(NASA-CR-194879) A KNOWLEDGE BASED SYSTEM FOR SCIENTIFIC DATA VISUALIZATION Final Report (George Washington Univ.) 23 p

N94-23507

Unclas

G3/61 0203556

A Knowledge Based System for Scientific Data Visualization

Hikmet Senay and Eve Ignatius

Department of Electrical Engineering and Computer Science
The George Washington University
801 22nd Street, T-624A, N.W.
Washington, D.C. 20052

Abstract

This paper describes a knowledge-based system, called visualization tool assistant (VISTA), which was developed to assist scientists in the design of scientific data visualization techniques. The system derives its knowledge from several sources which provide information about data characteristics, visualization primitives, and effective visual perception. The design methodology employed by the system is based on a sequence of transformations which decomposes a data set into a set of data partitions, maps this set of partitions to visualization primitives, and combines these primitives into a composite visualization technique design. Although the primary function of the system is to generate an effective visualization technique design for a given data set by using principles of visual perception, the system also allows users to interactively modify the design, and renders the resulting image using a variety of rendering algorithms. The current version of the system primarily supports visualization techniques having applicability in earth and space sciences, although it may easily be extended to include other techniques useful in other disciplines such as computational fluid dynamics, finite-element analysis and medical imaging.

Introduction

Scientific data visualization has become an important discipline which supports scientists in exploring data, looking for patterns and relations, proving or disproving hypotheses, and discovering new phenomenon using graphical techniques. Although visualization has such an important role in scientific data processing, it is still an art which requires significant knowledge in various fields, such as data management, computer graphics, perceptual psychology, and visual art. Most scientists whose work involves data visualization usually lack this knowledge. Furthermore, there are very few guidelines for selecting and creating effective visualization techniques. In order to facilitate the visualization process, there is a strong need for systems that embody the required knowledge and provide guidance to their users.

The visualization process can generally be viewed as a sequence of transformations that convert a data set into a displayable image. There are typically three transformations in this process [Haber88]: (1) data manipulation, (2) visualization mapping, and (3) rendering. Data manipulation includes such operations as gridding and interpolation which convert a given data set into a form that is suitable for subsequent visualization operations. Visualization mapping defines an abstract visualization technique by establishing a set of bindings between the data manipulated in the previous stage and the visualization primitives such as positional parameters, color, texture, and animation. The primary objective of visualization mapping is to identify a set of visualization primitives that can effectively convey the informational content of data. During the final stage, an image of the data is rendered, according to the design produced by the visualization mapping, using such operations as projection, shading, and hidden surface removal. While there are several algorithmic techniques that are useful in the data manipulation and rendering phases, there is no well-defined methodology to guide the visualization mapping process. Since the effectiveness of a visualization technique is primarily determined by the bindings established at this stage, the visualization mapping is the most important stage of the visualization process. This is the stage at which scientists need considerable guidance in the design of effective data visualizations.

Although the existing visualization systems such as apE [Crusi87], NGS [Trein88], FOTO [Cogni89], AVS [Upson89], and DataScope [NCSA89] provide the basic mechanisms to establish such bindings, they generally do not provide much guidance on generating the most effective visualization technique for a given data set. For instance, the data flow diagrams that are conveniently used to specify visualization techniques in apE and AVS may also lead to the design of ineffective visualization techniques due to their seemingly unlimited flexibility. Assuming that the potential users of visualization tools are not visualization experts, a methodology leading to the design of effective visualization techniques is needed. Even though this need has been acknowledged by many researchers developing visualization tools, the design of visualization techniques has mostly been done in an ad hoc fashion. This is primarily attributable to our limited understanding of how the human visual system works and what types of visual presentations can most effectively harness its capabilities. Recently, a methodology that matches data characteristics and interpretation aims with properties of visualization techniques using a natural scene paradigm [Rober90] has been proposed to guide the design of visualization techniques for effective data presentation. Methodologies of this sort are indeed essential in order to simplify the scientific data visualization process for the average user.

In this paper, a visualization tool assistant (VISTA) which employs a compositional design methodology is described. In order to ensure the effectiveness of its designs, VISTA uses a large number of rules, mostly heuristic in nature, that have been acquired through literature surveys and discussions with visualization experts [Senay90]. While the primary function of VISTA is to automatically generate an effective visualization technique design for a given data set by applying these rules in a systematic fashion, it also allows users to interactively modify this design, and renders the resulting image using a variety of rendering algorithms. The development of VISTA is based on the research in graphical perception [Berti83, Cleve85, Tufte83] and extends the design methodology of APT [Macki86], a presentation tool for two dimensional graphics, to three dimensions.

Knowledge for Scientific Data Visualization

Visualization essentially offers another level of abstraction at which scientific and engineering data can be represented and manipulated quite effectively. In a sense, it has a similar role to that of a database schema for providing insight into a complex information space. Because of its ability to present a large amount of data at once, a visual data representation and manipulation scheme provides a much higher bandwidth in data processing than that of any other alternative. Since the process leading to the design of data visualization techniques involves several transformations which require knowledge in a variety of disciplines, the process of generating an effective visual presentation of data is difficult. Furthermore, there is no unique visual presentation that can convey the entire informational content of data. Alternative visualization techniques must be used to convey different aspects of the data. In general, the creation of an appropriate data visualization and its accurate interpretation requires knowledge about data, computer graphics, and visual perception.

As the term "data visualization" implies, data and graphics are the essential components of scientific data visualization techniques. In the visualization mapping stage where a visualization technique is defined abstractly, various characteristics of data dictate the type of graphical primitives that must be used to express the data content. For instance, a scalar field may be effectively displayed using color whereas a vector field cannot be represented using color alone.

Knowing the characteristics of data relevant for visualization mapping is the first important step in designing an effective data visualization technique. The most primitive components of visualization techniques are the marks (graphical symbols) which constitute the graphics. In general, a visualization technique can be viewed as a collection of marks which collectively encodes a particular data subset. While it is possible to describe visualization techniques in terms of simple marks, such as points, lines, areas, and volumes, a higher level description based on commonly used primitive visualization techniques which encode simple relations among a few data variables is more practical. It is also possible to combine several primitives for visualizing a multi-dimensional data set. Similar to grammar rules in a language, there are rules, albeit implicit and not well-known, for combining the visualization primitives. Knowledge about various graphical primitives as well as composition rules is essential for data visualization using advanced computer graphics techniques. Since graphical primitives, rather than the data itself, are presented to convey the data content, it is important that the graphical primitives used in visualization precisely express only the data. There are visualization techniques that are incapable of presenting certain phenomena that may exist in the data. For instance, particles do not necessarily show twisting effects in a flow field. If it is desirable to display twisting effects in a flow field, the use of flow ribbons is more appropriate than the use of particles [Shirl89]. In general, knowledge of this sort which is heuristic in nature and relates to how human perceptual system reacts to different graphical primitives is crucial for effective data visualization.

Based on an analysis of existing data visualization techniques [Senay91], the knowledge necessary for scientific data visualization has been classified into five knowledge categories: (1) data characteristics, (2) visualization vocabulary, (3) primitive visualization techniques, (4) composition rules, and (5) rules of visual perception. In the rest of this section, each of these knowledge categories is described in further detail.

Data Characteristics

Scientific data can broadly be classified into two groups: qualitative and quantitative. Qualitative data is further subdivided into two groups: nominal and ordinal. Nominal data types are unordered collections of symbolic names without units. For instance, the names of the orbiters, such as Hubble, Magellan, Mariner, Viking, and Voyager form a nominal data set. Ordinal data types are rank ordered only, where the actual magnitudes of the differences are not reflected in the ordering itself. A typical example of an ordinal data set is the names of the calendar months, January through December.

Compared to qualitative data, quantitative data is more common in all scientific disciplines. Quantitative data is typically classified along two dimensions; (1) based on the number of components which make up the quantity, and (2) based on the scales of the values. Along the first dimension, quantitative data can be scalar, vector, or tensor. Scalar data types possess a magnitude, but no directional information other than a sign. They are simply defined as single numbers. Vectors have both direction and magnitude. Quantitatively, their mathematical representation requires a number (equal to the dimensionality of the coordinate system) of scalar components. In general, a vector is a unified entity. This implies that the problem of visualizing vector fields is not equivalent to the problem of displaying independent, multi-variate scalar fields. The number of components which specify a tensor depends on the dimensionality of the coordinate system and the order of the tensor. Along the other dimension, quantitative data can be classified as interval, ratio, and absolute [Kossl89]. Interval data scales preserve the actual quantitative difference between values (such as farenheit degrees), but do not have a natural zero point. Ratio data scales are like interval scales but they do have a natural zero and can be defined in terms of arbitrary units. For instance, two hundred dollars is twice as much as one hundred dollars. Absolute data scales are also ratio scales which are well-defined in terms of non-arbitrary units, such as inches, feet, and yards.

Other important attributes of data, that play a role in selecting visualization primitives, include functional dependencies among data variables, spacing between sampling points, cardinality of the data set, upper and lower bounds of values, units of measurement, coordinate system, scale and continuity of data.

Visualization Vocabulary

The visualization vocabulary identifies the basic building blocks of scientific data visualization techniques. In general, a mark (any graphical symbol that is visible on a display medium) is the most primitive building block that can encode some useful information in data visualization. Each mark can be classified as either simple or compound. There are four types of simple marks: points, lines, areas, and volumes. A point has a single conceptual center that can indicate a meaningful position, a line has a conceptual spine that can indicate a meaningful length or connection, an area has a single conceptual interior that can indicate a meaningful region or cluster of marks, and a volume has a single conceptual interior that can indicate meaningful space in three dimensions. Of the four simple mark types, the first three are identified by Bertin [Berti83] as being the most primitive components of two dimensional graphics and used by Mackinlay [Macki86] to automate the design of graphical presentations. The fourth mark type, volume, is a natural extension of Bertin's classification to three dimensions and appropriate when the third dimension can be perceived effectively. A compound mark is a collection of simple marks which form a single perceptual unit. For instance, contour lines, glyphs, flow ribbons, and particles are all compound marks. A useful analogy is that simple marks are like letters in the alphabet, whereas compound marks are like words in a dictionary.

