Defining Data Equivalence for Efficient Access of Images in a Distributed Environment *

Melliyal Annamalai Bharat Bhargava
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907
{*melli,bb*}*@cs.purdue.edu*

Abstract

Technologies such as the World Wide Web have resulted in the access of information over a global network. Data can be text, images, video, or sound. Current network limitations and the large size of data items result in a high response time - the time taken for an image to be transmitted from a remote site to the user. In this paper we focus on reducing this response time for images. Traditionally in an information system exact copies of text data items had to be retrieved for the user but multimedia data items such as video and images can be represented by an *equivalent* data item which is an approximation of the original data item. Such an equivalent data item might satisfy an application equally well and also have a lower response time. Based on the assumption that such an approximation of a data item is sufficient for several applications, we have developed an architecture where an image equivalent to the original can be retrieved instead of the original image itself. Our architecture has two layers to correspond to the two types of equivalence we define: 1) representation layer, and 2) information content layer. At the representation layer we discuss the tradeoff in using lossy compression and lossy communication. At the information content layer we discuss the usage of an equivalent data object which is semantically the same as the original. Our architecture provides support for the representation of several "equivalence levels" and we study the tradeoff of these equivalence level with response time.

*This research is supported in part by NSF grant NCR-9405931 and a NASA fellowship.

1 Introduction

virtual classrooms, and several other fields. The decreasing cost of hardware has also contributed to the prolific increase of image databases. Research is continuing on: indexing, browsing, content-based retrieval, query languages, access structures, feature extraction, high-speed searching of multi-dimensional data, and data models [6, 7, 8, 9, 11].

The design and development of an image database system over a wide area network results in several interesting new research issues. Communication is a bottleneck in any wide area system [3] leading to a large *response time*. We define response time to be the delay between the time the user requests an image data item and the time it appears on the screen. Communication time is the most significant aspect of response time in a wide area network. For example, the roundtrip time for an image file of size 250K from Purdue University in Indiana to a site in California is 26 seconds [3]. The architecture we describe in this paper reduces response time for efficient transmission of image data in conjunction with other modules for access, retrieval, browsing, and display in an image database system.

The underlying idea in our approach is to associate many *levels of equivalence* with each image data item. We use the technique of substituting the image data item to be retrieved by an alternative image data item which will have an associated transmission time which is less than that of the original image. The assumption is that the user would accept a low quality, less accurate data item in exchange for lesser communication time. We base our approach on the assumption that in digital library applications the quality and accuracy of the image can be compromised in order to achieve better response time. The following examples will illustrate the validity of our assumption:

- NASA has a collection images of Mars illustrating its surface. These images constitute valuable data for scientists working in Space Science, Astrophysics, and related fields. They require images with a high resolution including all the technical details concerning the specific scientific field. On the other hand, a K-12 student looking up an image of Mars to do her homework will probably be satisfied with a low resolution of the image

- Medical personnel using computer aided tools for diagnosis will not compromise on the quality of the image. A user looking up the same medical images as part of an encyclopedia will not be interested in equally high quality images

- A user shopping from home might be browsing through the several shopping catalogs using her home computer. The user might want to just skim through products which are of a lesser interest and concentrate on products which specifically interest her. She should be given control to choose the quality of the image she wants for the different products in that shopping session.

- A user reading an article on eclipses might want to view an image to help clarify some points. Any image of an eclipse from the thousands that NASA has will satisfy the purpose. But if the same user wants to look up something about Mona Lisa, there can be only one possible image for her to look at !

- A user looking up the photograph of the United States President is not concerned about when exactly the photograph was taken: one taken three years ago will be as appropriate as one taken an year ago.

- A teacher might use any image of a mountain for a geology class, but a traveler interested in a trip to the Rockies will be interested only in the image of the Rocky mountains.

The first three examples deal with different *quality* levels of an image and the last three examples deal with different levels with respect to *information content*. We say that an image data item is *equivalent* to another if it can be substituted for that data item to the satisfaction of the user in a particular application. We deal with two kinds of equivalence in this paper. Equivalence with respect to quality as illustrated in the first three examples we refer to as *representation equivalence* (abbreviated as **R-equivalence**). Equivalence with respect to information content of the image as illustrated in the last three examples is referred to as *information equivalence* (abbreviated as **I-equivalence**). The two kinds of equivalence are associated with different aspects of an image and are incorporated independent of each other in our architecture.

