การสร้างภาพโมเสกโดยโปรแกรม MATLAB

Mosaic Image using MATLAB

นางสาวมัณฑนา เผ่าวิทยานนท์ รหัส 45370236

นางสาวสุจิตตา บรรดาศักดิ์ รหัส 45370335

I 438 4982

ห้องสมุดคณะวิศวกรรมศาสตร์
13 พ.ย. 2549
วันที่รับ...../...../.....
เลขทะเบียน 4900126
เลขเรียกหนังสือ ปร.
มหาวิทยาลัยนเรศวร 43359
2548

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2548

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	การสร้างภาพโมเสกโดยโปรแกรม MATLAB			
ผู้ดำเนินโครงงาน	นางสาวมัณฑนา	เผ่าวิทยานนท์	รหัส	45370236
	นางสาวสุจิตตา	บรรดาศักดิ์	รหัส	45370335
อาจารย์ที่ปรึกษา	ดร. สมยศ เกียรติวนิชวิไล			
สาขาวิชา	วิศวกรรมคอมพิวเตอร์			
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์			
ปีการศึกษา	2548			

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

..... ประธานกรรมการ

(ดร. สมยศ เกียรติวนิชวิไล)

..... กรรมการ

(ดร. สุรเชษฐ์ กานต์ประชา)

..... **กรรมการ**

(ดร. พนมชวัญ ริยะมงคล)

หัวข้อโครงงาน	การสร้างภาพโมเสกโดยโปรแกรม MATLAB			
ผู้ดำเนินโครงงาน	นางสาวมัณฑนา	เผ่าวิทยานนท์	รหัสนิสิต	45370236
	นางสาวสุจิตตา	บรรดาศักดิ์	รหัสนิสิต	45370335
อาจารย์ที่ปรึกษา	ดร. สมยศ	เกียรดิวนิชวิไล		
สาขา	วิศวกรรมคอมพิวเตอร์			
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์			
ปีการศึกษา	2548			

บทคัดย่อ

โครงงานนี้เป็นการศึกษาเกี่ยวกับการสร้างภาพโมเสกโดยใช้โปรแกรม MATLAB ในการเขียนโปรแกรมที่จะแบ่งภาพที่ต้องการ ให้เป็นหลายๆเฟรม และนำภาพหลายๆภาพมาเป็นตัวเลือกในการประกอบภาพลงในแต่ละเฟรม โดยจะนำภาพที่มีสีใกล้เคียงที่สุด เมื่อเปรียบเทียบกับแต่ละเฟรม มาประกอบภาพลงในเฟรม ซึ่งในโปรแกรมจะมีการใช้ GUI เพื่อให้ผู้ใช้กรอกข้อมูลที่ต้องการ โดยภาพโมเสกที่ได้จะมาจากการประกอบภาพที่มีสีใกล้เคียงกับภาพที่ต้องการ

ผลที่ได้จากโครงงานนี้ คือ โปรแกรมที่สามารถแสดงผลในการสร้างภาพโมเสก

Project title	Mosaic Image using MATLAB		
Name	Miss Mantana	Paowittayanon	ID. 45370236
	Miss Suchitta	Bandasak	ID. 45370335
Project advisor	Dr. Somyot	Kiattivanichvilai	
Major	Computer Engineering		
Department	Electrical and Computer Engineering		
Academic year	2005		

.....

Abstract

This project studies about Mosaic Image using MATLAB. MATLAB program is used as the developing tools.Firstly, a target image is divided to multi-frames and several pictures are used as input images that be selected to affix in each frame. Secondly, picture that has nearest color to an individual frame is selected to fill in the frame.The developed program has a graphic user interface(GUI) for user interfacing. Finally, the composition of pictures in this way will construct the Mosaic image which similar to a target images.

The results show the effectiveness of developed program for creating a Mosaic Images.

กิตติกรรมประกาศ

ในการทำโครงงานวิศวกรรมครั้งนี้ คณะผู้จัดทำขอกราบขอบพระคุณ ดร. สมยศ เกียรติวนิชวิไล ซึ่งเป็นที่ปรึกษาโครงงาน คามที่ท่านได้ให้ความกรุณาแนะนำวิธีในการทำงานให้เข้าใจถึงหลักการทฤษฎีต่างๆ อย่างเป็นระบบขั้นตอน อีกทั้งยังสละเวลาเพื่อตรวจสอบการทำงานและชี้แนะแนวทางแก้ไขในทุกขั้นตอนตลอดการศึกษา

นางสาวมัณฑนา เผ่าวิทยานนท์

นางสาวสุจิตตา บรรดาศักดิ์

สารบัญ

หน้า

ใบรับรองโครงงานวิศวกรรม..... ก

บทคัดย่อภาษาไทย..... ข

บทคัดย่ออังกฤษ..... ค

กิตติกรรมประกาศ..... ง

สารบัญ..... จ

สารบัญรูป..... ช

บทที่1 บทนำ

1.1 ที่มาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน..... 2

1.3 ขอบข่ายของโครงงาน..... 3

1.4 ขั้นตอนของการดำเนินงาน 3

1.5 แผนการดำเนินงาน..... 4

1.6 ผลที่คาดว่าจะได้รับ 4

1.7 งบประมาณของโครงงาน 4

บทที่2 ทฤษฎีเบื้องต้นและหลักการที่เกี่ยวข้อง

2.1 มาตรฐานของสี 5

2.2 พิกเซล(pixel)..... 6

บทที่3 วิธีการดำเนินงาน

3.1 การแบ่งภาพต้นฉบับ..... 8

3.2 การเลือกไทล์..... 9

3.3 ค่าเฉลี่ยของสี 9

สารบัญ (ต่อ)

บทที่4 ผลการทดลอง..... 11

บทที่5 สรุปผล

5.1 สรุปผล..... 27

5.2 ข้อเสนอแนะ..... 27

5.3 ปัญหาที่พบในการทำโครงงาน..... 27

5.4 แนวทางแก้ไข..... 28

บรรณานุกรม 29

ภาคผนวก 30

สารบัญรูป

รูปที่ หน้า

1.1 ตัวอย่างภาพโมเสกที่ประกอบขึ้นจากภาพเล็ก ๆ จำนวนมาก 1
2.1 แสดงพิกเซลของภาพเฉดสีขาว/ดำ และพิกเซลของภาพสี..... 6
2.2 เปรียบเทียบความละเอียดภาพ 72 dpi กับ 300 dpi 7
3.1 ตัวอย่างการแบ่งภาพต้นฉบับออกเป็นหลายๆเฟรม..... 8
3.2 ภาพต้นฉบับ 9
3.3 ภาพต้นฉบับเมื่อทำการแบ่งเป็นหลายๆเฟรม 9
4.1 การเรียกใช้ file.fig โดยพิมพ์ guide ที่ command window 11
4.2 การเรียกใช้ file.fig โดยเลือกที่ Open Existing GUI 11
4.3 เลือก file.fig ที่ต้องการ 12
4.4 หน้าจอที่สร้างใน f.fig 12
4.5 ทำการรันโปรแกรม 13
4.6 โปรแกรมจะให้กรอกข้อมูลที่ต้องการ 14
4.7 กรอกข้อมูลที่ต้องการ 15
4.8 กดปุ่ม Load Picture Background 16
4.9 เลือกภาพต้นฉบับ 17
4.10 กดปุ่ม Run Training file 18
4.11 กดปุ่ม Show result เพื่อทำการแสดงผล 19
4.12 ภาพต้นฉบับ..... 20
4.13 ไทล์ที่นำมาประกอบภาพโมเสก..... 20
4.14 แสดงภาพต้นฉบับและภาพโมเสกที่ 1..... 23
4.15 แสดงภาพที่ใช้เป็นไทล์ของรูปที่ 4.14..... 23
4.16 แสดงภาพต้นฉบับที่ 2..... 24
4.17 แสดงภาพโมเสกที่ 2 เมื่อแบ่งเป็น 5×5 10×10 และ 20×20 เฟรมตามลำดับ..... 24
4.18 แสดงภาพต้นฉบับและภาพโมเสกที่ 3..... 25
4.19 แสดงภาพต้นฉบับและภาพโมเสกที่ 4..... 25
4.20 แสดงการเปรียบเทียบระหว่างภาพต้นฉบับและภาพโมเสก..... 26
ก-1 เริ่มการติดตั้งโปรแกรม MATLAB 30
ก-2 ใส่ Personal License Password (PLP)..... 30

สารบัญรูป (ต่อ)

ก-3 หน้าต่าง License Agreement 31
ก-4 ใส่ชื่อและชื่อบริษัท 31
ก-5 เลือก Product 32
ก-6 หน้าต่าง Product Configuration Notes 32
ก-7 หน้าต่าง Installation Complete เมื่อทำการ install เสร็จ 33

บทที่ 1
บทนำ

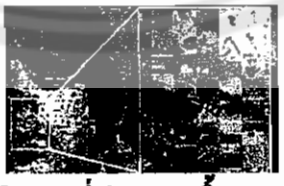
1.1 ที่มาและความสำคัญของโครงงาน

ในปัจจุบันโลกของเราได้พัฒนาขึ้นมาก มีเทคโนโลยีใหม่ๆเข้ามามากมาย รวมทั้งมีความก้าวหน้าทางเทคโนโลยีคอมพิวเตอร์ขึ้นมากจึงทำให้มนุษย์มีความเกี่ยวข้องกับการใช้คอมพิวเตอร์อยู่เป็นประจำ เช่นในการทำงานต้องใช้คอมพิวเตอร์ในการจัดเก็บข้อมูล หรือทำอะไรหลายๆอย่างที่ช่วยแบ่งเบาภาระของมนุษย์ได้ รวมทั้งการใช้คอมพิวเตอร์กราฟิกในการปรับปรุงภาพ การสร้างแอนิเมชัน สร้างภาพเกินจริง และอื่นๆ ซึ่งในอนาคตอาจมีการใช้คอมพิวเตอร์ในการสร้างสิ่งที่แปลกใหม่เกิดขึ้นมาอีกก็เป็นได้

ปัจจุบันมีบริษัทผลิตสินค้าที่เป็นความต้องการของประชาชนขึ้นมามากมายและหลากหลายรูปแบบ ธุรกิจจึงมีการแข่งขันกันอย่างมาก ทำให้การทำสื่อโฆษณามีบทบาทมากยิ่งขึ้น ซึ่งส่งผลให้เกิดการพัฒนาในการทำสื่อโฆษณาออกมาในหลายรูปแบบ เพื่อดึงดูดความสนใจของลูกค้า หรือผู้พบเห็นเพื่อให้สื่อโฆษณานั้นเด่น หรือแปลกใหม่สะดุดตา ทำให้ลูกค้าจดจำสินค้า หรือบริษัทผู้ผลิตได้ จึงมีการนำคอมพิวเตอร์กราฟิกมาใช้ในการปรับปรุงภาพ ติดต่อภาพ สร้างภาพเกินจริง และอื่นๆ รวมทั้งการสร้างภาพโมเสกด้วย

การสร้างภาพโมเสกเป็นวิธีการหนึ่ง ในการดึงดูดความสนใจของผู้พบเห็น ซึ่งไม่เพียงเป็นประโยชน์การทำสื่อโฆษณาเท่านั้น ภาพโมเสกยังสามารถทำเป็นภาพเบลอเพื่อใช้ปกปิดภาพที่ไม่ต้องการให้ผู้พบเห็นได้เห็นภาพนั้นอย่างชัดเจน แต่สามารถรู้ได้ว่าสิ่งที่ปกปิดนั้นคืออะไรได้อีกด้วย เช่น การนำภาพโมเสกมาใช้เป็นภาพเบลอในการปกปิดภาพคนสูบบุหรี่ หรือทำเป็นภาพเบลอในการปกปิดการดื่มสุราในการแสดงละคร เพื่อไม่ให้เด็กนำไปเป็นตัวอย่างในการเลียนแบบ เป็นต้น

