FlowVisualizer

a system for combined scalar and vector visualization

Harald Ruckser

Institut für Informatik, Abteilung für graphische und parallele Datenverarbeitung,
Universität Linz, Altenbergerstrasse 69, 4040 Linz, Austria

Abstract

In this paper a new system for visualization of combined scalar and vector fields is presented. The methods implemented in this system are based on the use of combination AVOs (abstract visualization objects) consisting of arrows, spheres, isosurfaces, streamlines and cylinders. These AVOs improve the understanding of correlations between the spatial distributions of one vector and up to three scalar quantities, and enable simple forms of topological filtering. Furthermore, new classification methods for visualization methods and AVOs are introduced, which, among others, describe the dimensionality of the data that a single AVO displays (DD - displayed dimensions). An extended model of the visualization process reveals important details of the microstructure of visualization, its transformational phases and objects.

1 Introduction

The purpose of visualization is to gain insight into complex phenomena [Miya93]. These phenomena can be examined by simulations, observations or measurements producing what is usually called 'scientific data' [Haber90]. The visualization process can thus be considered as a tool for interpreting and communicating scientific data by transferring them into displayable images:

phenomenon
⇩ *simulation, experiment, or observation*
scientific data
⇩ *visualization process*
displayable image
⇩ *human perception and interpretation*
insight and understanding

Simulation, experiment, and observation include modelling and solution phases. The principal problem of visualization is to find the optimum way of transforming these scientific data into a displayable und understandable image, while preserving the accuracy of the information. This makes the visualization process depend strongly upon what it is meant to display: the properties of the data and the phenomenon to be explored.

It is this dependence that has to be kept in mind when examining complex physical phenomena that cannot be described sufficiently by scalar, vector or tensor quantities alone; these phenomena rather require a combined description considering scalar, vector and tensor properties. In fluid dynamics, for example, a description of a flow can be given by one vector (e.g. the momentum or velocity vector) and two scalars (e.g. pressure and temperature) for each point of the flow field. Vector fields are usually visualized by arrows, streamlines, stream surfaces or moving particle tracks, while scalar fields can be visualized using volume rendering, isosurfaces, scalar glyphs and other techniques [Brodlie92] [Hin93] [Wijk93].

Even though flow phenomena are basically understandable as an interplay of scalar, vector and tensor quantities, existing flow visualization techniques hardly allow for such a combined display; this is valid even for sophisticated vector visualization techniques (e.g. flow topology visualization [Helman91] [Globus91]) and tensor visualization techniques (e.g. [Delmarcelle93] [Haber90] [Leeuw93]). In this work, a systematic approach for combined display is presented and specified: Combination AVOs consisting of different parts (elementary AVOs) that represent up to six quantities in up to three dimensions. This allows an improved adaptation of visualization techniques to the requirements of complex flow phenomena.

2 Classifications of the visualization process

Visualization of scientific data has been a quickly emerging field of research over the past few years, both from an experimental and from a theoretical point of view. The principal model presented by Haber and McNabb [Haber90] describes visualization as a process leading from simulation data to a displayable image by a sequence of three generalized mappings. These mappings transform raw data obtained from simulation, observation, or experiment into a geometric abstraction (called AVO), which is rendered to a displayable image (for details see [Haber90]):

simulation data
⇩ *data enrichment / enhancement*
derived data
⇩ *visualization mapping*
abstract visualization objects (AVOs)
⇩ *rendering*
displayable image

However, some important questions on the structure of the involved data and objects are left open in this model.

2.1 Extended model of visualization

Figure 1 illustrates an extended model of the visualization process. It is based on the model presented by Haber, but adds some important details on the 'microstructure' of the data and objects involved in the visualization process.

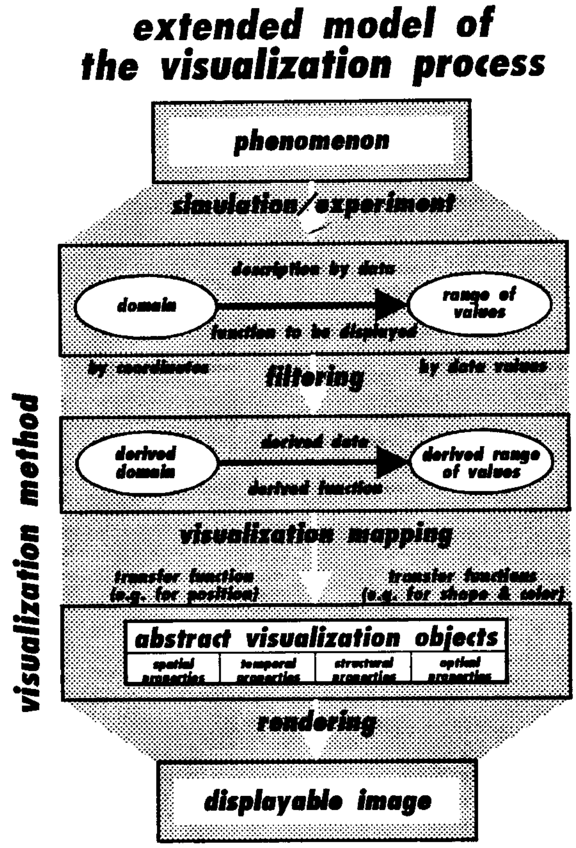

Figure 1. Extended model of visualization

These details turn out to be crucial for understanding the classification schemes for the visualization process that are both reviewed and newly presented in the theory part of this paper. A thorough understanding of these classification schemes, on the other hand, facilitates the comprehension of combination AVOs as used in the FlowVisualizer.

