2EC502

DIGITAL SIGNAL PROCESSING

COMPREHENSIVE EVALUATION REPORT

GUIDED BY:

PROF. RUTUL PATEL

DR. BHUPENDRA FATANIYA

SUBMITTED BY:

SHUBHAM SONI (18BEC110)

PERCEPTUAL CODING

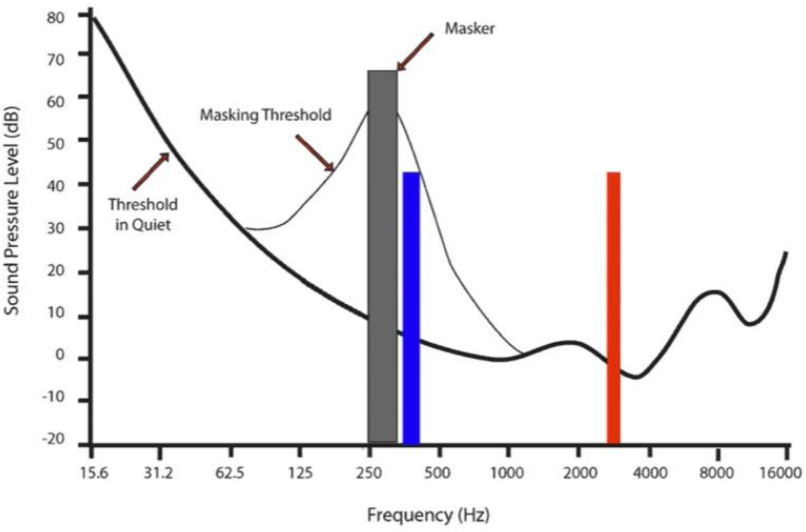
Objective:

In the recent years, high quality audio and video has replaced CD-quality digital audio. The emerging applications in various domains like multimedia systems and wireless network face a series of problems like channel bandwidth and limited storage. This has created a demand for compression of audio and video at low cost. As a result, many algorithms have been proposed which are now used internationally and commercially and one of them is perceptual coding.

Introduction

Perceptual coding is a method of coding which reduces the amount of data needed to make high-quality of sound. This coding method takes advantage of the human ear, screening out a certain amount of sound which is perceived as noise. The perceptual coding would screen out or remove the sound which the human ears perceives as noise and thus reducing the size of the audio file. Thus, perceptual coding can be defined as compression technique which removes the extra noise from the digital audio.

The weaker tunes are supressed as compared to the louder tunes. A threshold is selected at louder tune frequency due to which the weaker tune is not audible which is known as masking effect. In the few years since the first systems and the first standardization efforts, perceptual coding of audio signals has found its way to a growing number of consumer applications. In addition, the technology has been used for a large number of low volume professional applications.

