Audio Engineering Society
Convention e-Brief 127
Presented at the 135th Convention
2013 October 17–20 New York, USA

PsychoMasker: An iOS Application for the Visualization of PsychoAcoustic Principles

Andrew Ayers[1], *Robert Rehrig*[2], *Christopher L. Bennett*[3], *Colby Leider*[4]

University of Miami, Coral Gables, FL, 33156, USA
[1] *a.ayers@umiami.edu*, [2] *r.rehrig@umiami.edu*, [3] *bennett@miami.edu*, [4] *cleider@miami.edu*

ABSTRACT

The concept of masking in psychoacoustics has invaded the daily lives of almost every audio listener since the initial release of the MPEG-1 standard. With the ubiquity of the MP3, the consumption of perceptually coded audio is impossible to avoid. While many people understand the concept of perceptual coding, it can be difficult to visualize what is actually happening to the information in the audio files. PsychoMasker is an App that provides real-time visualization of the psychoacoustic principles used in MPEG encoding to anyone with an iPad. The PsychoMasker App shows the user how the encoding process affects any song in the user's iTunes library step-by-step.

1. INTRODUCTION

The PsychoMasker app was created with the aim of visualizing the concept of spectral masking and demystifying the idea of perceptual encoding. The app allows the user to view the nine-part psychoacoustic model laid out in the ISO/IEC 11172-3 specification step-by-step, visualizing each step in real time for any audio file in the user's music library, though it works best with uncompressed audio. [1] The MPEG standard is most closely associated with the MPEG-1 layer 3 codec, or MP3, but this app more closely follows the MPEG-1 layer 2 scheme. The layers increase in complexity as they ascend in number, so layer 2 provides an encoding scheme of intermediate intricacy aimed at bit rates around 128 kbits/s per channel. It is not as sophisticated as the layer 3 standard, but it does employ many of the same psychoacoustic principles such as spectral masking and the absolute hearing threshold.

2. PSYCHOACOUSTIC MODEL

The MPEG model uses the psychoacoustic principles of spectral masking and the hearing threshold of the human ear to reduce file size by eliminating indistinguishable and unhearable audio. The concept of masking provides that when the ear is stimulated by a loud tone, it's ability to hear quieter tones that are close in pitch is diminished. Using the bark scale as a measure of the perceptual distance between two tones, a masking curve can be drawn around a tone under which other notes cannot be heard. Also, by employing an equal-loudness curve as a model of the frequency response of the human ear, audio at different frequencies can be treated with a different threshold. [2]

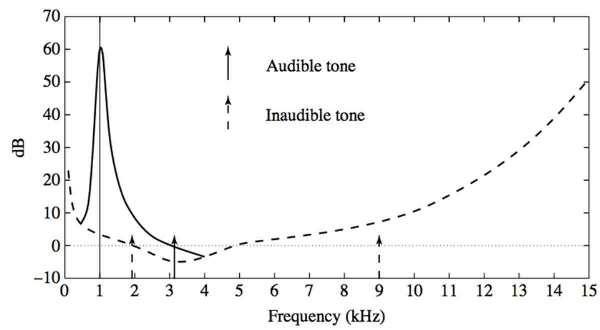

Fig. 1: Effect on threshold of human hearing for a 1kHz masking tone.

The psychoacoustic model given in the MPEG specification is comprised of nine steps:

Step 1: Calculation of the FFT for time to frequency conversion.
Step 2: Determination of the sound pressure level in each subband.
Step 3: Determination of the threshold in quiet (absolute threshold).
Step 4: Finding of the tonal (more sinusoid-like) and non-tonal (more noise-like) components of the audio signal.
Step 5: Decimation of the maskers, to obtain only the relevant maskers.
Step 6: Calculation of the individual masking thresholds.
Step 7: Determination of the global masking threshold.
Step 8: Determination of the minimum masking threshold in each subband.
Step 9: Calculation of the signal-to-mask ratio in each subband

The final result of the pyschoacoustic model section of the encoder is a calculation of the signal-to-mask ratio, which gives a measure of the number of bits needed to encode each band. This app does not deal with bit-allocation, or any of the steps past the psychoacoustic model.

