.4 นั้น อิมเมจส่วนประกอบสีน้ำเงิน จะมีความสว่างมากที่สุด หรือมีค่าระดับ Gray level ก่อนข้างสูง เมื่อเปรียบเทียบกับอิมเมจส่วนประกอบสีอื่น ๆ อิมเมจส่วนประกอบสีเขียวจะมีความมืดมากที่สุด หรืออิมเมจไฟล์ Tak.Bmp จะมีส่วนประกอบสีเขียวน้อยกว่าอิมเมจส่วนประกอบสีอื่น ๆ เพื่อที่จะเปรียบเทียบข้อมูลได้ถูกต้องมากกว่านี้ เราจะอาศัยการคำนวณทางสถิติ ซึ่งในที่นี้จะมีการคำนวณค่าความถี่ที่นับได้สูงสุด ค่าเฉลี่ยค่าเบี่ยงเบนมาตรฐาน และจำนวนพิกเซลทั้งหมด

2.5 การอ่านข้อมูลอิมเมจในหน่วยความจำ

ขนาดหน่วยความจำที่ต้องใช้ เมื่อคลาอ่านข้อมูลจากไฟล์อิมเมจชนิดต่าง ๆ ดังข้างต้นนี้ โดยใช้ฟังก์ชันของ ออบเจกต์ จะจองหน่วยความจำให้เท่ากับขนาดของอิมเมจ หรือเรียกว่า บล็อก หน่วยความจำขนาดของบล็อกหน่วยความจำอิมเมจนี้ จะคำนวณได้จากความกว้างคูณกับความสูงและคูณกับความลึกของอิมเมจโดยไม่สนใจว่าอินพุตอิมเมจเป็นชนิดใด เพราะเมื่ออ่านเข้าสู่หน่วยความจำจะถูกแปลงเป็นชนิดบิตแมพ (Bmp) เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการอ่านอิมเมจ เช่น อิมเมจชนิด Jpeg มีความกว้างและความสูงเท่ากับ 256 หน่วยและความลึกเท่ากับ 24 บิต เราจะต้องใช้หน่วยความจำในเครื่องคอมพิวเตอร์อย่างน้อยที่สุด เท่ากับ $256*256*24_{10} = 180 \text{ Kbyte}$ (1 ไบท์=8 บิต)

การหาพอยน์เตอร์ที่ชี้ไปยังจุดเริ่มต้นหรือเกือบท้ายสุด การหาพอยน์เตอร์ที่ชี้ไปยังจุดพักเริ่มต้นของบิตแมพ หรือเกือบท้ายสุดในบล็อกหน่วยความจำ จะใช้ฟังก์ชัน GetBits ร่วมกับฟังก์ชัน GetPitch และหาพอยน์เตอร์ที่ชี้ไปยังพิกเซลเฉพาะ จะใช้ฟังก์ชัน GetPixelAddress

ในการหาพอยน์เตอร์ จะเริ่มต้นด้วยการหา Pitch ของอิมเมจก่อน โดยใช้ฟังก์ชัน GetPitch ถ้าฟังก์ชันนี้คืนค่าเป็นบวก แสดงว่า การอ่านบิตแมพจะสแกนจากบนลงล่าง และเมื่อใช้ฟังก์ชัน GetBits จะคืนค่าพอยน์เตอร์ที่มุมบนซ้ายสุดของบล็อกหน่วยความจำอิมเมจ ซึ่งจะชี้ ณ จุดพิกเซลเริ่มต้นที่แถวบนสุด คอลัมน์แรก

ถ้าฟังก์ชัน GetPitch คืนค่าเป็นลบ แสดงว่า การอ่านค่าบิตแมพจะสแกนจากแถวล่างสุดขึ้นไปด้านบนเมื่อใช้ฟังก์ชัน GetBits จะคืนค่าพอยน์เตอร์ที่มุมล่างซ้ายสุดของบล็อกหน่วยความจำอิมเมจ ซึ่งจะชี้ ณ จุดพิกเซลเริ่มต้นที่แถวล่างสุด คอลัมน์แรก หรือเกือบสุดท้าย

2.6 การทำกระบวนการจุดพิกเซล (Point Processing)

การทำรูปแบบนี้จะเป็นแบบที่ง่ายที่สุดในการปรับปรุงคุณภาพอิมเมจให้ดีกว่าอิมเมจต้นฉบับ ในการกระทำกับจุดพิกเซลนั้นสามารถทำได้หลาย ๆ กรณีด้วยกัน ไม่ว่าจะเป็นการนำค่าคงที่ใด ๆ มาดำเนินการทางคณิตศาสตร์ เช่น การบวก ลบ คูณ หาร รวมทั้งดำเนินการด้วยลอจิก And, Or และหรือ Xor การแปลงที่เกี่ยวกับความสว่างเป็นต้น

ในที่นี้เราจะพิจารณาเฉพาะการแปลงที่เกี่ยวกับความสว่าง ได้แก่ การแปลงสีให้เป็นขาวดำ การกลับสี (Invert Color) และการแก้ไขพิกเซลด้วย Gamma การสร้างฮิสโทแกรม การทำ Histogram Equalized และการบวกลบอิมเมจ การใช้ลอจิก And, Or และหรือ Xor ดังรายละเอียดต่อไปนี้

2.6.1 การแปลงที่เกี่ยวกับความสว่าง (Luminance Transformation)

บางครั้งการแปลงนี้อาจจะเรียกว่า การแปลงความเข้มขึ้น (Intensity Transformation) โดยใช้สมการ $1/3(R + G + B)$ หรือ $0.299R + 0.587G + 0.114B$ ค่าที่ได้จากการแปลงจะอยู่ในระดับความสว่าง ถ้าแสดงบนจอคอมพิวเตอร์จะเป็นสีขาว

วิธีการแปลง เริ่มต้นด้วยการหาพอยน์เตอร์ที่ชี้ไปยังพิกเซล ณ ตำแหน่งพิกัดที่ต้องแปลง โดยใช้ฟังก์ชัน GetPixelAddress จากนั้นจึงอ่านข้อมูลของแต่ละสี มาเข้าสมการข้างต้นนี้ เราก็จะได้พิกเซลใหม่ที่จะมาแทนที่พิกเซลเดิม

ในการใช้พอยน์เตอร์ที่ได้จากฟังก์ชัน GetPixelAddress จะชี้ไปยังสีน้ำเงินก่อน ตามด้วยสีเขียว และสีแดง (ไม่ใช่ สีแดง เขียว และน้ำเงิน)

ถ้าต้องการแปลงพิกเซลสี ให้เป็นขาวดำตลอดทั้งอิมเมจ เราก็สแกนทุกแถวทุกคอลัมน์ด้วยการใช้คำสั่งลูป for ซึ่งสามารถเขียนโปรซีเจอร์ได้ดังนี้

2.6.2 โปรซีเจอร์แปลงพิกเซลสีเป็นขาวดำ

BYTE avg, r, g, b;

for (y = 0; y < maxHeight ; y++) begin

for (x = 0; x < maxWidth ; x++) begin

pSrc = GetPixelAddress (x,y) ;

r = pSrc [2] ; g = pSrc [1] ; b = pSrc [0] ;

avg = 0.299*r + 0.587*g + 0.114*b;

*pSrc++ = avg;

*pSrc++ = avg;

*pSrc = avg;

end

end

2.6.3 การกลับสี (Invert Color)

เราจะพบได้บ่อยในฟิล์ม Negative ซึ่งจะกลับสีจากสว่างเป็นสีมืดแทน โดยปกติแต่ละสีจะมีค่าตั้งแต่ 0 ถึง 255 ถ้าเราเขียนการแปลงการกลับสีในรูปสมการจะได้ดังนี้

$$R_s = 256 - 1 - R_r$$

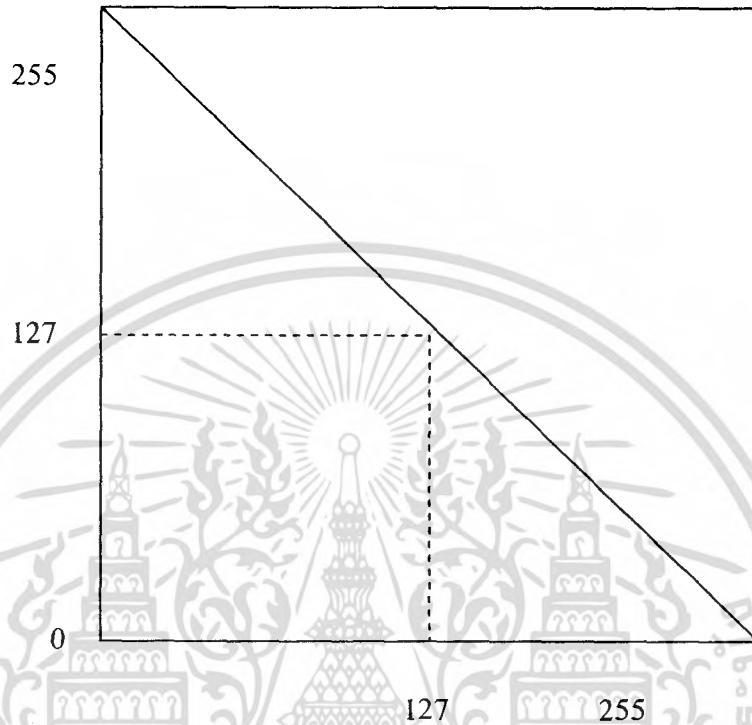
$$G_s = 256 - 1 - G_r \quad \dots(4.)$$

$$B_s = 256 - 1 - B_r$$

โดยที่ s คือ เอชท์พูด และ r คืออินพุต

จากสมการที่ 4 จะได้ความสัมพันธ์แสดงได้ดังรูปที่ 2.5

เอาต์พุต (R_s, G_s, B_s)



รูปที่ 2.5 การแปลงแบบ Negative

จากรูปที่ 2.5 ถ้าอินพุตเป็น 0 เอาต์พุตจะเป็น 255 และอินพุตเป็น 255 เอาต์พุตจะเป็น 0 แทนประโยชน์ในการใช้การกลับสีนั้น จะเหมาะกับการดูบริเวณที่มีมืด ๆ ของอิมเมจ ซึ่งเมื่อกลับสีแล้ว จะกลายเป็นบริเวณสว่างแทน

2.6.4 การแก้ไขพิกเซลด้วย Gamma

การปรับค่าพิกเซลของอิมเมจโดยการชดเชยด้วยแกมมา (Gamma) หรือเรียกว่า Gamma Correction Transformation ที่ชดเชยด้วยแกมมาก็เพราะอุปกรณ์ที่เกี่ยวข้องกับการแสดงผลอิมเมจ ตัวอย่างเช่น มอนิเตอร์ที่ประกอบด้วยอุปกรณ์หลอดภาพ Cathode Ray Tube (CRT) ทำให้ทำงานไม่ลิเนียร์ (Non Linear) จะส่งผลให้ภาพอิมเมจแสดงบนจอมอนิเตอร์ไม่ลิเนียร์ไปด้วย เมื่อแก้ไขหรือปรับค่าด้วยแกมมาแล้ว ภาพอิมเมจก็จะแสดงผลลิเนียร์ วิธีการแก้ไขคือ ใช้สมการแกมมาดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$R_s = R_r^{1/\gamma}$$

$$G_s = G_r^{1/\gamma}$$

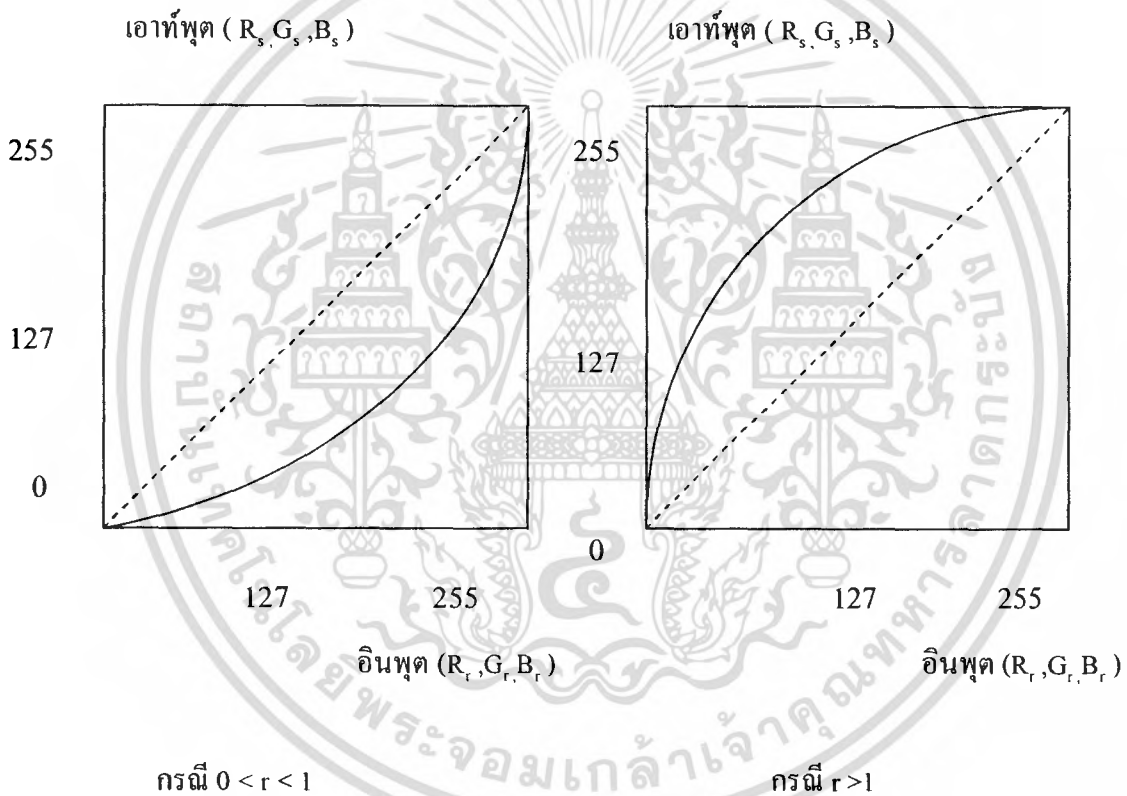
$$B_s = B_r^{1/\gamma}$$

โดยที่ s คือเอาต์พุต r คืออินพุต

$0 < \gamma < 1$ จะเป็นการแปลงแบบเส้นโค้งเอกซ์โปเนนเชียล

$\gamma = 1$ จะเหมือนกับอินพุต (ไม่มีการแปลง)

$\gamma > 1$ จะเป็นการแปลงแบบเส้นโค้งลอการิทึม แสดงดังรูปที่ 2.6



รูปที่ 2.6 Gamma Correction Transformation

จากรูปที่ 2.6 จะเห็นว่าถ้า $0 < \gamma < 1$ ภาพอิมเมจใหม่ จะมีความมืดมากกว่าอิมเมจต้นฉบับ

ขณะที่

$\gamma > 1$ ภาพอิมเมจจะมีความสว่างมากกว่าปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการแปลง เริ่มต้นด้วยการหาพอยน์เตอร์ที่ชี้ไปยังพิกเซล ณ ตำแหน่งพิกัดที่ต้องการแปลง โดยใช้ฟังก์ชัน GetPixelAddress จากนั้นจึงอ่านข้อมูลของแต่ละสี มาคำนวณค่าใหม่โดยใช้สมการที่

(5.)

ถ้าต้องการแปลงพิกเซลทั้งหมด เราก็สแกนตลอดทั้งอิมเมจด้วยการใช้คำสั่งรูป for ซึ่งสามารถเขียนเป็นโปรแกรมได้ดังนี้

2.6.5 โปรแกรมเมอร์ Gamma Correction Transformation

// อินพุต G

// double gamma = 1/G

for (y = 0; y < maxHeight; y++) begin

for (x = 0; x < maxWidth; x++) begin

pSrc = GetPixelAddress (x,y) ;

rGamma = pow (pSrc [2] / 255 , gamma) * 255 ;

gGamma = pow (pSrc [1] / 255 , gamma) * 255 ;

bGamma = pow (pSrc [0] / 255 , gamma) * 255 ;

*pSrc++ = bGamma;

*pSrc++ = gGamma;

*pSrc = rGamma;

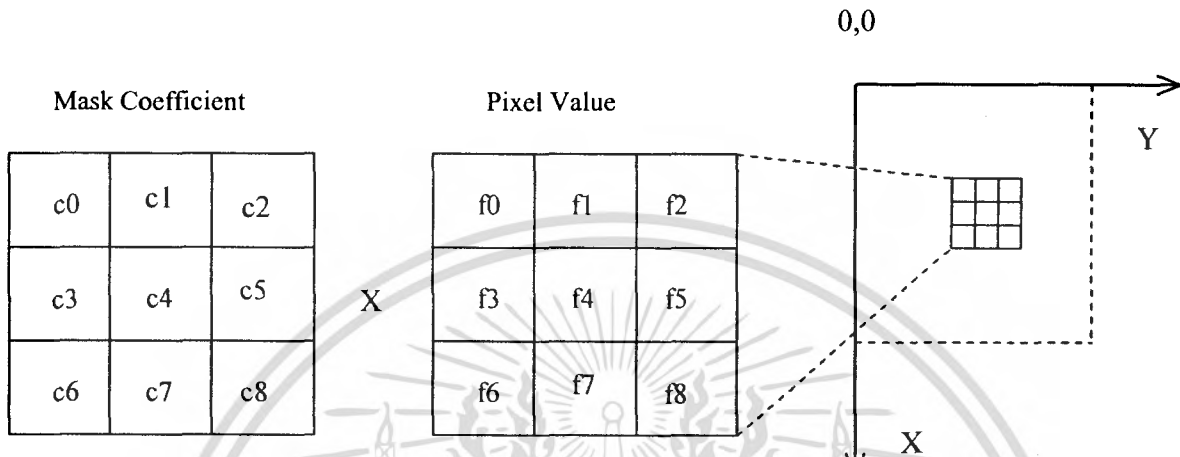
end

end

2.7 การทำกระบวนการรอบ ๆ จุดพิกเซล (Area processing)

การทำกระบวนการนี้จะเอาค่าบริเวณรอบ ๆ พิกเซลมาคำนวณ แล้วนำมาแทนค่าพิกเซลเดิม บริเวณรอบ ๆ พิกเซลนี้เรียกได้หลาย ๆ แบบ ตัวอย่างเช่น Filter Window, Kernel หรือ Mask ส่วนใหญ่มักจะนิยมเรียก Kernel หรือ Mask การคำนวณบริเวณรอบ ๆ พิกเซล อาจเรียกว่า การทำ Convolution

วิธีการทำ Convolution คือ การหาผลรวมที่ได้ถ่วงน้ำหนักแล้วบริเวณรอบ ๆ จุดพิกเซล ค่าถ่วงน้ำหนักนี้เรียกว่า Mask Coefficient หรือ Kernel Coefficient โดยปกติจะกำหนดค่าถ่วงน้ำหนักเป็นเมตริกขนาด $n \times n$ โดย n จะเป็นเลขคี่ เช่น 3, 5, 7, 9, ... เป็นต้น เพราะว่าสามารถหาค่าจุดพิกเซลตรงกลางได้ (ถ้าเป็น 1 จะเหมือนกับการทำกระบวนการจุดพิกเซล) ในที่นี่จะใช้เมตริกขนาด 3×3 เป็นหลัก เราสามารถแสดงวิธีการคำนวณและสมการได้ดังรูปที่ 2.7



รูปที่ 2.7 วิธีการคำนวณ Convolution

ค่าพิกเซลใหม่ $f(4)$ จะเท่ากับ $(C_0f_0 + C_1f_1 + C_2f_2 + C_3f_3 + C_4f_4 + C_5f_5 + C_6f_6 + C_7f_7 + C_8f_8)$

$$\text{หรือ } f(4) = \sum_{i=0}^8 C_i f_i \quad \dots(5.)$$

สมการที่ (5) เรียกว่า Linear filter ของอิมเมจ

ผลรวมค่าถ่วงน้ำหนัก (Mask Coefficient) แสดงได้ดังสมการ $\sum_{i=0}^8 C_i$
 ... (6.)

