Examination of lossy audio compression methods

Laboratory guide

Laboratory on Multimedia Systems & Services I.

Péter Rucz
rucz@hit.bme.hu

2018.

1 Introduction

Lossy audio compression methods became ubiquitous together with their portable playback devices. This is due to the efficiency of the compression methods and the good quality of the encoded audio content. The effectiveness of such compression techniques is illustrated by the following numbers. An MP3 bitstream of average quality has a data rate of $128\,\mathrm{kbit/s}$. At the standard sampling rate of $48\,\mathrm{kHz}$ this means 1.33 average bits per sample. Compared to the 16-bit samples of an audio CD, the data is twelve times compressed while the audio quality is about the same. Such compression rates are achievable only by using lossy psychoacoustics-based methods. The basis of lossy psychoacoustical compression methods is the omission of information from the audio signal so that it does not result in perceived difference. That is, the details that are omitted would be inaudible anyway. In order to determine which components are unperceivable to the human auditory system the psychoacoustical analysis of the signal is needed.

A general model of a lossy audio compression procedure is depicted in Figure 1. The audio input is processed simultaneously both by the coder analyzer unit and the psychoacoustical unit. In this step, the psychoacoustical unit can already provide information for the analyzer module, e.g. time window types to be used in the analysis and encoding processes. In our current exercises, the psychoacoustical analyzer unit is the most important one, and is examined in detail in Section 3. For the time being it is sufficient for our discussion that the main task of the psychoacoustical module is to tell the (desired) number of bits to be allocated for each audio sample in the frequency domain. To achieve a low number of allocated bits—and hence high compression rates—frequency and temporal masking phenomena are exploited. The samples are transformed into the frequency domain by the coder unit, then, re-quantization is performed based on the calculated number of bits. The requantized samples are written into

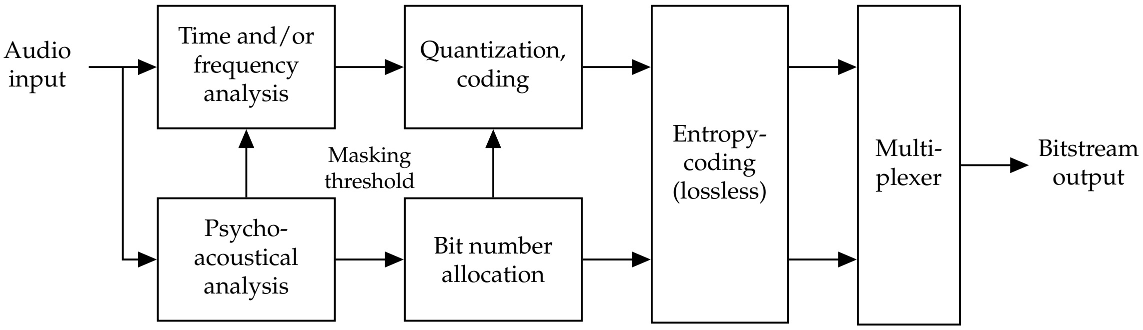


Figure 1: A general block scheme of lossy psychoacoustic coding

the final output bit-stream after lossless entropy coding (Huffman coding, for example). The output stream must also contain additional non-audio information required for decoding or synchronizing the stream. A common requirement for the encoder is to keep the maximal data rate of the output stream constant, such as 192 kbit/s. To achieve a steady data rate better encoder algorithms determine the number of allocated bits in an iterative manner (analysis and resynthesis iteration). This way, maximal audio quality can be achieved with constrained data rates.

In the present laboratory exercises a psychoacoustical model is examined that can be used for the lossy encoding of audio signals. The model exploits frequency domain simultaneous masking phenomena, see in Section 2. In our application the model is examined in a simplified requantization process. In order to make the encoder model and the compression method clear and concise, nonlinear quantization techniques and entropy coding of the quantized samples are not addressed here.

