Faculty Publications

2003-08-01

Just-In-Time Browsing for Digitized Microfilm and Other Similar Image Collections

William A. Barrett
william_barrett@byu.edu

Douglas J. Kennard

Original Publication Citation

BYU ScholarsArchive Citation

Just-In-Time Browsing for Digitized Microfilm and Other Similar Image Collections

Douglas J. Kennard
kennard@cs.byu.edu

William A. Barrett
barrett@cs.byu.edu

Brigham Young University
Computer Science Department
3361 TMCB, BYU, Provo, Utah, USA

Abstract

This paper describes "Just-In-Time Browsing" (JITB), a method for image browsing (at modem-like speed) in which image data is transmitted and presented to the user progressively, in prioritized order, based on image content and user interaction. Spatial resolution and grayscale or color fidelity is increased first for the portions of the image that are immediately of most interest to the user. JITB is specifically geared toward digitized microfilm and other similar document image collections, although it can also be used for other types of images. A series of common browsing tasks performed by multiple users demonstrates that JITB compares favorably with two other methods that are currently used for browsing images on the Internet.

1. Introduction

There are many cases in which we wish to be able to quickly browse through collections of large images over a low-bandwidth medium such as a modem. For example, in family history and genealogical research, the ability to browse unindexed microfilm images from home, without a high bandwidth connection to the Internet, is very desirable.

One of the major problems with providing digital microfilm over the Internet is that the images are often very large (potentially, up to about 6,000 x 4,000 pixels grayscale) to provide the resolution necessary for reading the small print. Large images require a great deal of time to download, especially over a modem, making a visual search through numerous microfilm images an impractical task.

When we browse a collection of images, we want to glean the image data that is of interest to us without spending valuable time downloading all of the unwanted detail. It is important to note the distinction between what this pa-

per refers to as *browsing*, as opposed to *viewing* images in a collection. When a person *views* an image, the goal is to present that image to the person in a form that is as visually similar to the original as possible, although some trade-off is often made between the quality of the image and its file size in order to access the image more efficiently. When a person *browses* images, on the other hand, the main goal, initially, is to allow the person to determine the content of each image as soon as possible so the person can decide whether or not it is of interest. If the image is of interest, a more detailed examination can follow, but if it is not of interest, the person can immediately move on to the next image.

Most commonly used techniques for presenting image content on the Internet focus on giving the viewer a representation of the original image that is as visually accurate as possible, whereas for many applications, browsing speed is much more important than complete visual fidelity.

This paper describes a method called "Just-In-Time Browsing" (JITB), developed and used for browsing collections of large images, at modem-like speeds, but at interactive rates. Although our method can be used for various types of images, including color images, the main focus of JITB is to provide more efficient browsing of digital microfilm and other similar types of document image collections.

In addition to describing the JITB system, this paper also compares the system's performance to that of two methods (DjVu [1] and MrSID [4])that are currently used for providing access to digital microfilm images over the Internet. For certain routine tasks that would be typical in many cases when browsing digital microfilm, JITB is shown to be significantly faster. While maintaining this advantage, JITB is still shown to be at least comparable in performance to the commercial systems for most other types of browsing tasks.

2. The Just-In-Time Browsing system

The JITB system keeps the goals of image *browsing* in mind, instead of simultaneously trying to cater to the goals of image *viewing*. Spatial context of images is provided almost instantaneously by means of pixel replicating using a low-resolution thumbnail. More detailed image data (both spatial resolution detail and grayscale or color information) arrives "just in time," or as the user needs it.

The JITB system consists of four main parts, as illustrated in Figure 1. Image encoding for the JITB system (Section 2.1) takes place as a preprocessing step, in which
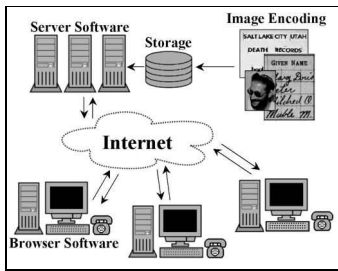

Figure 1. The JITB system at a glance.

the original images are scaled, tiled, encoded, and compressed. Once the images are encoded in the preprocessing portion of the system, they are stored, along with metadata, in a format that allows for easy random access to the various resolution layers and tiles of the images. The JITB server software (Section 2.3) provides online access to the stored images, which are then browsed over the Internet using the JITB browser software (Section 2.2).

