การสืบค้นไฟล์วีดีโอด้วยวิธี AVI

Content-Based Video Retrieval using AVI method

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ -9 S.A. 2547
เลขทะเบียน 4700182
เลขเรียกหนังสือ
มหาวิทยาลัยนเรศวร
15067017
ปร.
ก637ก
2546
ฉ.2

นายกิจจา จีรพรชัย รหัส 43360338

นายอรรถสิทธิ์ ทรัพย์เมฆ รหัส 43360643

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2546

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	การสืบค้นไฟล์วีดีโอด้วยวิธี AVI
ผู้ดำเนินโครงงาน	นายกิจจา จีรพรชัย รหัส 43360338
	นายอรรถสิทธิ์ ทรัพย์เมฆ รหัส 43360643
อาจารย์ที่ปรึกษา	ดร.ไพศาล มุณีสว่าง
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2546

....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

....ประธานกรรมการ
(ดร.ไพศาล มุณีสว่าง)

....กรรมการ
(ดร.สุชาติ แย้มเม่น)

....กรรมการ
(ดร.สุรเชษฐ์ กานต์ประชา)

หัวข้อโครงงาน	การสืบค้นไฟล์วีดีโอด้วยวิธี AVI
ผู้ดำเนินโครงงาน	นายกิจจา จีรพรชัย รหัส 43360338
	นายอรรถสิทธิ์ ทรัพย์เมฆ รหัส 43360643
อาจารย์ที่ปรึกษา	ดร.ไพศาล มุณีสว่าง
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2546

....

บทคัดย่อ

งานวิจัยนี้มุ่งเน้นที่จะศึกษาถึงเทคนิคในการสืบค้นไฟล์วีดีโอที่มีอยู่ในฐานข้อมูลโดยใช้วิธี Adaptive Video Indexing (AVI) ซึ่งเป็นเทคนิคการสืบค้นไฟล์วีดีโอโดยนำจำนวนเฟรมทั้งหมดของไฟล์วีดีโอที่ใช้เป็นคำถามมาวิเคราะห์หาค่าความถี่ของแต่ละเฟรมแล้วนำไปเปรียบเทียบกับไฟล์วีดีโอที่มีอยู่ในฐานข้อมูล วิธีการ AVI จะสามารถสืบค้นไฟล์วีดีโอได้ทั้งในระดับ Shot, Scene และ Story ซึ่งจะมีความสามารถมากกว่าการสืบค้นไฟล์วีดีโอโดยใช้วิธีเฟรมตัวแทน(KFVI – key frame based video indexing) ที่นิยมใช้ในปัจจุบัน จากผลการทดลองพบว่าวิธีการแบบ AVI สามารถใช้ในการสืบค้นไฟล์วีดีโอได้ดีในทุก ๆ ระดับ โดยสามารถสืบค้นข้ามลำดับชั้นได้ เช่นใช้ Shot หรือ Scene เป็น query เพื่อสืบค้น Scene และ Story นอกจากนั้นพบว่า AVI สามารถใช้ในการค้นหาได้ดีกว่า KFVI

Project Title	Content-Based Video Retrieval using AVI method
Name	Mr. Kitja Jeerapornchai ID. 43360338
	Mr. Attasit supmek ID. 43360643
Project Advisor	Dr. Paisarn Muneeswang
Major	Computer Engineering
Department	Electrical and Computer Engineering
Academic Year	2003

....

ABSTRACT

This project studies content-based video retrieval (CBVR) by using adaptive video indexing technique (AVI). This technique indexes video file by considering all frames within video to obtain a frame-frequency vector used for search and retrieval. The AVI technique can retrieve video at all levels (i.e., shot, scene and story), which is very useful when compared to the key-frame based video indexing technique (KFVI). The proposed search engine can retrieve video file at shot, scene and story levels, as well as using shot and scene as a query to retrieve video scene and story. Based on the experimental results, the AVI video search engine provided better results than that of KFVI at all levels.

กิตติกรรมประกาศ

โครงงานวิศวกรรมคอมพิวเตอร์ สำเร็จได้ด้วยดีก็เนื่องด้วยความอนุเคราะห์จากท่านอาจารย์ที่ปรึกษาโครงงานคือ ดร.ไพศาล มุณีสว่าง ซึ่งได้ให้ทั้งคำแนะนำในด้านต่างๆ ตั้งแต่ทางด้านทฤษฎี ด้านโปรแกรมมิ่ง ด้านการเขียนโครงงานวิจัย และอื่นๆ รวมทั้งขอขอบคุณเพื่อนๆที่คอยให้ความช่วยเหลือในทุกๆด้าน อีกทั้งกำลังใจที่ได้รับจากทางบ้าน

ในโอกาสนี้ทางคณะผู้จัดทำโครงงานจึงขอขอบคุณทุกๆท่านที่มีส่วนช่วยทำให้โครงงานนี้ประสบความสำเร็จได้ด้วยดี

นายกิจจา จีรพรชัย
นายอรรถสิทธิ์ ทรัพย์เมฆ

มหาวิทยาลัยนเรศวร

สารบัญ

หน้า

บทคัดย่อภาษาไทย .... ก

บทคัดย่อภาษาอังกฤษ .... ข

กิตติกรรมประกาศ .... ค

สารบัญ .... ง

สารบัญตาราง .... ช

สารบัญรูป .... ซ

บทที่ 1 บทนำ 1

1.1 ปัญหาและที่มาของปัญหา 1

1.2 วัตถุประสงค์ 2

1.3 ขอบข่ายของงาน 3

1.4 ผลที่คาดว่าจะได้รับ 3

บทที่ 2 งานวิจัยที่เกี่ยวข้อง 6

2.1 ระบบสี (Color Models) 6

2.1.1 ระบบสีแบบRGB (RGB Color Model) 7

2.1.2 ระบบสี HSV (HSV Color Model) 8

2.2 การอธิบายถึงภาพต่างๆ โดยใช้ Color Histograms 11

2.2.1 การทำ Histogram ในระบบสี HSV 12

2.3 การประเมินค่าตัวแทนของเวคเตอร์ (Vector Quantization) 15

2.4 การใช้ความเหมือนระหว่างเวคเตอร์ในการสืบค้น 17

2.5 ระบบวีดีโอ (Video System) 19

2.5.1 การวิเคราะห์วีดีโอ 20

2.5.1.1 การกำหนดขอบเขตของ shot (Shot Boundary detection) 21

2.5.1.2 การเลือกเฟรมหลัก (Key frame extraction) 21

2.5.2 การแสดงวีดีโอ (Video Representation) 21

2.5.2.1 การแสดงเฟรมหลัก (Sequential key frame representation) 22

2.6 การสืบค้นไฟล์วีดีโอในฐานข้อมูล 22

สารบัญ (ต่อ)

หน้า

2.6.1 การค้นหาโดยใช้คำสำคัญ (Key word search) 22

2.6.2 การค้นหาโดยใช้คีย์เฟรม 23

2.7 แนะนำเกี่ยวกับ JSP 24

2.7.1 ข้อดีของ JSP 25

2.7.2 โครงสร้างและขั้นตอนการทำงานของ JSP 26

บทที่ 3 วิธีดำเนินงาน 28

3.1 การนำไฟล์วีดีโอมาทำ Histogram 28

3.2 การสืบค้นไฟล์วีดีโอในฐานข้อมูล Shot Scene และ Story โดยวิธี AVI 30

3.3 กระบวนการในการสร้างเวคเตอร์ AVI แม่แบบ (AVI template) 31

3.4 สร้างเวคเตอร์ AVI (AVI vector) 33

บทที่ 4 การทดลองการสืบค้นและส่วนที่เกี่ยวข้องกับงานวิจัย 36

4.1 ผลการทดลองการแบ่งช่วงของ Histogram 36

4.2 ผลการทดลองการสืบค้นไฟล์วีดีโอในระดับ Shot 37

4.3 ผลการทดลองการสืบค้นไฟล์วีดีโอในระดับ Scene และ Story 39

บทที่ 5 บทสรุปผลงานวิจัยและการนำงานวิจัยไปใช้ 41

5.1 การหาค่า Histogram 41

5.2 การสืบค้นไฟล์วีดีโอ 41

5.2.1 การสืบค้นไฟล์วีดีโอในระดับ Shot 41

5.2.2 การสืบค้นไฟล์วีดีโอในระดับ Scene และ Story 41

5.3 คำแนะนำในการนำกระบวนการสืบค้นไปใช้ 43

เอกสารอ้างอิง 44

ภาคผนวก ก 46

สารบัญ (ต่อ)

หน้า

ภาคผนวก ข 56

ประวัติผู้เขียนโครงงาน 58

สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงตัวอย่างของชุดตัวแทน (Code Book) 16

3.1 แสดงตัวอย่างการเลือกเวคเตอร์ที่ใช้ในการสืบค้นในไฟล์วีดีโอต่างระดับกัน 34

4.1 จำนวนของช่วงที่ใช้ในการทดลองการทำ 36

4.2 จำนวนของ Shot ในแต่ละเรื่องใน Database 38

ข-1 แสดงค่าของจำนวนเม็ดสีที่ตกลงในแต่ช่วงของ Histogram 57

มหาวิทยาลัยนเรศวร

สารบัญรูป

รูปที่		หน้า
2.1	แสดงตัวอย่างของรูปแบบของสีในระบบต่างๆ	6
2.2	แสดงตัวอย่างภาพแบบอ้างอิงตาราง	7
2.3	แสดงลูกบากศ์ของสีในระบบ RGB	8
2.4	แสดงระบบสี HSV	9
2.5	แสดงตัวอย่างการใช้ฟังก์ชัน rgb2hsv ใน Matlab	11
2.6	แสดงตัวอย่างการใช้ฟังก์ชัน rgb2hsv	11
2.7	แสดงตัวอย่างกานำภาพมาเก็บไว้ในรูปแบบดิจิติลไฟล์	12
2.8	แสดงตัวอย่างการทำ Histogram	12
2.9	แสดงตัวอย่างการแบ่งช่วงสีในระบบสี HSV	13
2.10	แสดงตัวอย่างของเวคเตอร์ในปริภูมิเวคเตอร์	14
2.11	แสดงตัวอย่างตารางสีที่ใช้อ้างอิงภาพในระบบ RGB	15
2.12	แสดงตัวอย่างการทำ Vector Quantization	16
2.13	แสดงการหาความเหมือนระหว่างเวคเตอร์	18
2.14	แสดงลำดับชั้นของกระบวนการที่เกี่ยวข้องกับเนื้อหาของวีดีโอ	19
2.15	แสดงขั้นตอนในวิเคราะห์วีดีโอ	20
2.16	แสดงการเรียงตัวของเฟรมในไฟล์วีดีโอ	22
2.17	แสดงเฟรมจากไฟล์วีดีโอในระดับ Shot	23
2.18	แสดงการสร้างเนื้อหาแบบไดนามิคด้วยสมาชิกต่างๆของ JSP	25
2.19	แสดงผังการทำงานของ JSP	26
2.20	แสดงขั้นตอนการประมวลผลไฟล์ JSP	27
3.1	แสดงตัวอย่างแบ่งเฟรมออกเป็น ส่วนของค่าในระบบสี HSV	29
3.2	แสดงตัวอย่างไฟล์วีโอในระดับ Scene	30
3.3	แสดงรูปแบบการสืบค้นไฟล์วีดีโอแบบ AVI	31
3.4	แสดงการเรียงของ Histogram ใน AVI	33
4.1	กราฟการทดลอง	37
4.2	กราฟทดลองเปรียบเทียบ KFVI กับ AVI	38
4.3	แสดงตัวอย่างการค้นหาไฟล์จากระดับ Shot	39
4.4	การทดลองระดับ Scene และ Story	40

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.5 แสดงตัวอย่างผลการค้นหาไฟล์จากระดับ Shot ใน Scene 40

5.1 แสดงตัวอย่างผลการค้นหาไฟล์วีดีโอ scene to story 42

ก-1 ตั้งชื่อการเชื่อต่อฐานข้อมูล 46

ก-2 ระบุรายละเอียดที่ใช้สำหรับการติดต่อฐานข้อมูล 47

ก-3 พิมพ์ชื่อฐานข้อมูลที่ต้องการจะสร้างใหม่ 48

ก-4 กำหนดชื่อตารางและคอลัมน์แรกที่ต้องการ 48

ก-5 กำหนดคุณสมบัติของฟิลด์ temp1 49

ก-6 กำหนดคุณสมบัติของฟิลด์ temp2 50

ก-7 ผลลัพธ์ที่ได้เมื่อสร้างตารางเรียบร้อยแล้ว 50

ก-8 ผลลัพธ์ที่ได้เมื่อสร้างตารางเรียบร้อยแล้ว 51

ก-9 เมื่อต้องการลบตาราง 52

ก-10 คลิกไปที่แท็บ Data เพื่อเพิ่มเรคอร์ดใหม่ 53

ก-11 เพิ่มเรคอร์ดใหม่ในตาราง 53

ก-12 อิมพอร์ทเทคไฟล์ 54

ก-13 เลือกเทคไฟล์เพื่อนำมาใส่ในฐานข้อมูล 54

ก-14 ใส่รายละเอียดสำหรับเทคไฟล์ที่จะใส่ฐานข้อมูล 55

ข-1 ภาพที่ได้ที่ได้จากการหาค่า Histogram โดยใช้ฟังก์ชัน hsvhist 56

ข-2 แสดงกระบวนการในการสร้างเวคเตอร์ AVI 58

บทที่ 1
บทนำ

1.1 ปัญหาและที่มาของปัญหา

โลกในยุคปัจจุบันเป็นยุคของเทคโนโลยีสารสนเทศและข้อมูลข่าวสาร สังคมที่เต็มไปด้วยการแข่งขันทำให้ทุกคนต้องอยู่กันอย่างเร่งรีบ และแข่งกับเวลา เพื่อให้ได้ข้อมูลข่าวสารที่ถูกต้องและแม่นยำในเวลารวดเร็วเพื่อที่จะนำสิ่งเหล่านั้นไปใช้ประโยชน์ด้านต่างๆต่อไป ทำให้เกิดเทคโนโลยีต่างๆเพื่อช่วยในการค้นหาข้อมูล จึงมีการสร้างเครื่องมือค้นหาที่เรียกว่า Search engine ขึ้นมา เช่นใน อินเตอร์เนตก็มีเวปไซต์ต่างๆที่ใช้ในการสืบค้นข้อมูลอยู่มากมาย เช่น google yahoo sanook เป็นต้น โดยจะสามารถสืบค้นหาข้อมูลที่ต้องการได้โดยใช้คำสำคัญในการค้นหา (Keyword) กล่าวคือ ถ้าจะหาข้อมูลเกี่ยวกับอะไรก็จะสามารถพิมพ์คำๆนั้นลงไป แล้วก็จะได้ข้อมูลของสิ่งที่ต้องการ แต่ของเสียของวิธีนี้ก็คือการจัดเก็บจะต้องมีการจัดทำดรรชนีของข้อมูลให้ถูกต้องเสียก่อน ซึ่งในการกระทำงานนี้จำเป็นที่จะต้องใช้ความรู้ในการทำ เป็นเหตุให้มีความล่าช้าในการจัดเก็บ ยิ่งถ้าต้องการจัดเก็บข้อมูลในระบบที่ใหญ่มากๆแล้วก็จะมีผู้ที่ทำการจัดเก็บข้อมูลหลายฝ่ายหลายคนทำให้มีปัญหาเกิดขึ้นหลายอย่าง ไม่ว่าจะเป็นความผิดพลาดของตัวผู้จัดเก็บเอง หรือแม้แต่มาตราฐานของตัวผู้จัดเก็บแต่ละคน ก็เป็นสาเหตุให้เกิดการผิดพลาดในการค้นหาได้ทั้งสิ้น จึงเป็นสาเหตุทำให้ไม่สามารถหาสิ่งที่ต้องการได้ครบถ้วน ตัวอย่างเช่น การสืบค้นหารูปภาพของสถานที่สำคัญต่างๆ หรือการหาไฟล์ดิจิตอลวีดีโอ เป็นต้น

เป็นที่ยอมรับกันว่าในยุคปัจจุบันระบบมัลติมิเดียมีส่วนเข้ามาเกี่ยวข้องกับชีวิตประจำวันเป็นอย่างมาก เช่น ทางด้านบันเทิงที่เกี่ยวข้องโดยตรง สถานศึกษาซึ่งใช้กันในการศึกษาทางไกลหรือการศึกษาด้วยตัวเอง ซึ่งการใช้ระบบมัลติมีเดียนั้นมักจะเกี่ยวข้องกับเสียงและภาพ ซึ่งมักจะจัดเก็บในลักษณะไฟล์ประเภทดิจิตอลวีดีโอ จะเห็นได้ว่าจากที่ในปัจจุบัน ไฟล์ประเภทนี้มีการใช้กันอย่างแพร่หลายในทุกสาขาอาชีพจึงทำให้มีไฟล์ประเภทนี้อยู่มากมาย แต่กลับมีวิธีที่ใช้ในการสืบค้นไฟล์ดิจิตอลวีดีโอนี้มาใช้งานไม่มากนัก ได้แก่ การสืบค้นโดยใช้คำสำคัญ (Keyword) ซึ่งวิธีนี้มีข้อเสียดังที่กล่าวมาแล้วในขั้นต้น จึงได้มีการสร้างวิธีสืบค้นโดยใช้ไฟล์วีดีโอต้นแบบ หรือการค้นหาโดยเปรียบเทียบเนื้อหาของไฟล์ที่มีลักษณะเหมือนและใกล้เคียงกับไฟล์ที่มีเนื้อหาที่เราต้องการมาจากในฐานข้อมูลทั้งหมด ตัวอย่างเช่น ต้องการหาไฟล์ดิจิตอลวีดีโอที่เกี่ยวกับฟุตบอลก็จะกำหนดรูปภาพที่เกี่ยวข้องกับฟุตบอล เช่น ภาพสนามฟุตบอล ภาพการยิงประตู ภาพคนดู เป็นต้น จากนั้นก็นำภาพที่กำหนดให้มาเปรียบเทียบกับเนื้อหาของวีดีโอว่ามีไฟล์วีดีโอใดที่มีเนื้อหาใกล้เคียงกับภาพที่

ต้องการ ก็จะทำให้สามารถหาเรื่องที่เกี่ยวข้องกับฟุตบอลได้จากฐานข้อมูลที่เรามีอยู่ทั้งหมดเป็นต้น ซึ่งจะทำให้ได้ข้อมูลที่มีอยู่มาอย่างรวดเร็วและง่ายดาย เพราะไม่ต้องเสียเวลาทำดรรชนีของไฟล์วีดีโอ ได้ไฟล์วีดีโอที่ต้องการและไฟล์วีดีโอที่มีส่วนที่เกี่ยวข้องมาด้วย ไม่จำเป็นต้องคำนึงถึงมาตราฐานในการจัดเก็บ อีกทั้งการสืบค้นแบบนี้ปัจจุบันมีการศึกษาและพัฒนากันอย่างแพร่หลาย และมีอยู่ด้วยกันหลายวิธี ซึ่งส่วนมากจะใช้เฟรมสำคัญ (Key Frame) ในการสืบค้น คือ การนำเฟรมซึ่งเป็นตัวแทนของแต่ละช่วง (Shot) ของวีดีโอมาเปรียบเทียบกับภาพตัวอย่าง แล้วตรวจว่ามีเฟรมตัวแทนจากไฟล์วีดีโอใดที่ตรงกันก็จะถือได้ว่าวีดีโอนั้นเป็นวีดีโอที่ต้องการค้นหา แต่เนื่องจากในความเป็นจริงแล้วไฟล์วีดีโอยังแบ่งออกเป็นอีกหลายระดับ ซึ่งจะพบว่าวิธีที่ใช้วิธีการเฟรมสำคัญในการสืบค้นนั้นสามารถหาได้เฉพาะวีดีโอที่อยู่ในระดับ Shot เท่านั้น ซึ่งในความจริงนั้นไฟล์ต้นแบบอาจเป็นแบบ Scene ก็ได้ และยังอาจเกิดการผิดพลาดจากการที่เลือกเฟรมตัวแทนผิดพลาด ดังนั้น ผู้จัดทำโครงการจึงได้มีความคิดที่จะจัดทำวิธีซึ่งสามารถค้นหาไฟล์วีดีโอซึ่งสามารถค้นหาได้ทั้งสามระดับของวีดีโอ

การค้นหาไฟล์จากฐานข้อมูลที่มีอยู่ทั้งหมดเป็นเรื่องที่ยุ่งยาก เนื่องจากต้องมีการทำดัชนี ต้องทราบถึงคีย์เวิร์ดที่เป็นไปได้ที่ใช้ในการค้นหา อีกทั้งต้องเก็บให้ถูกวิธีซึ่งถ้าไม่ใช่ผู้ที่เชี่ยวชาญในการจัดเก็บแล้วจะเป็นไปได้ยากที่จะทำการจัดเก็บไฟล์ต่างๆลงในฐานข้อมูล ไฟล์วีดีโอนั้นก็เป็นไฟล์ประเภทหนึ่งที่มีอยู่จำนวนมากมายในปัจจุบัน การจัดเก็บไฟล์ลงในฐานข้อมูลนั้นจะมีความยุ่งยากและซับซ้อน ทำให้ยากต่อการค้นหา ดังนั้นจึงได้มีการคิดค้นวิธีการค้นหาไฟล์วีดีโอโดยใช้เนื้อหาของไฟล์ต้นแบบมาเป็นตัวสืบค้นซึ่งก็จะมีเทคนิคและวิธีต่างๆกันอยู่หลายวิธี แต่ก็ยังสมารถค้นหาไฟล์วีดีโอได้เพียงในระดับ Shot เท่านั้นไม่สามารถค้นหาไฟล์วีดีโอในระดับที่สูงขึ้นไปอีกได้

1.2 วัตถุประสงค์

1.2.1 เพื่อสร้างระบบการจัดการข้อมูลของไฟล์ดิจิตอลวีดีโอ ที่ใช้กระบวนการสืบค้นโดยเนื้อหาของวีดีโอ ที่มีความถูกต้องและน่าเชื่อถือ

1.2.2 นำระเบียบวิธีการ Adaptive-Video Indexing (AVI) ซึ่งเป็นการใช้เนื้อหาของไฟล์วีดีโอต้นแบบมาใช้ในการสืบค้นไฟล์วีดีโอที่มีลักษณะอย่างที่ต้องการ จึงทำให้มีคุณสมบัติที่ดีกว่าการสืบค้นแบบเดิม(แบบเฟรมสำคัญ (Key frame)) คือมีความสามารถสืบค้นครอบคุมทั้งในระดับ Shot, Scene และ Story

1.2.3 นำภาษาจาวามาสร้างโปรแกรมเพื่อสืบค้นไฟล์ภาพวีดีโอที่ต้องการด้วยวิธีการ Adaptive-Video Indexing (AVI) ที่สามารถค้นหาได้ทั้งในเครื่องของตนเองและยังสามารถค้นหาผ่านระบบเครือค่ายได้

1.3 ขอบข่ายของงาน

สร้างกระบวนการและเครื่องมือที่ช่วยในการสืบค้น (Search engine) ไฟล์ดิจิตอลวีดีโอที่ต้องการได้อย่างถูกต้อง โดยการออกแบบการจัดเก็บไฟล์ดิจิตอลวีดีโอทั้งสามระดับ (Shot, Scene และ Story) นั้นนำมาจากกระบวนการจัดเก็บที่มีอยู่ทั่วไป นำภาษาจาวามาใช้ในการพัฒนาโปรแกรมที่นำมาใช้งานซึ่งจะทำให้โปรแกรมสามารถสืบค้นไฟล์ประเภทนี้ได้จากทั้งในเครื่องและในระบบเครือข่าย (Internet) โดยใช้ JAVA พัฒนาเครื่องมือทั้งใน sever และ client

1.4 ผลที่คาดว่าจะได้รับ

1.4.1 สามารถพัฒนาและสร้างสรรค์ web server ในลักษณะที่เป็น web service ให้บริการข้อมูลประเภทไฟล์ดิจิตอลวีดีโอผ่านทางระบบ Internet โดยสามารถรองรับการทำงานกับหลาย user ในเวลาเดียวกันได้ ทั้งนี้ server ดังกล่าวจะต้องให้ผลการทำงานมีความถูกและมีเสถียรภาพ

1.4.2 มีความรู้ความเข้าใจหลักการทำงานของระบบ Server – Client ในระดับที่สามารถพัฒนาและประยุกต์ใช้ Application ให้เกิดประโยชน์บนระบบ Internet ได้โดยอาศัย Java เป็นเครื่องมือในการพัฒนา web server

1.4.3 มีความรู้ความเข้าใจในการที่จะพัฒนา เพื่อรองรับการทำงานในลักษณะ multi user รวมถึงสามารถออกแบบโครงสร้างและการทำงานของ ระบบฐานข้อมูลได้อย่างมีเสถียรภาพ

มหาวิทยาลัยนเรศวร

กิจกรรมการดำเนินการ

กิจกรรม \ สัปดาห์,เดือน,ปี	ปี 2545								ปี 2546															
	พ.ย.				ธ.ค.				ม.ค.				ก.พ.				มี.ค.				เม.ย.			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. ศึกษาข้อมูลและทฟษฎีโดยทั่วๆไปทั้งหมดที่เกี่ยวข้องกับ Video Retrieval	■	■	■	■																				
2. ศึกษาทฤษฎีของการจัดเก็บข้อมูลของวีดีโอและสร้างฐานข้อมูลทั้งแบบ Shot Scence และ Story					■	■	■																	
3. ศึกษาวิธีการค้นหาไฟล์วีดีโอโดยใช้วิธีแฟรมสำคัญ (Key frame)						■	■																	
4. ค้นหาข้อมูลและตัวอย่างของการค้นหาแบบใช้แฟรมสำคัญ								■	■															
5. ศึกษาโปรแกรมภาษาจาวา(J2EE, JSP)					■	■	■	■	■	■	■	■												
6. สร้าง Search engine สำหรับค้นหาไฟล์วีดีโอโดยใช้แฟรมสำคัญ										■	■	■												
7. ทดสอบและและวิเคาระห์ข้อดีข้อเสียของการค้นหาโดยใช้แฟรมสำคัญ											■	■												
8. แก้ไขปรับปรุงฐานข้อมูลไฟล์วีดีโอให้มีประสิทธิภาพ													■	■	■									
9. ศึกษาและสร้างโปรแกรมที่ใช้ในการสืบค้นไฟล์วีดีโอโดยวิธี Adaptive-Video Indexing (AVI)																								
* ศึกษาทฤษฎีและออกแบบอัลกอลิทึมที่ใช้ Shot ในการสืบค้นไฟล์วีดีโอ																■	■							
* ทำการสร้างโปรแกรมที่ใช้ Shot ในการค้นหาไฟล์วีดีโอ																	■	■	■					
* ศึกษาทฤษฎีและออกแบบอัลกอลิทึมที่ใช้ Sence ในการสืบค้นไฟล์วีดีโอ																			■	■	■	■	■	
* ทำการสร้างโปรแกรมที่ใช้ Sence ในการค้นหาไฟล์วีดีโอ																					■	■	■	■

