The Structure of the Information Visualization Design Space

Conference Paper · November 1997

DOI: 10.1109/INFVIS.1997.636792 · Source: IEEE Xplore

2 authors:

Stuart K. Card
Stanford University

Jock D. Mackinlay
Tableau Software

The Structure of the Information Visualization Design Space

Stuart K. Card and Jock Mackinlay
Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304 USA
{card, mackinlay}@parc.xerox.com

ABSTRACT
Research on information visualization has reached the place where a number of successful point designs have been proposed and a number of techniques of been discovered. It is now appropriate to begin to describe and analyze portions of the design space so as to understand the differences among designs and to suggest new possibilities. This paper proposes an organization of the information visualization literature and illustrates it with a series of examples. The result is a framework for designing new visualizations and augmenting existing designs.

Keywords
information visualization, taxonomy, design space, morphological analysis

INTRODUCTION

In recent years, information visualization, the computer assisted use of visual processing to gain understanding, has become a topic of significant development and research. Advances in this area are spurred on by increases in the power and availability of graphically agile computers and by advances in communications, particularly the growth of the World-Wide Web, which increases the amount of data available to a worker by orders of magnitude.

This new field has grown to a series of point designs that exploit the new graphical capabilities. It is typical for technologies to proceed at this point from point designs to abstractions that organize regions on the design space. In this paper, we propose a framework and illustrate the framework from samples from the literature. Our analysis builds on recent attempts to understand parts of the design space. Keller[1] lists techniques used in scientific visualization. Chuah and Roth[2] taxonomizes the tasks of information visualization. Shneiderman[3] proposes a data type by task matrix. Our analysis is closest in spirit to Tweedie's [4], who also starts from Bertin. Our analysis, starts from an expanded version of Bertin's [5, 6] and Mackinlay's [7] analysis of the semiotics of graphics and notes groups of techniques based on similarities of their data to visualization mappings.

SEMIOLOGY OF GRAPHICAL DATA COMMUNICATION

Graphics, according to Bertin[5], have at least two distinct uses, which should not be confused: first, as the means of communicating some information (in which case the communicator already understands this information in advance) and second, for graphical processing (in which case a person uses the manipulation and perception of graphical objects to solve a problem). As Bertin puts this latter use:

> *Graphics is the visual means of resolving logical problems.* [5, p. 16].

It is this visual processing use with which we are mostly concerned in information visualization, but interactive visual processing depends on a series of visual communication acts by the machine. These communicative acts map data and intent into visualization.

Data. Information visualization starts with information in the form of data. There are many forms that this data could take, from spreadsheets to the text of novels, but much of it can be represented as cases by variable arrays or can be transformed (perhaps with loss of information) into this form. Text, for example, can be used to compute document vectors, normalized vectors in a space with dimensionality as large as the number of words. Each document becomes a case and the direction of the vector becomes a variable. The different data types are important in their own right; text has its own characteristic operations, in fact the subcategories of patent text or financial report text have their own unique characteristics and potential unique operations on them, but for our purposes in this paper, we start with what can eventually be represented as the set of values taken on by a set of variables.

The major distinction we wish to make for data is whether their values are

Nominal (are only = or • to other values),
Ordered (obey a < relation), or are
Quantitative (can manipulated by arithmetic).

We notate these as **N**, **O**, and **Q** respectively. In a more detailed analysis, we would also note the cardinality of a variable, since one of the points of information visualization is to allow visual processing in regions of high cardinality. We distinguish subtypes of **Q** for intrinsically spatial variables and spatial variables that are actually geophysical coordinates. We also distinguish between data ***D*** that is in an original dataset and data ***D'*** that has been selected from this set and possibly transformed by some filter or recoding function **F**,

D --> F --> D' .

Visualizations. Visualizations are basically made from (1) Marks, (2) their Graphical Properties, and (3) elements requiring human Controlled Processing (such as text)[7]. Human visual processing works on two levels: automatic and controlled processing[8]. *Automatic processing*, which works on visual properties such as position and color, is highly parallel, but limited in power; *controlled processing*, which works on for example text, has powerful operations, but is limited in capacity. The distinction between these two types of capacity is important for visual design.

