Visualization Display Models – ways to classify visual representations

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

Abstract

Visualizations are generated of diverse data and use many different techniques and visual methods. It is often beneficial to evaluate what is being visualized and how the visualization is made-up. Such an analysis may aid the developer, to understand what 'tools' are available, and help the user to reference and compare different realizations.

Display models specifically classify the data by what type of output can be created. We review many 'display oriented models' and discuss important aspects of these methods and ideas. In presenting these models we encourage their use. In particular we focus on the symbolic reference model of Jacques Bertin [2]. He used this model to describe images and 2D visualizations. We translate Bertin's scheme into an algebraic form as a method to describe visualizations.

Key words: Display models, Visualization reference models, Bertin's semiology.

Introduction

Visualization is both a process of *presentation* and *discovery*. Through using different techniques the visualizer displays the data in such a way as to allow the user to gain a deep understanding of the underlying information. The user and visualizer may be the same person, then an exploration environment is formed, where the user can try different scenarios and parameters to gain a correct understanding of the information.

The original data may be numeric or textual information, sampled from the real world, simulated using mathematics or gathered from ideas; whatever its origin the aim is the same: to present the concept or information in order that insight and understanding is gained by the user or observer.

There are many diverse data types, data storage methods, system configurations and dimensions all with different names, terminology and described by different models. Thus, it is beneficial to classify the 'systems', so that they may be grouped and compared. In this paper we focus on the 'display methods' of visualization.

Display models classify the information by what type of output may be created. For example, it may be possible to say "this visualization is one of those $O^S(O_1)$". Simple, and complex, classifications exist. For example, it is possible to use the dimensionality of the data to classify the images. But it is useful to also use the dimensionality of the output and the primitives used in the realization, because it is possible to generate different outputs of the same information (Multiform representations [15]).

In multiform visualization, a three dimensional construction of a building, for example, may be visualized in two ways (1) using a solid surface rendering that shows a more realistic representation of the design and (2) with a wire-frame representation, that allows internal parts of the construction to be viewed. Both of these examples have their advantages and disadvantages, occlusion is the major problem in the surface view, whilst depth cueing is an issue in the wire frame.

In this paper we present different display models. Such models (1) aid the developer and visualizer to understand what tools and techniques are available, and (2) help the observer to compare different generated visualizations. One of the models we review is Bertin's model. This model [1, 2] allows 'image space' components to be described with a graphical method. It is possible to transfer this method into an algebraic form, and we present one such algebraic scheme and use it to describe different visualization techniques.

Visualization – a conceptual Model

Visualization is about seeing the unseen and gaining an understanding of the underlying information. The Oxford English Dictionary [17], defines visualization to be "The action or fact of Visualizing; the power or process of forming a mental picture or vision of something not actually present to the sight; a picture thus formed".

In essence there are two main processes (see Figure 1). One of *visualization* that changes data, concepts and information into a representation that viewed by the user, and the other of *perception*, where the user tries to understand the information.

Within visualization, the graphical medium is often

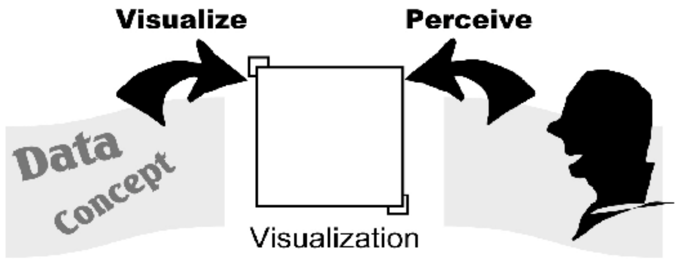

Figure 1: Thinking about visualizations, an abstract model

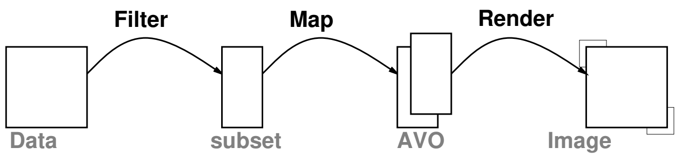

Figure 2: The dataflow model of visualization. The information is filtered, to select the required information, mapped into an Abstract Visualization Object (an abstract geometric representation of the information), and rendered to generate the image.

used to portray the information. However, other mediums may be used, such as touch – in the form of movement, from (say) force feedback armatures or joysticks – or sound (known as sonification). The art here is to generate a representation that may be correctly understood. Visual cues such as shading, colour, shadows, perspective [21] may be used to aid the user in the perception task.

