

AN INTERACTIVE ICON INDEX - IMAGES OF THE OUTER PLANETS

William F. Eddy and Audris Mockus,
Dept. of Statistics, Carnegie Mellon University,
Pittsburgh, PA 15213-3890

Key Words: Exploratory Data Analysis, Image Database

Abstract

We are interested in the exploratory analysis of large collections of complex objects. As an example, we are studying a large collection of digital images having nearly 30,000 members. We regard each image in the collection as an individual observation. To facilitate our study we construct an index of the images in the collection. The index utilizes a small copy of each image (an icon or a “thumbnail”) to represent the full-size version. A large number of these thumbnails are laid out in a workstation window. We can interactively arrange and rearrange the thumbnails within the window. For example, we can sort the thumbnails by the values of a function computed from them or by the values of data associated with each of them. By the use of specialized equipment (a single-frame video disk recorder/player), we can instantaneously access any individual full-size image in the collection as a video image. We regard our software as the beginning of the development of exploratory tools for studying collections of images and other complex objects as we routinely study batches of numbers. We expect that the concept of a visual index will extend to other collections of complex objects besides images, e.g., time series, functions, and text.

1 Introduction

To explore a very large dataset (e.g., an art collection, a library, or the Internet) one needs advanced methods of access to it. Long ago it was recognized that

simple linear access is not efficient. Art galleries, libraries, and Internet users, for example, have all faced a somewhat similar problem. They are interested in the selection of subsets of the available material based on the contents of the individual items.

The basic solution is to create an index, so that information can be accessed by some attribute. For a book, several indices usually are created, such as a subject index, an author index, a concordance, a KWIK (keyword in context) index, etc. The problem is greatly exacerbated when the dataset consists of objects which cannot easily be subdivided, such as images or objects that can not be usefully reduced in size, such as digital images of text; with books in a library, for example, the title can be easily separated from the rest of the book and cataloged. A traditional index uses one of three forms to create the index entries:

1. free text (this is the simplest to create and use but, as a consequence, has a much higher probability of missing an appropriate object when searching);
2. keywords (this requires the expertise of an editor to limit the vocabulary and provide some uniformity of usage);
3. codes (the Dewey decimal system devised in 1876 by Melvil Dewey is one of the first).

We are currently working with a collection of approximately 30,000 images. We are only peripherally interested in “image processing” but much more interested in finding “interesting” images and groups of images in the collection. Of course, inspection and manipulation of individual images is important to us also. We are developing a specialized system of hardware and software to help us explore and interact with this collection of images; it currently includes two workstation monitors, several CD-ROM players, and some specialized video equipment (a laser video disk recorder/player and a TV monitor). We regard this system as a first tool for interactive exploratory analysis of a large collection of complex objects.

The main principle of our system is to represent each image in the database of images by a small icon (a “thumbnail”) and then simultaneously display and interactively rearrange those icons. We can detect and classify the thumbnails of interest via visual inspection and retrieve the original images for further analysis. Because of our ability to interactively reorganize our index, we refer to our system as an **Interactive Icon Index**; hence **I³**; hence, **Iccube**.

In the next section we describe the collection of images in our example database of images and the objectives that we had in mind. In Section 3 we give an outline

of the design and implementation of our system. Section 4 briefly describes our future plans and related work in the database management literature. We conclude with a brief summary.

2 Our Example Database of Images

Our example database of images was obtained from NASA's Voyager Project and Planetary Data System. The description here was derived from Eliason et al. (1991). This collection includes all of the images acquired by Voyager 2 as it passed by the planets Uranus and Neptune, and selected near-encounter images of Jupiter, Saturn, and their satellites taken by both Voyager 1 and 2. The archive resides on twelve CD-ROM volumes. Each volume contains approximately 2500 images stored in individual files.

We quote slightly more detail from Eliason et al. (1991):

There are two cameras on board each Voyager spacecraft, one with a wide field of view and the other with a narrow field of view. Both cameras return data in a digital raster-format containing 800 scan lines and 800 samples per scan line. Each picture element (pixel) in the two dimensional image array is represented as an 8-bit value, and the value—in the range 0 to 255—is proportional to the amount of light detected at that point. The cameras are equipped with color filters so images taken through complementary filters can be combined during ground processing to produce color images.

