การสืบค้นวีดีโอคลิปโดยใช้เว็บแอพพลิเคชั่น

Web-Based Application for retrieval of video clips

นายวรุตม์ แสงขัติ รหัส 45360385
นางสาวศศิตา ธนวรรณ รหัส 45360468
นายเกิดพงศ์ พึ่งกริม รหัส 45360641

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2548

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	การสืบค้นวีดีโอคลิปโดยใช้เว็บแอพพลิเคชั่น		
ผู้ดำเนินโครงงาน	นายวรุฒม์	แสงขัติ	รหัส 45360385
	นางสาวศศิตา	ธนวรรณ	รหัส 45360468
	นายเกิดพงศ์	พึ่งกริม	รหัส 45360641
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. สุชาติ แย้มเม่น		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2548		

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของ**การศึกษา**ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. สุชาติ **แย้มเม่น)**

.....กรรมการ

(ดร.พนมขวัญ ริยะมงคล)

.....กรรมการ

(ดร.ไพศาล มุณีสว่าง)

หัวข้อโครงงาน	การสืบค้นวีดีโอคลิปโดยใช้เว็บแอพพลิเคชั่น		
ผู้ดำเนินโครงงาน	นายวรุตม์	แสงขัติ	รหัส 45360385
	นางสาวศศิตา	ธนวรรณ	รหัส 45360468
	นายเกิดพงศ์	พึ่งกริม	รหัส 45360641
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. สุชาติ		แย้มเม่น
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2548		

.....

บทคัดย่อ

โครงงานนี้ศึกษาและพัฒนาโปรแกรมการค้นหาไฟล์วีดีโอคลิปในฐานข้อมูล โดยใช้หลักการของ Color feature ของระบบสี RGB เพื่อทำการเปรียบเทียบค่าที่ได้จากการ quantized RGB ของแต่ละคีย์เฟรม แล้วหาค่าผลต่าง เรียงลำดับจากน้อยไปมาก ทำให้ได้ผลการค้นหาที่เกิดจากการเปรียบเทียบค่าของการ quantized RGB ซึ่งเราจะมีการศึกษาทั้งหมด 3 แนวทางด้วยกัน คือ การค้นหาแบบรูปภาพ , การค้นหาโดยใช้วีดีโอ 1 คีย์เฟรม , การค้นหาโดยใช้วีดีโอ 3 คีย์เฟรม โดยโปรแกรมที่พัฒนานี้จะใช้ C# WindowApplication ของ Microsoft visual studio.NET และมีการจัดการฐานข้อมูลโดยใช้ MYSQL เป็นเครื่องมือในการจัดการ

โดยผลการทดลองที่ได้สามารถเปรียบเทียบการค้นหาของทั้ง 3 วิธี โดยใช้หลักการของ Color feature ผลที่ได้อาจไม่แม่นยำเมื่อเทียบกับการใช้หลักการอื่นๆ อย่างเช่น Shape feature , Texture feature

Project title	Web-Based Application for retrieval of video clips		
Name	Mr. Warut	Saengcut	ID. 45360385
	Miss Sasita	Thanawan	ID. 45360468
	Mr. Kerdpong	Pheungkrim	ID. 45360641
Project advisor	Assistant Professor Suchart Yammen , Ph.D.		
Major	Computer Engineering		
Department	Electrical and Computer Engineering		
Academic year	2005		

Abstract

In this project, studies video clip searching program in database is studied and developed by color feature procedure of RGB color system and substitute quantized for histogram of RGB in order to find the difference of values and arrange them from the smallest to largest value. Then get the output that is from comparison quantizing RGB. There are 3 methods for searching by picture, video 1 key frame and video 3 key frame. The C# Window Application of Microsoft Visual Studio .NET 2003 develops video clip searching program and manages database by using MYSQL.

Experimental results of the three methods for searching are compared with each other method based on the color feature procedure. All results may have not precision when compared with other procedures such as Shape feature and Texture feature.

กิตติกรรมประกาศ

ในการทำโครงงานวิศวกรรมครั้งนี้ คณะผู้จัดทำขอกราบขอบพระคุณ ดร.ไพศาล มุณีสว่าง ที่ได้ให้คำปรึกษาโครงงานนี้ ทั้งทฤษฎีและขั้นตอนการปฏิบัติงานต่างๆ และขอบพระคุณ ผศ.ดร.สุชาติ แย้มแม่น และ ดร.พนมขวัญ ริยะมงคล ที่ได้ ที่ได้เสียสละเวลาเพื่อทำการตรวจสอบการทำงานและชี้แนวทางในการแก้ไขปัญหาโครงงานนี้

นายวรุตม์ แสงขัติ

นางสาวศศิตา ธนวรรณ

นายเกิดพงศ์ พึ่งกริม

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูปภาพ

สารบัญรูป(ต่อ)

บทที่ 1
บทนำ

1.1 ปัญหาและที่มาของโครงงาน

เนื่องจากปัจจุบันได้มีเทคโนโลยีต่างๆมากมาย หนึ่งในนั้นก็เป็นเทคโนโลยีการสื่อสารที่ใช้อินเตอร์เน็ต ที่นิยมใช้กันอย่างแพร่หลาย ทำให้มีข้อมูลในโลกอินเตอร์เน็ตเป็นจำนวนมากมีทั้งแฟ้มข้อมูล ,รูปภาพ, เพลง , วีดีโอ ,โปรแกรม ฯลฯ จึงเป็นการยากที่จะค้นหาข้อมูลที่เราต้องการได้อย่างถูกต้อง ทำให้ต้องมีการสร้างเครื่องมือที่จะช่วยในการค้นหาข้อมูล หรือที่เรียกว่า Search engine ขึ้นมา ซึ่งในปัจจุบันนี้โดยมากแล้วจะมีแต่ Search engine ประเภทคำหลัก Key word (การค้นหาโดยเปรียบเทียบคำหลัก (Key word) ที่ต้องการค้นหากับคำที่มีอยู่ในเว็บไซต์ทั้งหมดบนโลกอินเตอร์เน็ตนี้) ถ้าตัวอักษรในเว็บไซต์นั้นตรงกับคำหลัก (Key word) ที่เราต้องการก็จะนำเว็บไซต์ทั้งหมดมาแสดงเพื่อให้เราได้เลือกเว็บไซต์ที่เราต้องการ ด้วยเหตุนี้การที่เราใช้ Key word ในการค้นหาไฟล์วีดีโอ (File video)จะทำให้การค้นหานั้นผิดพลาดได้ง่าย อย่างเช่น ชื่อไฟล์วีดีโอ (File video) ไม่ตรงกับ key word ที่เราต้องการ ทำให้มีไฟล์วีดีโอ (File video)ที่เราไม่ต้องการแสดงออกมามากมาย ดังนั้นเราจึงคิดว่าน่าจะมีวิธีการค้นหาไฟล์วีดีโอ (File video) ด้วยวิธีอื่น ที่มีความรวดเร็วและค้นหาไฟล์วีดีโอ (File video) ได้อย่างแม่นยำมากยิ่งขึ้น ฉะนั้นในปัจจุบันจึงได้มีแนวคิดในการค้นหาไฟล์วีดีโอ (File video) โดยใช้กระบวนการ Content-Based Video Retrieval (CBVR)

กระบวนการ CBVR จะประกอบไปด้วย Content-based Image Retrieval (CBIR) และ Content- Based Audio Retrieval (CBAR) ซึ่งเราจะเน้นหลักการทำงานของ CBIR เป็นหลัก โดยใช้หลักการง่ายๆคือ การจัดเตรียมฐานข้อมูล(Data Base) จะนำไฟล์วีดีโอ (File video) มาทำการแบ่งออกเป็นเฟรม (Frame) ซึ่งวีดีโอแต่ละเรื่องก็จะมีหลายเฟรมด้วยกัน ต่อจากนั้นก็จะทำการวิเคราะห์ให้ได้เฟรมหลัก (Key frame) ออกมาและเก็บเอาเฟรมหลัก (Key frame) ที่ได้มานี้ให้อยู่ในรูปของการประเมินค่าตัวแทนของค่าสี (Quantized RGB) ไว้ในฐานข้อมูล (Data base)

จากนั้นก็จะเป็นในส่วนของ Search engine โดยใน Search engine นี้จะนำวีดีโอที่ถูกตัดเป็นไฟล์เล็กๆ (Clip video) ที่เราต้องการมาทำไฟล์ออกเป็นส่วนๆ (Segmentation) ซึ่งจะทำให้ได้เฟรมหลัก (Key frame) ออกมา แล้วแทนเฟรมหลักที่ได้นี้ในรูปของการประเมินค่าตัวแทนของเวคเตอร์ เหมือนการจัดการแบ่งไฟล์วีดีโอในส่วนของฐานข้อมูล

ในการค้นหานั้น จะนำเฟรมหลัก (Key frame) ของวีดีโอตัวอย่างที่ต้องการค้นหา (Clip Video) มาเปรียบเทียบกับเฟรมหลัก (Key frame) ของไฟล์วีดีโอในฐานข้อมูลที่เก็บในค่าที่ได้จากการ Quantized RGB ซึ่งจะใช้หลักการวิเคราะห์จุดเด่นของสี (Color feature) ในการวิเคราะห์ โดย

จะเปรียบเทียบความแตกต่างระหว่างสีจาก Color histogram ดังนั้นเราจะได้ผลการค้นหาไฟล์วีดีโอ (File video) ที่มีความใกล้เคียงกับวีดีโอตัวอย่าง Clip video ที่เราต้องการ

ซึ่งวิธีการค้นหาแบบ CBVR นั้นมีข้อดี คือ มีความรวดเร็ว แม่นยำในการค้นหา Video ที่เราต้องการ เหมาะสำหรับการค้นหา File video clip ต่างๆ เช่น การ์ตูน , ภาพยนตร์ , ข่าว , มิวสิควีดีโอ ฯลฯ และสามารถที่จะนำไปประยุกต์ใช้ในอาชีพต่างๆ ระกาศได้ เช่น ผู้ปข่าวใช้ค้นหา File video เหตุการณ์ต่างๆ , พัฒนาสื่อการเรียนการสอนประเภท E-learning และอื่นๆ อีกมากมาย

วิธีการค้นหาโดยใช้กระบวนการ CBVR

- **Segmentation**

เป็นขั้นตอนการแยกไฟล์วีดีโอ (File video) ออกเป็นเฟรม (Frame)

- **Key frame**

เป็นเฟรมที่สามารถแสดงลักษณะเด่นของแต่ละคลิปวีดีโอ Clip ซึ่งเราจะหา key frame ทั้งหมด 3 key frame คือ เฟรมแรก (Frame0) , เฟรมกลาง (Frame1) , เฟรมท้าย (frame2)

- **Video Indexing**

เป็นขั้นตอนการนำ Key frame ทั้ง 3 คีย์เฟรม ที่ได้มาเก็บในรูปตัวแทนค่าสี RGB (quantized RGB) ในฐานข้อมูล

- **Video retrieval (หรือ Content Matching)**

เป็นกระบวนการค้นหาวีดีโอตัวอย่าง (Clip Video) โดยใช้ Key frame จาก Clip video มาเปรียบเทียบกับ Key frame ของไฟล์วีดีโอในฐานข้อมูล

1.2 วัตถุประสงค์

1.2.1 ศึกษาการวิเคราะห์เฟรมหลัก (Key frame) โดยการวิเคราะห์โดยใช้หลักการของ Color feature

1.2.2 จัดทำฐานข้อมูลโดยใช้ตัวจัดการฐานข้อมูล MYSQL ในการเก็บแฟ้มวีดีโอ (File video) และ Key frame ที่ได้จากการแยกไฟล์วีดีโอ (Video Segmentation) เพื่อความเป็นระเบียบและเพื่อให้ง่ายต่อการค้นหา

1.2.3 สร้าง search engine โดยใช้ โปรแกรม Microsoft Visual Studio.NET 2003 ที่สามารถค้นหาไฟล์วีดีโอ (File video) ในฐานข้อมูลที่ได้เตรียมไว้แล้ว โดยใช้เฟรมหลัก (Key frame) จากวีดีโอตัวอย่าง (Video clip) ในการค้นหา ซึ่งสามารถทำการค้นหาได้ 3 แบบ คือ การค้นหาแบบใช้รูปภาพ , การค้นหาวีดีโอแบบใช้ 1 key frame , การค้นหาวีดีโอแบบใช้ 3 key frame ซึ่งผู้ใช้สามารถที่จะตั้งคำถาม (query) โดยใช้วีดีโอตัวอย่างได้

1.3 ขอบข่ายของงาน

สร้าง Search engine ในการ search ซึ่งเราจะใช้หลักการวิเคราะห์จาก color feature ในการเปรียบเทียบระหว่าง key frame ของวีดีโอตัวอย่าง (video clip) ที่ต้องการค้นหา กับ key frame ของไฟล์ วีดีโอ (File video) ในฐานข้อมูล โดยในฐานข้อมูลที่เราสร้างขึ้นจะนำ file video มาแบ่งแยกออกเป็นเฟรม (frame) แล้วทำการวิเคราะห์หาค่า key frame ออกมา ซึ่งจะแบ่งการค้นหาออกเป็นเป็น 3 แบบในการค้นหา คือ การค้นหาแบบใช้รูปภาพ , การค้นหาวีดีโอแบบใช้ 1 key frame , การค้นหาวีดีโอแบบใช้ 3 key frame ซึ่งการค้นหาวีดีโอแบบใช้ 1 key frame นั้นจะใช้เฟรมตรงกลางเป็น key frame ส่วนการค้นหาวีดีโอแบบ 3 key frame จะใช้ เฟรมหน้า(Frame0) , เฟรมกลาง(Frame1) และเฟรมหลัง(Frame2) เป็น key frame โดยในการพัฒนานั้นจะใช้ โปรแกรมที่เขียนด้วยภาษา C# ช่วยในการแบ่งไฟล์วีดีโอออกเป็นเฟรม ,และโปรแกรม Microsoft Visual Studio. NET ช่วยในการ พัฒนา Web search engine, ตัวจัดการฐานข้อมูล Mysql ช่วยในการจัดการกับฐานข้อมูล

1.4 ขั้นตอนการดำเนินโครงงาน

กิจกรรม	2547								2548																																			
	พ.ย.				ธ.ค.				ม.ค.				ก.พ.				มี.ค.				เม.ย.				พ.ค.				มิ.ย.				ก.ค.				ส.ค.				ก.ย.				ต.ค.			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. ศึกษาข้อมูลและทฤษฎีของ Content-based Video Retrieval (CBVR)																																															
2. ศึกษาทฤษฎีการวิเคราะห์วีดีโอในส่วนต่างๆ โดยเฉพาะ ส่วนของ frame และ key frame																																															
3. ศึกษาทฤษฎีการวิเคราะห์สี (Color feature) และมาตรฐานของสีในระบบ RGB																																															
4. ศึกษาทฤษฎี Color histogram และการทำ quantized RGB																																															
5. ศึกษาภาษา C# และเขียนโปรแกรมในส่วนของการตัดวีดีโอเก็บไว้ในฐานข้อมูล (Video indexing)																																															
6. ทดลองโปรแกรมตัดวีดีโอและทำการเก็บวีดีโอที่ได้ลงไว้ในฐานข้อมูลในรูปของเวกเตอร์																																															
7. สร้างโปรแกรม Search engine โดยใช้ C#.NET																																															
8. ทำการเชื่อมต่อระหว่างโปรแกรม search engine กับฐานข้อมูล																																															
9. ทดลองใช้งานโปรแกรม search engine ที่สร้าง																																															
10. แก้ไขและเพิ่มเติมส่วนที่ยังบกพร่องของโปรแกรม Search engine และ ฐานข้อมูล																																															
11. สรุปข้อดีข้อเสียของการศึกษาแบบ Content-base Video Retrieval (CBVR)																																															
12. จัดทำเอกสารประกอบโครงงาน																																															

1.5 ผลที่คาดว่าจะได้รับ

1.5.1 สามารถสร้าง Web search engine เพื่อใช้ในการค้นหาไฟล์ (File video) ได้อย่างมีประสิทธิภาพ

1.5.2 สร้างฐานข้อมูลที่จะใช้เก็บไฟล์วีดีโอ (File video) และสามารถแก้ไขปรับปรุงฐานข้อมูลให้มีประสิทธิภาพได้และนำมาใช้กับ Web search engine ที่สร้างขึ้น

1.5.3 สามารถวิเคราะห์เฟรมหลัก (Key frame) โดยวิเคราะห์จาก Color feature ของ key frame ได้

1.6 งบประมาณที่ใช้

1.6.1 ค่าใช้จ่ายในการซื้ออุปกรณ์ทางคอมพิวเตอร์	1500 บาท
1.6.2 ค่าใช้จ่ายในการทำรายงาน	1000 บาท
1.6.3 ค่าใช้จ่ายเบ็ดเตล็ด	500 บาท
รวมทั้งสิ้น	3000 บาท

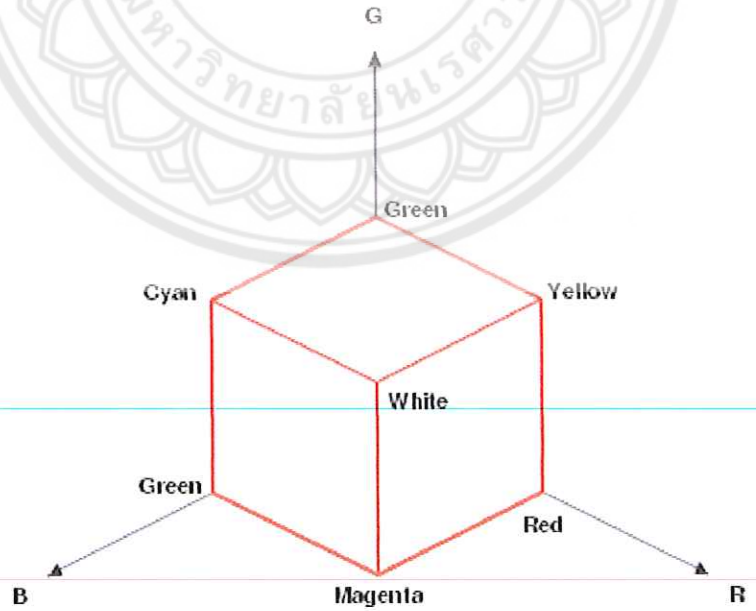
บทที่ 2
ทฤษฎีเบื้องต้น

2.1 มาตรฐานของสี

มาตรฐานของสีที่ใช้อยู่ในปัจจุบันมีอยู่หลายระบบด้วยกัน ทั้งนี้จะขึ้นอยู่กับการนำไปใช้ แต่โดยทั่วไปแล้วทุกมาตรฐานจะมีแนวคิดเดียวกันคือ การแทนจุดสีด้วยจุดที่อยู่ภายในสเปส 3 มิติ โดยจะมีแกนอ้างอิงสำหรับจุดสี นั้นในสเปสซึ่งแต่ละแกนจะมีความเป็นอิสระต่อกัน ตัวอย่างเช่น ในระบบ RGB จะมีแกนสีคือ แกนสีแดง เขียว และน้ำเงินในระบบ HLS จะมีแกนเป็น ค่าสี(hue) ความสว่าง(lightness)และความบริสุทธิ์ของสี(saturation) ตัวอย่างระบบสีที่เราสนใจในการพิจารณาคือ ระบบ RGB (Red Green Blue)

