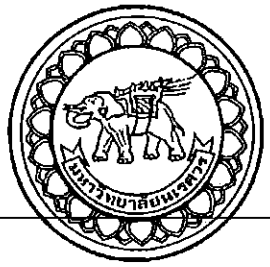


# ภาพโมเสก

# IMAGE MOSAIC


นายพิชญา ประทุมศิริ รหัส 45360302

นายอมรเทพ มหิวรรณ รหัส 45360591



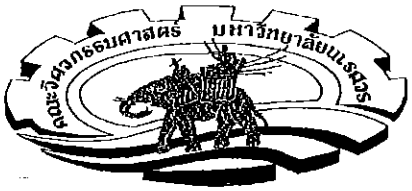
ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 2 มิ.ย. 2553
เลขทะเบียน 15000294
เลขเรียกหนังสือ ปร.
มหาวิทยาลัยนเรศวร ฟ. 639ภ
2549


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2549




# ใบรับรองโครงงานวิศวกรรม

| | | | |
|---|---|---|---|
| **หัวข้อโครงงาน** | ภาพโมเสก | | |
| **ผู้ดำเนินโครงงาน** | นายพิชญา | ประทุมศิริ | รหัส 45360302 |
| | นายอมรเทพ | มหิวรรณ | รหัส 45360591 |
| **อาจารย์ที่ปรึกษา** | ดร.พนมขวัญ | ริยะมงคล | |
| **สาขาวิชา** | วิศวกรรมคอมพิวเตอร์ | | |
| **ภาควิชา** | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ | | |
| **ปีการศึกษา** | 2549 | | |

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

คณะกรรมการการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ
(ดร.พนมขวัญ ริยะมงคล)

.....กรรมการ
(ผศ.ดร.สุชาติ แย้มเม่น)

.....กรรมการ
(ดร.อัครพันธ์ วงศ์กังแห)

| | | | |
|---|---|---|---|
| **หัวข้อโครงงาน** | ภาพโมเสก | | |
| **ผู้ดำเนินโครงงาน** | นายพิชญา | ประทุมศิริ | รหัส 45360302 |
| | นายอมรเทพ | มหิวรรณ | รหัส 45360591 |
| **อาจารย์ที่ปรึกษา** | ดร.พันมขวัญ | ริยะมงคล | |
| **สาขาวิชา** | วิศวกรรมคอมพิวเตอร์ | | |
| **ภาควิชา** | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ | | |
| **ปีการศึกษา** | 2549 | | |


## บทคัดย่อ

การนำเสนอข้อมูลโดยใช้รูปภาพมีความจำเป็นและใช้กันอย่างแพร่หลาย ซึ่งในบางครั้งภาพถ่ายเพียงภาพเดียวไม่สามารถสื่อถึงรายละเอียดได้ครอบคลุมสิ่งที่ต้องการนำเสนอ จึงต้องแสดงภาพหลาย ๆ ภาพ

โครงงานนี้ได้ศึกษาและพัฒนาโปรแกรมภาพโมเสก สำหรับสร้างภาพโมเสกจากภาพ 2 ภาพที่มีองค์ประกอบเหมือนกัน โดยใช้ทฤษฎีและหลักการของการหาค่าแอบโซลูท เออเรอและทฤษฎีสี ในโครงงานนี้ได้พัฒนาด้วยโปรแกรมภาษา แมทแลบ และมีการพัฒนาแอพพลิเคชันให้การใช้งานมีความสะดวกและรวดเร็ว

จากผลการทดลองโครงงานนี้ทำให้เราสามารถสร้างภาพโมเสกจากภาพ 2 ภาพที่องค์ประกอบเหมือนกันโดยที่ขนาดของภาพไม่จำเป็นต้องเท่ากันได้

| | | |
|---|---|---|
| **Project title** | Image Mosaic | |
| **Name** | Mr. Pichaya Pratoomsiri | ID. 45360302 |
| | Mr. Amornthep Mahiwan | ID. 45360591 |
| **Project advisor** | Dr. Panomkhawn Riyamongkol | |
| **Major** | Computer Engineering | |
| **Department** | Electrical and Computer Engineering | |
| **Academic year** | 2006 | |


## Abstract

The presentation by using picture is necessary and wide spread. Sometimes, the only one picture is not enough to show details in the presentation.

This project studied and developed a program for image mosaic and create an image mosaic from 2 similar pictures. Absolute Error and color theory were used in this project. This project used matlab as a tool in developing a program to be fast and easy to use.

From the experiment, the project result can get a complete image mosaic from 2 similar pictures which have not equal sizes.

# กิตติกรรมประกาศ

โครงงานฉบับนี้คงไม่อาจสำเร็จลงได้ หากไม่ได้รับความช่วยเหลือ และการให้คำแนะนำปรึกษาจาก ท่านอาจารย์ ดร.พนมขวัญ ริยะมงคล ขอขอบคุณมากครับ

ขอขอบคุณอาจารย์ทุกท่านในภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยนเรศวร ที่ได้สั่งสอนและให้ความรู้แก่เรา ซึ่งสามารถนำไปประยุกต์ใช้ในการทำโครงงานนี้ได้อย่างมากมาย

และที่สุดคือพวกเราเอง ต้องขอบคุณซึ่งกันและกันที่เป็นเพื่อนร่วมงานที่ดี จนทำให้โครงงานนี้สำเร็จลงได้

นายพิชญา ประทุมศิริ

นายอมรเทพ มหิวรรณ

# สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ฉ

สารบัญรูป ..... ช

**บทที่ 1** บทนำ

1.1 ความเป็นมาและความสำคัญของโครงงาน ..... 1

1.2 วัตถุประสงค์ของโครงงาน ..... 1

1.3 ขอบเขตของโครงงาน ..... 1

1.4 ขั้นตอนการดำเนินงาน ..... 1

1.5 แผนการดำเนินงาน ..... 2

1.6 ผลที่คาดว่าจะได้รับ ..... 2

1.7 งบประมาณที่ใช้ ..... 3

**บทที่ 2** ทฤษฎีและหลักการ

2.1 ทฤษฎีที่ใช้ ..... 4

2.1.1 Color Histrogram ..... 4

2.1.2 Absolute Error ..... 8

2.2 การพัฒนาแอพพลิเคชัน ..... 9

2.2.1 การพัฒนาแอพพลิเคชันด้วย matlab ..... 10

2.2.2 จุดเด่นของ matlab ..... 10

**บทที่ 3 วิธีการดำเนินการ**

3.1 การออกแบบโปรแกรม ..... 13

3.2 การออกแบบแอพพลิเคชัน ..... 18

# สารบัญ (ต่อ)

หน้า

**บทที่ 4 ผลการทดลอง**

4.1 การทดสอบการรวมภาพที่ 1 ..... 21

4.2 การทดสอบการรวมภาพที่ 2 ..... 25

**บทที่ 5 สรุปผลและข้อเสนอแนะ**

5.1 สรุปผลการทดลอง ..... 28

5.2 ปัญหาในการทดลอง ..... 28

5.3 ข้อเสนอแนะ ..... 28

**บรรณานุกรม** ..... 29

**ประวัติผู้เขียนโครงงาน** ..... 30

# สารบัญตาราง

ตารางที่ หน้า

1.1 แสดงขั้นตอนการดำเนินงาน ..... 2

# สารบัญรูป


| รูปที่ | หน้า |
|---|---|
| 2.1 ภาพแสดงระบบสีของ RGB | 5 |
| 2.2 ภาพแสดงกราฟ 3 มิติในรูปกรวย | 6 |
| 2.3 ภาพแสดงการเกี่ยวข้องของ RGB กับ HSV | 7 |
| 2.4 ภาพแสดงค่าของพิกเซลรูปที่ 1 | 8 |
| 2.5 ภาพแสดงค่าของพิกเซลรูปที่ 2 | 8 |
| 2.6 ภาพแสดงโปรแกรม matlab | 10 |
| 2.7 ภาพแสดงตัวอย่างการเขียนโปรแกรม matlab | 11 |
| 3.1 ภาพแสดงแผนผังการทำงานของโปรแกรม | 13 |
| 3.2 ภาพแสดงแบบฟอร์มของโปรแกรม | 18 |
| 3.3 ภาพแสดงแบบฟอร์มหน้าจอใส่ภาพ | 19 |
| 3.4 ภาพแสดงแบบฟอร์มหน้าจอแสดงกราฟการทำงาน | 20 |
| 4.1 ภาพแสดงหน้าจอเริ่มต้นของโปรแกรม | 21 |
| 4.2 ภาพแสดงภาพหน้าจอโปรแกรมเมื่อใส่ภาพอินพุต | 22 |
| 4.3 ภาพแสดงภาพที่ได้หลังจากการรวม | 23 |
| 4.4 ภาพแสดงรูปกราฟการทำงาน | 24 |
| 4.5 ภาพแสดงภาพหน้าจอโปรแกรมเมื่อใส่ภาพอินพุต | 25 |
| 4.6 ภาพแสดงภาพที่ได้หลังจากการรวม | 26 |
| 4.7 ภาพแสดงรูปกราฟการทำงาน | 27 |

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากในปัจจุบัน การนำเสนอข้อมูลโดยใช้ภาพนิ่งมีความจำเป็นค่อนข้างสูง เพราะว่าภาพนิ่งเป็นสื่อที่สามารถเข้าใจได้ง่าย จึงมีการใช้กันอย่างแพร่หลาย ซึ่งในบางครั้งการแสดงผลโดยใช้ภาพนิ่ง ก็จะพบปัญหาเช่น ภาพถ่ายเพียงภาพเดียวไม่สามารถสื่อถึงรายละเอียดได้ครอบคลุมสิ่งที่ต้องการนำเสนอ ซึ่งบางครั้งถ้าต้องถ่ายภาพหลาย ๆ ภาพ ยกตัวอย่างเช่น ภาพถ่ายทางดาวเทียมหรือภาพถ่ายทางอากาศ หรือแม้แต่ภาพถ่ายแบบพานอรามา(Panorama) เป็นต้น และจากปัญหานี้จึงนำมาเพื่อแก้ไขและพัฒนา โดยใช้หลักการของภาพโมเสก(Mosaic)ซึ่งจะเป็นการนำภาพหลาย ๆ ภาพมาเชื่อมต่อกันในส่วนที่เหมือนกันให้เป็นภาพรวมภาพเดียว และอาศัยเทคนิคทางด้านอิมเมจโปรเซสซิ่ง(Image Processing) โดยใช้หลักการของการเปรียบเทียบความเหมือนโดยการหาค่า แอบโซลูท เออเรอ(Absolute Error) เข้ามาช่วย

## 1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาหลักการเปรียบเทียบความเหมือนโดยการหาค่า แอบโซลูท เออเรอ แล้วนำมาประยุกต์ใช้ในโครงงาน

1.2.2 เพื่อศึกษาวิธีการใช้โปรแกรม แมทแลบ (Matlab)

1.2.3 เพื่อให้ได้ภาพโมเสก จากรูป 2 รูปที่มีองค์ประกอบเหมือนกัน

## 1.3 ขอบข่ายของโครงงาน

เพื่อศึกษาและสามารถนำภาพ 2 ภาพ ที่มีองค์ประกอบของภาพเหมือนกันมารวมกัน โดยที่ภาพที่ได้จากการรวมนั้น ไม่ขัดต่อการแสดงผลภาพและสามารถแสดงรายละเอียดของภาพได้ครบถ้วนและสมบูรณ์

