Speaker Recognition Enhanced Voice Conference

Yancheng Li, Liang Wei, Zhaoyuan Zhang

I. Introduction

With the development of the whole world, more and more people need to communicate with each other even if they live in different parts of the world, especially for the businessmen who tried to make profits all over the world. In that situation, it is clear that an audio conference system will be popular because it permits people live in different places to discuss something with each other easily. Therefore, our team chooses to design an audio conference system in this project.

The audio conference system can not only provide service for people to talk about something with each other, it can also automatically recognize the user who is speaking and send his/her name to all the other clients. With this special function, people who take part in the conference can easily know who is speaking even if they have not met before.

There are two parts in this audio conference system. The first part is a voice recognition system, which is used to train a record sample and recognize the user who is speaking during an audio conference. The second part is a client-server program. The client program can send the content a user says to the server and receive the recognition result from the server; the server program is responsible for forwarding the contents from one client to others and calls voice recognition system to recognize the user who is speaking.

II. Usage Model

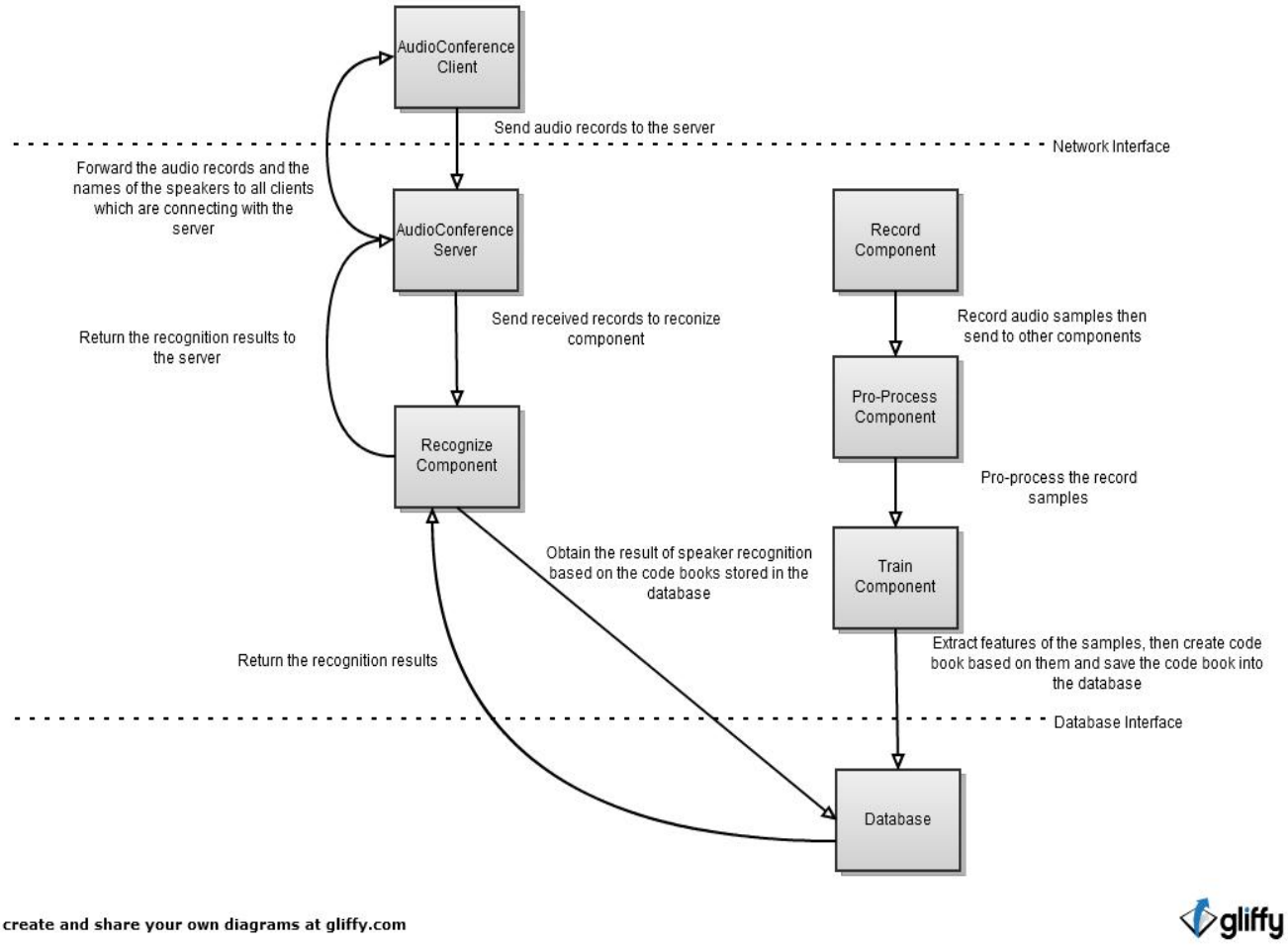
From user's perspective, the first thing needed to do is that s/he must send a good enough record sample to the system. The system will pre-process the sample at first and then train it. The user needs to input his/her name so that the system can save the codebook with right corresponding name into the database. Actually, this process is just like a registration, which is used to create personal profile for each user.

Secondly, after the registration, the user can open a client program to connect with the server. The server is responsible for forwarding the packets (records) and the recognition result to other clients when the conference is carrying on. The user must connect the server successfully or they cannot get any service at all.

