KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

„Image Processing and Computer Graphics”

Prezentacja multimedialna współfinansowana przez Unię Europejską w ramach Europejskiego Funduszu Społecznego w projekcie pt.

„Innowacyjna dydaktyka bez ograniczeń - zintegrowany rozwój Politechniki Łódzkiej - zarządzanie Uczelnią, nowoczesna oferta edukacyjna i wzmacniania zdolności do zatrudniania osób niepełnosprawnych”

Politechnika Łódzka

Politechnika Łódzka, ul. Żeromskiego 116, 90-924 Łódź, tel. (042) 631 28 83
www.kapitalludzki.p.lodz.pl

Contribution

Lecturer:

Piotr M. Szczypiński, Dr inż.
Institute of Electronics, Technical University of Lodz
Wolczanska 211/215, 90-924 Lodz, Poland
tel. +48426312638
http://www.eletel.p.lodz.pl/pms
pms@p.lodz.pl

Fragments of the lecture were also provided by

Andrzej Materka, Michał Strzelecki, Paweł Strumiłło, Piotr Romaniuk, Piotr Makowski

and were used with permission.

Image & video data compression

- Image compression motivation
- Lossless vs. lossy compression
  (Information redundancy & Psychovisual redundancy)
- Lossless compression – selected methods
  - Run-length encoding
  - Huffman coding
  - Delta encoding
  - Deflate & LZW
- Lossy compression – selected methods (*Quid pro quo*)
  - Transform methods (Fourier, Cosine, Wavelet transforms)
  - Subsampling (chroma subsampling)
  - Posterization and paletted images
- Image compression applications: JPEG, JPEG 2000
- Video compression applications: MPEG1, MPEG2, MPEG4
- Compression in medical imaging

Need for image compression

Standard definition TV PAL image:
576 lines x 720 rows x 3 B = 1.2 MB

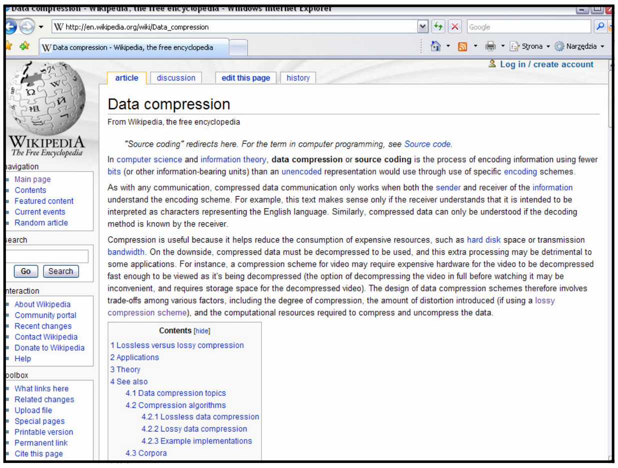
Frames in 2-hour video:
25 fps x 7200 s = 180 000 frames

Capacity of video file:
180 000 frames x 1.2 MB = 210 GB

DVD:
4.7 GB

Compress data 44x !

KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY



Data compression

From Wikipedia, the free encyclopedia

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

In computer science and information theory, **data compression** or **source coding** is the process of encoding information using fewer bits (or other information-bearing units) than an unencoded representation would use through use of specific encoding schemes.

As with any communication, compressed data communication only works when both the sender and receiver of the information understand the encoding scheme. For example, this text makes sense only if the receiver understands that it is intended to be interpreted as characters representing the English language. Similarly, compressed data can only be understood if the decoding method is known by the receiver.

Compression is useful because it helps reduce the consumption of expensive resources, such as hard disk space or transmission bandwidth. On the downside, compressed data must be decompressed to be used, and this extra processing may be detrimental to some applications. For instance, a compression scheme for video may require expensive hardware for the video to be decompressed fast enough to be viewed as it's being decompressed (the option of decompressing the video in full before watching it may be inconvenient, and requires storage space for the decompressed video). The design of data compression schemes therefore involves trade-offs among various factors, including the degree of compression, the amount of distortion introduced (if using a lossy compression scheme), and the computational resources required to compress and uncompress the data.

Contents [hide]

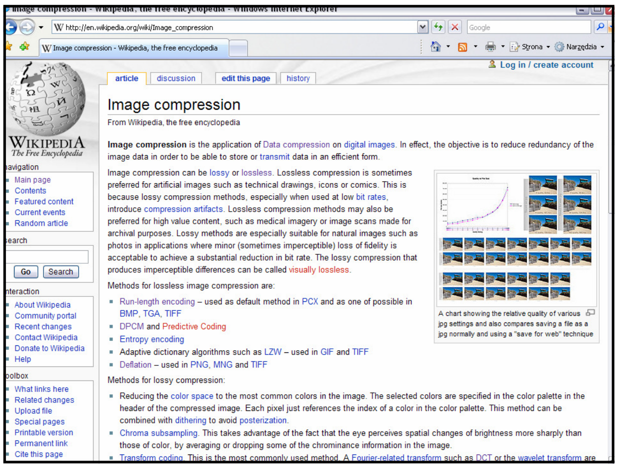


Image compression

From Wikipedia, the free encyclopedia

Image compression is the application of Data compression on digital images. In effect, the objective is to reduce redundancy of the image data in order to be able to store or transmit data in an efficient form.

Image compression can be lossy or lossless. Lossless compression is sometimes preferred for artificial images such as technical drawings, icons or comics. This is because lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossless compression methods may also be preferred for high value content, such as medical imagery or image scans made for archival purposes. Lossy methods are especially suitable for natural images such as photos in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces imperceptible differences can be called visually lossless.

A chart showing the relative quality of various jpg settings and also compares saving a file as a jpg normally and using a "save for web" technique

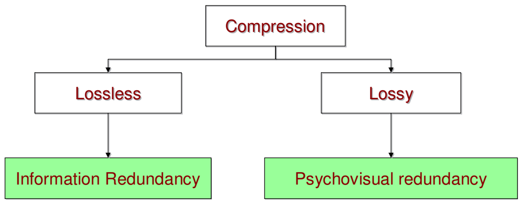
Methods for lossless image compression are:

- Run-length encoding – used as default method in PCX and as one of possible in BMP, TGA, TIFF
- DPCM and Predictive Coding
- Entropy encoding
- Adaptive dictionary algorithms such as LZW – used in GIF and TIFF
- Deflation – used in PNG, MNG and TIFF

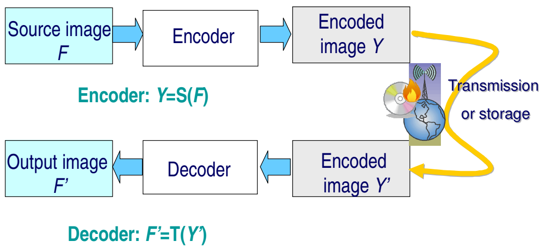
Methods for lossy compression:

- Reducing the color space to the most common colors in the image. The selected colors are specified in the color palette in the header of the compressed image. Each pixel just references the index of a color in the color palette. This method can be combined with dithering to avoid posterization.
- Chroma subsampling. This takes advantage of the fact that the eye perceives spatial changes of brightness more sharply than those of color, by averaging or dropping some of the chrominance information in the image.
- Transform coding. This is the most commonly used method. A Fourier-related transform such as DCT or the wavelet transform are

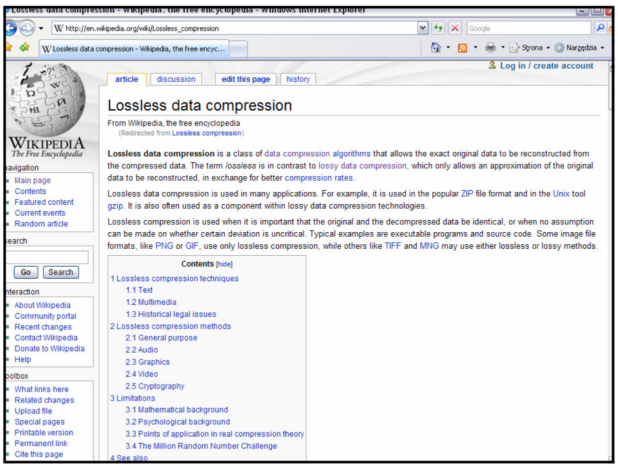
Lossless and Lossy compression



Lossless and Lossy compression



If $F' = F$, $(T = S^{-1})$ then lossless compression
If $F' \neq F$, $(T \neq S^{-1})$ then lossy compression



Lossless data compression - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/Lossless_compression

Log in / create account

article | discussion | edit this page | history

Lossless data compression

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

Lossless data compression is a class of data compression algorithms that allows the exact original data to be reconstructed from the compressed data. The term *lossless* is in contrast to lossy data compression, which only allows an approximation of the original data to be reconstructed, in exchange for better compression rates.

Lossless data compression is used in many applications. For example, it is used in the popular ZIP file format and in the Unix tool gzip. It is also often used as a component within lossy data compression technologies.

Lossless compression is used when it is important that the original and the decompressed data be identical, or when no assumption can be made on whether certain deviation is uncritical. Typical examples are executable programs and source code. Some image file formats, like PNG or GIF, use only lossless compression, while others like TIFF and MNG may use either lossless or lossy methods.

Contents [hide]

Source entropy

Source ***entropy*** :

$$E_F = -\sum_{i=0}^{L-1} p(l_i)\log_2 p(l_i)$$

represents the average information per source output.

For data source given in the form of an image, $p(l_i)$ – is the probability of occurrence of gray level l_i.

Entropy can be interpreted as the average number of bits required for coding a single image pixel.

E = 7.01

E = 5.31

KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA

Entropy (information theory) - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/Entropy_(Information_theory)

Entropy (information theory)

From Wikipedia, the free encyclopedia
(Redirected from Entropy (Information theory))

This article is about the Shannon entropy in information theory. For other uses, see Entropy (disambiguation).

In information theory, **entropy** is a measure of the uncertainty associated with a random variable. The term by itself in this context usually refers to the **Shannon entropy**, which quantifies, in the sense of an expected value, the information contained in a message, usually in units such as bits. Equivalently, the Shannon entropy is a measure of the average information content one is missing when one does not know the value of the random variable. The concept was introduced by Claude E. Shannon in his 1948 paper "A Mathematical Theory of Communication".

Shannon's entropy represents an absolute limit on the best possible lossless compression of any communication, under certain constraints: treating messages to be encoded as a sequence of independent and identically-distributed random variables, Shannon's source coding theorem shows that, in the limit, the average length of the shortest possible representation to encode the messages in a given alphabet is their entropy divided by the logarithm of the number of symbols in the target alphabet.

A fair coin has an entropy of one bit. However, if the coin is not fair, then the uncertainty is lower (if asked to bet on the next outcome, we would bet preferentially on the most frequent result), and thus the Shannon entropy is lower. Mathematically, a coin flip is an example of a Bernoulli trial, and its entropy is given by the binary entropy function. A long string of repeating characters has an entropy rate of 0, since every character is predictable. The entropy rate of English text is between 1.0 and 1.5 bits per letter,[1] or as low as 0.6 to 1.3 bits per letter, according to estimates by Shannon based on human experiments.[2]

Contents [hide]

Entropy (information theory) - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/Entropy_(Information_theory)

Definition [edit]

The entropy H of a discrete random variable X with possible values $\{x_1, ..., x_n\}$ is

$$H(X) = \mathrm{E}(I(X)).$$

Here E is the expected value function, and $I(X)$ is the information content or self-information of X.

$I(X)$ is itself a random variable. If p denotes the probability mass function of X then the entropy can explicitly be written as

$$H(X) = \sum_{i=1}^{n} p(x_i)\, I(x_i) = -\sum_{i=1}^{n} p(x_i)\log_b p(x_i),$$

where b is the base of the logarithm used. Common values of b are 2, Euler's number e, and 10, and the unit of entropy is bit for $b = 2$, nat for $b = e$, and dit (or digit) for $b = 10$.[3]

In the case of $p_i = 0$ for some i, the value of the corresponding summand $0 \log_b 0$ is taken to be 0, which is consistent with the limit

$$\lim_{p\to 0+} p\log p = 0.$$

Characterization [edit]

Shannon entropy is characterized by a small number of criteria, listed below. Any definition of entropy satisfying these assumptions has the form

$$-K\sum_{i=1}^{n} p_i \log p_i$$

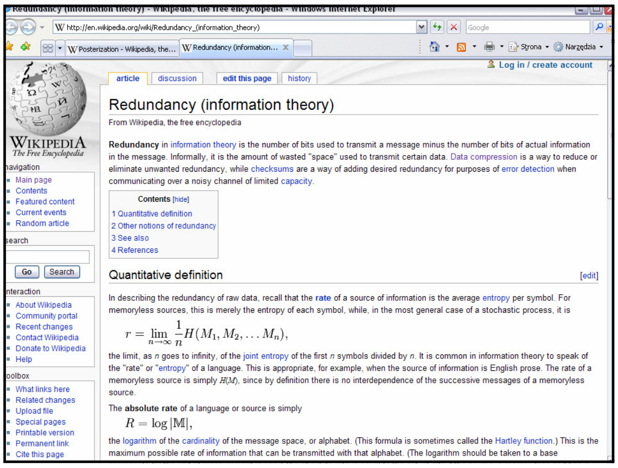
where K is a constant corresponding to a choice of measurement units.

In the following, $p_i = \Pr(X = x_i)$ and $H_n(p_1, \ldots, p_n) = H(X)$.

Continuity [edit]

The measure should be continuous, so that changing the values of the probabilities by a very small amount should only change the entropy by a small amount.

Symmetry [edit]



Redundancy (information theory)

From Wikipedia, the free encyclopedia

Redundancy in information theory is the number of bits used to transmit a message minus the number of bits of actual information in the message. Informally, it is the amount of wasted "space" used to transmit certain data. Data compression is a way to reduce or eliminate unwanted redundancy, while checksums are a way of adding desired redundancy for purposes of error detection when communicating over a noisy channel of limited capacity.

