BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

ISO 9001:2008

PHÚ THỊ QUYÊN

LUẬN VĂN THẠC SĨ
NGÀNH HỆ THỐNG THÔNG TIN

HẢI PHÒNG, 2016

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

PHÚ THỊ QUYÊN

XÂY DỰNG HỆ THỐNG TÌM KIẾM ÂM THANH THEO NỘI DUNG DỰA TRÊN ĐẶC TRƯNG MIỀN TẦN SỐ

LUẬN VĂN THẠC SĨ
NGÀNH CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH: HỆ THỐNG THÔNG TIN
MÃ SỐ: 60 48 01 04

NGƯỜI HƯỚNG DẪN KHOA HỌC:
PGS.TS. ĐẶNG VĂN ĐỨC

MỤC LỤC

Trang

DANH MỤC CÁC CHỮ VIẾT TẮT

Từ viết tắt	Từ đầy đủ
QoS (Quality of service)	Chất lượng dịch vụ
IR(Information Retrival)	Truy tìm thông tin
dB(Decibend)	Âm lượng
STFT(Short Time Fourier Transform)	
IDFT	
MFCC(Mel Frequency cepstral coefficients)	
LPC(Linear Predictive coding)	

DANH MỤC CÁC BẢNG BIỂU

MỞ ĐẦU

Xã hội ngày càng phát triển lượng thông tin lưu trữ ngày càng lớn dẫn tới việc tìm kiếm dữ liệu đa phương tiện càng trở nên khó khăn. Do đó cần có các hệ thống tìm kiếm thông tin hỗ trợ người sử dụng tìm kiếm một cách chính xác, nhanh chóng, hiệu quả và tiết kiệm thời gian.

Hơn nữa, Công nghệ thông tin truyền thông, mạng máy tính và các giao thức truyền thông phát triển mạnh mẽ, kết hợp với khả năng mô tả, đồ họa phong phú của các trình duyệt đã mang lại sự đa dạng về các dữ liệu cho người dùng đầu cuối.

Do đó, đòi hỏi làm thế nào để tổ chức và cơ cấu một lượng rất lớn các dữ liệu đa phương tiện để có thể dễ dàng nhận được thông tin cần thiết một cách nhanh chóng tại bất kỳ thời điểm nào.

Từ đó, cơ sở dữ liệu đa phương tiện được xây dựng để trở thành một công cụ quản lí, lưu trữ và truy cập một lượng lớn các đối tượng đa phương tiện. Đó chính là cơ hội cũng như là nguyên nhân để các công nghệ về cơ sở dữ liệu đa phương tiện phát triển và ứng dụng rộng rãi trong đời sống kinh tế xã hội.

Các dữ liệu đa phương tiện gồm có: văn bản, hình ảnh tĩnh, hình ảnh động, âm thanh, âm nhạc, video… Hiệu quả của các ứng dụng đa phương tiện phụ thuộc vào sức mạnh của cơ sở dữ liệu đa phương tiện, cụ thể là cấu trúc, cách tổ chức, khả năng truy cập nhanh, chính xác… Công nghệ đa phương tiện được ứng dụng trong nhiều trường hợp như: elearning, hội thảo video, thư điện tử, hiện thực ảo, trò chơi điện tử… Việc tìm hiểu bản chất cũng như là các đặc trưng, các thuộc tính, các kỹ thuật số hoá của từng loại dữ liệu đa phương tiện là yêu cầu để triển khai và ứng dụng công nghệ đa phương tiện vào đời sống.

Trong đó, việc tìm hiểu các đặc trưng, phương pháp số hoá, phương pháp trích chọn, tìm kiếm của dữ liệu âm thanh trong cơ sở dữ liệu âm thanh hiện đang được quan tâm đặc biệt bởi các đặc thù của dữ liệu âm thanh như:

đa dạng thông dụng với người dùng, thân thiện với mọi đối tượng, truyền tải một lượng lớn thông tin trong khoảng thời gian ngắn, ứng dụng nhiều trong đời sống, đó chính là lí do tôi chọn đề tài ***"Xây dựng hệ thống tìm kiếm âm thanh theo nội dung dựa trên các đặc trưng miền tần số"***

8. Đối tượng và phạm vi nghiên cứu

- Các khái niệm cơ bản về cơ sở dữ liệu đa phương tiện.

- Các khái niệm cơ bản về đặc trưng âm thanh.

- Một số kỹ thuật ứng dụng phát triển cơ sở dữ liệu âm thanh.

9. Hướng nghiên cứu của đề tài

- Nghiên cứu giải thuật liên quan đến các kỹ thuật tìm kiếm âm thanh trong cơ sở dữ liệu âm thanh.

- Nghiên cứu giải pháp công nghệ cài đặt chương trình thử nghiệm.

10. Những nội dung nghiên cứu chính

Nội dung nghiên cứu của luận văn bao gồm:

- Giới thiệu về cơ sở dữ liệu đa phương tiện

- Các đặc trưng âm thanh và cơ sở dữ liệu âm thanh

- Xây dựng chương trình thử nghiệm hệ thống tìm kiếm âm thanh.

11. Phương pháp nghiên cứu

Tổng hợp các tài liệu đã được công bố về dữ liệu âm thanh.

Thực nghiệm một số thuật toán biến đổi trong xử lý âm thanh

Nhận xét, đánh giá kết quả thử nghiệm.

12. Ý nghĩa khoa học và thực tiễn

- Luận văn nghiên cứu kỹ thuật tìm kiếm âm thanh theo nội dung.

- Cài đặt thử nghiệm các kỹ thuật xử lí âm thanh.

- Giải quyết bài toán xử lí âm thanh trong cơ sở dữ liệu âm thanh đa phương tiện.

13. Bố cục của luận văn

Luận văn bao gồm 3 chương cùng với phần Mở đầu, phần Kết luận, phần Mục lục, phần Tài liệu tham khảo.

CHƯƠNG 1: TỔNG QUAN CƠ SỞ DỮ LIỆU ÂM THANH

Trình bày một số khái niệm về CSDL đa phương tiện nói chung và CSDL âm thanh nói riêng. Các vấn đề cơ bản được trình bày bao gồm Kiến trúc tổng quan của hệ thống CSDL đa phương tiện, các loại dữ liệu đa phương tiện và mô hình của chúng. Các nhiệm vụ phát triển hệ thống CSDL đa phương tiện. Giới thiệu tình hình nghiên cứu trong và ngoài nước về vấn đề liên quan.

CHƯƠNG 2: TRÍCH CHỌN CÁC ĐẶC TRƯNG ÂM THANH

Trình bày tổng quan một số phương pháp, trích chọn đặc trưng âm thanh. Tiếp theo là nghiên cứu các thuộc tính và đặc trưng chính của âm thanh, bao gồm các đặc trưng trong miền thời gian biên độ, trong miền biến đổi và trong miền ảnh phổ. Các thuộc tính và đặc trưng chính của CSDL đa phương tiện, phân lớp âm thanh phục vụ tìm kiếm dữ liệu âm thanh trong CSDL âm thanh.

CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH THỬ NGHIỆM HỆ THỐNG TÌM KIẾM ÂM THANH

Giới thiệu bài toán thử nghiệm, dữ liệu thử nghiệm, các công cụ phần mềm hỗ trợ phát triển CSDL âm thanh. Thiết kế hệ thống, viết chương trình thử nghiệm. Dự định sử dụng MatLab để xây dựng chương trình demo.

CHƯƠNG 1: GIỚI THIỆU VỀ CƠ SỞ DỮ LIỆU ĐA PHƯƠNG TIỆN

1.1. CÁC DỮ LIỆU ĐA PHƯƠNG TIỆN

Đa phương tiện (multimedia) là một phương pháp giới thiệu thông tin trên máy tính bằng cách sử dụng nhiều phương tiện truyền thông tin như: Text (văn bản), graphic (biểu đồ, đồ thị), animation (hoạt hình), image (ảnh chụp), video (hình ảnh), audio (âm thanh), hoặc kết hợp các media với nhau (video + audio + văn bản diễn giải)... [2]

Người ta thường phân media thành hai loại dựa trên quan hệ của chúng với thời gian. Đó là:

- Static media: Không có chiều thời gian. Thông tin không liên quan tới thời gian. Ví dụ cho loại này là văn bản, hình họa, ảnh chụp.
- Dynamic media: Có chiều thời gian. Thông tin có quan hệ chặt chẽ với thời gian và thông tin phải được trình diễn với thời gian xác định. Ví dụ các loại audio, video, animation, game online...

So với dữ liệu truyền thông như văn bản và số, dữ liệu đa phương tiện có một số đặc điểm rất khác biệt, đó là:

- ***Kích thước và số lượng dữ liệu đồ sộ***

- Kích thước dữ liệu lớn: dữ liệu đa phương tiện có kích thước lớn hơn nhiều so với các kiểu dữ liệu số và văn bản thông thường. Một văn bản thô có 200 từ (khoảng 1000 ký tự) chỉ có kích thước là 1kByte, nhưng nếu lưu văn bản đó bằng định dạng ảnh GIF thì kích thước gấp khoảng 10 lần. Một giọng nói đơn sắc được lưu với định dạng .WAVE trong thời gian 1 phút có kích thước khoảng 2640 kByte (đã nén) hoặc xấp xỉ 6-8 MB (chưa nén). Một cảnh video rất ngắn chứa hàng trăm bức ảnh với kích thước có thể lên đến hàng chục MB..., xem bảng minh họa:

Kiểu	Mô tả	Kích thước
Plain text	khoảng 200 từ (1000 ký tự)	1 kByte
Tệp Winword	khoảng 200 từ (1000 ký tự)	15 kByte
Ảnh GIF	khoảng 200 từ (1000 ký tự, 210 x 100mm)	10 kByte
Âm thanh WAVE	Giọng nói (1 phút, 22KHz, 16 bit, mono)	2640 kByte

- Số lượng dữ liệu đồ sộ: người ta ước tính, chỉ riêng trên WWW có số lượng lên đến hàng tỉ ảnh, hàng trăm triệu bài hát MP3 và vài chục triệu phim video.

- ***Một số dữ liệu đa phương tiện phụ thuộc thời gian***

Audio và video có thêm chiều thời gian. Khi trình diễn audio và video thì chất lượng của chúng phụ thuộc chặt chẽ vào tốc độ trình diễn. Ví dụ, video phải được trình diễn với tốc độ 25 đến 30 hình/giây để có thể cảm nhận được hình ảnh chuyển động trơn tru.

- ***Tìm kiếm dựa trên cơ sở tương tự***

Trong cơ sở dữ liệu quan hệ, phương pháp tìm kiếm truyền thông đối với dữ liệu dạng văn bản và số là tìm kiếm chính xác, hay còn gọi là "exact search". Đối với dữ liệu đa phương tiện, người dùng thường đặt ra yêu cầu tìm kiếm một đối tượng tương tự theo nội dung mà họ đưa ra. Ví dụ, một nghiên cứu khoa học cho biết con người có khả năng nhận biết một bài hát thông qua giai điệu (humming) tốt hơn thông qua tên bài hát. Mặt khác, có rất nhiều bài hát có cùng tên và chỉ khác nhau về giai điệu. Vì vậy, việc tìm kiếm một bài hát dựa trên giai điệu sẽ đáp ứng tốt hơn nhu cầu đầy tiềm năng của ngành công nghiệp giải trí.

Tuy nhiên, việc tìm kiếm tương tự có thể phải dựa trên các đặc trưng phức tạp (ví dụ, video có thể chứa văn bản, âm thanh, hình ảnh...).

- ***Đồng bộ***

Một số ứng dụng đa phương tiện sử dụng hệ thống thời gian thực. Hệ thống thời gian thực là hệ thống mà trong đó sự đúng đắn của việc thực hiện

thao tác không chỉ phụ thuộc vào việc thu được kết quả đúng mà còn phải đưa ra kết quả đúng thời điểm. Ví dụ, các tệp phim, bài giảng, truyền hình trực tiếp, hội nghị, hội thảo qua mạng (video conference), xem video theo yêu cầu (video on demand) ... thì yêu cầu hình ảnh phải được đồng bộ với âm thanh.

- ***Chất lượng dịch vụ (Quality of Service- QoS)***

QoS là một tập các yêu cầu về chất lượng đối với các hoạt động tổng thể chung của một hoặc nhiều đối tượng. Các tham số QoS mô tả tốc độ và độ tin cậy của việc truyền dữ liệu như thông lượng, trễ, tỷ lệ lỗi... Các ứng dụng đa phương tiện khi truyền qua mạng thường đòi hỏi yêu cầu cao về QoS, nhất là các dịch vụ đa phương tiện tương tác thời gian thực như điện thoại internet, hội thảo qua mạng. Các dịch vụ này thường đòi hỏi khắt khe về độ trễ (tối đa là vài trăm ms). Để xác định QoS, người ta dựa vào các tham số sau đây:

- Độ trễ: là khoảng thời gian cực đại để truyền dữ liệu.

- Jitter: là độ biến đổi độ trễ.

- Thông lượng: là tổng số dữ liệu cực đại được truyền đi trên một đơn vị thời gian.

- Tỷ số mất tin: là số dữ liệu cực đại bị mất trên một đơn vị thời gian.

1.2. TỔNG QUAN CỦA CSDL ĐA PHƯƠNG TIỆN

1.2.1. Khái niệm

Hệ thống quản trị cơ sở dữ liệu đa phương tiện là hệ thống tổ chức và lưu giữ, bao gồm các dữ liệu truyền thông và các loại dữ liệu trừu tượng.

Một định nghĩa khác, theo Libor Janek và Goutham Alluri, hệ thống quản trị cơ sở dữ liệu đa phương tiện là một cơ cấu tổ chức quản lý các kiểu dữ liệu khác nhau, có khả năng thể hiện trong các định dạng trên một phạm vi các nguồn phương tiện đa dạng. [2]

Lượng dữ liệu đa phương tiện phát sinh theo nhu cầu hiện nay được lưu

trữ là một con số khổng lồ. Chỉ riêng với dữ liệu video, người ta ước tính có khoảng 21264 trạm truyền hình phát 16 giờ hàng ngày, sinh ra khoảng 31 tỉ giờ. Tuy nhiên, các hệ quản trị cơ sở dữ liệu đã được sử dụng rộng rãi như cơ sở dữ liệu quan hệ, chủ yếu tập trung vào quản lý các tài liệu văn bản thì không đáp ứng đầy đủ đối với việc quản lý các dữ liệu đa phương tiện, bởi các tính chất cũng như các yêu cầu đặc biệt của chúng như đã nêu ở trên. Do đó, hệ thống quản trị cơ sở dữ liệu đa phương tiện là sự cần thiết để quản lý dữ liệu đa phương tiện một cách có hiệu quả.

1.2.2. Kiến trúc cơ sở dữ liệu đa phương tiện (MMDBMS)

Phát triển một MMDBMS bao gồm các bước sau:

Bước 1. Thu thập media

Các dữ liệu media được thu thập từ các nguồn khác nhau như ti vi, CD, www...

Bước 2. Xử lý media

Mô tả các đoạn trích media và các đặc trưng của chúng, bao gồm cả lọc nhiễu và tách thô...

Bước 3. Lưu trữ media

Dựa vào yêu cầu cụ thể của ứng dụng để lưu dữ liệu và các đặc trưng của chúng vào hệ thống.

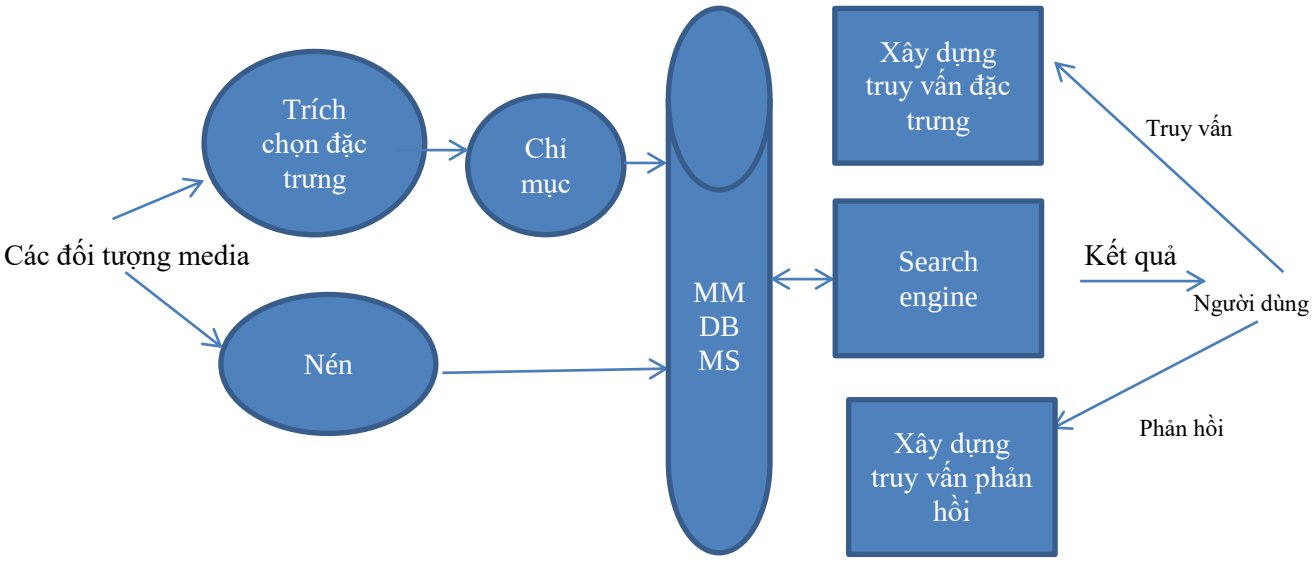
Bước 4. Tổ chức media

Tổ chức các đặc trưng để phục vụ việc truy tìm. Ví dụ, chỉ mục các đặc trưng với các cấu trúc giúp khai thác hiệu quả.

Bước 5. Xử lý truy vấn media

Là quá trình làm cho thích nghi với cấu trúc chỉ mục. Thiết kế các giải thuật tìm kiếm hiệu quả.

Kiến trúc chung cho một MMDBMS được minh họa như sau:



Hình 1.1: Kiến trúc chung của một MMDBMS

Hệ thống cơ sở dữ liệu đa phương tiện có nhiều môđun chức năng khác nhau nhằm hỗ trợ các thao tác trên dữ liệu đa phương tiện. Bao gồm các môđun chính sau đây:

- Giao diện người dùng.

- Bộ trích chọn đặc trưng.

- Chỉ số hóa và môtơ tìm kiếm.

- Quản lý truyền thông.

Trong đó, có hai thao tác cơ bản là:

- ***Bổ sung dữ liệu đa phương tiện mới***

Thao tác bổ sung được thực hiện theo trình tự các bước như sau:

- Bước 1. Dữ liệu đa phương tiện mới được bổ sung thông qua nhiều cách khác nhau như nhập trực tiếp từ bàn phím, từ microphone hay từ

bất kỳ thiết bị nhập kỹ thuật số khác. Dữ liệu đa phương tiện cũng có thể được lấy từ các tệp đã lưu sẵn.

- Bước 2. Sau khi dữ liệu đa phương tiện được bổ sung, nội dung của chúng được trích chọn bằng công cụ trích chọn đặc trưng.

- Bước 3. Các dữ liệu đa phương tiện được bổ sung cùng với các đặc trưng của nó, thông qua bộ quản lý truyền tin được gửi về máy chủ.

- Bước 4. Tại máy chủ, các đặc trưng được bố trí về các vị trí phù hợp dựa vào lược đồ chỉ số hóa.

- Bước 5. Các dữ liệu đa phương tiện bổ sung cùng với các đặc trưng và chỉ số hóa phát sinh được lưu vào bộ quản lý lưu trữ.

- ***Truy vấn***

Thao tác truy vấn được thực hiện theo trình tự các bước như sau:

- Bước 1. Tại giao diện người dùng, người sử dụng truy vấn thông tin thông qua một thiết bị nhập nào đó, thông qua tệp đã được lưu trước đó hoặc có thể lấy trực tiếp từ cơ sở dữ liệu MMDBMS.

- Bước 2. Nếu truy vấn của người sử dụng không được lấy trực tiếp từ cơ sở dữ liệu trong MMDBMS thì thực hiện như sau:

+ Thực hiện trích chọn đặc trưng truy vấn.

+ Gửi các trích chọn đặc trưng đó đến máy chủ.

+ Môtơ chỉ số hóa tìm kiếm các mục dữ liệu phù hợp với truy vấn trong cơ sở dữ liệu.

+ Hiển thị kết quả đến người sử dụng thông qua giao diện người dùng.

