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 point where a number of successful point designs have been proposed and a variety of techniques of been discovered. It is now appropriate 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

1. 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 such a framework and illustrate it with examples 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.

2. 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 a person already understands the information) and second, for graphical processing (in which case a person uses the manipulation and perception of graphical objects to understand the information). 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 variables" arrays or can be transformed (perhaps with loss of information) into this form. Text, for example, can be used to compute document vectors, which are 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. However, 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 make for data is whether their values are

Nominal (are only = or ≠ to other values),
Ordered (obeys a < relation), or are
Quantitative (can do arithmetic on them).

We denote 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 Q_{xy} and spatial variables that are actually geophysical coordinates Q_{lon}. We also distinguish between data *D* that is in the original dataset from data *D'* that has been selected from this set and possibly transformed by some filter or recoding function F.

Visualizations. Human visual processing involves two levels: automatic and controlled processing[8]. Automatic

Authorized licensed use limited to: University of Michigan Library. Downloaded on October 20,2026 at 03:05:02 UTC from IEEE Xplore. Restrictions apply.

processing works on visual properties such as position and color. It is highly parallel, but limited in power. Controlled processing works on abstract encodings such as text. It has powerful operations, but is limited in capacity. The distinction between these two types of capacity is important for visual design.

Visualizations are basically made from *marks* and their *graphical properties* [7]. There are a limited set of marks and properties, identified by Bertin [6], expanded by Mackinlay [7], and expanded further here. An elementary visual presentation consists of a set of marks (such as Points, Lines, Areas, Surfaces, or Volumes), their retinal properties (such as Color and Size), and their position in space and time (such as the XY plane in classical graphics and XYZT or 3D space plus time in information visualization). We also include, following [7], the properties of Connection (denoted "—") and Enclosure (denoted "[]"). Thus, visualizations are composed from the following visual vocabulary:

Marks: (Point, Line, Area, Surface, Volume)
Controlled Processing Graphical Features
Automatically Processed Graphical Properties
Retinal encodings: (Color, Size, Shape, Gray-level, Orientation, Texture, Connection, Enclosure)
Position: (X, Y, Z, T)

A key feature of visualizations is that they use the mappings between data and visual vocabulary to provide interactive access to the data. We focus here on two interactive techniques: View techniques (such as focus+context), which distort the space-time of the visualization, and Widget techniques, which add user interface objects (such as buttons) to the visualization.

To make comparisons easy, we use a common table format to describe visualizations:

Data				*Controlled*		*Automatic*					*Interaction*	
Variable	D	F	D'	CP	M	R	X	Y	Z	T	V	W

The table is conceptually structured with the data on the left and the user to the right. The vertical bold line in the table separates the data from the visualization vocabulary. The double lines indicate the major visual sections of Controlled Processing, Automatic Processing, and Interaction. The following summarizes the notation we use in the table (which we develop in context):

Symbol	*Meaning*
Variable	Name of case or variable dimension
D	Data Type ::= *N* (Nominal), *O* (Ordinal), *Q* (Quantitative). Q_X (Intrinsically spatial), Q_{lon} (Geographical) *NxN* (Set mapped to itself - graphs)
F	Function for recoding data ::= *f* (unspecified) > (filter) *s* (sorting) *mds* (multidimensional scaling) ↑ (interactive input to a function)
D'	Recoded Data Type (see D)
CP	Control Processing *tx* (text)
M	Mark types ::= *P* (Point), *L* (Line), *S* (Surface), *A* (Area), *V* (Volume)
R	Retinal properties ::= *C* (Color), *S* (Size), — (Connection), [] (Enclosure)
XYZT	Position in space time ::= *N, O, Q*, * (non-semantic use of space-time)
V	View transformation ::=*hb*(hyperbolic mapping)
W	Widget ::= *sl*(slider) *rb*(radio buttons)

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

3. 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 Figure 1). Because spatial and geographical variables are so frequent, we adopt the special notation of Q_X and Q_Y for Q (Quantitative) variables that are intrinsically spatial and Q_{lon} and Q_{lat} for Q variables that are earth coordinates. Scientific visualizations, then, usually have mappings

$Q_X \rightarrow X$ (i.e., a spatial quantitative variable is mapped into a position in X)

$Q_Y \rightarrow Y$,

and often

$Q_Z \rightarrow Z$

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

Figure 1. Ozone concentration[9]



Authorized licensed use limited to: University of Michigan Library. Downloaded on October 20,2026 at 03:05:02 UTC from IEEE Xplore. Restrictions apply.

