I formati grafici

Davide Rossi

Table of contents

Part I
Colors and Color Systems
Still Images: Bitmaps, Vectors & Metafiles

Colors and Color Systems

The human eye can percept light frequencies in the range 380-770 nanometers and can distinguish about 10000 different color simultaneously.

The color the eye is more sensible to is the green, followed by red and blue.

In computer graphics we typically use a trichromatic colorimetric system.Depending on the device used these systems can be separated in two categories:

Additive
colors are added to black to create new colors; the more color is added, the more the resulting color tends towards white.
CRTs are additive.

Subtractive
colors are subtracted from white to create new colors; the more color is added, the more the resulting color tends towards black.
Printers are subtractive.

Color Spaces

RGB Red-Green-Blue is an additive color system. In a [0,1] color intensity range (0,0,0) is black, (1,1,1) is white.

CMY Cyan-Magenta-Yellow is a subtractive color system. (0,0,0) is white, (1,1,1) is black.

HSV Cyan-Magenta-Yellow is a subtractive color system. (0,0,0) is white, (1,1,1) is black.

YUV Luminance-Chrominance. Is a linear encoding of RGB used in television trasmission. Y contains Luminance (brightness) information; U and V are color information. (Similar color spaces are YCrCb and YPbPr0).

Displays and Colors

In a computer display the images are rendered by a grid of dots called ***pixels***.

The pixel grid is stored in an ad hoc memory of ***the Video Adapter*** usually referred to as ***Video RAM*** or ***Video Memory***.

Depending on the number of colors associated to each pixel, the amount of memory needed to contain the display data can be very different.If our display can only contain black and while pixels we can encode the video memory in such a way each byte represents 8 pixels. Thus a 1024x768grid can be stored in 98304 bytes. If the display can show 16777216simultaneous colors we need three bytes per pixel for a total amount of 2359296 bytes (i.e. 24 times more than the black and white case).

Usually, if the display adapter maps directly the video memory to RGB components, the memory can be arranged in such a way each pixels is encoded in two or three bytes (5-5-5, 5-6-5, 8-8-8 bits format) often referred to as ***hi-color*** and ***true-color*** modes, respectively.

Palettes

Mostly because of physical limitations of the output devices the number of colors that can be used simultaneously can be limited.

Suppose we have a video adapter that uses the RBG color space and is able to handle 256 levels of intensity range for each primary color.

This video adapter has a grid of 1024 * 768 pixels but only 1MByte of video memory; using three bytes per pixel is then impossible since we would need more than 2MByte. To solve this problem the device uses a color palette tostore 256 different colors encoded using three bytes each and uses each byte in the video memory as an index to select the color from the palette. This way only 787200 bytes of memory are needed but only 256 colors can be displayed simultaneously.

Bitmaps, Vectors & Metafiles

Depending on the use they are created for, the input devices they are generated by (digital cameras, scanners, etc), the output devices they are destined to (displays, printers, VCRs, plotters, etc), whether they are animated or not, images can be encoded using:

- Bitmap
- Vector
- Metafile
- Scene
- Animation
- Multimedia formats.

Still Images: Vectors

Vector images are built from mathematical descriptions of one or more ***image elements***. Vectors are in fact line segments defined by a starting point a direction and a length; usually not just simple vectors are used in the encoding of vector images but also curves, arcs and splines.

Using these simple components we can define complex geometrical shapes such as circles, rectangles, cubes and polyhedrons.

Vector images are then encoded using sequences of basic shapes and lines with their parameters (starting point, length, etc).

Vector images are useful to encode drawings, computer-generated images and,in general, each image that can easily be decomposed in simple geometrical shapes.

Editing Vector Images

Vector images can be edited by adding/removing shapes and by changing shapes parameters by applying ***transformations*** (such as scale, translation, etc).

It is important to remark that by applying transformations no information is lost: in fact we can always apply new transformations to restore the previous state of the image.

Vector Files

Vector Files are used to store elements and their parameters (and,optionally, their colors).

The structure of a Vector file is something like:

```
Header
Image Data
```

Where Image Data is a sequence of elements descriptions (possibly in text format) such as:

```
CIRCLE 40,100,100,BLUE;
LINE 200,50,200,80,RED;
```


Pros and Cons of Vector Formats

Advantages:

- Vector data can be easily scaled in order to accommodate the resolution of the output device.
- Vector Image files are often text files and can be easily edited.
- It is easy to convert a Vector Image to a Bitmap Image.
- Translate well to plotters.

