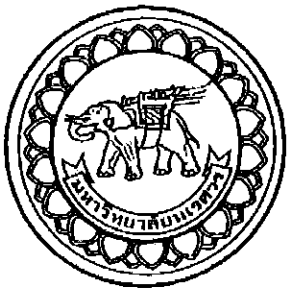


การศึกษาการวัดค่าความเหมือนเพื่อค้นหาภาพ

The Study of Similarity Measure for Pictures Search

นายชเยศ กองจันทร์ รหัส 47380017

ห้องสมุดคณะวิศวกรรมศาสตร์
25 พ.ค. 2553
วันที่รับ.....
เลขทะเบียน.....150053283
เลขเรียกหนังสือ.....ปร. ร19ปก 2550
มหาวิทยาลัยนเรศวร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2550

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	**การศึกษาการวัดค่าความเหมือนเพื่อค้นหาภาพ**
ผู้ดำเนินโครงงาน	นายชเยศ กองจันทร์ รหัส 47380017
อาจารย์ที่ปรึกษา	ดร. ไพศาล มุณีสว่าง
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2550

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ดร.ไพศาล มุณีสว่าง)

.....กรรมการ

(ดร.ชัยรัตน์ พินทอง)

.....กรรมการ

(อาจารย์ศิริพร เดชะศิลารักษ์)

หัวข้อโครงงาน	การศึกษาการวัดค่าความเหมือนเพื่อค้นหาภาพ
ผู้ดำเนินโครงงาน	นายชเยศ กองจันทร์ รหัส 47380017
อาจารย์ที่ปรึกษา	ดร. ไพศาล มุณีสว่าง
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2550

.....

บทคัดย่อ

โครงงานนี้เป็นโครงงานที่ศึกษาเกี่ยวกับการการพัฒนาโปรแกรมค้นหารูปภาพในฐานข้อมูล โดยการวิเคราะห์รูปภาพจะใช้ทฤษฎีองค์ประกอบพื้นฐานของรูปภาพ Content Based Image Retrieval (CBIR) ในการวิเคราะห์รูปภาพเพื่อสำหรับเก็บเป็นดัชนี (Index) เก็บไว้ในฐานข้อมูล โดยดัชนี (Index) ในฐานข้อมูลจะมีไว้สำหรับเปรียบเทียบกับดัชนี (Index) ของรูปภาพตัวอย่าง การเปรียบเทียบระหว่างดัชนี (Index) ของรูปภาพตัวอย่าง กับดัชนี (Index) ในฐานข้อมูล โครงงานนี้ได้จัดให้มีวิธีการเปรียบเทียบให้เลือกใช้ 5 วิธี เพื่อเพิ่มประสิทธิภาพในการค้นหาและได้วิเคราะห์วิธีการเปรียบเทียบว่าวิธีใดมีประสิทธิภาพในการค้นหามากน้อยเพียงใด ซึ่งโปรแกรมจะถูกพัฒนาบนวินโดว์แอพพลิเคชั่น (Windows Application) ของ Microsoft Visual Studio.NET 2005 โดยใช้ภาษา C# และใช้โปรแกรม Microsoft Office Access 2003 จัดการกับฐานข้อมูล

Project title	The Study of Similarity Measure for Pictures Search
Name	Mr.Chayet Kongchan ID 47380017
Project advisor	Paisarn Muneesawang , Ph.D.
Major	Computer Engineering.
Department	Electrical and Computer Engineering.
Academic year	2007

Abstract

This project studies and develops search engine, a window software application for indexing and retrieval of pictures. The system analysis image by Content Based Image Retrieval (CBIR) to obtain image for indexing image database. These indexes are used by content matching matrix to compare of a given query with other pictures stored in the database. It offers five methods for increasingly searching efficiency. The programs we used in this project are c# Window Application of Microsoft Visual Studio .NET 2005 and Microsoft Access office 2003.

กิตติกรรมประกาศ

ในการทำโครงงานวิศวกรรมครั้งนี้ ผู้จัดทำขอกราบขอบพระคุณ ดร.ไพศาล มุณีสว่าง ที่ได้ให้คำปรึกษาโครงงานนี้ ทั้งทฤษฎีและขั้นตอนการปฏิบัติงานต่างๆ

และขอกราบขอบพระคุณ ดร.ชัยรัตน์ พินทอง และ อาจารย์ศิริพร เคชะศิลารักษ์ ที่ได้เสียสละเวลาเพื่อทำการตรวจสอบการทำงานและชี้แนวทางในการแก้ไขปัญหาโครงงานนี้

นายชเยศ กองจันทร์

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญรูป ..... ฉ

สารบัญตาราง ..... ช

บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญของเรื่อง 1

1.2 วัตถุประสงค์ของโครงงาน 2

1.3 ขอบข่ายของโครงงาน 2

1.4 ขั้นตอนการดำเนินงานของโครงการ 2

1.5 แผนการดำเนินโครงการ 3

1.6 ผลที่คาดว่าจะได้รับ 4

1.7 งบประมาณทั้งโครงงาน 4

บทที่ 2 ทฤษฎีเบื้องต้น

2.1 การค้นหาภาพโดยใช้องค์ประกอบพื้นฐานของภาพ (CBIR) 5

2.2 การอธิบายสี (Color Descriptors) 5

2.3 การอธิบายพื้นผิว (Texture Descriptors) 7

2.4 การอธิบายรูปร่าง (Shape Descriptors) 9

2.5 การเปรียบเทียบ 9

2.5.1 Euclidean Distance 10

2.5.2 Cosine Similarity 10

2.5.3 Histogram Intersection 10

2.5.4 Manhattan Distance 11

2.5.5 Kullback-Leibler (K-L) Divergence 11

สารบัญ (ต่อ)

หน้า

2.6 Quick Sort 11

บทที่ 3 วิธีการดำเนินโครงงาน

3.1 การค้นหารูปภาพ 13

3.1.1 เลือกรูปภาพต้นแบบ 14

3.1.2 เลือกวิธีการเปรียบเทียบ 14

3.1.3 เปรียบเทียบค่าในฐานข้อมูล 14

3.1.4 นำค่าที่ได้จากการเปรียบเทียบมาเรียงลำดับ 17

3.1.5 แสดงรูปภาพที่มีความคล้ายคลึงกับรูปภาพต้นแบบ 17

3.2 การจัดการกับฐานข้อมูล 17

บทที่ 4 ผลการทดลอง

4.1 การค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Euclidean Distance 19

4.2 การค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Cosine Similarity 23

4.3 การค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Histogram Intersection 26

4.4 การค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Manhattan Distance 29

4.5 การค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Kullback-Leibler Divergence 32

4.6 กราฟแสดงค่าความถูกต้องของการค้นหาทั้ง 5 วิธีจากผลการทดลอง 35

4.7 เปรียบเทียบผลการทดลอง 37

บทที่ 5 บทสรุป

5.1 สรุปผลการทดลอง 39

5.2 ปัญหาที่พบ 39

5.3 ข้อเสนอแนะ 40

เอกสารอ้างอิง 41

ประวัติผู้เขียนโครงงาน 42

สารบัญรูป

รูปที่		หน้า
2.1	แสดงขั้นตอนการหาค่าฟีเจอร์เวกเตอร์	8
2.2	แสดงการอธิบายรูปร่าง	9
2.3	การเรียงลำดับแบบ Quick Sort	12
3.1	แผนผังการทำงานของโปรแกรม	13
3.2	ตัวอย่างการเก็บข้อมูลทั้งหมดลงในฐานข้อมูล	18
3.3	ตัวอย่างโปรแกรม	18
4.1	แสดงผลการการค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Euclidean Distance	20
4.2	แสดงผลการการค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Cosine Similarity	23
4.3	แสดงผลการการค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Histogram Intersection	26
4.4	แสดงผลการการค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Manhattan Distance	29
4.5	แสดงผลการการค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Kullback-Leibler (K-L) Divergence	32
4.6	รูปกราฟแสดงค่าความถูกต้องของการค้นหารูปภาพโดยใช้ Euclidean Distance	35
4.7	รูปกราฟแสดงค่าความถูกต้องของการค้นหารูปภาพโดยใช้ Cosine Similarity	35
4.8	รูปกราฟแสดงค่าความถูกต้องของการค้นหารูปภาพโดยใช้ Histogram Intersection	36
4.9	รูปกราฟแสดงค่าความถูกต้องของการค้นหารูปภาพโดยใช้ Manhattan Distance	36
4.10	รูปกราฟแสดงค่าความถูกต้องของการค้นหารูปภาพโดยใช้ Kullback-Leibler	37
4.11	รูปกราฟเปรียบเทียบค่าความถูกต้องการค้นหาของทั้ง 5วิธี	38

สารบัญตาราง

ตารางที่		หน้า
1.1	แสดง Gantt chart	3
4.1	บันทึกค่าผลการทดลองการค้นหารูปภาพโดยใช้วิธี Euclidean Distance	21
4.2	บันทึกค่าผลการทดลองการค้นหารูปภาพโดยใช้วิธี Cosine Similarity	24
4.3	บันทึกค่าผลการทดลองการค้นหารูปภาพโดยใช้วิธี Histogram Intersection	27
4.4	บันทึกค่าผลการทดลองการค้นหารูปภาพโดยใช้วิธี Manhattan Distance	30
4.5	บันทึกค่าผลการทดลองการค้นหารูปภาพโดยใช้วิธี Kullback-Leibler Divergence..	33
4.6	เปรียบเทียบค่า Average Precision ของ 5 วิธี	37

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของเรื่อง

ปัจจุบันเทคโนโลยีอินเตอร์เน็ตได้เข้ามามีบทบาทกับมนุษย์มากขึ้น เนื่องจากอินเตอร์เน็ตเป็นแหล่งรวบรวมข้อมูลข่าวสารต่างๆ ไว้มากมาย รวมทั้งการค้นหารูปภาพด้วย ตัวอย่างเช่น Google การค้นหาของ Google เป็นการค้นหารูปภาพด้วยใช้คำบ่งชี้ (Key Word) หากไม่ทราบชื่อรูปภาพที่ต้องการก็จะทำให้ประสบปัญหา ดังนั้นจึงมีการสร้างเครื่องมือค้นหารูปภาพขึ้น ซึ่งโปรแกรมค้นหา (Search Engine) ที่มี ณ ตอนนี้จะทำการหารูปภาพโดยค้นหารูปภาพตามที่ผู้ค้นหาป้อนความต้องการให้โปรแกรม แต่ก็ยังมีรูปภาพมากมายที่ไม่ใช่รูปภาพที่ตรงตามความต้องการ ซึ่งโปรแกรมค้นหา (Search Engine) ส่วนใหญ่จะใช้คำบ่งชี้ (Key Word) เป็นตัวอ้างอิงในการค้นหา ซึ่งการใช้ คำบ่งชี้ (Key Word) ในการค้นหารูปภาพนั้นยังไม่เจาะจงความต้องการมากนัก อย่างเช่น ต้องการค้นหารูปภาพของ "นก" เราใช้ คำบ่งชี้ (Key Word) ว่า "bird" เป็นตัวอ้างอิงในการค้นหารูปภาพ รูปภาพที่แสดงออกมาอาจจะมีทั้งรูปคน รูปป้ายต่างๆ หรือจะเป็นรูปของบริษัท เนื่องจากคนที่มีชื่อว่า bird หรือมีรูปภาพอื่นที่ไม่เกี่ยวข้องกับรูปภาพที่ต้องการค้นหา แต่คนเซฟชื่อว่า bird จึงทำให้รูปภาพเหล่านั้นแสดงออกมา

ดังนั้นโครงการนี้จึงมีความต้องการที่จะแก้ไขปัญหาดังกล่าว จึงมีแนวคิดที่จะสร้างโปรแกรมค้นหาไฟล์รูปภาพ ที่มีความแม่นยำ โดยจะนำรูปภาพตัวอย่างที่ต้องการค้นหามาเป็นตัวอ้างอิงในการค้นหารูปภาพแทน คำบ่งชี้ (Key Word) แต่เนื่องจากแนวคิดนี้ได้มีคนนำไปสร้างโปรแกรมค้นหาไฟล์รูปภาพขึ้นแล้ว แต่มี อัลกอริทึม ในการเปรียบเทียบรูปภาพเพียง 1 วิธีจึงไม่สามารถรู้ได้ว่า อัลกอริทึม ที่ใช้ในการเปรียบเทียบนั้นมีประสิทธิภาพในการค้นหามากน้อยเพียงใด โครงการนี้จึงได้เพิ่ม อัลกอริทึม ในการเปรียบเทียบรูปภาพขึ้นเป็น 5 วิธี เพื่อเพิ่มประสิทธิภาพในการค้นหารูปภาพให้แก่โปรแกรมค้นหารูปภาพ และยังศึกษาวิเคราะห์ อัลกอริทึม ในการเปรียบเทียบต่างๆ ว่า อัลกอริทึม ใดมีประสิทธิภาพในการค้นหาสูง และ อัลกอริทึม ใดมีประสิทธิภาพในการค้นหาน้อย

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 สร้างโปรแกรมค้นหารูปภาพ (Search Engine) ที่สามารถค้นหาไฟล์รูปภาพ ในฐานข้อมูลที่เตรียมไว้แล้ว โดยใช้วิธีการหารูปภาพโดยการนำเอารูปภาพตัวอย่าง มาค้นหารูปภาพที่มีความคล้ายคลึงกัน

1.2.2 ศึกษาการใช้ภาษา Microsoft Visual C#.NET ในการพัฒนาโปรแกรมประเภท Search Engine

1.2.3 ศึกษาและวิเคราะห์ อัลกอริทึม ที่ใช้ในการเปรียบเทียบรูปภาพจำนวน 5 วิธี

1.3 ขอบข่ายของโครงงาน

สร้างโปรแกรมค้นหา (Search Engine) ในการค้นหาภาพซึ่งเราจะใช้หลักการวิเคราะห์จากดัชนีภาพ (Image Index) ของรูปภาพตัวอย่างที่ต้องการค้นหา เปรียบเทียบกับดัชนีภาพ (Image Index) ที่มีอยู่ในฐานข้อมูลอยู่แล้ว โดยโปรแกรมค้นหาจะมี อัลกอริทึม ในการเปรียบเทียบรูปภาพให้เลือกใช้ 5 วิธี เพื่อเพิ่มประสิทธิภาพให้แก่โปรแกรมค้นหา (Search Engine) และศึกษาวิเคราะห์อัลกอริทึม ในการเปรียบเทียบต่างๆ ว่า อัลกอริทึม มีประสิทธิภาพในการเปรียบเทียบมากน้อยเพียงใด

