Display Models for Visualization

Jonathan C. Roberts
Computing Laboratory,
University of Kent at Canterbury,
England, UK, CT2 7NF
J.C.Roberts@ukc.ac.uk

Abstract

Models for visualization are important, helping the developer and user to understand the visualization process; to follow the connections and the data paths through the system; and to reference and compare the functionality and the limitations of different systems or techniques.

Display models specifically classify the data by what type of output can be created.

Jacques Bertin [2] described a symbolic reference model that he used to describe images and displays. In this paper we review his and other 'display orientated models' describing important aspects of these methods and ideas. We then translate Bertin's scheme into an algebraic form as a method to describe visualizations.

Keywords: Visualization models, Display models, Bertin's semiology.

1. Introduction

There are many diverse data types, data storage methods, system configurations and dimensions all with different names, terminology and models applied to them. A classification model allows systems, data and algorithms to be grouped and compared.

Display models classify the information by what type of output may be created. In this paper we present different display models including Bertin's model. Indeed, Bertin's display model [1, 2] describes many 'image space' components with a graphical method for representing the "Utilization of the Image Space" and it is possible to transfer this method into an algebraic form. Thus, within this paper we also present one such algebraic scheme and use it to describe different visualization techniques.

Visualization is an issue that has been addressed by many researchers but there is currently no standard model for visualization, whether for the visualization system, flow of data or display aspects. A commonly adopted model, describing the visualization flow, is the *dataflow* model (Upson [15], and Haber and McNabb [7]), this is a good general model and is used in many visualization systems, for example, AVS [15], IBM Data Explorer [11] and IRIS Explorer [8].

Within this paper we focus on the 'display methods' of visualization. In the following sections we provide a review of many 'display models', then extend Bertins method into an algebraic form.

2. Data and Dimensionality

Earnshaw and Wiseman [6] provide a general "Data and Display Dimensionality" classification scheme that organises current output representation techniques by comparing the output primitives dimension with the dimension of the data.

Collins [5] extends Earnshaw and Wiseman's model (Table 1) to include the data types of Scalar (S), Vector (V) and Tensor (T). The Table includes multiple display examples; some of these are explained below:

Attribute Mapping maps attributes to a surface, using colours and textures.

Colour Maps are formed by mapping colour, from the range of the data values, onto a 2D image.

Dot Surfaces are surfaces that are made from points.

Glyphs represent symbols that change in appearance depending on the values and position within the data, and can depict values, vectors and tensors.

Height Fields are generated from creating a height (terrain) at each point on two dimensional data.

Moreover, Bertin specifically presents a display categorization.

Display Dimension	Dimensionality of the Data 1D	2D	3D	nD
0D	Points	Scatter Plots(S,M)	3D Scatter Plots(S, M) Tri-Scatter Plot(S) Dot Surfaces(S)	
1D	Lines / Curves(S)	Contour Maps(S)	Vector Arrows(V)/ Streamlines(V)	
2D	–	Height Fields(S) Colour Maps(S)	Tiled Surfaces(S) Ribbons(V)	Attribute mapping (S,M)
3D	–	–	Solid (S)/ Volume Modelling (S)	Glyph (T) icon (M)
Scalar(S), Vector(V), Tensor(T), Multivariate(M)				

Table 1. Examples classified by Data and Display Dimensions, after Earnshaw/Wiseman [6]

3. Bertin's Display Primitive Model

Bertin [1, 2] describes a display primitive classification model dividing the output primitives into four categories:

Diagrams including bar charts, scatter plots, histograms and schematics.

Networks including trees and path connections.

Maps including geographical maps and diagrams in which the positions are constrained by a "real life" object. Often maps incur a non-uniform projection (that must be understood when reading the map); for example, the spherical surface of the Earth is often mapped onto a flat two dimensional geographical map.

Symbols including signs and icons.

Bertin splits these into components that represent the dimensions of the output primitive, with each output primitive split into three categories of either *point*, *line* or *area*. He designates a level of *organization* to each primitive. A primitive can therefore be:

Associative ($\equiv$) where any object can be immediately isolated as belonging to the same category, and each object can be considered as *similar*.

Selective ($\neq$) where each object can be grouped into a category *differenced* by this variable (forming families).

Ordered (O) that allows each element to be grouped into an order of scale.

Quantitative (Q) where each element can be compared to be greater or less than another element. This includes values as percentages and logarithms.

