Article   Talk

Read   Edit   View history

Search

Data compression

From Wikipedia, the free encyclopedia
(Redirected from Video compression)

"Source coding" redirects here. For the term in computer programming, see Source code.

In digital signal processing, **data compression**, **source coding**,[1] or **bit-rate reduction** involves encoding information using fewer bits than the original representation.[2] Compression can be either lossy or lossless. Lossless compression reduces bits by identifying and eliminating statistical redundancy. No information is lost in lossless compression. Lossy compression reduces bits by identifying unnecessary information and removing it.[3] The process of reducing the size of a data file is referred to as data compression. In the context of data transmission, it is called source coding (encoding done at the source of the data before it is stored or transmitted) in opposition to channel coding.[4]

Compression is useful because it helps reduce resource usage, such as data storage space or transmission capacity. Because compressed data must be decompressed to use, this extra processing imposes computational or other costs through decompression; this situation is far from being a free lunch. Data compression is subject to a space–time complexity trade-off. For instance, a compression scheme for video may require expensive hardware for the video to be decompressed fast enough to be viewed as it is being decompressed, and the option to decompress the video in full before watching it may be inconvenient or require additional storage. The design of data compression schemes involves trade-offs among various factors, including the degree of compression, the amount of distortion introduced (when using lossy data compression), and the computational resources required to compress and uncompress the data.[5][6]

Lossless   [edit]

Lossless data compression algorithms usually exploit statistical redundancy to represent data more concisely without losing information, so that the process is reversible. Lossless compression is possible because most real-world data have statistical redundancy. For example, an image may have areas of colour that do not change over several pixels; instead of coding "red pixel, red pixel, ..." the data may be encoded as "279 red pixels". This is a basic example of run-length encoding; there are many schemes to reduce file size by eliminating redundancy.

The Lempel–Ziv (LZ) compression methods are among the most popular algorithms for lossless storage.[7] DEFLATE is a variation on LZ optimized for decompression speed and compression ratio, but compression can be slow. DEFLATE is used in PKZIP, Gzip and PNG. LZW (Lempel–Ziv–Welch) is used in GIF images. Also noteworthy is the LZR (Lempel-Ziv–Renau) algorithm, which serves as the basis for the Zip method. LZ methods use a table-based compression model where table entries are substituted for repeated strings of data. For most LZ methods, this table is generated dynamically from earlier data in the input. The table itself is often Huffman encoded (e.g. SHRI, LZX). A current LZ-based coding scheme that performs well is LZX, used in Microsoft's CAB format.

The best modern lossless compressors use probabilistic models, such as prediction by partial matching. The Burrows–Wheeler transform can also be viewed as an indirect form of statistical modelling.[8]

The class of grammar-based codes are gaining popularity because they can compress *highly repetitive* input extremely effectively, for instance, a biological data collection of the same or closely related species, a huge versioned document collection, internet archival, etc. The basic task of grammar-based codes is constructing a context-free grammar deriving a single string. Sequitur and Re-Pair are practical grammar compression algorithms for which software is publicly available.

In a further refinement of the direct use of probabilistic modelling, statistical estimates can be coupled to an algorithm called arithmetic coding. Arithmetic coding is a more modern coding technique that uses the mathematical calculations of a finite-state machine to produce a string of encoded bits from a series of input data symbols. It can achieve superior compression to other techniques such as the better-known Huffman algorithm. It uses an internal memory state to avoid the need to perform a one-to-one mapping of individual input symbols to distinct representations that use an integer number of bits, and it clears out the internal memory only after encoding the entire string of data symbols. Arithmetic coding applies especially well to adaptive data compression tasks where the statistics vary and are context-dependent, as it can be easily coupled with an adaptive model of the probability distribution of the input data. An early example of the use of arithmetic coding was its use as an optional (but not widely used) feature of the JPEG image coding standard. It has since been applied in various other designs including H.264/MPEG-4 AVC and HEVC for video coding.

Lossy   [edit]

Lossy data compression is the converse of lossless data compression. In these schemes, some loss of information is acceptable. Dropping nonessential detail from the data source can save storage space. Lossy data compression schemes are informed by research on how people perceive the data in question. For example, the human eye is more sensitive to subtle variations in luminance than it is to the variations in color. JPEG image compression works in part by rounding off nonessential bits of information.[9] There is a corresponding trade-off between preserving information and reducing size. A number of popular compression formats exploit these perceptual differences, including those used in music files, images, and video.

Lossy image compression can be used in digital cameras, to increase storage capacities with minimal degradation of picture quality. Similarly, DVDs use the lossy MPEG-2 video coding format for video compression.

