การสืบค้นข้อมูลดิจิตอลวิดีโอ โดยใช้เทคนิคดรรชนีเฟรมหลัก และระบบป้อนกลับ

VIDEO RETRIEVAL USING KEY-FRAME VIDEO INDEXING TECHNIQUE AND RELEVANCE FEEDBACK

นายพิษณุกรณ์ เทียนสว่าง รหัส 43370584

นายเกษมสรรญ์ จันทรา รหัส 43370774

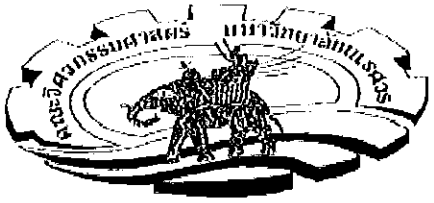
ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 11 ก.ค. 2553
เลขทะเบียน 1500840
เลขเรียกหนังสือ ปร ว546 2546
มหาวิทยาลัยนเรศวร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2546



ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	การสืบค้นข้อมูลดิจิตอลวิดีโอ โดยใช้เทคนิคดรรชนีเฟรมหลักและระบบป้อนกลับ	
ผู้เสนอโครงงาน	นายพิษณุกรณ์ เทียนสว่าง	รหัส 43370584
	นายเกษมสรรญ์ จันทรา	รหัส 43370774
อาจารย์ที่ปรึกษา	ดร.ไพศาล มุณีสว่าง	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์	
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์	
ปีการศึกษา	2546	

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะกรรมการสอบโครงงานวิศวกรรม

.....กรรมการ
(ดร.ไพศาล มุณีสว่าง)

.....กรรมการ
(ดร.สุชาติ แย้มเม่น)

.....กรรมการ
(ดร.สุรเชษฐ์ กานต์ประชา)

หัวข้อโครงงาน	การสืบค้นข้อมูลดิจิตอลวิดีโอ โดยใช้เทคนิคดรรชนีเฟรมหลักและระบบการป้อนกลับ
ผู้ดำเนินโครงงาน	นายพิษณุกรณ์ เทียนสว่าง รหัส 43370584
	นายเกษมสรรญ์ จันทรา รหัส 43370774
อาจารย์ที่ปรึกษา	ดร. ไพศาล มุณีสว่าง
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2546

บทคัดย่อ

โครงงานนี้เป็นการศึกษาและพัฒนาโปรแกรมเพื่อช่วยในการค้นหา Video File ในระดับ Shot โดยให้ผู้ใช้งานทำการเลือก Key-Frame ที่ต้องการค้นหา และเมื่อได้ผลลัพธ์แล้ว ผู้ใช้งานจะสามารถทำการป้อนกลับของกลุ่ม Key-Frame เพื่อนำไปประมวลผลอีกครั้ง โดยให้ผลลัพธ์ที่มีความถูกต้องมากขึ้น จึงได้มีการนำหลักการและทฤษฎีเกี่ยวกับ Relevance Feedback เข้ามาช่วยในการป้อนกลับของกลุ่มข้อมูลที่เป็น Key-Frame พร้อมทั้งต้องการให้ผู้ใช้งานได้มีการโต้ตอบกับคอมพิวเตอร์

Project Title	Video Retrieval Using Key-Frame Video Indexing Technique And Relevance Feedback
Name	Mr. Phitsanukorn Theansawang ID. 43370584
	Mr. Kasemsan Juntra ID. 43370774
Project Advisor	Dr. Paisan Muneesawang
Major	Computer Engineering
Department	Electrical and Computer Engineering
Academic Year	2003

ABSTRACT

This project is studied and developed a program for help to find Video File in level Shot. By user have to choose the Key-frame that you want to find. And when you get the result. The user can return the result of the Key-frame group the user can bring to process again. By the result is more accurate. So we bring principle and theory about Relevance Feedback to help to return the result which data that Key-frame. And also would like the user can make the communicate with computer.

กิตติกรรมประกาศ

โครงงานวิศวกรรมคอมพิวเตอร์ สำเร็จได้ด้วยดีก็เนื่องด้วยความอนุเคราะห์จากอาจารย์ที่ปรึกษาโครงงานคือท่านดร.ไพศาล มุณีสว่าง ผู้ซึ่งกรุณาให้ความรู้ คำแนะนำและเอาใจใส่เป็นอย่างดีระหว่างการดำเนินโครงงาน อีกทั้งยังตรวจสอบแก้ไขข้อบกพร่องต่าง ๆ จนโครงงานนี้เสร็จสมบูรณ์ คณะผู้จัดทำจึงขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบคุณท่านดร.สุชาติ แย้มเม่นและท่านดร.สุรเชษฐ์ กานต์ประชา ที่กรุณารับเป็นกรรมการตรวจสอบโครงงานและให้คำแนะนำ ตรวจสอบแก้ไขโครงงานทำให้โครงงานนี้สมบูรณ์ยิ่งขึ้น

ขอขอบคุณเพื่อน ๆ ที่มีส่วนช่วยเหลือในการทำโครงงานนี้ทุก ๆ คน ที่คอยแนะนำทั้งในส่วนของตัวโปรแกรมเองและในส่วนของเนื้อหาของรายงานโครงงาน

ในโอกาสนี้ทางคณะผู้จัดทำโครงงานจึงขอขอบคุณทุก ๆ ท่านที่มีส่วนช่วยทำให้โครงงานนี้ประสบความสำเร็จได้ด้วยดี

นายพิษณุกรณ์ เทียนสว่าง
นายเกษมสรรญ์ จันทรา

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... จ

สารบัญรูป ..... ฉ

บทที่ 1 บทนำ

1.1 ปัญหาและที่มาของปัญหา 1

1.2 วัตถุประสงค์ 2

1.3 ขอบข่ายของโครงงาน 3

1.4 ผลที่คาดว่าจะได้รับ 3

1.5 งบประมาณ 4

บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.1 ระบบสี 6

2.2 ระบบสีแบบ RGB 6

2.3 ระบบสีแบบ HSV 8

2.4 การอธิบายถึงภาพต่าง ๆ โดยใช้ Color Histogram 10

2.5 การทำ Histogram ในระบบสี HSV 11

2.6 วิธีการทางวิดีโอในระดับส่วนย่อย ๆ ของวิดีโอ (Shot Segmentation) 11

2.7 การหาส่วนย่อย ๆ ของวิดีโอไฟล์ (Shot Boundary) 12

2.8 การทำดรรชนีเฟรมหลัก (Key-Frame) 14

2.9 การเปรียบเทียบความคล้ายคลึงกัน (Similarity Matching) 15

2.10 เทคนิคการป้อนกลับ (Relevence Feedback) 17

2.11 ทำความรู้จักกับ JSP 17

สารบัญ (ต่อ)

หน้า

บทที่ 3 วิธีดำเนินการวิจัย

3.1 ขั้นตอนในการแบ่งแยกวิดีโอไฟล์ออกเป็นส่วยย่อย ๆ ของวิดีโอ (Video Segmentation) 22

3.2 ขั้นตอนในการหาดรรชนีเฟรมหลัก (Key-Frame) ของวิดีโอ Shot 24

3.3 การนำ Key-Frame มาทำการหาค่าสี (Color Model) 28

3.4 การนำ Key-Frame มาทำ Color Histogram 29

3.5 การค้นหา Video Shot โดยใช้ค่าความแตกต่างระหว่างรูปภาพของ Key-Frame 30

3.6 การเพิ่มประสิทธิภาพในการค้นหาวิดีโอ Shot โดยใช้ Relevance Feedback 33

บทที่ 4 ผลการทดลอง

4.1 ผลการทดลองโดยใช้ดรรชนีเฟรมหลักในการค้นหาวีดีโอ Shot 36

4.2 ผลการทดลองโดยใช้หลักการและทฤษฎีเกี่ยวกับเทคนิคการป้อนกลับ (Relevance Feedback) 39

บทที่ 5 สรุปผลและวิเคราะห์ผล

5.1 สรุปผลการทดลองที่ได้จากการพัฒนาโปรแกรม 47

5.2 ผลที่ได้จากการวิเคราะห์ 47

5.3 ปัญหาและการแก้ปัญหา 48

5.4 แนวทางสำหรับการพัฒนาในครั้งต่อไป 48

เอกสารอ้างอิง 49

ภาคผนวก

ก คู่มือการติดตั้งโปรแกรมและการใช้งานของโปรแกรม 50

ข Source Code JAVA and JSP 66

ประวัติผู้จัดทำโครงงาน 90

สารบัญตาราง

ตาราง		หน้า
4.1	ผลที่ได้จากการสุ่มค้นหาวิดีโอ Shot ที่มีลักษณะใกล้เคียงกับวิดีโอ Shot ตั้งต้น	43

มหาวิทยาลัยนเรศวร

สารบัญรูป

รูปที่		หน้า
2.1	แสดงระบบสี	6
2.2	แสดงลูกบาศก์ของสีในระบบแบบ RGB	7
2.3	แสดงรูปแบบของ RGB Model	8
2.4	แสดงรูปแบบของ RGB Model	8
2.5	แสดงตัวอย่างของระบบสีแบบ HSV	9
2.6	แสดง HSV Color Model	10
2.7	แสดงกระบวนการจัดการวิดีโอ (Video Segmentation) และการหาเฟรมหลัก	12
2.8	แสดงค่าของฮิสโตแกรม (Histogram) ที่ได้จากเฟรมแต่ละเฟรม	13
2.9	การค้นหาครรชนีเฟรมหลัก (Key-Frame) จาก Shot แต่ละ Shot	14
2.10	แสดงค่าฮิสโตแกรมที่ได้จากครรชนีเฟรมหลักแต่ละอัน	15
2.11	แสดงการสร้างเนื้อหาแบบไดนามิคด้วยสมาชิกต่าง ๆ ของ JSP	18
2.12	แสดงผังการทำงานของ JSP	20
2.13	แสดงขั้นตอนการประมวลผลไฟล์ JSP ในช่วง Translation	21
3.1	แสดงการเรียกใช้โปรแกรม Windows Movie Maker	22
3.2	การ Import วิดีโอไฟล์ที่ต้องการจะแบ่งออกเป็นวิดีโอ Shot	23
3.3	ผลลัพธ์ที่ได้จากการแบ่งวิดีโอไฟล์ออกเป็นส่วนย่อย ๆ ของวิดีโอ (Video Shot)	23
3.4	แสดงลักษณะของโปรแกรม TMPGEnc	25
3.5	แสดงการเลือกไฟล์วิดีโอที่ต้องการแปลงไฟล์นามสกุล	25
3.6	แสดงไฟล์ต่าง ๆ ที่มีอยู่ในพาธไดเรกทอรี C:\MATLAB6p5\work	26
3.7	แสดงลำดับขั้นตอนในการใช้คำสั่งเพื่อหา Key-Frame	27
3.8	แสดงตัวอย่างของการเลือกภาพที่จะใช้ในการค้นหา	30
3.9	แสดงขั้นตอนหลังจากที่เลือก Key-Frame ที่ต้องการได้แล้ว	31
3.10	แสดงผลลัพธ์ของการค้นหาจาก Key-Frame ที่เลือกในรูปที่ 3.2	33
3.11	แสดงขั้นตอนในการเลือก Key-Frame เพื่อนำไปทำ Relevance Feedback	34
4.1	แสดงรูปภาพที่เป็น Key-Frame จากฐานข้อมูลบางส่วนเพื่อจะทำการเลือก	36
4.2	แสดงขั้นตอนการเลือก Key-Frame เพียงอันเดียวเพื่อให้ในการค้นหาวิดีโอ Shot	37
4.3	แสดงผลลัพธ์ที่ได้บางส่วนจากการใช้ Key-Frame ชื่อ Video2-clip84	37

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.4	แสดงผลลัพธ์ทั้งหมดจากการค้นหาภาพโดยใช้ Key-Frame	38
4.5	แสดงขั้นตอนการเลือกกลุ่ม Key-Frame เพื่อนำไป Feedback	39
4.6	แสดงผลลัพธ์ที่ได้จากการทำ Relevance Feedback	40
4.7	แสดงกลุ่มของKey-Frame ที่ทำการป้อนกลับอีกครั้งเพื่อให้ได้ข้อมูลที่ถูกต้องมากขึ้น	41
4.8	แสดงผลลัพธ์จากการป้อนกลับอีกครั้งของกลุ่ม Key-Frame	41
4.9	ลำดับของผลลัพธ์ที่มีลักษณะของวิดีโอ Shot ที่คล้ายคลึงกัน	45
ก-1	ขั้นตอนการเลือกไดเรกทอรีที่ต้องการติดตั้ง J2SE	51
ก-2	แสดงการเลือกคอมโพเนนต์ที่ต้องการติดตั้ง	52
ก-3	แสดงขั้นตอนความคืบหน้าในการติดตั้งโปรแกรม	53
ก-4	แสดงขั้นตอนการเสร็จสิ้นการติดตั้งโปรแกรม J2SE	54
ก-5	แสดงไฟล์และไดเรกทอรีย่อยภายในไดเรกทอรีที่ติดตั้ง J2SE	54
ก-6	แสดงขั้นตอนการกำหนดพาธให้กับ Window 2000 และ Windows XP	56
ก-7	แสดงการใช้คำสั่ง startup เพื่อเริ่มการทำงานของ Tomcat	58
ก-8	แสดงหน้าต่างที่ pop up ขึ้นมาเมื่อใช้คำสั่ง startup	58
ก-9	แสดงโฮมเพจ Tomcat ภายในเครื่องของเรา	59
ก-10	แสดงความคืบหน้าของการติดตั้งโปรแกรม MySQL	60
ก-11	แสดงขั้นตอนการเสร็จสิ้นการติดตั้งโปรแกรม MySQL	61
ก-12	แสดงการตั้งชื่อเพื่อเชื่อมการติดต่อกับฐานข้อมูล MySQL	62
ก-13	แสดงข้อมูลตารางและฟิลด์ที่มีอยู่ในฐานข้อมูล DatabaseShot	62
ก-14	แสดงข้อมูลที่เป็นฟิลด์ทั้งหมดในตารางฐานข้อมูลที่ชื่อว่า Hsvshot	63
ก-15	แสดงหน้าต่างที่ใช้เลือกฐานข้อมูลที่จะ import มายัง MySQL	64
ก-16	แสดงพาธของข้อมูลที่ได้ทำการเลือกเพื่อที่จะ import ลง MySQL	65

บทที่ 1

บทนำ

ในการค้นข้อมูลประเภทวีดีโดผ่านทางอินเตอร์เน็ต (Internet) หรือในข้อมูลฐานวิดีโอ (Video Database) ได้มีการนำเอาเทคนิคการค้นหาข้องมูลที่เรียกว่า Content-based Video Retrieval (CBVR) มาใช้เพื่อให้สนองความต้องการของผู้ใช้ที่ต้องการความถูกต้องและความรวดเร็ว ในระบบ CBVR นี้ผู้ใช้สามารถค้นหาวีดีโอในระดับ Shot ได้โดยการส่งตัวอย่างของวิดีโอ Shot ให้กับคอมพิวเตอร์และให้คอมพิวเตอร์เป็นตัวประมวลผลเนื้อหาไฟล์ของวิดีโอที่ผู้ใช้ต้องการจากฐานข้อมูล เทคนิคที่นำมาใช้ในระบบ CBVR ประกอบไปด้วยส่วนต่าง ๆ ดังนี้

1. Video Segmentation
2. Color Model HSV
3. Color Histogram
4. Relevance Feedback

1.1 ปัญหาและที่มาของปัญหา

การทำดรรชนีของวีดีโอไฟล์ในฐานข้อมูลสามารถทำได้โดยใช้เทคนิคที่เรียกว่า ดรรชนีของเฟรมหลัก Key-Frame Based Video Identity (KFVI) ในหลักการนี้จะมีการเลือกเฟรมหลักภายในวีดีโอมาเป็นตัวแทนของเนื้อหาของวิดีโอ เพื่อไว้ในการเปรียบเทียบความคล้ายคลึงกันของวีดีโอในกระบวนการค้นหา หลักการของ KFVI จะให้ความถูกต้องมากกว่าการใช้เทคนิคของคำอธิบาย (Content-Based Methods) เพราะว่า KFVI สามารถอธิบายข้อมูลพื้นฐานของวีดีโอ ตัวอย่างเช่น สีและการเคลื่อนไหวของวัตถุได้ดีกว่าตัวหนังสือที่นำมาไว้ในระบบ Text-Based Method นอกจากนี้ KFVI ยังสามารถใช้ในการทำดรรชนีเพื่อจัดเก็บข้อมูลที่มีอยู่อย่างมหาศาลได้อย่างอัตโนมัติ ในขณะที่ Content-Based Methods ต้องใช้คนเป็นผู้ใส่ข้อมูลอธิบายวิดีโอ

ดังนั้นคณะผู้จัดทำจะดำเนินการศึกษาและค้นคว้าการทำดรรชนีของข้อมูลวิดีโอโดยใช้วิธี KFVI เพื่อมาจัดการฐานข้อมูลได้อย่างมีประสิทธิภาพ

จากการศึกษาเรื่องของ CBVR ได้มีการค้นพบว่าปัญหาขั้นพื้นฐานประกอบด้วยสองส่วน คือ

1. ความไม่ถูกต้องของการประมวลผลของคอมพิวเตอร์

2. ความไม่เข้าใจระหว่างคอมพิวเตอร์กับผู้ใช้

ในปัญหาข้อที่ 1 คอมพิวเตอร์ใช้ข้อมูลพื้นฐานของวิดีโอมาตัดสินความเหมือนของวิดีโอ (ตัวอย่างเช่น สีและการเคลื่อนไหวของวัตถุในวิดีโอ) ซึ่งโดยทั่วไปแล้วข้อมูลดังกล่าวไม่มีความถูกต้องและไม่สามารถอธิบายความหมาย (Semantic Meaning) ของวิดีโอได้ดีเท่าที่ควร

ในปัญหาข้อที่ 2 ผู้ใช้ที่มีความต้องการในการค้นหาข้อมูลวิดีโอที่ทำให้คอมพิวเตอร์มีความเข้าใจยาก ผู้ใช้บางท่านอาจจะมีความต้องการที่แตกต่างกัน ขึ้นอยู่กับทัศนคติของแต่ละบุคคลรวมทั้งเวลา

มีการค้นคว้าและศึกษาว่าการแก้ปัญหาทั้งสองข้อดังกล่าวทำได้โดยการให้คอมพิวเตอร์โต้ตอบกับผู้ใช้เพื่อศึกษาและปรับปรุงความเข้าใจระหว่างทั้งสองฝ่าย วิธีการนี้เรียกว่า User Interactive Interface หรือ Relevance Feedback ในวิธีนี้ผู้ใช้นำเสนอข้อมูลวิดีโอต่อคอมพิวเตอร์และคอมพิวเตอร์จะนำไปประมวลผลและนำวิดีโอไฟล์ในฐานข้อมูลแสดงให้กับผู้ใช้ หลังจากนั้นผู้ใช้กระทำการตรวจสอบข้อมูลดังกล่าวว่าถูกต้องหรือไม่และจะทำการป้อนกลับผลการตรวจสอบให้กับคอมพิวเตอร์ เพื่อให้คอมพิวเตอร์ได้วิเคราะห์อย่างละเอียดโดยเอาผลการตรวจสอบเป็นตัวมาตรฐาน

วิธีการ Relevance Feedback มีประสิทธิภาพมากและมีการนำมาใช้เพื่อแก้ปัญหาที่ค้นพบในระบบของ CBIR แต่อย่างไรก็ตามการนำระบบ User Interactive Interface มาใช้ในการค้นหาวิดีโอซึ่งมีการกระทำกันน้อยเพราะว่าเทคโนโลยีนี้ยังใหม่

ในโครงงานนี้ผู้จัดจะนำเอาโปรแกรม JAVA ซึ่งเป็นภาษาทางด้าน Object Orientation มาใช้ในการพัฒนาระบบเพราะเห็นว่าเป็นโปรแกรมดังกล่าวมีประสิทธิภาพและมีการอำนวยความสะดวกในการพัฒนาโปรแกรมที่ใช้กับอินเตอร์เน็ตและมัลติมีเดียในปัจจุบัน

1.2 วัตถุประสงค์

1.2.1 เพื่อเรียนรู้การทำงานด้านการสืบค้นข้อมูลดิจิตอลประเภทวิดีโอไฟล์แล้วนำไปใช้ในการจัดการข้อมูลจำพวกวิดีโอไฟล์ที่มีอยู่มากมายอย่างกระจัดกระจาย ทำการจัดการข้อมูลให้เป็นระเบียบเรียบร้อยโดยจัดเป็นกลุ่ม ๆ

1.2.2 เพื่อสร้างระบบการสืบค้นข้อมูลโดยใช้ความรู้เกี่ยวกับ Content-Based Video Retrieval ในการค้นหาวิดีโอ Shot ที่ผู้ใช้ต้องการโดยใช้ดรรชนีเฟรมหลัก (Key-Frame) เป็นตัวบ่งชี้ถึงรายละเอียดในวิดีโอไฟล์นั้น ๆ

1.2.3 เพื่อพัฒนาประสิทธิภาพการค้นหาวิดีโอ Shot โดยใช้ความรู้เกี่ยวกับเทคนิคการป้อนกลับของข้อมูล (Relevance Feedback) เข้ามาช่วยในการติดต่อระหว่างผู้ใช้กับคอมพิวเตอร์ เพื่อความถูกต้องแม่นยำของผลลัพธ์

1.2.4 เพื่อสร้างฐานข้อมูลที่เกี่ยวข้องกับวิดีโอไฟล์ให้มีลักษณะการเรียกใช้ที่สะดวกและง่ายต่อการเรียกใช้ อีกทั้งมีความยืดหยุ่นต่อการนำไปใช้งานด้านต่าง ๆ โดยใช้โปรแกรม JAVA ในการพัฒนา Application ในกระบวนการ Content-Based Video Retrieval และสามารถรองรับการทำงานผ่านเครือข่าย Internet สำหรับผู้ใช้งาน

1.3 ขอบข่ายของโครงงาน

ในการทำงานนั้นเราได้วางขอบข่ายการทำงานอย่างคร่าว ๆ ที่เราสนใจไว้คือ เริ่มจากการพัฒนาตอนต้นเข้าใจลักษณะของ Video Modeling ลงไปจนถึงการทำโปรแกรมที่ได้จากการ เรียนรู้ปัญหาและพัฒนาเพื่อแก้ปัญหาดังกล่าว เพื่อรองรับใช้งานของผู้ใช้ที่มีความหลากหลายของงานให้สามารถใช้ความรู้จากโครงงาน ซึ่งจำแนกได้ดังนี้คือ

1.3.1 เพื่อพัฒนาระบบฐานข้อมูลในการสืบค้นข้อมูลประเภทวิดีโอหรือวิดีโอ Shot โดยใช้ความรู้เกี่ยวกับ Video Segmentation, Video Indexing และ Video Matching ช่วยในการแบ่งแยกความแตกต่างของวิดีโอ Shot แต่ละ Shot ในฐานข้อมูลประเภทวิดีโอไฟล์

1.3.2 เพื่อพัฒนาการสืบค้นข้อมูลโดยใช้ CBVR (Content-Based Video Retrieval) ในการจำแนกความแตกต่างทางองค์ประกอบของวิดีโอ Shot ที่ผู้ใช้ต้องการโดยใช้ Key-Frame เป็นตัวบ่งชี้ในการค้นหาข้อมูลในฐานข้อมูลประเภทวิดีโอไฟล์

1.3.3 มีการตอบโต้กับผู้ใช้โดยใช้ความรู้เกี่ยวกับเทคนิคการป้อนกลับ (Relevance Feedback) มาช่วยในการเพิ่มประสิทธิภาพให้แก่ระบบมากขึ้นกว่าเก่า

1.3.4 เพื่อที่สามารถนำโปรแกรมที่ได้นี้ไปใช้งานได้ง่าย ผู้จัดทำจึงใช้โปรแกรม JAVA ในการเขียนโปรแกรมเพราะสามารถรองรับการทำงานที่ซับซ้อนได้ และสามารถนำไปใช้งานบนเครือข่ายอินเตอร์เน็ตได้อย่างมีประสิทธิภาพ

1.4 ผลที่คาดว่าจะได้รับ

1.4.1 มีฐานข้อมูลเกี่ยวกับวิดีโอไฟล์ที่เป็นระบบระเบียบมากขึ้น

1.4.2 เพื่อความสะดวกรวดเร็วในการใช้งานทางด้านการสืบค้นข้อมูลทางด้านวิดีโอและง่ายต่อการที่จะนำไปใช้ในงานด้านต่าง ๆ ที่สนับสนุนการทำงานด้าน Video Retrieval

1.4.3 นำความรู้ความเข้าใจทางด้าน Image And Video Retrieval Processing ที่ได้จากการทำโครงงานไปประยุกต์ใช้งานด้านต่าง ๆ ได้อย่างมีประสิทธิภาพขึ้น

1.4.4 เพื่อใช้เป็นมาตรฐานทางด้าน Video Classification ในการพิจารณาหรือวิเคราะห์จำแนกวิดีโอ Shot ที่มีลักษณะคล้ายคลึงกัน

1.5 งบประมาณ

1.5.1 ค่าใช้จ่ายในการซื้อหนังสือ	1,200 บาท
1.5.2 ค่าใช้จ่ายในการทำรายงาน	400 บาท
1.5.3 ค่าใช้จ่ายเบ็ดเตล็ด	400 บาท
รวมทั้งสิ้น	2,000 บาท

กิจกรรมการดำเนินงาน

กิจกรรม	ปี 2545			ปี 2546									
	ต.ค.	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. ศึกษาความรู้เกี่ยวกับ Video Segmentaion													
2. ศึกษากลไกในการ Indexing Video Shot													
3. ศึกษาการ Matching ของ Video Shot													
4. ทำการทดลองผลที่ได้จากการศึกษา													
5. ศึกษาปัญหาที่เกิดขึ้น													
6. ค้นคว้าหาความรู้เรื่อง Relevance Feedback													
7. ศึกษา JAVA ในการพัฒนาโปรแกรม													
8. ทดลองปรับปรุงและแก้ไขให้สมบูรณ์													
9. ตรวจสอบระบบ													
10. จัดทำเอกสารประอบโครงงาน													

บทที่ 2

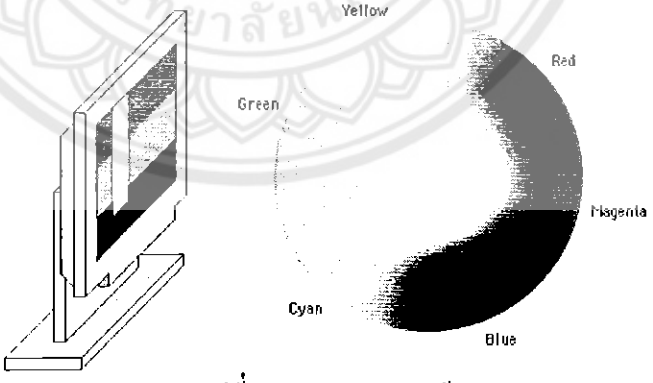
เอกสารและงานวิจัยที่เกี่ยวข้อง

ความรู้ที่ใช้ในการแก้ปัญหาเกี่ยวกับโครงงานนี้มีอยู่หลายหัวข้อ แต่ทางผู้จัดทำมุ่งสนใจในเรื่องของกระบวนการจัดการคุณภาพของสีและการจัดการข้อมูลดิจิตอลประเภทวิดีโอ โดยใช้องค์ประกอบหลัก ๆ ของระบบสีและและองค์ประกอบของข้อมูลดิจิตอลประเภทวิดีโอในระดับส่วนย่อย ๆ ของวิดีโอไฟล์ (Shot) เพื่อใช้เป็นแนวทางในการศึกษาและการทำโครงงาน จึงขออธิบายถึงส่วนของเนื้อหาที่นำมาใช้ในการทำโครงงานเป็นหัวข้อย่อย ๆ ดังต่อไปนี้

2.1 ระบบสี (Color Models)

จุดประสงค์ของการสร้างรูปของระบบสีขึ้นมาเพื่อที่จะสร้างมาตรฐานในการนำสีต่าง ๆ ที่มีอยู่ในธรรมชาติซึ่งมีลักษณะเป็นคลื่นต่อเนื่องกันมาจัดเก็บไว้ในคอมพิวเตอร์ โดยจะจัดเก็บเป็นลักษณะของสัญญาณดิจิตอล ลักษณะของการจัดเก็บโดยทั่วไปจะเก็บไว้ในลักษณะของคู่อันดับของตัวเลขต่าง ๆ เพื่อที่จะแสดงค่าของสีต่าง ๆ ออกมาให้มีความเข้าใจได้ง่ายขึ้น

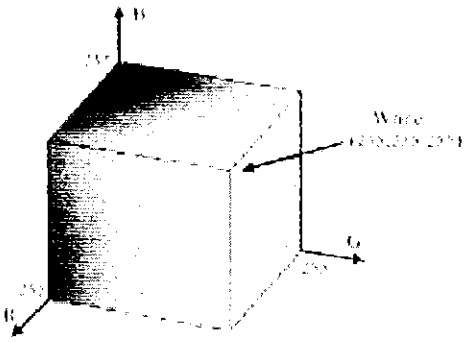
ในปัจจุบันจึงได้มีการนำหลักความรู้ของระบบสีไปใช้ทำเป็นอุปกรณ์ใช้งานเช่น จอมอนิเตอร์ต่าง ๆ หรือนำไปใช้ควบคู่กับงานทางด้าน Graphic Design



รูปที่ 2.1 แสดงระบบสี

2.2 ระบบสีแบบ RGB (RGB Color Model)

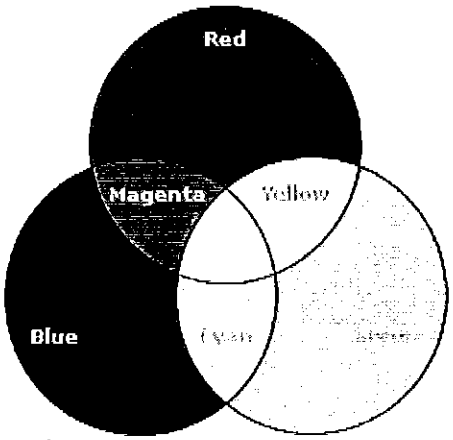
ในระบบสีแบบ RGB นั้นสีแต่ละสีที่แสดงออกมานั้นจะประกอบไปด้วย สีแดง, สีเขียว และสีน้ำเงิน ซึ่งในระบบสีแบบ RGB นี้สีแต่ละสีจะถูกแทนด้วยคู่อันดับต่าง ๆ ตั้งแต่ 0 ถึง 255



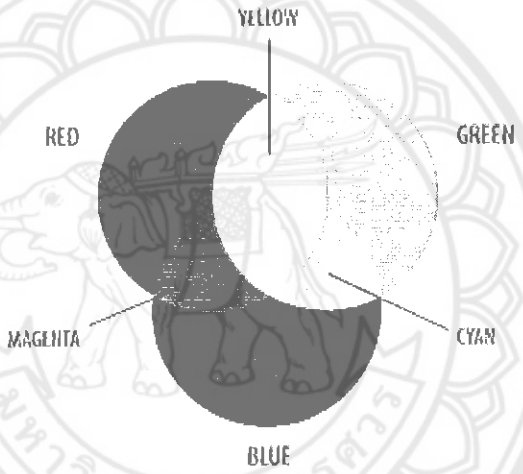
รูปที่ 2.2 แสดงลูกบาศก์ของระบบสีแบบ RGB

ในระบบสี RGB นี้ค่าของสีแต่ละสีจะถูกแทนด้วยตัวเลขจำนวน 8 บิต ดังนั้นจะมีค่าของความเป็นสีแดงทั้งหมดคือ 2^8 หรือ 256 สี (เริ่มตั้งแต่ 0 ถึง 255) ในลักษณะเดียวกันค่าของความเป็นสีเขียวและสีน้ำเงินก็จะมีค่าทั้งหมดอย่างละ 256 สี ดังนั้นจำนวนค่าของสีที่เกิดจากการนำค่าของความเป็นสีแดง, สีเขียวและสีน้ำเงินมารวมกันทั้งหมดจึงมีจำนวนถึง 16,777,216 สี ($2^8*2^8*2^8$) โดยจำนวนค่าของสีที่เกิดขึ้น 16,777,216 สีนี้สามารถใช้ตัวเลขจำนวน 24 บิตในการบ่งบอกค่าของสีได้ สังเกตได้จากตอนที่ผู้ใช้งานคอมพิวเตอร์ต้องการปรับ Resolution ของภาพบนจอมอนิเตอร์ ตัวของระบบปฏิบัติการจะแสดงจำนวนบิตของค่าของสีที่สามารถปรับได้ตั้งแต่ 8 บิต, 16 บิต, 24 บิต และ 32 บิต

ในการทำโครงงานนี้ส่วนที่ใช้เป็นตัวแบ่งแยกค่าของสีนั้นก็คือสายตามนุษย์ซึ่งก็คือผู้ที่ใช้งาน ดังนั้นระบบประสาทตาของมนุษย์จึงไม่สามารถอธิบายค่าของสีได้มากเท่ากับคอมพิวเตอร์ ซึ่งมนุษย์สามารถแบ่งแยกค่าของสีได้ประมาณ 200 ถึง 300 สีเท่านั้น และในระบบสีแบบ RGB นี้จะให้ค่าของสีที่ไม่สอดคล้องกับระบบประสาทตาของมนุษย์ เพราะระบบประสาทตาของมนุษย์ในการมองเห็นค่าของสีนั้นจะสามารถบ่งบอกได้ว่าค่าของสีที่มองเห็นมีความสว่าง ความสดของสีและอื่น ๆ ที่ระบบสีแบบ RGB ไม่สามารถบ่งบอกได้ ดังนั้นจึงได้มีการใช้ระบบสีที่เหมาะสมกับการวิเคราะห์ของอธิบายด้วยระบบประสาทตาของมนุษย์ซึ่งนั้นก็คือระบบสีแบบ HSV โดยลำดับต่อไปจะขออธิบายระบบสีแบบ HSV ที่นำมาใช้เป็นหลักการในการทำโครงงาน



รูปที่ 2.3 แสดงรูปแบบของ RGB Model



รูปที่ 2.4 แสดงรูปแบบของ RGB Model

2.3 ระบบสีแบบ HSV (HSV Color Model)

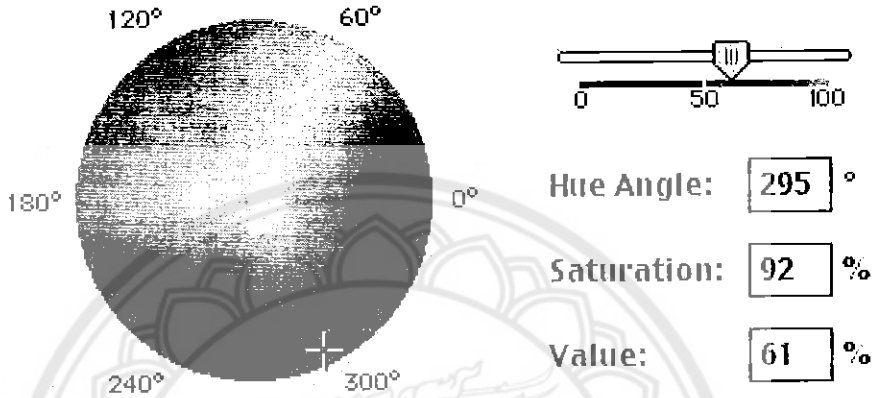
แม้ว่าระบบสีแบบ RGB และระบบสีอื่น ๆ ที่ใกล้เคียงจะสามารถนำมาใช้แสดงผลในอุปกรณ์ดิจิตอลได้เป็นอย่างดีรวมไปถึงระดับการมองเห็นของตามนุษย์ที่รับรู้ค่าของสีแดง, สีเขียว และสีน้ำเงินในแต่ละค่าของสีที่มองเห็นก็ตาม แต่กลับไม่สามารถบ่งบอกถึงรายละเอียดของสีได้ชัดเจนมากและไม่สามารถตอบสนองได้กับรูปแบบในการมองเห็นของมนุษย์ กล่าวโดยย่อคือมนุษย์จะมองเห็นสีในลักษณะของสี, ความเป็นสี (สีจางหรือสีสด) และความมืด, ความสว่างของสีเท่านั้น

ดังนั้นเมื่อมนุษย์มองเห็นวัตถุหนึ่ง ๆ ก็สามารถอธิบายสีของวัตถุนั้นได้ว่ามีสีอะไร ลักษณะของสีเป็นอย่างไร (สีจางหรือสีสด) และมีความสว่างมากน้อยเพียงใดจึงได้มีระบบสีแบบ HSV (Hue, Saturation, Value) ที่สามารถอธิบายได้ถึงสีที่ใกล้เคียงกับระดับสายตาที่มนุษย์มองเห็นมากที่สุด โดยที่ค่าของระบบสีแบบ HSV สามารถอธิบายได้นั้นมีดังต่อไปนี้

H (Hue) จะหมายถึงค่าของสีบริสุทธิ์ว่าเป็นสีอะไร (สีแดง, สีเขียว, สีฟ้า, ...)

