Joint Photographic Experts Group (JPEG) compression

How it works – a chapter from The Internet and Business by Peter Scott © 1999

Fourier analysis and synthesis

At the heart of the JPEG process is a method which is highly mathematical. If pure maths is not for you, then do not feel upset if you have to give up on this section. Do give it a try, though, because of key nature of its role, and because it will give you a feel for the amount of work needed to manipulate pictures.

Joseph Fourier showed, in the 1820s, that any repeating shape can be created by a series of sine waves of different frequencies and heights. Sine waves are shown in fig 1.

One example is that of the sound produced by a musical instrument. When you listen to the same note played on different musical instruments, each sounds different. This is because you are not only listening to the note but also to a series of notes of double, treble, four times the frequency, and so on. These are called 'harmonics'. Some instruments, such as the oboe, have a very pure and simple sound because they have few harmonics. Others have a very complex sound, because they have many harmonics. On a violin, for example, you will hear up to the thirteenth harmonic.

In fig 1 you can see how three waves can be added to form a fourth wave. The second and third waves are double and quadruple the frequency of the first. If the heights of all three waves, called their 'amplitudes', are added together you start to get the triangular shape shown as the fourth wave.

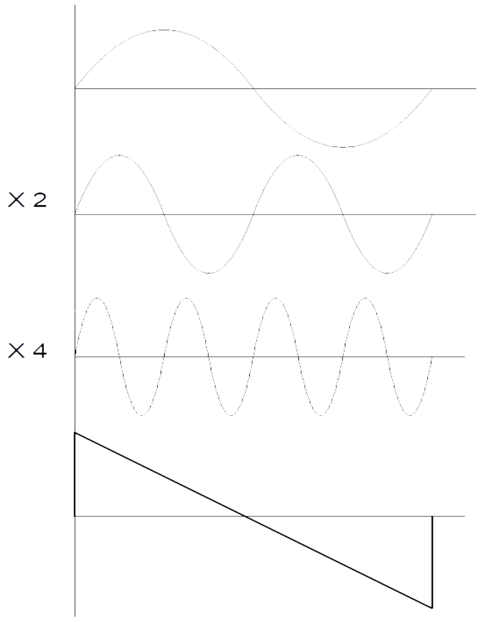


Fig 1 Sine waves being added together using Fourier synthesis

To get a perfect 'saw-tooth' shape you would need to add an infinite series of harmonics. Incidentally this is one of the reasons why computers can interfere with other electrical equipment. As explained in chapter one, computers use signals that are close to square in shape. These waves are made from a very long series of odd harmonics of three times, five times, seven times, nine times the frequency, and so on. The wires carrying the computer's clock signal can act like a radio transmitter aerial if design is poor.

Fourier created a mathematical method for analysing waves to find their harmonics called 'Fourier analysis'. He also created a formula to enable the original shape to be re-created, called 'Fourier synthesis'. The formula has many parts to it, each having a number, called a 'coefficient'. There is one part for each harmonic, and the coefficient determines the amplitude to be added. Fourier methods have been used on a variety of quantities that repeat over time or space, including planetary motion, the business cycle, and sound.

The example in fig 1 is simple because the sound repeats regularly over time. An extension of the method can be used on quantities that have no regularly repeating pattern. One example is the distribution of the brightness of the three colours over the two dimensions of a picture. A version of the Fourier technique, called the 'discrete cosine transformation' or DCT, is used in JPEG for data compression. DCT can be used to turn an area of a picture into a series of numbers, also called 'coefficients'. These numbers can then be used to re-create the picture. Some of the numbers are less important than others for the quality of the picture, so can be discarded. Discarding numbers means that fewer are stored, so the data is compressed. There will be loss of quality, but in a controlled way.

