# โปรแกรมสืบค้นภาพบนดอทเน็ต (.NET) แพลทฟอร์ม

## IMAGE RETRIEVAL SOFTWARE APPLICATION

## ON ".NET" PLATFORM


นายกิตติ์ ก้อนทอง รหัส 45360054

นายจตุรัฐ คำขาว รหัส 45360088

นายวีรชัย วรวิริยะประเสริฐ รหัส 45360450



ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 3 พ.ค. 2553
เลขทะเบียน 14993421
เลขเรียกหนังสือ ปร. 06 ก 2549
มหาวิทยาลัยนเรศวร


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2548

# ใบรับรองโครงงานวิศวกรรม

| | | | |
|---|---|---|---|
| **หัวข้อโครงงาน** | โปรแกรมสืบค้นภาพบนดอทเน็ต (.NET) แพลทฟอร์ม | | |
| **ผู้ดำเนินโครงงาน** | นายกิตติ์ | ก้อนทอง | รหัส 45360054 |
| | นายจตุรัฐ | คำขาว | รหัส 45360088 |
| | นายวีรชัย | วรวิริยะประเสริฐ | รหัส 45360450 |
| **อาจารย์ที่ปรึกษา** | ผู้ช่วยศาสตราจารย์ ดร. สุชาติ แย้มเม่น | | |
| **สาขาวิชา** | วิศวกรรมคอมพิวเตอร์ | | |
| **ภาควิชา** | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ | | |
| **ปีการศึกษา** | 2548 | | |

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. สุชาติ แย้มเม่น)

.....กรรมการ

(อ. ศิริพร เดชะศิลารักษ์)

.....กรรมการ

(ดร.ไพศาล มุณีสว่าง)

| | | | |
|---|---|---|---|
| **หัวข้อโครงงาน** | โปรแกรมสืบค้นภาพบนดอทเน็ต (.NET) แพลทฟอร์ม | | |
| **ผู้ดำเนินโครงงาน** | นายกิตติ์ | ก้อนทอง | รหัส 45360054 |
| | นายจตุรัฐ | คำขาว | รหัส 45360088 |
| | นายวีรชัย | วรวิริยะประเสริฐ | รหัส 45360450 |
| **อาจารย์ที่ปรึกษา** | ผู้ช่วยศาสตราจารย์ ดร. สุชาติ แย้มเม่น | | |
| **สาขาวิชา** | วิศวกรรมคอมพิวเตอร์ | | |
| **ภาควิชา** | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ | | |
| **ปีการศึกษา** | 2548 | | |

.....


## บทคัดย่อ

โครงงานนี้เป็นโครงงานที่ศึกษาเกี่ยวกับการพัฒนาโปรแกรมค้นหาภาพในฐานข้อมูล โดยการวิเคราะห์ภาพจะใช้ทฤษฎีกราฟแสดงความถี่ความเข้มสี (Color Histrogram) และทฤษฎีกาบอร์เวฟเลต (Gabor wavelet) ในการวิเคราะห์สีและพื้นผิวของภาพเพื่อสำหรับเก็บเป็นดัชนี (Index)ไว้ในฐานข้อมูล โดยดัชนีในฐานข้อมูลจะมีไว้สำหรับเปรียบกับดัชนีของภาพตัวอย่างรวมถึงมีฟังก์ชั่นการป้อนกลับ (Relevance Feedback) เพื่อ ทำให้การค้นหานั้นมีประสิทธิภาพ จากการค้นหานั้นเราศึกษา 4 กรณี คือ ค้นหาจากสี ค้นหาจากพื้นผิว ค้นหาจากสีและพื้นผิวพร้อมกัน และค้นหาจากสีและพื้นผิวรวมทั้งมีการป้อนกลับด้วย ซึ่งโปรแกรมจะถูกพัฒนาบนวินโดวแอพพลิเคชั่น (WindowApplication) ของไมโครซอฟต์วิชวลสตูดิโอ ดอทเน็ต (Microsoft visual studio.NET) โดยใช้ภาษา C# และใช้โปรแกรม Mysql จัดการกับฐานข้อมูล

จากผลการทดลองได้ผลสรุปว่าผลที่ได้จะตรงตามความต้องการมากที่สุดเมื่อมีการป้อนกลับ (Relevance Feedback) ด้วย ซึ่งทั้งนี้ขึ้นอยู่กับขนาดของฐานข้อมูลที่มีด้วยเช่นกัน

| | | | |
|---|---|---|---|
| **Project title** | Image Retrieval Software Application on .NET Platform | | |
| **Name** | Mr. Khitt | kongtong | ID. 45360054 |
| | Mr. Chaturat | Khamkhao | ID. 45360088 |
| | Mr. Weerachai | Voraviriyaprasert | ID. 45360450 |
| **Project advisor** | Assistant Professor Suchart Yammen , Ph.D. | | |
| **Major** | Computer Engineering | | |
| **Department** | Electrical and Computer Engineering | | |
| **Academic year** | 2005 | | |


## Abstract

This project studies and develops search engine, a window software application for indexing and retrieval of pictures. The system analysts image content by color histogram and Gabor wavelet transformation to obtain color and texture descriptors for indexing image database. These indexes are used by content matching matrix to compare the content of a given query with other pictures stored in the database. The proposed search engine has a relevance feedback function which implements user-computer interface to increase its retrieval accuracy. It offers for methods for searching: search-by-color, search-by-texture, search-by-color-and-texture, and relevance feedback. The programs we used in this project include C# Window Application of Microsoft Visual Studio .NET 2003 and MySQL. The experiment results show that the proposed system performs well with the relevance feedback function, and the retrieval performance depends on the size of database used.

# กิตติกรรมประกาศ

ในการทำโครงงานวิศวกรรมครั้งนี้ คณะผู้จัดทำขอกราบขอบพระคุณ ดร.ไพศาล มุณีสว่าง ที่ได้ให้คำปรึกษาโครงงานนี้ ทั้งทฤษฎีและขั้นตอนการปฏิบัติงานต่างๆ และขอบพระคุณ ผศ.ดร.สุชาติ แย้มเม่น และ อ. ศิริพร เดชะศิลารักษ์ ที่ได้ ที่ได้เสียสละเวลาเพื่อทำการตรวจสอบการทำงานและชี้แนวทางในการแก้ไขปัญหาโครงงานนี้

นายกิตติ์ ก้อนทอง

นายจตุรัฐ คำขาว

นายวีรชัย วรวิริยะประเสริฐ

# สารบัญ

หน้า


บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ฉ

สารบัญรูป ..... ช

บทที่ 1 บทนำ

1.1 ปัญหาและที่มาของปัญหา..... 1

1.2 วิธีการค้นหาภาพโดยใช้องค์ประกอบพื้นฐานของภาพ..... 2

1.3 วัตถุประสงค์..... 2

1.4 ขอบข่ายของโครงงาน..... 2

1.5 ผลที่คาดว่าจะได้รับ..... 3

1.6 กิจกรรมดำเนินงาน..... 3

1.7 งบประมาณที่ใช้..... 4

บทที่ 2 ทฤษฎีเบื้องต้น

2.1 การค้นหาภาพโดยใช้องค์ประกอบพื้นฐานของภาพ..... 5

2.2 ส่วนของสี (Color) ..... 5

2.2.1 กราฟแสดงความถี่ความเข้มสี (Color Histogram)..... 5

2.2.2 การอธิบายภาพโดยใช้ Color Histogram..... 6

2.3 ส่วนของพื้นผิว (Texture) ..... 8

2.3.1 กาบอร์เวฟเลต (Gabor Wavelet)..... 9

2.3.2 ทฤษฎีกาบอร์เวฟเลต (Garbor Wavelet)..... 11

2.4 ระบบการป้อนกลับ (Relevance Feedback)..... 12

2.5 ดอทเน็ต แพลทฟอร์ม (.NET Platform) โดย C#..... 14

# สารบัญ (ต่อ)

หน้า


บทที่ 3 วิธีการดำเนินโครงงาน

3.1 การค้นหาภาพ ..... 16

3.1.1 เลือกภาพต้นแบบ ..... 17

3.1.2 ทำการวิเคราะห์สีและพื้นผิว ..... 17

3.1.3 เพิ่มข้อมูลของภาพลงในฐานข้อมูล ..... 21

3.1.4 เปรียบเทียบค่าในฐานข้อมูล ..... 21

3.1.5 แสดงภาพที่ใกล้เคียงกับภาพต้นแบบ ..... 22

3.1.6 ระบบป้อนกลับ(Relevance Feedback) ..... 22

3.1.7 ได้ภาพตรงกับความต้องการมากขึ้น ..... 23

3.2 ฐานข้อมูล (Database) ..... 24

บทที่ 4 ผลการทดลอง

4.1 การทดลองค้นหาภาพโดยวิเคราะห์ด้วยสี (Color) ..... 26

4.2 การทดลองค้นหาภาพโดยวิเคราะห์ด้วยพื้นผิว(Texture) ..... 28

4.3 การทดลองค้นหาภาพโดยวิเคราะห์ด้วย สีและพื้นผิวพร้อมกัน ..... 29

4.4 การทดลองค้นหาภาพโดยวิเคราะห์ด้วยสีและพื้นผิวพร้อมด้วยการป้อนกลับ ..... 31

4.5 เปรียบเทียบผลการทดลอง ..... 32

บทที่ 5 บทสรุป

5.1 สรุปผลการทดลอง ..... 33

5.2 ปัญหาที่พบ ..... 35

5.3 ข้อเสนอแนะ ..... 35

เอกสารอ้างอิง ..... 36

ประวัติผู้เขียนโครงงาน ..... 37

# สารบัญตาราง

ตารางที่ หน้า

4.1 บันทึกค่าการทดลองการค้นหาด้วยสี (Color)..... 27
4.2 บันทึกค่าการทดลองการค้นหาด้วยพื้นผิว (Texture)..... 29
4.3 บันทึกค่าการทดลองการค้นหาด้วยสีและพื้นผิว (Color&Texture)..... 30
4.4 บันทึกค่าการทดลองการค้นหาด้วยสีและพื้นผิวพร้อมทั้งกระบวนการป้อนกลับ..... 32
5.1 เปรียบเทียบค่าเฉลี่ยค่าความถูกต้อง (Average Precision)ของทั้ง 4 วิธี..... 34

# สารบัญรูป


| รูปที่ | | หน้า |
|---|---|---|
| 2.1 | รูประบบของ RGB | 6 |
| 2.2 | รูปการเก็บค่าสี RGB | 6 |
| 2.3 | รูปแสดงคัลเลอร์ฮิสโตรแกรม (Color Histogram) | 7 |
| 2.4 | รูปแสดงการแบ่งคัลเลอร์ฮิสโตรแกรม(Color Histrogram) ออกเป็นบิน (Bin) | 8 |
| 2.5 | รูปตัวอย่างของพื้นผิว (Texture) | 8 |
| 2.6 | รูปตัวอย่างที่ทำการทดลองผ่านฟังก์ชั่นกาบอร์เวฟเลต (Gabor Wavelet) | 9 |
| 2.7 | รูปกระบวนการผ่านกาบอร์ฟิลเตอร์ (Garbor filter) (1) | 10 |
| 2.8 | รูปกระบวนการผ่านกาบอร์ฟิลเตอร์ (Garbor filter) (2) | 10 |
| 2.9 | รูปผลการแสดงเมื่อผ่านกาบอร์ฟิลเตอร์ (Garbor filter) แล้ว | 11 |
| 2.10 | รูปการทำงานของระบบการป้อนกลับ(Relevance Feedback) | 13 |
| 3.1 | รูปแบบการดำเนินงาน | 16 |
| 3.2 | แผนผังการทำงานของโปรแกรม | 17 |
| 3.3 | การแสดงส่วนประกอบของ RGB | 18 |
| 3.4 | การแสดงค่าฮิสโตรแกรม(Histrogram)เป็นกราฟ | 19 |
| 3.5 | รูปตัวอย่างการแบ่งกราฟแสดงความถี่ความเข้มสี (Color Histrogram) ออกเป็นบิน | 20 |
| 3.6 | ตัวอย่างการเก็บค่าความถี่ของสีลงในฐานข้อมูล | 20 |
| 3.7 | ตัวอย่างการเก็บค่าความถี่ของพื้นผิวลงในฐานข้อมูล | 21 |
| 3.8 | รูปการหาค่าระยะห่าง(Distance) | 22 |
| 3.9 | ตัวอย่างโปรแกรม | 23 |
| 3.10 | รูปฐานข้อมูลสี | 25 |
| 3.11 | รูปฐานข้อมูลพื้นผิว | 25 |
| 4.1 | แสดงการค้นหาจากฐานข้อมูล โดยการวิเคราะห์จากสี (Color) | 27 |
| 4.2 | แสดงการค้นจากฐานข้อมูล โดยการวิเคราะห์จากพื้นผิว (Texture) | 28 |
| 4.3 | แสดงการค้นหาจากฐานข้อมูล โดยการวิเคราะห์จากสีและพื้นผิวพร้อมกัน | 30 |
| 4.4 | แสดงการค้นหาจากฐานข้อมูลโดยการวิเคราะห์จากสีและพื้นผิวพร้อมทั้งกระบวนการป้อนกลับ | 31 |
| 4.5 | กราฟเปรียบเทียบการทำงานของแต่ละวิธี | 32 |
| 5.1 | กราฟเปรียบเทียบการทำงานของแต่ละวิธี | 34 |

# บทที่ 1
# บทนำ

## 1.1 ปัญหาและที่มาของปัญหา

ปัจจุบันเทคโนโลยีนั้นได้ก้าวหน้าไปอย่างรวดเร็ว ทำให้เกิดข้อมูลหรือทรัพยากรมากมายหนึ่งในนั้นก็คือ รูปภาพ การค้นหารูปภาพในปัจจุบันนั้นเป็นการยากที่จะหารูปภาพให้ได้ตรงตามความต้องการของผู้ค้นหา และใช้เวลารวดเร็ว ดั้งนั้นจึงมีการสร้างเครื่องมือค้นหารูปภาพขึ้น ซึ่งในปัจจุบันโปรแกรมค้นหา (Search Engine) ที่มีอยู่นั้นจะทำการหารูปภาพโดยค้นหารูปภาพตามที่ผู้ค้นหาป้อนความต้องการให้กับโปรแกรม แต่ก็มีรูปภาพมากมายที่ไม่ใช่ภาพที่ต้องการและไม่ตรงกับคุณสมบัติที่ป้อนให้ไป ดังนั้นจึงคิดว่าน่าจะมีการค้นหาไฟล์รูปภาพด้วยวิธีอื่นที่สะดวกแม่นยำ และรวดเร็วกว่าเดิม โดยน่าจะมีการนำภาพตัวอย่างที่ผู้ต้องการหานั้นมาใช้ค้นหารูปภาพต่าง ๆ ในฐานข้อมูลเพื่อที่จะให้ตอบสนองต่อผู้ใช้งานมากขึ้น ฉะนั้นในปัจจุบันจึงมีการค้นหารูปภาพแบบโดยใช้องค์ประกอบพื้นฐานของภาพ (Content-based Image Retrieval (CBIR)) ขึ้น

**การค้นหาภาพโดยใช้องค์ประกอบพื้นฐานของภาพ (Content-based Image Retrieval:CBIR)**

หลักการง่าย ๆ ของการค้นหาภาพโดยใช้องค์ประกอบพื้นฐาน (CBIR) นั้นก็ คือ การนำภาพมาวิเคราะห์ให้ได้ค่าดัชนี (Index) ของภาพซึ่งเป็นคุณสมบัติพื้นฐานของภาพนั้น คุณสมบัติพื้นฐานของภาพจะประกอบไปด้วย สี (Color) และพื้นผิว (Texture) คุณสมบัติเหล่านี้จะใช้ในการอธิบายองค์ประกอบหรือลักษณะของภาพเพื่อบอกให้เข้าใจว่าภาพนั้นเป็นภาพอะไร ซึ่ง CBIR จะยอมให้ผู้ใช้นั้นกำหนดภาพปรัศนี (ภาพที่ต้องการหา) โดยการส่งภาพตัวอย่างนั้นไปเป็นภาพเริ่มต้น และระบบจะทำการค้นหารูปภาพเหล่านั้นจากการวิเคราะห์รูปร่างลักษณะ (Image index) ที่เหมือนกับรูปตัวอย่างออกมาซึ่งจะมีความใกล้เคียงกับรูปภาพตัวอย่าง

โดยในส่วนของโปรแกรมค้นหานั้น จะใช้การเปรียบเทียบภาพตัวอย่างกับฐานข้อมูลด้วยการวิเคราะห์ด้วย สี และพื้นผิว (Color and Texture) และแสดงผลภาพที่ใกล้เคียงกับรูปตัวอย่างออกมา

วิธีการค้นหาแบบ CBIR มีข้อดี คือ ทำการค้นหารูปภาพได้รวดเร็ว ใกล้เคียงกับรูปตัวอย่าง เหมาะสำหรับที่จะค้นหารูปที่ต้องการในฐานข้อมูลขนาดใหญ่

## 1.2 วิธีการค้นหาภาพโดยใช้องค์ประกอบพื้นฐานของภาพ

**1.2.1 การค้นหาโดยวิเคราะห์ด้วยสี (Color Retrieval)**

เป็นกระบวนการวิเคราะห์ความคล้ายคลึงของสี โดยการคำนวณจากกราฟแสดงความถี่ความเข้มสี (Color Histrogram) แสดงสัดส่วนของ จุด (Pixels) ของแต่ละสีในรูปภาพ