S (Saturation) จะกล่าวถึงค่าการปนเปื้อนของแสงสีขาวในค่าของสีนั้น ๆ ซึ่งจะทำให้ได้สีที่มีความสดไปจนถึงสีซีดจางจนกระทั่งเป็นสีขาว

V (Value) ใช้บอกถึงการปนเปื้อนของสีเทาซึ่งทำให้สีมีความเข้มขึ้น



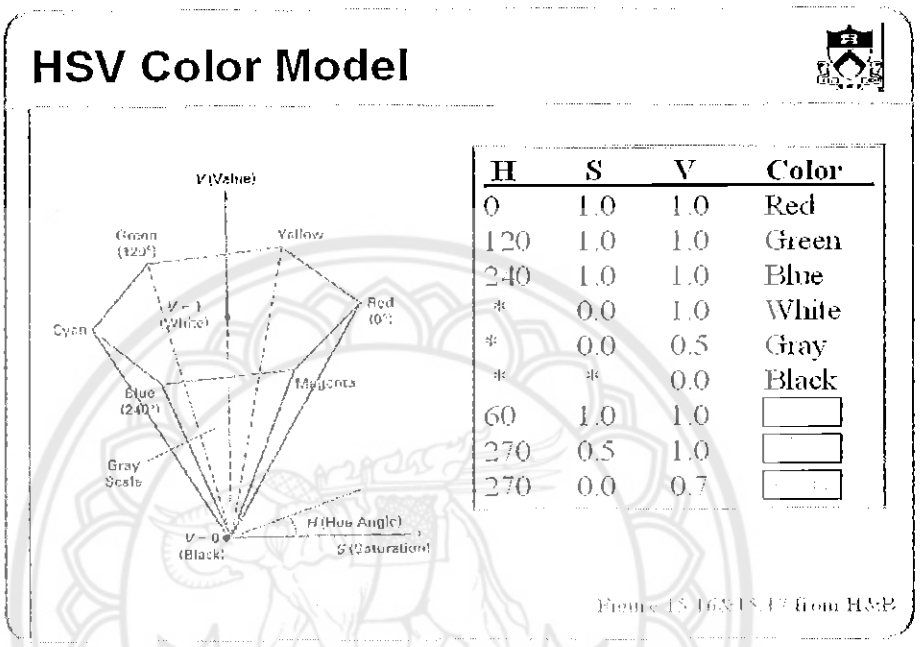
รูปที่ 2.5 แสดงตัวอย่างของระบบสีแบบ HSV

ระบบสีแบบ HSV นี้จึงเป็นพื้นฐานที่ใช้ในการพัฒนากระบวนการต่าง ๆ ที่เกี่ยวกับการมองเห็นของสายตามนุษย์ ภาพที่ได้มาจากการหาดรรชนีเฟรมหลักซึ่งจัดเก็บไว้ในระบบสีแบบ RGB แต่ถ้าจะนำทำกระบวนการใดกระบวนการหนึ่งกับภาพนั้นจะต้องนำภาพนั้นมาผ่านกระบวนการแปลงให้อยู่ในระบบสีแบบ HSV ก่อนที่จะทำในกระบวนการต่อไป

ถ้านำภาพขาวดำ (Intensity Image) ภาพหนึ่งมาอธิบายด้วยค่าของสีในระบบ RGB แล้วนำค่าของจุดสีที่ได้มาพล็อตลงในรูปลูกบาศก์ของระบบสีแบบ RGB ก็จะพบว่าจุดที่ได้จะอยู่บนเส้นที่ลากระหว่างจุด (0, 0, 0) ถึงจุด (1, 1, 1) ซึ่งเส้นนี้เองที่เรียกว่าเส้นความเป็นสีเทาหรือค่าของ V (Value) ในระบบสี HSV นั้นถ้ามีสีใด ๆ อยู่ด้วย ณ ตำแหน่งของค่าสีเท่าค่าใดค่าหนึ่งก็จะอยู่บนระนาบที่ตัดตั้งฉากกับแกนของเส้นแสดงความเป็นสีเทานั้นเอง โดยจุดดังกล่าวจะวางตัวอยู่บนระนาบรอบ ๆ แกนอยู่บนเส้นดังกล่าว ซึ่งองศาของจุดที่วางอยู่นั้นคือค่าของสี (H, Hue) ในระบบสีแบบ HSV นั้นส่วนค่าของความปนเปื้อนของสีขาว (S, Saturation) นั้นจะสามารถอธิบายได้ด้วยระยะห่างจากแกนของสีเทา จากที่กล่าวมาจึงทำให้สามารถสรุปได้ว่าระบบสีแบบ HSV สามารถนำมาใช้แทนที่ระบบสีแบบ RGB ได้

ถ้านำระนาบตัดที่มีค่าของความเป็นสีเทา (Value) ค่าหนึ่งมาเขียนจะได้ภาพในระบบสี HSV จากที่กล่าวไปแล้วว่าระนาบจะมีความปนเปื้อนของสีเทาเปลี่ยนไปตามแกนของค่าความเป็นสีเทา

ส่วนค่าความปนเปื้อนของสีขาวหรือค่าความซีดจางของสี (S, Saturation) ซึ่งจะมีความซีดจางของสีมากที่บริเวณใกล้จุดศูนย์กลางและสีจะมีความสดมากขึ้นเมื่อเคลื่อนที่ออกไปไกลจากจุดศูนย์กลาง



รูปที่ 2.6 แสดง HSV Color Model

2.4 การอธิบายถึงภาพต่าง ๆ โดยใช้ Color Histogram

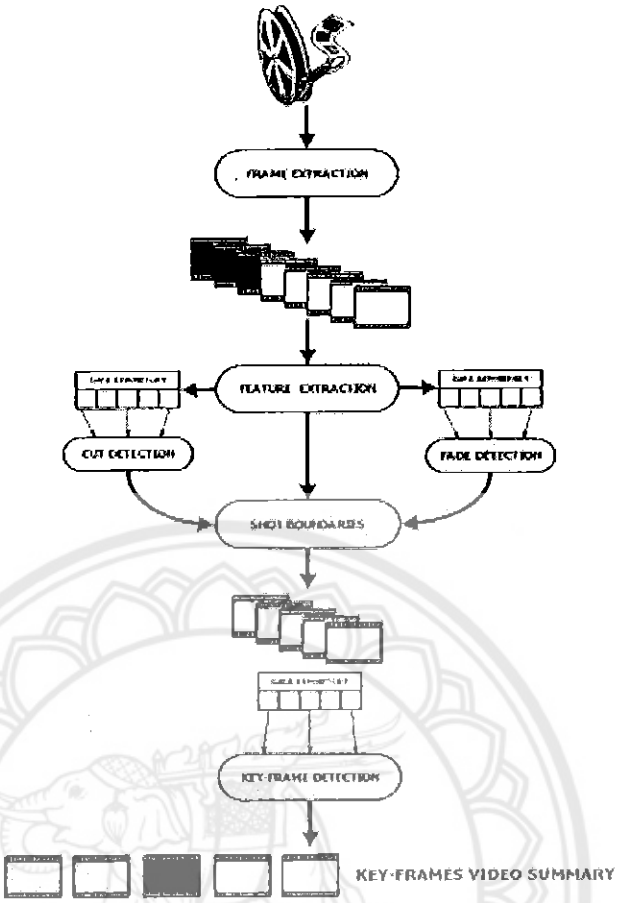
ภาพโดยทั่วไปนั้นจะประกอบขึ้นด้วยจุดของเม็ดสี (Pixel) ต่าง ๆ เป็นจำนวนมากมารวมกันจนเกิดภาพ โดยที่แต่ละจุดเม็ดสีนั้นจะมีค่าที่อธิบายว่าเป็นสีอะไร เช่นในระบบสี RGB ก็จะอธิบายว่าแต่ละจุดเม็ดสีมีค่าของความเป็นสีแดง, สีเขียวและสีน้ำเงินเท่าไร ในการทำฮิสโตแกรม (Histogram) นั้นสามารถทำได้โดยเริ่มจากการสร้างกราฟที่มีจำนวนช่วงเท่ากับจำนวนของช่วงสีที่ต้องการ การกำหนดลำดับชั้นของสีที่ต้องการจากค่าสูงสุดของแต่ละช่วงมาแบ่งเป็นช่วง ๆ ตามจำนวนของลำดับชั้นที่ต้องการ จากนั้นจึงนำจุดของเม็ดสีทุกจุดในภาพมาเปรียบเทียบว่าอยู่ในช่วงใดของกราฟ เมื่อมีจุดของเม็ดสีตกอยู่ในช่วงใดก็จะทำการเพิ่มค่าในช่วงดังกล่าวเป็นจำนวนหนึ่งค่า โดยทำตั้งแต่จุดของเม็ดสีแรกของภาพไปจนถึงจุดของเม็ดสีสุดท้ายของภาพ โดยทั่วไปแล้วการทำ Histogram นั้นจะนำค่าที่ได้ในแต่ละช่วงมาหาอัตราส่วนระหว่างค่าของจำนวนสีในช่วง ๆ นั้นกับจำนวนจุดของเม็ดสีทั้งหมดในภาพ เพื่อที่จะหาอัตราส่วนของช่วงสีหนึ่ง ๆ ซึ่งจะทำให้ทราบว่ามีช่วงสีใดมากในภาพดังกล่าว

2.5 การทำฮิสโตแกรม (Histogram) ในระบบสี HSV

ในการทำฮิสโตแกรมของภาพในระบบสี HSV นั้นจะเป็นการแสดงถึงค่าขององค์ประกอบของแต่ละค่าสีในรูปของระบบสี HSV ถ้าภาพมีสีที่อยู่ในช่วงใดมากช่วงนั้นก็จะได้กราฟที่มีความสูงมาก ค่าของฮิสโตแกรมในช่วงของระบบสี HSV จะมีจำนวนช่วงสีที่ต้องการคือ ค่าผลคูณระหว่างค่าของจำนวนช่วงในแกน Hue, Saturation และ Value ซึ่งจะได้เป็นจำนวนช่วงสีทั้งหมดที่ต้องการ จากนั้นนำค่าของจุดสีที่อยู่ในระบบของ HSV มาเปรียบเทียบว่าอยู่ในช่วงใดโดยการหาค่าของความกว้างของช่วงสีในแต่ละแกนต่อไป

2.6 วิธีการทางวิดีโอในระดับส่วนย่อย ๆ ของวิดีโอ (Shot Segmentation)

ลักษณะของข้อมูลดิจิตอลประเภทวิดีโอไฟล์นั้นจะประกอบไปด้วยฉากต่าง ๆ (Scene) ที่มีความเหมือนกันหรือแตกต่างกันนำมาเรียงต่อกันเป็นเรื่องราวของวิดีโอไฟล์นั้น ๆ เช่นถ้าวิดีโอไฟล์นั้นเป็นเรื่องราวเกี่ยวกับธรรมชาติหรือเป็นเรื่องราวเกี่ยวกับกีฬาแล้ว ในแต่ละฉากนั้นก็ต้องมีส่วนของเนื้อหาที่แตกต่างกัน แต่ใน Scene หนึ่ง ๆ จะมีส่วนที่คล้ายกันคือเฟรม (Frame) ที่นำมาต่อ ๆ กันเป็น Scene แต่ละ Scene ในวิดีโอไฟล์ ถ้าจะอธิบายอย่างคร่าว ๆ ให้เห็นภาพแล้วจะขอยกตัวอย่างเกี่ยวกับการจับภาพในวิดีโอไฟล์ ซึ่งขณะที่เราทำการบันทึกภาพเกี่ยวกับการแข่งขันฟุตบอล ขณะนั้นกำลังจับภาพของการทำประตูอยู่ แล้วเปลี่ยนมาจับภาพของผู้คนที่มาดูฟุตบอล ส่วนนี้เองจะเกิดความแตกต่างกันของภาพที่เราบันทึก ดังนั้น Scene แต่ละ Scene ที่เราทำการบันทึกอยู่นั้นจะแบ่งเป็นส่วนของ Scene ที่มีเรื่องราวเกี่ยวกับการทำประตูกับส่วนของ Scene ที่เกี่ยวกับผู้คนที่มาดูฟุตบอลเป็นต้น ซึ่งในการค้นหาข้อมูลที่ต้องการ Shot ที่เราต้องการซ้ำ ๆ กัน ซึ่งในกระบวนการนี้จะขออธิบายในส่วนต่อไปเพื่อช่วยให้ผู้ใช้มีความสะดวกมากขึ้นต่อการค้นหาข้อมูลดิจิตอลประเภทวิดีโอไฟล์ในระดับ Shot ได้อย่างมาก

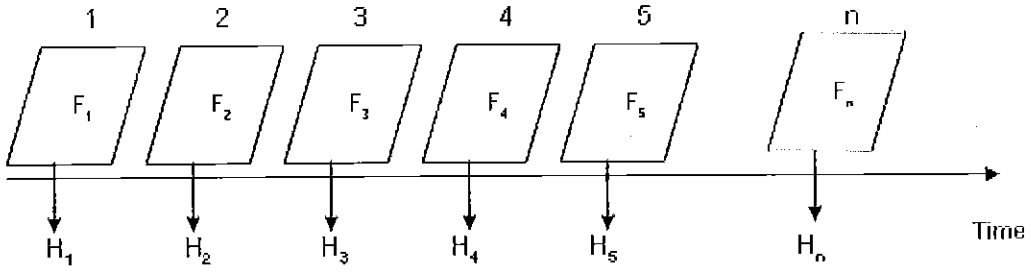


รูปที่ 2.7 แสดงกระบวนการจัดการวิดีโอ (Video Segmentation) และการหาดรรชนีเฟรมหลัก

2.7 การหาส่วนย่อย ๆ ของวิดีโอไฟล์ (Shot Boundary)

ซึ่งในส่วนนี้จะนำความรู้จากการทำฮิสโตแกรม (Histogram) มาใช้โดยอาศัยหลักการที่ว่าเมื่อเฟรมที่นำมาเรียงต่อกันเป็น Shot หรือ Scene นั้นการที่จะเป็น Shot หรือ Scene หนึ่ง ๆ ได้นั้นเฟรมที่นำมาเรียงต่อกันจะต้องมีความคล้ายคลึงกันหรือมีองค์ประกอบของภาพที่เหมือนกันนั่นเอง ถ้าเกิดว่า Shot นั้นเกิดเปลี่ยนไปเป็นอีก Shot หนึ่งเฟรมที่ต่อกันระหว่าง Shot สอง Shot จะต้องมีองค์ประกอบของภาพที่แตกต่างกันมาก

ซึ่งค่าของฮิสโตแกรม (Histogram) ที่ได้สามารถใช้เป็นตัวบ่งบอกองค์ประกอบของภาพได้ ดังนั้นจึงนำเอาค่าฮิสโตแกรมของเฟรมแต่ละเฟรมมาหาค่าเปรียบเทียบความแตกต่างกันของค่าฮิสโตแกรม ซึ่งในที่นี้ค่าความแตกต่างกันของฮิสโตแกรมกำหนดค่าให้เท่ากับ ΔH_i โดยที่ค่า i มีค่าตั้งแต่ 1 ไปเรื่อย ๆ จนกว่าจะหมด Scene ในวิดีโอไฟล์ ซึ่งถ้าในวิดีโอไฟล์นี้มีความยาวของเนื้อหามาก ค่าที่นำมาเปรียบเทียบระหว่างเฟรมต่อเฟรมไปเรื่อย ๆ นี้ก็จะมีค่ามากตามไปด้วย ดังนั้นจึงขอยกตัวอย่างค่าความแตกต่างของฮิสโตแกรมที่ใช้ในการเปรียบเทียบดังนี้



รูปที่ 2.8 แสดงค่าของฮิสโตแกรม (Histogram) ที่ได้จากเฟรมแต่ละเฟรม

ต่อไปจะแสดงรูปแบบของสมการทั่วไปของการเปรียบเทียบฮิสโตแกรม

$$H_i = [h_{R1}\ h_{R2}\ h_{R3}\ h_{R4}\ h_{G1}\ h_{G2}\ h_{G3}\ h_{G4}\ h_{B1}\ h_{B2}\ h_{B3}\ h_{B4}]$$

$$\Delta H_1 = |H_1 - H_2|$$

$$\Delta H_2 = |H_2 - H_3|$$

$$\Delta H_3 = |H_3 - H_4|$$

$$\Delta H_4 = |H_4 - H_5|$$

$$\vdots$$

$$\Delta H_n = |H_n - H_{n+1}| \qquad (2.1)$$

ซึ่งค่าฮิสโตแกรมของเฟรมหรือรูปภาพแต่ละอันที่นำมาเรียงต่อกันในช่วงเวลา โดยจะแสดงให้อยู่ในรูปทั่วไปในการเปรียบเทียบฮิสโตแกรมของเฟรมหรือภาพได้ดังสมการที่ 2.1

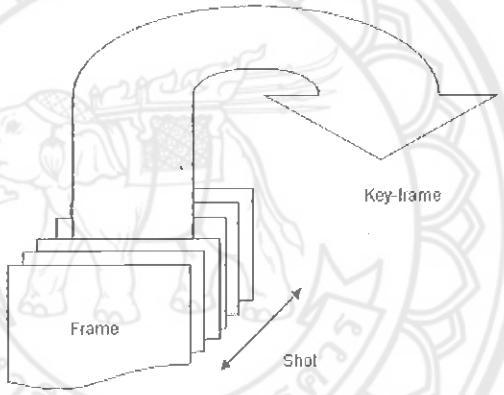
โดยในตัวอย่างที่ยกมาอ้างอิงนี้ค่าความแตกต่างจะเกิดจากการนำเอาค่าความเข้มของสีในระบบสีแบบ RGB ของแต่ละเฟรมหรือภาพ โดยค่าความเข้มที่เลือกมานี้จะต้องมีความเท่ากันหมดในแต่ละเฟรมหรือภาพนั้น ๆ โดยกำหนดให้ค่า h_{R1} มีความหมายคือค่าฮิสโตแกรมที่นับได้ของค่าความเข้มของสีแดง R1 ซึ่งจากตัวอย่างที่เราได้กล่าวไว้แล้ว R1 เปรียบได้กับค่าความเข้มของสีแดงเท่ากับ 50 R2 เท่ากับ 100 R3 เท่ากับ 150 และ R4 เท่ากับ 200 ซึ่งเราอาจจะนำเอาค่าความเข้มของสีค่าอื่นก็ได้มาใช้ในการเปรียบเทียบ ส่วนในค่าความเข้มของสีแดงและค่าความเข้มของสีน้ำเงินก็เช่นเดียวกันโดยค่าความแตกต่างถ้าจะมองให้เข้าใจง่าย ๆ อาจจะแสดงให้อยู่ในรูปสมการที่ 2.2 ดังนี้ โดยค่า ΔH_i จะมีค่าเป็นเลขจำนวนเต็ม

$$\begin{aligned}\Delta H_1 = |H_1 - H_2| = {} & |h_{R11} - h_{R21}| + |h_{R12} - h_{R22}| + |h_{R13} - h_{R23}| + |h_{R14} - h_{R24}| \\ & + |h_{G11} - h_{G21}| + |h_{G12} - h_{G22}| + |h_{G13} - h_{G23}| + |h_{G14} - h_{G24}| \\ & + |h_{B11} - h_{B21}| + |h_{B12} - h_{B22}| + |h_{B13} - h_{B23}| + |h_{B14} - h_{B24}| \qquad (2.2)\end{aligned}$$

2.8 การทำดรรชนีเฟรมหลัก (Key-Frame)

เพื่อให้ง่ายต่อการเปรียบเทียบหรือในการค้นหาข้อมูลเกี่ยวกับวิดีโอไฟล์ที่ต้องการเรื่องราวของเนื้อหาหรือส่วนที่ผู้ใช้ต้องการจะนำไปใช้งานด้านต่าง ๆ นั้นกระบวนการทำงานในระดับ Shot นั้นมีความยุ่งยาก ดังนั้นเราจึงนำเอาความรู้ทางด้าน Image Processing มาช่วยในการค้นหาหรือจัดทำข้อมูลประเภทวิดีโอไฟล์ โดยในส่วนย่อย ๆ ของวิดีโอไฟล์ (Shot) นี้จะนำเอา Shot แต่ละ Shot ที่ประกอบกันเป็นวิดีโอไฟล์นี้มาทำการหาดรรชนีเฟรมหลัก (Key-Frame) ที่จะใช้เป็นตัวบ่งบอกลักษณะของเรื่องราวหรือเนื้อหาในแต่ละ Shot ของวิดีโอไฟล์นั้น ๆ

การที่เราจะหาดรรชนีเฟรมหลักนั้นส่วนใหญ่ ๆ จะนำเอาค่าตรงกลางของเฟรมในแต่ละ Shot มาใช้เป็นส่วนบ่งบอกลักษณะของ Shot นั้น ๆ หรือนำมาเป็นดรรชนีเฟรมหลักนั้นเองโดยแสดงให้เข้าใจง่าย ๆ เป็นรูปภาพได้ดังนี้



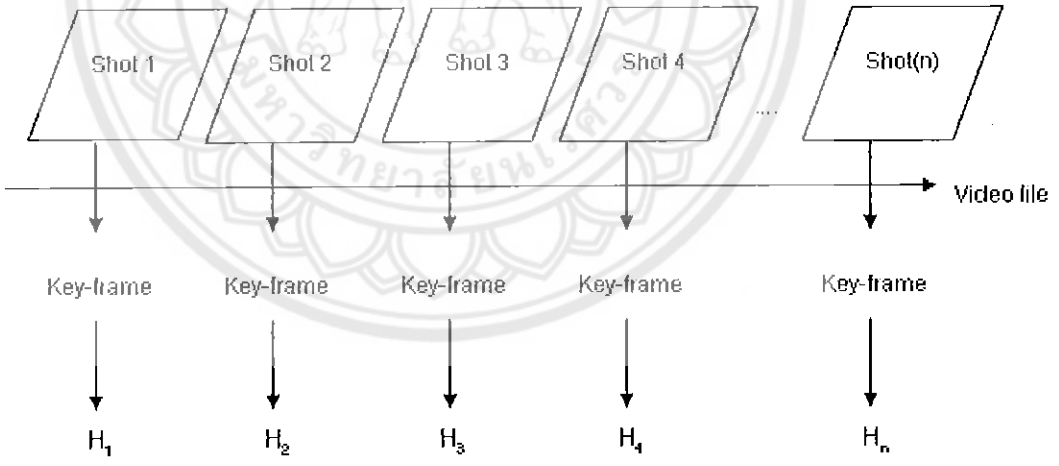
รูปที่ 2.9 การค้นหาดรรชนีเฟรมหลัก (Key-Frame) จาก Shot แต่ละ Shot

สิ่งที่ได้จากการหาดรรชนีเฟรมหลักของวิดีโอไฟล์ในระดับส่วนย่อย ๆ (Shot) นี้มีความสำคัญมากในขั้นตอนต่อ ๆ ไป เพราะเฟรมนี้จะใช้เป็นส่วนในการบ่งบอกเนื้อหาสาระ เรื่องราวที่เกี่ยวข้อง ฉากที่ผู้ใช้ต้องการในแต่ละส่วนของวิดีโอไฟล์และอื่น ๆ โดยนำเสนอออกมาเป็นรูปภาพหรือเป็นเฟรมของ Shot หนึ่ง Shot ในวิดีโอไฟล์ที่ผู้ใช้ต้องการจะค้นหา ซึ่งถ้าเฟรมที่ใช้ในการบ่งบอกลักษณะของเฟรมใน Shot นั้นมีความสามารถในการบ่งบอกลักษณะของ Shot นั้นต่ำถ้าเกิดต้องการค้นหา Shot ที่มีความเหมือนกันหรือคล้ายคลึงกันนั้นจะมีความผิดพลาดเกิดขึ้นได้ และในกระบวนการต่อไปนี้จะเป็นกระบวนการทำงานในส่วนของการหาความคล้ายคลึงกันของเนื้อหาสาระหรือส่วนที่ผู้ใช้มีความต้องการในระดับ Shot ที่ซ้ำกัน

2.9 การเปรียบเทียบความคล้ายคลึงกัน (Similarity Matching)

ในปัจจุบันงานเกี่ยวกับด้านกราฟิกหรืองานทางด้านมัลติมีเดียมีบทบาทมากขึ้นในปัจจุบัน ที่สังเกตเห็นได้ง่าย ๆ เช่นงานเกี่ยวกับสื่อโฆษณา งานเกี่ยวกับภาพยนตร์ ฯลฯ ซึ่งถ้าเป็นงานเกี่ยวกับภาพยนตร์แล้วจำนวนของข้อมูลในวิดีโอไฟล์หนึ่งไฟล์มีอยู่อย่างมากมาย ซึ่งสิ่งนี้เองที่จะช่วยเพิ่มประสิทธิภาพการทำงานให้กับงานทางด้านนี้ไม่มากก็น้อย โดยในหัวข้อที่ศึกษานี้จะศึกษาหาความเหมือนหรือคล้ายคลึงกันของวิดีโอไฟล์ในระดับส่วนย่อย ๆ ของวิดีโอ (Shot) เพื่อเปรียบเทียบหา Shot ที่มีลักษณะคล้าย ๆ กันได้ตามที่ผู้เรียกใช้ต้องการ

โดยในการเปรียบเทียบความคล้ายคลึงกันเราได้นำความรู้หรือนำลักษณะเด่นของดรรชนีเฟรมหลัก (Key-Frame) มาใช้เป็นตัวเปรียบเทียบความคล้ายคลึงกันของวิดีโอไฟล์ในระดับ Shot ซึ่งอย่างที่เราศึกษาในส่วนก่อนนี้แล้วว่าเฟรมกับรูปภาพมีแง่มุมในการสังเกตที่เหมือนกัน ซึ่งดรรชนีเฟรมหลักที่นำมาใช้เป็นตัวบ่งบอกวิดีโอไฟล์ในระดับ Shot นี้ก็คือรูปภาพที่แทนลักษณะของเนื้อหาสาระหรืออื่น ๆ ที่เป็นสิ่งแทนวิดีโอไฟล์ในระดับ Shot นี้ ดังนั้นหลักการในการเปรียบเทียบความเหมือนก็สามารถใช้ความรู้เกี่ยวกับการหาส่วนย่อย ๆ ของวิดีโอไฟล์ (Shot Boundary) มาใช้ได้โดยจะแสดงเป็นรูปได้ดังนี้



รูปที่ 2.10 แสดงค่าฮิสโตแกรมที่ได้จากดรรชนีเฟรมหลักแต่ละอัน

โดยในที่นี้เองตัวที่บ่งบอกสิ่งที่ใช้เป็นตัวแทนวิดีโอไฟล์ในระดับ Shot นี้ก็คือค่าฮิสโตแกรมของแต่ละอันจากดรรชนีเฟรมหลักแต่ละอัน ซึ่งถ้าผู้ใช้มีความต้องการหา Shot ที่มีความคล้ายคลึงกันกับดรรชนีเฟรมหลักที่ใช้แสดงแทน Shot นี้ก็เอาค่าฮิสโตแกรม (Histogram Value) มาเป็นตัวเปรียบเทียบแล้วทำการเปรียบเทียบความแตกต่างของค่าฮิสโตแกรมในแต่ละ Key-Frame ได้โดยแสดงสมการการเปรียบเทียบได้ดังนี้

$$
\begin{aligned}
H_q &= [h_{R0}\ h_{R1}\ h_{R2}\ h_{R3} \ldots h_{R255}] \\
\Delta H_{q1} &= |H_q - H_1| \\
\Delta H_{q2} &= |H_q - H_2| \\
\Delta H_{q3} &= |H_q - H_3| \\
&\vdots \\
\Delta H_{qi} &= |H_q - H_i| \qquad (2.3)
\end{aligned}
$$

จะเห็นว่าค่า H_{qi} คือค่าฮิสโตแกรมของดรรชนีเฟรมหลักที่ใช้เป็นตัวเปรียบเทียบความคล้ายคลึงกัน ซึ่งถ้าเกิด Shot ที่ใช้ดรรชนีเฟรมหลัก (Key-Frame) เป็นสิ่งที่ใช้เป็นตัวแทนในวิดีโอไฟล์ระดับ Shot แล้วค่าฮิสโตแกรมของดรรชนีเฟรมหลักต้องมีค่าความแตกต่างกันน้อย

ถ้าค่าความแตกต่างของดรรชนีเฟรมหลักในระดับ Shot มีค่าความแตกต่าง ΔH_{qi} มากแล้วจะทำให้รู้ว่าค่าของ Shot ที่สนใจกับ Shot ที่อยู่ในวิดีโอไฟล์นั้นไม่มีลักษณะความคล้ายคลึงกันหรือความเหมือนกันน้อย แต่ถ้าค่าความแตกต่างของฮิสโตแกรมนั้นมีค่าความแตกต่างกันน้อยก็จะทำให้รู้ว่า Shot ที่สนใจกับ Shot ที่เท่าไรในวิดีโอไฟล์ที่มีความคล้ายคลึงกันของเนื้อหาสาระในแต่ละ Shot โดยในวิธีการทำนั้นจะนำเอาค่าความแตกต่างมาเรียงลำดับจากน้อยไปหามาก โดยค่ามีจะนำมาเป็นคำตอบนั้นจะต้องมีค่าความแตกต่างของฮิสโตแกรมน้อยที่สุด แล้วแต่ว่าจะเลือกกี่ Shot ที่มีลักษณะคล้ายคลึงกันโดยสมมติว่าเรียงลำดับได้ดังนี้

$$\Delta H_{q5} < \Delta H_{q23} < \Delta H_{q1} < \Delta H_{q60} < \Delta H_{q15} < \Delta H_{q8} < \Delta H_{q25} < \Delta H_{q43} < \Delta H_{q55} < \Delta H_{q50} < \Delta H_{q41}$$

ดังนั้นจึงสามารถหาค่าของ Shot ที่มีความคล้ายคลึงกันได้โดยขึ้นอยู่กับผู้ใช้จะต้องการ Shot ที่มีลักษณะความคล้ายคลึงกันเท่าไร เช่นจากตัวอย่างข้างต้นถ้าต้องการจำนวน 3 Shot ก็จะได้ว่าใน Shot5, Shot23 และ Shot1 มีความคล้ายคลึงกันของวิดีโอไฟล์ในระดับ Shot กับ Shot ที่ผู้ใช้ต้องการ

2.10 เทคนิคการป้อนกลับ (Relevance Feedback)

