Dynamic Visualization in Modelling and Optimization of Ill Defined Problems Case Studies and Generalizations

William F. Eddy and Audris Mockus

Abstract

We consider visualization as a decision optimization tool in problems where the model and/or the objectives are not well defined. We investigate four specific problems representing different degrees of determination. The first problem concerns a smooth dynamic representation of data collected at fixed locations. In the example we want to minimize the deviations from a desired temperature over space and time. The second and third problems are the dynamic representation of observations in the form of averages over regions in space and time, and they are exemplified by epidemiological data. We are looking for spatial-temporal patterns that can suggest the most efficient ways of prevention and control. The fourth problem may be referred to as visual indexing. We perform exploratory analysis of a large collection of complex objects. The example is a dynamic index to a collection of 30,000 images. We search for the "most interesting" subsets of images via visual inspection of the index. In all cases we define appropriate techniques for the visual representation. We describe the software and hardware.

Key Words: Global Optimization, Interactive Graphics, Noninteractive Graphics, Dynamic Index

1 Introduction

For the description of well-defined global optimization problems see Horst and Pardalos (1995). An application of some elements of visualization in those problems was considered by Mockus and Mockus (1990).

In real-life applications we often encounter ill-defined problems. In this paper we call a problem ill-defined if it can not be accurately described in terms of finding the optimum of a function (as opposed to the conventional meaning of ill-posed problem which originates in the area of integral equations). In such cases we expect to improve the mathematical model and/or the objective after inspecting the results of modelling and optimization.

We think that in ill-defined problems visualization is an essential part of efficient problem solving. It seems that as the model and the objective specification becomes less precise visualization becomes more important. We use dynamic visualization as a modelling and optimization tool in problems where the complete formalization of the model and/or of the objectives is difficult or impossible.

We start the investigation from the relatively "well-defined" problem of minimization of spatial and temporal temperature deviations on a thin metal plate. Then we consider a quantity which varies as a smooth function in space and time but is measured as average values over a region in space-time. As an example we consider the incidence rate of the disease mumps. A natural objective is to minimize the impact of the disease using the minimum amount of resources but, unfortunately, it is not well-defined.

There was a common subproblem in both examples, namely to find the optimal smoothing function that represents the temperature distribution in the first example and the incidence rates in the second and third examples. Finding the optimal function can be transformed into a scalar optimization problem by taking a scalar measure of deviation from the best function, e.g., integrated squared error, maximum absolute error. However, those scalar measures do not represent all the information about the difference between the best and the considered function. Visualization of this difference conveys more information and helps to select a more appropriate solution.

In the fourth example both, the model and the objective, are not well defined. We are studying a large collection of digital images. To facilitate the search for the "most interesting" subsets of images we construct an index of the images in the collection. The index utilizes a small copy of each image (a “thumbnail”) to represent the full-size version. We consider various layout techniques, including the Peano curve. We investigate ways to speed-up the search.

In this paper we test our techniques using real data: temperature measurements in the first example, disease incidence rates in the second and third example, and a large collection of images in the last example. We use those data as "teaching" sets for our visualization techniques. Similar techniques, software, and hardware may also be used in the cases when the data is obtained via computer simulation.

In practical optimization problems a set of objective function arguments corresponds to parameters of the model that is being optimized. Visualizing that model in the optimization process can provide valuable insight and might lead to a revision of the objective function itself (after all, the objective function is just a scalar representation of optimality for the model of interest).

The common theoretical problem in all cases is to define the best visualization techniques to facilitate better formulation of the problem and of the objective. It is important to design specific software and hardware systems to process the data efficiently and to display the results. There are two ways to display the results: noninteractive (on a videotape) or interactive (on a computer screen). Each way has its advantages and disadvantages. Thus we consider both.

2 Interactive and Noninteractive Dynamic Graphics

2.1 Graphical Techniques

Graphical methods are widely used for the presentation of results. Here we consider visualization methods as a solution tool rather than as a way to present results. The main advantage of visualization methods is a possibility to communicate large amounts of information (“one picture is worth a thousand words”) in a short time period.

In optimization problems the objective function is a scalar or (sometimes) a vector representation of the “quality” of a very complicated model. One scalar value is, often, an incomplete specification of a model in scalar optimization problems. We, usually, are not certain how to relate different components of a vector-valued objective function or which one of many different Pareto optimal decisions to take in vector optimization problems. Communicating more information about the model in the process of optimization can be effectively handled using visualization methods. As the numeric optimization process is dynamic by nature so must be the visualization methods used in optimization problems.

If, on the other hand, it is assumed that all the information about the model is contained in the values of the objective function another approach can be taken. In such case the visualization should be performed on the values of objective function and values of its arguments, namely given function values f^i at points $x_1^i, \ldots, x_n^i$ visualize a set of vectors $(f^i, x_1^i, \ldots, x_n^i)$. This problem presents a serious challenge in any nontrivial case when $n > 2$. The visualization system described in the last example addresses this problem in a new way (see "**Icecube** and Optimization" subsection).

In this paper we consider interactive and noninteractive dynamic graphical displays. The choice of interactive or noninteractive graphical display depends on two factors:

- the amount of computer time needed to generate a graphical display
- the amount of time a user is willing to wait for the display to be generated

As computers become faster the advantages of interactive display seem to be obvious. But modelling and visualization tools become more complicated too, consuming all the increase in computing power.

The first two examples of this paper use noninteractive graphics. We discuss the new interactive dynamic graphics system in the third example. The fourth example is interactive except for the preprocessing stage.

2.2 Noninteractive Dynamic Graphics Equipment

We have to obtain special equipment, if we wish to use noninteractive dynamic graphics systems conveniently and economically. Therefore a potential user of those systems needs information not only about the algorithms and software, but about the hardware, too.

A noninteractive dynamic graphical display (animation) can be created by recording successive video frames, which are replayed at the rate of thirty frames per second (the NTSC video system used in the United States and Japan). We can play back such recording on any standard home VHS video cassette recorder (VCR). We need special video equipment (more sophisticated than a home VCR) to record individual video frames.

