JPEG File Interchange Format Version 1.02

September 1, 1992

Eric Hamilton C-Cube Microsystems 1778 McCarthy Blvd. Milpitas, CA 95035

+1 408 944-6300 Fax: +1 408 944-6314 E-mail: eric@c3.pla.ca.us

JPEG File Interchange Format Version 1.02

Why a File Interchange Format

JPEG File Interchange Format is a minimal file format which enables JPEG bitstream s to be exchanged between a wide variety of platforms and applications. This minimal format does not include any of the advanced features found in the TIFF JPEG specification or any application specific file format. Nor should it, for the only purpose of this simplified format is to allow the exchange of JPEG compressed image s.

JPEG File Interchange Format features

- o Uses JPEG compression
- o Uses JPEG interchange format compressed image representation
- o PC or Mac or Unix workstation compatible
- o Standard color space: one or three components. For three components, YCbCr (CCIR 601-256 levels)
- o APPO marker used to specify Units, X pixel density, Y pixel density, thumbn ail
- o APPO marker also used to specify JFIF extensions
- o APPO marker also used to specify application-specific information

JPEG Compression

Although any JPEG process is supported by the syntax of the JPEG File Interchange Format (JFIF) it is strongly recommended that the JPEG baseline process be used for the purposes of file interchange. This ensures maximum compatibility with all applications supporting JPEG. JFIF conforms to the JPEG Draft International Standard (ISO DIS 10918-1).

The JPEG File Interchange Format is entirely compatible with the standard JPEG int erchange format; the only additional requirement is the mandatory presence of the APPO marker right after the SOI marker. Note that JPEG interchange format requires (as does JFIF) that all table specifications used in the encoding process be coded in the bitstream prior to their use.

Compatible across platforms

The JPEG File Interchange Format is compatible across platforms: for example, it does not use any resource forks, supported by the Macintosh but not by PCs or works tations.

Standard color space

The color space to be used is YCbCr as defined by CCIR 601 (256 levels). The RGB components calculated by linear conversion from YCbCr shall not be gamma corrected (gamma = 1.0). If only one component is used, that component shall be Y.

APPO marker used to identify JPEG FIF

The APPO marker is used to identify a JPEG FIF file. The JPEG FIF APPO marker is mandatory right after the SOI marker.

The JFIF APPO marker is identified by a zero terminated string: "JFIF". The APPO c an be used for any other purpose by the application provided it can be distinguish ed from the JFIF APPO.

The JFIF APPO marker provides information which is missing from the JPEG stream: version number, X and Y pixel density (dots per inch or dots per cm), pixel aspect ratio (derived from X and Y pixel density), thumbnail.

APPO marker used to specify JFIF extensions

Additional APPO marker segment(s) can optionally be used to specify JFIF extension s. If used, these segment(s) must immediately follow the JFIF APPO marker. Decoders should skip any unsupported JFIF extension segments and continue decoding.

The JFIF extension APPO marker is identified by a zero terminated string: "JFXX". The JFIF extension APPO marker segment contains a 1-byte code which identifies the extension.

This version, version 1.02, has only one extension defined: an extension for defining thumbnails stored in formats other than 24-bit RGB.

APPO marker used for application-specific information

Additional APPO marker segments can be used to hold application-specific informati on which does not affect the decodability or displayability of the JFIF file. Application-specific APPO marker segments must appear after the JFIF APPO and any JFX X APPO segments. Decoders should skip any unrecognized application-specific APPO segments.

Application-specific APPO marker segments are identified by a zero terminated string which identifies the application (not "JFIF" or "JFXX"). This string should be an organization name or company trademark. Generic strings such as dog, cat, tree, etc. should not be used.

