Design Considerations for Audio Compression

Alexander Benjamin
8/01/10

The world of music is very different than it was ten years ago, which was markedly changed from ten years before that. Just as the ways in which people listen to music have evolved steadily over the last couple of decades, so have the demands and considerations of music listeners. People want music to have optimal sound quality while requiring as little space as possible (so they can listen to it on an MP3 player, for example). This paper will discuss the considerations and tradeoffs that go into audio compression as a determinant of audio quality and the amount of memory that the file occupies.

Music files are compressed to reduce the amount of data needed to play a song while minimizing the loss (if any) of perceptible sound quality. A decade ago, hard drive storage was much more expensive, and so the norm was to compress music. With the advent of digital audio players (MP3 players), compression is desired so that people can fit more songs onto a device. In addition, music compression is very useful for transferring the data over the internet or between devices. While download and transfer speeds have increased notably over the years, it takes much longer to download an uncompressed music file over the internet in comparison to a compressed one.

Different types of compression algorithms have been developed that aim to reduce or alter the digital audio data in order to reduce the number of bits needed for playback. The compression process is called encoding while the reverse is called decoding. From this, compression algorithms are commonly termed codecs (a combination of coder-decoder).

There are two types of general data compression: lossless and lossy. As suggested by the name, lossless compression algorithms do not permanently eliminate any of the original data or transform the digital data in an irreversible way. Thus, it is possible to reproduce an exact duplicate of the original digital data by decoding a losslessly compressed file. Conversely, lossy compression algorithms alter or completely remove digital data, rendering it impossible to reverse the process. As could be expected, lossy compression algorithms compress more than lossless algorithms. While a lossless algorithm generally reduces file size to about 50-60% of the original size, a lossy algorithm can typically achieve 5-20% of the original size and maintain reasonable sound quality.

All non-trivial compression algorithms use techniques to reduce information redundancy, which is when more bits than necessary are used to represent a sequence that could be represented with fewer bits. The most common techniques include coding, linear prediction, and pattern recognition.

Coding is altering the format in which data is represented. For example, a naïve form of coding used by fax machines, called run-length encoding, would represent "WWWWBBWWWWWW", the letters denoting pixel colors white or black on a page, as simply "4W2B6W"[i]. It is important to note that certain coding methods are only useful when applied to appropriate data. Using the same example, run-length encoding would not be useful for a format which varied regularly from each sample to the next (an image for example). If the sequence were instead "WBRO", it would encode to "1W1B1R1O", doubling the original size.

Linear prediction is a mathematical technique that attempts to estimate future values of a discrete-time signal as a linear function of previous examples[ii]. Linear predictive coding (abbreviated LPC) uses a linear predictive model to represent the spectral envelope of the digital data. It represents the intensity and frequency of a digital sample with numbers from which the original data can be reproduced.

Pattern recognition essentially uses a statistical model to analyze a set of data and determine parts that are repeated. Instead of preserving each instance of the repeated sequence, the pattern can be symbolically represented each time in order to save space. A unique symbolic description will be computed for each pattern, and this symbolic description will substitute the pattern[iii].

It is especially difficult to effectively compress certain music data losslessly because the waveforms are often much more complex than other audio waveforms such as human speech and don't lend as well to pattern detection or predictive models. Also, the variation is often chaotic even between individual digital samples – this means that consecutive sequences of bits (the "WWWWBBWWWWWW" example) don't occur very often. However, the data can be manipulated with certain signal processing techniques that make the previously described techniques more applicable. Codecs such as FLAC and Shorten, two well-known lossless compression algorithms, use linear prediction to estimate the spectrum of the signal. In order to use this technique, these codecs take the convolution with the filter [-1 1] which helps to whiten (i.e., make the power spectrum flatter) the signal. By reducing the variability in the spectrum, the linear predictive models are more effective. The whitening process can easily be reversed when the data is decoded by convolving with the reverse filter. Other codecs use similar filtering techniques in order to whiten the spectrum for linear predictive coding[iv].

Lossless music compression algorithms retain all of the high-fidelity audio data and can reproduce the original digital data via decompression. Thus, these algorithms are not judged on the quality of compression, but mainly on the effectiveness and speed. Certain algorithms optimize slightly more for the degree of compression (reducing file size), while others optimize slightly more for faster speeds of compression and decompression (useful especially for real-time applications).

