Appendix P: JPEG Compression

This appendix, written after the release of Revision 5.0 of the TIFF specification, is still in draft form and is subject to change. Please send any comments to Eric Hamilton of C-Cube Microsystems.

Introduction

Image compression reduces the storage requirements of pictorial data. In addition, it reduces the time necessary for access, communication and display of images. In order to address the standardization of compression techniques an international standards group was formed: the Joint Photographic Experts Group (JPEG). JPEG has as its objective to create a joint ISO/CCITT standard for continuous tone image compression (color and grayscale).

JPEG decided that because of the broad scope of the standard, no one algorithmic procedure was able to satisfy the requirements of all applications. It was decided to specify different algorithmic processes where each process is targeted to satisfy the requirements of a class of applications. Thus the JPEG standard became a "toolkit" whereby the particular algorithmic "tools" are selected according to the needs of the application environment.

The algorithmic processes fundamentally fall into two classes: lossy and lossless. Those based on the Discrete Cosine Transform (DCT) are lossy, and typically provide for substantial compression without significant degradation of the reconstructed image with respect to the source image.

The simplest DCT-based coding process is the baseline process. It provides a capability which is sufficient for most applications. There are additional DCT-based processes which extend the baseline process to a broader range of applications.

The second class of coding processes is targeted for those applications requiring lossless compression. The lossless processes are not DCT-based and are utilized independently of any of the DCT-based processes.

This appendix describes the JPEG baseline and the JPEG lossless processes and the extensions to TIFF defined to support JPEG compression.

JPEG Baseline Process

The baseline process is a DCT-based algorithm which compresses images having 8 bits per sample. The baseline process operates only in sequential mode. In sequential mode, the image is processed from left to right and top to bottom in a single pass by compressing the first row of data followed by the second row and continuing until the end of image is reached. Sequential operation has minimal buffering requirements and thus permits inexpensive implementations.

The JPEG baseline process is an algorithm which inherently introduces error into the reconstructed image and cannot be utilized for lossless compression. The algorithm accepts as input only those images having 8 bits per sample. Images with fewer than 8 bits per sample may be compressed using the baseline process algorithm by left justifying each input sample within a byte before compression.

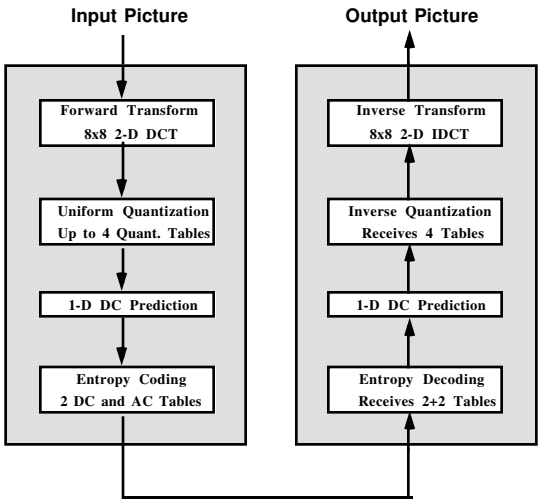

Figure 1. Baseline Process Encoder and Decoder

A functional block diagram of the Baseline encoding and decoding processes is contained in Figure 1. Encoder operation consists of dividing each component of the input image into 8x8 blocks, performing the two-dimensional DCT on each block, quantizing each DCT coefficient uniformly, subtracting the quantized DC coefficient from the corresponding term in the previous block, and finally entropy coding the quantized coefficients using variable length codes (VLCs). Decoding is performed by inverting each of the encoder operations in the reverse order.

