BỘ GIÁO DỤC VÀ ĐÀO TẠO BỘ QUỐC PHONG

HỌC VIỆN KỸ THUẬT QUÂN SỰ

ĐÀO SỸ NHIÊN

SỬ DỤNG ĐIỂM CẮT ZERO TÌM HIỂU ĐẶC TRƯNG CỦA MỘT SỐ PHỤ ÂM TIẾNG VIỆT PHỤC VỤ CHO BÀI TOÁN NHẬN DẠNG

Chuyên ngành: Khoa học máy tính

LUẬN VĂN THẠC SĨ KỸ THUẬT

Hà Nội, năm 2011

BỘ GIÁO DỤC VÀ ĐÀO TẠO BỘ QUỐC PHONG

HỌC VIỆN KỸ THUẬT QUÂN SỰ

ĐÀO SỸ NHIÊN

SỬ DỤNG ĐIỂM CẮT ZERO TÌM HIỂU ĐẶC TRƯNG CỦA MỘT SỐ PHỤ ÂM TIẾNG VIỆT PHỤC VỤ CHO BÀI TOÁN NHẬN DẠNG

Chuyên ngành: Khoa học máy tính

Mã số: 60 48 01

LUẬN VĂN THẠC SĨ KỸ THUẬT

Hà Nội, năm 2011

C□NG TR□NH Ă□□C HOÀN THÀNH T□I

HỌC VIỆN KỸ THUẬT QUÂN SỰ

Cán bộ h- ớng dẫn chính: PGS.TS Nguyễn Văn Xuất

Cán bộ chấm phản biện 1:.....

Cán bộ chấm phản biện 2:.....

Luận văn thạc sĩ đ- ợc bảo vệ tại:

H□I Ă□NG CH□M LU□N V□N TH□C S□

H□C VI□N K□ THU□T QUÂN S□

HỌC VIỆN KỸ THUẬT QUÂN SỰ
PHÒNG SAU ĐẠI HỌC

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

Hà Nội, ngày tháng năm 2011

NHIỆM VỤ LUẬN VĂN THẠC SĨ

Họ và tên học viên: **ĐÀO SỸ NHIÊN** Giới tính: Nam

Ngày, tháng, năm sinh: 09-8-1979 Nơi sinh: Hoa Lư, Ninh Bình

Chuyên ngành: Khoa học máy tinh Mã số: 60 48 01

I- TÊN ĐỀ TÀI: Sử dụng điểm cắt Zero tìm hiểu đặc trưng của một số phụ âm Tiếng việt phục vụ cho bài toán nhận dạng.

II- NHIỆM VỤ VÀ NỘI DUNG:

- Nghiên cứu về âm thanh và cách số hoá âm thanh, cấu trúc file Wave, cách thức thu âm.

- Nghiên cứu về hệ thống nhận dạng tiếng nói, cách lấy từng đặc trưng của tiếng nói từ đó xác định cách nhận dạng.

- Nghiên cứu về điểm cắt Zero, việc tổ chức chức dữ liệu và ý tưởng của thuật toán nhận dạng dựa vào điểm cắt zero. Áp dụng ngôn ngữ Visual C# trên bộ công cụ Visual Studio 2008 để xây dựng, thiết kế chương trình.

III- NGÀY GIAO NHIỆM VỤ: 16/10/2010

IV- NGÀY HOÀN THÀNH NHIỆM VỤ: 04/05/2011

V- CÁN BỘ HƯỚNG DẪN: PGS.TS NGUYỄN VĂN XUẤT

CÁN BỘ HƯỚNG DẪN
(*Học hàm, học vị, họ tên và chữ ký*)

CHỦ NHIỆM BỘ MÔN
QL CHUYÊN NGÀNH

Nội dung và đề c- ơng luận văn thạc sĩ đã đ- ợc Hội đồng chuyên ngành thông qua.

Ngày tháng năm 2011

TRƯỞNG PHÒNG SĐH

TRƯỞNG KHOA QL NGÀNH

MỤC LỤC

Chương 3

TRÍCH RÚT ĐẶC TRƯNG CỦA TIẾNG NÓI DỰA VÀO DÃY ĐIỂM CẮT ZERO

Chương 4

XÂY DỰNG CHƯƠNG TRÌNH THỰC NGHIỆM

KẾT LUẬN VÀ KIẾN NGHỊ

TÓM TẮT LUẬN VĂN

Họ và tên học viên: **Đào Sỹ Nhiên**

Lớp: Khoa học Máy tính Khoá: K21

Cán bộ hướng dẫn: PGS. TS Nguyễn Văn Xuất

Tên đề tài: Sử dụng điểm cắt Zero tìm hiểu đặc trưng của một số phụ âm Tiếng việt phục vụ cho bài toán nhận dạng.

Tóm tắt:

Nghiên cứu về lớp bài toán nhận dạng tiếng nói, cách trích rút đặc trưng của tiếng nói, cách số hoá âm thanh, cấu trúc file Wave. Nghiên cứu về điểm cắt Zero từ đó tìm hiểu các tập dữ liệu đặc trưng nhằm phục vụ công việc nhận dạng tiếng nói.

Tổ chức dữ liệu dựa vào các đặc tính của điểm cắt Zero có lưu dữ thông tin liên quan đến âm thanh và xây dưng thuật toán nhận dạng dựa vào điểm cắt zero, áp dụng ngôn ngữ Visual C# để xây dựng, thiết kế chương trình nhằm kiểm nghiệm.

DANH MỤC CÁC KÝ HIỆU

ZCR: Zero Crossing

PCM: Pulse Code Modulation

SWC: Spectral Waveform Coding

MBC: Model Based Coding

TWC: Temporal Waveform Coding

RIFF: Resource Interchange File Format

DWF: Digitized Waveform Files

LPC: Linear Predictive Coding

SWC: Spectral Waveform Coding

MFCC: Mel-frequency Cepstral Coefficients

PLP: Packet Level Protocol

DANH MỤC CÁC BẢNG

DANH MỤC HÌNH VẼ

MỞ ĐẦU

Đối với con người, việc nghe, nhất là nghe tiếng mẹ đẻ là một vấn đề khá đơn giản. Còn đối với máy tính, xác định một chuỗi tín hiệu âm thanh là sự phát âm của một từ nào hoàn toàn không đơn giản, khó khăn cũng như việc học nghe ngoại ngữ của chúng ta.

Lĩnh vực nhận dạng tiếng nói đã được nghiên cứu hơn 4 thập kỉ và hiện nay mới chỉ có một số thành công. Có thể kể đến hệ thống nhận dạng tiếng Anh (ví dụ: phần mềm Via Voice của IBM, hệ thống nhận dạng tiếng nói tích hợp của OfficeXP…). Các hệ thống này hoạt động khá tốt (cho độ chính xác khoảng 90 - 95%) nhưng còn xa mới đạt đến mức mơ ước của chúng ta: có một hệ thống có thể nghe chính xác và hiểu hoàn toàn những điều ta nói.

Riêng với tiếng Việt, lĩnh vực nhận dạng tiếng nói còn khá mới mẻ. Chưa hề thấy xuất hiện một phần mềm nhận dạng tiếng Việt hoàn chỉnh trên thị trường. Số công trình nghiên cứu về nhận dạng tiếng nói tiếng Việt được công bố rất hiếm hoi, và kết quả còn hạn chế về bộ từ vựng, độ chính xác…. Tiếng Việt có nhiều đặc tính khác với các ngôn ngữ đã được nghiên cứu nhận dạng nhiều như tiếng Anh, tiếng Pháp. Do đó việc nghiên cứu nhận dạng tiếng Việt là rất cần thiết.

Mục đích của luận văn:

Nghiên cứu về âm thanh, tín hiệu tiếng nói. Sử dụng điểm cắt zero và các công cụ toán học để rút trích các đặc trưng của một số phụ âm tiếng việt với các giọng nói và người nói khác nhau nhằm hỗ trợ cho việc nhận dạng một số phụ âm tiếng việt.

Nội dung của luận văn gồm các phần:

Phần mở đầu dành cho việc giới thiệu tổng quan về mục đích và nội dung của luận văn.

Chương 1: Lớp bài toán nhận dạng tiếng nói.

Chương này giới thiệu tổng quan về lớp các bài toán nhận dạng tiếng nói và tình hình nghiên cứu về lớp các bài toán này.

Chương 2: Số hóa tín hiệu âm thanh và đặc trưng tín hiệu tiếng nói

Chương này giới thiệu về âm thanh, tiếng nói, kỹ thuật số hóa âm thanh, tiếng nói và các phương pháp trích rút đặc trưng tín hiệu tiếng nói.

Chương 3: Trích rút đặc trưng của tiếng nói dựa vào dãy điểm cắt Zero

Chương này giới thiệu đặc điểm của điểm cắt zero đối với âm thanh, đưa ra thuật toán rút trích đặc trưng của phụ âm bằng cách sử dụng điểm cắt zero và áp dụng hệ số tương quan và trình bày thuật toán nhận dạng.

Chương 4: Xây dựng chương trình thực nghiệm.

Chương này giới thiệu về mô hình bài toán nhận dạng tiếng nói. Cài đặt thuật toán sử dụng điểm cắt zero để rút trích đặc trưng và nhận dạng một số phụ âm Tiếng việt. Chương này cũng trình bày giao diện chương trình demo, giao diện các chức năng dùng để nhận dạng các file wave phụ âm tiếng việt và kết quả thực nghiệm đề tài.

Tôi xin bày tỏ lòng biết ơn đến PGS.TS **Nguyễn Văn Xuất** đã tận tình giúp tôi hoàn thành luận văn này, đồng thời tôi cũng xin cảm ơn các ban ngành, các thầy cô trong Khoa công nghệ thông tin - HVKTQS đã tạo điều kiện giúp tôi trong quá trình làm luận văn. Xin cảm ơn các bạn trong lớp cao học CNTT-K21 đã giúp tôi về tài liệu và bổ trợ thêm kiến thức để tôi nghiên cứu đề tài này.

Chương 1

LỚP CÁC BÀI TOÁN NHẬN DẠNG TIẾNG NÓI

1.1. Nhận dạng tiếng nói và một số phương pháp nhận dạng tiếng nói

1.1.1. Nhận dạng tiếng nói

Hiểu một cách đơn giản, nhận dạng tiếng nói (speech recognition by machine) là dùng máy tính chuyển đổi tín hiệu ngôn ngữ từ dạng âm thanh thành dạng văn bản.

Nhận dạng tiếng nói có nhiều ứng dụng:

- Đọc chính tả: Là ứng dụng được sử dụng nhiều nhất trong các hệ nhận dạng. Thay vì nhập liệu bằng tay thông qua bàn phím, người sử dụng nói với máy qua micro và máy xác định các từ được nói trong đó.

- Điều khiển - giao tiếp không dây: Chẳng hạn hệ thống cho phép máy tính nhận lệnh điều khiển bằng giọng nói của con người như: *"chạy chương trình", "tắt máy"*… Một số ưu điểm của việc sử dụng tiếng nói thay cho các thiết bị vào chuẩn như bàn phím, con chuột là: thuận tiện, tốc độ cao, không bị ảnh hưởng của cáp, khoảng cách, không đòi hỏi huấn luyện sử dụng...

- Điện thoại-liên lạc: Một số hệ thống (chẳng hạn ở máy điện thoại di động) cho phép người sử dụng đọc tên người trong danh sách thay vì bấm số. Một số hệ thống khác (ở ngân hàng, trung tâm chứng khoán…) thực hiện việc trả lời tự động đối với các các cuộc gọi hỏi về tài khoản…

Tuy nhiên vấn đề nhận dạng tiếng nói gặp rất nhiều khó khăn. Một số khó khăn chủ yếu là: Tiếng nói là tín hiệu thay đổi theo thời gian. Mỗi người có một giọng nói, cách phát âm khác nhau... Thậm chí một người phát âm cùng một từ mà mỗi lần khác nhau cũng không giống nhau (chẳng hạn về tốc độ, âm lượng...).

Các phương pháp nhận dạng hiện tại của máy tính khá "máy móc", còn xa mới đạt đến mức độ tư duy của con người. Nhiễu là thành phần luôn gặp trong môi trường hoạt động của các hệ thống nhận dạng và ảnh hưởng rất nhiều đến kết quả nhận dạng.

Do những khó khăn đó, nhận dạng tiếng nói cần tri thức từ rất nhiều từ ngành khoa học liên quan:

- Xử lí tín hiệu: tìm hiểu các phương pháp tách các thông tin đặc trưng, ổn định từ tín hiệu tiếng nói, giảm ảnh hưởng của nhiễu và sự thay đổi theo thời gian của tiếng nói.

- Âm học: tìm hiểu mối quan hệ giữa tín hiệu tiếng nói vật lí với các cơ chế sinh lí học của việc phát âm và việc nghe của con người.

- Nhận dạng mẫu: nghiên cứu các thuật toán để phân lớp, huấn luyện và so sánh các mẫu dữ liệu...

- Lí thuyết thông tin: nghiên cứu các mô hình thống kê, xác suất; các thuật toán tìm kiếm, mã hoá, giải mã, ước lượng các tham số của mô hình…

- Ngôn ngữ học: tìm hiểu mối quan hệ giữa ngữ âm và ngữ nghĩa, ngữ pháp, ngữ cảnh của tiếng nói.

- Tâm-sinh lí học: tìm hiểu các cơ chế bậc cao của hệ thống nơron của bộ não người trong các hoạt động nghe và nói.

- Khoa học máy tính: nghiên cứu các thuật toán, các phương pháp cài đặt và sử dụng hiệu quả các hệ thống nhận dạng trong thực tế.

Do tính phức tạp của bài toán nhận dạng tiếng nói người ta chia bài toán này thành các lớp bài toán sau:

- Nhận dạng tiếng nói trong môi trường không có nhiễu.

- Nhận dạng tiếng nói trong môi trường có nhiễu.

- Nhận dạng tiếng nói liên tục: Nghĩa là giữa các từ, các câu không có khoảng lặng.

- Nhận dạng tiếng nói rời rạc: Nghĩa là giữa các từ, các câu có khoảng lặng.

- Nhận dạng tiếng nói với số lượng từ hạn chế, số người nói hạn chế.

- Nhận dạng tiếng nói với số lượng từ hạn chế, số người nói không hạn chế.

- Nhận dạng tiếng nói số lượng từ không hạn chế, số người nói không hạn chế.

- Nhận dạng tiếng nói kết hợp các bài toán của 7 dạng trên.

Thực tế cho đến nay, mặc dù người ta đã đầu tư nhiều công sức để giải quyết các bài toán nhận dạng tiếng nói, song độ tin cậy đạt được chưa cao. Vì vậy vẫn chưa được ứng dụng rộng rãi trong thực tiễn.

1.1.2. Các phương pháp nhận dạng tiếng nói

*** Phương pháp Âm học - Ngữ âm học:**

Phương pháp này dựa trên lý thuyết về Âm học - Ngữ âm học. Lý thuyết đó cho biết: tồn tại các đơn vị ngữ âm xác định, có tính phân biệt trong lời nói và các đơn vị ngữ âm đó được đặc trưng bởi một tập các tín hiệu tiếng nói. Các bước nhận dang của phương pháp gồm:

Bước 1: Phân đoạn và gán nhãn. Bước này chia tín hiệu tiếng nói thành các đoạn có đặc tính âm học đặc trưng cho một (hoặc một vài) đơn vị ngữ âm, đồng thời gán cho mỗi đoạn âm thanh đó một hay nhiều nhãn ngữ âm phù hợp.

Bước 2: Nhận dạng. Bước này dựa trên một số điều kiện ràng buộc về từ vựng, ngữ pháp v.v…để xác định một hoặc một chuỗi từ đúng trong các

chuỗi nhãn ngữ âm được tạo ra sau bước:

Sơ đồ khối của phương pháp này được biểu diễn ở (Hình 1.1). Nguyên lý hoạt động của phương pháp có thể mô tả như sau:

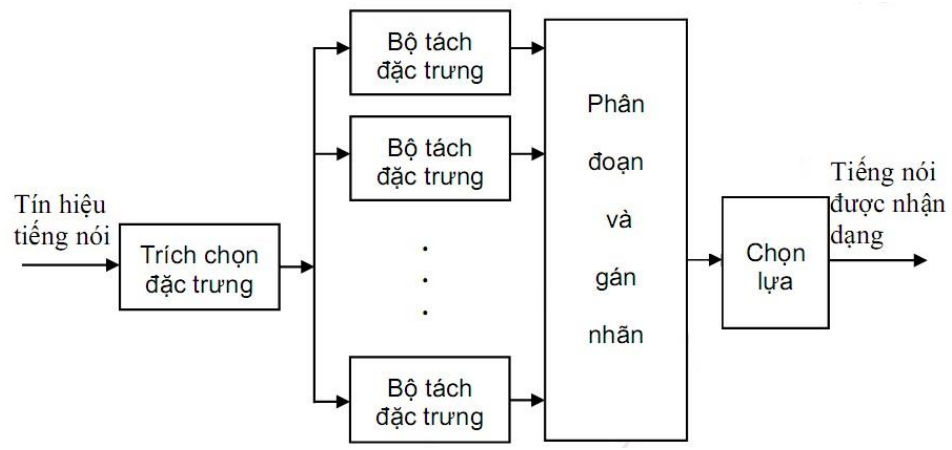
Trích chọn đặc trưng: Tín hiệu tiếng sau khi số hóa được đưa tới khối trích chọn đặc trưng nhằm xác định các phổ tín hiệu. Các kỹ thuật trích chọn đặc trưng tiếng nói phổ biến là sử dụng băng lọc (filter bank), mã hóa dự đoán tuyến tính (LPC)… Tách tín hiệu tiếng nói nhằm biến đổi phổ tín hiệu thành một tập các đặc tính mô tả các tính chất âm học của các đơn vị ngữ âm khác nhau. Các đặc tính đó có thể là: tính chất các âm mũi, âm xát; vị trí các formant; âm hữu thanh, vô thanh; tỷ số mức năng lượng tín hiệu…

Phân đoạn và gán nhãn: Ở bước này hệ thống nhận dạng tiếng xác định các vùng âm thanh ổn định (vùng có đặc tính thay đổi rất ít) và gán cho mỗi vùng này một nhãn phù hợp với đặc tính của đơn vị ngữ âm. Đây là bước quan trọng của hệ nhận dạng tiếng nói theo khuynh hướng Âm học - Ngữ âm học và là bước khó đảm bảo độ tin cậy nhất.

Nhận dạng: Chọn lựa để kết hợp chính xác các khối ngữ âm tạo thành các từ nhận dạng.

