STUDIA UNIV. BABEŞ-BOLYAI, INFORMATICA, VOLUME XLIII, NUMBER 2, 1998

ON SCIENTIFIC VISUALIZATION SYSTEMS DESIGN

DUMITRU RADOIU

Abstract. A "decomposition" of the scientific visualization process allows more accurate definitions of the conditions on the visualization function. This is realized through the following steps: defining a model for the aggregate objects, defining operations between objects, defining a structure on the object set, finding the conditions on which similar models and structures could be defined over all intermediary objects (sets), including the output visual objects and defining the analytic conditions for which the visualization function preservers the structure while fulfilling the fundamental visualization requirements. Usually the analytic conditions describe a sort of isomorphism.

A way to define operations between experimental data objects usually defined with the help of predicates is also given. Entropy and average value are used to order data objects interpreting the relative order as a precision order. The most general conditions of the scientific visualization are reformulated and given a formal description. The visualization process could be called "scientific visualization" if the classes of functions - expressing different visualization goals – that satisfy the mentioned conditions are identified and used.

1. Introduction

Scientific Visualization is a computational process that transforms scientific data in visual objects [8]. In this paper a "decomposition" of the scientific visualization is introduced (Table1). The purpose is to clear identifying the objects involved in the process and more accurate formulating the conditions on the visualization functions. The idea of using the mathematical structures defined over data sets to find conditions imposed on the visualization function has been promoted by others [1],[2],[9].

Scientific data could be obtained in many different ways, e.g. by running a simulation or through a DAQ process. Usually, scientific data objects are finite representations of complex mathematical objects. We note by **O** the set of such objects, $o \in \mathbf{O}$. During the visualization process, initial data objects, o, are processed through different transformation functions $Mat(o) = o'$, into a new set $o' \in \mathbf{O'}$. Objects o' are then mapped $Map(o') = g$ into a set of ideal geometrical objects $g \in \mathbf{G}$, through a set of graphical primitives. Objects g are usually n-dimensional (nD), animated (t) and interactive. A group of g objects is usually called **logical visualization** of a scene. Ideal geometrical objects g, nD, animated (t) and interactive are usually represented $Rep(g) = g'$, $g' \in \mathbf{G'}$, on real 2D screens. A group of g' objects is usually called a **physical visualization** of a scene. Functions $Rep(g) = g'$ implement classical graphical operations like composition of the scene, volume generation, isosurface generation, simulation of transparency, reflectivity and lighting conditions, nD $\rightarrow$ 2D projection, clipping, hidden surface removal, shading, animation (t), setting user interactivity (zoom, rotate, translate, pan, etc), etc.

By **interactivity** we understand the attributes of visual objects (logical and/or physical) whose setting permits nD → 2D projection (zoom, rotate, translate, pan, etc), animation control (t), control of the objects composing the scene and control of the scene as a composite object.

By scientific visualization we understand the process realized by the function $Viz(o) = g'$

$Viz(o) = Rep(Map(Mat(o))) = g'$

The general problems of scientific visualization modeling are:

- defining acceptable models for objects of O, O', G and G' sets
- defining mathematical structures on O, O', G and G' sets
- finding the general conditions for $Viz(o)$

Yet, we are aware that a specific scientific visualization problem must address other issues as well, like:

- particularities of a specific algorithm
- design of the visualization pipeline
- comparative and/or syncron visualization, etc

2. Fundamental conditions of scientific visualization

There are many requirements relative to a certain process of scientific visualization. Here are the two fundamental ones we will consider in this paper. The first one is the condition of distinctiveness. This condition (although very week) enables users to distinguish between different data objects based on their display. The condition is necessary for one can imagine many visualization functions that generate images with no use, that reveal none of the data objects characteristics/attributes.

a. Condition of distinctiveness

Different data objects must be mapped into different visual objects.