2.1. Image encoding and storage

In the JITB system, source images are scaled to create an image pyramid, of sorts, for hierarchical browsing, as visualized in Figure 2. Each resolution layer of the pyramid
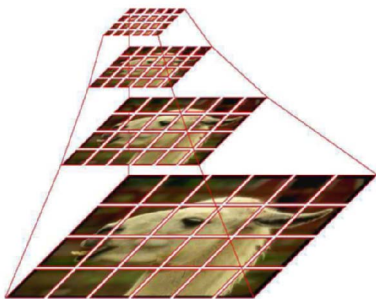

Figure 2. Multiple layers of a scaled and tiled image, forming an image pyramid.

is split into rows and columns of tiles, which are then compressed. The compressed tile image data, along with pertinent metadata, is stored in a file format developed for the JITB system, allowing random access to any tile of the image at any layer of the pyramid. In addition, *default browsing templates* can be (manually) created using *a priori* knowledge about images to specify regions of interest that should be given higher priority, by default.

Within this framework, just about any compression method could be applied to each individual tile, although only a few (JPEG, for example) are currently supported. The JITB system goes beyond simply compressing tiles, however, and actually permits various amounts of grayscale or color data to be provided to the user, as needed. This is accomplished by using the idea of *bitplane encoding* [5, 7], in which the most significant bit of each pixel in the image (the most significant bitplane) is transmitted first to provide a bitonal approximation of the image, and then each successive bitplane doubles the number of grayscale levels in the image. Each bitplane is then independently compressed using a bitonal compression scheme, such as JBIG [2]. We have seen that using JBIG compression and bitplane encoding generally results in less data than using full grayscale or color compression, such as PNG, JPEG, or JPEG2000, as long as only the first one to two bitplanes are needed. As can be seen in Figure 3, one or two bitplanes are often sufficient for browsing microfilm.
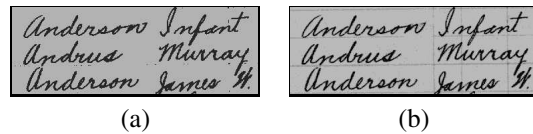

(a) (b)

Figure 3. Minimal grayscale information is often sufficient for browsing document images, as seen in this cropped region of an image that has been bitplane encoded using the method described in this paper. (a) One bitplane of three available (b) Two bitplanes

We have extended the idea of progressive bitplane encoding so that it can be used with both grayscale and color images in JITB. The image at each layer of the pyramid is quantized to a maximum of 2^p colors (or grayscale values), where p is the number of bitplanes to be made available. The palette is then ordered such that similar colors are adjacent to each other. For grayscale palettes, this is trivial, as the grayscale values are just sorted in increasing order. For color images, one of the palette sorting methods described in [3] or [6] can be used.

Next, an *encoded value* is calculated for each palette color, which can be thought of as "directions" to the color's index within the palette if the color needed to be found with

a binary search, as shown in Figure 4. After one bitplane of the encoded image is transmitted, only the most significant bit of each pixel's encoded value is known. If the first bit of a specific pixel is a zero, then the representative color for that pixel is on the left half of the palette. If the first bit is a one, then it is on the right half of the palette. With each successive bit of the pixel's encoded value, the position within the palette is narrowed down more, until the exact palette index is known when all bits have arrived.
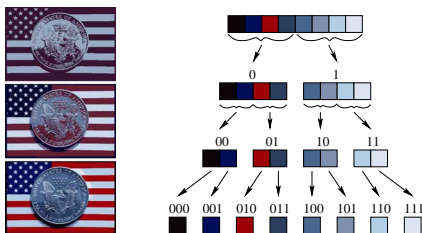
Figure 4. Progressive bitplane encoding: 1, 2, and 3 bits, top to bottom (left). Encoded values determined by palette position (right).

Next, palettes are chosen that will be used to represent the image as each successive bitplane becomes available to the user. Since the colors of the palette were ordered, similar colors are grouped within the palette, so that the colors having the same encoded value after any given number of bitplanes arrive are fairly similar. The color chosen to represent each grouping of colors is chosen to be an average of the colors of all pixels that, when quantized, map to any color within that grouping, as shown in Figure 5.
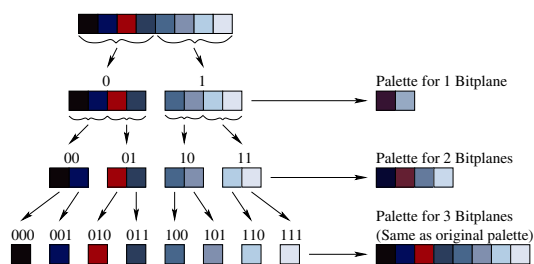
Figure 5. Palettes for use with each successive number of bitplanes, created by averaging the pixel colors of all pixels that map to each group of palette colors.