In the book 'The art of thought' Graham Wallas [20] distinguishes four thought stages: Preparation, Incubation, Illumination and Verification. These, represented by the right side of Figure 1, provide an abstract series of methods describing one way we 'think'. At the preparation stage we are conscious of learning and gaining knowledge, that perhaps needs to incubate, while we question the phenomena and allow our subconscious to work out the spatial quantities before illumination occurs. Then we can use our 'logic' to question and verify the notions gained.

Display models are the left side of Figure 1 and describe visual methods that may be employed to generate a visualization. There are many processes involved in generating a visualization. A commonly adopted reference is the dataflow model, Figure 2 (Upson [19], and Haber and McNabb [8]). Here, the information is said to *flow* through a series of processes being changed into a visual representation that is displayed as an image. A *filter* process selects the information required for this representation, the *map* process then exchanges the data into an Abstract Visualization Object (AVO) – an abstract (often geometric) form representing the selected information – which is then rendered and displayed as an image. This is a good general model and is used in many visualization systems, for example, AVS [19], IBM Data Explorer [13] and IRIS Explorer [9].

There are many models that are relevant to visualization, including user interaction, user requirements and communication models, see Figure 3. The 'display methods' focus on how the information is represented, they consider the visual primitives of the mapping stage (Figure 3A) and/or the form of the final image (Figure 3B).

We start be looking at the data and dimensionality of the display, as a simple method of categorizing a visual representation. We then describe a 'display primitive' classifi-

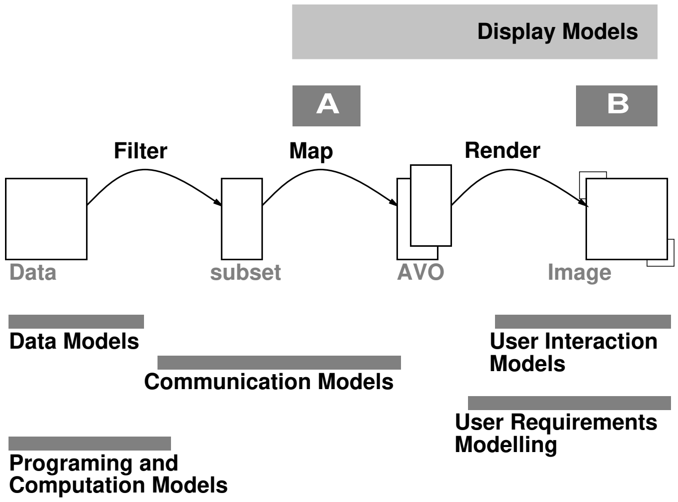

Figure 3: Models relevant to visualization. They are depicted how they relate to the dataflow paradigm.

cation, then include the 'output primitives', describe some 'underlying field' methods and look at display models for automated visualization design. Finally, we describe an algebraic realization of Bertin's graphical schema. Alternatively, the methods may be themselves grouped by their styles, into catalog and tabular models, graphical and finally algebraic models.

Data and Dimensionality

Our first classification categorizes the realizations by the dimensionality of the output. For example, a time line is a 1D visualization, a scatter plot is represented on a 2D plane, while a representation of a building is described in three dimensional space. Note, in this case, we are evaluating the dimensionality of the representation, and not of the image – as image space is 2D! The dimension is one classifier, but is is useful to include other methods to generate a greater range and therefore a more detailed classification.

Another classifier is the dimensionality of the data. For example, Earnshaw and Wiseman [6] provide such a general "Data and Display Dimensionality" classification scheme, Table 1.