Table 1. Ozone visualization (See Fig. 1)

Variable	D	F	D'	CP	M	R	X	Y	Z	T	V	W
Samples	*N*				*P*							
Ozone	*Q*	*f*	*O*			*C*						
Lon.	Q_{lon}						*Q*					
Lat.	Q_{lat}							*Q*				
Height	*Q*								*Q*			
Date	*Q*									*Q*		

Table 1 describes the mapping between the ozone data and the graphical vocabulary that is used in Figure 1. The rows of the table describe the variables with the case variable ("Samples") at the top and the value variables below. The nominal (*N*) set of Samples is mapped to point marks (P in column M), which have their retinal property of color (C in column R) mapped to the Ozone variable. The ozone mapping includes a function (*f*) that converts the quantitative (*Q*) ozone measurements to an ordinal (*O*) set that can be easily mapped to a set of colors. The quantitative (*Q*) variables of Longitude, Latitude, and Height are mapped to the positions X, Y, and Z, which determine the position of the point marks. The Date variable is mapped to time (T), which creates an animated visualization. Table 1 makes it clear that Figure 1 is a 3D animated visualization involving colored points.

4. GIS

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

$Q_{lon} \rightarrow X$,
$Q_{lat} \rightarrow Y$.

In Figure 2, an example from Visible Decisions, the free *Z*-axis is used for another variable, Profit:

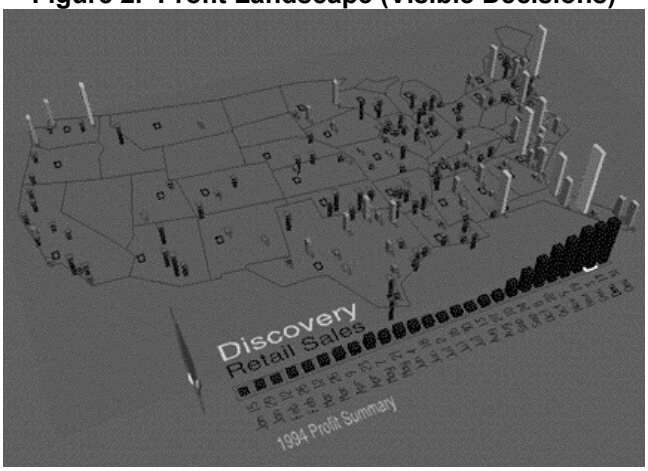
$Q \rightarrow Z$.

Table 2 describes the map part of Figure 2. The Offices variable is mapped to line marks (*L*). The Profit variable is mapped to the size of these lines (*Sz* in the R column). Profits are also mapped to the Z-axis and via a function (f) to a nominal set indicating the sign of the profits. This nominal set is mapped to the color of the lines (*C* in the R column). Table 2 clearly reveals that multiple graphical techniques are used to describe the Profit variable in order to enhance the perception of this important data variable.

Table 2. Visible Decisions (See Fig. 2)

Variable	D	F	D'	CP	M	R	X	Y	Z	T	V	W
Offices					*L*							
Lon.	Q_{lon}						*Q*					
Lat.	Q_{lat}							*Q*				
Profit	*Q*					*Sz*			*Q*			
		f	*N*			*C*						

Figure 2. Profit Landscape (Visible Decisions)



5. Multi-dimensional Plots

Multi-dimensional plots take variables that are not intrinsically spatial and map them onto X and Y, e.g.,

$Q \rightarrow$ X,
$Q \rightarrow$ Y.

When point marks are positioned on these axes, the result is the conventional scatterplot that is often used in statistical graphics.

The FilmFinder [11], shown in Figure 3 is a scatterplot that has been turned into an interactive visualization by using sliders (*sl*) and radio buttons (*rb*) to control a filter function (>) that determines which Films are shown on the scatterplot. Table 3 uses a "↑" in the F column to indicate which variables control the filter. Double lines between the rows indicate that these widgets are separate, visual presentations (not all variables are shown). Both interaction columns of the table are used. The Widget (W) column is used to indicate the interaction widgets and the View (V) column is used to indicate that the user can zoom and pan their view of the scatterplot. Table 3 clearly reveals that the users must switch their attention between the scatterplot and the widgets.

