

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...

16 lib/jpegli/huffman.h

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

24	32	<code>uint8_t bits; // number of bits used for this symbol</code>
----	----	---

✓ ↕ 16

lib/jxl/jpeg/enc_jpeg_huffman_decode.h

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

0 comments on commit 9cc451b

Please [sign in](#) to comment.