

NAME

ppmtompeg – encodes MPEG-1 bitstreams

SYNOPSIS

ppmtompeg [*options*] *parameter-file*

DESCRIPTION

ppmtompeg produces an MPEG-1 video stream. *param_file* is a parameter file which includes a list of input files and other parameters. The file is described in detail below. The **-gop**, **-combine_gops**, **-frames**, and **-combine_frames** options are all exclusive. This man page is probably incomplete. For complete usage, see the User's Guide.

OPTIONS

- stat *stat_file*** : causes the encoder to append the statistics to the file *stat_file*. In any case, the statistics are output to stdout. The statistics use the following abbreviations: bits per block (bpb), bits per frame (bpf), seconds per frame (spf), and bits per second (bps).
- quiet *num_seconds*** : causes the program to not report remaining time for at least *num_seconds* seconds. A negative values tells the program not to report at all. 0 is the default (reports once after each frame). Note that the time remaining is an estimate and does not take into account time to read in frames.
- realquiet** : causes the encoder to run silently, with the only screen output being errors. Particularly useful when reading input from stdin.
- no_frame_summary** : prevents the program from printing a summary line for each frame
- float_dct** : forces the encoder to use a more accurate, yet more computationally expensive version of the DCT.
- gop *gop_num*** : causes the encoder to only encode the numbered GOP (first GOP is 0). The parameter file is the same as for normal usage. The output file will be the normal output file with the suffix ".gop.<gop_num>". No sequence info is output.
- combine_gops** : causes the encoder to simply combine some GOP files into a single MPEG stream. A sequence header/ender are inserted. In this case, the parameter file need only contain the YUV_SIZE value, an output file, and perhaps a list of input GOP files (see below).
- frames *first_frame last_frame*** : causes the encoder to only encode the frames from *first_frame* to *last_frame*, inclusive. The parameter file is the same as for normal usage. The output will be placed in separate files, one per frame, with the file names being the normal output file with the suffix ".frame.<frame num>". No GOP header information is output. (Thus, the parameter file need not include the GOP_SIZE value)
- combine_frames** : causes the encoder to simply combine some frames into a single MPEG stream. Sequence and GOP headers are inserted appropriately. In this case, the parameter file need only contain the YUV_SIZE value, the GOP_SIZE value, an output file, and perhaps a list of frame files (see below).
- nice** : causes the program to run any remote processes 'nicely.' This is only relevant if the program is using parallel encoding. (see 'man nice.')
- max_machines *num_machines*** : causes the program to use no more than *num_machines* machines as slaves for use in parallel encoding.
- snr** : print the signal-to-noise ratio. Prints SNR (Y U V) and peak SNR (Y U V) for each frame. In summary, prints averages of luminance only (Y). SNR is defined as $10 \cdot \log(\text{variance of original} / \text{variance of error})$. Peak SNR is defined as $20 \cdot \log(255 / \text{RMSE})$. Note that the encoder will run a little slower if you want it to print the SNR.
- mse** : computes the mean squared error per block. Also automatically computes the quality of the images when set, so there is no need to specify -snr then.

- bit_rate_info rate_file** : prints bit rate information into the file rate_file. Bit rate info is bits per frame, and also bits per I-frame-to-I-frame.
- mv-histogram** : prints histogram of motion vectors as part of statistics. There are three histograms -- one for P, forward B, and backward B vectors. Each histogram is a 2-dimensional array, and there is one entry for each vector in the search window.

PARAMETER FILE

The parameter file **MUST** contain the following lines (except when using the **-combine_gops** or **-combine_frames** options):

PATTERN <pattern>

OUTPUT <output file>

INPUT_DIR <directory>

all input files must reside in this directory. If you want to refer to the current directory, use '.' (an empty INPUT_DIR value would refer to the root directory). If input files will be coming in from standard input, use 'stdin'.

