



**TITLE:** JPEG2000 requirements and profiles version 6.3

**SOURCE:** Requirements ad hoc group

**PROJECT:** JPEG2000

**STATUS:** Draft in progress

**REQUESTED ACTION:** Provide feed-back and comments to update

**DISTRIBUTION:** WG1 Arles Meeting, WG1 Web pages.

**Contact:**

ISO/IEC JTC 1/SC 29/WG 1 Convener - Dr. Daniel T. Lee  
Hewlett-Packard Company, 11000 Wolfe Road, MS42U0, Cupertino, California 95014, USA  
Tel: +1 408 447 4160, Fax: +1 408 447 2842, E-mail: Daniel\_Lee@hp.com

# 1 Introduction

This new work item is intended to create a new image coding system that for different types of still images (bi-level, gray-level, color, multi-component) with different characteristics (natural images, scientific, medical, remote sensing imagery, text, rendered graphics, etc.) allowing different imaging models (client/server, real-time transmission, image library archival, limited buffer and bandwidth resources, etc.) preferably within a unified system.

This coding system is intended to provide low bit-rate operation with rate-distortion and subjective image quality performance superior to existing standards, without sacrificing performance at other points in the rate-distortion spectrum. In addition, this system could include many modern features, listed in this document.

The standard will strive for openness and royalty-free licensing. It should be completed by the end of the millennium and offer state-of-the-art compression for at least ten years and beyond.

This standard will serve still image compression needs that are current not served by the current JPEG standards (IS 10918-1, IS 10918-2, IS 10918-3). (For example, very low bit-rate, progression for the WWW, medical imagery, pre-press, etc.) It is intended to compliment, not replace, the current JPEG standards. Indeed, this standard is expected to include an architectural context that will allow the previous standards to be used as desired on different tiles and/or components *within a single image*. This architecture will allow utilizing all of the current and future work of the JPEG committee to the best advantage.

## Vision for JPEG 2000

As digital imagery expands in quality, size, and application there is a greater need for image compression with great flexibility and efficient interchangeability. JPEG 2000 is a new type of image compression system. While offering state-of-the-art compression, JPEG 2000 also offers unprecedented access into the image while still in compressed form. Thus, images can be accessed, manipulated, edited, transmitted, and stored in a minimal information form. JPEG 2000 is a compressed format capable of being the foundation framework for image processing systems. Further, with a capable yet computationally modest decoder, interchangeability between applications is straight forward.

*Note to editor: Open standard concept and identification and protection should be included. This section will be updated after discussion in the requirements group and the panel session of seminar on digital imaging security and JPEG2000*

## 2 Purpose and justification

This standard is intended to advance standardized image coding systems to serve applications into the next millennium. It will provide a set of features vital to many high-end and emerging imaging applications by taking advantage of new modern technologies. Specifically, this new standard will address areas where current standards fail to produce the best quality or performance including the following. It will also provide capabilities to markets that currently do not use compression.

- **Low bit-rate compression performance:** Current standards, such as IS 10918-1 (JPEG), offer excellent rate-distortion performance in the mid and high bit-rates. However, at low bit-rates (e.g., below 0.25 bpp for highly detailed gray-level images) the distortion, especially when judged subjectively, becomes unacceptable.
- **Lossless and lossy compression:** There is no current standard that can provide superior lossless compression and lossy compression in a single codestream.

- **Large images:** Currently, the JPEG image compression algorithm does not allow for images greater than 64K by 64K without tiling.
- **Single decompression architecture:** The current JPEG standard has 44 modes, many of which are application specific and not used by the majority of the JPEG decoders. Greater interchange between applications can be achieved if a single common decompression architecture encompasses these features.
- **Transmission in noisy environments:** The current JPEG standard has provision for restart intervals, but image quality suffers dramatically when bit errors are encountered.
- **Computer generated imagery:** The current standard was optimized for natural imagery and does not perform well on computer generated imagery.
- **Compound documents:** Currently, JPEG is seldom used in the compression of compound documents because of its poor performance when applied to bi-level (text) imagery.

### 3 Objectives for next generation compression

JPEG 2000 will be defined by its features. It will fill a gap in the rate-distortion spectrum (low bit-rate) of current still image compression standards and provide a set of features vital to many high-end and emerging image applications. *Superior low bit-rate performance is primary.* It is desirable to include as many of the other following features as possible.

- **Superior low bit-rate performance:** This standard should offer performance superior to the current standards at low bit-rates (e.g., below 0.25 bpp for highly detailed gray-scale images). This significantly improved low bit-rate performance should be achieved *without sacrificing performance on the rest of the rate-distortion spectrum.* Examples of applications that need this feature include network image transmission and remote sensing. This is the highest priority feature.
- **Continuous-tone and bi-level compression:** It is desired to have a standard coding system that is capable of compressing both continuous-tone and bi-level images. If feasible, this standard should strive to achieve this with similar system resources. The system should compress and decompress images with various dynamic ranges (e.g., 1 bit to 16 bit) for each color component. Examples of applications that can use this feature include compound documents with images and text, medical images with annotation overlays, and graphic and computer generated images with binary and near to binary regions, alpha and transparency planes, and facsimile.
- **Lossless and lossy compression:** It is desired to provide lossless compression naturally in the course of progressive decoding (difference image encoding, or any other technique, which allows for the lossless reconstruction is valid). Examples of applications that can use this feature include medical images where loss is not always tolerated, image archival applications where the highest quality is vital for preservation but not necessary for display, network applications that supply devices with different capabilities and resources, and pre-press imagery.
- **Progressive transmission by pixel accuracy and resolution:** Progressive transmission that allows images to be reconstructed with increasing pixel accuracy or spatial resolution is essential for many applications. This feature allows the reconstruction of images with different resolutions and

pixel accuracy, as needed or desired, for different target devices. Examples of applications include the World Wide Web, image archival applications, printers, etc. The image architecture provides for the efficient delivery of image data in many applications such as client/server applications (World Wide Web).

- **Fixed-rate, fixed-size, limited workspace memory:** Fixed-rate (fixed local rate) means that the number of bits for a given number of consecutive pixels equals (or is less than) a certain value. This allows the decoder to run in real-time through channels with limited bandwidth. Examples are remote imaging, motion coding, etc. Fixed-size (fixed global rate) means that the total size of the codestream for a complete image equals a certain value. This allows hardware with a limited memory space to hold the complete codestream regardless of the image. Examples include scanners, printers, etc.
- **Random codestream access and processing:** Often there are parts of an image that are more important than others. This feature allows user defined Regions-Of-Interest (ROI) in the image to be randomly accessed and/or decompressed with less distortion than the rest of the image. Also, random codestream processing could allow operations such as rotation, translation, filtering, feature extraction, scaling, etc.
- **Robustness to bit-errors:** It is desirable to consider robustness to bit-errors while designing the codestream. One application where this is important is wireless communication channels. Portions of the codestream may be more important than others in determining decoded image quality. Proper design of the codestream can aid subsequent error correction systems in alleviating catastrophic decoding failures.
- **Open architecture:** It is desirable to allow open architecture to optimize the system for different image types and applications. This may be done either by the development of a highly flexible coding tool or adoption of a syntactic description language which should allow the dissemination and integration of new compression tools. It is desired to allow the user to select tools appropriate to their application and provide for future growth. With this feature, the decoder is only required to implement the core tool set and a parser that understands the codestream. If necessary, unknown tools are requested by the decoder and sent from the source.
- **Sequential build-up capability (real time coding):** The standard should be capable of compressing and decompressing images with a single sequential pass. This standard should also be capable of processing an image using component interleave order or non-interleaved order. During compression and decompression, the standard should use context limited to a reasonable number of lines. However, there is no requirement of optimal compression performance during sequential build-up operation.
- **Backwards compatibility with JPEG:** It is desirable to provide for backwards compatibility (or easy transcoding) with the current JPEG standards.
- **Content-based description:** Finding an image in a large database of images is an important problem in image processing. For example, a doctor could request only images from a set that are recognized to have a certain type of tumor. This could have major applicability to the medical, law

enforcement and environmental communities, and for image archival applications. Regardless of the techniques used, JPEG 2000 should strive to provide the opportunity for solutions to this problem.

- **Protective image security:** Protection of a digital image can be achieved by means of one or more of methods such as: watermarking, labeling, stamping, fingerprinting, encryption, scrambling, etc. Watermarking and fingerprinting are invisible marks set inside the image content to pass a protection message to the user. Labeling is already implemented in SPIFF and must be easy to transfer back and forth to JPEG 2000 image file. Stamping is a mark set on top of a displayed image that can only be removed by a specific process. Encryption, and scrambling can be applied on the whole image file or limited to part of it (header, directory, image data) to avoid unauthorized use of the image.