This is could be stated as:

$o_1 = o_2 \Leftrightarrow Viz(o_1) = Viz(o_2) \Leftrightarrow Rep(Map(Mat(o_1))) = Rep(Map(Mat(o_2))) \Leftrightarrow g_1' = g_2'$

$o_1, o_2 \in O,\ g_1', g_2' \in G'$

The interpretation of this condition is that $Viz()$, $Mat()$, $Map()$ and $Rep()$ functions define an isomorphism between equality in O,O', G and G'.

b. Condition of expressiveness

Visual objects express all facts about data objects and only those facts.

Facts are interpreted as attributes and attributes could be described by monotone predicates, the same attribute being described by different predicates defined on each set so that:

$$\forall P_O : O \rightarrow \{\text{undefined, true}\},\ \exists P_{O'} : O' \rightarrow \{\text{undefined, true}\},$$

$$\forall P_{O'} : O' \rightarrow \{\text{undefined, true}\},\ \exists P_G : G \rightarrow \{\text{undefined, true}\},$$

$$\forall P_G : G \rightarrow \{\text{undefined, true}\},\ \exists P_{G'} : G' \rightarrow \{\text{undefined, true}\},$$

so that

$$\forall P_O : O \rightarrow \{\text{undefined, true}\} \Rightarrow$$

$$P_O(o) = P_{O'}(Mat(o)) = P_G(Map(Mat(o))) =$$

$$= P_{G'}(Rep(Map(Mat(o)))) = P_{G'}(Viz(o)), \forall o \in O$$

and

$$\forall P_{G'} : G' \rightarrow \{\text{undefined, true}\},$$

$$P_{G'}(g') = P_G(Rep^{-1}(g')) = P_{O'}(Map^{-1}(Rep^{-1}(g')) =$$

$$= P_O(Mat^{-1}(Map^{-1}(Rep^{-1}(g'))) = (P_O(Viz^{-1}(g')$$

3. Data objects: models, basic operations and order relation

An object model should address:

a. Data types (primitive variable values)
b. How primitive variables are aggregated in complex data/visual objects
c. Basic operations between data/visual objects, a order relation, a topology and a metric
d. Metadata (extra information)

A data objects (o ∈ **O**) is a collection of primitive variable values. As primitive variable values could be described with classical data types (e.g. real, float, array, etc), when considering the data type of an aggregate data object, the most direct approach is to extend the data type of the primitive variable values to the aggregated data object. The next step is to extend the mathematical structures defined over sets of primitive values to sets of aggregate data objects.
We will consider without demonstration that we can extend all aspects discussed above to objects o' ∈ **O'** and also g ∈ **G** and g' ∈ **G'**.

After defining legal operations between aggregate data objects, the most important aspect to be addressed is the relative order of objects in a set. An order relation could be interpreted as a precision relation. A more precise data object will generate (through the visualization pipeline and under certain conditions of function *Viz(o)*) a more precise visual object.
In the following example we will consider a certain data object, define some fundamental operations, and a precision relation.

We consider the case where aggregate data objects are structured grids with scalar values in each node (one type - O_i - of aggregate object data among many, $O_i \subset$ **O**). If measured values are generated by a scientific experiment, it is legal to assume that some of the values are missing. We agree to describe all facts about data objects through monotone predicates $P : O_i \rightarrow \{\text{undefined, true}\}$. For an object $o_n \in O_i$, we may consider $P(o_{np})$ = true if there is a value in the p-th node of the object data o_n, and $P(o_{np})$ = undefined if there is no measured value in the p-th node of the object data o_n.

