NAME

DjVu - DjVu and DjVuLibre.

INTRODUCTION

Although the Internet has given us a worldwide infrastructure on which to build the universal library, much of the world knowledge, history, and literature is still trapped on paper in the basements of the world's traditional libraries. Many libraries and content owners are in the process of digitizing their collections. While many such efforts involve the painstaking process of converting paper documents to computer-friendly form, such as SGML based formats, the high cost of such conversions limits their extent. Scanning documents, and distributing the resulting images electronically is not only considerably cheaper, but also more faithful to the original document because it preserves its visual aspect.

Despite the quickly improving speed of network connections and computers, the number of scanned document images accessible on the Web today is relatively small. There are several reasons for this.

The first reason is the relatively high cost of scanning anything else but unbound sheets in black and white. This problem is slowly going away with the appearance of fast and low-cost color scanners with sheet feeders.

The second reason is that long-established image compression standards and file formats have proved inadequate for distributing scanned documents at high resolution, particularly color documents. Not only are the file sizes and download times impractical, the decoding and rendering times are also prohibitive. A typical magazine page scanned in color at 100 dpi in JPEG would typically occupy 100 KB to 200 KB , but the text would be hardly readable: insufficient for screen viewing and totally unacceptable for printing. The same page at 300 dpi would have sufficient quality for viewing and printing, but the file size would be 300 KB to 1000 KB at best, which is impractical for remote access. Another major problem is that a fully decoded 300 dpi color images of a letter-size page occupies 24 MB of memory and easily causes disk swapping.

The third reason is that digital documents are more than just a collection of individual page images. Pages in a scanned documents have a natural serial order. Special provision must be made to ensure that flipping pages be instantaneous and effortless so as to maintain a good user experience. Even more important, most existing document formats force users to download the entire document first before displaying a chosen page. However, users often want to jump to individual pages of the document without waiting for the entire document to download. Efficient browsing requires efficient random page access, fast sequential page flipping, and quick rendering. This can be achieved with a combination of advanced compression, pre-fetching, pre-decoding, caching, and progressive rendering. DjVu decomposes each page into multiple components (text, backgrounds, images, libraries of

common shapes...) that may be shared by several pages and downloaded on demand. All these requirements call for a very sophisticated but parsimonious control mechanism to handle on-demand downloading, pre-fetching, decoding, caching, and progressive rendering of the page images. What is being considered here is not just a document image compression technique, but a whole platform for document delivery.

DjVu is an image compression technique, a document format, and a software platform for delivering documents images over the Internet that fulfills the above requirements.

DJVU IMAGE COMPRESSION

The DjVu image compression is based on three technologies:

DjVuPhoto

DjVuPhoto, also known as IW44, is a wavelet-based continuous-tone image compression technique with progressive decoding/rendering. It is best used for encoding photographic images in colors or in shades of gray. Images are typically half the size as JPEG for the same distortion.

DjVuBitonal

DjVuBitonal, also known as JB2, is a bitonal image compression that takes advantage of repetitions of nearly identical shapes on the page (such as characters) to efficiently compress text images. It is best used to compress black and white images representing text and simple drawings. A typical 300 dpi page in DjVuBitonal occupies 5 to 25 KB (3 to 8 times better than TIFF-G4 or PDF).

DjVuDocument

DjVuDocument is a compression technique specifically designed for color digital documents images containing both pictures and text, such as a page of a magazine. DjVuDocument represents images into separately compressed layers. The foreground layer is usually compressed with DjVu Bitonal and contains the text and drawings. The background layer is usually compressed with DjVuPhoto and contains the background texture and the pictures at lower resolution.

DJVU DOCUMENT DELIVERY PLATFORM

The DjVu technology is designed from the ground up to support the efficient delivery of digital documents over the Internet. It provides various ways to deal with multi-page documents, and various ways to enrich the content with hyper-links, meta-data, searchable text, etc.

