A visualization model based on the mathematics of fiber bundles

D. M. Butler and M. H. Pendley

Citation: Computers in Physics **3**, 45 (1989); doi: 10.1063/1.168345
View online: https://doi.org/10.1063/1.168345
View Table of Contents: https://aip.scitation.org/toc/cip/3/5
Published by the American Institute of Physics

A visualization model based on the mathematics of fiber bundles

D. M. Butler
Limit Point Systems, 39120 Argonaut Way, Suite 397, Fremont, California 94538
M. H. Pendley
Sandia National Laboratories, Livermore, California 94551-0969

(Received 3 October 1988; accepted 29 December 1988)

In this paper, a visualization model based on the mathematics of fiber bundles is described. A brief, intuitive description of the mathematics of fiber bundles is given, introducing the concepts using typical application examples and emphasizing aspects relevant to our visualization model. Three important classes of operations on fiber bundles are described. A flexible scheme is developed for constructing graphic representations of fiber bundles and a simple but useful visualization taxonomy.

INTRODUCTION

In this article, we describe a specific visualization model based on the mathematics of fiber bundles. In a companion article,[1] we introduced the visualization management system (ViMS), a new approach to the development of software for visualization in scientific computing (ViSC). A visualization management system is an abstraction of data structures and functions common to many ViSC applications. It is a generalized, application-independent facility for the definition, analysis, and presentation of visual representations of scientific data. A ViMS implements an abstract visualization model which specifies a class of geometric objects, the graphic representations of the objects, and the operations on both. In the model described here, the geometric objects are sections of fiber bundles. The fiber bundle formalism provides a unified, dimension-independent framework that encompasses both simple and complex visualization problems, forming the basis for a powerful and widely applicable visualization management system.

The article is organized as follows. In Sec. I, we discuss the general notion of a visualization model and review the requirements it must satisfy. In Sec. II, we introduce and motivate the use of fiber bundles as the geometric objects of our model. We give a brief, intuitive description of the mathematics of fiber bundles, emphasizing aspects relevant to our visualization model. In Sec. III, we describe the graphic representations and operations of our model. In Sec. IV, we evaluate our model with respect to the requirements described in Sec. I. In Sec. V, we develop a simple but useful taxonomy for fiber bundles and their graphic representations.

I. VISUALIZATION MODEL REQUIREMENTS

As we described in the Introduction, a visualization model specifies a class of geometric objects, the graphic representations of the objects, and the operations on both. The requirements a visualization model must satisfy were developed in the companion article; we review them here.

(1) Application independence: the visualization model must provide a class of geometric objects suitable for a wide range of applications.

(2) Integrated visualization and computation: the geometric objects must provide useful and widely applicable computational operations.

(3) Flexible geometric representation: the geometric objects must support multiple dimensions and complex topologies.

(4) Flexible graphic representation: the model must provide a flexible and complete set of graphic representations.

(5) Data representation independence: the geometric objects defined must provide general, multilevel access to the geometric information.

II. GEOMETRIC OBJECTS AND OPERATIONS

A. Motivation for fiber bundles

Although the requirements described in Sec. I are formidable, the following considerations suggest the mathematical theory of fiber bundles provides a suitable class of geometric objects. The requirement for multiple dimensions and complicated topologies, coupled with the requirement for wide applicability, strongly suggests we use the formalism of differential geometry. Differential geometry explicitly encompasses multiple dimensions and complicated topologies and has recently gained increasing recognition as a powerful and precise mathematical formalism applicable to a wide range of problems in the physical sciences. Differential geometry provides a number of candidates for the geometrical objects of our model. However, the requirement for integrated visualization and computation, for widely applicable computational operations, singles out structures called fiber bundles. This is because the majority of computations require calculus and the differential geometrical generalization of calculus takes place on fiber bundles. Thus, fiber bundles are the natural geometrical objects for a visualization model that supports calculus and related computations in arbitrary dimensions and topologies.