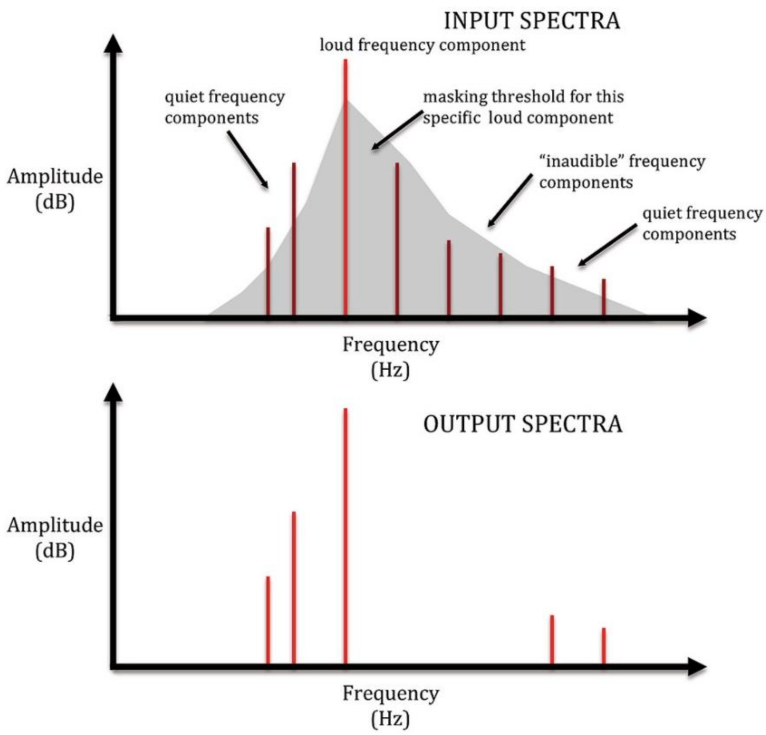


All Perceptual audio coding depends on the audio masking effect which is proven to be human's natural auditory process. The above curve represents the hearing sensitivity with respect to audio frequency. The thick black line represents the threshold in quiet, so a human being can't hear anything of sound pressure level below that. If we add a tone which is shown as the grey masker in the graph and modify the curve, now any audio having pressure level below that particular value won't be audible to the ear of a healthy human being. So the blue bar shown known as maskee in the curve at 400Hz won't be audible by the ear and doesn't need to be coded by any perceptual codec. Further a person can hear the sound at 3000 Hz represented by the red bar and thus it needs to be coded. Thus, only the peaks of the audio need to be coded and thus the size compression can be achieved.

The psychoacoustic principles have made enough progress towards characterization of the human ear that the main aim of perceptual coding lies behind the human auditory facts.

Perceptual algorithms are based on the two features of the human ear:

1) The human ear sensitivity is not same for all types of frequencies.
2) Because of the louder tone, the weaker tone is not audible.



In the above input frequency spectra, the components that are below the grey masking area need not to be coded as they would not be audible for a human ear, while the threshold components above the masking area needs to be coded as they are the loud frequency components. Thus, the output spectra will be achieved as shown above. The bit allocation should be done in such a way that the noise level resulting out of the quantization noise should never exceed the masking level and as long as it is kept below the masking level, no extra bits are needed to allocate and thus bit saving can be done using the masking phenomenon.

Absolute threshold of hearing is typically expressed in terms of sound pressure level (dB). The quiet threshold can be explained by the following non-linear function:

$$T_q(f) = 3.64\left(\frac{f}{1000}\right)^{-0.8} - 6.5e^{-0.6\left(\frac{f}{1000}-3.3\right)^2} + 10^{-3}\left(\frac{f}{1000}\right)^4 \ (dB\ SPL)$$

which is representative of a young acute ear hearing.

Critical Bands

The first step of the perceptual coding is to shape the coding distortion spectrum. The detection threshold for quantization noise is a modified version of the absolute threshold, with its shape defined by the stimuli pre-sent at any given time. The critical bandwidth can be defined as a function of frequency that quantifies the cochlear filter passbands.

Critical bandwidth can also be referred to as the result of auditory detection efficacy in accordance with SNR principles. For eg: The detection threshold which is present for a narrowband noise source presented between two masking tones remains constant as long as the frequency difference between the tones remain within the critical bandwidth.

As long as the masking tones are introduced within the passband of the auditory filter (critical bandwidth) that is targeted or tuned to the probe noise, the SNR presented to the auditory system remains constant, and hence the detection threshold does not change. As the tones spread further apart and transit into the filter stopband, however, the SNR introduced to the auditory system improves, and which is why the detection task becomes easier, ultimately causing the detection threshold to decrease for the probe noise.

