

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

การพัฒนาโปรแกรมการซ้อนทับฉากหลังด้วยเทคนิค

การประมาณค่าความโปร่งแสงของพิกเซล

Development of Digital Matting Program using Alpha Estimation



H002361



วัน เดือน ปี.....	21 ก.พ. 2558
เลขทะเบียน.....	02361
เลขเรียกหนังสือ.....	อท. 8186 2548
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."	

61170 767x
1128 48554

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 1 ปีการศึกษา 2548
คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การพัฒนาโปรแกรมการซ้อนทับฉากหลังด้วยเทคนิคการประมวลค่าความโปร่งแสงของพิกเซล
นักศึกษา	นางสาวสุนิสา วงษ์ที
อาจารย์ที่ปรึกษา	รศ.ดร.นพพร โชติคำธรร
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2548

บทคัดย่อ

ในปัจจุบันมีการนำคอมพิวเตอร์เข้ามาช่วยในการประมวลผลภาพเพื่อสร้างสรรค์สื่อต่างๆ เช่น สื่อสิ่งพิมพ์ เป็นต้น มีเทคนิคการประมวลผลภาพมากมายที่ถูกพัฒนาขึ้น เทคนิคหนึ่งที่น่าสนใจคือ เทคนิคการซ้อนทับฉากหลัง (ซึ่งอาจเป็นภาพธรรมชาติ หรือฉากหลังพื้นสีน้ำเงินที่เรียกว่า บลูสกรีน) แต่ปัญหาของเทคนิคนี้คือ บริเวณขอบของภาพวัตถุที่เป็นเส้นผม ขนสัตว์ ใบไม้ หรือคว้น ซึ่งมีลักษณะของขอบที่เป็นเส้นโค้งและมีขนาดเล็กกว่า 1 พิกเซล บริเวณดังกล่าวเมื่อทำการแยกภาพวัตถุออกจากฉากหลังแล้วจะพบว่า ยังคงมีสีของฉากหลังติดมาด้วย โครงการพัฒนาระบบงานนี้ได้ทำการศึกษาการแยกภาพวัตถุออกจากฉากหลังเดิมด้วยเทคนิคการประมวลค่าความโปร่งแสงของพิกเซลเพื่อแก้ปัญหาที่เกิดขึ้น โดยทำการประมวลค่าความโปร่งแสงของพิกเซลแล้วคำนวณหาค่าสีที่แท้จริงของพิกเซลบริเวณขอบของภาพวัตถุ โดยใช้ข้อมูลค่าสีของพิกเซลในส่วนของภาพวัตถุและฉากหลังมาช่วยในการประมวลผล เพื่อให้ได้ค่าประมวลค่าความโปร่งแสงของพิกเซลและค่าประมวลค่าสีจริงของพิกเซลที่อยู่บริเวณขอบของภาพวัตถุที่ใกล้เคียงค่าจริงมากที่สุด ในการพัฒนาโปรแกรมเพื่อการประมวลผลภาพดังกล่าวใช้ Microsoft Visual Basic 6.0 ในการพัฒนาบนระบบปฏิบัติการ Microsoft Windows XP โดยโปรแกรมที่พัฒนาขึ้นสามารถอ่านไฟล์ภาพ (มีนามสกุลเป็น .bmp เท่านั้น) แล้วทำการแยกภาพวัตถุออกจากฉากหลังเดิม เพื่อนำไปซ้อนทับบนฉากหลังใหม่ และสามารถบันทึกภาพเป็นไฟล์ภาพที่มีนามสกุลเป็น .bmp ได้

Title	Development of Digital Matting Program using Alpha Estimation
Student	Miss Sunisa Vongtee
Advisor	Assoc. Prof. Dr. Nopporn Chotikakamthorn
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2005

ABSTRACT

Nowadays, computers are widely used to perform image processing for multimedia content creation. There are several image processing techniques available. One of them is known as a digital matting technique. However, a problem of this technique is that pixels at foreground - background boundary involving natural objects (e.g., strands of hair, fur, leaves, and smoke) are often smaller than one pixel. When extracting a foreground image from a background, it is often found that these pixels contain part of the background color. The purpose of this system development project is to study an algorithm to extract a foreground image from its background by estimating the foreground alpha value. The associated RGB colors and alpha channel of pixels in boundary regions are calculated by examining the colors of nearby pixels in foreground and background. Microsoft Visual Basic 6.0 was chosen as a development tool. The program runs on Microsoft Windows XP. This digital matting program can read a bitmap image (bmp) to extract foreground image from its background for compositing foreground image on a new background image. The program can read and save image in BMP image file format only.

กิตติกรรมประกาศ

ขอกราบขอบพระคุณ รศ.ดร.นพพร โชติภักดิ์ อาจารย์ที่ปรึกษาโครงการ ที่
กรุณาให้ความรู้ คำปรึกษา และคำแนะนำต่างๆ ในการพัฒนาโครงการให้สำเร็จลุล่วงได้ด้วยดี

ขอกราบขอบพระคุณคณาจารย์ทุกท่านในคณะเทคโนโลยีสารสนเทศที่กรุณา
ถ่ายทอดความรู้ และเทคนิคต่างๆ ในการศึกษา ตลอดระยะเวลาที่ศึกษาในสถาบันเทคโนโลยีพระ
จอมเกล้าเจ้าคุณทหารลาดกระบัง

ขอขอบคุณพี่ๆ และเพื่อนๆ ในห้องปฏิบัติการสื่อประสมและระบบเสมือน และ
คณะเทคโนโลยีสารสนเทศทุกคนที่คอยให้คำปรึกษา คำแนะนำ และกำลังใจในยามที่มีปัญหา

สุดท้ายขอกราบขอบพระคุณคุณพ่อ คุณแม่ และบุคคลในครอบครัว ที่ให้กำลังใจ
และให้การอุปการะมาโดยตลอด ขอกราบขอบพระคุณเป็นอย่างสูง

คุณความดีและประโยชน์ที่ได้รับจาก โครงการนี้ ผู้เขียนขอบอบแด่ผู้มีพระคุณทุก
ท่าน

สุนิสา วงษ์ที

27 ตุลาคม 2548

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญภาพ.....	VIII
บทที่	
1. บทนำ.....	1
1.1 โครงการพัฒนาระบบงาน.....	1
1.2 แนวทางการพัฒนา.....	1
1.3 วัตถุประสงค์.....	2
1.4 ขอบเขตและข้อจำกัดของโครงการ.....	2
1.5 อุปกรณ์และเครื่องมือในการดำเนินโครงการ.....	2
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	3
2. การแยกภาพวัตถุออกมาจากฉากหลัง.....	4
2.1 หลักการทั่วไป.....	4
2.2 ขั้นตอนในการแยกภาพวัตถุออกมาจากฉากหลัง.....	4
2.3 การกำหนดส่วนต่างๆ ของภาพ.....	5
2.4 การจัดกลุ่มค่าสี.....	6
2.4.1 การจัดกลุ่มข้อมูลด้วยวิธี K-Means Clustering.....	7
2.5 การจับคู่ค่าสีของพิกเซล.....	8
2.6 การประมาณค่าความโปร่งแสงของพิกเซล.....	9
2.7 การประมาณค่าสีจริงของพิกเซล.....	11
2.8 การซ้อนทับภาพ.....	12

สารบัญ (ต่อ)

บทที่	หน้า
3. การทำงานของโปรแกรมและอัลกอริทึม.....	13
3.1 โครงสร้างการทำงานของโปรแกรม.....	13
3.2 การทำงานของโปรแกรมและอัลกอริทึม.....	14
3.2.1 การเปิดไฟล์ภาพ.....	14
3.2.2 การเก็บข้อมูลค่าสีของพิกเซล.....	15
3.2.3 การจัดกลุ่มค่าสีของพิกเซล.....	16
3.2.4 การจับคู่ค่าสีของพิกเซล.....	22
3.2.5 การประมาณค่าความโปร่งแสงของพิกเซล.....	24
3.2.6 การประมาณค่าสีจริงของพิกเซล.....	25
3.2.7 การซ้อนทับภาพวัตถุลงบนฉากหลังใหม่.....	25
4 การออกแบบ User-interface และการใช้งาน โปรแกรม.....	26
4.1 การออกแบบ User-interface.....	26
4.2 การใช้งาน โปรแกรม.....	27
4.2.1 การเปิดโปรแกรม.....	27
4.2.2 การเปิดไฟล์ภาพ.....	28
4.2.3 การเริ่มฟังก์ชันการคำนวณของ โปรแกรม.....	29
4.2.4 การบันทึกไฟล์ภาพ.....	30
4.2.5 การเริ่มต้นทำงานใหม่.....	31
4.2.6 การออกจากโปรแกรม.....	31
5 สรุปผลการดำเนินโครงการ.....	32
5.1 การเปรียบเทียบภาพผลลัพธ์.....	32
5.2 สรุปผลการดำเนินโครงการ.....	33
5.3 ข้อจำกัดและปัญหาของ โครงการ.....	33
5.4 ข้อเสนอแนะ.....	33

สารบัญ (ต่อ)

	หน้า
บรรณานุกรม.....	34
ภาคผนวก.....	35
ก. การกำหนดส่วนต่างๆ ของภาพโดยใช้โปรแกรม Adobe Photoshop.....	36
ข. การติดตั้งโปรแกรม Blue-screen Matting.....	40
ประวัติผู้เขียน.....	43



สารบัญตาราง

ตารางที่	หน้า
3.1 แสดงอัลกอริทึมของ OpenImage.....	14
3.2 แสดงอัลกอริทึมของ GetPixelColor.....	15
3.3 แสดงอัลกอริทึมของ RGBcubeCluster.....	16
3.4 แสดงอัลกอริทึมของ CalculateKmeansClustering.....	21
3.5 แสดงอัลกอริทึมของ MatchPixel.....	23
3.6 แสดงอัลกอริทึมของ minDistance.....	23
3.7 แสดงอัลกอริทึมของ EstimateAlpha.....	24
3.8 แสดงอัลกอริทึมของ CalculateGaussianFunc.....	24
3.9 แสดงอัลกอริทึมของ EstimateTrueColor.....	25
3.10 แสดงอัลกอริทึมของ CompositeImage.....	25

สารบัญภาพ

รูปที่	หน้า
2.1	แสดงตัวอย่างภาพต้นฉบับ..... 5
2.2	ก) แสดงการกำหนดส่วนต่างๆ ของภาพ..... 6
2.2	ข) แสดงภาพ Mask ที่จะนำไปใช้ในการประมวลผล..... 6
2.3	แสดงภาพของ RGB Cube..... 7
2.4	แสดงช่วงของค่าสีที่แบ่ง RGB Cube ออกเป็น 64 กลุ่ม..... 7
2.5	แสดงการจับคู่ค่าสีของพิกเซล..... 9
2.6	แสดงกราฟของค่าความน่าจะเป็นของค่าความ โปร่งแสง..... 10
2.7	แสดงภาพของค่าประมาณของค่าความ โปร่งแสงที่คำนวณได้..... 11
2.8	แสดงภาพที่ได้จากการซ้อนทับภาพวัตถุลงบนภาพฉากหลังใหม่..... 12
3.1	โครงสร้างทั้งหมดของโปรแกรม..... 13
4.1	แสดงแผนภาพการออกแบบ User-interface..... 26
4.2	แสดงการเปิด โปรแกรม..... 27
4.3	แสดงลักษณะของ โปรแกรม Digital Matting..... 27
4.4	แสดงหน้าต่างการเปิดไฟล์ภาพต้นฉบับ..... 28
4.5	แสดงปุ่ม Open ของการเปิด ไฟล์ภาพ Mask..... 28
4.6	แสดงหน้าต่างของไฟล์ภาพ Mask และปุ่ม Open ของการเปิดไฟล์ภาพฉากหลัง ใหม่..... 29
4.7	แสดงหน้าต่างของภาพฉากหลังใหม่และปุ่ม Calculate..... 29
4.8	แสดงหน้าต่างแสดงภาพค่าประมาณความ โปร่งแสงและภาพผลลัพธ์จากการ ซ้อนทับภาพ..... 30
4.9	แสดงเมนูของการบันทึก ไฟล์ภาพ..... 30
4.10	แสดงการคลิกเลือกเมนูเพื่อเริ่มการทำงานใหม่..... 31
4.11	แสดงการคลิกเลือกเมนูเพื่อปิดการทำงานของโปรแกรม..... 31
5.1	ก) แสดงภาพผลลัพธ์ที่ได้จาก โปรแกรมที่พัฒนาขึ้นจากโครงการนี้..... 32
5.1	ข) แสดงภาพผลลัพธ์ที่ได้จาก โปรแกรม Adobe Photoshop..... 32

บทที่ 1

บทนำ

1.1 โครงการพัฒนาระบบงาน

ในปัจจุบันมีวิธีการสร้างสรรค์ภาพหลากหลายวิธี สำหรับการนำไปใช้ในสื่อต่างๆ ซึ่งมีการนำคอมพิวเตอร์เข้ามาช่วยในการประมวลผล หรือที่เรียกว่า Image Processing วิธีการสร้างสรรค์ภาพอย่างหนึ่งที่น่าสนใจ คือ การแยกภาพวัตถุออกมาจากฉากหลัง (ซึ่งอาจเป็นภาพธรรมชาติ หรือฉากหลังสีน้ำเงินที่เรียกว่า ฉากบลูสกรีน) แล้วสามารถนำภาพวัตถุนั้นไปวางซ้อนทับลงบนภาพอื่นๆ ได้ ตัวอย่างเช่น การถ่ายภาพคนที่มีฉากหลังเป็นฉากบลูสกรีน แล้วทำการประมวลผลเพื่อแยกภาพคนออกมา จากนั้นก็สามารถนำภาพคนไปวางซ้อนทับลงบนภาพวิวทิวทัศน์อื่นๆ ได้ เป็นต้น ทำให้สามารถสร้างสรรค์ภาพได้ตามจินตนาการที่เราต้องการ

