# Beyond Query Languages For Geographic Databases:

## Data Cubes And Maps[1]

Andrew U. Frank
Department of Surveying Engineering
and
National Center for Geographic Information and Analysis
University of Maine
Orono ME 04469
(207) 581-2147 (Fax: (207) 581-2206)
FRANK@MECAN1.bitnet

## ABSTRACT

After several years of efforts to extend query languages for Geographic Information Systems (GIS), the limits of SQL-like languages for the selection of geographic data to be displayed in map form became evident. While SQL provides a means for selecting data upon well-specified criteria, it does not support exploratory access. Geo-scientists need methods for an explorative access to the data. Icons and direct manipulation are the appropriate user interface techniques. The traditional work methods of cartographers provided the concepts for the layer based GIS, where overlay (similar to the physical overlay of maps drawn on transparent material) is the central operation. The same source can be used to find visual metaphors to select icons and manipulations that direct the execution of operations. Two different concepts, one is 'data sets' the other 'maps', are involved and can be represented by icons for 'data cubes' and 'maps'. Operations can be succinctly expressed by dragging data cubes onto maps and the icons show - by 'repelling forces' - if the user is about to do an illegal or not meaningful operation. The interface and its operations are described and we conclude with open detail questions and mention a formal method that will be used to further develop the interface design to achieve consistency.

## 1. Introduction

Ease of access to the data in a Geographic Information System (GIS) is of utmost importance and determines the usability of the system. Attempts to extend standard query languages like SQL (ANSI/X3/H2 1985) to include sufficient facilities for selection and presentation of spatial data were not entirely satisfactory (Egenhofer 1991a). They require that the user understands the structure of the data stored and how they relate to his information needs, and is able to write a boolean expression to qualify the data to be selected. The presentation of the data retrieved is crucial for the effective comprehension of the information, and graphical, map like formats are indispensable for a GIS (Frank 1982). Most of the spatial extensions to query languages do not include tools for the user to direct the presentation of the data retrieved (Egenhofer 1991b, Egenhofer 1991c). A proposal for a query language with rich tools to select the rendering of the data as a map showed that such an extended language is very complex and the query expression a user had to type unacceptably long (Egenhofer and Frank 1988).
The current query languages, which are essentially retrieval languages, are applicable only for the few cases in which the user completely understands the structure of the data stored and needs only a very few data elements displayed in a default format. They may also be useful as a

[1] Funding from NSF for the NCGIA under grant SES 88-10917, from Intergraph Corp. and Digital Equipment Corp. is gratefully acknowledged.

standardized interface between programs that exploit spatial data and the database management system; however, they are inappropriate for *ad hoc* access to GIS data in situations when problems to be solved are poorly defined, the available data is not well understood, and highly trained users - geographers, planers, oceanographers or geologists - explore the data guided by their intuition and expertise. For these very important applications a tool supporting an **exploratory behavior** is necessary. Such tools exist for GIS' commercially and they allow the manipulation of thematic data sets and their visualization in the form of computer maps (Buttenfield and Ganter 1990), but their user interfaces are difficult to learn and cumbersome to use. The user interfaces are typically based on typed commands and systems provide users with very large sets of commands (more than 1000 in a commercial system (Antenucci and Brown 1989)) each with its own set of parameters and modifiers. Their use is not intuitive, requiring substantive planning from the users to translate the exploratory steps into system commands, thus breaking the direct link between the user and the data. Often a GIS specialist is necessary as a mediator between the scientist and the data to guide the data processing.

There is an urgent need for a visual, direct manipulation interface to support exploratory spatial data analysis and presentation by experts. Most notable is the widely publicized research efforts to detect Global Change and the enormous databases that will be necessary (Mounsey and Tomlinson 1988). Exploratory interfaces which support an unstructured experimentation with the data are requested by the user community (Smith and Frank 1990). Scientists ask for 'browsing' capabilities, which current generation GIS do not provide.

We present a user interface, based on a visual metaphor and direct manipulation (Shneiderman 1983, Shneiderman 1987). Observing the traditional, manual work methods of planners and cartographers provided the concepts for many GIS programs, which organize the data in thematic layers which are then overlaid over each other (Tomlin 1983, Tomlin 1989). We identified three kinds of objects: first, the data sets the user is interested in, second, the visualization parameters and third, the actual rendering in map form as the result of applying the visualization parameters to the data set.

