

# CHAPTER CONTENTS

---

[6.0 Compression](#)

[6.1 The Need for Compression](#)

[6.2 Audio Compression](#)

[6.2.1 ADPCM](#)

[6.2.2 MP3](#)

[6.3 Bit Stream Compression](#)

[6.3.1 Entropy](#)

[6.3.2 Huffman Code](#)

[6.3.3 Shannon–Fano Code](#)

[6.3.4 Run-Length Coding](#)

[6.3.5 Arithmetic Coding](#)

[6.4 Image Compression](#)

[6.4.1 JPEG](#)

[6.4.2 Discrete Cosine Transform](#)

[6.5 Video Compression](#)

[6.5.1 MPEG](#)

[6.6 Other Compression Techniques](#)

[6.6.1 Wavelets](#)

[6.6.2 Fractals](#)

[Assignment Questions](#)

[For Additional Research](#)



## 6.0 Compression

---

### Objectives

The purpose of this brief section is to:

- Examine the need for compression
  - Examine audio compression
  - Observe how bit stream compression can be performed
  - Examine static image compression
  - Consider moving picture compression
- 

<http://www.image.cityu.edu.hk/~loben/thesis/thesis.html>

[http://tssg3.epfl.ch/pub\\_files/brigger/thesis\\_html/PhD\\_html.html](http://tssg3.epfl.ch/pub_files/brigger/thesis_html/PhD_html.html)

<http://www.dogma.net/DataCompression/index.shtml>

<http://www.data-compression.com/>

http://

### 6.1 The Need for Compression

Audio signals can typically be compressed by a factor of 5:1 to 10:1. Video signals require an even higher compression ratio.

CCIR recommendation 601, samples the luminance signal at 13.5 MHz and the two chromance signals at 6.75 MHz. Since each component is digitized with an 8-bit resolution, the total bit rate is 216 Mbps. This must be reduced before it can be conveyed over most communication links.

The debate over image quality has delayed the widespread deployment of multimedia video technology. Most people expect new videoconferencing technologies to offer image quality equal to if not better than commercial broadcast television. The picture quality of the standard domestic TV and VCR is quite good and is expected to improve to 35-mm film quality with the deployment of HDTV. The FCC has decided that HDTV will be digital, hence the need for video compression.

Some of the characteristics of compression include:

- Lossless vs. lossy
- Spatial/temporal resolution
- Compression/decompression time
- Stop/pause capability
- Real time imaging
- Hardware/software processing

A lossless compression algorithm allows the original video image to be accurately regenerated, while lossy methods lose some detail. Lossless compression is important for data or document-based applications. However,

present videophone systems cannot support broadcast quality and loss is inevitable.

Some compression techniques require long processing times because they compare successive frames. While this does not affect non real-time applications such as movie distribution, the processing delay is not acceptable for videophone.

Home-based movie applications require the end-user to be able to scroll forwards or backwards through the movie. This is possible by placing the signal on a hard drive prior to viewing.

Algorithm compression and decompression times are not always equal. Thus, time consuming, high compression techniques that are suitable for video CDs are not suitable for videophone.

The nature of compression algorithms is also dependent on the amount of movement in an image.

There is also the question of where the compression algorithm is implemented. In computer-based applications, it can be implemented in either software or on dedicated hardware.

## 6.2 Audio Compression

### 6.2.1 ADPCM

[MXCOM - CVSD Modulation Tutorial](#)

[Harris - Delta Modulation for Voice](#)

[Harris - Introduction to Sigma Delta](#)

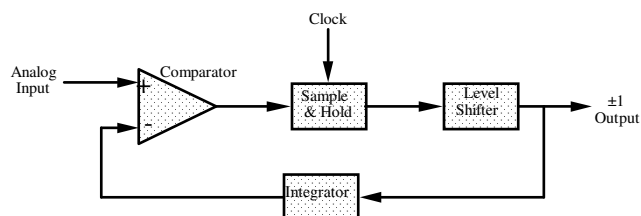
[Harris - H17190 Eval Kit](#)

[KineticSystems - Sigma Delta Converter](#)

[TI - Sigma Delta Converter](#)



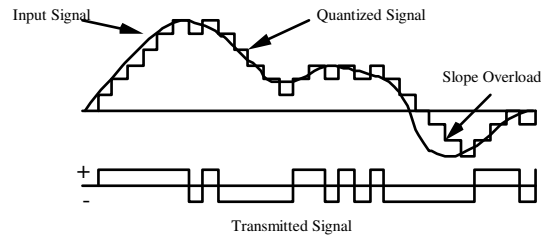
ADPCM<sup>†</sup> is used to encode all sounds including speech. Instantaneous sound levels are always changing since there is no such thing as a dc frequency. However, since sound occurs at relatively low frequencies, the actual change between any two adjacent samples is generally small.



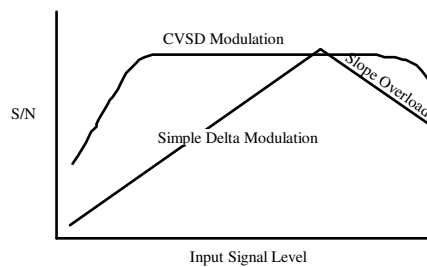
This circuit is sometimes referred to as a 1-bit DAC, and is used in audio CDs.

<sup>†</sup> Adaptive Differential Pulse Code Modulation

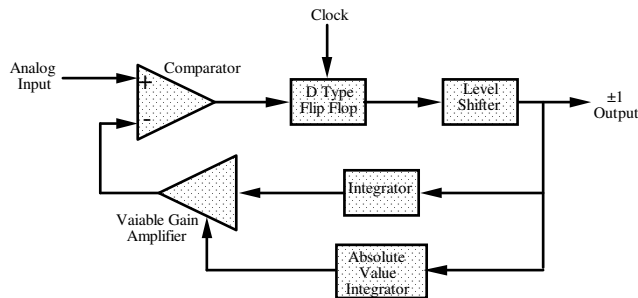
Output Waveform



Since the differences between samples is small, they can be encoded with fewer bits. This can provide a compression of 4:1. If however, the signal changes faster than the coding algorithm, a condition called slope overload occurs, and the signal is severely distorted.