As an additional qualification it is possible to evaluate

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)

Table 1: Examples classified by Data and Display Dimensions, after Earnshaw/Wiseman [6]. The table also includes the Scalar (S), Vector (V) and Tensor (T) categories of Collins [5] and notes some as being Multivariate (M) methods.

what style of data is being represented. For example, an imaginary dataset of construction materials may include a list of different parts stored with an associated location coordinate. This dataset represents scalar (or single valued data) in position space. In another dataset, air-flow may be monitored within a room of a building. In this example, the data is stored as position (3D coordinate) information with the direction of the air flow represented by three fields at every position: the data is represented by a vector field. Finally, the third field-type is named a Tensor, that enables stretch and compression forces to be stored.

Collins [5] extends Earnshaw and Wiseman's model to include the data types of Scalar (S), Vector (V) and Tensor (T). Table 1 presents these S,V and T classifiers with 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.

Further aspects of the data could be used as classifiers, such as the organisation of the data: whether regular, irregular, sparse or curvilinear. However, this then becomes an evaluation of 'data models'. It is useful to evaluate the data for use within the 'display models' and we revisit this in the "Underlying Field Models" section. Refer to [7, 18] for more information about data types, styles and management.

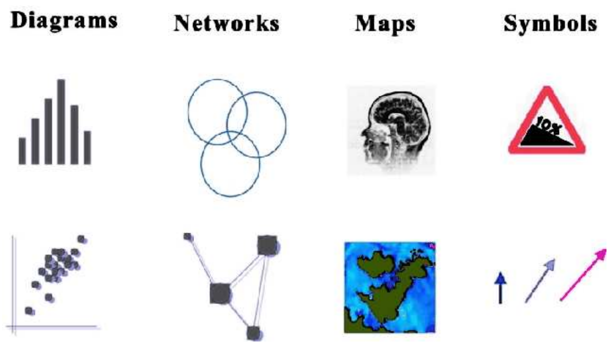

Figure 4: Diagrams, Networks, Maps and Symbol display classification, after Bertin.

This 'data and dimensionality' categorization is a useful method. Here every realization may be evaluated and included in the categorization. The scheme enables visualization methods to be compared. Indeed, it is a good way to evaluate the functionality of (say) different visualization systems. This is because it is easy to *see* what 'style' of data and output the system supports. However, such tabular techniques have a major disadvantage, that it is impossible to label individual representations.

Display Primitives

At a more abstract level than the previous tabular method, we describe the visualizations by their style of representation, distinguishing between (say) a diagram and a map. Bertin [1, 2] presents one such 'display primitive' model. He categorizes the displays as Diagrams, Networks, Maps and Symbols (DNMS) depicted pictorially in Figure 4.

Diagrams include bar charts, scatter plots, histograms and schematics;

Networks describe trees and path connections, for example.

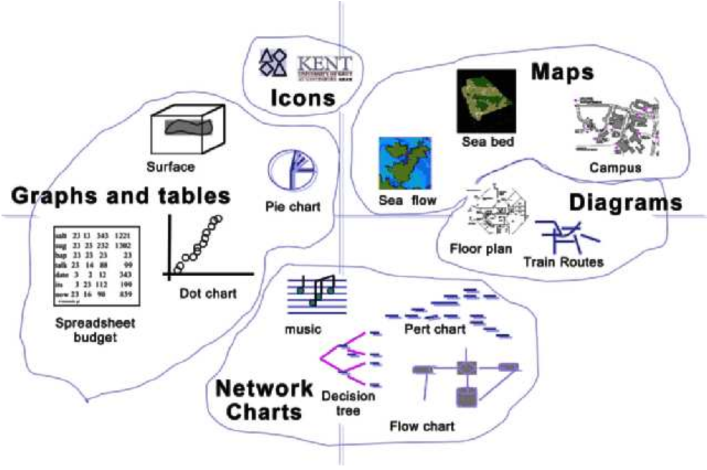

Figure 5: A mapping of visualization techniques as a graph, after Lohse [12]. These are similar to Bertin's Diagrams, Networks, Maps and Symbol categories.

Maps include any geographical maps and diagrams that the positions are constrained by a "real life" object. The maps often incur a non-uniform spatial 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 include signs and icons.