$i=0$

จากสมการที่ (6) ค่าผลรวมถ่วงน้ำหนักจะมีผลกระทบต่อระดับความสว่างตลอดทั้งอิมเมจ โดยทั่วไปแล้วผลรวมจะเป็นค่า 1 แต่ถ้าผลรวมเท่ากับหรือน้อยกว่า 0 อิมเมจจะมีความมืดมากขึ้น ซึ่งจะพบได้ในการตรวจหาขอบอิมเมจ หรือ Edge Detector

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ (5) ถ้านำมาหารด้วยสมการที่ (10.) เราจะใช้สมการฟิลเตอร์ (Filter) ในรูปแบบ
ทั่ว ๆ ไป ดังนี้

$$F'(4) = \sum_{i=0}^8 C_i f_i / \sum_{i=0}^8 C_i f_i \quad \dots(7.)$$

โดยที่สมการที่ (6) ต้องไม่เท่ากับ 0 หรือถ้าสมการที่ (6) = 0 จะเซตเป็น 1
โปรซีเจอร์ในการหาค่า Convolution สามารถเขียนได้ดังนี้

2.7.1 โปรซีเจอร์ Convolution

```
// อินพุต : mask [3] [3] และ offset
// หาค่าถ่วงน้ำหนัก โดยใช้สมการที่ (6)
double factor = 0.0
for (int j = 0 ; j < 3 ; j++) begin
    for (int i = 0 ; i < 3 ; i++) begin
        factor = factor + mask (j) [i]
    end
end
// คำนวณ โดยใช้สมการที่ (7)
int center = 1
for (y = 1 ; y < maxHeight - 1 ; y++) begin
    for (x = 1 ; x < maxWidth - 1 ; x++) begin
        redCenter = greenCenter = blueCenter = 0;
        for (int j = 0 ; j < 3 ; j++) begin
            for (int i = 1 ; i < 3 ; i++) begin
                p = อ่านค่าพิกเซลอิมเมจ ณ (x + i-center, y + j-center);
                redCenter      += p[2] * mask [j] [i] ;
                greenCenter    += p[1] * mask [j] [i] ;
                blueCenter     += p[0] * mask [j] [i] ;
            end
        end
    end
end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if factor = 0 then factor = 1;

red = redCenter / factor + offset ;           // new red
green = greenCenter / factor + offset ;       // new green
blue = blueCenter / factor + offset ;         // new blue

end

end

```

2.7.2 ค่า Mask Coefficient

สามารถกำหนดได้หลายรูปแบบ ในแต่ละรูปแบบก็จะมีผลต่อพิกเซลตลอดทั้งอิมเมจทำให้ค่า Mask Coefficient มีชื่อเรียกที่แตกต่างกัน ได้แก่ ฟิลเตอร์ Low Pass ฟิลเตอร์ High Pass การตรวจหาขอบอิมเมจ (Edge Detector) ซึ่งมีได้หลาย ๆ วิธีด้วยกัน และการทำอิมเมจให้นูน (Emboss) ดังรายละเอียดต่อไปนี้

2.8 การทำกระบวนการรูปร่างลักษณะ (Morphology Processing)

ในที่นี้จะอธิบายเฉพาะการทำกระบวนการรูปร่างลักษณะสำหรับอิมเมจดิจิทัลขาวดำหรือสี โดยส่วนใหญ่แล้วจะใช้กับอิมเมจขาวดำมากกว่า ส่วนการทำกระบวนการอิมเมจดิจิทัลไบนารี (Binary Image) มีเฉพาะสีขาวกับสีดำเท่านั้น ดูรายละเอียดในหนังสือที่ว่าด้วยการทำอิมเมจไบนารี โดยเฉพาะ

การทำกระบวนการรูปร่างลักษณะจะคล้ายกับการทำ Convolution ซึ่งจะต้องใช้ Mask Coefficient แต่ในการทำนี้ จะเรียก Mask Coefficient ใหม่ว่า Structure Element (S.E) การทำกระบวนการรูปร่างลักษณะมีได้หลายแบบด้วยกัน ตัวอย่างเช่น การขยายพิกเซล (Dilation) การลดขนาดพิกเซล (Erosion) การทำอิมเมจในพื้นที่ว่างให้เปิดมากขึ้น (Opening) และการทำอิมเมจให้ปิดมาก จะขึ้นอยู่กับ S.E.

2.8.1 การขยายพิกเซล (Dilation)

การทำ Dilation คือ การขยายพิกเซลที่สว่างให้ใหญ่ขึ้น สำหรับอิมเมจขาวดำ หรือ Gray Scale Morphology แสดงสมการได้ดังต่อไปนี้

ให้ A เป็นค่าพิกเซล ณ x, y ใด ๆ และ H เป็น Structure Element ขนาด i*j ทั้ง A และ H เป็นจำนวนจริง หรือเขียนใหม่ได้ว่า A และ H ∈ R

$$A + H_{ij} = \max_{ij} \{A(x-i, y-j) + H(i,j)\} \quad \dots(8)$$

จากสมการที่ (8) เขียนเป็น โปรซีเจอร์ Dilation ได้ดังนี้

โปรซีเจอร์ Dilation

```

int center = (Kernel - 1) / 2 // โดยปกติ kernel = 3
for (int y = 1; y < maxHeight - 1 ; y++) begin
    for (int x = 1; x < maxWidth - 1 ; x++) begin
        max = 0 ;
        for (int j = 0 ; j < kernel ; j++) begin
            for (int i = 0; i < kernel ; j++) begin
                p = (BYTE*) GetPixelAddress (x+i-center, y+j -center) ;
                redValue = p [2] + Se [j] [i] ;
                greenValue = p [1] + Se [j] [i] ;
                blueValue = p [0] + Se [j] [i] ;
                red = max (red, redValue) ; // new red
                green = max (green, greenValue); // new green
                blue = max (blue, blueValue); // new blue
            end
        end
    end
end
end

```

2.8.2 การลดขนาดพิกเซล (Erosion)

วิธีการนี้จะตรงข้ามกับวิธีการ Dilation คือลดขนาดพิกเซลที่สว่างลง ฉะนั้นจากสมการที่ (8.) จะเขียนใหม่ได้ดังดังนี้

$$A - H_{ij} = \min_{ij} \{A(x - i, y - j) - H(i, j)\}$$

ส่วนโปรซีเจอร์การคำนวณจะเหมือนกับโปรซีเจอร์ของการทำ Dilation เพียงแต่เปลี่ยนจากค่า max เป็นค่า min แทน

บทที่ 3

.NET Framework และ Visual C#**3.1 .NET Framework**

.NET Framework เป็นหนึ่งในเครื่องมือของการพัฒนาเพื่อสร้างแอปพลิเคชันทั่วไป และแอปพลิเคชันบนเว็บหรือ Web Services เป็นเหมือนตัวจัดการสภาพแวดล้อมในการทำงาน ทำให้เกิดความปลอดภัยและจัดการให้แอปพลิเคชันสร้างขึ้นสนับสนุนมาตรฐาน XML อย่างสมบูรณ์ ซึ่งเทคโนโลยีหลักใน .NET Framework ได้แก่ Common Language Runtime, คลาสไลบรารี และ ASP.NET

3.1.1. Common Language Runtime

Common Language Runtime ได้รวมเทคโนโลยีที่ทำให้แอปพลิเคชันมีความน่าเชื่อถือสูงขึ้น เช่น การกำจัดเรื่องของการสูญเสียหน่วยความจำ เป็นส่วนที่ทำหน้าที่รันโปรแกรมต่าง ๆ ที่สร้างขึ้นบน .NET ทำให้คอมพิวเตอร์และ Web Services ที่สร้างขึ้นเข้ากันได้อย่างสมบูรณ์ โดยไม่คำนึงถึงภาษาที่ใช้ในการเขียนโปรแกรม เพราะความจริงแล้วการสร้างแอปพลิเคชัน .NET นั้นไม่ใช่สามารถสร้างได้จากเฉพาะภาษาหลักใน Visual Studio .NET เท่านั้น ยังสามารถสร้างได้จากภาษาอื่น ๆ อีกด้วย เช่น Jscript, Perl หรือแม้กระทั่ง COBOL ก็ตาม

3.1.2. คลาสไลบรารี (Class Library)

เป็นคลาสที่รวมเข้าด้วยกัน ทำให้เกิดรูปแบบที่แน่นอนในการเข้าถึงฟังก์ชันต่าง ๆ ของแพลตฟอร์มทำให้ช่วยลดความจำเป็นในการเรียนรู้และควบคุมสถาปัตยกรรม API ที่แตกต่างกัน สำหรับการเขียนแอปพลิเคชัน

3.1.3. ASP.NET

ASP.NET เป็นเวอร์ชันใหม่ของ ASP (Active Server Pages) ที่ได้รับการปรับปรุงให้ทำงานบนแพลตฟอร์ม .NET ได้อย่างมีประสิทธิภาพมากขึ้น และมีความสามารถต่าง ๆ เพิ่มขึ้นอย่างมาก เราสามารถเรียกใช้ ASP.NET ซึ่งเป็นเหมือนโมเดลเว็บแอปพลิเคชัน ในรูปของชุดคอนโทรลและโครงสร้างพื้นฐานที่ง่ายต่อการสร้างแอปพลิเคชัน และ Web Services

72835

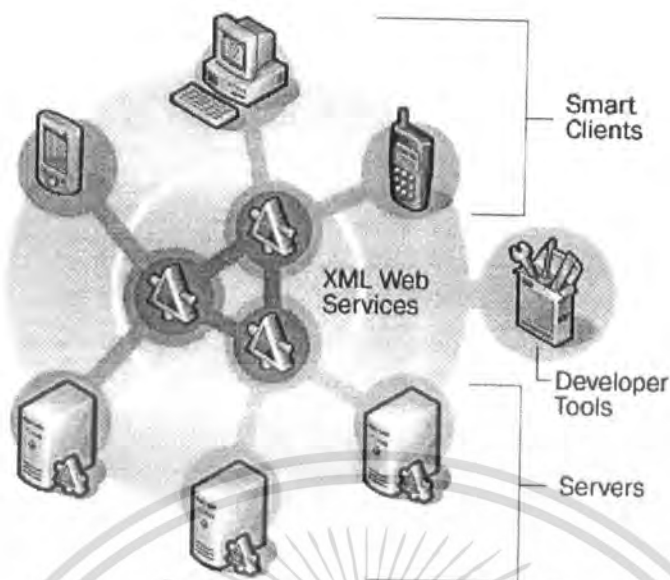
3.2 ที่มา .NET

บริษัท ไมโครซอฟท์ ได้คิดค้นเทคโนโลยีนี้ขึ้นมาเพื่อสร้างเป็น “โครงสร้างพื้นฐาน” (Infrastructure) เฉพาะของตนเองขึ้นมา ซึ่งจุดเด่นของ ไมโครซอฟท์ ความสะดวกได้แก่ ผลิตภัณฑ์ต่างๆ มากมายที่รองรับกับ ไมโครซอฟท์ ความสะดวกในการจัดตั้งระบบ และประสิทธิภาพ ที่ระบบมีให้ เทคโนโลยีนี้จะทำให้เกิดวัฒนธรรมในการดำรงชีวิตแบบใหม่ขึ้นมาอย่างแน่นอน โดยเฉพาะการใช้งานเทคโนโลยีสารสนเทศ และอินเทอร์เน็ตจะเป็นสิ่งที่ทุกคนสามารถเข้าถึงได้อย่างง่ายดาย, รวดเร็ว, เต็มไปด้วยข้อมูล และเข้าถึงได้จากที่ใดก็ได้ทั่วโลก

แกนหลักของ .NET คือแนวคิดในการทำให้เว็บไซต์และโปรแกรมต่าง ๆ ที่ทำงานบนเครื่องพีซี และอุปกรณ์ต่าง ๆ ที่อาจทำงานแยกจากกันในอดีตสามารถสื่อสารและแลกเปลี่ยนข้อมูลระหว่างกันได้ โดยจะสร้างซอฟต์แวร์ที่เป็นบริการ หรือ Web Services ไว้ในผลิตภัณฑ์ต่าง ๆ ของ ไมโครซอฟท์ ตั้งแต่ Windows ที่ใช้กับเครื่องพีซี ไปจนถึง Windows ที่ใช้กับเครื่องเซิร์ฟเวอร์ และนำไปผนวกกับเว็บไซต์ต่าง ๆ ด้วย ซึ่ง ไมโครซอฟท์ เล็งว่าในที่สุดผู้คนจะหันมาใช้ Web Services แทนแอปพลิเคชันต่าง ๆ แบบเดิมในเครื่องพีซี

3.3 องค์ประกอบของ .NET

องค์ประกอบของเทคโนโลยี .NET มีทั้งหมด 4 ส่วนด้วยกัน ซึ่งองค์ประกอบต่างๆ เหล่านี้จะทำงานประสานเชื่อมโยงกันอย่างเป็นระบบและสอดคล้อง จะขาดสิ่งใดสิ่งหนึ่งไม่ได้ และแต่ละองค์ประกอบก็สามารถใช้ผลิตภัณฑ์ที่แตกต่างกันได้โดยไม่จำกัด ทำให้ผู้พัฒนาระบบสามารถออกแบบให้ระบบงานของตนเองทำงานบน .NET ได้อย่างที่ตนเองต้องการในรูปแบบผสมผสาน (Mix and Match) นั่นเอง องค์ประกอบของ .NET มีดังนี้



รูปที่ 3.1 องค์ประกอบของ .NET

3.3.1 อุปกรณ์ลูกข่ายแสนฉลาด (Smart Clients)

ได้แก่เครื่องคอมพิวเตอร์ส่วนบุคคลขนาดต่างๆ เช่น เครื่อง PC, Palmtop (หรือ Notebook), PDA (Personal Digital Assistant), Set top box (อุปกรณ์ที่วางอยู่บนโทรทัศน์หรือเครื่องไฟฟ้าต่างๆ), มือถือ หรืออุปกรณ์เคลื่อนที่แบบอื่นๆ ที่ใช้ความสามารถของไมโคร โปรเซสเซอร์ตระกูล intel x86 ซึ่งฝัง (embedded) อยู่ในอุปกรณ์ต่างๆ เหล่านี้นอกจากนั้นอุปกรณ์ต่างๆ เหล่านี้ก็จะมียอซเฟิร์มแวร์ ระบบปฏิบัติการของไมโครซอฟท์ติดตั้งอยู่ภายในด้วย เช่น Windows95, 98, NT, ME, XP เป็นต้น โดยที่อุปกรณ์บางตัวอาจจะมีเพียงบางส่วนของระบบปฏิบัติการเหล่านี้ก็ได้

ลูกข่ายแสนฉลาดเหล่านี้จะต้องมีส่วนหนึ่งที่สำคัญก็คือ “การเชื่อมต่อและสื่อสาร” ซึ่งมีหลักการก็คือ สามารถเข้าถึงได้อย่างง่ายดายและทุกที่ สำหรับเครือข่ายหลักที่จะใช้เป็นเส้นทางในการสื่อสารของข้อมูลก็ได้แก่ “เครือข่ายอินเทอร์เน็ต” (หรืออินทราเน็ต หากใช้เฉพาะในองค์กร) ด้วยเหตุของการโยงโย่ที่กว้างขวางและการพัฒนาอย่างก้าวกระโดด เครือข่ายอินเทอร์เน็ตจึงกลายเป็นเส้นทางหลักที่อุปกรณ์ลูกข่ายเหล่านี้จะใช้ในการติดต่อสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 รูปตัวอย่างอุปกรณ์ บางส่วนที่มีการใช้งานเพื่อเชื่อมต่อเข้ากับเครือข่ายในยุคของ .NET

3.3.2 บริการบนเว็บแบบ XML (XML Web Services)

