# Implementation of MPEG Audio Layer 1

Karol Wadolowski

## I. Introduction

THIS report first explains the basics of MPEG Audio Layer 1. Afterwards my implementation of the algorithm will be explained. This will be followed by the simulation results of the algorithms use on two pieces of music.

### *A: The MPEG Audio Layer 1 Algorithm*

MPEG-1 audio encoding and decoding can be visualized through the block diagram in Fig. 1. The audio input first gets passed into a filter bank and into a psycho-acoustic model. The filter bank takes the incoming signal and breaks it into 32 frequency sub-bands. The psycho-acoustic model takes the same input signal and determines the power and tones present in a portion of the signal. Using the power and tone locations of the signal a certain number of bits are allocated to encoding to each of the 32 sub-bands. The number of bits allocated is dependent on the selected data-rate for encoding. After allocating a certain number of bits to each sub-band the data is quantized accordingly. The quantized data is then formatted in a bit stream. This concludes encoding. Decoding starts with the bit stream of the encoded signal. The bit stream or frame is unpacked and recreates the quantized 32 sub-band signals. The sub-band signals are then put through a filter bank that synthesizes the audio signal. More depth on the analysis (encoding) and synthesis (decoding) procedures will be provided in Section B.

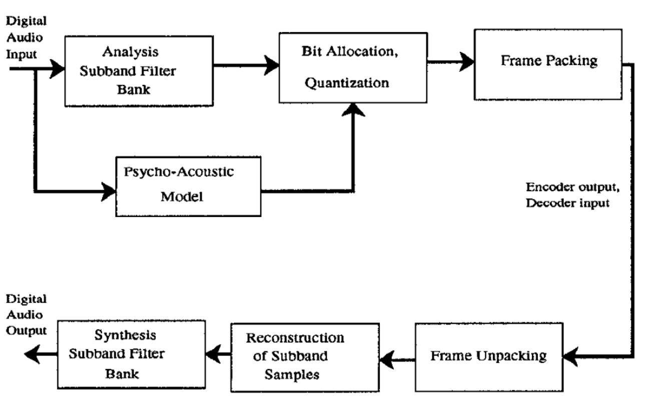


Figure 1. MPEG-1 Audio Encoding and Decoding Algorithm from Murphy et al [1].

### *B: MPEG-1 Audio Implementation*

The first step in my implementation was to break the signal into its 32 polyphase components. This was done using the process shown in Fig. 2. To start off the signal, $s$, that was to be encoded was zero padded so that it had a minimum size of 512 and so that its length was divisible by 32. The first 32 samples were pushed into a FIFO register, $X$, of size 512 such that $X(0) = s(31)$, $X(1) = s(30)$, ..., $X(31) = s(0)$. The indexing is like this because index 511 of the FIFO register represents the earliest samples and index 0 the most current sample. It is also done this was because $s(0)$ is the first part of the signal. When running the 32 most current samples are pushed into the FIFO register. The FIFO samples are then point-wise multiplied by the C coefficients defined in the standard. The resulting vector, Z, is then reshaped into a matrix of size $64 \times 8$. The rows of the matrix are summed to get the column vector Y as in Fig. 2. The Modified Discrete Cosine Transform (MDCT) is then applied to Y. The MDCT can be implemented as a matrix M with M defined as the $32 \times 64$ matrix in Eq. 1. To get the 32 sub-band samples just do a matrix multiply, MY. The next step is to apply the psycho-acoustic model.