There are a limited set of available Graphical Properties, the basic set of which have been identified by Bertin [6] and expanded by Mackinlay [7] (and we expand further here): An elementary visual presentation consists of a set of *Marks* (which could be Points, Lines, Areas, Surfaces, or Volumes), a *Position* in space and time (the **X**, **Y** plane in classical graphics, but **X**, **Y**, **Z**, **T** 3D space plus time in information visualization), and a set of "*Retinal*" properties, such as Color and Size). We also add, following [7], the properties of *Connection* (notated "—") and *Enclosure* (notated "[]"). Thus, visualizations are composed from the following visual vocabulary:

Marks: (Point, Line, Area, Surface, Volume)
Automaticity Processed Graphical Properties
- *Position*: (**X**, **Y**, **Z**, **T**)
- *Retinal encodings*: (Color, Size, Shape, Gray-level, Orientation, Texture)
- *Connections*
- *Enclosure*

Controlled Processing Graphical Properties

To make comparisons easy, we use a common table format for these properties:

Data				*Automatic Properties*							*Controlled*
Variable	D	F	D'	X	Y	Z	T	R	—	[]	CP

The table is divided into the major sections of Data, Automatic Processing, and Controlled Processing by double lines. In the table we use codings (which we develop in context) summarized as follows:

Symbol	*Meaning*
Variable	Name of data dimension
D	Data Type ::= N (= Nominal), **O** (= Ordinal), **Q** (= Quantitative). **Q**X (= Quantitative and intrinsically spatial), **QXlon** (= Geographical) **NxN** (=Nominal set mapped to itself as in graphs)
F	Filter or function for recode data ::= f (= unspecified filter fn) **fs** (= sorting) **mds** (= multidimensional scaling > (= data reduction filter vis sliders and menus) **sl** (= slider)
D'	Recoded Data Type
XYZT	Position in space time
*****	Non-semantic use of space-time
R	Retinal properties ::= C (= Color), S (= Size)
----	Connection
[]	Enclosure
CP	Control Processing (tx)
P,L,A,S,V	Mark types ::= P (= Point), L (= Line), S (= Surface), A (= Area), V (= Volume)

Using these distinctions, we can see the major types of visualizations that have emerged.

SCIENTIFIC VISUALIZATION

Scientific visualization generally starts from data whose variables are intrinsically spatial. An example is Treinish's animated and very beautiful map of the earth's ozone layer[9](see Fig. 1). Because spatial and geographical variables are so frequent, we adopt the special notation of **QX** and **QY** for **Q** (Quantitative) variables that are intrinsically spatial and **QXlon** and **QYlat** for **Q** variables that are earth coordinates. Scientific visualizations, then, usually have mappings

QX-->X:P (i.e., a spatial quantitative variable is mapped into a position in **X**)

QY-->Y:P,

and often

QZ--> Z:P

as in Table 1 (We ignore for now the distinction between Cartesian and radial coordinates). Ozone density is mapped into the Retinal variable Color.

Fig. 1. Ozone concentration[9]

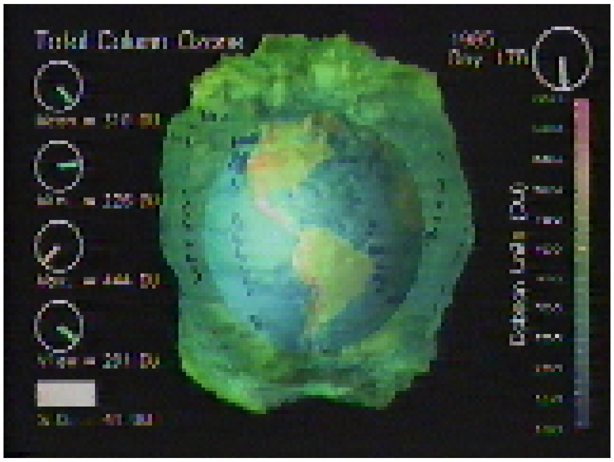

Table 1. Ozone visualization (See Fig. 1)

Name	D	F	D’	X	Y	Z	T	R	—	[]	CP
Lon.	QX lon	f	QX lon	P							
Lat.	QY lat	f	QY lat		P						
Height	QZ	f	QZ			P					
Ozone	Q		O					C			

GIS

GIS-based visualizations are similar to other scientific visualizations, but more specialized, with intrinsically geo-coordinate variables mapped onto **X** and **Y**:

QXlon-->X:P,
QYlat -->Y:P.

In Fig. 2, from Roth’s group (Fig. 2), this leaves the **Z** axis free and it is used for another data variable.

Q --> Z:L.

A final variable, Profit, is mapped onto a Retinal presentation, Color,

Q --> R:Color.