1.4 ขั้นตอนการดำเนินงานของโครงการ

1.4.1 ศึกษาการใช้ภาษา Microsoft Visual C#.NET

1.4.2 ออกแบบโปรแกรมค้นหา (Search Engine) ด้วยภาษา Visual C#.NET

1.4.3 ศึกษาและค้นหาข้อมูลเกี่ยวกับ อัลกอริทึม ในการเปรียบเทียบ

1.4.4 จัดการเกี่ยวกับฐานข้อมูล (Database)

1.4.5 สร้างโปรแกรมค้นหา (Search Engine) ด้วยภาษา Visual C#.NET

1.4.6 ทดสอบโปรแกรม และปรับปรุงแก้ไขโปรแกรม

1.4.7 จัดทำเอกสารโครงงาน

1.5 แผนการดำเนินโครงการ

ตารางที่ 1.1 แสดง Gantt chart

ขั้นตอนการดำเนินงาน	ปี 2550		ปี 2551			
	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
ศึกษาการใช้ภาษา Microsoft Visual C#.NET	⟷					
ออกแบบโปรแกรมค้นหา (Search Engine) ด้วยภาษา Visual C#.NET		⟷				
ศึกษาและค้นหาข้อมูลเกี่ยวกับ อัลกอริทึม ในการเปรียบเทียบ			⟷			
จัดการเกี่ยวกับฐานข้อมูล (Database)				⟷		
สร้างโปรแกรมค้นหา (Search Engine) ด้วยภาษา Visual C#.NET				⟷	⟷	⟷
ทดสอบโปรแกรม และปรับปรุงแก้ไขโปรแกรม						⟷
จัดทำเอกสารโครงงาน						⟷

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 สามารถสร้างโปรแกรมค้นหา (Search Engine) ที่สามารถค้นหารูปภาพอย่างมีประสิทธิภาพและตรงตามความต้องการของผู้ใช้

1.6.2 สามารถวิเคราะห์ได้ว่า อัลกอริทึม การเปรียบเทียบ ทั้ง 5 วิธี วิธีใดมีประสิทธิภาพในการเปรียบเทียบได้ดีที่สุดและน้อยที่สุด

1.7 งบประมาณทั้งโครงงาน

หมวดค่าวัสดุและใช้สอย

งบที่จะได้ 1000 บาท

บทที่ 2
ทฤษฎีเบื้องต้น

หลักการที่ใช้ในโปรแกรมคือการค้นหาภาพโดยใช้องค์ประกอบพื้นฐาน Content Base Image Retrieval (CBIR) ซึ่งจะเป็นการนำรูปภาพตัวอย่างที่ต้องการค้นหามาวิเคราะห์ แล้วนำไปเปรียบเทียบกับรูปภาพในฐานข้อมูล โดยจะมีวิธีการเปรียบเทียบรูปภาพให้เลือกใช้ 5 วิธี เพื่อเพิ่มประสิทธิภาพให้แก่โปรแกรม และได้ศึกษาและวิเคราะห์ อัลกอริทึม ในการเปรียบเทียบรูปภาพ ซึ่งทฤษฎีต่างๆ นั้นจะกล่าวดังต่อไปนี้

2.1 การค้นหาภาพโดยใช้องค์ประกอบพื้นฐานของภาพ (CBIR)

โดยจะนำรูปภาพตัวอย่างที่ต้องการค้นหามาวิเคราะห์องค์ประกอบพื้นฐานของรูปภาพ Content Base Image Retrieval (CBIR) โดยค่าที่ได้จากการวิเคราะห์จะมีค่าเป็น ดัชนี (Index) แล้วนำดัชนีที่ได้จากการวิเคราะห์ไปเปรียบเทียบกับดัชนีของรูปภาพในฐานข้อมูล แล้วคำนวณหาค่าความแตกต่าง (Distance) เพื่อนำค่าที่ได้ไปค้นหารูปภาพที่มีความคล้ายคลึงในฐานข้อมูล

2.2 การอธิบายสี (Color Descriptors)

เป็นเทคนิคสำหรับการสร้างดัชนีฐานข้อมูลรูปภาพและการค้นหา โดยหลักการคือ ทำการวิเคราะห์รูปภาพแต่ละรูป เพื่อคำนวณหาค่าองค์ประกอบหรือลักษณะที่แสดงสัดส่วน pixels ของแต่ละรูปภาพ เช่น การหาค่าองค์ประกอบของสีแต่ละรูปถูกวิเคราะห์ แล้วเก็บไว้ที่ฐานข้อมูล เมื่อผู้ใช้จะค้นหา สามารถเลือกรูปภาพเจาะจงลงไป ตามสัดส่วนที่ต้องการของแต่ละสี (เช่น สีเขียว 75% สีแดง 10% และสีน้ำเงิน 15%) หรือการนำรูปภาพตัวอย่างที่เลือก ไปคำนวณหาค่าองค์ประกอบของสี แล้วนำไปเปรียบเทียบกับรูปภาพที่มีองค์ประกอบของสีใกล้เคียงกันมากที่สุด แล้วจึงแสดงรูปภาพให้ผู้ใช้งาน

การสร้างกราฟสี (Color Histogram)

Color Histogram [3] คือ การสร้างกราฟที่มีจำนวนช่วงสีเท่ากับจำนวนที่ต้องการ โดยแต่ละช่วง จะแบ่งช่วงสีจากต่ำสุดถึงสูงสุดเท่า ๆ กันทุกช่วงสี จากนั้นนำจุดสี (Pixel) ทุกจุดในรูปภาพมาเก็บค่าความถี่ของจำนวนจุดสีที่มีอยู่แต่ละช่วงสี โดยช่วงที่จะแบ่ง ยิ่งมีความถี่มาก ก็จะสามารถเปรียบเทียบได้ละเอียดมาก ในที่นี้จะแบ่งช่องสี RGB สีละ 16 ช่วง รวมทั้งหมด 48 ช่วง จะได้เวคเตอร์ที่ประกอบขึ้นดังต่อไปนี้

$$F = \{R_1, R_2, ..., R_{16}, G_1, G_2, ..., G_{16}, B_1, B_2, ..., B_{16}\}$$

R_{1-16} คือ ค่าความถี่ของแต่ละช่วงสี จำนวน 16 ช่วง ของ สีแดง

G_{1-16} คือ ค่าความถี่ของแต่ละช่วงสี จำนวน 16 ช่วง ของ สีเขียว

B_{1-16} คือ ค่าความถี่ของแต่ละช่วงสี จำนวน 16 ช่วง ของ สีน้ำเงิน

ซึ่งการได้มาซึ่งสมการนี้ จะทำการรับค่าสีจาก Pixel ทั้งรูปภาพมาเก็บค่าความถี่ ดังที่เราได้แบ่งไว้ แล้วนำไปเก็บไว้ในฐานข้อมูล เพื่อใช้ในการเปรียบเทียบรูปภาพต่อไป

การวัดน้ำหนักสี (Color Moment)

Color Moment [3] เป็นวิธีการหาเวคเตอร์ ซึ่งประกอบไปด้วยทั้งหมด 2 ส่วน ที่ใช้ในการพิจารณา คือ การหาค่าเฉลี่ย (mean) และค่าส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation : SD) ของรูปภาพ

วิธีการ คือ เก็บค่าจุดสี (Pixel) ทุกจุดที่มีอยู่ในรูปภาพโดยตรง แล้วนำค่าที่ได้ มาหาค่าเฉลี่ยและค่าส่วนเบี่ยงเบนมาตรฐานของรูปภาพ โดยจะทำการคำนวณทั้ง 3 สี ในระบบ RGB ซึ่งมีสมการดังต่อไปนี้

$$M_1 = \frac{1}{N}\sum_{i=1}^{N} P_i$$

M_1 คือ ค่าเฉลี่ยของ Pixel ทั้งภาพ (Mean)

N คือ จำนวนจุดสี (Pixel) ทั้งหมด

P_i คือ ค่าของจุดสี (Pixel) ณ จุดที่ i

$$M_2 = \left[\frac{1}{N}\sum_{i=1}^{N}(P_i - M_1)^2\right]^{1/2}$$

M_2 คือ ค่าส่วนเบี่ยงเบนมาตรฐานของ Pixel ทั้งภาพ (SD)

N คือ จำนวนจุดสี (Pixel) ทั้งหมด

P_i คือ ค่าของจุดสี (Pixel) ณ จุดที่ i

M_1 คือ ค่าเฉลี่ยของ Pixel ทั้งภาพ (Mean)

เมื่อได้ค่าทั้งสองค่าแล้ว จะได้เวคเตอร์ออกมาตามจำนวนภาพดังต่อไปนี้

$$F_1 = \{RM_{11}, RM_{12}, GM_{11}, GM_{12}, BM_{11}, BM_{12}\}$$
$$F_2 = \{RM_{21}, RM_{22}, GM_{21}, GM_{22}, BM_{21}, BM_{22}\}$$
.....
.....
$$F_M = \{RM_{M1}, RM_{M2}, GM_{M1}, GM_{M2}, BM_{M1}, BM_{M2}\}$$

F_M	คือ	เวคเตอร์ที่ได้จากการทำ Color Moment ที่ M
R_{M1}	คือ	ค่าเฉลี่ย (Mean) ที่ได้จากการคำนวณของสีแดง
R_{M2}	คือ	ค่าส่วนเบี่ยงเบนมาตรฐานที่ได้จากการคำนวณของสีแดง
G_{M1}	คือ	ค่าเฉลี่ย (Mean) ที่ได้จากการคำนวณของสีเขียว
G_{M2}	คือ	ค่าส่วนเบี่ยงเบนมาตรฐานที่ได้จากการคำนวณของสีเขียว
B_{M1}	คือ	ค่าเฉลี่ย (Mean) ที่ได้จากการคำนวณของสีน้ำเงิน
B_{M2}	คือ	ค่าส่วนเบี่ยงเบนมาตรฐานที่ได้จากการคำนวณของสีน้ำเงิน

แล้วนำไปเก็บไว้ในฐานข้อมูล เพื่อใช้ในการเปรียบเทียบรูปภาพต่อไป

2.3 การอธิบายพื้นผิว (Texture Descriptors)

พื้นผิว (Texture) [6] เป็นคุณลักษณะที่มีอยู่ในทุก ๆ รูปภาพ โดยสามารถใช้พื้นผิว (Texture) เป็นหลักในการค้นหาข้อมูลรูปภาพที่มีลักษณะพื้นผิว (Texture) ตามต้องการ หรือใกล้เคียงกับที่ต้องการมากที่สุด การใช้พื้นผิว (Texture) เป็นคุณลักษณะหลักในการค้นหาข้อมูลรูปภาพนั้น ต้องหาพื้นผิว (Texture Descriptor) ที่ใช้แสดงได้ว่าสามารถบ่งชี้ ถึงลักษณะของพื้นผิว (Texture) รูปภาพนั้น ซึ่งในที่นี้ได้ใช้ทฤษฎีกาบอร์เวฟเลต (Gabor Wavelet) ในการหาตัวแทนพื้นผิว (Texture Descriptor) ของภาพ

กาบอร์ฟังก์ชันและเวฟเลตฟังก์ชัน (Gabor Function and Wavelet Function) [4]

การวิเคราะห์รูปภาพโดยใช้วิธีกาบอร์ฟิลเตอร์ (Gabor Filter) เป็นการวิเคราะห์เพื่อหาลักษณะเด่นของรูปภาพ โดยค่าที่แสดงลักษณะเด่นนั้น เรียกว่า ฟีเจอร์เวคเตอร์ (Feature Vector) โดยวิธี กาบอร์ฟิลเตอร์เป็นการใช้ความถี่เพื่อหาฟีเจอร์เวคเตอร์ ซึ่งความถี่ที่ใช้มีหลายความถี่โดยแบ่งเป็นช่วงๆ แล้วแต่ว่าจะกำหนดใช้ช่วงความถี่ในช่วงใด และช่วงความถี่ดังกล่าวนั้น สามารถปรับเปลี่ยนจุดศูนย์กลางของความถี่ (Bandpass) ในแต่ละช่วงได้ อีกทั้งยังสามารถหมุนแกนไปใน

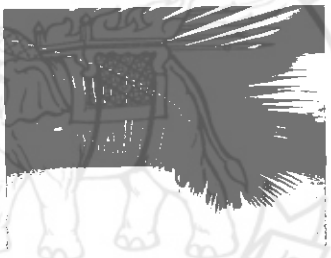
ทิศทางที่ทำมุมต่างกันได้อีกด้วย เพื่อให้การวิเคราะห์ครอบคลุมในทุกช่วงความถี่ และในส่วนของจำนวนฟิลเตอร์ จะขึ้นอยู่กับว่าต้องการกรองข้อมูลหรือรูปภาพให้มีความละเอียดมากน้อยเพียงใด ซึ่งจะไม่เป็นค่าที่ตายตัว

วิธีการหาค่าฟีเจอร์เวคเตอร์ คือ การนำภาพที่ต้องการวิเคราะห์ นำไปเข้ากับกาบอร์ฟิลเตอร์ โดยที่จำนวนฟิลเตอร์ กำหนดให้มีขนาด 4 คูณ 6 ฟิลเตอร์จะได้ค่าฟีเจอร์เวคเตอร์ ของแต่ละภาพย่อย มีค่าเฉลี่ย (Mean) 24 ค่า และค่ามาตรฐาน (Standard) 24 ค่า

$$\mu(\text{mean}) = [\mu_{00}, \mu_{01}, \ldots, \mu_{23}] \quad , \quad \sigma(\text{standard}) = [\sigma_{00}, \sigma_{01}, \ldots, \sigma_{23}]$$

หลังจากนั้นทำการแปลงค่าที่ได้ให้เป็นเวคเตอร์ขนาด 48 แล้ว นำค่า Mean และค่า Standard มารวมกันจึงได้ค่าฟีเจอร์เวคเตอร์ดังนี้

$$\text{ค่าฟีเจอร์เวคเตอร์} = [\mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \ldots, \mu_{47}, \sigma_{47}]$$



