ANALELE Universitatii “Stefan cel Mare” Suceava – Anul V, 1998, Nr. 9 SECTIUNEA ELECTRICA

SCIENTIFIC VISUALISATION – A MATURE FIELD OF RESEARCH IN COMPUTER SCIENCE

Marius Dorian Zaharia
POLITEHNICA University Bucharest
str. Splaiul Independentei, nr. 313, 77206 Bucharest
zaharia@cs.pub.ro

Abstract. *The paper describes the main concepts and tools used by scientific visualisation, a technology for improving the man machine interface. It is a field of application that was rapidly growing in the last years through the development of new algorithms, new theoretical approaches and interface technologies.*
Keywords: *scientific visualisation, data models, visualisation pipeline, data visualisation.*

Introduction

Many categories of applications are producing large and complex structured data sets; this raises problems of interpretation of the information these data sets include. Starting with 1987 a new research direction “scientific visualisation” is proposing the use of computer graphics techniques to create images in order to help understanding the massive numerical data representations which result mainly from scientific applications. The paper of Mc Cormick and De Fanti ([8]) is considered a real manifesto of this research area.

Such large data sets could be obtained through numerical simulation processes, measurements (for example applications in the domains of medicine, astronomy and seismology) or modelling processes (for example molecular modelling).

Some scientific visualisation systems could be of general use. These systems allow the user to select a visualisation type appropriate to a certain data set. Such systems serve to explore the structure and characteristics of the data sets based on different modalities to bind the data values and their visual representations. Usually the data representations are done by markers (icons or glyphs) which could be points, lines, surfaces, volumes and have associated graphic attributes (shape, colour, texture, length, direction) that allow the system to provide more information concerning these data.

Other scientific visualisation systems are those specialised to a narrow field of application (computational fluid dynamics, computer aided tomography etc.)

The process of visualising large data collections includes mainly the following stages:

- Data generation (through measurements or simulation processes) and validity check.
- Data analysis and reduction. This stage consists in detecting the structure of the data or some quantitative features that allow the data to be partitioned (classified) in subsets of smaller dimensions that could more easily be manipulated.
- Data modelling (using for example surface models, volume models or multiresolution models)
- The graphic representation of the models.

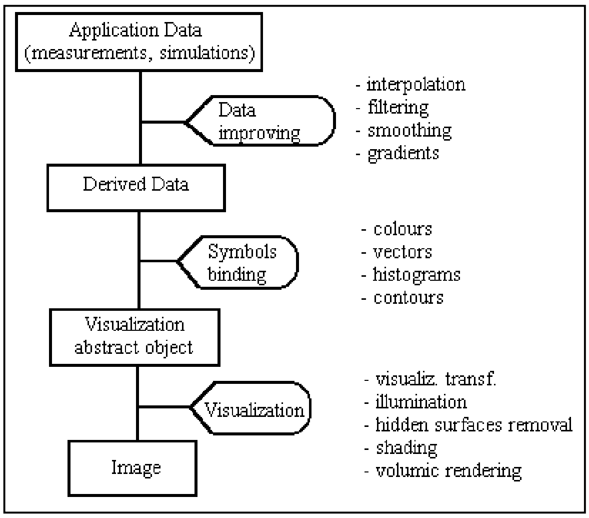

Figure 1 – Structure of a generic scientific visualisation application

]

A visualisation application could be represented by a directed graph whose nodes specify visualisation processes or data stores (that allow data storage and manipulation). The graph edges represent the direction of the data flow. This kind of functional model is referred as “visualisation pipeline”.

Data models in scientific visualisation

The applications from the three above-mentioned categories (simulation, modelling, data acquisition) are producing large multidimensional data sets. These data could be stored in shared data spaces implemented through data servers. The access to the information in distributed databases containing scientific data requires utilisation of standard data formats suitable to describe the previously specified data.

A data set represent a way to organise the data and the associated attributes. The organising structure is composed of cells and points; it has topologic and geometric properties. The topologic properties specify the relations between cells and points for example the data set could have a regular topology (if the cells are placed in regular positions) or a irregular one.