บริการเหล่านี้จะทำให้เกิดการพัฒนาทางการใช้ข้อมูลและสารสนเทศอย่างมากและจะหลายข้อจำกัดต่าง ๆ ในอดีต ที่เกิดจากการใช้ CGI บนเครื่องเว็บเซิร์ฟเวอร์ (Web Server) ที่เราสามารถส่งข้อมูลในรูปแบบของ HTML เท่านั้น แต่ด้วย Web Services เราสามารถส่งข้อมูลได้ทุกรูปแบบ ด้วยการเรียกใช้ในรูปแบบของการเรียกฟังก์ชัน โดยทั่วไป ผ่านการส่งข้อมูลในรูปแบบของ XML (ดูรายละเอียดของ XML ได้ในภาคผนวก) ซึ่งจะทำการพัฒนาแอปพลิเคชันในแบบกระจาย (Distributed Processing) ทำได้ง่ายขึ้น

ในปัจจุบันก็มีบริการ Web Services ที่ให้บริการฟรี ที่เราสามารถนำมาใช้กับแอปพลิเคชันหรือเว็บไซต์ของเราได้ ซึ่งจะทำการประหยัดทรัพยากรและเพิ่มความน่าเชื่อถือให้กับเว็บไซต์ของเรา ยกตัวอย่างเช่น หากเราต้องการจะทำการคำนวณการพยากรณ์อากาศด้วยตนเอง สิ่งนี้ก็อาจจะเป็นไปได้ยาก เพราะข้อมต้องการเครื่องคอมพิวเตอร์ที่มีพลังในการประมวลผลสูง ดังนั้นเราอาจจะไปใช้บริการจากไซต์ที่มี Web Services ประเภทนี้ให้เราได้ก็จะทำให้ข้อจำกัดที่เคยมีนั้นลดลงไปได้

3.3.3 แม่ข่ายหรือเครื่องให้บริการ (Servers)

สำหรับการให้บริการ XML Web Services นั้น ต้องการเครื่องที่เป็นแม่ข่ายหรือเซิร์ฟเวอร์ (Server) ที่จะมารองรับการทำงาน เพราะว่า Web Services นั้นจะส่งข้อมูลในรูปแบบของการอธิบายการทำงานของตนเอง (Service Description) ให้กับผู้ที่เรียกใช้ แต่การประมวลผลของ Web Services นั้นจะเกิดขึ้นที่เครื่องเซิร์ฟเวอร์ และสุดท้ายก็จะส่งผลลัพธ์กลับไปให้ผู้เรียกใช้ โดยที่เครื่องนี้จะต้องเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต อีกทั้งยังต้องมีทรัพยากร และพลังการประมวลผลที่เพียงพอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

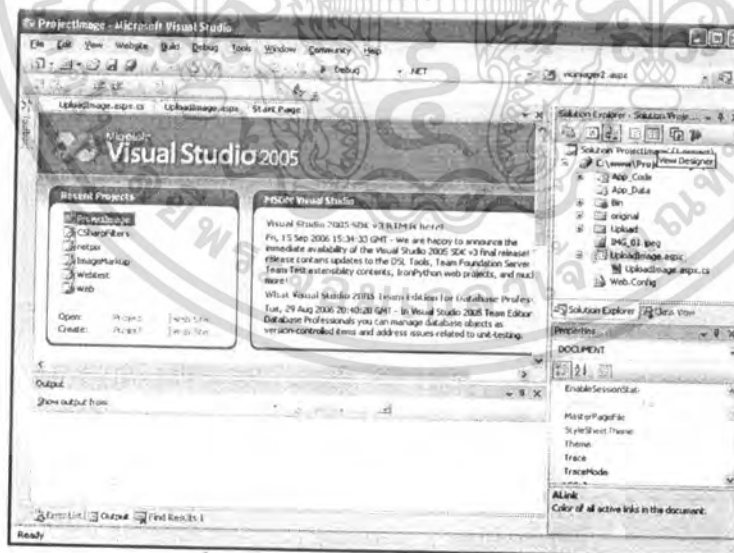


รูปที่ 3.3 เครื่องให้บริการ Servers

3.3.4 เครื่องมือของนักพัฒนา (Developer Tools)

ส่วนสุดท้ายนี้ก็เป็นส่วนที่สำคัญ เพราะคงไม่มีใครสร้างแอปพลิเคชันประเภท .NET ขึ้นมาได้ด้วยตนเอง หากไม่มีเครื่องมือช่วยหรือ Compiler สำหรับเครื่องมือที่ใช้พัฒนาแอปพลิเคชันให้ทำงานได้บน .NET นั้น ได้แก่ Visual Studio .NET ซึ่งหากเราใช้เครื่องมือนี้แล้ว งานต่าง ๆ ก็จะลดความยุ่งยากลงไปมาก

ที่กล่าวมาทั้งหมดนั้นเป็นส่วนของเทคโนโลยี .NET โดยในหนังสือเล่มนี้ จะใช้เป็นแกนหลักในการศึกษาถึงการสร้างแอปพลิเคชันประเภทต่างๆ



รูปที่ 3.4 โปรแกรม Visual Studio .NET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 Microsoft Visual C#.net

ภาษา C# ถูกสร้างขึ้นมาเพื่อที่จะใช้ความสามารถของการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming) และประสิทธิภาพในการทำงานของโปรแกรมที่พัฒนาด้วยภาษา C หรือ C++ อีกทั้งยังรวมเอาจุดเด่นของภาษา Visual Basic คือการใช้งานง่ายรวดเร็วในการพัฒนาเข้าไว้ด้วย นอกจากนี้แล้วภาษา C# ยังเพิ่มเทคโนโลยีอื่น ๆ เข้ามาใหม่อีกอย่างมากมาย ยกตัวอย่างเช่น

การรองรับเทคโนโลยี Microsoft.NET

เทคโนโลยีของ .NET Framework

เทคโนโลยีการพัฒนาแอปพลิเคชันแบบใหม่ ได้แก่ Component-Oriented

ความเข้ากันได้กับระบบเก่า ได้แก่ COM, Win32 API เป็นต้น

การใช้งาน IDE ที่เต็มไปด้วยประสิทธิภาพ IntelliSense ที่ช่วยเพิ่มคำในโค้ดคำสั่งให้สมบูรณ์เมื่อเราพิมพ์แค่เพียงอักษรบางตัว, เครื่องมือ Toolbox ที่สามารถลากคอนโทรลต่าง ๆ มาวางบนหน้าต่าง ออกแบบได้ทันที หน้าต่าง Task List กำหนดลำดับการทำงานของโปรแกรม

การพัฒนาแอปพลิเคชันเพื่อทำงานบนอุปกรณ์เคลื่อนที่ เช่น Pocket PC, Palm หรือโทรศัพท์มือถือ

3.4.1 ไวยากรณ์ C#

ความคล้ายคลึงกัน ของ Code ของ C# กับ C++ และ Java นั้นอาจดูค่อนข้างสับสน และมันก็คล้ายกับการเขียนภาษาอังกฤษ น้อยกว่าภาษาอื่น ๆ มาก แต่อย่างไรก็ตาม เราจะพบว่า ตัวเราเองอยู่ในโลกของการโปรแกรม C# ที่มี Style อ่อนไหวนุ่มนวล และสามารถอ่าน Code ได้ง่ายและไม่สับสนมากนัก

C# นั้นไม่มีการแจ้งเตือนเกี่ยวกับช่องว่างที่อยู่ใน Code ไม่ว่าคุณจะใช้ช่องว่างจำนวนมาก อักขระ Carriage return หรือ Tab เป็นอักขระที่รู้จักกันในชื่อว่า White space นั้นหมายความว่าเรามีอิสระในการที่จะจัดรูปแบบ Source Code ของเราได้ ถึงแม้ว่าการทำตามกฎที่แน่นอน สามารถช่วยให้เราทำสิ่งต่างๆ ให้อ่านได้ง่ายขึ้นก็ตาม Code C# นั้นสร้างขึ้นจาก Statement ชุดหนึ่ง แต่ละ Statement จะจบด้วย ; เนื่องจาก White space นั้นถูกมองข้ามไป เราจึงสามารถมีหลายๆ Statement ในบรรทัดเดียวกันได้ แต่เพื่อความง่ายในการอ่าน มันจะมีประโยชน์ ถ้าเพิ่ม Carriage return เข้าไปหลัง ; ดังนั้นเราก็จะไม่มีหลายๆ Statement ในบรรทัดเดียวกัน อย่างไรก็ตาม มันเป็นสิ่งที่ยอมรับได้และเป็นสิ่งปกติ

C# เป็นภาษา โครงสร้างบล็อก (Block structured) หมายความว่า ทุกๆ Statement ถือเป็นส่วนหนึ่งของ Block Code และ Block เหล่านี้ถูกจำกัดด้วย เครื่องหมาย { } เหมือนกับ C++ ทุกประการ รูปแบบจึงเหมือนกับ C++ และมีโครงสร้างเหมือน C++ ดังนั้นหากเราได้ศึกษาโครงสร้าง ของ C++ แล้ว ก็จะเข้าใจ หากยังไม่เข้าใจให้ไปศึกษาโครงสร้างใน C++

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 ตัวแปร

สำหรับตัวแปรใน C# จะคล้ายกับ C++ มาก ดังนั้นสำหรับผู้ที่ได้ศึกษาเรื่องของ C++ มาแล้วก็จะเป็นไปได้ง่าย ในการศึกษาเรื่องนี้ ในการที่จะใช้ตัวแปรนั้น เราจะต้องประกาศตัวแปรก่อน นั่นคือจะต้องกำหนดชื่อ และชนิดของตัวแปรนั้น ทั้งนี้ที่เราได้ประกาศชนิดของตัวแปรเสร็จสิ้น เราจะใช้มันเป็นตัวจัดเก็บข้อมูล ตามชนิดที่เราได้ประกาศไว้ได้ ซึ่งมีรูปแบบการประกาศดังนี้ Type name

ถ้าเราพยายามใช้ตัวแปรที่ยังไม่มีการประกาศเอาไว้ก่อนแล้ว Code ของเราจะไม่ถูก Compile แต่ Compiler จะบอกถึงสาเหตุที่แท้จริงที่เกิดขึ้น ดังนั้นจึงไม่ใช่ข้อผิดพลาดที่ร้ายแรงเท่าใดนัก นอกจากนี้การพยายามที่จะใช้ตัวแปรที่ไม่ได้กำหนดค่าเริ่มต้น ให้กับมัน มักจะเกิดข้อผิดพลาด แต่ก็อีกนั่นแหละ ตัว Compiler จะตรวจสอบพบมันได้ ตัวแปรอย่างง่าย ๆ ที่จะแนะนำนี้ คือชนิดของตัวแปร เช่น ตัวเลข และ Boolean ที่ใช้เป็นรากฐานในการสร้าง Block สำหรับ Application ของเรา และสำหรับสิ่งอื่นๆ คือตัวแปรที่สลับซับซ้อน ตัวแปรอย่างง่าย ส่วนใหญ่แล้ว คือ ตัวเลข ซึ่งเมื่อเริ่มต้นนั้น มันดูเหมือนแปลกตาอยู่บ้าง แน่แน่นอนว่าจำเป็นใหม่ ที่เราต้องการแค่ชนิดเดียวเท่านั้นที่ใช้เก็บตัวเลข

เหตุผลที่ตัวแปรชนิดตัวเลข มีอยู่มากมายนั้น เป็นไปตามกลไกของการจัดเก็บตัวเลข ในลักษณะอนุกรม ของ 0 และ 1 ในหน่วยความจำของคอมพิวเตอร์ สำหรับค่าจำนวนเต็ม เราเพียงแค่นำจำนวน Bit แล้วแทนที่จำนวนของเราในแบบ Binary ตัวแปรที่จัดเก็บ N Bit จะอนุญาตให้เราจัดเก็บจำนวนใดๆ ระหว่าง 0 ถึง $2^n - 1$ ได้ ถ้าจำนวนใดๆ ที่มากกว่าค่านี้ จะใหญ่เกินไป ที่จะใส่ไปในตัวแปรได้ ถ้าเราต้องการให้สามารถเก็บจำนวนได้มากขึ้น เราต้องใช้ Bit มากขึ้นด้วย ดังนั้น เราจึงจำเป็นต้องกำหนดตัวแปรให้เหมาะสม กับขนาดของข้อมูลที่เราจะใช้ ดังตารางต่อไปนี้ ซึ่งเป็นตารางตัวแปรชนิดตัวเลขจำนวนเต็ม

ชนิด	มาจาก	ช่วงข้อมูล
sbyte	System.Sbyte	-128 ถึง 127
byte	System.Byte	0 ถึง 255
short	System.Int16	-32768 ถึง 32767
ushort	System.UInt16	0 ถึง 65535
int	System.Int32	-2147483648 ถึง 21474483647
uint	System.UInt32	0 ถึง 4294967295
long	System.Int64	-9223372036854775808 ถึง 9223372036854775807
ulong	System.UInt64	0 ถึง 18446744073709551615

ตัว u ที่อยู่ก่อนชื่อชนิดตัวแปร ย่อมาจาก "unsigned" หมายความว่า ไม่สามารถจัดเก็บจำนวนติดลบใน ตัวแปรชนิดนี้ได้ หรือพูดอีกอย่างว่า ตัวแปรแบบ ไม่คิดเครื่องหมาย นั่นเอง

นอกจาก ตัวแปรชนิดตัวเลขจำนวนเต็มแล้ว เรายังต้องการจัดเก็บแบบ ตัวเลขทศนิยม (Floating point) อีกด้วย ซึ่งเป็นอะไรก็ได้ที่ไม่ใช่ตัวเลขเสียทั้งหมด มีชนิดตัวแปรอยู่ 3 ชนิด สำหรับตัวเลข ทศนิยม ที่เราสามารถใช้ได้ ได้แก่ float, double และ decimal สองชนิดแรกในจำนวนนี้ จะจัดเก็บ ทศนิยม ในรูปของ $\pm m \times 2^e$ ซึ่งอนุญาติ ให้จัดเก็บค่าสำหรับ m และ e ต่างกันไปตามแต่ละชนิด ชนิด ตัวแปร decimal ใช้รูปแบบเพิ่มเข้ามาอีกอย่างหนึ่งคือ $\pm m \times 10^e$ ชนิดตัวแปรทั้ง 3 เหล่านี้ แสดงไว้ใน ตารางข้างล่าง พร้อมทั้งช่วงของข้อมูล ดังนี้

ชนิด	มาจาก	ช่วง m	ช่วง e	ช่วงข้อมูล
float	System.Single	0 ถึง 244	-149 ถึง 104	1.5×10^{-45} ถึง 3.4×10^{38}
double	System.Double	0 ถึง 253	-1075 ถึง 970	5.0×10^{-324} ถึง 1.7×10^{308}
decimal	System.Decimal	0 ถึง 296	-26 ถึง 0	1.0×10^{-28} ถึง 7.9×10^{28}

นอกจากนี้ยังมีชนิดตัวแปรอีก 3 ชนิด ดังตารางข้างล่าง

ชนิด	มาจาก	ค่าข้อมูล
char	System.Char	อักษร Unicode 0 ถึง 65535
bool	System.Boolean	ค่า Boolean คือ true & false
string	System.String	สายลำดับอักขระ

จะสังเกตเห็นว่า ไม่มีขีดจำกัดของจำนวนอักษร ที่เป็น String เนื่องจากมันสามารถใช้หน่วยความจำ ในจำนวนที่แปรเปลี่ยนได้

ชนิดตัวแปรแบบ bool เป็นหนึ่งในชนิดตัวแปร ที่ใช้กันโดยทั่วไปใน C# และจริงๆ แล้ว ชนิด ข้อมูลที่คล้ายกันนี้ ก็สืบทอดไปใน Code ของภาษาอื่นๆ อย่างเท่าเทียมกัน การที่มีตัวแปรที่สามารถ เป็น true หรือ false มีความสำคัญเชิงขยาย เมื่อมันถูกนำมาใช้กับ Flow ของ ตรรกะ ใน Application หนึ่งๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 การประกาศตัวแปร

การประกาศตัวแปรนั้น กระทำแบบเดียวกันกับ C++ เช่น

```
int a;
char b;
float c;
double d;
string e;
```

หลังจากที่ประกาศตัวแปร เราก็จะกำหนดค่าให้กับตัวแปร โดยใช้ Operator = ดังนี้ a = 1;

```
b = 'd';
c = 1.05f;
d = 1234.44f;
e = "สกลนคร";
```

จงจำไว้ว่า ใน C# นั้น ตัวแปรทั้งหลาย จะต้องกำหนดค่าเริ่มต้นให้กับมัน ก่อนที่เราจะใช้งานมัน

เช่น

```
int a;
char b;
float c;
double d;
string e;
a = 1;
b = 'd';
c = 1.05f;
d = 1234.44f;
e = "Notfound";
```