In a simple delta modulator, each output pulse has the same quantization level. In ADPCM the step size is dynamically adjusted, thus reducing the risk of slope overload.



Companding tends to keep the signal to noise ratio more or less constant over a wide range of input signal levels.

6.2.2 MP3

[ATSC - Digital Audio Compression Standard](#)

[Sony - ATRAC](#)



MP3 stands for Mpeg 1 Audio Layer 3.

It uses a psycho-acoustic model, which recognizes that the human ear cannot perceive all audio frequencies. Human hearing ranges from 20 Hz to 20 KHz and is most sensitive from 2 to 4 KHz. MP3 compression removes the frequency components that cannot be heard. Therefore, it is a lossy compression method.

MPEG1 defines three audio modes layers I, II, and III, each with increased compression and encoding complexity.

Layer I is the simplest layer that is suitable for consumer use. It features sub-band filtering with 32 equal-width bands, adaptive bit allocation and block companding. Bit rates range from 32 Kbps (mono) up to 448 Kbps (stereo). Depending on the complexity of the encoder, a high (near CD) audio quality requires a bit rate of about 256 - 384 Kbps per stereo program. The complexity of the decoder is low, the encoder complexity is about 1.5 to 3 times as high. Layer I is used in DCC and for Solid State Audio.

Layer II has numerous applications in both consumer and professional audio, such as audio broadcasting, television, telecommunications, and multi-media. Bit rates range from 32 - 192 Kbps for mono, and from 64 - 384 Kbps for stereo.

Layer III extends MPEG1 applications into N-ISDN telecommunications and certain specialist areas of professional audio. MPEG compression can shrink data by a factor of 12 and still provide CD-like quality.

## 6.3 Bit Stream Compression

### 6.3.1 Entropy

Entropy is the average amount of information present in a source and is defined in terms of probabilities. Entropy encoding reduces the bit rate by minimizing or eliminating wasted states. The result is a compressed bit stream that more closely approximates the actual information content.

Two methods used to improve transmission efficiency are Huffman and Shannon-Fano coding. Both of these methods use variable length codes. The events that are more likely to occur are given short codes while the least likely events are given the longest codes.

### 6.3.2 Huffman Code

<http://www.compressconsult.com/huffman/>

<http://www.image.cityu.edu.hk/~loben/thesis/node13.html>

<http://www.dogma.net/DataCompression/Huffman.shtml>

<http://www.data-compression.com/lossless.html> (see animation below)

http://

The code is determined by creating an encoding tree, using the following steps:

- Arrange the messages in decreasing probability of occurrence
- Combine the two branches having the lowest probability into a single branch
- Assign the value 0 to the higher probability branch and 1 to the lower probability branch. In the event that 2 branches have the same probability, assign 0 to the top branch

[Huffman Coding Animation](#)

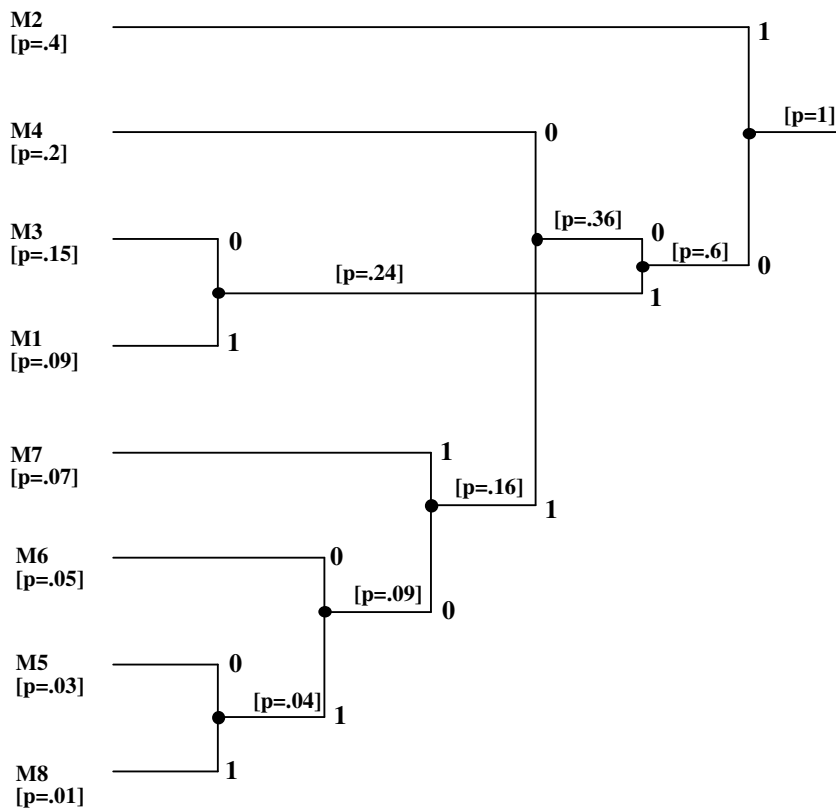
The code value for each message is found by retracing the path back along the tree. For a given distribution, there may be several Huffman codes, but the overall compressed length remains the same. This is the technique used in pkzip, lha, zoo, and arj.