Data content is generally encoded by varying the positional, temporal, and retinal properties of marks in a visualization technique. A positional encoding of information is a variation of the positions of the marks in the image. A temporal encoding of information is a variation of the mark properties over time. A retinal encoding of information is any variation of the "retinal" properties of the marks that the retina of the eye is sensitive to independent of the position of the marks. The retinal properties of marks include size, texture, orientation, shape, and the three parameters of color, namely hue, saturation, and brightness.

The marks can further be classified as to whether they represent single or multiple data variables and single or multiple data points. A single variable (SV) mark is associated with one variable, whereas a multiple variable (MV) mark is associated with several variables. A single data (SD) mark conveys a single value for a single data point, whereas a multiple data (MD) mark shows a range of summary information regarding the local distribution of several data points. This classification is particularly useful when visualizing large multi-variate data sets.

Primitive Visualization Techniques

Primitive visualization techniques are those which encode one dependent and up to four independent variables. Additional variables (dependent or independent) that may exist in a given data set can further be encoded by manipulating retinal properties of marks within the primitive visualization techniques or equivalently combining two or more primitive visualization techniques into a single design. In general, each primitive visualization technique can be classified into one of three categories, that is, positional, temporal and retinal, depending on the primary mark property that is manipulated by a given technique. Positional techniques can be one, two, and three dimensional such as single axis, contour plot, and surface diagram, respectively. There is only one temporal technique, namely, animation, and a set of retinal techniques that corresponds to the set of retinal properties of marks such as shape, size, orientation, texture, and color. Following the analogy made previously between simple (or compound) marks and letters (or words), the primitive visualization techniques may be viewed as forming simple sentences in a language. The primitive visualization techniques that are supported by VISTA are shown in Figure 1.

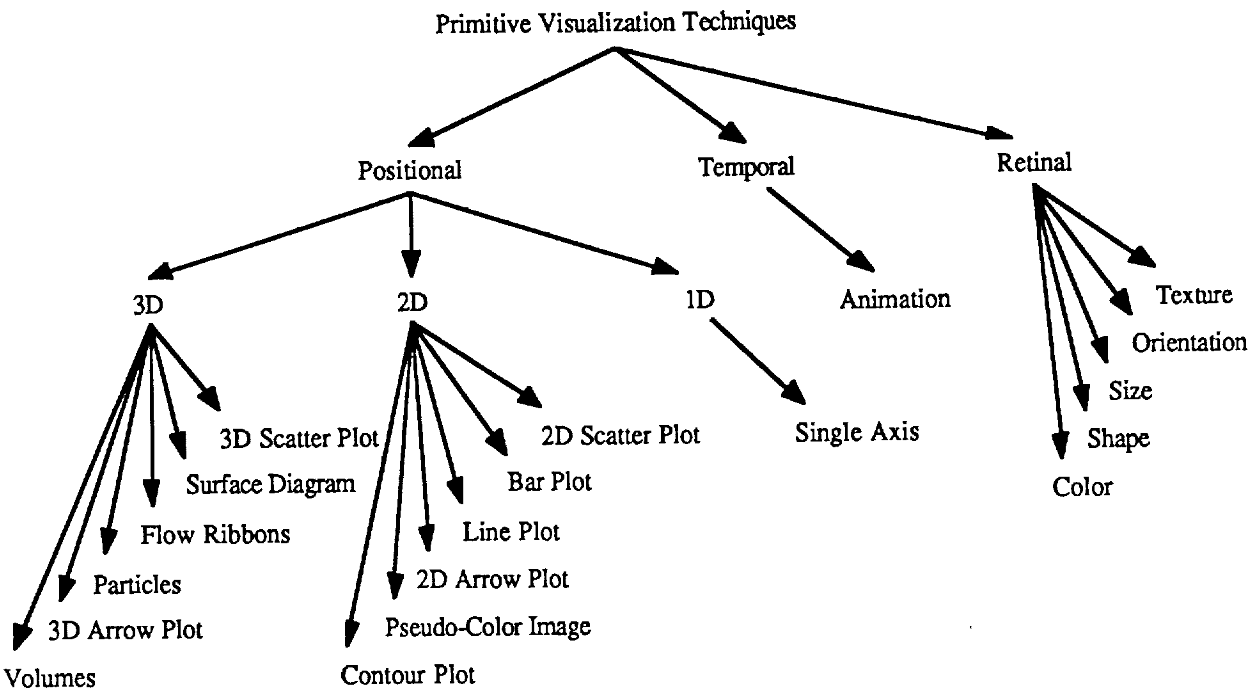


Figure 1. Primitive Visualization Techniques.

Although some of these techniques may be considered to be compositions of others, it is more appropriate to include them in the set of primitive visualization techniques rather than to construct them in terms of others. For instance, an arrow plot may be viewed as a composition of a scatter plot and a shape, where the scattered points have an arrow shape. The set also contains techniques that manipulate more than one of the positional, temporal, and retinal properties of marks. For instance, a pseudo-color image is basically a positional technique, which also uses one or more of the color parameters, namely hue, saturation, and brightness.

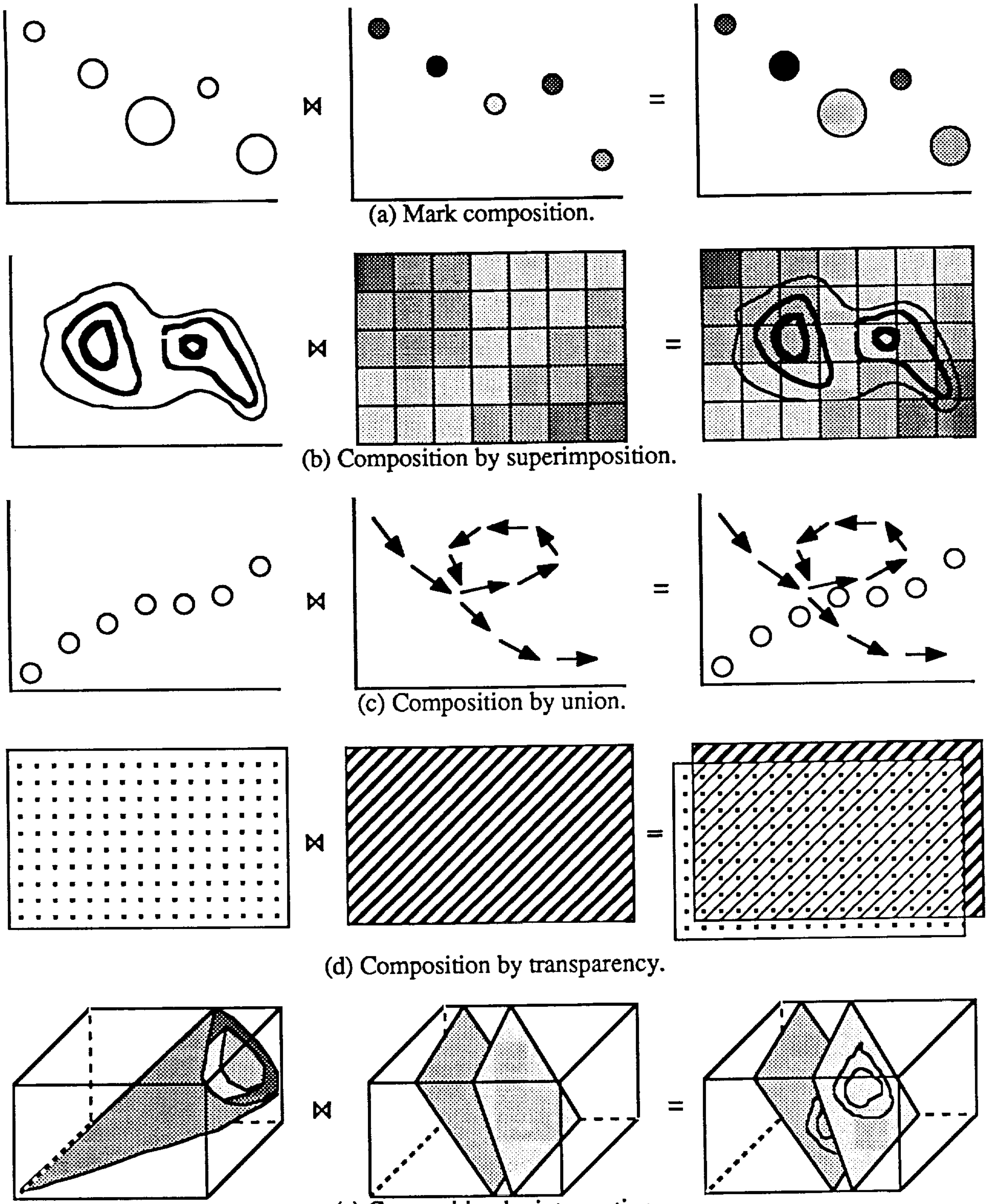
Composition Rules

Information spaces that are of interest to scientists and engineers are mostly multi-dimensional. Even though such information spaces can be investigated in a traditional manner by looking at a small number of dimensions one at a time and repeating the process until all dimensions are correlated, it is often more beneficial and desirable to visualize several, or if possible all, dimensions at once. Fortunately, the recent advances in computer graphics hardware and software have made the visualization of multi-dimensional data sets a reality. There are already examples of visualization techniques that effectively display several data variables at once. Our analyses of these techniques have revealed a set of composition rules that describes the generation of composite visualization techniques from primitives. These rules not only describe how a composite design is obtained from a set of primitive visualization techniques, but may also be used to generate novel visualization techniques for multi-dimensional data sets.