The exploration of the phenomenon starts with a first step generating the raw data to be displayed in the visualization process. This step may include simulation, observation and/or experiment; it is not a part of the visualization process as defined in this context, but it determines the properties of the raw data. To be accurate, visualization is defined as the process leading from raw data to a displayable image; the techniques used in this process are defined as visualization methods. The data correspond to a function that the data are created by. (In the case of a spatial distribution of temperature, for example, the function $f\colon (x,y,z) \rightarrow T$ corresponds to data tuples (x,y,z,T) containing all dependent and independent variables.) The domain and the range of values of this function are main points of some of the classifications presented below.

In the filtering step, derived data are created from the raw data. Their structure (data - function - domain / range of values) is basically the same.

2.2 AVO properties

Visualization mapping generates geometric abstractions called abstract visualization objects (AVOs); their properties may be classified as spatial, temporal, structural, or optical (see figure 1). Spatial properties include form, shape, orientation, relative scaling, and position; temporal properties can be temporal presence or disappearance of objects and a temporal variation of all other properties.

Structural properties is a term used for object hierarchy and structure, determining the subdivision and connection between structures (We differentiate between 'indivisible' elementary AVOs and compound AVOs consisting of other compound AVOs and/or elementary AVOs).

Optical properties can be roughly divided into three classes: (I) properties that, in a real object, would depend on the dielectric function of the material (see [Wooten72]), including the derivable quantities such as refraction index and reflectivity: colour, transparency/opacity, and the optical modelling properties for ambient, diffuse, and specular light, as well as the colour and intensity of light generated by the object itself; (II) mechanical properties of the surface, such as roughness or structure (as long as they are so small that they influence the looks rather than the shape of the object); (III) properties that are generated by a spatial variation of classes (I) and (II), such as surface and volume textures.

Further insight into the principles of a given visualization technique can be gained by analysing which variables of the domain (independent variables) and which variables of the value range (dependent variables) are mapped to a certain category of AVO properties.

2.3 Three step model of visualization mapping

The definition and generation of all these properties is a part of visualization mapping and is performed by the help of transfer functions (*"... visualization mapping constructs an... AVO ... from the derived data..."*, *"Transfer functions define simple mappings between the... data and the AVO fields"* [Haber90]). So far, there have been very few hints on an accurate formulation of the procedural structure of visualization mapping and of its relationship to transfer functions. Evidently, transfer functions make up for an important part of visualization mapping, but not for all of it. In my opinion, visualization mapping can be considered a three step process involving definition, relation and generation:

(I) defining the properties of the AVO (creative phase, AVO definition phase), answering the question: Which properties (see 2.2) is the AVO supposed to have ?

(II) relating transfer functions to the AVO properties (relation phase, transfer function relation phase); it is here that the properties of the dependent and independent variables of the derived data are related to the AVO properties defined in step (I).

(III) generating the AVOs (AVO generation phase, calculation phase).

Steps (I) and (II) have to be carried out or influenced interactively by the programmer or user developing the visualization, whereas step (III) is usually done by the computer. Finding the best combination between AVOs and transfer functions is, in fact, a cyclic and iterative process since the properties of the data may have considerable impact on the designing of the AVOs. This requires steps (I) and (II) to be performed simultaneously, as they show two sides of the same medal.

2.4 Classification schemes

The extended model of the visualization process presented above makes it clear where a classification of visualization can set in: It may give information on both *what* is displayed (the data) and on *how* it is displayed (the visualization process itself, involving the three steps filtering, visualization mapping and rendering).

The synopsis of AVO properties (spatial - temporal - structural - optical), the three phase model of visualization (filtering - visualization mapping - rendering) and the three step model of visualization mapping (definition - relation - generation) just mentioned focus on the visualization process (i.e. the '*how*' question). But what about actual classifications of the data and objects involved in this process?

The scheme by Brodlie et al. [Brodlie92] is essentially a classification for the functions (and corresponding data) that are depicted in the visualization process, thus giving an answer to the question *what* is being displayed. It describes these functions by the dimension of their domain and their range of values (the latter determines the type of function - scalar, vector, or tensor), thus limiting the choice of suitable visualization methods in a sensible and helpful way.

A distribution of a scalar variable with domain of dimension *n* corresponds to category E_n^S. Vectors and tensors are symbolized by ^{V} and ^{T}, respectively; $E_3^{V_3}$ corresponds to the display of a 3-element vector in 3D space. For details, see [Brodlie92].

Figure 2 visualizes the scope of the classification by Brodlie et al. (according to our interpretation): It describes the function (i.e. data) to be displayed by its dimensionality, which predetermines the selection of visualization methods. So, the original classification of the function to be displayed is transferred to the pertinent visualization methods: For example, isosurfaces and direct volume rendering techniques suitable for 3D scalar functions are classified as E_3^S visualization methods. This makes it clear that the starting point of this classification is not the visualization method itself, but what it is meant to display; it does, however, not give any direct information on the objects used for visualization.

