Usage Patterns of a Web-Based Image Collection

Nisha Talagala, Satoshi Asami, David Patterson
Computer Science Division
University of California at Berkeley
Berkeley, California

Abstract

This paper presents a study of user access patterns to a large, web-based, image collection. The images are the entire collection of the Fine Arts Museums of San Francisco, the largest on-line collection of high resolution art images in the world. The images are served using a tile-based solution that allows a user to zoom-in and navigate within an image. We studied five months of web log data for this collection. Our analysis revealed the following: less than 10% of all available documents on the site were accessed in the five month period and document popularity appears to follow a Zipf distribution. Also, images have interesting areas which are viewed more than others, some image resolutions are viewed far more than others, and user navigation patterns vary between resolutions and are sensitive to download time. The paper discusses these results and their implications for the design of caches and archival storage systems to support this type of workload.

1.0 Introduction

In the past decade, the popularity of the World Wide Web has grown exponentially. By the year 2000, the number of hosts on the web is expected to pass a hundred million [1]. This explosive growth has to led to many studies of web site content and access patterns [2,3,4,5]. However, most of these studies have focused on HTTP logs from more traditional web sites, sites with small amounts of storage and relatively small files.

At the same time, institutions with large archives of documents (for example, museums and libraries) are beginning to digitize their holdings and put them on their web sites [6]. These web accessible digital libraries are becoming more and more common. These sites differ from traditional web sites in several ways. First, they will contain tens to hundreds of thousands of files, the entire content of an institution's archive. Second, these files will not be simple HTML documents, they could be images, video, or other multimedia files. Such files are large by web server standards; over 1MB [6], compared to files on traditional web servers that are between 256 and 512 bytes [7]. Third, since the site will contain hundreds of thousands of documents, access to most documents will have to be through a search engine and not through a sequence of HTTP links.

Putting an archive of documents on the web raises several problems. First, low internet bandwidth makes it impossible to access high quality multimedia interactively. For instance, downloading a 1MB document over a 28.8K modem takes about 5 minutes. Also, copyright considerations may prevent a site from making high quality copies available for download. For images, both of these problems can be addressed with a tile-based solution. In other words, images are available as a set of tiles in several resolutions. The user can zoom in to view parts of an image in more detail. This way, users are allowed to see details without ever being allowed to download a high quality version of the whole image. This technique has been standardized as the FlashPix format [8] and is becoming popular for high quality images.

In this paper, we present a study of user access patterns to a web-accessible image collection. Our image collection now contains over 75,000 high resolution images of art work [9], the entire collection of the Fine Arts Museums of San Francisco. At 75,000 images, it is by far the largest art image database in the world. In second place is the National Archive, with 3500 images. The images are available through the Museum's web site at *http://www.thinker.org/*. Access to the images is through a keyword search. Each image is available in eight resolutions, from 12.5% to 1600%. The images are stored and served as a sequence of tiles in each resolution.

Our results are based on HTTP server logs for this site over a five month period; March - July 1998. During this period, the image count on the site increased form 20,000 on March 1st, to 59,000 at the end of July. In this paper, we describe the access patterns observed through our logs, and draw conclusions about the system architectures and caching policies that would work well

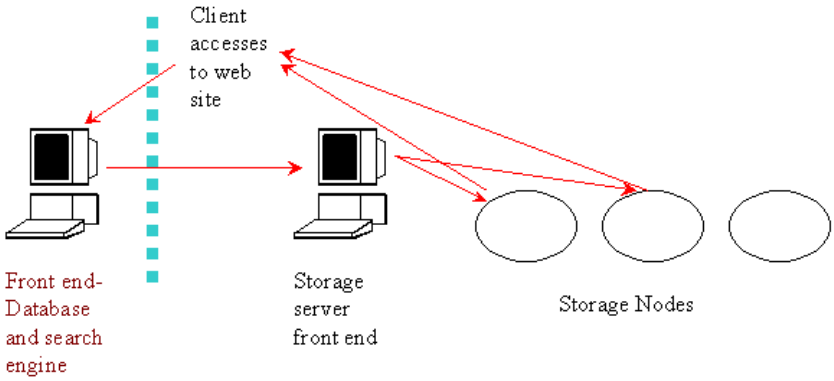


Figure 1: How the site works. Clients access images through a keyword search at the front end. Once an image is selected, tiles are transferred to the client from a storage server that is holding the image. The storage servers are managed by a storage front end. The vertical line separating the database and storage system indicates that these two servers are in different geographical locations.

for this type of workload. To our knowledge, we present the first study of user access patterns for a large collection of such tiled images.