**1.2.2 การค้นหาโดยวิเคราะห์ด้วยพื้นผิว (Texture Retrieval)**

เป็นกระบวนการวัดความคล้ายคลึงของความละเอียด(Texture) ไว้สำหรับวัดพื้นผิวที่มีความละเอียดใกล้เคียงกันแยกออกจากกันเช่น สีของ น้ำทะเลและท้องฟ้า

**1.2.3 ดัชนีข้อมูลภาพ (Image Indexing)**

เป็นกระบวนการทำภาพให้เป็นข้อมูลในรูปแบบของตัวเลขเพื่อที่จะนำไปคำนวณและเก็บในฐานข้อมูล

**1.2.4 เปรียบเทียบภาพ (Content Matching)**

เป็นกระบวนการนำข้อมูลของแต่ละภาพที่ต้องการหามาเปรียบเทียบกัน

**1.2.5 การป้อนกลับ (Relevance Feedback)**

เป็นกระบวนการที่ให้ผู้ใช้สามารถป้อนกลับ ภาพหลังจากที่คอมพิวเตอร์สามารถเรียนรู้ลักษณะของภาพต้นแบบแล้วนำไปเปรียบเทียบกับในฐานข้อมูล

## 1.3 วัตถุประสงค์

1.3.1 ศึกษากระบวนการป้อนกลับ (Relevance Feedback) และ การค้นหาโดยใช้องค์ประกอบของภาพ (Content-based Image Retrieval) ซึ่งแยกออกเป็น 2 วิธี คือ การค้นหาด้วยสี (Colour Retrival) และการค้นหาด้วยพื้นผิว (Texture Retrival )

1.3.2 จัดทำฐานข้อมูลโดยใช้ตัวจัดการฐานข้อมูลมายเอสคิวแอล (M ySQL) ในการเก็บแฟ้มภาพ (File Picture) และดัชนีของภาพ (Image Index) ที่ได้จากการวิเคราะห์รูป เพื่อความ เป็นระเบียบและง่ายต่อการค้นหา

1.3.3 สร้างโปรแกรมค้นหาภาพ (Search Engine)โดยใช้ โปรแกรมดอทเน็ต (.NET) ที่สามารถค้นหาไฟล์ภาพ (Picture) ในฐานข้อมูลที่ได้เตรียมไว้แล้ว โดยใช้วิธีการหารูปภาพโดยการเอารูปภาพค้นหารูปภาพที่คล้ายคลึงกัน

## 1.4 ขอบข่ายของงานโครงงาน

สร้างโปรแกรมค้นหา (Search Engine) ในการค้นหาภาพซึ่งเราจะใช้หลักการวิเคราะห์จากดัชนีภาพ (Image Index) ในการเปรียบเทียบลักษณะ ของรูปตัวอย่าง (Picture) ที่ต้องการค้นหา กับ ดัชนีภาพ (Image Index) ของไฟล์ (File picture) ในฐานข้อมูล โดยในการพัฒนานั้นจะใช้วิชวลสตูดิโอ ดอท เน็ต

(Visual Studio .NET) วิเคราะห์รูปภาพและสร้างโปรแกรมค้นหาภาพ (Search Engine) และใช้ตัวจัดการฐานข้อมูล MySql ช่วยในการจัดการฐานข้อมูล

## 1.5 ผลที่คาดว่าจะได้รับ

1.5.1 สามารถสร้างโปรแกรมค้นหา (Search Engine) เพื่อใช้ในการค้นหาไฟล์ (File picture) ได้อย่างรวดเร็วและมีประสิทธิภาพ

1.5.2 สร้างฐานข้อมูลที่จะใช้เก็บไฟล์รูปภาพ (File Picture) และสามารถแก้ไขปรับปรุงฐานข้อมูลให้มีประสิทธิภาพได้และนำมาใช้ร่วมกับโปรแกรมค้นหาที่สร้างขึ้น

## 1.6 กิจกรรมดำเนินงาน

| เวลา / ขั้นตอนการดำเนินงาน | ปี 2548 | | ปี 2549 | | | | |
|---|---|---|---|---|---|---|---|
| | พ.ย. | ธ.ค. | ม.ค. | ก.พ. | มี.ค. | เม.ย. | พ.ค. |
| หาวิธีวิเคราะห์รูปภาพ | | | | | | | |
| ศึกษาวิธีการวิเคราะห์รูปภาพและคิดวิธีการค้นหารูปภาพ | | | | | | | |
| ศึกษาโปรแกรมที่ใช้ในการเขียนโปรแกรมค้นหารูปภาพ | | | | | | | |
| ทำการเขียนโปรแกรมค้นหารูปภาพ | | | | | | | |
| ทดลองตัวโปรแกรม | | | | | | | |
| แก้ไขข้อบกพร่องของโปรแกรม | | | | | | | |
| ทดสอบการนำโปรแกรมไปใช้งาน | | | | | | | |
| ตรวจสอบและแก้ไข | | | | | | | |
| สรุปและจัดทำเอกสาร | | | | | | | |

## 1.7 งบประมาณที่ใช้

| | | | |
|---|---|---|---|
| 1.7.1 | ค่าใช้จ่ายในการซื้ออุปกรณ์ทางคอมพิวเตอร์ | 2000 | บาท |
| 1.7.2 | ค่าใช้จ่ายในการทำรายงาน | 500 | บาท |
| 1.7.3 | ค่าใช้จ่ายเบ็ดเตล็ด | 500 | บาท |
| รวมทั้งสิ้น | | 3000 | บาท |

# บทที่ 2
# ทฤษฎีเบื้องต้น

วิธีที่ใช้ในโปรแกรมคือการค้นหาภาพโดยใช้องค์ประกอบพื้นฐานของภาพ (Content-Based - Image Retrieval) ซึ่งจะเป็นการนำรูปตัวอย่างมาวิเคราะห์ แล้วนำเปรียบเทียบกับภาพตัวอย่างในฐานข้อมูล ซึ่งในที่นี้โปรแกรมนี้จะทำการวิเคราะห์ สี และพื้นผิว (Color and Texture) โดยจะใช้ทฤษฎีเกี่ยวกับ สีและพื้นผิว ซึ่งทฤษฎีต่างๆนั้นจะกล่าวดังต่อไปนี้

## 2.1 การค้นหาภาพโดยใช้องค์ประกอบพื้นฐานของภาพ(Content-Based Image Retrieval)

ในที่นี้จะนำรูปภาพผ่านคุณสมบัติพื้นฐาน (low-level features) จากนั้นจะใช้วิธีวิเคราะห์รูปภาพ 2 แบบคือ สี (Color) และพื้นผิว (Texture) เมื่อได้ ดัชนี(index) แล้วก็จะนำดัชนีที่ได้ไปเปรียบเทียบกับภาพในฐานข้อมูล แล้วคำนวณหาค่าความแตกต่าง (Distance) เพื่อนำค่านี้ไปหาภาพที่ต้องการในฐานข้อมูล

ซึ่งในส่วนของการวิเคราะห์นั้น ได้ทำการวิเคราะห์ 2 วิธีคือ สี (Color) และ พื้นผิว (Texture) โดยทฤษฎีที่เกี่ยวข้องกับกับ 2 วิธีนี้มีดังนี้

1. สี (Color) วิเคราะห์ด้วยทฤษฎีกราฟแสดงความถี่ความเข้มสี (Color Histrogram)
2. พื้นผิว (Texture) วิเคราะห์ด้วยทฤษฎีกาบอร์เวฟเลต (Gabor Wavelet)

ส่วนของการค้นหาภาพแบบย้อนกลับจะใช้ทฤษฎีการป้อนกลับ (Relevance Feedback)

## 2.2 ส่วนของสี (Color)

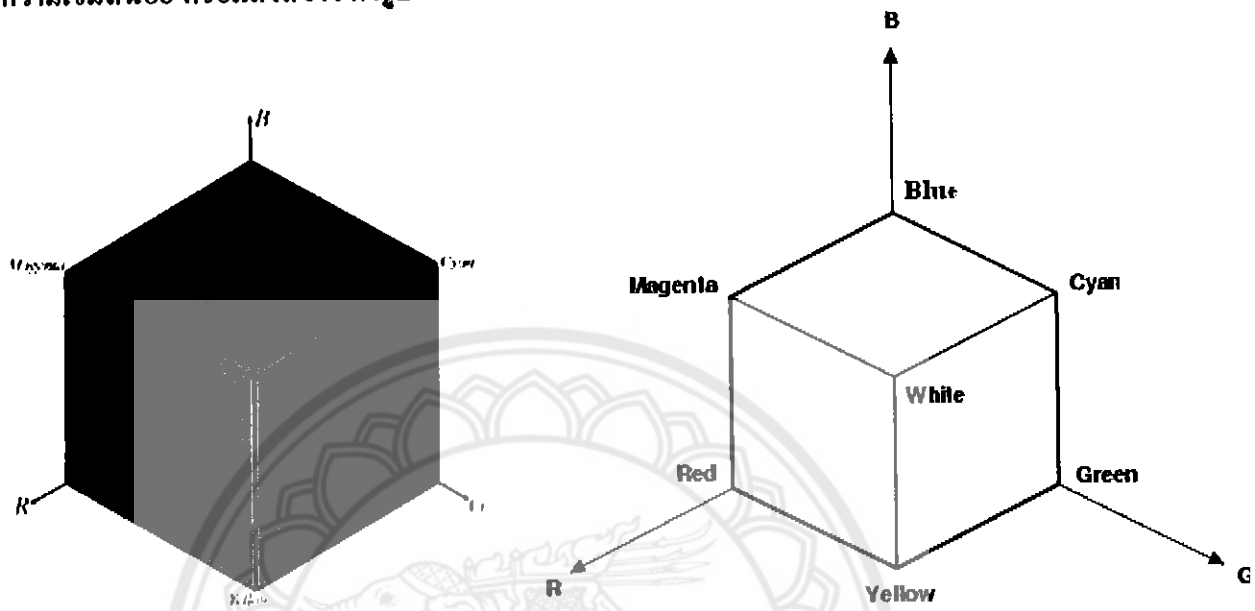
### 2.2.1 กราฟแสดงความถี่ความเข้มสี (Color Histrogram)

**มาตรฐานของสี**

ในปัจจุบันนี้มีการใช้มาตรฐานสีอยู่หลายระบบในแต่ละระบบก็จะมีการกำหนดแตกต่างกันไป แต่โดยทั่วไปก็จะมีการแบ่งที่คล้ายกันอยู่คือการแทนสีด้วยจุด โดยจะแทนจุดในสเปส 3 มิติ มีแกนอ้างอิงแต่ละแกน อย่างเช่น แกนของสี( สีแดง, สีเขียว, สีน้ำเงิน) แกนของความสว่าง (Lightness) แกนของความบริสุทธิ์ของสี (Saturation) ซึ่งแกนแต่ละแกนนี้จะเป็นอิสระต่อกัน เราจะยกตัวอย่างของระบบที่นิยมใช้ ระบบอาร์จีบี ( RGB) และเอชเอสวี (HSV) ทั้ง 2 ระบบนี้ เป็นระบบของมาตรฐานของสีที่มีความคล้ายคลึงกัน

**ระบบสี RGB**

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




รูปที่ 2.1 รูป ระบบของ RGB

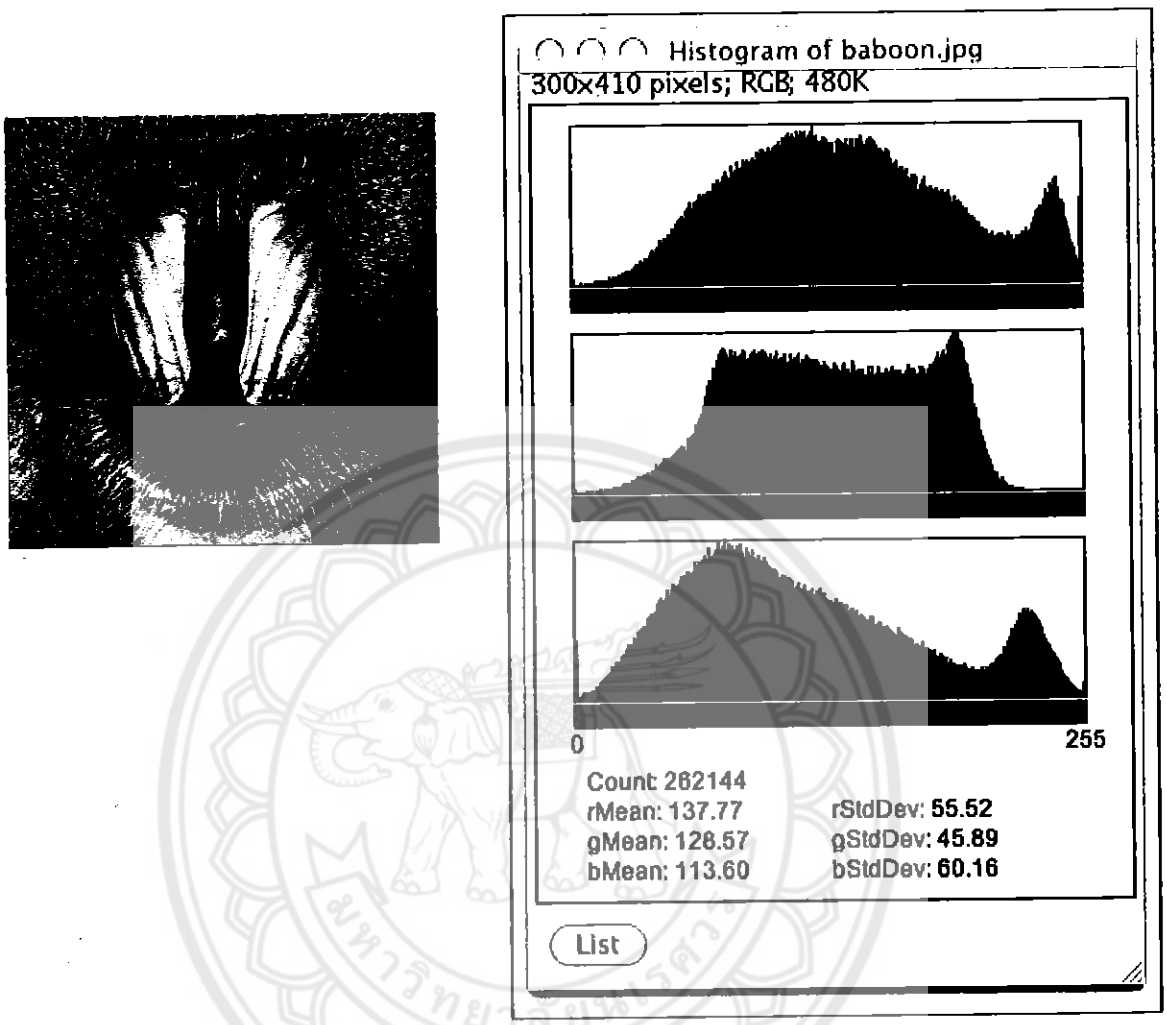
**2.2.2 การอธิบายภาพโดยใช้กราฟแสดงความถี่ความเข้มสี (Color Histogram)**

ภาพ ๆ หนึ่งจะประกอบด้วยจุดของเม็ดสีต่างๆ หลายจุดด้วยกัน ซึ่งถ้ามองด้วยตาเปล่าจะเป็นเม็ดสีรวมกันเป็นภาพโดยแต่ละเม็ดสีจะมีค่าของสีที่แตกต่างกันออกไป ซึ่งถ้าเป็นระบบสี RGB ก็จะประกอบด้วยสี แดง เขียว น้ำเงินดังภาพ

| R | G | B | |
|---|---|---|---|
| 0 | 0 | 0 | black |
| 255 | 0 | 0 | red |
| 0 | 255 | 0 | green |
| 0 | 0 | 255 | blue |
| 0 | 255 | 255 | cyan |
| 255 | 0 | 255 | magenta |
| 255 | 255 | 0 | yellow |
| 255 | 255 | 255 | white |

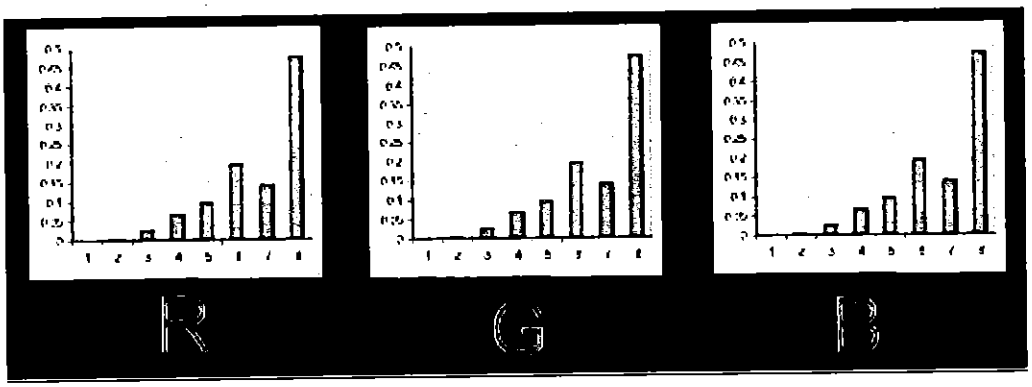
**รูปที่ 2.2 รูปการเก็บค่าสี RGB**

แล้วจะนำค่าสีแดง เขียว และ น้ำเงินของแต่ละจุดมาพล๊อตกราฟได้ดังภาพ




รูปที่ 2.3 รูปแสดงคัลเลอร์ฮิสโตรแกรม (Color Histogram)