2 Some basic concepts of psychoacoustics

2.1 Tonal and atonal components

Sounds whose perceived pitch can be identified are referred to as *tonal* sounds; and in case of sounds composed of multiple components, such components are called tonal components. In contrast, sounds that can not be described by a perceived pitch are referred to as *atonal*. For example, the sound generated by a wind or bowed musical instrument, or that of a xylophone bar or bell hit by a mallet, or the sound of a plucked string are tonal sounds. Atonal sounds are for example, broadband noises, the sound of a cymbal or a muted and heavily distorted sound of plucked string of an electric guitar. In complex sounds, such as the noise generated by aircrafts, or in a piece of music in which different instruments are played, tonal and atonal components are both present at the same time. Distinguishing between tonal and atonal components is of key importance in psychoacoustical analysis, because the two have significantly different masking properties.

2.2 The Bark scale and critical bandwidths

The *Bark scale* is a special frequency scale, on which equal distances correspond to equal differences in the perceived pitch. The scale is named after the German physicist Heinrich Barkhausen. The Bark scale is conform with the *critical bandwidths* defined by Harvey Fletcher. Fletcher examined the disturbing effect of filtered white noises having different bandwidths on the detection of a pure sinusoidal tone whose frequency is at the middle of the band. He found that there exists a frequency dependent so called critical bandwidth, above which increasing the bandwidth of the noise signal does not deteriorate the detection of the pure tone any further. The critical bandwidth on the Bark scale is 1 Bark at all frequencies. The Bark scale is shown in Figure 2. It is observed that at low frequencies the scale is nearly linear, while above 500 Hz it has a logarithmic slope.

In audio compression the Bark scale is used in the calculation of masking curves. On the one hand, the strength of the masking effect that the *masker* exerts on the *maskee* is strongly dependent on the frequency distance of the components. This dependence is best expressed over the Bark scale. On the other hand, it is also useful to store the allocated bit numbers in the standardized critical bands. (Limits of the standardized critical bands are displayed as the vertical lines of Figure 2.)

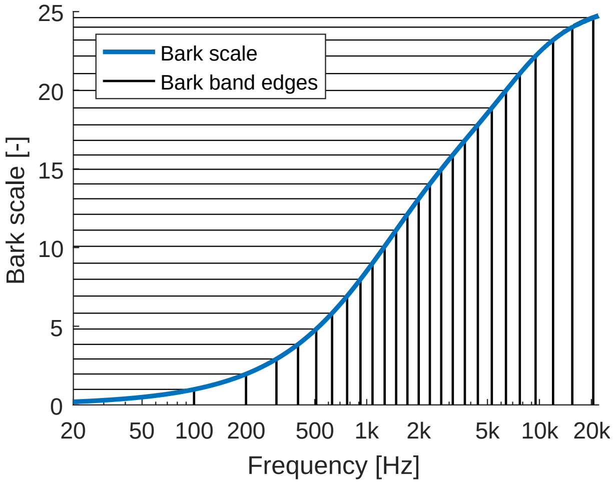


Figure 2: The Bark scale in the audible frequency range

2.3 Masking in the frequency domain

It is a known phenomenon that a perceived sound "disturbs" the perception of other simultaneously perceived audio signals. If the frequencies of the perceived sounds are close, one component can completely hide, i.e. *mask* the other. This phenomenon is referred to as frequency domain or simultaneous masking, the disturbing and the hidden components are called as the *masker* and the *maskee*, respectively. The effect is caused by the physical behavior of the *basilar membrane* located in the human inner ear. Excitations with different frequencies generate vibrations at different locations of the membrane whose stiffness has a spatial dependence. However, a pure tone consisting of a single frequency does not excite a single point, yet a somewhat spread section of the membrane. Thus, a narrowband excitation generates a wider band response. This effect is called as the *spreading* of the frequency band and is quantified by the so called *spreading function*.