Encoding Example

Suppose there were 8 different messages, with varying probabilities of occurring:

Message or Event	Probability of Occurrence	Binary Representation
M1	.09	001
M2	.40	010
M3	.15	011
M4	.20	100
M5	.03	101
M6	.05	110
M7	.07	111
M8	.01	000

Following the rules of Huffman coding, we can rearrange the message sequence and create the following tree:



[Huffman Coding Animation](#)

## Resulting Huffman Code Table

Message or Event	Probability of Occurrence	Binary Representation	Huffman Code
M2	0.40	010	1
M4	0.20	100	000
M3	0.15	011	010
M1	0.09	001	011
M7	0.07	111	0011
M6	0.05	110	00100
M5	0.03	101	001010
M8	0.01	000	001011

Although this code would appear to be less efficient than the standard binary format, it actually will result in 20% fewer bits being transmitted. This can be observed by calculating the entropy for the messages in question and the comparing it to the number of bits sent per encoded message.

The average number of bits transmitted is the sum of the individual message probabilities, times the number of bits per message:

$$.4 \times 1 + .2 \times 3 + .15 \times 3 + .09 \times 3 + .07 \times 4 + .05 \times 5 + .03 \times 6 + .01 \times 6 = 2.49 \text{ bits}$$

Notice that on the average 2.49 bits are required to send each message, rather than the 3 bits used in standard binary coding.

The Huffman scheme requires a backward operation. This means that all of the probabilities must be known before the codes can be assigned. This technique requires a lot of memory and processing time. It is therefore often performed by a dedicated processor.

The Huffman procedure produces the smallest average codeword length. Although other techniques can do as well, none can do better.

### 6.3.3 Shannon–Fano Code

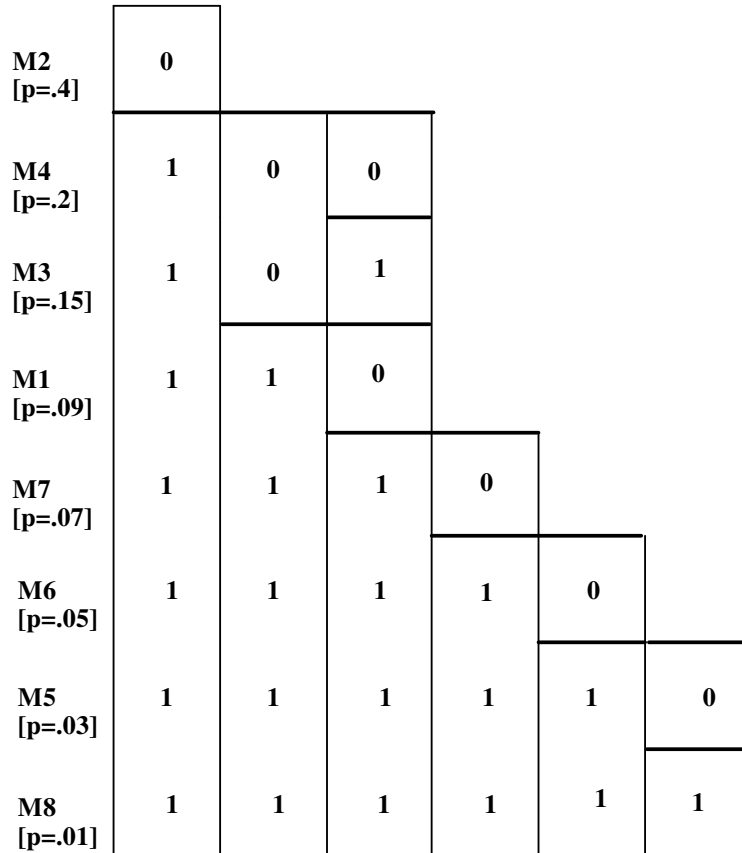
This process can be done in a forward mode, thus reducing memory requirements and processing time. However, it does not always result in the same degree of compression as the Huffman process.

One way the code can be determined is by the following procedure:

- Arrange the messages in decreasing probability of occurrence
- Divide the messages into 2 equally probable groups
- If there are two possible divisions, select the one with the fewest states above the division
- Assign one group a value of 0 and the other a 1
- Continue the division process until all there is nothing left to divide

Using the same example as before, we can acquire the Shannon-Fano code as follows:





[Shannon-Fano Coding Animation](#)

Resulting Shannon-Fano Code Table

Message or Event	Probability of Occurrence	Binary Representation	Shannon-Fano Code
M2	0.4	010	0
M4	0.2	100	100
M3	0.15	011	101
M1	0.09	001	110
M7	0.07	111	1110
M6	0.05	110	11110
M5	0.03	101	111110
M8	0.01	000	111111

On the average, this code will transmit 2.5 bits per message:

$$.4 \times 1 + .2 \times 3 + .15 \times 3 + .09 \times 3 + .07 \times 4 + .05 \times 5 + .03 \times 6 + .01 \times 6 = 2.49 \text{ bits}$$

This coding algorithm is often less efficient than that of Huffman, but can be implemented with simpler hardware.

The Huffman and Shannon-Fano coding techniques are very effective if the symbol probabilities are integral powers of 1/2. However, this is often not the case. A more sophisticated compression technique is arithmetic coding.

### 6.3.4 Run-Length Coding

RLE takes advantage of the fact that many times adjacent data blocks or pixels will have the same value. A special escape code is inserted into the data stream to indicate that the following data or pixel value is to be repeated a certain number of times. The escape code must be a value that cannot possibly occur naturally in the data stream.

As an example, let's assume an 8-bit code and use an escape code of 255. In decimal notation, a data string may resemble:

12, 14, 22, 33, 85, 85, 85, 85, 85, 85, 85, 12, 16, 22 ...

This data string can be compressed by replacing the value 85 by the escape code, the value, and the number of times the value occurred:

12, 14, 22, 33, 255, 85, 7, 12, 16, 22 ...

This technique does not work for audio signals since they do not remain constant.

### 6.3.5 Arithmetic Coding

This technique is based on the idea that the probability of a bit occurring is uniformly distributed throughout the byte. The overall probability of a 1 or 0 occurring is determined. This value is then used in each bit position and used to determine the probability of any particular sequence. The final code word is determined by finding the nearest binary fractional equivalent.

