Signal Compression and decomposition problem

1. Problem Description

Introduction to Signal Compression

Signal compression is a vital technique used to reduce the amount of data required to represent a signal, making storage and transmission more efficient. This process is crucial in various fields such as telecommunications, audio and video streaming, and data storage, where bandwidth and space are often limited. By compressing signals, we can improve the efficiency of data transfer and storage while maintaining an acceptable level of quality.

Lossless vs. Lossy Compression

Signal compression can be broadly categorized into two types: lossless and lossy compression.

Lossless Compression:

Definition: This method allows the original signal to be perfectly reconstructed from the compressed data. No information is lost during the compression process.

Pros: The main advantage is that it preserves the original quality of the signal. This is essential in applications where data integrity is critical, such as text files, medical imaging, or certain audio formats.

Cons: Lossless compression generally results in larger file sizes compared to lossy methods, as it does not remove any data.

Lossy Compression:

Definition: In contrast, lossy compression reduces file size by removing some data deemed less important. This results in a loss of fidelity but can significantly decrease the amount of data.

Pros: The primary benefit is the substantial reduction in file size, making it ideal for streaming applications where bandwidth is a concern, such as in audio and video formats like MP3 or JPEG.

Cons: The trade-off is the potential loss of quality, which can be noticeable in some cases, especially if the compression is aggressive.

Introduction to the MELP Compression Method

The Mixed Excitation Linear Prediction (MELP) is an advanced method of speech compression designed to provide high-quality speech encoding while maintaining low bit rates. MELP efficiently compresses speech signals by modeling the human vocal tract and incorporating mixed excitation techniques, which enhances the quality of synthetic speech. This method is particularly advantageous in scenarios such as secure communications and low-bitrate voice applications, where maintaining intelligibility and naturalness is crucial.

Problem Statement

In this competition, participants will be tasked with decompressing a MELP-compressed signal without any provided training datasets. The challenge lies in developing learning-based (AI) algorithms capable of reconstructing the original signal from an unknown test signal that has been compressed using the MELP method.

Participants must leverage their skills in machine learning to create models that can adapt and learn from the characteristics of MELP compression. The effectiveness of the algorithms will be compared using various similarity metrics and timing assessments. Participants should be prepared to justify their approaches and demonstrate the efficiency of their algorithms in achieving optimal decompression results in the context of MELP compression.

1-1. Basic Concept:

- **Compression Ratio**: The degree to which the data size is reduced.
- **Similarity Criteria**: Metrics to evaluate how closely the decompressed signal matches the original signal.
- **MELP Algorithm**: A lossy speech compression method that operates at low bitrates, designed for applications where bandwidth is constrained.

2. Solution Requirements

2-1. Solution File:

The submission must include:

- **Source Code**: Implementation of the MELP algorithm for both compression and decompression at various bitrates (300 bps, 600 bps, 1200 bps, and 2400 bps).

- **Documentation**: Detailed descriptions of the compression-decompression process, the trade-offs at different bitrates, and how the solution was optimized for both signal quality and computational efficiency.

2-2. Programming Languages:

Participants may use any of the following languages to implement their solution:

- **Python**: Suitable for quick prototyping and includes powerful libraries for signal processing.
- **C/C++ (Provides bons score)**: Ideal for real-time applications and optimization of computational performance.
- **MATLAB**: A popular tool for signal processing, offering built-in functions for speech analysis.

2-3 suggestion:

- Generative Model Effectiveness (Optional but Encouraged) Participants who use generative models to improve decompression quality will be evaluated based on: The degree of quality improvement achieved through the use of neural networks or other AI techniques, particularly at low bit rates. How well the model restores intelligibility and naturalness of the speech, especially when compared to traditional decompression methods. 4.3 Computational Efficiency Execution Time: The total time taken to decompress the MELP-compressed signal will be measured. Algorithms optimized for real-time performance will be favored, especially in contexts where low-latency decompression is essential. 4.4 Subjective Quality Evaluation (Optional) If applicable, a subjective listening test could be conducted to assess the perceived quality of the decompressed speech. This would provide additional insight into the effectiveness of the decompression algorithm and generative models used. 4.5 Overall Performance The final evaluation score will be a composite of the similarity metrics, generative model effectiveness, computational efficiency, and subjective quality. This holistic approach ensures participants are recognized for both the quality of their decompressed speech and the efficiency of their algorithms. 5. Submission Instructions Submit the source code, documentation, and any pretrained models or training scripts used for deep learning-based solutions. Include a detailed readme file with clear instructions on how to run your decompression algorithm and any dependencies required. Ensure that the decompressed signals are evaluated using the provided similarity metrics and that execution time is measured during the evaluation. 6. Important Notes the MELP compression code will be provided to participants, so you are only required to implement the decompression algorithm. Participants are encouraged, but not required, to use generative models such as neural networks to enhance decompression quality, especially at low bit rates (600 bps or above(optional)). The competition aims to push the boundaries of low-bit rate speech coding by combining traditional signal processing methods with cutting-edge AI techniques. You can download the MELP compression source code here, or you can download it from GitHub or other resources available on the internet.

3. Evaluation

The evaluation of participants' algorithms will be conducted using a combination of similarity metrics and timing assessments to ensure a comprehensive analysis of performance. The following criteria will be used to gauge the effectiveness of the decompression algorithms developed for MELP-compressed signals:

3.1 Similarity Metrics

To assess how closely the decompressed signal resembles the original signal, the following similarity metrics will be employed:

Normalized Cross-Correlation (NCC): This metric measures the similarity between the decompressed and original signals by calculating the correlation coefficient normalized by the energy of the signals. A higher NCC value indicates a closer match, thus demonstrating the algorithm's effectiveness in reconstructing the original signal.

Percent Root Mean Square Difference (PRD): This metric quantifies the difference between the decompressed signal and the original signal as a percentage of the root mean square of the original signal. A lower PRD value signifies better performance, reflecting the ability of the algorithm to minimize distortion and maintain fidelity.

3.2 Computational Efficiency

In addition to quality metrics, the algorithms will be evaluated based on their computational efficiency, specifically:

Execution Time: The total time taken to decompress the MELP-compressed signal will be measured. This assessment will help determine the practicality of the algorithms for real-time applications, particularly in scenarios with strict latency requirements.

3.3 Overall Performance

The final evaluation score for each submission will be a composite of the similarity metrics and computational efficiency. This holistic approach ensures that participants are recognized not only for achieving high-quality decompression but also for optimizing their algorithms to run efficiently under various conditions.

3.4 Justification and Documentation

Participants must provide a rationale for their chosen algorithms and techniques, along with detailed documentation that outlines their approach. This should include explanations of how the similar metrics were optimized, as well as any challenges encountered during implementation and how they were addressed.

4. Important Notes

Participants may implement the MELP compression algorithm independently or utilize publicly available code resources. For reference, below are links to some available resources:

https://github.com/gegel/pairphone/blob/master/melpe/melpe.c

For instance, below are some sites where you can download sample audio files to train your model. Please note that the evaluation will be conducted on a diverse range of datasets, which vary in nature and are not limited to the provided samples. Therefore, participants are advised to train their models using a wide variety of voice categories.

https://www.voices.com/

https://freesound.org/

https://www.kaggle.com/datasets/shabareesharyan/voice-dataset

Your code should be able to read audio files in .mp3, .m4a, and .wav formats.

The deadline for submitting your code, report, and any other necessary materials is December 7, 2024. Please be advised that no submissions will be accepted after this date. We kindly request that all required documents and materials be submitted on time to ensure proper evaluation and consideration of your work. Late submissions will not be processed under any circumstances.

All necessary materials should be submitted in a single zip file to the following site: "AIsoft24.shirazu.ac.ir". Please ensure that all required documents, including the code, report, and any additional materials, are properly compiled and submitted in this format. Submissions that do not adhere to this requirement may not be accepted.