JPEG

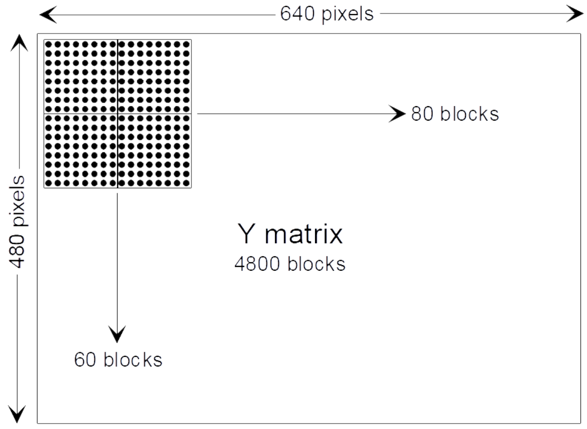
JPEG is an abbreviation of Joint Photographic Experts Group, which has developed a set of data compression standards. The letters JPEG are also used for the standards themselves, which cover the compression of the data for still pictures, and use a variety of algorithms, including DCT. The resulting files have the name extension .jpg. From the range of standards we will look at one which is in use in digital devices, namely a low resolution picture with a resolution of 640 x 480 pixels, and 24 bit colour. Each pixel is made up of the three colours- red (R), green (G), and blue (B), and each colour needs one byte. The number of bytes needed for this frame, if uncompressed, is 640 x 480 x 3, or just over 920 000 bytes.

For colour television information must be sent about each of the three colours which will be mixed on the screen. This information is often called a 'signal'. The Phase Alternate Line (PAL) method used in televisions in Britain and elsewhere does not send a signal for red, green, and blue brightnesses. Instead it mixes the colours in different proportions to create three new signals. The first, called 'luminance', and given the symbol 'Y', has a mixture of colours which matches the sensitivity of the human eye. The luminance signal, on its own, can be used to create a 'black and white' picture, with grey tones. The other two signals, called 'chrominance' and having symbols 'U' and 'V', add further colour information. U and V are less important to the perceived picture quality, so can be compressed more. JPEG uses Y, U, and V as well.

JPEG compression has six steps.

Step 1 Block preparation

Block preparation involves creating the three sets of data for Y, U, and V versions of the picture. Each set is called a 'matrix'. Each element of the matrix is still stored in eight bits, and each matrix on this low resolution screen has 640 x 480 elements, so there is no compression so far. Because of the lesser importance of the chrominances, the U and V matrices are now divided into blocks of 2 by 2 elements. The four elements of each block are averaged and stored in one element. The U and V matrices are now 320 x 240, so have been compressed four times. The last stage here is to divide each matrix into 'blocks' of eight by eight elements. The Y matrix will now have 4800 blocks, with 80 horizontally and 60 vertically, as shown in fig 2. The U and V will have 1200 each, with 40 horizontally and 30 vertically.



Block size shown much too large

Fig 2 Dividing into blocks the screen produced by the luminance

One block might look like fig 3. The numbers show the 'brightness' in each square in the block. They have values from 0 to 255 because each square in stored in eight bits.

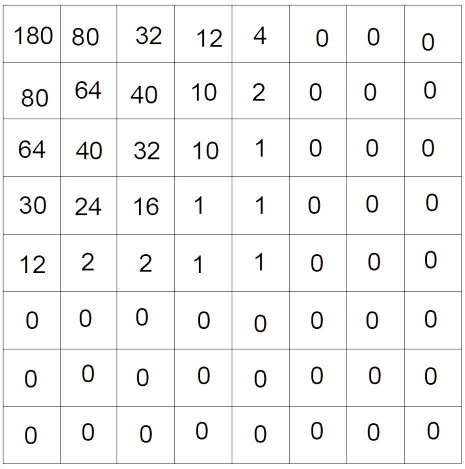
Remember that this a mixture of three colours.