This process can be enhanced by considering intersymbol probabilities. An example of this is dynamic Markov coding.

### 6.3.6 Lempel-Ziv Coding

<http://www.data-compression.com/lempelziv.html>

[Lempel-Ziv Animation](#)

Encoding Algorithm

1. Initialize the dictionary to contain all blocks of length one ( $D=\{a,b\}$ ).
2. Search for the longest block  $W$  which has appeared in the dictionary.
3. Encode  $W$  by its index in the dictionary.
4. Add  $W$  followed by the first symbol of the next block to the dictionary.
5. Go to Step 2.

## 6.4 Image Compression

<http://www.data-compression.com/image.html>

### 6.4.1 JPEG

The JPEG standard was developed by the ISO/IEC primarily for CD still images. It can be implemented in hardware or software to compress image sequences. It supports four encoding modes:

- Sequential - each image component is encoded as scanned
- Progressive - the image is encoded in several passes, each adding more detail
- Lossless - produces an exact replica
- Hierarchical - the image is encoded at several resolutions

The lossy methods use the DCT while the lossless method uses predictive coding. Each is followed by statistical [entropy] encoding. The image compression factor is 20:1 to 30:1.

#### 6.4.1.1 The Gibbs Effect

One of the most common artifacts that afflict MPEG and JPEG compression is the Gibbs effect. This is most noticeable around artificial objects such as plain colored, large text and geometric shapes. It shows up as a blurring or haze around the object with a sudden transition. It is caused by the discrete cosine transform. This phenomenon is also apparent around more natural shapes like a human figure. The area of the background around the subject appears to shimmer as the subject moves slightly.



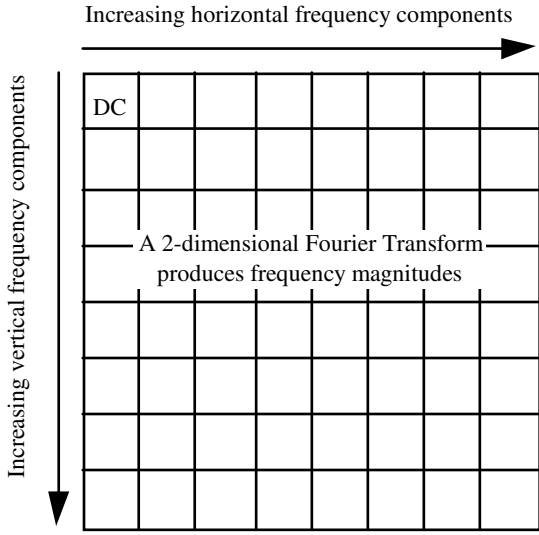
#### 6.4.1.2 Blockiness

When video footage involving high-speed motion is digitized, the individual 8x8 blocks that make up the picture become more pronounced.

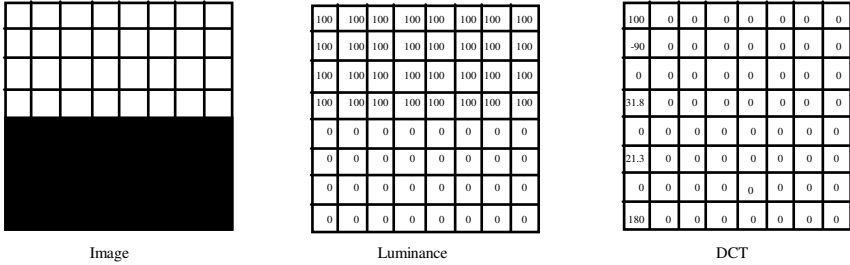
### 6.4.2 Discrete Cosine Transform

The DCT is similar to the Fourier Transform, in that the coefficients produced are the amplitudes of frequency components.

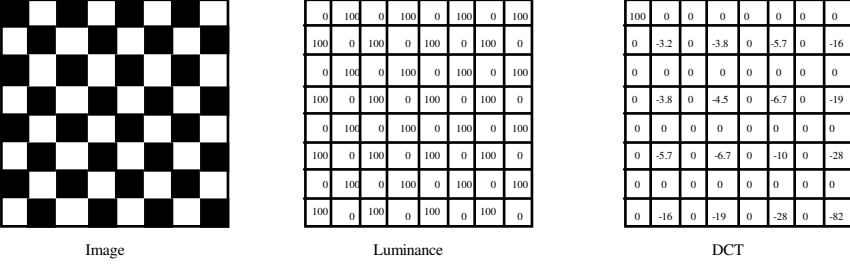
The image is decomposed into blocks of 8x8 pixels. Each of these is assigned an 8-bit digital value corresponding to its luminance. Then a two-dimensional Fourier Transform is performed on the block. This spatial frequency represents how much the luminance has changed between adjacent pixels. The first segment is given the average luminance or DC value. The coefficients corresponding to increasing frequency in the horizontal direction are arranged from left to right and those corresponding to increasing frequency in the vertical direction are arranged from top to bottom.



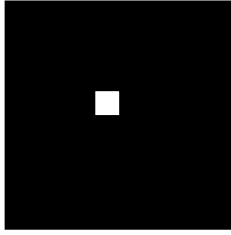
This two-stage process can be seen in the following example:



It is interesting to note that the picture with the most rapidly changing pixels has more than half of its DCT coefficients go to zero. This is expected since the Fourier coefficients of half the frequency components of a square wave are also zero.



It is interesting to observe that a single pixel element in an otherwise uniform screen generates a large number of coefficients. Likewise, the Fourier Transform of single pulse contains numerous harmonic components.