Conversion to and from RGB

Y, Cb, and Cr are converted from R, G, and B as defined in CCIR Recommendation 601 but are normalized so as to occupy the full 256 levels of a 8-bit binary encoding. More precisely:

```
Y = 256 * E' y
Cb = 256 * [ E' Cb ] + 128
Cr = 256 * [ E' Cr ] + 128
```

where the E'y, E'Cb and E'Cb are defined as in CCIR 601. Since values of E'y have a range of 0 to 1.0 and those for E'Cb and E'Cr have a range of -0.5 to +0.5, Y, Cb, and Cr must be clamped to 255 when they are maximum value.

RGB to YCbCr Conversion

```
YCbCr (256 levels) can be computed directly from 8-bit RGB as follows: 

Y = 0.299 R + 0.587 G + 0.114 B

Cb = -0.1687 R - 0.3313 G + 0.5 B + 128

Cr = 0.5 R - 0.4187 G - 0.0813 B + 128
```

NOTE - Not all image file formats store image samples in the order R0, G0, B0, \dots Rn, Gn, Bn. Be sure to verify the sample order before converting an RGB file to JFIF.

YCbCr to RGB Conversion

```
RGB can be computed directly from YCbCr (256 levels) as follows: 
 R = Y + 1.402 (Cr-128)
 G = Y - 0.34414 (Cb-128) - 0.71414 (Cr-128)
 B = Y + 1.772 (Cb-128)
```

Image Orientation

In JFIF files, the image orientation is always top-down. This means that the firs t image samples encoded in a JFIF file are located in the upper left hand corner of the image and encoding proceeds from left to right and top to bottom. Top-down orientation is used for both the full resolution image and the thumbnail image.

The process of converting an image file having bottom-up orientation to JFIF must include inverting the order of all image lines before JPEG encoding

Spatial Relationship of Components

Specification of the spatial positioning of pixel samples within components relative to the samples of other components is necessary for proper image post processing and accurate image presentation. In JFIF files, the position of the pixels in subsampled components are defined with respect to the highest resolution component. Since components must be sampled orthogonally (along rows and columns), the spatial position of the samples in a given subsampled component may be determined by specifying the horizontal and vertical offsets of the first sample, i.e. the sample in the upper left corner, with respect to the highest resolution component.

```
The horizontal and vertical offsets of the first sample in a subsampled component, Xoffseti[0,0] and Yoffseti[0,0], is defined to be Xoffseti[0,0] = (Nsamplesref / Nsamplesi) / 2 - 0.5 Yoffseti[0,0] = (Nlinesref / Nlinesi) / 2 - 0.5
```

where

Nsamplesref is the number of samples per line in the largest component, Nsamplesi is the number of samples per line in the ith component, Nlinesref is the number of lines in the largest component, Nlinesi is the number of lines in the ith component. Proper subsampling of components incorporates an anti-aliasing filter which reduce s the spectral bandwidth of the full resolution components. Subsampling can easily be accomplished using a symmetrical digital filter with an even number of taps (coefficients). A commonly used filter for 2:1 subsampling utilizes two taps (1/2, 1/2).

NOTE - This definition is compatible with industry standards such as Postcript Lev el 2 and QuickTime. This definition is not compatible with the conventions used by CCIR Recommendation 601-1 and other digital video formats. For these formats, p re-processing of the chrominance components is necessary prior to compression in o rder to ensure accurate reconstruction of the compressed image.

JPEG File Interchange Format Specification

The syntax of a JFIF file conforms to the syntax for interchange format defined in Annex B of ISO DIS 10918-1. In addition, a JFIF file uses APPO marker segments a nd constrains certain parameters in the frame header as defined below.