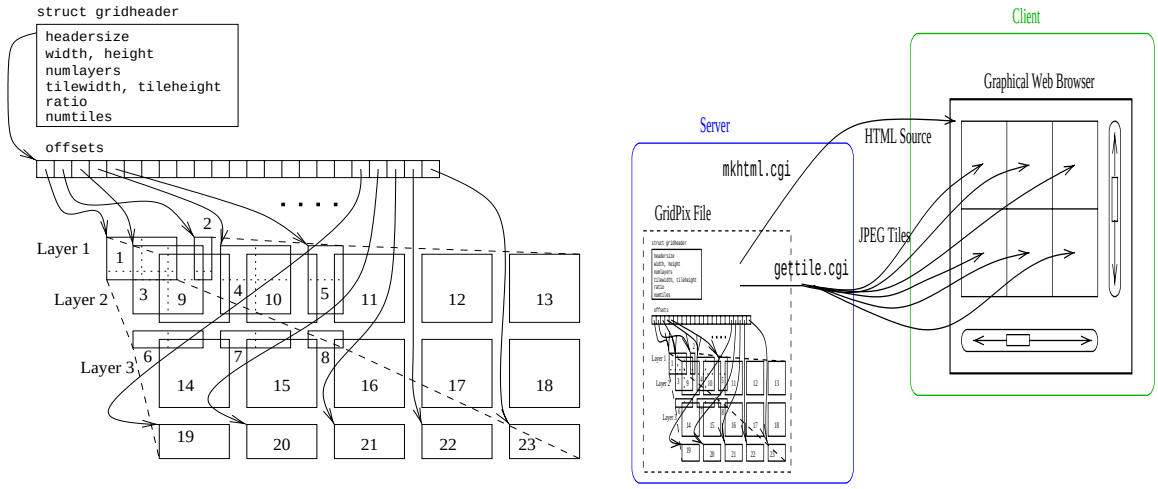
Section 2 describes our system, our tiled image format and our user interface. Section 3 describes our http logs and how they were processed. Section 4 presents the results that we have obtained from these logs. Section 5 discusses the implications of our results on the design of web servers for this style of workload. Section 6 describes related work in this area and Section 7 concludes with a summary and possible future work.

2.0 Our System

Our web service, called The Zoom Project, has been available to users since March 1st 1998. This site provides a database of high resolution images of art work. Figure 1 shows how the site works. The front end contains a search engine. Each image is searchable by title, artist, time period, and other descriptive keywords. This database contains only the image attributes, not the images themselves. Each image is identified by a 16 digit key. When a user wishes to view an image, the image is requested from the storage servers using this key. This request is passed down to the storage front end, which forwards the request to a storage server with a copy of the image. From then on, all image data transfer occurs between the storage server and the client. The storage servers are PCs with 8GB, 7200RPM SCSI drives. We do not describe the storage system in detail here; a description is available in [10]. Note: We have two front ends because the database and the storage server portions of the site are at different geographical locations. The second front end eases management by creating a level of indirection between the search engine and the storage servers. This way, we are able to reconfigure the storage system component without modifying the search database.

Each image is available at resolutions of up to 3072x2048 pixels. Our tiled image format, called GridPix [11], is similar in concept to the FlashPix standard which is becoming popular for presenting high resolution images over the web [8]. Figure 2(a) shows the GridPix format; a GridPix file contains a 2KB header structure, a variable sized index of offsets, and a sequence of tiles in resolutions from 12.5% to 100%. Tiles for resolutions higher than 100% are generated on the fly from the 100% resolution tiles. The index of offsets marks the positions in the file where various tiles begin. Each tile contains 144x144 pixels and is individually compressed in JPEG. The average tile size is 2.9KB, and an average GridPix file is about 1.2MB.

Figure 2(b) shows our user interface. We describe the user interface in detail because it is important to know what the interface looks like before discussing usage patterns. Our interface is entirely HTML based. A CGI script (`mkhtml.cgi`) creates a graphical viewer with zoom and panning controls. The user can choose the size of the viewer; five sizes are available, 14', 15', 17', 20', and 24'. At any time, only the area of the image that fits within the viewer will be transferred to the client. Each tile is retrieved separately by a second CGI script



Figures 2(a) and 2(b): Figure 2(a) shows the structure of a GridPix file. Figure 2(b) shows how image tiles are displayed in a graphical browser created using HTML

Figures 2(c) and 2(d): The GridPix viewer. Figure 2(c) shows a Picasso art work at 12.5% resolution within the GridPix viewer. Figure 2(d) shows the artist's signature at the lower right hand corner of the same image, at 50% resolution. Since the entire image appears in the viewer at 12.5% resolution, no scrollbars are displayed. Since JPEG is grid-based, tiles can be placed side by side in an HTML file and appear as if it is a single image.

```
saffron.cs.berkeley.edu - - [11/Apr/1998:21:17:33 -0700] "GET /cgi-bin/
mkhtml.cgi?31542013070000028&2&432&288&0&0 HTTP/1.0" 200 2994
saffron.cs.berkeley.edu - - [11/Apr/1998:21:17:34 -0700] "GET /cgi-bin/
gettile.cgi?31542013070000028&7&0 HTTP/1.0" 200 2990
```

Figure 3: Sample HTTP log showing `mkhtml` and `gettile` calls

(`gettile.cgi`). All images are initially displayed at 12.5% resolution. At this size, most images fit entirely within the window in all viewer sizes. Once the image becomes too large for the viewing area, the user can scroll up/down or left/right by clicking on the scrollbars that appear at the bottom and right of the image window. When a user zooms-in or scrolls, the necessary tiles are extracted from the GridPix file and sent to the user. Since each tile arrives at the browser as a separate JPEG image, the browser can cache tiles and the storage servers don't have to send the same tiles again and again. Figures 2(c) and 2(d) show the GridPix viewer in action. More information about the implementation of GridPix and the GridPix viewer is available in [11].

3.0 Logs and Log Analysis

In this section we describe our logs and how we visualized the information contained in them. For the most part, this section lists our assumptions and defines the terms we use to quantify and describe our results.

