

Commit

✖ Port the Huffman lookup table size fix from brunsl. (#3871)

Browse files

See also: https://www.youtube.com/watch?v=_ACCK0AUQ8Q&t=696s

main (#3871)

szabadka authored on Oct 4 Verified

1 parent 42422b4 commit 9cc451b

Showing 2 changed files with 24 additions and 8 deletions.

Whitesp...

Ignore whitespace

Split

Unifi...

Filter changed files

lib

jpegli

huffman.h

jxl/jpeg

enc_jpeg_huffman...

lib/jpegli/huffman.h

```
15 | 15 |
16 | 16 |     constexpr int kJpegHuffmanRootTableBits = 8;
17 | 17 |     // Maximum huffman lookup table size.
18 | 18 |     - // According to zlib/examples/enough.c, 758
19 | 19 |     - // an alphabet of 257 symbols (256 + 1 special
20 | 20 |     - // max bit length 16 if the root table has 8 bits.
21 | 21 |     - constexpr int kJpegHuffmanLutSize = 758;
18 | 18 |     + // Requirements: alphabet of 257 symbols (256 + 1
19 | 19 |     + // code) and max bit length 16, the root table has
20 | 20 |     + // zlib/examples/enough.c works with an assumption
21 | 21 |     + // "complete". Input JPEGs might have this
22 | 22 |     + // following sum is used as estimate:
23 | 23 |     + // + number of 1-st level cells
24 | 24 |     + // + number of symbols
25 | 25 |     + // + asymptotic amount of repeated 2nd level
26 | 26 |     + // The third number is 1 + 3 + ... + 255 i.e. it
27 | 27 |     + // each "size" might be almost completely be
28 | 28 |     + // Total sum is slightly less than 1024,...
29 | 29 |     + constexpr int kJpegHuffmanLutSize = 1024;
22 | 30 |
23 | 31 |     struct HuffmanTableEntry {
```

```
24      32      uint8_t bits;    // number of bits used for this
                    symbol
```

✓ ↕ 16  lib/jxl/jpeg/enc_jpeg_huffman_decode.h 

```
15      15
16      16      constexpr int kJpegHuffmanRootTableBits = 8;
17      17      // Maximum huffman lookup table size.
18      18      - // According to zlib/examples/enough.c, 758
                    entries are always enough for
19      19      - // an alphabet of 257 symbols (256 + 1 special
                    symbol for the all 1s code) and
20      20      - // max bit length 16 if the root table has 8 bits.
21      21      - constexpr int kJpegHuffmanLutSize = 758;
                    18 + // Requirements: alphabet of 257 symbols (256 + 1
                    special symbol for the all 1s
                    19 + // code) and max bit length 16, the root table has
                    8 bits.
                    20 + // zlib/examples/enough.c works with an assumption
                    that Huffman code is
                    21 + // "complete". Input JPEGs might have this
                    assumption broken, hence the
                    22 + // following sum is used as estimate:
                    23 + // + number of 1-st level cells
                    24 + // + number of symbols
                    25 + // + asymptotic amount of repeated 2nd level
                    cells
                    26 + // The third number is 1 + 3 + ... + 255 i.e. it
                    is assumed that sub-table of
                    27 + // each "size" might be almost completely be
                    filled with repetitions.
                    28 + // Total sum is slightly less than 1024,...
                    29 + constexpr int kJpegHuffmanLutSize = 1024;
22      30
23      31      struct HuffmanTableEntry {
24      32      // Initialize the value to an invalid symbol so
                    that we can recognize it
```

0 comments on commit `9cc451b`

Please [sign in](#) to comment.