This classification model allows the inclusion of methods such as pie diagrams, bar charts, scatter plots and three dimensional isosurface diagrams. Bertin classifies the output display method as an icon, using lines, arcs and arrows (Figure 1). The Diagrams, Networks, Maps and Symbols are represented by how they 'utilize' the image space, whether circular or linear, and this is represented in Figure 1 as "Utilization of the Image Space".

3.1. Display technique Catalogs

Lohse et al [10], with many volunteers, have classified multiple visualization representations. Subjects sorted the visual representations into clusters of objects, from which a hierarchical tree diagram was created. The clusters formed groups of graphs, tables, maps, diagrams, icons and network charts; using similar classifications to Bertin [2]. A scatter plot, of icons to networks (on one axis) against graphs, tables to maps and diagrams (on the other axis) was generated.

4. Bertin's Symbolic Schema

We extend Bertin's representation to include a Composite classification. This Composite category includes images that use multiple primitives, such as, a diagram of glyphs or a map (showing geometric information) with a network (showing connectivity information). Figure 5 shows some visualization techniques within Bertin's classification structure, however, the Organization of each component is not depicted in the diagram. This information can easily be included but the component of Organization often depends on the data being represented and on the method of representation. Figure 4 shows some examples with one particular Organization classification.

Bertin describes six representation methods, named *Retinal Variables*, of shape, orientation, colour, texture, value,

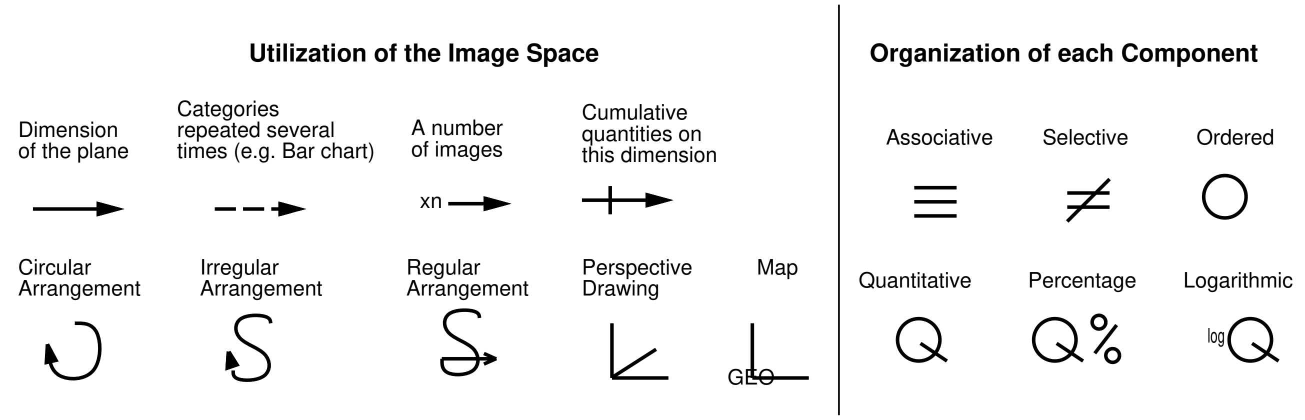

Figure 1. Graphical Classification Scheme, after Bertin

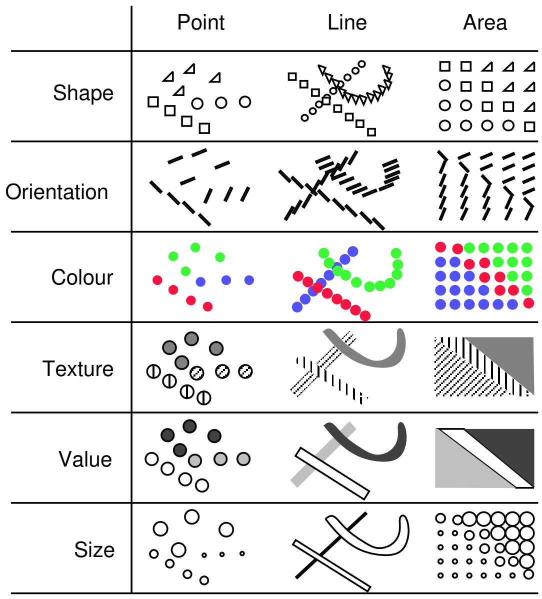

Figure 2. Retinal Variable Examples, after Bertin

Retinal Variable	Point, Line or Area	$\equiv$	$\neq$	O	Q
Shape	p,l,a	√			
Orientation	p	√	√		
	l	√	√		
	a	√			
Colour	p,l,a	√	√		
Texture	p,l,a	√	√	√	
Value	p,l,a		√	√	
Size	p,l,a		√	√	√
Planar Dimensions	–	√	√	√	√
$\equiv$ Associative, $\neq$ Selective, O Ordered, Q Quantitative					