Fig. 2. SDM[10]

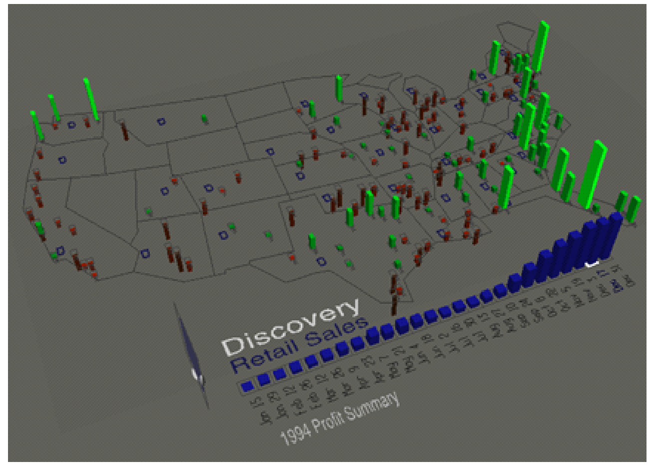

Table 2. SDM (See Fig. 2)

Name	D	F	D’	X	Y	Z	T	R	—	[]	CP
Long	QX lon		QX lon	P							
Lat	QY lat		QY lat		P						
Profit	Q		Q			L					
Region	N		N					C			

MULTI-DIMENSIONAL SCATTERGRAPHS

These type of visualizations take variables which are not intrinsically spatial and map them onto **X** and **Y**, e.g.,

Q --> X:P

Other (often ordinal) variables can be placed on sliders

O --> sl

and the sliders used to control the variables for filters. In the FilmFinder [11], Fig. 3, sliders (which appear as *sl* in Table 3), control filters on which cases are shown on the scattergraph. The sliders are separate, visual presentations, and so are separated from the rest of the table. The essence of the dynamic queries technique, of which the FilmFinder is an example, is that changes in the sliders have instantaneous effect on the items included. In this way, the effects of multiple variables with a large number of values can be taken into account without being coded as Retinal variables, keeping the display simple and easily interpretable.

Fig. 3. FilmFinder [11]

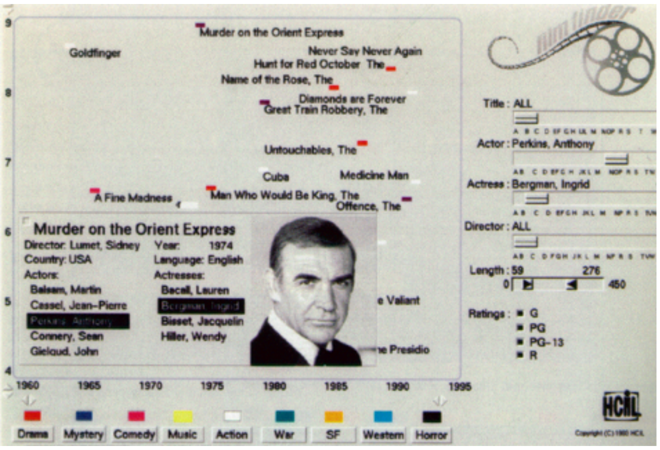

Table 3. FilmFinder (See Fig. 3)

Name	D	F	D'	X	Y	Z	T	R	—	[]	CP
Year	Q	>	Q	P							
Quality	Q	>	Q		P						
Type	N	>	N					C			
Title	O	sl>									
Actor	O	sl>									
Actress	O	sl>									
Director	O	sl>									
Length	Q	br>									
Rating	N	br>									

Feiner's Worlds-Within-Worlds technique is another way of showing higher dimension data. Three variables are mapped

$\mathbf{Q_1}$ **-->X**
$\mathbf{Q_2}$ **-->Y**, and
$\mathbf{Q_3}$ **--> Z**.

At the places in this coordinate system where other variables are to be examined, the x, y, z location is used as the origin of another coordinate system (for only a few points at a time) allowing another overloaded mapping

$\mathbf{Q_4}$ **-->X**
$\mathbf{Q_5}$ **-->Y**, and
$\mathbf{Q_6}$ **--> Z**.

The overlapped coordinate space essentially is a kind of details-on-demand display at the cost of occlusions in the original coordinate system. As in the dynamic queries technique, allowing the user to move rapidly in time through the first three variables increases the amount of the variable space that can be examined with a simple display.