การสร้างภาพโมเสก เป็นกระบวนการสร้างภาพ โดยภาพผลลัพธ์จะประกอบขึ้นจากภาพเล็ก ๆ จำนวนมาก (ซึ่งภาพเล็ก ๆเหล่านี้เรียกว่า ไทล์(Tile) มาเรียงต่อกันอย่างมีระบบทำให้เกิดภาพใหม่ ดังในรูปที่ 1.1[2]



รูปที่ 1.1 ตัวอย่างภาพโมเสกที่ประกอบขึ้นจากภาพเล็ก ๆ จำนวนมาก

การสร้างภาพโมเสก เป็นการรวมกันอย่างเหมาะสมระหว่างศาสตร์แห่งการถ่ายภาพ และศาสตร์แห่งการประมวลผลภาพทางดิจิตอล ซึ่งไม่เพียงทำให้เกิดภาพที่มีลักษณะเฉพาะเท่านั้น แต่ยังทำให้เกิดภาพที่มีความสวยงามอีกด้วย

การสร้างภาพโมเสกนั้นสามารถทำได้หลายวิธีในการเลือกไทล์ ที่เหมาะสม ในการแทนไทล์ แต่ละไทล์ ลงในภาพที่ต้องการ เพื่อให้ภาพผลลัพธ์เป็นภาพโมเสกที่มีแนวโน้มไปในทิศทางที่ต้องการมากที่สุด โดยจะศึกษาถึงความสัมพันธ์ของขนาดของภาพที่จะใช้เป็นภาพต้นฉบับ กับขนาดไทล์ ที่เหมาะจะนำมาใช้สร้างภาพโมเสก ส่วนเทคนิคที่ใช้เลือกไทล์ เพื่อให้ได้ภาพผลลัพธ์ที่ดีนั้นมีหลายวิธีเช่น การเลือกไทล์ ที่มีค่าเฉลี่ยของสีใกล้เคียงส่วนของภาพมากที่สุด หรือการเลือกไทล์ ที่มีค่าเฉลี่ยของสีใกล้เคียงส่วนของภาพมาชุดหนึ่ง แล้วเลือกใช้ไทล์ ที่มีค่าเบี่ยงเบนมาตรฐานใกล้เคียงกับส่วนของภาพที่สุดมาใช้ เป็นต้น[3]

ในโครงงานนี้ใช้วิธีการเลือกไทล์ ที่มีค่าเฉลี่ยของสีใกล้เคียงส่วนของภาพมากที่สุด เนื่องจากเป็นวิธีที่ให้ภาพผลลัพธ์ เป็นภาพโมเสกที่มีแนวโน้มไปในทิศทางที่ต้องการมากที่สุด ซึ่งในการหาค่าเฉลี่ยของสีใกล้เคียงส่วนของภาพมากที่สุดนั้น จะใช้วิธีการหาค่าผิดพลาด ซึ่งเป็นการเปรียบเทียบค่าเฉลี่ยของสี RGB ในแต่ละไทล์ กับเฟรม(frame) ที่แบ่งมาจากภาพต้นฉบับ โดยไทล์ใดมีค่าผิดพลาดน้อยที่สุด นั่นคือไทล์ที่เหมาะสมที่จะนำมาแปะลงในเฟรม นั้น เช่น เฟรม แรกในภาพต้นฉบับมีค่าเฉลี่ยของสี RGB เป็น 240 ซึ่งเป็นภาพที่มีลักษณะในโทนสีขาว และในแต่ละไทล์ จะมีค่าเฉลี่ยของสี RGB แตกต่างกันออกไป นั่นคือมีโทนสีที่แตกต่างกัน ซึ่งถ้าไทล์ ใดมีค่าเฉลี่ยของสี RGB ใกล้เคียง 240 มากที่สุด นั่นคือไทล์ ที่เหมาะสมที่สุดที่จะนำมาแปะลงในเฟรม แรก นั่นคือไทล์ นั้นจะมีโทนสีขาวเช่นเดียวกับเฟรม แรก ไทล์ควรมีหลากหลาย จำนวนมาก และควรมีการกระจายของภาพให้ครอบคลุมช่วงของค่าข้อมูล เพื่อให้ภาพผลลัพธ์ที่ออกมา มีแนวโน้มไปในทิศทางที่ต้องการมากที่สุด

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อต้องการเรียนรู้หลักการสร้างภาพโมเสกของภาพโมเสก

1.2.2 เพื่อเรียนรู้คุณสมบัติของสี

1.2.3 สามารถสร้างโปรแกรมสร้างและแสดงผลการใช้ภาพ

1.3 ขอบข่ายของโครงงาน

1.3.1 สร้างโปรแกรมในการสร้างภาพโมเสกโดยใช้โปรแกรม MATLAB

1.3.2 ประยุกต์ใช้ทฤษฎีปัญญาประดิษฐ์ในการควบคุมการทำงาน

1.4 ขั้นตอนของการดำเนินงาน

1.4.1 ศึกษาและค้นคว้าข้อมูลเกี่ยวกับหลักการในการสร้างภาพโมเสก

1.4.2 ค้นคว้าและศึกษาเกี่ยวกับทฤษฎีของสี RGB

1.4.3 เขียนโปรแกรมเพื่อสร้างภาพโมเสก

1.4.4 ทดสอบการทำงาน

1.4.5 สรุปผลการทดลองและจัดทำรูปเล่มโครงงาน

1.5 แผนการดำเนินงาน

กิจกรรม	ปี2547		ปี2548									
	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.
1. ศึกษาและค้นคว้าข้อมูลเกี่ยวกับหลักการในการสร้างภาพโมเสก	←			→								
2. ค้นคว้าและศึกษาเกี่ยวกับทฤษฎีของสี RGB					←		→					
3. เขียนโปรแกรมเพื่อสร้างภาพโมเสก							←			→		
4. ทดสอบการทำงาน									←		→	
5. สรุปผลการทดลองและจัดทำรูปเล่มโครงงาน										←	→	

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 สามารถนำไปใช้ในการสร้างภาพโมเสกได้

1.6.2 สามารถนำไปใช้ในสื่อโฆษณาเพื่อเกิดความสวยงามได้

1.6.3 สามารถนำไปประยุกต์ใช้ในการสร้างเทคนิคพิเศษต่างๆได้

1.6.4 สามารถนำไปใช้ในการทำภาพเบลอเพื่อปกปิดสิ่งที่ไม่ต้องการให้เห็นชัดได้

1.7 งบประมาณของโครงงาน

1.7.1 ค่าถ่ายเอกสารและค่าเข้าเล่มโครงงาน

1.7.2 ค่าแผ่นซีดี

1.7.3 ค่าหนังสือโปรแกรม MATLAB

1.7.4 ค่าหมึกพิมพ์

รวมเป็นเงิน 2,000 บาท(สองพันบาทถ้วน)

บทที่2
ทฤษฎีเบื้องต้นและหลักการที่เกี่ยวข้อง

2.1 มาตรฐานของสี

ในปัจจุบันจะพบว่ามาตรฐานของสีที่ใช้นั้นจะมีหลายระบบด้วยกันซึ่งขึ้นอยู่กับกานำมาใช้แต่ส่วนใหญ่แล้วก็จะมีแนวคิดที่คล้ายๆกัน ไม่ค่อยแตกต่างกันเท่าไรนัก ซึ่งส่วนใหญ่จะใช้หลักการแทนจุดสีด้วยจุดที่อยู่ภายในสเปส 3 มิติโดยจะใช้การอ้างอิงสำหรับจุดสีนั้นในสเปสซึ่งแต่ละแกนมีความเป็นอิสระต่อกันเช่น RGB จะมีแกนสีคือ แกนสีแดง เขียว และน้ำเงิน ในระบบของ HLS จะมีแกนเป็นค่าสี (hue) และ ความสว่าง(lightness)

2.1.1 ระบบสี RGB

ระบบสี RGB เป็นระบบสีที่เกิดจากการรวมกันของแสงสีแดง เขียวและน้ำเงินโดยมีการรวมกันแบบ Additive ซึ่งปรกติจะนำไปใช้ในจอภาพ CRT (Cathode ray tude) ในการใช้งานในระบบสี RGB นั้นยังมีการใช้มาตรฐานแต่มีความแตกต่างกันอีกคือ ระบบสีแบบ RGBของ CIE หรือ RGB_{CIE}เป็นระบบที่พัฒนาขึ้น โดย CIE(Commission International I'Eclairage) ซึ่งนำสีแดงมาใช้ในการอ้างอิงคือ อ้างอิงสีแดงที่ 700 nm. นำสีเขียวมาอ้างอิงที่สีเขียวมีค่าเท่ากับ 546.1 nm. และสีน้ำเงินมาอ้างอิงที่ สีน้ำเงินเท่ากับ 435.8 nm. ระบบสีแบบ และRGB ของ NTSC หรือ RGB_{NTSC} เป็นระบบสีที่พัฒนาขึ้นโดย NTSC (National Television System Committee) เพื่อใช้สำหรับแสดงภาพของจอภาพแบบ CRT เป็นมาตรฐานของผู้ผลิตแบบ CRTให้มีลักษณะที่เดียวกันทั้งหมด

2.1.2 ระบบสี HSV(Hue Saturation Value)

เป็นการพิจารณาสีโดยใช้ Hue Saturation และ Value ซึ่ง Hue คือค่าของสีหลัก(Red Green and Blue)ในทางปฏิบัติจะอยู่ระหว่าง 0 และ 255 ซึ่งถ้าแทน Hue มีค่าเท่ากับ 0 จะแทนสีแดงและเมื่อ Hue มีค่าเพิ่มขึ้นเรื่อยๆสีนั้นก็จะมีการเปลี่ยนแปลงเรื่อยๆ ตามคุณสมบัติของสเปกตรัมของสีจนถึง 256 จึงจะกลับมาเป็นสีแดงอีกครั้งหนึ่ง

2.1.3 ระบบสี CMYK

เป็นระบบสีชนิดที่เป็นวัตถุ คือสีแดง เหลือง น้ำเงิน แต่ไม่ใช่สีน้ำเงินที่เป็นแม่สีวัตถุธาตุ แม่สีในระบบ CMYK เกิดจากการผสมกันของแม่สีของแสง หรือ ระบบสีRGB คือ

แสงสีน้ำเงิน + แสงสีเขียว = สีฟ้า (Cyan)

แสงสีน้ำเงิน + แสงสีแดง = สีแดง (Magenta)

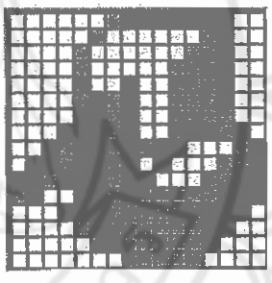
แสงสีแดง + แสงสีเขียว = สีเหลือง (Yellow)

สีฟ้า สีแดง สีเหลือง นี้นำมาใช้ในระบบการพิมพ์ และ มีการเพิ่มเติม สีดำเข้าไป เพื่อให้มีน้ำหนักเข้มขึ้นอีก เมื่อรวมสีดำ (Black = K) เข้าไป จึงมีสี่สี โดยทั่วไปจึงเรียกระบบการพิมพ์นี้ว่าระบบการพิมพ์สี่สี (CMYK) ระบบการพิมพ์สี่สี (CMYK) เป็นการพิมพ์ภาพในระบบที่ทันสมัยที่สุด และได้ภาพใกล้เคียงกับภาพถ่ายมากที่สุด โดยทำการพิมพ์ทีละสี จากสีเหลือง สีแดง สีน้ำเงิน และสีดำต้องใช้แว่นขยายส่องดู ผลงานพิมพ์ชนิดนี้ จะพบว่า จะเกิดจากจุดสีเล็ก ๆ สี่สีอยู่เต็มไปหมด การที่เรามองเห็นภาพมีสีต่าง ๆ นอกเหนือจากสี่สีนี้ เกิดจากการผสมของเม็ดสีเหล่านี้ใน ปริมาณต่าง ๆ คิดเป็นเปอร์เซนต์ของปริมาณเม็ดสี ซึ่งกำหนดเป็น 10-20-30-40-50-60-70-80-90 จนถึง 100 เปเซนต์