Note: The issue of whether JPEG2000 should cover specific solutions to encryption, watermarking, authentication algorithms should be revisited

- **Compatibility with ITU-T recommendations for image exchange:** JPEG 2000 is a joint activity between ISO/IEC JTC1 and ITU-T. Facsimile is one of the targeted applications of JPEG 2000, and the ITU-T Recommendations for image exchange will be used for the interchange and delivery of JPEG 2000 encoded images. Using JPEG 2000 for image delivery in conformance with the ITU-T Recommendations requires the ability to insert a JPEG 2000 codestream in an ITU-specified data stream or file format. T.42. In particular, JPEG 2000 data streams must be usable in Group 3 fax (ITU-T Rec. T.4), in the TIFF-FX files used for Store & Forward Internet Fax (ITU-T Rec. T.37) and in Mixed Raster Content (ITU-T Rec. T.44). Also, JPEG 2000 encoded images may be able to conform to the ITU-T requirements for grayscale and color image encoding, as defined in ITU-T Rec.
- **Interface with MPEG-4:** The ongoing standardization process for the “Coding of moving pictures and audio” (ISO/IEC JTC1/SC29/WG11), MPEG-4, is developing a content based coding scheme in which coding tools are chosen from a repertoire so as to address in an optimal way a wide range of functionalities. It is desirable that the coding tool (or tools) developed for the compression of still images in JPEG 2000 are provided with an appropriate interface allowing the interchange and the integration of such tools into the framework of a syntax oriented coding scheme such as MPEG-4. In particular IPR information should be maintained in both insertion and extraction of still pictures to/from moving images.
- **Side channel spatial information (transparency):** Side channel spatial information, such as alpha planes and transparency planes, are useful for transmitting information for processing the image for display, print, or editing, etc. An example of this is the transparency plane used in World Wide Web applications.
- **Object Based Functionality -**  
**Object based composition:** The current standard was designed to encode/decode rectangular framed images, which made it difficult to represent specific regions within the image. Multiple objects with arbitrary shape (or transparency) information will enable users to composite different still images with as much flexibility.

**Object based information embedding:** The side information (e.g., description of object) attached to each object will enable the new standard to provide more efficient way to digital image indexing and retrieval

It is important to note that, while this standard addresses the needs of this wide variety of images and applications, it desirable that the decoder be able to interpret codestreams with minimal complexity. There must be a cost/benefit assessment of each function that adds complexity to the decoder.

## 4 Markets and applications

JPEG 2000's desired capabilities shall serve markets and applications such as:

- **Internet**
- **Facsimile (including color and internet fax applications)**

The application "Facsimile" contains not only conventional Black-and-white G3 Facsimile, but also Color-Facsimile, Internet-Facsimile, and ones to be recommended in the future, which JPEG2000 might be applied to.

- **Printing**
- **Scanning (Consumer, Pre-Press)**
- **Digital Photography**
- **Remote Sensing**
- **Mobile**
- **Medical**
- **Digital Library**
- **E-Commerce**

## 5 Application Requirements

This section contains requirements of specific applications that were envisioned as major users of the JPEG2000 standard. The goal of this section is to identify these requirements and to ensure that appropriate tools and technologies are adopted in JPEG2000 standard to fulfill them.

*Editor's note: Scenarios received in Seoul are to be added. The structure under which this should be done is to be decided in Vancouver meeting.*

*Editor's note: We need to define the meaning of requirement as an absolute minimum*

### 5.1 Image Type

#### **Definition**

Width, height, component bit depth, number of components, and the span of each component (sub-sampling) determine the image type. Also the source of the image should be considered.

#### **Aggregate requirements**

The image type should not be limited by the algorithm. The codestream syntax should be capable of the following:

Image width and height	1 to $(2^{32} - 1)$
Component depth	1 to 32 bits
Number of components	1 to 255 (or more)
Dissimilar component depths (each component can be a different depth)	
Dissimilar component spans (each component can have a different coverage)	

#### **Changes to the application profiles**

**Internet:** The JPEG2000 should be able to process compound images, with sizes from 32 x 32 up to at least 4K x 4K pixels with 1, 3 (Y, RGB, YUV,...) or 4 components including alpha channel and from 1 to 8 bits/component precision.

**Facsimile:** JPEG2000 should be able to process compound images, with sizes, number of planes and bit depth as specified by the color Fax standard

**Printing:** The JPEG2000 should be able to process compound images, with typical sizes of 4800 by 6600 pixels (600ppi, 8in by 11in image) with 1, 3, and 4 components, and 8 bits/component precision.

Scanning: The JPEG2000 should be able to process compound images, with typical sizes of 10K x 10K up to at least 20K x 20K pixels with 1, 3 and 4 components, and up to 16 bits/component precision.

Digital Photography: The JPEG2000 should be able to efficiently process natural images, with sizes of at least up to 4K x 4K pixels with 1, 3 components (with spatially correlated components), with a minimum of 8 bits/component and a maximum of 16 bits/component precision. In case of a three component image, JPEG2000 should be able to handle both the full image case, where all 3 components are present for each pixel. Remote Sensing: The JPEG2000 should be able to process infra-red, electro-optical, multi-spectral, hyper-spectral, and SAR images, with virtually unlimited vertical definition and fixed horizontal definition depending on the line scan sensor upto 24000 pixels with 1 up to 500 components, and 8 up to 20 bits/component precision. It should also be able to process one component data with complex pixels (i.e phase information). Fixed transmission bitrate is a special requirement for satellite links (e.g. 50-100 Mb/s)

Mobile: The JPEG2000 should be able to process compound images, with sizes from 32 x 32 up to at least 4K x 4K pixels with 1 or 3 components (Luminance, RGB, ...) and 1 to 8 bits/component.

Medical: The JPEG2000 should be able to process natural images, with sizes from 32 x 32 to at least 10K x 10K pixels with 1 and 3 (Luminance, RGB, ...) or 4 components (plus alpha) and up to 16 bits/component.

Digital library: Same as the Internet.

E-commerce: Same as the Internet.

## 5.2 Uncompressed (no compression)

### ***Definition***

The image is stored in the bitstream without compression. This allows applications to take advantage of the JPEG2000 bitstream syntax and its associated features without compressing the image data.

### ***Aggregate requirements***

JPEG2000 should provide a mode in which the image is archived as an uncompressed form, leaving other syntax elements (embedding information, etc.) intact.

## 5.3 Lossless Compression

### ***Definition***

The reconstructed image is identical, bit for bit, to the original image. A typical compression ratio for this type of compression is 2:1 using JPEG LS.

### ***Aggregate requirements***

Provide performance at least as good as JPEG LS. Performance includes speed, complexity, memory requirements, etc.

## 5.4 Visually Lossless Compression

### ***Definition***

The reconstructed image may differ numerically from the original image, but any such differences are not perceptible under normal viewing conditions. Current baseline JPEG can generally attain visually lossless performance at rates between 1 and 2 bpp for grayscale images and slightly lower rates, in bpp per band, for color images.

### ***Aggregate requirements***

Provide 30% improvement over JPEG baseline.

## 5.5 Visually Lossy Compression

### ***Definition***

The reconstructed image contains perceptible differences from the original image under normal viewing conditions. Current baseline JPEG generally exhibits visually lossy performance at rates under 1 bpp for grayscale images and lower rates, in bpp per band, for color images.

### ***Aggregate requirements***

Provide 30% improvement over JPEG baseline.

## 5.6 Progressive Spatial Resolution

### ***Definition***

Ability to extract lower resolution images from a codestream without redundant decoding.

### ***Aggregate requirements***

At least 9 resolution levels. Should be combinable with progressive quality resolution (see next section), whenever this is feasible with the size of the image.

### ***Changes to the application profiles***

Facsimile: Should also require progressive spatial resolution. This would enable facsimile systems where the resolution capability of the destination facsimile is determined and the appropriate resolution sent. This is useful for Internet facsimile, facsimile from a database and “broadcast” facsimile. It would also allow facsimile standards to incorporate generational improvements in performance without altering the standard.

Printing: Should also require progressive spatial resolution. This would enable images to be sent to printers of different resolution. Also quick resolution scaling of images could be performed as called for by the user, page-description language, or other controller.

Scanning: Should also require progressive spatial resolution. This would support interchange with the Internet, Facsimile, and Printer applications.

## **5.7 Progressive Quality Resolution**

### ***Definition***

Ability to extract lower bit-rate images from a codestream without redundant decoding or sacrifice of image quality (at that bit-rate).