The set of images on the CD-ROMs are raw; no processing has been performed other than organizing the original telemetry into raster-formatted files and compressing the image data using a Huffman encoding algorithm.

To make full scientific use of the image collection, it is necessary to understand the radiometric and geometric properties of the camera system and perform corrections to the data. A number of image processing systems are available which provide radiometric and geometric corrections, display capabilities, and analysis tools for Voyager images.

One example image from the collection is given in Figure 1.

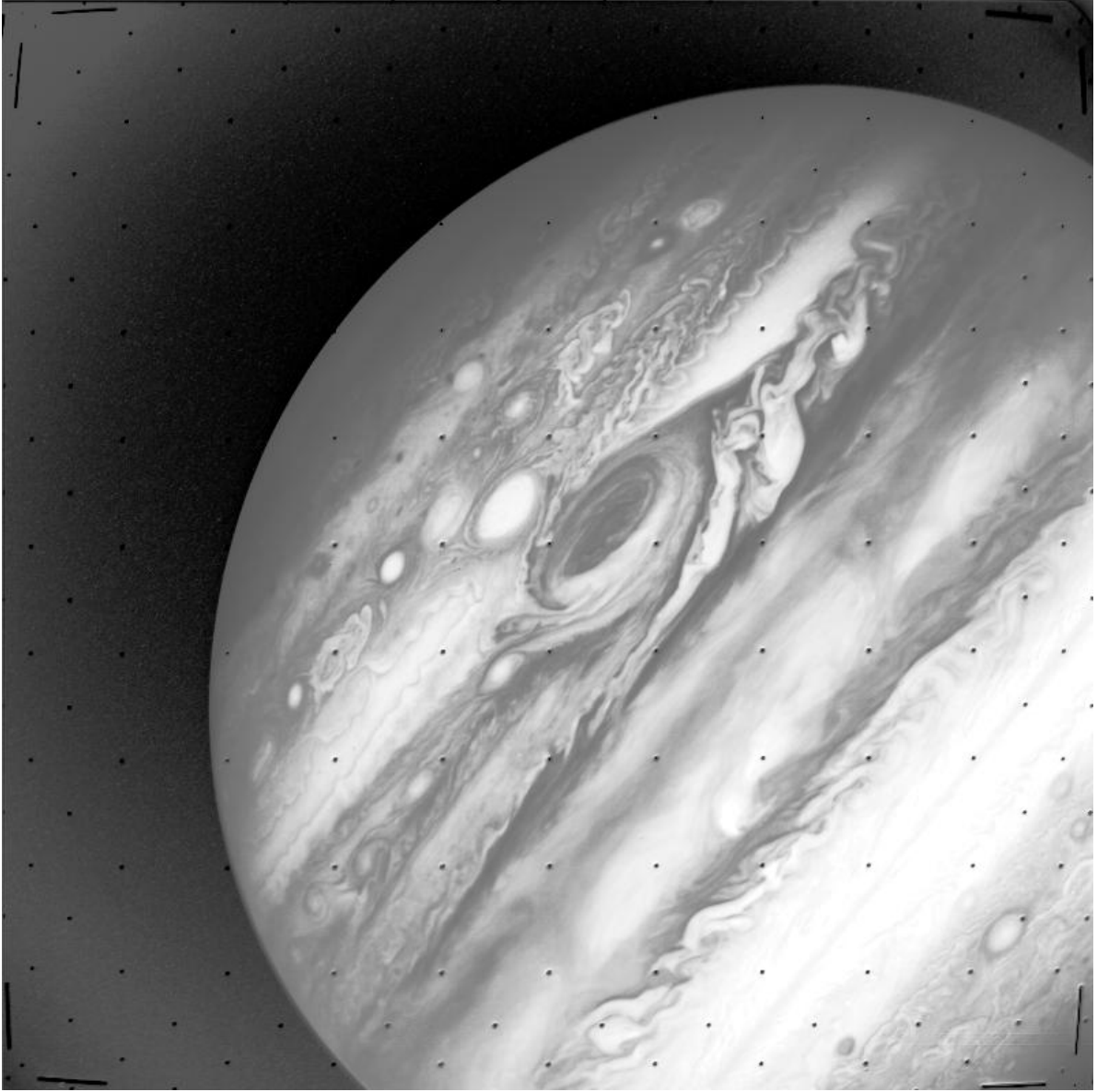


Figure 1: An example image of Jupiter

Our main objective in analyzing this dataset was not concerned with the goals of the scientist who took those images. Our main concern was to find interesting images and groups of images in the large collection. Although the collection is in digital, computer-readable form, the selection of an image based on, for example, its aesthetic properties, is extremely difficult as one has to look at each individual image. The time required to look at each image was on the order of 15 to 30 seconds that were needed to read the image from the CD-ROM and to decompress it. This was unacceptably slow since we had almost 30,000 images in the collection. Since aesthetic qualities of an image can not be easily quantified, we had no other alternatives than to inspect the whole collection. We recognize that many other criteria (besides aesthetic quality) might be as difficult to quantify. Selection and classification of a large collection of complex objects based on such criteria can benefit from the techniques that we have developed.

3 Design and Implementation of Icecube

Each object in a digital image archive is an image with associated attributes, such as information about what is in the image, a histogram of the pixel intensities, etc. Retrieving and sorting images using those attributes as keys can be accomplished as an enhancement of standard database technology with the ability to handle images. Such attempts are usually referred to as multi-media databases (see, e.g., Cantell, 1991). We want to index images by their contents to better use the ability of the human visual system to detect relationships and unusual features.

Our index of the images contains a small version of each image (a thumbnail) that is displayed in a window. We refer to the entire collection of thumbnails as a *contact sheet* by analogy to photography. The index is interactive and dynamic. The thumbnails can be rearranged in the contact sheet and can be sorted by the various attributes of each image. A subset of images can be selected either by hand (selecting individual thumbnails) or by their attributes.

An example window of our Interactive Icon Index appears in Figure 2.

This type of index is substantially different than one may find in a book, an art museum, or a library. The order of the thumbnails can be changed interactively by sorting according to various keys. The possibility of seeing many of the icons at the same time and being able to rearrange them by keys and in a contact sheet enables searching of the index by the contents of image itself.