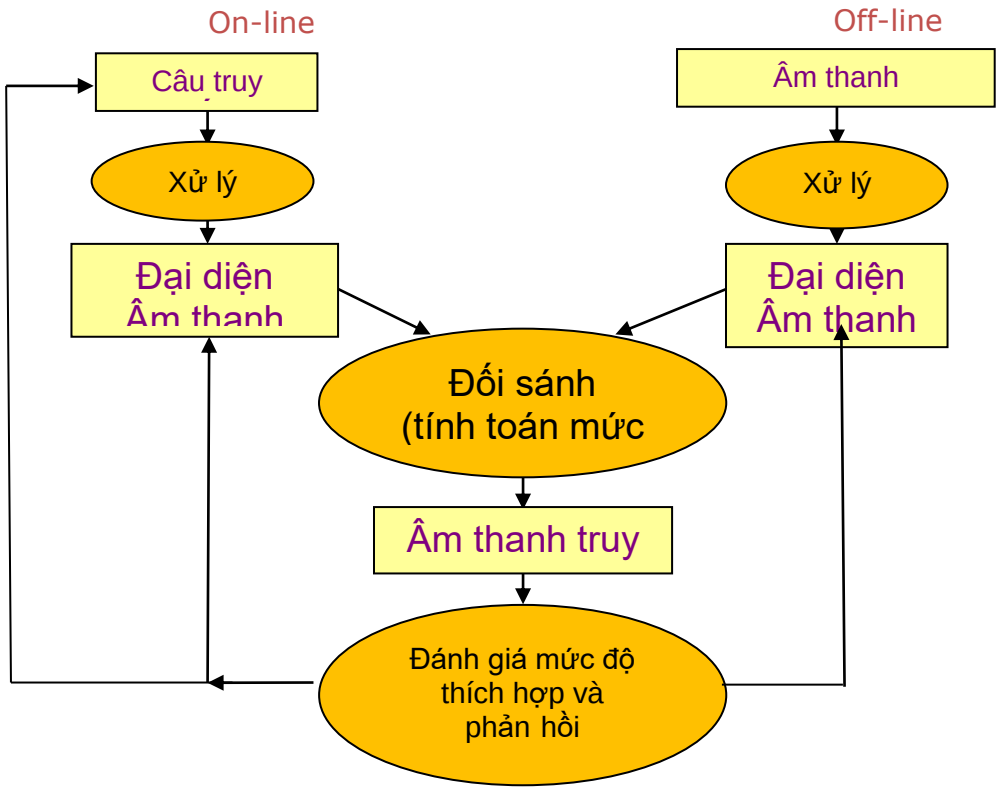
1.2.3. Đặc trưng của một cơ sở dữ liệu đa phương tiện

Các đặc trưng chủ yếu của MMDBMS bao gồm:

- *Quản lý dữ liệu đa phương tiện đã được lưu trữ*: các dữ liệu đa phương tiện được lưu trữ để quản lý gồm cả các thiết bị bên trong và bên ngoài

máy tính, ví dụ dữ liệu lưu trữ trên CD ROM...

- *Các phương pháp tìm kiếm dựa theo mô tả*: ví dụ, người dùng có thể đưa ra một mô tả để tìm kiếm "tiếng chuông điện thoại"...
- *Giao diện người dùng độc lập với thiết bị*: người dùng không cần biết cách thức lưu trữ dữ liệu đa phương tiện như thế nào.
- *Giao diện người dùng độc lập với các định dạng*: các truy vấn dữ liệu đa phương tiện có thể độc lập với định dạng dữ liệu. Nó cho phép có thể sử dụng các kỹ thuật lưu trữ mới mà không cần thay đổi ứng dụng cơ sở dữ liệu hiện có.
- *Cho phép thực hiện nhiều truy cập dữ liệu đồng thời*: dữ liệu đa phương tiện có thể truy cập đồng thời qua nhiều câu truy vấn khác nhau bởi một số ứng dụng. Cách truy cập nhất quán nhằm chia sẻ dữ liệu có thể được thực hiện, và cần có cơ chế để thỏa mãn việc tránh tạo ra các xung đột.
- *Quản lý một lượng dữ liệu lớn*: hệ thống cần phải có khả năng lưu trữ và quản lý lượng dữ liệu lớn và thỏa mãn các truy vấn đối với các quan hệ của dữ liệu.
- *Vấn đề truyền dữ liệu đa phương tiện dựa trên thời gian thực*: điều khiển việc đọc/ghi dữ liệu liên tục phải được thực hiện dựa trên thời gian thực. Do lượng dữ liệu có thể là rất lớn (ví dụ, truyền video) nên việc truyền dữ liệu có thể tốn nhiều thời gian và nó còn đòi hỏi phải được thực hiện một cách chính xác.



Hình 1.2:Tìm kiếm dữ liệu đa phương tiện

1.3. KHÁI QUÁT CƠ SỞ DỮ LIỆU ÂM THANH

1.3.1. Một số khái niệm

1.3.1.1.Truy tìm thông tin

Truy tìm thông tin - Information Retrieval (IR) là kỹ thuật tìm kiếm thông tin được lưu trữ trên máy tính. Đối với dữ liệu đa phương tiện, việc truy tìm thông tin hiệu quả là dựa trên tìm kiếm tương tự. Hệ thống lưu trữ một tập các đối tượng đa phương tiện trong cơ sở dữ liệu. Người dùng đưa ra các truy vấn, và hệ thống tìm ra các đối tượng tương tự truy vấn trong cơ sở dữ liệu đã lưu trữ thỏa mãn yêu cầu của người dùng. Truy tìm thông tin trong MMDBMS có một số đặc điểm sau đây: [4]

- Sử dụng một khối lượng dữ liệu đặc tả lớn và phức tạp.
- Việc tiếp cận IR chủ yếu dựa trên các đặc trưng.
- Các dữ liệu thường có kích thước lớn.
- Sự cần thiết phải có các kỹ thuật chỉ mục dữ liệu kích thước lớn để xử lý các

truy vấn một cách hiệu quả và thực hiện nhanh hơn so với phương pháp tìm kiếm tuần tự.

- Sự cần thiết phải tích hợp các đặc trưng media phức tạp một cách thường xuyên (ví dụ, dữ liệu ảnh có thể chứa các đặc trưng như: hình dạng, biểu đồ màu, kết cấu...).

Ý tưởng của phương pháp tìm kiếm tương tự đưa ra như sau:

- Cho một tập các đối tượng đa phương tiện trong MMDBMS.

- Tìm ra một hoặc một số K đối tượng tương tự (giống) nhất với đối tượng truy vấn mong muốn một cách nhanh chóng.

Đo tính tương tự

a. <u>Mô tả</u>:

Cho một tập các đối tượng đa phương tiện DB hoặc cho một điểm P nào đó trong một không gian mảng d chiều $DS=[0,1]^d$.

Truy vấn Q là một vectơ đặc trưng d chiều được tách ra từ đối tượng cần truy vấn. Biểu thức truy vấn có thể thay đổi (ví dụ, trọng số...).

Gọi D(P,Q) là hàm khoảng cách về tính tương tự giữa P và Q.

b. <u>Các thao tác</u>:

Thao tác thực hiện chi tiết các mô tả nêu trên bao gồm:

- Chỉ mục

Ban đầu, dữ liệu trong cơ sở dữ liệu được tiền xử lý để trích chọn đặc trưng và được chỉ số hóa dựa trên cơ sở đặc trưng và ngữ nghĩa. Kết quả được vectơ đặc trưng của dữ liệu đó.

- Truy vấn

Khi người sử dụng truy vấn thông tin thì câu truy vấn thông tin của người sử dụng được trích chọn các đặc trưng chính. Kết quả được vectơ truy vấn.

- Đo tính tương tự

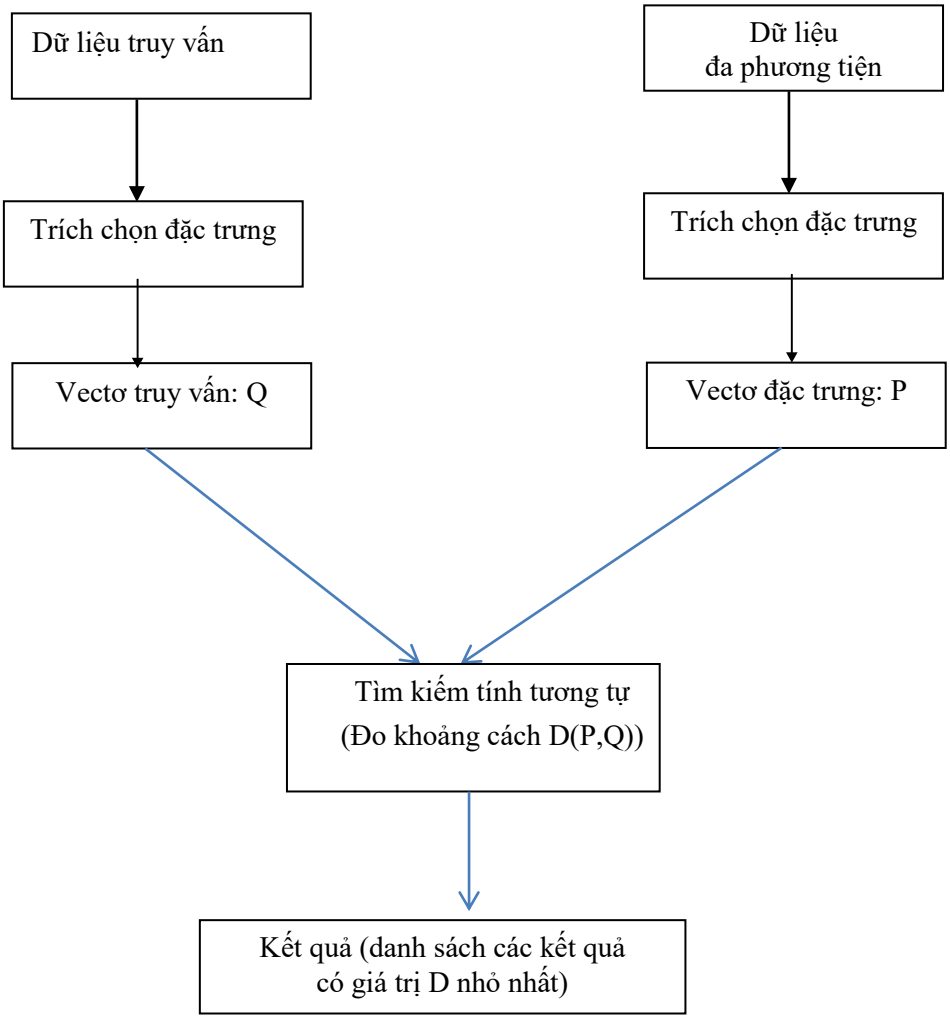
Các đặc trưng của vectơ đặc trưng trong cơ sở dữ liệu và vectơ truy vấn được đem ra so sánh, giá trị so sánh cho ta khoảng cách d.

- Kết quả

Nếu vectơ đặc trưng nào trong cơ sở dữ liệu gần với vectơ truy vấn nhất, tức là khoảng cách d nhỏ nhất thì được tìm ra và trình diễn cho người sử dụng.

c. Mô hình:

Mô hình thao tác MMDBMS nêu trên được thể hiện như sau:



Hình 1.3: Mô hình thao tác MMDBMS

d. Tính chất:

Cho P và Q là hai đối tượng trong không gian Metric. Khoảng cách D(P,Q) đo tính tương tự của P và Q có một số tính chất sau đây:

- Tính đối xứng (Symmetry): $D(P,Q) = D(Q,P)$
- Tính bất biến (Constancy of Self- Similarity): $D(P,P) = 0$
- Tính tuyệt đối (Positivity): $D(P,Q)>0$ nếu $P \neq Q$
- Tính không đều tam giác (Triangular Inequality): $D(P,Q) \leq D(P,O)+D(Q,O)$

1.3.2. Dữ liệu âm thanh

1.3.2.1. Các đặc trưng cơ bản của âm thanh

Âm thanh là sự lan truyền áp suất không khí trong không gian, âm thanh có các đặc trưng vật lý và sinh lý.

Các đặc trưng Vật lý :

- Tần số âm thanh : là tần số dao động của sóng âm, tính theo đơn vị Hz, KHz.
- Cường độ âm thanh : độ lớn biên độ sóng âm, đặc trưng cho công suất của nguồn phát âm. Đơn vị của cường độ là W/m^2.

Các đặc trưng sinh lý : liên quan đến sự cảm nhận âm thanh của tai người.

- Âm sắc : là sắc thái cao thấp, trầm bổng (liên quan đến phổ tần số của sóng âm)
- Âm lượng : cảm giác to, nhỏ của tai người khi nghe, âm lượng liên quan đến cường độ của sóng âm. Âm lượng là một đại lượng tương đối, được đo bằng Decibend (dB). Người ta quy ước giá trị cường độ ngưỡng nhỏ nhất mà tai người còn có thể cảm nhận được âm thanh là 10^{-12} W/m^2 ứng với mức âm lượng 0 dB [4]. Từ đó xác định được :

Âm lượng của âm thanh trong hội trường lớn là cỡ 60 dB ;

Nhà máy : 80 dB;

Ngưỡng cảm nhận lớn nhất của tai người: 120 dB;

Ngưỡng đau của tai người : 140 dB.

1.3.2.2. Âm thanh số

Số hoá là quá trình biểu diễn âm thanh tương tự dưới dạng rời rạc và được mã hoá dưới dạng các con số nhị phân để xử lý trong máy tính hoặc các thiết bị đa phương tiện số.

Một âm thanh có thể là tổ hợp của nhiều tần số, tần số chính bao trùm trong âm được gọi là tần số cơ bản. Trong tiếng nói tần số cơ bản là đáp ứng của sự rung động các dây thanh âm, tần số cơ bản thường được ký hiệu là $\mathbf{F_0}$.

Đơn vị của tần số là Hertz, ký hiệu là Hz. Mỗi Hz bằng một dao động/một giây. Và 1 KHz sẽ bằng 1000 Hz.

Các thông số chủ yếu của âm thanh số hoá là :

a. Lấy mẫu âm thanh

Lấy mẫu âm thanh là quá trình tạo ra tín hiệu âm thanh rời rạc hoặc tín hiệu số từ tín hiệu âm thanh dạng tương tự. Tần số lấy mẫu là số lần lấy mẫu được tính trong một đơn vị thời gian, thông thường là giây. Tần số lấy mẫu ký hiệu là Fs

Khoảng thời gian mà quá trình lấy mẫu được lặp lại gọi là chu kỳ lấy mẫu.

Ví dụ: Fs = 11025Hz nghĩa là 1s ta thu được 11025 mẫu và 1ms thu được $11025/1000 \approx 11$ mẫu.

Định lý lấy mẫu Shannon :

Định lý Shannon: Để đảm bảo thu được tín hiệu số hoá trung thực trong mức cho phép với tín hiệu lấy mẫu, tần số lấy mẫu phải tối thiểu lớn hơn hai lần tần số lớn nhất xuất hiện trong tín hiệu lấy mẫu.

Các âm thanh số hóa tiêu chuẩn thường được lấy mẫu với các tần số từ 6000 đến 192000 Hz, và thường là các tần số 6000, 8000, 11025 , 22050 ,

44100 , 48000, 96000 Hz.

Tần số âm thanh con người có thể cảm nhận được nằm trong khoảng từ 20 đến 20000 Hz. Tuy nhiên, tần số tiếng nói của con người chỉ nằm trong khoảng 8000 Hz. Tai người đặc biệt nhạy cảm với những tần số trong tín hiệu tiếng nói chứa thông tin phù hợp nhất với việc liên lạc (những tần số xấp xỉ **200 – 5600 Hz**). Người nghe có thể phân biệt được những sự khác biệt nhỏ trong thời gian và tần số của những âm thanh nằm trong vùng tần số này.

Do vậy, theo định lý lấy mẫu Shannon, tần số lấy mẫu cho tiếng nói chỉ cần cỡ **11025 Hz** hoặc 22050 Hz là vừa. Nếu lấy mẫu với tần số quá cao thì số lượng mẫu thu được rất lớn và gây khó khăn hơn trong việc xử lý chúng, ngược lại, nếu lấy mẫu với tần số quá thấp thì sẽ làm biến dạng và mất mát thông tin trong âm thanh.

b. Lượng tử hoá âm thanh

Quá trình biểu diễn trị số của các mẫu bởi một giá trị xác định nằm trong phạm vi biểu diễn bởi số byte mã hoá được gọi là quá trình lượng tử hoá. Số byte dùng trong mã hoá âm thanh thường là 1, 2 hoặc 4 byte. VD mã hoá âm thanh bởi 8 bit (1 byte) sẽ phân chia giá trị các mẫu âm thanh ra làm 256 mức, trong khoảng từ 0 đến 511 hoặc từ -256 đến 255.

Nếu mã hóa âm thanh bởi ít byte thì số mức để biểu diễn trị số của các mẫu thu được là ít, do đó phải làm tròn trị số của các mẫu với sai số lớn, điều này đồng nghĩa với việc làm sai lệch hay làm biến dạng âm thanh ở một mức độ nào đó, tuy nhiên thu được lợi điểm là dung lượng tệp âm thanh thu được là nhỏ. Ngược lại, nếu dùng quá nhiều byte để mã hoá thì sẽ thu được âm thanh với độ trung thực cao, tuy nhiên phải trả giá cho dung lượng lớn của tệp âm thanh số hoá thu được.

	F_0 trung bình (Hz)	F_0 thấp nhất	F_0 cao nhất
Nam	125	80	200

Nữ	225	150	350
Trẻ em	300	200	500

Vì tần số cơ bản là tần số dao động của dây thanh âm nên đối với mỗi người, giá trị này khá ổn định đối với các nguyên âm khác nhau. Một số kết quả khảo sát cho thấy nó chỉ thay đổi khoảng 5% giữa các nguyên âm khác nhau. Với cùng một người, khi phát âm ở các thời điểm khác nhau, tần số cơ bản cũng có sự thay đổi nhỏ.

Tần số cơ bản càng lớn thì âm thanh phát ra có cao độ càng lớn, hay nói cách khác đặc trưng Pitch của âm thanh đó càng cao.

Trong lĩnh vực nhận dạng tiếng nói, tần số cơ bản được sử dụng phối hợp với các đặc trưng khác để tăng cường độ chính xác.

1.3.3. Giới thiệu Cơ sở dữ liệu âm thanh

Tương tự dữ liệu ảnh và dữ liệu video, dữ liệu âm thanh được đặc trưng bởi hai cách cơ bản: sử dụng *metadata* để diễn giải nội dung tệp âm thanh hay tách đặc trưng thích hợp của dữ liệu âm thanh nhờ kỹ thuật xử lý tín hiệu. Chúng ta sẽ khảo sát tổng quan cả hai kỹ thuật này.

- Biểu diễn nội dung âm thanh bằng metadata

Tổng quát thì *metadata* được sử dụng để biểu diễn nội dung âm thanh được xem như tập các đối tượng trải dài theo đường thời gian, tương tự video. Các đối tượng, đặc trưng và hoạt động xảy ra trong âm thanh hoàn toàn tương tự như trong video. Sự khác biệt ở chỗ, âm thanh để nghe, còn video để cả nghe và nhìn. Như vậy, chúng ta có thể chỉ số hóa *metadata* kết hợp với

âm thanh theo cách tương tự cách chỉ số hoá video, và kỹ thuật xử lý truy vấn video cũng được sử dụng lại ở đây.

Phần lớn CSDL âm thanh đang tồn tại sử dụng lược đồ chỉ số hoá trên cơ sở *metadata*.

- Nội dung âm thanh trên cơ sở tín hiệu

Sử dụng *metadata* là tin cậy và được khuyến cáo khi có cách tạo ra *metadata*. Thí dụ, nếu ta tạo ra CSDL âm thanh của đài phát thanh hay ghi âm nhạc, thì hầu như không có vấn đề khi tạo ra *metadata*. Tuy nhiên, trong ứng dụng khác, như cảnh sát nghe trộm điện thoại của kẻ tình nghi bán ma tuý, việc tạo *metadata* sẽ phức tạp hơn bởi vì nhận danh của người nói có thể không được biết trước, thậm chí nội dung của hội thoại có thể không rõ ràng (nếu có sử dụng thiết bị trộn âm).

Trong trường hợp như vậy, quan niệm về *nội dung* được mô tả bằng khái niệm của các phương pháp xử lý tín hiệu trên đây.

CSDL âm thanh có thể được chỉ số hóa bằng các đặc trưng của tín hiệu âm thanh như: Cường độ, âm lượng, độ trong, ...

CHƯƠNG 2: TRÍCH CHỌN ĐẶC TRƯNG ÂM THANH

2.1. Khái quát về đặc trưng chính của âm thanh

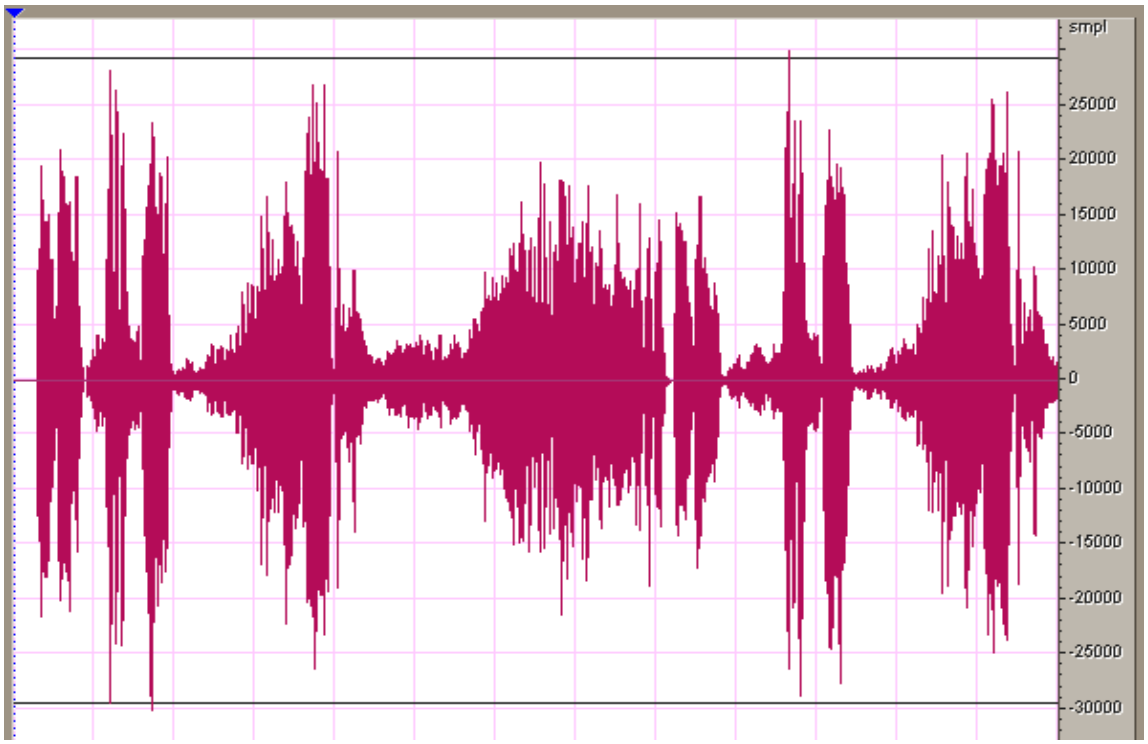
Trong thực tế, trích chọn đặc trưng đóng vai trò rất quan trọng trong vấn đề phân lớp âm thanh. Chúng cho thấy đặc trưng quan trọng của các loại tín hiệu âm thanh khác nhau. Để nâng cao tính chính xác của việc phân lớp âm thanh, ta cần phải lựa chọn các đặc trưng tốt. Đa số các phương pháp, giải thuật trích chọn đặc trưng âm thanh hiện nay đều xem các đặc trưng sau đây là hiệu quả để phân lớp và phân đoạn âm thanh.