Fig. 4. World within worlds[12]

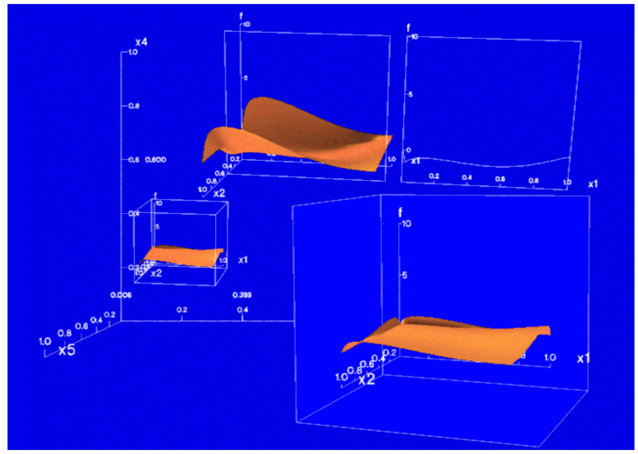

Table 4. Worlds Within Worlds (Fig. 5)

Name	D	F	D'	X	Y	Z	T	R	—	[]	CP
V1	Q	f	P	P							
V2	Q	f	P		P						
V3	Q	f	P			P					
V4	Q	f>	Q	S							
V5	Q	f>	Q		S						
V6	Q	f>	Q			S					
V7	Q		Q					C			

MULTI-DIMENSIONAL TABLES

Another interesting visualization for multidimensional data is to start with a matrix or table and to add visual properties to it directly as in the Table Lens [13]. The mixing of data and visualization makes it possible to drill down in place. This hybrid visualization produces an analysis quite different from previous analyses. In Fig. 6, *Team* is a Nominal variable and *Batting Ave*. a Quantitative variable, but we have used ?s in the Filter column because the variable may or may not be sorted, according to the action of the user. This is important because sorting determines whether **Y** in the diagram is a semantically meaningful ordering. In the Table Lens, the user is changing the particular mapping

Fig. 5. Table Lens [13]

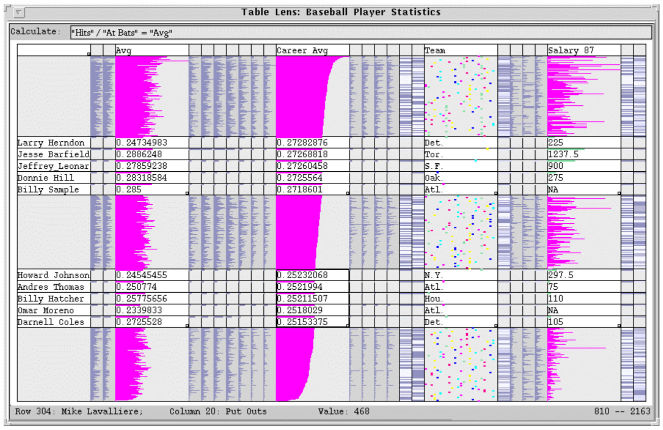

Table 5. Table Lens (Fig. 6)

Name	D	F	D'	X	Y	Z	T	R	—	[]	CP
Team	N	?s	N	L	?P			S			>A
Hits	Q	?s	Q	P	?P						>A

INFORMATION LANDSCAPES AND SPACES

Landscapes lay information out on a surface, typically the **XY** plane. Landscapes may be of several sorts: They may be a mapping of real geographical coordinates into the **XY** plane,

QXlon --> **X**:P
QYlat --> **Y**:P

as in Fig. 2. They could be just real spatial variables **QX**, **QY** into **X** and **Y** or they could be completely abstract mappings

Q --> **X**:P
Q --> **Y:**P.

If the mapping extends to

Q--> **Z**:P

Then we call it an information space. In Fig. 6, the actual physical trading room of the New York Stock exchange is mapped into an information space. The stock names on the kiosks are Ordinal variables mapped onto the **Z** axis (as in the physical room). But the visualization overloads stock volume, mapping it onto the **X** axis (actually onto a radial axis from the center of each kiosk, but we approximate here for simplification). The **Y** and **Z** locations of the bar are not semantic mappings. This bar is a Length mark, which is an instance of a Retinal Size encoding.

Fig. 6. New York Stock Exchange (Visible Decisions)

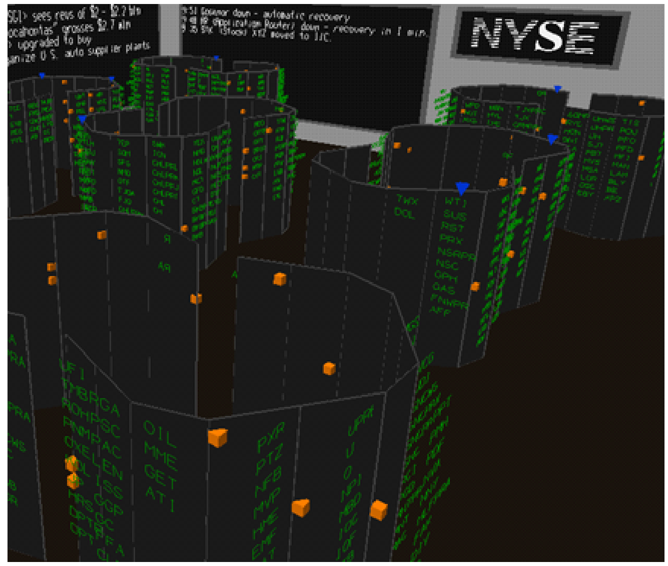

Table 6. New York Stock Exchange

Name	D	F	D'	X	Y	Z	T	R	—	[]	CP
Lat	Q		Q	P							
Lon	Q				P						
Name	T	s	O			P					
Volume	Q		Q	L	*	*		S			