Unlike on the printed page where layout is determined by going from left to

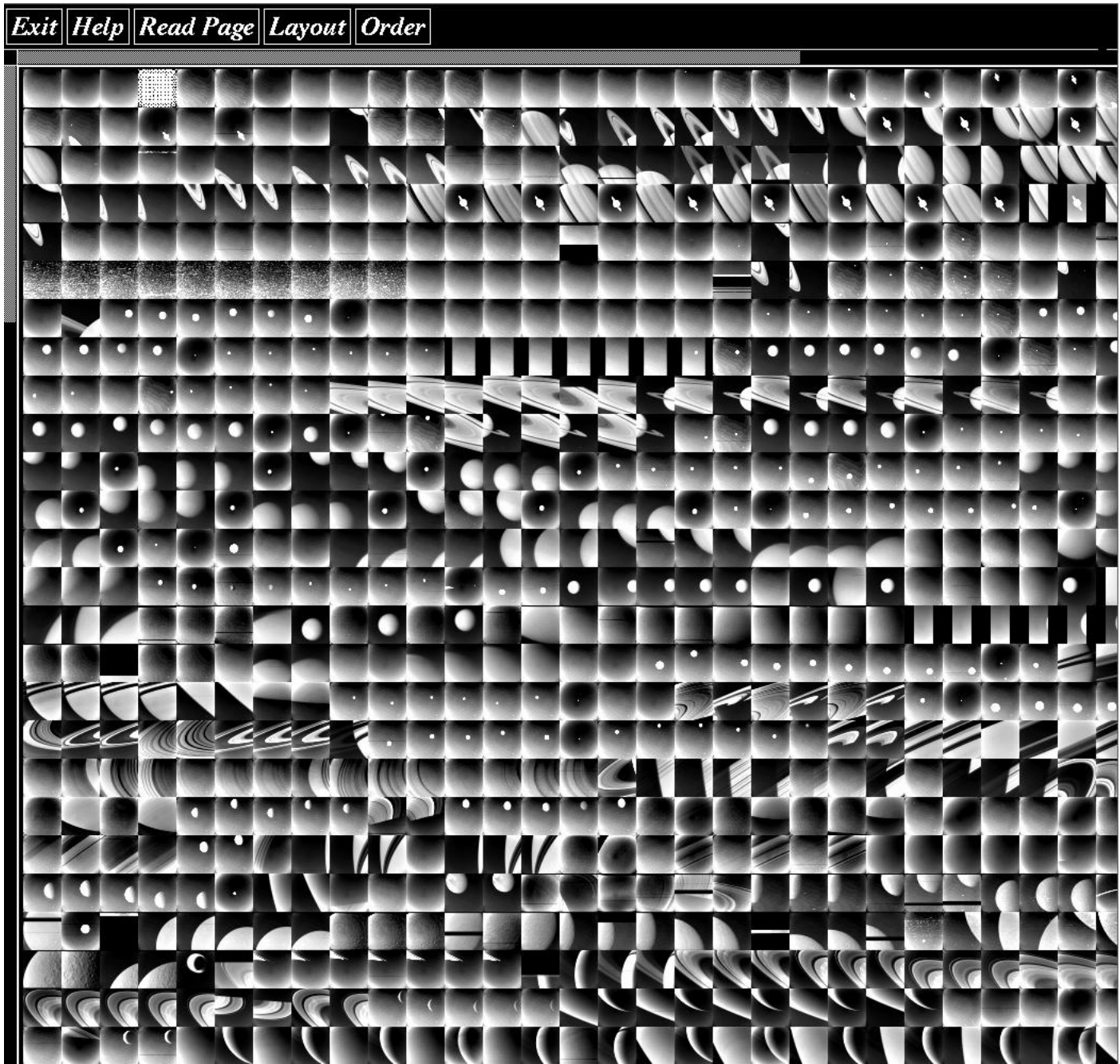


Figure 2: An example window of Interactive Icon Index with a contact sheet containing thumbnails of Saturn