กิจกรรมการดำเนินการ

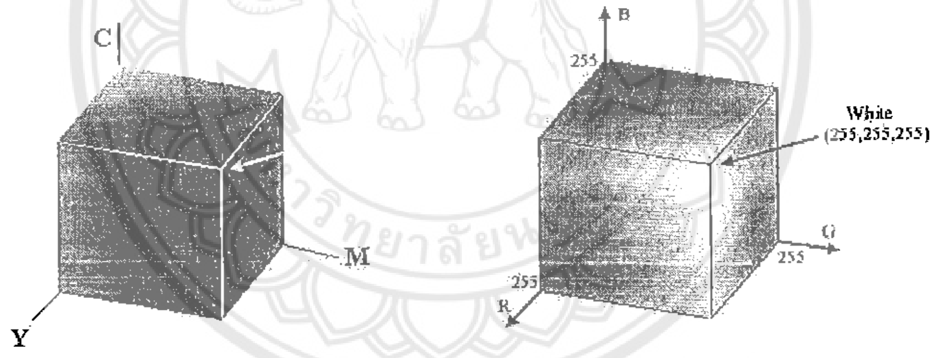
สัปดาห์,เดือน,ปี / กิจกรรม	ปี 2546																							
	พ.ค.				มิ.ย.				ก.ค.				ส.ค.				ก.ย.				ต.ค.			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
* ทำการสร้างโปรแกรมที่ใช้ Sence ในการค้นหาไฟล์วีดีโอ	■	■																						
* ศึกษาและออกแบบอัลกอลึทึมในการใช้ Story ในการสืบค้นไฟล์วีดีโอ			■	■	■	■	■	■																
* ทำการสร้างโปรแกรมที่ใช้ Story ในการค้นหาไฟล์วีดีโอ					■	■	■	■	■	■	■													
10. นำข้อมูลและโปรแกรมที่ได้จากข้อที่ 9. มาพัฒนาให้สามารถใช้ร่วมกันได้												■	■	■	■	■								
11. สร้าง Search engine สำหรับสืบค้นไฟล์วีดีโอโดยใช้กระบวนวิธีการ Adaptive-Video Indexing														■	■	■	■							
12. แก้ไขและเพิ่มเติมส่วนที่ยังบกพร่องของโปรแกรมและ Search engine ให้มีประสิทธิภาพ																	■	■	■	■				
13. สรุปและวิเคราะห์ผลของการสืบค้นแบบต่างๆที่ได้ทำมา																					■	■		
14. จัดทำเอกสารประกอบโครงการ																					■	■	■	■

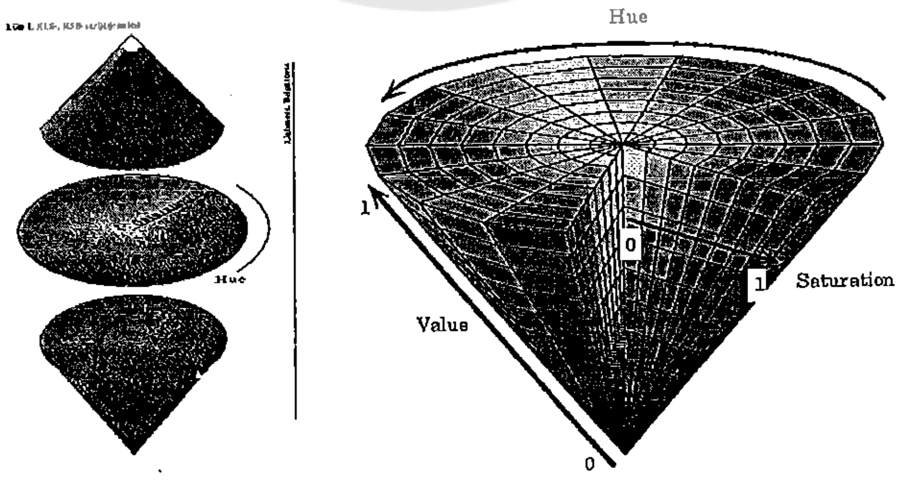
บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การสืบค้นไฟล์ในฐานข้อมูลนั้นเราจะต้องทำการสร้างสิ่งที่เอาไว้อ้างอิงถึงไฟล์นั้นๆ ซึ่งจะเก็บไว้ในฐานข้อมูลรูปแบบต่างๆ [11] เช่น เป็นเวคเตอร์ที่ได้จากการหาคำที่ปรากฏในฐานข้อมูล หรืออื่นๆ ในการสืบค้นไฟล์วีดีโอ เราจะต้องทำการสร้างเวคเตอร์สำหรับอ้างอิงถึงไฟล์นั้นๆเช่นกัน โดยอาจจะให้ผู้ที่มีความรู้สร้างเวคเตอร์ของไฟล์วีดีโอเป็นคนใส่เข้าไป หรืออาจจะใช้เนื้อหาของไฟล์วีดีโอเป็นต้นแบบในการสืบค้นที่ซึ่งโดยส่วนมากมักจะใช้การหาค่า Histogram ของเฟรมในไฟล์วีดีโอนั้นๆเพื่อใช้ในการค้นหา ในบทนี้จะเป็นการแนะนำทฤษฎีที่นำมาใช้ในโครงงานนี้ ซึ่งมีเรื่อง ระบบสี ทฤษฎีที่นำมาในการสร้างเวคเตอร์ เพื่อใช้อ้างถึงไฟล์วีดีโอ การสืบค้นข้อมูลวีดีโอ การวิเคราะห์และระบบวีดีโอ และ JSP

2.1 ระบบสี (Color Models)

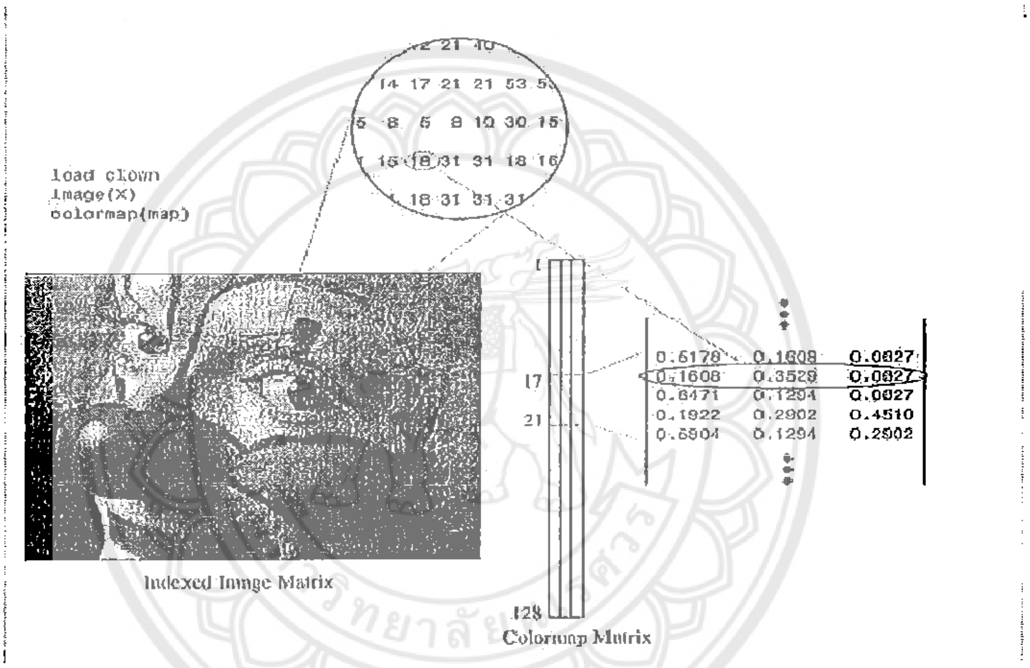




จุดประสงค์ของการสร้างรูปของระบบสีขึ้นมานั้นก็เพื่อที่จะสร้างมาตรฐานในการนำสีต่างๆ ในธรรมชาติที่มองเห็นซึ่งมีลักษณะเป็นคลื่นต่อเนื่องกันมาเก็บไว้ในคอมพิวเตอร์ที่จะเก็บเป็นลักษณะของสัญญาณดิจิตอล [12] โดยทั่วไปแล้วในระบบการเก็บนั้นจะเก็บไว้ในลักษณะของคู่อันดับของตัวเลขต่างๆเพื่อที่จะแสดงค่าของสีต่างๆ รูปแบบของสีในระบบต่างๆ แสดงในรูปที่ 2.1

ในปัจจุบันระบบสีจะถูกแบ่งแยกไปตามลักษณะของอุปกรณ์ที่ที่ใช้ (อาทิเช่น จอมอนิเตอร์, เครื่องพิมพ์ เป็นต้น) หรือที่องค์ประกอบของการนำไปใช้ (อาทิเช่น สำหรับสร้างภาพการฟึก) [10]

ตัวอย่างการอ้างอิงภาพจากรูปที่ 2.2



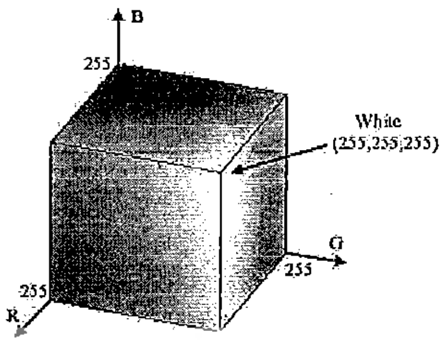
รูปที่ 2.2 แสดงตัวอย่างภาพแบบอ้างอิงตาราง

2.1.1 ระบบสีแบบRGB (RGB Color Model)

ในระบบสีแบบRGBนั้นสีแต่ละแสงสีที่แสดงออกมานั้นจะประกอบไปด้วย สีแดง สีเขียว และสีน้ำเงิน ในระบบนี้แสงสีแต่สีจะถูกแทนด้วยคู่อันดับดังแสดงในรูปที่ 2.3

ในระบบสี RGB นี้ค่าของสีแต่ละสีจะถูกแทนด้วยตัวเลขจำนวน 8 บิต ดังนั้นจะมีค่าระบบความเข้มของสีแดงทั้งหมดคือ 2^8 หรือ 256 สี(เริ่มตั้งแต่ 0 ถึง 255) ในสีเขียวและสีน้ำเงินก็เช่นกัน เนื่องจากแต่ละจุดสีในคอมพิวเตอร์นั้นจะประกอบไปด้วยค่าความเข้มของสีแดง,สีเขียวและสีน้ำเงิน ซึ่งจะใช้จำนวนบิตทั้งหมด 24 บิตในการแสดงสีแต่ละสี ดังนั้นจึงหมายความว่าจะสามารถแทนสีต่างๆได้จำนวนสีทั้งหมด $2^8 \times 2^8 \times 2^8$ หรือ 16,777,216 สีนั้นเอง โดยที่เมื่อค่าของสีแดง,สีเขียวและสีน้ำเงินมีค่ามากที่สุดจะให้ค่าของสีออกมาเป็นสีขาว ในทางตรงข้ามเมื่อค่าของสีแดง , สีเขียว

และสีน้ำเงินมีค่าเป็นศูนย์จะให้ค่าของสีเป็นสีดำ ดังนั้นเส้นที่ลากระหว่างมุมทั้งสองของลูกบาศก์ก็จะเป็นระดับสีเทานั้นเอง



รูปที่ 2.3 แสดงลูกบากศ์ของสีในระบบ RGB

เมื่อใช้ระบบสี RGB ในการแสดงผลในจอภาพที่มีประสิทธิภาพสูง มักจะจำใช้จำนวนบิตที่ใช้ในการแสดงผลต่อหนึ่งจุดสีคือ 24 บิต แต่โดยทั่วไปแล้วมักจะใช้จำนวนบิตต่อหนึ่งจุดสีน้อยกว่าเนื่องจากระบบประสาทมักจะแยกแยะถึงแสงสีได้ประมาณสองร้อยถึงสามร้อยสีเท่านั้น ดังนั้นวิธีที่ดีคือการใช้สีเทียมซึ่งเป็นที่นิยมกันอยู่ในปัจจุบัน โดยจะเก็บสีต่างๆไว้ในตารางในรูปแบบของสีต่างๆ การจะนำสีนั้นมาแสดงผลนั้นจะใช้เวคเตอร์ในการชี้ไปยังตาราง ณ จุดสีนั้น ซึ่งจะเก็บค่าของสีแดง, เขียว และสีน้ำเงินไว้ วิธีดังกล่าวจะใช้เพียง 8 บิตต่อหนึ่งจุดสีเท่านั้น แต่วิธีดังกล่าวจะทำให้ค่าความละเอียดของสีได้ไม่สมบูรณ์นักเพราะจะแสดงสีได้เพียง 256 สีเท่านั้น

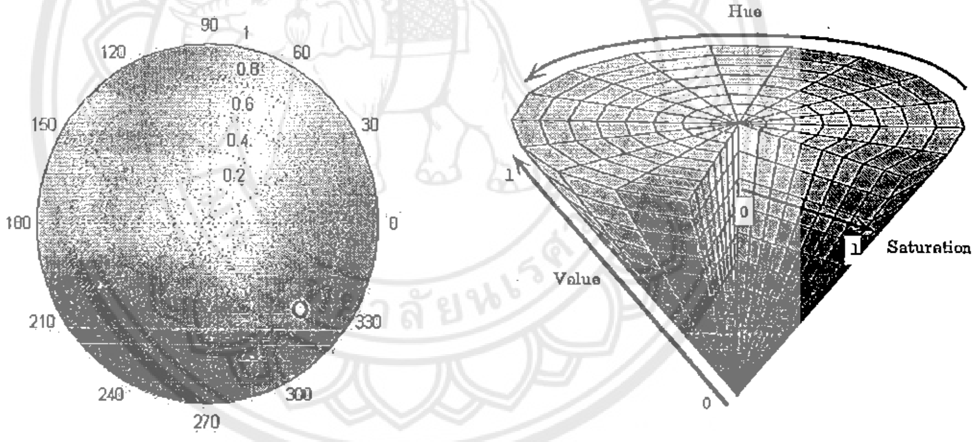
2.1.2 ระบบสี HSV (HSV Color Model)

แม้ว่าระบบสี RGB และระบบสีอื่นๆที่ใกล้เคียงกันจะสามารถนำมาใช้แสดงผลในอุปกรณ์ดิจิตอลได้เป็นอย่างดี รวมไปถึงสามารถมองเห็นได้ดีด้วยตาของมนุษย์ที่รับรู้สีจากแสงสีแดง, แสงสีเขียวและแสงสีน้ำเงินในแต่ละน้ำหนักของแสดงสีก็ตาม แต่กลับทำให้ไม่มามารถบอกถึงรายละเอียดของสีได้ชัดเจนนักและไม่สามารถตอบสนองได้กับรูปแบบในการมองเห็นของมนุษย์ กล่าวคือมนุษย์จะมองเห็นสีในลักษณะของสี,ความเป็นสี(สีซีดหรือสีจาง)และความมือและสว่างของสี [12]

เมื่อมนุษย์มองเห็นวัตถุหนึ่ง เขาจะอธิบายสีของวัตถุนั้นว่ามีสีอะไร (สีแดง, สีเขียว, สีส้ม...) ลักษณะของสีเป็นอย่างไรสีสดหรือสีซีด และมีความสว่างหรือมีความมืดของรูปภาพเป็นอย่างไร จึงได้มีระบบสี HSV (Hue, Saturation, Value) ที่สามารถอธิบายถึงสีได้ใกล้เคียงกับที่มนุษย์เห็นที่สุด โดยที่ค่าของ H (Hue) จะหมายถึงค่าของสีบริสุทธิ์ว่าเป็นสีอะไร (สีแดง, สีเขียว, สีส้ม...) ในขณะที่ค่าของ S (Saturation) จะบอกถึงค่าการปนเปื้อนของแสงสีขาวในจุดสีนั้นๆ ซึ่งจะทำให้ได้สีที่มีความสดไปถึงสีซีดจางจนกระทั่งเป็นสีขาว ส่วนค่าของ V (Value) ใช้บอกถึงการปนเปื้อนของสี

เทาซึ่งทำให้สีมีความเข้มขึ้น ระบบสี HSV นี้เป็นพื้นฐานที่ใช้ในการพัฒนากระบวนการต่างๆที่เกี่ยวกับการมองเห็นของมนุษย์ เรานำภาพที่ได้มาจากกล้องดิจิตอลซึ่งจะเก็บไว้ในระบบสี RGB แต่ถ้าจะนำมาทำกระบวนการใดกระบวนการหนึ่งกับภาพนั้น จะต้องนำภาพนั้นมาผ่านกระบวนการแปลงให้อยู่ในระบบสี HSV ก่อนที่จะทำต่อไป [10, 12]

ถ้านำภาพขาวดำ (Intensity image) ภาพหนึ่งมาอธิบายด้วยค่าของสีในระบบ RGB แล้วนำค่าของจุดสีที่ได้มาพลอตลงในรูปลูกบาศก์ของระบบสี RGB ก็จะพบว่าจุดที่ได้จะอยู่บนเส้นที่ลากระหว่างจุด (0, 0, 0) ถึงจุด (1, 1, 1) ซึ่งเส้นนี้เราเรียกว่าเส้นความเป็นสีเทา หรือความของ V (Value) ในระบบสี HSV นั้นเองถ้ามีสีใดๆอยู่ด้วย ณ ตำแหน่งของค่าของสีเทาค่าใดค่าใดค่าหนึ่งก็จะอยู่บนระนาบที่ตัดตั้งฉากกับแกนของเส้นแสดงความเป็นสีเทานั้นเองโดยจุดดังกล่าวจะวางตัวอยู่บนระนาบรอบ ๆ แกนอยู่เส้นดังกล่าว ซึ่งองศาของจุดที่วางอยู่นั้นคือค่าของสี (H, Hue) ในระบบสี HSV นั้นเอง ส่วนค่าของความปนเปื้อนของสีขาว (S, Saturation) นั้นจะสามารถอธิบายได้ด้วยระยะห่างจากแกนของสีเทา จากที่กล่าวมาจึงทำให้สามารถสรุปได้ว่าระบบสี HSV สามารถนำมาใช้แทนที่ระบบสี RGB ได้นั้นเอง



รูปที่ 2.4 แสดงระบบสี HSV

ถ้านำระนาบตัดที่มีค่าของความเป็นสีเทา (Value) ค่าหนึ่งมาเขียนจะได้ภาพในระบบสี HSV ลักษณะดังรูปที่ 2.4 จากที่กล่าวไปแล้วว่าระนาบจะมีความปนเปื้อนของสีเทาเปลี่ยนไปตามแกนของค่าความเป็นสีเทา ในรูปแรกจะพบว่ามีจะสี (สีแดง, สีเขียว, สีน้ำเงิน) หลักอยู่ที่บริเวณค่าของสี (H, Hue) ที่ 0 องศา 120 องศา 240 องศา โดยเริ่มนับศูนย์องศาที่สีแดง ดังนั้นจะพบว่าที่ค่าของสีที่ 0 องศา กับที 360 องศา จะมีค่าสีเป็นสีเดียวกันคือสีแดง ส่วนค่าความปนเปื้อนของสีขาวหรือค่าความซีดจางของสี (S, Saturation) ซึ่งจะมีความซีดจางของสีมากที่บริเวณใกล้จุดศูนย์กลางและสีจะมีความสดมากขึ้นเมื่อขึ้นเมื่อเคลื่อนที่ออกไกลจากจุดศูนย์กลาง ตามรูปที่ 2.4

การแปลงสีจากระบบสี RGB ไปเป็นระบบสี HSV

การแปลงจุดสีหนึ่งๆในระบบสี RGB ไปเป็นระบบสี HSV นี้จะมีสมการ [12] ดังนี้

ค่าของสี (H, Hue)

$$H = \begin{cases} \theta & \text{เมื่อ B น้อยกว่าหรือเท่ากับ G} \\ 360 - \theta & \text{เมื่อ B มากกว่า G} \end{cases} \quad (2.1)$$

$$\theta = \cos^{-1}\left\{\frac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\left[(R-G)^2+(R-B)(G-B)\right]^{1/2}}\right\} \quad (2.2)$$

ค่าของความปนของสีขาว (S, Saturation)

$$S = 1 - \frac{3}{(R+G+B)}\left[\min(R,G,B)\right] \quad (2.3)$$

และค่าของความปนของสีเทา (V, Value)

$$V = \frac{1}{3}(R+G+B) \quad (2.4)$$

กำหนดให้ค่าดังกล่าวเป็นค่าปกติ (Normalized) คือมีค่าสูงสุดเป็นหนึ่งในแต่ละระบบสี

ฟังก์ชันที่แปลงระบบสี RGB เป็นระบบสี HSV ใน Matlab

ใน Matlab จะมีฟังก์ชันที่ใช้สำหรับการแปลสีจากระบบสี RGB ไปเป็นสีในระบบสี HSV คือ ฟังก์ชัน rgb2hsv ซึ่งค่าที่ได้จะเป็นค่าปกติ (Normalize) รูปแบบการใช้งานฟังก์ชันดังกล่าวมีอยู่ด้วยกัน 2 วิธีดังนี้

- ✓ แปลงรูปภาพที่เก็บแบบอาร์เรย์ 3 มิติ (กว้างxยาวx3) ซึ่งอยู่ใน Matlab ในระบบสี RGB ไปเป็นระบบสี HSV [4]
- ✓ แปลงรูปภาพที่เก็บในลักษณะสีที่อ้างอิงกับตารางสี RGB ไปเป็นอ้างอิงตารางสี HSV ที่มีอยู่ใน Matlab [10] ตามรูปที่ 2.5 และรูปที่ 2.6

```
>> RGB = imread('flowers.tif');
>> HSV = rgb2hsv(RGB);
>> H=HSV(:,:,1);S=HSV(:,:,2); V=HSV(:,:,3);
>> subplot(2,2,2), imshow(H,64), title('Hue')
>> subplot(2,2,3), imshow(S,64), title('Saturation')
>> subplot(2,2,4), imshow(V,64), title('Value')
>> subplot(2,2,1), imshow(RGB), title('Original')
```

(a)

```
>> load trees
>> hsv = rgb2hsv(map);
>> [H,S,V] = ind2rgb(X,hsv);
>> subplot(2,2,2), imshow(H,64), title('Hue')
>> subplot(2,2,3), imshow(S,64), title('Sturation')
>> subplot(2,2,4), imshow(V,64), title('Value')
>> subplot(2,2,1), imshow(X,map), title('Original')
```

(b)

รูปที่ 2.5 แสดงตัวอย่างการใช้งานฟังก์ชัน rgb2hsv ใน Matlab

รูปที่ 2.6 แสดงผลของการใช้ฟังก์ชัน rgb2hsv

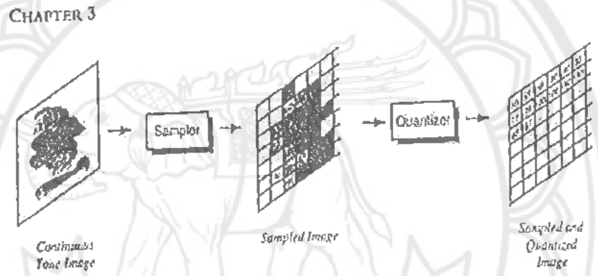
2.2 การอธิบายถึงภาพต่างๆโดยใช้ Color Histograms

ภาพดิจิตอลจะประกอบขึ้นด้วยจุดของเม็ดสี (pixel) ตามรูปที่ 2.7 จำนวนมากมายมารวมกัน [10, 12] จนเกิดภาพโดยที่แต่ละจุดสีนั้นจะมีค่าที่อธิบายว่าเป็นสีอะไร เช่นในระบบสี RGB ก็จะอธิบายแต่จุดสีว่ามีความของสีแดง,สีเขียวและสีน้ำเงินเท่าไร ในการทำ Histograms นั้นสามารถทำได้โดยเริ่มจากสร้างกราฟที่มีจำนวนช่วงเท่ากับจำนวนของช่วงสีที่ต้องการ การกำหนดลำดับชั้นของสีที่ต้องการจากค่าสูงสุดของแต่ละช่วงมาแบ่งเป็นช่วง ๆ ตามจำนวนของลำดับชั้นที่ต้องการ

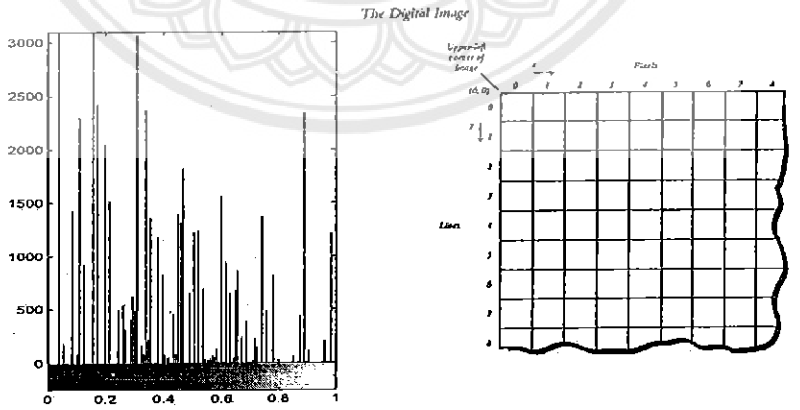
จากนั้นจึงนำจุดสีทุกจุดในภาพมาเปรียบเทียบว่าอยู่ในช่วงใดของกราฟ เมื่อมีจุดสีใดตกอยู่ในช่วงใดก็ให้ทำการเพิ่มค่าในช่วงดังกล่าวจำนวนหนึ่งค่า ทำอย่างนี้ตั้งแต่จุดสีแรกของภาพจนถึงจุดสีสุดท้ายของภาพ โดยทั่วไปแล้วการทำ Histogram นั้นเรามักจะเอาค่าที่ได้มาหาอัตราส่วนระหว่างค่าของจำนวนสีในช่วงนั้นๆกับจำนวนเม็ดสีทั้งหมดในภาพเพื่อที่จะหาอัตราส่วนของความหนาแน่นของช่วงสีหนึ่งๆเท่า ซึ่งจะทำให้ทราบว่ามีช่วงสีใดมากในภาพดังกล่าว สมการที่ใช้ในการหาค่าของ Histogram [14] คือ

$$h[m] = \frac{1}{XY} \sum_{y=Y} \sum_{x=X} \begin{cases} 1 & \text{เมื่อค่าสีของจุดนั้นอยู่ใน } m \\ 0 & \text{เมื่อไม่อยู่} \end{cases} \qquad (2.5)$$