The same source can be used to find visual metaphors for these concepts. The design is similar to the familiar desktop metaphor (Smith, et al. 1982), widely publicized by the Macintosh and now also the Windows user interface. The interface described here is substantially richer than the 'files and folders' on the Macintosh desktop. We use two different icons to differentiate the two different object types, namely 'data cubes' for data sets and 'maps' for the visualization parameters. Several specific operations are expressed by direct manipulation, i.e., a data set is rendered by dragging the data cube icon on top of the map icon. We also introduce reactive icons, which use visual 'force feedback' to indicate to the user that she is about to perform a meaningless or illegal operation.

New GIS interfaces often use 'windows, icons, menus and pointing device' (WIMP)(Nielson 1990) but their structure reveals that they are mostly a translation of command interfaces into menus, which - surprising to many - do not make learning or using the system substantially easier. Menus and icons help users select data and provide the necessary parameters for operations, but do not use a visually attractive and intuitive metaphor like the desktop with files and folders, which makes previous experience with the physical world applicable to the computer environment and therefore reduces the learning time needed (Kuhn and Frank 1991). Just translating commands into menu choices cannot reduce the effort of the user to translate from his concepts to the system commands.

The approach presented here is currently restricted to GIS' that organize their data in thematic layers and use the special concept of 'space with properties' as opposed to objects with attributes embedded in space (Frank 1990, Mark and Frank 1989).

In the next section we briefly characterize what we understand by exploratory behavior and exploration of geoscientific data. We then describe the workspace and manual methods of cartographers, planners and geographers. In Section 4 we detail the concept from the user's point of view and in Section 5 we detail the visualization of these concepts. In the conclusion we indicate the next steps necessary to develop a consistent user interface.

## 2. Exploratory Behavior

A scientist using a GIS to explore some data is looking for the 'unusual' and searches for 'interesting' facts. Visualization (Mandle 1990, McCormick, et al. 1987, McLaren 1989) is an eminent tool, because it brings to bear the human's outstanding ability to detect subtle visual patterns. Visual exploration benefits from a direct control of the user over the display. Scientific visualization has developed tools to render a data set and to allow the scientist to control the display parameters easily. The GIS environment can use some of the same tools, but must extend them to include the manipulation and combination of multiple data sets.

As a source of inspiration we can ask, how do humans cope with situations when they look for something 'unusual' or 'interesting' without further clues? Where is one exposed to such situations? A basic assumption in our interface design work is that the design of a human interface should be modelled after 'standard' human behavior, as it is found in non-computer problem solving and interaction. We learned that human beings are surprisingly flexible and willing to adapt to artificial environments, but at a high cost. The better alternative is the metaphorical use of familiar situations to structure new situations (Lakoff 1987, Lakoff and Johnson 1980). The Xerox Star and Apple Macintosh desktop metaphor is a prime example.

An example of exploratory behavior is observed if a human is confronted with a physical object (or collection of objects) of unknown purpose. How does one explore the object? One typically looks at it from different angles and perspectives, from far and near, one may even get a magnifying glass to see details, may change the illumination, or move the parts of the object to see how they could connect to each other and what their interaction could be. Detection of patterns leads to hypotheses, which are then tested by comparing them with observations, using the same tools. This behavioral paradigm is also found when geoscientists work with their data, most typically seen on a cartographer's or planner's desk.

## 3. The Geo-Scientist's Workspace

A planner or cartographer, like any geo-scientist, assembles data from different sources describing different aspects of an area. Some of this data is in tabular form, some is already mapped and only graphically available. This source material is combined to reveal 'interesting' patterns, which are then further examined. The major technique for combination is graphical overlay, where data sets are drafted on transparent material and then placed over each other (Chan and White 1987, White, et al. 1989). It is then possible to see boolean combinations of data values. In Figure 1c one can visually see that most of the forest is publicly owned, while most of the residential land is in private ownership. Often a new map is drawn as a result of the analysis of an overlay, and this map can then be used in the overlay process again.

... fig.1..

