JPEG 2000 .jpm file format: a layered imaging architecture for document imaging and basic animation on the web

Louis H. Sharpe II[a], and Robert Buckley[b]
[a]Picture Elements Inc.
[b]Xerox Architecture Center

ABSTRACT

The JPEG 2000 compression standard includes several optional file formats called the JP family of file formats. One, the JPM file format (file extension: .jpm) is aimed at compression of compound images: those have multiple regions, each with differing requirements for spatial resolution and tonality. Document images are common instances of compound images. B applying multiple compression methods, each matched to the characteristics of a distinct region, significant compression advantages can be achieved over use of a single compression method for the entire image.

1. JPEG 2000 FEATURES RELEVANT TO DOCUMENT IMAGING

JPEG 2000 is typically thought of as a wavelet-based compression method primarily aimed at compression of photographic images, but in fact it is intended to support a wide range of image applications, from digital photography to medical imaging to remote sensing and document imaging. Significant classes of document images now contain tonal and color information.

1.1 Support for reprocessibility

JPEG 2000 promises to become a commonly used compression means for moving high-quality image data in and out of high-speed document scanners and printers. Since it provides a full-tonality representation of a document image, the smooth characteristic of edges can be preserved, allowing a wide variety of post-processing steps to occur to create derivative images suited to disparate applications or devices. Many systems have been deployed which deliver thi flexibility using baseline JPEG DCT coding; JPEG 2000 offers superior image quality or smaller image sizes. The use of compressed tonal images makes such post-processing architectures practical. Scanners delivering JPEG 2000 data can have the skew removed b rotation of the decompressed data. Data can be scaled to multiple different resolutions for different purposes (e.g. web viewing and printing). Color forms can have different colors dropped-out in different region from the same source image fo data extraction. Printers receiving JPEG 2000 data can adjust the bitonal image sent to the print head to compensate for the specific tonal reproduction curve (TRC) of the device.

1.2 Browsing and viewing

The JPEG 2000 compression standard offers several mechanisms to ease browsing and these are directly applicable to document imaging.

Compressed datastreams may be organized as progressive by quality or progressive by resolution. This allows a rough overview image to be quickly presented to the end user who can then make decisions about which portions of the image should be viewed in greater detail. This can occur at a relatively fine spatial granularity determined by the spatial size of the code blocks. Fragmented codestreams allow greater flexibility in streaming data.

Tonal images are much more suitable for viewing on current moderate-resolution displays, even when viewing primarily bi-modal content such as text.

The artifacts introduced by the wavelet transform in highly-compressed JPEG 2000 images are much less objectionable than those produced by DCT JPEG. A flattening of the tonality within strokes of a document image is a much less jarring artifact than the ringing and blockiness seen with the DCT at high compression ratios. Such a flattening can actually idealize the strokes of a document image, by removing accidental subtleties in marking device pressure or smoothness.

1.3 Palletized color

The addition of palletized color compression to the JPEG 2000 standard offers interesting possibilities for document images, such as providing low tonality (2, 3, or 4 bit/pixel) images for anti-aliasing bitonal content on moderate resolution displays.

While 1-bit per pixel data will not compress well compared to other schemes, the palletized color compression method allows system designs having both tonal and bitonal data, but only one encoder-decoder.

1.4 Metadata architecture

The JP family of file formats offers a rich metadata mechanism. XML metadata boxes may be standardized fo metadata structures of wide industry applicability. An alternate method using globally unique identifiers allows proprietary or ad-hoc metadata to be added to files as well.

1.5 JPEG codestream

The JPEG 2000 coding engine makes use of tiles, components, decomposition levels (resolutions), precincts (spatial region across the 3 subbands at a given resolution level for a tile and component), codeblocks and layers. (For a description, see Reference 1.) The resulting JPEG 2000 codestream is organized as a series of packets, where a packet contains a given number of bit-planes from a codeblock in the precinct for one component of a tile. The order in which the packets are presented to a JPEG 2000 decoder determines the order in which the image is decompressed, and different packet orderings lead to different progression modes upon decompression. For example, if the packets are grouped by resolution level, and these groupings of packets are presented in order from the highest decomposition level (lowest resolution level) to the lowest decomposition level, then the result is progressive decompression by resolution. Altogether, JPEG 2000 supports progression by resolution, quality, spatial location and component (or color). While the packets for a given codestream are normally contiguous in a file, they can also be fragmented or interleaved with other, either image or non-image data.

1.6 Multiple layer imaging

Multiple layer imaging is defined in the JPM section of the JPEG 2000 standard. It is of key interest to document imaging and is covered in the remaining sections of this paper.

2. MULTIPLE LAYER IMAGING

Multiple layer imaging enables the use of different compression methods in different regions of a compound image. Pixels o regions of pixels may be sorted to any of multiple layers where different pixel depths, spatial resolutions and compression methods may be used within the same composite image file.