$$(M)_{k,n} = cos\left(\frac{\pi(k-16)(2n+1)}{64}\right) \tag{1}$$

For k = 0, 1, ..., 63, n = 0, 1, ..., 31

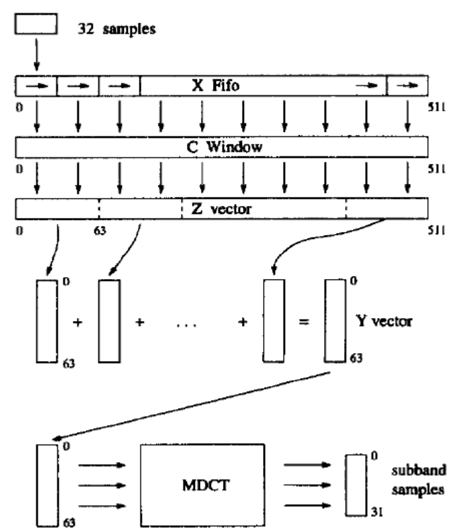


Figure 2. Creating the 32 polyphase components for 32 new input samples. Taken from Shlien [2]

The psycho-acoustic model that I implemented was of my own creation as I couldn't find adequate resources to implement a known one. The perceptual model starts by taking the 32 sub-band signals. My model takes 12 adjacent samples from sub-band j and uses them to compute a mask for those 12 samples of sub-band j. To compute the mask, calculate the power of each sample. The power of each sample is given by $20log_{10}|S_j(t)|$ where $S_j$ is sub-band j, t varies in a range of 12 adjacent values, representing the sample. Compare the power of all 12 samples against the maximum. If the value of the power is within a user defined threshold then the mask allows that sample through. If the value is not within the threshold the mask will zero out that sample. This is done for each of the sub-bands and for all blocks of 12 samples with no overlap between blocks. Also each channel can be given a different threshold as to give the user the option to emulate the Fletcher-Munson curves. After this is done the mask is applied to each of the 12 samples. Note that a threshold of infinity keeps all the values as they are were as a threshold of zero keeps only the sample corresponding to the max power. This concludes the psycho-acoustic model.

Encoding of the 32 sub-bands into a frame and the associated frame decoding was not implemented.

After being masked by the psycho-acoustic model, the 32 sub-band signals were synthesized into a single audio signal. The synthesising process is summarize in Fig. 3. The synthesis begins by shifting in 64 new samples into a FIFO register. The 64 new samples are calculated by performing the inverse MDCT or IMDCT on the 32 current samples, 1 sample coming from each of the 32 sub-bands. The IMDCT like the MDCT can be performed using a matrix multiply. The IMDCT matrix $M^{-1}$ is defined as the $32 \times 64$ given in Eq. 2. Note that the $M^{-}1$ is just an abuse of notation. This equation is very similar to Eq. 1 but not exactly. After calculating 64 new samples for the V FIFO in Fig. 3 the implementation follows the rest of the block diagram of Fig. 3. This concludes the reconstruction of the signal.

$$(M^{-1})_{n,k} = cos\left(\frac{\pi(k+16)(2n+1)}{64}\right) \tag{2}$$

For k = 0, 1, ..., 63, n = 0, 1, ..., 31

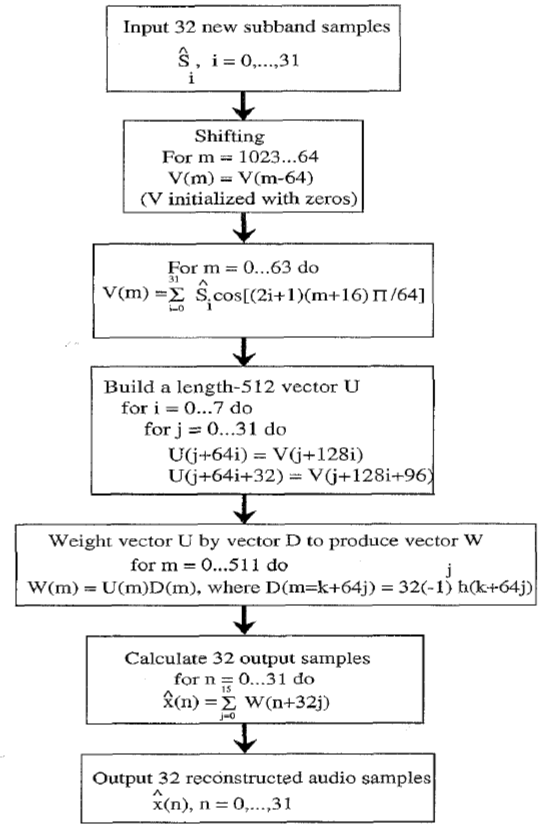


Figure 3. Synthesizing the 32 sub-bands into a single audio signal. Taken from [1].