หรือ

```
int a = 1;
char b = 'd';
float c = 1.05f;
double d = 1234.44f;
string c = " Notfound " ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5 นิพจน์

C# นั้น บรรจุ Operator หรือตัวดำเนินการ จำนวนหนึ่งเพื่อวัตถุประสงค์นี้ การผสมผสาน Operator กับตัวแปร และค่าคงที่นั้น สิ่งเหล่านี้เรียกว่า Operand หรือตัวถูกดำเนินการ เราก็จะสามารถสร้างนิพจน์ หรือ Expression ซึ่งเป็น Block การสร้างการคำนวณพื้นฐานได้

Operator ที่มีให้ใช้ มีตั้งแต่ระดับ ง่าย จนถึงระดับที่ ซับซ้อน Operator อย่างง่ายประกอบไปด้วย การปฏิบัติการทาง คณิตศาสตร์ ชั้นพื้นฐาน ทั้งหมด เช่น บวก ลบ คูณ หาร เป็นต้น และ Operator ที่ ซับซ้อนนั้น ประกอบไปด้วย การจัดการเนื้อหาตัวแปร โดยการแทนที่เนื้อหาเหล่านี้ด้วย ค่าของ Binary เป็นต้น ยังมี Operator ทางตรรกะที่จะใช้ในการ ติดต่อกำประเภท Boolean โดยเฉพาะ Operator ทั้งหมดถูกแบ่งออกเป็น 3 กลุ่มดังนี้

Unary ทำงานกับ Operand เดียว Binary ทำงานกับ Operand 2 ตัว Ternary ทำงานกับ Operand 3 ตัว

Operator ส่วนใหญ่อยู่ในกลุ่ม Binary มี Operand แบบ Unary น้อยมาก และ Operator แบบ Ternary เพียงตัวเดียวที่เรียกว่า เงื่อนไข

3.4.6 Operator ทางคณิตศาสตร์

มี Operator ทางคณิตศาสตร์ แบบง่ายๆ อยู่ 5 ตัว ซึ่ง 6 ตัวอยู่ในกลุ่มของ Binary และ Unary ด้วย ดังในตารางข้างล่าง

Operator	กลุ่ม	ตัวอย่าง	ความหมาย
+	Binary	$a = b + c;$	a เท่ากับผลลัพธ์ของ b บวกกับ c
-	Binary	$a = b - c;$	a เท่ากับผลลัพธ์ของ b ลบกับ c
*	Binary	$a = b * c;$	a เท่ากับผลลัพธ์ของ b คูณกับ c
/	Binary	$a = b / c;$	a เท่ากับผลลัพธ์ของ b หารกับ c
%	Binary	$a = b \% c;$	a เท่ากับผลลัพธ์ของเศษของ b หารกับ c
+	Unary	$a = +b;$	a เท่ากับผลลัพธ์ของ b
-	Unary	$a = -b;$	a เท่ากับผลลัพธ์ของ b คูณกับ -1
++	Unary	$a = ++b;$	a เท่ากับผลลัพธ์ของ $b+1$ b ถูกเพิ่มค่าอีก 1
--	Unary	$a = --b;$	a เท่ากับผลลัพธ์ของ $b-1$ b ถูกลดค่าอีก 1
++	Unary	$a = b ++;$	a เท่ากับผลลัพธ์ของ b b ถูกเพิ่มค่าอีก 1
--	Unary	$a = b --;$	a เท่ากับผลลัพธ์ของ b b ถูกลดค่าอีก 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Operator กำหนดค่า

เมื่อเรารู้จัก Operator อย่างง่าย ไปแล้วคือ = แต่ก็ยังมี Operator กำหนดค่าอย่างอื่นอีก Operator กำหนดค่าทั้งหมดที่ว่ามี จะทำงานคล้ายกับ = ซึ่งจะให้ผลลัพธ์ จากค่าที่ถูกกำหนด ให้กับตัวแปรที่อยู่ทางซ้ายมือ โดยขึ้นอยู่กับ Operand และ Operator ที่อยู่ทางด้านขวามือ ดังตาราง

Operator	กลุ่ม	ตัวอย่าง	ความหมาย
=	Binary	a = b;	a เท่ากับผลลัพธ์ของ b
+=	Binary	a += b;	a เท่ากับผลลัพธ์ของ a บวก b
-=	Binary	a -= b;	a เท่ากับผลลัพธ์ของ a ลบ b
*=	Binary	a *= b;	a เท่ากับผลลัพธ์ของ a คูณ b
/=	Binary	a /= b;	a เท่ากับผลลัพธ์ของ aหาร b
%=	Binary	a %= b;	a เท่ากับผลลัพธ์ของเศษของ a หาร b

3.4.7 ความสำคัญของ Operator

เมื่อนิพจน์หนึ่งๆ ถูกประเมินค่า แต่ละ Operator ก็จะถูกประมวลตามลำดับของมัน อย่างไรก็ตาม นี่ไม่ได้หมายความว่า การประเมินค่า Operator เหล่านี้จะเริ่มจากซ้ายไปขวา แต่จะเริ่มจากความสำคัญของ Operator ดังตารางลำดับความสำคัญ

ลำดับความสำคัญ	Operator
1	++ (ใช้นำหน้า) เช่น ++a
2	-- (ใช้นำหน้า) เช่น --a
3	+ (ใช้นำหน้า) เช่น +a
4	- (ใช้นำหน้า) เช่น -a
5	* เช่น a * b
6	/ เช่น a / b
7	% เช่น a % b
8	+ เช่น a + b
9	- เช่น a - b
10	= เช่น a = b

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11	*= เช่น a *= b
12	/= เช่น a /= b
13	%= เช่น a %= b
14	+= เช่น a += b
15	-= เช่น a -= b
16	++ (ใช้ตามหลัง) เช่น a++
17	-- (ใช้ตามหลัง) เช่น a--

จากตารางความสำคัญ หาก มีเครื่องหมาย () จะทำใน () ก่อนเป็นอันดับแรก เช่น (a-b)+c*d จะทำ a-b ก่อน แล้วเก็บค่าไว้ ก่อน หลังจากนั้น จะทำ c*d แล้วจึงนำค่า ดังกล่าวมาบวกกัน เช่น

$$(5-3) + 4*2 = 10$$

$$(15-3)+5-4 = 3$$

$$4/2*5 = 0.4$$

3.4.8 Namespace

Namespace เป็นวิธีการของ .Net ในการจัดหา Container ให้กับ Code Application เนื่องจาก Code และเนื้อหา ของมันนั้นอาจจะต้องถูก บ่งชี้แบบยูนิค (Unique ไม่ซ้ำกัน) Namespace ยังถูกใช้ใน ความหมาย ของการจัดกลุ่ม Item ใน .Net Framework เข้าด้วยกัน Item เหล่านี้ส่วนใหญ่เป็น นิยามชนิด ข้อมูล

โดย Default แล้ว C# นั้น ถูกกำหนดเอาไว้ใน Global namespace นี้หมายความว่า Item ทั้งหมดที่ บรรจุเอาไว้ใน Code จะถูกเข้าถึงได้จาก Code อื่นๆ ที่อยู่ใน Global namespace โดยการอ้างถึงด้วยชื่อ ใดๆก็ตามเราสามารถ ใช้ Keyword namespace เพื่อกำหนด namespace สำหรับ Block ของ Code ให้ ชัดเจนได้ ชื่อทั้งหลายที่อยู่ใน namespace หนึ่งๆ นั้น ต้องมีคุณสมบัติ ถ้ามันถูกนำไปใช้จาก Code ที่อยู่ นอก namespace นี้

ชื่อที่มีคุณสมบัติ คือชื่อที่บรรจุข้อมูลแบบลำดับชั้นของมันทั้งหมด หมายความว่า ถ้าเรามี Code ใน namespace หนึ่งๆ ซึ่งต้องใช้ชื่อที่กำหนดเอาไว้ ใน namespace ที่แตกต่างกัน เราต้องใส่การอ้างถึง namespace ดังกล่าวด้วย ชื่อที่มีคุณสมบัติ นั้นใส่ตัวอักษร "." เอาไว้ระหว่างระดับของ namespace เช่น namespace MySpace

{

```
// Code ที่อยู่ใน MySpace
// ชื่อหรืออะไรก็ตาม ที่ประกาศภายในนี้
}
// Code ที่อยู่ใน Global namespace
```

ภายใน namespace สามารถกำหนด namespace ลดหลั่นกันได้ โดยการใช้ Keyword namespace ดังนี้

```
namespace MySpace
```

```
{
```

```
    // Code ที่อยู่ใน MySpace
    // ชื่อหรืออะไรก็ตาม ที่ประกาศภายในนี้
    int a;
    int b;
    namespace MySpace2
```

```
    {
// Code ที่อยู่ใน MySpace2
```

```
// ชื่อหรืออะไรก็ตาม ที่ประกาศภายในนี้
```

```
        int c;
        int d;
    }
```

```
}
```

```
// Code ที่อยู่ใน Global namespace
```

ในการอ้างอิง ก็ใช้วิธีการอ้างอิง จั๋วแปรหรือสมาชิก ต่างๆ ใช้วิธีการอ้างอิง ดังนี้

MySpace.a; เป็นการอ้างอิงตัวแปร a ที่อยู่ใน namespace MySpace

MySpace.b; เป็นการอ้างอิงตัวแปร b ที่อยู่ใน namespace MySpace

MySpace.MySpace2.c; เป็นการอ้างอิงตัวแปร c ที่อยู่ใน namespace MySpace2 ซึ่งอยู่ภายใต้ namespace MySpace

MySpace.MySpace2.d; เป็นการอ้างอิงตัวแปร d ที่อยู่ใน namespace MySpace2 ซึ่งอยู่ภายใต้ namespace MySpace

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ASP.NET เว็บแอปพลิเคชัน

4.1 ที่มาของ ASP

แอคทีฟเซิร์ฟเวอร์เพจ(ASP) ของบริษัทไมโครซอฟท์ คือเทคโนโลยีใหม่ในการสร้างเว็บเพจแบบไดนามิก หน้าของ ASP นั้น ประกอบด้วย HTML markup และสคริปต์ทางฝั่งเซิร์ฟเวอร์ ซึ่งจะประมวลผลตามอินพุตที่ได้รับเข้ามา ณ ขณะนั้น และสร้างโค้ด HTML ขึ้นมาจากการประมวลผล สคริปต์ทางฝั่งเซิร์ฟเวอร์จะทำงานเมื่อมีการร้องขอ (request) หน้า ASP ส่งมาถึงฝั่งเซิร์ฟเวอร์ อินพุตของหน้า ASP จะมาจากเบราว์เซอร์ทางฝั่งไคลเอนต์ผ่านทางเมธอด HTTP POST และ GET ASP มี อ็อบเจกต์โมเดลเพื่อความสะดวกขึ้นในการทำงานของผู้พัฒนา นอกจากการใช้อ็อบเจกต์จาก ASP อ็อบเจกต์โมเดล เช่น Application , Server , Request ,Respond และ Session แล้ว ผู้พัฒนายังสามารถใช้ COM คอมโพเนนต์ อื่นๆที่มีอยู่บนเซิร์ฟเวอร์ ได้

ในการพัฒนาแอปพลิเคชันโดยใช้ ASP มักจะมีความยุ่งยากระหว่าง HTML markup และ สคริปต์ทางฝั่งเซิร์ฟเวอร์ รูปแบบของหน้า ASP ทำให้เกิดความยุ่งยากในการจัดการและการนำมาใช้ใหม่ จึงมีการพัฒนาโดยเพิ่มไฟล์และฟังก์ชันในสคริปต์ทางฝั่งเซิร์ฟเวอร์ อย่างไรก็ตาม ความพยายามนี้ทำให้เกิดข้อเสียในด้านเวลา การจัดการเครือข่ายของไฟล์ที่เกี่ยวข้อง และยังมีผลกระทบทางด้านประสิทธิภาพจากการมี include ซ้อนกัน และการจัดการอ็อบเจกต์ ไรต์ และ ขอบเขตของตัวแปร ผู้พัฒนาที่ต้องทำงานที่เกี่ยวข้องกับการทำงานกับเบราว์เซอร์หลายตัวมักพบปัญหาในการทำเว็บแอปพลิเคชันให้สามารถทำงานได้กับหลายๆเบราว์เซอร์ซึ่งจะต้องคำนึงถึงความสามารถของเบราว์เซอร์ที่ไคลเอนต์ใช้ ซึ่งวิธีการแก้ไขปัญหามักจบลงด้วยการใช้แท็กHTMLที่เป็นแท็กพื้นฐานที่สุด และใช้สคริปต์ทางฝั่งไคลเอนต์เพื่อให้เป็นที่เข้าใจได้โดยทุกๆเบราว์เซอร์บางครั้งมีความพยายามสร้างHTML marker สำหรับแต่ละเบราว์เซอร์โดยเฉพาะ ซึ่งได้ผลดีแต่เสียเวลาและต้องใช้ความพยายามอย่างมาก การเขียนสคริปต์ใน ASP ใช้ภาษาได้จำกัดเช่น VB สคริปต์ และ Java สคริปต์เป็นต้น และสคริปต์ทางฝั่งเซิร์ฟเวอร์ จะถูกแปลทุกครั้งทีหน้าASPนั้นถูกเรียก ซึ่งไม่เป็นผลดีต่อประสิทธิภาพในการทำงาน นอกจากนั้นในเรื่องการจัดการกับสถานะ ใน ASP นักพัฒนาจะต้องทำทุกอย่างด้วยตนเอง โดยจะต้องเก็บสถานะไว้ใน SQL เซิร์ฟเวอร์ หรือเนื้อที่เก็บข้อมูลภายนอกอื่นๆ จะเห็นได้ว่า จะเห็นได้ว่าแม้ASP จะเป็นเทคโนโลยีที่ดีในการสร้างเว็บเพจแบบไดนามิก แต่ก็ยังมีปัญหาอยู่อีกหลายด้าน จึงมีการแก้ไขและพัฒนามาเป็น ASP.NET

4.2 ที่มาของ ASP.NET

นักพัฒนาที่ใช้ Visual Basic มักจะคุ้นเคยกับความสะดวกสบายในการเขียนโปรแกรมโดยใช้ฟอร์มและคอนโทรลซึ่งมีรูปแบบที่สะดวก โดยลากสิ่งที่ต้องการลงมาไว้บนฟอร์มและเขียนโค้ดโดยใช้การเขียนโปรแกรมแบบ event-handling เนื่องจากวิธีนี้เป็นวิธีที่ไม่ยุ่งยาก จึงได้ถูกนำมาใช้ใน ASP.NET ASP.NET ทำให้การพัฒนาหน้าเว็บสะดวกขึ้นด้วยการเขียนโปรแกรมโดยใช้ฟอร์ม ใน ASP.NET เราเรียกฟอร์มเหล่านี้ว่า เว็บฟอร์ม (Web Forms) ซึ่งนำมาแทนที่หน้า ASP การเขียนโปรแกรมแบบเว็บฟอร์มนั้นเป็นแบบ event base เช่นเดียวกับใน visual basic นอกจากนี้ ASP.NET ยังแยกส่วนระหว่างส่วนแอปพลิเคชันและส่วนในการแสดงผลอีกด้วย ASP.NET เกี่ยวข้องกับรูปแบบการเขียนโปรแกรม ASP โดยมีคุณสมบัติพิเศษเพิ่มเติมขึ้นมา ดังนี้

- มีการแยกส่วนเค็ดขาดระหว่างส่วนโค้ดการประมวลผลของแอปพลิเคชันและส่วนแสดงผล (HTML markup) จึงไม่มีความสับสนในโค้ดทั้งสองส่วนอีก
- มีกลุ่มของเซิร์ฟเวอร์คอนโทรลจำนวนมากให้เลือกใช้ ซึ่งจะถูกละเปลี่ยนเป็น HTML โดยอัตโนมัติ
- มีการจัดการสถานะ โดยใช้ฮ็อบเจกต์เซสชัน
- เป็นรูปแบบการเขียนโปรแกรมแบบ event-base ทางฝั่งเซิร์ฟเวอร์ ทำให้ง่ายขึ้น
- สามารถเขียนโค้ดส่วนแอปพลิเคชันโดยใช้ภาษาใดใน Microsoft.NET ก็ได้ เช่น ภาษา VB, C# เป็นต้น โค้ดของแอปพลิเคชันบนฝั่งเซิร์ฟเวอร์จะถูกคอมไพล์เพื่อประสิทธิภาพการทำงานที่ดีกว่า
- มี Visual Studio.NET เป็นเครื่องมืออำนวยความสะดวกในการพัฒนาเว็บฟอร์ม

4.3 รู้จักกับ System.Web.UI Namespace

ในส่วนนี้จะอธิบายถึงคลาส ที่สำคัญที่อยู่ในเนมสเปซชื่อ System.Web.UI ในเฟรมเวิร์คของ ASP.NET เนมสเปซ System.Web.UI นั้นระบุคลาสและอินเทอร์เฟซที่ใช้ในการสร้างสิ่งต่างๆ ในคลาสเว็บฟอร์ม ที่สำคัญที่สุดใน System.Web.UI คือคลาส Control ซึ่งบอกคุณสมบัติ เมธอด และ event ของแต่ละเซิร์ฟเวอร์คอนโทรลในเฟรมเวิร์คของเว็บฟอร์ม อีกคลาสที่สำคัญคือ Page ซึ่งจะกล่าวถึงในลำดับต่อไป

4.3.1 คลาส Control

คลาส Control เป็นรากฐานของคอนโทรลทั้งหมด ตัวอย่างเช่น textbox หรือ button ก็คือคอนโทรลโดยทั่วไปคลาส Control จะมีฟังก์ชันและคุณสมบัติของส่วนติดต่อกับผู้ใช้ทุกตัว ทุกอย่างที่เราเห็นใน ASP.NET ก็เป็นคลาสที่สืบทอดมาจากแหล่งใดแหล่งหนึ่งทั้งสิ้น

4.3.1.1 คุณสมบัติของ Control

คลาส Control มีคุณสมบัติที่สำคัญดังต่อไปนี้ Control, ID, Parent, EnableViewState, Visible, Context และ ViewState คุณสมบัติของคอนโทรลเหล่านี้แทนลูกของ instance ของคอนโทรลคุณสมบัติของคลาสแม่จะบอกคลาส แม่ของคอนโทรล คุณสมบัติทั้งสองนี้ ทำให้มีลำดับชั้น(hierarchy)ของคอนโทรลในหน้าเว็บคุณสมบัติ ID ทำให้คอนโทรลถูกเข้าถึงจากการเขียนโปรแกรมได้โดยใช้ ID ตามด้วยจุดและชื่อคุณสมบัติหรือเมธอดอื่นๆที่เราต้องการจะเข้าถึงของคอนโทรลนั้น เช่น MyObjectId.propertyname และยังทำให้เราสามารถเขียนตัวจัดการกับเหตุการณ์ที่เกิดขึ้นกับคอนโทรลได้ด้วย EnableViewState เป็น flag ที่บอกว่าคอนโทรลจะยังคงเก็บview ของสถานะมันไว้หรือไม่ เช่นเดียวกับ view สถานะทั้งหมดของคอนโทรลลูกของมัน คุณสมบัติของ Context ทำให้เราสามารถดึงข้อมูลของHTTP request ได้ ViewState เป็นอินสแตนซ์ของคลาส StateBag ซึ่งใช้ในการเก็บชื่อและค่าของข้อมูลที่สามารถสร้างให้เข้าถึงได้โดยหลาย request สำหรับหน้าเพจเดียว ชื่อและค่าเหล่านี้คืออินสแตนซ์ของคลาส StateItem

4.3.1.2 เมธอดในคลาส Control

เมธอดของคลาส Control นั้นมีเป็นจำนวนมาก อย่างไรก็ตามเราจะกล่าวถึงเพียงเมธอดที่มีความสำคัญเมธอด DataBindเป็นการโยกคอนโทรลเข้ากับแหล่งข้อมูล เมื่อเมธอด นี้ถูกเรียก แท็กของการผูกข้อมูล <%#%> จะถูกนำมาพิจารณาใหม่เพื่อให้ข้อมูลใหม่ถูกโยกเข้ากับตำแหน่งtagที่เหมาะสมเมธอด CreateChildControlsจะถูกเรียกก่อนที่คอนโทรลcompositional custom จะถูกจัดแสดงคอนโทรล compositional custom นั้นคล้ายกับคอนโทรล ActiveX คือมันจะจัดการสร้างคอนโทรลอื่นๆในการพัฒนาคอนโทรล custom เมธอดนี้สามารถถูกoverride เพื่อให้ผู้พัฒนาคอนโทรล custom สามารถสร้าง คอนโทรลลูก ก่อนทำการแสดงคอนโทรลไม่ว่าในการทำครั้งแรกหรือครั้งต่อไปก็ตาม เมธอด Renderคล้ายกับ CreateChildControls เริ่มต้นใช้สำหรับพัฒนาคอนโทรล custom ผู้พัฒนาคอนโทรลจะoverrideเมธอดนี้ เพื่อแสดงตัวเนื้อหาในคอนโทรลผ่านทางพารามิเตอร์ HtmlTextWriter

ที่เตรียมไว้เมธอด SaveViewState และ LoadViewStateเก็บบันทึกและเรียกซ้ำสถานะของคอนโทรล คอนโทรลของเซิร์ฟเวอร์ คงสถานะของมันระหว่างการrequest ผ่านทางเมธอด นี้

4.3.2 คลาสเพจ

คลาสเพจนี้สืบทอดมาจากคลาส Control ซึ่งหมายความว่ามันสืบทอดมาทั้งคุณสมบัติเมธอด และevent ที่แสดงโดยคลาส Control นอกจากการสืบทอดแล้วคลาส เพจ ระบุคุณสมบัติเมธอด และ event ที่เจาะจงมากขึ้น สำหรับหน้าเว็บในเฟรมเวิร์คของ ASP.NET

ใน ASP.NET ทั้ง Application , Request , Respond , Server และ Session เป็นคุณสมบัติของ คลาสเพจ นอกเหนือจากอ็อบเจกต์เหล่านี้คลาส เพจ ยังมีคุณสมบัติอื่นๆอีก เช่น Cache, ErrorPage, Postback, IsValid, Trace และ Validators

4.3.2.1 ตัวอย่างคุณสมบัติและเมธอดของคลาสเพจได้แก่

คุณสมบัติ Cacheซึ่งไปยังอ็อบเจกต์ cache ของขอบเขตของหน้าปัจจุบัน ในที่นี้ ทรัพยากรเช่น การติดต่อฐานข้อมูล(database connection)จะถูกเก็บเพื่อนำมาใช้ใหม่โดยไม่ต้องสร้างการติดต่อใหม่ หากว่า cache item นั้นยังไม่หมดอายุคุณสมบัติ ErrorPageบอกหน้าที่จะถูกเรียกให้แสดงเมื่อมีerror เกิดขึ้น อาจใช้ @Pageแทนได้คุณสมบัติ IsPostBackบอกว่าการเรียกหน้าเว็บเป็นการ โหลดครั้งแรกหรือไม่ หาก IsPostBack มีค่าเป็น true แสดงว่าไม่ใช่การ โหลดครั้งแรก จึงไม่ควรทำการ initialize หน้าเพจซ้ำ เพื่อให้ประสิทธิภาพดีขึ้นคุณสมบัติ Validatorรวมกลุ่มของเซิร์ฟเวอร์คอนโทรลซึ่งสามารถ กำหนดระยะเวลาการทำงานของตัวเองลงในคุณสมบัติValidatorของหน้าเว็บ เมื่อหน้าเว็บใดต้องมีการ กำหนดอายุของตัวเอง ก็จะกำหนดอายุของคอนโทรลทั้งหมดในหน้านั้นแล้วกำหนดค่าที่เหมาะสม ให้แก่คุณสมบัติ IsValidคุณสมบัติ Traceอ้างอิงจากอ็อบเจกต์ TraceContextซึ่งใช้บอกค่าเตือนหรือ ข้อความบอกความผิดพลาดที่เกิดขึ้นสามารถกำหนดให้ทำการตรวจสอบหรือไม่ทำก็ได้โดยกำหนดใน web.config ซึ่งเป็นเท็กซ์ไฟล์ที่เก็บค่าขณะรันไทม์ของแอปพลิเคชันของASP.NET

4.4 ไวยากรณ์สำหรับ Web Form

4.4.1 ไคเร็กทีฟ

สำหรับ ASP แบบเก่า นั้นไคเร็กทีฟจะมีรูปแบบเป็น %@[attribute= value]% เพราะว่าไคเร็กทีฟในASP นั้นมีเพียงแบบเดียวเท่านั้นใน ASP.NET ได้มีการเพิ่มไคเร็กทีฟใหม่ซึ่งได้แก่ Page,

Control, Import, Register, Assemble และOutputCache และไวยากรณ์ใหม่ที่นำมาใช้คือ `<%@ directive [attribute=value] %>` โดยถ้าการประกาศกระทำโดยใช้ไวยากรณ์แบบเก่าคือไม่มีการใส่ไคเร็กทีฟ ก็จะมีหมายถึง ประกาศให้เป็นไคเร็กทีฟแบบปกติซึ่งก็คือ Page

4.4.1.1 @Page

ใน ASP.NET นอกจากจะประกอบด้วยไคเร็กทีฟแบบ ASP เก่าแล้ว ยังสนับสนุน attribute ที่สำคัญเช่น ErrorPage, Inherits, Src, และ EnableViewState อีกด้วยตัวอย่างการประกาศใช้งาน`<%@ Page Language="VB" ErrorPage = "URL" EnableViewState = "true"%>`

4.4.1.2 @Control

เช่นเดียวกันกับ Page ไคเร็กทีฟ จะมีการประกาศใช้ในไฟล์ (.aspx) สำหรับ Control ไคเร็กทีฟ นั้นจะใช้ใน ASP.NET Control คือประกาศในไฟล์ (.ascx)

4.4.1.3 @Import

เราสามารถใส่ Import ไคเร็กทีฟ ในการอ้างถึง เนมสเปซที่จะนำมาใช้ในหน้านั้นๆ ซึ่งโปรแกรมที่เราสร้างขึ้นจะสามารถเข้าถึงทุกๆคลาสที่มีอยู่ในเนมสเปซนั้น ตัวอย่างเช่นถ้าเราต้องการใช้งาน ADO.NET เราต้องใส่ไคเร็กทีฟดังนี้คือ