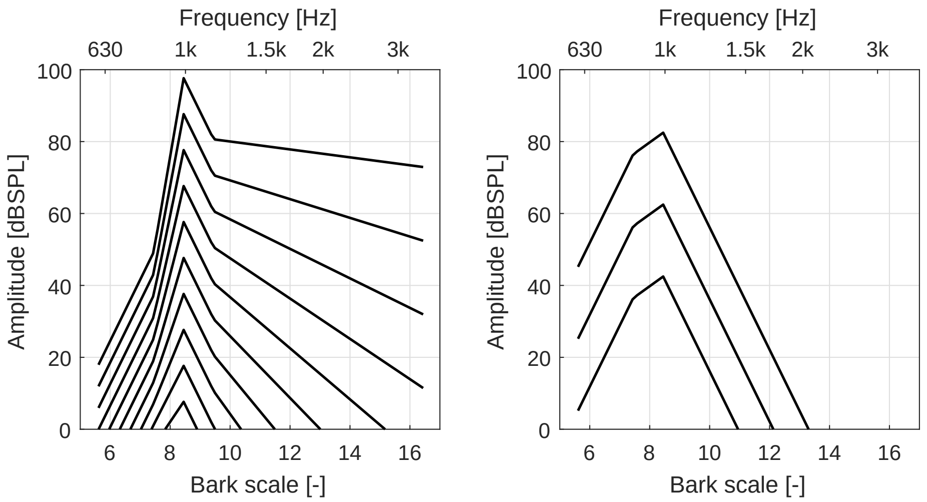
The strength of simultaneous masking is dependent on the nature of the masker component (tonal or atonal), its amplitude and the frequency distance of the masker and the maskee. The masking curves of a harmonic and a noise signal is displayed in Figure 3. Observe that the shapes of the curves is strongly dependent on the amplitude of the signal in case of the tonal sound, while it is independent of the amplitude for the atonal one. It is also worth noting that the curves are strongly asymmetric. Tonal components with high amplitudes mask a signficiantly larger area, especially at frequencies higher than that of the masker signal. This is indicated by the decreasing slope of the masking curve when the amplitude is increased. The masking curves are well approximated by piecewise linear functions over the bark scale.

Simultaneous masking is the most important psychoacoustical effect in lossy audio encoding. By detecting and omitting masked components a great coding gain can be achieved.

2.4 Time domain masking

An audio signal exerts masking effect not only simultaneously, but also after its decay. This phenomenon is referred to as *forward masking*. The duration of forward masking is strongly dependent on the frequency of the masker signal, the relation is approximately inversely proportional. The duration of the effect for low frequencies can be as high as $120\,\mathrm{ms}$ and its strength decays exponentially with time. As this duration can exceed the length of the analysis window used in psychoacoustics-based encoders, the effect is exploited in most audio compression methods.

The interesting effect called *backward masking* is also worth mentioning. In this case, a strong sound at a later time instance can mask a weaker sound at a previous time instance. This is



(a) Tonal

(b) Atonal

Figure 3: Simultaneous masking curves of tonal (a) and atonal (b) signals having different amplitudes in the frequency domain. The tonal signal is harmonic signal with 1 kHz frequency, while the atonal one is a narrowband noise whose band center is the same.

explained by the fact that high amplitude stimulation propagates faster on the auditory nerve than low amplitude stimulation. The time duration of backward masking is in between 2 ms and 5 ms depending on the amplitude of the masker signal. As this duration is shorter than the length of the time windows used in most audio encoding methods, the effect is not exploited in most of such encoders.

3 Psychoacoustical analysis

The functionality of a psychacoustical analysis module that can be applied for lossy audio compression is reviewed in this section. The model discussed here is given in the MPEG standard (ISO/IEC 11172-3 MPEG-1, layer 1) as a recommendation and is referred to as "psychoacoustical model I." More details on the model are found in the book [1], for example. Naturally, there exist a number of other psychoacoustical analyzer modules; however, this model is adequate for our examination thanks to its simplicity.

The most important quantity to be determined by the analyzer module is the frequency domain masking curve depicted in Figure 4, which represents the overall masking effect of all the components present in the audio signal at the same time. This frequency dependent function gives the level that is masked by the masker components, thus, levels below this threshold are not perceived. By exploiting this, the number of bits used for the coding can be reduced. It is sufficient to choose the number of bits in each frequency band such that the resulting quantization noise[1] is just below the masking curve of the coded components, i.e. the quantization noise remains under the masking threshold curve.

Figure 4 shows the masking curve of a harmonic signal with 2 kHz frequency together with the intermediate results of each step of the analysis. The steps are as follows.

1. Calculation of the power spectrum
   The first step of the analysis is to calculate the *spectral power* of one section (also called as slice or window) of the signal. In order to reduce spectral leakage the signal is multiplied

[1] The expression "quantization distortion" seems more adequate as the error signal due to quantization depends on the input signal.