Figure 3 shows two lines of our HTTP logs. The first several fields (client name, date, time) are the same as what is found in most HTTP logs. While we use this data in our analysis, most of the information that is specific to our workload is embedded in the GET messages. Since our user interface is completely HTML based, image specific data (such as the image key, resolution level, navigation info, etc.) are embedded as arguments in the CGI scripts called by GET.

Two CGI scripts are central to our interface implementation; they are the `mkhtml.cgi` and `gettile.cgi` calls. When a user clicks a zoom or a scroll button, a new HTML page is created that shows the new state of the viewer. This page is constructed by `mkhtml.cgi`. Each tile is retrieved by calling `gettile.cgi`. In addition, several other images (the zoom icons, scrollbar icons etc.) are also retrieved as needed.

As Figure 3 shows, `mkhtml.cgi` has six arguments. The first argument is the 16 digit image key. The second is the zoom level of the image; level 2 in this example is the 50% resolution. Eight zoom levels are permitted, from 12.5% to 1600%. The third and fourth arguments, 432x288 in the example, are the dimensions for the user's selected screen size. They are used to calculate the size of the viewer that will be displayed. The fifth and sixth arguments are x and y coordinates that specify the area of the image that is being viewed, 0x0 in the example. These parameters are used to embed the correct `gettile.cgi` calls in the HTML page. The remaining two entries in the line are the status code and the number of bytes transferred.

We now define some terms that we use in the next section to describe our results:

Client: Due to the lack of better information in the log files, we distinguish clients by the IP address. Although HTTP servers have the option of reporting the login name as well as the IP address, we have found that this info is rarely reported by the client. However, we don't expect this assumption to create much error since we only use the client name to gather summary statistics about user sessions. The remainder of our measurements use both the client name and image number. It is unlikely that two separate users will be accessing the same image through the same client at the same time.

Session: A session is the unit of access to the web site for a single user. We assume a session to last for 24 hours. That is, if the same client name is found on two log entries that are less than 24 hours apart, we assume that both entries belong to the same session. This parameter is only used for computing basic statistics.

ImagePage: An ImagePage is a the display of a single HTML page. Since the viewer is entirely HTML, a new HTML page is loaded whenever the user clicks a zoom or scroll button. Each such load constitutes a different "view" of the image. Each `mkhtml.cgi` call creates a new ImagePage.

TimeForImagePage: This is the time taken to download a view, that is, for the client to receive the

HTML page, the icons that make up the graphical browser, and all the image tiles. We use this time to study whether the users' access patterns are affected by the performance of the web server. We approximate this time as the difference between the time of the mkhtml call, and the last tile transferred to that client to fill the ImagePage.

ImageView: An ImageView comprises all the ImagePages for a single image in a single user session. Data about navigation within an image is contained in the ImageView.

Our log processing software parses each log and gathers data on sessions, ImagePages and ImageViews. For each ImagePage, the software also calculates the TimeforImagePage. All results are obtained by aggregating the information for Sessions, ImagePages and ImageViews.

4.0 Results

In this section we describe the results of our log analysis. We begin by listing some basic statistics and creating a simple user profile. While this information is not detailed enough to be of use in creating better caches or improving system architecture, it is useful for putting other results in perspective. Next, we characterize access patterns in two ways, *inter-image* and *intra-image*. In inter-image analysis, we ignore individual tiles and consider each image as a single access. In intra-image analysis, we focus on user navigation patterns within an image. Finally, we describe other factors that affect access patterns, such as viewer size and download time.

4.1 Basic Statistics and User Profile

Table 1 lists the average values for basic user statistics in our system. The average time for a session is a little over seven minutes. During this time, the user accesses 2.9 images. During one ImageView (as defined in Section3), 23.5 tiles are transferred, which is on average about 68KB of image data. The total data transferred per ImageView is 93KB. The 25KB difference accounts for the HTML pages that create the viewer and all icons that complete the viewer. Therefore, the data transferred to create the viewer imposes a 37% tax on the image data itself. These transfer sizes are discussed in more detail in Section 4.2. Since the average size of a GridPix encoded file in our system is 1.2MB, only 5.7% of the available data is transferred. This is because the average user does not zoom into the 50% and 100% resolutions, where the bulk of the image detail is available. We have found that average user uses the zoom feature only once, reaching only the 25% resolution.

We also checked whether we could divide users into different categories (simple and experienced users) by the number of images accessed. We found that most users (around 87%) access 5 images or less in one session. Of the remaining users, more than half access less than 10 images. There are only a very few users who access more than 10 images in one session.