## 1.4 ขั้นตอนการดำเนินงาน

1.4.1 รวบรวมข้อมูล

1.4.2 ศึกษาทฤษฎี และโปรแกรมที่ใช้ในโครงงาน

1.4.3 ออกแบบโปรแกรม

1.4.4 พัฒนาโปรแกรมและแอพพลิเคชัน

1.4.5 ทดสอบโปรแกรม

1.4.6 วิเคราะห์และสรุปผล

## 1.5 แผนการดำเนินงาน

| กิจกรรม | เดือน – ปี | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ธ.ค.46 | ม.ค.47 | ก.พ.47 | มี.ค.47 | เม.ย.47 | พ.ค.47 | มิ.ย.47 | ก.ค.47 | ส.ค.47 | ก.ย.47 |
| 1. รวบรวมข้อมูล | ← | → | | | | | | | | |
| 2. ศึกษาทฤษฎีและโปรแกรมที่ใช้ในโครงงาน | | | ← | → | | | | | | |
| 3. ออกแบบโปรแกรม | | | | | ← | → | | | | |
| 4. พัฒนาโปรแกรม | | | | | | | ← | → | | |
| 5. ทดสอบโปรแกรม | | | | | | | | ← | → | |
| 6. วิเคราะห์และสรุปผล | | | | | | | | | ← | → |

ตารางที่ 1.1 ขั้นตอนการดำเนินโครงงาน

## 1.6 ผลที่คาดว่าจะได้รับ

1.6.1 ได้รับความรู้เกี่ยวกับ

- ทฤษฎีพื้นฐานของทฤษฎีสี และแอบโซลูทเออเรอ
- วิธีการใช้โปรแกรมแมทแลบ
- วิธีการสร้างอัลกอลิทึ่ม เพื่อใช้ในโครงงาน ภาพโมเสก

1.6.2 สามารถที่จะ

- นำทฤษฎีสีและแอบโซลูทเออเรอ มาประยุกต์ ใช้ในโครงงาน ภาพโมเสกได้
- ใช้โปรแกรม แมทแลบในการรวมภาพ 2 ภาพเข้าด้วยกันได้
- สร้างและพัฒนาโปรแกรมภาพโมเสกให้ได้ผลลัพธ์ ตามที่ต้องการได้

1.6.3 ผู้ที่สนใจสามารถนำไปศึกษาและพัฒนาไปใช้ในงานด้านต่างๆ ต่อไปได้

**1.7 งบประมาณที่ใช้**

| | | |
|---|---|---|
| 1.7.1 ค่าจัดพิมพ์รายงาน | 500 | บาท |
| 1.7.2 ค่าหมึกพิมพ์ | 1,000 | บาท |
| 1.7.3 ค่าถ่ายเอกสาร | 500 | บาท |

รวมเป็นเงินทั้งสิ้น 2,000 บาท(สองพันบาทถ้วน)

# บทที่ 2
# ทฤษฎีและหลักการ

การที่จะสร้าง ภาพโมเสก (Image Mosaic) นั้นจำเป็นจะต้องมีความรู้ทั้งในด้านทฤษฎีและการพัฒนาแอพพลิเคชันด้วยโปรแกรมแมทแลบ (Matlab) เป็นอย่างดี เหตุผลที่ใช้ โปรแกรมแมทแลบ ก็เนื่องมาจากแมทแลบ สามารถสร้างโปรแกรมที่มีการใช้งานกับ อิมเมจโปรเซสซิ่ง (Image Processing) ได้อย่างมีประสิทธิภาพ และง่ายต่อการออกแบบ

## 2.1 ทฤษฎีที่ใช้

ในส่วนของทฤษฎีที่ใช้จะแบ่งออกเป็น 2 ส่วนคือ ส่วนของทฤษฎีสี และการเปรียบเทียบความเหมือนของพิกเซล (pixel) โดยพิจารณาจากค่าสี

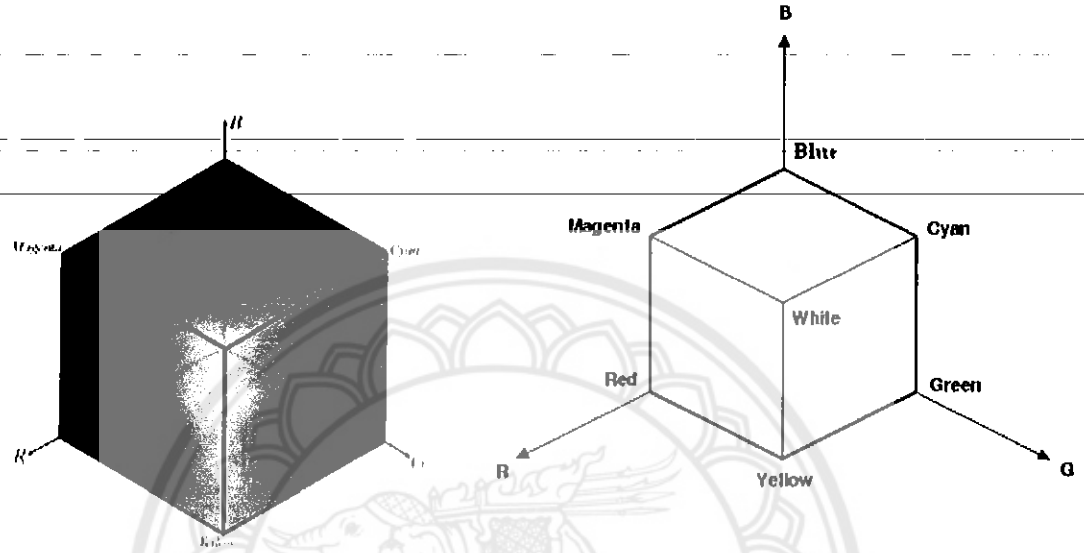
### 2.1.1ระบบสี[2]

มาตรฐานของสี

ในปัจจุบันนี้มีการใช้มาตรฐานสีอยู่หลายระบบในแต่ละระบบก็จะมีการกำหนดแตกต่างกันไป แต่โดยทั่วไปก็จะมีการแบ่งที่คล้ายกันอยู่คือการแทนสีด้วยจุด โดยจะแทนจุดในสเปซ 3 มิติ โดยมีแกนอ้างอิงแต่ละแกน อย่างเช่น แกนของสี( สีแดง, สีเขียว, สีน้ำเงิน) แกนของความสว่าง (lightness) แกนของความบริสุทธ์ของสี (saturation) ซึ่งแกนแต่ละแกนนี้จะเป็นอิสระต่อกัน เราจะยกตัวอย่างของระบบที่นิยมใช้ ระบบ RGB และ HSV [อ้างอิงที่ 3 จากหน้าอ้างอิง] ทั้ง2ระบบนี้เป็นระบบของมาตรฐานของสีที่มีความคล้ายคลึงกัน

- ระบบสี RGB

ระบบนี้จะเป็นการรวมกันของแสงสี มีแสงสีแดง แสงสีเขียว และ แสงสีน้ำเงิน จะแบ่งแต่ละสีออกจะเป็นแกน3แกน และ มีการไล่ความเข้มสี จากค่า 0→1 โดยค่า 0 ก็คือความเข้มสีมาก,ค่า 1 คือความเข้มสีน้อย หรือแสงสีขาว ดังรูป

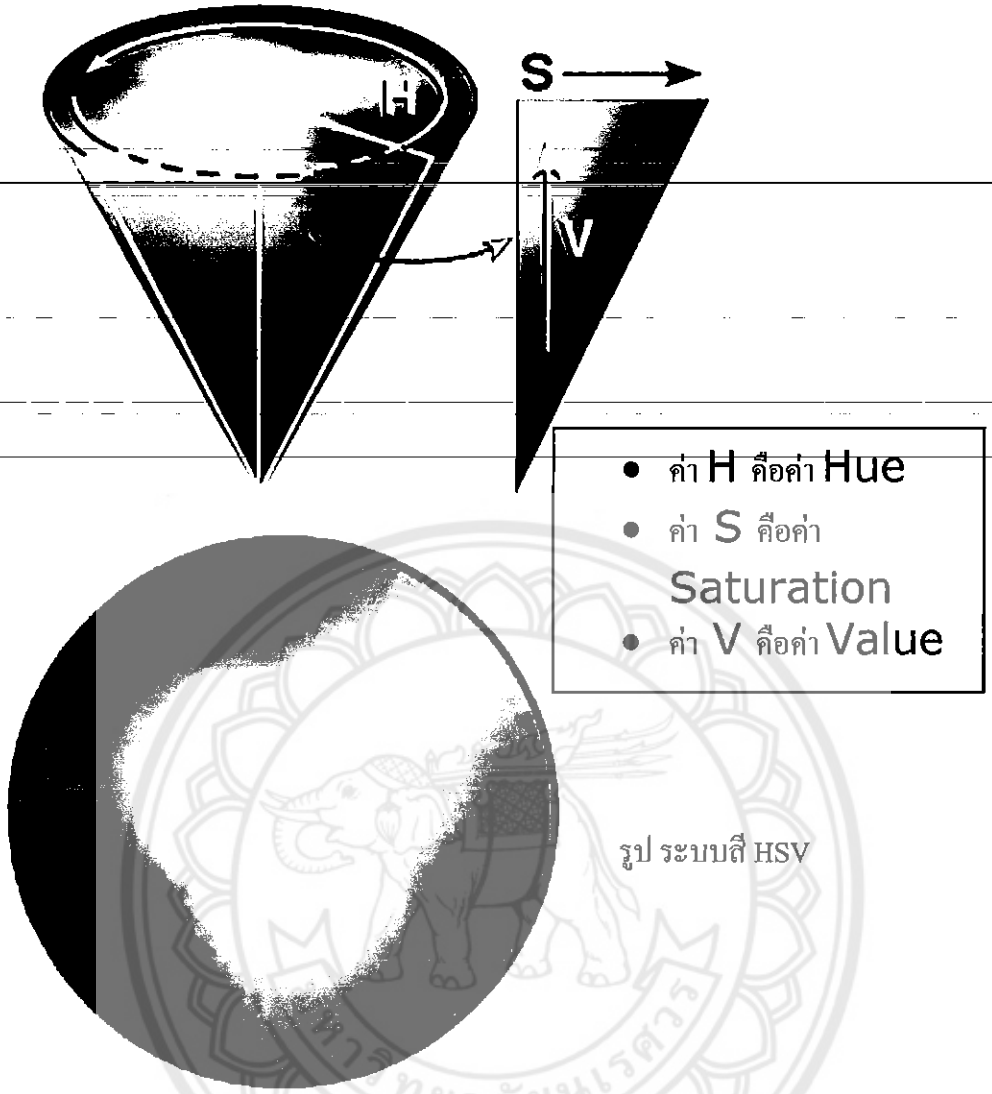



รูปที่ 2.1 แสดงระบบสีของ RGB

- **ระบบสี HSV [2]**

ระบบสี HSV (Hue Saturation Value) จะเป็นการพิจารณาสีโดยใช้ Hue Saturation และ Value ซึ่ง

Hue คือ ค่าสี (แดง เขียว และ น้ำเงิน) โดยจะมีค่าตั้งแต่ 0 ถึง 255 โดยค่า 0 จะเป็นสีแดง แล้วเมื่อ Hue มีค่าเพิ่มขึ้นเรื่อยๆ สีก็จะเปลี่ยนไปตามสเปกตรัมของสี แล้วเมื่อถึง 256 แสงจะกลับแสงสีแดงอีก แล้วนำมาทำเป็นกราฟ 3 มิติในรูปกรวย และ แบ่งให้อยู่ในรูปองศา จะได้สีแดงที่ 0 องศา สีเขียวที่ 120 องศา สีน้ำเงิน ที่ 240 องศา ดังรูป