### ***Aggregate requirements***

At least 8 levels of quality. However, continuous progression has even more flexibility and advantage. Progression for many applications requires lossless to very lossy (low bit-rate). In some applications, moderate compression (near lossless to just noticeable artifacts) is the minimum acceptable. Should be combinable with progressive spatial resolution (see previous section).

### ***Changes to the application profiles***

Internet: Progression from lossless to visually lossy.

Facsimile: Should also require progressive quality resolution. This would enable facsimile systems where the buffering, pixel, or component depth capability of the destination facsimile is determined and the appropriate bits sent. Also, this would allow fixed-time or interrupted transmission of facsimile images. It would also allow facsimile standards to incorporate generational improvements in performance without altering the standard. Lossless to very lossy progression is required.

Printing: Should also require progressive quality resolution. This would enable images to be sent to printers of different sizes, allowing the management of buffers and display lists. Lossless to very lossy progression is required.

Scanning: Should also require progressive quality resolution. This would support interchange with the Internet, Facsimile, and Printer applications. Lossless to very lossy progression is required.

Digital photography: Moderate lossy to very lossy. Lossless is optional.

Remote sensing: Lossless to moderate lossy.

Mobile: Moderate lossy to very lossy.

Medical: Lossless to moderate lossy.

Digital library: Same as the Internet.

E-commerce: Same as the Internet.

## 5.8 Security

### ***Definition***

Three purposes: 1) protect access to the image, 2) identify the image, source or owner in a secure way that cannot be removed or modified by unauthorized parties, 3) indication of the integrity.

### ***Aggregate requirements***

To be visually transparent, yet perform the purposes in electronic form and is preserved after image compression and other image processing operations.

### ***Changes to the application profiles***

Internet: Read and write.

Facsimile: Read and write.

Printing: Read only.

Scanning: Write only.

Digital photography: Read and write.

Medical: Read and write.

Digital library: Read and write.

E-commerce: Read and write.

## 5.9 Error resilience

### ***Definition***

To be “robust” (allow complete or acceptable partial decoding) in the presence of errors in the codestream such as random errors, burst errors, and packet or byte loss or insertion errors.

### ***Aggregate requirements***

Should have capabilities that make the codestream robust to errors.

- Medical:. Medical imagery does not require more stringent error control within JPEG 2000 than other applications.

In almost (if not all) medical imaging applications, integrity is maintained at a lower level through the use of reliable, robust, error-correcting media or network communication channels, and to provide additional error robustness within JPEG 2000 at the expense of additional complexity or reduced performance would not be helpful. Neither is it a requirement to “partially” recover some part of an image.

## 5.10 Complexity Scalability

### ***Definition***

The ability to be able to decode the same bitstream with less complex decoders and obtaining results as close as possible to the one obtained while decoding the same bitstream with a more complex decoder.

### ***Aggregate requirements***

JPEG2000 should be scalable in complexity, so that depending on the applications, different levels of complexity can be implemented.

## 5.11 Strip Processing

### ***Definition***

The ability to compress and decompress images with a single sequential pass.

Encoder: The ability to generate bitstreams with appropriate number of lines in a single sequential path without holding a whole image in memory of the encoder.

Decoder: The ability to reconstruct image from bitstreams corresponding to the appropriate number of lines in a single sequential pass without holding the whole bitstreams in memory of the decoder.

### ***Aggregate requirements***

JPEG2000 should support strip processing with 1 or more lines per strip.

## 5.12 Sensor-Specific Compression Flexibility

### ***Definition***

The ability to specify compression algorithm components and/or parameters, such as particular transforms, transform kernels, quantizers or coding strategies, based upon the characteristics of the imaging device and/or data.

### ***Aggregate requirements***

JPEG2000 should provide the capability to specify compression algorithm components and/or parameters, such as particular transforms, transform kernels, quantizers or coding strategies, based upon the characteristics of the imaging device and/or data.

## 5.13 Information embedding

### ***Definition***

Efficient embedding of non-image information such as text, voice annotation, web links, and other types of meta-data information into compressed images. Information can be embedded to objects of arbitrary shape or (predefined) rectangular shape. The latter will be referred to as rectangular object information embedding.

### ***Aggregate requirements***

JPEG2000 should allow an efficient embedding of additional object-based information and other types of meta-data information into compressed images. For example, text, voice annotation, web links, digitalization tool used; color calibration made; look-up tables to be used; backtrack of all the processing done on the image after it has been digitized and other types of meta-data information into compressed images.

## **5.14 Repetitive Encoding/Decoding**

### ***Definition***

The ability to decode and re-encode iteratively without adding distortion. (Idempotency)

### ***Aggregate requirements***

JPEG2000 should be capable of iteratively decoding and re-encoding without adding distortion

## **5.15 Object-Based Functionality**

### ***Definition***

The ability to access and manipulate or decode individual objects constituting the image independently.

### ***Aggregate requirements***

To be done

## **5.16 MPEG4 VTC Compatibility**

### ***Definition***

### ***Ability to decode bitstreams of MPEG4 still texture object*** ***Aggregate requirements***

JPEG2000 should be able to decode bitstreams of MPEG4 still texture object directly or after simple transcoding.

### ***Scenario***

Some applications of JPEG2000 and MPEG4 are overlapped each other, such as Internet, mobile, and digital photography. There should be demands for simple decoder, which can decode both bitstreams of JPEG2000 and bitstreams of MPEG4 still texture object.

## **5.17 Backward compatibility/easy transcoding on the client side**

### ***Definition***

Ability to decode current JPEG baseline images This does not include a bitstream backward compatibility.

### ***Aggregate requirements***

## **5.18 Dynamic ROI Decoding**

### ***Definition***

To be able to decode specific regions of the image on the fly while decoding the bitstream, without necessarily defining the regions on the encoder side.

### ***Aggregate requirements***

The standard should allow the decoder to specify which part of the receiving image (during the stage that the user is receiving an image in progressive transmission schemes) will be decoded at better quality ( up to lossless, provided that the image was stored in a lossless mode). The user at the decoder side should be able to decode only specific ROI's lossless and accept a graceful degradation of the image outside the ROI. Notice that the ROI selected need not be known to the encoder during the encoding.

## **5.19 Fast/Random Data Access**

### ***Definition***

Efficient random access to portions of the data and to be able to decode them without necessarily decoding the rest of the bitstream.

### ***Aggregate requirements***

The JPEG2000 decoder should allow a quick and efficient random access to portions of the compressed image without necessarily decoding the whole image. This should not result in excessive overhead. The speed of the random access should be close to proportional to the size of the image decoded.

## **5.20 Implementation Complexity**

### ***Definition***

Complexity is related to the handling of three different processes, encoding, transfer and decoding. JPEG2000 has to support the concept of scalable complexity, i.e. terminals or other devices based on different hardware platforms must be able to generate or decode bitstreams according to their computational capabilities. With regard to transfer, in case of server applications the parsing of JPEG2000 bitstreams must be possible for bitrate reduction purposes without the need for transcoding. For battery based applications such as mobile or digital camera, computational complexity and memory costs such as the amount of memory and the memory bandwidth have to be in application typical ranges.

### ***Aggregate requirements***

Algorithms should also be considered based on memory usage and accesses.

## **5.21 Static ROI coding**

### ***Definition***

To decode a given region of interest in the bitstream with higher quality decided when the encoding was performed. This is in contrast to the Dynamic version in which the ROI can be changed by the decoder.

### ***Aggregate requirements***

The JPEG2000 decoder should allow the user at the encoder side to specify which part (of arbitrary shape) of the image (Region of Interest - ROI) should be coded at better quality, up to lossless, and which lossy. It should allow the user to trade off compression performance with image quality by providing the ability to code an ROI to a higher degree of quality (lossless or lossy).

## **5.22 Geometric manipulation of compressed images**

### ***Definition***

Efficient geometric manipulation of JPEG2000 compressed images, limited to rotations of 90, 180, 270 degrees and horizontal and vertical flipping, and any combinations.

### ***Aggregate requirements***

Any number of consecutive geometric manipulations should not result in a significant change in the size of the compressed file, nor in any degradation in image quality. The JPEG2000 standard should allow geometric manipulations to support low complexity and low memory applications..