ด้วยความก้าวหน้าของเทคโนโลยีทางคอมพิวเตอร์และการกำเนิดเวิร์ลไวด์เว็บทำให้มีข้อมูลที่เป็นดิจิตอลและมีข้อมูลที่เป็นมัลติมีเดียเป็นจำนวนมาก ซึ่งประกอบไปด้วย Digital Image, Video Audio, Graphics และข้อมูลประเภท Text โดยในการสืบค้นข้อมูลประเภทมัลติมีเดียบนฐานข้อมูลนั้นได้ไม่ใช่เพียงแต่ใช้ข้อความในการค้นหาข้อมูลเท่านั้น แต่มีการใช้วิธีการอื่นในการหา เช่นใช้ภาพต้นแบบเป็นตัวกำหนดในการหาข้อมูล บางครั้งในการประมวลผลที่คอมพิวเตอร์แสดงผลลัพธ์ให้กับผู้ใช้ในการทำงานแต่ละอย่างอาจมีบางส่วนที่ผู้ใช้ไม่ต้องการให้เป็นผลลัพธ์ออกมา ดังนั้นจึงได้มีการเพิ่มส่วนของผู้ใช้งานที่ต้องการในส่วนของ Interactive กับคอมพิวเตอร์เองโดยป้อนข้อมูลกลับเข้าไปใหม่ เพื่อให้คอมพิวเตอร์นั้นประมวลผลได้ดียิ่งขึ้นและมีผลลัพธ์ที่ตรงกับความต้องการของผู้ใช้งานมากขึ้น ซึ่งในโครงงานเกี่ยวกับการค้นหาวิดีโอไฟล์ในระดับ Shot โดยใช้ดรรชนีเฟรมหลักนี้จึงได้นำเอาความรู้เกี่ยวกับเทคนิคการป้อนกลับของข้อมูลเข้ามาร่วมใช้งาน เพื่อให้การค้นหาข้อมูลมีประสิทธิภาพมากยิ่งขึ้นกว่าการใช้งานโดยใช้ดรรชนีเฟรมหลักเพียงอย่างเดียวและให้ผู้ใช้งานแต่ละคนสามารถที่จะเลือกใช้งานตามแนวความคิดของตนเอง โดยแต่ละคนอาจมีความต้องการของผลลัพธ์ที่ต่างกันออกไป โดยเทคนิคการป้อนกลับของข้อมูลนี้จะช่วยให้ผู้ใช้กับคอมพิวเตอร์ได้มีการโต้ตอบกันได้และให้ผลลัพธ์ที่ตรงตามความต้องการของผู้ใช้มากที่สุด

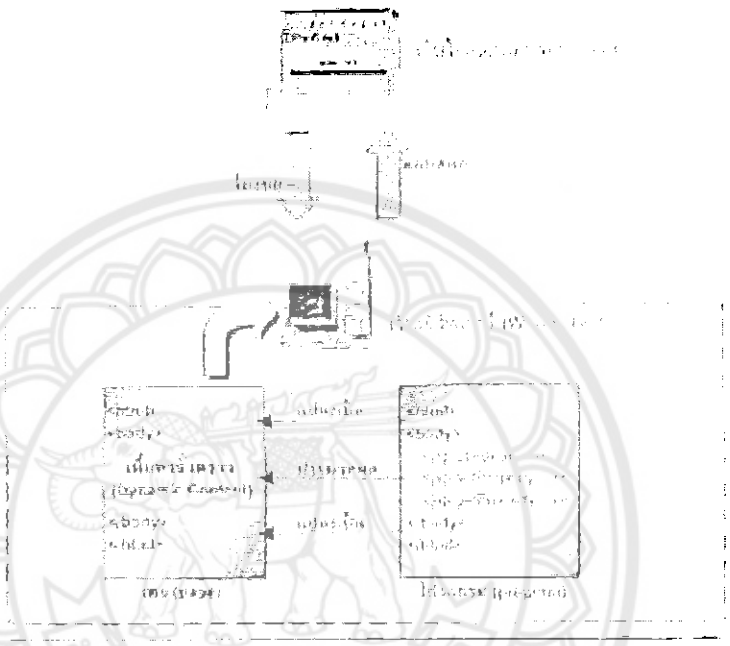
2.11 ทำความรู้จักกับ JSP

JSP (JAVA Server Page) เป็นเทคโนโลยีสำหรับพัฒนาเว็บต่าง ๆ ที่มีเนื้อหาแบบไดนามิค (Dynamic) ซึ่งสามารถเปลี่ยนแปลงได้และโต้ตอบกับผู้ใช้ไม่เหมือนกับเพจของ HTML ตามปกติซึ่งจะบรรจุเนื้อหาคงที่ เพจของ JSP สามารถบรรจุเนื้อหาต่าง ๆ, การจำแนกผู้ใช้, ประเภทบราวเซอร์ของผู้ใช้, สารสนเทศต่าง ๆ เกี่ยวกับผู้ใช้และสิ่งต่าง ๆ เกี่ยวกับเว็บเพจที่ผู้ใช้ชื่นชอบ

เพจของ JSP นั้นจะบรรจุไปด้วยสมาชิกต่าง ๆ ของภาษาที่กำหนดเครื่องหมายมาตรฐาน เช่น แท็กต่าง ๆ ของ HTML ทั่วไป อย่างไรก็ตามเพจของ JSP ที่บรรจุสมาชิกต่าง ๆ ของ JSP ใช้งานได้อย่างกว้างขวาง เช่น การนำสารสนเทศจากฐานข้อมูลออกมาดำเนินการหรือการลงทะเบียนผู้ใช้เมื่อผู้ใช้ถามหาเพจของ JSP แล้วส่งผลกลับไปยังบราวเซอร์ดังรูปที่ 2.11 JSP จะกำหนดสมาชิกมาตรฐานต่าง ๆ ใช้ประโยชน์สำหรับแอพพลิเคชันของเว็บใด ๆ เช่น การเข้าถึงคอมโพเนนท์ต่าง ๆ ของ JAVA Beans, การผ่านคอนโทรลระหว่างเพจต่าง ๆ, และการใช้สารสนเทศร่วมกันระหว่างเพจต่าง ๆ ที่ร้องขอและผู้ใช้ต่าง ๆ โปรแกรมเมอร์สามารถเพิ่มความสามารถให้คำสั่งของ JSP โดยการสนับสนุนสมาชิกต่าง ๆ ที่ระบุแอพพลิเคชันที่ดำเนินงานต่าง ๆ เช่น การเข้าถึงฐานข้อมูลต่าง ๆ และ EJB (Enterprise JAVA Beans), การส่งอีเมลและการสร้างเอกสารของ

HTML เพื่อป้องกันข้อมูลของแอพพลิเคชัน สมาชิกมาตรฐานต่าง ๆ และสมาชิกต่าง ๆ ที่สร้างขึ้นจะช่วยสร้างแอพพลิเคชันของเว็บที่มีประสิทธิภาพสูงมาก

2.11.1 ข้อดีของ JSP มีอะไรบ้าง



รูปที่ 2.11 แสดงการสร้างเนื้อหาแบบไดนามิกด้วยสมาชิกต่าง ๆ ของ JSP

1. JSP นั้นจะทำงานโดยไม่ยึดติดกับแพลตฟอร์มใด ๆ JSP นั้นได้สืบทอดคุณสมบัติเด่นของภาษา JAVA ไว้คือการทำงานโดยไม่ขึ้นกับแพลตฟอร์มหรือระบบปฏิบัติการใด ๆ ไม่ว่าจะเป็นระบบ Windows, Linux, Unix, Mac OS ฯลฯ ดังนั้นเมื่อพัฒนาเว็บด้วย JSP ในแพลตฟอร์มหนึ่งก็สามารถย้ายไปใช้งานกับแพลตฟอร์มอื่น ๆ ได้ไม่ยาก

2. การใช้งาน JAVA API สามารถใช้งานได้หลากหลาย JSP นั้นสามารถเรียกใช้งาน JAVA API ได้หลากหลายมากซึ่ง JAVA API คือกลุ่มของคลาสที่ช่วยอำนวนความสะดวกในการใช้งานต่าง ๆ เช่น การจัดการเกี่ยวกับเน็ตเวิร์ก, การติดต่อฐานข้อมูล, การจัดการทางด้านกราฟิก, การจัดการเกี่ยวกับออบเจ็กต์ต่าง ๆและการรับ-ส่งอีเมลเป็นต้น

3. นำคอมโพแนนต์กลับมาใช้ได้อีก ไม่ต้องเสียเวลาสร้างใหม่ โดยสามารถนำ JAVA Beans มาใช้งานร่วมกับสคริปต์ของ JSP เพราะ JAVA Bean เป็นคอมโพแนนต์ที่เขียนขึ้นมาเพื่อใช้สำหรับทำงานหรือทำหน้าที่อย่างใดอย่างหนึ่งและสามารถนำกลับมาใช้งานได้เสมอ ดังนั้นไม่ต้องเสียเวลาเขียนสคริปต์ JSP เพื่อใช้งานนั้นทุกครั้งจึงพัฒนาเว็บไซต์เสร็จเร็วขึ้น

4. มีความยืดหยุ่นในการใช้งาน ในการเขียนสคริปต์ของ JSP สามารถกำหนดแท็กใหม่ขึ้นมาใช้งานให้เหมาะสมกับความต้องการได้ นอกจากนี้ภาษาที่ใช้เขียนสคริปต์ของ JSP ไม่ได้จำกัดเฉพาะภาษา JAVA เท่านั้น ตามหลักการแล้วสามารถใช้ภาษาอื่น ๆ ในการเขียนสคริปต์ได้ รวมทั้งยังสามารถนำไปใช้งานร่วมกับ XML ได้อย่างมีประสิทธิภาพอีกด้วย

5. มีความปลอดภัยสูง JSP มีระบบการจัดการข้อผิดพลาดต่าง ๆ ไม่ว่าจะเป็นข้อผิดพลาดที่เกิดขึ้นในระหว่างการเขียนสคริปต์หรือข้อผิดพลาดที่เกิดขึ้นเมื่อนำสคริปต์ JSP ไปใช้งานจริง

2.11.2 โครงสร้างและขั้นตอนการทำงานของ JSP

สิ่งที่มีบทบาทสำคัญในการทำงานของ JSP ได้แก่ JSP Container (หรือเรียกอีกอย่างว่า JSP Engine) ซึ่งเป็นส่วนประกอบสำคัญที่อยู่ในเว็บเซิร์ฟเวอร์ เพราะทำหน้าที่ควบคุมและประมวลผลไฟล์ JSP ที่มีการร้องขอ (Request) เข้ามาและตอบสนอง (Response) คำร้องขอนั้นไปยังไคลเอนต์

2.11.3 ขั้นตอนการประมวลไฟล์ JSP ทั้งหมดแบ่งเป็น 8 ขั้นตอนดังนี้

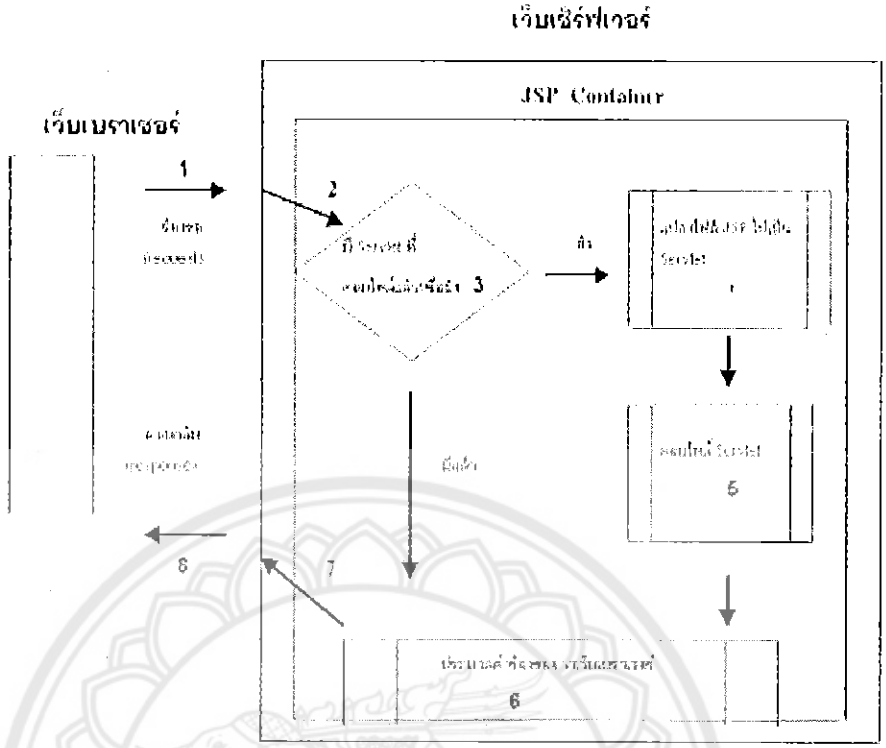
ฝั่งไคลเอนต์ส่งคำร้องขอเอกสาร JSP ไปที่เว็บเซิร์ฟเวอร์ เว็บเซิร์ฟเวอร์ตรวจสอบการร้องขอพบว่าเป็นไฟล์ JSP จึงส่งต่อไปให้ JSP Container จากนั้น JSP Container ตรวจสอบว่าไฟล์ JSP ที่ร้องขอมาเคยแปลงเป็น Servlet และคอมไพล์เป็นไฟล์ .class แล้วหรือยังโดยดูว่ามีไฟล์ .class อยู่หรือเปล่าถ้ายังไม่มีก็จะกระโดดไปทำตามข้อ 4 ต่อแต่ถ้ามีอยู่แล้วก็จะตรวจสอบอีกว่าหลังจากที่แปลงไฟล์ JSP เป็น Servlet และคอมไพล์เป็นไฟล์ .class ครั้งล่าสุดแล้วไฟล์ JSP นั้นมีการเปลี่ยนแปลงแก้ไขหรือเปล่า ถ้ามีการแก้ไขก็จะกระโดดไปทำงานตามขั้นตอนที่ 4 ต่อเช่นกัน แต่ถ้าไม่มีการแก้ไขแสดงว่าไฟล์ JSP นั้นยังคงเดิมไม่เปลี่ยนแปลงจึงไม่มีความจำเป็นที่ต้องเปลี่ยนแปลงเป็น Servlet และคอมไพล์ใหม่ ก็ข้ามไปยังขั้นตอนในส่วนของข้อ 6 ได้เลย

JSP Container แปลงไฟล์ JSP เป็น JAVA Servlet

JSP Container คอมไพล์ไฟล์ JAVA Servlet เป็นไฟล์ .class

JSP Container ประมวลผลตามคำร้องขอนั้น

JSP Container ส่งผลลัพธ์ที่ได้จากการประมวลผลให้แก่เซิร์ฟเวอร์ เว็บเซิร์ฟเวอร์ส่งผลลัพธ์นั้นไปยังไคลเอนต์หรือเว็บเบราเซอร์อีกทอดหนึ่ง

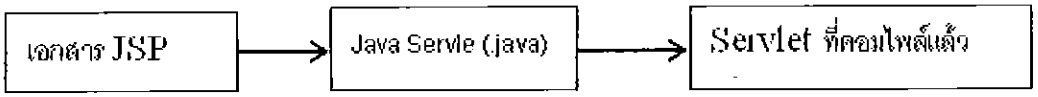


รูปที่ 2.12 แสดงผังการทำงานของ JSP

จากขั้นตอนการประมวลผลไฟล์ JSP ที่แจกแจงข้างต้น สามารถแบ่งออกได้เป็น 2 ช่วงหลัก ๆ คือ ช่วง Translation และช่วง Execution โดยช่วง Translation ได้แก่ขั้นตอนในข้อ 4 และขั้นตอนในข้อ 5 ซึ่งเป็นการแปลงเอกสาร JSP (ไฟล์ .JSP) ให้เป็น Servlet (ไฟล์ .java) จากนั้นก็จะคอมไพล์ไฟล์ Servlet ให้เป็นไฟล์ .class ดังรูปที่ 2.12

ส่วนช่วงของ Execution ได้แก่ขั้นตอนในข้อที่ 6 ซึ่งเป็นการนำเอาไฟล์ .class ที่ได้จากการคอมไพล์มาประมวลผลหรือทำงานตามคำร้องขอจากไคลเอนต์นั่นเอง

โดยปกติแล้วกระบวนการทำงานในช่วง Translation จะกินเวลาพอสมควร แต่โชคดีว่าช่วง Translation จะไม่เกิดขึ้นทุกครั้งที่มีการร้องขอไฟล์ เพราะตราบใดที่ไฟล์ JSP ต้นฉบับไม่มีการเปลี่ยนแปลงอะไร เมื่อมีการร้องขอไฟล์เข้ามาใหม่ ก็ย่อมไม่มีความจำเป็นที่จะแปลงไฟล์เป็น Servlet และคอมไพล์เป็นไฟล์ .class อีก ระบบจะเข้าสู่ช่วง Execution ทันทีโดยใช้ไฟล์ .class ที่มีอยู่แล้ว การทำงานจึงรวดเร็วยิ่งขึ้น แต่ถ้ามีการเปลี่ยนแปลงหรือแก้ไขไฟล์ JSP ใหม่ก็จะต้องเข้าสู่กระบวนการ Translation ใหม่ทุกครั้ง



รูปที่ 2.13 แสดงขั้นตอนการประมวลผลไฟล์ JSP ในช่วง Translation

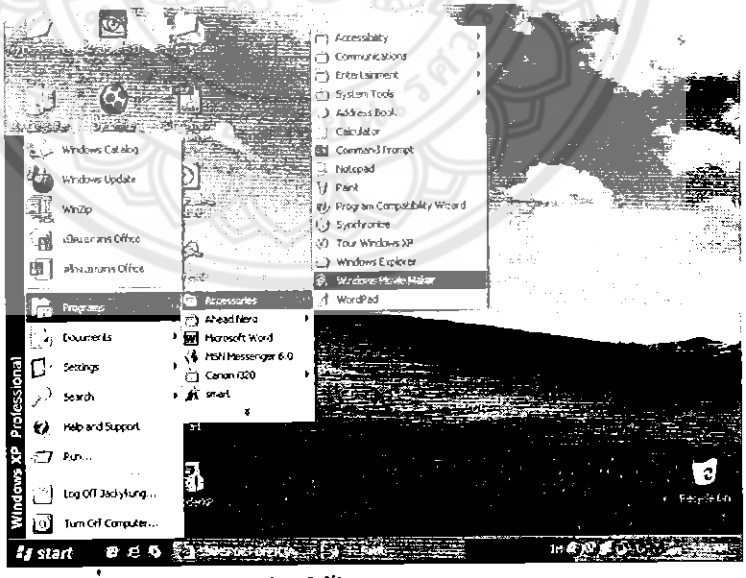
สรุปว่ากระบวนการ Translation มีโอกาสจะเกิดขึ้นได้ 2 กรณี กรณีแรกคือไฟล์ JSP ที่ร้องขอมาเป็นไฟล์ใหม่ที่ยังไม่มีการเปลี่ยนแปลงและคอมไพล์มาก่อน กับอีกกรณีหนึ่งคือ ไฟล์ JSP ที่ร้องขอมาเคยผ่านการแปลงและคอมไพล์มาแล้ว แต่ภายหลังมีการเปลี่ยนแปลงแก้ไขไฟล์ JSP นั้นไปจากเดิม

บทที่ 3
วิธีดำเนินการวิจัย

ในการค้นหาข้อมูลดิจิตอลประเภทวิดีโอไฟล์นี้จะมุ่งสนใจข้อมูลวิดีโอในระดับ Shot ดังนั้นข้อมูลประเภทวิดีโอไฟล์ที่รับเข้ามานี้จะต้องมาผ่านกระบวนการในการแบ่งแยกวิดีโอไฟล์ออกเป็นส่วนย่อย ๆ ซึ่งเรียกกระบวนนี้ว่า Video Segmentation โดยกระบวนการในการแบ่งแยกวิดีโอไฟล์ออกเป็นส่วนย่อย ๆ หรือแบ่งวิดีโอไฟล์ให้เป็นวิดีโอ Shot นี้ได้นำโปรแกรม Windows Movie Maker เข้ามาช่วยเพื่อความสะดวกรวดเร็ว อีกทั้งโปรแกรมนี้ยังมีมาพร้อมกับการติดตั้ง Windows ME และ Windows XP ในลำดับต่อไปจะขออธิบายขั้นตอนในการแบ่งแยกวิดีโอไฟล์ออกเป็นวิดีโอในระดับ Shot

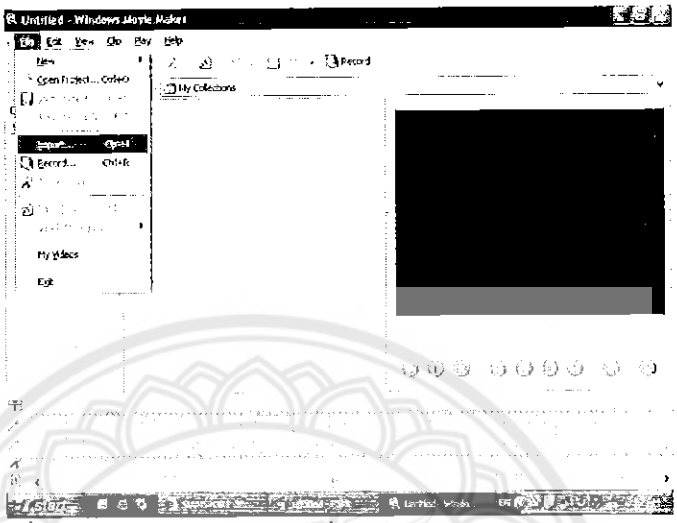
3.1 ขั้นตอนในการแบ่งแยกวิดีโอไฟล์ออกเป็นส่วนย่อย ๆ ของวิดีโอ (Video Segmentation)

เริ่มต้นมาสังเกตหน้าตาของตัวโปรแกรม Windows Movie Maker กันก่อนซึ่งตอนนี้ตัวโปรแกรม Windows Movie Maker มีการพัฒนาถึง Version 2.0 แล้วแต่ในการทำโครงงานนี้จะใช้ Version 1.0 ซึ่งมีมาให้หลังจากติดตั้ง Windows XP แล้ว



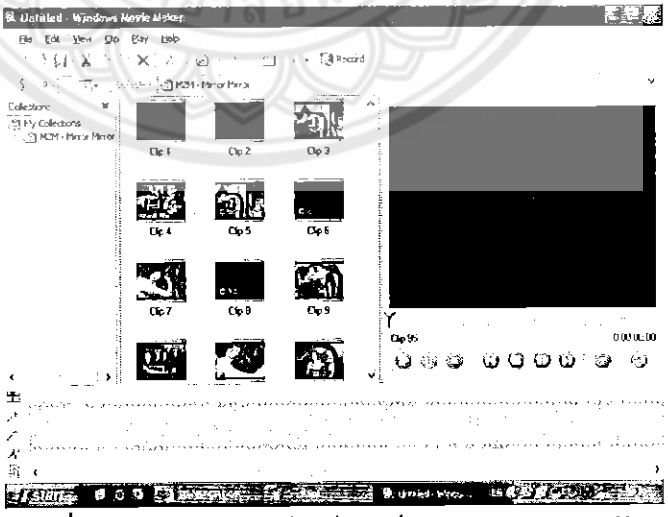
รูปที่ 3.1 แสดงการเรียกใช้โปรแกรม Windows Movie Maker

จากนั้นเมื่อผู้ใช้ต้องการจะนำวิดีโอไฟล์ที่ต้องการมาทำแบ่งออกเป็นวิดีโอ Shot ให้ทำการ Import เข้ามาไว้ใน Windows Movie Maker จากนั้นทำการเลือกไดเรกทอรีที่เก็บวิดีโอไฟล์และเลือกวิดีโอไฟล์ที่ต้องการจะ Import



รูปที่ 3.2 การ Import วิดีโอไฟล์ที่ต้องการจะแบ่งออกเป็นวิดีโอ Shot

เมื่อเลือกวิดีโอไฟล์ที่ต้องการจะ Import ได้แล้วจากนั้นให้คลิ๊กที่ปุ่ม Open จะสังเกตว่าเมื่อกดปุ่ม Open แล้วจะมีแถบเปอร์เซ็นต์และรูปภาพปรากฏขึ้นมา ซึ่งแถบเปอร์เซ็นต์นี้จะเป็นตัวบอกว่าเวลาที่ต้องรอในการแบ่งแยกวิดีโอไฟล์ ส่วนรูปภาพที่ปรากฏขึ้นคือภาพแรกของวิดีโอ Shot ที่โปรแกรมทำการแบ่งวิดีโอไฟล์ให้อัตโนมัติดังรูปที่ 3.3



รูปที่ 3.3 ผลลัพธ์ที่ได้จากการแบ่งวิดีโอไฟล์ออกเป็นส่วนย่อย ๆ ของวิดีโอ (Video Shot)

เมื่อโปรแกรมทำการแบ่งแยกวิดีโอไฟล์ออกเป็นวิดีโอในระดับ Shot แล้วผู้ใช้สามารถเลือกชมวิดีโอใน Shot นั้น ๆ ได้และทำการ Save วิดีโอ Shot ที่ต้องการโดยคลิ๊กรูปภาพแล้วลากไปวางไว้บน Time line ด้านล่างจากนั้นจึงคลิ๊กที่ปุ่ม Save Movie

จากนั้นทำการปรับแต่งความละเอียดของภาพโดยเลือก

Setting: เป็น Other…

Profile: เป็น Video for broadband NTSC (768 Kbps)

จากนั้นคลิ๊กที่ปุ่ม OK จะปรากฏหน้าต่าง Save As เพื่อเลือกไดเรกทอรีที่ต้องการจะเก็บไว้โดยระบุ Save as type: เป็น All files (*.*) และตั้งชื่อไฟล์พร้อมทั้งไฟล์นามสกุลด้วยส่วนของ File name ตัวอย่างเช่น Morning-cilp.mpeg จากนั้นก็จะได้วิดีโอ Shot ที่ผู้ใช้ต้องการซึ่งเมื่อได้ไฟล์วิดีโอที่เป็นส่วนของวิดีโอ Shot นั้นเพียงอย่างเดียวแล้วก็จะเป็นการเสร็จสิ้นการแบ่งแยกวิดีโอออกเป็นส่วนย่อย ๆ ของวิดีโอไฟล์ โดยหลังจากที่ได้วิดีโอ Shot แล้วจะต้องมาดำเนินการหา Key-Frame เพื่อนำไปใช้เป็นตัวบ่งชี้ลักษณะเนื้อหาของวิดีโอ Shot ในส่วนต่อไปจะอธิบายเกี่ยวกับการหาดรรชนีเฟรมหลัก (Key-Frame) ของวิดีโอ Shot นั้น ๆ โดยมีลำดับขั้นตอนดังต่อไปนี้

3.2 ขั้นตอนในการหาดรรชนีเฟรมหลัก (Key-Frame) ของวิดีโอ Shot

ขั้นตอนในการหา Key-Frame นี้ได้ใช้โปรแกรม Matlab 6.5 กับโปรแกรม TMPGEnc เข้ามาช่วยในการหา Key-Frame ซึ่งตัวโปรแกรม Matlab 6.5 จะใช้ในการหา Key-Frame ส่วนโปรแกรม TMPGEnc จะใช้ในการแปลงไฟล์วิดีโอให้เป็นไฟล์นามสกุลต่าง ๆ ซึ่งในที่นี้จะใช้ในการแปลงไฟล์วิดีโอที่เป็น MPEG ให้เป็นไฟล์วิดีโอ AVI โดยในโครงงานนี้เมื่อทำการแบ่งวิดีโอไฟล์ออกเป็นส่วนย่อย ๆ ได้ทำการ Save ไฟล์นามสกุลเป็นแบบ MPEG ซึ่งในการหา Key-Frame โดยใช้โปรแกรม Matlab จะประมวลผลกับวิดีโอไฟล์ที่เป็น AVI เท่านั้นและต้องไม่มีการบีบอัดข้อมูลด้วย (Compression) ดังนั้นจึงต้องใช้โปรแกรม TMPGEnc ในการแปลงไฟล์นามสกุล โดยลำดับต่อไปจะอธิบายขั้นตอนการแปลงไฟล์วิดีโอให้เป็น AVI

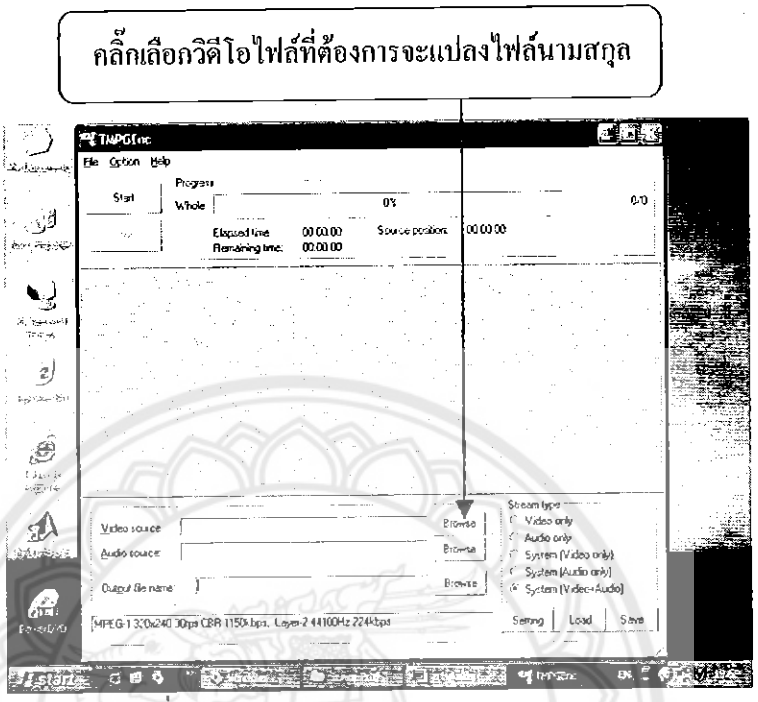
ในการแปลงไฟล์วิดีโอให้เป็นไฟล์นามสกุลตามที่ผู้ใช้ต้องการนั้นก็เหมือนกับโปรแกรมทั่ว ๆ ไปที่ต้องทำการเปิดข้อมูลที่ต้องการ โดยคลิ๊กที่ปุ่ม Browse ของส่วนที่เป็น Video source: ดังรูปที่ 3.4 ซึ่งจะปรากฏหน้าต่าง Open ขึ้น ให้ทำการเลือกไดเรกทอรีและชื่อไฟล์ที่ต้องการ จากนั้นคลิ๊กที่ปุ่ม Open ที่อยู่ทางขวามือดังรูปที่ 3.5 ซึ่งลักษณะของวิดีโอไฟล์ที่ได้ทำการเปิดนั้นจะแสดงภาพแรกของวิดีโอไฟล์นั้น ๆ ออกมาให้เห็น เมื่อเปิดวิดีโอไฟล์ที่ต้องการเรียบร้อยแล้วก็มาถึงขั้นตอนในการเปลี่ยนไฟล์นามสกุลกัน โดยเข้าไปที่เมนู File ซึ่งอยู่ทางมุมซ้ายบนของตัวโปรแกรม จากนั้นเลื่อนมาที่คำสั่ง Output to file เมื่อเลื่อนมาที่คำสั่งนี้แล้วจะปรากฏไฟล์นามสกุลที่เป็นวิดีโอไฟล์ต่าง ๆ ในที่นี้ให้เลือกคำสั่ง AVI file เมื่อคลิ๊กแล้วโปรแกรมจะบอกให้ผู้ใช้เลือกได

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

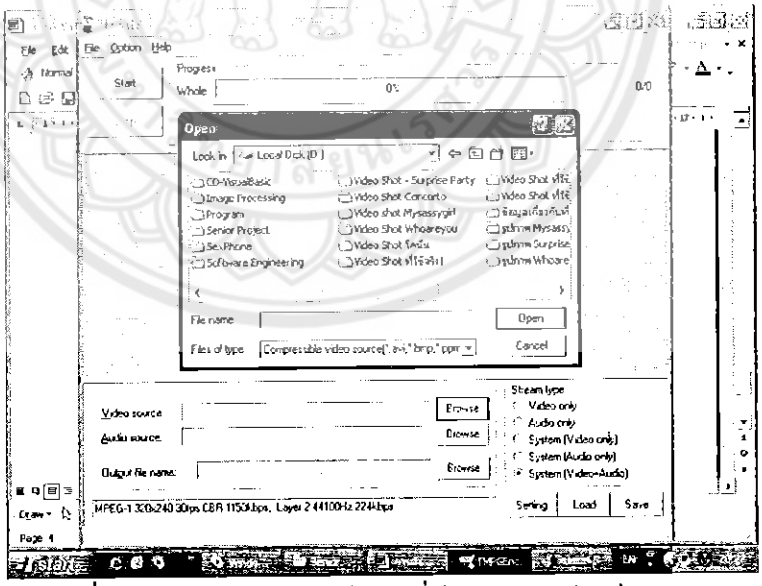
150088540

เรกทอรีที่ต้องการจะเก็บข้อมูล ให้ทำการเลือกแล้วคลิ๊กที่ปุ่ม Save เป็นอันเสร็จสิ้นการเปลี่ยนไฟล์วิดีโอให้เป็นวิดีโอไฟล์ AVI

ปร.
มาแรก
2546



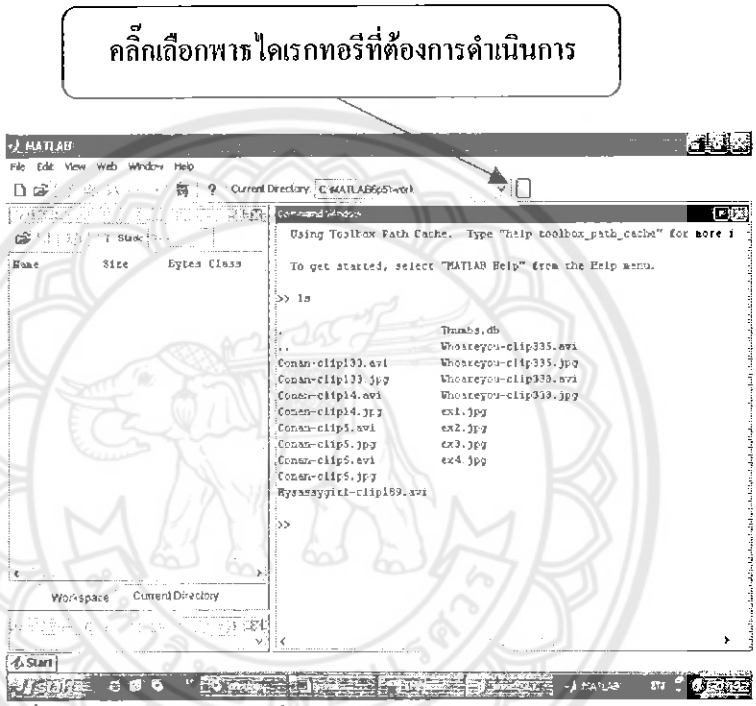
รูปที่ 3.4 แสดงลักษณะของโปรแกรม TMPGEnc



รูปที่ 3.5 แสดงการเลือกไฟล์วิดีโอที่ต้องการแปลงไฟล์นามสกุล

เมื่อเสร็จสิ้นขั้นตอนการแปลงวิดีโอไฟล์ในเป็นแบบ AVI เรียบร้อยแล้วก็มาถึงขั้นตอนการหา Key-Frame ของวิดีโอ Shot กันบ้าง โดยการหาอย่างที่ได้บอกไว้ในตอนต้นได้ใช้โปรแกรม Matlab เวอร์ชั่น 6.5 ในการหา Key-Frame ของวิดีโอ Shot