The DCT

Before performing the foward DCT input pixels are level-shifted so that they range from -128 to +127. Blocks of 8x8 pixels are transformed with the two-dimensional 8x8 DCT:

$$F(u,v) = \frac{1}{4} C(u)C(v) \sum\sum f(x,y) \cos\frac{\pi(2x+1)u}{16} \cos\frac{\pi(2y+1)v}{16}$$

and blocks are inverse transformed by the decoder with the Inverse DCT:

$$f(x,y) = \frac{1}{4} \sum \sum C(u)C(v) F(u,v) \cos\frac{\pi(2x+1)u}{16} \cos\frac{\pi(2y+1)v}{16}$$

with $u, v, x, y = 0, 1, 2, \ldots 7$
where x, y = spatial coordinates in the pel domain
u, v = coordinates in the transform domain

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u, v = 0 \\ 1 & \text{otherwise} \end{cases}$$

Although the exact method for computation of the DCT and IDCT is not subject to standardization and will not be specified by JPEG, it is probable that JPEG will adopt DCT conformance specifications which designate the accuracy to which the DCT must be computed. The DCT conformance specifications will assure that any two JPEG implementations will produce visually similar reconstructed images.

Quantization

The coefficients of the DCT are quantized to reduce their magnitude and increase the number of zero-value coefficients. The DCT coefficients are independently quantized by uniform quantizers. A uniform quantizer divides the real number line into steps of equal size as shown in Figure 2. The quantization step-size applied to each coefficient is determined from the contents of a 64-element quantization table.

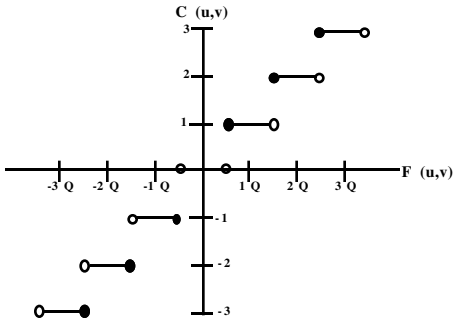

Figure 2. Uniform Quantization

The baseline process provides for up to 4 different quantization tables to be defined and assigned to separate interleaved components within a single scan of the input image. Although the values of each quantization table should ideally be determined through rigorous subjective testing which estimates the human psycho-visual thresholds for each DCT coefficient and for each color component of the input image, JPEG has developed quantization tables which work well for CCIR 601 resolution images and has published these in the informational section of the proposed standard.

DC Prediction

The DCT coefficient located in the upper-left hand corner of the transformed block represents the average spatial intensity of the block and is referred to as the "DC coefficient". After the DCT coefficients are quantized but before they are entropy coded, DC prediction is performed. DC prediction simply means that the DC term of the previous block is subtracted from the DC term of the current block prior to encoding.

Zig-Zag Scan

Prior to entropy coding, the DCT coefficients are ordered into a one-dimensional sequence according to a "zig-zag" scan. The DC coefficient is coded first followed by AC coefficient coding proceeding in the order illustrated in Figure 3.

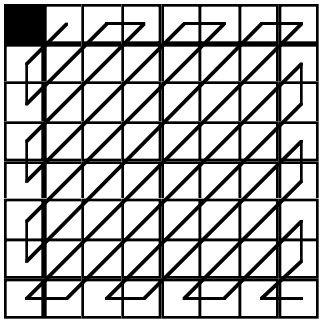

Figure 3. Zig-Zag Scan of DCT Coefficients

Entropy Coding

The quantized DCT coefficients are further compressed using entropy coding. The baseline process performs entropy coding using variable length codes (VLCs) and variable length integers (VLIs).

VLCs, commonly known as Huffman codes, compress data symbols by creating shorter codes to represent frequently occurring symbols and longer codes for occasionally occurring symbols. One reason for using VLCs is that they are easily implemented by means of lookup tables.

Separate code tables are provided for the coding of DC and AC coefficients. The following sections describe the respective coding methods used for coding DC and AC coefficients.