ก็จะได้ Histogram ดังตัวอย่างรูปที่ 2.8



รูปที่ 2.7 แสดงตัวอย่างกานำภาพมาเก็บไว้ในรูปแบบดิจิตอลไฟล์



รูปที่ 2.8 แสดงตัวอย่างการทำ Histogram

2.2.1 การทำ Histogram ในระบบสี HSV

ในการทำ Histogram ของภาพสีในระบบ HSV นั้นจะเป็นการแสดงถึงค่าขององค์ประกอบของแต่ละค่าของสีในรูปของระบบสี HSV ถ้าภาพมีสีที่อยู่ในช่วงใดมากช่วงนั้นก็จะได้กราฟที่มี

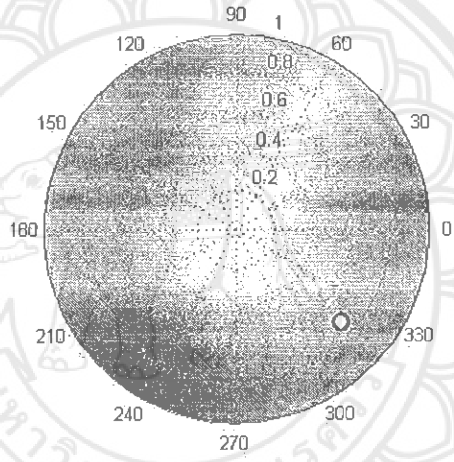
ความสูงมาก ค่าของ Histogram ในช่วงของระบบสี HSV จะมีจำนวนของช่วงสีที่ต้องการคือ ค่าผลคูณระหว่างค่าของจำนวนช่วงในแกน Hue, Saturation และ Value ซึ่งจะได้เป็นจำนวนช่วงสีทั้งหมดที่ต้องการ จากนั้นนำค่าของจุดสีที่อยู่ในระบบของ HSV มาเปรียบเทียบว่าอยู่ในช่วงใด โดยการหาค่าของความกว้างของช่วงสีในแต่ละแกนหาได้จากสมการ

$$scale = n/A \tag{2.6}$$

เมื่อ scale คือ ค่าของความกว้างในแต่ละช่วงที่ต้องการ

n คือ ค่าของจำนวนของช่วงที่ต้องการในแต่ละแกน

A คือ ค่าสูงสุดในแต่ละช่วง



รูปที่ 2.9 แสดงตัวอย่างการแบ่งช่วงสีในระบบสี HSV

เมื่อได้ค่าของความกว้างในแต่ละช่วงมาแล้ว เช่น ตัวอย่างการแบ่งช่วงสีในรูปที่ 2.9 ก็นำจุดสีมาหาว่าอยู่ในช่วงใดของกราฟจากสมการข้างล่าง

เมื่อ S และ V ไม่เท่ากับ 0

$$id = [ceil(H \times scaleH) + (nH \times ((S \times scaleS) - 1)) + ((nH \times nS) \times ((V \times scaleV) - 1))] \tag{2.7}$$

เมื่อ S เท่ากับ 0

$$id = [ceil(H \times scaleH) + ((nH \times nS) \times ((V \times scaleV) - 1))] \tag{2.8}$$

เมื่อ V เท่ากับ 0

$$id = [ceil(H \times scaleH) + (nH \times ((S \times scaleS) - 1))] \tag{2.9}$$

โดยที่

id คือ ค่าของตำแหน่งช่วงที่จะทำการเพิ่มค่า

H, S, V คือ ค่าของจุดสีในระบบ HSV

nH, nS, nV คือ ค่าของจำนวนช่วงสีที่ต้องการในแต่ละแกน

ceil(x) คือ การหาค่าบนของ x

การหาค่าความต่างระหว่าง Histogram (Histogram Distance)

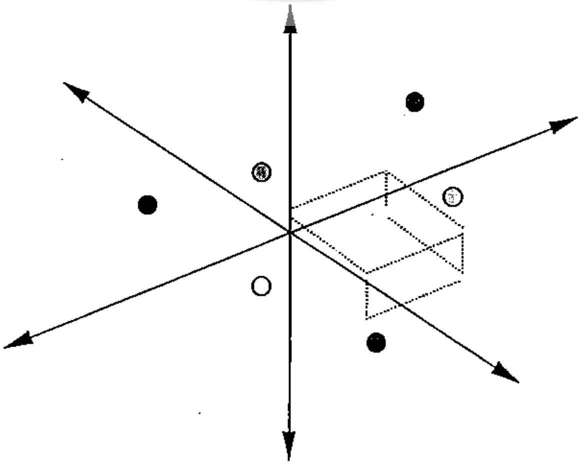
ถ้าเรามองว่า Histogram ที่หามาได้จากขั้นตอนต่างๆตามที่กล่าวมาเป็นเวคเตอร์ในปริภูมิเวคเตอร์ที่มีหลายมิติก็จะพบว่า Histogram ของภาพแต่ละภาพที่ได้ [11, 14] จะเป็นจุดๆหนึ่งในปริภูมิเวคเตอร์นั้นเอง ยกตัวอย่างเช่นในรูปที่ 2.10 ดังนั้นเราจึงสามารถหาระยะห่างระหว่างเวคเตอร์ดังกล่าวได้จากสมการข้างล่าง

$$D = \left(\sum \left| h_f[m] - h_q[m]^2 \right| \right)^{1/2} \tag{2.10}$$

เมื่อ D คือ ค่าของความต่างของ Histogram

h_i คือ ค่าของ Histogram ที่ i

ค่าที่ได้จะบอกถึงระยะห่างระหว่างจุดสองจุดบนปริภูมิเวคเตอร์ ซึ่งก็คือค่าของความแตกต่างระหว่าง Histogram ทั้งสองตัวนั้นเอง



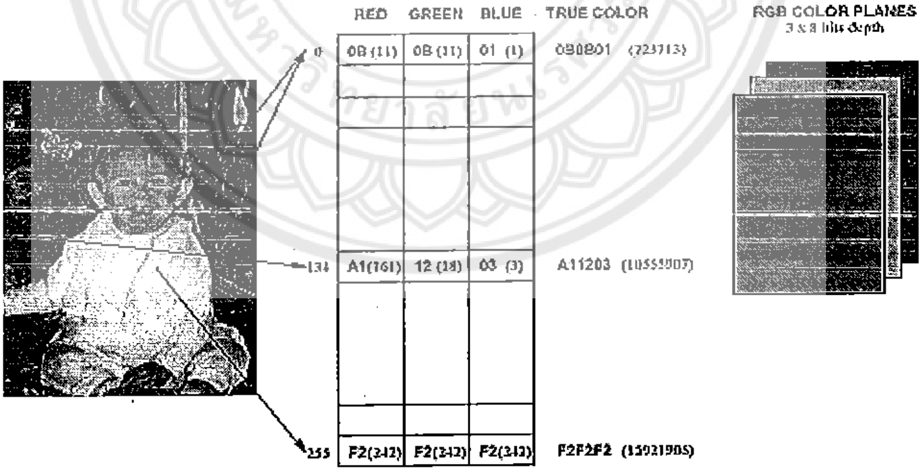
รูปที่ 2.10 แสดงตัวอย่างของเวคเตอร์ในปริภูมิเวคเตอร์

2.3 การประเมินค่าตัวแทนของเวคเตอร์ (Vector Quantization)

การประเมินค่าตัวแทนของเวคเตอร์ คือ การหาเวคเตอร์ที่เป็นตัวแทนของกลุ่มเวคเตอร์ที่ต้องการซึ่งจะมีประโยชน์ในการบีบอัดจำนวนค่าของความจุที่ใช้ต่อหนึ่งภาพ [13] ถ้าเรามีภาพดิจิตอลที่มีจำนวนจุดสีซึ่งสามารถอธิบายได้โดยค่าของสีต่างๆในระบบสีใดระบบสีหนึ่ง เช่นในระบบ RGB จุดสีจะมีค่าของสีแดง,สีเขียวและสีน้ำเงินที่อธิบายจุดสีนั้นใน ถ้าเราใช้จำนวนบิต 8 บิตต่อการอ้างถึงสีแดง 8 บิตต่อการอ้างถึงสีเขียว และอีก 8 บิตต่อการอ้างถึงสีน้ำเงิน ก็หมายความว่าเราต้องใช้จำนวนบิตที่ใช้อธิบายถึงจุดสีดังกล่าวจำนวน 24 บิต และถ้าภาพมีขนาด 256x256 ก็จะทำให้ต้องใช้หน่วยความจำที่ใช้เก็บภาพทั้งหมดจำนวน 1,572,864 หรือ ประมาณ 1.6 เมกะไบต์ ซึ่งจะมีค่าสูงมาก

การเก็บภาพในปัจจุบันส่วนใหญ่จะเก็บไว้ในรูปแบบของภาพแบบอ้างอิงซึ่งจุดสีตำแหน่งต่างๆจะนำไปชี้ค่าในตารางสีที่มีอยู่ซึ่งจะทำให้ใช้จำนวนบิตที่ใช้อธิบายจุดสีหนึ่งๆน้อยกว่าเดิม [10, 13]

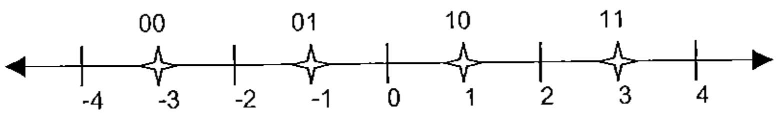
จากรูปที่ 2.11 กระบวนการในการหาสีในภาพมาใส่ตารางมักจะพบในการบวนการในการบีบอัด โดยหาค่าตัวแทนของสีในภาพนั้นๆ โดยจุดสีในระบบสี RGB นั้นจะถูกแทนด้วยค่าของคู่อันดับในปริภูมิเวคเตอร์และนำมาผ่านกระบวนการหาตัวแทนของกลุ่มเวคเตอร์ให้ได้จำนวนค่าของสีตามที่ต้องการ



รูปที่ 2.11 แสดงตัวอย่างตารางสีที่ใช้อ้างอิงภาพในระบบ RGB

ในปริภูมิเวคเตอร์ขนาดหนึ่งมิตินั้นการเราสามารถหาได้จากการประมาณค่าตัวแทนเช่นในการที่แสดงข้างล่าง เราประมาณว่าทุกค่าของเวคเตอร์ที่มีค่าน้อยกว่าหรือเท่ากับ –2 จะให้มีค่าเป็น –

3 ทุกค่าที่มีค่ามากกว่า –2 แต่น้อยกว่าหรือเท่ากับ 0 จะให้มีค่าเท่ากับ –1 ทุกค่าที่มากกว่า 0 แต่น้อยกว่าหรือเท่ากับ 2 จะให้มีค่าเท่ากับ 1 และถ้ามีค่ามาก 2 จะมีค่าเท่ากับ 3



รูปที่ 2.12 แสดงตัวอย่างการทำ Vector Quantization

ดังที่เห็นในรูปที่ 2.12 จะพบว่าเราจะเพียงใช้ 2 บิตในการแทนจุดต่างๆจากที่ต้องใช้ถึง 3 บิต ในการอธิบายให้ครบทุกๆจุดบนเวคเตอร์ดังกล่าว จุดที่ใช้แทนเวคเตอร์ ณ ตำแหน่งที่อยู่ใกล้ๆนั้น เรียกว่า เวคเตอร์ตัวแทน (Code Vectors) กลุ่มของเวคเตอร์ตัวแทนดังกล่าวจะเรียกว่า ชุดตัวแทน (Code Book) จากกราฟข้างบนจะได้ชุดตัวแทนชี้ในตารางที่ 2.1

ตารางที่ 2.1 แสดงตัวอย่างของชุดตัวแทน (Code Book)

Code Vectors	vector
00	-3
01	-1
10	1
11	3

ถ้าปริภูมิเวคเตอร์มีขนาดมากกว่า 1 มิติ (ตั้งแต่ 2 มิติขึ้นไป)นั้นจะเริ่มที่การรับค่าของจำนวนเวคเตอร์ที่ต้องการหลังจากผ่านกระบวนการแล้วจากนั้นจึงเข้าสู่กระบวนการคำนวณ

ถ้าเวคเตอร์ที่จะนำหามีจำนวนทั้งหมด M เวคเตอร์

$$T = \{x_1, x_2, \ldots, x_M\}$$

เมื่อ T คือจำนวนเวคเตอร์ทั้งหมดที่มีอยู่ในฐานข้อมูลที่จะนำมาทำการหาชุดตัวแทน ถ้าเวคเตอร์ที่เข้ามาเป็นค่าของ Histogram ของภาพที่เข้ามา ดังนั้นค่าของเวคเตอร์ที่จะนำมาหาแต่ละตัวจะมีขนาด k มิติ

$$x_m = (x_{m,1}, x_{m,2}, \ldots, x_{m,k})$$

$$x_m = (x_{m,1}, x_{m,2}, \ldots, x_{m,k})$$

จำนวนของเวคเตอร์ตัวแทน (Code Vectors) ที่ต้องการมีจำนวน N จะได้ชุดเวคเตอร์ (Code Book) คือ

$$C = \{c_1, c_2, \ldots, c_N\}$$

เวคเตอร์ตัวแทนแต่ละเวคเตอร์มีจำนวน k มิติ

$$c_n = (c_{n,1}, c_{n,2}, \ldots, c_{n,k})$$

การหาว่าเวคเตอร์ใดใช้เวคเตอร์ตัวแทนใดนั้นสามารถหาได้จากการหาระยะห่างระหว่างเวคเตอร์กับเวคเตอร์ตัวแทนในชุดเวคเตอร์ เวคเตอร์ตัวแทนที่มีค่าความห่างน้อยที่สุดจะใช้เป็นเวคเตอร์ตัวแทนของเวคเตอร์นั้น การหาระยะห่างระหว่างสองเวคเตอร์ใช้สมการที่ (2.11)

$$D = \|x_m - Q(x_m)\|^2 \tag{2.11}$$

เมื่อได้ค่าของความต่างแล้วเราจะแบ่งส่วนของเวคเตอร์ที่ใกล้เคียงกับเวคเตอร์ตัวแทนออกเป็นกลุ่มๆแล้ว

$$Q(x_m) = c_n \text{เมื่อ} x_m \text{อยู่ในกลุ่มของ} c_n \tag{2.12}$$

ค่าความต่างกำลังสองระหว่างชุดเวคเตอร์กับเวคเตอร์ทั้งหมดสามารถหาได้จากสมการ

$$D_{ave} = \frac{1}{Mk}\sum_{m=1}^{M}\|x_m - Q(x_m)\|^2 \quad \text{เมื่อ } \|e\|^2 = e_1^2 + e_2^2 + \ldots + e_k^2 \tag{2.13}$$

2.4 การใช้ความเหมือนระหว่างเวคเตอร์ในการสืบค้น

โดยที่กล่าวไปแล้วการที่เวคเตอร์หนึ่งๆนั้นจะมีขนาดแค่เพียงสองมิติเท่านั้นจึงทำให้การหาค่าความต่างระหว่างเวคเตอร์โดยใช้สมการยูครีเดียนจึงสามารถได้ง่ายและมีความถูกต้องสูง แต่ถ้าเวคเตอร์ที่มีขนาดมากกว่าสองมิติการจะอ้างอิงถึงเวคเตอร์นั้นโดยใช้เวคเตอร์ต้นแบบในการหา[2] เราหาความเหมือนกันระหว่างเวคเตอร์ที่เข้ามากับเวคเตอร์ที่มีอยู่โดยจะต้องใช้สมการหาความ

เหมือนของเวคเตอร์ จากนั้นนำมาเรียงลำดับของความเหมือนว่าเวคเตอร์ใดใกล้เวคเตอร์ที่เข้ามามากที่สุด ค่าที่เรียงลำดับจะเป็นคำตอบในการค้นหาไฟล์ที่ใกล้เคียงที่ต้องการ

ถ้าเวคเตอร์ที่เข้ามาเป็นค่าของ Histogram ของภาพที่เข้ามา ดังนั้นค่าของเวคเตอร์ที่จะนำมาหาแต่ละตัวจะมีขนาด k มิติ

$$x_m = (x_{m,1}, x_{m,2}, \ldots, x_{m,k})$$

และเวคเตอร์ต้นแบบที่ใช้ในการหาคือ q ต้องมีขนาดเท่ากับ k มิติด้วย

$$q = (q_1, q_2, \ldots, q_k)$$

การหาความเหมือนจะใช้สมการดังนี้

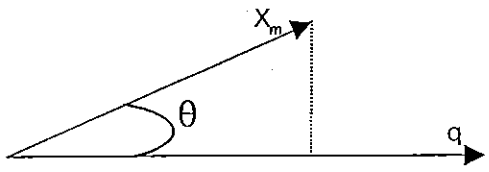
$$sim(x_m, q) = \frac{\vec{x}_m * \vec{q}}{|\vec{x}_m| \times |\vec{q}|} \tag{2.14}$$

หรือ

$$sim(x_m, q) = \frac{\sum_{l=1}^{k} \vec{x}_{m,l} \times \vec{q}_l}{\sqrt{\sum_{l=1}^{k} \vec{x}^{\,2}_{m,l}} \times \sqrt{\sum_{l=1}^{k} \vec{q}^{\,2}_l}} \tag{2.15}$$

เมื่อ M คือจำนวนวเวคเตอร์ที่มีอยู่ในฐานข้อมูลทั้งหมด

$$m = \{1, 2, 3, \ldots, M\}$$

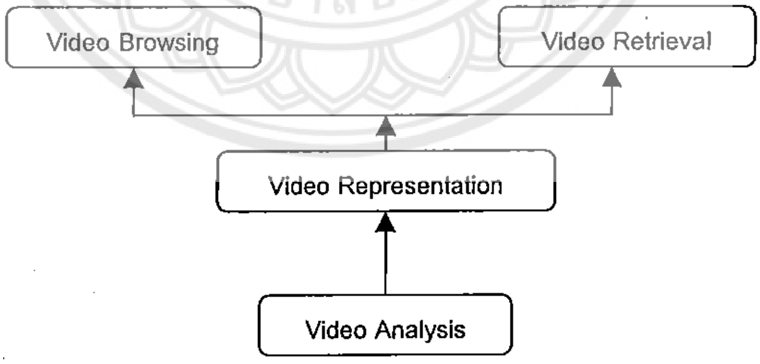


รูปที่ 2.13 แสดงการหาความเหมือนระหว่างเวคเตอร์

จากสมการด้านบนและรูปที่ 2.13 จะพบว่าการหาค่าความเหมือนระหว่างสองเวคเตอร์จะมีเป็นค่าเดียวกันกับการหาค่าของเวคเตอร์ในฐานข้อมูลบนเวคเตอร์ต้นแบบที่เข้ามา ($x_m \cos\theta$) นั่นเอง

2.5 ระบบวีดีโอ (Video System)

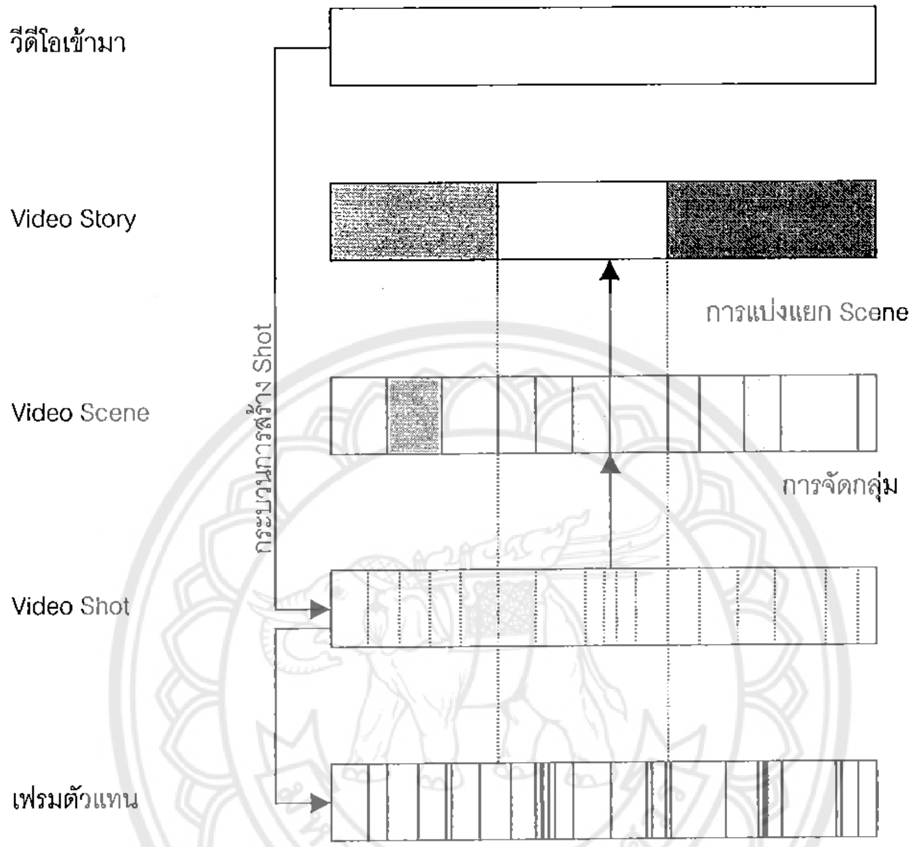
วีดีโอในระบบดิจิตินั้นคือการนำไฟล์ภาพหลายๆไฟล์มาเรียงต่อกันตามแนวของแกนของเวลา เมื่อเรานำไฟล์ภาพมาเรียงต่อกันไปแล้วแสดงภาพออกมาทีละภาพตามแนวของแกนเวลาก็จะเกิดภาพที่มีการเคลื่อนไหว ในกระบวนการที่เกี่ยวกับดิจิตอลวีดีโอในถูกแบ่งออกเป็น 4 หัวข้อหลักๆ การวิเคราะห์วีดีโอ (Video Analysis) เป็นกระบวนการจัดการเกี่ยวกับวิธีที่จะนำวีดีโอที่ถ่ายมาแปลงในรูปของอนาล๊อกมาแปลงให้เป็นไฟล์วีดีโอในระบบดิจิตอล การแสดงไฟล์ดิจิติลวีดีโอ (Video Representation) คือการแปลค่าไฟล์วีดีโอที่เก็บอยู่ในรูปแบบของตัวเลขให้ออกมาเป็นภาพเพื่อที่จะนำไปแสดงผล การแสดงเนื้อหาโดยคราวๆของไฟล์วีดีโอ (Video Browsing) เป็นการนำเทคนิคกระบวนการในการแสดงไฟล์วีดีโอมาใช้เพื่อแสดงเนื้อหาคราวๆของไฟล์วีดีโอนั้นๆ การสืบค้นไฟล์วีดีโอ (Video Retrieval) เป็นการค้นหาไฟล์วีดีโอที่ต้องการ ในปัจจุบันงานวิจัยที่มีอยู่ส่วนใหญ่จะเกี่ยวข้องกับ การวิเคราะห์วีดีโอและการแสดงไฟล์ดิจิตอลวีดีโอ ซึ่งเป็นส่วนแรกๆในกระบวนการที่เกี่ยวข้องกับไฟล์วีดีโอ แต่ยังไม่ค่อยมีการวิจัยที่เกี่ยวกับการแสดงเนื้อหาแบบคราวๆและการสืบค้นไฟล์วีดีโอนัก ซึ่งทั้งสองอย่างจะช่วยให้ค้นหาส่วนที่ต้องการไฟล์วีดีโอต่างๆได้อย่างรวดเร็วขึ้น



รูปที่ 2.14 แสดงลำดับชั้นของกระบวนการที่เกี่ยวข้องกับเนื้อหาของวีดีโอ

จากรูปที่ 2.14 จะพบว่าส่วนที่อยู่บนสุดของไดอะแกรมจะเป็นการแสดงไฟล์ดิจิตอลวีดีโอแบบคราวๆ และการสืบค้นไฟล์วีดีโอจะช่วยให้สามารถเข้าถึงเนื้อหาไฟล์วีดีโอได้ดีกว่า การนำไฟล์วีดีโอมาแสดงกับการวิเคราะห์วีดีโอ ซึ่งอยู่ในส่วนล่างของไดอะแกรม การสืบค้นไฟล์วีดีโอจะใช้

คำถามที่ผู้ใช้ป้อนเข้ามาเพื่อหาว่ามีไฟล์วีดีโอที่ตรงกับความต้องการหรือไม่ ในขณะที่การแสดงไฟล์วีดีโอแบบคราวๆจะเป็นการนำเนื้อหาของไฟล์มาแสดงให้ผู้ใช้เลือกว่าจะเข้าถึงไฟล์วีดีโอส่วนไหน



รูปที่ 2.15 แสดงขั้นตอนในวิเคราะห์วีดีโอ

ตัวอย่างของลักษณะที่คล้ายๆกันก็คือ ถ้าหนังสือที่มีจำนวนหน้าทั้งหมด 1000 หน้า ผู้อ่านสามารถใช้สารบัญเพื่อที่จะหาว่าบทหรือส่วนในที่เข้าต้องการใช้งาน แต่ถ้าเขามีคำถามอยู่ในใจคำถามหนึ่งแล้วต้องการหาส่วนที่เกี่ยวข้องกับคำถามนั้นก็สามารถค้นหาได้ในดรรชนีแล้วไปยังส่วนหรือหน้าที่เกี่ยวข้องกับคำถามนั้นๆ ในหนังสือที่มีจำนวนหน้าไม่มากสารบัญสามารถช่วยในการสำรวจและดรรชนีสามารถช่วยในการสืบค้น ซึ่งทั้งสองมีความสำคัญพอๆกันในการเข้าถึงเนื้อหาของหนังสือเล่มนั้น แต่ในวีดีโอนั้นจะไม่มีทั้ง สารบัญ และดรรชนี จึงได้มีการคิดค้นวิธีที่จะมาทดแทนสารบัญและดรรชนี เพื่อช่วยในการเข้าถึงเนื้อวีดีโอนั้นง่ายขึ้น

2.5.1 การวิเคราะห์วีดีโอ

กระบวนการในการนำวีดีโอมาทำเป็นไฟล์ดิจิตอลวีดีโอนั้นจะต้องทำการแบ่งภาพวีดีโอที่มีลักษณะเป็นคลื่นต่อเนื่องมาแบ่งออกเป็นภาพหลายภาพ ในกระบวนการดังกล่าวทำให้เกิดภาพจำนวนมากมาย จากนั้นนำมาเรียงต่อกันบนแกนเวลา เป็นการหาขอบเขตของไฟล์ภาพว่าจะทำการ

แบ่งช่วงที่มีความต่างของคลื่นแค่ไหนมาแบ่งออกเป็นหนึ่งภาพ กระบวนการดังกล่าวทำให้เกิดการกำหนดลำดับชั้นของการเข้าถึงไฟล์วีดีโอขึ้นมา