We record the generated images on a laser video disk recorder (LVR). Our model is a Sony LVR-5000A. This recorder uses twelve-inch write-once disks. Each side of each video disk holds roughly 43,000 frames which is equivalent to about 23 minutes of dynamic graphics at 30 frames/sec. This laser video disk recorder has some special capabilities:

First, we may access any individual frame almost instantaneously because the disk has random access as compared to sequential access tape device. Therefore the LVR does not limit the recording speed.

Second, we can control the LVR from the workstation over a standard RS-232 serial line using a simple protocol.

We may jump instantaneously between arbitrary frames in the playback mode. This is an additional benefit of the LVR, important for comparison. One can not do that with tape. Also we can play back the LVR at many different speeds both forward and backward.

We generate the images in a window of a computer screen. We use a video scan converter (Otto Graphics converter Model 9500 produced by Folsom Research) to convert the high resolution component video (input of a workstation display) to the NTSC system.

The video equipment is completed by a VHS video cassette recorder, which is used to transfer animations onto conventional VHS videotapes, and a color monitor that can display both RGB component video signal and NTSC composite video signal.

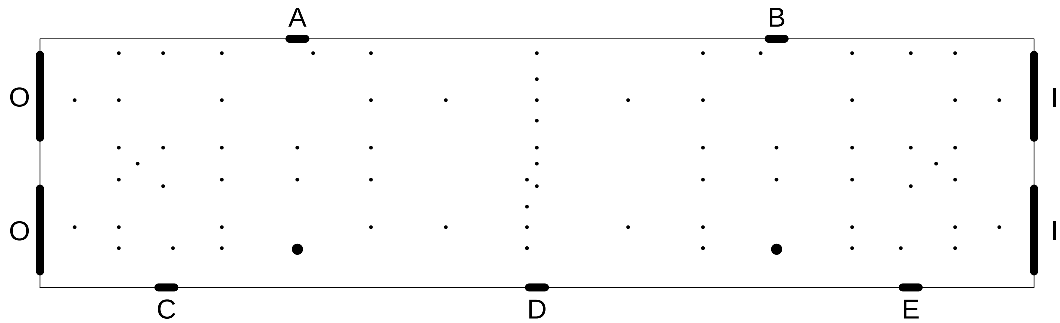

Figure 1: Schematic of the thin metal plate showing locations of thermocouples and electrical conductors

3 Dynamic Graphics in Manufacturing Process

3.1 Background

Here we describe a manufacturing process in general terms. See Eddy and Mockus (1993) for more details. The manufacturing process generates product on a continuous basis. We are investigating the spatial and temporal distribution of temperature over a thin rectangular metal plate. Maintaining a constant desired temperature over the plate is necessary to improve the quality of generated product. The product flows through a large number of small holes in the plate. A diagram of the plate is given in Figure 1.

The dots in the diagram show the locations of thermocouples used for gathering test data and will be described below. The holes which pass the product are not shown but are distributed nearly uniformly over the plate.

3.2 Objectives

It is essential to have the temperature maintained constant (in time) and uniform (in space) over the surface of the plate. Over a range of operating temperatures, the mass flow rate of the product is proportional to temperature. As a result low variability of the temperature increases uniformity of the product. The process fails to produce the desired product for temperatures outside the narrow operating range.

The temperature of the plate is affected by three main factors:

1. the flow of product through holes in the plate;

2. the (uncontrolled) directed flow of coolant across the discharge surface of the plate from front to back;

3. the (controlled) flow of electric current from right to left within the plate.

In the future we hope to describe the spatial distribution of temperature over the plate by use of the heat equation (a partial differential equation) constrained by measured temperatures at the thermocouples (taking account both of the measurement error and the incomplete specification the partial differential equations).

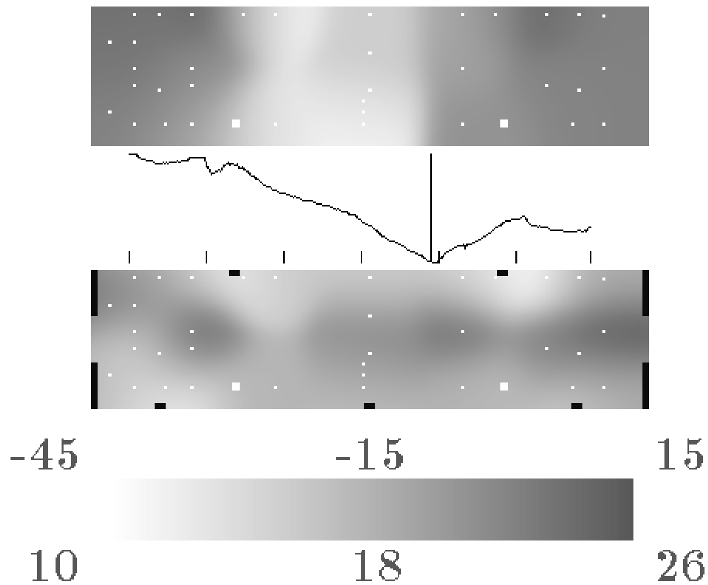

Figure 2: Heat distribution on metal plate

3.3 Data Collection

To gather data the process engineers designed a special metal plate with 74 thermocouples at various locations. Subject to engineering constraints, the locations were chosen so that the distance from any point on the plate to the nearest thermocouple was small. The result was not a regular pattern. Figure 1 uses dots to indicate the locations on the plate of the 74 thermocouples (the two large dots correspond to thermocouples used for the control of electric current).

Data were gathered automatically during continuous operation of the process under various test conditions from the 74 thermocouples. Data were gathered while a single unit of product was completed. The temperature observations were taken at 300 equally spaced time moments. This data gathering process was repeated several times under varying conditions resulting in several data sets.

3.4 Dynamic Display

An example of the graphical display is given in Figure 2. Because the spatial variation in the data is ten times as great as the temporal variation at any fixed point in space any dynamic display which does not remove the spatial variation would not reveal very much of the temporal variation. We combine two views of the plate: one static view showing a typical state and one dynamic view showing the deviations from that typical state.