Sections 2 and 3 discuss the concepts which influenced the design of the various modules. Section 4 describes the architecture and we conclude with a summary of our work.

2 The Representation Layer

In this section we discuss the underlying concepts in defining representation level equivalence or *R-equivalence*. The issues here concern the low level manipulation of the data bytes and not with the information content of the image data item. There are several ways in which different quality levels can be obtained: different levels of resolution, different numbers of colors, different levels of quality based on lossy compression, different levels based on lossy transmission and reconstruction etc. We discuss two approaches: one based on lossy compression techniques and the second based on lossy transmission techniques.

Image data at the storage level is a set of pixels. Each pixel represents the information of one point in the image. We have many different image formats based on how the pixel information is stored [5]: Bitmaps, Pixmaps, GIF, JPEG, HDF, RIFF to name a few. Selection of the format is important in our approach. Different quality levels should be computed with ease and interconversions between them should be computationally simple.

For any given requirement (storage, efficient communication, good quality and so on), there is no one single format that is appropriate to all images. For instance, if we have a vast expanse of one color, then a format designed using dictionary based compression scheme (such as GIF) would be the most appropriate scheme because of the high compression ratio that can be achieved. If there are several colors and a natural progression from one color region to another (such as in natural scenes) then JPEG would be the most appropriate scheme [5].

2.1 Lossy Compression

There are two basic types of compression: *lossless* and *lossy*. Lossy compression results in a much higher compression ratio than lossless compression but the original pixel intensities cannot be perfectly recovered.

The quality of the image degrades as more and more data is lost. In this case the different quality levels correspond to the amount of data that is lost.

The lossy compression scheme that is used as a case study is JPEG. JPEG lends itself naturally to represent different quality levels in an image. The compression procedure includes a quantization step which is the lossy step. The quantization coefficients decide how much data is lost and hence the extent of compression. The notation is usually in terms of percentages. A 10% JPEG file is one which has retained 10% of the original file. 10%, 30%, 50% etc. are R-equivalent to each other and correspond to different quality levels.

A lower quality file does not always result in lower transmission time (see Table 1). A compressed file has to be decompressed at the receiver's site. If the size of the file is very small, then time saved in transmission is offset by the time spent in decompression. If the distance through which the image is transmitted is very small, the time saved in transmission can again be offset by the time spent in decompression. The decompression time is influenced by the computation power at the receiver's site. If a supercomputer is used at the receiver's site, decompression time might be negligible when compared to the transmission time for any file size and distance. We define the total response time to consist of three factors:

$$\textit{response time} = t_a + t_t + t_d$$

t_a: Time taken to access the data object on the remote computer
t_t: Time taken to transmit the data object over the network
t_d: Time taken to display on the local computer

We identify the various parameters that influence response time: transmission distance, image attributes, time of day, computation power of sender and receiver machines, physical media, and storage available [1].

2.1.1 Information Loss Tolerable by an Image

One question that should be addressed is how much can one allow the quality of an image to deteriorate for the image to be acceptable, that is, what the lowest possible quality level is. For digital library applications we have attempted to quantify this loss with respect to the human eye [2]. We measure the amount of data an image can lose during lossy compression before the compressed image becomes distinguishable from the original to the human eye. As long as the compressed image is indistinguishable from the original it will satisfy several digital library applications. Figure 1 shows two images: original and compressed (10% JPEG) image, and the compressed image is indistinguishable from the original. But at some quality level the image will deteriorate beyond acceptance. The amount of data that the image can "afford" to lose imposes a bound on lowering the quality of the image infinitely. We have quantified the amount of data an image can lose using the color information in the image [2]. The scheme is based on computing the data loss in an image by comparing the color histograms of the compressed image with the original. Each image X has a bound X_b associated with it and the image should not lose more information than its associated X_b.

2.1.2 Specification of a Quality Level by the User

When the user specifies a quality level q she is interested in, the system should find out the fastest way to get an image of that quality on the user's screen. If it is cheaper to get a higher quality image (because of decompression time) than that requested by the user, then the better image should be transmitted.

Let q = user specified quality level
$q100$ = 100% quality

Here is a simple procedure to determine the image that should be transmitted:

If $q100 - q \geq X_b$
  Print "Quality q specified too low for image X" **Exit**
If $(t_a^q + t_t^q + t_d^q \leq t_a + t_t^{q100} + t_d^{q100})$
  then *compress and transmit*
else *transmit as is*

The transmission time depends on many factors which are difficult to represent in a closed form expression. It varies with time and we know of no analytical model to accurately model the behavior of the network taking all the above characteristics into account. We have table lookups at several steps to estimate the transmission time.