Image

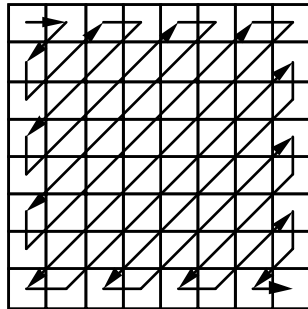
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	100	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Luminance

3.1	0.9	-4.1	-2.5	3.1	3.7	-1.7	-4.3
0.9	0.2	-1.1	-0.7	0.9	1	-0.5	-1.1
-4.1	-1.1	5.3	3.2	-4.1	-4.8	2.2	5.7
-2.5	-0.7	3.2	1.9	-2.5	2.9	1.3	3.4
3.1	0.9	-4.1	-2.5	3.1	3.7	-1.7	-4.3
3.7	1	-4.8	2.9	4.3	4.3	-1.9	-5.1
-1.7	-0.5	2.2	1.3	-1.9	-1.9	0.9	2.3
-4.3	-1.1	5.7	3.4	-5.1	-5.1	2.3	6

DCT

Each pixel is encoded into 8 bits. Each block therefore contains 512 bits. However since many blocks will contain long strings of zeros, entropy coding can be used to reduce the total number of bits to be sent.



This process produces coefficients that are often arranged in ascending frequency but decreasing amplitude. From here, the quantizer assigns weighting factors to each of the four frequency bands. The higher magnitude, low frequency components are given the most weighting since they are the ones most readily perceived by the eye.

## 6.5 Video Compression

One of the best ways to compress video images is to use motion compensation. This technique is the picture equivalent of ADPCM. A reference frame containing the first image is produced. All subsequent frames are simply the changes from one frame to the next.

This unfortunately requires a great deal of processing and makes editing very difficult. It also allows errors to propagate.

### 6.5.1 MPEG

[MPEG Technical Overview by C-Cube](#)

[MPEG Guide by Tektronics](#)

[Digital Television MHEG-5 Specification](#)



The Motion Pictures Expert Group was established in 1988 to develop new code standards for audio and video signals. The committee's formal name, WG11 of SC29, is a joint technical committee of ISO/IEC.

The formal name of this standard is: *ISO/IEC Standard, Coded Representation of Picture, Audio and Multimedia/hypermedia Information, ISO 11172*. The standard consists of five parts: systems, video, audio, conformance testing, and software simulation.

The standard is flexible enough that it can be implemented in software or on PC plug-in cards.

The main purpose of MPEG-1 is to compresses a full motion audio/video signal to about 1 to 1.5 Mbps so that it can be stored on a compact disk. It is very similar to the CCITT Recommendation H.261. MPEG is a generic standard that can be adapted to meet the needs of any number of applications. Some of the features, which were included in the design requirements, include:

- Support of symmetrical/asymmetrical applications
- Random access
- Fast forward/reverse searches
- Reverse playback
- Audio/video synchronization
- Error robustness
- Edibility
- Support various display resolutions/sizes/rates
- Chip implementable

Random access and edibility can be achieved by intraframe coding. Unfortunately, this does not provide a great deal of compression, since most redundant information is found in adjacent frames. This suggests that a combination of inter and intra frame coding is needed to achieve high compression and random access.

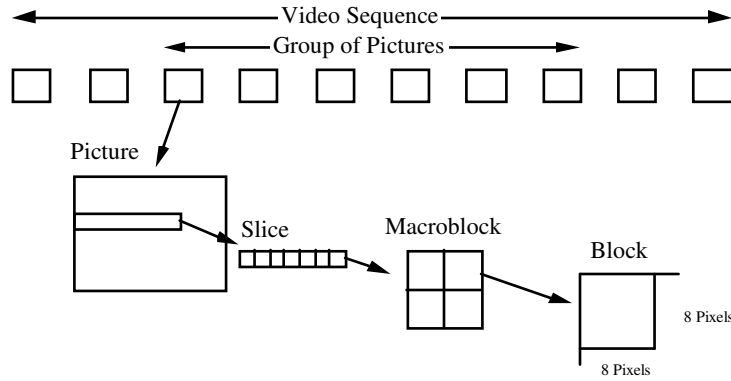
Much of the following information was obtained from [www.c-cube.com](http://www.c-cube.com)



[C-Cube - MPEG Technical Overview](#)

#### 6.5.1.1 Video Stream Data Hierarchy

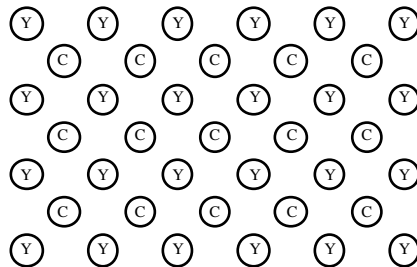
The MPEG standard defines the data structures in the video stream



A sequence begins with a header, contains one or more groups of pictures, and ends with an end-of-sequence code.

A group of pictures contains a header and a series of one or more pictures intended to allow random access into the sequence.

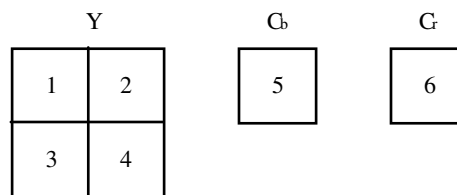
A picture consists of three rectangular matrices representing luminance (Y) and two chrominance (Cb and Cr) values. The Y matrix has an even number of rows and columns. The Cb and Cr matrices are one-half the size of the Y matrix in each direction (horizontal and vertical).



One or more 'contiguous' macroblocks is called a slice. The order of the macroblocks within a slice is from left-to-right and top-to-bottom.

Slices are important in the handling of errors. If the bit stream contains an error, the decoder can skip to the start of the next slice. Having more slices in the bit stream allows better error concealment, but uses bits that could otherwise be used to improve picture quality.

A macroblock is a 16-pixel by 16-line section of luminance components and the corresponding 8-pixel by 8-line section of the two chrominance components. A macroblock contains four Y blocks, one Cb block and one Cr block. The numbers correspond to the ordering of the blocks in the data stream, with block 1 first.