There are several panels in the display. The lower of the two large panels is static. It displays the median (across time) temperature. The locations of the thermocouples used for fitting are indicated in white and the locations of the electrodes and current taps are indicated in black. The upper panel displays the deviation from that median. The middle panel is a time series plot of the median temperature (over the thermocouples at each time point). A distance between two tick marks corresponds to 50 time intervals (roughly 5 minutes).

In the dynamic display a vertical bar moves horizontally through the time series plot. The horizontal position of the bar indicates the frame number. A temperature color scale is given at the very bottom of the display. The numbers below the scale refer to the lower region of the display. To preserve confidentiality they have been subjected to a linear transformation. The numbers above the scale refer the residuals in the upper region and are given in degrees.

We are convinced that the most important information in this example is conveyed by the dynamic displays. Also, it seems the most persuasive way to convey this information to the process engineers.

3.5 Results

There were several decisions the process engineers took as as result of the dynamic graphics. First, they began tests to control the coolant flow across the metal plate to reduce the spatial variation of the temperature. These tests introduced devices to redirect the coolant flow across the plate.

Second, they ran a number of tests to find the optimal locations for the small number of control thermocouples to be used in the actual production process. As a result of those tests a control system based on six thermocouples was designed. Defining the optimal weights in the linear combination of the six thermocouples used to control the electric current is an optimization problem for the future.

The engineers have built a system to allow local control of the electric current across the plate. The system works by automatic shunting of current across three roughly equal disjoint regions of the plate. Tests are currently under way to evaluate various control strategies.

We see that in this case optimization of the manufacturing process was indirect. The effect was achieved mainly by better understanding of the system using the dynamic graphics[1]. We may apply the same dynamic graphic techniques for the direct optimization, too, using the data generated by computer simulation. It is the future task.

A direct optimization subproblem was to select such a smoothing procedure that the predicted temperatures would be close to the actual temperatures on the metal plate. The actual temperatures can be computed by solving heat transfer and electric current equations. Then we may optimize [2] the smoothing procedure. The work on this problem is ongoing. In the example we chose smoothing parameter by selecting the visually acceptable dynamic display.

4 Noninteractive Dynamic Maps in Epidemiological Model

4.1 Outline

Analysis of geographic data is important in many applications. Often, geographic data are reported as counts or averages over a region in space for an interval in time. We want to display the intensity function underlying the data using dynamic graphics. We will refer to the dynamic display of such function as a dynamic map. We think that this particular model (the intensity function) is, in many cases, the most appropriate way to visualize the spatial-temporal data. The aim of the display of the intensity function is to detect spatial-temporal patterns.

We will exemplify this problem with data on the mumps disease collected monthly in the United States from 1968 until 1988. Here we give a brief summary of the problem and the results. For a more detailed description of this example see Eddy and Mockus (1994).

[1] In our opinion this is as legitimate a way of optimization as any other.

[2] We search for a smoothing technique which satisfies the appropriate physical equations as closely as possible.

We begin with a description of the problem we are trying to solve, namely, estimation and display of smoothly varying function of space and time. We also provide a description of the data in the specific example we consider (mumps).

We describe in some detail the animation methods we use. We consider some statistical models for estimating a smooth function from averages over regions. Two different animations with varying degrees of smoothness in space and time are described.

4.2 Objectives

We consider a smoothly varying scalar function f with a three dimensional argument (x, y, t) where x and y indicate spatial coordinates and t indicate a time coordinate. The observations that we use to estimate the function f represent average values of the function over regions in space-time. Our goal is a dynamic display of the function f. Because we consider the arguments (x, y, t) to represent space and time we call such dynamyc display a dynamic map. The dynamic map is a sequence of maps (frames) shown one after the other in a rapid sequence. Each map (frame) is a rectangular array of color values corresponding to every pixel of the display. A pixel (picture element) is the smallest element of a display that can be individually manipulated. The color values indicate values of the function f, horizontal and vertical indexes of a pixel represent the two dimensions x and y, and the sequence number of a frame represents the time dimension t.

4.3 The Data

The mumps disease is of current public health interest in the United States in part because of a large outbreak which occurred in 1986-1987, primarily among unvaccinated adolescents and young adults in states without requirements for mumps vaccination.

The Centers for Disease Control in partnership with the Council of State and Territorial Epidemiologists (CSTE) operates the National Notifiable Diseases Surveillance System (NNDSS) to provide weekly information on the occurrence of diseases that are defined as "notifiable" by CSTE. Details concerning NNDSS can be found in Chorba et al. (1989).

The raw data consist of the number of cases of mumps reported from each state for each month for the period 1968-1988. Data are not available for some states and periods. One reason for this is that mumps has become reportable at differing times in the various states. Another reason is that small numbers of cases are less likely to be reported. We presume there are other reasons, also. The total data set contains 10,342 records. Note that 48 states times 21 years times 12 months per year yields 12,096 possible state-month combinations. There are 1787 missing observations, approximately 15% of those possible.

The raw counts were converted to incidence rates for each state (cases per 100,000 population) by dividing by the estimated population in units of 100,000 people. The state population estimates were obtained by linearly interpolating (or extrapolating) on a monthly basis from the 1970 and 1980 decennial census estimates of state population.