In general, the composition rules define conditions under which a pair of visualization techniques can be combined to form composite visualization techniques for displaying multi-dimensional data. Each of these rules also describes a combination scheme that can effectively convey the correlation between data variables encoded by the component visualization techniques being combined. The conditions associated with each rule test (1) the compatibility of component visualization techniques, (2) the visibility of each component upon composition, and (3) the distinguishability of components in the composite design. The compatibility conditions check whether the component visualization techniques have identical axes, that is, the axes are associated with the same data variables, have the same upper and lower bounds, units of measurement, coordinate systems, and scales. Furthermore, the compatibility of marks of the components are tested by each rule to determine whether a mark property used to encode a data variable by one component is constrained by the other in such a way as to prevent the composition associated with the rule. The visibility conditions check whether an acceptable portion of both component visualization techniques would be visible after composition. The acceptable visibility is determined based on the characteristics of data being visualized, the type of primitive visualization techniques involved in the composition, and the amount of data that can be seen interactively after composition. For instance, if the data is either discrete or discontinuous then all marks of the component visualization technique should be visible. If the data is continuous and visualized with a primitive visualization technique, such as a contour plot, which already takes advantage of the continuity, then again, all marks of the primitive visualization technique should be visible. If only a small portion of data would be visible after composition then interaction methods, such as cutting planes, adjustable transparency, and peelable surfaces, must be available to allow the user to view the entire data. The final class of conditions associated with composition rules checks whether the marks of one component visualization technique involved in the composition process are distinguishable from the marks of the other component by manipulating an appropriate retinal property of the marks. For instance, if two contour plots are to be combined then the contour lines belonging to different contour plots must be distinguishable. Since color and size (or line thickness) are the only manipulable retinal properties of contour lines, either the color or the thickness of lines belonging to one contour plot should be different than that of the other contour plot.

Although additional or alternative composition rules may be identified through further analyses of existing visualization techniques, our experience shows that the following six composition rules are useful in describing and generating a large number of composite visualization techniques.

1. **Mark composition** merges marks of the component visualization techniques by pairing each and every mark of one technique with a compatible set of marks of the other (see Figure 2a). For mark composition to be applicable, each component involved in the composition must be manipulating different and nonconflicting mark properties. The number of marks in the composite design resulting from mark composition is equal to the number of marks in one of the components having a larger mark set. Despite the fact that the number of marks in the composite is less than the total number of marks in the components, all marks of both components remain visible and distinguishable after composition. Another important property of mark composition is its commutativity. That is, the resulting composite visualization technique is invariant regardless of the order in which the components are combined.
2. **Composition by superimposition** merges marks of the component visualization techniques by superimposing one mark set onto the other (see Figure 2b). For composition by superimposition to be applicable, all data fields encoded by one of the components must be continuous. Furthermore, the component visualization technique encoding these continuous fields should mostly be visible through the marks of the other component. After superimposition, the number of marks in the composite design is equal to the sum of marks in both components. Since the marks of one component are superimposed onto those of the other, composition by superimposition is not commutative.
3. **Composition by union** combines marks of a pair of component visualization techniques by using set union (see Figure 2c). This is by far the simplest composition technique since it forms the mark set of the composite by using each and every mark of the components. For it to be applicable, the marks belonging to each component must be at least partially distinguishable in space. This requirement is satisfied when the marks of each component visualization technique are either sparsely distributed in the same coordinate system or positioned at a distance from the marks of the other component along one axis. In the former



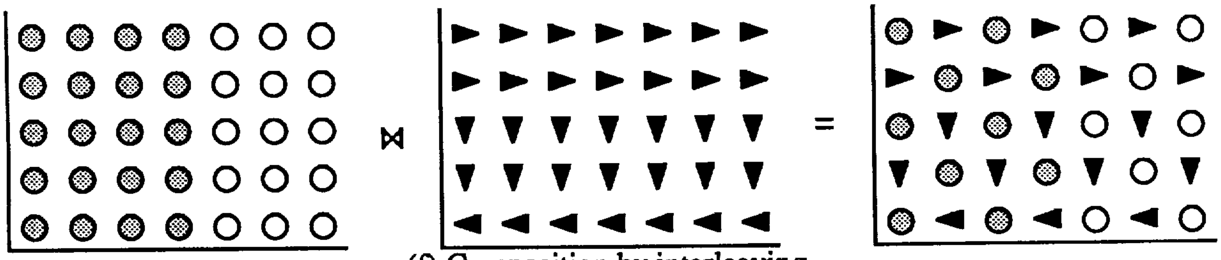
(a) Mark composition.

(b) Composition by superimposition.

(c) Composition by union.

(d) Composition by transparency.

(e) Composition by intersection.



(f) Composition by interleaving.

Figure 2. A simple graphical illustration of composition techniques. In each case, the pair of graphs on the left represents the component visualization techniques and the graph on the right represents the result of composition.

case, the marks belonging to different components must further be distinguishable by at least one retinal property such as color or shape. In the latter case, the use of an additional retinal property is not necessary since the marks belonging to different components are clearly separated positionally. The number of marks in the composite design resulting from composition by union is equal to the sum of marks of both components. Like set union, composition by union is also commutative.

4. **Composition by transparency** combines a pair of visualization techniques by manipulating the opacity values of marks belonging to either or both visualization techniques (see Figure 2d). The opacity of marks in each component visualization technique may either be set to a constant value for insuring visibility of all marks at all times or be altered under user control for increasing the contribution of one technique or the other to the composite image. Unlike the continuity requirement in composition by superimposition, the data fields encoded by either component visualization techniques need not be continuous for composition by transparency. After the composition, the number of marks in the composite design is effectively equal to the sum of marks in both components. However, the number of marks that are visible at any point in time is a function of the opacity values associated with each mark set at that point. The operator associated with the composition by transparency rule is in general commutative provided that the positional (spatial) parameters of marks belonging to either component are kept unchanged. If, on the other hand, one of the component is translated in any spatial dimension, for instance, to bring that component to the foreground, the composition by transparency becomes noncommutative.
5. **Composition by intersection** combines a pair of visualization techniques by first computing their intersection and then superimposing the intersection onto one of the components (see Figure 2e). For composition by intersection to be applicable, at least one data field encoded by each component must be continuous. Furthermore, the marks of both components should be densely populated in the same viewing space in a way to make it hard to view both visualization techniques simultaneously. In general, composition by intersection reduces the number of data points visualized in a viewing space. Thus, the number of marks of the composite visualization technique is less than the total number of marks in the components but greater than that of the component onto which the intersection is superimposed. The data points that are missing from the composite can subsequently be visualized provided that appropriate interaction techniques are available to manipulate the composite. However, the interactive manipulation of the composite may not even be necessary when it is possible to mentally interpolate the missing data points from the intersection points. Since the intersection of marks is superimposed onto the marks of one component, composition by intersection is not commutative.
6. **Composition by interleaving** selects marks of the composite technique by alternating between the marks of the component techniques (see Figure 2f). This is by far the least common composition scheme among all. Even though its effectiveness may be questionable,

there is a small number of designs illustrating its use. Composition by interleaving is typically applicable if each component encodes a continuous field using a large number of marks. The number of marks in the composite design is one half of the total number of marks in both components. Composition by interleaving is also not a commutative composition scheme.

Rules of Visual Perception

The primary objective in data visualization is to gain insight into an information space by mapping data onto graphical primitives. In general, such mapping defines an abstract visualization technique for a given data set in terms of marks, primitive visualization techniques, and composition rules. However, there are several possible mappings which lead to different visualization technique designs. Selecting and creating the most effective design among all alternatives for a given situation usually requires considerable knowledge about principles and rules of visual perception. While the knowledge about characteristics of data, such as types, units, scales, and spacing among measurement points, as well as graphical primitives, which eventually form a composite design, is important in constructing visualization techniques, the knowledge about how the resulting image is perceived and which aspects of data can or cannot be conveyed by the design is essential for effective presentation of the data. Even though the latter type of knowledge is crucial for the design of effective visualization techniques, it is imprecise, incomplete, and inexact, and thus, hard to formalize. Existing theories on visual perception offer valuable guidelines for effective visualization, but they are often limited in scope because of the inherent complexity of the human visual system. In the absence of in-depth theories on visual perception, visualization designers often have to resort to heuristic rules that are developed through experience and experimentation.

Through literature surveys and discussions with visualization experts, several heuristic rules that are useful in visualization technique design have been acquired. In general, these rules relate to the expressiveness and effectiveness of visualization primitives, the use of descriptors that aid in understanding the meaning of visualizations, and other issues, such as scaling, image sequencing, depth perception, and handling peculiarities in data. While the expressiveness and effectiveness rules determine the characteristics of visualization mappings, the remaining classes of rules enhance the images corresponding to those mappings. The expressiveness rules identify visualization primitives that are capable of expressing the desired information, whereas the effectiveness rules identify those primitives which are the most effective in a given situation at exploiting the capabilities of the output medium and the human visual system [Macki86]. For instance, an expressiveness rule, that can be used to determine how magnitude of vectors can be conveyed in an arrow plot without using the length of arrows, states that "If there is a wide range of magnitudes in a vector field, then let each arrow represent the local direction of the vector field, but make all arrows the same length. Let such attributes as color, line thickness, or the vertical projection of the arrow be used to represent the vector magnitude" [Haber88]. A typical effectiveness rule states that "Our visual perception system is better tuned to quantitative understanding using geometry rather than color" [Berti83]. Thus, a visualization technique that primarily uses geometry, such as a surface diagram, to display quantitative data is more effective than a technique that primarily uses color, such as a pseudo-color image, in presenting the same data.

Despite their importance in visualization, the rules of visual perception are far from being complete and extensive. Furthermore, some of the rules are somewhat controversial and conflicting due to the lack of formal evaluation criteria in their development. Nevertheless, these rules provide a basis for generating effective data visualization techniques in the absence of in-depth theories. However, there is still a strong need for formal studies to identify additional rules and to refine the existing ones.

VISTA Architecture

The software architecture underlying the operation of VISTA is based on the visualization process model having three transformational phases. The main components of the architecture corresponding to the phases of this model are: (1) *data unit* through which data manipulations can be performed, (2) *design unit* which defines a visualization mapping between data and graphical primitives, and (3) *rendering unit* which produces an image of the data according to the visualization mapping defined in the design unit. The overall system architecture, including various modules in each unit along with the input and output elements of those modules, are shown in Figure 3. While conforming to the visualization process model, the architecture further supports modularity in visualization software development. This modularity facilitates a clear separation of data manipulation, visualization mapping, and rendering issues in data visualization. Since VISTA has been developed to assist users in the visualization mapping phase, rather than the data manipulation or rendering phases, the design unit forms the core of the architecture. Although the data and rendering units are not central in the VISTA architecture, both units have been designed to support complete automation of the visualization process. Following the specification of data to be visualized, VISTA can generate an appropriate visualization technique and render the corresponding image automatically. Since visualization is essentially an interactive process, each unit also provides interaction tools for users to modify the visualization techniques or to manipulate the images that are generated by the system.