โครงการพัฒนาระบบงานนี้ ได้ทำการศึกษาวิธีการประมวลผลเพื่อแยกภาพวัตถุออกมาจากฉากหลังด้วยเทคนิคการประมาณค่าความโปร่งแสงของพิกเซล เนื่องจากปัญหาหนึ่งที่เกิดขึ้นคือ บริเวณขอบของภาพวัตถุที่เป็นเส้นผม ขนสัตว์ ใบไม้ หรือควัน จะมีลักษณะของขอบที่เป็นเส้นโค้งและมีขนาดเล็กกว่า 1 พิกเซล เมื่อทำการแยกภาพวัตถุออกมาจากฉากหลังแล้ว จะพบว่าบริเวณดังกล่าวยังคงมีสีของฉากหลังติดมาด้วย ถ้าฉากหลังเป็นฉากบลูสกรีน บริเวณขอบของภาพวัตถุจะมีสีน้ำเงินติดอยู่ด้วย ดังนั้นจึงต้องทำการประมาณค่าความโปร่งแสงและค่าสีจริงของแต่ละพิกเซล ในบริเวณขอบของภาพวัตถุ เพื่อกำจัดค่าสีของฉากหลังสำหรับทุกพิกเซล ในบริเวณแนวขอบของภาพวัตถุ

1.2 แนวทางการพัฒนา

ในการพัฒนาโครงการนี้จำเป็นต้องมีขั้นตอนในการพัฒนา ซึ่งประกอบไปด้วย

1. ศึกษาเทคนิคและวิธีการประมวลผลภาพที่ใช้ในโครงการ
2. วิเคราะห์และออกแบบการทำงานของทั้งโครงการ
3. พัฒนาโปรแกรมที่ใช้ในโครงการ
4. ทดสอบการทำงานของโปรแกรม
5. บันทึกผลการดำเนินโครงการ

6. สรุปผลการดำเนินโครงการ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเชิงในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. เขียนรายงานประกอบการทำโครงการ

1.3 วัตถุประสงค์

1. เพื่อศึกษาวิธีการและเทคนิคการแยกภาพวัตถุออกมาจากฉากหลังด้วยการประมาณค่าความโปร่งแสงของพิกเซล
2. เพื่อศึกษาวิธีการแบ่งกลุ่มข้อมูลที่มีจำนวนข้อมูลมาก
3. เพื่อศึกษาวิธีการประมาณค่าความโปร่งแสงและการประมาณค่าสีจริง
4. เพื่อพัฒนาโปรแกรมการแยกภาพวัตถุออกมาจากฉากหลังด้วยเทคนิคการประมาณค่าความโปร่งแสงของพิกเซล

1.4 ขอบเขตและข้อจำกัดของโครงการ

ในการพัฒนาโครงการมีขอบเขตและข้อจำกัดในการทำงานและการใช้งานดังนี้

1. พัฒนาระบบปฏิบัติการ Microsoft Windows XP
2. โปรแกรมที่พัฒนาขึ้นจากโครงการนี้ใช้สำหรับการแยกภาพวัตถุออกมาจากฉากหลังที่มีลักษณะเป็นภาพนิ่งเท่านั้น
3. ภาพวัตถุที่ต้องการแยกออกมาจากฉากหลังจะต้องไม่มีค่าสีที่ใกล้เคียงกับค่าสีของฉากหลัง
4. ขนาดของภาพที่จะนำมาประมวลผลต้องมีขนาดไม่เกิน 90,000 พิกเซล เนื่องจากขนาดหน่วยความจำของ โปรแกรมมีจำกัด
5. ภาพที่จะนำมาใช้ในโปรแกรมที่พัฒนาขึ้นจากโครงการนี้ จะต้องเป็นภาพบิตแมปที่มีไฟล์นามสกุลเป็น .bmp เท่านั้น

1.5 อุปกรณ์และเครื่องมือในการดำเนินโครงการ

เครื่องคอมพิวเตอร์ ระบบปฏิบัติการ และ โปรแกรมที่ใช้ในการพัฒนาและดำเนินโครงการประกอบด้วย

1. CPU Pentium4 2.4 GHz, RAM 512 MB, Harddisk 60 GB
2. ระบบปฏิบัติการ Microsoft Windows XP
3. โปรแกรม MATLAB สำหรับการทดลองประมวลผลตามขั้นตอนต่างๆ
4. โปรแกรม Microsoft Visual Basic 6.0 สำหรับการสร้างและพัฒนาโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. โปรแกรม Microsoft Word 2002 สำหรับเขียนรายงานประกอบการทำโครงการ

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. มีโปรแกรมการแยกภาพวัตถุออกมาจากฉากหลังที่มีขนาดเล็ก ใช้งานง่าย โดยผู้ที่มีปฏิสัมพันธ์กับ โปรแกรมน้อยที่สุด
2. เป็นแนวทางในการพัฒนาโปรแกรมการแยกภาพวัตถุออกมาจากฉากหลังสำหรับภาพเคลื่อนไหว
3. สามารถนำภาพที่ได้จากโปรแกรมมาสร้างสรรค์ภาพสำหรับนำไปใช้ในสื่อต่างๆ ได้



บทที่ 2

การแยกภาพวัตถุออกมาจากฉากหลัง

2.1 หลักการทั่วไป

โครงการพัฒนาระบบงานนี้ จะทำการประมวลผลเพื่อแยกภาพวัตถุออกมาจากฉากหลังแล้วนำไปซ้อนทับลงบนภาพใหม่ที่ต้องการ แต่ปัญหาหนึ่งที่พบในการแยกภาพวัตถุออกมาจากฉากหลังคือ บริเวณขอบของภาพวัตถุที่เป็นเส้นผม ขนสัตว์ ใบไม้ หรือคว้น จะมีลักษณะของขอบที่เป็นเส้นโค้งและมีขนาดเล็กกว่า 1 พิกเซล เมื่อทำการแยกภาพวัตถุออกมาจากฉากหลังแล้ว จะพบว่าบริเวณดังกล่าวยังคงมีสีของฉากหลังติดมาด้วย ดังนั้นภาพที่จะนำมาประมวลผลด้วยโปรแกรมที่พัฒนาขึ้นนี้ ในภาพวัตถุจะต้องไม่มีพิกเซลใดที่มีค่าสีใกล้เคียงกับค่าสีของฉากหลัง เพราะค่าสีทุกพิกเซลของภาพวัตถุและฉากหลังจะนำมาใช้ในการคำนวณสำหรับการประมาณค่าความโปร่งแสงและค่าสีจริงของแต่ละพิกเซลในบริเวณขอบของภาพวัตถุ

ค่าความโปร่งแสง (Alpha Channel - α) เป็นแชนแนลที่ 4 ซึ่งเพิ่มเข้าไปในระบบสี RGB เพื่อใช้เก็บค่าความโปร่งแสงและทึบแสงของแต่ละพิกเซลบนภาพบิตแมป เรียกว่าระบบสี RGBA ซึ่ง A ก็คือค่า Alpha นั่นเอง ค่าความโปร่งแสงเป็นคุณสมบัติของพิกเซลที่ใช้แสดงว่าพิกเซลนั้นยอมให้ค่าสีของพิกเซลนั้นปรากฏออกมาได้มากน้อยแค่ไหน ค่าความโปร่งแสงมีค่าอยู่ระหว่าง 0 ถึง 1 ถ้าค่า $\alpha = 0$ หมายถึง พิกเซลนั้นทึบแสง (Opaque) ซึ่งจะไม่ยอมให้ค่าสีนั้นปรากฏออกมาได้เลย แต่ถ้าค่า $\alpha = 1$ หมายถึง พิกเซลนั้นโปร่งแสง (Transparent) ซึ่งจะยอมให้ค่าสีนั้นปรากฏออกมาได้ 100 %

2.2 ขั้นตอนในการแยกภาพวัตถุออกมาจากฉากหลัง

ภาพที่สามารถนำมาใช้ในการประมวลผลสำหรับ โปรแกรมที่พัฒนาขึ้นนี้ จะต้องเป็นภาพวัตถุหรือภาพคนที่มีฉากหลังเป็นภาพธรรมชาติหรือฉากบลูสกรีน ซึ่งอาจจะถ่ายด้วยกล้องดิจิทัลแล้วโหลดเข้าคอมพิวเตอร์ หรือสแกนรูปภาพเข้าคอมพิวเตอร์ แล้วเก็บข้อมูลของภาพให้อยู่ในรูปแบบของภาพบิตแมปที่มีไฟล์นามสกุลเป็น .bmp เท่านั้น ซึ่งแต่ละพิกเซลของภาพบิตแมปจะแสดงผลการผสมสีของแสงสีแดง สีเขียว และสีน้ำเงินหรือที่เรียกว่า แม่สี RGB และภาพควรมีขนาดไม่เกิน 90,000 พิกเซล เนื่องจากหน่วยความจำของ โปรแกรมมีขนาดจำกัด

การแยกภาพวัตถุออกมาจากฉากหลังประกอบด้วยขั้นตอนหลักๆ คือ เริ่มด้วยการกำหนดส่วนต่างๆ ของภาพ แล้วจัดกลุ่มค่าสีของพิกเซลในส่วนของภาพวัตถุกับส่วนของฉากหลัง จากนั้นทำการจับคู่ค่าสีระหว่างพิกเซลในส่วนของแนวขอบของภาพวัตถุกับกลุ่มค่าสีในส่วนของภาพวัตถุและกลุ่มค่าสีในส่วนของฉากหลัง ซึ่งจะเป็นข้อมูลสำหรับการคำนวณเพื่อประมาณค่าความโปร่งแสงและค่าสีจริงของพิกเซลในส่วนของแนวขอบของภาพวัตถุ ขั้นตอนสุดท้ายคือ การซ้อนทับภาพวัตถุที่แยกออกมาจากฉากหลังเดิมลงบนฉากหลังใหม่

ภาพตัวอย่างที่นำมาใช้ในการอธิบายการทำงานของโปรแกรมเป็นภาพดอกไม้ที่มีฉากหลังเป็นฉากบลูสกรีน ดังรูปที่ 2.1



รูปที่ 2.1 แสดงตัวอย่างภาพต้นฉบับ

2.3 การกำหนดส่วนต่างๆ ของภาพ

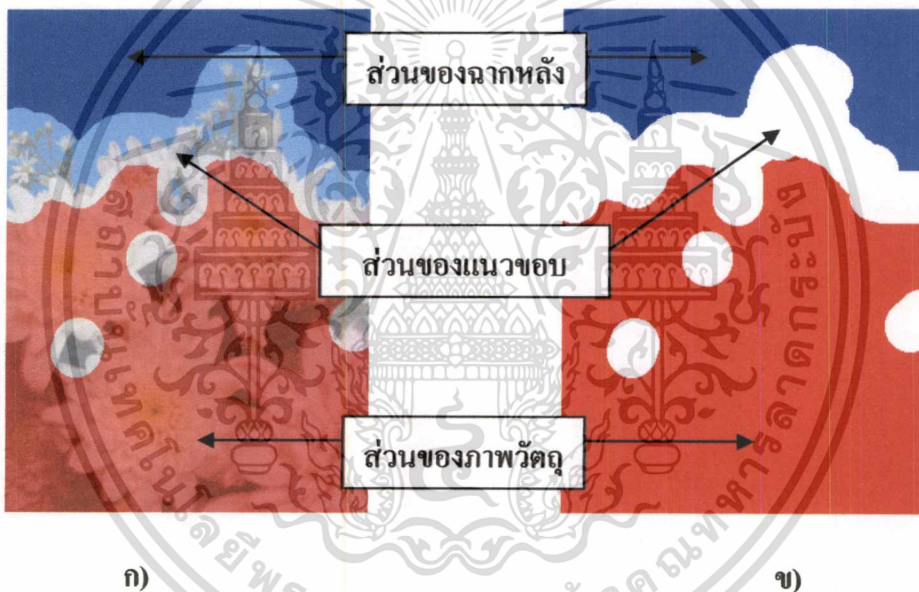
ในการแยกภาพวัตถุออกจากฉากหลัง จะต้องนำค่าสีในส่วนของภาพวัตถุและฉากหลังมาใช้ในการคำนวณสำหรับการประมาณค่าความโปร่งแสงและค่าสีจริงของแต่ละพิกเซลในส่วนของแนวขอบของภาพวัตถุ ดังนั้นจะต้องทำการแบ่งส่วนภาพทั้งหมดออกเป็น 3 ส่วน คือ

1. ส่วนของภาพวัตถุ (Foreground)
2. ส่วนของฉากหลัง (Background)
3. ส่วนของแนวขอบของภาพวัตถุ (Boundary) ซึ่งเป็นบริเวณระหว่างภาพวัตถุกับฉากหลัง

การกำหนดส่วนต่างๆ ของภาพ สามารถทำได้โดยใช้เครื่องมือ Pencil Tool ในโปรแกรม Adobe Photoshop สร้าง Mask ให้กับส่วนต่างๆ ของภาพดังรูปที่ 2.2 ก) และ 2.2 ข) ซึ่งจะกำหนดด้วยค่าสีต่างกัน ดังนี้

1. ส่วนของภาพวัตถุ กำหนดให้ Mask มีสีแดง (255, 0, 0)
2. ส่วนของฉากหลัง กำหนดให้ Mask มีสีน้ำเงิน (0, 0, 255)
3. ส่วนของแนวขอบ กำหนดให้ Mask มีสีขาว (255, 255, 255)

โดยรายละเอียดขั้นตอนวิธีการกำหนดส่วนต่างๆ ของภาพด้วยโปรแกรม Adobe Photoshop จะอธิบายในภาคผนวก ก