In lossy audio compression, methods of psychoacoustics are used to remove non-audible (or less audible) components of the audio signal. Compression of human speech is often performed with even more specialized techniques; speech coding, or voice coding, is sometimes distinguished as a separate discipline from audio compression. Different audio and speech compression standards are listed under audio coding formats. Voice compression is used in internet telephony, for example, audio compression is used for CD ripping and is decoded by the audio players.[8]

Theory   [edit]

The theoretical background of compression is provided by information theory (which is closely related to algorithmic information theory) for lossless compression and rate–distortion theory for lossy compression. These areas of study were essentially forged by Claude Shannon, who published fundamental papers on the topic in the late 1940s and early 1950s. Coding theory is also related to this. The idea of data compression is also deeply connected with statistical inference.[10]

Machine learning   [edit]

See also: Machine learning

There is a close connection between machine learning and compression: a system that predicts the posterior probabilities of a sequence given its entire history can be used for optimal data compression (by using arithmetic coding on the output distribution) while an optimal compressor can be used for prediction (by finding the symbol that compresses best, given the previous history). This equivalence has been used as a justification for using data compression as a benchmark for "general intelligence."[11]

Data differencing   [edit]

Main article: Data differencing

Data compression can be viewed as a special case of data differencing.[12][13] Data differencing consists of producing a *difference* given a *source* and a *target*, with patching producing a *target* given a *source* and a *difference*, while data compression consists of producing a compressed file given a target, and decompression consists of producing a target given only a compressed file. Thus, one can consider data compression as data differencing with empty source data, the compressed file corresponding to a "difference from nothing." This is the same as considering absolute entropy (corresponding to data compression) as a special case of relative entropy (corresponding to data differencing) with no initial data.

When one wishes to emphasize the connection, one may use the term *differential compression* to refer to data differencing.

Uses [edit]

Audio [edit]

See also: Audio codec and Audio coding format

Audio data compression, as distinguished from dynamic range compression, has the potential to reduce the transmission bandwidth and storage requirements of audio data. Audio compression algorithms are implemented in software as audio codecs. Lossy audio compression algorithms provide higher compression at the cost of fidelity and are used in numerous audio applications. These algorithms almost all rely on psychoacoustics to eliminate less audible or meaningful sounds, thereby reducing the space required to store or transmit them.[2]

In both lossy and lossless compression, information redundancy is reduced, using methods such as coding, pattern recognition, and linear prediction to reduce the amount of information used to represent the uncompressed data.

The acceptable trade-off between loss of audio quality and transmission or storage size depends upon the application. For example, one 640MB compact disc (CD) holds approximately one hour of uncompressed high fidelity music, less than 2 hours of music compressed losslessly, or 7 hours of music compressed in the MP3 format at a medium bit rate. A digital sound recorder can typically store around 200 hours of clearly intelligible speech in 640MB.[14]

Lossless audio compression produces a representation of digital data that decompress to an exact digital duplicate of the original audio stream, unlike playback from lossy compression techniques such as Vorbis and MP3. Compression ratios are around 50–60% of original size,[15] which is similar to those for generic lossless data compression. Lossless compression is unable to attain high compression ratios due to the complexity of waveforms and the rapid changes in sound forms. Codecs like FLAC, Shorten and TTA use linear prediction to estimate the spectrum of the signal. Many of these algorithms use convolution with the filter [-1 1] to slightly whiten or flatten the spectrum, thereby allowing traditional lossless compression to work more efficiently. The process is reversed upon decompression.

When audio files are to be processed, either by further compression or for editing, it is desirable to work from an unchanged original (uncompressed or losslessly compressed). Processing of a lossily compressed file for some purpose usually produces a final result inferior to the creation of the same compressed file from an uncompressed original. In addition to sound editing or mixing, lossless audio compression is often used for archival storage, or as master copies.

A number of lossless audio compression formats exist. Shorten was an early lossless format. Newer ones include Free Lossless Audio Codec (FLAC), Apple's Apple Lossless (ALAC), MPEG-4 ALS, Microsoft's Windows Media Audio 9 Lossless (WMA Lossless), Monkey's Audio, TTA, and WavPack. See list of lossless codecs for a complete listing.

Some audio formats feature a combination of a lossy format and a lossless correction; this allows stripping the correction to easily obtain a lossy file. Such formats include MPEG-4 SLS (Scalable to Lossless), WavPack, and OptimFROG DualStream.

Other formats are associated with a distinct system, such as:

- Direct Stream Transfer, used in Super Audio CD
- Meridian Lossless Packing, used in DVD-Audio, Dolby TrueHD, Blu-ray and HD DVD

Lossy audio compression [edit]

Lossy audio compression is used in a wide range of applications. In addition to the direct applications (mp3 players or computers), digitally compressed audio streams are used in most video DVDs, digital television, streaming media on the internet, satellite and cable radio, and increasingly in terrestrial radio broadcasts. Lossy compression typically achieves far greater compression than lossless compression (data of 5 percent to 20 percent of the original stream, rather than 50 percent to 60 percent), by discarding less-critical data.[16]

The innovation of lossy audio compression was to use psychoacoustics to recognize that not all data in an audio stream can be perceived by the human auditory system. Most lossy compression reduces perceptual redundancy by first identifying perceptually irrelevant sounds, that is, sounds that are very hard to hear. Typical examples include high frequencies or sounds that occur at the same time as louder sounds. Those sounds are coded with decreased accuracy or not at all.