Data Base

VISTA assumes that the data to be visualized is stored in Common Data Format (CDF) which has been developed by the National Space Science Data Center (NSSDC) at NASA Goddard Space Flight Center [Gough88]. For simplifying the overall system design, the data is further assumed to be stored in regularly gridded form. The VISTA database (DB) provides both data and its description using various attributes that characterize data at a high-level of abstraction. Both data and design units operate on data description. Only the rendering unit requires access to the actual data.

Data Unit

The function of the data unit is to provide users with a set of operators for interactive data manipulation. In the current implementation, the only operators supported are the data selection operators which allow the users to specify a data subset for visualization. A data subset can be obtained from the database by selecting several data variables from one or more data sets that may be gridded differently. Selection of data sets and data variables from these sets are done through dynamically organized menus. The order of variable selection defines an importance ordering in visualizing the selected data subset. However, the user has the option and the facilities to redefine the importance ordering. Even though the current implementation of the data unit is limited to the data selection operators, the modularity of VISTA architecture facilitates a straightforward extension of the data unit to include a full range of data manipulation operators.

Design Unit

Starting with a data subset which is output from the data unit, the design unit generates a composite visualization technique by using a three step design process. Once the composite visualization technique is generated, it can be modified within the design unit by the user.

In the first step of the design, the data subset is decomposed into a set of simple data partitions each of which can be visualized by a primitive visualization technique. The partitioning is based on functional decomposition in relational data base theory [Ullma80]. Essentially, each data

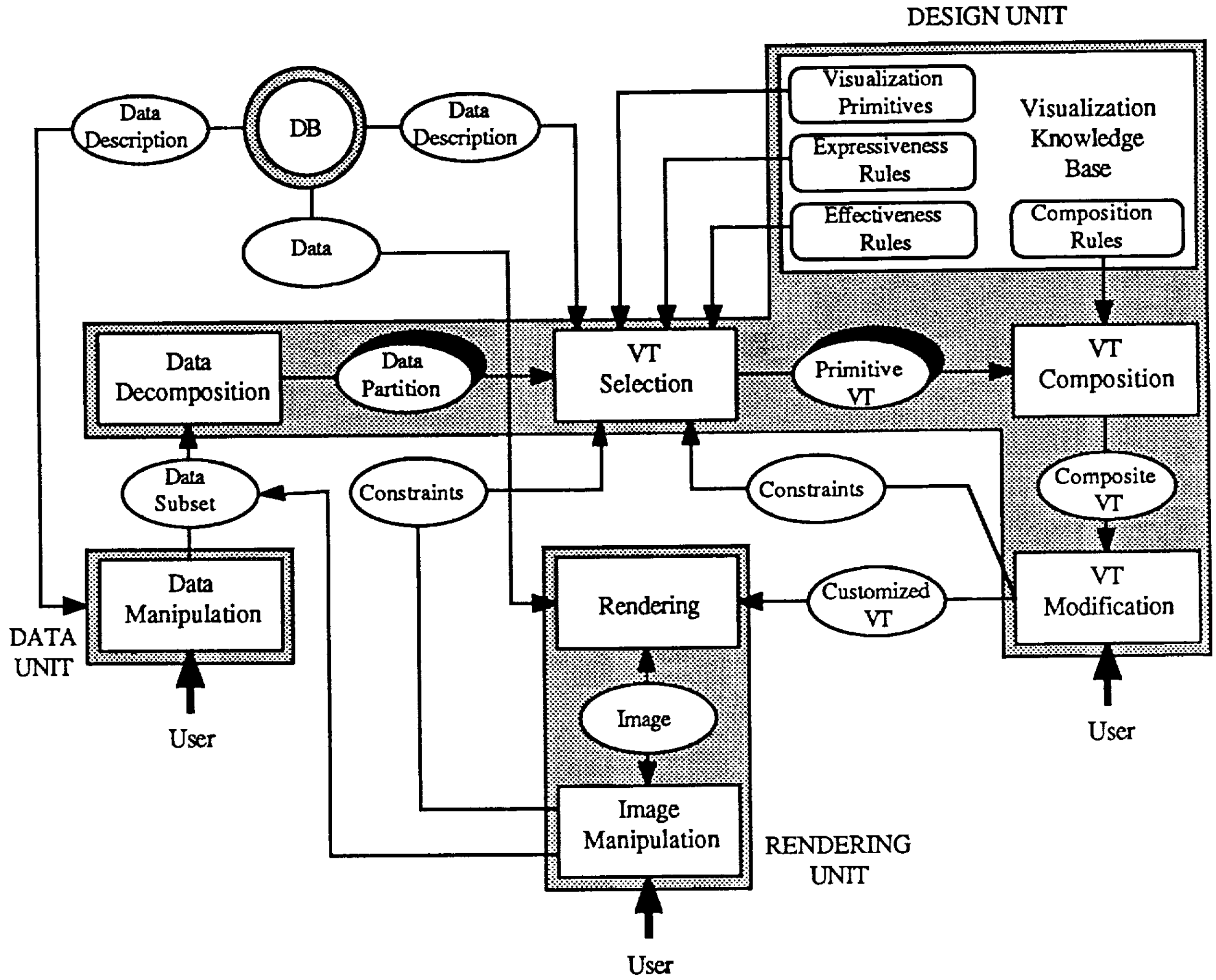


Figure 3. The VISTA architecture.

partition corresponds to a relation in one of the following forms having one dependent and up to four independent variables: (1) X → A, (2) X,Y → A, (3) X,Y,Z → A, or (4) X,Y,Z,T → A.

In the second step of the design, a primitive visualization technique is found for each partition by the visualization technique selection module using the expressiveness and effectiveness rules. In order to facilitate the selection process, the search space for primitive visualization techniques is organized in such a way that the primitive visualization technique that can most effectively convey the most important data partition is selected first. For each data partition, the visualization technique selection module iteratively examines the search space depth-first from left to right until it finds an appropriate technique that has not already been used to visualize another partition or not ruled out because of design constraints imposed by the user. Although the initial design of the composite visualization technique, including the initial selection of primitive visualization techniques, is done without any constraints, subsequent design modifications that are directed by the user may bring constraints into the visualization technique selection process. If and when such constraints are imposed, the visualization technique selection module tries to find a primitive visualization technique for each data partition without violating those constraints. Following a successful mapping between a data partition and a primitive visualization technique, a data structure in the form of a frame describing characteristics of the mapping is passed to the visualization technique composition module.

In the final step of the design, the visualization technique composition module combines a pair of component visualization techniques by applying appropriate composition rules. The component visualization techniques are either the primitives that have been generated in the preceding design step or the composites that have been formed previously by composition of other components. The visualization technique composition module attempts to apply composition rules in the order that they appear in the preceding section by checking the compatibility, visibility, and distinguishability conditions associated with each rule. If any one of these conditions are not met for a specific composition, an alternative rule is tried until either the composition is successful or all composition rules are exhausted without success. If the visualization technique composition module is unable to combine a pair of components, then it initiates backtracking in an attempt to find an alternative design. This backtracking may result in the selection of an alternative primitive visualization technique for a data partition in the selection module or the application of an alternative composition rule for a previously combined component pair.

Once all primitive visualization techniques corresponding to the data partitions are combined, a design description of the composite visualization technique is presented to the user for possible modifications that may improve the design. If a modification is simple, that is, does not violate compatibility, visibility, and distinguishability requirements while preserving the consistency of the composite design generated by the system, it can be handled within the visualization technique modification module without ever going back to the preceding steps of the design. Otherwise, it imposes a constraint that must be satisfied in the visualization technique selection module. Depending on the type of changes that must be made to satisfy a constraint, any arbitrarily complex part of the composite may be redesigned starting from a selection step. The design process terminates with a customized visualization technique, encoding the entire data subset selected at the data manipulation stage, upon instruction from the user.

Rendering Unit

A design description of the customized visualization technique that is output from the design unit forms the primary input for the rendering unit. The main function of the rendering unit is to create an image according to the design description by using appropriate rendering algorithms. Each rendering algorithm creates an image that corresponds to a primitive visualization technique in the design by accessing data stored in the data base. Once all images corresponding to the primitives in the design are created, they are combined using operators associated with the composition rules in the design description. The image resulting from this process is then displayed on the screen of the user's terminal.

The other important function of the rendering unit is to provide interactive facilities for image manipulation. The image manipulation operators, that are available in the current implementation, are confined to various viewing and image modification operations. Using the viewing operations that include rotation, translation, zooming and others, the user can interactively change the viewpoint and see the image from different perspectives. The image modification operators are functionally similar to those provided for visualization technique modification in the design unit. However, all image modification operations are graphically performed on the actual image, rather than, on its abstract description. Using the image manipulation operators, the user can select any component of the visualization technique corresponding to a data partition and modify its attributes, including the retinal properties of its marks, interactively. Each component visualization technique can also be removed from the image in order to reduce image complexity or be brought back, if it was removed previously. Furthermore, the image can be modified drastically by applying a different composition rule from the one that was used to combine a particular pair of components in the original design.