รูป ระบบสี HSV

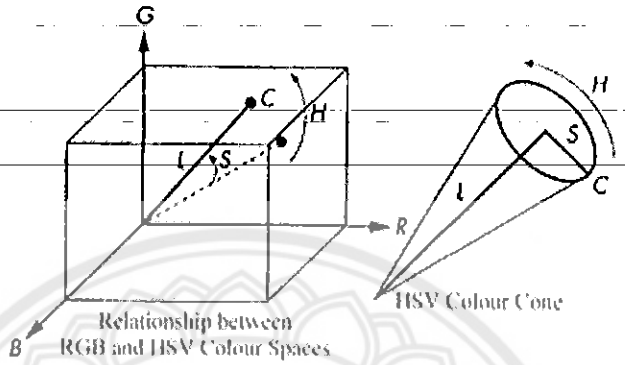
**รูปที่ 2.2** แสดงกราฟ 3 มิติในรูปกรวย

จากลักษณะโมเดลของระบบ Hue พบว่าจะมีค่าอย่างน้อยหนึ่งค่าที่จะเท่ากับ 0 แต่ถ้ามีสองค่าเท่ากับ 0 แล้ว Hue จะเป็นมุมของสี(ค่าสี)มีค่าเป็นไปตามสี่ที่สามและถ้าทั้งสามสีมีค่าเท่ากับ 0 แล้วจะทำให้ไม่มีค่าของ Hue หรือสีที่ได้จะมีค่าเท่ากับสีขาวนั่นเอง ตัวอย่างเช่น จอภาพขาว- ดำ ถ้าเกิดมีสีใดสีหนึ่งมีค่าเท่ากับ 0 จะทำให้ค่าสีที่ได้เป็นไปตามสีที่เหลือ การให้น้ำหนักในการพิจารณาเมื่อสีแดงมีค่าเท่ากับ 0

$$Hue = \frac{(240 \times blue_h) + (120 \times green_h)}{blue_h + green_h}$$

Hue สามารถคำนวณได้จากระบบสี RGB ได้ดังนี้

$$red_h = red - \min(red, green, blue)$$
$$green_h = green - \min(red, green, blue)$$
$$blue_h = blue - \min(red, green, blue)$$




รูปที่ 2.3 แสดงการเกี่ยวข้องของ RGB กับ HSV

Saturation คือความบริสุทธิ์ของสีซึ่งถ้า Saturation มีค่าเท่ากับ 0 แล้วสีที่ได้จะไม่มี Hue ซึ่งจะเป็นสีขาวล้วนแต่ถ้า Saturation มีค่าเท่ากับ 255 แสดงว่าจะไม่มีแสงสีขาวผสมอยู่เลย Saturation สามารถคำนวณได้ดังนี้

$$Saturation = \frac{\max(red, green, blue) - \min(red, green, blue)}{\max(red, green, blue)}$$

Value คือความสว่างของสี ซึ่งสามารถวัดได้โดยค่าความเข้มของความสว่างของแต่ละสีที่ประกอบกันสามารถคำนวณได้จาก

$$value = \max(red, green, blue)$$

**2.1.2 Similarity Comparison[3]** เป็นการเปรียบเทียบความเหมือนของภาพในโครงงานนี้ โดยพิจารณาจากค่าสัมบูรณ์ของผลต่างของค่าพิกเซลในแต่ละแถวและหลักที่เรากำหนด (Absolute Error) โดยสามารถอธิบายได้ดังนี้

กำหนดให้รูป 2.4 และ 2.5 แทนรูปภาพที่มีค่าของพิกเซล แสดงดังรูป

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 50 | 87 | 54 | 54 | 64 | 54 | 5 | 77 | 54 | 15 |
| 64 | 93 | 74 | 77 | 87 | 8 | 48 | 45 | 87 | 14 |
| 82 | 50 | 35 | 98 | 96 | 65 | 12 | 54 | 23 | 41 |
| 35 | 78 | 44 | 44 | 55 | 22 | 32 | 21 | 56 | 23 |
| 12 | 25 | 33 | 82 | 56 | 47 | 156 | 3 | 54 | 26 |
| 78 | 32 | 69 | 28 | 54 | 57 | 44 | 26 | 98 | 29 |
| 63 | 74 | 96 | 75 | 23 | 68 | 89 | 65 | 15 | 31 |
| 47 | 56 | 111 | 67 | 12 | 98 | 78 | 49 | 12 | 30 |
| 17 | 42 | 78 | 75 | 9 | 20 | 82 | 89 | 16 | 36 |
| 112 | 24 | 20 | 83 | 7 | 74 | 16 | 21 | 13 | 44 |

รูปที่ 2.4 แสดงค่าของพิกเซลรูปที่ 1

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 87 | 54 | 64 | 15 | 77 | 54 | 54 | 5 | 50 | 45 |
| 93 | 77 | 87 | 14 | 45 | 74 | 87 | 48 | 64 | 14 |
| 50 | 98 | 96 | 41 | 54 | 35 | 23 | 12 | 82 | 48 |
| 78 | 44 | 55 | 23 | 21 | 44 | 56 | 32 | 35 | 23 |
| 25 | 82 | 56 | 26 | 3 | 33 | 54 | 156 | 12 | 62 |
| 32 | 28 | 54 | 29 | 26 | 69 | 98 | 44 | 78 | 29 |
| 74 | 75 | 23 | 31 | 65 | 96 | 15 | 89 | 63 | 82 |
| 56 | 67 | 12 | 30 | 49 | 111 | 12 | 78 | 47 | 30 |
| 42 | 75 | 9 | 36 | 89 | 78 | 16 | 82 | 17 | 36 |
| 24 | 83 | 7 | 44 | 21 | 20 | 13 | 16 | 112 | 33 |

รูปที่ 2.5 แสดงค่าของพิกเซลรูปที่ 2

เมื่อเราต้องการเปรียบเทียบค่าความเหมือนของภาพทั้งสอง จะทำได้โดยนำค่าของพิกเซล 6 ค่าในหลักสุดท้ายของภาพที่ 1 (รูป 2.4) มาเปรียบเทียบกับหลักแรกของภาพที่ 2 (รูป 2.5) เพื่อหาผลต่างของค่าพิกเซลในภาพทั้งสอง โดยผลต่างที่ได้จะอยู่ในรูปของค่าสัมบูรณ์เพื่อให้ค่าที่ได้เป็นค่าบวก จากนั้นนำค่าผลต่างในแต่ละตำแหน่งมาบวกกันเป็นผลรวมแล้วเก็บค่าไว้ที่ตำแหน่งของภาพ

$|15 - 87| = 72$

$|14 - 93| = 79$

$|41 - 50| = 9$

$|23 - 78| = 55$

$|26 - 25| = 1$

$|29 - 32| = 3$

ดังนั้นผลรวมของค่าผลต่างพิกเซลจะมีค่าเท่ากับ 72+79+9+55+1+3 = 219 โดยที่ค่า 219 จะถูกเก็บไว้ที่ตำแหน่ง (0,0) ของภาพที่ 2 จากนั้นก็ค่อย ๆเลื่อนการหาผลต่างลงมาทีละพิกเซล จนกระทั่งครบทุกตัวในหลักที่ 1 จึงเริ่มหาต่อในหลักที่ 2 ไปเรื่อย ๆ จนครบทั้งหมด

โดยการเก็บค่าจะเก็บค่าในรูปของเมตริกซ์ แต่เนื่องจากหนึ่งภาพจะหาทั้ง 3 พารามิเตอร์คือ R G และ B จึงทำให้ได้เมตริกซ์ 3 เมตริกซ์

ในการหาตำแหน่งที่จะนำภาพมารวมกันนั้นเราก็ทำได้โดย นำตำแหน่งของภาพที่ 1 ที่นำมาหาค่าแอบโซลูท เออเรอ มาวางไว้ในตำแหน่งที่มีค่าแอบโซลูท เออเรอน้อยที่สุดของภาพที่ 2 ก็จะได้ภาพที่รวมกัน

## 2.2 การพัฒนาแอพพลิเคชัน

แต่ก่อนนั้นการสร้างแอพพลิเคชัน (Application) ด้วยการเขียนโปรแกรม โดยในอดีตนั้นนักเขียนโปรแกรม หรือโปรแกรมเมอร์ต่างต้องผ่านความยุ่งยากซับซ้อนของการเขียนโปรแกรมมาด้วยกันทุกคน ทำให้มองกันว่าการเขียนโปรแกรมเป็นเรื่องที่ซับซ้อน

การใช้งาน Windows นั้นเป็นเรื่องที่แสนจะง่าย แต่การสร้างแอพพลิเคชันให้ทำงานกับ Windows นั้นกลับทำได้ยาก แต่เดิมนั้นผู้ที่จะสร้างแอพพลิเคชันในการทำงานบน Windows จำเป็นต้องรู้การทำงานภายใน Windows เป็นอย่างดีเพื่อที่จะเขียนโปรแกรมได้อย่างถูกต้อง

### 2.2.1 การพัฒนาแอพพลิเคชันด้วย Matlab

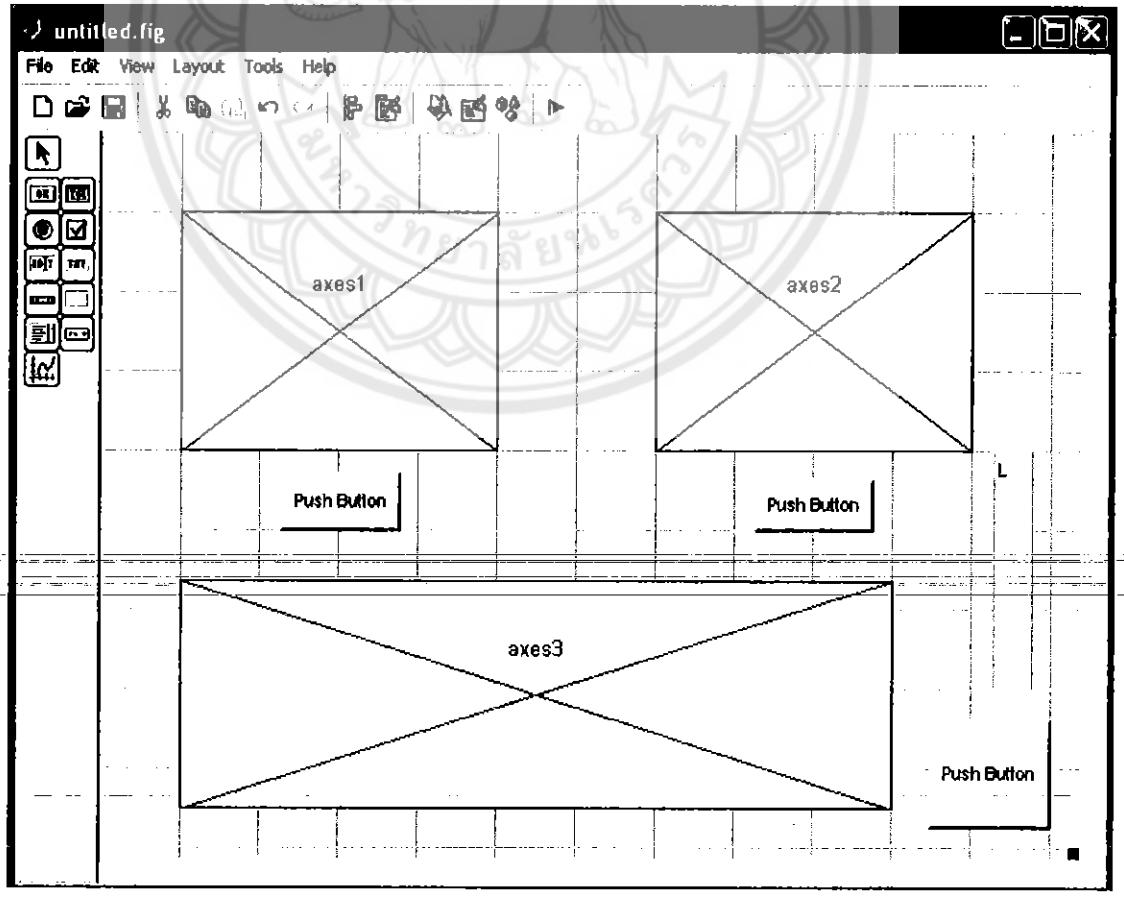
ภาษาในการเขียนโปรแกรมถูกสร้างขึ้นมาด้วยวัตถุประสงค์ที่แตกต่างกันแต่ละภาษามีจุดเด่นที่ต่างกันออกไป จึงเหมาะกับงานคนละประเภท เช่น ภาษา C เป็นภาษาที่นักเขียนโปรแกรมทั่วโลกนิยมกันมากเพราะมีความยืดหยุ่นสูง ทำงานได้เร็ว แต่โปรแกรมเมอร์ที่เขียนโปรแกรมเกี่ยวกับอิมเมจโปรเซสซิ่ง จะใช้โปรแกรมแมทแลบ เสียส่วนใหญ่