Motivated by the preceding observations, we developed a visualization model based on the mathematics of fiber bundles. In this paper, we give a very informal and intuitive description of the mathematics of fiber bundles. For a more complete treatment the interested reader should see the discussion in Burke,[2] in Nash and Sen,[3] or in Abraham *et al.*[4]

B. Structure of fiber bundles

Roughly speaking, a fiber bundle is a space that is constructed from two spaces, a base space and a fiber space, by attaching a copy of the fiber space to each point of the base space. The bundle is the Cartesian product of the base and the fiber. In Fig. 1, we show a simple example of a fiber bundle. If we pick a particular point on each fiber of the bundle, we define a (cross) section of the bundle, also shown in Fig. 1.

In the example, the base and fiber are both the real number line, a one-dimensional space with a familiar topology. However, in general, the fiber and base can both be arbitrary, multidimensional spaces. Figure 2 shows a more complicated example. The base space is a torus, while the fiber is a three-dimensional vector space. A section of this bundle is a vector field on the torus. This example is typical of scientific applications: the fiber is a multidimensional vector space and the base is a multidimensional space with a nontrivial topology. For the simple example given in Fig. 1, familiar geometric ideas such as ordinate, abscissa, and curve are adequate, and the machinery of fiber bundles is unnecessary. However, in the more complicated applications, the familiar ideas become inadequate. The fiber bundle formalism provides a unified, dimension-independent framework that encompasses both simple and complex visualization problems.

To connect the fiber bundle formalism with a more familiar formalism, a fiber bundle may be viewed as a generalization of a function of one or more variables. The base space is analogous to the independent variables of a function; the fiber space is analogous to the dependent variable of a function; a section corresponds to the function itself; and the fiber bundle is the space in which we graph the function. To reinforce this analogy and help the reader become familiar with the fiber bundle interpretation, we return to the example of Fig. 2. The field is given by

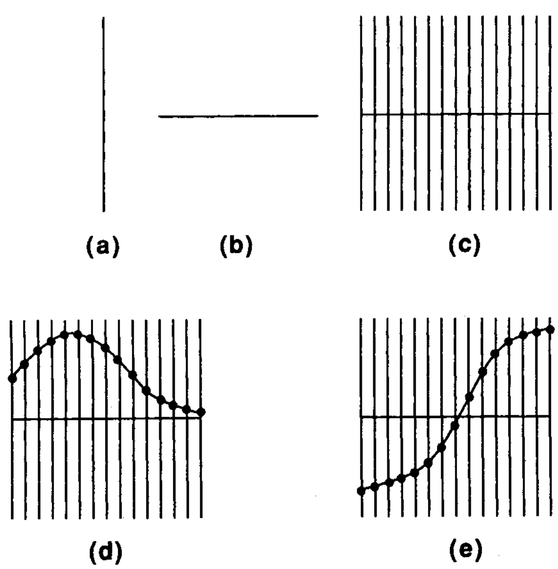

FIG. 1. A trivial fiber bundle. Attaching a copy of the fiber (a) to each point of the base (b) gives the bundle (c). Picking a point in each fiber space defines a section (d). Another section of the same bundle (e).

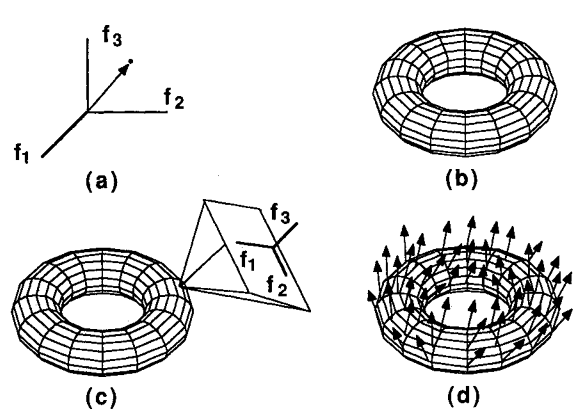

FIG. 2. A more complicated trivial fiber bundle. The fiber (a) is a three-dimensional vector space; a point in this space is represented by an arrow. The base (b) is a torus. The bundle (c) is formed by attaching a copy of the fiber to each point of the base. Choosing a point, i.e., a vector, in each fiber defines a section (d).