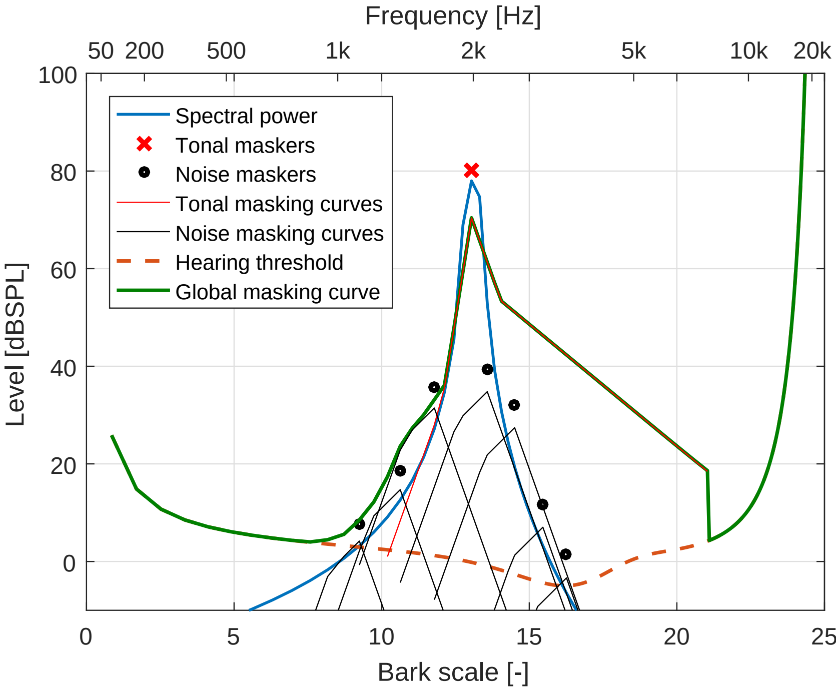


Figure 4: Masking curve of a narrowband signal

by a predefined window function, such as the Hann window. The strength of simultaneous masking effects depends on the absolute level of the signal, therefore a nominal playback level must be assigned to the non-dimensional recorded waveform. During encoding, the playback level is not known, hence the nominal level must be a *worst case* level. This level is defined such that in the band with 4 kHz center frequency, in which the sensitivity of the human ear is the highest, a harmonic signal of 1 bit amplitude corresponds to a signal of 0 dBSPL level if the sample is quantized to 16 bits. Thus, a full-scale signal has a nominal level of 90 dBSPL.

2. Finding masker components

   (a) From the power spectrum *tonal masker* components are sought first in the signal. Tonal components are assumed to be located at the maxima of the power spectrum, with a small modification that during searching for local maxima, not only values in the neighboring frequency bins are compared, but a wider neighborhood of each bin is considered. In Figure 4 red crosses denote the tonal components. Their amplitude is determined as the sum of the power in the bin of the maximal value and its two neighbor bins.

   (b) In each critcal band *noise maskers* are also determined. Noise maskers are taken at the band center of each critical band, their amplitude is derived from the total spectral power in the frequency band with omitting the spectral power associated with tonal maskers if there were such in the respective band.

   (c) Then, the masking components are sorted and filtered by a sliding window of 1 Bark and only the strongest masker component is kept in each window. The tonal or atonal nature of the kept maskers are also recorded. This way, we arrive at one masker component in each critical band.

3. Calculation of the masking curve

   (a) For each of the masker components an *individual masking curve* is computed by using the simultaneous masking curves shown previously in Figure 3. The individual