## **5.23 Compatibility with ITU-T Recommendations for image exchange**

### ***Definition***

The ability to insert a JPEG 2000 codestream in an ITU-T specified data stream or file format

### ***Aggregate requirements***

The grayscale and color image data in a JPEG 2000 codestream must be able to conform to the color encodings specified in ITU-T Rec. T.42:

- CIELAB color space, with encoding parameters
- optional gamut range and illumination

It must be possible to use a JPEG 2000 codestream in Group 3 facsimile transmission; the JPEG 2000 facsimile mode would be defined in an annex of ITU-T Rec. T.4. This requires:

- 1 or 3 components per pixel
- 8 bits per component, 12 bits optional

- chroma subsampling and chroma positioning
- spatial resolution: 100, 200, 300, 400 dpi
- streaming, so that it is not required to buffer the entire image before encoding, transmission and decoding can start
- coordination of marker codes in the JPEG 2000 codestream and the ITU-T data stream
- coordination between Error Correction Mode (ECM) in Group 3 fax and optional error resilience in JPEG 2000

It must be possible to insert a JPEG 2000 codestream in a TIFF-FX file for use by Store & Forward Internet Fax, as defined in ITU-T Rec. T.37 “Procedures for the Transfer of Facsimile Data via Store and Forward on the Internet”. A JPEG 2000 codestream (encoded image) could be inserted in a separate TIFF-FX profile or as the Foreground and/or Background layer images of a TIFF-FX Profile M file. This requires coordinating JPEG 2000 codestream parameters with TIFF-FX field values.

It must be possible to insert a JPEG 2000 codestream as an odd-numbered layer image data in a T.44 data stream, as specified in ITU-T Rec. T.44. Additionally, when a T.44 data stream is used for Group 3 facsimile, as specified in ITU-T Rec. T.4, Annex H, it must be possible to split a page into 256-line stripes, so that it is necessary to be able to limit the JPEG 2000 codestream to independent strips with a maximum of 256 lines of image data. (Page-sized stripes are an option.)

## 5.24 Interactive streaming

### ***Definition***

Interactive Streaming herein defined and used in the JPEG-2000 context refers to the capability and mechanism that allow a JPEG-2000 image to be interactively and progressively accessed in part or in whole over a network.

In a generalized scenario, a request for access to parts of a JPEG-2000 bitstream may be issued from one end of the network and served across the network. The user may interactively change the access request during the transaction, e.g., ask for a cancellation of the previous request and substitute with a new access request. The requested access may be for a random portion (including the whole image) at a specified resolution/quality, a coded region of interest (ROI), and/or a component of the image. With each request, the JPEG 2000 bitstream should be progressively delivered to the user so that intermediate version of the image can be rendered.

### ***Aggregate requirements***

The following specific requirements have been identified as necessary to enable interactive streaming

#### **A. Low latency**

There are several aspects of latencies. One aspect is to minimize the delay between the time when a user makes a request and the time when the initial content is delivered. The other aspect is to minimize the total time span when the request is fulfilled and all the content related to the request has been delivered. The delivery should be performed in a fashion such that the user can get the best experience with the available delivered content. Low latency ensures good interactivity and user experience.

#### **B. Content negotiation: optimal access/delivery**

In the interactive streaming environment, it is essential to fulfill a user's request as soon as possible by delivering only the content appropriate to the client. The client should negotiate with the server about its capability so that useless content is not delivered.

#### **C. Client/server complexity trade-offs.**

There are tradeoffs between thin-client/fat-server, and thin-server/fat-client implementations. The thin-client application may be common in a mobile device, where the client device should be as simple as possible. The fat-client application is more common in a desktop environment, where the client is very powerful and a better responsive system can be established when the client did more works. A desired JPEG-2000 standard shall be flexible enough to allow implementation trade-offs.

Note: These requirements are should be added to the table in the next section and examined for each identified application profile.

This section contains a table that indicates the mandatory and optional requirements associated with the individual application profiles.

Note: a core profile should be defined. In addition, some of the following profiles should be merged in order to define other more generic profiles (main, simple, etc.) profiles for clusters of applications.

Note: a space application profile is intended to be added to these profiles. It has been identified that space application profile is very closely related to remote sensing profile and could probably match its major requirements. The main characteristic requirement of space applications seems to be low encoder complexity.

Application Profile		Overall System													Decoder Specific									
		5.1 Image Type	5.2 Uncompressed	5.3 Lossless Compression	5.4 Visually Lossless Compression	5.5 Visually Lossy Compression	5.6 Progressive Spatial	5.7 Progressive Quality	5.8 Security	5.9 Error Resilience	5.10 Complexity Scalability	5.11 Strip Processing	5.12 Compression Flexibility	5.13 Information Embedding	5.14 Repetitive Encoding/Decoding	5.15 Object-Based Functionality	5.16 MPEG4 VTC Compatibility	5.17 Backward Compatibility Client Side Ease of Transcoding	5.18 Dynamic ROI	5.19 Fast/Random Data Access Implementation	5.20 Complexity	5.22 Geometric Manipulation	5.21 Static ROI	5.23 ITU-T compatibility
6.1	Internet	M(1,3) O(4+)			M	M	M	M	O	O				M(rec. obj)	M	O	M(Baseline) O(non-baseline)	O	M				O	M(ifax)
6.2	Facsimile	M		O	O		O	O		O	M							O		O	M	M		M
6.3	Printing	M			M	O				O		M						O		O	M	O		
6.4	Scanning (Consumer, pre-press)	M	O	M	M				O											O				
6.5	Digital Photography	M		O	M	O	O	O	O	O			M(rec.obj.)	O	O	O		O			O	O		
6.6	Remote Sensing	M(1,3) O(4+)	O	M	M	M	M	O	O	O	O	M	M	M(LUTs)		O		O		M	M		O	M
6.7	Mobile	M(1,3) O(4+)		O	M	M	O	M	O	M	O					O	O	O		O	M	M		O
6.8	Medical	M(1,3) O(4+)		M	M	M	M	M	M	O		M	M	M(rec.obj)	O	O		M(Baseline) O(non-baseline)	O	M				M
6.9	Digital Library	M(1,3) O(4+)		O	M	M	M	M	O	O				M(rec.obj)		M		M(Baseline) O(non-baseline)	O	M			O	O
6.10	E-Commerce	M(1,3) O(4+)		O	M	M	M	M	M	O				M(rec.obj)		M		M(Baseline) O(non-baseline)	O	M			O	

~~Legend:~~ M - Mandatory. O - Optional

Overall System

	Application Profile	Streaming low latency 5.24 A	Streaming Content negotiation 5.24 B	Streaming Complexity trade- 5.24 C
6.1	Internet	M	M	M
6.2	Facsimile			
6.3	Printing			
6.4	Scanning (Consumer, pre-press)			
6.5	Digital Photography	O		
6.6	Remote Sensing	M	M	M
6.7	Mobile	M	M	M
6.8	Medical	M	M	M
6.9	Digital Library	M	M	O
6.10	E-Commerce	M	M	M

**Legend:** M - Mandatory, O - Optional

## **Appendix A - Terms of references**

This appendix contains definitions of different terms used in this document.

### **Errors**

Understood as residual errors in the bitstream input to the decoder after error correction at the system layer.

### **Robust**

Understood as the decoder's ability to continue to decode the image in the presence of errors with a graceful degradation of image quality avoiding a catastrophic breakdown (failure) as the channel conditions deteriorate.

### **Error types**

In environments where JPEG2000 is likely to be used, the transmission noise may be from one of the following 3 major classes:

1. Random bit errors (Binary symmetric channel)
2. Burst errors (Mobile and wireless fading channels)
3. Packet loss (byte deletion/insertion in IP/ATM communication)

The transmission will in some way provide error correction/detection at system layer. When this fails, the JPEG2000 decoder will have to face a bitstream with errors. These again will belong to one of the classes mentioned above.

The following is an attempt to better clarify the term error used in various applications considered in this document

**Internet:** Packet loss in Internet transmission not having guaranteed packet delivery

**Hardcopy Facsimile:** random errors

**Digital photography:** Considering hand held devices with transmission equipment this will be very similar to mobile. Considering storage on a device in the camera, random errors at a very small probability.

**Remote sensing:** Random errors, burst errors (Considering satellite systems and radio based surveillance equipment)

**Mobile:** Random errors, burst errors, packet loss/insertion (IP)

**Medical:** None, need the ability for error detection

### **Region of Interest (ROI)**

A *Region Of Interest* (ROI) is defined as an object in the bitstream that the user might access and manipulate (cut, paste, etc.). It can be made of Y,U, and V component plus shape information. The encoder might transmit together with each ROI composition information to indicate where each ROI shall be displayed. At the decoder the user can change the composition of the scene displayed by interacting on the composition information.

The ROI's can have an arbitrary shape and can be of any type (synthetic texture, text, graphics, images,...). The ROI's can be known by construction of the images (for example synthetic images) or can be defined by semi-automatic segmentation. The background is also defined as a ROI. ROI's can be inside one other ROI. Each ROI might have some data, i.e. semantic information linked to it.

