

# Tonal and Non-Tonal Components for the MPEG-1 Model

VLADISLAV SKORPIL, ABDULHAKIM ABUZAHO

Department of Telecommunications

Brno University of Technology

Purkynova 118, 612 00 BRNO, Czech Republic,

CZECH REPUBLIC

---

*Abstract:* This paper describes one part of the digital data compression of multimedia transmission system research. The whole research is focused to encoding and decoding process of MPEG-1 standard. MATLAB is used for simulation of mathematical calculations. The part, which is describes in this paper, is the tonality of a masking components. The tonality of a masking component has an influence on the masking threshold. For this reason, it is worthwhile to discriminate between tonal and non-tonal components. For calculating the global masking threshold, it is necessary to derive the tonal and non-tonal components from the FFT.

*Key words:* MPEG -1, MATLAB, Tonal, Non-tonal, labelling, local maximum

## 1 Introduction

Digital audio compression is a scheme, which allows efficient storage and transmission of audio data. The main objective of this research is to show the program in MATLAB for MPEG-1 audio encoder and decoder for layer-1. Compression is required for efficient transmission in order to send more data in the available bandwidth, or to send the same data in less bandwidth and more users can use it on the same bandwidth. It can also be used for storage purposes to compress more data and can compress for local storage, put details on cheaper media. It is also useful for progressive reconstruction, scalable delivery, browsing and as a front end to other signal processing.

The whole research was divided to 15 areas.

Mapping the actual status of audio signal encoding and bit rate compression

Focused to MPEG-1 method

Encoder and decoder MPEG-1 block diagrams

Design of MPEG-1 model simulated by MATLAB.

This model will be suitable for testing and optimizing of MPEG-1 principle

Analysis and simulation of the Filter Bank

Design and modification of scale-factor calculating

Modeling of FFT for MPEG-1

Determination of the sound pressure level

Study of tonal and non-tonal components

Decimation of tonal and non-tonal masking components design

Individual masking threshold and global masking threshold

Determining the minimum masking threshold, signal-to-mask ratio

Design of bit allocation

Optimal quantisation and encoding of subbands samples

To make conclusion for optimizing of MPEG-1 model and for DSP assembler program for future work

## 2 Problem Formulation

The step finding of tonal and non-tonal components was divided into two parts. First is aimed to find tones and second to find noise with high power, which is necessary to transmit. In fact, it is named noise, but it should be many tones closer together. In addition, there is one more part called 'labelling of local maxima', but it is concluded in the part for tones.

This step of standard starts with labelling of local maxima. It then extracts tonal components (sinusoids) and calculates the intensity of the non-tonal components within a bandwidth of a critical band. The bandwidth of the critical band varies with the frequency with a bandwidth of about 0.1 kHz at low and with a bandwidth of about 4 kHz at high frequencies. It is known from psychoacoustic experiments that the ear has a better frequency resolution in the lower frequency region than in the higher ones [1]. To determine if a local maximum is examined, the frequency range  $df$  is given by standard for all sampling frequencies of MPEG-1.

For further information about df for different frequencies and different layers is better to study the standard [1].

### 3 Listing of tonal components and calculation of the sound pressure level

A local maximum is put in the list of tonal components if

$$X(k) - X(k + j) \geq 7 \text{ dB} \quad (1)$$

where j is chosen according to Layer-I

$$\begin{aligned} j &= -2, +2 && \text{for } 2 < k < 63 \\ j &= -3, -2, +2, \dots, +6 && \text{for } 63 \leq k < 127 \\ j &= -6, \dots, -2, +2, \dots, +6 && \text{for } 127 \leq k \leq 250 \end{aligned} \quad (2)$$

If X(k) is found to be a tonal component, then the tonal flag is set up, and the sound pressure level is computed:

$$X_m(k) = 10 \log_{10} \left( 10^{\frac{X(k-1)}{10}} + 10^{\frac{X(k)}{10}} + 10^{\frac{X(k+1)}{10}} \right) \text{ dB} \quad (3)$$

Next spectral lines within the examined frequency range are set to  $-\infty$ .