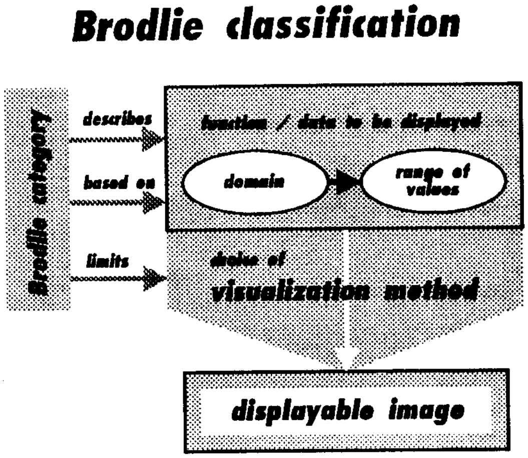

Figure 2. Principle of the classification by Brodlie et al.

2.5 DD classification

The author hence suggests introducing a novel classification named 'DD-classification', which describes transfer functions used for visualization mapping and the abstract visualization objects they create.

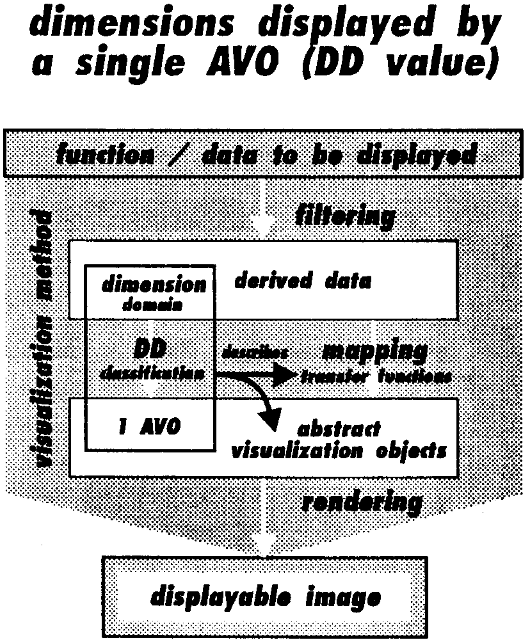

Figure 3. Principle of the DD classification

DD stands for 'displayed dimensions'; the DD value represents the dimension of the subset of the domain that is displayed by one single AVO. The DD value determines whether an AVO is a point, line, surface, or volume AVO. A point AVO represents data values at a zero-dimensional point; a line AVO represents data values along a one-dimensional, continuous entity (a curve), and so on. Figure 3 illustrates how the DD classification can be related to the visualization process. Unlike the Brodlie classification, the DD value is actually a property originating in the visualization mapping method and the AVOs it generates.

Table 1 shows the DD classification of some common visualization objects for data in two or more dimensions; for comparison, the corresponding Brodlie classes and the extended DD classes (explained below) are also specified.

For an AVO depicting a function with a domain of dimension n (Brodlie category E_n), DD $\leq n$. This follows from the fact that a single AVO displays the data from coordinates within a subset S of the domain D ($S \subseteq D$).

The rareness (or even lack) of AVOs with DD > 2 for vectors and DD > 1 for tensors shows the potentialities for future 'AVO developers'. The problem with AVOs with higher DD values for vectors and tensors is the large amount of information such an AVO would have to carry. This makes them difficult to realize; consequently, AVOs with low DD values dominate for vectors and tensors.

DD	AVO type	scalars (S)	vectors (V)	tensors (T)
0	point	*S0* $E_{2-3}^{(n)S}$ scalar glyphs	*V0* $E_{2-3}^{V_{2-3}}$ arrow glyphs hedgehogs	*T0* $E_3^{T_{3:3}}$ ellipsoids Haber glyphs [Haber90] local flow probe [Leeuw93]
1	line	*S1* E_2^S isolines	*V1* $E_{2-3}^{V_{2-3}}$ streamlines particle tracks topology analysis [Helman89]	*T1* $E_3^{T_{3:3}}$ hyperstreamlines [Delmarcelle93]
2	surface	*S2* E_2^S coloured surfaces slices 2D contour plots *S2* E_3^S isosurfaces basket weave [Sewell88]	*V2* $E_3^{V_3}$ stream surfaces (ribbons, tubes), surface particles [Wijk93b] [Hultquist92] topology analysis [Helman91]	
3	volume	*S3* E_3^S volumetric AVO (for volume rendering)[Drebin 88] [Elvins92]	*V3* $E_3^{V_3}$ stream volumes [Wijk93]	

Table 1. DD classification of common visualization methods (including extended DD classes and Brodlie categories)

In table 1, the term glyphs is used for point AVOs, in particular, containing information for single points ('Glyphs are symbols, signs, markers or icons' [Brodlie92]).

A useful extension of the DD classification describes an AVO (or the transfer function, respectively) by a letter saying what the AVO displays and a number giving its DD value (extended DD classification). The letter can be S, V, or, T for scalar, vector, or tensor; see table 1.

It must be noted that the DD classification and its extended form basically describe elementary AVOs displaying one single quantity. Nevertheless, they can also be generalized to classify combination AVOs depicting two or more quantities; in this case, each AVO part must be classified separately (see the following section 2.6).