Contents [hide]

Quantitative definition [edit]

In describing the redundancy of raw data, recall that the **rate** of a source of information is the average entropy per symbol. For memoryless sources, this is merely the entropy of each symbol, while, in the most general case of a stochastic process, it is

$$r = \lim_{n \to \infty} \frac{1}{n} H(M_1, M_2, \dots M_n),$$

the limit, as n goes to infinity, of the joint entropy of the first n symbols divided by n. It is common in information theory to speak of the "rate" or "entropy" of a language. This is appropriate, for example, when the source of information is English prose. The rate of a memoryless source is simply $H(M)$, since by definition there is no interdependence of the successive messages of a memoryless source.

The **absolute rate** of a language or source is simply

$$R = \log |\mathbb{M}|,$$

the logarithm of the cardinality of the message space, or alphabet. (This formula is sometimes called the Hartley function.) This is the maximum possible rate of information that can be transmitted with that alphabet. (The logarithm should be taken to a base

Redundancy

Let n_1, n_2 denote data units carrying the same information.

Relative data redundancy is given by:

$$R_D = 1 - 1/C_R$$

where C_R – is termed the ***compression ratio:***

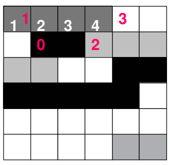
$$C_R = n_1 / n_2$$

for $n_1 = n_2$: $C_R = 1$; $R = 0$
for $n_1 >> n_2$: $C_R \rightarrow \infty$; $R \rightarrow 1$
for $n_1 << n_2$: $C_R \rightarrow 0$; $R \rightarrow -\infty$

Interpixel redundancy removal

Gray levels of adjacent pixels are strongly correlated.

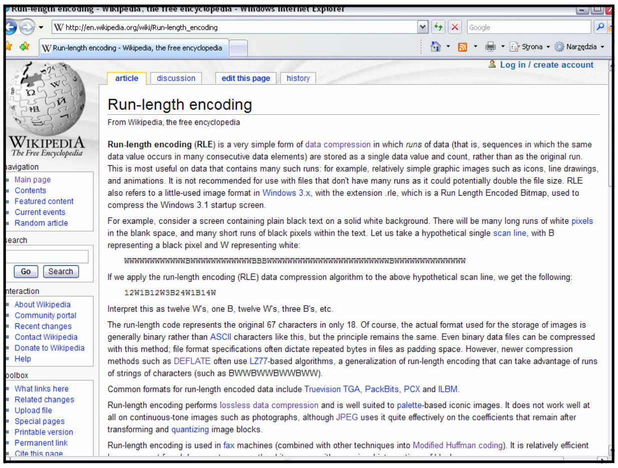
Run length encoding (RLE) – an image is coded by symbol pairs (g_i, l_i) where g_i denotes i-th gray level and l_i is the run length



RL sequences: (1,4); (3,3); (0,2); (2,5); (3,2); (0,7); (3,11); (1,2)
8*3=24 bytes are neded for image coding
g – coded by 1 byte,
l – coded by 2 bytes

6x6=36 bytes

C_R = 36/24 = 1.5



Run-length encoding

From Wikipedia, the free encyclopedia

Run-length encoding (**RLE**) is a very simple form of data compression in which *runs* of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run. This is most useful on data that contains many such runs: for example, relatively simple graphic images such as icons, line drawings, and animations. It is not recommended for use with files that don't have many runs as it could potentially double the file size. RLE also refers to a little-used image format in Windows 3.x, with the extension .rle, which is a Run Length Encoded Bitmap, used to compress the Windows 3.1 startup screen.

For example, consider a screen containing plain black text on a solid white background. There will be many long runs of white pixels in the blank space, and many short runs of black pixels within the text. Let us take a hypothetical single scan line, with B representing a black pixel and W representing white:

```
WWWWWWWWWWWWBWWWWWWWWWWWWBBBWWWWWWWWWWWWWWWWWWWWWWWWBWWWWWWWWWWWWWW
```

If we apply the run-length encoding (RLE) data compression algorithm to the above hypothetical scan line, we get the following:

```
12W1B12W3B24W1B14W
```

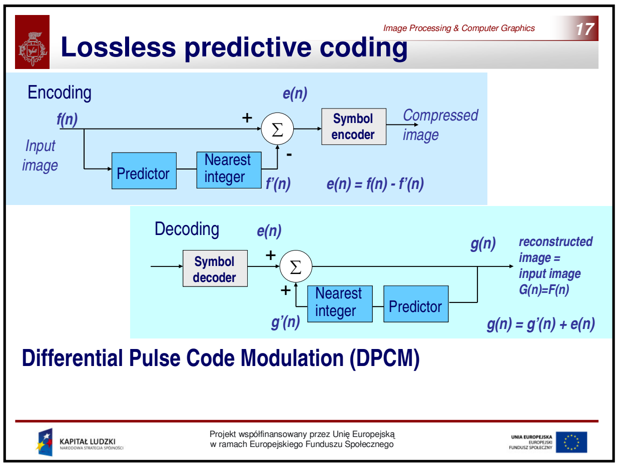
Interpret this as twelve W's, one B, twelve W's, three B's, etc.

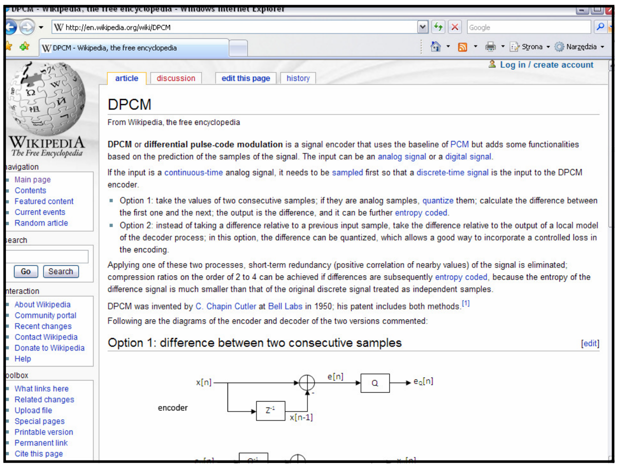
The run-length code represents the original 67 characters in only 18. Of course, the actual format used for the storage of images is generally binary rather than ASCII characters like this, but the principle remains the same. Even binary data files can be compressed with this method; file format specifications often dictate repeated bytes in files as padding space. However, newer compression methods such as DEFLATE often use LZ77-based algorithms, a generalization of run-length encoding that can take advantage of runs of strings of characters (such as BWWBWWBWWBWW).

Common formats for run-length encoded data include Truevision TGA, PackBits, PCX and ILBM.

Run-length encoding performs lossless data compression and is well suited to palette-based iconic images. It does not work well at all on continuous-tone images such as photographs, although JPEG uses it quite effectively on the coefficients that remain after transforming and quantizing image blocks.

Run-length encoding is used in fax machines (combined with other techniques into Modified Huffman coding). It is relatively efficient





DPCM

From Wikipedia, the free encyclopedia

DPCM or **differential pulse-code modulation** is a signal encoder that uses the baseline of PCM but adds some functionalities based on the prediction of the samples of the signal. The input can be an analog signal or a digital signal.

If the input is a continuous-time analog signal, it needs to be sampled first so that a discrete-time signal is the input to the DPCM encoder.

- Option 1: take the values of two consecutive samples; if they are analog samples, quantize them; calculate the difference between the first one and the next; the output is the difference, and it can be further entropy coded.
- Option 2: instead of taking a difference relative to a previous input sample, take the difference relative to the output of a local model of the decoder process; in this option, the difference can be quantized, which allows a good way to incorporate a controlled loss in the encoding.

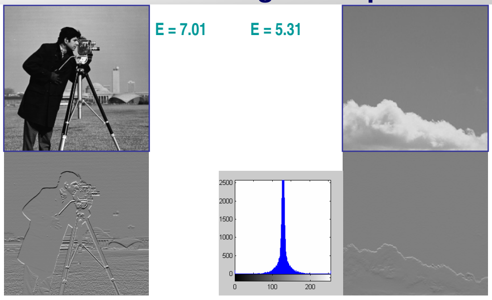
Applying one of these two processes, short-term redundancy (positive correlation of nearby values) of the signal is eliminated; compression ratios on the order of 2 to 4 can be achieved if differences are subsequently entropy coded, because the entropy of the difference signal is much smaller than that of the original discrete signal treated as independent samples.

DPCM was invented by C. Chapin Cutler at Bell Labs in 1950; his patent includes both methods.[1]

Following are the diagrams of the encoder and decoder of the two versions commented:

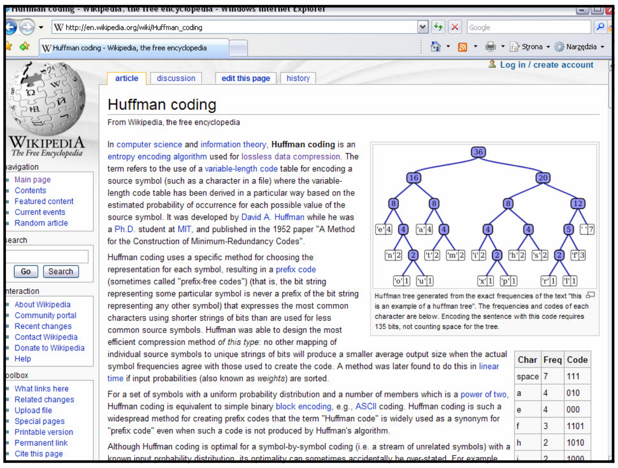
Option 1: difference between two consecutive samples

Predictive coding - example



Variable length coding (Example)

l_k	$p(l_k)$	code 1	$m^1(l_k)$	code 2	$m^2(l_k)$
0	0.19	000	3	11	2
1	0.25	001	3	01	2
2	0.21	010	3	10	2
3	0.16	011	3	001	3
4	0.08	100	3	0001	4
5	0.06	101	3	00001	5
6	0.03	110	3	000001	6
7	0.02	111	3	000000	6



Huffman coding

From Wikipedia, the free encyclopedia

In computer science and information theory, **Huffman coding** is an entropy encoding algorithm used for lossless data compression. The term refers to the use of a variable-length code table for encoding a source symbol (such as a character in a file) where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol. It was developed by David A. Huffman while he was a Ph.D. student at MIT, and published in the 1952 paper "A Method for the Construction of Minimum-Redundancy Codes".

Huffman coding uses a specific method for choosing the representation for each symbol, resulting in a prefix code (sometimes called "prefix-free codes") (that is, the bit string representing some particular symbol is never a prefix of the bit string representing any other symbol) that expresses the most common characters using shorter strings of bits than are used for less common source symbols. Huffman was able to design the most efficient compression method *of this type*: no other mapping of individual source symbols to unique strings of bits will produce a smaller average output size when the actual symbol frequencies agree with those used to create the code. A method was later found to do this in linear time if input probabilities (also known as *weights*) are sorted.

For a set of symbols with a uniform probability distribution and a number of members which is a power of two, Huffman coding is equivalent to simple binary block encoding, e.g., ASCII coding. Huffman coding is such a widespread method for creating prefix codes that the term "Huffman code" is widely used as a synonym for "prefix code" even when such a code is not produced by Huffman's algorithm.

Although Huffman coding is optimal for a symbol-by-symbol coding (i.e. a stream of unrelated symbols) with a known input probability distribution, its optimality can sometimes accidentally be over-stated. For example

Huffman tree generated from the exact frequencies of the text "this is an example of a huffman tree". The frequencies and codes of each character are below. Encoding the sentence with this code requires 135 bits, not counting space for the tree.

Char	Freq	Code
space	7	111
a	4	010
e	4	000
f	3	1101
h	2	1010
i	2	1000

DEFLATE - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/DEFLATE

article | discussion | edit this page | history

Log in / create account

DEFLATE

From Wikipedia, the free encyclopedia

This article **does not cite any references or sources**. Please help improve this article by adding citations to reliable sources (ideally, using *inline citations*). Unsourced material may be challenged and removed. *(January 2009)*

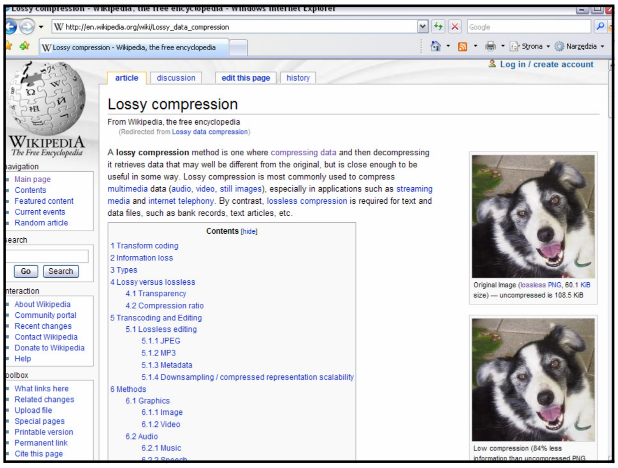
"Deflate" redirects here. For other uses, see Deflation (disambiguation).

Deflate is a lossless data compression algorithm that uses a combination of the LZ77 algorithm and Huffman coding. It was originally defined by Phil Katz for version 2 of his PKZIP archiving tool, and was later specified in RFC 1951.

Deflate is widely thought to be free of any subsisting patents, and at a time before the patent on LZW (which is used in the GIF file format) expired, this has led to its use in gzip compressed files and PNG image files, in addition to the ZIP file format for which Katz originally designed it.

Contents [hide]

Stream format [edit]



Lossy compression - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/Lossy_data_compression

article | discussion | edit this page | history

Log in / create account

Lossy compression

From Wikipedia, the free encyclopedia
(Redirected from Lossy data compression)

A **lossy compression** method is one where compressing data and then decompressing it retrieves data that may well be different from the original, but is close enough to be useful in some way. Lossy compression is most commonly used to compress multimedia data (audio, video, still images), especially in applications such as streaming media and internet telephony. By contrast, lossless compression is required for text and data files, such as bank records, text articles, etc.