ซึ่งค่าที่ได้ของแต่ละสี จะมีค่าตั้งแต่ 0 ถึง 255 เราจะแบ่งออกเป็น บิน (Bin) เพื่อให้ง่ายต่อการเก็บลงฐานข้อมูลโดยจำนวนบินนั้นขึ้นอยู่กับ ความละเอียดของภาพที่เราต้องการ ยิ่งจำนวนบินมากเท่าไหร่ในการค้นหาภาพก็จะมีความถูกต้องมากขึ้นเท่านั้น แต่ทั้งนั้นฐานข้อมูลก็ต้องเก็บข้อมูลที่มากด้วยเช่นกัน

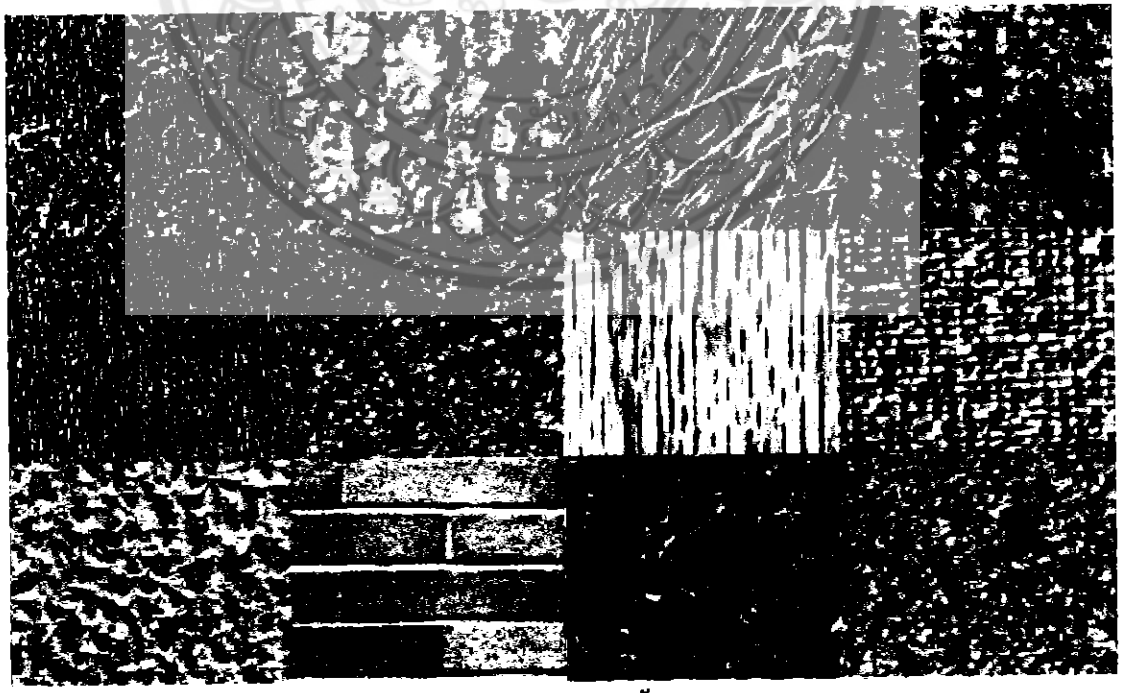



**รูปที่ 2.4 รูปแสดงการแบ่งคัลเลอร์ฮิสโตรแกรม(Color Histrogram) ออกเป็นบิน (Bin)**

โดยเราจะเก็บค่าสีลงในฐานข้อมูลตามความถี่ของแต่ละบิน

## 2.3 ส่วนของพื้นผิว (Texture)

พื้นผิวในที่นี้จะหมายถึงส่วนที่แสดงถึงความลึกความตื้นของรูปภาพ โดยพื้นผิวนี้จะทำให้รูปภาพดูมีมิติ รวมถึงทั้งรายละเอียดที่รวมอยู่ในรูปภาพหรือสิ่งที่แสดงในรูปภาพนั้น ซึ่งรายละเอียดเหล่านี้จะประกอบรวมกันขึ้นทำให้รูปมีความลึกหรือมีมิตินั้นเอง



**รูปที่ 2.5 รูปตัวอย่างของ พื้นผิว (Texture)**

ในการวิเคราะห์ส่วนของพื้นผิว (Texture) นั้นเราจะใช้ทฤษฎีที่เกี่ยวข้องคือ Gabor Wavelet ซึ่งทฤษฎีจะมีดังนี้

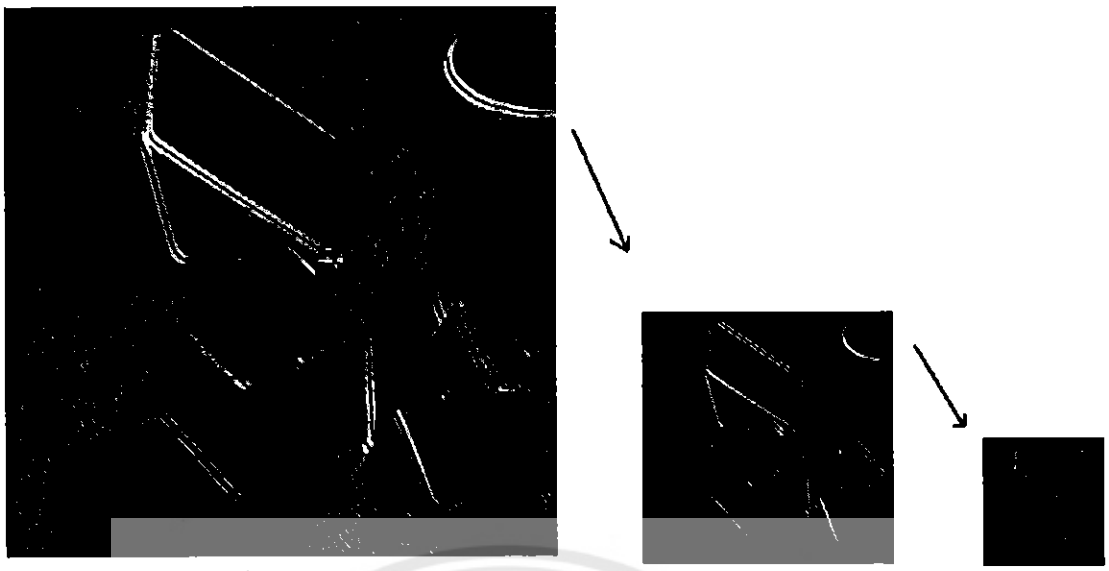
**2.3.1 กาบอร์เวฟเลต (Gabor Wavelet)**

ทฤษฎีกาบอร์เวฟเลต (Gabor Wavelet) จะใช้ฟังก์ชั่นของกาบอร์เวฟเลต (Gabor Wavelet) มาเป็นฟิลเตอร์หรือตัวกรอง โดยจะนำรูปมาผ่านฟิลเตอร์เพื่อให้ได้ความถี่หรือข้อมูลที่เราต้องการ ให้อยู่ในรูปแบบข้อมูลของรูปภาพ

โดยเมื่อทำการนำรูปมาผ่านฟิลเตอร์หรือฟังก์ชั่นของกาบอร์เวฟเลตแล้วจะได้ผลดังตัวอย่างนี้

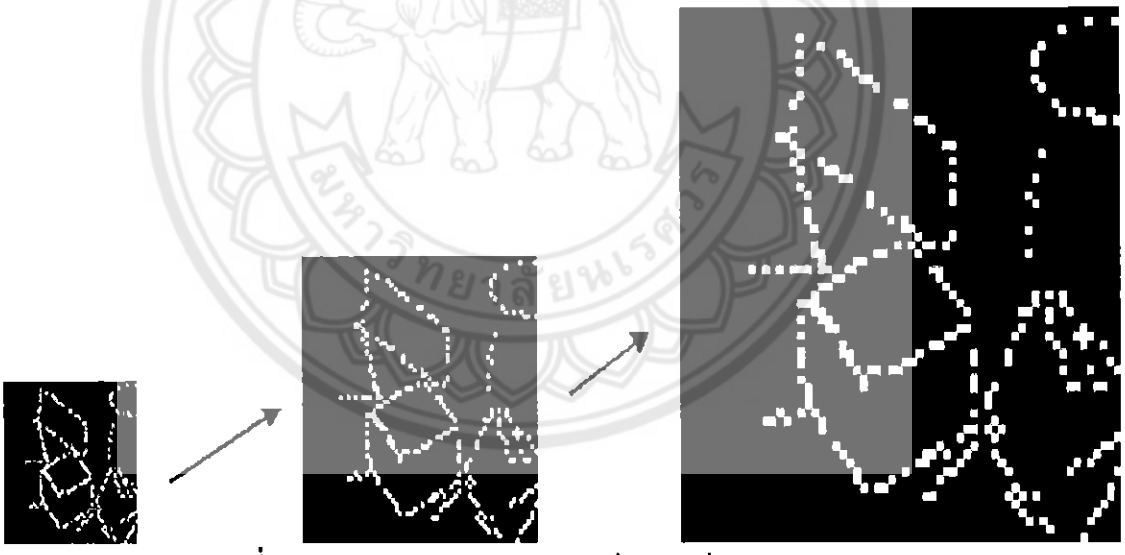
รูปที่ 2.6 รูปตัวอย่างที่ทำการทดลองผ่านฟังก์ชั่นกาบอร์เวฟเลต (Gabor Wavelet)

ซึ่งเมื่อทำการนำรูปผ่านเข้าฟิลเตอร์หรือฟังก์ชั่นกาบอร์เวฟเลตแล้ว รูปก็จะเปลี่ยนแปลงโดยจะเปลี่ยนรูปให้ไปเป็นระบบภาพขาว-ดำ (Gray Scale) เพื่อให้เห็นความแตกต่างของความลึกและมิติ จากนั้นทำการลดขนาดของรูปลงจนสามารถแยกส่วนที่สามารถแสดงให้เห็นถึงพื้นผิวที่แตกต่าง ดังรูป



รูปที่ 2.7 รูปกระบวนการผ่านกาบอร์ฟิลเตอร์ (Garbor filter) (1)

จากนั้นเมื่อทำการกำหนดส่วนที่แบ่งส่วนของพื้นผิวได้แล้ว ก็ทำการกำหนดจุดไว้ดังรูปและทำการขยายภาพให้กลับมาขนาดปกติเพื่อแสดงให้เห็นส่วนที่แสดงไว้



**รูปที่ 2.8** รูปกระบวนการผ่านกาบอร์ฟิลเตอร์ (Garbor filter) (2)

เมื่อผ่านกระบวนการกาบอร์เวฟเลต แล้วจะได้รูปดังนี้

รูปที่ 2.9 รูปผลการแสดงเมื่อผ่านกาบอร์ฟิลเตอร์ (Garbor filter) แล้ว

**2.3.2 ทฤษฎีกาบอร์เวฟเลต (Garbor Wavelet )**

ในทางการวิเคราะห์พื้นผิว (Texture) นั้น เราใช้ กาบอร์เวฟเลต (Gabor Wavelet) มาวิเคราะห์ ซึ่งมีการใช้กาบอร์ฟิลเตอร์ (Garbor filter) มาช่วยในการกรองภาพที่เราต้องการ โดยที่กาบอร์ฟิลเตอร์จะมีฟังก์ชั่นการทำงานเป็นสมการทางคณิตศาสตร์ ซึ่งกาบอร์เวฟเลตจะวิเคราะห์พื้นผิว (texture) และนำมาเปรียบเทียบความคล้ายคลึงกันซึ่งจะใช้สมการที่คอนโวลูชั่น (Convolution) ดังนี้

$$G_{mn}(x,y) = \sum_{x}\sum_{t} I(x-s, y-t)\psi_{mn}(s,t) \qquad (2.1)$$

$s,t$ เป็นขนาดของฟิลเตอร์

$\psi_{mn}$ เป็นส่วนของฟิลเตอร์

I คือ Image(x,y) ซึ่งมีขนาด $P \times Q$

สมการนี้หมายถึงกระบวนการที่นำรูป (Image) ซึ่งในที่นี้คือตัว I นำมาคอนโวลูชั่น (Convolution) กับฟังก์ชั่นของกาบอร์เวฟเลต (Gabor Wavelet) ซึ่งฟังก์ชั่นของกาบอร์เวฟเลต ในที่นี้หมายถึงตัว

$\psi_{mn}$ หมายถึง ฟิลเตอร์ซึ่งมีขนาด $(s,t)$ จากนั้นเมื่อคอนโวลูชั่นกับรูปภาพแล้วก็จะได้ $G_{mn}(x,y)$ ซึ่งเป็นตัวที่ผ่านกระบวนการ กาบอร์เวฟเลตแล้ว

Gabor Function (Mother wavelet)

$$\psi(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp[-\frac{1}{2}(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2})]\bullet\exp(j2\pi Wx) \qquad (2.2)$$

โดย W คือ Modulation Frequency

และ $\exp[-\frac{1}{2}(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2})]$ คือ ฟังก์ชั่นเกาส์เซี้ยน

ส่วนนี้จะอธิบายถึงตัวฟิลเตอร์ของฟังก์ชั่นกาบอร์เวฟเลต (Gabor Wavelet) ซึ่งจะเป็นสมการทางคณิตศาสตร์ ประกอบขึ้นจากการนำ Modulation Frequency และฟังก์ชั่นเกาส์เซี่ยนมาประกอบกัน

โดยเมื่อได้ $G_{mn}(x,y)$ แล้ว จะทำการหาค่าเฉลี่ย ( $\mu_{mn}$ ) และส่วนเบี่ยงเบนมาตรฐาน ($\sigma_{mn}$) จากสมการ

$$E(m,n) = \sum_x \sum_y |G_{mn}(x,y)|, m = 0,1,...,M-1; n = 0,1,...,N-1 \qquad (2.3)$$

$$\mu_{mn} = \frac{E(m,n)}{PxQ} \qquad (2.4)$$

$$\sigma_{mn} = \frac{\sqrt{\sum_x \sum_y (|G_{mn}(x,y)| - \mu_{mn})^2}}{PxQ} \qquad (2.5)$$

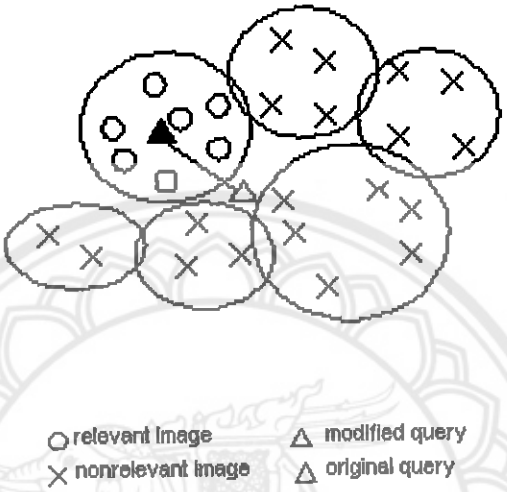
ซึ่งเมื่อได้ ค่าเฉลี่ย (Mean) และส่วนเบี่ยงเบนมาตรฐาน (Standard Divation) แล้วจะนำไปเก็บลงในฐานข้อมูลเพื่อใช้ในการเปรียบเทียบต่อไป

## 2.4 ระบบการป้อนกลับ (Relevance Feedback)

ระบบป้อนกลับ คือการสื่อสาร (Interactive) กันระหว่างผู้ใช้กับโปรแกรม โดยผู้ใช้จะทำการป้อนข้อมูลที่เป็นความต้องการของผู้ใช้กลับไป เพื่อให้โปรแกรมทำการเรียนรู้จากสิ่งที่ผู้ใช้ต้องการแล้วแสดงผลลัพธ์ที่ได้จากการป้อนกลับออกมา ซึ่งในการป้อนกลับแต่ละครั้งจะทำให้ผลลัพธ์ที่ได้นั้น

เป็นไปตามความต้องการของผู้ใช้ ยิ่งผู้ใช้ทำการป้อนกลับหลายครั้ง ก็ยิ่งทำให้ผลลัพธ์เป็นไปตามความต้องการของผู้ใช้มากขึ้น

โดย การป้อนกลับ (Relevance Feedback )จะมีหลักการทำงาน คือ จะเลือกแบ่งกลุ่มภาพที่ค้นหาได้ในรอบแรกออกเป็นกลุ่มที่ตรงกับภาพต้นแบบกับกลุ่มที่ไม่ตรง แล้วทำการย้ายจุดศูนย์กลางจากจุดศูนย์กลางของกลุ่มเก่า ไปจุดศูนย์กลางของกลุ่มใหม่ที่ตรงกับภาพต้นแบบมากที่สุด ดังในภาพ




**รูปที่ 2.10** รูปการทำงานของระบบการป้อนกลับ(Relevance Feedback)

| | | | |
|---|---|---|---|
| O | relevant image | = | ภาพที่ตรงตามที่ต้องการ |
| X | nonrelevant image | = | ภาพที่ไม่ตรงกับที่ต้องการ |
| ▲ | modified query | = | ค่าที่เรียงลำดับ (query) ที่ผ่านการคำนวณใหม่แล้ว |
| △ | original query | = | ค่าที่เรียงลำดับ (query) จากภาพเดิม |