2.6 Combination AVOs

The concept of combination AVOs helps to improve visualization techniques for complex phenomena describable by scalars, vectors and tensors. A coloured arrow, for example, might represent a vector by its length and direction and a scalar quantity by its colour, displaying both quantities for a single point (DD = 0) in space; this makes it a S0-V0 combination AVO. As the arrow does not consist of different parts with each of them showing different quantities, it is an elementary combination AVO (in contrast to a compound combination AVO consisting of two or more elementary parts). In isodimensional combination AVOs, all AVO parts have the same DD value (as in the arrow example above; this implies that isodimensional combination AVOs consist of elementary AVOs placed in the same row of table 1). The opposite is named heterodimensional: Haber et al. [Haber90] place their cylindrical tensor AVOs in a coloured plane representing scalar values, thus combining a single surface scalar and several pointwise tensor AVOs (compound S2-T0 AVO). Heterodimensional combination AVOs consist of elementary AVOs from different rows of table 1.

The classifications presented in section 2 clarify how the visualization techniques forming the 'FlowVisualizer' fit into the basic concepts of visualization and why they meet the requirements of a combined scalar and vector display.

3 FlowVisualizer

The main motivation for developing the system called 'FlowVisualizer' was to create a new and better way of visualizing (static) flow fields, which allows for a combined display of scalar and vector quantities. As there is evidently no unique optimum AVO meeting the requirements of all possible flow fields, a wide choice of combination AVOs is offered. This enables the user to adapt the visualization process to the data set by an interactive selection and optimization of methods and parameters.

The FlowVisualizer was realized as an AVS application network (see [Upson89] [Convex92]) consisting of both existing standard and newly developed modules. The system runs on a Convex C3440 supercomputer with Convex-AVS. Figure 4 shows a simplified block diagram of the system. It shows how the three transformations of visualization (filtering, visualization mapping and rendering) are carried out and how data types change during the process.

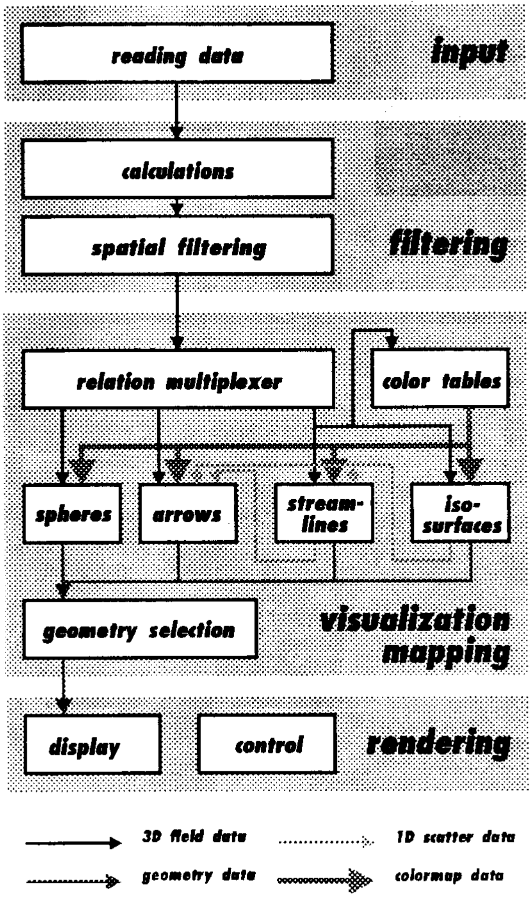

Figure 4. Block diagram of the FlowVisualizer

3.1 Input and filtering

Input data are read in using Plot3D format [Buning90], giving the fluid properties density, x-, y-, z-momentum, and stagnation for each gridpoint of a structured, irregular grid [Gelberg90]. In a prefiltering step seven additional physical quantities are calculated (energy, pressure, enthalpy, mach number, temperature, total pressure, and total temperature; see [Buning90]); this and simple spatial filterings such as cropping (selecting a logical subcube of the data set) and downsizing (selecting every n-th gridpoint in each direction, which reduces the size of a three-dimensional data set by a factor of n^3) can be performed using standard modules.

More sophisticated filtering techniques include scalar limit selection and iso point selection. Scalar limit selection is performed by selecting those gridpoints where one of the scalar quantities lies within a certain interval. Iso point selection is done by interpolating a set of points lying between gridpoints where a scalar quantity has a certain value. This corresponds to the point coordinates of a generated isosurface, discarding the connectivity and surface direction information. These filtering algorithms render it possible to focus on a particular region of interest and to look into the inside of complicated flow features. Moreover, the data reduction shortens calculation times for the following mapping and rendering steps.

3.2 Visualization mapping and AVOs

The filtered data can be related to properties of the selected combination AVOs by using a relation multiplexer structure, allowing any physical quantity contained in the data to be mapped to any of the AVO properties. Vector properties (i.e. the momentum vector) can be mapped to the direction / size of arrows and streamlines, scalar properties to colour, size, positions of isosurfaces, and others. Scalar and vector mapping is performed by a selection of iso- and heterodimensional combination AVOs.

Isodimensional combination AVOs are sphere-arrow AVOs (extended DD class: S0-V0) and cylinder-arrow AVOs (S0-V0). In the sphere-arrow AVO, colour and size of the sphere plus colour of the arrow represent three scalars, direction and length of the arrow represent a vector (see figure 5). In the cylinder-arrow AVO, the colours of an arrow and up to three concentric cylinders around the arrow display up to four scalars, while the direction and height of the resulting conic AVO display a vector; the steplike conic structure is created by the decreasing height of the interlocked cylinders; see figure 6.