Drawbacks:

- Vector cannot easily be used to encode extremely complex images (such as photographic images) where the contents vary on a dot-by-dot basis (but:fractal image compression)
- The rendering of a Vector Image may vary depending on the application used to display the image
- The rendering of an image may be slow (each element must be drawn individually and in sequence)

Still Images: Bitmaps

Bitmap images are generated by scanners, digital cameras (and few other devices) and are the ``natural'' formats for displays and printers.

Bitmap images are built by a grid of colors.

In a display the image is grid of pixels, in a printer is a grid of dots.

Depending on the capability of the device the pixels/dots can have from two colors to millions of colors.

Editing Bitmaps Images

Bitmap images can easily be edited using interactive or batch programs.

We can apply them filters, modify colors, edit small parts.

Usual operations include:

- Blur and Sharpen.
- Despeckling.
- Color correction.
- Brightness/Contrast adjustment.
- Touch up.

The drawback is that they don't scale well. If we shrink a bitmap image and then we enlarge it back to its original size, information is lost!

Bitmap Files

Bitmap files are used to store color grids. The dimension of the the grid is usually referred to as the ***size*** of the bitmap.

The color space used to encode the colors in the bitmap can be different among file formats and color can also be encoded using a palette.

The structure of a bitmap file is something like:

```
header
palette*
bitmap data
footer*
```

`*` marks optional sections

Headers

A header contains the data needed to reconstruct the original image bitmap such as:

```
File Identifier
File Version
Number of lines
Number of pixels per line
Number of bits per pixel
Compression type
Origin of the image
Comments
```


Pros and Cons of Bitmap Formats

Advantages:

- Easily encoded in array of bytes.
- Are produced by many input devices.
- Easy to edit.
- Translate well to grid output devices such as CRTs and printers.

Drawbacks:

- Large.
- They do not scale well (it is easy to lose information).

Still Images: Metafiles

Metafiles has been created to overcome platform- and device-dependence problems related to bitmap and vector formats.

A metafile can then contain both vector and bitmap information, but vectors and bitmaps are very different concepts and handling metafiles implies handling both formats adding complexity to the application.

In some case, however, having both a bitmap and a vector representation of the same image in a unique file can be useful as in the case of Encapsulated PostScript.

Pro and Cons:

Portability (often metafiles are ASCII text files).

Useful to store bitmap previews of vector images.

Files are large and complex.

Bitmap vs. Vectors

Converting images from one format to the other is troublesome and, also if the operation is archived with success, further issues must be considered.

Vectors to Bitmap

The operation is quite easy: the application has simply to render the vector image.

Bitmap to Vectors

The operation is troublesome: complex math algorithms come into play and, for complex images, they often fail!
The resulting image can be much bigger (as in the case of photographic images) and the rendering can take lot of time.

Do I have to use Bitmaps or Vectors, then?

It depends on the nature of the images and on the output device the final copy is destined to.

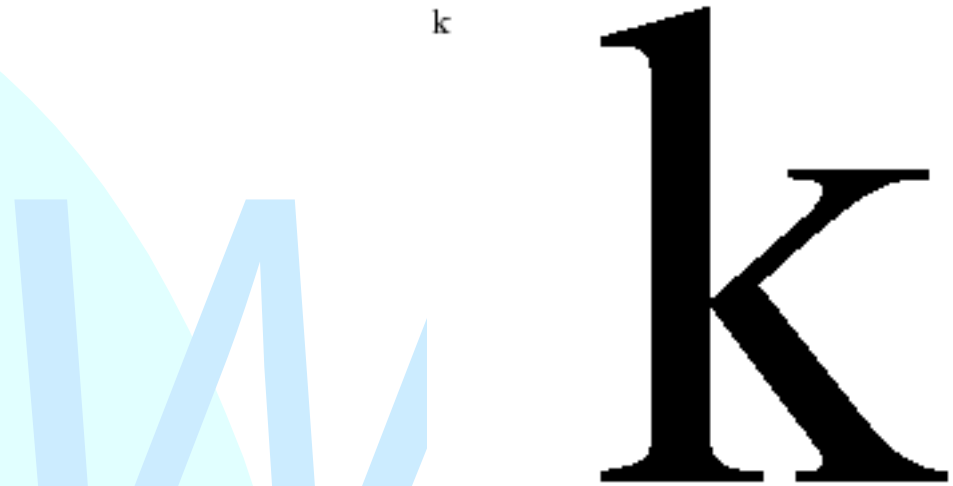
Just think about the characters in a text document. They can be encoded as both bitmaps or vectors. But, while bitmaps are probably good enough when you are looking at the document on your computer display, you will be disappointed by the appearance of the characters if you print the document using a high-quality printer.