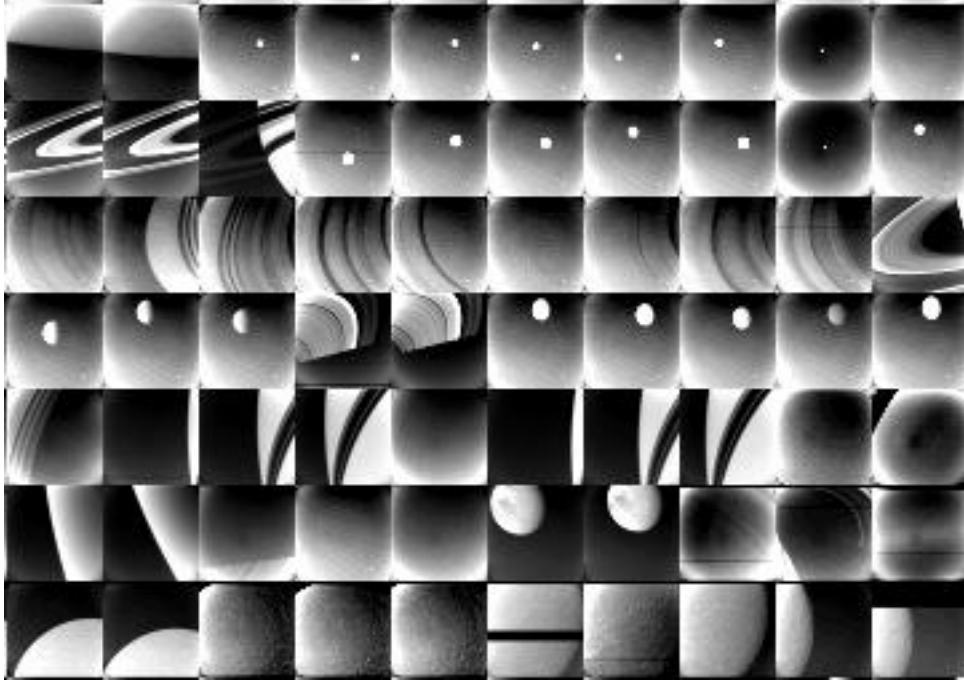


Figure 3: A closeup of several thumbnails of Saturn

right and then down to the next line, we can layout thumbnails in the contact sheet from left to right, top to bottom, or any other order. A particularly useful layout is a Peano curve, because the thumbnails that are close in linear order are also close in the two-dimensional contact sheet. If the linear order represents visual similarity of the thumbnails, the Peano layout divides the contact sheet into visually distinct spatial clusters of thumbnails. The Peano curve is a space filling curve that provides continuous mapping from an interval in R^1 to a box in R^2 (see Peano (1890), Hilbert (1891)). Two discrete examples of Peano curves (on $2^3 + 1 \times 2^3 + 1$ and $2^3 \times 2^3$ grids) are in Figure 4. The second curve is sometimes referred to as Hilbert curve. Those discrete curves can be easily defined on any $2^{2n+1} + 1 \times 2^{2n+1} + 1$ and $2^{2n+2} \times 2^{2n+2}$, $n = 0, \dots, \infty$ grids, respectively. As $n \rightarrow \infty$ the discrete curve converges to a continuous version of Peano curve. The desirable property of the curves is that all points in the domain (icons in the linear order) are also close in the range (icons on a contact sheet). Also, most icons that are close on the contact sheet are also close in the linear order.

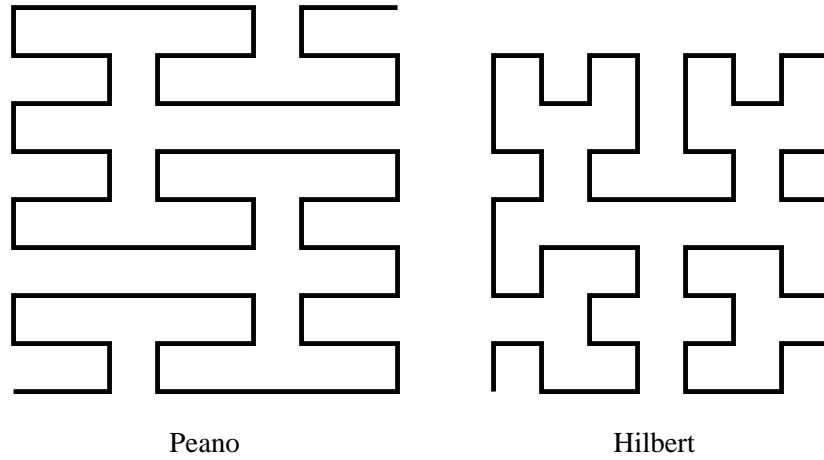


Figure 4: An example of space-filling curves

Creation of the thumbnails. An important aspect of having the collection of thumbnails as the image index is the possibility of determining the image content from the thumbnail. To create a thumbnail one has to reduce the original image in size without losing the image content (information that distinguishes the image from the other images in the collection). The direct subsampling of pixels from the original image would destroy the texture and other small scale features of the image. To preserve such features (if they are an important part of image content) we might need to design a special image size reduction technique. In our database of images the small scale features of images were not important, so we used simple pixel averages of the brightness adjusted image to produce the thumbnail. In another large collection a different visual representation might be useful. More generally, a number of visual representations might be precomputed and changed interactively to reveal different features of the database of images.

Hierarchical structure. The contact sheet of the collection of thumbnails is an image by itself and we could create thumbnails of the contact sheets to form an index of indices. The reduction of the contact sheet to the size of a thumbnail would effectively remove all visual information about the thumbnails within that contact sheet, so we need some other iconic representation of the contact sheet. A simple solution would be to choose (in some way) a thumbnail from the contact sheet and then use it as a thumbnail representation for that contact sheet in the index of contact sheets.

3.1 Design

Our basic goal is to have a tool which allows one to explore a large database of images. We think of certain operations as basic. First, one wishes to be able to issue queries as one would with a more traditional database or index: Find all instances of "X." Second, one would like to be able to browse through the database of images and interactively select items of interest: This is analogous to flipping the pages of a book until something catches one's fancy.

