DjVu: a Compression Method for Distributing Scanned Documents in Color over the Internet

Yann LeCun, Léon Bottou, Patrick Haffner and Paul G. Howard
AT&T Labs-Research
100 Schultz Drive
Red Bank, NJ 07701-7033
{*yann,leonb,haffner,pgh*}*@research.att.com*

¿

Abstract

We present a new image compression technique called "DjVu" that is specifically geared towards the compression of scanned documents in color at high revolution. DjVu enable any screen connected to the Internet to access and display images of scanned pages while faithfully reproducing the font, color, drawings, pictures, and paper texture. With DjVu, a typical magazine page in color at 300dpi can be compressed down to between 40 to 60 KB, approximately 5 to 10 times better than JPEG for a similar level of subjective quality. A real-time, memory efficient version of the decoder is available as a plug-in for popular web browsers.

1. Introduction

Until now, most work on document image compression has focused on black and white images (CCITT-G4, JBIG1). Little attention has been given to color document images with mixtures of text, drawings, and pictures. However, with electronic storage, retrieval, and distribution of documents becoming faster and cheaper, it is clear that the complete digitization of the world's major library collections is only a matter of time [1]. Several authors have proposed image-based approaches to digital libraries [2, 3, 4] which are restricted to black and white images, but it seems paradoxical that there exist no universal standard for efficient storage, retrieval, and transmission of high-quality document images in color.

Existing documents are usually OCRed or re-typed and converted to HTML or to Adobe's PDF format, a tedious and expensive task. Even if pictures and drawings are scanned and integrated into the web page, much of the visual aspect of the original document is likely to be lost.

A simple alternative would be to scan the original page and simply compress the image as a JPEG or GIF file. Unfortunately, those files tend to be quite large if one wants to preserve the readability of the text. Compressed with JPEG, a color image of a typical magazine page scanned at 100dpi (dots per inch) would be around 100 KB to 200 KB, and would be barely readable. The same page at 300dpi would be of acceptable quality, but would occupy around 500 KB. Even worse, not only would the decompressed image fill up the entire memory of an average PC, but only a small portion of it would be visible on the screen at once. A just-readable black and white page in GIF would be around 50 to 100 KB.

To make remote access to digital libraries a pleasant experience, pages must appear on the screen after only a few seconds delay. Assuming a 56 kilobits per second (kbps) connection, this means that the most relevant parts of the document (the text) must be compressed down to about 20 to 30 KB. With a progressive compression technique, the text would be transmitted and displayed first. Then the pictures, drawings, and backgrounds would be transmitted and displayed, improving the quality of the image as more bits arrive. The overall size of the file should be on the order of 50 to 100 KB to keep the overall transmission time and storage requirements within reasonable bounds.

Another peculiarity of document images, their large size, makes current image compression techniques inappropriate. A magazine-size page at 300 dots per inch is 3300 pixel high and 2500 pixel wide. Uncompressed, it occupies 25 MB of memory, more than what the average PC can handle. A practical document image viewer would need to keep the image in a compressed form in the memory of the machine, and only decompress on-demand the pixels that are being displayed on the screen.

2. The DjVu Compression Method

The DjVu document image compression technique[5] is an answer to all the above problems. With DjVu, scanned pages at 300dpi in full color can be compressed down to 30 to 60 KB files from 25 MB originals with excellent quality. Black and white pages typically occupy 10 to 30 KB once compressed. This puts the size of high-quality scanned pages in the same order of magnitude as an average HTML page (44 KB according to the latest statistics). DjVu pages are displayed within the browser window through a plug-in. The DjVu plug-in allows easy panning and zooming of very large images. This is made possible by an on-the-fly decompression method which allows images that would normally require 25 MB of RAM once decompressed to require only 2 MB of RAM.