Here the problem is that you display has a low resolution (typically 72-96 dpi) while your printer has a higher resolution (typically 300-600 dpi). To appear in the same size the characters sent to the printer have to be enlarged and their dot-based nature become more evident.

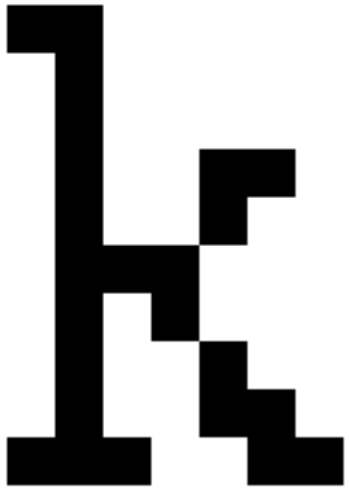
On the other hand vector formats cannot be used for photographic images and are not as well supported by applications as bitmap formats.

Bitmap & Vector Characters

k



k



Part II
Data Compression

Data Compression

As stated before one of the drawbacks of the bitmap format is that it need slots of memory to encode an image. This affects mostly the file size of a bitmap image and the time needed to transmit the image over a network.

A wide variety of data compression algorithm have been applied to bitmap images in order to reduce the resulting file size.

While conceptually every data compression algorithm may be used to compress a bitmap image we will see that some algorithm results more effective than others on image data.

Compression Terminology

Lossless/Lossy

- The first distinction we have to make about compression methods is whether they allow or not perfect data restoring (we say they are,respectively, ***lossless*** or ***lossy***)

Raw and Compressed Data

- We use these terms to refer to the original image data and to the compressed image data

Compression Ratio

- The ratio of raw data to compressed data

Symmetrical and Asymmetrical Compression

- When a compression algorithm uses roughly the same amount of work to archive both compression and decompression is said to be ***symmetrical***

Common Bitmap Compression Methods

Lossless methods

- Pixel Packing
- Run-Length Encoding (RLE)
- Lempel-Ziv(-Welch) Compression
- Huffman Encoding

Lossy methods

- DCT Compression (JPEG)
- Wavelet compression
- Fractal Compression

Compression: Pixel Packing

Pixel Packing is not a compression method per se: it is simply a convenient way to store the color data in a byte array. Suppose you have a palette-based,four color image. We can use one byte for each pixel but we could also encode the color information so that each byte is used to store four pixels by splitting the byte in four couples of bits.

Compression: Run-Length Encoding (RLE)

RLE is mostly useful when we have to deal with palette-based images that contain large sequences of equal colors.

The idea in RLE is in fact to encode long sequences of the same value with the shortest possible encoding.

A possible RLE encoding is the following:

each sequence in the file is a control number followed by a variable number of bytes.

If control number n is positive then the next n bytes are raw data; if n is negative then the next byte is repeated -n times in the raw data.

For example:
`45367777644457000011`
becomes
`4 4536 -4 7 1 6 -4 4 2 57 -4 0 -2 1`

RLE is used in the TARGA file format and in Windows Bitmap (.bmp) file format.

Compression: LZ77, LZW

LZ77 (Abraham Lempel, Jakob Ziv 1977) is a dictionary-based compression scheme and is the first of a set of similar data compressors often referred to as the LZ family.

In LZ compression substring are identified in the source data stream and are matched to entries in a dictionary.

If the substring is not already in the dictionary it is added to it with a newly generated index code and the index code is sent to the output.

If the substring is already in the dictionary its index code is sent to the output.

LZW (Terry Welch 1984) is a LZ compressor with a fixed entries size dictionary with a pre-initialized contents (256 entries with length one ranging from 0 to 255 on a total of 4096 entries).

When the dictionary is full an old entry is removed from it to make room for a new substring.

LZ77 is used in the Portable Network Graphics (.png) file format.

LZW is used in the Graphics Interchange Format (.gif) and in the Tagged Image File Format (.tiff).

Compression: Huffman Encoding

Huffman encoding is a well known encoding scheme based on statistical properties of the source data. Each code from the source is associated to a variable bit length code used in the output. Compression is archived by associating shorter output codes to more frequent input codes.

