Abstract
This document outlines the system design using high and low level block diagrams, algorithm flow charts, system flow chart and module deliverable requirements
Table of Contents
1 Introduction

1.1 Purpose
The purpose of this document is to give a concrete outline of the system design. This document will provide both high and low level block diagrams, flow charts and the input-output requirements of each module. The goal of this document is to allow the developers to implement the system using this document.

The intended audience of this document is the developers.

2 High Level Design
There are two modules in the speaker verification system. The User Enrollment module and the User Verification module. Figure 1 shows a high-level block diagram outlining the subsystems within each module. Each subsystem will be discussed briefly in the following sections.

![Figure 1. High Level Block Diagram of System Verification System](image)

2.1 User Enrollment Module
The User Enrollment Module is used when a new user is added to the system. This module is used to essentially “teach” the system the new user’s voice. The input of this module is a predetermined series of words and sentences spoken by the user. By analyzing this training speech, the module outputs a codebook that essentially parameterizes the user’s voice. This codebook will be used later in the User Verification Module.

2.1.1 Signal Preprocessing Subsystem
The signal preprocessing subsystem conditions the raw speech signal and prepares it for subsequent manipulations and analysis. This subsystem performs analog-to-digital conversion and noise-reduction.
2.1.2 Feature Extraction Subsystem
The feature extraction subsystem analyzes the user’s digitized voice signal and creates a series of values to use as a model for the specific user’s speech pattern. These values are in the form of vectors called acoustic vectors.

2.1.3 Feature Data Compression Subsystem
The Feature Extraction subsystem discussed in section 2.1.2 creates a large number of acoustic vectors. In order to store this data effectively, a form of data compression is used. A vector quantization method will be used. Specifically, the Linde Budo Gray (LBG) Vector Quantization algorithm will be used. The result of this compression will be a codebook. This codebook will be stored for later use in the User Verification Module.

2.2 User Verification Module
The User Verification Module is used when the system tries to verify a user. The module performs the same signal pre-processing and feature extraction to the raw speech signal as the User Enrollment Module. After that, a set of acoustic vectors are produced. These acoustic vectors are then compared to the clustered vectors stored in the codebook. Based on these comparisons, a similarity factor will be produced. Finally, this similarity factor will be compared to a system parameter called the Threshold and a verdict is generated.

2.2.1 Signal Preprocessing Subsystem
The Signal Preprocessing Subsystem discussed in section 2.1.1 can be reused here.

2.2.2 Feature Extraction Subsystem
The Signal Preprocessing Subsystem discussed in section 2.1.2 can be reused here.

2.2.3 Comparison Subsystem
The acoustic vectors presented to this subsystem by the Feature Extraction Subsystem will be compared to the clustered vectors stored in the codebook. An overall similarity factor will be produced at the end of the comparison.

2.2.4 Decision Subsystem
Based on the similarity factor produced by the Comparison Subsystem in section 2.2.3, and the Administrator specified Threshold value, a verdict will be given by this subsystem.

3 Low Level Design
The following section describes the information required for implementation of each subsystem. The input and output requirements of each module will be specified. Block diagrams and flow charts will be provided as necessary.

The aim of this section is to provide a black-box approach for developers when implementing their assigned subsystems.

3.1.1 Signal Preprocessing Subsystem
Input: Raw speech signal
Output: Digitized speech signal
The subsystem will sample the analog speech signal and produce a digital signal in the form of a vector or array. Sampling rate will be chosen appropriately to satisfy the Nyquist Sampling Theorem.

Prior to sampling, a noise reduction filter may be used depending on the developer.

3.1.2 Feature Extraction Module

**Input:** Digital speech signal (vector of sampled values)

**Output:** A set of acoustic vectors

In order to produce a set of acoustic vectors, the original vector of sampled values is framed into overlapping blocks. Each block will contain N samples with adjacent frames being separated by M samples where M < N. The first overlap occurs at N-M samples. Since speech signals are quasi-stationary between 5msec and 100msec, N will be chosen so that each block is within this length in time. In order to calculate N, the sampling rate needs to be determined. N will also be chosen to be a power of 2 in order to make use of the Fast Fourier Transform in a subsequent stage. M will be chosen to yield a minimum of 50% overlap to ensure that all sampled values are accounted for within at least two blocks.

Each block will be windowed to minimize spectral distortion and discontinuities. A Hamming window will be used.

The Fast Fourier Transform will then be applied to each windowed block as the beginning of the Mel-Cepstral Transform. After this stage, the spectral coefficients of each block are generated.

The Mel Frequency Transform will then be applied to each spectrum to convert the scale to a mel scale. The following approximate transform can be used.
\[ \text{mel}(f) = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \] (1)

Finally, the Discrete Cosine Transform will be applied to each Mel Spectrum to convert the values back to real values in the time domain.

### 3.1.3 Feature Data Compression Algorithm

**Inputs:** A set of acoustic vectors  
**Output:** Codebook

**Figure 3. Feature Data Compression Subsystem Low-Level Block Diagram**

Acoustic Vectors → Feature Data Compression Algorithm → Updated Codebook  
Set of Mel-Cepstral Coefficients (Acoustic Vectors) → Linde Buzo Gray Clustering Encoding Algorithm → Codebook

In order to store the large set of acoustic vectors, the Linde Buzo Gray Vector Quantization Algorithm is used.

Given a training sequence of \( M \)

The following is a flow chart outlining the LBG-VQ Algorithm.

**Figure 4. LBG VQ Algorithm Flow Chart**

### 3.1.4 Comparison Subsystem

**Inputs:** Set of acoustic vectors from verification session; codebook  
**Outputs:** Normalized average distortion factor

**Figure 5. Comparison Subsystem Low-Level Block Diagram**

The acoustic vectors generated by the trial voice signal will be individually compared to the codebook. The codeword closest to each test vector is found based on Euclidean Distance. This Euclidean
Distance, or Distortion Factor, is then stored until the Distortion Factor for each test vector has been calculated. The Average Distortion Factor is then found and normalized.

A flow chart describing the Distortion Factor calculation is shown.

**Figure 6.** Distortion Calculation Algorithm Flow Chart

![Flow Chart](image)

3.1.5 Decision Subsystem

**Inputs:** Normalized average distortion factor; Threshold

**Outputs:** Verdict

**Figure 7.** Comparison Subsystem Low-Level Block Diagram
The Normalized Average Distortion Factor will be normalized to a value between 0 and 1. Therefore, the threshold will be a percentage. Based on a simple comparison, a Pass verdict will be given if the distortion is less than the threshold and a Fail verdict is given if the distortion is greater than the threshold.

4 Graphical User Interface

5 Attention

- Need to optimize codebook search algorithm in comparison module for speed
- Check LBG decode algorithm flow chart for correctness
- Find suitable values for all parameters such as codebook size, acoustic vector length, thresholds, etc.
- Note comment in LBG encode flow chart
- See if there should be any additional modules for Signal Pre-Processing
- Thinking about solutions for variations in sampling sessions (i.e. microphone, environment acoustics)