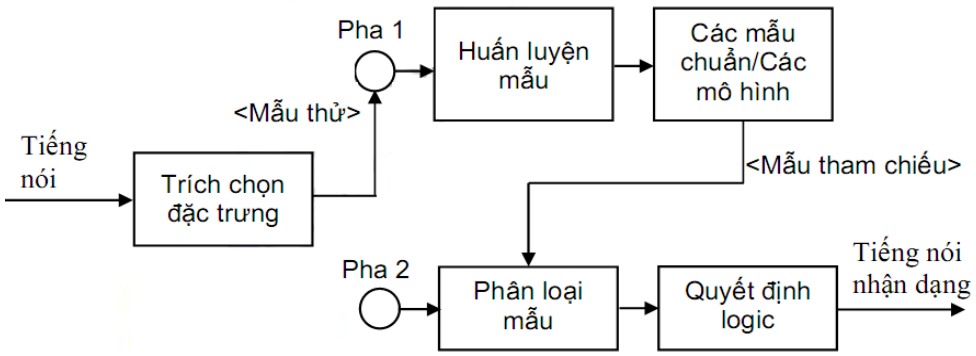
Đặc điểm của phương pháp nhận dạng tiếng nói theo hướng tiếp cận Âm học - Ngữ âm học:

- Người thiết kế phải có kiến thức khá sâu rộng về Âm học - Ngữ âm học.
- Phân tích các khối ngữ âm mang tính trực giác, thiếu chính xác.
- Phân loại tiếng nói theo các khối ngữ âm thường không tối ưu do khó sử dụng các công cụ toán học để phân tích.



Hình 1.1- Sơ đồ khối nhận dạng tiếng nói theo Âm học - Ngữ âm học

*** Phương pháp nhận dạng mẫu**



Hình 1.2- Sơ đồ khối hệ nhận dạng tiếng nói theo phương pháp mẫu

Phương pháp nhận dạng mẫu không cần xác định đặc tính âm học hay phân đoạn tiếng nói mà sử dụng trực tiếp các mẫu tín hiệu tiếng nói trong quá trình nhận dạng. Các hệ thống nhận dạng tiếng nói theo phương pháp này được phát triển theo hai bước (Hình 1.2), cụ thể là:

Bước 1: Sử dụng tập mẫu tiếng nói (cơ sở dữ liệu mẫu tiếng nói) để đào tạo các mẫu tiếng nói đặc trưng (mẫu tham chiếu) hoặc các tham số hệ thống.

Bước 2: Đối sánh mẫu tiếng nói từ ngoài với các mẫu đặc trưng để ra quyết định.

Trong phương pháp này, nếu cơ sở dữ liệu tiếng nói cho đào tạo có đủ các phiên bản mẫu cần nhận dạng thì quá trình đào tạo có thể xác định chính xác các đặc tính âm học của mẫu (các mẫu ở đây có thể là âm vị, từ, cụm từ…). Hiện nay, một số kỹ thuật nhận dạng mẫu được áp dụng thành công trong nhận dạng tiếng nói là lượng tử hóa vector, so sánh thời gian động (DTW), mô hình Markov ẩn (HMM), mạng nơron nhân tạo (ANN).

Hệ thống bao gồm các hoạt động sau:

Trích chọn đặc trưng: Tín hiệu tiếng nói được phân tích thành chuỗi các số đo để xác định mẫu nhận dạng. Các số đo đặc trưng là kết quả xử lý của các kỹ thuật phân tích phổ như: lọc thông dải, phân tích mã hóa dự đoán tuyến tính (LPC), biến đổi Fourier rời rạc (DFT).

Huấn luyện mẫu: Nhiều mẫu tiếng nói ứng với các đơn vị âm thanh cùng loại dùng để đào tạo các mẫu hoặc các mô hình đại diện, được gọi là mẫu tham chiếu hay mẫu chuẩn.

Nhận dạng: Các mẫu tiếng nói được đưa tới khối phân loại mẫu. Khối này đối sánh mẫu đầu vào với các mẫu tham chiếu. Khối nhận dạng căn cứ vào các tiêu chuẩn đánh giá để quyết định mẫu tham chiếu nào giống mẫu đầu vào.

Một số đặc điểm của phương pháp nhận dạng mẫu:

- Hiệu năng của hệ phụ thuộc vào số mẫu đưa vào. Nếu số lượng mẫu càng nhiều thì độ chính xác của hệ càng cao; tuy nhiên, dung lượng nhớ và thời gian huấn luyện mẫu tăng.

- Các mẫu tham chiếu phụ thuộc vào môi trường thu âm và môi trường truyền dẫn.

- Không đòi hỏi kiến thức sâu về ngôn ngữ.

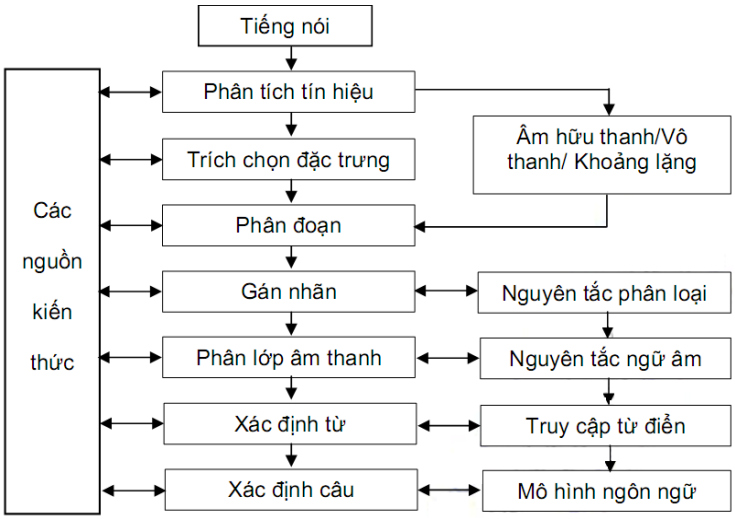
*** Phương pháp ứng dụng trí tuệ nhân tạo**

Phương pháp ứng dụng trí tuệ nhân tạo kết hợp các phương pháp trên nhằm tận dụng tối đa các ưu điểm của chúng, đồng thời bắt chước các khả năng của con người trong phân tích và cảm nhận các sự kiện bên ngoài để áp dụng vào nhận dạng tiếng nói. Sơ đồ khối của phương pháp trí tuệ nhân tạo theo mô hình từ dưới lên (bottom-up) (Hình 1.3).

Đặc điểm của các hệ thống nhận dạng theo phương pháp này là:

Sử dụng hệ chuyên gia để phân đoạn, gán nhãn ngữ âm. Điều này làm đơn giản hóa hệ thống so với phương pháp nhận dạng ngữ âm.

Sử dụng mạng nơron nhân tạo để học mối quan hệ giữa các ngữ âm, sau đó dùng nó để nhận dạng tiếng nói.



Hình 1.3- Sơ đồ khối hệ nhận dạng tiếng nói theo phương pháp từ dưới lên

Việc sử dụng hệ chuyên gia nhằm tận dụng kiến thức con người vào hệ nhận dạng:

Kiến thức về âm học: Để phân tích phổ và xác định đặc tính âm học của các mẫu tiếng nói.

Kiến thức về từ vựng: sử dụng để kết hợp các khối ngữ âm thành các

từ cần nhận dạng.

Kiến thức về cú pháp: nhằm kết hợp các từ thành các câu cần nhận dạng.

Kiến thức về ngữ nghĩa: nhằm xác định tính logic của các câu đã được nhận dạng.

Có nhiều cách khác nhau để tổng hợp các nguồn kiến thức vào bộ nhận dạng tiếng nói. Phương pháp thông dụng nhất là xử lý "từ dưới lên". Theo cách này, tiến trình xử lý của hệ thống được triển khai tuần tự từ thấp lên cao. Trong (Hình 1.3), các bước xử lý ở mức thấp (phân tích tín hiệu, tìm đặc tính, phân đoạn, gán nhãn) được triển khai trước khi thực hiện các bước xử lý ở mức cao (phân lớp âm thanh, xác định từ, xác định câu). Mỗi bước xử lý đòi hỏi một hoặc một số nguồn kiến thức nhất định. Ví dụ: bước phân đoạn tiếng nói cần hiểu biết sâu sắc về đặc tính Âm học - Ngữ âm học của các đơn vị ngữ âm; bước xác định từ đòi hỏi kiến thức về từ vựng; bước xác định câu đòi hỏi kiến thức về mô hình ngôn ngữ (nguyên tắc ngữ pháp).

Phương pháp này đã và đang được áp dụng thành công trong các ứng dụng nhận dạng tiếng nói thực tế. Bước đầu tiên của quá trình nhận dạng là trích chọn các tham số tín hiệu tiếng nói.

Phân tích tham số tiếng nói:

Trong nhận dạng, tổng hợp, mã hóa tiếng nói đều cần phân tích các tham số. Dưới đây, mô tả phương pháp phân tích cepstral theo thang đo mel để tính các hệ số MFCC thông qua việc sử dụng dãy các băng lọc.

Khái niệm cơ bản trong phân tích tín hiệu tiếng nói là phân tích thời gian ngắn (Short Time Analysis). Trong khoảng thời gian dài, tín hiệu tiếng nói là không dừng, nhưng trong khoảng thời gian đủ ngắn (10-30 ms) tiếng

nói được coi là dừng. Do đó, trong các ứng dụng xử lý tiếng nói người ta thường chia tiếng nói thành nhiều đoạn có thời gian bằng nhau được gọi là khung (frame), mỗi khung có độ dài từ 10 đến 30 ms.

Phát hiện tiếng nói:

Phát hiện thời điểm bắt đầu, điểm kết thúc của tiếng nói (tách tiếng nói ra khỏi khoảng lặng) là phần cần thiết trong chương trình nhận dạng tiếng nói, đặc biệt trong chế độ thời gian thực.

1.2. Nhận dạng tiếng Việt

1.2.1. Một số đặc điểm ngữ âm tiếng Việt

Một đặc điểm dễ thấy là tiếng Việt là ngôn ngữ đơn âm (monosyllable - mỗi từ đơn chỉ có một âm tiết), không biến hình (cách đọc, cách ghi âm không thay đổi trong bất cứ tình huống ngữ pháp nào). Tiếng Việt hoàn toàn khác với các ngôn ngữ Ấn-Âu như tiếng Anh, tiếng Pháp là các ngôn ngữ đa âm, biến hình.

Theo thống kê trong tiếng Việt có khoảng 6000 âm tiết. Nhìn về mặt ghi âm: âm tiết tiếng Việt có cấu tạo chung là: phụ âm-vần. Ví dụ âm *tin* có phụ âm *t*, vần *in*. Phụ âm là một âm vị và âm vị này liên kết rất lỏng lẻo với phần còn lại của âm tiết (hiện tượng nói lái).

Vần trong tiếng Việt lại được cấu tạo từ các âm vị nhỏ hơn, trong đó có một âm vị chính là nguyên âm.

Quan sát phổ các âm tiết chúng ta có thể rút ra kết luận: các phụ âm và nguyên âm đều phân biệt với nhau rất rõ qua sự phân bố năng lượng tại các miền tần số, ví dụ: phụ âm ở tần số thấp, năng lượng nhỏ, nguyên âm có năng lượng lớn ở cả vùng tần số cao. Vùng không có tín hiệu tiếng nói (nhiễu nền và khoảng lặng) có năng lượng thấp và chỉ tập trung ở các tần số rất thấp.

Sự khác biệt về cách phát âm tiếng Việt rất rõ rệt theo giới, lứa tuổi và

đặc biệt là theo vị trí địa lí (giọng miền Bắc, miền Trung và miền Nam khác nhau rất nhiều).

1.2.2. Những thuận lợi và khó khăn đối với nhận dạng tiếng nói tiếng Việt

*** Thuận lợi**

Những đặc điểm ngữ âm tiếng Việt cho thấy nhận dạng tiếng nói tiếng Việt có một số thuận lợi sau:

- Tiếng Việt là ngôn ngữ đơn âm, số lượng âm tiết không quá lớn. Điều này sẽ giúp hệ nhận dạng xác định ranh giới các âm tiết dễ dàng hơn nhiều. Đối với hệ nhận dạng các ngôn ngữ Ấn-Âu (tiếng Anh, tiếng Pháp...) xác định ranh giới âm tiết (endpoint detection) là vấn đề rất khó và ảnh hưởng lớn đến kết quả nhận dạng.

- Tiếng Việt là ngôn ngữ không biến hình từ. Âm tiết tiếng Việt ổn định, có cấu trúc rõ ràng. Đặc biệt không có 2 âm tiết nào đọc giống nhau mà viết khác nhau. Điều này sẽ dễ dàng cho việc xây dựng các mô hình âm tiết trong nhận dạng; đồng thời việc chuyển từ phiên âm sang từ vựng (lexical decoding) sẽ đơn giản hơn so với các ngôn ngữ Ấn-Âu. Theo [10], việc chuyển từ phiên âm sang từ vựng cũng là một vấn đề khó khăn trong nhận dạng các ngôn ngữ Ấn-Âu.

*** Khó khăn**

Ngoài những thuận lợi trên, nhận dạng tiếng nói tiếng Việt cũng gặp rất nhiều khó khăn như sau:

- Tiếng Việt là ngôn ngữ có thanh điệu (6 thanh). Thanh điệu là âm vị siêu đoạn tính, đặc trưng về thanh điệu thể hiện trong tín hiệu tiếng nói không rõ nét như các thành phần khác của âm tiết.

- Cách phát âm tiếng Việt thay đổi nhiều theo vị trí địa lí. Giọng

địa phương trong tiếng Việt rất đa dạng (mỗi miền có một giọng đặc trưng).

- Hệ thống ngữ pháp, ngữ nghĩa tiếng Việt rất phức tạp, rất khó để áp dụng vào hệ nhận dạng với mục đích tăng hiệu năng nhận dạng. Hệ thống phiên âm cũng chưa thống nhất.

- Các nghiên cứu về nhận dạng tiếng Việt cũng chưa nhiều và ít phổ biến. ***Đặc biệt khó khăn lớn nhất là hiện nay chưa có một bộ dữ liệu chuẩn cho việc huấn luyện và kiểm tra các hệ thống nhận dạng tiếng Việt.***

1.3. Mục tiêu của luận văn

Điều kiện tiên quyết của bài toán nhận dạng về bản chất là phải xây dựng được các đặc trưng của tiếng nói. Mục tiêu của luận văn là đi tìm các đặc trưng của một số phụ âm trong tiếng việt phục vụ cho bài toán nhận dạng tiếng nói rời rạc, số người nói không hạn chế, số từ không hạn chế và có nhiễu.

Chương 2

SỐ HOÁ TÍN HIỆU ÂM THANH VÀ ĐẶC TRƯNG TÍN HIỆU TIẾNG NÓI

2.1. Âm thanh

2.1.1. Âm thanh và tiếng nói

Âm thanh là các dao động cơ học (biến đổi vị trí qua lại) của các phân tử, nguyên tử hay các hạt làm nên vật chất và lan truyền trong vật chất như các sóng. Âm thanh, giống như nhiều sóng, được đặc trưng bởi tần số, bước sóng, chu kỳ, biên độ và vận tốc lan truyền (tốc độ âm thanh).

Đối với thính giác của người, âm thanh thường là sự dao động, trong dải tần số từ khoảng 20 Hz đến khoảng 20 kHz, của các phân tử không khí, và lan truyền trong không khí, va đập vào màng nhĩ, làm rung màng nhĩ và kích thích bộ não. Tuy nhiên âm thanh có thể được định nghĩa rộng hơn, tuỳ vào ứng dụng, bao gồm các tần số cao hơn hay thấp hơn tần số mà tai người có thể nghe thấy, và không chỉ lan truyền trong không khí, mà trong bất cứ vật liệu nào. Trong định nghĩa rộng này, âm thanh là sóng cơ học và theo lưỡng tính sóng hạt của vật chất, sóng này cũng có thể coi là dòng lan truyền của các hạt phonon, các hạt lượng tử của âm thanh.

2.1.2. Mô hình toán của sóng âm thanh

Như trên đã đề cập, đại lượng mô tả sự biến đổi của sóng âm thanh theo thời gian là đại lượng liên tục, về nguyên tắc ta có thể coi nó là hàm phụ thuộc thời gian g(t) liên tục theo t. Vì g(t) là hàm mô tả sự biến đổi của sóng, nên nó có chu kỳ T và tần số f, nghĩa là:

- Tồn tại T: g(t+T) = g(t) với mọi t.

- Tồn tại tần số f, G(f) là kết quả của phép biến đổi Fourier ngược của hàm g(t), hàm G(f) là hàm phụ thuộc tần số.

- Giữa chu kỳ T và tần số f có quan hệ sau: T=1/f

2.1.3. Các đặc tính cơ bản của tiếng nói

- Tốc độ lan truyyền: vận tốc dẫn truyền trong không khí là 344m/s ở 20^0C và ở mực nước biển, và tăng lên theo nhiệt độ và độ cao.

- Khoảng cách nghe được: dưới 100m

- Miền tần số cơ bản: 80 – 8000Hz

- Dải tần trung bình: 300 – 3400 Hz

- Cường độ: tỉ lệ thuận với thể tích và áp lực của nguồn hơi đi qua khe thanh môn. Người ta đã đo áp lực của luồng hơi đi qua khe thanh môn tương đương với cường độ âm thanh phát ra như sau:

- Áp lực bằng 60cm nước trong tiếng gọi to (60dB).
- Áp lực bằng 100cm nước trong tiếng gọi to (70dB.)
- Áp lực bằng 160cm nước trong tiếng nói của nhà hùng biện đang diễn thuyết (80dB).
- Áp lực bằng 360cm nước trong giọng hát nam cao đại hợp xướng (120dB).

Khi phát âm với tần số tăng dần thì thường cường độ sẽ giảm dần. Và trong một câu nói thì sẽ có khoảng trống hoặc khoảng có cường độ thấp giữa các tiếng với tần số khoảng 10Hz.

- Cao độ: cao độ của tiếng nói có liên quan trực tiếp đến tần số rung của dây thanh. Ở người bình thường, tần số này càng thấp và trầm nếu dây thanh càng dài.

- Âm sắc: âm sắc của giọng nói do nhiều yếu tố tạo ra, nó phụ thuộc vào cách khép cửa lại của hai dây thanh, vào sự hình thành của hoà âm tại các hốc cộng hưởng mũi, miệng, họng…Âm sắc là một đặc điểm riêng biệt của từng cá nhân.

2.2. Số hoá âm thanh

2.2.1. Lượng hoá

Giả sử nguồn analog phát đi ở dạng sóng là hàm x(t).

Giả sử x(t) thỏa mãn các điều kiện sau:

- Là hàm của quá trình ngẫu nhiên (stochatic)

- X(t) có giải băng thông hữu hạn (bandlimited)

- X(t) là quá trình tĩnh (station stochatic - không thay đổi khi dịch chuyển t; x(t) = x(t+p))

Tốc độ lấy mẫu là số mẫu lấy trong một giây kí hiệu f_s, đơn vị tính là Hez.

Khi đó theo định lý lấy mẫu Nyquit thì chúng ta có thể khôi phục x(t) bằng dãy các mẫu tín hiệu lấy theo tốc độ Nyquit (Nyquits rate).