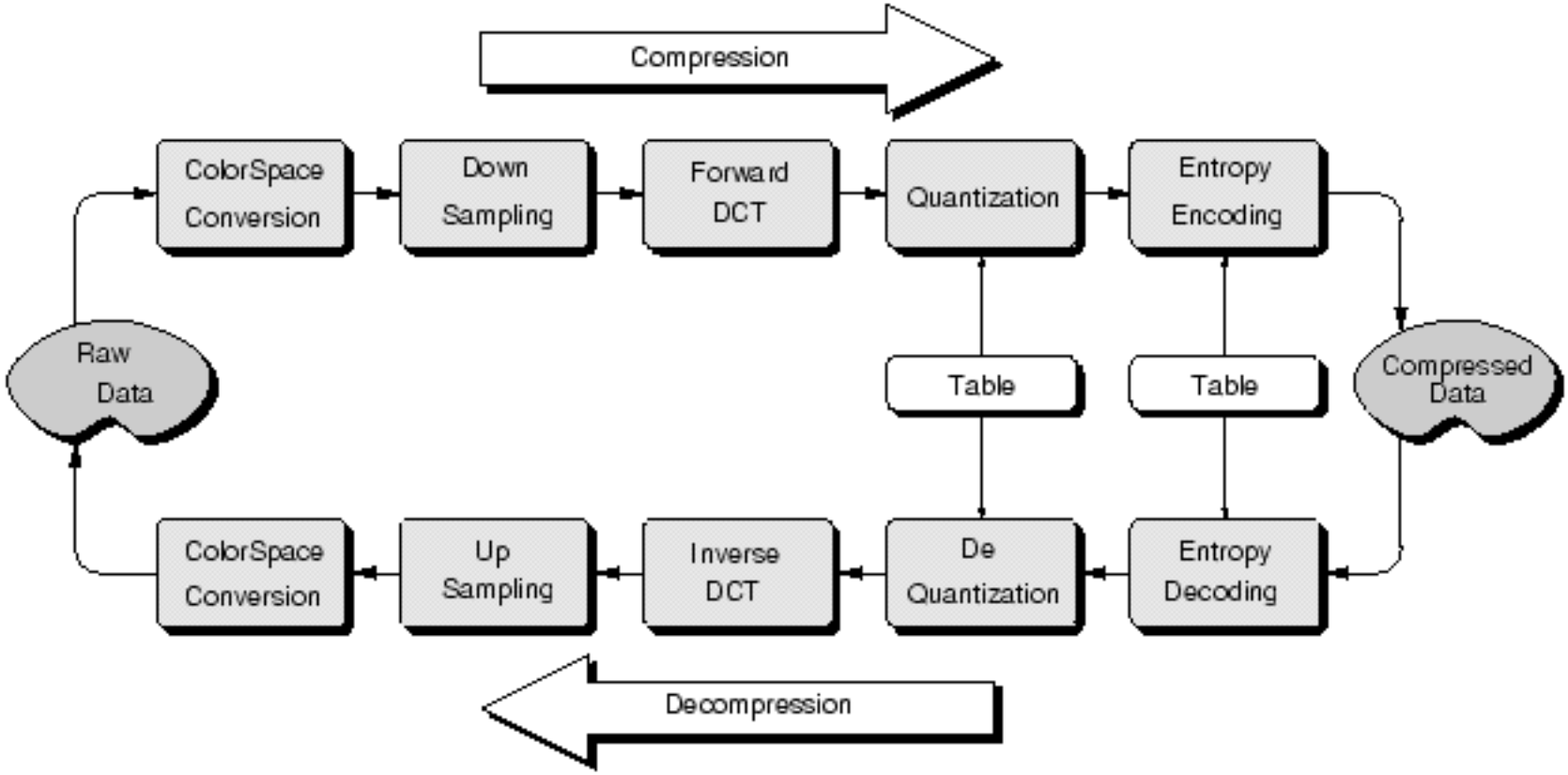
The association between input and output codes can be pre defined or calculated at run time.

Compression: JPEG Compression

The ***baseline*** JPEG (Joint Photographic Experts Group) compression (from now JPEG) is a lossy compression scheme based on color space conversion and discrete cosine transform (DCT).

JPEG works on true color (24 bits per pixel) continuous-tone images and achieves easily compression ratio of 25:1 with no visible loss of quality.

JPEG Encoding Flow Chart



Compression: Wavelet

Wavelet compression is similar (in principle) to JPEG compression. The main difference is the use of wavelet based techniques in place of DCT-IDCT transformations.

Wavelets are mathematical functions that cut up data into different frequency components. They have advantages over traditional Fourier and DCT methods in analizing signals that have discontinuities and spikes.

Comparative researches indicate that wavelet compression is slightly better than DCT-based JPEG but compression and decompression times are longer.

