Image BOSS: An Image Database System Designed for Research

Mahlon C. Stacy, Kurt E. Augustine, Richard A. Robb

THE PRIMARY PURPOSE for the use of a database is to help manage the vast amounts of data generated in today's "information age." Like other types of data, biomedical image data is currently being generated at enormous rates, and shows no signs of decreasing. Large volumes of image data become unmanageable under current computational systems.

Mayo Foundation supports a significant number of research studies, which include medical images (computed tomography, magnetic resonance, etc) and biological images (optical microscope, confocal microscope, etc). Research topics range from basic science through clinical diseases. The Biomedical Imaging Resource (BIR) is a core resource at Mayo with a mission to provide expertise and computer resources for advanced image visualization and analysis. The BIR affiliated research activities generate large volumes of image data. For example, gastrointestinal nerve studies, which derive images from optical microscopy typically generate 50 to 70 images per nerve, each image containing 512×512 8 bit pixels, or approximately 262 kilobytes each. Similar cell types are imaged using different staining techniques. Several sets of images are collected from different organs in the body, and several different types of animals are under study, resulting in hundreds of sets of raw image data. After the raw data is collected, image processing is used to register, segment, and scale the images into an isotropic volume image. From these volume images, measurements are taken, objects are segmented and various visualization techniques, such as volume and surface rendering generate further images, which are used to view the cells and study the 3-dimensional relationships in the data. To date, the BIR has accumulated over 5,500 images for this single physiology investigator's work, requiring over 40 Gigabytes of storage.

Problems to be Solved

Finding the right image. For projects with a small number of images, the Unix filesystem hierarchy usually provides sufficient organization for management of the data. However, as the number of images grows, it becomes impossible for investigators to remember the arcane mnemonics used in naming directories and files, suggesting the need for a better method. Historically, this information has been kept in laboratory notebooks, which impose a project-based organization scheme on the data. These are difficult to search for specific files, which may be created or referenced in other projects. Compounding this difficulty, much of the research is conducted by postdoctoral fellows and graduate students, and the investigator does not personally have the visual memory cues necessary to find a particular image based on someone else's file naming system. Furthermore, when these fellows and students leave, only the notebooks retain the information about the images, and searching the notebooks manually is difficult.

Capture of data from a dynamic, existing system. While the typical database is intended to store the data, storing all of the Mayo images inside a database would interfere with existing image processing software, which expects to find images in the filesystem. A number of different software packages are used in a variety of computing environments to massage and view the images. Imposition of a database system as the sole interface to the image data would prohibit users from accessing their image data directly. Consequently, the system needs to capture image metadata from the filesystem into the database on a regular, but unobtrusive basis. The image data itself must remain freely accessible in the filesystem.

This "uncontrolled data" design creates other problems. For example, users would not be able to move files without disconnecting the image from its metadata in the database. The design approach taken makes attempts to handle this gracefully, but work remains to provide multilevel storage of image files between magnetic disk, optical disk systems, and tape archives while keeping the images correctly connected to the metadata in the database.

From Biomedical Imaging Resource, Mayo Foundation, Rochester, MN.

Address reprint requests to Mahlon C. Stacy, Biomedical Imaging Resource, Mayo Foundation, Rochester, MN 55905.

*Copyright © 1997 by W.B. Saunders Company
0897-1889/97/1003-1024\$3.00/0*

Flexible attributes. The diverse studies conducted at Mayo generates a variety of research protocols and attendant attributes needed and used to characterize the image information. The design for the database needs to provide flexible attribute naming schemes to accommodate an individual or laboratory's particular needs.

Sharing of images and information. While much research is conducted independently by each laboratory, some research crosses disciplines. Where such collaborative needs arise, the system must provide properly balanced control of access to the database combined with appropriate security.

Provision for content-based queries and selective batch processing. Historically, image analysis in the BIR has been primarily based on ANALYZE,¹ which provides a highly integrated interactive image exploration environment. Often explorative research on images creates repetitive methods to be applied precisely to a set of images. The database design should incorporate the ability to select sets of images and execute the same image processing tasks on each image in the set, effecting a batch processing scheme. Furthermore, if the results produced by this processing can be measured against predetermined criteria and images from the set accepted or rejected, this operation can be viewed as a content-based query on the image set.

To address these and other problems, the BIR has been conducting a research project to develop an image database system dubbed the "Image BOSS" (Biomedical Object Storage System).

IMAGE BOSS SYSTEM OVERVIEW

One of the key features of Image BOSS is an automated image data capture facility. The goal was to provide a database that works with the Unix filesystem rather than hiding it from the user. Creating a closed system in which users must interface to the database may be easier to manage, but becomes much less flexible for the user. Experience shows that most users work first with the filesystem until their image files become too numerous to manage, then they want to interface with a database. Image BOSS is designed such that the interface to the user will be an up-to-date representation of the filesystem.

Image capture. To capture the image file data for Image BOSS, a simple database administration tool called "scavenge" is executed on a regular

basis. Scavenge scans the filesystem searching for new files and probing them to determine if they are images. The frequency of "scavenges" depends on the size of the filesystem and how quickly users need to see new images reflected in the database. A typical frequency is once per day, after normal working hours if possible.

Image browser/organizer. The primary purpose for Image BOSS is to provide the user with a means for managing complicated sets of image data. The tool that provides this service is the image browser/organizer. Derived from the way researchers tend to organize their data, this tool uses a bookcase/workbook metaphor as its primary interface (see Fig 1).