ก)

ข)

รูปที่ 2.2 ก) แสดงการกำหนดส่วนต่างๆ ของภาพ

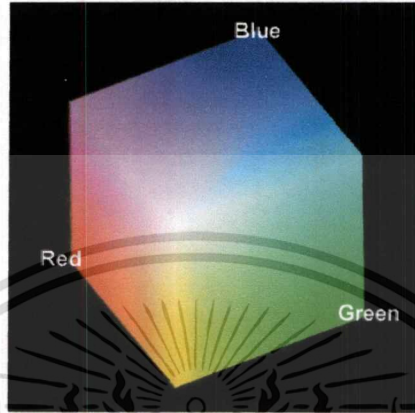
รูปที่ 2.2 ข) แสดงภาพ Mask ที่จะนำไปใช้ในการประมวลผล

2.4 การจัดกลุ่มค่าสี

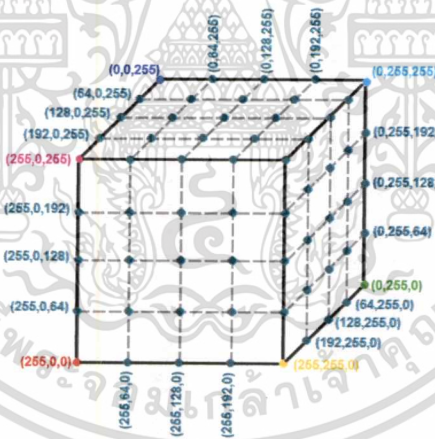
โปรแกรมที่พัฒนาขึ้นในโครงการนี้ จะจำแนกกลุ่มค่าสีของทุกพิกเซลในส่วนของภาพวัตถุและส่วนของฉากหลัง ด้วยเทคนิคการจัดกลุ่มข้อมูลด้วยวิธี K-Means Clustering ซึ่งจะต้องกำหนดจำนวนกลุ่มของกลุ่มข้อมูลเสียก่อน ดังนั้นจึงต้องหาจำนวนกลุ่มที่เหมาะสมด้วย RGB Cube (รูปที่ 2.3) โดยการกำหนดช่วงของกลุ่มค่าสีให้กับ RGB Cube ซึ่งสำหรับโครงการนี้จะแบ่งแต่ละแกนของแม่สีแสง RGB ออกเป็น 4 ช่วง ซึ่งจะได้ช่วงกลุ่มทั้งหมด 64 กลุ่ม ดังรูปที่ 2.4 จากนั้นจัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มค่าสีของแต่ละพิกเซลลงในแต่ละกลุ่มตามค่าที่กำหนดไว้ แล้วนับจำนวนกลุ่มที่มีค่าสีของพิกเซล ก็จะ ได้จำนวนกลุ่มที่เหมาะสมสำหรับการจัดกลุ่มข้อมูลด้วยวิธี K-Means Clustering



รูปที่ 2.3 แสดงภาพของ RGB Cube



รูปที่ 2.4 แสดงช่วงของค่าสีที่แบ่ง RGB Cube ออกเป็น 64 กลุ่ม

2.4.1 การจัดกลุ่มข้อมูลด้วยวิธี K-Means Clustering

การจัดกลุ่มข้อมูลเป็นการแบ่งกลุ่มข้อมูลออกเป็นกลุ่มย่อยๆ ตั้งแต่ 2 กลุ่มขึ้นไป โดยข้อมูลที่อยู่ในกลุ่มเดียวกันจะมีลักษณะคล้ายกันหรือมีค่าใกล้เคียงกัน ส่วนข้อมูลที่อยู่ต่างกลุ่มกันจะมีลักษณะที่แตกต่างกันหรือมีค่าต่างกัน ดังนั้นการพิจารณาเลือกลักษณะหรือปัจจัยที่จะนำมาใช้ในการแบ่งกลุ่มข้อมูลจึงมีความสำคัญ ซึ่งสำหรับ โครงการนี้ลักษณะหรือปัจจัยที่จะนำมาใช้ในการแบ่งกลุ่มข้อมูลจะใช้การคำนวณหาค่ากลางหรือค่า Centroid ของกลุ่มข้อมูลใหม่ทุก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ครั้งที่ข้อมูลถูกจัดเข้ามาเพิ่มในกลุ่ม นอกจากนั้น ข้อมูลใดข้อมูลหนึ่งจะต้องอยู่ในกลุ่มใดกลุ่มหนึ่งเพียงกลุ่มเดียวเท่านั้น

เมื่อจัดกลุ่มค่าสีในแต่ละส่วนแล้วจึงคำนวณหาค่าความแปรปรวน (Variance) ของค่าสีสำหรับแต่ละกลุ่มทั้งในส่วนของภาพวัตถุและส่วนของฉากหลัง ดังสมการ (2.1) เพื่อเป็นตัวแทนค่าสีของแต่ละกลุ่ม สำหรับใช้ในการคำนวณต่อไป

$$Variance(\sigma^2) = \frac{\sum_{i=1}^N (X_i - \bar{X})^2}{N} \quad (2.1)$$

โดยที่

X_i แทนค่าสี RGB ของแต่ละพิกเซล คือ $X_i = [x_r, x_g, x_b]$

\bar{X} แทนค่าเฉลี่ยของแต่ละกลุ่ม คือ $\bar{X} = [\bar{X}_r, \bar{X}_g, \bar{X}_b]$ ซึ่งได้จากการจัดกลุ่มด้วยวิธี K-Means Clustering

N แทนจำนวนพิกเซลในแต่ละกลุ่ม

จะได้ค่าความแปรปรวนของค่าสีในแต่ละกลุ่ม 3 ค่า คือ $[\sigma_r^2, \sigma_g^2, \sigma_b^2]$

2.5 การจับคู่ค่าสีของพิกเซล

ในขั้นตอนนี้จะทำการจับคู่ระหว่างค่าสีของแต่ละพิกเซลในส่วนของแนวขอบของภาพวัตถุกับค่าเฉลี่ยซึ่งเป็นตัวแทนของแต่ละกลุ่มค่าสีในส่วนของภาพวัตถุและจับคู่กับค่าเฉลี่ยซึ่งเป็นตัวแทนของแต่ละกลุ่มค่าสีในส่วนของฉากหลังว่าใกล้เคียงกับค่าสีใดมากที่สุด โดยใช้การคำนวณระยะห่างระหว่างค่าสีของแต่ละพิกเซลด้วยวิธี Euclidean Distance ดังสมการ (2.2) โดยที่ค่าสีของพิกเซลในส่วนของแนวขอบของภาพวัตถุกับค่าเฉลี่ยของกลุ่มสีใดในกลุ่มสีทั้งหมดของส่วนของภาพวัตถุที่มีระยะห่างน้อยที่สุด ก็จะจับคู่กับค่าสีของกลุ่มนั้น การจับคู่กับกลุ่มสีของส่วนของฉากหลังก็เช่นเดียวกัน ดังรูปที่ 2.5

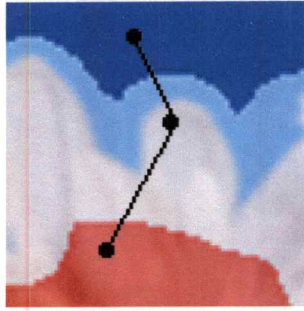
$$Dist(X_i, Y_i) = \sqrt{(X_i - Y_i)^2} \quad (2.2)$$

โดยที่

X_i แทนค่าสี RGB ของพิกเซล คือ $X_i = [x_r, x_g, x_b]$ ของส่วนแนวขอบ

Y_i แทนค่าเฉลี่ยของสี RGB ของแต่ละกลุ่ม คือ $Y_i = [y_r, y_g, y_b]$ ของส่วนของภาพวัตถุกับส่วนของฉากหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แสดงการจับคู่ค่าสีของพิกเซล

2.6 การประมาณค่าความโปร่งแสงของพิกเซล

เมื่อจับคู่ระหว่างค่าสีของพิกเซลในส่วนของแนวขอบกับค่าเฉลี่ยของกลุ่มสีในส่วนของภาพวัตถุและส่วนของฉากหลังแล้ว ก็จะนำค่าของคู่สีที่ได้มาคำนวณเพื่อหาค่าประมาณของค่าความโปร่งแสงและค่าสีจริงของแต่ละพิกเซลในส่วนของแนวขอบ

การประมาณค่าความโปร่งแสงของพิกเซลในส่วนของแนวขอบจะต้องทำการหาค่าความน่าจะเป็นของค่าความโปร่งแสงที่เป็นไปได้ทั้งหมดของพิกเซลนั้นๆ ด้วยสมการของการแจกแจงปกติ ค่าความโปร่งแสงที่มีค่าความน่าจะเป็นสูงสุดของการแจกแจงปกติจะเป็นค่าประมาณของค่าความโปร่งแสงของพิกเซลนั้นๆ โดยกำหนดให้ค่า t เป็นค่าความโปร่งแสงที่เป็นไปได้ทั้งหมดของพิกเซล มีค่าอยู่ระหว่าง 0 ถึง 1 กำหนดความละเอียดเท่ากับ 0.01 และกำหนดให้ฟังก์ชัน $f(t)$ เป็นฟังก์ชันความน่าจะเป็นของค่าความโปร่งแสงที่เป็นไปได้ทั้งหมดสำหรับทุกค่าของ t ซึ่งจะมีการแจกแจงแบบปกติ ดังสมการ (2.3) และ (2.4)

$$f(t) = p_t(c) \quad , 0 \leq t \leq 1 \quad (2.3)$$

$$p_t(c) = \sum_{i=1}^n a_i G_i(c; \mu_i(t), \sigma_i^2(t)) \quad (2.4)$$

โดยที่

- c เป็นเวกเตอร์ของค่าสี RGB ของพิกเซล คือ $c = [c_r, c_g, c_b]$
- n เป็นจำนวนของพิกเซลในกลุ่มนั้น
- a_i เป็นค่าผลคูณจุด (Dot Product) ของค่าเฉลี่ยของกลุ่มสีในส่วนของภาพวัตถุกับฉากหลังซึ่งเป็นคู่กัน ดังสมการ (2.5) โดยที่ค่าเฉลี่ยจะต้องมีค่าอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้ระหว่าง 0 ถึง 1 เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$a_i = \bar{X}_i \cdot \bar{Y}_j = x_{r_i} y_{r_j} + x_{g_i} y_{g_j} + x_{b_i} y_{b_j} \tag{2.5}$$

$G_i(c; \mu_i(t), \sigma_i^2(t))$ เป็นค่าฟังก์ชัน Gaussian distribution ซึ่งสามารถคำนวณได้
 ดังสมการ (2.6)

$$G_i(c; \mu_i(t), \sigma_i^2(t)) = \frac{1}{\sqrt{2\pi\sigma_i^2(t)}} \cdot e^{\frac{-(c-\mu_i(t))^2}{2\sigma_i^2(t)}} \tag{2.6}$$

เมื่อ

$$\mu_i(t) = (1-t)\mu_{j(i)} + t\nu_{k(i)} \tag{2.7}$$

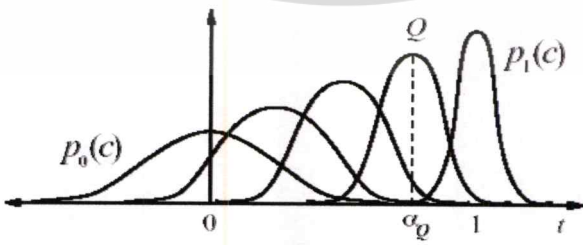
$$\sigma_i^2(t) = (1-t)\sigma_{u_{j(i)}}^2 + t\sigma_{v_{k(i)}}^2 \tag{2.8}$$

โดยที่

$\mu_{j(i)}$ และ $\nu_{k(i)}$ แทนค่าเฉลี่ยของกลุ่มสีในส่วนของภาพวัตถุและฉากหลัง
 ตามลำดับ ดังสมการ (2.7)

$\sigma_{u_{j(i)}}^2$ และ $\sigma_{v_{k(i)}}^2$ แทนค่าความแปรปรวนของค่าสีในแต่ละกลุ่มสีของส่วนของ
 ภาพวัตถุและฉากหลัง ตามลำดับ ดังสมการ (2.8)

เมื่อคำนวณหาค่า $f(t)$ หรือค่าฟังก์ชันความน่าจะเป็นของค่าความโปร่งแสงที่
 เป็นไปได้ทั้งหมดของพิกเซลแล้ว นำค่าที่ได้มาพล็อตกราฟจะได้กราฟของเส้นโค้งปกติ ดังรูปที่ 2.6
 ค่า t ที่ทำให้ค่าฟังก์ชันความน่าจะเป็นมีค่าสูงสุดของโค้งปกติ ก็คือค่าประมาณของค่าความโปร่ง
 แสงของแต่ละพิกเซล ดังรูปที่ 2.7



รูปที่ 2.6 แสดงกราฟของค่าความน่าจะเป็นของค่าความโปร่งแสง



รูปที่ 2.7 แสดงภาพของค่าประมาณของค่าความโปร่งแสงที่คำนวณได้

2.7 การประมาณค่าสีจริงของพิกเซล

เนื่องจากค่าสีของพิกเซลในส่วนของแนวขอบของภาพวัตถุเกิดจากการผสมกันระหว่างค่าสีในส่วนของภาพวัตถุกับค่าสีในส่วนของฉากหลัง ดังนั้นจึงต้องทำการประมาณค่าสีจริงของพิกเซลด้วยสมการ 2.10 ซึ่งได้มาจากสมการของการซ้อนทับภาพ (Matting Equation) ทำให้เราสามารถคำนวณหาค่าสีจริงของพิกเซลที่อยู่ในส่วนของแนวขอบของภาพวัตถุได้

$$C = (1 - \alpha)F + \alpha B \quad (2.9)$$

$$F = \frac{C - \alpha B}{1 - \alpha} \quad (2.10)$$

โดยที่

- C แทนค่าสีที่เกิดจากการผสมกันของค่าสีในส่วนของภาพวัตถุกับฉากหลัง
- B แทนค่าสีของพิกเซลในส่วนของฉากหลัง
- α แทนค่าความโปร่งแสงที่ประมาณได้
- F แทนค่าสีจริงของพิกเซลที่ต้องการประมาณค่า