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 in the Scatterplot. In this way, the effect of many variables with a large number

Table 3. FilmFinder (See Fig. 3)

Variable	D	F	D'	CP	M	R	X	Y	Z	T	V	W
Film	*N*	>	*N'*		*P*							
Year	*Q*						*Q*				*Q'*	
Quality	*Q*							*Q*			*Q'*	
Type	*N*					*C*						
		↑			*P*	*C*	*N*					*rb*
Title	*O*	↑			*P*		*Q*					*sl*
Rating	*N*	↑			*P*	*C*	*N*					*rb*

Authorized licensed use limited to: University of Michigan Library. Downloaded on October 20,2026 at 03:05:02 UTC from IEEE Xplore. Restrictions apply.

Figure 3. FilmFinder [11]

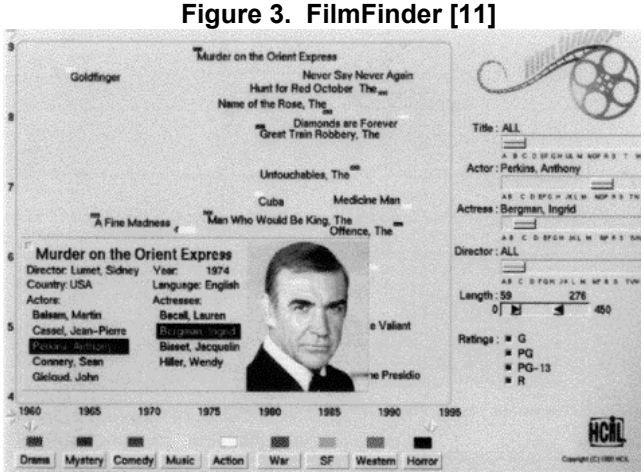
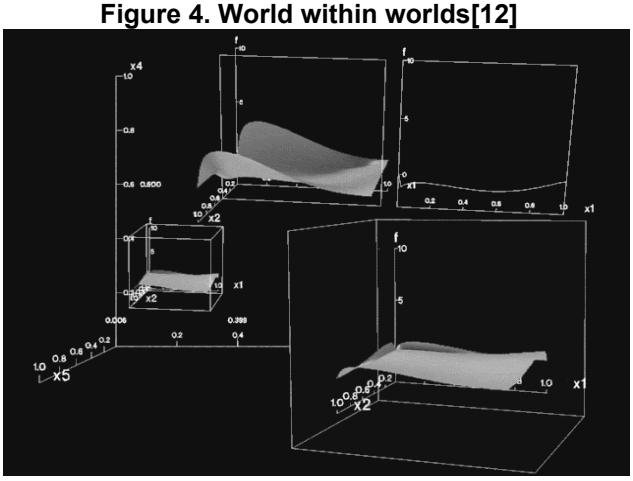


Figure 4. World within worlds[12]



of values can be taken into account without being coded or recoded as Retinal variables, keeping the display simple and easily interpretable. The *Q'* in the V indicate that the view is zoomed and panned.

Feiner's Worlds-Within-Worlds technique is another way of showing higher dimensional data (Figure 4). Three variables are mapped to the spatial dimensions:

Q_1 -->X
Q_2 -->Y, and
Q_3 --> Z.

This coordinate system is then used to interpret the meaning of additional coordinate systems (for only a few points at a time) allowing an overloaded mapping of the space:

Q_4 -->X
Q_5 -->Y, and
Q_6 --> Z.

Table 4 clearly shows this overloading with two values in the XYZ columns The overlapped coordinate space is a kind of details-on-demand display at the cost of occlusions in the original coordinate system. The view (V) column indicates which variables are clipped to reduce the impact of these occlusions. Overloading is a powerful visualization technique that must be used with care to make sure the users can perceive how the spatial dimensions are being used. As in the dynamic queries technique, allowing the user to move rapidly in time through the first three variables also increases the amount of the variable space that can be examined with a simple display.