A block is an 8-pixel by 8-line set of values of a luminance or a chrominance component. Note that a luminance block corresponds to one-fourth as large a portion of the displayed image as does a chrominance block.

### 6.5.1.2 Picture Types

Two block based motion compensated interframe coding techniques are used to take advantage of temporal redundancy, namely: predictive [P pictures, causal] and interpolative [B pictures, noncausal] coding. Additional compression is obtained by using the DCT. The more the movement in the picture, the more difficult it is to perform interframe compression.

This standard defines four different types of pictures:

- I pictures - These independent pictures are encoded without reference to any other image and are similar to JPEG pictures.
- P pictures - Predicted pictures use motion compensation from previous I or P pictures and require about 1/3 of the bits of an I picture.
- B pictures - These are pictures interpolated between a past and future pictures. In order to decode a B picture, the future I or P picture must be transmitted before the predicted B picture. This naturally creates a great deal of latency, however, it provides the greatest amount of compression.
- D pictures - This format is used to support a fast search mode.

The MPEG standard takes advantage of temporal redundancy by inter-picture coding. Some pictures are defined as differences from a reference picture.

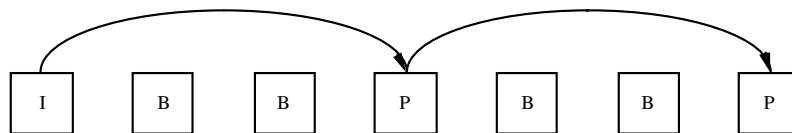
#### I-pictures

Independent pictures are complete and allow random access points into a compressed video stream. They can be compressed by the DCT to about 2-bits per pixel.

The MPEG algorithm allows the encoder to determine how many I-pictures are required. I-pictures typically occur twice a second in applications requiring random access.

#### P-pictures

Predicted or P-pictures, are coded with respect to the nearest previous I or P-picture. This technique is called forward prediction.

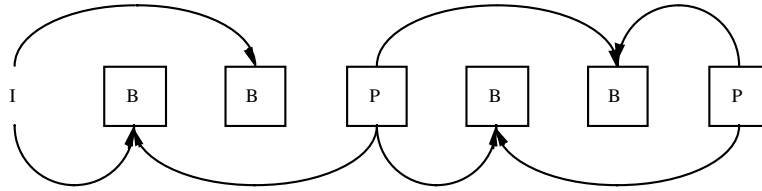


Like I-pictures, P-pictures serve as a prediction reference for B-pictures and future P-pictures. However, P-pictures use motion compensation to increase compression. P-pictures can propagate coding errors because they are predicted from previous reference (I or P) pictures.

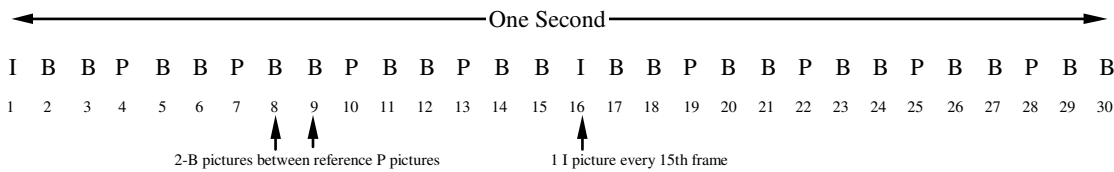


**B-pictures**

Bidirectional or B-pictures use both a past and future picture as a reference [bidirectional prediction] and provides the bulk of video compression. Bidirectional prediction decreases the effect of noise by averaging two pictures.

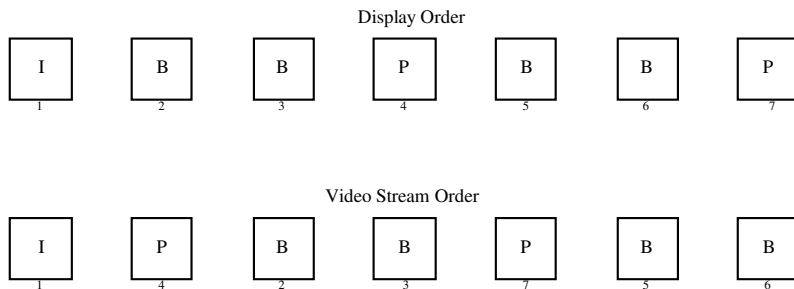


The encoder memory and picture dynamics determine the number of B-pictures transmitted. A large class of scenes however, has two bidirectional pictures between successive reference pictures.



**Picture Sequencing**

The MPEG encoder reorders pictures to simplify the decoding process. In particular, the reference pictures needed to reconstruct B-pictures are sent before the associated B-pictures.



**Motion Compensation**

Motion compensation minimizes temporal redundancy at the macroblock level and enhances the compression of P and B-pictures. It improves the compression by about a factor of three over intra-picture coding.

A motion compensated compressed macroblock contains:

- A motion spatial vector between the reference macroblock and the macroblock being coded
- The content differences or errors between the reference macroblock(s) and the macroblock being coded.

Not all information in a picture can be predicted from a previous picture. In such situations, a macroblock in a P-picture cannot be efficiently represented by

motion compensation. Instead, it is coded in the same way as a macroblock in an I-picture using transform coding techniques.

Macroblocks in a P-picture use the previous reference (I or P-picture), while macroblocks in a B-picture are coded using any combination of a previous or future reference picture.

There are four way to encode a B-picture macroblock:

- Intra coding: no motion compensation
- Forward prediction: the previous reference picture is used as a reference
- Backward prediction: the next picture is used as a reference
- Bidirectional prediction: two reference pictures are used, the previous reference picture and the next reference picture

Backward prediction can be used to predict uncovered areas that do not appear in previous pictures.