$$\mathbf{V} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} f_1(\theta_1,\theta_2) \\ f_2(\theta_1,\theta_2) \\ f_3(\theta_1,\theta_2) \end{bmatrix},$$

where $0 \leqslant \theta_1, \theta_2 < 2\pi$, and f_1, f_2, and f_3 are real-valued functions.

To interpret this function as a fiber bundle, we emphasize not the individual variables $(\theta_1,\theta_2,v_1,v_2,v_3)$ but the spaces in which they take their values. The independent variables θ_1 and θ_2 are coordinates on the surface of a torus. Thus the base space is the surface of the torus T^2. The dependent variables v_1, v_2, v_3 are coordinates of the vector space R^3. Hence the fiber is the vector space R^3. The fiber bundle is the product space $T^2 \times R^3$. The section is defined by the values of the three functions (f_1,f_2,f_3) at each point of the base. Note that by emphasizing the space rather than the variables, the toroidal topology of the base is explicitly incorporated in the interpretation.

In this manner, a scalar, vector, or tensor field can be interpreted as a fiber bundle. The base is the space for which the independent variables are coordinates. The fiber is a vector space: R^1 for scalar fields, R^2 or R^3 for vector fields, and R^n $(n>3)$ for tensor fields.

Technically, the bundles we have described are called trivial bundles; they are locally and globally the Cartesian product of the base and the fiber. Nontrivial bundles, which we will not discuss, contain an additional structure that describes the way in which the bundle is "twisted" so that it is not a product globally. The analytical power of the theory of fiber bundles is mostly due to the nontrivial bundles, and they provide a very general basis for a visualization model as well. However, the full theory substantially complicates the model and much progress can be made

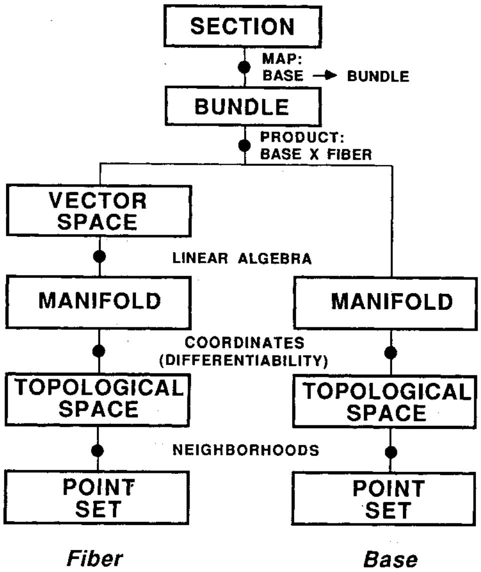

FIG. 3. Layers in the geometry of a vector bundle. The layers are indicated by boxes. The lines joining the layers are labeled with the structure added to obtain each layer from the one below it. The simple product structure of the bundles results from our restriction to trivial bundles.

with only the trivial bundles. Indeed, as suggested by the examples above, we are most interested in vector bundles, that is, fiber bundles in which the fiber is a vector space. A nontrivial vector bundle can always be mapped to a trivial vector bundle having the same base space but a fiber space of higher dimension than the original. Thus we can support nontrivial vector bundles indirectly by supporting trivial vector bundles and mappings. To simplify both the presentation and implementation of the visualization model, we restrict the model to trivial vector bundles for the remainder of this discussion.

