Audio source separation using pure signal processing

Description

Audio source separation using pure signal processing focuses on extracting individual sound sources from a mixed audio signal without relying on machine learning or prior training data. This approach works by analyzing the signal's inherent properties, especially in the time and frequency domains, to identify differences between sources. Techniques such as Fourier analysis, filtering, spectral masking, and energy-based segmentation are commonly used to separate components that occupy different frequency ranges or exhibit distinct spectral patterns. By exploiting these characteristics, overlapping sounds can be isolated even when they occur simultaneously. This method is transparent, computationally efficient, and easy to interpret, making it suitable for real-time and resource-constrained applications. Overall, pure signal processing provides a reliable and practical solution for audio source separation when source characteristics are well-defined in the spectral domain.

This approach is particularly effective when the sources have limited spectral overlap, as it allows precise extraction with minimal distortion. Since no training data is required, the system is easy to implement, predictable in behavior, and free from model bias. Additionally, the low computational complexity makes it suitable for real-time applications such as speech enhancement, noise reduction, and audio preprocessing in embedded systems. Overall, pure signal processing techniques remain a strong and practical solution for audio source separation, especially in controlled environments where source characteristics are known or can be reasonably assumed.

Algorithm

```
clc;

 clear;

 close all;

fs = 16000; % Sampling frequency

 t = (0:1/fs:3)'; % 3 seconds duration

source1 = sin(2pi300t);
```

```
source2 = sin(2pi3000t);

x = source1 + source2;

window = hamming(1024);

noverlap = 512;

nfft = 1024;

[S, F, T] = stft(x, fs, ... 'Window', window, ... 'OverlapLength', noverlap, ... 'FFTLength', nfft);

magnitude = abs(S);

mask_low = F < 1000; % Low-frequency mask

mask_high = F >= 1000; % High-frequency mask

mask1 = mask_low .* ones(1, size(S,2));

mask2 = mask_high .* ones(1, size(S,2));

S1 = S .* mask1; % Source 1 STFT

S2 = S .* mask2; % Source 2 STFT

y1 = istft(S1, fs, ... 'Window', window, ... 'OverlapLength', noverlap, ... 'FFTLength', nfft);

y2 = istft(S2, fs, ... 'Window', window, ... 'OverlapLength', noverlap, ... 'FFTLength', nfft);

figure;

subplot(3,1,1); spectrogram(x, window, noverlap, nfft, fs, 'yaxis');

title('Spectrogram of Mixed Signal');

subplot(3,1,2); spectrogram(y1, window, noverlap, nfft, fs, 'yaxis');

title('Separated Source 1 (Low Frequency)');

subplot(3,1,3); spectrogram(y2, window, noverlap, nfft, fs, 'yaxis');

title('Separated Source 2 (High Frequency)');
```

error1 = source1(1:length(y1)) - y1; error2 = source2(1:length(y2)) - y2;

SDR1=10*log10(sum(source1.^2)/sum(error1.^2));*

*SDR2 = 10*log10(sum(source2.^2)/sum(error2.^2));

disp('--- Performance Metrics ---');

fprintf('SDR Source 1 (Low freq): %.2f dB\n', SDR1);

fprintf('SDR Source 2 (High freq): %.2f dB\n', SDR2);