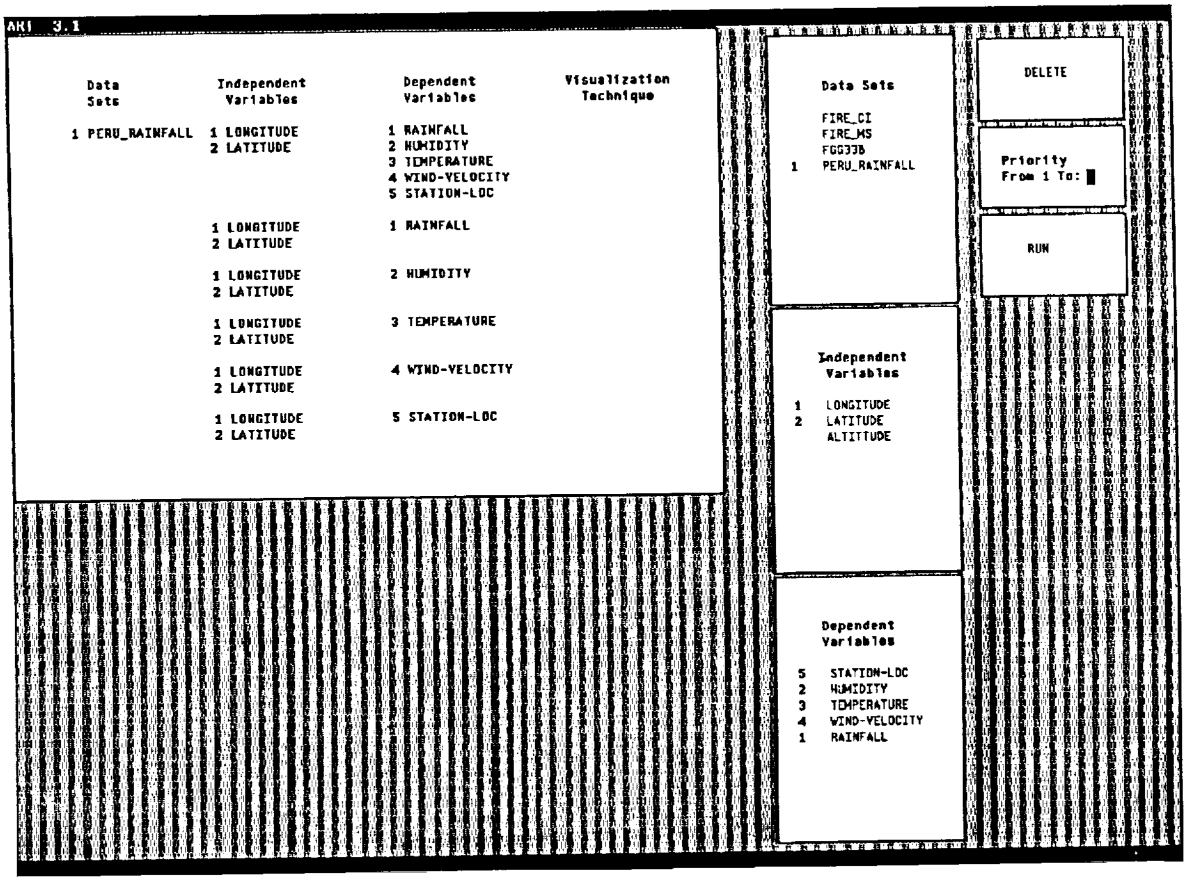
Even though the current implementation provides a small set of rather simple operators for image manipulation, the modularity of VISTA architecture facilitates an easy extension of the rendering unit to include a full range of image manipulation operators. The modular structure of VISTA also makes the addition of sophisticated rendering algorithms to the current framework possible without much effort.

Implementation

A major part of VISTA architecture, including data and design units, has been implemented using Automated Reasoning Tool (ART), a Lisp-based knowledge engineering system, on a Sun 3 workstation. The rendering unit has been implemented in C language using Graphics Library (GL) routines on an SGI 4D/210GTX workstation. In this implementation, components of data, such as data subsets and partitions, primitive and composite visualization techniques are represented using frames while composition, effectiveness, and expressiveness rules are represented using condition-action pairs. The communication between the rendering unit and the other units are handled by passing message files through the network.

Visualization Technique Design

In order to illustrate the operation of VISTA in more detail, a sample visualization technique design is presented in this section. The sample data to be visualized is a small subset of a rainfall data set obtained in Northwest Peru during the 1982-83 El Nino period. The data set contains 3 independent and 5 dependent variables that are gridded regularly.



Data Sets	Independent Variables	Dependent Variables	Visualization Technique
1 PERU_RAINFALL	1 LONGITUDE 2 LATITUDE	1 RAINFALL 2 HUMIDITY 3 TEMPERATURE 4 WIND-VELOCITY 5 STATION-LOC	
	1 LONGITUDE 2 LATITUDE	1 RAINFALL	
	1 LONGITUDE 2 LATITUDE	2 HUMIDITY	
	1 LONGITUDE 2 LATITUDE	3 TEMPERATURE	
	1 LONGITUDE 2 LATITUDE	4 WIND-VELOCITY	
	1 LONGITUDE 2 LATITUDE	5 STATION-LOC	

Data Sets

FIRE_CI
FIRE_MS
FGG33B
1 PERU_RAINFALL

Independent Variables

1 LONGITUDE
2 LATITUDE
ALTITTUDE

Dependent Variables

5 STATION-LOC
2 HUMIDITY
3 TEMPERATURE
4 WIND-VELOCITY
1 RAINFALL

Priority
From 1 To:

Figure 4. A snapshot of data selection screen.

Since the data selection is the first step of the visualization process, VISTA starts a visualization session by presenting a menu listing all data sets that are accessible through the data unit. Following the selection of a data set from the menu, VISTA forms two additional menus from which independent and dependent variables, belonging to the data set, can be selected one by one. The order of variable selection from each of these menus defines an importance ordering that can be used during the visualization technique selection steps. If it is desirable to include variables from other data sets, the user can always select another data set from the first menu and continue choosing among its dependent and independent variables. As the data selection proceeds in this manner, the currently selected data subset along with its partitions is displayed in a data selection summary window. A snapshot of a data selection screen including the summary window, the three selection menus, and a few control buttons is shown in Figure 4. The number appearing on the left hand side of each data variable indicates the relative position of that variable in the importance ordering. For instance,"rainfall" and "station-location" are respectively the most and the least important variables among the currently selected dependent variables in Figure 4.

Since data partitions are formed by interleaving decomposition and selection of the data subset, the next design step following the completion of data selection is mapping data partitions to appropriate primitive visualization techniques. The mapping starts with the data partition that includes the most important dependent variable. For instance, the data partition having "longitude" and "latitude" as independent and "rainfall" as dependent variables would be considered first in the sample design. There are several primitive visualization techniques, such as surface diagram, contour plot, and pseudo-color image, that can be used to convey the informational content of the data partition under consideration because each of these primitives satisfies the expressiveness requirements. The next thing to consider is the effectiveness of each candidate primitive. In the absence of any information about the tasks to be supported by the resulting visualization technique, the effectiveness rules would qualify the surface diagram as the most effective primitive. Thus, the data partition associated with "rainfall" distribution should be encoded by the position of marks on a surface diagram. The selection of surface diagram, rather than contour plot or pseudo-color image, is essentially based on the following effectiveness rules:

(1) Position is more effectively perceived than color;

(2) Our visual perception system is better tuned to quantitative understanding using geometry rather than color; and

(3) Contour plots require some effort on the part of the viewer to establish quantitative relations between different contour levels - it is not always obvious whether a local extremum is a minimum or a maximum.

The mapping process continues for the remaining data partitions unless there is a previously generated visualization technique, either primitive or composite, with which the recently selected primitive can be combined. Since there is no possible composition at the current state of the sample design, the data partition that includes the second most important dependent variable in the importance ordering must be processed next. Accordingly, the data partition having "longitude" and "latitude" as independent and "humidity" as dependent variables is processed next in the sample design. This data partition has similar characteristics to those of the first data partition. Thus, it can be visualized using one of the three primitives, all satisfying expressiveness requirements, that were considered before for the visualization of the "rainfall" data partition. Since the most effective primitive visualization technique, surface diagram, has already been used in the design, a contour plot should be selected to encode the data partition being processed. The decision to select a contour plot rather than a pseudo-color image is again based on the effectiveness criteria. Once the type of primitive visualization technique is identified, for instance, as being a contour plot, the next step is to determine the values of its attributes including a mark set, retinal properties of marks, how data values can be conveyed most effectively and so on. For a contour plot, this step involves defining the contour levels that bounds the mark set, selecting a retinal property that can most effectively convey the data values associated with contour lines, and setting the values of the remaining retinal properties of the mark set. Based on the effectiveness criteria, line thickness (size) is chosen to be the most effective retinal property for conveying the data values associated with contour lines in the current design. Furthermore, the number of contour levels is defined to be equal to 5, which is the default value, and the color of contour lines, being the only unused retinal property, is set to the default value for color.

After a primitive is completely specified, it is combined with another visualization technique, either primitive or composite, that has been formed before, provided that the most recent primitive is not the first one. This means the contour plot that has just been generated should be combined with the surface diagram that was generated previously in the sample design. The combination of these techniques can be achieved based on either the composition by superimposition or the composition by union rules. In the current design, VISTA uses the composition by superimposition rule since it is the first applicable rule in the rule application order. The result of composition is a composite design that can be represented by a visualization tree.

A visualization tree is a data structure similar to a parse tree that is commonly used to represent the meaning of a sentence in a formal language. For instance, a visualization tree representing the sample composite design obtained by combining a surface diagram and a contour plot is shown in Figure 5. In a visualization tree, the leaf nodes correspond to primitive techniques and the internal nodes, including the root, correspond to composite techniques. The values of the relevant attributes of an internal node, which completely describe a composite design, are computed based on its components by applying the operator specified in that node. For instance, the application of the superimposition operator to the components, vt-1 and vt-2, determines the values of the technique, partition, x-axis, y-axis, z-axis, and mark-set attributes of the sample composite design using the composition by superimposition rule.