เพื่อที่เป็นทางเลือกในการสร้างแอพพลิเคชันที่หลากหลายบน Windows ทำได้อย่างรวดเร็ว ไมโครซอฟท์ได้แนะนำ Matlab [เวอร์ชัน 6.5.1 ของบริษัทแมทเวิร์ค][1] ซึ่งเป็นเครื่องมือที่สำหรับสร้างแอพพลิเคชัน Windows ที่มีเครื่องมือให้โปรแกรมเมอร์ได้เลือกใช้งาน

### 2.2.2 จุดเด่นของ Matlab

2.2.2.1 สามารถสร้างแอพพลิเคชันได้อย่างรวดเร็ว

โปรแกรมแมทแลบได้รับการวางตัวให้เป็นเครื่องมือที่ช่วยให้สร้างแอพพลิเคชันได้อย่างรวดเร็วและง่ายเพื่อที่จะได้ลดเวลาในการสร้างแอพพลิเคชันลง




**รูปที่ 2.6** แสดงการออกแบบ GUI ของโปรแกรมแมทแลบ

ทั้งนี้เพราะมีการขจัดงานที่โปรแกรมเมอร์ต้องทำซ้ำๆ ออกไปขจัดสิ่งที่ไม่จำเป็นต้องรู้เกี่ยวกับการควบคุมฮาร์ดแวร์ (Hardware) การจัดการภายในของ windows ออกไปเหลือเฉพาะที่ต้องโฟกัสเกี่ยวกับปัญหาของงานจริงๆแล้วเขียนโปรแกรมจัดการปัญหานั้น และส่วนอื่นก็ให้ Matlab จัดการ

2.2.2.2 ง่ายต่อการแก้ไขและเพิ่มเติม

ถ้ามีโอกาสที่จะเขียนโปรแกรมด้วยโปรแกรมแมทแลบแล้วจะเห็นว่า โปรแกรมแมทแลบนั้นสามารถแก้ไขและเพิ่มเติมส่วน ของโปรแกรมและแอพพลิเคชันได้อย่างรวดเร็ว เพราะแต่ละส่วนของโปรแกรมมีการเชื่อมโยงกัน

**รูปที่ 2.7** แสดงการเขียนโปรแกรมในแมทแลบ

จากรูปที่ 2.7 เป็นการแสดงการเขียนโปรแกรมโดยใช้แมทแลบ จะเห็นว่าเมื่อเราต้องการแก้ไขหรือเพิ่มเติมแอพพลิเคชั่น ของโปรแกรมในส่วนใด ก็เพียงแค่คลิกจากแอพพลิเคชัน ก็จะสามารถเข้ามาแก้ไขตรงส่วนของโปรแกรมได้อย่างสะดวกและรวดเร็ว ถึงแม้โครงสร้างของภาษา

แมทแลบจะมีความซับซ้อน แต่ถ้าเขียนโปรแกรมที่ต้องใช้งานเกี่ยวกับการประมวลผลภาพและมีแอพพลิเคชันด้วยก็จะเห็นว่าสามารถทำได้เป็นอย่างดี

2.2.2.3 รวมเครื่องมืออำนวยความสะดวกในการเขียนโปรแกรม

นอกจากง่ายต่อการเรียนรู้แล้ว แมทแลบยังมีเครื่องมือที่ช่วยในการเขียนโปรแกรมเป็นเรื่องที่ไม่ยุ่งยากอีกต่อไปเพราะจะมีเครื่องมือที่ช่วยให้ไม่ต้องจดจำไวยากรณ์ภาษาที่ยุ่งยาก ตรวจสอบอัตโนมัติว่าโปรแกรมที่เขียนนั้นถูกต้องหรือไม่ มีการแยกแยะส่วนของโปรแกรมอย่างเป็นระเบียบทำให้งานของโปรแกรมเมอร์ลดลงไปได้มาก

นอกจากจะมีเครื่องมือที่ช่วยในกรเขียนโปรแกรมแล้ว ยังมีเครื่องมือทดสอบแก้ไขโปรแกรม (Debugger) ที่เขียนขึ้นมาว่าทำงานได้อย่างถูกต้องหรือไม่ มีการรับการขอความช่วยเหลือ (Online Help) ไว้อ้างอิง และข้อความช่วยเหลือในจุดที่เราสงสัย

# บทที่ 3
# วิธีการดำเนินการ

เนื้อหาในบทนี้จะกล่าวถึงขั้นตอนในการดำเนินการต่าง ๆ โดยจะเป็นรายละเอียดในส่วนของการพัฒนาภาพโมเสก (Image Mosaic)

## 3.1ออกแบบโปรแกรม

ในส่วนของโปรแกรมสามารถอธิบายด้วยโฟลชาร์ด (Flow Chart) ได้ดังนี้

**รูปที่ 3.1** แสดงแผนผังการทำงานของโปรแกรม

จาก โฟลชาร์ค สามารถอธิบายได้ดังนี้

**- อ่านภาพ input 2ภาพ**

คือการใส่ภาพอินพุท 2 ภาพที่ใส่เข้าไปในโปรแกรม โดยไฟล์ภาพที่สามารถใส่เข้าคือ .jpg จากนั้นโปรแกรมจะเก็บค่าพิกเซลของภาพทั้ง 3 เลเยอร์ คือ R G B โดยจะเก็บไว้ในรูปแบบเมตตริกซ์

**- นำภาพทั้ง 2 มาเปรียบเทียบทีละพิกเซล หาค่า Absolute Error**

คือเมื่อใส่ภาพ อินพุทเข้าไปครบแล้ว โปรแกรมจะนำภาพทั้งสองมาหาค่า แอบโซลูท เออเรอ โดยจะหาทีละหลักดังนี้

เมื่อเราต้องการเปรียบเทียบค่าความเหมือนของภาพทั้งสอง จะทำได้โดยนำค่าของพิกเซลในหลักสุดท้ายของภาพที่ 1 มาเปรียบเทียบกับหลักแรกของภาพที่ 2 เพื่อหาผลต่างของค่าพิกเซลในภาพทั้งสอง โดยผลต่างที่ได้จะอยู่ในรูปของค่าสัมบูรณ์เพื่อให้ค่าที่ได้เป็นค่าบวก จากนั้นนำค่าผลต่างในแต่ละตำแหน่งมาบวกกันเป็นผลรวมแล้วเก็บค่าไว้ที่ตำแหน่งของภาพ ดังนี้

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 183 | 178 | 201 | 163 | 194 | 172 | 222 | 136 | 112 | 212 |
| 168 | 115 | 213 | 182 | 172 | 184 | 124 | 134 | 134 | 204 |
| 171 | 123 | 211 | 199 | 168 | 125 | 63 | 120 | 143 | 209 |
| 198 | 133 | 221 | 117 | 177 | 187 | 57 | 111 | 144 | 198 |
| 154 | 174 | 197 | 128 | 122 | 93 | 48 | 103 | 177 | 184 |
| 147 | 198 | 177 | 132 | 78 | 50 | 34 | 78 | 93 | 189 |
| 213 | 101 | 183 | 63 | 67 | 35 | 27 | 87 | 95 | 200 |
| 79 | 99 | 64 | 58 | 65 | 11 | 63 | 72 | 195 | 213 |
| 86 | 77 | 98 | 74 | 57 | 26 | 66 | 77 | 63 | 171 |
| 91 | 63 | 116 | 89 | 43 | 34 | 88 | 63 | 88 | 163 |

ค่าของพิกเซลรูปที่ 1ในรูปเมตริกซ์

$$\begin{bmatrix}
117 & 187 & 172 & 93 & 99 & 102 & 224 & 228 & 134 & 154 \\
241 & 163 & 150 & 97 & 107 & 143 & 236 & 246 & 167 & 156 \\
93 & 213 & 143 & 101 & 109 & 91 & 248 & 200 & 193 & 187 \\
163 & 194 & 172 & 222 & 136 & 112 & 211 & 212 & 184 & 123 \\
182 & 172 & 184 & 124 & 134 & 134 & 203 & 178 & 167 & 131 \\
199 & 168 & 125 & 63 & 120 & 143 & 208 & 192 & 179 & 201 \\
117 & 177 & 187 & 57 & 111 & 144 & 197 & 177 & 218 & 178 \\
128 & 122 & 93 & 48 & 103 & 177 & 185 & 187 & 94 & 192 \\
132 & 78 & 50 & 34 & 78 & 93 & 191 & 198 & 134 & 163 \\
63 & 67 & 35 & 27 & 87 & 95 & 221 & 163 & 164 & 191
\end{bmatrix}$$

ค่าของพิกเซลรูปที่ 2 ในรูปเมตริกซ์

จากสูตร $S = \sum |[A - B]$

เมื่อ S เป็นค่าผลรวมของค่าแอบโซลูท เออเรอ

เมื่อ A เป็นค่าพิกเซลของรูปที่ 1

เมื่อ B เป็นค่าพิกเซลของรูปที่ 2

จะได้

$$\begin{vmatrix}
212 - 117 \\
204 - 241 \\
209 - 93 \\
198 - 163 \\
184 - 182 \\
189 - 199
\end{vmatrix}
\begin{matrix}
= 105 \\
= 37 \\
= 116 \\
= 35 \\
= 2 \\
= 2
\end{matrix}$$

ดังนั้นผลรวมของค่าผลต่างพิกเซลจะมีค่าเท่ากับ 105+37+116+35+2+2 = 297

โดยที่ค่า 297 จะถูกเก็บไว้ที่ตำแหน่ง (0,0) ของภาพที่ 2 จากนั้นก็ค่อย ๆเลื่อนการหาผลต่างลงมาทีละพิกเซล

เมื่อเลื่อนลงมา 1 พิกเซลจะได้

| | |
|---|---|
| 212 – 241 | = 29 |
| 204 – 93 | = 111 |
| 209 – 163 | = 46 |
| 198 – 182 | = 16 |
| 184 – 199 | = 15 |
| 189 – 117 | = 72 |

ดังนั้นผลรวมของค่าผลต่างพิกเซลจะมีค่าเท่ากับ 29+111+46+16+15+72 = 289
โดยที่ค่า 275 จะถูกเก็บไว้ที่ตำแหน่ง (1,0) ของภาพที่ 2 จากนั้นก็ค่อย ๆเลื่อนการหาผลต่างลงมาทีละพิกเซล เมื่อทำเช่นนี้เรื่อย ๆ จนครบก็จะได้เมตริกซ์ที่มีค่า แอบโซลูท เออเรอ เป็นดังนี้

