

MPEGPLAY Index

Commands

[File Menu](#)

[Movie Menu](#)

[Dithers Menu](#)

[Options Menu](#)

[Help Menu](#)

Reference

[Title page](#)

[An Introduction to MPEG](#)

[How to register your copy of MPEGPLAY](#)

[Disclaimer \(Please Read!\)](#)

Windows NT, Win32 and Win32s are trademarks of the Microsoft Corporation

MPEGPLAY

A MPEG player for Win32(tm) and Win32s(tm)
(c) 1993 by Michael Simmons

MPEGPLAY is a program that enables you to play back MPEG encoded Digital Video Files.

MPEG is an International standard for the encoding of Video.

The code was derived from the Berkeley MPEG Video Software Decoder developed at the University of California at Berkeley by Lawrence A. Rowe, Ketan Patel, and Brian Smith. It has been ported and optimised for the WIN32(tm) Windows system by Michael Simmons

This port is **shareware**. If you find this program useful then you must register it. By registering your copy you will also assist in the funding of future enhancements and bug fixes to the current Win32(tm) Player. You will also assist in the porting and/or development of an MPEG encoder/decoder.

[Return to Contents](#)

[How to Register](#)

The File Menu

- Open** - Opens a movie file.
- Close** - Closes a movie file.
- Split MPEG** - Splits a multistream MPEG into separate files call stmxxx.mpg
Where xxx is the stream Number (192-223 are Audio,224-239 are Video,190 is padding).
- Exit** - Exits the Player.

[Return to Contents](#)

The Movie Menu

The Movie menu is used to control the playing of the movie file.

- Stop** - Stops the Player.
- Play** - Starts the Player playing.
- Rewind** - Rewinds the Movie to the beginning. If the movie was playing it is restarted at the beginning
- Advance** - Advances the Movie one frame.
- Timed Play** - Plays the movie and then Displays the Number of Frames per Second.

[Return to Contents](#)

The Dither Menu

The Dither menu is used to select a dithering algorithm used to translate the 8 bit Luminance, 8 bit Cr and the 8 bit Cb into a Windows DIB.

- Mono Threshold** - if luminance >128 then pixel is set.
- Mono** - Dithers to a monochrome bitmap
- gray** - Luminance is mapped to a 256 gray scale palette
- gray 2x2** - As above but scaled up by two. Interpolates extra pixels
- Ordered** - Dithers to a 128 fix color palette
- 2x2** - As above but scaled up by two.
- Full Color** - Luminance,Cr and Cb are converted to a 24 bit DIB
- Fs2fast** - Error diffusion of 2 error values.
- Fs2** - Error diffusion of 2 errors values.
- Fs4** - Error diffusion of 4 errors values.
- Hybrid** - Mixture of Ordered dither for luminance and error diffusion for Cr and Cb.

[Return to Contents](#)

The Options Menu

The Option menu is used to select various display and functional options of the player

- Loop** - When set the Player will Loop while playing
- Stretch to Window** - The image is Stretch to fill the Windows Client area. Instead of being copied Pixel by Pixel.
- Use Palette Indices** - BitMap bytes are used directly as Palette Indices. NT Palette Modes Only.
- Transfer Small Dibs** - Breaks the image up into 64KB Block before sending to the screen.
- Skip B Frames** - The player will skip over B frames.
- Skip P Frames** - The player will skip over P frames. (Note B Frames are also skipped)

[Return to Contents](#)

The Help Menu

- Help** - Call's up this help file
- About** - Displays an information dialog box

[Return to Contents](#)

To receive the latest version of the Player, send a \$25 US bank cheque and a list of suggestions and/or bugs to

Michael Simmons
34 Shillington Way
THORNLIE WA 6108
AUSTRALIA

(Note the registered version will play >1MB Mpeg file. The About Box is also not displayed at startup

[Return to Contents](#)

In the following, the Term DEVELOPERS is taken to mean the Developers of this software. That is THE UNIVERSITY OF CALIFORNIA and myself
MICHAEL SIMMONS

IN NO EVENT SHALL THE DEVELOPERS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE DEVELOPERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE DEVELOPERS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE DEVELOPERS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

[Return to Contents](#)
[Goto to Introduction](#)

An Introduction to MPEG

written by Mark Adler

Warning this information is out of date and may contain incorrect information
it may be updated in the next version of this player. See [Disclaimer](#)

