Identifying Perceived Loudness in Audio Signals

Bill S Lin, Wesley Fisher
University of Waterloo, Waterloo, CANADA

Abstract

In processing audio, it can be helpful to have algorithms that can extract volume and loudness information. One application of this would be finding speech in long intervals of silence. Digital audio files, which contain raw information, may not represent how a human can perceive the sound. For example, high and low frequencies of sound with equal intensities will be perceived as having different levels of loudness. This paper presents a series of processing operations to be performed on sound signals, which attempt to find the loudness of sounds as perceived by a human. Previously used methods of calculating perceived loudness includes looking at the frequency spectrum of sounds [1]. The first operation accounts for the human perception of loudness at different frequencies. The second creates an envelope of the sound signal, while preventing impulses from getting filtered out. These impulses are short bursts of sound such as a gunshot or a handclap. The third operation accounts for the human perception of impulse loudness. The output of this process is an envelope of the original sound signal, which represents how a human would perceive the audio volume. This removes the sinusoidal components in the sound signal and maintains the sound amplitude information, with some other minor adjustments.

Introduction

In large sound audio files, it can sometimes be desirable to have a computer quickly identify the louder sections of an audio clip for further processing. For example, algorithms that perform speech processing need to identify the periods of silence in an audio clip, which are of no interest to the algorithm. Similarly, algorithms processing audio recordings of birds chirping may want to identify where in the audio file the chirping actually occurs. A simple naive approach to solve this problem could be to set a threshold on the audio amplitude. This way, any signal with amplitude less than the threshold could be deemed as “silence” and any signal with amplitude greater than the threshold could be deemed as “sound”. The problem with this technique is that it does not consider how humans perceive sound. For example, the human ear is much less sensitive to very high frequency sound [2]. Similarly, short sound impulses are perceived as quieter by humans [2]. On top of this, because sound waves oscillate up and down (like a sine wave), it becomes more difficult to identify the actual peak amplitude of a given sound signal.

This paper presents a signal processing method that attempts to extract the perceived loudness information from an audio signal. This is done by taking the envelope of the audio signal. On top of this, adjustments based on audio frequency and short impulses are made. The resulting signal makes it easy to programmatically identify the loudness of an audio file at any point in time.

Methodology

To extract the perceived loudness from an audio signal, three steps are performed. Firstly, the frequency components of the signal are modified to reflect what is being perceived by humans. For example, frequencies to which humans are less sensitive are attenuated, and frequencies to which humans are more sensitive are amplified, by amounts described by equal-loudness curves and transfer functions used by Rimell et al. [3]. Secondly, the envelope of the audio signal is found to eliminate the actual sound vibrations and preserve the amplitude information of the vibrations. Finally, short sound impulses in the audio signal are identified. vThese impulses are then attenuated based on their duration, according to a human's ability to perceive these impulses as described by Everest [2]. The following subsections discuss the detailed steps used for each operation.

Frequency Adjustment

Given equal sound pressures, different frequencies are perceived with different levels of loudness by humans. In particular, low and high frequencies are attenuated by human perception [2]. A study of this phenomenon lead to the creation of a standard set of equal-loudness curves, as discussed in the referenced work [4]. These curves represent the sound pressures required across all frequencies to create sounds with the same perceived loudness.

One of the works in the references identified a transfer function that approximates the filtering effect of the human ear and human perception [3]. This filtering effect corresponds to an equal-loudness curve. The transfer function from the referenced work is applied in this paper in order to perform A-weighting on audio signals that attenuates the high and low sound frequencies.

Signal Envelope

To identify the amplitude of an audio signal, the envelope of the sound signal needs to be identified. The envelope signal traces the maximum peaks of the original signal and removes the oscillatory components. Although there already exist methods to do this, such as the Hilbert Transform Filter [5], a technique is presented here that also preserves the duration of impulses in audio. Doing so allows further processing to be done to attenuate the amplitude of the impulses to better reflect how they are perceived by the human ear.