โดยขั้นแรกในการใช้งานในโปรแกรม Matlab จะต้องระบุพาธไดเรกทอรีที่ต้องการจะดำเนินการ โดยเลือกที่ Current Directory: จากนั้นทำการเรียกดูไฟล์ที่มีอยู่โดยใช้คำสั่ง ls โดยจะปรากฏไฟล์ต่าง ๆ ที่อยู่ในพาธไดเรกทอรีนี้ดังรูปที่ 3.6



รูปที่ 3.6 แสดงไฟล์ต่าง ๆ ที่มีอยู่ในพาธไดเรกทอรี C:\MATLAB6p5\work

จากนั้นจึงเริ่มดำเนินการ Key-Frame โดยมีลำดับคำสั่งในการทำงานดังนี้

3.2.1 aviinfo(' ชื่อไฟล์')

ใช้เพื่อดูข้อมูลเกี่ยวกับวิดีโอไฟล์นี้ตัวอย่างเช่น aviinfo('Whoareyou-clip335.avi') ในที่นี้ให้สังเกตที่ NumFrames: ซึ่งจะบอกจำนวนเฟรมทั้งหมดในวิดีโอ Shot นี้ ซึ่งจากตัวอย่างจะได้จำนวนเฟรมทั้งหมดเท่ากับ 730 ดังนั้น Key-Frame จึงอยู่ที่เฟรมที่ 365

3.2.2 mov = aviread(‘ ชื่อไฟล์ ‘, ลำดับที่ของเฟรม);

ใช้เพื่ออ่านข้อมูลประเภทวิดีโอไฟล์ที่เป็น AVI เข้ามาเก็บไว้ในตัวแปร mov ซึ่งลำดับที่ของเฟรมนี้คือลำดับที่ของเฟรมที่เป็น Key-Frame ของวิดีโอ Shot ตัวอย่างเช่น mov = aviread(‘Whoareyou-clip335.avi’, 365); โดยจะได้ผลลัพธ์เป็นภาพของวิดีโอ ณ เฟรมที่ 365

3.2.3 movie(mov);

ใช้ในการเรียกดูภาพในเฟรมนั้น ๆ หรือดูวิดีโอไฟล์

3.2.4 img = mov.cdata;

ใช้ในการเปลี่ยนภาพในเฟรมนั้น ๆ ให้กลายเป็นรูปภาพโดย img คือตัวแปรที่ใช้เก็บรูปภาพ

3.2.5 imwrite(img, ‘ ตั้งชื่อไฟล์ ‘);

ใช้ในการเขียนข้อมูลที่เป็นรูปภาพพร้อมทั้งตั้งชื่อให้กับไฟล์รูปภาพนั้นด้วย ตัวอย่างเช่น Imwrite(img, ‘Whoareyou-clip335.jpg’);

เมื่อทำการเขียนข้อมูลที่เป็นรูปภาพแล้ว ไฟล์รูปภาพนี้จะถูกเก็บอยู่ในพาธไดเรกทอรีที่ได้เลือกดำเนินการในตอนต้น เมื่อเสร็จสิ้นขั้นตอนการหา Key-Frame ดังกล่าวก็จะได้ Key-Frame ที่เป็นรูปภาพเพื่อนำไปดำเนินการในการค้นหาวิดีโอไฟล์ในระดับ Shot ต่อไป

```
>> mov = aviread('Whoareyou-clip335.avi', 365);
>> movie(mov);
>> img = mov.cdata;
>> imwrite(img, 'Whoareyou-clip335.jpg');
>> ls

.                       Thumbs.db
..                      Whoareyou-clip335.avi
Conan-clip133.avi       Whoareyou-clip335.jpg
Conan-clip133.jpg       Whoareyou-clip338.avi
Conan-clip14.avi        Whoareyou-clip338.jpg
Conan-clip14.jpg        ex1.jpg
Conan-clip5.avi         ex2.jpg
Conan-clip5.jpg         ex3.jpg
Conan-clip6.avi         ex4.jpg
Conan-clip6.jpg
Mysassygirl-clip189.avi

>>
```

รูปที่ 3.7 แสดงลำดับขั้นตอนในการใช้คำสั่งเพื่อหา Key-Frame

เมื่อได้ทำการ Shot Segmentation ตัดวิดีโอไฟล์ที่เป็นข้อมูลดิบออกเป็นส่วนย่อย ๆ ในระดับ Shot ของวิดีโอไฟล์แล้ว ก็จะมาเริ่มในส่วนของการนำ Shot ของวิดีโอไฟล์ไปประมวลผล ซึ่งจากที่ได้ทำการกล่าวไว้ข้างต้นแล้วในหลักการและทฤษฎีที่เกี่ยวข้อง จะขอนำมายกตัวอย่างประกอบให้เข้าใจในกระบวนการทำงานของวิดีโอไฟล์ในระดับ Shot อีกครั้งหนึ่ง

ซึ่งโดยส่วนใหญ่แล้ว Shot ของวิดีโอไฟล์นั้นจะประกอบไปด้วยเฟรมที่ถูกนำมาเรียงต่อกันในช่วงเวลาหนึ่ง และเฟรมที่เรียงต่อกันนี้เองเราจะทำการดึง Key-Frame เพื่อใช้เป็นตัวบ่งบอกถึงลักษณะของ Shot ๆ นี้ เมื่อได้ Key-Frame แล้วก็มาถึงขั้นตอนของการนำ Key-Frame ไปใช้งานในการประมวลผลที่แท้จริง ซึ่งขั้นตอนข้างต้นที่ได้กล่าวมานี้เป็นเพียงส่วนของการดำเนินงานในขั้นต้นก่อนจะทำการประมวลผล (Preprocessing) โดยจะข้อกล่าวเป็นข้อ ๆ เกี่ยวกับลำดับการทำงานของ Key-Frame ดังนี้

3.3 การนำ Key-Frame มาทำการหาค่าสี (Color Model)

โดย Key-Frame ที่ได้มานี้สามารถเปรียบได้กับรูปภาพ ๆ หนึ่ง ดังนั้นการทำงานด้านการประมวลผลเกี่ยวกับรูปภาพจึงได้มีแนวคิดว่าสิ่งใดที่จะเป็นตัวบ่งบอกถึงค่าที่เป็นเอกลักษณะเฉพาะของรูปภาพ ๆ นั้นขึ้นมา จึงได้มีการนำหลักการและทฤษฎีเกี่ยวกับ Color Model หรือค่าสีของภาพ ซึ่งได้นำหลักการ Color Model ของค่าสีที่เรียกว่า RGB และ HSV มาใช้บ่งบอกถึงลักษณะของภาพ โดยกระทำการหาค่าสี RGB ของรูปภาพที่เป็นตัว Key-Frame ของ Shot วิดีโอนั้นออกมาก่อน ซึ่งในรูปภาพ ๆ หนึ่งจะประกอบไปด้วยพิกเซล (Pixel) หลาย ๆ พิกเซลนำมาเรียงต่อกันตัวอย่างเช่น ถ้าเราได้นำรูปภาพ ๆ หนึ่งที่มีความกว้าง 800 และความสูง 600 พิกเซลเราก็จะได้ภาพที่มีจำนวนพิกเซลทั้งหมดเท่ากับ 800 X 600 (480,000 พิกเซล) ซึ่งในแต่ละพิกเซลนี้จะประกอบด้วยค่าสีที่เกิดจากการรวมค่าสีหลักสามสีคือ Red (สีแดง) Green (สีเขียว) และ Blue (สีน้ำเงิน) โดยเรียกย่อ ๆ ว่าค่าสี RGB ซึ่งสีของภาพที่มองเห็นก็เกิดจากการรวมค่าสีหลักทั้งสามนี้ จากนั้นเมื่อเราได้ค่าสี RGB ในแต่ละพิกเซลแล้วก็มาถึงขั้นตอนการเปลี่ยนค่าสีจาก RGB ไปเป็น HSV เนื่องจากค่าสี HSV นั้นมีความละเอียดที่ใกล้เคียงกับระดับสายตาของมนุษย์มากที่สุด พอถึงขั้นตอนนี้แล้วต่อไปจะเป็นขั้นตอนของการจัดเก็บค่าสี HSV ในแต่ละพิกเซลเหล่านี้ให้อยู่ในรูปของเวกเตอร์เพื่อให้ง่ายต่อการนำไปใช้งานในขั้นตอนต่อไป ซึ่งก่อนหน้าที่จะเก็บค่าสีให้อยู่ในรูปเวกเตอร์นี้ จะต้องทำการเก็บค่าสี HSV เหล่านี้ในรูปของ Histogram เสียก่อน

โดยในขั้นตอนต่อไปจะเป็นการอธิบายขั้นตอนการทำงานในส่วนของการจัดเก็บค่าสี HSV ในรูปของ Color Histogram

3.4 การนำ Key-Frame มาทำ Color Histogram

จากขั้นตอนการหาค่า HSV ของ Key-Frame ที่เป็นรูปภาพซึ่งเราได้ค่าสี HSV ในแต่ละพิกเซลมาแล้วต่อไปเราจะมาทำการนับค่าสีที่ซ้ำกันในแต่ละพิกเซลเก็บอยู่ในรูปของฮิสโตแกรม (Histogram) เพื่อให้เข้าใจง่ายขึ้นและมองให้เห็นถึงลักษณะของรูปภาพที่เป็น Key-Frame มากขึ้น โดยในการทำส่วนนี้นั้นจะทำการนับค่าสีในแต่ละพิกเซลโดยจะแสดงลำดับการหาค่าฮิสโตแกรมดังนี้ อันดับแรกจะทำการอ่านค่าสี HSV ในแต่ละพิกเซลมาเก็บไว้โดยตั้งช่วงค่าสี HSV ไว้ดงที่ ซึ่งถ้าค่าสีในแต่ละพิกเซลอยู่ในช่วงไหนก็ให้เพิ่มค่าความถี่ในช่วงนั้นเป็น 1 โดยทำอย่างนี้ทุกพิกเซล ซึ่งค่าสีของ HSV ที่ได้ในแต่ละพิกเซลนั้นจะอยู่ในช่วง 0 – 1 เท่านั้น โดยในที่นี้จะทำการแบ่งค่า Hue ออกเป็น 8 ส่วน ค่า Saturation เป็น 4 ส่วน และ ค่า Value ออกเป็น 2 ส่วนดังนี้

ส่วนของค่า Hue (ค่าสีในแถบสีรุ้ง) โดยเริ่มจากสีแดงเป็นค่าใกล้ 0 ไปจนถึงสีม่วงเป็นค่าใกล้ 1

1. ช่วงค่า Hue ตั้งแต่ 0 ถึง 0.125
2. ช่วงค่า Hue ตั้งแต่ 0.125 ถึง 0.25
3. ช่วงค่า Hue ตั้งแต่ 0.25 ถึง 0.375
4. ช่วงค่า Hue ตั้งแต่ 0.375 ถึง 0.5
5. ช่วงค่า Hue ตั้งแต่ 0.5 ถึง 0.625
6. ช่วงค่า Hue ตั้งแต่ 0.625 ถึง 0.75
7. ช่วงค่า Hue ตั้งแต่ 0.75 ถึง 0.875
8. ช่วงค่า Hue ตั้งแต่ 0.875 ถึง 1

ส่วนของค่า Saturation (ค่าความบริสุทธิ์ของสี) เริ่มจากค่าน้อยคือ Unsaturation บอกความเป็นสีขาวไปถึงค่ามากคือ Saturation หมายถึงสีที่มีค่า Hue เท่านั้น

1. ช่วงค่า Saturation ตั้งแต่ 0 ถึง 0.25
2. ช่วงค่า Saturation ตั้งแต่ 0.25 ถึง 0.5
3. ช่วงค่า Saturation ตั้งแต่ 0.5 ถึง 0.75
4. ช่วงค่า Saturation ตั้งแต่ 0.75 ถึง 1

ส่วนของค่า Value (ค่าความสว่างของสี)

1. ช่วงค่า Value ตั้งแต่ 0 ถึง 0.5
2. ช่วงค่า Value ตั้งแต่ 0.5 ถึง 1

ซึ่งจะแบ่งรูปแบบของค่าสี HSV ได้เป็น 64 ส่วน (Hue(8) x Saturation(4) x Value(2)) ซึ่งเมื่ออ่านค่าในแต่ละพิกเซลมาเก็บแล้วจะได้จำนวนความถี่ในแต่ละส่วน โดยในแต่ละส่วนจะมีค่าเหมือนกันหรือแตกต่างกันมากเท่าไรก็อยู่ที่ค่าสี HSV ที่อ่านได้ในแต่ละพิกเซลว่าจะมีค่าอยู่ในช่วงใดมากช่วงใดน้อย ซึ่งเมื่อได้ค่าแล้วจะต้องมาทำการ Normalize ค่า Histogram เพราะในแต่ละ Key-Frame อาจจะมีจำนวนพิกเซลไม่เท่ากันดังนั้นเมื่อได้จำนวนความถี่ในแต่ส่วน จึงควรที่จะต้อง

นำมาหารด้วยจำนวนพิกเซลทั้งหมดเพื่อเฉลี่ยให้แต่ละ Key-Frame มีค่าไม่แตกต่างกันมากนัก เมื่อทำการ Normalize แล้วก็จะสามารถนำค่าที่ได้ในแต่ละส่วนไปพล็อตเป็นกราฟเพื่อแสดงให้เห็นถึงลักษณะของภาพ โดยแสดงค่าของ Histogram ของ Key-Frame ได้ดังนี้

Histogram = [Hsv1, Hsv2, Hsv3, Hsv4, … , Hsv63, Hsv64]

และจากค่าที่ได้นี้ก็สามารถนำไปเก็บเป็น Vector เพื่อนำไปใช้ในการประมวลผลในเรื่องของการหาค่าความแตกต่างของ Key-Frame

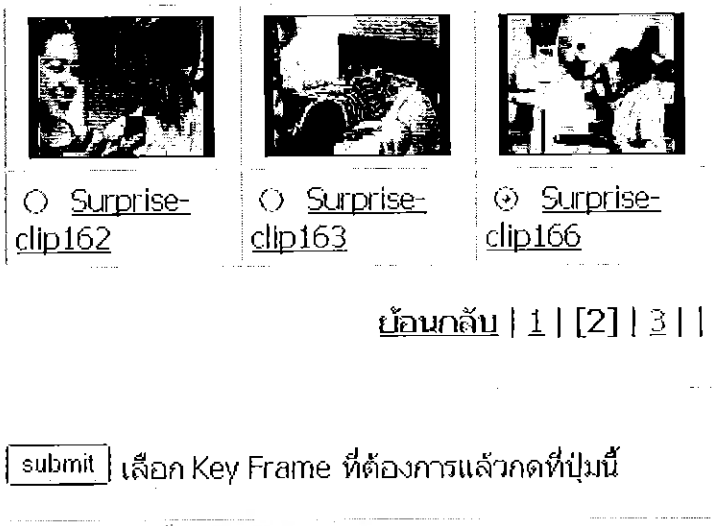
3.5 การค้นหาวิดีโอ Shot โดยใช้ค่าความแตกต่างระหว่างรูปภาพของ Key-Frame

ในการหาค่าความแตกต่างของรูปภาพที่เป็น Key-Frame นั้นจะต้องมี Key-Frame ที่เป็นตัวที่ใช้ในการค้นหาก่อน ซึ่งขั้นตอนนี้จะแสดงรูปภาพที่เป็น Key-Frame ทั้งหมดจากฐานข้อมูลที่เรามีอยู่ออกมาแสดง โดยแสดงหน้าละ 24 รูป

รูปที่ 3.8 แสดงตัวอย่างของการเลือกภาพที่จะใช้ในการค้นหา

ซึ่งในขั้นตอนนี้การเลือกภาพครั้งแรกจะระบุให้เลือกเพียงแค่ Key-Frame เดียวเท่านั้นซึ่งในขั้นตอนการเลือกจะเรียกว่า ขั้นตอนการค้นหาวิดีโอไฟล์ในระดับ Shot โดยใช้ดรรชนีเฟรมหลัก (Video Retrieval Using Key-Frame Video Indexing) ซึ่งจากรูปที่ 3.8 จะเห็นว่าส่วนของรูปภาพที่นำมาแสดงในตารางนี้ก็คือ Key-Frame และในขั้นตอนนี้สามารถเปิดดูวิดีโอใน Shot ๆ นี้ได้โดยคลิ๊กที่ชื่อไฟล์ ผลคือจะทำให้ผู้ใช้ระบุความต้องการในการค้นหา Shot ของวิดีโอไฟล์นี้ได้ชัดเจนมากขึ้น

จากนั้นเมื่อผู้ใช้ได้ทำการเลือก Key-Frame ที่ต้องการแล้วให้กดคลิ๊กที่ปุ่มเพื่อนำ Key-Frame ที่ต้องการนี้ไปประมวลผลในขั้นต่อไปเพื่อหา Key-Frame ที่ใกล้เคียงกันขึ้นมา



รูปที่ 3.9 แสดงขั้นตอนหลังจากที่เลือก Key-Frame ที่ต้องการได้แล้ว

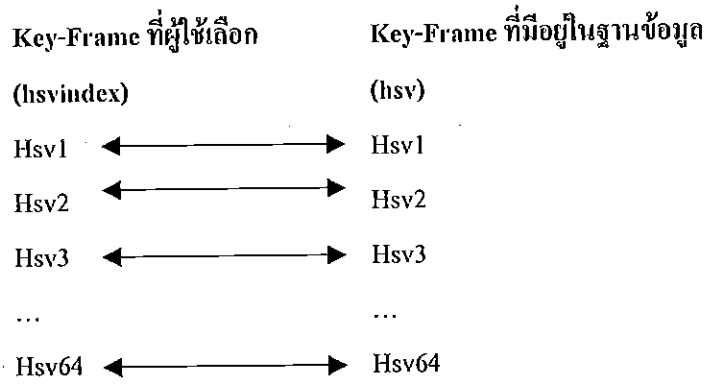
จากรูปที่ 3.9 นี้เป็นการเลือก Key-Frame ของวิดีโอ Shot ที่ชื่อว่า Surprise-clip166 เพื่อใช้ในการค้นหาวิดีโอ Shot ที่ใกล้เคียงกับวิดีโอ Shot นี้ พอเราป้อนส่วนของอินพุตให้กับโปรแกรมแล้วก็จะมาถึงกระบวนการในการค้นหาวิดีโอ Shot ที่ใกล้เคียงกับ Key-Frame ตัวอย่าง

ซึ่งจาก Key-Frame ที่ได้ทำการเลือกจากรูปที่ 3.9 นี้ โปรแกรมจะทำการดึงค่าข้อมูลที่เป็นค่าฮิสโตแกรมของ Key-Frame นี้ขึ้นมา โดยได้ทำการเก็บข้อมูลที่เป็นค่าฮิสโตแกรมนี้ไว้ในฐานข้อมูลอยู่แล้ว ค่าฮิสโตแกรมที่โปรแกรมได้ทำการดึงข้อมูลออกมานี้จะอยู่ในรูปของ Vector เพื่อนำมาใช้ในการเปรียบเทียบความแตกต่างระหว่างรูปภาพที่เป็น Key-Frame

จากนั้นก็จะทำการเปรียบเทียบค่าฮิสโตแกรมของ Key-Frame ที่ผู้ใช้ได้ทำการเลือกกับค่าฮิสโตแกรมของฐานข้อมูลทุกตัว (รวมทั้ง Key-Frame ที่ผู้ใช้ได้ทำการเลือกด้วย) นำมาหาค่า Distance ซึ่งเมื่อได้ทำการเปรียบเทียบกับ Key-Frame ในฐานข้อมูลแล้วก็จะได้ค่าความแตกต่างระหว่าง Key-Frame ที่ผู้ใช้ได้ทำการเลือกกับ Key-Frame ในฐานข้อมูลทุกตัว โดยจะขอยกตัวอย่างให้เข้าใจง่ายขึ้นดังนี้

สมมติฐานข้อมูลของเราทั้งหมดมี 10 Key-Frame เมื่อเราทำการเปรียบเทียบค่าความแตกต่างแล้ว ผลที่ได้คือเราจะได้ค่า Distance ทั้งหมด 10 ค่า ซึ่งจาก 10 ค่านี้ก็คือค่าที่เปรียบเทียบกันระหว่าง Key-Frame ที่ผู้ใช้เลือกกับ Key-Frame ในฐานข้อมูลทั้งหมดนั่นเอง

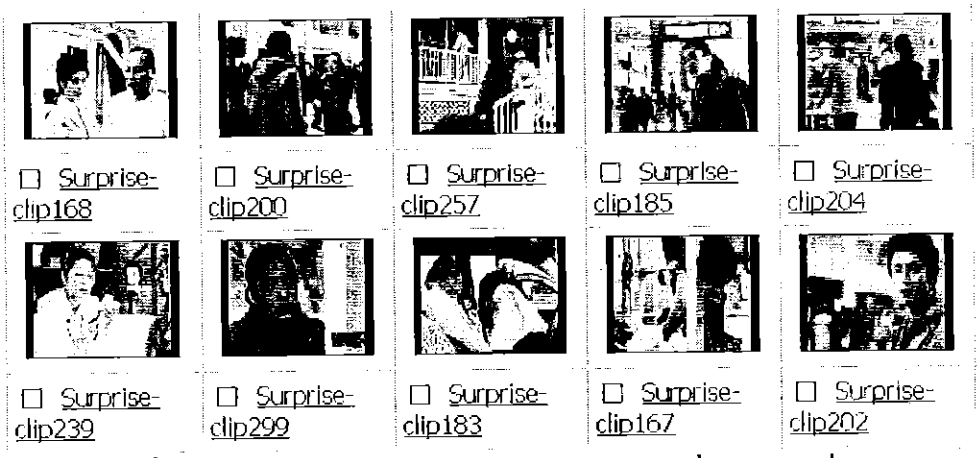
ในการเปรียบเทียบค่าฮิสโตแกรมก็จะทำการเปรียบเทียบทีละส่วนย่อยของค่าสี (ซึ่งได้ทำการแบ่งไว้เป็น 64 ค่าฮิสโตแกรม bin)



$$\text{Distance} = \sqrt{\sum_{i=1}^{64} \left(\text{hsvindex}_i - \text{hsv}_i\right)^2} \qquad (3.1)$$

จากสมการนี้จะได้ผลรวมของค่าความแตกต่างของฮิสโตแกรมในแต่ละ bin (ส่วนย่อยของค่าสี) สังเกตจากสมการนี้ทำไมถึงต้องทำการยกกำลังสองของผลต่างระหว่าง hsvindex_i-hsv_i เพราะเนื่องจากค่าฮิสโตแกรมในแต่ละ bin ของแต่ละ Key-Frame นั้นอาจมีค่าไม่ตรงกันอาจจะมีค่ามากบ้างน้อยบ้าง ดังนั้นค่าที่ออกมาจากผลต่างนี้อาจติดค่าลบได้จึงต้องทำการยกกำลังสองทุกครั้ง และเมื่อได้ผลรวมของค่าความแตกต่างในแต่ละ bin แล้วจากนั้นจึงนำมาถอด Square Root เพื่อกลับไปยังค่าความแตกต่างเดิม

เมื่อกระทำการเปรียบเทียบกับทุก Key-Frame ในฐานข้อมูลแล้วก็จะได้ค่า Distance ตามจำนวน Key-Frame ในฐานข้อมูล ซึ่งจะมีค่าเฉพาะเป็นของตนเอง ซึ่งถ้าผลลัพธ์ของค่าความแตกต่างที่ได้เท่ากับศูนย์ แสดงว่าเป็น Key-Frame อย่างเดียวกับ Key-Frame ที่ผู้ใช้เลือก ซึ่งจากหลักการและทฤษฎีถ้าค่าความแตกต่างมีค่าน้อยแสดงว่าเป็น Key-Frame ที่มีความเหมือนกันมากซึ่งเมื่อทำขั้นตอนการหาค่าความแตกต่างเสร็จสิ้นแล้ว จะต้องมาทำการเรียงลำดับค่าความแตกต่างของ Key-Frame ในฐานข้อมูลทุกตัวจากน้อยไปหามากเพื่อดูว่า Key-Frame อันไหนที่ใกล้เคียงกับที่ผู้ใช้เลือกมากที่สุด ซึ่งแน่นอนว่าค่าแรกที่เรียงลำดับออกมาก็คือค่า Distance ของ Key-Frame ที่ผู้ใช้เลือกนั่นเองมีค่า Distance เท่ากับศูนย์ โดยการเรียงลำดับค่า Distance นี้เราได้ใช้วิธีการ Bubble Sort ในการเรียงลำดับข้อมูลจากน้อยไปหามาก ซึ่งจากรูปที่ 3.9 แสดงผลลัพธ์ของการค้นหาได้ดังรูปที่ 3.10



รูปที่ 3.10 แสดงผลลัพธ์ของการค้นหาจาก Key-Frame ที่เลือกในรูปที่ 3.2

ผลลัพธ์ที่ได้จากการค้นหานั่นอาจจะมีบางส่วนที่ไม่ตรงตามความต้องการของผู้ใช้ จึงได้ทำการเพิ่มประสิทธิภาพในการค้นหาขึ้นโดยได้นำหลักการและทฤษฎีเกี่ยวกับเทคนิคการป้อนกลับ (Relevance Fedback) มาช่วยในหัวข้อต่อไปนี้จะกล่าวถึงขั้นตอนในการค้นหาโดยใช้หลักการและทฤษฎีเกี่ยวกับ Relevance Feedback

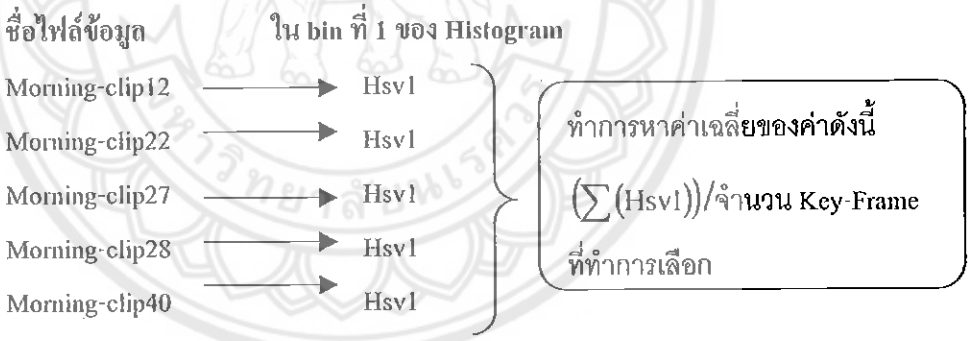
3.6 การเพิ่มประสิทธิภาพในการค้นหาวิดีโอ Shot โดยใช้ Relevance Feedback

การใช้หลักการและทฤษฎีเกี่ยวกับเทคนิคการป้อนกลับเข้ามาช่วยเพิ่มประสิทธิภาพในการค้นหานี้ เพราะต้องการให้ผู้ใช้ได้มีส่วนร่วมในการเลือก Key-Frame ที่ต้องการได้มากขึ้น โดยทำการเลือก Key-Frame ที่ผู้ใช้คิดว่า Key-Frame นี้น่าจะใกล้เคียงกับ Key-Frame ตั้งต้นที่ผู้ใช้ได้ทำการเลือก จากนั้นจึงทำการหาค่าเบี่ยงเบนมาตรฐานของกลุ่ม Key-Frame ที่ผู้ใช้ได้ทำการเลือก

ซึ่งขั้นตอนในการหาค่าเบี่ยงเบนมาตรฐานนี้ก็เพื่อจะหาค่าที่เป็นค่าใกล้เคียงกับค่าที่เปลี่ยนแปลงไปของกลุ่มตัวอย่าง โดยจะทำการหาค่าเบี่ยงเบนมาตรฐานในแต่ละ bin ของฮิสโตแกรม ในการกระทำแบบนี้เองจะเรียกว่าการหาค่า Weight ของกลุ่ม Key-Frame ซึ่งการหาค่า Weight ของกลุ่มที่ได้ทำการเลือกนี้จะถูกนำไปใช้ในการประมวลผลต่อไปเพื่อเพิ่มประสิทธิภาพในการค้นหายิ่งขึ้น

รูปที่ 3.11 แสดงขั้นตอนในการเลือก Key-Frame เพื่อนำไปทำ Relevance Feedback

วิธีการหาค่า Weight ของกลุ่ม Key-Frame ที่ผู้ใช้เลือกมีวิธีการดังนี้ โดยเมื่อรับกลุ่ม Key-Frame ที่ผู้ใช้ได้เลือกแล้ว จากนั้นโปรแกรมจะทำการดึงข้อมูลที่เป็นฮิสโตแกรมของรูปภาพที่เป็น Key-Frame ในแต่ละอัน จากนั้นจะมาทำการหาค่าเบี่ยงเบนมาตราฐานในแต่ละ bin ของฮิสโตแกรมโดยจะขอยกตัวอย่างเพื่อให้เข้าใจง่ายขึ้น



โดยข้างบนนี้จะเป็นการหาค่าเฉลี่ยของค่าฮิสโตแกรมใน bin ที่ 1 ของ Key-Frame แต่ละอันที่ผู้ใช้ได้ทำการเลือก จากนั้นก็มาคำนวณหาค่าเบี่ยงเบนมาตราฐานใน bin ที่ 1 ซึ่งในที่นี้จะสมมติค่าเฉลี่ยของ Hsv1 เก็บอยู่ในตัวแปรชื่อ avgHsv(average Histogram) และจำนวน Key-Frame ที่ผู้ใช้ได้ทำการเลือกเก็บอยู่ในตัวแปรชื่อ N

$$\text{weight}_i = \frac{\sqrt{\sum(\text{Hsv}_i - \text{avgHsv})^2}}{N} \qquad (3.2)$$

ซึ่งค่า Hsv_i นี้ก็คือค่าฮิสโตแกรมใน bin ที่ i ของ Key-Frame แต่ละอันที่ผู้ใช้ได้ทำการเลือก เมื่อหาค่าเบี่ยงเบนมาตรฐานใน bin ที่ i ของฮิสโตแกรมได้แล้ว ต่อไปก็จะทำการหาค่าเบี่ยงเบนมาตรฐานอย่างนี้กับทุก bin ในฮิสโตแกรมโดยในโปรแกรมนี้จะเริ่มหาค่าเบี่ยงเบนมาตรฐานใน bin ที่ i ของฮิสโตแกรมก่อนแล้วทำไปจนถึง bin ที่ 64 ของฮิสโตแกรม (เพราะในที่นี้เราได้ทำการแบ่งค่าสีไว้ 64 ค่า) ซึ่งในการหาค่าเบี่ยงเบนมาตรฐานนี้จะทำการ Normalize ค่าโดยนำค่าเบี่ยงเบนมาตรฐานที่ได้ในแต่ละ bin ของฮิสโตแกรมมาหาร 1 (1 / ค่าเบี่ยงเบนมาตรฐาน) เพื่อให้ค่าที่ถูกต้องและเป็นมาตรฐานมากขึ้น

ซึ่งจะพบว่ามีบาง bin ที่มีค่าเบี่ยงเบนมาตรฐานมีค่าเป็นศูนย์ซึ่งหมายความว่าค่าฮิสโตแกรมในช่วงค่าสีนี้ไม่มีผลกระทบต่อการเปลี่ยนแปลงของกลุ่ม Key-Frame ซึ่งเมื่อนำมาหารแล้วจะได้ค่าเป็น Infinity ดังนั้นจึงปรับค่าเบี่ยงเบนมาตรฐานใน bin นั้นให้มีค่าเป็นศูนย์ ซึ่งเมื่อได้ทำการหาค่าเบี่ยงเบนมาตรฐานในแต่ละ bin ของฮิสโตแกรมเรียบร้อยแล้วก็จะได้ค่า Weight เพื่อนำไปหาค่าความแตกต่างของ Key-Frame อีก

ในการหาค่าความแตกต่างนี้ (Distance) จะได้นำค่า Weight นี้มาร่วมใช้ด้วยในการหาค่าความแตกต่างแสดงเป็นสมการได้ดังนี้

$$\text{Weight Distance} = \sqrt{\sum_{i=1}^{64}\left(\text{weight}_i \times \left(\text{hsvindex}_i - \text{hsv}_i\right)^2\right)} \qquad (3.3)$$