2.8 การซ้อนทับภาพ

การนำภาพวัตถุที่แยกได้มาซ้อนทับลงบนฉากหลังใหม่ จะต้องนำค่าความโปร่งแสงมาคำนวณร่วมกับค่าประมาณของค่าสีจริงของพิกเซลในส่วนขอบของภาพวัตถุของภาพต้นฉบับและค่าสีของฉากหลังใหม่ ด้วยสมการการซ้อนทับภาพ (Matting Equation) ดังสมการ 2.9 ข้างต้น โดยกำหนดให้

- α เป็นค่าความโปร่งแสงที่ประมาณได้
- F เป็นข้อมูลค่าสีของภาพต้นฉบับและค่าประมาณของค่าสีจริง
- B เป็นข้อมูลค่าสีของภาพพื้นหลังใหม่
- C เป็นภาพผลลัพธ์ที่ได้จากการคำนวณการซ้อนทับภาพ

เมื่อทำการประมวลผลการซ้อนทับภาพวัตถุลงบนฉากหลังใหม่แล้วจะได้ภาพผลลัพธ์ดังรูปที่ 2.8 ซึ่งจะเห็นว่าพิกเซลในส่วนขอบของภาพวัตถุจะมีค่าสีที่ใกล้เคียงกับค่าสีจริงของภาพวัตถุมากที่สุด



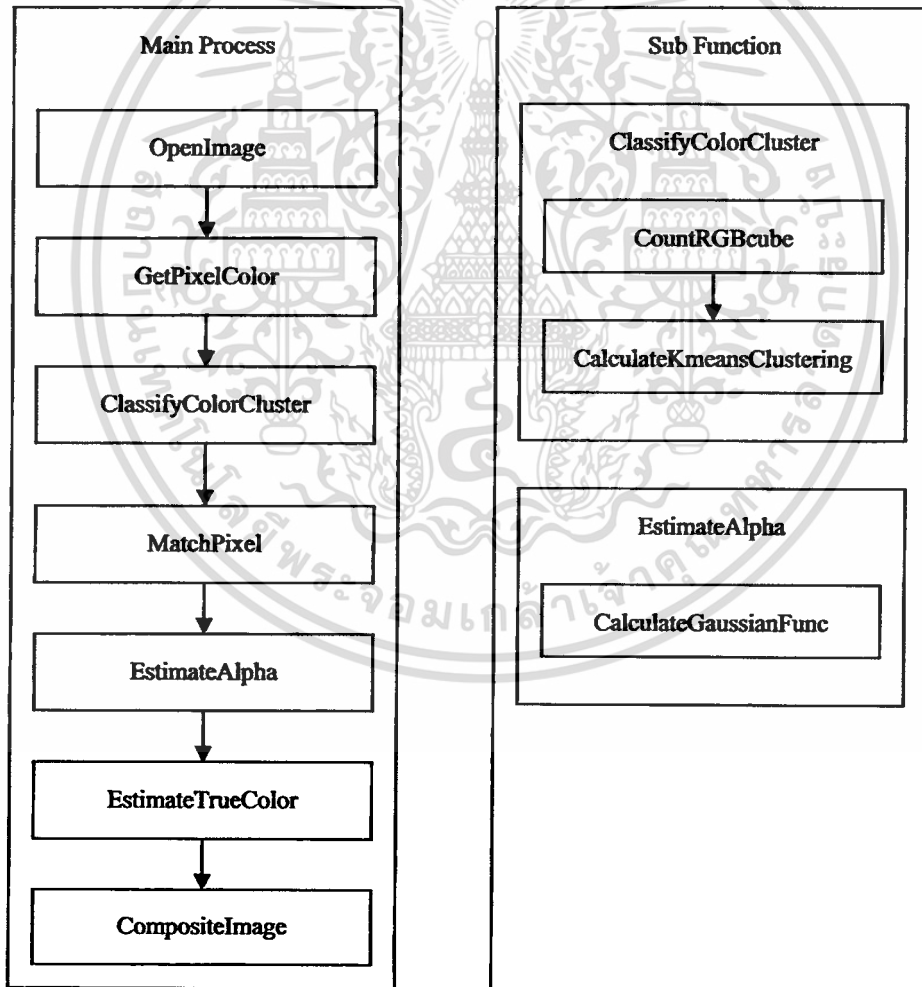
รูปที่ 2.8 แสดงภาพที่ได้จากการซ้อนทับภาพวัตถุลงบนฉากหลังใหม่

บทที่ 3

การทำงานของโปรแกรมและอัลกอริทึม

3.1 โครงสร้างการทำงานของโปรแกรม

โปรแกรมทั้งหมดของโครงการนั้นพัฒนาโดยใช้เครื่องมือ Microsoft Visual Basic โครงสร้างการทำงานของโปรแกรมทั้งหมดแสดงดังรูปที่ 3.1



รูปที่ 3.1 โครงสร้างทั้งหมดของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การทำงานของโปรแกรมและอัลกอริทึม

การทำงานของโปรแกรมจะแบ่งออกเป็นขั้นตอนการทำงานหลักๆ คือ

1. การเปิดไฟล์ภาพ
2. การเก็บข้อมูลค่าสีของพิกเซล
3. การจัดกลุ่มข้อมูลค่าสีของพิกเซล
 - 3.1 การหาจำนวนกลุ่มข้อมูลที่เหมาะสมด้วย RGB cube
 - 3.2 การจัดกลุ่มข้อมูลด้วยวิธี K-Means Clustering
4. การจับคู่ค่าสีของพิกเซล
5. การประมาณค่าความโปร่งแสงของพิกเซล
 - 5.1 การคำนวณค่าความน่าจะเป็นทั้งหมดของค่าความโปร่งแสง
6. การประมาณค่าสีจริงของพิกเซล
7. การซ้อนทับภาพวัตถุลงบนฉากหลังใหม่

3.2.1 การเปิดไฟล์ภาพ

เริ่มต้นการทำงานของโปรแกรมด้วยการเปิดไฟล์ภาพ 3 ภาพ คือ ไฟล์ภาพต้นฉบับที่ต้องการแยกภาพวัตถุออกมา ไฟล์ภาพ Mask ซึ่งกำหนดพื้นที่ส่วนต่างๆ ของภาพด้วยโปรแกรม Adobe Photoshop และ ไฟล์ภาพฉากหลังใหม่ที่ต้องการนำภาพวัตถุซ้อนทับลงไป

ตารางที่ 3.1 แสดงอัลกอริทึมของ OpenImage

```

...// OpenImage
open(original image file); ' เปิดไฟล์ภาพต้นฉบับ
OrgImg = read(original image); ' เก็บข้อมูลภาพต้นฉบับ
open(mask image file); ' เปิดไฟล์ภาพ Mask
MaskImg = read(mask image); % เก็บข้อมูลภาพ Mask
open(new background image file); ' เปิดไฟล์ภาพฉากหลังใหม่
NewBackImg = read(new background image); ' เก็บข้อมูลภาพฉากหลังใหม่
...

```

3.2.2 การเก็บข้อมูลค่าสีของพิกเซล

การเก็บข้อมูลค่าสีจะเก็บข้อมูลตามส่วนต่างๆ ของภาพ ซึ่งกำหนดโดยค่าสีของแต่ละพิกเซลของภาพ Mask เพื่อแบ่งข้อมูลออกเป็น 3 ส่วน ประกอบด้วย ส่วนของภาพวัตถุ ส่วนของฉากหลัง และส่วนของแนวขอบของภาพวัตถุ ดังที่ได้กล่าวมาแล้ว ซึ่งจะนำค่าสีของพิกเซลซึ่งอยู่ในส่วนของแนวขอบของภาพวัตถุและส่วนของฉากหลังไปคำนวณเพื่อประมาณค่าความโปร่งแสงและค่าสีจริง เพื่อให้ได้ค่าประมาณของค่าความโปร่งแสงและค่าสีจริงที่ใกล้เคียงค่าจริงมากที่สุด พร้อมทั้งการกำหนดค่าความโปร่งแสงให้พิกเซลซึ่งอยู่ในส่วนของภาพวัตถุให้เท่ากับ 1 และในส่วนของฉากหลังให้เท่ากับ 0 มีดังนี้

ตารางที่ 3.2 แสดงอัลกอริทึมของ GetPixelColor

```

...// GetPixelColor
for each i, j
    ' เก็บข้อมูลสีของพิกเซลในส่วนของภาพวัตถุ
    if (MaskImg(i, j, 1) == 255) & (MaskImg(i, j, 2) == 0) & (MaskImg(i, j, 3) == 0)
        n1 = n1 + 1 ' จำนวนพิกเซลในส่วนของภาพวัตถุ
        AlphaImage(i, j) = 1 ' กำหนดค่าความโปร่งแสงของพิกเซลในส่วนของภาพวัตถุ
        DataForeground = GetImageData
    ' เก็บข้อมูลสีของพิกเซลในส่วนของฉากหลัง
    elseif (MaskImg(i, j, 1) == 0) & (MaskImg(i, j, 2) == 0) & (MaskImg(i, j, 3) == 255)
        n2 = n2 + 1 ' จำนวนพิกเซลในส่วนของฉากหลัง
        AlphaImage(i, j) = 0 ' กำหนดค่าความโปร่งแสงของพิกเซลในส่วนของฉากหลัง
        DataBackground = GetImageData
    ' เก็บข้อมูลสีของพิกเซลในส่วนแนวขอบของวัตถุ
    elseif (MaskImg(i, j, 1) == 255) & (MaskImg(i, j, 2) == 255) & (MaskImg(i, j, 3) == 255)
        n3 = n3 + 1 ' จำนวนพิกเซลในส่วนแนวขอบของวัตถุ
        DataBoundary = GetImageData
    ...

```

3.2.3 การจัดกลุ่มค่าสีของพิกเซล

การจัดกลุ่มค่าสีประกอบด้วยฟังก์ชันการทำงาน 2 ส่วน คือ การหาจำนวนกลุ่มข้อมูลที่เหมาะสมด้วยช่วงกลุ่มย่อยของ RGB cube 64 ช่วงกลุ่ม และการจัดกลุ่มข้อมูลด้วยวิธี K-Means Clustering ซึ่งจะทำให้การจัดกลุ่มข้อมูลค่าสีใน 2 ส่วน คือ ส่วนของภาพวัตถุ และส่วนของฉากหลัง

ตารางที่ 3.3 แสดงอัลกอริทึมของ RGBcubeCluster

```

...// RGBcubeCluster
For each i
  R = ImageData(i, r)
  G = ImageData(i, g)
  B = ImageData(i, b)
If (R >= 0) And (R < 64) And (G >= 0) And (G < 64) And (B >= 0) And (B < 64)
  nCube(1, 1) = nCube(1, 1) + 1
ElseIf (R >= 0) And (R < 64) And (G >= 0) And (G < 64) And (B >= 64) And (B < 128)
  nCube(1, 2) = nCube(1, 2) + 1
ElseIf (R >= 0) And (R < 64) And (G >= 0) And (G < 64) And (B >= 128) And (B < 192)
  nCube(1, 3) = nCube(1, 3) + 1
ElseIf (R >= 0) And (R < 64) And (G >= 0) And (G < 64) And (B >= 192) And (B <= 255)
  nCube(1, 4) = nCube(1, 4) + 1
ElseIf (R >= 0) And (R < 64) And (G >= 64) And (G < 128) And (B >= 0) And (B < 64)
  nCube(1, 5) = nCube(1, 5) + 1
ElseIf (R >= 0) And (R < 64) And (G >= 64) And (G < 128) And (B >= 64) And (B < 128)
  nCube(1, 6) = nCube(1, 6) + 1
ElseIf (R >= 0) And (R < 64) And (G >= 64) And (G < 128) And (B >= 128) And (B < 192)
  nCube(1, 7) = nCube(1, 7) + 1
ElseIf (R >= 0) And (R < 64) And (G >= 64) And (G < 128) And (B >= 192) And (B <= 255)
  nCube(1, 8) = nCube(1, 8) + 1
ElseIf (R >= 0) And (R < 64) And (G >= 128) And (G < 192) And (B >= 0) And (B < 64)