Due to the nature of lossy algorithms, audio quality suffers when a file is decompressed and recompressed (digital generation loss). This makes lossy compression unsuitable for storing the intermediate results in professional audio engineering applications, such as sound editing and multitrack recording. However, they are very popular with end users (particularly MP3) as a megabyte can store about a minute's worth of music at adequate quality.

Comparison of acoustic spectrograms of a song in an uncompressed format and lossy formats. That the lossy spectrograms are different from the uncompressed one indicates that they are, in fact, lossy, but nothing can be assumed about the effect of the changes on perceived quality.

Coding methods [edit]

To determine what information in an audio signal is perceptually irrelevant, most lossy compression algorithms use transforms such as the modified discrete cosine transform (MDCT) to convert time domain sampled waveforms into a transform domain. Once transformed, typically into the frequency domain, component frequencies can be allocated bits according to how audible they are. Audibility of spectral components calculated using the absolute threshold of hearing and the principles of simultaneous masking—the phenomenon wherein a signal is masked by another signal separated by frequency—and, in some cases, temporal masking—where a signal is masked by another signal separated by time. Equal-loudness contours may also be used to weight the perceptual importance of components. Models of the human ear-brain combination incorporating such effects are often called psychoacoustic models.[17]

Other types of lossy compressors, such as the linear predictive coding (LPC) used with speech, are source-based coders. These coders use a model of the sound's generator (such as the human vocal tract with LPC) to whiten the audio signal (i.e., flatten its spectrum) before quantization. LPC may be thought of as a basic perceptual coding technique: reconstruction of an audio signal using a linear predictor shapes the coder's quantization noise into the spectrum of the target signal, partially masking it.[16]

Lossy formats are often used for the distribution of streaming audio or interactive applications (such as the coding of speech for digital transmission in cell phone networks). In such applications, the data must be decompressed as the data flows, rather than after the entire data stream has been transmitted. Not all audio codecs can be used for streaming applications, and for such applications a codec designed to stream data effectively will usually be chosen.[16]

Latency results from the methods used to encode and decode the data. Some codecs will analyze a longer segment of the data to optimize efficiency, and then code it in a manner that requires a larger segment of data at one time to decode. (Often codecs create segments called a "frame" to create discrete data segments for encoding and decoding.) The inherent latency of the coding algorithm can be critical; for example, when there is a two-way transmission of data, such as with a telephone conversation, significant delays may seriously degrade the perceived quality.

In contrast to the speed of compression, which is proportional to the number of operations required by the algorithm, here latency refers to the number of samples that must be analysed before a block of audio is processed. In the minimum case, latency is zero samples (e.g., if the coder/decoder simply reduces the number of bits used to quantize the signal). Time domain algorithms such as LPC also often have low latencies, hence their popularity in speech coding for telephony. In algorithms such as MP3, however, a large number of samples have to be analyzed to implement a psychoacoustic model in the frequency domain, and latency is on the order of 23 ms (46 ms for two-way communication)).

Speech encoding [edit]

Speech encoding is an important category of audio data compression. The perceptual models used to estimate what a human ear can hear are generally somewhat different from those used for music. The range of frequencies needed to convey the sounds of a human voice are normally far narrower than that needed for music, and the sound is normally less complex. As a result, speech can be encoded at high quality using a relatively low bit rate.

If the data to be compressed is analog (such as a voltage that varies with time), quantization is employed to digitize it into numbers (normally integers). This is referred to as analog-to-digital (A/D) conversion. If the integers generated by quantization are 8 bits each, then the entire range of the analog signal is divided into 256 intervals and all the signal values within an interval are quantized to the same number. If 16-bit integers are generated, then the range of the analog signal is divided into 65,536 intervals.

This relation illustrates the compromise between high resolution (a large number of analog intervals) and high compression (small integers generated). This application of quantization is used by several speech compression methods. This is accomplished, in general, by some combination of two approaches:

- Only encoding sounds that could be made by a single human voice.
- Throwing away more of the data in the signal—keeping just enough to reconstruct an "intelligible" voice rather than the full frequency range of human hearing.

Perhaps the earliest algorithms used in speech encoding (and audio data compression in general) were the A-law algorithm and the μ-law algorithm.

History [edit]

A literature compendium for a large variety of audio coding systems was published in the IEEE Journal on Selected Areas in Communications (JSAC), February 1988. While there were some papers from before that time, this collection documented an entire variety of finished, working audio coders, nearly all of them using perceptual (i.e. masking) techniques and some kind of frequency analysis and back-end noiseless coding.[18] Several of these papers remarked on the difficulty of obtaining good, clean digital audio for research purposes. Most, if not all, of the authors in the JSAC edition were also active in the MPEG-1 Audio committee.

The world's first commercial broadcast automation audio compression system was developed by Oscar Bonello, an engineering professor at the University of Buenos Aires.[19] In 1983, using the psychoacoustic principle of the masking of critical bands first published in 1967,[20] he started developing a practical application based on the recently developed IBM PC computer, and the broadcast automation system was launched in 1987 under the name Audicom. Twenty years later, almost all the radio stations in the world were using similar technology manufactured by a number of companies.