The following is an analysis of how mainstream lossless music codecs perform in relation to each other and demonstrates performance differences in the above criteria. The music albums used in the test were selected to possess varying styles of music in order to show the effect of music style on the effectiveness and speed of the codecs. As a standard of comparison, the wave format is uncompressed and so can be used to judge the degree of compression.

Album	Wave	FLAC	WavPack	Shorten	Monkey's Audio	OptimFROG	album average
Life of Destructor	103	101	119	142	130	252	141.2
Exit Planet Dust	107	105	122	150	137	267	148
Demon Days	109	108	125	153	141	273	151.5
The W	123	123	151	176	162	313	174.7
Endtroducing.	126	127	153	179	176	328	181.5
The Best of Nick Cave...	136	146	192	203	204	381	210.3
Superunkown	140	146	187	208	209	381	211.8
Barrio Fino	140	150	198	209	212	392	216.8
Resident Evil: Apocalypse	135	151	196	211	211	398	217
The Essential Classics Collection	146	200	173	203	278	372	228.7
format average	**126.5**	**135.7**	**161.6**	**183.4**	**186**	**335.7**	188.2

Fig 1. Lossless: Time in seconds needed to rip and encode each album with each codec

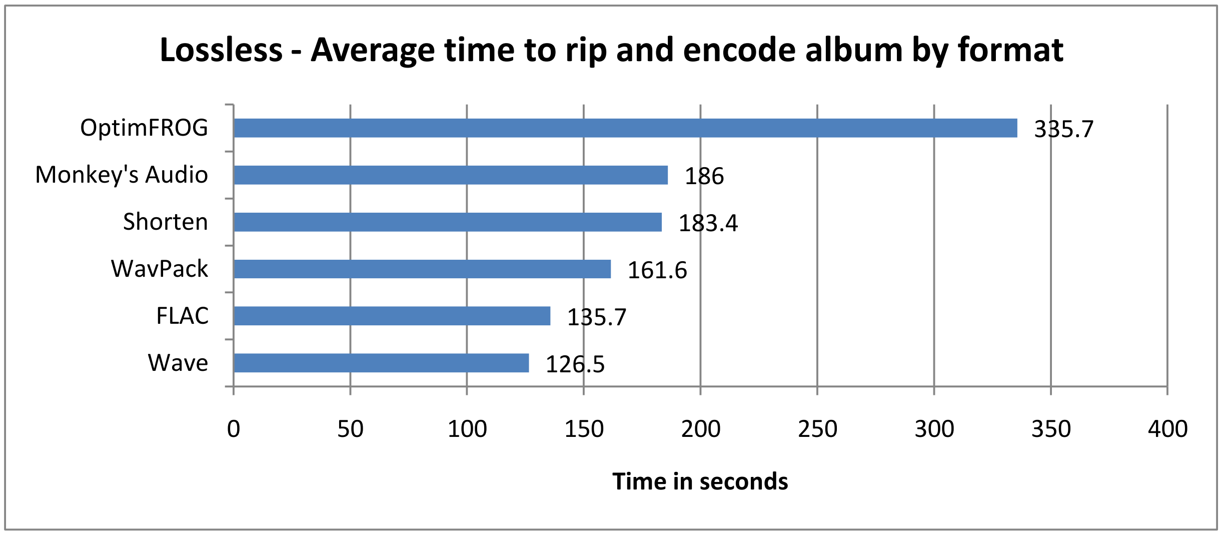

Fig 2. Lossless: Average time to and encode each album by format

As can be seen, there is significant variation in both the time to rip and encode different albums based on the complexity of the digital data as well as the performance of each codec. The tradeoff to speed of the compression process is the degree of compression that each codec achieves. It should be noted that the tradeoff is not linear, and that some codecs are better at encoding and decoding, as well as compressing, some styles of music.