To obtain the amplitude envelope of the audio signal, the absolute values of the signal are first obtained. After this, a double sliding window is applied to each sample in the input signal. As shown in Figure 1, the sample on which the double window is being applied is shared by each of the half windows. For each half window, the maximum value is found. Finally, the output sample value is the minimum value of the two half window maxima. In the case of the example, the maximum in the first half window is 9, while the maximum in the second half window is 10. This means that the output sample value is 9.

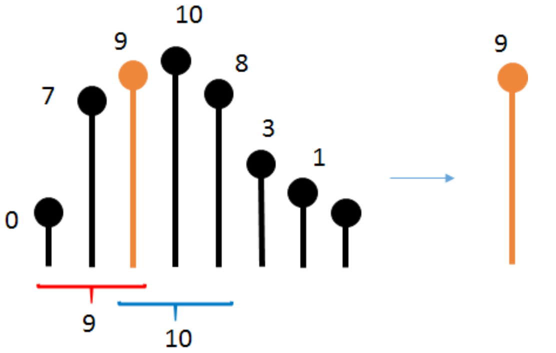

Fig. 1. Double sliding window.

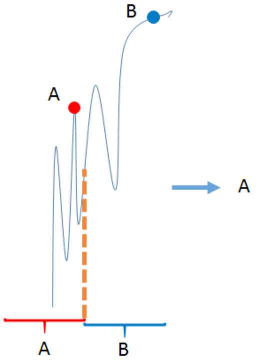

Fig. 2. Double sliding window, removing high frequency.

The double sliding window eliminates higher frequency vibrations by replacing oscillation values with nearby peak values. Figure 2 shows an example of this, where the input signal dips down because of a higher frequency vibration but is replaced with a neighboring maximum value, A, in the output.

This signal filter is able to preserve the width of impulses because this filter will only output a high value when samples from both sub-windows are high. For an impulse, this would only happen with the sliding window is directly centered underneath the impulse. Figure 3 provides an example of this.

The size of the double sliding window deter-

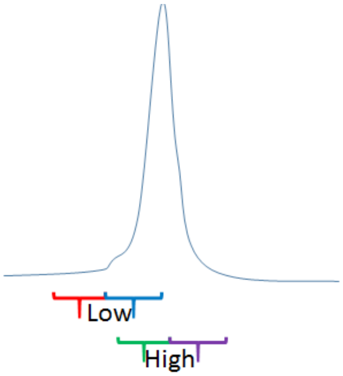

Fig. 3. Impulse double sliding window.

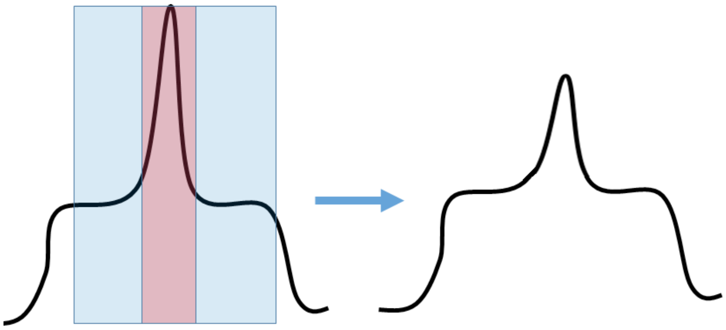

Fig. 4. Impulse correction.

mines the frequencies that will be eliminated in order to obtain the envelope function. Since humans are only able to hear sounds with frequency above 20 Hz [2], the window is selected such that it covers exactly one period of a 20 Hz sine wave. This eliminates all human hearable oscillations and only preserves the envelope of the signal.

Impulse Correction

The human ear is not as sensitive to short impulse sounds as to other sounds. The technique in this subsection provides a way to modify the signal so that short impulses are attenuated to match how they would be perceived. According to a figure in The Master Handbook of Acoustics, a 10 ms impulse needs to be roughly 10 dB higher than a 200 ms impulse for humans to perceive them as the same loudness. Moreover, the relationship between impulse duration (in milliseconds) and perceived loudness (in decibels) is roughly linear with respect to the logarithm of the impulse duration up until 200 ms [2]. Using this information, a simple linear equation can be found that relates the duration of an impulse to the sound pressure increase to maintain equal loudness. This relation is shown in Equation 1.