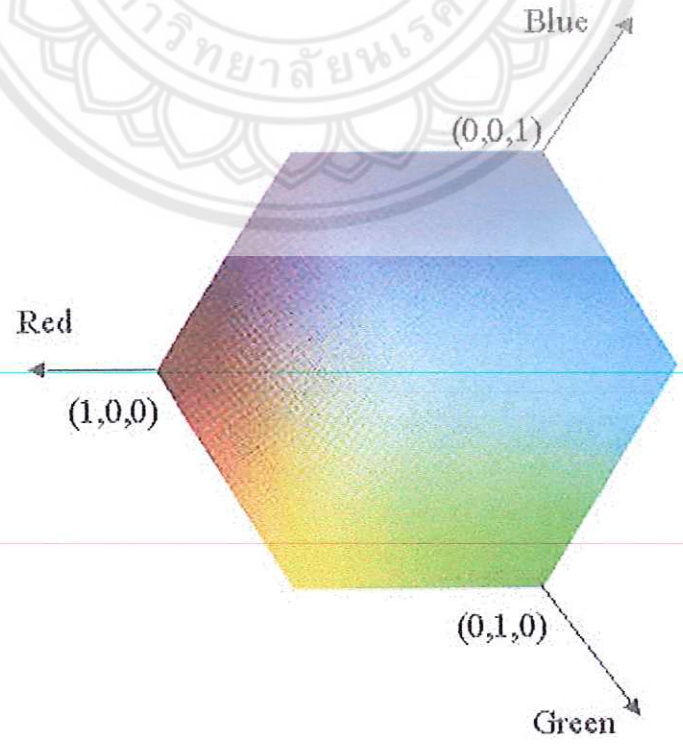
2.1.1 ระบบสี RGB

ระบบสี RGB เป็นระบบสีที่เกิดจากการรวมกันของแสงสีแดง เขียวและน้ำเงินโดยมีการรวมกันแบบ Additive ก็จะเป็นกราฟที่มี 3 แกน โดยแต่ละแกนก็จะแทนค่าสี R,G,B โดยไล่จากค่า 0 ที่มีความเข้มสีมาก จนไปถึงค่า 1 ที่มีความเข้มสีน้อยดังรูป



รูปที่ 2.1 RGB coordinates system

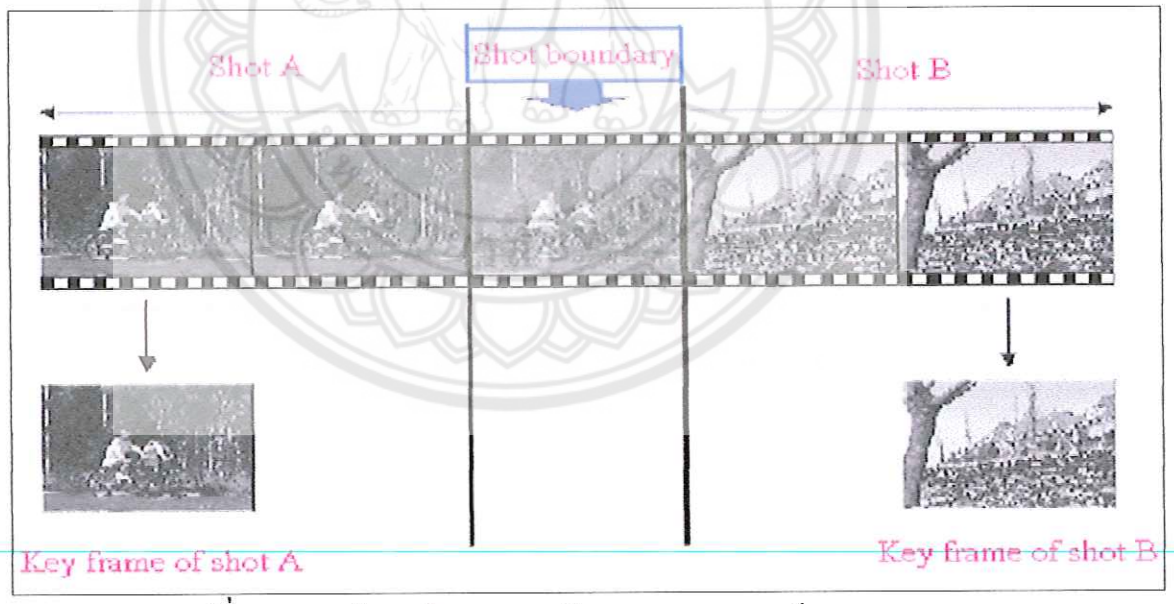
การบอกหรือระบุสีต่างๆ โดยใช้ "พิกัด" หรือ "ค่าของแม่สี" สามค่า ในการแทนค่าของสีต่างๆที่เราใช้งาน เช่น ในงานพิมพ์แยกสี จึงทำให้ดูเหมือนว่า "สี" นั้นมี สามมิติ ตามมิติของ "แม่สี" ทั้งสาม คือ แดง เหลือง และฟ้า โดยสมมติ ให้เป็นค่า R Y B ต่างๆ (ตัวย่อของ Red Yellow และ Blue) หรือเราอาจจะใช้ "ค่าแม่สีของแสง" คือ R G B (Red Green Blue) ในการสร้างแสงสีจาก "แม่สีของแสง" สามสีซึ่งต่างจาก "แม่สีของสี" ตรงที่สีเหลืองกับ สีเขียว กล่าวคือแม่สีของสีจะเป็น "สีเหลือง" แต่ แม่สีของแสงจะเป็น "สีเขียว ซึ่งสามารถเทียบเคียงได้กับการระบุพิกัดแบบใช้ค่า X Y Z ในการระบุตำแหน่งของวัตถุต่างๆในระบบพิกัดแบบคาร์ทีเซียนนั่นเอง เมื่อเป็นเช่นนี้ จึงทำให้ดูเหมือนว่า สี ต่างๆนั้นมี สามมิติ คือค่า R Y B หรือ ค่า R G B ต่างๆ ตามส่วนประกอบของมัน ซึ่งเราก็สามารถ สร้าง "ลูกบาศก์ของสี" (Color Cube) ขึ้นได้จากค่า R Y Bหรือ R G Bทั้งสามค่านั้นได้ ดังนั้นสีจึงดูกลายเหมือนเป็น"สามมิติ"ไปในทันที ทั้งๆที่ความจริงแล้ว สีต่างๆนั้นมันอยู่ในมิติเดียวกัน คือถ้าหาก เราเอาสีต่างๆมาเรียงต่อๆกัน มันจะ กลายเป็น "แถบสเปคตรัมของแสง"หรือ" แถบสีรุ้ง" อย่างที่เราคุ้นเคยกันดี เพราะในธรรมชาติมันก็เป็นเช่นนั้นจริงๆคือ เวลาที่เราเห็น รุ้งกินน้ำบนท้องฟ้าหรือ เราเห็นแสงหักเหผ่านแท่งแก้วปริซึมที่กระจายออกเป็นแถบสีรุ้ง ตาม Spectrum ของแสงคือ VIBGYOR หรือ ROYGBIV กล่าวคือ แดง ส้ม เหลือง เขียว น้ำเงิน คราม ม่วง(Red Orange Yellow Green Blue Indigo Violet)เรียงลำดับตามความยาวคลื่นหรือความถี่ของแสงนั่นเอง นั่นคือแสงสีต่างๆนั้นต่างอยู่ในมิติเชิงเส้นอันเดียวกัน ดังนั้นการที่ เกิดค่า R Y B หรือ R G B ของสีต่างๆ นั้นจึงเป็น "รงคมิติ" หรือเกิด"มิติเทียม" ของสี ขึ้น จากส่วนผสมของแม่สีทั้งสาม ซึ่งไม่ใช่มิติที่แท้จริง!



รูปที่ 2.2 RGB color model เมื่อมองจากทางด้าน white

2.2 ลำดับของไฟล์วีดีโอ (Video Segmentation)

ลักษณะของข้อมูลดิจิตอลประเภทวีดีโอไฟล์นั้น จะประกอบด้วยฉากต่างๆ (Scene) ที่มีความเหมือน หรือแตกต่างกันนำมาเรียงต่อกันเป็นเรื่องราวของวีดีโอไฟล์นั้นๆ ยกตัวอย่างเช่น ถ้าวีดีโอไฟล์นั้นเป็นเรื่องเกี่ยวกับกีฬา ในแต่ละฉากหรือ scene ก็ต้องมีส่วนของเนื้อหาที่แตกต่างกัน แต่ใน scene หนึ่งๆ จะมีส่วนที่คล้ายกันคือ เฟรม (Frame) ที่นำมาต่อๆ กันเป็น scene แต่ละ scene ของวีดีโอไฟล์ ถ้าจะอธิบายอย่างคร่าวๆ ให้เห็นภาพแล้ว จะขอยกตัวอย่างการจับภาพในวีดีโอไฟล์ ซึ่งขณะที่เราทำการบันทึกภาพการแสดงคอนเสิร์ตหนึ่ง ขณะนั้นกำลังจับภาพของนักร้องนำอยู่ ซึ่งนักร้องก็เต้นแล้วก็เปลี่ยนอิริยาบถไปเรื่อยๆ ต่อมาก็เปลี่ยนมาจับภาพของผู้คนที่มาดูคอนเสิร์ตโดยแพลนกล้องถ่ายคนดูโดยรอบ ซึ่งเหตุการณ์สองเหตุการณ์ที่บันทึกนี้รวมกันเราเรียกว่า scene การเปลี่ยนฉากนี้เองจะทำให้เกิดความแตกต่างกันของภาพที่เราบันทึก โดย scene ที่เราทำการบันทึกอยู่นี้จะแบ่งเป็นสอง shot คือ ส่วนของ shot ที่มีเรื่องราวของนักร้องบนเวที กับส่วนของ shot ที่เกี่ยวกับผู้คนที่มาดูคอนเสิร์ต เป็นต้น ซึ่งในการค้นหาข้อมูลที่ต้องการ เราต้องการ shot ที่เหมือนหรือคล้ายกันเพื่อใช้ในการค้นหา ในกระบวนการนี้จะขออธิบายในส่วนต่อไปเพื่อช่วยให้ผู้ใช้มีความสะดวกมากขึ้นต่อการค้นหาข้อมูลดิจิตอลประเภทวีดีโอไฟล์ในระดับ scene ได้อย่างมาก

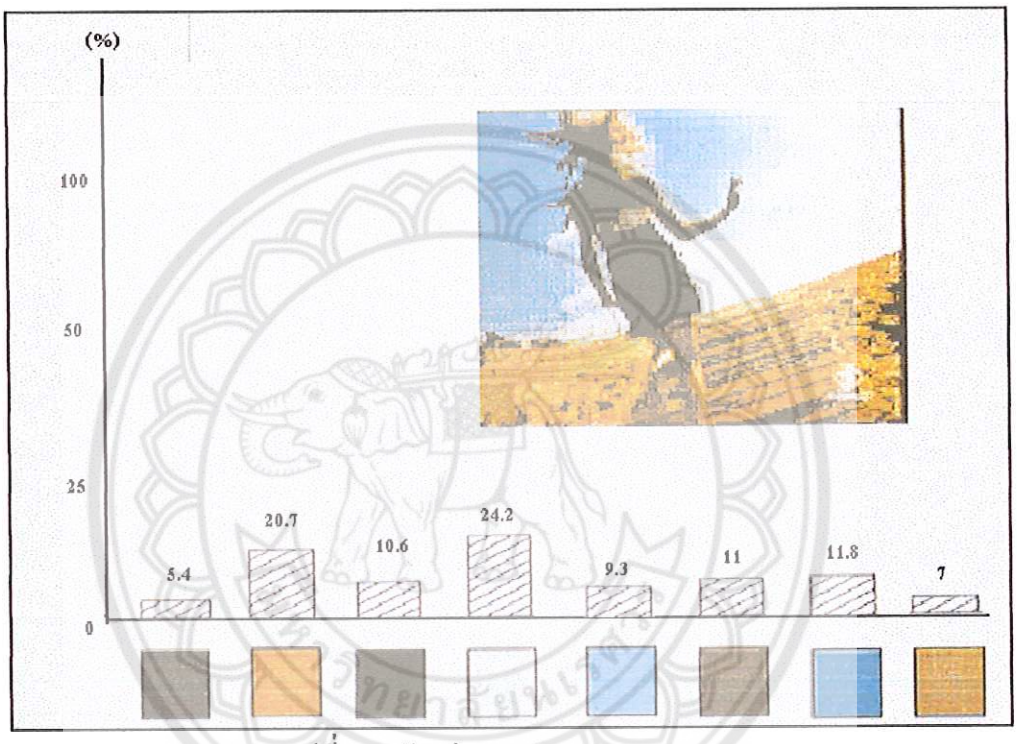


รูปที่ 2.3 การแบ่งภาพใน 1 Scene มี 2 shot และ 1 shot มี 1 key frame

2.3 Color histograms

ในระดับความเข้มของ Histogram เราจะพิจารณาที่การกระจายตัวของ Histogram ใน Image ซึ่งในการพิจารณาสีของภาพนั้น ต้องกำหนดขอบเขต Color Histogram ให้ชัดเจน โดยปกติแล้วเราจะใช้ระบบสีของ RGB

เป็นกราฟที่ประกอบไปด้วย แกนในแนวตั้งที่แสดงความถี่ของสี และในแนวแกนนอนที่แสดงช่วงของสี ดังรูป



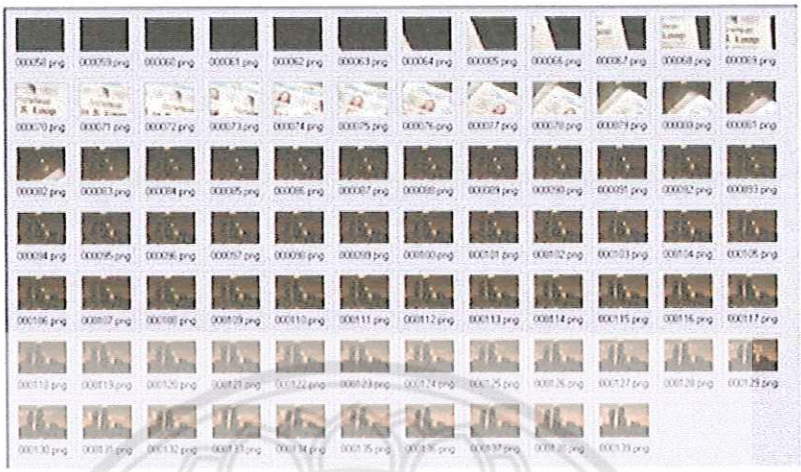
รูปที่ 2.4 ตัวอย่างการทำ color histogram

2.4 การทำดรรชนีเฟรมหลัก (Key-frame)

ในการค้นหาข้อมูลเกี่ยวกับวีดีโอไฟล์ที่ต้องการในเรื่องราวหรือเนื้อหาที่ผู้ใช้จะนำไปใช้งานด้านต่างๆ นั้น เป็นเรื่องที่มีความยุ่งยาก ดังนั้นเพื่อให้ง่ายต่อการเปรียบเทียบหรือในการค้นหาข้อมูลเกี่ยวกับวีดีโอไฟล์ เราจึงนำเอาความรู้ทางด้าน Image processing เข้ามาช่วยในการค้นหาหรือจัดทำข้อมูลประเภทวีดีโอไฟล์ ซึ่งโดยทั่วไปแล้วเราจะหาตัวแทนของส่วนย่อยของคลิปวีดีโอเพื่อใช้แทนลักษณะของเรื่องราวหรือเนื้อหาในแต่ละ คลิปวีดีโอนั้นๆ ซึ่งเราจะเรียกตัวแทนของคลิปวีดีโอนี้ว่า ดัชนีเฟรมหลัก (Key-frame) นั่นเอง

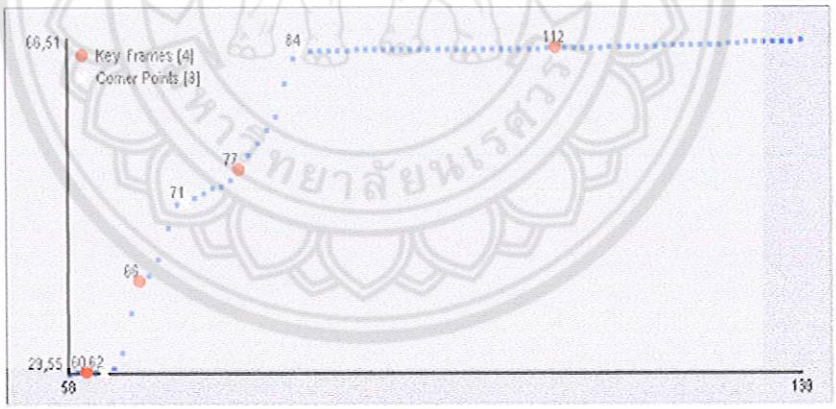
ซึ่งการหาดัชนีเฟรมหลักนั้นสามารถหาได้หลายแบบ ไม่ว่าจะเป็นการนำเฟรมที่ส่วนเริ่มต้นของ คลิปวีดีโอหรือนำส่วนท้ายสุดของคลิปวีดีโอมาเป็นดัชนีเฟรมหลัก แต่โดยส่วนใหญ่แล้วเราจะนำค่าตรงกลางของเฟรมในแต่ละคลิปวีดีโอ มาเป็นส่วนบ่งบอกลักษณะหรือเป็นตัวแทน

ของวีดีโอแต่ละคลิป นั้นๆ หรือ นำมาเป็นค่าของดัชนีเฟรมหลัก (Key-frame นั่นเอง) ดังแสดงตัวอย่างดังรูปที่ 2.5 นี้



รูปที่ 2.5 แสดงเฟรม (Frame) ทั้งหมดของคลิปวีดีโอ

และเราสามารถหาดัชนีเฟรมหลัก (Key-frame) โดยดูจากกราฟด้านล่าง ซึ่งเป็นการหาค่ากลางของวีดีโอแต่ละคลิป



รูปที่ 2.6 แสดงการหา ดรรชนีเฟรมหลัก (Key-frame) จากค่ากลางของแต่ละคลิปวีดีโอ

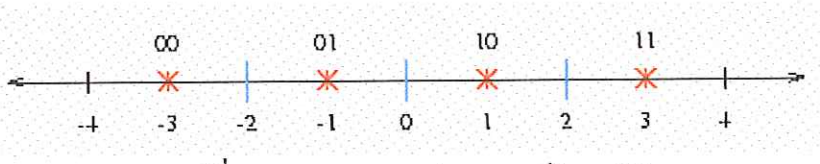
ซึ่งภาพหรือเฟรมที่ได้จากการหาค่าดรรชนีเฟรมหลัก (Key-frame) ของตัวอย่างนี้คือ

รูปที่ 2.7 แสดงดรรชนีเฟรมหลักที่หาได้จากตัวอย่าง

2.5 Vector Quantization (VQ)

2.5.1 การทำเวกเตอร์ 1 มิติ

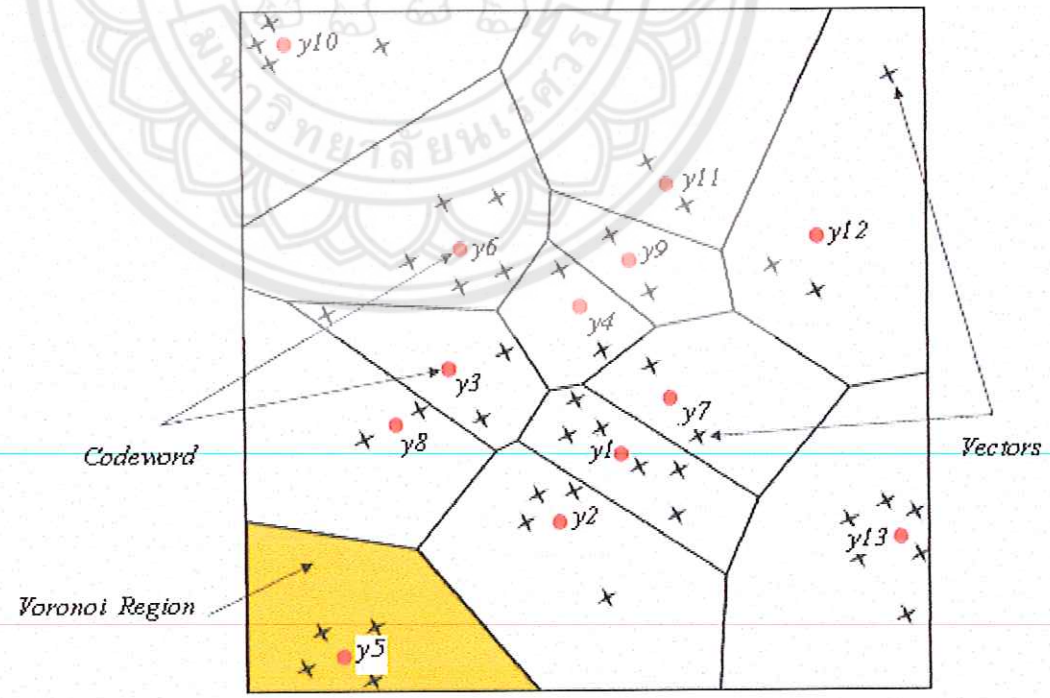
เป็นการเก็บค่าของภาพที่มีขนาดใหญ่ให้มีขนาดเล็กลง โดยการเก็บให้อยู่ในรูปของ Vector เราจะประมาณค่าของสีที่มีความใกล้เคียงกัน ประมาณค่าให้เป็นค่าเดียวกัน ดังตัวอย่าง Vector ใน 1 มิติ