2.1 Background

2.1.1 The LFr system

In January 1992, one of the authors (LS) submitted a Picture Elements white paper to Vinton Cerf of the Corporation fo National Research Initiatives (CNRI) describing a document-progressive browsing system called LFr (pronounced “leafer”) designed for browsing document images over a low-bandwidth network. The output of a document understanding operation resulted in several layers of progressively-refined rectangle descriptors in the server representation of the document. The coarser, initial layers described paragraphs of text and photographs as typed rectangles. As bandwidth permitted, deeper layers revoked each of these initial rectangles for a page and replaced them with smaller, more refined rectangles emblematic of text lines and search hit words. At the deepest layers, actual image data was sent for entire paragraphs, with 37, 75, 150 and 300 dpi ITU-T T.6 compressed bitonal images providing progressive refinement. A protocol was proposed which permitted the order of refinement of layout objects to be defined by the client user’s mouse position over the current compositing of the page. This approach of using document understanding to inform a remote document browsing system was described during a talk at the February 1992 SPIE conference in San Diego [2]. The image browsing LFr system was subsequently prototyped and demonstrated at an exhibition in the National Digital Library Visitors’ Center in the Library of Congress in Washington, DC in 1993.

2.1.2 Layered imaging at the Library of Congress

Michael Ott, Cynthia Ott, Rick Crowhurst and one of the authors (LS) at Picture Elements developed and demonstrated a layered imaging compression scheme for the Library of Congress in 1995-1996. Two approaches were described. The first promotes information associated with high-strength edges to one of two high-resolution bitonal layers, a drive-white layer or a drive-black layer which are compressed using ITU-T T.6 compression. These features are removed from the source b averaging. The residual source image is lowered in spatial resolution and compressed using baseline JPEG. The image i assembled from the three planes, such that all pixel locations not driven black or driven white are transparent and the residual image shows through. The second method had three planes, an edge-location plane at high resolution, and two low-resolution luminance planes: a luminance dark plane and a luminance light plane. The latter two planes identified the shade of gray used for the dark and light pixels of the bitonal image in the corresponding region. This method was implemented and had good performance except in halftone regions, and thus would have required an additional method to exclude them. Sample images were generated, but the layered imaging approaches were not recommended for use by the Library owing to the fact that the were not standardized. RFC 2301, issued in 1998, solved that standards problem [3].

2.1.3 Antique Books at CMU

The Antique Books web site at Carnegie Mellon University demonstrated a means of representing a two-layer image by laying a transparent GIF over an HTML background JPEG image [4].

2.2 Mixed Raster Content

2.2.1 Facsimile applications

Mixed Raster Content is an ITU-T standard that originated in facsimile [5, 6]. By the mid-1990's, facsimile had a number of methods for the transmission of black-and-white and color documents. For black-and-white (bitonal) content, there was Group 3, Group 4 and JBIG compression; for lossy color, JPEG; and for lossless color, JBIG, including palette color. Each of these compression methods was designed for a particular kind of content and was applied to all pages in a facsimile session. The problem however was what to do with a page that contained a combination of bitonal and color content. Originall described in a joint proposal by Xerox and Hewlett Packard in 1996, the ITU-T Mixed Raster Content (MRC) standard was designed to solve this problem. Instead of inventing a new compression method, MRC defines a model for using existing compression methods to represent images that contain a mixture of raster content.

The base mode of MRC decomposes a mixed-content image into 3 layers: a bitonal Mask layer and color Foreground and Background layers. Recomposing the image is a relatively simple operation that uses the Mask to select pixels from th Foreground or Background layer for the recomposed image. Wherever the Mask layer pixel value is 1, the corresponding Foreground layer pixel is used; when its value is 0, then the corresponding Background layer pixel is used. The power of MRC comes from distributing the content from the original image into different layers, with like content being placed in th same layer where it can be encoded in a way best suited to the content of that layer and differently than the content of the other layers. Besides compression, the encoding includes the choice of spatial resolution and bit depth.