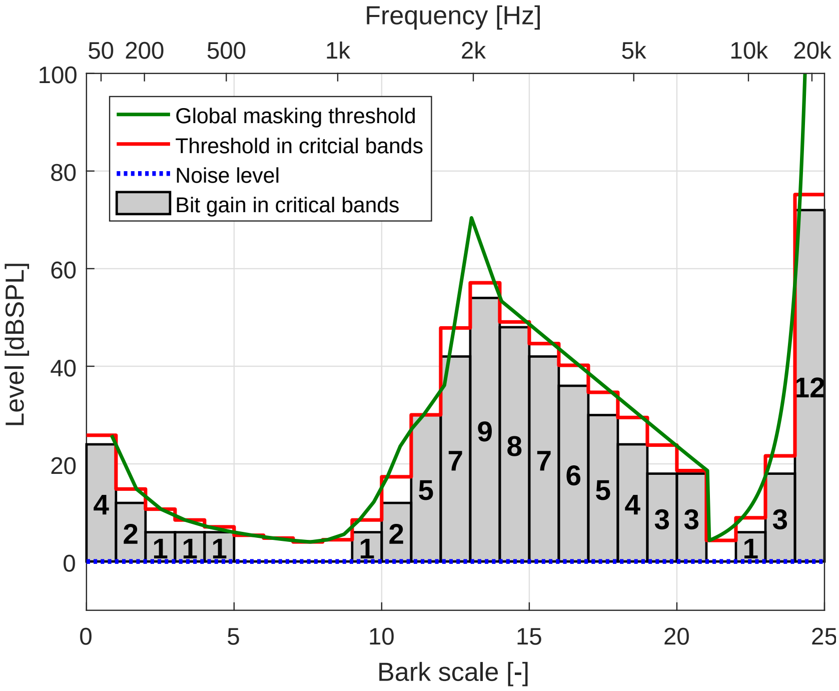


Figure 5: Bit allocation based on noise to mask ratio

masking curves are shown in Figure 4 by the narrow red (tonal) and black (atonal) lines.

(b) By taking the individual masking curves and the hearing threshold (dashed brown line in Figure 4) into account the *global masking threshold* curve is obtained in the full frequency range of interest. The global masking threshold is shown by the thick green line in Figure 4.

4. Bit allocation
   Based on the global masking threshold curve, bit numbers can be assigned to each critical frequency band, which is illustrated in Figure 5. Reducing the number of bits in a given frequency band increases the spectral power of the quantization noise in the given band. The number of bits can be reduced as far as the enhanced quantization noise remains under the global masking threshold curve. This way, in each frequency band, the difference of the minimal value of the masking threshold curve (*threshold in critcal bands*, red lines of Figure 5) and the noise level of the original signal (dashed blue line of Figure 5) determines the extent to which the quantization noise can be enhanced in the given band. This difference is called as *noise to mask ratio* and is abbreviated as NMR. As oberved in Figure 5, in case of $0\,\mathrm{dBSPL}$ noise level, the noise to mask ratio is the minimal masking level in each band. The reduction of bit numbers can easily be determined, as reducing the accuracy of the representation by one bit leads to a $20\log_{10}2 \approx 6\,\mathrm{dB}$ increase of the quantization noise.

The spectrum of an audio signal (a piece of music or speech, for instance) is naturally time dependent. Therefore, the frequency analysis is not performed over the whole length of the recording, but over smaller, consecutive *slice*s (also called *frame*s or *window*s) and the bit number allocation is also computed slice by slice. Consecutive slices are not independent, as they affect each other by time domain masking, for example. However, these effects are not taken into account in our simplified model.