3. IMPLEMENTATION

The app follows the nine-step specification given in section II, dividing the steps into four classes of operations:

- Audio Playback and Transform
- Tonal and Non-Tonal Component Calculation
- Decimation of the Tonal and Non-Tonal Components
- Calculation of the Masking Curves and Signal-to-Noise Ratio

3.1. Audio Playback and Transform

The audio playback and FFT are handled by Apple's Audio Units and vDSP frameworks respectively. The app takes a 1024-point hamming-windowed FFT of the audio's left channel and converts the resultant vector to decibels (dB). After discarding the redundant frequency information from Nyquist to the sampling rate, the 512-point magnitude vector is then plotted for the user, and passed to the next class for further operation.

Additionally, the next step is handled outside of the four main groupings of operations. The prescribed threshold in quiet is given as a table in the specification, but for ease of programming, a continuous function that approximates the equal-loudness curve was used in this app and is given in equation (1). [3]

$$T_q(f) = 3.64\left(\frac{f}{1000}\right)^{-0.8} - 6.5e^{-0.6(f/1000-3.3)^2} + 10^{-3}\left(\frac{f}{1000}\right)^{4} \quad (1)$$

3.2. Tonal and Non-Tonal Component Calculation

The tonal and non-tonal components are calculated by an algorithm that first finds local maxima in the magnitude vector, and then evaluates each local maxima to see if it is a tonal component. A local maxima is considered a tonal component if the bins to either side are at least 7dB less in magnitude. The number of bins surrounding the local maxima that are considered depends on the frequency of the maxima; the higher in frequency, the more bins to either side that must be considered. This goes back to the idea of perceptual bands, and that the human ear has better spectral resolution at lower frequencies. The numbers of bins and the ranges are all given in the specification. The non-tonal components represent the bark band averages. A non-tonal component has a calculated mean magnitude of all the bins within a bark band, and it is located at the center bin of the band. Tonal components are not included in the non-tonal calculations.

3.3. Decimation of the Tonal and Non-Tonal Components

After having found all of the possible maskers, tonal and non-tonal, the next step is to decimate what them into a set of the most relevant maskers. The first step is to compare the amplitude of all of the maskers to the absolute hearing threshold. Any masker which is below the threshold is thrown out. The second step is an application of a 0.5-bark sliding window to the maskers. If any two tonal maskers or any two non-tonal maskers are within 0.5-bark of each other, only the higher amplitude masker is kept.

3.4. Calculation of the Masking Curves and Signal-to-Noise Ratio

The last major section of the app handles the calculation of the masking curves. For each of the maskers that are left after decimation, a masking curve is calculated according to a piecewise function provided in the specification. [1] Originally the specification called for only a decimated set of bins to be considered, presumably to save computations. Because computational power is not at a premium in this application, the masking curve is calculated here for each masker against every other bin. By converting each bin from frequency to bark according to the function given in equation (2),

$$Bark = 13\arctan(0.00076f) + 3.5\arctan\left(\left(\frac{f}{7500}\right)^2\right) \tag{2}$$

a bark distance between the masker and the bin being considered can be calculated. The applicable section of the piecewise function for the masking curve is then applied to the bin, taking both the magnitude of the masker and its distance from the bin into consideration. The tonal and non-tonal masking curves then become the maximum contributors for each bin from the individually calculated masking curves for each masker. Likewise, the total masking curve is the maximum value at each bin of the tonal masking curve, non-tonal masking curve, and the absolute hearing threshold. Assuming the most conservative case, the lowest point on the total masking curve in each bark band is chosen as the overall masking level in that band. The very last step in the process is to calculate the difference between the level of each bin and the overall masking level, giving the signal-to-mask ratio. In the original specification, the signal-to-mask ratio was only calculated per sub-band, but again since the computational power in this application isn't restricted, additional resolution was prioritized at this step over increased speed.

4. CONCLUSION

The final version of this app is a complete implementation of the psychoacoustic model presented in the MPEG-1 specification and a product that makes the concept of frequency masking easy to understand and visualize. It is the hope of the authors of this software that this product could be a useful tool in teaching or explaining the concepts of psychoacoustics to future students.

5. ACKNOWLEDGMENTS

This project was based in part on an assignment completed by the authors as well as the rest of Dr. Chris Bennett's psychoacoustics class in the spring of 2013 at the University of Miami. This research was supported by the University of Miami's Music Engineering program.

6. REFERENCES

[1] ISO/IEC 11172-3:1993 Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s - Part 3: Audio.

[2] Pan, D. (1995). A tutorial on MPEG/Audio Compression. *Multimedia, IEEE*, vol. 2, no. 2, pp. 60-74.