จากสมการข้างต้นค่า hsvindex ก็คือค่าฮิสโตแกรมของ Key-Frame ที่ผู้ใช้ได้ทำการเลือกไว้ในตอนแรกเพียงภาพเดียว ส่วนค่า hsv ก็คือค่าฮิสโตแกรมของ Key-Frame แต่ละอันในฐานข้อมูลทั้งหมด ส่วนค่า i คือค่าที่กำกับค่า bin ในฮิสโตแกรม เช่น $hsvindex_i$ ก็คือค่าฮิสโตแกรมของ Key-Frame ตั้งต้นใน bin ที่ i ซึ่งขั้นตอนนี้จะคล้าย ๆ กับการหาค่า Distance แต่ได้ทำการเพิ่มค่าของ Weight Distance เข้ามาเพื่อค้นหาภาพใหม่อีกครั้ง โดยพยายามให้ผลลัพธ์จากการค้นหาวิดีโอ Shot โดยใช้ Relevance Feedback มีความถูกต้องมากขึ้น จากนั้นเมื่อได้ค่าของ Weight Distance ของ Key-Frame ในแต่ละอันแล้วก็ทำการเรียงลำดับค่า Weight Distance โดยใช้วิธีการ Bubble Sort จากนั้นแสดงผลลัพธ์ที่ได้ ซึ่งผู้ใช้สามารถป้อนกลับของข้อมูลที่เป็น Key-Frame เข้าไปได้อีกจนกว่าจะได้ผลลัพธ์ที่ต้องการ

บทที่ 4

ผลการทดลอง

ในการทดลองได้มีการเปรียบเทียบประสิทธิภาพระหว่างการค้นหาโดยใช้ดรรชนีเฟรมหลัก (Key-Frame) กับการค้นหาวิดีโอ Shot โดยใช้หลักการและทฤษฎีเกี่ยวกับเทคนิคการป้อนกลับ (Relevance Feedback) ว่ามีข้อแตกต่างกันอย่างไร ผลลัพธ์ที่ได้ตรงตามความต้องการของผู้ใช้หรือไม่ จึงได้ทำการศึกษาผลของการค้นหาวิดีโอ Shot โดยใช้ดรรชนีเฟรมหลักก่อนว่ามีประสิทธิภาพเป็นอย่างไร

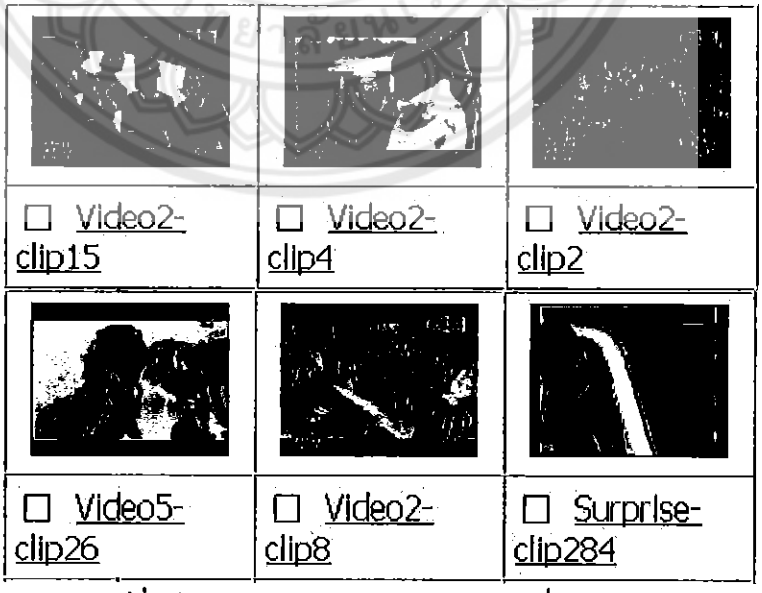
4.1 ผลการทดลองโดยใช้ดรรชนีเฟรมหลักในการค้นหาวีดีโอ Shot

ในการทดลองนี้ผู้ใช้จะทำการเลือกรูปภาพที่เป็นดรรชนีเฟรมหลัก (Key-Frame) ได้เพียงรูปเดียวเท่านั้น โดยในขั้นตอนนี้ตัวโปรแกรมจะยังไม่สามารถทำการตอบโต้กับผู้ใช้ได้ ดังนั้นส่วนของอินพุตที่ป้อนให้กับโปรแกรมในขั้นตอนนี้จึงมีเพียงอินพุตเดียวซึ่งก็คือรูปภาพที่เป็น Key-Frame ที่ผู้ใช้ได้ทำการเลือกนั่นเอง

รูปที่ 4.1 แสดงรูปภาพที่เป็น Key-Frame จากฐานข้อมูลบางส่วนเพื่อจะทำการเลือก

รูปที่ 4.2 แสดงขั้นตอนการเลือก Key-Frame เพียงอันเดียวเพื่อให้ในการค้นหาวิดีโอ Shot

จากนั้นเมื่อผู้ใช้ทำการเลือกรูปภาพที่เป็น Key-Frame ที่จะใช้ในการค้นหาได้แล้ว ต่อไปผู้ใช้จะทำการตกลงที่จะเลือกรูปภาพที่เป็น Key-Frame นี้ในการค้นหา ซึ่งจากรูปที่ 4.2 นี้ได้ทำการเลือก Key-Frame ที่ชื่อว่า Video2-clip84 เพื่อใช้เป็น Key-Frame ตั้งต้นในการค้นหาโดยกดที่ปุ่ม Submit เพื่อตกลงทำการค้นหาซึ่งจะได้ผลลัพธ์ดังรูปที่ 4.3



รูปที่ 4.3 แสดงผลลัพธ์ที่ได้บางส่วนจากการใช้ Key-Frame ชื่อ Video2-clip84 ในการค้นหา

จากรูปที่ 4.3 นี้รูปภาพที่เป็น Key-Frame จะทำการเรียงลำดับความความแตกต่างของภาพ ซึ่งรูปภาพที่เป็น Key-Frame ที่ชื่อว่า Video2-clip15 จะมีค่าความแตกต่างน้อยซึ่งแสดงว่ามีค่าความเหมือนของรูปภาพที่เป็น Key-Frame นี้มากที่สุด โดยไล่ลำดับความเหมือนของภาพได้ดังนี้

1. Video2-clip15
2. Video2-clip4
3. Video2-clip2
4. Video5-clip26
5. Video2-clip8
6. Surprise-clip284

ซึ่งลำดับความเหมือนข้างต้นนี้ ได้ทำการยกตัวอย่างเพียงแค่ 6 ภาพแรกของ Key-Frame ที่มีลักษณะความเหมือนกันกับภาพของ Key-Frame ที่ชื่อ Video2-clip84 ซึ่งในโปรแกรมการทำงานจริงนั้นจะแสดงรูปภาพของ Key-Frame ที่ประมวลผลได้ออกมา 18 Key-Frame

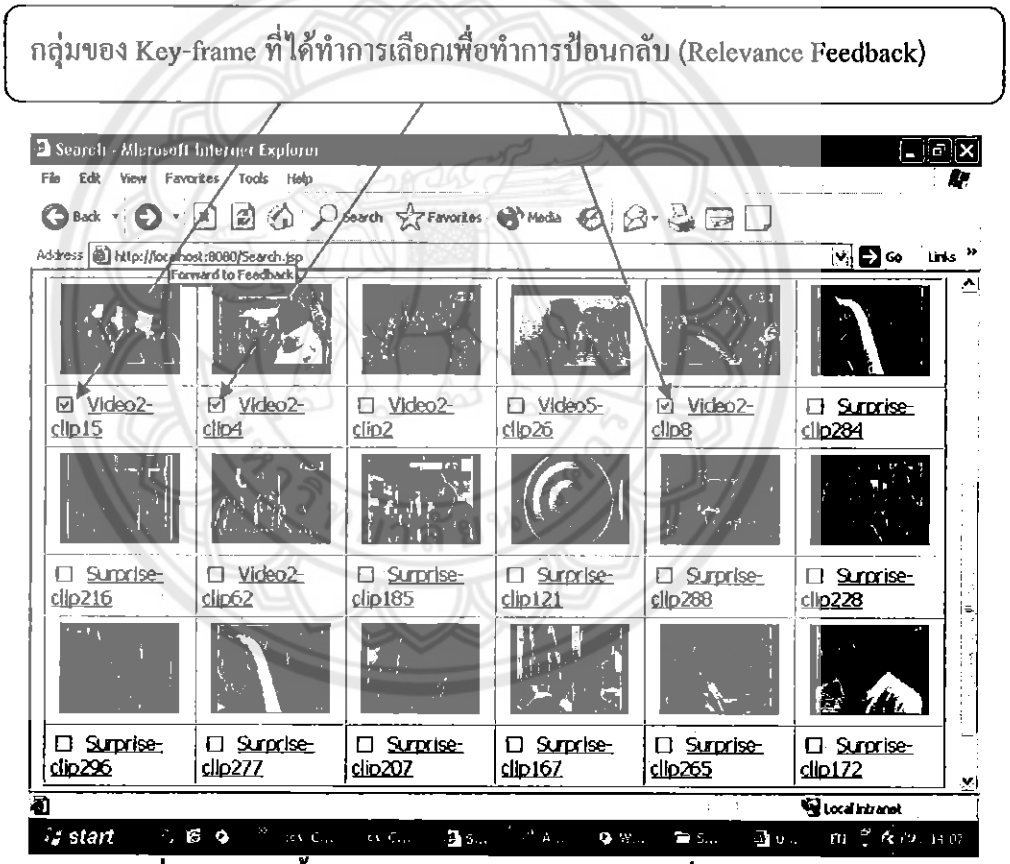
รูปที่ 4.4 แสดงผลลัพธ์ทั้งหมดจากการค้นหาภาพโดยใช้ Key-Frame

ซึ่งจากผลลัพธ์ที่ได้จากการค้นหาโดยใช้ Key-Frame ที่ชื่อว่า Video2-clip84 จะได้ผลลัพธ์ดังรูปที่ 4.4 ซึ่งจะสังเกตว่ายังมีบางรูปภาพที่เป็น Key-Frame นั้นยังมีบาง Key-Frame ที่ไม่ใช่

หรือไม่ตรงตามความต้องการของผู้ใช้ จากขั้นตอนนี้เองต่อไปจะมีการใช้หลักการและทฤษฎีเกี่ยวกับเทคนิคการป้อนกลับ (Relevance Feedback) เข้ามาใช้เพื่อให้ผลลัพธ์ที่ถูกต้องมากขึ้น

4.2 ผลการทดลองโดยใช้หลักการและทฤษฎีเกี่ยวกับเทคนิคการป้อนกลับ (Relevance Feedback)

จากรูปที่ 4.4 จะสังเกตว่าช่องที่ใช้ในการเลือกรูปภาพที่เป็น Key-Frame นี้จะเปลี่ยนไปจากเดิมที่เป็น Ratio Box ก็จะมาใช้เป็น Check Box เพื่อให้ผู้ใช้ได้เลือก Key-Frame ที่ตรงตามความต้องการของผู้ใช้มากขึ้น ซึ่งในหน้าจอนี้จะขึ้นอยู่กับผู้ใช้ว่าต้องการรูปภาพที่เป็น Key-Frame อันไหนบ้างที่ต้องการจะค้นหาและให้ได้ Key-Frame ที่ใกล้เคียงกับความต้องการมากขึ้น



รูปที่ 4.5 แสดงขั้นตอนการเลือกกลุ่ม Key-Frame เพื่อนำไป Feedback

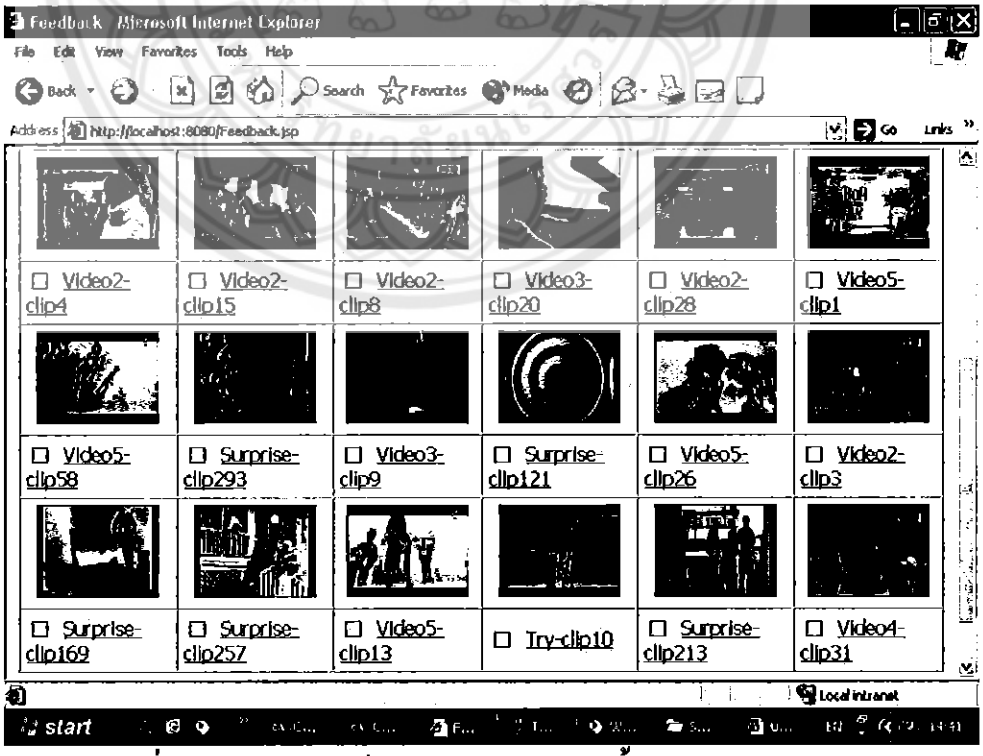
จากรูปที่ 4.5 เราได้ทำการทดลองโดยเลือกกลุ่ม Key-Frame ซึ่ง Key-Frame ที่เลือกมี 3 Key-Frame ดังนี้ 1. Video2-clip5, 2. Video2-clilp4 และ 3. Video2-clip8 เมื่อเลือกกลุ่มของ Key-Frame ได้แล้วก็ให้กดปุ่ม Submit เพื่อนำกลุ่มของ Key-Frame ไปประมวลผล และแสดงผลลัพธ์กลับมาให้ผู้ใช้ โดยแสดงผลลัพธ์จากการทำ Relevance Feedback ดังรูปที่ 4.6

รูปที่ 4.6 แสดงผลลัพธ์ที่ได้จากการทำ Relevance Feedback

จากรูปที่ 4.6 จะสังเกตว่ากลุ่มรูปภาพของ Key-Frame ที่ได้ทำการเลือกในรูปที่ 4.5 จะได้ว่าลำดับของผลลัพธ์จากการทำ Relevance Feedback จะเปลี่ยนไป โดยในลำดับต้น ๆ จะมีกลุ่มรูปภาพของ Key-Frame ที่ได้ทำการเลือกเพื่อป้อนกลับนี้ ซึ่งจากรูปภาพที่เป็น Key-Frame ตั้งต้นนั่นมีค่าความเป็นสีส่วนใหญ่อยู่ในโทนสีเขียว ดังนั้นผลลัพธ์ที่ได้ส่วนใหญ่จะอยู่ในแถบสีเขียวนี้ด้วย(ใช้หลักการเปรียบเทียบค่า Histogram ของค่าสี)

จากขั้นตอนนี้เองผู้ใช้สามารถป้อนกลับของกลุ่มรูปภาพที่เป็น Key-Frame นี้เข้าไปได้อีกเพื่อต้องการข้อมูลที่มีความถูกต้องมากยิ่งขึ้น ซึ่งส่วนที่ผู้ใช้ทำการป้อนกลับอีกครั้งนี้ จะขึ้นอยู่กับผู้ใช้แต่ละคนในการเลือกรูปภาพที่เป็น Key-Frame ในการป้อนกลับ โดยในที่นี้จะทำการเลือกกลุ่มรูปภาพที่เป็น Key-Frame ที่มีลักษณะของเนื้อหาและดูจากโทนสีของพื้นหลัง โดยจะทำการเลือกให้ใกล้เคียงกับรูปภาพที่เป็น Key-Frame ตั้งต้นให้มากที่สุด

รูปที่ 4.7 แสดงกลุ่มของ Key-Frame ที่ทำการป้อนกลับอีกครั้งเพื่อให้ได้ข้อมูลที่ถูกต้องมากขึ้น



รูปที่ 4.8 แสดงผลลัพธ์จากการป้อนกลับอีกครั้งของกลุ่ม Key-Frame

ผลลัพธ์ที่ได้จากการใช้เทคนิคการป้อนกลับ (Relevance Feedback) นี้จะมีความถูกต้องและตรงตามความต้องการของผู้ใช้มากขึ้น และให้คอมพิวเตอร์ที่ประมวลผลกับผู้ใช้ที่ทำการเลือกได้มีการโต้ตอบกันและกันมากขึ้น ซึ่งในการทำงานด้านอื่น ๆ ก็มีการนำหลักการและทฤษฎีเกี่ยวกับเทคนิคการป้อนกลับมาช่วยในการประมวลผล เพื่อลดความผิดพลาดของผลลัพธ์ที่ได้ ซึ่งจากการทดลองนี้ การค้นหาโดยใช้ Key-Frame ในการค้นหาเพียงอย่างเดียวนี้จะมีความรวดเร็วในการค้นหามากกว่า อย่างไรก็ตามความถูกต้องที่แสดงออกมาก็น้อยตามไปด้วย

ดังนั้นในการวัดประสิทธิภาพในการค้นหาวิดีโอไฟล์ตามที่ผู้ใช้ต้องการนี้ ได้มีการทดลองโดยสุ่มค้นหาข้อมูลโดยใช้ Key-Frame จำนวน 20 Key-Frame แล้วหาเปอร์เซ็นต์ความถูกต้องในการค้นหาข้อมูลที่มีอยู่จริง ซึ่งผลลัพธ์ที่ได้จากการค้นหาวิดีโอ Shot ที่ใกล้เคียงกับวิดีโอ Shot ตั้งต้นนี้จะแสดงผลลัพธ์จำนวน 18 Shot ซึ่งผลลัพธ์จริง ๆ ที่ได้นี้อาจมีไม่ถึง 18 Shot ดังนั้นเปอร์เซ็นต์ความถูกต้องของข้อมูลที่ได้นี้จะวัดจากวิดีโอ Shot ที่ใกล้เคียงกับวิดีโอ Shot ตั้งต้นทั้งหมด ยกตัวอย่างเช่น วิดีโอ Shot ที่มีชื่อว่า Mysassygirl-clip43 จะมีวิดีโอ Shot ที่มีลักษณะของเนื้อหาใกล้เคียงกับวิดีโอ Shot นี้จำนวน 10 Shot ซึ่งจะสังเกตว่าข้อมูลที่มีลักษณะใกล้เคียงกับวิดีโอ Shot ตั้งต้นจริง ๆ นี้มีไม่ถึง 18 Shot ถ้าจะหาเปอร์เซ็นต์จากจำนวนผลลัพธ์ที่ออกมาทั้งหมดที่แสดงให้ผู้ใช้งานเทียบกับความถูกต้องของวิดีโอ Shot ที่ใกล้เคียงกับวิดีโอ Shot ตั้งต้นจะทำให้เปอร์เซ็นต์ความถูกต้องคาดเคลื่อนได้ ดังนั้นตารางผลการทดลองที่ได้นี้จะแสดงเปอร์เซ็นต์ความถูกต้องของข้อมูลที่ได้ว่ามีค่าเป็นอย่างไร

ตารางที่ 4.1 ผลที่ได้จากการสุ่มค้นหาวิดีโอ Shot ที่มีลักษณะใกล้เคียงกับวิดีโอ Shot ตั้งต้น

Name	Key-frame All	Using Key-frame		Feedback 1		Feedback 2	
		Correct	Precision (%)	Correct	Precision (%)	Correct	Precision (%)
1. Mysassygirl-clip43	9	5	56%	8	89%	9	100%
2. Mysassygirl-clip103	6	6	100%	6	100%	6	100%
3. Mysassygirl-clip100	3	3	100%	3	100%	3	100%
4. Mysassygirl-clip117	9	9	100%	9	100%	9	100%
5. Mysassygirl-clip306	6	6	100%	6	100%	6	100%
6. Mysassygirl-clip272	4	4	100%	4	100%	4	100%
7. Mysassygirl-clip146	6	3	50%	5	83%	6	100%
8. Whoareyou-clip55	5	5	100%	5	100%	5	100%
9. Whoareyou-clip228	5	3	60%	5	100%	5	100%
10. Whoareyou-clip91	9	4	44%	9	100%	9	100%

ตารางที่ 4.1 (ต่อ)

Name	Key-frame All	Using Key-frame		Feedback 1		Feedback 2	
		Correct	Precision (%)	Correct	Precision (%)	Correct	Precision (%)
11. Concerto-clip276	3	2	67%	3	100%	3	100%
12. Concerto-clip257	7	4	57%	7	100%	7	100%
13. Conan-clip50	11	7	64%	10	91%	11	100%
14. Mysassygirl-clip199	8	6	75%	8	100%	8	100%
15. Concerto-clip224	4	2	50%	3	75%	4	100%
16. Surprise-clip220	3	3	100%	3	100%	3	100%
17. Surprise-clip236	4	3	75%	4	100%	4	100%
18. Surprise-clip129	4	4	100%	4	100%	4	100%
19. Mysassygirl-clip111	6	6	100%	6	100%	6	100%
20. Mysassygirl-clip123	13	9	69%	12	92%	13	100%

หมายเหตุ จากตารางที่ 4.1 ตัวอักษรภาษาอังกฤษนี้จะใช้แทนความหมายเกี่ยวกับการค้นหาวิดีโอ Shot ดังนี้

Name = ชื่อวิดีโอไฟล์ที่ทำการค้นหา **Key-Frame All** = จำนวน Key-Frame ทั้งหมดที่มีลักษณะของเนื้อหาใกล้เคียงกับวิดีโอ Shot ตั้งต้น

Feedback 1 = การค้นหาวิดีโอ Shot โดยใช้เทคนิคการป้อนกลับครั้งที่ 1 **Feedback 2** = การค้นหาวิดีโอ Shot โดยใช้เทคนิคการป้อนกลับครั้งที่ 2

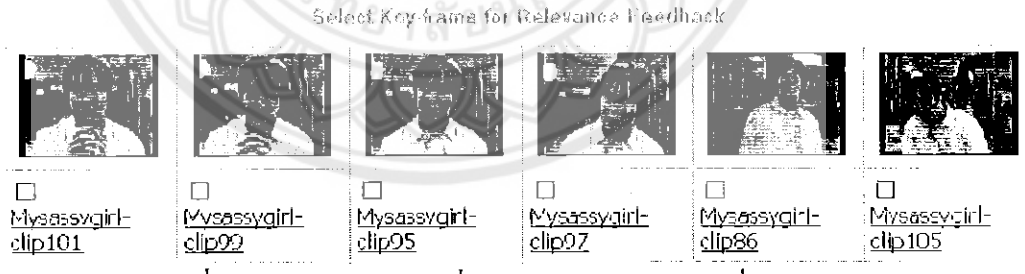
Correct = จำนวน Key-Frame ที่ถูกต้อง **Precision (%)** = เปอร์เซ็นต์ความถูกต้องของผลลัพธ์ที่แสดง

Using Key-Frame = การค้นหาวิดีโอ Shot โดยใช้ Key-Frame เปรียบเทียบความคล้ายคลึงกัน

จากตารางที่ 4.1 นี้จะแบ่งออกเป็น 2 ส่วนด้วยกันคือ ส่วนข้อมูลวิดีโอ Shot ที่ได้ทำการสุ่มค้นหาใน 10 ลำดับแรกนี้ ลักษณะของเนื้อหา, องค์ประกอบภาพของ Key-Frame และลักษณะของโทนสีในภาพมีลักษณะที่ชัดเจนเป็นเอกลักษณ์

แต่ในส่วนที่สองคือส่วนข้อมูลวิดีโอ Shot ในลำดับที่ 11-20 นี้จะดูจากลักษณะหลาย ๆ อย่างและขึ้นอยู่กับความต้องการของผู้ใช้แต่ละคนในการเลือกภาพที่เป็น Key-Frame โดยค่า Key-Frame All ในตารางนี้จะมีค่าไม่แน่นอนขึ้นอยู่กับผู้ใช้แต่ละคนที่จะเลือก โดย Key-Frame ส่วนใหญ่ที่เลือกนี้จะดูจากตัวละครที่เด่นชัดในวิดีโอ Shot, ลักษณะสีของเสื้อผ้า และองค์ประกอบของฉากหลัง (Background) และอีกอย่างคือความต่อเนื่องของเรื่องราวของวิดีโอ Shot ที่ใกล้เคียงกัน ถ้ามองดูแล้วภาพที่เป็น Key-Frame ในแต่ละวิดีโอ Shot ที่แสดงผลให้ผู้ใช้งานนี้จะมีลักษณะของภาพที่เป็น Key-Frame ไม่ค่อยใกล้เคียงกัน แต่เมื่อทำการชมวิดีโอ Shot แล้วก็จะพบว่าเนื้อหาข้างในนั้นเป็นเรื่องราวอย่างเดียวกัน ซึ่งในโครงงานนี้ถ้าสังเกตจะพบว่าชื่อวิดีโอไฟล์นี้ จะตั้งชื่อไว้อย่างเดียวกันแต่แตกต่างกันตรงลำดับ Shot ยกตัวอย่างเช่น Concerto-clip123 ซึ่งตัว clip123 นี้คือลำดับวิดีโอ Shot ของวิดีโอไฟล์ที่ชื่อว่า Concerto นั่นเอง จึงทำให้ผู้ใช้งานทราบว่านี่คือวิดีโอไฟล์อย่างเดียวกันและง่ายต่อการเลือกวิดีโอ Shot ที่ใกล้เคียงกัน

ในส่วนต่อไปนี้จะอธิบายเกี่ยวกับผลลัพธ์ที่แสดงใช้ผู้ใช้งานทราบ ซึ่งจากการทดลองจะสังเกตว่าในลำดับการสุ่มวิดีโอ Shot ที่ 1-10 นี้ลำดับความคล้ายคลึงกันของวิดีโอ Shot ที่ใช้ Key-Frame เป็นตัวบ่งบอกลักษณะของเนื้อหานี้จะมีลักษณะของผลลัพธ์ที่เกาะกลุ่มกัน จะมีบางวิดีโอ Shot ที่ลำดับความคล้ายคลึงกันไม่เกาะกลุ่ม ซึ่งรูปที่ 4.9 นี้ได้ใช้ Key-Frame ของวิดีโอ Shot ที่ชื่อว่า Mysassygirl-clip103 เป็นตัวค้นหา



รูปที่ 4.9 ลำดับของผลลัพธ์ที่มีลักษณะของวิดีโอ Shot ที่คล้ายคลึงกัน

แต่ในลำดับการสุ่มตัวอย่างที่ 11-20 ผลลัพธ์ที่แสดงส่วนใหญ่จะอยู่กันอย่างกระจายไม่เกาะกลุ่ม มีบางตัวอย่างเมื่อทำการป้อนกลับแล้วพบว่าผลลัพธ์ที่ได้อยู่เกาะกลุ่มกัน จึงทำให้สามารถวิเคราะห์ผลลัพธ์และสรุปผลลัพธ์ที่ได้ออกมาเป็นหัวข้อใหญ่ ๆ ดังนี้

สรุปผลการทดลองจากตารางการทดลอง

1. ในการค้นหาวิดีโอ Shot ที่ใกล้เคียงกับวิดีโอ Shot ตั้งต้นนี้ ถ้าวิดีโอ Shot ตั้งต้นมี Key-Frame ที่บ่งบอกลักษณะของเนื้อหาได้ชัดเจนรวมทั้งวิดีโอ Shot ที่ใกล้เคียงก็มี Key-Frame ที่ใกล้เคียงกันมากกับ Key-Frame ตั้งต้นของวิดีโอ Shot ที่ต้องการหาแล้ว จะทำให้การค้นหาครั้งแรกโดยเปรียบเทียบความคล้ายคลึงกันของภาพนี้มีเปอร์เซ็นต์ความถูกต้องอยู่ในระดับสูง

2. การค้นหาวิดีโอ Shot โดยใช้วิธีการหาความคล้ายคลึงกันของภาพกับการค้นหาวิดีโอ Shot โดยใช้เทคนิคการป้อนกลับ (Relevance Feedback) แล้วทำให้ทราบว่าการใช้เทคนิคการป้อนกลับเข้ามาช่วยนี้ทำให้การค้นหาข้อมูลมีความถูกต้องมากขึ้น ซึ่งจากตารางการทดลองนี้การใช้เทคนิคการป้อนกลับในครั้งที่ 1 จะมีเปอร์เซ็นต์ความถูกต้องเพิ่มขึ้นกว่าการใช้วิธีการหาความคล้ายคลึงกันของภาพ

3. การใช้เทคนิคการป้อนกลับ (Relevance Feedback) เมื่อวิดีโอ Shot ที่คล้ายคลึงกันกับวิดีโอ Shot ตั้งต้นถูกแสดงออกมาครบแล้วเมื่อผู้ใช้ทำการป้อนข้อมูลกลับเข้าไปให้คอมพิวเตอร์ประมวลผล ผลลัพธ์ที่ได้ก็เหมือนเดิมไม่มีการเปลี่ยนแปลง จึงทำให้เปอร์เซ็นต์ความถูกต้องมีค่าเท่าเดิมด้วย

4. ถ้าดรรชนีเฟรมหลัก (Key-Frame) ที่เป็นตัวบ่งบอกลักษณะของเนื้อหาในวิดีโอ Shot นั้น ๆ มีลักษณะของภาพที่คลุมเครือไม่ชัดเจน รวมทั้งมีจำนวนวิดีโอ Shot ที่ใกล้เคียงอยู่น้อยก็จะทำให้เปอร์เซ็นต์ความถูกต้องมีค่าลดลงตามไปด้วย

5. ผลลัพธ์ที่ได้จากการค้นหาพบว่ามีบาง Key-Frame ของวิดีโอ Shot ที่มีลักษณะของโทนสีที่ไม่เหมือนกันเลยกับ Key-Frame ตั้งต้น ซึ่งทำให้วิเคราะห์ได้ว่า Key-Frame ตั้งต้นที่เป็นตัวบ่งบอกลักษณะของเนื้อหาในวิดีโอ Shot นั้น ๆ เพียงแค่ใช้องค์ประกอบภาพทางด้านสีอย่างเดียวไม่สามารถบ่งบอกลักษณะของเนื้อหาได้ชัดเจน

6. ถ้า Key-Frame ตั้งต้นของวิดีโอ Shot ไหนที่มีองค์ประกอบทางด้านสีอย่างชัดเจน จะทำให้ผลลัพธ์ที่ได้มีลักษณะขององค์ประกอบทางด้านสีที่ใกล้เคียงกับ Key-Frame ตั้งต้น ตัวอย่างเช่น วิดีโอ Shot ที่มีลักษณะของเนื้อหาเกี่ยวกับน้ำ ก็จะทำให้ได้ผลลัพธ์ที่มีลักษณะของเนื้อหาที่เกี่ยวกับโทนสีฟ้า

บทที่ 5
สรุปผลและวิเคราะห์ผล

5.1 สรุปผลการทดลองที่ได้จากการพัฒนาโปรแกรม

จากหลักการและทฤษฎีเกี่ยวกับเทคนิคการป้อนกลับ (Relevance Feedback) ทำให้ได้ผลสรุปว่า การประมวลผลของโปรแกรมนั้นมีประสิทธิภาพมากขึ้นอย่างเห็นได้ชัด เพราะเมื่อทำการป้อนกลับของกลุ่มรูปภาพที่เป็น Key-Frame แล้ว ผลลัพธ์ที่ได้นั้นมีความถูกต้องมากขึ้นโดยมี Key-Frame ที่ใกล้เคียงกับ Key-Frame ตั้งต้นเพิ่มมากขึ้น และเมื่อทำการป้อนกลับไปอีกครั้ง ผลลัพธ์ที่ได้ก็จะมีความถูกต้องมากขึ้น ซึ่งจากการวิเคราะห์การทำงานในขั้นตอนนี้ได้ผลคือ เมื่อมีการตอบโต้กับผู้ใช้จะทำให้ผู้ใช้นั้นได้มีส่วนคิดและเลือกข้อมูลที่จะนำไปป้อนกลับอีกครั้ง จึงทำให้ข้อผิดพลาดที่จะเกิดขึ้นน้อยลงไปด้วย ส่วนในเรื่องของเวลาในการประมวลผลนี้ การประมวลผลในขั้นตอนของการค้นหาวิดีโอ Shot โดยใช้หลักการและทฤษฎีเกี่ยวกับเทคนิคการป้อนกลับนั้นใช้เวลาพอสมควรในการประมวลผล เนื่องจากต้องทำการหาค่าเบี่ยงเบนมาตรฐานของกลุ่ม Key-Frame ที่ได้ทำการป้อนกลับให้กับโปรแกรม และจากนั้นจะต้องนำค่าที่ได้ไป Weight เพื่อหาค่าความแตกต่างของภาพอีกครั้ง แล้วค่อยนำไปเรียงลำดับค่าจากน้อยไปหามาก ซึ่งผลกระทบจากเวลาในการประมวลผลนี้เอง ทำให้วิเคราะห์ได้ดังนี้

5.2 ผลที่ได้จากการวิเคราะห์

5.2.1 จำนวน Key-Frame ในฐานข้อมูลทั้งหมดนี้จะมีผลกระทบโดยตรงต่อเวลาในการประมวลผล

5.2.2 ความต้องการของผู้ใช้แต่ละคนนั้นมีความต้องการไม่เหมือนกันทำให้ผลลัพธ์บางส่วนขึ้นอยู่กับการเลือก Key-Frame ของผู้ใช้ที่ต้องการจะป้อนกลับให้กับโปรแกรม

5.2.3 ประสิทธิภาพในการประมวลผล ซึ่งบางฟังก์ชันนั้นน่าจะมีความรวดเร็วในการประมวลผลมากกว่านี้ เช่น การเรียงลำดับข้อมูลจากน้อยไปหามาก ซึ่งในการประมวลผลในเรื่องของการเรียงลำดับข้อมูล ได้ทำการใช้วิธีการ Bubble Sort ซึ่งมีหลายวิธีการที่มีความรวดเร็วมากกว่านี้ ยกตัวอย่างเช่น การเรียงลำดับโดยใช้วิธีการ Quick Sort ซึ่งมีความรวดเร็วกว่า Bubble Sort มาก แต่การเรียงลำดับค่าความแตกต่างนี้ก็แปรผันตรงกับจำนวน Key-Frame ด้วย ถ้าข้อมูลมีน้อยก็จะทำให้การเรียงลำดับมีความรวดเร็วมากขึ้นตามไปด้วย แต่ถ้าข้อมูลมีมากเวลาที่ใช้ในการเรียงลำดับก็จะมากตาม