2.2 Lossy Transmission

Current applications which involve data transfer like FTP, telnet, etc. use TCP (Transmission Control

Protocol) as their transport protocol. TCP provides reliable stream delivery. This reliability is achieved at a cost of performance. We believe that a protocol with a *bounded loss* [10] can be used to transmit image data. A guarantee that a specified percentage of the data is transmitted is provided by such a protocol. The data which is lost in transmission can be *reconstructed* using the data that is received. The amount lost in transmission determines the quality levels. The quality of the reconstruction depends on the method used and the amount of data received.

We conducted experiments by simulating an unreliable protocol by randomly losing packets in an image. The reconstruction technique was extremely simple - a lost row was reconstructed by replacing it with the average of the row above and below it. Some images could tolerate only the loss of one in two rows (50%), while some images could tolerate the loss of three out of four rows (75%). Thus the bound specified on the unreliable protocol varies for each image. Table 2 gives the percentage of data that can be lost while transmitting each image. This gives the lower bound for the quality level. For instance, 10% loss, 20% loss and 40% loss are R-equivalent in the image flowers. The quality level can be ensured by imposing the corresponding bound on the underlying transport protocol. We assume a random loss. Random loss should not be difficult to guarantee by interleaving rows during transmission to avoid the loss of consecutive rows due to bursty transmission.

2.3 Segmentation

We have assumed the same format or unreliable transmission mechanism is used for the entire image. Alternatively one can segment an image into different color regions. Depending on the level of detail in each segment, an appropriate quality level can be chosen for that segment. This will enable the user to choose a low quality for unimportant details such as the background and a higher quality for the objects in the image.

3 The Information Content Layer

If one uses abstraction on physical details of the representation and focuses on the information content of the image, new ideas for reducing the response time can be developed. Equivalence can be defined on the basis of *what* the image represents as opposed to *how* it is represented. One data item is *I-equivalent* to another with respect to satisfying the information need by a user. Figure 2 shows two images which are I-equivalent with respect to the concept sunset. Queries should be routed to a site which is equivalent to the requested data item and is *network closer* to the user than the site the data item is requested from.

Network Distance

In a wide area network, the *network distance* is a more accurate measure of the distance between two sites than geographical distance. We characterize network distance between two sites using two measures:

- Hop Count: This has been mentioned as a distance measure between two points in the internet in [4]. A hop count of n means that n routers separate the two sites.
- Response Time: Experimentally measured response times can be used to estimate the network latency.

These two factors are not the only factors associated with the network distance, but they define "nearness" to the user as a first step.

3.1 Information Level Equivalence Metrics

I-equivalence between two images is determined by the number of attributes which match. Figure 2 shows two images of the sun setting over a ocean. Both images are colored a mixture of orange and yellow which indicate it is sunset. The exact hues of orange and yellow might be different, but they both represent sunset. The common attributes in both images can be described as follows:

```
Main Theme: Sunset

Object 1  : Sun

Object 2  : Ocean

Object 3  : Clouds
```

The attributes depend on the metric used. Some of the metrics are the objects in the image, color content of the image, and texture of the image. With respect to the objects in the image, data equivalence can be defined using the following parameters:

- Number of objects in the image: For example, an image with four clouds is I-equivalent to an image with five clouds

- Position of objects in the image: An image with green trees to the left is I-equivalent to an image with green trees to the right

- Size of the objects in the image: A small flower is I-equivalent to a big flower

With respect to color:

- The 'light' or 'dark' hue of a color. For example, blue and light blue are more equivalent to each other than blue and green.

Color is especially useful to measure I-equivalence between images which have no rigid contours. For example, two images of a blue sky with clouds (of a different shape in each image) can be compared more accurately on the basis of color than on shape. Color-based I-equivalence can be determined using a percentage based match.

4 Architecture

We present an architecture incorporating the concepts discussed in the previous sections in Figure 3. I-modules are located at sites where the user can submit a query. R-modules are located at each site. The user specifies her requirements through a user interface. The I-module selects the site to retrieve the image from based on the I-equivalence requirements of the user. Once the site is selected, it is accessed and the R-equivalence requirements is passed on to it. The R-module at the site determines the quality of image to be transmitted and accesses the image data repository. A module can generate the required quality of the image if it is not available in the repository.