ภาพที่ต้องการวิเคราะห์

⇩

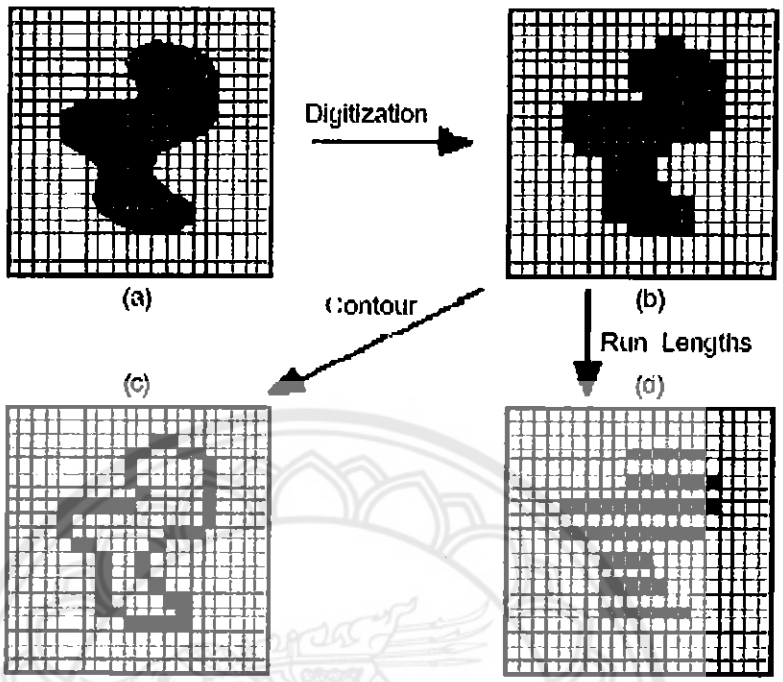
Gabor Features

⇩

$$\overline{f}_q = [\mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \ldots, \mu_{47}, \sigma_{47}]$$

รูปที่ 2.1 แสดงขั้นตอนการหาค่าฟีเจอร์เวกเตอร์

2.4 การอธิบายรูปร่าง (Shape Descriptors)



รูปที่ 2.2 แสดงการอธิบายรูปร่าง

รูปร่าง (Shape) แบ่งออกเป็น 2 ประเภทคือ เส้นขอบ (Boundary-based) และบริเวณ (Region-based) โดยเส้นขอบจะพิจารณาเฉพาะเส้นล้อมรอบวัตถุ ส่วนบริเวณจะพิจารณาพื้นที่ทั้งหมดที่อยู่ในบริเวณล้อมรอบวัตถุ

ภาพวัตถุในธรรมชาติ (a) เมื่อนำเข้าสู่ความจำของคอมพิวเตอร์ ภาพที่ได้เป็นภาพเชิงดิจิตอล (b) และสามารถแยกพิจารณาเป็นขอบ (c) หรือบริเวณภายใน (d) โดยการอธิบายรูปร่าง (Shape Descriptors) ของรูปภาพได้มาจาก Fourier Descriptor [5]

2.5 การเปรียบเทียบ

เมื่อต้องการหาความคล้ายคลึงของรูปภาพต้นแบบกับรูปภาพในฐานข้อมูล จะต้องนำข้อมูลของรูปภาพต้นแบบมาเปรียบเทียบกับข้อมูลรูปภาพในฐานข้อมูล โดยการคำนวณหาค่าระยะห่าง (Distance) ระหว่างรูปภาพต้นแบบกับรูปภาพในฐานข้อมูล โดยจะมี อัลกอริทึม ในการเปรียบเทียบให้เลือกใช้จำนวน 5 วิธี ดังนี้

2.5.1 Euclidean Distance [7]

การหาระยะห่าง (Euclidean Distance) ระหว่างจุด $P=(p_1,p_2,\ldots,p_n)$ และ $Q=(q_1,q_2,\ldots,q_n)$ จะได้ว่า

$$d(P,Q)=\sqrt{(p_1-q_1)^2+(p_2-q_2)^2+\ldots(p_n-q_n)}=\sqrt{\sum_{i=1}^{n}(p_i-q_i)^2}$$

d(P,Q) = ระยะห่าง (Euclidean Distance)

p_i = เป็นค่าดัชนี (Index) ของรูปภาพต้นแบบ

q_i = เป็นค่าดัชนี (Index) ของรูปภาพในฐานข้อมูล

2.5.2 Cosine Similarity [8]

การหาค่าความเหมือน (Cosine Similarity) ระหว่างจุด $P=(p_1,p_2,\ldots,p_n)$ และ $Q=(q_1,q_2,\ldots,q_n)$ จะได้ว่า

$$d(P,Q)=\frac{\vec{p}_i\cdot\vec{q}_i}{|\vec{p}_i||\vec{q}_i|}=\frac{\sum_{i=1}^{n}p_iq_i}{\sqrt{\sum_{i=1}^{n}p_i^{\,2}}\sqrt{\sum_{i=1}^{n}q_i^{\,2}}}$$

d(P,Q) = ค่าความเหมือน (Cosine Similarity)

p_i = เป็นค่าดัชนี (Index) ของรูปภาพต้นแบบ

q_i = เป็นค่าดัชนี (Index) ของรูปภาพในฐานข้อมูล

2.5.3 Histogram Intersection [9]

การหาค่าความเหมือน (Histogram Intersection) ระหว่างจุด $P=(p_1,p_2,\ldots,p_n)$ และ $Q=(q_1,q_2,\ldots,q_n)$ จะได้ว่า

$$d(P,Q)=\frac{\sum_{i=1}^{n}\min(p_i,q_i)}{\sum_{i=1}^{n}q_i}$$

d(P,Q) = ค่าความเหมือน (Histogram Intersection)

p_i = เป็นค่าดัชนี (Index) ของรูปภาพต้นแบบ

q_i = เป็นค่าดัชนี (Index) ของรูปภาพในฐานข้อมูล

2.5.4 Manhattan Distance [10]

การหาระยะห่าง (Manhattan Distance) ระหว่างจุด P =(p_1,p_2, …,p_n) และQ = (q_1,q_2,… ,q_n) จะได้ว่า

$$d(P,Q) = \sum_{i=1}^{n} |p_i - q_i|$$

d(P,Q) = ระยะห่าง (Manhattan Distance)

p_i = เป็นค่าดัชนี (Index) ของรูปภาพต้นแบบ

q_i = เป็นค่าดัชนี (Index) ของรูปภาพในฐานข้อมูล

2.5.5 Kullback-Leibler (K-L) Divergence [11]

การหาค่าความเหมือน (Kullback-Leibler (K-L) Divergence) ระหว่างจุด P =(p_1,p_2, …,p_n) และ Q = (q_1,q_2,… ,q_n)จะได้ว่า

$$d(P,Q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i}$$

d(P,Q) = ค่าความเหมือน (Kullback-Leibler (K-L) Divergence)

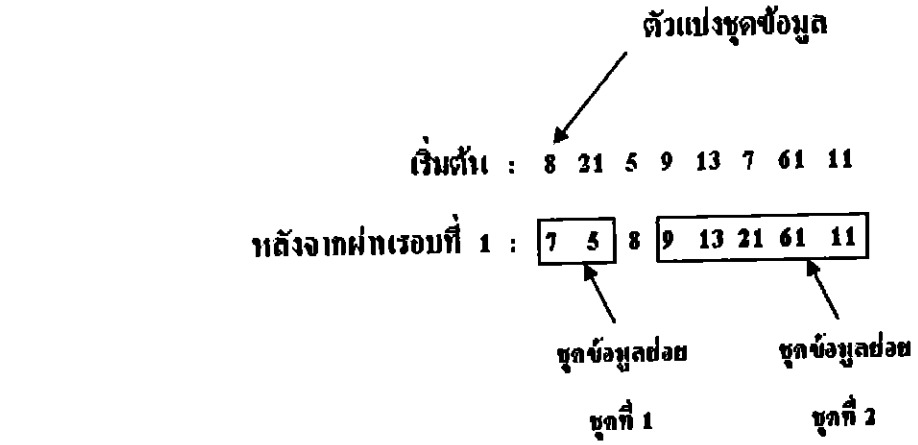
p_i = เป็นค่าดัชนี (Index) ของรูปภาพต้นแบบ

q_i = เป็นค่าดัชนี (Index) ของรูปภาพในฐานข้อมูล

2.6 Quick Sort

การเรียงลำดับในลักษณะนี้ เป็นการปรับปรุงมาจากการเรียงลำดับแบบ Bubble เพื่อให้การเรียงลำดับเร็วขึ้น วิธีนี้เหมาะกับการเรียงข้อมูลที่มีจำนวนมาก หรือมีขนาดใหญ่ และเป็นวิธีการเรียงข้อมูลที่ให้ค่าเฉลี่ยของเวลาน้อยที่สุดเท่าที่ค้นพบวิธีหนึ่ง

การเรียงลำดับแบบ Quick Sort จะเป็นการเปรียบเทียบสมาชิกที่ไม่อยู่ติดกัน โดยกำหนดข้อมูลค่าหนึ่ง เพื่อแบ่งชุดข้อมูลที่ต้องการเรียงลำดับออกเป็น 2 ส่วน จากนั้นก็จะทำการแบ่งย่อยชุดข้อมูล 2 ส่วนนั้นลงไปอีก ทำแบบนี้ไปเรื่อยๆจนข้อมูลแต่ละชุดมีสมาชิกเหลือเพียงตัวเดียว และทำให้ชุดข้อมูลทั้งหมดมีการเรียงลำดับ ดังรูปที่ 2.3



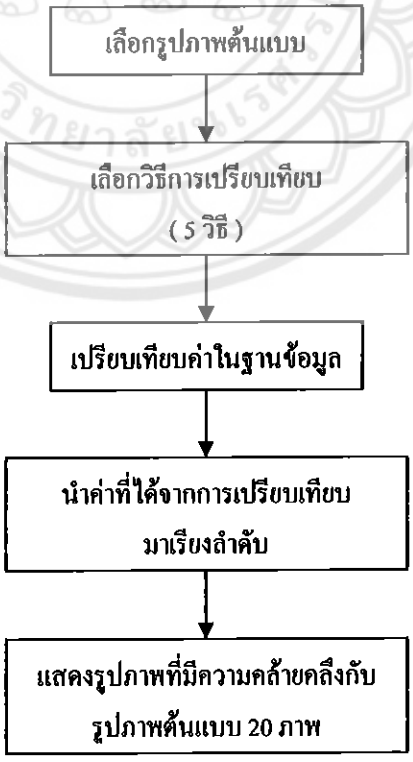
รูปที่ 2.3 การเรียงลำดับแบบ Quick Sort

บทที่ 3
วิธีการดำเนินการ

ในการทดลองการค้นหาข้อมูลรูปภาพ จะแบ่งขั้นตอนการดำเนินการงานออกเป็นส่วนต่างๆ ได้ดังนี้

3.1 การค้นหารูปภาพ

โปรแกรมค้นหาภาพนี้ได้พัฒนาขึ้นภายใต้สถาปัตยกรรม .NET โดยใช้ภาษา C# ซึ่งการที่จะค้นหาภาพได้นั้น เราต้องมีข้อมูลของรูปภาพอยู่ในฐานข้อมูลก่อน ในการค้นหานั้นจะเริ่มจากเลือกรูปภาพต้นแบบที่ต้องการค้นหาแล้วเลือกวิธีการเปรียบเทียบรูปภาพ โดยการเปรียบเทียบรูปภาพนี้จะมีวิธีการเปรียบเทียบให้เลือกใช้ 5 วิธี เมื่อเลือกรูปภาพต้นแบบได้แล้วก็นำค่าดัชนี (Index) ของรูปภาพต้นแบบไปเปรียบเทียบกับค่าดัชนี (Index) ของรูปภาพทั้งหมดในฐานข้อมูลตามวิธีการเปรียบเทียบที่ได้ทำการเลือกไว้ข้างต้น เพื่อที่จะหารูปภาพที่ใกล้เคียงกับรูปภาพต้นแบบมากที่สุด 20 ภาพ และนำไปแสดงผลโดยลำดับการแสดงผลจะเรียงลำดับความคล้ายคลึงของรูปภาพที่มีความคล้ายคลึงที่สุดแล้วลดลงมาเรื่อยๆ จากหลักการทำงานดังกล่าวสามารถนำมาเขียนแผนภาพได้รูปที่ 3.1



รูปที่ 3.1 แผนผังการทำงานของโปรแกรม

3.1.1 เลือกรูปภาพต้นแบบ

เลือกรูปภาพต้นแบบเพื่อนำมาหารูปภาพที่คล้ายคลึงกันในฐานข้อมูล และเก็บค่าดัชนี (Index) ของรูปภาพต้นแบบ เพื่อจะเอาไปเปรียบเทียบกับค่าดัชนี (Index) ของรูปภาพในฐานข้อมูล

3.1.2 เลือกวิธีการเปรียบเทียบ

เลือกวิธีการเปรียบเทียบรูปภาพ โดยโปรแกรมนี้จะมีวิธีการเปรียบเทียบรูปภาพให้เลือกใช้ 5 วิธี เพื่อเพิ่มประสิทธิภาพของโปรแกรม และยังได้วิเคราะห์วิธีการเปรียบเทียบว่าวิธีใดมีความสามารถในการค้นหารูปภาพได้ดีกว่า โดยวิธีการเปรียบเทียบมีให้เลือกใช้ 5 วิธี มีดังนี้

- Euclidean Distance
- Cosine Similarity
- Histogram Intersection
- Manhattan Distance
- Kullback-Leibler (K-L) Divergence

3.1.3 เปรียบเทียบค่าในฐานข้อมูล

การเปรียบเทียบค่าในฐานข้อมูลนั้นจะต้องนำค่าดัชนี (Index) ของรูปภาพต้นแบบที่ได้เลือกมาข้างต้น แล้วนำมาเปรียบเทียบกับค่าดัชนี (Index) รูปภาพในฐานข้อมูล โดยการคำนวณหาค่าระยะห่าง (Distance) ระหว่างรูปภาพต้นแบบกับรูปภาพในฐานข้อมูลตามวิธีการเปรียบเทียบที่ได้เลือกไว้ โดยจะนำค่าดัชนี (Index) มาเปรียบเทียบตามหลักคณิตศาสตร์ สามารถศึกษาหลักการและทฤษฎีได้จากหัวข้อ การเปรียบเทียบ ในบทที่ 2 และสามารถศึกษาวิธีการคำนวณทั้ง 5 วิธีได้จากตัวอย่าง ข้างล่างนี้

Euclidean Distance

$$d(P,Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots (p_n - q_n)} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

การคำนวณหาค่าระยะห่าง (Distance) โดยวิธี Euclidean Distance สามารถคำนวณได้ตามตัวอย่าง ดังนี้

