CITI Technical Report 92-5

Digitized Image Data Compression and Transfer

Wafik Farag
farag@citi.umich.edu

Fred Remley
fred@citi.umich.edu

ABSTRACT

Information represented in a digitized image format (such as remote sensing, satellite, or photographic images) is becoming more readily available. These image files contain massive amounts of information requiring more efficient storage and transfer methods. The large size of these image files requires investigation of quality issues, transfer methods, and storage mechanisms.

Digitized image data compression allows for faster file transfer over networks and requires less storage space for large image files. While moving over networks, compressed files are less affected by network errors because of the reduced transfer time. With data compression, however, the quality of the image is sometimes sacrificed for the sake of size and speed.

This report discusses our research of digitized image data compression and transfer, including storage mechanisms used, compression techniques currently available for Macintosh and UNIX platforms, and the methods used to transfer image files over networks.

October 30, 1992

Center for Information Technology Integration
University of Michigan
519 West William Street
Ann Arbor, MI 48103–4943

Digitized Image Data Compression and Transfer

Wafik Farag

Fred Remley

October 30, 1992

The capability of moving image files across networks is a critical factor for the future of video imaging. Because of the large size of the image files, problems with storage, slow transfer times, and loss of quality have been concerns. With these concerns in mind, we began investigating adequate storage mechanisms, a variety of display software packages for image compression, and image file transfer methods.

1. Storage Facilities

Students in most University of Michigan classes have access to personal computers—in particular the Apple Macintosh. As image files are normally larger than word-processing, spread sheet, or graphic files, and floppy disks and conventional hard drives do not sufficiently store large files, it is essential to use a centralized method of storage.

We found two alternative mechanisms to store the larger image files—the Institutional File System (IFS) and Kodak's Photo CD.

1.1 The IFS

The IFS is a mainframe-based, distributed file system that provides the flexibility and power of combining local workstations and centralized file sharing. It allows access to stored files from different locations using a variety of networked platforms. Due to the distributed nature of the file system, image input devices (scanners, compact disk players, or disks) need not be available at every point on the network.

We used the IFS as one of our storage mediums because of its capacity to store large files.

NOTE: The IFS Project is a collaborative effort between the University of Michigan's Center for Information Technology Integration and IBM. AFS (Transarc Corporation) is the file system upon which the IFS is built.

1.2 Kodak Photo CD

The Kodak Photo CD has made contributions to the storage of high resolution image files and enhanced color quality. The CD's storage capacity is determined by the size and type of the image file and the number of disk-writing

sessions. The color quality is as well-defined as that of a photograph.

The Kodak Photo CD begins with 35mm image negatives encoded into 3K pixels x 2K pixels to a depth of 24 bits. Images of this size would result in a storage capacity of only 30 image files on each CD. However, Kodak Photo CD uses compression (e.g. Chroma Subsample, Quantization, and Symbol Encoding) to reduce each image size from ~18 MB to ~4.5 MB. Using this compression technology, approximately 100 image files may fit on a 600MB CD. Each ~4.5 MB image file contains five resolution levels. Following are the five resolution levels that together make up for one Photo CD image file called "Image Pac."

- **Base/16** (192xl28 pixels): Best used for index prints and low resolution display.
- **Base/4** (384x256 pixels): Best used for browsing, low resolution display, or small prints.
- **Base** (768x512 pixels): Best used for small prints and computer displays.
- **Base*4** (1536x1024 pixels): Best used for high quality prints.
- **Base*16** (3072x2048 pixels): Best used for highest resolution prints.

The first three components (Base/16, Base/4, and Base) make up an Image Pac of uncompressed components. The last two components (Base*4 and Base*16) are Huffman encoded (a data standard). The recommended hardware configuration for reading a Photo CD is the Mode 2 CD-ROM XA multi-session. Mode 2 is used for Photo CD image data. Mode 1 is used for computer software. The XA version is used for Photo CDs with interleaved audio. The multi-session format permits a Photo CD to store image files from multiple rolls of film. The image files on each roll of film may be written at different times.
A number of CD-ROM drive manufacturers are supporting the Photo CD format version. For example, the Philips CDD 461 Audio ROM, the Pioneer DRM-604X CD-ROM mini-changer, the Sony double-speed CD-ROM, and the Toshiba XM-3301 are all suited for use with the Photo CD.

2. Platforms and Display Software

Two platforms, the Apple Macintosh (Macintosh IIci with RasterOps 24 bit display) and the UNIX-based Sun (Sun SPARC station IPC) were used for testing. We analyzed several display software packages, image sources, compression methods, and image formats for each platform. The following text details display packages best support images, what image sources are available for each platform, what formats are most commonly used for each platform, and which transfer mechanisms respond best.