เริ่มจากการนำวีดีโอมาทำการวิเคราะห์หาลำดับของภาพซึ่งในกระบวนการที่เกี่ยวกับวีดีโอ แล้วเราจะเรียกภาพหนึ่งภาพที่มาจากไฟล์วีดีโอว่า เฟรม (Frame) แล้วแบ่งกลุ่มของเฟรมที่มีลักษณะใกล้เคียงกันออกเป็นกลุ่มที่เรารียกว่า 'Shot' ในแต่ Shot ก็จะมีการเลือกเฟรมที่เป็นตัวแทน (Key frame) ที่จะใช้อ้างอิงว่าเป็น Shot นั้นอยู่ด้วย ส่วนของไฟล์วีดีโอที่มีความยาวขึ้นมาในระดับหนึ่งซึ่งเรียกว่า Group of Shot หรือ Scene ซึ่งเกิดจากการนำ Shot ที่มีขนาดไม่ยาวนักมารวมกันนั้นเอง จากนั้นนำ Scene มาต่อกันจะได้ช่วงของวีดีโอจะได้ช่วงของวีดีโอที่เรียกว่า Story ในไฟล์ดิจิตอลวีดีโอนั้นเราไม่สามารถหาเฟรมตัวแทนในระดับ Scene และ Story ได้เพราะวีดีโอมีขึ้นยาวขึ้นก็จะมีความยากในการหาเฟรมตัวแทนจนถึงอาจจะไม่สามารถหาเฟรมตัวแทนได้

ลำดับชั้นของวีดีโอ

- ✓ Video shot หมายถึง ลำดับของเฟรมที่ไม่สามารถแบ่งย่อยลงได้ซึ่งถูกบันทึกจากกล้องเพียงตัวเดียวเท่านั้น
- ✓ Key frame หมายถึง เฟรมซึ่งสามารถแสดงเนื้อหาเด่นเฉพาะของแต่ละ shot อาจมี Key frame ที่ถูกคัดออกมาได้มากกว่า 2 key frame
- ✓ Video scene หมายถึง การรวมของ shot ที่มีความเกี่ยวข้องกันและเวลาใกล้ เคียงกันซึ่งอธิบายถึงเนื้อหาของเรื่องราวในระดับที่สูงกว่า shot ถ้า shot มีจุดเด่นที่คุณสมบัติทางกายภาพ scene ก็มีจุดเด่นด้านความสัมพันธ์

<u>2.5.1.1 การกำหนดขอบเขตของ shot (Shot Boundary detection)</u>

ถ้าจะต้องใช้ทั้ง Video clip มาทำการวิเคราะห์ อาจทำให้มีประสิทธิภาพต่ำ หากมีการนำมาย่อย Video clip ออกเป็น shot ก่อน ก็จะทำให้ได้ประโยชน์ ที่จะนำไปวิเคราะห์ ในระดับ shot ก่อนก็จะทำให้ได้ประโยชน์ ที่จะทำไปวิเคราะห์ ในระดับ shot โดยทั่วไป การกำหนดขอบเขตของ shot โดยอัตโนมัตินั้นมีอยู่ 5 วิธี คือ pixel based, statistics based, transform basc, feature base และ histogram base พบว่า วิธี histogram-base เป็นวิธีที่ได้รับความนิยม นักวิจัยส่วนใหญ่ ยืนยันว่าเป็นวิธีที่ มีความแม่นยำและรวดเร็ว

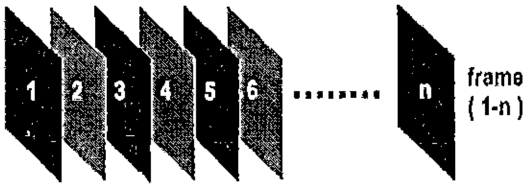
<u>2.5.1.2 การเลือกเฟรมหลัก (Key frame extraction)</u>

หลังจาก ขอบเขตของ shot ถูกกำหนด เราก็จะสามารถเลือก key frame ได้ โดยอาจจะเลือกเอา frame แรก หรือสุดท้ายของแต่ละ shot มาเป็น key frame

2.5.2 การแสดงวีดีโอ (Video Representation)

ดังรูปที่ 2.16 พิจารณาแต่ละ video frame เป็นวัตถุ 2 มิติ และแกนเวลา (temporal axis) เป็นมิติที่ 3 (third dimension) นั่นคือ Video Representation เป็นการทำแบบ 3 มิติ ให้เป็น ภาพ 2 มิติ

กระบวนการในขั้นตอนนี้จะเป็นวิธีในการจัดเก็บไฟล์วีดีโอในคอมพิวเตอร์ให้เหมาะสมที่สุด ซึ่งรวมถึงรูปแบบในการบีบอัดขนาดของไฟล์วีดีโอด้วย



รูปที่ 2.16 แสดงการเรียงตัวของเฟรมในไฟล์วีดีโอ

2.5.2.1 การแสดงเฟรมหลัก (Sequential key frame representation)

หลังจากได้รับ shot และเฟรมหลัก(key frame) ก็นำเฟรมหลัก(key frame)มาเรียงต่อเนื่องกันจากบนลงล่าง และจากซ้ายมาขวา วิธีพื้นฐานนี้ ทำงานได้ดีเมื่อจำนวนของเฟรมหลัก(key frame) มีขนาดน้อย แต่เมื่อ video clip มีขนาดยาว เราไม่ควรใช้วิธีนี้ เพราะมันไม่จับข้อมูลที่ฝังอยู่ใน video clip

2.6 การสืบค้นไฟล์วีดีโอในฐานข้อมูล

ไฟล์วีดีโอที่อยู่เก็บอยู่ในฐานข้อมูลนั้นจะถูกแบ่งออเป็น 3 ระดับ คือ Shot, Scene และ Story ในไฟล์วีดีโอระดับ Shot นั้นเราสามารถหาเฟรมตัวแทนที่จะสามารถอธิบายถึง Shot นั้นได้ไม่ยากนักเพราะทุกเฟรมใน Shot มักจะมีค่าใกล้เคียงกันจึงทำให้สามารถหาเฟรมตัวแทนได้ โดยอาจหาจากการนำเฟรมแรกหรือเฟรมสุดท้ายมาเป็นเฟรมตัวแทนเป็นต้น

2.6.1 การค้นหาโดยใช้คำสำคัญ **(Key word search)**

การสืบค้นข้อมูลโดยใช้คำสำคัญเป็นที่นิยมใช้กันมากเพราะประหยัดเวลาในการค้นหาเป็นอย่างมากการค้นหานั้นอาจจะได้ผลของการค้นหาเป็นกลุ่มของภาพโดยภาพแต่ละภาพนั้นจะมีคุณสมบัติเฉพาะของตัวเอง กล่าวคือ มีชื่อภาพ ชื่อผู้สร้าง วันเดือนปีที่ผลิต และแหล่งที่มา การค้นหา ภาพประเภทนี้อาจจะใช้เวลารวดเร็วถ้าจัดการข้อมูลโดยใช้การจัดการฐานข้อมูลที่มีประสิทธิภาพ เช่น SQL relational database language ซึ่งเป็นมาตรฐานสำหรับการจัดการฐานข้อมูล ตัวอย่างเช่น [1]

```
SELECT * FROM VideoDB
WHERE CATEGORY = 'Movie' AND SOURCE = 'MILAMAX'
AND (KEYWORD = 'John' OR KEYWORD = 'Action'
OR KEYWORD = 'Romance'
```

จากตัวอย่างดังกล่าวเราจะได้รูปภาพที่มีดรรชนีที่ประกอบไปด้วย ชื่อ VideoDB จัดอยู่ในประเภท Movie แหล่งที่มาคือ MILAMAX และมี คำสำคัญ คือ John, Action หรือ Romance

จากตัวอย่างดังกล่าวผู้ใช้สามารถสืบค้นข้อมูลโดยการใช้คำสำคัญได้ แต่อย่างไรก็ตามการสืบค้น โดยการใช้คำสำคัญอาจจะได้ผลลัพธ์ที่ไม่ตรงกับความต้องการนักเนื่องจากมีข้อมูลมากจนทำให้เกิดความซ้ำซ้อนขึ้น หรือการให้คำนิยามที่ไม่ครอบคลุมกับสิ่งที่จะค้นหาหรือภาพเหล่านั้นอาจมี คำสำคัญ ที่เหมือนกันแต่เนื้อหารายละเอียดแตกต่างกัน นอกจากการค้นหาข้อมูลโดยใช้ คำสำคัญ แล้วยังมีแนวทางอื่นในการค้นหาอีก คือ การใช้เนื้อหาของตัวไฟล์ในการค้นหา

2.6.2 การค้นหาโดยใช้คีย์เฟรม

การค้นหาโดยการใช้เฟรมตัวแทนเราจะพยายามอธิบายถึงวีดีโอต่างๆโดยเวคเตอร์ที่ใช้สำหรับอ้างอิงถึงวีดีโดนั้นๆ โดยการที่จะสามารถอาศัยเวคเตอร์ที่ใช้สำหรับอ้างอิงเฟรมตัวแทนในวีดีโอนั้นซึ่งเวคเตอร์ที่สร้างจะได้ต้องนำเฟรมที่เป็นเฟรมตัวแทนซึ่งเลือกได้จากเฟรมทั้งหมดของไฟล์วีดีโอดังรูปที่ 2.17 นั้นไปหาค่า Histogram เมื่อผ่านกระบวนการดังกล่าวก็จะได้เวคเตอร์ที่ใช้สำหรับอ้างอิงถึงไฟล์วีดีโอนั้นๆได้ จากนั้นจึงเอาเวคเตอร์ที่ได้ไปเก็บไว้ในฐานข้อมูลเพื่อไว้สำหรับเป็นตัวอ้างอิงเปรียบเทียบกับไฟล์ต้นแบบที่ผู้ใช้ป้อนเข้ามา

รูปที่ 2.17 แสดงเฟรมจากไฟล์วีดีโอในระดับ Shot

ดังนั้นจะเห็นว่ากระบวนการดังกล่าวจะเป็นกระบวนการที่เป็นกึ่งอัตโนมัติ ไม่จำเป็นต้องอาศัยคนที่มีความรู้ในการแยกแยะฐานข้อมูลมาทำการกรอกข้อมูลเลย เพราะในการทำเวคเตอร์ที่ใช้อธิบายถึงเฟรมตัวแทนของไฟล์วีดีโอนั้นสามารถสร้างได้โดยการนำภาพที่มีอยู่ไปหา Histogram เท่านั้นเอง

เมื่อผู้ใช้ทำการป้อนไฟล์วีดีโอต้นแบบเข้ามา เราจะนำของไฟล์วีดีโอที่เป็นไฟล์วีดีโอต้นแบบมาเทียบหาความต่างกับเวคเตอร์ที่มีอยู่ในฐานข้อมูล กำหนดให้เวคเตอร์ที่ป้อนเข้ามาและเวคเตอร์ในฐานข้อมูลมีขนาด k มิติ เราอาจสามารถหาค่าความต่างของเวคเตอร์ได้โดยสมการที่ (2.10)

ถ้าในฐานข้อมูลมีอยู่ทั้งหมด M ไฟล์ จะได้ว่า

$$D_1 = [\sum (f_{1q} - f_{1x})^2]^{1/2}$$

$$D_2 = [\sum (f_{2q} - f_{2x})^2]^{1/2}$$

$$\vdots$$

$$D_M = [\sum (f_{Mq} - f_{Mx})^2]^{1/2}$$

จากนั้นก็นำค่า D_1, D_2, …D_M มาทำการเรียงจากที่มีค่าน้อยไปหาค่ามากค่าของ D ที่น้อยที่สุดคือไฟล์ที่มีความใกล้เคียงกับไฟล์ต้นแบบมากที่สุด

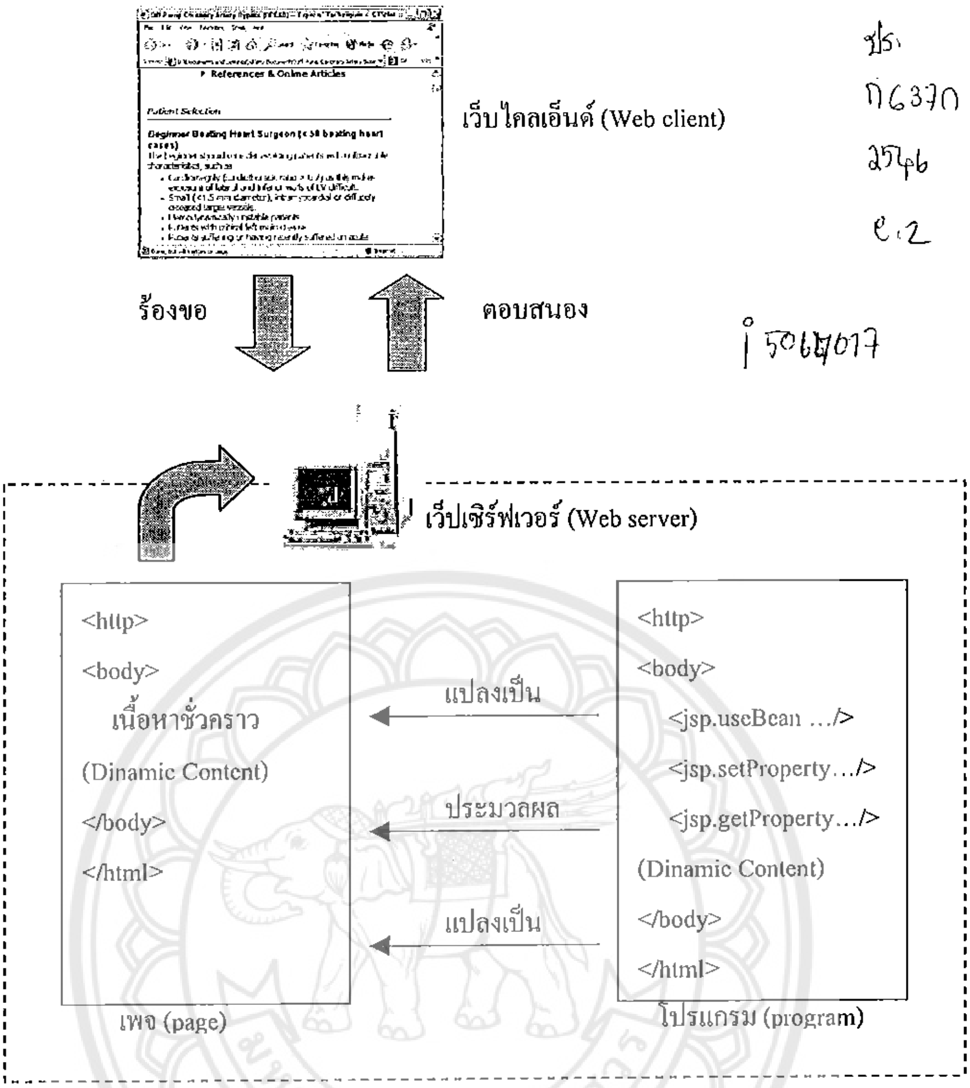
2.7 แนะนำเกี่ยวกับ JSP

JSP (Java Server Page) เป็นเทคโนโลยีสำหรับพัฒนาเว็บต่างๆ ที่มีเนื้อหาแบบไดนามิค (dynamic) ซึ่งสามารถเปลี่ยนแปลงได้และโต้ตอบกับผู้ใช้ ไม่เหมือนกับเพจของ HTML ตามปกติซึ่งจะบรรจุเนื้อหาคงที่ เพจของ JSP สามารถบรรจุเนื้อหาต่างๆ, การจำแนกผู้ใช้, ประเภทบราวเซอร์ของผู้ใช้, สารสนเทศต่างๆเกี่ยวกับผู้ใช้, และสิ่งต่างๆเกี่ยวกับเว็บเพจที่ผู้ใช้ชื่นชอบ [9]

เพจของ JSP จะบรรจุสมาชิกต่างๆของภาษาที่กำหนดเครื่องหมายมาตราฐาน เช่น แท็กต่างๆของ HTML ทั่วไป อย่างไรก็ตาม เพจของ JSP ที่บรรจุสมาชิกต่างๆ ของ JSP ใช้งานได้อย่างกว้างขวาง เช่น การนำสารสนเทศจากฐานข้อมูลออกมาดำเนินการหรือการลงทะเบียนผู้ใช้ เมื่อผู้ใช้ถามหาเพจของ JSP แล้วส่งผลกลับไปยังบราวเซอร์ (ดังรูป) JSP จะกำหนดสมาชิกมาตราฐานต่างๆใช้ประโยชน์สำหรับแอพพลิเคชันของเว็บใดๆ เช่น การเข้าถึงคอมโพเนนท์ต่างๆของ JavaBeans, การผ่านคอนโทรลระหว่างเพจต่างๆ, และการใช้สารสนเทศร่วมกันระหว่างเพจต่างๆที่ร้องขอ, และผู้ใช้ต่างๆโปรแกรมเมอร์สามารถเพิ่มความสามารถให้คำสั่งของ JSP โดยการสนับสนุนสมาชิกต่างๆ ที่ระบุแอพพลิเคชันที่ดำเนินงานต่างๆ เช่น การเข้าถึงฐานข้อมูลต่างๆ และ EJB(Enterprise JavaBeans), การส่งอีเมล์, และการสร้างเอกสารของ HTML เพื่อป้องกันข้อมูลของแอพพลิเคชันสมาชิกมาตราฐานต่างๆ และสมาชิกต่างๆ ที่สร้างขึ้นจะช่วยสร้างแอพพลิเคชันของเว็บที่มีประสิทธิภาพสูงมาก จากรูปที่ 2.18 แสดงตัวอย่างการสร้างเนื้อหาแบบไดนามิค

9 ธ.ค. 2547 ห้องสมุดคณะวิศวกรรมศาสตร์ 4700182

ปร.
ก6370
2546
ฉ.2
i 5014017



รูปที่ 2.18 แสดงการสร้างเนื้อหาแบบไดนามิคด้วยสมาชิกต่างๆของ JSP

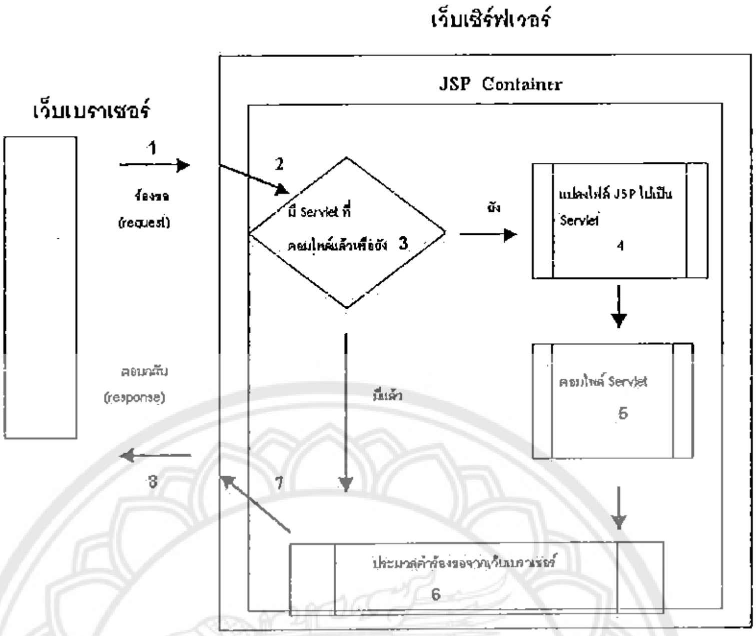
2.7.1 ข้อดีของ JSP

ทำงานโดยไม่ยึดติดแพลตฟอร์มใด ๆ JSP ได้สืบทอดคุณสมบติเด่นของ Java คือการทำงานโดยไม่ขึ้นกับแพลตฟอร์มหรือระบบปฏิบัติการใดๆไม่ว่าจะเป็นระบบ Windows, Linux, Unix, Mac OS ฯลฯ ดังนั้น เมื่อพัฒนาเว็บด้วย JSP ในแพลตฟอร์มหนึ่ง ก็สามารถย้ายไปใช้งานกับแพลตฟอร์มอื่นๆ ได้ไม่ยาก [6, 8, 9]

ใช้งาน Java API ได้หลากหลาย JSP สามารถเรียกใช้งาน Java API ได้หลากหลาย ซึ่ง Java API คือกลุ่มของคลาสที่ช่วยอำนวยความสะดวกในการใช้งานต่างๆ เช่น การจัดการเกี่ยวกับเน็ตเวิร์ค, การติดต่อฐานข้อมูล, การจัดการทางด้านกราฟิก, การจัดการเกี่ยวกับอ๊อบเจ๊กต์ต่างๆ, การับ-ส่งอีเมล เป็นต้น

นำคอมโพเนนต์กลับมาใช้ได้อีก ไม่ต้องเสียเวลาสร้างใหม่ เราสามารถนำ JavaBeans มาใช้งานร่วมกับสคริปต์ JSP เพราะ JavaBeans เป็นคอมโพเนนต์ที่เขียนขึ้นมาเพื่อใช้สำหรับทำงาน

หรือทำหน้าที่อย่างใดอย่างหนึ่งและสามารถนำกลับมาใช้งานได้เสมอ ดังนั้นเราไม่ต้องเสียเวลาเขียนคริปต์ JSP เพื่อใช้งานนั้นทุกครั้ง จึงพัฒนาเว็บไซต์เสร็จเร็วขึ้น ดังผังการทำงานรูปที่ 2.19



รูปที่ 2.19 แสดงผังการทำงานของ JSP

มีความยืดหยุ่นในการใช้งาน ในการเขียนสคริปต์ JSP เราสามารถกำหนดแท็กใหม่ขึ้นมาใช้งานให้เหมาะสมกับความต้องการได้ นอกจากนี้ภาษาที่ใช้เขียนสคริปต์ JSP ไม่ได้จำกัดเฉพาะภาษา Java เท่านั้น ตามหลักการแล้วเราสามารถใช้ภาษาอื่นๆในการเขียนสคริปต์ได้ รวมทั้งยังสามารถนำไปใช้งานร่วมกับ XML ได้อย่างมีประสิทธิภาพอีกด้วย [9]

ความปลอดภัยสูง JSP มีระบบการจัดการข้อผิดพลาดต่างๆ ไม่ว่าจะเป็นข้อผิดพลาดที่เกิดขึ้นในระหว่างการเขียนสคริปต์ หรือข้อผิดพลาดที่เกิดขึ้นเมื่อนำสคริปต์ JSP ไปใช้งานจริง (ระหว่างที่ผู้ชมเรียกดูและใช้บริการเว็บไซต์ที่พัฒนาขึ้นด้วย JSP) [6, 8]

2.7.2 โครงสร้างและขั้นตอนการทำงานของ JSP

สิ่งที่มีบทบาทสำคัญในการทำงานของ JSP ได้แก่ JSP Container (หรือเรียกอีกอย่างว่า JSP Engine) ซึ่งเป็นส่วนประกอบสำคัญที่อยู่ในเว็บเซิร์ฟเวอร์ เพราะทำหน้าที่ควบคุมและประมวลผลไฟล์ JSP ที่มีการร้องขอ (request) เข้ามา และตอบสนอง (response) คำร้องขอนั้นไปยังไคลเอนต์

ขั้นตอนการประมวลไฟล์ JSP [8] ทั้งหมด แบ่งเป็น 8 ขั้นตอน ดังนี้

1. ฝั่งไคลเอนต์ส่งคำร้องขอเอกสาร JSP ไปที่เว็บเซิร์ฟเวอร์

2. เว็บเซิร์ฟเวอร์ตรวจสอบการร้องขอ พบว่าเป็นไฟล์ JSP จึงส่งต่อไปให้ JSP Container

3. JSP Container ตรวจสอบว่าไฟล์ JSP ที่ร้องขอมา เคยแปลงเป็น Servlet และ คอมไพล์เป็นไฟล์ .class แล้วหรือยัง โดยดูว่ามีไฟล์ .class อยู่หรือเปล่า ถ้ายังไม่มี ก็จะกระโดดไปทำตามข้อ 4

ต่อ แต่ถ้ามีอยู่แล้ว ก็จะตรวจสอบอีกว่า หลังจากที่แปลงไฟล์ JSP เป็น Servlet และคอมไพล์เป็นไฟล์ .class ครั้งล่าสุดแล้ว ไฟล์ JSP นั้นมีการเปลี่ยนแปลงแก้ไขหรือเปล่า ถ้ามีการแก้ไข ก็จะกระโดดไปทำงานตามขั้นตอนที่ 4 ต่อเช่นกัน แต่ถ้าไม่มีการแก้ไข แสดงว่าไฟล์ JSP นั้นยังคงเดิมไม่เปลี่ยนแปลง จึงไม่มีความจำเป็นที่ต้องเปลี่ยนแปลงเป็น Servlet และคอมไพล์ใหม่ ก็ข้ามไปยังขั้นตอนข้อ 6 ได้เลย

4. JSP Container แปลงไฟล์ JSP เป็น Java Servlet

5. JSP Container คอมไพล์ไฟล์ Java Servlet เป็นไฟล์ .class

6. JSP Container ประมวลผลตามคำร้องขอนั้น

7. JSP container ส่งผลลัพธ์ที่ได้จากการประมวลผล ให้แก่เว็บเซิร์ฟเวอร์เว็บเซิร์ฟเวอร์ส่งผลลัพธ์นั้นไปยังไปยังไคลเอนต์หรือเว็บเบราเซอร์อีกทอดหนึ่ง

จากขั้นตอนการประมวลผลไฟล์ JSP ที่แจกแจงข้างต้น สามารถแบ่งออกได้เป็น 2 ช่วงหลักๆ คือ ช่วง translation และช่วง execution โดยช่วง translation ได้แก่ขั้นตอนข้อ 4 และขั้นตอนข้อ 5 ซึ่งเป็นการแปลงเอกสาร JSP (ไฟล์ .JSP) ให้เป็น Servlet (ไฟล์ .java) จากนั้นก็จะคอมไพล์ไฟล์ Servlet ให้เป็นไฟล์ .class ดังรูปที่ 2.18

ส่วนช่วง execution ได้แก่ขั้นตอนข้อ 6 ซึ่งเป็นการนำเอาไฟล์ .class ที่ได้จากการคอมไพล์มาประมวลผลหรือทำงานตามคำร้องขอจากไคลเอนต์นั่นเอง

ปกติแล้วกระบวนการทำงานในช่วง translation จะกินเวลาพอสมควร แต่โชคดีว่าช่วง translation จะไม่เกิดขึ้นทุกครั้งที่มีการร้องขอไฟล์ เพราะตราบใดที่ไฟล์ JSP ต้นฉบับไม่มีการเปลี่ยนแปลงอะไร เมื่อมีการร้องขอไฟล์เข้ามาใหม่ ก็ย่อมไม่มีความจำเป็นที่จะแปลงไฟล์เป็น Servlet และคอมไพล์เป็นไฟล์ .class อีก ระบบจะเข้าสู่ช่วง execution ทันทีโดยใช้ไฟล์ .class ที่มีอยู่แล้ว การทำงานจึงรวดเร็วยิ่งขึ้น แต่ถ้ามีการเปลี่ยนแปลงหรือแก้ไขไฟล์ JSP ใหม่ ก็จะต้องเขาสู่กระบวนการ translation ใหม่ทุกครั้ง ดังรูปที่ 2.20