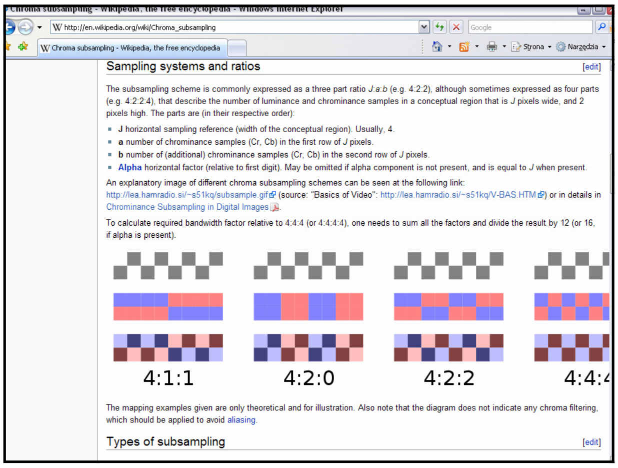
Contents [hide]

Original Image (lossless PNG, 60.1 KiB size) — uncompressed is 108.5 KiB

Low compression (84% less information than uncompressed PNG

Psychovisual redundancy

Visual perception of humans does not work as a camera of a predefined characteristic.
The eye is not equally sensitive to all image features, e.g. the Mach bands



Chroma subsampling - Wikipedia, the free encyclopedia - Windows Internet Explorer
http://en.wikipedia.org/wiki/Chroma_subsampling
Sampling systems and ratios
[edit]
The subsampling scheme is commonly expressed as a three part ratio J:a:b (e.g. 4:2:2), although sometimes expressed as four parts (e.g. 4:2:2:4), that describe the number of luminance and chrominance samples in a conceptual region that is J pixels wide, and 2 pixels high. The parts are (in their respective order):
J horizontal sampling reference (width of the conceptual region). Usually, 4.
a number of chrominance samples (Cr, Cb) in the first row of J pixels.
b number of (additional) chrominance samples (Cr, Cb) in the second row of J pixels.
Alpha horizontal factor (relative to first digit). May be omitted if alpha component is not present, and is equal to J when present.
An explanatory image of different chroma subsampling schemes can be seen at the following link: http://lea.hamradio.si/~s51kq/subsample.gif (source: "Basics of Video": http://lea.hamradio.si/~s51kq/V-BAS.HTM) or in details in Chrominance Subsampling in Digital Images.
To calculate required bandwidth factor relative to 4:4:4 (or 4:4:4:4), one needs to sum all the factors and divide the result by 12 (or 16, if alpha is present).
4:1:1
4:2:0
4:2:2
The mapping examples given are only theoretical and for illustration. Also note that the diagram does not indicate any chroma filtering, which should be applied to avoid aliasing.
Types of subsampling
[edit]

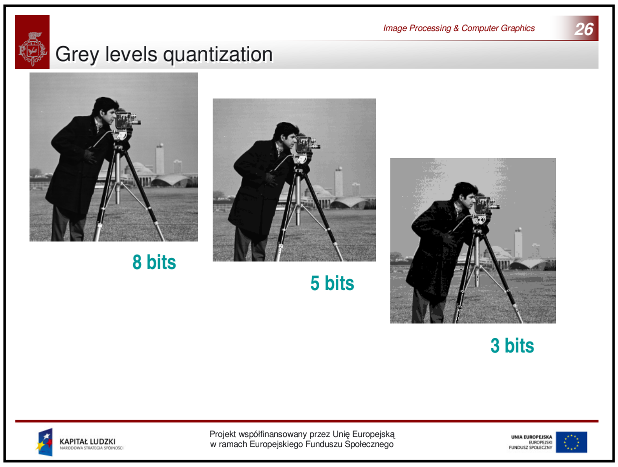
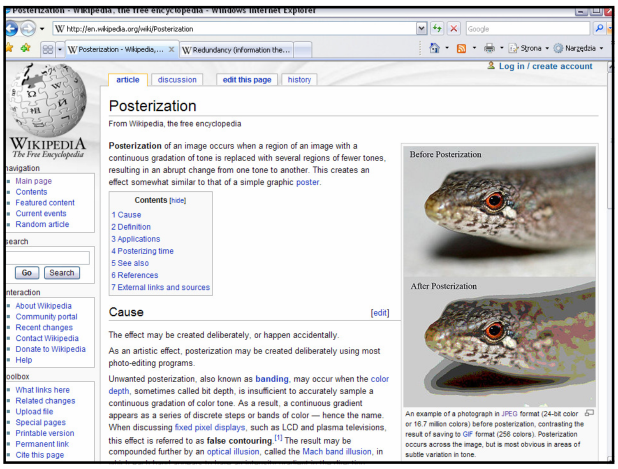


Image Processing & Computer Graphics
26
Grey levels quantization
8 bits
5 bits
3 bits
KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego



Posterization - Wikipedia, the free encyclopedia - Windows Internet Explorer
http://en.wikipedia.org/wiki/Posterization
Log in / create account
article
discussion
edit this page
history
Posterization
From Wikipedia, the free encyclopedia
Posterization of an image occurs when a region of an image with a continuous gradation of tone is replaced with several regions of fewer tones, resulting in an abrupt change from one tone to another. This creates an effect somewhat similar to that of a simple graphic poster.
Contents [hide]
1 Cause
2 Definition
3 Applications
4 Posterizing time
5 See also
6 References
7 External links and sources
Cause
[edit]
The effect may be created deliberately, or happen accidentally.
As an artistic effect, posterization may be created deliberately using most photo-editing programs.
Unwanted posterization, also known as banding, may occur when the color depth, sometimes called bit depth, is insufficient to accurately sample a continuous gradation of color tone. As a result, a continuous gradient appears as a series of discrete steps or bands of color — hence the name. When discussing fixed pixel displays, such as LCD and plasma televisions, this effect is referred to as false contouring.[1] The result may be compounded further by an optical illusion, called the Mach band illusion, in
Before Posterization
After Posterization
An example of a photograph in JPEG format (24-bit color or 16.7 million colors) before posterization, contrasting the result of saving to GIF format (256 colors). Posterization occurs across the image, but is most obvious in areas of subtle variation in tone.

Transform coding

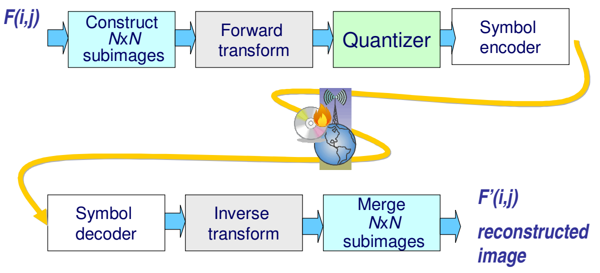
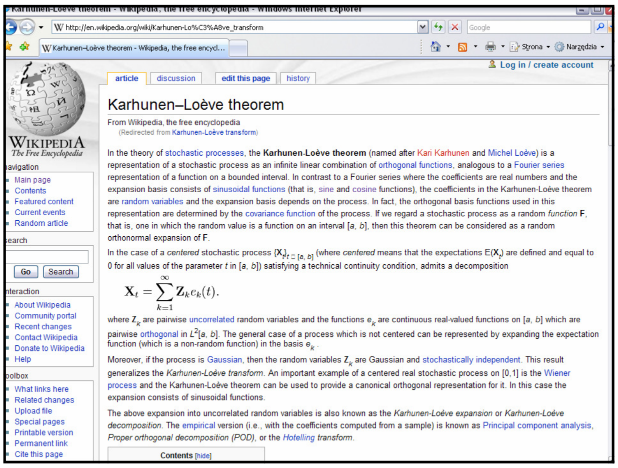


Image transforms

- **Fourier Transform -** impractical
- **The Karhunen-Loeve expansion** - minimizes the mean square reconstruction error, computationally complex, no fast transform
- **Discrete Cosine Transform (DCT)** – high compression ratios achievable (fast vanishing of cosine coefficients), fast transform exists
- **The wavelet transform** – new compression metods developed, competitive or better than DCT, fast transform exists



Karhunen–Loève theorem

From Wikipedia, the free encyclopedia
(Redirected from Karhunen-Loève transform)

In the theory of stochastic processes, the **Karhunen-Loève theorem** (named after Kari Karhunen and Michel Loève) is a representation of a stochastic process as an infinite linear combination of orthogonal functions, analogous to a Fourier series representation of a function on a bounded interval. In contrast to a Fourier series where the coefficients are real numbers and the expansion basis consists of sinusoidal functions (that is, sine and cosine functions), the coefficients in the Karhunen-Loève theorem are random variables and the expansion basis depends on the process. In fact, the orthogonal basis functions used in this representation are determined by the covariance function of the process. If we regard a stochastic process as a random *function* **F**, that is, one in which the random value is a function on an interval [*a*, *b*], then this theorem can be considered as a random orthonormal expansion of **F**.

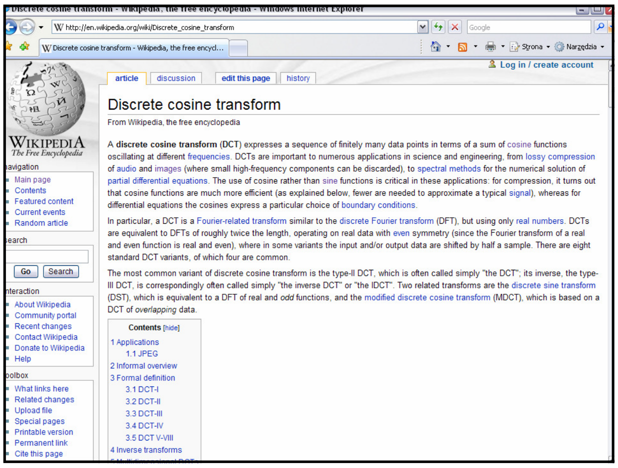
In the case of a *centered* stochastic process $\{X_t\}_{t \in [a,b]}$ (where *centered* means that the expectations $E(X_t)$ are defined and equal to 0 for all values of the parameter *t* in [*a*, *b*]) satisfying a technical continuity condition, admits a decomposition

$$\mathbf{X}_t = \sum_{k=1}^{\infty} \mathbf{Z}_k e_k(t).$$

where Z_k are pairwise uncorrelated random variables and the functions e_k are continuous real-valued functions on [*a*, *b*] which are pairwise orthogonal in $L^2[a, b]$. The general case of a process which is not centered can be represented by expanding the expectation function (which is a non-random function) in the basis e_k.

Moreover, if the process is Gaussian, then the random variables Z_k are Gaussian and stochastically independent. This result generalizes the *Karhunen-Loève transform*. An important example of a centered real stochastic process on [0,1] is the Wiener process and the Karhunen-Loève theorem can be used to provide a canonical orthogonal representation for it. In this case the expansion consists of sinusoidal functions.

The above expansion into uncorrelated random variables is also known as the *Karhunen-Loève expansion* or *Karhunen-Loève decomposition*. The empirical version (i.e., with the coefficients computed from a sample) is known as Principal component analysis, *Proper orthogonal decomposition (POD)*, or the *Hotelling transform*.



Discrete cosine transform - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/Discrete_cosine_transform

Discrete cosine transform

From Wikipedia, the free encyclopedia

A **discrete cosine transform** (DCT) expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies. DCTs are important to numerous applications in science and engineering, from lossy compression of audio and images (where small high-frequency components can be discarded), to spectral methods for the numerical solution of partial differential equations. The use of cosine rather than sine functions is critical in these applications: for compression, it turns out that cosine functions are much more efficient (as explained below, fewer are needed to approximate a typical signal), whereas for differential equations the cosines express a particular choice of boundary conditions.

In particular, a DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even), where in some variants the input and/or output data are shifted by half a sample. There are eight standard DCT variants, of which four are common.

The most common variant of discrete cosine transform is the type-II DCT, which is often called simply "the DCT"; its inverse, the type-III DCT, is correspondingly often called simply "the inverse DCT" or "the IDCT". Two related transforms are the discrete sine transform (DST), which is equivalent to a DFT of real and *odd* functions, and the modified discrete cosine transform (MDCT), which is based on a DCT of *overlapping* data.

Contents [hide]

Discrete wavelet transform - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/Discrete_wavelet_transform

Discrete wavelet transform

From Wikipedia, the free encyclopedia

In numerical analysis and functional analysis, a **discrete wavelet transform** (DWT) is any wavelet transform for which the wavelets are discretely sampled. As with other wavelet transforms, a key advantage it has over Fourier transforms is temporal resolution: it captures both frequency *and* location information.

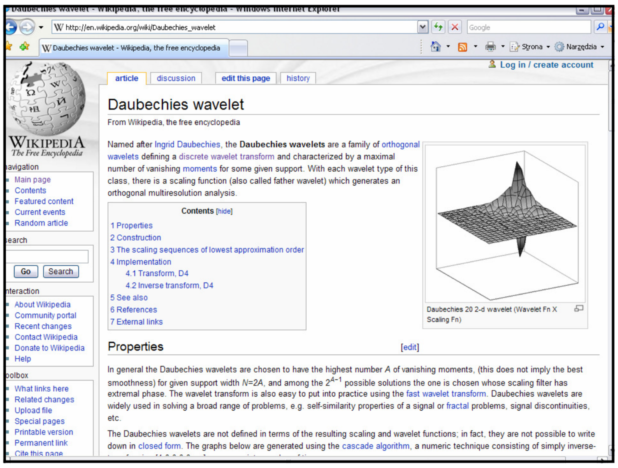
Contents [hide]

An example of the 2D discrete wavelet transform that is used in JPEG2000. The original image is high-pass filtered, yielding the three large images, each describing local changes in brightness (details) in the original image. It is then low-pass filtered and downscaled, yielding an approximation image; this image is high-pass filtered to produce the three smaller detail images, and low-pass filtered to produce the final approximation image in the upper-left.

Examples [edit]

Haar wavelets [edit]

Main article: Haar wavelet



Daubechies wavelet - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/Daubechies_wavelet

Daubechies wavelet

From Wikipedia, the free encyclopedia

Named after Ingrid Daubechies, the **Daubechies wavelets** are a family of orthogonal wavelets defining a discrete wavelet transform and characterized by a maximal number of vanishing moments for some given support. With each wavelet type of this class, there is a scaling function (also called father wavelet) which generates an orthogonal multiresolution analysis.