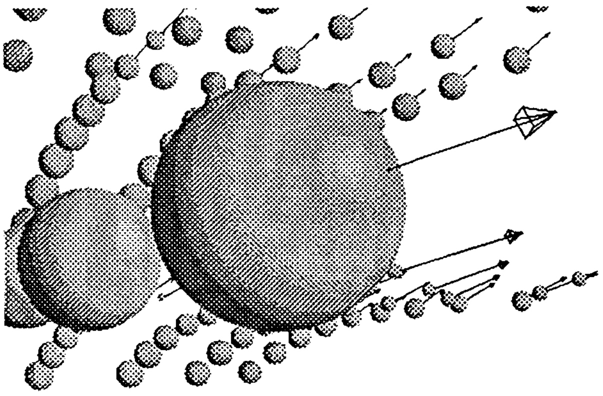

Figure 5. Sphere-arrow AVO

In both cases, the combination improves depth and direction recognition of the AVOs (as compared to arrows alone). All properties are shown pointwise (DD = 0) and for all gridpoints selected in the spatial prefiltering; as the eye is not well adapted to interpreting hundreds of little arrowlike glyphs [Helman89], this technique is limited to a small number of points. Nevertheless, depth perception is considerably improved by the integration of surface-like elements (spheres and cylinders) as compared to arrows alone (drawn by simple line graphics).

Heterodimensional combination AVOs are the isosurface-arrow AVO (extended DD class: S2-V0) and the isosurface-streamline-belt AVO (S2-V1-S0). The former consists of an isosurface (with its colour and its position / isolevel representing two different scalars) and several arrows, which are attached to the cornerpoints of the surface facets and display the local values of a vector field (by their direction and length) and a scalar field (by their colour); see figures 10 and 11. This is an example of surfacelike scalar display (extended DD class S2) combined with pointwise vector (plus scalar) display (class V0 or V0-S0). The cornerpoints of the surface facets are marked by small spheres, which improve the recognition of the arrow direction. This AVO renders it possible to display the behaviour of additional scalar and vector properties at points with an identical scalar property (i.e. an isosurface). This connects the advantages and disadvantages of surface fitting and point-glyph visualization methods. The good depth perception of the surface is further improved by the arrows and spheres attached to it, which mark and emphasize the corners and edges of the surface facets. By placing the isosurface at a low mach number level, for example, the velocity vectors around critical points in a flow field can be displayed (as the mach number is proportional to the amount of the velocity vector).

The isosurface-streamline-belt AVO consists of an isosurface (as above, extended DD class: S2) and streamlines (class V1), which are integrated from the cornerpoints of the surface facets (see figures 8, 9, and 12). The streamlines display a vector field (by their shape and direction) and a scalar field (by their colour); a number of belts around each streamline display an additional scalar field for selected points (class S0). As there is a belt around every *n*-th integration point of the streamline, the belts are lying more closely along the streamlines in regions of faster flow, which improves the velocity display. Since the streamlines are depicted as tubes with a variable, finite diameter, spatial perception is further enhanced by illumination effects and mutual occultation (as compared to simple lines of width zero). This AVO can be used to integrate streamlines beginning near critical points (with velocity $v \approx 0$, corresponding to mach number $M \approx 0$), which allows a simple form of topological selection of starting points. Other possibilities include visualizing heat convection by selecting starting points on an isosurface of high temperature, i.e. near a source of heat. In addition to displaying scalar information, the belts (or cylinders) around the streamlines improve the perception of streamline direction and orientation.

To sum it up, combination AVOs do not only show the properties represented by the elementary AVO parts they consist of, but also enhance the understanding of relations between the spatial distributions of several independent variables. Depth perception is improved by (standard shading) illumination effects and by cues given by AVO parts covering other AVO parts (i.e. mutual occultation): The orientation of an arrow drawn as a simple line plus an arrow head may be hard to discern, for example; by placing the arrow on top of a sphere, however, one can easily see that the starting point of an arrow pointing away from the viewer is on the hidden side of the sphere, whereas the starting point of an arrow pointing towards the viewer is clearly visible. Cylinders or tubes (without top and bottom) wrapped around arrows and streamlines also improve direction perception since the side - bottom or top - through which the inside of the cylinder is visible is likely to be closer to the viewer. The form of an isosurface can be elucidated by spheres attached to the cornerpoints of its facets; in this case, the direction of the line of intersection between the surface and each sphere gives an important cue. All these examples show that a compound combination AVO is definitely more than just the sum of the parts it consists of.