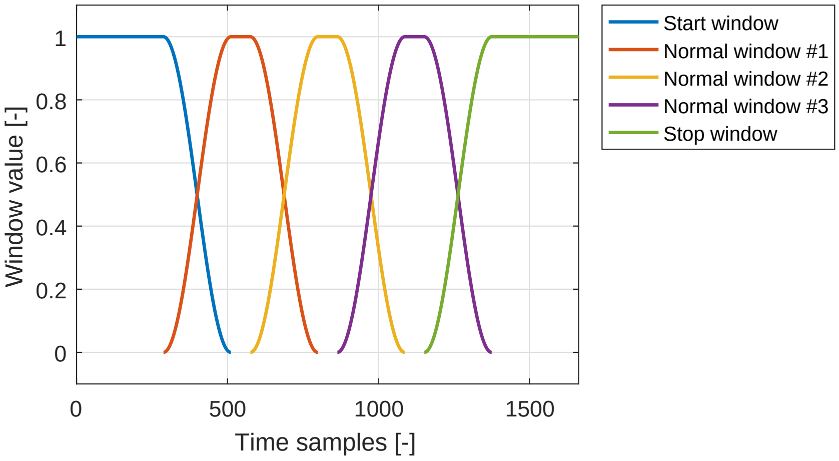


Figure 6: Overlapping time windows

4 Transformation and re-quantization

In the previous section it was seen that the phenomenon of frequency domain (or simultaneous) masking can be exploited in order to reduce the number of bits needed to represent a sound sample, and hence compression can be achieved. However, the analysis gives bit numbers in the frequency domain, while the signal is a sequence of time domain samples. Thus, to make the compression feasible, the samples are converted into the frequency domain, and the frequency domain samples are re-quantized and stored in the file with a reduced number of bits. Restoring the samples in the time domain is performed during decoding.

There are different methods that can be applied for the transformation, for example the MP3 standard uses a modified version of the DCT (*discrete cosine transform*). Fast Fourier transform (FFT) or wavelet transform are also suitable methods. The DCT has an advantage over the FFT that it converts real valued time domain samples into real valued frequency domain samples, thus, this is the preferred transformation method in our exercises.

4.1 The necessity of windowing

By reducing the number of bits in the frequency domain, the added noise is spread over the whole time window in the time domain. As the added noise varies from slice to slice, the slices that are re-quantized and back-transformed into the time domain can contain sudden jumps at the slice boundaries, which results in periodic "pops" during playback. To mitigate this effect it is necessary to take overlapping slices of the original sample and re-quantize the overlapping slices (this naturally means additional information to store), and finally to apply the appropriate weighting windows to get a smooth transition in the time domain.

Overlapping time windows are exemplified in Figure 6. In the figure all time windows have the same length and their overlapping parts have a cosine characteristic. It is also observed that the sum of all the windows gives the value of 1 at each time instance. Normal windows have a finite slope at both ends, while start and stop windows are rounded on one end only; thus, the latter can be applied at the beginning or the end of the recording.

4.2 Variable window lengths

Figure 7 demonstrates an interesting effect, where the quantization noise that is distributed uniformly in the time window appears noticeably before a quick transient. This is caused by the fact that the analysis method allocates the bit numbers based on the spectral content of the whole slice. Thus, if significant differences in amplitude are present in one slice, the noise

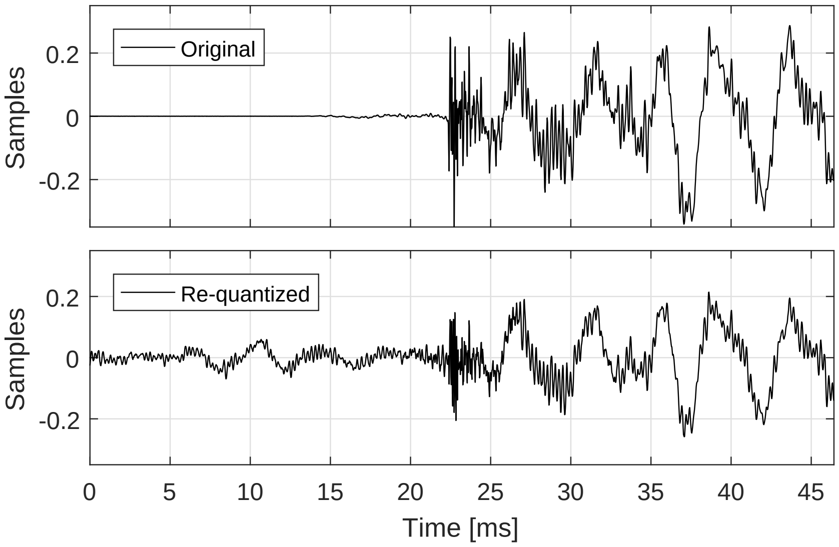


Figure 7: Spreading of quantization noise in the time window (*pre-echo*)

can emerge from the background where the original sample was "quiet." As quantization distortion is strongly correlated with the input signal, a distorted "echo" of the quick transient appears before the transient itself. Therefore the phenomenon is also called *pre-echo*. In case of a quick decay, the problem is insignificant, as in this case, time domain masking helps rendering the "echo" inaudible.

To avoid this unwanted effect, the psychoacoustical analysis module aims at detecting quick transients in the audio signal. When such a transient is found, it is useful to locally reduce the length of the time windows used in the encoding process. For example, in case of quick transients MP3 uses three shorter windows of 192 sample length instead of the original slice length of 576 samples.