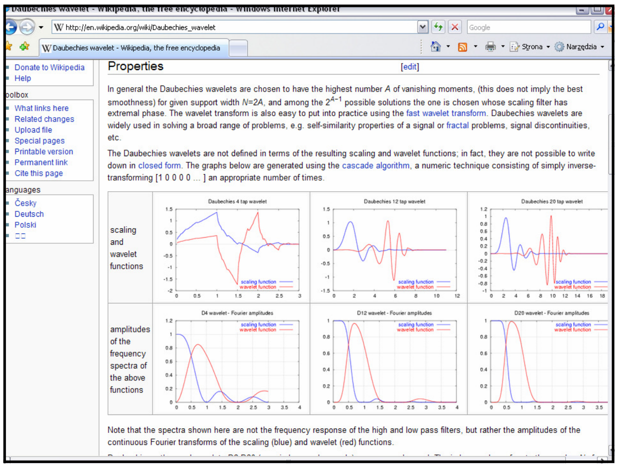
Contents [hide]

Daubechies 20 2-d wavelet (Wavelet Fn X Scaling Fn)

Properties [edit]

In general the Daubechies wavelets are chosen to have the highest number *A* of vanishing moments, (this does not imply the best smoothness) for given support width $N=2A$, and among the 2^{A-1} possible solutions the one is chosen whose scaling filter has extremal phase. The wavelet transform is also easy to put into practice using the fast wavelet transform. Daubechies wavelets are widely used in solving a broad range of problems, e.g. self-similarity properties of a signal or fractal problems, signal discontinuities, etc.

The Daubechies wavelets are not defined in terms of the resulting scaling and wavelet functions; in fact, they are not possible to write down in closed form. The graphs below are generated using the cascade algorithm, a numeric technique consisting of simply inverse-


Daubechies wavelet - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/Daubechies_wavelet

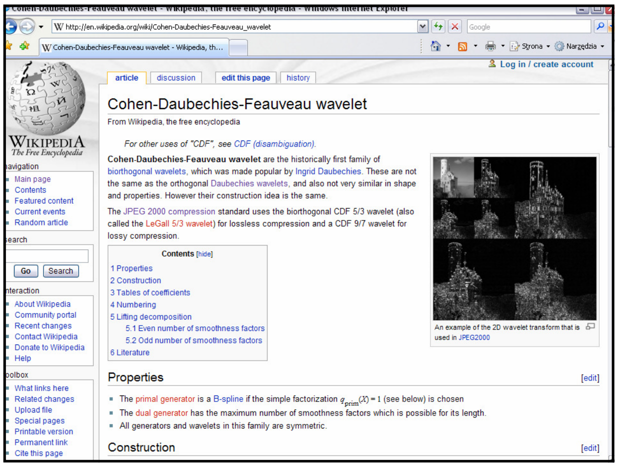
Properties

[edit]

In general the Daubechies wavelets are chosen to have the highest number *A* of vanishing moments, (this does not imply the best smoothness) for given support width $N=2A$, and among the 2^{A-1} possible solutions the one is chosen whose scaling filter has extremal phase. The wavelet transform is also easy to put into practice using the fast wavelet transform. Daubechies wavelets are widely used in solving a broad range of problems, e.g. self-similarity properties of a signal or fractal problems, signal discontinuities, etc.

The Daubechies wavelets are not defined in terms of the resulting scaling and wavelet functions; in fact, they are not possible to write down in closed form. The graphs below are generated using the cascade algorithm, a numeric technique consisting of simply inverse-transforming [1 0 0 0 0 ...] an appropriate number of times.

Note that the spectra shown here are not the frequency response of the high and low pass filters, but rather the amplitudes of the continuous Fourier transforms of the scaling (blue) and wavelet (red) functions.


Cohen-Daubechies-Feauveau wavelet - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/Cohen-Daubechies-Feauveau_wavelet

article | discussion | edit this page | history

Cohen-Daubechies-Feauveau wavelet

From Wikipedia, the free encyclopedia

For other uses of "CDF", see CDF (disambiguation).

Cohen-Daubechies-Feauveau wavelet are the historically first family of biorthogonal wavelets, which was made popular by Ingrid Daubechies. These are not the same as the orthogonal Daubechies wavelets, and also not very similar in shape and properties. However their construction idea is the same.

The JPEG 2000 compression standard uses the biorthogonal CDF 5/3 wavelet (also called the LeGall 5/3 wavelet) for lossless compression and a CDF 9/7 wavelet for lossy compression.

Contents [hide]

An example of the 2D wavelet transform that is used in JPEG2000

Properties

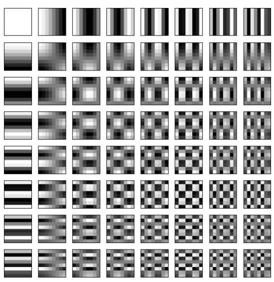
[edit]

- The primal generator is a B-spline if the simple factorization $q_{\mathrm{prim}}(X)=1$ (see below) is chosen
- The dual generator has the maximum number of smoothness factors which is possible for its length.
- All generators and wavelets in this family are symmetric.

Construction

[edit]

DCT basis functions



DCT coefficients of image blocks assign a weight to each of the basis functions

Image 8x8 blocks are obtained by computing a linear combination of these DCT basis functions

KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

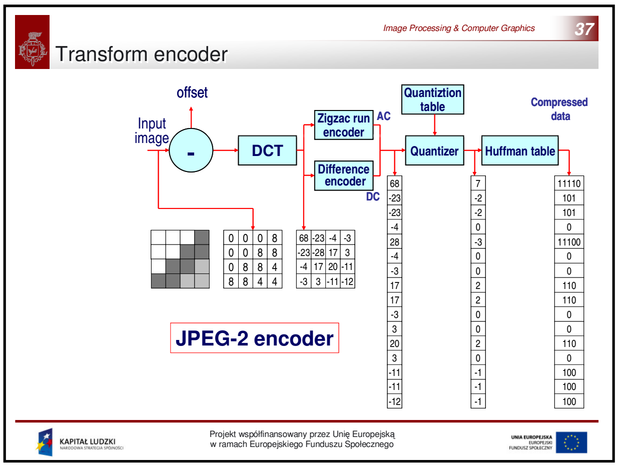


Image Processing & Computer Graphics
37
Transform encoder
offset
Input image
DCT
Zigzac run encoder
AC
Difference encoder
DC
Quantiztion table
Quantizer
Huffman table
Compressed data
JPEG-2 encoder
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego
KAPITAŁ LUDZKI

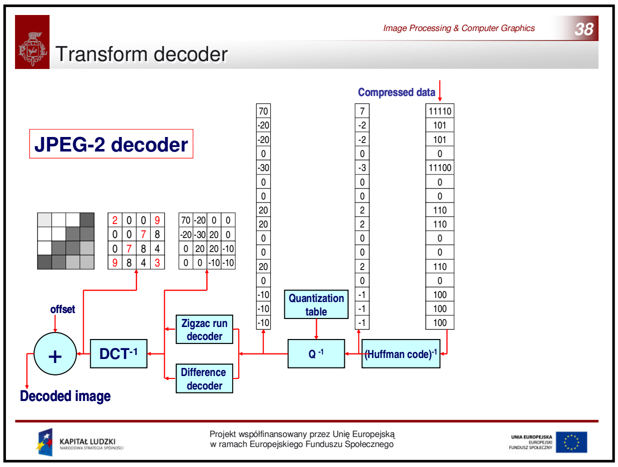
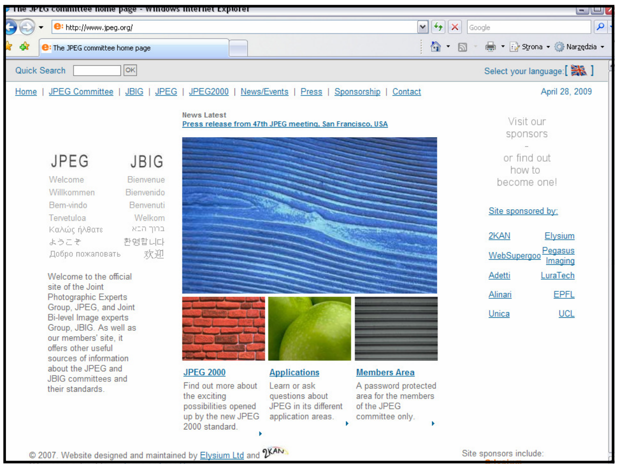
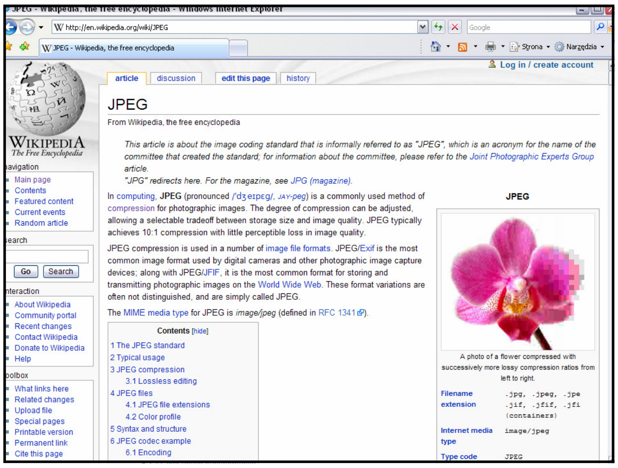


Image Processing & Computer Graphics
38
Transform decoder
Compressed data
JPEG-2 decoder
offset
DCT-1
Zigzac run decoder
Difference decoder
Quantization table
Q-1
(Huffman code)-1
Decoded image
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego
KAPITAŁ LUDZKI


http://www.jpeg.org/
The JPEG committee home page
Home | JPEG Committee | JBIG | JPEG | JPEG2000 | News/Events | Press | Sponsorship | Contact
April 28, 2009
JPEG
JBIG
Welcome to the official site of the Joint Photographic Experts Group, JPEG, and Joint Bi-level Image experts Group, JBIG. As well as our members' site, it offers other useful sources of information about the JPEG and JBIG committees and their standards.
JPEG 2000
Applications
Members Area
Site sponsored by:
© 2007. Website designed and maintained by Elysium Ltd and 2KAN



JPEG - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/JPEG

article | discussion | edit this page | history

Log in / create account

JPEG

From Wikipedia, the free encyclopedia

This article is about the image coding standard that is informally referred to as "JPEG", which is an acronym for the name of the committee that created the standard; for information about the committee, please refer to the Joint Photographic Experts Group article.

"JPG" redirects here. For the magazine, see JPG (magazine).

In computing, **JPEG** (pronounced /ˈdʒeɪpɛɡ/, JAY-peg) is a commonly used method of compression for photographic images. The degree of compression can be adjusted, allowing a selectable tradeoff between storage size and image quality. JPEG typically achieves 10:1 compression with little perceptible loss in image quality.

JPEG compression is used in a number of image file formats. JPEG/Exif is the most common image format used by digital cameras and other photographic image capture devices; along with JPEG/JFIF, it is the most common format for storing and transmitting photographic images on the World Wide Web. These format variations are often not distinguished, and are simply called JPEG.

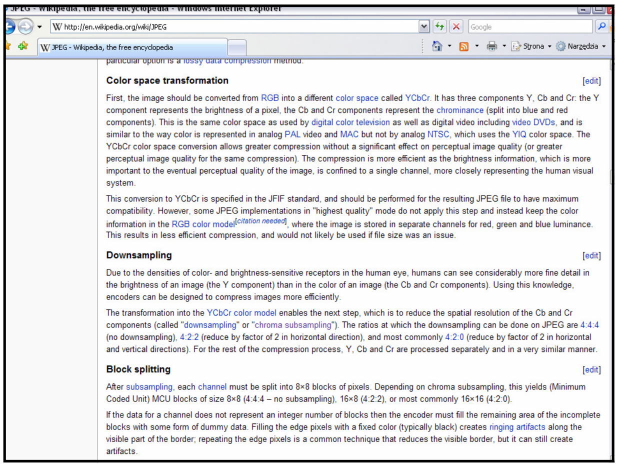
The MIME media type for JPEG is *image/jpeg* (defined in RFC 1341).

Contents [hide]

JPEG

A photo of a flower compressed with successively more lossy compression ratios from left to right.

Filename extension	.jpg, .jpeg, .jpe .jif, .jfif, .jfi (containers)
Internet media type	image/jpeg
Type code	JPEG



JPEG - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/JPEG

particular option is a lossy data compression method.

Color space transformation [edit]

First, the image should be converted from RGB into a different color space called YCbCr. It has three components Y, Cb and Cr: the Y component represents the brightness of a pixel, the Cb and Cr components represent the chrominance (split into blue and red components). This is the same color space as used by digital color television as well as digital video including video DVDs, and is similar to the way color is represented in analog PAL video and MAC but not by analog NTSC, which uses the YIQ color space. The YCbCr color space conversion allows greater compression without a significant effect on perceptual image quality (or greater perceptual image quality for the same compression). The compression is more efficient as the brightness information, which is more important to the eventual perceptual quality of the image, is confined to a single channel, more closely representing the human visual system.

This conversion to YCbCr is specified in the JFIF standard, and should be performed for the resulting JPEG file to have maximum compatibility. However, some JPEG implementations in "highest quality" mode do not apply this step and instead keep the color information in the RGB color model[citation needed], where the image is stored in separate channels for red, green and blue luminance. This results in less efficient compression, and would not likely be used if file size was an issue.