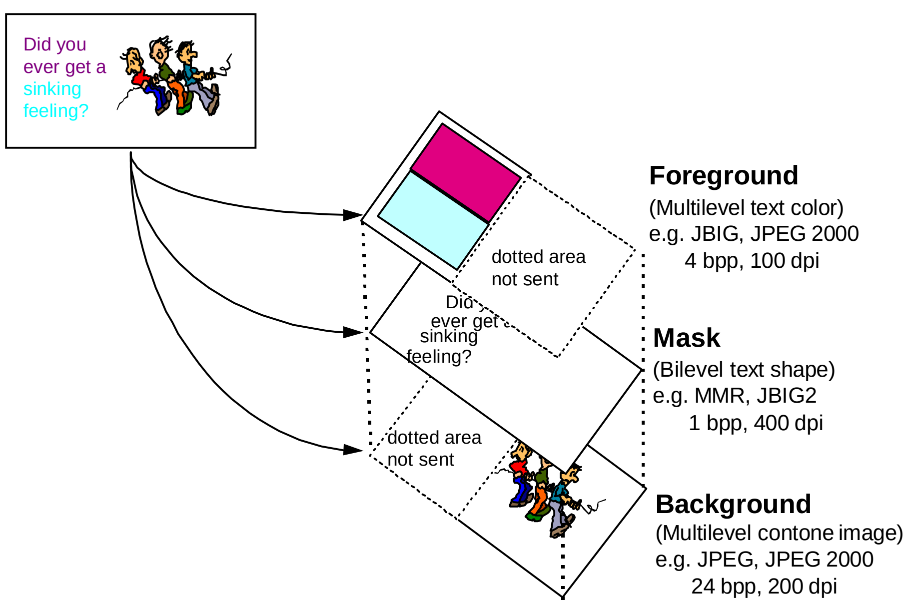
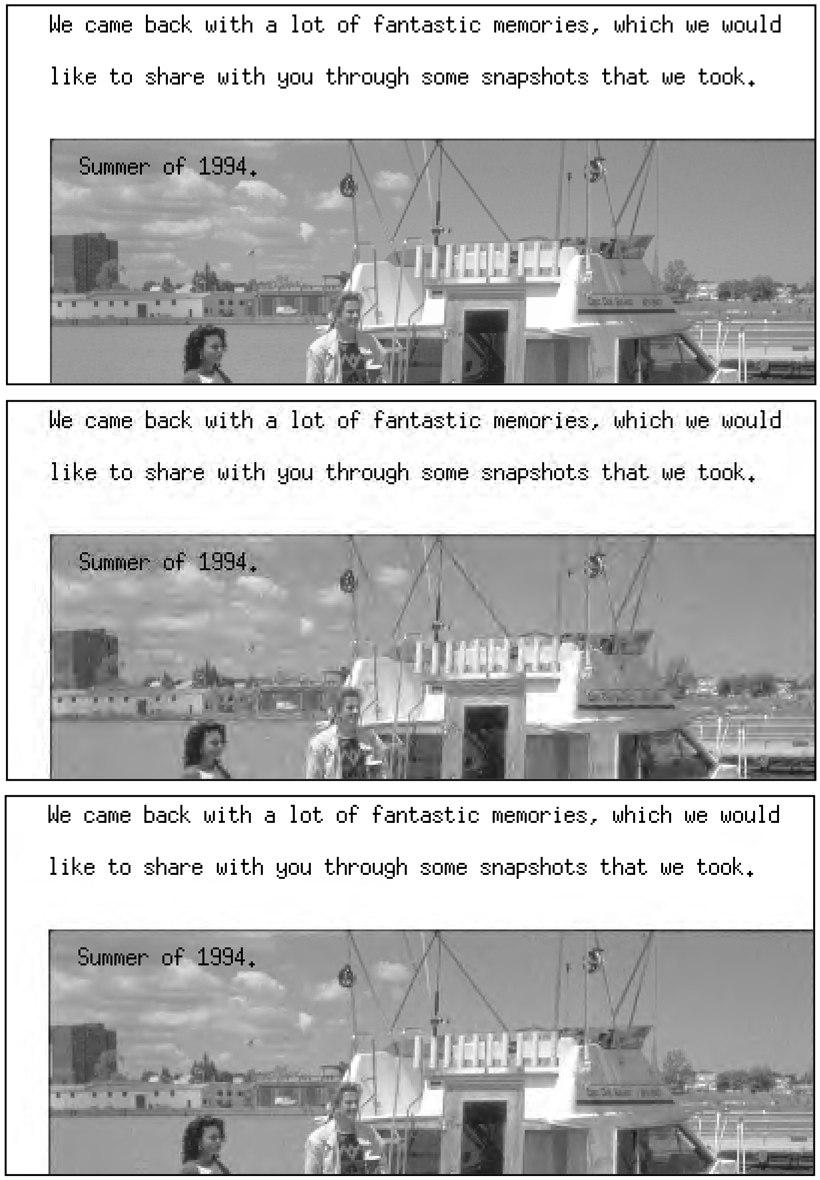


Figure 1. MRC Example

Consider a simple example, shown in Figure 1, with a facsimile page containing a color image and colored text. In this case, the text shapes are placed in the Mask layer, the color image in the Background layer, and the text color as rectangles in the Foreground layer. Since text typically requires higher spatial resolution for edge rendition than images, the Mask layer can be at a higher resolution than the Background layer, or alternatively, the Background layer can be subsampled with respect to the text in the Mask layer. The Mask layer would be compressed losslessly, using Group 4 or JBIG2 for example, and the Background layer would be compressed using an algorithm, such as JPEG or JPEG 2000, suitable for color images. In this example, the Foreground layer could be encoded as a palette color and losslessly compressed. In addition, only those portions of the layers containing valid image data need be encoded; thi white-space skipping further improves the compression performance of MRC. Extensions to the ITU-T MRC standard now support more than 3 layers, where additional layers are added as Mask-Foreground pairs.