2.2 พิกเซล(pixel)

พิกเซล (Pixel) เป็นคำผสมของคำว่า Picture กับคำว่า Element หรือหน่วยพื้นฐานของภาพ เทียบได้กับ "จุดภาพ" 1 จุด แต่ละพิกเซลเปรียบได้กับสี่เหลี่ยมเล็กๆ ที่บรรจุค่าสี โดยถูกกำหนดตำแหน่งไว้บนเส้นกริดของแนวแกน x และแกน y หรือในตารางเมตริกซ์สี่เหลี่ยม ภาพบิตแมปจะประกอบด้วยพิกเซลหลายๆ พิกเซล

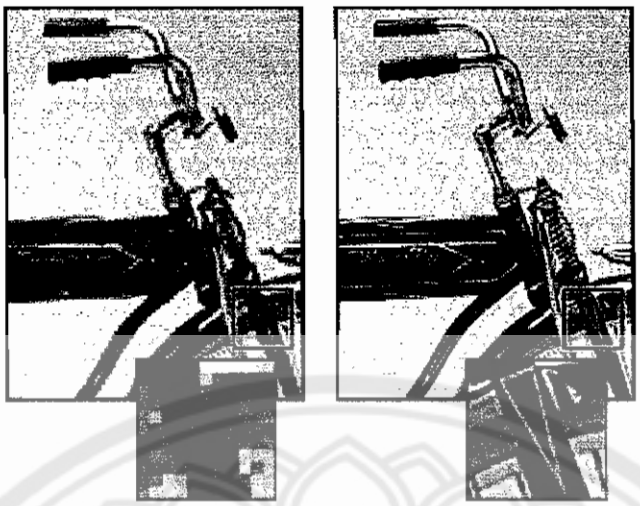
พิกเซลของภาพเฉดสีขาว/ดำ



พิกเซลของภาพสี

รูปที่2.1 แสดงพิกเซลของภาพเฉดสีขาว/ดำ และ พิกเซลของภาพสี

จำนวนพิกเซลของภาพแต่ละภาพ จะเรียกว่า ความละเอียด หรือ Resolution โดยจะเทียบจำนวนพิกเซลกับความยาวต่อนิ้ว ดังนั้นจะมีหน่วยเป็น พิกเซลต่อนิ้ว (ppi: pixels per inch) หรือจุดต่อนิ้ว (dpi; dot per inch) ภาพขนาดเท่านั้นแต่มีความละเอียดต่างกัน แสดงว่าจำนวนพิกเซลต่างกัน และขนาดของจุดพิกเซลก็ต่างกันด้วย ดังรูปต่อไปนี้



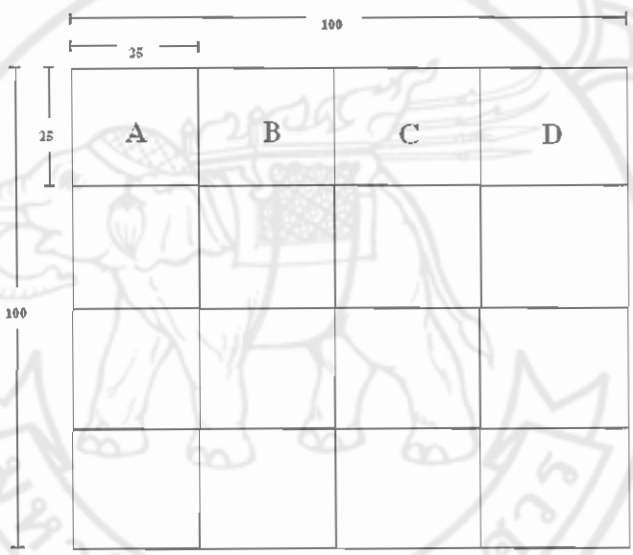
รูปที่2.2 เปรียบเทียบความละเอียดภาพ 72 dpi กับ 300 dpi

จะเห็นว่าเมื่อเปรียบเทียบจุดต่อนิ้วแล้วจะเห็นว่า ความละเอียดของภาพที่เปรียบเทียบจุดต่อนิ้ว เท่ากับ 300 dpi นั้นภาพจะมีความคมชัดและละเอียดมากว่าที่เปรียบเทียบจุดต่อนิ้วเท่ากับ 72 dpi

บทที่ 3
วิธีการดำเนินงาน

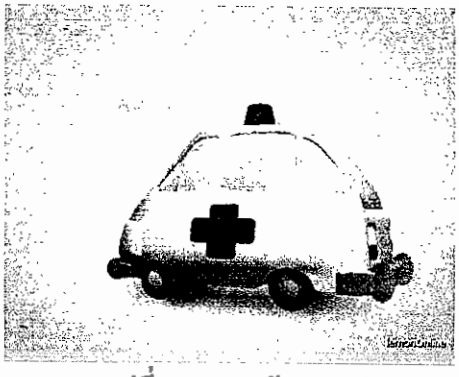
3.1 การแบ่งภาพต้นฉบับ

ในการสร้างภาพโมเสกนั้น ขั้นแรกคือ เลือกภาพที่จะใช้เป็นภาพต้นฉบับ และทำการแบ่งภาพต้นฉบับออกเป็นหลายๆเฟรม ดังตัวอย่างในรูปที่ 3.1 แสดงการแบ่งภาพต้นฉบับที่มีขนาดพิกเซล 100×100(กว้าง100 สูง100) ออกเป็น 4 แถว(row) 4 คอลัมน์(column) โดยในเฟรม A จะเป็นภาพที่ 1,1 (แถวที่1 คอลัมน์ที่1) เฟรม B จะเป็นภาพที่ 1,2 (แถวที่1 คอลัมน์ที่2) ตามลำดับ

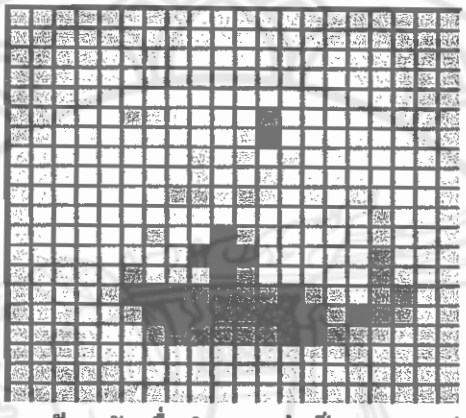


รูปที่ 3.1 ตัวอย่างการแบ่งภาพต้นฉบับออกเป็นหลายๆเฟรม

เมื่อใช้ภาพต้นฉบับที่มีขนาดพิกเซล 800×600(กว้าง 800 สูง 600) และทำการแบ่งออกเป็น 20 แถว 20 คอลัมน์ดังนั้นจะได้เฟรมทั้งหมด 400 เฟรม โดยแต่ละเฟรม จะมีขนาดพิกเซลเท่ากันคือ 40×30(กว้าง40 สูง 30) และมีสีที่แตกต่างกันออกไป ดังรูปที่ 3.2 แสดงภาพต้นฉบับ และเมื่อทำการแบ่งภาพต้นฉบับแล้วจะได้ผลดังรูปที่ 3.3



รูปที่ 3.2 ภาพต้นฉบับ



รูปที่ 3.3 ภาพต้นฉบับเมื่อทำการแบ่งเป็นหลายๆเฟรม(frame)

3.2 การเลือกไทล์(Tile)

ไทล์ คือภาพหลากหลายภาพที่เป็นตัวเลือกในการนำมาประกอบภาพลงในแต่ละเฟรม โดยในการประกอบภาพ จะเลือกไทล์ ที่มีค่าเฉลี่ยของสี RGB ที่ใกล้เคียงที่สุดกับเฟรม นั้น ไทล์ ควรมีหลากหลาย จำนวนมาก และควรมีการกระจายของภาพให้ครอบคลุมช่วงของค่าข้อมูล เพื่อให้ภาพผลลัพธ์ออกมามีแนวโน้มไปในทิศทางที่ต้องการมากที่สุด

ในการค้นหาภาพที่มีโทนสีใกล้เคียงกับเฟรม ที่สนใจมากที่สุดนั้น จะใช้วิธีการหาค่าผิดพลาด ซึ่งเป็นการเปรียบเทียบค่าเฉลี่ยของสี RGB ดังจะกล่าวต่อไป

3.3 ค่าเฉลี่ยของสี

การเลือกไทล์ ที่นำมาประกอบภาพลงในแต่ละเฟรม นั้น จะเลือกไทล์ที่มีค่าเฉลี่ยของสี RGB ที่ใกล้เคียงกับเฟรม นั้นๆมากที่สุด ยกตัวอย่างเช่น เฟรม A ดังในรูปที่ 3.1 มีค่าเฉลี่ยของสี RGB เป็น 250 ซึ่งมีลักษณะของสีไปในโทนสีขาว เมื่อเลือกไทล์ จะเลือกไทล์ ที่มีค่าเฉลี่ยของสี RGB ที่ใกล้เคียง 250 มากที่สุด นั่นคือไทล์ นั้นมีลักษณะของสีไปในโทนสีขาวเช่นเดียวกับเฟรม A

ในการเปรียบเทียบค่าเฉลี่ยของสี RGB ของแต่ละไทล์ กับเฟรม ที่สนใจ จะเปรียบเทียบโดยการหาค่าผิดพลาด ไทล์ ใดที่มีค่าผิดพลาดน้อยที่สุดเมื่อเปรียบเทียบกับเฟรม ที่สนใจ ไทล์ นั้นคือไทล์ ที่เหมาะสมที่สุดในการนำมาประกอบภาพลงในเฟรมนั้นๆ โดยการหาค่าผิดพลาดสามารถคำนวณได้ดังนี้

$$error = \sqrt{(R_{tile} - R_{frame})^2 + (G_{tile} - G_{frame})^2 + (B_{tile} - B_{frame})^2} \quad (3.1)$$

R_{tile} คือ ค่าเฉลี่ยของสี R ในไทล์ นั้น

R_{frame} คือ ค่าเฉลี่ยของสี R ในเฟรมที่สนใจ

G_{tile} คือ ค่าเฉลี่ยของสี G ในไทล์ นั้น

G_{frame} คือ ค่าเฉลี่ยของสี G ในเฟรม ที่สนใจ

B_{tile} คือ ค่าเฉลี่ยของสี B ในไทล์นั้น

B_{frame} คือ ค่าเฉลี่ยของสี B ในเฟรม ที่สนใจ

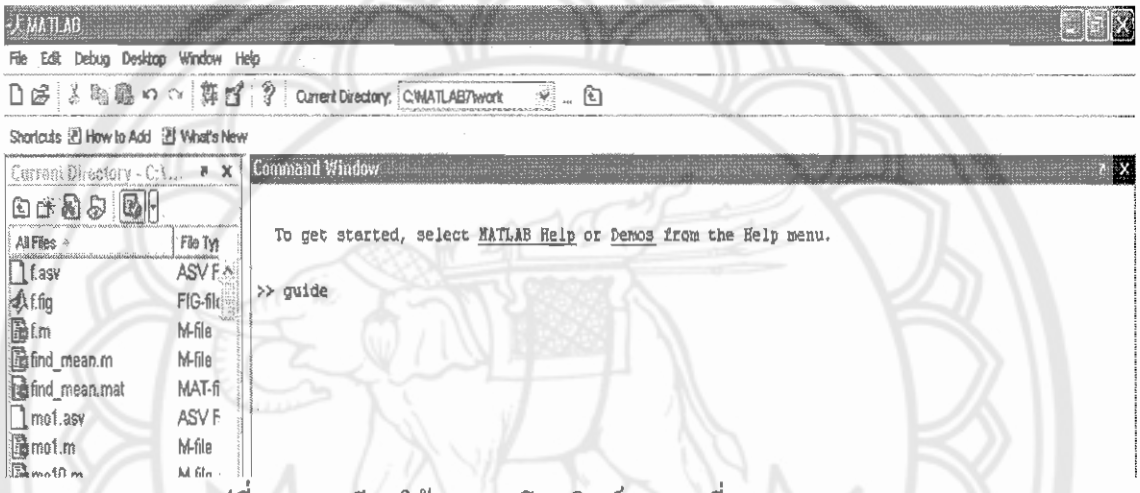
เมื่อได้ค่าผิดพลาดออกมา ก็จะทราบว่าไทล์ ใดที่เหมาะสมที่สุดหรือมีสีใกล้เคียงที่สุดเมื่อเปรียบเทียบกับเฟรมที่สนใจ