NODE AND LINK DIAGRAMS

Node and link diagrams allow the encoding of linkage information between entities. They can be thought of as a mapping from a set of Nominal variables into itself {**NxN**}. These are then mapped into **XY** coordinates . One type of node and link diagram uses points that comes from physical space, then links them. Cox and Eick's mapping of Internet traffic [14] is an example of this type. As in landscapes, longitude and latitude are mapped onto **XY** (really onto a sphere)

QXlon --> **X**:P
QYlat --> **Y**:P

and the links are mapped onto Connection Lines:

NxN --> **X**:*, **Y**:*, Connection: Line.

Other variables are mapped onto Retinal properties.

Table 7. Internet traffic on earth (See Fig. 7)

Name	D	F	D'	X	Y	Z	T	R	—	[]	CP
Lon.	QX lon		Q	P							
Lat.	QY lat		Q		P						
Set	NxN	f	xxy	*	*				L		
Type	O		O					C			

TREES

But of course, the nodes for graphs do not have to be anchored in any spatial variable and the plane can be used merely as a substrate to keep the visual identities of the nodes distinct. An especially interesting visualization of this sort is the Hyperbolic Browser [15], where the space itself is distorted into hyperbolic coordinates (then projected back into the Euclidean plane). Since the space expands exponentially, it is a good place to lay out exponentially-expanding graphs, such as trees.

Table 8. Hyperbolic Browser (See Fig. 8)

Name	D	F	D'	X	Y	Z	T	R	—	[]	CP
Set	NxN	hb	xxy	*	*				L		

Trees can also be visualized as nested enclosures

NxN --> **X**:*, Y**:*,** Enclosure: *

Shneiderman and colleagues [16] have done a space-filling form of enclosure tree called Tree-Maps. At one level in a tree, the children of a node divide up the **X** dimension of the visualization, at the next level they divide up the Y dimension of the node in which they are enclosed. The division proceeds alternating between **X** and **Y** until the leaves of the tree are reached. This method uses all of the space. An example showing the use of space by the Mac filing system is shown in Fig. 9. The problem is that the same variable is mapped onto two different position presentations, each half of the time

Q --> **X** (half time)

Q --> **Y** (half time)

giving an inconsistent mapping and prohibiting the user from forming an easy image. What the user should be able to take from the image is essentially Retinal: Size coding, but the same Size can have many different visual manifestations, each with a different aspect ratio. Thus the space-filling property of the visualization comes at a cost.

Fig. 7. Internet traffic [14]

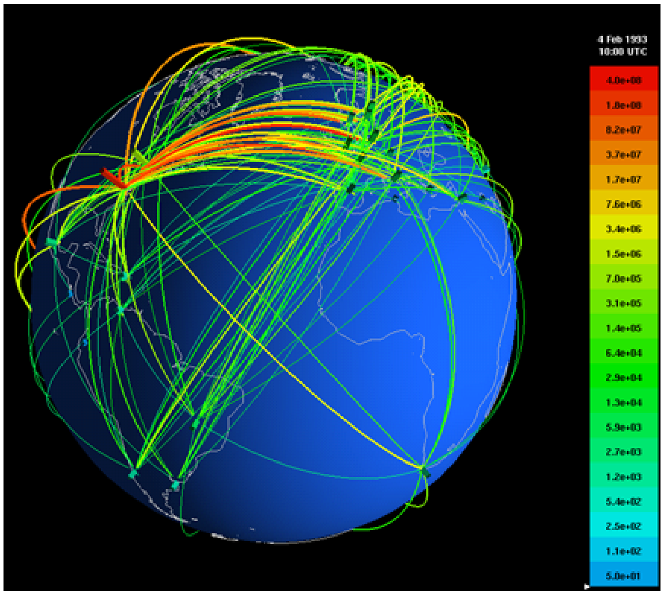

Fig. 8. Hyperbolic browser [15]