The geometry of a section of a vector bundle is built up from set theory by repeatedly aggregating simpler objects, creating a layered structure. We describe the layered structure here; in Sec. IV, we discuss its importance for the visualization model. We need to discuss only the general features of the structure, as shown in Fig. 3. At the lowest level, both the base and fiber are point sets, just collections of points. For the next layer, we add the notion of neighborhoods to obtain a topological space. Then we add the notion of coordinates and differentiability to get a manifold. The fiber is a vector space, which can be considered a manifold with an additional layer of structure, its linear algebra structure. The next layer aggregates the base and fiber into a bundle. Finally, the bundle is aggregated with a map, specifying a value in each fiber, to give the section. Thus the section is a multilayer structure and each layer contains a specific type of abstract geometric data.

C. Visualization model

Having given a brief description of the structure of fiber bundles, we can state the visualization model more precisely. Sections of trivial vector bundles are the geometric objects of the model. A geometric representation of an application data structure is a mapping between the data structure and one or more sections of a vector bundle. We will refer to the number of sections as the multiplicity of a representation. A typical application will have several data structures associated with it and will have one or more geometric representations for each data structure. The collection of representations associated with an application typically consists of several bundles and several to many sections on each bundle.

D. Operations

Vector bundles support a number of useful operations. To give the reader an idea of the abilities that can be implemented in the context of the model, we outline some of the important classes of operations and briefly show how each might be applied to the example of Fig. 2. The fundamental categories are constructors, mappings, and section operations.

Constructors allow us to construct geometric objects, for instance, a base or a section. In an implementation of the visualization model, the constructors are responsible for allocating and initializing the program variables which store the geometric objects. In general, values for these variables can be defined explicitly or by importing some application data. In the discussion below, we assume some imported application data corresponding to the example. The most important classes of constructors are the following.

(1) Base constructors: operations that specify the structure of the base space, especially definition of the point set and topology or assignment to an application data structure. In the example, base constructors would be used to allocate a variable b of type base and store a torus in it. The grid of values for the coordinates θ_1 and θ_2 could be defined directly or by using some application data.

(2) Fiber constructors: operations that specify the structure of the fiber by definition or by assignment to an application data structure. In the example, fiber constructors would be used to define the fiber f, setting its dimension to three and the data type of its components v_1, v_2, nd v_3 to match the application data, perhaps REAL or DOUBLE PRECISION.

(3) Bundle constructors: operations that assign a base or a fiber to a bundle. To construct the bundle for the example, we would allocate a variable B of type bundle, with base b and fiber f.

(4) Section constructors: operations that specify a section of a bundle, especially assignment to an application data structure. In the example, we could allocate a variable S of type section and initialize its components with the values of the application data.

Mappings are used to create new objects from existing objects. A bundle mapping is a mapping from one bundle to another that preserves the fibers; it maps base to base and fiber to fiber. We describe two major classes, bundle restriction and subbundle, and some useful subclasses.

(1) Bundle restriction: a bundle mapping in which the base of one bundle is a subspace of the base of the other but the fibers are the same. In other words, we form a new bundle by extracting some subspace of the base and the fibers just get dragged along. Or, by interpreting the mapping in the opposite direction, we can immerse the base as a

subspace in a higher dimensional base, dragging along the fibers.

(2) Subbundle: a bundle mapping in which the fiber of one bundle is a subspace of the fiber of the other but the bases are the same. In other words, we form a new bundle by just discarding some of the fiber. Again, we can interpret the mapping in the opposite direction, adding dimensions to the fiber.

(3) Boolean operations: bundle restrictions corresponding to the boolean operations of constructive solid geometry.[5] The boolean operations are essentially the set operations union, intersection, and difference applied to the point sets representing objects. They are widely used in computer-aided design applications for composing and decomposing three-dimensional objects. Extending the definition to fiber bundles, boolean operations on the base induce restrictions on the bundle, allowing the user to compose or decompose the base while carrying along the fibers. In the example, boolean operations could be used to cut the torus in two, forming two half-donuts with a vector field on each half.