More recently Lohse et al [12], 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; similar groupings to the DNMS classification of Bertin. A scatter plot, of icons to networks (on one axis) against graphs, tables to maps and diagrams (on the other axis) was generated, Figure 5.

Such a DNMS classification is very useful, simple and the classifications may be labelled on the visualizations themselves. It is possible to further sub-divide the categories (such as the diagrams into specific types of schematics), but the finer detailed categories become more subjective. Additionally, it is useful to include the dimension, or number of components of the mapped-data, as shown in Figure 8.

Output Primitives

In order to generate a more detailed classification we include the output primitives within the DNMS categorization.

As we see from the aforementioned dataflow model, each component of the data is mapped into the AVO form.

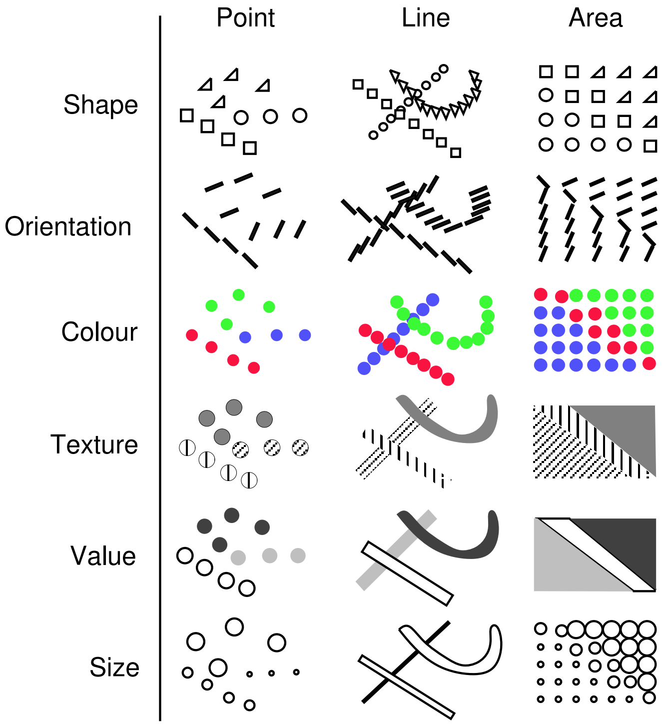

Figure 6: Retinal Variable Examples, after Bertin

Thus, the individual components are exchanged for an output primitive. Bertin [1, 2] describes six such representation methods, that he names *Retinal Variables*. These are shape, orientation, colour, texture, value and size. He also describes that the variables may be classified and configured as a point, line or area. Figure 6 describes the retinal variables as *points*, *lines* or *area*. It is also possible to include a volume configuration, this is relevant for modern 3D visualizations and displays.

In addition to the six retinal variables, Bertin includes position. Most of his examples were two dimensional pictures, however, it is possible to extend this method to include three (or possibly N) spatial dimensions – if this is understandable by the user. Other people have extended these six; indeed, colour may be separated into categories of Hue, Saturation and Brightness, and attributes such as transparency and animation (time) may be added to the list of primitives [11].

Component Organization

Bertin continues by designating a level of *organization* to each primitive. 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 2. 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*.

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

Table 2: Retinal Variables, after Bertin

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 $Q\%$ and logarithms $\log Q$.

Utilization of the Image Space

Additional to the *retinal variables*, the *configuration* (as points, lines or areas) and their *organization*, it is possible to describe how the mapped parameters utilize the image space. Indeed, Bertin describes different arrangements such as circular and irregular, and how the quantities are repeated or use the dimension of the plane. He represents these as lines, arcs and arrows, Figure 7.

An Iconic representation

Each image (or visualization) may be represented by an icon, made from the *configuration*, *organization* and *utilization* of the image space.

We extend Bertin's representation to include a Composite classification and allow the perspective drawing method (of Bertin) to represent three-dimensional space. 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 8 shows some visualization techniques within Bertin's classification structure.