จากรูปที่ 2.10 การทำงานของระบบการป้อนกลับ(Relevance Feedback) เมื่อมีการค้นหารูปออกมาแล้ว ภาพที่ได้อาจจะมีทั้งภาพที่ตรง(วงกลม)และภาพที่ไม่ตรงตามความต้องการ(กากบาท) ซึ่งเมื่อทำการป้อนกลับ แล้วระบบจะทำการย้ายจุดศูนย์กลางในการค้นหาจากที่เดิม(สามเหลี่ยมสีขาว) ซึ่งเป็นศูนย์กลางของภาพทั้งหมดไปไว้ที่จุดศูนย์กลางของภาพกลุ่มที่เราต้องการ(สามเหลี่ยมสีดำ) ดังรูป แล้วนำภาพในกลุ่มนั้นมาแสดง ซึ่งจะได้ภาพที่ตรงความต้องการมากขึ้นจากเดิม

ซึ่งสมการที่ใช้ในการย้ายจุดศูนย์กลางไปยังกลุ่มภาพที่ตรงความต้องการ คือ

$$Z = Z' + \alpha_R(\bar{X}' - Z') - \alpha_N(\bar{X}'' - Z') \quad (2.6)$$

$\alpha_N, \alpha_R$ = ค่าคงที่ค่าหนึ่ง
$\bar{x}'$ = ค่าเฉลี่ยของกลุ่มภาพที่เราเลือก
$\bar{x}''$ = ค่าเฉลี่ยของกลุ่มภาพที่เราไม่ได้เลือก
$Z'$ = ค่าดัชนี(Index) ของภาพต้นแบบ
$Z$ = ค่าดัชนี(Index) ค่าใหม่ที่จะนำไปใช้ค้นหาในครั้งต่อไป

## 2.5 ดอทเน็ต แพลทฟอร์ม (.NET Platform) โดย C#

C# เป็นภาษาที่ถูกสร้างขึ้นมาเพื่อทำงานบนดอทเน็ต แพลทฟอร์ม (.NET Platform) สร้างและทำงานในลักษณะของการเขียนโปรแกรมเชิงวัตถุ Object Oriented Programming (OOP) ได้อย่างสมบูรณ์ ซึ่งต่างกับ C++ ที่ยังทำงานในลักษณะของการเขียนโปรแกรมเชิงวัตถุได้บางส่วน ไลบรารีของ C# ถูกสร้างขึ้นเพื่อให้ทำงานได้ครอบคลุมตั้งแต่การสร้างรูปแบบการติดต่อแบบ GUI ไปจนถึงการแอ็คเซสฐานข้อมูลผ่านอินเตอร์เน็ตหรือแม้แต่การทำงานร่วมกับ XML เพื่อทำให้การแลกเปลี่ยนข้อมูลระหว่างแอพพลิเคชั่นทำได้อย่างสมบูรณ์ไม่ว่าข้อมูลนั้นจะอยู่บนแพลทฟอร์มใดก็ตาม

เมื่อเปรียบเทียบกับ C++ แล้ว การสร้างแอพพลิเคชั่นจะทำได้ง่ายกว่ามาก เนื่องจาก C# ถูกออกแบบมาเพื่อการสร้างแอพพลิเคชั่นให้ทำงานบนอินเตอร์เน็ต โดยตรง (.NET Framework) นอกจากนี้ C# เป็น Object Oriented Programming (OOP) อย่างสมบูรณ์ ไม่ว่าจะเป็น

- Encapsulation การรวมกลุ่มฟังก์ชั่นการทำงานของออบเจ็กต์ต่างๆ (Object Blueprint, Class) เพื่อให้โค้ดถูกเขียนขึ้นมาอย่างเป็นระเบียบ
- Polymorphism (Inheritance, Interfacing และ Overloading) การนำโค้ดที่เขียนขึ้นมาแล้วนั้นมาใช้ในงานอื่นได้อีก

.NET Framework คือ กรอบการทำงานของการเขียนโปรแกรมที่ไมโครซอฟท์คิดขึ้นมาเพื่อรับรองการติดต่อสื่อสาร เพื่อแลกเปลี่ยนข้อมูล (Exchange Data) ระหว่างกันหรือแลกเปลี่ยนระหว่างแพลตฟอร์ม(Platform) ให้มีความสมบูรณ์ยิ่งขึ้น

แน่นอน การเขียนโปรแกรมบน Visual Basic ทำได้ง่ายกว่าแต่ประสิทธิภาพของโปรแกรมจะด้อยกว่าโปรแกรมที่เขียนขึ้นมาจาก C++ ในบางกรณี อย่างเช่นโปรแกรมที่ต้องติดต่อกับฮาร์ดแวร์ จะเลือกใช้ C++ แต่ถ้าต้องการความง่ายในการเขียนโปรแกรม โดยไม่ต้องคำนึงถึงประสิทธิภาพการทำงานมากนัก จะเลือกใช้ Visual Basic, C# จะรวมเอาลักษณะการเขียนโปรแกรมจากภาษาทั้งสอง เข้ามาไว้ เช่น C# จะไม่มี Overhead มากนัก เมื่อเทียบกับ Visual Basic

**จุดเด่นของภาษา C# ที่ถูกพัฒนาขึ้น**

ภาษา C# สนับสนุนการเขียนโปรแกรมแบบคอมโพเนนท์ (Component-Oriented Programming) ซึ่งประกอบด้วย พร็อพเพอร์ตี้ อีเวนต์ เดลอิเกต (Delegate) และ แอตทรีบิวต์ (Attribute) โดยที่แอตทรีบิวต์จะนำมาใช้เพื่อเพิ่มชนิดข้อมูลเมต้าดาต้า (Meta-Data) ก่อนออบเจ็กต์ใดๆ ส่วนในเวอร์ชั่นถัดไปจะสนับสนุนด้านการเขียนโปรแกรมแบบ Generic (คล้ายกับการเขียนโปรแกรม C++ ที่ใช้เทมเพลต ตัวอย่างเช่น Standard Template Library)

ภาษา C# ยังเพิ่มคำสั่งสำหรับการทำเอกสาร XML (eXtensible Markup Language) โดยตัวคอมไพล์ C# จะสร้างเอกสารโดยตรงจากซอร์สโค้ดโปรแกรม

ภาษา C# ยังปรับปรุงในเรื่องเกี่ยวกับหน่วยความจำที่จะช่วยลดเวลาในการแก้บักได้มาก การปรับปรุงนี้ก็คือการใช้ข้อมูลแบบ Type Safety และ Garbage Collection ซึ่งจะคอยปกป้องการรั่วไหลหน่วยความจำเมื่อมีการสร้างอินสแตนซ์ หรือ อ๊อบเจ็คต์ใหม่ โดยใช้โอเปอเรเตอร์ new และถ้าพบว่าอ๊อบเจ็คต์ไม่มีการใช้อีกต่อไป มันก็จะลบออกจากหน่วยความจำโดยอัตโนมัติ ภายใต้สภาปัตยกรรม .NET เราเรียกว่าโค้ดแบบถูกจัดการ (Managed code) ด้วยเหตุผลข้างต้นนี้ ภาษา C# จึงไม่มีพอยน์เตอร์และดีสตรัคเตอร์(แต่สามารถเพิ่มได้ และจะเป็นโค้ดที่ไม่ปลอดภัย หรือ Unsafety)

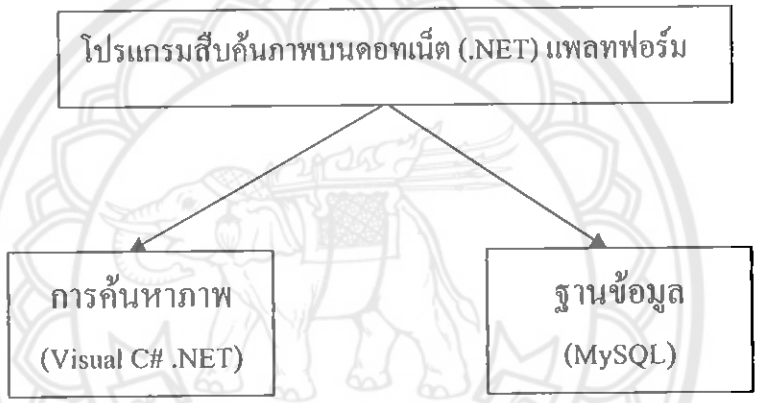
นอกจากนนี้แล้วภาษา C# ยังสามารถใช้ร่วมกับโค้ดเดิมที่มีอยู่แล้ว ไม่ว่าจะเป็น C++ หรือ Basic ซึ่งเป็นโค้ดที่ไม่ถูกจัดการ (Unmanaged or Native Code) เหตุผลหนึ่งที่ต้องสนับสนุนโค้ดดั้งเดิมเพราะไม่ต้องเสียเวลามาเขียนโค้ดไลบรารีใหม่ และบางหน่วยงานยังใช้โค้ดแบบเดิมๆอยู่ รวมทั้งไม่ต้องเสียค่าใช้จ่ายมาก

**ผู้สร้างภาษา C#**

ภาษา C# โดยรับการพัฒนาโดยนาย Andrers Hejlsberg ซึ่งเป็นผู้ที่มีชื่อเสียงคนหนึ่งในการเขียนคอมไพเลอร์ Turbo Pascal ในช่วงศตวรรษที่ 1980 นอกจากนั้นเขายังเป็นหัวหน้าทีมออกแบบแอพพลิเคชั่น Borland Delphi

# บทที่ 3
# วิธีการดำเนินการ

ในการดำเนินการ โปรแกรมสืบค้นภาพบนดอทเน็ต (.NET) แพลทฟอร์ม นั้นจะเป็นการสร้างโปรแกรมค้นหาภาพโดยใช้โปรแกรมทำงานร่วมกันของ 2 โปรแกรม โดยในส่วนแรกเป็นส่วนของโปรแกรมหลักที่ใช้ค้นหาจะเขียนขึ้นด้วยวิชวลซีชาร์ปดอทเน็ต (Visual C# .NET) ส่วนที่ 2 ใช้ในการเก็บข้อมูลและที่มาของภาพ (Path) ของภาพซึ่งใช้มายเอสคิวแอล (MySQL) ในการเก็บ โดยได้แบ่งการดำเนินงานออกเป็นขั้นตอนดังนี้




รูปที่ 3.1 รูปแบบการดำเนินงาน

## 3.1 การค้นหาภาพ

โปรแกรมที่ใช้ค้นหาภาพนั้น ได้พัฒนาขึ้นภายใต้สถาปัตยกรรม .NET โดยใช้ภาษา C# ซึ่งการที่จะค้นหาภาพได้นั้น เราต้องมีข้อมูลของภาพอยู่ในฐานข้อมูลก่อน โดยการสร้างโปรแกรมขึ้นเพื่อวิเคราะห์องค์ประกอบของภาพให้ออกมาเป็นค่าตัวเลข ซึ่งใช้แทนองค์ประกอบของสี และ พื้นผิว ในภาพ ในการค้นหานั้นจะเริ่มจากการนำภาพต้นแบบมาวิเคราะห์องค์ประกอบของสีและพื้นผิว เพื่อให้ได้ค่าเป็นตัวเลขแล้วนำไปเปรียบเทียบกับข้อมูลของภาพในฐานข้อมูล เพื่อที่จะหาภาพที่ใกล้เคียงกับภาพต้นแบบมากที่สุด 20 ภาพ และนำไปแสดงผลโดยลำดับการแสดงผลจะเรียงลำดับความคล้ายคลึงของภาพจากคล้ายมากลงมาเรื่อยๆ แต่อาจจะได้ภาพที่ไม่ตรงกับความต้องการทุกภาพ ดังนั้นจึงให้มีการป้อนกลับ (Feedback) เพื่อสอนให้ คอมพิวเตอร์ ทราบถึงความต้องการของผู้ใช้ สุดท้ายโปรแกรมก็จะแสดงภาพที่ได้จากการประมวลผลออกมา โดยจะเลือกภาพที่มีลักษณะภาพตามที่ผู้ใช้ได้ทำการ

ป้อนกลับ ทำให้การค้นหาภาพที่ขอบเขตได้น้อยลง สุดท้ายโปรแกรมก็จะแสดงภาพที่ได้จากการประมวลผลออกมา

จากหลักการทำงานดังกล่าวนำมาเขียนแผนภาพได้ ดังนี้

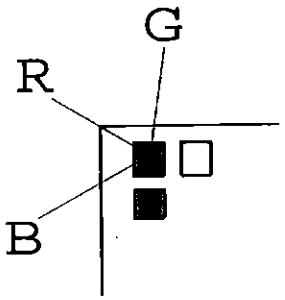
รูปที่ 3.2 แผนผังการทำงานของโปรแกรม

**3.1.1 เลือกภาพต้นแบบ**

เลือกภาพต้นแบบเพื่อนำมาหาภาพที่คล้ายคลึงกันในฐานข้อมูล หรือเลือกภาพที่ต้องการเพิ่มข้อมูลลงในฐานข้อมูล

**3.1.2 ทำการวิเคราะห์สีและพื้นผิว**

ในการวิเคราะห์สี จะใช้มาตรฐานสีแบบ RGB ในการวิเคราะห์คุณสมบัติของภาพโดยมาตรฐานสีแบบ RGB นี้จะมีการหาค่าสีแม่สี 3 สี ได้แก่ สีแดง สีเขียว และสีน้ำเงิน ซึ่งแม่สี 3 สีนี้นำมาประสมประสานกันจะได้สีออกมาแตกต่างกันหลากหลายสี



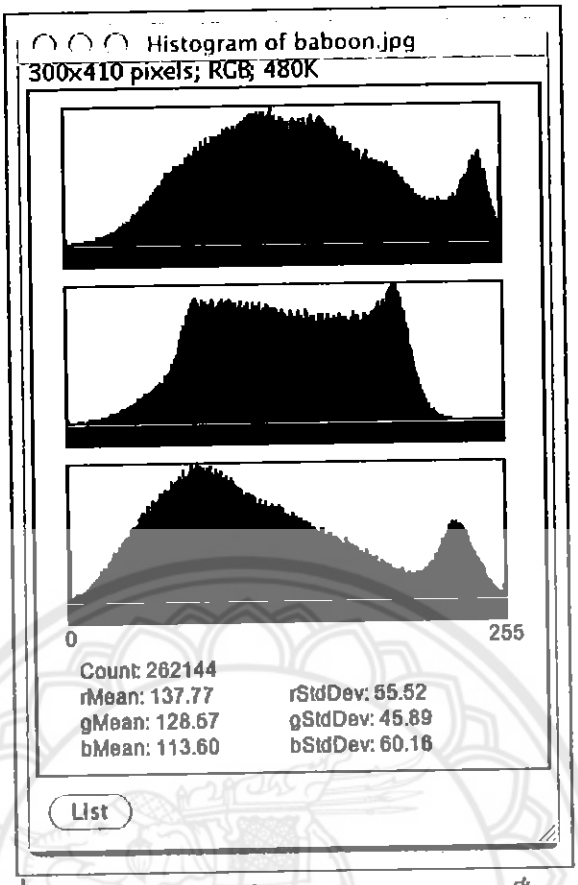

**รูปที่ 3.3** การแสดงส่วนประกอบของ RGB

รูปภาพแต่ละภาพก็คือการรวมตัวของเม็ดสีจำนวนมาก โดยรูปภาพที่แปลงเป็นไฟล์ดิจิตอลแล้วจะมีการมองรูปภาพออกเป็นพิกเซล โดยแต่ละรูปภาพก็มีขนาดแตกต่างกันออกไป

ตัวอย่างดังรูป รูปนี้มีขนาด 300 x 410 pixel

รูปที่มีขนาด 300 x 410 pixel รูปนั้นก็มีขนาด 123000 pixel ซึ่งก็คือมีเม็ดสี 123000 เม็ด ซึ่งการที่จะวิเคราะห์เม็ดสีแต่ละเม็ดนี้ว่ามีค่าสีแดง สีเขียว และสีน้ำเงินอย่างละเท่าไรทำได้โดยการพล็อตกราฟแสดงความถี่ความเข้มสี (Color Histogram)

โดยกราฟแสดงความถี่ความเข้มสีของรูปภาพแต่ละรูปนั้น จะมีการแยกสี แดง เขียว และน้ำเงินออกมาจึงทำให้ได้กราฟออกมา 3 กราฟ ซึ่งการกราฟแต่ละกราฟก็คือกราฟความถี่ของความเข้มสีแต่ละพิกเซล




รูปที่ 3.4 การแสดงค่าฮิสโตรแกรม(Histrogram)เป็นกราฟ

และเพื่อที่จะง่ายต่อการนำมาเปลี่ยนเทียบเราจึงเก็บค่าความถี่ของแต่ละสีเอาไว้ ตัวอย่างเช่นทำการเป็นค่าความถี่ของสีแดง ซึ่งสีแดงมีระดับความเข้มของสี 256 ระดับ ดังนั้นจึงต้องทำการการเก็บค่าความถี่ของสีแต่ระระดับลงในฐานข้อมูล ตัวเลขทั้งหมด 256 ค่า และทำการเก็บค่าความถี่ของสีเขียวและสีน้ำเงินแบบเดียวกัน จึงต้องมีฐานข้อมูลที่จะเก็บค่าคุณสมบัติของรูปภาพหนึ่งรูปถึง 768 ค่า ซึ่งมีความละเอียดมากเกินความจำเป็นต่อการใช้งาน และขนาดของฐานข้อมูลมีขนาดใหญ่เกินไป ดังนั้นจึงมีการตั้งเงื่อนไขในการแบ่งระดับความเข้มของสีแต่ละสี โดย สีหนึ่งสี จะแบ่งระดับออกเป็น 8 ระดับ จากเดิมที่มี 256 ระดับทำการตั้งเงื่อนไขให้เหลือ 8 ระดับ ดังนี้