We anticipate additional goals in the future, particularly statistical ones, as our understanding of the database of images improves. A useful feature would be to cluster the images and then use the images corresponding to the cluster centers in our index. If the number of cluster centers is much smaller than the number of images we could inspect a very large database of images using a relatively small index.

3.2 Implementation

There are three stages involved in the use of the system. The first two stages are not interactive and must be completed before the interaction in the third stage.

1. Building the database of images
 - (a) Each image was brightness adjusted (equalized) because many of them were very dark and the details of the image could not be seen.
 - (b) Each image was converted to video format and recorded on the laser video disk device.
 - (c) A thumbnail of each image was created and entered into the contact sheet.
 - (d) Each image was inserted into a linear index giving its location on the video disk, its location on the contact sheet, its location on the CD ROMs, and the location of the associated data.
2. Preprocessing
 - (a) Attributes associated with each image were entered into a traditional database.

- (b) For each new study of the database of images it might be necessary to calculate new measures of interest and recreate the database (referred to in step (a)) associated with the database of images. Because of the amount of computation this step must be performed off-line. As an example, suppose that after initial study we decide that the pixel at location (5,13) has special importance. Since initially we had not stored the value of that pixel in our database we cannot use its value for interactive reorganization without rebuilding the dataset.

3. Interaction with the system

This is the most important stage and we will discuss it in more detail below.

Current interactive capabilities of the system include the ability to:

- display small copies of the images (the contact sheet);
- scroll the contact sheet;
- change the geometry (height or width) of contact sheet;
- select and display any individual image in about one half-second;
- select and play back any subsequence of images;
- sort the contact sheet by the value of any variable in the database;
- layout the thumbnails within the contact sheet in certain predetermined patterns;
- create a subset of the contact sheet from the selected thumbnails;
- manipulate any image using standard image processing software.

The simultaneous display of small copies of the images is achieved through the use of the thumbnails. We can see about 500 thumbnails simultaneously in a window. We can scroll the contact sheet to see all the images.

Selecting and displaying any individual image in about one half-second is accomplished by use of a SONY LVR5000 laser video disk recorder. We use a 19 inch color monitor, so the video version of an image is substantially larger than the tiny icon on the computer screen. The images are prerecorded in standard video format rather than digital format. The disk recorder (as opposed to a tape

recorder) allows us the nearly instantaneous access. The user merely points and clicks on the desired thumbnail. This is one of the most valuable features of our system. We can select an image (within the context of the nearby images), look at it and select again, as often as desired.

Selection and play back of any subsequence of images is accomplished in essentially the same way. The subsequence may be chosen by specifying the first and last thumbnail in the sequence or it may be specified by selecting each individual thumbnail.

Sorting of the contact sheet by the value of any predetermined variable is simple. We have, in addition to the images recorded on video disk, the database of information associated with each image. The sorting method that we use is *stable* (see, e.g. Knuth, 1973, page 4); a stable sorting method preserves the initial order of records with equal sorting keys. We have chosen to use a stable sort so that we may provide hierarchical sorting without sorting on multiple keys simultaneously.

Scrolling the contact sheet across the database of images is accomplished through standard scroll bars provided by the windowing system. Because our example database of images could naturally be subdivided according to individual planets, we were able to create separate pages (contact sheets) for each planet which could be handled more easily by our windowing system. Creation of an individual page can be done at the preprocessing stage once the database of images has been created.

We can layout the thumbnails within the contact sheet in certain predetermined patterns. The obvious patterns are across from left to right and down from top to bottom. We can interactively specify the number of columns or rows using such layouts. We found that one of the activities that was useful was to discover clusters of similar images. The (broken) linear layout did not accomplish the simple goal of having images that were near in the linear ordering near each other on the screen. We added the capability to lay out the images along a space-filling curve. We chose the "Peano" curve. See, e.g., Peano (1890) and Butz (1971). We have an option which draws the curve on top of the contact sheet so the user will know which images are near each other on the curve.