Table 4. Worlds Within Worlds (Fig. 4)

Variable	D	F	D'	CP	M	R	X	Y	Z	T	V	W
X1	*Q*						*Q*				*Q'*	
X2	*Q*				*S*			*Q*			*Q'*	
X3	*Q*								*Q*		*Q'*	
X6	*Q*						*Q*					
X5	*Q*							*Q*				
X4	*Q*								*Q*			

6. 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, shown in Figure 5, produces an analysis quite different from our previous examples. Table 5 shows the cases (Players) mapped to the Y-axis and the variables mapped to the X-axis (only a few variables are shown). The *?s* in the function column indicates that the Players can be sorted on one of the variables. The V column indicates that the view can be distorted to provide extra room for some of the rows and columns. The ?tx in the Control Processing column indicates that text labels can be placed in this extra room.

This example indicates that the V column can be used for a more sophisticated distortion than just a pan and zoom.

Figure 5. Table Lens [13]

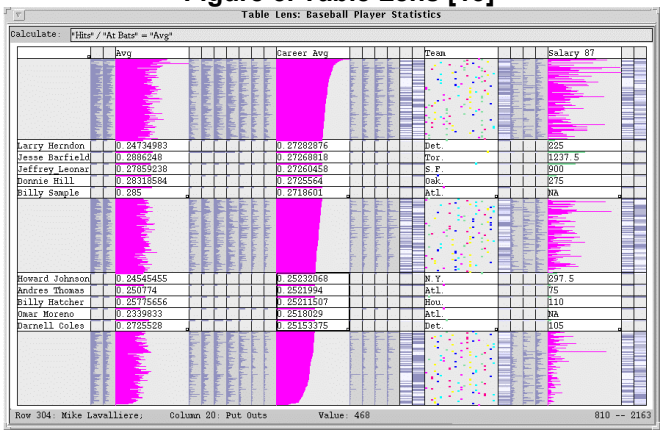


Table 5. Table Lens (Fig. 5)

Case/Var	D	F	D'	CP	M	R	X	Y	Z	T	V	W
Players	*N*	?s	*?O*	*?tx*				*?O*			*O'*	
Avg	*Q*	↑		*?tx*	*L*	*Sz*	*Q*				*Q'*	
Team	*O*	↑		*?tx*	*P*	*C*	*O*				*O'*	

Authorized licensed use limited to: University of Michigan Library. Downloaded on October 20,2026 at 03:05:02 UTC from IEEE Xplore. Restrictions apply.

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

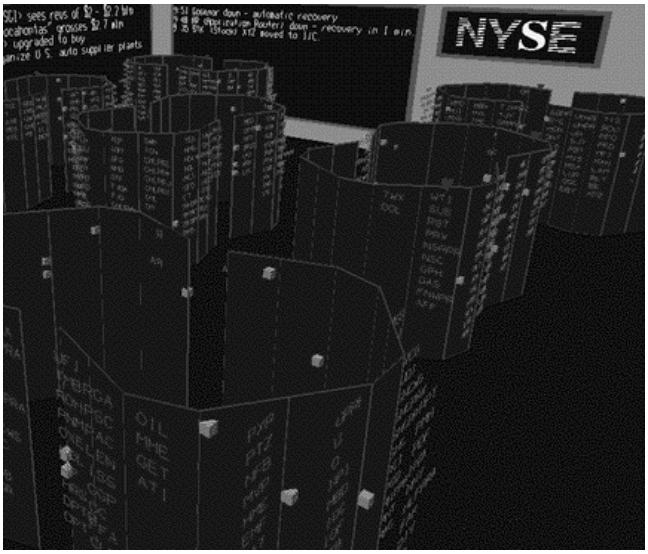


Table 6. New York Stock Exchange (Fig. 6)

Variable	D	F	D'	CP	M	R	X	Y	Z	T	V	W
Stock	*N*			*tx*	*L*							
Kiosk	*O*								*O*			
Lon.	Q_{lon}						*Q*					
Lat.	Q_{lat}							Q				
Volume	*Q*					*Sz*	*	*				

The key point revealed by Table 5 is that users must be able to perceive the original visualization through the distortion so that they can invert the mapping from the visualization to the data set.

7. Information Landscapes and Spaces

Landscapes lay information out on a surface, typically the XY plane. Landscapes may be of several sorts: real geographical coordinates, real spatial variables, or completely abstract mappings

$\{Q_{lon}$ or $Q_{X,}$ or $Q\} \rightarrow$ X

$\{Q_{lat}$ or Q_{Y} or $Q\} \rightarrow$ Y.