Album	Monkey's Audio	Optim FROG	FLAC	WavPack	Shorten	Wave (uncomp.)	album average	Compression
Life of Destructor	300E+6	302E+6	317E+6	319E+6	336E+6	488E+6	344E+6	70.45%
Exit Planet Dust	314E+6	315E+6	331E+6	336E+6	372E+6	523E+6	365E+6	69.81%
The Essential Classics Collection	282E+6	285E+6	303E+6	300E+6	310E+6	742E+6	371E+6	49.93%
Demon Days	321E+6	322E+6	344E+6	344E+6	372E+6	538E+6	373E+6	69.42%
The W	352E+6	354E+6	372E+6	373E+6	435E+6	626E+6	419E+6	66.84%
Endtroducing.	358E+6	359E+6	382E+6	383E+6	420E+6	668E+6	428E+6	64.17%
Superunknown	501E+6	504E+6	522E+6	525E+6	558E+6	780E+6	565E+6	72.44%
The Best of Nick Cave...	492E+6	499E+6	519E+6	526E+6	553E+6	800E+6	565E+6	70.59%
Barrio Fino	544E+6	547E+6	569E+6	571E+6	594E+6	808E+6	605E+6	74.90%
Resident Evil: Apocalypse	554E+6	559E+6	576E+6	583E+6	610E+6	807E+6	615E+6	76.22%
format average	**402E+6**	**405E+6**	**424E+6**	**426E+6**	**456E+6**	**678E+6**	**465E+6**	**68.58%**

Fig 3. Lossless: File size in bytes of each album encoded with each codec

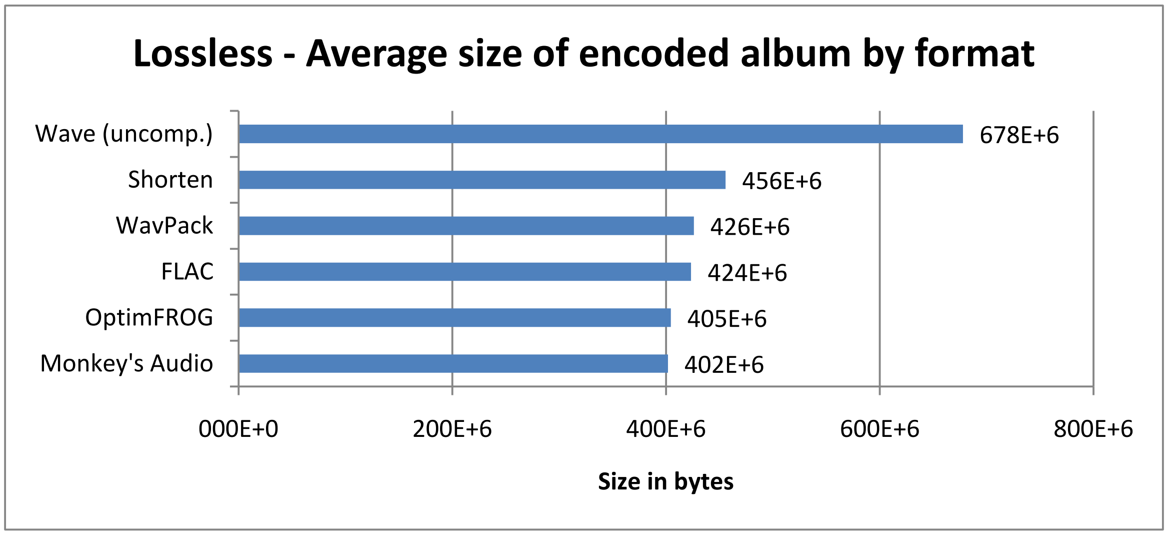

Fig. 4 Lossless: Average size of each encoded album for each codec

Album	Monkey's Audio	Optim FROG	FLAC	WavPack	Shorten	Wave (uncomp.)
Life of Destructor	61.60%	61.87%	65.94%	65.46%	68.85%	100.00%
Exit Planet Dust	60.04%	60.18%	63.30%	64.22%	71.08%	100.00%
The Essential Classics Collection	38.04%	38.44%	40.82%	40.45%	41.82%	100.00%
Demon Days	59.68%	59.81%	63.90%	64.00%	69.12%	100.00%
The W	56.21%	56.45%	59.39%	59.61%	69.42%	100.00%
Endtroducing.	53.57%	53.85%	57.24%	57.40%	62.95%	100.00%
Superunknown	64.20%	64.63%	66.94%	67.35%	71.52%	100.00%
The Best of Nick Cave...	61.50%	62.35%	64.89%	65.76%	69.05%	100.00%
Barrio Fino	67.31%	67.63%	70.41%	70.59%	73.48%	100.00%
Resident Evil: Apocalypse	68.74%	69.35%	71.40%	72.23%	75.58%	100.00%
Average codec compressed size:	59.09%	59.46%	62.30%	62.71%	67.29%	100.00%