2.2. Các đặc trưng âm thanh trong miền thời gian

Biểu diễn trong miền thời gian hay thời gian - biên độ là kỹ thuật trình diễn tín hiệu cơ bản nhất, trong đó tín hiệu được biểu diễn như biên độ biến

đổi theo thời gian. [2]

Hình 2.1 là thí dụ tín hiệu âm thanh số trong miền thời gian. Im lặng (câm) được biểu diễn bởi giá trị 0. Giá trị tín hiệu có thể âm hay dương phụ thuộc vào áp suất âm thanh cao hơn hay thấp hơn áp suất cân bằng khi im lặng. Giả sử rằng sử dụng 16 bít để mã hóa mẫu audio, thì ta có giá trị tín hiệu sẽ trong khoảng từ 32767 đến -32767.



Hình 2.1: Tín hiệu âm thanh số theo miền thời gian

Từ cách biểu diễn trên đây ta dễ dàng có được năng lượng trung bình, tốc độ vượt qua

0 (zero crossing rate) và tỷ lệ câm (silence ratio).

2.2.1. Năng lượng trungbình

Năng lượng trung bình chỉ ra âm lượng (loudness) của tín hiệu audio. Có nhiều cách để tính nó. Một cách tính đơn giản như sau:

$$E = \frac{\sum_{N=0}^{N-1} x(n)^2}{N}$$

trong đó, E là năng lượng trung bình của đoạn audio, N là tổng số mẫu trong đoạn audio, x(n) là giá trị của mẫu n.

2.2.2. Zero crossing rate

Tốc độ vượt q ua 0 chỉ ra tần số thay đổi của dấu biên độ tín hiệu. Nói cách khác nó chỉ ra tần số trung bình của tín hiệu. Tốc độ vượt qua 0 được tính như sau:

$$ZC = \sum_{N=1}^{N} \frac{|sgn\ x(n) - sgn\ x(n-1)|}{2N}$$

trong đó, sgn x(n) là dấu của x(n) và có giá trị 1 nếu x(n) dương, giá trị -1 nếu x(n) có giá trị âm.

2.2.3. Silence ratio

Tỷ lệ câm chỉ ra kích thước đoạn âm thanh câm. Câm được định nghĩa như chu kỳ trong đó giá trị biên độ tuyệt đối của một số mẫu nhỏ hơn ngưỡng nào đó. Có hai loại ngưỡng: ngưỡng biên độ và ngưỡng thời gian.

Với ngưỡng biên độ, mẫu được xem như là câm khi biên độ của nó nhỏ hơn ngưỡng biên độ. Chỉ một mẫu câm không được xem như chu kỳ câm. Chỉ khi tổng các mẫu câm liên tục vượt qua ngưỡng thời gian nào đó thì các mẫu này hình thành chu kỳ câm (silence period).

Tỷ lệ câm được tính bằng tỷ lệ giữa tổng chu kỳ câm và tổng độ dài của đoạn âm thanh.

2.3.Các đặc trưng âm thanh trong miền tần số

2.3.1. Phổ âm thanh

Biểu diễn miền thời gian không chỉ ra được các thành phần tần số và phân bổ tần số của tín hiệu âm thanh. Biểu diễn miền tần số suy diễn từ biểu diễn miền không gian bằng biến đổi Fourier. Biến đổi Fourier được xem như

tách tín hiệu thành các thành phần tần số. Trong miền tấn số, tín hiệu được biểu diễn bởi biên độ biến đổi theo tần số, chỉ ra tổng năng lượng tại các tần số khác nhau. Biểu diễn miền tần số của tín hiệu được gọi là phổ của tín hiệu. [2]

Hình 2.2 là phổ của tín hiệu âm thanh của hình 2.1. Xuất phát từ phổ tín hiệu, dễ dàng nhận ra phân bổ năng lượng theo dải tần số. Vì quan tâm đến tín hiệu số cho nên ta sử dụng DFT để suy diễn ra phổ tín hiệu. Công thức tính DFT nhưsau:

$$X(k) = \sum_{N=0}^{N-1} x(n) e^{jn}\, \omega_k$$

trong đó, $\omega_k = \frac{2\pi k}{N}$, x(n) là tín hiệu rời rạc với N mẫu, k là DFT bin.

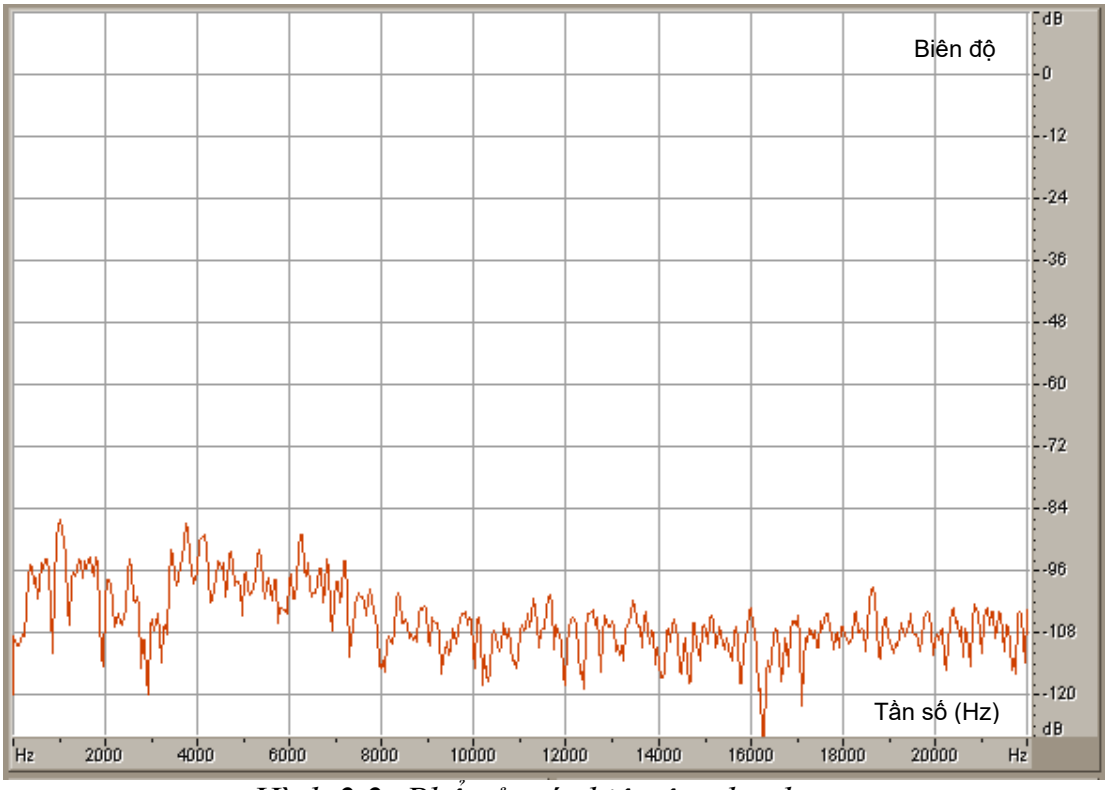
Nếu tần số lấy mẫu tín hiệu là f_s Hz thì tần số f_k của bin k sẽ là:

$$f_{k=f_s}\ \frac{\omega_k}{2\pi} = f_s\ \frac{k}{N}$$

Nếu x(n) có giới hạn thời gian là N thì nó có thể khôi phục hoàn toàn bằng IDFT của N mẫu tần số như sau:

$$x(n) \quad = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{jn\omega_k}$$

Các giá trị DFT và IDFT được tính toán hiệu quả bằng thuật toán FFT.



Hình 2.2: Phổ của tín hiệu âm thanh

Như nói trên, DFT làm việc với tín hiệu rời rạc có giới hạn độ dài (N). Thực tế, rất nhiều tín hiệu trong khoảng thời gian dài. Rất khó tính toán DFT với N rất lớn. Để giải quyết vấn đề này, người ta sử dụng STFT (Short Time Fourier Transform). Trong đó, tín hiệu với độ dài tùy ý được chia thành các khối gọi là frame và DFT áp dụng cho từng frame. Frame được hình thành bằng cách nhân tín hiệu gốc với hàm cửa sổ. Thông thường độ dài frame khoảng 10 đến 20 ms được sử dụng vào phân tích không gian.

Sau đây là một số đặc trưng suy diễn từ phổ tín hiệu.

2.3.2. Bandwidth

Băng thông chỉ ra dải tần số của âm thanh. Tín hiệu nhạc thường có băng thông cao hơn tín hiệu tiếng nói. Cách tính băng thông đơn giản nhất là

lấy chênh lệch tần số giữa tần số cao nhất với tần số thấp nhất của các thành phần phổ khác không. Trong một số trường hợp, "nonzero" được xác định khoảng 3 dB trên mức câm.

2.3.3. Phân bổ năng lượng

Từ phổ tín hiệu, chúng ta dễ dàng nhận thấy phân bổ tín hiệu theo **các** thành phần tần số. Thí dụ, chúng ta có thể qua sát thấy nó nếu tín hiệu có thành phần tần số cao đáng kể. Thông tin này có ích cho phân lớp audio bởi vì âm nhạc thường có các thành phần tần số cao hơn tiếng nói.

Việc tính toán năng lượng dải tần số cao và tần số thấp là cần thiết. Thực sự, khái niệm "low", "high" phụ thuộc vào ứng dụng. Thí dụ tần số tín hiệu tiếng nói ít khi vượt qua 7 kHz. Do vậy, ta có thể chia toàn bộ phổ dọc theo đường ngang 7 kHz: nửa dưới thuộc tần số thấp và nửa trên thuộc tần số cao. Tổng năng lượng cho mỗi băng được tính bằng tổng năng lượng mỗi mẫu trong băng.

Một đặc trưng quan trọng suy diễn từ phân bổ năng lượng là trọng tâm phổ (centroid). Nó là điểm giữa của phân bổ năng lượng phổ tín hiệu. Tiếng nói có trọng tâm thấp so với âm nhạc. Trọng tâm còn được gọi là độ chói (brightness).

2.3.4. Điều hòa (Harmonicity)

Đặc trưng thứ hai trong miền tần số của âm thanh là điều hòa. Trong âm thanh điều hòa, các thành phần phổ là số lần nguyên của tần số thấp nhất và tần số thường xuyên cao nhất. Tần số thấp nhất được gọi là tần số cơ bản. Âm nhạc thường điều hòa hơn âm thanh khác. Để xác định được âm thanh có điều hòa hay không hãy kiểm tra xem tần số của các thành phần trội là số lần tần số cơ bản hay không.

Thí dụ, phổ âm thanh nốt G4 của tiếng sáo có dãy tần số cao độ (pitch) như sau:

400 Hz, 800 Hz, 1200 Hz, 1600 Hz...

Ta có thể viết dãy trên như sau:

f, 2f, 3f, 4f...

trong đó, f=400 Hz là tần số cơ bản của âm thanh. Các thành phần có tần số nf được gọi là điều hòa của nốt nhạc. [1]

2.3.5. Cao độ (Pitch)

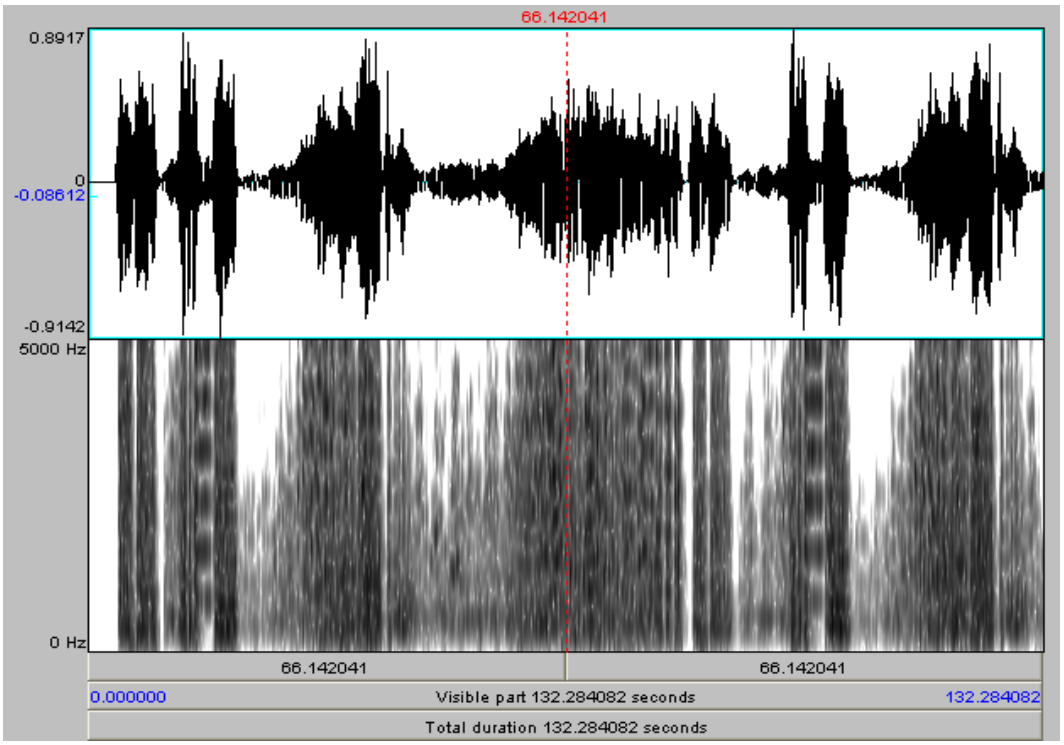
Đặc trưng thứ ba của miền tần số là cao độ. Thuộc tính cảm nhận các tần số âm thanh còn được gọi là pitch. Tần số càng cao thì pitch càng cao và bước sóng càng ngắn. Có thể xếp hàng âm thanh theo mức độ cao độ. Cao độ là đặc trưng chủ quan, nó quan hệ nhưng không tương đương với tần số cơ bản. Tuy nhiên, trong thực tế chúng ta sử dụng các tần số cơ bản để xấp xỉ cao độ.

2.3.6. Ảnh phổ (Spectrogram)

Biểu diễn âm lượng - thời gian và phổ là hai biểu diễn tín hiệu đơn giản nhất. Hạn chế của biểu diễn âm lượng - thời gian là không chỉ ra các thành phần tần số của tín hiệu và phổ, không chỉ ra khi nào các thành phần tần số khác nhau xuất hiện. Để giải quyết vấn đề này, biểu diễn tổ hợp hay còn gọi là ảnh phổ được sử dụng. Ảnh phổ của tín hiệu chỉ ra quan hệ giữa ba biến: nội dung tần số, thời gian và cường độ. Trong ảnh phổ, nội dung tần số được biểu diễn theo các trục tung, thời gian theo trục hoành. Cường độ (intensity, power) của các thành phần tần số khác nhau của tín hiệu được chỉ ra bằng độ xám, cường độ lớn hơn biểu diễn bởi mức độ xám cao hơn. [2]

Hình 2.3 chỉ ra ảnh phổ của tín hiệu âm thanh từ hình 2.2. Ảnh phổ mô tả rõ ràng các quan hệ giữa thời gian, tần số và biên độ.

Ta có thế xác định tính xuất hiện đều của một vài thành phần tần số từ ảnh phổ tín hiệu ảnh phổ âm nhạc đều hơn.



Hình 2.3: Ảnh phổ của tín hiệu âm thanh

2.3.7. Các đặc trưng chủ quan

Trừ cao độ (pitch), mọi đặc trưng mô tả trên có thể đo trực tiếp trong miền thời gian hay miền tần số. Còn những đặc trưng khác là có tính chủ quan, thí dụ âm sắc (timbre).

Âm sắc liên quan đến chất lượng âm thanh. Đặc trưng này chưa có định nghĩa và sự hiểu biết rõ ràng. Nó chứa đựng mọi chất lượng âm thanh khác với pitch (cao độ), loudness (âm lượng) và duration (độ dài). Các thành phần quan trọng của âm sắc bao gồm hình bao biên độ, điều hòa và hình bao phổ.

2.4. Đặc trưng âm thanh MFCC

2.4.1. Các bước tính MFCC

MFCC (*Mel Frequency Cepstral Coefficients*) là các hệ số biểu diễn phổ của phổ (*spectrum-of-a-spectrum*) của đoạn âm thanh.

Các hệ số *ceptral c(k)* là cách thuận tiện cho việc mô hình hóa phân bổ

năng lượng phổ [2]

$$c(k) = IDFT\{\log|DFT\{x(n)\}|\}$$

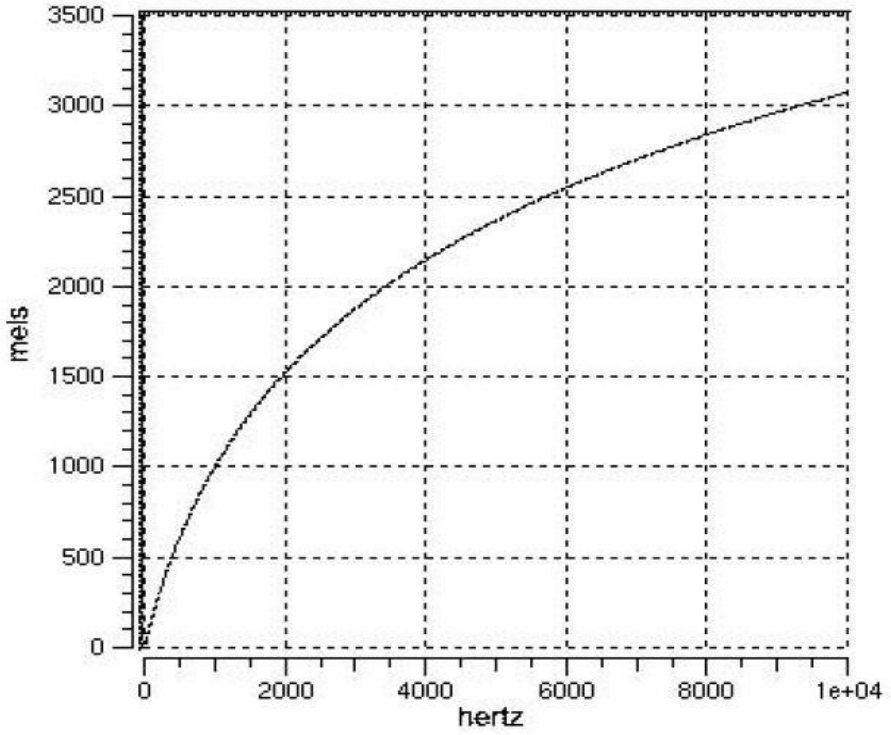
Các hệ số *ceptral* được tính toán cho mỗi khoảng thời gian ngắn của tín hiệu âm thanh.

Hệ số *ceptral* được sử dụng trong MMDBMS (tìm kiếm âm thanh) và trong nhận dạng tiếng nói

2.4.2. Đặc trưng âm thanh MFCC

Một cặp âm thanh cảm nhận có độ cao như nhau nếu giá trị mel của chúng như nhau.

Mel-scale: Xấp xỉ tuyến tính dưới 1 kHz, và loga trên 1 kHz.



Hình 2.4: Đặc trưng âm thanh MFCC

Hệ số *ceptral* được sử dụng trong MMDBMS (tìm kiếm âm thanh) và trong nhận dạng tiếng nói.