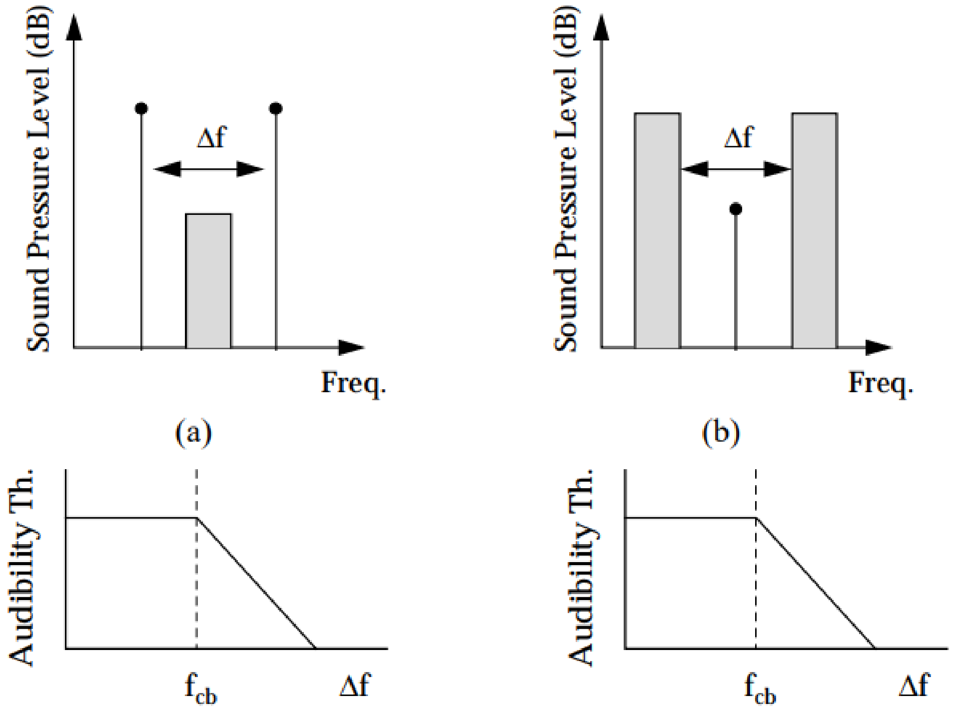


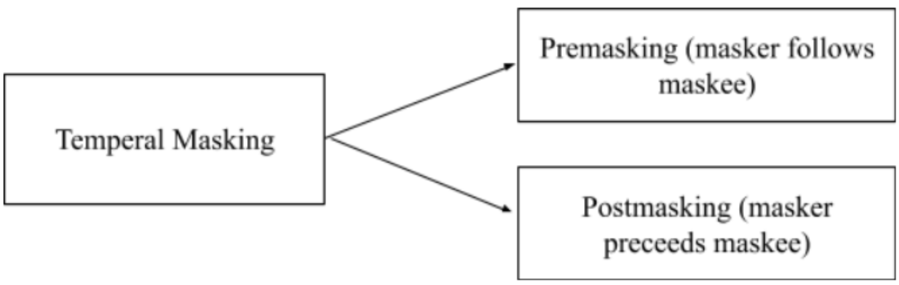
Figure taken from (IEEE paper on perceptual coding for digital audio)[1]

A notched-noise experiment having a similar interpretation can be achieved with masker and maskee roles reversed. Critical bandwidth tends to remain constant (about 100 Hz) up to 500 Hz, and increases to about 20% of the center frequency above 500 Hz. For an average listener, critical bandwidth is conveniently calculated or approximated by:

$$BW(f) = 25 + 75\left[1 + 1.4\left(\frac{f}{1000}\right)^2\right]^{0.69}$$

Temporal Masking

Other than simultaneous masking, temporal masking also exists in which masker and maskee may not happen at the same instant. as shown in the figure



If we have a tone and when the tone stops, it takes a moment for the mask to get back to the initial pressure level. Due to the post masking period of 75-100ms, the human brain would not interpret the sound even though the sound has stopped. Similarly, in the pre-masking period the sound level below the threshold value won’t be heard.