```

```

nCube(1, 9) = nCube(1, 9) + 1
Elseif (R >= 0) And (R < 64) And (G >= 128) And (G < 192) And (B >= 64) And (B < 128)
nCube(1, 10) = nCube(1, 10) + 1
Elseif (R >= 0) And (R < 64) And (G >= 128) And (G < 192) And (B >= 128) And (B < 192)
nCube(1, 11) = nCube(1, 11) + 1
Elseif (R >= 0) And (R < 64) And (G >= 128) And (G < 192) And (B >= 192) And (B <= 255)
nCube(1, 12) = nCube(1, 12) + 1
Elseif (R >= 0) And (R < 64) And (G >= 192) And (G <= 255) And (B >= 0) And (B < 64)
nCube(1, 13) = nCube(1, 13) + 1
Elseif (R >= 0) And (R < 64) And (G >= 192) And (G <= 255) And (B >= 64) And (B < 128)
nCube(1, 14) = nCube(1, 14) + 1
Elseif (R >= 0) And (R < 64) And (G >= 192) And (G <= 255) And (B >= 128) And (B < 192)
nCube(1, 15) = nCube(1, 15) + 1
Elseif (R >= 0) And (R < 64) And (G >= 192) And (G <= 255) And (B >= 192) And (B <= 255)
nCube(1, 16) = nCube(1, 16) + 1
Elseif (R >= 64) And (R < 128) And (G >= 0) And (G < 64) And (B >= 0) And (B < 64)
nCube(1, 17) = nCube(1, 17) + 1
Elseif (R >= 64) And (R < 128) And (G >= 0) And (G < 64) And (B >= 64) And (B < 128)
nCube(1, 18) = nCube(1, 18) + 1
Elseif (R >= 64) And (R < 128) And (G >= 0) And (G < 64) And (B >= 128) And (B < 192)
nCube(1, 19) = nCube(1, 19) + 1
Elseif (R >= 64) And (R < 128) And (G >= 0) And (G < 64) And (B >= 192) And (B <= 255)
nCube(1, 20) = nCube(1, 20) + 1
Elseif (R >= 64) And (R < 128) And (G >= 64) And (G < 128) And (B >= 0) And (B < 64)
nCube(1, 21) = nCube(1, 21) + 1
Elseif (R >= 64) And (R < 128) And (G >= 64) And (G < 128) And (B >= 64) And (B < 128)
nCube(1, 22) = nCube(1, 22) + 1
Elseif (R >= 64) And (R < 128) And (G >= 64) And (G < 128) And (B >= 128) And (B < 192)
nCube(1, 23) = nCube(1, 23) + 1
Elseif (R >= 64) And (R < 128) And (G >= 64) And (G < 128) And (B >= 192) And (B <= 255)

```

$$nCube(1, 24) = nCube(1, 24) + 1$$

Elseif (R >= 64) And (R < 128) And (G >= 128) And (G < 192) And (B >= 0) And (B < 64)

$$nCube(1, 25) = nCube(1, 25) + 1$$

Elseif (R >= 64) And (R < 128) And (G >= 128) And (G < 192) And (B >= 64) And (B < 128)

$$nCube(1, 26) = nCube(1, 26) + 1$$

Elseif (R >= 64) And (R < 128) And (G >= 128) And (G < 192) And (B >= 128) And (B < 192)

$$nCube(1, 27) = nCube(1, 27) + 1$$

Elseif (R >= 64) And (R < 128) And (G >= 128) And (G < 192) And (B >= 192) And (B <= 255)

$$nCube(1, 28) = nCube(1, 28) + 1$$

Elseif (R >= 64) And (R < 128) And (G >= 192) And (G <= 255) And (B >= 0) And (B < 64)

$$nCube(1, 29) = nCube(1, 29) + 1$$

Elseif (R >= 64) And (R < 128) And (G >= 192) And (G <= 255) And (B >= 64) And (B < 128)

$$nCube(1, 30) = nCube(1, 30) + 1$$

Elseif (R >= 64) And (R < 128) And (G >= 192) And (G <= 255) And (B >= 128) And (B < 192)

$$nCube(1, 31) = nCube(1, 31) + 1$$

Elseif (R >= 64) And (R < 128) And (G >= 192) And (G <= 255) And (B >= 192) And (B <= 255)

$$nCube(1, 32) = nCube(1, 32) + 1$$

Elseif (R >= 128) And (R < 192) And (G >= 0) And (G < 64) And (B >= 0) And (B < 64)

$$nCube(1, 33) = nCube(1, 33) + 1$$

Elseif (R >= 128) And (R < 192) And (G >= 0) And (G < 64) And (B >= 64) And (B < 128)

$$nCube(1, 34) = nCube(1, 34) + 1$$

Elseif (R >= 128) And (R < 192) And (G >= 0) And (G < 64) And (B >= 128) And (B < 192)

$$nCube(1, 35) = nCube(1, 35) + 1$$

Elseif (R >= 128) And (R < 192) And (G >= 0) And (G < 64) And (B >= 192) And (B <= 255)

$$nCube(1, 36) = nCube(1, 36) + 1$$

Elseif (R >= 128) And (R < 192) And (G >= 64) And (G < 128) And (B >= 0) And (B < 64)

$$nCube(1, 37) = nCube(1, 37) + 1$$

Elseif (R >= 128) And (R < 192) And (G >= 64) And (G < 128) And (B >= 64) And (B < 128)

$$nCube(1, 38) = nCube(1, 38) + 1$$

Elseif (R >= 128) And (R < 192) And (G >= 64) And (G < 128) And (B >= 128) And (B < 192)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การเชิงพาณิชย์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ทางการ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$nCube(1, 39) = nCube(1, 39) + 1$
ElseIf ($R \geq 128$) *And* ($R < 192$) *And* ($G \geq 64$) *And* ($G < 128$) *And* ($B \geq 192$) *And* ($B \leq 255$)
 $nCube(1, 40) = nCube(1, 40) + 1$
ElseIf ($R \geq 128$) *And* ($R < 192$) *And* ($G \geq 128$) *And* ($G < 192$) *And* ($B \geq 0$) *And* ($B < 64$)
 $nCube(1, 41) = nCube(1, 41) + 1$
ElseIf ($R \geq 128$) *And* ($R < 192$) *And* ($G \geq 128$) *And* ($G < 192$) *And* ($B \geq 64$) *And* ($B < 128$)
 $nCube(1, 42) = nCube(1, 42) + 1$
ElseIf ($R \geq 128$) *And* ($R < 192$) *And* ($G \geq 128$) *And* ($G < 192$) *And* ($B \geq 128$) *And* ($B < 192$)
 $nCube(1, 43) = nCube(1, 43) + 1$
ElseIf ($R \geq 128$) *And* ($R < 192$) *And* ($G \geq 128$) *And* ($G < 192$) *And* ($B \geq 192$) *And* ($B \leq 255$)
 $nCube(1, 44) = nCube(1, 44) + 1$
ElseIf ($R \geq 128$) *And* ($R < 192$) *And* ($G \geq 192$) *And* ($G \leq 255$) *And* ($B \geq 0$) *And* ($B < 64$)
 $nCube(1, 45) = nCube(1, 45) + 1$
ElseIf ($R \geq 128$) *And* ($R < 192$) *And* ($G \geq 192$) *And* ($G \leq 255$) *And* ($B \geq 64$) *And* ($B < 128$)
 $nCube(1, 46) = nCube(1, 46) + 1$
ElseIf ($R \geq 128$) *And* ($R < 192$) *And* ($G \geq 192$) *And* ($G \leq 255$) *And* ($B \geq 128$) *And* ($B < 192$)
 $nCube(1, 47) = nCube(1, 47) + 1$
ElseIf ($R \geq 128$) *And* ($R < 192$) *And* ($G \geq 192$) *And* ($G \leq 255$) *And* ($B \geq 192$) *And* ($B \leq 255$)
 $nCube(1, 48) = nCube(1, 48) + 1$
ElseIf ($R \geq 192$) *And* ($R \leq 255$) *And* ($G \geq 0$) *And* ($G < 64$) *And* ($B \geq 0$) *And* ($B < 64$)
 $nCube(1, 49) = nCube(1, 49) + 1$
ElseIf ($R \geq 192$) *And* ($R \leq 255$) *And* ($G \geq 0$) *And* ($G < 64$) *And* ($B \geq 64$) *And* ($B < 128$)
 $nCube(1, 50) = nCube(1, 50) + 1$
ElseIf ($R \geq 192$) *And* ($R \leq 255$) *And* ($G \geq 0$) *And* ($G < 64$) *And* ($B \geq 128$) *And* ($B < 192$)
 $nCube(1, 51) = nCube(1, 51) + 1$
ElseIf ($R \geq 192$) *And* ($R \leq 255$) *And* ($G \geq 0$) *And* ($G < 64$) *And* ($B \geq 192$) *And* ($B \leq 255$)
 $nCube(1, 52) = nCube(1, 52) + 1$
ElseIf ($R \geq 192$) *And* ($R \leq 255$) *And* ($G \geq 64$) *And* ($G < 128$) *And* ($B \geq 0$) *And* ($B < 64$)
 $nCube(1, 53) = nCube(1, 53) + 1$
ElseIf ($R \geq 192$) *And* ($R \leq 255$) *And* ($G \geq 64$) *And* ($G < 128$) *And* ($B \geq 64$) *And* ($B < 128$)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการศึกษา

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$nCube(1, 54) = nCube(1, 54) + 1$$

Elseif (R >= 192) And (R <= 255) And (G >= 64) And (G < 128) And (B >= 128) And (B < 192)

$$nCube(1, 55) = nCube(1, 55) + 1$$

Elseif (R >= 192) And (R <= 255) And (G >= 64) And (G < 128) And (B >= 192) And (B <= 255)

$$nCube(1, 56) = nCube(1, 56) + 1$$

Elseif (R >= 192) And (R <= 255) And (G >= 128) And (G < 192) And (B >= 0) And (B < 64)

$$nCube(1, 57) = nCube(1, 57) + 1$$

Elseif (R >= 192) And (R <= 255) And (G >= 128) And (G < 192) And (B >= 64) And (B < 128)

$$nCube(1, 58) = nCube(1, 58) + 1$$

Elseif (R >= 192) And (R <= 255) And (G >= 128) And (G < 192) And (B >= 128) And (B < 192)

$$nCube(1, 59) = nCube(1, 59) + 1$$

Elseif (R >= 192) And (R <= 255) And (G >= 128) And (G < 192) And (B >= 192) And (B <= 255)

$$nCube(1, 60) = nCube(1, 60) + 1$$

Elseif (R >= 192) And (R <= 255) And (G >= 192) And (G <= 255) And (B >= 0) And (B < 64)

$$nCube(1, 61) = nCube(1, 61) + 1$$

Elseif (R >= 192) And (R <= 255) And (G >= 192) And (G <= 255) And (B >= 64) And (B < 128)

$$nCube(1, 62) = nCube(1, 62) + 1$$

Elseif (R >= 192) And (R <= 255) And (G >= 192) And (G <= 255) And (B >= 128) And (B < 192)

$$nCube(1, 63) = nCube(1, 63) + 1$$

Elseif (R >= 192) And (R <= 255) And (G >= 192) And (G <= 255) And (B >= 192) And (B <= 255)

$$nCube(1, 64) = nCube(1, 64) + 1$$

For each j

If $nCube(1, j) > 0$ *Then* Cluster = Cluster + 1 ' นับจำนวนกลุ่ม

...

การจัดกลุ่มข้อมูลด้วยวิธี K-Means Clustering เหมาะสำหรับการจัดกลุ่มข้อมูลที่มีข้อมูลเป็นจำนวนมาก ($n > 200$) โดยจะต้องกำหนดจำนวนกลุ่มที่ต้องการก่อน เทคนิค K-Means Clustering จะมีการทำงานหลายๆ รอบ (Iterations) โดยในแต่ละรอบจะมีการจัดรวมข้อมูลให้ไปอยู่ในกลุ่มที่ข้อมูลนั้นมีระยะห่างจากค่ากลางของกลุ่มน้อยที่สุด แล้วคำนวณค่ากลางของกลุ่มใหม่ จะทำเช่นนี้จนกระทั่งค่ากลางของกลุ่มไม่เปลี่ยนแปลง หรือครบจำนวนรอบที่กำหนดไว้ ซึ่งวิธี K-Means Clustering จะคำนวณหาระยะห่างโดยวิธี Euclidean Distance

ตารางที่ 3.4 แสดงอัลกอริทึมของ CalculateKmeansClustering

```

...// CalculateKmeansClustering
isStillMoving = True
If totalData <= numCluster Then
    Data(totalData, 0) = totalData ' cluster No.
    Centroid(totalData, 1) = Data(totalData, 1) ' R
    Centroid(totalData, 2) = Data(totalData, 2) ' G
    Centroid(totalData, 3) = Data(totalData, 3) ' B
Else
    ' calculate minimum distance to assign the new data
    R = Data(totalData, 1)
    G = Data(totalData, 2)
    B = Data(totalData, 3)
    Cluster = minDistance(R, G, B, Centroid, numCluster)
    Data(totalData, 0) = Cluster
Do While isStillMoving
    ReDim sumRGB(1 To numCluster, 1 To 4)
    For i = 1 To totalData
        sumRGB(Data(i, 0), 1) = sumRGB(Data(i, 0), 1) + Data(i, 1)
        sumRGB(Data(i, 0), 2) = sumRGB(Data(i, 0), 2) + Data(i, 2)
        sumRGB(Data(i, 0), 3) = sumRGB(Data(i, 0), 3) + Data(i, 3)
        sumRGB(Data(i, 0), 4) = sumRGB(Data(i, 0), 4) + 1

```

```

For j = 1 To numCluster
    n = sumRGB(j, 4)
    If n <> 0 Then
        Centroid(j, 1) = sumRGB(j, 1) / n
        Centroid(j, 2) = sumRGB(j, 2) / n
        Centroid(j, 3) = sumRGB(j, 3) / n
    ' assign all data to the new centroids
isStillMoving = False
For i = 1 To totalData
    R = Data(i, 1)
    G = Data(i, 2)
    B = Data(i, 3)
    Cluster = minDistance(R, G, B, Centroid, numCluster)
    If Data(i, 0) <> Cluster Then
        Data(i, 0) = Cluster
        isStillMoving = True
...