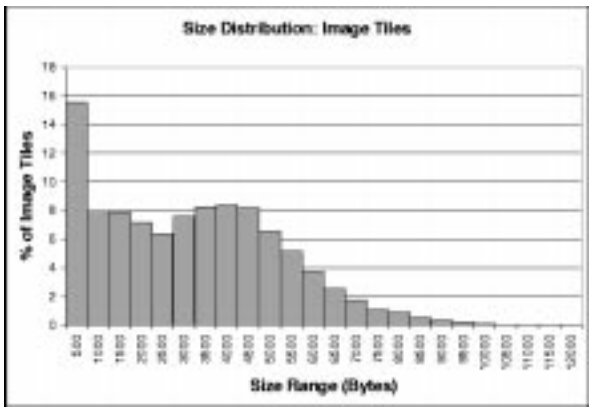
Parameter	Average Value
Time/Session (Minutes)	7:02
Images/Session	2.9
Tiles/ImageView	23.5
Image Data Transferred/ImageView	68.4
Total Data Transferred/ImageView (KB)	93.7
Tile Size (KB)	2.9
Image Size (KB)	1200
Maximum Level Zoomed	25%

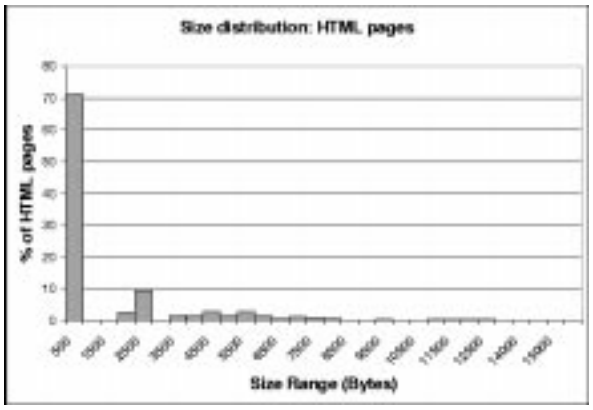
Table 1: Basic Statistics

4.2 Request Sizes

Even though our file sizes are larger than traditional web sites, our user requests are still relatively small, typically 3-5KB each. As stated earlier, all image requests come as either `mkhtml` or `gettile` calls. Our system transfers three types of data; image tiles, HTML files that create the viewer, and icons that complete the viewer. The last category has only a few distinct files. There are a fixed number of icons (less than 15) and the same icons are used for every image served. Therefore, in this section we focus only on the first two types of accesses.

Figure 4(a) shows the distribution of tile sizes. As the figure shows, most image tiles are between 1-6KB. Figure 4(b) shows the distribution of request sizes for the mkhtml calls. Over 70% of the HTML pages are less than 1KB in size. The bulk of the data transferred is in the image tiles and the icons. Section 4.1 showed that the viewer places a 37% tax, on average, on each





Figures 4(a) and 4(b): Figure 4(a) shows the distribution of tile sizes for the images. Figure 4(b) shows the size distribution for the HTML page that constructs the GridPix viewer.

ImageView. This cost comes mostly from the data transferred to create the viewer's zoom and scrollbar icons.

4.3 Inter-Image Analysis

In this section we present access patterns to images in our system. Each ImageView is treated as a single access.

Since all documents on our site are accessed through a keyword search, many documents on the site are very rarely accessed, if ever. Although the site has offered 20,000 to 59,000 images over the five month period, only 5451 unique images were actually retrieved over the five month period.

Prior studies of web traffic have found that web document popularity follows Zipfs Law [12]. Zipfs Law [13,14], originally applied to the relationship between a word's popularity rank and its frequency of use, states the following: the frequency of occurrence of some event (P), as a function of the rank (i) when the rank is determined by the above frequency of occurrence, is a power-law function $P_i \sim 1/i^a$ with the exponent a close to unity. In our case, if images are ranked by their popularity, Zipf's Law states that access frequency and image popularity will have the following relationship:

$$AccessFrequency \sim \frac{1}{PopularityRank}$$

In other words, the n^{th} ranked document is twice as likely to be accessed as the $2n^{th}$ ranked document; popular documents are very popular.

When Access Frequency (Number of Hits) and Popularity Rank are plotted on a log-log scale, a straight line indicates a Zipf distribution. Figure 5 shows image popularity based on number of user requests. As the figure shows, the image popularity appears to follow a Zipf distribution [5,6]. There are a few very popular images and a large number of images that are rarely viewed. Figure 5 also shows that while the more popular documents appear to follow a Zipf distribution, the less popular documents do not. There are a very large number of images that receive only one hit.

Since our document space is flat (no hierarchy of links), we find that the more popular images correspond to popular keywords, such as *picasso*, *monet*, and so on. Some images also become popular for a short time because they are advertised on the Museum main web pages. For instance, the "Image of the Month" advertised in the Fine Arts Museum's newsletter will be one of the top ten most accessed images for that month.

Figure 6 shows the popularity levels of the ten most popular images of each month. For the images that remained on the "top ten list" from month to month, their popularity levels are joined by lines. A few images appear on the top ten list for only one month. This usually happens because that image has been advertised somewhere else on the site (like the Image of the Month). A few images do remain very popular from month to

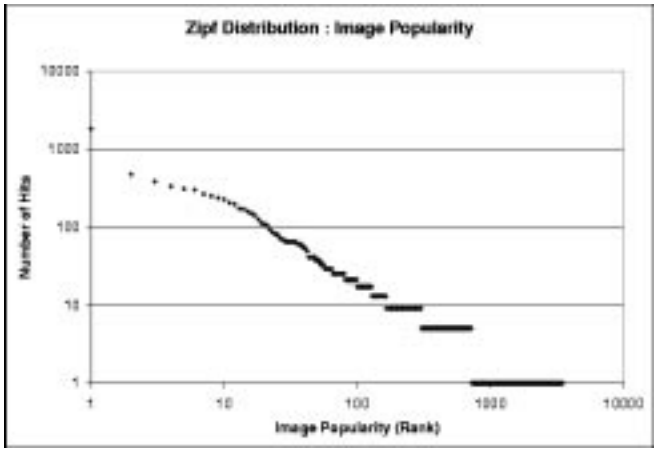


Figure 5: This figure shows Number of Hits per image, sorted by Popularity Rank. While the graph is mostly linear for the more popular images, it becomes a step function for the least popular images.

