Motion JPEG Format

Draft 2
April 15, 1996

Background

On March 15 1996 a meeting was held on Apple's Cupertino campus to discuss how to reduce the confusingly large number of Motion JPEG formats in use today. This problem effects both users and software developers. It makes it almost impossible to work with data captured with boards from different vendors. There was agreement that a standard was necessary. This document attempts to summarize the discussion.

Who was there

Peter Hoddie, Apple
Brian Yamabe, Adobe
Mitchel Wienstock, Brooktree
Jean-Michel Berthoud, Truevision
Marshal Johnson, Truevision
Nick Moss, Radius
Jay Cuccarese, Avid
Rod Sheffield, Horizon
Michel Rynderman, Avid
Randy Ubillios, Macromedia
Michael Piper, Scitex
Raymond Picard, Truevision
Louis Shay, C-Cube
Scott Bronson, Diamond

Definition

Motion JPEG (M-JPEG) is a variant of the ISO JPEG specification for use in digital video. Its largest divergence from standard JPEG is that instead of compressing the entire image into one bit stream, it compresses each field separately, and returns the resulting bit streams consecutively.

Current Situation

There are two basic flavors of M-JPEG data in use today. These flavors exist because of differences in the hardware chip sets used by various board vendors. The first is more or less standard ISO JPEG, which is supported by chips from Zoran and C-Cubed. The second is close to ISO JPEG but cannot use the markers that ISO JPEG requires. This format is supported by LSI.

For purposes of this discussion, the Zoran/C-Cubed style will be referred to as M-JPEG Format A and the LSI style will be referred to as M-JPEG Format B.

High Level Decision

On the Macintosh, both of these formats are widely used. The formats are different enough that trying to convert between then in real time is not realistically possible. However, it is possible to convert (transcode) between them efficiently and with no quality loss. This is because the real differences are not in the content of the bit streams but in how that content is structured.

Because the differences in the formats reach into the lowest levels of the bit streams (i.e. they aren't limited to differences in the header, for example) there is no single bit stream definition which could accurately represent both formats. Therefore, it was decided to simply have two different M-JPEG formats, as described above.

User Implications

Today, every board vendor has a unique M-JPEG format. It is believed that those formats can be absorbed by the M-JPEG Formats A and B. As vendors revise their software to support both direct capture, compression, and display of either M-JPEG Format A or Format B it will be seamless for them to work in environments which support these formats. Further, it will be possible to easily transcode between Format A and B so that video editing applications can quickly, easily, and losslessly move between formats when "outputting" a movie.

Apple's Deliverables

Apple will provide a software implementation of M-JPEG Format A and B for both compression and decompression. Apple makes no claims that these are suitable for real time use. They will operate on both 68k and PowerPC based Macintoshes. At some point in the future they will also operate on Windows (and whatever other platforms the QuickTime common code base is moved to).

Apple will provide services in the Image Compression Manager for transcoding operations so that applications can easily convert from existing formats to the new standard formats.

Apple will provide two transcoders, to allow for conversion between M-JPEG Format A and B (from A to B and B to A).

Apple will modify the Image Compression Manager so that if a decompressor is not available for a given compressed image format, a transcoding operation will be used to convert the data to a supported format so it can be displayed. This will allow users, with the appropriate transcoder, to open any old style

M-JPEG stream on any QuickTime implementation that contains the previously mentioned enhancements.

Board Vendors Deliverables

Board vendors will modify their video digitizer components to capture into one or both of the new M-JPEG formats.

Board vendors will modify their compressor and decompressor components to support one or both of the new M-JPEG formats.

Board vendors will supply transcoders to convert from their custom formats into one or both of the new M-JPEG formats.

Application Developer Deliverables

Application developers will modify their applications to support transcoding. This only applies to applications which take in a movie as an input and generate another movie as an output. Supporting transcoding will provide users with a fast and lossless way to move between data in different formats. Supporting transcoding may require no human interface changes, since it is only a change to how a frame is generated (i.e. the user still chooses the desired compression format from the standard compression dialog).

Application developers will modify their applications to make use of the field dominance information stored in the image description when possible, instead of requiring the user to specify the field dominance.

M-JPEG Format A

M-JPEG Format A is in full compliance with the ISO JPEG specification. It contains two fields, each is a standard JPEG stream, one following the other. The first field in the stream is always the top field. The only enhancement made to the format is the addition of a new application marker. The decision was made to use the app 1 marker. This is because both JFIF (a standard interchange format for JPEG still images) and OpenDML (another M-JPEG specification) both use the app 0 marker. By using the app 1 marker, we believe it will be possible to create a single stream of compressed data that can be used both by QuickTime and by OpenDML.