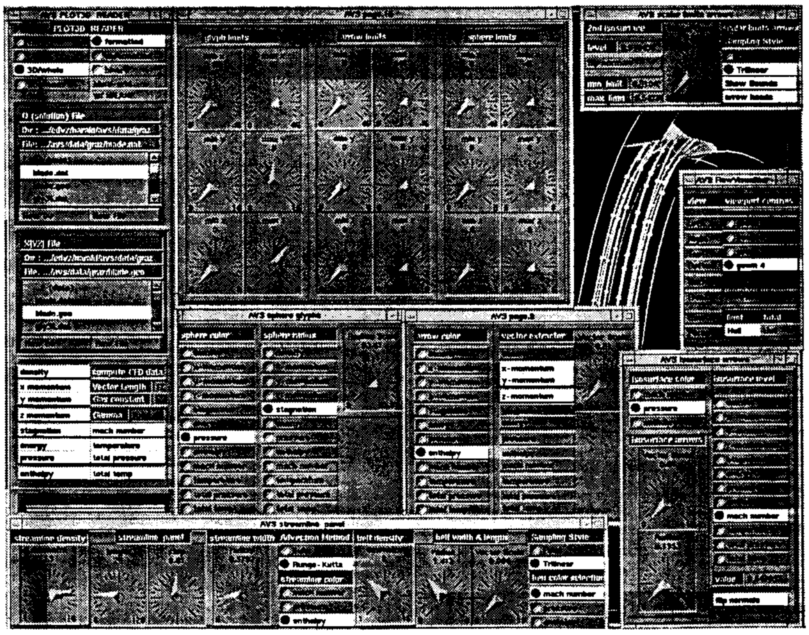

Figure 7. FlowVisualizer: user interface

3.3 User interface

Figure 7 shows the AVS-based graphical user interface of the FlowVisualizer. It ensures an interactive control of visualization techniques and parameters.

The windows on the left top (named plot3d reader) and on the middle top (with 6x3 dials) are used to define input sources and parameters as well as filtering steps. The other windows contain the relation multiplexer structures (with choice buttons for selecting the physical quantity displayed by a particular AVO property; the selected quantity is highlighted) and additional switches and dials that control visualization mapping and display parameters.

Two parts of the user interface are not shown in figure 7: the colour legends that visualize the mappings between colours of AVO parts and numerical values of displayed data as well as the choice buttons for selecting the scalar quantities displayed by the cylinder-arrow-glyphs.

3.4 Application examples

Figures 8, 9, 10, and 12 show details of the flow between the blades of a turbine; figure 11 depicts the flow inside a Laval nozzle. Figures 8, 9, and 12 use isosurface-streamline combination AVOs, figures 10 and 11 isosurface-arrow AVOs; for details of displayed quantities, see table 2. In all these figures, the directions of streamlines and arrows visualize the momentum vector.

Working with the FlowVisualizer has turned out to be useful for examining flow phenomena; zooming in on flow details, direct interaction, variation of mapping and display parameters make it a useful and easy-to-use tool for interpreting large data sets.

fig.	isosurface level	isosurface colour	arrow colour	streamline colour	belt colour
8,12	mach no.	pressure		enthalpy	mach no.
9	total press.	pressure		enthalpy	pressure
10	total press.	temperature	energy		
11	total press.	pressure	energy		

Table 2. Physical quantities depicted in figures 8 - 12

4 Conclusion

In this work, new classifications of the visualization process and new visualization techniques for combined scalar and vector fields are presented. New classifications include an extended model of visualization, a three step model of visualization mapping, the DD (displayed dimensions) classification for abstract visualization objects and categories of AVO properties, as well as an extended DD classification used for describing combination AVOs. These classifications enable a systematic look at how visualization works and what its potentialities are. New flexible and versatile visualization techniques presented make use of combination AVOs consisting of elementary parts such as spheres, arrow glyphs, isosurfaces, streamlines, cylinders, and tubes. They are integrated into an AVS-based visualization network called FlowVisualizer. These AVO combinations render it possible to display several phyical properties of a flow field - scalars and vectors - at a time, and allow for an improved perception of depth, relations between different variables and topological peculiarities of the flow.

Acknowledgements

Thanks to W. Sanz (Technische Universität Graz) for supplying interesting CFD data of turbines and nozzles.

References

[Buning90] P. Buning: *PLOT3D Reference Manual*, NASA Technical Memorandum 101067, March 1990

[Brodlie92] K.W. Brodlie ed.: *Visualization Techniques*, in: K.W. Brodlie, L.A. Carpenter, R.A. Earnshaw, J.R. Gallop, R.J. Hubbold, A.M. Mumford, C.D. Osland, P. Quarendon eds.: *Scientific Visualization - Techniques and Applications*, Springer, Berlin 1992

[Convex92] *Using ConvexAVS to Visualize Data*, Convex Corp., 1992

[Delmarcelle93] Th. Delmarcelle, L. Hesselink: *Visualizing Second-Order Tensor Fields with Hyperstreamlines*, IEEE Computer Graphics & Applications, July 1993, pp. 25-33

[Drebin88] R.A. Drebin, L. Carpenter, P. Hanrahan: *Volume Rendering*, Computer Graphics, Volume 22 (4), August 1988, pp. 65-74

[Elvins92] T.T. Elvins: *A Survey of Algorithms for Volume Visualization*, Computer Graphics, Vol. 26 (3), August 1992, pp. 194-201

[Gelberg90] L. Gelberg, D. Kamins, D. Parker, J. Sacks: *Visualization Techniques for Structured and Unstructured Scientific Data*, Siggraph '90 Course Notes, 1990

[Globus91] A. Globus, C. Levit, T. Lasinski: *A Tool for Visualizing the Topology of Three-Dimensional Vector Fields*, Proceedings Visualization '91, pp. 33-40