### *C: Simulation Results*

To test the effects of my perceptual model I selected the thresholds first. For sub-bands 0-7 a threshold of 5 dB was chosen, for 8-15 a threshold of 3 dB, for 16-23 a threshold of 2 dB, and for 24-31 a threshold of 1 dB was chosen. There are larger thresholds for the earlier sub-bands as they correspond to lower frequencies. These thresholds will correspond to the compressed version. The loss less version will correspond to all thresholds being infinite. The spectrograms of the original signal, loss less, and compressed version were produced.

The first piece of music used to test the compression scheme was 'Paint It, Black' by The Rolling Stones. The loss less version had no audible differences. The compressed version was audibly different. The cymbals on the drums were clearly compressed and some portions of it were almost completely removed. The spectrograms can be viewed in Fig. 4. From the spectrograms, you can see that the original and loss less versions look the same. The lossy or compressed version clearly has more lower frequency power than the original. This can be seen in the smear near the bottom of the lossy spectrogram.

The second piece of music chosen was 'Music of the Spheres' from Portal 2's original sound track. The loss less version for this piece also had no audible differences from the original. The lossy version had a consistent low frequency buzz present through out. The spectrograms for all three versions can be seen in Fig. 5. The original and loss less look exactly the same again. The lossy version again has more low frequency power as again seen by smearing.

## II. Conclusion

From the two pieces of music tested it is clear that for the compressed (lossy) case the threshold for the lower sub-bands is too low. Since the thresholds are too low, many samples are zeroed out which causes the artifacts that can be heard and seen from the spectrograms. The higher frequency components