2.4.3. Phương pháp phân tích MFCC

a. Quá trình lọc theo thang Mel Cepstral

MFCC gồm các bước chính sau:

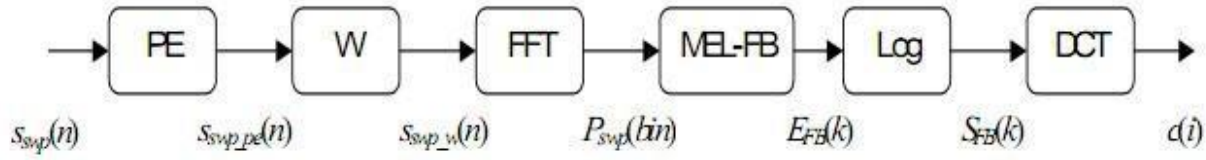
1. Phân khung tín hiệu

2. Cửa sổ hóa

3. Chuyển đổi sang miền tần số

4. Chuyển đổi sang thang Mel

5. Thực hiện biến đổi Cosine rời rạc.



Hình 2.5: Quy trình biến đổi MFCC

Quan sát quá trình trên ta thấy, âm thanh được chia thành những khung có độ dài cố định. Mục đích là để lấy mẫu những đoạn tín hiệu nhỏ (theo lý thuyết là ổn định). Hàm cửa sổ bỏ đi những hiệu ứng phụ và vector đặc trưng cepstral được thực hiện trên mỗi khung cửa sổ. Biến đổi Fourier rời rạc của mỗi khung được tính toán và lấy logarithm biên độ phổ. Thông tin về pha bị bỏ qua do biên độ phổ là quan trọng hơn pha. Thực hiện lấy logarithm biên độ phổ do âm lượng của tín hiệu là xấp xỉ logarith. Tiếp theo biến đổi phổ theo thang Mel. Từ kết quả này, trong vector Mel – spectral của các thành phần tương quan cao, bước cuối cùng là thực hiện biến đổi cosine rời rạc để tổng hợp vector phổ Mel để tương quan lại các thành phần này [2]. Mỗi phần này được trình bày chi tiết trong các phần sau.

b. Lấy mẫu

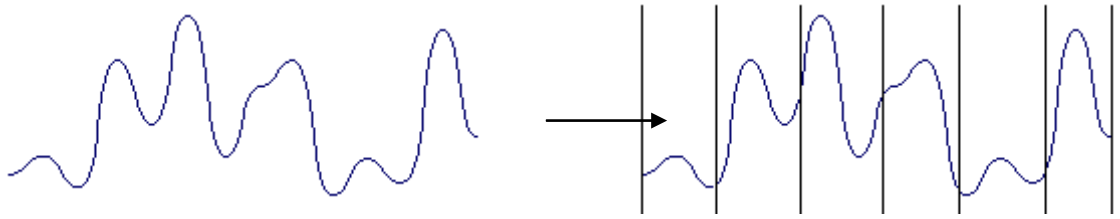
Trong việc lấy mẫu dữ liệu, chúng ta xem xét đến tín hiệu âm thanh

đã được số hóa bằng việc rời rạc hóa các giá trị trên những khoảng đều nhau vì vậy cần phải chắc chắn rằng tốc độ lấy mẫu là đủ lớn để mô tả tín hiệu dạng sóng. Tấn số lấy mẫu nên ít nhất gấp đôi tần số dạng sóng như trong định lý của Nyquist. Tốc độ lấy mẫu phổ biến là 8000, 11025, 22050, 44000. Thông thường sử dụng tần số trên 10kHz

c. Phân khung tín hiệu

Phân khung là quá trình chia mẫu tín hiệu thành một số khung chồng lấp lên nhau hoặc không. Mục đích của phân khung là để lấy mẫu các đoạn tính hiệu nhỏ (theo lý thuyết là ổn định). Vấn đề là bản chất của âm thanh là không

ổn định. Vì vậy, biến đổi Fourier sẽ thể hiện tần số xảy ra trên toàn miền thời gian thay vì thời gian cụ thể. Bởi thế khi tín hiệu là không ổn định, tín hiệu đó nên được chia nhỏ thành các cửa sổ rời rạc nhờ đó mỗi tín hiệu trong một cửa sổ trở nên tĩnh và phép biến đổi Fourier có thể thực hiện trên mỗi khung. Quá trình phân khung được thể hiện trong hình sau:



Hình 2.6: Phân khung tín hiệu

Trong khối này tín hiệu hiệu chỉnh $\tilde{s}(n)$ được phân thành các khung, mỗi

khung có N mẫu, hai khung kề lệch nhau M mẫu. Khung đầu tiên chứa N mẫu, khung thứ hai bắt đầu chậm hơn khung thứ nhất M mẫu và chồng lên khung thứ nhất N-M mẫu. Tương tự, khung thứ ba chậm hơn khung thứ nhất 2M mẫu (chậm hơn khung thứ hai M mẫu) và chờm lên khung thứ nhất N-2M mẫu. Quá trình này tiếp tục cho đến khi tất cả các mẫu tiếng nói cần phân tích

thuộc về một hoặc nhiều khung.

d. Lấy cửa sổ tín hiệu

Bước tiếp theo là lấy cửa sổ cho mỗi khung riêng rẽ nhằm giảm sự gián đoạn của tín hiệu tiếng nói tại đầu và cuối mỗi khung. Nếu w(n), $0 \leq n \leq N-1$, sau khi lấy cửa sổ được:

$\tilde{x}_n$ (k)=$x_n(k)w(n)$, $0 \leq n \leq N-1$

Thông thường, cửa sổ Hamming được sử dụng, cửa sổ này có dạng:

W(n)=0.54-0.46cos $\left(\frac{2\pi n}{N-1}\right)$ $0 \leq n \leq N-1$

0

n

N

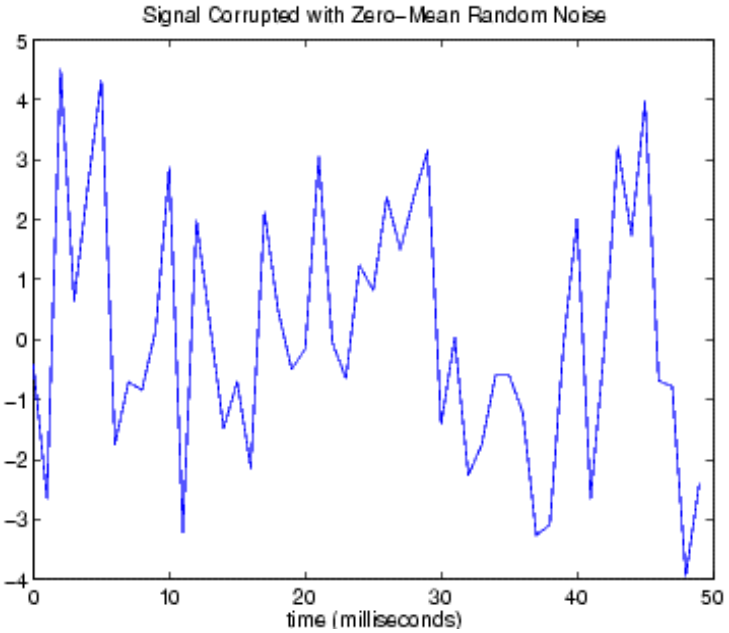
1

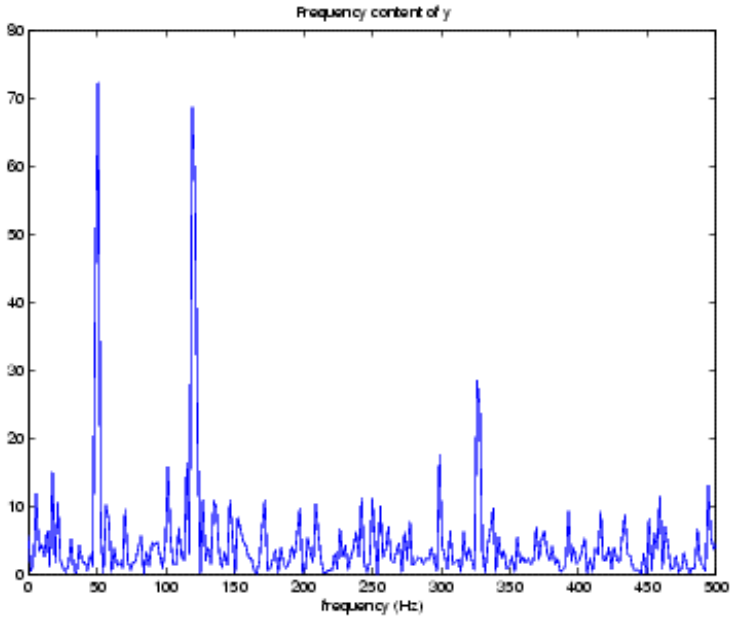
Ý tưởng ở đây là giảm bớt sự méo phổ bằng việc sử dụng các cửa sổ để giảm tín hiệu về không tại điểm bắt đầu và kết thúc mỗi khung.

Sự chồng lấp các khung để làm nhẵn khung đến khung kế tiếp. Lấy cửa sổ tín hiệu đảm bảo tất cả các phần của tín hiệu được khôi phục và loại trừ được khe hở giữa các khung. Việc này được thực hiện để khử tính không liên tục tại đường viền khung cho biến đổi Fourier thực hiện sau đó. Các khung có độ dài lớn hơn có cao độ pitch chính xác hơn và nén dữ liệu tốt hơn nhưng chất lượng giảm.

e. Biến đổi Fourier nhanh

Biến đổi Fourier rời rạc (DFT) hoặc biến đổi Fourier nhanh (FFT) được thực hiện để chuyển đổi mỗi khung với N mẫu từ miền thời gian sang miền tần số. Tín hiệu gốc cần được thực hiện biến đổi Fourier qua bộ lọc thông dải để xử lý độ lệch tần số Mel. Biến đổi Fourier chuẩn không được sử dụng do tín hiệu âm thanh không xác định trên toàn miền thời gian. Thông thường hay sử dụng biến đổi DFT. Hình sau thể hiện tín hiệu trên miền thời gian và mô tả tần số tương ứng củanó.





Hình 2.7: Tín hiệu trên miền thời gian và tần số tương ứng của nó

f. Chuyển đổi sang thang tần số Mel

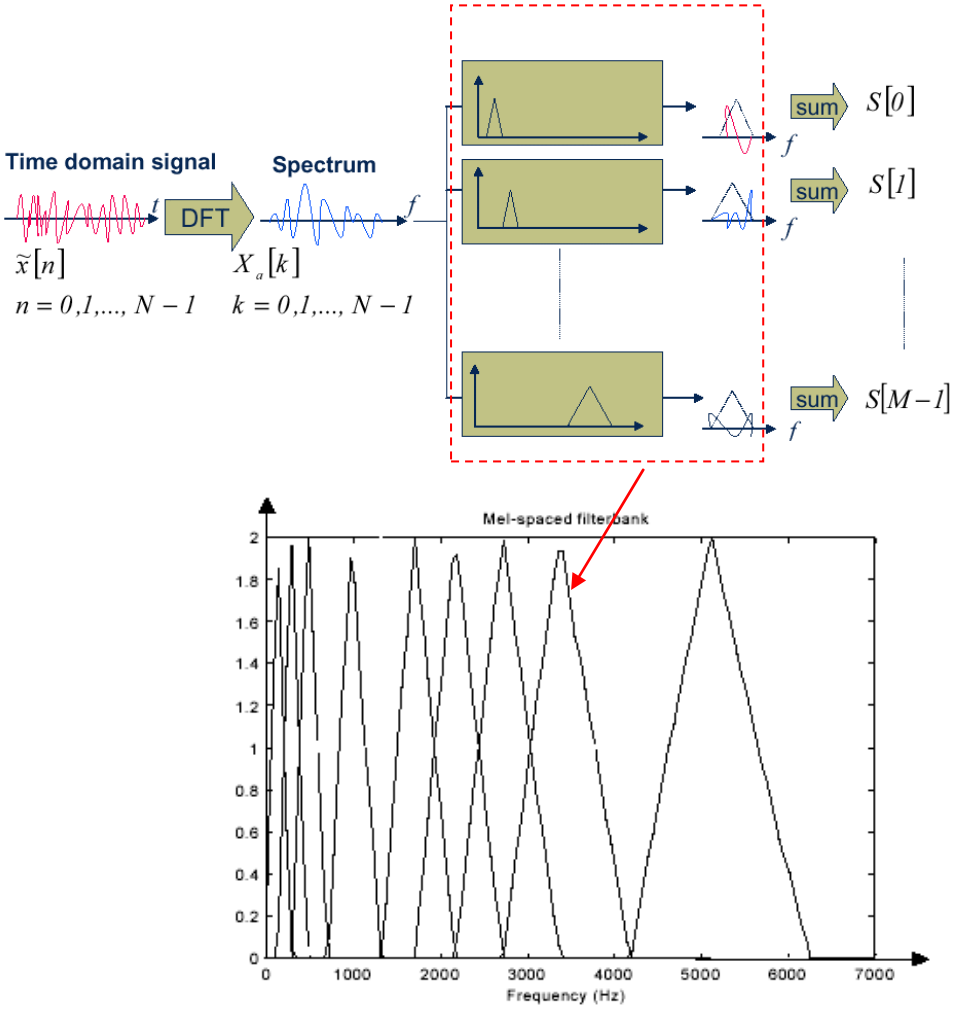
Để mô tả chính xác sự tiếp nhận tần số của hệ thống thính giác, người ta xây dựng một thang khác – thang Mel.

Việc chuyển đổi sang miền tần số Mel làm nhẵn phổ và làm nổi lên các tần số cảm thụ có nghĩa. Biến đổi Fourier lên tín hiệu qua bộ lọc thông dải để làm đơn giản phổ mà không làm mất dữ liệu. Điều này được thực hiện bằng cách tập hợp các thành phần phổ thành một dải tần số. Phổ được làm đơn giản hóa do sử dụng một dàn bộ lọc để tách phổ thành các kênh. Các bộ lọc được đặt cách đều nhau trên thang Mel và lấy logarit trên thang tần số, các kênh có tần số thấp là không gian tuyến tính trong khi các kênh có tần số cao là không gian logarit.

Tai người không cảm nhận sự thay đổi tần số của tiếng nói tuyến tính mà theo thang mel. Thang tần số Mel tuyến tính ở tần số dưới 1kHz và logarit ở tần số cao hơn 1kHz. Ta chọn tần số 1kHz, 40 dB trên ngưỡng nghe 1000 Mel. Do đó công thức gần đúng biểu diễn quan hệ tần số ở thang mel và thang tuyến tính nhưsau:

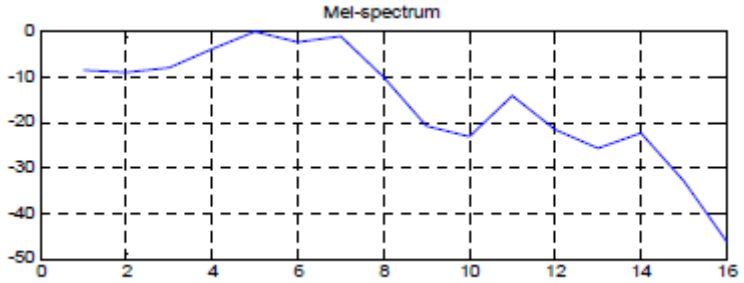
$$mel(f) = 2595 * \log_{10}(1 + \frac{f}{700})$$

Một phương pháp để chuyển đổi sang thang Mel là sử dụng băng lọc. Khoảng cách của băng lọc được định nghĩa bởi một hằng số tần số mel theo thời gian. Biến đổi phổ S(w) bao gồm đầu ra của bộ lọc khi S(w) là đầu vào. Băng lọc này được áp dụng trong miền tần số, nó có thể xem như các điểm thu được của bộ lọc chính. (Hình 2.8). Với các khung nhỏ tốt nhất là sử dụng các bộ lọc dạng tam giác hoặc thậm chí hình chữ nhật vì độ phân giải là quá thấp trong miền tần số thấp.



Hình 2.8: Băng lọc khoảng cách theo tần số mel

Mỗi bộ lọc trong băng lọc được nhân với phổ tín hiệu vì vậy chỉ có một giá trị đơn của cường độ trên bộ lọc được trả lại. Điều này có thể đạt được qua các tính toán của ma trận đơn. Kết quả là tổng của biên độ trong dải lọc và vì vậy làm giảm độ chính xác tới mức tai của con người. Hình 2.9 cho thấy kết quả này. Trục hoành mô tả chỉ số của bộ lọc và theo thang mel



Hình 2.9: Phổ sau khi lọc theo thang Mel

Quá trình chuyển đổi sang thang tần số mel được thực hiện theo ba bước sau:

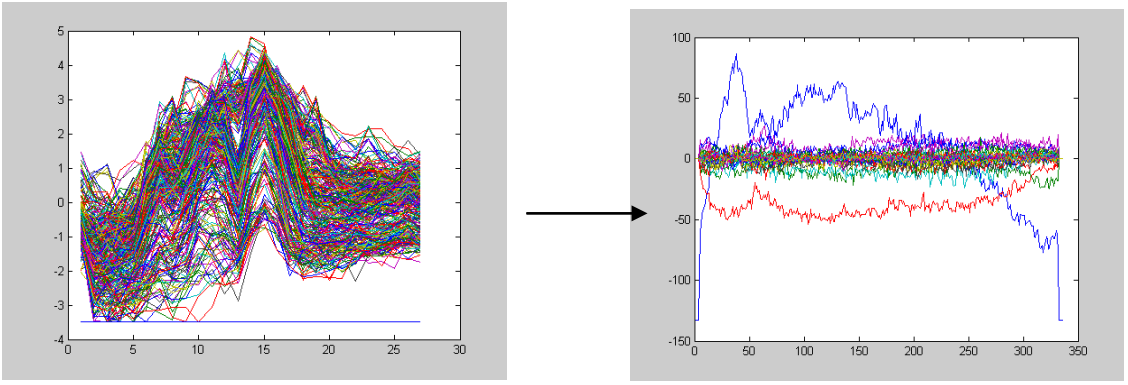
1. Cố định vùng giá trị dưới mỗi bộ lọc và đôi khi đưa thang về 1. Đặt M = số băng lọc yêu cầu

2. Phân bố đều trên thang tần số Mel

3. Chuyển đổi từ Hz sang $\omega i's$ trên thang tuyến tính. Mối quan hệ giữa mel và frq được cho bởi công thức: m=ln(1+f/700)*1000/ln(1+1000/700)

h. Biến đổi Cosin rời rạc

Ở bước cuối cùng này, sử dụng biến đổi cosin rời rạc để làm tương quan cường độ phổ logarit mel với hệ số tần số mel MFCC. Cepstrum là biến đổi ngược của tín hiệu theo log biên độ. Vì nguồn phổ là cân xứng nên thay biến đổi Fourier ngược thành biến đổi cosin rời rạc (DCT) . Thêm vào đó, DCT có khả năng tương quan cao hơn và hệ số cepstral chặt chẽ hơn. Hình dưới mô tả vector Mel-spectral với các thành phần tương quan cao tương quan lại thành hệ số tần số Mel 13



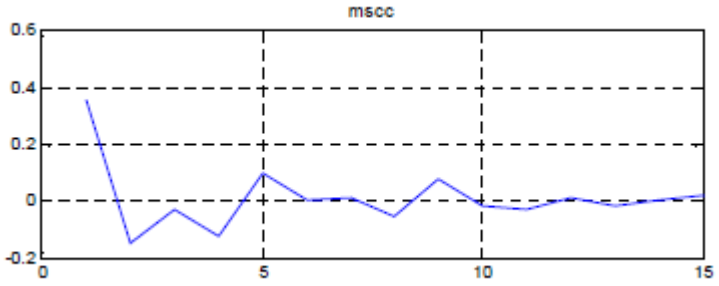
Hình 2.10: vector Mel-spectral với các thành phần tương quan cao tương quan lại thành hệ số tần số Mel 13

Sự rời rạc tính hiệu x(n) được định nghĩa trong biểu thức

$$y(k) = w(k)\sum_{n=1}^{N} x(n)\cos\frac{\pi(2n-1)(k-1)}{2N}, k = 1,..., N$$

Với: $w(k) = \begin{cases} \sqrt{1/N}, k = 1 \\ \sqrt{2/N}, 2 \le k \le N \end{cases}$

Thực hiện DCT, thu được tần số Mel Cepstrum (Hình 2.11) Có thể xem như hệ số thứ 0: C_0 được loại trừ. Lý do là nó đại diện cho các giá trị của tín hiệu vào mang ít thông tin. Beth Logan đã nói rằng hệ số cepstral thứ 0 chỉ chứa thông tin về năng lượng.



Hình 2.11: Mel Cepstrum

Quan sát hình trên chúng ta có thể thấy được độ suy giảm biên độ ở các tần số cao hơn.

2.5 Phân lớp âm thanh

2.5.1 Giới thiệu về phân lớp âm thanh

Việc phân âm thanh thành các lớp cụ thể có ý nghĩa rất quan trọng. Do với mỗi lớp âm thanh khác nhau có các giá trị đặc trưng khác nhau (ví dụ, lớp giọng nói có giá trị đặc trưng ZR lớn hơn so với âm nhạc). Các giá trị đặc trưng khác nhau của mỗi lớp âm thanh của mỗi lớp âm thanh sẽ giúp cho việc chỉ mục và tìm kiếm các loại âm thanh khác nhau được thuận tiện và nhanh chóng hơn. Mặt khác, việc phân lớp âm thanh khác nhau có ý nghĩa tương ứng với mỗi ứng dụng trên thực tế khác nhau (ví dụ, ứng dụng trong lĩnh vực nhận dạng giọng nói, chuyển giọng nói thành văn bản... từ lớp âm thanh là giọng nói, hoặc người ta có thể dựa vào giai điệu để tìm bài hát tương ứng từ lớp âm thanh là âm nhạc). Tóm lại, việc phân lớp âm thanh có các ý nghĩa sau đây:

1. Các lớp âm thanh khác nhau đòi hỏi các tiến trình xử lý và các kỹ thuật truy tìm, chỉ mục khác nhau.

2. Các lớp âm thanh khác nhau có ý nghĩa đối với các ứng dụng khác nhau.

3. Việc phân lớp âm thanh rất hữu ích đối với một số ứng dụng.

4. Không gian tìm kiếm sau khi phân lớp sẽ giảm đáng kể, phục vụ cho việc phân thành các lớp âm thanh riêng biệt hơn hoặc để xử lý, truy tìm âm thanh.