```

หลังจากการจัดกลุ่มข้อมูลด้วยวิธี K-Means Clustering จะได้ค่าเฉลี่ยของค่าสีของแต่ละกลุ่มข้อมูล ซึ่งจะใช้เป็นตัวแทนค่าสีสำหรับการประมาณค่าความโปร่งแสงและค่าสีจริงของพิกเซลในส่วนของแนวขอบของภาพวัตถุ และต้องคำนวณหาค่าความแปรปรวนของค่าสีของพิกเซลในแต่ละกลุ่มด้วย

3.2.4 การจับคู่ค่าสีของพิกเซล

การคำนวณเพื่อประมาณค่าความโปร่งแสงและค่าสีจริงของพิกเซลในส่วนของแนวขอบของภาพวัตถุ จะต้องนำข้อมูลค่าสีของพิกเซลในส่วนของภาพวัตถุและส่วนของฉากหลังมาใช้ในการคำนวณด้วย ดังนั้นจึงต้องทำการจับคู่ระหว่างค่าสีของแต่ละพิกเซลในส่วนของแนวขอบของภาพวัตถุกับตัวแทนค่าสีของกลุ่มข้อมูลในส่วนของภาพวัตถุ และระหว่างค่าสีของแต่ละพิกเซลในส่วนของแนวขอบของภาพวัตถุกับตัวแทนค่าสีของกลุ่มข้อมูลในส่วนของฉากหลัง ซึ่ง

การจับคู่ค่าสีทำได้โดยการคำนวณหาค่าระยะห่างระหว่างค่าสีของพิกเซลด้วยวิธี Euclidean Distance ที่มีค่าน้อยที่สุด เพื่อให้ได้คู่ของค่าสีที่มีค่าสีใกล้เคียงกันมากที่สุด

ตารางที่ 3.5 แสดงอัลกอริทึมของ MatchPixel

```

...// MatchPixel
For each i
    DistFore = minDistance(UnknownData(i), ForeCluster, numForeCluster)
    MeanFore = ForeCluster(DistFore)
    VarFore = VarForeCluster(DistFore)
    DistBack = minDistance(UnknownData(i), BackCluster, numBackCluster)
    MeanBack = BackCluster(DistBack)
    VarBack = VarBackCluster(DistBack)
...

```

ตารางที่ 3.6 แสดงอัลกอริทึมของ minDistance

```

...// minDistance
For each i
    R = (Data(l, r) - Ref(r)) ^ 2
    G = (Data(l, g) - Ref(g)) ^ 2
    B = (Data(l, b) - Ref(b)) ^ 2
    d = Sqr(R + G + B)
    If d < min Then
        min = d
minDistance = i
...

```

3.2.5 การประมาณค่าความโปร่งแสงของพิกเซล

การประมาณค่าความโปร่งแสงของพิกเซลทำได้โดยการคำนวณหาค่าความน่าจะเป็นที่เป็นไปได้ทั้งหมดของค่าความโปร่งแสงซึ่งกำหนดด้วยค่า T ที่มีค่าอยู่ระหว่าง 0 ถึง 1 และมีความละเอียดเท่ากับ 0.01 ด้วยสมการของการแจกแจงปกติหรือ Gaussian Distribution และค่าประมาณของค่าความโปร่งแสงของพิกเซลคือ ค่า T ที่ทำให้ค่าความน่าจะเป็นของค่าความโปร่งแสงมีค่าสูงสุด

ตารางที่ 3.7 แสดงอัลกอริทึมของ EstimateAlpha

```

...// EstimateAlpha
For T = 0.01 To 0.99 Step 0.01
    aMean = (1 - T) * mFore+ (T) * mBack
    aVar = (1 - T) * vFore+ (T) * vBack
    a = DotProductColor(mFore, mBack)
    G = a * CalculateGaussian(aMean, aVar, X)
    If (G > maxProp) Then
        maxProp = G
        Alpha = T
EstimateAlpha = Alpha
...

```

ตารางที่ 3.8 แสดงอัลกอริทึมของ CalculateGaussianFunc

```

...// CalculateGaussianFunc
pi = 3.141593
g1 = 1 / Sqr(2 * pi * Var)
x1 = (X - Mean) ^ 2
g2 = ((-1) * x1) / (2 * Var)
Gaussian = g1 * Exp(g2)
...

```

3.2.6 การประมาณค่าสีจริงของพิกเซล

เนื่องจากข้อมูลค่าสีของพิกเซลในส่วนของแนวขอบของภาพวัตถุเกิดจากการผสมกันระหว่างค่าสีในส่วนของภาพวัตถุกับค่าสีในส่วนของฉากหลัง ดังนั้นจึงต้องทำการประมาณค่าสีจริงของพิกเซลด้วยสมการ $F = \frac{(C - \alpha B)}{1 - \alpha}$ ซึ่งได้มาจากสมการของการซ้อนทับภาพ (Matting Equation)

ตารางที่ 3.9 แสดงอัลกอริทึมของ EstimateTrueColor

```
...// EstimateTrueColor
    T = Alpha * B
    TrueColor = (C - T) / (1 - Alpha)
...
```

3.2.7 การซ้อนทับภาพวัตถุลงบนฉากหลังใหม่

เมื่อคำนวณค่าประมาณของค่าความโปร่งแสงและค่าสีจริงของพิกเซลในส่วนของแนวขอบของภาพวัตถุแล้ว จะนำค่าประมาณที่ได้มาคำนวณด้วยสมการของการซ้อนทับภาพ (Matting Equation) คือ $C = (1 - \alpha)F + \alpha B$

ตารางที่ 3.10 แสดงอัลกอริทึมของ CompositeImage

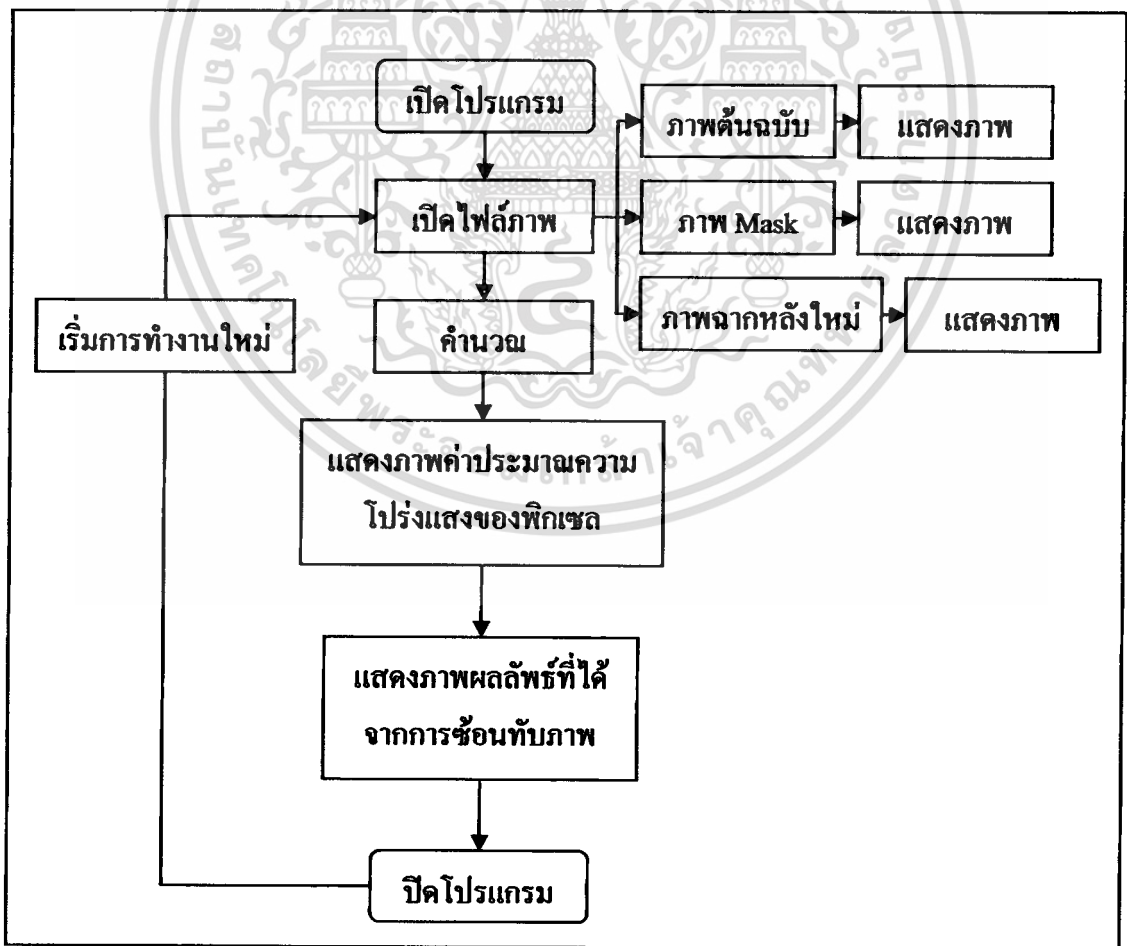
```
...// CompositeImage
    CompositeImage = ((1 - Alpha) * OrgImage) + (Alpha * NewImg)
...
```

บทที่ 4

การออกแบบ User-interface และการใช้งานโปรแกรม

4.1 การออกแบบ User-interface

User-interface หรือส่วนติดต่อกับผู้ใช้ คือส่วนรับข้อมูลจากผู้ใช้โปรแกรม ซึ่งออกแบบให้มีลักษณะเป็น window เนื่องจากผู้ใช้งานจะมีความคุ้นเคยกับ โปรแกรมการทำงานลักษณะนี้อยู่แล้ว สำหรับ โปรแกรมนี้จะติดต่อกับผู้ใช้โปรแกรมผ่านเมาส์ด้วยการคลิกเมาส์ที่ปุ่มการทำงานต่างๆ ของโปรแกรมตามลำดับการทำงานของโปรแกรม โดยการออกแบบ User-interface สำหรับการใช้งานโปรแกรมแสดงดังรูปที่ 4.1



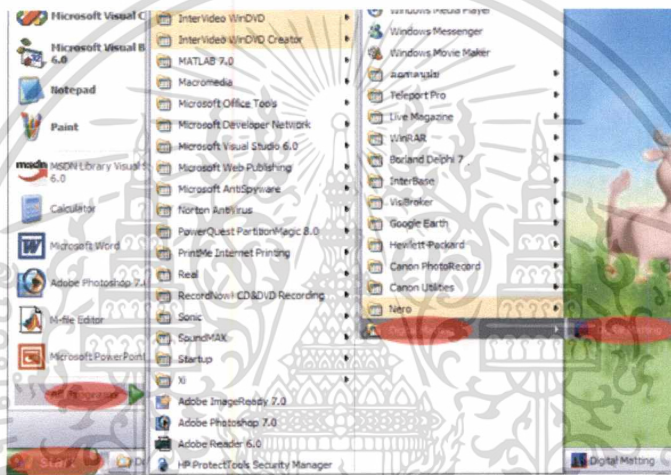
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ 4.1 แสดงแผนภาพการออกแบบ User-interface โปรแกรมที่นำเสนอไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การใช้งานโปรแกรม

หลังจากติดตั้งโปรแกรม Digital Matting ตามขั้นตอนในภาคผนวก ข แล้วก็เปิดโปรแกรม Digital Matting เพื่อเริ่มการทำงานของโปรแกรม

4.2.1 การเปิดโปรแกรม

เปิดโปรแกรม Digital Matting โดยการคลิกที่ปุ่ม Start > All Program > Digital Matting ดังรูปที่ 4.2 และจะได้โปรแกรมซึ่งมีลักษณะดังรูปที่ 4.3



รูปที่ 4.2 แสดงการเปิดโปรแกรม

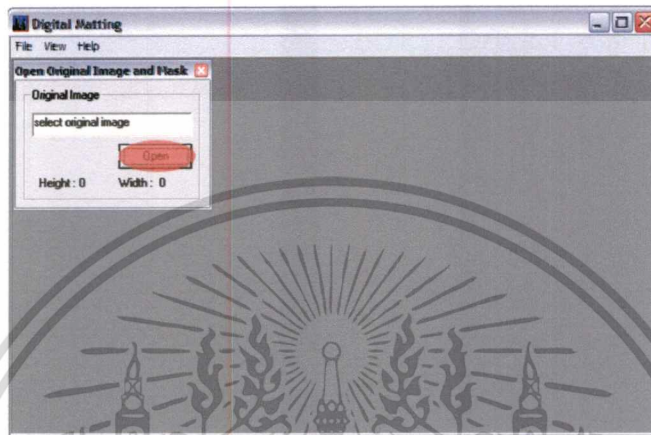


รูปที่ 4.3 แสดงลักษณะของโปรแกรม Digital Matting

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การเปิดไฟล์ภาพ

เมื่อเปิดโปรแกรม Digital Matting ขึ้นมา จะพบหน้าต่างของการเปิดไฟล์ภาพ ต้นฉบับและภาพ Mask ให้คลิกที่ปุ่ม Open เพื่อเปิดไฟล์ภาพ ดังรูปที่ 4.4



รูปที่ 4.4 แสดงหน้าต่างการเปิดไฟล์ภาพต้นฉบับ

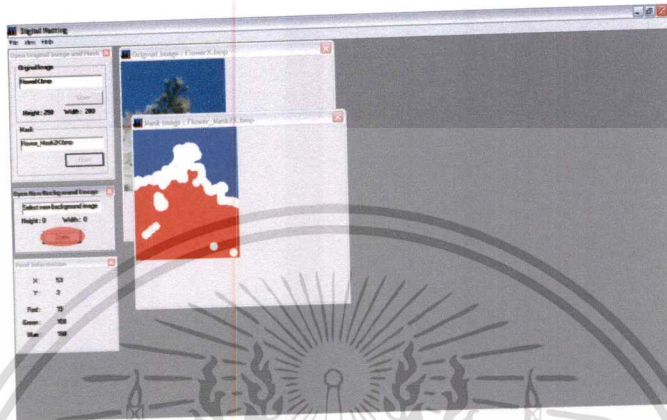
เมื่อหน้าต่างแสดงภาพต้นฉบับแสดงไฟล์ภาพแล้ว จะพบปุ่ม Open ของการเปิดไฟล์ภาพ Mask แสดงขึ้นมา ให้คลิกที่ปุ่ม Open เพื่อเปิดไฟล์ภาพ Mask ดังรูปที่ 4.5