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 297 | 198 | 267 | 314 | 199 | 98 | 34 | 32 | 121 | 142 |
| 289 | 219 | 213 | 117 | 164 | 77 | 23 | 78 | 109 | 316 |
| 317 | 233 | 197 | 111 | 142 | 111 | 12 | 36 | 199 | 277 |
| 275 | 217 | 194 | 123 | 217 | 86 | 7 | 15 | 157 | 251 |
| 375 | 319 | 156 | 136 | 195 | 52 | 27 | 92 | 146 | 197 |
| 241 | 289 | 143 | 178 | 154 | 118 | 36 | 82 | 218 | 193 |
| 369 | 197 | 176 | 219 | 136 | 127 | 22 | 88 | 149 | 173 |
| 345 | 168 | 209 | 211 | 67 | 99 | 51 | 101 | 126 | 162 |
| 288 | 145 | 200 | 186 | 98 | 71 | 43 | 37 | 100 | 150 |
| 260 | 178 | 187 | 165 | 113 | 216 | 28 | 45 | 89 | 147 |

เมตริกซ์แสดงค่า แอบโซลูท เออเรอ

จะเห็นว่าค่า แอบโซลูท เออเรอ ที่มีค่าน้อยที่สุดจะอยู่ที่ตำแหน่ง (6,3) จากนั้นก็นำตำแหน่ง (0,9) ของภาพที่ 1 มาวางไว้ที่ตำแหน่ง (6,3) ของภาพที่ 2 ก็จะได้ภาพทั้งสองรวมกันในตำแหน่งที่มีค่าแอบโซลูทเออเรอน้อยที่สุด ดังนั้นจะได้เมตริกซ์ที่ได้จากการรวมดังนี้

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 117 | 187 | 172 | 93 | 99 | 102 | 224 | 228 | 134 | 154 |
| | | | 241 | 163 | 150 | 97 | 107 | 143 | 236 | 246 | 167 | 156 |
| | | | 93 | 213 | 143 | 101 | 109 | 91 | 248 | 200 | 193 | 187 |
| 183 | 178 | 201 | 163 | 194 | 172 | 222 | 136 | 112 | 212 | 212 | 184 | 123 |
| 168 | 115 | 213 | 182 | 172 | 184 | 124 | 134 | 134 | 204 | 178 | 167 | 131 |
| 171 | 123 | 211 | 199 | 168 | 125 | 63 | 120 | 143 | 209 | 192 | 179 | 201 |
| 198 | 133 | 221 | 117 | 177 | 187 | 57 | 111 | 144 | 198 | 177 | 218 | 178 |
| 154 | 174 | 197 | 128 | 122 | 93 | 48 | 103 | 177 | 184 | 187 | 94 | 192 |
| 147 | 198 | 177 | 132 | 78 | 50 | 34 | 78 | 93 | 189 | 198 | 134 | 163 |
| 213 | 101 | 183 | 63 | 67 | 35 | 27 | 87 | 95 | 200 | 163 | 164 | 191 |
| 79 | 99 | 64 | 58 | 65 | 11 | 63 | 72 | 195 | 213 | | | |
| 86 | 77 | 98 | 74 | 57 | 26 | 66 | 77 | 63 | 171 | | | |
| 91 | 63 | 116 | 89 | 43 | 34 | 88 | 63 | 88 | 163 | | | |

เมตริกซ์ที่ได้จากการรวมภาพที่ 1 และ ภาพที่ 2

โดยการเก็บค่าจะเก็บค่าในรูปของเมตริกซ์ แต่เนื่องจากหนึ่งภาพจะหาทั้ง 3 พารามิเตอร์คือ R G และ B จึงทำให้ได้เมตริกซ์ 3 เมตริกซ์

- เลือกจุดที่มีค่าผลรวมของ Absolute Error น้อยที่สุดเพื่อเป็นจุดที่ใช้รวมภาพเมื่อเราได้ค่าแอบโซลูท เออเรอ แล้วเราก็จะเลือกตำแหน่งของภาพที่เก็บค่าน้อยที่สุดเป็นจุดที่ใช้ในการรวมภาพ

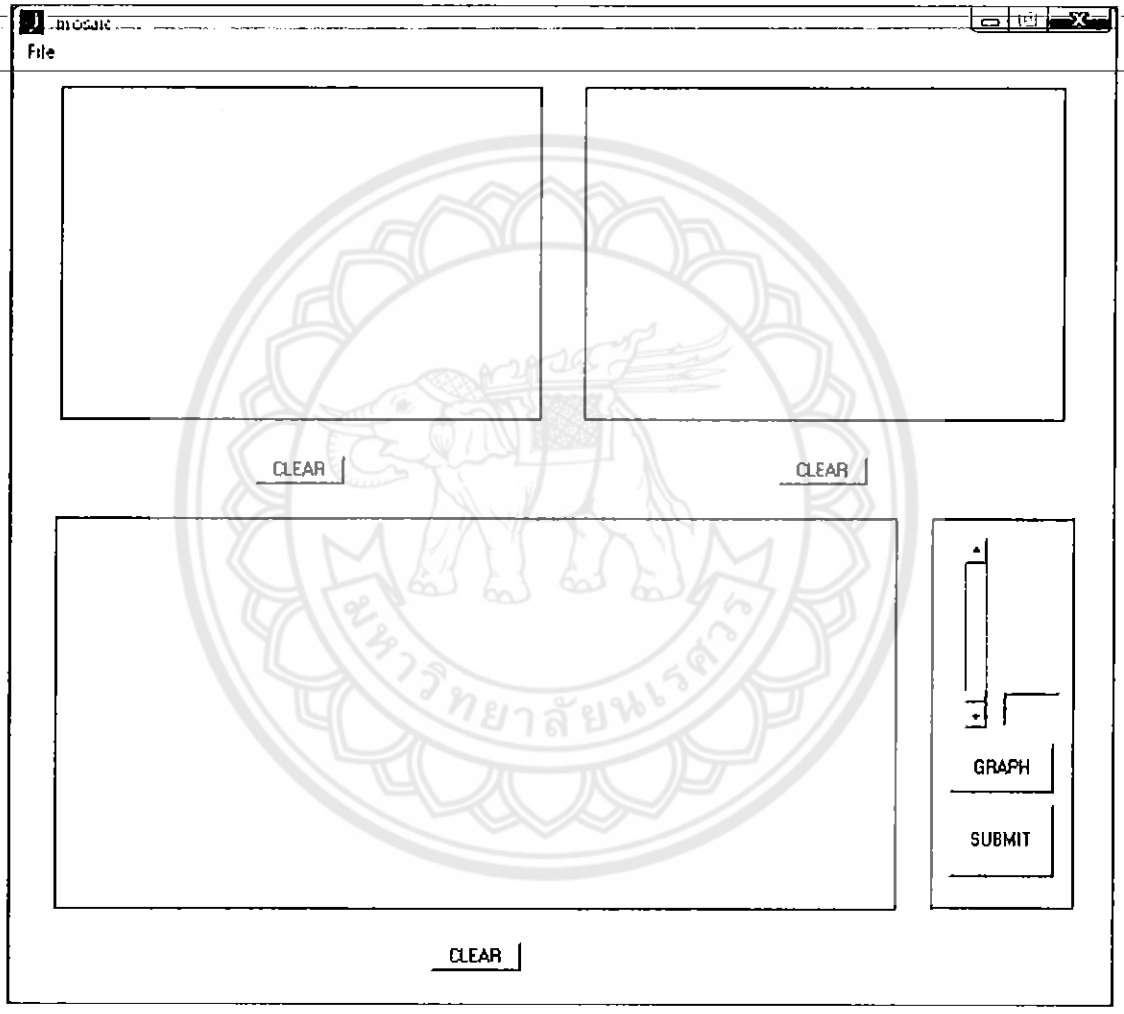
- รวมภาพทั้งสองเข้าด้วยกัน เมื่อได้ตำแหน่งที่เหมือนกันของภาพทั้ง 2 แล้ว โปรแกรมจะตัดส่วนของภาพที่ 2 ที่เหมือนกับภาพแรกออกไป แล้วนำส่วนของภาพที่เหลือมาต่อกับภาพแรก ซึ่งขนาดของภาพที่ได้จะเท่ากับ

(ภาพที่ 1 – ส่วนของภาพที่ 2 ที่เหมือนกันของภาพที่ 1) + ส่วนที่เหลือของภาพที่ 2

## 3.2 การออกแบบแอพพลิเคชัน

ในส่วนนี้จะเป็นการออกแบบโดยใช้ โปรแกรม แมทแลบ เป็นการออกแบบแอพพลิเคชันของโปรแกรมเพื่อให้ผู้ใช้สามารถใช้งานโปรแกรมได้ง่ายและสะดวกขึ้น