- Theo Nyquit $f_s >= 1/2f$, ở đây f là tần số lớn nhất của x(t).

Để nghiên cứu x(t) trên máy tính, người ta phải số hóa hàm analog x(t), để số hóa thường người ta phải thực hiện quá trình gồm hai bước là lượng tử hóa và mã hóa (quantized and encoded).

- Lượng hóa thực chất là quá trình lấy x’(t) gần đúng giá trị x(t)

- Mã hóa là quá trình chuyển x’(t) thành dãy các bit.

Một trong các phương pháp lượng tử hóa và mã hóa đơn giản là biểu diễn mỗi mức biên độ (amplitude level) bằng dãy số nhị phân. Giả sử chúng ta có L mức, gọi $R=\log_2 L$, khi đó mỗi mẫu cần R bit(R bit/mẫu).

- Trong thực tế nếu L là bội mũ của cơ số 2 ta sử dụng số bit cho mẫu là $R=\log_2 L$. Nếu L không là bội mũ của cơ số 2 ta chọn R.

$$R = Int(\log_2 L)+1 \qquad (2.1)$$

- Nếu các mức có xác suất không bằng nhau và chúng ta lại biết được xác suất của mỗi mức, khi đó chúng ta có thể sử dụng mã Huffman (hay loại theo Entropy để tăng hiệu quả của việc mã hóa).

- Việc lượng hóa biên độ của các tín hiệu mẫu cũng tương tự như nén dữ liệu, nó sẽ làm mất mát thông tin, vấn đề làm thế nào để đánh giá được sự mất mát, sai lệch trong quá trình số hóa.

2.2.2. Đánh giá sai số trong quá trình lượng hoá

Ta xét độ sai lệch trong trường hợp mẫu của nguồn vào được lượng hóa thành một số bit cố định.

Giả sử $\{x_k\}$ là dãy các mẫu tín hiệu vào từ nguồn analog và $\{x'_k\}$ là dãy giá trị lượng hóa tương ứng của $\{x_k\}$

Ta gọi $d(x_k, x'_k) = (x_k, x'_k)^2$ là độ đo sai lệch bình phương (quared – error distortion).

Ta gọi $d(x_k, x'_k) = (x_k, x'_k)^p$ với p nguyên dương là độ đo sai lệch cấp p.

Giả sử $X_n = \{x_1,x_2,\ldots,x_n\}$ và $X'_n = \{x'_1,x'_2,\ldots,x'_n\}$ ta gọi độ sai lệch của hai dãy hữu hạn tín hiệu là :

$$d(Xn, X'n) = (d(x1, x'1)+ d(x2, x'2)+\ldots..+ d(xn, x'n))/n \qquad (2.2)$$

Vì nguồn ra là quá trình ngẫu nhiên, n mẫu của X_n là biến ngẫu nhiên, do đó $d(X_n, X'_n)$ cũng là biến ngẫu nhiên, Giá trị kỳ vọng (expected value được coi là độ sai lệch của hai dãy kí hiệu là D.

$$D=E[d(X_n,X'_n)]=(E[d(x_1,x'_1)]+E[d(x_2,x'_2)]+\ldots+E[d(x_n,x'_n)])/n \quad (2.3)$$

Ở đây E – là kỳ vọng của đại lượng ngẫu nhiên.

2.2.3. Thang lượng hoá

Việc lượng hóa tín hiệu nguồn có thể tối ưu nếu chúng ta biết được hàm phân bố xác suất p(x) của cường độ tín hiệu ở đầu vào của quá trình lượng hóa.

Giả sử dãy tín hiệu vào $\{x_n\}$ của quá trình lượng hóa có hàm mật độ phân bố xác xuất p(x) và giả sử $L=2^R$ là số mức cần lượng hóa, chúng ta muốn thiết kế thang lượng hóa tối ưu thì làm cực tiểu hàm lỗi f(x - x') của quá trình lượng hóa, ở đây x' là giá trị lượng hóa của x khi đó kết quả của lượng hóa sinh ra lỗi là:

$$D = \int_{-\infty}^{+\infty} f(x'-x)p(x)dx \quad (2.4)$$

Nói chung để tối ưu hóa việc lượng hóa ta cần tìm cách làm cực tiểu D. Người ta có thể chọn các mức đầu ra tương ứng với dãy tín hiệu vào theo thang lương hóa để làm cực tiểu D. Bài toán tối ưu này đã được giải quyết bởi Lloyd (1982) và Max(1960), vì vậy người ta thường gọi lượng hóa Lloyd-Max (Lloyd-Max quantizer).

2.2.4. Một số kỹ thuật mã hoá nguồn Analog

Để minh họa cho mô hình nêu trên, phần này giới thiệu một số kỹ thuật mã hóa nguồn analog đã được phát triển từ những năm 1940, các kỹ thuật này thường được áp dụng để mã hóa tiếng nói hoặc hình ảnh .

Người ta phân chia kỹ thuật mã hóa các nguồn analog thành ba loại:

- Mã hóa dạng sóng theo thời gian (Temporal waveform coding).

- Mã hóa theo phổ của dạng sóng.

- Mã hóa theo mô hình toán học (model – based coding).

Dưới đây chúng ta xem xét một vài cách mã hóa thuộc các dạng trên:

a. Kỹ thuật mã PCM (pulse code modulation)

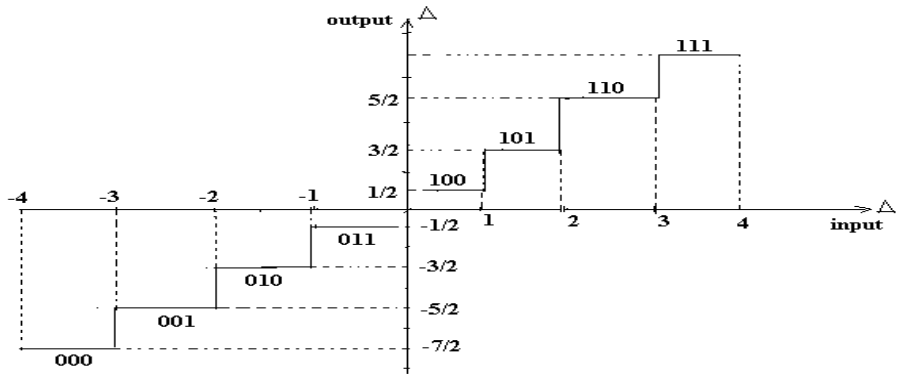
Kỹ thuật mã hóa PCM là kỹ thuật mã theo thời gian (Temporal waveform coding).

Ý tưởng của kỹ thuật mã PCM là điều chỉnh mã theo từng mức của biên độ. Giả sử x(t) là hàm mẫu (sample function) phát ra bởi nguồn, giả sử x_n là mẫu lấy theo tần số $f_s>=2f$ ở đây f là tần số cao nhất của các tần số trong phổ của x(t). Trong kỹ thuật này mỗi tín hiệu mẫu được lượng hóa thành một giá trị thuộc 2^R mức biên độ (amplitude level), với R là số chữ số nhị phân dùng để biểu diễn mẫu. Khi đó tốc độ truyền đi của nguồn là Rf_s bit/s.

Quá trình lượng hóa có thể mô hình hóa ở dạng toán học như sau:

Gọi x'_n là giá trị lượng hóa tín hiệu x_n và q_n là lỗi lượng hóa mà chúng ta có thể coi như nhiễu. Khi đó ta có: $x'_n= x_n+q_n$ (2.5)

Chúng ta sử dụng cách lượng hóa đều .Hình 2.1 minh họa cho cách mã hóa của kỹ thuật PCM



Hình 2.1- Mô tả thang lượng tử

Hình 2.5 mô tả thang lượng tử với R=3; số mức lượng tử là 8; số bít cho mỗi tín hiệu mẫu là 3;

ví dụ: Nếu $-1/2 \le x_n < 3/2$ thì $x_n' = 100$; $3/2 \le x_n < 5/2$ thì $x_n' = 101$

Lượng hóa nhiễu được lấy theo thống kê:

lấy $P(q) = 1/\text{delta}$ với $-1/2\ \text{delta} \le q \le 1/2\ \text{delta}$

- Khi đó sai số trung bình được tính theo công thức :

$$E(q^2) = 1/12 * (\text{delta})^2 = 2^{-2R}/12$$

- Độ đo tính theo decibel, nghĩa là bình phương giá trị nhiễu ta có:

$$10 \log E(q^2) = -6R - 10.8\ \text{dB}$$

b. Kỹ thuật mã theo phổ (spectral waveform coding)

Trong kỹ thuật này người ta cho lọc nguồn và đưa tín hiệu ra ở một số dải tần số (subband) và mã riêng biệt từng tín hiệu theo mỗi tần số.

Trong kỹ thuật mã tiếng nói và ảnh người ta dùng kỹ thuật SBC, theo kỹ thuật này các tín hiệu được phân chia thành một số tần số và thời gian di chuyển của sóng theo từng tần số để mã. Chẳng hạn trong mã hóa tiếng nói giải tần thấp chứa năng lượng chủ yếu của phổ tiếng nói, ngoài ra nhiễu nghe rõ hơn trong tần số thấp. Vì lý do đó người ta dùng nhiều bít hơn khi mã tín hiệu ở tần số thấp và ít bít hơn để mã tín hiệu ở tần số cao.

Trong kỹ thuật SBC người ta thiết kế bộ lọc (filter) có thể đạt các thông số kỹ thuật cao, đó là bộ lọc QMF nó khả năng phân chia liên tiếp các tần số thấp thành hai thành phần. Ví dụ giải tần của tiếng nói là 3200 Hz. QMF sẽ phân chia phổ của hai cặp tần số thành hai phần, phần ứng tần tố thấp là 0-1666Hz; và phần ứng với tần số cao là 1600 - 3200 Hz. Sau đó phần ứng với

tần số thấp lại được phân chia thành 0 - 800; 800 - 1600v.v… Cuối cùng chúng ta nhận được dải tần số cho tiếng nói là:

0 - 400 Hz; 400 - 800 Hz; 800 -1600Hz; 1600 - 3200 Hz. Trong miền thời gian mỗi tần số có thể mã hóa theo độ chính xác khác nhau. Trong thực tế kỹ thuất mã thích nghi PCM mã các tín hiệu dạng sóng trong mỗi giải tần.

2.3. Các file âm thanh

Các tín hiệu analog sau khi đã số hóa (lượng hóa + mã hóa) sẽ được lưu lại ở dạng file. Đối với âm thanh người ta đưa ra rất nhiều dạng chuẩn như *.wav ;*.wma; *.cda ; *.mp4; *.m3d; *.mid; *.aif; *.aiff.....Để tiện cho việc thực hành chúng ta sẽ quan tâm đến các file âm thanh dạng chuẩn trong môi trường windows là *.wav và file *.mid.

Chúng ta cần chú ý file âm thanh dùng để ghi dữ liệu của âm thanh nói chung, nếu âm thanh phát ra có làn điệu, cung bậc thì các file lưu dữ liệu của loại âm thanh này gọi là các file audio, chẳng hạn các file lưu dữ liệu của các tác phẩm âm nhạc đều gọi là các file audio.

2.3.1. File dạng wav

Cách tạo:

Thiết bị để tạo file wave gồm :

- Máy tính có SoudCard, mic hoặc cassette, phần mềm window media player, hoặc SoundRecorder nối theo hình sau:

Cassette nối với máy tính có SoundCard. Tín hiệu âm thanh từ Cassette đưa vào máy tính là tín hiệu Analog.

SoundCard có hai khối A, B. Khối A nhận tín hiệu Analog. Khối B có nhiệm vụ sau một khoảng thời gian nào đó lại lấy tín hiệu từ khối A.

Số lần khối B lấy tín hiệu ở khối A trong một giây gọi là tần số lấy mẫu. Với một cách khác Số mẫu/Giây cũng gọi là tần số lấy mẫu.

Giá trị của tín hiệu hay của mẫu lấy được của khối B có thể biểu diễn ở dạng số 8 bít, 16 bít, 32 bít,.. gọi là độ phân giải của mẫu. Độ phân giải càng cao lớn thì sai số để biểu diễn mẫu càng nhỏ. Rõ ràng độ lớn của file *.wav phụ thuộc vào các yếu tố sau: tần số lấy mẫu, độ phân giải, thời gian ghi và số kênh ghi (mono hay stereo).

Giá trị của các mẫu có thể ghi lại ở dạng file trên máy tính. Đưa vào các mẫu này người ta lại khôi phục lại âm thanh.

2.3.2. Cấu trúc file Wave

Định dạng file wave là tập con trong định dạng file multimedia dạng RIFF của Micrrosoft. Các tập wave của Windows ứng dụng cho cả 2 dạng tệp âm thanh nổi (stereo) và dạng đơn (mono) với một tập các độ phân giải và tần số lấy mẫu. Kiểu tệp này cho phép sự định rõ RIFF (Resource Information File Format), và cho phép thông tin phụ của người sử dụng được nhúng vào và được ghi cùng với tệp âm thanh. Dạng âm thanh PCM dùng cho Windows chuẩn chứa dữ liệu đã được mã hóa, dữ liệu đã được định dạng theo kiểu điều biến mã xung dạng không bị nén.

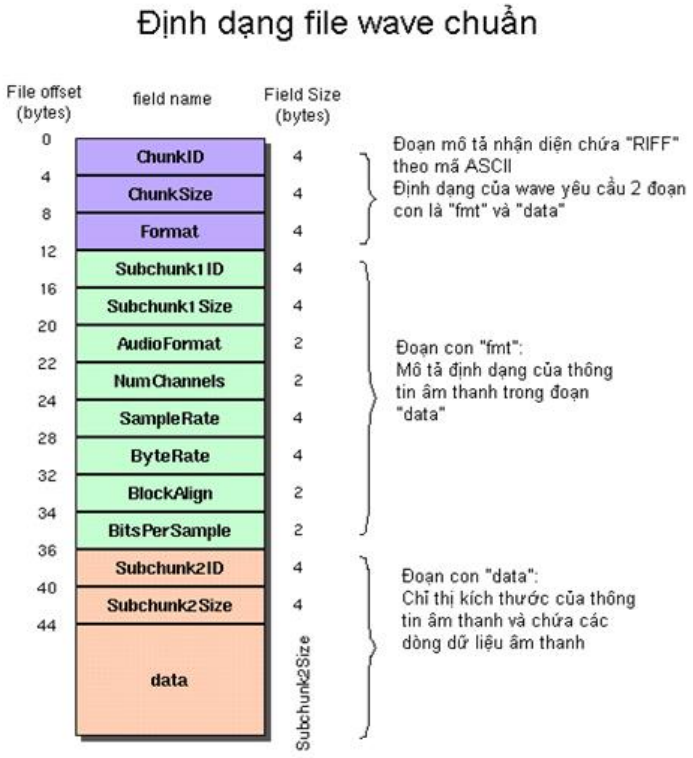
Do dạng tệp Wave là một dạng âm thanh tự nhiên được Microsoft Windows sử dụng, nên nó trở thành một trong các dạng âm thanh phổ biến nhất. Nói chung, cấu trúc của nó được phát triển dựa trên dạng khởi đầu.

Do dạng tệp Wave là một dạng âm thanh tự nhiên được Microsoft Windows sử dụng, nên nó trở thành một trong các dạng âm thanh phổ biến nhất. Microsoft định nghĩa một dạng tệp chung được gọi là Resource Interchange File Format (RIFF). Tệp Riff được tổ chức như một tập các khúc gài vào nhau. Và hai biến dạng chung là: Tệp Wave (hay Wav) lưu trữ âm

thanh và tệp AVI lưu trữ các hình ảnh video.

Cấu trúc tổng quát của một tệp Wave: Đây là khuôn dạng phổ biến nhất để lưu trữ âm thanh số trong thế giới PC. Nó được thiết kế cho các ứng dụng Multimedia chạy dưới Microsoft Windows. Tệp Wav lưu trữ các mẫu dạng sóng của một hoặc nhiều kênh theo các tần số lấy mẫu. Tệp Wav tự mã hóa và mô tả dữ liệu của nó trong phần mềm mà ta sử dụng. Nó không giới hạn độ dài tệp, có thể lến tới 4GB.