2.1 The Macintosh Platform

We conducted most of our research on the Macintosh IIci because of the advanced display software packages that it accommodates and its prevalence on the U-M campus.

2.1.1 Macintosh Display Software

- **ImagePak - RasterOps:** One of the best features of ImagePak is its ability to display the file size and the type of file prior to opening it. Users may click on the file name, or icon, and the file data will display. ImagePak is capable of presenting a preview using a thumbnail image display, but the images must be created before the thumbnail can be seen. The package also allows access to more than one image file at a time. The type of image and its size are displayed in the lower left hand corner of the image. When choosing image files, the ImagePak menu lists only images stored in ImagePak format.

  ImagePak uses the compression techniques lossless and lossy. These techniques are

completed either through hardware (using a board), or by software. From testing several images, we found the difference between hardware (more costly) and software is not worth the extra expense. ImagePak compression has two good features. It provides an auto-decompress option, that lets users decompress an ImagePak compressed file—even if they move their image file to a machine that does not have the ImagePak software. Secondly, it allows users to build a Huffman encoding table for an image file.

ImagePak requires more memory for startup and has a slower compression speed than similar packages. However, importing images from other packages is not feasible.

- **PICTCompressor - Apple Computer:** PICTCompressor is one of the most useful packages found in our testing for PICT format. It is capable of creating a thumbnail image as the file is read, and allows saving the thumbnail for future use. It requires very little memory (under 1500K) in comparison to other packages. It stores compression information, such as compression time and the quality level, which is helpful when analyzing results. PICTCompressor displays image format and the compression method in the top left-hand side of the image. It provides the image size as well as the percentage of the displayed image. PICTCompress lets the user choose an off-screen buffer to display images. Fewer problems were experienced displaying images from other applications while using the PICTCompressor.

  One of the major drawbacks of the PICTCompressor is its inability to display more than one image at a time.

  The compression in PICTCompressor is derived from QuickTime, where options such as video, graphics, and animation exist, but options for still images are few. Nevertheless, its compression is quite good for still images, when Photo-JPEG is selected.

- **Adobe Photoshop:** Photoshop is a frequently used tool for displaying and editing quality images. It operates on input drivers for reading different image formats called “plug-ins.” For example, for reading images from a CD player, users need the correct plug-in, which is usually sent by the image format producer (in this case the CD player). The different plug-ins make Photoshop adaptable to many formats.

  Photoshop reads images saved in PICT format by other vendors, such as ImagePak, PICTCompress, and KIC (a Kodak format).

We did not fully test the other packages we examined because of problems we encountered. The packages included: JPEGView, which has part of the functionality of PICTCompress; Giffer, which is for GIF (Graphic Interchange Format); and QuickGIF, which has similar functionality to Giffer.

The biggest problem we encountered while testing these packages was the inconsistent implementation of the JPEG standard. Nearly all of the display software packages tested claimed to be using the standard JPEG implementation. However, the ability to compress an image with one package did not guarantee the ability to read it or decompress it with another package. This lack of a standard implementation caused difficulties in evaluating these software packages.

2.1.2 Compression Rate

One of the significant results of our testing was the ability to achieve up to a 99% compression rate on the images. We were able to achieve notable compression of some images. For example, starting with a 4610K image file in raw format, we converted it to a 4589K PICT format file. Next, we reduced the file size to 654K with JPEG techniques, with good results.

Applying further compression, we reduced the image file to 31K, but with reduced image quality. We determined the number of colors and detail in an image file affects the maximum compression ratio.

2.2 UNIX-Based Platforms

Because UNIX workstations are not as common as Macs on the U-M campus, they tend to be less frequently used as image display systems. For this reason, we devoted less attention to the UNIX workstation.

The UNIX-based platforms on the U-M campus are mainly represented by Sun, DEC, and IBM. We selected a Sun SPARC station IPC for our work.

2.2.1. UNIX Display Software

- ***xv:*** *xv* is one of the best image display packages available for UNIX platforms. It can display at least nine image formats and has many options, including the ability to change the intensity, brightness, sharpness, RGB, gamma values, and saturation. Users can also resize, dither, rotate, crop, or smooth an image. It provides image file size information as well as the number of colors in the color matrix used in the image file. The new *xv* version has built-in JPEG image compression.

- ***xloadimage:*** The *xloadimage* package is one of the most popular display packages due to its simplicity of use and its varied options. It is not as comprehensive as some of the other packages, but has enough options to display images robustly.