Solidyne 922: The world's first commercial audio bit compression card for PC, 1990

Video [edit]

See also: Video coding format and Video codec

Video compression uses modern coding techniques to reduce redundancy in video data. Most video compression algorithms and codecs combine spatial image compression and temporal motion compensation. Video compression is a practical implementation of source coding in information theory. In practice, most video codecs also use audio compression techniques in parallel to compress the separate, but combined data streams as one package.[21]

The majority of video compression algorithms use lossy compression. Uncompressed video requires a very high data rate. Although lossless video compression codecs perform an average compression factor of over 3, a typical MPEG-4 lossy compression video has a compression factor between 20 and 200.[22] As in all lossy compression, there is a trade-off between video quality, cost of processing the compression and decompression, and system requirements. Highly compressed video may present visible or distracting artifacts.

Some video compression schemes typically operate on square-shaped groups of neighboring pixels, often called macroblocks. These pixel groups or blocks of pixels are compared from one frame to the next, and the video compression codec sends only the differences within those blocks. In areas of video with more motion, the compression must encode more data to keep up with the larger number of pixels that are changing. Commonly during explosions, flames, flocks of animals, and in some panning shots, the high-frequency detail leads to quality decreases or to increases in the variable bitrate.

Encoding theory [edit]

Video data may be represented as a series of still image frames. The sequence of frames contains spatial and temporal redundancy that video compression algorithms attempt to eliminate or code in a smaller size. Similarities can be encoded by only storing differences between frames, or by using perceptual features of human vision. For example, small differences in color are more difficult to perceive than are changes in brightness. Compression algorithms can average a color across these similar areas to reduce space, in a manner similar to those used in JPEG image compression.[23] Some of these methods are inherently lossy while others may preserve all relevant information from the original, uncompressed video.

One of the most powerful techniques for compressing video is interframe compression. Interframe compression uses one or more earlier or later frames in a sequence to compress the current frame, while intraframe compression uses only the current frame, effectively being image compression.[24]

The most powerful used method works by comparing each frame in the video with the previous one. If the frame contains areas where nothing has moved, the system simply issues a short command that copies that part of the previous frame, bit-for-bit, into the next one. If sections of the frame move in a simple manner, the compressor emits a (slightly longer) command that tells the decompressor to shift, rotate, lighten, or darken the copy. This longer command still remains much shorter than intraframe compression. Interframe compression works well for programs that will simply be played back by the viewer, but can cause problems if the video sequence needs to be edited.[25]

Because interframe compression copies data from one frame to another, if the original frame is simply cut out (or lost in transmission), the following frames cannot be reconstructed properly. Some video formats, such as DV, compress each frame independently using intraframe compression. Making 'cuts' in intraframe-compressed video is almost as easy as editing uncompressed video: one finds the beginning and ending of each frame, and simply copies bit-for-bit each frame that one wants to keep, and discards the frames one doesn't want. Another difference between intraframe and interframe compression is that, with intraframe systems, each frame uses a similar amount of data. In most interframe systems, certain frames (such as "I frames" in MPEG-2) aren't allowed to copy data from other frames, so they require much more data than other frames nearby.[16]

It is possible to build a computer-based video editor that spots problems caused when I frames are edited out while other frames need them. This has allowed newer formats like HDV to be used for editing. However, this process demands a lot more computing power than editing intraframe compressed video with the same picture quality.

Today, nearly all commonly used video compression methods (e.g., those in standards approved by the ITU-T or ISO) apply a discrete cosine transform (DCT) for spatial redundancy reduction. The DCT that is widely used in this regard was introduced by N. Ahmed, T. Natarajan and K. R. Rao in 1974.[26] Other methods, such as fractal compression, matching pursuit and the use of a discrete wavelet transform (DWT) have been the subject of some research, but are typically not used in practical products (except for the use of wavelet coding as still-image coders without motion compensation). Interest in fractal compression seems to be waning, due to recent theoretical analysis showing a comparative lack of effectiveness of such methods.[24]

Timeline [edit]

The following table is a partial history of international video compression standards.

History of Video Compression Standards

Year	Standard	Publisher	Popular Implementations
1984	H.120	ITU-T	
1988	H.261	ITU-T	Videoconferencing, Videotelephony
1993	MPEG-1 Part 2	ISO, IEC	Video-CD
1995	H.262/MPEG-2 Part 2	ISO, IEC, ITU-T	DVD Video, Blu-ray, Digital Video Broadcasting, SVCD
1996	H.263	ITU-T	Videoconferencing, Videotelephony, Video on Mobile Phones (3GP)
1999	MPEG-4 Part 2	ISO, IEC	Video on Internet (DivX, Xvid ...)
2003	H.264/MPEG-4 AVC	Sony, Panasonic, Samsung, ISO, IEC, ITU-T	Blu-ray, HD DVD, Digital Video Broadcasting, iPod Video, Apple TV, videoconferencing
2009	VC-2 (Dirac)	SMPTE	Video on Internet, HDTV broadcast, UHDTV
2013	H.265	ISO, IEC, ITU-T	