(4) Slicing: a bundle restriction induced by decomposing the base into a family of subspaces, producing a family of bundles of lower dimension. Slicing converts a representation of given dimensionality and multiplicity into a representation of lower dimensionality and higher multiplicity. In our example, we can slice the torus into a collection of circular cross sections, each a bundle itself. The fiber is still the three-dimensional vector space R^3, but the base is now the circle S^1, a one-dimensional space.

(5) Stacking: the inverse of slicing, composes a family of lower dimension bundles into a higher dimension bundle. Stacking converts a representation of given dimensionality and multiplicity into a representation of higher dimensionality and lower multiplicity. In our example, we can stack the collection of circular cross sections back into a torus.

(6) Component projection: a subbundle created by ignoring all the components of the fiber except one. We can use component projection repeatedly, once for each component, to reduce a representation of fiber dimension d_f and multiplicity one to a representation of fiber dimension one and multiplicity d_f. In our example, we can project the three components of the vector field. The resulting subbundle still has the torus as a base, but the fiber is now a one-dimensional vector space R^1, and we have three sections defined on it.

(7) Component direct product: a bundle formed by taking the direct product of the fibers of two bundles defined over the same base. We can use the direct product to increase the dimensionality and reduce the multiplicity of a representation, forming the inverse of component projection. In our example, starting with the subbundle we produced with component projecton, we can reassemble the three one-dimensional sections into a three-dimensional vector using the component direct product.

The mappings we have described are not an exhaustive classification. However, they do provide the basis for manipulating the dimensionality of a representation. We will use this ability in the visualization taxonomy we develop in Sec. V.

Section operations provide integrated visualization and computation. These operations can be defined entirely in terms of the geometric objects and hence are application independent. Some of the useful classes of operations for vector bundles are the following.

(1) Approximation: interpolating or smoothing a section. For instance, we could fit multidimensional splines to the vector field on the torus.

(2) Linear algebra: multiplying a section by a constant, adding or subtracting two sections defined on the same bundle. For instance, if we had two vector fields on the torus, perhaps corresponding to two different experimental measurements, we could calculate the difference or the average of the two fields.

(3) Calculus: integrating and differentiating a section. In the example, we could calculate the divergence or curl of the vector field.

III. GRAPHIC REPRESENTATIONS AND OPERATIONS

A graphic representation in our visualization model is a mapping between one or several sections of a vector bundle and a visual display. In this section we describe the general structure of the graphic representation, assuming the facilities of a modern computer graphics environment: a representation space in which graphic objects can be defined, a structured database for storing and retrieving objects, and operations for manipulating the graphic attributes of the objects.

Our representation scheme provides two levels of graphic composition, the graph level and the visualization level. The graph level provides graphic representations of individual geometric structures. The visualization level integrates these individual graphic objects into a complete visualization.

The graph level provides for the creation and manipulation of graphic objects representing individual geometric structures. We refer to a graphic object associated with a geometric structure as a "geometric structure graph," or just "graph" for short. These graphs are constructed by mapping the geometric structure to the graphic representation space. Since our visualization model supports geometric structures of arbitrary dimension, not all structures can be mapped to the representation space in their entirety. We discuss this problem in Sec. V; in this section we limit the discussion to those geometric structures which can be represented directly.

Two categories of representations can be constructed, explicit representations and parametric representations. Parametric representations can only be constructed for multiple sections of bundles with the same base. For simplicity, we describe only the explicit representation.

In the explicit representation, a section has four graphs associated with it: the fiber graph, the base graph, the bundle graph, and the section graph itself. Figure 4 shows typical graphs for the example in Fig. 1. The fiber graph, Fig. 4(a), depicts the geometric structure of the fiber space by indicating a coordinate axis for the space. The base graph, Fig. 4(b), does the same for the base space. The bundle graph, Fig. 4(c), combines the base and fiber graphs to indicate the strucure of the bundle space. The section graph, Figs. 4(d) and 4(e), depicts the actual data. A fifth type of graph, the text graph, is an independent