### 4 Tones under MATLAB

This part of psychoacoustic model is good for giving some graphical information for the user. Here some parts of the tonal labelling in MATLAB is shown. According to (2), variable j is called run and it is pointed at the maximal spaces around the tone. Later, if it was labeled as tone, program calculates sound pressure level according to (3). After that it is everything around the tone to set up on very low sound pressure level. It is constantly defined by author to tell the program that there is no signal. Now tonal labelling by plots from MATLAB will be introduced. Firstly, it shows how are data labeled after a local maxima. For it in MATLAB variable enspr is used. The test signal is the same as before.

After starting labelling of tones according to Equations (1) and (2). This cuts many local maximums and keep only tones as shown in Fig. (1).

Next picture shows what is done after labelling. In frequency spectrum is power around a tone labeled as minimum of energy is presented in Fig.2 But on position of tone, new value is calculated from surrounding levels according to (3). As pictures show, this calculation is very efficient and is

functioning very well. After tone labelling of other important signals in input audio signal is started.

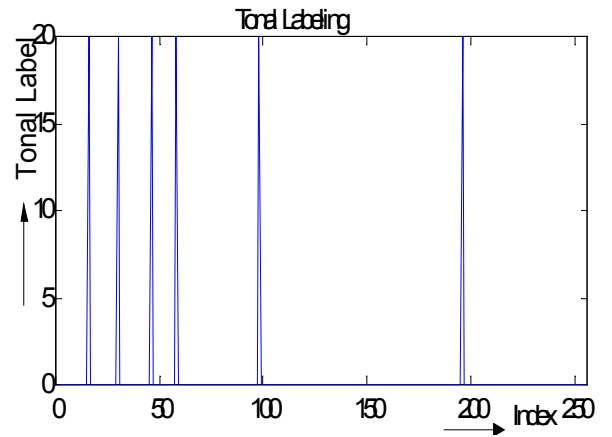


Fig. 1. Tone labels after labelling test signal

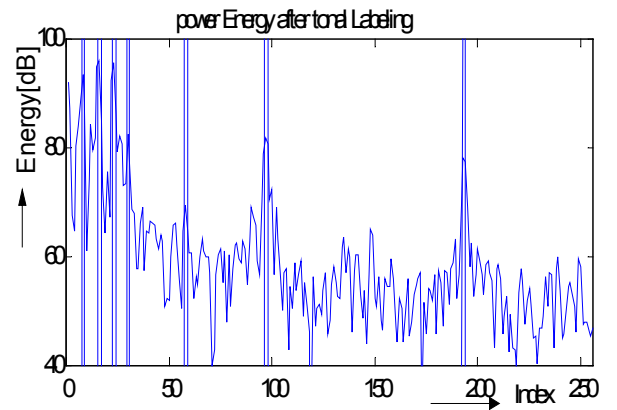


Fig. 2. New energy value for tones and surrounding positions

### 5 Listing of Non-Tonal Components and Calculation of Power

The remaining spectral lines are used to calculate the non-tonal (noise) components. To calculate the non-tonal components from these spectral lines  $X(k)$ , the critical bands are determined using the 'Critical band boundaries' table (Table 1.). In Layer 1, 24 critical bands for  $F_s = 44.1\text{kHz}$  are used. Within each critical band, the powers of the spectral lines (remaining after the tonal components have been zeroed) are added together to form the sound pressure level of the new non-tonal component  $X_{nm}(k)$  that corresponds to the given critical band. Its index number will be the index number  $k$  of the spectral line nearest to the geometric mean of the critical band. In addition, the non-tonal flag is set up.

Table 1 Critical band boundaries,  $F_s = 44.1$  kHz

| Critical Bound | Index of Spectral Line for Boundary | Critical Bound | Index of Spectral Line for boundary |
|----------------|-------------------------------------|----------------|-------------------------------------|
| 0              | 1                                   | 13             | 27                                  |
| 1              | 2                                   | 14             | 32                                  |
| 2              | 3                                   | 15             | 37                                  |
| 3              | 5                                   | 16             | 45                                  |
| 4              | 6                                   | 17             | 52                                  |
| 5              | 8                                   | 18             | 62                                  |
| 6              | 9                                   | 19             | 74                                  |
| 7              | 11                                  | 20             | 88                                  |
| 8              | 13                                  | 21             | 108                                 |
| 9              | 15                                  | 22             | 132                                 |
| 10             | 17                                  | 23             | 180                                 |
| 11             | 20                                  | 24             | 232                                 |
| 12             | 23                                  | ***            | *****                               |