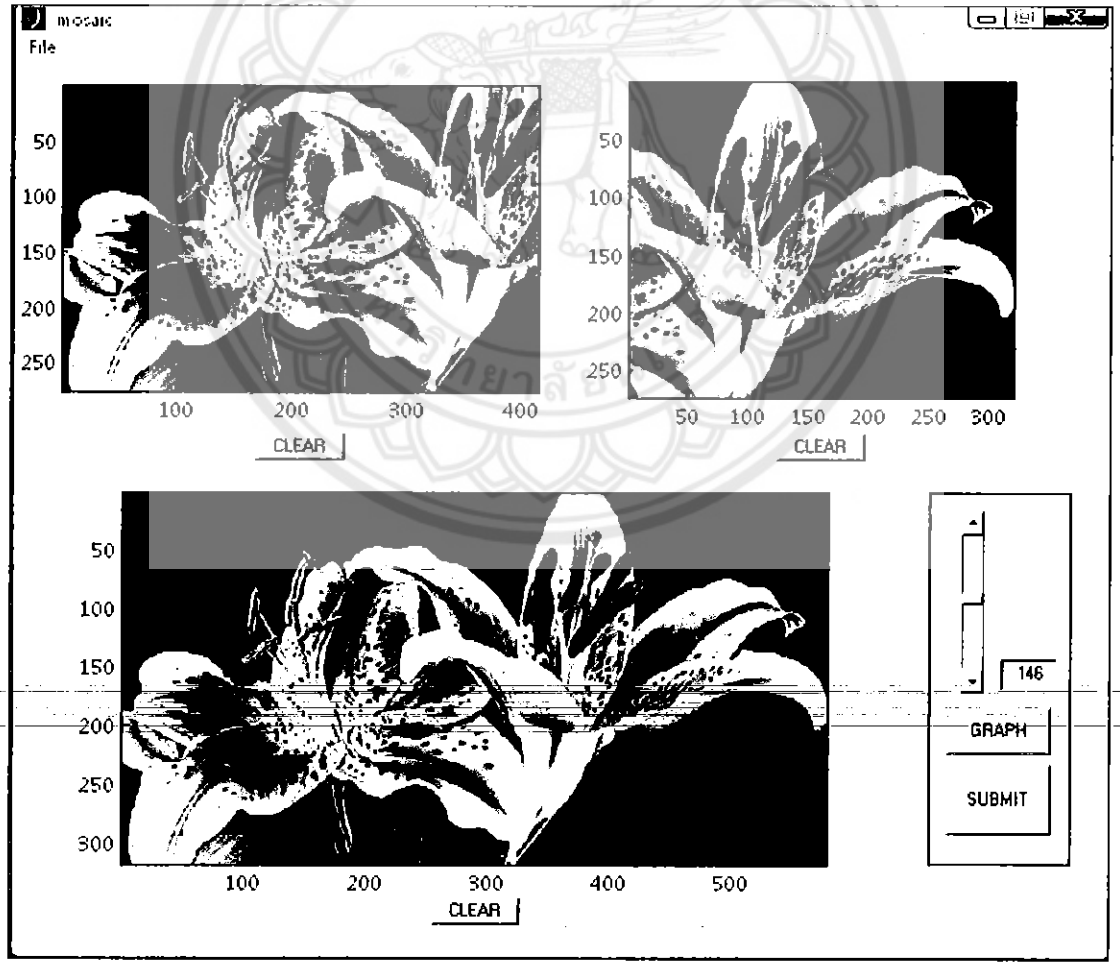
3.2.1ออกแบบแบบฟอร์มของโปรแกรม



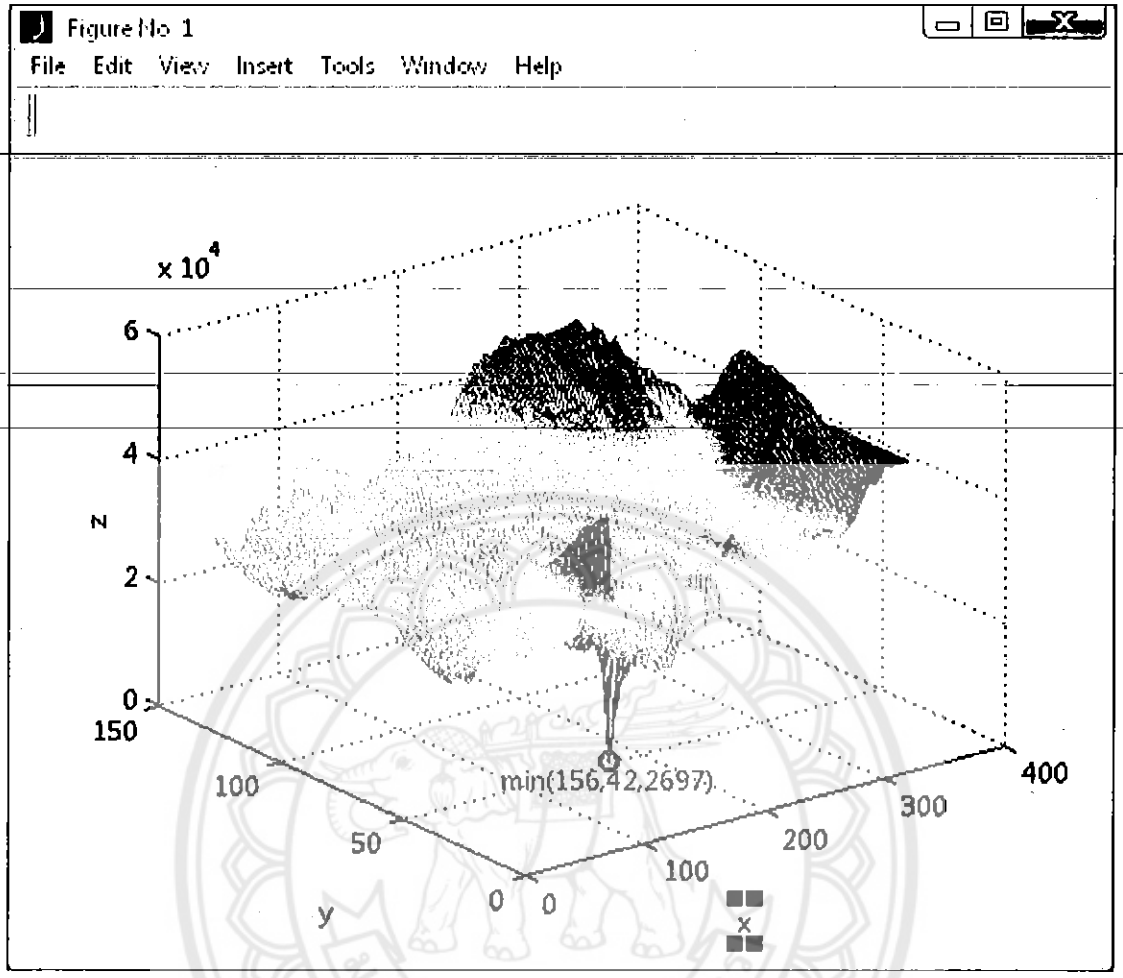

**รูปที่ 3.2** แสดงแบบฟอร์มของโปรแกรม

จากรูป 3.2 จะอธิบายได้ดังนี้

- การใส่รูปอินพุท ทำได้โดยไปที่ File → Open → จากนั้นเลือกว่าจะใส่รูปทางซ้ายหรือรูปทางขวา
- เมื่อใส่ภาพอินพุทเข้าไปครบแล้ว จะสามารถรวมภาพทั้งสองได้โดยการคลิกปุ่ม SUBMIT ก็จะได้ภาพออกมาทางด้านล่าง
- เมื่อคลิกปุ่ม GRAPH จะเป็นการแสดงกราฟกระบวนการทำงานของการรวมภาพ โดยกราฟที่ได้จะเป็นกราฟ 3 มิติ
- เมื่อต้องการใส่ภาพอื่นเราก็สามารถทำได้โดยการคลิกปุ่ม CLEAR ข้างล่างรูป จากนั้นก็ทำการใส่ภาพอินพุตเข้ามาเหมือนขั้นตอนแรก
- เมื่อต้องการ save ภาพที่ได้จากการทำการรวมภาพสามารถทำได้โดย ไปที่ File จากนั้นเลือก Save




**รูปที่ 3.3** แสดงแบบฟอร์มหน้าจอที่ใส่ภาพ




รูปที่ 3.4 แสดงแบบฟอร์มหน้าจอที่แสดงกราฟการทำงาน

โดยที่ แกน X เป็นจำนวนหลักของพิกเซล

แกน Y เป็นจำนวนหลักของพิกเซล

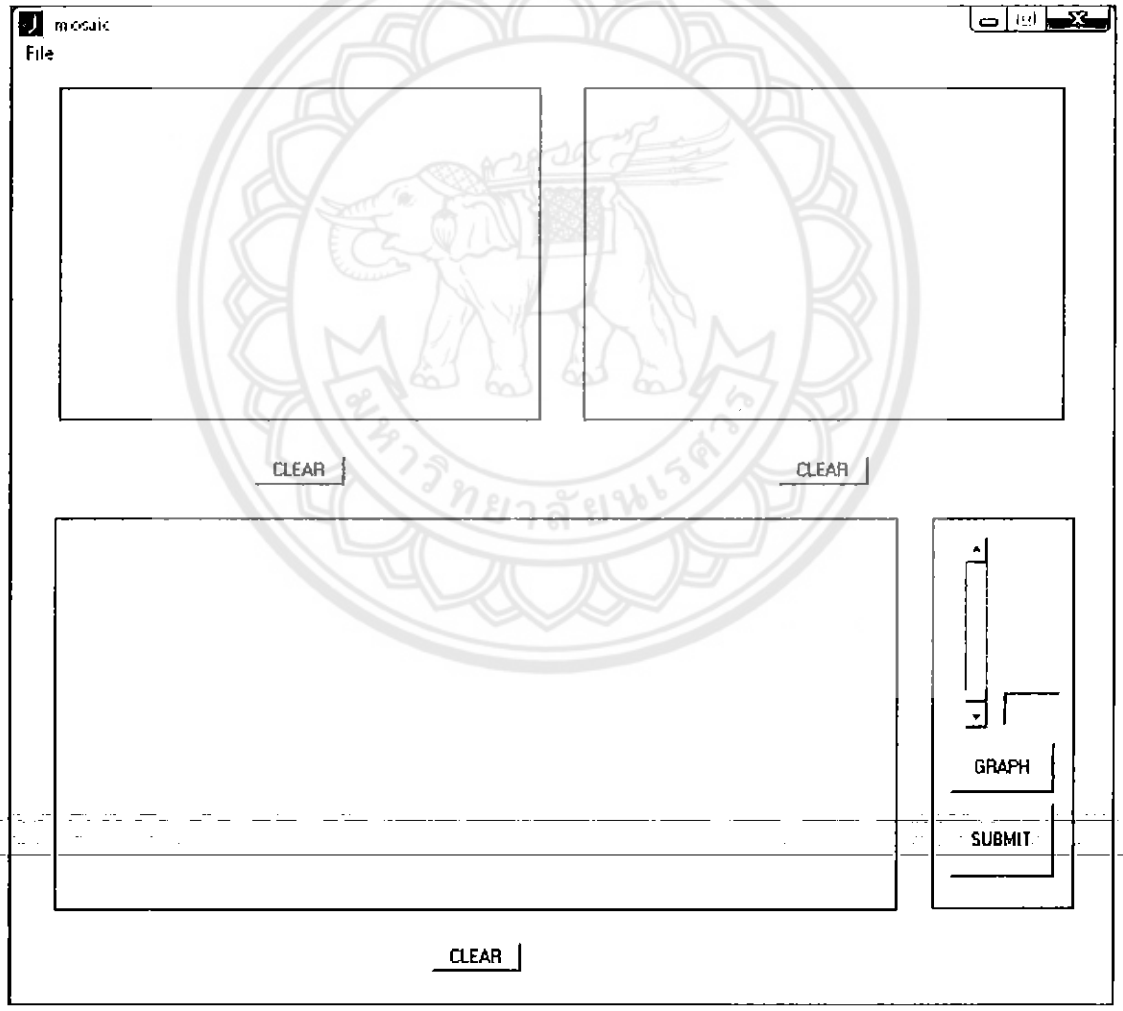
แกน Z เป็นค่าของ แอบโซลูทเออเรอ(Absolute Error)