MIME types

The DjVu format has an official MIME type of **image/vnd.djvu**, which is the preferred content-type to

be given by http servers for DjVu files. Unofficial mime types used historically are **image/x.djvu** and **image/x-djvu**, which may still be encountered. Ideally, clients should be configured to handle all three.

Bundled multi-page documents

Bundled multi-page DjVu document uses a single file to represent the entire document. This single file contains all the pages as well as ancillary information (e.g. the page directory, data shared by several pages, thumbnails, etc.). Using a single file format is very convenient for storing documents or for sending email attachments.

When you type the URL of a multi-page document, the DjVu browser plugin starts downloading the whole file, but displays the first page as soon as it is available. You can immediately navigate to other pages using the DjVu toolbar. Suppose however that the document is stored on a remote web server. You can easily access the first page and see that this is not the document you wanted. Although you will never display the other pages the browser is transferring data for these pages and is wasting the bandwidth of your server (and the bandwidth of the Internet too). You could also see the summary of the document on the first page and jump to page 100. But page 100 cannot be displayed until data for pages 1 to 99 has been received. You may have to wait for the transmission of unnecessary page data. This second problem (the unnecessary wait) can be solved using the ‘‘byte serving’’ options of the HTTP/1.1 protocol. This option has to be supported by the web server, the proxies, the caches and the browser. Byte serving however does not solve the first problem (the waste of bandwidth).

Indirect multi-page documents

Indirect multi-page DjVu documents solve both problems. An indirect multi-page DjVu document is composed of several files. The main file is named the index file. You can browse a document using the URL of the index file, just like you do with a bundled multi-page document. The index file however is very small. It simply contains the document directory and the URLs of secondary files containing the page data. When you browse an indirect multi-page document, the browser only accesses data for the pages you are viewing. This can be done at a reasonable speed because the browser maintains a cache of pages and sometimes pre-fetches a few pages ahead of the current page. This model uses the web serving bandwidth much more effectively. It also eliminates unnecessary delays when jumping ahead to pages located anywhere in a long document.

Annotations

Every DjVu image optionally includes so-called annotation chunks. The annotation chunk is often used to define hyper-links to other document pages or to arbitrary web pages. Annotation chunks can also be used for other purposes such as setting the initial viewing mode of a page, defining highlighted zones, or storing arbitrary meta-data about the page or the document.

Hidden text

Every DjVu image optionally includes a hidden text layer that associated graphical features with the corresponding text. The hidden text layer is usually generated by running an Optical Character Recognition software. This textual information provides for indexing DjVu documents and copying/pasting text from DjVu page images.

Thumbnails

DjVu documents sometimes contain pre-computed page thumbnails.

Outline

DjVu documents sometimes contain a navigation chunk containing an outline, that is, a hierarchical table of contents with pointers to the corresponding document pages.

DJVUZONE AND DJVULIBRE

The DjVu technology was initially created by a few researchers in AT&T Labs between 1995 and 1999. Lizardtech, Inc. then obtained a commercial license from AT&T and continued the development. The current owner of the DjVu commercial rights is Cuminas (**https://www.cuminas.jp/en/about_djvu**), offers solutions for producing and distributing documents using the DjVu technology, as well as a DjVu viewer packaged as a Chrome extension.

The DjVu.org web site (**http://www.djvu.org**) is managed by the few AT&T Labs researchers who created the DjVu technology in the first place. We promote the DjVu technology by providing an independent source of information about DjVu.

Understanding how little room there is for a proprietary document format, Lizardtech released the DjVu Reference Library under the GNU Public License in December 2000. This library entirely defines the compression format and the elementary codecs. Six month later, Lizardtech released an updated DjVu Reference Library as well as the source code of the Unix viewer.

These two releases form the basis of our initial DjVuLibre software. We modified the build system to comply with the expectations of the open source community. Various bugs and portability issues have been fixed. We also tried to make it simpler to use and install, while preserving the essential structure of the Lizardtech releases.