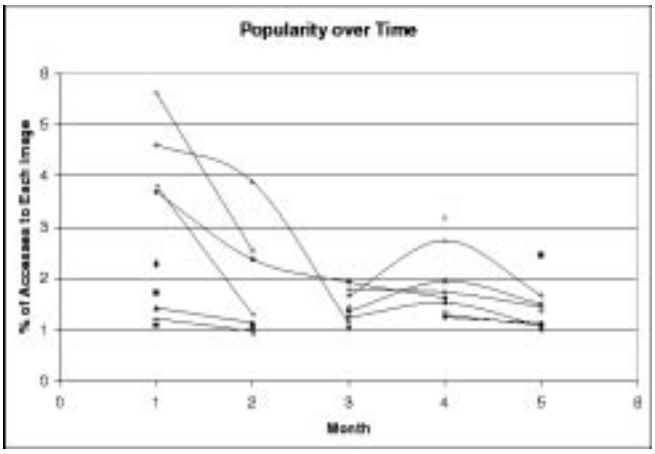


Figure 6: This figure shows the how the popularity of images varies over time. It shows the access frequency for the ten most popular images for each month.

month. These are the images from popular artists. The figure also shows that the percentage of total ImageViews that are due to these ten images decreases over time. In the first month, the top ten images account for almost 30% of all images seen by users. By the fifth month, they account for less than 15%. This is most likely because the size of the on-line collection has almost tripled within the five month period.

4.3 Intra-Image Navigation

This section examines how users navigate within a single image. All images start at the 12.5% resolution, and most images fit within the viewer window at that resolution. Although the maximum available resolution is 1600%, no new information is available after 100%. Once the image becomes too big to fit entirely in the viewer's window, scrollbars are available to navigate north, south, east and west (see Figure 2(d)).

Figure 7 shows the percentage of ImageViews that *stopped at* each zoom level. Since there is no way for a user to reach a high zoom level without visiting all the previous levels, the users that reach a given level are a subset of all users that reached the previous level. The figure shows that the zoom feature is used on approximately 48% of the images viewed. The remaining 52% are viewed only in the 12.5% resolution. As the resolution increases, the number of ImageViews decreases. However some users do zoom all the way to the highest allowed level.

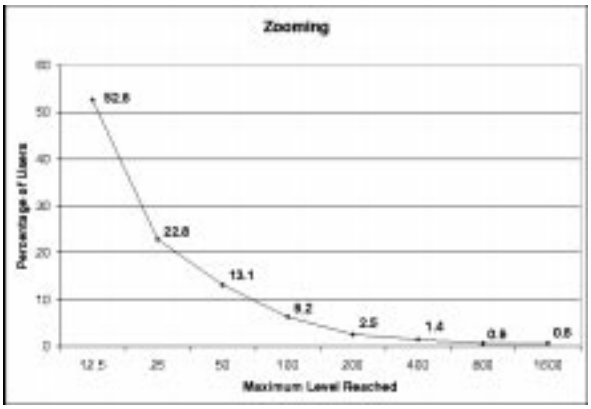
At higher resolutions, users can navigate an image by zooming in, zooming out, and panning. We define panning as scrolling up, down, left or right. Table 2 summarizes statistics on zooming and panning. We find that most ImageViews use zooming as the primary way to navigate within an image. Of all ImageViews that actually do some form of navigation (~50%), more than half only use zoom-in. Only about one third do any form of panning along with zoom-in. The explanation for this behavior could be as simple as the design of the interface. Zooming in only requires that the user click somewhere within the image, a fairly intuitive task and what most users will try first. All other types of navigation (zoom-out, panning) require the user to click either the zoom-out button or one of the scrollbars.

Of the ImageViews that use zoom-in and panning, we have found that most navigation happens at the 50% and 100% resolution. At 12.5% resolution, none of the ImageViews do any panning. At 25% resolution, less than 10% of all ImageViews do panning. At the 50% and 100% resolutions though, 50% of the ImageViews do some panning. There are several possible reasons for this behavior. The first is simply that at the 50% and 100% resolutions, enough detail is available that further zoom-in may not be necessary. The second is that at 50% and 100%, most images will not fit entirely in the viewer window in any screen size.

When panning is being used, it is easy to predict which image tiles are to be retrieved next; they will be the tiles that lie along the four edges of the image. Similarly, with zoom-out, it is quite likely that the tiles that will be needed are tiles that have been viewed before during the same session, since all user's must start viewing the image at the lowest resolution. In our case, since each tile is a separate JPEG file, these tiles will automatically be cached by the user's browser. Zoom-in, however, is not quite that easy. A user can click anywhere within the window area to zoom-in. Therefore, we examined tile access frequencies to determine whether there were any "interesting areas" within an image, areas that were more likely to be zoomed into more than others.