Downsampling [edit]

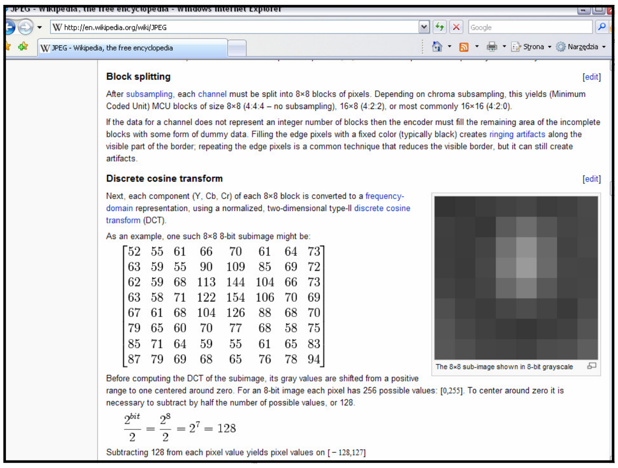
Due to the densities of color- and brightness-sensitive receptors in the human eye, humans can see considerably more fine detail in the brightness of an image (the Y component) than in the color of an image (the Cb and Cr components). Using this knowledge, encoders can be designed to compress images more efficiently.

The transformation into the YCbCr color model enables the next step, which is to reduce the spatial resolution of the Cb and Cr components (called "downsampling" or "chroma subsampling"). The ratios at which the downsampling can be done on JPEG are 4:4:4 (no downsampling), 4:2:2 (reduce by factor of 2 in horizontal direction), and most commonly 4:2:0 (reduce by factor of 2 in horizontal and vertical directions). For the rest of the compression process, Y, Cb and Cr are processed separately and in a very similar manner.

Block splitting [edit]

After subsampling, each channel must be split into 8×8 blocks of pixels. Depending on chroma subsampling, this yields (Minimum Coded Unit) MCU blocks of size 8×8 (4:4:4 – no subsampling), 16×8 (4:2:2), or most commonly 16×16 (4:2:0).

If the data for a channel does not represent an integer number of blocks then the encoder must fill the remaining area of the incomplete blocks with some form of dummy data. Filling the edge pixels with a fixed color (typically black) creates ringing artifacts along the visible part of the border; repeating the edge pixels is a common technique that reduces the visible border, but it can still create artifacts.



JPEG - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/JPEG

Block splitting [edit]

After subsampling, each channel must be split into 8×8 blocks of pixels. Depending on chroma subsampling, this yields (Minimum Coded Unit) MCU blocks of size 8×8 (4:4:4 – no subsampling), 16×8 (4:2:2), or most commonly 16×16 (4:2:0).

If the data for a channel does not represent an integer number of blocks then the encoder must fill the remaining area of the incomplete blocks with some form of dummy data. Filling the edge pixels with a fixed color (typically black) creates ringing artifacts along the visible part of the border; repeating the edge pixels is a common technique that reduces the visible border, but it can still create artifacts.

Discrete cosine transform [edit]

Next, each component (Y, Cb, Cr) of each 8×8 block is converted to a frequency-domain representation, using a normalized, two-dimensional type-II discrete cosine transform (DCT).

As an example, one such 8×8 8-bit subimage might be:

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

The 8×8 sub-image shown in 8-bit grayscale

Before computing the DCT of the subimage, its gray values are shifted from a positive range to one centered around zero. For an 8-bit image each pixel has 256 possible values: $[0,255]$. To center around zero it is necessary to subtract by half the number of possible values, or 128.

$$\frac{2^{bit}}{2} = \frac{2^8}{2} = 2^7 = 128$$

Subtracting 128 from each pixel value yields pixel values on $[-128,127]$

range to one centered around zero. For an 8-bit image each pixel has 256 possible values: [0,255]. To center around zero it is necessary to subtract by half the number of possible values, or 128.

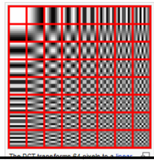
$$\frac{2^{bit}}{2} = \frac{2^8}{2} = 2^7 = 128$$

Subtracting 128 from each pixel value yields pixel values on $[-128,127]$

$$\overset{x}{\longrightarrow}$$

$$\begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix} \Big\downarrow y$$

The next step is to take the two-dimensional DCT, which is given by:



JPEG - Wikipedia, the free encyclopedia - Windows Internet Explorer
http://en.wikipedia.org/wiki/JPEG
JPEG - Wikipedia, the free encyclopedia
Strona Narzędzia

Quantization [edit]

The human eye is good at seeing small differences in brightness over a relatively large area, but not so good at distinguishing the exact strength of a high frequency brightness variation. This allows one to greatly reduce the amount of information in the high frequency components. This is done by simply dividing each component in the frequency domain by a constant for that component, and then rounding to the nearest integer. This is the main lossy operation in the whole process. As a result of this, it is typically the case that many of the higher frequency components are rounded to zero, and many of the rest become small positive or negative numbers, which take many fewer bits to store.

A typical quantization matrix, as specified in the original JPEG Standard[5], is as follows:

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

The quantized DCT coefficients are computed with

$$B_{j,k} = \mathrm{round}\left(\frac{G_{j,k}}{Q_{j,k}}\right) \text{ for } j = 0,1,2,\cdots,N_1-1; k = 0,1,2,\cdots,N_2-1$$

where G is the unquantized DCT coefficients; Q is the quantization matrix above; and B is the quantized DCT coefficients. (Note that this is in **no way** matrix multiplication.)

Using this quantization matrix with the DCT coefficient matrix from above results in:

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

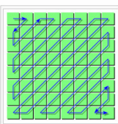
JPEG - Wikipedia, the free encyclopedia - Windows Internet Explorer
http://en.wikipedia.org/wiki/JPEG
JPEG - Wikipedia, the free encyclopedia
Strona Narzędzia

Entropy coding [edit]

Main article: Entropy encoding

Entropy coding is a special form of lossless data compression. It involves arranging the image components in a "zigzag" order employing run-length encoding (RLE) algorithm that groups similar frequencies together, inserting length coding zeros, and then using Huffman coding on what is left.

The JPEG standard also allows, but does not require, the use of arithmetic coding, which is mathematically superior to Huffman coding. However, this feature is rarely used as it is covered by patents and because it is much slower to encode and decode compared to Huffman coding. Arithmetic coding typically makes files about 5% smaller.

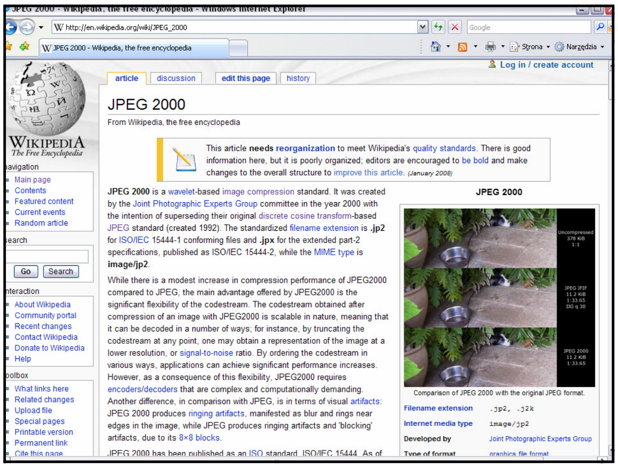


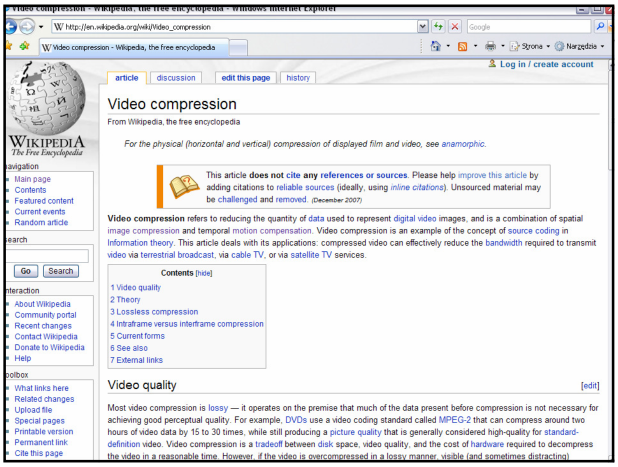
Zigzag ordering of JPEG image components

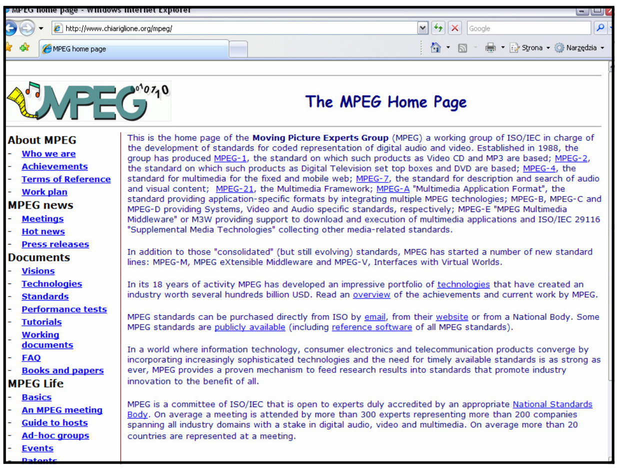
The zigzag sequence for the above quantized coefficients are shown below. (The format shown is just for ease of understanding/viewing.)

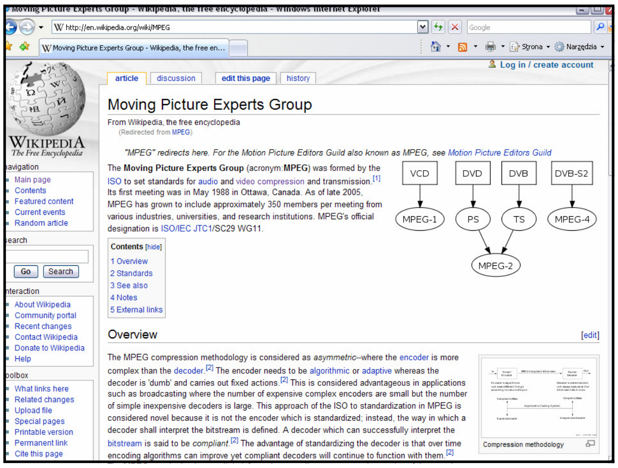
```
−26
−3   0
−3  −2  −6
 2  −4   1  −4
 1   1   5   1   2
−1   1  −1   2   0   0
 0   0   0  −1  −1   0   0
 0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0
 0   0   0   0   0   0
 0   0   0   0   0
 0   0   0   0
 0   0   0
 0   0
 0
```

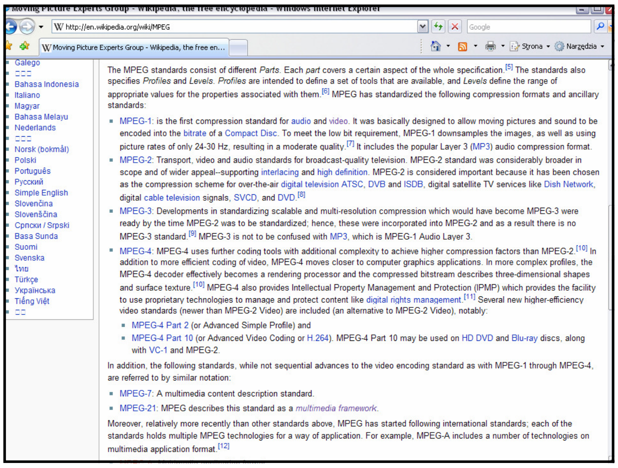
If the i-th block is represented by Bi and positions within each block are represented by (p,q) where p = 0, 1, ..., 7 and q = 0, 1, ..., 7,

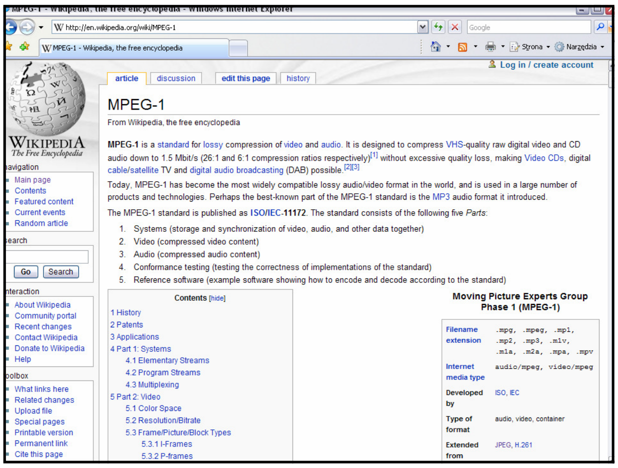

JPEG 2000 - Wikipedia, the free encyclopedia - Windows Internet Explorer
http://en.wikipedia.org/wiki/JPEG_2000
JPEG 2000
From Wikipedia, the free encyclopedia
This article needs reorganization to meet Wikipedia's quality standards. There is good information here, but it is poorly organized; editors are encouraged to be bold and make changes to the overall structure to improve this article. (January 2008)
JPEG 2000 is a wavelet-based image compression standard. It was created by the Joint Photographic Experts Group committee in the year 2000 with the intention of superseding their original discrete cosine transform-based JPEG standard (created 1992). The standardized filename extension is .jp2 for ISO/IEC 15444-1 conforming files and .jpx for the extended part-2 specifications, published as ISO/IEC 15444-2, while the MIME type is image/jp2.
While there is a modest increase in compression performance of JPEG2000 compared to JPEG, the main advantage offered by JPEG2000 is the significant flexibility of the codestream. The codestream obtained after compression of an image with JPEG2000 is scalable in nature, meaning that it can be decoded in a number of ways; for instance, by truncating the codestream at any point, one may obtain a representation of the image at a lower resolution, or signal-to-noise ratio. By ordering the codestream in various ways, applications can achieve significant performance increases. However, as a consequence of this flexibility, JPEG2000 requires encoders/decoders that are complex and computationally demanding. Another difference, in comparison with JPEG, is in terms of visual artifacts: JPEG 2000 produces ringing artifacts, manifested as blur and rings near edges in the image, while JPEG produces ringing artifacts and 'blocking' artifacts, due to its 8×8 blocks.
Comparison of JPEG 2000 with the original JPEG format.
Filename extension .jp2, .j2k
Internet media type image/jp2
Developed by Joint Photographic Experts Group


Video compression - Wikipedia, the free encyclopedia - Windows Internet Explorer
http://en.wikipedia.org/wiki/Video_compression
Video compression
From Wikipedia, the free encyclopedia
For the physical (horizontal and vertical) compression of displayed film and video, see anamorphic.
This article does not cite any references or sources. Please help improve this article by adding citations to reliable sources (ideally, using inline citations). Unsourced material may be challenged and removed. (December 2007)
Video compression refers to reducing the quantity of data used to represent digital video images, and is a combination of spatial image compression and temporal motion compensation. Video compression is an example of the concept of source coding in Information theory. This article deals with its applications: compressed video can effectively reduce the bandwidth required to transmit video via terrestrial broadcast, via cable TV, or via satellite TV services.
Contents
1 Video quality
2 Theory
3 Lossless compression
4 Intraframe versus interframe compression
5 Current forms
6 See also
7 External links
Video quality
Most video compression is lossy — it operates on the premise that much of the data present before compression is not necessary for achieving good perceptual quality. For example, DVDs use a video coding standard called MPEG-2 that can compress around two hours of video data by 15 to 30 times, while still producing a picture quality that is generally considered high-quality for standard-definition video. Video compression is a tradeoff between disk space, video quality, and the cost of hardware required to decompress the video in a reasonable time.


