Compressing High Dynamic Range Photos Using the JPEG Standard

Amir Said
Media Technologies Laboratory
HP Laboratories Palo Alto
HPL-2007-96
June 7, 2007*

compression, consumer photography, standards, imaging

In the consumer digital photography market we have nearly all images stored in the JPEG compression format. While many new compression methods were proposed for providing somewhat better compression, and also a variety of new features, most users currently do not see compelling reasons for adopting new formats. The high dynamic range feature seems to be an exception, standing out in its appeal to consumers, because it can effectively improve photo quality. This feature is not supported by the baseline mode of JPEG, since it requires more than 8 bits of precision for light and color intensities. What is not well known is that the JPEG standard has one extended mode allowing 12-bit precision, which can be implemented with very small changes to current software and hardware, because the standard uses a technique that allows complexity to grow linearly instead of exponentially with the number of bits. We consider how this extended JPEG mode can enable high dynamic range photography in the consumer market, with much smaller investments than required for adopting a completely new format. We explain the technical differences among modes, what are the required changes, and the consequences in terms of complexity and compression efficiency.

* Internal Accession Date Only
Approved for External Publication

Compressing High Dynamic Range Photos Using the JPEG Standard

Amir Said
Media Technologies Lab
OST – HP Labs, Palo Alto, CA

Abstract

In the consumer digital photography market we have nearly all images stored in the JPEG compression format. While many new compression methods were proposed for providing somewhat better compression, and also a variety of new features, most users currently do not see compelling reasons for adopting new formats. The high dynamic range feature seems to be an exception, standing out in its appeal to consumers, because it can effectively improve photo quality. This feature is not supported by the baseline mode of JPEG, since it requires more than 8 bits of precision for light and color intensities. What is not well known is that the JPEG standard has one extended mode allowing 12-bit precision, which can be implemented with very small changes to current software and hardware, because the standard uses a technique that allows complexity to grow linearly instead of exponentially with the number of bits. We consider how this extended JPEG mode can enable high dynamic range photography in the consumer market, with much smaller investments than required for adopting a completely new format. We explain the technical differences among modes, what are the required changes, and the consequences in terms of complexity and compression efficiency.

1 Introduction

The consumer digital photography market continues to grow at a very fast pace [13]. However, since this market remains very competitive, with great pressure on camera prices on all market segments, the equipment manufacturers are quite reluctant to adopt any new image processing feature that is not recognized as highly desirable by users.

Currently nearly all consumer images are in the JPEG compressed image format, and market research shows that most users prefer camera settings corresponding to high quality and moderate compression (from 20:1 to 6:1). Because of plummeting costs of flash memory, even with this reduced amount of compression it is common for users to have more in-camera memory than they are able to fill. So, it is becoming increasingly harder to justify changing to new formats that provide only better compression.

When we consider other market trends, there is still a consistent desire for more image resolution, but its growth has been constrained by technological problems concerning noise levels and the cost of optical elements. There are some other features supported by new image formats, which are very attractive in some specialized imaging fields (e.g., medical, satellite), but failed to create enough interest or are are not really suited for consumer applications.

A more interesting feature for the consumer digital photography market is the support for higher dynamic ranges, using more than 8 bits of precision for light intensity and color information. It is particularly appealing because it can improve the quality of digital photos in scenes with wide ranges of illumination, preserving contrast information nearly as faithfully as silver halide film [5].

In Section 2 we provide some more information about high dynamic range photography, and in Section 3 we start discussing how it can be supported using the JPEG compression standard. We briefly present the characteristics of the different coding processes specified by the standard, noting that only what is called *baseline* mode was widely accepted. In fact, the expression "compressed with JPEG" almost always means compressed using its baseline process.

We next present, in Section 4, what is called a sequential DCT-based extended process, which supports 12 bit samples, and is the most similar to the baseline process. In Section 5 we explain what changes are required in a JPEG implementation for moving from 8-bit samples to 12-bit samples, and in Section 6 we discuss the compression efficiency. In Section 7 we present the conclusions.