Genetics [edit]

See also: Compression of Genomic Re-Sequencing Data

Genetics compression algorithms are the latest generation of lossless algorithms that compress data (typically sequences of nucleotides) using both conventional compression algorithms and genetic algorithms adapted to the specific datatype. In 2012, a team of scientists from Johns Hopkins University published a genetic compression algorithm that does not use a reference genome for compression. HAPZIPPER was tailored for HapMap data and achieves over 20-fold compression (95% reduction in file size), providing 2- to 4-fold better compression and in much faster time than the leading general-purpose compression utilities. For this, Chanda, Elhaik, and Bader introduced MAF based encoding (MAFE), which reduces the heterogeneity of the dataset by sorting SNPs by their minor allele frequency, thus homogenizing the dataset.[27] Other algorithms in 2009 and 2013 (DNAZip and GenomeZip) have compression ratios of up to 1200-fold—allowing 6 billion basepair diploid human genomes to be stored in 2.5 megabytes (relative to a reference genome or averaged over many genomes).[28][29]

Outlook and currently unused potential [edit]

It is estimated that the total amount of data that is stored on the world's storage devices could be further compressed with existing compression algorithms by a remaining average factor of 4.5:1. It is estimated that the combined technological capacity of the world to store information provides 1,300 exabytes of hardware digits in 2007, but when the corresponding content is optimally compressed, this only represents 295 exabytes of Shannon information.[30]
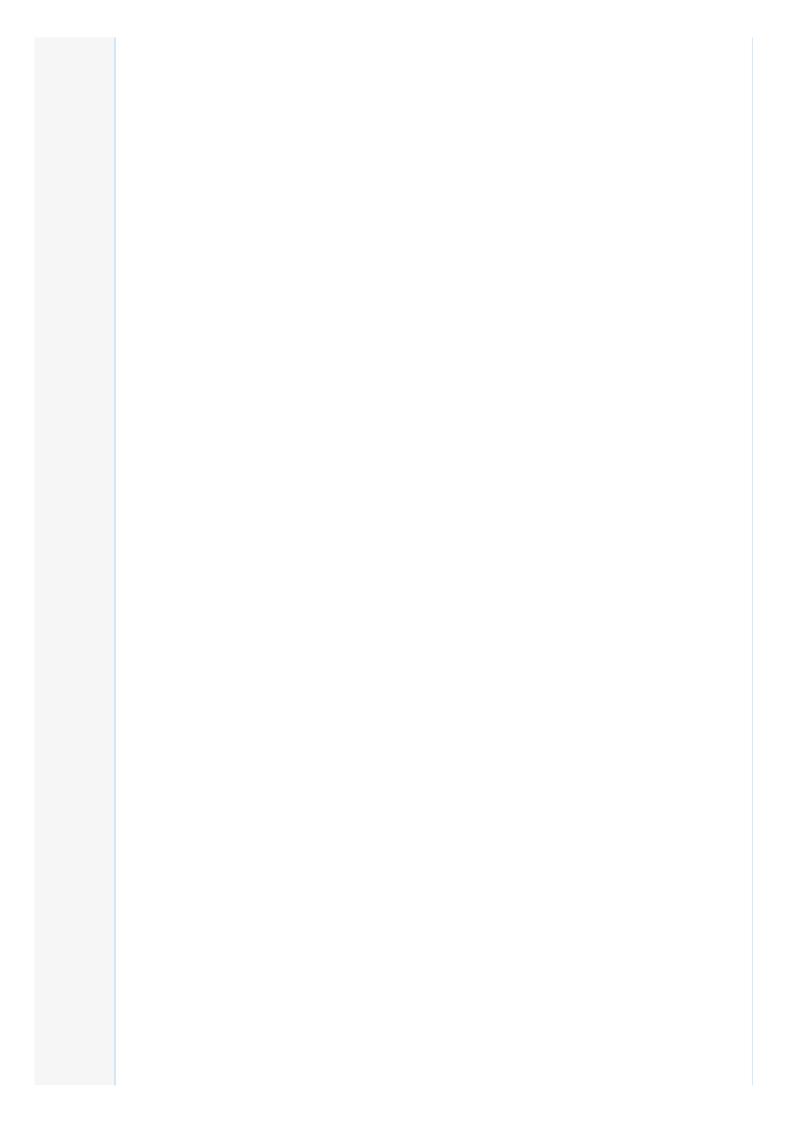
See also [edit]

> This "see also" section **may contain an excessive number of suggestions**. Please ensure that only the most relevant suggestions are given and that they are not red links, and consider integrating suggestions into the article itself. *(November 2011)*

- Auditory masking
- Calgary corpus
- Canterbury corpus
- Context mixing
- Compression artifact
- Data compression ratio
- Data compression symmetry
- Dictionary coder
- Distributed source coding
- Dyadic distribution
- Dynamic Markov compression
- Elias gamma coding
- Entropy encoding
- Fibonacci coding
- Golomb coding
- HTTP compression
- Kolmogorov complexity
- Magic compression algorithm
- Minimum description length
- Modulo-N code
- Range encoding
- Sub-band coding