MPEG home page - Windows Internet Explorer
http://www.chiariglione.org/mpeg/
The MPEG Home Page
About MPEG
Who we are
Achievements
Terms of Reference
Work plan
MPEG news
Meetings
Hot news
Press releases
Documents
Visions
Technologies
Standards
Performance tests
Tutorials
Working documents
FAQ
Books and papers
MPEG Life
Basics
An MPEG meeting
Guide to hosts
Ad-hoc groups
Events
This is the home page of the Moving Picture Experts Group (MPEG) a working group of ISO/IEC in charge of the development of standards for coded representation of digital audio and video. Established in 1988, the group has produced MPEG-1, the standard on which such products as Video CD and MP3 are based; MPEG-2, the standard on which such products as Digital Television set top boxes and DVD are based; MPEG-4, the standard for multimedia for the fixed and mobile web; MPEG-7, the standard for description and search of audio and visual content; MPEG-21, the Multimedia Framework; MPEG-A "Multimedia Application Format", the standard providing application-specific formats by integrating multiple MPEG technologies; MPEG-B, MPEG-C and MPEG-D providing Systems, Video and Audio specific standards, respectively; MPEG-E "MPEG Multimedia Middleware" or M3W providing support to download and execution of multimedia applications and ISO/IEC 29116 "Supplemental Media Technologies" collecting other media-related standards.
In addition to those "consolidated" (but still evolving) standards, MPEG has started a number of new standard lines: MPEG-M, MPEG eXtensible Middleware and MPEG-V, Interfaces with Virtual Worlds.
In its 18 years of activity MPEG has developed an impressive portfolio of technologies that have created an industry worth several hundreds billion USD. Read an overview of the achievements and current work by MPEG.
MPEG standards can be purchased directly from ISO by email, from their website or from a National Body. Some MPEG standards are publicly available (including reference software of all MPEG standards).
In a world where information technology, consumer electronics and telecommunication products converge by incorporating increasingly sophisticated technologies and the need for timely available standards is as strong as ever, MPEG provides a proven mechanism to feed research results into standards that promote industry innovation to the benefit of all.
MPEG is a committee of ISO/IEC that is open to experts duly accredited by an appropriate National Standards Body. On average a meeting is attended by more than 300 experts representing more than 200 companies spanning all industry domains with a stake in digital audio, video and multimedia. On average more than 20 countries are represented at a meeting.


Moving Picture Experts Group
From Wikipedia, the free encyclopedia
(Redirected from MPEG)
"MPEG" redirects here. For the Motion Picture Editors Guild also known as MPEG, see Motion Picture Editors Guild
The Moving Picture Experts Group (acronym: MPEG) was formed by the ISO to set standards for audio and video compression and transmission.[1] Its first meeting was in May 1988 in Ottawa, Canada. As of late 2005, MPEG has grown to include approximately 350 members per meeting from various industries, universities, and research institutions. MPEG's official designation is ISO/IEC JTC1/SC29 WG11.
VCD
DVD
DVB
DVB-S2
MPEG-1
PS
TS
MPEG-4
MPEG-2
Contents [hide]
1 Overview
2 Standards
3 See also
4 Notes
5 External links
Overview
[edit]
The MPEG compression methodology is considered as asymmetric--where the encoder is more complex than the decoder.[2] The encoder needs to be algorithmic or adaptive whereas the decoder is 'dumb' and carries out fixed actions.[2] This is considered advantageous in applications such as broadcasting where the number of expensive complex encoders are small but the number of simple inexpensive decoders is large. This approach of the ISO to standardization in MPEG is considered novel because it is not the encoder which is standardized; instead, the way in which a decoder shall interpret the bitstream is defined. A decoder which can successfully interpret the bitstream is said to be compliant.[2] The advantage of standardizing the decoder is that over time encoding algorithms can improve yet compliant decoders will continue to function with them.[2]
Compression methodology


The MPEG standards consist of different Parts. Each part covers a certain aspect of the whole specification.[5] The standards also specifies Profiles and Levels. Profiles are intended to define a set of tools that are available, and Levels define the range of appropriate values for the properties associated with them.[6] MPEG has standardized the following compression formats and ancillary standards:
MPEG-1: is the first compression standard for audio and video. It was basically designed to allow moving pictures and sound to be encoded into the bitrate of a Compact Disc. To meet the low bit requirement, MPEG-1 downsamples the images, as well as using picture rates of only 24-30 Hz, resulting in a moderate quality.[7] It includes the popular Layer 3 (MP3) audio compression format.
MPEG-2: Transport, video and audio standards for broadcast-quality television. MPEG-2 standard was considerably broader in scope and of wider appeal--supporting interlacing and high definition. MPEG-2 is considered important because it has been chosen as the compression scheme for over-the-air digital television ATSC, DVB and ISDB, digital satellite TV services like Dish Network, digital cable television signals, SVCD, and DVD.[8]
MPEG-3: Developments in standardizing scalable and multi-resolution compression which would have become MPEG-3 were ready by the time MPEG-2 was to be standardized; hence, these were incorporated into MPEG-2 and as a result there is no MPEG-3 standard.[9] MPEG-3 is not to be confused with MP3, which is MPEG-1 Audio Layer 3.
MPEG-4: MPEG-4 uses further coding tools with additional complexity to achieve higher compression factors than MPEG-2.[10] In addition to more efficient coding of video, MPEG-4 moves closer to computer graphics applications. In more complex profiles, the MPEG-4 decoder effectively becomes a rendering processor and the compressed bitstream describes three-dimensional shapes and surface texture.[10] MPEG-4 also provides Intellectual Property Management and Protection (IPMP) which provides the facility to use proprietary technologies to manage and protect content like digital rights management.[11] Several new higher-efficiency video standards (newer than MPEG-2 Video) are included (an alternative to MPEG-2 Video), notably:
MPEG-4 Part 2 (or Advanced Simple Profile) and
MPEG-4 Part 10 (or Advanced Video Coding or H.264). MPEG-4 Part 10 may be used on HD DVD and Blu-ray discs, along with VC-1 and MPEG-2.
In addition, the following standards, while not sequential advances to the video encoding standard as with MPEG-1 through MPEG-4, are referred to by similar notation:
MPEG-7: A multimedia content description standard.
MPEG-21: MPEG describes this standard as a multimedia framework.
Moreover, relatively more recently than other standards above, MPEG has started following international standards; each of the standards holds multiple MPEG technologies for a way of application. For example, MPEG-A includes a number of technologies on multimedia application format.[12]


MPEG-1
From Wikipedia, the free encyclopedia
MPEG-1 is a standard for lossy compression of video and audio. It is designed to compress VHS-quality raw digital video and CD audio down to 1.5 Mbit/s (26:1 and 6:1 compression ratios respectively)[1] without excessive quality loss, making Video CDs, digital cable/satellite TV and digital audio broadcasting (DAB) possible.[2][3]
Today, MPEG-1 has become the most widely compatible lossy audio/video format in the world, and is used in a large number of products and technologies. Perhaps the best-known part of the MPEG-1 standard is the MP3 audio format it introduced.
The MPEG-1 standard is published as ISO/IEC-11172. The standard consists of the following five Parts:
1. Systems (storage and synchronization of video, audio, and other data together)
2. Video (compressed video content)
3. Audio (compressed audio content)
4. Conformance testing (testing the correctness of implementations of the standard)
5. Reference software (example software showing how to encode and decode according to the standard)
Contents [hide]
1 History
2 Patents
3 Applications
4 Part 1: Systems
4.1 Elementary Streams
4.2 Program Streams
4.3 Multiplexing
5 Part 2: Video
5.1 Color Space
5.2 Resolution/Bitrate
5.3 Frame/Picture/Block Types
5.3.1 I-Frames
5.3.2 P-frames
Moving Picture Experts Group Phase 1 (MPEG-1)
Filename extension .mpg, .mpeg, .mp1, .mp2, .mp3, .m1v, .m1a, .m2a, .mpa, .mpv
Internet media type audio/mpeg, video/mpeg
Developed by ISO, IEC
Type of format audio, video, container
Extended from JPEG, H.261

MPEG-1 - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/MPEG-1

Part 2: Video

[edit]

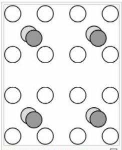
Part 2 of the MPEG-1 standard covers video and is defined in **ISO/IEC-11172-2**. It is heavily based on H.261.

MPEG-1 Video exploits perceptual compression methods to significantly reduce the data rate required by a video stream. It reduces or completely discards information in certain frequencies and areas of the picture that the human eye has limited ability to fully perceive. It also utilizes effective methods to exploit temporal (over time) and spatial (across a picture) redundancy common in video, to achieve better data compression than would be possible otherwise. (See: Video compression)

Color Space

[edit]

Before encoding video to MPEG-1, the color-space is transformed to Y'CbCr (Y'=Luma, Cb=Chroma Blue, Cr=Chroma Red). Luma (brightness, resolution) is stored separately from chroma (color, hue, phase) and even further separated into red and blue components. The chroma is also subsampled to 4:2:0, meaning it is decimated by one half vertically and one half horizontally, to just one quarter the resolution of the video.[1]



Example of 4:2:0 subsampling. The 2 overlapping center circles represent chroma blue and chroma red (color) pixels, while the 4 outside circles represent the luma (brightness).

Because the human eye is much less sensitive to small changes in color than in brightness, chroma subsampling is a very effective way to reduce the amount of video data that needs to be compressed. On videos with fine detail (high spatial complexity) this can manifest as chroma aliasing artifacts. Compared to other digital compression artifacts, this issue seems to be very rarely a source of annoyance.

Because of subsampling, Y'CbCr video must always be stored using even dimensions (divisible by 2), otherwise chroma mismatch ("ghosts") will occur, and it will appear as if the color is ahead of, or behind the rest of the video, much like a shadow.

Y'CbCr is often inaccurately called YUV which is only used in the domain of analog video signals. Similarly, the terms luminance and chrominance are often used instead of the (more accurate) terms luma and chroma.

Resolution/Bitrate

[edit]

MPEG-1 supports resolutions up to 4095×4095 (12-bits), and bitrates up to 100 Mbit/s.[4]

MPEG-1 videos are most commonly seen using Source Input Format (SIF) resolution: 352x240, 352x288, or 320x240. These low resolutions, combined with a bitrate less than 1.5 Mbit/s, make up what is known as a constrained parameters bitstream (CPB), later renamed the "Low Level" (LL) profile in MPEG-2. This is the minimum video specifications any decoder should be able to handle, to be

MPEG-1 - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/MPEG-1

Macroblocks

[edit]

Main article: Macroblock

MPEG-1 operates on video in a series of 8x8 blocks for quantization. However, because chroma (color) is subsampled by a factor of 4, each pair of (red and blue) chroma blocks corresponds to 4 different luma blocks. This set of 6 blocks, with a resolution of 16x16, is called a **macroblock**.

A macroblock is the smallest independent unit of (color) video. Motion vectors (see below) operate solely at the macroblock level.

If the height and/or width of the video is not exact multiples of 16, a full row of macroblocks must still be encoded (though not displayed) to store the remainder of the picture (macroblock padding). This wastes a significant amount of data in the bitstream, and is to be strictly avoided.

Some decoders will also improperly handle videos with partial macroblocks, resulting in visible artifacts.

Motion Vectors

[edit]

To decrease the amount of spatial redundancy in a video, only blocks that change are updated, (up to the maximum GOP size). This is known as **conditional replenishment**. However, this is not very effective by itself. Movement of the objects, and/or the camera may result in large portions of the frame needing to be updated, even though only the position of the previously encoded objects has changed. Through **motion estimation** the encoder can compensate for this movement and remove a large amount of redundant information.

The encoder compares the current frame with adjacent parts of the video from the anchor frame (previous I- or P- frame) in a diamond pattern, up to a (encoder-specific) predefined radius limit from the area of the current macroblock. If a match is found, only the direction and distance (i.e. the *vector* of the *motion*) from the previous video area to the current macroblock need to be encoded into the inter-frame (P- or B- frame). The reverse of this process, performed by the decoder to reconstruct the picture, is called **motion compensation**.

A predicted macroblock rarely matches the current picture perfectly, however. The differences between the estimated matching area, and the real frame/macroblock is called the **prediction error**. The larger the error, the more data must be additionally encoded in the frame. For efficient video compression, it is very important that the encoder is capable of effectively and precisely performing motion estimation.

Motion vectors record the *distance* between two areas on screen based on the number of pixels (called **pels**). MPEG-1 video uses a motion vector (MV) precision of one half of one pixel, or **half-pel**. The finer the precision of the MVs, the more accurate the match is likely to be, and the more efficient the compression. There are trade-offs to higher precision, however. Finer MVs result in larger data size, as larger numbers must be stored in the frame for every single MV, increased coding complexity as increasing levels of

Optical flow - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/Optical_flow

field of motion - Search resul... | Motion field - Wikipedia, the ... | Optical flow - Wikipedia, ...