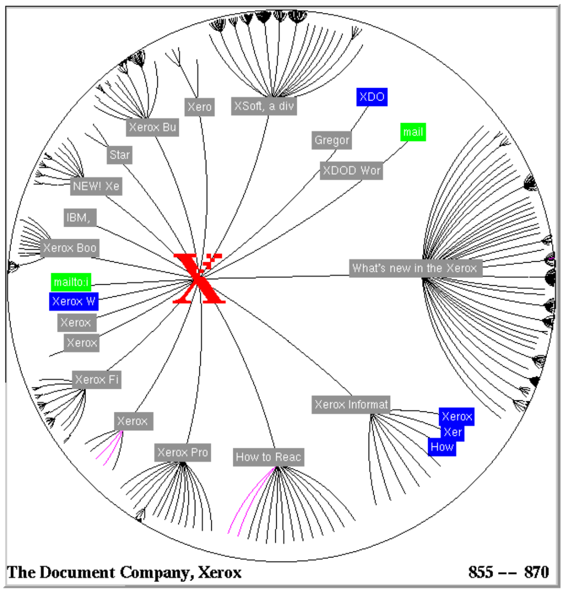

Fig. 9. [16]

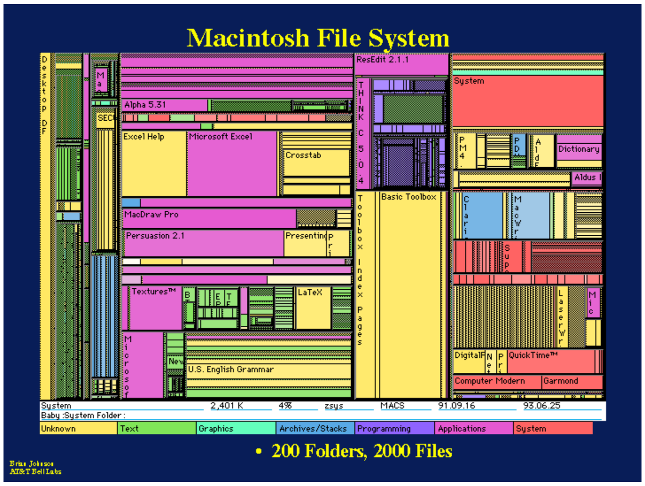

Table 9. Tree-Map (See Fig. 9)

Name	D	F	D'	X	Y	Z	T	R	—	[]	CP
Files	NxN	tree	xxy	*	*					*	
Type	N		N					C			
Size	Q		Q	L/2 <--	L/2 -->			(S)			

Cone trees try to handle the problem that trees are large in width because they exponentially increase in this dimension as a function of depth. The cone tree solution is to wrap the width around in a circle, then use time to allow the user to make the relevant part of the circle come forward.

NxN --> **X**:*, **Y**:**P**, **Z**:*, Connection: Line.

This visualization uses space to establish position in the tree. It thereby has a more consistent mapping than the Tree-Map, but at the cost of spending space to establish the structure. The Tree-Map spends almost all of its space on content.

Fig. 10. Cone tree[17]

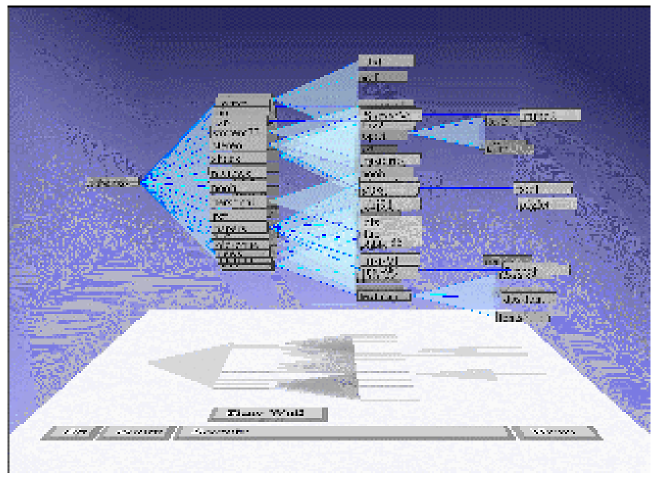

Table 10. Cone-tree (See Fig. 10)

Name	D	F	D'	X	Y	Z	T	R	—	[]	CP
Files	NxN	tree	xyz	*	P	*			L		tx

SPECIAL DATA TRANSFORMS: TEXT

We have discussed some of the main classes of visualizations. But an important point to make is that techniques for transforming data types into the data forms that can be mapped into these visualizations are also important. A case in point is text. Text itself can, of course, be visualized directly.

Q --> **CP**.