Một tệp Wave là một dạng đặc biệt của tệp RIFF, và mọi tệp đều bắt đầu với các ký tự RIFF. Tiếp theo đó là 4-byte độ dài và mã định dạng. Tệp Wav gồm các phần nhỉ gọi là khúc (chunk). Có hai loại khúc được dùng trong tệp wav: khúc dữ liệu (data chunk), như một định danh biểu diễn độ dài và bản thân dữ liệu; và khúc định dạng (forrmat chunk) chứa các dữ liệu mô tả thông tin trong nó. Khuôn dạng tệp Wave như sau:



Hình 2.2- Định dạng file WAVE chuẩn

Bảng 2.1- Định dạng file WAVE chuẩn được bắt đầu với RIFF header

0	4	**ChunkID**	Chứa từ “RIFF” mã ASCII
4	4	**ChunkSize**	36 + SubChunk2Size
8	4	**Format**	Chứa từ “WAVE”

Định dạng “WAVE” bao gồm 2 đoạn con: “fmt” và “data”

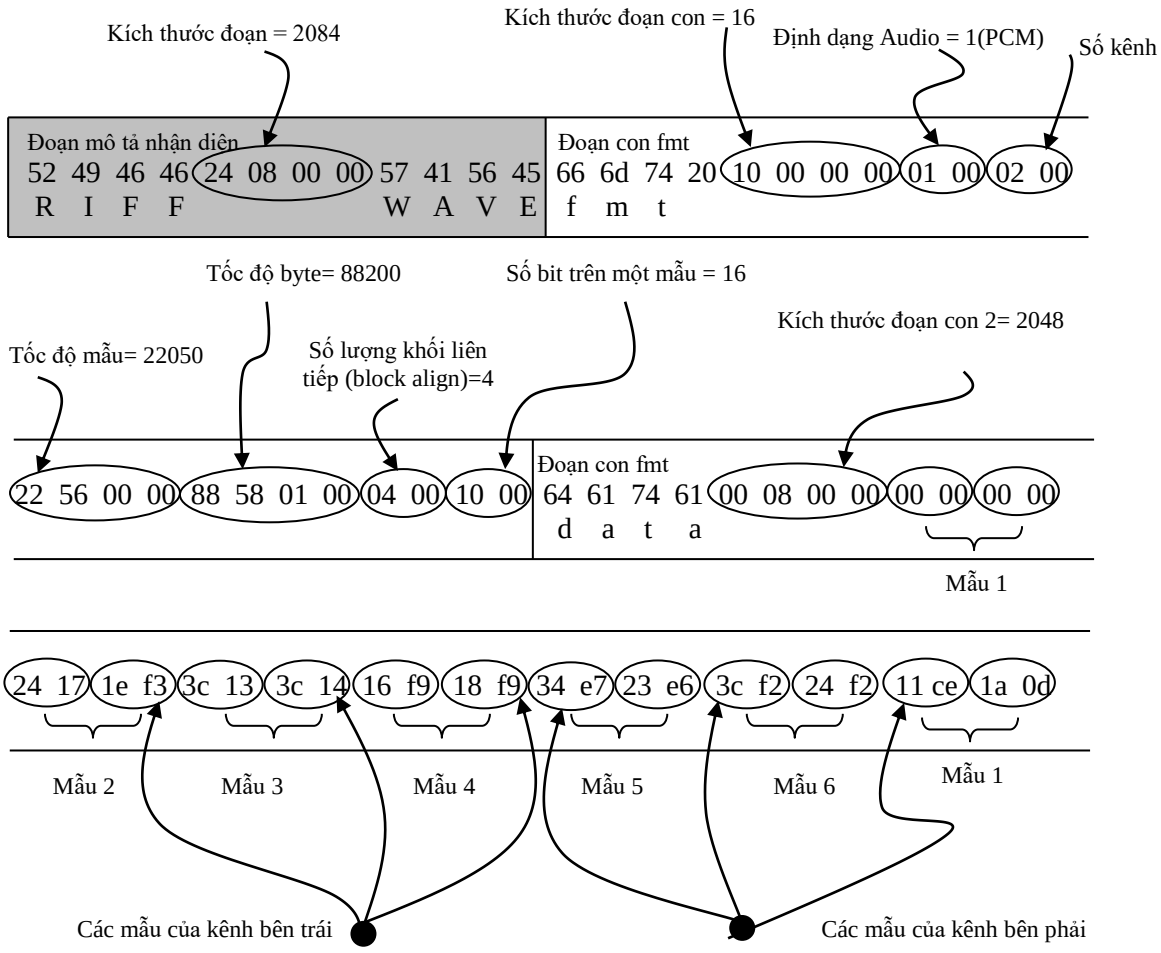
Bảng 2.2- Đoạn con “fmt” để mô tả định dạng dữ liệu âm thanh

12	4	**SubChunk1ID**	Chứa từ “fmt”
16	4	**SubChunk1Size**	16 cho PCM. Là độ dài của một mẫu dữ liệu
20	2	**Audio Format**	PCM = 1
22	2	**NumChannels**	Mono = 1, Stereo = 2
24	4	**SampleRate**	8000, 44100,…
28	4	**ByteRate**	= SampleRate* NumChannels * BitsPerSample/8
32	2	**BlockAlign**	=NumChannels*BitsPerSample/8. Là số byte của một mẫu chứa tất cả các kênh
34	2	**BitsPerSample**	8 bits = 8, 16 bits = 16

Bảng 2.3- Đoạn con “data” chứa kích thước của dữ liệu và dữ liệu âm thanh thực thi

36	4	**SubChunk2ID**	Chứa từ “data”
40	4	**SubChunk2Size**	= NumSamples * NumChannels * BitsPerSample/8. Là số byte của một phần tử dữ liệu âm thanh.
44	*	**Data**	Là phần dữ liệu âm thanh thực thi

Dưới đây là một ví dụ mở một file Wave với 71 bytes đầu file được biểu diễn dưới dạng số Hexa:



Hình 2.3- Phần Wave data

52 49 46 46 24 08 00 00 57 41 56 45 66 6d 74 20 10 00 00 00 01 00 02 00
22 56 00 00 88 58 01 00 04 00 10 00 64 61 74 61 00 08 00 00 00 00 00 00
24 17 1e f3 3c 13 3c 14 16 f9 18 f9 34 e7 23 a6 3c f2 24 f2 11 ce 1a 0d

Sự phụ thuộc độ lớn của file WAVE vào các yếu tố sau:

+ Tần số lấy mẫu: nếu tần số lấy mẫu càng cao thì dung lượng file càng lớn.

+ Độ phân giải: nếu độ phân giải càng lớn thì dung lượng file càng lớn.

+ Số kênh: số kênh càng nhiều thì dung lượng file càng lớn.

Do đó có rất nhiểu tệp wave có cùng cấu trúc cơ sở này nên nhiều

chương trình xử lý các tệp wave giống như chúng có một header nhất định. Điều này tạo thuận lợi cho việc viết các tệp wave dài như khi chỉ viết các dữ liệu PCM và thiết lập một cách thích đáng các trường kích thước. Sau đây là mô tả dạng tệp wave cơ bản:

Bảng 2.4. Mô tả file wave cơ bản

Kích thước (byte)	Mô tả
4	Dạng khúc: RIFF
4	Kích thước tệp tổng thể trừ 8
4	Dạng RIFF container: WAVE
4	Dạng khúc: fmt
4	Độ dài dữ liệu khúc định dạng: thông thường là 16
16	Dữ liệu khúc định dạng
4	Dạng khúc: data
4	Độ dài dữ liệu âm thanh
n	Các mẫu âm thanh hiện thời

2.4. Đặc trưng tín hiệu tiếng nói

Để có thể nhận dạng được tiếng nói người ta phải tìm cách trích lọc ra các đặc trưng cuả nó. Chất lượng hoặc độ tin cậy của nhận dạng tiếng nói phụ thuộc vào đặc trưng. Hiện nay người ta đã đưa ra rất nhiều phương pháp lấy đặc trưng. Dưới đây nêu một số phương pháp trích rút đặc trưng phổ biến.

2.4.1. Phương pháp mã dự đoán tuyến tính LPC (Linear Predictive Coding)

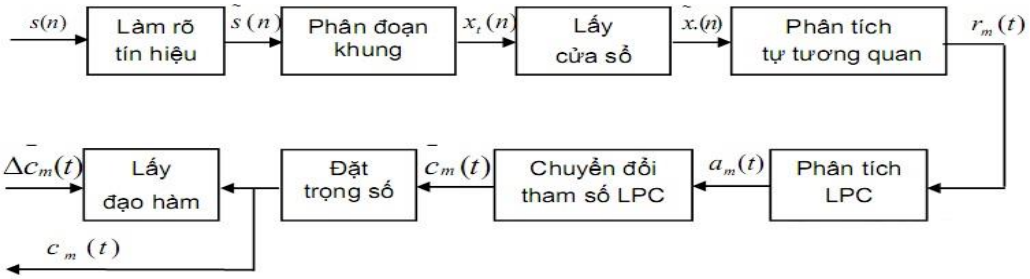
Mô hình LPC được sử dụng để trích lọc các tham số đặc trưng của tín hiệu tiếng nói. Kết quả của quá trình phân tích tín hiệu thu được một chuỗi gồm các khung tiếng nói. Các khung này được biến đổi nhằm sử dụng cho việc phân tích âm học.

Nội dung phân tích dự báo tuyến tính là: một mẫu tiếng nói được xấp xỉ bởi tổ hợp tuyến tính của các mẫu trước đó. Thông qua việc tối thiểu hóa tổng

bình phương sai số giữa các mẫu hiện tại với các mẫu dự đoán có thể xác định được một tập duy nhất các hệ số dự báo. Các hệ số dự báo này là các trọng số được sử dụng trong tổ hợp tuyến tính. Với dãy tín hiệu tiếng nói s(n) giá trị dự báo được xác định bởi:

$$\tilde{s}(n) = \sum_{k=1}^{p} a_k s(n-k) \qquad (2.6)$$

Trong đó; a_k: là các hệ số đặc trưng cho hệ thống.



Hình 2.3- Sơ đồ xử lý LPC dùng trích chọn đặc trưng tiếng nói

Sơ đồ khối bộ phân tích LPC dùng cho trích chọn các tham số đặc trưng của tín hiệu tiếng nói (Hình 2.3). Hàm sai số dự báo được tính theo công thức:

Để cực tiểu hóa lỗi cần tìm tập giá trị $\{\alpha_k\}$ phù hợp nhất.

Do tín hiệu tiếng nói thay đổi theo thời gian nên các hệ số dự báo phải được ước lượng từ các đoạn tín hiệu ngắn. Vấn đề đặt ra là tìm một tập các hệ số dự báo để tối thiểu hóa sai số trung bình trên một đoạn ngắn.

Hàm lỗi dự báo trong một thời gian ngắn xác định bởi:

$$E_n = \sum_m e_n^2(n) = \sum_m \left[s_n(m) - \sum_{k=1}^{p} \alpha_k s_n(m-k) \right]^2$$

$$= \sum_m s_n^2(m) = \sum_m \left\{ 2 s_n(m) \sum_{k=1}^{p} \alpha_k s_n(m-k) \right\} + \sum_m \left\{ \sum_{k=1}^{p} \alpha_k s_n(m-k) \right\}^2$$

$$= \sum_m e_n^2(n) = \sum_m \left\{ s_n(m) - \sum_{k=1}^{p} \alpha_k s_n(m-k) \right\}^2 \quad (2.7)$$

Trong đó; $s_n(m)$: là một đoạn tín hiệu tiếng nói lân cận mẫu thứ n;

Tìm tập giá trị α_k để tối thiểu hóa E_n bằng cách đặt $\partial E_n / \partial \alpha_i = 0$ với: $i = 1,2,...,p$

$$\frac{\partial E}{\partial \alpha_i} = 0 = -\sum_m 2 s_n(m) s_n(m-i) + 2 \sum_m \left\{ \sum_{k=1}^{p} \alpha_k s_n(m-k) \right\} s_n(m-i) \quad (2.8)$$

Từ đó nhận được phương trình:

$$\sum_m s_n(m-i) s_n(m) = \sum_{k=1}^{p} \alpha_k \sum_m s_n(m-i) s_n(m-k) \quad 1 \le i \le p \quad (2.9)$$

Đặt:

$$\phi_n(i,k) = \sum_m s_n(m-i) s_n(m-k) \quad (2.10)$$

Phương trình trên có thể viết:

$$\phi_n(i,0) = \sum_m \alpha_k \phi_n(i,k) \quad i = 1,2,3...p \quad (2.11)$$

Giải hệ p phương trình này tìm được p ẩn cuả $\{\alpha_k\}$. Tập các hệ số $\{\alpha_k\}$ sẽ tối thiểu sai số trung bình bình phương dự đoán cho đoạn tín hiệu $s_n(m)$. Sai số dự đoán được xác định:

$$E_n = \sum_m s_n^2(m) = \sum_{k=1}^{p} \alpha_k \sum_m s_n(m) s_n(m-k) \quad (2.12)$$

Sử dụng phép thế ta có:

$$E_n = \theta_n(0,0) - \sum_{k=1}^{p} \alpha_k \phi_n(0,k) \quad (2.13)$$

Theo nguyên tắc, phân tích dự đoán tuyến tính rất đơn giản nhưng việc tính toán $\theta_n(i,k)$ và tìm nghiệm của hệ phương trình rất phức tạp. Phương pháp khắc phục là sử dụng sự tương quan để giải các phương trình này.

Giả sử đoạn tín hiệu $s_n(m) = 0$ nếu chúng nằm ngoài khoảng $0 \le m \le N - 1$. Điều đó có nghĩa là có thể biểu diễn đoạn tín hiệu đó dưới dạng: $s_n(m) = s(n + m)w(m)$, trong đó: w(m) là cửa sổ có chiều dài hữu hạn (thường dùng cửa sổ Hamming). Sai số dự đoán $E_m(m)$:

$$E_n = \sum_{m=0}^{N+p-1} e_n^2(m) \tag{2.14}$$

Khi đó (3.5) trở thành:

$$\phi_n(i,k) = \sum_{m=0}^{N+p-1} s_n(m)s_n(m-k) \qquad \begin{matrix} 1 \le i \le p \\ 0 \le k \le p \end{matrix}$$

$$\phi_n(i,k) = \sum_{m=0}^{N-1-(i-k)} s_n(m)s_n(m+i-k) \qquad \begin{matrix} 1 \le i \le p \\ 0 \le k \le p \end{matrix}$$

Gọi $R_n(k)$ là hàm tự tương quan dạng:

$$R_n(k) = \sum_{m=0}^{N-1-k} s_n(m)s_n(m+k) \tag{2.15}$$

Do $R_n(k)$ là hàm chẵn nên:

$$\phi_n(i,k) = R_n(|i-k|) \qquad i = 1,2,\ldots p;\ k = 0,1,2,\ldots p$$

Do đó:

$$\sum_{k=1}^{p} \alpha_k r_n(|i-k|) = R_n(i) \tag{2.16}$$

Hệ phương trình này có thể viết dưới dạng ma trận:

$$\bar{\alpha} = \bar{R}^{-1}\bar{r} \tag{2.17}$$

Trong đó:

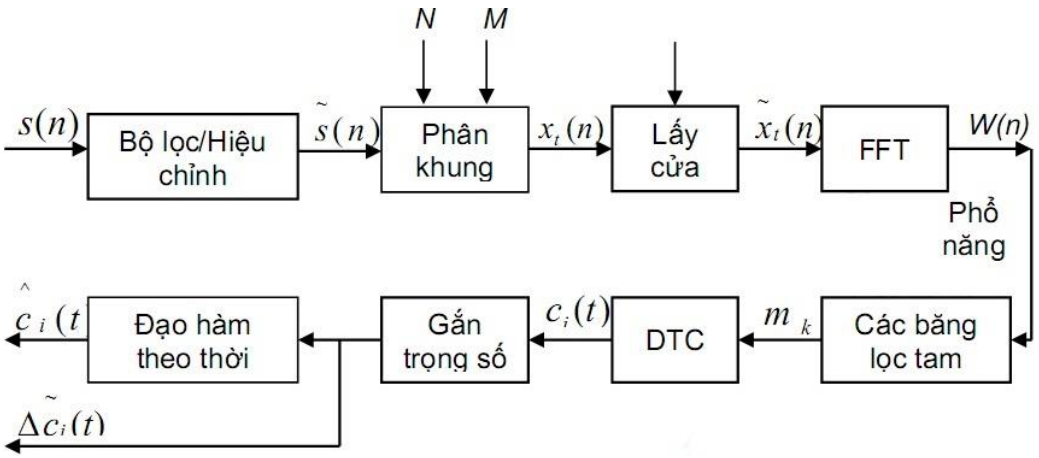
$$\underline{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_p \end{bmatrix} \quad \underline{R} = \begin{bmatrix} r(0) & r(1) & \dots & r(p-1) \\ r(1) & r(0) & \dots & r(p-2) \\ \dots & \dots & \dots & \dots \\ r(p-1) & r(p-2) & & r(0) \end{bmatrix} \quad \underline{r} = \begin{bmatrix} r(1) \\ r(2) \\ \dots \\ r(p) \end{bmatrix}$$

Chú ý: R là ma trận đối xứng. Tất cả các phần tử thuộc đường chéo của ma trận này đều có giá trị bằng nhau, điều đó có nghĩa là nghịch đảo của nó luôn tồn tại và có nghiệm.

2.4.2. Phân tích cepstral theo thang đo mel:

Davis và Mermelstein đưa ra thuật ngữ hệ số cepstral theo tần số Mel (Mel-frequency Cepstral Coefficients - MFCC) vào năm 1980 khi họ kết hợp các bộ lọc cách khoảng không đều với biến đổi Cosine rời rạc (DCT) thành một thuật toán đầu-cuối ứng dụng trong lĩnh vực nhận dạng tiếng nói liên tục. Phương pháp tính các hệ số MFCC là phương pháp trích chọn tham số tiếng nói được sử dụng rộng rãi bởi tính hiệu quả của nó thông qua phân tích cepstral theo thang đo mel.

Phương pháp được xây dựng dựa trên sự cảm nhận của tai người đối với các dải tần số khác nhau. Với các tần số thấp (dưới 1000 Hz), độ cảm nhận của tai người là tuyến tính. Đối với các tần số cao, độ biến thiên tuân theo hàm logarit. Các băng lọc tuyến tính ở tần số thấp và biến thiên theo hàm logarit ở tần số cao được sử dụng để trích chọn các đặc trưng âm học quan trọng của tiếng nói. Mô hình tính toán các hệ số MFCC được mô tả như (Hình 2.4).



Hình 2.4-Sơ đồ tính toán các hệ số MFCC

Ý nghĩa và phương pháp xác định tham số ở các khối trong sơ đồ trên mô tả như sau:

Khối 1: Bộ lọc hiệu chỉnh (Pre-emphasis)

Tín hiệu tiếng nói s(n) được đưa qua bộ lọc số bậc thấp để phổ đồng đều hơn, giảm ảnh hưởng gây ra cho các xử lý tín hiệu sau này. Thường bộ lọc này cố định bậc một, có dạng:

$$H(z) = 1 - az\text{-}1 \quad 0.9 \leq a \leq 1.0 \qquad (2.18)$$

Quan hệ giữa tín hiệu ra với tín hiệu vào tuân theo phương trình

$$\tilde{s}(n) = s(n) - a.s(n-1) \qquad (2.19)$$

Giá trị a thường được chọn là 0,97

Khối 2: Phân khung (Frame Blocking)

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.

Khối 3: Lấy cửa sổ (Windowing)

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), $0 \le n \le N-1$

Thông thường, của sổ Hamming được sử dụng. Cửa sổ này có dạng:

$$w(n) = 0.54 - 0.46\cos(\frac{2\pi n}{N-1}) \qquad 0 \le n \le N - 1 \qquad (2.20)$$

Khối 4: Biến đổi Fourier rời rạc (FFT)

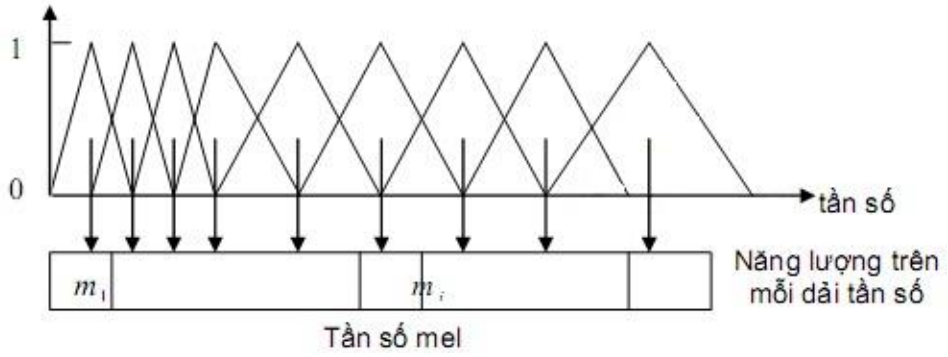
Tác dụng của FFT là chuyển đổi mỗi khung với N mẫu từ miền thời gian sang miền tần số. FFT là thuật toán tính DFT nhanh. DFT được xác định

$$x(k) = \sum_{k=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}nk} \qquad (2.21)$$

Khối 5: Biến đổi sang thang đo Mel trên miền tần số

Như đã nói ở trên, tai người không cảm nhận sự thay đổi tần số của tiếng nói tuyến tính mà theo thang Mel. Người ta chọn tấn số 1kHz, 40 dB trên ngưỡng nghe là 1000 Mel. Do đó, công thức gần đúng biểu diễn quan hệ tần số ở thang mel và thang tuyến tính như sau:

$$mel(f) = 2595 * \log(1 + \frac{f}{700}) \qquad (2.22)$$



Hình 2.5- Các băng lọc tam giác theo tần số Mel

Một phương pháp để chuyển đổi sang thang mel là sử dụng băng lọc (Hình 2.5), trong đó mỗi bộ lọc có đáp ứng tần số dạng tam giác. Số băng lọc sử dụng thường trên 20 băng. Thông thường, người ta chọn tần số từ 0 đến Fs/2 (Fs là tần số lấy mẫu tiếng nói). Nhưng cũng có thể một dải tần giới hạn từ LOFREQ đến HIFREQ sẽ được dùng để lọc đi các tần số không cần thiết cho xử lý. Chẳng hạn, trong xử lý tiếng nói qua đường điện thoại có thể lấy giới hạn dải tần từ LOFREQ=300 đến HIFREQ=3400.