DC Coefficient Coding

DC prediction produces a "differential DC coefficient" which is typically small in magnitude due to the high correlation of neighboring DC coefficients. Each differential DC coefficient is encoded by a VLC which represents the number of significant bits in the DC term followed by a VLI representing the value itself. The VLC is coded by first determining the number of significant bits, SSSS, in the differential DC coefficient through the following table:

SSSS	Differential DC Value
0	0
1	-1, 1
2	-3,-2, 2,3
3	-7..-4, 4..7
4	-15..-8, 8..15
5	-31..-16, 16..31
6	-63..-32, 32..63
7	-127..-64, 64..127
8	-255..-128, 128..255
9	-511..-256, 256..511
10	-1023..-512, 512..1023
11	-2047..-1024, 1024..2047
12	-4095..-2048, 2048..4095

SSSS is then coded from the selected DC VLC table. The VLC is followed by a VLI having SSSS bits which represents the value of the differential DC coefficient itself. If the coefficient is positive, the VLI is simply the low order bits of the coefficient. If the coefficient is negative, then the VLI is the low order bits of the coefficient-1.

AC Coefficient Coding

In a similar fashion, AC coefficients are coded with alternating VLC and VLI codes. The VLC table, however, is a two-dimensional table which is indexed by a composite 8-bit value. The lower 4 bits of the 8-bit value, i.e. the column index, is the number of significant bits, SSSS, of a non-zero AC coefficient. SSSS is computed through the same table as that used for coding the DC coefficient. The higher order 4 bits, the row index, is the number of zero coefficients, NNNN, which precede the non-zero AC coefficient. The

first column of the two-dimensional coding table contains codes which represent control functions. Figure 4 illustrates the general structure of the AC coding table.

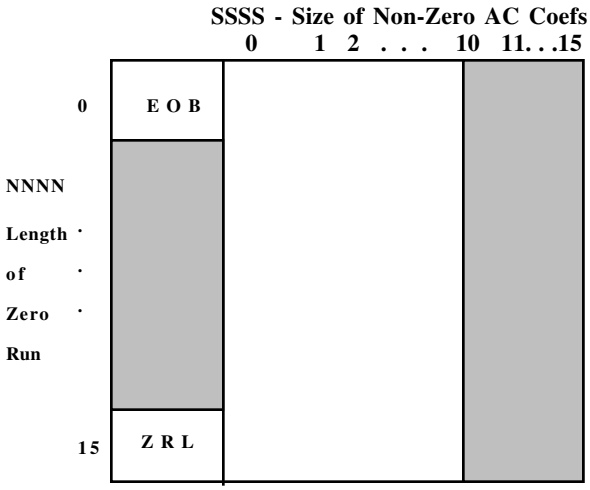

Figure 4. 2-D Run-Size Value Array for AC Coefs
The shaded portions are undefined in the baseline process

The flow chart in Figure 5 specifies the AC coefficient coding procedure. AC coefficients are coded by traversing the block in the zig-zag sequence and counting the number of zero coefficients until a non-zero AC coefficient is encountered. If the count of consecutive zero coefficients exceeds 15, then a ZRL code is coded and the zero run-length count is reset. When a non-zero AC coefficient is found, the number of significant bits in the non-zero coefficient, SSSS, is combined with the zero run-length which precedes that coefficient, NNNN, to form an index into the two-dimensional VLC table. The selected VLC is then coded. The VLC is followed by a VLI which represents the value of the AC coefficient. This process is repeated until the end of block is reached. If the last AC coefficient is zero, then an End of Block (EOB) VLC is encoded.