Log in / create account

article | discussion | edit this page | history

WIKIPEDIA
The Free Encyclopedia

avigation
- Main page
- Contents
- Featured content
- Current events
- Random article

earch

Go | Search

nteraction
- About Wikipedia
- Community portal
- Recent changes
- Contact Wikipedia
- Donate to Wikipedia
- Help

oolbox
- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link
- Cite this page

Optical flow

From Wikipedia, the free encyclopedia

Optical flow or **optic flow** is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene.[1][2] Optical flow techniques such as motion detection, object segmentation, time-to-collision and focus of expansion calculations, motion compensated encoding, and stereo disparity measurement utilize this motion of the objects surfaces, and edges.[3][4]

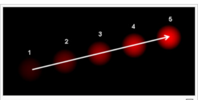


Figure 1: The optical flow vector of a moving object in a video sequence.

Contents [hide]

Estimation of the optical flow

[edit]

Sequences of ordered images allow the estimation of motion as either instantaneous image velocities or discrete image displacements.[5] Fleet and Weiss provide a tutorial introduction to gradient based optical flow .[6] John L. Barron, David J. Fleet, and Steven Beauchemin provides a performance analysis of a number of optical flow techniques. It emphasizes the accuracy and density of measurements.[7]

The optical flow methods try to calculate the motion between two image frames which are taken at times t and $t+\delta t$ at every voxel position. These methods are called differential since they are based on local Taylor series approximations of the image signal, that is: they use partial derivatives with respect to the spatial and temporal coordinates.

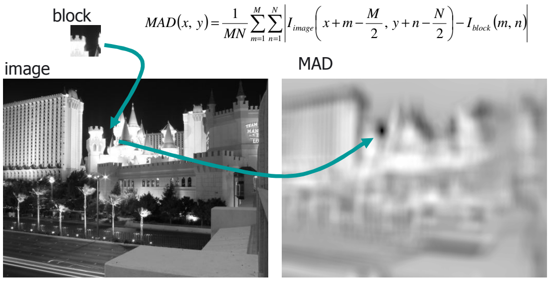
For a 2D+t dimensional case (3D or n-D cases are similar) a voxel at location (x,y,t) with intensity $I(x,y,t)$ will have moved by δx, δy and δt between the two image frames, and the following *image constraint equation* can be given:

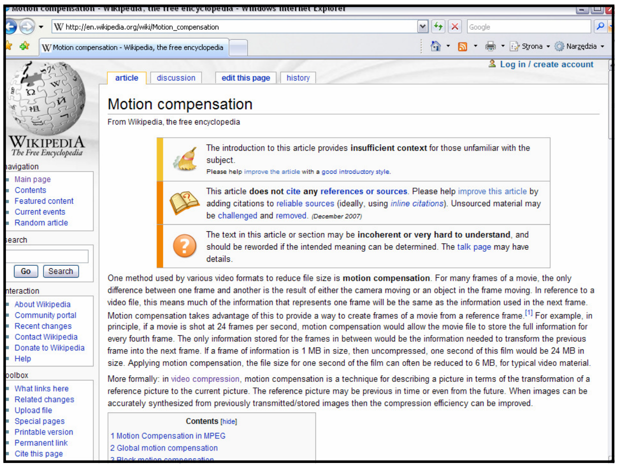
$$I(x,y,t) = I(x+\delta x, y+\delta y, t+\delta t)$$

Block matching

- MAD – Mean Absolute Difference
- SAD - Sum of Absolute Differences
- MI – Mutual Information
- NCC - Normalized Cross-Corelation

$$MAD(x,y)=\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left|I_{image}\left(x+m-\frac{M}{2},\,y+n-\frac{N}{2}\right)-I_{block}(m,n)\right|$$





Motion compensation

From Wikipedia, the free encyclopedia

The introduction to this article provides **insufficient context** for those unfamiliar with the subject.
Please help improve the article with a good introductory style.

This article **does not cite any references or sources**. Please help improve this article by adding citations to reliable sources (ideally, using *inline citations*). Unsourced material may be challenged and removed. *(December 2007)*

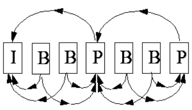
The text in this article or section may be **incoherent or very hard to understand**, and should be reworded if the intended meaning can be determined. The talk page may have details.

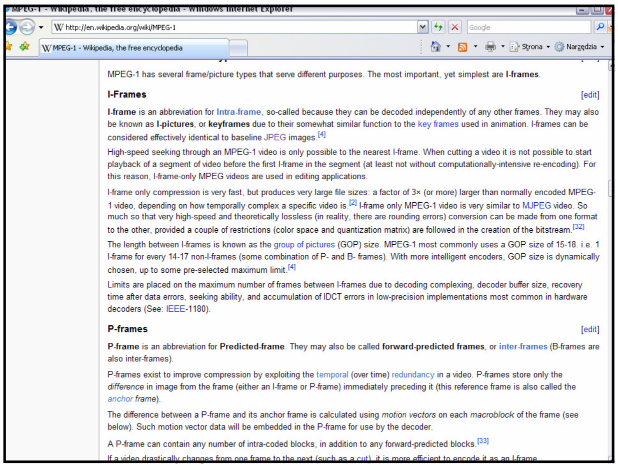
One method used by various video formats to reduce file size is **motion compensation**. For many frames of a movie, the only difference between one frame and another is the result of either the camera moving or an object in the frame moving. In reference to a video file, this means much of the information that represents one frame will be the same as the information used in the next frame. Motion compensation takes advantage of this to provide a way to create frames of a movie from a reference frame.[1] For example, in principle, if a movie is shot at 24 frames per second, motion compensation would allow the movie file to store the full information for every fourth frame. The only information stored for the frames in between would be the information needed to transform the previous frame into the next frame. If a frame of information is 1 MB in size, then uncompressed, one second of this film would be 24 MB in size. Applying motion compensation, the file size for one second of the film can often be reduced to 6 MB, for typical video material.

More formally: in video compression, motion compensation is a technique for describing a picture in terms of the transformation of a reference picture to the current picture. The reference picture may be previous in time or even from the future. When images can be accurately synthesized from previously transmitted/stored images then the compression efficiency can be improved.

Contents [hide]

IPB frames





MPEG-1 - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/MPEG-1

MPEG-1 has several frame/picture types that serve different purposes. The most important, yet simplest are **I-frames**.

I-Frames

[edit]

I-frame is an abbreviation for Intra-frame, so-called because they can be decoded independently of any other frames. They may also be known as **I-pictures**, or **keyframes** due to their somewhat similar function to the key frames used in animation. I-frames can be considered effectively identical to baseline JPEG images.[4]

High-speed seeking through an MPEG-1 video is only possible to the nearest I-frame. When cutting a video it is not possible to start playback of a segment of video before the first I-frame in the segment (at least not without computationally-intensive re-encoding). For this reason, I-frame-only MPEG videos are used in editing applications.

I-frame only compression is very fast, but produces very large file sizes: a factor of 3× (or more) larger than normally encoded MPEG-1 video, depending on how temporally complex a specific video is.[2] I-frame only MPEG-1 video is very similar to MJPEG video. So much so that very high-speed and theoretically lossless (in reality, there are rounding errors) conversion can be made from one format to the other, provided a couple of restrictions (color space and quantization matrix) are followed in the creation of the bitstream.[32]

The length between I-frames is known as the group of pictures (GOP) size. MPEG-1 most commonly uses a GOP size of 15-18. i.e. 1 I-frame for every 14-17 non-I-frames (some combination of P- and B- frames). With more intelligent encoders, GOP size is dynamically chosen, up to some pre-selected maximum limit.[4]

Limits are placed on the maximum number of frames between I-frames due to decoding complexing, decoder buffer size, recovery time after data errors, seeking ability, and accumulation of IDCT errors in low-precision implementations most common in hardware decoders (See: IEEE-1180).

P-frames

[edit]

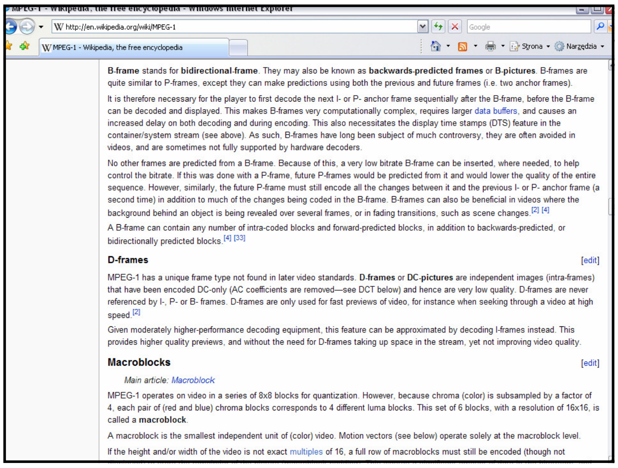
P-frame is an abbreviation for **Predicted-frame**. They may also be called **forward-predicted frames**, or **inter-frames** (B-frames are also inter-frames).

P-frames exist to improve compression by exploiting the temporal (over time) redundancy in a video. P-frames store only the *difference* in image from the frame (either an I-frame or P-frame) immediately preceding it (this reference frame is also called the *anchor frame*).

The difference between a P-frame and its anchor frame is calculated using *motion vectors* on each *macroblock* of the frame (see below). Such motion vector data will be embedded in the P-frame for use by the decoder.

A P-frame can contain any number of intra-coded blocks, in addition to any forward-predicted blocks.[33]

If a video drastically changes from one frame to the next (such as a cut), it is more efficient to encode it as an I-frame.



MPEG-1 - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/MPEG-1

B-frame stands for **bidirectional-frame**. They may also be known as **backwards-predicted frames** or **B-pictures**. B-frames are quite similar to P-frames, except they can make predictions using both the previous and future frames (i.e. two anchor frames).

It is therefore necessary for the player to first decode the next I- or P- anchor frame sequentially after the B-frame, before the B-frame can be decoded and displayed. This makes B-frames very computationally complex, requires larger data buffers, and causes an increased delay on both decoding and during encoding. This also necessitates the display time stamps (DTS) feature in the container/system stream (see above). As such, B-frames have long been subject of much controversy, they are often avoided in videos, and are sometimes not fully supported by hardware decoders.

No other frames are predicted from a B-frame. Because of this, a very low bitrate B-frame can be inserted, where needed, to help control the bitrate. If this was done with a P-frame, future P-frames would be predicted from it and would lower the quality of the entire sequence. However, similarly, the future P-frame must still encode all the changes between it and the previous I- or P- anchor frame (a second time) in addition to much of the changes being coded in the B-frame. B-frames can also be beneficial in videos where the background behind an object is being revealed over several frames, or in fading transitions, such as scene changes.[2] [4]

A B-frame can contain any number of intra-coded blocks and forward-predicted blocks, in addition to backwards-predicted, or bidirectionally predicted blocks.[4] [33]

D-frames

[edit]

MPEG-1 has a unique frame type not found in later video standards. **D-frames** or **DC-pictures** are independent images (intra-frames) that have been encoded DC-only (AC coefficients are removed—see DCT below) and hence are very low quality. D-frames are never referenced by I-, P- or B- frames. D-frames are only used for fast previews of video, for instance when seeking through a video at high speed.[2]

Given moderately higher-performance decoding equipment, this feature can be approximated by decoding I-frames instead. This provides higher quality previews, and without the need for D-frames taking up space in the stream, yet not improving video quality.

Macroblocks

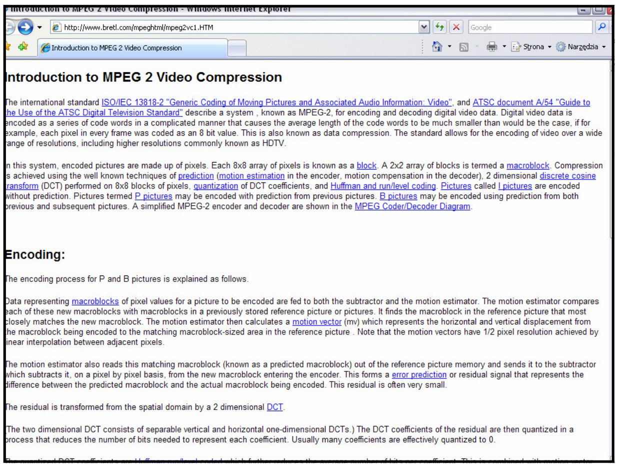
[edit]

Main article: Macroblock

MPEG-1 operates on video in a series of 8x8 blocks for quantization. However, because chroma (color) is subsampled by a factor of 4, each pair of (red and blue) chroma blocks corresponds to 4 different luma blocks. This set of 6 blocks, with a resolution of 16x16, is called a **macroblock**.

A macroblock is the smallest independent unit of (color) video. Motion vectors (see below) operate solely at the macroblock level.

If the height and/or width of the video is not exact multiples of 16, a full row of macroblocks must still be encoded (though not



Introduction to MPEG 2 Video Compression - Windows Internet Explorer

http://www.bretl.com/mpeghtml/mpeg2vc1.HTM

Introduction to MPEG 2 Video Compression

The international standard ISO/IEC 13818-2 "Generic Coding of Moving Pictures and Associated Audio Information: Video", and ATSC document A/54 "Guide to the Use of the ATSC Digital Television Standard" describe a system , known as MPEG-2, for encoding and decoding digital video data. Digital video data is encoded as a series of code words in a complicated manner that causes the average length of the code words to be much smaller than would be the case, if for example, each pixel in every frame was coded as an 8 bit value. This is also known as data compression. The standard allows for the encoding of video over a wide range of resolutions, including higher resolutions commonly known as HDTV.

In this system, encoded pictures are made up of pixels. Each 8x8 array of pixels is known as a block. A 2x2 array of blocks is termed a macroblock. Compression is achieved using the well known techniques of prediction (motion estimation in the encoder, motion compensation in the decoder), 2 dimensional discrete cosine transform (DCT) performed on 8x8 blocks of pixels, quantization of DCT coefficients, and Huffman and run/level coding. Pictures called I pictures are encoded without prediction. Pictures termed P pictures may be encoded with prediction from previous pictures. B pictures may be encoded using prediction from both previous and subsequent pictures. A simplified MPEG-2 encoder and decoder are shown in the MPEG Coder/Decoder Diagram.