Finally, each pixel in the image is mapped to the encoded value determined for its palette color, resulting in an encoded image that can be transmitted progressively, one bitplane at a time. It is this image that is split into bitplanes and compressed with a bitonal compression method, currently JBIG.

2.2. Browser

The JITB browser (Figure 6) allows users to navigate digitized microfilm (or collections of images) by providing quick access to thumbnail previews of the images, and allowing users to inspect any image of interest with as little or as much resolution and color/grayscale detail as needed (within the limits imposed during the encoding process).

The order in which the various metadata, preview data, and detailed image data is requested from the server is based on the current needs of the user. These needs are determined using default browsing templates (Section 2.1), heuristic algorithms, and explicit user interaction, allowing the browser to attach a priority value to each request it makes of the server.

In general, image tiles are assigned their priorities based on the information provided in the default browsing template, if one is associated with the image, and also taking into account their position within the image pyramid. Lower resolution versions of the tile image are transmitted (each with only one bitplane) before higher resolutions, causing a progressive refinement of the spatial data, which allows the user to begin to get an idea of the image contents without waiting for the full resolution data to arrive. If the image layer being browsed is larger than the view window of the browser, then those tiles which are visible are given higher priority. Likewise, thumbnails that are not within the viewable area of the preview window are given lower priority than thumbnails that are visible.
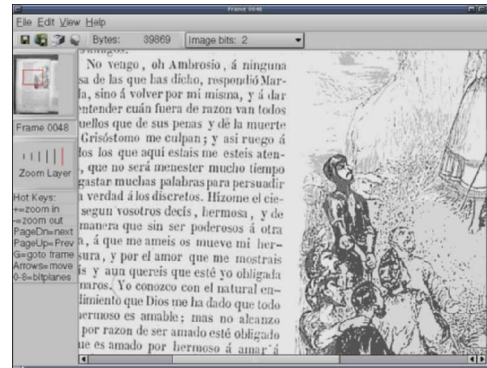
While these heuristics (based on the current view of the user, position in the image hierarchy, and default browsing templates) give a fairly good default order in which to request data from the server, the user can explicitly request regions of interest at any time by using the mouse to rubber-band the desired area. Explicit user requests always take precedence over the default priorities.

In addition, JITB provides an "Intelligent Template," activated at the touch of a button, that records the most recent explicit requests made by the user, and automatically applies those requests to the other images as the user moves from frame to frame in the microfilm. In situations where the user is viewing many frames that all have the same layout (such as census records), this feature can be very helpful.

In short, the browser requests information from the server, decides how that information should be prioritized, interprets the responses from the server, decompresses the image data, and provides the user with a visual representation of the image. Low resolution versions of images are provided early, giving spatial context to the user, and higher resolution versions, with increased color or grayscale data, are provided progressively as the user needs or wants them.
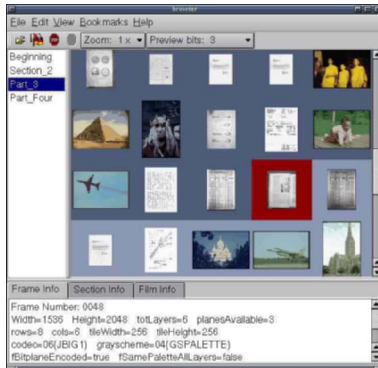
Preview Window for browsing/selecting thumbnails   Frame view window for more detailed inspection

Figure 6. The JITB Browser Interface.

2.3. Server

The server responds to requests made by the browser for metadata, palette data, and compressed image data, sending the responses in order based upon the priorities assigned by the browser.

To keep the system interactive, the amount of response data sent to the browser (per unit time) is limited to the bandwidth available on the user's end of the connection. Therefore, once a request is received by the server, it must "wait its turn" before actually being sent. In some cases, the browser needs the the priority of a request to be changed while it is waiting, for example, if an image tile was requested at its low default priority, and then the user made an explicit request for it with the mouse. The server, when signaled by the browser to do so, handles changing the priority ordering of requests "on the fly."