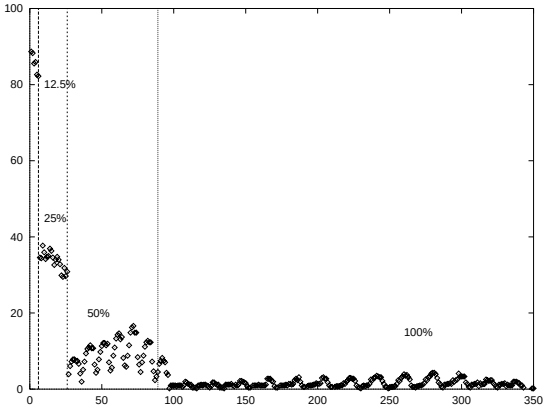
As an example of interesting areas within an image, Figures 8(a) and 8(b) show the average tile access frequency for the two most popular images over 5 months. The tiles are numbered in row-major order, starting from the upper left hand corner of the 12.5% resolution version to the lower right hand corner of the 100% resolution version. Only tiles up to the 100% resolution are shown, because all views at higher resolutions are generated from the same tiles. Vertical lines separate the tiles from each resolution. The figures show that at the first (12.5%) level, there is little difference between the tile access frequencies. However, as the zoom level increases, the gap between the most accessed and least accessed tiles also grows. For both images, at 50% resolution, the most popular tile is accessed more than 15 times as often as the least accessed tile. This example shows that images do have "interesting areas" that are observable at certain zoom levels. However, the figures also show how the overall access frequency decreases as the zoom level increases. Therefore, while there is a disparity between the most popular and least popular tile at each level, the disparity becomes less and less significant as the access frequency of both decreases. The knee of the curve for both images appears to be at the 50% resolution. At this level, the overall access frequency is still between 10 and 20 percent, and the disparity between the most accessed and least accessed tile is more than a factor of 15.

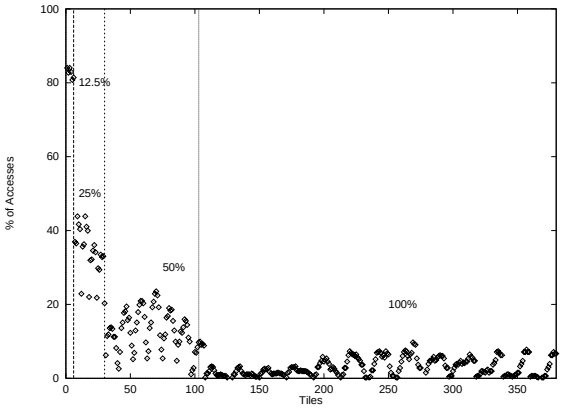
Figures 9(a) and 9(b) describe aggregate statistics on tile popularity. Figure 9(a) shows the average ratios between the access frequencies of the most popular and least popular tiles at each resolution. This data is calculated from all images in the log with 100 hits or



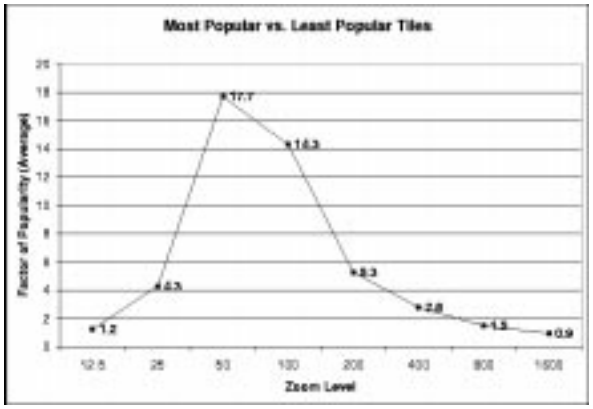
Activity	% of ImageViews
View only (No zoom)	51.8%
Zoom in only	30.8%
Zoom in and out only	2.3%
Zoom in and pan	10.1%
Zoom in, out and pan	4.8%

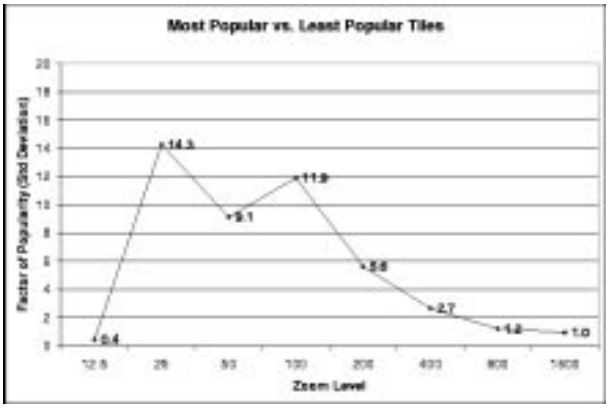
Figure 7 and Table 2: Figure 7 shows average zooming activity. Table 2 lists the percentage of image views over which zooming and panning occurred.





Figures 8(a) and 8(b): Frequency of tile access for the top two images. Tiles are numbered starting from the top left hand corner of the 12.5% resolution version. Vertical lines separate the tiles from each resolution





Figures 9(a) and 9(b): Access frequency ratio between the most popular and least popular tile of each resolution. These values are calculated from all images that received 100 hits or more. Figure 9(a) shows the average values and Figure 9(b) shows the standard deviations.

more. Figure 9(b) shows the standard deviations. The greatest ratios between most popular and least popular tiles happen at the 50% and 100% resolutions (in agreement with Figures 8(a) and 8(b).