If the mapping extends to

Q--> Z,

we call it an *information space*.

In Figure 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 line sizes placed radially from the center of each. Table 6 uses a "*" to indicate that the bar does not use the X and Y dimensions semantically.

Figure 7. Internet traffic [14]

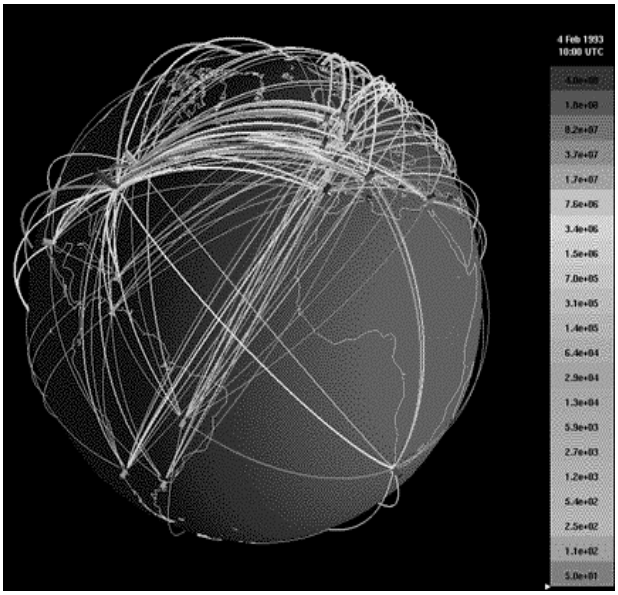


Table 7. World-Wide Internet traffic (See Fig. 7)

Variable	D	F	D'	CP	M	R	X	Y	Z	T	V	W
Site	*N*x*N*				L	—			*			
Lon.	Q_{lon}						*Q*					
Lat.	Q_{lat}							*Q*				
Volume	*O*					*C*						

8. Node and Link

Node and link diagrams allow the encoding of linkage information between entities. They can be thought of as a mapping from a Nominal set to itself {*N*x*N*}. These are then mapped into XY.

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 of the site are mapped onto XY (really onto a sphere). The links are mapped onto Connection Lines, which are shown as a special kind of retinal property in Table 7. The * indicates that these lines do not have any special meaning in Z (although it might be argued that they are proportional to the distance between the computers).

9. Trees

The nodes for graphs also 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] shown in Figure 8. Since hyperbolic space expands exponentially, it is a good place to lay out exponentially expanding graphs, such as trees.

Authorized licensed use limited to: University of Michigan Library. Downloaded on October 20,2026 at 03:05:02 UTC from IEEE Xplore. Restrictions apply.

Figure 8. Hyperbolic browser [15]

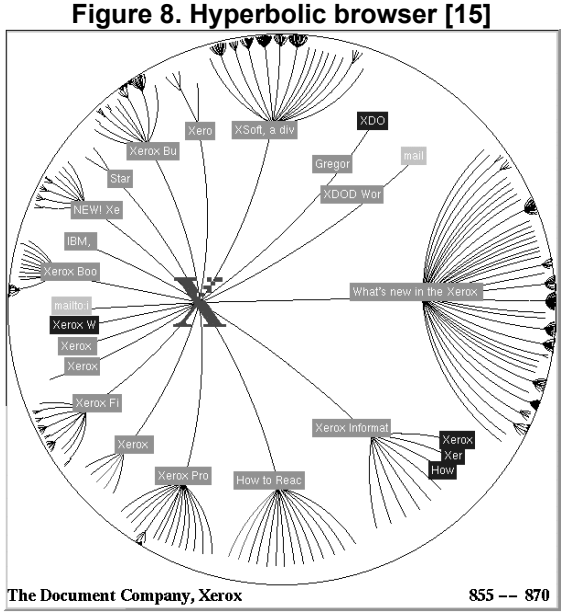


Table 8. Hyperbolic Browser (See Fig. 8)

Variable	D	F	D'	CP	M	R	X	Y	Z	T	V	W
Page	*NxN*	*tree*	Q_{xy}		P	—	*	*			*hb*	

.