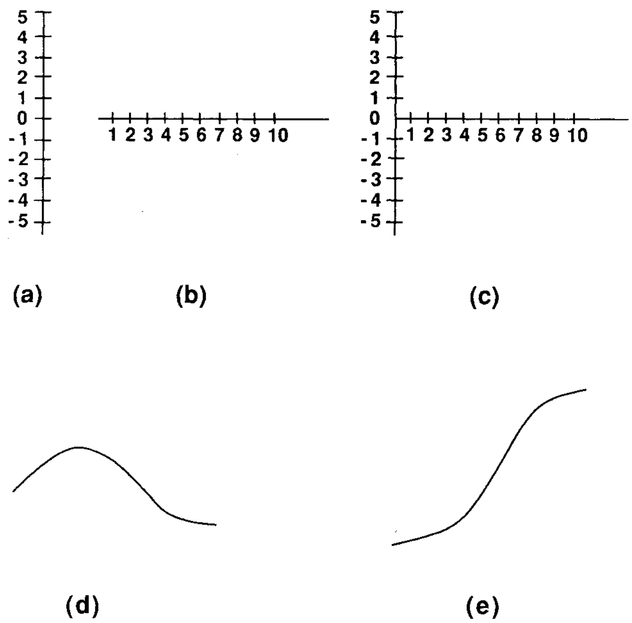

FIG. 4. Graphic representations at the graph level. The fiber graph (a), the base graph (b), the bundle graph (c), the section graph (d), and another section graph (e) are graphic representations of the corresponding geometric objects of Fig. 1.

component which provides arbitrary annotation.

The visualization level provides for the integration of the graphs into a complete visualization. A visualization is an arbitrary collection of graphs. Graphs can be flexibly combined in any way; Fig. 5 shows two visualizations constructed from the graphs of Fig. 4.

IV. EVALUATION OF MODEL

While the efficacy of the vector bundle visualization model can ultimately be determined only from experience with a ViMS based on it, we can review its properties with regard to the requirements we have defined.

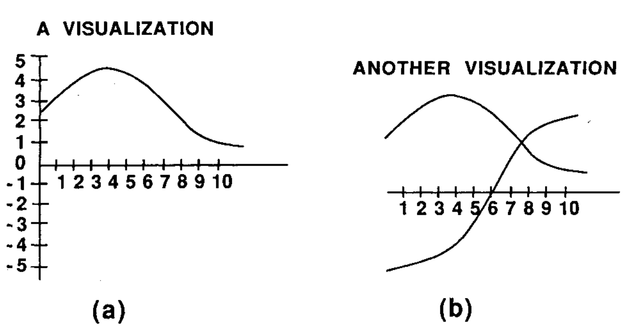

FIG. 5. Graphic representations at the visualization level. A visualization (a) is an arbitrary collection of graphs. Another visualization (b) contains two section graphs but no fiber graph. Both contain a text graph.

A. Application independence

Vector bundles, the geometric objects provided by the model, are suitable for a very wide range of applications. As the examples given above indicate, scalar, vector, and tensor fields on arbitrary spaces can be interpreted as vector bundles. Two additional observations further indicate the generality of the model.

The first observation is the wide and growing acceptance of differential geometry as a mathematical formalism for problems in the physical sciences. Point mechanics, continuum mechanics, fluid dynamics, thermodynamics, statistical mechanics, classical field theory, and quantum field theory all have well-established differential geometric formalisms in which vector bundles appear explicitly as important objects of study. Thus it is likely that a visualization model based on vector bundles will be useful in all these application areas.

A second observation is that the central notion provided by the definition of a section of a fiber bundle is the notion of relationship between points in the base and points in the fiber. In fact, a section of a fiber bundle can be considered a geometrization of a relation, in the sense of set theory, between the point sets of the base and the fiber. By imposing geometry on otherwise unstructured sets, the model can be used to visualize arbitrary relations between arbitrary sets. This interpretation essentially formalizes and generalizes common practice. For instance, in constructing a bar chart of grant revenue versus faculty member, geometry must be imposed on the set of faculty members.