บทที่ 4

ผลการทดลอง

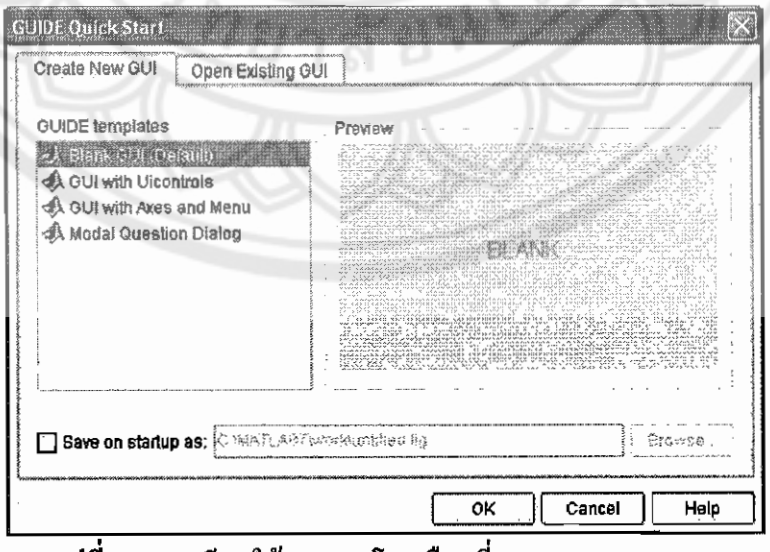
จากที่ได้ศึกษาทฤษฎี และทำการออกแบบโปรแกรมการสร้างภาพโมเสก โดยโปรแกรมจะมีไฟล์ในการใช้งานคือ f.fig f.m และ training.m ซึ่งการเรียกใช้โปรแกรมสามารถทำได้ดังนี้

1. เรียกใช้ file.fig ในโปรแกรม MATLAB โดยพิมพ์ guide ที่ command window



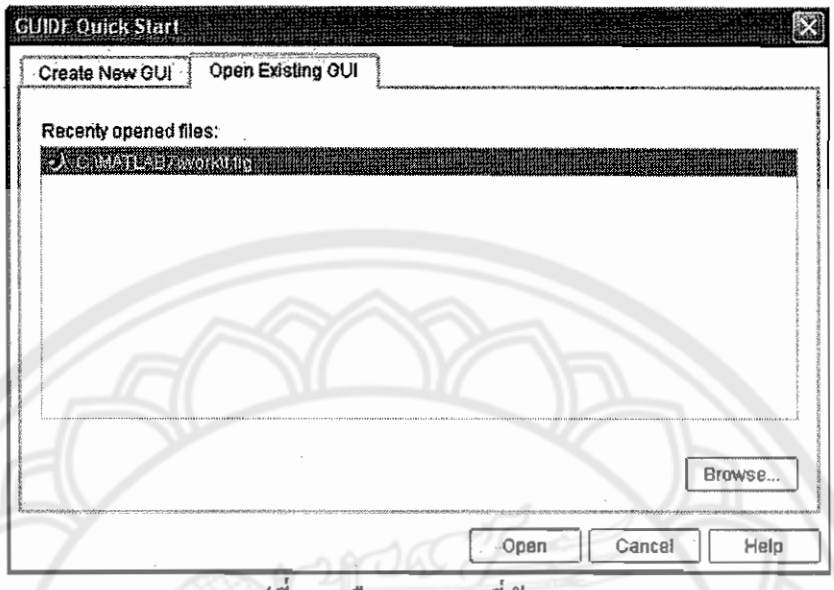
รูปที่ 4.1 การเรียกใช้ file.fig โดยพิมพ์ guide ที่ command window

2. จากนั้นเลือกที่ Open Existing GUI กด OK



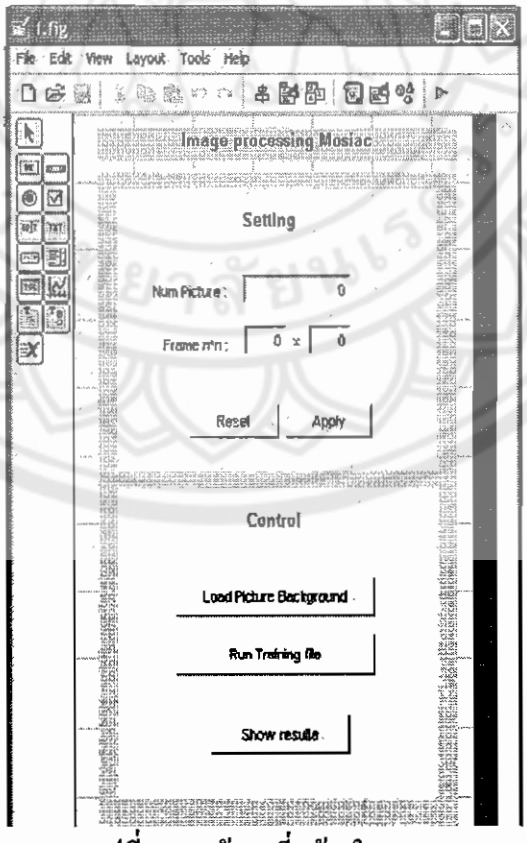
รูปที่ 4.2 การเรียกใช้ file.fig โดยเลือกที่ Open Existing GUI

3. จะเห็นว่ามี file.fig ขึ้นมาให้เลือก ซึ่งในโครงงานนี้ตั้งชื่อ file เป็น f.fig เลือก file ที่ต้องการกด open



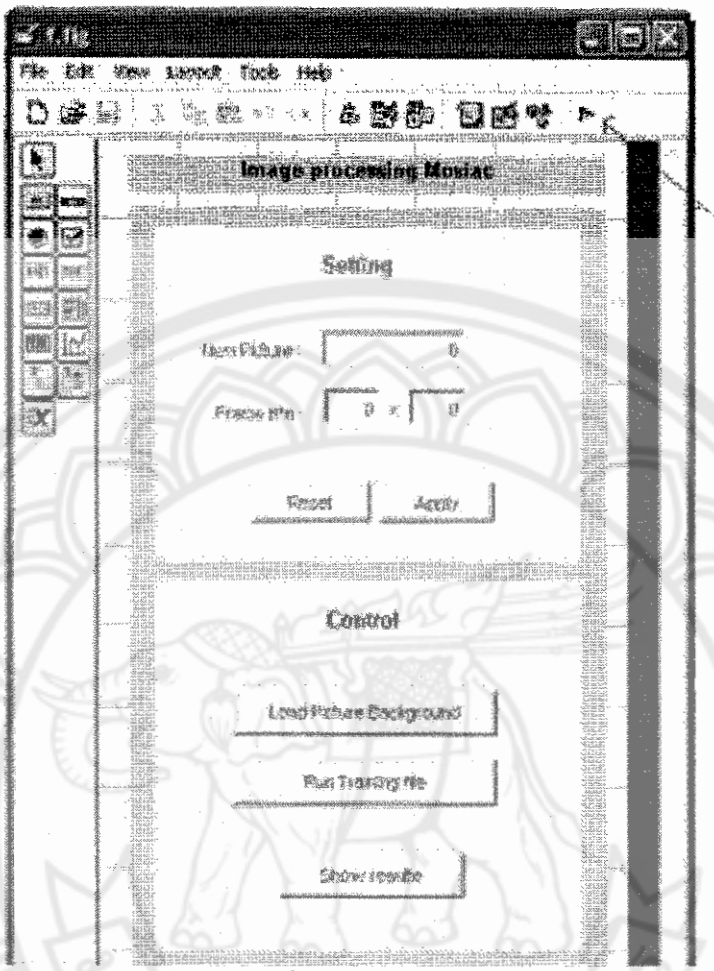
รูปที่ 4.3 เลือก file.fig ที่ต้องการ

4. โปรแกรมจะแสดงหน้าจอที่สร้างไว้ใน f.fig ขึ้นมา



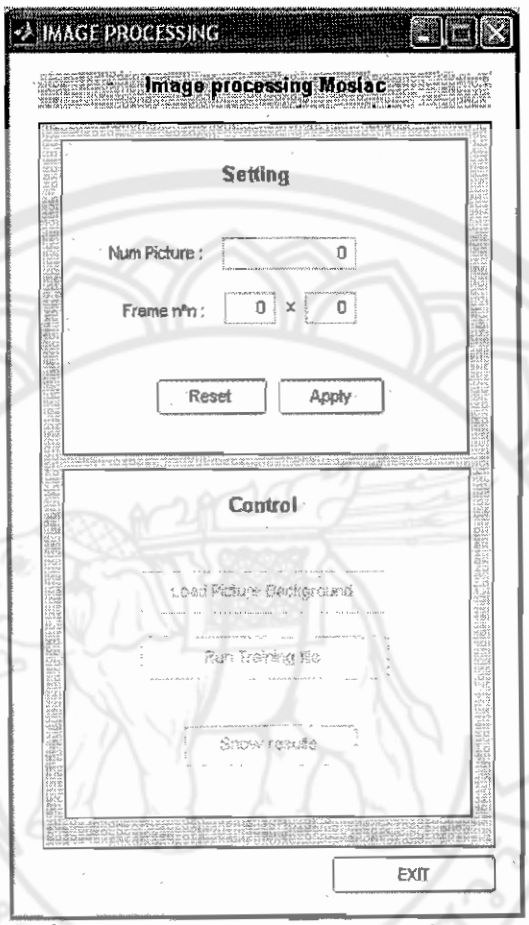
รูปที่ 4.4 หน้าจอที่สร้างใน f.fig

5. ทำการรันโปรแกรมโดยกดที่ปุ่มสามเหลี่ยมสีเขียว



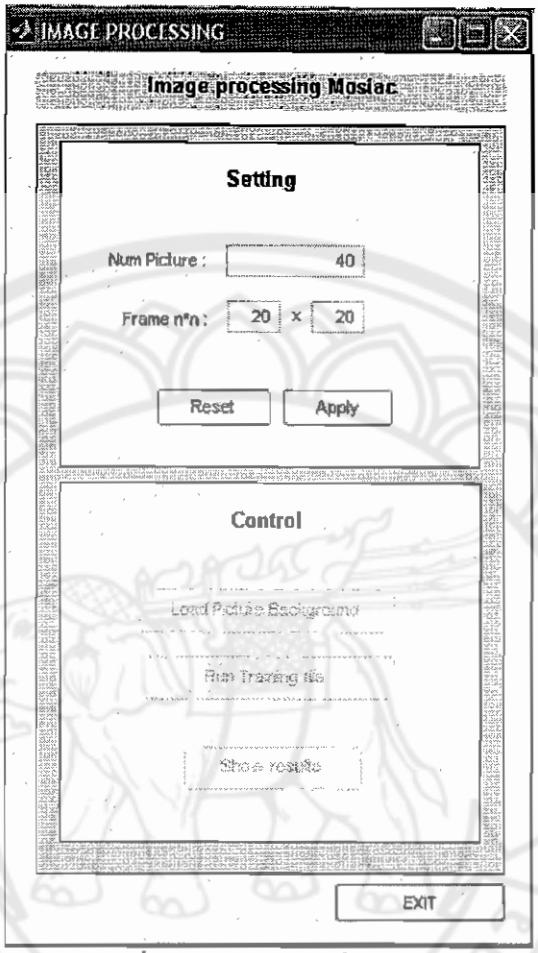
รูปที่ 4.5 ทำการรันโปรแกรม

6. โปรแกรมจะให้กรอกข้อมูลที่ต้องการโดย Num Picture คือ จำนวนไทล์ที่ต้องการใช้ และ Frame n*n คือเลือกแบ่งภาพต้นฉบับโดยช่องแรกคือจำนวนแถว ช่องที่สองคือจำนวนคอลัมน์