- ***pbm* (portable bitmap file format):** *pbm* is one of the most commonly used packages for image type conversion and display. Its extensive set of filters allows conversion from almost any image format to another. *pbmplus* has added more functionality for new types of images.

- ***urt*(Utah Raster Toolkit):** The *urt* package adds flexibility for displaying images on a variety of platforms (e.g. Mac, Sun, SGI). *urt* is similar to *pbm* because of its image type conversion filters. It also contains some of the functionality found in *xv* as well as other tools. Its ability to accommodate different platform displays is its most desirable feature.

UNIX workstations are useful in handling large image files, including some files we encountered that were 150 MB or larger. These image files are nearly impossible to handle on a PC.

Before converting files on UNIX workstations, users must check three areas:

- The bit representation of the machine being used. Different machines have different architectures for bit representation. This process might require the user to write programs for bit conversion, or they may use the *conv option* in the *dt* command.

- The cache size. For running any computation on a large image file, users must have a large cache—especially when using AFS. In cases where cache size is limited to a small quantity, any conversion or computation might run slowly or crash.

- The screen size. Even after reduction, image files might still be so large that normal computer screens cannot display them. We used *tvt~n* as an X-Window manager because of its virtual capabilities in displaying images.

The larger image files usually are in *raw* format; however, most of the images we experimented with were in GIF format.

For converting to Macintosh PICT format or from PICT to GIF, *pbmplus* display software performs well. Using some of the *urt* utilities, we reduced a 156MB image to 1.3MBs. With such a large reduction, the quality of the image was sacrificed.

3. Image Transfer

As most image applications are used on personal computers, we investigated possibilities for network transfer of images to large servers.

Network interconnection is the most logical way to connect to large capacity file servers. Two methods of moving images across networks are currently available at the U-M—the IFS and the File Transfer Protocol (FTP).

3.1. IFS

The IFS includes an AFP/AFS translator that enables Macintosh clients using the AppleTalk Filing Protocol (AFP) to access files stored in AFS. It provides an efficient method for moving such images. Macintosh users store and retrieve AFS files using procedures of their native Macintosh interface; for example, by dragging the appropriate file icon to or from an IFS volume icon.

Transferring large files, such as image files, revealed some problems with the original translator. The image files took a long period of time to transfer to IFS. A single image file of 4 to 5 MB could take up to 30 minutes or more to transfer. Data loss sometimes occurred while transferring image files from IFS back to the Macintosh. This problem was more evident while transferring image files of high resolution, e.g. Photo CD Base*4 image files. By working with IFS Project personnel, we were able to resolve many problems; however, a few remain. For example, during peak load times on the network, problems still persist. When the network isn't busy, the transfer process is quite rapid.

In some cases, retrieving image files from the IFS is faster than using a CD. This is most noticeable with high resolution images that require longer periods of time (80 seconds or more) to transfer. AFS is 7 to 10% faster when displaying Base*4 Photo CD images using Photoshop. In lower resolutions, the IFS did not out-perform CD access time. For example, a 20 second CD image display, took the IFS 22 seconds, a reasonable tradeoff in comparison to having the hundreds of images available on the IFS.

3.2 FTP

Images can also be transferred via the File Transfer Protocol (FTP). Because this technique could be cumbersome, some knowledge of UNIX (IP address numbers and directory locations) is required for this procedure. In contrast, the IFS only requires a Macintosh working knowledge to transfer a file. In cases where the IFS translator is not accessible from another AppleTalk network, the FTP becomes a viable solution for image transfer.

FTP has good error checking methods that prevent data loss from occurring. FTP provides data on the time spent during the transfer and the size of the file. This feature is helpful for monitoring the transfer time.

Our tests conclude that FTP is sometimes more robust and faster for image transfer, but the IFS is easier to use and requires no difficult setup.

4. Conclusion

We find that Kodak Photo CD technology contributes to the quality and portability of image access, but having a CD reader on every workstation is costly. Using the IFS as a storage medium proved to be a solution for a distributed computing environment. The problems we encountered while using the IFS to store large files are being addressed.

The results of this project demonstrated that image data compression is a useful tool. We

also found that the level of acceptable compression varies as a function of image content. One experiment resulted in as much as a 99% data reduction. Data compression results in faster access time for rapid image access. We recommend that users evaluate the final use of the image before deciding on the most suitable compression ratio.

5. Acknowledgements

Many thanks to the Eastman Kodak Company for its donations of software, disks, and technical information on the Photo CD system.

The brands and product names appearing in this report may be trademarks or registered trademarks of third parties.