Figure 2 shows an example of MRC with JPEG 2000 applied to cmpnd1, one of the compound images from the JPEG 2000 test image set. Figure 2(a) shows a portion of the original image. Figure 2(b) shows the same portion compressed with JPEG 2000 at 0.5 bits per pixel. In Figure 2(c), MRC has been applied, with JPEG 2000 used for the image regions and JBIG2 for the text regions; the overall bit rate in Figure 2(c) is 0.5 bits per pixel. In this case, only 2 MRC layers are used, the Mask and Background, with the Foreground layer absent and defaulted to black. This simple example shows the improvement in quality (or alternatively compression) with MRC.



(a) Original

(b) JPEG 2000 at 0.5 bpp

(b) MRC at 0.5 bpp

Figure 2: Comparison of Original, JPEG 2000 compressed and MRC compressed images

2.2.2 TIFF-FX File Format

While originally developed for Group 3 facsimile, MRC can also be used with Internet facsimile, where transmission occurs over the Internet instead of phone lines. In Internet Fax and store-and-forward applications, the same compressed image data used in Group 3 facsimile transmissions is wrapped in a TIFF file. The file format standard for Internet Fax is TIFF-FX (TIFF for Fax eXtended), which specifies several profiles, one for each of the standard methods of facsimile [3]. Among these is Profile M, which specifies the TIFF representation of MRC data. The TIFF-FX standard, including Profile M, ha undergone interoperability testing, as required by IETF procedures, and at the time this was written, was in the process of becoming a Draft Internet Standard. The MRC formats used in fax applications support in effect a spatially progressive mode of decompression, where a layer is further decomposed in non-overlapping strips, which are rendered sequentially. Thi approach can be traced to the early days of fax, when image transmission occurred between synchronous senders and receivers without memory.

2.2.3 MRC Applications

The MRC concept is now found in other applications besides facsimile. One of the first was to apply it was the ScanSoft (formerly Xerox) Pagis product, which uses the multi-layer XIFF image format, with a combination of standard and proprietary methods for compressing the image layers. The proprietary methods are a wavelet-based one for color and a symbol-matching precursor to JBIG2 for bitonal [7]. Th DjVu product developed by AT&T and now owned by LizardTech offers proprietary wavelet compression for the color Foreground and Background layers and symbol-matching compression for the bitonal Mask layer [8, 9]. DjVu was developed intended for the distribution of color document images over the Web. Luratech has a similar product called LuraDocument, which also uses proprietary wavelets for the color layers and context-based bitonal compression for the Mask layer [10]. The Image Powe Power Compressor Toolkit also offers MRC functionality, with optional wavelet compression [11]. Most of these products are designed for the management of scanned images with mixed color and bitonal content.

There are two observations to be made here. One is that there are multiple MRC-based products with wavelets. Another i that these products for the most part use proprietary compression methods and non-interoperable formats. JPEG 2000 with an MRC-based format that takes advantage of the JPEG 2000 capabilities promises an open and interoperable standard to promote use of mixed-content images in new and emerging applications.

2.3 Simple animation model

While support for animation will now also occur elsewhere in the JP family of file formats, one short data structure with looping, persistence and timing parameters makes a simple form of animation possible in JPM. The basic concept of laying image content down in rectangles on a composited frame is basically the same in MRC and in animation. Support fo animation using JPEG 2000 is also likely from established animation standards such as MNG/JNG [12].

3. JPM FILE FORMAT IN JPEG 2000

The MRC-based format in the current JPEG 2000 working draft is called JPM (JPEG 2000 Multilayer), which is a member of the JPEG 2000 family of file formats. Like the other file formats in JPEG 2000, it is based on the architecture of JP2 , the proposed minimal file format for JPEG 2000 compressed image data. One of the ways of thinking about JPM is offering another level of progression, by MRC layer, that intersects with progression modes enabled now by JPEG 2000 codestream packets.

3.1 Layout objects

The current version of JPM is based on the idea of layout objects, a name that may change as the JPEG 2000 progresses, but which captures the essential idea of MRC in JPEG 2000. A layout object consists of an operator, a mask object and an image object. The operator describes how the mask and image are combined. In terms of the MRC model, a layout object is basically a Mask-Foreground image pair, where each image of the pair is represented by an independent codestream, possibly fragmented into groups of packets.

Mask objects and image objects are constructed the same way. These objects describe where the mask or image is placed within a frame and point to the mask or image data. The frame is the area within which the layout objects are imaged; it has a width, height and resolution.

3. 2 Layout object table

The layout object table identifies in one place the codestream layout information for the page. The layout table contains all the layout objects in the file, along with their descriptions.