4.3 Estimation of the coding gain

The resulting coding gain can not be determined merely from the bit numbers used for the re-quantization. This is due to several reasons. First, the number of bits used for the re-quantization varies from slice to slice, hence, beside storing the actual value of the samples, the number of bits used for the representation also has to be stored somehow, which means additional information. Therefore, it is not worth to assign different number of bits for each frequency domain sample. Instead, the number of bits is kept constant across a band (e.g. a critical band), which enables storing the number of bits band by band. Second, after the re-quantization lossless entropy coding is utilized (MP3 uses Huffman coding, for example), which reduces the number of bits to store by a great extent. Moreover, in a given file format, the encoded file must also be able to hold other type of information beside the audio samples that require data storage. Such information can be special frames for synchronization, descriptive data of the file format, meta-data of the recording, etc.

Nevertheless, it is useful to estimate the coding gain resulting from the re-quantization. The easiest way of attaining the coding gain is to sum the bit numbers of the frequency domain samples and divide it by the product of the original number of bits per sample and the number of samples. A better prediction is obtained if we take into accout that a great number of frequency domain samples become zero after the re-quantization. It is usually worth coding the zeros using a single bit even at the cost of increasing the length of other code words by one.

4.4 Evaluation

In psychoacoustics-based lossy compression an interesting difficulty is the evaluation of the encoded audio material. As the model exploits psychoacoustical phenomena, the perceived quality can not be estimated by means of the standard objective measures, such as the mean square error. Therefore, the quality is often assessed by means of subjective listening tests. Objective evaluation is possible by utilizing a standardized psychoacoustical analysis model, called PEAQ (*perceptual evaluation of audio quality*) [2].

5 Questions and exercises for preparation

1. What does the expression "psychoacoustics-based lossy audio compression" mean?
2. Draw the block scheme of a general lossy audio compression method!
3. Give the definitions of the Bark scale and the critical bandwidth!
4. What does frequency domain or simultaneous masking mean?
5. What is time domain masking?
6. What is the difference between a tonal and an atonal signal?
7. Why it is important to distinguish between tonal and atonal components in lossy audio compression?
8. Which parts constitute the global masking curve of an audio signal?
9. What is the definition of NMR (*Noise to Mask Ratio*) and how is it related to bit allocation?
10. Lossy audio compression methods store the signal in the frequency domain. Why is this necessary?
11. Why windowing is needed in the re-quantization process? What is the result of omitting the window functions?
12. What does "pre-echo" mean?
13. By which means can the quality of a lossy encoded audio material be assessed?
14. Write a Matlab code that creates a sine signal having a length $T = 2\,\mathrm{s}$, amplitude $A = 1$, and frequency $f = 2\,000\,\mathrm{Hz}$ with a sampling frequency of $f_\mathrm{s} = 48\,\mathrm{kHz}$!

6 Laboratory exercises

Prepared files for the exercises can be downloaded from the link below:
`http://last.hit.bme.hu/download/fospeclabor1/audio/audio_lab.zip`

All tasks need to be solved in `Matlab` environment, thus, a working knowledge of `Matlab` is necessary for solving the following problems. The files prepared for the laboratory exercises contain the `tools` folder which is the collection of some useful functions and the `exercises` folder that contains the files prepared for the problems themselves. Before proceeding with the problems, it is worth examining and interpreting these codes first. Comments and descriptions for the prepared functions can be shown by using the `help` or `doc` commands. The functions implement the psychoacoustical analysis method discussed above and contain some tools for the frequency domain transformation and re-quantization of the samples.

1. Setting up the environment
   Run the `ex_00_setup.m` Matlab script, which sets up the path variables needed for accessing the used functions. (Once it was executed successfully, this step does not need to be repeated.)

2. Examination of the effect of quantization

   (a) Write a Matlab script that creates a sine signal having a length $T = 2\,\mathrm{s}$, amplitude $A = 1/2$, and frequency $f = 1\,\mathrm{kHz}$ with a sampling frequency of $f_\mathrm{s} = 44\,100\,\mathrm{Hz}$. (It is worth storing the samples as a column vector, as the helper functions expect the audio signals as column vectors.) Listen to the signal using the Matlab function `sound`. Display the spectrum of the signal, using the `spec_aver` for the calculation.