Fig 5. Lossless: Percentage of original file size after compression with each codec

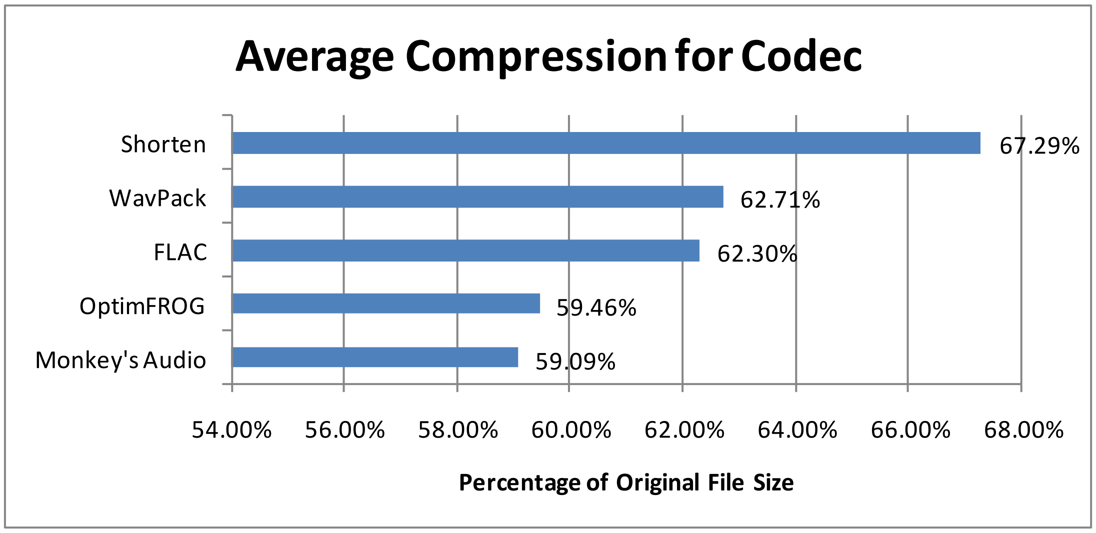

Fig 6. Lossless: Percentage of original file size after compression with each codec

The first point that becomes immediately obvious is that none of the encoders ‘wins’ in multiple categories. This reinforces the point that the effectiveness of compression must be traded in order to achieve speed. The results do suggest that certain codecs have optimized the tradeoff better; in other words, they are not sacrificing as much time in order to achieve good compression levels.

From these results, one can see that OptimFROG compresses very well in relation to the other codecs; however, the time it takes for compression is almost double that of any of the other codecs. This result suggests what one would have expected given familiarity with the codec – the makers of OptimFROG attempted to optimize almost entirely for the degree of compression[v].

The codecs which, based on these results, appear to achieve the best results are Monkey's Audio and FLAC. The former achieves notably greater compression, while the latter achieves notably better compression time. Depending on the specific desire of the user, one of these codecs seems best for general use.

To verify that compression was indeed lossless, MD5, SHA1 and CRC-32 hash values were generated for the uncompressed wave and then a cycle of compression/decompression was done with each codec[vi]. The hash values were also generated for the resulting file after the compression/decompression. The hash values from the original file were identical to the other files, and the files were the same size.

The other type of compression is termed as 'lossy'. Lossy compression is actually much more widely used than lossless, which might seem surprising at first. However, lossy compression can achieve size reductions to about 5-20% of the original file size and still have good quality such that the music is indistinguishable from the original file for the average listener. The appeal for mainstream applications is primarily in the transfer rate of the digital data. Digital television, radio broadcasts, DVDs, and satellite all use lossy compression to make transfer rates faster and have the data more accessible to end-users.