[What is MPEG](#)

[What does MPEG stand for](#)

[Does it have anything to do with JPEG](#)

[What is JBIG and MHEC](#)

[What has MPEG accomplished](#)

[How does MPEG work](#)

[Is each frame predicted from the last](#)

[Where did they get 352x240](#)

[What about Europe](#)

[What about audio compression](#)

[How much does it compress](#)

[What is MPEG phaseII](#)

[Why not just scale up MPEG I](#)

[What works for MPEG II](#)

[When will MPEG II be released](#)

[How do I join MPEG](#)

[How do I get the MPEG standards documents](#)

[Return to Contents](#)

Q. What is MPEG?

A. MPEG is a group of people that meet under ISO (the International Standards Organization) to generate standards for digital video (sequences of images in time) and audio compression. In particular, they define a compressed bit stream, which implicitly defines a decompressor. However, the compression algorithms are up to the individual manufacturers, and that is where proprietary advantage is obtained within the scope of a publicly available international standard. MPEG meets roughly four times a year for roughly a week each time. In between meetings, a great deal of work is done by the members, so it doesn't all happen at the meetings. The work is organized and planned at the meetings.

[Return to Introduction](#)

[Return to Contents](#)

Q. So what does MPEG stand for?

A. Moving Pictures Experts Group.

[Return to Introduction](#)

[Return to Contents](#)

Q. Does it have anything to do with JPEG?

A. Well, it sounds the same, and they are part of the same subcommittee of ISO along with JBIG and MHEG, and they usually meet at the same place at the same time. However, they are different sets of people with few or no common individual members, and they have different charters and requirements. JPEG is for still image compression.

[Return to Introduction](#)

[Return to Contents](#)

Q. Then what's JBIG and MHEG?

A. JBIG is for lossless binary image compression (like faxes)MHEG is for multi-media data standards (like integrating stills, video, audio, text, etc.).

[Return to Introduction](#)

[Return to Contents](#)

Q. What has MPEG accomplished?

A. So far (as of January 1992), they have completed the "Committee Draft" of MPEG phase I, colloquially called MPEG I. It defines a bit stream for compressed video and audio optimized to fit into a bandwidth (data rate) of 1.5 Mbits/s. This rate is special because it is the data rate of (uncompressed) audio CD's and DAT's. The draft is in three parts, video, audio, and systems, where the last part gives the integration of the audio and video streams with the proper timestamping to allow synchronization of the two. They have also gotten well into MPEG phase II, whose task is to define a bitstream for video and audio coded at around 3 to 10 Mbits/s.

[Return to Introduction](#)

[Return to Contents](#)

Q. So how does MPEG I work?

A. First off, it starts with a relatively low resolution videosequence (possibly decimated from the original) of about 352 by 240 frames by 30 frames/s (US--different numbers for Europe), but original high (CD) quality audio. The images are in color, but converted to YUV space, and the two chrominance channels (U and V) are decimated further to 176 by 120 pixels. It turns out that you can get away with a lot less resolution in those channels and not notice it, at least in "natural" (not computer generated) images. The basic scheme is to predict motion from frame to frame in the temporal direction, and then to use DCT's (discrete cosine transforms) to organize the redundancy in the spatial directions. The DCT's are done on 8x8 blocks, and the motion prediction is done in the luminance (Y) channel on 16x16 blocks. In other words, given the 16x16 block in the current frame that you are trying to code, you look for a close match to that block in a previous or future frame (there are backward prediction modes where later frames are sent first to allow interpolating between frames). The DCT coefficients (of either the actual data, or the difference between this block and the close match) are "quantized", which means that you divide them by some value to drop bits off the bottom end. Hopefully, many of the coefficients will then end up being zero. The quantization can change for every "macroblock" (a macroblock is 16x16 of Y and the corresponding 8x8's in both U and V). The results of all of this, which include the DCT coefficients, the motion vectors, and the quantization parameters (and other stuff) is Huffman coded using fixed tables. The DCT coefficients have a special Huffman table that is "two-dimensional" in that one code specifies a run-length of zeros and the non-zero value that ended the run. Also, the motion vectors and the DC DCT components are DPCM (subtracted from the last one) coded.