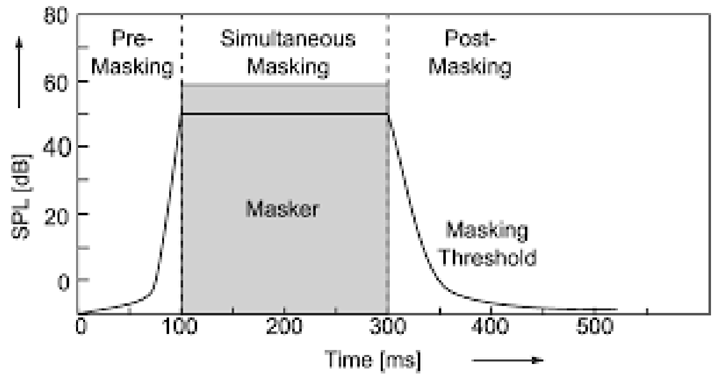


Figure taken from (IEEE paper on perceptual coding for digital audio)[1]

Filter banks for Audio Coding

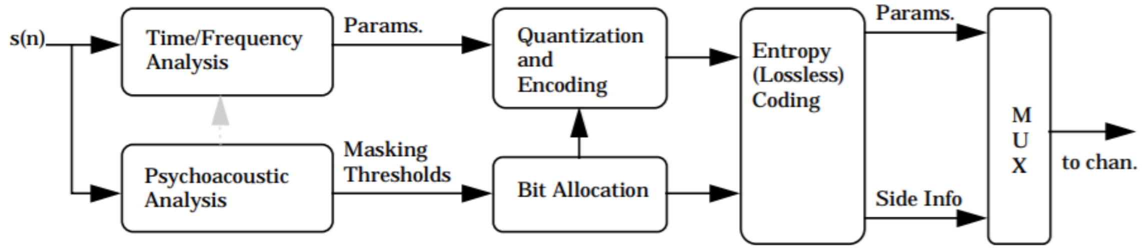
In actual form, the frequency masking is continuous in nature but in a practical codec it is applies over bands. Hence, first the signal is divided in frequency bands and then identification is done if any frequency bands will go un-audible in the presence of frequency bands which are louder. Thus, filter banks are used. The success of the perceptual audio coder depends heavily upon the type of filter bank used. Adequate matching of the filter bank to the characteristics of the input signal is critical for the efficient coding. Following are the bank characteristics which are highly desirable for audio coding:

- Good channel separation
- Strong Stop Band attenuation
- Critical Sampling
- Accurate reconstruction

Filter banks enable lossless reconstruction on the signal which is given as input in an analysis system if the quantization is not used. Usually for perceptual coding, MDCT filter banks are used.

Perceptual Audio Encoder

MPEG Audio uses perceptual encoding to remove parts of the signal that most people can’t hear. The encoder also applies standard lossless data-compression techniques to compress the audio even more. The amount of information discarded, and therefore the sound quality, is dependent on parameters (such as bit-rate and sampling rate) that are chosen by the creator.

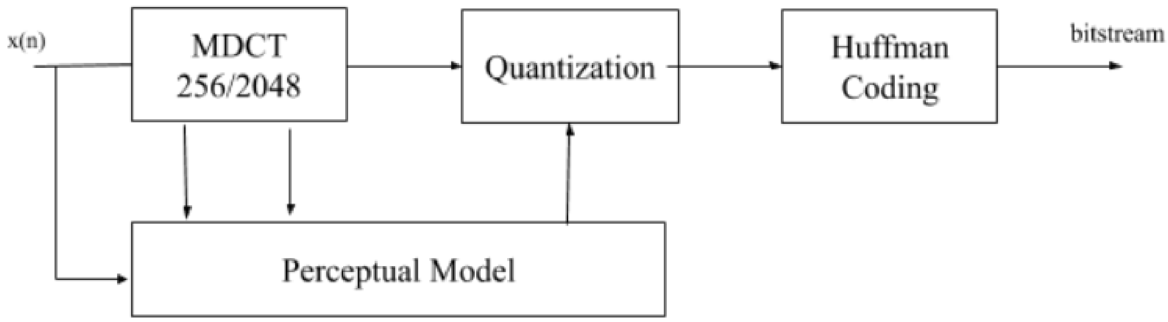


The above figure shows a generic perceptual encoder. Perceptual distortion control is achieved by a psychoacoustic signal analysis that predicts the masking power of the signal which are based on the psychoacoustic principles which are derived.