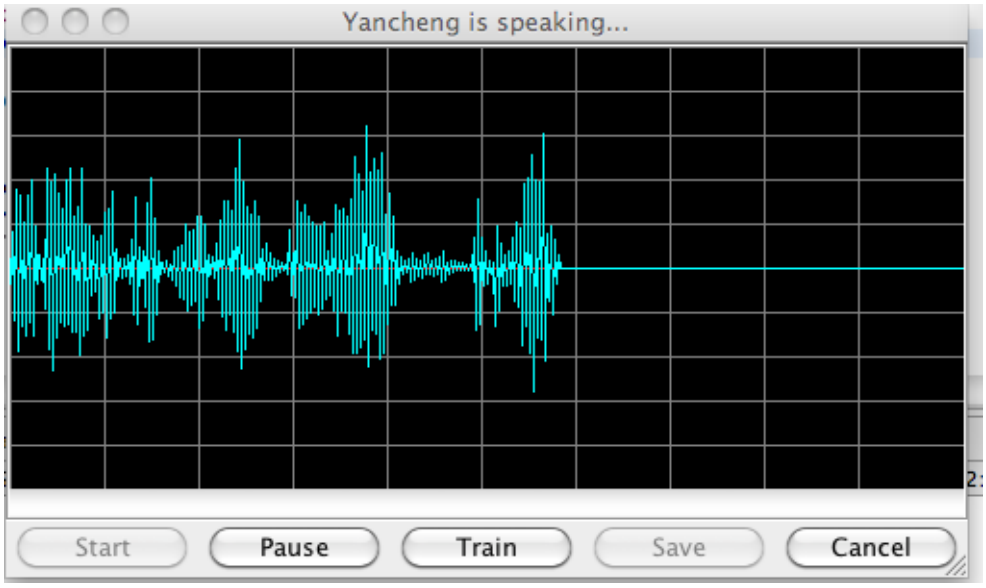
Thirdly, when the client program connects with the server successfully, the user can try to make a speech to others. The voice samples are sent to the server at first. Then the server will forward it to other clients and played automatically. Also there is a visual display of voice samples received. In the process of forwarding the record, the server will also invoke the voice recognition system to extract the features of this record and compare it with all codebooks in the database. The system will choose the most similar one as the recognition result and send the name of it (the codebook is named by the user name) to all the other clients.

This is the main working process of the system we designed. When the conference is over, the user just needs to close the client program, while the server program will keep running so that it can provide service all the time.

III. System Architecture



IV. Client Terminal Overview



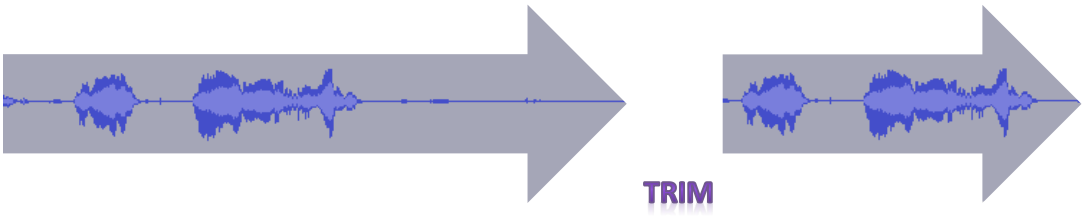
V. Speaker Recognition Algorithm

The speaker recognition algorithm is based on the algorithm introduced in "Speaker recognition, Model & Method" by Chaohui Wu, Tsinghua University Press. [1]

A. Training

1. Preprocessing

The preprocessing step is to trim the training sequence of voice sample, remove the leading and tailing non-speech section. The idea is to find the start point and end point of speech. The start/end point is judged by the short-time frame energy and the cross-zero frequency.



2. Feature Extraction

The voice samples are divided into frames, each frame contains 128 bits. Voice features are extracted on each frame, the features include:

1) LPC: The 12-order LPC is calculated by using Levinson-Durbin recursion.[2]

2) LPCC: LPC-derived cepstral coefficient, calculated from LPC.

3) Overlapped LPC: LPC calculated on half-frame overlapped data.

4) Voice Pitch

These features compose the feature as following (a double array actually):

Feature extraction is performed on each frame, so the final feature vector for this user will be the average of all frame feature vectors. To improve recognition effectiveness, if the frame energy is very low, it is quite possible to be silent, and then this frame feature vector will not be included in average calculation.

The last step, the feature vector will be saved in the database associated with user's name.

B. Recognition Process

The recognition process is quite similar to training; the data samples are processed as in A (preprocess + feature extraction). Then the feature vectors will be compared to the vectors within the database, the user with most similar vector will be returned. The similarity between two vectors is measured by Euclidean distance.

VI. Project Status

Currently, most of the general functionalities of our Speaker Recognition Enhanced Voice Conference system have been implemented. These functionalities include: collects the voice sample from an end user; distribute voice samples to all clients with the help of voice server; extracts the voice feature of a speaker and converts to codebook; compares voice sample's feature vector and return most similar one. We also have implemented graphic client user interface.

We have tested the system on voice recording, training and recognition with different speakers including both male and female speakers. It turns out that the system works pretty well at this moment. The speaker recognition accuracy at this moment is about 80%, which is somehow close to our initial expectation. The system is still in a beta testing version. It may contain some potential bugs but we will fix them as soon as we find them.

VII. Evaluation

Most of the core functionalities for speaker recognition described in the project plan have been implemented as described. Some of the minor functionalities were discarded or modified for the reason of impractical issues or time insufficiency.

VIII. Individual Contribution

Design of the System Architecture: Yancheng Li

Algorithm Research & Selection: Yancheng Li

Algorithm Implementation: Whole team

Server End of the Voice Conference implementation: Liang Wei & Zhaoyuan Zhang

Client End of the Voice Conference implementation: Yancheng Li

System Integration: Whole team

Sample Collecting: Liang Wei & Zhaoyuan Zhang

System Test: Whole team

[1] "*Speaker recognition, Model & Method" Chaohui Wu, Tsinghua University Press*.

[2] http://en.wikipedia.org/wiki/Levinson_recursion