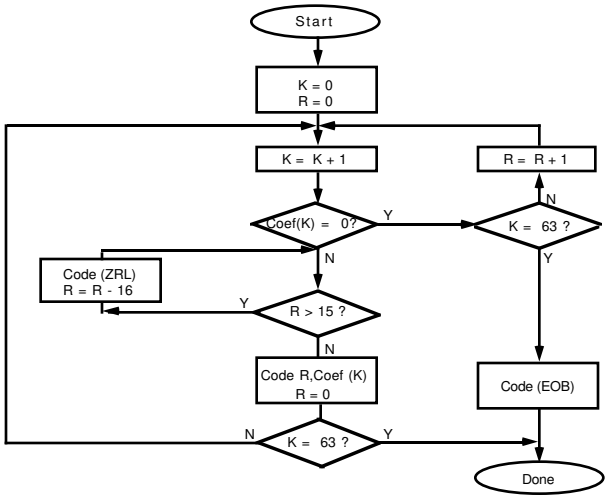

Figure 5. Encoding Procedure for AC Coefs

JPEG Lossless Processes

The JPEG lossless coding processes utilize a spatial prediction algorithm based upon a two-dimensional Differential Pulse Code Modulation (DPCM) technique. They are compatible with a wider range of input pixel precision than the DCT-based algorithms (2 to 16 bits per sample). Although the primary motivation for specifying a spatial algorithm is to provide a method for lossless compression, JPEG allows for quantization of the input data, resulting in lossy compression and higher compression rates.

Although JPEG provides for use of either the Huffman or Arithmetic entropy coding models by the processes for lossless coding, only the Huffman coding model is supported by this version of TIFF. The following is a brief overview of the lossless process with Huffman coding.

Control Structure

Much of the control structure developed for the sequential DCT procedures is also used for sequential lossless coding. Either interleaved or non-interleaved data ordering may be used.

Coding Model

The coding model developed for coding the DC coefficients of the DCT is extended to allow a number of one-dimensional and two-dimensional predictors for the lossless coding function. Each component uses an independent predictor.

Prediction

Figure 6 shows the relationship between the neighboring values used for prediction and the sample being coded.

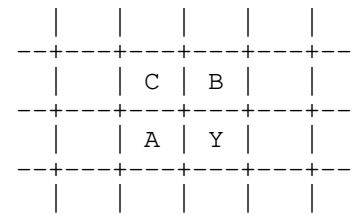

Figure 6. Relationship between sample and prediction samples

Y is the sample to be coded and A, B, and C are the samples immediately to the left, immediately above, and diagonally to the left and above.

The allowed predictors are listed in the following table.

Selection-value	Prediction
0	no prediction (differential coding)
1	A
2	B
3	C
4	A+B-C
5	A+((B-C)/2)
6	B+((A-C)/2)
7	(A+B)/2

Selection-value 0 shall only be used for differential coding in the hierarchical mode. Selections 1, 2 and 3 are one dimensional predictors and selections 4, 5, 6, and 7 are two dimensional predictors. The divide by 2 in the prediction equations is done by a arithmetic-right-shift of the integer values.

The difference between the prediction value and the input is calculated modulo 2**16. Therefore, the prediction can also be treated as a modulo 2**16 value. In the decoder the difference is decoded and added, modulo 2**16, to the prediction.

Huffman Coding of the Prediction Error

The Huffman coding procedures defined for coding the DC coefficients are used to code the modulo 2**16 differences. The table for DC coding is extended to 17 entries which allows for coding of the modulo 2**16 differences.

Point Transformation Prior to Lossless Coding

For the lossless processes only, the input image data may optionally be scaled (quantized) prior to coding by specifying a nonzero value in the point transformation parameter. Point transformation is defined to be division by a power of 2.

If the point transformation field is nonzero for a component, a point transformation of the input is performed prior to the lossless coding. The input is divided by 2**Pt, where Pt is the value of the point transform signaling field. The output of the decoder is rescaled to the input range by multiplying by 2**Pt. Note that the scaling of input and output can be performed by arithmetic shifts.

Overview of the JPEG extension to TIFF

In extending the TIFF definition to include JPEG compressed data, it is necessary to note the following:

- JPEG is effective only on continuous-tone color spaces:

Grayscale	(Photometric Interpretation = 1)	
RGB	(Photometric Interpretation = 2)	
CMYK	(Photometric Interpretation = 5)	(See Appendix K)
YC_bC_r	(Photometric Interpretation = 6)	(See Appendix O)

- Color conversion to YC_bC_r is often used as part of the compression process because the chrominance components can be subsampled and compressed to a greater degree without significant visual loss of quality. Tags are defined to describe how this conversion has taken place and the degree of subsampling employed (see Appendix O).