INPUT

This line must be followed by a list of the input files (in display order) and then the line

END_INPUT

There are three types of lines between INPUT and END_INPUT. First, a line may simply be the name of an input file. Secondly, the line may be of the form

<single_star_expr> [x-y]

single_star_expr can have a single '*' in it. It is replaced by all the numbers between x and y inclusive. So, for example, the line

tennis*.ppm [12-15]

is replaced by tennis12.ppm, tennis13.ppm, tennis14.ppm, tennis15.ppm. Uniform zero-padding occurs, as well. For example, the line

football*.ppm [001-130]

is replaced by football.001.ppm, football.002.ppm, ..., football.009.ppm, football.010.ppm, ..., football.130.ppm. The third type of line is:

<single_star_expr> [x-y+s]

Where the line is treated exactly as above, except that we skip by s. Thus, the line

football*.ppm [001-130+4]

is replaced by football.001.ppm, football.005.ppm, football.009.ppm, football.013.ppm, etc.

BASE_FILE_FORMAT <YUV or PPM or PNM or JPEG or JMOVIE>

All the input files must be converted to YUV, JPEG(v4), JMOVIE, PNM, or PPM format. This line specifies which of the three formats (actually PPM is a subset of PNM). The reason for having a separate PPM option is for simplicity. If your files are RAWBITS ppm files, then use the PPM option rather than the PNM. Also, depending on the system, file reads will go much faster with the PPM option (as opposed to PNM).

INPUT_CONVERT <conversion command>

You must specify how to convert a file to the base file format. In the conversion command, each '*' is replaced by the filename (the items listed between INPUT and END_INPUT). If no conversion is necessary, then you would just say:

INPUT_CONVERT *

If you had a bunch of gif files, you might say:

INPUT_CONVERT giftoppm *

If you have a bunch of separate a.Y, a.U, and a.V files, then you might say:

INPUT_CONVERT cat *.Y *.U *.V

Input conversion is not allowed with input from stdin.

GOP_SIZE <n>

n is roughly the number of frames in a Group of Pictures (roughly because a GOP must begin with an I-frame)

SLICES_PER_FRAME <n>

n is roughly the number of slices per frame. Note, at least one MPEG player may complain if slices do not start at the left side of an image. To ensure this does not happen, make sure the number of rows is divisible by SLICES_PER_FRAME.

PIXEL <FULL or HALF>

use half-pixel motion vectors, or only full-pixel ones

RANGE <n>

use a search range of +/- n pixels

PSEARCH_ALG <algorithm>

algorithm must be one of {EXHAUSTIVE, TWOLEVEL, SUBSAMPLE, LOG-ARITHMIC}. Tells what kind of search procedure should be used for P-frames. Exhaustive gives the best compression, but logarithmic is the fastest. You select the desired combination of speed and compression. TWOLEVEL is an exhaustive full-pixel search, followed by a local half-pixel search around the best full-pixel vector (the PIXEL option is ignored for this search algorithm).

BSEARCH_ALG <algorithm>

algorithm must be one of {SIMPLE, CROSS2, EXHAUSTIVE}. Tells what kind of search procedure should be used for B-frames. Simple means find best forward and backward vectors, then interpolate. Cross2 means find those two vectors, then see what backward vector best matches the best forward vector, and vice versa. Exhaustive does an n-squared search and is EXTREMELY slow in relation to the others (Cross2 is about twice as slow as Simple).

IQSCALE <n>

use n as the qscale for I-frames

PQSCALE <n>

use n as the qscale for P-frames

BQSCALE <n>

use n as the qscale for B-frames

REFERENCE_FRAME <ORIGINAL or DECODED>

If ORIGINAL is specified, then the original images are used when computing motion vectors. To be more accurate, use DECODED, in which the decoded images are used. This should increase the quality of the image, but will take a bit longer to encode.

The following lines are optional:

FORCE_I_ALIGN

This option is only relevant for parallel execution (see below). It forces each processor to encode a block of N frames, where N must be a multiple of the pattern length. Since the first frame in any pattern is an I-frame, this forces each block encoded by a processor to begin with an I-frame.

foo

NOTES

If the BASE_FILE_FORMAT is YUV, then the parameter file must contain:

YUV_SIZE <w>x<h>

where w = width, h = height (in pixels) of image, and

YUV_FORMAT <ABEKAS or PHILLIPS or UCB or EYUV or pattern>.

See the file doc/INPUT.FORMAT for more information.