Between objects of the same type, elements of a subset O_i, ($O_i \subset \mathbf{O}$) operations could be defined. When one or more attributes of the data object are missing (attribute p of the data object o_n is labeled o_{np}), operations could be defined through a predicate [5],[6],[7] $P : o_n \rightarrow \{\text{undefined, true}\}$. When the attribute is present then $P(o_{np}) \rightarrow true$

And when the attribute is missing $P(o_{np}) \rightarrow undefined$

a) SUM

If for $\forall o_n, o_m \in O_i$ there is a predicate

$P : o_n \rightarrow \{\text{undefined, true}\}$ and

$P : o_m \rightarrow \{\text{undefined, true}\}$.

$$SUM(o_n, o_m, o) = (o = o_n + o_m) = \begin{cases} SUM(o_{np}, o_{mp}), P(o_{mp}) = true \,\&\, P(o_{np}) = true; \\ o_{mp}, P(o_{mp}) = true \,\&\, P(o_{np}) = undefined; \\ o_{np}, P(o_{mp}) = undefined \,\&\, P(o_{np}) = true.; \\ false, P(o_{mp}) = undefined \,\&\, P(o_{np}) = undefined; \end{cases}$$

b) DIF

If for $\forall o_n, o_m \in O_i$ there is a predicate

$P : o_n \rightarrow \{\text{undefined, true}\}$ and

$P : o_m \rightarrow \{\text{undefined, true}\}$.

$$DIF(o_n, o_m, o) = (o = o_n - o_m) = \begin{cases} DIF(o_{np}, o_{mp}), P(o_{mp}) = true \,\&\, P(o_{np}) = true; \\ o_{mp}, P(o_{mp}) = true \,\&\, P(o_{np}) = undefined; \\ -o_{np}, P(o_{mp}) = fundefined \,\&\, P(o_{np}) = true; \\ false, P(o_{mp}) = undefined \,\&\, P(o_{np}) = undefined; \end{cases}$$

c) Scalar multiplication

If for $\forall o_n \in O_i, \forall \lambda \in \Re$ there is a $P : o_n \rightarrow \{\text{undefined, true}\}$.

$$(\lambda \cdot o_n) = \begin{cases} \lambda \cdot o_{np}, P(o_{np})=true; \\ undefined, P(o_{np})=undefined; \end{cases}$$

d) Scalar division

If for $\forall o_n \in O_i$, $\forall \lambda \in \Re$ there is a $P : o_n \rightarrow$ {undefined, true}.

$$(\cdot o_n / \lambda) = \begin{cases} o_n / \lambda,, P(o_{np})=true; \\ undefined, P(o_{np})=undefined; \end{cases}$$

e) Cartesian distance between two data objects

For any $\forall o_n, o_m \in O_i$ the distance is defined as

$$\mathrm{DIST}(o_n, o_m, d_{mn}) \equiv d_{mn} \equiv |o_m - o_n| = \left[(o_{m1} - o_{n1})^2 + (o_{m2} - o_{n2})^2 + ..\right]^{1/2}$$

f) Average value

Let us consider $o_1, o_2, \ldots, o_n \in O_i$ By definition <o> =AVERAGE ($o_1, o_2, \ldots, o_n$). and<o> $\in O_i$ is the average (arithmetic, harmonic, etc). between attributes having the same position within the data objects. When the attribute is missing in that location we use the predicate $P : o_{np} \rightarrow$ {nedefined,true} and the attribute is part of the average value only if $P : o_{np} \rightarrow$ true.

Logical operations could be introduced (and most of the time make sense) for about any type of data objects: intersection, reunion, and the like.
Other operations could be also defined considering a certain data type (e.g. images, regular grids, etc) for which the operations make sense.
As we stated in the previous paragraph, after defining operations between data objects, the most important issue is the relative order of objects in a subset as the order relation could be interpreted as a precision relation [3],[4]. A more precise data object will generate a more precise visual object.
We will find a way to order data objects belonging to a subset O_i, i.e. aggregate data objects of the same type.