B. Integrated visualization and computation

The model integrates visualization with calculus in arbitrary dimension and topology. Hence, it provides an integrated environment applicable to most of scientific computing.

C. Flexible geometric representation

The model provides geometric representations with arbitrary dimension and topology. In particular, the model allows geometric interpretation of high-dimensional, nonvisualizable structures and provides operations for mapping such structures to lower dimensional, visualizable ones.

D. Flexible graphic representation

The model provides flexible graphic representation based on two levels of graphic composition. At the lower level, the model allows multiple graphic representations of each of the four graph types: base, fiber, bundle, and section. At the upper level, it provides arbitrary combination of these objects into a complete visualization.

E. Data representation independence

A section of a vector bundle has associated with it the layers shown in Fig. 3: point set, topological space, manifold,

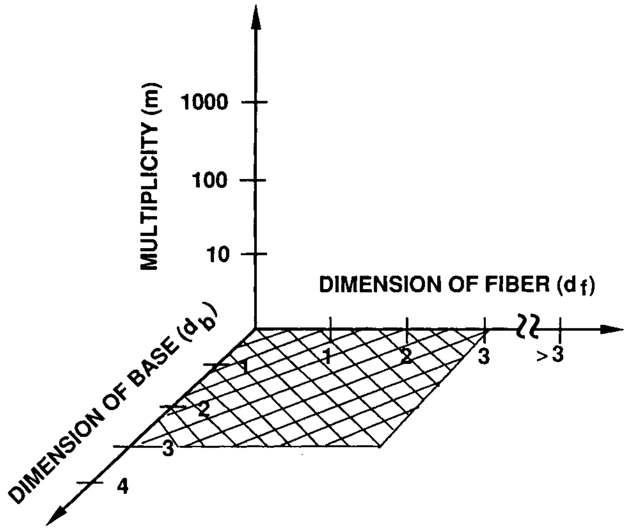

FIG. 6. The space of geometric representations. In this space, a geometric representation of some application data is a point, specified by the triple (d_b, d_f, m). The directly visualizable region is crosshatched.

vector space, bundle, and section. Each layer provides an interface for the exchange of the geometric data associated with that layer. Together they provide general and complete access to the geometry, independent of any particular data structure chosen to store the geometry.

VI. TAXONOMY

We conclude by describing a simple visualization taxonomy based on the vector bundle model. While a complete taxonomy for this model is both beyond the scope of this paper and an open research question of some complexity, the taxonomy we present covers many practical cases and indicates the value of a visualization taxonomy.

We can parametrize the set of geometric representations by the dimension of the base d_b, the dimension of the fiber d_f, and the multiplicity m. We can then visualize the three-dimensional "space" of geometric representations as shown in Fig. 6, which shows the region spanning most scientific applications. Most applications can be represented as bundles having a base of dimension four or less, because the base is typically space, time, or space-time, with space having a dimension between one and three. The dimension of the fiber can be grouped into four categories: dimension one, dimension two, dimension three, and dimension greater than three. The categories arise from scalar, vector (dimension two and three), and tensor fields, respectively. The multiplicity varies widely, as suggested by the logarithmic scale in the figure. Large multiplicities can be intrinsic to the application or can be generated in the course of visualization.

Having classified the geometric representations, the next task is to develop graphic representations for each class. First we consider the geometric representations with multiplicity greater than one. Graphic representation of an object $(d_b, d_f, m > 1)$ can always be constructed using a representation of $(d_b, d_f, m = 1)$ and either serialization or superposition. Serialization presents the graphic representations in a static or dynamic series, while superposition presents them at once, superimposed on each other. With these techniques, we need to develop graphic representations for only the $m = 1$ plane of the geometric representation space.

A graphic representation maps the geometric object to the graphic representation space. For simplicity, we assume the graphic representation space has three spatial dimensions and supports one local attribute,[6] intensity or color. Most color models are three dimensional, but the effectiveness of color for representing multidimensional quantities is limited,[7] so we will consider the dimension of the local attribute to be (at least) one.