Sau khi tính FFT ta thu được phổ tín hiệu $S(f_n)$. Thực chất đây là một dãy năng lượng $W(m)=|s(fn)|^2$. Cho W(m) đi qua một dãy K băng lọc dạng tam giác, ta được một dãy các $\tilde{w}(n)$. Tính tổng của các dãy $\tilde{w}(n)$ trong từng băng lọc, ta thu được một dãy các hệ số $m_k(k=1,2,3\ldots,k)$.

Khối 6: Biến đổi Cosine rời rạc (DCT) Trong bước này ta sẽ chuyển log của các giá trị m_k về miền thời gian bằng cách biến đổi Cosine rời rạc (DCT). Kết quả của phép biến đổi này ta thu được các hệ số MFCC.

$$w(n) = 0.54 - 0.46\cos(\frac{2\pi n}{N-1}) \quad 0 \le n \le N\text{-}1 \qquad (2.23)$$

Thông thường, chỉ có một số giá trị đầu tiên của c_i được sử dụng. Trong các ứng dụng nhận dạng tiếng nói, người ta thường lấy 12 hệ số MFCC và

thêm 1 hệ số năng lượng của khung sau khi đã được chuẩn hóa làm tham số đặc trưng cho tín hiệu tiếng nói (như vậy tổng cộng có Q = 13 hệ số).

Khối 7: Cepstral có trọng số

Vì độ nhạy của các hệ số cepstral bậc thấp làm cho phổ toàn bộ bị đổ dốc, độ nhạy của các cepstral bậc cao gây ra nhiễu nên người ta thường sử dụng cửa sổ cepstral để cực tiểu hóa độ nhạy này. Công thức biểu diễn các hệ số cepstral có trọng số:

$$\hat{c}_i = \left[1 + \frac{Q}{2}\sin\left(\frac{\pi i}{Q}\right)\right] c_i \quad 1 \le i \le Q \qquad (2.24)$$

Khối 8: Lấy đạo hàm các hệ số MFCC theo thời gian

Để nâng cao chất lượng nhận dạng, người ta đưa thêm các giá trị đạo hàm theo thời gian của các giá trị hệ số MFCC vào vector hệ số tiếng nói. Các giá trị đó được tính theo:

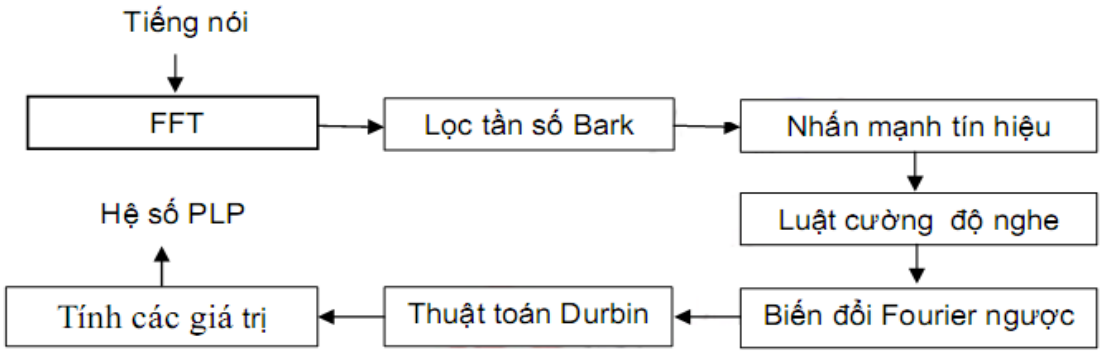
$$\Delta\hat{c}_i = \frac{\sum_{\theta=0}^{\theta} \theta(\hat{c}_{t+\theta} - \theta\hat{c}_{t-\theta})}{2\sum_{\theta=0}^{\theta}\theta^2} \quad 1 \le i \le Q \qquad (2.25)$$

Trong đó; θ: là độ dài cửa sổ tính delta (thường chọn là 2 hoặc 3).

Kết thúc các bước trên với mỗi khung ta thu được một vector có 2Q thành phần biểu diễn tham số đặc trưng của tiếng nói.

2.4.3. Phương pháp PLP

PLP (Packet Level Protocol): Giao thức chuyển mạch gói. Phương pháp này là sự kết hợp của hai phương pháp đã trình bày ở trên. Hình 2.6 mô tả các bước xác định hệ số PLP.



Hình 2.6- Sơ đồ các bước xác định hệ số PLP

Các khối xử lý

Khối 1: Biến đổi Fourier nhanh (FFT)

Tương tự như phương pháp MFCC, tín hiệu tiếng nói được chia thành các khung và được chuyển sang miền tần số bằng thuật toán FFT.

Khối 2: Lọc theo thang tần số Bark

Tín hiệu tiếng nói được lọc qua các bộ lọc phân bố theo thang tần số phi tuyến, trong trường hợp này là thang tần số Bark:

$$\mathrm{Bark}(f) = 6\ln\left\{\frac{f}{1200} + \left[\left(\frac{f}{1200}\right)^2 + 1\right]^{\frac{1}{2}}\right\} \tag{2.26}$$

Khối 3: Nhấn mạnh tín hiệu dùng hàm cân bằng độ ồn (equal-loudnes)

Bước này tương tự bước nhấn mạnh (preemphais) của phương pháp MFCC. Hàm này mô phỏng đường cong cân bằng độ ồn (Equal-Loudnes Curve)

$$E(\omega) = \frac{(\omega^2 + 56{,}8*10^6)\omega^4}{(\omega^2 + 6.3*10^6)(\omega^6 + 9.58*10^{26})} \tag{2.27}$$

Khối 4: Dùng luật cường độ nghe (Power Law of Hearing)

Bước xử lý này giống như bước lấy giá trị logarit trong phương pháp MFCC. Hàm căn lập phương được dùng có dạng:

$$\Phi(f) = \psi(f)^{0.33} \tag{2.28}$$

Khối 5: Biến đổi Fourier ngược (Inverse DFT)

Các hệ số tự tương quan được biến đổi Fourier ngược là giá trị đầu vào cho LPC.

Khối 6: Thuật toán Durbin

Thuật toán Durbin được sử dụng để tính các hệ số dự báo tuyến tính như phương pháp LPC

Khối 7: Tính các giá trị delta

Phương pháp tính tương tự như phương pháp hệ số MFCC.

2.4.4. Biến đổi Fourier rời rạc

Tín hiệu (của một frame) sau khi nhân với hàm cửa sổ, được chuyển sang miền tần số bằng biến đổi Fourier rời rạc:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{i2\pi kn}{N}}; k = 0...N-1 \tag{2.29}$$

2.4.5. Logarit và biến đổi Fourier ngược

Lấy logarit của tín hiệu ở miền tần số (spectrum) rồi biến đổi Fourier ngược sẽ đưa tín hiệu về một miền gọi là cepstrum có đơn vị thời gian (thuật ngữ là cepstrum đảo ngược của âm đầu tiên trong từ spectrum: spectrum → cepstrum). Biến đổi từ spectrum sang cepstrum là một biến đổi đồng hình (homomorphic). Theo biến đổi đồng hình chuyển biểu diễn tín hiệu từ dạng tích về dạng tổng, như vậy cho phép sử dụng các hệ tuyến tính để xử lí các tín hiệu không tuyến tính.

Công thức tính của bước này là:

$$c(n) = \sum_{k=0}^{M} \log(F(k)) \cos\left(\frac{2i\pi n(k-1)}{2M}\right); N = 1...P \quad (2.30)$$

Chú ý: mặc dù biến đổi từ spectrum sang cepstrum là biến đổi Fourier ngược, tuy nhiên do ta dùng spectrum và cepstrum thực nên chỉ sử dụng biến đổi cosine rời rạc (DCT) để tăng hiệu năng tính toán.

Sau bước này ta được vector cepstral (ở độ đo mel) p thành phần. Thông thường người ta thường nhân thêm vào kết quả một hàm cửa sổ sóng sin (gọi là thủ tục liftering) để giảm bớt ảnh hưởng của các biến đổi đến kết quả.

$$w(n) = 1 + \frac{L}{2} \sin\left(\frac{n\pi}{L}\right) \quad (2.31)$$

$$c(n) = c(n)\, w(n) \quad (2.32)$$

3.1.6. Tính toán năng lượng

Kèm thêm thông tin về năng lượng của tín hiệu sẽ tăng thêm thông tin cho nhận dạng (ví dụ: phân biệt các khoảng chứa tín hiệu âm và khoảng lặng, phân biệt vùng tín hiệu chứa nguyên âm và phụ âm…)

Năng lượng của cả frame được tính qua công thức:

$$E = \sum_{n=0}^{N-1} (x(n))^2 \quad (2.33)$$

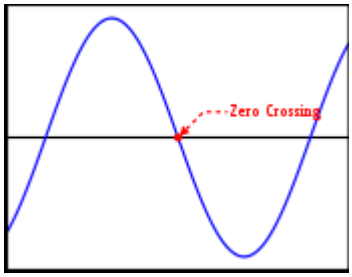
Chương 3

TRÍCH RÚT ĐẶC TRƯNG CỦA TIẾNG NÓI DỰA VÀO DÃY ĐIỂM CẮT ZERO

3.1. Điểm cắt Zero

3.1.1. Khái niệm về điểm cắt Zero

Zero-crossing là một khái niệm được sử dụng phổ biến trong kỹ thuật điện, toán học và xử lý ảnh. Trong các khái niệm toán học, "zero-crossing" là điểm mà ở đó hàm số đổi dấu, ví dụ từ dương sang âm và được biểu diễn bằng điểm cắt trên hoành độ.



Hình 3.1- Điểm cắt Zero biểu thị tương quan giữa điện áp và thời gian

3.1.2. Đường mức không

Tiếng nói hay âm thanh là tín hiệu dao động, giả sử các giá trị tín hiệu nhận từ 0 đến L-1, khi im lặng các tín hiệu này sẽ nhận giá trị L/2, ví dụ mẫu 8 bit có L = 256 thì mức không là 128, giá trị này là giá trị không. Thực tế khi thu âm, soundcard thực hiện số hóa âm thanh có thể mức không không là giá trị nói trên. Để xác định đường mức không thực hiện như sau:

Bước 1. Lấy mức thu của soundcard.

Bước 2. Đặt mức thu là nhỏ nhất.

Bước 3. Thu khoảng 3 giây.

Bước 4. Tính Histogram H của đoạn tín hiệu thu được.

Bước 5. Xác định giá trị mà có histogram là lớn nhất, vị trí đó sẽ là giá trị đường mức không.

Khi đã xác định được đường mức không, giá trị tín hiệu tiếng nói sẽ dao động xung quanh đường mức không.

3.1.3. Ứng dụng điểm cắt Zero trong xử lý ảnh

Trong lĩnh vực xử lý ảnh, một vấn đề quan trọng đó là việc tìm ra biên của ảnh và khái niệm zero-crossing được sử dụng trong việc tìm ra biên giới ảnh và trong bộ lọc gradient. Bộ lọc gradient là một bộ lọc tìm ra các vùng có sự thay đổi đột ngột về giá trị điểm ảnh. Những điểm ảnh này thường được xem như là đường biên của ảnh. Bộ lọc Laplace cũng là một bộ lọc tuân theo nguyên tắc này, bộ lọc này tìm ra các điểm mà tại đó tín hiệu số của ảnh đi qua một giá trị 0 được thiết lập trước, điểm này sẽ được đánh dấu như là một điểm biên tiềm năng. Bởi vì tín hiệu đi qua điểm có giá trị 0, nên điểm này còn được gọi là "zero-crossing".

3.2. Hệ số tương quan và ứng dụng của nó

Do tín hiệu âm thanh có tính tuần hoàn, khi sử dụng điểm cắt Zero ta thu được dữ liệu rời rạc, để tìm đặc trưng từ tập dữ liệu đó ta phải tìm các dãy dữ liệu lặp và so sánh các dãy tín hiệu này có xấp xỉ giống nhau hay không nên ta sử dụng hệ số tương quan để so sánh.

Có thể sử dụng nhiều công thức tính hệ số tương quan khác nhau cho những tình huống khác nhau. Hệ số tương quan được biết đến nhiều nhất là hệ số tương quan Pearson được tính bằng cách chia hiệp phương sai (covariance) của hai biến với tích độ lệch chuẩn (standard deviation) của chúng. Cách tính này được đưa ra trước tiên bởi Francis Galton.

Gọi x_i và y_i là hai biến quan sát được của X và Y cho cá nhân i. Giả sử chúng ta có n đối tượng i = 1,2,3…n. Gọi $\bar{x}$ và $\bar{y}$ là hai số trung bình của biến

quan sát được x và y; s^2_x và s^2_y lần lượt là phương sai của hai biến, được định nghĩa như sau:

$$s_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2 \quad (3.1)$$

và

$$s_y^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2 \quad (3.2)$$

Do đó, nếu X và Y độc lập chúng ta có thể viết:

$$s_{x+y}^2 = s_x^2 + s_y^2 \quad (3.3)$$

Nhưng X và Y có liên hệ với nhau, công thức trên không đáp ứng được vấn đề mô tả. Chúng ta cần tìm một chỉ số mô tả khác mô tả giữa hai biến, bằng cách nhận độ lệch của biến x từ số trung bình $(x_i - \bar{x})$, cho độ lệch của biến y, $(y_i - \bar{x})$ thay vì bình phương độ lệch từng biến riêng lẻ như công thức (3.3). Nói cách khác, tính số hai độ lệch chính là hiệp biến. Đối với mỗi cá nhân hiệp biến là:

$$\mathrm{Cov}\left(x_i, y_i\right) = (x_i - \bar{x})(y_i - \bar{y}) \quad (3.4)$$

Nhưng ở đây chúng ta có n đối tượng, cho nên cần phải cộng tất cả lại và chia cho số đối tượng:

$$\mathrm{Cov}\left(x_i, y_i\right) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \quad (3.5)$$

Công thức (3.5) chính là định nghĩa hiệp biến. Từ hai công thức trên chúng ta có những nhận xét sau:

- Phương sai lúc nào cũng phải là số dương, bởi vì chúng tính toán từ bình phương, nhưng hiệp biến cũng có thể âm hoặc dương vì được ước tính tích của hai độ lệch.

- Một hiệp biến là số dương có nghĩa là độ lệch từ số trung bình của x tuân theo chiều hướng thuận của y.

- Một hiệp biến là số âm có nghĩa là độ lệch từ số trung bình của x tuân theo chiều hướng nghịch của y.

- Nếu hiệp biến bằng 0, thì hai biến x và y độc nhau, tức không có tương quan gì với nhau.

Một cách để "chuẩn hoá" hiệp biến và phương sai là lấy tỉ số của hai chỉ số này và đó chính là định nghĩa hệ số tương quan. Hệ số tương quan được kí hiệu là r:

$$r = \frac{Cov(x,y)}{\sqrt{var(x).var(y)}} = \frac{Cov(x,y)}{s_x s_y} \qquad (3.6)$$

(Căn số bậc hai của phương sai là độ lệch chuẩn: $s_x = \sqrt{s_x^2}$ và $s_y = \sqrt{s_y^2}$, cho nên công thức trên được mô tả bằng độ lệch chuẩn, thay vì phương sai).

Công thức (3.6) có thể viết lại như sau:

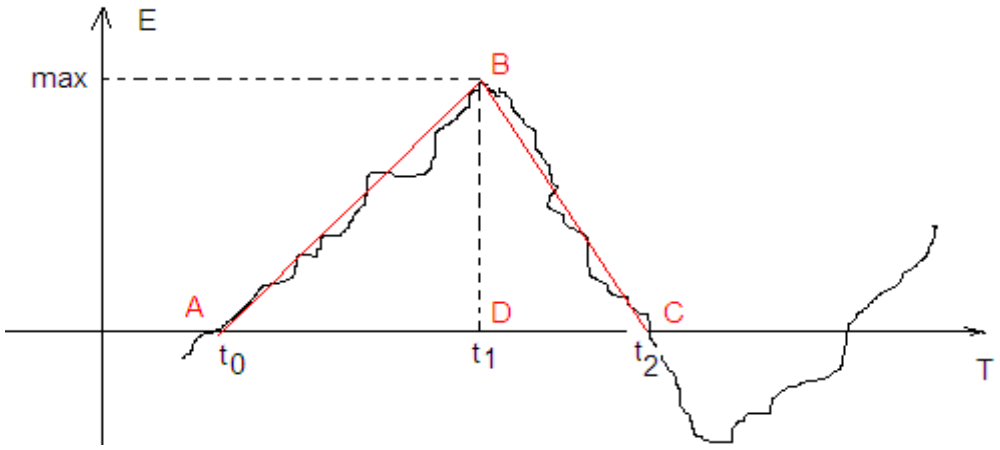
$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} = \frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{2}\left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{(1-n)s_x s_y} \qquad (3.7)$$

3.3. Trích rút đặc trưng

Nhằm đưa thêm vào hệ thống cách trích chọn đặc trưng tiếng nói nên đề tài này quan tâm chủ yếu cách trích rút đặc trưng bằng phương pháp dựa vào điểm cắt zero và áp dụng hệ số tương quan.

Chúng ta coi đường cong tạo bởi các tín hiệu của âm thanh là đường hình sin liên tục theo thời gian t, khi đó điểm cắt zero là điểm đường cong cắt

trục T. Thay cho việc lưu giữ giá trị của các tín hiệu trên cung ABC chúng ta chỉ lưu thông tin về tam giác ABC.