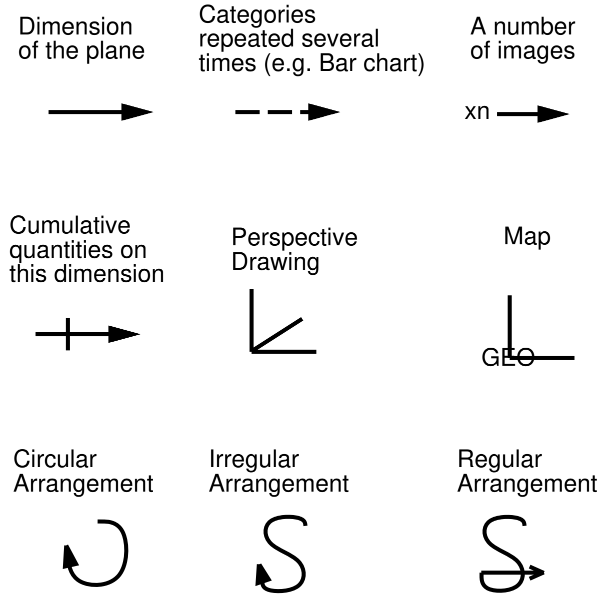

Figure 7: Utilization of the image space, after Bertin

The Organization of each component is not depicted in this large diagram, to keep the clarity of the schematic. However, 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 9 shows some examples with one particular Organization classification.

Bertin's 'icon' representation and classification mode are very useful. The icons may be described on a per-image basis, which allows the classification to be transmitted with the realization. Although it is more difficult to compare different representations using the symbolic representation, the categorization itself aids the developer to 'evaluate' and generate different representations (by, for example, exchanging the retinal variables). This model is able to categorize diverse methods such as pie and bar charts, scatter plots and three dimensional isosurface diagrams.

Underlying Field Models

Display models that evaluate solely the output image are fine, but perhaps the underlying data-fields should have more prominence within a display model. 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 10. He uses an algebraic expression to describe the underlying field and display.