รูปที่ 2.8 แสดงการหาค่าเวกเตอร์ใน 1 มิติ

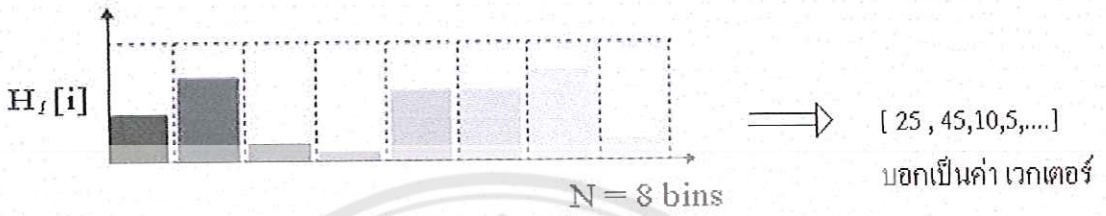
โดยค่าที่น้อยกว่า -2 จะประมาณให้เป็น -3, ค่าที่อยู่ระหว่าง -2 กับ 0 ให้มีค่าเป็น -1, ค่าที่อยู่ระหว่าง 0 กับ 2 ให้มีค่าเป็น 1, ค่าที่มากกว่า 2 ให้มีค่าเป็น 3 จะเห็นว่าจากที่เราต้องเก็บค่าทั้งหมด 3 bit เราสามารถให้เหลือ 2 bit ทำให้เก็บค่าได้ง่ายขึ้น

2.5.2 การทำเวกเตอร์ 2 มิติ



รูปที่ 2.9 แสดงตำแหน่งของ Codeword, Vectors และ Voronoi Region

จากการหาค่า Color histogram ในแต่ละ Key frame จะได้กราฟในระนาบ 2 มิติ ระหว่างค่าความถี่ของสี กับ เฉดสี ก็จะมีกราฟอยู่ 3 กราฟ เราจะนำกราฟแต่ละอันมาเก็บให้อยู่ในรูปของ Vector ขนาด โดยเก็บค่าความถี่ในแต่ละเฉดสี ก็จะได้ Vector 1 ค่า และเมื่อนำ Vector ในแต่ละเฉดสีมาต่อกันโดยเขียนเรียงกันไปก็จะได้ Vector ที่มีขนาดใหญ่ และนี้ก็คือค่า Vector ตัวแทนของ Key frame นั้นๆ



รูปที่ 2.10 ตัวอย่างการแปลงค่า Color histogram เป็นค่าที่เกิดการ quantized RGB

2.5.3 การเปรียบเทียบค่า Quantize ของ RGB

โดยในขั้นตอนแรก เราจะนำค่า Quantize RGB ที่ได้ทั้งสองค่าเปรียบเทียบมาทำการหาค่าส่วนต่าง (distance)

$$d(Q,I)=\sum_{i=1}^{N}\left|H_Q[i]-H_I[i]\right| \qquad (2.1)$$

โดยที่ d (Q,I) คือค่าความต่าง N คือ จำนวนของข้อมูลใน Vector

H_Q และ H_I คือค่าของการ Quantize RGB ที่ตำแหน่ง i ดังนั้นจะได้สมการความแตกต่างดังนี้

2.6 ทำความรู้จักกับ C#

C# เป็นภาษาที่ถูกสร้างขึ้นมาเพื่อทำงานบน .NET Platform สร้างและก็ทำงานในลักษณะของ Object Oriented ได้อย่างสมบูรณ์ ซึ่งเมื่อเปรียบเทียบกับ C++ ที่ยังทำงานในลักษณะของ Object Oriented Programming (OOP) ได้บางส่วน, ไลบรารีของ C# ถูกสร้างขึ้นเพื่อให้ทำงานได้ครอบคลุมตั้งแต่การสร้างรูปแบบการติดต่อแบบ GUI ไปจนถึงการแอ็คเซสฐานข้อมูลผ่านอินเตอร์เน็ตหรือแม้แต่การทำงานร่วมกับ XML เพื่อทำให้การแลกเปลี่ยนข้อมูลระหว่างแอพพลิเคชั่นทำได้อย่างสมบูรณ์ไม่ว่าข้อมูลนั้นจะอยู่บนแพลทฟอร์มใดก็ตาม

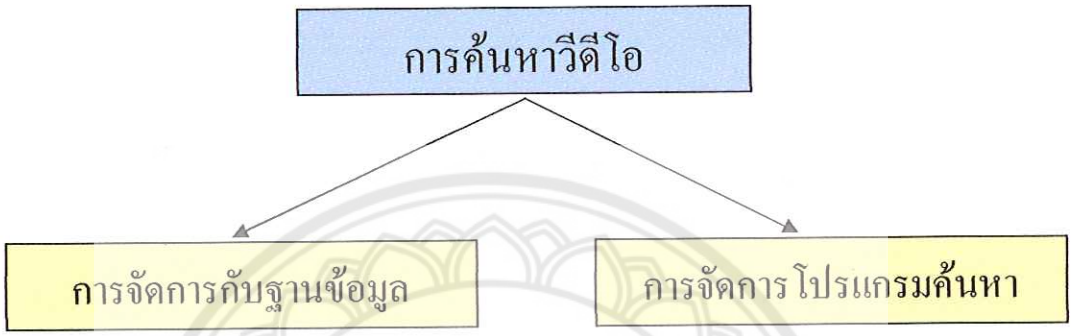
เมื่อเปรียบเทียบกับ C++ แล้ว การสร้างแอพพลิเคชั่นจะทำได้ง่ายกว่ามาก เนื่องจาก C# ถูกออกแบบมาเพื่อการสร้างแอพพลิเคชั่นให้ทำงานบนอินเตอร์เน็ตโดยตรง (.NET Framework) นอกจากนี้ C# เป็น Object Oriented Programming (OOP) อย่างสมบูรณ์ ไม่ว่าจะเป็น

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

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

บทที่ 3
วิธีการดำเนินการ

ในการทดลองการสืบค้นวีดีโอคลิปโดยใช้เว็บแอพพลิเคชันนั้น จะแบ่งขั้นตอนการดำเนินการงานออกเป็นส่วนต่างๆ ได้ดังนี้



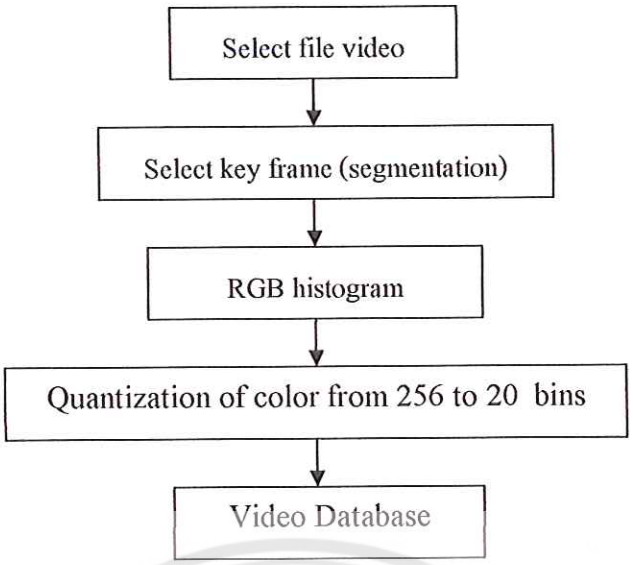
รูปที่ 3.1 ส่วนประกอบของขั้นตอนการค้นหาวีดีโอ

โดยจะมีการทำงานย่อยๆ ลงไปอีกในส่วนต่างๆ โดยจะมีขั้นตอนการทำงานดังต่อไปนี้

ขั้นตอนการทำงาน

3.1 การจัดการฐานข้อมูล

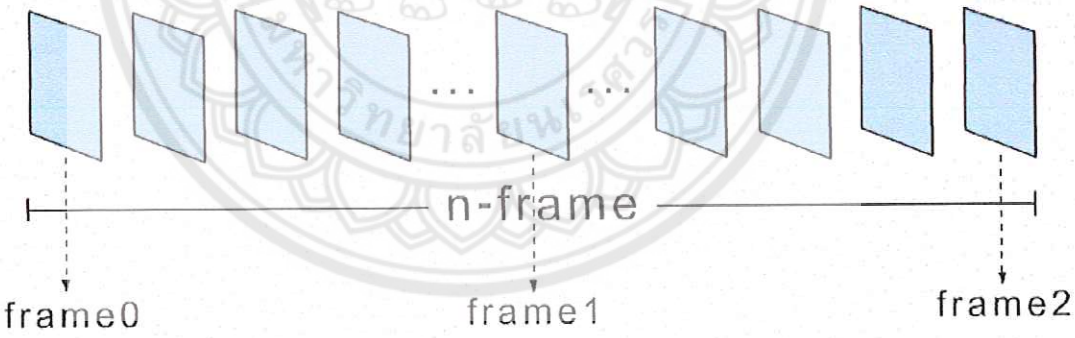
การจัดการฐานข้อมูลนั้นจะมีส่วนของ Interface ช่วยในการจัดการ โดยโปรแกรมในส่วนนี้จะทำการประมวลผลเพื่อเก็บค่าที่ได้จากการ quantized RGB ของ Key frame ของวีดีโอที่จะนำมาเก็บเป็นดาต้าเบส โดยจะมีหลักการในการหาค่าดังต่อไปนี้



รูปที่ 3.2 ขั้นตอนการจัดการฐานข้อมูล

3.1.1 การเลือก Key frame จาก Video

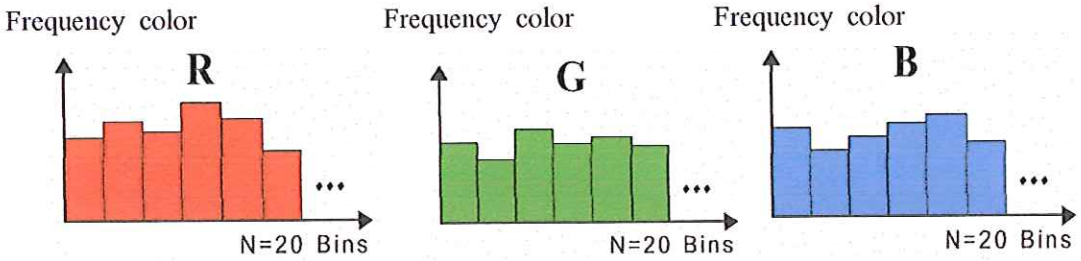
ไฟล์วีดีโอมาหาค่า Key frame ของวีดีโอแต่ละอัน โดยกำหนดค่า key frame ที่เก็บในฐานข้อมูลทั้งหมด 3 key frame/Video ซึ่งจะเลือกเอาเฟรม หน้า , กลาง , หลัง ของไฟล์วีดีโอ



รูปที่ 3.3 การหาค่า Key frame ของวีดีโอ

3.1.2 การหาค่า RGB Color histogram

การวิเคราะห์ค่าสีแต่ละ Pixel จะทำให้ได้ค่าสีที่ออกมาเป็น Histogram ซึ่งแต่ละ key frame จะมีค่า Histogram 3 ค่า คือ



รูปที่ 3.4 ทำการแยกค่าสี RGB

เมื่อได้ค่า Histogram แล้ว จะนำค่าทั้ง 3 อันมาเรียงต่อกัน ซึ่งแต่ละรูปเกิดจากการนำค่าสี RGB 256 สี (0 – 255) มาทำการแบ่งช่วง scale ออกเป็น 20 ช่วง (bin) ดังรูปที่ 3.4

รูปที่ 3.5 นำค่า Histogram ของ RGB มารวมกัน เป็น 60 bin ได้เป็นค่า ที่เกิดจากการ quantized RGB

$$Hi = [V_1\ V_2\ V_3\ V_4\ V_5\ V_6\ V_7\ V_8\ V_9\ V_{10}\ldots\ V_{60}] \qquad (3.1)$$

โดยที่ค่า $V_1 - V_{20}$ เป็นค่า histogram ของสีแดง Red

โดยที่ค่า $V_{21} - V_{40}$ เป็นค่า histogram ของสีเขียว Green

โดยที่ค่า $V_{41} - V_{60}$ เป็นค่า histogram ของสีน้ำเงิน Blue

ได้เป็นค่าที่ quantized RGB ที่เก็บในฐานข้อมูลทั้งหมด 60 ค่าดังนี้

path	framenumber	v1	v2	v3	v4	v5	v6	v7	v8	v9
E:\project\Pro	frame2	86254	19035	18820	27291	24208	24935	21823	13773	10848
E:\project\Pro	frame1	66548	15575	12876	18379	17541	17600	10532	6121	3322
E:\project\Pro	frame0	34434	11680	6542	6443	5312	3793	2937	3152	2588

รูปที่ 3.6 ค่า quantized RGB ของแต่ละ Key frame ที่เก็บในฐานข้อมูล (V1-V9)

	v10	v11	v12	v13	v14	v15	v16	v17	v18	v19	v20
▶	10672	6329	5095	3819	2726	2430	2061	1526	1464	1220	19799
	3021	2942	3181	2663	2080	1884	1608	1399	1463	1220	12797
	2249	2360	2670	2227	1818	1672	1380	1143	1097	1004	6875
*											

รูปที่ 3.7 ค่า quantized RGB ของแต่ละ Key frame ที่เก็บในฐานข้อมูล (V10-V20)

	v21	v22	v23	v24	v25	v26	v27	v28	v29	v30	v31
▶	82687	13424	17405	26145	23886	19014	18689	18433	15428	14829	14829
	63071	10095	13348	19741	17464	13841	13529	13034	8511	5904	5904
	30738	7811	10408	14575	6536	5183	4577	3470	2651	2157	2157
*											

รูปที่ 3.8 ค่า quantized RGB ของแต่ละ Key frame ที่เก็บในฐานข้อมูล (V21-V31)

	v32	v33	v34	v35	v36	v37	v38	v39	v40	v41	v42
▶	8673	8359	4128	1753	1464	664	733	673	18691	73107	14045
	2213	1949	1428	1169	983	538	725	673	11801	54120	12720
	1653	1318	845	638	562	273	273	193	5963	26757	8126
*											

รูปที่ 3.9 ค่า quantized RGB ของแต่ละ Key frame ที่เก็บในฐานข้อมูล (V32-V42)

	v43	v44	v45	v46	v47	v48	v49	v50	v51	v52	v53
▶	13232	18901	19828	21712	23600	21877	21224	16314	12033	10525	5951
	9606	13916	13297	14915	15574	13176	15086	10670	4344	2368	2073
	7495	12150	10602	6256	5897	4931	3486	2387	1626	1055	845
*											

รูปที่ 3.10 ค่า quantized RGB ของแต่ละ Key frame ที่เก็บในฐานข้อมูล (V43-V53)

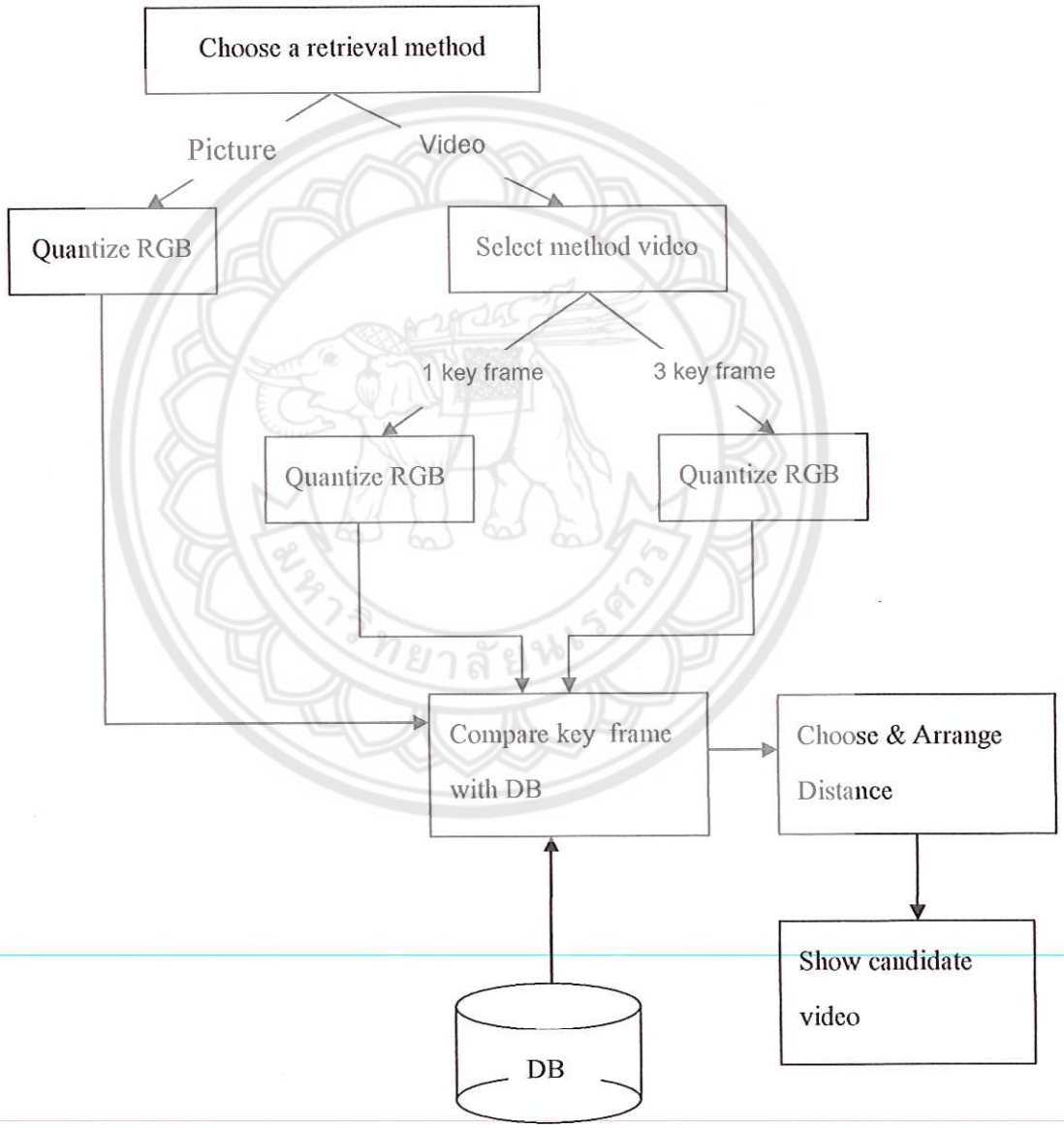
v54	v55	v56	v57	v58	v59	v60	diff
3370	2406	1602	1264	898	964	21275	0
1604	1203	945	764	712	922	14137	0
622	469	429	373	312	420	7138	0

รูปที่ 3.11 ค่า quantized RGB ของแต่ละ Key frame ที่เก็บในฐานข้อมูล (V54-V60)

3.2 การจัดการโปรแกรมการค้นหาวีดีโอ

ในส่วนนี้โปรแกรมจะทำการค้นหา Video ในฐานข้อมูลของเรา โดยสามารถเลือกที่จะทำการ search ได้ 3 รูปแบบด้วยกัน