This compression technology is the compression technology used in the JPEG-2000 standard.

Compression: Fractal

Fractal compression is a very complex (lossy) compression technique.

It is based on the transformation of a bitmap image to a vector-like mathematical representation using ***iterated function systems*** (e.g.fractals).

Fractal compression is asymmetrical as the compression step is very much slower than decompression (decompression is, in fact, just a rendering algorithm) but there is a lot of work going on to overcome this problem.

The advantages of fractal compression are the big compression ratio that can be archived with little degradation of the image quality and the ability(just like with vector formats) to scale the image without losing information and without adding noise.

The drawback is that not everyone agrees on the advantages.

Notes on using lossy compression

It should be noted that all the lossy compression schemes are always lossy: a decompressed image is never the same as the original one.

This means that re-compressing a JPEG compressed image results in added information lost so lossy compression is never a good choice for intermediate storage.

Part III
Still Graphics File Formats
The GIF 87a File Format

Still Graphics File Formats
The GIF 87a File Format

The GIF87a (Graphics Interchange Format)file format is useful for storing palette based images with a maximum of 256 colors.

The compression technique adopted by the GIF format is LZW so it is possible to achieve high compression ratios only with non-photographic images.

Within a single GIF file multiple images can be stored (with their own palettes called local color tables).

Since LZW is a quite simple compression scheme it is quite easy to write a GIF decoder and this has lead to a wide adoption of this format among different applications

The GIF 87a File Format (2)

Images can be stored in a GIF file using the interleaving format: images line are not stored sequentially in a top-bottom order but using the following scheme:

```
0
3
2
3
1
3
2
3
```


The GIF 89a File Format

GIF 89a is an extension of the GIF 87a file format.

If GIF89a we have ***Control Extension blocks*** that can be used to render the multiple images in the same file in a multimedia presentation.

Control Extension blocks include Graphics Control Extension (how to display images),Plain Text Extension (text that have to be overlapped to the image),Comment Extension (human readable comments)and Application Extension (proprietary application information).

Since images could overlap during the rendering it is possible to define a palette index that is rendered as ***transparent***.

The JFIF File Format

The JFIF (JPEG File Interchange Format)format is the standard file format adopted for JPEG compressed images.

A JFIF file is composed by segments identified by markers.

An optional segment in the file can contain a thumbnail of the image in uncompressed RGB format.

The JFIF format does not allow the storage of multiple images in the same file.

JFIF supports progressive JPEG encoded images: the decoder returns a set of images progressively close to the original image.

The PNG File Format

The PNG format has been designed by the internet community to overcome patenting issues related to the use of LZW compression in GIF files.

PNG uses in fact a patented-free version of LZ encoding that archives higher compression ration than LZW.

Here is a (incomplete) list of improvements of PNG w.r.t. GIF:

- support for true-color images
- support for alpha channels
- 16 bits for channel optional accuracy

The FlashPix File Format

FlashPix is a still file format developed by Eastman Kodak, Hewlett-Packard, LivePicture and MicroSoft.

Images are stored in a per-tile basis (a tile is a small rectangular area).Each tile can be stored in compressed (using JPEG) or uncompressed format.

Each image is stored in a hierarchical structure so multiple version of the same image, at different resolutions, are stored within the same FlashPix file in order to allow viewing/editing on reasonably small images, images can also be substituted by (proprietary) links.

The file structure of a FlashPix picture is complex and is based on MicroSoft OLE's structured storage.

Pros: ?

Cons: proprietary storage format, non-standard link format, missing lossless compression, etc.

Part IV
Animation & Multimedia
The MPEG Motion Image Compression

Animation & Multimedia
The MPEG Motion Image Compression

MPEG is a compression scheme for motion images and audio developed by the Motion Picture Expert Group committee. Its image compression scheme is based on the DCT and is quite similar to JPEG.

The main difference between JPEG and MPEG is the usage of motion-compensation techniques to archive higher compression ratios.

A MPEG video stream is a sequence of I (Intra), P (Predicted) and B (Bi-directional) frames.

The MPEG Motion Image Compression (2)

I-frames are encoded using only information from the original frame;their encoding scheme is very similar to that used in baseline JPEG.

P-frames contain motion-compensated information w.r.t. the previous I- or P-frame. The image in decomposed in macroblocks (16 by 16 pixels); each macroblock is enocoded either as new or as moved from a given position. Each moved macroblock has an associated 8x8 error block. The encoding of a moved macroblock is represented by a motion vector and the error block.