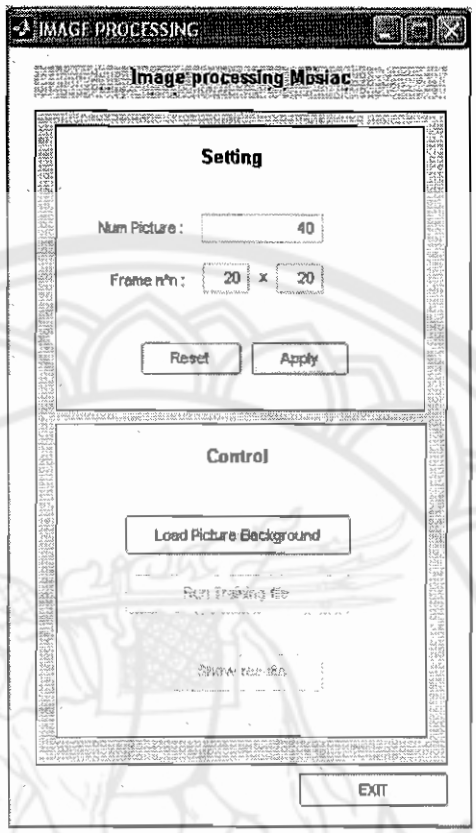
รูปที่ 4.6 โปรแกรมจะให้กรอกข้อมูลที่ต้องการ

7. ทำการกรอกข้อมูล ยกตัวอย่างเช่นใช้ไทล์ทั้งหมด 40 ภาพและแบ่งภาพต้นฉบับออกเป็น 20×20 เฟรม



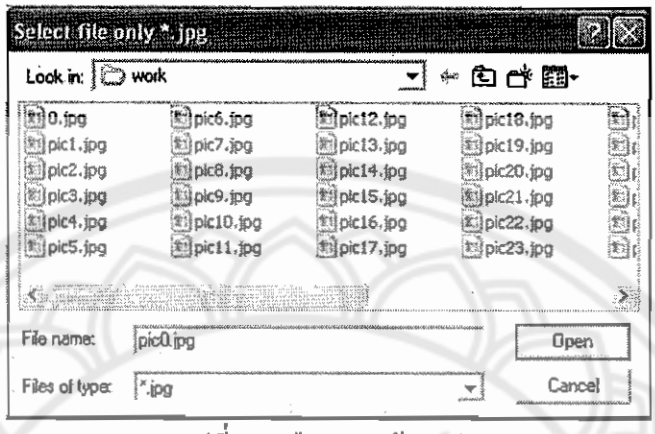
รูปที่ 4.7 กรอกข้อมูลที่ต้องการ

8. เมื่อกด Apply ในส่วน Control จะมีปุ่ม Load Picture Background ขึ้นมา กดปุ่ม Load Picture Background เพื่อเลือกภาพต้นฉบับ



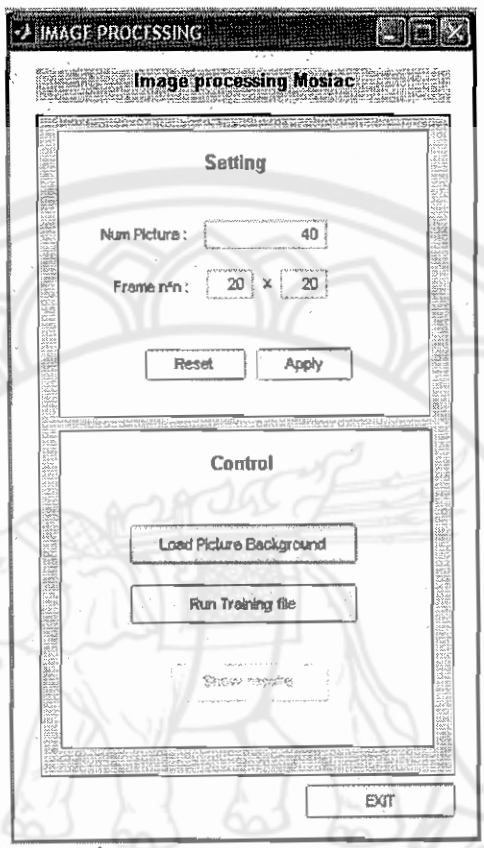
รูปที่ 4.8 กดปุ่ม Load Picture Background

9. โปรแกรมจะให้เลือกภาพต้นฉบับ เมื่อเลือกเสร็จกด Open (ในการตั้งชื่อไทล์จะต้องตั้งชื่อเป็น pic แล้วตามด้วยตัวเลขเรียงตามลำดับ เช่น pic1 pic2 pic3 ... เป็นต้น เนื่องจากในโปรแกรมออกแบบให้เลือกไทล์จาก pic1 pic2 pic3 ...)



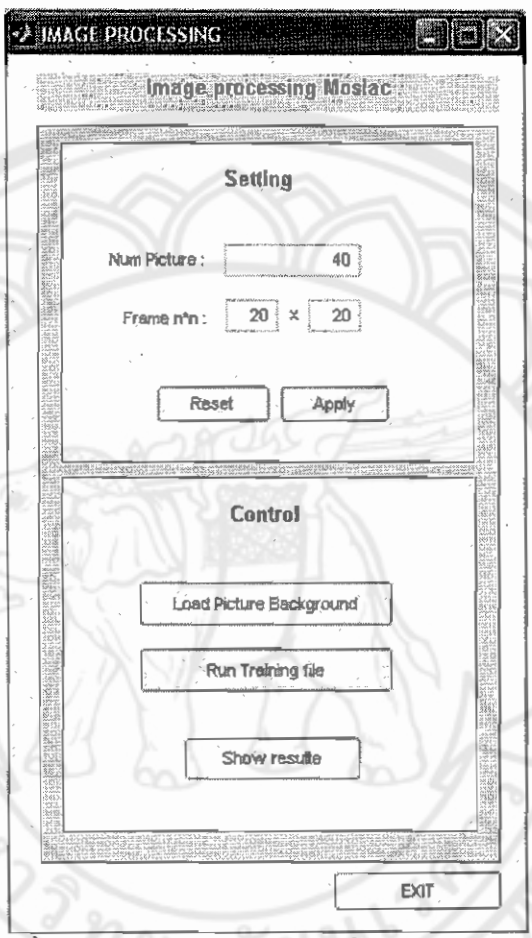
รูปที่ 4.9 เลือกภาพต้นฉบับ

10. เมื่อโปรแกรมทำการโหลดรูปภาพต้นฉบับแล้วจะมีปุ่ม Run Training file ขึ้นมา กดปุ่ม Run Training file เพื่อหาค่าเฉลี่ยของสีในแต่ละเฟรมของภาพต้นฉบับเปรียบเทียบกับไทล์ทั้งหมดที่ได้เลือกไว้ โดยโปรแกรมจะเรียกไฟล์ training.m ขึ้นมาใช้งาน



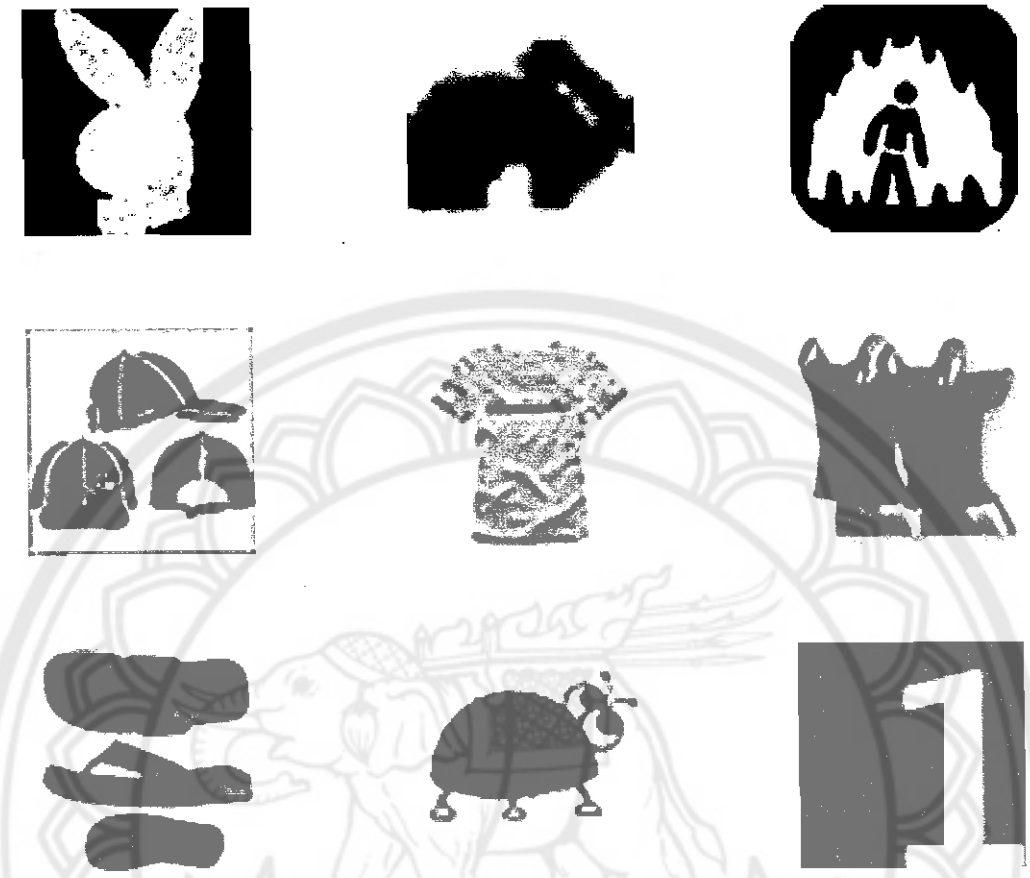
รูปที่ 4.10 กดปุ่ม Run Training file

11. เมื่อโปรแกรมเปรียบเทียบค่าเฉลี่ยของสีในแต่ละเฟรมของภาพต้นฉบับกับไทล์ทั้งหมดที่ได้เลือกไว้เรียบร้อยแล้วจะได้ค่าความผิดพลาดที่น้อยที่สุดที่เหมาะสมจะนำมาประกอบลงบนแต่ละเฟรม และจะมีปุ่ม Show result ขึ้นมาเพื่อทำการแสดงผล