## 6 Non-Tonal Components in MATLAB

The standard was strictly followed in this program. In first step, the sound pressure level is calculated by summing all energy power of the spectral lines between critical boundaries. These are taken from Table 1. As is written below, **tones are not added**. For example, the sound-pressure level for critical bound number 14 is composed of the summed spectral lines 27+28+29+30+31, if no one of them is a tone. Moreover, when a signal was previously zeroed, it is not used for adding too. Also, all added values are zeroed. In the second step, the geometric mean is calculated from the indexes of the critical boundaries' spectral lines. In third step, the calculated value for all energy in the critical sub band on the geometric mean position is saved. It receives the label NOISE (tones have the label TONE). These three steps are done through all critical boundaries. There are 24 for  $F_s = 44.1$  kHz. **Note:** Figure 3 shows in the end no change to energy power. This is due to the value of the last boundary, the last critical band. In present example from Table 1 the number 232 is given; this is the last boundary number of the last spectral line in the summing procedure. This means that spectral lines with higher indexes will remain unchanged, as can be observed in the figure.

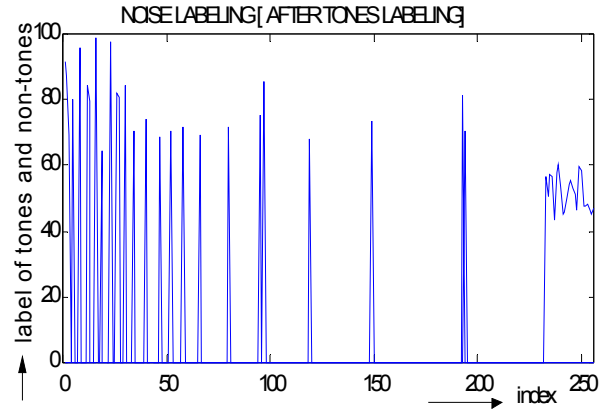


Fig. 3 New energy values after noise labelling

## 7 Decimation of Tonal and Non-Tonal Masking Components

Decimation is a procedure used to allow considering fewer maskers when calculating the global masking threshold. The first operation reduces the tonal  $X_{im}(k)$ , or non-tonal components  $X_{NM}(k)$ . These are considered for the calculation of the masking threshold only if

$$X_{im}(k) \geq LT_q(k) \quad \text{or} \quad X_{nm}(k) \geq LT_q(k) \quad (4)$$

In this expression,  $LT_q(k)$  is the absolute threshold (the "threshold in quiet") at the frequency of index  $k$ . The second operation reduces the number of tonal components that are within a distance of less than 0.5 Bark units keeping the component with higher power, and removing the one with a smaller pressure level. A Bark is defined as the unit of the human ear's critical band rate, and Bark values corresponding to the sub band samples are presented in tables in the MPEG-1 standard.

**Note:** in the next calculation, tonal and non-tonal components are merely indexed. Tones and noise are not specially separated. The character  $j$  serves as an index.

## 8 Decimation in MATLAB

The decimation was separated into three steps. Each step is represented by one loop, with a comparison at the beginning. After the comparison, a tone or noise is kept or removed, depending on the comparison's result. These three parts are decimation for tonal components, decimation for non-tonal components, and decimation for tonal components within a distance of less than 0.5 Bark. The pictures show how many spectral lines were deleted in the calculation of a tested signal. Three

steps were used to determine which ones to delete. But one was imported for the tested signal. It was step two, because steps one and three are for tonal components. There are only two tones, and both had a high power; therefore, both were left in the first step. A large distance also separates them. Step three thus also had no effect, and so the second step removed many spectral lines. This depended on the value of the power. The surrounding tones are high noise values, caused by filters. These are far from ideal, and so each filter lets some signals through from nearby sub bands.

