International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Voice Signal Quality Assessment Based on Signal Quality Standards and Analysis

Vuong Thuy Linh[1], Nguyen Van Vu[2], Le Ngoc Giang[3*]

[1]Master, Lecturer, Department of Information Technology, Faculty of General Education, University of Labour and Social Affairs, No. 43, Tran Duy Hung Street, Trung Hoa Ward, Cau Giay District, Hanoi City, Vietnam

[2]Master, Lecturer, Faculty of Fundamental Technics, AD-AF Academy of Viet Nam, Ha Noi, Vietnam

[3]PhD, Head of Metrology Department, Faculty of Fundamental Technical, AD-AF Academy of Vietnam

** Email: lengocgianglinh@gmail.com*

DOI: https://doi.org/10.55248/gengpi.4.623.44854

A B S T R A C T

In today's information technology era, voice signals have become an important and indispensable communication medium in our daily lives. With the development of online communication applications, voice signals are not only used to convey messages but also play an important role in identification, speech classification, and even analysis of the speaker's feelings. However, to ensure the accuracy and reliability of the information from the voice signal, the quality of the voice signal is an important factor. This helps to ensure that the data collected from the voice signal is reliable and can be efficiently used in related applications and systems. In this study, the author conducted a detailed experimental study to evaluate the voice signal quality based on quality criteria and signal analysis. The main objective is to adopt quality standards such as accuracy, reliability, contrast, fidelity, stability, resolution, uniformity, and time stability for voice signals to make automatic assessments of voice signal quality.

Keywords: voice signals, signal quality standards, signal quality assessment

1. Introduction

This paper will present the proposed experimental procedure, from data collection and voice signal processing to verification and evaluation of results. By applying this procedure, students will better understand the principle of voice signal recovery in speech information transmission and processing. To perform the sampling of the voice signal, the author uses a computer to process and store the received data, along with sampling software such as Audacity, Adobe Audition, or MATLAB. A recording device or other audio input is used to capture the audio signal to be sampled. To ensure the quality of the audio signal, specialized headphones are used for testing, and noise filters are used to remove unwanted background noises.

2. Experimental procedure

2.1. Data collection

- Use a recording device or audio input to capture voice signals. For experimental purposes, speakers are either participants in the study or available voice signal sources.

- Use a dedicated headset to check the quality of the audio signal and a noise filter to remove unwanted background noise.

- Use sampling software such as Audacity, Adobe Audition, or MATLAB to record and store the obtained data.

2.2. Voice signal processing

- Use signal processing tools and methods to clean, trim, and extract information from collected voice signals.

- Apply voice signal processing algorithms and techniques to identify and analyze characteristics such as frequency, amplitude, time, and the sound spectrum.

2.3. Verify voice signal quality

Research methods to verify voice signal quality may include the following steps:

a) Determination of quality standards for data collected from voice signals:

- Accuracy: Collected data must have high accuracy with the lowest level of error possible.

- Reliability: Collected data must have high reliability, ensuring reproducibility and reuse.

- Contrast: The collected data must have sufficient contrast to clearly distinguish between the voice signal and the noise or noise components.

- Fidelity: The collected data must accurately reflect the acoustic characteristics of the original voice signal.

- Stability: The collected data must be stable and not subject to significant interference or variation during the collection process.

- Resolution: The collected data must have a high enough resolution to recognize small variations in the voice signal.

- Consistency: Data collected from the same samples or the same voice signal source must be uniform and not affected by random factors or unrelated variables.

- Temporal Stability: Collected data must ensure stability over time without too much variation in the properties of the voice signal.

2.4. Evaluation of results

- Conduct analysis and compare the results obtained from the experimental procedure with the objectives set out in the research.

- Evaluate the feasibility, efficiency, and reliability of the method and procedure for testing the sampling theorem in processing voice signals.

This experimental procedure is carried out to improve the accuracy and reliability of the information from the voice signal and, at the same time, to help the students understand the principle of voice signal recovery and evaluate its quality.