Equation 1: Sound Pressure to maintain equal loudness

$$P_{increase} \approx \frac{10\,(\log t\ -\ \log 200)}{(\log 10 -\ \log 200)}$$

Pincrease is the increase in sound pressure required to maintain equal perceived loudness in decibels and t is the duration of the impulse in milliseconds. The equation is valid for impulse durations less than 200 ms. Equation 2 shows the conversion between decibels and magnitude.

Equation 2: Pressure increase conversion from dB to gain ratio

$$P_{increase\_Mag} \approx 10^{\left(\frac{P_{increase}}{20}\right)}$$

In order to modify the signal, the impulses first need to be identified. Afterwards, for each impulse, the amplitude is scaled down by dividing the original impulse amplitude by $P_{increase\_Mag}$.

In order to identify the impulses in the signal, all of the peaks are identified in the signal. A peak in this context is where a sample is greater in value than both of its adjacent samples. Afterwards, for each peak, the area under the surrounding 200 ms is compared to that of an ideal (rectangular) impulse. The ideal impulse rectangle is then gradually shrunk in duration until the area under the signal is close to that of the rectangular impulse. When the area of the signal is within a certain threshold to that of an ideal impulse, the impulse in the signal is then shrunk based on the ratio in Equation 2.

Figure 4 shows an example of this impulse correction. The blue rectangle is the starting 200 ms area around the peak. The w idth of this rectangle is gradually reduced until it reaches the size of the red rectangle. At this point, the area within both the red rectangle and the signal envelope is greater than a certain proportion of the red rectangle's area. Experimentally, a threshold of 0.4 times the ideal rectangular impulse (red rectangle) area is effective at identifying most of the short sound spikes in the signal. Finally, the part of the signal overlapped by the red rectangle is scaled using the factor calculated in Equation 2. Equation 3 shows the formula used to calculate the corrected signal value. The corresponding values used in the equation, found in the original signal, can be found in Figure 5.

Equation 3: Impulse Correction Formula

$$Value_{new} = \frac{Value_{current} - \max(Base_R, Base_L)}{P_{increase_{Mag}}} + \max(Base_R, Base_L)$$

Testing the Methodology

In order to test this algorithm, three very different sound clips from the SYDE252 course at the University of Waterloo [6] are used. The first one is a drum sound clip containing many impulses. The second sound clip is that of a robin chirping, which contains high frequency sounds. Lastly, there is a sound clip of a person saying a sentence. The sound signals can be found in Figure 6, Figure 7, and Figure 8 respectively.

Results and Discussion

The three filtering techniques described in the previous section were applied successively to each signal.

For the drum loop audio signal, Figure 9 shows the frequency adjusted audio signal overlaid on top of the original drum loop signal. As shown, the higher frequency vibrations, which are less audible to the human ear are attenuated to better reflect the human's perception of sound.

In Figure 10, the blue signal shows the envelope of the signal in Figure 9. As shown, in the image, the double sliding window was able to effectively remove all of the higher frequency oscillations and output a signal that roughly represented the amplitude of the input. Finally, the orange signal in Figure 10 shows the output after the impulses have been shrunk accordingly. From the figure, it is possible to see that all of the short sound spikes have been scaled down.

A similar procedure has been done to the other sound clips. The resulting audio amplitude signals are shown in Figure 11 and Figure 12. From

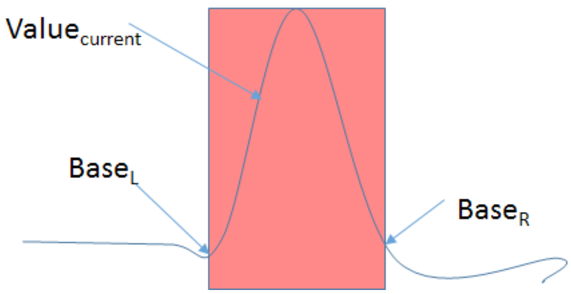

Fig. 5. Impulse correction variables.