- Picture
- Video แบบ 1 key frame
- Video แบบ 3 key frame

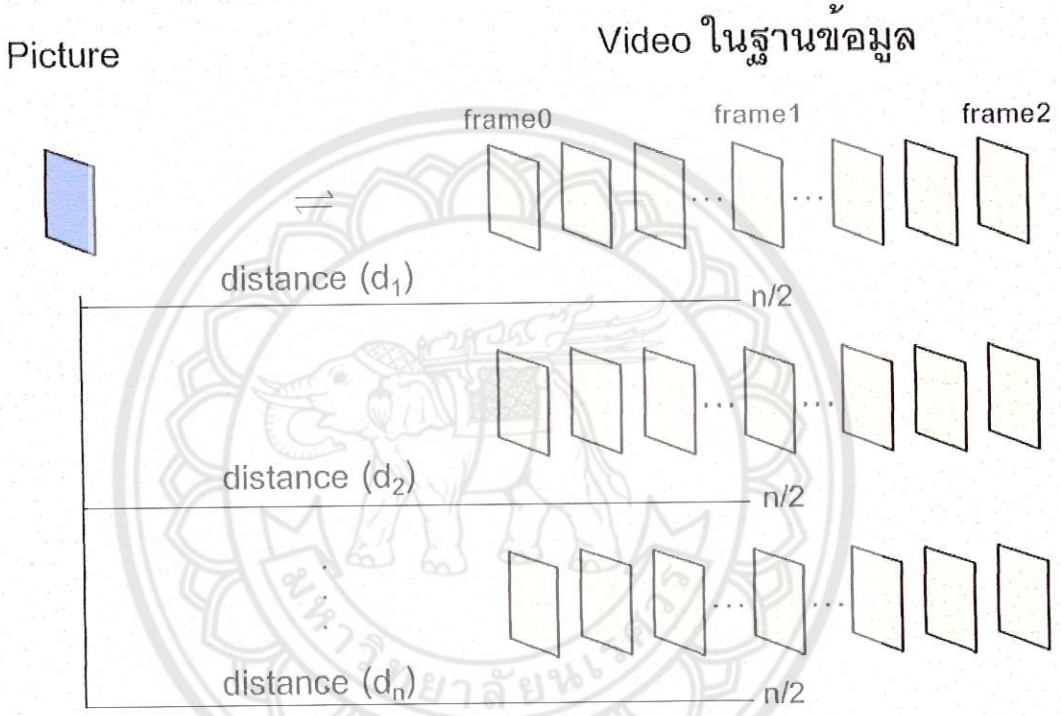


รูปที่ 3.12 แสดงขั้นตอนการค้นหา Video

3.2.1 การค้นหาวีดีโอจากไฟล์รูปภาพ (Picture)

การค้นหาไฟล์รูปภาพที่ต้องการค้นหาวีดีโอ ไปทำการแปลงเป็นค่า Histogram ซึ่งจะได้ค่าของ histogram ทั้งหมด 60 bins เช่นเดียวกับการหาค่า histogram ในฐานข้อมูล ซึ่งในฐานข้อมูลไฟล์วีดีโอ 1 ไฟล์ จะมีคีย์เฟรมทั้งหมด 3 คีย์เฟรม (frame 0 , frame 1 , frame 2)

แต่ในกรณีของการ Search จากไฟล์รูปภาพ (Picture) เราจะกำหนดการเปรียบเทียบค่า Vector เพียงแค่เฟรมเดียว นั่นคือ เฟรมตรงกลาง (frame 1) โดยใช้หลักการเปรียบเทียบ ดังนี้



รูปที่ 3.13 แสดงการเปรียบเทียบรูปภาพ (Picture) ที่นำมาค้นหากับ key frame ของวีดีโอในฐานข้อมูล ซึ่งจะเปรียบเทียบกับเฟรมตรงกลาง (Frame 1)

$$\text{distance } 1 = \sum_{i=1}^{60}\left|\vec{V}_{1\_i} - \vec{b}_i\right| = \left|\vec{V}_{1\_1} - \vec{b}_1\right| + \left|\vec{V}_{1\_2} - \vec{b}_2\right| + \left|\vec{V}_{1\_3} - \vec{b}_3\right| + \ldots + \left|\vec{V}_{1\_60} - \vec{b}_{60}\right|$$

$$\text{distance } 2 = \sum_{i=1}^{60}\left|\vec{V}_{2\_i} - \vec{b}_i\right| = \left|\vec{V}_{2\_1} - \vec{b}_1\right| + \left|\vec{V}_{2\_2} - \vec{b}_2\right| + \left|\vec{V}_{2\_3} - \vec{b}_3\right| + \ldots + \left|\vec{V}_{2\_60} - \vec{b}_{60}\right|$$

$$\text{distance } 3 = \sum_{i=1}^{60}\left|\vec{V}_{3\_i} - \vec{b}_i\right| = \left|\vec{V}_{3\_1} - \vec{b}_1\right| + \left|\vec{V}_{3\_2} - \vec{b}_2\right| + \left|\vec{V}_{3\_3} - \vec{b}_3\right| + \ldots + \left|\vec{V}_{3\_60} - \vec{b}_{60}\right|$$

$$\vdots$$

$$\text{distance } N = \sum_{i=1}^{60}\left|\vec{V}_{N\_i} - \vec{b}_i\right| = \left|\vec{V}_{N\_1} - \vec{b}_1\right| + \left|\vec{V}_{N\_2} - \vec{b}_2\right| + \left|\vec{V}_{N\_3} - \vec{b}_3\right| + \ldots + \left|\vec{V}_{N\_60} - \vec{b}_{60}\right| \quad (3.2)$$

เมื่อได้ค่า distance(d) ทั้งหมด N ค่าแล้ว จะนำค่าของ distance(d) มาเรียงลำดับจากน้อยไปมาก โดยเลือกค่า distance(d) ที่ได้ออกมาแค่ 16 ค่าด้วยกัน

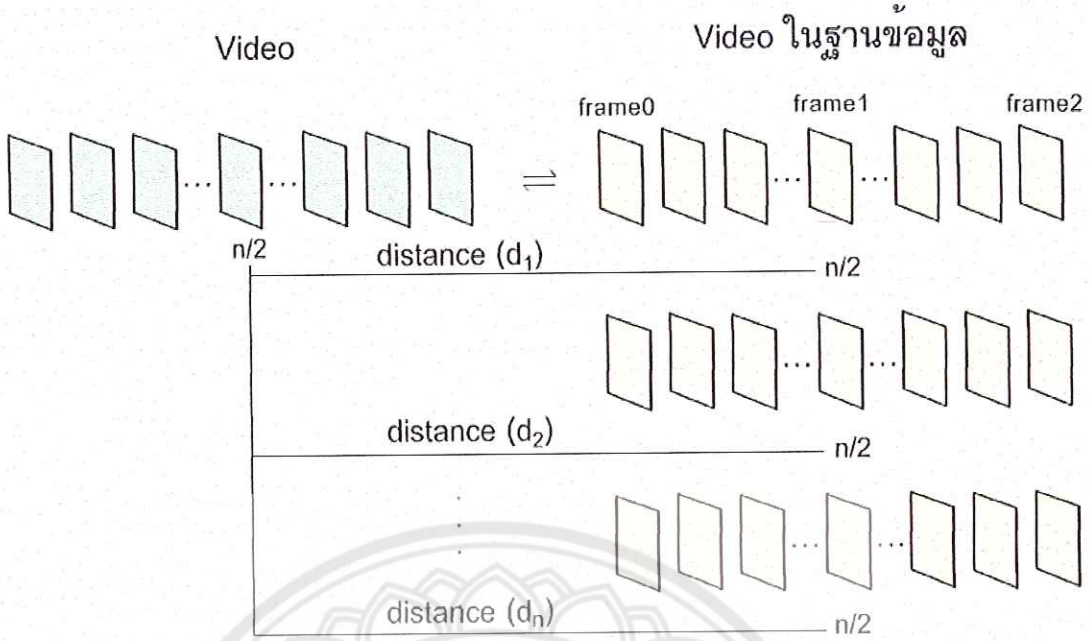
$$d1 < d2 < d3 < d4 < d5 < d6 < d7 < d8 < d9 < \ldots \quad (3.3)$$

เพราะฉะนั้นเราจะนำค่า Path ของ frame ที่เกิดความแตกต่าง (distance) ออกมาแสดงเป็นผลการ Search วีดีโอ

3.2.2 การค้นหาไฟล์วีดีโอ โดยใช้วีดีโอเปรียบเทียบ 1 key frame

ในขั้นตอนนี้จะต้องนำ Clip video ที่ต้องการที่จะค้นหามาทำการหา Key frame ซึ่งในการค้นหาจาก Clip video นี้เราจะใช้ Key frame ตรงกลางของวีดีโอมาเปรียบเทียบค่า Histogram กับ Key frame ที่ 1 (ซึ่งเป็นเฟรมตรงกลาง) ของวีดีโอในฐานข้อมูล

โดยหลักการเปรียบเทียบจะใช้หลักการเดียวกับการเปรียบเทียบความแตกต่าง (Distance) ของ Picture เพราะฉะนั้นจะใช้สมการการเปรียบเทียบเหมือนสมการที่ 3.1 กับสมการที่ 3.2



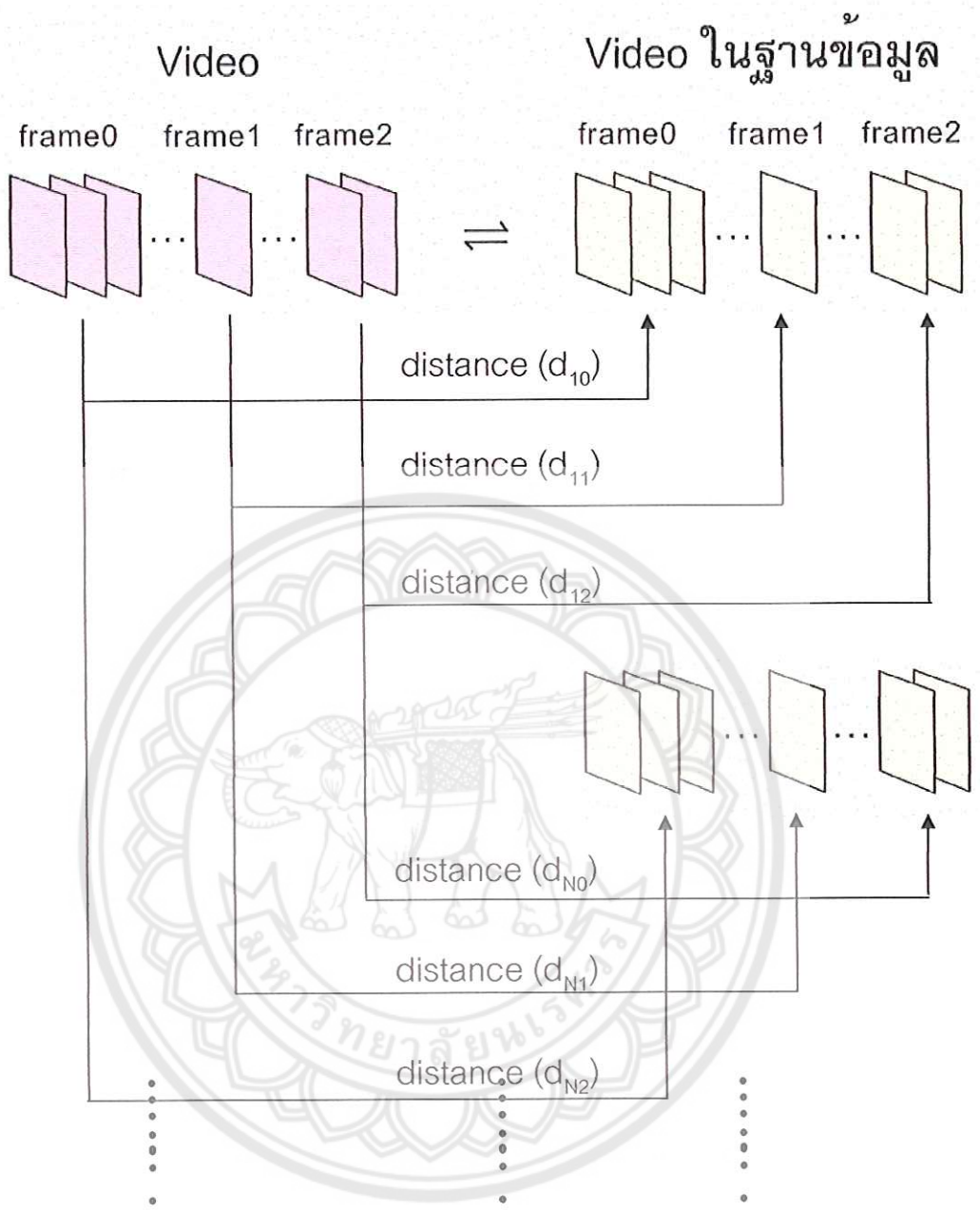
รูปที่ 3.14 แสดงการเปรียบเทียบ Key frame ของ Clip Video ที่นำมาค้นหากับ key frame ของวีดีโอในฐานข้อมูล โดยเปรียบเทียบ 1 key frame

เมื่อทำการเปรียบเทียบแล้วก็นำค่า Distance มาเรียงลำดับเช่นเดียวกับสมการที่ 3.3 จึงได้ผลการค้นหาไฟล์วีดีโอจาก Clip video ตามต้องการ

3.2.3 การค้นหาไฟล์วีดีโอ โดยใช้วีดีโอเปรียบเทียบ 3 key frame

เราต้อง Clip Video ที่ต้องการค้นหามาหา key frame ก่อน โดยในระดับนี้เราจะหา key frame ทั้งหมด 3 frame ด้วยกัน นั่นคือ frame0 (หน้า) , frame1 (กลาง) และ frame2 (หลัง) แล้วนำ key frame ที่ได้ทั้งหมด 3 frame ไปหาค่า histogram โดยหลักการจะเหมือนกับขั้นตอนการเตรียมข้อมูลในฐานข้อมูล ที่จะต้องมี 3 key frame ต่อ 1 วีดีโอ

แล้วในการเปรียบเทียบของ key frame แต่ละ frame ของ Clip video จะเปรียบเทียบค่าความแตกต่าง Distance กับ ทุกๆ key frame ในฐานข้อมูล ดังรูปที่ 3.15



รูปที่ 3.15 แสดงการเปรียบเทียบ Key frame ของ Clip Video ที่นำมาค้นหากับ key frame ของวีดีโอในฐานข้อมูล โดยเปรียบเทียบ 3 key frame

นำค่า disttance (d_{N0}) , disttance (d_{N1}) , disttance (d_{N2}) รวมกันเป็นผลบวก ดังสมการที่ 3.4 แล้ว

$$
\begin{aligned}
distance1 &= | d_{10} + d_{11} + d_{12} | \\
distance2 &= | d_{20} + d_{21} + d_{22} | \\
distance3 &= | d_{30} + d_{31} + d_{32} | \\
&\vdots \\
distanceN &= | d_{N0} + d_{N1} + d_{N2} |
\end{aligned}
\tag{3.4}
$$

นำมาเรียงลำดับ จากน้อยไปหามาก ดังสมการที่ 3.3

บทที่ 4

ผลการทดลอง

ในบทนี้จะทำการทดลองค้นหา Video ในฐานข้อมูลที่ได้สร้างไว้แล้ว โดยจะใช้หลักการเปรียบเทียบผลของค่า Precision ที่นี่ โดยค่า Precision เป็นค่าที่ได้จากการคำนวณดังต่อไปนี้

$$\% \text{ ความถูกต้อง} = (\text{จำนวนวีดีโอที่ถูกเลือก} / \text{จำนวนวีดีโอทั้งหมด}) \times 100 \qquad (4.1)$$

$$\text{เช่น} \quad \% \text{ ความถูกต้อง} = \left(\frac{5}{16}\right) \times 100 = 31.25\%$$

แล้วนำค่า Precision (ความถูกต้อง) ของการ Query ทั้งหมด มารวมกันแล้วหาค่าเฉลี่ย ก็จะได้เป็น Precision ของ Method นั้นๆ ซึ่งเมื่อนำไปเปรียบเทียบกับ Method อื่นๆ ก็จะเห็นความแตกต่างของการค้นหาได้จากการสร้างกราฟ

ในการทดลองเราจะแบ่งออกเป็น 3 แบบด้วยกัน คือ

1) การ Search วีดีโอจากไฟล์รูปภาพ
2) การ Search วีดีโอจากไฟล์วีดีโอ
   - 2.1) 1 Key frame
   - 2.2) 3 key frame

4.1 สิ่งที่ต้องเตรียมก่อนทำการทดลอง

1) วีดีโอในฐานข้อมูลทั้งหมดประมาณ 5864 ไฟล์ ซึ่งเป็นไฟล์ .AVI ทั้งหมด
2) วีดีโอตัวอย่างที่จะนำมา Search จำนวน 30 ไฟล์ โดยจะต้องเป็น ไฟล์ .AVI เท่านั้น
3) รูปภาพที่จะนำมา Search จำนวน 30 ไฟล์
4) โปรแกรม Search วีดีโอ ที่พัฒนามาจากภาษา C#.Net
5.) โปรแกรมหาค่า Key frame เพื่อเก็บไว้ในฐานข้อมูล

4.2 การทดลองค้นหาวีดีโอ จากไฟล์รูปภาพ

1) นำไฟล์รูปภาพตัวอย่างที่ต้องการค้นหาทั้งหมด 30 ไฟล์ มาทำการ Query
2) โปรแกรมจะทำการเปรียบเทียบค่าเวคเตอร์ระหว่าง ไฟล์รูปภาพ กับคีย์เฟรม ในฐานข้อมูล
3) เมื่อเปรียบเทียบค่า Distance $d1 < d2 < d3 < d4 < d5 < d6 < d7 < d8 < d9 <.....< d16$
4) โชว์ผลการค้นหา ทั้งหมด 16 วีดีโอ โดยเรียงลำดับค่า Distance จาก MV1 , MV2 , MV3 ไปจนถึง MV16
5) ทำการเลือกวีดีโอที่เหมือน โดยการทดสอบ ในกรณีนี้เราจะพิจารณาองค์ประกอบของคลิปวีดีโอว่ามีส่วนที่เกี่ยวข้องภายในภาพอยู่ในคลิปวีดีโอหรือไม่

รูปที่ 4.1 แสดงการ Query จากไฟล์รูปภาพตัวอย่าง

จากตัวอย่างในรูปที่ จะได้วีดีโอที่ถูกต้อง 6 ไฟล์ นำมาคำนวณหาค่า Precision จะได้ว่า

$$\%\text{Precision} = \left(\frac{11}{16}\right) \times 100 \quad = 68.75\ \%$$

ฉะนั้นเราจะนำค่าของ Precision ในแต่ละไฟล์รูปภาพ มาบันทึกผลการทดลองได้ดังนี้

ตารางที่ 4.1 บันทึกการทดลองค้นหาวีดีโอจากไฟล์รูปภาพ

ลำดับ	ชื่อไฟล์ภาพ	จำนวนคลิปวีดีโอที่เลือก	ตำแหน่งคลิปวีดีโอที่เลือก																Precision
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1	Pic.1	9	★	*	*	*	*	*		*			*				*		56.25%
2	Pic.2	9	★	*	*	*	*	*					*	*	*				56.25%
3	Pic.3	10	*	*	*	*		*	*		*	*					*	*	62.5%
4	Pic.4	3	*	★														*	18.75%
5	Pic.5	5	★	*	*		*		*										31.25%
6	Pic.6	5	★			*			*		*	*							31.25%
7	Pic.7	7	*	*	★				*	*			*					*	43.75%
8	Pic.8	12		*		*	*	*	*	★	*	*		*		*	*	*	75%
9	Pic.9	7	★	*		*	*		*			*		*					43.75%
10	Pic.10	8	*	*		★	*		*					*	*		*		50%
11	Pic.11	10	★	*	*	*	*		*	*			*	*			*		62.5%
12	Pic.12	10	*	★	*	*	*	*	*		*	*			*				62.5%
13	Pic.13	7	★	*	*		*			*		*			*				43.75%
14	Pic.14	7				*		★		*	*	*		*				*	43.75%
15	Pic.15	11	★	*	*	*	*	*		*	*	*		*	*				68.75%
16	Pic.16	5	★	*	*			*	*										31.25%
17	Pic.17	8	*	★	*	*		*		*		*		*					50%
18	Pic.18	8	★		*				*		*	*	*	*	*				50%
19	Pic.19	4	★		*					*		*							25%
20	Pic.20	4	★		*	*	*												25%
21	Pic.21	3	*			*						★							18.75%
22	Pic.22	7					★	*	*	*		*		*	*				43.75%
23	Pic.23	3					★	*									*		18.75%
24	Pic.24	3	★		*								*						18.75%
25	Pic.25	14	*	*	★	*	*	*	*			*	*	*	*	*	*	*	87.5%
26	Pic.26	7	*	★	*	*	*	*									*		43.75%
27	Pic.27	3		★		*												*	18.75%
28	Pic.28	10	*	*	*			*		*	*	*			★	*	*		62.5%
29	Pic.29	5		★				*					*		*			*	31.25%
30	Pic.30	8					*	*	*	*	*			*		*	★		50%
SUM		212	23	20	18	17	16	16	14	12	9	15	8	12	10	4	10	8	1325.00
AVG		7.06																	44.16%