Hình 3.2- Hình mô tả điểm cắt zero - cross

Thông tin về tam giác ABC gồm:

- Độ dài cạnh AC được đo bằng $x= t_2-t_0$

- Vị trí cực đại trên cung ABC $y=t_1-t_0$

- Giá trị cực đại max của tín hiệu kí hiệu là z

Khi đó quá trình trích đặc trưng là đưa file âm thanh về file text mà mỗi khoảng không điểm ứng với bộ ba tham số < x,y,z>

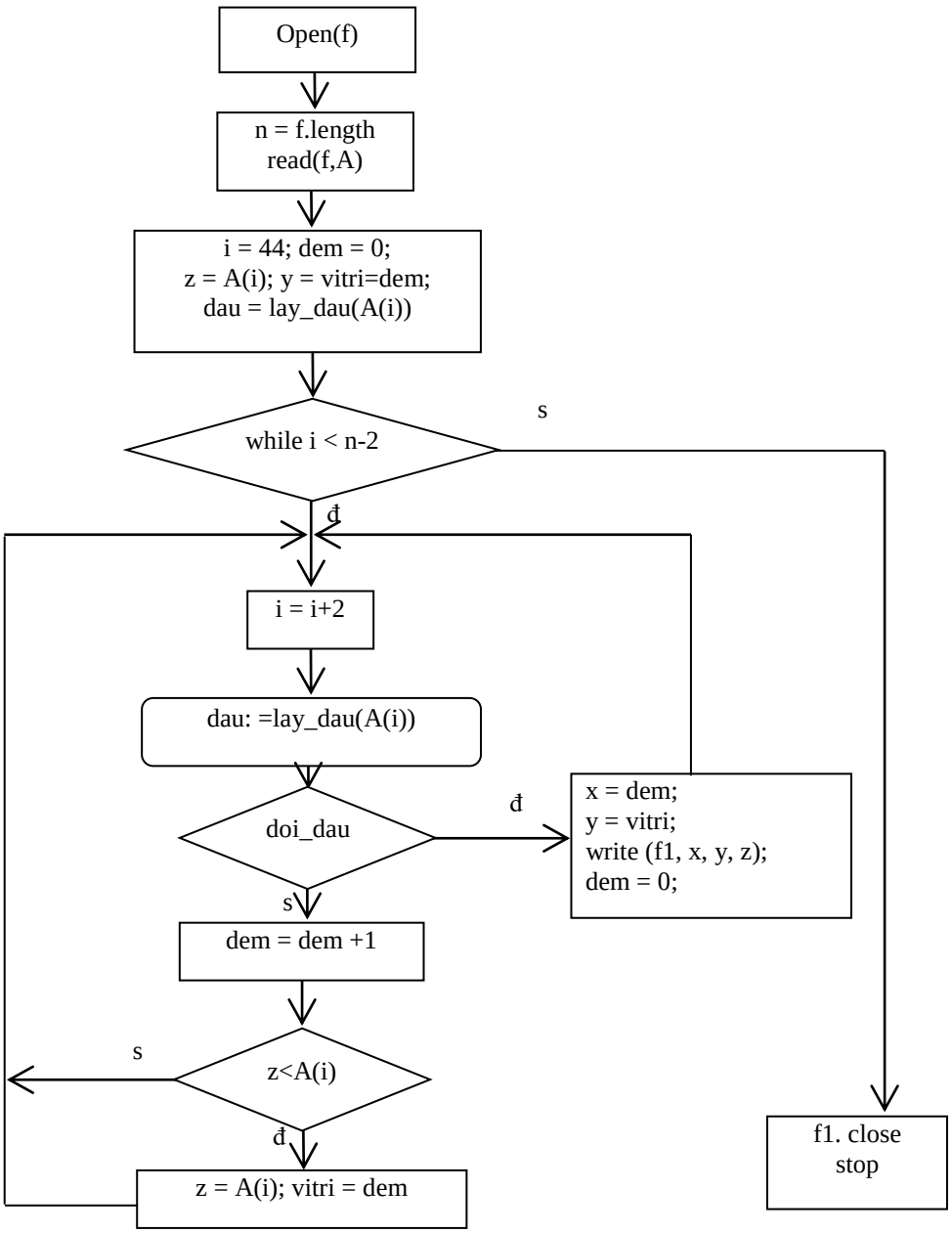
3.3.1. Thuật toán xác định dãy không điểm

Đầu vào: Dãy tín hiệu tiếng nói

Đầu ra: File text gồm một dãy các số mà mỗi khoảng không điểm ứng với bộ ba x,y,z tương ứng với số mẫu trong đoạn đang xét; vị trí mẫu đạt giá trị lớn nhất (nhỏ nhất) đó.

Đặt n = độ dài file f.wave, dùng mảng A để đọc dữ liệu tiếng nói từ file f. Duyệt từ đầu đến cuối mảng A, xét dấu của từng tín hiệu, nếu có đổi dấu nghĩa là xuất hiện điểm cắt zero trên trục T. Trong đoạn giữa các không điểm

này, tìm z = max {A(i)}, y = vị trí đạt max {A(i)}, và x = Số điểm lấy mẫu trong đoạn, lưu bộ 3 giá trị này vào file f1. Tiếp tục thực hiện như trên cho đến khi hết file f, file f1.txt nhận được sẽ là file <x,y,z> của file f.wave.



Hình 3.3- Sơ đồ mô tả thuật toán tạo ra dãy {x,y,z}

Các biến được sử dụng trong thuật toán trên được mô tả trên Hình 3.3:

dau: nhận giá trị - hoặc + để nhận biết dãy tín hiệu đổi dấu có nghĩa là

tín hiệu có cắt trục 0 (có điểm cắt zero).

A: lưu giá trị tín hiệu

x: lưu số mẫu của một bước sóng.

y: vị trí mẫu đạt giá trị lớn nhất (nhỏ nhất)

z: giá trị lớn nhất (nhỏ nhất) của tín hiệu

n: số mẫu trong một dãy tín hiệu

Dem: biến trung gian đếm số mẫu trong một bước sóng (giới hạn bởi 2 điểm cắt zero).

File f: chứa dữ liệu tiếng nói đầu vào và **File f1**: chứa dữ liệu file text đầu ra.

3.3.2. Thuật toán tìm các dãy lặp

Từ file dữ liệu dạng wav ta được dãy {xi, yi, zi} i = 1,2...n. Vấn đề là cần phải tìm các đặc trưng cho dãy {xi, yi, zi}.

Dựa vào tính tuần hoàn của song âm thanh ta suy ra $\{x_i, y_i, z_i\}$ phải chứa các dãy con lặp lại.

Trong dãy {x,y,z} có giá trị z là độ lớn biên độ của tiếng nói, điều này đồng nghĩa với việc có thể thay đổi giá trị z để âm thanh có thể to hay nhỏ đi. Vậy dãy lặp cần tìm có tính đặc trưng chỉ là tập <x,y>.

Từ tập {x,y,z} ta tách ra tập {x}, sau đó từ tập {x} ta lọc ra các phần tử tập $\{X_{max}\}$ có giá trị từ a trở lên đến b.

Cách thức trích ra dãy $\{X_{max}\}$

(1) Từ tập dãy {x,y,z} ta rút ra dãy {x}

(2) Ta đánh chỉ số từ 1 đến n cho dãy {x}, ta lấy ra mảng $\{X_{max}\}$ có giá trị giảm dần từ 100 đến 6, ứng từng giá trị X_{max} ta đánh chỉ số id cho nó chính

là bằng chỉ số thứ tự trong dãy {x}.

Thuật toán phát hiện ra các dãy lặp:

Đầu vàoV: Dãy các giá trị $\{X_{max}\}$, {X}

Đầu ra: Tập các dãy lặp { V_1, V_2, ...V_k}. Trong đó: $V_i = \{x_{i1}, x_{i2}, ... x_{ip}\}$

Mô tả thuật toán như sau:

Begin

Read(X_{max}[i], X}

i: = 0;

B1: id[1] = id[i]; Max = X_{max}[i];

l: = 0;

B2: dem = 0;

p: = 4;

delta: = p;

While ((Max – X(id1 + delta) <= 1) and (delta + id1 <= X.lenght)

Begin

dem: = dem + 1;

delta: = delta + p;

End;

If (dem >= 4) then

Write(kq[l] = “id” and “p”); l: = l + 1;

Else

P: = p + 1;

If (p<= 20) then Goto: B2;

Else

i: = i + 1;

if (i <= X_{max}.lenght) then Goto: B1

else Stop

End

Giải thích thuật toán:

Khi đưa mảng Xmax, mảng X, thuật toán kiểm tra thoả mãn các điều kiện là:

- $4 \leq p \leq 20$ (Số lượng phần tử thuộc dãy lặp)
- Max - X[id + p] ≤ 1 (Sai khác 1 giá trị)
- Dem ≥ 4 (Số các dãy lặp lớn hơn hoặc bằng 4)

Thoả mãn các điều kiện trên ta đưa ra kết quả là: "id" và "p", dựa vào "id" và "p" ta chiếu vào mảng X,Y lấy ra các giá trị của những dãy lặp.

Các biến được sử dụng trong thuật toán phát hiện ra các dãy lặp:

i: Chỉ số các phần tử của dãy X_{max}

l: chỉ số của mảng kq[l] = id,p

Xmax[i]: Dãy các các số lớn hơn hoặc bằng m và nhỏ hơn hoặc bằng n.

id: Chỉ số các phần tử X_{max} trong với dãy các phần tử thuộc X.

p: Khoảng lặp lại.

dem: Biến đếm trong quá trình phát hiện các dãy lặp lai.

delta: Biến gán ứng giá trị p.

X.length: Độ dài của dãy các phần tử thuộc X.

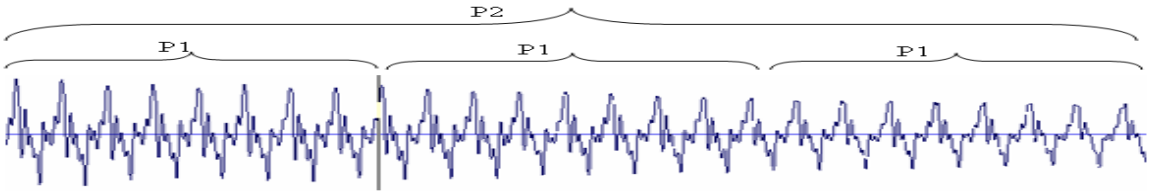
Xmax.length: Độ dài của dãy các phần tử thuộc Xmax.

Kq[l]: Kết quả của thuật toán.

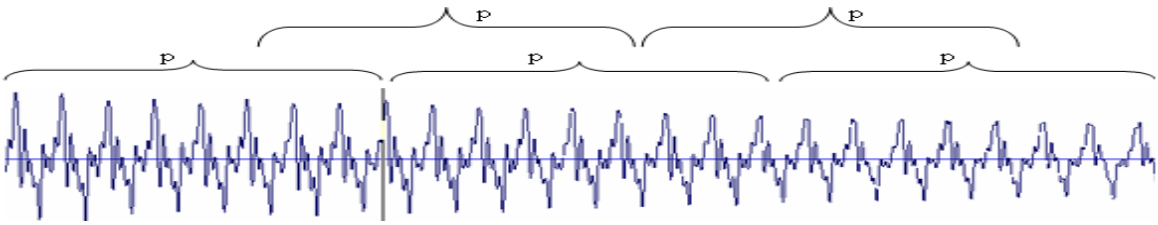
3.3.3. Phương pháp rút gọn trích chọn đặc trưng

Từ tập dãy lặp ta thực hiện các bước sau ra được tập dãy có tính đặc trưng:

(1) Loại dãy dữ liệu con thuộc dãy lớn: Giả sử P1 < P2 thì loại Xip1 nếu Xip1 thuộc Xip2.



(2) Loại dãy dữ liệu nếu thoả mãn điều kiện: id2 = id1 + (dem1 – dem2)*p với điều kiện có số phần tử bằng nhau (cùng p)



(3) Loại dãy dữ liệu tương quan với nhau: Giả sử {x11, x12…, x1p},{x'11, x'12…, x'1p} có hệ số tương r ≥ 0.9 thì loại dãy sau đi với điều kiện có số phần tử bằng nhau (cùng p).

(4) Sắp xếp lại theo "id" tăng dần.

3.4. Xây dựng thuật toán nhận dạng

Đầu vào: Phụ âm dạng file wav.

Đầu ra: Kết quả nhận dạng: Phát âm ra phụ âm đó.

Khi đưa phụ âm cần được nhận dạng ta thu được tập P{ P_1, P_2, ...P_m}

gọi đặc trưng của phụ âm cần được nhận dạng.

Các kí hiệu trong thuật toán:

- Mau[j]: tập dữ liệu đặc trưng của môt phụ âm thuộc không gian mẫu.

- P: Tập đặc trưng của phụ âm cần nhận dạng

- n: số dãy phần tử của một tập Mau[j]

- m: số dãy phần tử của tập P

- i: Chỉ số của khối tập đặc trưng trong bộ dữ liệu thuộc không gian mẫu

- k: Số lượng khối đặc trưng trong bộ dữ liệu hoặc không gian mẫu

- Vitri: Vị trí của khối tập đặc trưng trong bộ dữ liệu mà từ nhận dạng được

- c: Hệ số tin cậy để đạt mức nhận dạng được giữa Mau[j] và P

Thuật toán gồm các bước như sau:

- Tạo lập được bộ dữ liệu đặc trưng hay gọi là "Không gian mẫu".

- Nhập từ cần nhận dạng vào thu được các dãy đặc trưng của mẫu cần nhận dạng

- Chọn hệ số tin cậy "c".

- Sử dụng tính tương quan để đối sánh giữa "Không gian mẫu" và "P", khi giá trị tương quan đạt lớn hơn bằng "c" thì nhận dạng được.

tuongquan: Hàm tính hệ số tương quan giữa đại lượng Mau[j] và P

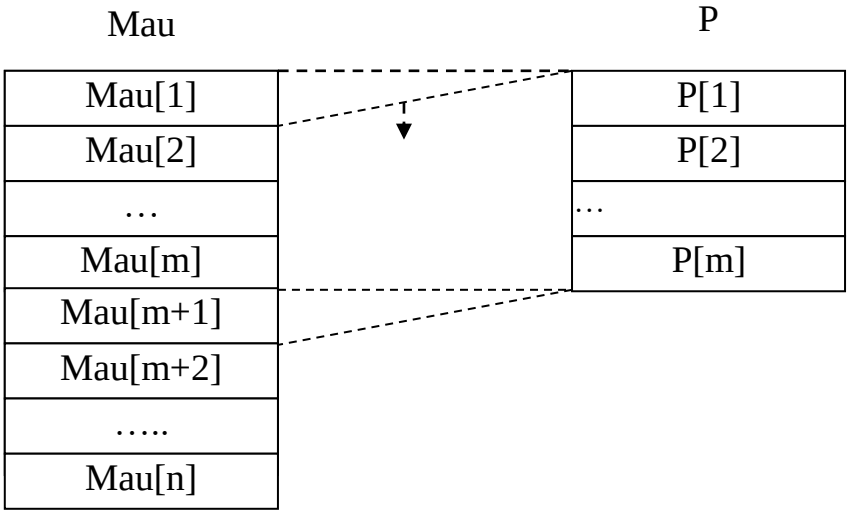
Xét các trường hợp khi tính hàm tuongquan:

n = | Mau[j] | ; m = | P |

- Trường hợp 1: n = m với r_i = tuongquan(Mau[j], P) ≥ 0,9 thì Mau[j] ≡ P;

- Trường hợp 2: n > m

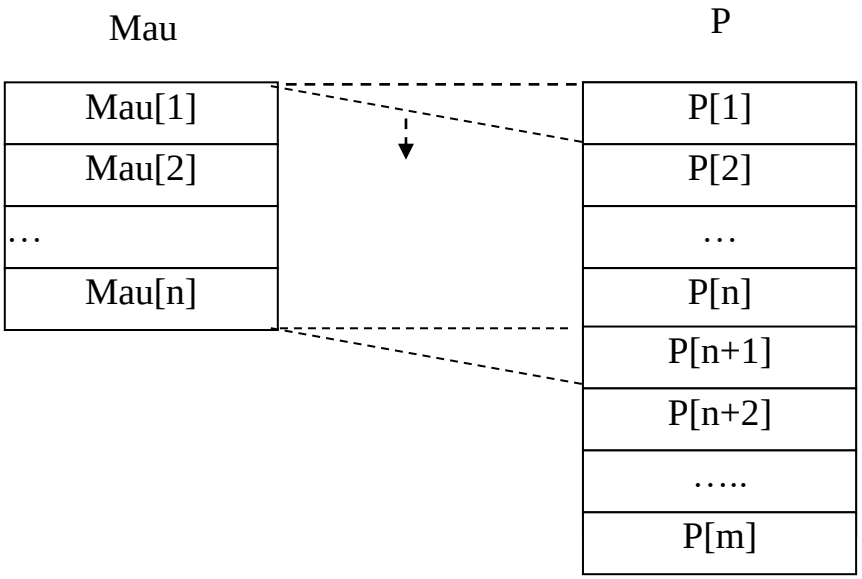
Ta thực hiện cách đối sánh theo mô hình sau:



Hình 3.4- Xét sự tương quan giữa hai mảng trường hợp n > m

- Trường hợp 3: n < m

Ta thực hiện cách đối sánh theo mô hình sau:



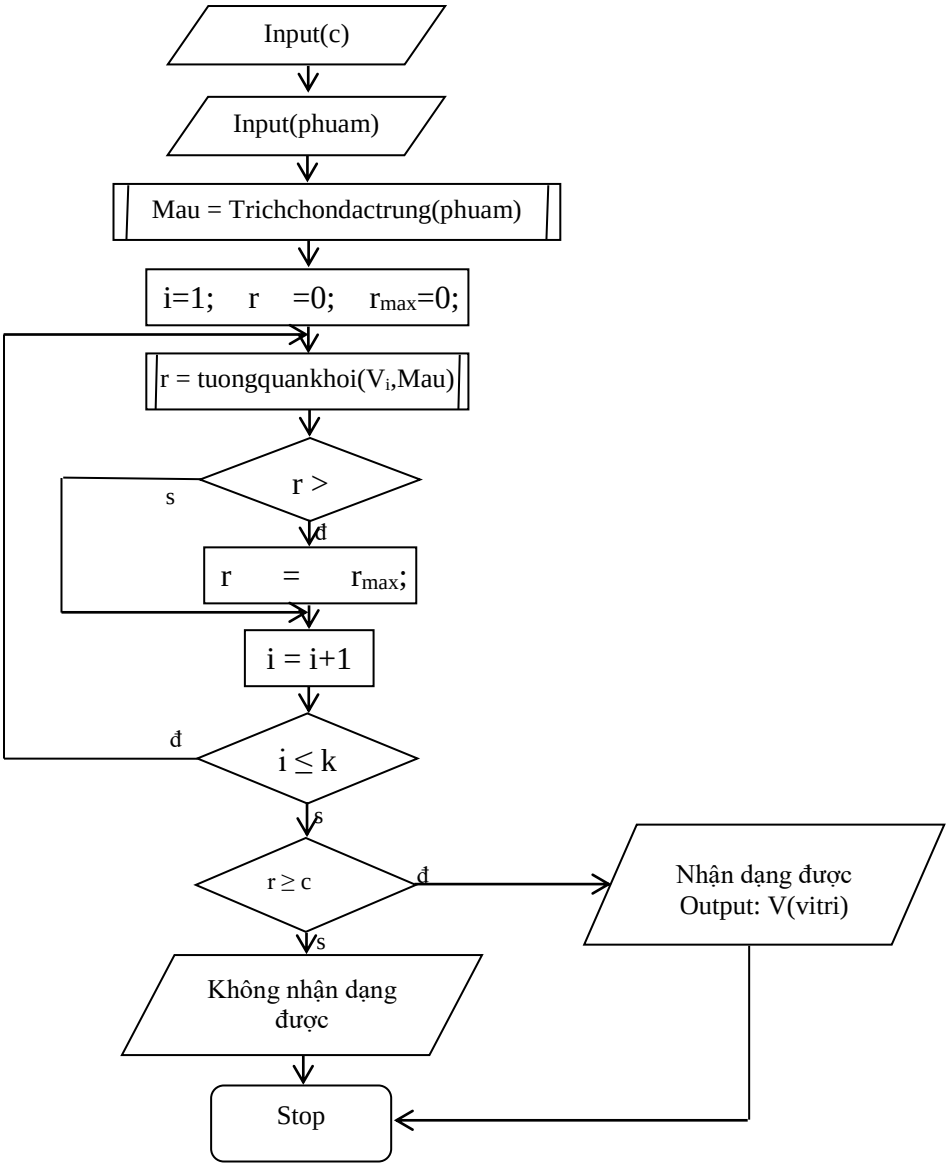
Hình 3.5- Xét sự tương quan giữa hai mảng trường hợp n < m

Tính hệ số tương quan cho cả ba trường hợp trên:

$$r_i = \text{tuongquan(Mau[j], P)} = \frac{\text{sodongtuongquan}}{\max(n, m)}$$

với sodongtuongquan: số dòng tương quan giữa hai mảng.