   (b) Simulate quantization on 16 bits by scaling the signal. (Quantization on b bits means multiplication by 2^{b-1} and then applying rounding or truncation.) You can use the function `requantize`. Compare the spectrum of the quantized signal with that of the original signal. What do you observe?

   (c) Re-quantize the signal using the function `requantize` at 12, 8, 6, and 5 bit precision. Listen to the resampled signals using the function `sound`. (Note that the function `sound` expects the samples to be normalized between -1 and $+1$, thus, a rescaling needs to be applied before playback.) Examine the spectrum of the signals at low number of bits. What effects can you observe? Can the quantization error be regarded as a noise independent of the quantized signal?

3. Frequency domain masking

   (a) Create a copy of the script you made for the previous problem and use 16 bit precision. Take a slice of the signal (it is advised to use 512, $1\,024$, or $2\,048$ samples) and display the masking curve of the slice. For this task use the prepared function `analyze_slice`. In the resulting figure observe the tonal and atonal (noise) components and their corresponding masking curves.

   (b) Display the noise to mask ratio and the number of bits that can be dropped in each frequency band. (You can do this easily with calling the function `analyze_slice` using the appropriate parameters.)

   (c) Using the function `requantize_windows` re-quantize the whole audio sample (all the $2\,\mathrm{s}$). Compare the spectrum of the original and the re-quantized samples. Does the results match your expectations?

   (d) Mix (add) another harmonic signal to the existing one that is close in frequency (like $1.1\,\mathrm{kHz}$ to $1\,\mathrm{kHz}$) with smaller amplitude. (When setting the amplitudes ensure that the signal is not clipped, i.e. the maximum absolute value does not exceed 1 at any time.) Check if the added signal is below the masking threshold with the chosen amplitudes. Try to set the amplitude of the added signal such that it is close to the masking threshold. Listen to the original and re-quantized samples and check if the difference is audible. It is also useful to examine the spectra of the signals. Does the calculated masking threshold match the limit of audibility?

4. Transformation coding of audio recordings

   (a) Choose a sample recording of your preference from the `samples` folder. Load the recording into Matlab using the function `audioread`. (If the recording has multiple channels, use only one channel of them for the further steps.)

   (b) Re-quantize the recording using the function `requantize_windows`. Create a plot of the original and the re-quantized signals, and also their difference in a separate diagram. Listen to the original and re-quantized recordings. Also listen to the difference signal! What do you experience when you listen to the difference signal?

(c) Create a diagram of the allocated number of bits as a function of both frequency and time. (Use the function `pcolor`.) Estimate the coding gain!

(d) Modify the re-quantization function (`requantize_windows`) such that it drops 1 or 2 more bits in each frequency band than the number determined by the psychacoustical analysis. Listen to the result! Is the effect audible?

(e) Try the transformation coding using different window lengths and different overlap settings. What do you experience when the overlapping is zero?

(f) Perform some of the previous tests on different sound samples. Carry out a blind-test with one of your classmates: can you tell by listening which is the original and the re-quantized recording?

5. Time domain effects

   (a) Load the file `castanets.wav` from the sample recordings! (Use `audioread`.) Find a part in the recording which contains a quick transient, such as the guitar suond at the beginning of the recording. Cut a slice of the signal that contains this transient and use the functions `analyze_slice` and `requantize_slice` to re-quantize this segment of the recording. Plot the original and the re-quantized slices. What do you observe?

   (b) Examine the same effect using longer and shorter windows too.

References

[1] A. Spanias, T. Painter, and V. Atti. *Audio signal processing and coding*. John Wiley & Sons, Inc., New York, 2007.

[2] T. Thiede, W. C. Treurniet, R. Bitto, C. Schmidmer, T. Sporer, J. G. Beerends, C. Colomes, M. Keyhl, G. Stoll, K. Brandenburg, and B. Feiten. PEAQ—The ITU standard for objective measurement of perceived audio quality. *Journal of the Audio Engineering Society*, 48(1–2):3–29, 2000.