MPEG provides an average compression ratio of about 50:1. The CD-ROM version of this standard is comparable to a domestic VCR, has a bit rate of 1.2 Mbps, displays 30 frames/sec, and has a resolution of 352 x 240 pixels. The audio is encoded by means of ADPCM and interleaved with the video.

#### Audio Data Stream

The MPEG audio stream consists of a series of packets. Each audio packet contains an audio packet header and one or more audio frames.

Each audio packet header contains:

- Packet start code
- Packet length - Indicates the number of bytes in the audio packet.

An audio frame contains:

- Audio frame header - Contains synchronization, ID, bit rate, and sampling frequency information
- Error-checking code - Contains error-checking information
- Audio data - Contains information used to reconstruct the sampled audio data.
- Ancillary data - Contains user-defined data.

#### 6.5.1.3 MPEG-2

The official title of this standard is: *Generic Coding of Moving Pictures and Audio, ISO/IEC 13818*. It extends the functionality of MPEG-1 to broadcast applications. This necessitates increasing the bit rate to about 10 Mbps.

This system must also be able to handle picture interlacing. This presents a significant complication over MPEG-1

MPEG-3

This standard has been merged into MPEG-2

MPEG-4

This is very low bit rate audio/video coding.

## 6.6 Other Compression Techniques

### 6.6.1 Wavelets

Here are some links to wavelet sites:

<http://elecard.com/v1/wavelet.html>

[www.summus.com/](http://www.summus.com/)

[www.amara.com](http://www.amara.com) A wavelet software company

<http://www.wavelet.org/wavelet/index.html> Lucent technologies

<http://www.mathsoft.com/wavelets.html> Lots of links to other wavelet sites

<http://hp0tlc.polito.it/~dovis/fabio/wlet.html> Some wavelet links

<http://paos.colorado.edu/research/wavelets/>

http://

Here are some wavelet tutorials:

[Wavelet Overview](#) by Amara Graps

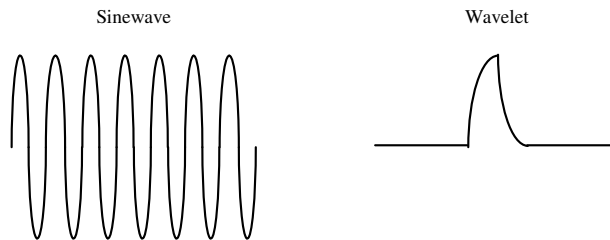
[A Practical Guide to Wavelet Analysis](#) by Torrance and Compo

<http://www.public.iastate.edu/~rpolikar/WAVELETS/WTtutorial.html>



http://

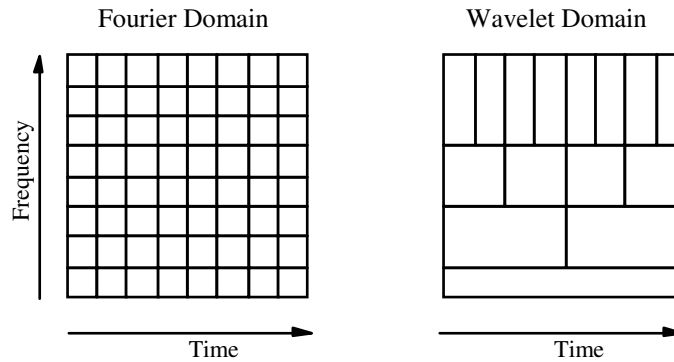
A wavelet is an oscillation that exists for a finite time. It has attributes that define its position and duration or scale. These types of waveforms are particularly useful in describing signals with discontinuities and can be multi dimensional.



Like sinewaves of increasing frequency, higher order (shorter) wavelets, can be used to describe increasing amount of detail. However, unlike sinewaves, the detail can be localized.

Single dimension wavelets can be used to analyze audio and other analog signals. Two-dimensional wavelets are useful in image processing, and three-dimensional wavelets can be used to compress moving images.

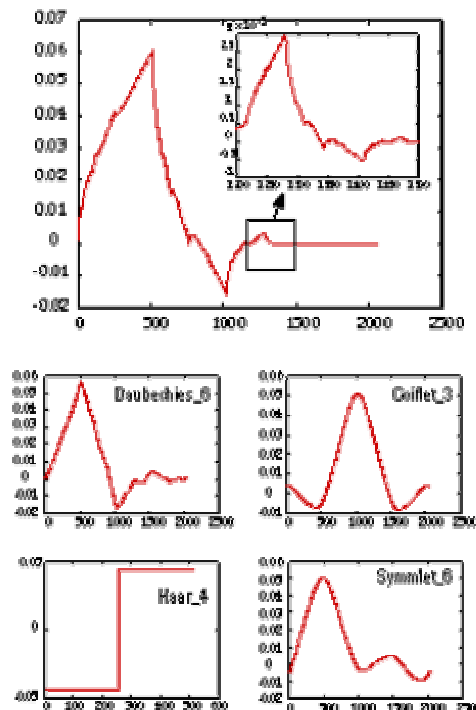
There is a similarity between wavelet analysis and Fourier analysis. The Fourier transform shows that signals can be decomposed into a series of sinewaves. These sinewaves however, are not localized in time and stretch to infinity. Therefore, when analyzing a signal, it is not possible to isolate a specific frequency component in time or space. Consequently, all frequencies must be examined simultaneously in order to perform Fourier analysis. Since wavelets are defined in both space and time, this restriction does not apply.



At any instant in time or space, a signal or image may be decomposed into a series of wavelets of different duration.

Wavelet compression is just as good as that provided by JPEG or MPEG. However, it has yet to be standardized and is currently limited to market niches.

Wavelets come in a wide range of shapes and sizes. Some even have a fractal structure, such as the, Daubechies wavelet family.