23	16	176	176	67	78	78	78
25	16	176	176	43	32	31	23
27	16	154	154	143	32	33	33
22	14	201	187	32	35	56	66
15	10	245	205	44	39	60	65
11	9	0	0	50	43	28	28
8	8	56	65	52	47	31	33
5	6	78	87	54	45	43	38

Fig 3 One of the 8 x 8 screen blocks

Step 2 Apply discrete cosine transformation to each block

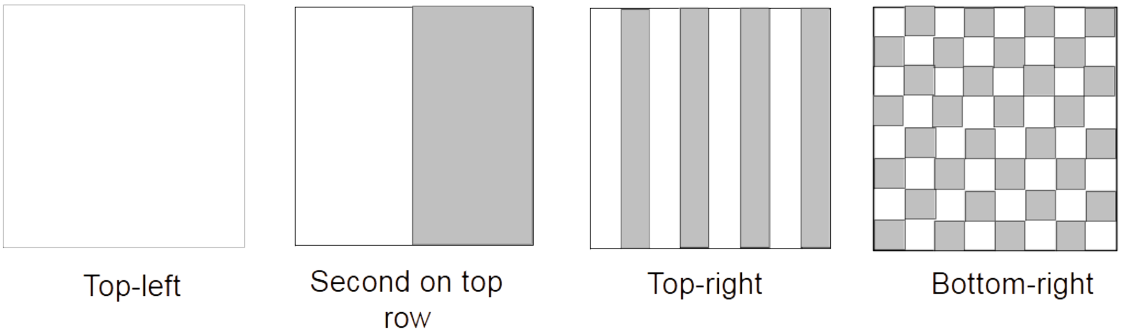
Discrete cosine transformation (DCT) converts the brightness numbers into a series of numbers called 'coefficients'. If you managed to get through the section on Fourier analysis you will know what these are. If not, it is enough to know that the numbers can be used to recreate the block and you can jump over the next paragraph.



180	80	32	12	4	0	0	0
80	64	40	10	2	0	0	0
64	40	32	10	1	0	0	0
30	24	16	1	1	0	0	0
12	2	2	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

Fig 4 Coefficients resulting from DCT

There are 64 coefficients, which are also stored in a matrix of eight by eight elements as shown in fig 4. The top-left value is the average for the block. As you move across the top of the matrix from the left, the frequency of the cosine goes up. The next one left splits the picture into two vertically. It stores the brightness of the left half, compared with the right. By the time you get to the top right, the picture block is split into eight vertical stripes each being a column of single pixels. Again this coefficient stores the relative brightness of the stripes. The same thing happens when moving vertically on the matrix. Thus by the time you get to the bottom right, the picture block is split into sixty-four, single-pixel areas. Some examples are shown in fig 5. When all of the patterns are put back together the picture is re-created. You will see that the values of the coefficients fall away from top-left to bottom-right. Please note that the values shown in fig 4 are typical, but not correct for the block in fig 3. This step does not produce any compression, and is lossless apart from small value changes produced when the analogue values are rounded to digital numbers by the computer.



Top-left

Second on top row

Top-right

Bottom-right

Fig 5 Some examples of DCT coefficient patterns

Step 3 Apply quantisation to each block

The coefficients towards the bottom-right are smaller in value because they provide finer adjustments. They are also less important to picture quality because they affect smaller squares on the picture. To make the drop-off in value more marked, each coefficient is now divided by a number. The numbers, called 'weightings', are taken from another eight by eight matrix. If each number is set to one, there will be no change to the coefficients. The values shown in fig 6 are typical. Where the result of the division is a small number, rounding will reduce it to zero. More of the coefficients are therefore now zero in value as shown in fig 7.