[Haber90] R.B. Haber, D.A. McNabb: *Visualization Idioms - A Conceptual Model for Scientific Visualization Systems*, in [Nielson90]

[Helman89] J. Helman, L. Hesselink: *Representation and Display of Vector Field Topology in Fluid Flow Data Sets*, Computer, August 1989, pp. 61-72

[Helman91] J. Helman, L. Hesselink: *Visualizing Vector Field Topology in Fluid Flows*, IEEE Computer Graphics & Applications, May 1991, pp. 36-46

[Hultquist92] J.P. Hultquist: *Constructing Stream Surfaces in Steady 3D Vector Fields*, Proceedings Visualization '92, pp. 171-178

[Leeuw93] W.C. de Leeuw, J.J. van Wijk: *A Probe for Local Flow Visualization*, Proceedings Visualization '93, pp. 39-45

[Mendez90] R.H. Mendez ed.: *Visualization in Supercomputing*, Springer, 1990

[Miya93] E.N. Miya: *Visualization - Frequently Asked Questions*, Internet newsgroup comp.graphics.visualization, March 1993

[Nielson90] G.M. Nielson, B.D. Shriver, L. Rosenblum eds.: *Visualization in Scientific Computing*, IEEE Computer Press, 1990

[Sewell88] G. Sewell: *Plotting Contour Surfaces of a Function of Three Variables*, ACM Transactions on Mathematical Software, Vol. 14, No. 1, March 1988, pp. 33-44

[Tufte83] E.R. Tufte: *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, Conn., 1983

[Upson89] C. Upson, T. Faulhaber, D. Kamins, D. Laidlaw, D. Schlegel, J. Vroom, R. Gurwitz, A. van Dam: *The Application Visualization System: A Computational Environment for Scientific Visualization*, IEEE Computer Graphics & Applications, Vol. 9, No. 4, pp. 30-42, 1989

[Wijk93] J.J. van Wijk: *Flow Visualization with Surface Particles*, IEEE Computer Graphics & Applications, July 1993, pp. 18-24

[Wijk93b] J.J. van Wijk: *Implicit Stream Surfaces*, Proceedings Visualization '93, CS Press, Los Alamitos, Calif., pp. 245-252

[Wooten72] F. Wooten: *Optical Properties of Solids*, Academic Press, New York, 1972