The method relies on a concept of space (Frank 1990, Mark and Frank 1989) where each point is characterized by some data values and data values describing the same properties for an area are arranged in a layer or coverage. This concept stresses the space as the organizational structure and the data that characterize space. This is different from a concept of space, where objects and their relations are located in space.

Computers have been used to formalize and improve this process. Dana Tomlin described it in his seminal thesis (Tomlin 1983) as a MAP algebra. The basic concept is the combination of spatially corresponding values in data sets by boolean or other operations.

The concept of map overlay is very powerful and has been successfully moved to computer implementations. There are widely distributed, but limited public domain systems (Sen and Radhakrishna 1988) used for teaching and powerful, feature laden systems for professional use (Aronson and Morehouse 1983, Dangermond 1988, Morehouse 1989). None of the implementations we know of uses a visualization of the objects and operations. A straightforward implementation displaying a desktop with maps that can be overlaid is obvious. It would be an improvement over the current command driven interfaces, but does not capture the appropriate operations well. A careful analysis of the overlay process reveals that the geo-scientist deals with two different kinds of objects, namely the data sets and their renderings as maps and the applicable operations are different.

## 4. User interface Concept Level

The design of the user interface must start with an analysis of the concepts used to structure the tasks from the user's view point. Unnecessarily complex, unclear structures at the concept level cannot be corrected with impressive icons or colorful menus. Clean concepts will lead to clean interfaces that are easy to learn. There is some evidence that learning a system depends, among other things, on the number of concepts a user has to master.

In this section we detail the three object concepts we identify to structure the task and explain the most important operations on them.

### 4.1 Data Sets

The geo-scientist collects data sets that describe some aspects of the world in which he is interested. These data sets contain a data value for each point in an area and are called thematic layers or coverages. Such data sets are produced by remote sensing instruments, but other data collection methods often result in coverages as well - from statistics that give the population data per census block to maps that show the soil types of an area. This concept of spatially registered thematic layers is so prevalent that it is often identified with the concept of a GIS and is visualized as a stack of layers as a symbol for a GIS (figure 2).

figure 2

Each data set has a title and some description of the data it contains. It covers a certain, typically rectangular area, where for each point in space a specific value is recorded. Conceptually these values are available for an infinite number of points but practically both space and attribute values are discretized (Goodchild 1990). Such values may be uniform for regular or irregular shaped areas. Additional descriptors are the scale on which the values are measured, the source of the data and its spatial resolution, the quality or lineage of the date, and so on.

### 4.2 Maps

Maps are renderings of data values in a graphical format. On a geoscientist's desk, maps are often used as the primary means of data collection and the difference between data values and graphic encoding is not always considered (Peuquet 1988). Operations on data values that yield new data values must be separated from mappings of data values onto graphical variables (Bertin 1983) for visualization.

Maps are a stable graphical product, that can be reproduced, stored and retrieved for later usage. Physical maps can be copied and some minor changes in their appearances can be made - small variations in scale, in color intensity, and so on. Major changes require a reconstruction of the map, either using the original data or using the existing map as source of data for the constructing of a new map.

### 4.3 Rendering Instructions

Maps can be seen from two different perspectives:

* The graphical presentation of some data in order to make them visually comprehensible,
* The map, separated from the data it represents, can be seen as an instruction (or procedure) of how to render a data set. It is a mapping (!) from data values to graphical representations.

The second point of view - even if abstract - is quite natural; we speak about 'a 7 1/2 minute quad' meaning the general appearance of such a map sheet without reference to a particular map sheet rendering a specific data set. A cartographer is often given a data set and an existing map and asked to draw the data in the style of the example - the map example stands for the abstract rendering instructions, which are more easily specified by example than by detailed instructions.

An example map determines how a data set is rendered. It controls the following aspects:
- The area and the scale used for graphical representation.
- The translation of data values to graphical variables (i.e., the map legend).
- The placement of map title and other auxiliary elements on a map (scale, north arrow, etc.).

Operations on the rendering instructions include detailed tools to control each aspect of the rendering process. The rendering instructions may include some graphical elements, from map neat lines, grid to backdrops, which provide context for the thematic data rendered. The rendering instructions, as a complex object, can be applied to one or several data sets to produce a graphical map.