ระดับที่1 ค่าสี มากกว่าหรือเท่ากับ 0 ถึง 31

ระดับที่2 ค่าสี มากกว่าหรือเท่ากับ 32 ถึง 63

ระดับที่3 ค่าสี มากกว่าหรือเท่ากับ 64 ถึง 95

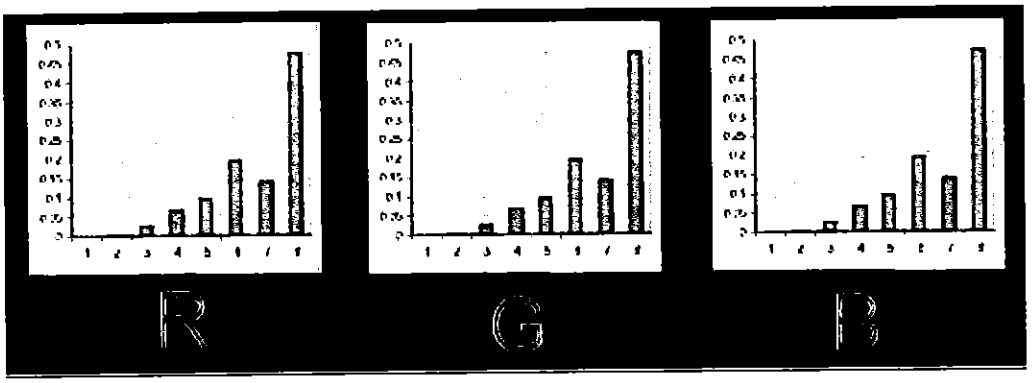
ระดับที่4 ค่าสี มากกว่าหรือเท่ากับ 96 ถึง 127

ระดับที่5 ค่าสี มากกว่าหรือเท่ากับ 128 ถึง 159

ระดับที่6 ค่าสี มากกว่าหรือเท่ากับ 160 ถึง 191

ระดับที่7 ค่าสี มากกว่าหรือเท่ากับ 192 ถึง 223

ระดับที่8 ค่าสี มากกว่าหรือเท่ากับ 224 ถึง 255




**รูปที่ 3.5 รูปตัวอย่างการแบ่งกราฟแสดงความถี่ความเข้มสี (Color Histrogram) ออกเป็นบิน(Bin)**

โดยการตั้งเงื่อนไขแบบนี้จะเป็นการลดขนาดของฐานข้อมูลลงมา จากที่ต้องเก็บค่าตัวเลขถึง 768 ลดลงมาเหลือ 24 ค่า ทั้งนี้เพื่อที่โปรแกรมจะทำงานได้รวดเร็วขึ้น

การแปลงค่าความถี่ของความเข้มสีเป็นตัวเลขลงในฐานของมูลทำได้โดย

ค่าความถี่ในแต่ละช่วง / จำนวนพิกเซลของรูป = ค่าที่เก็บในฐานข้อมูล

การแปลงค่าถี่แบบนี้จะทำให้คุณสมบัติของรูปที่เหมือนกันแต่ขนาดต่างกัน สามารถเก็บค่าตัวเลขที่เหมือนกันได้ ตัวอย่างเช่น รูปขนาด 300 x 410 (123000) พิกเซล ย่อรูปให้เหลือ ขนาด 150 x 205 (30750) พิกเซล มีค่าความเข้มระดับที่ 1 เท่ากับ 60700 และ 15000 ตามลำดับ ค่าที่เก็บในฐานข้อมูล คือ 0.49349 และ 0.48781 เมื่อนำค่าความถี่ที่ทำแปลงเป็นสัดส่วนแล้วมาเปลี่ยนเทียบกันจะได้ค่าที่ใกล้เคียงกว่านำค่าความถี่มาเปรียบเทียบกันโดยตรง

| path | vr1 | vr2 | vr3 | vr4 | vr5 | vr6 | vr7 | vr8 | vg1 | vg2 | vg3 | vg4 | vg5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C:\Bitmaps\Cars\CARAVAN.JPG | 4.84991 | 1.86375 | 2.95096 | 2.22525 | 1.35869 | 1.06569 | 2.78969 | 2.89005 | 4.44628 | 2.56421 | 4.50148 | 2.24886 | 1.20449 |

**รูปที่ 3.6 ตัวอย่างการเก็บค่าความถี่ของสีลงในฐานข้อมูล**

ในส่วนของพื้นผิวนั้นเมื่อทำการผ่านค่าข้อมูลของรูปภาพเข้าไปในฟังก์ชั่นสมการกาบอร์เวฟเลต (Gabor Wavelet) จากนั้นวิเคราะห์ข้อมูลภาพและทำงานตามฟังก์ชั่นกาบอร์เวฟเลตที่ทำการเขียนโปรแกรมขึ้นซึ่งจะทำงานวิเคราะห์ข้อมูลภาพให้ได้ข้อมูลค่าของพื้นผิวของรูปภาพเป็นตัวเลข จากสมการ

$$G_{mn}(x,y) = \sum_{x}\sum_{t} I(x-s, y-t)\psi_{mh}(s,t) \qquad (3.1)$$

ฟังก์ชั่นกาบอร์เวฟเลต (Gabor Wavelet) ที่ใช้เขียนโปรแกรมเพื่อหาค่าพื้นผิวของรูป

โดยเมื่อได้ $G_{mn}(x,y)$ แล้ว จะทำการหาค่าเฉลี่ย ( $\mu_{mn}$ ) และส่วนเบี่ยงเบนมาตรฐาน ($\sigma_{mn}$) จากสมการ

$$E(m,n) = \sum_{x} \sum_{y} |G_{mn}(x,y)|, m = 0,1,...,M-1; n = 0,1,...,N-1 \qquad (3.2)$$

$$\mu_{mn} = \frac{E(m,n)}{PxQ} \qquad (3.3)$$

$$\sigma_{mn} = \frac{\sqrt{\sum_{x}\sum_{y}(|G_{mn}(x,y)| - \mu_{mn})^2}}{PxQ} \qquad (3.4)$$

ส่วนการเก็บค่าลงฐานข้อมูลนั้น จะเก็บค่าเฉลี่ย (Mean) และส่วนเบี่ยงเบนมาตรฐาน (Standard -Divation) โดยเก็บลงในเมทริกซ์ขนาด 3x4 ก่อนแล้วนำค่าในนั้นไปเก็บในฐานข้อมูลอีกทีซึ่งจะได้ข้อมูลเป็น mn00,sd00,mn01,sd01,....,mn23,sd23 ตามตัวอย่าง

| sd00 | mn01 | sd01 | mn02 | sd02 | mn03 | sd03 | mn10 | sd10 | mn11 | sd11 | mn12 | sd12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 14.29551 | 8.48266 | 12.12410 | 4.95343 | 7.30791 | 3.99890 | 4.69834 | 3.75142 | 5.46070 | 1.37716 | 3.08094 | 0.88928 | 1.98114 |

รูปที่ 3.7 ตัวอย่างการเก็บค่าความถี่ของพื้นผิวลงในฐานข้อมูล

### 3.1.3 เพิ่มข้อมูลของภาพลงในฐานข้อมูล

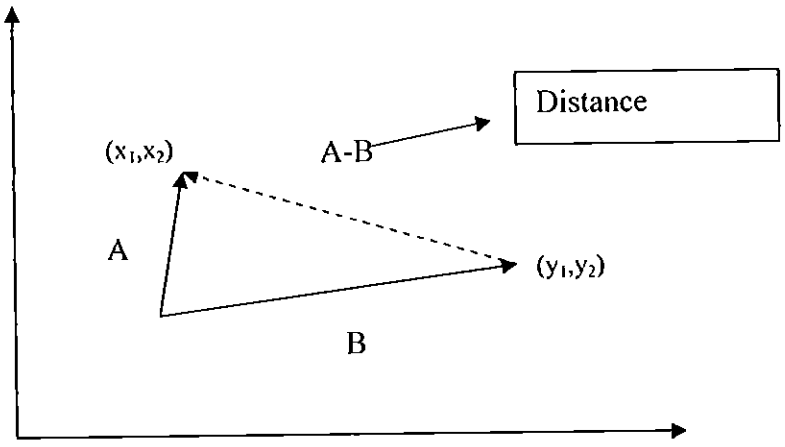
นำข้อมูลของภาพต้นแบบที่ผ่านการวิเคราะห์ให้เป็นเวคเตอร์แทนสีและพื้นผิว ไปเก็บในฐานข้อมูล MySQL เพื่อที่จะใช้ในการเปรียบเทียบหาภาพที่คล้ายคลึงต่อไป

### 3.1.4 เปรียบเทียบค่าในฐานข้อมูล

จะนำค่าที่ได้จากการวิเคราะห์จากภาพต้นแบบ ไปเปรียบเทียบกับค่าในฐานข้อมูล เพื่อหาภาพที่คล้ายกับภาพต้นแบบ ซึ่งมีวิธีหาความคล้ายคลึงของภาพ ดังนี้

**การหาความคล้ายคลึงของภาพ**

เมื่อต้องการหาความคล้ายคลึงของภาพต้นแบบกับภาพในฐานข้อมูล ต้องนำภาพต้นแบบไปผ่านอัลกอริทึมเช่นเดียวกับการเพิ่มข้อมูลลงในฐานข้อมูล เสร็จแล้วนำข้อมูลนั้น มาเปรียบเทียบกับข้อมูลในฐานข้อมูลโดยการคำนวณหานั้นจะทำการหาค่าระยะห่าง (Distance) ระหว่างภาพต้นแบบกับภาพในฐานข้อมูล แล้วนำมาเปรียบเทียบกับข้อมูลของภาพทุกๆภาพในฐานข้อมูล จากนั้นนำค่าระยะห่างที่น้อยที่สุดมาแสดงผล โดยค่าฮิสโตรแกรม (Histogram) ที่ถูกเก็บอยู่ในรูปเวคเตอร์จะทำการเปรียบเทียบได้ดังนี้




รูปที่ 3.8 รูปการหาค่าระยะห่าง(Distance)

จากสูตรการหาค่า Vector

$$D(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots} \qquad (3.5)$$

เมื่อทำการเปรียบเทียบรูปภาพทุกรูปภาพในฐานข้อมูลแล้ว จะได้ค่าระยะห่าง(Distance)ของแต่ละรูปภาพออกมา โดยค่าระยะห่างที่ได้ออกมามีความแตกต่างกันออกไปจึงทำการจัดเรียงลำดับจากน้อยไปหามาก ซึ่งค่าระยะห่างที่มีค่าน้อยที่สุดคือค่าของรูปภาพในฐานข้อมูลที่มีความใกล้เคียงมากที่สุด โดยโปรแกรมจะทำการนำรูปภาพที่มีค่าระยะห่างน้อยที่สุด 20 อันดับแรกขึ้นไปแสดงผล

**3.1.5 แสดงภาพที่ใกล้เคียงกับภาพต้นแบบ**

เมื่อภาพที่ใกล้เคียงกับภาพต้นแบบ 20 ภาพแล้ว จะนำมาแสดงในโปรแกรม ตามลำดับจากคล้ายมากไปหาน้อยลงมาเรื่อยๆ แต่ภาพที่ได้ ยังไม่ตรงตามความต้องการมากนัก

**3.1.6 ระบบป้อนกลับ(Relevance Feedback)**

เป็นขั้นตอนสุดท้ายในการค้นหาภาพคือเมื่อโปรแกรมได้หาค่าสี (color) และพื้นผิว (texture) ของภาพต้นแบบแล้ว อินพุต (Input) ตัวต่อไปของโปรแกรมคือค่าสีและพื้นผิวที่ผู้ใช้ได้เลือกจาก ภาพที่โปรแกรมได้แสดงขึ้นมาโดยให้ผู้ใช้ทำการป้อนกลับ (Relevance Feedback) ลักษณะของรูปภาพที่ผู้ใช้ต้องการสอนเพื่อให้โปรแกรมรู้ถึงความต้องการของผู้ใช้ ซึ่งหลักการในการทำระบบป้อนกลับ คือ การนำของภาพที่ผู้ใช้เลือกมาคำนวณให้ได้ ค่าเรียงลำดับ(Query) ตัวใหม่ แล้วนำค่านี้ไปเปรียบเทียบ กับเว็ดเตอร์ ในฐานข้อมูลโดยการคำนวณตามสูตรดังนี้

$$Z = Z' + \alpha_R(\bar{X}' - Z') - \alpha_N(\bar{X}'' - Z') \quad (3.6)$$

$\alpha_N, \alpha_R$ = ค่าคงที่ค่าหนึ่ง

$\bar{x}'$ = ค่าเฉลี่ยของกลุ่มภาพที่เราเลือก

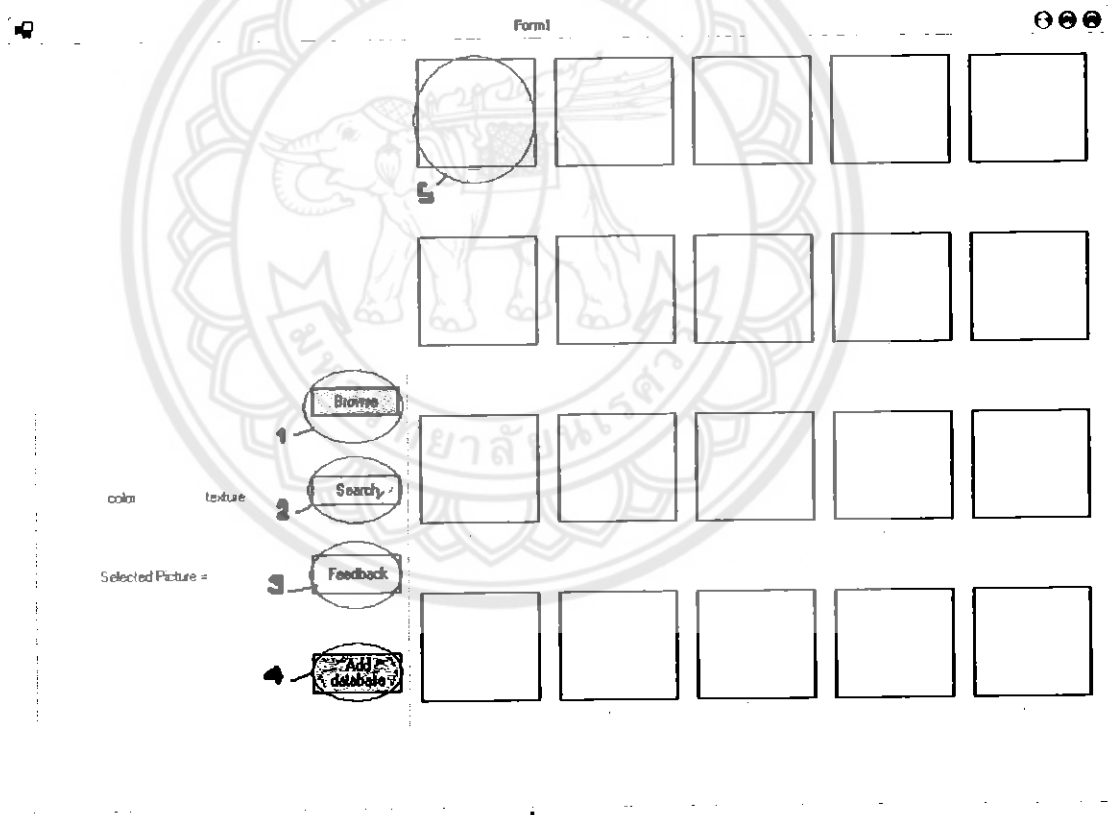
$\bar{x}''$ = ค่าเฉลี่ยของกลุ่มภาพที่เราไม่ได้เลือก

$Z'$ = ค่าดัชนี(Index) ของภาพต้นแบบ

$Z$ = ค่าดัชนี(Index) ค่าใหม่ที่จะนำไปใช้ค้นหาในครั้งต่อไป

### 3.1.7 ได้ภาพตรงกับความต้องการมากขึ้น

หลังจากที่ได้ทำการป้อนกลับ (Relevance Feedback)แล้ว เราจะได้ภาพที่ตรงกับความต้องการมากขึ้น ซึ่งเราสามารถทำการป้อนกลับได้อีกเพื่อความแม่นยำยิ่งขึ้น




**รูปที่ 3.9 ตัวอย่างโปรแกรม**

ในตัวโปรแกรมจะมีส่วนหลักๆ อยู่ 5 ส่วน คือ

1. ส่วนที่ใช้เลือกรูปภาพต้นแบบ โดยปุ่ม "Browse" จะเป็นการเลือกภาพที่เราต้องการนำมาเป็นภาพต้นแบบ เพื่อใช้หาภาพที่คล้ายๆกันนี้