5.2.4 ความเหมือนกันของ Key-Frame ที่ต้องการมีอยู่น้อยคือ เมื่อผู้ใช้ทำการ Feedback ไปเรื่อย ๆ ผลลัพธ์ที่ได้ยังคงเหมือนเดิมไม่มีการเปลี่ยนแปลง ซึ่งแสดงว่าผลลัพธ์ที่ได้นั้นไม่

สามารถประมวลผลหา Key-Frame ที่ต้องการได้มากไปกว่านี้แล้ว อาจจะสรุปได้ว่าข้อมูลที่ผู้ใช้ต้องการมีอยู่น้อยจึงไม่เพียงพอต่อความต้องการของผู้ใช้งาน

จากสิ่งต่าง ๆ เหล่านี้เองทำให้สามารถนำไปปรับปรุงแก้ไขกระบวนการทำงานในส่วนต่าง ๆ ให้มีประสิทธิภาพมากขึ้น ซึ่งขึ้นอยู่กับปัจจัยหลายอย่างที่จะนำมาปรับปรุงแก้ไข โดยอาจจะนำหลักการและทฤษฎีอย่างอื่นที่เกี่ยวข้องเข้ามาร่วมใช้งานเพื่อเพิ่มประสิทธิภาพในการทำงานมากยิ่งขึ้น

5.3 ปัญหาและการแก้ปัญหา

จากการทำการศึกษาเกี่ยวกับผลลัพธ์ที่ได้จากการค้นหาโดยใช้ดรรชนีเฟรมหลัก(Key-Frame) สรุปผลที่ได้จากการศึกษาและทำการทดลองคือ กลุ่มของรูปภาพที่เป็น Key-Frame นั้นมีบางรูปภาพที่ไม่ต้องตรงตามความต้องการของผู้ใช้ ซึ่งจากการวิเคราะห์ผลที่ได้นี้ทำให้ทราบว่า Content Base ที่เราได้นำมาใช้ในการประมวลผลนี้ ได้ใช้องค์ประกอบทางด้านสีของภาพ (Color Model) มาใช้ในการประมวลผลเพียงด้านเดียว ซึ่งในองค์ประกอบของภาพจริง ๆ จะต้องประกอบด้วยองค์ประกอบทางด้านรูปร่างของภาพ (Shape Model) และองค์ประกอบทางด้านข้อความในภาพ (Texture Model) ด้วยเพื่อที่จะบ่งบอกลักษณะของภาพได้มากขึ้น โดยในขั้นตอนนี้จะสังเกตได้ว่าเวลาที่ใช้ในการประมวลผลเพื่อหารูปภาพที่เป็น Key-Frame นี้จะใช้เวลาน้อยกว่าการค้นหาโดยใช้หลักการและทฤษฎีเกี่ยวกับเทคนิคการป้อนกลับ ซึ่งจากการวิเคราะห์พบว่าขั้นตอนนี้จะนำมาเปรียบเทียบความแตกต่างกันของภาพตั้งต้นที่เป็น Key-Frame เพียงครั้งเดียว แล้วทำการเรียงลำดับค่าความแตกต่างกันของภาพ ทำให้ใช้เวลาในการประมวลผลน้อยกว่า แต่ถ้าข้อมูลที่เป็น Key-Frame มีมากขึ้นอาจทำให้ต้องเสียเวลาในการประมวลผลเพิ่มมากขึ้นตามไปด้วย

5.4 แนวทางสำหรับการพัฒนาในครั้งต่อไป

แนวทางในการพัฒนาตัวโปรแกรมสำหรับผู้ที่จะนำไปใช้งานนั้นควรมีการปรับปรุงในส่วนของฐานข้อมูลให้มีความเป็นระเบียบมากขึ้น ควรจัดเก็บข้อมูลให้เป็นประเภทและหมวดหมู่มากขึ้น และในส่วนของประสิทธิภาพในการประมวลผลนั้นซึ่งบางฟังก์ชันนั้นน่าจะมีความรวดเร็วในการประมวลผลมากกว่านี้ เช่น การเรียงลำดับข้อมูลจากน้อยไปหามาก ซึ่งในการประมวลผลในเรื่องของการเรียงลำดับข้อมูล ทางผู้จัดทำได้ทำการใช้วิธีการ Bubble Sort ซึ่งมีหลายวิธีการที่มีความรวดเร็วมากกว่านี้ ยกตัวอย่างเช่น การเรียงลำดับโดยใช้วิธีการ Quick Sort ซึ่งมีความรวดเร็วกว่า Bubble Sort มาก

เอกสารอ้างอิง

[1] ดร.วีรศักดิ์ ซึงถาวร . **Java Programming Volume I** . กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น. 2543

[2] ดร.วีรศักดิ์ ซึงถาวร. **Java Programming Volume II** . กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น. 2543

[3] กิตติ ภักดีวัฒนะกุล. **คัมภีร์ Java** . พิมพ์ครั้งที่ 1 กรกฎาคม 2546 . กรุงเทพมหานคร : เคทีพี คอม แอนด์ คอลซัลท์ . 2546

[4] สาธิต ชัยวิวัฒน์ตระกูล. **เก่ง JSP ให้ครบสูตร** . พิมพ์ครั้งที่ 1 ธันวาคม 2545 . กรุงเทพมหานคร : วิตตี้กรุ๊ป จำกัด . 2545

[5] รศ.ดร. มนัส สังวรศิลป์ ,วรรัตน์ ภัทรอมรกุล . **คู่มือโปรแกรม Matlab ฉบับสมบูรณ์** . พิมพ์ครั้งที่ 1 เมษายน 2543 . กรุงเทพมหานคร : อินโฟเพรส . 2543

[6] มณีโชติ สมานไทย . **สร้างและปรับแต่งเว็บเพจอย่างง่ายด้วย HTML** . กรุงเทพมหานคร : อินโฟเพลส . 2544

[7] ศิวณัฏฐ์ มาศสุรางค์ . **Java Server Pages** . พิมพ์ครั้งที่ 1 มกราคม 2545 . กรุงเทพมหานคร : พิมพ์ดี . 2545

[8] Douglas A.LYON . **Image Processing in Java** . United State of America : Prentice Hall PTR . Inc.1999

[9] Yong Rui , Thamas S. Huang . **Image retrieval** . Dept. Of ECE & Beckman Institute University of Illinois at Urbana – Champaign Urbana, IL 61801

[10] G. Ciocca, R. Schettini . **Content – based similarity retrieval of trademarks using relevance feedback** . Pattern Recognition, Vol 34, 2001.

[11] Michele L. Jamrozik Georgia Tech Lorraine . **A Compressed domain video object segmentation system** . Dept. Of Elec.& Comp. Engineering F-57070 Metz, France

[12] John S. Boreczky , Lawrence A. Rowe . **Comparison of video shot boundary detection techniques** . Computer Science Division – EECS University of California Berkeley, CA 94720

ภาคผนวก ก.

คู่มือการติดตั้งโปรแกรมและการใช้งานของโปรแกรม

ในโครงงานนี้ตัวของโปรแกรมที่ใช้ในการพัฒนาได้ใช้ Jakarta – Tomcat - 4.0.1 และ J2sdk-1.3.1_01 ซึ่งตัวแรกคือโปรแกรมที่ใช้ติดต่อกับ Web server โดยใช้โค้ดการเขียนสคริปต์ JSP เข้ามาช่วยในส่วนของ Web page ส่วนตัวที่สองคือโปรแกรม Java ที่ใช้เขียนคำสั่งควบคุมการประมวลผล ซึ่งตัว Editor ที่ใช้ในการเขียนโค้ดที่เป็นสคริปต์ JSP และโค้ดของ Java ก็ตามเราได้ใช้ Editplus 2.0 ซึ่งเป็นโปรแกรมที่ใช้เขียน Editor ทั่วไป ซึ่งในส่วนของคู่มือการติดตั้งโปรแกรมนี้จะขออธิบายขั้นตอนการติดตั้งโปรแกรม Tomcat, Java และ MySQL ซึ่งตัว MySQL นี้เราได้นำมาใช้เป็นฐานข้อมูลให้กับโครงงานนี้

1. ขั้นตอนการเตรียมซอฟต์แวร์ที่ใช้ในการพัฒนาโครงงาน

1.1 Java 2 SDK, Standard Editon (เรียกย่อ ๆ ว่า J2SE) ซึ่งเป็นคอมไพเลอร์ของภาษา Java ที่จะต้องใช้ในการคอมไพล์โค้ดของ JSP ที่แปลงเป็น Servlet แล้ว (ซึ่งก็คือการคอมไพล์ไฟล์ Servlet ที่มีนามสกุล .java ให้กลายเป็นไฟล์ .class นั่นเอง) สามารถดาว์นโหลดได้ที่ http://java.sun.com/j2se/ สำหรับโครงงานนี้ได้ใช้เวอร์ชัน 1.3.1_01 ได้แก่ไฟล์ j2sdk-1_3_1_01-win.exe

1.2 Tomcat เป็นเว็บเซิร์ฟเวอร์ที่สนับสนุนการทำงานของ Servlet และ JSP โดย JSP Container ซึ่งอยู่ใน Tomcat จะทำหน้าที่แปลงไฟล์ .jsp เป็นไฟล์ .java โดยสามารถดาวน์โหลด Tomcat ได้ที่ http://jakarta.apache.org/site/binindex.html โดยเลือก Tomcat เวอร์ชันที่ต้องการ ซึ่งอยู่ในหัวข้อ Release Builds ในโครงงานนี้ได้เลือก Tomcat เวอร์ชัน 4.0.1

1.3 ตัว JSP เองนี้มีความสามารถในการติดต่อและจัดการกับฐานข้อมูล โดยอาศัยเทคโนโลยีที่เรียกว่า JDBC

JDBC หรือที่เรียกกันอีกอย่างหนึ่งว่า Java Database Connectivity เป็น API ที่ช่วยอำนวยความสะดวกในการติดต่อกับฐานข้อมูล หรือจะกล่าวง่าย ๆ ก็คือ JDBC เป็นกลุ่มของคลาสที่ใช้ในการติดต่อกับฐานข้อมูลนั่นเอง ดังนั้นเมื่อต้องการเขียนสคริปต์ JSP ติดต่อกับฐานข้อมูล ก็จะต้องเรียกใช้ไดรเวอร์ของ JDBC (JDBC driver) ในการติดต่อซึ่ง JDBC driver จะอยู่ในรูปของคลาส ซึ่งโปรแกรมระบบฐานข้อมูลที่นำมาใช้ในการติดต่อกับฐานข้อมูลนี้คือ MySQL เนื่องจากเป็นโปรแกรมที่ไม่ต้องเสียเงินซื้อ สามารถหาดาวน์โหลดได้ฟรีจากอินเทอร์เน็ต มีประสิทธิภาพในการใช้งานสูง และเป็นที่นิยมใช้กันอย่างแพร่หลาย ซึ่งแหล่งดาวน์โหลดโปรแกรม MySQL อย่าง

เป็นทางการอยู่ที่ http://www.mysql.com/downloads/index.html ซึ่งในโครงงานนี้ใช้ MySQL เวอร์ชัน 4.0.13

2. ขั้นตอนการติดตั้งโปรแกรม

ในขั้นตอนการติดตั้งโปรแกรมนี้จะขออธิบายการติดตั้งในแต่ละส่วนของโปรแกรมที่ใช้ โดยจะเริ่มจาก Java 2 SDK ก่อนเป็นอันดับแรกเพื่อป้องกันการเกิดปัญหาในการติดตั้งครั้งต่อไป

2.1 การติดตั้ง J2SE มีขั้นตอนดังนี้

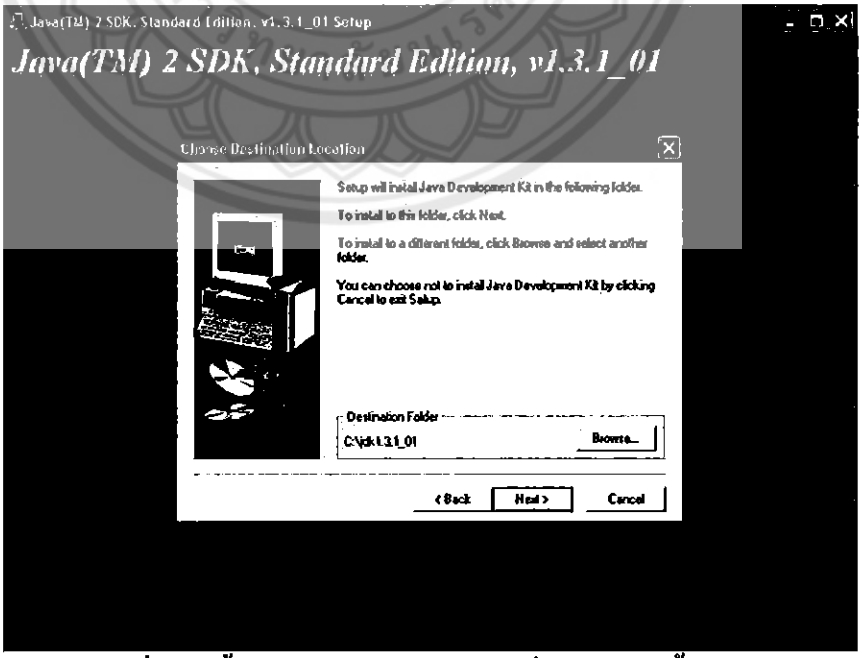
2.1.1 เริ่มต้นโดยดับเบิ้ลคลิ๊กที่ไฟล์ j2sdk-1_3_1_01-win.exe ที่ได้ดาวน์โหลดมา

2.1.2 โปรแกรมจะ extract ไฟล์เพื่อเตรียมการติดตั้ง

2.1.3 จากนั้นจึงเข้าสู่หน้าจอแสดงข้อความต้อนรับการติดตั้งโปรแกรม

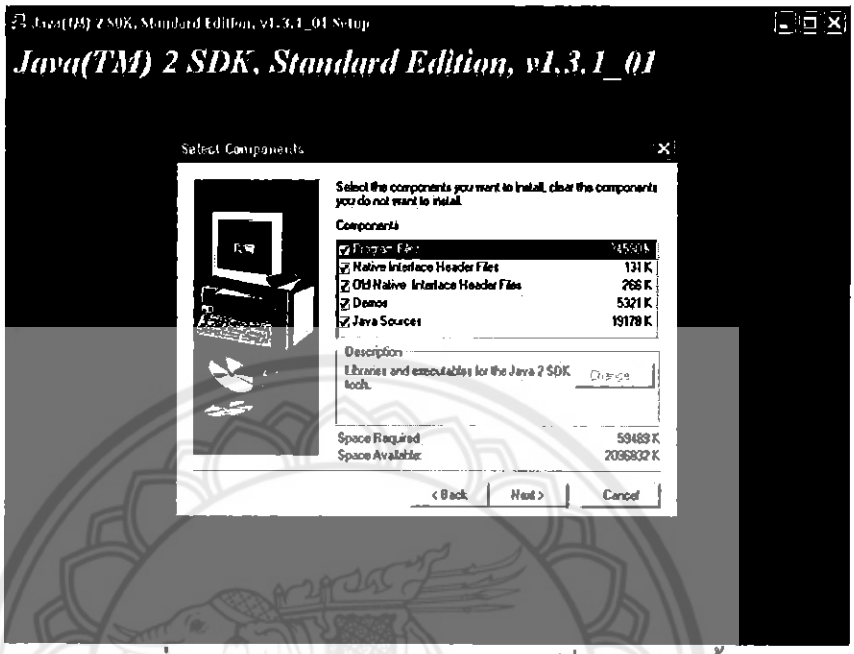
2.1.4 เมื่อคลิ๊กปุ่ม Next ก็จะเข้าสู่หน้าจอแสดงรายละเอียดเงื่อนไขข้อตกลงเกี่ยวกับการนำโปรแกรมไปใช้งาน หรือที่เรียกว่า Licence Agreement ตามธรรมเนียม

2.1.5 พอคลิ๊กปุ่ม Yes ยอมรับเงื่อนไข หน้าต่างถัดไปจะให้เลือกไดเรกทอรีที่ต้องการติดตั้ง J2SE ลงไปดังรูปที่ ก-1 ในที่นี้จะติดตั้งในไดเรกทอรีที่กำหนดเป็นดีฟอลต์ (Default) ไว้ล่วงหน้าแล้วคือ C:\Jdk1.3.1_01 แต่ถ้าต้องการติดตั้งในไดเรกทอรีอื่น ๆ ก็สามารถเปลี่ยนได้โดยคลิ๊กปุ่ม Browse



รูปที่ ก-1 ขั้นตอนการเลือกไดเรกทอรีที่ต้องการติดตั้ง J2SE

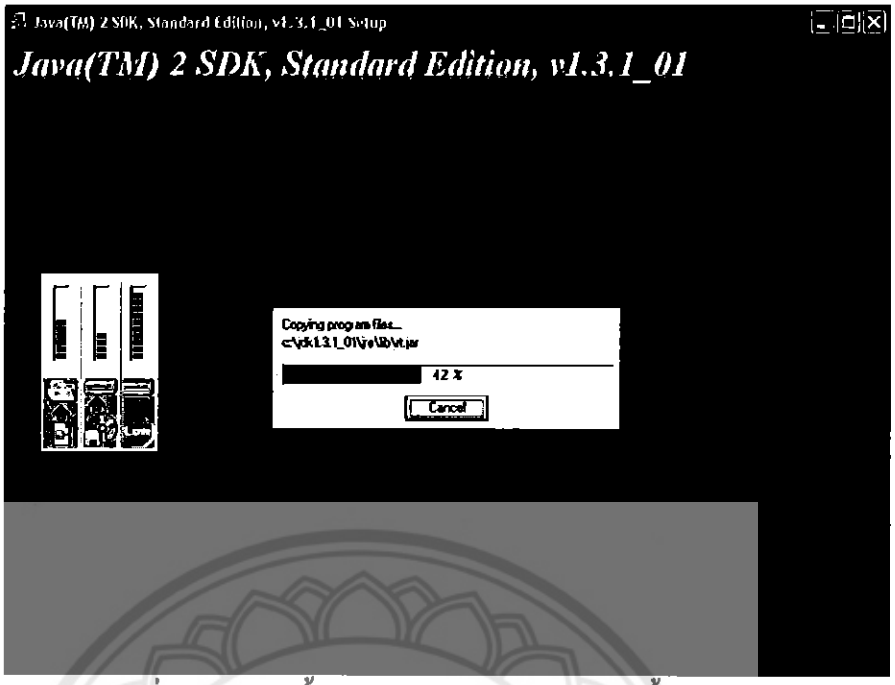
2.1.6 เมื่อเลือกไดเรกทอรีและคลิ๊กปุ่ม Next ก็จะไปที่หน้าต่างดังรูปภาพที่ ก-2 เพื่อให้เลือกส่วนประกอบที่ต้องการติดตั้ง ในที่นี้ให้เลือกหมดทุกอย่างแล้วคลิ๊กปุ่ม Next



รูปที่ ก-2 แสดงการเลือกคอมโพแนนต์ที่ต้องการติดตั้ง

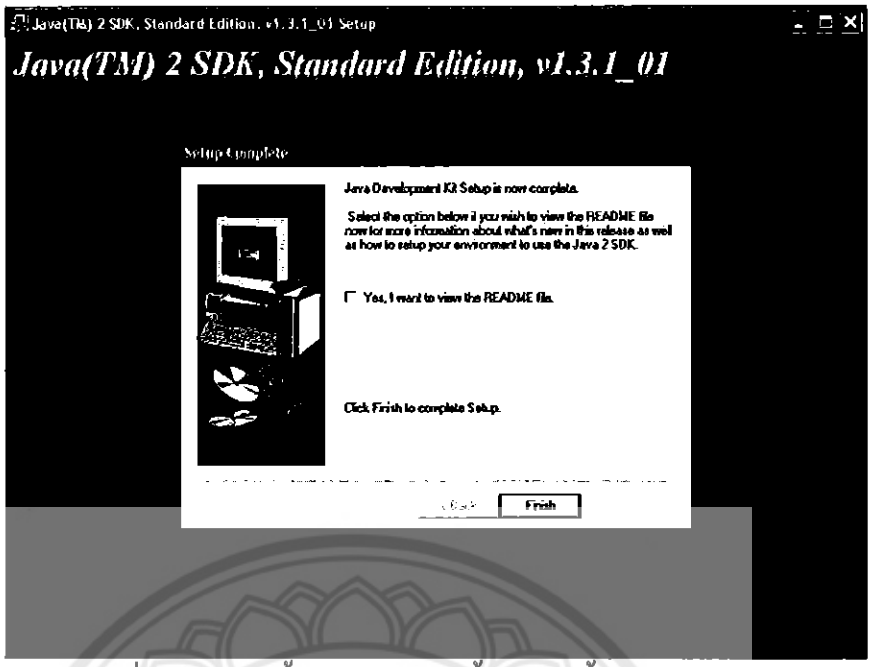
2.1.7 ต่อจากนั้นจะไปยังหน้าจอของการอัปเกรด Java Virtual Machine (JVM) ที่มีอยู่ในโปรแกรมเบราเซอร์ภายในเครื่องของเรา โดย JVM ทำหน้าที่เป็นตัวช่วยรันโค้ดของ Java หน้าจอนี้จะให้เราระบุว่าต้องการอัปเกรด JVM ในโปรแกรม IE หรือ Netscape ซึ่งส่วนใหญ่จะเลือกในโปรแกรม IE (Microsoft Internet Explorer) แล้วคลิ๊กปุ่ม Next

2.1.8 หลังจากนั้นโปรแกรมจะเริ่มต้นติดตั้งพร้อมทั้งแสดงเปอร์เซ็นต์ความคืบหน้าดังรูปที่ ก-3

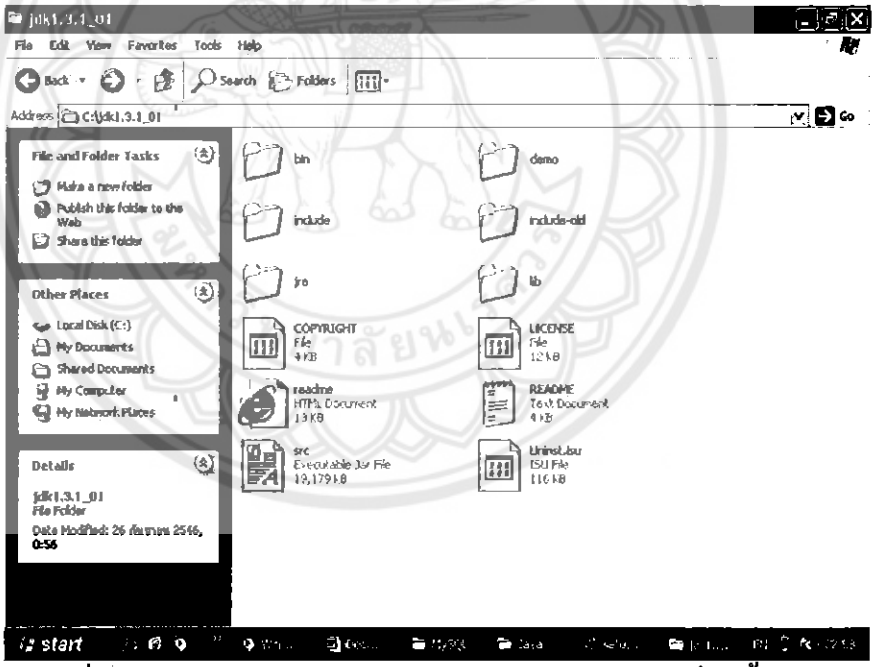


รูปที่ ก-3 แสดงขั้นตอนความคืบหน้าในการติดตั้งโปรแกรม

2.1.9 ถ้าการติดตั้งสมบูรณ์แล้วจะปรากฏหน้าจอดังรูปภาพที่ ก-4 เมื่อถึงตอนนี้ให้คลิก Finish ได้เลยแล้วก็จะได้โปรแกรม J2SE ที่ติดตั้งอยู่ในไดเรกทอรี jdk1.3.1_01 ซึ่งประกอบด้วยไดเรกทอรีย่อยต่าง ๆ ดังรูปภาพที่ ก-5 ซึ่งเมื่อทำการติดตั้ง J2SE เสร็จเรียบร้อยแล้วจะต้องทำการกำหนดตัวแปร PATH เพื่อให้เรียกใช้งานได้สะดวกมากขึ้น



รูปที่ ก-4 แสดงขั้นตอนการเสร็จสิ้นการติดตั้งโปรแกรม J2SE



รูปที่ ก-5 แสดงไฟล์และไดเรกทอรีย่อยภายในไดเรกทอรีที่ติดตั้ง J2SE

2.1.10 การกำหนดตัวแปร PATH ใน Windows 95 หรือ Windows 98 โดยคลิ๊กที่ปุ่ม Start ของระบบ Windows แล้วเลือกคำสั่ง RUN โดยเมื่อขึ้นหน้าต่าง RUN จะมี ComboBox ให้พิมพ์คำสั่ง sysedit ลงในช่อง Open แล้วคลิ๊ก OK ซึ่งเมื่อคลิ๊ก OK แล้วหน้าต่าง System Configuration Editor จะเปิดออกมา ให้คลิ๊กไปยังหน้าต่างย่อยของไฟล์ autoexec.bat ซึ่งหน้าต่างนี้

อาจจะมีข้อมูลอยู่แล้วหรืออาจจะเป็นเอกสารเปล่า ๆ ก็ได้ โดยตรวจดูว่าภายในหน้าต่างของไฟล์ autoexec.bat มีการกำหนดตัวแปร PATH ไว้หรือเปล่า ถ้ายังไม่มีก็พิมพ์คำสั่ง

path C:\jdk1.3.1_01\bin เข้าไป แต่ถ้ามีการกำหนดตัวแปร PATH ไว้แล้วเช่น

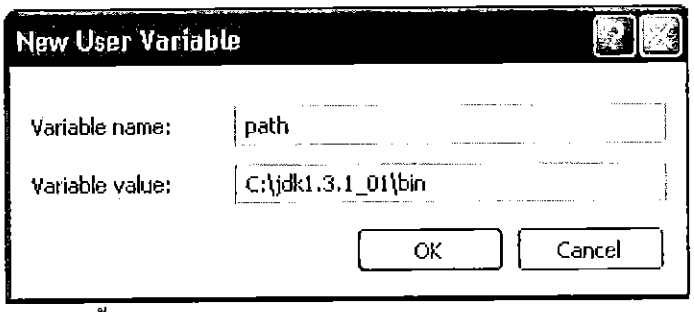
path C:\window;C:\windows\command ถ้าเป็นกรณีอย่างนี้ต้องเติมเครื่องหมาย ; เพิ่มเข้าไปแล้วตามด้วยพาธดังนี้

path C:\window;C:\windows\command;C:\jdk1.3.1_01\bin โดยพาธสุดท้ายที่เพิ่มเข้าไปไม่ต้องใส่เครื่องหมาย ; เมื่อกำหนดตัวแปร PATH เรียบร้อยแล้วให้เลือกคำสั่ง File > save จากหน้าต่าง System Configuration Editor เพื่อบันทึกการเปลี่ยนแปลงลงในไฟล์ autoexec.bat แล้วปิดหน้าต่างได้เลย

2.1.11 การกำหนดตัวแปร PATH ใน Window Me อันดับแรกให้คลิ๊กที่ปุ่ม Start ของ Windows Me แล้วเลือกคำสั่ง Run จากนั้นพิมพ์คำสั่ง msconfig ลงในช่อง Open แล้วคลิ๊ก OK จากนั้นจะขึ้นหน้าต่าง System Configuration Utility จากนั้นให้คลิ๊กที่แท็บ Environment แล้วคลิ๊กที่ PATH จากนั้นจึงคลิ๊กปุ่ม Edit ทำให้หน้าต่าง Edit Variable จะเปิดออก โดยในที่นี้ช่อง Variable Name มีชื่อตัวแปร PATH อยู่แล้ว ก็ให้ทำการเพิ่มพาธของ javac โดยพิมพ์ข้อความ ;C:\jdk1.3.1_01\bin ต่อท้ายพาธเดิมเข้าไป แล้วคลิ๊ก OK จากนั้นหน้าต่าง Edit Variable จะหายไปและกลับมายังแท็บ Environment ในหน้าต่าง System Configuration Utility เหมือนเดิม ซึ่งตอนนี้คอลัมน์ Value ของตัวแปร PATH มีพาธใหม่เพิ่มเข้าไปให้คลิ๊กปุ่ม OK เพื่อยืนยันการแก้ไขตัวแปร PATH ซึ่งหลังจากนั้นจะมีหน้าต่างปรากฏออกมาแจ้งให้เรา restart ระบบ Windows ซึ่งเป็นอันเสร็จสิ้นการกำหนด PATH ใน Windows Me

2.1.12 การกำหนด PATH ใน Windows 2000 และ Windows XP โดยทำการคลิ๊กที่ปุ่ม Start ของ Windows แล้วเลือกคำสั่ง Settings > Control Panel ซึ่งจากขั้นตอนนี้จะปรากฏหน้าต่างของ Control Panel จากนั้นให้ดับเบิ้ลคลิ๊กที่ไอคอน System ในหน้าต่าง Control Panel นี้ ซึ่งทำให้หน้าต่าง System Properties เปิดออกมา จากนั้นคลิ๊กไปที่แท็บ Advanced แล้วคลิ๊กที่ปุ่ม Environment Variables ซึ่งหน้าต่าง Environment Variables จะเปิดขึ้นมาโดยให้สังเกตภายในช่อง System variables ด้านล่างว่ามีการกำหนดตัวแปร PATH ไว้แล้วหรือยัง ถ้ามีแล้วให้คลิ๊กที่ตัวแปรนั้น แล้วคลิ๊กที่ปุ่ม Edit แต่ถ้ายังไม่มีให้คลิกปุ่ม New แทน ในกรณีนี้ให้คลิ๊กปุ่ม New ทำให้หน้าต่าง New System Variable เปิดออกมาโดยให้กำหนดค่าลงไปตามรูปภาพที่... แล้วคลิ๊กปุ่ม OK ส่วนในกรณีที่คลิ๊กปุ่ม Edit จะมีหน้าต่างเปิดออกมาในลักษณะเดียวกัน แต่มีค่าเดิมของตัวแปร PATH อยู่แล้ว เราก็เพิ่ม **;C:\jdk1.3.1_01\bin** ต่อท้ายค่าเดิมลงไปแล้วคลิ๊กปุ่ม OK ก็จะเป็นการเสร็จสิ้นขั้นตอนการกำหนดตัวแปร PATH ใน Windows 2000 และ Windows XP ซึ่งค่าตัวแปรที่กำหนดเข้าไปจะมีผลกับระบบทันที โดยไม่จำเป็นต้อง restart เครื่องใหม่ แต่ถ้ามีการเปิดหน้าต่าง

Command Prompt ในระหว่างที่กำหนดตัวแปร PATH ค่าที่กำหนดไว้จะไม่มีผลต่อหน้าต่างนั้น ต้องเปิดหน้าต่างใหม่ขึ้นมา ค่าที่กำหนดในตัวแปร PATH ใหม่จึงจะมีผลตามต้องการ