เอกสาร JSP (.jsp)	→	Java Servle (.java)	→	Servlet ที่คอมไพล์แล้ว (.jsp)

รูปที่ 2.20 แสดงขั้นตอนการประมวลผลไฟล์ JSP

สรุปว่ากระบวนการ translation มีโอกาสจะเกิดขึ้นได้ 2 กรณี กรณีแรกคือ ไฟล์ JSP ที่ร้องขอมา เป็นไฟล์ใหม่ที่ยังไม่เปลี่ยนแปลงและคอมไพล์มาก่อน กับอีกกรณีคือ ไฟล์ JSP ที่ร้องขอมาเคยผ่านการแปลงและคอมไพล์มาแล้ว แต่ภายหลังมีการเปลี่ยนแปลงแก้ไขไฟล์ JSP นั้นไปจากเดิม

บทที่ 3

วิธีดำเนินงาน

เมื่อนำวีดีโอมาแปลงเป็นไฟล์ดิจิตอลวีดีโอแล้วเราจะทำการแบ่งไฟล์วีดีโอนั้นออกเป็น Shot, Scene และ Story เพื่อจะทำการแยกไปเก็บในฐานข้อมูลในแต่ละแบบ จากนั้นเราจะนำเฟรมซึ่งมีอยู่ด้วยกันหลายเฟรมในแต่ไฟล์วีดีโอในแต่ละระดับ ซึ่งในแต่ละเฟรมจะประกอบไปด้วยจุดสีจำนวนมากมาเรียงต่อกันจนเกิดเป็นภาพ จากที่กล่าวมาแล้วในทฤษฎีเรื่องระบบสีในบทที่ 2 มาแล้วว่าจุดสีแต่ละจุดจะถูกอธิบายด้วยค่าของคู่อันดับของสีในแกนต่างๆ อาทิเช่นในระบบ HSV จุดสีหนึ่งจุดก็จะถูกอธิบายด้วยค่าของสี (Hue), ค่าการเจือของสีขาว(Saturation) และค่าการเจือของสีเทา(Value) เราจะต้องนำเฟรมมาทำการสร้างเวคเตอร์สำหรับอ้างอิงถึงเฟรมนั้นๆเพื่อเอาไว้ใช้เปรียบเทียบกับเวคเตอร์อ้างอิงของไฟล์ต้นแบบที่ผู้ใช้ต้องการหา

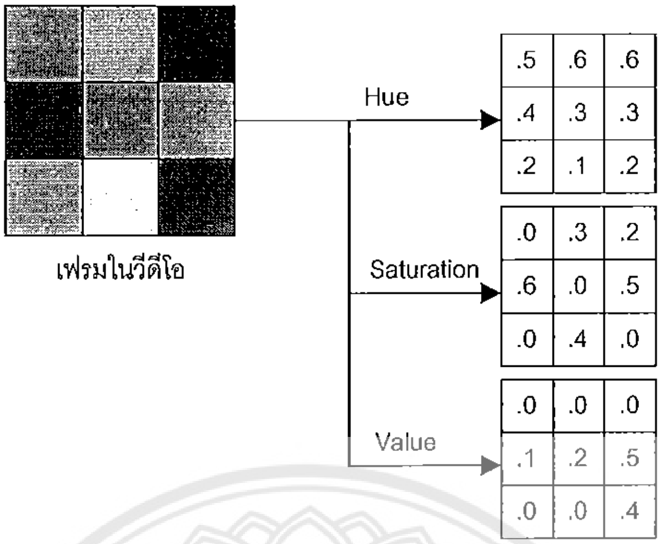
3.1 การนำไฟล์วีดีโอมาทำ Histogram

กระบวนการในการนำไฟล์วีดีโอมาทำ Histogram นั้นเราอาจนำเฟรมใดเฟรมหนึ่งของไฟล์วีดีโอในระดับนั้นมาหาค่าของ Histogram ก็ได้ โดยวิธีการก็จะเหมือนกับการทำ Histogram ของภาพสี คือ นำจุดสีที่มีอยู่ในเฟรมนั้นๆมาหาว่าอยู่ในช่วงใดของกราฟ Histogram โดยทำกับทุกจุดสีในเฟรมดังกล่าว โดยนำสมการที่ (2.7), (2.8) และ (2.9) มาใช้คำนวณหาว่าอยู่ในช่วงใดของกราฟ

เมื่อได้ค่าของ Histogram แล้วเราจะทำการ Normalize ค่า Histogram ที่ได้โดยการนำมาหาอัตราส่วนกัน กับค่าของจำนวนจุดสีที่มีอยู่ทั้งหมดในแต่ละเฟรมนั้น ๆ เราจะได้ Histogram ของเฟรมในไฟล์วีดีโอซึ่งจะอยู่ในรูปแบบของเวคเตอร์ที่มีขนาด nHxnSxnV มิตินั้นเอง

$$H = \{h_1, h_2, \ldots, h_k\} \text{ เมื่อ } k = nH \times nS \times nV \qquad (3.1)$$

ดังนั้นจะได้ดังตัวอย่างในรูปที่ 3.1



รูปที่ 3.1 แสดงตัวอย่างแบ่งเฟรมออกเป็น ส่วนของค่าในระบบสี HSV

ตัวอย่างการทำ Histogram ของเฟรมในไฟล์วีดีโอ ดังต่อไปนี้

กำหนดให้ค่าในแต่ละช่วงของระบบสีมีค่าเท่ากับหนึ่ง (A=1) และจำนวนหนวนช่วงที่ต้องการคือ 16,4,1 ในแกนของ H, S, V ตามลำดับ จะได้

$$scaleH = nH/A = 16/1 = 16$$

$$scaleS = nS/A = 4/1 = 4$$

$$scaleV = nV/A = 1/1 = 1$$

เมื่อได้ค่าสเกลของแกนต่างๆมาแล้วก็นำไปหาค่าอ้างอิงของแต่ละจุดสีว่าอยู่บนกราฟ

id_1=[ceil(0.5*16)] = 8

id_2=[ceil(0.6*16)+(16*((ceil(0.3*4)-1))] = 26

id_3=[ceil(0.6*16)+(16*((ceil(0.2*4)-1))] = 10

id_4=[ceil(0.4*16)+(16*(ceil(0.6*4)-1)+((16*4)*(ceil(0.1*1)-1))] = 39

id_5=[ceil(0.3*16)+((16*4)*(ceil(0.2*1)-1))] = 5

id_6=[ceil(0.3*16)+(16*((ceil(0.5*4)-1))+((16*4)*(ceil(0.5*1)-1))] = 21

id_7=[ceil(0.2*16)] = 4

id_8=[ceil(0.1*16)+(16*((ceil(0.4*4)-1))] = 18

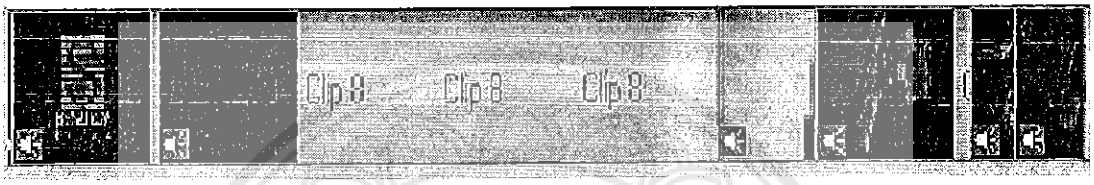
id_9=[ceil(0.2*16)+((16*4)*(ceil(0.4*1)-1))] = 4

จากนั้นนำค่าที่ได้ไปเขียนลงในกราฟเราจะได้ว่าค่ากราฟ Histogram ดังกล่าวคือ

$$H=\{0,0,0,2,1,\ldots,0\} \quad (3.2)$$

แล้วนำค่าที่ได้ไปทำการ Normalize ก็จะได้เวคเตอร์ที่ใช้อ้างอิงเฟรมดังกล่าวของไฟล์วีดีโอ

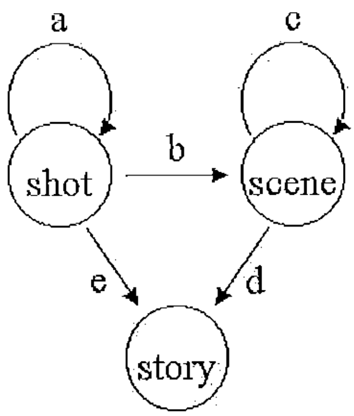
3.2 การสืบค้นไฟล์วีดีโอในฐานข้อมูล Shot Scene และ Story โดยวิธี



รูปที่ 3.2 แสดงตัวอย่างไฟล์วีโอในระดับ Scene

จากรูปที่ 3.2 เราจะเห็นว่าการหาเฟรมตัวแทนนั้นเราสามารถทำได้ในไฟล์วีดีโอระดับ Shot เท่านั้น ในวีดีโอระดับ Shot ทุกๆเฟรมในไฟล์วีดีโอจะมีลักษณะเหมือนกันหรือใกล้เคียงกันมากๆ ดังนั้นเมื่อนำเฟรมใดก็ตามในไฟล์วีดีโอไปทำการหา Histogram ก็ได้ค่า Histogram ที่ไม่ต่างกันมานัก อาจกล่าวได้ว่าการเลือกเฟรมตัวแทนนั้นสามารถนำเฟรมไหนก็ได้ไปเป็นเฟรมตัวแทนก็จะสามารถทำให้สามารถอธิบายถึงลักษณะของทุกๆเฟรมในไฟล์วีดีโอนั้นได้อย่างถูกต้องเหมือนกัน แต่ในไฟล์วีดีโอระดับ Scene และ Story นั้นเราไม่สามารถหาเฟรมที่จะมาเป็นเฟรมตัวแทนที่จะสามารถอธิบายถึงไฟล์วีดีโอนั้นได้อย่างถูกต้อง ดังนั้นกระบวนการสืบค้นไฟล์วีดีโอจึงเกิดขึ้นในระดับของไฟล์วีดีโอ Shot เท่านั้น ไม่สามารถหาได้ในระดับ Scene และ Story ได้

ในกระบวนการสร้างเวคเตอร์ที่ใช้สำหรับอธิบายถึงไฟล์วีดีโอในระดับ Shot, Scene และ Story โดยเราจะนำทุกๆเฟรมในไฟล์วีดีโอนั้นๆไปทำ Histogram แล้วนำไปสร้างเวคเตอร์ AVI ซึ่งหาได้จากความบ่อยครั้งในการปรากฏของเฟรมหนึ่งๆในไฟล์วีดีโอนั้นๆ ด้วยกระบวนการดังกล่าวทำให้เราสามารถ อธิบายถึงคุณลักษณะของไฟล์วีดีโอนั้นๆได้ และยังสามารถใช้กระบวนการในการสืบค้นนี้ในไฟล์วีดีโอซึ่งอยู่ในต่างระดับกันได้ดังรูปที่ 3.3 (a) สืบค้นไฟล์วีดีโอในระดับ Shot ด้วยกัน (shot-to-shot) (b) สืบค้นไฟล์วีดีโอในระดับ Shot ไประดับ Scene (shot-to-scene) (c) สืบค้นไฟล์วีดีโอในระดับ Scene ด้วยกัน (scene-to-scene) (d) สืบค้นไฟล์วีดีโอในระดับ Scene ไประดับ Story (scene-to-story) (e) สืบค้นไฟล์วีดีโอในระดับ Shot ไประดับ Story (shot-to-story)



รูปที่ 3.3 แสดงรูปแบบการสืบค้นไฟล์วีดีโอแบบ AVI

ด้วยโครงสร้างของการค้นหาไฟล์วีดีโอที่ทำให้เราสามารถค้นหาไฟล์วีดีโอที่อยู่ในระดับสูงกว่าด้วยไฟล์วีดีโอที่อยู่ในระดับต่ำกว่าได้ ตัวอย่างเช่น เราต้องการหาไฟล์วีดีโอของข่าว ข่าวหนึ่ง โดยเราป้อนคำถามหรือ ไฟล์วีดีโอต้นแบบที่เป็นส่วนหนึ่งของข่าวนั้นๆ เท่านั้นก็จะสามารถหาได้

3.3 กระบวนการในการสร้างเวคเตอร์ AVI แม่แบบ (AVI template)

ก่อนที่เราจะทำการสร้าง เวคเตอร์ AVI เราต้องทำการสร้างต้นแบบของเวคเตอร์ในแต่ละช่วงของแม่แบบ (template) โดยเราจะนำค่าของ histogram ของเฟรมทุกเฟรมในฐานข้อมูลมาคัดเลือกตัวแทนของกลุ่ม เพื่อละเป็นช่วงๆหนึ่งของเวคเตอร์แม่แบบ โดยใช้กระบวนการประเมินค่าตัวแทนของเวคเตอร์ (Vector Quantization) โดยอัลกรอลิทึมที่ใช้ในงานวิจัยนี้คือ LBG Design Algorithm ซึ่งมีกระบวนการดังนี้

1. รับจำนวนของเวคเตอร์ตัวแทน (Code Vector) ทั้งหมดที่ต้องการ หลังจากผ่านกระบวนการมาแล้วจะได้ เวคเตอร์ตัวแทนทั้งหมดคือ 2^n เมื่อ n = (1, 2, 3, …)
2. คำนวนหาจุดกึ่งกลางระหว่างเวคเตอร์ทั้งหมด

$$c = \frac{1}{M}\sum_{m=1}^{M} x_m \qquad (3.3)$$

หาค่าระหว่างเวคเตอร์ที่ได้กับเวคเตอร์ทั้งหมด

$$D_{ave} = \frac{1}{Mk}\sum_{m=1}^{M} \left\| x_m - Q(x_m) \right\|^2 \qquad (3.4)$$

3. เพิ่มจำนวนของเวคเตอร์ตัวแทน โดย i = 1, 2, …, N

$$c_i = (1+0.001)c_i \tag{3.5}$$

$$c_{N+i} = (1-0.001)c_i \tag{3.6}$$

4. เลื่อนตำแหน่งของเวคเตอร์ตัวแทนให้อยู่

I) หาระยะห่างระหว่าง เวคเตอร์ทั้งหมดกับเวคเตอร์ตัวแทน

$$\|x_m - c_n\|^2 \tag{3.7}$$

m = 1, 2, …,M

n = 1, 2, …,N

ค่าใดมีค่าใกล้ที่สุดระหว่างเวคเตอร์ที่มีอยู่กับเวคเตอร์ตัวแทนก็แบ่งกลุ่ม

$$Q(x_m) = c_n \tag{3.8}$$

II) เลื่อนตำแหน่งเวคเตอร์ตัวแทน

$$c_n = \frac{\sum Q(x_m) = c_n x_m}{\sum Q(x_m) = c_n^{\ 1}} \tag{3.9}$$

III) i = i + 1

IV) หาระยะห่างเฉลี่ยของเวคเตอร์แต่ละกลุ่ม

$$D_{ave} = \frac{1}{Mk}\sum_{m=1}^{M}\|x_m - Q(x_m)\|^2 \tag{3.10}$$

$$\text{ถ้า } \frac{D_{ave(\text{เก่า})} - D_{ave(\text{ใหม่})}}{D_{ave(\text{เก่า})}} > 0.001 \text{ ทำ i} \tag{3.11}$$

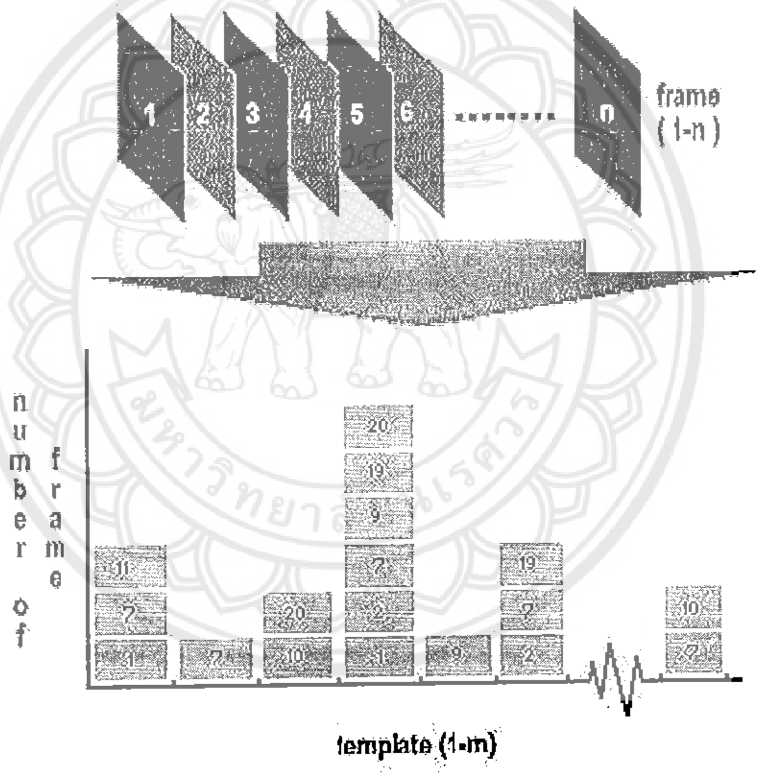
5. ทำข้อ 3 และ 4 จนกว่าจะได้จำนวนเวคเตอร์ตัวแทนมากกว่าหรือเท่ากับที่ต้องการ

มหาวิทยาลัยนเรศวร

3.4 สร้างเวคเตอร์ AVI (AVI vector)

เมื่อได้เวคเตอร์ตัวแทนที่ใช้เป็นแม่แบบในการสร้างเวคเตอร์แล้วเราจะทำการสร้างเวคเตอร์ AVI โดยนำ Histogram เฟรมทุกเฟรมในไฟล์วีดีโอ มาเทียบกับเวคเตอร์แม่แบบว่าใกล้เคียงกับเวคเตอร์นั้น ซึ่งกระบวนการนี้คล้ายกับการทำ Histogram นั่นเอง ดังรูปที่ 3.4 แต่ จากเดิมที่จะทำการเพิ่มค่าในช่วงของค่าที่ใกล้ที่สุดเพียงค่าเดียว เท่านั้น เราจะทำการใส่ค่าเข้าไปในช่วงของค่าที่ใกล้เคียงทั้งหมด 5 ช่วง เพื่อจะช่วยลดการเกิดช่วงของค่าที่มีความถี่เป็นศูนย์

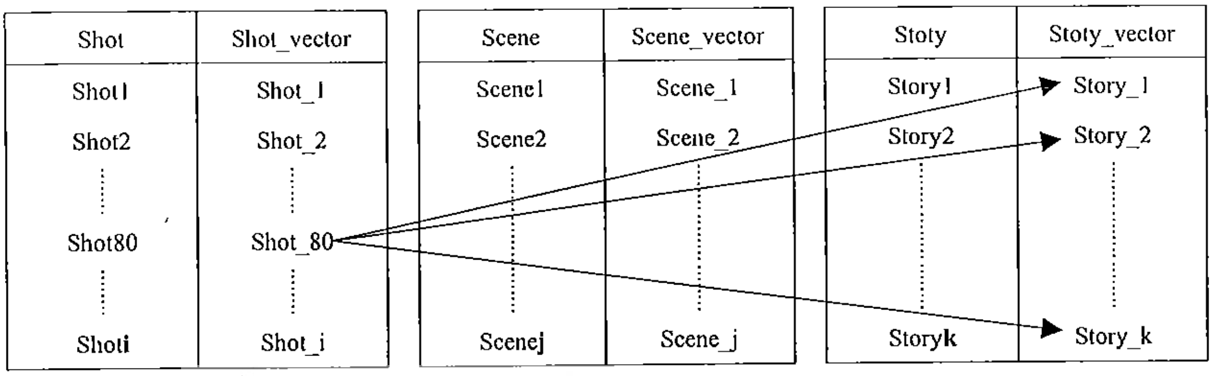
หลังจากได้ AVI เวคเตอร์มาแล้วก็นำเวคเตอร์นั้นมาเปรียบเทียบกับเวคเตอร์ในฐานข้อมูล shot, scene หรือ story ซึ่งจะเปรียบเทียบกับเวคเตอร์ใดนั้น ก็ขึ้นอยู่กับ ผู้ใช้เป็นผู้เลือกว่าต้องการผลลัพธ์ เป็น shot, scene หรือ story



รูปที่ 3.4 แสดงการเรียงของ Histogram ใน AVI

เช่น ผู้ใช้ต้องการนำ Shot80 เป็นตัวต้นแบบ แล้วนำมาค้นหาไฟล์ในระดับ story ซึ่งมีอยู่ในฐานข้อมูลดังตารางที่ 3.1

ตารางที่ 3.1 แสดงตัวอย่างการเลือกเวคเตอร์ที่ใช้ในการสืบค้นในไฟล์วีดีโอต่างระดับกัน



จากนั้นเป็นการหา วีดีโอไฟล์ที่ดีที่สุดจากการสืบค้น

ขั้นแรก นำเวคเตอร์ของ Shot80 ซึ่งก็คือ Shot_80 ไปเปรียบเทียบกับเวคเตอร์ ในฐานข้อมูล Story

$$d1 = \frac{\overrightarrow{Story_1} \bullet \overrightarrow{Shot_{80}}}{\left|\overrightarrow{Story_1}\right| \times \left|\overrightarrow{Shot_{80}}\right|}$$

$$d2 = \frac{\overrightarrow{Story_2} \bullet \overrightarrow{Shot_{80}}}{\left|\overrightarrow{Story_2}\right| \times \left|\overrightarrow{Shot_{80}}\right|}$$

$$d3 = \frac{\overrightarrow{Story_3} \bullet \overrightarrow{Shot_{80}}}{\left|\overrightarrow{Story_3}\right| \times \left|\overrightarrow{Shot_{80}}\right|}$$

$$\vdots$$

$$dk = \frac{\overrightarrow{Story_k} \bullet \overrightarrow{Shot_{80}}}{\left|\overrightarrow{Story_k}\right| \times \left|\overrightarrow{Shot_{80}}\right|}$$

ขั้นที่ 2 หลังจากได้ค่า d1, d2, d3, d4,… , dk แล้ว เราก็จะทำการลำดับค่า ว่าค่าของเวคเตอร์คู่ใดมีความแตกต่างกันน้อยที่สุด ซึ่งก็จะเป็นเวคเตอร์ของ file VDO ที่ถูกต้องที่สุด เช่น

d8 > d15 > d2 > d50 > ……

บทที่ 4

การทดลองการสืบค้นและส่วนที่เกี่ยวข้องกับงานวิจัย

การทดลองในบทนี้จะเป็นการค่าของจำนวนช่วงของ Histogram ที่เหมาะสมในการนำไปทำค่า Histogram ในระบบสี HSV และการทดลองในการสืบค้นไฟล์วีดีโอจากฐานข้อมูลเปรียบเทียบระหว่างวิธีการใช้เฟรมตัวแทนกับวิธีการ AVI ในระดับ Shot และการสืบค้นไฟล์วีดีโอในระดับ Scene และ Story

4.1 ผลการทดลองการแบ่งช่วงของ Histogram

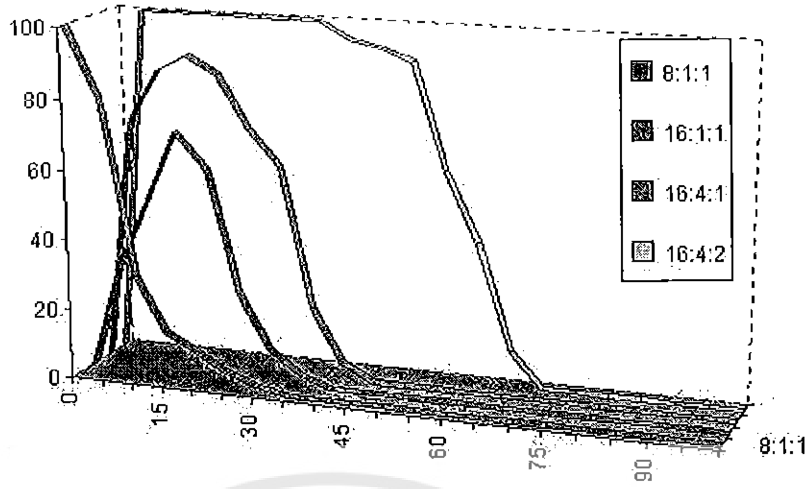
เพื่อที่จะหาจำนวนช่วงของสีในระบบสี HSV ที่จะมาใช้ในการทำ Histogram ที่เหมาะสมในการนำค่าของเฟรม ในไฟล์วีดีโอมาทำการสร้างเวคเตอร์ Histogram โดยที่ค่าที่ได้นั้นไม่ควรมีช่วงที่ไม่มีสีปรากฏอยู่หรือค่า Histogram มากเกินไป และสามารถบรรยายคุณลักษณะของเฟรมภาพได้ใกล้เคียง

ทำการทดลองโดยสุ่มเลือกนำไฟล์วีดีโอจากฐานข้อมูล shot จำนวน 20 ไฟล์ มาจาก Histogram จำนวนเฟรมที่ได้คือ 1883 เฟรม ขนาดของแต่ละเฟรมคือ 320x240 เราจะทำการเพิ่มจำนวนช่วง ดังตารางที่ 4.1

ตารางที่ 4.1 จำนวนของช่วงที่ใช้ในการทดลองการทำ Histogram

nH	nS	nV
8	1	1
16	1	1
16	4	1
16	4	2

เมื่อนำค่า Histogram ของภาพมาลองทำการสืบค้น ผลปรากฏว่าค่าช่วงของสี 8:1:1 จะได้ภาพที่มีความถูกต้องน้อยที่สุดและจะมากขึ้นเมื่อใช้จำนวนช่วงของสีเป็น 16:1:1 และ 16:4:1 ตามลำดับแต่เมื่อเราทำการเพิ่มจำนวนช่วงของสีในแนวแกน V จะพบว่าค่าความถูกต้องที่ได้มีความใกล้เคียงกัน



รูปที่ 4.1 กราฟการทดลอง

จากกราฟการทดลองรูปที่ 4.1 ค่าของจำนวนช่วง 8:1:1 โอกาสที่จะเกิดช่วงที่มีค่าของ Histogram เป็นศูนย์มีค่าน้อยมาก แต่นำไปสืบค้น จะได้โดยหาภาพหรือเฟรมที่ใกล้เคียงจะได้ที่ถูกต้องน้อยกว่า 16:4:2 ที่มีจำนวนช่วงที่เป็นศูนย์มากที่สุด

ในขณะที่จำนวนช่วง 16:1:1 และ 16:4:1 นั้นค่าของจำนวนช่วงที่เป็นศูนย์ที่พบมีเปอร์เซ็นต์ใกล้เคียงกันแต่จำนวนช่วง 16:4:1 จะสืบค้นได้ดีกว่าดังนั้นเราจึงใช้จำนวนช่วงของ Histogram 16:4:1 ในงานวิจัย