Figure 3 shows an example of a simple compound image with two layout objects. The first layout object represents a color image, and consists of: an image object for the color image itself and a mask object that in this case is the same size as the color image. The second layout object represents a body of text. Its mask object is the text itself; its image object is the colo of the text, which is black. In general, layout objects can overlap, although in this example, they do not.

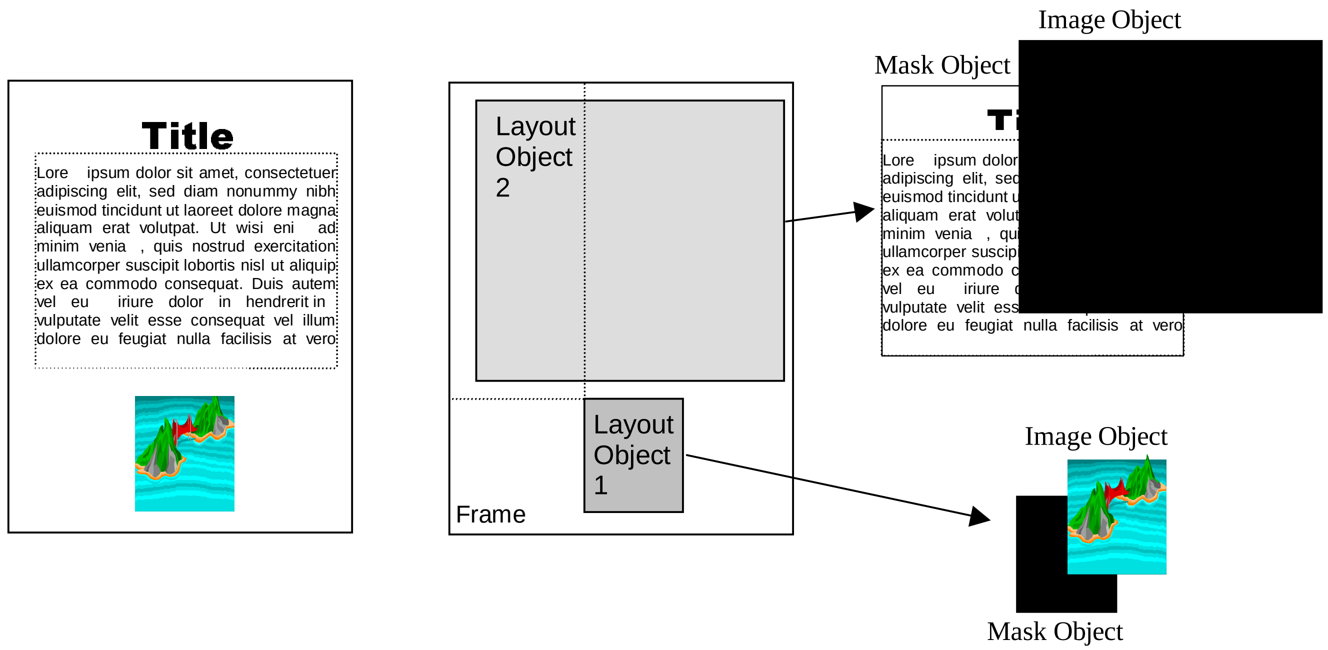


Figure 3: Layout Objects in the JPM file format

The mask and image objects that make up a layout object are each defined by a set of parameters The parameters are: an object identifier; the object coder identifier; the height and width of the area within the frame in which the object is imaged; the horizontal and vertical offset of the origin of this area with respect to the origin of the frame; an object base color, which is the pixel value to use when there is no coded data is present; and finally the location of the object's image data. The image data can either be stored within the JPM file, in which case the location is the offset within the file to the start of the object's image data, or it can be stored external to the file. In the latter case, the object's location is a reference, possibly a URL, to th data file or resource that contains the object's image data.

In the example of Figure 3, the Mask Object of Layout Object 1 and the Image Object of Layout Object 2 are both black images (pixel value 1) and can be compactly represented using the Object Base Color. For compatibility with the base MRC model, the Mask Object of Layout Object 1 (the Background Layout Object) would be a black image. Also, the image and mask objects of both layout objects have the same size in the frame, although there is no requirement for this to be so in general.

The width and height of an object, given in frame resolution units, can be different than the width and height of the image data. When this happens, the image data is scaled before it is imaged in the frame. Normally the resolution of the frame is the same as that of the highest resolution Mask object.

In a typical JPM file, the Image Object Coder is JPEG 2000. The Mask Object coder can be any suitable binary coder, such as JBIG2.

3.3 Thumbnail and JP2 compatibility

A thumbnail or other image may appear in a JPM file to provide compatibility with the simple JP2 file format defined as an optional feature in JPEG 2000 Part 1. This ensures that simple readers which are JP2 compatible, but not JPM compatible, will give the user some idea of what the compound image looks like. Previewing functions within operating system file browsers can make use of this feature as well.