2 High Dynamic Range Photography

Since the human eye has a limited capability of distinguishing different light intensity levels, digital imaging displays have been commonly designed to support only up to 256 different levels in each primary color [4]. Digital cameras have also been designed for that range and precision, assuming that it would be inefficient to have an acquisition device with higher precision than that of the rendering device.

The main problem with this approach is that the current limits in range and precision can cause a significant loss of image information. In scenes with both very brightly and dimly lighted areas, the camera will choose some intermediary range, losing information in both the upper and lower range of light intensities. Professional photographers deal with this problem by carefully analyzing the scene, doing several light measurements, and if possible, trying different exposures.

This, of course, does not fit the "one-click" approach needed for consumer photography. The software embedded in digital camera tries to identify the best exposure, but it cannot always decide correctly, and in some cases loss is inevitable. For the customer, that means an opportunity for a great shot that is forever lost.

Similarly, images are commonly represented in the RGB color space, which has been created to match the gamut of CRT displays. Colors outside this gamut, which can now be cheaply reproduced using ink jet technology, correspond to RGB values outside the range [0,255].

The solution is to increase the precision of the light intensity measurements, using more that 8 bits of precision per primary color, creating what is called *high dynamic range* (HDR) photos [5]. There are many approaches for creating HDR photos, and their analysis is beyond the scope of this document (see, for instance, [7, 8, 10, 12] and their references). Here we are mainly interested on studying how HDR photography can be supported, with minimum impact on current consumer imaging systems, equipment, and workflows.

3 The JPEG Standard Processes

The JPEG standard for image compression [2] has been an important factor in the growth of consumer digital photography, and it estimated that currently we have hundreds of billions of images [13] in that format. It provides specifications for four different coding processes [2, §4.11], as shown in Table 1. However, for a variety of reasons only the baseline compression mode has been widely used, and the features supported by the other processes are not well-known.

For instance, it is commonly believed that the extended processes are only for supporting features like progressive decoding, and employ arithmetic codes (which used to require royalty payments). However, we can see in Table 1 (box II) that this process also has a mode that uses the same form of sequential decoding with Huffman coding as the baseline mode.

There is also a common misconception regarding how JPEG defines the compression of color

Table 1: The four compression processes specified by the JPEG standard.

I – Baseline process (required for all DCT-based decoders)
• DCT-based process
• Source image: 8-bit samples within each component
• Sequential decoding
• Huffman coding: 2 AC and 2 DC tables
• Decoders shall process scans with 1, 2, 3, and 4 components
• Interleaved and non-interleaved scans

II – Extended DCT-based processes
• DCT-based process
• Source image: 8-bit or 12-bit samples
• Sequential or progressive decoding
• Huffman or arithmetic coding: 4 AC and 4 DC tables
• Decoders shall process scans with 1, 2, 3, and 4 components
• Interleaved and non-interleaved scans

III – Lossless processes
• Predictive process (not DCT-based)
• Source image: from 2-bit to 16-bit samples
• Sequential decoding
• Huffman or arithmetic coding: 4 DC tables
• Decoders shall process scans with 1, 2, 3, and 4 components
• Interleaved and non-interleaved scans

IV – Hierarchical processes
• Multiple frames (non-differential and differential)
• Uses extended DCT-based or lossless processes
• Decoders shall process scans with 1, 2, 3, and 4 components
• Interleaved and non-interleaved scans

images. In reality, the standard does not specify anything about color components, or color spaces, transforms, or profiles. It simply defines how up to four images are to be compressed together, with different choices of subsampling factors.

The Exif standard, created by the Japan Electronic Industry Development Association, defines the image file format that is currently used in consumer digital cameras, using the baseline JPEG compression, and a color transformation from RGB to YCrCb [6]. It also specifies some JPEG compression choices, like the order of the color components, the chrominance subsampling conventions, and how the images are interleaved.

Starting from the baseline process, we can consider what extended process would support the larger number of bits per sample required for high dynamic range photography, with minimum changes. The options are shown in Annex F of the JPEG standard [2]:

1. baseline sequential;
2. extended sequential, Huffman coding, 8-bit sample precision;
3. extended sequential, arithmetic coding, 8-bit sample precision;
4. extended sequential, Huffman coding, 12-bit sample precision;
5. extended sequential, arithmetic coding, 12-bit sample precision.