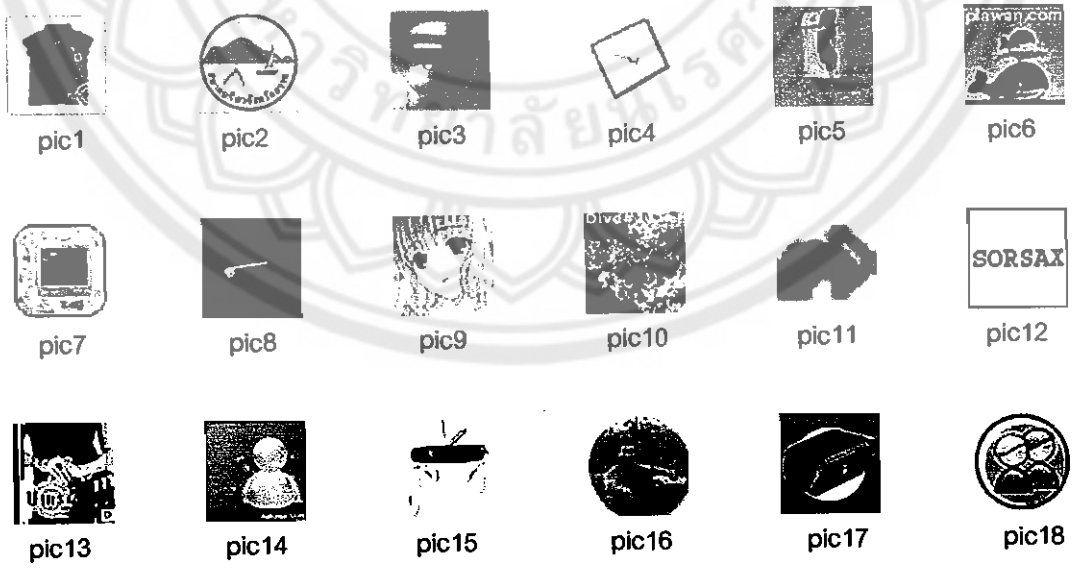


รูปที่ 4.11 กดปุ่ม Show result เพื่อทำการแสดงผล

12. ผลการทดลอง



รูปที่ 4.12 ภาพต้นฉบับ



pic19

pic20

pic21

pic22

pic23

pic24

pic25

pic26

pic27

pic28

pic29

pic30

pic31

pic32

pic33

pic34

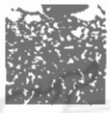
pic35

pic36

pic37

pic38

pic39


pic40

pic41

pic42

pic43

pic44

pic45

pic46

pic47

pic48

pic49

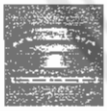
pic50

pic51

pic52

pic53

pic54


pic55

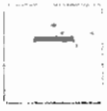
pic56

pic57

pic58

pic59

pic60


pic61

pic62

pic63

pic64

pic65

pic66

pic67

pic68

pic69

pic70

pic71

pic72

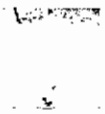
pic73

pic74

pic75

pic76

pic77


pic78

pic79

pic80

pic81

pic82

pic83

pic84

pic85

pic86

pic87

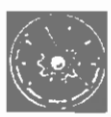
pic88

pic89

pic90

pic91

pic92


pic93

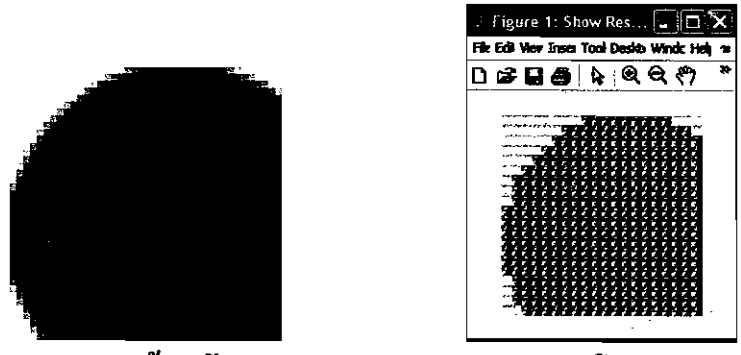
pic94

pic95

pic96

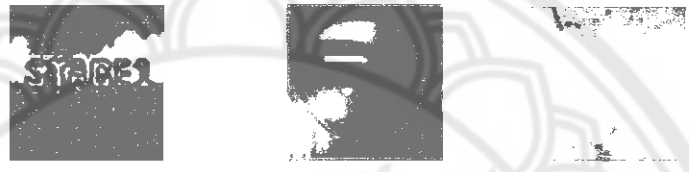
pic97

รูปที่ 4.13 ไทล์ที่นำมาประกอบภาพโมเสก



ภาพต้นฉบับ ภาพโมเสก

รูปที่ 4.14 แสดงภาพต้นฉบับและภาพโมเสกที่ 1



pic1 pic2 pic3

รูปที่4.15 แสดงภาพที่ใช้เป็นไทล์ของรูปที่ 4.14

จากรูปที่ 4.14 และ 4.15 แสดงให้เห็นว่าเมื่อแบ่งภาพต้นฉบับออกเป็น 20×20 เฟรม จะได้เฟรมทั้งหมด 400 เฟรม โดยแถวที่ 1 คอลัมน์ที่ 8 มีลักษณะของสีไปในโทนสีขาว ค่าความผิดพลาดของแถวที่ 1 คอลัมน์ที่ 8 เมื่อเปรียบเทียบกับ pic1 pic2 และ pic3 ค่าความผิดพลาดที่ได้คือ 169.42 170.65 35.88 ตามลำดับ ซึ่งค่าที่ได้นี้มาจากการคำนวณในสมการที่ 3.1 ดังนี้

$$error = \sqrt{(R_{tile} - R_{frame})^2 + (G_{tile} - G_{frame})^2 + (B_{tile} - B_{frame})^2} \quad (4.1)$$

ตัวอย่างการวิเคราะห์

แถวที่ 1 คอลัมน์ที่ 8 เปรียบเทียบกับ pic1

$$error = \sqrt{(53.6004 - 209)^2 + (167.7920 - 215.75)^2 + (196.0368 - 243.5)^2} \quad (4.2)$$

error = 169.42

แถวที่ 1 คอลัมน์ที่ 8 เปรียบเทียบกับ pic2

$$error = \sqrt{(77.0288 - 209)^2 + (133.0924 - 215.75)^2 + (173.6972 - 243.5)^2} \quad (4.3)$$

error = 170.65

แถวที่ 1 คอลัมน์ที่ 8 เปรียบเทียบกับ pic3

$$error = \sqrt{(220.1888-209)^2 + (224.0824-215.75)^2 + (210.4424-243.5)^2} \quad (4.4)$$

error = 35.88

จะเห็นว่า pic3 (รูปสีขาว)มีค่าความผิดพลาดน้อยที่สุด ดังนั้น pic3 (รูปสีขาว)จึงเป็นภาพที่เหมาะสมที่สุดในการนำมาประกอบลงในแถวที่ 1 คอลัมน์ที่ 8 ส่วนแถวที่ 1 คอลัมน์ที่ 9 มีลักษณะของสีไปในโทนสีน้ำเงิน ค่าความผิดพลาดของแถวที่ 1 คอลัมน์ที่ 9 เมื่อเปรียบเทียบกับ pic1 pic2 และ pic3 ค่าความผิดพลาดที่ได้คือ 82.38 72.67 113.28 ตามลำดับ จะเห็นว่า pic2 มีค่าความผิดพลาดน้อยที่สุด ดังนั้น pic2 จึงเป็นภาพที่เหมาะสมที่สุดในการนำมาประกอบลงในแถวที่ 1 คอลัมน์ที่ 9

รูปที่ 4.16 แสดงภาพต้นฉบับที่ 2

รูปที่ 4.17 แสดงภาพโมเสกที่ 2 เมื่อแบ่งเป็น 5×5 10×10 และ 20×20 เฟรม ตามลำดับ

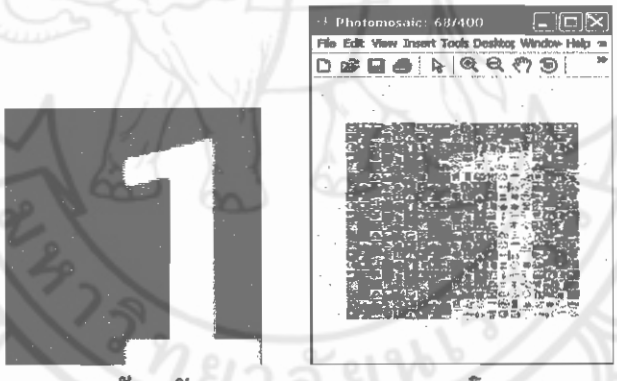
จากรูปที่ 4.16 และ 4.17 แสดงให้เห็นว่าเมื่อแบ่งภาพต้นฉบับออกเป็น 5×5 เฟรม ภาพโมเสกที่ได้ออกมาจะยังดูไม่ค่อยออกว่าเป็นรูปอะไร เมื่อแบ่งเป็น 10×10 เฟรม ภาพโมเสกที่ได้จะดูสวยงามกว่ารูปที่แบ่งเป็น 5×5 เฟรม แต่ยังมีความสวยงามน้อยกว่ารูปที่แบ่งเป็น 20×20 เฟรม และจะเห็นว่าภาพต้นฉบับมีลักษณะของสีเป็นโทนสีชมพูแต่มีระดับความเข้มของสีแตกต่างกัน ในการเลือกไทล์จึงไม่เลือกเพียงแต่ภาพที่มีสีชมพูเท่านั้น แต่จะเลือกไทล์ที่มีค่าเฉลี่ยของสี RGB ที่ใกล้เคียงที่สุดกับเฟรมนั้น ซึ่งอาจเป็นสีขาว หรือสีแดง เป็นต้น จะเห็นว่ายิ่งแบ่งกาพต้นฉบับเป็นจำนวนหลายเฟรมภาพโมเสกที่ได้ออกมาจะมีความสวยงามมากกว่าการแบ่งภาพต้นฉบับเป็นจำนวนน้อย

4900126
I 1384982
ปร
ฝ335ก
2548

ภาพต้นฉบับ ภาพโมเสก

รูปที่ 4.18 แสดงภาพต้นฉบับและภาพโมเสกที่ 3

จากรูปที่ 4.13 และ 4.18 แสดงให้เห็นว่าเมื่อแบ่งภาพต้นฉบับออกเป็น 20×20 เฟรม จะได้เฟรมทั้งหมด 400 เฟรม โดยแถวที่ 1 คอลัมน์ที่ 1 มีลักษณะของสีไปในโทนสีขาว ในการเลือกไทล์จึงเลือกไทล์ที่มีลักษณะของสีไปในโทนสีขาวเช่นเดียวกัน แต่จะเห็นว่าพื้นหลังของภาพต้นฉบับมีลักษณะของสีไปในโทนสีขาวแต่มีการกระจายของสีที่แตกต่างกัน ภาพโมเสกส่วนของพื้นหลังที่ได้ออกมาจึงไม่เพียงแต่ใช้ไทล์ที่มีลักษณะของสีไปในโทนสีขาวเท่านั้น แต่จะใช้ไทล์ที่มีค่าเฉลี่ยของสี RGB ที่ใกล้เคียงที่สุดกับเฟรมนั้น

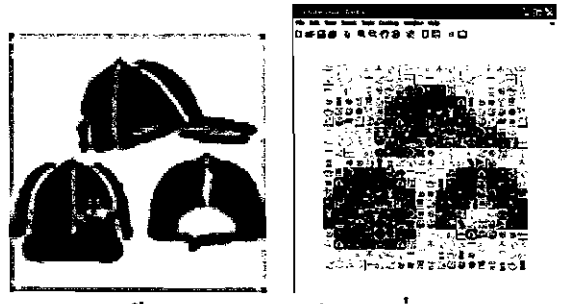


ภาพต้นฉบับ ภาพโมเสก

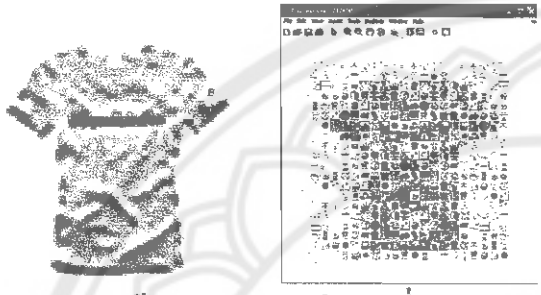
รูปที่ 4.19 แสดงภาพต้นฉบับและภาพโมเสกที่ 4