สมมุติให้ P = {2 , 9 , 4} และ Q = {6 , 8 ,1} จากสมการดังกล่าว จะได้ว่า

$$d(P,Q) = \sqrt{(2-6)^2 + (9-8)^2 + (4-1)^2}$$

$$d(P,Q) = \sqrt{(-4)^2 + (1)^2 + (3)^2}$$

$$d(P,Q) = \sqrt{16+1+9}$$

$$d(P,Q) = \sqrt{26}$$
$$d(P,Q) = 5.099019514$$

ดังนั้นการคำนวณหาค่าระยะห่าง (Distance) โดยวิธี Euclidean Distance ระหว่างจุด P กับจุด Q มีค่าระยะห่าง (Distance) เท่ากับ 5.099019514

Cosine Similarity

$$d(P,Q) = \frac{\vec{p}_i \cdot \vec{q}_i}{|\vec{p}_i||\vec{q}_i|} = \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^{\,2}}\sqrt{\sum_{i=1}^{n} q_i^{\,2}}}$$

การคำนวณหาค่าความเหมือนโดยวิธี Cosine Similarity สามารถคำนวณได้ตามตัวอย่างดังนี้

สมมุติให้ P = {2 , 9 , 4} และ Q = {6 , 8 ,1} จากสมการดังกล่าว จะได้ว่า

$$d(P,Q) = \frac{(2\times 6)+(9\times 8)+(4\times 1)}{\sqrt{(2^2+9^2+4^2)}\times\sqrt{(6^2+8^2+1^2)}}$$
$$d(P,Q) = \frac{12+72+4}{\sqrt{101}\times\sqrt{101}}$$
$$d(P,Q) = \frac{88}{101}$$
$$d(P,Q) = 0.8712871287$$

ดังนั้นการคำนวณหาค่าความเหมือนโดยวิธี Cosine Similarity ระหว่างจุด P กับจุด Q มีค่าความเหมือน เท่ากับ 0.8712871287

Histogram Intersection

$$d(P,Q) = \frac{\sum_{i=1}^{n} \min(p_i, q_i)}{\sum_{i=1}^{n} q_i}$$

การคำนวณหาค่าความเหมือนโดยวิธี Histogram Intersection สามารถคำนวณได้ตามตัวอย่าง ดังนี้

สมมุติให้ P = {2 , 9 , 4} และ Q = {6 , 8 ,1} จากสมการดังกล่าว จะได้ว่า

$$d(P,Q) = \frac{\min(2,6) + \min(9,8) + \min(4,1)}{(6+8+1)}$$

$$d(P,Q) = \frac{2+8+1}{6+8+1}$$

$$d(P,Q) = \frac{11}{15}$$

$$d(P,Q) = 0.733333333$$

ดังนั้นการคำนวณหาค่าความเหมือน (Distance) โดยวิธี Histogram Intersection ระหว่างจุด P กับจุด Q มีค่าความเหมือน (Distance) เท่ากับ 0.733333333

Manhattan Distance

$$d(P,Q) = \sum_{i=1}^{n} |p_i - q_i|$$

การคำนวณหาค่าระยะห่าง (Distance) โดยวิธี Manhattan Distance สามารถคำนวณได้ตามตัวอย่าง ดังนี้

สมมุติให้ P = {2 , 9 , 4} และ Q = {6 , 8 ,1} จากสมการดังกล่าว จะได้ว่า

$$d(P,Q) = |2-6| + |9-8| + |4-1|$$

$$d(P,Q) = |-4| + |1| + |3|$$

$$d(P,Q) = 4+1+3$$

$$d(P,Q) = 8$$

ดังนั้นการคำนวณหาค่าระยะห่าง (Distance) โดยวิธี Manhattan Distance ระหว่างจุด P กับจุด Q มีค่าระยะห่าง (Distance) เท่ากับ 8

Kullback-Leibler (K-L) Divergence

$$d(P,Q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i}$$

การคำนวณหาค่าระยะห่าง (Distance) โดยวิธี Kullback-Leibler (K-L) Divergence สามารถคำนวณได้ตามตัวอย่าง ดังนี้

สมมุติให้ P = {2 , 9 , 4} และ Q = {6 , 8 ,1} จากสมการดังกล่าว จะได้ว่า

$$d(P,Q) = (2\log\frac{2}{6}) + (9\log\frac{9}{8}) + (4\log\frac{4}{1})$$

$$d(P,Q) = 2(-04771212547) + 9(0.0511525224) + 4(0.6020599913)$$

$$d(P,Q) = -0.9542425094 + 0.4603727016 + 2.408239965$$

$$d(P,Q) = 1.914370157$$

ดังนั้นการคำนวณหาค่าระยะห่าง (Distance) โดยวิธี Kullback-Leibler (K-L) Divergence ระหว่างจุด P กับจุด Q มีค่าระยะห่าง (Distance) เท่ากับ 1.914370157

3.1.4 นำค่าที่ได้จากการเปรียบเทียบมาเรียงลำดับ

เมื่อทำการเปรียบเทียบรูปภาพทุกรูปภาพในฐานข้อมูลแล้ว จะได้ค่าระยะห่าง (Distance) ของแต่ละรูปภาพออกมา โดยค่าระยะห่าง (Distance) ที่ได้ออกมามีความแตกต่างกันออกไปจึงทำการจัดเรียงลำดับจากน้อยไปหามากด้วยวิธี Quick Sort ซึ่งค่าระยะห่างที่มีค่าน้อยที่สุด คือค่าของรูปภาพในฐานข้อมูลที่มีความคล้ายคลึงมากที่สุด ยกเว้นการเปรียบเทียบโดยวิธี Cosine Similarity และวิธี Histogram Intersection จะทำการจัดเรียงลำดับจากมากไปหาน้อยด้วยวิธี Quick Sort ซึ่งค่าระยะห่างที่มีค่ามากที่สุด คือค่าของรูปภาพในฐานข้อมูลที่มีความคล้ายคลึงมากที่สุด

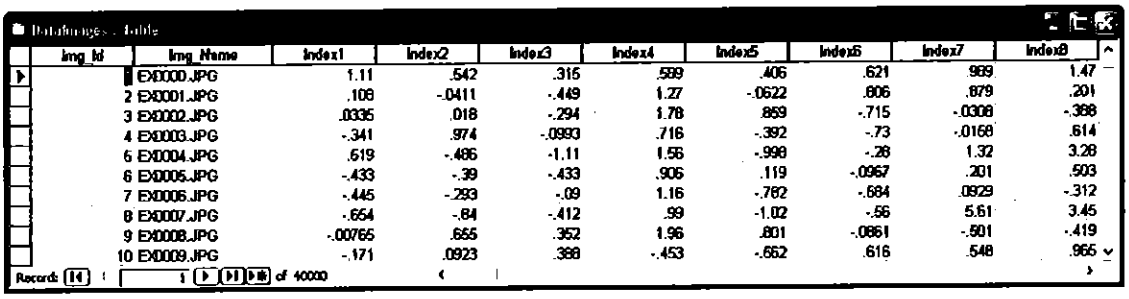
3.1.5 แสดงรูปภาพที่มีความคล้ายคลึงกับรูปภาพต้นแบบ

เมื่อได้ค่าระยะห่าง (Distance) ของรูปภาพต้นแบบกับรูปภาพในฐานข้อมูลที่ได้ทำการจัดเรียงลำดับค่าจากน้อยไปหามากแล้ว โปรแกรมจะทำการนำรูปภาพที่มีค่าระยะห่าง (Distance) น้อยที่สุด 20 อันดับแรกไปแสดงผล

3.2 การจัดการกับฐานข้อมูล

การจัดการกับฐานข้อมูล ขั้นแรกต้องหาวิธีการดึงข้อมูลดัชนี (Index) ของรูปภาพจาก MATLAB มาเก็บไว้ใน โปรแกรม Microsoft Office Access ก่อน เพราะการเก็บข้อมูลใน MATLAB ในส่วนของภาษา C#.NET ไม่สามารถทำการเชื่อมต่อได้โดยตรง ดังนั้นจึงนำข้อมูลส่วนที่อยู่ใน MATLAB มาเก็บไว้ในโปรแกรม Microsoft Office Excel ก่อน เนื่องจากไม่สามารถนำข้อมูลจาก MATLAB มาเก็บไว้ในโปรแกรม Microsoft Office Access ได้โดยตรง

จากนั้นนำข้อมูลจากโปรแกรม Microsoft Office Excel มาเก็บไว้ในโปรแกรม Microsoft Office Access โดยลำดับรูปภาพ ชื่อรูปภาพ และดัชนี (Index) ของรูปภาพ เก็บไว้ในตารางชื่อ DataImages โดยข้อมูลทั้งหมดจะถูกเก็บ 40000 แถว 115 คอลัมน์ ดังรูปที่ 3.2



img_id	img_Name	Index1	Index2	Index3	Index4	Index5	Index6	Index7	Index8
1	EX0000.JPG	1.11	.542	.315	.599	.406	.621	.969	1.47
2	EX0001.JPG	.108	-.0411	-.449	1.27	-.0622	.606	.879	.201
3	EX0002.JPG	.0335	.018	-.294	1.78	.859	-.715	-.0308	-.388
4	EX0003.JPG	-.341	.974	-.0993	.716	-.392	-.73	-.0168	.614
5	EX0004.JPG	.619	-.486	-1.11	1.56	-.998	-.28	1.32	3.28
6	EX0005.JPG	-.433	-.39	-.433	.906	.119	-.0967	.201	.503
7	EX0006.JPG	-.445	-.293	-.09	1.16	-.782	-.684	.0929	-.312
8	EX0007.JPG	-.654	-.84	-.412	.99	-1.02	-.56	5.61	3.45
9	EX0008.JPG	-.00765	.655	.352	1.96	.801	-.0861	-.501	-.419
10	EX0009.JPG	-.171	.0923	.388	-.453	-.662	.616	.548	.965

Record: 1 of 40000

รูปที่ 3.2 ตัวอย่างการเก็บข้อมูลทั้งหมดลงในฐานข้อมูล

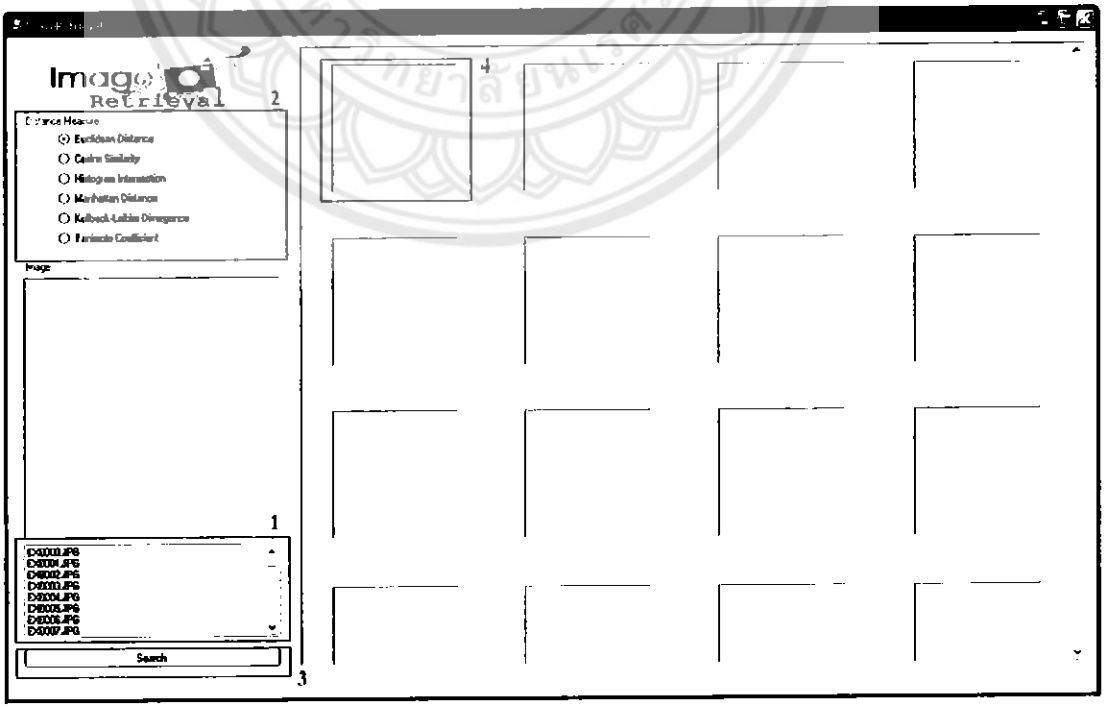
รูปที่ 3.3 แสดงตัวอย่างโปรแกรม โดยแบ่งออกเป็น 4 ส่วน ดังนี้

1. ส่วนที่ใช้เลือกรูปภาพต้นแบบ โดยมีรายชื่อรูปภาพให้เลือกรูปภาพที่ต้องการนำมาเป็นภาพต้นแบบ เพื่อใช้เป็นตัวอ้างอิงในการค้นหา

2. ส่วนที่ใช้เลือกวิธีในการเปรียบเทียบรูปภาพ

3. ส่วนที่ใช้ในการค้นหารูปภาพ โดยจะนำค่าดัชนีของรูปภาพต้นแบบไปเปรียบเทียบกับค่าดัชนีของรูปภาพในฐานข้อมูล ตามวิธีการเปรียบเทียบที่ได้เลือกไว้ แล้วนำรูปภาพที่มีความคล้ายคลึงกับรูปภาพต้นแบบมาแสดงผล

4. ส่วนที่ใช้แสดงผลของการค้นหารูปภาพ ซึ่งจะนำรูปภาพที่คล้ายคลึงกับรูปภาพต้นแบบมากที่สุด 20 ภาพ มาแสดงผล



รูปที่ 3.3 ตัวอย่างโปรแกรม

บทที่4

ผลการทดลอง

จากการที่ได้ทำการทดลองใช้งานโปรแกรมค้นหารูปภาพแล้ว เราจะทำการทดสอบผลการทดลองโดยใช้หลักการเปรียบเทียบผลของค่าความถูกต้อง (Precision) โดยค่าความถูกต้องเป็นค่าที่ได้จากการคำนวณดังต่อไปนี้

$$\% \text{ ความถูกต้อง} = (\text{จำนวนรูปภาพที่ใกล้เคียง}/\text{จำนวนรูปภาพทั้งหมด}) \times 100 \qquad 4.1$$

$$\text{เช่น } \% \text{ ความถูกต้อง} = \left(\frac{10}{20}\right) \times 100 = 50\,\%$$