# บทที่ 4

# ผลการทดลอง

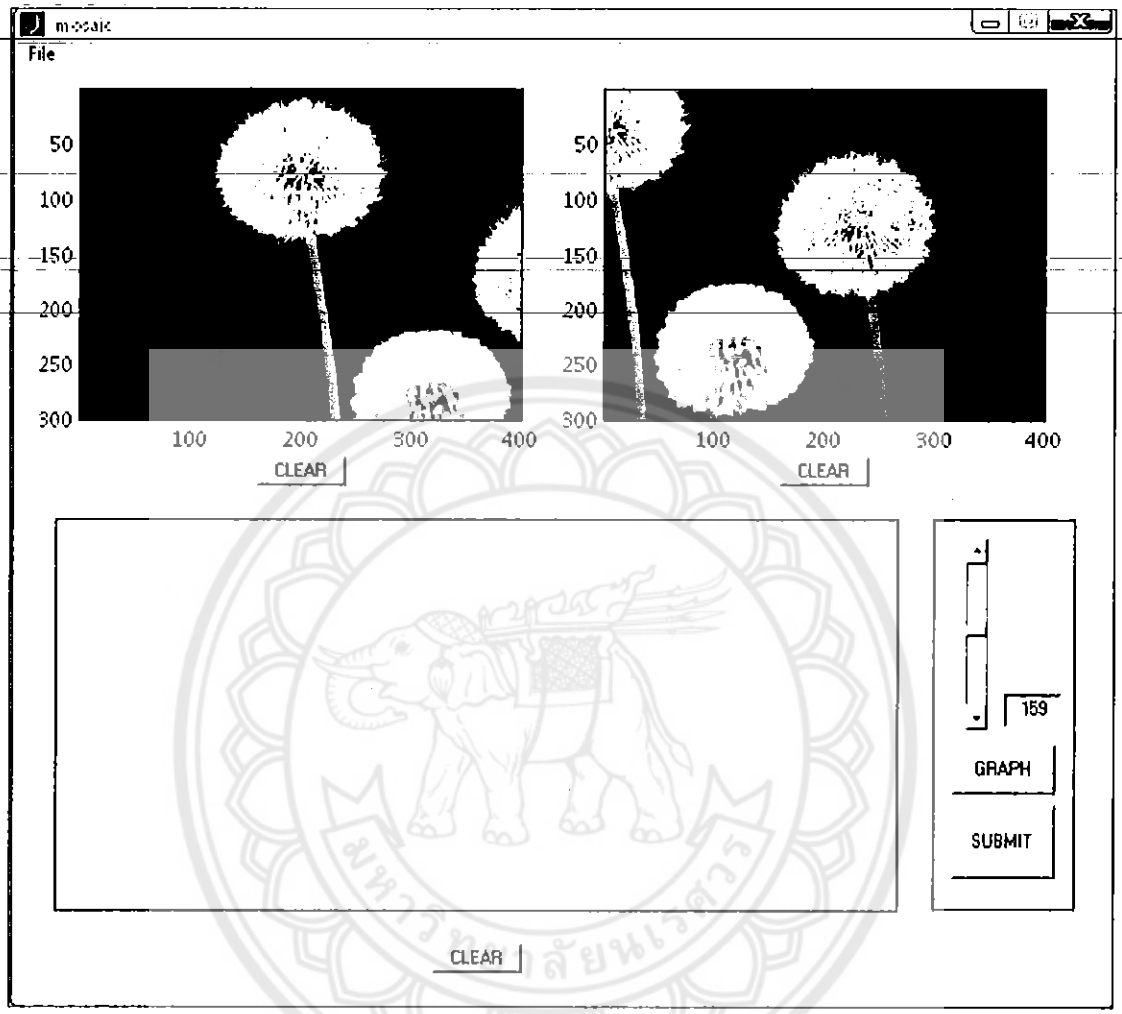
ในบทนี้เป็นการทำการทดลองเกี่ยวกับ ภาพโมเสก (Image Mosaic) ซึ่งเราได้ศึกษาและพัฒนาจากบทที่ผ่านมาแล้ว การทดลองในบทนี้จะเป็นการทดสอบการทำงานของโปรแกรมในการรวมภาพ 2 ภาพที่มีองค์ประกอบของภาพเหมือนกัน โดยผลการทดลองเป็นดังนี้

เมื่อทำการรันโปรแกรมขึ้นมาจะได้หน้าจอดังรูปที่ 4.1




**รูปที่ 4.1** แสดงหน้าจอเริ่มต้นของโปรแกรม

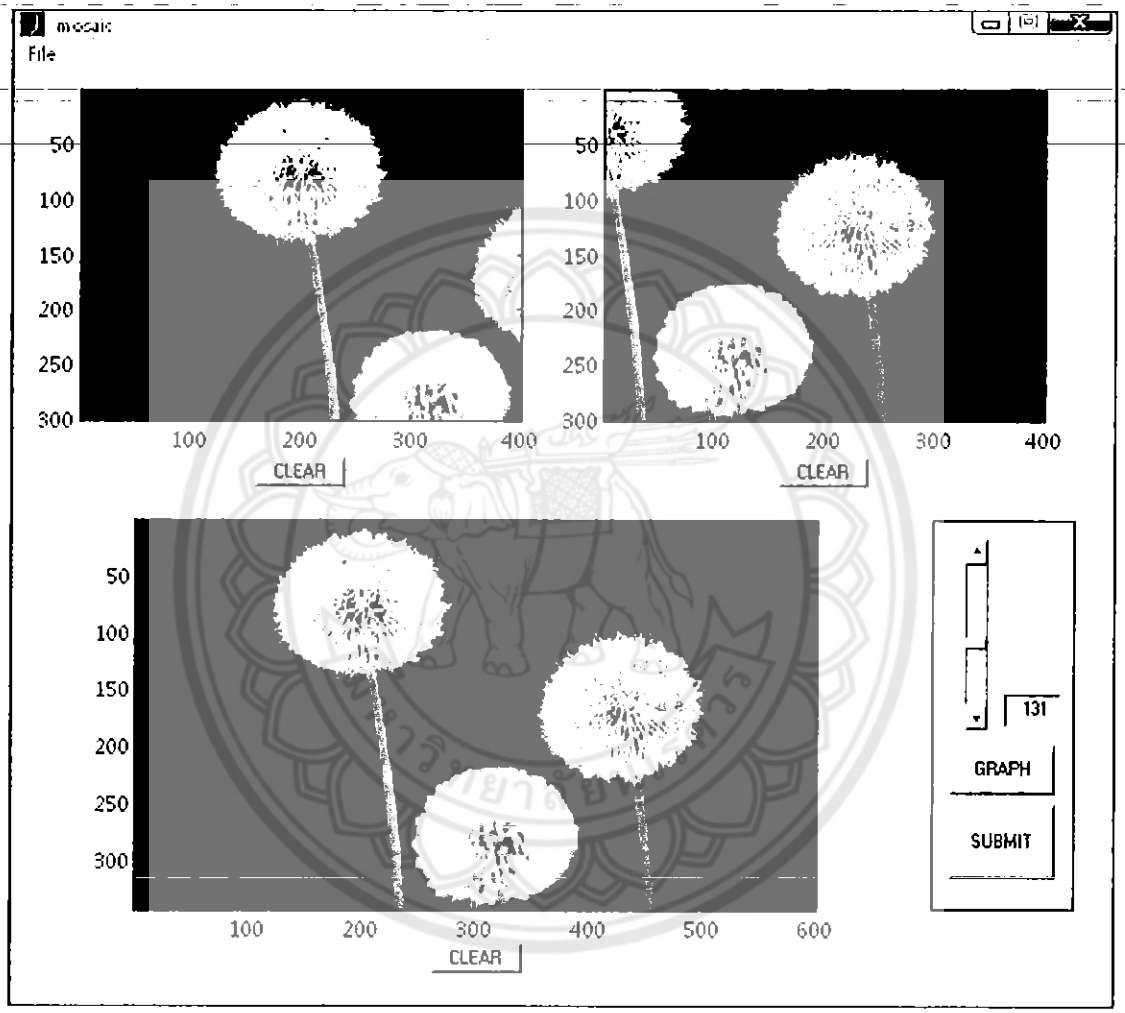
จากนั้นทำการใส่ภาพอินพุท เข้าไปในโปรแกรม โดยเลือกที่ File → Open จากนั้นเลือกรูปที่ต้องการใส่ และกดตกลง จะได้ดังรูปที่ 4.2



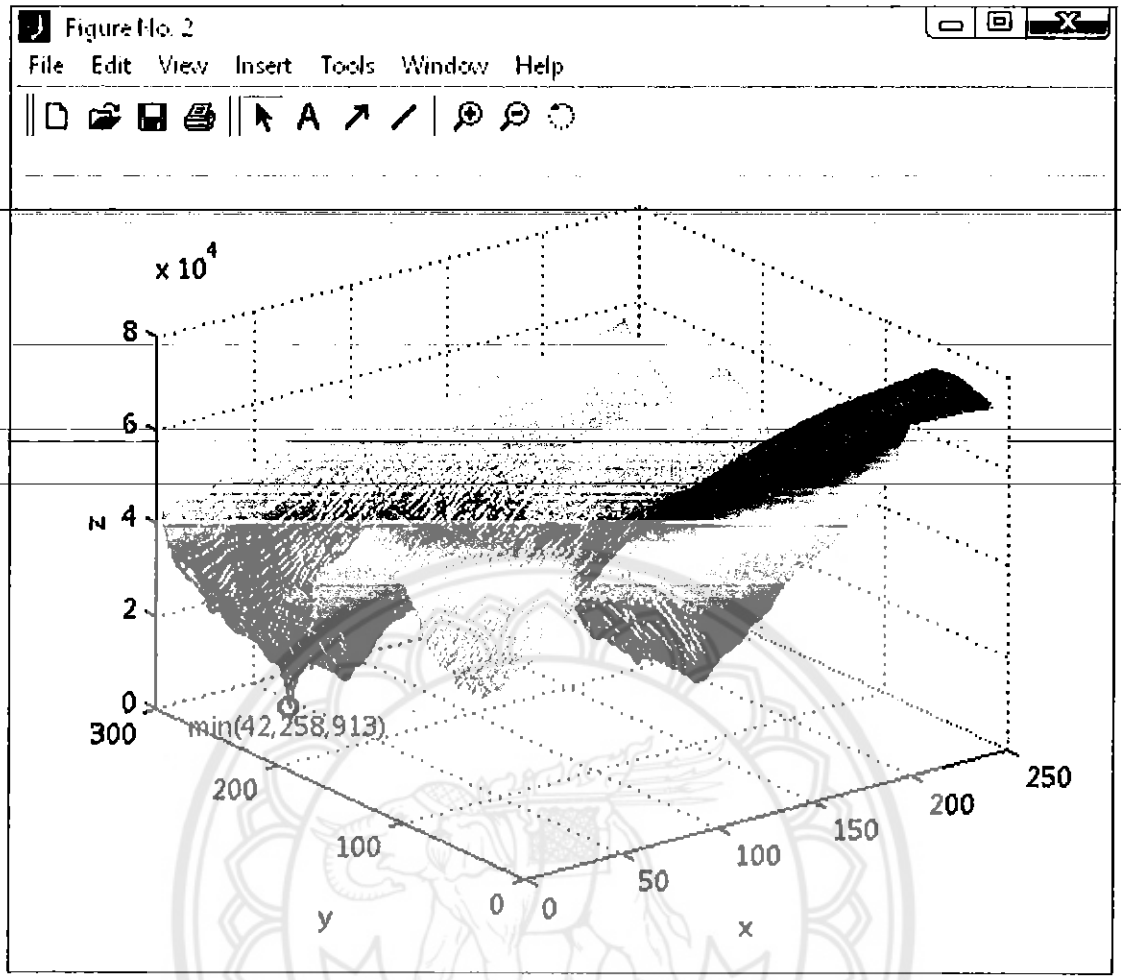

รูปที่ 4.2 แสดงหน้าจอโปรแกรมเมื่อทำการใส่ภาพอินพุต

หลังจากที่ใส่ภาพอินพุทครบทั้ง 2 ภาพแล้ว จะสามารถรวมภาพได้โดยคลิกปุ่ม SUBMIT ก็จะได้ภาพที่รวมกันแล้วดังรูปที่ 4.3

ถึงขั้นตอนนี้ถ้าเราต้องการที่จะรวมภาพอื่นก็สามารถทำได้โดยคลิกที่ปุ่ม CLEAR จากนั้นก็เลือกภาพอินพุทที่ต้องการเข้ามาเหมือนกับขั้นตอนแรก




**รูปที่ 4.3** แสดงภาพที่ได้หลังจากการรวม




**รูปที่ 4.4** แสดงรูปกราฟแสดงการทำงาน

จากรูปเป็นการแสดงหน้าจอของกราฟแสดงการทำงาน โดยจะแสดงจุดที่กราฟมีค่า Error น้อยที่สุด นั่นคือแสดงจุดที่ภาพรวมกันนั่นเอง