In some cases, the server is in the process of transmitting a relatively large chunk of data when a higher priority requests arrives. The JITB server can postpone large responses part way through, service higher priority requests, and then continue sending the original request from where it left off. The server also handles complete cancellation of requests if they are no longer desired, for example, when the user decides to move on to a different image and any image data previously requested is no longer needed.

3. Results

The JITB system performance was tested in a lab environment, using software to limit the bandwidth between the server and browser to a speed fairly comparable to a 56-Kbit modem in order to simulate a low-bandwidth connection. Overall, the prototype implementation of the JITB system works well. Even at modem-like speeds, collections of large images, such as digitized microfilm, can be browsed at a reasonable rate. Users can scan through thumbnail ver-

sions of images to look for those that might be of interest. Any image that seems to be of interest can be perused more carefully at a variety of higher resolutions. Image data is transmitted progressively in prioritized order, allowing the user to preempt the default ordering and interactively request regions of the image that are of particular interest.

When compared with two popular methods for distributing large images over the Internet (DjVu and MrSID), JITB performs comparably for most types of browsing tasks, and in some cases, JITB is many times faster than either of the other two methods. JITB excels at tasks that require the user to read fine details from large images, especially if details must be viewed in several images that all have a similar spatial layout, as is the case with census images.

In a controlled user study (described in more detail in [3]), users were timed as they performed a variety of browsing tasks. The times taken by users to accomplish each task when using JITB, DjVu, and MrSID are shown graphically in Figure 7, along with the mean time taken for each task while using each method. The times (in seconds) are also listed in Table 1.

In almost every case, JITB performed at least slightly better (on average) than either of the other methods. Overall, completing the tasks using DjVu and MrSID both took more than twice as long (on average) than when using JITB.

4. Conclusion

Unlike viewers, which are designed for full resolution acquisition of digital documents, JITB provides a scalable alternative for browsing images over the Internet. Portions of documents can be retrieved on demand at interactive rates, under user control and with a user-selectable level of detail. Alternatively, document components can be pre-assigned a priority using a default browsing template. With scalable variation in bit-depth and color, essential information can be browsed and extracted "just in time."
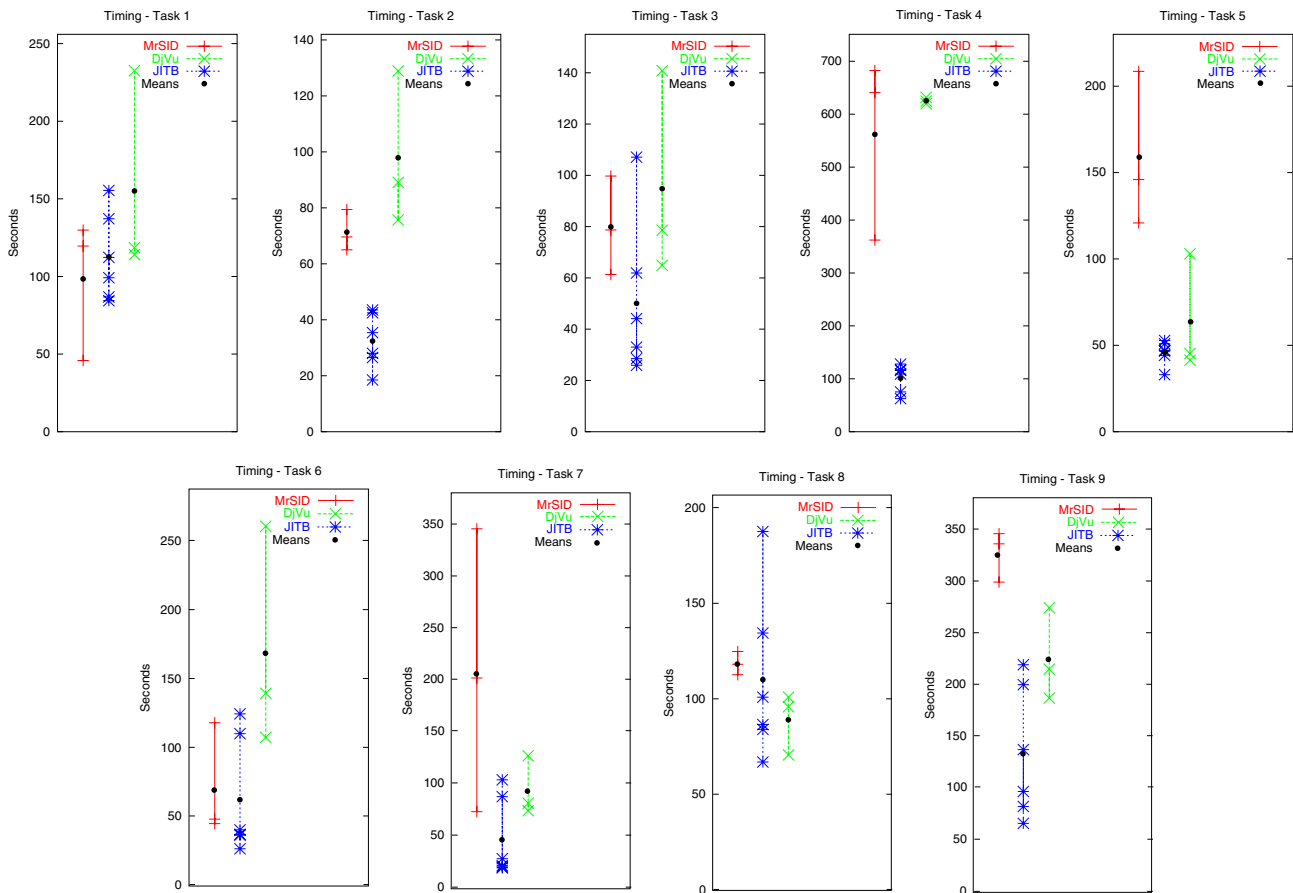
Figure 7. Comparison of times (in seconds) for task completion using each browsing method.