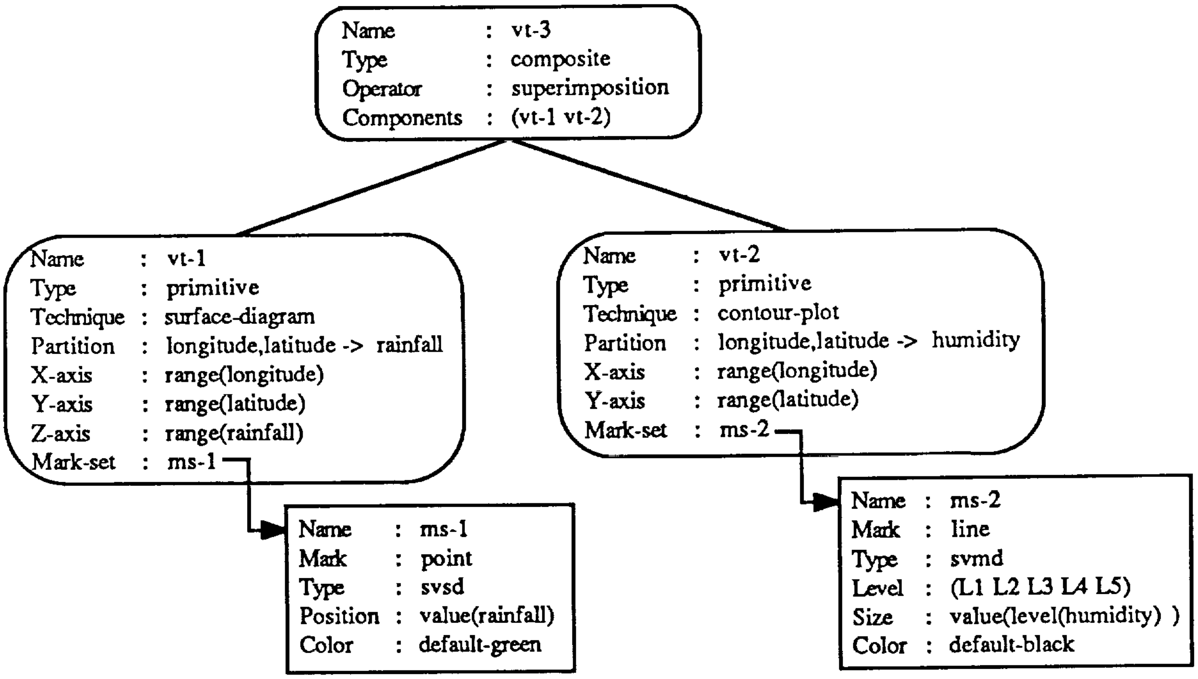


Figure 5. A visualization tree representing the composite formed by combining two primitives.

Although the design encoding the entire Peruvian rainfall data is not complete yet, the partial design given in Figure 5 completely describes an image that can be rendered even at this point. If the current design had been passed to the rendering unit, the resulting image would have been as shown in Figure 6.

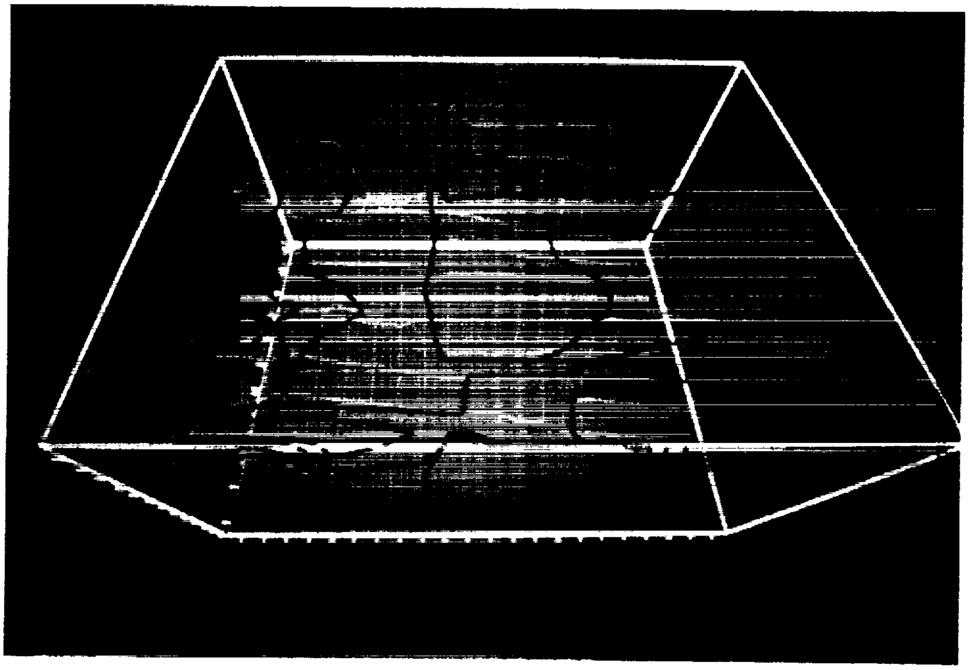


Figure 6. The image corresponding to the partial design shown in Figure 5.

A visualization tree grows bottom-up until all data partitions are encoded by the leaves of the tree. The composite design represented by a complete visualization tree essentially encodes the initial data subset selected for visualization. For instance, the composite design encoding the Peruvian rainfall data set is outlined by the visualization tree in Figure 7. Rather than describing the composite design in full detail, the tree shows only a summary of the design for the sake of concise presentation. In the actual design, each node of the visualization tree has several attributes similar to those shown in Figure 5.

Following the completion of a composite design that may be used to visualize a given data set, a summary of the design is presented to the user in the textual form for possible modifications. Any part of the design can be accessed through a menu-driven interface and be modified before rendering the corresponding image. The interface also provides a set control buttons that may be used for passing design constraints to the visualization technique selection module or initiating the rendering process if the current design is acceptable.

Upon initiating the rendering process, the rendering unit automatically generates an image of the composite by accessing data from the data base and executing a set of graphic routines associated with primitive visualization techniques and composition operators. For instance, given the composite design shown in Figure 7, the rendering unit generates the image shown in Figure 8.

Even if the initial design has not been altered before rendering, there are several ways it can be modified interactively once the corresponding image is generated. Most of these modifications are intended to provide alternative views of the data being visualized while others can be performed to change retinal properties of each primitive visualization technique in the image. The modifications providing alternative views of the data generally fall into two operation categories: (1) data reducing, and (2) image compositing. All operations in both categories are performed through a menu-driven interface.

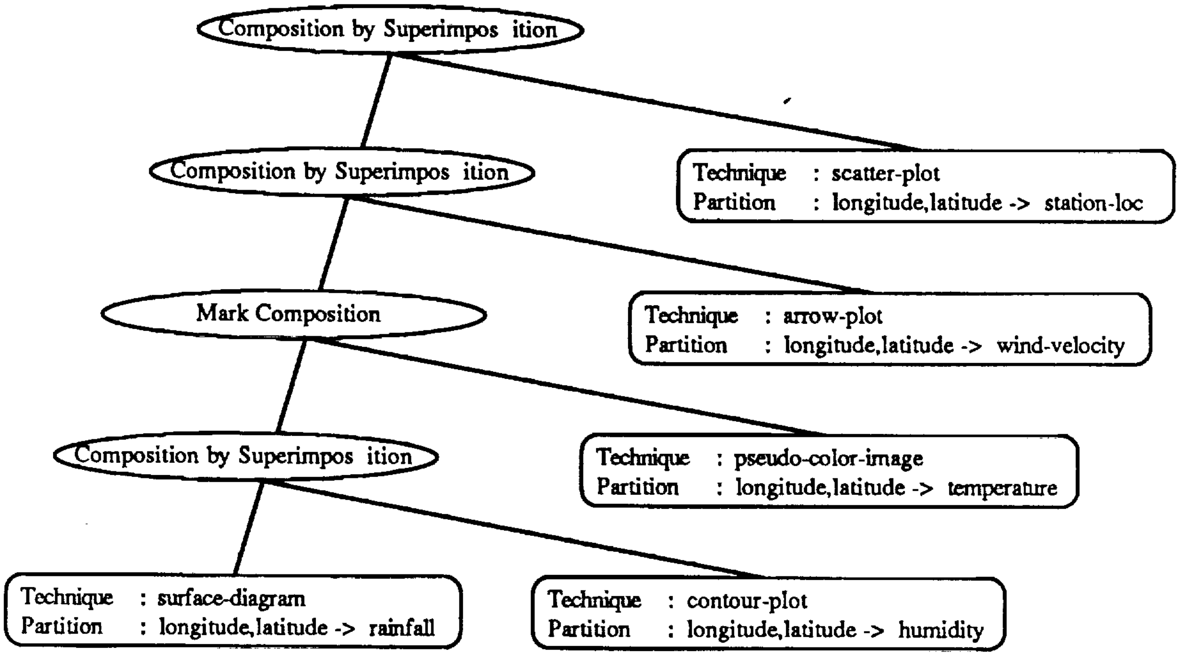

Composition by Superimpos ition
Composition by Superimpos ition
Technique : scatter-plot
Partition : longitude,latitude -> station-loc
Mark Composition
Technique : arrow-plot
Partition : longitude,latitude -> wind-velocity
Composition by Superimpos ition
Technique : pseudo-color-image
Partition : longitude,latitude -> temperature
Technique : surface-diagram
Partition : longitude,latitude -> rainfall
Technique : contour-plot
Partition : longitude,latitude -> humidity

Figure 7. A visualization tree describing the composite design that encodes the Peruvian rainfall data.