แล้วนำค่าความถูกต้อง (Precision) มารวมกันแล้วหาค่าเฉลี่ย ก็จะได้เป็นค่าความถูกต้องของวิธีนั้นๆ ซึ่งเมื่อนำไปเปรียบเทียบกับวิธีการอื่นๆ ก็จะเห็นความแตกต่างของการค้นหาได้จากการสร้างกราฟ

โดยจะทำการเปรียบเทียบการค้นหารูปภาพ ดังนี้

1. การค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Euclidean Distance
2. การค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Cosine Similarity
3. การค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Histogram Intersection
4. การค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Manhattan Distance
5. การค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Kullback-Leibler Divergence

ซึ่งในการเปรียบเทียบแต่ละครั้งจะเปรียบเทียบกับรูปภาพที่เก็บไว้ในฐานข้อมูล Microsoft Office Access ซึ่งมีข้อมูลรูปภาพทั้งหมด 40,000 ภาพ

4.1 การค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Euclidean Distance

4.1.1 เลือกรูปภาพตัวอย่างที่ต้องการค้นหา

4.1.2 เลือกใช้วิธีการเปรียบเทียบ Euclidean Distance

4.1.3 โปรแกรมจะทำการเปรียบเทียบค่าดัชนี (Index) ระหว่าง รูปภาพตัวอย่างกับรูปภาพในฐานข้อมูล

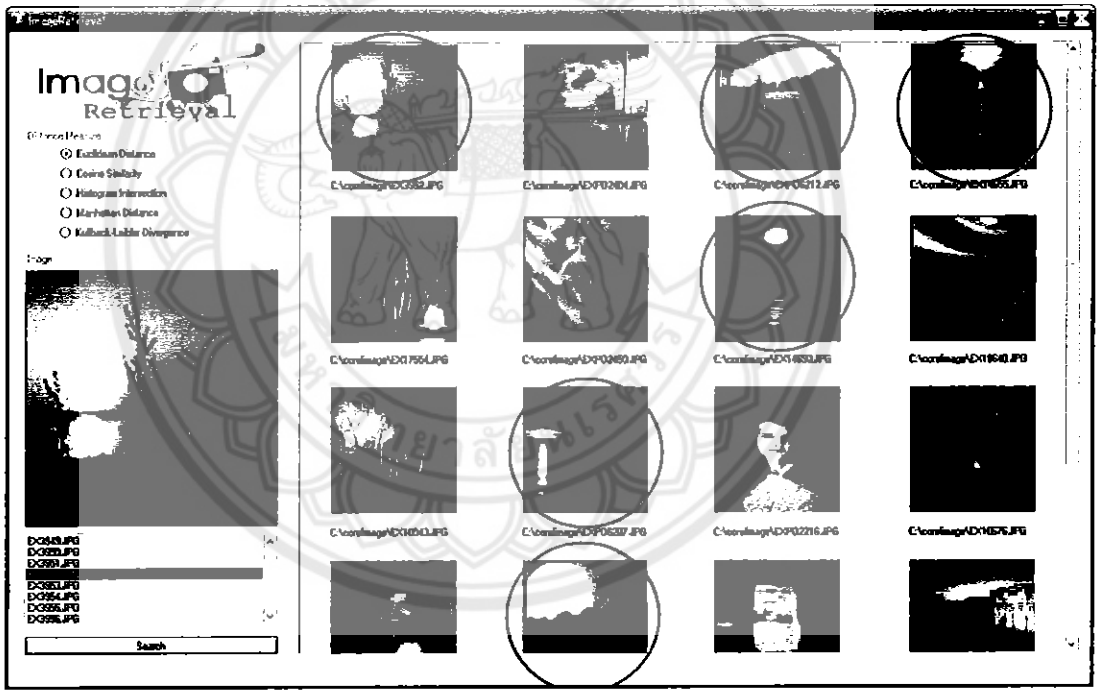
4.1.4 นำค่าระยะห่าง (Distance) มาเรียงจากน้อยไปมาก

4.1.5 แสดงผลการค้นหา ทั้งหมด 20 ภาพ โดยเรียงลำดับค่าระยะห่าง (Distance) จาก PIC1 ไปจนถึง PIC20

จากรูปที่ 4.1 เราหาค่าความถูกต้อง (Precision) ของการค้นหารูปภาพ จะเห็นได้ว่ารูปภาพที่ถูกต้องที่ถูกเลือกมา 6 ภาพ จากสมการ 4.1

$$\text{จะได้} \quad \%\,\text{ความถูกต้อง} = \left(\frac{6}{20}\right)\times 100 = 30\%$$

จากผลการทดลองทำการสุ่มตัวอย่างรูปภาพ 50 รูปภาพแล้วทำการค้นหาโดยใช้วิธีการเปรียบเทียบ Euclidean Distanceในการค้นหาแล้วบันทึกค่าผลการทดลองในตารางที่ 4.1



รูปที่ 4.1 แสดงผลการการค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Euclidean Distance

ตารางที่ 4.1 บันทึกค่าผลการทดลองการค้นหารูปภาพโดยใช้วิธี Euclidean Distance

ลำดับที่	ชื่อไฟล์รูปภาพ	จำนวนรูปภาพที่ถูกเลือก (จาก 20 ภาพ)	ค่าความถูกต้องการค้นหา Precision
1	EX0115.jpg	9	45%
2	EX0379.jpg	8	40%
3	EX0612.jpg	7	35%
4	EX1913.jpg	9	45%
5	EX2108.jpg	9	45%
6	EX3019.jpg	9	45%
7	EX3614.jpg	13	65%
8	EX3952.jpg	6	30%
9	EX3955.jpg	9	45%
10	EX4936.jpg	5	25%
11	EX5042.jpg	9	45%
12	EX7229.jpg	18	90%
13	EX7807.jpg	13	65%
14	EX8246.jpg	10	50%
15	EX9868.jpg	8	40%
16	EX10122.jpg	8	40%
17	EX10403.jpg	17	85%
18	EX11517.jpg	15	75%
19	EX12344.jpg	9	45%
20	EX13072.jpg	10	50%
21	EX14029.jpg	8	40%
22	EX15309.jpg	5	25%
23	EX15419.jpg	7	35%
24	EX19992.jpg	10	50%
25	EXPO0187.jpg	8	40%
26	EXPO0661.jpg	10	50%

ลำดับที่	ชื่อไฟล์รูปภาพ	จำนวนรูปภาพที่ถูกเลือก (จาก 20 ภาพ)	ค่าความถูกต้องการค้นหา Precision
27	EXPO1911.jpg	10	50%
28	EXPO2700.jpg	17	85%
29	EXPO4351.jpg	9	45%
30	EXPO4508.jpg	15	75%
31	EXPO5515.jpg	13	65%
32	EXPO5532.jpg	12	60%
33	EXPO5817.jpg	8	40%
34	EXPO6306.jpg	19	95%
35	EXPO7082.jpg	12	60%
36	EXPO9905.jpg	20	100%
37	EXPO10016.jpg	12	60%
38	EXPO10168.jpg	7	35%
39	EXPO10335.jpg	11	55%
40	EXPO11772.jpg	6	30%
41	EXPO12720.jpg	6	30%
42	EXPO13205.jpg	8	40%
43	EXPO13310.jpg	18	90%
44	EXPO13540.jpg	13	65%
45	EXPO14081.jpg	8	40%
46	EXPO14829.jpg	7	35%
47	EXPO14973.jpg	18	90%
48	EXPO15008.jpg	14	70%
49	EXPO15291.jpg	14	70%
50	EXPO17261.jpg	6	30%
SUMATION		532	2660%
AVERAGE		10.64	53.2%

4.2 การค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Cosine Similarity

4.2.1 เลือกรูปภาพตัวอย่างที่ต้องการค้นหา

4.2.2 เลือกใช้วิธีการเปรียบเทียบ Cosine Similarity

4.2.3 โปรแกรมจะทำการเปรียบเทียบค่าดัชนี (Index) ระหว่าง รูปภาพตัวอย่างกับรูปภาพในฐานข้อมูล

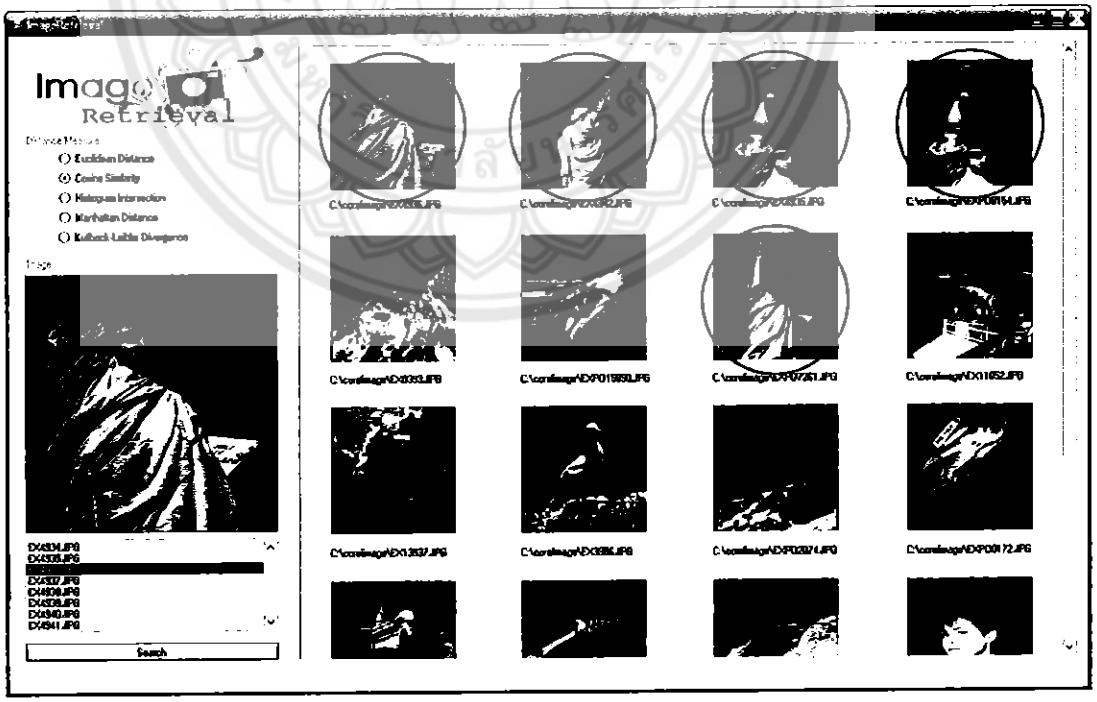
4.2.4 นำค่าความเหมือน มาเรียงจากมากไปน้อย

4.2.5 แสดงผลการค้นหา ทั้งหมด 20 ภาพ โดยเรียงลำดับ ค่าความเหมือนจาก PIC1 ไปจนถึง PIC20

จากรูปที่ 4.2 เราหาค่าความถูกต้อง (Precision) ของการค้นหารูปภาพ จะเห็นได้ว่ารูปภาพที่ถูกต้องที่ถูกเลือกมา 5 ภาพ จากสมการ 4.1

$$\text{จะได้} \quad \%\ \text{ความถูกต้อง} = \left(\frac{5}{20}\right) \times 100 = 25\%$$

จากผลการทดลองทำการสุ่มตัวอย่างรูปภาพ 50 รูปภาพแล้วทำการค้นหาโดยใช้วิธีการเปรียบเทียบ Cosine Similarity ในการค้นหาแล้วบันทึกค่าผลการทดลองในตารางที่ 4.2



รูปที่ 4.2 แสดงผลการการค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Cosine Similarity

ตารางที่ 4.2 บันทึกค่าผลการทดลองการค้นหารูปภาพโดยใช้วิธี Cosine Similarity

ลำดับที่	ชื่อไฟล์รูปภาพ	จำนวนรูปภาพที่ถูกเลือก (จาก 20 ภาพ)	ค่าความถูกต้องการค้นหา Precision
1	EX0115.jpg	8	40%
2	EX0379.jpg	8	40%
3	EX0612.jpg	8	40%
4	EX1913.jpg	12	60%
5	EX2108.jpg	3	15%
6	EX3019.jpg	8	40%
7	EX3614.jpg	12	60%
8	EX3952.jpg	18	90%
9	EX3955.jpg	4	20%
10	EX4936.jpg	5	25%
11	EX5042.jpg	12	60%
12	EX7229.jpg	18	90%
13	EX7807.jpg	11	55%
14	EX8246.jpg	19	45%
15	EX9868.jpg	18	40%
16	EX10122.jpg	7	35%
17	EX10403.jpg	13	65%
18	EX11517.jpg	13	65%
19	EX12344.jpg	14	70%
20	EX13072.jpg	8	40%
21	EX14029.jpg	6	30%
22	EX15309.jpg	6	30%
23	EX15419.jpg	7	35%
24	EX19992.jpg	10	50%
25	EXPO0187.jpg	7	35%
26	EXPO0661.jpg	11	55%

ปร.
ชพ๒๓
2550 ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร 15005383

ลำดับที่	ชื่อไฟล์รูปภาพ	จำนวนรูปภาพที่ถูกเลือก (จาก 20 ภาพ)	ค่าความถูกต้องการค้นหา Precision
27	EXPO1911.jpg	6	30%
28	EXPO2700.jpg	16	80%
29	EXPO4351.jpg	10	50%
30	EXPO4508.jpg	16	80%
31	EXPO5515.jpg	16	80%
32	EXPO5532.jpg	18	90%
33	EXPO5817.jpg	8	40%
34	EXPO6306.jpg	20	100%
35	EXPO7082.jpg	12	60%
36	EXPO9905.jpg	20	100%
37	EXPO10016.jpg	11	55%
38	EXPO10168.jpg	5	25%
39	EXPO10335.jpg	11	55%
40	EXPO11772.jpg	6	30%
41	EXPO12720.jpg	5	25%
42	EXPO13205.jpg	12	60%
43	EXPO13310.jpg	18	90%
44	EXPO13540.jpg	15	75%
45	EXPO14081.jpg	7	35%
46	EXPO14829.jpg	7	35%
47	EXPO14973.jpg	18	90%
48	EXPO15008.jpg	17	85%
49	EXPO15291.jpg	12	60%
50	EXPO17261.jpg	6	30%
SUMATION		568	2690%
AVERAGE		11.36	53.8%

4.3 การค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Histogram Intersection

4.3.1 เลือกรูปภาพตัวอย่างที่ต้องการค้นหา

4.3.2 เลือกใช้วิธีการเปรียบเทียบ Histogram Intersection

4.3.3 โปรแกรมจะทำการเปรียบเทียบค่าดัชนี (Index) ระหว่าง รูปภาพตัวอย่างกับรูปภาพในฐานข้อมูล