There is no natural, direct way to represent bundles with base or fiber dimension greater than three, since there is no structure in the representation space of dimension greater than three. Thus there are nine geometric representations for which we can construct direct graphic representations. Table I presents canonical forms for the graphic representations of these geometric representations.

Bundles which cannot be visualized directly can be reduced to collections of visualizable ones using bundle mappings. In terms of Fig. 6, bundle mappings move the point corresponding to the representation into the directly visualizable region by decreasing the dimensionality of the base or fiber and increasing the multiplicity. Serialization or superposition must then be used to present the graphic representations. For instance, base dimension four, which usually corresponds to one time dimension and three space

TABLE I. Canonical graphic representations of the $m = 1$ geometric representations.

d_b, d_f	Canonical representation
1,1	(a) line plot (b) color scale
1,2	(a) 1-D distribution of 2-D arrows (b) trajectory
1,3	(a) 1-D distribution of 3-D arrows (b) trajectory
2,1	(a) pseudocolor image (b) contour plot (c) surface plot
2,2	(a) 2-D distribution of 2-D arrows
2,3	(a) 2-D distribution of 3-D arrows
3,1	(a) pseudocolor volume image
3,2	(a) 3-D distribution of 2-D arrows
3,3	(a) 3-D distribution of 3-D arrows

dimensions, is naturally represented by serializing on time. Subbundle mappings, component projection in particular, can reduce representations with fiber dimension greater than three to visualizable representations.

To summarize the taxonomy, the geometric representations are classified by the base dimension, the fiber dimension, and the multiplicity. Canonical graphic representations are provided for the geometric representations with base dimension three or less, fiber dimension three or less, and multiplicity one. The remaining bundles must be reduced to collections of visualizable bundles using bundle mappings. The collection of reduced bundles can then be visualized using superposition or serialization of their canonical graphic representations.

The visualization taxonomy benefits both ViMS developers and ViMS users. The ViMS developers benefit from a taxonomy because it provides a framework for identifying, analyzing, and implementing necessary visualization functionality, in particular the canonical graphic representations. The ViMS users benefit because an arbitrary visualization problem can be formulated and its solution communicated using this taxonomy.

The taxonomy we have described can be extended in a number of ways. A more complex graphic representation space, supporting attributes such as reflectivity, transparency, and texture, can be used. Direct graphic representations of geometric representations with multiplicity greater than one can be developed. A more extensive and detailed set of canonical graphic representations can be provided. The specific bundle mappings needed to reduce various classes of nonvisualizable bundles can be identified. Extending the taxonomy would enhance its utility to both developers and users. We plan to extend the taxonomy in the future.

ACKNOWLEDGMENTS

We would like to thank Marc Lowe for several helpful discussions concerning the graphic representations and for programming several examples.

This work was supported by the United States Department of Energy under Contract No. DE-ACO4-76DP00789.

REFERENCES

1. D. M. Butler and M. H. Pendley, Comput. Phys. **3**(5), 40 (1989).
2. W. L. Burke, *Applied Differential Geometry* (Cambridge U. P., Cambridge, England, 1985).
3. C. Nash and S. Sen, *Topology and Geometry for Physicists* (Academic, Orlando, 1983).
4. R. Abraham, J. E. Marsden, and T. Ratiu, *Manifolds, Tensor Analysis, and Applications* (Addison–Wesley, Reading, MA, 1983).
5. M. E. Mortenson, *Geometric Modeling* (Wiley, New York, 1985).
6. With this assumed structure, the graphic representation space can be interpreted as a fiber bundle itself. The three-dimensional space is the base and the local attribute is the fiber. However, the mapping from the geometrical representation to the graphic representation is, in general, not a bundle mapping.
7. J. Bertin, *Semiology of Graphics* (Univ. of Wisconsin Press, Madison, WI, 1983).