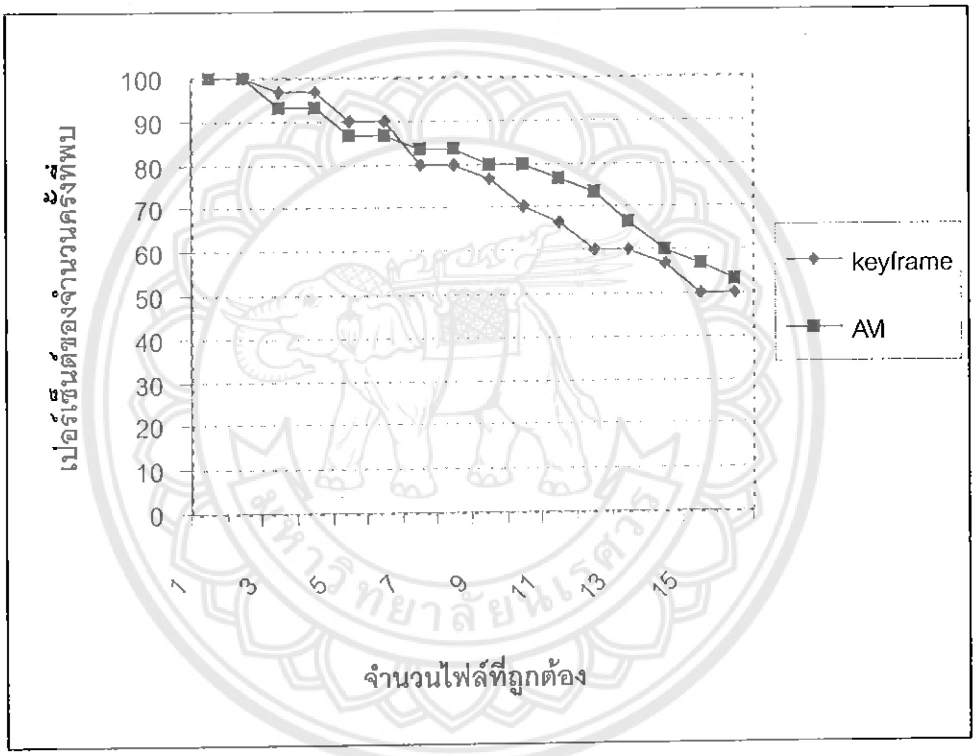
4.2 ผลการทดลองการค้นหาไฟล์วีดีโอในระดับ Shot

ในส่วนของการทดลองนี้เราจะทำการเปรียบเทียบ กระบวนการสืบค้นไฟล์วีดีโอจากฐานข้อมูล Shot โดยทำการเปรียบเทียบการสืบค้น โดยใช้อัลกอริทึม KFVI มาเปรียบเทียบกับการสืบค้น AVI โดยให้ไฟล์วีดีโอ จากฐานข้อมูลจำนวน 751 ไฟล์ จำนวนเฟรมทั้งหมดประมาณ 120,000 เฟรม จากข้อมูลในตารางที่ 4.2

การทดลองในการสืบค้นโดยเลือกเฟรมแรกจากไฟล์วีดีโอมาเป็นเฟรมตัวแทน ส่วน AVI ใช้จำนวน AVI template 512 ช่อง โดยการเพิ่มค่าในช่วง Histogram AVI ที่ใกล้เคียงจำนวน 5 ช่องต่อเฟรมหนึ่งเฟรมของไฟล์วีดีโอ โดยที่ Histogram ของแต่ละเฟรมใช้ 16:4:1 ในระบบสี HSV

ตารางที่ 4.2 จำนวนของ Shot ในแต่ละเรื่องใน Database

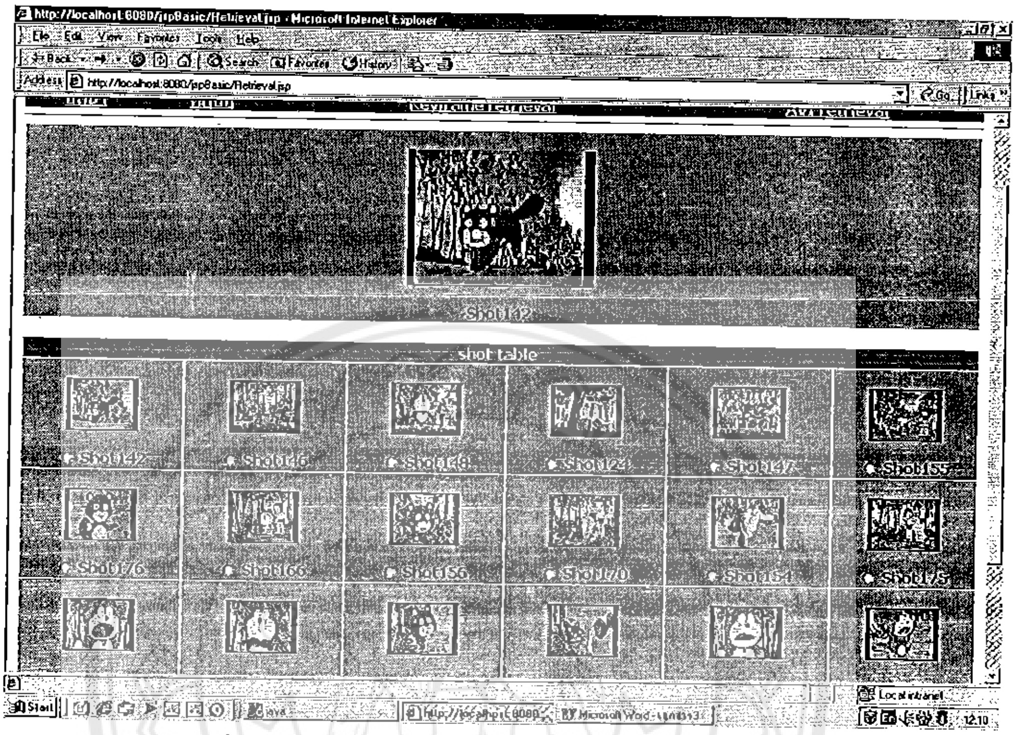
video	จำนวน	รวม	เฟรมทั้งหมด
เรื่อง 1	2	745	120,205
เรื่อง 2	69		
เรื่อง 3	106		
เรื่อง 4	568		



รูปที่ 4.2 ตารางการทดลองเปรียบเทียบ KFVI กับ AVI

จากกราฟการทดลองรูปที่ 4.2 โดยการทดสอบกับไฟล์ต้นแบบจำนวน 30 ไฟล์ ไฟล์ที่ได้รับจากการสืบค้นแต่ละครั้งในทั้งสองวิธีจะให้ผลที่ใกล้เคียงกัน มีค่าประมาณ 55 เปอร์เซนต์ที่ค่าของความต้องการที่ 16 ไฟล์วีดีโอ และมีค่าใกล้เคียงกันที่ค่าจำนวนไฟล์ที่ต้องการน้อยลงไม่ต่างกันมากนัก แสดงว่าวิธี AVI สามารถนำมาใช้แทนการสืบค้นแบบเฟรมตัวแทนได้

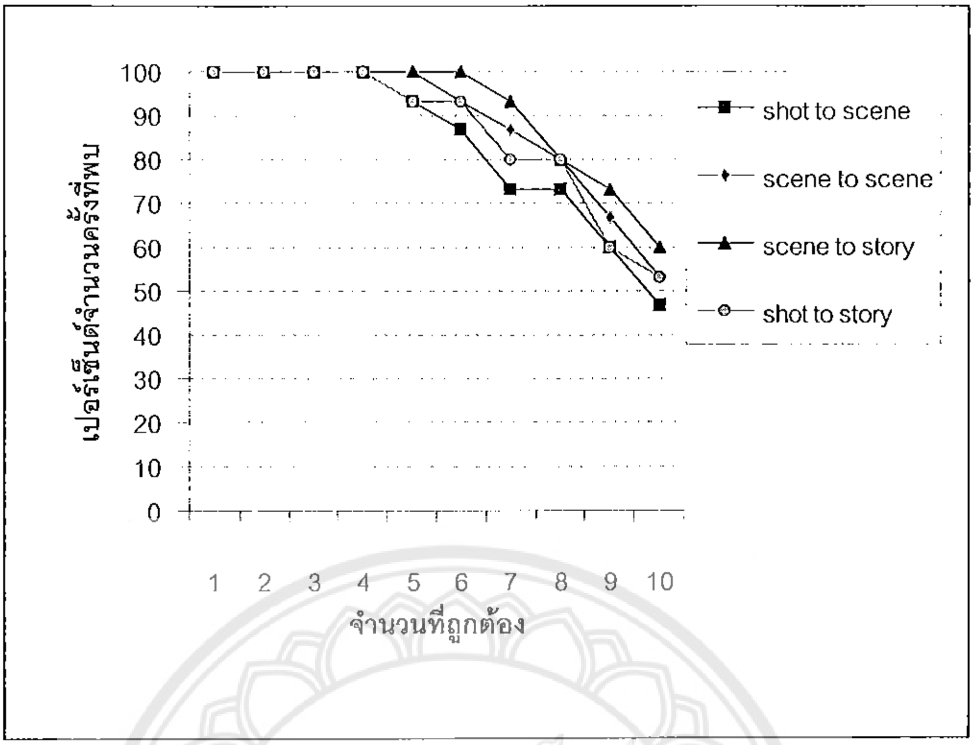
รูปที่ 4.3 แสดงถึงตัวอย่างการสืบค้นไฟล์วีดีโอในระดับ shot จากโปรแกรมที่สร้างขึ้น



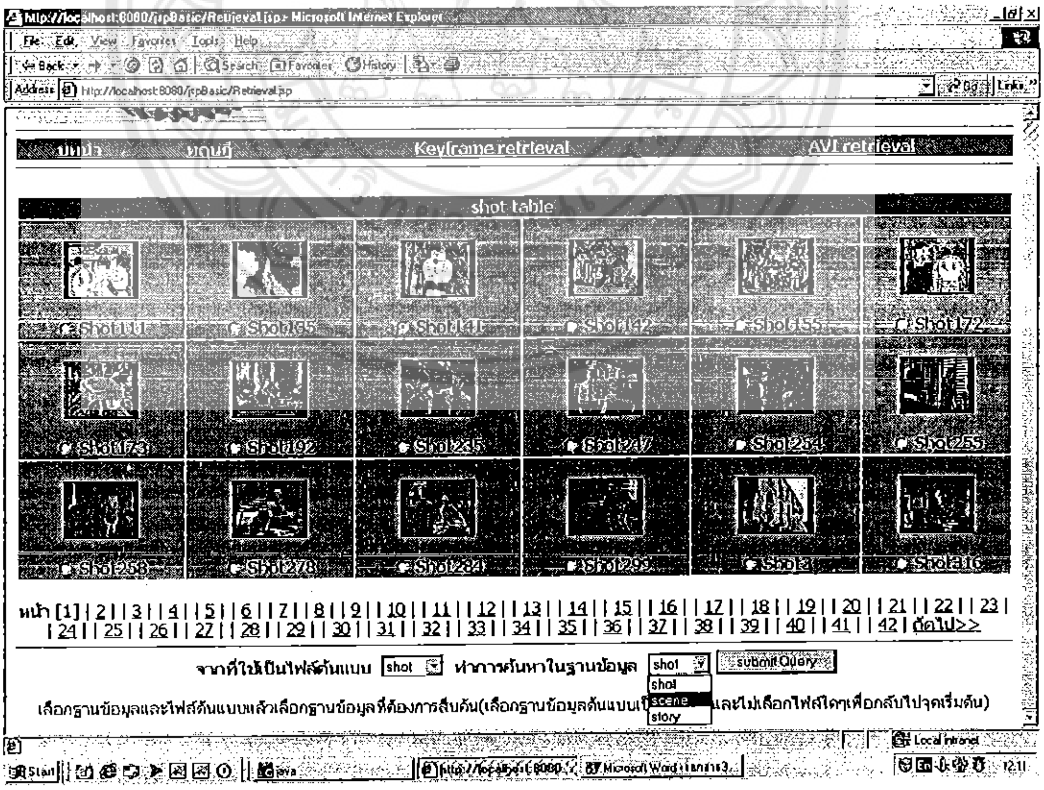
รูปที่ 4.3 แสดงตัวอย่างการสืบค้นไฟล์วีดีโอในระดับ Shot

4.3 การสืบค้นไฟล์วีดีโอในระดับ Scene และ Story

ในวีธีการที่เราใช้ในกาสืบค้นไฟล์วีดีโอนั้นเราจะสามารถสืบค้นในระดับ Scene และ Story ได้ โดยเราจะใช้การป้อนไฟล์ต้นแบบในการสืบค้นจำนวน 15 ไฟล์ สำหรับหาไฟล์วีดีโอเดียวกันในแต่ละระดับ ซึ่งเมื่อทำการสืบค้นมาได้แล้ว เราจะดูผลจากเนื้อหาในไฟล์วีดีโอว่าตรงกับความต้องการหรือไม่ทีละไฟล์วีดีโอ (จะทำการทดสอบที่ 10 ไฟล์วีดีโอ ต่อ 1 การสืบค้น) จะได้กราฟตามรูปที่ 4.4 และรูปที่ 4.5 เป็นรูปตัวอย่างจากโปรแกรมในการสืบค้นข้ามลำดับขั้น



รูปที่ 4.4 แสดงกราฟผลทดลองระดับ Scene และ Story



รูปที่ 4.5 แสดงตัวอย่างการค้นหาไฟล์จากระดับ Shot ใน Scene

บทที่ 5

สรุปผลงานวิจัยและการนำงานวิจัยไปใช้

เมื่อได้ทำการวิจัยเสร็จสิ้นลงแล้ว เราจะพบว่ากระบวนการในการสืบค้นวีดีโอจากฐานข้อมูลนั้นจะต้องมีการสร้างเวคเตอร์ที่ใช้ในการอ้างอิงถึงวีดีโอให้มีขนาดเหมาะสมกับฐานข้อมูล กล่าวคือ ถ้าเวคเตอร์มีขนาดใหญ่ไปก็จะสิ้นเปลืองที่เก็บ ถ้าเล็กไปก็จะทำให้ ทำการสืบค้นได้ไม่ดีเท่าที่ควร

5.1 การหาค่า Histogram

ในการหาค่า Histogram ในระบบสี HSV นั้นเมื่อเรากำหนดให้จำนวนช่วงมากขึ้นก็จะทำให้ค่าในความถูกต้องในการหาภาพที่มีลักษณะเหมือนกันมีค่าถูกต้องมากขึ้นแต่ในขณะเดียวกันขนาดของเวคเตอร์ที่ใช้อธิบายภาพแต่ละภาพก็จะมีขนาดใหญ่ขึ้นและจะพบช่วงของ Histogarm ที่เป็นศูนย์มากขึ้นไปด้วยเมื่อเพิ่มค่าของจำนวนช่วงสีในแกน V เราจะพบว่าค่าของช่วงที่เพิ่มขึ้นจะให้ผลการสืบค้นที่ไม่แตกต่างกันมากนัก แต่กลับทำให้จำนวนของช่วง Histogram ที่เป็นศูนย์มีค่ามากขึ้น

ดังนั้นในการทำ Histogram นั้นระนาบที่อยู่ข้างล่างระบบสี HSV จะมีสีที่เป็นสีดำและค่าของสีจะมีค่าใกล้เคียงกันมากจนการแบ่งสีนั้นจะไปตกอยู่ในช่วงสี H แรกเป็นส่วนมากดังนั้นเราอาจทำ Histogram ให้ละเอียดโดยให้จำนวนของช่วงสี H และ S ในแต่ละระนาบตามแนวแกน V มีค่าไม่เท่ากัน หรือทำการให้ค่าน้ำหนักของแต่ละช่วงของ Histogram ไม่เท่ากันก็ได้

5.2 การสืบค้นไฟล์วีดีโอ

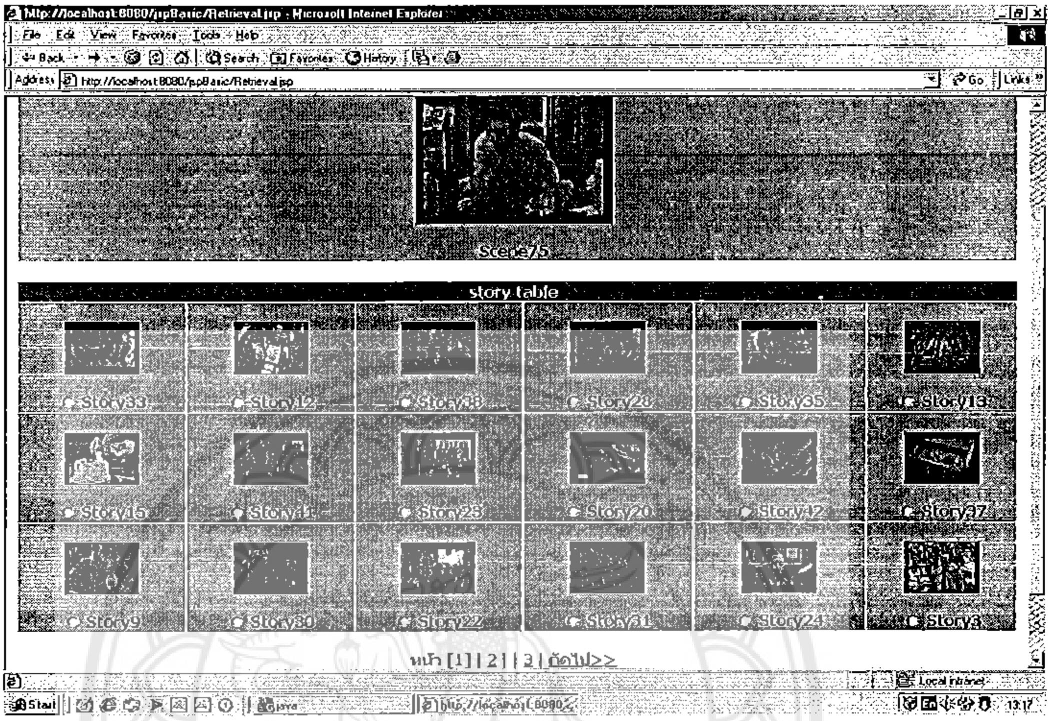
5.2.1 การสืบค้นไฟล์วีดีโอในระดับ Shot

จากผลการทดลองเราสามารถสรุปได้ว่าการใช้การสืบค้นไฟล์วีดีโอโดยใช้เฟรมตัวแทน (KFVI) กับ การสืบค้นโดยใช้วิธี Adaptive Video Indexing (AVI) นั้นจะให้ผลที่ไม่ต่างกันมากนัก ดังนั้นเราจึงสามารถนำวิธีนี้มาอธิบายไฟล์วีดีโอในระดับ Shot แทนวิธีการใช้เฟรมตัวแทนได้โดยที่การหาจะให้ผลที่ไม่ต่างกัน แต่วิธี AVI ที่สามารถจะอธิบายถึงเนื้อหาของไฟล์วีดีโอได้อย่างครบถ้วนนั้นจะมีค่าของตามหน้าเชื่อถือมากกว่าการใช้เฟรมตัวแทน โดยค่าที่ได้จากการสืบค้นจะมีค่ามากกว่า 50 เปอร์เซนต์ในกรณีที่ต้องการจำนวนไฟล์ 16ไฟล์

5.2.2 การสืบค้นไฟล์วีดีโอระดับ Scene และ Story

จากการทดลองจะรวมไปถึงการสืบค้นข้ามลำดับชั้นของไฟล์วีดีโอด้วย ซึ่งก็คือการสืบค้นใน Shot-to-Scene, Scene-to-Scene, Scene-to-Story และ Shot-to-Story พบว่าการสืบค้นแบบ AVI นี้

สามารถนำไปใช้ในการสืบค้นไฟล์วีดีโอได้ในทุกๆ ระดับ เนื่องจากผลที่ออกมานั้น มีค่าความผิดพลาดที่อยู่ในขอบข่ายที่ยอมรับได้ คือ



รูปที่ 5.1 แสดงตัวอย่างผลการค้นหาไฟล์วีดีโอ scene to story

Shot to Scene จำนวนความถูกต้องเริ่มลดลงที่จำนวน 5 ไฟล์วีดีโอ และต่ำสุดอยู่ที่ประมาณ 50 เปอร์เซ็นต์ จากที่เห็นการสืบค้นในระดับนี้ ก็มีข้อผิดพลาดพอสมควรแต่ก็ไม่มากจนเกินไป

Scene to Scene จำนวนความถูกต้องเริ่มลดลงที่จำนวน 6 ไฟล์วีดีโอ และต่ำสุดอยู่ที่ประมาณ 55 เปอร์เซ็นต์ การสืบค้นในระดับนี้ ความผิดพลาดมีไม่ค่อยมากนัก อาจเนื่องมาจากเป็นการสืบค้นที่อยู่ในระดับเดียวกัน

Scene to Story จำนวนความถูกต้องเริ่มลดลงที่จำนวน 7 ไฟล์วีดีโอ และต่ำสุดอยู่ที่ประมาณ 60 เปอร์เซ็นต์ ความผิดพลาดค่อนข้างน้อย แสดงให้เห็นว่าเป็นระดับการสืบค้นที่น่าเชื่อถือเลยทีเดียว ดังรูปที่ 5.1

Shot to Story จำนวนความถูกต้องเริ่มลดลงที่จำนวน 5 ไฟล์วีดีโอ และต่ำสุดอยู่ที่ประมาณ 55 เปอร์เซ็นต์ เป็นการสืบค้นข้ามลำดับชั้นที่ลำดับชั้นห่างกันมากที่สุดแต่ก็ไม่เป็นปัญหา กับการสืบค้นด้วยวิธี AVI

โดยภาพรวมแล้ว ไม่ว่าจะเป็นการสืบค้นในระดับใดก็ตาม วิธี AVI นี้ก็มีประสิทธิภาพพอ ๆ กัน ทำให้เห็นว่าไม่ว่าจะเป็นการสืบค้นในระดับใดก็ไม่เป็นอุปสรรคต่อการสืบค้นด้วยวิธี AVI เลยก็ว่าได้ วิธีสืบค้นวิธีนี้จึงเป็นวิธีที่น่าสนใจต่อการนำไปใช้และค้นคว้าต่อไปอย่างมาก

5.3 คำแนะนำในการจำกระบวนการสืบค้นไปใช้

การสืบค้นไฟล์วีดีโดยใช้วิธี AVI นี้จะสามรถใช้กับไฟล์วีดีโอได้ในทุกระดับของไฟล์วีดีโอ เพราะใช้ทุกเฟรมในไฟล์วีดีโอในการอธิบายถึงไฟล์วีดีโอนั้นๆ ดังนั้นค่าของจำนวนแม่แบบหรือ template ที่จะใช้ในการเป็นต้นแบบในการสืบค้นในควรจะมีค่ามากเพราะเมื่อเราเพิ่มไฟล์วีดีโอเข้ามาใหม่ได้โดยไม่ต้องทำการหา template ใหม่ ในงานวิจัยนี้ได้เลือกฐานข้อมูล คือ MySQL ซึ่งมีข้อได้เปรียบเป็นซอฟ์ทแวร์ที่สามารถดาวน์โหลดมาใช้ได้ฟรี และใช้โปรแกรม Matlab ในการสร้างเวคเตอร์ AVI ดังนั้นผู้ที่จะใช้งานโปรแกรมที่ได้มาจากงานวิจัยจึงควรจะมีความรู้ในด้าน MySQL และ Matlab ด้วยเพื่อที่จะสามารถสร้างฐานข้อมูลได้ถูกต้อง

กระบวนการสืบค้นที่ใช้ในงานวิจัยนี้เป็นกระบวนการเบื้องต้นในการสืบค้นไฟล์ ในฐานข้อมูลคือหลังจากที่ผู้ใช้ทำการป้อนเวคเตอร์ของไฟล์ต้นแบบที่ใช้สำหรับหาเข้าไปแล้วคอมพิวเตอร์ก็จะคำนวนหาว่าไฟล์ใดมีเวคเตอร์ที่ใกล้เคียงกันกับเวคเตอร์ที่เข้ามาแล้วแสดงผลลัพธ์ออกมาเท่านั้น ในกระบวนการดังกล่าวจะเห็นว่าเครื่องจะทำการแสดงผลลัพธ์ออกมาโดยไม่สามารถทราบได้เลยว่าผลลัพธ์ที่แสดงออกมานั้นถูกต้องตามความต้องการของผู้ใช้หรือไม่ ดังนั้นเราข้อแนะนำที่จะให้ผู้ใช้ไปพัฒนาต่อคือการใช้ การโต้ตอบระหว่างผู้ใช้กับคอมพิวเตอร์ โดยอาจใช้กระบวนการ Feedback และ Neural network เข้ามาช่วยในการสืบค้นให้ได้ค่าของไฟล์ที่มีความถูกต้องเพิ่มขึ้น

เอกสารอ้างอิง

[1] กิตติ ภักดีวัฒนะกุล, อังศุมาลิน เวชนารายณ์ และ กิตติพงษ์ ธีรวัฒน์เสถียร. (2545). **PHP ฉบับโปรแกรมเมอร์.** กรุงเทพฯ: บริษัท เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด.

[2] ชมรม Computer & IT. (2546). **เขียนโปรแกรมเชิงวัตถุด้วย JAVA.** มหาวิทยาลัยนเรศวร: คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร.

[3] ณัฐดนัย สุขรัตน์. (2546). **คัมภีร์ JAVA เล่ม 2.** กรุงเทพฯ: บริษัท เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด.

[4] รศ.ดร.มนัส สังวรศิลป์ และ วรรัตน์ ภัทรอมรกุล. (2543). **คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์.** นนทบุรี: สำนักพิมพ์ อินโฟเพรส.

[5] อุดม จีนประดับ และ ผศ.ดร.สมคิด เรืองธนะสกุลไทย. (1997). **ทฤษฎีและตัวอย่างโจทย์ โครงสร้างข้อมูล DATA STRUCTURES.** กรุงเทพฯ: แมคกรอ-ฮิล อินเตอร์เนชั่นแนล เอ็นเตอร์ไพร์ส, อิงค์.

[6] นรินทร์ โอฬารกิจอนันต์. (2545). **สร้างโฮมเพจระดับมืออาชีพ ด้วย JSP จาวา เซิร์ฟเวอร์ เพจ สำหรับเวบโปรแกรมเมอร์.** กรุงเทพฯ: สำนักพิมพ์ เดคิซูกิ ดอทเนต.

[7] ผศ.นุกูล กระจาย. (2541). **การเขียนโปรแกรมและประมวลผลข้อมูลด้วย เทอร์โบปาสคาล.** กรุงเทพฯ: บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน).

[8] สาธิต ชัยวิวัฒน์ตระกูล. (2545). **เก่ง JSP ให้ครบสูตร.** กรุงเทพฯ: บริษัท วิตตี้ กรุ๊ป จำกัด.