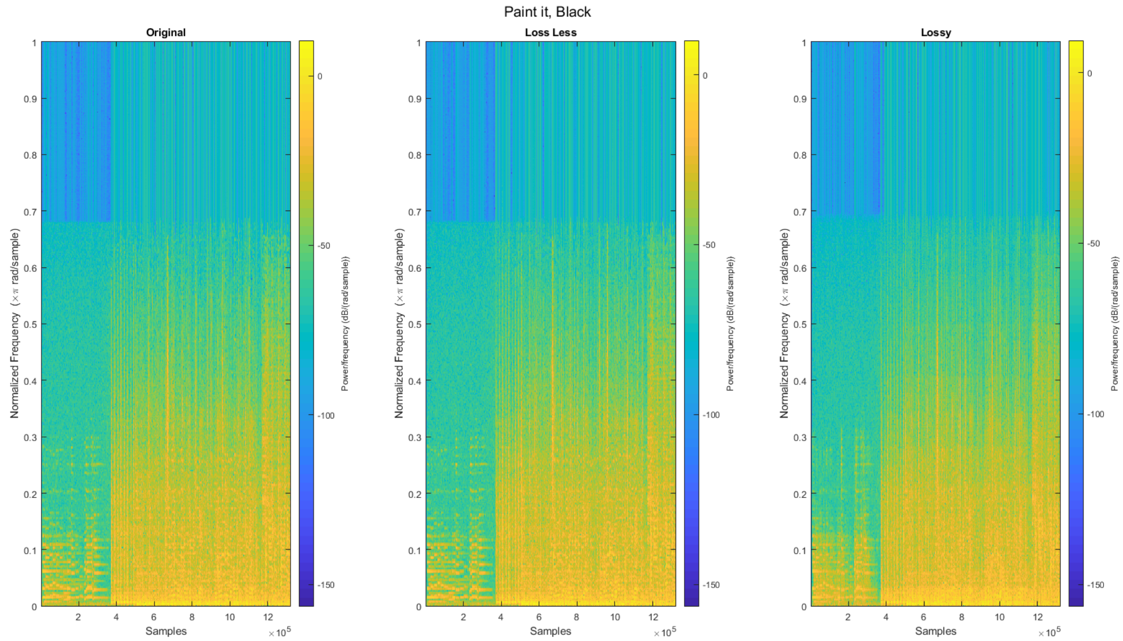


Figure 4. Spectrograms for various versions of 'Paint It, Black.'

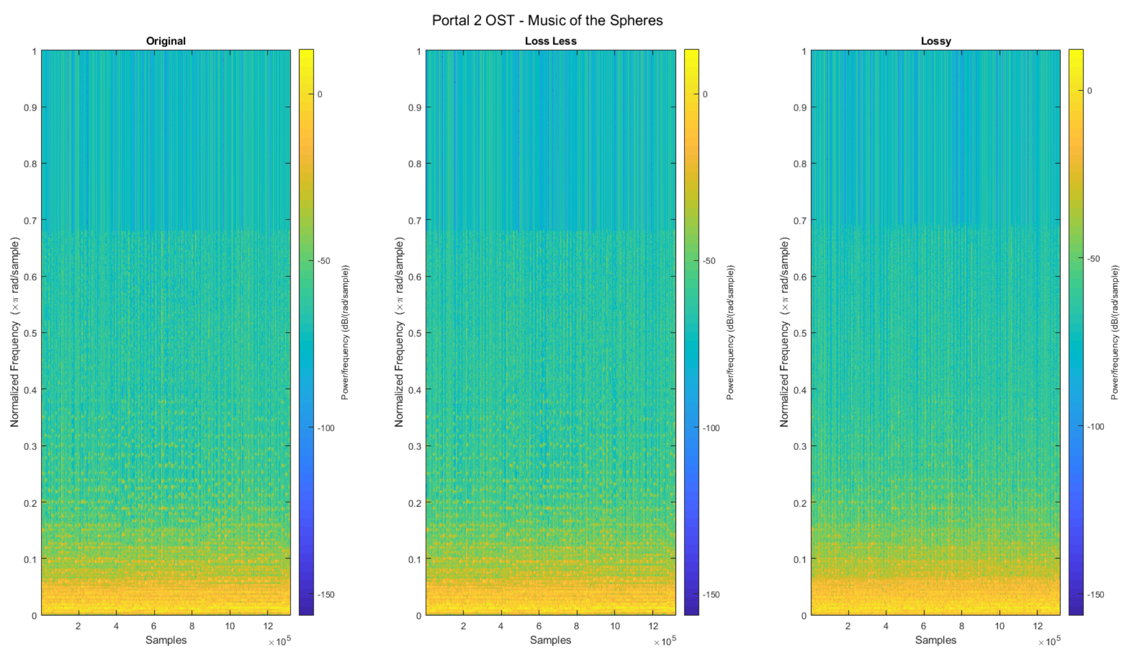


Figure 5. Spectrograms for various versions of 'Music of the Spheres.'

did not have this issue which suggests the thresholding used for the later sub-bands was adequate. This makes sense if you consider the Fletcher-Munson curves which show that humans can more easily detect frequencies in the 500 - 5 kHz range most easily. If framing and quantization was implemented, this would mean that lower frequency components should have more bits allocated to them and higher frequency components less since more of them can be zeroed out without audible artifacts.

## References


[1] Charles D. Murphy and K. Anandakumar *Real-Time MPEG-1 Audio Coding and Decoding on a DSP Chip*. IEEE Transactions on Consumer Electronics, Vol. 43, No. 1, February 1997

[2] Seymour Shlien *Guide to MPEG-1 Audio Standard*. IEEE Transactions on Broadcasting, Vol. 40, No. 4, December 1994