B-frames contain motion-compensated information w.r.t. the previous I- or P- frame and the next I- or P-frame.

The MPEG Motion Image Compression (3)

While MPEG techniques allows for a good compression ratio it turns out that to decode P-frames we have to store in memory a previously decoded I- or P-frame, and to decode B-frame we have also to decode frames that come later in the input stream.

A typical sequence of a MPEG stream looks like: `IBBPBBPBBPBBIBBPBBPBBPBBI`

Typically I-frames recur every 12 frames in order to allow re-synchronization and to avoid error propagation.It should also be noted that the usage of B- and P-frames implies that MPEG is an asymmetrical compression scheme.

References: `http://www.mpeg.org`

The AVI File Format

AVI (Audio Video Interleaved) is a general purpose file format introduced by MicroSoft in the context of RIFF (Resource Interchange File Format).

AVI does not introduce new technologies, it simply defines a file format to store audio/video information that can be compressed using different schemes (e.g. the popular INDEO compression technology from Intel).INDEO is a video compression technology that uses a hierarchical image decomposition: the image is decomposed in smaller areas until the contents of each area can be considered as uniform. Motion compensation techniques among uniform areas are used to achieve higher compression ratios.

The QuickTime File Format

Yes, there is also a QuickTime file format.

The MPEG Audio Compression

MPEG Layer1, Layer2, Layer3 and AAC are audio compression schemes based on psychoacoustic models.Technically speaking Layer1 and Layer2 are based on subband coding whileLayer3 and AAC are based on hybrid (subband/transform) coding.The input signal is sampled at 32, 44.1 or 48 kHz.Typical bit rates for the 4 compression systems are:

Layer1	32-448 kbps
Layer2	32-384 kbps
Layer3	32-320 kbps
AAC	32-192 kbps

MPEG audio uses monophonic, dual-phonic, stereo and joint-stereo models.

Psychoacoustics

Psychoacoustic models used in MPEG audio compression are based on:

ear sensitivity w.r.t. frequency

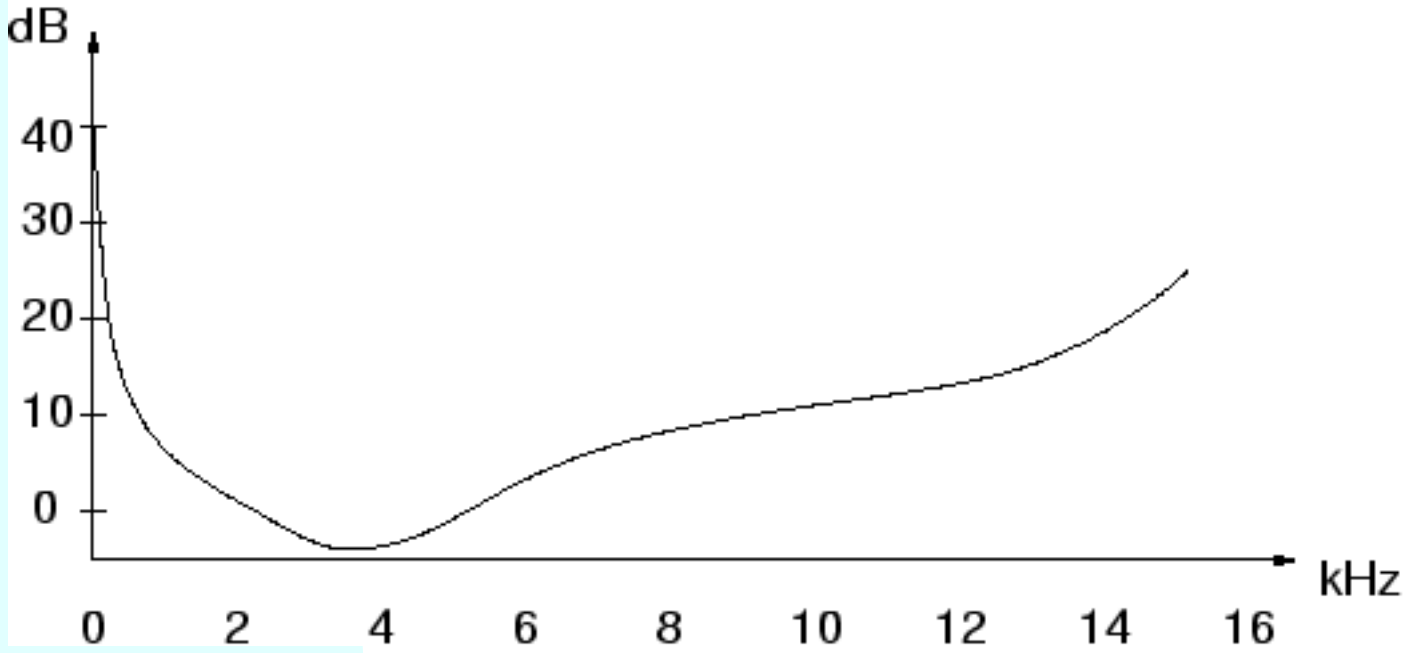
simultaneous frequency masking

temporal frequency masking

Ear Sensitivity and Frequency

The human ear is not equally sensible to signals at different frequencies.