Encoding:

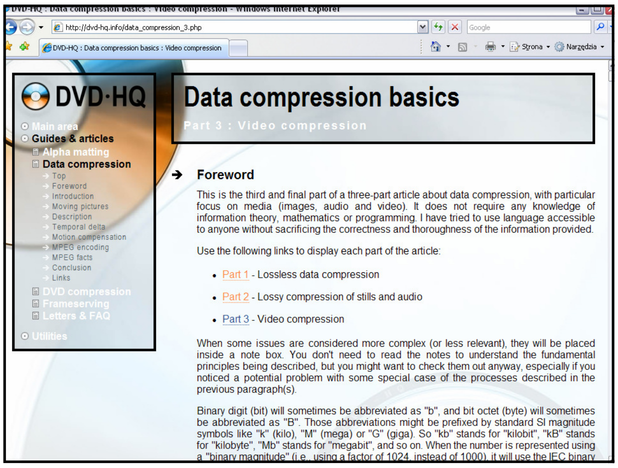
The encoding process for P and B pictures is explained as follows.

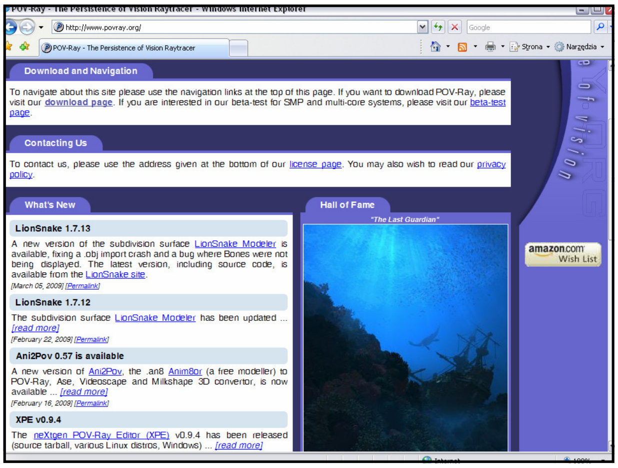
Data representing macroblocks of pixel values for a picture to be encoded are fed to both the subtractor and the motion estimator. The motion estimator compares each of these new macroblocks with macroblocks in a previously stored reference picture or pictures. It finds the macroblock in the reference picture that most closely matches the new macroblock. The motion estimator then calculates a motion vector (mv) which represents the horizontal and vertical displacement from the macroblock being encoded to the matching macroblock-sized area in the reference picture . Note that the motion vectors have 1/2 pixel resolution achieved by linear interpolation between adjacent pixels.

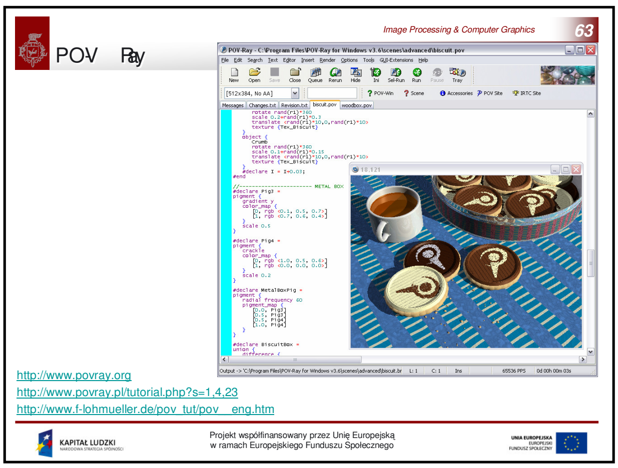
The motion estimator also reads this matching macroblock (known as a predicted macroblock) out of the reference picture memory and sends it to the subtractor which subtracts it, on a pixel by pixel basis, from the new macroblock entering the encoder. This forms a error prediction or residual signal that represents the difference between the predicted macroblock and the actual macroblock being encoded. This residual is often very small.

The residual is transformed from the spatial domain by a 2 dimensional DCT.

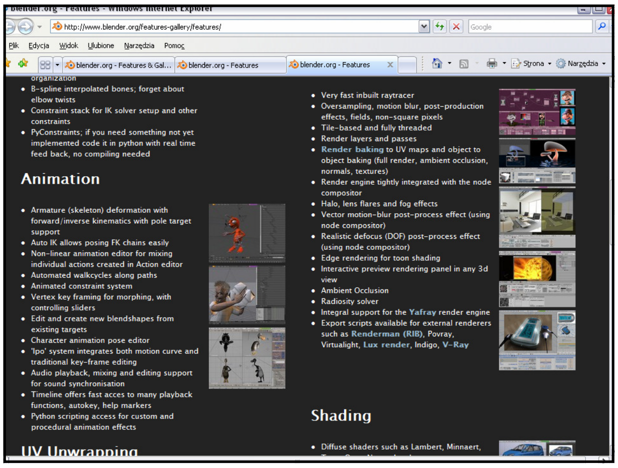
(The two dimensional DCT consists of separable vertical and horizontal one-dimensional DCTs.) The DCT coefficients of the residual are then quantized in a process that reduces the number of bits needed to represent each coefficient. Usually many coefficients are effectively quantized to 0.

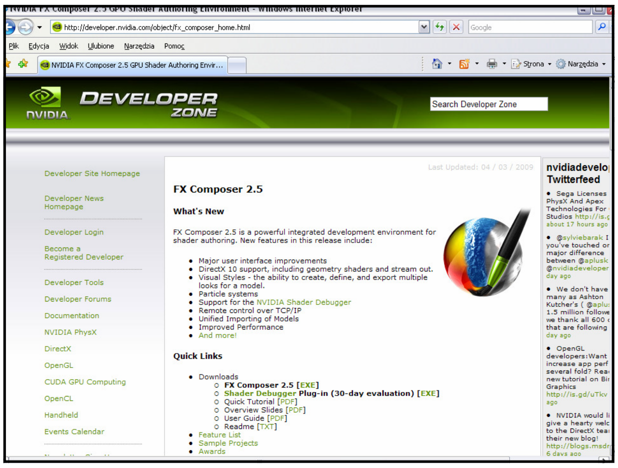

DVD-HQ : Data compression basics : Video compression - Windows Internet Explorer
http://dvd-hq.info/data_compression_3.php
DVD·HQ
Data compression basics
Part 3 : Video compression
Main area
Guides & articles
Alpha matting
Data compression
Top
Foreword
Introduction
Moving pictures
Description
Temporal delta
Motion compensation
MPEG encoding
MPEG facts
Conclusion
Links
DVD compression
Frameserving
Letters & FAQ
Utilities
Foreword
This is the third and final part of a three-part article about data compression, with particular focus on media (images, audio and video). It does not require any knowledge of information theory, mathematics or programming. I have tried to use language accessible to anyone without sacrificing the correctness and thoroughness of the information provided.
Use the following links to display each part of the article:
Part 1 - Lossless data compression
Part 2 - Lossy compression of stills and audio
Part 3 - Video compression
When some issues are considered more complex (or less relevant), they will be placed inside a note box. You don't need to read the notes to understand the fundamental principles being described, but you might want to check them out anyway, especially if you noticed a potential problem with some special case of the processes described in the previous paragraph(s).
Binary digit (bit) will sometimes be abbreviated as "b", and bit octet (byte) will sometimes be abbreviated as "B". Those abbreviations might be prefixed by standard SI magnitude symbols like "k" (kilo), "M" (mega) or "G" (giga). So "kb" stands for "kilobit", "kB" stands for "kilobyte", "Mb" stands for "megabit", and so on. When the number is represented using

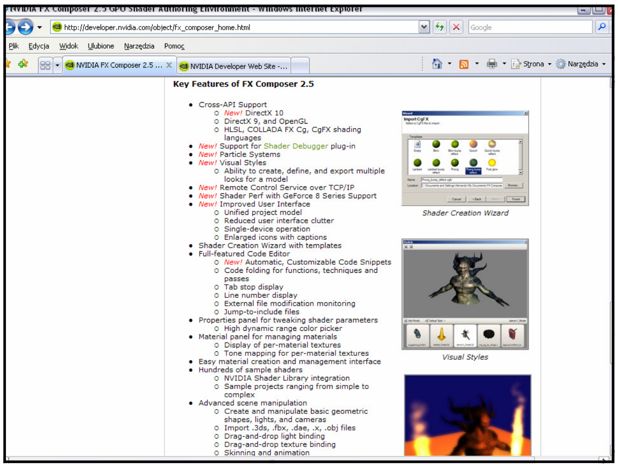

POV-Ray - The Persistence of Vision Raytracer - Windows Internet Explorer
http://www.povray.org/
Download and Navigation
To navigate about this site please use the navigation links at the top of this page. If you want to download POV-Ray, please visit our download page. If you are interested in our beta-test for SMP and multi-core systems, please visit our beta-test page.
Contacting Us
To contact us, please use the address given at the bottom of our license page. You may also wish to read our privacy policy.
What's New
LionSnake 1.7.13
A new version of the subdivision surface LionSnake Modeler is available, fixing a .obj import crash and a bug where Bones were not being displayed. The latest version, including source code, is available from the LionSnake site.
LionSnake 1.7.12
The subdivision surface LionSnake Modeler has been updated ... [read more]
Ani2Pov 0.57 is available
A new version of Ani2Pov, the .an8 Anim8or (a free modeller) to POV-Ray, Ase, Videoscape and Milkshape 3D convertor, is now available ... [read more]
XPE v0.9.4
The neXtgen POV-Ray Editor (XPE) v0.9.4 has been released (source tarball, various Linux distros, Windows) ... [read more]
Hall of Fame
"The Last Guardian"
amazon.com Wish List


POV Ray
Image Processing & Computer Graphics
63
http://www.povray.org
http://www.povray.pl/tutorial.php?s=1,4,23
http://www.f-lohmueller.de/pov_tut/pov__eng.htm
KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA


blender.org - Home - Windows Internet Explorer
http://www.blender.org/
Features & Gallery
Download
Education & Help
Community
Development
Blender
model · shade · animate · render · composite · interactive 3d
Blender is the free open source 3D content creation suite, available for all major operating systems under the GNU General Public License.
version 2.48
Download Now
News Headlines
from BlenderNation
Making a model look 'aged'
April 28, 2009
Blender Animation Commercial
April 28, 2009
Blender calculator for Shapeways printing costs
April 28, 2009
25 Useful Blender tricks that aren't so obvious
Announcements
Blender Foundation Official Updates
Blender 2.49 Release Candidates
April 24, 2009
You can help us getting bugs fixed by testing the first release candidate for 2.49.
Blender Python script licensing
April 17, 2009
Book + DVD
Special offer!
Blender Institute 2009 training
LEARN
CHARACTER ANIMATION
USING BLENDER


blender.org - Features - Windows Internet Explorer
http://www.blender.org/features-gallery/features/
B-spline interpolated bones; forget about elbow twists
Constraint stack for IK solver setup and other constraints
PyConstraints; if you need something not yet implemented code it in python with real time feed back, no compiling needed
Animation
Armature (skeleton) deformation with forward/inverse kinematics with pole target support
Auto IK allows posing FK chains easily
Non-linear animation editor for mixing individual actions created in Action editor
Automated walkcycles along paths
Animated constraint system
Vertex key framing for morphing, with controlling sliders
Edit and create new blendshapes from existing targets
Character animation pose editor
'Ipo' system integrates both motion curve and traditional key-frame editing
Audio playback, mixing and editing support for sound synchronisation
Timeline offers fast acces to many playback functions, autokey, help markers
Python scripting access for custom and procedural animation effects
UV Unwrapping
Very fast inbuilt raytracer
Oversampling, motion blur, post-production effects, fields, non-square pixels
Tile-based and fully threaded
Render layers and passes
Render baking to UV maps and object to object baking (full render, ambient occlusion, normals, textures)
Render engine tightly integrated with the node compositor
Halo, lens flares and fog effects
Vector motion-blur post-process effect (using node compositor)
Realistic defocus (DOF) post-process effect (using node compositor)
Edge rendering for toon shading
Interactive preview rendering panel in any 3d view
Ambient Occlusion
Radiosity solver
Integral support for the Yafray render engine
Export scripts available for external renderers such as Renderman (RIB), Povray, Virtualight, Lux render, Indigo, V-Ray
Shading
Diffuse shaders such as Lambert, Minnaert,


NVIDIA FX Composer 2.5 GPU Shader Authoring Environment - Windows Internet Explorer
http://developer.nvidia.com/object/fx_composer_home.html
NVIDIA
DEVELOPER ZONE
Search Developer Zone
Developer Site Homepage
Developer News Homepage
Developer Login
Become a Registered Developer
Developer Tools
Developer Forums
Documentation
NVIDIA PhysX
DirectX
OpenGL
CUDA GPU Computing
OpenCL
Handheld
Events Calendar
FX Composer 2.5
What's New
FX Composer 2.5 is a powerful integrated development environment for shader authoring. New features in this release include:
Major user interface improvements
DirectX 10 support, including geometry shaders and stream out.
Visual Styles - the ability to create, define, and export multiple looks for a model.
Particle systems
Support for the NVIDIA Shader Debugger
Remote control over TCP/IP
Unified Importing of Models
Improved Performance
And more!
Quick Links
Downloads
FX Composer 2.5 [EXE]
Shader Debugger Plug-in (30-day evaluation) [EXE]
Quick Tutorial [PDF]
Overview Slides [PDF]
User Guide [PDF]
Readme [TXT]
Feature List
Sample Projects
Awards



Summary, discussion and quiz

Summary, discussion and quiz

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

„ Image Processing and Computer Graphics"

Prezentacja multimedialna współfinansowana przez Unię Europejską w ramach Europejskiego Funduszu Społecznego w projekcie pt.

„Innowacyjna dydaktyka bez ograniczeń - zintegrowany rozwój Politechniki Łódzkiej - zarządzanie Uczelnią, nowoczesna oferta edukacyjna i wzmacniania zdolności do zatrudniania osób niepełnosprawnych"

Politechnika Łódzka
Politechnika Łódzka, ul. Żeromskiego 116, 90-924 Łódź, tel. (042) 631 28 83
www.kapitalludzki.p.lodz.pl