Ta có sơ đồ khối mô tả thuật toán nhận dạng như sau:



Hình 3.6- Thuật toán nhận dạng

Theo như thuật toán trên: Ta đưa phụ âm dạng file wav cần được nhận dạng thu được tập P{ P_1, P_2, ...P_m} là các dòng mẫu đặc trưng cần được nhận dạng gọi là P, lấy giá lớn nhất của hàm tuongquan(Mau[j], P]) gọi là r_{max}, khi đạt sự tương quan giữa tập Mau[j] và P ta thu được kết qủa nhận dạng của phụ đó.

Chương 4

XÂY DỰNG CHƯƠNG TRÌNH THỰC NGHIỆM

4.1. Mô hình bài toán

4.1.1. Yêu cầu của bài toán nhận dạng

Trong tất cả các bài toán nhận dạng tiếng nói, chất lượng nhận dạng phụ thuộc rất nhiều vào quá trình xác định và rút trích các đặc trưng. Mô hình chung của bài toán nhận dạng như sau:

- Trích được đặc trưng của tín hiệu tiếng nói
- Xây dựng được bộ dữ liệu mẫu hay gọi là không gian mẫu
- Xây dựng được thuật toán đối sánh giữa không gian mẫu và tín hiệu tiếng nói cần nhận dạng.

Nếu khối rút trích đặc trưng tốt, chất lượng nhận dạng sẽ cao hơn.

4.1.2. Chức năng chính của bài toán

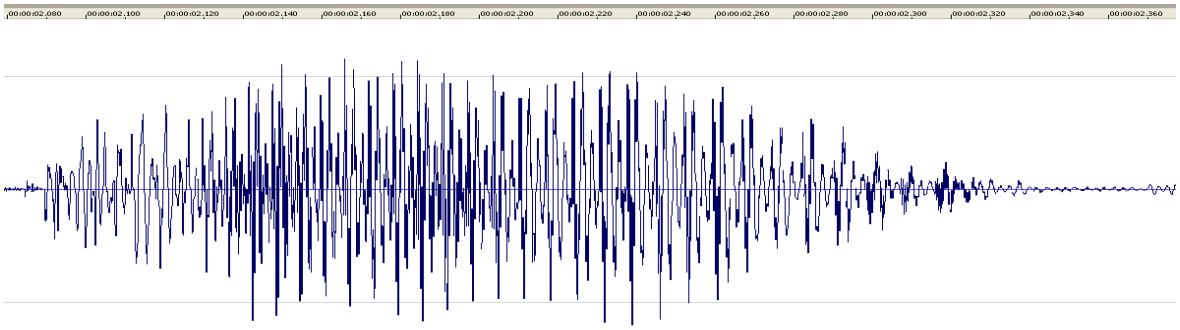
- Mở và đọc file wave của phụ âm vào cơ sở dữ liệu mẫu
- Trích rút đặc trưng của phụ âm Tiếng việt.
- Nhận dạng phụ âm tiếng việt: “c”.
- Yêu cầu dữ liệu đầu vào là tín hiệu tiếng nói chứa trong file wave với tần số lấy mẫu 11025 Hz, độ phân giải 16 bit. Được ghi âm ở chế độ bình thường (nhiễu tự nhiên, không quá lớn).

4.2. Thu file wave của phụ âm “c” và một số phụ âm khác.

Ta sử dụng hệ thống ghi âm, sau đó dùng phần mềm Sound Forge để hiển thị sóng của phụ âm đó.

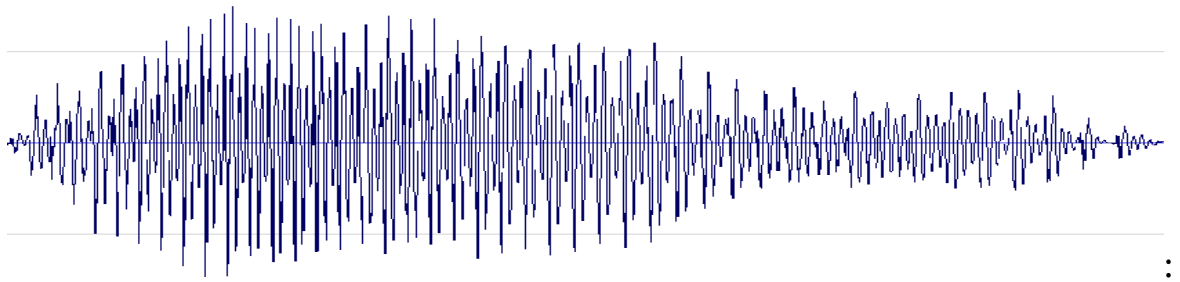
Sóng của phụ âm “c” ghi đối với người nói khác nhau:

- Người thứ nhất:



Hình 4.1- Sóng của phụ âm “c” ghi của người nói thứ nhất

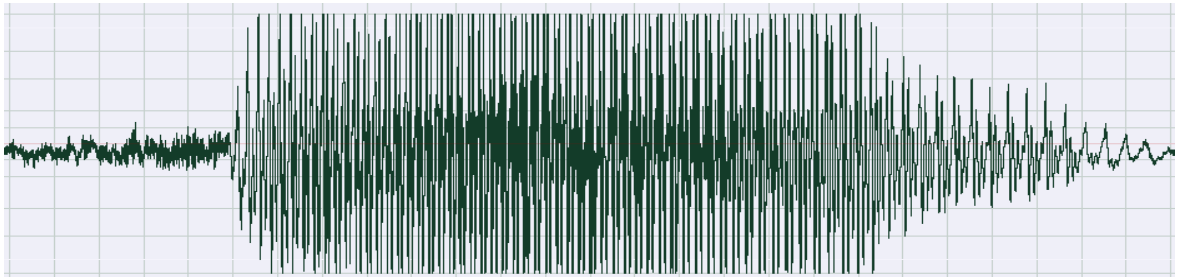
- Người thứ hai:



:

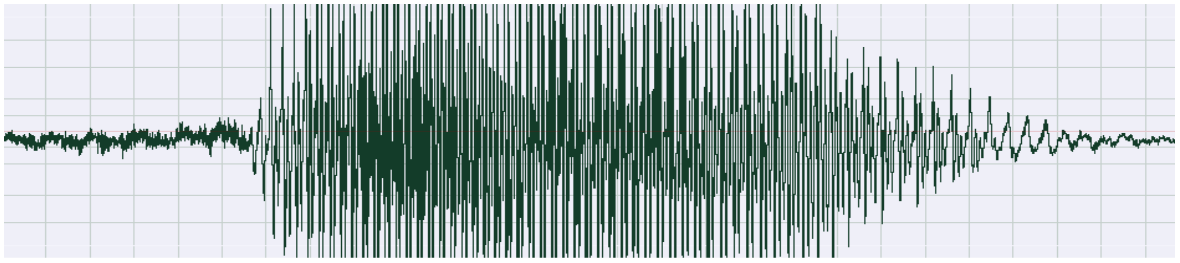
Hình 4.2- Sóng của phụ âm “c” ghi của người nói thứ hai

- Người thứ ba:



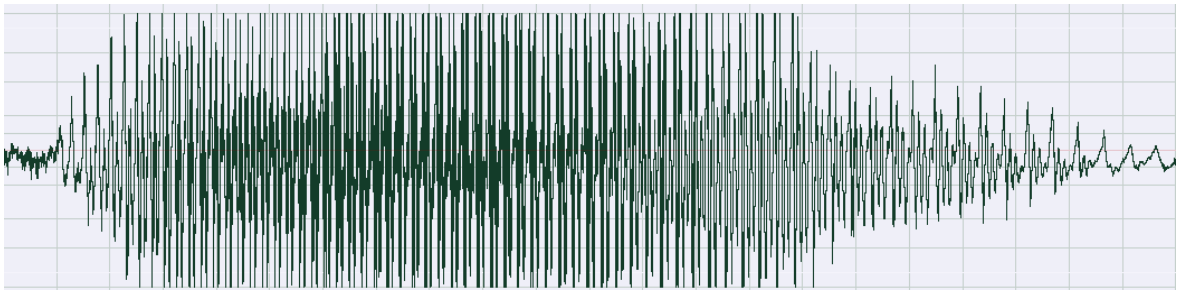
Hình 4.3- Sóng của phụ âm “c” ghi của người nói thứ ba

- Người thứ tư:



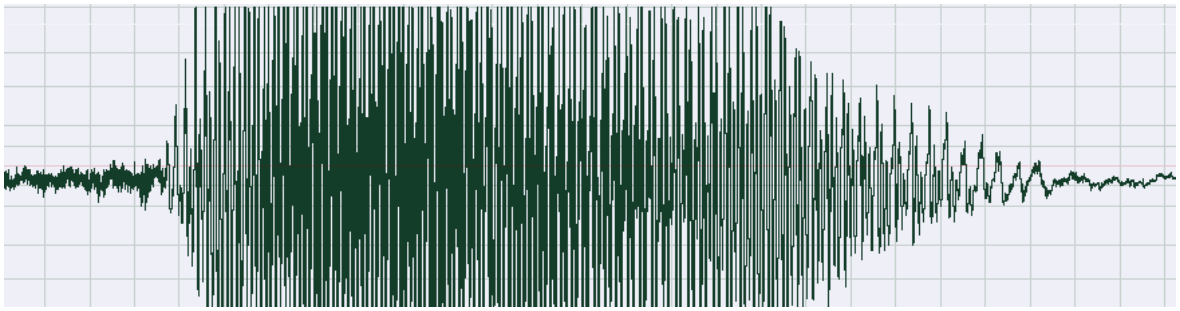
Hình 4.4- Sóng của phụ âm “c” ghi của người nói thứ tư

- Người thứ năm:



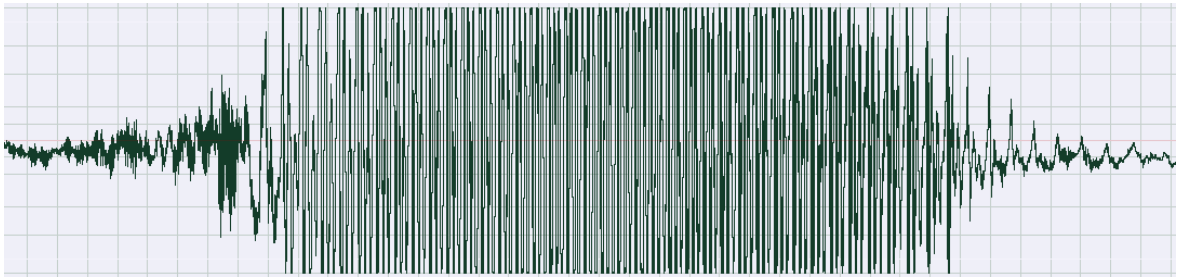
Hình 4.5- Sóng của phụ âm “c” ghi của người nói thứ năm

- Người thứ sáu:



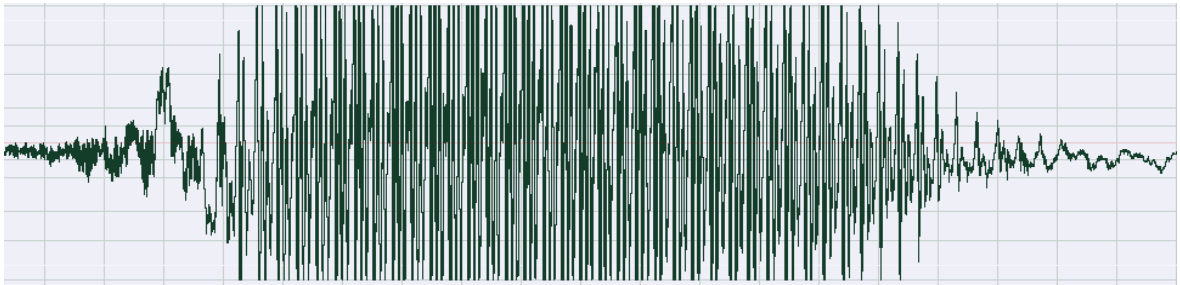
Hình 4.6- Sóng của phụ âm “c” ghi của người nói thứ sáu

- Người thứ bảy:



Hình 4.7- Sóng của phụ âm “c” ghi của người nói thứ bảy

- Người thứ tám:



Hình 4.8- Sóng của phụ âm “c” ghi của người nói thứ tám

4.3. Hàm xác định đặc trưng dựa trên điểm cắt Zero

4.3.1. Hàm xác định tập dãy {x,y,z}

Áp dụng thuật toán xác định tập x,y,z ở mục 3.3.1 với hàm xác định x,y,z như sau:

```
private void file_xyz()
string filename,str1;
string str5 = "  files_wav (*.wav) |*.wav";
//filename = @"c:\thu_am\aaux.wav"; FileStream read_file;
read_file = new FileStream(filename, FileMode.Open,
FileAccess.Read, FileShare.Read);
byte[] buffer = new byte[read_file.Length];
int count = read_file.Read(buffer, 0, buffer.Length);
read_file.Close();
// lấy ra buffer rồi
int m, dem,  dau;
int vitri,max;
int j = 44;
filename = @"c:\thu_am\file_xyz.txt";
using (StreamWriter sw = File.CreateText(filename))
        {
        //-lấy ra hai byte một chuyển sang int
        m = System.BitConverter.ToInt16(buffer, 44);
        dau = ham_dau(m);
        max =Math.Abs( m);
        vitri = 0;
        tr1 = "";
        dem =0;
```

```
while (j < buffer.Length - 2)
        {
        j = j + 2;
        m = System.BitConverter.ToInt16(buffer, j);
        if (ham_dau(m) == dau)
        {
dem = dem + 1;
if (Math.Abs( m)>max )
{ max=Math.Abs( m); vitri = dem; }
}
else
{
max=dau*max ;
str1 = Convert.ToString(dem) + ';' +      Convert.ToString(max)
+ ';' + Convert.ToString(vitri);
sw.WriteLine(Convert.ToString(str1));
dau = ham_dau(m);
max = Math.Abs(m);
vitri = 0;
str1 = "";
dem = 0;
}
}//====het while
MessageBox.Show("kết thúc");
}//hêt using
}
```

4.3.2. Hàm tính hệ số tương quan

Từ các dãy lặp và khoảng lặp “p”. Mục đích là sử dụng hệ số tương quan “r” để so sánh giữa các dãy lặp với khoảng lặp giá trị “p”, với hàm xác định hệ số tương quan như sau:

```
private double heso_tuongquan(int[] x, int[] y, int n)
{
int i;
double r, tongx = 0, tongy = 0, x_ngang, y_ngang, tongxbp = 0,
tongybp = 0, tong_ngang = 0, mauso;
for (i = 0; i < n; i++)
        {
        tongx = tongx + x[i];
        tongy = tongy + y[i];
        }
x_ngang = tongx / n; y_ngang = tongy / n;
for (i = 0; i < n; i++)
        {
        tongxbp = tongxbp + (x[i] - x_ngang) * (x[i] - x_ngang);
        tongybp = tongybp + (y[i] - y_ngang) * (y[i] - y_ngang);
        tong_ngang = tong_ngang + (x[i] - x_ngang) * (y[i] - y_ngang);
        }// het for i
mauso = Math.Sqrt(tongxbp * tongybp);
r = tong_ngang * 1.0 / (mauso * 1.0);
return r;
}
```

4.3.3. Hàm trích rút đặc trưng

Áp dụng thuật toán xác định dãy lặp ở mục 3.3.2, theo sơ đồ khối hình 3.4 ta xác định được các dãy lặp vơí khoảng lặp “p” và áp dụng hàm tương quan ta thu được tập dữ liệu đặc trưng. Hàm trích rút đặc trưng như sau:

```
Private void TaoDacTrung(string strfile, string filepath)
{
int[] Xmax = null;
int[] IdXmax = null;
int[] X = null;
```

```
int[] Y = null;
TrichRutXmax(strfile,ref Xmax, ref IdXmax, ref X,ref Y, 100, 6);
int[] kqp = new int[10000];
int[] kqid = new int[10000];
int[] kqdem = new int[10000];
int id1 = 0, l = 0, p = 0, dem = 0, delta, max = 0;
for (int i = 0; i < Xmax.Length; i++)
        {
        id1 = IdXmax[i];
        max = Xmax[i];
        p = 4;
        while (p <= 20)
        {
        dem = 0;
        delta = p;
        while (delta + id1 < X.Length)
        {
        if (Math.Abs(max - X[id1 + delta]) <= 1)
                {
                dem = dem + 1;
                delta = delta + p;
                }
                else
                {
                if (dem >= 4)
                        {
                        kqp[l] = p;
                        kqid[l] = id1;
                        kqdem[l]=dem;
                        l = l + 1;
                        break;
```

```
                    }
                    else
                    {
                    break;
                    }
                }
            }
            p = p + 1;
            }
}
//Loc trung
for (int i = 0; i < l - 1; i++)
{
        for (int j = i + 1; j < l; j++)
        {
        if (kqid[j] == kqid[i] + (kqdem[i] - kqdem[j]) * kqp[j])
                {
                for (int k = j; k < l; k++)
                kqid[k] = kqid[k + 1];
                l--;
                j--;
                }
        }
}
for (int t = 0; t < l; t++)
listBox3.Items.Add(kqid[t].ToString());
int[] A;
int[] B;
int[] C;
int[] D;
int[] F;
FileStream fs = new FileStream(filepath, FileMode.Append,
FileAccess.Write);
StreamWriter sw = new StreamWriter(fs);
```

```
        string strchuoi = "", tenfile = "", strY="";
        tenfile = "c:\\" + filenameshort;
        sw.WriteLine(tenfile);
        for (int i = 0; i < l; i++)
                {
                A = new int[kqp[i]];
                B = new int[kqp[i]];
                C = new int[kqp[i]];
                D = new int[kqp[i]];
                F = new int[kqp[i]];
                for (int j = 0; j < A.Length; j++)
                        {
                        A[j] = X[kqid[i] + j];
                        F[j] = Y[kqid[i] + j];
                        B[j] = X[kqid[i] + j + A.Length];
                        C[j] = X[kqid[i] + j + A.Length * 2];
                        D[j] = X[kqid[i] + j + A.Length * 3];
                        }
                if((tinh_he_so_tquan(A,B,A.Length)>=0.9)&&
(tinh_he_so_tquan(B,C,B.Length)>=0.9)&&(tinh_he_so_tquan(C,D,C.Length)>=0.9))
                {
                strchuoi = A[0].ToString(); strY = F[0].ToString();
                for (int k = 1; k < A.Length; k++)
                        {strchuoi =strchuoi+","+ A[k].ToString();
                        strY = strY + "," + F[k].ToString();}
                sw.WriteLine(kqp[i].ToString() + ";" + strchuoi+";"+strY);
strchuoi = ""; strY = "";
                }
                }
        sw.Close();
        }
```