[Return to Introduction](#)

[Return to Contents](#)

Q. So is each frame predicted from the last frame?

A. No. The scheme is a little more complicated than that. There are three types of coded frames. There are "I" or intra frames. They are simply a frame coded as a still image, not using any past history. You have to start somewhere. Then there are "P" or predicted frames. They are predicted from the most recently reconstructed I or P frame. (I'm describing this from the point of view of the decompressor.) Each macroblock in a P frame can either come with a vector and difference DCT coefficients for a close match in the last I or P, or it can just be "intra" coded (like in the I frames) if there was no good match. Lastly, there are "B" or bidirectional frames. They are predicted from the closest two I or P frames, one in the past and one in the future. You search for matching blocks in those frames, and try three different things to see which works best. (Now I have the point of view of the compressor, just to confuse you.) You try using the forward vector, the backward vector, and you try averaging the two blocks from the future and past frames, and subtracting that from the block being coded. If none of those work well, you can intra- code the block.

The sequence of decoded frames usually goes like:

IBBPBBPBBPBBIBBPBBPB...

Where there are 12 frames from I to I (for US and Japan anyway.) This is based on a random access requirement that you need a starting point at least once every 0.4 seconds or so. The ratio of P's to B's is based on experience. Of course, for the decoder to work, you have to send that first P *before* the first two B's, so the compressed data stream ends up looking like: 0x312645... where those are frame numbers. xx might be nothing (if this is the true starting point), or it might be the B's of frames -2 and -1 if we're in the middle of the stream somewhere. You have to decode the I, then decode the P, keep both of those in memory, and then decode the two B's. You probably display the I while you're decoding the P, and display the B's as you're decoding them, and then display the P as you're decoding the next P, and so on.

[Return to Introduction](#)

[Return to Contents](#)

Q. Where did they get 352x240 image size?

A. That derives from the CCIR-601 digital television standard which is used by professional digital video equipment. It is (in the US) 720 by 243 by 60 fields (not frames) per second, where the fields are interlaced when displayed. (It is important to note though that fields are actually acquired and displayed a 60th of a second apart.) The chrominance channels are 360 by 243 by 60 fields a second, again interlaced. This degree of chrominance decimation (2:1 in the horizontal direction) is called 4:2:2. The source input format for MPEG I, called SIF, is CCIR-601 decimated by 2:1 in the horizontal direction, 2:1 in the time direction, and an additional 2:1 in the chrominance vertical direction. And some lines are cut off to make sure things divide by 8 or 16 where needed.

Q. What if I'm in Europe?

A. For 50 Hz display standards (PAL, SECAM) change the number of lines in a field from 243 or 240 to 288, and change the display rate to 50 fields/s or 25 frames/s. Similarly, change the 120 lines in the decimated chrominance channels to 144 lines. Since 288×50 is exactly equal to 240×60 , the two formats have the same source data rate.

[Return to Introduction](#)

[Return to Contents](#)

Q. What about audio compression.

A. Well, I don't know as much about the audio compression. Basically they use very carefully developed psychoacoustic models derived from experiments with the best obtainable listeners to pick out pieces of the sound that you can't hear. There are what are called "masking" effects where, for example, a large component at one frequency will prevent you from hearing lower energy parts at nearby frequencies, where the relative energy vs. frequency that is masked is described by some empirical curve. There are similar temporal masking effects, as well as some more complicated interactions where a temporal effect can unmask a frequency, and vice-versa. The sound is broken up into spectral chunks with a hybrid scheme that combines sine transforms with subband transforms, and the psychoacoustic model written in terms of those chunks. Whatever can be removed or reduced in precision is, and the remainder is sent. It's a little more complicated than that, since the bits have to be allocated across the bands. And, of course, what is sent is entropy coded.

[Return to Introduction](#)

[Return to Contents](#)

Q. So how well does MPEG compress?

A. As I mentioned before, audio CD data rates are about 1.5 Mbits/s. You can compress the same stereo program down to 256 Kbits/s with no loss in discernable quality. (So they say. For the most part it's true, but every once in a while a weird thing might happen that you'll notice. However the effect is very small, and it takes a listener trained to notice these particular types of effects.) That's about 6:1 compression. So, a CD MPEG I stream would have about 1.25 Mbits/s left for video. The number I usually see though is 1.15 Mbits/s (maybe you need the rest for the system data stream). You can then calculate the video compression ratio from the numbers here to be about 26:1. If you step back and think about that, it's little short of a miracle. Of course, it's lossy compression, but it can be pretty hard sometimes to see the loss, if you're comparing the SIF original to the SIF decompressed. There is, however, a very noticeable loss if you're coming from CCIR-601 and have to decimate to SIF, but that's another matter. I'm not counting that in the 26:1. The standard also provides for other bit rates ranging from 32Kbits/s for a single channel, up to 448 Kbits/s for stereo.