[3] Terhardt, E. (1979). Calculating virtual pitch. *Hearing research*, vol. 1, no. 2, pp. 155-182.

[4] Traunmüller, H. (1990). Analytical expressions for the tonotopic sensory scale. *The Journal of the Acoustical Society of America*, vol. 88, pp 97.

[5] Li, Z. N., Drew, M. S. (2004). *Fundamentals of multimedia, ISBN: 0130618721* (Vol. 7458). Pearson Education, Inc., Upper Saddle River, NJ.

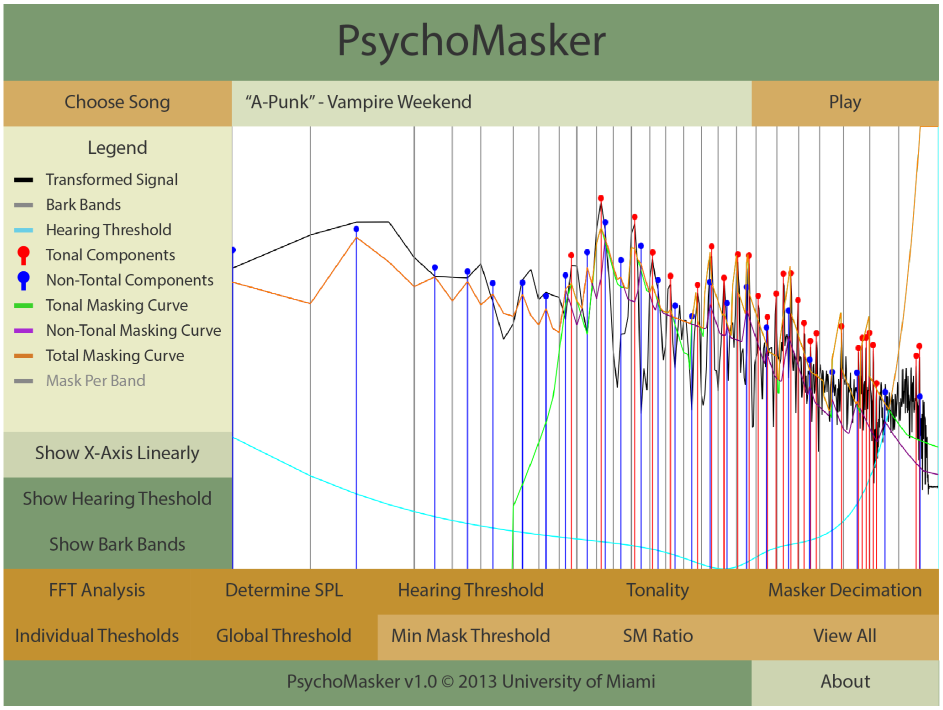

Fig. 2: Screenshot of PsychoMasker app.

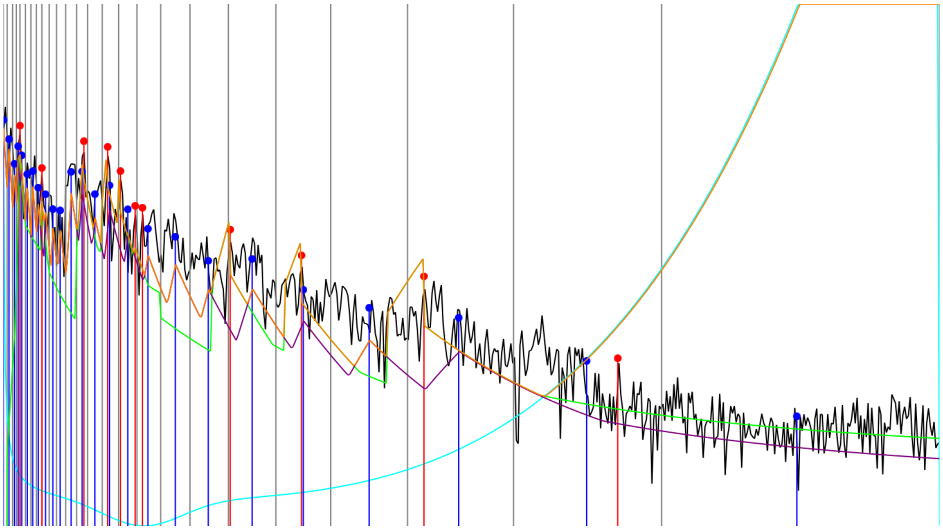

Fig. 3: Zoomed in view of PyschoMasker plot with linear spacing.