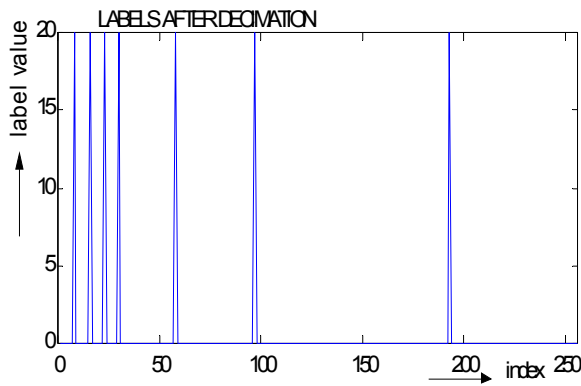


Fig. 4 New labels (types) are set up after decimation process.

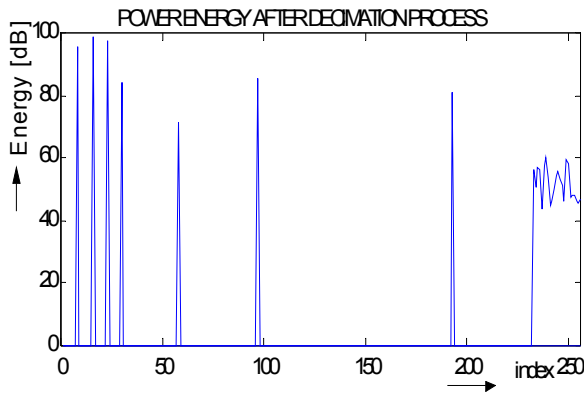


Fig. 5 According to new labels a lot of spectral lines was removed

## 9 Calculating the Individual Masking Threshold

The individual masking thresholds of both tonal and non-tonal components are calculated according to the MPEG-1 standard [1]. The following expressions are applied:

$$LT_{tm}[z(j), z(i)] = X_{tm}[z(j)] + av_{tm}[z(j)] + vf[z(j), z(i)] \quad (5)$$

$$LT_{nm}[z(j), z(i)] = X_{nm}[z(j)] + av_{nm}[z(j)] + vf[z(j), z(i)] \quad (6)$$

Where  $LT_{tm}$  and  $LT_{nm}$  are the individual masking thresholds at critical band rate  $z$  in Bark of the masking component at the critical band rate of the masker  $z_m$  in Bark. The value in dB can be either positive or negative. The term  $X_{tm}[z(j)]$  is the sound pressure level of the masking component with the index number  $j$  at the corresponding critical band rate  $z(j)$ . The term  $av$  is called the masking index, and  $vf$  the masking component  $X_{tm}[z(j)]$ . The masking index  $av$  is different for tonal and non-tonal maskers ( $av_{tm}$  and  $av_{nm}$ ).

For tonal maskers, it is given by

$$av_{tm} = -1.525 - 0.275 * z(j) - 4.5 \text{ dB} \quad (7)$$

and for non-tonal maskers, by

$$av_{nm} = -1.525 - 0.175 * z(j) - 0.5 \text{ dB} \quad (8)$$

The masking function  $vf$  of a masker is characterized by different lower and upper slopes, which depend on the distance in Bark  $dz = z(i) - z(j)$  to the masker. In this expression,  $i$  is the index of the spectral line at which the masking function is calculated and  $j$  that of the masker. The critical band rates  $z(j)$  and  $z(i)$  can be found in the tables in the standard. These are the same tables as before. The masking function, which is the same for tonal and non-tonal maskers, is given by

$$vf = 17 * (dz + 1) - (0.4 * X[z(j)] + 6) \text{ dB} \quad \text{for } -3 \leq dz \leq -1 \quad (9)$$

$$vf = (0.4 * X[z(j)] + 6) * dz \text{ dB} \quad \text{for } -1 \leq dz < 0 \quad (10)$$

$$vf = -17 * dz \text{ dB} \quad \text{for } 0 \leq dz < 1 \quad (11)$$

$$vf = -(dz - 1) * (17 - 0.15 * X[z(j)]) - 17 \text{ dB} \quad \text{for } 1 \leq dz < 8 \quad (12)$$

In these expressions,  $X[z(j)]$  is the sound pressure level of the  $j$ 'th-masking component in dB. For reasons of implementation complexity, the masking is no longer considered ( $LT_{tm}$  and  $LT_{nm}$  are set to  $-\infty$  dB outside this range) if  $dz < -3$  Bark, or  $dz \geq 9$  Bark).