[Return to Introduction](#)

[Return to Contents](#)

Q. What is MPEG phase II?

A. There is a considerable loss of quality in going from CCIR-601 to SIF resolution. For entertainment video, it's simply not acceptable. You want to use more bits and code all or almost all the CCIR-601 data. From subjective testing at the Japan meeting in November 1991, it seems that 4 Mbits/s can give very good quality compared to the original CCIR-601 material. The objective of phase II is to define a bit stream optimized for these resolutions and bit rates.

[Return to Introduction](#)

[Return to Contents](#)

Q. Why isn't MPEG II just a scaled up version of MPEG I?

A. The main difficulty is the interlacing. The simplest way to extend MPEG I to interlaced material is to put the fields together into frames (720x486x30/s). This results in bad motion artifacts that stem from the fact that moving objects are in different places in the two fields, and so don't line up in the frames. Compressing and decompressing without taking that into account somehow tends to muddle the objects in the two different fields. The other thing you might try is to code the even and odd field streams separately. This avoids the motion artifacts, but as you might imagine, doesn't get very good compression since you are not using the redundancy between the even and odd fields where there is not much motion (which is typically most of image). Or you can code it as a single stream of fields. Or you can interpolate lines. Or, etc. etc. There are many things you can try, and the point of MPEG II is to figure out what works well. MPEG II is not limited to consider only derivations of MPEG I. There were several non-MPEG I-like schemes in the competition in November, and some aspects of those algorithms may or may not make it into the final standard for entertainment video compression.

[Return to Introduction](#)

[Return to Contents](#)

Q. So what works?

A. Basically, derivations of MPEG I worked quite well, with one that used wavelet subband coding instead of DCT's that also worked very well. Also among the worked-very-well's was a scheme that did not use B frames at all, just I and P's. All of them, except maybe one, did some sort of adaptive frame/field coding, where a decision is made on a macroblock basis as to whether to code that one as one frame macroblock or as two field macroblocks. Some other aspects are how to code I-frames--some suggest predicting the even field from the odd field. Or you can predict evens from evens and odds or odds from evens and odds or any field from any other field, etc. Ok, we're not really sure what works best yet. The next step is to define a "test model" to start from, that incorporates most of the salient features of the worked-very-well proposals in a simple way. Then experiments will be done on that test model, making a mod at a time, and seeing what makes it better and what makes it worse. Example experiments are, B's or no B's, DCT vs. wavelets, various field prediction modes, etc. The requirements, such as implementation cost, quality, random access, etc. will all feed into this process as well.

[Return to Introduction](#)

[Return to Contents](#)

Q. When will all this be finished?

A. I don't know. I'd have to hope in about a year or less.
MPEG II is due in March 1993. After being expanded to cover HDTV

[Return to Introduction](#)

[Return to Contents](#)

Q. How do I join MPEG?

A. You don't join MPEG. You have to participate in ISO as part of a national delegation. How you get to be part of the national delegation is up to each nation. I only know the U.S., where you have to attend the corresponding ANSI meetings to be able to attend the ISO meetings. Your company or institution has to be willing to sink some bucks into travel since, naturally, these meetings are held all over the world. (For example, Paris, Santa Clara, Kurihama Japan, Singapore, Haifa Israel, Rio de Janeiro, London, etc.)

[Return to Introduction](#)

[Return to Contents](#)

Q. How do I get ISO documents, like the MPEG I draft?

A. MPEG is a draft ISO standard.

It's exact name is ISO CD 11172. The draft consists of three parts: System, Video, and Audio. The System part (11172-1) deals with synchronization and multiplexing of audio-visual information, while the Video (11172-2) and Audio part (11172-3) address the video and the audio compression techniques respectively. You may order it from your national standards body (e.g. ANSI in the USA) or buy it from companies like

OMNICOM

phone +44 438 742424

FAX +44 438 740154

[Return to Introduction](#)

[Return to Contents](#)