X'FF', SOI X'FF', APPO, length, identifier, version, units, Xdensity, Ydensity, Xthumbnail, Ythumbnail, (RGB)n

length (2 bytes) Total APPO field byte count, including the byte count value (2 bytes), but excluding the APPO marker itself

version (2 bytes) = X'0102'

The most significant byte is used for major revisions, the least significant byte for minor revisions. Version 1.02 is the current released revision.

units (1 byte) Units for the X and Y densities.

units = 1: X and Y are dots per inch
units = 2: X and Y are dots per cm

Xdensity (2 bytes) Horizontal pixel density Ydensity (2 bytes) Vertical pixel density

Xthumbnail (1 byte) Thumbnail horizontal pixel count

Ythumbnail (1 byte) Thumbnail vertical pixel count

(RGB)n (3n bytes) Packed (24-bit) RGB values for the thumbnail pixels, n = Xthumbnail \ast Ythumbnail

[Optional JFIF extension APPO marker segment(s) - see below]

0

```
X'FF', SOFn, length, frame parameters

Number of components Nf = 1 or 3

1st component C1 = 1 = Y component

2nd component C2 = 2 = Cb component

3rd component C3 = 3 = Cr component

o
o
o
X'FF', EOI
```

JFIF Extension APPO Marker Segment

Immediately following the JFIF APPO marker segment may be a JFIF extension APPO This JFIF extension APPO marker segment may only be present for JFIF vers ions 1.02 and above. The syntax of the JFIF extension APPO marker segment is: X'FF', APPO, length, identifier, extension code, extension data Total APPO field byte count, including the byte length (2 bytes) count value (2 bytes), but excluding the APPO marker itself = X' 4A', X' 46', X' 58', X' 58', X' 00'identifier (5 bytes) This zero terminated string ("JFXX") uniquely identifies this APPO marker. This string shall have zero parity (bit 7=0). extension code (1 byte) = Code which identifies the extension. In t his version, the following extensions are defined: = X' 10'Thumbnail coded using JPEG = X'11'Thumbnail stored using 1 byte/pixel = X'13'Thumbnail stored using 3 bytes/pixel

extension_data (variable) = The specification of the remainder of t

extension APPO marker segment varies with the extension. See below for a specification of extension_data for each extension.

JFIF Extension: Thumbnail coded using JPEG

he JFIF

This extension supports thumbnails compressed using JPEG. The compressed thumbnail immediately follows the extension_code (X'10') in the extension_data field and the length of the compressed data must be included in the JFIF extension APPO mark er length field.

The syntax of the extension_data field conforms to the syntax for interchange form at defined in Annex B of ISO DIS 10918-1. However, no "JFIF" or "JFXX" marker segments shall be present. As in the full resolution image of the JFIF file, the syntax of extension_data constrains parameters in the frame header as defined below:

```
X'FF', SOI
             0
    X'FF', SOFn, length, frame parameters
         Number of components
                                   Nf
                                          = 1 \text{ or } 3
         1st component
                            C1
                                  = 1 = Y component
                                  = 2 = Cb component
         2nd component
                            C2
         3rd component
                            С3
                                  = 3 = Cr component
             0
             0
X'FF', EOI
```

JFIF Extension: Thumbnail stored using one byte per pixel

This extension supports thumbnails stored using one byte per pixel and a color pal ette in the extension_data field. The syntax of extension_data is:

JFIF Extension: Thumbnail stored using three bytes per pixel

This extension supports thumbnails stored using three bytes per pixel in the exten sion_data field. The syntax of extension_data is:

Useful tips

o you can identify a JFIF file by looking for the following sequence: X'FF', SOI, X'FF', APPO, <2 bytes to be skipped>, "JFIF", X'00'.

o if you use APPO elsewhere, be sure not to have the strings "JFIF" or "JFXX" right after the APPO marker.

o if you do not want to include a thumbnail, just program Xthumbnail = Ythumbnail = 0.

o be sure to check the version number in the special APPO field. In general, if the major version number of the JFIF file matches that supported by the decoder, the file will be decodable.

o if you only want to specify a pixel aspect ratio, put 0 for the units field in t he special APPO field. Xdensity and Ydensity can then be programmed for the desire d aspect ratio.

Xdensity = 1, Ydensity = 1 will program a 1:1 aspect ratio. Xdensity and Ydensity should always be non-zero.