4.3 การทดลองค้นหาวีดีโอจากไฟล์วีดีโอตัวอย่าง

ในขั้นตอนนี้จะทำการทดลอง 2 Method ด้วยกัน

1) การ Search โดยใช้ 1 key frame/video

2) การ Search โดยใช้ 3 key frame/video

แล้วนำผลการทดลองทั้ง 2 Method มาทำการเปรียบเทียบว่าประสิทธิภาพของการค้นหาไฟล์วีดีโอ แตกต่างกันอย่างไร

4.3.1 การ Search โดยใช้ 1 key frame/video

1) นำ Video ตัวอย่างที่ต้องการค้นหามา 30 ไฟล์ แล้วทำการ Query

2) โปรแกรมจะทำการเปรียบเทียบค่าเวคเตอร์ระหว่าง คีย์เฟรมของวีดีโอที่เตรียมไว้ (เฟรมกลาง) กับคีย์เฟรมในฐานข้อมูลเฟรมกลางเหมือนกัน (frame 1)

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

4) โชว์ผลการค้นหา ทั้งหมด 16 วีดีโอ โดยเรียงลำดับค่า Distance จาก MV1 , MV2 , MV3 ไปจนถึง MV16

5) ทำการเลือกวีดีโอที่เหมือน โดยการทดสอบ ในกรณีนี้จะทำการเลือกวีดีโอที่เรียงลำดับตามค่า Distance ไว้แล้ว โดยดูจากความเหมือนกับวีดีโอที่นำมา search

รูปที่ 4.2 แสดงการ Query จากไฟล์วีดีโอ โดย Method 1 key frame

เพราะฉะนั้นการคำนวณหาค่า Precision จึงคิดแบบเดียวกับการคำนวณของไฟล์รูปภาพ แล้วบันทึกผลการทดลองได้ดังตาราง

ตารางที่ 4.2 บันทึกการทดลองการค้นวีดีโอจากไฟล์วีดีโอ 1 Key frame

ลำดับ	ชื่อไฟล์คลิปวีดีโอ	จำนวนคลิปวีดีโอที่เลือก	ตำแหน่งคลิปวีดีโอที่เลือก																Precision (%)
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1	1_patriot_clip1322	12	*	*	*	*	*	*	*	★	*	*	*	*					75%
2	2_patriot_clip533	10	*	*	*		★	*	*	*	*		*			*			62.5%
3	3_junaid1_clip7	9		★	*	*	*		*				*	*	*			*	56.25%
4	4_junaid2_clip8	8	★	*				*		*	*		*	*	*				50%
5	5_junaid5_clip39	10	★	*	*	*	*	*		*					*	*		*	62.5%
6	6_titanic_clip482	7	*					*		*		*				*	*	★	43.75%
7	7_junaid14_clip11	10	★	*	*	*	*		*	*	*	*					*		62.5%
8	8_patriot_clip832	10	*	*	*	*	★	*	*					*		*	*		62.5%
9	9_junaid(601)	7	*			*		*			★		*			*	*		43.75%
10	10_postman_clip1629	4	★	*			*			*									25%
11	11_patriot_clip492	11	★	*	*	*	*			*	*	*		*			*	*	68.75%
12	12_postman_clip13	7	*	*	*	*	*	★			*								43.75%
13	13_junaid (463)	11	★	*	*	*	*	*		*	*		*	*	*				68.75%
14	14_junaid (590)	8	*	*	*	★	*	*				*					*		50%
15	15_postman_clip1485	10	★	*	*	*	*	*	*	*			*		*				62.5%
16	16_patriot_clip1299	5	*	★		*					*	*							31.25%
17	17_patriot_clip130	6		*								*	*	★			*	*	37.5%
18	18_junaid8_clip2	6	*	*	★	*		*	*										37.5%
19	19_patriot_clip150	8		*		★	*		*		*	*				*	*		50%
20	20_junaid8_clip29	9	★	*	*	*		*	*			*			*	*			56.25%
21	21_junaid2_clip9	10	★	*	*	*	*	*		*				*		*		*	62.5%
22	22_patriot_clip170	12	*		★	*	*	*	*	*	*	*	*	*			*			75%
23	23_patriot_clip164	6				*	*	*	*			*					*		37.5%
24	24_postman_clip236	8	*	*	★	*	*		*				*					*	50%
25	25_junaid1_clip22	13	*	*	★	*	*	*	*	*	*	*	*	*	*				81.25%
26	26_titanic_clip359	9	*	*		★		*	*	*	*	*			*				56.25%
27	27_titanic_clip824	7	*	*	*	*	★								*		*		43.75%
28	28_junaid7_clip40	10	★	*	*	*	*				*	*	*		*	*			62.5%
29	29_ad2_clip22	8	★	*	*		*	*			*			*		*			50%
30	30_titanic_clip875	11	*	*	★	*	*					*	*	*	*	*	*		68.75%
SUM		262	26	26	21	24	22	19	14	14	15	15	13	12	11	11	12	7	1638.00
AVG		8.73																	54.6%

4.3.2 การ Search โดยใช้ 3 key frame/video

นำไฟล์วีดีโอตัวอย่างที่ใช้ในการทดลองขั้นที่ 4.3.1 มาทำการทดลองใน Method นี้ด้วย เพื่อทำการเปรียบเทียบค่าความแตกต่างที่ได้อย่างชัดเจน แต่ในกรณีนี้เราจะเปรียบเทียบ key frame ถึง 3 อันด้วยกัน คือ เฟรมหน้า , เฟรมกลาง , เฟรมหลัง (frame0 , frame1 , frame2)

ผลการทดลองโดยใช้ตัวอย่าง Video เช่นเดียวกับขั้นตอนที่ 4.3.1 โดยผลการทดลองจะเป็นดังรูปด้านล่างนี้

รูปที่ 4.3 แสดงการ Query จากไฟล์วีดีโอโดย Method 3 key frame

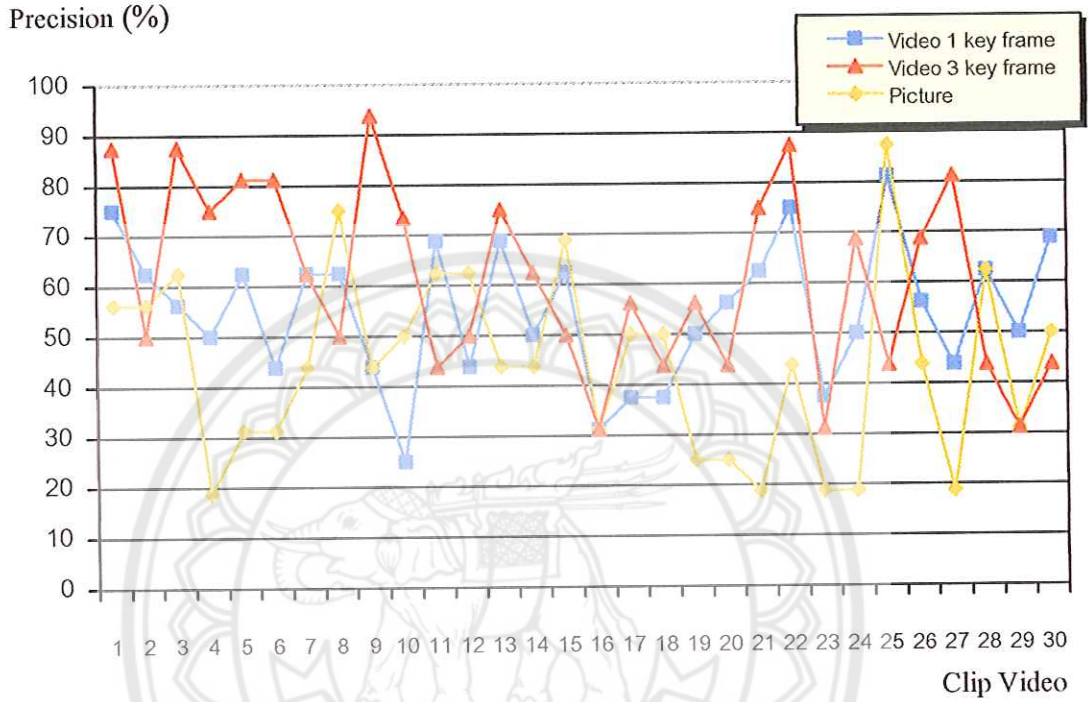
บันทึกออกมาเป็นตารางได้ดังตารางที่ 4.3

ตารางที่ 4.3 บันทึกการทดลองการค้นวีดีโอจากไฟล์วีดีโอ 3 Key frame

ลำดับ	ชื่อไฟล์คลิปวีดีโอ	จำนวนคลิปวีดีโอที่เลือก	ตำแหน่งคลิปวีดีโอที่เลือก																Precision (%)
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1	1_patriot_clip1322	14	★	*	*	*	*	*	*	*	*	*	*	*	*		*		87.5%
2	2_patriot_clip533	8	★	*	*	*	*		*			*	*						50%
3	3_junaid1_clip7	14	★	*	*	*	*	*	*	*		*	*	*	*	*	*		87.5%
4	4_junaid2_clip8	12	★	*	*	*	*	*	*	*		*	*	*	*				75%
5	5_junaid5_clip39	13	★	*	*	*		*	*	*	*	*	*	*	*	*			81.25%
6	6_titanic_clip482	13	★	*	*	*	*	*	*	*	*		*	*		*		*	81.25%
7	7_junaid14_clip11	10	★	*	*	*		*	*	*		*	*	*					62.5%
8	8_patriot_clip832	8	★	*	*	*	*	*	*		*					*			50%
9	9_junaid(601)	15	★	*	*	*	*	*	*	*	*	*	*	*	*	*	*		93.75%
10	10_postman_clip1629	6	★		*	*				*		*					*		37.5%
11	11_patriot_clip492	7	★	*	*	*	*	*	*										43.75%
12	12_postman_clip13	8	★	*	*	*	*	*	*	*									50%
13	13_junaid (463)	12	★	*	*	*	*	*	*	*	*			*	*			*	75%
14	14_junaid (590)	10	★	*	*	*	*	*	*	*	*		*						62.5%
15	15_postman_clip1485	8	★		*	*	*	*		*	*		*						50%
16	16_patriot_clip1299	5	★	*	*	*	*							*			*		31.25%
17	17_patriot_clip130	9	★	*	*	*	*	*	*		*					*			56.25%
18	18_junaid8_clip2	7	★			*		*		*		*	*		*				43.75%
19	19_patriot_clip150	9	★	*	*	*	*	*	*	*			*						56.25%
20	20_junaid8_clip29	7	★	*		*	*	*		*				*					43.75%
21	21_junaid2_clip9	12	★	*	*	*	*	*		*			*	*	*	*		*	75%
22	22_patriot_clip170	14	★	*	*	*	*	*	*	*	*	*	*		*	*		*	87.5%
23	23_patriot_clip164	5	★		*	*		*	*										31.25%
24	24_postman_clip236	11	★	*	*	*	*	*	*	*		*		*			*		68.75%
25	25_junaid1_clip22	7	★	*	*		*	*	*	*									43.75%
26	26_titanic_clip359	11	★	*	*		*	*		*			*	*	*	*	*		68.75%
27	27_titanic_clip824	13	★	*	*	*	*	*	*	*		*	*	*	*		*		81.25%
28	28_junaid7_clip40	7	★	*	*	*	*	*		*									43.75%
29	29_ad2_clip22	5	★	*	*	*				*									31.25%
30	30_titanic_clip875	7	★	*			*			*		*		*				*	43.75%
SUM		287	30	26	27	27	24	25	20	24	10	13	16	15	11	9	8	5	1793.75
AVG		9.56																	59.79%

4.4 เปรียบเทียบผลการทดลอง

นำผลการทดลองที่หาค่าความถูกต้อง Precision ของคลิปวีดีโอตัวอย่างแต่ละคลิปมา plot กราฟเปรียบเทียบการค้นหาทั้ง 3 วิธี จะเห็นความแตกต่างของการทดลองโดยที่เส้นกราฟที่เกิดจากการค้นหาแบบ ดังรูปที่ 4.4



รูปที่ 4.4 กราฟเปรียบเทียบการทำงานของแต่ละ Method

เมื่อได้ผลการเปรียบเทียบการทดลองในแต่ละขั้นตอนออกมาแล้ว จะสังเกตเห็นถึงความแตกต่างของแต่ละ Method ได้อย่างชัดเจน

บทที่ 5
บทสรุป

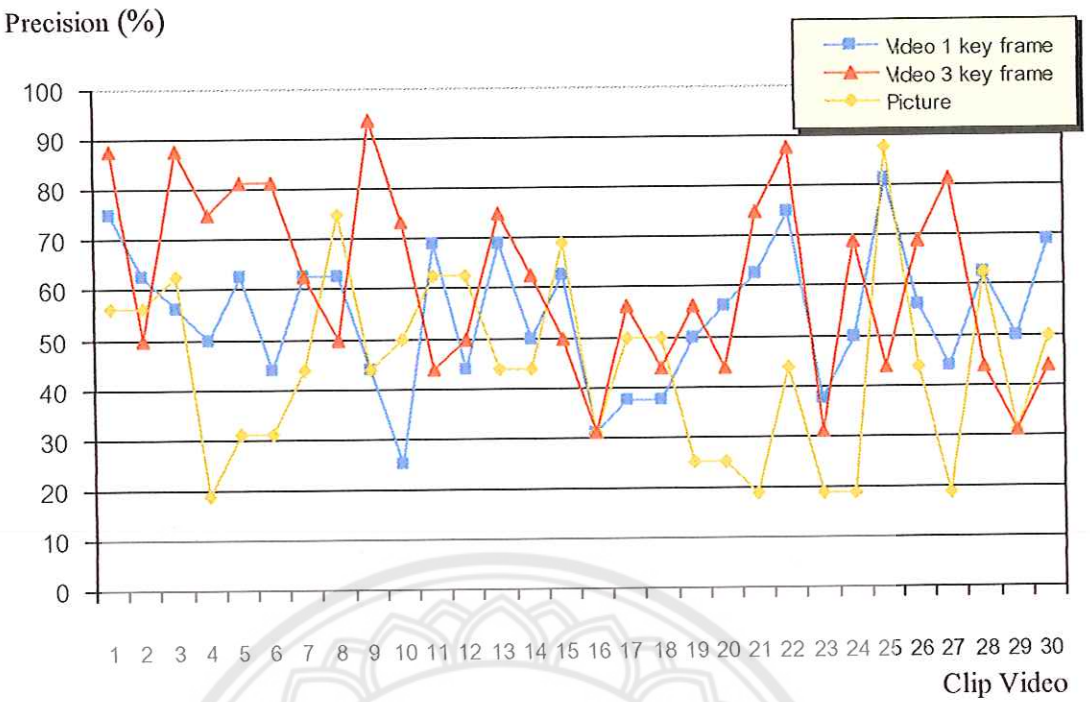
5.1 สรุปผลการทดลอง

เราจะได้ผลการทดลองดังตารางที่ 4.1 , 4.2 และ 4.3 เมื่อเปรียบเทียบค่าความแตกต่างระหว่างการค้นหาทั้ง 3 วิธี ทำให้เราสามารถสรุปได้ว่า

- การค้นหาจากไฟล์รูปภาพจะสามารถพบไฟล์คลิปวีดีโอที่มีองค์ประกอบภายในคลิปวีดีโอเหมือนกับไฟล์รูปภาพที่นำมา Search โดยภาพที่ถูกเลือกอาจจะมีภาพที่ถูกต้องตรงกับคีย์เฟรมในฐานข้อมูลเลย หรืออาจจะมีองค์ประกอบเหมือนกันภายในคลิปวีดีโอนั้นๆ และจากตารางที่ 4.1 จะเห็นได้ว่า ตรงสัญลักษณ์ ★ ก็คือเป็นคลิปวีดีโอที่ตรงกับไฟล์ภาพที่นำมาค้นหา แต่จะมีบางคลิปวีดีโอที่อาจจะไม่มีสัญลักษณ์ ★

- การค้นหาแบบไฟล์วีดีโอ 1 คีย์เฟรม จะสามารถพบไฟล์วีดีโอที่มีลักษณะใกล้เคียงกับไฟล์วีดีโอตัวอย่าง โดยเลือกจากองค์ประกอบภายในคลิปวีดีโอที่เหมือนๆกัน ดังตารางที่ 4.2 โดยที่ตรงสัญลักษณ์ ★ นั้นเป็นตัวแทนของคลิปวีดีโอในฐานข้อมูล ที่มีความเหมือนกับคลิปวีดีโอที่นำมาค้นหามากที่สุด ซึ่ง สัญลักษณ์ ★ อาจจะกระจายไปทั่วๆไป แสดงให้เห็นถึงว่าการใช้การเปรียบเทียบ 1 คีย์เฟรม อาจจะให้ผลการทดลองที่ไม่สมบูรณ์เท่าที่ควรนัก

- การค้นหาแบบไฟล์วีดีโอ 3 คีย์เฟรม จะสามารถพบไฟล์วีดีโอที่มีลักษณะใกล้เคียงกับไฟล์วีดีโอตัวอย่าง โดยผลที่ออกมาตามตาราง 4.3 จะสามารถเห็นถึงความถูกต้องของการ ค้นหาได้มากกว่า การค้นหาแบบ 1 คีย์เฟรม โดยสังเกตจากสัญลักษณ์ของ ★ เกิดที่บริเวณช่องแรกทุกช่องซึ่งเป็นสัญลักษณ์ที่เกิดจากการเปรียบเทียบที่ถูกต้องที่สุด ทำให้เราสรุปได้ว่าการค้นหาโดยวิธีนี้จะให้ความถูกต้องได้มากที่สุด



รูปที่ 5.1 กราฟเปรียบเทียบการทำงานของแต่ละ Method

จากกราฟจะเห็นว่า เส้นกราฟที่เกิดจากวิธีค้นหาแบบ Video 3 key frame (เส้นสีแดง) มีค่า % precision ที่เกิดจากการ Query วีดีโอตัวอย่างทั้ง 30 ตัวอย่าง สูงกว่าค่า % precision ที่เกิดจากการค้นหาของ Video 1 key frame (เส้นสีฟ้า) ทำให้เราสามารถที่จะสรุปผลการเปรียบเทียบการค้นหาได้ในลักษณะหนึ่งนั้น

และในส่วนของการค้นหาจากไฟล์รูปภาพ (เส้นสีเหลือง) ซึ่งเราสามารถที่จะเปรียบเทียบผลการทดลองกับการค้นหากับ การค้นหาของ Video 1 key frame (เส้นสีฟ้า) ที่มีลักษณะการเปรียบเทียบเฟรมแค่เฟรมเดียวเหมือนกันนั้นคือ เฟรมตรงกลาง ผลที่ได้ออกมาเส้นกราฟจะมีระดับที่ใกล้เคียงกัน