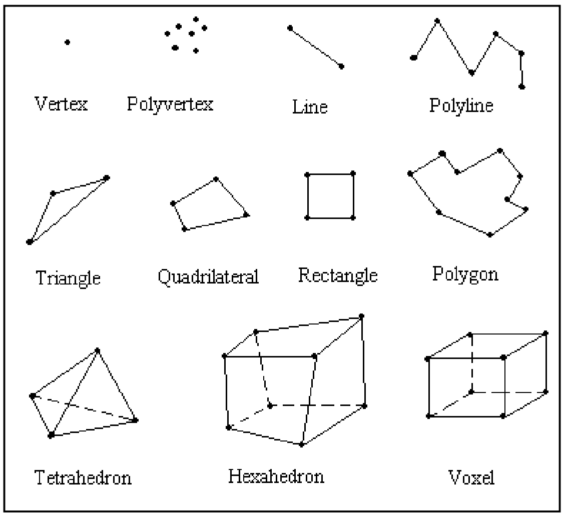

Figure 2 – Different cells categories

Typical examples of data sets in visualisation systems are: structured points (points and cells are placed on a regular, rectangular lattice), structured grids (the data set has a regular topology but an irregular geometry, in that case typical cells are quadrilaterals (2D) or hexahedrons (3D)), unstructured points (no topology), unstructured grids (both the topology and geometry are unstructured and is possible to use any combination of cell types).

Multidimensional scientific data could be generally assimilated with points in a N-dimensional geometric space. Each N-dimensional point includes coordinates (location parameters) that specify the point position in a geometric space. The other parameters included in the N-dimensional point are called data parameters.

Bergeron ([6]) proposed a general data model suitable to be used in scientific visualisation applications “the extended lattice model”. A lattice has three components:

- Data (the data values stored in the data tuples)
- Topology (defines the adjacency information between data elements). The connectivity of a lattice can be specified by connectivity methods defined on the indexes of the associated array.
- Geometry. This component includes the specification of the location parameters and a mapping from (topology)x(location)x(data) to (geometric space). Through different geometry mappings one could obtain different interpretations of the data being stored in one lattice.

Other researchers ([4]) developed a model based on the concept of “fiber bundles”. This description defines a base to describe the domain of independent variables and a space of dependent data. A field could be considered as a function from the base to the dependent data.

The data models appropriate for scientific visualisation systems must allow:

- The possibility to specify the sampling grids of the data space (these grids could be regular, rectilinear or irregular).
- The possibility to specify variables with associated dimensionality (spatial, spectral or temporal) as well as relations between

variables. Each data set could be described through one or more variables.
- Specification of singular data values.
- Specification of metainformation as the limits of the data values, the units used in the measurement process or the discontinuities of the data space.

Attribute data includes information describing the structure of the data set. This structure specifies the topology and geometry of the data set. The attributes could be associated to each of the points or cells of the spatial visualisation grid as well as to the whole data space. The type of a data set specifies the structure of the data and the relationship between cells and points.

In the following, three widely accepted scientific data models will be mentioned.