- New tags have been defined to specify the JPEG parameters used for compression and to allow quantization tables and Huffman code tables to be incorporated into the TIFF file.

- TIFF is compatible with compressed image data which conforms to the syntax of the JPEG interchange format for compressed image data. Tags are defined which may be utilized to facilitate conversion from TIFF to interchange format.

- The PlanarConfiguration Tag is used to specify whether or not the compressed data is interleaved as defined by JPEG. For any of the JPEG DCT-based processes, the interleaved data units are coded 8x8 blocks rather than component samples.

- Alhough JPEG codes consecutive image blocks in a single contiguous bitstream, it is extremely useful to employ the concept of tiles in an image. Appendix L proposes some new tags for tiles, which should be used in place of the older tags for strips. The concept of tiling an image in both dimensions is important because JPEG hardware may be limited in the size of each block that is handled.

- Note that the nomenclature used in the TIFF specification is different from the JPEG Draft International Standarditee Draft (ISO DIS 10918-1) in some respects. The following terms should be equated when reading this appendix:

TIFF name	JPEG DIS name
ImageWidth	Number of Pixels
ImageLength	Number of Lines
SamplesPerPixel	Number of Components
JPEGQTable	Quantization Table
JPEGDCTable	Huffman Table for DC coefficients
JPEGACTable	Huffman Table for AC coefficients

Strips and Tiles

The JPEG extension to TIFF has been designed to be consistent with the existing TIFF strip and tile structures and to allow quick conversion to and from the stream-oriented compressed image format defined by JPEG.

Compressed images conforming to the syntax of the JPEG interchange format can be converted to TIFF simply by defining a single strip or tile for the entire image and then concatenating the TIFF image description fields to the JPEG compressed image data. The strip or tile offset tag points directly to the start of the entropy coded data (not to a JPEG marker).

Multiple strips or tiles are supported in JPEG compressed images using restart markers. Restart markers, inserted periodically into the compressed image data, delineate image segments known as restart intervals. At the start of each restart interval, coding state is reset to default values, allowing every restart interval to be decoded independently of previously decoded data. TIFF strip and tile offsets shall always point to the start of a restart interval. Equivalently, each strip or tile contains an integral number of restart intervals. Restart markers need not be present in a TIFF file; they are implicitly coded at the start of every strip or tile.

In order to maximize interchangeability of TIFF files with other formats, a restriction is placed on tile height for files containing JPEG compressed image data conforming to the JPEG interchange format syntax. The restriction, imposed only when the tile width is defined to be shorter than the image width and when the JPEGInterchangeFormat Tag is present and non-zero, states that the tile height must be equal to the height of one JPEG Minimum Coded Unit (MCU). This restriction ensures that TIFF files may be converted to JPEG interchange format without undergoing decompression.

Extensions to Existing Tags

Compression
Tag = 259 (103 hex)
Type = SHORT
N = 1

This Tag indicates the type of compression used.

1 = No compression
2 = CCITT Group 3
5 = LZW
6 = JPEG
32773 = PackBits

A new value of 6 indicates that the image data in the TIFF file has been JPEG compressed.

New Tags

JPEGProc
Tag = 512 (200 hex)
Type = SHORT
N = 1

This Tag indicates the JPEG process used to produce the compressed data. The values for this tag are defined to be consistent with the numbering convention used in ISO DIS 10918-2. Two values are defined at this time:

1 = Baseline sequential process
14 = Lossless process with Huffman coding

When the lossless process with Huffman coding is selected by this Tag, the Huffman tables used to encode the image are specified by the JPEGDCTables tag, and the JPEGACTables tag is not used.