colour tables

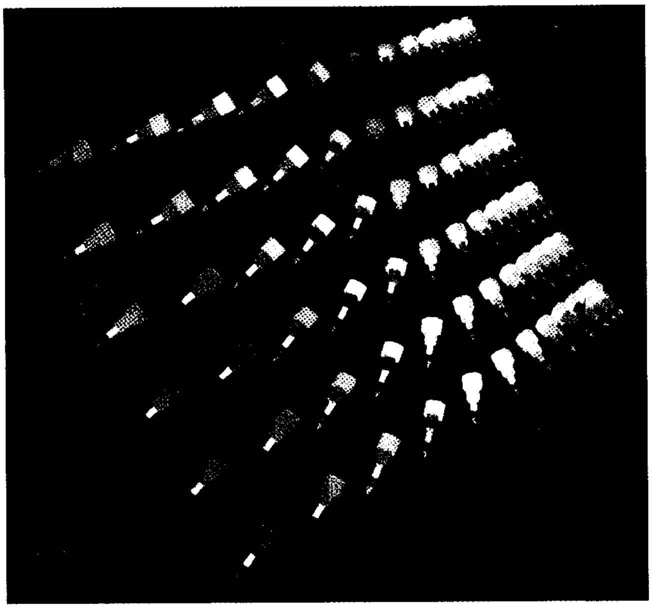

figure 6

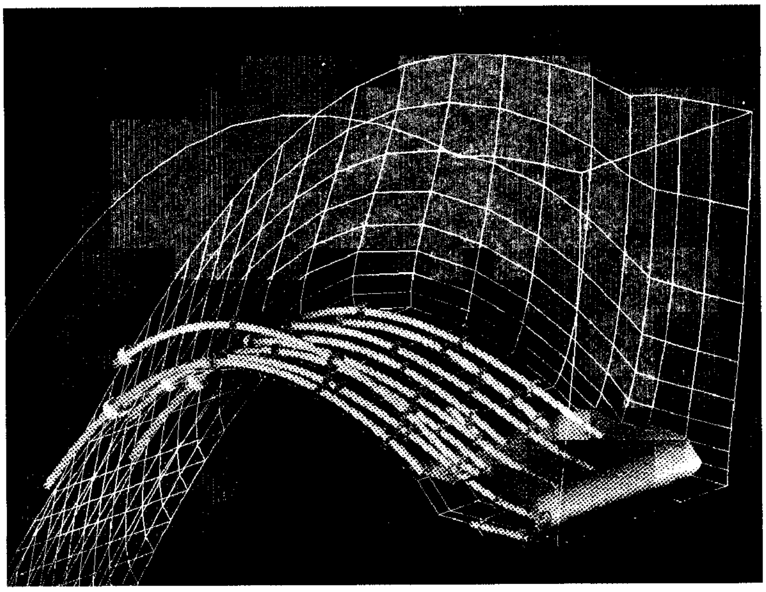

figure 8

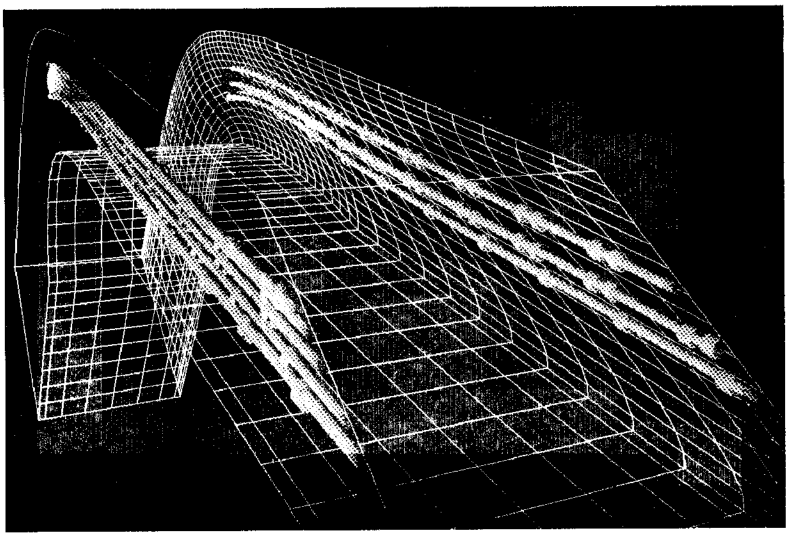

figure 9

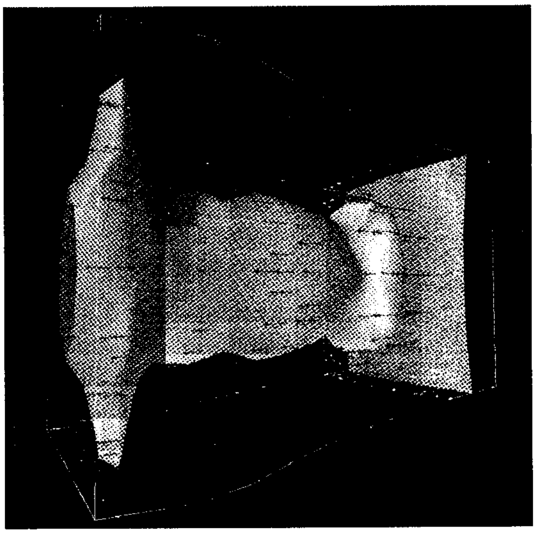

figure 10

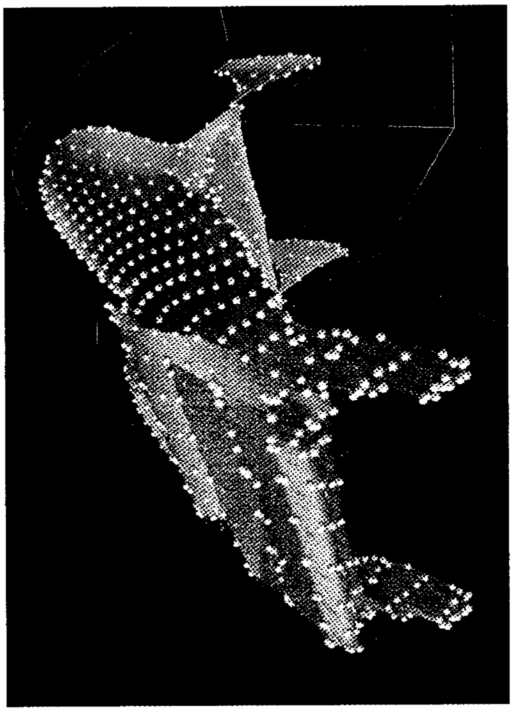

figure 11

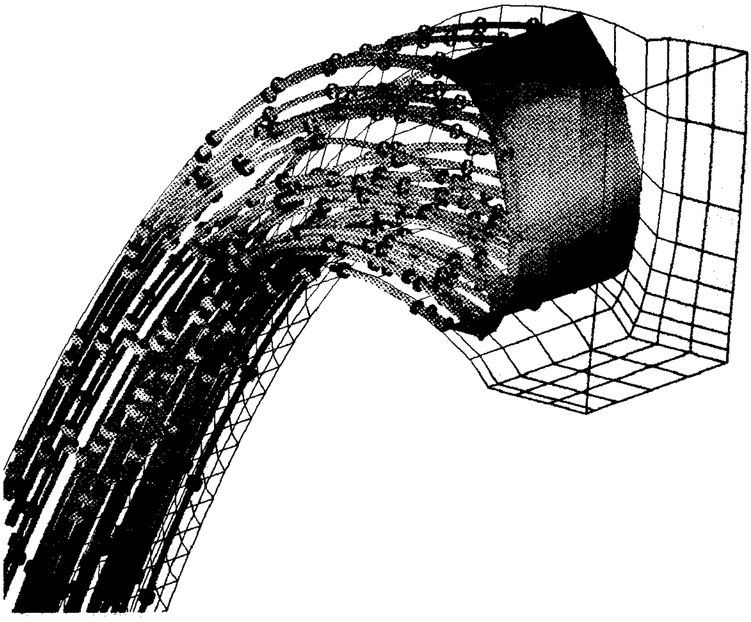

figure 12