Creating a separate contact sheet from the selected thumbnails allows us to reduce the size of the database of images and concentrate on the interesting (selected) images. Having a smaller database of images decreases the time of the various operations on the contact sheet and improves interactive performance.

Manipulation of any image using standard image processing software is not as easy as it might seem. A large collection of images can not be stored online without

great expense. In uncompressed form, our example database of images would occupy approximately 30 gigabytes of hard disk storage. If stored in compressed form, inspection of individual images can be difficult and time consuming. Ability to find an image of interest is thus very limited unless all the characteristics required for search are stored in some database of image attributes.

The images in our example are stored in compressed form on multiple CD-ROMs. Access to a particular digital image requires that the correct CD-ROM be mounted on-line, that the compressed image be read off the CD-ROM, decompressed, and displayed using an image processing package. This requires approximately twenty seconds to access when the appropriate CD-ROM is mounted and substantially longer otherwise.

4 Future Capabilities and Related Work

In this section we describe other capabilities which could be implemented in **Iccube**. We also will attempt to relate **Iccube** to other systems that have been developed by computer scientists and others.

We intend to implement some but not all of the capabilities described below.

- move an image to an arbitrary position in the sequence;
- discard an image;
- select all thumbnails which are "similar" to a particular one;
- select all images with certain properties;
- selection and brushing of images using external statistical-graphical data exploration tool such as XGobi, see Swayne, Cook, and Buja (1991);
- perform layout according to two or more dimensional order (as opposed to a linear order we use now).

To provide flexibility similar to that of a text editor the ability to move an image to a particular location and the ability to discard an image would be useful.

The selection of images which are "similar" to a particular one requires a distance function between the images. Some of the approaches in defining a distance function between images are given in Barber et al. (1994) and Faloutsos

et al. (1993) and the references therein. Properties of potential interest include texture, color, shape, etc. One could also use a model such as the picture description language of Leung et al. (1994).

The selection of all images with certain properties could be used to select smaller subsets of a large database of images to improve interactive performance.

Since the data associated with each image could be in numeric form we could use statistical-graphical tools like XGobi (Swayne, Cook, and Buja, 1991) to produce sections, do brushing, and have a dynamically-linked thumbnail contact sheet that changes in accordance with those operations.

The contact sheet could be more than a two-dimensional rectangular array. We could assign a coordinate in R^d for each thumbnail and display the thumbnails according to their coordinates, e.g., perform a projection from R^d to the screen coordinates. There are obvious difficulties in trying to avoid overlap of the thumbnails on the screen.

A related iconographic display approach in Pickett and Grinstein (1988) describes a way to map multivariate data into an icon. They illustrate their approach with weather satellite imagery data. Each location (pixel) in the satellite image contains five numbers (intensities for five different values spectra). Using icons instead of pixels one can show all five images simultaneously. The particular icons described in the paper can not be used for data of very high dimension (e.g., images). The observations were assumed to be distributed on a plane, so that the icon location on a display was determined by the spatial location of observation. A generalization of iconographic display is in Ribarsky et al. (1994). The main idea of the later approach is glyph representation of an object in a database. Glyphs are graphical objects whose attributes - position, size, color, shape, etc. - are bound to data.

Problems concerned with distributed multimedia information on the Internet are somewhat different, though related to our problem. Bowman et al. (1994) consider concepts of indexing, browsing, and searching for the resources on the Internet. They propose that with such large and heterogeneous database as the Internet, the basic browsing and searching paradigms will break down and they propose to use scalable (hierarchical) content-based search mechanisms.

The single most important extension of Icecube would be the ability to handle other types of objects. The basic idea of simultaneous display and interactive rearrangement of a large collection of icons (each icon being a visual representation of an object in a large database of objects) in order to facilitate visual exploration does not depend on the nature of the objects in the database. To explore a specific

collection of objects it is essential to find an appropriate “content preserving” visual (iconic or thumbnail) representation for the particular collection of objects.

5 Summary

We perform exploratory analysis of a large image collection. Our paradigm is to place a small copy of each image (a thumbnail) in a virtual window (a contact sheet) and then interactively sort and layout the thumbnails in the contact sheet. We can detect the thumbnails of interest via visual inspection of the contact sheet and retrieve the original images for further analysis.