Values indicating JPEG processes other than those specified above will be defined in the future.

JPEGProc (cont'd)

Not all of the tags described in this appendix are relevant to the JPEG process selected by this Tag. The following table specifies the tags which are applicable to each value defined by this Tag.

Tag Name	JPEGProc =1	JPEGProc =14
JPEGInterchangeFormat	X	X
JPEGInterchangeFormatLength	X	X
JPEGRestart Interval	X	X
JPEGLosslessPredictors		X
JPEGPointTransforms		X
JPEGQTables	X	
JPEGDCTables	X	X
JPEGACTables	X	

This Tag is mandatory whenever the Compression Tag is JPEG (no default).

JPEGInterchangeFormat
Tag = 513 (201 hex)
Type = LONG
N = 1

This Tag indicates whether or not a JPEG interchange format bitstream is present in the TIFF file. If a JPEG interchange format bitstream is present then this Tag points to the Start of Image (SOI) marker code.

If this Tag is zero or not present, a JPEG interchange format bitstream is not present.

JPEGInterchangeFormatLength
Tag = 514 (202 hex)
Type = LONG
N = 1

This Tag indicates the length in bytes of the JPEG interchange format bitstream. This Tag is useful for extracting the JPEG interchange format bitstream without parsing the bitstream.

This Tag is relevant only if the JPEGInterchangeFormat Tag is present and is non-zero.

JPEGRestartInterval
Tag = 515 (203 hex)
Type = SHORT
N = 1

This Tag indicates the length of the restart interval used in the compressed image data. The restart interval is defined as the number of Minimum Coded Units (MCUs) between restart markers.

JPEGRestartInterval (cont'd)

Restart intervals are used in JPEG compressed images to provide support for multiple strips or tiles. At the start of each restart interval, coding state is reset to default values, allowing every restart interval to be decoded independently of previously decoded data. TIFF strip and tile offsets shall always point to the start of a restart interval. Equivalently, each strip or tile contains an integral number of restart intervals. Restart markers need not be present in a TIFF file; they are implicitly coded at the start of every strip or tile.

See the JPEG Draft International Standard (ISO DIS 10918-1) for more information about the restart interval and restart markers.

If this Tag is zero or is not present, the compressed data does not contain restart markers.

JPEGLosslessPredictors

Tag = 517 (205 hex)
Type = SHORT
N = SamplesPerPixel

This Tag points to a list of lossless predictor selection-values, one per component.

The allowed predictors are listed in the following table.

Selection-value	Prediction
1	A
2	B
3	C
4	A+B-C
5	A+((B-C)/2)
6	B+((A-C)/2)
7	(A+B)/2

A, B, and C are the samples immediately to the left, immediately above, and diagonally to the left and above the sample to be coded respectively.

See the JPEG Draft International Standard (ISO DIS 10918-1) for more details.

This Tag is mandatory whenever the JPEGProc Tag specifies one of the lossless processes (no default).

JPEGPointTransforms

Tag = 518 (206 hex)
Type = SHORT
N = SamplesPerPixel

This Tag points to a list of point transform values, one per component. This Tag is relevant only for lossless processes.

JPEGPointTransforms (cont'd)

If the point transformation value is nonzero for a component, a point transformation of the input is performed prior to the lossless coding. The input is divided by 2**Pt, where Pt is the point transform value. The output of the decoder is rescaled to the input range by multiplying by 2**Pt. Note that the scaling of input and output can be performed by arithmetic shifts.

See the JPEG Draft International Standard (ISO DIS 10918-1) for more details.

The default value of this Tag is 0 for each component (no scaling).

JPEGQTables

Tag = 519 (207 hex)
Type = LONG
N = SamplesPerPixel

This Tag points to a list of offsets to the quantization tables, one per component.

Each table consists of 64 BYTES (one for each DCT coefficient in the 8x8 block). The quantization tables are stored in zigzag order.