- Universal code (data compression)

References [edit]

1. ^ Wade, Graham (1994). *Signal coding and processing* ⧉ (2 ed.). Cambridge University Press. p. 34. ISBN 978-0-521-42336-6. Retrieved 2011-12-22. "The broad objective of source coding is to exploit or remove 'inefficient' redundancy in the PCM source and thereby achieve a reduction in the overall source rate R."
2. ^ a b Mahdi, O.A.; Mohammed, M.A.; Mohamed, A.J. (November 2012). "Implementing a Novel Approach an Convert Audio Compression to Text Coding via Hybrid Technique" 📄 (PDF). *International Journal of Computer Science Issues* **9** (6, No. 3): 53–59. Retrieved 6 March 2013.
3. ^ Pujar, J.H.; Kadlaskar, L.M. (May 2010). "A New Lossless Method of Image Compression and Decompression Using Huffman Coding Techniques" 📄 (PDF). *Journal of Theoretical and Applied Information Technology* **15** (1): 18–23.
4. ^ Salomon, David (2008). *A Concise Introduction to Data Compression*. Berlin: Springer. ISBN 9781848000728.
5. ^ S. Mittal and J. Vetter, "A Survey Of Architectural Approaches for Data Compression in Cache and Main Memory Systems ⧉", IEEE Transactions on Parallel and Distributed Systems, 2015.
6. ^ Tank, M.K. (2011). *Implementation of Limpel-Ziv algorithm for lossless compression using VHDL. Thinkquest 2010: Proceedings of the First International Conference on Contours of Computing Technology*. Berlin: Springer. pp. 275–283.
7. ^ Navqi, Saud; Naqvi, R.; Riaz, R.A.; Siddiqui, F. (April 2011). "Optimized RTL design and implementation of LZW algorithm for high bandwidth applications" 📄 (PDF). *Electrical Review* **2011** (4): 279–285.
8. ^ a b Mahmud, Salauddin (March 2012). "An Improved Data Compression Method for General Data" 📄 (PDF). *International Journal of Scientific & Engineering Research* **3** (3): 2. Retrieved 6 March 2013.
9. ^ Arcangel, Cory. "On Compression" 📄 (PDF). Retrieved 6 March 2013.
10. ^ Marak, Laszlo. "On image compression" 📄 (PDF). University of Marne la Vallee. Retrieved 6 March 2013.
11. ^ Mahoney, Matt. "Rationale for a Large Text Compression Benchmark" ⧉. *http://cs.fit.edu/~mmahoney/* ⧉. Florida Institute of Technology. Retrieved 5 March 2013.
12. ^ Korn, D. et al. "RFC 3284: The VCDIFF Generic Differencing and Compression Data Format" ⧉. Internet Engineering Task Force. Retrieved 5 March 2013.
13. ^ Korn, D.G.; Vo, K.P. (1995), B. Krishnamurthy, ed., *Vdelta: Differencing and Compression*, Practical Reusable Unix Software, New York: John Wiley & Sons, Inc.
14. ^ The Olympus WS-120 digital speech recorder, according to its manual, can store about 178 hours of speech-quality audio in .WMA format in 500MB of flash memory.
15. ^ Coalson, Josh. "FLAC Comparison" ⧉. Retrieved 6 March 2013.
16. ^ a b c d Jaiswal, R.C. (2009). *Audio-Video Engineering*. Pune, Maharashtra: Nirali Prakashan. p. 3.41. ISBN 9788190639675.
17. ^ Faxin Yu, Hao Luo, Zheming Lu (2010). *Three-Dimensional Model Analysis and Processing*. Berlin: Springer. p. 47. ISBN 9783642126512.
18. ^ "File Compression Possibilities" ⧉. *A Brief guide to compress a file in 6 different ways.*
19. ^ "Summary of some of Solidyne's contributions to Broadcast Engineering" ⧉. *Brief History of Solidyne*. Buenos Aires: Solidyne. Retrieved 6 March 2013.
20. ^ Zwicker, Eberhard et al. (Originally published in 1967; Translation published in 1999). *The Ear As A Communication Receiver* ⧉. Melville, NY: Acoustical Society of America. Check date values in: |date= (help)
21. ^ "Video Coding" ⧉. *Center for Signal and Information Processing Research*. Georgia Institute of Technology. Retrieved 6 March 2013.
22. ^ Graphics & Media Lab Video Group (2007). *Lossless Video Codecs Comparison* 📄 (PDF). Moscow State University.
23. ^ Lane, Tom. "JPEG Image Compression FAQ, Part 1" ⧉. *Internet FAQ Archives*. Independent JPEG Group. Retrieved 6 March 2013.
24. ^ a b Faxin Yu, Hao Luo, Zheming Lu (2010). *Three-Dimensional Model Analysis and Processing*. Berlin: Springer. p. 47. ISBN 9783642126512.
25. ^ Bhojani, D.R. "4.1 Video Compression" 📄 (PDF). *Hypothesis*. Retrieved 6 March 2013.
26. ^ Ahmed, N.; Natarajan, T.; Rao, K.R. (January 1974). "Discrete Cosine Transform". *IEEE Transactions on Computers* **C–23** (1): 90–93. doi:10.1109/T-C.1974.223784 ⧉.
27. ^ Chanda P, Bader JS, Elhaik E; Elhaik; Bader (27 Jul 2012). "HapZipper: sharing HapMap populations just got easier" 📄 (PDF). *Nucleic Acids Research* **40** (20): e159. doi:10.1093/nar/gks709 ⧉. PMC 3488212 ⧉. PMID 22844100 ⧉.
28. ^ Christley S, Lu Y, Li C, Xie X; Lu; Li; Xie (Jan 15, 2009). "Human genomes as email attachments". *Bioinformatics* **25** (2): 274–5. doi:10.1093/bioinformatics/btn582 ⧉. PMID 18996942 ⧉.
29. ^ Pavlichin DS, Weissman T, Yona G; Weissman; Yona (September 2013). "The human genome contracts again". *Bioinformatics* **29** (17): 2199–202. doi:10.1093/bioinformatics/btt362 ⧉. PMID 23793748 ⧉.
30. ^ Hilbert, Martin; López, Priscila (1 April 2011). "The World's Technological Capacity to Store, Communicate, and Compute Information" ⧉. *Science* **332** (6025): 60–65. Bibcode:2011Sci...332...60H ⧉. doi:10.1126/science.1200970 ⧉. PMID 21310967 ⧉. Retrieved 6 March 2013.