4.4 Video Display

4.4.1 Raw Data Animation

The entire dataset consists of 252 months. After several experiments we decided that displaying the data at the rate of two-thirds of one second per month was a reasonable compromise between the time

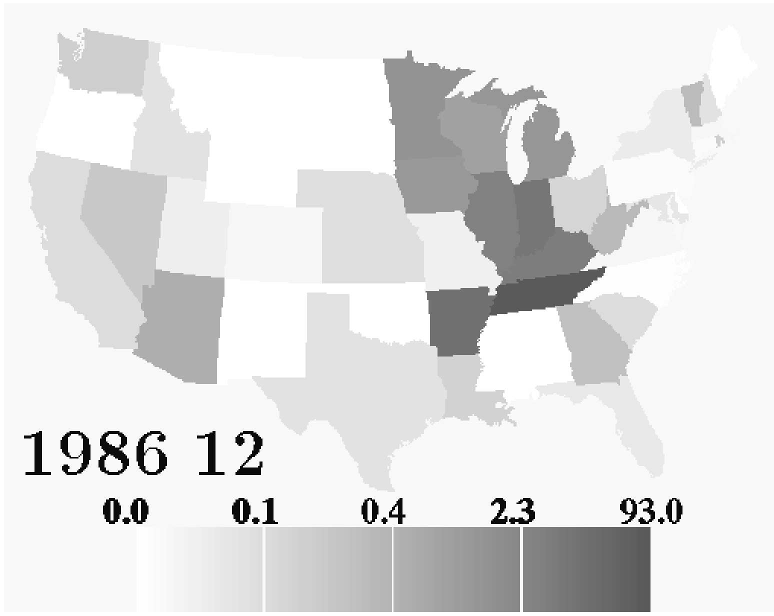

Figure 3: Raw incidence rates in December 1986

required to look at the entire data set and the apparent speed with which changes take place. In NTSC video (NTSC is the television signal used in the United States and Japan), 30 video frames (images) are shown in one second, so one month would be shown in 20 frames.

The speed of interactive animation would depend on the computer speed and the complexity of the animation algorithm, so the number of frames per month would vary accordingly. If the recording were done so that all identical frames were recorded and then the switch were made to the next month's data, the viewer would be distracted by the jumpiness of the resulting images.

Consequently, we choose to interpolate linearly between consecutive months. Precisely, the correctly colored maps for two consecutive months are calculated and then the intermediate maps are calculated by linear interpolation in the color scale. This approach gives substantially smoother appearance.

4.4.2 Smooth Animation

It seems natural to assume that the incidence rate for a disease is actually given by a smooth function over the entire United States. The incidence rates computed for each state are the integral of this unknown function over the respective state (divided by the area of the state). Our problem then is to estimate the smooth function given its integral over the different states. A discussion on smoothing methods appropriate in this case can be found in Mockus (1994) and Eddy and Mockus (1994). We sampled locations randomly in each state, assigned the state-month value to the sampled locations, and used a kernel smoothing technique to generate the map for each month.

As with the "raw" data, we interpolate linearly the intermediate frames between the monthly smoothed maps. Thus we smooth in space and in time using different techniques. A single frame corresponding to December 1986 is displayed in Figure 4.

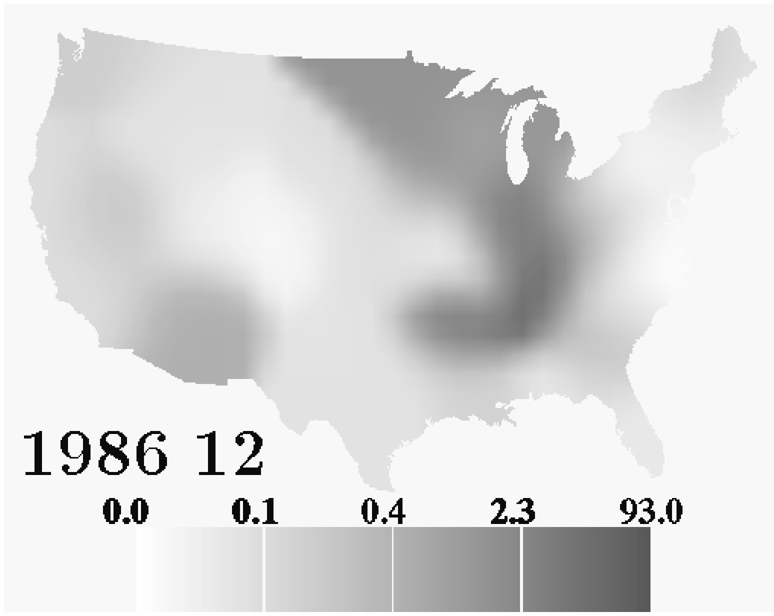

Figure 4: Smoothed incidence rates in December 1986

4.5 Effect of Dynamic Maps

Mumps in the US is a seasonal disease. The disease peaks in early spring, while lowest incidence rates can be observed in autumn. This can be in part explained by the school year because most of the cases are school age children. In the long term, the mumps disease had high incidence before the vaccination programs started at the end of 1960's. In the 1970's vaccination reduced the incidence rates by two orders of magnitude, leaving only a few cases per state per month.

We can observe annual periodicity in the incidence rate for mumps in both the raw data and the smoothed version. The periodic effect is most visible in the early years of the data set, before the widespread use of the mumps vaccine reduced the typical monthly incidence rate below .1 cases per 100,000 people. However, the effect can be seen throughout the data set, especially in the smoothed version.

It is difficult to see the geographic spread of mumps in the raw data; however, repeated viewing eventually allows one to make such an interpretation. The most interesting pattern can be observed in the winter of 1986-1987 in the states surrounding Illinois. The disease spreads from Illinois to Arkansas and Tennessee and in the subsequent winter when the disease spreads to all the neighboring states.

The vaccination programs were stopped in some states in the early 1980's and large outbreaks of the disease occurred in 1986-1987 and in 1989, primarily among unvaccinated adolescents and young adults in states without requirements for mumps vaccination. The explanation is well supported by the graph (see Figure 5) of the logarithm of the incidence rates in California and Wisconsin. We can see seasonal periodicity (high in spring and low in autumn) and an outbreak in Wisconsin in the second half of the eighties.

We see that a noninteractive dynamic graphics system is useful for indirect optimization by filtering the general patterns. We need an interactive dynamic graphics system for the direct optimization.