```
<%@ Import Namespace="System.Data" %>
```

```
<%@ Import Namespace="System.Data.OleDb" %>
```

4.4.1.4 @Register

เป็นไคเร็กทีฟที่ใช้ ลงทะเบียนการใช้งานเซิร์ฟเวอร์คอนโทรลในหน้า Page นั้นๆตัวอย่างเช่น

```
<%@ Register Tagprefix = "Ch07"
```

```
TagName = "MyCustomControl"
```

```
Src = "MyCustomControl.ascx" %>
```

โดยชื่อของคอนโทรลคือ MyCustomControl และ คำนำหน้าเมื่อการประกาศใช้คอนโทรลคือ Ch07

ที่อยู่ของคอนโทรลคือ MyCustomControl.ascx

4.4.1.5 @Assembly

เป็นไคเร็กทีฟที่ใช้บ่งบอกว่า Page นั้นๆ เป็นของ assemble ไคตัวอย่างเช่น

```
<%@ Assembly Name = "Ch07" %>
```

หมายความว่า โปรแกรมในหน้า Page นี้สามารถเข้าถึงอะไรก็ได้ใน assembly Ch07

4.4.1.6 @OutputCache

เราสามารถใส่ Output Cache ได้เรียกทีพในการควบคุม ระยะเวลาของ output caching ระหว่างที่หน้า Page นั้นเปิดขึ้น

4.4.2 ขอบเขตการประกาศคำสั่ง

เช่นเดียวกับ ASP ในการประกาศขอบเขตคำสั่งทำงาน จะหมายถึงการกำหนด ขอบเขตคำสั่งที่จะถูกประมวลผลและสั่งให้ทำงานในหน้า Page นั้นๆ ซึ่งในขอบเขตนั้นจะมี คุณลักษณะที่เรียกว่า “runat” ซึ่งเป็นการใช้กำหนดว่า ขอบเขตของคำสั่งเหล่านี้ จะประมวลผลที่ฝั่งไคลเอนต์ หรือ ฝั่งเซิร์ฟเวอร์ ซึ่งถ้าต้องการให้มีการทำงานที่ฝั่งเซิร์ฟเวอร์ จะต้องมีการกำหนดให้มี คุณลักษณะเป็น “server” และถ้าไม่มีการกำหนดจะหมายความว่าให้ทำงานในฝั่งไคลเอนต์ ตัวอย่างเช่น

```
<script runat= "server" [language="codelanguage"]>
```

```
'code
```

```
</script>
```

สำหรับการทำงานทั้งสองแบบเราสามารถกำหนดที่อยู่ของคำสั่งได้ โดยการทำให้เป็นการแยกส่วนของคำสั่งออกจากส่วนของเอกสาร HTML การประกาศกระทำโดย

```
<script runat="server"
```

```
[language="codelanguage"]
```

```
[src="externalfilename"]/>
```

4.4.3 HTML-Control Syntax

HTML คอนโทรลมีความคล้ายกับมาตรฐาน HTML ปกติอยู่มาก จะแตกต่างกันที่มีการเพิ่มการประกาศ “id” และ “runat” แอททริบิวต์ ซึ่งโดยปกติการใช้ id กับการเขียนโปรแกรมนั้นจะเจอกับการเขียนในลักษณะ Dynamic HTML จะแตกต่างกันตรงที่ HTML คอนโทรล จะเป็นลักษณะของคอนโทรลที่มีทำงานในฝั่งเซิร์ฟเวอร์ ตัวอย่างเช่น

```
<input id="cmd1" runat="server"
```

```
type="button" value="Click Me" />
```

โดยสังเกตว่าการประกาศต้องเป็นแบบ server เพราะว่าเว็บฟอร์มจะมีการใช้ POST เมธอดในการควบคุมเรื่องเหตุการณ์ทำงานเมื่อทำการประมวลผลคำสั่ง ASP.NET จะทำการสร้างสคริปต์ขึ้นจากออปเจ็กต์ HtmlControl จากตัวอย่างข้างต้น ASP.NET จะทำการสร้างคอนโทรลขึ้นจาก HtmlInputButton โดยมี id คือ cmd1 นอกจากนี้ยังสามารถควบคุม อีเวนต์ของคอนโทรลได้อีกด้วย โดยการประกาศจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระทำผ่าน HTML Tag ตัวอย่างเช่น ถ้าต้องการควบคุมอีเวนต์ onclick สามารถเพิ่มในส่วนของ HTML Tag ดังนี้

```
Onserverclick = "handleServerClick"
```

4.4.4 Custom-Control Syntax

ในส่วนของ คอนโทรลที่กำหนดเอง ก็จะคล้ายกับ HTML เซิร์ฟเวอร์คอนโทรล โดย Custom คอนโทรลจะมีส่วนของ id และ runat แอททริบิวต์ แต่อย่างไรก็ตาม Custom คอนโทรลก็ไม่เหมือนกับ HTMLมาตรฐาน ในการที่จะใช้งานเราสามารถประกาศตามรูปแบบดังนี้คือ

```
< tagprefix:tagname id="controlID" runat="server" eventname ="eventHandler" />
```

ในการเรียกใช้งาน จะต้องใส่ prefix เป็น asp ดังตัวอย่างต่อไปนี้

```
<asp:TextBox id=txt1 runat=server></asp:TextBox>
```

4.4.5 Data-Binding Expressions

การผูกข้อมูลหรือ Data Source เข้ากับคอนโทรลใดๆ เราสามารถทำได้ตามไวยากรณ์ดังนี้คือ

```
<%# data-binding-expression %>
```

โดยการใช้ Data Bind นั้น Web Form จะไม่ทำการ Bind ให้อัตโนมัติ เราจะต้องทำการใส่เมธอด DataBind() เองภายในโค้ดโปรแกรม ตัวอย่างการใช้ Data Binding ดังนี้คือ

```
<asp:Label text='<%# TestData %>' runat=server/>
```

```
<html>
```

```
<head><title>Data Binding</title></head>
```

```
<body>
```

```
<script language="C#" runat= server>
```

```
/*declare the variable we want to bind to.
```

```
Public string TestData;
```

```
Void Page_Load(Object oSender, EventArgs oEvent) {
```

```
TestData = "HelloWorld";
```

```
Page.DataBind();
```

```
}
```

```
</script>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
</body>
```

```
</html>
```

4.5 การพัฒนาเว็บแอปพลิเคชันด้วย ASP.NET

4.5.1 ส่วนประกอบคอมโพเนนต์ของ Web Form

ใน Web Forms pages นั้น การเขียน program ของ user interface แบ่งเป็นสองส่วนคือ visual Component และ logic ซึ่งก็คล้ายกับการที่ Visual Basic มีการแบ่งระหว่างส่วนของ form และส่วน code เบื้องหลังที่ form นั้นทำงานด้วยนั่นเอง ส่วนแสดงผลที่มองเห็นได้นั้นก็คือ Web Form page ซึ่งประกอบด้วยไฟล์ ที่มี HTML หรือ ASP.NET server control. Web Form page ทำงานเหมือนเป็น container สำหรับ static text และส่วนควบคุมที่เราต้องการแสดงผล ในส่วน logic ของ Web Form page ประกอบด้วย โค้ด ที่เรา สร้างขึ้นเพื่อสื่อสารกับ form logic การเขียน โปรแกรม จะอยู่แยกกับส่วน file ของ user interface ซึ่งไฟล์นี้จะเปรียบเสมือน โค้ดที่อยู่เบื้องหลังการทำงาน สามารถเขียนด้วยภาษา Visual Basic หรือ Visual C#

ไฟล์ที่ประกอบด้วยคลาสต่างๆและ เบื้องหลังการทำงาน สำหรับ Web Form ทั้งหมดจะถูกคอมไพล์เป็น ไฟล์ dynamic-link library(.dll) page ที่เป็น .aspx ก็จะถูก compile เช่นกัน แต่มีกระบวนการที่ต่างกันครั้งแรกที่ ผู้ใช้ เรียก file หน้า .aspx ASP.NET จะสร้างไฟล์ .NET class มาเป็นตัวแทนของ page โดยอัตโนมัติ แล้ว คอมไพล์ เป็น ไฟล์ .dll ตัวที่สอง class ที่ถูกสร้างสำหรับ .aspx page สืบทอดคุณสมบัติมาจากclass ที่เป็น โค้ดการทำงานเบื้องหลังที่ถูก compile ไปแล้วก่อนหน้านี้ เมื่อผู้ใช้ร้องขอ URL ของ web page ไฟล์ .dll จะ run บน server และสร้างผลเป็น HTML แบบไดนามิกสำหรับ page ที่เรียก

4.5.2 วงจรชีวิตของ Web Form

ใน ASP เว็บเพจจะเริ่มต้นวงจรชีวิตเมื่อ ไคลเอนท์เริ่มทำการร้องขอเพจที่เจาะจงมา IIS ทำการกระจายคำและทำการใช้งานสคริปต์บน ASP เพจ เพื่อทำให้เป็นเนื้อความ HTML ในที่สุดการทำการแปลงก็จะสิ้นสุดอายุของเพจนั้นก็สิ้นสุด ถ้าคุณมีฟอร์มที่ส่งข้อมูลกลับไป ASP เพจก็จะเริ่มกระบวนการใหม่เหมือนกับ ASP กระทำคำร้องใหม่ โดยไม่ต้องรู้เกี่ยวกับขั้นตอนก่อนหน้าเลย การส่งผ่านข้อมูลกลับไปเพจเริ่มต้นสำหรับการประมวลผลก็เป็นการถูกอ้างแบบโพสต์แบ็ค (postback) ด้วยใน ASP.NET มีหลายสิ่งที่แตกต่างกันเล็กน้อย เพจยังคงเริ่มที่การร้องขอของไคลเอนท์ อย่างไรก็ตามมันจะยังคงอยู่เรื่อยๆเท่าที่ไคลเอนท์ยังคงติดต่อกับเพจอยู่ เพื่อให้ง่ายขึ้นอาจจะกล่าวได้ว่าเพจยังคงอยู่ไปเรื่อยแต่ในความเป็นจริงมีเพียงขั้นตอนการดู(view)ของเพจระหว่างการร้องขอเท่านั้น

ขั้นตอนการดูเท่านั้นที่อนุญาตให้ควบคุมบนเซิร์ฟเวอร์ที่จะปรากฏว่า ถ้ามันยังคงครองอีเวนต์ของเซิร์ฟเวอร์ เราสามารถค้นหาขั้นตอนการโพสต์แบ็คของเพจผ่านคุณสมบัติ IsPostBack ของเพจอ็อบเจกต์และการกำหนดค่าเริ่มต้นใหม่ การครองอีเวนต์ระหว่างการโพสต์แบ็คมันก็คือการทำ ASP.NET แตกต่างกว่าการพัฒนา ASP ตามแบบธรรมเนียมอย่างมาก

จากตัวอย่างเราทำการขยายส่วนต้น เช่นการครองโพสต์แบ็ค เมื่อทำการโหลดอีเวนต์ที่ถูกถือครองครั้งแรก เราจะทำการย้ายข้อมูลลงใน drop-down list box หลังจากนั้นเราจะระบุเวลาของอีเวนต์ที่เพิ่มขึ้นโดยไม่มีการโหลดข้อมูลเข้ามาใหม่เท่านั้น ตัวอย่างนี้แสดงให้เห็นว่าเซิร์ฟเวอร์อีเวนต์แฮนด์เลอร์(handler)ที่ชื่อhandlerButtonClick เป็นขอบเขตของอีเวนต์ที่ชื่อ ServerClick ของปุ่ม