รูปที่ ก-6 แสดงขั้นตอนการกำหนดพาธให้กับ Window 2000 และ Windows XP

2.2 การติดตั้ง Tomcat โดยมีขั้นตอนในการติดตั้งดังนี้

หลังจากที่ทำการติดตั้ง J2SE เรียบร้อย ก็มาถึงขั้นตอนการติดตั้งเว็บเซิร์ฟเวอร์ Tomcat บ้าง โดยให้ unzip ไฟล์ jakarta-tomcat-4.0.1 ไปเก็บไว้ในไดรฟ์ C: ซึ่งจะทำการสร้างไดเรกทอรีชื่อ jakatar-tomcat-4.0.1 ขึ้นมาโดยอัตโนมัติ แต่ควรเปลี่ยนชื่อไดเรกทอรีนี้ให้สั้นและกระชับเป็น tomcat เพื่อให้ดูง่ายในการใช้งาน ภายในไดเรกทอรี tomcat จะมีไฟล์และไดเรกทอรีย่อยมากมาย โดยแต่ละไดเรกทอรีย่อยมีหน้าที่ดังนี้

webapps เป็นไดเรกทอรีหลักที่ใช้เก็บไฟล์ภายในเว็บไซต์ของเราเอง โดยไฟล์สำคัญ ๆ เช่น ไฟล์ .jsp หรือไฟล์ .html จะต้องเก็บอยู่ในไดเรกทอรี ROOT ซึ่งซ้อนอยู่ในไดเรกทอรีนี้อีกชั้น ไดเรกทอรี webapps\ROOT จึงมีสถานะเป็น default directory คือเมื่อมีการเรียกเปิดไฟล์ต่าง ๆ ทางเบราเซอร์ เว็บเซิร์ฟเวอร์ Tomcat ก็จะมองหาไฟล์ใน webapps\ROOT เป็นหลักดังนั้นสคริปต์ JSP ที่เราจะเขียนขึ้นมาจึงต้องเก็บไว้ในไดเรกทอรีนี้ แต่ถ้าต้องการให้ไดเรกทอรีใดมีสถานะเป็น default directory เพื่อจะได้เก็บไฟล์สคริปต์ JSP ไว้ในไดเรกทอรีนั้นแทน webapps\ROOT ก็สามารถทำได้

bin เป็นไดเรกทอรีสำหรับเก็บไฟล์ที่เกี่ยวข้องกับการทำงานของ Tomcat เช่น การ Start โปรแกรมหรือการ Shutdown โปรแกรม Tomcat เป็นต้น

classes และ lib เป็นไดเรกทอรีสำหรับเก็บคลาสต่าง ๆ ที่นำมาใช้จากภายนอก เช่น คลาสที่เขียนขึ้นเองหรือคลาสที่ดาวน์โหลดมาจากเว็บไซต์ เป็นต้น โดยถ้าเป็นไฟล์ .class ต้องเก็บไว้ในไดเรกทอรี classes แต่ถ้าเป็นไฟล์คลาสหลาย ๆ ไฟล์ที่นำมาบีบอัดรวมกันเป็นไฟล์ .jar ต้องเก็บไว้ในไดเรกทอรี lib

common และ server เป็นไดเรกทอรีสำหรับเก็บคลาสมาตรฐานต่าง ๆ ของ JSP และ Servlet

conf เป็นไดเรกทอรีสำหรับเก็บไฟล์ที่เก็บค่าจำพวก configuration ซึ่งเกี่ยวข้องกับการทำงานของ Tomcat เช่น ไฟล์ server.xml หรือไฟล์ web.xml เป็นต้น

logs เป็นไดเรกทอรีสำหรับเก็บไฟล์ .log ของ Tomcat โดยไฟล์จำพวกนี้ทำหน้าที่เก็บบันทึกสถิติต่าง ๆ ในลักษณะ text file ธรรมดา เช่น ผู้ที่เข้ามาชมเว็บไซต์มาจาก IP Address ไหน เปิดเข้ามาชมเว็บเพจไหนหรือไฟล์ไหนในวันและเวลาใด เป็นต้น

temp สำหรับเก็บไฟล์ชั่วคราวต่าง ๆ ของ Tomcat

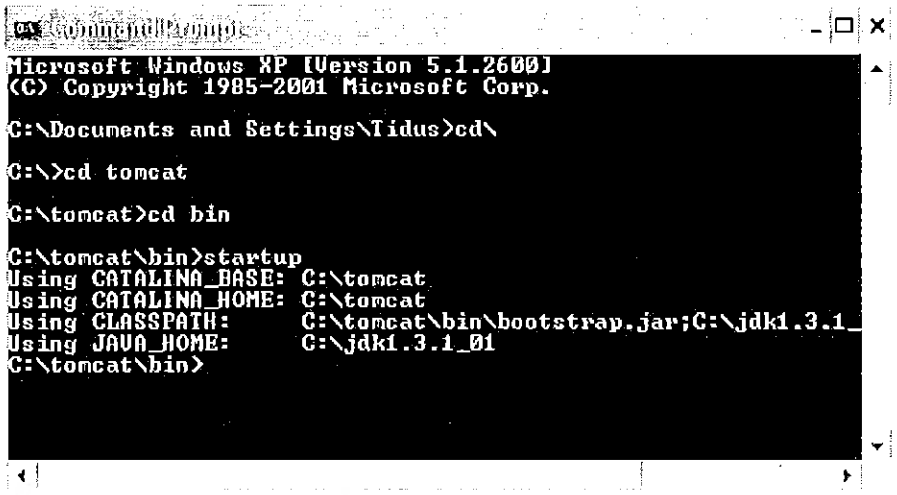
work เป็นไดเรกทอรีสำหรับเก็บไฟล์ .java และ .class ซึ่ง JSP Container แปลงมาจากไฟล์ .jsp ที่เขียนขึ้น

ซึ่งหลักจากติดตั้ง Tomcat เสร็จเรียบร้อยแล้ว จะต้องเพิ่มตัวแปรเข้าไปในระบบเหมือนกับตัวแปร PATH ซึ่งขั้นตอนการกำหนดตัวแปรก็จะคล้าย ๆ กัน นั่นคือ ในกรณีของ Windows 95, 98 ให้เพิ่มบรรทัดนี้ลงไปในไฟล์ C:\autoexec.bat

```
set JAVA_HOME = C:\j2sdk1.3.1_01
set CATALINA_HOME = C:\tomcat
```

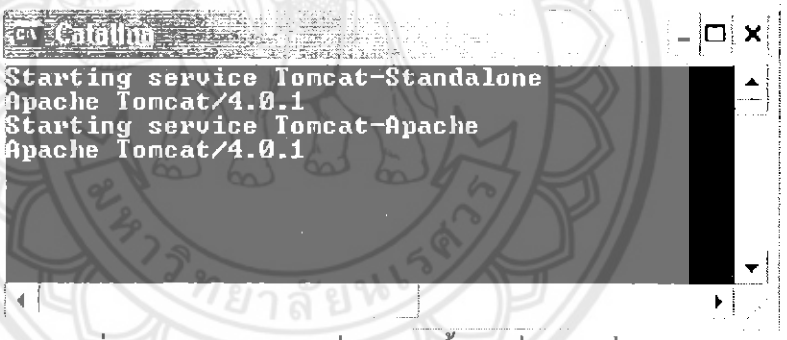
ส่วน Window Me, 2000, XP จะเป็นหน้าต่างให้กรอกข้อมูลในลักษณะเดียวกับรูปภาพที่ ก-6 โดยกำหนดตัวแปรใหม่ชื่อ JAVA_HOME มีค่าเป็น C:\j2sdk1.3.1_01 และตัวแปรอีกตัวหนึ่งชื่อ CATALINA_HOME มีค่าเป็น C:\tomcat หน้าที่ของ CATILINA_HOME เป็นตัวบ่งบอกให้ระบบรู้ว่าไดเรกทอรีที่ติดตั้ง Tomcat คือไดเรกทอรีไหน ส่วนตัวแปร JAVA_HOME ก็เป็นตัวบ่งบอกให้ระบบรู้ว่า ไดเรกทอรีที่ติดตั้ง J2SE คือไดเรกทอรีไหน

หลังจากที่ติดตั้งและกำหนดค่าต่าง ๆ เสร็จเรียบร้อยแล้ว ลองมาดูขั้นตอนการทดสอบการทำงานของ Tomcat โดยก่อนอื่นจะต้องสร้างชอร์ตคัตของ MS-DOS Prompt ไว้ที่หน้าจอ Desktop ของ Windows เพื่อความสะดวกในการเรียกใช้งาน เพราะการสั่ง Start การทำงานของ Tomcat ผ่านคอมมานด์ไลน์ และยังต้องใช้ชอร์ตคัตนี้ในการปรับเนื้อที่หน่วยความจำเมื่อมีปัญหาเกิดขึ้น หลังจากสร้างชอร์ตคัตแล้ว ก็ดับเบิ้ลคลิ๊กเพื่อเปิดหน้าต่าง MS-DOS Prompt ขึ้นมา จากนั้นจึงเข้าไปในไดเรกทอรี C:\tomcat\bin แล้วใช้คำสั่ง startup เพื่อเริ่มต้นการทำงานของ Tomcat ดังรูปที่ ก-7 ซึ่งโปรแกรมจะแจ้งค่าตัวแปรที่กำหนดไว้ออกมา สังเกตว่ามีค่าตัวแปร JAVA_HOME และ CATALINA_HOME ตามที่เราได้กำหนดก่อนหน้านี้แล้ว แจ้งออกมาด้วย



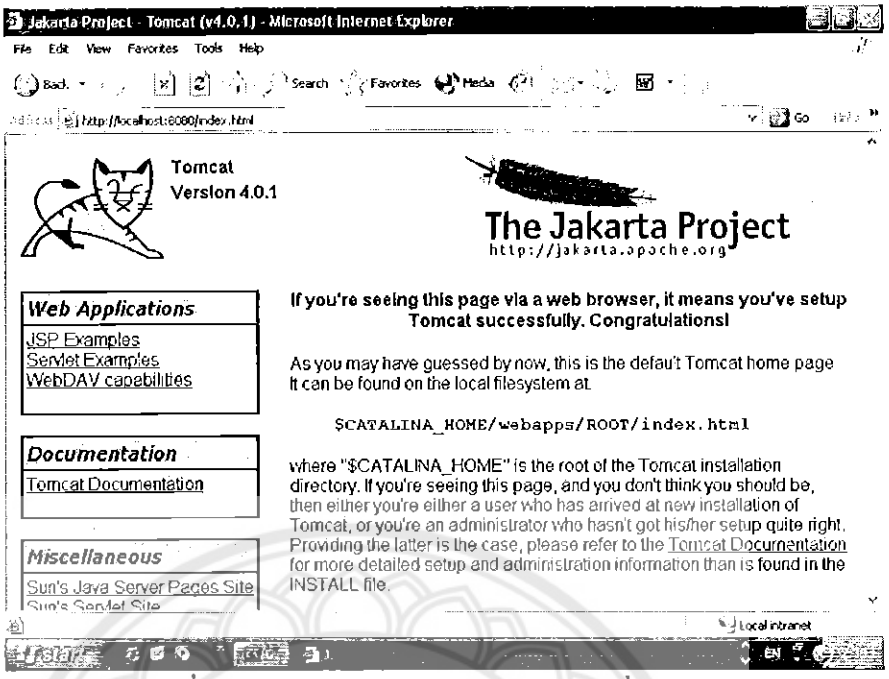
รูปที่ ก-7 แสดงการใช้คำสั่ง startup เพื่อเริ่มการทำงานของ Tomcat

ในขณะเดียวกัน ก็จะมีหน้าต่างดังรูปที่ ก-8 ปรากฏซ้อนออกมาอีกหน้าต่าง แสดงว่าเว็บเซิร์ฟเวอร์ Tomcat เริ่มต้นทำงานแล้ว



รูปที่ ก-8 แสดงหน้าต่างที่ pop up ขึ้นมาเมื่อใช้คำสั่ง startup

ขั้นตอนต่อไป คือทำการเปิดเบราเซอร์ขึ้นมาเพื่อทดสอบการทำงาน โดยพิมพ์ http://localhost:8080 ลงไปที่ช่องกรอก URL ในเบราเซอร์ ถ้าได้ผลดังรูปที่ ก-9 แสดงว่าการติดตั้ง Tomcat เสร็จสมบูรณ์เรียบร้อยแล้ว



รูปที่ ก-9 แสดงโฮมเพจ Tomcat ภายในเครื่องของเรา

ซึ่ง localhost นี้เป็นคำที่ใช้เรียกแทนชื่อ โอสต์ ซึ่งหมายถึงเครื่องพีซีของเราที่ติดตั้ง Tomcat และรันสคริปต์ JSP ในเครื่องเดียวกัน นอกจากคำว่า localhost แล้วเราอาจจะระบุเป็นหมายเลข IP Address แทนก็ได้โดย IP Address ที่หมายถึง localhost ได้แก่ 127.0.0.1

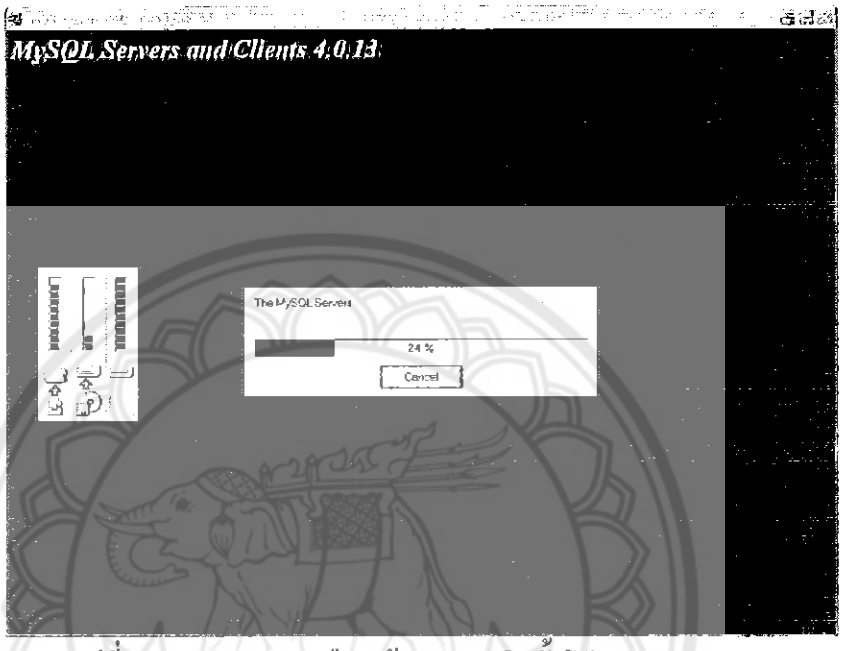
สำหรับ :8080 ที่ระบุต่อท้ายนั้นเป็นหมายเลขพอร์ตที่เซิร์ฟเวอร์ใช้อยู่ โดยการติดตั้งทั้งตัว J2SE และ Tomcat นี้ก็สามารถทำการพัฒนาโปรแกรมได้แล้วเหลือเพียงแค่ฐานข้อมูลที่จะใช้ทำการติดต่อกับเว็บเซิร์ฟเวอร์ ซึ่งต่อไปจะอธิบายขั้นตอนการติดตั้ง MySQL ที่ใช้เป็นฐานข้อมูลในโครงงานนี้

2.3 ขั้นตอนการติดตั้งฐานข้อมูล MySQL

โดยฐานข้อมูลที่เป็น MySQL นี้จะเป็นฐานข้อมูลที่ใช้เก็บชื่อไฟล์ของรูปภาพที่เป็น Key-frame และทำการเก็บค่าฮิสโตแกรมของ Key-frame นี้ ซึ่งในการประมวลผลเพื่อหาค่าฮิสโตแกรมของทุก Key-frame ในฐานข้อมูลนี้ ได้ทำการนำค่าฮิสโตแกรมที่ได้ไปเก็บไว้ในฐานข้อมูลที่เป็น Access ก่อน จากนั้นเมื่อจะนำฐานข้อมูลนี้มาประมวลผลจึงค่อยทำการ import ฐานข้อมูลที่เป็น Access เข้ามายัง MySQL ซึ่งต่อไปจะขออธิบายการติดตั้ง MySQL

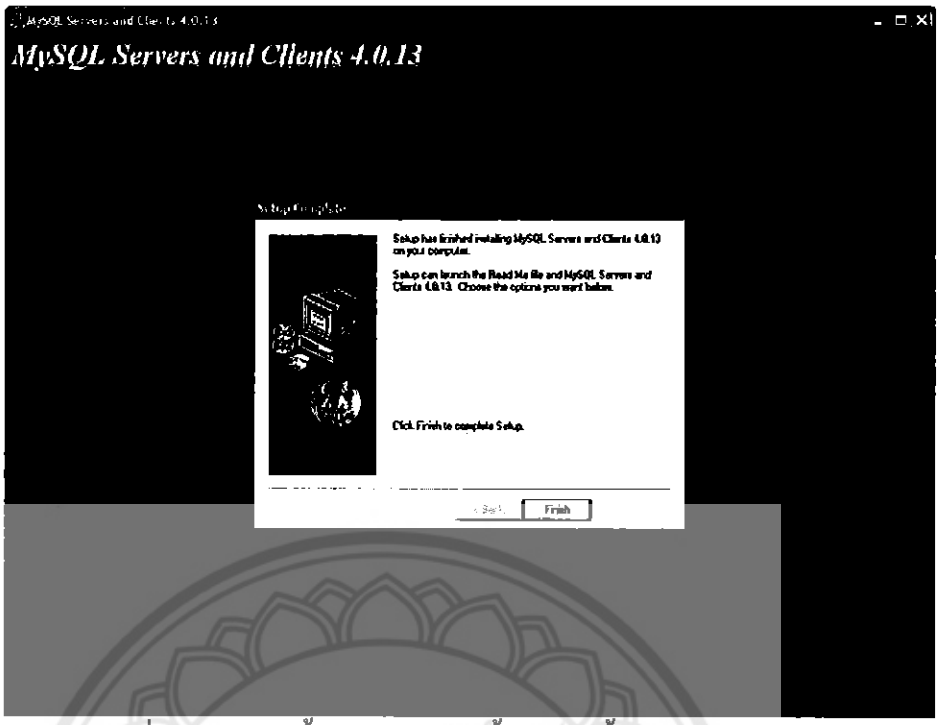
เมื่อได้ทำการดาวน์โหลดโปรแกรม MySQL แล้วให้ unzip ไฟล์ mysql-4.0.13-win.zip ออกมาจะได้ไฟล์ต่าง ๆ จำนวน 10 กว่าไฟล์ เมื่อดับเบิ้ลคลิ๊กที่ไฟล์ setup.exe โปรแกรมติดตั้งจะดำเนินไปตามขั้นตอนเหมือนกับการติดตั้งโปรแกรมทั่วไปในระบบ Windows คือ เริ่มต้นจากหน้าต่าง Welcome แสดงข้อความต้อนรับ คำแนะนำ และคำเตือนเบื้องต้น, หน้าต่าง Information

แสดงคำแนะนำเพิ่มเติม, หน้าต่าง Choose Destination Location ให้เลือกไดเรกทอรีที่จะติดตั้ง MySQL ในที่นี้ได้ยึดตามค่าดีฟอลต์คือ C:\mysql, หน้าต่าง Setup Type ให้เลือกลักษณะการติดตั้ง ในที่นี้ให้เลือกเป็น Typical ซึ่งเป็นการติดตั้งแบบทั่วไป ในที่สุด โปรแกรมจะเริ่มต้นการติดตั้ง และแสดงเปอร์เซ็นต์ความคืบหน้าดังรูปที่ ก-10



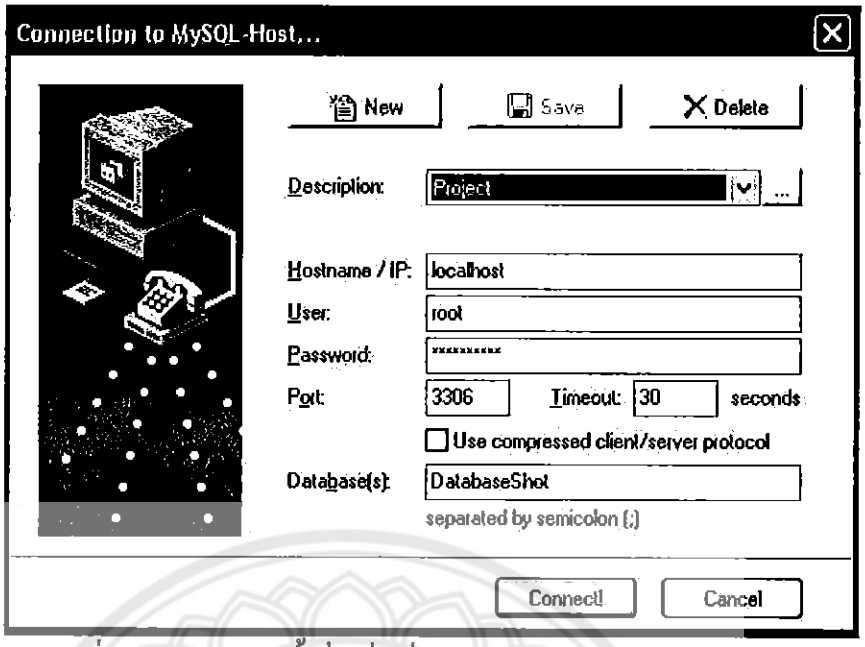
รูปที่ ก-10 แสดงความคืบหน้าของการติดตั้งโปรแกรม MySQL

หลังจากติดตั้งเสร็จเรียบร้อยแล้ว หน้าต่าง Setup Complete จะปรากฏออกมาดังรูปที่ ก-11 ซึ่งเมื่อคลิ๊ก finish แล้วก็จะเสร็จสิ้นการติดตั้งฐานข้อมูล MySQL ต่อไปเราจะทำการติดตั้ง MySQL-Front เพื่อความสะดวกในการใช้ทำงานกับ MySQL โดยโปรแกรม MySQL-Front นี้เป็นเครื่องมือที่ช่วยเอื้ออำนวยในความสะดวกในการใช้งานระบบฐานข้อมูล MySQL ซึ่งโปรแกรมนี้สามารถดาวน์โหลดมาใช้งานได้ฟรีจาก http://www.anse.de/mysqlfront/ ซึ่งโปรแกรม MySQL-front ที่ใช้ในการพัฒนานี้ได้ใช้เวอร์ชัน 2.4 ชื่อว่า MySQL-Front_2.4_Setup.exe ซึ่งมีขนาดไม่ถึง 1 MB หลังจากได้มาแล้วก็ดับเบิ้ลคลิ๊กเพื่อติดตั้งใช้งานได้ทันทีเหมือนกับโปรแกรมทั่ว ๆ ไปที่รันในระบบ Windows



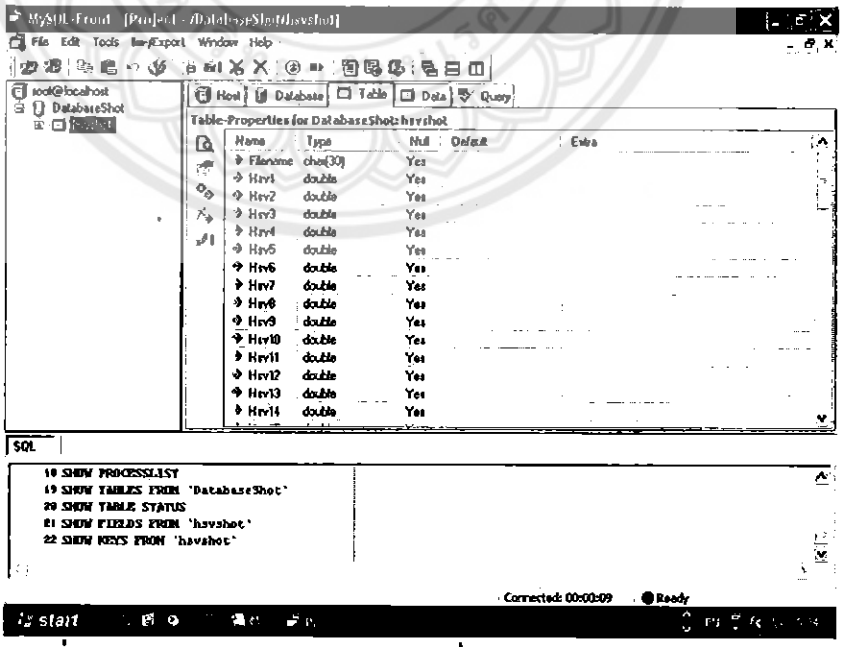
รูปที่ ก-11 แสดงขั้นตอนการเสร็จสิ้นการติดตั้งโปรแกรม MySQL

ซึ่งพอติดตั้งโปรแกรม MySQL-Front เรียบร้อยแล้ว จะปรากฏไอคอนที่เป็น shortcut ของโปรแกรมที่หน้าจอ Destop เมื่อดับเบิ้ลคลิ๊กที่ shortcut แล้ว โปรแกรมจะทำงานโดยแสดงหน้าต่างขึ้นมา โดยต้องทำการคลิ๊กที่ปุ่ม New ก่อนเพื่อตั้งชื่อการเชื่อมติดต่อฐานข้อมูลโดยส่วนสำคัญของการตั้งชื่อเพื่อเริ่มการเชื่อมต่อกับฐานข้อมูลนี้ส่วนของ password จะมีความสำคัญมากเพราะเป็นการป้องกันฐานข้อมูลของระบบไม่ให้ผู้อื่นทำการละเมิดแก้ไขหรือปรับเปลี่ยนข้อมูลได้ ส่วนใหญ่จึงควรกำหนด password ทุกครั้งในการติดต่อกับ MySQL โดยผู้ใช้หรือผู้พัฒนาสามารถทำการเปลี่ยนแปลง password ได้ ซึ่งในขั้นตอนการตั้งชื่อเพื่อเชื่อมการติดต่อกับฐานข้อมูลนี้เองควรใส่ password ให้ตรงกับ password ที่ได้ทำการเปลี่ยนแปลงใน MySQL ด้วยโดยแสดงการตั้งชื่อเพื่อเชื่อมการติดต่อกับฐานข้อมูลดังรูปที่ ก-12



รูปที่ ก-12 แสดงการตั้งชื่อเพื่อเชื่อมการติดต่อกับฐานข้อมูล MySQL

จากนั้นคลิ๊กที่ปุ่ม Connect! เพื่อเชื่อมการติดต่อ โดยจะแสดงโครงสร้างของฐานข้อมูลที่เป็น ตาราง และฟิลด์หรือคอลัมน์ต่าง ๆ ซึ่งคล้ายกับโครงสร้างของไดเรกทอรีในระบบ Windows นั่นเอง และเมื่อคลิ๊กที่ตารางทางฝั่งซ้ายมือ จะพบรายละเอียดเกี่ยวกับฟิลด์ทุกฟิลด์ในตารางนี้ก็จะแสดงออกมาที่บริเวณพื้นที่ตรงกลางของหน้าต่างตามรูปที่ ก-13

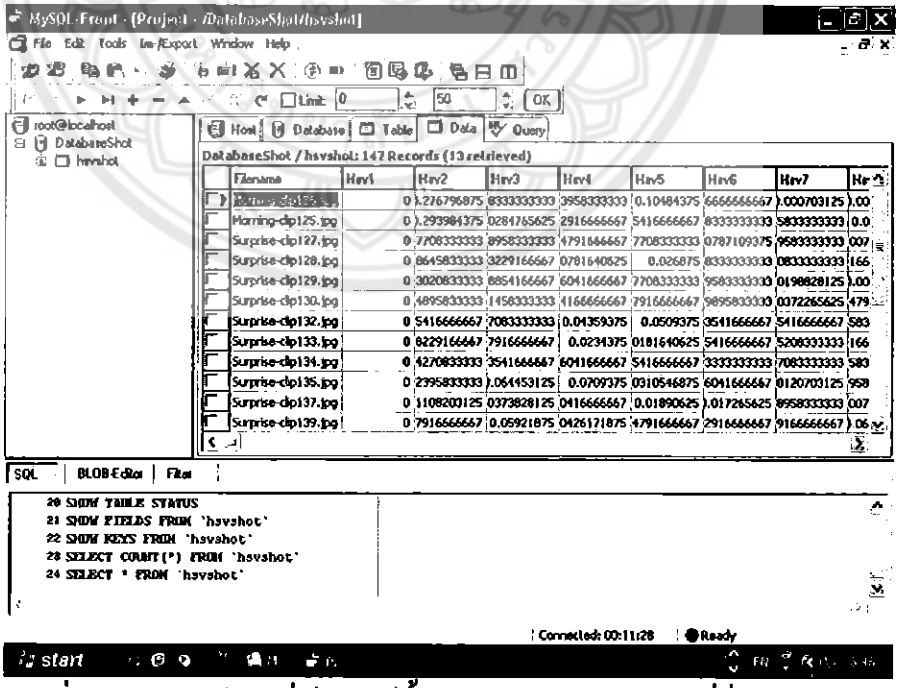


รูปที่ ก-13 แสดงข้อมูลตารางและฟิลด์ที่มีอยู่ในฐานข้อมูล DatabaseShot

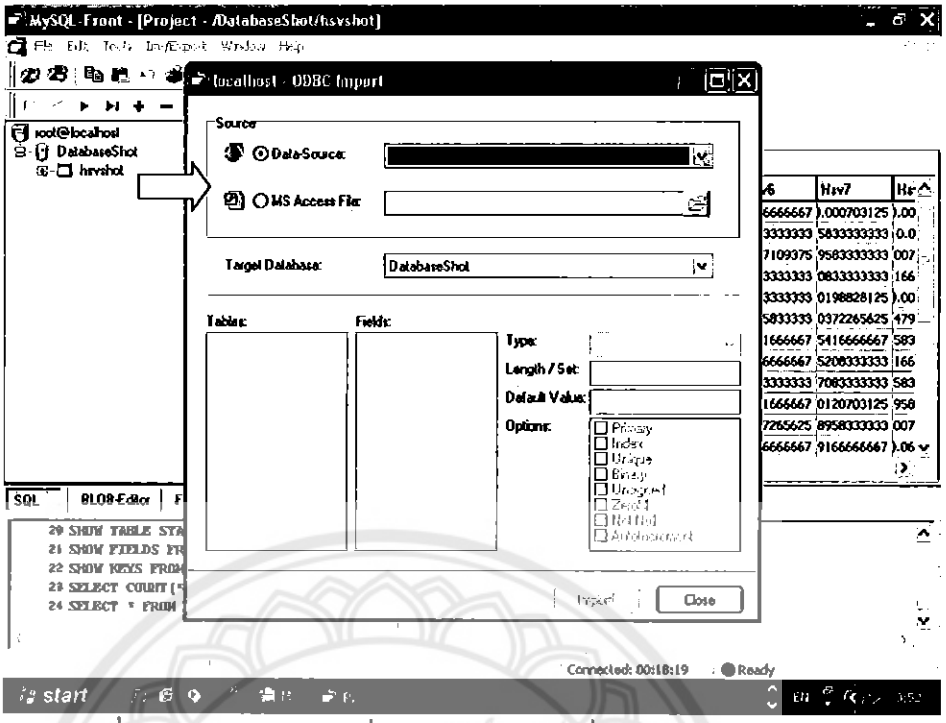
ซึ่งจากขั้นตอนนี้เองทำให้สามารถติดต่อกับฐานข้อมูลที่เป็น MySQL ได้เรียบร้อยแล้ว ซึ่งในส่วนของโปรแกรมการทำงานเกี่ยวกับฐานข้อมูลข้างต้นเราได้ทำการเก็บข้อมูลเกี่ยวกับชื่อไฟล์ที่เป็นรูปภาพของ Key-frame และค่าฮิสโตแกรมของ Key-frame นั้นไว้ในฐานข้อมูลที่เป็น Access ของโปรแกรม Microsoft Access จากนั้นค่อย import ค่าของฐานข้อมูลที่เป็น Access มาไว้ในฐานข้อมูลที่เป็น MySQL ขั้นตอนต่อไปจะเป็นการย้ายฐานข้อมูลจาก Access มาไว้ที่ฐานข้อมูลที่เป็น MySQL

ซึ่งในขั้นตอนนี้เราได้ทำการสร้างตารางฐานข้อมูลชื่อ Hsvshot ขึ้นมาแล้วโดยใช้โปรแกรม Java ในการสร้างตารางฐานข้อมูลที่เป็น MySQL ขึ้นมาโดยมีฟิลด์ของข้อมูลดังนี้คือ ในฟิลด์ช่องแรกนั้นจะเก็บชื่อไฟล์ของรูปภาพที่เป็น Key-frame เช่น Morning-clip27.jpg และในฟิลด์ช่องที่สองนี้จะเก็บค่าฮิสโตแกรมใน bin ที่ 1 ของ Key-frame นี้ไว้และทำการเพิ่มค่าในช่องฟิลด์อย่างนี้ไปเรื่อย ๆ จนถึงค่าฮิสโตแกรมใน bin ที่ 64

โดยแสดงข้อมูลของฟิลด์ทั้งหมดดังรูปที่ ก-14 และจากนั้นเราจะทำการ import ค่าจากฐานข้อมูลที่เป็น Access มาเก็บไว้ในฐานข้อมูลที่เป็น MySQL เพื่อนำไปใช้ในการทำเว็บเซิร์ฟเวอร์ต่อไป ซึ่งจะทำให้การค้นหาข้อมูลนั้นมีความรวดเร็วมากขึ้นและง่ายต่อการเรียกใช้ โดยวิธีการ import นั้นให้คลิกที่แท็บเมนู Im-/Export จากนั้นเลือกคลิกชื่อ ODBC Import ซึ่งเป็นเมนูย่อยของเมนู Im-/Export ซึ่งจะปรากฏหน้าต่างดังรูปที่ ก-15

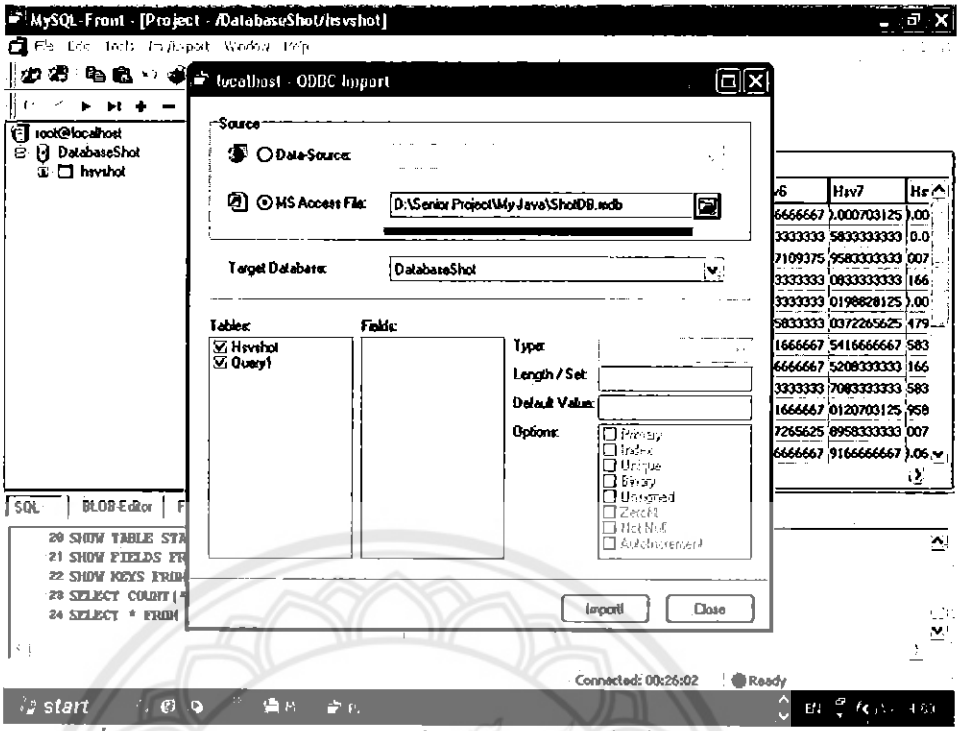


รูปที่ ก-14 แสดงข้อมูลที่เป็นฟิลด์ทั้งหมดในตารางฐานข้อมูลที่ชื่อว่า Hsvshot



รูปที่ ก-15 แสดงหน้าต่างที่ใช้เลือกฐานข้อมูลที่จะ import มายัง MySQL

จากนั้นคลิ๊กที่ Ratio Box ที่ชื่อว่า Ms Access File เพื่อเลือกฐานข้อมูล Access ที่จะ import มายังฐานข้อมูลของ MySQL ซึ่งเมื่อทำการคลิ๊กแล้วจะให้คลิ๊กช่องว่างทางขวามือเพื่อเลือกไฟล์ที่ต้องการจะ import เข้ามา โดยเมื่อทำการเลือกไฟล์ของฐานข้อมูลที่เป็น Access ได้แล้วจะแสดงพาธที่อยู่ของข้อมูลและชื่อไฟล์ในช่องทางขวามือของ Ratio Box โดยแสดงดังรูปที่ ก-16