We define a subset O_i^k

$$O_i^k = \{o_n \in O_i \mid S(o_n) = S_k\}$$

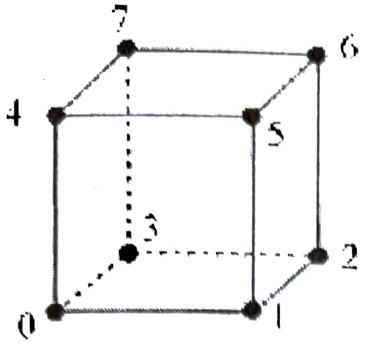
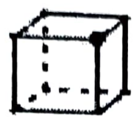
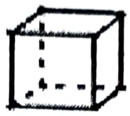


Figura 1 (Voxel)





of data objects of the same type (O_i) with the same entropy, S_k, where $S = k \ln \Omega$, and k is the Boltzmann constant and Ω is the thermodynamic probability. Agregate data objects, of the same type but missing a different number of attributes will have a different entropy. For instance, in the most simple 3D data object (a voxel, fig. 1) attribute could be missing in one, two, three,...or up to eight location or could be present in all. By analogy with a cristal (we consider missing a attribute in a data object similar to missing an atom in a cristal) we can define entropy by:

$\Omega = const \cdot N$

where N is the number of macrostates compatible with a given structure. If no atributes are missing the structure is considered perfect (N=1) , the entropy has the lowest value.

For a data object with one attribute missing, N=8 (the missing attribute could be in any of the 8 positions). We can generalize for a data object with **p** out of **n** attribute missing, where $N = C_n^p$. We can now use the value of the entropy to order data objects

o_n having different entropy $o_n \in O_i$ ($O_i \subset \mathbf{O}$)

The problem we might encounter is that more than one object have the same entropy. If we note O_i^k a subset of O_i having the same entropy S_k then the problem we need to solve is to order isoentropic data objects $o_1^k, o_2^k, \ldots, o_n^k \in O_i^k$, having th entropy S_k.

One way to do it is to use the cartesian distance d_j to their respective average value $<o>$.

$$<o>= AVERAGE(o_1^k, o_2^k, \ldots, o_n^k)$$

$$d_j \equiv \left|\langle o \rangle - o_j\right| \text{, j = 1,2,..,n}$$

One more issue to address here is whether the model, operations and structure defined over the data objects $o \in O$ could be extended over the other sets, i.e. O', G and G'.

In what concerns objects $o' \in O'$ this could be very easy checked as rigorous visualization (mathematical) algorithms are applied to data objects to filter initial values and/or generate new values.

In what concerns objects g∈G and g'∈G', since computers generate displays as data objects, models for physical visualization objects (g') are similar to logical visualization objects [1].

4. Writing analytic requirements on the visualization function

The visualization process could be called "scientific visualization" if the classes of functions - expressing different visualization goals – that satisfy the above conditions are identified and used. The first concern is to make sure that data objects' models operations and structures are preserved over the visualization process.

In this section particular requirements on the visualization function are presented. The case studied is

$Viz(o) = Rep(Map(Mat(o))) = g'$

Assuming the model, the operations and the algebric structure introduced in section 3 of this paper. The requirement we impose on Viz() – for some particular visualization purpose - is that the function must be linear. As stated before, the conditions must be defined in terms of structures on O,O',G and G'.

$$SUM_O(o,o_1,o_2) = SUM_{O'}(Mat(o), Mat(o_1), Mat(0_2)) =$$
$$= SUM_G((Map(Mat(o), Map(Mat(o_1), Map(Mat(o_2)) =$$
$$= SUM_{G'}(Viz(0), Viz(o_1), Viz(o_2))$$

and

$$o = \lambda \bullet o_1 \Rightarrow Mat(o) = \lambda \bullet Mat(o_1) \Rightarrow Map(Mat(o) = \lambda \bullet Map(Mat(o_1) \Rightarrow$$
$$Viz(0) = \lambda \bullet Viz(o_1)$$

5. Conclusions

The design of the scientific visualization requires the identification of the classes of functions that meeting different visualization goals while satisfying at least the fundamental conditions of visualization: distinctiveness and expressiveness. The "decomposition" of the scientific visualization process as presented in Table 1, allows more accurate formulation of the analytic conditions on these classes of functions. The detailed example given on defining a model for the data object, defining operations and mathematical structures on the data sets and expressing the fundamental conditions on the visualization function could be extended to other data models and mathematical structures.