3.4 Fragmented codestreams

A key benefit of fragmented codestreams is to allow interleaving of multiple codestreams. This gives a better user experience as the initial rough layers can be sent first for all regions of the page. Interactive feedback from the end-user's mouse position could also be used by a client program to request alternative orderings of fragments, different from the ordering found on the server. Thus, a layout object of interest, or a sub-region of a layout object, could be refined to a greater degree than the surrounding regions of the page in a fashion much like the LFr system described above. This re-ordered sequence of fragments could be written into the client web browser's cache with a new fragment table.

The fragment table system adopted into the JP family of formats from the MP4 syntax (which originated in the Apple QuickTime format) allows the fragment table to reference sequences of bytes in the current file, in external files or within Internet resources specified via a URL. This allows the client representation of a serve image, however rough, initial, incomplete, or re-ordered, to be a perfect mirror of the server image, by referencing all missing data with URLs in the fragment table. This provides a powerful means of pushing very lightweight thumbnails to a user who then has the ability to draw the complete file from the serve to any level of detail desired.

3.5 Associating metadata with a specific codestream

An important aspect of the JPM file format is the fact that the powerful metadata architecture of the JP family of file formats may be applied to each of the codestreams individually. This allows vendor-specific or industry-standard data to be associated with sub-regions of the page.

3.6 Profiles

A profiling mechanism is used to identify subsets of the general architecture for use in specific applications. The profile suited for web browsing of documents uses JPEG 2000 and JBIG 2 compression. Other profiles exist which provide compatibility with the 3-layer model found in the MRC and TIFF-FX color fax formats.

4. EXAMPLES

Figure 4 illustrates the use of a JPM file in a scenario that takes advantage of the ability to fragment a JPEG 2000 codestream and interleave fragments from multiple codestreams. In this particular example, the JPM file contains two layout objects: one for the color image and the other for text and clip art. The color image and clip art are both compressed using JPEG 2000 and the resulting codestream from each is split into two fragments. The text is compressed using a lossless, binary compression algorithm that produces a single, contiguous codestream. The six parts of the figure show the progressive rendering of objects and fragments within an object, starting with a blank frame (Figure 4a) before any coded data has been received o decompressed. The coded data in the JPM file is ordered so that the text or Mask (the mask object of the second layout object) comes first, followed by the first fragment of the Foreground (the image object of the second layout object), the first fragment of the Background (the image object of the first layout object), the second fragment of the Foreground and finally the second fragment of the Background. Figures 4b through 4f shows the image that results after each of these 5 fragments is received, decompressed and rendered.