2. ส่วนที่ใช้ในการค้นหาภาพ ซึ่งเราสามารถเลือกได้ว่าจะเน้นภาพที่คล้ายกับสี(color) ของภาพต้นแบบ หรือ คล้ายกับพื้นผิว(texture) หรือจะใช้ทั้ง 2 อย่างช่วยในการค้นหา ซึ่งปุ่ม "Search" จะมีอัลกอริทึมคำนวณเพื่อวิเคราะห์ สีและพื้นผิวเช่นเดียวกับการเก็บภาพลงในฐานข้อมูล โดยเมื่อคำนวณได้ค่าเป็นเวคเตอร์แล้ว จะนำไปเปรียบเทียบกับภาพในฐานข้อมูล แล้วนำภาพที่ใกล้เคียงกับภาพต้นแบบ มาแสดงผล

3. ส่วนนี้จะใช้ในการทำการป้อนกลับ (Relevance Feedback)เพื่อให้โปรแกรมเรียนรู้ภาพที่ได้เลือกไว้ แล้วนำไปคำนวณเพื่อให้ได้ค่านำมาค้นหาใหม่ และนำไปเปรียบเทียบกับฐานข้อมูลอีกครั้ง ซึ่งจะได้ภาพที่ใกล้เคียงกับภาพต้นแบบมากขึ้น

4. ส่วนที่ใช้ในการเพิ่มข้อมูล ลงในฐานข้อมูล โดยปุ่ม "Add Database" จะมีไว้สำหรับเพิ่มข้อมูลรูปภาพลงในฐานข้อมูล ซึ่งนำรูปที่ Browse เข้ามา ผ่านอัลกอริทึมเพื่อวิเคราะห์สี และ พื้นผิว ออกมา ให้อยู่ในรูป เวคเตอร์แล้วเก็บ เวคเตอร์นั้น ลงในฐานข้อมูล

5. ส่วนที่ใช้แสดงผลของการหาภาพ ซึ่งจะนำภาพที่คล้ายคลึงกับภาพต้นแบบมากที่สุด 20 ภาพ พร้อมทั้งมี "box" ให้เลือกภาพนั้นๆ เพื่อนำภาพนั้นมาทำการป้อนกลับ (Relevance - Feedback) เพื่อให้ได้ภาพที่ตรงตามความต้องการมากขึ้น

## 3.2 ฐานข้อมูล (Database)

ในฐานข้อมูลจะเก็บข้อมูลที่ได้จากการวิเคราะห์ผ่านอัลกอริทึมของสีและพื้นผิว พร้อมทั้งเก็บ path ซึ่งเป็นที่อยู่ของภาพ ภาพนั้น โดยค่าสีจะถูกเก็บในฐานข้อมูลดังภาพตามนี้

| | |
|---|---|
| vr1 – vr8 | เก็บฮิสโทรแกรมของสีแดง |
| vg1 – vg8 | เก็บฮิสโทรแกรมของสีเขียว |
| vb1 – vb8 | เก็บฮิสโทรแกรมของสีน้ำเงิน |

14993421
ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร


| path | vr1 | vr2 | vr3 | vr4 | vr5 | vr6 | vr7 | vr8 | vg1 | vg2 | vg3 | vg4 | vg5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C:\Bitmaps\Cars\CARAVAN.JPG | 4.84991 | 1.86975 | 2.95096 | 2.22525 | 1.35869 | 1.06569 | 2.78969 | 2.69005 | 4.44628 | 2.56421 | 4.50148 | 2.24886 | 1.20449 |
| C:\Bitmaps\Cars\CHEROKEE.JPG | 8.74595 | 3.41350 | 2.47826 | 1.11154 | 0.64721 | 0.73899 | 2.17733 | 0.68722 | 1.56137 | 2.62462 | 2.56344 | 1.53897 | 0.80111 |
| C:\Bitmaps\Cars\CIVIC01.JPG | 8.61377 | 1.42496 | 0.74111 | 1.09095 | 1.51151 | 1.45537 | 2.78845 | 2.37397 | 5.52274 | 3.67699 | 1.56072 | 0.94620 | 0.22455 |
| C:\Bitmaps\Cars\EAGLE01.JPG | 8.92760 | 2.39718 | 2.54637 | 2.68810 | 1.18979 | 1.76320 | 0.28244 | 0.00532 | 1.59549 | 1.92779 | 1.40689 | 1.99560 | 2.69052 |
| C:\Bitmaps\Cars\ELDORAD1.JPG | 8.51583 | 0.75966 | 1.58573 | 2.57134 | 1.39268 | 1.09539 | 1.96063 | 2.11873 | 5.26244 | 2.28104 | 1.60146 | 2.69757 | 1.23181 |
| C:\Bitmaps\Cars\GMCVAN01.JPG | 7.90710 | 2.74210 | 1.40078 | 1.30447 | 2.02870 | 1.96583 | 2.21960 | 0.43142 | 5.47264 | 1.89214 | 1.14287 | 1.41111 | 2.03186 |
| C:\Bitmaps\Cars\GMCVAN02.JPG | 7.79524 | 2.30006 | 2.04263 | 1.44734 | 2.67963 | 1.68355 | 1.32328 | 0.52228 | 4.06674 | 1.94878 | 1.97975 | 1.46345 | 2.68991 |
| C:\Bitmaps\Cars\GMCVAN03.JPG | 8.29415 | 3.46435 | 2.46942 | 1.77628 | 1.10101 | 0.77867 | 1.21627 | 0.69985 | 4.85589 | 2.74513 | 2.33926 | 1.99627 | 1.09041 |
| C:\Bitmaps\Cars\GRANDAM.JPG | 7.35974 | 5.63457 | 2.53367 | 0.98498 | 0.69041 | 1.60998 | 0.99612 | 0.19053 | 2.57703 | 3.56722 | 3.19295 | 1.28307 | 0.77253 |
| C:\Bitmaps\Cars\INTEGRA.JPG | 8.03570 | 3.88308 | 3.21301 | 1.59738 | 1.14630 | 0.76475 | 0.56917 | 0.79061 | 2.45435 | 1.98701 | 3.34668 | 2.29015 | 1.64232 |
| C:\Bitmaps\Cars\JAGUAR03.JPG | 8.48043 | 3.27295 | 2.99784 | 1.83808 | 0.78375 | 0.53881 | 1.02268 | 1.00547 | 2.52909 | 2.80400 | 3.46747 | 2.75574 | 2.07404 |
| C:\Bitmaps\Cars\JIMMY01.JPG | 7.45576 | 2.90700 | 1.13155 | 1.15743 | 1.24087 | 1.98056 | 1.63327 | 2.49356 | 4.91753 | 2.84268 | 2.32399 | 1.39143 | 1.28683 |
| C:\Bitmaps\Cars\LEGEND02.JPG | 7.93181 | 2.20737 | 2.84590 | 1.75709 | 1.20683 | 0.87607 | 1.15335 | 2.02159 | 3.10363 | 1.77631 | 2.69888 | 2.47208 | 1.04381 |
| C:\Bitmaps\Cars\M8ZE320.JPG | 8.58683 | 4.50774 | 2.58286 | 1.19971 | 0.72353 | 0.54173 | 0.60912 | 1.24847 | 3.97781 | 3.42121 | 2.26191 | 1.62856 | 0.88243 |
| C:\Bitmaps\Cars\ODYSSY95.JPG | 7.14236 | 1.13362 | 1.87967 | 1.50910 | 1.57531 | 1.69420 | 2.46564 | 2.60010 | 5.93955 | 1.21169 | 1.06944 | 0.90529 | 1.03041 |
| C:\Bitmaps\Cars\PASSAT95.JPG | 6.88578 | 2.34398 | 3.64242 | 2.94966 | 1.53739 | 0.75473 | 0.52169 | 1.36435 | 4.08938 | 2.65181 | 3.19933 | 1.81500 | 0.96247 |
| C:\Bitmaps\Cars\SAAB9502.JPG | 11.0870 | 4.05362 | 1.58069 | 0.96346 | 0.63188 | 0.36830 | 0.55214 | 0.74286 | 8.37274 | 1.86333 | 1.70543 | 0.73242 | 0.36464 |
| C:\Bitmaps\Cars\SAAB9503.JPG | 8.44312 | 2.49481 | 2.67052 | 2.01298 | 1.22946 | 1.00670 | 1.04974 | 1.09267 | 8.24211 | 1.65993 | 0.64684 | 0.85629 | 1.36635 |

**รูปที่ 3.10 รูปฐานข้อมูลสี**

ส่วนการเก็บค่าพื้นผิวลงฐานข้อมูลนั้น จะเก็บค่าเฉลี่ย (Mean) และส่วนเบี่ยงเบนมาตรฐาน (Standard -Divation) โดยเก็บลงในเมทริกซ์ขนาด 3x4 ก่อนแล้วนำค่าในนั้นไปเก็บในฐานข้อมูลอีกที ซึ่งจะได้ข้อมูลเป็น mn00,sd00,mn01,sd01,....,mn23,sd23 ตามตัวอย่าง

| vb2 | vb3 | vb4 | vb5 | vb6 | vb7 | vb8 | mn00 | sd00 | mn01 | sd01 | mn02 | sd02 | mn03 | sd03 | mn10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2.09375 | 3.00148 | 2.41537 | 1.77441 | 0.92062 | 0.56478 | 0.11314 | 15.95281 | 14.29551 | 8.48256 | 12.1241 | 4.95343 | 7.30791 | 3.99880 | 4.69834 | 3.75142 |
| 2.32464 | 2.48682 | 1.54302 | 1.34243 | 1.10680 | 1.39891 | 0.82097 | 13.32362 | 16.60506 | 8.19077 | 10.9173 | 6.59777 | 10.97802 | 6.15886 | 9.78754 | 2.89361 |
| 5.44218 | 1.64783 | 0.80202 | 0.21978 | 0.15784 | 0.16224 | 0.06699 | 10.11564 | 9.27845 | 8.76444 | 13.8787 | 5.70185 | 10.80510 | 7.40001 | 9.58732 | 2.36530 |
| 2.41367 | 1.21325 | 0.94584 | 1.30746 | 3.06186 | 2.86996 | 0.02612 | 11.99417 | 13.98952 | 7.38522 | 10.4670 | 3.79352 | 6.00302 | 7.70210 | 7.74036 | 2.37403 |
| 1.48499 | 1.44850 | 2.10804 | 1.26049 | 0.53495 | 0.35115 | 0.12181 | 15.81097 | 14.47000 | 8.50247 | 12.2607 | 5.68883 | 8.50391 | 3.90832 | 5.06152 | 4.44036 |
| 2.84472 | 1.29608 | 1.25973 | 1.99781 | 1.36053 | 2.62269 | 0.43276 | 24.70832 | 23.55334 | 10.2327 | 11.3106 | 9.32778 | 11.14069 | 8.05157 | 9.25905 | 5.22042 |
| 2.68981 | 1.95709 | 1.91667 | 2.55816 | 2.65760 | 1.92052 | 0.16869 | 32.62720 | 27.47511 | 12.6669 | 13.3948 | 8.72666 | 11.07763 | 5.84459 | 6.39537 | 6.32316 |
| 3.16944 | 2.19562 | 2.05362 | 1.18903 | 0.66676 | 0.75029 | 1.65099 | 29.12274 | 28.38665 | 10.7591 | 17.0182 | 6.14420 | 8.86476 | 4.36202 | 5.55505 | 5.70980 |
| 3.92563 | 3.25703 | 1.29167 | 0.75375 | 0.76332 | 1.99625 | 0.41146 | 17.83281 | 17.45222 | 7.84826 | 9.35125 | 4.32704 | 7.07803 | 5.25856 | 7.66968 | 3.84380 |
| 2.48514 | 2.60183 | 2.37746 | 1.02107 | 1.17287 | 1.53394 | 1.56882 | 24.56033 | 24.29716 | 11.0537 | 16.6876 | 7.20737 | 13.61872 | 7.23873 | 12.49339 | 3.95285 |
| 2.75783 | 2.41482 | 1.97185 | 1.47077 | 1.49885 | 1.82735 | 2.71274 | 24.35678 | 25.83965 | 10.2456 | 14.4130 | 9.34019 | 14.05981 | 5.82670 | 6.62990 | 5.27186 |
| 3.10631 | 1.98582 | 1.38778 | 1.36555 | 1.27620 | 0.56186 | 0.01224 | 28.07588 | 26.83946 | 8.84832 | 13.9448 | 6.03987 | 9.48797 | 5.27769 | 6.81795 | 5.87051 |
| 1.07571 | 2.43965 | 2.10674 | 1.64401 | 1.10095 | 1.33913 | 2.17547 | 18.27345 | 18.00563 | 13.9657 | 20.7565 | 11.51504 | 14.25684 | 8.29274 | 9.81867 | 5.17116 |
| 3.85586 | 2.27697 | 1.57003 | 1.08439 | 0.80680 | 0.87510 | 0.72516 | 23.77145 | 25.60838 | 7.15144 | 8.76687 | 5.77953 | 10.19501 | 3.87347 | 5.46618 | 5.80072 |
| 1.17409 | 0.98803 | 1.02035 | 1.24725 | 1.77886 | 2.84183 | 0.14811 | 21.55296 | 24.46654 | 9.64955 | 14.7132 | 6.96067 | 8.44133 | 4.90465 | 7.14143 | 5.06513 |
| 2.35936 | 2.08991 | 3.11159 | 1.38427 | 0.86251 | 0.94650 | 0.56986 | 18.13879 | 21.85968 | 8.27298 | 12.3763 | 5.20230 | 9.15920 | 3.84449 | 6.65924 | 2.91332 |
| 1.98451 | 1.46570 | 1.05954 | 0.52977 | 0.35158 | 0.47040 | 0.61151 | 18.77834 | 15.88230 | 9.66572 | 11.3589 | 12.05419 | 16.63369 | 11.45905 | 10.92768 | 3.74914 |
| 1.98962 | 0.85748 | 0.98562 | 0.80266 | 0.65398 | 0.79055 | 0.72383 | 24.70425 | 25.88530 | 4.97606 | 7.14992 | 3.60139 | 5.55065 | 2.77887 | 3.18633 | 6.13036 |
| 0.91735 | 3.05345 | 1.85594 | 1.17338 | 1.09797 | 1.04657 | 1.95234 | 16.88476 | 17.27440 | 8.46609 | 15.0477 | 6.52255 | 9.12927 | 6.39570 | 8.23538 | 3.43419 |

**รูปที่ 3.11 รูปฐานข้อมูลพื้นผิว**

# บทที่ 4
# ผลการทดลอง

จากการที่ได้ทำการทดลองใช้งานโปรแกรมค้นหาภาพจากฐานข้อมูลภาพแล้ว เราจะทำการทดสอบผลการทดลองโดยจะใช้หลักการเปรียบเทียบผลของค่าความถูกต้อง (Precision) โดยค่าความถูกต้องเป็นค่าที่ได้จากการคำนวณดังต่อไปนี้

% ความถูกต้อง = (จำนวนภาพที่ใกล้เคียง / จำนวนภาพทั้งหมด) × 100

เช่น % ความถูกต้อง = $\left(\frac{4}{20}\right) \times 100 = 20\%$

แล้วนำค่าความถูกต้อง (Precision) มารวมกันแล้วหาค่าเฉลี่ย ก็จะได้เป็นค่าความถูกต้องของวิธี (Method) นั้น ๆ ซึ่งเมื่อนำไปเปรียบเทียบกับวิธีการอื่นๆ ก็จะเห็นความแตกต่างของการค้นหาได้จากการสร้างกราฟ

โดยเราจะทำการเปรียบเทียบ การค้นหาภาพดังนี้

1. การค้นหาภาพโดยวิเคราะห์ด้วยสี (Color)
2. การค้นหาภาพโดยวิเคราะห์ด้วยพื้นผิว(Texture)
3. การค้นหาภาพโดยวิเคราะห์ด้วยสี(Color), พื้นผิว(Texture) พร้อมกัน
4. การค้นหาภาพโดยวิเคราะห์ด้วยสี(Color),พื้นผิว(texture)พร้อมด้วยการการป้อนกลับ(Feedback)

ซึ่งในการเปรียบเทียบแต่ละครั้งจะเปรียบเทียบกับภาพที่เก็บไว้ในฐานข้อมูล MySQL ซึ่งมีข้อมูลภาพทั้งหมดกว่า 1,000 ภาพ