1	1	2	4	8	16	32	64
1	1	2	4	8	16	32	64
2	2	2	4	8	16	32	64
4	4	4	4	8	16	32	64
8	8	8	8	8	16	32	64
16	16	16	16	16	16	32	64
32	32	32	32	32	32	32	64
64	64	64	64	64	64	64	64

Fig 6 A weighting matrix

180	80	16	3	0	0	0	0
80	64	20	2	0	0	0	0
32	20	16	2	0	0	0	0
7	6	4	0	0	0	0	0
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

Fig 7 Coefficients after quantisation

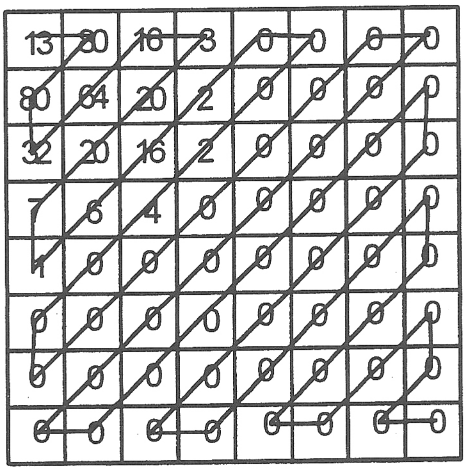
Clearly this step is lossy. Changing the values of the coefficients means that the block cannot now be exactly re-created, because a zero can never be restored to its original value. Indeed if the weighting numbers were large then all of the coefficients would become zero except the top-left. The whole block would take on the average value, and be of a single colour and brightness. The choice of weighting values is matched with the use to which it is put. It is a compromise between loss of quality and the need for compression. It is these numbers you select when you choose the degree of compression as you save the picture.

Step 4 Apply differential encoding

This is a very small step that replaces the top-left average value with the difference between it and the value for the previous block.

Step 5 Apply run-length encoding (RLE) to the block numbers

The quantised block values are now assembled into a sequence using a zig-zag pattern as shown in fig 8. A zig-zag is used so that most of the zeroes appear in a row at the end of the sequence, in this case 45. Run-length encoding is now used to replace a series of identical numbers, in this case mostly the trailing zeroes, with a single RLE code. This results in further compression.



13	80	16	3	0	0	0	0
80	64	20	2	0	0	0	0
32	20	16	2	0	0	0	0
7	6	4	0	0	0	0	0
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

Fig 8 The zig-zag pattern used in run-length encoding on a block

Run-length encoding

This algorithm, usually abbreviated to RLE, is part of a group of techniques called 'entropy encoding'. No account is taken of what the characters represent. They are simply examined to see if any are repeated. Entropy is a measure of how disorganised something is.

Suppose we want to compress this series of character bytes:
198611111111111111111111376842222222298794333333333333333333545765
There are 66 characters in all.

We then split the stream into repeating and non-repeating groups:

1986	There are no adjacent repeats
111111111111111111111	There are twenty-one 1s
37684	There are no adjacent repeats
22222222	There are eight 2s
98794	There are no adjacent repeats
33333333333333333	There are seventeen 3s
545765	There are no adjacent repeats

This method replaces the repeating characters with a group of symbols. The group starts with a special character. In this case 'A' is used, though in practice it would not be. Next the repeating character is added, followed by a number which shows how many of them there are.

111111111111111111111	becomes	A121
22222222	becomes	A208

33333333333333333 becomes A317

The method does nothing at all to the groups of non-repeating characters.

The result will be the following series of 32 characters, without the gaps that are included here for clarity:
1986 A121 37684 A208 98794 A317 545765
The original 66 characters have been compressed into 32. This means that the compression is:

Compression ratio = 66 / 32 : 1 or 2.06 : 1

Step 6 Apply Huffman encoding

It is likely that a particular RLE code will be used more than once. Each is replaced a shorter Huffman code, again compressing the data, and completing the process.

Huffman codes

A file is made up from a dictionary of characters. For example this book is written using a hundred or so characters. Some characters occur more often than others. If we just consider the alphabet, the letter 'e' occurs most often, followed by 't', 'i', and so on. In the Huffman method, characters are replaced by a code of bits. The codes are not fixed length. The maximum number of bits is chosen so that there is a code for each character. The shortest codes, 0 and 1, are used for the most frequently occurring characters, and longer ones are used for the less common. There is no room here to describe yet another algorithm in detail. In brief though, the frequencies of the characters are found, for example 'e' makes up 23% and 'z' 1% of the characters in the file. A tree diagram is built up which allows the bit code for each character to be read for compression and re-read for decompression. Morse code worked in a similar way, having the code '.' for 'e', and '--.-' for 'q'.