Lossy compression focuses on discarding or de-emphasizing pieces of the digital data that are either imperceptible to the human ear, or else are less critical. Unlike with lossless compression, data that is compressed lossily is compressed until it reaches a target file size. The target file size is specified as a bit rate; in other words, how many bits can be used to describe each sample of music. One can think of this as similar to a color palette for images. If a photograph is redrawn using only 16 colors, details will be missing and the image will appear drastically different than the original. The more colors that are used to draw the image, the more details will be noticeable. While there could theoretically be an infinite number of color shades, at a certain point the human eye will no longer be able to detect further detail. This analogy translates very well to the idea of lossy music compression and bit rates. A high-fidelity music file contains far more information, or detail, than the human ear can detect. For this reason, it is easy to take away a significant amount of digital data without noticeably changing the way a person perceives it. The average compression threshold, or bit rate, at which a person can notice the difference in sound is a topic that is regularly under debate and of interest to both music listeners and music producers.

It is important to understand how lossy compression codecs determine what digital data is less-critical in relation to other data. While the same techniques used in lossless compression are still applied to lossy compression, lossy codecs have the added responsibility of deciding what sounds to remove rather than just how to compress. Also, the codecs need to calculate how many bits (i.e., how much detail) to devote to certain sounds.

The guiding theory in lossy compression comes primarily from the study of psychoacoustics. Psychoacoustics recognizes that not all sounds are perceptible by the human ear. There is a finite frequency range at which a person can detect sounds, and also a minimum sound level (called the absolute threshold of hearing). While humans can, under certain circumstances, hear sounds in the range of 20 Hz to 20,000 Hz, the upper and lower thresholds are generally lower for most people and the range decreases with age[vii]. There is also a limit at which the ear can perceive changes in pitch. The frequency resolution between 1-2 kHz is 3.6 Hz, meaning that any pitch change smaller than that is unnoticeable[viii]. Finally, the concept of auditory masking describes how the perception of one sound can be affected by the presence of another sound based on their relative volumes and frequencies[ix]. Depending on the attributes of simultaneous sounds, some sounds may be completely masked, or less prominent, to a listener. Lossy compression algorithms take advantage of psychoacoustics to determine which sounds can be entirely removed, or else described less completely (with less bits).

Since lossy compression algorithms are generally set for a target bit rate, the pieces of digital data that become categorized as less-critical will change depending on the degree to which a file is compressed. Since high-fidelity music contains much more digital data than is critical, these algorithms can achieve very high levels of compression without perceptibly changing what a human can hear. Obviously, to reduce the bit rate to a very low level, lossy algorithms are forced to sacrifice noticeable sound quality.

Modern codecs can also encode music at a constant bit rate (CBR) or variable bit rate (VBR). Constant bit rate means that every sample will be described using the same number of bits. If the target bit rate is 128 Kbits/s, every digital data sample will have that level of detail. Variable bit rate allows the encoder to use more or less bits per individual sample while maintaining the average rate across the song. Using VBR, an encoder could use fewer bits to describe a more simple part of a song, and then use the excess bits to describe the more complex passage. A variable bit rate is almost always desirable as it allows music to be more accurately represented at lower bit rates.

The following is an analysis of how mainstream lossy music codecs perform in relation to each other and demonstrates performance differences. It is important to note that this analysis does not address the quality of the music when compressed which is clearly a vital component when judging a lossy compression algorithm. It would require polling and blind testing from a wide audience beyond the current scope of this analysis to determine reliable results. Instead, other studies in this area will be addressed to discuss perceptible quality differences as a factor of bit rate. This music is encoded at 192 Kbits/s except for AAC (MP4) at 150 Kbits/s, and Musepack at 210 Kbits/s (these levels are generally accepted to have no obvious quality problems for their respective codecs). All are encoded using VBR with the exception of WMA which uses CBR. The music albums used in the test were selected to possess varying styles of music to show the effect of music style on the effectiveness and speed of the codecs.

Album	Musepack	MP3	Ogg Vorbis	AAC	WMA	album average
Life of Destructor	199	278	305	345	409	307.2
Exit Planet Dust	213	301	325	372	437	329.6
Demon Days	220	301	335	388	438	336.4
The W	251	366	387	429	492	385
Endtroducing.	272	372	406	484	503	407.4
Superunknown	319	433	481	575	595	480.6
The Best of Nick Cave...	326	441	501	573	631	494.4
Resident Evil: Apocalypse	335	453	511	602	629	506
The Essential Classics Collection	397	498	540	644	454	506.6
Barrio Fino	334	453	515	558	707	513.4
format average	**286.6**	**390**	**430.6**	**497**	**529.5**	**426.7**