Figure 3. Retinal Variables, after Bertin

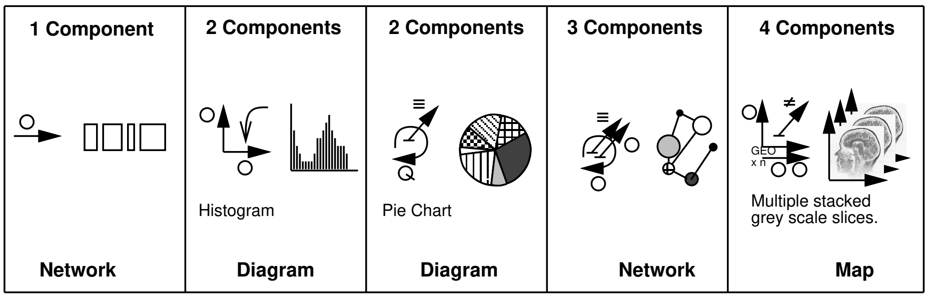

Figure 4. Output Classification Model with Component Organization, after Bertin

Object Classes	Operation Classes
scalar	identify
scalar field	locate
nominal	distinguish
direction	categorize
direction field	cluster
shape	distribution
position	rank
spatially extended object	compare
structure	within and between relations
	associate
	correlate

Table 2. Object and Operation Classes, after Wehrend and Lewis

size. Each variable can be classified using points, lines and areas. Figure 2 shows examples of the six retinal variables. Moreover, colour may be described by Hue, Saturation and Brightness, and attributes such as transparency, and animation may be added [9]. The level of organization can be compared with the retinal variables in classifications of point, line or area; this is shown, with the planar dimensions, in Table 3.

Wehrend and Lewis [16] generate a matrix of display techniques of 'Object Classes' against 'Operation Classes'. Object Classes are defined by the nature of the target domain, such as a scalar value and the shape of an object. The Operation Classes define the user's goal, whether to read off an actual value (Identify) or to compare two such values (Compare), for example. Table 2 lists the Object and Operation Classes. This 'catalog' of techniques does not hold information about the difference, similarity or merits of each technique, but can be used as a reference into techniques that are available.

4.1. Underlying Field Models

Brodlie [4] describes a classification model that "models the underlying field rather than the dimensionality and order of the sampled data", creating a conceptual model, Figure 6. He then describes a classification scheme that allows the underlying field and display to be classified with an algebraic expression. He splits the data into two cases of *ordinal (O)* and *nominal (N)* which describe order and no associated order, respectively. Scalar, Vector and Tensor details are referenced as S, V and T, that represent the type of the data, and are applied to the basic type (N or O) as superscripts. Independent variables are noted inside parenthesis and a range, or an aggregate, is labelled inside square brackets. The dimensions of each variable is noted as subscripts; positions in two dimensional space can be represented by O_2; similarly the number of components of a Vector (V), such as two, is represented by V_2 and the components of a Tensor (each separated by colons), such as a three by three dimensional tensor, are represented by $T_{3:3}$. Some examples are described in Table 3.

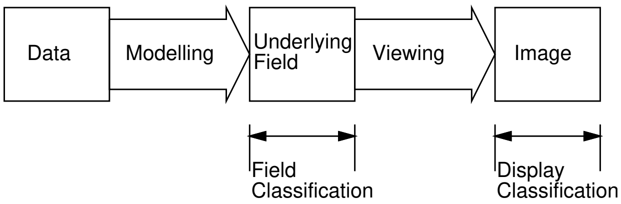

Figure 6. Underlying Field Visualization Process Model

Output Method	Field Classification	Display Classification
Histogram	$O^S(O_1)$	$O^S([O_1])$
Bar Chart	$O^S(N_1)$	$O^S(N_1)$
2D Contouring	$O^S(O_2)$	$O^S(O_1)$
Surface Rendering (from 3D data)	$O^S(O_3)$	$O^S(O_2)$
Volume Rendering	$O^S(O_3)$	$O^S(O_3)$
3D wind, arrow plot	$O^{V_3}(O_3)$	$O^{V_3}(O_3)$
Ordinal(O), Nominal(N), Scalar(S), Vector(V), Tensor(T) Dimensions 1, 2, 3 etc.,		

Table 3. Visualization Classification Examples, after Brodlie

Brodlie explains that this system allows the underlying data field and the display technique to be classified, but it does not classify multiple techniques. For example, "temperature over an aircraft wing, is a two dimensional subspace within three dimensional space".