4.1 Query Interface

The users want to select the equivalence level in a query language. Two variables can be used to constrain the query: *quality_level* and *content_match*. Both are quantified using numerical values that can range from 1 to 5, where 1 is the highest level and 5 is the lowest level. Quality_level and content_match are used to specify the degree of equivalence in the representation level and information content level respectively. Since our architecture supports the availability of various metrics, our query language should provide the means of specifying the metric chosen for a query along with the degree of match chosen for the metric. Following is an example:

```
Select all images
   where quality_level = 3 and
         content_match = 2 for colors and
         content_match = 4 for objects
```

4.2 User Interface

The user can also choose to graphically select a required level. The user interface is designed to facilitate choosing of the different levels, overriding default system settings and assist the user in the interactive selection of a level. Icons representing the different representation levels for a standard image are presented in the interface. The different metrics for the semantic levels should be specified in a manner illustrating the differences between them so that the user can choose the metric most appropriate to her use.

Probable times of retrieval for different parts of the network for different file sizes and times of day are also included in the interface. For instance, on Monday mornings the Internet is congested and retrieving a large image file might take a substantial amount of time. If the user is provided with the information that response times are high at that particular time, then she can opt for a very low quality to try and fit it in a busy Monday morning schedule.

5 Conclusions

Accessing images across a wide area network is a problem which is gaining increasing attention because of applications such as digital libraries. Users would like to transparently access an image as soon as possible. They tend to get impatient if they are waiting for an image to appear on the screen for more than a few seconds. We have presented the concept of data equivalence to support the substitution of the requested image data item by another image data item which will have a lesser transmission time than the original. We defined two independent forms of equivalence and discussed the issues involved in supporting equivalence and providing the user with an image with the best possible quality and accuracy. We presented an architecture to enable the user to control the selection of the quality and accuracy of the image in order to have efficient transmission.

References

[1] M. Annamalai and B. Bhargava. An evaluation of transmitting compressed images in a wide area network. Technical Report CSD 95-064, Department of Computer Sciences, Purdue University, October 1995.

[2] M. Annamalai, A. Sundaram, and B. Bhargava. A color-based technique for measuring visible loss for use in image data communication. In *Lecture Notes in Computer Science Series (LNCS) 1134, Database and Expert Systems Applications (DEXA '96)*, pages 39–48. Springer-Verlag, September 1996.

[3] B. Bhargava and M. Annamalai. Communication costs in digital library databases. In *Lecture Notes in Computer Science Series (LNCS) 978, Database and Expert Systems Applications (DEXA '95)*, pages 1–13. Springer-Verlag, September 1995.

[4] D. E. Comer. *Internetworking with TCP/IP Volume 1*, page 569. Prentice Hall, 1995.

[5] D. C. Kay and J. R. Levine. *Graphics File Formats, second edition.* Windcrest/McGraw-Hill, 1995.

[6] T. L. Kunii. *Visual Database Systems.* North Holland, 1989.

[7] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: Querying image by content using color, texture, and shape. In *Proceedings of Storage and Retrieval for Image and Video Databases*, pages 173–182, San Jose, CA, USA, February 1993. SPIE - The International Society for Optical Engineering.

[8] F. Rabitti and P. Savino. Querying semantic image databases. In *Proceedings of SPIE Conference on Image Storage and Retrieval Systems.* SPIE - The International Society for Optical Engineering, February 1992.

[9] A. P. Sistla, C. Yu, and R. Haddad. Reasoning about spatial relationships in picture retrieval systems. In *Proceedings of the 20th VLDB Conference*, Santiago, Chile, 1995.

[10] S. Wang and B. Bhargava. A multi-pass transmission policy: An effective method of transmitting large multimedia objects in a wide-area network. Technical Report CSD 96-057, Department of Computer Sciences, Purdue University, September 1996.

[11] A. Zhang, B. Cheng, and R. Acharya. Texture-based image retrieval in image database systems. In *Proceedings of the 6th International Workshop on Database and Expert Systems*, October 1995.