## **Scalability**

Realizing that many applications require images to be simultaneously available for decoding at a variety of resolutions or qualities this architecture shall support scalability. In general, scalable coding of still images means the ability to achieve coding of more than one resolution and/or quality simultaneously. Scalable image coding involves generating a coded representation (bitstream) in a manner which facilitates the derivation of images of more than one resolution and/or quality by scalable decoding. Bitstream Scalability is the property of a bitstream that allows decoding of appropriate subsets of a bitstream to generate complete pictures of resolution and/or quality commensurate with the proportion of the bitstream decoded. If a bitstream is truly scalable, decoders of different complexities, from low performance decoders to high performance decoders can coexist, and while low performance decoders may decode only small portions of the bitstream producing basic quality, high performance decoders may decode much more and produce significantly higher quality. The most important types of scalability are *SNR scalability* and *spatial scalability*.

*SNR scalability* is intended for use in systems with the primary common feature that a minimum of two layers of image quality are necessary. *SNR scalability* involves generating at least two image layers of same spatial resolution but different qualities from a single image source. The lower layer is coded by itself to provide the basic image quality and the enhancement layers are coded to enhance the lower layer. The enhancement layer when added back to the lower layer regenerates a higher quality reproduction of the input image.

*Spatial scalability* is intended for use in systems with the primary common feature that a minimum of two layers of spatial resolution are necessary. *Spatial scalability* involves generating at least two spatial resolution layers from a single source such that the lower layer is coded by itself to provide the basic spatial resolution and the enhancement layer employs the spatially interpolated lower layer and carries the full spatial resolution of the input image source.

*An additional advantage of spatial and SNR scalability types is their ability to provide resilience to transmission errors as the more important data of the lower layer can be sent over channel with better error performance, while the less critical enhancement layer data can be sent over a channel with poor error performance.*

Both types of scalability are very important for internet and database access applications, and bandwidth scaling for robust delivery. The *SNR* and *Spatial* scalability types include the progressive and hierarchical coding modes already defined in the current JPEG but they are more general. In addition, the following types of scalability are useful:

*Complexity scalability* in the decoder allows a given texture or image bitstream to be decoded by decoders of different levels of complexity. The reconstruction quality, in general is related to the complexity of the decoder used. This may involve that less powerful decoders decode only a part of the bitstream. This relates to the spatial scalability, where a decoder might decode only the low resolution image.

*Quality scalability* allows a bitstream to be parsed into a number of bitstream layers of different bitrate such that the combination of a subset of the layers can still be decoded into a meaningful signal. The bitstream parsing can occur either during transmission or in the decoder. An application of this could be in SNR scalability with two layers, where the decoder selects some ROI's from the base layer and some ROI's from the enhancement layer and composites an image which has a mixed quality.

Notice that scalability applies both to texture as well as in shape.

## **Composition information**

The composition information carried in the composition layer carries information related to the composition rule for each ROI (i.e. how the regions will be displayed, if they will overlay, etc.). One important application is cases where we have text on top of a background image. In that case it should be possible to have the text described as a separate region. The region description header of that regions should specify the text description language (e.g. postscript) and the composition rule (e.g. superposition).

A scene/image can be composed from objects of various types. Composition information is transmitted and at the received site, the user can interact with the compositor and manipulate the objects.

## **User Interaction**

User interaction can have two forms: client-side interaction and server-side interaction.

Client-side interaction involves content manipulation which is handled locally at the end-user's terminal, and can take the following forms:

1. Modification of an attribute of a ROI. This includes changing the position of a ROI, making it visible or invisible, changing the font size of a synthetic text node, etc.
2. Hyperlinking. This involves switching to a new content source after the user selects (e.g., with a mouse click) a particular ROI (or portion of it).

The first form can be easily implemented by applications using JPEG 2000 by providing internal (implementation dependent) mechanisms that link user events (e.g., mouse clicks or keyboard commands) to scene update commands. These scene update commands can be processed by the terminal in exactly the same way as if they originated from the original content source. As a result, this type of interaction does not require standardization.

Hyperlinking requires the presence of location information and the necessary support to access it. This information can be carried by the semantic information data, but it is up to the application to implement the necessary functionality (i.e., switch to the new source when the anchor is selected).

Server-side interaction involves content manipulation that occurs at the transmitting end, initiated by the receiver. For example, the receivers terminal might send information to the transmitter on which ROI's to selectively refine, or to whether he would like to see the image or some ROI's/tiles in higher resolution, etc.

## **Architecture**

A description of a unifying or coherent form or structure. A compression architecture describes the structure to support compression and expansion. An architecture will define a set of tools and their possible associations to allow step-by-step procedures necessary to complete compression or expansion

Backward Compatibility - DecoderA JPEG2000 decoder is backward compatible if it is able to interpret, and expand the compressed imagery information (codestream) produced by a JPEG compliant encoder.

## ***Backward Compatibility - Encoder***

A JPEG2000 encoder is backward compatible if it produces a codestream which can be parsed, translated or transcoded by a JPEG compliant decoder and interpreted to the extent that catastrophic decoder failure is avoided. The decoder may skip the expansion process, or expand an image and either flag or not flag its lack of meaning.

## ***Bitstream***

A bitstream is the actual sequence of binary digits resulting from the coding of a sequence of symbols (compressed data stream). As such it does not include bits containing information regarding the coder type, its parameters, or other (overhead) information required to decode the sequence.

## ***Codestream***

A codestream is a collection of one or more bitstreams and associated (overhead) information required for their decoding and expansion into image data. Such overhead information is restricted to that required for expansion and may include, but is not limited to, markers indicating locations of particular bitstreams, segments indicating transform, quantization and coding types, etc.

## ***File Format***

A file format consists of a codestream and additional support data and information not explicitly required for the expansion of image data. Examples of such support data are text fields providing titling, security and historical information, markers to support placement of multiple codestreams within a given data file, and markers to support exchange between platforms or conversion to other file formats.

## ***Compound image***

A compound image is a composed of parts generated synthetically and parts coming from natural images. A compound image may degenerate to a fully natural or a fully synthetic image.

## ***Part I of the standard***

A good set of requirements (features) as shown in appendix B which fulfills a good proportion of applications (20% of JPEG2000 technologies in all parts of the standard, covering 80% of its applications), while keeping a reasonable complexity in implementation

## ***Feature:***

Once a requirement is fulfilled by a technology in the standard it is considered as the feature of JPEG2000.

*Editor's note: error resilience definitions to be added*

## Appendix B – Requirements for part I

Requirements identified to be included in Part I of the standard.

Requirement Number	Requirement Description	Comments
1	<p>Image Type</p> <p>Types:</p> <p>Natural <input checked="" type="checkbox"/></p> <p>Compound <input checked="" type="checkbox"/></p> <p>Infrared Sensor <input type="checkbox"/></p> <p>Electro-Optical Sensor <input type="checkbox"/></p> <p>Synthetic Aperture Radar <input type="checkbox"/></p> <p>Hyper-spectral <input type="checkbox"/></p> <p>Other: _____ <input type="checkbox"/></p> <p>Pixels count per dimension (Width, Height):</p> <p>1 to 255 (1 to <math>2^8-1 = 1</math> byte) <input type="checkbox"/></p> <p>1 to 65535 (1 to <math>2^{16}-1 = 2</math> bytes) <input type="checkbox"/></p> <p>1 to 16,777,215 (1 to <math>2^{24}-1 = 3</math> bytes) <input type="checkbox"/></p> <p>1 to 4,294,967,295 (1 to <math>2^{32}-1 = 4</math> bytes) <input checked="" type="checkbox"/> Addressing range, image size limits has to be defined</p> <p>Other: _____ <input type="checkbox"/></p> <p>Component quantization:</p> <p>1 byte (8 bits) <input checked="" type="checkbox"/> mandatory for bitstream and decoder**</p> <p>2 bytes (16 bits) <input checked="" type="checkbox"/> mandatory for the bitstream**</p> <p>3 bytes (24 bits) <input type="checkbox"/></p> <p>4 bytes (32 bits) <input type="checkbox"/></p> <p>Other: _____</p> <p>Number of components:</p> <p>1 to 4 (2 bits) <input checked="" type="checkbox"/> Mandatory on the decoder**</p> <p>1 to 16 (4 bits) <input type="checkbox"/></p> <p>1 to 64 (6 bits) <input checked="" type="checkbox"/> Mandatory on the bitstream**</p> <p>1 to 256 (8 bits) <input type="checkbox"/></p> <p>Other: _____ <input checked="" type="checkbox"/></p> <p>Independent component quantizations <input checked="" type="checkbox"/></p> <p>Independent component spatial sizes <input checked="" type="checkbox"/></p> <p>Colour spaces:</p> <p>Luminance (Y) <input checked="" type="checkbox"/></p> <p>RGB <input checked="" type="checkbox"/> There are several versions!</p> <p>YUV <input checked="" type="checkbox"/></p> <p>Other(s): _____ CMYK, CIE Lab* _____</p>	
2	Uncompressed	<input checked="" type="checkbox"/> e.g. small size images (thumbnail, ...)
3	Lossless Compression	<input checked="" type="checkbox"/> Mandatory for the bitstream** <input checked="" type="checkbox"/> For decoder to be further discussed
4	Visually Lossless Compression	<input checked="" type="checkbox"/>
5	Visually Lossy Compression	<input checked="" type="checkbox"/>
6	Progressive Spatial	<input checked="" type="checkbox"/> Mandatory on bitstream only(1)**
7	Progressive Quality	<input checked="" type="checkbox"/> Mandatory on bitstream only(1)**
8	Security	<input checked="" type="checkbox"/> To be clarified further