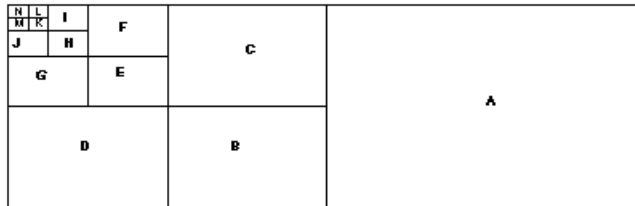
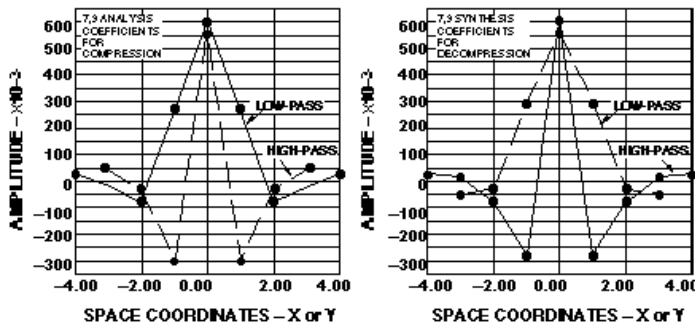
### 6.6.1.1 Wavelet Compression Codec

Wavelet codecs can be used for video capture, remote CCTV surveillance, camcorders, high quality teleconferencing and video distribution systems, video insertion equipment, image and video archival systems, and digital video tape applications.

The Analog Devices ADV601 codec compression algorithm is based on the bi-orthogonal wavelet transform, using 7-tap high-pass and 9-tap low-pass filters, and field-independent sub-band coding.

The sub-band coder transforms two-dimensional spatial video data into spatial frequency filtered sub-bands. Adjustable quantization and entropy encoding are used to provide compression.

The wavelet kernel contains filters and decimators that work on the image in both horizontal and vertical directions. The filters are based on segmented wavelet functions.



This process can be seen in the image compression of an office building.



The image has been transformed into 14 new images, each containing a different set of information about the original image.



### 6.6.2 Fractals

<http://www.htcomp.net/tfunk/Fractals.html> a very extensive link site

<http://math.rice.edu/~lanius/frac/>

<http://members.home.net/zsolt/mandelbrot/mandel.html>

<http://www.mhri.edu.au/~pdb/fractals/fracintro/>

http://

<http://www.gozen.demon.co.uk/godric/godric.html>

<http://library.thinkquest.org/12740/msie4/index.html>

<http://www.geocities.com/CapeCanaveral/2854/>

<http://spanky.triumf.ca/www/fractint/fractint.html>

<http://sprott.physics.wisc.edu/fractals.htm>



[An Introduction to Fractals](#) by Paul Bourke

[A Brief Introduction to Fractals](#) by Godric

Nature does not appear to be composed of the standard geometric shapes. Instead, it is full of discontinuities and irregularities. This observation led to the development of IFS<sup>†</sup>, more commonly known as fractals. If for example, one were to examine a cloud, one would find that it looked remarkably similar at all scales. This self-similarity has been used by Hollywood for years to create disasters in miniature, which are nearly indistinguishable from disasters at full scale.

Using relatively simple number generators, it is possible to create remarkable landscapes. This suggests that rather than trying to encode and compress an image, it may be possible to determine the mathematical expression [affine

---

<sup>†</sup> Iterated Functions Systems

transformation] which when iterated, creates the image. With this approach, compression ratios of 10,000:1 are achievable.

However, there is no known method for finding the affine transform for an arbitrary image.

Fractal compression is widely used computer animation, but not as much in standard image compression.

Although overall, everyday scenes do not obey fractal geometry, there are often similar features scattered throughout an image. By relating similar parts of the image together, image compression is possible. This process is called PIFS<sup>†</sup>.

The image is divided into large segments called domain blocks, and sub divided into range blocks. The range blocks are associated with the domain block that it most closely resembles. The term that would generate the fractal pattern must then be determined. It is transmitted along with the position of the associated block.

There are many different ways to implement PIFS. However, although the decompression process is very fast, the encoding compression process is computationally intensive. Using this technique to compress moving images, is probably unrealistic.

---

<sup>†</sup> Partial Iterated Functions Systems

## Assignment Questions



### QUICK QUIZ

1. In the DCT, the picture is decomposed into blocks of \_\_\_\_ x \_\_\_\_ pixels.
2. What type of coding is used by the DCT to reduce the number of zero bits?
3. Which picture in the MPEG resembled the JPEG?
4. Which MPEG standard applies to broadcast video?

### COMPOSITION QUESTIONS

1. What is inter-picture coding?
2. What are the essential characteristics of the I, P, and B pictures in MPEG?
3. What are the 4 ways to encode a B picture macroblock?
4. Do some research on the Mandelbrot set or Sierpinski-Gasket and explain how they are generated.
5. Explain why images, which have been compressed using a fractal process, may appear to contain infinite detail, but ultimately the detail will be meaningless.



## For Additional Research

---

Little Waves Big Squeeze, New Scientist, 2 March 1996,  
Wavelet Analysis, IEEE Spectrum, October 1996

### MPEG

[http://www.bbc.co.uk/rd/pubs/papers/paper\\_14/paper\\_14.htm](http://www.bbc.co.uk/rd/pubs/papers/paper_14/paper_14.htm)

<http://www.mpeg.org/index.html/>

<http://nebo.cs.byu.edu/~scoville/luigi/mpeg2.html>

<http://www.mpeg2.de/doc/luigi/mpeg2.htm>

<http://www.c-cube.com/technology/mpeg.html>

<http://bmrc.berkeley.edu/projects/mpeg/>

<http://www.crs4.it/~luigi/MPEG/mpegfaq.html>

<http://www.cinax.com/4MPEG/index.html>

<http://www.optibase.com/>

### Delta Modulation

<http://www.mxcom.com/appnotes.htm>

### Fractals

<http://www.deepleaf.com/fractal/>