You can see that the JPEG method throws almost every possible compression technique at the problem. It can achieve a compression ratio of up to about 20 : 1. Despite its mind-boggling complexity, the method is applied as the user saves the file as a JPEG, taking just a few seconds. JPEG is symmetrical, taking roughly equal times to compress and decompress. One of its great strengths is its ability to vary the compression and loss by changing the numbers in the quantisation table.

MPEG

MPEG is an abbreviation for 'Moving Picture Experts Group', and has the unlikely official title of ISO/IEC JTC1 SC29 WG11. For those who like the bizarre this stands for the 'International Standards Organisation, International Electrotechnical Commission, Joint Technical Committee 1, Sub-Committee 29, Working Group 11'. As with JPEG, the abbreviation is also used for the standards produced. MPEG is a set of standards for compressing sounds and moving pictures. Believe it or not, MPEG includes the JPEG technique and adds yet more compression to it, by removing temporal redundancy.

To understand why it is needed, consider the situation where we want to send frames as described above for JPEG, at a rate of fifty frames per second to create a moving picture. Each frame took up 920 000 bytes or just under 7.5 Mbits. At 50 frames per second this requires a transmission speed of 375 Mbit/s. This can not at present even be achieved from a fast hard disk, so certainly cannot be done over the Internet or from a compact disk. To transmit moving video pictures over the Internet, even over high-speed connections, we need to achieve a further ten-times compression.

At the time of writing there were three MPEG standards:

MPEG-1
This is used for images of video recorder quality, and covers both audio and video. It needs a data transmission rate of 1.2 Mbit/s and is also used in CDROM movies. It is this standard which we will be looking at in detail, though only the video part.

MPEG-2
MPEG-2 is used to compress broadcast quality moving pictures for television. In the US NTSC system it requires 4 Mbit/s and for PAL 6 Mbit/s. It is also adopted for High-Definition Television (HDTV), in which context it was to be called MPEG-3. We will be looking at the audio side of MPEG-2, called MPEG-2 Layer 3, or MP3.

MPEG-4
This is designed for medium-resolution video-conferencing over low speed lines.

MPEG-1

MPEG-1 makes use of the JPEG compression method, modified to work on a series of frames. JPEG on its own does not produce enough compression. A single JPEG frame is known as an 'Intracoded' frame or 'I-frame'. MPEG makes use of the fact that adjacent frames are likely to be similar even over a period of seconds. This is known as 'temporal redundancy'. Full I-frames are sent periodically, but between these a series of 'Predictive' or 'P-frames' are sent. Such a frame contains only information about those parts of the picture which have changed. It is therefore smaller than an I-frame, and is further compressed using JPEG. How P-frames are created is covered later. The only time that a sequence of full I-frames would be needed is where each frame might need to be viewed individually, for example 'Let's see that offside again!'.

If a camera is panning fast, if there is very fast action, or if one object passes behind another, 'Bi-directional' or 'B-frames' are sometimes included. B-frames contain the differences between the present picture frame and both the last and next frame.

To enable rapid searching of a video sequence, there is a fourth type of frame called a 'D-frame'. Each block is of a single average colour and brightness, and is known as a 'macroblock'. It gives just enough information to see what is happening. From your knowledge of JPEG you will realise that only the top-left coefficient is stored with the remaining 63 being zero and stored as one RLE/Huffman character. The data is highly compressed.