# 5. User Interface Visualization Level

Once we have identified the concepts the user deals with, we have to decide on the visualization for them. We strongly prefer to visualize conceptual objects with icons and the operations as 'direct manipulation' either (Shneiderman 1983) of the objects or menu items. Therefore we use icons for data and for maps.

## 5.1. Data Cubes

One can visualize a single data set as a surface over an area, often shown as block diagrams (figure 3). The concrete example of a real valued function f (x,y) -> z can be generalized to data sets that tabulate values in dependence of more variables, while the domain can be any kind of value - nominal, ordinal, etc. (Stevens 1946).
figure 3, 4

Variables are associated with a default rendering, for example height data as a range of colors from green to brown, temperature data as a color range from blue to red, and so on, and these associations are global for all data sets with the same data type on the desk. Each data set is represented on the 'map desk' as a data cube icon (figure 4) with a title that describe the data set (similar to a file name on the desktop). It contains some description of the data. On the top surface of the cube, a much reduced rendering of the data values is given. This exploits the ability of humans to recognize maps from color and rough shape. This idea has been proposed for text files (Nielson 1990) but it was found that a reduced picture of a title page of a text is not a good representation; this is not a problem in the map domain.

Data cubes can be opened by double clicking, which allows access to a spreadsheet like editor for data values. Shift-clicking on the edge of a data cube turns it one quarter around this edge, which affects the surface on which the data are projected for rendering (figure 5).Data cubes can be stacked one atop of another to indicate that the data should be combined (figure 6).
Figure 5, 6

## 5.2. Maps

The rendering instructions are mapped as an icon and called maps. The visualization of a map is simply a map sheet (figure 7) as a flat surface in perspective. A data set (or a stack of data sets) is rendered by stacking it on a map that describes how it should be rendered(figure 8).
figure 7, 8

The combination of map and data sets stacked on it can be opened (by double-clicking) to see the rendition. Once a map is open for viewing, the editor used to influence the graphical rendering parameters is available and its effects can be directly seen (immediate feedback). For example we can use pan and zoom (Kuhn and Jackson 1990) to visualize the selection of area and the scale. Map legends can be 'active' such that the user can change them and thereby changes the underlying mapping from data values to graphical variables.The results of the editing are stored in the map (but do not affect the data in the data cube).

The link between the data set and the map is 'hot', meaning that changes in the data are immediately reflected in the map. So a user can use, in two different windows at the same time, the data set editor to change data values and the map editor to change the graphical representation.

It is also possible to construct a graphical map, which contains only the graphical rendering and not the original data. This is the equivalent of a traditional paper map (a map instance). The data are 'squished' into the map and irreversibly converted into graphics according to the transformation rules. Only very limited changes to the rendition of the map (comparable to the operations applicable to a paper map) are then possible.

Maps can be copied and the copies contain all the properties of the original map (but without the data) and can then be combined with other data sets. Maps can also be stacked and represent the sum of all the rendition rules and graphics of the individual parts.

### 5.3. The Mapper's Desk

The data cubes and the maps sit on the familiar desktop. It is an implementation question, if this is the standard desktop that is used for other operations, like editing files, or if this is a separate 'mapper's desk'. While the icons respond individually to the standard operations on the desktop, e.g., 'open', 'duplicate', and can be arranged in folders if desired, there are also the visual manipulations of 'stacking' and 'turning' that evoke specific operations:

Data cubes can be **stacked** to indicate that they should be combined along the common range (figure 6). Most obvious is the stacking using the x-y plane as the common base, but they can also be first **turned** and then stacked to construct a diagram which shows two different variables (for example temperature and rainfall) against the y axis - effectively constructing two overlaid profiles. Data cubes can be stacked on maps (figure 9), or maps can be stacked on each other - all evoking consistently the same general operation of 'overlay', but modified depending on the specific object type (polymorphism). The visualization of the metaphor serves both for visualization of objects as icons and for visualization of operations by actions (not by icons).