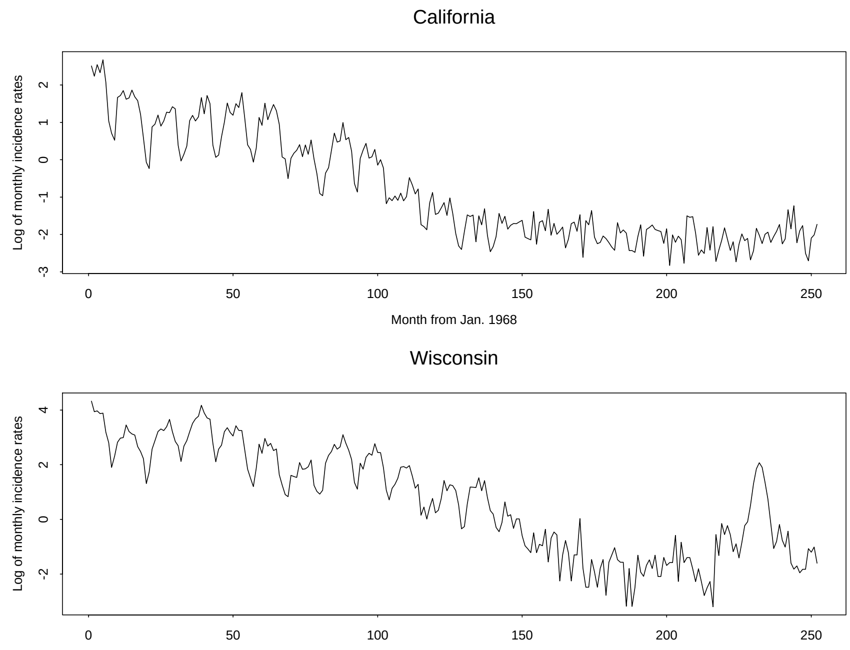

Figure 5: Log of the mumps monthly incidence rates versus months from Jan. 1968 to December 1988

5 Interactive Dynamic Maps

5.1 Background

In this section we extend the results of the previous example to a more general setting. We consider an interactive dynamic map of the vector quantities that were aggregated over regions in space and time. Unlike the noninteractive case, for the interactive graphics we need a much larger set of visualization and modelling tools.

As an example we consider 19 notifiable diseases for spatial and temporal trends. The reported data are given on a state-by-month basis (for the period 1962-1992). We convert reported cases to incidence rates before analyzing the data further. We produce interactive dynamic maps of those diseases trying to show spatial and temporal behavior at the same time. The objective and the data in this case are very similar to the mumps example. There are two differences: we visualize multiple quantities (diseases); we present an interactive approach.

5.2 Interactions

The interactive tools can be divided into several classes according to their functionality A brief discussion of each class follows.

5.2.1 Transformations

The usefulness of transformations in modelling is well known. The creation of a graphic display is a mapping from the data (or the model) to the range of display attribute values (e.g., range of pixels, colors, patterns, and glyphs). To emphasize different features in the data or in the model we may want to use appropriate transformations. In our system (it is currently under development) the user may choose different transformations: linear, general power, logarithmic, and rank.

5.2.2 Smoothing

We need smoothing methods to predict the value at a point for a quantity given as an average over the region. The smooth model is easier to perceive visually, especially with the large amounts of information in a dynamic display. The region boundaries may contain no essential visual information and their display may hide important features of the model or of the data.

Different prediction techniques may be appropriate for different data/model combinations. A simple method is to use a constant value over the whole region. The disadvantage is jumps at the boundaries of the regions. An alternative method is to interpolate so that the interpolant has correct averages over the regions. Such interpolated values are more difficult to compute and may be outside the range of observations. The method described in Eddy and Mockus (1994) is currently being used. We are adapting a kriging method (see, e.g., Mockus (1994)) and a histospline method (see, e.g., Dyn and Wahba (1982)) for our system.

5.2.3 Multiple Views

We call the mapping from the model to a particular display as a view. Interactive selection of a view facilitates visual inspection of different types of features present in the model and in the data.

A quantity of interest f is a function of three arguments x, y, and t corresponding to latitude, longitude, and time. The dynamic display can be described as a function d_f of three variables: horizontal and vertical offsets d_x and d_y (describing the pixel location) and a frame number d_t. The range of possible values for d_f includes available colors, patterns, glyphs, and the combinations of those named above. Hence, the view is a mapping from the quadruple (f, x, y, t) to (d_f, d_x, d_y, d_t). The mapping with variable values of d_t would correspond to a dynamic map. When the view has fixed value for d_t we get a static map. The view could be in the form of an XY plot where, for example, the quantity f is mapped to d_y and t is mapped to d_x. In such case we could map the remaining arguments x, y to the frame number t. We have implemented three types of views corresponding to a dynamic map, to a static map, and to a time series plot.

5.2.4 Sections and Aggregation

A view, as described in the previous section, is the mapping $(f, x, y, t) \to (d_f, d_x, d_y, d_t)$. A mapping that selects particular values of x, y, or t is a section, while the mapping that aggregates the values of f over subsets of x, y, or t is an aggregation. Section is a specific case of aggregation. For several quantities of interest $f_1, \ldots, f_n$, the section may select one of those quantities, and aggregation may be a function on a subset of the quantities.

We envision four types of aggregation functions: arithmetic (sum, variance); order (minimum, maximum, median); selection (section, several sections); composition of all of the above.

5.2.5 Display of Vector Quantities

In the mumps example we dealt with the incidence rates of one disease (mumps). If the incidence rates are reported for several diseases then one might be interested in detecting relationships between them. This raises the question of simultaneous display of multiple quantities that we try to address in our interactive dynamic graphics system.