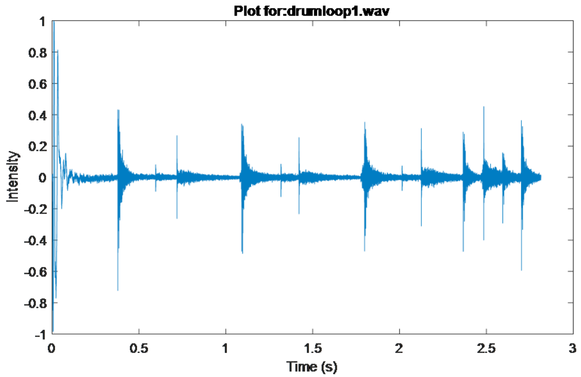

Fig. 6. Drum loop (audio signal).

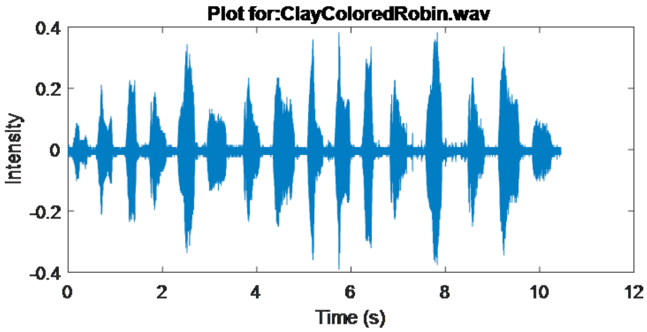

Fig. 7. Robin chirp (audio signal).

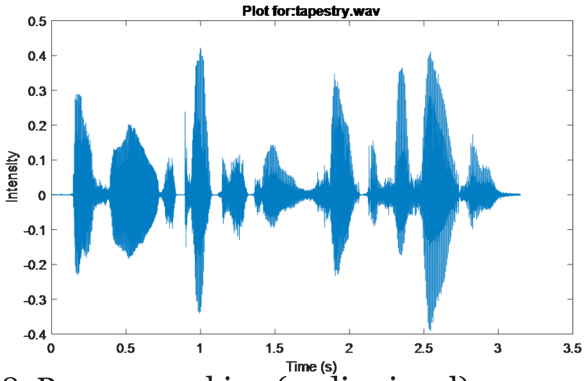

Fig. 8. Person speaking (audio signal).

these two figures, the amplitude of the audio signal at any given point can be easily estimated.

Once the amplitude of the sound file at any given point can be found, a computer algorithm can easily read off an estimate of how loud a sound signal is just by finding the signal's value at a particular point. Moreover, if it is desired to see whether sound amplitude reaches above a certain threshold, a simple thresholding algorithm can be applied to the processed signals.

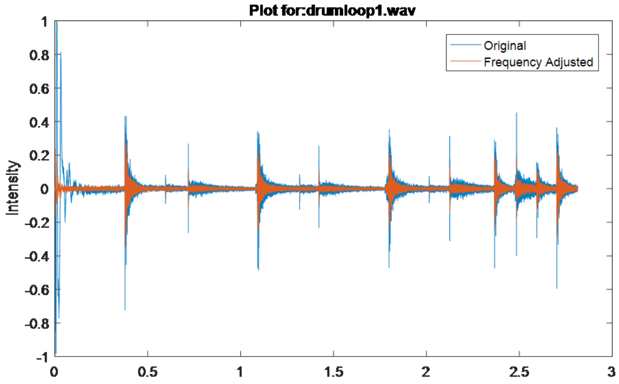

Fig. 9. Drum loop, frequency-adjusted.