The goal was to provide a way for the user to browse and organize images in a familiar structure. A user may "page through" any one of a number of workbooks in search of a particular image or set of images, which are represented by a "thumbnail" version of the image. Images may be organized on a per page basis, or at a higher level by creating specific sets of workbooks. To bring up the browser/organizer tool the user executes the "db" command, which pops up their personal bookcase window. The user views all of his/her personal workbooks, the first of which is a primary workbook, followed alphabetically by secondary workbooks. The primary workbook is generated and updated automatically by the scavenge program along with several other database administration tools. It is a current representation of all the user's images found in the filesystem and is read-only. Each page is associated with a directory in the filesystem and contains thumbnails representing all images within that directory that are owned (in the Unix sense of ownership) by the given user.

Secondary workbooks are created by the user and can contain two types of pages: image pages and text pages. Any image thumbnail may be placed on the image pages, including thumbnails representing images that are not owned by the user. Principally, this is accomplished through the drag and drop feature. Queries may be performed based on image attributes, returning a pages of image icons as a result. The text pages are used to annotate image pages or for any other purpose as determined by the user.

External tools. A variety of other tools and applications may interface with the db tool as well. If a user sees an image thumbnail believed to

Fig 1. Image BOSS workbook with image page and text page displayed.

represent the data wanted, they may view it in full scale and color by dragging the thumbnail and dropping it on an image viewer tool called “iv.”

One of the key features of Image BOSS is its facile integration with ANALYZE_{AVW}. If a user wishes to analyze or process an image but is not sure where the image is in the filesystem, they can find it in a workbook, drag the thumbnail onto the ANALYZE_{AVW} window and have it loaded into that application.

IMAGE BOSS DESIGN

The key to understanding and managing large amounts of data is to abstract the useful information from the original data. This must be done in a way that minimizes the amount of information provided to the user, but not to the point of rendering it useless. The Unix filesystem offers a form of image data abstraction by using filenames and directory structures, which are very efficient

from a data volume management standpoint. The problem is that too much information about the image file is hidden. As the number of files grows, it is increasingly difficult to search and differentiate one image set from another. Typically, some type of naming convention is implemented to aid in image file management. In research, this technique inevitably breaks down at some point, prohibiting efficient use of image data. In some cases, utility can be extended through “file managers” which hide some of the operating system detail, but these offer little additional information about the image file.

Thumbnail abstraction. Visual cues are a powerful tool to aid in recognition, differentiation and management of information. This is especially true for image data. A primary abstraction of image data should therefore include some representation of the image itself. The most common form of image data abstraction is a small representative image called a

Fig 2. Image thumbnails showing various format and color model icons.

“thumbnail” (see Fig 2). A thumbnail of a 2-dimensional image is simply a scaled down version of the original. In Image BOSS, this implementation is a 64 pixel by 64 pixel image, dithered to 64 predefined colors. For three-dimensional (multiple 2-D slices) images, these thumbnails default to the middle slice of the volume image. The exception is the case where the image data set is actually multiple disjoint 2-D images (eg, set of slides for a slide presentation), in which case the first image in the set is used for the representative thumbnail.

While image thumbnails provide a useful form of data abstraction, there is significant information still hidden. For this reason, additional information is abstracted from the original data, which is helpful in managing the image data. A primary interface to the actual image data is still the Unix filesystem, so the full pathname associated with the data is displayed with each thumbnail. Because the thumbnail is a 2-D representation, the user must have a simple way to determine the dimensionality of the dataset. This is accomplished with a small icon attached to each thumbnail. Because of the scaling and dithering that takes place to produce the thumbnail from the original image data, users are not able to determine the format of the image (binary, gray scale, color mapped, true color) so an additional icon representing the image format is attached to the thumbnail.

Attribute abstraction. Each of the above image data abstractions are always provided and visible to the user because they are common to all images. The user that obtained or generated the image data may wish to associate other information with an image, which would require user-specified abstractions. Researchers often associate characteristics with their sets of images that are unique to their particular discipline. This calls for a way of assigning data abstractions that are unique to each user. For example, a biologist may wish to assign a

“stain” attribute to the images, where as an anatomist might wish to assign an “anatomical structure” attribute. This feature is provided in the form of text based image attributes. Associated with each thumbnail is a set of 15 attributes, which the user is free to define in any way s/he sees fit. This information, while considered an abstraction of the original data, is accessed through a secondary dialog interface to minimize clutter on the thumbnail.

RESULTS

The Image BOSS project provides new and advanced capabilities for support of imaging science investigators in a large research organization, which generates copious amounts of images and image related data. While the development of the system is still in progress, sufficient work has been done to demonstrate the feasibility and power of the approach. A number of problems encountered with this technique have been solved. Users are provided with intuitive interfaces and useful tools to collect and organize their images and related image metadata. As the images remain in the native filesystem, all existing software systems are unaffected. The use of icons organized into workbooks give users intuitive visual cues to image content, which can be organized into classical sets in a familiar lab workbook metaphor. Duplication of images is avoided as an image may have many icon representations pointing to the same (raw) image. Sharing of image information is enabled through the sharing of workbooks.

Work remains to be done on the implementation of content-based queries and batch processing methods. The current system infrastructure is designed to incorporate these features. Additional areas of study include simplified methods of archive storage and the definition and implementation of collections of image related files.

REFERENCE

1. Robb RA, Hanson DP: The ANALYZE software system for visualization and analysis in surgery simulation, in Lavallée

S, Taylor R, Burdea G, et al (eds): Computer Integrated Surgery. Cambridge, MA, MIT Press, 1995, pp 175-190