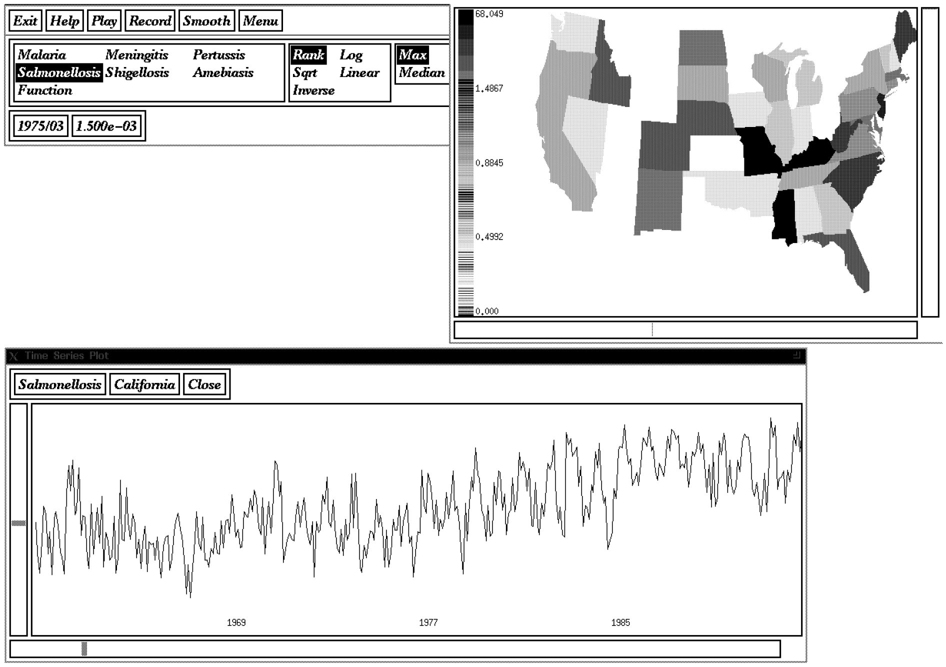

Figure 6: Control window and two views of the Interactive Map Animator

We may use for vector quantities the following forms of display:side by side; alternate in time; use different attributes (color, pattern, height, transparency) for different quantities; use aggregation to produce a single quantity.

5.2.6 Display of Missing Values

We implemented the following ways for handling missing values:

a) leave out:

  – use neutral color;
  – use background color;

b) fill in:

  – impute the value from available data (we use median for calculated value);
  – indicate that the value was imputed (we added a pattern to the calculated color).

5.3 Display

An example display of our system is in Figure 6.

The system consists of the main control window and view windows. The control window contains menus and selection lists. Modelling and transformation methods are controlled from the main window. In Figure 6 the data on the disease salmonellosis is selected and the rank transformation is being used.

The main window also contains the current date and time for the dynamic map view shown to the right. The bottom view contains a time series plot of the disease incidence in California. The state and the disease can be selected interactively using scrollbars at the bottom and at the left of the time series plot.

5.4 Optimization Potential

We consider the interactive dynamic maps as an integral part of interactive optimization of various space-time systems, such as epidemiological, ecological, economical, social, public relation, etc., using the collected data and/or data from computer simulation of those systems.

6 Interactive Icon Index

6.1 Outline

We are interested in interactive exploration of a large collection of complex objects, e.g, images, functions, and text, to name a few. We regard each object in the collection as an individual observation. To simplify the search of the "most interesting" object we construct an index of the objects in the collection. The index utilizes a small icon (a “thumbnail”) to represent each object. A large number of these thumbnails are laid out in a workstation window. We refer to our system as an **I**nteractive **I**con **I**ndex, hence $\mathbf{I}^3$, hence, **Icecube.**

We can interactively arrange and rearrange the thumbnails within the window. For example, we can order the thumbnails by the values of a function computed from them or by the values of data associated with each of them. We can access any individual object by simply clicking on the corresponding thumbnail. We can select the subsets of objects by their attributes and values. We can perform various operations on the selected subsets, e.g., animate, index, retrieve the original objects. We are currently extending our system to have a hierarchic index, a possibility to link different indexes, and a history function to facilitate navigation.

We regard our software as the beginning of the development of exploratory tools for studying collections of complicated objects as we routinely study batches of numbers. Our focus to date has been on developing a tool that will assist in selecting the "best" individual objects or “best” subsets for detailed inspection. We anticipate that in the future we may consider summary information and distributional properties of various aspects of the objects, as well as a definition of optimal classification and search strategies.

We describe the problem and our approach to its solution. We conclude with the description of a system optimizing layout strategies and list potential applications.

6.2 Objectives

We design an interactive dynamic index that would be appropriate for the complex objects more general than text.

We use the term "objectbase", to refer to a structured collection of objects. So we distinguish a database of general objects from a database of numerical and textual information. In our example we consider an objectbase containing approximately 30,000 images.

We are working on techniques of search for interesting objects and groups of objects in the objectbase. Inspection and manipulation of individual objects is just a part of this task. We are developing a hardware

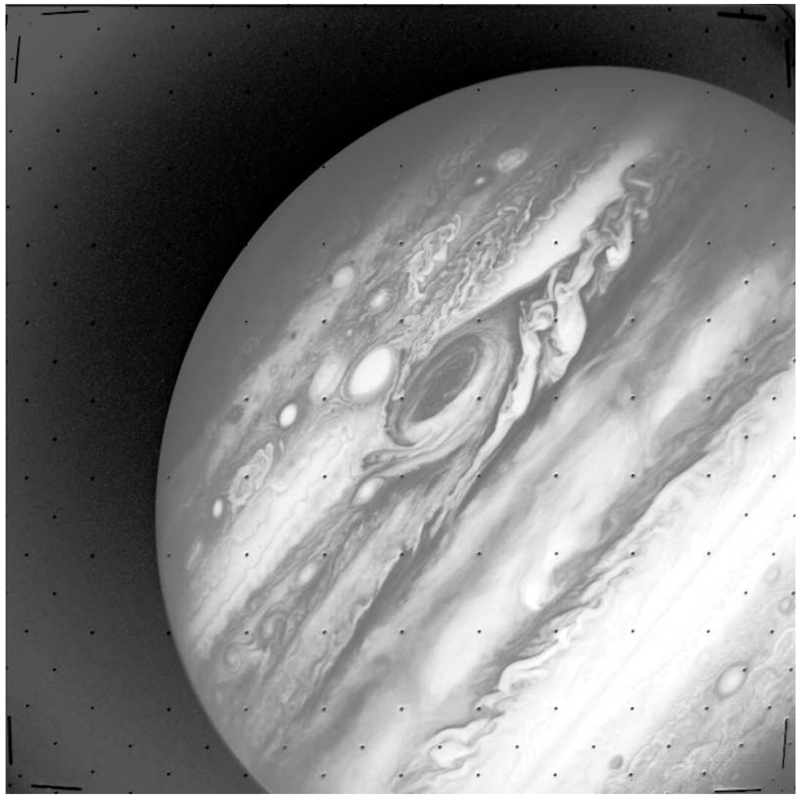

Figure 7: An example image of Jupiter

and software system to explore and to interact with the objectbase. The system 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 this particular kind of large objectbase.