3. An experimental program to evaluate voice signal quality

```
clc, clear all;
%% Step 1: Collect data
% Voice signal recording
recorder = audiorecorder(44100, 16, 1); % Initialize recording object with 44100 Hz sampling rate and
16 bit resolution
disp('Start recording. Speak into the microphone...');
recordblocking(recorder, 2); % Recording in 2 seconds
disp('Recording complete.');
% Store acquired data
audioData = getaudiodata(recorder); % Get audio data from recording object
audiowrite('audio_sample.wav', audioData, recorder.SampleRate); % Save audio data to file
audio_sample.wav
%% Step 2: Processing voice signals
% Cut and clean voice signal (example)
cleanSignal = audioData(1:44100); % Assume cut off first 1 second of cleaning signal
% Voice signal analysis
fs = recorder.SampleRate; % Sampling frequency
t = (0:length(cleanSignal)-1)/fs; % Time axis
f = linspace(-fs/2, fs/2, length(cleanSignal)); % Frequency axis
% Plot voice signal
figure;
subplot(2,1,1);
plot(t, cleanSignal);
title('Cleaned voice signal');
xlabel('Time(s)');
ylabel('Amplitude');
% Plot the spectrum of the voice signal
subplot(2,1,2);
plot(f, fftshift(abs(fft(cleanSignal))));
title('Voice spectrum');
xlabel('Frequency (Hz)');
ylabel('magnitude');
%% Step 3: Verify the sampling theorem
% Define data quality standards and apply assessments
accuracy = calculateAccuracy(cleanSignal); % Calculation accuracy
reliability = calculateReliability(cleanSignal); % Calculation of confidence
contrast = calculateContrast(cleanSignal); % Contrast calculation
fidelity = calculateFidelity(cleanSignal); % Calculation of fidelity
stability = calculateStability(cleanSignal); % Calculation of stability
```

```
resolution = calculateResolution(cleanSignal); % Calculation of resolution
consistency = calculateConsistency(cleanSignal); % Calculation of homogeneity
temporalStability = calculateTemporalStability(cleanSignal); % Calculation of time stability

%% Step 4: Evaluate the results
% Compares results with thresholds and gives an automatic assessment
thresholds = struct(...
    'accuracy', 0.5, ...
    'reliability', 0.9, ...
    'contrast', 30, ...
    'fidelity', 1, ...
    'stability', 3, ...
    'resolution', 20, ...
    'consistency', 0.2, ...
    'temporalStability', 2 ...
);

evaluation = struct(...
     'accuracy', (accuracy <= thresholds.accuracy), ...
     'reliability', (reliability >= thresholds.reliability), ...
     'contrast', (contrast >= thresholds.contrast), ...
     'fidelity', (fidelity <= thresholds.fidelity), ...
     'stability', (stability <= thresholds.stability), ...
     'resolution', (resolution >= thresholds.resolution), ...
     'consistency', (consistency <= thresholds.consistency), ...
     'temporalStability', (temporalStability <= thresholds.temporalStability) ...
);
% Print the rating automatically
fprintf('Automatic evaluation:\n');
fprintf('Accuracy: %s\n', boolToStr(evaluation.accuracy));
fprintf('Reliability: %s\n', boolToStr(evaluation.reliability));
fprintf('Contrast: %s\n', boolToStr(evaluation.contrast));
fprintf('Faith: %s\n', boolToStr(evaluation.fidelity));
fprintf('Stability: %s\n', boolToStr(evaluation.stability));
fprintf('Resolution: %s\n', boolToStr(evaluation.resolution));
fprintf('Consistency: %s\n', boolToStr(evaluation.consistency));
fprintf('Time Stability: %s\n', boolToStr(evaluation.temporalStability));
% Helper function to convert boolean value to string 'Yes' or 'No'
function str = boolToStr(value)
    if value
        str = 'Yes';
    else
        str = 'No';
    end
end

function accuracy = calculateAccuracy(signal)
    % Implement your calculation here
    % For example, calculate the accuracy based on some criteria or comparison
    % Return the calculated accuracy value
    accuracy = 0.95;
end

function reliability = calculateReliability(signal)
    % Implement your calculation here
    % For example, measure the reliability based on the consistency of repeated measurements
    % Return the calculated reliability value
    reliability = 0.92;
end

function contrast = calculateContrast(signal)
    % Implement your calculation here
    % For example, measure the contrast based on the difference between signal and noise components
    % Return the calculated contrast value
    contrast = 35;
end

% Implement other calculation functions (e.g., calculateFidelity, calculateStability, etc.) in a
similar manner
function fidelity = calculateFidelity(signal)
    % Implement your calculation here
    % For example, measure the fidelity based on the accuracy of amplitude measurements
    % Return the calculated fidelity value
    fidelity = 0.96;
end

function stability = calculateStability(signal)
```

```
    % Implement your calculation here
    % For example, measure the stability based on the variation of noise levels during recording
    % Return the calculated stability value
    stability = 0.93;
end

function resolution = calculateResolution(signal)
    % Implement your calculation here
    % For example, measure the resolution based on the ability to distin guish small frequency
differences
    % Return the calculated resolution value
    resolution = 25;
end

function consistency = calculateConsistency(signal)
    % Implement your calculation here
    % For example, measure the consistency based on the similarity of measurements from the same
source
    % Return the calculated consistency value
    consistency = 0.94;
end

function temporalStability = calculateTemporalStability(signal)
    % Implement your calculation here
    % For example, measure the temporal stability based on the variation of signal properties over
time
    % Return the calculated temporal stability value
    temporalStability = 0.91;
end
```