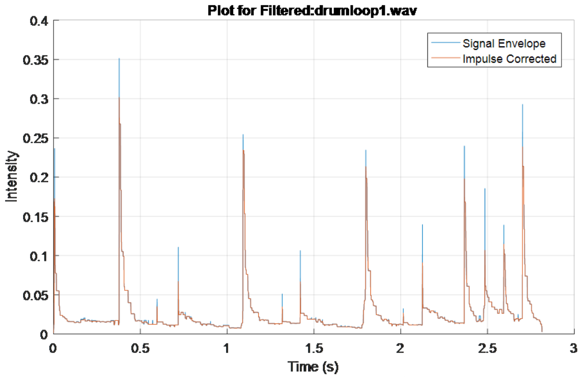

Fig. 10. Drum loop, output signal.

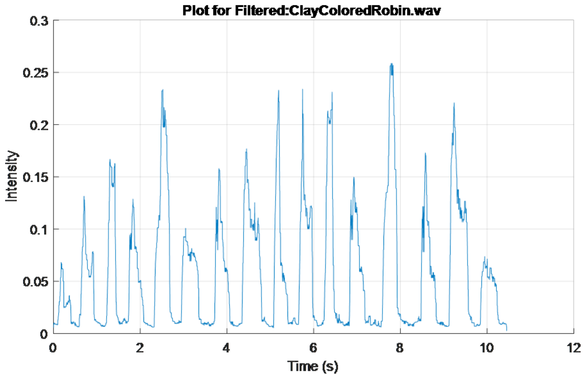

Fig. 11. Robin chirp, output signal.

Conclusion and Future Work

In conclusion, this paper presents a combination of techniques that can output the amplitude information of an audio signal. On top of this, certain adjustments are made so that the amplitude values better reflect what would be perceived by a human being. The method presented can be used so that a computer can algorithmically identify portions of an audio signal that humans would perceive as louder or quieter.

Although the method presented here does seem to work well for the three audio signals used in the experiment, more rigorous testing of this method should be done in the future.

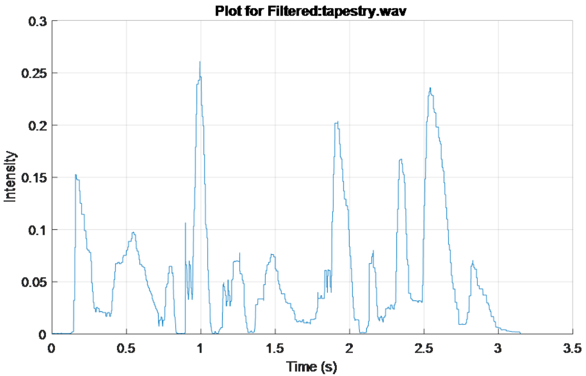

Fig. 12. Person speaking, output signal.

Works Cited

[1] G. Seshadri and B. Yegnanarayana, "Perceived loudness of speech based on the characteristics of glottal excitation source," Journal of the Acoustic Society of America, vol. 126, no. 4, pp. 2061-2071, 2009.

[2] F. A. Everest, The Master Handbook of Acoustics, New York: McGraw-Hill, 2001.

[3] A. N. Rimell, N. J. Mansfield and G. S. PAddan, "Design of digital filters for frequency weightings (A and C) required for risk assessments of workers exposed to noise," Industrial Health, vol. 53, no. 1, pp. 21-27, 2015.

[4] H. Moller and C. Pedersen, "Hearing at low and infrasonic frequencies," Noise & Health, vol. 6, no. 23, pp. 37-57, 2004.

[5] J. O. Smith III, Mathematics of the Discrete Fourier Transform (DFT): with Audio Applications Second Edition, W3K Publishing, 2007.

[6] University of Waterloo, "SYDE 252 Linear Systems & Signals," 2015. [Online]. Available: http://www.eng.uwaterloo.ca/~jzelek/teaching/syde252signals/Syde252/syde252-home.html. [Accessed 14 10 2016].