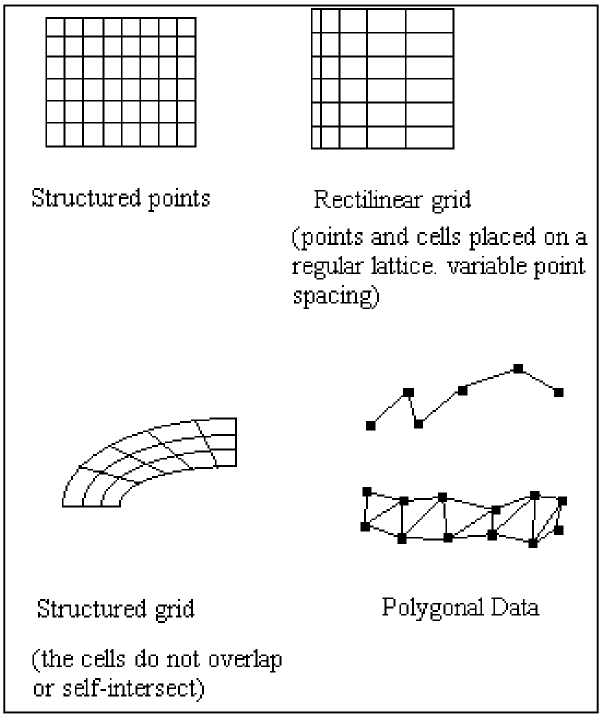

Figure 3 – Different data sets

A standard data format used for multidimensional data specification the Network Common Data Format (netCDF) was developed at NASA Science Data Systems Standards Office and University Corporation for atmospheric research (UCAF). This format is provided with a library whose routines are used to access the data. The data entities supported by netCDF are: variables, dimensions, attributes. The variables are multidimensional arrays (having all the same base type). A dimension is a descriptive name of one axis of a multidimensional array. A variable is characterised by name, base type and shape (i.e. list of dimensions). The attributes that describe metainformations about the entire file (or only a single variable) could be character strings, scalar values or vectors.

The HDF (Hierarchical Data Format) developed at National Centre for Supercomputing (NCSA) at the University of Illinois at Urbana-Champain is also a well known multiobject file format for transferring graphical or scientific data between machines. This format allows representation of 2D raster data as well as N-dimensional data (through the data type SDS – Scientific Data Set) i. e. N-dimensional arrays (with integer or float components), Vsets (allow storage of unstructured grids or other non-Cartesian aggregates), scales and units.

N-dimensional data could be described in AVS data format. It is a format specific “Application Visualisation System” a scientific visualisation program build at Stardent Computer Inc. This format allows also specification of data fields representing N-dimensional arrays. These data are samples of the data space. Sampling could be done conforming to uniform, rectilinear or irregular grids.

The system allows creating, editing and manipulating visualisation graphs. AVS has an executive that could control the execution of a visualisation graph even in case of distributed or parallel applications.

The data model could specify primitive data (as byte, integer, real or string) and aggregate data (fields, colour maps, geometries and pixel maps). The field is an N-dimensional array whose components could be scalar or vector data. This array could have any number of dimensions. Each dimension has an arbitrary size. An associated mapping function describes the relationship between the data elements and the coordinates of the points corresponding to the data elements (i.e. the correspondence between the computational space and the coordinate space).

Algorithms used in scientific visualisation applications

In this section, the main classes of algorithms used to transform the scientific data from one representation to another will be described.

From a structural point of view these algorithms could modify the data topology (i.e. modify only the adjacency relationships between data elements), data geometry or the data attributes (for example creating new attributes for data without changing the structure of the data set).

According to the type of data they manage the algorithms could be classified in:

Scalar algorithms (that operate on scalar data). Some examples are colour mapping (where the values in a scalar domain are mapped into a finite colour set) contour finding (where boundaries between spatial regions with similar characteristics are determined). An algorithm from the last mentioned class is the marching cube algorithm ([7]) it determines a polyhedral boundary of a 3D region described by its interior.

Vector algorithms. Such categories of algorithms associate oriented line segments (or other types of icons) to selected points of a data space. The vectorial information could be emphasised by deforming the data geometry according to the vector field information. Animation algorithms are sometimes used to describe the directional information; for example in applications of computational fluid dynamics the vector fields are represented by lines (particle traces) that are traced by fluid particles during a time interval.

Some authors ([5]) classify the methods for representing vector fields as:

- Elementary point icons (arrow-like glyphs drawn in selected points of the data space)
- Elementary line icons (for example particle traces)
- Elementary stream surfaces (i.e. surfaces tangent, in each point of the data space, to the vectors of the field)
- Local icons are representations that specify information about vector gradients. For example critical points of the vector field i.e. points where the amplitude of the vectors equals 0 and the slope of the streamlines is not determined. The critical points of an N-dimensional vector field are classified according to the eigenvalues of the Jacobian matrix.
- The global icons describe the global structure of the data field through simple diagrams that abstract the overall behaviour of the local representations. The global representations of a vector field are based on topological concepts; the field is divided in regions bounded by 3D surfaces (or 2D curves) that are intersecting in critical points.

In case of tensor data there are methods to visualise a tensor field by representing his eigenvectors corresponding to points placed on a discrete grid. Delmarcelle ([2]) propose a representation of a 3x3 symmetric tensor, based on hyperstreamlines. A hyperstreamline is a sweep resulting from sweeping the 3D space by an ellipse whose centre is moving along a field line (tangent in each point to one of the tensor eigenvectors). The semiaxes of the ellipse are oriented along the other two eigenvectors of the tensor field.