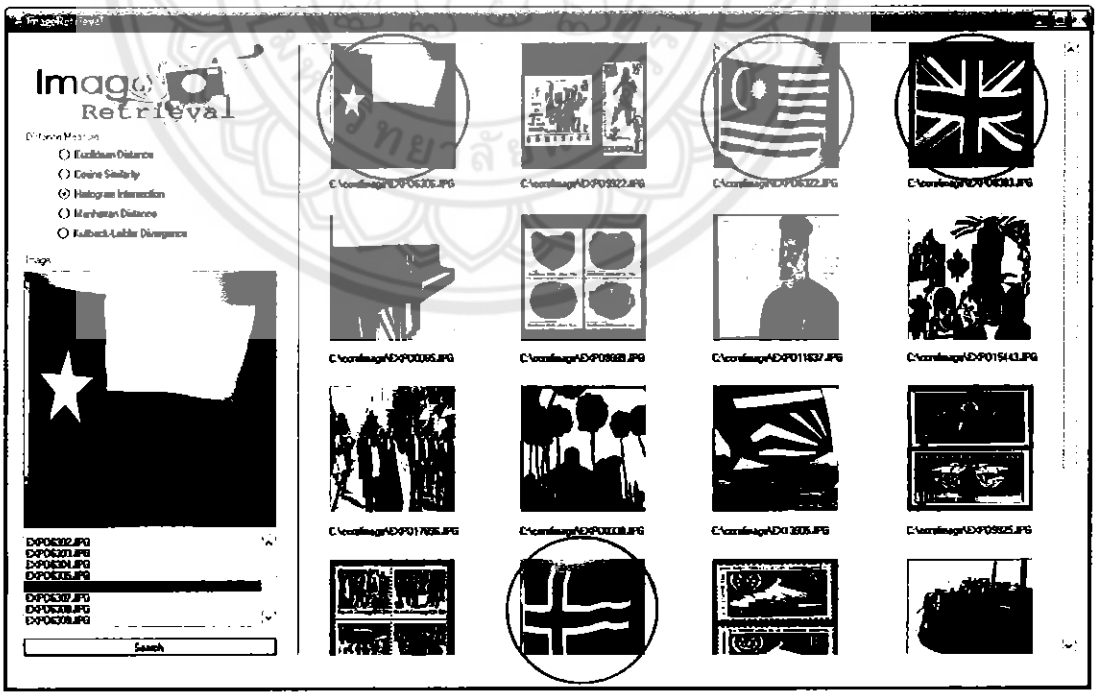
4.3.4 นำค่าความเหมือน มาเรียงจากมากไปน้อย

4.3.5 แสดงผลการค้นหา ทั้งหมด 20 ภาพ โดยเรียงลำดับ ค่าความเหมือนจาก PIC1 ไปจนถึง PIC20

จากรูปที่ 4.3 เราหาค่าความถูกต้อง (Precision) ของการค้นหารูปภาพ จะเห็นได้ว่ารูปภาพที่ถูกต้องที่ถูกเลือกมา 3 ภาพ จากสมการ 4.1

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

จากผลการทดลองทำการสุ่มตัวอย่างรูปภาพ 50 รูปภาพแล้วทำการค้นหาโดยใช้วิธีการเปรียบเทียบ Histogram Intersection ในการค้นหาแล้วบันทึกค่าผลการทดลองในตารางที่ 4.3



รูปที่ 4.3 แสดงผลการการค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Histogram Intersection

ตารางที่ 4.3 บันทึกค่าผลการทดลองการค้นหารูปภาพโดยใช้วิธี Histogram Intersection

ลำดับที่	ชื่อไฟล์รูปภาพ	จำนวนรูปภาพที่ถูกเลือก (จาก 20 ภาพ)	ค่าความถูกต้องการค้นหา Precision
1	EX0115.jpg	2	10%
2	EX0379.jpg	2	10%
3	EX0612.jpg	1	5%
4	EX1913.jpg	4	20%
5	EX2108.jpg	4	20%
6	EX3019.jpg	3	15%
7	EX3614.jpg	5	25%
8	EX3952.jpg	1	5%
9	EX3955.jpg	3	15%
10	EX4936.jpg	1	5%
11	EX5042.jpg	8	40%
12	EX7229.jpg	6	30%
13	EX7807.jpg	12	60%
14	EX8246.jpg	20	100%
15	EX9868.jpg	3	15%
16	EX10122.jpg	1	5%
17	EX10403.jpg	1	5%
18	EX11517.jpg	2	10%
19	EX12344.jpg	2	10%
20	EX13072.jpg	1	5%
21	EX14029.jpg	4	20%
22	EX15309.jpg	1	5%
23	EX15419.jpg	2	10%
24	EX19992.jpg	3	15%
25	EXPO0187.jpg	2	10%
26	EXPO0661.jpg	4	20%

ลำดับที่	ชื่อไฟล์รูปภาพ	จำนวนรูปภาพที่ถูกเลือก (จาก 20 ภาพ)	ค่าความถูกต้องการค้นหา Precision
27	EXPO1911.jpg	1	5%
28	EXPO2700.jpg	11	55%
29	EXPO4351.jpg	4	20%
30	EXPO4508.jpg	5	25%
31	EXPO5515.jpg	7	35%
32	EXPO5532.jpg	5	25%
33	EXPO5817.jpg	1	5%
34	EXPO6306.jpg	4	20%
35	EXPO7082.jpg	4	20%
36	EXPO9905.jpg	20	100%
37	EXPO10016.jpg	5	25%
38	EXPO10168.jpg	2	10%
39	EXPO10335.jpg	11	55%
40	EXPO11772.jpg	7	35%
41	EXPO12720.jpg	1	5%
42	EXPO13205.jpg	9	45%
43	EXPO13310.jpg	7	35%
44	EXPO13540.jpg	7	35%
45	EXPO14081.jpg	4	20%
46	EXPO14829.jpg	2	10%
47	EXPO14973.jpg	5	25%
48	EXPO15008.jpg	6	30%
49	EXPO15291.jpg	10	50%
50	EXPO17261.jpg	1	5%
SUMATION		237	1185%
AVERAGE		4.74	23.7%

4.4 การค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Manhattan Distance

4.4.1 เลือกรูปภาพตัวอย่างที่ต้องการค้นหา

4.4.2 เลือกใช้วิธีการเปรียบเทียบ Manhattan Distance

4.4.3 โปรแกรมจะทำการเปรียบเทียบค่าดัชนี (Index) ระหว่าง รูปภาพตัวอย่างกับรูปภาพในฐานข้อมูล

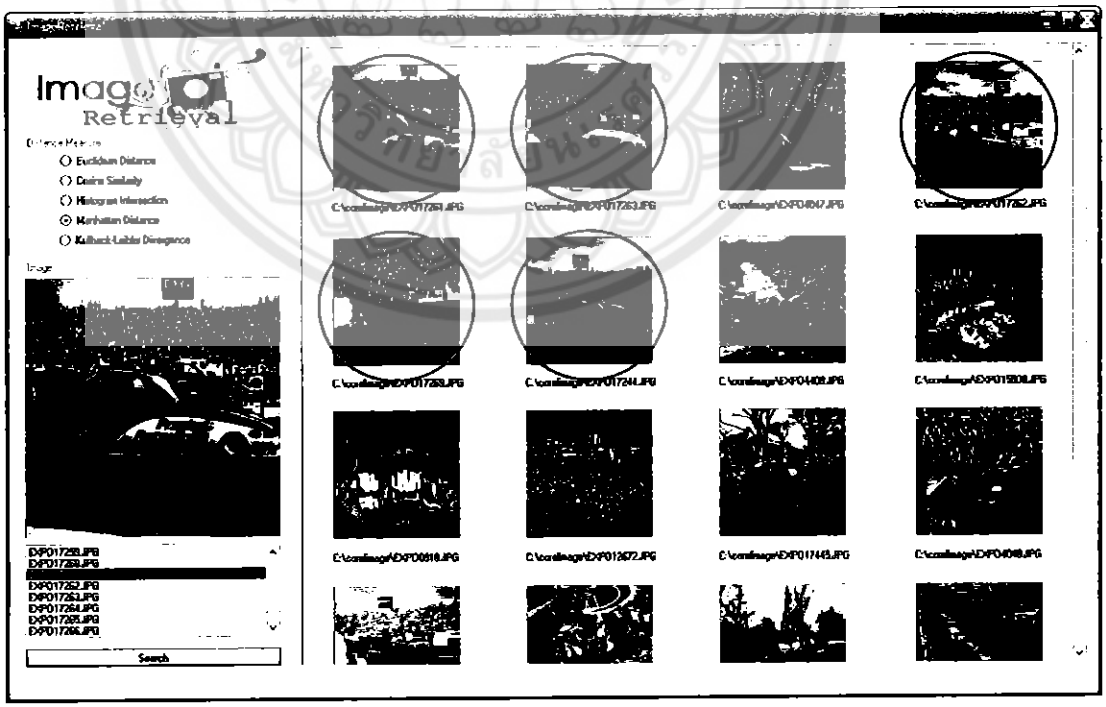
4.4.4 นำค่าระยะห่าง (Distance) มาเรียงจากน้อยไปมาก

4.4.5 แสดงผลการค้นหา ทั้งหมด 20 ภาพ โดยเรียงลำดับค่าระยะห่าง (Distance) จาก PIC1 ไปจนถึง PIC20

จากรูปที่ 4.4 เราหาค่าความถูกต้อง (Precision) ของการค้นหารูปภาพ จะเห็นได้ว่ารูปภาพที่ถูกต้องที่ถูกเลือกมา 5 ภาพ จากสมการ 4.1

$$\text{จะได้} \quad \%\ \text{ความถูกต้อง} = \left(\frac{5}{20}\right) \times 100 = 25\%$$

จากผลการทดลองทำการสุ่มตัวอย่างรูปภาพ 50 รูปภาพแล้วทำการค้นหาโดยใช้วิธีการเปรียบเทียบ Manhattan Distance ในการค้นหาแล้วบันทึกค่าผลการทดลองในตารางที่ 4.4



รูปที่ 4.4 แสดงผลการการค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Manhattan Distance

ตารางที่ 4.4 บันทึกค่าผลการทดลองการค้นหารูปภาพโดยใช้วิธี Manhattan Distance

ลำดับที่	ชื่อไฟล์รูปภาพ	จำนวนรูปภาพที่ถูกเลือก (จาก 20 ภาพ)	ค่าความถูกต้องการค้นหา Precision
1	EX0115.jpg	9	45%
2	EX0379.jpg	7	35%
3	EX0612.jpg	11	55%
4	EX1913.jpg	10	50%
5	EX2108.jpg	8	40%
6	EX3019.jpg	10	50%
7	EX3614.jpg	12	60%
8	EX3952.jpg	9	45%
9	EX3955.jpg	7	35%
10	EX4936.jpg	6	30%
11	EX5042.jpg	12	60%
12	EX7229.jpg	19	95%
13	EX7807.jpg	12	60%
14	EX8246.jpg	8	40%
15	EX9868.jpg	8	40%
16	EX10122.jpg	9	45%
17	EX10403.jpg	17	85%
18	EX11517.jpg	15	75%
19	EX12344.jpg	15	75%
20	EX13072.jpg	8	40%
21	EX14029.jpg	7	35%
22	EX15309.jpg	7	35%
23	EX15419.jpg	7	35%
24	EX19992.jpg	11	55%
25	EXPO0187.jpg	5	25%
26	EXPO0661.jpg	10	50%

ลำดับที่	ชื่อไฟล์รูปภาพ	จำนวนรูปภาพที่ถูกเลือก (จาก 20 ภาพ)	ค่าความถูกต้องการค้นหา Precision
27	EXPO1911.jpg	13	65%
28	EXPO2700.jpg	19	95%
29	EXPO4351.jpg	11	55%
30	EXPO4508.jpg	16	80%
31	EXPO5515.jpg	16	80%
32	EXPO5532.jpg	13	65%
33	EXPO5817.jpg	10	50%
34	EXPO6306.jpg	19	95%
35	EXPO7082.jpg	13	65%
36	EXPO9905.jpg	20	100%
37	EXPO10016.jpg	8	40%
38	EXPO10168.jpg	7	35%
39	EXPO10335.jpg	11	55%
40	EXPO11772.jpg	8	40%
41	EXPO12720.jpg	7	35%
42	EXPO13205.jpg	8	40%
43	EXPO13310.jpg	17	85%
44	EXPO13540.jpg	16	80%
45	EXPO14081.jpg	7	35%
46	EXPO14829.jpg	6	30%
47	EXPO14973.jpg	18	90%
48	EXPO15008.jpg	15	75%
49	EXPO15291.jpg	14	70%
50	EXPO17261.jpg	5	25%
SUMATION		556	2780%
AVERAGE		11.12	55.6%

4.5 การค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Kullback-Leibler Divergence

4.5.1 เลือกรูปภาพตัวอย่างที่ต้องการค้นหา

4.5.2 เลือกใช้วิธีการเปรียบเทียบ Kullback-Leibler Divergence

4.5.3 โปรแกรมจะทำการเปรียบเทียบค่าดัชนี (Index) ระหว่าง รูปภาพตัวอย่างกับรูปภาพในฐานข้อมูล