The basic idea behind DjVu is to separate the text from the backgrounds and pictures and to use different techniques to compress each of those components. Traditional methods are either designed to compress natural images with few edges (JPEG), or to compress black and white document images almost entirely composed of sharp edges (CCITT G3, G4, and JBIG-1). The DjVu technique improves on both, and combines the best of both approaches. A foreground/background separation algorithm generates and encodes separately three images from which the original image can be reconstructed: the background image, the foreground image and the mask image. The first two are low-resolution color images (generally 100dpi), and the latter is a high-resolution bi-level image (300dpi). A pixel in the decoded image is constructed as follows: if the corresponding pixel in the mask image is 0, the output pixel takes the value of the corresponding pixel in the appropriately upsampled background image. If the mask pixel is 1, the pixel color is chosen as the color of the connected component (or taken from the foreground image). The background image can be encoded with a method suitable for continuous-tone images. DjVu uses a progressive, wavelet-based compression algorithm called IW44 for this purpose. IW44 is described in the next section. The mask image can be encoded with a bi-level image compression algorithm. DjVu uses a method called JB2 for this purpose. This method is a variation of AT&T's proposal to the new JBIG2 fax standard. The foreground/background representation was proposed in the ITU MRC/T.44 recommendation (Mixed Raster Content[6]).

The next section briefly describes the DjVu encoding process. The following section focuses on IW44, the wavelet-based image compression technique used in DjVu to code the foreground and background layers. Then the viewer/browser is briefly described. Finally, results of experimental comparisons are provided in Section 6.

3. The DjVu Encoder

The main ideas behind the foreground/background separation algorithm are the following. The image is partitioned into square blocks of pixels. A clustering algorithm finds the two dominant colors within each block. Then, a relaxation algorithm ensures that neighboring blocks assign similar colors to the foreground and the background. After this phase, each pixel is assigned to the foreground if its color is closer to the foreground cluster prototype than to the background cluster prototype. A subsequent phase cleans up and filters foreground components using a variety of criteria.

DjVu achieves high compression ratios by using new compression algorithms for the mask layer (JB2) as well as for the background and foreground layers (IW44). The JB2 bilevel encoding method eliminates much of the redundancy in the text image by identifying quasi identical shapes on the page, such as multiple occurrences of a character in a particular font. JB2 first codes the bitmap of each unique shape by taking advantage of the redundancy between similar shapes. It then codes the locations at which each shape appears on the page. The IW44 wavelet compressor is progressive and has the considerable advantage of being decompressable in real time. In addition, a new masking technique based on multiscale successive projections[7] is used to avoid spending bits to code areas of the background that are covered by foreground characters or drawings. Both JB2 and IW44 rely on a new type of adaptive binary arithmetic coder called the ZP-coder[8], that squeezes out any remaining redundancy within a few percent of the Shannon limit. The ZP-coder is adaptive, and faster than other approximate binary arithmetic coders.

4. Wavelet Compression of Background Images

Multi-resolution wavelet decomposition is one of the most efficient transforms for coding color images [9, 10], and is the preferred candidate for many ongoing standardizations efforts in color and gray-scale image compression. The image is first represented as a linear combination of locally supported wavelets. The image's local smoothness ensures that the distribution of the wavelet coefficients is sparse and sharply concentrated around zero. High compression efficiency is achieved using a quantization and coding scheme that takes advantage of this peaked distribution.

Because of the smoothness assumption, it is natural to use wavelet-based algorithms for encoding the image backgrounds. The digital library applications for which DjVu is designed puts extreme requirements on the technique used. The viewer (decoder) must be able to decode and display a page in at most 2 seconds to decode a page on a low-end PC. It needs to be progressive so that the image quality improves as more bits arrive. It needs to do real-time decompression from a compact data structure so as not to exceed the RAM capacity of an average PC with the fully decoded image. The background image is typically a 100dpi color image containing one to two million pixels. It may contain a nearly uniform background color, or it may contain colorful pictures and illustrations that should be displayed incrementally while the DjVu data is coming.

DjVu uses a new wavelet compression technique called IW44. IW44 computes a very fast, five stage lifting decomposition using Deslauriers-Dubuc interpolating wavelets with four analyzing moments and four vanishing moments [11]. The encoding process sorts the significant bits of the non-zero wavelet coefficients and codes their values, together with their locations by decreasing order of importance. The entropy coding of those values is performed using the ZP-coder, a new binary adaptive arithmetic coder.

During decompression, the wavelet coefficients are decoded into a compact sparse array that uses almost no memory for zero coefficients. Using this technique, the complete background image can be represented in memory using only one quarter of the memory required by the image pixels. The piece of the image that is displayed on the screen can be generated on-the-fly from this sparse array at the required zoom factor. This greatly reduces the memory requirements of the viewer.

The foreground layer is typically coded at one twelveth the original resolution (25dpi for a 300dpi original) using IW44.