Figure 1: Original and 10% JPEG Versions of the Lion Image

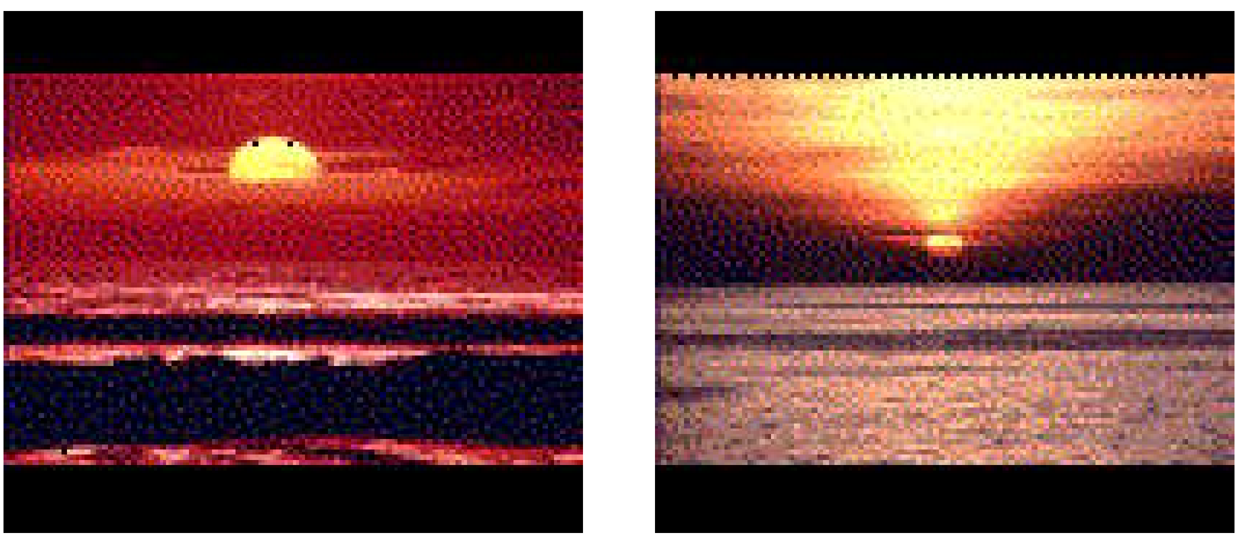

Figure 2: Two *I-equivalent* Images of a Sunset

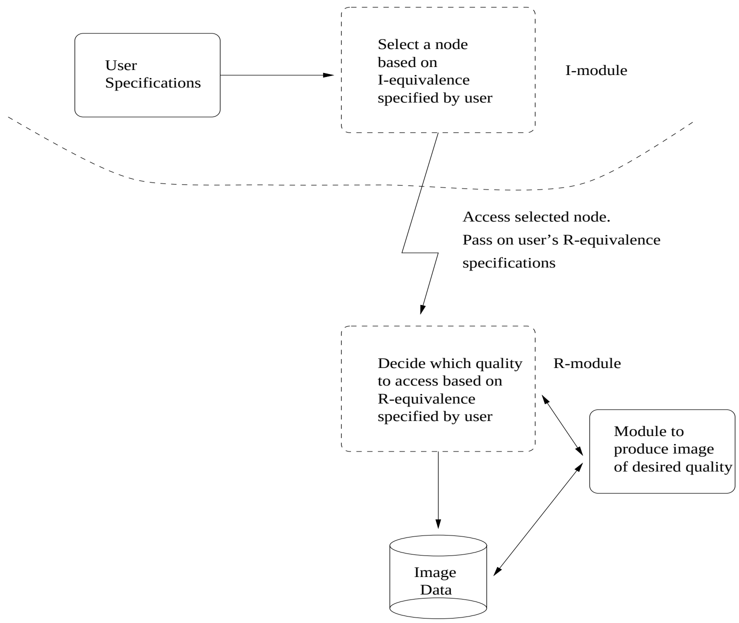

Figure 3: Architecture of a Distributed Image Database System Supporting Equivalence

Time taken	Case 1: WAN	Case 2: LAN
Round trip Uncompressed image	16.3 s	1.2 s
Round trip Compressed image	3.6 s	0.9 s
Convert Decompression time	2.9 s	2.9 s

Table 1: An Example: Using an Compressed Image is Sometimes Viable

Image	Size (Kbytes)	% that can be lost
Flowers	287287	40%
Curve	308278	75%
Lion	168910	50%
Scenery	173446	66%
Mars	116678	75%
Galaxy	389278	66%
Red	289326	50%

Table 2: Percentage of Data that can be Lost in an Image