External links [edit]

- Data Compression Basics (Video) ⧉
- Video compression 4:2:2 10-bit and its benefits ⧉
- Why does 10-bit save bandwidth (even when content is 8-bit)? ⧉
- Which compression technology should be used ⧉
- Wiley - Introduction to Compression Theory 📄
- EBU subjective listening tests on low-bitrate audio codecs 📄
- Audio Archiving Guide: Music Formats ⧉ (Guide for helping a user pick out the right codec)
- MPEG 1&2 video compression intro (pdf format) 📄 at the Wayback Machine (archived September 28, 2007)
- hydrogenaudio wiki comparison ⧉
- Introduction to Data Compression 📄 by Guy E Blelloch from CMU
- HD Greetings - 1080p Uncompressed source material for compression testing and research ⧉
- Explanation of lossless signal compression method used by most codecs ⧉
- Interactive blind listening tests of audio codecs over the internet ⧉
- TestVid - 2,000+ HD and other uncompressed source video clips for compression testing ⧉
- Videsignline - Intro to Video Compression ⧉
- Data Footprint Reduction Technology 📄
- What is Run length Coding in video compression. ⧉

v · t · e		Data compression methods
Lossless	Entropy type	Unary · Arithmetic · Golomb · Huffman (Adaptive · Canonical · Modified) · Range · Shannon · Shannon–Fano · Shannon–Fano–Elias · Tunstall · Universal (Exp-Golomb · Fibonacci · Gamma · Levenshtein)
	Dictionary type	Byte pair encoding · DEFLATE · Lempel–Ziv (LZ77 / LZ78 (LZ1 / LZ2) · LZJB · LZMA · LZO · LZRW · LZS · LZSS · LZW · LZWL · LZX · LZ4 · Statistical)
	Other types	BWT · CTW · Delta · DMC · MTF · PAQ · PPM · RLE
Audio	Concepts	Bit rate (average (ABR) · constant (CBR) · variable (VBR)) · Companding · Convolution · Dynamic range · Latency · Nyquist–Shannon theorem · Sampling · Sound quality · Speech coding · Sub-band coding
	Codec parts	A-law · μ-law · ACELP · ADPCM · CELP · DPCM · Fourier transform · LPC (LAR · LSP) · MDCT · Psychoacoustic model · WLPC
Image	Concepts	Chroma subsampling · Coding tree unit · Color space · Compression artifact · Image resolution · Macroblock · Pixel · PSNR · Quantization · Standard test image
	Methods	Chain code · DCT · EZW · Fractal · KLT · LP · RLE · SPIHT · Wavelet
Video	Concepts	Bit rate (average (ABR) · constant (CBR) · variable (VBR)) · Display resolution · Frame · Frame rate · Frame types · Interlace · Video characteristics · Video quality
	Codec parts	Lapped transform · DCT · Deblocking filter · Motion compensation
Theory		Entropy · Kolmogorov complexity · Lossy · Quantization · Rate–distortion · Redundancy · Timeline of information theory