The answer is the DCT-based processes for sequential decoding that use Huffman coding, which are able to support images with 8-bit or 12-bit samples, i.e., modes 1, 2, and 4.

Note that Mode 2 is a simple extension to the baseline, allowing different Huffman codes for each of the components (normally luminance plus two chrominance components). Mode 4 is the extension to 12-bit samples, which also allows different Huffman codes for each image.

4 The Baseline and Extended Sequential-DCT Modes

Figure 1 shows the main stages that are shared by all the DCT-based JPEG compression processes with Huffman coding and sequential decoding. As explained before, the color transform is not really a part of the JPEG standard, but the *de facto* standard is a transformation from RGB to YCrCb according to the Exif standard [6]. The following stage corresponds to the organization of image pixels in blocks. The minimum coded unit (MCU) is composed of sets of 8×8 pixels, organized according to the scheme used for chrominance [2, Annex A.2].

The next stages are the computation of the discrete cosine transform (DCT) on blocks of 8×8 pixels, and the quantization of the DCT coefficients. These stages are commonly implemented together because the divisions required by the quantization stage are avoided by scaling the last set of multiplications in the DCT.

The last two stages correspond to the entropy coding process. Not all quantized DCT coefficients are coded in the same way. The difference between DC coefficients in adjacent blocks is coded first, and then the AC coefficients are coded following a zigzag scan.

The most interesting feature of the JPEG entropy coding method is that the DCT coefficient values (signed integers) are never used directly as symbols in a Huffman code. Instead, the values are coded in two steps. First, the *category* (SSSS) of the value (cf. Table 2) is compressed using

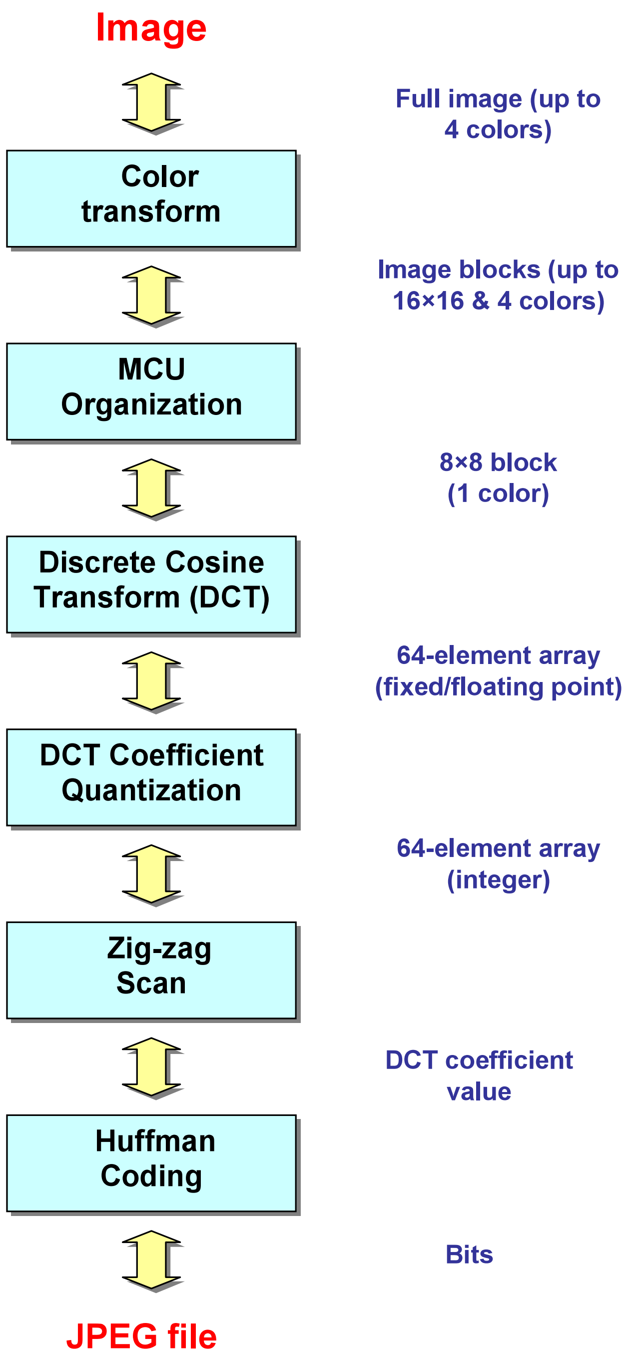

Figure 1: Main stages of JPEG compression, sequential DCT-based modes with Huffman coding.