รูปที่ 4.5 แสดงปุ่ม Open ของการเปิดไฟล์ภาพ Mask

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่างแสดงภาพ Mask และหน้าต่างการเปิดไฟล์ภาพจากหลังใหม่จะแสดง ขึ้นมาดังรูปที่ 4.6 ให้คลิกที่ปุ่ม Open เพื่อเปิดไฟล์ภาพจากหลังใหม่ จะได้หน้าต่างแสดงไฟล์ภาพ จากหลังใหม่ ดังรูปที่ 4.7



รูปที่ 4.6 แสดงหน้าต่างของไฟล์ภาพ Mask และปุ่ม Open ของการเปิดไฟล์ภาพจากหลังใหม่

4.2.3 การเริ่มฟังก์ชันการคำนวณของโปรแกรม

หลังจากเปิดไฟล์ภาพจากหลังใหม่แล้ว จะพบปุ่ม Calculate แสดงขึ้นมา ให้คลิก เพื่อคำนวณฟังก์ชันการทำงานต่างๆ ทั้งหมดของ โปรแกรม ดังรูปที่ 4.7



รูปที่ 4.7 แสดงหน้าต่างของภาพจากหลังใหม่และปุ่ม Calculate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

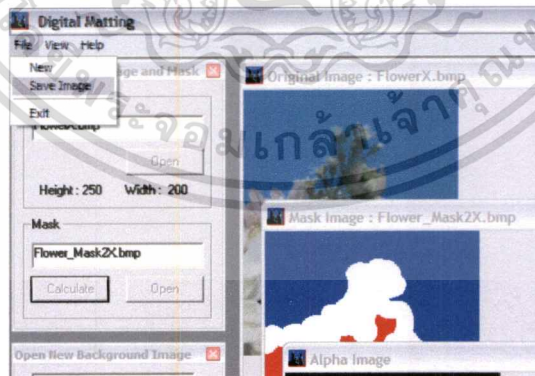
หลังจากคลิกปุ่ม Calculate แล้ว โปรแกรมจะทำการคำนวณฟังก์ชันการทำงานต่างๆ ภายในโปรแกรม เมื่อคำนวณเรียบร้อยแล้วจะแสดงภาพค่าประมาณความโปร่งแสงของภาพต้นฉบับและภาพผลลัพธ์ที่ได้จากการซ้อนทับภาพในหน้าต่างของโปรแกรกดังรูปที่ 4.8



รูปที่ 4.8 แสดงหน้าต่างแสดงภาพค่าประมาณความโปร่งแสงและภาพผลลัพธ์จากการซ้อนทับภาพ

4.2.4 การบันทึกไฟล์ภาพ

ผู้ใช้งานสามารถบันทึกไฟล์ภาพที่ต้องการได้โดยคลิกเลือกที่เมนู File > Save Image ดังรูปที่ 4.9 แล้ว โปรแกรมจะบันทึกข้อมูลของไฟล์ภาพในรูปแบบของไฟล์นามสกุล .bmp เท่านั้น

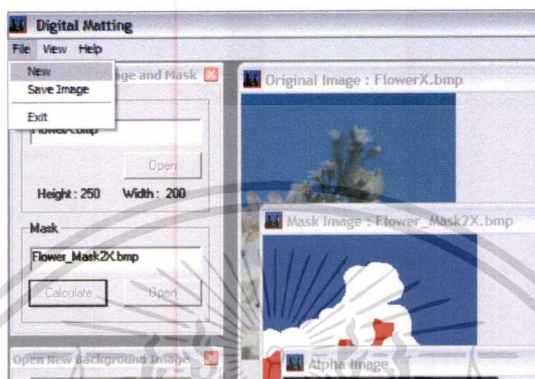


รูปที่ 4.9 แสดงเมนูของการบันทึกไฟล์ภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5 การเริ่มต้นทำงานใหม่

ถ้าต้องการเริ่มการทำงานของโปรแกรมใหม่สามารถทำได้โดยคลิกเลือกเมนู File > New ดังรูปที่ 4.10 หรือปิด โปรแกรมแล้วเปิดโปรแกรมเพื่อเริ่มทำงานใหม่



รูปที่ 4.10 แสดงการคลิกเลือกเมนูเพื่อเริ่มการทำงานใหม่

4.2.6 การออกจากโปรแกรม

ถ้าผู้ใช้ต้องการออกจากโปรแกรม ให้คลิกที่ปุ่ม Close ซึ่งเป็นรูปกากบาทสีแดง เพื่อปิด โปรแกรม หรือคลิกเลือกเมนู File > Exit ดังรูปที่ 4.11



รูปที่ 4.11 แสดงการคลิกเลือกเมนูเพื่อปิดการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการดำเนินโครงการ

5.1 การเปรียบเทียบภาพผลลัพธ์

ภาพผลลัพธ์ที่ได้จากการการศึกษาและพัฒนาโปรแกรมการซ้อนทับฉากหลังด้วยเทคนิคการประมาณค่าความโปร่งแสงของพิกเซลในโครงการนี้ เมื่อนำไปเปรียบเทียบกับภาพผลลัพธ์ที่ได้จากการใช้ฟังก์ชัน Extract ของ โปรแกรม Adobe Photoshop เพื่อแยกภาพวัตถุออกมาจากฉากหลังแล้ว จะเห็นว่า ภาพผลลัพธ์ที่ได้จากโปรแกรม Adobe Photoshop ไม่สามารถแยกภาพวัตถุในบริเวณขอบของภาพวัตถุออกมาได้



ก)

ข)

รูปที่ 5.1 ก) แสดงภาพผลลัพธ์ที่ได้จากโปรแกรมที่พัฒนาขึ้นจากโครงการนี้ และ

รูปที่ 5.1 ข) แสดงภาพผลลัพธ์ที่ได้จากโปรแกรม Adobe Photoshop

5.2 สรุปผลการดำเนินโครงการ

จากการศึกษาและพัฒนาโปรแกรมการซ้อนทับฉากหลังด้วยเทคนิคการประมาณค่าความโปร่งแสงของพิกเซล โดยได้ศึกษาจากงานวิจัยและทฤษฎีที่เกี่ยวข้อง ความรู้ที่ได้จากการศึกษางานวิจัยที่ผ่านมาและจากหลักการที่ได้นำเสนอในบทที่ผ่านมานั้น ช่วยให้สามารถพัฒนาโปรแกรมการซ้อนทับฉากหลังด้วยเทคนิคการประมาณค่าความโปร่งแสงของพิกเซลได้ ซึ่งมีผลลัพธ์เป็นที่น่าพอใจ

5.3 ข้อจำกัดและปัญหาของโครงการ

การดำเนินโครงการครั้งนี้ยังมีข้อจำกัดอยู่หลายประการ คือ

1. ขนาดของภาพที่จะนำมาประมวลผลต้องมีขนาดไม่เกิน 90,000 พิกเซล หรือ 300 x 300 เนื่องจากขนาดหน่วยความจำของ โปรแกรมมีจำกัด
2. ภาพที่จะนำมาใช้ใน โปรแกรมที่พัฒนาขึ้นจาก โครงการนี้ จะต้องเป็นภาพบิตแมปที่มีไฟล์นามสกุลเป็น .bmp เท่านั้น

5.4 ข้อเสนอแนะ

1. ในการดำเนินการต่อ ส่วนติดต่อกับผู้ใช้ต้องทำให้ดีกว่านี้
2. เพิ่มรูปแบบของภาพให้สามารถนำภาพที่มีไฟล์นามสกุลอื่นๆ ใช้กับโปรแกรมได้ และสามารถบันทึกไฟล์ภาพเป็นนามสกุลอื่นๆ ได้
3. พัฒนาให้สามารถใช้กับข้อมูลของภาพเคลื่อนไหวได้
4. พัฒนาโปรแกรมให้เป็นระบบมากกว่านี้โดยใช้การออกแบบเชิงวัตถุ

บรรณานุกรม

กัลยา วานิชย์บัญชา. 2546. การวิเคราะห์สถิติขั้นสูงด้วย SPSS for Windows. พิมพ์ครั้งที่ 3.

กรุงเทพฯ: ธรรมสาร.

ชัชวาลย์ เรืองประพันธ์. 2539. สถิติพื้นฐาน พร้อมตัวอย่างการวิเคราะห์ด้วยโปรแกรม MINITAB SPSS/PC+ และ SAS. ขอนแก่น: โรงพิมพ์คณิศรนาวิกวิทยา.

ปยุตต์รัตน์ พิษณุไพบูลย์. 2542. Computer Graphics: สำหรับนักออกแบบ. กรุงเทพฯ: สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.

Ruzon, A Mark. and Tomasi, Carla. Alpha Estimation in Natural Images. IEEE Conference on Computer Vision and Pattern Recognition. June 2000(1):18-25.

Smith, Alvy Ray and Blinn, F James. Blue Screen Matting. SIGGRAPH96. August 1996:259-268.



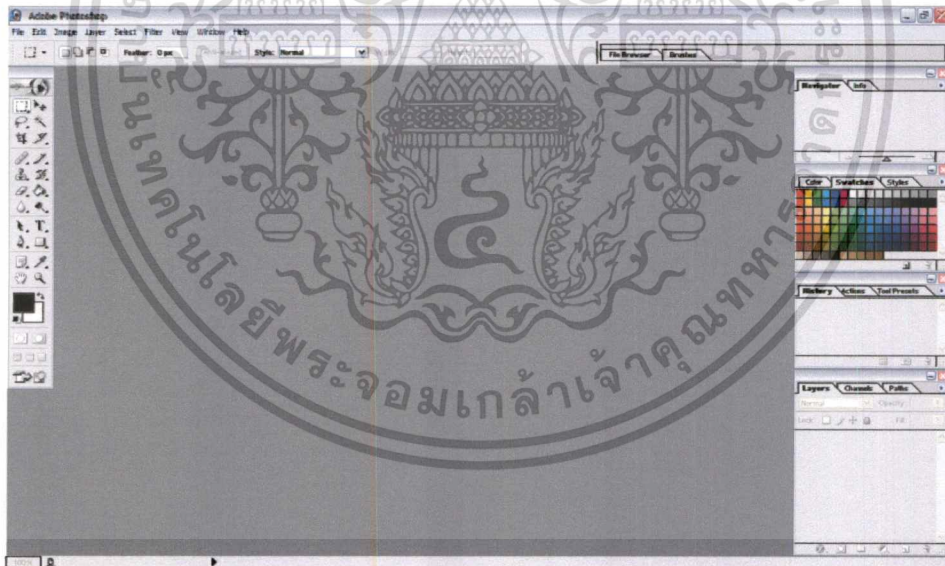
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

การกำหนดส่วนต่างๆ ของภาพโดยใช้โปรแกรม Adobe Photoshop

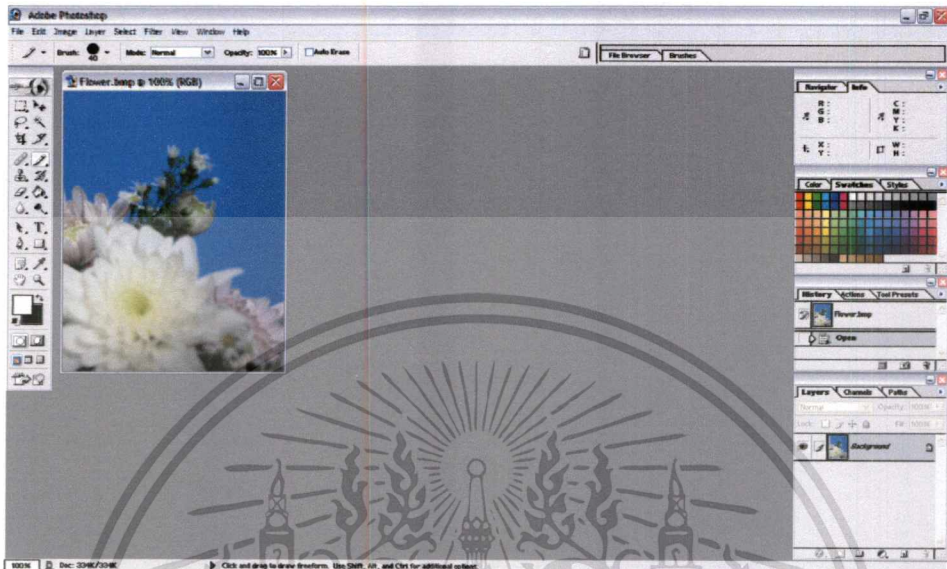
ในการทำงานของโปรแกรม Blue-screen Matting ของโครงการนี้ ผู้ใช้จะต้องสร้างภาพ Mask เพื่อกำหนดพื้นที่ส่วนต่างๆ ของภาพต้นฉบับที่ต้องการแยกภาพวัตถุออกมาเสียก่อน สำหรับใช้ในการคำนวณของโปรแกรม โดยพื้นที่ส่วนต่างๆ ของภาพประกอบด้วย 3 ส่วนคือ ส่วนของภาพวัตถุ ส่วนของฉากหลัง และส่วนของแนวขอบของภาพวัตถุ ซึ่งจะกำหนดให้มีค่าสีต่างกัน ดังที่จะกล่าวต่อไป