เมื่อเราพิจารณาและเปรียบเทียบค่าเฉลี่ยของค่า Precision ในแต่ละ Method จะได้ดังตารางที่ 5.1

ตารางที่ 5.1 เปรียบเทียบค่า Average Precision ของ 3 methods

Methods	Average Precision
Picture	44.16%
Video 1 key frame	54.6%
Video 3 key frame	59.79%

จะเห็นได้ว่า ค่าเฉลี่ยของการ Search แบบรูปภาพ (Picture) จะมีค่า % precision ที่น้อยที่สุด และการ search แบบ Video 1 key frame จะให้ค่า % precision ที่ใกล้เคียงกับ 3 key frame แต่สำหรับผลการ Search แบบ Video 3 key frame จะให้ค่า Precision ที่สูงที่สุด เพราะเนื่องจากในการค้นหาจะใช้ Key frame ทั้งหมด 3 Key frame (Frame0 , Frame1 , Frame2) ในการเปรียบเทียบ จึงทำให้ค่า % precision มีค่าสูงกว่า 2 วิธีแรก ความถูกต้องจึงมากยิ่งขึ้น ต่อไปเป็นการเปรียบเทียบตำแหน่ง MV ที่เลือก

ตารางที่ 5.2 เปรียบเทียบตำแหน่งของวีดีโอที่เลือกของแต่ละ Method

ตำแหน่งวีดีโอ / Method	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Picture	23	20	18	17	16	16	14	12	9	15	8	12	10	4	10	8
Video 1 key frame	26	26	21	24	22	19	14	14	15	15	13	12	11	11	12	7
Video 3 key frame	30	26	27	27	24	25	20	24	10	13	16	15	11	9	8	5

จากตารางที่ 5.2 จะเห็นได้ว่า วิธี การ Search แบบ Video 3 key frame จะให้ผลของตำแหน่ง MV1 ครบทุก Query ส่วนผลการ Search ของ Picture (รูปภาพ) กับ Video 1 key frame จะได้ผลการ search ที่ MV1 ใกล้เคียงกัน

ส่วนในกรณีของลำดับวีดีโออื่นๆ จะมีผลการ Search เรียงๆ ไปตามลำดับ

จากการทดลองทั้งหมดจะเห็นได้ว่า เมื่อเราเปรียบเทียบผลการ Search ทั้งหมด ทำให้ได้ผลที่ถูกต้องตรงตามความต้องการได้ผลพอสมควร โดยเฉพาะการเปรียบเทียบคีย์เฟรม 3 คีย์เฟรม จะทำให้ได้ผลถูกต้องยิ่งขึ้น แต่ความคลาดเคลื่อนอาจเกิดขึ้นได้บ้าง ซึ่งอาจจะเกิดจากสาเหตุที่ว่าจำนวนข้อมูลในฐานข้อมูลอาจไม่มากพอ หรือการเปรียบเทียบค่าสีอาจใช้ bin ไม่เพียงพอ บางทีอาจต้องใช้ ทฤษฎีอื่นเข้ามาช่วย เช่น Texture Analysis หรือ Shapes Analysis เพื่อให้ได้ผลที่ถูกต้องมากยิ่งขึ้น

5.2 ปัญหาที่พบ

1. เนื่องจากโปรแกรมที่พัฒนามาเพื่อทำการค้นหาไฟล์วีดีโอนี้ สามารถใช้ได้เฉพาะไฟล์วีดีโอที่เป็น .AVI เท่านั้น เพราะฉะนั้นถ้าหากใช้ไฟล์ชนิดอื่นจะทำให้ผลการทดลองเกิดการผิดพลาดได้
2. ถ้าหากคลิปวีดีโอในฐานข้อมูลมีไม่เพียงพอผลการค้นหาก็จะไม่เป็นไปตามที่ต้องการ แต่ถ้าหากข้อมูลมีมากเกินไปก็จะทำให้การค้นหานั้นต้องใช้เวลานานพอสมควรเพราะต้องทำการเปรียบเทียบกับคีย์เฟรมของคลิปวีดีโอทุกอัน

5.3 ข้อเสนอแนะ

1. ควรจะมีข้อมูลในฐานข้อมูลที่เพียงพอ เพราะผลการค้นหาที่ไม่ตรงนั้นอาจเกิดมาจากข้อมูลใน ฐานข้อมูลที่ไม่เพียงพอ
2. หากมีการพัฒนาต่อไป ควรจะศึกษาในส่วนของ Texture และ Shapes Analysis เพิ่มเติม
3. ควรจะพัฒนาให้สามารถค้นหากับไฟล์วีดีโอได้ทุกๆชนิด และค้นหากับภาพได้ทุกๆขนาด

เอกสารอ้างอิง

[1] Corinna John . "A Simple C# Wrapper for the AviFile Library." [Online], Available: http://www.codeproject.com/. 2004.

[2] Daniel Strigl. "DirectShow MediaPlayer in C#." [Online]. Available: http://www.codeproject.com/. 2002.

[3] Eric-Paul. "ImageConverter - Converts images to a specific image format, changing sizes on the flow." [Online]. Available: http://www.codeproject.com/. 2004.

[4] สงกรานต์ ทองสว่าง. MySQL ระบบฐานข้อมูลสำหรับอินเตอร์เน็ต.กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น . 2545 .

[5] สันติ ศรีลาศักดิ์. วินัย สุขอารีย์ชัย. รันเว็บบน .NET Framework . บริษัทออฟเซ็ทเพรส จำกัด . 2546.

ภาคผนวก ก

Source Code การจัดการฐานข้อมูล

ก-1 การกำหนดการรับค่าไฟล์วีดีโอชนิด .avi

```
private void button1_Click(object sender, System.EventArgs e)
{
      String fileName = GetFileName("Videos
      (*.avi)|*.avi;*.mpe;*.mpeg");
            if (fileName != null)

                  {
                        textBox1.Text = fileName;

                  }
}
```

ก-2 รับค่า path ไฟล์วีดีโอจาก textBox1 แล้วทำการตัดวีดีโอออกเป็นเฟรมเก็บไว้ที่ path เดียวกับ textBox4

```
private void button2_Click(object sender, System.EventArgs e)
      {


            AviManager aviManager = new AviManager(textBox1.Text,
true);
            VideoStream stream = aviManager.GetVideoStream();
            stream.GetFrameOpen();


            String path = textBox4.Text+"\\";
            for (int n = 0; n < stream.CountFrames; n++)
            {
                  stream.ExportBitmap(n, path +n.ToString() +
".bmp");

            }

            stream.GetFrameClose();
            aviManager.Close();
            fillLV();

      }
```

ก-3 การหาค่า color histogram แล้วนำมาทำ vector ของสี RGB โดยแต่ละค่าสีจะเก็บทั้งหมด 20 bin และแต่ละ คีย์เฟรมจะมีค่า vector 60 bin ที่เกิดจากการต่อกันของ สี แดง ,เขียว,เหลือง

```
private void fillLV()
{

      string a = textBox1.Text;
      AviManager aviManager = new AviManager(textBox1.Text, true);
      VideoStream stream = aviManager.GetVideoStream();
      int i,j,c1,e;
      double
br1=0,br2=0,br3=0,br4=0,br5=0,br6=0,br7=0,br8=0,br9=0,br10=0,br11=0,b
r12=0,br13=0,br14=0,br15=0,br16=0,br17=0,br18=0,br19=0,br20=0;
      double
bg1=0,bg2=0,bg3=0,bg4=0,bg5=0,bg6=0,bg7=0,bg8=0,bg9=0,bg10=0,bg11=0,b
g12=0,bg13=0,bg14=0,bg15=0,bg16=0,bg17=0,bg18=0,bg19=0,bg20=0;
      double
bb1=0,bb2=0,bb3=0,bb4=0,bb5=0,bb6=0,bb7=0,bb8=0,bb9=0,bb10=0,bb11=0,b
b12=0,bb13=0,bb14=0,bb15=0,bb16=0,bb17=0,bb18=0,bb19=0,bb20=0;
      string fn;
      int A=352;  //Width
      int B=288;  //Height


     int c=stream.CountFrames;
      c1=c/2;


      int[] RGBredp1 = new int[A*B];//
      int[] RGBred1 = new int[A];
      int[] RGBgreenp1 = new int[A*B];//
      int[] RGBgreen1 = new int[A];
      int[] RGBbluep1 = new int[A*B];//
      int[] RGBblue1 = new int[A];
      int[] RGB = new int[3*A*B];
      int d=1;

      double[] binred = new double[60];
      double[] bingreen = new double[60];
      double[] binrblue = new double[60];
      double[] bin = new double[60];
      byte pixelColor;


      // การหาค่า Histogram ของ keyframe
      for(e=0;e<3;e++)
      {
            string path = (textBox4.Text+"\\"+d.ToString()+".bmp");
            txtFileNames.Text += path+"\r\n";
            Bitmap img = new Bitmap(path);

      //pic1
            for (i=0; i<B; i++)
            {
                  for (j=0; j<A; j++)
                  {
```

```
//red*****************************
                pixelColor = img.GetPixel(j,i).R;

                if(pixelColor>=0&&pixelColor<12)
                {
                      br1=br1+1;
                }
                else if(pixelColor>=13&&pixelColor<24)
                {
                      br2=br2+1;
                }
                else if(pixelColor>=25&&pixelColor<36)
                {
                      br3=br3+1;
                }
                else if(pixelColor>=37&&pixelColor<48)
                {
                      br4=br4+1;
                }
                else if(pixelColor>=49&&pixelColor<60)
                {
                      br5=br5+1;
                }
                else if(pixelColor>=61&&pixelColor<72)
                {
                      br6=br6+1;
                }
                else if(pixelColor>=73&&pixelColor<84)

                      br7=br7+1;
                }
                else if(pixelColor>=85&&pixelColor<96)
                {
                      br8=br8+1;
                }
                else if(pixelColor>=97&&pixelColor<108)
                {
                      br9=br9+1;
                }
                elseif(pixelColor>=109&&pixelColor<120)
                {
                      br10=br10+1;
                }
                else if(pixelColor>=121&&pixelColor<132)
                {
                      br11=br11+1;
                }
                else if(pixelColor>=133&&pixelColor<144)
                {
                      br12=br12+1;
                }
                else if(pixelColor>=145&&pixelColor<156)
                {
                      br13=br13+1;
                }
```

```
                else if(pixelColor>=157&&pixelColor<168)
                {
                      br14=br14+1;
                }
                else if(pixelColor>=169&&pixelColor<180)
                {
                      br15=br15+1;
                }
                else if(pixelColor>=181&&pixelColor<192)
                {
                      br16=br16+1;
                }
                else if(pixelColor>=193&&pixelColor<204)
                {
                      br17=br17+1;
                }
                else if(pixelColor>=205&&pixelColor<216)
                {
                      br18=br18+1;
                }
                else if(pixelColor>=217&&pixelColor<228)
                {
                      br19=br19+1;
                }
                else
                {
                      br20=br20+1;
                }


//green*******************************
               pixelColor = img.GetPixel(j,i).G;

               if(pixelColor>=0&&pixelColor<12)
                {
                      bg1=bg1+1;
                }
                else if(pixelColor>=13&&pixelColor<24)
                {
                      bg2=bg2+1;
                }
                else if(pixelColor>=25&&pixelColor<36)
                {
                      bg3=bg3+1;
                }
                else if(pixelColor>=37&&pixelColor<48)
                {
                      bg4=bg4+1;
                }
                else if(pixelColor>=49&&pixelColor<60)
                {
                      bg5=bg5+1;
                }
                else if(pixelColor>=61&&pixelColor<72)
                {
                      bg6=bg6+1;
                }
```

```
else if(pixelColor>=73&&pixelColor<84)
{
      bg7=bg7+1;
}
else if(pixelColor>=85&&pixelColor<96)
{
      bg8=bg8+1;
}
else if(pixelColor>=97&&pixelColor<108)
{
      bg9=bg9+1;
}
else if(pixelColor>=109&&pixelColor<120)
{
      bg10=bg10+1;
}
else if(pixelColor>=121&&pixelColor<132)
{
      bg11=bg11+1;
}
else if(pixelColor>=133&&pixelColor<144)
{
      bg12=bg12+1;
}
else if(pixelColor>=145&&pixelColor<156)
{
      bg13=bg13+1;
}
else if(pixelColor>=157&&pixelColor<168)
{
      bg14=bg14+1;
}
else if(pixelColor>=169&&pixelColor<180)
{
      bg15=bg15+1;
}
else if(pixelColor>=181&&pixelColor<192)
{
      bg16=bg16+1;
}
else if(pixelColor>=193&&pixelColor<204)
{
      bg17=bg17+1;
}
else if(pixelColor>=205&&pixelColor<216)
{
      bg18=bg18+1;
}
else if(pixelColor>=217&&pixelColor<228)
{
      bg19=bg19+1;
}
else
{
      bg20=bg20+1;
}
```

```
//blue**********************************
                    pixelColor = img.GetPixel(j,i).B;

                    if(pixelColor>=0&&pixelColor<12)
                    {
                          bb1=bb1+1;
                    }
                    else if(pixelColor>=13&&pixelColor<24)
                    {
                          bb2=bb2+1;
                    }
                    else if(pixelColor>=25&&pixelColor<36)
                    {
                          bb3=bb3+1;
                    }
                    else if(pixelColor>=37&&pixelColor<48)
                    {
                          bb4=bb4+1;
                    }
                    else if(pixelColor>=49&&pixelColor<60)
                    {
                          bb5=bb5+1;
                    }
                    else if(pixelColor>=61&&pixelColor<72)
                    {
                          bb6=bb6+1;
                    }
                    else if(pixelColor>=73&&pixelColor<84)
                    {
                          bb7=bb7+1;
                    }
                    else if(pixelColor>=85&&pixelColor<96)
                    {
                          bb8=bb8+1;
                    }
                    else if(pixelColor>=97&&pixelColor<108)
                    {
                          bb9=bb9+1;
                    }
                    else if(pixelColor>=109&&pixelColor<120)
                    {
                          bb10=bb10+1;
                    }
                    else if(pixelColor>=121&&pixelColor<132)
                    {
                          bb11=bb11+1;
                    }
                    else if(pixelColor>=133&&pixelColor<144)
                    {
                          bb12=bb12+1;
                    }
                    else if(pixelColor>=145&&pixelColor<156)
                    {
                          bb13=bb13+1;
                    }
                    else if(pixelColor>=157&&pixelColor<168)
                    {
                          bb14=bb14+1;
                    }
```

```
                    else if(pixelColor>=169&&pixelColor<180)
                    {
                          bb15=bb15+1;
                    }
                    else if(pixelColor>=181&&pixelColor<192)
                    {
                          bb16=bb16+1;
                    }
                    else if(pixelColor>=193&&pixelColor<204)
                    {
                          bb17=bb17+1;
                    }
                    else if(pixelColor>=205&&pixelColor<216)
                    {
                          bb18=bb18+1;
                    }
                    else if(pixelColor>=217&&pixelColor<228)
                    {
                          bb19=bb19+1;
                    }
                    else
                    {
                          bb20=bb20+1;
                    }

                    }
              }
                    fn = "frame"+e.ToString();
```

ก-4 การนำค่า Vector ที่หาได้ไปใส่ในดาต้าเบส

```
// นำค่า vector ที่หาได้ไปใส่ในดาต้าเบส
      if (conn != null)
            conn.Close();

            string connStr = String.Format("server=localhost;user
id=root; password= ; database=testavi; pooling=false" );

            try
            {
                  conn = new MySqlConnection( connStr );
                  conn.Open();


            }
            catch (MySqlException ex)
            {
            MessageBox.Show( "Error connecting to the server: " +
ex.Message );
            }
```

```
            data = new DataTable();
            cb = new MySqlCommand();
            cb.CommandText=   "insert into avi
(path,framenumber,v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,
v16,v17,v18,v19,v20,v21,v22,v23,v24,v25,v26,v27,v28,v29,v30,v31,v32,v
33,v34,v35,v36,v37,v38,v39,v40,v41,v42,v43,v44,v45,v46,v47,v48,v49,v5
0,v51,v52,v53,v54,v55,v56,v57,v58,v59,v60) values
(?pa,?fn,?br1,?br2,?br3,?br4,?br5,?br6,?br7,?br8,?br9,?br10,?br11,?br
12,?br13,?br14,?br15,?br16,?br17,?br18,?br19,?br20,?bg1,?bg2,?bg3,?bg
4,?bg5,?bg6,?bg7,?bg8,?bg9,?bg10,?bg10,?bg12,?bg13,?bg14,?bg15,?bg16,
?bg17,?bg18,?bg19,?bg20,?bb1,?bb2,?bb3,?bb4,?bb5,?bb6,?bb7,?bb8,?bb9,
?bb10,?bb11,?bb12,?bb13,?bb14,?bb15,?bb16,?bb17,?bb18,?bb19,?bb20)";

                        cb.Connection = conn;
                        cb.Parameters.Add("?pa", a);
                        cb.Parameters.Add("?fn", fn);
                        cb.Parameters.Add("?br1", br1);
                        cb.Parameters.Add("?br2", br2);
                        cb.Parameters.Add("?br3", br3);
                        cb.Parameters.Add("?br4", br4);
                        cb.Parameters.Add("?br5", br5);
                        cb.Parameters.Add("?br6", br6);
                        cb.Parameters.Add("?br7", br7);
                        cb.Parameters.Add("?br8", br8);
                        cb.Parameters.Add("?br9", br9);
                        cb.Parameters.Add("?br10", br10);
                        cb.Parameters.Add("?br11", br11);
                        cb.Parameters.Add("?br12", br12);
                        cb.Parameters.Add("?br13", br13);
                        cb.Parameters.Add("?br14", br14);
                        cb.Parameters.Add("?br15", br15);
                        cb.Parameters.Add("?br16", br16);
                        cb.Parameters.Add("?br17", br17);
                        cb.Parameters.Add("?br18", br18);
                        cb.Parameters.Add("?br19", br19);
                        cb.Parameters.Add("?br20", br20);

                        cb.Parameters.Add("?bg1", bg1);
                        cb.Parameters.Add("?bg2", bg2);
                        cb.Parameters.Add("?bg3", bg3);
                        cb.Parameters.Add("?bg4", bg4);
                        cb.Parameters.Add("?bg5", bg5);
                        cb.Parameters.Add("?bg6", bg6);
                        cb.Parameters.Add("?bg7", bg7);
                        cb.Parameters.Add("?bg8", bg8);
                        cb.Parameters.Add("?bg9", bg9);
                        cb.Parameters.Add("?bg10", bg10);
                        cb.Parameters.Add("?bg11", bg11);
                        cb.Parameters.Add("?bg12", bg12);
                        cb.Parameters.Add("?bg13", bg13);
                        cb.Parameters.Add("?bg14", bg14);
                        cb.Parameters.Add("?bg15", bg15);
                        cb.Parameters.Add("?bg16", bg16);
                        cb.Parameters.Add("?bg17", bg17);
                        cb.Parameters.Add("?bg18", bg18);
                        cb.Parameters.Add("?bg19", bg19);
                        cb.Parameters.Add("?bg20", bg20);

                        cb.Parameters.Add("?bb1", bb1);
                        cb.Parameters.Add("?bb2", bb2);
                        cb.Parameters.Add("?bb3", bb3);
```

```
                cb.Parameters.Add("?bb4", bb4);
                cb.Parameters.Add("?bb5", bb5);
                cb.Parameters.Add("?bb6", bb6);
                cb.Parameters.Add("?bb7", bb7);
                cb.Parameters.Add("?bb8", bb8);
                cb.Parameters.Add("?bb9", bb9);
                cb.Parameters.Add("?bb10", bb10);
                cb.Parameters.Add("?bb11", bb11);
                cb.Parameters.Add("?bb12", bb12);
                cb.Parameters.Add("?bb13", bb13);
                cb.Parameters.Add("?bb14", bb14);
                cb.Parameters.Add("?bb15", bb15);
                cb.Parameters.Add("?bb16", bb16);
                cb.Parameters.Add("?bb17", bb17);
                cb.Parameters.Add("?bb18", bb18);
                cb.Parameters.Add("?bb19", bb19);
                cb.Parameters.Add("?bb20", bb20);
                cb.ExecuteNonQuery();

                if(e==0)
                { d=c1; }
                else if(e==1)
                { d=c-1; }

            }

             stream.GetFrameClose();
            aviManager.Close();
            MessageBox.Show("Complete Extract");
    }
```

