PodCast Script

MATH 2590

Mike Zabrocki

Podcast on Digital Sound

We are Kimberly, Vhenus and Mark, and we aim to explain to you what sound is, and how it is represented digitally.

Have you ever wondered what is sound? How is it that MP3s can one tenth of the size[2] and yet sound the same as CDs?

All sound is caused by some object vibrating. For example, when a guitar string vibrates, it moves from side to side. When it moves to one side, it pushes air particles on that side, causing them to push the ones in front of them, causing them to push other ones, and so on. This causes an increase in air pressure on that side and is called compression. Then, when the string begins to move back to the other side, it pulls back on the air particles, dropping pressure. This is called decompression. So we can say that compression and decompression of air causes the sound that we hear.

Now we all know that all sounds don't sound the same. There are two main characteristics that differentiate different sounds: volume and pitch. We all know what volume is – how loud or soft the sound is. Pitch is how high or low it sounds, relative to other sounds. For example, a flute has a higher pitch than a trombone. Mathematically, pitch is represented by frequency, which is measured in Hertz (Hz). Hz is the number of times the compressions and decompressions happen per second.

We can't hear all sounds, however. Humans typically have a hearing range of between 20Hz and 20000Hz. Dogs, on the other hand, can hear between 40Hz and 60000Hz. This is why humans can't hear dog whistles!

Volume is measured in decibels. At two extremes, a whisper quiet library is at 30dB while a loud rock concert is at approximately 115dB.

Music is just a combination of different sounds at different volume levels and pitches. When we listen to a CD, we're listening to an uncompressed recording that contains way more sound than our brain can actually process. On the other hand, MP3s sound very similar, but are one tenth of the size. How does this work?

There are two techniques that MP3 technology uses to compress file size. The first has to do with psychoacoustics, the study of how humans perceive sound, and the second has to do with compression of data.

Humans don't actually hear all the sound that surrounds them. For example, if there is one sound that is much louder than another, we only hear the louder one. This is why you can't hear a normal conversation happening when a bus is passing by, as you can only hear the bus.

When compressing a file, MP3 technology takes advantage of this fact by discarding sounds that are "masked" by other louder sounds. After the fact, we don't hear a difference as we wouldn't have heard the left out sound anyway. When the bus is passing by, whether or not your friend is talking, you're not going to hear them. So to save space, MP3 compression discards such conversations in music.

The second technique that MP3 technology uses has to do with file compression – getting rid of redundancies. This can be better explained by using a picture.

Think of a picture with just a blue sky and yellow sand. From our discussion on digital images, we know that computers store individual pixels. Therefore, for the entire blue sky, it might store a blue pixel 20,000 times in the picture. This is redundant as all 20,000 pixels are exactly the same. So when compressed, the compression algorithm would just store 1 blue pixel and then say it is repeated 20,000 times. Essentially, compression works by combining all similar pixels into one, thus saving space.

When compressing a picture, you can set how much the degree of compression should be. This is referred to as the tolerance. For instance, a low tolerance level would treat light blue and medium blue pixels as redundant and would store them as one colour. This would reduce the size, but would reduce quality as well. On the other hand, a high tolerance level would result in the compressor being more particular about colour differences, thus leading to a more-detailed picture but a larger image file. The same file compression technique can be used on sound.

So how does the whole process work? First, you set a tolerance level in terms of the maximum number of bits that can be used to store one second of audio. Once you set this tolerance, the song is broken down into frames, similar to frames of a movie. The compressing algorithm analyzes each frame and then discards masked audio based on psychoacoustic models. This eliminates sounds that humans wouldn't have heard anyway.

The song is then put back together to form a bit-stream and this is then run through the process of Huffman encoding, eliminating redundant data. Overall, after both compression techniques, the total number of bits per second will not exceed the set tolerance level. Thus, the same principle applies to sound as it did pictures – the lower the tolerance, the more data that has to be discarded, and the lower the quality of the resulting music.

Some principles to note are that the initial psychoacoustic-based compression is "lossy", indicating that the process destroys some information, and therefore it cannot be reversed. On the other hand, the Huffman compression, which is referred to as "lossless", works similar to a zip file. It gets rid of redundant data, but can be put back to form the original at any time.

Huffman compression, and by extension most lossless compression, works in the following manner. For text, characters are typically represented by 8 bits each. This allows for 256 possible characters. However, in most circumstances, all the possible characters are not used, and thus there are an unnecessary number of possible combinations of those 8 bits. Therefore, the number of bits required to represent each character is reduced based on the total number of characters in the content. For example, if the compressed content only requires 50 different characters, these can be represented using only 6 bits per character.

Huffman compression takes this one step further by using a variable number of bits for each character. Frequently used characters are given a shorter bit sequence while others that are more seldom used are given longer bit sequences. This further reduces the total number of bits that have to be used to represent a set of data.

All of this enables each of us to compress enormous volumes of music and carry in our pockets entire libraries of music!

Bibliography

[1] http://en.wikipedia.org/wiki/Audio_compression_(data)

[2] http://en.wikipedia.org/wiki/MP3

[3] http://www.cs.wfu.edu/~burg/CCLI/Templates/home.php

[4] http://oreilly.com/catalog/mp3/chapter/ch02.html#80629