figure 9

The pasting of a reduced sketch of the data in a cube or map on its surface has already be mentioned as an mnemonic aid. It is a 'passive' information about the content of the object. There is also an opportunity for active feedback to the user when attempting illegal or meaningless operations. Each data cube's data cover a certain geographic area; if one tries to stack cubes which do not substantially overlap in the area, the result of the combination operation is meaningless. Cubes can warn the user of this problem, by sliding partially away from each other if one stacks cubes which are not covering a common area (figure 10). A similar behavior applies to the interaction of maps and data cubes. If the map is fixed in its geographic coverage - which it may or may not be - then only cubes that cover substantially the same area sit squarely on the map when put there. Lack of substantial overlap between areas is again indicated by the cube 'sliding' away (figure 11).

figure 10, 11

The 'squishing' of a data set into a map is an important feature to fix a map and detach it from potential changes in the data values. This may be important to assure that one can keep an unaltered copy of a map once produced. It is of even more importance to construct backdrop maps which are used to provide context to the map reader. A backdrop map is produced by stacking the relevant topics from the topographic data cubes (for example the river, county boundaries, highways and major city cubes) on a map. Next the rendering parameters are set to produce a black and white line map, in gray (say 60% screened). Then the data are 'squished' into the map to produce a map instance that is independent of the data cubes and can be used as a backdrop with other data cubes (figure 12). Such maps can be duplicated and overlaid (stacked) with other maps to combine rendition methods and a backdrop.

figure 12

## 6. Conclusions And Future Work

Exploration of the data collected in a GIS is not sufficiently supported by the current extensions of query languages. New tools are necessary to give the user control over the selection of data when the result is a large data collection and to allow him to direct the graphical presentation of

the data in map form. Scientific visualization has shown how the enormous potential of the human eye to detect patterns can be exploited to support exploratory data analysis. A direct and intuitive user interface is required to have minimal distraction of the expert.
Starting with the traditional methods of data analysis and map making, as used by planners, geologists, geographers and cartographers, we identify a source domain for a metaphor, which is the rendering of data sets in graphical form and the graphical overlay of such maps to construct boolean combinations. These methods have been implemented and generalized in computer programs, but the interfaces to the available systems are mostly command driven and require memorizing large numbers of commands with parameters. They are difficult to learn and lack the direct and intuitive character which is important for exploration.
We presented a visualization of this metaphor, introducing two icons for data sets (data cubes) and maps. Direct manipulation of these icons and interaction between the icons on the desktop allow one to express a number of operations beyond the move, open and copy operations, familiar from the Macintosh desktop. In addition, the icons can react and warn the user of inappropriate actions.
In order to confirm the suitability of the proposed interface we will write algebraic specifications for the source domain describing the concrete behavior of the traditional work of a cartographer. The target domain - the MAP algebra - is already formalized (Dorenbeck 1991). Then specifications for the visual interface will be drawn up . There should exist good mappings between these domains (specifically morphism) to make it easy for the user to manipulate and understand the abstract objects in terms of the concrete visualization (Kuhn and Frank 1991). These morphisms show how to design the details in the semantics of the operations and the visualization such that a consistent interface results.
The proposed interface has a major limitation in its reliance on the organization of data in layers or coverages, either in raster or in a vector data structure. It is not clear how a similar intuitive method can be provided for data that is structured in spatial objects with attributes. It is an open question if a single interface concept can cover both view points or if two related but separate methods are the best we can do.

## References


ANSI/X3/H2. 1985. *American National Standard Database Language SQL.* : American National Standards Database Committee.

J. Antenucci and K. Brown. 1989. An interview with Jack Dangermond. *URISA* 1 (1) : 50-59.

P. Aronson and S. Morehouse. 1983. "The ARC/INFO Map Library: A Design for a Digital Data Base". In *Sixth International Symposium on Automated Cartography.* Edited by B. S. Wellar. 372-382. Ottawa, Ontario, Canada: The Steering Committee for the Sixth International Symposium on Automated Cartography.

J. Bertin. 1983. *Semiology of Graphics.* Madison, WI: The University of Wisconsin Press.

B. P. Buttenfield and J. H. Ganter. 1990. "Visualization and GIS: What Should We See? What Might We Miss?". In *4th International Symposium on Spatial Data Handling.* Edited by K. Brassel. 307-316. Zurich, Switzerland: International Geographical Union IGU, Commission on Geographic Information Systems.