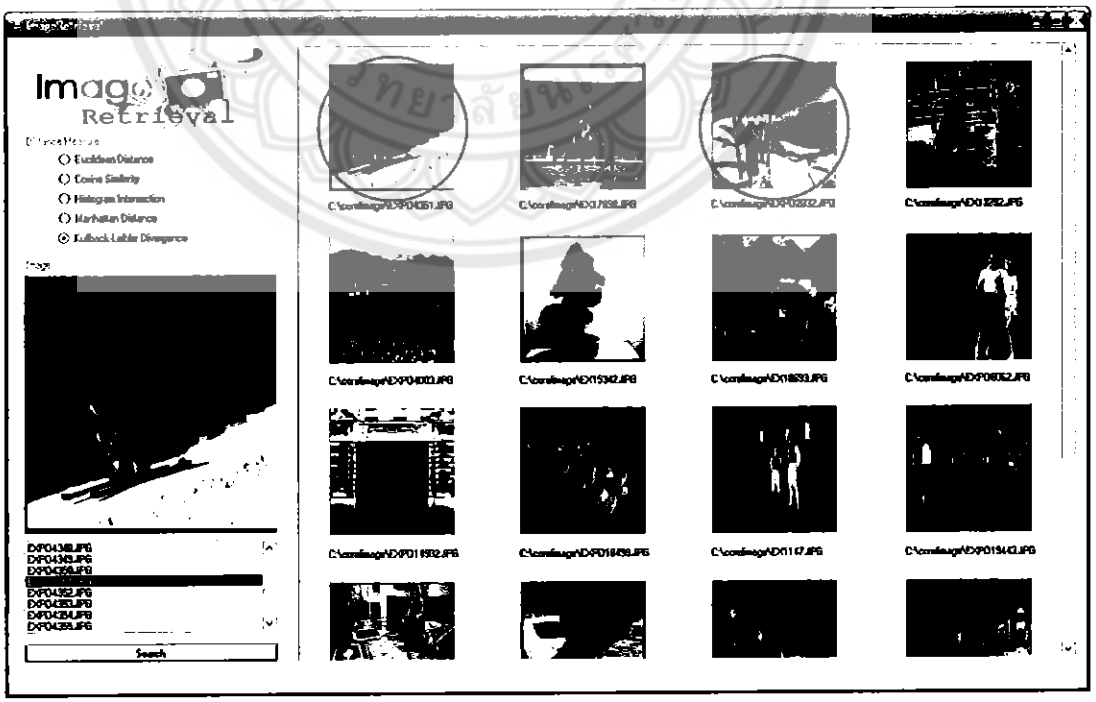
4.5.4 นำค่าความเหมือน มาเรียงจากน้อยไปมาก

4.5.5 แสดงผลการค้นหา ทั้งหมด 20 ภาพ โดยเรียงลำดับ ค่าความเหมือน จาก PIC1 ไปจนถึง PIC20

จากรูปที่ 4.5 เราหาค่าความถูกต้อง (Precision) ของการค้นหารูปภาพ จะเห็นได้ว่ารูปภาพที่ถูกต้องที่ถูกเลือกมา 2 ภาพ จากสมการ 4.1

$$\text{จะได้} \quad \%\,\text{ความถูกต้อง} = \left(\frac{2}{20}\right) \times 100 = 10\%$$

จากผลการทดลองทำการสุ่มตัวอย่างรูปภาพ 50 รูปภาพแล้วทำการค้นหาโดยใช้วิธีการเปรียบเทียบ Kullback-Leibler Divergence ในการค้นหาแล้วบันทึกค่าผลการทดลองในตารางที่ 4.5



รูปที่ 4.5 แสดงผลการการค้นหารูปภาพโดยใช้วิธีการเปรียบเทียบ Kullback-Leibler Divergence

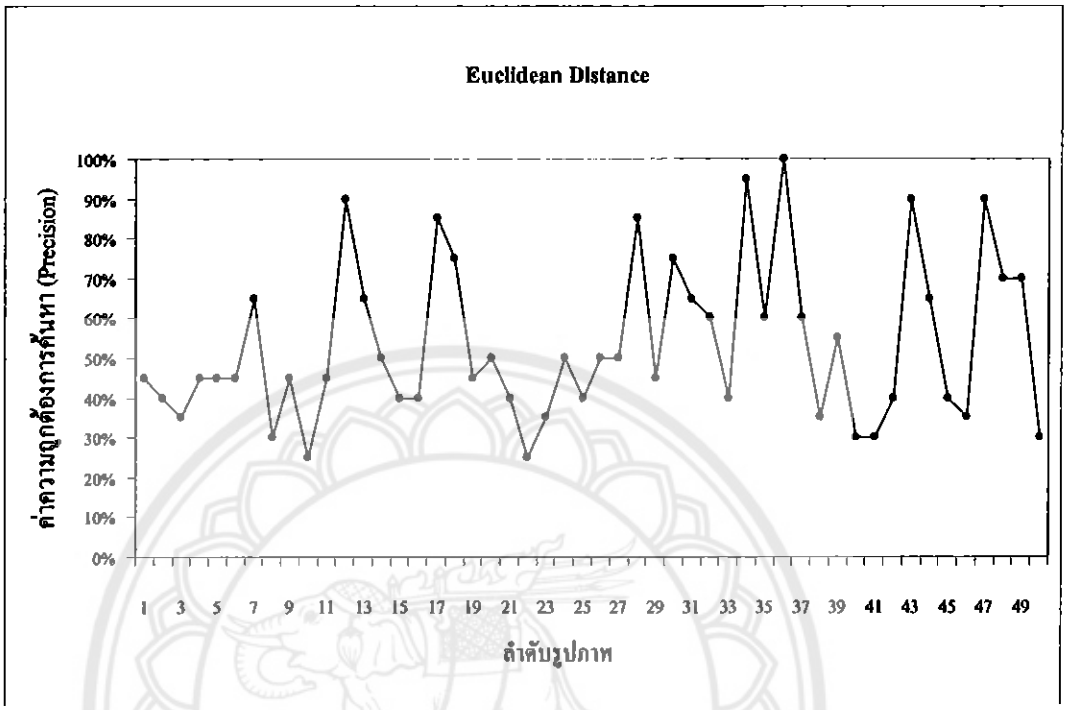
ตารางที่ 4.5 บันทึกค่าผลการทดลองการค้นหารูปภาพโดยใช้วิธี Kullback-Leibler Divergence

ลำดับที่	ชื่อไฟล์รูปภาพ	จำนวนรูปภาพที่ถูกเลือก (จาก 20 ภาพ)	ค่าความถูกต้องการค้นหา Precision
1	EX0115.jpg	1	5%
2	EX0379.jpg	2	10%
3	EX0612.jpg	1	5%
4	EX1913.jpg	2	10%
5	EX2108.jpg	2	10%
6	EX3019.jpg	6	30%
7	EX3614.jpg	1	5%
8	EX3952.jpg	1	5%
9	EX3955.jpg	1	5%
10	EX4936.jpg	1	5%
11	EX5042.jpg	1	5%
12	EX7229.jpg	4	20%
13	EX7807.jpg	1	5%
14	EX8246.jpg	3	15%
15	EX9868.jpg	1	5%
16	EX10122.jpg	1	5%
17	EX10403.jpg	1	5%
18	EX11517.jpg	1	5%
19	EX12344.jpg	1	5%
20	EX13072.jpg	1	5%
21	EX14029.jpg	2	10%
22	EX15309.jpg	1	5%
23	EX15419.jpg	1	5%
24	EX19992.jpg	1	5%
25	EXPO0187.jpg	4	20%
26	EXPO0661.jpg	2	10%

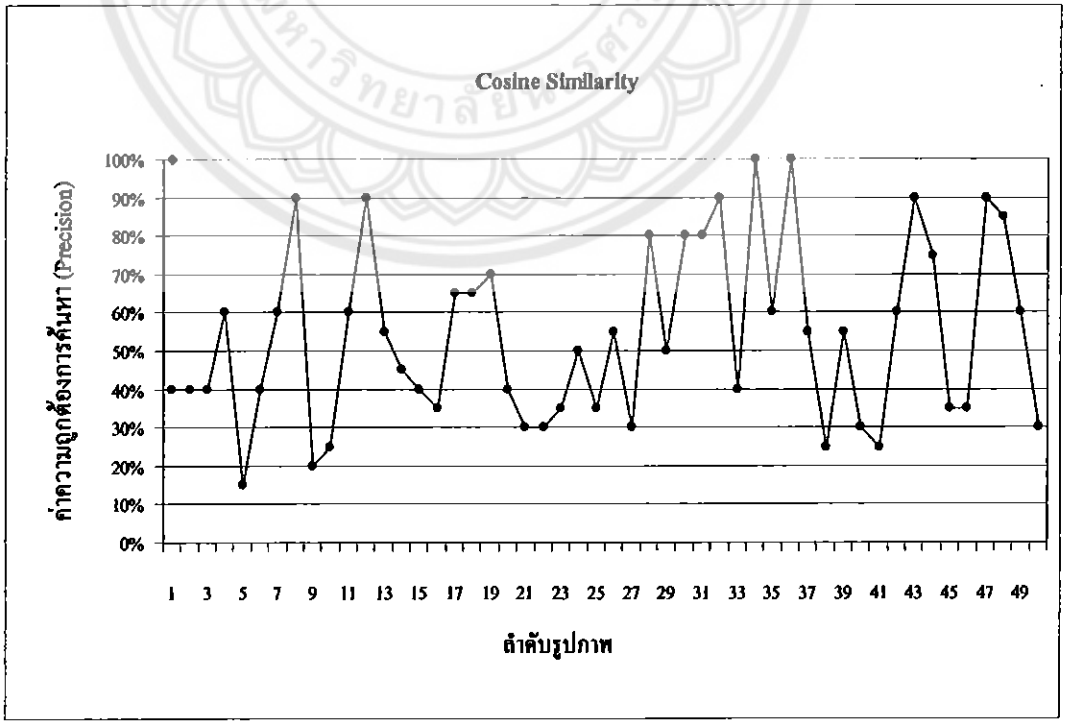
ลำดับที่	ชื่อไฟล์รูปภาพ	จำนวนรูปภาพที่ถูกเลือก (จาก 20 ภาพ)	ค่าความถูกต้องการค้นหา Precision
27	EXPO1911.jpg	1	5%
28	EXPO2700.jpg	9	45%
29	EXPO4351.jpg	2	10%
30	EXPO4508.jpg	1	5%
31	EXPO5515.jpg	1	5%
32	EXPO5532.jpg	2	10%
33	EXPO5817.jpg	1	5%
34	EXPO6306.jpg	1	5%
35	EXPO7082.jpg	1	5%
36	EXPO9905.jpg	20	100%
37	EXPO10016.jpg	1	5%
38	EXPO10168.jpg	1	5%
39	EXPO10335.jpg	1	5%
40	EXPO11772.jpg	3	15%
41	EXPO12720.jpg	1	5%
42	EXPO13205.jpg	8	40%
43	EXPO13310.jpg	3	15%
44	EXPO13540.jpg	1	5%
45	EXPO14081.jpg	1	5%
46	EXPO14829.jpg	2	10%
47	EXPO14973.jpg	4	20%
48	EXPO15008.jpg	1	5%
49	EXPO15291.jpg	1	5%
50	EXPO17261.jpg	1	5%
SUMATION		112	560%
AVERAGE		2.24	11.2%

4.6 กราฟแสดงค่าความถูกต้องของการค้นหาทั้ง 5 วิธีจากผลการทดลอง

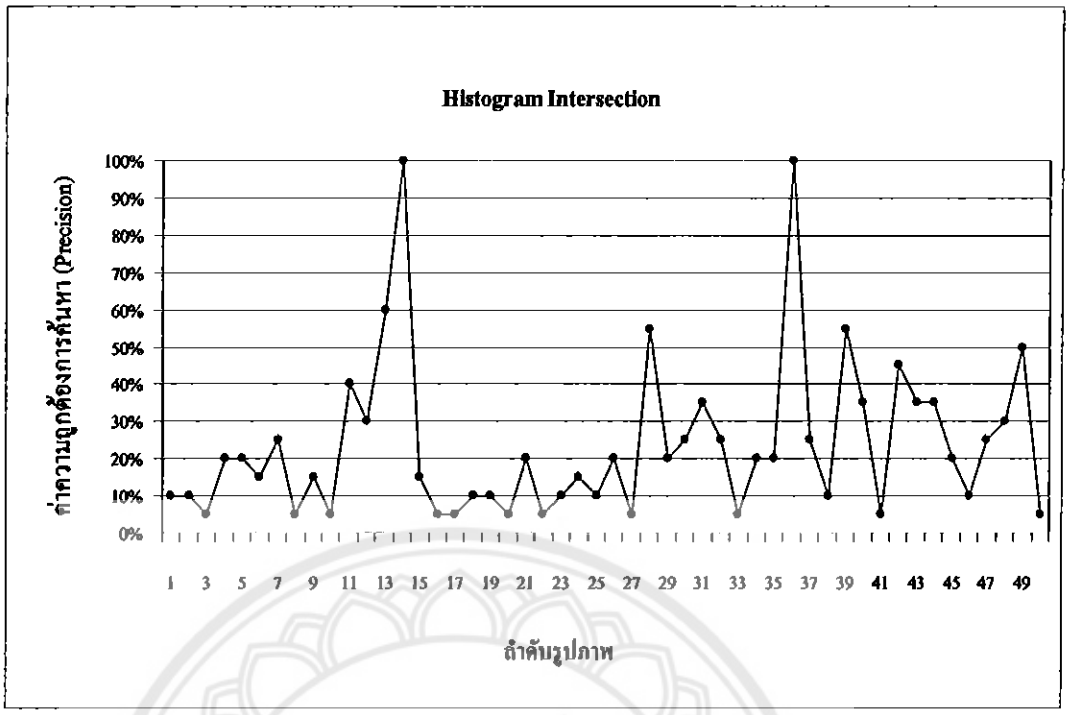
จากตาราง 4.1 ถึง 4.5 เมื่อนำมาพล็อตกราฟจะได้ดังรูปกราฟที่ 4.6 ถึง 4.10 ตามลำดับ



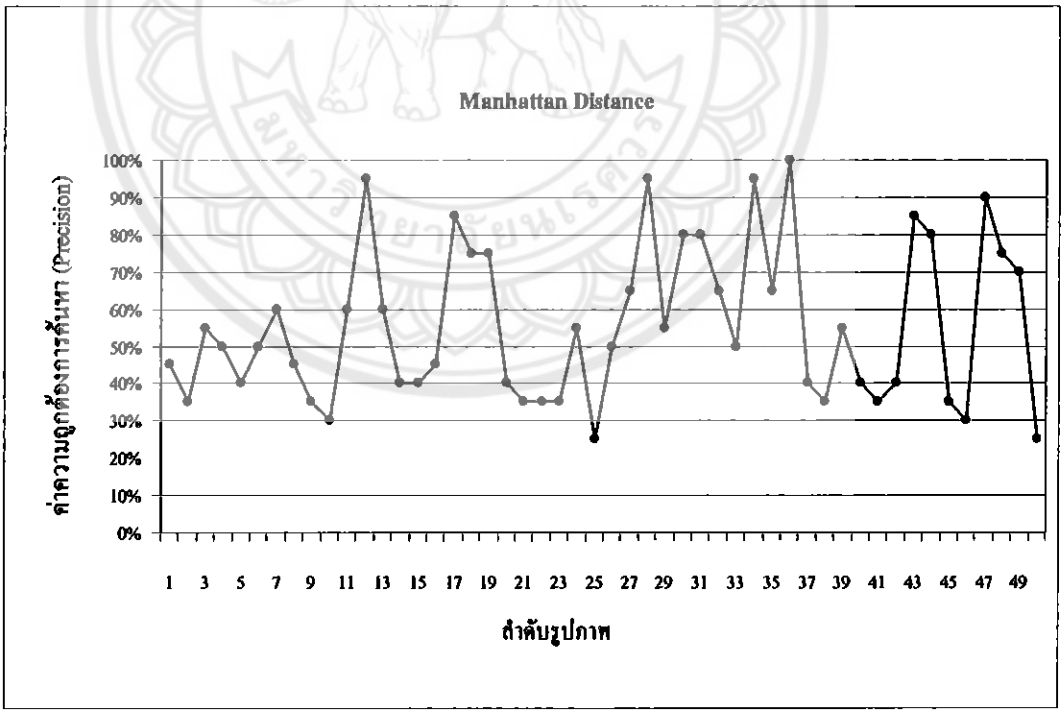
รูปที่ 4.6 รูปกราฟแสดงค่าความถูกต้องของการค้นหารูปภาพโดยใช้ Euclidean Distance