If the -combine-gops option is used, then only the YUV_SIZE and OUTPUT values need be specified in the parameter file. In addition, the parameter file may specify input GOP files in the same manner as normal input files -- except instead of using INPUT_DIR, INPUT, and END_INPUT, use GOP_INPUT_DIR, GOP_INPUT, and GOP_END_INPUT. If no input GOP files are specified, then the default is to use the output file name with suffix ".gop.<gop_num>" starting from 0 as the input files.

If the -combine-frames option is used, then only the YUV_SIZE, GOP_SIZE, and OUTPUT values need be specified in the parameter file. In addition, the parameter file may specify input frame files in the same manner as normal input files -- except instead of using INPUT_DIR, INPUT, and END_INPUT, use FRAME_INPUT_DIR, FRAME_INPUT, and FRAME_END_INPUT. If no input frame files are specified, then the default is to use the output file name with suffix ".frame.<frame_num>" starting from 0 as the input files.

Any number of spaces and tabs may come between each option and value. Lines beginning with '#' are ignored. Any other lines are ignored except for those between INPUT and END_INPUT. This allows you to use the same parameter file for normal usage and for -combine_gops and -combine_frames.

The encoder is case-sensitive so, except for file names and directories, everything should be in upper case.

The lines may appear in any order, except the following exceptions. INPUT must appear before END_INPUT (also, GOP_INPUT before GOP_END_INPUT and FRAME_INPUT before FRAME_END_INPUT). All lines between INPUT and END_INPUT must be the frames in play order.

The encoder is prepared to handle up to 16 B frames between reference frames when encoding with input from stdin. To increase this amount, change the constant B_FRAME_RUN in frame.c and recompile.

PARALLEL OPERATION

The encoder may be run on multiple machines at once. To do so, add a line "PARALLEL" in the parameter file, followed by a listing, one machine per line, then "END_PARALLEL". Each of the lines should be in one of two forms. If the machine has access to the file server, then the line should be:

```
<machine> <user> <executable>
```

The executable is normally ppmtompeg (you may need to give the complete path if you've built for different architectures). If the machine is a remote machine, then the line should be:

```
REMOTE <machine> <user> <executable> <parameter file>
```

Full paths should generally be used when describing executables and parameter files. This INCLUDES the parameter file given as an argument to the original call to ppmtompeg. Also, .rhosts files on the appropriate machines should have the appropriate information.

The encoder will use the original machine for the master and I/O server processes, and uses the listed machines as slaves to do the computation.

Optional lines are

RSH <remote shell command>

The encoder uses the remote shell command to start processes on other machines. The default command is 'rsh.' If your machine supports a different command, specify it here.

PARALLEL_TEST_FRAMES <n>

n is the number of frames to encode initially on each processor

PARALLEL_TIME_CHUNKS <t>

subsequently, each slave processor will be asked to encode for approximately t seconds. Smaller values of <t> increase communication, but improve load balancing.

The default values for these two options are $n = 3$ frames and $t = 30$ seconds.

PARALLEL_PERFECT

If this line is present, then scheduling is done on the assumption that work distribution will be perfectly even -- meaning that each machine is about the same speed. The frames will simply be divided up evenly between the processors. This has the advantage of very minimal scheduling overhead, but is obviously wrong if machines have varying speeds, or if the network load makes performance uneven.

VERSION

This is version 1.5 it contains new features and bug fixes from version 1.3.

BUGS

Not really a bug, but at least a limitation: If writing to an output file, ppmtompeg sometimes uses `<filename>.*` as temporary files.

No known bugs, but if you find any, report them to mpeg-bugs@plateau.cs.berkeley.edu.

AUTHORS

Kevin Gong - University of California, Berkeley, keving@cs.berkeley.edu

Ketan Patel - University of California, Berkeley, kpatel@cs.berkeley.edu

Dan Wallach - University of California, Berkeley, dwallach@cs.berkeley.edu

Darryl Brown - University of California, Berkeley, darryl@cs.berkeley.edu

Eugene Hung - University of California, Berkeley, eyhung@cs.berkeley.edu

Steve Smoot - University of California, Berkeley, smoot@cs.berkeley.edu