4.2. Display Models for Automated Visualization Design

Some visualization systems automatically create the visualizations from a database of knowledge (metadata information) and user requirements. These tools classify the display variables to generate an appropriate visualization automatically.

The Vista tool [14], for example, creates appropriate visualizations by asking the user to preference each variable. Perception rules are applied to the variables such as "position is more effectively perceived than colour" and quantitative information is easier to perceive "by using geometry rather than colour". Vista divides the primitive visualiza-

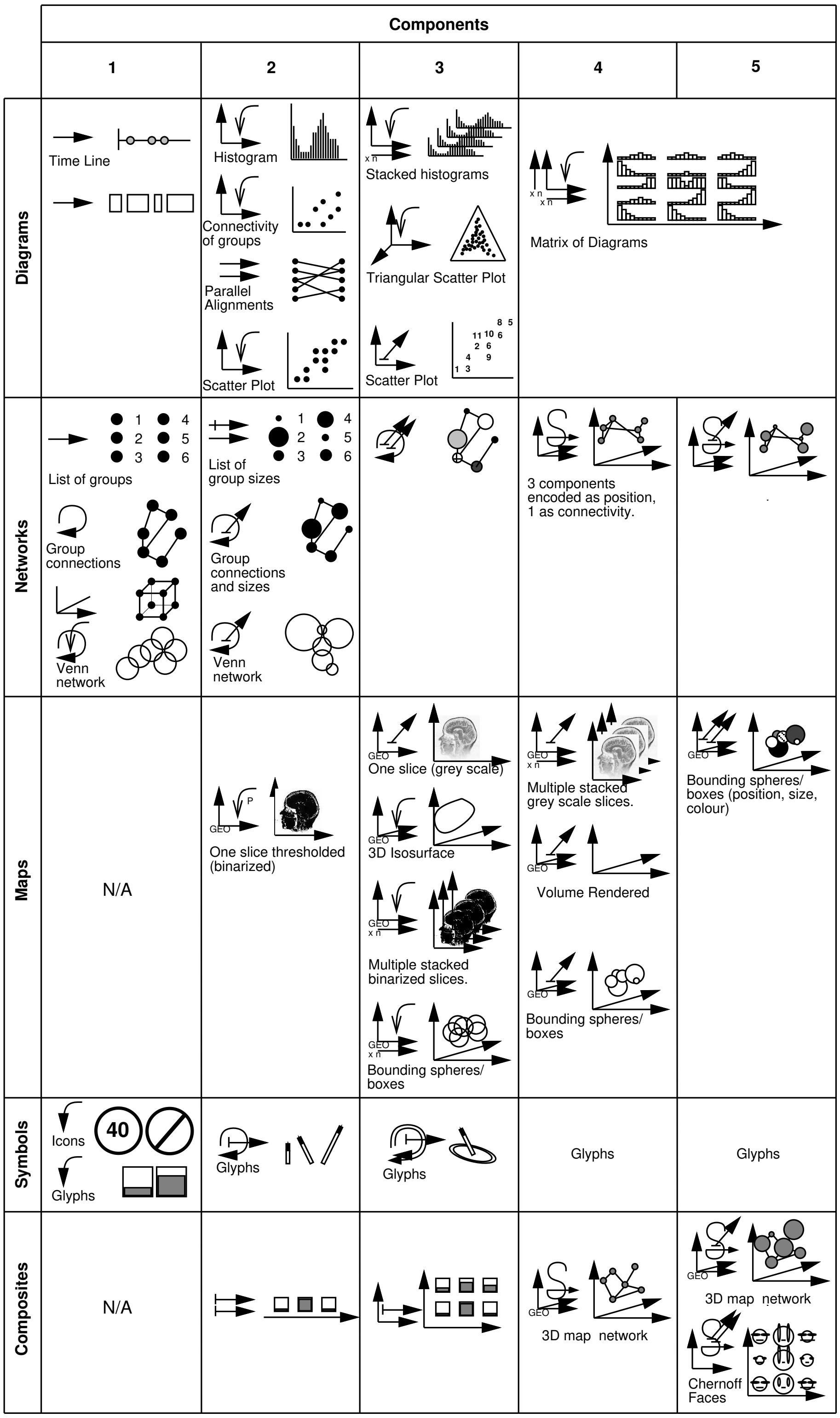

Figure 5. Output Classification Model, after Bertin, diagram representations by Roberts

tion techniques into Positional, Temporal and Retinal variables. Positional is divided further into one, two and three dimensions. Animation is used to depict the Temporal variables and the Retinal variables are divided (like Bertin) into colour, shape, size, orientation and texture.