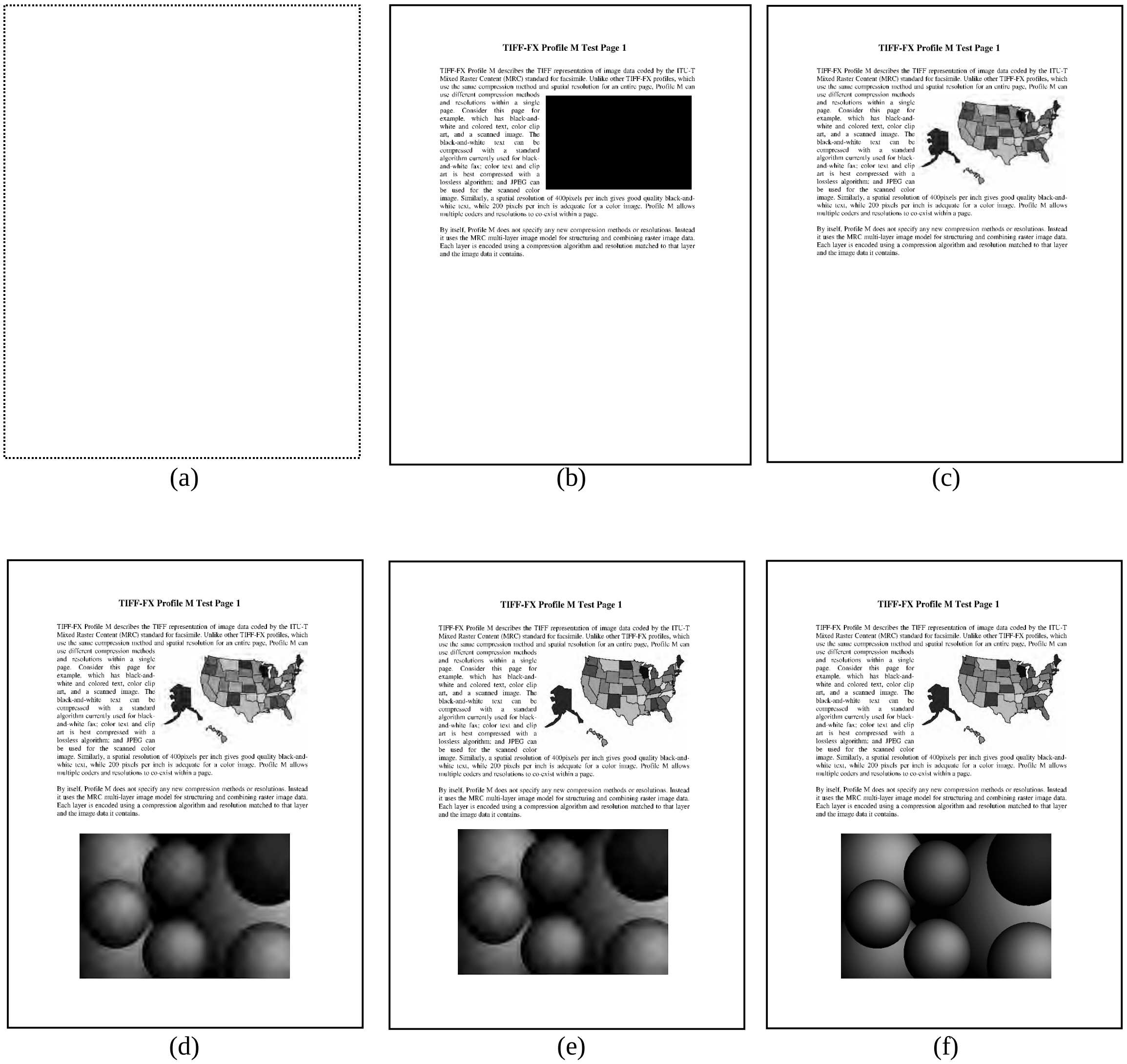


Figure 4. Progressive decompression of a JPM file with fragmented codestreams; (a) through (f) represent successive views as more packets of compressed data in a JPM file are received.

Figure 5 illustrates another example of a JPM file.

This case emphasizes the ability of the format to support high-interactivity progressive image rendering over the Internet. Th image at Figure 5(a) represents the original image. The JPM format allows an extremely small representation of the layout objects of the page to come ove first , as in Figure 5(b). No actual image data is required for this representation, but it can carry the full metadata of the server image (if desired) and contains data references to all the remaining data residing on the server for this JPM image, making it a complete stand-in representation of the server file, which conceivably could be a multi-gigabyte lossless file. The rectangles are each layout objects and can use the fixed color descriptor found in th description of the foreground image of the layout object; no actual foreground image is initially required.

In Figure 5 (c) and (d), progressive refinement of the various images and text pieces occurs in an interleaved fashion, owing to the fragmenting and interleaving of packets from the various codestreams. Rectangles are replaced with images as bandwidth allows. This permits very quick, interactive browsing of pages, with only the required data moved from server to client. Motion of the user's mouse could be used to dynamically guide the ordering of these interleaved fragments, with the fragment table being fixed-up by the web browser when the user ceases to browse the page (for example when moving to the next or previous page). Absent user mouse movements, other refinement strategies could take over to improve the page in a logical fashion. For example, larger textual elements or search term hit regions could be the first rectangles replaced b JBIG2 text images, on the theory they are more significant to navigation or determination of relevance. Refinement of adjacent pages could occur in anticipation of the user's next actions. JBIG2 would be the preferred bitonal compression method in this example. OCR metadata could be sent with the initial small representation, allowing text searching to be a component of the user's browsing. Text search hits are represented by white rectangles in Figure 5(d).