9	Error Resilience	<input checked="" type="checkbox"/> Mandatory on bitstream** To be clarified further for decoder
10	Complexity Scalability	<input checked="" type="checkbox"/>
11	Strip Processing	<input checked="" type="checkbox"/>
12	Sensor Specific Compression Flexibility	<input type="checkbox"/>
13	Information Embedding	<input checked="" type="checkbox"/> see annex C
14	Repetitive Encoding/Decoding	<input checked="" type="checkbox"/> N/A to bitstream, quality issue
15	Object-Based Functionality	<input type="checkbox"/>
16	MPEG4 VTC Compatibility	<input type="checkbox"/>
17	Backward Compatibility Client-Side Ease of Transcoding	<input checked="" type="checkbox"/>
18	Dynamic ROI	<input checked="" type="checkbox"/>
19	Fast/Random Data Access	<input checked="" type="checkbox"/>
20	Implementation Complexity	<input checked="" type="checkbox"/>
21	Static ROI	<input checked="" type="checkbox"/>
22 Part I	Geometric Manipulation	<input checked="" type="checkbox"/>
23	Compatibility with ITU-T Recommendations for image exchange	<input checked="" type="checkbox"/> T.4 (group 3 fax), T.42 (color encoding), T.37 (S&F internet fax), T.44 (MRC)

(1) Means that decoder should be able to stop at any layer without entirely decoding the bitstream

\* It is also possible to invoke the RCT as a reversible component transform within the encoder to provide improved compression efficiency performance on the lossless compression of an RGB color image. RCT does not define a color space.

\*\* Mandatory on the bitstream means that its has to be provided in the bitstream. Mandatory for the decoder means that any decoder shall be able to fulfill the requirement.

## Annex C : Features of JPEG2000 part I

Requirement Number	Requirement Description	Comments	Feature
1	<p>Image Type</p> <p>Natural</p> <p>Compound</p> <p>Pixels count per dimension (Width, Height): 1 to 4,294,967,295 (1 to <math>2^{32}-1 = 4</math> bytes)</p> <p>Component quantization:</p> <p>1 byte (8 bits)</p> <p>2 bytes (16 bits)</p> <p>Number of components:</p> <p>1 to 4 (2 bits)</p> <p>1 to 256 (8 bits)</p> <p>Independent component quantizations</p> <p>Independent component spatial sizes</p> <p>Colour spaces:</p> <p>Luminance (Y)</p> <p>RGB</p> <p>YUV</p> <p>Other(s): ____ CMYK, CIElab* ____</p>	<p>Mandatory for bitstream and decoder</p> <p>Mandatory for the bitstream</p> <p>Mandatory on the decoder</p> <p>Mandatory on the bitstream</p> <p>Not completely random</p>	<p>Y</p> <p>S</p> <p>Y</p> <p>Y</p> <p>Y</p> <p>Y</p> <p>Y</p> <p>Y</p> <p>Y</p> <p>S</p> <p>S</p> <p>S</p>

2	Uncompressed	e.g. small size images (thumbnail, ...)	Y
3	Lossless Compression		Y
4	Visually Lossless Compression		Y
5	Visually Lossy Compression		Y
6	Progressive Spatial	Mandatory on bitstream only	Y
7	Progressive Quality	Mandatory on bitstream only	Y
8	Security	Watermarks survive lossless Does not specify but is potentially compatible with various security solutions	C
9	Error Resilience	Mandatory on bitstream	Y
10	Complexity Scalability		?
11	Strip Processing		Y
13	Information Embedding	?	Y
14	Repetitive Encoding/Decoding	Lossless mode	?
17	Backward Compatibility Client-Side Ease of Transcoding		N
18	Dynamic ROI		S
19	Fast/Random Data Access		S
20	Implementation Complexity		?
21	Static ROI		Y
22	Geometric Manipulation		?
23	Compatibility with ITU-T Recommendations for image exchange	T.4 (group 3 fax), T.42 (color encoding), T.37 (S&F internet fax), T.44 (MRC)	S

Y: YES, S: Supports, C: Is compatible but does not specify, N: No

## Annex D : File format requirements

This section contains requirements regarding file format in JPEG2000. For practical editing reasons, this section is currently in the form of an annex but it is planned that these requirements be merged inside the body of the text in a later release.

### *Definitions*

- **Entity:** An group of related data structures. One entity may be composed of other entities. The smallest entity may refer to a group of tags or properties in the file, but in general, the term entity does not refer to a single tag. Examples of entities are the complete codestream, a directory of metadata, a single tile packet, ...
- **Minimal file:** the simplest file that can be decoded and interpreted by a majority of the application profiles. However, it is important to note that a minimal file will not meet the requirements for those profiles. Each application profile will extend the minimal file by adding metadata (defined either in Part 3 or by JURA registration) in order to meet the requirements for that application profile.
- **Full featured files:** a file that contains additional metadata to allow applications to provide advanced features to users of that file.
- **Conforming file reader:** an application that conforms to part 2 of the standard (the specification of the file format). This is not the same as an application that conforms to only part 1, and thus can only decode the codestream. However, all conforming file readers must also conform to part 1.

### *Requirements for binary container architecture*

- **Single scalable architecture:** There will be one and only one scalable file format architecture that can meet the needs of many different application profiles. Different application profiles will use the architecture in different ways to meet the needs of that profile.
- **Simple parsability:** The binary container must be parsable using software or hardware of similar complexity to the parsing of the codestream. However, this does not preclude the architecture from including hooks that allow a higher performance parsing mode.
- **Locating the codestream within the file:** The location of the compressed codestream and the essential metadata are well defined.
- **Location of entities:** The binary container must provide for random read/write access to individual entities in the file. This means that there must be some form of “directory” segment in a well defined location that specifies the location of other entities in the file. These entries include both portions of the compressed codestream and groups of metadata fields or objects. This architecture must allow individual entities in the file to be edited without forcing the entire file to be rewritten.
- **Optimizing files:** It must be possible to produce an “optimized” version of any file. For example, to stream an image from a server to a client, certain elements must be found in the file in a certain order.
- **Evolving minimal files to full-featured files:** It must be possible to easily “evolve” a minimally decodable file into a more complex (and thus less optimized) file without rewriting the entire file.
- **Format overhead:** The minimum overhead carried by the file format must be small.

## ***Requirements for metadata***

### **Architecture**

The architecture describes what the metadata specification should be able to facilitate. It does not include specific types of metadata or attributes of that metadata.

Only the architecture and a very small number of metadata properties related to the physical characteristics of the image and the intellectual property will be defined as part of the minimal data format. Additional metadata will be defined in extensions to the minimal format.

- **Metadata syntax:** The architecture should describe some mechanism for accessing the syntactic meaning of the metadata.
- **Preventing conflicts:** The architecture should provide a means to prevent conflict between multiple metadata entities.
- **Locating the metadata:** The architecture should make it possible to determine the locations of all of the metadata within the file.
- **Determining the type of metadata:** The architecture should make it possible to determine the types of metadata and their attributes in the file without parsing all of the metadata.
- **Metadata features of the codestream:** The architecture should make it possible to treat the codestream as metadata.
- **Referencing metadata:** The architecture should support a method for one metadata object to refer to another object.
- **Securing metadata:** The architecture should make it possible to provide a method to secure metadata.
- **Extending entities:** The architecture should support extensibility. This means that it must be possible to expand the fields of the defined metadata types while maintaining backward compatibility.
- **Ignoring non-essential metadata:** The architecture should support a method of extracting essential metadata (such as image width) without necessarily understanding how to extract the non-essential metadata.
- **Coding efficiency of metadata:** The architecture should support a method of storing essential metadata using an amount of space that does not hinder the efficient transmission of the file.