K. K. L. Chan and D. White. 1987. "Map Algebra: An Object Oriented Implementation". In *International Geographic Information Systems (IGIS) Symposium (IGIS'87).* Edited by R. T. Aangeenbrug and Y. M. Schiffman. 127-150. Arlington, Virginia:

J. Dangermond. 1988. "A Technical Architecture for GIS". In *GIS/LIS'88, Third Annual International Conference.* 561-570. San Antonio, Texas: ACSM, ASPRS, AAG, URISA.

C. Dorenbeck. 1991. Formalization of map algebra. Dipl. inf. thesis (unpublished, forthcomming) thesis, University of Bremen.

M. J. Egenhofer. 1991a. "Deficiencies of SQL for a GIS Query Language". In *Cognitive and Linguistic Aspects of Geographic Space.* Edited by D. M. Mark and A. U. Frank. (in press): Kluwer Academic Publishers.

M. J. Egenhofer. 1991b. Extending SQL for Graphical Display. *Cartography and Geographic Information Systems* (in press)

M. J. Egenhofer. 1991c. Interactin with Geographic Information Systems via Spatial Queries. *Journal of Visual Languages and Computing* 1 (4) : (in press).

M. J. Egenhofer and A. U. Frank. 1988. "Towards a Spatial Query Language: User Interface Considerations". In *14th International Conference on Very Large Data Bases*. Edited by D. DeWitt and F. Bancilhon. 124-133. Los Angeles, CA:

A. U. Frank. 1982. MAPQUERY: Database Query Language for Retrieval of Geometric Data and its Graphical Representation. *ACM SIG Computer Graphics* 16 (3) : 199 - 207.

A. U. Frank. 1990. "Spatial Concepts, Geometric Data Models and Data Structures". In *GIS Design Models and Functionality*. Edited by D. Maguire. Leicester, UK: Midlands Regional Research Laboratory, University of Leicester.

M. F. Goodchild. 1990. "A geographical perspective on spatial data models". In *GIS Design Models and Functionality*. Edited by D. Maguire. Leicester: Midlands Regional Research Laboratory.

W. Kuhn and A. U. Frank. 1991. "A Formalization of Metaphors and Image-Schemas in User Interfaces". In *Cognitive and Linguistic Aspects of Geographic Space*. Edited by D. M. Mark and A. U. Frank. in press. Dordrecht: Kluver Academic Press.

W. Kuhn and J. Jackson. 1990. *Metaphors for Interaction with Geographic Information Systems*. Report : University of Maine.

G. Lakoff. 1987. *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind.* Chicago, IL: University of Chicago Press.

G. Lakoff and M. Johnson. 1980. *Metaphors We Live By*. Chicago, IL: University of Chicago Press.

P. Mandle. 1990. "A Shell for a Geographical Data Visualization System (GDVS) on PC Using Some Commercial Available Programs". In *4th International Symposium on Spatial Data Handling*. Edited by K. Brassel. 443-449. Zurich, Switzerland: International Geographical Union IGU, Commission on Geographic Information Systems.

D. M. Mark and A. U. Frank. 1989. "Concepts of space and spatial language". In *Auto-Carto 9*. Edited by E. Anderson. 538-556. Baltimore, MA: ASPRS & ACSM.

B. H. McCormick, T. A. DeFanti and M. D. Brown. 1987. Visualization in Scientific Computing - Report of the Panel on Graphics, Image Processing and Workstation. *ACM SIGGRAPH Newsletter* 26 (6)

R. A. McLaren. 1989. "Visualization techniques and applications within GIS". In *Auto-Carto 9*. Edited by E. Anderson. 5 - 15. Baltimore, MA: ASPRS & ACSM.

S. Morehouse. 1989. "Architecture of Arc/Info". In *Auto-Carto 9*. Edited by E. Anderson. 266-277. Baltimore, MA: ASPRS & ACSM.

H. Mounsey and R. Tomlinson ed. 1988. *Building Databases for Global Science - Proceedings of the IGU Global Database Planning Project, Tylney Hall, Hampshire, UK.* London: Taylor & Francis.

