An Industry Standard Benchmark Consortium

DENBench™ Version 1.0 MPEG-2 Decode

Highlights

- **Benchmarks potential performance of an MPEG-2 Decoder**
- **Five different test files stress different decoder aspects**
- **Integer implementation**
- **Implements PSNR to check output quality**
- **Based on the ISO reference source**

Application

The MPEG-2 Decoder benchmark provides an indication of the potential performance of a microprocessor subsystem running an MPEG-2 Decoder application, such as those found in a DVD player or a digital set-top box.

Benchmark Description

The benchmark contains a fixed point (integer) implementation of the MSSG ISO sources. The MPEG-2 Decoder uses a standard reference implementation of the core algorithm, including Huffman decoding and modified Inverse Discrete Cosine Transform (iDCT) routines.

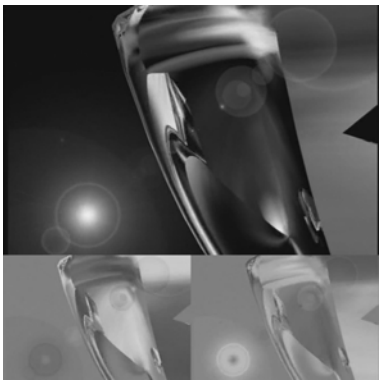
The fixed-point implementation base algorithms of `fdctint.c` and `jfdctint.c` are based on C. Loeffler, A. Ligtenberg and G. Moschytz, “Practical Fast 1-D DCT Algorithms with 11 Multiplications,” Proc. Int’l. Conf. on Acoustics, Speech, and Signal Processing 1989 (ICASSP '89), pp. 988-991.

The `libjpeg jfdctfst.c` algorithm is based on Arai, Agui, and Nakajima’s algorithm for scaled DCT. Their original paper (Trans. IEICE E-71(11):1095) is in Japanese, but the algorithm is described in the Pennebaker and Mitchell’s *JPEG Still Image Data Compression Standard* textbook.

The benchmark’s input data is a series of .MPEG files, and the output is a series of .PPM files, which can be viewed using any suitable graphic file viewer. Correctness is checked by Cyclical Redundancy Checksum (CRC checking); quality is measured using Peak Signal to Noise Ratio analysis (PSNR). CRC is used as a checkpoint only, not as a canonical validation. Out-of-the-box certifications, most of the time and for most compilers, will have the same CRC values.

The datasets are a superset of the MPEG-2 Encode datasets in terms of the number of frames processed.

Description of Datasets



Graphic

Graphic is a black background ray-traced sequence with reflections, combined with moving light sources with coronas.

The primary elements are the reflections, a secondary halo from the first light source, and a few small artifacts on the front-most graphic.

It is derived from an MPEG transport stream of encapsulated video.

The MPEG 2 parameter file is set to NTSC source parameters.

SEQUENCE MPEG2 MP@ML 720x480 chroma 360x240 fps 30

maxBps 1000000 vbv 229376

Picture 720x480 display 720x480 pixel 8x9

A sequence of 50 frames is used for encoding. This results in a 3 second run, and keeps the RAM file requirements under 4 megabytes. The resulting MPEG file size is 232,677 bytes.

Railgrind

Railgrind is a sequence of a skateboarder doing a grind move down a handrail and landing in an open space. The camera is centered on the skateboarder, which results in a fast moving color background.

The artifacts to watch for are tearing of the lower background at the bottom part of the rail move.

The original file is an MPEG system stream with video on channel 0.

SEQUENCE MPEG2 MP@ML PROG 320x240 chroma 160x120

fps 25 maxBps 100000 vbv 65536

Picture 320x240 display 320x240 pixel 1x1

A sequence of 30 frames is used for railgrind decoding.

Sign

"Sign" shows a person using sign language. There is a zoom-in effect to the speaker, with a complex color background of people.

It is derived from an MPEG system stream with video on channel 1.

Artifacts may appear as small color blocks appearing in the bottom lines of the picture.

The original input dimensions are 352x256, however the decoder only correctly decodes this with image size at 352x240

SEQUENCE MPEG2 MP@ML PROG 352x240 chroma 176x120 fps 25

maxBps 95000 vbv 32768

Picture 352x240 display 352x240 pixel 1x1

300 frames of sign are decoded.

Zoom

Zoom is a beach scene with a rapid zoom-out effect. The original input was an AVI file, extracted to bitmaps. These bitmaps were converted to PPM files, and re-decoded at 30 fps. The final YUV files were generated from the decoded file.

SEQUENCE MPEG2 MP@ML PROG 320x240 chroma 160x120

fps 30 maxBps 95000 vbv 65536

Picture 320x240 display 320x240 pixel 1x1

65 frames are used for decoding.

Marsface

Marsface is a rotating black and white radar picture of a Mars feature. The feature is 3 dimensional with a perspective view. The original is a 24-fps MPEG file. This file format is maintained for the decoder. For encoding, the bitrate is increased as well as the fps. An fps of 25 is the closest available setting in the decoder.

Original Attributes:

SEQUENCE PROG 192x192 chroma 96x96 fps 24

maxBps 0 vbv 32768

picture 192x192 display 192x192 pixel 1x1

Generated attributes:

SEQUENCE MPEG2 MP@ML PROG 192x192 chroma 96x96

fps 25 maxBps 95000 vbv 65536

picture 192x192 display 192x192 pixel 1x1

All 49 frames are used for decoding.

Benchmark Processing

Processing consists of:

1. Read the MPEG-2 file
2. Read and interpret the header information
3. Read and decode frames of data
4. Process the data based on the header information
5. Output the .PPM file into memory
6. Calculate a PSNR value

A single iteration of the benchmark is complete when the end of the input file is reached and no more data is available to be processed.

Output quality is measured using Peak Signal to Noise Ratio (PSNR) code developed by EEMBC. PSNR is a decibel measurement of noise power and is consistent for the industry, and widely used to measure picture and audio quality. PSNR is measured outside the benchmark timing loop.

Analysis of Computing Resources

This benchmark concentrates mostly on computational processing rather than file I/O, with the key algorithms being the inverse discrete cosine transform.