4.3.4. Bảng các đặc trưng của một số phụ âm

Áp dụng cách trích rút đặc trưng từ các phần trên, ta thu được bảng các đặc trưng của một của phụ âm “c” như sau:

Người nói	Dãy các đặc trưng
1	c:\c.wav *(phụ âm “c” dạng file wav)*,1*(một dòng đặc trưng)* 10*(số phần tử x,y)***;**14,7,13,9,0,2,10,0,0,9**;**6,0,7,6,0,0,1,0,0,3
2	c:\c.wav,4 8;14,6,8,11,9,0,1,8;8,2,3,5,3,0,0,2 8;11,9,0,1,8,14,6,10;5,3,0,0,2,6,3,5 8;9,0,1,9,14,6,8,11;1,0,1,3,8,2,3,5 8;7,10,10,10,0,0,9,13;2,5,5,1,0,0,3,8
3	c:\c.wav,5 8;14,3,10,11,9,0,0,9;8,2,5,5,0,0,0,3 8;11,9,0,0,9,14,3,12;5,0,0,0,3,8,0,6 8;9,0,0,9,14,3,12,10;0,0,0,3,8,0,6,5 8;9,14,3,12,10,9,0,1;3,8,0,6,5,1,0,1 8;9,0,1,9,13,6,10,12;1,0,1,3,8,1,8,6
4	c:\c.wav,1 20;9,1,0,1,0,6,1,0,0,0,4,0,1,0,0,4,0,1,0,0;1,1,0,0,0,3,0,0,0,0,1,0,0,0, 0,3,0,0,0,0
5	c:\c.wav,2 18;9,3,0,4,0,8,1,1,5,1,0,4,0,2,9,3,0,2;4,1,0,2,0,5,0,0,0,0,0,3,0,1,1,1,0,1 18;8,1,1,5,1,0,4,0,2,9,3,0,2,9,2,0,4,1;5,0,0,0,0,0,3,0,1,1,1,0,1,1,0,0,2,0
6	c:\c.wav,5 12;10,0,1,2,1,7,1,0,9,3,0,2;7,0,1,2,1,6,0,0,3,2,0,1 12;9,3,0,2,9,1,0,3,1,6,1,1;3,2,0,1,7,1,0,2,0,5,1,0 10;8,3,0,2,12,3,1,7,1,0;2,2,0,1,7,1,0,6,0,0 12;7,1,0,9,3,0,2,9,1,0,3,1;6,0,0,3,2,0,1,7,1,0,2,0 10;7,1,1,8,3,0,2,12,3,1;6,0,0,2,2,0,1,7,1,0
7	c:\c.wav,3 12;9,2,1,1,5,3,0,2,5,6,2,7;8,0,0,0,2,2,0,1,1,3,1,1 12;7,2,7,9,2,1,1,5,3,0,2,3;4,1,1,8,0,1,0,2,2,0,1,1 8;7,3,2,2,1,18,1,29;6,1,0,1,0,1,0,5
8	c:\c.wav,6 8;23,5,0,1,7,0,1,1;6,4,0,0,0,0,1,0 8;9,4,1,4,1,0,3,2;6,0,0,3,0,0,3,1 16;8,0,1,1,24,5,1,0,7,1,0,2,24,5,1,0;1,0,1,0,6,4,0,0,1,1,0,1,6,4,0,0

	12;7,1,1,5,1,0,1,0,0,1,1,1;1,0,0,4,0,0,1,0,0,1,1,0 8;7,1,0,2,24,5,1,0;1,1,0,1,6,4,0,0

Bảng 4.1- Bảng các đặc trưng của phụ âm “c”

4.4. Nhận dạng phụ âm

Khi nhận dạng phụ âm dạng file wav ta sử dụng thuật toán mục 3.4 theo sơ đồ khối hình 3.8. Các bước được thực hiện như sau:

Bước 1: Tạo lập bộ dữ liệu đặc trưng. Cách trích rút đặc trưng trình bày ở phần 4.3.

Bước 2: Nhập file wav cần nhận dạng.

Bước 3: Chọn hệ số tương quan giữa Mau[j] và P (Mau[j]: thuộc bộ dữ liệu đặc trưng, P: đặc trưng của mẫu cần nhận dạng)

Bước 4: Nhận dạng: Nếu phát hiện được thì phát âm phụ âm cần nhận dạng, ngược lại nạp đặc trưng của phụ âm đó vào bộ dữ liệu đặc trưng.

Hàm nhận dạng như sau:

```
private void btnNhanDang_Click(object sender, EventArgs e)
{
string dataRow = "";
string row = "";
double rX = 0,rY=0, maxtq = 0;
int dem = 0;
StreamReader srMau = new StreamReader("D:\\Dactrung.txt");
StreamReader sr = new StreamReader("D:\\Nhan_Dang.txt");
row = sr.ReadLine();
string[] M = new string[2];
M = row.Split(',');
int m = int.Parse(M[1].ToString());
```

```
string[] N = new string[m];
for (int j = 0; j < m; j++)
N[j] = sr.ReadLine();
while (srMau.Peek() >= 0)
        {
        dataRow = srMau.ReadLine();
        string[] A = new string[2];
        A = dataRow.Split(',');
        int n = int.Parse(A[1].ToString());
        if (n == m)
                {
                B = new string[n];
                for (int i = 0; i < n; i++)
                {
                B[i] = srMau.ReadLine();
                string[] C = new string[3];
                string[] P = new string[3];
                C = B[i].Split(';');
                P = N[i].Split(';');
                string[] DX = new string[int.Parse(C[0].ToString())];
                string[] DY = new string[int.Parse(C[0].ToString())];
                string[] QX = new string[int.Parse(P[0].ToString())];
                string[] QY = new string[int.Parse(P[0].ToString())];
                DX = C[1].Split(',');
                DY = C[2].Split(',');
                QX = P[1].Split(',');
                QY = P[2].Split(',');
```

```
            int[] EX = new int[DX.Length];
            int[] EY = new int[DY.Length];
            int[] TX = new int[QX.Length];
            int[] TY = new int[QY.Length];
            for (int j = 0; j < DX.Length; j++)
            {
            EX[j] = int.Parse(DX[j].ToString());
            EY[j] = int.Parse(DY[j].ToString());
            }
            for (int j = 0; j < QX.Length; j++)
            {
            TX[j] = int.Parse(QX[j].ToString());
            TY[j] = int.Parse(QY[j].ToString());
            }
            if (EX.Length == TX.Length)
            {
            rX = tinh_he_so_tquan(EX, TX, EX.Length);
            rY = tinh_he_so_tquan(EY, TY, EY.Length);
            }
            if ((rX >= 0.6)&(rY>=0.5))
            dem = dem + 1;
            }
            if (dem/n >= 0.5)
            play_sound(A[0].ToString());
            }
        else if (n > m)
            {
            B = new string[n];
```

```
for (int i = 0; i < n; i++)
B[i] = srMau.ReadLine();
for (int i = 0; i < n - m + 1; i++)
{
for (int j = 0; j < m; j++)
{
string[] C = new string[3];
string[] P = new string[3];
C = B[j + i].Split(';');
P = N[j].Split(';');
string[] DX = new string[int.Parse(C[0].ToString())];
string[] DY = new string[int.Parse(C[0].ToString())];
string[] QX = new string[int.Parse(P[0].ToString())];
string[] QY = new string[int.Parse(P[0].ToString())];
DX = C[1].Split(',');
DY = C[2].Split(',');
QX = P[1].Split(',');
QY = P[2].Split(',');
int[] EX = new int[DX.Length];
int[] EY = new int[DY.Length];
int[] TX = new int[QX.Length];
int[] TY = new int[QY.Length];
for (int k = 0; k < DX.Length; k++)
{EX[k] = int.Parse(DX[k].ToString());
EY[k] = int.Parse(DY[k].ToString());}
for (int k = 0; k < QX.Length; k++)
{
TX[k] = int.Parse(QX[k].ToString());
TY[k] = int.Parse(QY[k].ToString());
```

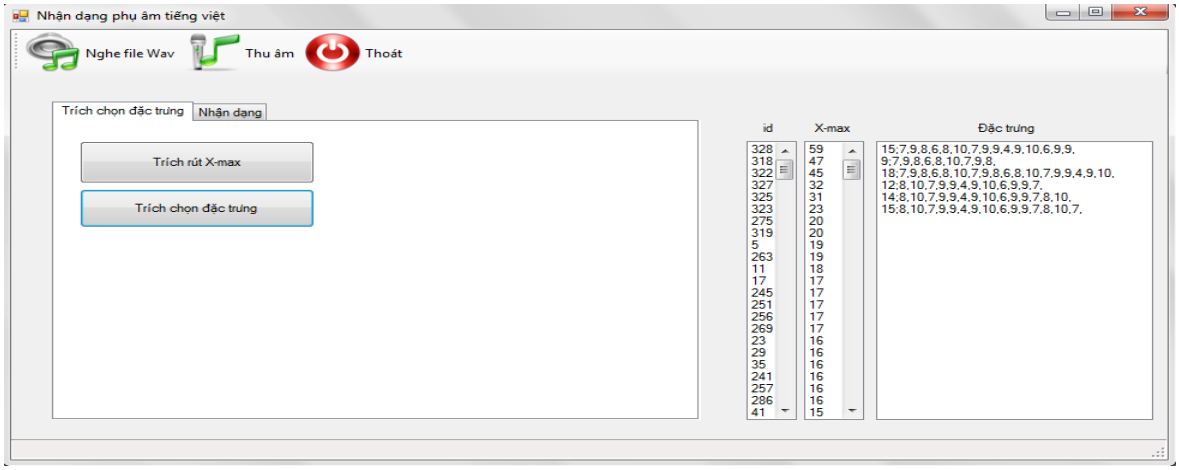
```
            }
            if (EX.Length == TX.Length)
            {
            rX = tinh_he_so_tquan(EX, TX, EX.Length);
            rY = tinh_he_so_tquan(EY, TY, EY.Length);
            }
            if ((rX >= 0.6)&(rY>=0.5))
            dem = dem + 1;
            }
            if (dem == m)
            play_sound(A[0].ToString());
            }
            }
else
            {
            B = new string[n];
            for (int i = 0; i < n; i++)
            B[i] = srMau.ReadLine();
            for (int i = 0; i < m - n + 1; i++)
            {
            for (int j = 0; j < n; j++)
            {
            string[] C = new string[3];
            string[] P = new string[3];
            C = B[j].Split(';');
            P = N[j+i].Split(';');
            string[] DX = new string[int.Parse(C[0].ToString())];
            string[] DY = new string[int.Parse(C[0].ToString())];
            string[] QX = new string[int.Parse(P[0].ToString())];
            string[] QY = new string[int.Parse(P[0].ToString())];
            DX = C[1].Split(',');
            DY = C[2].Split(',');
```

```
                QX = P[1].Split(',');
                QY = P[2].Split(',');
                int[] EX = new int[DX.Length];
                int[] EY = new int[DY.Length];
                int[] TX = new int[QX.Length];
                int[] TY = new int[QY.Length];
                for (int k = 0; k < DX.Length; k++)
                {EX[k] = int.Parse(DX[k].ToString());
                EY[k] = int.Parse(DY[k].ToString());}
                for (int k = 0; k < QX.Length; k++)
                {TX[k] = int.Parse(QX[k].ToString());
                TY[k] = int.Parse(QY[k].ToString());}
                if (EX.Length == TX.Length)
                {
                rX = tinh_he_so_tquan(EX, TX, EX.Length);
                rY = tinh_he_so_tquan(EY, TY, EY.Length);
                }
                if ((rX >= 0.6) & (rY >= 0.5))
                dem = dem + 1;}
                }
                if (dem == n)
                play_sound(A[0].ToString());
                }
                }
        }
}
```

4.5. Chương trình áp dụng và kết quả

4.5.1. Chương trình áp dụng

Dựa trên kết quả nghiên cứu, luận văn đã xây dựng chương trình cho phép trích chọn các đặc trưng và nhận dạng một số phụ âm tiếng việt. Chương trình được viết bằng ngôn ngữ Visual C# 2008.



Hình 4.9- Giao diện chính của chương trình

4.5.2. Kết quả thực nghiệm

Nạp bộ dữ liệu đặc trưng là 100 phụ âm "c" của 10 người nói khác nhau. Sử dụng 20 phụ âm "c" mới và 20 phụ âm khác như "k", "n" đưa vào để nhận dạng.

tuongquan = 0,5.

Input \ Output	C	Phụ âm	Không biết
20 phụ âm "C"	5		15
20 phụ âm khác "C"	1	1	18

tuongquan = 0,4

Input \ Output	C	Phụ âm	Không biết
20 phụ âm "C"	15		5
20 phụ âm khác "C"	7	2	11

Hình 4.10- Kết quả thực nghiệm đề tài

KẾT LUẬN VÀ KIẾN NGHỊ

1. Kết luận

Luận văn đã thống kê, tóm tắt các tham số ảnh hưởng đến hệ thống nhận dạng tiếng nói cũng như các vấn đề cơ bản về nhận dạng tiếng nói. Phân tích, so sánh một số mô hình nhận dạng tiếng nói tiếng việt dựa trên quan niệm truyền thống về từ. Trên cơ sở đó, đưa ra kỹ thuật nhận dạng từ độc lập theo quan điểm mới. Luận văn đã dựa vào một số đặc điểm của phụ âm để tách ra các đặc trưng bằng cách sử dụng điểm cắt zero và các công cụ toán học. Từ đó, phân tích một số đặc trưng của các thành phần của phụ âm phục vụ cho nhận dạng tiếng nói.

Kết quả luận văn:

Có hướng tiếp cận mới vào bài toán nhận dạng tiếng nói, nội dung cụ thể:

(1) Sử dụng điểm cắt Zero và hệ số tương quan để tìm đặc trưng của tiếng nói.

(2) Xây dựng được bộ dữ liệu đặc trưng của phụ âm "c" với 10 người nói.

(3) Xây dựng thuật toán nhận dạng để thử nghiệm với những mẫu đặc trưng trên.

2. Những hạn chế và kiến nghị:

Do trình độ có hạn và trong phạm vi luận văn thạc sỹ nên kết quả đề tài còn hạn chế. Để có kết quả tốt hơn, tác giả đề xuất một số định hướng sau:

(1) Chưa đánh giá được chất lượng bộ dữ liệu đặc trưng tìm được.

(2) Rút gọn tập dữ liệu đặc trưng.

(3) Thử nghiệm thêm nhiều người nói ở vùng miền và lứa tuổi khác nhau để có thể chọn ra tập đặc trưng thích hợp nhất.

TÀI LIỆU THAM KHẢO

Tiếng việt

1. Nguyễn Văn Xuất (2010), *Bài giảng Multimedia,* Học viện Kỹ thuật quân sự.

2. Nguyễn Văn Xuất (2006*), Xem xét tiếng việt đưới góc độ công nghệ thông tin – ITMath'06 một số nguyên âm đề thời sự trong CNTT và ứng dụng toán học 10/2006*, Học viện Kỹ thuật quân sự.

3. Dương Tử Cường(2003), *Xử lý tín hiệu số,* NXB Quân đội nhân dân.

4. Bạch Hưng Khang(2004), *Nghiên cứu phát triển công nghệ nhận dạng tổng hợp và xử lý ngôn ngữ tiếng việt, Đề tài khoa học*, Viện CNTT.

5. Bùi Minh Toán, Đặng Thị Lanh, Lê Hữu Tỉnh(1993), *Tiếng việt*, Bộ GDĐT - Vụ giáo viên, Hà nội

6. Hồ Tú Bảo, Lương Chi Mai, *Về xử lý tiếng Việt trong Công nghệ thông tin*.

Tiếng Anh

1. David Salomon(2004), *Data Compression The Complete Reference, 3ed* (Springer).

2. Wiley (2003), *Speech Coding Algorithms Foundation and Evolution of Standardized Coders*, Ebooks.

3. Deller J.R; Hansen J.H.L; Proakis J.G. (2000), *Discrete –Time Processing of Speech Signals*, IEEE Press.

4. Dr Roj Reddy(2001), *Spoken Language Processing – A Guide To Theory, Algorithms And System Development,* Prentice Hall Inc

5. Ian H.Witten, Radford M. Neal and John G,(1987), *Arithmetic coding for data compression, Clearyin Communications of the ACM.*

LÝ LỊCH TRÍCH NGANG

Họ và tên: Đào Sỹ Nhiên

Ngày tháng năm sinh: 09/8/1979; Nơi sinh: Hoa L- , Ninh Bình

Địa chỉ liên lạc: Khoa Ngoại ngữ - Tin học, Tr- ờng Đại học Hoa L- , Ninh Bình

QUỸ TR□NH ĂÀO T□O:

Từ năm 1998 – 2003 học tại khoa Công Nghệ Thông tin, trường Đại học Bách Khoa Hà Nội.

Từ năm 2009 – 2011 học cao học ngành Khoa học máy tính, khoa Công nghệ Thông tin, Học viện Kỹ thuật Quân Sự.

QUỸ TR□NH C□NG TỸC:

- Từ năm 2003 – 2005 làm việc tại Công ty HDIC.

- Từ năm 2006 đến nay làm việc tại Tr- ờng Đại học Hoa L- , Ninh Bình.