Ngày nay, một số phương pháp phân lớp âm nhạc, giọng nói và các âm thanh khác được đề xuất như: Saunders sử dụng phương pháp tốc độ vượt 0 (ZCR) trung bình và năng lượng thời gian ngắn (Short Time Energy) làm các

đặc trưng, và áp dụng một phương pháp giới hạn đơn giản để phân biệt giọng nói và âm nhạc từ đài phát thanh. Scheirer và các đồng nghiệp thì sử dụng mười ba đặc trưng đồng thời để phân lớp âm thanh. Kimber đưa ra một cách tiếp cận về việc phân đoạn âm thanh, phần lớn được ứng dụng trong việc phân đoạn âm thanh trong bản ghi âm các hội nghị. Zang phân các lớp âm thanh bằng phương pháp di truyền thành hai giai đoạn. Giai đoạn đầu, các tín hiệu âm thanh được phân đoạn và phân thành các lớp thô là giọng nói, âm nhạc, bài hát, giọng nói kèm nhạc nền, tiếng động của môi trường kèm với nhạc nền, sáu loại tiếng động của môi trường và tĩnh lặng (âm câm). Trong giai đoạn hai, sự phân lớp được nâng ở mức cao hơn và được đưa vào từng loại cơ bản. Giọng nói được phân loại bằng giọng của người nam, nữ và trẻ em. Âm nhạc được phân lớp thành các lớp như loại nhạc blue, nhạc jazz, nhạc rock & roll, nhạc kèm lời và nhạc không lời, tùy theo các loại hoặc các nhạc cụ. Âm thanh môi trường được phân lớp thành các lớp theo ngữ nghĩa ví dụ như tiếng vỗ tay, tiếng chuông kêu, tiếng bước chân, tiếng gió bão, tiếng cười, tiếng chim hót...Theo Lu, một phương pháp phân lớp âm thanh gồm hai giai đoạn thô để phân lớp âm thanh thành giọng nói, âm nhạc, tiếng động của môi trường và tĩnh lặng. Còn Xi Shao, Changsheng Xu và Mohan S Kankanhalli đưa ra phương pháp phân lớp âm thanh tự động bằng cách: ban đầu, trích chọn các đặc trưng từ nội dung âm thanh và xây dựng ra các vectơ đặc trưng, sau đó, ứng dụng mạng nơron để tiến hành phân lớp âm thanh, và sử dụng giải thuật di truyền với giải thuật BP đồng thời để huấn luyện mạng. [4]

2.5.2. Đặc điểm chính của phân lớp âm thanh

Ban đầu người ta phân lớp âm thanh thành ba lớp thô cơ bản là tiếng nói, âm nhạc và nhiễu. Vì giọng nói và âm nhạc là mối quan tâm chủ yếu của người dùng nên đa số ứng dụng hiện nay đều tập trung vào nghiên cứu và phát triển dựa trên hai lớp cơ bản này. Sau đây xin trình bày đặc trưng khác biệt chính của lớp âm nhạc và lớp giọng nói theo nghiên cứu của Leung:

Đặc trưng	Giọng nói	Âm nhạc
Băng thông	0 - 10 kHz	0 - 20 kHz
Trọng tâm phổ	thấp	Cao
Tỷ lệ câm	Cao	thấp
Tốc độ vượt 0	biến đổi nhiều	biến đổi ít
Tiếng gõ đều	Không	Thường là có

2.5.3. Kỹ thuật phân lớp âm thanh

Việc phân lớp âm thanh dựa trên cơ sở tính toán các giá trị đặc trưng. Ta xem xét một số cách phân lớp âm thanh như sau:

a. Phân lớp âm thanh theo từng bước

Là phương pháp phân lớp âm thanh theo từng bước lọc dựa vào phân biệt đặc trưng của âm thanh, từ đó xác định lớp của âm thanh đó. Mỗi đặc trưng được sử dụng một cách riêng biệt trong các bước phân lớp khác nhau. Thứ tự trong mỗi đặc trưng khác nhau được sử dụng để phân lớp là rất quan trọng, chúng thường được quyết định dựa trên độ phức tạp tính toán và các khả năng khác nhau của mỗi đặc trưng.

Trình tự phân lớp âm thanh theo từng bước được xác định như sau: [4]

Bước 1. Tính trọng tâm (centroid) của các đoạn âm thanh (Lọc âm nhạc và giọng nói hoặc nhạc sô lô)

Giọng nói và nhạc sô lô có trọng tâm thấp hơn so với âm nhạc. Vậy, nếu dữ liệu âm thanh nhập vào có trọng tâm cao thì đó là lớp âm nhạc. Ngược lại, nó là giọng nói hoặc nhạc sô lô.

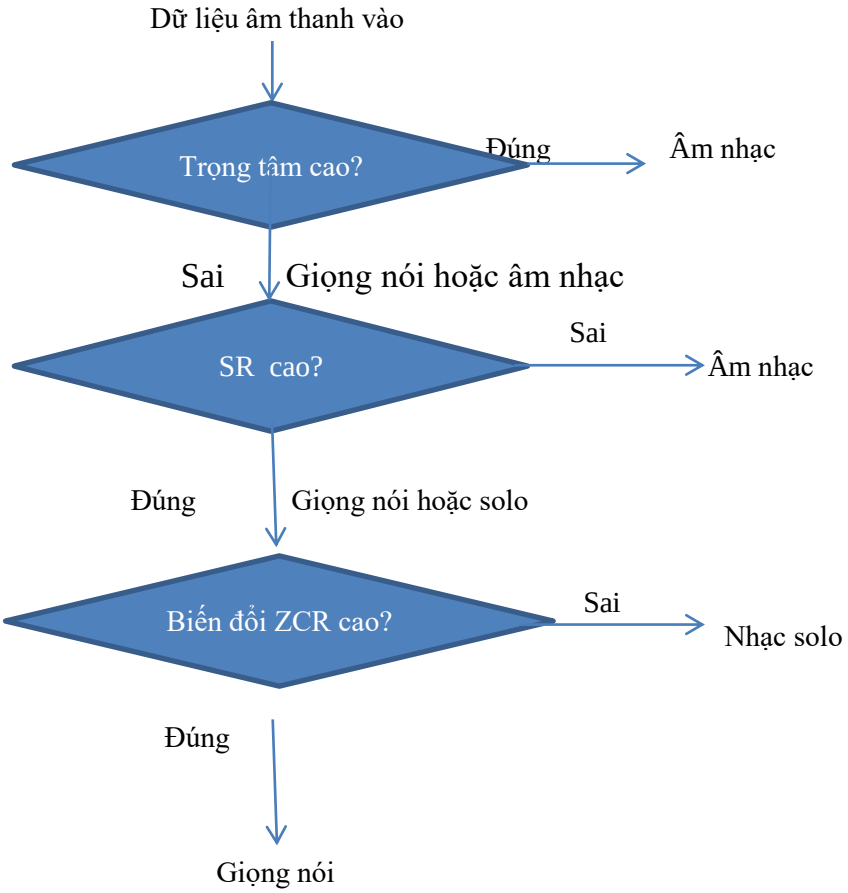
Bước 2. Tính tỷ lệ câm(SR)

(Lọc âm nhạc với giọng nói và nhạc sô lô)

Giọng nói và nhạc sô lô có tỷ lệ câm thấp hơn so với âm nhạc. Vậy, nếu dữ liệu âm thanh nhập vào có tỷ lệ câm cao thì đó là lớp âm nhạc. Ngược lại, nó là giọng nói hoặc hát sô lô.

Bước 3. Tính tỷ lệ vượt qua 0 trung bình (ZCR) (Lọc giọng nói và nhạc sô lô)

Trong quá trình phát âm, ZCR của giọng nói có mức biến đổi lớn hơn nhiều so với âm nhạc. Như vậy, nếu giá trị biến đổi ZCR cao thì nó là giọng nói, ngược lại là nhạc sô lô.



Hình 2.12: Phân lớp âm thanh theo từng bước

b. Phân lớp âm thanh theo vectơ đặc trưng

Phân lớp âm thanh theo vectơ đặc trưng là trích ra những nội dung âm thanh đặc trưng theo cảm quan để xây dựng vectơ đặc trưng. Theo phương pháp phân lớp này, một tập các đặc trưng được sử dụng đồng thời như một vectơ để tính toán chặt chẽ đầu vào của tập huấn luyện.

Để phân lớp các âm thanh khác nhau, với mỗi tín hiệu âm thanh, ta dùng 17 tham số để xây dựng vectơ đặc trưng, bao gồm:

- ❖ Giá trị trung bình của âm lượng (average of the loudness).
- ❖ Độ biến thiên âm lượng (variance of the loudness).
- ❖ Giá trị trung bình của cao độ (average of the pitch).
- ❖ Độ biến thiên cao độ (variance of the pitch).
- ❖ Giá trị trung bình của độ trong (average of the brightness).
- ❖ Độ biến thiên độ trong (variance of the brightness).
- ❖ Giá trị trung bình của băng thông (average of the bandwidth).
- ❖ Độ biến thiên băng thông (variance of the bandwidth).
- ❖ Đạo hàm trung bình của âm lượng (average derivatives of the loudness).
- ❖ Độ biến thiên đạo hàm âm lượng (variance of the derivatives of the loudness).
- ❖ Đạo hàm trung bình của cao độ (average derivatives of the pitch).
- ❖ Độ biến thiên đạo hàm của cao độ (variance of the derivatives of the pitch).
- ❖ Đạo hàm trung bình của độ trong (average derivatives of the brightness).
- ❖ Độ biến thiên đạo hàm của độ trong (variance of the derivatives of

the brightness).

- ❖ Đạo hàm trung bình của băng thông (average derivatives of the bandwidth).
- ❖ Độ biến thiên đạo hàm của băng thông (variance of the derivatives of the bandwidth).
- ❖ Tỷ lệ câm của các khung năng lượng thấp (the radio of low energy frames).

2.6. Một số kỹ thuật phân cụm dữ liệu

Ý tưởng chính là sắp xếp đối tượng/véctơ đặc trưng tương tự vào cùng nhóm/cụm và việc tìm kiếm chỉ thực hiện trong các cụm liên quan

Mỗi cụm được biểu diễn bởi trọng tâm của các véctơ đặc trưng trong cụm

Trong khi truy vấn, ta tính toán độ tương tự giữa câu truy vấn và từng cụm (đại diện bởi véctơ đặc trưng)

Cụm nào có mức độ tương tự lớn hơn ngưỡng cho trước là được chọn

Tiếp theo đối sánh véctơ câu truy vấn với từng véctơ đặc trưng trong cụm và k đối tượng gần nhất là kết quả.

Phần này đề cập đến hai kỹ thuật phân lớp dữ liệu phổ biến là kỹ thuật phân cụm Kmean và kỹ thuật phân lớp dùng giải thuật thời gian động DTW.

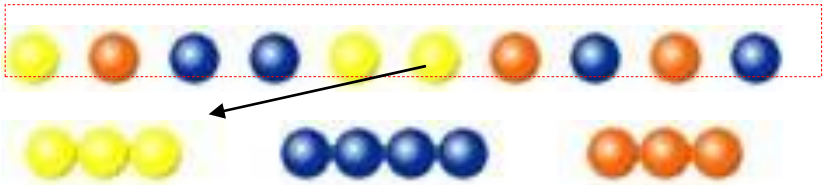
Kỹ thuật phân cụm Kmean là một trong những kỹ thuật phân cụm phổ biến và thành công nhất có sử dụng hệ số cepstral. Những nghiên cứu về sự cảm thụ cho thấy đây là phương thức biến đổi tốt để lấy mẫu các hệ số và nó là quá trình tương đối hiệu quả. Phương thức này gồm 3 tham số: t, k, n với n là số đối tượng, k là số cụm và t là số lần lặp, với k, k<<n.

Thuật toán DTW được ứng dụng để giải quyết việc so sánh giữa hai mẫu tín hiệu có độ dài khác nhau theo thời gian. Như ta biết, kết quả của quá trình phân tích tính hiệu theo phương pháp mã hoá dự báo tuyến tính (Linear Predictive Coding LPC) hay băng lọc (Filter Bank) bao giờ cũng cho ta kết

quả dạng chuỗi các véctơ đặc trưng. Độ dài của chuỗi véc tơ này phụ thuộc vào độ dài của của hai tín hiệu mà ta phân tích. Như vậy, quá trình so sánh hai tín hiệu sẽ tương ứng với quá trình so sánh 2 chuỗi véc tơ đặc trưng của hai tín hiệu. Thuật toán DTW sẽ thực hiện việc so sánh 2 chuỗi véc tơ này theo một số luật sao cho tổng độ lệch giữa hai chuỗi là nhỏ nhất tương ứng với đường đi giữa các cặp véc tơ của hai chuỗi là tối ưu nhất. Việc chọn lựa luật chọn đường đi và giới hạn biên của các đường đi sẽ cho ta kết quả tính toán là nhỏ nhất và hiệu quả nhất.

2.6.1. Tổng quan về phân cụm

Các kỹ thuật phân cụm phân lớp dữ liệu thành hai hoặc nhiều hơn các nhóm dựa vào sự kết hợp nhiều nhân tố. Mục đích của quá trình phân cụm là nhóm dữ liệu tương tự nhau vào một nhóm, trong khi dữ liệu trong các nhóm là khác nhau. Điều này được minh họa trong hình sau



Hình 2.13: Minh họa cho kỹ thuật phân cụm. Phân lớp các quả bóng thành các nhóm có cùng màu

Từ hình trên, chúng ta có thể thấy rằng việc phân cụm là để nhóm dữ liệu hoặc chia dữ liệu lớn thành các phần dữ liệu nhỏ hơn có một số điểm tương tự nhau.

Phương pháp phân cụm làm việc khác với các phương pháp phân loại hoặc thuật toán phân lớp dạng cây. Không có độ ưu tiên về lớp, cả về số lượng cụm hoặc các luật để chỉ định thành các cụm. Phép phân cụm cho phép có nhiều cách gộp nhóm.

Thông thường, phép phân cụm có thể chia thành các kỹ thuật phân cụm có thứ bậc và các kỹ thuật phân cụm không có thứ bậc. Ví dụ về kỹ thuật phân

cụm có thứ bậc là kết nối đơn, kết nối hoàn toàn, kết nối trung bình, giữa, phân khu. Kết nối không có thứ bậc gồm K-mean, K-mean thích ứng, K-medoid, phân cụm mờ. Phép phân cụm K-mean đơn chịu ảnh hưởng của kỹ thuật phân cụm không có thứ bậc, chúng ta sẽ tìm hiểu kỹ về vấn đề này.

2.6.2. Kỹ thuật phân cụm không phân cấp

Phân cụm không phân cấp gồm một dãy đơn điệu tăng về cường độ tức là tăng số cụm để trở thành viên của các cụm lớn hơn. Các cụm mới có dạng liên tục hoặc rời rạc.

Partitioning là một phương pháp như vậy. Kỹ thuật này cho phép nhóm đối tượng lại thông qua quá trình hình thành cụm. Giả sử rằng chúng ta cần có k cụm và quá trình phân chia đối tượng thu được k cụm yêu cầu.

Quá trình phân cụm bắt đầu bằng một giải pháp ban đầu, sau đó phân bố lại theo một số tiêu chí tối ưu. Phương thức phân cụm xây dựng k cụm từ những dữ liệu sau:

- Mỗi cụm tồn tại ít nhất một đối tượng n và mỗi đối tượng k phải thuộc một cụm nào đó. Điều kiện: $k \leq n$
- Các cụm khác nhau không có cùng một đối tượng và xây dựng các cụm trên toàn bộ dữ liệu đã thiết lập.

Cụm thứ k có thể do người dùng quy định hoặc tự động tạo ra để chọn k tốt nhất.

2.6.3. Phương pháp phân cụm K-means

K-mean là một phương pháp phân cụm. Phương pháp này quan sát k cụm trong dữ liệu, và trả lại vector chỉ số của k cụm đã quan sát.

K-mean quan sát trong dữ liệu và tìm cách phân vùng dữ liệu sao cho dữ liệu trong một cụm càng gần nhau càng tốt và so với dữ liệu trong các cụm khác phải càng xa càng tốt. Mỗi cụm được xác định bởi các thành phần của nó và bởi thành phần trung tâm của nó. Thành phần trung tâm của mỗi cụm là thành phần mà có tổng khoảng cách từ các đối tượng trong cụm đến nó là nhỏ

nhất. Cụm trung tâm được tính toán khác nhau với mỗi thước đo khoảng cách, để tổng khoảng cách là nhỏ nhất với mỗi tiêu chuẩn đánh giá.

2.6.4. K-means đầy đủ

Để thực hiện phương thức K-mean đầy đủ ta sử dụng một thuật toán lặp để tính tổng khoảng cách từ mỗi đối tượng tới cụm trung tâm là nhỏ nhất trên toàn bộ cụm. Thuật toán này di chuyển các đối tượng giữa các cụm cho tới khi tổng khoảng cách không thể giảm hơn được nữa. Kết quả là tạo được các cụm có khoảng cách đủ nhỏ và có độ phân cách hợp lý. Độ nhỏ của dữ liệu có thể được chỉ ra bằng việc thay đổi các tham số đầu vào giống với số lượng cụm trung tâm và số lần lặp.

Ý tưởng chính ở đây là tìm cách xác định cụm trung tâm k từ mỗi cụm. Nên lựa chọn điểm trung tâm vì các vị trí khác nhau cho các kết quả khác nhau. Trong điều kiện lý tưởng chúng phải cách xa các điểm khác tối đa khả năng có thể. Mỗi điểm trong dữ liệu được gắn với điểm trung tâm gần nhất. Điểm trung tâm thứ k mới sẽ được tính toán lại từ kết quả phân cụm của bước trước và quá trình nhóm các điểm dữ liệu với các điểm trung tâm gần nhất sẽ được thực hiện lặp đi lặp lại và điều đó sẽ tiếp tục cho tới khi xác định được điểm trung tâm chính.

Phương pháp phân cụm K-mean tìm nhóm có kích thước nhỏ nhất trong tổng bình phương các cụm, chúng ta sử dụng thuật toán sai số bình phương để tính bình phương khoảng cách Euclidean.

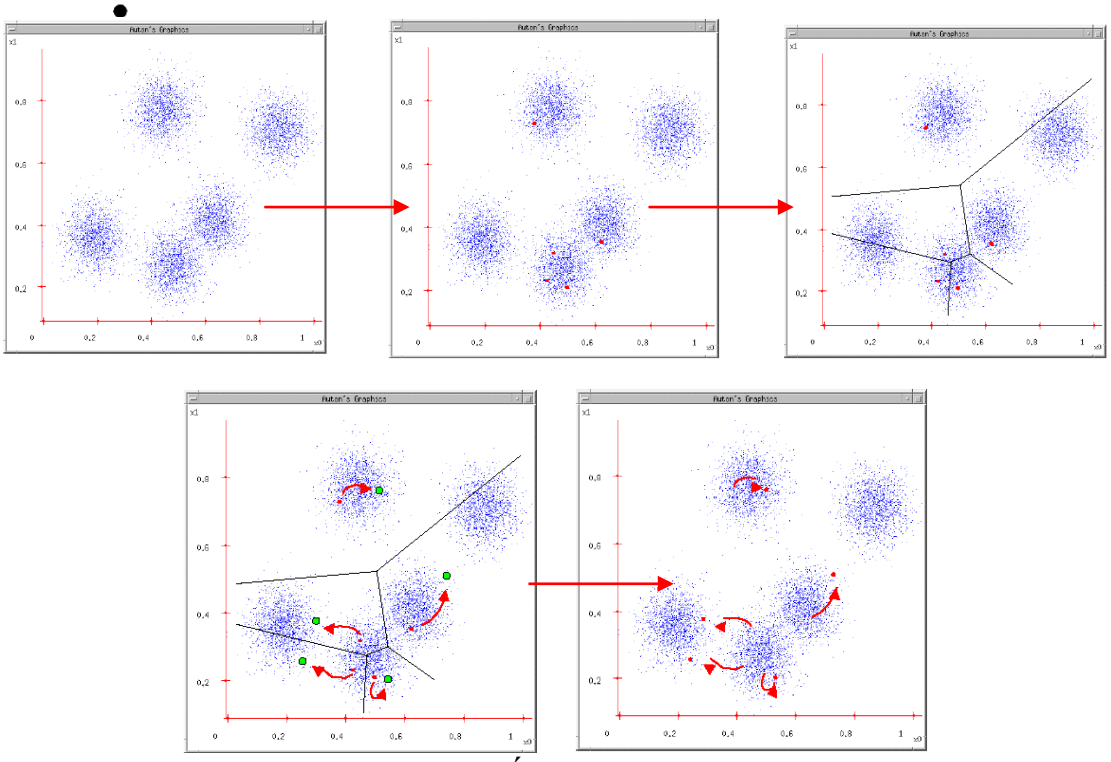
Thuật toán Kmean thực hiện theo các bước sau:

1. Đặt K điểm vào vùng phân cụm các đối tượng. Các điểm này mô tả nhóm trung tâm đầu tiên.
2. Gán mỗi đối tượng vào một nhóm có điểm trung tâm gần nhất.
3. Khi tất cả các đối tượng đã được đưa vào các nhóm, tính toán lại vị trí của K điểm trung tâm.
4. Thực hiện lặp lại bước 2 và 3 cho tới khi bỏ đi được các điểm trung tâm

ở xa. Điều này giúp phân cách các đối tượng thành các nhóm có kích thước nhỏ nhất có thể.

Thủ tục lặp sẽ luôn kết thúc khi điểm trung tâm không thay đổi. Tuy nhiên, cần lưu ý rằng các thuật toán không nhất thiết phải đưa ra những kết quả tối ưu. Hình 2.14 mô tả các bước đã nêu trên. Mỗi bước dưới đây tương ứng với trình tự của biểu đồ.