Fig 7. Lossy: The time in seconds to rip and encode each album with each codec

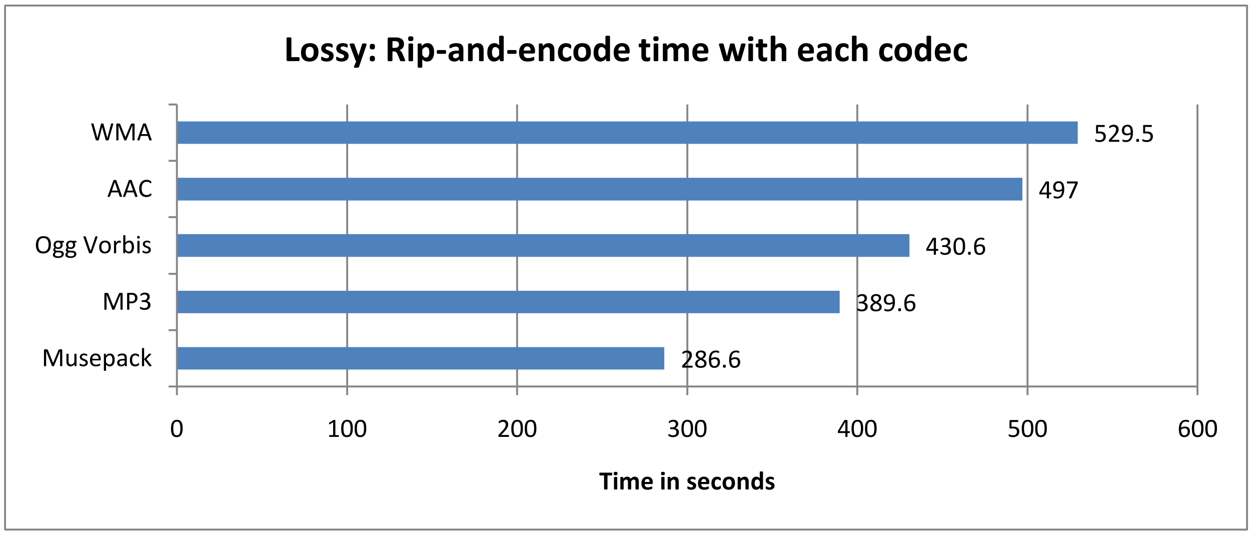

Fig 8. Lossy: Average rip-and-encode time for each codec

There are several interesting points to note. First, the timing was fairly widely distributed between the codecs; no two results were within 30 seconds of each other. Also, the times for lossy encoding were uniformly much longer than the general times for lossless encoding. This demonstrates that the calculations and models that use psychoacoustic theory to remove or de-emphasize add significant run-time.

Album	Musepack	MP3	Ogg Vorbis	AAC	WMA	album average
Life of Destructor	62,463,762	65,799,179	66,873,414	73,298,256	67,403,568	67,167,636
Exit Planet Dust	62,097,805	73,377,984	71,790,841	69,945,526	66,040,295	68,650,490
Demon Days	67,126,673	73,605,551	73,985,105	71,938,995	75,217,395	72,374,744
The W	72,919,908	81,801,789	86,010,215	80,429,332	72,028,211	78,637,891
Endtroducing.	76,155,602	78,872,240	91,563,952	84,542,617	87,753,080	83,777,498
Superunknown	90,920,669	82,060,838	101,489,253	96,157,939	111,596,278	96,444,995
The Best of Nick Cave...	98,835,214	104,856,944	106,900,824	113,618,837	111,940,344	107,230,433
Resident Evil: Apocalypse	104,893,796	104,790,872	109,567,328	110,509,197	115,598,661	109,071,971
The Essential Classics Collection	108,742,066	111,773,130	110,507,870	128,621,332	118,082,946	115,545,469
Barrio Fino	115,936,750	115,799,822	111,235,664	120,693,381	130,805,720	118,894,267
format average	**86,009,225**	**89,273,835**	**92,992,447**	**94,975,541**	**95,646,650**	**91,779,539**