[9] พ.อ.เจนวิทย์ เหลืองอร่าม และ ปิยวิทย์ เหลืองอร่าม **"การพัฒนา Web Applications ด้วย JavaServer Pages และ Servlet, JavaBeans,XML"** Se-ed Ucation Lt.d , Bs2545 pp. 4-5

[10] Clay M. Thompson & Loren Shure. (1993). **Image Processing TOOLBOX For Use with MATLAB.** Mass: The MathWorks, Inc.

[11] Ricardo Baeza-Yates & Berthier Ribeiro-Neto. **Modern Information Retrieval.** Oxford: ACM European Service Center.

[12] Refael C. Gonzalez & Richard E. Woods. (2002). **Digital Image Processing.** New Jersey: Prentice-Hall, Inc.

[13] **http://www.datacompression.com**

[14] **http://www.ctr.columbia.edu/~jrsmith/html/pubs/acmmm96/node8.html**

[15] **http://www.mitre.org/support/papers/lvr/**

เอกสารอ้างอิง(ต่อ)

[16] Paisarn Muneesawang and Ling Guan **"Video Retrieval using an Adaptive Video Indexing Technique and Automatic Relevance Feefback"** IEEE Transaction on Neural Networks, Sept 2002.

มหาวิทยาลัยนเรศวร

ภาคผนวก ก

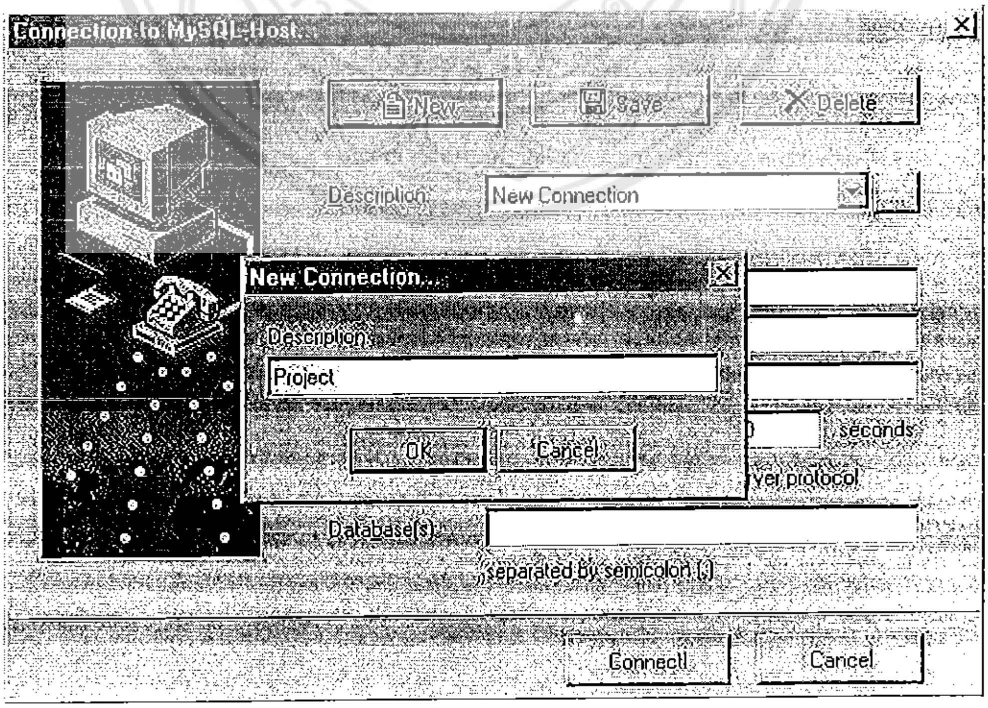
ทำงานกับ MySQL โดยใช้ MySQL-Front

โปรแกรม MySQL-Front เป็นเครื่องมือที่ช่วยอำนวยความสะดวกในการใช้งานระบบฐานข้อมูล MySQL โดยเฉพาะโปรแกรมนี้ทำงานในระบบ Windows หน้าตาการใช้งานจึงมีลักษณะเป็นกราฟิก ซึ่งจะช่วยอำนวยความสะดวกในการใช้งานสำหรับผู้ใช้มือใหม่ โดยที่ไม่จำเป็นต้องพิมพ์คำสั่งต่างๆ ทาง command line

โปรแกรมนี้ดาวน์โหลดมาใช้งานได้ฟรีจาก http://www.anse.de/mysqlfront/ เป็นไฟล์ของเวอร์ชัน 2.4 ชื่อว่า MySQL-Front_2.4_Setup.exe ซึ่งมีขนาดไม่ถึง 1 MB หลังจากได้มาแล้วก็ดับเบิลคลิกเพื่อติดตั้งใช้งานได้ทันทีเหมือนกับโปรแกรมทั่วไปที่รันในระบบ Windows

ล็อกอินเข้าสู่ฐานข้อมูล

พอติดตั้งโปรแกรม MySQL-Front เสร็จเรียบร้อย จะปรากฏไอคอน shortcut ของโปรแกรมที่หน้าจอ Desktop เมื่อดับเบิลคลิกที่ shortcut แล้ว โปรแกรม จะทำงานโดยแสดงหน้าต่างขึ้นมาดังรูปที่ ก-1 (เวลาจะใช้โปรแกรม MySQL-Front ต้องใช้คำสั่ง mysqld เพื่อรันโปรแกรม MySQL ให้อยู่ในสภาพพร้อมทำงานก่อนแล้ว มิฉะนั้นจะไม่สามารถติดต่อกับฐานข้อมูลได้)

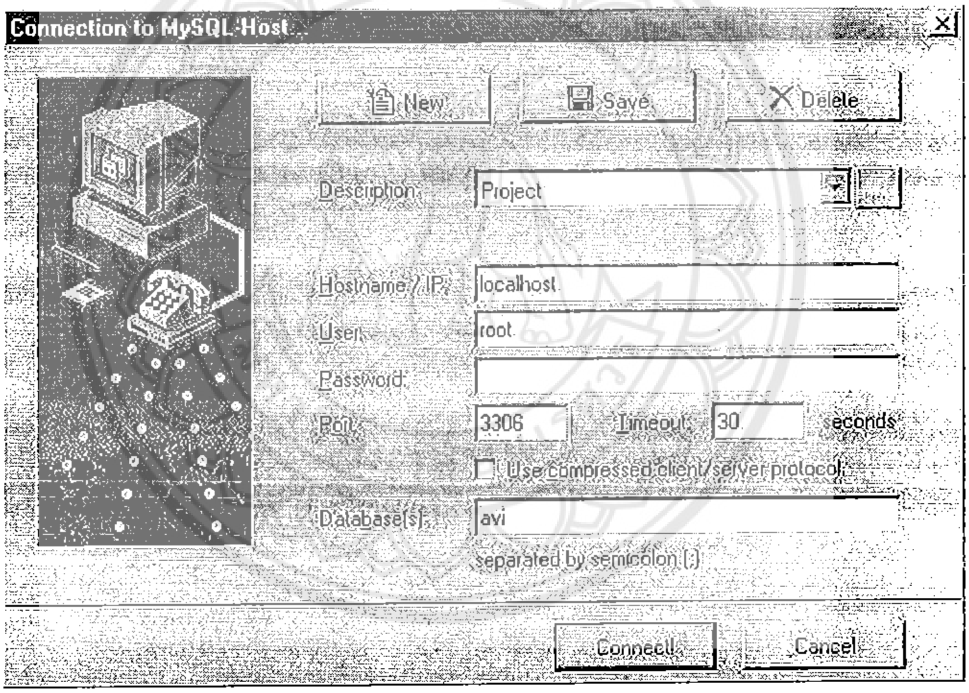


รูปที่ ก-1 ตั้งชื่อการเชื่อต่อฐานข้อมูล

คลิกที่ปุ่ม New เพื่อตั้งชื่อการเชื่อมต่อฐานข้อมูล โดยจะตั้งชื่อเป็นอะไรก็ได้ จากนั้น จะได้ผลลัพธ์ดังรูปที่ ก-2 ซึ่งต้องกำหนดค่าต่างๆ ที่จะใช้ติดต่อกับฐานข้อมูลลงไป ดังนี้

- **ช่อง Hostname / IP:** ให้ระบุชื่อเครื่องหรือหมายเลข IP address ของเครื่องที่ติดตั้ง MySQL
- **ช่อง User:** ให้ระบุชื่อผู้ใช้
- **ช่อง Password:** ให้ระบุรหัสผ่าน
- **ช่อง Databases(s):** ให้ระบุชื่อฐานข้อมูลที่ต้องการใช้งาน ถ้าไม่ระบุ จะหมายความว่าต้องการใช้งานทุกๆฐานข้อมูล

สำหรับ**ช่อง Port:** และ**ช่อง Timeout:** มีค่าดีฟอลต์กำหนดมาให้อยู่แล้วคือ หมายเลขพอร์ตที่ใช้ติดต่อกับ MySQL เป็น 3306 และค่า timeout เป็น 30 วินาที

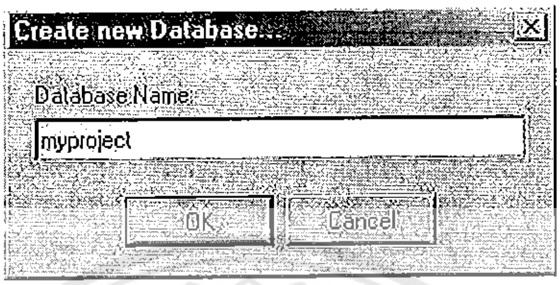


รูปที่ ก-2 ระบุรายละเอียดที่ใช้สำหรับการติดต่อฐานข้อมูล

เมื่อกรอกข้อมูลครบแล้ว คลิกปุ่ม Connect! จะได้ผลลัพธ์ดังรูปที่ ก-3 สังเกตทางฝั่งซ้ายของภาพ จะเห็นโครงสร้างของฐานข้อมูล ตาราง และฟิลด์หรือคอลัมน์ต่างๆ ซึ่งคล้ายกับโครงสร้างของไดเรกทอรีในระบบ Windows นั่นเอง และเมื่อคลิกที่ตารางทางฝั่งซ้าย รายละเอียดเกี่ยวกับฟิลด์ทุกฟิลด์ในตารางนี้ก็จะแสดงออกมาที่บริเวณพื้นที่ตรงกลางของหน้าต่างในรูปที่ ก-3

สร้างฐานข้อมูลใหม่

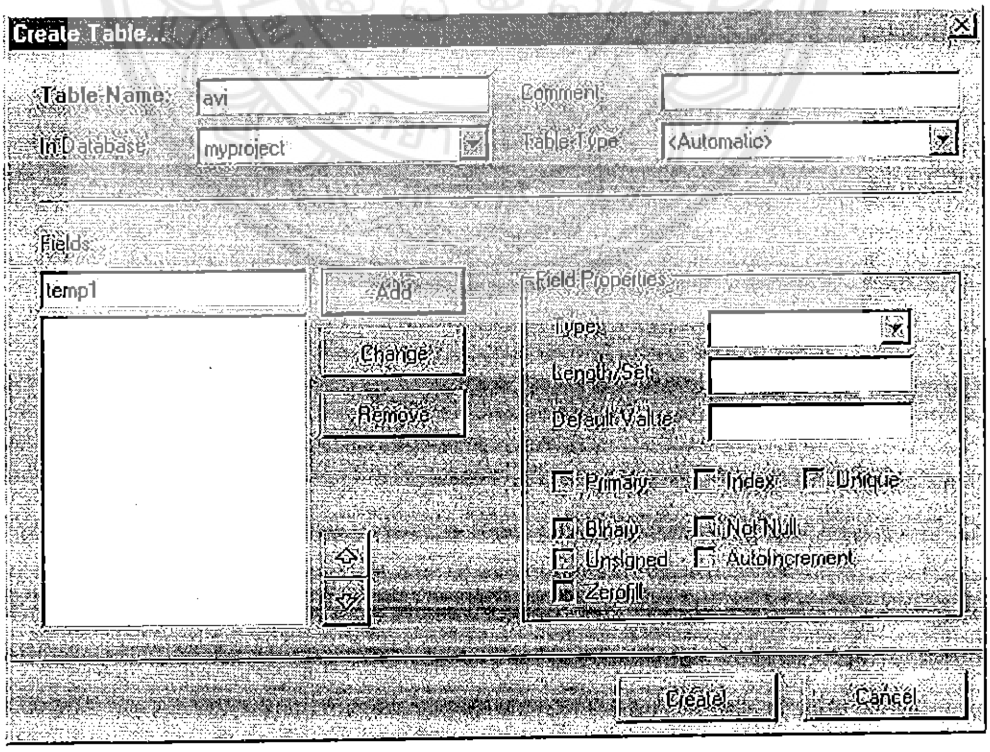
เลือกคำสั่ง Tools > Create Database หลังจากนั้นจะมีหน้าต่าง Create new Database... ปรากฏขึ้นมาดังรูปที่ ก-3 ให้พิมพ์ชื่อฐานข้อมูลที่ต้องการสร้างใหม่ จากนั้นจึงคลิก OK ก็ถือว่าสร้างฐานข้อมูลใหม่เรียบร้อยแล้ว



รูปที่ ก-3 พิมพ์ชื่อฐานข้อมูลที่ต้องการจะสร้างใหม่

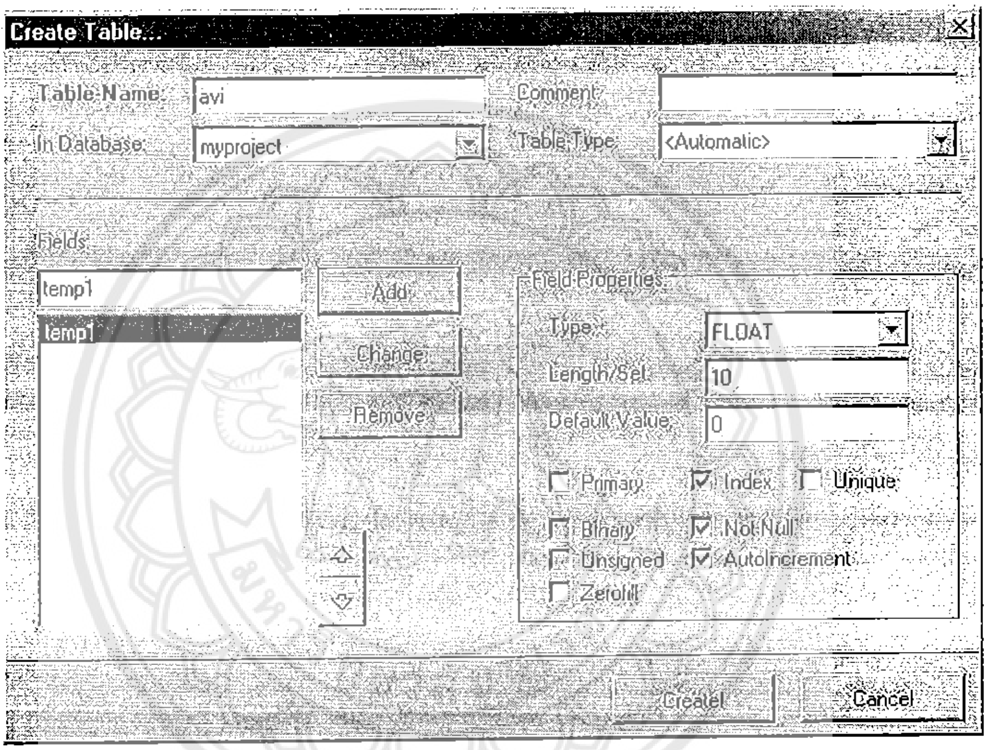
สร้างตารางใหม่

เลือกคำสั่ง Tools > Create Table หลังจากนั้นจะมีหน้าต่าง Create Table... ปรากฏขึ้นมาดังรูปที่ ก-4 เพื่อให้เรากรอกรายละเอียดเกี่ยวกับตารางที่ต้องการสร้าง ดังนี้



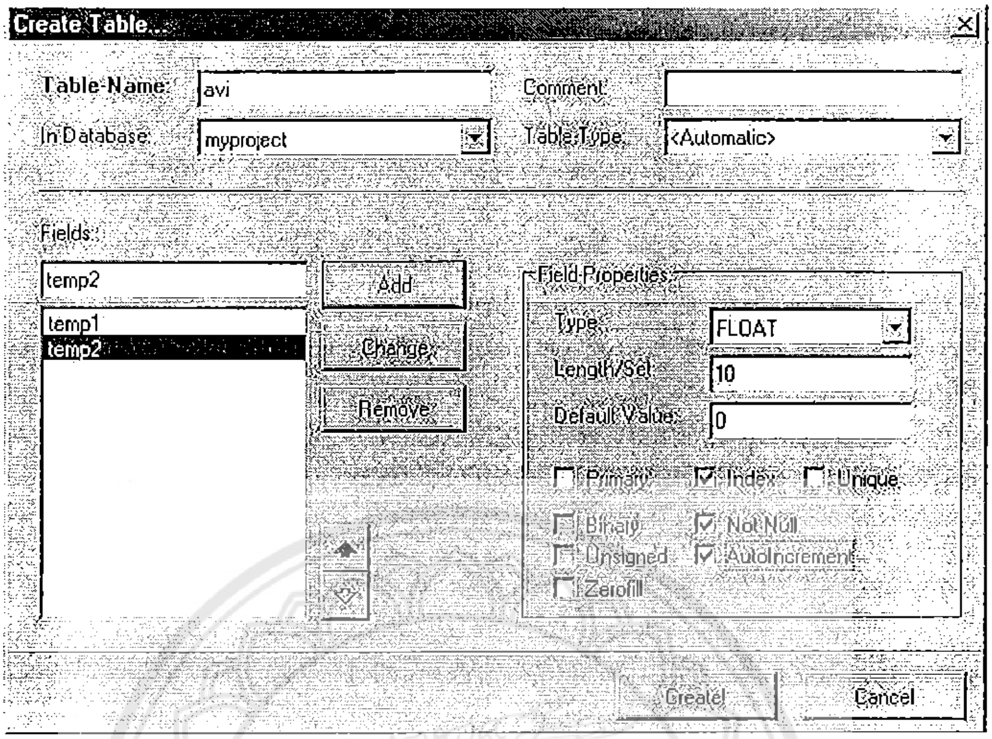
รูปที่ ก-4 กำหนดชื่อตารางและคอลัมน์แรกที่ต้องการ

- **ช่อง Table-Name** เอาไว้ให้ตั้งชื่อตาราง
- **ช่อง In Database** มีไว้ให้เลือกว่าจะสร้างตารางในฐานข้อมูลใด
- **ช่อง Fields** ให้กรอกชื่อฟิลด์แรกของตารางลงไป สมมติว่าชื่อ temp1 แล้วคลิกที่ปุ่ม Add ปรากฏว่า temp1 จะย้ายลงมาอยู่ในช่องข้างล่างดังรูปที่ ก-6 หลังจากนั้นเราก็สามารถกำหนดคุณสมบัติต่างๆ ของคอลัมน์นี้ได้ในหัวข้อ Field-Properties ทางด้านขวามือ

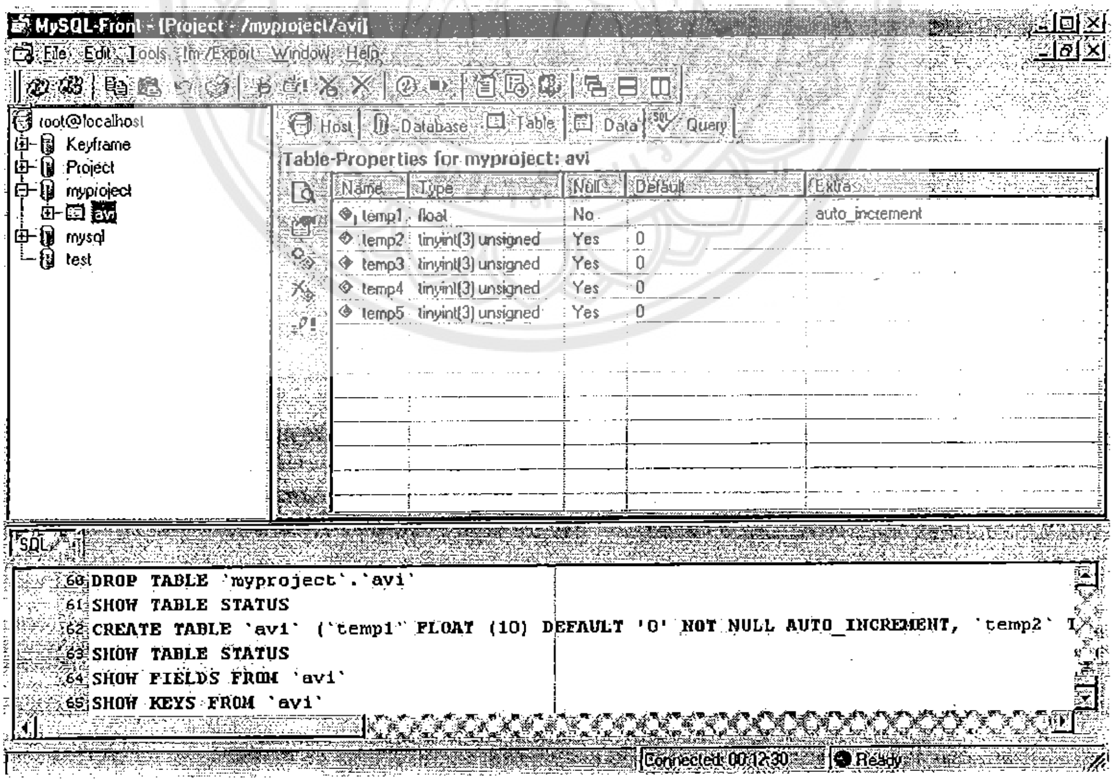


รูปที่ ก-5 กำหนดคุณสมบัติของฟิลด์ temp1

ขั้นต่อไป ให้ตั้งชื่อฟิลด์ temp2 ลงในช่อง Fields และคลิกปุ่ม Add แล้วจึงกำหนดคุณสมบัติต่างๆ ของฟิลด์นี้ในหัวข้อ Field-Properties เช่นเดียวกัน ดังรูปที่ ก-7



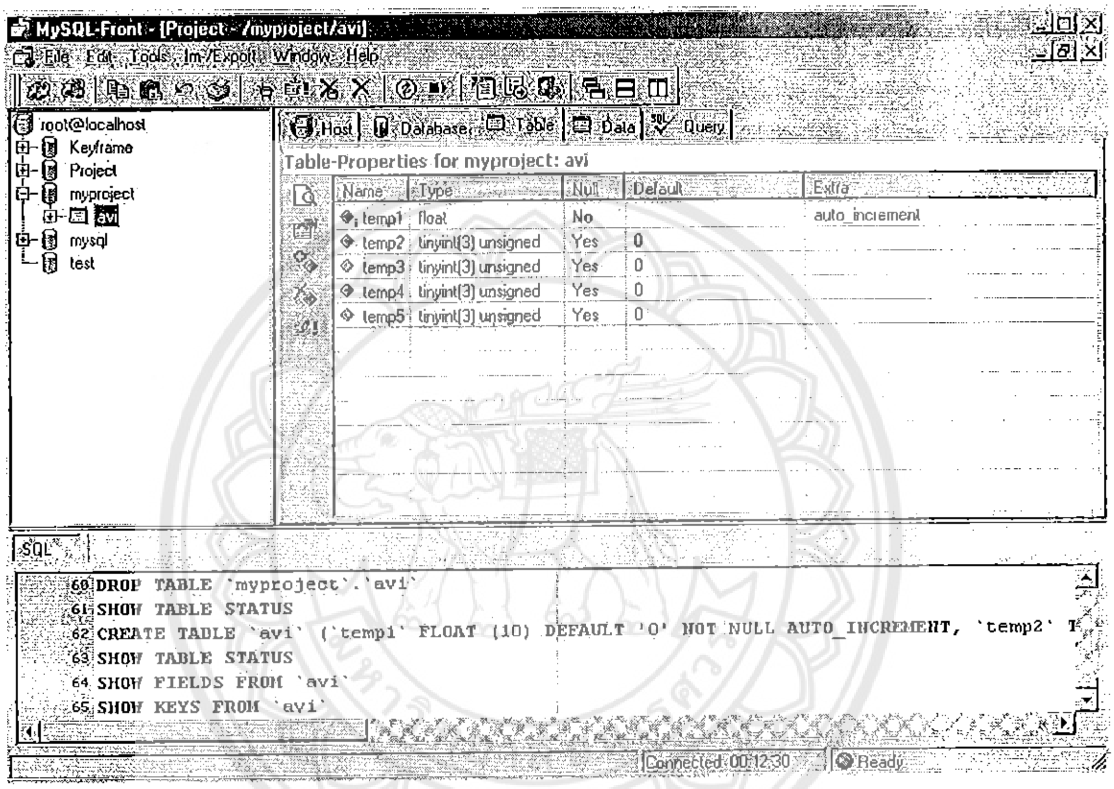
รูปที่ ก-6 กำหนดคุณสมบัติของฟิลด์ temp2



รูปที่ ก-7 ผลลัพธ์ที่ได้เมื่อสร้างตารางเรียบร้อยแล้ว

สำหรับฟิลด์อื่นๆ ที่เหลือ ก็ทำตามวิธีการเดียวกันนี้จนครบ จากนั้นให้คลิกที่ปุ่ม Create! จะได้ผลลัพธ์ดังรูปที่ ก-8 ซึ่งแสดงโครงสร้างของตาราง ที่สร้างเสร็จเรียบร้อยแล้ว

ในกรณีที่คุณต้องการลบตารางที่สร้างไว้ ก็สามารถทำได้โดยคลิกขวาที่ตารางที่ต้องการลบ แล้วเลือกคำสั่ง Drop Table... ในเมนูที่เปิดออกมาดังแสดงในรูปที่ ก-9 จากนั้นจะมีหน้าต่างให้ตอบยืนยันการลบอีกครั้ง ถ้าคุณคลิก OK ตารางนั้นก็จะถูกลบทันที