## 10 Individual Masking Thresholds in MATLAB

This part of the psycho acoustic model consists of many calculations. There are two main loops, one for tonal components, and the other for non-tonal components. In each loop, a calculation is done for  $LT_{tm}$  and  $LT_{nm}$  (depending on whether it is a tonal loop or a non-tonal loop). These outputs are used immediately in the next procedure to compute the ‘minimum masking threshold.’

The labels are TONE and NOISE. This takes less time to compute. When decimated-tonal equals 10, it means it is a non-tonal component. If it equals 20, it is a tonal component. This principle is very efficient. No conflict is possible, because the tone and non-tone do not have the same position. Moreover, in the previous procedure, ‘Finding of non-tonal components,’ the label was checked every time. If it was a tone, it was not changed, and the noise power was preserved for the next position (it cannot be a tone, because two local maxim cannot be close to each other).

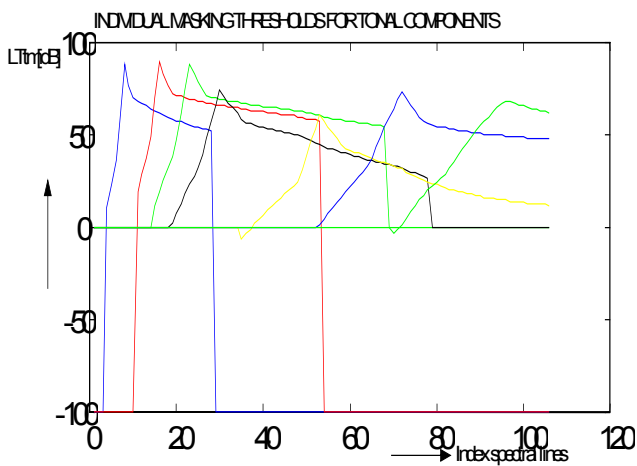


Fig.6 Individual masking threshold for tested signal for tonal components

## 11 Calculation of the Global Masking Threshold $LT_{gg}$

The global masking threshold  $LT_g(i)$  at the  $i$ 's frequency sample is derived from the upper and lower slopes of the individual masking thresholds of each of the  $j$  tonal and non-tonal maskers and from  $LT_q(i)$ , the threshold in quiet. The global masking threshold is found by summing the powers

corresponding to the individual masking thresholds and the threshold-in-quiet.

## 12 Global Masking in MATLAB

This part of the calculation was entirely included in the previous section. As was described, to determine global masking, we add up the powers corresponding to the individual masking threshold (tones and non-tones) and the threshold in quiet (defined in the standard).

The decision was made to include this part, based on addition alone, in the previous section. This is ‘Computing of the individual masking threshold.’ This section had two parts, for tones and non-tones.  $LT_g$  is initialised in the program using  $LT_q$  before the starting of the two loops. After this, at the end of the first loop (for tones)  $LT_{tm}$  is added, and  $LT_{nm}$  is added at the end of loop two (for non-tonal components). All of this is run many times, up to the last spectral line (number 106 for  $F_s = 44.1\text{kHz}$ ).

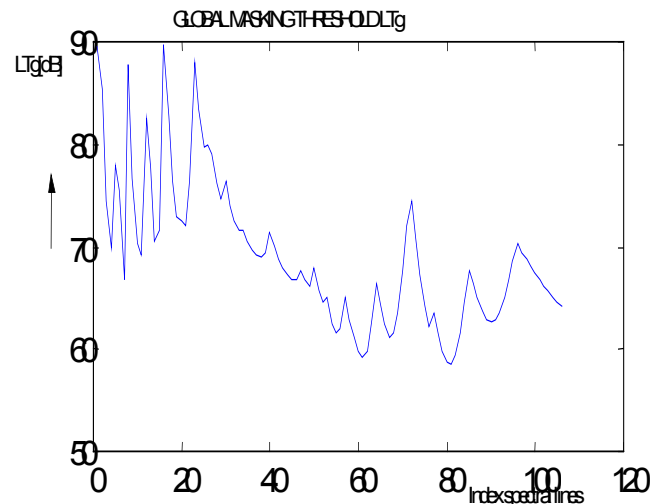


Fig. 7 Global masking threshold for tested signal

## 13 Determining the Minimum Masking Threshold

The minimum masking level,  $LT_{min}(n)$  in sub band  $n$  is determined by the following expression in the MPEG-1 standard:

$$LT_{\min}(n) = \underset{f(i) \text{ in subband } n}{\text{MIN}} [LT_g(i)] \quad (13)$$



Where  $f(i)$  is the frequency of the  $i$ 'th frequency sample. This variable is tabulated in the same table, D1, in the standard as before. But in the present programs, only indexes are used, because this gives us the possibility of changing the sampling frequency with just a few changes.