J. Nielson. 1990. Traditional Dialogue Design Applied to Modern User Interfaces. *Communications of the ACM* 33 (10) : 109 - 118.

D. J. Peuquet. 1988. Representations of geographic space: toward a conceptual synthesis. 78 : 375-394.

L. Sen and A. V. Radhakrishna. 1988. "Expansion of the analytical capabilities of GIS: Integrating an optimization programming package. Package with OSU MAP-for-the-PC". In *Third International Symposium on Spatial Data Handling*. Edited by D. F. Marble. 269-292. Sydney, Australia: International Geographical Union, Commission on Geographical Data Sensing and Processing.

B. Shneiderman. 1983. Direct Manipulation: A Step Beyond Programming Languages. 16 (Aug.) : 57-69.

B. Shneiderman. 1987. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading, MA: Addison-Wesley.

D. C. Smith, E. Harslem, C. Irby, R. Kimball and B. Verplank. 1982. Designing the Star User Interface. 7 (4 (April))

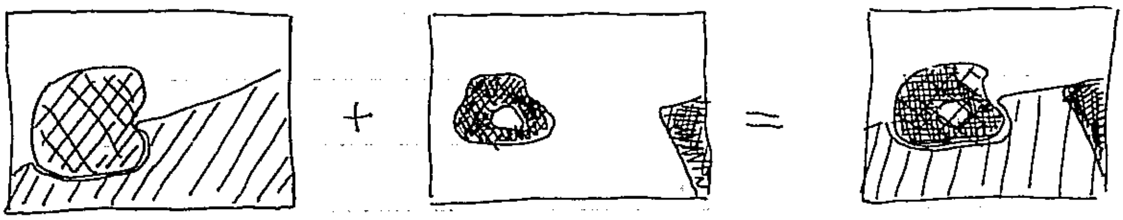
T. R. Smith and A. U. Frank. 1990. Very Large Spatial Databases - Report from the Specialist Meeting. *Journal of Visual Languages and Computing* 1 (3) : 291 - 309.

S. S. Stevens. 1946. On the theory of scales of measurement. *Science Magazine* 103

C. D. Tomlin. 1983. "A Map Algebra". In *Harvard Computer Graphics Conference.* Cambridge, MA:

C. D. Tomlin. 1989. *Geographic Information System and Cartographic Modeling.* New York: Prentice Hall.

D. White, K. Chan, M. Maizel and J. Corson-Rikert. 1989. "Polygon overlay to support point sample mapping: The national resources inventory". In *Auto-Carto 9.* Edited by E. Anderson. 384-390. Baltimore, MA: ASPRS & ACSM.



XXXX town
//// fores
[ ] agricultural

[cross-hatched] private
[ ] public

[hatched] town and private
XXXX town and public
[cross-hatched] forest and private
||||| forest and public
etc.

Fig 1 a. b. c.