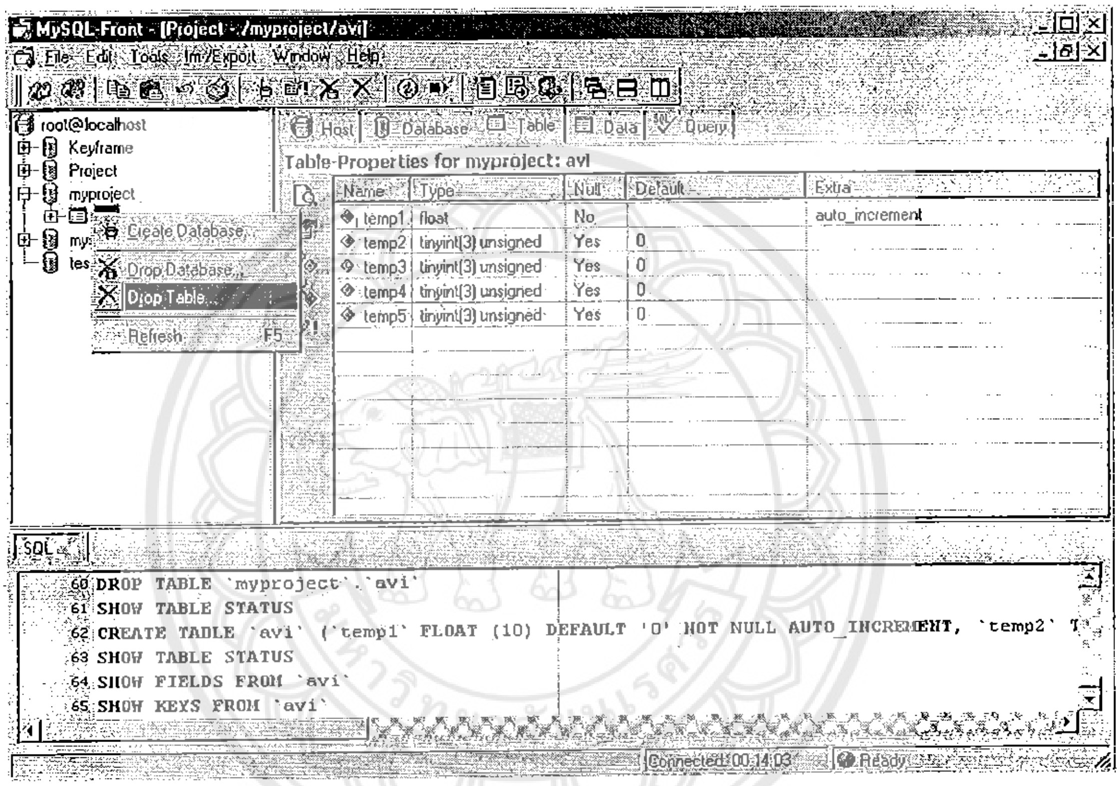
รูปที่ ก-8 ผลลัพธ์ที่ได้เมื่อสร้างตารางเรียบร้อยแล้ว

เพิ่ม-ลบ-แก้ไขเรคอร์ดในตาราง

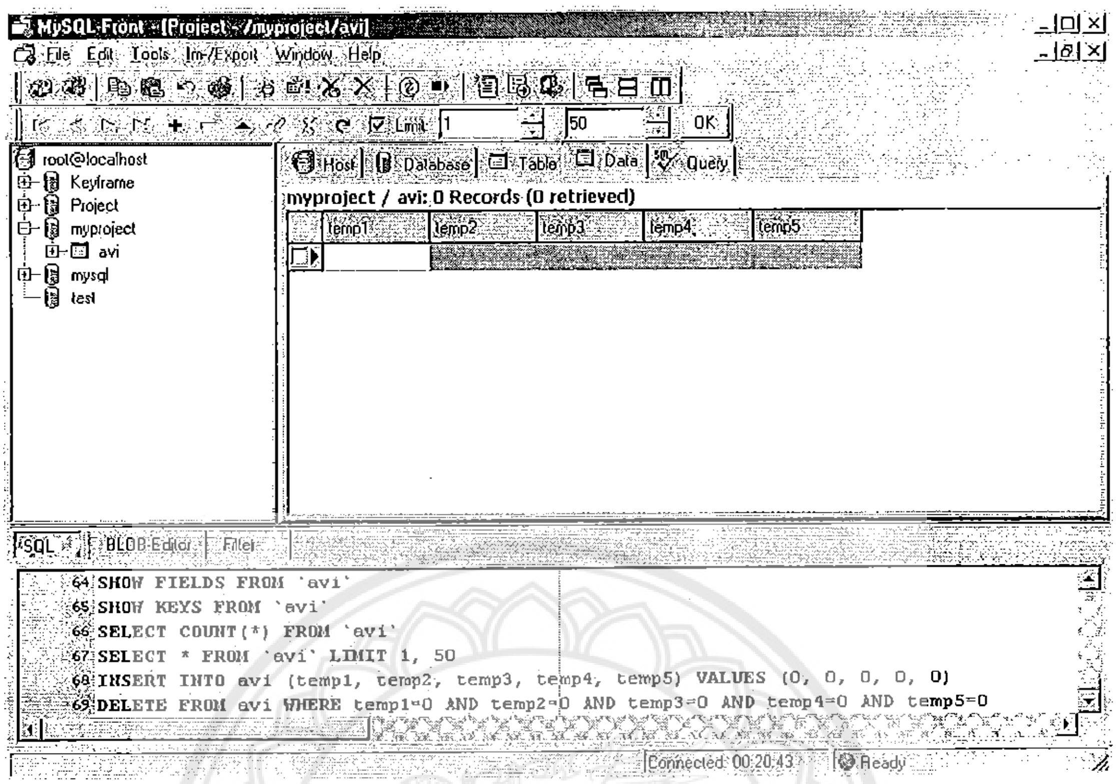
จากรูปที่ ก-9 เราสามารถเพิ่มเรคอร์ดใหม่ในตาราง โดยคลิกไปที่แท็บ Data ปกติแล้วภายในแท็บนี้จะแสดงเรคอร์ดของข้อมูลทั้งหมดที่มีอยู่ในตาราง แต่ถ้ายังไม่มีข้อมูลในตาราง ก็จะแสดงเป็นช่องว่างให้กรอกข้อมูลใหม่เข้าไปได้ ดังรูปที่ ก-10

เมื่อกรอกข้อมูลที่ต้องการครบแล้ว ให้สังเกตว่าแถบทูลบาร์ด้านบนจะปรากฏไอคอนไว้ให้คลิก โดย

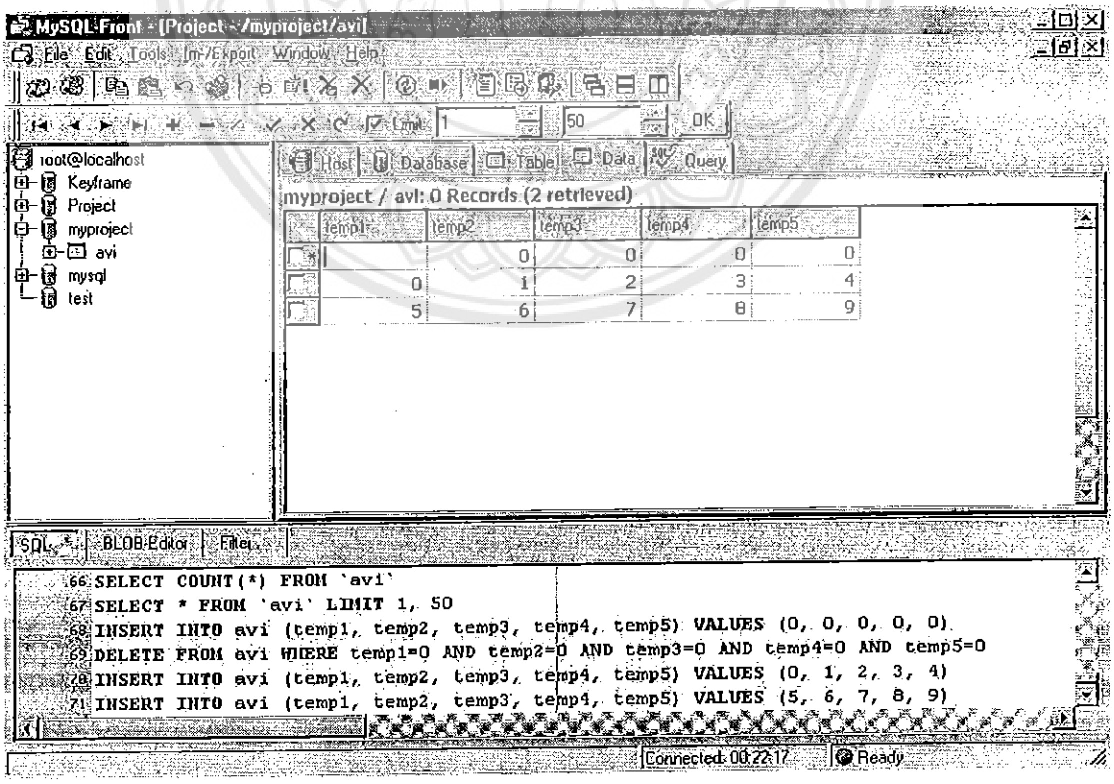
- ถ้าต้องการกรอกข้อมูลเรคอร์ดใหม่ ให้คลิกที่ปุ่ม เพื่อแทรกช่องให้กรอกข้อมูลใหม่ ดังตัวอย่างในรูปที่ ก-11
- คลิกที่ปุ่ม ถ้าต้องการยกเลิกข้อมูลที่แก้ไข
- ถ้าต้องการลบเรคอร์ดใด ให้คลิกที่เรคอร์ดนั้นแล้วคลิกที่ปุ่ม
- คลิกที่ปุ่ม ถ้าต้องการ save ข้อมูลที่แก้ไขหรือเพิ่มเข้าไปใหม่



รูปที่ ก-9 เมื่อต้องการลบตาราง



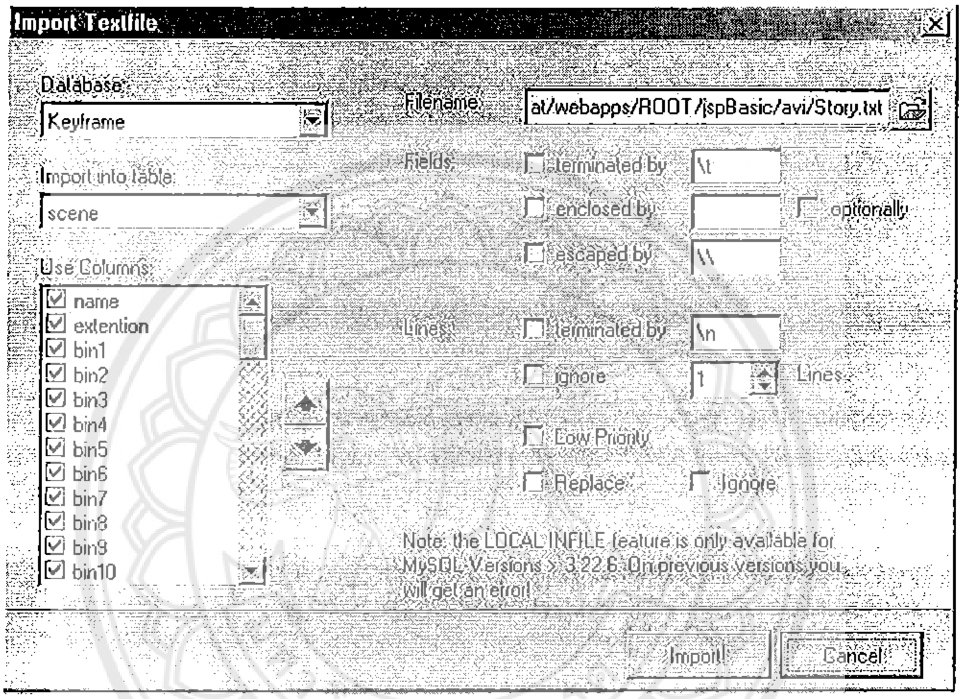
รูปที่ ก-10 คลิกไปที่แท็บ Data เพื่อเพิ่มเรคอร์ดใหม่



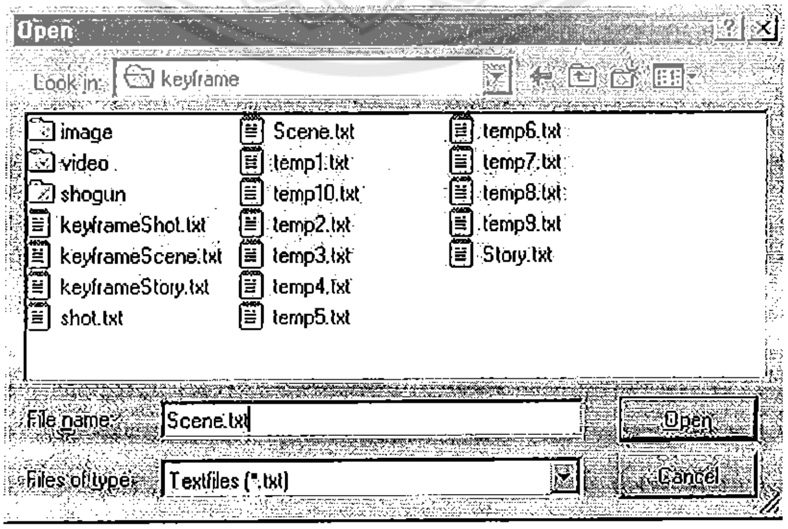
รูปที่ ก-11 เพิ่มเรคอร์ดใหม่ในตาราง

การอิมพอร์ทเทคไฟล์

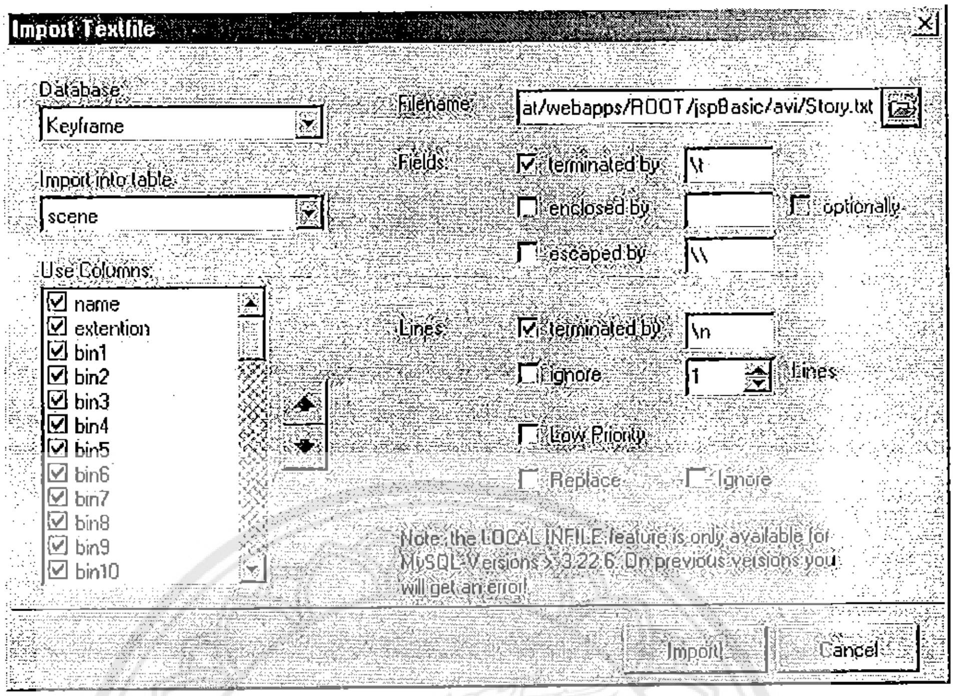
เลือกคำสั่ง Im-/Export > Import Textfile จะได้ตามรูปที่ ก-12 จากนั้นเลือกไฟล์ที่ต้องการอิมพอร์ท โดยพิมพ์ใส่ในช่อง Filename: หรืออาจ Browse โดยกดที่ปุ่ม ดังรูปที่ ก-13 แล้วเลือก Database และ Table ที่ต้องการจะอิมพอร์ทเทคไฟล์ไปไว้ และเลือกใช้ /t ที่ Fields แล้ว /n ที่ Lines จะได้ออกมาดังรูปที่ ก-14 แล้วคลิกที่ปุ่ม Import!



รูปที่ ก-12 อิมพอร์ทเทคไฟล์



รูปที่ ก-13 เลือกเทคไฟล์เพื่อนำมาใส่ในฐานข้อมูล



รูปที่ ก-14 ใส่รายละเอียดสำหรับเทคไฟล์ที่จะใส่ฐานข้อมูล

เนื้อหาที่อธิบายในภาพผนวกนี้เป็นเพียงตัวอย่างการใช้งานเบื้องต้น จริงๆแล้ว MySQL-Front ยังมีความสามารถอื่นๆ อีกมากมายซึ่งคุณสามารถทดลองใช้งานได้ตามที่ต้องการ เช่น การแบ็คอัพข้อมูล, การกำหนดชื่อผู้ใช้และรหัสผ่านให้แก่ MySQL เป็นต้น

ภาคผนวก ข

ตัวอย่างและการใช้งานฟังก์ชันในงานวิจัย

การใช้งานฟังก์ชัน hsvhist.m

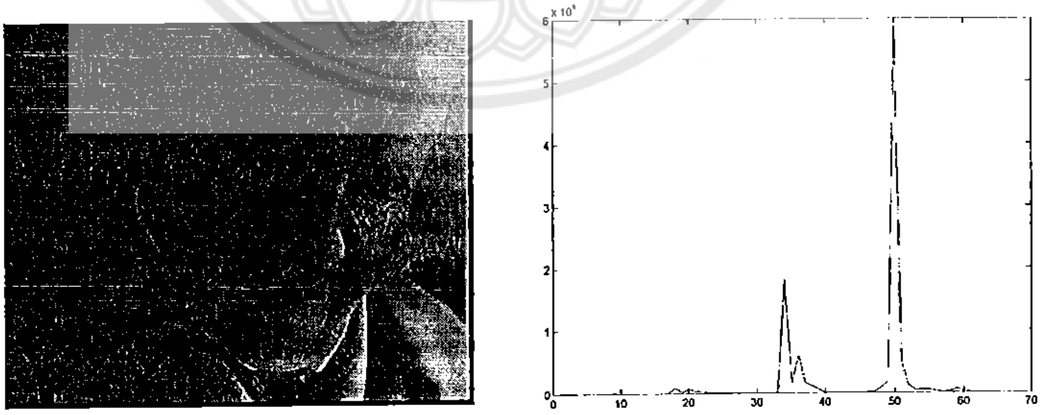
ฟังก์ชันที่ใช้ในการหา Histogram ของภาพที่เข้ามาโดยจะรับค่าเป็นภาพในระบบสี RFB จาก Matlab โดยจะทำการแปลงภาพให้อยู่ในระบบสี HSV ก่อนจากนั้นทำมาหาค่าของ Histogram มีรูปแบบการใช้งานดังนี้

[histogram]=hsvhist[Image,nH,nS,nV]

ค่าของ nH,nS และ nV จะเป็นค่าของจำนวนช่วงของสีที่ต้องการในแต่ละแนวแกนของระบบสี HSV โดยจะคืนค่าเวคเตอร์ไว้ใน histogram ขนาด nHxnSxnV ช่วงสี ภาพที่จะทำการใส่ไปในฟังก์ชันจะต้องอยู่ในรูปของเมทริกซ์ขนาด กว้างxยาวx3

ตัวอย่างการใช้งานฟังชัน hsvhist.m

```
>> I=imread('Video6.jpg');
>> histogram=hsvhist(I,16,4,1);
>> plot(histogram);
```



รูปที่ ข-1 ภาพที่ได้ที่ได้จากการหาค่า Histogram โดยใช้ฟังก์ชัน hsvhist

ตารางที่ ข-1 แสดงค่าของจำนวนเม็ดสีที่ตกลงในแต่ช่วงของ Histogram

1-10	39	3	2	0	4	0	3	5	1	0
11-20	0	0	0	0	0	0	21	839	134	671
21-30	305	211	42	4	6	8	15	5	10	3
31-40	4	10	58	18150	1607	5592	1634	1077	542	71
41-50	1	24	34	25	23	2	18	683	1663	58678
51-60	4893	1199	377	474	453	166	31	122	513	304
61-64	71	34	56	56						

การใช้งานฟังก์ชัน calculation.m

ฟังก์ชัน calculation.m เป็นฟังก์ชัน Matlab ที่ใช้ในการนำเฟรมในไฟล์วีดีโอทุกๆเฟรมมาหาค่า Histogram โดยเมื่อทำการเรียกใช้งานฟังก์ชันนี้ใน Matlab จะปรากฏข้อความขึ้นมาว่า "Please enter directory (etc. 'C:\\mathlab\\' or 'C:\\mathlab\\shot1.avi') >>" ให้เราทำการใส่ชื่อไฟล์วีดีโอที่ต้องการหา Histogram หรือโฟลเดอร์ที่เก็บไฟล์ที่เราต้องการจะหาค่าของ Histogram ซึ่งในการป้อนชื่อไฟล์หรือชื่อโฟลเดอร์เข้าไปนั้นจะต้องมีเครื่องหมาย ' ' คร่อมอยู่เสมอดังตัวอย่างต่อไปนี้

```
>> calculation
Please enter directory (etc. 'C:\mathlab\' or 'C:\mathlab\shot1.avi') >>  'c:\video\'
```

ฟังก์ชันจะทำการค้นหาไฟล์วีดีโอที่อยู่ในโฟลเดอร์นั้นทั้งหมด แล้วทำการหาค่า Histogram ของทุกเฟรมในทุกไฟล์วีดีโอในโฟลเดอร์นั้น แต่ถ้าเราต้องการหาค่า Histogram ของไฟล์วีดีโอเพียงไฟล์เดียวเราสามารถทำได้โดยใส่ชื่อไฟล์นั้นลงไปได้ทันที เช่น 'c:\video\shot1.avi

ประเภทของไฟล์วีดีโอที่จะสามารถนำมาหาค่าของ Histogram ได้นั้นจะต้องเป็นไฟล์วีดีโอที่มีนามสกุลเป็น '.avi' หรือ '.m1v' เท่านั้น ค่าของ Histogram ที่ได้จะถูกเก็บไว้ในเท็กซ์ไฟล์ที่มีชื่อเดียวกันกับไฟล์วีดีโอ เช่น ถ้าเราหาค่า Histogram ของไฟล์วีดีโอที่ชื่อว่า shot1.avi ก็จะได้ค่าของ Histogram ใน ไฟล์ที่มีชื่อว่า shot1.txt

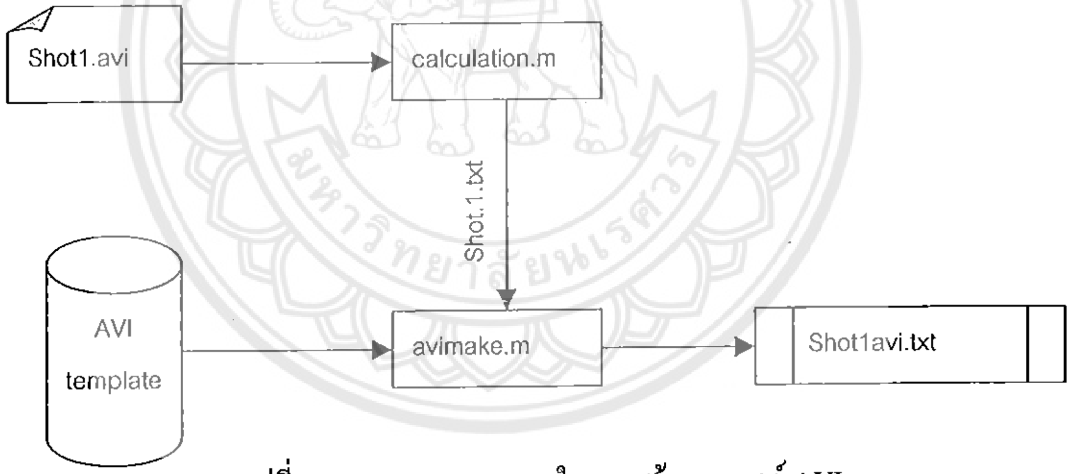
การประเมิณหาค่าเวคเตอร์ตัวแทนโดยใช้ฟังก์ชัน VQ

การหาค่าตัวแทนของกลุ่มเวคเตอร์ที่มีอยู่โดยการหาอัลกอริทึม LBG ในการหาในฟังก์ชันนี้จะต้องเก็บค่าของเวคเตอร์ทั้งหมดในไฟล์ data.txt เมื่อทำการเรียกใช้งานฟังก์ชันจะทำการโหลดกลุ่มของเวคเตอร์ที่ต้องการหาเวคเตอร์ตัวแทนจากไฟล์ดังกล่าว และจะให้เราทำการเลือกว่า

ต้องการได้เวคเตอร์ที่เป็นเวคเตอร์ตัวแทนทั้งหมดเท่าไรจากนั้นก็จะทำการประเมิณค่าของเวคเตอร์ที่ได้ออกในได้ตัวแปร codebk บน Matlab

การสร้างเวคเตอร์ AVI โดยใช้ฟังก์ชัน avimake.m

สิ่งที่ต้องมีก่อนที่จะทำการสร้างเวคเตอร์ของไฟล์วีดีโอได้คือ ค่าของ Histogram ของทุก ๆ เฟรมในไฟล์วีดีโอนั้นๆซึ่งสามารถใช้ฟังก์ชัน calculation.m ในการหาได้ และแม่แบบหรือ template ที่ใช้สำหรับเปรียบเทียบกับค่า Histogram ของเฟรมที่เข้ามา โดยเราสามารถสร้างแม่แบบได้จากการนำไฟล์วีดีโอทั้งหมดมาหาเวคเตอร์แล้วใช้ฟังก์ชัน VQ.m ในการหาตัวแทนของเวคเตอร์ที่เหมาะสม จากนั้นก็ทำการสร้างเวคเตอร์ AVI บน Matlab เมื่อสั่งให้ฟังก์ชันทำงานจะถามหาว่าไฟล์แม่แบบที่ใช้ในการสร้างเวคเตอร์อยู่ที่ไหน จากนั้นจะโหลดเข้ามาเก็บไว้ แล้วก็จะถามถึงไฟล์ที่ต้องการจะสร้างเวคเตอร์ ซึ่งจะต้องถูกเก็บเอาไว้ในเท็กซ์ไฟล์ เมื่อผ่านการทำเวคเตอร์ AVI เสร็จแล้วเราจะได้ไฟล์ที่มีชื่อเป็นชื่อไฟล์ดังกล่าวแต่จะมีคำว่า avi ต่อท้ายตัวอย่างเช่น shot1.txt จะได้ไฟล์ shot1avi.txt เป็นต้น



รูปที่ ข-2 แสดงกระบวนการในการสร้างเวคเตอร์ AVI

ประวัติผู้เขียนโครงงาน

ชื่อ นายกิจจา จีรพรชัย

เกิดวันที่ 18 กันยายน 2524

ภูมิลำเนา 77 หมู่ 10 ต.สาริกา อ.เมือง จ.นครนายก

ประวัติการศึกษา

- จบระดับประถมศึกษาจากโรงเรียนอนุบาลนครนายก
- จบระดับมัธยมศึกษาจากโรงเรียนนครนายกวิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

e-Mail : kitja_j@hotmail.com

ชื่อ นายอรรถสิทธิ์ ทรัพย์เมฆ

เกิดวันที่ 16 มิถุนายน 2524

ภูมิลำเนา 2 ซอย 13 ต.ตะพานหิน อ.ตะพานหิน จ.พิจิตร

ประวัติการศึกษา

- จบระดับประถมศึกษาจากโรงเรียนเทศบาลตะพานหิน
- จบระดับมัธยมศึกษาจากโรงเรียนตะพานหิน
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

e-Mail : lufecad@hotmail.com