4.4 Factors affecting Access Patterns

In this section we try to determine other factors that may affect user access patterns, in particular screen size and download time. Recall that the GridPix system has five screen sizes available:14', 15', 17', 20', 24'. We grouped ImageViews by screen size and analyzed the intra-image navigation patterns of each group separately. The 14', 17' and 24' screen sizes are more popular than the other two; each of the three was used for 22-26% of all ImageViews while the remaining two screen sizes each accounted for 13-14% of all ImageViews. We found that ImageViews done with a 14' screen contained much less zoom-ins: 63.5% of these ImageViews stopped at 12.5% resolution while only 44-50% of ImageViews done with the other screen sizes stopped at 12.5%. After manually inspecting the logs, we suspect that the reason for this behavior is that some users choose a different screen size after viewing one image with the 14' screen. We also found that the number of ImageViews doing zoom-in only increases with larger screen sizes. The amount of panning, on the other hand, decreases with larger screen sizes. This behavior is not surprising, as more of the image is visible in the windows of the larger screen sizes.

Next, we investigated the relationship between access patterns and download time. We estimate the download time (TimeForImagePage) as the time between the load of the first and last tiles for an ImagePage. We assume that the download of the first tile begins immediately after the HTML page that constructs the viewer has been downloaded. Since the HTML file is much smaller than the tiles themselves (see Figures 4(a) and 4(b)), the error introduced by not accounting for the time to download the HTML file is minimal.

Figure 10 shows the relationship between the time taken to download an image view and the probability of a user continuing to zoom-in/navigate within that image. As the figure shows, user access patterns are affected by the image download time. As the file download time increases, the probability of a user interacting with that image further decreases quite dramatically between 1 and 50 seconds. After about 100 seconds, the probability levels off. We also noticed that 92% of all ImagePages were loaded in less that 50 seconds. In fact, 15% of all TimeForImagePages were 1 second or less, 50% were 10 seconds or less. Therefore, most users are very sensitive to download time.

5.0 Discussion

The prior section described our log analysis results. Now we discuss how these results can be used to design

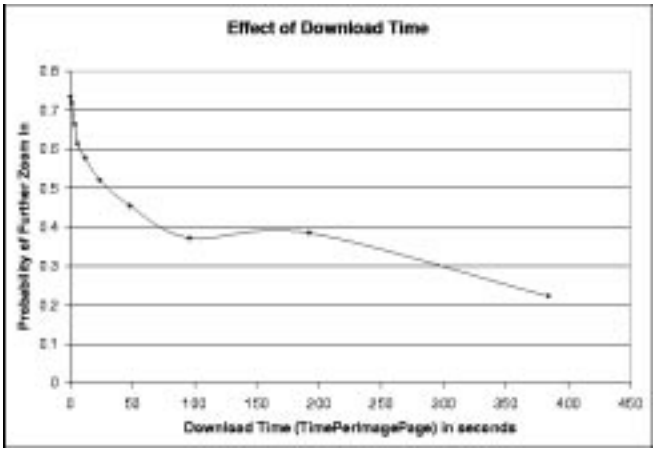


Figure 10: This figures shows the effect of download time on the navigation pattern within a single image. On the X axis is the time to download all the tiles for a single ImagePage. The Y axis is the percentage of users who zoomed/ panned within the image after waiting this much time for the page to download.

systems that exploit the nature of this type of workload. While any quantitative analysis comparing system designs is beyond the scope of this paper, we outline some ideas for improving performance based on the characteristics of this workload. To begin, the analysis revealed some properties that we expect will carry over to other web accessible image collections:

(1) Most images are never accessed: Of the 50,000 images on the site, less than 10% were accessed over a five month period. This is not surprising when the collection is so large, and many art works are little known. The collection behaves the same way as a library, with a few popular books and a vast collection of little known works.

(2) Zipf model of document popularity: Studies have shown document popularity on traditional web sites tends to follow the Zipf model well [12]. In our case, the Zipf distribution models the more popular images quite well, but does not hold up for the least popular images. Since this behavior is most likely because our collection is large and only accessible through keyword search, this feature should carry over to other large web-accessible archives that use keyword search.

(3) Intra-Image navigation patterns: Our analysis of intra-image navigation patterns revealed a few trends. Users very rarely zoom to the maximum image resolution offered. Only 50% of views go beyond the first resolution. Since most users tend to browse, this is not surprising. Also, we noticed that images tend to have "interesting areas" and some tiles are downloaded a lot more frequently than others

(4) Effects of download time: If an image takes a long time to download, the chances of a user interacting with it further decreases very rapidly. This suggests the value of a tile-based image format with multiple resolutions.

What do these observations imply for system design? (1) and (2) imply that such an archive could be implemented using a hierarchy of disk and tape systems (good news for many archives whose data is too large to be kept entirely on disk). Since the more popular documents follow the Zipf model for access frequency, a small fraction of the documents could be kept in secondary storage, with the remaining in tertiary storage. Tertiary storage could also hold the majority of the documents, which may never be accessed. One the other hand, (4) has bad news for archival systems; users are very sensitive to access time.