จากรูปที่ 4.13 และ 4.19 แสดงให้เห็นว่าพื้นหลังของภาพต้นฉบับมีลักษณะของสีไปในโทนสีเขียวอ่อนแต่ภาพโมเสกส่วนของพื้นหลังที่ได้ออกมากลับไม่เป็นสีเขียวอ่อนเนื่องจากไทล์ที่เลือกมาไม่มีภาพที่มีสีเขียวอ่อนเลยจะมีเพียงแต่ภาพที่มีสีใกล้เคียงกับสีเขียวอ่อน ในการเลือกไทล์มาประกอบลงบนแต่ละเฟรมจึงเลือกไทล์ที่มีค่าเฉลี่ยของสี RGB ที่ใกล้เคียงที่สุดกับเฟรมนั้น

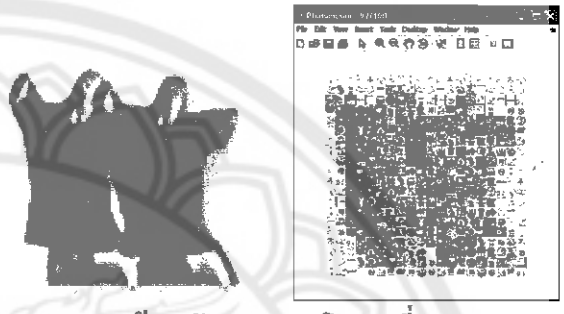
ภาพต้นฉบับและภาพโมเสกที่ 5



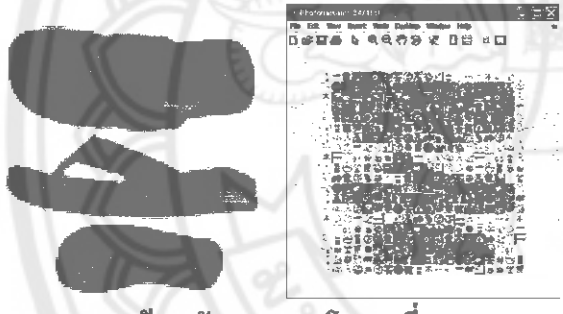
ภาพต้นฉบับและภาพโมเสกที่ 6



ภาพต้นฉบับและภาพโมเสกที่ 7



ภาพต้นฉบับและภาพโมเสกที่ 8



ภาพต้นฉบับและภาพโมเสกที่ 9

ภาพต้นฉบับและภาพโมเสกที่ 10

รูปที่4.20 แสดงการเปรียบเทียบระหว่างภาพต้นฉบับและภาพโมเสก

จากรูปที่ 4.20 แสดงให้เห็นว่าภาพต้นฉบับมีส่วนที่มีรายละเอียดของภาพอยู่ แต่ภาพโมเสกที่ได้ออกมานั้นจะไม่เห็นรายละเอียดชัดเจน เช่น ในรูปภาพต้นฉบับและภาพโมเสกที่ 6 (รูปหมวก) ส่วนที่เป็นลายสีขาว ภาพโมเสกที่ได้จะไม่เห็นเป็นลายสีขาวเหมือนในภาพต้นฉบับ เนื่องจากการแบ่งเฟรมได้แบ่งเป็นจำนวน 20×20 เฟรม ซึ่งทำให้ภาพโมเสกที่ได้ออกมายังไม่เห็นรายละเอียดชัดเจน และมีความสวยงามน้อยกว่าการแบ่งภาพต้นฉบับเป็นจำนวนหลายๆเฟรม หรืออย่างเช่นในรูปภาพต้นฉบับและภาพโมเสกที่ 10 (รูปแมลง) ในส่วนที่เป็นขาและหนวด ภาพโมเสกที่ได้ออกมาจะไม่เห็นชัดเจนว่าเป็นขาและหนวด และในส่วนของลายสีดำบนตัวแมลงซึ่งมีลักษณะเป็นทรงกลม แต่ภาพโมเสกที่ได้ออกมาจะไม่เป็นทรงกลม เนื่องจากว่าการแบ่งเฟรมยังแบ่งเป็นจำนวนน้อย ถ้ามีการแบ่งเฟรมเป็นจำนวนมากก็จะเห็นลายสีดำบนตัวแมลงเป็นทรงกลมใกล้เคียงกับภาพต้นฉบับมากขึ้น

บทที่ 5
สรุปผล

5.1 สรุปผล

โครงงานนี้เป็นโครงงานในรูปแบบของการประมวลผลภาพ จัดทำขึ้นโดยมีจุดประสงค์เพื่อดึงดูด ความสนใจของผู้พบเห็น เนื่องจากภาพโมเสกเป็นภาพที่มีความสวยงามและสามารถนำไปใช้ประโยชน์ได้มากมายเช่นในสื่อโฆษณา จากผลการทดลองพบว่า เมื่อนำไทล์ ที่มีลักษณะของสีแตกต่างกันจำนวนหลายภาพมาเป็นไทล์ที่จะประกอบลงในภาพต้นฉบับ ภาพโมเสกที่ได้ออกมาจะมีความใกล้เคียงกับภาพต้นฉบับมากกว่าการใช้ไทล์ จำนวนน้อย

5.2 ข้อเสนอแนะ

การสร้างภาพโมเสกที่สวยงามนั้นต้องแบ่งภาพต้นฉบับให้มีจำนวนเฟรมมากๆ ยิ่งแบ่งมากเท่าไรภาพที่ได้ออกมาก็จะยิ่งมีความสวยงามมากขึ้นเท่านั้น และไทล์ ควรมีหลากหลาย จำนวนมาก และควรมีการกระจายของภาพให้ครอบคลุมช่วงของค่าข้อมูล เพื่อให้ภาพผลลัพธ์ที่ออกมา มีแนวโน้มไปในทิศทางที่ต้องการมากที่สุด

5.3 ปัญหาที่พบในการทำโครงงาน

5.3.1 เมื่อแบ่งภาพต้นฉบับเป็นจำนวนเฟรมมากๆ จะทำให้การประมวลผลช้า

5.3.2 เมื่อแบ่งภาพต้นฉบับเป็นจำนวนเฟรมน้อยๆ ภาพที่ได้ออกมาจะดูไม่ออกว่าเป็นภาพอะไร และไม่มีความสวยงาม

5.3.3 ใช้การเปรียบเทียบค่าเฉลี่ยของสี RGB เพียงอย่างเดียวในการค้นหาภาพที่ดีที่สุดเพื่อนำมาแปะลงบนแต่ละเฟรมยังได้ภาพผลลัพธ์ไม่ดีเท่าที่ควร

5.3.4 ไทล์ยังมีจำนวนน้อย ไม่หลากหลาย ทำให้ภาพผลลัพธ์ที่ได้ออกมายังไม่ดีเท่าที่ควร

5.3.5 เมื่อใช้ภาพที่มีขนาดใหญ่ เช่น 800×600 พิกเซลจะทำให้การประมวลผลช้ากว่าภาพที่มีขนาดเล็ก เช่น 50×50 พิกเซล

5.4 แนวทางแก้ไข

5.4.1 ใช้การเปรียบเทียบค่าเฉลี่ยของสี RGB รวมกับวิธีการอื่นเช่น การเลือกไทล์ ที่มีค่าเฉลี่ยของสีใกล้เคียงส่วนของภาพมาชุดหนึ่ง แล้วเลือกใช้ไทล์ ที่มีค่าเบี่ยงเบนมาตรฐานใกล้เคียงกับส่วนของภาพที่สุดมาใช้ในการค้นหาภาพที่ดีที่สุดเพื่อนำมาแปะลงบนแต่ละเฟรม[3] เพื่อให้ภาพผลลัพธ์ที่ได้มีความสวยงามมากยิ่งขึ้น

5.4.2 ไทล์ ควรมีหลากหลาย จำนวนมาก และควรมีการกระจายของภาพให้ครอบคลุมช่วงของค่าข้อมูล เพื่อให้ภาพผลลัพธ์ที่ออกมา มีแนวโน้มไปในทิศทางที่ต้องการมากที่สุด

5.4.3 ใช้ภาพที่มีขนาดเล็กเพื่อให้การประมวลผลเร็วยิ่งขึ้นแต่ก็ลดความสวยงามดังนั้นอาจมีการหาจุดและสมดุลที่สุดในการสร้างภาพโมเสก

เอกสารอ้างอิง

[1] รศ.ดร.มนัส สังวรศิลป์.วรรัตน์ ภัทรอมรกุล.**คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์**.นนทบุรี :สำนักพิมพ์ อินโฟเพรส.2543

[2] olej."Mosaic Creater." [Online]. Available: http://aolej.com/mosaic/index.html.2005

[3] u4511088."เทคนิคการสร้างภาพโมเสก." [Online].เข้าถึงได้จาก: http://www.thaitechinfo.com/article/mosaics.html.2546.

[4] "การค้นหาภาพ." [Online].เข้าถึงได้จาก: http://202.28.94.55/web/320417/2548/work1/g25/technoreport1.htm.2548

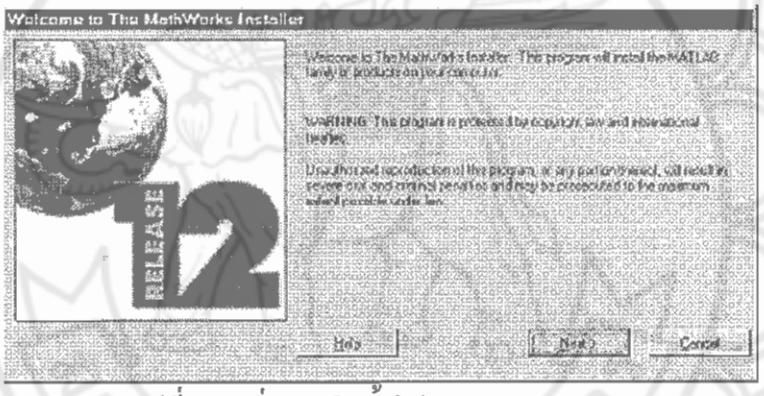
ภาคผนวก ก.

คู่มือการติดตั้งและการใช้งานของโปรแกรม MATLAB

ในโครงงานนี้ใช้โปรแกรม MATLABในการสร้างภาพโมเสก ซึ่งการติดตั้งโปรแกรม MATLAB สามารถทำได้ดังนี้

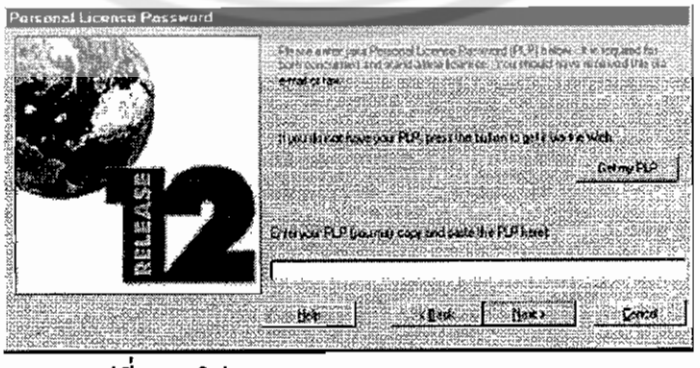
1. ขั้นตอนการติดตั้งโปรแกรม

1.1 เริ่มการติดตั้ง โดยการใส่ CD-ROM หรือ ดาวน์โหลด(download) จากเว็บไซต์(Web site) MathWorks แล้ว double-click บน installer file จะได้หน้าต่าง Welcome to The MathWorks Installer จากนั้นคลิกที่ Next