Table 8 indicates there is a *tree* function that converts *NxN* set into a set of XY positions (Q_{xy}). The * indicates that these positions do not use the X and Y semantically. The V column is used to indicate the hyperbolic distortion of the tree layout. Again, such distortions are only as effective as the user's ability to perceive the undistorted graphical vocabulary mapped to the data.

Trees can also be visualized as nested enclosures. 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 Figure 9. The problem is that the same variable is mapped onto two different position presentations, each half of the time

$Q \rightarrow$ X (half time)

$Q \rightarrow$ 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 perceptual cost, which is clearly shown in Table 9.

Figure 9. Tree-Map [16]

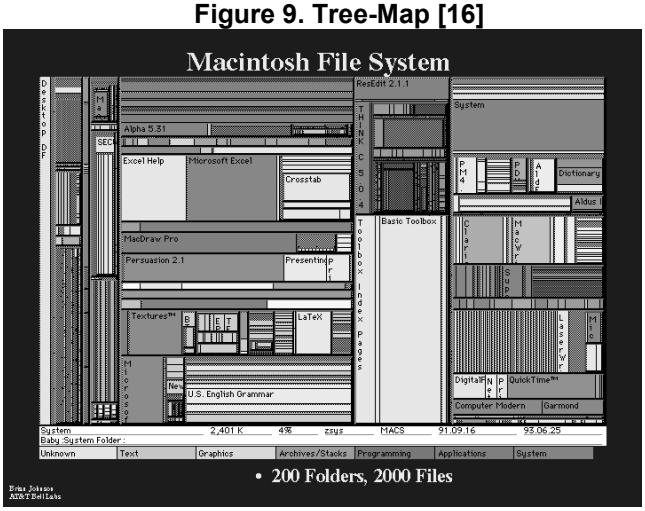


Table 9. Tree-Map (See Fig. 9)

Variable	D	F	D'	CP	M	R	X	Y	Z	T	V	W
Files	*NxN*	*tree*	*QQ*		*A*	[]	*	*				
Size	*Q*					*Sz*	*Q*⇔*Q*					
Type	*N*					*C*						

Cone-trees address the problem that trees are large in width by increasing 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. 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.

Figure 10. Cone tree[17]

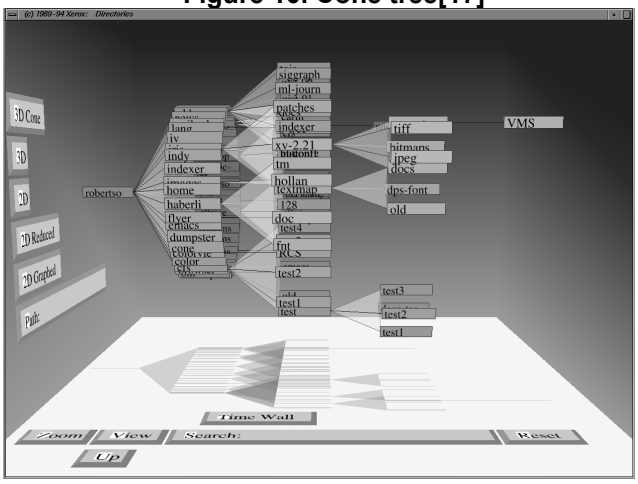


Table 10. Cone-tree (See Fig. 10)

Variable	D	F	D'		M	R	X	Y	Z	T	V	W
File	*NxN*	*tree*	Q_{xyz}		*P*		*	*	*			

Authorized licensed use limited to: University of Michigan Library. Downloaded on October 20,2026 at 03:05:02 UTC from IEEE Xplore. Restrictions apply.

10. Special Data Transforms: Text

We have discussed some of the main classes of visualizations. But an important point to make is that much of the power of visualization comes from the techniques for transforming data types into the data forms that can be mapped into these visualizations. For example, text itself can, of course, be visualized directly.

Q --> CP.

But this does not work for a large text datasets and the text has to be processed with Controlled instead of Automatic processing [8]. One approach to visualizing large amounts of text is to map it line by line onto long strips

Q --> X,

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, Figure 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.

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 with multi-dimensional scaling. 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. The result is shown in Figure 12, which depicts 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.