References

[1] L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio, and Y. LeCun. High quality document image compression with DjVu. From URL http://www.djvu.att.com/djvu/djvu-tech/, visited on 16 October 1999.

[2] ITU-T Recommendation T.82. Information technology - coded representation of picture and audio information - progressive bi-level image compression, 1993.

[3] D. Kennard. Just-in-time browsing for digital images. Master's thesis, Brigham Young University, 2003.

[4] MrSID knows the internet. *Dateline: Los Alamos*, pages 7–9, Oct 1997. Edited by M. Coonley.

[5] M. Rabbani and P. W. Jones. *Digital Image Compression Techniques*. SPIE Optical Engineering Press, Bellingham, WA, 1991.

[6] U. Rauschenbach. Compression of palettized images with progressive coding of the color information. In *SPIE Visual Communications and Image Processing (VCIP2000), Perth, Australia*, June 2000.

[7] K.-H. Tzou. Progressive image transmission: a review and comparison of techniques. *Optical Engineering*, 26(7):581–589, July 1987.

User	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9
1	84.4	42.5	28.5	115.9	33.0	44.9	73.8	125.4	345.0
2	155.4	43.5	44.1	127.8	50.9	48.1	345.3	113.2	335.2
3	137.2	28.0	25.9	117.0	46.7	117.8	202.1	118.7	298.4
4	129.9	79.4	78.7	640.9	146.1	125.1	29.0	101.8	220.1
5	119.6	65.0	99.7	682.7	208.7	110.8	88.6	188.6	200.9
6	45.9	69.6	61.3	362.1	121.0	38.0	22.4	135.4	83.6
7	99.2	18.5	33.0	62.6	44.2	*139.6*	*127.1*	*71.4*	*274.3*
8	112.3	35.4	107.1	75.3	47.1	*260.4*	*74.4*	*96.6*	*187.4*
9	86.8	26.5	61.9	108.9	52.8	*107.8*	*81.6*	*101.7*	*215.3*
10	*118.5*	*128.9*	*64.9*	*631.9*	*45.4*	41.2	21.2	84.9	98.0
11	*114.1*	*75.7*	*140.8*	*624.7*	*103.1*	37.4	20.2	87.4	138.4
12	*232.6*	*89.1*	*78.6*	*619.9*	*41.5*	27.5	104.7	67.8	67.3
	Gray=MrSID			*Italic=DjVu*			Black=JITB		

Table 1. Seconds to complete each task.

IEEE
COMPUTER
SOCIETY