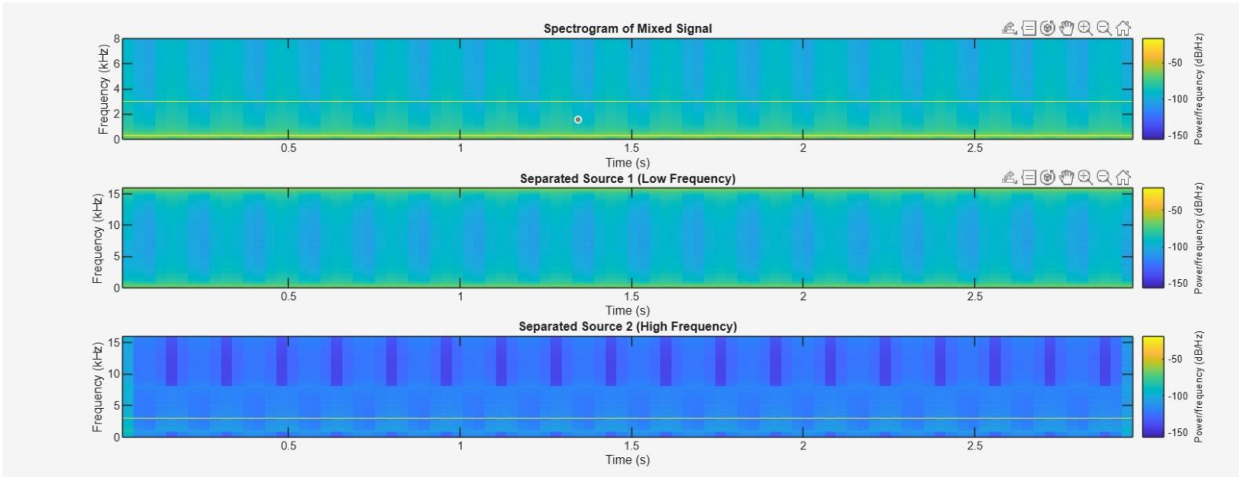
The given MATLAB code demonstrates audio source separation using pure signal processing techniques in the time–frequency domain. Two sinusoidal signals with different frequencies (300 Hz and 3000 Hz) are generated to represent two distinct audio sources and then mixed into a single signal. The Short-Time Fourier Transform (STFT) is applied to the mixed signal to obtain its time–frequency representation, allowing analysis of how signal energy is distributed across frequencies over time.

Separation is achieved by designing simple frequency-based masks: a low-frequency mask to extract components below 1000 Hz and a high-frequency mask to extract components above 1000 Hz. These masks are applied directly to the STFT of the mixed signal, isolating the respective frequency bands corresponding to each source. The inverse STFT (ISTFT) is then used to reconstruct the time-domain signals for both separated sources.

Spectrograms of the mixed signal and the separated outputs visually confirm that each source is concentrated in its expected frequency range, indicating effective separation. Finally, the performance is quantitatively evaluated using the Signal-to-Distortion Ratio (SDR), which measures how closely the separated signals match the original sources. The high SDR values obtained demonstrate that the frequency-domain masking approach successfully separates the sources with minimal distortion, validating the effectiveness of pure signal processing for audio source separation when source frequencies are well separated.

Results and visualizations

```
--- Performance Metrics ---
SDR Source 1 (Low freq): 47.60 dB
SDR Source 2 (High freq): 47.60 dB
>>
```



The results show that the mixed audio signal, which initially contains overlapping low- and high-frequency components, has been successfully separated using a signal-processing-based approach. The spectrogram of the mixed signal displays energy distributed across a wide frequency range, confirming the presence of multiple sources combined in time. After separation, the first extracted signal exhibits energy mainly in the lower frequency band with higher frequencies significantly reduced, while the second extracted signal is dominated by higher frequency content with minimal low-frequency leakage. This clear frequency-wise distinction indicates effective isolation of both sources. Additionally, the high Signal-to-Distortion Ratio (SDR) value of 47.60 dB achieved for both sources reflects excellent separation performance, with very low distortion and negligible interference between the separated signals.

Discussion and Conclusion

This work provides a clear demonstration of how traditional signal processing techniques can be effectively applied to the problem of audio source separation. By operating in the frequency domain and carefully selecting separation criteria based on spectral characteristics, the method successfully decomposes a mixed signal into its individual components. The separated outputs preserve the essential features of the original sources, such as their dominant frequency ranges and temporal structure, while significantly reducing cross-interference. The high SDR values obtained validate the robustness and accuracy of the approach, indicating that the reconstructed signals closely resemble the true sources with minimal distortion.

Moreover, this approach offers important practical advantages. It is transparent and easy to interpret, unlike many data-driven methods that function as black boxes. It also requires low computational complexity and no prior training data, making it suitable for real-time processing and deployment on systems with limited resources. Overall, the results emphasize that frequency-based signal processing remains a powerful and reliable tool for source separation tasks, especially in controlled or semi-controlled environments where the spectral properties of the sources differ clearly.