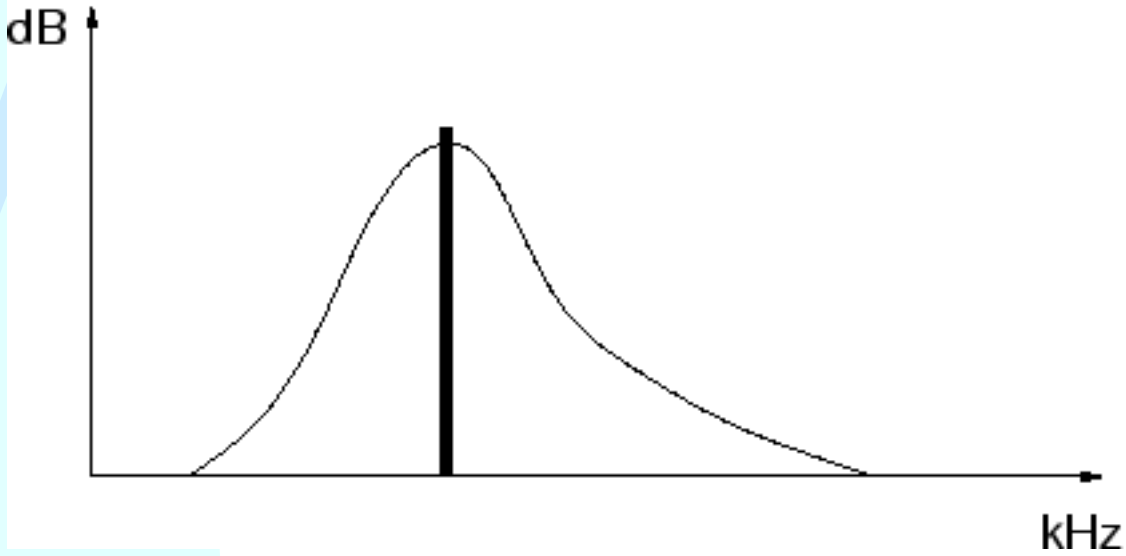
The diagram below plots the ear threshold in quiet.



Frequency Masking

High level tones at a given frequency mask lower tones at close frequencies.The masking band depends on the frequency of the masking signal.

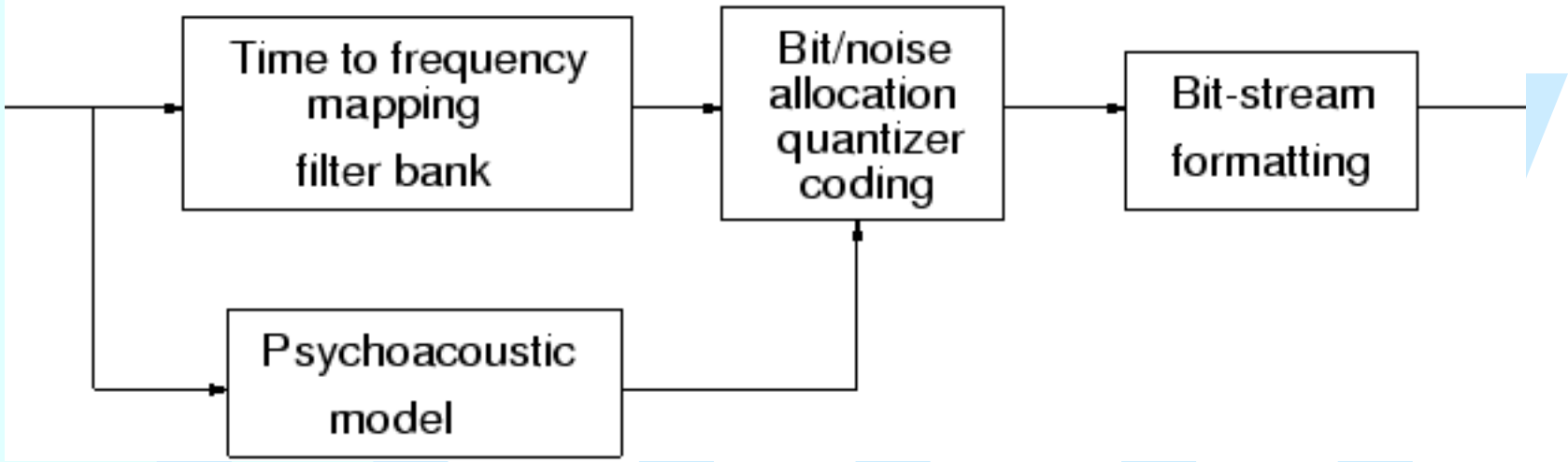
The diagram below plots the masking for a tone at about 2 kHz.