We anticipate additional goals in the future, particularly the optimization ones, as our understanding of the objectbase improves. A useful feature would be to cluster the images in the objectbase 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 objectbase using a relatively small index.

6.3 The Data

This example is described in more detail in Eddy and Mockus (1994b). The example objectbase was obtained from NASA's Voyager Project and Planetary Data System. The data is described more completely in 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. An example image from the collection is given in Figure 7.

Each object in our example 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

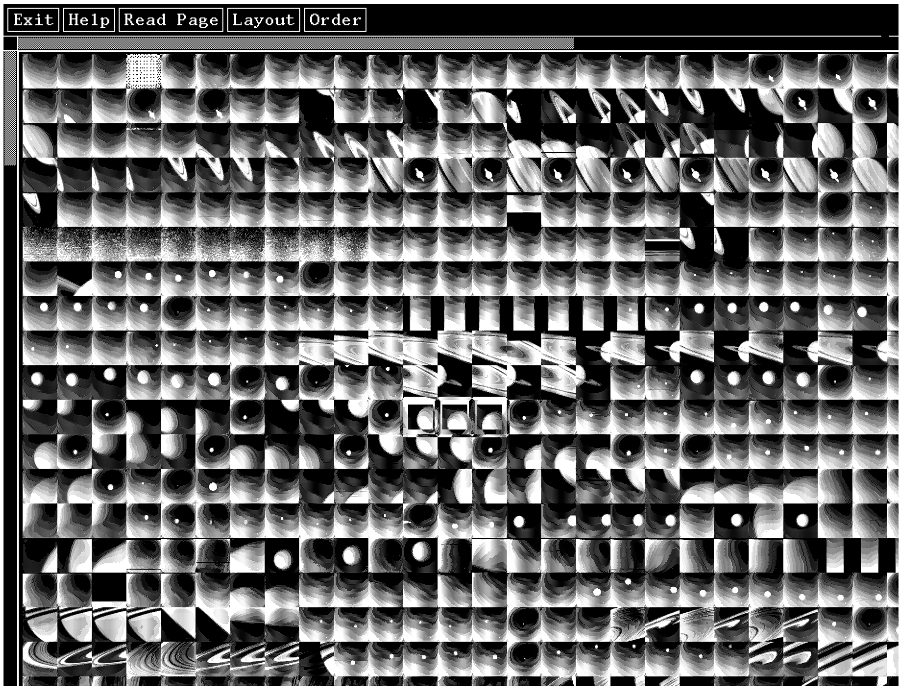

Figure 8: An example window of Interactive Icon Index with a contact sheet containing thumbnails of Saturn

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.

6.4 Display and Interactions

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 each image has. 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 8.

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.

Scrolling the contact sheet across the objectbase is accomplished through standard scroll bars provided by the windowing system. Because our example objectbase could naturally be subdivided according to major 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 objectbase has been created.

6.4.1 Ordering

We order all the thumbnails to facilitate fast layout and selection. A simple layout (see "Layout" subsection) is a function giving a position in the contact sheet for each order number. The selection of thumbnails by point-and-click requires an inverse function, namely, to identify a thumbnail located at a given position in a contact sheet. We may change the order interactively by sorting according to various keys (that represent image attributes) or by randomly assigning order numbers to each thumbnail. We have chosen to use a *stable* sort (see, e.g. Knuth, 1973, page 4) so that we may provide hierarchical sorting without sorting on multiple keys simultaneously.

6.4.2 Layout

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 to emphasize any possible periodicities in the sequence of thumbnails. We found it useful to discover clusters of similar images. The simple layouts did not accomplish the goal of "keeping the neighbors" [3]. Thus we added the capability to lay out the images along a space-filling curve. We chose the "Peano" curve (see Peano (1890) and Butz (1971)).

6.4.3 Selection

We prerecord the images in standard video format on a Laser Video Recorder (see the "Noninteractive Dynamic Graphics Equipment" subsection) rather than digital format. The video disk recorder (as opposed to reading the digital image from a CD-ROM) allows us nearly instantaneous access. The user merely points and clicks on the desired thumbnail. This is the single most valuable feature 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. The objects other than images might not be as convenient to display using video recording. Fast retrieval of a non-image type object might need other implementation. We may also can select and play back any subsequence of images, or create a separate contact sheet from the selected thumbnails.

[3] By "keeping the neighbors" we mean having images that were near in the linear ordering to be near to each other on the screen, too.

6.4.4 Creation of the Thumbnails

An important assumption concerning the collection of thumbnails as the object index is the possibility of determining the object content from the thumbnail. When indexing a collection of images 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). In our objectbase the small scale features of images were not important, so we used pixel averages of the brightness adjusted image to produce the thumbnail.

For the objects representing X-Y plots (e.g., f(x)), we can just scale the plot to fit into the thumbnail. For a scalar function of two arguments $f(x, y)$ we may encode the values of a function by colors of the corresponding pixels in a thumbnail.

6.4.5 Future Capabilities and Related Work

In this section we describe other capabilities which could be implemented in **Icecube** and attempt to relate **Icecube** to other systems.

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

- select all thumbnails which are "similar" to a particular one;
- selection and brushing of images using another **Icecube** or external statistical-graphical data exploration tool such as XGobi, see Swayne, Cook, and Buja (1991);
- hierarchy the thumbnails, when one thumbnail may represent several objects and may be expanded into several thumbnails;
- possibility to return to the previous views in interactive navigation.

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. One could also use a model such as the picture description language of Leung et al. (1994).