Brodlie 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

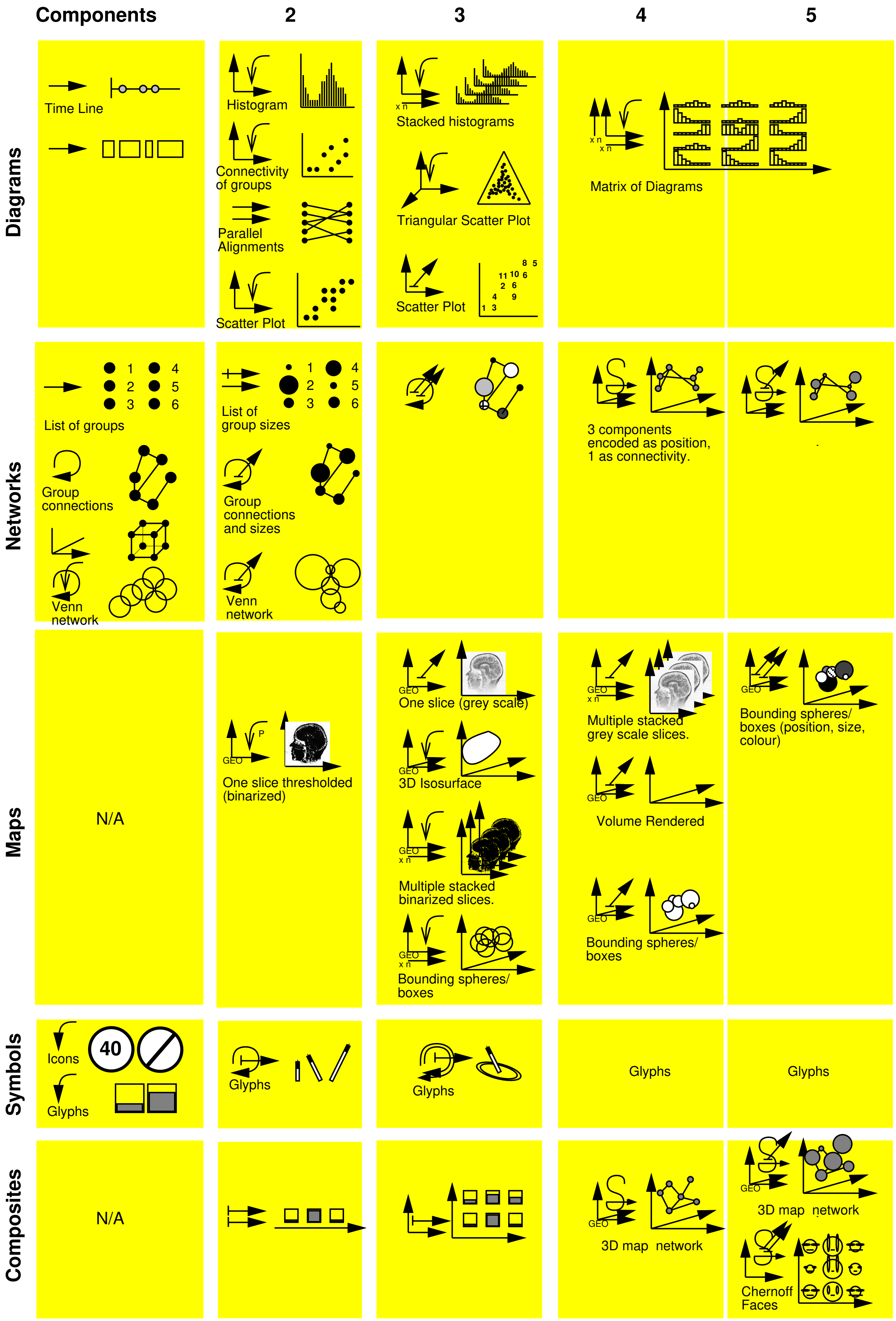

Figure 8: Output Classification Model, after Bertin, diagram representations by Roberts

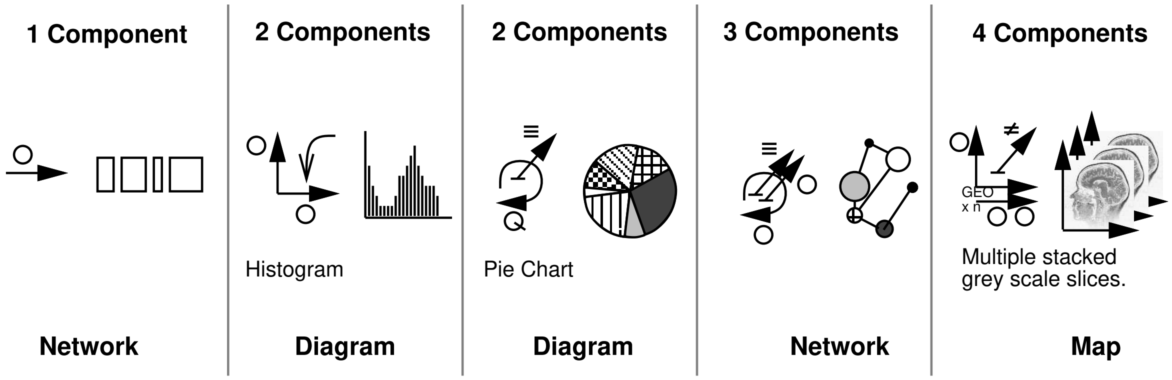

Figure 9: Output Classification Model with Component Organization, after Bertin

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.

Brodlie, in his model, distinguishes Ordered and Nominal categories, these are comparable to the Associative $\equiv$, Selective $\neq$, Ordered O and Quantitative Q categories of Bertin. It is worth to highlight the fact – as mentioned by Hibbard et al [10] – that scalar quantities in the data map to primitives that represent continuous values (such as time to animation or position) and data that is discrete is mapped to discrete display variables. Likewise, the ordered quantities may appropriately map onto orderable display primitives.

Further, when data is depicted with multiple variables it is mapped onto many different retinal variables. Now, some of the mappings are better at representing different styles of data (e.g. nominal or ordered), as seen in Table 2. Indeed, perception rules have been developed that address this issue – these are used in the Automated Visualization systems, see section below. It is possible to see that the representations of the information are more or less accurate to an 'ideal display'. Indeed the representations are 'approximations' as they are displayed with a finite resolution in space. Thus it is possible to form a Lattice of different realiztions 'ordered according to their information content' [10] to provide a notion of how precise the representations are compared to an 'ideal display'. Moreover, we could explain that the multiforms – generated through mapping different display primitives – are more, less or equal abstractions of the underlying information.