In case of nonsymmetric tensor fields the eigenvectors are not orthogonal and the eigenvalues could be complex numbers. In such cases the scientific visualisation applications have developed tensor representation methods based on: symmetric/antisymmetric decomposition or the polar decomposition of the field ([5]).

A special class of fundamental algorithms used in scientific visualisation applications is formed of modelling algorithms. This category of algorithms creates or modifies the geometry or the topology of the data set. Such algorithms are used in the early stages of the visualisation pipeline, they create geometry (or combinations of geometric primitives as spheres, cones, wedges, etc.) from scratch or they read data files that contain data sets descriptions. This kind of simple geometric objects could specify real world objects or auxiliary information necessary to improve the quality of data representation (for example the reference axes of the 3D data space). It is equally possible to create (or modify) procedurally, data attributes (for example textures mapped on the surfaces of different geometric objects).

Surfaces form the basis of modelling objects in computer graphics. They describe common geometric shapes, region separation (in a 3D Euclidean data space) and are used to visualise mathematical descriptions. Another surface based technique to visualise multidimensional data is data cutting. This method consists of sectioning a data set with a surface and then displaying interpolated data values from the surface. Surface models can be used to model scattered data sets ([9]) (i.e. data that do not have any special configuration).

Volume modelling include those methods used to represent the attributes of 3D objects and their interiors (which, in this case are not considered to be homogenous).

Multiresolution models are useful when visualising large amounts of data (for example in case of direct volume rendering or surface rendering of scalar fields). In order to obtain a good interactivity, an application should be able to manipulate data at different levels of detail. Multiresolution models are appropriate to use in iterative algorithms to obtain good initial approximations of the solutions.

Conclusions

The domain of scientific visualisation has known a rapid evolution in the last years. General commercial products as AVS, Iris Explorer (Silicon Graphics) or Data Explorer (IBM), allow automatic generation of various representations of large scientific data sets. The methods specific to that field of research find a lot of applications in domains as: databases (where scientific visualisation techniques could be used in post-processing phase), medicine (Computer Aided Tomography and Magnetic Resonance Imaging applications require application of volume data visualisation techniques), chemistry, electromagnetism, meteorology, seismology, social sciences. Data visualisation is deeply interrelated with the fields of computer graphics and image processing.

This paper described topics as: user interface design, data modelling and processing algorithms required to obtain visual representations of large data sets.

References

[1] Brown, C. W. and Sheperd, B. J. (1995) *Graphics File Formats - reference and guide*, Manning Publications Co.

[2] Delmarcelle, T. and Hesselink, L. (1993) Visualizing Second Order Tensor Fields with Hyperstreamlines, IEEE Computer Graphics and Applications, july.

[3] Gallop, J. (1994) *Underlying Data Models and Structures for Visualization*, in Scientific Visualization Advances and Challenges (coordinator Rosenblum, L.), Academic Press Inc.

[4] Haber, R. B., Lucas, B. and Collins, N. (1991) *A Data Model for Scientific Visualization with Provisions for Regular and Irregular Grids*, in Proceedings of Visualization ’91 IEEE Computer Society Press, Los Alamitos, CA.

[5] Hesselink, L. and Delmarcelle, T. (1994) *Visualization of Vector and Tensor Data Sets*, in Scientific Visualization, Advances and Challenges (coordinator Rosenblum, L.) Academic Press Inc.

[6] Kao, D. T., Bergeron, D. and Sparr, T. (1994) *An Extended Schema Model for Scientific Data*, Database Issues for Data Visualization (edited by Goos, G., Hartmanis, J. and van Leeuwen, J.), Springer Verlag, Berlin, Heidelberg.

[7] Lorensen, W. E. and Cline, H. E. (1987) *Marching Cubes: A High Resolution 3D Surface Construction Algorithm*, Computer Graphics, July.

[8] Mc Cormick, B., De Fanti, H. and Brown, M. D. (1987) *Visualization in Scientific Computing*, Computer Graphics, No. 21.

[9] Nielson, G. M. (1993) Scattered Data Modelling, IEEE Computer Graphics and Applications, january.

[10] Schroeder, W., Martin, K. and Lorensen, B. (1998) *Visualization Toolkit, An Object Oriented Approach to 3D Graphics*, Prentice Hall Inc.