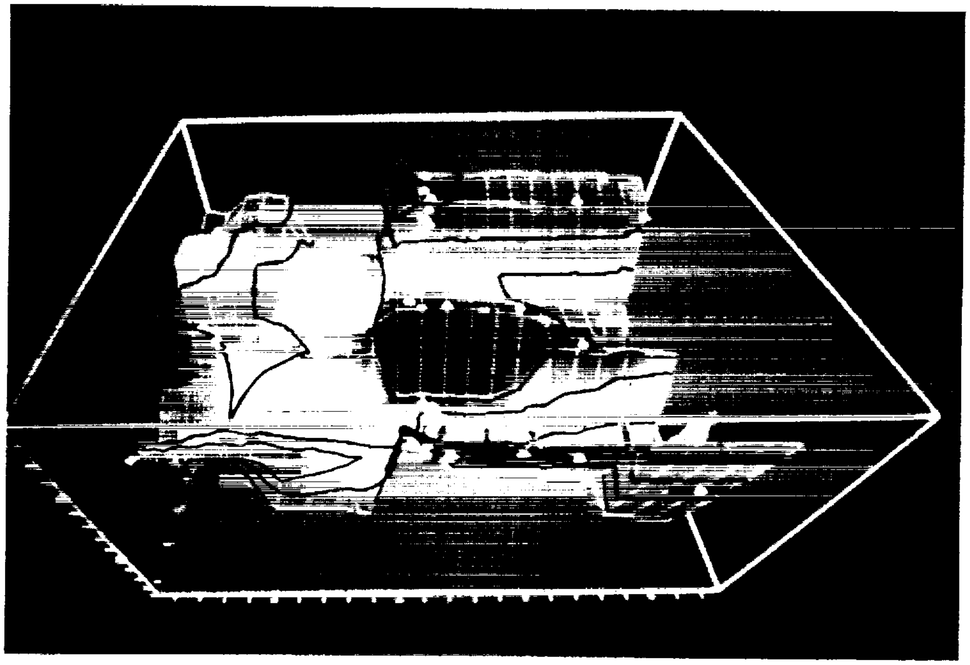


Figure 8. A visualization of the Peruvian rainfall data according to the design description shown in Figure 7.

The data reducing operations simply reduce image complexity by removing one or more primitives from the image when it is desirable to correlate a few data variables. The data reducing operations have inverses, that is, it is possible to bring any primitive removed from the image back. For instance, the image encoding only the "rainfall" and "temperature" partitions after data reduction is shown in Figure 9. Like any other image generated by the rendering unit, an image created after data reduction has a corresponding visualization tree that is a subtree of the original visualization tree. For instance, the visualization tree of the image obtained after data reduction is the subtree having two leaf nodes corresponding to the "rainfall" and "temperature" partitions that are combined by a mark composition operator in Figure 10. To show the correlation between original and reduced visualization trees, the nodes removed from the original along with the composites that cannot be viewed as a result of data reduction is shown as empty nodes in Figure 10. Empty nodes corresponding to the primitives and composites removed from the original image can be filled with the previous contents of those nodes by applying the inverse of data reduction operators. For instance, if the primitives highlighted by thick lines are included in the design again, an image corresponding to the visualization tree with four primitives can be rendered as shown in Figure 11.

The image compositing operations allow the user to apply alternative image composition rules in an attempt to get a somewhat different view of the data. Each composition operator in a visualization tree can be replaced by an alternative composition operator as long as they are compatible. For instance, if the composition by superimposition operators in the original design are replaced by the composition by union operators, another image encoding the same data set alternatively can be obtained as shown in Figure 12.

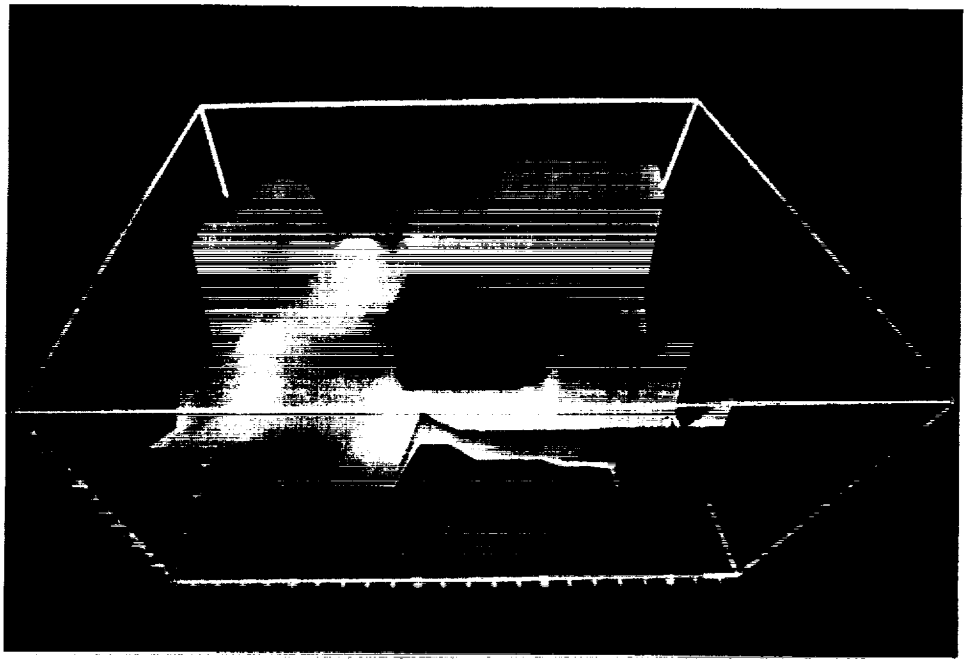


Figure 9. The image encoding the "rainfall" and "temperature" partitions after data reduction.

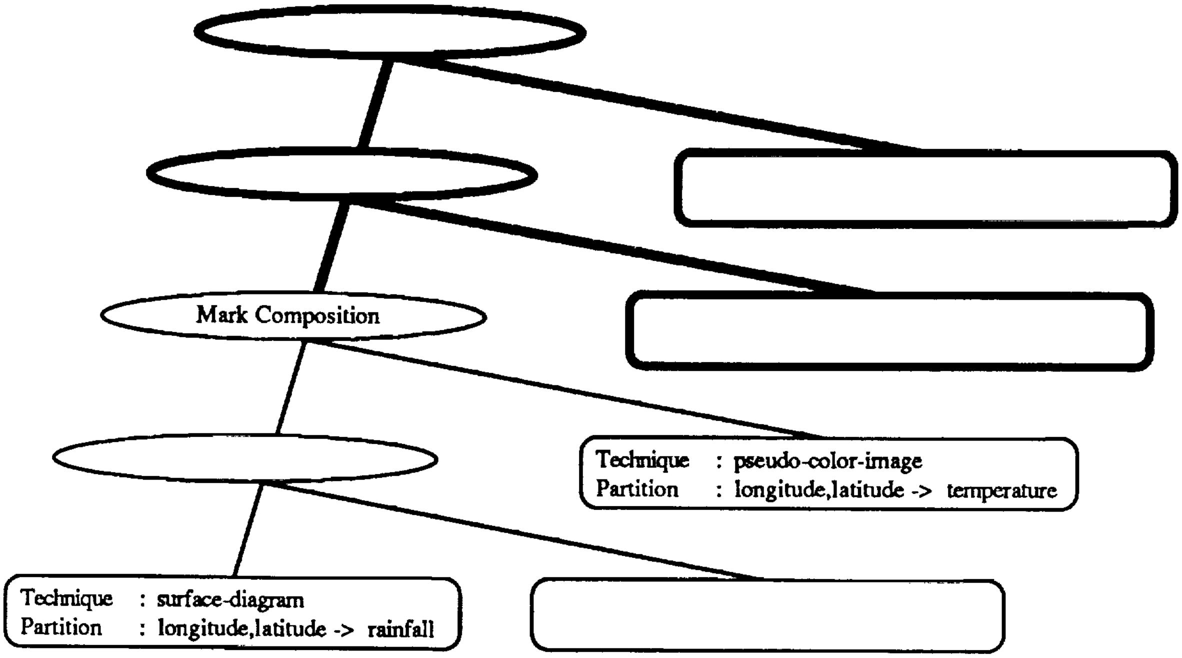

Mark Composition
Technique : pseudo-color-image
Partition : longitude,latitude -> temperature
Technique : surface-diagram
Partition : longitude,latitude -> rainfall

Figure 10. A visualization subtree having two primitives and its relation to the original visualization tree. The highlighted part of the tree shows the effect of including two more primitives.

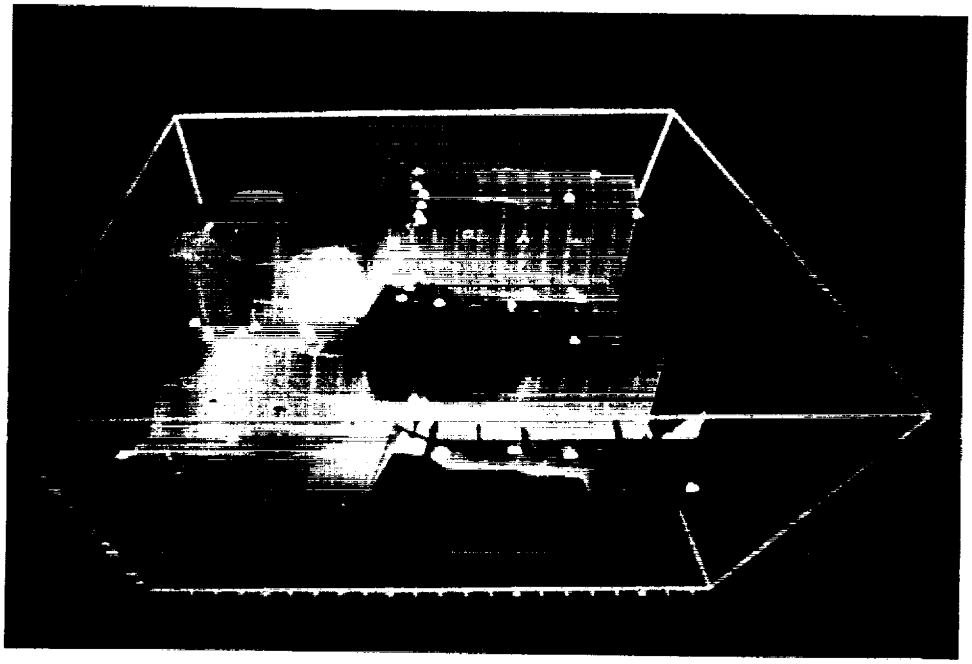


Figure 11. The image encoding the "rainfall", "temperature", "wind-velocity", and "station-location" partitions after the application of inverse data reduction operations.