These 'underlying field' models enable the user to evaluate the data to provide an appropriate representation, in-

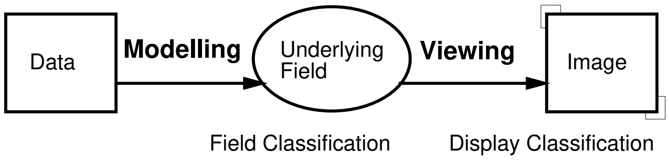

Figure 10: Underlying Field Visualization Process Model

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), Dimensions 1, 2, 3 etc., Scalar(S), Vector(V), Tensor(T)		

Table 3: Visualization Classification Examples, after Brodlie

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 4: Object and Operation Classes, after Wehrend and Lewis

deed, its not the data itself, we wish to represent, but the underlying phenomena. Such models form a useful grounding for the 'display models'. Additionally, with Brodlie's method, individual representations may be labelled. 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".

Categorize by tasks

A user observes the visualization by using 'perception tasks'. These may be described as the operations required to understand the retinal variables, such as working out a length of an object or its position in regard to the axis.

Such tasks are used by Wehrend and Lewis [22] who 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 4 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.

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 [16], 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 visualization 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 [14] 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.

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.

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).

Multiple images, or multiple views [15], 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.

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.

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.

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 $X^1(Mr^2R^1)$.

Examples and Summary

Table 5 lists some display methods with their appropriate Algebraic notation; the algebraic display methods are taken from the schematics in Figure 8.

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 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. Also the temporal domain is not explicitly included, this could be represented by the X quantity or another symbol added. However, this algebraic form provides a useful method to classify different visualizations.

	Display Methods	Algebraic Form
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, 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 5: Display Methods with an equivalent algebraic form.

Conclusion

Within this paper we have described many display models. The tabular methods are useful to categorize and compare the functionality of (say) a visualization system. Whereas Bertin's symbolic form and the algebraic methods enable individual images to be labelled.

The perception rules, of the automatic presentation tools, are extremely useful, and allow the developer to create appropriate representations.

Bertin developed some nice models for image classification and labelling, which are relevant for today's modern visualizations (perhaps with some minor adjustments and extensions). His symbolic representation, albeit descriptive, is probably not so practical, but an algebraic representation, such as presented within this paper, provides a useful equivalent labelling strategy.

Acknowledgements

A version of this paper was presented at the IEEE Information Visualization Conference 1999, in London. I am indebted to those people who have made suggestions and improvements of the above work.

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] B. Fortner. *The Data Handbook (Second Edition) – A guide to Understanding the organisation and Visualization of Technical Data*. Springer-Verlag, 1995.

[8] 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.

[9] 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).

[10] W. L. Hibbard, C. R. Dyer, and B. E. Paul. A lattice model for data display. In *Proceedings Visualization '94 – sponsored by the IEEE Computer Society*, pages 310–317, 1994.

[11] 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.

[12] 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.

[13] 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.

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

[15] J. C. Roberts. Multiple-view and multiform visualization. In R. F. Erbacher, P. Chen, J. C. Roberts, and C. Wittenbrink, editors, *Visual Data Exploration and Analysis VII, Proceedings of SPIE*, volume 3960. IS&T and SPIE, January 2000.

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

[17] J. A. Simpson and E. S. C. Weiner. *The Oxford English dictionary*. Oxford: Clarendon Press, 1989.

[18] L. A. Treinish. Unifying principles of data management for scientific visualization. In R. E. Earnshaw and D. Watson, editors, *Animation and Scientific Visualization – Tools and Applications*, pages 141–170. Academic Press, 1993.

[19] 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.

[20] G. Wallas. *The art of thought*. London, Jonathan Cape, 1858.

[21] L. R. Wanger, J. A. Ferwerda, and D. P. Greenberg. Perceiving spatial relationships in computer-generated images. *IEEE Computer Graphics and Applications*, 12(3):44–58, May 1992.

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