### **Attributes**

This section describes the attributes that JPEG2000 should define. These attributes will be used to provide additional information about the individual metadata in a file.

- **Association of metadata with regions:** The specification should define a mechanism by which metadata can be associated with particular regions in the image to realize object-based information embedding.
- **Editing metadata:** The specification should support some method of determining whether or not a particular metadata is editable.
- **Download priority:** The specification should indicate at what point metadata should be downloaded in a network application (for use when a file is being transmitted in parts in an interactive process)

- **Updating metadata:** The specification should support the ability to indicate that the value of this metadata can be updated automatically from a server and the location of the update.
- **Metadata creation and modification dates:** The specification should define creation and modification dates of individual metadata.

## Types

### Colorspace specification

- **Interoperable color management:** The file format shall include a color management specification that will be required to be interoperable by all conforming readers. Other color management schemes may be optionally included.
- **Standardized colorspaces:** The file format must allow the colorspace to be specified using a flag indicating that the colorspace is one of a small number of well-known, unambiguous colorspaces, or that the image is to be treated as if it is in that colorspace. A defined subset of these listed colorspaces shall be supported in all applications that support the file format.
- **Non-photographic images:** The file format must provide for the specification of component definition for non-photographic images (multi-spectral or scientific data, ...)

### Other physical or IPR metadata

- **Unambiguous resolution:** The file format shall allow files to specify the original scanning resolution of the image
- **Intellectual property:** The file format shall specify a mechanism by which intellectual property metadata is located.

### *Requirements on applications that read the JPEG 2000 file format*

- All readers must be able to decode completely all low-complexity files, and all readers must be able to decode ALL files in a fail-safe manner (meaning that they either decode the file correctly or they are aware that they can't).

## Requirements for further extension of the standard beyond part 1

The following are requirements of the extensions of the standard. It is intended that this part of the standard contain enhancements to the minimal file format based on the architecture defined in part 1.

These requirements imply a set of entities that must be defined as part of the standard. However, the use of these entities is optional with respect to conformance.

### *Requirements for metadata*

## Types

This section describes specific types of metadata that JPEG2000 should define. It is hoped that JPEG2000 will use the same definitions for specific metadata (i.e. audio, text) that other standards such as MPEG4 and MPEG7. However, there will be metadata types that are unique to JPEG2000 that require JPEG2000 to define them.

- The specification should define a small, fixed length data structure (“passport”) which links a derived image back to an original image. The passport specifies generic information about the original image, such as pixel dimensions, and number of components. For images registered with a registration authority, this passport will contain the license plate of the image. For all images, the passport is a small data structure.
- The specification should define a means to specify the service or class of features that are provided by the file.
- The specification should define metadata to specify the IPR of the image.

## **Scenarios for further extensions of the standard**

In this section, we provide two examples of application scenarios for information embedding. We also discuss how particular metadata are necessary to realize these scenarios in practice.

### **Application 1: Enhanced Electronic Postcard**

Example Scenario: I took a picture of John in France containing a chateau in the background. John wants to email an enhanced version of this JPEG2000 image to his brother Tom. John decides to embed information into two regions in the image. Regions are specified by two rectangles, one rectangle surrounding his face and the other overlapping with a part of the background building. Using an appropriate application, he embeds an audio clip, and an URL for his personal web site into his own rectangle. The audio clip is a personal message to his brother, and his web site contains other pictures from France. John embeds a brief text paragraph about the history of the chateau into the rectangle associated with the chateau.

Upon the receipt of the image, the sequence of events in the viewer application are as follows:

- E1: The viewer application will decode and display the image;
- E2: Tom will discover the two regions containing information;
- E3: Tom will interactively find out the nature of information (e.g., audio, URL) associated with the regions;
- E4: Tom will retrieve and enjoy the information of his choice for each one of the regions.

Specific requirements enabling the scenario: The above events are enabled by the tools that satisfy the current set of metadata requirements. We reference below to the associated requirements for each one of the above events. (Legend: *arch#x* = Architecture requirement #x; *type#y* = Type requirement #y; *attr#z* = Attribute requirement #z). (See the Appendix for requirements).

- E1: The viewer application decodes and displays the image using the essential metadata;
- E2: The essential metadata contains a catalogue/list of types and attributes of metadata according to *arch#3*, where metadata include position data for image regions according to *attr#1*. The viewer application then immediately decodes and utilizes the rectangular region position data since metadata can

be prioritized according to *attr#3*. In this case the rectangular region position data is of highest priority and meant to be utilized immediately;

E3: The viewer application is capable of realizing this event according to *arch#1*, *arch#2*, and *arch#5*. In particular, *arch#5* is for associating a particular metadata with a particular region.

## ***Application 2: Self-Descriptive Map Service***

Example Scenario: I am interested in a "self-descriptive" map of the city of Portland, OR in the USA. We assume that such maps are available at a particular web site as JPEG2000 files. This special site serves JPEG2000 images of maps that are self-descriptive by way of embedded audiovisual information, for example some descriptive text about a landmark, an audio message about a street, or a thumbnail image of the area. This particular server is assumed to be intelligent and has full capability of parsing a JPEG2000 file format.

The sequence of events in a typical application are as follows:

E1: The client application downloads, decodes and displays the map image ;

E2: The user discovers the regions in the map that contain additional information;

E3: The user finds out the nature of information (e.g., audio versus text) associated with a region and downloads a particular information that is of interest. The user repeats this process as needed for other regions.

Specific Requirements Enabling the Scenario: The above events are enabled by the tools that satisfy the current set of metadata requirements. We reference below to the associated requirements for each one of the above events. (Legend: *arch#x* = Architecture requirement #x; *type#y* = Type requirement #y; *attr#z* = Attribute requirement #z). (See the Appendix for requirements).

E1: The viewer application decodes and displays the image using the essential metadata;

E2: The essential metadata contains a catalog of metadata according to *arch#3*, where metadata include region position data according to *attr#1*. The rectangular region position data are also downloaded immediately since it is flagged accordingly on the basis of *attr#3*;

E3: Server and the client applications are capable of realizing this event according to *arch#1*, *arch#2*, and *arch#5*. In particular, *arch#5* is for associating a particular metadata with a particular image region.

## **JPX file format requirements**

This document defines version 1.0 of the requirements for the JPX file format. While these requirements are described with respect to specific applications, it is well understood that many if not all of these requirements are applicable to many different application scenarios.

### ***Digital photography and commercial graphics***

1. The standard must allow for the storage and specification of many different colorspace.
  - (1) Images are captured, edited and output in many different colorspace, including non-RGB colorspace. Most commercial scanning and output today is done using CMYK class colorspace.
  - (2) Applications can often not control exactly which colorspace must be handled.
2. Color information for the image must be accurately communicated

- The color information for an image represents the colorspace as well as other parameters related to its capture and rendition.
- Optimized processing can often be performed on specific well-understood colorspace
- Applications will natively implement only a small number of colorspace. General methods are needed to provide maximum interoperability
- In the commercial graphics workflow, it is becoming more and more desirable to encode images in full-gamut device independent RGB colorspace instead of storing the images in device specific CMYK spaces. The transformation from device independent to device specific CMYK is not performed until the proofing stage or when final films and printing plates are created. However, there are many options within the scope of this transformation, such as the compression of the wide-gamut to fit within the limited gamut of the printing press, or the desired tone reproduction.

The combination of a full-gamut RGB image, its colorimetric definition, and a set of desired reproduction intents for output on a CMYK device is often referred to as "virtual CMYK."

In order for a proofing or film output operator to reproduce the image as intended by the original creator, information about the desired reproduction intent must be submitted with the image data. Without this information, the proofing system operator, for example, does not have the enough information regarding the desired reproduction intent to create an accurate proof from the full-gamut image.

Note that this is not needed in all workflows, but is a requirement in North America for the distribution of advertising where the agency wants control of the final output reproduction. This workflow is one of the major application areas that we see for the JPEG 2000 file format standards.