Mackinlay [12] designed APT (A Presentation Tool), based on terminology from Bertin [2] and the effectiveness of visual perception from the work of Cleveland and McGill. Mackinlay composes complex presentations from simpler presentations, where each less complex presentation displays a subset of the overall information. The tool can create effective displays of bar charts, scatter plots and connected graphs.

Beshers and Feiner [3] discuss many other Automated Visualization Design tools.

5. Algebraic Extension to Bertin's Model

Bertin's model uses a graphical notation to describe the different display techniques. He encodes the type of the display, utilization of the image space and the organisation of the component. This classification scheme could be represented in algebraic notation. We propose one method to encode the type of the display and the utilisation of the image space. The organisation of the component depends on the data and the type of retinal variable used to display the component, so it is not encoded in this scheme.

5.1. Classifiers

We break, as before, the images into: Diagrams (D), Networks (N), Maps (M) and Symbols (S). Therefore D, N, M and S represent the classifiers. However within an image display some of the components are represented by the retinal variables (such as size, shape and colour). The classifiers are extended to include these Retinal Variables (R). The scheme encodes no distinction between the types of the retinal variables, but does encode the total amount of retinal variables used in a particular view. (See Classifier Quantity, section 5.2).

Multiple images, or multiple views [13], such as a matrix of histograms, are represented by Bertin with '$\times n$' symbols. Our scheme represents these by the letter X.

Symbols (including icons and glyphs) and Retinal variables can be explained as describing the same component. For example, a glyph, such as a temperature gauge, is represented by the retinal variable size. In some instances the reverse is also possible where, for example, every point in a diagram is represented by a circle symbol. Our scheme therefore overloads the retinal operator (R) to represent both retinal variables (R) and the symbol (S) classifiers. Moreover, the classifiers now only include D, N, M, X and R; i.e. S is excluded.

5.2. Classifier Quantity

Bertin represents each component as a single closed arrow, the total number of components therefore being calculated from the number of arrows in the graphical representation. We represent the number of components (for a particular classifier) as a power. A scatter plot diagram (of x component against a y component) could then be represented by D^2.

The total number of components for a particular display can be calculated by adding the powers together.

Moreover, if the Symbol (S) classifier was included (with the Retinal Variable R), the quantity classifier would need to represent zero components. The example of a temperature gauge (with the retinal variable size) could be represented by (the composite form of) S^0R^1.

5.3. Utilization of the Image Space

Bertin describes the utilization of the image space in categories of: regular, irregular, circular and perspective arrangement; our scheme divides them similarly, and names them: r, i, c and p respectively.

The symbols, for example, do not easily fall under this classification as not having any particular arrangement; however, we represent the symbols under the irregular classification.

5.3.1. Expression Form

The algebraic expressions are formed from Classifiers and a 'Method of utilizing the image space' and a power represents the number of components for this classifier. For example, a circular network of objects depicting their connectivity with one retinal variable (representing the number of elements in the object) is represented by Nc^1R^1, the total amount of components being two.

Composite forms are generated by joining the single expressions together. Brackets are used to disambiguate the scope of the the multiple classifier X. For example a group of stacked grey scale slices (maps) can be represented by $X^1(Mr^2R^1)$.

6. Examples and Summary

Table 4 lists some display methods with their appropriate Algebraic notation; the algebraic display methods are taken from the schematics in Figure 5.

The algebraic form allows complicated displays to be described as composite groups of statements, but the scheme disregards information about the organization of each component (whether selective, ordered or associative). The origin of the data and the exact description of the display is not

Display Methods		
Type	Description	Algebraic Classification
D	Time Line	Dr^1
	Histogram	Dr^2
	Stacked Histogram	X^1Dr^2
	Matrix of Histograms	X^2Dr^2
N	List of Groups	Nr^1
	Circular Group Connections	Nc^1
	Venn Network	Nr^1
	List of Group and Sizes	Nr^1R^1
	Network of Groups and Sizes	Nc^1R^1
	Network of Groups, Sizes and Texture	Nc^1R^2
M	Binary Threshold Slice	Mr^2
	Grey level Slice	Mr^2R^1
	Stacked, Grey level Slices	$X^1(Mr^2R^1)$
	Volume Rendering	Mp^3R^1
S	Road Sign	R^1
	Temperature Gauge	R^1
C	Network with Size, inside 3D Map	$Mp^3Nr^1R^1$
D Diagrams, **N** Networks, **M** Maps, **S** Symbols, **C** Composites		

Table 4. Display Methods with Algebraic Classification

represented; for example, both an X-Ray image and a two dimensional slice (through *real-life* data) are represented as Mr^2R^1, both having a total of three components. However, the algebraic form provides a method to classify abstractions and visualizations.