## 4.1 การทดลองค้นหาภาพโดยวิเคราะห์ด้วยสี (Color)

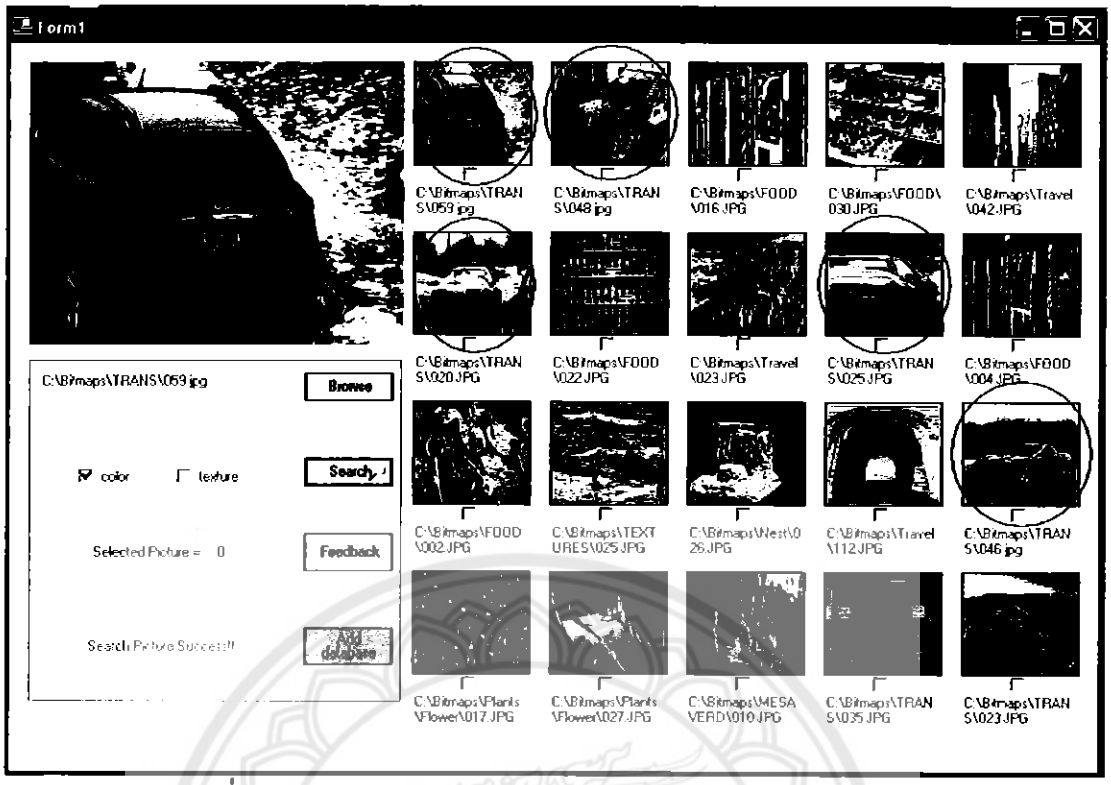
4.1.1 นำไฟล์รูปภาพตัวอย่างที่ต้องการค้นหา โดยวิเคราะห์แต่สีเท่านั้น

4.1.2 โปรแกรมจะทำการเปรียบเทียบค่าเวคเตอร์ระหว่าง ไฟล์รูปภาพ ในฐานข้อมูล

4.1.3 เมื่อเปรียบเทียบค่า Distance $d1 < d2 < d3 < d4 < d5 < d6 < d7 < d8 < d9 <.....< d16$

4.1.4 โชว์ผลการค้นหา ทั้งหมด 20 PC โดยเรียงลำดับค่าระยะห่าง (Distance) จาก PC1, PC2, PC3 ไปจนถึง PC16

4.1.5 ทำการเลือกภาพที่เหมือน โดยการทดสอบ ในกรณีนี้เราจะพิจารณาองค์ประกอบของภาพว่ามีส่วนที่เกี่ยวข้องภายในภาพหรือไม่




**รูปที่ 4.1** แสดงการค้นหาจากฐานข้อมูล โดยการวิเคราะห์จากสี (Color)

**เมื่อทำการทดลองเสร็จสิ้นจะทำการคำนวณค่าความถูกต้อง (Precision) จะได้ดังนี้**

**จากตัวอย่างในรูปที่ จะได้ภาพที่คล้ายคลึง 4 ภาพ นำมาคำนวณหาค่าความถูกต้องจะได้ว่า**

$$\%\text{Precision} = \left(\frac{4}{20}\right) \times 100 \quad = 20\ \%$$

**ฉะนั้นเราจะนำค่าความถูกต้อง (Precision) ในแต่ละไฟล์รูปภาพ มาบันทึกผลการทดลองได้ดังนี้**

**ตารางที่ 4.1 บันทึกค่าการทดลองการค้นหาด้วยสี (Color)**

| ครั้งที่ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| ค่า precision | 20 | 35 | 40 | 30 | 45 | 25 | 40 | 35 | 30 | 30 |

**จากการทดลองหาภาพทั้งหมด 10 ภาพ ได้ค่าความถูกต้อง (Precision) อยู่ในช่วง 20-45 ซึ่งสรุปได้ว่าค่าเฉลี่ยของค่าความถูกต้องสำหรับการหาด้วยสี (Color) คือ 33 %**

## 4.2 การทดลองค้นหาภาพโดยวิเคราะห์ด้วยพื้นผิว(Texture)

4.2.1 นำไฟล์รูปภาพตัวอย่างที่ต้องการค้นหา โดยวิเคราะห์แต่พื้นผิวเท่านั้น

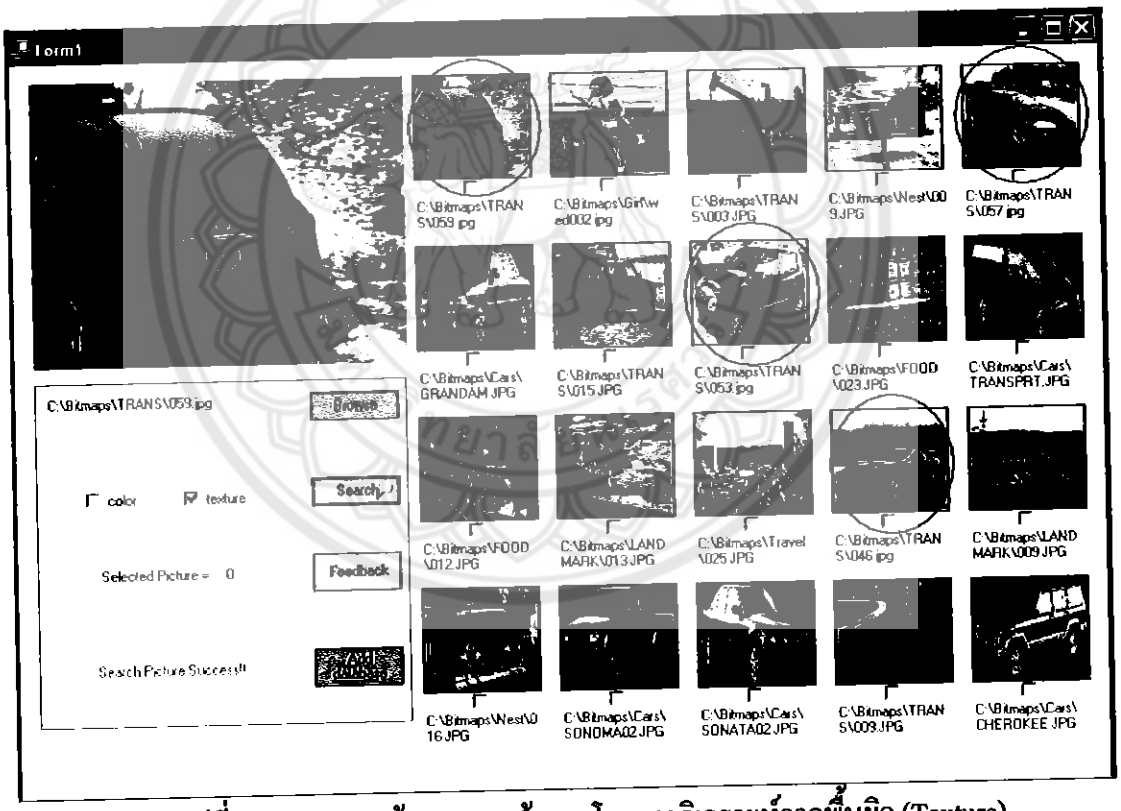
4.2.2 โปรแกรมจะทำการเปรียบเทียบค่าเวคเตอร์ระหว่าง ไฟล์รูปภาพ ในฐานข้อมูล

4.2.3 เมื่อเปรียบเทียบค่า Distance $d1 < d2 < d3 < d4 < d5 < d6 < d7 < d8 < d9 <.....< d16$

4.2.4 โชว์ผลการค้นหา ทั้งหมด 20 PC โดยเรียงลำดับค่าระยะห่าง (Distance) จาก PC1, PC2, PC3 ไปจนถึง PC20

4.2.5 ทำการเลือกภาพที่เหมือน โดยการทดสอบ ในกรณีนี้เราจะพิจารณาองค์ประกอบของภาพว่ามีส่วนที่เกี่ยวข้องภายในภาพหรือไม่

เมื่อทำการทดลองเสร็จสิ้นจะทำการคำนวณค่าความถูกต้อง (Precision) จะได้ดังนี้




รูปที่ 4.2 แสดงการค้นจากฐานข้อมูล โดยการวิเคราะห์จากพื้นผิว (Texture)

เมื่อทำการทดลองเสร็จสิ้นจะทำการคำนวณค่าความถูกต้อง (Precision) จะได้ดังนี้

จากตัวอย่างในรูปที่ จะได้ภาพที่คล้ายคลึง 4 ภาพ นำมาคำนวณหาค่าความถูกต้องจะได้ว่า

$$\%\text{Precision} = \left(\frac{4}{20}\right)\times 100 \quad = 20\%$$

ฉะนั้นเราจะนำค่าความถูกต้อง (Precision) ในแต่ละไฟล์รูปภาพ มาบันทึกผลการทดลองได้ดังนี้

**ตารางที่ 4.2** บันทึกค่าการทดลองการค้นหาด้วยพื้นผิว (Texture)

| ครั้งที่ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| ค่า precision | 20 | 25 | 15 | 10 | 15 | 20 | 20 | 15 | 20 | 20 |

จากการทดลองหาภาพทั้งหมด 10 ภาพ ได้ค่าความถูกต้อง (Precision) อยู่ในช่วง 10-25 ซึ่งสรุปได้ว่าค่าเฉลี่ยของค่าความถูกต้องสำหรับการหาด้วยพื้นผิว (Texture) คือ 18 %

## 4.3 การทดลองค้นหาภาพโดยวิเคราะห์ด้วย สี(Color), พื้นผิว(Texture) พร้อมกัน

4.3.1 นำไฟล์รูปภาพตัวอย่างที่ต้องการค้นหา โดยวิเคราะห์ สี(Color) และ พื้นผิว(Texture)

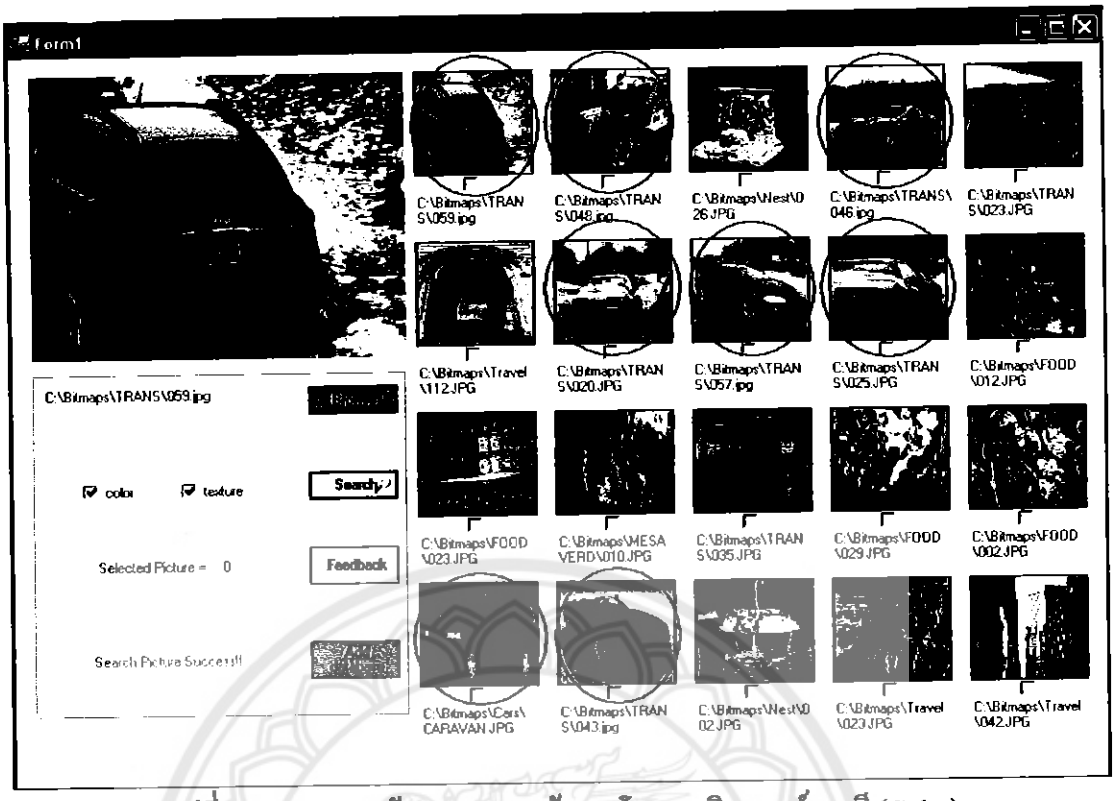
4.3.2 โปรแกรมจะทำการเปรียบเทียบค่าเวคเตอร์ระหว่าง ไฟล์รูปภาพ ในฐานข้อมูล

4.3.3 เมื่อเปรียบเทียบค่า Distance $d1 < d2 < d3 < d4 < d5 < d6 < d7 < d8 < d9 <.....< d16$

4.3.4 โชว์ผลการค้นหา ทั้งหมด 20 PC โดยเรียงลำดับค่าระยะห่าง (Distance) จาก PC1, PC2, PC3 ไปจนถึง PC20

4.3.5 ทำการเลือกภาพที่เหมือน โดยการทดสอบ ในกรณีนี้เราจะพิจารณาองค์ประกอบของภาพว่ามีส่วนที่เกี่ยวข้องภายในภาพหรือไม่

เมื่อทำการทดลองเสร็จสิ้นจะทำการคำนวณค่าความถูกต้อง (Precision) จะได้ดังนี้




**รูปที่ 4.3 แสดงการค้นหาจากฐานข้อมูล โดยการวิเคราะห์จากสี (Color) และพื้นผิว (Texture) รวมกัน**

**จากตัวอย่างในรูปที่ จะได้ภาพที่คล้ายคลึง 8 ภาพ นำมาคำนวณหาค่าความถูกต้อง (Precision) จะได้ว่า**

$$\%\text{Precision} = \left(\frac{8}{20}\right) \times 100 \quad = 40\%$$

**ฉะนั้นเราจะนำค่าความถูกต้อง (Precision) ในแต่ละไฟล์รูปภาพ มาบันทึกผลการทดลองได้ดังนี้**

**ตารางที่ 4.3 บันทึกค่าการทดลองการค้นหาด้วยสีและพื้นผิว (Color&Texture)**

| การทดลองครั้งที่ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| ค่า precision | 40 | 60 | 65 | 55 | 50 | 65 | 45 | 55 | 50 | 45 |

**จากการทดลองหาภาพทั้งหมด 10 ภาพ ได้ค่าความถูกต้อง (Precision) อยู่ในช่วง 40-65 ซึ่งสรุปได้ว่าค่าเฉลี่ยของค่าความถูกต้องสำหรับการหาด้วยสี (Color) และ พื้นผิว(Texture) คือ 53%**

**4.4 การทดลองค้นหาภาพโดยวิเคราะห์ด้วยสี (Color) และพื้นผิว (Texture) พร้อมด้วยการป้อนกลับ (Relevance Feedback)**

4.4.1 นำไฟล์รูปภาพตัวอย่างที่ต้องการค้นหา โดยวิเคราะห์สี(Color)และพื้นผิว(texture) พร้อมด้วยการป้อนกลับ (Feedback)

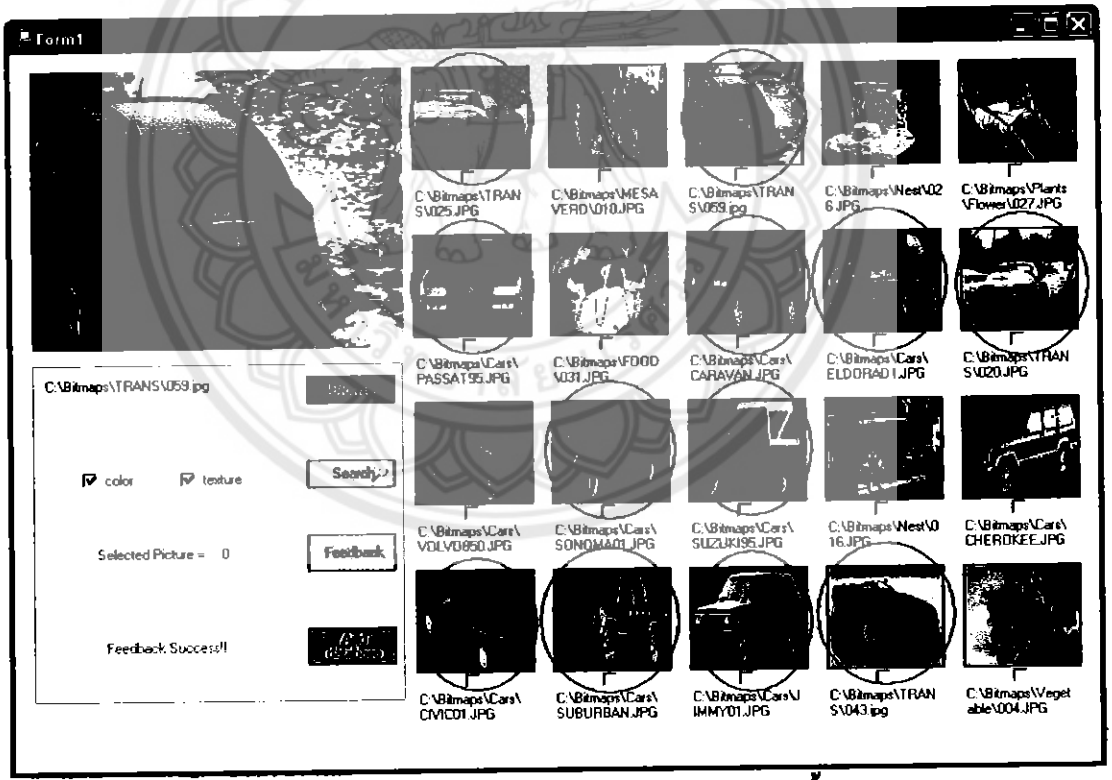
4.4.2 โปรแกรมจะทำการเปรียบเทียบค่าเวคเตอร์ระหว่าง ไฟล์รูปภาพ ในฐานข้อมูล