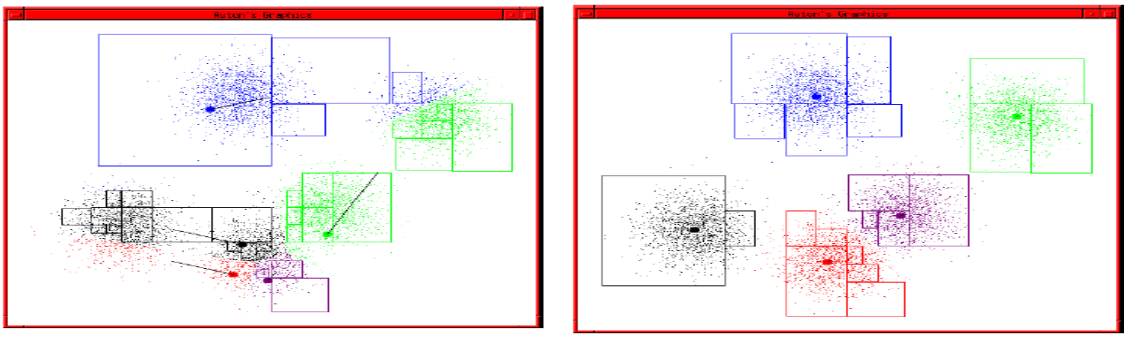
- Chọn số lượng cụm k. Ví dụ k=5
- Tạo ra ngẫu nhiên vị trí trung tâm cụm
- Tại mỗi Centre tìm điểm trung tâm của chính nó
- Và thực hiện bước nhảy
-



Thực hiện lặp lại cho tới khi kết thúc

Hình 2.14: Thủ tục K-mean

Hình trên minh họa phương thức phân cụm K. Chú ý rằng những dữ liệu tương tự được nhóm cùng nhau.



Bắt đầu Kết thúc

Hình 2.15: Phương pháp phân cụm K-mean

2.6.5. Kỹ thuật phân lớp dùng thời gian động DTW

Cho chuỗi âm tiết đầu vào w ω $w_1, w_2, ...w_L$ có độ dài L và có chuỗi vector đặc tính X $x_1, x_2, ...x_T$, nhiệm vụ của hệ thống là phải nhận dạng xem chuỗi âm đầu vào là các ký tự gì và trong quá trình xử lý cần phải giảm thiểu tối đa các sai số quyết định. Mỗi tín hiệu âm tiết đầu vào W_l sẽ được so sánh với các mẫu Y_l. Mỗi Y_l là chuỗi các vector đặc tính của tín hiệu âm tiết W_l. Nhằm tăng khả năng nhận dạng, mỗi âm tiết có một tập hợp các mẫu khác nhau: $Y_{l,1}, ..., Y_{l,M}$. Quá trình quyết định âm tiết phù hợp với một mẫu dựa theo nguyên tắc sau:

$$l^* = \arg\min_l \min_m D(X, Y_{l,m})$$

Như vậy âm tiết W_{l*} là âm tiết phù hợp nhất với mẫu Y_l tìm được.

Khoảng cách D(X,Y) giữa dữ liệu đầu vào và dữ liệu mẫu $Y=y_1....y_s$ có độ dài thời gian khác nhau $S \neq T$ được xác định bằng tổng các khoảng cách cục bộ $d_{ij} = d(x_i, y_j)$ trên cả đường đi của quá trình biến dạng thời gian.

Khoảng cách tích luỹ D_{ij} $D(x_1...x_i, y_1...y_j)$ được xác định theo công thức

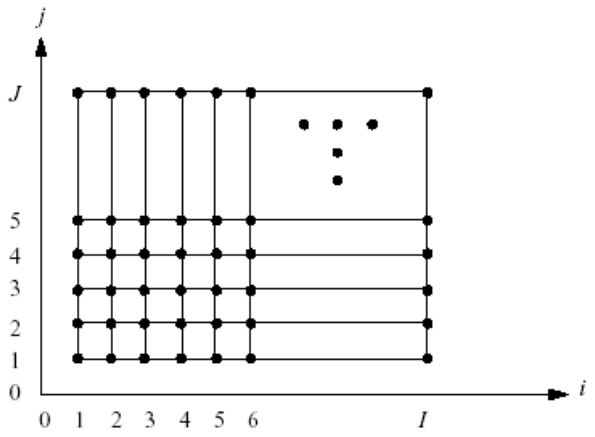
$$\begin{cases} 0 & I=J=0 \\ \min D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1} + d_{ij} & I>0,\ J>0 \\ \infty & \text{Khác} \end{cases}$$

Và khoảng cách tổng D(X,Y)=D_{TS}.

Giả sử cho hai chuỗi vec tơ tương ứng với mẫu tín hiệu là $\vec{a} = a_1, a_2, a_3, ...a_I$ và $\vec{b} = b_1, b_2, b_3, ...b_J$. Cho rằng tín hiệu mẫu $\vec{a}$ có chiều dài

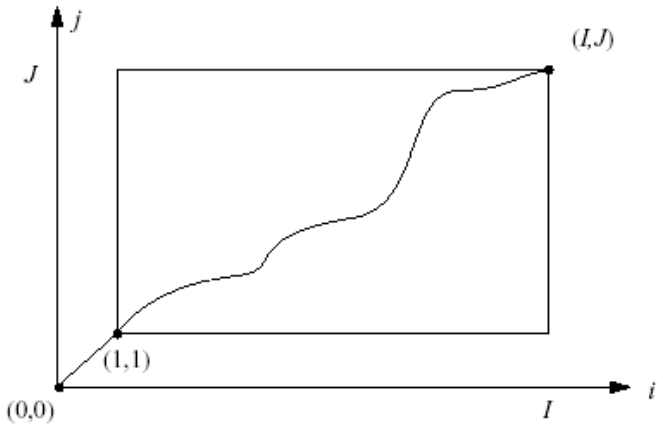
lớn hơn mẫu $\vec{b}$ tức là giá trị ($I > J$). Thuật toán sẽ thực hiện việc tìm đường đi tối ưu của chuỗi b theo chuỗi a (tức là các vị trí khác nhau giữa hai chuỗi theo thời gian) sao cho tổng chênh lệch giữa hai chuỗi vec tơ là nhỏ nhất.

Để thực hiện được điều này thuật toán dùng ma trận lưới các điểm (hình 2.16)



Hình 2.16: Ma trận lưới các điểm

Hai chuỗi véc tơ sẽ tương ứng với hai cạnh của ma trận. Giả sử , véc tơ a theo trục x và véc tơ b theo trục y. Các nút của ma trận tương ứng với khoảng cách tính được của hai chuỗi véc tơ tại các thời điểm thứ i của véc tơ a tương ứng thời điểm thứ j của véc tơ b tương ứng nút *(i,j)*. Như vậy, đường đi tối ưu trong ma trận sẽ có dạng như hình dưới



Hình 2.17: Hình dạng đường đi trong ma trận

Việc xác định đường đi tối ưu trong ma trận lưới được thực hiện sao tổng khoảng cách sai lệch giữa các cặp véc tơ của hai chuỗi là nhỏ nhất. Ký hiệu, *d(i,j)* là độ chênh lệch của hai véc tơ a và b tại thời điểm *i* và *j* tương ứng.

Yêu cầu của thuật toán DTW cho hai chuỗi vec tơ bất kỳ là cùng bắt đầu tại các vị trí *(0,0)* và kết thúc tại vị trí *(I,J)*. Giá trị tại nút *(0,0)* xác định bằng *0*.

Đường đi được xác định theo các cặp nút liên tiếp $(i_{k-1},j_{k-1}) \rightarrow (i_k,j_k)$.

Dùng ký hiệu i_k để biểu diễn chỉ số của véc tơ *a* tại thời điểm *k* và j_k là chỉ số của véc tơ *b* tại thời điểm *k*. Như vậy tổng khoảng cách giữa hai chuỗi véc tơ là :

$$D(i_k, j_k) = D(i_{k-1}, j_{k-1}) + d(i_k, j_k)$$

Việc tìm giá trị min *D(i,j)* theo công thức sau:

$$D^*(i_k, j_k) = \min D(i_{k-1}, j_{k-1}) + d(i_k, j_k)$$

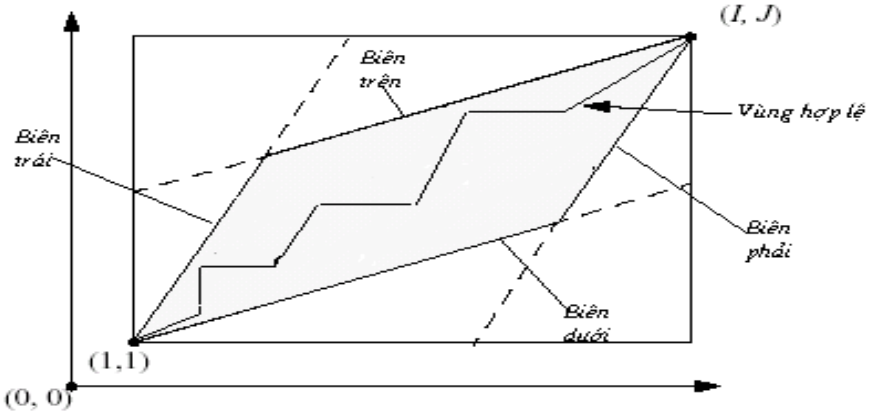
$$\min \left[\sum_{m=0}^{m=k} d(i_m, j_m) \right.$$

Một số bắt buộc của DTW:

- Chỉ số của *i* phải tăng đều tức là : $i_k - i_{k-1} = 1$

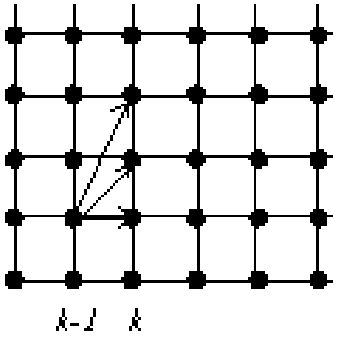
- Chỉ số của *j* tăng theo *i* với điều kiện: $j_k - j_{k-1} \geq 0$

Giới hạn của đường đi không thể tuỳ ý được vì như thế nó sẽ gây ra kết quả sai lệch và làm tăng khối lượng tính toán (nếu xét trên toàn bộ ma trận điểm). Vì vậy, cần phải giới hạn phạm vi của đường đi sao cho việc tính toán giảm và độ chính xác cao. Phạm vi cho đường đi được chọn như hình dưới:



Hình 2.18: Phạm vi cho đường đi

Luật đường đi được lựa chọn theo như hình sau :



Hình 2.19: Luật đường đi

Giả sử vị trí hiện tại đang ở thời điểm i_{k-1} và điểm đi tiếp là i_k. Như vậy

các giá trị j_k có thể là j_k, j_{k+1}, j_{k+2} tương ứng với các mũi tên trên ma trận.

Kết quả:

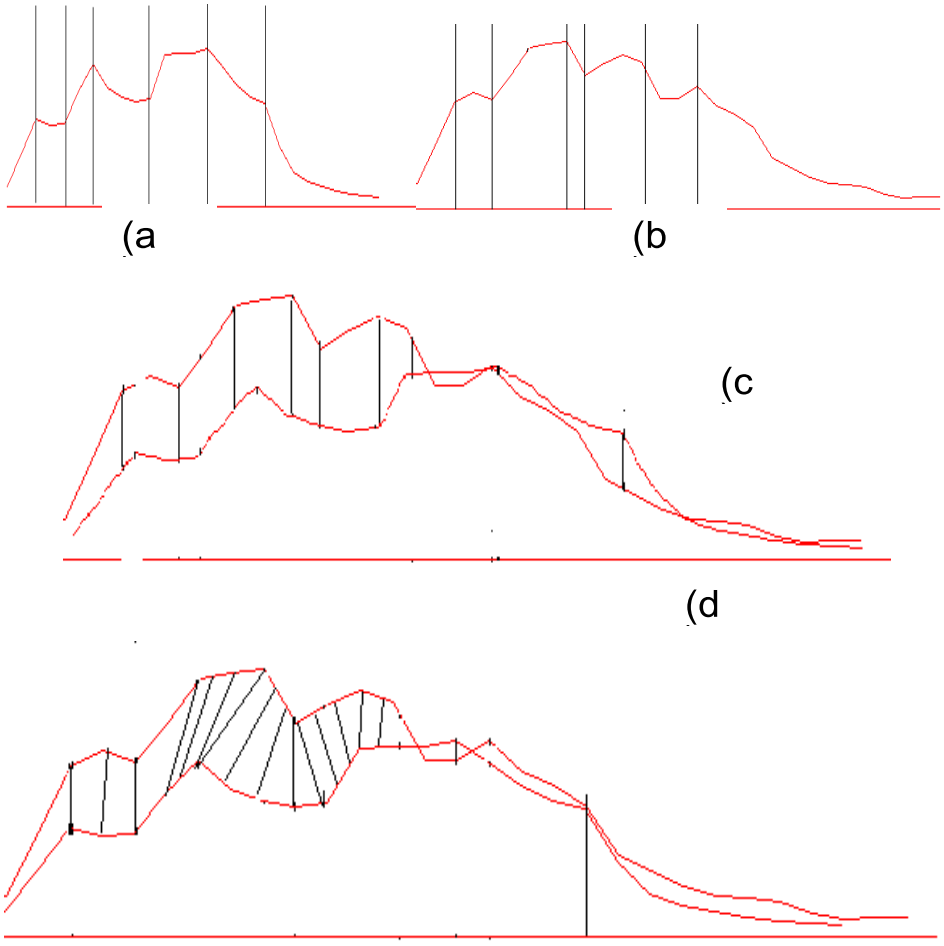
Kết quả được so sánh với phương pháp biến dạng khác là biến dạng tuyến tính theo thời gian hay còn gọi là đồng bộ theo thời gian với thuật toán biến dạng nhưsau:

Chỉ số theo thời gian của tín hiệu *b* liên quan đến chỉ số theo thời gian của tín hiệu *a* là : $j \underset{\omega}{\overset{J}{—}} i$ và kết quả cho như trên hình ở trang sau.

Trên hình vẽ 2.20 thể hiện đường đặc trưng của âm số 2 *("hai")* có độ dài khác nhau *(a)* và *(b)*.

Với phương pháp biến dạng tuyến tính thì giá trị thời gian chỉ số mẫu *b* được giãn đều theo chỉ số thời gian của mẫu tín hiệu *a*. Như vậy, hai khoảng thời gian được kéo dãn bằng nhau song các giá trị thì vẫn còn độ sai lệch lớn do tỷ lệ biến dạng là đều mà không có sự chọn lựa theo giá trị hình *(c)*. Thể hiện việc kéo giãn tại các điểm có đường nối ngang giữa hai tín hiệu.

Còn sử dụng thuật toán DTW ta thấy tỷ lệ biến dạng không đồng đều tại các thời điểm tuỳ thuộc vào giá trị tích luỹ từ trước nên hai mẫu so sánh sẽ có độ chênh lệch là nhỏ nhất hình *(d)*, các đường nối chéo thể hiện sự biến dạng không tuyến tính theo thời gian.



Hình 2.21: Biểu diễn thuật toán biến dạng âm "hai"

(a)(b) Hai tín hiệu có chiều dài khác nhau.

(c) Biến dạng tuyến tính theo thời gian

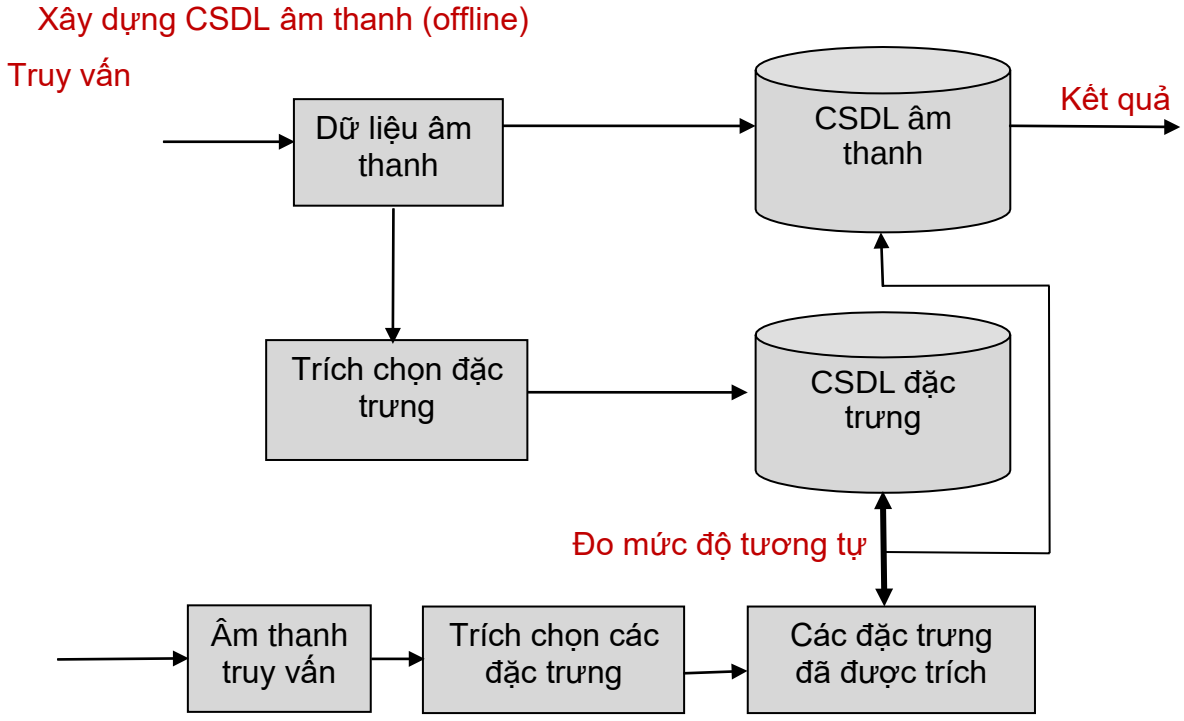
(d) Biến dạng DTW

2.7. Mô hình hệ thống cơ sở dữ liệu âm thanh

Hệ thống cơ sở dự liệu âm thanh được mô tả như hình dưới đây:

Hệ thống bao gồm 2 pha làm việc: Xây dựng cơ sở dự liệu, khai thác (tìm kiếm) âm thanh.

Trong pha xây dựng cơ sở dữ liệu, đầu vào là các tệp âm thanh. Chúng được trích trọn các đặc trưng như: MFCC, năng lượng trung bình, khoảng nặng... Các tệp âm thanh và đặc trùn được lưu trữ đồng thời trong cơ sở dữ liệu. [2]



Hình 2.22: Mô hình hệ thống CSDL âm thanh

Các đặc trưng thông thường được lưu trữ theo một cấu trúc dữ liệu dạng cây nào đó (ví dụ: B+, cây R, ...). Trong các hệ thống thực nghiệm đơn giản, các tệp âm thanh và đặc trưng của chúng được lưu trữ trong tệp hệ thống của hệ điều hành.

Pha tìm kiếm tệp âm thanh được thực hiện online. Tệp âm thanh mẫu để tìm kiếm (tệp âm thanh truy vấn) được trích trọn đặc trưng. Các đặc trưng này được đối sánh với các đặc trưng có sẵn trong cơ sở dữ liệu. Những tệp âm thanh nào trong cơ sở dữ liệu có đặc trưng tượng tự với đặc trưng của tệp âm thanh trong câu truy vấn sẽ là kết quả.

Hệ thống sử dụng độ đo Minkopsky (Euclidean, Mahattan) trong đối sánh tương tự giữa các đặc trưng.

Luận văn này sử dụng mô hình trong hình 2.21 trên đây để xây dựng chương trình thử nghiệm. Việc phát triển được mô tả đầy đủ trong chương 3.

CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH THỬ NGHIỆM HỆ THỐNG TÌM KIẾM ÂM THANH

3.1. Giới thiệu bài toán thử nghiệm

a. Phát biểu bài toán:

Xây dựng hệ thống lưu trữ dữ liệu các giọng nói, bài hát, bài giảng. Hệ thống có khả năng tìm kiếm tệp âm thanh theo nội dung.

Dữ liệu đầu vào: Các file ca nhạc, file ghi âm trong khuôn dạng wav được lưu trữ trong bảng cơ sở dữ liệu.

Các kỹ thuật lưu trữ âm thanh: có thể lưu trữ tệp âm thanh gốc, có thể lưu trữ các tên tệp âm thanh, loại tệp âm thanh, hoặc mô tả các loại tệp đó).

Các kỹ thuật tìm kiếm âm thanh: có nhiều cách, có thể tìm theo tên, theo mô tả, ...

Kết quả: Các tệp âm thanh tương tự với tệp âm thanh ở đầu vào của hệ thống.

b. Lựa chọn thuật toán

Với mục đích cài đặt thực nghiệm một thuật toán trích chọn đặc trưng âm thanh để xây dựng một chương trình tìm kiếm âm thanh theo nội dung, học viên lựa chọn phương pháp trích chọn đặc trưng MFCC vì trong phép biến đổi sử dụng đặc trưng MFCC, tín hiệu âm thanh được lọc theo thang cảm nhận của hệ thống thính giác của con người- một hệ thống nhận dạng âm thanh hoàn hảo. MFCC làm nổi bật các thành phần tín hiệu nhạy cảm với tai người cũng như các đột biến theo thời gian của tín hiệu, có thể giúp phát hiện những thay đổi trong giai điệu của tín hiệu, thể hiện sự vượt trội khi sử dụng để nhận dạng âm thanh theo thời gian, đặc biệt là nhận dạng tiếng nói.