## 14 Minimum Masking Threshold in MATLAB

Very disorienting indexing is used in this part of the program, but this algorithm is very efficient. A number of spectral lines ( $dl\_line$ ) are requested from the program; this allows—via changes to the table—simple exchanges of the sampling frequency.

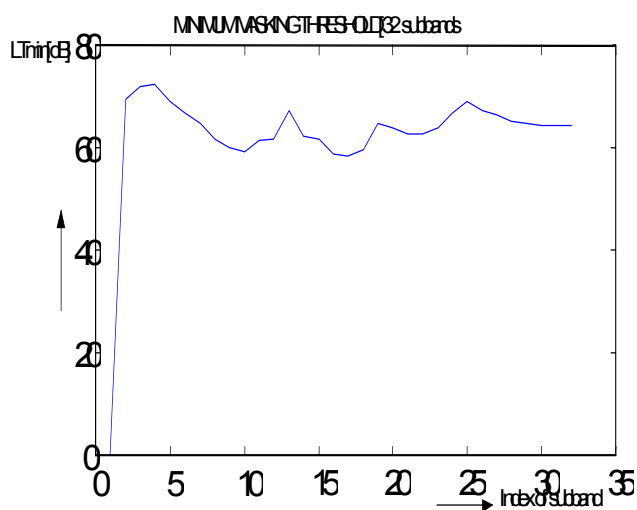


Fig. 8 Minimum-masking threshold for tested signal

## 15 Conclusion

The research, part of which is described in this paper, is focused to the optimizing of data compression for audio signal used in advanced multimedia transmission. For this was designed using of MATLAB program for the testing and simulation of selected theoretical conclusions from the theoretical analysis of the MPEG audio signal compression. The simulation results were used for optimizing of the MPEG compression principle and they are ready for future real time signal processor programming. The main parts of the thesis are: Mapping the actual status of audio signal encoding and bit rate compression, focusing to MPEG-1 method, design of encoder and decoder MPEG-1

block diagrams, design of MPEG-1 model simulated by MATLAB, analysis and simulation of the Filter Bank, design and modification of scale-factor calculating, modelling of FFT for MPEG-1, determination of the sound pressure level, decimation of tonal and non-tonal masking components design, individual masking threshold and global masking threshold, determining the minimum masking threshold, signal-to mask ratio, design of bit allocation, optimal quantization and encoding of subbands samples.

## Acknowledgement

This research was supported by the grants: No 102/03/0434 Limits for broad-band signal transmission on the twisted pairs and other system co-existence. The Grant Agency of the Czech Republic (GACR)

No 102/03/0260 Development of network communication application programming interface for new generation of mobile and wireless terminals. The Grant Agency of the Czech Republic (GACR)

No CZ 400011(CEZ 262200011) Research of communication systems and technologies (Research design) Grant New approach to the subject High-speed Communication Systems (grant of the Czech Ministry of Education, Youth and Sports)

## References:

- [1] EN ISO/IEC 11172-5 - Information technology – Coding of Moving Pictures and Associated Audio for Digital Storage Media up to about 1,5 Mbit/
- [2] SERANTES, C., PENA, A. and PRELCIC, N. A Fast NOISE-scaling Algorithm for Uniform Quantization in Audio Coding schemes. IEEE Proceedings, 1997, p 339-342
- [3] ISO/IEC 13818-7 - Information Technology, Generic coding of moving pictures and Associated Audio Information.
- [4] BOSI, M. and BRANDENBURG, K. ISO/IEC MPEG-2 Advanced Audio Coding. Journal of the Audio Engineering Society, Vol.45, 1997, No. 10, p. 789-814
- [5] ABUZAHU, A. Digital Audio Compression, Doctoral Thesis, BUT, Brno 2003
- [6] WEI, X. and SHAW, M.: Optimum Allocation and Decomposition for High Quality Audio Coding, IEEE proceedings, 1997, PP. 315-318