The DjVuLibre software contains the following components:

bzz(1)

A general purpose compression command line program. Many internal DjVu data structures are compressed using this technique.

c44(1)
A DjVuPhoto command line encoder. This state-of-the-art wavelet compressor produces DjVuPhoto images from PPM or JPEG images.

cjb2(1)
A DjVuBitonal command line encoder. This soft-pattern-matching compressor produces DjVuBitonal images from PBM images. It can encode images without loss, or introduce small changes in order to improve the compression ratio. The lossless encoding mode is competitive with that of the Lizardtech commercial encoders.

cpaldjvu(1)
A DjVuDocument command line encoder for images with few colors. This encoder is well suited to compressing images with a small number of distinct colors (e.g. screen-shots). The dominant color is encoded by the background layer. The other colors are encoded by the foreground layer.

csepdjvu(1)
A DjVuDocument command line encoder for separated images. This encoder takes a file containing pre-segmented foreground and background images and produces a DjVuDocument image.

ddjvu(1)
A command line decoder for DjVu images. This program produces a PNM image representing any segment of any page of a DjVu document at any resolution.

djview(1)
A stand-alone viewer for DjVu images. This sophisticated viewer displays DjVu documents. It implements document navigation as well as fast zooming and panning.

nsdejavu(1)
A web browser plugin for viewing DjVu images. This small plugin allows for viewing DjVu documents from web browsers. It internally uses djview to perform the actual work.

djvups(1)
A command line tool for converting DjVu documents into PostScript .

djvm(1)
A command line tool for manipulating bundled multi-page DjVu documents. This program is often used to collect individual pages and produce a bundled document.

djvmcvt(1)

A command line tool for converting bundled documents to indirect documents and conversely.

djvused(1)
A powerful command line tool for manipulating multi-page documents, creating or editing annotation chunks, creating or editing hidden text layers, pre-computing thumbnail images, and more...

djvutxt(1)
A command line tool to extract the hidden text from DjVu documents.

djvudump(1)
A command line tool for inspecting DjVu files and displaying their internal structure.

djvuextract(1)
A command line tool for dis-assembling DjVu image files.

djvumake(1)
A command line tool for assembling DjVu image files.

djvuserve(1)
A CGI program for generating indirect multi-page DjVu documents on the fly.

djvutoxml(1), **djvuxmlparser**(1)
Command line tools to edit DjVu metadata as XML files.

DJVU ENCODERS AND ANY2DJVU

DjVuLibre comes with a variety of specialized encoders, **c44**(1) for photographic images, **cjb2**(1) for bitonal images, and **cpaldjvu**(1) for images with few distinct colors. Although these encoders perform well in their specialized domain, they cannot handle complex tasks involving segmentation and multipage encoding.

The Lizardtech commercial products (see **http://www.lizardtech.com/solutions/document**) can perform these complex encoding tasks

Another solution is provided by the compression server at (**http://any2djvu.djvu.org**). This machine uses pre-lizardtech prototype encoders from AT&T Labs and performs almost as well as the commercial Lizardtech encoders. Please note that the Any2DjVu compression server comes with no guarantee, that nothing is done to ensure that your documents will remain confidential, and that there is

only one computer working for the whole planet.

CREDITS

Numerous people have contributed to the DjVu source code during the last five years. Please submit a sourceforge bug report to update the following list.

Yoshua Bengio, Lèon Bottou, Chakradhar Chandaluri, Regis M. Chaplin, Ming Chen, Parag Deshmukh, Royce Edwards, Andrew Erofeev, Praveen Guduru, Patrick Haffner, Paul G. Howard, Orlando Keise, Yann Le Cun, Artem Mikheev, Florin Nicsa, Joseph M. Orost, Steven Pigeon, Bill Riemers, Patrice Simard, Jeffery Triggs, Luc Vincent, Pascal Vincent.