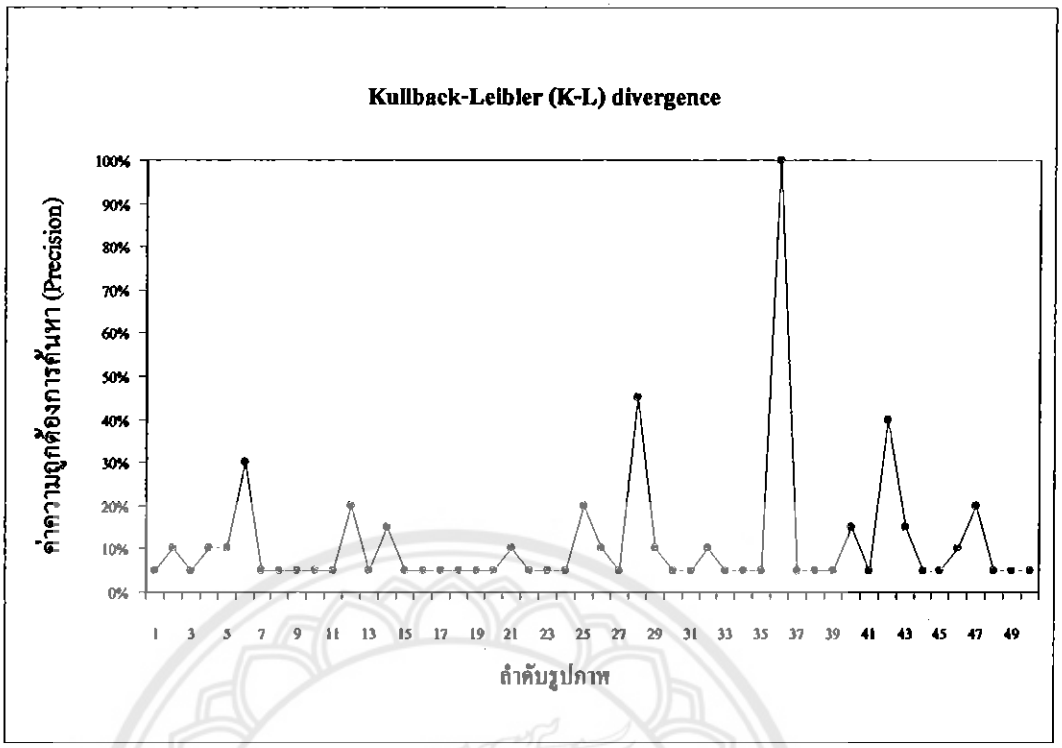
รูปที่ 4.7 รูปกราฟแสดงค่าความถูกต้องของการค้นหารูปภาพโดยใช้ Cosine Similarity



รูปที่ 4.8 รูปกราฟแสดงค่าความถูกต้องของการค้นหารูปภาพโดยใช้ Histogram Intersection



รูปที่ 4.9 รูปกราฟแสดงค่าความถูกต้องของการค้นหารูปภาพโดยใช้ Manhattan Distance



รูปที่ 4.10 รูปกราฟแสดงค่าความถูกต้องของการค้นหารูปภาพโดยใช้ Kullback-Leibler

4.7 เปรียบเทียบผลการทดลอง

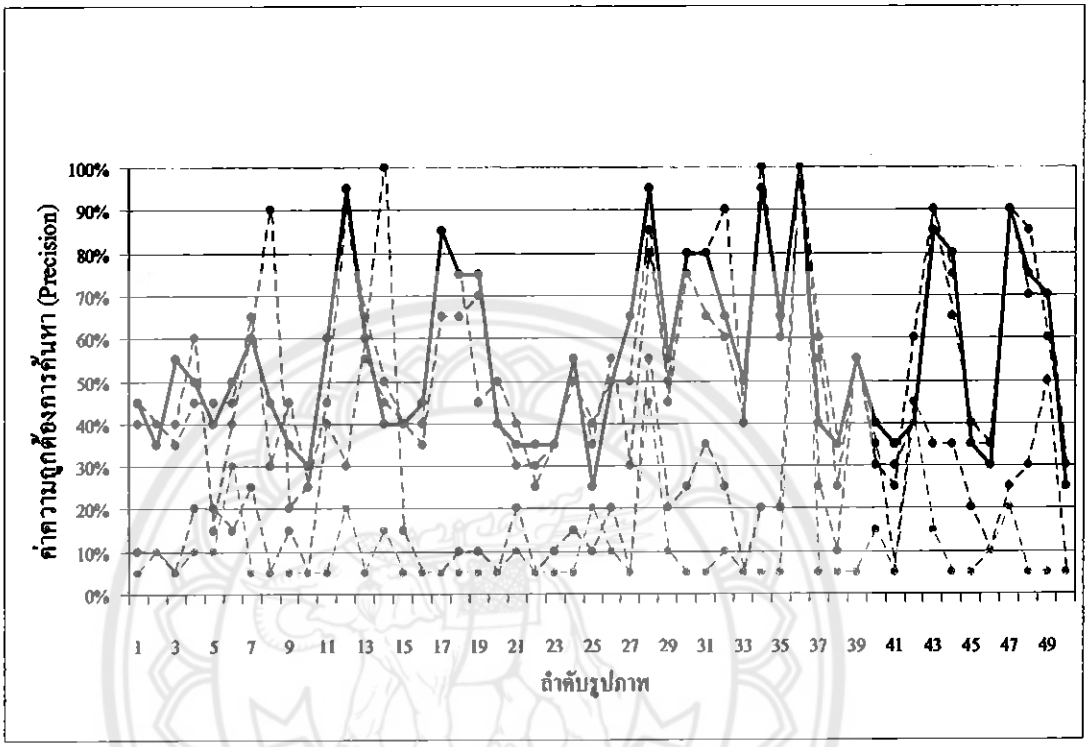
เมื่อทำการค้นหาและบันทึกผลแล้ว จึงนำผลการทดลองหาค่าความถูกต้องของการค้นหา (Precision) ของรูปภาพที่สุ่มตัวอย่างแต่ละรูปภาพมาทำการพล็อตเป็นกราฟเปรียบเทียบการค้นหาทั้ง 5 วิธี จะเห็นความแตกต่างของการทดลองได้จากรูปกราฟที่ 4.11

เมื่อเราพิจารณาและเปรียบเทียบค่าเฉลี่ยของความถูกต้องของการค้นหา(Precision) ในแต่ละวิธีจะได้ดังตารางที่ 4.6

Methods	Average Precision
Euclidean Distance	53.2%
Cosine Similarity	53.8%
Histogram Intersection	23.7%
Manhattan Distance	55.6%
Kullback-Leibler Divergence	11.2%

ตารางที่ 4.6 เปรียบเทียบค่า Average Precision ของ 5 วิธี

เมื่อสังเกตผลการเปรียบเทียบค่าความถูกต้องของการค้นหาแล้ว จะเห็นความแตกต่างของทั้ง 5 วิธี ได้อย่างชัดเจนว่าค่าความถูกต้องของการค้นหาโดยใช้วิธี Manhattan Distance สามารถค้นหารูปภาพและองค์ประกอบที่ใกล้เคียงกันได้มากกว่าการค้นหารูปภาพโดยใช้วิธีอื่นๆ



รูปที่ 4.11 รูปกราฟเปรียบเทียบค่าความถูกต้องการค้นหาของทั้ง 5 วิธี

---◆--- Euclidean Distance

---◆--- Cosine Similarity

---◆--- Histogram Intersection

—◆— Manhattan Distance

◆··· Kullback-Leibler (K-L) divergence

บทที่ 5
บทสรุป

5.1 สรุปผลการทดลอง

จากการเปรียบเทียบการทดลองทั้งหมด จะเห็นได้ว่าค่าเฉลี่ยของความถูกต้องของการค้นหา (Average Precision) โดยวิธี Manhattan Distance จะมีค่ามากที่สุด คือ 55.6% ดังนั้นการค้นหารูปภาพโดยวิธี Manhattan Distance จะสามารถค้นหารูปภาพและองค์ประกอบที่ใกล้เคียงกันได้มากที่สุด รองลงมาคือ การค้นหารูปภาพโดยวิธี Cosine Similarity มีค่าเฉลี่ยของความถูกต้องของการค้นหา (Average Precision) เท่ากับ 53.8% รองลงมาคือ การค้นหารูปภาพโดยวิธี Euclidean Distance มีค่าเฉลี่ยของความถูกต้องของการค้นหา (Average Precision) เท่ากับ 53.2% รองลงมาคือ การค้นหารูปภาพโดยวิธี Histogram Intersection มีค่าเฉลี่ยของความถูกต้องของการค้นหา (Average Precision) เท่ากับ 23.7% และน้อยที่สุดคือ การค้นหารูปภาพโดยวิธี Kullback-Leibler Divergence มีค่าเฉลี่ยของความถูกต้องของการค้นหา (Average Precision) เท่ากับ 11.2%

ดังนั้นจะสรุปได้ว่าการค้นหารูปภาพโดยวิธี Manhattan Distance จะสามารถค้นหารูปภาพและองค์ประกอบที่ใกล้เคียงกันได้มากที่สุด ด้วยมีการค้นหารูปภาพโดยวิธี Cosine Similarity และการค้นหารูปภาพโดยวิธี Euclidean Distance สามารถค้นหารูปภาพได้ใกล้เคียงกับการค้นหารูปภาพโดยวิธี Manhattan Distance มากแตกต่างกันเพียงเล็กน้อย ส่วนการค้นหารูปภาพโดยวิธี Histogram Intersection และ Kullback-Leibler สามารถค้นหารูปภาพที่ต้องการค้นหาได้ค่อนข้างน้อยมาก

5.2 ปัญหาที่พบ

1. เนื่องจากรูปภาพในฐานข้อมูลมีจำนวนมาก ทำให้การค้นหารูปภาพภาพค่อนข้างช้า
2. การเปรียบเทียบแต่ละวิธีมีวิธีการเปรียบเทียบแตกต่างกัน จึงอาจทำให้ประสิทธิภาพความถูกต้องของการค้นหารูปภาพลดลงมาบ้าง
3. รูปภาพตัวอย่างที่นำมาเปรียบเทียบอาจจะมีเพียง 1 หรือ 2 ภาพ ซึ่งเป็นรูปภาพโดดๆ ที่ไม่มีคล้ายคลึงกับรูปภาพอื่นในฐานข้อมูล ดังนั้นโปรแกรมที่สร้างขึ้นอาจจะค้นหารูปภาพไม่ตรงตามความต้องการ

5.3 ข้อเสนอแนะ

1. เมื่อมีการพัฒนาโปรแกรมต่อ ควรเพิ่มส่วนที่สามารถรับข้อมูลจากภายนอกมาเปรียบเทียบกับข้อมูลภายในโปรแกรมได้ด้วย

2. เมื่อมีการพัฒนาโปรแกรมต่อ ควรพัฒนาโปรแกรมค้นหาให้มีความถูกต้องแม่นยำมากกว่านี้

3. เมื่อมีการพัฒนาโปรแกรมต่อ ควรพัฒนาให้โปรแกรมค้นหาข้อมูลเร็วกว่านี้

เอกสารอ้างอิง

[1] ศุภชัย สมพานิช.คู่มือการเขียนโปรแกรมและใช้งาน Visual C#.NET .DEV BOOK. 2546

[2] ศุภชัย สมพานิช.Database Programming ด้วย VB2005 & VC#2005.DEV BOOK. 2549

[3] ธีรภักดิ์ ทักสี, เอกโรจน์ ศัลยพงษ์. การค้นหาภาพวาดในคลังรูปภาพศิลปะ (Retrieval of Painting in Digital Art Library). มหาวิทยาลัยนเรศวร. 2548

[4] ธนินทร์ พนาศุภวุฒิ, สุขสันห์ สุภาพรหม. การวิเคราะห์และค้นหาภาพสำหรับระบบสารสนเทศภูมิศาสตร์ (Image Analysis and Search for Geographical Information System). มหาวิทยาลัยนเรศวร. 2548

[5] Olarik Surinta, Supot Nitsuwat. " Handwritten Thai Character Recognition Using Fourier Descriptors and Robust C-Prototype" [Online]. Available: http://suanpalm3.kmitnb.ac.th/journal/pdf/vol3/ch3-13.pdf

[6] P. Wu, B.S. Manjunath, S. Newsam, and H.D. Shin. A "Texture descriptor for browsing and similarity retrieval" Department of Electrical and Computer Engineering. University of California. Santa Barbara, CA 93106-9560. USA.

[7] "Euclidean Distance." [Online].Available: http://en.wikipedia.org/wiki/Euclidean_distance

[8] "Cosine Similarity." [Online].Available: http://www10.org/cdrom/papers/519/node12.html

[9] "Histogram Intersection." [Online].Available: http://www.eng.tau.ac.il/~hayit/553.5547/ass2/ass2part1.html

[10] "Manhattan Distance." [Online].Available: http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Manhattan_Distance_Metric.htm

[11] "Kullback-Leibler Divergence." [Online].Available: http://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence

ประวัติผู้เขียนโครงงาน

ชื่อ นายชเยศ กองจันทร์

ภูมิลำเนา 3 หมู่ 11 ตำบลปง อำเภอปง จังหวัดพะเยา

ประวัติการศึกษา

- จบการศึกษาระดับมัธยมจาก โรงเรียนพะเยาพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยนเรศวร

E-mail: en_comp_@hotmail.com

มหาวิทยาลัยนเรศวร