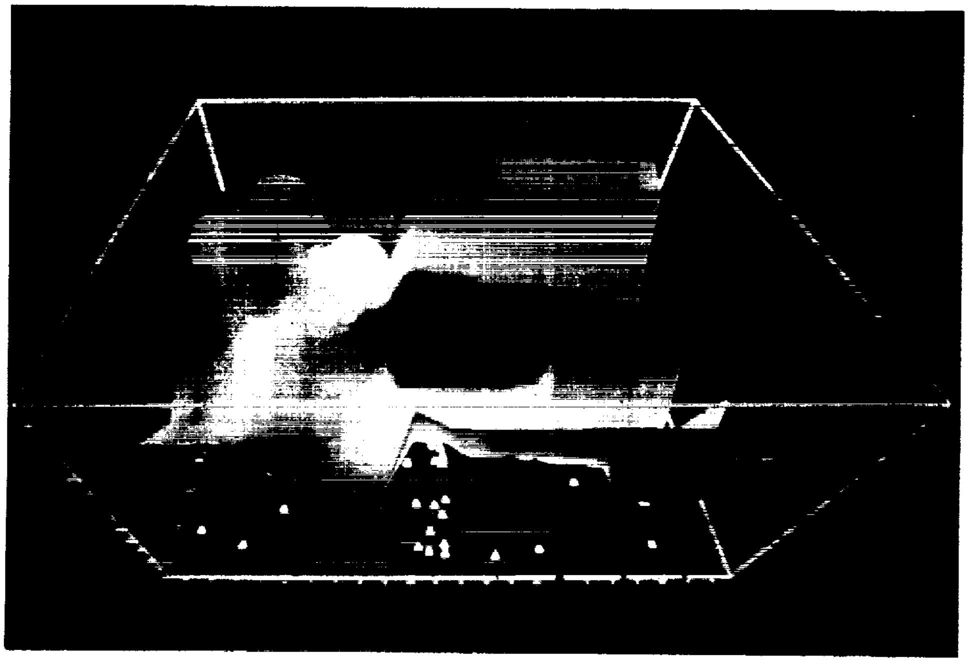


Figure 12. An alternative visualization of the Peruvian rainfall data after substituting a union operator for each superimposition operator in the original design given in Figure 7.

Conclusions

Scientific data visualization is a complex process requiring significant knowledge in a variety of disciplines. In order to facilitate this process, it is essential to develop methodologies which provide support and guidance to users. Based on a compositional design methodology having three data transformation phases, a knowledge-based system (VISTA) was developed to assist scientists in the design of scientific data visualization techniques. The system derives its knowledge from several sources which provide information about data characteristics, visualization primitives, and effective visual perception. The design methodology employed by the system is based on a sequence of transformations which decomposes a data set into a set of data partitions, maps this set of partitions to visualization primitives, and combines these primitives into a composite visualization technique design. Although the primary function of the system is to generate an effective visualization technique design for a given data set by using principles of visual perception, the system also allows users to interactively modify the design, and renders the resulting image using a variety of rendering algorithms. The software architecture underlying the system operation conforms to the visualization process model and is highly modular. While clearly separating the data manipulation, visualization mapping, and rendering tasks in data visualization, the modularity of the system simplifies potential modifications and extensions of the modules performing these tasks.

Although VISTA is capable of providing design support and guidance for scientific data visualization, there are several areas in which its capabilities can be extended considerably. First, the knowledge utilized by VISTA is not complete by any means. In particular, the current implementation of the knowledge base do not provide information about the effective use of color and animation, and the selection of lighting and projection schemes in visualization. Furthermore, only a small number of primitives are implemented. Second, VISTA does not consider the characteristics of user tasks to be supported by visualization in the design process. In general, visualization supports a variety of data processing tasks, for instance, identification of certain attributes of data, clustering a set of data points, and correlating a set of data variables. However, not all visualization techniques can be used to perform a given task with the same effectiveness. The utility of visualization, like any other data presentation, is a function of the task to be performed by the user [Casne91]. Hence, it is important that visualization software must provide support and guidance for users to select the most appropriate visualization technique for each task. The third area of extension is the integration of VISTA with other data processing tools and techniques such as statistical analysis software packages and simulation programs. Such integration is crucial because visualization is just a small and integral part of the overall data processing activity.

Acknowledgements

The authors would like to thank Professors James D. Foley and John L. Sibert who made valuable suggestions during the design and development of VISTA. Special thanks to Santos Lazzeri who implemented the rendering unit. The VISTA project was funded by USRA Center of Excellence in Space Data and Information Sciences (CESDIS) under NASA contract S/C 550-64.

References

[Berti83] Bertin, J., *Semiology of Graphics,* University of Wisconsin Press, Madison, WI, 1983. Translated by W. Berg and P. Scott from *La Graphique et le Traitement Graphique de l'Information,* Flammarion, Paris, 1977.

[Casne91] Casner, S.M., "A Task-Analytic Approach to the Automated Design of Graphic Presentations," *ACM Transactions on Graphics*, Vol. 10, No. 2, April 1991, 111-151.

[Cleve85] Cleveland, W.S. & McGill, R., "Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods", *Science*, 229, pp.828-833, Aug. 1985.

[Cogni89] Cognivision Incorporated, FOTO Reference Manual, Cognivision, Inc., Westford, MA, 1989.

[Crusi87] Crusi , C., et al., "apE: A Flexible Integrated Environment for Supercomputers and Workstations," *Proceedings of the Third International Symposium on Science and Engineering on Cray Supercomputers*, Sept. 1987, 533-588.

[Gough88] Gough, M., Goucher, G. & Treinish, L., CDF Implementers Guide, National Space Science Data Center, NASA Goddard Space Flight Center, Greenbelt, MD, 1988.

[Grins88] Grinstein, G., Pickett R. & Williams, M., "EXVIS: An Exploratory Visualization Environment," *Proceedings Graphics Interface '89*, CIPS,Toronto, pp. 254-261, 1989.

[Haber88] Haber, R.B., "Visualization in Engineering Mechanics: Techniques, Systems, and Issues", *Visualization Techniques in the Physical Sciences*, SIGGRAPH'88 Tutorial Notes, pp. 89-111, Aug. 1988.

[Infer87] Inference Corporation, *ART Reference Manual*, Inference Corporation, Los Angeles, CA, 1987.

[Kossl89] Kosslyn, S., "Understanding Charts and Graphs", *Applied Cognitive Psychology*, Vol. 3, 1989, 185-226.

[Macki86] Mackinlay, J.D., *Automatic Design of Graphical Presentations*, PhD Thesis, Dept. of Computer Science, Stanford University, Stanford, CA, 1986.

[NCSA89] National Center for Supercomputer Applications (NCSA), NCSA DataScope Reference Manual, NCSA, University of Illinois, Campaign-Urbana, IL, 1989.

[Rober91] Robertson, P.K., "A Methodology for Choosing Data Presentations," *IEEE Computer Graphics and Applications*, May 1991, 56-67.

[Senay90] Senay, H. & Ignatius, E., "Rules and Principles of Scientific Data Visualization", Technical Report No: GWU-IIST-90-13, Department of Electrical Engineering and Computer Science, The George Washington University, Washington, D.C. 20052. Also in *State of the Art in Scientific Visualization*, SIGGRAPH'90 Tutorial Notes, Aug. 1990.

[Senay91] Senay, H. & Ignatius, E., "Compositional Analysis and Synthesis of Scientific Visualization Techniques," *Conference proceedings of Computer Graphics International (CGI' 91)*, Boston, MA, June 1991.

[Trein88] Treinish, L.A., "An Interactive, Discipline-Independent Data Visualization System", Tech. Report: NSSDC NASA/Goddard Space Flight Center, 1988.

[Tufte83] Tufte, E., *The Visual Display of Quantitative Information*, Graphics Press, Chesire, CT, 1983.

[Ullma80] Ullman, J.D., *Principles of Database Systems*. Computer Science Press, 1980.

[Upson89] Upson, C., Faulhaber, T., Kamins, D., Laidlaw, D., Schlegel, D., Vroom, J., Gurwitz, R. & van Dam, A., "The Application Visualization System: A Computational Environment for Scientific Visualization", *IEEE Computer Graphics and Applications*, 9(4), July 1989, pp. 30-42.

NASA-CR-194879

NASA
National Aeronautics and Space Administration

Report Documentation Page

IN-61-CR

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No. 203556

4. Title and Subtitle

A Knowledge Based System for Scientific Data Visualization

5. Report Date

January 1992 23P

6. Performing Organization Code

7. Author(s)

Hikmet Senay
Eve Ignatius

8. Performing Organization Report No.

10. Work Unit No.

506-59-11

9. Performing Organization Name and Address

Department of Electrical Engineering & Computer Science
The George Washington University
801 22nd Street, T-624A, NW
Washington, DC 20052

11. Contract or Grant No.

NAS5-30428

13. Type of Report and Period Covered

Final

12. Sponsoring Agency Name and Address

CESDIS (Center of Excellence in Space Data & Information
Sciences
Code 930.5
Goddard Space Flight Center/Greenbelt, MD 20771

14. Sponsoring Agency Code

15. Supplementary Notes

CESDIS (the Center of Excellence in Space Data and Information Sciences) is operated by Universities Space Research Association (USRA), American City Building, Suite 212, 10227 Wincopin Circle, Columbia, MD 21044.

16. Abstract

This paper describes a knowledge-based system, called visualization tool assistant (VISTA), which was developed to assist scientists in the design of scientific data visualization techniques. The system derives its knowledge from several sources which provide information about data characteristics, visualization primitives, and effective visual perception. The design methodology employed by the system is based on a sequence of transformations which decomposes a data set into a set of data partitions, maps this set of partitions to visualization primitives, and combines these primitives into a composite visualization technique design. Although the primary function of the system is to generate an effective visualization technique design for a given data set by using principles of visual perception, the system also allows users to interactively modify the design, and renders the resulting image using a variety of rendering algorithms. The current version of the system primarily supports visualization techniques having applicability in earth and space sciences, although it may easily be extended to include other techniques useful in other disciplines such as computational fluid dynamics, finite-element analysis and medical imaging.

17. Key Words (Suggested by Author(s)) scientific data visualization; visualization mapping; visual perception; data decomposition; image composition; visualization tree	18. Distribution Statement Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages 25	22. Price