⊙ Compression formats · ⊙ Compression software (codecs)

v · t · e		Multimedia compression and container formats
Video compression	ISO/IEC	MJPEG · Motion JPEG 2000 · MPEG-1 · MPEG-2 (Part 2) · MPEG-4 (Part 2/ASP · Part 10/AVC) · MPEG-H (Part 2/HEVC)
	ITU-T	H.120 · H.261 · H.262 · H.263 · H.264 · H.265
	Others	Apple Video · AVS · Bink · CineForm · Cinepak · Daala · Dirac · DV · DVI · FFV1 · Huffyuv · Indeo · Microsoft Video 1 · MSU Lossless · Lagarith · OMS Video · Pixlet · ProRes 422 · ProRes 4444 · QuickTime (Animation · Graphics) · RealVideo · RTVideo · SheerVideo · Smacker · Sorenson Video, Spark · Theora · VC-1 · VC-2 · VC-3 · VP3 · VP6 · VP7 · VP8 · VP9 · WMV · XEB · YULS
Audio compression	ISO/IEC	MPEG-1 Layer III (MP3) · MPEG-1 Layer II (Multichannel) · MPEG-1 Layer I · AAC (HE-AAC · AAC-LD) · MPEG Surround · MPEG-4 ALS · MPEG-4 SLS · MPEG-4 DST · MPEG-4 HVXC · MPEG-4 CELP · MPEG-D USAC · MPEG-H 3D Audio
	ITU-T	G.711 · G.718 · G.719 · G.722 · G.722.1 · G.722.2 · G.723 · G.723.1 · G.726 · G.728 · G.729 · G.729.1
	Others	ACELP · AC-3 · AMR · AMR-WB · AMR-WB+ · ALAC · Asao · ATRAC · CELT · Codec2 · DRA · DTS · EVRC · EVRC-B · FLAC · GSM-HR · GSM-FR · GSM-EFR · iLBC · iSAC · Monkey's Audio · TTA (True Audio) · MT9 · A-law · μ-law · Musepack · OptimFROG · Opus · OSQ · QCELP · RCELP · RealAudio · RTAudio · SD2 · SHN · SILK · Siren · SMV · Speex · SVOPC · TwinVQ · VMR-WB · Vorbis · VSELP · WavPack · WMA
	IEC, ISO,	CCITT Group 4 · JPEG · JPEG 2000 · JPEG XR · Lossless JPEG · JBIG · JBIG2 · PNG · TIFF/EP · TIFF/IT · HEVC

Image compression	ITU-T		
	Others	APNG · BPG · DjVu · EXR · GIF · ICER · MNG · PGF · QTVR · TIFF · WBMP · WebP	
Containers	ISO/IEC	MPEG-PS · MPEG-TS · ISO base media file format · MPEG-4 Part 14 (MP4) · Motion JPEG 2000 · MPEG-21 Part 9 · MPEG media transport	
	ITU-T	H.222.0 · T.802	
	Others	3GP and 3G2 · AMV · ASF · AIFF · AVI · AU · BPG · Bink (Smacker) · BMP · DivX Media Format · EVO · Flash Video · GXF · IFF · M2TS · Matroska (WebM) · MXF · Ogg · QuickTime File Format · RatDVD · RealMedia · RIFF (WAV) · MOD and TOD · VOB, IFO and BUP	
See Compression methods for methods and Compression software for codecs			

v · t · e	Data compression software		
Archivers with compression (comparison)	Free software	7-Zip · Ark · Expander · File Roller · FreeArc · Info-ZIP · KGB Archiver · PAQ · PeaZip · The Unarchiver (decompression only) · tar · Xarchiver · Zipeg · ZPAQ	
	Freeware	Filzip · LHA · StuffIt Expander (decompression only) · TUGZip · ZipGenius	
	Commercial	ARC · ALZip · Archive Utility · ARJ · BetterZip · BulkZip · JAR · MacBinary · PKZIP/SecureZIP · PowerArchiver · StuffIt · WinAce · WinRAR · WinZip	
Non-archiving compressors	Generic	bzip2 · compress · gzip · lzip · lzop · pack · rzip · Snappy · XZ Utils	
	For code	UPX	
Audio compression (comparison)	Lossy	Fraunhofer FDK AAC · Nero AAC Codec · Freeware Advanced Audio Coder (FAAC) · Helix DNA Producer · I3enc · LAME · TooLAME · libavcodec · libcelt · libopus · libspeex · Musepack · libvorbis · Windows Media Encoder	
	Lossless	ALAC · FLAC · libavcodec · Monkey's Audio · mp4als · OptimFROG · Shorten · TTA (True Audio) · WavPack	
Video compression (comparison)	Lossy	MPEG-4 ASP	3ivx · DivX · Nero Digital · FFmpeg · HDX4 · Xvid
		H.264 / MPEG-4 AVC	CoreAVC · Blu-code · DivX · FFmpeg · Nero Digital · OpenH264 · QuickTime · x264
		HEVC	DivX · x265
		Others	CineForm · Cinepak · Daala · DNxHD · Helix DNA Producer · Indeo · libavcodec · Schrödinger (Dirac) · SBC · Sorenson · VP7 · libtheora · libvpx · Windows Media Encoder
	Lossless	FFV1 · Huffyuv · Lagarith · MSU Lossless · YULS	
See also: compression methods and compression formats			

Authority control	NDL: 00942229

Categories: Audio engineering | Computer storage | Data compression | Digital audio | Digital television | Film and video technology | Video compression | Videotelephony | Utility software types