```
<html>
<head><title>Testing Page Event</title></head>
<body>
<script language="C#" runat=server>
void Page_Init(Object oSender,EventArgs oEvent){
labelInit.Text = DateTime.Now.ToString( );
}
void Page_Load (Object oSender, EventArgs oEvent) {
labelInit.Text = DateTime.Now.ToString( );
if (!IsPostBack){
selectCtrl.Items.Add("Acrua");
selectCtrl.Items.Add("BMW");
selectCtrl.Items.Add("Cadillac");
selectCtrl.Items.Add("Mercedes");
selectCtrl.Items.Add("Porche");
}else{
labelLoad.text += "(Postback)";
}
}

void handleButtonClick(Object oSender, EventArgs oEvent){
labelOutput.Text = "You've selected: "+ selectCtrl.Value;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

labelEvent.Text = DateTime.Now.ToString ( );
}
</script>
<form runat = server>
Init Time: <asp:Label id=labelInit runat=server/><br/>
Load Time: <asp:Label id=labelLoad runat=server/><br/>
Event Time: <asp:Label id=labelEvent runat=server/><br/>
Choice: <select id=selectCtrl runat=server/><br/>
<asp:Label id=labelOutput runat=server/></select><br/>
<input type=button value=update
onServerClick="handleButtonClick" runat=server />
</form>
</body>
</html>

```

วงจรกิจกรรมของเว็บฟอร์ม ประกอบด้วยสามขั้นตอนหลัก การกำหนดรูปแบบ การถือครองอีเวนต์ และการสิ้นสุด โดยขั้นตอนเหล่านี้จะข้ามผ่านคำร้องขอที่เหมือนกันซึ่งแยกกันโดยแบบ serving-one-page-at-a-time ที่พบใน ASP

4.5.2.1 การกำหนดรูปแบบ

ขั้นตอนการกำหนดรูปแบบนี้ โหลดอีเวนต์ของเพจจะถูกยกขึ้นมันเป็นงานของคุณในการถือครองอีเวนต์นี้เพื่อติดตั้งเพจคุณ เพราะว่าโหลดอีเวนต์จะถูกยกขึ้นเมื่อการควบคุมพร้อมแล้วงานคุณตอนนี้คือการอ่านและทำการอัปเดตคุณสมบัติการควบคุมเหมือนเป็นส่วนของการติดตั้งเพจ ในตัวอย่างโค้ดข้างต้น เราจะถือครอง โหลดอีเวนต์เพื่อย้ายข้อมูลบางอย่างลง drop-down list เราอัปเดตข้อความของการควบคุม labelLoad เพื่อแสดง เวลาโหลดอีเวนต์ที่เกิดขึ้น ในแอปพลิเคชันของคุณคุณจะต้องข้อมูลที่ เป็นไปได้จาก ฐานข้อมูลและทำการหนดค่าเริ่มต้นฟอร์มฟิลด์ด้วยค่าดีฟอลต์คุณสมบัติ IsPostBack ของเพจแสดงว่าเพจแรกได้ถูกโหลดหรือมันเป็นการโพสต์แบคเช่นถ้าคุณควบคุมเนื้อหารายการของ ข้อมูลคุณจะต้องโหลดการควบคุมเพจแรกที่ถูกโหลดโดยการตรวจสอบคุณสมบัติIsPostBackของเพจนี้ เท่านั้นเมื่อ IsPostBack มีค่าเป็นจริงคุณจะรู้ว่ารายการควบคุมอ็อบเจกต์ได้ถูกโหลดด้วยข้อมูลเรียบร้อยแล้ว ไม่จำเป็นต้องย้ายข้อมูล ในรายการใหม่ตัวอย่างโค้ดข้างต้นเราจะข้ามการย้ายข้อมูล

ของ drop-down และแสดงเพียง สดริงว่า “(postback)” คุณอาจจะต้องกระทำการการผูก(bind)ข้อมูลและ กำหนดค่าใหม่ data-binding expressions ในตอนแรก และลำดับต่อมาของเพจนี้

4.5.2.2 การถือครองอีเวนต์

ในขั้นตอนกลางนี้ ฟังก์ชัน server event-handlingของเพจจะถูกเรียกเป็นผลของบางอีเวนต์เป็น การถูกเรียกให้ทำงานจากฝั่ง ไคลเอนต์ อีเวนต์เหล่านี้มาจากการควบคุมที่ผู้แทนที่บนเว็บฟอร์ม

4.5.2.3 การจบการทำงาน

ที่ขั้นตอนนี้เพจจะสิ้นสุดการแปลงและถูกกำจัดทิ้งแล้วสิ่งที่คุณต้องรับผิดชอบคือการทำความสะอาดไฟล์แฮนด์เลอร์ ยกเลิกการติดต่อกับฐานข้อมูลและปล่อยอ็อบเจกต์ถึงแม้คุณจะเชื่อมั่นการทำงานของตัวกำจัดขยะ(garbage collection)ของ CLR ก็ตามแต่เรายังแนะนำอย่างยิ่งยวดว่าควรทำความสะอาดด้วยตัวเองเพราะตัวกำจัดขยะทำงานเป็นระยะๆเท่านั้น บนการทำงานหนักๆถ้าตัวกำจัดขยะไม่เหมาะสมจะมีทรัพยากรที่ไม่ได้คืนให้ระบบทำให้หน่วยความจำหมดไปได้และทำให้ระบบคุณ ค้างไปเราสามารถทำความสะอาดตัวอย่างข้างต้นได้ด้วยการอันโหลดอีเวนต์แฮนด์เลอร์ดังที่จะแสดง ต่อไปนี้เพราะว่า ไม่มีแสดงในตัวอย่างเราจะแสดงเฉพาะฟังก์ชันเป็น เทมเพลต

```
Void Page_Unload(Object oSender,EventArgs oEvent){
//cleaning up code here
}
```

4.6 การจัดการ State และ Scalability

4.6.1 ASP.NET Session-State Management

ASP.NET ได้ปรับปรุงการจัดการเรื่องสแตทใน ASP เดิม โดยปรับปรุงมาเป็นลักษณะโมเดล แบบout-of-process โดยการจัดการสแตทจะมองลักษณะของฟาร์มของ Web Server โดยการสั่งให้มีการ สลับไปยังเพจใดๆ ในฟาร์มจะไม่เกิดการหายของเซสชันในการเปลี่ยนแปลงค่าเกี่ยวกับการจัดการการ บริหาร session ใน ASP.NET เราสามารถกระทำผ่านไฟล์ที่ชื่อว่า web.config โดยความเป็นจริงแล้วเรา สามารถทำการกำหนดค่าได้สองทางได้แก่ทางระดับ แมทชีน และระดับ แอปพลิเคชัน โดยในระดับแม ทชีนสามารถกำหนดผ่านไฟล์ที่อยู่ในที่Microsoft.NET\Framework\

```
<configuration>
<system.web>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<sessionstate  
mode="Inproc"  
cookieless="false"  
timeout="20" />  
</system.web>  
</configuration>
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การทดลองและผลการทดลอง

ในหัวข้อนี้จะกล่าวถึง การทดลองและผลการทดลอง ใช้ซึ่งเนื้อหาในบทนี้ จะกล่าวถึงการทดลองและผลการทดลองใช้งานระบบในส่วนต่างๆในเว็บแอปพลิเคชัน(Web Application) ประกอบด้วยส่วนของการเข้าใช้งานหน้าเว็บและรูปแบบต่างๆในการใช้งาน

รูปแบบของการใช้งาน

Blur คือการปรับรูปภาพให้เปลี่ยนไปเป็นรูปภาพที่มีลักษณะเบลอขึ้นกว่าเดิม เมื่อต้องการเพิ่มการเบลอของรูปภาพก็เพียงคลิกที่ปุ่ม Blur

Contras คือการปรับความเข้มของรูปภาพเมื่อต้องการเพิ่มความเข้มของรูปภาพก็สามารถเพิ่มได้โดยคลิกที่ปุ่ม Contrast

Sharpen คือการปรับให้รูปภาพให้มีการคมชัดมากขึ้น โดยการคลิกที่ปุ่ม Sharpen

Brightness คือการปรับความสว่างของรูปภาพเมื่อต้องการเพิ่มความสว่างของรูปภาพก็สามารถเพิ่มได้โดยคลิกที่ปุ่ม Brightness

Negative คือการปรับรูปภาพไปเป็นสีที่ตรงข้ามของรูปภาพ โดยการคลิกที่ปุ่ม Negative

Remove noise คือการปรับให้รูปภาพมีความเรียบเนียนมากขึ้น

Grayscale คือการปรับรูปภาพให้อยู่ในโทนสีเทา

Flip Horizon คือการปรับรูปภาพ 180 องศาในแนวนอน

Flip Vertical คือการปรับรูปภาพ 180 องศาในแนวตั้ง

Kodalith คือการปรับรูปภาพให้เป็นโทนสีขาวดำ

Watercolor คือการปรับรูปภาพให้ไปเป็นเหมือนภาพวาดสีน้ำ

Mosaic Effect คือการปรับรูปภาพให้ไปเป็นเหมือนภาพตัดปะ โดยสามารถใส่ขนาดของช่องได้

Outline คือการปรับรูปภาพไปเป็นเหมือนการ LayOut ภาพ

Frame Black คือการใส่กรอบสีดำให้กับรูปภาพ

Frame White คือการใส่กรอบสีขาวให้กับรูปภาพ

Resize คือการปรับขนาดของรูปภาพ

Text คือการใส่ข้อความลงไปยังรูปภาพ โดยพิมพ์ข้อความลงในช่อง Text

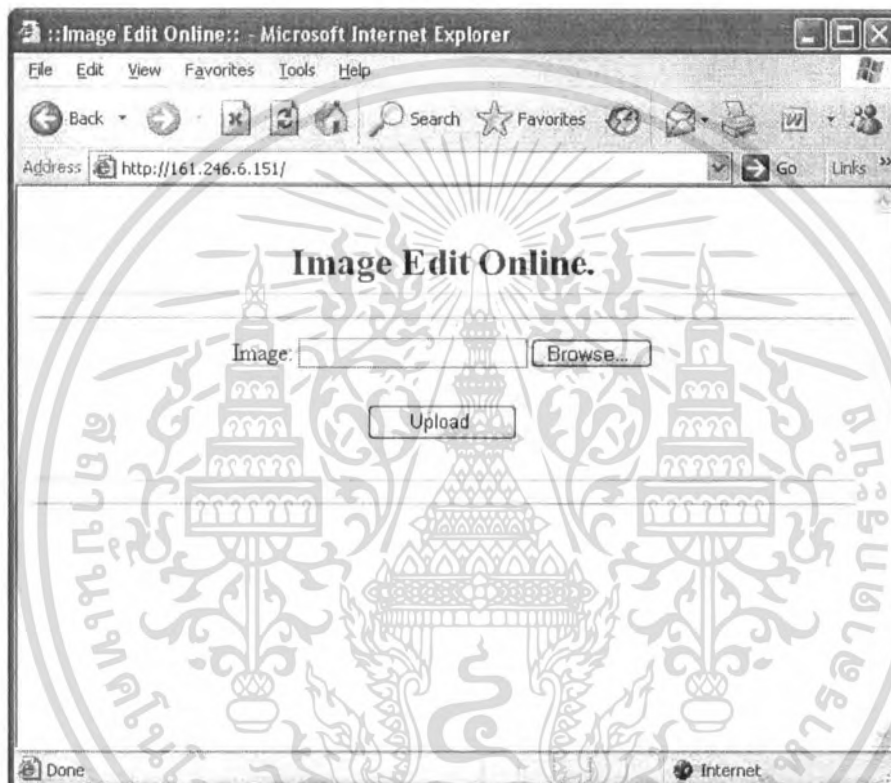
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Zoom คือการขยายรูปภาพเพื่อมุมมองที่ชัดเจน

Undo คือการเรียกเอาภาพเดิมกลับมา ผู้ใช้ต้องการปรับแต่งใหม่ โดยคลิกที่ Undo

5.1 การใช้งานเบื้องต้น

เปิดหน้าเว็บขึ้นมาจะเห็นหน้าตาหลัก

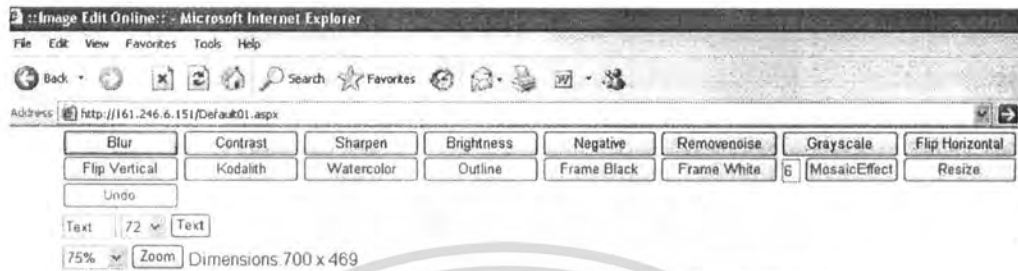


รูปที่ 5.1 แสดงรูปหน้าเว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การเข้าใช้ระบบ ในส่วนของเว็บแอปพลิเคชัน

กดที่ปุ่ม Browse จะแสดงหน้าต่างขึ้นมาให้เลือกรูป เมื่อเลือกรูปที่ต้องการแล้วกดคลิกที่ Upload

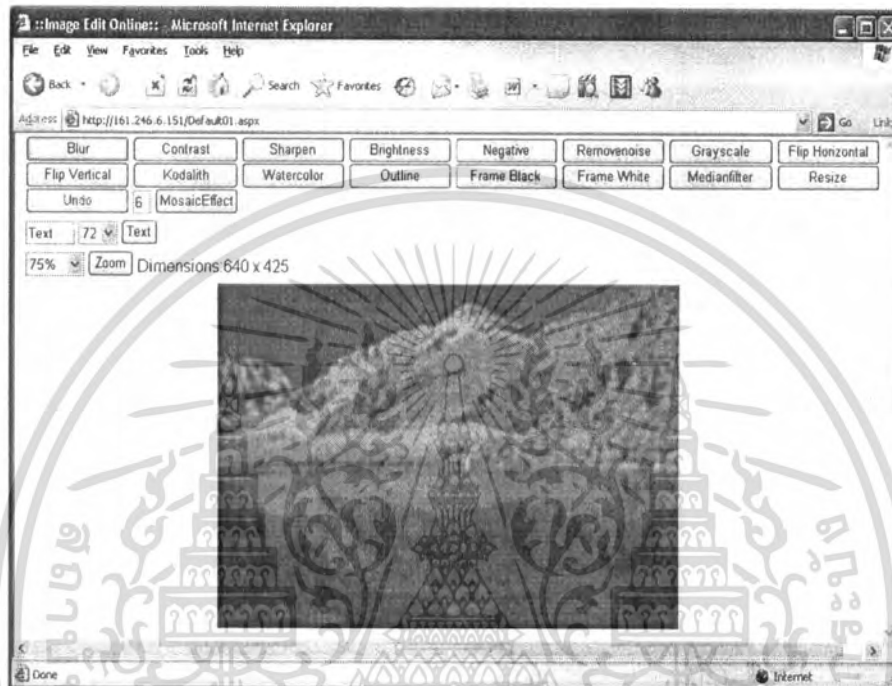


รูปที่ 5.2 แสดงรูปหน้าต่างเมื่อเลือกรูปภาพที่ต้องการแล้วคลิกที่ Upload

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 การใช้งานในรูปแบบต่างๆ

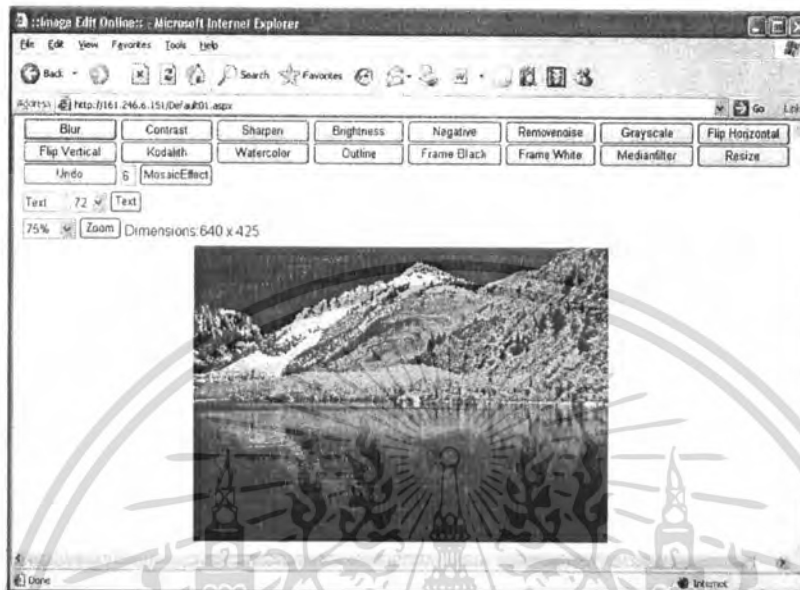
5.3.1 **Blur** คือการปรับรูปภาพให้เปลี่ยนไปเป็นรูปภาพที่มีลักษณะเบลอขึ้นกว่าเดิม เมื่อต้องการเพิ่มการเบลอของรูปภาพก็เพียงคลิกที่ปุ่ม Blur