5. The DjVu Browser

The digital library user experience depends critically on the performance of the browsing tools. Much more time is spent viewing documents than formulating queries. As a consequence, browsers must provide very fast response, smooth zooming and scrolling abilities, good color reproduction and sharp pictures.

These requirements put stringent constraints on the browsing software. The full resolution color image of a page requires about 25 MBytes of memory. Decompressing such images before displaying them would exceede the memory limits of average desktop computers.

We developed a plug-in for Netscape Navigator and Internet Explorer. It relies on an intermediate representation of the page image and requires less than 2 MB of main memory per page. Unlike with many document browsers, each page of a DjVu document is associated with a single URL. Behind the scenes, the plug-in implements information caching and sharing. This design allows the digital library designer to set up a navigation interface using well-known Web technologies like HTML, JavaScript, or Java. This provides more flexibility than other document browsing systems where multipage documents are treated as a single entity, and the viewer handles the navigation

3.1sec/23k: the *mask* (text, 23K) is loaded.

4.8sec/35K: The background is still blurred.

9.4sec/67K: Loading is finished.

Figure 1: Downloading through a 56K modem

JPEG at 100dpi only. DjVu.

Figure 2: Comparison of JPEG at 100dpi (left) with quality factor 30% and DjVu (right). The file sizes for the complete pages are 82 KB for JPEG and 67 KB for DjVu

between the pages. Figure 1 shows the *progressive* displaying of a document while it is downloaded through a 56K modem. First comes the text, then the background, with increasing quality as more bits arrive.

6. Experimental Results

Typical magazine or catalog pages in color at 300dpi compressed with DjVu occupy 40 to 70 KB. Black and white pages such as technical papers, are between to 15 and 40 KB. Ancient books, where most of the color is on the background, occupy 30 to 60 KB. Ancient manuscripts are around 100KB. This is 5 to 10 times better than the 500 KB that would typically be required to achieve a satisfactory level of readability using JPEG compression.

Figure 2 shows a comparison between DjVu and “JPEG, 100dpi” on a larger segment of an image. It is interesting to note that the rendering of the photographs is also better with DjVu that with JPEG, even though DjVu uses the same 100dpi for the background image.

7. conclusion

DjVu, a new compression technique for color document images is described. It fills the gap between the world of paper and the world of bits by allowing scanned document to be easily published on the Internet.

The DjVu software is available free for research, evaluation and non-commercial use at `http://djvu.research.att.com`. The DjVu plug-in is available for Linux, and other unix platforms, Windows 95, NT, and Mac. The compressor is available on various Unix platforms. The above web site also contains a digital library with over 800 pages of scanned documents from various origins.

References

[1] M. Lesk. *Practical Digital Libraries: Books, Bytes and Bucks*. Morgan Kaufmann, 1997.

[2] G. Story, L. O’Gorman, D. Fox, L. Shaper, and H Jagadish. The RightPages image-based electronic library for alerting and browsing. *IEEE Computer*, 25(9):17–26, 1992.

[3] T. Phelps and R. Wilensky. Towards active, extensible, networked documents: Multivalent architecture and applications. In *Proceedings of the 1st ACM International Conference on Digital Libraries*, pages 100–108, 1996.

[4] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Van Nostrand Reinhold, New York, 1994.

[5] L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio, and Y. LeCun. High quality document image compression with djvu. *Journal of Electronic Imaging*, 7(3):410–428, 1998.

[6] MRC. Mixed rater content (MRC) mode. ITU Recommendation T.44, 1997.

[7] L. Bottou and S. Pigeon. Lossy compression of partially masked still images. In *Proceedings of IEEE Data Compression Conference*, Snowbird, UT, March-April 1998.

[8] L. Bottou, P. G. Howard, and Y. Bengio. The Z-coder adaptive binary coder. In *Proceedings of IEEE Data Compression Conference*, pages 13–22, Snowbird, UT, 1998.

[9] E. H. Adelson, E. Simoncelli, and R. Hingorani. Orthogonal pyramid transform for image coding. In *Proc. SPIE vol 845: Visual Communication and Image Processing II.*, pages 50–58, Cambridge, MA, October 1987.

[10] J. M. Shapiro. Embedded image coding using zerotrees of wavelets coefficients. *IEEE Transactions on Signal Processing*, 41:3445–3462, December 1993.

[11] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Journal of Applied Computing and Harmonic Analysis*, 3:186–200, 1996.