1. เปิดโปรแกรม Adobe Photoshop โดยคลิก Start > All Program > Adobe Photoshop 7.0

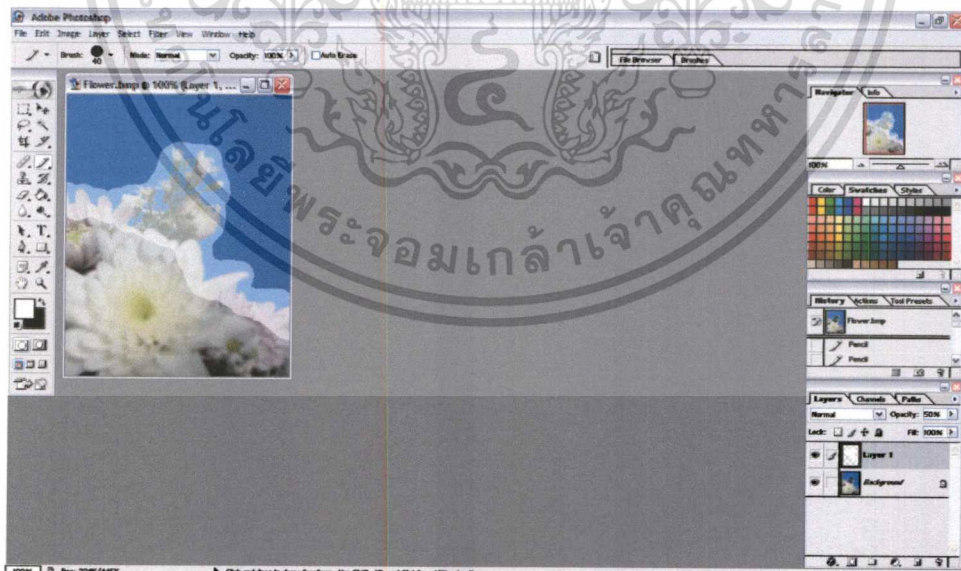


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เปิดภาพต้นฉบับที่ต้องการแยกภาพวัตถุออกมา ซึ่งจะต้องมีไฟล์นามสกุลเป็น .bmp เท่านั้น

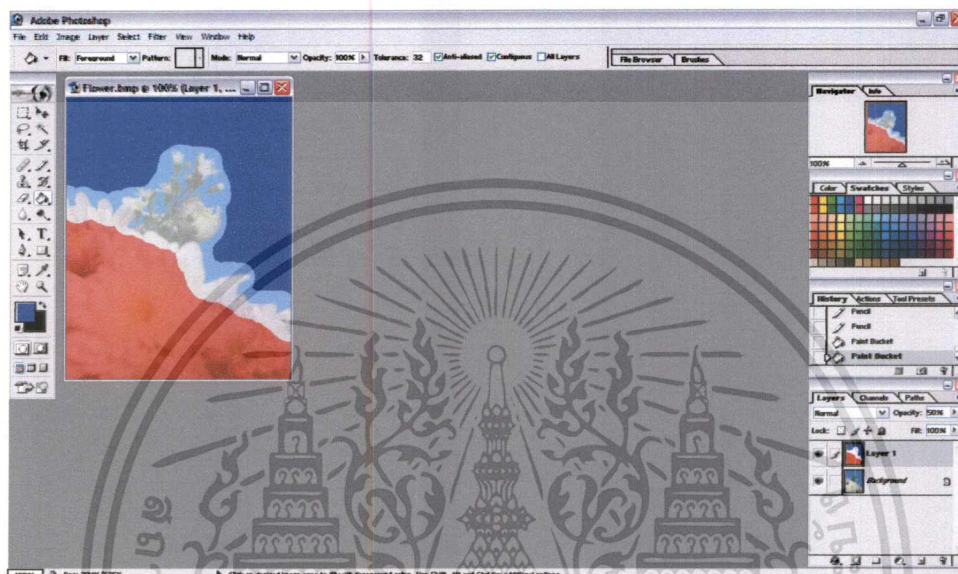


3. เลือกเครื่องมือ Pencil Tool แล้วเลือกขนาดของดินสอ และกำหนดสีของดินสอเป็นสีขาว (255, 255, 255) เพื่อกำหนดพื้นที่ของภาพในส่วนขอบของภาพวัตถุก่อน

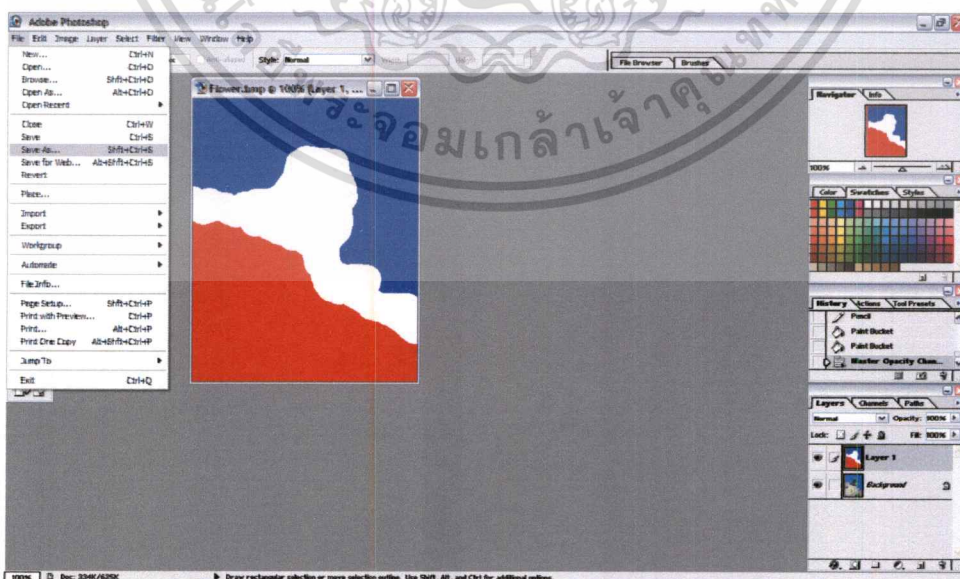


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เลือกเครื่องมือ Paint Bucket Tool โดยเทสีแดง (255, 0, 0) ลงในพื้นที่ส่วนที่เป็นส่วนของภาพวัตถุ และเทสีน้ำเงิน (0, 0, 255) ลงในพื้นที่ส่วนที่เป็นส่วนของฉากหลังอาจจะกำหนดค่า opacity ให้เท่ากับ 50% เพื่อให้สามารถมองเห็นและกำหนดส่วนต่างๆ ของภาพได้สะดวกขึ้น

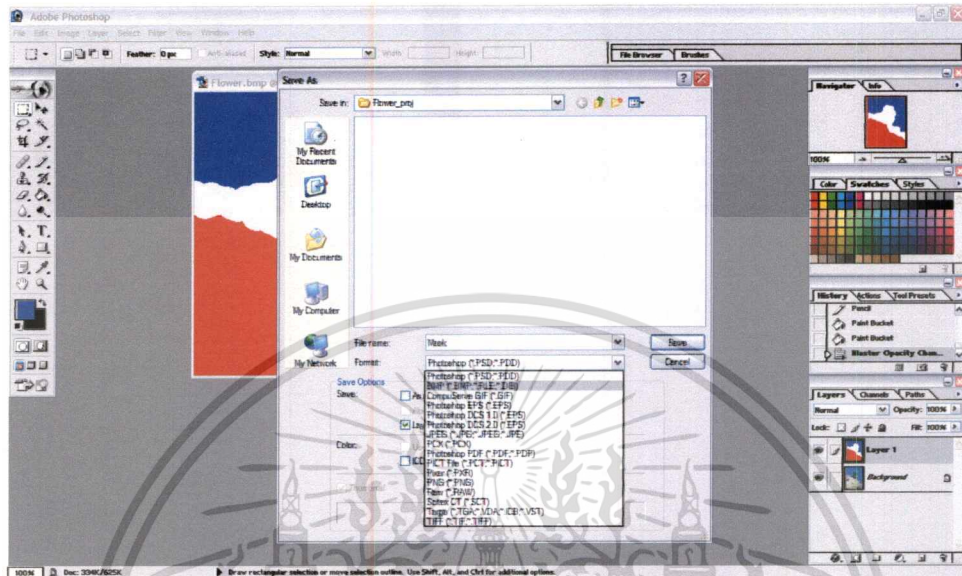


5. กำหนดให้ค่า opacity ให้เท่ากับ 100% แล้วคลิกเมนูเลือก File > Save As... เพื่อบันทึกภาพ Mask ของภาพต้นฉบับ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. บันทึกภาพ Mask ให้มีไฟล์นามสกุลเป็น .bmp เท่านั้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

การติดตั้งโปรแกรม Digital Matting

ความต้องการระบบขั้นต่ำ

1. Pentium PC 1.7 GHz
2. Windows XP with Service Pack 2
3. โปรแกรม Adobe Photoshop

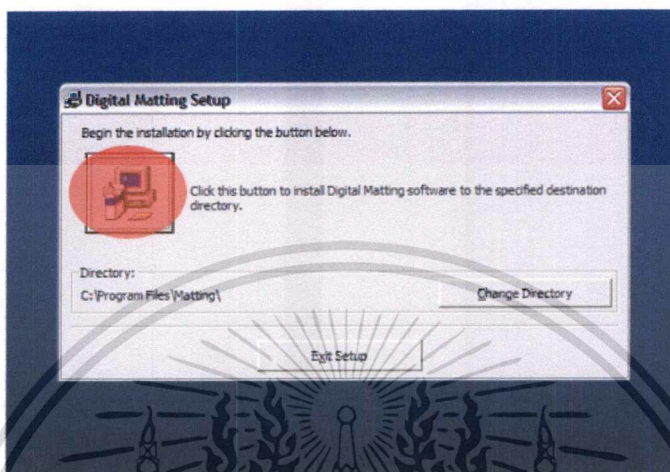
ขั้นตอนการติดตั้งใช้งาน

1. เปิดโปรแกรม Setup.exe ของโปรแกรม Digital Matting เลือก OK

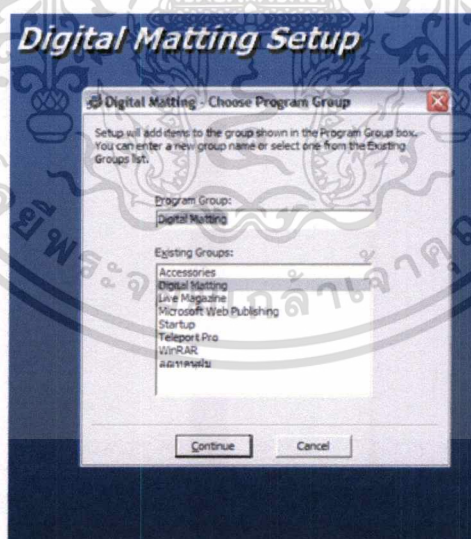


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เลือก Directory ที่จะทำการติดตั้ง แล้วคลิกปุ่มที่มีไอคอนรูป PC เพื่อเริ่มการติดตั้งโปรแกรม

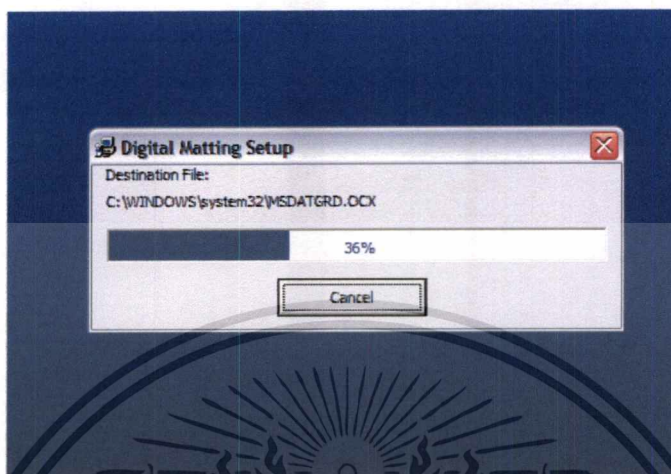


- สามารถเปลี่ยนชื่อของโปรแกรมที่จะทำการติดตั้งได้ ถ้าต้องการ เมื่อเปลี่ยนเรียบร้อยแล้วให้คลิกเลือกปุ่ม Continue

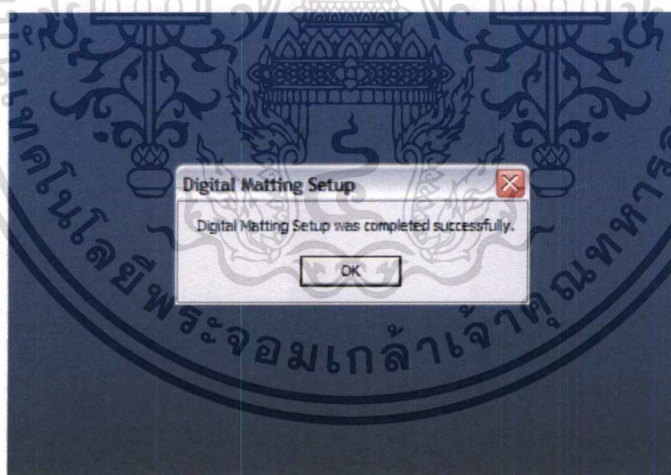


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. โปรแกรมจะเริ่มทำการติดตั้ง ให้รอจนการติดตั้งสิ้นสุด



5. เมื่อการติดตั้ง โปรแกรมเสร็จเรียบร้อยแล้ว จะมีหน้าต่างปรากฏดังรูป ให้คลิกปุ่ม OK เพื่อสิ้นสุดการติดตั้ง



6. การเรียกใช้งาน โปรแกรมคือ Start > All Programs > Digital Matting

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน นางสาวสุนิสา วงษ์ที
วันเดือนปีเกิด 9 สิงหาคม 2523
วุฒิการศึกษาระดับมัธยมปลาย ราชวินิตบางแก้ว สมุทรปราการ
วุฒิการศึกษาระดับปริญญาตรี วิทยาศาสตร์บัณฑิต วิชาเอกสถิติ มศว. ประสานมิตร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้