C. Lựa chọn ngôn ngữ lập trình

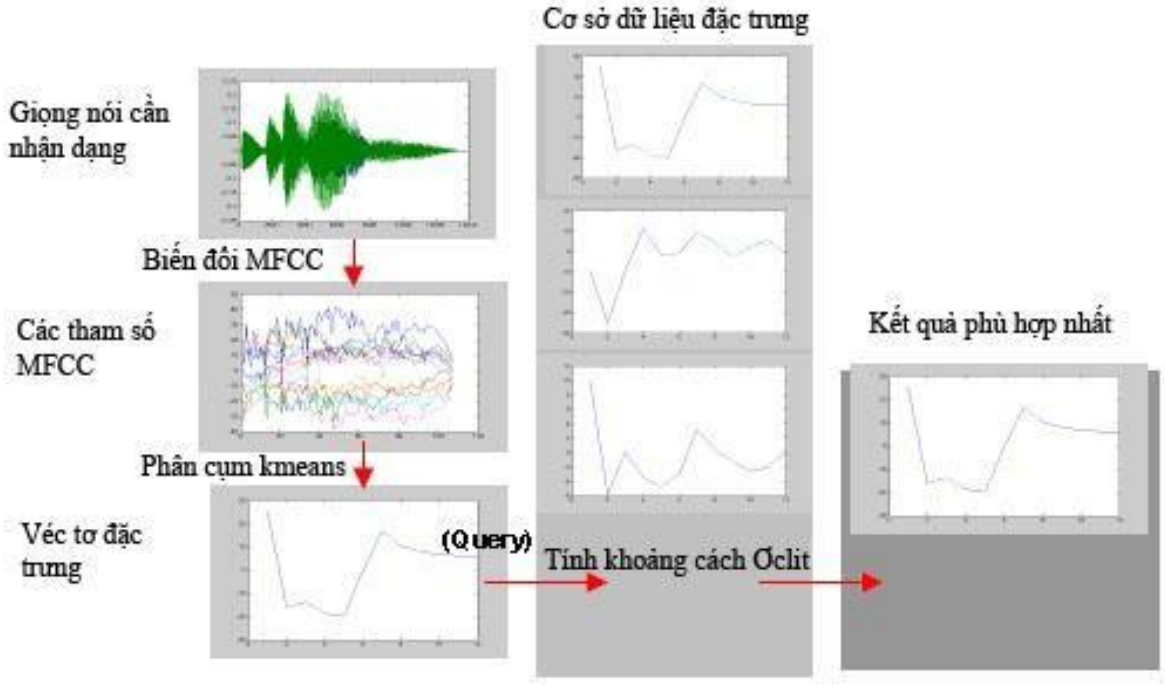
Học viên lựa chọn sử dụng Matlab làm môi trường lập trình vì Matlab là ngôn ngữ lập trình khoa học kỹ thuật hỗ trợ rất mạnh các hàm tính toán

toán học và xử lý tín hiệu số như hàm tính toán biến đổi Furier, biến đổi Cosin rời rạc (DCT), ...là các phép tính sử dụng trong phép phân tích MFCC. Thêm vào đó, cộng đồng người sử dụng MATLAB ngày càng phát triển nhiều module phần mềm, kể cả những mô đun mã nguồn mở để bổ sung thêm nhiều chức năng mới, cũng như các Toolbox (bộ công cụ) với phạm vi chức năng chuyên dụng cho từng chuyên ngành cụ thể như xử lý ảnh, xử lý âm thanh, xử lý video, tính toán song song, tính toán thống kê, mô phỏng cho MATLAB.

3.2. Cài đặt thử nghiệm hệ thống tìm kiếm âm thanh

Hệ thống nhận dạng âm thanh SpeakIden được học viên phát triển trên nền ngôn ngữ lập trình kỹ thuật Matlab phiên bản R2009a, sử dụng một số thư viện mã nguồn mở của tác giả Roger Jang thuộc nhóm phát triển Mirlab và thư viện mã nguồn mở Voicebox của tác giả Mike Brookes, khoa kỹ thuật điện và điện tử, đại học kỹ thuật Hoàng gia, London, UK.

3.2.1. Mô hình hệ thống

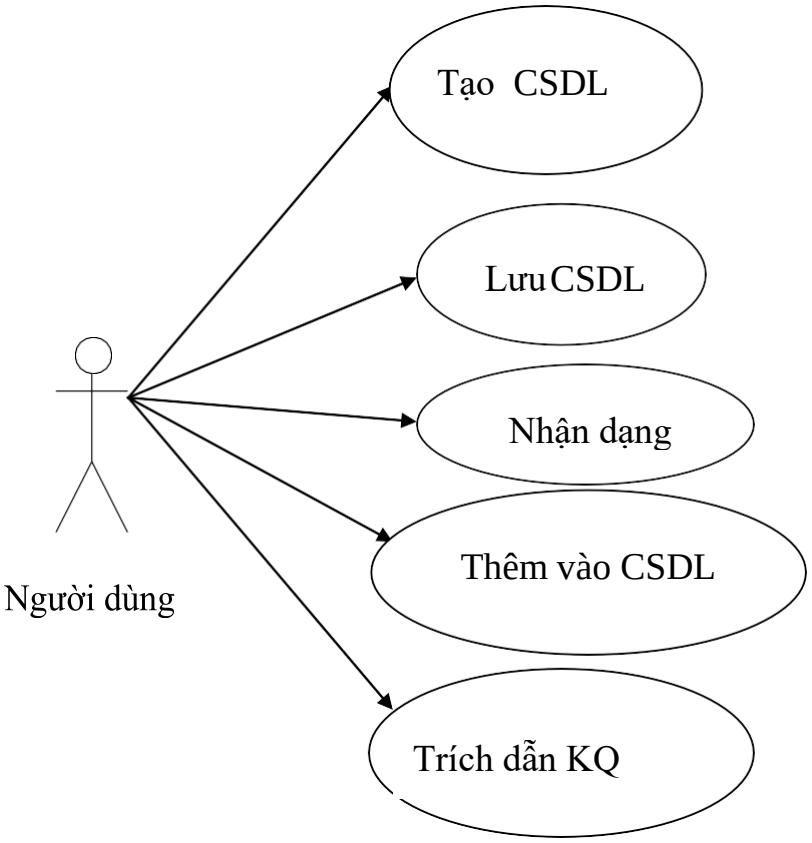


Hình 3.1 : Mô hình hệ thống nhận dạng giọng nói

Hệ thống sử dụng đặc trưng MFCC để nhận dạng âm thanh. Ở bước chuẩn bị, thực hiện trích chọn đặc trưng toàn bộ các tệp âm thanh trong CSDL âm thanh theo phương pháp MFCC để thu được các vector đặc trưng, tạo thành CSDL đặc trưng. Ở bước nhận dạng, tệp âm thanh cần nhận dạng sẽ được trích chọn vector đặc trưng MFCC rồi đối sánh với các vector đặc trưng

trong CSDL đặc trưng để tìm ra vector đặc trưng có độ tương đồng cao nhất bằng cách sử dụng độ đo khoảng cách Ơclit để so sánh khoảng cách giữa 2 vector, từ đó truy lục được tệp âm thanh phù hợp nhất.

Mô hình Use case của hệ thống như sau



:

3.2.2. Luồng dữ liệu trong chương trình và các tham số thực nghiệm

3.2.2.1. Chuẩn bị dữ liệu

Dữ liệu âm thanh gồm giọng đọc của một số học sinh Trường THPT

Trần Tất Văn, các học sinh lớp 11B1, 11B2, 11B3, 10C1, 10C2,10C4 và một số đồng nghiệp được thu âm trong thời gian khoảng 30 giây, tần số lấy mẫu 22050 Hz (hệ thống cho phép tùy chỉnh tần số lấy mẫu), sử dụng 16 bit mã hóa. Dữ liệu âm thanh của mỗi người được tách làm nhiều phần với nội dung đọc hoàn toàn khác nhau, sau đó một phần được đưa vào trích chọn đặc trưng để tạo cơ sở dữ liệu đặc trưng, các phần còn lại dùng để nhận dạng.

3.2.2.2. Xây dựng cơ sở dữ liệu đặc trưng

Các tệp âm thanh được phân khung với kích thước mỗi khung được tính toán động theo tần số lấy mẫu theo công thức :

samples/frame = pow2(floor(log2(0.03*fs)))

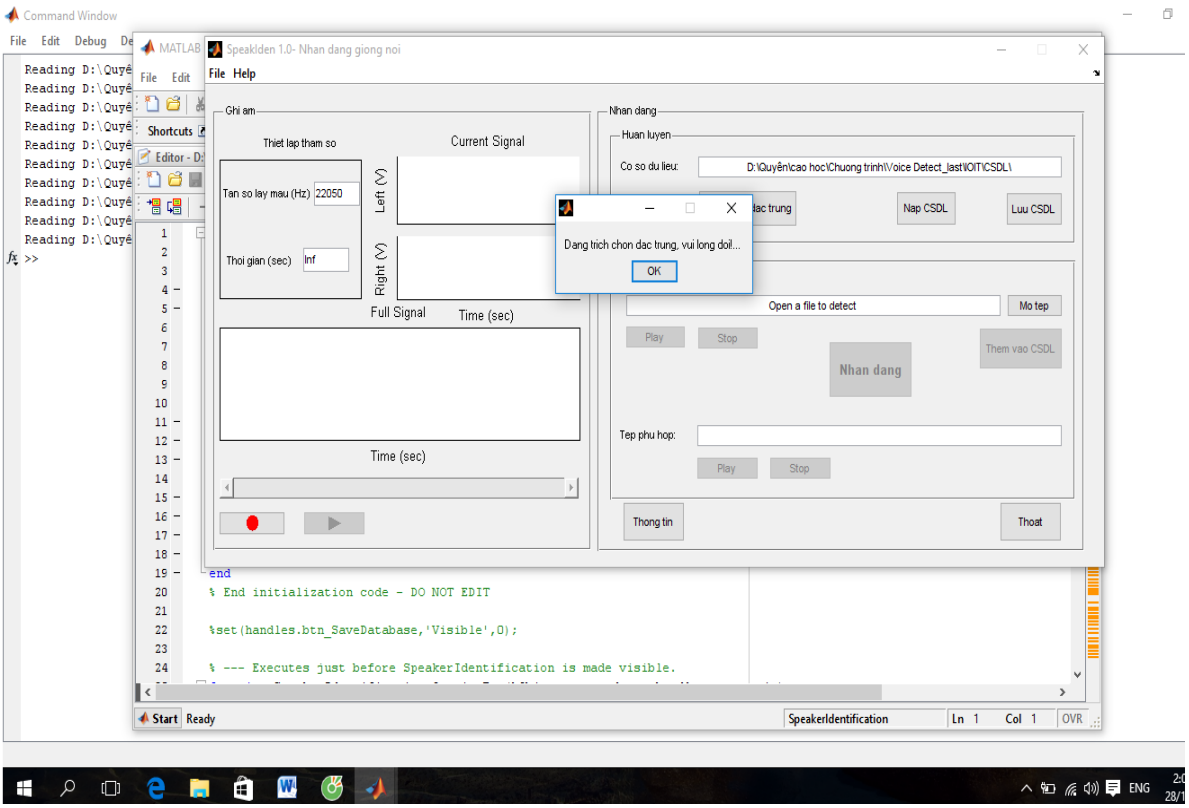
Các khung được lấy chồng phủ lên nhau với độ phủ là một nửa khung. Các khung dữ liệu sau đó được cửa sổ hóa sử dụng cửa sổ Hamming (có thể tùy chỉnh). Tiếp theo, các khung dữ liệu được biến đổi sang miền tần số sử dụng phép biến đổi Fourier nhanh. Sau đó, sử dụng 20 băng lọc mel, lấy 12 hệ số Ceptral để làm nổi bật các tần số cảm thụ của tai người. Kết quả, thu được 12 vector hệ số MFCC, sử dụng thuật toán K-means để tính toán vector MFCC trung bình của 12 vector hệ số MFCC rồi lưu vào cơ sở dữ liệu đặc trưng.

3.2.2.3. Nhận dạng âm thanh

Tệp âm thanh cần nhận dạng sẽ được thực hiện trích chọn đặc trưng theo các bước như trên để thu được vector đặc trưng MFCC. Sau đó vector này được đối sánh với các vector đặc trưng trong CSDL đặc trưng bằng cách sử dụng hàm Ơclit để tính toán khoảng cách giữa 2 vector. Kết quả sau bước này sẽ thu được vector đặc trưng "gần nhất" với vector đặc trưng của tệp âm thanh cần nhận dạng, từ đó sẽ truy lục được tệp âm thanh phù hợp nhất trong cơ sở dữ liệu.

3.2.3. Một số chức năng của chương trình

Hình dưới minh họa giao diện đồ họa dạng hộp thoại của chương trình.



Hình 3.2: Giao diện của phần mềm SoundFinder

Chức năng thu âm trực tiếp:

Cho phép thu âm giọng nói của người dùng, lưu lại dưới dạng tệp .wav. Có thể tùy chỉnh tần số lấy mẫu và thời gian thu âm trên cửa sổ giao diện của chương trình.

Chức năng tạo mới cơ sở dữ liệu đặc trưng:

Tạo cơ sở dữ liệu đặc trưng từ các tệp âm thanh định dạng .wav trên máy tính

Chức năng lưu cơ sở dữ liệu đặc trưng:

Lưu cơ sở dữ liệu đặc trưng dưới định dạng .mat của Matlab

Chức năng tải cơ sở dữ liệu đặc trưng đã có sẵn:

Nạp cơ sở dữ liệu đặc trưng có sẵn trên máy vào bộ nhớ của chương trình

Chức năng thêm mới người nói vào cơ sở dữ liệu:

Thêm đặc trưng của một tệp âm thanh mới thu vào cơ sở dữ liệu đặc trưng đã có sẵn.

Chức năng nhận dạng âm thanh:

Nhận dạng tệp âm thanh trong cơ sở dữ liệu có độ tương đồng cao nhất với các đặc trưng của tệp âm thanh cần nhận dạng.

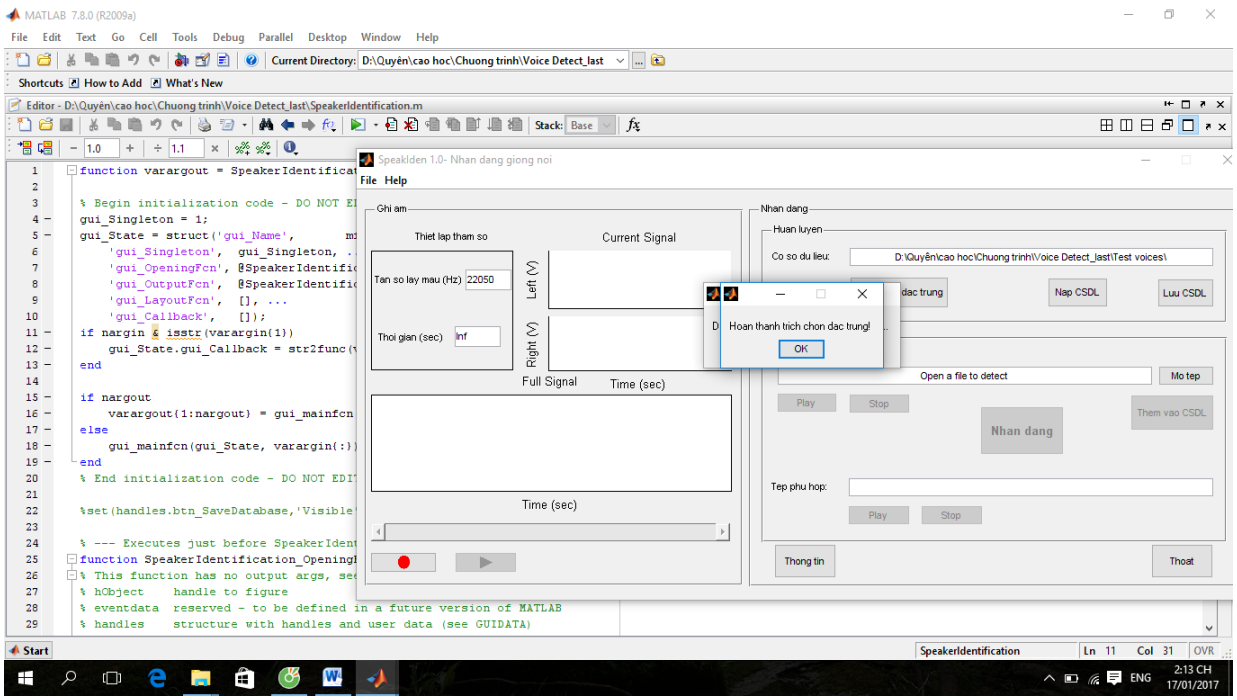
Chức năng trình diễn tệp âm thanh

Xuất tín hiệu âm thanh ra lối ra âm thanh analog của máy tính.

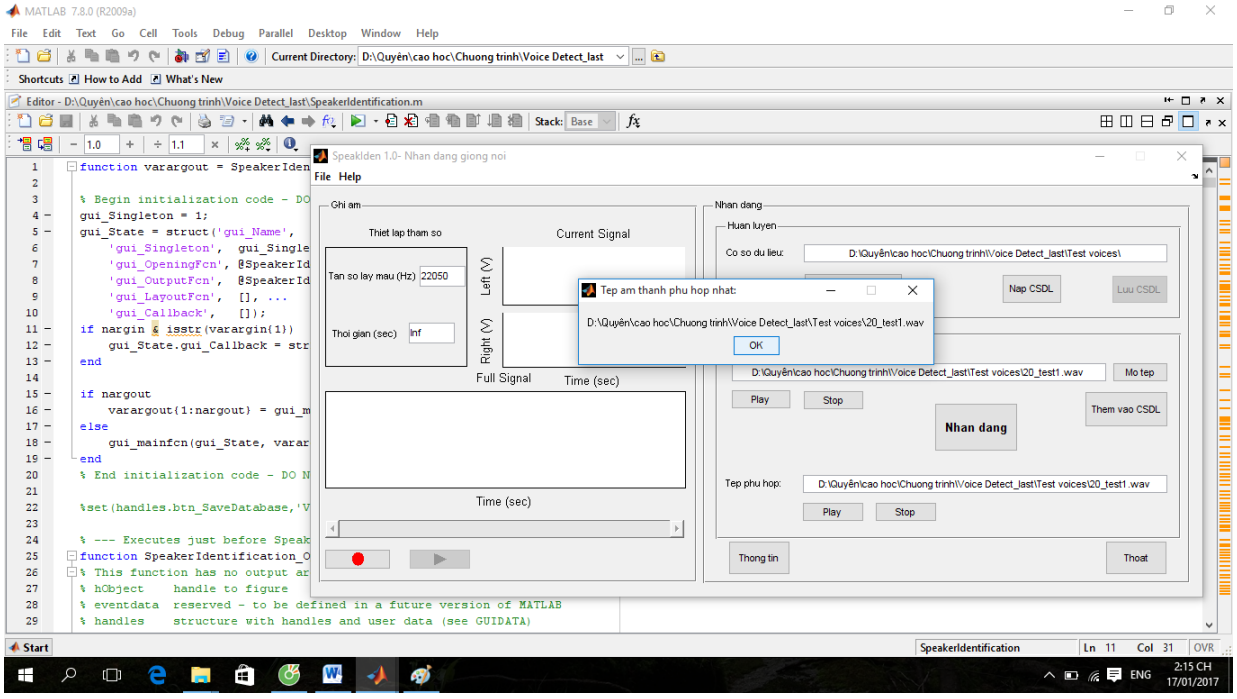
3.2.4. Kết quả thực nghiệm

Dữ liệu âm thanh gồm giọng đọc của một số học sinh Trường THPT Trần Tất Văn, các học sinh lớp 11B1, 11B2, 11B3, 10C1, 10C2,10C4 và một số đồng nghiệp được thu âm trong thời gian khoảng 30 giây, tần số lấy mẫu dạng chính xác đạt khoảng 95%. Với thử nghiệm trên tập dữ liệu âm thanh 22050 Hz (hệ thống cho phép tùy chỉnh tần số lấy mẫu), sử dụng 16 bit mã hóa kết quả tỷ lệ nhận gồm các bài hát, kết quả tìm kiếm chính xác đạt hơn 90%

Kết quả thử nghiệm cho thấy, với các tệp âm thanh được tách ra từ một lần thu âm nhưng nội dung đọc khác nhau hì tỷ lệ nhận dạng đạt được khá cao, cỡ trên 90%. Với các tệp thu âm ở thời điểm và môi trường khác nhau, chưa qua lọc nhiễu, tỷ lệ nhận dạng kém hơn, đạt trên 80%. Kết quả tìm kiếm các tệp bài hát đạt độ chính xác kém hơn so với tìm kiếm giọng nói do trong bài hát không chỉ có giọng hát của các ca sĩ mà còn có âm thanh hòa âm phối khí, với các bài hát khác nhau nhưng có hòa âm phối khí tương đối giống nhau, khả năng tìm kiếm sai là hoàn toàn dễ xảy ra



Trích chọn đặc trưng



Nhận dạng âm thanh trong CSDL

KẾT LUẬN VÀ ĐỀ NGHỊ

Luận văn trên đây đã giới thiệu một số phương pháp tìm kiếm âm thanh theo nội dung, bao gồm tình hình nghiên cứu và hướng phát triển của việc tìm kiếm dữ liệu âm thanh hiện nay, các khái niệm cơ sở và chọn lọc một số phương pháp, giải thuật nghiên cứu đã được chứng minh là hiệu quả thông qua các thí nghiệm trong thời gian gần đây.

Việc tìm kiếm âm thanh theo nội dung nói chung và nhận dạng giọng nói nói riêng là một vấn đề khó, đòi hỏi kết hợp nhiều phương pháp khác nhau, sử dụng nhiều bộ tham số đặc trưng khác nhau. Trong khuôn khổ luận văn mới chỉ cài đặt thử nghiệm hệ thống nhận dạng giọng nói sử dụng một đặc trưng MFCC, chưa kết hợp thêm các đặc trưng khác như tần số cơ bản, formant… Luận văn đã thực hiện được:

- Tìm hiểu các đặc trưng của âm thanh và tiếng nói.
- Tìm hiểu một số phương pháp trích chọn đặc trưng sử dụng MFCC, tần số cơ bản F_0 và Formant.
- Đề xuất một mô hình cho hệ thống nhận dạng tìm kiếm âm thanh. Cài đặt được phần mềm thử nghiệm trên nền Matlab.