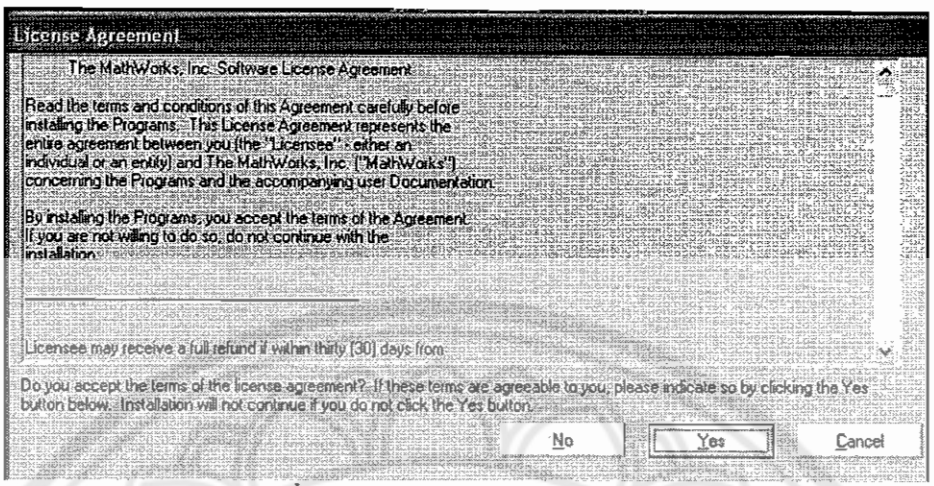
รูปที่ ก-1 เริ่มการติดตั้งโปรแกรม MATLAB

1.2 เมื่อคลิกที่ Next แล้วจะปรากฏหน้าต่างดังรูปที่ ก-2 จากนั้นใส่ Personal License Password (PLP) ซึ่งได้จาก ไฟล์ Serrial.txt โดย PLP จะมีลักษณะที่เริ่มต้นด้วยตัวเลข 2 หลักแล้วตามด้วยตัวเลข 5 หลัก เช่น 12-12345-12345-12345-12345-12345... เมื่อใส่ PLP เรียบร้อยแล้วคลิก Next



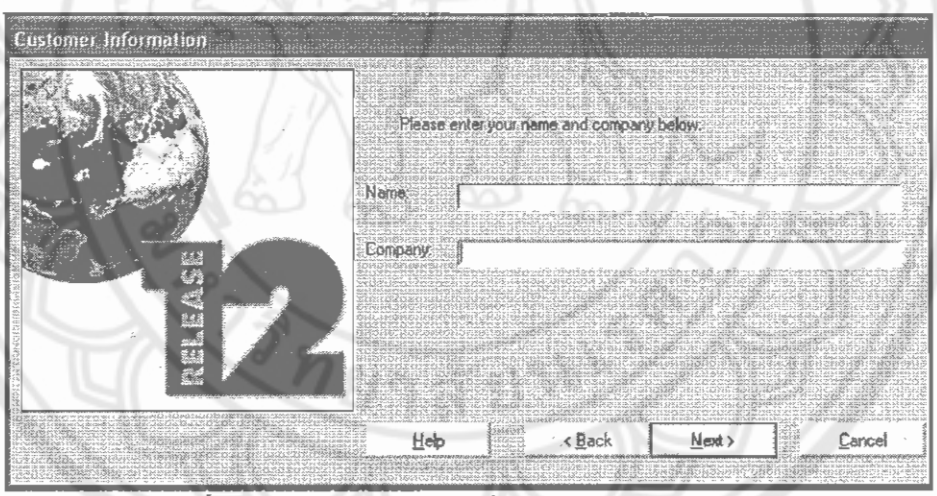
รูปที่ ก-2 ใส่ Personal License Password (PLP)

1.3 จากนั้นจะปรากฏหน้าต่าง License Agreement ให้คลิก Yes เพื่อทำการ install ต่อ



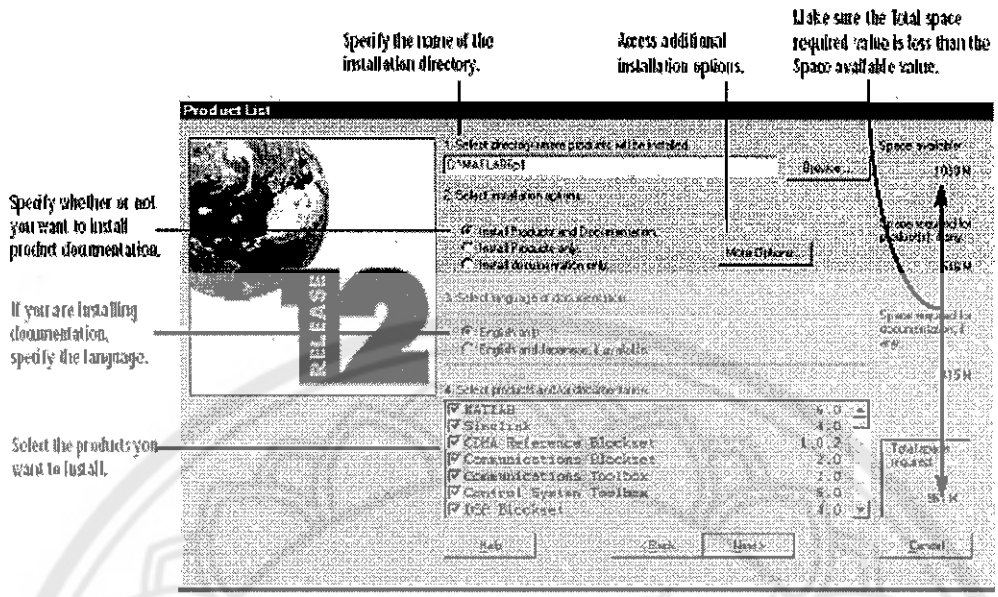
รูปที่ ก-3 หน้าต่าง License Agreement

1.4 ใส่ชื่อ(Name) และชื่อบริษัท(company name) ในหน้าต่าง Customer Information จากนั้นคลิก Next



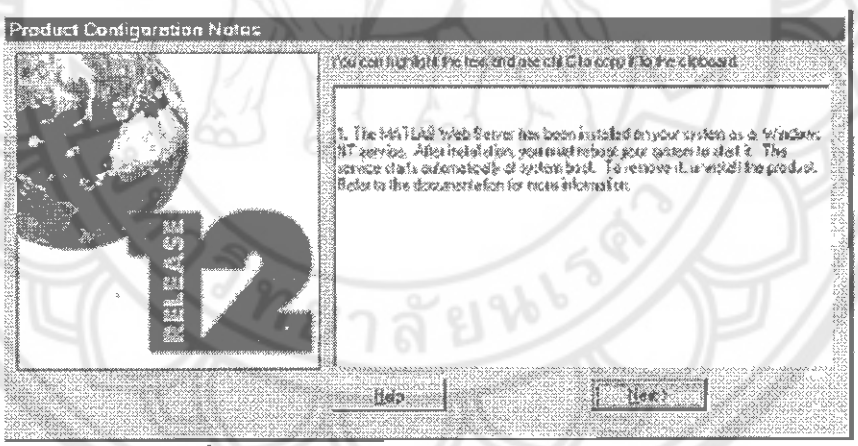
รูปที่ ก-4 ใส่ชื่อ(Name) และชื่อบริษัท(company name)

1.5 เลือก Product ที่ต้องการ install จากหน้าต่าง Product List จากนั้นคลิก Next



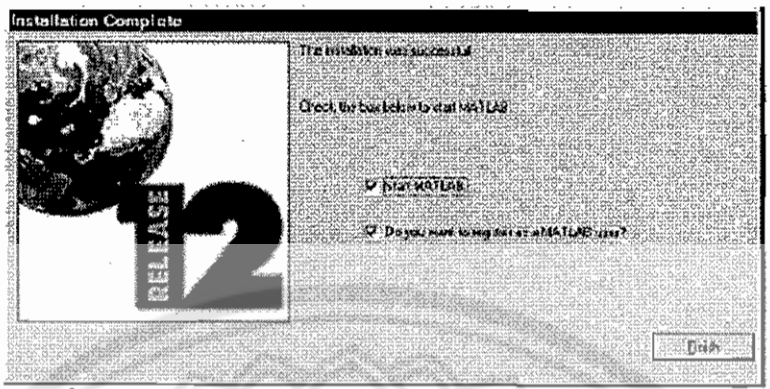
รูปที่ ก-5 เลือก Product

1.6 โปรแกรมจะทำการ install จนปรากฏหน้าต่าง Product Configuration Notes จากนั้นคลิก Next



รูปที่ ก-6 หน้าต่าง Product Configuration Notes

1.7 เมื่อทำการ install เสร็จเรียบร้อยแล้วจะปรากฏหน้าต่าง Installation Complete คลิก Finish



รูปที่ ก-7 หน้าต่าง Installation Complete เมื่อทำการ install เสร็จ

2. ตัวอย่างการใช้งาน

ในโปรแกรม MATLAB นี้มีคำสั่งที่ช่วยในการจัดการเกี่ยวกับรูปภาพมากมาย ซึ่งสามารถเรียกดูคำสั่งต่างๆได้โดยการพิมพ์คำสั่ง help images ใน command windows โปรแกรมจะแสดงคำสั่งต่างๆ พร้อมทั้งคำอธิบายคำสั่งอย่างย่อ ยกตัวอย่างดังนี้

Image display.

colorbar - Display colorbar (MATLAB Toolbox).

getimage - Get image data from axes.

image - Create and display image object (MATLAB Toolbox).

imagesc - Scale data and display as image (MATLAB Toolbox).

immovie - Make movie from multiframe image.

imshow - Display image in Handle Graphics figure.

imview - Display image in the image viewer.

montage - Display multiple image frames as rectangular montage.

movie - Play recorded movie frames (MATLAB Toolbox).

subimage - Display multiple images in single figure.

truesize - Adjust display size of image.

warp - Display image as texture-mapped surface.

Image file I/O.

dicominfo - Read metadata from a DICOM message.

dicomread - Read a DICOM image.

dicomuid - Generate DICOM unique identifier.

dicomwrite - Write a DICOM image.

dicom-dict.txt - Text file containing DICOM data dictionary.

imfinfo - Return information about image file (MATLAB Toolbox).

imread - Read image file (MATLAB Toolbox).

imwrite - Write image file (MATLAB Toolbox).

โปรแกรม MATLAB นี้ยังสามารถอธิบายรายละเอียดของแต่ละคำสั่ง รวมทั้งมีตัวอย่างการใช้งานให้ดูอีกด้วย เช่น ผู้ใช้ต้องการสร้างภาพโดยการซ้อนภาพถ่ายหลายๆภาพ สามารถทำได้โดยการใช้คำสั่ง montage และผู้ใช้สามารถทราบรายละเอียดรวมทั้งตัวอย่างการใช้งานได้ โดยพิมพ์คำสั่ง help montage ใน command windows โปรแกรมจะอธิบายคำสั่งและมีตัวอย่างการใช้งาน พร้อมทั้งมีคำสั่งที่ใช้งานในลักษณะเดียวกันให้อีกด้วย ดังนี้

```
>> help montage
  MONTAGE Display multiple image frames as rectangular montage.
  MONTAGE displays all the frames of a multiframe image array
  in a single image object, arranging the frames so that they
  roughly form a square.
 MONTAGE(I) displays the K frames of the intensity image array
  I. I is M-by-N-by-1-by-K.
 MONTAGE(BW) displays the K frames of the binary image array
  BW. BW is M-by-N-by-1-by-K.
 MONTAGE(X,MAP) displays the K frames of the indexed image
  array X, using the colormap MAP for all frames. X is
  M-by-N-by-1-by-K.
 MONTAGE(RGB) displays the K frames of the truecolor image
  array RGB. RGB is M-by-N-by-3-by-K.
 H = MONTAGE(...) returns the handle to the image object.
 Class support
  -------------
  The input image can be logical, uint8, uint16, or double.  The map must
  be double.  The output is a handle to the graphics objects produced by
  this function.
 Example
  -------
```

```
load mri
montage(D,map)
```

See also IMMOVIE.