The thresholds are delivered by the psychoacoustic model and it allows encoding section to exploit perceptual irrelevancies in the time-frequency set. Further the quantization can be uniform and it might be performed on scalar or vector data. On the formation of quantized compact parametric set, the remaining redundancies can be removed using entropy coding like Huffman coding, arithmetic or LZW coding. Since the output of psychoacoustic distortion control model is signal dependent, the algorithms are inherently variable rate.

Problem Statement: To achieve compression of mp3 audio/speech.

Block Diagram:



Matlab Code:

```
% The compression of audio file is achieved using MDCT filter
bank, masking threshold and huffman coding
```

```
clear all;
clc;


blockLength = 1024;
inputFile = 'C:\Users\shubh\Documents\MATLAB\18BEC110.mp3'

% read and normalize audio file
[x,fs] = audioread(inputFile);
x = x/max(x);


% get size of original file to compare later
original = dir(inputFile);
original_size = original.bytes;

% compute how many blocks are needed and allocate the input
block matrix
numBlocks = floor(length(x)/blockLength);
X = zeros(numBlocks, blockLength);

% blocking of the input audio data
for k = 1:numBlocks
    for m = 1:blockLength
        X(k,m) = x((k-1)*blockLength + m);
    end
end

%% Analysis filter bank and quantization

Y_mdct = mdct_analysis(X);
[Y_quant, masking_thresh] = masking_quantization(Y_mdct, X,
fs);

% Plot one masking curve to get an impression
figure(1);
plot(masking_thresh(:,10), 'b');
hold on;
S = 20*log10(abs(fft(X(10,:))));
plot(S(1:blockLength/2));
legend('Masking threshold', 'FFT spectrum');
xlabel('FFT bin'),
ylabel('Amplitude');
hold off;

%% Encode and write bitstream to file

% cell array to hold the huffman dictionaries for the M
different blocks
dictionary{numBlocks} = [];
```

```
[~, fn, ~] = fileparts(inputFile);
codedFile =
['C:\Users\shubh\Documents\MATLAB\18BEC110_encoded.mp3' fn
'_coded'];
fileID = fopen(codedFile,'w');

% vector holding the lengths of the bitstreams for each block,
this is
% used for reading out the data later
bitstreamlength = zeros(1,numBlocks);

for m = 1:numBlocks
 % create huffman dictionary of one block of MDCT coefficients
    dictionary{m} = huffman_table(Y_quant(m,:));
    % do the huffman encoding and write to file
    output = huffmanenco(Y_quant(m,:),dictionary{m});
    bitstreamlength(m) = length(output);
    fwrite(fileID, output, 'ubit1');
end
fclose(fileID);

% compute compression rate
coded = dir(codedFile);
coded_size = coded.bytes;
disp(['The compression rate is ' num2str(100*(1 -
coded_size/original_size)) '%']);

% compare the waveforms of the original and decoded signal
figure(2), subplot(2,1,1), plot(x), title('Original audio');
subplot(2,1,2), plot(x_encoded), title('Encoded audio');
```

Working:

The following steps are undergone for achievement of perceptual coding:

1. The audio file in mp3 format is read with the help of audioread function and it is normalized.

2. Further the size of the audio file is stored in a variable.

3. The computing of number of blocks needed and allocation of input block matrix is done.

4. Quantization is done using the MDCT filter bank.

5. Using the user-defined function maskingthresh the masking curve is plotted.

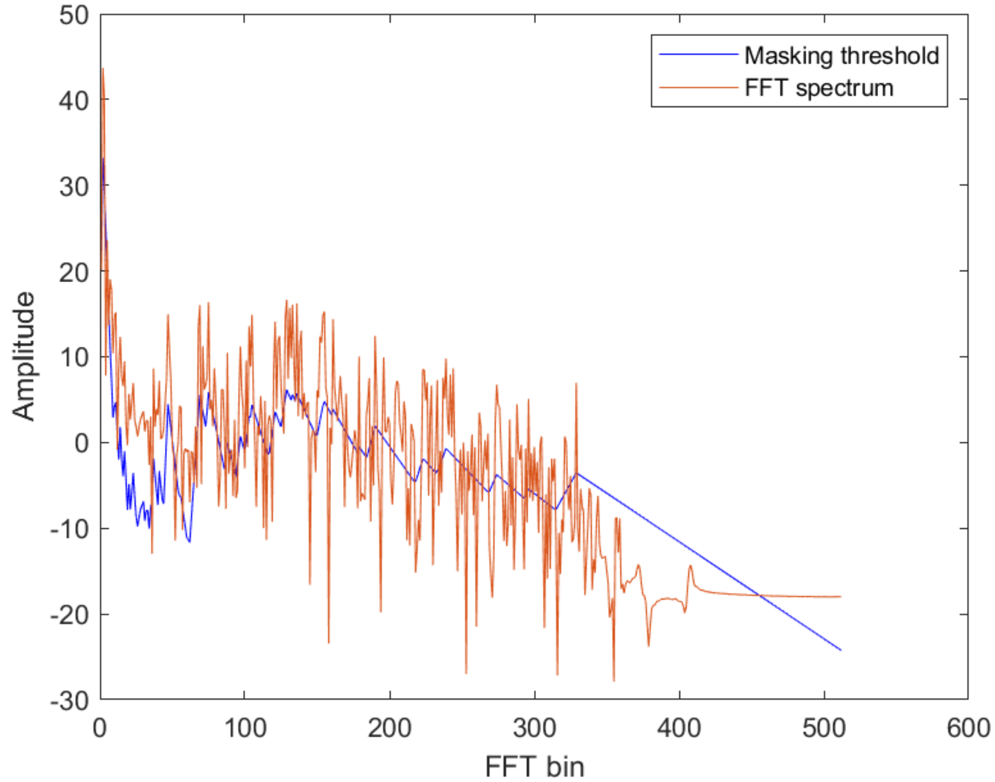
6. Cell array to hold the huffman dictionaries for the M different blocks is initialized and the Huffman dictionaries are created for blocks of MDCT coefficients.

7. Huffman encoding and writing to the file

8. In the last step, the compression rate is calculated and the waveforms of original and encoded signal is plotted.

Output:

The size of the input mp3 file is 816 kB



The masking threshold obtained is shown by the blue curve and the frequency spectra is shown by the red curve. The audio having amplitude below the masking area need not be coded.

```
inputFile =

    'C:\Users\shubh\Documents\MATLAB\18BEC110.mp3'

The compression rate is 70.5679%
```

The size of the compressed audio file saved in the system is 241 kB.

18BEC110	26-11-2020 13:21	MP3 File	816 KB
18BEC110_encoded	30-11-2020 12:14	File	241 KB

The achieved compression rate of mp3 audio is 70%

Application areas of perceptual audio coding:

Current application areas include

- Digital Broadcasting and Satellite Broadcasting.
- Accompanying audio for digital video: This includes all of digital TV.
- Storage of music including hard disc recording for the broadcasting environment.
- Audio transmission for FM broadcast stations
- Audio transmission via the Internet
- In MPEG and AC-3 audio coding

Conclusion

In this report on perceptual coding, we looked into the necessity of compression techniques in objective section. We also built the practical model for the compression of mp3 audio using MATLAB. Further we have seen concepts like critical bands, MDCT filter banks, masking and an introduction to psychoacoustic model. At the end, we have also seen the applications of perceptual coding.

References

1. IEEE paper on Perceptual Coding for digital audio by Ted Painer
2. IEEE paper on Filter Banks for perceptual coding by Marina Josh
3. Perceptual Audio Coder on WikiPedia