But this does not work for a large mount of text and the text has to be processed with Controlled instead of Automatic processing from a human point of view. One approach to visualizing large amounts of text is to map it line by line onto long strips

Q --> **Y**:P

as Eick and colleagues have done for English text and program code in SeeSoft [18]. By means of a slider, those lines of text having certain properties can be turned off or on as with the dynamic queries work. For example Fig.

11 shows the entire text of the Jungle Book. Lines mentioning different characters are rendered in different colors. Various combinations of characters can be rendered together by means of the slider.

Fig. 11. SeeSoft[18]

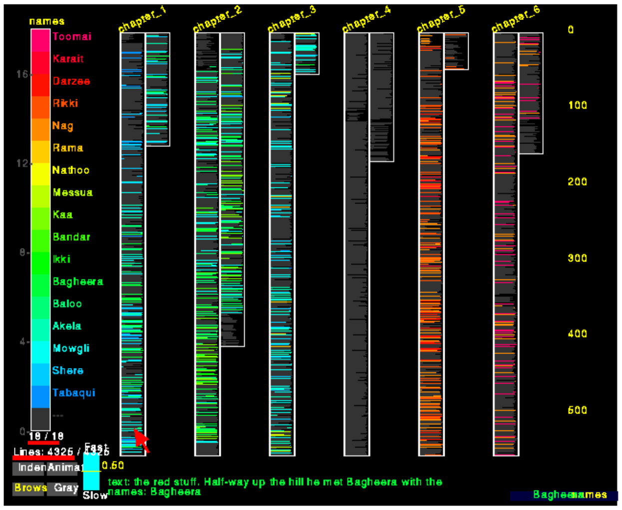

Table 13. SeeSoft (See Fig. 11)

Name	D	F	D'	X	Y	Z	T	R	—	[]	CP
Line#	Q		Q	*	P						
Type	O	>	O					C			
Type	O	sl>	O					C			

Another mapping of text is represented in Themescapes [19]. The text for each document (for example, a news story) is transformed into a document vector. Document vectors are compared giving rise to a matrix of similarities. The matrix is mapped onto a 2D landscape by means of multi-dimensional scaling.

A --> **mds**-->**X**:P, **Y**:P

This gives a 2D map of "themes". The frequency with which the various themes occur is mapped onto Z in the form of a surface.

Q --> **Z**:S .

The result is shown in Fig. 12, which depicts as a landscape, themes from CNN news. Thus in text, as in other specialized data areas, the transformations from the raw data type to a visualizable data type can be as important as the actual visualization.

Fig. 12. Themescapes[19]

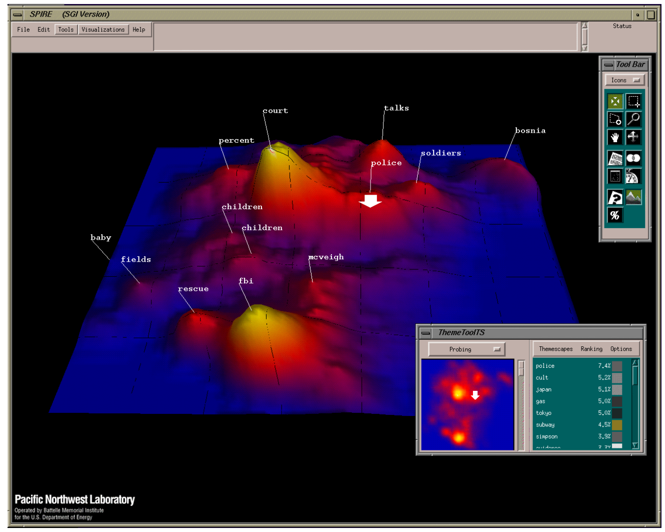

Table 14. Themescapes

Name	D	F	V	X	Y	Z	T	R	—	[]	A
Content	A	mds	xy	P	P						
Number	Q					S		C			

SUMMARY

In this paper we have sketched part of a scheme for mapping the morphology of the design space of visualizations. Because of space limitations, we have only sampled from the set of techniques being used. Two levels of analysis we have not addressed in this short paper are the larger organizational structure of information spaces and the organization of user tasks. With respect to the larger organizational structure, we have suggested in the text area an analysis into information space, workspace, sensemaking tools, and documents and surveyed systems in each of these areas [20]. For users tasks, we have suggested notions of "knowledge crystallization", comprising in part "information foraging" [21] and "sensemaking"[22]. These notions have been applied to the analysis of users of information visualization [23, 24] but considerable work remains to be done before a characterization of parts of the design space is possible. Our present analytical scheme does not express all of the important distinctions that could be made relative to these distinctions. Partially, this is because a more elaborate notation would require a much longer paper with more examples and partially this is because too complex a notation reduces its practical use—our main object. Besides helping to organize the literature, our present analysis suggests regions of new visualizations because it concentrates on the mappings between data and presentation.