Tuy nhiên, luận văn vẫn còn một số hạn chế cần phát triển thêm :

- Bộ dữ liệu thử nghiệm quá nhỏ.
- Chưa nghiên cứu và cài đặt các kỹ thuật chỉ mục cho cơ sở dữ liệu.
- Chưa khảo sát đánh giá được tính hiệu quả về mặt thời gian tìm kiếm cũng như sự ảnh hưởng của các tham số như độ dài tệp nhận dạng, số lượng mẫu âm thanh trong cơ sở dữ liệu âm thanh, số hệ số MFCC và số băng lọc sử dụng…

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Đặng Văn Chuyết, Ngô Minh Dũng "*Khảo sát tính ổn định của một số đặc trưng ngữ âm trong nhận dạng người nói*"

[2] Đặng Văn Đức, CSDL đa phương tiện, Bài giảng cho cao học, Đại học Công nghệ thông tin và truyền thông, Đại học Bách khoa - Hà Nội, Đại học Công nghệ 2005-2014

[3] ETSI, "*ES 202 050 V1.1.5 (2007-01)"* , Chuẩn cho xử lý, truyền dẫn và nhận dạng tiếng nói của tổ chức tiêu chuẩn châu Âu ESTI.

[4] Phạm Văn Sự, Trương Xuân Thành, "*Giáo trình xử lý tiếng nói*", Học viện Công nghệ Bưu chính Viễn thông.

Tiếng Anh

[5] Dalibor Mitrovic et.al., *Features for Content-Based Audio Retrieval,* Vienna Universitv of Technologv. 2010.

[6] DALIBOR MITROVI´ C et al, *"Features for Content-Based Audio Retrieval"*, Vienna University of Technology, dalibor.mitrovic@computer.org.

[7] Dabbala Rajagopal Reddy, "*Speech Recognition*",Academic Press Inc., New York, New York, first edition, 1975.

[8] Guojun Lu, *Multimedia Database Management Systems*, Artech House, Boston – London,1999

[9] Liu Z. and Q.Huang, Content-based indexing and retrieval by example in audio, in *ICME* 2000.

[10] Subrahmanian V.S., *Principles of Multimedia Database Systems,* Morgan Kaufmann Publishers, Inc., California, 1998.

[11] Jyh-Shing Roger Jang, "Speech and Audio Processing Toolbox", available from the link at the author's homepage at "http://mirlab.org/jang".

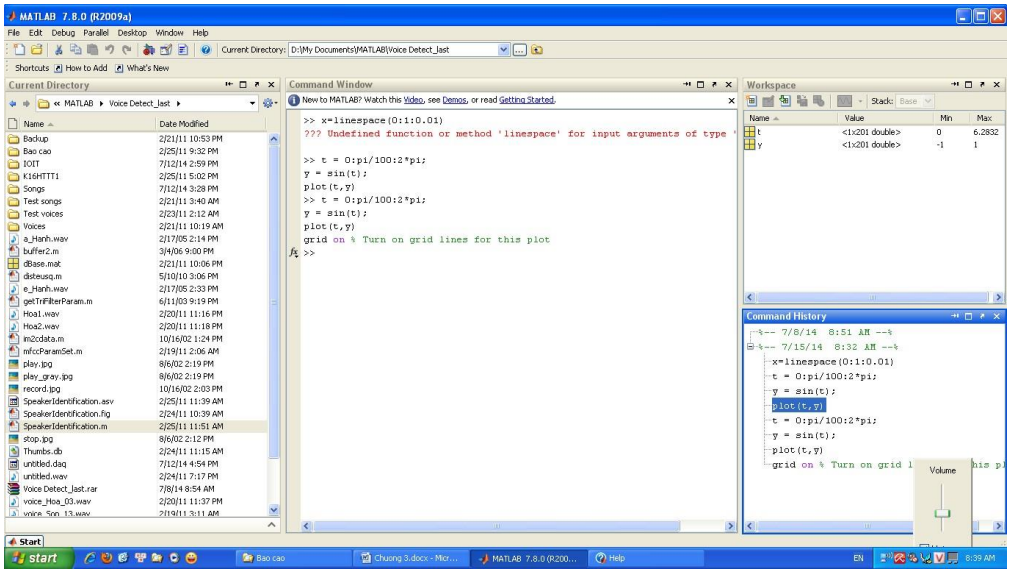
[12] VOICEBOX- toolbox for speech processing. Home page: http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html.

[13] Z.Liu and Q.Huang, "*Content-based indexing and retrieval by example in audio*," in ICME 2000, 2000.

PHỤ LỤC A

Sơ lược về MATLAB

MATLAB là ngôn ngữ lập trình khoa học kỹ thuật nổi tiếng của công ty MathWorks Inc. Ưu điểm nổi bật của MATLAB là khả năng tính toán và biểu diễn đồ hoạ kỹ thuật nhanh chóng, đa dạng và chính xác cao. Thư viện hàm của MATLAB bao gồm rất nhiều chương trình tính toán con; Các chương trình con này giúp người sử dụng giải quyết nhiều loại bài toán khác nhau, đặc biệt là các bài toán về ma trận, số phức, hệ phương trình tuyến tính cũng như phi tuyến. MATLAB cũng cho phép xử lý dữ liệu và biểu diễn đồ hoạ trong không gian 2D và 3D với nhiều dạng đồ thị thích hợp, giúp người sử dụng có thể trình bày kết quả tính toán một cách trực quan và thuyết phục hơn.



Hình 3.3: Cửa sổ giao diện của Matlab

Giao diện của Matlab gồm 4 của sổ giao diện sau:

- Cửa sổ Command Window: cửa sổ chính của MATLAB, tại đây ta thực hiện toàn bộ việc nhập lệnh và nhận kết quả tính toán. Dấu >> là dấu đợi lệnh, sau khi nhận lệnh và kết thúc bằng động tác nhấn phím Enter, MATLAB sẽ xử lí lệnh và xuất hiện kết quả ở dòng dưới.

- Command History: tất cả các lệnh đã sử dụng trong Command Window được lưu trữ và hiển thị tại đây. Có thể thực hiện lệnh cũ bằng cách nhắp đúp chuột vào lệnh đó. Cũng có thể cắt dán, sao chép, xóa cả nhóm lệnh hoặc từng lệnh riêng rẽ.
- Workspace Browser: là một vùng nhớ động trong vùng nhớ của chương trình tự động hình thành khi MATLAB được khởi động và xóa khi thoát MATLAB. Workspace lưu giữ các biến khi ta sử dụng MATLAB. Tất cả các biến trong MATLAB đều được hiển thị tại cửa sổ Workspace Browser với các thông tin về tên biến, giá trị, kích cỡ Byte và loại dữ liệu.

- Current Directory: Nhờ cửa sổ này người sự dụng có thể nhanh chóng nhận biết các thư mục con và các tập tin (file) đang có trong thư mục hiện hành. Các thao tác mở file, lưu file, tìm M-file để thực thi…có mức ưu tiên cao nhất trong thư mục hiện hành.

Lập trình trên Command Window

Với các bài toán đơn giản, chỉ cần dùng ít câu lệnh MATLAB, ta giải bằng cách nhập từng lệnh tại cửa sổ Command window.

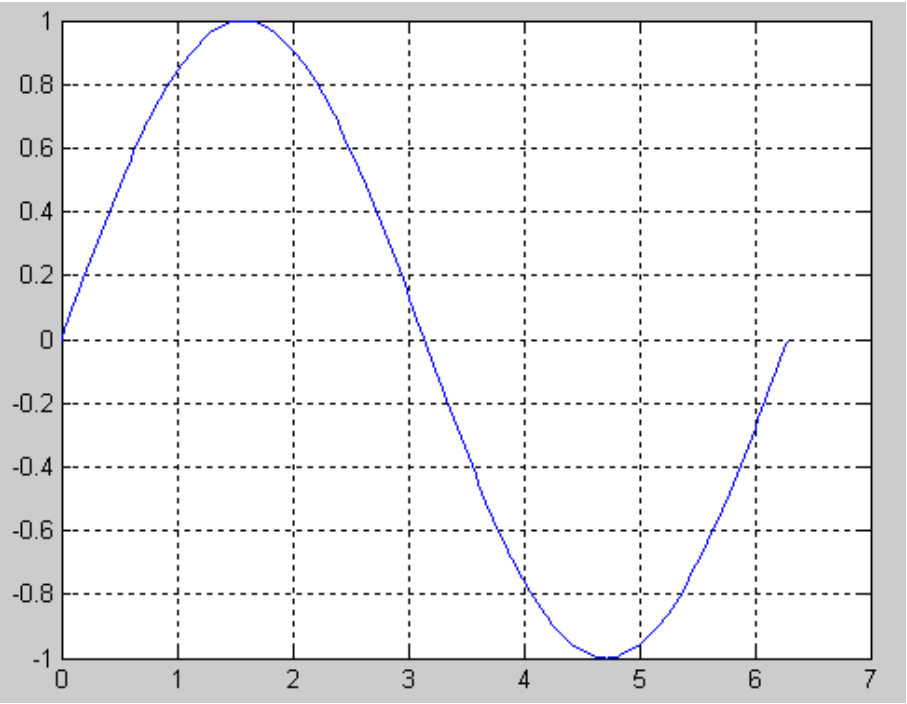
* Một số lưu ý khi nhập lệnh:

- MATLAB luôn hiển thị kết quả của câu lệnh trên màn hình. Nếu muốn MATLAB không hiển thị kết quả thì cuối câu lệnh ta đặt thêm dấu chấm phẩy (;).
- Nhiều câu lệnh có thể đặt chung trên một dòng nhưng bắt buộc phải phân cách nhau bởi dấu phẩy (,) hoặc chấm phẩy (;). Không cho phép phân cách các lệnh bằng khoảng trống. Nếu cuối lệnh nào có dấu phẩy thì MATLAB hiển thị kết quả, còn dấu chấm phẩy thì không hiển thị kết quả.

Ví dụ: Vẽ đồ thị hàm số: ta nhập các dòng lệnh sau vào cửa sổ Command Window

```
x = 0:pi/100:2*pi;
y = sin(x);
plot(x,y)
```

: Kết quả: MatLab sẽ vẽ đồ thị hàm số sin trên một cửa sổ vẽ như sau



Hình 3.4. Đồ thị hàm số sin

Lập trình M-File

Với các bài toán phức tạp, nhiều dòng lệnh, thay vì nhập và thực thi ta có thể soạn và lưu tất cả các từng câu lệnh tại cửa sổ Command window,

câu lệnh vào một file, sau đó chỉ cần gọi file để thực thi toàn bộ các câu lệnh của chương trình. Trong MATLAB, M-file là các file chương trình được soạn thảo và lưu ở dạng văn bản. Có hai loại M-file là Script file (file lệnh) và Function file (file hàm). Cả hai đều có phần tên mở rộng là ".m ". MATLAB có rất nhiều M-file chuẩn được xây dựng sẵn. Người dùng cũng có thể tạo các M-file mới tuỳ theo nhu cầu sử dụng.

Lập trình dạng SCRIPT FILE

Mở cửa sổ soạn thảo M-file:

Cách 1: Trong command window gõ lệnh edit Cách 2: Vào menu File >New >M-File

Cách 3: Nhắp chuột vào icon

Lưu: Vào menu File > Save > đặt tên tập tin > nhắp nút save.

Tập tin Scrift file có phần mở rộng là ".m", và được lưu vào thư mục hiện hành. Nếu không có sự lựa chọn khác thì thư mục hiện hành được mặc định là thư mục work của MATLAB. Tên tập tin phải bắt dầu bằng ký tự chữ, không có khoảng trống giữa các ký tự (giống như quy định về tên biến).

Gọi thực hiện SCRIPT FILE:

- Cách 1: Trong cửa sổ soạn thảo nhắp chuột vào nút run trên thanh toolbar.

- Cách 2: Trở về màn hình Command window và gõ tên file (không có phần mở rộng ".m"), sau đó nhấn Enter để thực thi.

Lưu ý là dù gọi thực hiện theo cách 1 hay cách 2 thì MATLAB cũng đều xuất kết quả tính toán tại cửa sổ Command Window.

Mở một M-file đang có để xem lại hay chỉnh sửa:

- Cách 1: Trong cửa sổ Editor hoặc Command window, vào menu File >open >…

- Cách 2: Vào cửa sổ Workspace, nhắp đúp chuột vào tên M-file cần mở.

- Cách 3: Tại Command window, gõ lệnh edit ('đường dẫn\tên file')

Ví dụ: Tính tổng của n số tự nhiên đầu tiên: S = 1+2+3+…+n

Ta mở một M- file mới rồi soạn thảo như sau

```
n=input('Nhap so so hang can tinh tong n = ');
k=0; S=0; %gia tri ban dau cua tong s
while (k<=n)    %k?t thúc vòng l?p khi không còn
th?a ?ki?n
S=S+k;
k=k+1;
end
fprintf('Tong so %d so tu nhien dau tien la %d'
,n,S)
```

Sau đó, lưu file với tên là "vd_tongN.m"

Khi chạy file, có thể gọi tên file trong cửa sổ Command Window bằng cách gõ lệnh

>> vd_tongN.m

rồi ấn Enter để chạy, hoặc có thể mở file trên rồi ấn nút Run trên thanh công cụ của trình soạn thảo để chạy file. Kết quả thực thi chương trình luôn luôn

hiện ra ở cửa sổ Command Window như sau

```
>> vd_tongN
Nhap so so hang can tinh tong n = 10
Tong so 10 so tu nhien dau tien la 55>>
```

Lập trình dạng FUNCTION FILE

Tương tự như trong toán học, các hàm (function) trong MATLAB sẽ

nhận vào giá trị của các đối số và trả về giá trị tương ứng của hàm. Trình tự tạo và thực thi một file hàm bao gồm các bước như sau:

Mở cửa sổ Editor: Thực hiện tương tự như Scrift file

Soạn thảo: Cấu trúc chuẩn của một hàm:

function [danh sách tham số ra] = tên hàm (danh sách tham số vào)

Lưu: Như cách lưu của Scrift file. Khi lưu hàm, MATLAB sẽ lấy tên hàm làm tên file, người lập trình không nên sửa lại tên này để tránh lẫn lộn khi gọi thực hiện hàm.

Đặc điểm của hàm:

- Các hàm chỉ thông tin với MATLAB thông qua các biến truyền vào cho nó và các biến ra mà nó tạo thành, các biến trung gian ở bên trong hàm thì không tương tác với môi trường MATLAB.
- Các hàm có thể sử dụng chung các biến với hàm khác hay với môi trường MATLAB nếu các biến được khai báo là biến toàn cục. Để có thể truy cập được các biến bên trong một hàm thì các biến đó phải được khai báo là biến toàn cục trong mỗi hàm sử dụng nó.
- Một M-file có thể chứa nhiều hàm. Hàm chính (main function)

trong M-file này phải được đặt tên trùng với tên của M-file. Các hàm khác được khai báo thông qua câu lệnh function được viết sau hàm đầu tiên. Các hàm con (local function) chỉ được sử dụng bởi hàm chính, tức là ngoài hàm chính ra thì không có hàm nào khác có thể gọi được chúng. Tính năng này cung cấp một giải pháp hữu hiệu để giải quyết từng phần của hàm chính một cách riêng rẽ, tạo thuận lợi cho việc lập một file hàm duy nhất để giải bài toán phức tạp.

Ví dụ: Viết hàm giải pt bậc 2: ax2+bx+c=0

Ta tạo mới một M-file rồi nhập các câu lệnh sau:

```
function [x1,x2]=vd_gptb2(a,b,c)
if  nargin<3
error('Vui long nhap du 3 he so cua phuong
```

```
trinh')
    elseif a==0
    x1=-c/b;
    x2=[ ];
    else
    D = b^ 2 - 4*a*c;
    x1 = (-b+sqrt(D))/(2*a);
    x2 = (-b-sqrt(D))/(2*a);
    end
```

Sau đó,

"vd_gptb2.m":

lưu file với tên giống như tên hàm khai báo ở dòng lệnh đầu tiên

=1, b=11, c=8 như sau:

Chạy thử hàm với các đối số truyền vào là các hệ số a

```
>>
[x1,x2]=vd_gptb2(1,11,8)
x1 = -0.7830
x2 = -10.2170
```

PHỤ LỤC B

Một số mã nguồn

Hàm tính toán MFCC

```
function [parameter, yPreEmp]= wave2mfcc(y, fs, FP)

% wave2mfcc: Wave to MFCC (Mel-Frequency Cepstral Cofficient) conversion

%   Usage:

%   parameter = wave2mfcc(y, fs, FP)

%   parameter: MFCC and log energy, plus their delta value if necessary.

%   fs: sampling rate

%    FP: Parameters for deriving the speech features. You can   use mfccParamSet.m
to obtain the parameters.

%

 if nargin<1, selfdemo; return; end
if nargin<2; fs=8000; end
if nargin<3, FP=mfccParamSet(fs); end



y=double(y);    % Convert  to  double
y=y-mean(y);    % Shift to zero mean


% ====== Step 1: pre-emphasis.
yPreEmp = filter([1, -0.95], 1, y);


% ====== Step 2: frame blocking.

framedY = buffer2(yPreEmp, FP.frameSize, FP.overlap);
```

```
filterBankParam = getTriFilterParam(FP.frameSize, fs, FP.tbfNum, 0);        %
Parameters for triangular filter bank
parameter = [];

for i = 1:size(framedY, 2),

   % ====== Step 3: hamming window.

   Wframe  = hamming(FP.frameSize).*framedY(:,i);

   % ====== Step 4: fast fourier transform.
   fftMag = abs(fft(Wframe));
   halfIndex = floor((FP.frameSize+1)/2);
   fftMag = fftMag(1:halfIndex);
   fftMag = interp1(1:halfIndex,fftMag,1:1/FP.alpha:halfIndex)'; % VTLN
   fftMag = [fftMag;zeros(halfIndex-length(fftMag),1)];
   % ====== Step 5: triangular bandpass filter.

% tbfCoef = triBandFilter(fftMag, FP.tbfNum, fstart, fcenter, fstop);
   tbfCoef = triBandFilter(fftMag, FP.tbfNum, filterBankParam);
   % ====== Step 6: cosine transform. (Using DCT to get L order mel-scale-
cepstrum parameters.)

   mfcc = melCepstrum(FP.cepsNum, FP.tbfNum, tbfCoef);
   parameter = [parameter mfcc'];
end
% ====== Add energy
if (FP.useEnergy==1)
   energy = sum(framedY.^2)/FP.frameSize;
   logEnergy = 10*log10(eps+energy);
```

```
    parameter = [parameter; logEnergy];
end
% ====== Compute delta energy and delta cepstrum

% with delta is better for telephone digit recognition HMM
if (FP.useDelta>=1)
    deltaWindow = 2;

    paraDelta = deltaFunction(deltaWindow, parameter);
    parameter = [parameter; paraDelta];
end

if (FP.useDelta==2)

    paraDeltaDelta = deltaFunction(deltaWindow, paraDelta);
    parameter = [parameter; paraDeltaDelta];
end
% ====== Subfunction ======

% === Self demo
function selfdemo
waveFile='what_movies_have_you_seen_recently.wav';
[y, fs]=wavread(waveFile);
FP=mfccParamSet(fs);
FP.useDelta=0;
mfcc0= feval(mfilename, y, fs, FP);

fprintf('No. of extracted frames = %d\n', size(mfcc0, 2));

subplot(3,1,1); surf(mfcc0); box on; axis tight; title(sprintf('MFCC of "%s"',
waveFile));

FP.useDelta=1;
```

```
mfcc1=feval(mfilename, y, fs, FP);
subplot(3,1,2); surf(mfcc1); box on; axis tight; title(sprintf('MFCC of "%s"', waveFile));
FP.useDelta=2;
mfcc2=feval(mfilename, y, fs, FP);
subplot(3,1,3); surf(mfcc2); box on; axis tight; title(sprintf('MFCC of "%s"', waveFile));


% === Triangular band-pass filters
function tbfCoef = triBandFilter(fftMag, P, filterBankParam)
fstart=filterBankParam(1,:);
fcenter=filterBankParam(2,:);
fstop=filterBankParam(3,:);
% Triangular bandpass filter.
for i=1:P
  for j = fstart(i):fcenter(i),
    filtmag(j) = (j-fstart(i))/(fcenter(i)-fstart(i));
  end
  for j = fcenter(i)+1:fstop(i),
    filtmag(j) = 1-(j-fcenter(i))/(fstop(i)-fcenter(i));
  end
  tbfCoef(i) = sum(fftMag(fstart(i):fstop(i)).*filtmag(fstart(i):fstop(i))');
end
tbfCoef=log(eps+tbfCoef.^2);


% === TBF coefficients to MFCC
```

```
function mfcc = melCepstrum(L, P, tbfCoef)

% DCT to find MFCC
for i = 1:L
  coef = cos((pi/P)*i*(linspace(1,P,P)-0.5))';
  mfcc(i) = sum(coef.*tbfCoef');
end

 % === Delta function

function parameter = deltaFunction(deltaWindow,parameter)

% compute delta cepstrum and delta log energy.
rows  = size(parameter,1);
cols  = size(parameter,2);

%temp  = [zeros(rows,deltaWindow) parameter zeros(rows,deltaWindow)];

temp = [parameter(:,1)*ones(1,deltaWindow) parameter
parameter(:,end)*ones(1,deltaWindow)];

temp2 = zeros(rows,cols);

denominator = sum([1:deltaWindow].^2)*2;
for i = 1+deltaWindow : cols+deltaWindow,
  subtrahend = 0;

  minuend    = 0;

  for j = 1 : deltaWindow,

    subtrahend = subtrahend + temp(:,i+j)*j;
    minuend = minuend + temp(:,i-j)*(-j);
  end;

  temp2(:,i-deltaWindow) = (subtrahend + minuend)/denominator;
end;parameter = temp2;
```