Source code การจัดการโปรแกรมค้นหาวีดีโอ

ก-5 ในการค้นหาแบบไฟล์รูปภาพ จะทำการรับไฟล์รูปภาพมา แล้วทำเป็นค่า Vector ของ RGB ทั้งหมด 60 bin ส่วนในกรณีของ การค้นหาแบบไฟล์วีดีโอนั้นจะใช้หลักการทำ Vector เหมือนกันเพียงแต่มีการเพิ่มค่าตัวแปร และ จำนวนคีย์เฟรมที่ใช้ค้นหาในแต่ละประเภท

```
private void button10_Click(object sender, System.EventArgs e)
{ CleanUp();
        // search vm Pic
        if(radioButton17.Checked )
        {

        String fileName1 = GetFileName("Images (*.bmp; *.jpg;
*.tif; *.png; *.gif)|*.bmp;*.jpg;*.tif;*.png;*.gif");
        if (fileName1 != null)
        {
              this.pictureBox1.Image = new Bitmap(fileName1);

        }

        string a = fileName1;
        int i,j,b=1;
        double
b1=0,b2=0,b3=0,b4=0,b5=0,b6=0,b7=0,b8=0,b9=0,b10=0,b11=0,b12=0,b
13=0,b14=0,b15=0,b16=0,b17=0,b18=0,b19=0,b20=0;
        double
b21=0,b22=0,b23=0,b24=0,b25=0,b26=0,b27=0,b28=0,b29=0,b30=0,b31=
0,b32=0,b33=0,b34=0,b35=0,b36=0,b37=0,b38=0,b39=0,b40=0;
double
b41=0,b42=0,b43=0,b44=0,b45=0,b46=0,b47=0,b48=0,b49=0,b50=0,b51=
0,b52=0,b53=0,b54=0,b55=0,b56=0,b57=0,b58=0,b59=0,b60=0;

        int A=352;  //Width
        int B=288;  //Height

        byte pixelColor;

        string path = (a);
        Bitmap img = new Bitmap(path);

//pic1
        for (i=0; i<B; i++)
        {
        for (j=0; j<A; j++)


{       //red*****************************
        pixelColor = img.GetPixel(j,i).R;

if(pixelColor>=0&&pixelColor<12)
        {
              b1=b1+1;
        }
```

```
else if(pixelColor>=13&&pixelColor<24)
{
      b2=b2+1;
}
else if(pixelColor>=25&&pixelColor<36)
{
      b3=b3+1;
}
else if(pixelColor>=37&&pixelColor<48)
{
      b4=b4+1;
}
else if(pixelColor>=49&&pixelColor<60)
{
      b5=b5+1;
}
else if(pixelColor>=61&&pixelColor<72)
{
      b6=b6+1;
}
else if(pixelColor>=73&&pixelColor<84)
{
      b7=b7+1;
}
else if(pixelColor>=85&&pixelColor<96)
{
      b8=b8+1;
}
else if(pixelColor>=97&&pixelColor<108)
{
      b9=b9+1;
}
else if(pixelColor>=109&&pixelColor<120)
{
      b10=b10+1;
}
else if(pixelColor>=121&&pixelColor<132)
{
      b11=b11+1;
}
else if(pixelColor>=133&&pixelColor<144)
{
      b12=b12+1;
}
else if(pixelColor>=145&&pixelColor<156)
{
      b13=b13+1;
}
else if(pixelColor>=157&&pixelColor<168)
{
      b14=b14+1;
}
else if(pixelColor>=169&&pixelColor<180)
{
      b15=b15+1;
}
else if(pixelColor>=181&&pixelColor<192)
{
      b16=b16+1;
}
```

```
        else if(pixelColor>=193&&pixelColor<204)
        {
              b17=b17+1;
        }
        else if(pixelColor>=205&&pixelColor<216)
        {
              b18=b18+1;
        }
        else if(pixelColor>=217&&pixelColor<228)
        {
              b19=b19+1;
        }
        else
        {
              b20=b20+1;
        }


//green**********************************
        pixelColor = img.GetPixel(j,i).G;
        if(pixelColor>=0&&pixelColor<12)
        {
              b21=b21+1;
        }
        else if(pixelColor>=13&&pixelColor<24)
        {
              b22=b22+1;
        }
        else if(pixelColor>=25&&pixelColor<36)
        {
              b23=b23+1;
        }
        else if(pixelColor>=37&&pixelColor<48)
        {
              b24=b24+1;
        }
        else if(pixelColor>=49&&pixelColor<60)
        {
              b25=b25+1;
        }
        else if(pixelColor>=61&&pixelColor<72)
        {
              b26=b26+1;
        }
        else if(pixelColor>=73&&pixelColor<84)
        {
              b27=b27+1;
        }
        else if(pixelColor>=85&&pixelColor<96)
        {
              b28=b28+1;
        }
        else if(pixelColor>=97&&pixelColor<108)
        {
              b29=b29+1;
        }
        else if(pixelColor>=109&&pixelColor<120)
        {
              b30=b30+1;
        }
```

```
        else if(pixelColor>=121&&pixelColor<132)
        {
              b31=b31+1;
        }
        else if(pixelColor>=133&&pixelColor<144)
        {
              b32=b32+1;
        }
        else if(pixelColor>=145&&pixelColor<156)
        {
              b33=b33+1;
        }
        else if(pixelColor>=157&&pixelColor<168)
        {
              b34=b34+1;
        }
        else if(pixelColor>=169&&pixelColor<180)
        {
              b35=b35+1;
        }
        else if(pixelColor>=181&&pixelColor<192)
        {
              b36=b36+1;
        }
        else if(pixelColor>=193&&pixelColor<204)
        {
              b37=b37+1;
        }
        else if(pixelColor>=205&&pixelColor<216)
        {
              b38=b38+1;
        }
        else if(pixelColor>=217&&pixelColor<228)
        {
              b39=b39+1;
        }
        else
        {
              b40=b40+1;
        }

//blue*********************************
        pixelColor = img.GetPixel(j,i).B;
        if(pixelColor>=0&&pixelColor<12)
        {
              b41=b41+1;
        }
        else if(pixelColor>=13&&pixelColor<24)
        {
              b42=b42+1;
        }
        else if(pixelColor>=25&&pixelColor<36)
        {
              b43=b43+1;
        }
        else if(pixelColor>=37&&pixelColor<48)
        {
              b44=b44+1;
        }
```

```
else if(pixelColor>=49&&pixelColor<60)
{
      b45=b45+1;
}
else if(pixelColor>=61&&pixelColor<72)

{
b46=b46+1;
}
else if(pixelColor>=73&&pixelColor<84)
{
      b47=b47+1;
}
else if(pixelColor>=85&&pixelColor<96)
{
      b48=b48+1;
}
else if(pixelColor>=97&&pixelColor<108)
{
      b49=b49+1;
}
else if(pixelColor>=109&&pixelColor<120)
{
      b50=b50+1;
}
else if(pixelColor>=121&&pixelColor<132)
{
      b51=b51+1;
}
else if(pixelColor>=133&&pixelColor<144)
{
      b52=b52+1;
}
else if(pixelColor>=145&&pixelColor<156)
{
      b53=b53+1;
}
else if(pixelColor>=157&&pixelColor<168)
{
      b54=b54+1;
}
else if(pixelColor>=169&&pixelColor<180)
{
      b55=b55+1;
}
else if(pixelColor>=181&&pixelColor<192)
{
      b56=b56+1;
}
else if(pixelColor>=193&&pixelColor<204)
{
      b57=b57+1;
}
else if(pixelColor>=205&&pixelColor<216)
{
      b58=b58+1;
}
else if(pixelColor>=217&&pixelColor<228)
{
      b59=b59+1;
```

```
            }

        else
        {
            b60=b60+1;
        }
```

ก-6 ทำการเปรียบเทียบค่าเวคเตอร์ของวีดีโอในฐานข้อมูลกับไฟล์ภาพ

```
//      เปรียบเทียบค่า กับใน ดาต้าเบส
cb = new MySqlCommand();
cb.CommandText=   "select path from avi where framenumber =
'frame1' order by ( abs(?b1-v1)+abs(?b2-v2)+abs(?b3-
v3)+abs(?b4-v4)+abs(?b5-v5)+abs(?b6-v6)+abs(?b7-v7)+abs(?b8-
v8)+abs(?b9-v9)+abs(?b10-v10)+abs(?b11-v11)+abs(?b12-
v12)+abs(?b13-v13)+abs(?b14-v14)+abs(?b15-v15)+abs(?b16-
v16)+abs(?b17-v17)+abs(?b18-v18)+abs(?b19-v19)+abs(?b20-
v20)+abs(?b21-v21)+abs(?b22-v22)+abs(?b23-v23)+abs(?b24-
v24)+abs(?b25-v25)+abs(?b26-v26)+abs(?b27-v27)+abs(?b28-
v28)+abs(?b29-v29)+abs(?b30-v30)+abs(?b31-v31)+abs(?b32-
v32)+abs(?b33-v33)+abs(?b34-v34)+abs(?b35-v35)+abs(?b36-
v36)+abs(?b37-v37)+abs(?b38-v38)+abs(?b39-v39)+abs(?b40-
v40)+abs(?b41-v41)+abs(?b42-v42)+abs(?b43-v43)+abs(?b44-
v44)+abs(?b45-v45)+abs(?b46-v46)+abs(?b47-v47)+abs(?b48-
v48)+abs(?b49-v49)+abs(?b50-v50)+abs(?b51-v51)+abs(?b52-
v52)+abs(?b53-v53)+abs(?b54-v54)+abs(?b55-v55)+abs(?b56-
v56)+abs(?b57-v57)+abs(?b58-v58)+abs(?b59-v59)+abs(?b60-v60) )
limit 16 ";

cb.Connection = conn;
cb.Parameters.Add("?b1", b1);
cb.Parameters.Add("?b2", b2);
cb.Parameters.Add("?b3", b3);
cb.Parameters.Add("?b4", b4);
cb.Parameters.Add("?b5", b5);
cb.Parameters.Add("?b6", b6);
cb.Parameters.Add("?b7", b7);
cb.Parameters.Add("?b8", b8);
cb.Parameters.Add("?b9", b9);
cb.Parameters.Add("?b10", b10);
cb.Parameters.Add("?b11", b11);
cb.Parameters.Add("?b12", b12);
cb.Parameters.Add("?b13", b13);
cb.Parameters.Add("?b14", b14);
cb.Parameters.Add("?b15", b15);
cb.Parameters.Add("?b16", b16);
cb.Parameters.Add("?b17", b17);
cb.Parameters.Add("?b18", b18);
cb.Parameters.Add("?b19", b19);
cb.Parameters.Add("?b20", b20);
cb.Parameters.Add("?b21", b21);
cb.Parameters.Add("?b22", b22);
cb.Parameters.Add("?b23", b23);
cb.Parameters.Add("?b24", b24);
cb.Parameters.Add("?b25", b25);
cb.Parameters.Add("?b26", b26);
```

```
cb.Parameters.Add("?b27", b27);
cb.Parameters.Add("?b28", b28);
cb.Parameters.Add("?b29", b29);

cb.Parameters.Add("?b30", b30);
cb.Parameters.Add("?b31", b31);
cb.Parameters.Add("?b32", b32);
cb.Parameters.Add("?b33", b33);
cb.Parameters.Add("?b34", b34);
cb.Parameters.Add("?b35", b35);
cb.Parameters.Add("?b36", b36);
cb.Parameters.Add("?b37", b37);
cb.Parameters.Add("?b38", b38);

cb.Parameters.Add("?b39", b39);
cb.Parameters.Add("?b40", b40);
cb.Parameters.Add("?b41", b41);
cb.Parameters.Add("?b42", b42);
cb.Parameters.Add("?b43", b43);
cb.Parameters.Add("?b44", b44);
cb.Parameters.Add("?b45", b45);
cb.Parameters.Add("?b46", b46);
cb.Parameters.Add("?b47", b47);
cb.Parameters.Add("?b48", b48);
cb.Parameters.Add("?b49", b49);
cb.Parameters.Add("?b50", b50);
cb.Parameters.Add("?b51", b51);
cb.Parameters.Add("?b52", b52);
cb.Parameters.Add("?b53", b53);
cb.Parameters.Add("?b54", b54);
cb.Parameters.Add("?b55", b55);
cb.Parameters.Add("?b56", b56);
cb.Parameters.Add("?b57", b57);
cb.Parameters.Add("?b58", b58);
cb.Parameters.Add("?b59", b59);
cb.Parameters.Add("?b60", b60);

MySqlDataReader myReader;
myReader = cb.ExecuteReader();
```

ก-7 การเปรียบเทียบค่าเวคเตอร์ของวีดีโอในฐานข้อมูลกับไฟล์วีดีโอที่นำมาค้นหา แบบ 1 Key frame

```
cb = new MySqlCommand();
cb.CommandText=   "select path from avi where framenumber =
'frame1' order by( abs(?b1-v1)+abs(?b2-v2)+abs(?b3-v3)+abs(?b4-
v4)+abs(?b5-v5)+abs(?b6-v6)+abs(?b7-v7)+abs(?b8-v8)+abs(?b9-
v9)+abs(?b10-v10)+abs(?b11-v11)+abs(?b12-v12)+abs(?b13-
v13)+abs(?b14-v14)+abs(?b15-v15)+abs(?b16-v16)+abs(?b17-
v17)+abs(?b18-v18)+abs(?b19-v19)+abs(?b20-v20)+abs(?b21-
v21)+abs(?b22-v22)+abs(?b23-v23)+abs(?b24-v24)+abs(?b25-
v25)+abs(?b26-v26)+abs(?b27-v27)+abs(?b28-v28)+abs(?b29-
v29)+abs(?b30-v30)+abs(?b31-v31)+abs(?b32-v32)+abs(?b33-
v33)+abs(?b34-v34)+abs(?b35-v35)+abs(?b36-v36)+abs(?b37-
v37)+abs(?b38-v38)+abs(?b39-v39)+abs(?b40-v40)+abs(?b41-
```

```
v41)+abs(?b42-v42)+abs(?b43-v43)+abs(?b44-v44)+abs(?b45-
v45)+abs(?b46-v46)+abs(?b47-v47)+abs(?b48-v48)+abs(?b49-
v49)+abs(?b50-v50)+abs(?b51-v51)+abs(?b52-v52)+abs(?b53-
v53)+abs(?b54-v54)+abs(?b55-v55)+abs(?b56-v56)+abs(?b57-
v57)+abs(?b58-v58)+abs(?b59-v59)+abs(?b60-v60) )  limit 16 ";

cb.Connection = conn;
cb.Parameters.Add("?b1", b1);
cb.Parameters.Add("?b2", b2);
cb.Parameters.Add("?b3", b3);
cb.Parameters.Add("?b4", b4);
cb.Parameters.Add("?b5", b5);
cb.Parameters.Add("?b6", b6);
cb.Parameters.Add("?b7", b7);
cb.Parameters.Add("?b8", b8);
cb.Parameters.Add("?b9", b9);
cb.Parameters.Add("?b10", b10);
cb.Parameters.Add("?b11", b11);
cb.Parameters.Add("?b12", b12);
cb.Parameters.Add("?b13", b13);
cb.Parameters.Add("?b14", b14);
cb.Parameters.Add("?b15", b15);
cb.Parameters.Add("?b16", b16);
cb.Parameters.Add("?b17", b17);
cb.Parameters.Add("?b18", b18);
cb.Parameters.Add("?b19", b19);
cb.Parameters.Add("?b20", b20);
cb.Parameters.Add("?b21", b21);
cb.Parameters.Add("?b22", b22);
cb.Parameters.Add("?b23", b23);
cb.Parameters.Add("?b24", b24);
cb.Parameters.Add("?b25", b25);
cb.Parameters.Add("?b26", b26);
cb.Parameters.Add("?b27", b27);
cb.Parameters.Add("?b28", b28);
cb.Parameters.Add("?b29", b29);
cb.Parameters.Add("?b30", b30);
cb.Parameters.Add("?b31", b31);
cb.Parameters.Add("?b32", b32);
cb.Parameters.Add("?b33", b33);
cb.Parameters.Add("?b34", b34);
cb.Parameters.Add("?b35", b35);
cb.Parameters.Add("?b36", b36);
cb.Parameters.Add("?b37", b37);
cb.Parameters.Add("?b38", b38);
cb.Parameters.Add("?b39", b39);
cb.Parameters.Add("?b40", b40);
cb.Parameters.Add("?b41", b41);
cb.Parameters.Add("?b42", b42);
cb.Parameters.Add("?b43", b43);
cb.Parameters.Add("?b44", b44);
cb.Parameters.Add("?b45", b45);
cb.Parameters.Add("?b46", b46);
cb.Parameters.Add("?b47", b47);
cb.Parameters.Add("?b48", b48);
cb.Parameters.Add("?b49", b49);
cb.Parameters.Add("?b50", b50);
cb.Parameters.Add("?b51", b51);
cb.Parameters.Add("?b52", b52);
cb.Parameters.Add("?b53", b53);
cb.Parameters.Add("?b54", b54);
```

```
cb.Parameters.Add("?b55", b55);
cb.Parameters.Add("?b56", b56);
cb.Parameters.Add("?b57", b57);
cb.Parameters.Add("?b58", b58);
cb.Parameters.Add("?b59", b59);
cb.Parameters.Add("?b60", b60);


MySqlDataReader myReader;
myReader = cb.ExecuteReader();
```