During the next several years we believe that information visualization will enter mainstream use and that much information will be learned to guide new designs or apply established techniques. As for any technology area, is necessary to develop abstractions that rise above particular point designs in order to allow this codification of art into technology to occur.

REFERENCES

[1] P. R. Keller and M. M. Keller, *Visual Cues*. Los Alamitos, California: IEEE Press, 1993.

[2] M. C. Chuah and S. F. Roth, "On the semantics of interactive visualizations," in *Proceedings of the Information Visualization '96 Conference*, 1996.

[3] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualization," presented at Visual Languages 96, 1996.

[4] L. Tweedie, "Describing Interactive Visualisation Artifacts," in *FADIVA 3*, T. Catarci, Ed. Gubbio, Italy, 1996, pp. 63-66.

[5] J. Bertin, *Graphics and Graphic Information-Processing*. Berlin: Walter de Gruyter, 1977/1981.

[6] J. Bertin, *Semiology of Graphics: Diagrams, Networks, Maps*. Madison, Wisconsin: The University of Wisconsin Press, 1967/1983.

[7] J. Mackinlay, "Automating the design of graphical presentations of relational information," *ACM Transactions on Graphics*, vol. 5, pp. 110–141, 1986.

[8] R. M. Shiffrin and W. Schneider, "Controlled and automatic human information processing: II. Perceptual learning, automatic attending, and a general theory," *Psychological Review*, vol. 84, pp. 127–190, 1977.

[9] L. Treinish, "Ozone animation," : IBM, 1994.

[10] M. C. Chuah, S. F. Roth, J. Mattis, and J. Kolojejchick, "SDM: Malleable information graphics," in *InfoVis '95*. New York: ACM, 1995.

[11] C. Ahlberg and B. Shneiderman, "Visual Information Seeking using the FilmFinder," presented at CHI'94, 1994.

[12] S. Feiner and C. Beshers, "Worlds within worlds: Metaphors for exploring n-dimensional virtual worlds," in *ACM Symposium on User Interface Software*, 1990.

[13] R. Rao and S. K. Card, "The Table Lens: Merging graphical and symbolic representations in an interactive focus + context visualization for tabular information," in *Proc. CCHI '94 Conference on Human Factors in Computing Systems*. New York: ACM, 1994, pp. 318–322.

[14] K. C. Cox and S. G. Eick, "3D displays of Internet traffic," in *InfoVis '95*. New York: ACM, 1995.

[15] J. Lamping, R. Rao, and P. Pirolli, "A focus + context technique based on hyperbolic geometry for visualizing large hierarchies," in *CHI '95, ACM Conference on Human Factors in Computing Systems*. New York: ACM, 1995.

[16] B. Johnson and B. Shneiderman, "Tree-maps: A Space-filling approach to the visualization of hierarchical information structures.," in *Proceedings of IEEE Visualization '91*, 1991, pp. 284–291.

[17] G. G. Robertson, J. D. Mackinlay, and S. K. Card, "Cone trees: Animated 3D visualizations of hierarchical information," in *Proceedings of the ACM SIGSHI conference on Human Factors in Computing Systems*. New York: ACM Press, 1991, pp. 189–194.

[18] S. G. Eick, J. L. Steffen, and E. E. Sumner, "Seesoft—A tool for visualizing software," *IEEE Transactions on Software Engineeering*, vol. 18, pp. 957–968, 1992.

[19] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, and A. Schur, "Visualizing the non-visual: Spatial analysis and interaction with information from text documents," in *InfoVis '95*. New York: ACM, 1995.

[20] S. K. Card, "Visualizing retrieved information: A Survey," *IEEE Computer Graphics and Applications*, vol. 16, pp. 63–67, 1996.

[21] S. Card and P. Pirolli, "Information Foraging in Information Access Environments," presented at CHI '95 Human Factors in Computing Systems, 1995.

[22] D. M. Russell, M. J. Stefik, P. Pirolli, and S. K. Card, "The cost structure of sensemaking," presented at INTERCHI '93 Conference on Human Factors in Computing Systems, Amsterdam, 1993.

[23] P. Pirolli, P. Schank, M. Hearst, and C. Diehl, "Scatter/Gather browsing communicates the topic structure of a very large text collection," presented at Conference on Human Factors in Computing Systems, CHI-96, 1996.

[24] P. Pirolli and R. Rao, "Table Lens as a tool for making sense of data," presented at Workshop on Advanced Visual Interfaces, AVI-96, Gubbio, Italy, 1996.