Any video sequence in MPEG-1 comprises a series of P-frames or B-frames, with an I-frame being inserted once every second or so. Without the I-frames the difference frames would be meaningless. Anyone who started viewing in the middle of a sequence, or who suffered from even a brief loss of data, would not immediately be able to recreate the correct picture. That is why you get breakup when a signal is interrupted

Summary of frames

I-frame	Full JPEG frame
P-frame	JPEG coded differences with last frame
B-frame	JPEG coded differences with last and next frame
D-frame	Macroblock frame for fast viewing

Encoding P-frames

Fig 9 shows a sequence of two frames. The only change from one to the next is the position of the aeroplane in the sky. Each frame is divided up into blocks like those for JPEG, again called 'macroblocks'. For each macroblock on the second frame the computer searches for one on the first frame which is similar or identical. Searching is done by looking at increasing amounts of vertical and horizontal movement. The degree of match for each position is calculated and the best match found. The difference between the new and the old position is then coded using JPEG, and stored as a 'motion vector' together with the block details. To create B-frames the comparison must be made with the previous and next frame.

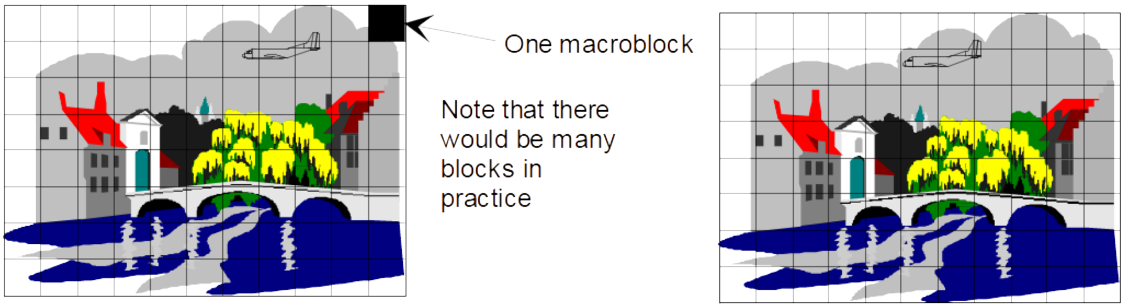


Fig 9 Two consecutive frames in a moving picture sequence

This takes time, so MPEG encoding is a very asymmetric process. Days, or even weeks, can be spent optimising the P-frames to get the best effect of motion for an hour of film. A balance must be made between encoding time and picture quality. Poor encoding results in a number of very visible faults, amongst which are 'mosquitoes', 'macroblocking', and 'digital rain'. Mosquitoes are fuzzy or jittery edges seen on some objects where accurate positioning has not been achieved. Macroblocking, or 'blocking', is where each picture block is of a single colour, and digital rain is where small objects disappear and reappear due to poor P-frame encoding.

MP3 FOR AUDIO

MP3 is an abbreviation for MPEG-2 layer 3. It does for sound what MPEG does for video, that is it compresses the data. MP3 produces a compression ratio of about ten. It is a lossy compression method, so the sounds suffer a small loss of quality.

MP3 encoder software usually requires that the sounds are in .wav form. The simplest way to create such files is to feed the sounds into the line input socket of your sound card from a hi-fi system or other sound source. You can use the Sound Recorder accessory that comes free with Windows, or some other similar software. Alternatively so-called 'Ripper' software allows you to take the digital data directly off a music compact disk, though of course you must be sure that you are not breaking the law in doing so. After all the name comes from the term to ‘rip off’.There is more sophisticated software available to create and edit MP3 files, much of it free for linux or Windows. One effect of this compression is that you can record about eleven hours of near CD quality sound on a standard recordable CD and can carry your entire music collection around with you on a tiny device or phone.

CONCLUSION

Whatever the computer does, the only data it can handle is a 1 and a 0. In this article you have learned how such disparate things as letters, numbers, pictures, and sounds are turned into these two characters. Letters and numbers take up very little space. Picture, video and sound files are very large in comparison, and without compression might take too long to be sent over the Internet even at high connection speeds. For this reason a great deal of effort has been put into designing methods to compress such files, and it is a tribute to the great power of the modern computer that they work at all.

Created 31 January 2022
Last edit 31 January 2022