Observations (1) and (3) are useful for caching. Since web site documents are known to fit the Zipf distribution for popularity, work has already been done on caching policies that exploit this behavior [15, 16]. Observation (3) provides a new dimension to this problem for tile based images. Since 50% of users never view an image beyond its first available resolution, lowest resolution tiles should be kept on-line as much as possible. Since the bulk of the image detail (and hence the data) is in the higher resolutions, the size of the smallest resolution is very small compared to the total file size. For instance, Section 4.1 showed that the average user transfers only 68KB of tile data, 6% of the average file size of 1.2MB. Also, since some tiles are more popular than others, tile-based caching will be more efficient than image based caching.

These observations indicate that there is an opportunity to reduce system cost by using archival storage for the least accessed images or least accessed parts of images. The trade-off is the user's download time, which must be kept low using caching. Exploring these issues further is left for future work.

6.0 Related Work

There are many studies on access patterns of web sites. We reference only a few of them here [2,3,4,5]. Our study differs from these in several ways; the size of the objects stored on the site, and the tile based nature of the content. To our knowledge, we present the first study of access patterns for a web service that has both a large number of large images and a tile based approach to delivering images.

7.0 Summary

This paper presented an analysis of user access patterns to a large collection of tile-based images. We used five months of web site logs to determine how users use a library of images. The images were accessed through a keyword search and available in resolutions from 12.5% to 1600%. We discovered several interesting characteristics of this workload. First, less than 10% of all available images were accessed in a five month period. Popularity appears to follow a Zipf distribution for the more popular images. However, the least popular images deviate from this distribution; there are a very large number of images that received only one hit. While analyzing navigation patterns within images, we discovered the following: 50% of all image views stopped at the first available resolution and the remaining

images had "interesting areas" that were viewed more than other areas. Finally we discovered that users' navigation patterns are very sensitive to download time.

These observations suggest that a disk-tape hierarchy could be used to serve such a workload, although caching will be needed to reduce download time to acceptable levels. The study of intra-image navigation patterns suggested that a tile-based caching scheme could help keep the more popular tiles available on-line. Further investigation of these ideas is left for future work.

We plan to make the logs used in this study publicly available. Anyone interested in obtaining them should contact *td@stampede.cs.berkeley.edu*

8.0 Acknowledgments

We would like to thank Bob Futernick, Dakin Hart and Sue Grinols, from the Fine Arts Museums of San Francisco, who made this project possible by photographing all their art work. Thanks also go to Gabriella Hernandez, Nicholas Hyunh, Kile Zhong, Victor Wong, Abby Thompson, Lila Tretikov, and Tony Le, for preprocessing the images.

This project is funded by the DARPA Roboline grant N00600-93-K-2481, donations of disk drives and machines from IBM and Intel, and the California State MICRO program.

9.0 References

[1] Internet Trends *http://www.genmagic.com/Internet/*

[2] Catledge, L. and Pitkow,J. Characterizing Browsing Strategies in the World-Wide Web, *Proceedings of the 3rd International World Wide Web Conference.* Darmstadt, Germany, Apr. 1995

[3] Arlitt, F. Williamson, C.L. Web Server Workload Characterization, The Search for Invariants. P*roceedings of the 1996 SIGMETRICS Conference on Measurement and Modeling of Computer Systems,* Philadelphia, PA, USA, 23-26 May 1996.

[4] Manley, S., Seltzer, M., Web Facts and Fantasy *Proceedings of the 1997 USENIX Symposium on Internet Technologies and Systems*, Monterey, CA, December 1997.

[5] Woodruff, A. Aoki, P. Brewer, E. Gauthier, P. Rowe, L. An Analysis of Documents from the WWW. *Proceedings of the 4th Annual World Wide Web Conference*, Paris, France, May 1996.

[6] Davis-Brown, B. Sound, Images and Video in the Global Digital Library: Visions and Challenges. Presented at the NISC Storage Users Symposium: Image and Sound in Storage. Monterey California, July, 1998.

[7] Crovella, M.E Bestavros, A. Self Similarity in World Wide Web Traffic: Evidence and Possible Causes. *Proceedings of 1996 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems,* May 1996.

[8] Flashpix information at Kodak. *http:// www.kodak.com/country/US/en/digital/flashPix/*

[9] Talagala, N. Asami. S. Patterson, D. Futernick, B. Hart, D. The Berkeley - San Francisco Fine Arts Database. *Proceedings of the 1998 IEEE Symposium on Mass Storage Systems.*

[10] Talagala, N. Asami, S. Anderson T, Patterson, D. Tertiary Disk: Large Scale Distributed Storage. UC Berkeley Technical Report UCB CSD 98-989.

[11] Asami, S. GridPix: A Method for Presenting Large Image Files Over the Internet. *http://now.cs.berkeley.edu/Td/Papers/.*

[12] Cunha, C. Bestavros, A. Crovella, M. Characteristics of WWW client based traces. Technical Report TR-95-010, Boston University Department of Computer Science, April 1995.

[13] Description of Zipfs Law, *http:// linkage.rockefeller.edu/wli/zipf/*

[14] G.K. Zipf, *Human Behavior and the Principle of Least Effort* Addison-Wesley, 1949

[15] Almeida, V. Bestavros, A. Crovella, M. Oliveira, A. Characterizing Reference Locality in the WWW. Technical Report TR-96-11, Department of Computer Science, Boston University, 1996.

[16] Glassman, S. A Caching Relay for the World Wide Web, *Proceedings of the First International World Wide Web Conference*, 1994.