Fig. 9: Lossy: File size in bytes of each album encoded with each codec

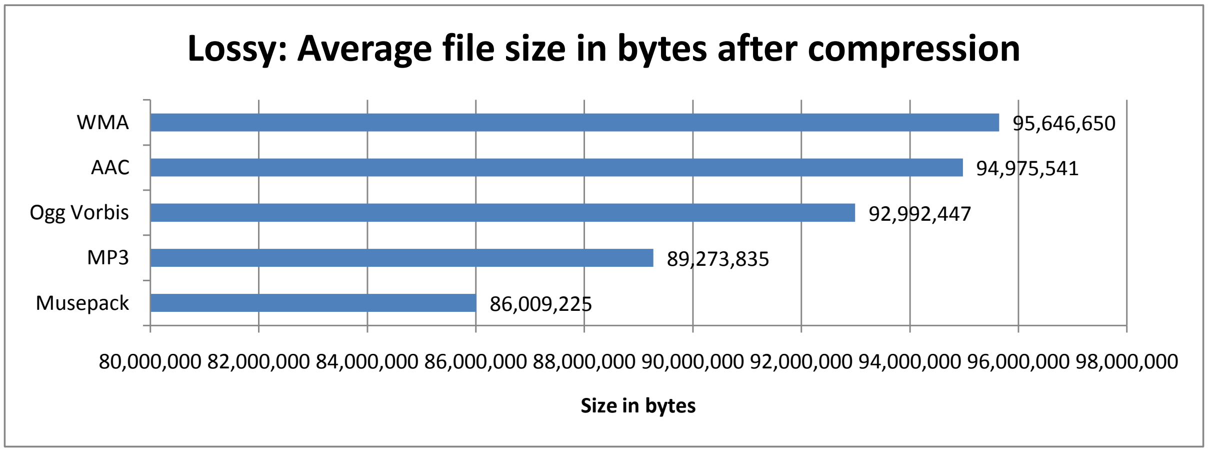

Fig. 10. Lossy: Average file size in bytes for each codec after compression

The results for the file size between different codecs were interesting. There was quite a bit of variation in the file size of each album between each codec. The results also show that the codec most successful in compressing varied between albums, suggesting that there is a potentially significant variation in the effectiveness of each codec for different music styles.

Album	Musepack	MP3	Ogg Vorbis	AAC	WMA
Life of Destructor	12.80%	13.49%	13.71%	15.03%	13.82%
Exit Planet Dust	11.87%	14.03%	13.72%	13.37%	12.63%
Demon Days	9.04%	9.91%	9.96%	9.69%	10.13%
The W	13.56%	15.21%	15.99%	14.95%	13.39%
Endtroducing.	12.16%	12.59%	14.62%	13.50%	14.01%
Superunknown	13.62%	12.29%	15.20%	14.41%	16.72%
The Best of Nick Cave...	12.68%	13.45%	13.71%	14.57%	14.36%
Resident Evil: Apocalypse	13.11%	13.10%	13.69%	13.81%	14.45%
The Essential Classics Collection	13.45%	13.83%	13.67%	15.91%	14.61%
Barrio Fino	14.37%	14.35%	13.79%	14.96%	16.21%
format average	**12.67%**	**13.22%**	**13.81%**	**14.02%**	**14.03%**