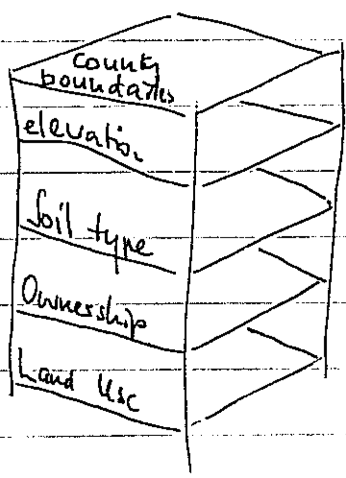



Fig 2

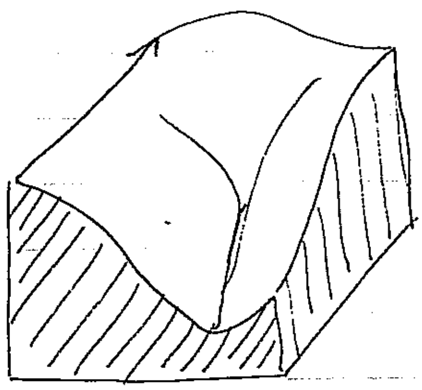


Fig 3 Blockdiagram

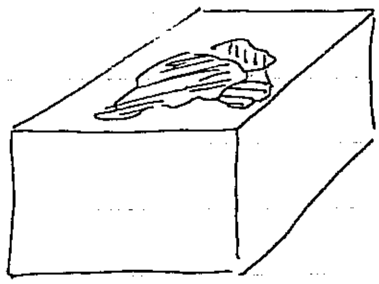



Fig 4 Data cube

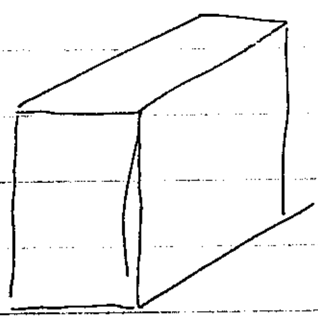



Fig 5 Data cube projected on y - Elevation plane

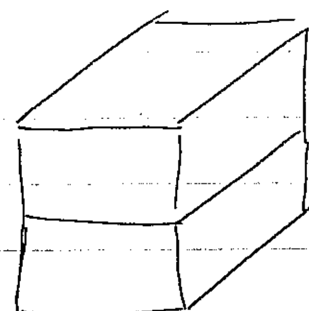



Fig 6 Two data cube stacked


Fig 7  A map icon

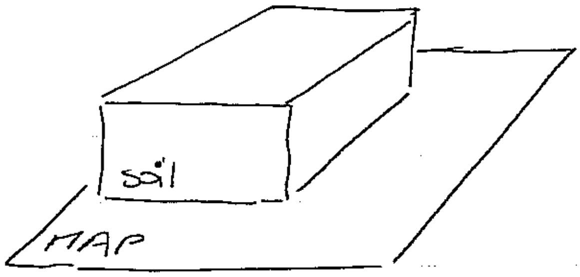



Fig 8  A data cube stacked on a map

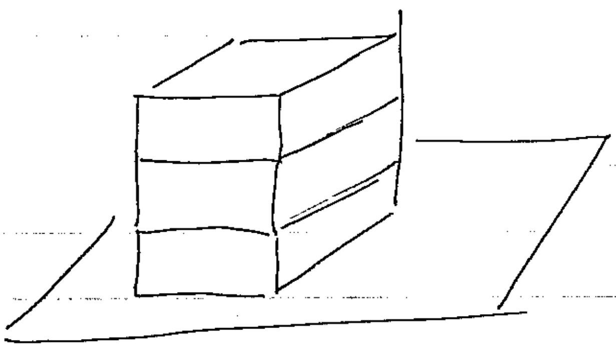


Fig 9  A set of data cubes stacked on a map

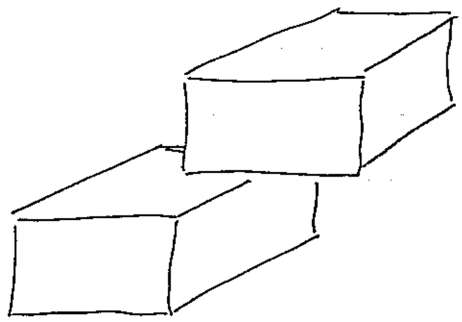


Fig 10 Cubes not covering same area

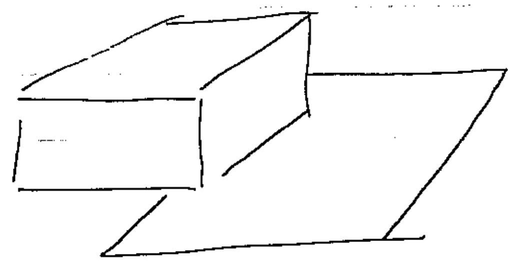


Fig 11 Cube not Covering area of map

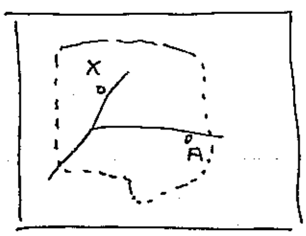



Fig 12 Back drop map

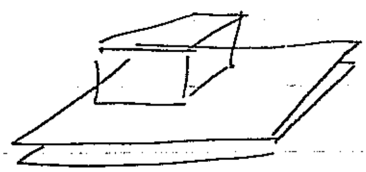


Fig 13 Two maps and one cube stacked