4.4.3 เมื่อเปรียบเทียบค่า Distance $d1 < d2 < d3 < d4 < d5 < d6 < d7 < d8 < d9 <....< d16$

4.4.4 โชว์ผลการค้นหา ทั้งหมด 20 PC โดยเรียงลำดับค่าระยะห่าง (Distance) จาก PC1, PC2, PC3 ไปจนถึง PC20

4.4.5 ทำการเลือกภาพที่เหมือน โดยการทดสอบ ในกรณีนี้เราจะพิจารณาองค์ประกอบของภาพว่ามีส่วนที่เกี่ยวข้องภายในภาพหรือไม่

เมื่อทำการทดลองเสร็จสิ้นจะทำการคำนวณค่าความถูกต้อง (Precision) จะได้ดังนี้




**รูปที่ 4.4 แสดงการค้นหาจากฐานข้อมูลโดยการวิเคราะห์จากสีและพื้นผิว (Color&Texture) พร้อมกระบวนการป้อนกลับ (Relevance Feedback)**

จากตัวอย่างในรูปที่ จะได้ภาพที่คล้ายคลึง 12 ภาพ นำมาคำนวณหาค่าความถูกต้อง (Precision) จะได้ว่า

$$\%\text{Precision} = \left(\frac{12}{20}\right) \times 100 \quad = 60\ \%$$

ฉะนั้นเราจะนำค่าความถูกต้อง (Precision) ในแต่ละไฟล์รูปภาพ มาบันทึกผลการทดลองได้ดังนี้

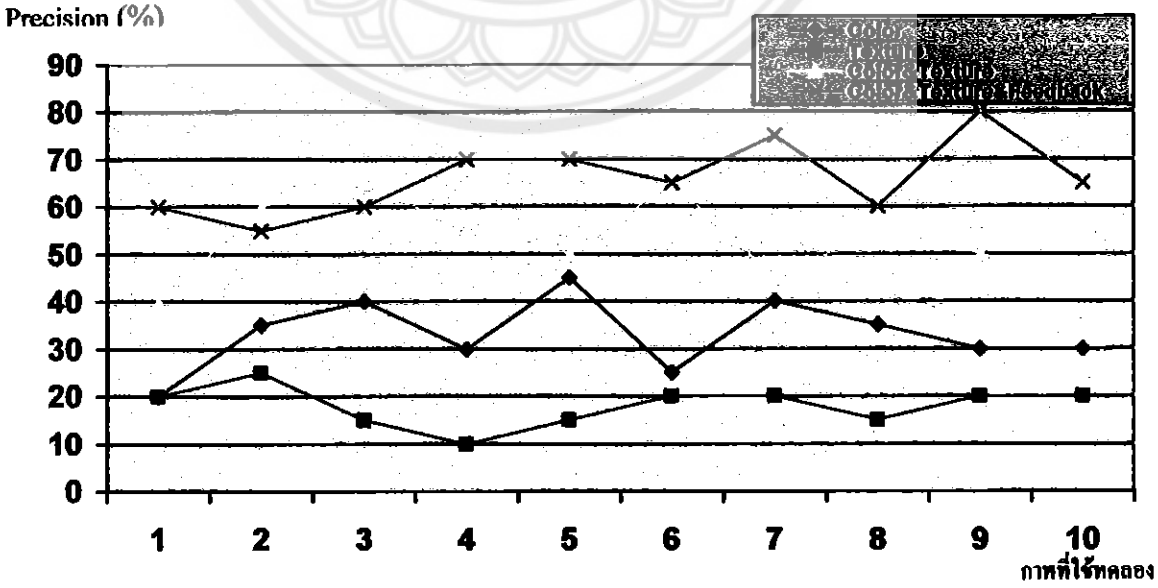
**ตารางที่ 4.4 บันทึกค่าการทดลองการค้นหาด้วยสีและพื้นผิว (Color&Texture) พร้อมทั้งกระบวนการ ป้อนกลับ(Relevance Feedback)**

| ครั้งที่ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| ค่า precision | 60 | 55 | 60 | 70 | 70 | 65 | 75 | 60 | 80 | 65 |

จากการทดลองหาภาพทั้งหมด 10 ภาพ ได้ค่าความถูกต้อง (Precision) อยู่ในช่วง 40-65 ซึ่งสรุปได้ว่าค่าเฉลี่ยของ ค่าความถูกต้องสำหรับการหาสีและพื้นผิว (Color&Texture) และการป้อนกลับ (Relevance Feedback) คือ 66 %

## 4.5 เปรียบเทียบผลการทดลอง

นำผลการทดลองที่หาค่าความถูกต้อง (Precision) ของภาพแต่ละวิธีมาพล็อตกราฟ เปรียบเทียบการค้นหาทั้ง 3 วิธี จะเห็นความแตกต่างของการทดลองโดยที่เส้นกราฟที่เกิดจากการค้นหาแบบ ดังรูปที่ 4.5



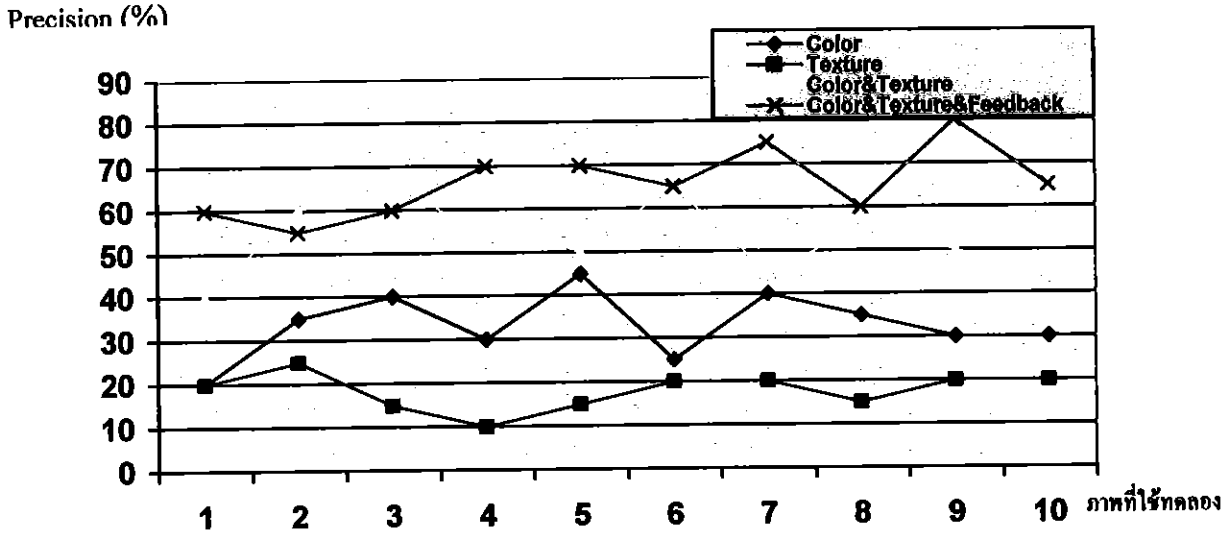

**รูปที่ 4.5 กราฟเปรียบเทียบการทำงานของแต่ละวิธี**

# บทที่ 5
# บทสรุป

## 5.1 สรุปผลการทดลอง

เราจะได้ผลการทดลองดังตารางที่ 4.1,4.2 4.3 และ4.4 เมื่อเปรียบเทียบค่าความแตกต่างระหว่างการค้นหาทั้ง 3 วิธี ทำให้เราสามารถสรุปได้ว่า

- การค้นหารูปภาพจากฐานข้อมูลนั้น โปรแกรมจะทำการหาภาพที่เพิ่มเข้ามาในฐานข้อมูลแล้วแสดงผลภาพที่ผ่านกระบวนการค้นหาออกมาแล้ว ซึ่งรวมถึงภาพต้นแบบที่ทำการค้นหาด้วย
- การค้นหารูปภาพโดยวิเคราะห์ด้วยสี (Color) จากตาราง 4.1 และ รูปที่ 4.1 จะพบว่าการค้นหาออกมาได้ผลคล้ายภาพต้นแบบในเฉพาะเรื่องสี แต่ในส่วนของพื้นผิว หรือความคล้ายของรูปยังไม่ชัดเจนเท่าที่ควร
- การค้นหารูปภาพโดยวิเคราะห์ด้วยพื้นผิว (Texture) จากตาราง 4.2 และ รูปที่ 4.2 จะพบว่าการค้นหาออกมาได้ผลคล้ายภาพต้นแบบในเฉพาะเรื่องพื้นผิว แต่ในส่วนของสี หรือความคล้ายของรูปยังไม่ชัดเจนเท่าที่ควร
- การค้นหารูปภาพโดยวิเคราะห์ด้วยสีและพื้นผิว (Color &Texture) จากตาราง 4.3 และ รูปที่ 4.3 จะพบว่าการค้นหาออกมาได้ผลคล้ายภาพต้นแบบ ทั้งในส่วนของสี พื้นผิว และความคล้ายของรูป ซึ่งมีความคล้ายคลึงกับรูปตัวอย่าง
- การค้นหารูปภาพโดยวิเคราะห์ด้วยสีและพื้นผิว (Color &Texture) และระบบการป้อนกลับ (Relevance Feedback) จากตาราง 4.4 และ รูปที่ 4.4 จะพบว่าการค้นหาออกมาได้ผลคล้ายภาพต้นแบบทั้งในส่วนของสี พื้นผิว และความคล้ายของรูป โดยเมื่อทำการใช้ระบบป้อนกลับ แล้วจะพบว่าสามารถค้นหารูปภาพได้ออกมาคล้ายคลึงมากกว่าการค้นหาครั้งที่ 1 โดยเมื่อทำการค้นหาโดยระบบป้อนกลับไปเรื่อย ๆ ก็จะพบว่าการค้นหานั้นมีรูปภาพที่คล้ายคลึงมากขึ้นตามลำดับ




**รูปที่ 5.1** กราฟเปรียบเทียบการทำงานของแต่ละวิธี

จากรูปกราฟ 5.1 นั้นจะเห็นว่าค่าความถูกต้อง (Precision) ของแต่ละวิธีนั้นแตกต่างกันอย่างชัดเจนโดย การค้นหาภาพด้วยพื้นผิว มีค่าความถูกต้องต่ำสุด ตามด้วยการค้นหาด้วยสี แต่เมื่อนำทั้ง 2 วิธีมารวมกันจากกราฟจะพบว่า การค้นหาด้วยสีและพื้นผิวพร้อมกันนั้นมีค่าความถูกต้องที่สูงโดยเมื่อนำการค้นหาแบบ 2 วิธีมารวมกับระบบการป้อนกลับ (Relevance Feedback) แล้วจะเห็นว่ามีค่าความถูกต้องสูงที่สุด แสดงถึงการค้นหาภาพนั้นมีประสิทธิภาพซึ่งจะเห็นได้จากค่าเฉลี่ยค่าความถูกต้อง (Average Precision) ในตาราง 5.1

**ตารางที่ 5.1** เปรียบเทียบค่าเฉลี่ยค่าความถูกต้อง (Average Precision)ของทั้ง 4 วิธี

| Methods | Average Precision |
|---|---|
| Color | 33 % |
| Texture | 18 % |
| Color & Texture | 53 % |
| Color & Texture+Feedback | 66 % |

จากการเปรียบเทียบทดลองทั้งหมดจากตาราง 5.1 และกราฟรูปที่ 5.1 จะเห็นได้ว่าค่าเฉลี่ยของค่าความถูกต้อง (Precision) นั้นแปรผันตามกราฟแสดงให้เห็นว่าการค้นหาด้วยการวิเคราะห์ด้วยสีและพื้นผิวพร้อมกันนั้นเป็นการค้นหาภาพที่ออกมาทีความคล้ายคลึงกับภาพต้นแบบพอสมควร ส่วนการวิเคราะห์สี (Color) อย่างเดียวและพื้นผิว (Texture) อย่างเดียวมีค่าค่าความถูกต้อง (Average Precision) รองลงมาตามลำดับ แสดงให้เห็นว่าการค้นหาภาพแบบใช้ 2 ทฤษฎีร่วมกันนั้นมีประสิทธิภาพมากกว่า

การค้นหาภาพแบบใช้ทฤษฎีเดียว อีกทั้งเมื่อนำการค้นหาภาพแบบใช้ 2 ทฤษฎีร่วมกัน มาผ่านกระบวนการป้อนกลับ (Relevance Feedback) ก็จะยิ่งทำให้การค้นหานั้นมีประสิทธิภาพมากขึ้น

ในการค้นหาภาพนั้นอาจจะมีความคลาดเคลื่อนได้บ้างซึ่งเกิดจากหลายปัจจัย เช่น ภาพในฐานข้อมูลไม่เพียงพอ ภาพที่ต้องการหามีความซับซ้อนมากเกินไป หรืออาจจะต้องใช้ทฤษฎีอื่นเข้ามาช่วย เพื่อให้การค้นหาภาพนั้นมีประสิทธิภาพและถูกต้องตามความต้องการของผู้ใช้มากขึ้น

## 5.2 ปัญหาที่พบ

5.2.1 ถ้าหากรูปภาพในฐานข้อมูลมีไม่เพียงพอผลการค้นหาก็จะไม่เป็นไปตามที่ต้องการหรือมีความคล้ายคลึงน้อย

5.2.2 ในการเพิ่มภาพในฐานข้อมูลนั้น จำเป็นที่จะต้องเพิ่มที่ละภาพ ไม่สามารถเพิ่มทีละหลาย ๆ ภาพพร้อมกันได้ ทำให้ใช้เวลานานสำหรับการเพิ่มฐานข้อมูลใหม่ทีละมาก ๆ

5.2.3 ถ้าขนาดของรูปภาพที่ใช้ในการค้นหาไม่ได้มาตรฐานอาจทำให้การค้นหามีความคลาดเคลื่อนหรือผิดพลาดได้เช่น รูปแนวพาโนรามา

## 5.3 ข้อเสนอแนะ

5.3.1 หากมีการพัฒนาต่อไปควรจะเพิ่มในส่วนของวิเคราะห์ด้วยรูปร่าง (Shapes Analysis) หรือทฤษฎีอื่นๆที่สามารถนำมาเพิ่มประสิทธิภาพในการวิเคราะห์รูปภาพเพิ่มเติมเพื่อให้การค้นหาภาพได้ดีขึ้น

5.3.2 ควรจะมีข้อมูลรูปภาพในฐานข้อมูลที่เพียงพอเพราะผลการค้นหาที่ไม่ตรงกับภาพที่ต้องการหรือคลาดเคลื่อนนั้นอาจเกิดมาจากข้อมูลในฐานข้อมูลที่ไม่เพียงพอ

# เอกสารอ้างอิง

[1] Ahmed Khalil and Rami Al-Tayeche. "CBIR:Content Based Image Retrieval" [Online], Available: http://www.sce.carleton.ca/faculty/cuhadar/CBIR/index.htm. 2002

[2] Danang Suharno Ngadinegaran. "Finding Similiar Images" [Online], Available: http://www.codeproject.com/. 2002

[3] T.Andrysiak and M.Choras'. "IMAGE RETRIEVAL BASED ON HIERARCHICAL GABOR FILTERS" [Online], Available: http://matwbn.icm.edu.pl/ksiazki/amc/amc15/amc1540.pdf. 2005

[4] ยุทธนา ลีลาศวัฒนกุล. "Visual C# .NET ฉบับวินโดวฟอร์ม" .บริษัท ดวงกมลสมัย จำกัด. 2545

# ประวัติผู้เขียนโครงงาน

ชื่อ นาย กิตติ์ ก้อนทอง

ภูมิลำเนา 276 หมู่ 6 ต.หล่มเก่า อ.หล่มเก่า จ.เพชรบูรณ์

ประวัติการศึกษา

- จบมัธยมศึกษาจาก
- ปัจจุบัน กำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : **gitt6184@hotmail.com**

ชื่อ นาย จตุรัฐ คำขาว

ภูมิลำเนา 80 หมู่ 10 ต.แม่ใจ อ.แม่ใจ จ.พะเยา

ประวัติการศึกษา

- จบมัธยมศึกษาจาก โรงเรียนพะเยาพิทยาคม
- ปัจจุบัน กำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : **best_cpe@hotmail.com**

ชื่อ นาย วีรชัย วรวิริยะประเสริฐ

ภูมิลำเนา 5/11 ถ.ชวนชื่น ต.ในเมือง อ.เมือง จ.ขอนแก่น

ประวัติการศึกษา

- จบมัธยมศึกษาจาก โรงเรียนสาธิตศึกษาศาสตร์ มหาวิทยาลัยขอนแก่น
- ปัจจุบัน กำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : **tee_cpeix@hotmail.com**