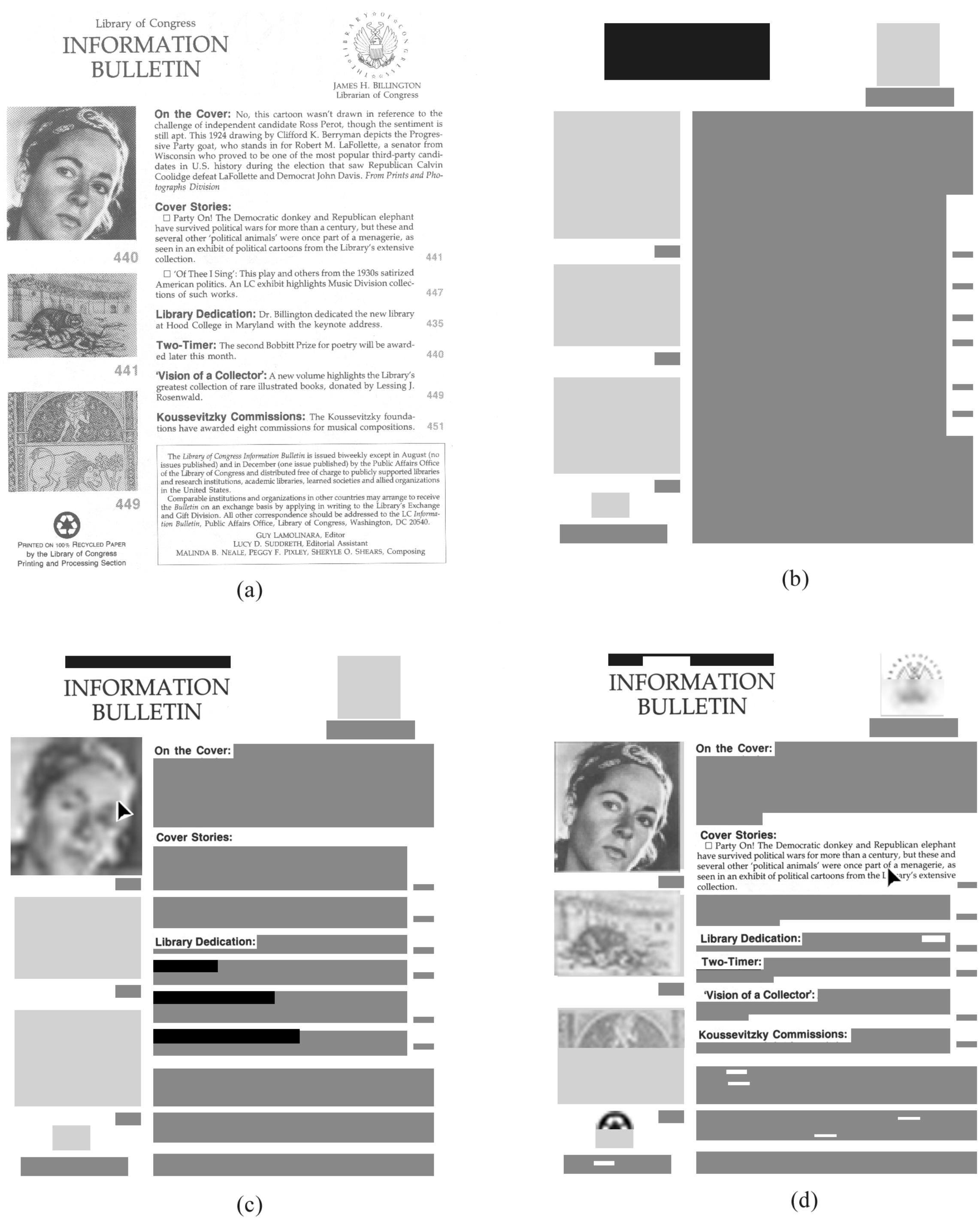


Figure 5. Progressive Internet rendering of a JPM file. (a) represents the original, high resolution file before decomposition into a JPM file. (b), (c) and (d) show progressive stages of rendering of the layout objects of a JPM file streaming from server to client. Note how the end-user client's cursor position in (c) and (d) guides which layout objects get refined next.

5. NEXT STEPS

This paper has described the JPM file format, which is at the working draft level in the JPEG 2000 standard currently under development. Part 1 of the JPEG 2000 standard specifies the minimal JPEG 2000 decoder and the optional JP2 file format. Part 1 is expected to become an International Standard in March 2001. JPM is in a following part of the standard and is expected to become an International Standard in July 2001.

A web site dedicated to providing information on the JPM file format is located at http://www.webimaging.org.

6. ACKNOWLEDGMENTS

The authors gratefully acknowledge the efforts of Rick Crowhurst of Picture Elements and Vladimir Misic of Xerox.

7. REFERENCES

1. M. W. Marcellin, M. J. Gormish, A. Bilgin, M. P. Boliek, “An overview of JPEG 2000,” Proceedings, 2000 Data Compression Conference, March 2000, Snowbird, Utah.
   (The paper is available at http://www-spacl.ece.arizona.edu/Publications/Papers/dcc2000.pdf
2. L.H. Sharpe II, B. Manns, "Document understanding using layout styles of title page images," Machine Vision Applications in Character Recognition, Donald P. D'Amato, Wolf-Ekkehard Blanz, Byron E. Dom, Sargur N. Srihari, Editors, Proc. SPIE 1661, March 1992, pp. 50-60.
3. RFC 2301, TIFF-FX File Format for Internet Facsimile
4. Antique Books project, http://www.antiquebooks.net
5. R. Buckley, D. Venable and L. McIntyre, “New developments in color facsimile and internet fax,” Proc. IS&T’s Fifth Color Imaging Conference, pp. 296-300, Scottsdale, AZ, Nov. 1997.
6. ITU-T Recommendation T.44, Mixed Raster Content, April 1999
7. Scansoft, Inc., http://www.scansoft.com
8. L. Bottou, P. Haffner, P. Howard, P. Simard, Y. Bengio and Y. LeCun, “High quality document image compression using DjVu,” Journal of Electronic Imaging, 7(3), pp. 410-425, July 1998.
9. LizardTech, Inc., http://www.lizardtech.com
10. LuraTech, Inc., http://www.luratech.com
11. Image Power, Inc., http://www.imagepower.com
12. Multiple-image Network Graphics (MNG), http://www.libpng.org/pub/mng
13. JPEG 2000 Part 2 Committee Draft, 18 August 2000.