The contents of the app 1 marker is described below. The app 1 marker appears in each field.

In keeping with the JPEG specification, all values appear in the stream with the most significant byte first (i.e. Motorola byte ordering). All fields below are 4 byte long integers. All sizes and offsets are calculated from the beginning of the compressed data stream.

unused	For historical (i.e. hardware specific reasons) this field's contents are undefined. If possible, it should be set to zero.
tag	This field is used to identify that the app 1 marker is actually the QuickTime M-JPEG specification. It should contain 'mjpg'.
field size	Contains the size of the image data for this field.
padded field size	The size of the image data including trailing pad bytes. Because some hardware may generate trailing pad bytes, or require image sizes to be multiples of a particular size.
offset to next field	Offset to next field of data. In the second field, this value should be set to zero. This value must be a multiple of 16 in size to ensure alignment for hardware decompressors. If the frame only contains a single field, then this field is always set to zero.
quantization table offset	Offset to the quantization table marker. If this field is zero, the image description should be checked for a default quantization table (see below).
huffman table offset	Offset to the huffman table marker. If this field is zero, the image description should be checked for a default huffman table (see below).
start of image offset	Offset to the start of image marker. This field must be non-zero, i.e. there must be start of image data.
start of scan offset	Offset to the start of scan marker. This field must be non-zero, i.e. there must be start of scan data.
start of data offset	Offset to the start of the data stream. Typically this immediately follows the start of scan data.

M-JPEG Format B

The M-JPEG Format B doesn't use the markers that the ISO JPEG specification defines. Therefore, there is no way to include the app 1 marker in the stream, as with M-JPEG Format A. Instead, before each stream of compressed data, a header is added. The header is the contents of the app 1 marker as defined above for Format A. The only differences in the header are owing to the lack

of markers. These are explained below. As with format A, the first field in the stream is always the top field.

unused	(same as Format A)
tag	(same as Format A)
field size	(same as Format A)
padded field size	(same as Format A)
offset to next field	(same as Format A)
quantization table offset	Instead of being the offset to the quantization table marker, this is an offset to the quantization table. This field must contain a multiple of 16.
huffman table offset	Instead of being the offset to the huffman table marker, this is an offset to the huffman table. This field must contain a multiple of 16.
start of image offset	Instead of being and offset to the start of image marker, this is an offset to the contents of the start of image data. This field must be non-zero, i.e. there must be start of image data.
start of scan offset	Instead of being and offset to the start of scan marker, this is an offset to the contents of the start of scan data. This field must be non-zero, i.e. there must be start of scan data.
start of data offset	This field must contain a multiple of 16 to ensure alignment for hardware decompressors

Format B also adds two trailing long words to the header to allow for future expansion (using QuickTime atoms, as defined in Inside Macintosh: QuickTime). Both should be set to zero. For hardware reasons, the Format B header must always be a multiple of 16 bytes in size. Any padding bytes added for this reason should be set to zero.

Finally, the compressed data itself is not “byte-stuffed”. Normally when an 0xff byte is written to a JPEG stream it followed by a 0x00 byte to indicate that it is not a marker. Because the Format B specification does not use markers, the addition of this extra 0x00 following each 0xff byte is not required (and in fact is an error).

The Image Description

QuickTime image descriptions can contain extensions for the storage of various format specific information. Both M-JPEG Format A and B allow default quantization and huffman tables to be stored in the image description.

These default tables are used whenever the compressed data stream doesn't contain the actual tables. For some hardware implementations, this may result in a modest savings in disk space. The extension type for the quantization table is 'mjqt'. The extension type for the huffman table is 'mjht'.

Also necessary is the now mythical extension to describe the presence of field based data and field dominance (i.e. which comes temporally first). This extension has the type of 'fiel'. It contains two bytes. The first byte is the field count. It should have a value of one or two. The second byte describes the field dominance. It has three possible values as shown by the following table. (Note: suggestions are welcome on how to describe each field - even and odd were already voted out.)

0	field dominance unknown
1	top field is temporally first
2	bottom field is temporally first

Note: If the M-JPEG data contains two fields, the Image Description must be present indicate this. If the field image description is not present, the software decoder may assume that the data is only field and the results will not be correct.