รูปที่ 5.3 แสดงการใช้งานรูปแบบ Blur

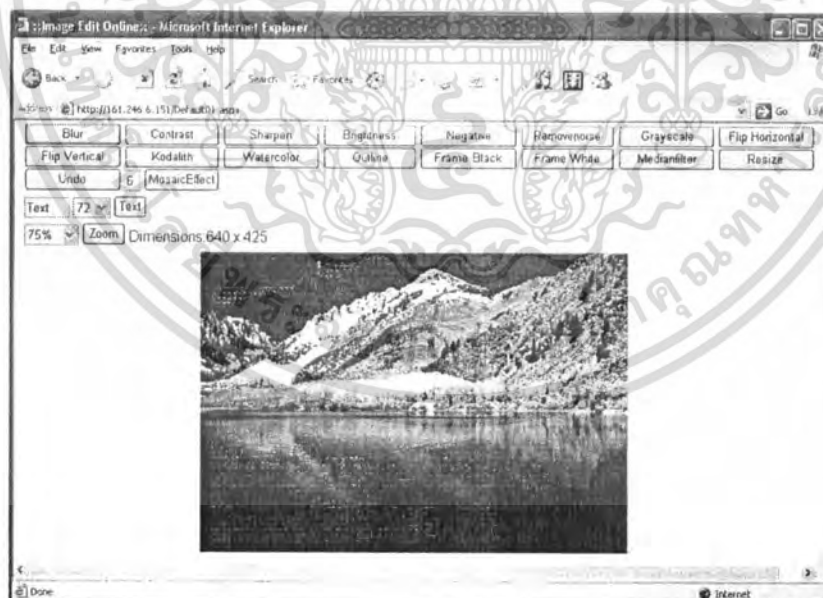
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.2 Contrast คือการปรับความเข้มของรูปภาพเมื่อต้องการเพิ่มความเข้มของรูปภาพก็สามารถเพิ่มได้ โดยคลิกที่ปุ่ม Contrast



รูปที่ 5.4 แสดงการใช้งานรูปแบบ Contrast

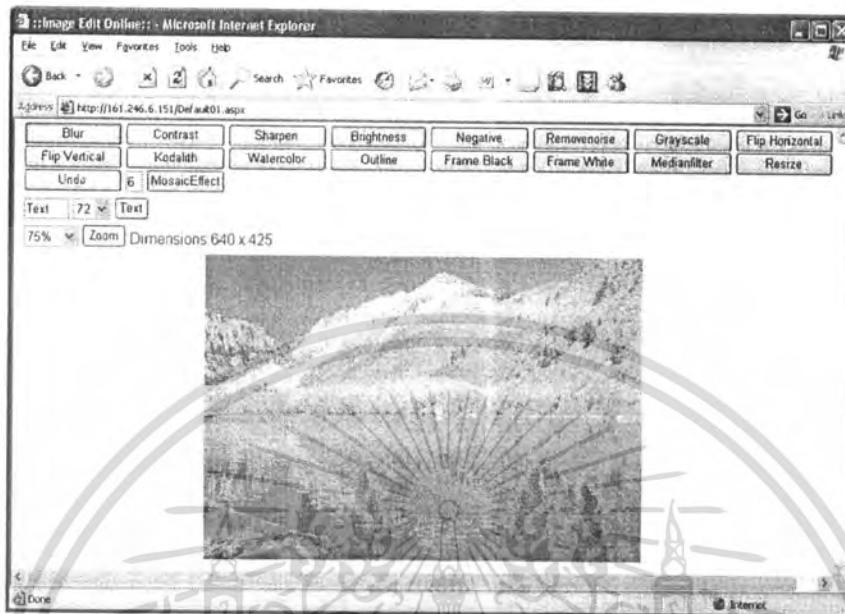
5.3.3 Sharpen คือการปรับให้รูปภาพให้มีการคมชัดมากขึ้น โดยการคลิกที่ปุ่ม Sharpen



รูปที่ 5.5 แสดงการใช้งานรูปแบบ Sharpen

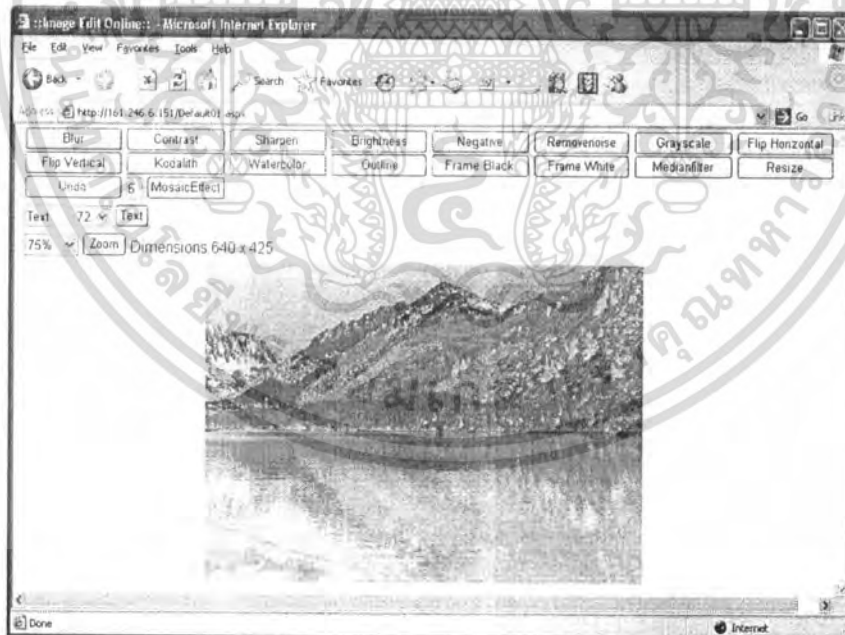
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.4 Brightness คือการปรับความสว่างของรูปภาพเมื่อต้องการเพิ่มความสว่างของรูปภาพก็สามารถเพิ่มได้โดยคลิกที่ปุ่ม Brightness



รูปที่ 5.6 แสดงการใช้งานรูปแบบ Brightness

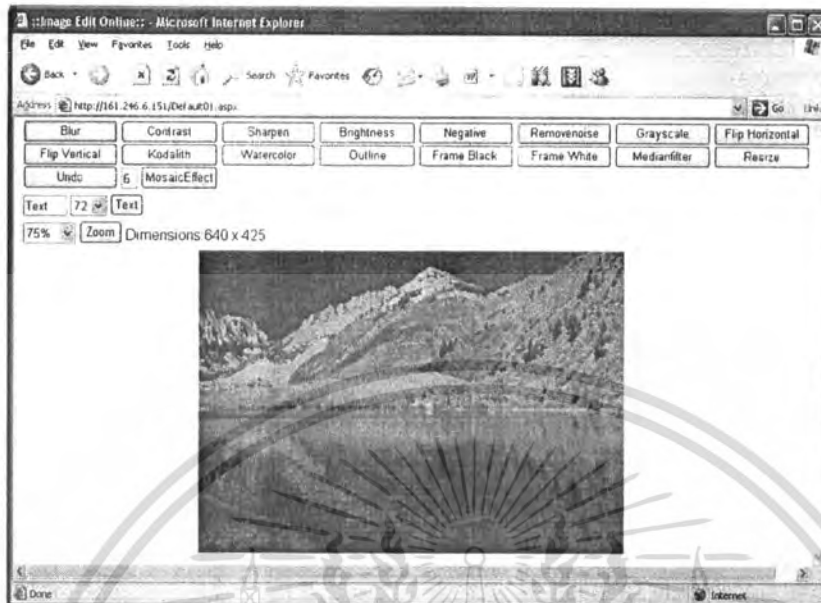
5.3.5 Negative คือการปรับรูปภาพไปเป็นสีที่ตรงข้ามของรูปภาพ โดยการคลิกที่ปุ่ม Negative



รูปที่ 5.7 แสดงการใช้งานรูปแบบ Negative

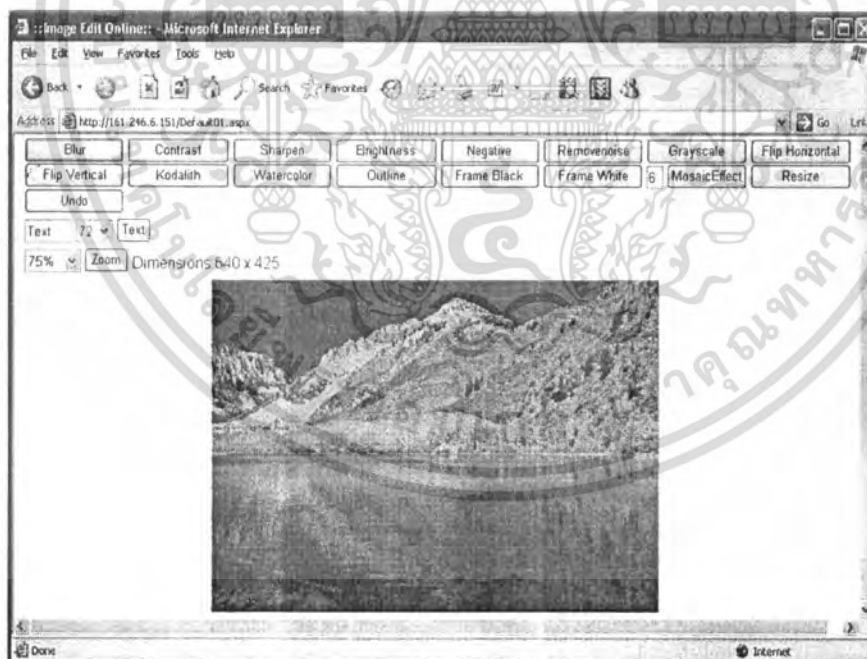
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.6 Remove noise คือการปรับให้รูปภาพมีความเรียบเนียนมากขึ้น



รูปที่ 5.8 แสดงการใช้งานรูปแบบ Remove noise

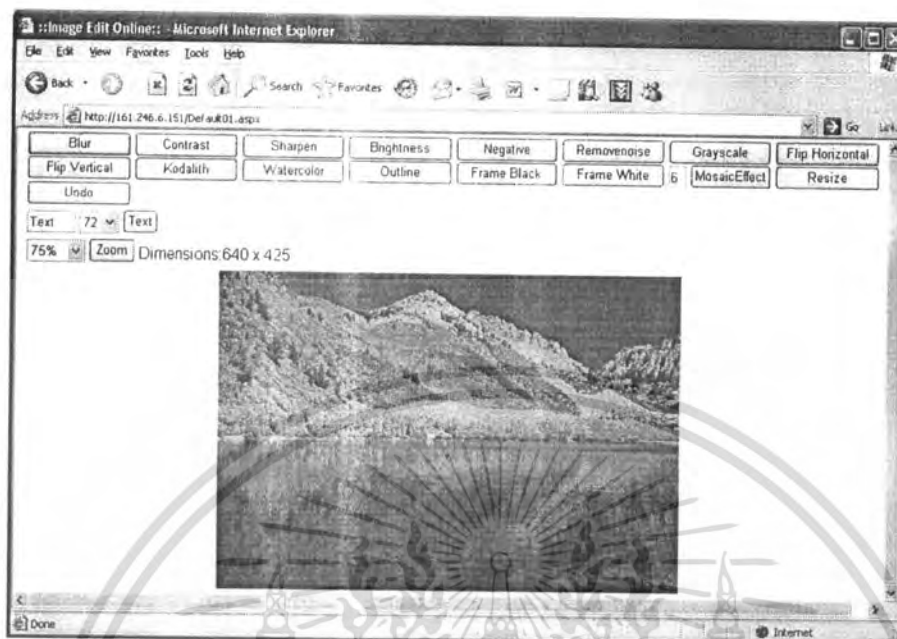
5.3.7 Grayscale คือการปรับรูปภาพให้อยู่ใน โทนสีเทา



รูปที่ 5.9 แสดงการใช้งานรูปแบบ Grayscale

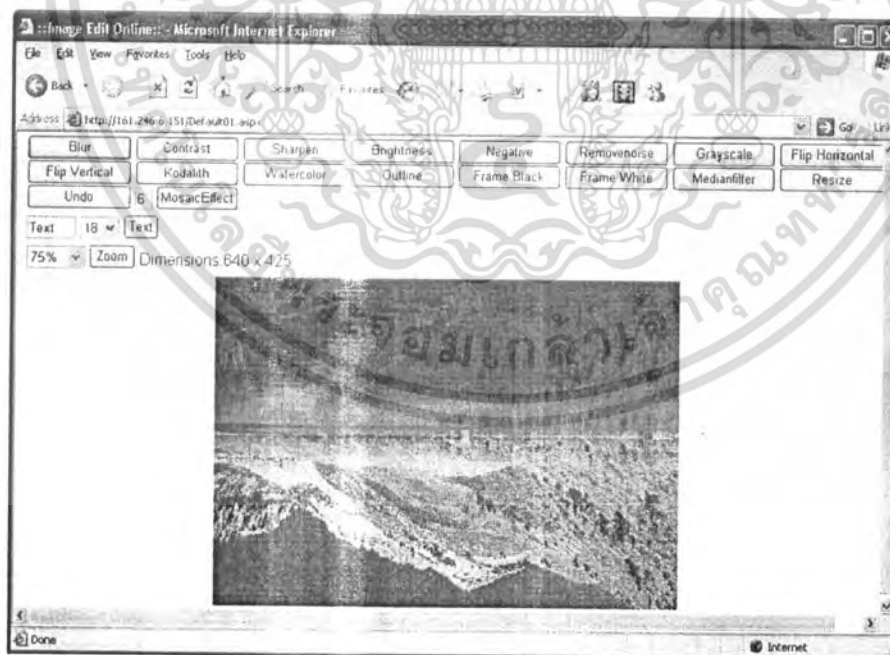
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.8 Flip Horizon คือการปรับรูปภาพ 180 องศาในแนวนอน



รูปที่ 5.10 แสดงการใช้งานรูปแบบ Flip Horizon

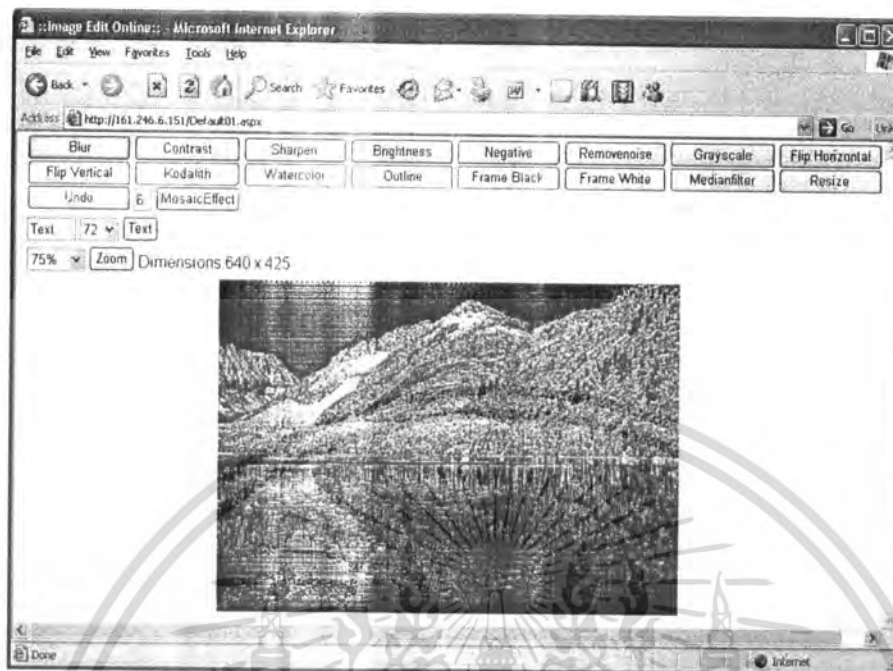
5.3.9 Flip Vertical คือการปรับรูปภาพ 180 องศาในแนวตั้ง



รูปที่ 5.11 แสดงการใช้งานรูปแบบ Flip Vertical

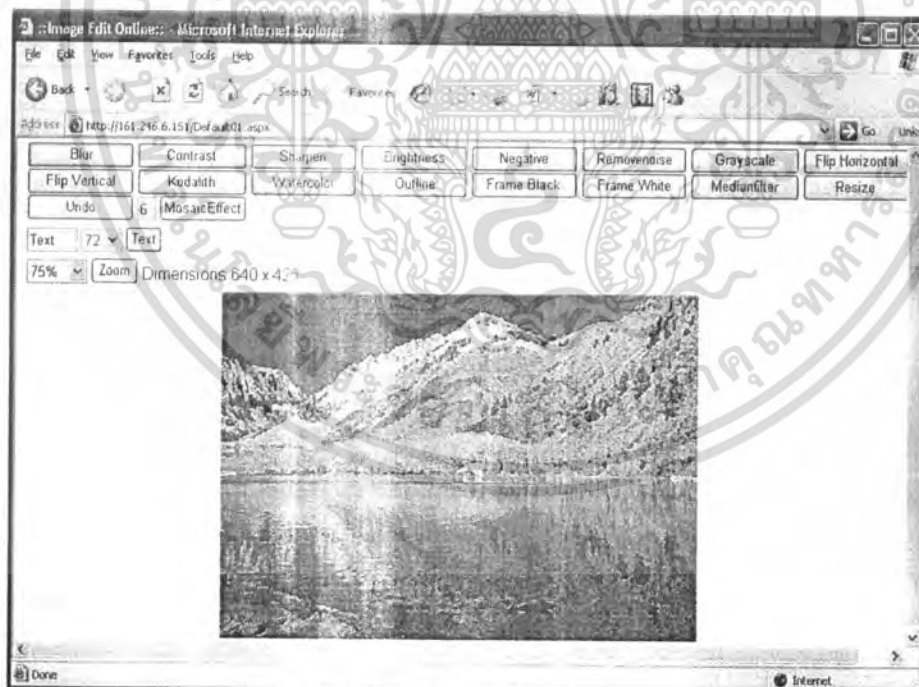
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.10 Kodalith คือการปรับรูปภาพให้เป็นโทนสีขาวดำ