Fig. 11. Lossy: Compression rates for each album by codec

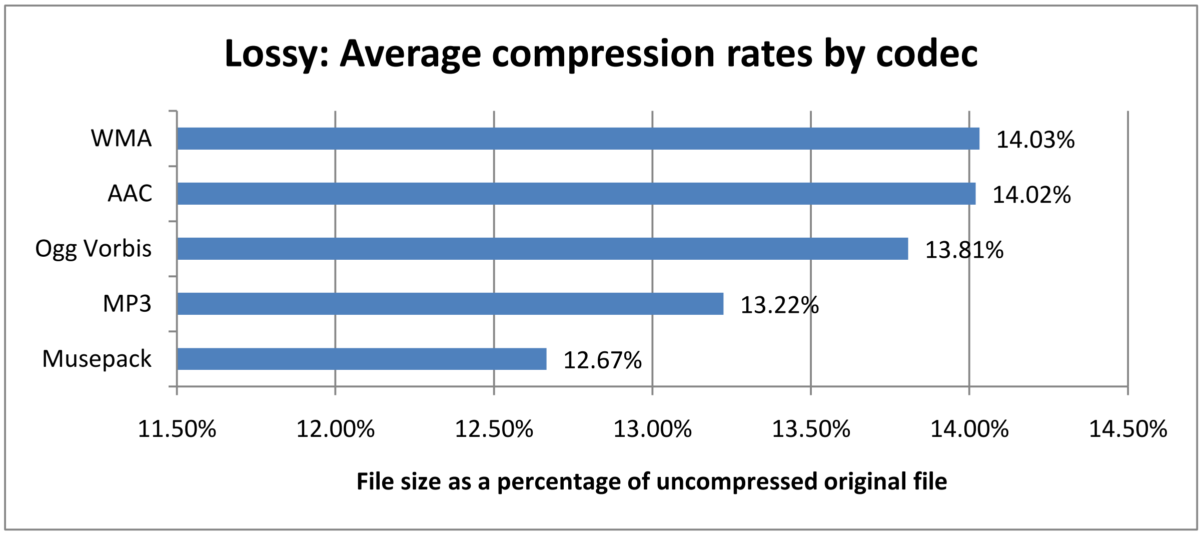

Fig. 12. Lossy: Average compression rates by codec

The degree of compression for lossy compression is quite astounding when taking into account the fact that the quality level at these compression levels is generally accepted to be almost, if not entirely, indistinguishable from an uncompressed file. This shows that lossy compression algorithms are probably desirable for applications other than archiving music. These files were compressed to roughly 192 Kbits/s and achieved around 13.5% of the original file size.

As mentioned previously, this still does not address the question of which codec produces the highest quality recording when compressed to a given bit rate. When music files are compressed, the compression can leave artifacts if the data is represented with too few bits to preserve all of the perceivable sound. Whether or not a listener will be able to hear the difference between music files at various rates of compression depends on a number of factors, including familiarity with the music, quality of the sound source, and sensitivity of the listener.

Conducting a reliable experiment regarding the bit rate threshold at which the music becomes distinguishably different is outside the scope of this analysis. The pursuit in itself seems somewhat futile as the answer will always depend on subjective factors such as the quality of the listener's sound system and how attuned his or her ear is to music.

The conclusion of a majority of studies shows that 128 Kbits/s will provide good quality for the average listener on a typical sound system. A bit rate any lower than that will probably yield obvious compression artifacts that may bother some listeners more than others. The obvious recommendation would be to encode one's music at no less than 128 Kbits/s and to experiment with one's own set of circumstances to determine at what bit rate one can no longer notice a difference. It is generally accepted that non-experts will not be able to tell the difference between music encoded at 192 Kbits/s and higher bit rates.

The field of audio compression is one in constant development as the models, optimizations, and algorithms continue to become more advanced. While different people will likely always have very individualized preferences for music format and quality, the topic is of interest to users on both ends of the spectrum. Those who produce music are especially interested, because it can affect the design decisions for making music. A song that is mixed to sound best at 320 Kbits/s may not sound as good at lower bit rates. However, recognizing that mainstream listeners use iPods and other portable devices that rarely use files greater than 128 Kbits/s, some music producers have started designing songs to sound their best when compressed at these levels. There are so many individual variables that factor into a specific user's music preferences that there is likely not any best answer. However, the above analysis should certainly provide some indication of the considerations and tradeoffs that should be considered.

---

[i] Pountain, D. (1978), "Run Length Encoding", *BYTE Publications Inc.*

[ii] Makhoul, J. (1975), "Linear Prediction: A Tutorial Review," *Proceedings of the IEEE, 63*.

[iii] Hornegger, Joachim; Paulus, Dietrich W. R. (1999). *Applied Pattern Recognition: A Practical Introduction to Image and Speech Processing in C++* (4th ed.)

[iv] Rocchesso, D., *Introduction to Sound Processing*. Edizioni di Mondo Estremo, Firenze, 2003

[v] OptimFROG creators: http://www.losslessaudio.org/

[vi] Using "HashTab Shell Extension": http://beeblebrox.org/

[vii] Plack, Christopher, J. (2005). *The Sense of Hearing*. Routledge. ISBN 0805848843.

[viii] Olson, Harry F. (1967). *Music, Physics and Engineering*. Dover Publications. pp. 248–251. ISBN 0486217698.

[ix] Gelfand, S.A. (2004) *Hearing: An Introduction to Psychological and Physiological Acoustics* 4th Ed. New York, Marcel Dekker