6.1. Acknowledgements

I acknowledge Dr. Tim Hopkins and Dr. Steve Hill for their help.

References

[1] J. Bertin. *Graphics and graphic information-processing*. Walter de Gruyter, 1981. William J. Berg and Paul Scott (Translators).

[2] J. Bertin. *Semiology of Graphics, translation from Sémilogie graphique (1967)*. The University of Winsonsin Press, 1983. William J. Berg (Translator).

[3] C. G. Beshers and S. K. Feiner. Automated design of data visualizations. In L. Rosenblum, R. A. Earnshaw, J. Encarnacao, H. Hagen, A. Kaufman, S. Klimenko, G. Nielson, F. Post, and D. Thalmann, editors, *Scientific Visualization Advances and Challenges*, pages 87–102. IEEE Computer Society Press and Academic Press, 1994.

[4] K. Brodlie. A classification scheme for scientific visualization. In R. E. Earnshaw and D. Watson, editors, *Animation and Scientific Visualization – Tools and Applications*, pages 125–140. Academic Press, 1993.

[5] B. M. Collins. Data visualization — has it all been seen before? In R. E. Earnshaw and D. Watson, editors, *Animation and Scientific Visualization – Tools and Applications*, pages 3–28. Academic Press, 1992. 0-12-227745-7.

[6] R. A. Earnshaw and N. Wiseman. *An Introductory Guide to Scientific Visualization*. Springer-Verlag, 1992.

[7] R. B. Haber and D. A. McNabb. Visualization idioms: A conceptual model for scientific visualization systems. In B. Shriver, G. M. Nielson, and L. J. Rosenblum, editors, *Visualization in Scientific Computing*, pages 74–93. IEEE Computer Society Press, 1990.

[8] M.-A. Halse, D. Young, and L. McCormick. *IRIS Explorer User's Guide*. Silicon Graphics Computer Systems – Silicon Graphics Inc., 1992. (Document Number 007-1371-020).

[9] Karen R. Atkinson and Jonathan C. Roberts. Graphics and Visualization within Cross-Stitch. In *Eurographics UK 1999 Conference Proceedings*, 17th Annual Conference, pages 129–141, Eurographics UK, PO Box 38, Abington, Oxon, OX14 1PX, April 1999. (http://www.cs.ukc.ac.uk/people/staff/jcr/eguk99/index.html).

[10] J. Lohse, H. Rueter, K. Biolsi, and N. Walker. Classifying visual knowledge representations: A foundation for visualization research. In *Proceedings Visualization '90*, pages 131–138. IEEE Computer Society Press, 1990.

[11] B. Lucas, G. D. Abram, N. S. Collins, D. A. Epstein, D. L. Gresh, and K. P. McAuliffe. An architecture for a scientific visualization system. In *Proceedings Visualization '92*, pages 107–114. IEEE Computer Society Press, 1992.

[12] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM transactions on Graphics*, 5(2):110–141, 1986.

[13] J. C. Roberts. On Encouraging Multiple Views for Visualization. In E. Banissi, F. Khosrowshahi, and M. Sarfraz, editors, *IV'98 – Proceedings International Conference on Information Visualization*, pages 8–14. IEEE Computer Society, July 1998.

[14] H. Senay and E. Ignatius. A knowledge-based system for visualization design. *IEEE Computer Graphics and Applications*, 14(6):36–47, November 1994.

[15] C. Upson, T. Faulhaber, D. Kamins, D. Schlegel, D. Laidlaw, F. Vroom, R. Gurwitz, and A. van Dam. The application visualization system: A computational environment for scientific visualization. *IEEE Computer Graphics and Applications*, 9(4):30–42, 1989.

[16] S. Wehrend and C. Lewis. A problem-oriented classification of visualization techniques. In *Proceedings Visualization '90*, pages 139–143. IEEE Computer Society Press, 1990.