See the JPEG Draft International Standard (ISO DIS 10918-1) for more details.
It is strongly recommended that, within the TIFF file, each component be assigned separate tables.

This Tag is mandatory whenever the JPEGProc Tag specifies a DCT-based process (no default).

JPEGDCTables

Tag = 520 (208 hex)
Type = LONG
N = SamplesPerPixel

This Tag points to a list of offsets to the DC Huffman tables or the lossless Huffman tables, one per component.

The format of each table is as follows:

> 16 BYTES of "BITS", indicating the number of codes of lengths 1 to 16;
> Up to 17 BYTES of "VALUES", indicating the values associated with those codes, in order of length.

See the JPEG Draft International Standard (ISO DIS 10918-1) for more details.

It is strongly recommended that, within the TIFF file, each component be assigned separate tables.

This Tag is mandatory for all JPEG processes (no default).

JPEGACTables
Tag = 521 (209 hex)
Type = LONG
N = SamplesPerPixel

This Tag points to a list of offsets to the Huffman AC tables, one per component.

The format of each table is as follows:

16 BYTES of "BITS", indicating the number of codes of lengths 1 to 16;
Up to 256 BYTES of "VALUES", indicating the values associated with those codes, in order of length.

See the JPEG Draft International Standard (ISO DIS 10918-1) for more details.

It is strongly recommended that, within the TIFF file, each component be assigned separate tables.

This Tag is mandatory whenever the JPEGProc Tag specifies a DCT-based process (no default).

Minimum Requirements for TIFF with JPEG Compression

The following table shows the minimum requirements of a TIFF file using tiling and containing JPEG data compressed with the Baseline process.

Tag = NewSubFileType (254) Type = Long Length = 1 Value = 0	Single image
Tag = ImageWidth (256) Type = Long Length = 1 Value = ?	
Tag = ImageLength (257) Type = Long Length = 1 Value = ?	
Tag = BitsPerSample (258) Type = Short Length = SamplesPerPixel Value = ?	8 : Monochrome 8,8,8 : RGB 8,8,8 : YCbCr 8,8,8,8 : CMYK
Tag = Compression (259) Type = Long Length = 1 Value = 6	6 : JPEG compression
Tag = PhotometricInterpretation (262) Type = Short Length = 1 Value = ?	0,1 : Monochrome 2 : RGB 5 : CMYK 6 : YCbCr
Tag = SamplesPerPixel (277) Type = Short Length = 1 Value = ?	1 : Monochrome 3 : RGB 3 : YCbCr 4 : CMYK
Tag = XResolution (282) Type = Rational Length = 1 Value = ?	
Tag = YResolution (283) Type = Rational Length = 1 Value = ?	
Tag = PlanarConfiguration (284) Type = Short Length = 1 Value = ?	1 : Block Interleaved 2 : Not interleaved
Tag = ResolutionUnit (296) Type = Short Length = 1 Value = ?	
Tag = TileWidth (322) Type = Short Length = 1 Value = ?	Multiple of 8
Tag = TileLength (323) Type = Short Length = 1 Value = ?	Multiple of 8
Tag = TileOffsets (324) Type = Long Length = Number of tiles Value = ?	
Tag = TileByteCounts (325) Type = Long Length = Number of tiles Value = ?	
Tag = JPEGProc (512) Type = Short Length = 1 Value = ?	1 : Baseline process
Tag = JPEGQTables (519) Type = Long Length = SamplesPerPixel Value = ?	Offsets to tables
Tag = JPEGDCTables (520) Type = Long Length = SamplesPerPixel Value = ?	Offsets to tables
Tag = JPEGACTables (521) Type = Long Length = SamplesPerPixel Value = ?	Offsets to tables

References

[1] Wallace, G., "Overview of the JPEG Still Picture Compression Algorithm", Electronic Imaging East '90.

[2] ISO/IEC DIS 10918-1, “Digital Compression and Coding of Continuous-tone Still Images”, Sept. 1991.