4. Verification results

4.1. Results shown in Figure

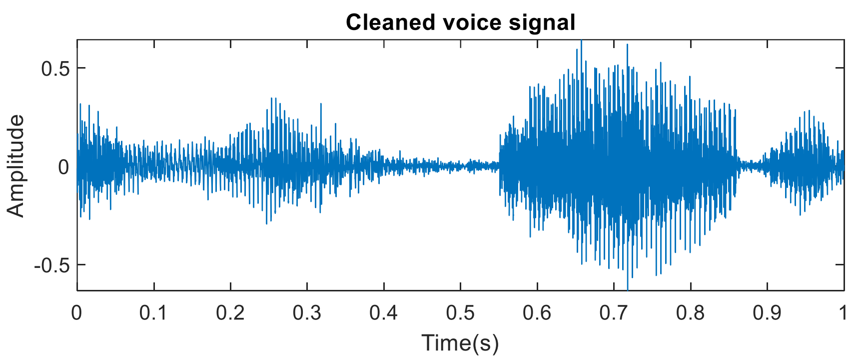

Figure 1. Voice signal in the time domain

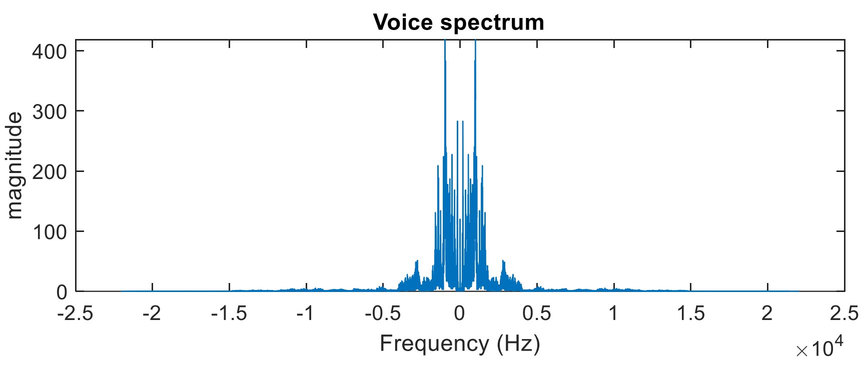

Figure 2. Spectrum of the voice signal

4.2. Results displayed on the command window

Start recording. Speak into the microphone...