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 dynamically-linked thumbnail contact sheet that changes in accordance with those operations. Several **Icecubes** could be linked together so that selection of a subset in one **Icecube** would highlight the thumbnails corresponding to the same objects in the other **Icecube**.

A very large collection of objects (more than 100,000) might exceed computer display resources and only thumbnails corresponding to selected objects could be shown in a contact sheet. To select the objects we can classify them into a limited number of classes and only one thumbnail from each class would be shown in a contact sheet. Such higher level thumbnails would serve as navigation tools to be expanded into the subset of thumbnails corresponding to objects belonging to the same class. This expansion could be done in the same contact sheet or into another contact sheet by starting another **Icecube**. An inverse operation would be to collapse a selection of thumbnails into one class and leave only one thumbnail as a representative from that class.

6.5 Results

6.5.1 Application Potential

To appreciate the real application potential one must collaborate with the experts in the corresponding fields. Our experience lets us suppose that the system may be applied to such diverse areas as medical imaging, earth imaging, organic chemistry, astronomy, and other large databases. In many scientific fields large amounts of data are being collected. It becomes more and more important to be able to retrieve relevant data. We have immediate plans to use the **Icecube** in at least two applications briefly described below.

We are currently working on the analysis of large collections of functional magnetic resonance (fMRI) images of the brain obtained under various experimental conditions. The general purpose of the project is the localization of the various cognitive functional areas of the human brain. Due to the small signal to noise ratio in those experiments a large number of images is needed to show brain activation patterns. Imaging equipment can take approximately four images per second and an average experiment takes about one hour bringing the total number of images to about 15 thousand per experiment. We intend to use the **Icecube** to select and to process relevant images from different fMRI experiments.

The other project is to use **Icecube** for non-image data. We have obtained data from the Centers for Disease Control on 19 notifiable diseases in the United States (see sections "Noninteractive Dynamic Maps in Epidemiological Model" and "Interactive Dynamic Maps"). The data represents counts for 372 months and 48 continental states. In **Icecube** we can consider this data as 48×19 different time series of length 372, as $48 \times 19 \times 31$ different time series of length 12 (corresponding to one year), or as 372×19 different maps.

6.5.2 Icecube and Optimization

In the example we demonstrated feasibility of the concept of visual indexing. The effect of visual indexing can be appropriately estimated only by using the index while solving the real problems. Therefore we just list some of the features which could be most useful for optimization: visual classification; search for patterns; selection of subsets; extensibility.

We are developing an extension of **Icecube** to include new features valuable for image and object optimization [4].

We may apply the **Icecube** for the visualization of some conventional optimization problems, too, using, for example, the following strategies:

- Take thumbnails to represent a projection of the values of objective function in a particular direction. Each thumbnail would correspond to a different direction.
- Take thumbnails to represent objects being optimized that correspond to the argument values of the objective function.

7 Summary

We here demonstrated the importance of visualization techniques in modelling and optimization of decisions in four different examples. We consider that as a first step and we hope that those techniques will be widely applied in global and discrete optimization eventually.

[4]By image and object optimization we understand the search for the "most interesting" image(s) or object(s).

We considered noninteractive visualization methods in the first two examples and interactive methods in the remaining two. The first three examples describe attempts to use visualization when the objective is infinite dimensional, i.e., a function of three arguments (two-dimensions in space and time). In the first example the objective was temperature distribution over a metal plate, in the second and third examples the objective was disease incidence rates in the United states.

The last example presents a different approach in which the emphasis is based on the ability of human visual system to detect patterns. We simultaneously display a large number of "thumbnails" (each representing a complex object) on a computer window and rearrange them to help classify and/or select the objects of interest.

In the considered cases there was no obvious scalar or vector function that would describe the objective completely. One may use dynamic graphics as a natural, convenient and efficient way of the direct interaction between the real or simulated data and a human decision maker. That is most important while solving ill defined optimization problems.

8 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.

Butz, A.R. (1971). Alternative Algorithm for Hilbert's Space-Filling Curve. *IEEE Transactions on Computers,* **C-20,** 424-426.

Centers for Disease Control (1989). Mumps – United States, 1985-1988, *Morbidity and Mortality Weekly Report,* **38**(7): 101-5.

Cantell, R.G.G. (1991). Special Issue on Next-generation Database Systems. *Comm. ACM,* **34.**

Chorba, T.L., Berkelman, R.L., Safford, S.K., Gibbs, N.P., and Hull, H.F. (1989). Mandatory Reporting of Infectious Diseases by Clinicians. *Journal of the American Medical Association,* **262**: 3018-3019.

Dyn, N. and Wahba, G. (1982). On the Estimation of Functions of Several Variables from Aggregated Data. *SIAM J. Math. Anal.,* **13**:1 134-152.

Eddy, W.F. and Mockus, A. (1994). An Example of the Estimation and Display of a Smoothly Varying Function of Time and Space - The Incidence of the Disease Mumps. *Journal of the American Society for Information Science*. **45**(9): 686-693.

Eddy, W.F. and Mockus, A. (1994b). An Interactive Image Index. Technical Report 601. Department of Statistics, Carnegie Mellon University.

Eddy, W.F. and Mockus, A. (1993). An Example of Noninteractive Dynamic Graphics for Manufacturing Process Data. *International Statistical Review,* **61**(1): 81-95.

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.

Horst, R. and Pardalos P. (1995). *Handbook of Global Optimization.* Kluwer, Dordrecht/Boston/London.

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.

Mockus, A. (1994). Predicting a Space-Time Process from Aggregate Data Exemplified by the Animation of Mumps Disease. PhD thesis. Department of Statistics, Carnegie Mellon University.

Mockus, A. and Mockus, L. (1990). Designing Software for Global Optimization. *Informatica,* **1**(1): 71-88.

Peano, G. (1890). Sur Une Courbe, Qui Remplit Toute Une Aire Plane. *Mathematische Annalen,* **36,** 157-160. English translation in *Selected Works of Guiseppe Peano,* (1973). Hubert C. Kennedy, Ed., University of Toronto Press.

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.