Figure 11. SeeSoft[18]

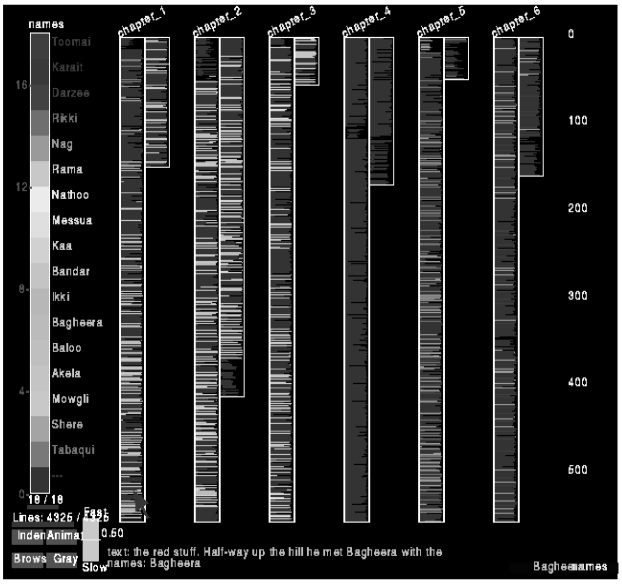


Table 11. SeeSoft (See Fig. 11)

Variable	D	F	D'	CP	M	R	X	Y	Z	T	V	W
Line	*tx*	*f*	*Q*		*L*	*Sz*	*Q*					
Line#	*Q*							*Q*				
Type	*O*					*C*						

Figure 12. Themescapes[19]

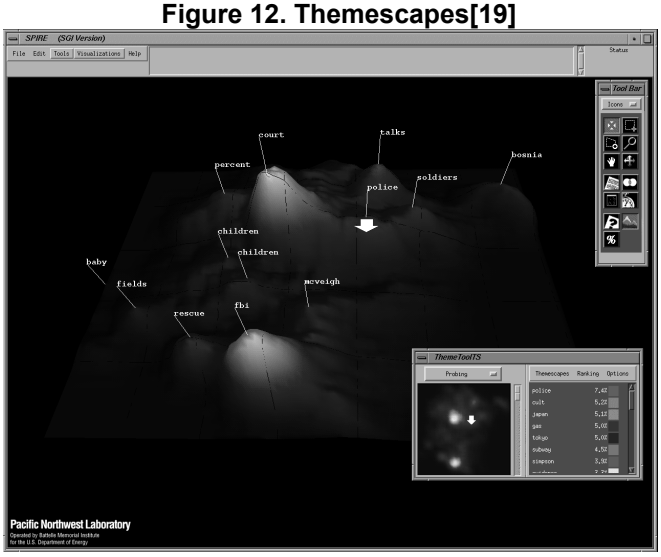


Table 12. Themescapes (See Fig. 12)

Variable	D	F	D'	CP	M	R	X	Y	Z	T	V	W
Content	*A*	*mds*	Q_{xy}		*S*		*Q*	*Q*				
Frequency	*Q*					*C*			*Q*			

11. 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 visualization techniques. Two levels of analysis 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 previously 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 user's tasks, we have previously 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 issues. This is partially because a more elaborate notation would require a much longer paper with more examples,

Authorized licensed use limited to: University of Michigan Library. Downloaded on October 20,2026 at 03:05:02 UTC from IEEE Xplore. Restrictions apply.

and partially 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. The table notation, in particular, organizes these mappings in a way that reveals when a data set is mapped to a graphical property in isolation, with overloading, or via distortion. The key issue for effective visualization is that users must be able to invert this mapping and perceive the data in the visualization.

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, it 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, pp.29-36.

[3] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualization," presented at Visual Languages 96, 1996, pp. 336-343.

[4] L. Tweedie, "Describing Interactive Visualization 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, pp. 36-42.

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

[12] C. Beshers and S. Feiner, "*AutoVisual*: Rule-Based Design of Interactive Multivariate Visualizations," *IEEE Computer Graphics and Application*, vol. 13(4), pp. 41-49, 1993.

[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. CHI '94 Conference on Human Factors in Computing Systems*. New York: ACM, pp. 318–322, 1994.

[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*, pp. 284–291, 1991.

[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, pp. 189–194, 1991.

[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.

Authorized licensed use limited to: University of Michigan Library. Downloaded on October 20,2026 at 03:05:02 UTC from IEEE Xplore. Restrictions apply.