Steps in MPEG Audio Compression (1)

- time-to-frequency transformation (uses a polyphase filter bank)
- split tonal and non-tonal components
- calculate mask spreading function
- apply masking
- calculate signal-to-noise ration (SNR)
- choose quantization
- (layer 3 and AAC) use entropy encoder

Steps in MPEG Audio Compression (2)



Usage of the Psychoacoustic model

The filter bank outputs 32, equally-spaced, signal bands (note: the bands are overlapping; they should map the critical bands but they don't).The output of the filter bank is used to compute the masking frequencies using the amplitude of the signal in each band and a spreading function.Signals in each band are then encoded using a quantization relative to the masking present for that band.

Block of 12 samples from each filter are analized at once in Layer1. Three12-samples blocks are analized for Layer2, Layer3 and AAC.The usage of three blocks at once allows for temporal masking to be taken into account; this also helps reducing data for level adjustment.

Main Enhancements in Layer3

Layer 3 MPEG audio encoding adds to Layer1 and 2 the following peculiarities:

- applies alias reduction
- applies non uniform quantization
- uses entropy encoding
- uses a bit reservoir

Main Enhancements in AAC

AAC MPEG audio encoding adds to Layer2 the following peculiarities:

- support for surround signals
- uses MDCT on filter bank's outputs
- uses Temporal Noise Shaping (TNS)
- uses backward adaptive prediction
- adds gain control and hybrid filter bank

Licensing

The myth: MPEG Audio is ``freeware''.

The truth: you probably need a license!

SOFTWARE CODECS

- Decoders. Freeware: OK; $1 per unit if sold
- Encoders. From $25 to $2.5 per year

HARDWARE CODECS

- Decoders. $1.25 per unit
- Encoders or full Codec. $2.50 per unit

MUSIC DISTRIBUTION

- 1% of revenue ($0.01 minimum)

MINIMUM ROYALTIES

- $15000 (!!!)

Part V
Graphics File Formats & WWW Publishing

Graphics File Formats & WWW Publishing

The ability to include bitmap images in text documents is probably the main reason for the success of Mosaic, the first WWW browser.

Nice-looking and well-structured documents are the key in WWW Publishing: a good appearence attracts the reader and good structuring helps finding information.

Images can't of course improve the structuring of your documents but can,for sure, improve their appearance.

Using inline bitmap graphics in HTML documents can also overcome some HTML formatting limitations.

Using GIFs

As per their own nature GIF files are better used with synthetic, non-photorealistic images.

It should also be noted that, for very small images, the file size difference between a LZW compressed and a JPEG compressed image is quite small.

For (relatively) large bitmaps the interleaved format should be used, but watch out: interleaved images often produces larger files.

If you paint the image on your own take care of using as less color as possible.

If you want to use GIF for continuous-tone images don't use dithering techniques when converting the image from a true-color format (otherwise be prepared to large files).

Using JFIFs

Store all photographic images using JFIF.

Use the higher compression that preserves your images quality (you can also use interactive programs to decide the compression ratio that's good for your purpose).

Always use progressive encoded images.

If you want to edit the image in the future, store a lossless compressed version of the image and modify that one.

GIF vs JFIF

Things you can do only with GIFs:

- Animation
- transparency

Things you can do only with JFIF:

- true-color images

Things for which GIF is better:

- Small, sharp images

Things for which JFIF is better:

- Large continuous-tone images

Tips and Tricks

Be kind with your visitors: use few, small images and as few animations as possible.

For (relatively) large images use interlaced/progressive encoding.

Compress your images as much as possible:

- For GIF images use as less colors as possible.
- For JFIF images use reasonable quality settings.

Always include image size information in the IMG tag.