Table 2: Definition of symbol categories used for coding DCT coefficients.

Category (SSSS)	Data Range	Application
0	$\{0\}$	A, B, C, D
1	$\{-1\} \cup \{1\}$	A, B, C, D
2	$\{-3,-2\} \cup \{2,3\}$	A, B, C, D
3	$\{-7,\ldots,-4\} \cup \{4,\ldots,7\}$	A, B, C, D
4	$\{-15,\ldots,-8\} \cup \{8,\ldots,15\}$	A, B, C, D
5	$\{-31,\ldots,-16\} \cup \{16,\ldots,31\}$	A, B, C, D
6	$\{-63,\ldots,-32\} \cup \{32,\ldots,63\}$	A, B, C, D
7	$\{-127,\ldots,-64\} \cup \{64,\ldots,127\}$	A, B, C, D
8	$\{-255,\ldots,-128\} \cup \{128,\ldots,255\}$	A, B, C, D
9	$\{-511,\ldots,-256\} \cup \{256,\ldots,511\}$	A, B, C, D
10	$\{-1023,\ldots,-512\} \cup \{512\ldots,1023\}$	A, B, C, D
11	$\{-2047,\ldots,-1024\} \cup \{1024,\ldots,2047\}$	A, C, D
12	$\{-4095,\ldots,-2048\} \cup \{2048,\ldots,4095\}$	C, D
13	$\{-8191,\ldots,-4096\} \cup \{4096,\ldots,8191\}$	C, D
14	$\{-16383,\ldots,-8192\} \cup \{8192,\ldots,16383\}$	C, D
15	$\{-32767,\ldots,-16384\} \cup \{16384,\ldots,32767\}$	C

Table 3: Situations in which the symbol categories are applied.

Application	Data type	Sample precision
A	DC difference	8 bits
B	AC coefficient	8 bits
C	DC difference	12 bits
D	AC coefficient	12 bits

Table 4: Number of symbols used by the Huffman code in the baseline and extended modes.

Sample precision	DC difference	AC value & run length
8 bits	12 symbols	176 symbols
12 bits	16 symbols	240 symbols

an optimized Huffman code. Next, an integer number of bits, containing a sign bit and an offset, is used to code the information necessary to identify the original DCT coefficient value.

This corresponds to a technique that we call *symbol grouping* [9, 11]. It allows great reductions in the complexity of entropy coding, with very little loss in compression efficiency. In our analysis, the important characteristic is observed in Table 2, where the column "Application" refers to the type of data that is coded, as shown in Table 3. In all cases the number of categories SSSS is quite small, compared to the possible number of DC and AC values. In addition, it grows logarithmically with the range of those values.

For example, when coding DC differences, the SSSS category can be between 0 and 11 in the baseline mode, and between 0 and 15 in the extended mode for 12-bit samples. Consequently, the sizes of the Huffman codes in each case are 12 and 16, respectively. For AC values, the SSSS category is combined with a 4-bit run-length of preceding zeros, before coding, so we have $16 \times 11 = 176$ Huffman codewords in the 8-bit case, and $16 \times 15 = 240$ codewords in the 12-bit case.

Table 4 shows a summary. Note that changing the number of sample bits from 8 to 12 increases the number of codewords (in the Huffman code) by a factor smaller than 1.5 (linearly), instead of a factor of 16 (exponentially).

5 Changes in Complexity

In this section we analyze in more depth how the complexity of the JPEG encoding and decoding processes changes as we move from the baseline mode with 8-bit samples to the extended mode with 12-bit samples. Note that we consider only the sequential order with Huffman coding. We follow the sequence of stages shown in Figure 1, from top to bottom.

5.1 Color Transformation

The color transformation is outside the JPEG standard. It defines the amount of compression that can be achieved, but is independent of the compression method. It needs to be studied when a standard for high dynamic range photos is created (similar to the Exif standard).