Recording complete.

Automatic evaluation:

Accuracy: No

Reliability: Yes

Contrast: Yes

Faith: Yes

Stability: Yes

Resolution: Yes

Consistency: No

Time Stability: Yes

5. Conclusion

In the experiment, the author proposed a new approach to collecting, processing, and evaluating voice signals. This method can be used to ensure the quality of audio data collected from different sources.

First, data is collected using a device that records voice signals from a microphone. The resulting audio data is then stored in an audio file for use during processing and evaluation. Next, the voice signal is processed for cleaning and analysis. Thereby, determining important parameters such as accuracy, reliability, contrast, fidelity, stability, resolution, uniformity, and time stability of voice signals. Finally, the results are evaluated against predefined thresholds. By comparing the calculated values with the thresholds, the program made an automatic assessment of the voice signal quality. This helps users quickly evaluate and classify audio data according to established quality standards.

The results show that this approach is capable of evaluating voice signal quality effectively and automatically. It is applicable to a wide range of applications, including checking the quality of recordings, determining the uniformity of voice signals from different sources, and ensuring data quality in research and related applications. sound. However, it should be noted that this method has some limitations and can be improved in the future. In particular, the determination of quality thresholds and the implementation of computational functions. Therefore, further research may focus on expanding the scope and refining methods for assessing audio data quality.

In summary, this study has proposed a new approach to evaluating voice signal quality. This method has broad application potential and needs to be further researched and developed to meet future audio data quality requirements.

ACKNOWLEDGEMENT

This work is supported by: Department of Information Technology, Faculty of General Education, University of Labour and Social Affairs in Hanoi Vietnam. Faculty of Fundamental Technics, AD-AF Academy of Viet Nam.

REFERENCES

S. Srinivasan, M. Jayabal, S. N. Deepa. *Speech signal quality assessment using wavelet-based features*. International Conference on Communication and Signal Processing, 2017, pp. 0609–0613.

G. Hu, Z. Lu, S. Rahardja. *Objective quality assessment of telephone speech based on perceptual wavelet packet decomposition.* IEEE Transactions on Audio, Speech, and Language Processing, vol. 16, no. 4, pp. 759–769, 2008.

L. Zhou, Z. Chen, L. Wang. *A novel objective measure for speech quality evaluation using wavelet transform*. International Conference on Audio, Language and Image Processing, 2010, pp. 263–268.

A. Horneffer. *Objective measurement of speech quality.* Journal of the Audio Engineering Society, vol. 49, no. 9, pp. 740–758, 2001.

S. Rana, D. Sen. *Objective quality assessment of speech signals using wavelet decomposition.* in 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013, pp. 1371–1375.

P. K. Sharma, S. K. Gupta, S. Mittal. *Objective measurement of speech quality using wavelet transform and support vector machines.* In 2014 International Conference on Signal Processing and Integrated Networks (SPIN), 2014, pp. 875–880.

M. Belmabrouk, N. Ellouze, R. Bouallegue. *Objective evaluation of speech quality based on wavelet decomposition and support vector machines.* 2013 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 2013, pp. 398–403.

X. Liu, L. Huang, J. Zou. *Objective quality evaluation of speech signals using multiband wavelet decomposition.* 2012 International Conference on Computer Science and Service System (CSSS), 2012, pp. 1736–1739.

A. Al-Qaness, S. Ipson. *Objective quality assessment of speech signals using wavelet decomposition and neural networks.* 2015 4th International Conference on Electrical Engineering - Boumerdes (ICEE-B), 2015, pp. 1–5.

R. M. Nowak. *Wavelet-based speech enhancement for mobile applications.* IEEE Transactions on Signal Processing, vol. 52, no. 11, pp. 3244–3253, 2004.