We found that in our example collection of images (images from the Voyager spacecraft) the thumbnails look similar to the original images and we can select the image of interest by inspecting the collection of thumbnails. We explore the image collection by visually inspecting the contact sheet. We can change the geometry of the contact sheet, the order of the thumbnails, and the way the thumbnails are layed out in the contact sheet. All those operations facilitate the ability to visually detect clusters of thumbnails in the contact sheet. Our hardware system allows us almost instantaneous retrieval of any particular image for viewing. We can select subsets of thumbnails for further, in depth, inspection. For example, our system allowed us to select “nice looking” images from this huge collection. Many quantitative characteristics for each image were recorded in the dataset and could be queried without even looking at a single image, but a qualitative query “find the best looking images” could not be reasonably answered without looking at all the images simultaneously.

We are currently working on a large collection of time series data and on a different collection of images and we expect that the concepts of visual indexing that we have introduced will carry over to other large collections of more general objects.

6 References

Barber, R., Flickner, M., Hafner, J., Lee, D., Niblack, W., Petkovic, D., Ashley, J., McConnell, T., Ho., J., Jang, J., Berkowitz, D., Yanker, P., Vo, M., Haas, D., Lassig, D., Tate, S., Chang, A., van Houten, P., Chang, J., Peterson, T., Lutrell, D., Sneden, M., Faust, P., Matteucci, C., Raynor, M.,

- Peters, R., Beck, W., and Witsett, J. (1994). Ultimedia Manager: Query by Image Content and its Applications. *Digest of Papers of the Spring COMPCON '94*, 424-429, San Francisco, CA.
- Bowman, C.M., Danzig, P.B., Manber, U., and Schwartz, M.F. (1994). Scalable Internet Resource Discovery: Research Problems and Applications. *Communications of the ACM*, **37**, 8, 98-107.
- Butz, A.R. (1971). Alternative Algorithm for Hilbert's Space-Filling Curve. *IEEE Transactions on Computers*, **C-20**, 424-426.
- Cantell, R.G.G. (1991). Special Issue on Next-generation Database Systems. *Comm. ACM*, **34**.
- Eliason, E., Davis, R., Martin, M. Avis, C., Hyon, J., Mehlman, B., McMacken, D. (1991). Archive of Digital Images from NASA's Voyager 1 and 2 Missions. NASA.
- Faloutsos, C., Flickner, M., Niblack, W, Petkovic, D., Equitz, W., and Barber, R. (1993). Efficient and Effective Querying by Image Content. *Research Report*. RJ 9453 (83074) August 3, 1993. IBM.
- Hilbert, D. (1891). Über die stetige Abbildung einer Linie auf ein Flächenstück. *Math. Ann.* **38**, 459-460. Also in *David Hilbert Gesammelte Abhandlungen* 1-2, Springer Verlag, 1970.
- Knuth, D.E. (1973). *The Art of Computer Programming*, Vol. 3, Sorting and Searching. Addison-Wesley, Reading.
- Leung, C.H.C., Hibler, J., and Mwara, N. (1994). Content-based Retrieval in Multimedia Databases. *Computer Graphics*, **28**, 24-28.
- Peano, G. (1890). Sur Une Courbe, Qui Remplit Toute Une Aire Plane. *Mathematische Annalen*, **36**, 157-160. English translation in *Selected Works of Guisepepe Peano*, (1973). Hubert C. Kennedy, Ed., University of Toronto Press.
- Pickett, R.M. and Grinstein, G.G. (1988). Iconographic Displays for Visualizing Multidimensional Data. *IEEE Conference on Systems, Man, and Cybernetics*, 514-519. IEEE Press.

Ribarsky, W., Ayers, E., Eble, J, and Mukherjea, S. (1994). Glyphmaker: Creating Customized Visualizations of Complex Data. *IEEE Computer*, **27**(7), 57-64.

Swayne, D.F., Cook, D., and Buja, A. (1991). XGobi: Interactive Dynamic Graphics in the X Window System with a Link to S. *ASA Proceedings of the Section on Statistical Graphics*, 1-8.