5.2 MCU Organization

The file header, coding conventions, and sequence in which the minimum coded units (MCU) are coded are the same in the baseline mode and in the extended mode with sequential decoding.

5.3 Discrete Cosine Transform and Quantization

In all JPEG implementations it is necessary to compute the discrete cosine transform (DCT) during compression, and its inverse during decompression. There is a variety of fast methods for computing the DCT using floating-point or integer (for fixed-point) arithmetic [1], but the details are outside the scope of this document.

For supporting 12-bit samples in software, no changes in the fast DCT algorithms are required for floating-point operations, which are very efficient in all the processors in current personal computers. Only in embedded applications we may have processors that need to use only integer arithmetic for DCT computations. In those cases, we should observe that the first JPEG implementations were done on 16-bit processors, which had very slow multiplications. Consequently, many techniques were developed for carefully controlling the precision of operations required for computing the DCT, and for obtaining approximate but very fast transforms. These techniques can be easily extended now to any processor capable of 32-bit integer arithmetic, in order to support 12 bit precision.

Hardware implementations of the DCT employ fixed-point integer arithmetic, and the extension to 12-bits requires the addition of four more bits of precision for additions and multiplications. It is important to note that in the DCT computation most of the multiplications are between a constant and a variable. This is simpler to implement that general multiplications, because the constant value can be "hardwired." Only one stage, which combines scaling with quantization, uses multiplication between two variables.

5.4 Scan Order and Category Computation

After the DCT computation and quantization, the extended JPEG mode uses the same zigzag scan used by the baseline mode. Before coding a DC or AC coefficient, it is necessary to compute its category SSSS, as shown in Figure 2. In a software implementation this is normally done using table look-up. For computing the category with a single table read, the tables should contain 2,048 and 32,768 elements, for 8 and 12 bits, respectively. The same computation can be done using a single table with 128 elements, by adding a single comparison, as

```
if (n < 128)
  then return SSSS_Table[n];
  else return 8 + SSSS_Table[n/256];
```

where `SSSS_Table` contains the category values, and the division by 256 is done using bit shifts.

A hardware implementation can use table look-up or a logic circuit to identify the position of the most significant bit with value 1. In the Intel processors the Bit Scan Reverse (BSR) assembler instruction can be used for quickly identifying the category.

The conversion from category and bits to signed integers is even simpler, requiring only bit shifts and addition.

5.5 Huffman Coding

The encoding stage is basically done using table look-up, with one table with codeword lengths, and another with codeword bit values. Table 4 shows that the number of codewords required for each type of code is always quite small. Note that the standard allows different codes per image (or image component), but they are all of the same type.

Decoding is also commonly done using table look-up too, but with larger tables. The JPEG standard constrains the codeword lengths to no more than 16 bits, in both the baseline and extended modes. This is very convenient, since it guarantees that the decoding table does not need more than 65,536 elements. It is also easy to implement a hybrid approach, with table look-up decoding for codewords of up to, say, 10 bits (and 1024 table elements), and tree-based decoding for the remaining codewords. Note that the complexity of these approaches depend only on the codeword's 16-bit limit, and not on the sample precision.

The main conclusion is that in software or hardware implementations of the baseline process, it is necessary to support two types of Huffman codes, and their corresponding tables. Changing from 8-bit samples to 12-bit samples requires only small increases (of about 33%) in the sizes of the tables with codeword lengths, and codeword bit values. The size of the main decoding table, on the other hand, may not have to be changed, and its maximum size is 65,536 elements.

Of course, with 12-bit samples we can have more binary data following a Huffman-coded symbol, but it is in fixed-length binary format, i.e., they correspond to simple binary reads and writes.

6 Compression Efficiency

It is hard to compare the compression efficiency of the extended JPEG format to new image compression formats like JPEG2000, without a more specific definition of how the high dynamic range images are created, color spaces, etc. However, there are some facts about JPEG and compression methods in general that can be considered.