รูปที่ 5.12 แสดงการใช้งานรูปแบบ Kodalith

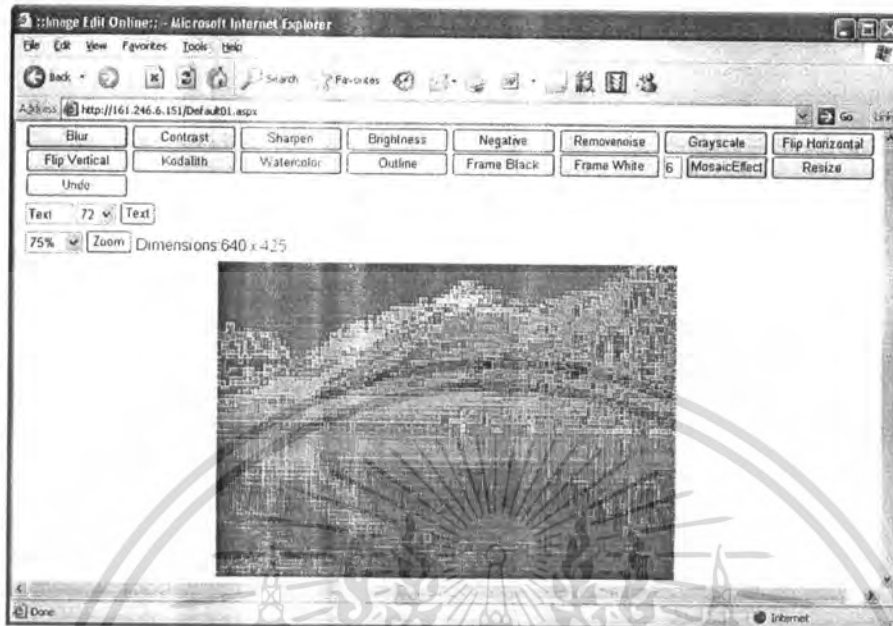
5.3.11 Watercolor คือการปรับรูปภาพให้ไปเป็นเหมือนภาพวาดสีน้ำ



รูปที่ 5.13 แสดงการใช้งานรูปแบบ Watercolor

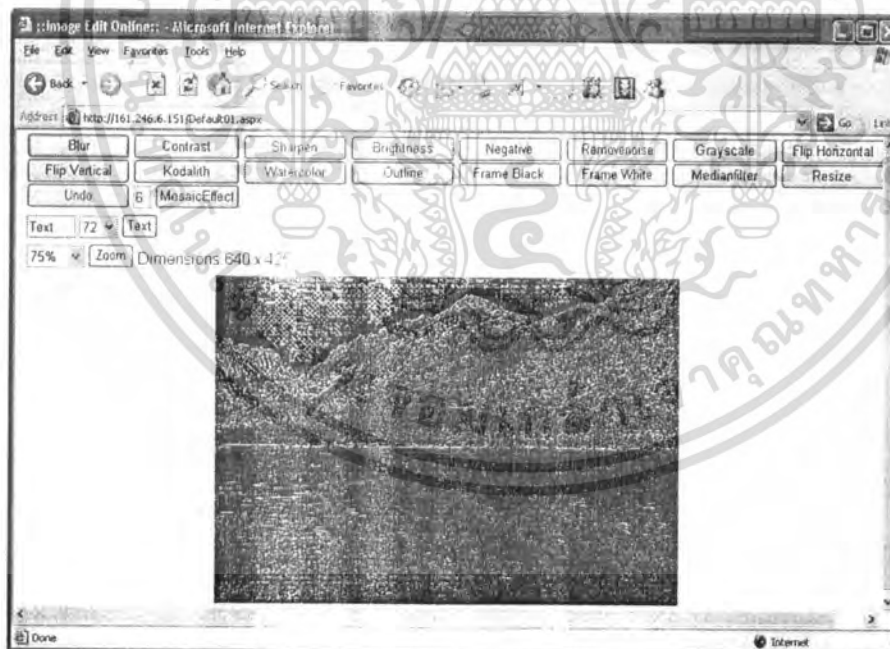
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.12 Mosaic Effect คือการปรับรูปภาพให้ไปเป็นเหมือนภาพตัดปะ โดยสามารถใส่ขนาดของช่องได้



รูปที่ 5.14 แสดงการใช้งานรูปแบบ Mosaic Effect

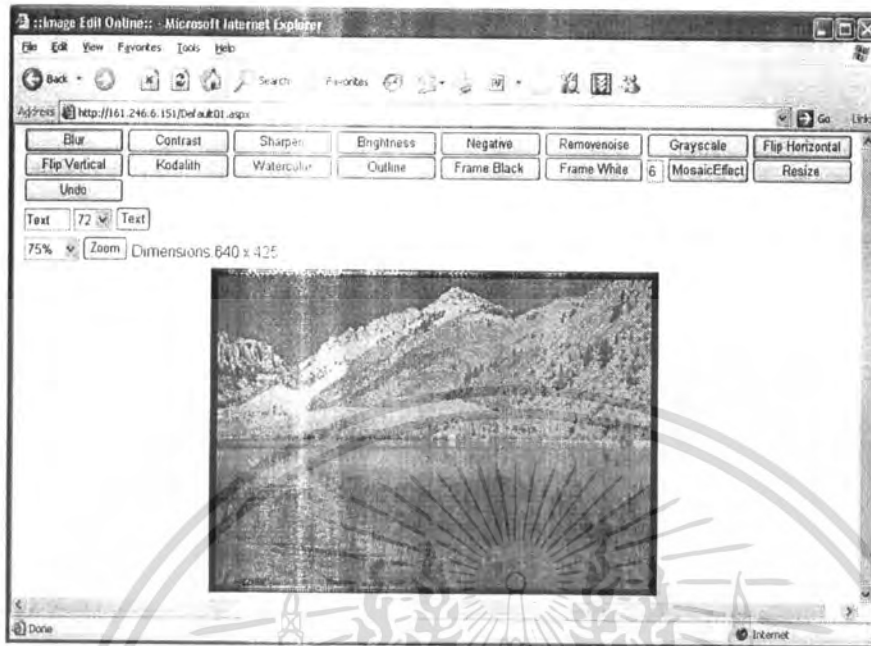
5.3.13 Outline คือการปรับรูปภาพไปเป็นเหมือนการ LayOut ภาพ



รูปที่ 5.15 แสดงการใช้งานรูปแบบ Outline

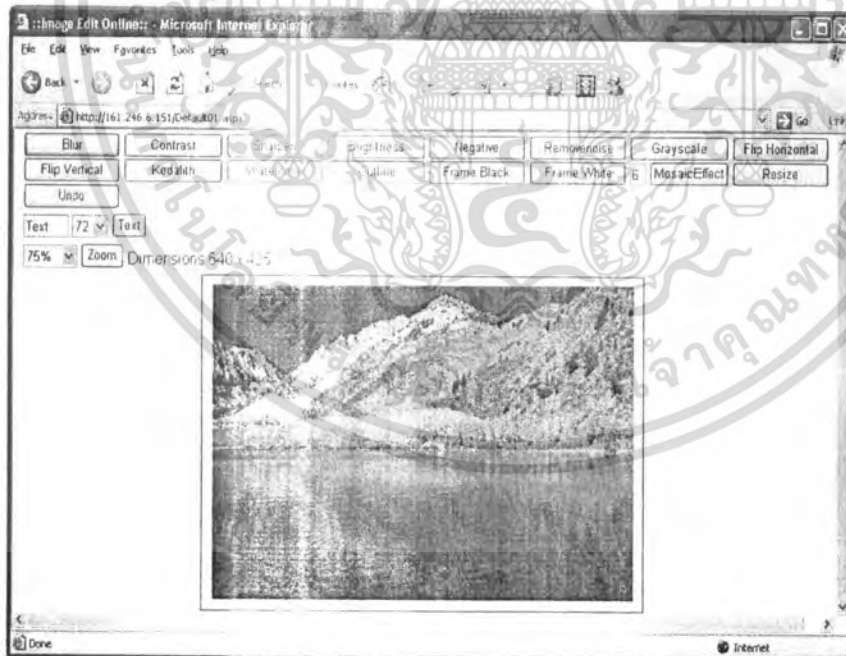
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.14 Frame Black คือการใส่กรอบสีดำให้กับรูปภาพ



รูปที่ 5.16 แสดงการใช้งานรูปแบบ Frame Black

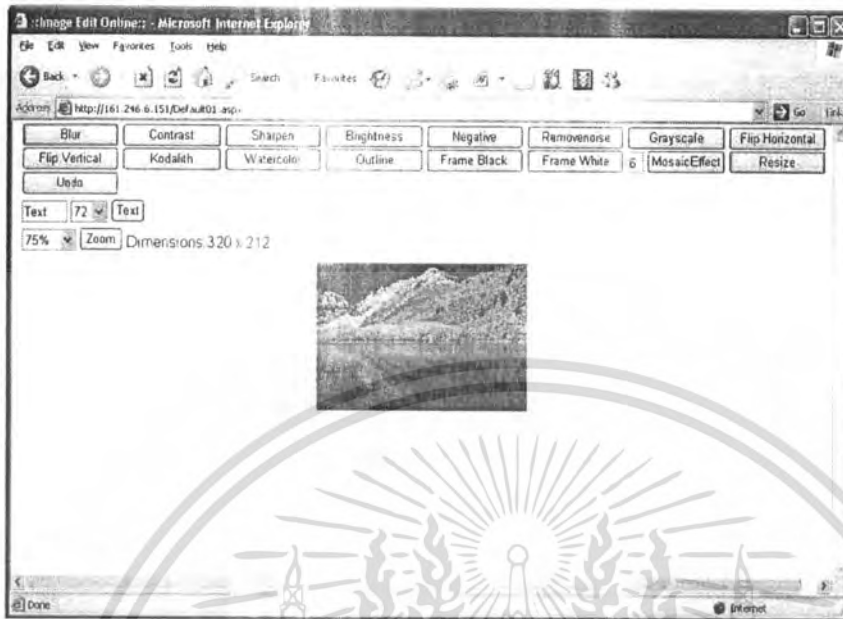
5.3.15 Frame White คือการใส่กรอบสีขาวให้กับรูปภาพ



รูปที่ 5.17 แสดงการใช้งานรูปแบบ Frame White

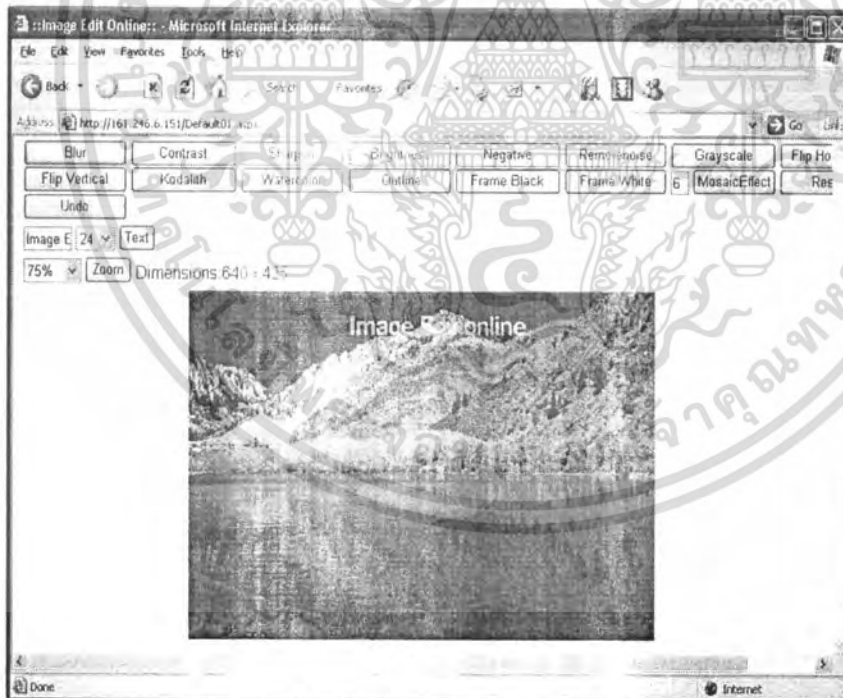
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.16 Resize คือการปรับขนาดของรูปภาพ



รูปที่ 5.18 แสดงการใช้งานรูปแบบ Resize

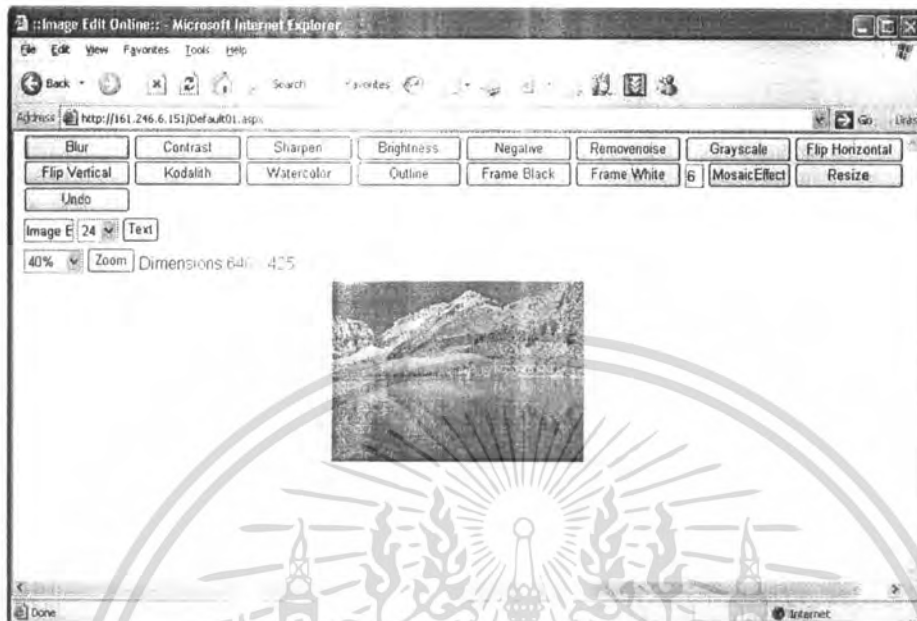
5.3.17 Text คือการใส่ข้อความลงไปยังรูปภาพโดยพิมพ์ข้อความลงในช่อง Text



รูปที่ 5.19 แสดงการใช้งานรูปแบบ Text

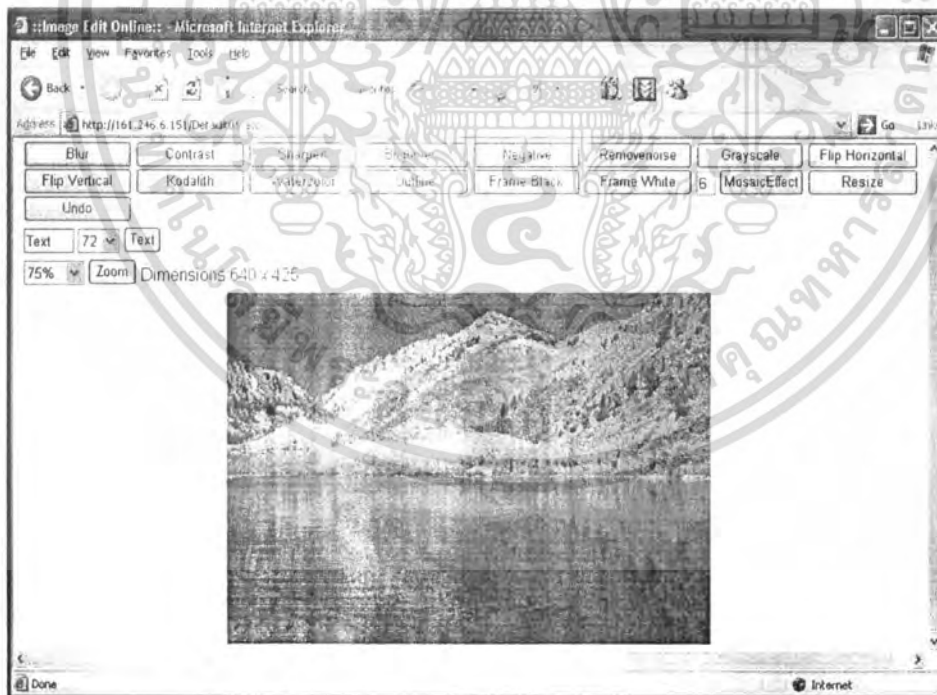
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.18 Zoom คือการขยายรูปภาพเพื่อมุมมองที่ชัดขึ้น สามารถใส่ขนาดจะต้องการ



รูปที่ 5.20 แสดงการใช้งานรูปแบบ Zoom

5.3.19 Undo คือการเรียกเอาภาพเดิมกลับมา ผู้ใช้ต้องการปรับแต่งใหม่ โดยคลิกที่ Undo



รูปที่ 5.21 แสดงการใช้งานรูปแบบ Undo

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทสรุป

6.1 บทสรุป

เว็บเซิร์ฟเวอร์สำหรับการประมวลผลภาพ (Image Processing Server) เป็นทางเลือกที่เข้ามาช่วยให้ผู้ใช้งานที่ต้องการปรับแต่งรูปภาพที่มีโดยอาจได้มาจากการถ่ายด้วยกล้องดิจิทัล ซึ่งผู้ใช้งานจะไม่ต้องติดตั้งโปรแกรมเพื่อที่จะปรับแต่งรูปภาพที่เครื่องเลย เพียงแต่ผู้ใช้งานทำการเชื่อมต่อมายังเว็บเซิร์ฟเวอร์โดยผ่านเครือข่ายอินเทอร์เน็ต และเมื่อผู้ใช้งานทำการเชื่อมต่อมายังเว็บเซิร์ฟเวอร์แล้ว ก็สามารถทำการปรับแต่งรูปภาพตามความต้องการ เมื่อตกแต่งเสร็จแล้วผู้ใช้งานก็สามารถโหลดไฟล์รูปภาพกลับเครื่องผู้ใช้งานได้

6.2 ปัญหาและอุปสรรค

การศึกษาเกี่ยวกับการประมวลผลภาพนั้นมีหลายภาษาที่สามารถทำได้จึงเสียเวลาในการทดลองและเลือกภาษาที่จะมาพัฒนา จึงทำให้เสียเวลาไปส่วนหนึ่ง และเป็นการเริ่มต้นการใช้งาน Visual C# ซึ่งไม่เคยใช้มาก่อนจึงต้องเสียเวลาในการศึกษาและพัฒนา

6.3 แนวทางการพัฒนาต่อ

1. เพิ่มรูปแบบในการปรับแต่งรูปภาพให้มากขึ้นเพื่อตอบสนองความต้องการผู้ใช้งานให้มากที่สุดและตกแต่งหน้าของเว็บไซต์ให้มีความสวยงามแล้วนำใช้มากขึ้น
2. ทดสอบแล้วหาข้อบกพร่องแล้วทำการแก้ไข

บรรณานุกรม

<http://msdn.microsoft.com>

<http://www.codeproject.com>

<http://www.sourceforge.com>

<http://www.cshapdeverlop.com>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้