รูปที่ ก-16 แสดงพาธของข้อมูลที่ได้ทำการเลือกเพื่อที่จะ import ลง MySQL

โดยจากรูปที่ ก-16 จะปรากฏทางด้านซ้ายมือล่างมีชื่อของตารางฐานข้อมูลปรากฏ จากนั้นทำการเลือกตารางฐานข้อมูลที่จะ import แล้วคลิ๊กที่ปุ่ม Import! ก็จะเป็นการเสร็จสิ้นการย้ายฐานข้อมูลจาก Access มาเก็บไว้ที่ MySQL และเมื่อผู้พัฒนาจะเรียกใช้ข้อมูลจากฐานข้อมูล ก็ทำการติดต่อกับฐานข้อมูล MySQL เพียงอย่างเดียว

Source Code JAVA and JSP

ในการทำโครงงานนี้ได้มีการแบ่งส่วนของการทำงานออกเป็น 2 ส่วนใหญ่ ๆ ดังนี้

1. ซอร์ดโค้ดที่ใช้ JAVA ในการเขียนโปรแกรม
    - 1.1 **CreateTable.java** ใช้ในการสร้างตารางฐานข้อมูลที่เป็น MySQL
    - 1.2 **Histogram.java** ใช้ในการหาค่า Histogram ของรูปภาพที่เป็น Key-frame แล้วนำค่าผลลัพธ์ที่ได้บันทึกลงในฐานข้อมูล
    - 1.3 **Distance.java** ใช้ในการหาค่าความแตกต่างของภาพ ซึ่งซอร์ดโค้ดตัวนี้จะนำไปใช้ร่วมกับซอร์ดโค้ดที่ใช้ JSP ในการเขียนโปรแกรมที่ทำงานบนเว็บไซต์
    - 1.4 **Feedback.java** ใช้ในส่วนของการป้อนกลับให้โปรแกรมนำไปประมวลผลอีกครั้ง
2. ซอร์ดโค้ดที่ใช้ JSP ในการเขียนโปรแกรมที่ทำงานบนเว็บไซต์
    - 2.1 **Main.jsp** ส่วนของเว็บไซต์ที่เป็น Header
    - 2.2 **index.jsp** หน้าเว็บไซต์ครั้งแรกที่เปิดใช้งาน
    - 2.3 **Keyframe.jsp** หน้าเว็บไซต์ที่แสดงรูปภาพที่เป็น Key-frame
    - 2.4 **Search.jsp** หน้าเว็บไซต์ที่แสดงผลลัพธ์จากการค้นหาครั้งแรก (Simple Search)
    - 2.5 **Feedback.jsp** หน้าเว็บไซต์ที่แสดงผลลัพธ์ที่ได้จากการป้อนกลับของข้อมูล

โดยในลำดับต่อไปจะแสดงซอร์ดโค้ดที่ได้เขียนใช้งานในการทำโครงงานนี้ตามลำดับ

```
1  import java.sql.*;
2
3  public class CreateTable
4  {
5     public static void main(String[] args) {
6        try {
7           Class.forName("org.gjt.mm.mysql.Driver");
8        }catch (ClassNotFoundException ex){
9           System.out.println(ex.getMessage());
10       }
11       try {
12       String url = "jdbc:mysql://localhost/DatabaseShot?user=root&password=intervirus";
13       String str1 = "(`Filename` CHAR (30), ";
14       String str2 = null;
15       String sql = null;
16       for (int i=1; i<65; i++)
17       {
18          if (i == 1) {
19             str2 = "`Hsv"+i+"` DOUBLE, ";
20             sql = str1 + str2;
21          } else {
22             if (i == 64) {
23                str2 = "`Hsv"+i+"` DOUBLE)";
24                sql = sql + str2;
25             } else {
26                str2 = "`Hsv"+i+"` DOUBLE, ";
27                sql = sql + str2;
28       } } }
29       sql = "CREATE TABLE `Hsvshot` "+sql;
30       Connection con = DriverManager.getConnection(url);
31       Statement stmt1 = con.createStatement();
32       ResultSet rs1 = stmt1.executeQuery(sql);
33       } catch (SQLException ex) {
34          System.out.println(ex.getMessage());
35          System.out.println("Error");
36 } } }
```

```
1  import java.awt.*;
2  import java.awt.event.*;
3  import javax.swing.*;
4  import java.applet.*;
5  import java.sql.*;
6  import java.lang.*;
7  import java.io.*;
8  import java.awt.image.*;
9
10 class Histogram extends JApplet
11 {
12    ///////// ประกาศตัวแปร /////////
13    Image i = null;
14    Image img = null;
15    Image iicon = null;
16    int x, y;
17
18    public static void main(String[] args)
19    {
20       Histogram ht = new Histogram();
21       File d = new File("./image");
22       String n[] = d.list();
23       for (int i = 0; i < n.length; i++) {
24          // Path ที่ใช้เก็บรูปภาพที่เป็น key-frame //
25          String str = "D:/Senior Project/Histogram/image/"+n[i];
26          System.out.println(str);
27          Image img = ht.getImg(str);
28          int w = ht.getImgWidth(img);
29          int h = ht.getImgHeight(img);
30          ht.getImgRGB(w, h, img, n[i]);
31       }
32       System.exit(0);
33    }
34    public static Image getImg(String str)
35    {
36       Image i;
37       ImageIcon iicon = new ImageIcon(str);
38       i = iicon.getImage();
```

```
39        return i;
40    }
41    public int getImgWidth(Image i)
42    {
43        int w = i.getWidth(this);
44        return w;
45    }
46    public int getImgHeight(Image i)
47    {
48        int h = i.getHeight(this);
49        return h;
50    }
51    public void getImgRGB(int x, int y, Image img, String str)
52    {
53        int wh; // ตัวแปรที่ใช้เก็บจำนวนพิกเซลทั้งหมดของรูปภาพที่นำมาประมวลผล //
54        int r, g, b; // red, green, blue //
55        double []his = new double[70]; // array ที่ใช้เก็บค่า Histogram //
56        wh = (x*y);
57        for (int count=0; count <64; count++ ) {
58            // เริ่มต้นที่ his[0] จนถึง his[63] มี array his จำนวน 64 ช่อง //
59            his[count] = 0; // Clear ค่า Histogram ในตัวแปร //
60        }
61        int []p = new int[wh]; // สร้าง Array ที่มีขนาดเท่ากับรูป เป็น Array มิติเดียว //
62        PixelGrabber pg = new PixelGrabber(img, 0, 0, x, y, p, 0, x);
63        try {
64            pg.grabPixels();
65        } catch (Exception e) {}
66        System.out.println("Pixel All"+"\t\t"+wh); // ตรวจสอบจำนวนพิกเซลของรูปภาพ //
67        for (int count = 0; count < wh; count++) {
68            r = 0xff & (p[count] >> 16);
69            g = 0xff & (p[count] >> 8);
70            b = 0xff & p[count] ;
71            float hsb[] = Color.RGBtoHSB(r,g,b,null);
72            double h = hsb[0];
73            double s = hsb[1];
74            double v = hsb[2];
75            // End of Get Value HSV from Pixel in Image //
76            int ans = 0;
```

```
77          int oak = 0;
78          int nest = 0;
79          int jack = 0;
80          // วน Loop เพื่อ count Histogram //
81          // ส่วนของค่า Hue //
82          if (h>=0 && h<=0.125) {
83             oak = 1;
84          } else if (h>0.125 && h<=0.25) {
85             oak = 2;
86          } else if (h>0.25 && h<=0.375) {
87             oak = 3;
88          } else if (h>0.375 && h<=0.5) {
89             oak = 4;
90          } else if (h>0.5 && h<=0.625) {
91             oak = 5;
92          } else if (h>0.625 && h<=0.75) {
93             oak = 6;
94          } else if (h>0.75 && h<=0.875) {
95             oak = 7;
96          } else if (h>0.875 && h<=1) {
97             oak = 8;
98          }
99          // ส่วนของค่า Saturation //
100         if (s >=0 && s <= 0.25) {
101            nest=1;
102         } else if (s >0.25 && s <= 0.5) {
103            nest=2;
104         } else if (s > 0.5 && s <= 0.75) {
105            nest=3;
106         } else if (s > 0.75 && s<= 1) {
107            nest=4;
108         }
109         // ส่วนของค่า Value //
110         if (v >= 0 && v<=0.5) {
111            jack=0;
112         } else if (v >0.5 && v <= 1) {
113            jack=32;
114         }
```

```
115         // Count Histogram //
116         ans = (int)(((oak*4)-(4-nest)+jack));
117         his[ans] = his[ans] +1;
118       }
119       // สร้างคำสั่งวนลูป //
120       String txtSQL, txtSQL2;
121       txtSQL = ("Insert Into Hsvshot(Filename,");
122       txtSQL2 = (" Values(" + "?" +",");
123       int j = 1;
124       for (int i=2; i<66; i++) {
125         if (i < 65) {
126           txtSQL = (txtSQL + "Hsv" + j + ",");
127           j = j +1;
128           txtSQL2 = (txtSQL2 + "?" + ",");
129         } else if (i == 65) {
130           txtSQL = (txtSQL + "Hsv" + j + ")");
131           txtSQL2 = (txtSQL2 + "?" );
132         }
133       }
134       txtSQL2 = txtSQL2 + ")";
135       txtSQL = txtSQL + txtSQL2 ;
136       ////////////////// ส่วนที่ติดต่อกับฐานข้อมูลเพื่อบันทึกค่า Histogram //////////////////
137       try {
138         Class.forName("org.gjt.mm.mysql.Driver"); // ใช้สำหรับ MYSQL //
139         //Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); // ใช้สำหรับ MS-Access //
140       } catch (ClassNotFoundException ex) {
141         System.out.println(ex.getMessage());
142       }
143       try {
144         String url = "jdbc:mysql://localhost:3306/DatabaseShot?user=root&password=intervirus";
145         //String url = "jdbc:odbc:DatabaseShot"; // ใช้สำหรับ MS-Access //
146         Connection con = DriverManager.getConnection(url);
147         PreparedStatement stmt1 = con.prepareStatement(txtSQL);
148         stmt1.setString(1, str);
149         int l = 0;
150         for (int k=2; k<66; k++) {
151           stmt1.setDouble(k, (his[l])/wh);
152           l = l +1;
```

```
153             }
154             stmt1.executeUpdate();
155             con.close();
156         } catch (SQLException ex) {
157             System.out.println(ex.getMessage());
158         }
159     }
160 }
```

```
1  package mybean;
2  import java.sql.*;
3  public class Distance implements java.io.Serializable
4  {
5    ////////////// ประกาศตัวแปร //////////////
6    private int totalrow, count;
7    private double[] distancevalue;
8    private double hsvindex[] = new double[65];
9    private String[] filename;
10   private String sql, fileindex;
11   private Statement stmt;
12   private ResultSet result;
13
14   public void setHsvindex() throws ClassNotFoundException, SQLException
15   {
16     sql = "SELECT * FROM Hsvshot WHERE Filename='"+"'"+fileindex+"'";
17     result = setSQL(sql);
18     if (result.next()) {
19       for (int i=1; i<65; i++) {
20         hsvindex[i] = result.getDouble("Hsv"+i);
21       }
22     }
23     result.close();
24   }
25   public void setDistance() throws ClassNotFoundException, SQLException
26   {
27     count = 0;
28     double sum = 0, ans = 0;
29     sql = "SELECT * FROM Hsvshot";
30     result = setSQL(sql);
31     while (result.next()) {
32       count = count + 1;
33     }
34     totalrow = count;
35     String file[] = new String[totalrow];
36     double distance[] = new double[totalrow];
37     count = 0;
38     result.first();
```

```
39          while (!result.isAfterLast()) {
40            for (int i=1; i<65; i++) {
41              sum = (1*(Math.pow((hsvindex[i] - result.getDouble("Hsv"+i)),2))) + sum;
42            }
43            ans = Math.sqrt(sum);
44            distance[count] = ans;
45            file[count] = result.getString("Filename");
46            count = count + 1;
47            sum = 0;
48            result.next();
49          }
50          filename = file;
51          distancevalue = distance;
52          result.close();
53      }
54      public ResultSet setSQL(String sql) throws ClassNotFoundException, SQLException
55      {
56          this.sql = sql;
57          String url = "jdbc:mysql://localhost:3306/DatabaseShot?user=root&password=intervirus";
58          Class.forName("org.gjt.mm.mysql.Driver");
59          Connection conn = DriverManager.getConnection(url);
60          stmt = conn.createStatement();
61          result = stmt.executeQuery(sql);
62          return result;
63      }
64      public void Bubblesort()
65      {
66          double dist; String tidus;
67          for (int x=0; x<(totalrow-1); x++) {
68            for (int y=0; y < (totalrow-1) - x ; y++) {
69              if (distancevalue[y] > distancevalue[y+1] ) {
70                dist = distancevalue[y];
71                tidus = filename[y];
72                distancevalue[y] = distancevalue[y+1];
73                filename[y] = filename[y+1];
74                distancevalue[y+1] = dist;
75                filename[y+1] = tidus;
76      } } } }
```

```
77      ////////////// Method Set //////////////
78      public void setFileindex(String fileindex) {
79          this.fileindex = fileindex;
80      }
81      public void setDistancevalue(double[] Distancevalue) {
82          this.distancevalue = distancevalue;
83      }
84      public void setFilename(String[] filename) {
85          this.filename = filename;
86      }
87      ////////////// Method Get //////////////
88      public double[] getHsvindex() {
89          return hsvindex;
90      }
91      public double[] getDistancevalue() {
92          return distancevalue;
93      }
94      public String getFileindex() {
95          return fileindex;
96      }
97      public String[] getFilename() {
98          return filename;
99      }
100 }
```

```
1  package mybean;
2  import java.sql.*;
3  public class Feedback implements java.io.Serializable
4  {
5    ////////////////// ประกาศตัวแปร //////////////////
6    private ResultSet result;
7    private String sql;
8    private String[] frameselect;
9    private String[] weightstring;
10   private int manypicture, totalrow;
11   private double sum, sumbar, weight, sumweight, sumweightbar, ans;
12   private double arrayweight[] = new double[65];
13   private double hsvindex[] = new double[65];
14   private double[] weightdouble;
15   ////////////////// Method ที่ใช้ในการประมวลผลของ Class นี้ //////////////////
16   public void setArrayweight() throws ClassNotFoundException, SQLException
17   {
18     double temp[] = new double[manypicture];
19     for (int i=1; i<65; i++) {
20       int many;
21       sum = 0; sumbar = 0; weight = 0; sumweight = 0; many = 0;
22       for (int j=0; j<manypicture; j++) {
23         sql = "SELECT * FROM Hsvshot WHERE Filename="+"'"+frameselect[j]+"'";
24         result = setSQL(sql);
25         result.first();
26         temp[j] = result.getDouble("Hsv"+i);
27         sum = sum + temp[j];
28         many = many + 1;
29         if (many == manypicture) {
30           sumbar = sum / manypicture;
31           for (int k=0; k<manypicture; k++) {
32             weight = Math.pow((temp[k] - sumbar), 2);
33             sumweight = sumweight + weight;
34           }
35           sumweightbar = sumweight / manypicture;
36           if (sumweightbar == 0) {
37             arrayweight[i] = 0;
38           }else {
```

```
39                  arrayweight[i] = 1 / (Math.sqrt(sumweightbar));
40              }
41              sumbar = 0; sumweight = 0; sumweightbar = 0;
42              sum = 0; weight = 0; many = 0;
43      } } }
44      result.close();
45  }
46  public void setWeighteuclidean(double[] hsvindex, int totalrow) throws ClassNotFoundException, SQLExcep
47  {
48      this.hsvindex = hsvindex;
49      this.totalrow = totalrow;
50      sql = "SELECT * FROM Hsvshot";
51      result = setSQL(sql);
52      int count = 0;
53      String weightname[] = new String[totalrow];
54      double weightvalue[] = new double[totalrow];
55      sum = 0; ans = 0;
56      result.first();
57      while (!result.isAfterLast()) {
58        for (int i=1; i<65; i++) {
59          sum = sum + (arrayweight[i]*(Math.pow((hsvindex[i] - result.getDouble("Hsv"+i)),2)));
60        }
61        ans = Math.sqrt(sum);
62        weightvalue[count] = ans;
63        weightname[count] = result.getString("Filename");
64        count = count + 1;
65        sum = 0;
66        result.next();
67      }
68      weightdouble = weightvalue;
69      weightstring = weightname;
70      result.close();
71  }
72  public ResultSet setSQL(String sql) throws ClassNotFoundException, SQLException
73  {
74      this.sql = sql;
75      String url = "jdbc:mysql://localhost:3306/DatabaseShot?user=root&password=intervirus";
76      Class.forName("org.gjt.mm.mysql.Driver");
```

```
77        Connection conn = DriverManager.getConnection(url);
78        Statement stmt = conn.createStatement();
79        result = stmt.executeQuery(sql);
80        return result;
81     }
82     public void setBubblesort(String[] weightstring, double[] weightdouble)
83     {
84        double weight; String text;
85        for (int x=0; x<(totalrow-1); x++) {
86           for (int y=0; y < (totalrow-1) - x ; y++) {
87              if (weightdouble[y] > weightdouble[y+1] ) {
88                 weight = weightdouble[y];
89                 text = weightstring[y];
90                 weightdouble[y] = weightdouble[y+1];
91                 weightstring[y] = weightstring[y+1];
92                 weightdouble[y+1] = weight;
93                 weightstring[y+1] = text;
94     } } } }
95     ////////////////// Method SET //////////////////
96     public void setFrameselect(String[] frameselect) {
97        this.frameselect = frameselect;
98     }
99     public void setManypicture(String[] frameselect) {
100       this.frameselect = frameselect;
101       manypicture = frameselect.length;
102    }
103    public void setHsvindex(String fileindex) throws ClassNotFoundException, SQLException {
104       sql = "SELECT * FROM Hsvshot WHERE Filename='"+"'"+fileindex+"'";
105       result = setSQL(sql);
106       if (result.next()) {
107          for (int i=1; i<65; i++) {
108             hsvindex[i] = result.getDouble("Hsv"+i);
109       } }
110       result.close();
111    }
112    public void setTotalrow() throws ClassNotFoundException, SQLException {
113       sql = "SELECT * FROM Hsvshot";
114       result = setSQL(sql);
```

```
115         while(result.next()) {
116             totalrow = totalrow + 1;
117         }
118         result.close();
119     }
120     ////////////////// Method GET //////////////////
121     public String[] getWeightstring() {
122         return weightstring;
123     }
124     public double[] getWeightdouble() {
125         return weightdouble;
126     }
127     public double[] getHsvindex() {
128         return hsvindex;
129     }
130     public int getManypicture() {
131         return manypicture;
132     }
133     public double[] getArrayweight() {
134         return arrayweight;
135     }
136     public int getTotalrow() {
137         return totalrow;
138     }
139 }
```

```
1  <html>
2  <head>
3  <title>:: Main ::</title>
4  </head>
5  <body>
6  <table width="100%" border=0 cellPadding=0 cellSpacing=0>
7    <tbody>
8      <tr>
9        <td height=30 bgColor=#99CCCC vAlign=middle align=center rowspan=4>
10         <font face=ms-sans-serif size=4 color=#CC0000>Video Retrieval Using Key-Frame Video Indexing
           Technique And Relevance Feedback</font>
11       </td>
12     </tr>
13   </tbody>
14 </table>
15 <table width="100%" border=0 cellPadding=0 cellSpacing=0>
16   <tbody>
17     <tr>
18       <td height=20 background="./Image/menubg1.jpg" align=right>
19         <font face=ms-sans-serif size=2>
20           <img src="./Image/red_folder.gif" border=0> 
21             <a class=mainmenu href="/index.jsp">Home</a>
22             | <img src="./Image/icon_true.gif" border=0> 
23             <a class=mainmenu href="/Keyframe.jsp">Key Frame</a>
24             | <img src="./Image/icon_search.gif" border=0> 
25             <a class=mainmenu href="about blank">Search</a>
26             | <img src="./Image/icon_folder.gif" border=0> 
27             <a class=mainmenu href="about blank">หลักการและทฤษฎี</a>
28             |<a class=mainmenu href="about blank">การทดลอง</a>
29             |<a class=mainmenu href="about blank">ขั้นตอนในการค้นหา</a>   
30         </font>
31       </td>
32     </tr>
33   </tbody>
34 </table>
35 </body>
36 </html>
```

```
1  <%@ page contentType="text/html; charset = TIS-620" %>
2  <jsp:useBean id="mybean" class="mybean.Distance" scope="application" />
3  <HTML>
4  <HEAD>
5  <TITLE>:: Index ::</TITLE>
6  </HEAD>
7  <BODY>
8  <%@ include file = 'Main.jsp' %>
9  <%
10    out.println("<br>");
11    out.println("<br>");
12    out.println("<center><img src=./Image/Back.jpg></img></center>");
13    out.println("<p><hr>");
14    out.println("<center><tr><td>");
15    out.println("<FONT face=ms-sans-serif size=4 color=#CC0000>");
16    out.println("<FONT face=ms-sans-serif size=5
      color=#CC0000>ขอต้อนรับ</FONT>เข้าสู่เว็บไซต์ที่ช่วยในการค้นหาข้อมูลดิจิตอลประเภทวิดีโอไฟล์ในระดับ
      Shot<br>");
17    out.println("</FONT>");
18    out.println("<FONT face=ms-sans-serif size=4 color=#0066FF>");
19    out.println("โดยใช้ดรรชนีเฟรมหลัก(Key-frame) และเทคนิคการป้อนกลับของข้อมูล(Relevance Feedback)");
20    out.println("</FONT>");
21    out.println("</td></tr></center>");
22 %>
23 </BODY>
24 </HTML>
```

```
1  <%@ page contentType="text/html; charset = TIS-620" %>
2  <jsp:useBean id="mybean" class="mybean.Distance" scope="application" />
3  <HTML>
4  <HEAD>
5  <TITLE>:: Keyframe ::</TITLE>
6  </HEAD>
7  <BODY>
8  <%@ include file = "Main.jsp" %>
9  <%@ page import="java.sql.*"%>
10 <%
11    int row_page = 24;
12    int total_row = 0;
13    String url, sql;
14    url = "jdbc:mysql://localhost:3306/DatabaseShot?user=root&password=intervirus";
15    Class.forName("org.gjt.mm.mysql.Driver");
16    Connection mycon = DriverManager.getConnection(url);
17    Statement stmt = mycon.createStatement();
18    sql = "Select count(*) as num from Hsvshot";
19    ResultSet myresult = stmt.executeQuery(sql);
20    while (myresult.next()) {
21       total_row = myresult.getInt("num");
22    }
23    myresult.close();
24    int total_page = (int)Math.ceil((double)total_row/(double)row_page);
25    int screen, start;
26    // ถ้าไม่มีการส่งตัวแปร screen มา แสดงว่าผู้ชมเปิดข้อมูลหน้านี้เป็นหน้าแรก
27    if (request.getParameter("screen")==null) {
28       screen = 1;
29    } else { // ถ้ามีการส่งตัวแปร screen มา  แสดงว่าผู้ชมต้องการดูข้อมูลหน้านั้น
30       screen = Integer.parseInt(request.getParameter("screen"));
31    }
32    if (screen >= 1 && screen <= total_page) {
33       start = (screen-1)*row_page;
34       sql = "Select Filename From Hsvshot Order By Hsv32 ASC ";
35       sql = sql + "Limit "+start+", "+row_page;
36       myresult = stmt.executeQuery(sql);
37       out.println("<form action =Search.jsp method=post>");
38       out.println("<br>");
```

```
39      out.println("<center><b>แสดง Key-frame ทั้งหมดใน Database</b></center>");
40      out.println("<br><table align=center bgcolor=#FFFFCC border=1 cellPadding=5 cellSpacing=0>");
41      //กำหนดค่าตัวแปรก่อนเรียกข้อมูลจากตาราง Hsvshot
42      int row = 0;
43      int countString = 0;
44      String temp[] = new String[6];
45      String Video[] = new String[6];
46      //ทำการดึงข้อมูล  แล้วแสดงข้อมูลในรูปตาราง
47      while (myresult.next()) {
48        String name = myresult.getString("Filename");
49        temp[row] = name;
50        countString = name.length();
51        Video[row] = name.substring(0, (countString-4));
52        countString = 0;
53        if (row == 0) {
54          out.println("<tr>");
55        }
56        //แสดงรูปภาพลงตาราง
57        out.println("<td align=center><img src=/Picture/"+name+" width=100></img></td>");
58        //เลื่อนช่อง Array
59        row = row + 1;
60        if (row == 6) {
61          out.println("</tr>");
62          out.println("<tr>");
63          for (int i=0; i<6; i++) {
64            out.println("<td><input type=radio name=frame value="+temp[i]+ "> ");
65            out.println("<a href=Video/"+Video[i]+".mpeg>"+Video[i]+"</a></td>");
66          }
67          out.println("</tr>");
68          row = 0;
69        }
70      }
71      myresult.close();
72      stmt.close();
73      mycon.close();
74      out.println("</table>");
75      out.println("<br>");
76      ////////////////////////////////////////////////////////////////////
```

```
77        // แสดงหมายเลขหน้าทั้งหมดเป็นลิงก์ไว้ให้คลิก
78        out.println("<center>");
79        if (screen>1) {
80           out.println("<a href='Keyframe.jsp?screen="+(screen-1)+"' ><b>ย้อนกลับ</b></a>");
81        }
82        for (int i=1; i<=total_page; i++) {
83           if (i==screen)
84              out.println(" <b>["+i+"]</b>");
85           else
86              out.println(" | <a href='Keyframe.jsp?screen="+i+"'>"+i+"</a> | ");
87        }
88        if (screen<total_page) {
89           out.println("<a href='Keyframe.jsp?screen="+(screen+1)+"' ><b>ถัดไป</b></a>");
90        }
91        out.println("</center>");
92        ////////////////////////////////////////////////////////////////////////
93        out.println("<p><hr>");
94        out.println("<br><input type = submit value = submit>");
95        out.println("<td>เลือก Key Frame ที่ต้องการแล้วกดที่ปุ่มนี้</td>");
96        out.println("</form>");
97     }
98     else
99        out.println("<center><b>ไม่พบข้อมูล</b></center>");
100 %>
101 </BODY>
102 </HTML>
```

```
38      String temp[] = new String[6];
39      String movie[] = new String[filename.length];
40      out.println("<form action =Feedback.jsp method=post>");
41      out.println("<br><table align=center border=1 cellPadding=5 cellSpacing=0 bgcolor=#FFFFCC>");
42      out.println("<tr bgcolor=#F4F7C8><td colspan=6 align=center><b><font face=CordiaUPC size=5
        color=#3399FF>Select Key-frame for Relevance Feedback</font></b></td></tr>");
43      for (int i=1; i<19; i++) {
44         countstring = filename[i].length();
45         temp[column] = filename[i];
46         movie[column] = filename[i].substring(0, (countstring-4));
47         countstring = 0;
48         if (column == 0) {
49            out.println("<tr>");
50         }
51         out.println("<td align=center><img src=Picture/"+filename[i]+" width=100>");
52         out.println("</img></td>");
53         column = column + 1;
54         if (column == 6) {
55            out.println("</tr>");
56            out.println("<tr>");
57            for (int x=0; x<6; x++) {
58               out.println("<td><input type=checkbox name=frame value="+temp[x]+ "> ");
59               out.println("<a href=Video/"+movie[x]+".mpeg>"+movie[x]+"</a></td>");
60            }
61            out.println("</tr>");
62            column = 0;
63         }
64      }
65      out.println("</table>");
66      out.println("<br>");
67      out.println("<input type = submit value = submit>");
68      out.println("<td>เลือก Key Frame ที่ต้องการแล้วกดที่ปุ่มนี้</td>");
69      out.println("</form>");
70   %>
71   </BODY>
72   </HTML>
```

```
1  <%@ page contentType = "text/html; charset = TIS-620" %>
2  <jsp:useBean id="mybean" class="mybean.Distance" scope="application" />
3  <HTML>
4  <HEAD>
5  <TITLE>:: Search ::</TITLE>
6  </HEAD>
7  <BODY>
8  <%@ include file = "Main.jsp" %>
9  <%@ page import="java.sql.*"%>
10 <%
11    double hsvindex[] = new double[65];
12    double distancevalue[] = new double[65];
13    String filename[] = new String[65];
14    String fileindex = request.getParameter("frame");
15    String movieindex;
16    int cntString;
17    cntString = fileindex.length();
18    movieindex = fileindex.substring(0, (cntString-4));
19    cntString = 0;
20    session.setAttribute("Memframe", fileindex); // ส่วนของ Session
21    mybean.setFileindex(fileindex);
22    mybean.setHsvindex();
23    mybean.setDistance();
24    mybean.Bubblesort();
25    filename = mybean.getFilename();
26 %>
27 <%
28    out.println("<br><center><FONT face=ms-sans-serif size=4 color=#336600>Key-frame
      Indexing</FONT></center>");
29    out.println("<br><center><tr><td>");
30    out.println("<img src=Picture/"+fileindex+" width=20% align=middle></img>");
31    out.println("</td></tr></center>");
32    out.println("<br><center>");
33    out.println("<a href=Video/"+movieindex+".mpeg>"+movieindex+"</a></td>");
34    out.println("</center><p><hr>");
35 %>
36 <%
37    int countstring = 0, column=0;
```

```
1  <%@ page contentType = "text/html; charset = TIS-620" %>
2  <jsp:useBean id="feedback" class="mybean.Feedback" />
3  <HTML>
4  <HEAD>
5  <TITLE>:: Feedback ::</TITLE>
6  </HEAD>
7  <BODY>
8  <%@ include file = "Main.jsp" %>
9  <%@ page import="java.sql.*"%>
10 <%
11    String fileindex = (String)session.getAttribute("Memframe");
12    int manystring = fileindex.length();
13    String videoindex = fileindex.substring(0, (manystring-4));
14    String[] frameselect = request.getParameterValues("frame");
15 %>
16 <%
17    out.println("<br><center><tr><td>");
18    out.println("<img src=Picture/"+fileindex+" width=20% align=middle></img>");
19    out.println("</td></tr></center>");
20    out.println("<br>");
21    out.println("<center>");
22    out.println("<a href=Video/"+videoindex+".mpeg>"+videoindex+"</a></td>");
23    out.println("</center>");
24    out.println("<p><hr>");
25 %>
26 <%
27    int totalrow;
28    double[] hsvindex;
29    double[] weightdouble;
30    String[] weightstring;
31    // ขั้นตอนการ Set Method ก่อนที่จะประมวลผล
32    feedback.setFrameselect(frameselect);
33    feedback.setManypicture(frameselect);
34    feedback.setTotalrow();
35    feedback.setHsvindex(fileIndex);
36    // ขั้นตอนของการ Get Method เพื่อนำไปใช้ใน Method อื่น
37    totalrow = feedback.getTotalrow();
38    hsvindex = feedback.getHsvindex();
```

```
39    // ขั้นตอนของการประมวลผล
40    feedback.setArrayweight();
41    feedback.setWeighteuclidean(hsvindex, totalrow);
42    weightstring = feedback.getWeightstring();
43    weightdouble = feedback.getWeightdouble();
44    feedback.setBubblesort(weightstring, weightdouble);
45    // ขั้นตอนของการนำผลลัพธ์ที่ได้ไปแสดงผล
46    weightstring = feedback.getWeightstring();
47  %>
48  <%
49    int string = 0, column=0;
50    String temp[] = new String[6];
51    String movie[] = new String[weightstring.length];
52    out.println("<form action =Feedback.jsp method=post>");
53    out.println("<br><table align=center border=1 cellPadding=5 cellSpacing=0 bgcolor=#FFFFCC>");
54    for (int i=1; i<19; i++) {
55      string = weightstring[i].length();
56      temp[column] = weightstring[i];
57      movie[column] = weightstring[i].substring(0, (string-4));
58      string = 0;
59      if (column == 0) {
60        out.println("<tr>");
61      }
62      out.println("<td align=center><img src=Picture/"+weightstring[i]+" width=100>");
63      out.println("</img></td>");
64      column = column + 1;
65      if (column == 6) {
66        out.println("</tr>");
67        out.println("<tr>");
68        for (int x=0; x<6; x++) {
69          out.println("<td><input type=checkbox name=frame value="+temp[x]+ "> ");
70          out.println("<a href=Video/"+movie[x]+".mpeg>"+movie[x]+"</a></td>");
71        }
72        out.println("</tr>");
73        column = 0;
74    } }
75    out.println("</table>");
76    out.println("<br>");
```

```
77      out.println("<input type = submit value = submit>");
78      out.println("<td>เลือก KeyFrame ที่ต้องการแล้วกดที่ปุ่มนี้</td>");
79      out.println("</form>");
80  %>
81  </BODY>
82  </HTML>
```

ประวัติผู้เขียนโครงงาน

ชื่อ นายพิษณุกรณ์ เทียนสว่าง

ภูมิลำเนา 99/83 หมู่ 2 ค่ายพ่อขุนผาเมือง ม.พัน 18 ต.สะเดียง อ.เมือง จ.เพชรบูรณ์

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนเพชรพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

e-mail : phitsanukorn@hotmail.com

ชื่อ นายเกษมสรรญ์ จันทรา

ภูมิลำเนา 111/1 ถ.ชัยบุรี ต.ในเมือง อ.เมือง จ.พิษณุโลก

- จบระดับมัธยมศึกษาจากโรงเรียนเพชรพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

e-mail : jackykung0774@hotmail.com