โดยที่ แกน X เป็นจำนวนหลักของพิกเซล

แกน Y เป็นจำนวนหลักของพิกเซล

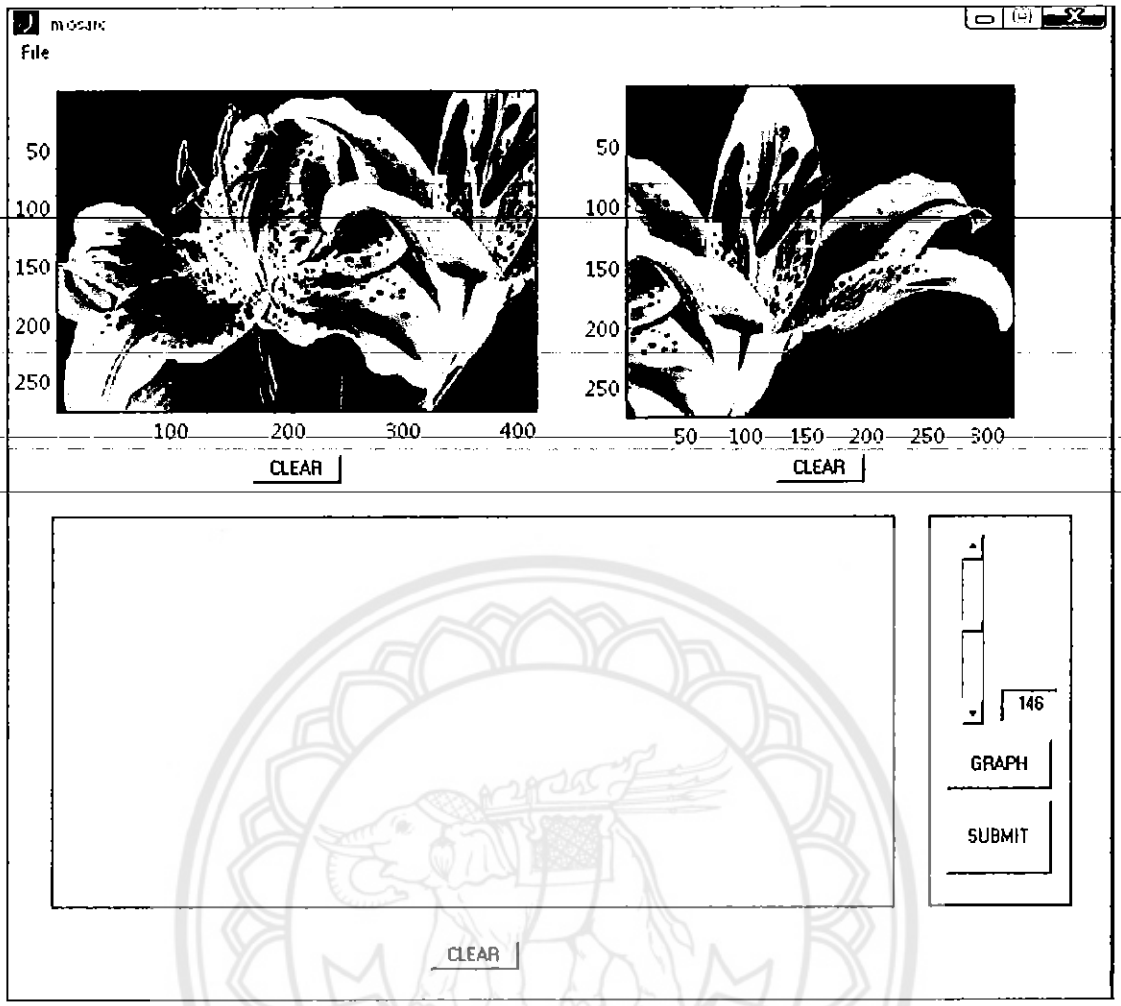
แกน Z เป็นค่าของ แอบโซลูทเออเรอ(Absolute Error)

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

i5000294

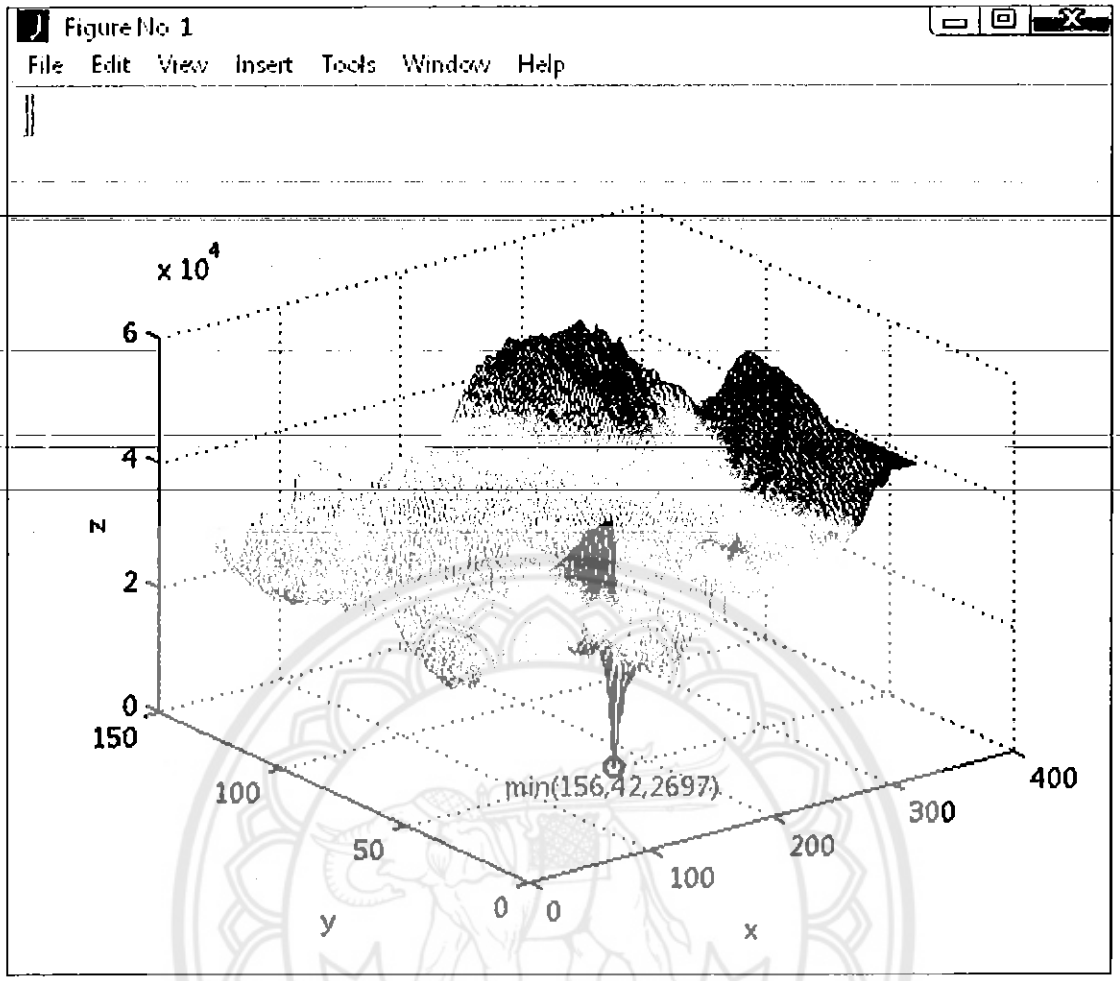


ชร.
WG39ว
2549




**รูปที่ 4.5** แสดงหน้าจอการใส่ภาพอินพุท

รูปที่ 4.6 แสดงภาพที่ได้จากการรวมภาพ




รูปที่ 4.7 แสดงภาพกราฟการทำงานของโปรแกรม

โดยที่ แกน X เป็นจำนวนหลักของพิกเซล

แกน Y เป็นจำนวนหลักของพิกเซล

แกน Z เป็นค่าของ แอบโซลูทเออเรอ(Absolute Error)

# บทที่ 5
# สรุปผลและข้อเสนอแนะ

## 5.1 สรุปผลการทดลอง

การทำงานของโปรแกรมที่ได้พัฒนาขึ้นเป็นการรวมภาพ 2 ภาพโดยที่ภาพทั้งสองมีองค์ประกอบเหมือนกัน โดยที่ภาพทั้งสองจะมีขนาดที่เท่ากันหรือไม่เท่ากันก็ได้

โดยที่ในการพัฒนาโปรแกรมแอพพลิเคชั่น จะใช้โปรแกรมแมทแลบในการพัฒนาโปรแกรมและหน้าจอแอพพลิเคชั่น เพื่อให้ง่ายต่อการใช้งาน

จากผลการทดลองในบทที่ 4 พบว่าเมื่อใส่ภาพอินพุทที่มีองค์ประกอบเหมือนกันเข้าไปในโปรแกรมก็จะสามารถรวมกันเป็นภาพเดียวกันได้ อย่างสมบูรณ์และไม่ขัดต่อการแสดงผลภาพพร้อมแสดงกราฟการทำงานของโปรแกรม

## 5.2 ปัญหาในการทดลอง

1. ถ้าส่วนของภาพที่ต้องการรวมมีความเบลอ จะทำให้ภาพที่มีความผิดพลาด
2. ภาพเดียวกันแต่มีมุม จะไม่สามารถทำการรวมกันได้
3. ถ้าภาพที่มีขนาดใหญ่มากมาทำการรวมภาพอาจทำให้ใช้เวลามากในการรวม

## 5.3 ข้อเสนอแนะ

1. โปรแกรมในโครงงานที่ทางผู้พัฒนาได้พัฒนาขึ้นใช้ Matlab เนื่องจากง่ายต่อการออกแบบและพัฒนาแอพพลิชัน หากมีการนำไปศึกษาพัฒนาต่อไป อาจเลือกใช้โปรแกรมภาษาอื่นในการพัฒนาเพื่อที่จะให้สามารถทำงานได้ซับซ้อนมากขึ้น

2. หากมีการนำไปพัฒนาต่อและลดข้อจำกัดในการรวมภาพ เช่น มุม ให้มีข้อจำกัดน้อยลง ก็จะสามารถใช้งานได้หลากหลายขึ้น

# บรรณานุกรม

[1] สัจจะ จรัสรุ่งรวีวร. **คู่มือการเขียนโปรแกรม Matlab 6.5.1 ฉบับโปรแกรมเมอร์,** สำนักพิมพ์อินโฟเพรส : 2546

[2] Yinning Deng, B.S.Manjunath, Charles Kenny, Micheal Moore and Hyundoo Shin "**An Efficient Color Representation for Image Retrieval**" IEEE Transection on Image Processing,Vol.10,No.1,January 2001 pp.140-147

[3] **A comparison of measures for visualizing image similarity** Kerry Rodden, Wojciech Basalaj University of Cambridge Computer Laboratory Pembroke Street, Cambridge CB2 3QG ,UK

# ประวัติผู้เขียนโครงงาน


ชื่อ นายพิชญา ประทุมศิริ

ภูมิลำเนา บ้านเลขที่ 80/309 ถ. ประชาอุทิศ อ. เมือง จ. พิษณุโลก 65000

ประวัติการศึกษา

- ระดับมัธยมต้น โรงเรียนจุฬาภรณราชวิทยาลัย พิษณุโลก
- ระดับมัธยมปลาย โรงเรียนจุฬาภรณราชวิทยาลัย พิษณุโลก

ขณะนี้กำลังศึกษาอยู่ที่ คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมคอมพิวเตอร์ ชั้นปีที่ 4 มหาวิทยาลัยนเรศวร

E-mail: myth_k@hotmail.com

ชื่อ นายอมรเทพ มหิวรรณ

ภูมิลำเนา บ้านเลขที่ 82 หมู่ 3 ต.บุ่งเลิศ อ.เมยวดี จ.ร้อยเอ็ด 45250

ประวัติการศึกษา

- ระดับมัธยมต้น โรงเรียนเมยวดีพิทยาคม ร้อยเอ็ด
- ระดับมัธยมปลาย โรงเรียนเมยวดีพิทยาคม ร้อยเอ็ด

ขณะนี้กำลังศึกษาอยู่ที่ คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมคอมพิวเตอร์ ชั้นปีที่ 4 มหาวิทยาลัยนเรศวร

E-mail: gaofongjr@hotmail.com