REFERENCES

[1] Williams L. Hibbard, Charles R. Dyer, Brian E. Paul, *"Towards a Systematic Analysis for Designing Visualizations"*, Scientific Visualization, IEEE Computer Society, 1997, pp. 229 - 251.

[2] MacKinlay, *"Automating the Design of Graphical Presentations of Relational Information"*, ACM Transactions on Graphics, Vol.5, Nr.2 1986, pp.110-141.

[3] W. Hibbard, C. Dyer, B. Paul, *"A lattice Model for Data Display"*, Proceedings of IEEE Visualization '94, 1994, pp. 310 - 317.

[4] B.A.Davey and H.A.Priestley, *"Introduction to Lattices and Order"*, Cambridge University Press, 1990.

[5] Gheorghe Fărcaş, Szilágyi Miklós, *"Fundamentele matematicii"*, Editura Universității "Petru Maior", 1997

[6] D. Rădoiu, G.D. Popescu, *"Introducere în ştiinţa sistemelor de calcul"*, Editura Universității "Petru Maior", 1999

[7] *Mica enciclopedie matematica*, Editura Tehnica, Bucuresti, 1975

[8] D. Radoiu, "*VTK in Desktop Scientific Visualization*", AET Conference Proceedings, Editura Universitatii Petru Maior, 1999, pp. 21-30

[9] C. Upson, Faulhaber, Jr. T., D. Kamins, D. Laidlau, D. Schelgel, J. Vroom, R. Gurwitz, A. van Dam, " The Application Visualization System: A Computational Environment for Scientific Visualization", Computer Graphics and Applications, vol9, nr.4, 1989

Computer Science Department, Petru Maior University of Tirgu Mures

Appendix
Table 1

VIZUALIZATION
$Viz(o) = Rep(Map(Mat(o))) = g'$

SCIENTIFIC MODELS		SCIENTIFIC DATA		TRANS- -FORMED DATA		IDEAL GEOMETRIC OBJECTS		VIZUAL OBJECTS	
	SIMULATION / DAQ	**O**	TRANSF. ALGORITHM *Mat(o)*	**O'**	MAPING *Map(o')*	**G**	RENDERING *Rep(g)*	G'	INTERAC—TION *I(g')*
		$o \in O$	$Mat(o) = o'$	$o' \in O'$	$Map(o') = g$	$g \in G$	$Rep(g) = g'$	$g' \in G'$	
Model generation: Mathematical models bazed on ideal mathematical objects. Models are checked and improved by simulation and/or experimental mesurements.	**Data acquisition:** (based on models) Aggregated scientific data objects.	**Defining models** for data objects "o" and **structures** over data sets, "O". Data objects are finite representations of complex mathematical objects.	Rigurous visualization (mathematical) **algorithms** applied to data objects filter initial values and/or generate new values (e.g. isorface generation, volume generation, associating a colour code to a scalar value)	Models and structures defined for input data should apply to **processed data objects**.	Processed data objects are mapped into ideal geometrical objects. The process is done through factorization of the complex data objects into primitive data and mapping them to gliphs and graphics primitives.	**Ideal geometrical objects.** A collection is such objects is called **logical vizualization** of a scene. These objects are usualy (nD), animated (t) and interactive (I).	The **rendering** process is done through traditional rendering algorithms: nD to 2D projection, rotation, translation, zoom, hidden surface removal, compozition, animation, lightning.	**2D visual objects**, finite reprezentations of the ideal geometrical objects. A collection is such objects is called **physical vizualization** of a scene. These objects are usualy (2D), *animated (t)* and *interactive*	**Visual observation** of the scene and **interaction** (I), i.e. control of the visual objects and navigation. Human perception could be described in terms of mathematical *strcutures.*