First, theoretical work [9, 11] has shown that the categorization coding technique (or symbol grouping) used in JPEG is quite robust to changes in the type of images coded. Furthermore, experimental results have shown that it is effective in a wide range of compression ratios, from low quality to lossless on 10 bpp images [3]. Thus, we can say that JPEG's coding method is not "fine-tuned" only for 8 bpp images, and should be as effective for 12 bpp images.

The second factor to take into account is that highly compressed images (low quality) have very little value in the consumer market. Consequently, consumer digital photos are much more commonly compressed with quality settings that do not produce the dreaded DCT blocking artifacts. Since the justification for high dynamic range is higher image quality, it does not make sense to make comparisons between compression methods at low quality settings.

The final aspect to be considered is that compression gains are harder to obtain with lower compression ratios, since a significant part of the compressed data is composed of image details that are hard to compress. This fact is well-known for lossless image compression, when even very sophisticated coding methods cannot improve compression beyond a few percent.

7 Conclusions

We analyzed the fact that in the current digital photography market it is quite difficult to introduce new compression formats, unless there is strong interest by consumers. We analyze the implications of adopting one feature that may be more promising, high dynamic range photography, since it can improve quality and preserve contrast information nearly as well as silver halide film.

Since high dynamic range photos need more than 8 bits, it cannot be supported by the universally used baseline mode of the JPEG compression standard. However, we have shown that this feature can be supported by an extended JPEG mode for 12-bit image samples, which is very similar to the baseline mode.

We explain that, thanks to the fact that JPEG uses a very effective technique for reduction of the entropy coding complexity, an increase on the sample precision from 8 to 12 bits translates in at most only a 36% increase in the tables required for implementing the Huffman codes.

References

[1] K. Rao and P. Yip, *Discrete Cosine Transform,* Academic Press, 1990.

[2] International Telecommunication Union (ITU), *Digital Compression and Coding of Continuous-Tone Still Image: Requirements and Guidelines*, ITU-T Recommendation T.81, Geneva, Switzerland, 1992 (JPEG standard).
(http://www.w3.org/Graphics/JPEG/itu-t81.pdf)

[3] A. Said and W.A. Pearlman, "Low-complexity waveform coding via alphabet and sample-set partitioning," *International Symposium on Visual Communications and Image Processing,* Proc. SPIE vol. 3024(1), pp. 25–37, Feb. 1997.

[4] R.C. Gonzalez and R.E. Woods, *Digital Image Processing,* 2nd ed., Prentice Hall, 2002.

[5] A. El Gamal, *High-Dynamic-Range Image Sensors,* tutorial presented at *IEEE Int. Solid State Circuits Conf.,* Feb. 2002.
(http://www-isl.stanford.edu/ abbas/group/papers_and_pub/isscc02_tutorial.pdf)

[6] Standard of Japan Electronics and Information Technology Industries Association, *Exchangeable image file format for digital still cameras,* Exif Version 2.2, April 2002.
(http://www.exif.org/Exif2-2.PDF)

[7] A. Pardo and G. Sapiro, "Visualization of high dynamic range images," *IEEE Trans. Image Processing,* vol. 12, pp. 639–647, June 2003.

[8] S. Kavusi and A. El Gamal, "A quantitative study of high dynamic range image sensor architectures," *Proc. SPIE, Electronic Imaging,* vol. 5301, Jan. 2004.

[9] A. Said, *On the Reduction of Entropy Coding Complexity via Symbol Grouping,* Hewlett-Packard Laboratories Report, HPL–2004–145, Palo Alto, CA, Aug. 2004.

[10] G. Qiu and J. Duan, "An optimal tone reproduction curve operator for the display of high dynamic range images," *Proc. IEEE Int. Symp. Circuits and Systems,* vol. 6, pp. 6276–6279, May 2005.

[11] A. Said, "Worst-case Analysis of the Low-complexity Symbol Grouping Coding Technique," *Proc. IEEE Int. Symp. Information Theory,* July 2006.

[12] L. Meylan and S. Süsstrunk, "High dynamic range image rendering with a retinex-based adaptive filter," *IEEE Trans. Image Processing,* vol. 15, pp. 2820–2830, Sept. 2006.

[13] Ed Lee, *The Image Storage and Management Dilemma,* InfoTrends Strategic Assessment, Weymouth, MA, Jan. 2007.