ก-8 การเปรียบเทียบค่าเวคเตอร์ของวีดีโอในฐานข้อมูลกับไฟล์วีดีโอที่นำมาค้นหา แบบ 3 Key frame

```
//frame0
cb = new MySqlCommand();
cb.CommandText=   "update avi set dist = ( abs(?b1-v1)+abs(?b2-
v2)+abs(?b3-v3)+abs(?b4-v4)+abs(?b5-v5)+abs(?b6-v6)+abs(?b7-
v7)+abs(?b8-v8)+abs(?b9-v9)+abs(?b10-v10)+abs(?b11-
v11)+abs(?b12-v12)+abs(?b13-v13)+abs(?b14-v14)+abs(?b15-
v15)+abs(?b16-v16)+abs(?b17-v17)+abs(?b18-v18)+abs(?b19-
v19)+abs(?b20-v20)+abs(?b21-v21)+abs(?b22-v22)+abs(?b23-
v23)+abs(?b24-v24)+abs(?b25-v25)+abs(?b26-v26)+abs(?b27-
v27)+abs(?b28-v28)+abs(?b29-v29)+abs(?b30-v30)+abs(?b31-
v31)+abs(?b32-v32)+abs(?b33-v33)+abs(?b34-v34)+abs(?b35-
v35)+abs(?b36-v36)+abs(?b37-v37)+abs(?b38-v38)+abs(?b39-
v39)+abs(?b40-v40)+abs(?b41-v41)+abs(?b42-v42)+abs(?b43-
v43)+abs(?b44-v44)+abs(?b45-v45)+abs(?b46-v46)+abs(?b47-
v47)+abs(?b48-v48)+abs(?b49-v49)+abs(?b50-v50)+abs(?b51-
v51)+abs(?b52-v52)+abs(?b53-v53)+abs(?b54-v54)+abs(?b55-
v55)+abs(?b56-v56)+abs(?b57-v57)+abs(?b58-v58)+abs(?b59-
v59)+abs(?b60-v60) ) where framenumber = 'frame0' ";
cb.Connection = conn;

cb.Parameters.Add("?b1", b1);
cb.Parameters.Add("?b2", b2);
cb.Parameters.Add("?b3", b3);
cb.Parameters.Add("?b4", b4);
cb.Parameters.Add("?b5", b5);
cb.Parameters.Add("?b6", b6);
cb.Parameters.Add("?b7", b7);
cb.Parameters.Add("?b8", b8);
cb.Parameters.Add("?b9", b9);
cb.Parameters.Add("?b10", b10);
cb.Parameters.Add("?b11", b11);
cb.Parameters.Add("?b12", b12);
cb.Parameters.Add("?b13", b13);
cb.Parameters.Add("?b14", b14);
cb.Parameters.Add("?b15", b15);
cb.Parameters.Add("?b16", b16);
cb.Parameters.Add("?b17", b17);
cb.Parameters.Add("?b18", b18);
cb.Parameters.Add("?b19", b19);
cb.Parameters.Add("?b20", b20);
```

```
cb.Parameters.Add("?b21", b21);
cb.Parameters.Add("?b22", b22);
cb.Parameters.Add("?b23", b23);
cb.Parameters.Add("?b24", b24);
cb.Parameters.Add("?b25", b25);
cb.Parameters.Add("?b26", b26);
cb.Parameters.Add("?b27", b27);
cb.Parameters.Add("?b28", b28);
cb.Parameters.Add("?b29", b29);
cb.Parameters.Add("?b30", b30);
cb.Parameters.Add("?b31", b31);
cb.Parameters.Add("?b32", b32);
cb.Parameters.Add("?b33", b33);
cb.Parameters.Add("?b34", b34);
cb.Parameters.Add("?b35", b35);
cb.Parameters.Add("?b36", b36);
cb.Parameters.Add("?b37", b37);
cb.Parameters.Add("?b38", b38);
cb.Parameters.Add("?b39", b39);
cb.Parameters.Add("?b40", b40);
cb.Parameters.Add("?b41", b41);
cb.Parameters.Add("?b42", b42);
cb.Parameters.Add("?b43", b43);
cb.Parameters.Add("?b44", b44);
cb.Parameters.Add("?b45", b45);
cb.Parameters.Add("?b46", b46);
cb.Parameters.Add("?b47", b47);
cb.Parameters.Add("?b48", b48);
cb.Parameters.Add("?b49", b49);
cb.Parameters.Add("?b50", b50);
cb.Parameters.Add("?b51", b51);
cb.Parameters.Add("?b52", b52);
cb.Parameters.Add("?b53", b53);
cb.Parameters.Add("?b54", b54);
cb.Parameters.Add("?b55", b55);
cb.Parameters.Add("?b56", b56);
cb.Parameters.Add("?b57", b57);
cb.Parameters.Add("?b58", b58);
cb.Parameters.Add("?b59", b59);
cb.Parameters.Add("?b60", b60);
cb.ExecuteNonQuery();

//frame1
cb = new MySqlCommand();
cb.CommandText=   "update avi set dist = ( abs(?bb1-
v1)+abs(?bb2-v2)+abs(?bb3-v3)+abs(?bb4-v4)+abs(?bb5-
v5)+abs(?bb6-v6)+abs(?bb7-v7)+abs(?bb8-v8)+abs(?bb9-
v9)+abs(?bb10-v10)+abs(?bb11-v11)+abs(?bb12-v12)+abs(?bb13-
v13)+abs(?bb14-v14)+abs(?bb15-v15)+abs(?bb16-v16)+abs(?bb17-
v17)+abs(?bb18-v18)+abs(?bb19-v19)+abs(?bb20-v20)+abs(?bb21-
v21)+abs(?bb22-v22)+abs(?bb23-v23)+abs(?bb24-v24)+abs(?bb25-
v25)+abs(?bb26-v26)+abs(?bb27-v27)+abs(?bb28-v28)+abs(?bb29-
v29)+abs(?bb30-v30)+abs(?bb31-v31)+abs(?bb32-v32)+abs(?bb33-
v33)+abs(?bb34-v34)+abs(?bb35-v35)+abs(?bb36-v36)+abs(?bb37-
v37)+abs(?bb38-v38)+abs(?bb39-v39)+abs(?bb40-v40)+abs(?bb41-
v41)+abs(?bb42-v42)+abs(?bb43-v43)+abs(?bb44-v44)+abs(?bb45-
v45)+abs(?bb46-v46)+abs(?bb47-v47)+abs(?bb48-v48)+abs(?bb49-
v49)+abs(?bb50-v50)+abs(?bb51-v51)+abs(?bb52-v52)+abs(?bb53-
v53)+abs(?bb54-v54)+abs(?bb55-v55)+abs(?bb56-v56)+abs(?bb57-
v57)+abs(?bb58-v58)+abs(?bb59-v59)+abs(?bb60-v60) ) where
framenumber = 'frame1' ";
```

```
cb.Connection = conn;

cb.Parameters.Add("?bb1", bb1);
cb.Parameters.Add("?bb2", bb2);
cb.Parameters.Add("?bb3", bb3);
cb.Parameters.Add("?bb4", bb4);
cb.Parameters.Add("?bb5", bb5);
cb.Parameters.Add("?bb6", bb6);
cb.Parameters.Add("?bb7", bb7);
cb.Parameters.Add("?bb8", bb8);
cb.Parameters.Add("?bb9", bb9);
cb.Parameters.Add("?bb10", bb10);
cb.Parameters.Add("?bb11", bb11);
cb.Parameters.Add("?bb12", bb12);
cb.Parameters.Add("?bb13", bb13);
cb.Parameters.Add("?bb14", bb14);
cb.Parameters.Add("?bb15", bb15);
cb.Parameters.Add("?bb16", bb16);
cb.Parameters.Add("?bb17", bb17);
cb.Parameters.Add("?bb18", bb18);
cb.Parameters.Add("?bb19", bb19);
cb.Parameters.Add("?bb20", bb20);
cb.Parameters.Add("?bb21", bb21);
cb.Parameters.Add("?bb22", bb22);
cb.Parameters.Add("?bb23", bb23);
cb.Parameters.Add("?bb24", bb24);
cb.Parameters.Add("?bb25", bb25);
cb.Parameters.Add("?bb26", bb26);
cb.Parameters.Add("?bb27", bb27);
cb.Parameters.Add("?bb28", bb28);
cb.Parameters.Add("?bb29", bb29);
cb.Parameters.Add("?bb30", bb30);
cb.Parameters.Add("?bb31", bb31);
cb.Parameters.Add("?bb32", bb32);
cb.Parameters.Add("?bb33", bb33);
cb.Parameters.Add("?bb34", bb34);
cb.Parameters.Add("?bb35", bb35);
cb.Parameters.Add("?bb36", bb36);
cb.Parameters.Add("?bb37", bb37);
cb.Parameters.Add("?bb38", bb38);
cb.Parameters.Add("?bb39", bb39);
cb.Parameters.Add("?bb40", bb40);
cb.Parameters.Add("?bb41", bb41);
cb.Parameters.Add("?bb42", bb42);
cb.Parameters.Add("?bb43", bb43);
cb.Parameters.Add("?bb44", bb44);
cb.Parameters.Add("?bb45", bb45);
cb.Parameters.Add("?bb46", bb46);
cb.Parameters.Add("?bb47", bb47);
cb.Parameters.Add("?bb48", bb48);
cb.Parameters.Add("?bb49", bb49);
cb.Parameters.Add("?bb50", bb50);
cb.Parameters.Add("?bb51", bb51);
cb.Parameters.Add("?bb52", bb52);
cb.Parameters.Add("?bb53", bb53);
cb.Parameters.Add("?bb54", bb54);
cb.Parameters.Add("?bb55", bb55);
cb.Parameters.Add("?bb56", bb56);
cb.Parameters.Add("?bb57", bb57);
cb.Parameters.Add("?bb58", bb58);
cb.Parameters.Add("?bb59", bb59);
```

```
cb.Parameters.Add("?bb60", bb60);
cb.ExecuteNonQuery();

//frame2
cb = new MySqlCommand();
cb.CommandText=    "update avi set dist =( abs(?bbb1-
v1)+abs(?bbb2-v2)+abs(?bbb3-v3)+abs(?bbb4-v4)+abs(?bbb5-
v5)+abs(?bbb6-v6)+abs(?bbb7-v7)+abs(?bbb8-v8)+abs(?bbb9-
v9)+abs(?bbb10-v10)+abs(?bbb11-v11)+abs(?bbb12-v12)+abs(?bbb13-
v13)+abs(?bbb14-v14)+abs(?bbb15-v15)+abs(?bbb16-
v16)+abs(?bbb17-v17)+abs(?bbb18-v18)+abs(?bbb19-
v19)+abs(?bbb20-v20)+abs(?bbb21-v21)+abs(?bbb22-
v22)+abs(?bbb23-v23)+abs(?bbb24-v24)+abs(?bbb25-
v25)+abs(?bbb26-v26)+abs(?bbb27-v27)+abs(?bbb28-
v28)+abs(?bbb29-v29)+abs(?bbb30-v30)+abs(?bbb31-
v31)+abs(?bbb32-v32)+abs(?bbb33-v33)+abs(?bbb34-
v34)+abs(?bbb35-v35)+abs(?bbb36-v36)+abs(?bbb37-
v37)+abs(?bbb38-v38)+abs(?bbb39-v39)+abs(?bbb40-
v40)+abs(?bbb41-v41)+abs(?bbb42-v42)+abs(?bbb43-
v43)+abs(?bbb44-v44)+abs(?bbb45-v45)+abs(?bbb46-
v46)+abs(?bbb47-v47)+abs(?bbb48-v48)+abs(?bbb49-
v49)+abs(?bbb50-v50)+abs(?bbb51-v51)+abs(?bbb52-
v52)+abs(?bbb53-v53)+abs(?bbb54-v54)+abs(?bbb55-
v55)+abs(?bbb56-v56)+abs(?bbb57-v57)+abs(?bbb58-
v58)+abs(?bbb59-v59)+abs(?bbb60-v60) ) where framenumber =
'frame2' ";
cb.Connection = conn;

cb.Parameters.Add("?bbb1", bbb1);
cb.Parameters.Add("?bbb2", bbb2);
cb.Parameters.Add("?bbb3", bbb3);
cb.Parameters.Add("?bbb4", bbb4);
cb.Parameters.Add("?bbb5", bbb5);
cb.Parameters.Add("?bbb6", bbb6);
cb.Parameters.Add("?bbb7", bbb7);
cb.Parameters.Add("?bbb8", bbb8);
cb.Parameters.Add("?bbb9", bbb9);
cb.Parameters.Add("?bbb10", bbb10);
cb.Parameters.Add("?bbb11", bbb11);
cb.Parameters.Add("?bbb12", bbb12);
cb.Parameters.Add("?bbb13", bbb13);
cb.Parameters.Add("?bbb14", bbb14);
cb.Parameters.Add("?bbb15", bbb15);
cb.Parameters.Add("?bbb16", bbb16);
cb.Parameters.Add("?bbb17", bbb17);
cb.Parameters.Add("?bbb18", bbb18);
cb.Parameters.Add("?bbb19", bbb19);
cb.Parameters.Add("?bbb20", bbb20);
cb.Parameters.Add("?bbb21", bbb21);
cb.Parameters.Add("?bbb22", bbb22);
cb.Parameters.Add("?bbb23", bbb23);
cb.Parameters.Add("?bbb24", bbb24);
cb.Parameters.Add("?bbb25", bbb25);
cb.Parameters.Add("?bbb26", bbb26);
cb.Parameters.Add("?bbb27", bbb27);
cb.Parameters.Add("?bbb28", bbb28);
cb.Parameters.Add("?bbb29", bbb29);
cb.Parameters.Add("?bbb30", bbb30);
cb.Parameters.Add("?bbb31", bbb31);
cb.Parameters.Add("?bbb32", bbb32);
cb.Parameters.Add("?bbb33", bbb33);
```

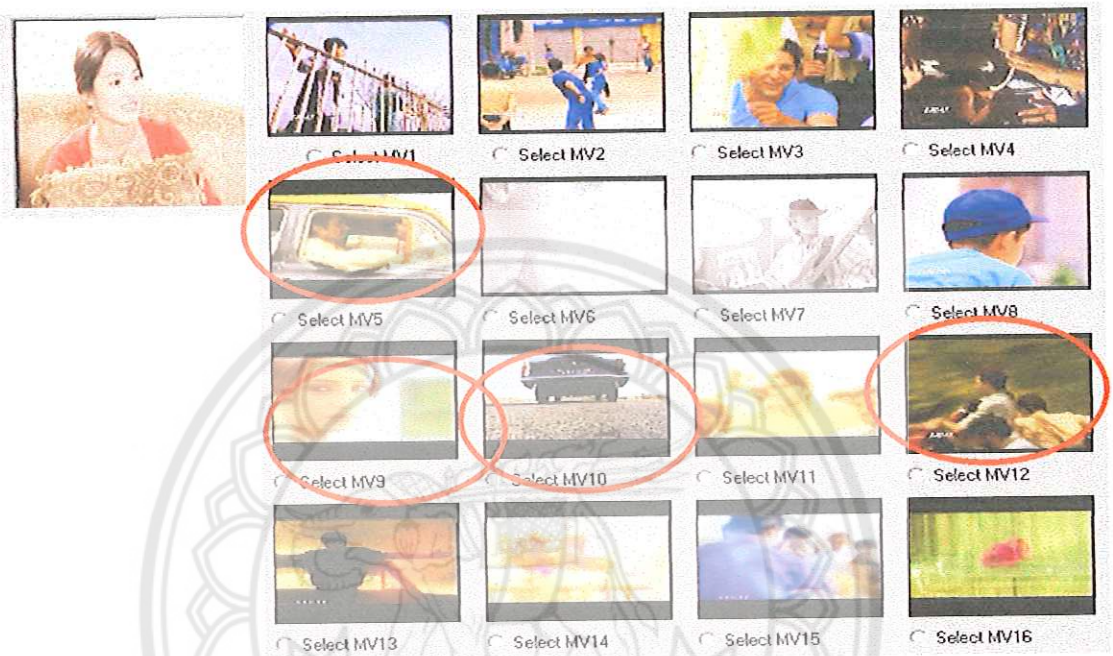
```
cb.Parameters.Add("?bbb34", bbb34);
cb.Parameters.Add("?bbb35", bbb35);
cb.Parameters.Add("?bbb36", bbb36);
cb.Parameters.Add("?bbb37", bbb37);
cb.Parameters.Add("?bbb38", bbb38);
cb.Parameters.Add("?bbb39", bbb39);
cb.Parameters.Add("?bbb40", bbb40);
cb.Parameters.Add("?bbb41", bbb41);
cb.Parameters.Add("?bbb42", bbb42);
cb.Parameters.Add("?bbb43", bbb43);
cb.Parameters.Add("?bbb44", bbb44);
cb.Parameters.Add("?bbb45", bbb45);
cb.Parameters.Add("?bbb46", bbb46);
cb.Parameters.Add("?bbb47", bbb47);
cb.Parameters.Add("?bbb48", bbb48);
cb.Parameters.Add("?bbb49", bbb49);
cb.Parameters.Add("?bbb50", bbb50);
cb.Parameters.Add("?bbb51", bbb51);
cb.Parameters.Add("?bbb52", bbb52);
cb.Parameters.Add("?bbb53", bbb53);
cb.Parameters.Add("?bbb54", bbb54);
cb.Parameters.Add("?bbb55", bbb55);
cb.Parameters.Add("?bbb56", bbb56);
cb.Parameters.Add("?bbb57", bbb57);
cb.Parameters.Add("?bbb58", bbb58);
cb.Parameters.Add("?bbb59", bbb59);
cb.Parameters.Add("?bbb60", bbb60);
cb.ExecuteNonQuery();

cb = new MySqlCommand();// ตัวแปร
cb.CommandText=   "select path,sum(dist)as alldist from avi
group by path order by alldist  limit 16 ";
cb.Connection = conn;

MySqlDataReader myReader;
myReader = cb.ExecuteReader();
```

ภาคผนวก ข
การทดลองค้นหาวีดีโอจากไฟล์ที่ไม่มีในฐานข้อมูล

ข-1 แบบไฟล์รูปภาพ



รูปที่ ข-1 แสดงผลการค้นหาวีดีโอจากไฟล์ภาพที่ไม่มีในฐานข้อมูล

ผลที่ได้อาจจะไม่เป็นไปตามที่ต้องการมากนักเพราะไฟล์ภาพตัวอย่างที่นำมาค้นหา จะไม่มีในฐานข้อมูล เพราะฉะนั้นผลที่ออกมาจึงมองที่องค์ประกอบโดยรวมของคลิปวีดีโอที่มีลักษณะที่คล้ายกับ ไฟล์รูปภาพ จากตัวอย่างทำให้ได้ไฟล์คลิปวีดีโอที่มีลักษณะใกล้เคียงกับไฟล์ภาพทั้งหมด 1 ไฟล์ และสามารถคำนวณค่า % precision ได้ออกมาเป็น

$$\%\text{Precision} = \left(\frac{4}{16}\right) \times 100 \quad = 25 \ \%$$

ข-2 แบบไฟล์วีดีโอ 1 Key frame

รูปที่ ข-2 แสดงผลการค้นหาวีดีโอจากไฟล์คลิปวีดีโอแบบ 1 Key frame โดยที่ไม่มีในฐานข้อมูล

เช่นเดียวกัน ในการค้นหาไฟล์วีดีโอแบบ 1 Key frame ที่ไม่มีในฐานข้อมูล เราก็จะพิจารณาที่องค์ประกอบโดยรวมของไฟล์คลิปวีดีโอที่มีความคล้ายคลึงกันระหว่างวีดีโอตัวอย่างกับวีดีโอในฐานข้อมูล จะทำให้ได้ผลการทดลองดังรูปที่ ข-2 เพราะฉะนั้นจะสามารถคำนวณค่า % precision ได้ดังนี้ คือ

$$\%\text{Precision} = \left(\frac{6}{16}\right) \times 100 \quad = 37.5\ \%$$

ข-3 แบบไฟล์วีดีโอ 3 Key frame

รูปที่ ข-2 แสดงผลการค้นหาวีดีโอจากไฟล์คลิปวีดีโอแบบ 3 Key frame โดยที่**ไม่มีในฐานข้อมูล**

ส่วนในกรณีของ การค้นหาแบบ 3 Key frame ทำให้ได้ภาพที่มีความ**ถูกต้องอยู่ใน**ตำแหน่งที่ดีขึ้น ส่วนค่า %Precision นั้นอาจมีค่าไม่แตกต่างกัน ซึ่งขึ้นอยู่กับข้อมูลในฐานข้อมูลนั่นเอง

$$\%\text{Precision} = \left(\frac{8}{16}\right) \times 100 \quad = \ 50 \ \%$$

ประวัติผู้เขียนโครงงาน

ชื่อ นายวรุตม์ แสงขัติ

ภูมิลำเนา 169 หมู่บ้านเชียงใหม่นิเวศน์ ต.สันพระเนตร
อ.สันทราย จ. เชียงใหม่ 50210

ประวัติการศึกษา

- จบมัธยมศึกษาจาก โรงเรียนยุพราชวิทยาลัย
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
  สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

E-mail : **marco2uja@hotmail.com**

ชื่อ นางสาวศศิตา ธนวรรณ

ภูมิลำเนา 226 หมู่ 5 ต.หนองหญ้าไซ อ.หนองหญ้าไซ
จ.สุพรรณบุรี 72240

ประวัติการศึกษา

- จบมัธยมศึกษาจาก โรงเรียนกรรณสูตศึกษาลัย
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
  สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

E-mail : **st_468@hotmail.com**

ชื่อ นายเกิดพงศ์ พึ่งกริม

ภูมิลำเนา 2 หมู่ 2 ต.สามเรือน อ.ศรีสำโรง จ.สุโขทัย 64120

ประวัติการศึกษา

- จบมัธยมศึกษาจาก โรงเรียนศรีสำโรงชนูปถัมภ์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
  สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

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