- Applications can make use of information about the original capture conditions to help the user improve the image
3. The standard should allow for the efficient encoding of many different colorspace
    - Different colorspace have different statistical distributions, and may require different decorrelation transformations, including a step to gamma correct a linearly encoded colorspace.
    - Note that this there must be a clean division between transforms used to interpret (display/print/analyze) the image data from transforms used to achieve better compression of the image data (e.g. multiple component transforms in the codestream).
  4. The standard should make a clear distinction between rendered and scene colorspace
    - For example, today's image editing applications can often only deal with rendered scenes; scene colorspace must be rendered upon input to these applications.
  5. The standard should maximize interoperability between applications
    - If the standard has options (individual applications may decide which features will and will not be implemented), there must be an easy way to recognize which features have been implemented.
    - Consumers are very intolerant about non-interoperable files. If a reader can't easily and learn (and thus inform the user) why a file cannot be read or what needs to be done to read the file, the perception of the application and/or JPEG 2000 suffers.
    - If the reader doesn't support the functionalities required by the file, it should be able to easily "fail gracefully."
    - The application must be able to determine which combination of features must be supported to successfully use the image file as intended by the writer.
    - It is desirable for a reader to be able to succeed reading a file to the extent of that readers capability.
    - File authors desire to provide multiple "interoperability" levels for individual files, allowing different readers to access the files in different ways, without providing conflicting views of the file.
  6. The standard support should define standard ways to store information about the image (metadata)
    - Customer satisfaction is greatly increased when we keep track of more information about their memories, such as the people, places, things in the image.
    - Professional photographers require that their IP rights are protected when creating digital image files. A standard can't protect their rights, but the standard must provide enough information to an application, which will protect a creator's rights. This means that the IPR information must be in a well defined format, and that it must be easy to determine if the image is protected.
    - Technical information about the capture conditions must often be communicated between applications

- It is important to record how an image has been edited. This can affect the application of IPR, as well as affect future processing of the image. For example, when printing the image, the image should not be sharpened in the printer if sharpening has already been applied.
- Applications can make use of information about the original capture conditions to help the user improve the image

### ***Serving images interactively***

This application area involves situations where some form of client and server that interactively communicate to request specific portions of the image and metadata. Note that for this application area, clients and servers don't have to be different machines.

7. The standard should allow for the storage of partial images
  - A web browser may be stopped after only receiving a portion of the image. It is beneficial (in terms of simplifying the image viewer) if there is a standard way to represent that partial file, and to point back to the original, complete file.
  - Streaming of images from a server can often be greatly improved if the codestream can be split among different disks. This is like a pseudo RAID system, except that the file writer can determine how to best divide up the codestream based on the expected access patterns
  - A high-performance web site can be built using multiple lower-cost components.
  - Scalability improved because different processors can be more quickly fed from multiple disks.
8. The standard should allow the individual images from a multi-image file to be accessed individually
  - Applications often require access to a particular page of a multi-page file.
  - Applications may desire to view a single frame from the middle of an animation.
9. The standard should facilitate information-aware delivery  
Information-aware delivery is where the reader uses information found within the file to determine which portions of the file should be downloaded or processed.
  - A web browser may wish to interrogate the metadata before requesting any image data, in order to determine available resolution levels, available images, ...
  - A web browser may request only part of the available image or image set.
  - An application may want to know "summary metadata" (height/width, ...) without displaying the image
  - An application may want to only display a thumbnail image.

### ***E-commerce***

10. The standard should allow for a single image to be split across multiple files/network
  - Vendors often send out CD's of thumbnail images to customers. The customers can then browse the images, select one for purchase and then go to a separate application or web site to purchase; the low resolution is on the CD, the high resolution is on the internet in a protected site. It would be valuable if that low res file pointed directly to the high resolution.
11. The standard should allow for portions of the file to be encrypted
  - A vendor can reduce cost by putting both the free thumbnail images and the "for sale" high-res images on the same CD in the same files. By encrypting the high-resolution layers, users can browse through the thumbnails, but can't gain access to the high-res until they have purchased a CD from the user.
  - Files may contain sensitive information, such as the name and credit card number of the person submitting an image for digital hardcopy fulfillment.
  - A vendor might want to sell the metadata independently from selling the image

### ***Photographic Image Editing Application***

A number of applications are used for creation, or touching-up of images. These applications will generate an output file which is the combination of all the image editing operations performed on the file. Also, there is the need to store an intermediate image which is a number of images placed on top of each other to produce the final image. The benefit of storing the number of intermediate images, is so that further image manipulation may easily be performed.

12. It must be clear how images of different colorspace are combined into a single output

- The combination of multiple colorspaces is problematic and complex. Users taking advantage of this feature must get a consistent result.
13. The standard must allow for a reader to only support the combination of multiple images that are in the same colorspace
    - The combination of multiple colorspaces is problematic and complex; it is an implementation burden in many scenarios.
  14. It must be possible to represent multiple images as multiple codestreams
    - It is often necessary to edit auxiliary channels (such as opacity) separately from the “color” channels. Grouping them into the same codestream complicates the rewriting of the edited file.
  15. The standard must specify how the images are to be combined (compositing operations)
    - Many applications use opacity channels to combine images, and use different Porter-Duff modes to affect the combination
    - Often, source images are different sizes (adding a company logo into the corner of a photograph)
  16. The standard should support a “fast-save” application behavior
    - Changing a single tile of the image (such as performing red-eye correction) for most compressed formats requires rewriting the entire file. This can be a cumbersome process if the user is just saving and then continuing work (and not quitting and doing something else). Many applications in other classes (such as word-processors) allow a fast-save option, where only the changes are written back to the file. Because of the highly segmented JPEG 2000 codestream, this fast-save behavior can be enabled for images by allowing a single portion of the codestream to be written to the end of the file instead of rewriting the entire codestream in order

## **Animation**

Dynamic composition of multiple still images has become a widely accepted model for animation in the web. Currently, there is a significant amount of animation on web pages. This appears to be done in one of two ways.

- Describing the animation in a single file
- Describing the animation using a number of separate files and a script

Describing animation within a JPEG 2000 image file could greatly increase the adoption of JPEG2000. This can be inferred by the degree of interest in emerging specifications such as MNG. The MNG group has taken steps to add alpha channel support to baseline JPEG as well as creating an animated file format. The implicit requirements: transparency in a lossy image format and simple animation

An animation is a sequence of frames, displayed one after another.

17. Minimal file size
  - This is a compression standard. Self-explanatory
18. Relatively simple decoder (player) implementation
  - Adoption in a wide variety of applications is dependent on ease of implementation with respect to other standards
  - Many of the expected JPX features are not required for simple animation. For example, it is reasonable to disallow things like full ICC support in animated players.
19. Be able to begin display the animation during transfer
  - Users are often not willing to wait for the entire file to download before starting it
  - There needs to be something to display as the rest of the file downloads
20. Displayed as a still image
  - To maximize interoperability, there needs to be a defined behavior (other than refusal to open the file) when a non-animated viewer opens the file.
  - Animations must have a printable form.
21. All well-known animation features should be supported
  - Animations both stop once they reach the end and loop
  - Varying time delay between displaying frames
  - Images that comprise frames may be of different sizes.
  - Sprite-animation, including multiple sprites
22. Artifacts generated during playback should be minimized
  - Using only pixel replacement can introduce artifacts.

### ***Digital image workflow (system interaction)***

Once a digital image is created, more than one application (such as editor / printer) will want to access data. Such applications may also be performed under quite separate conditions. For example, different printers or video display screens may be used.

23. Information on the colorspace in which the image data is stored must be included in the file
24. It must be easy to determine what information is stored in the file
  - The file should not have to parse the file byte by byte to determine what is in the file
25. It must be easy to determine how to process the information stored within a file
  - The file should not have to parse the file byte by byte to determine how to process the file

### ***Image manipulation (RIP/scaling/rotation)***

Quite often it is useful to combine a number of images (and other graphic objects together) to produce a result. Such a result may be printed / saved for later use or used as input to another process. This combination of images is a complex task, and may require a large amount of processing and memory. For this reason, a number of optimizations to the RIP process need to be performed. These optimizations require the following functionality:

26. Random access (spatially) to the image data.
  - This is required as a number of images may be being processed simultaneously, and all images may not fit in RAM. As such, partial images need to be read.
  - Rotating and scaling the image efficiently often requires that the image data be accessed out of order.
27. Access to the image at different resolutions.
  - This is required, as images may be zoomed to varying sizes on a single output page, and as such, for some images, lower than full resolution may be required.

### ***Network and internet transmission***

28. It should be possible to compress auxiliary data in the file (such as metadata)
  - XML metadata is very verbose. Transmission time can be greatly reduced if the text is compressed.
29. It should be possible to protect sensitive aspects of the image from prying eyes
  - The internet, in general, is not private. It is desirable to use other mechanisms to protect sensitive data within the file before the file is shipped over an open channel.

### ***Scientific imaging***

30. It must be possible to accurately specify the scientific meaning of the image data
31. It should be easy to determine that remote sensing data is such, ...