

Selective Visualization of Vector Fields

Theo van Walsum, Frits H. Post

Delft University of Technology, Faculty of Technical Mathematics and Informatics,
Julianalaan 132, 2628 BL Delft, The Netherlands
e-mail: {T.vanWalsum|Frits.Post}@twi.tudelft.nl

Abstract

In this paper, we present an approach to selective vector field visualization. This selective visualization approach consists of three stages: selection creation, selection processing and selective visualization mapping. It is described how selected regions, called selections, can be represented and created, how selections can be processed and how they can be used in the visualization mapping. Combination of these techniques with a standard visualization pipeline improves the visualization process and offers new facilities for visualization. Examples of selective visualization of fluid flow datasets are provided.

Key words: scientific visualization, data selection, selective visualization, vector field visualization

1 Introduction

Scientific visualization is concerned with techniques that transform large scientific datasets into images. The goal of visualization is to provide insight in the dataset; the numbers in the dataset must be transformed to useful information. Nowadays visualization is an indispensable tool for data exploration, and a wide variety of techniques have been developed to support, data exploration, mainly by directly visualizing the data.

However, as datasets are growing in size and dimensionality, most techniques tend to be less useful, either because they display too much data (e.g. a 3D arrow plot,) or because they require extensive searching of a large dataset for interesting regions (e.g. streamline visualization: where to start the streamline integration?).

The conceptual solution to this problem is quite simple: reduce the amount of displayed data, and simultaneously increase the information content of the visualization. Some existing techniques, such as slicing techniques, implement the first part of this solution. Other techniques increase the information content, by displaying specific features that appear in the data. In this paper, we propose an approach that integrates both sides of the solution in a general way. By selecting interesting regions in a dataset, and adjusting visualization techniques to incorporate this notion of "interestingness", we are able to extract useful information from large datasets, and to create visualizations that answer questions that are very tedious and difficult, or even impossible, to answer using standard visualization techniques.

In the next section we describe some existing approaches to selective visualization, and discuss the relation with our work. Then, in section 3, we describe our selective visualization approach, and illustrate it with a simple example. In section 4, techniques that are used in the selective visualization process are described in more detail. Section 5 contains implementation details, and shows examples of selective visualization. Conclusions and future research directions follow in section 6.

2 Related work

The most straightforward selective visualization technique for 3D datasets is to cut a (2D) part from the dataset, by using a slicing or cutting plane technique. This technique, however, does not, really solve the problem, it merely shifts the problem to the scientist who has to construct mentally the 3D structure of the data from a series of 2D visualizations.

Another approach is to preprocess the data, in order to detect and allocate certain features (anything that is interesting), and then visualize these features instead of the raw data. An example is the visualization of vector field topologies, as described by Helman and Hesselink (Helman and Hesselink 1991). Other examples are shock wave detection and visualization (Pagendarm and Seitz 1993), vortex tube tracking in turbulent, flow (Villasenor and Vincent 1992) and the visualization of vortex tube evolution in relation to the vortex stretching term (Silver, Gao and Zabusky 1991). The advantage of this approach is that the techniques can be developed for a specific type of feature. This is also the limitation of this approach: the techniques are applicable only for that specific: type of feature.

A more general approach can be thought of. Silver and Zabusky describe a “visiometric” approach to data visualization (Silver and Zabusky 1993). This approach consists of three basic steps: identification, quantification and understanding. First, important features must be identified in a dataset. This can be accomplished by visualization and feature recognition. Then these structures have to be quantified and finally classified, in order to understand the evolution of the observed phenomena.. As they are aiming at understanding the physics in the dataset, and relationships between features, their emphasis is on identification and quantification. Visualization is a means to identify features, and to display the final results.

We have worked on a general selective visualization approach. We primarily aim at, improving the whole visualization process, by adding techniques to create and manipulate selections, and techniques that use selections in the mapping stage. These selections might be analogous to the features Silver and Zabusky try to detect, but this is not, necessarily the case. As we are aiming at, improving the visualization process, we embedded our selection techniques in a visualization pipeline.

3 A selective visualization model

The process of scientific visualization can be described by a generally accepted pipeline model (Haber and McNabb 1990, Post, and Van Walsum 1993), consisting of the stages data generation, data preparation, visualization mapping, rendering and display. After the data have been generated and prepared for visualization, the data are mapped onto visual primitives in the visualization mapping. These visual primitives are then rendered and displayed.

The general process of visualization, as reflected by this visualization pipeline, becomes less adequate as datasets are growing in size and dimensionality. Techniques to select, data, and to focus the visualization on areas of interest should be incorporated in the visualization. Therefore, we introduce the concept of selections, being interesting regions, to the visualization process. The visualization process is extended with techniques that create selections, process selections and use selections in the mapping stage. This section describes the functions of these additions, and illustrates the usage of selective visualization techniques.

The first step in selective visualization is *selection creation*. Selections are created by applying selection criteria to the data set to filter out, interesting regions. In existing visualization systems the most common selection techniques available are cutting (identifying a sub-region of the data space), slicing (showing data in a moving cross-sectional plane), or probing (interrogating information on a certain point or region of interest). The meaning of selection creation is, however, much wider than such spatial selection techniques. The difference with the spatial selection techniques is that the selection is performed automatically by applying a criterion based on content of the data. The resulting selection may be distributed throughout the area and is not constrained to a sub-area with a simple geometry.

Once a selection of data has been created, this selection can be processed, before it, is used in the

visualization mapping. Generally, a transformation is applied which creates a new selection from the original one. We call this type of operation *selection processing*. Techniques that belong to this stage are techniques that are specific for selections, such as clustering, splitting, and filtering selections. The purpose of the techniques in this stage is to enhance the selection, so that the resulting visualization will be better.

The selection that results after creation and processing is then used in the *visualization mapping*. A selected part of the dataset can have different functions in the visualization mapping: it can be used to focus directly on areas of interest, to clip or highlight visualization objects (such as curves and surfaces), and to extract parameters for the visualization mapping

Below we will demonstrate the selective visualization process by visualizing spiralling flow in a backward facing step geometry. The dataset contains a 3D fluid flow on a curvilinear grid and consists of 8325 nodes.

First, we identify the region of spiralling flow. This can be done by using the value of the normalized helicity in the flow. We calculate the normalized helicity H_n with (Buning 1989):

$$H_n = \frac{v \cdot (\nabla \times v)}{|v| |\nabla \times v|}$$

Next, we select those regions where the absolute value of H_n is high. In figure 1a all grid nodes where $|H_n| > 0.6$ are shown. These regions are then processed: isolated nodes are removed, and the selected region is expanded. Figure 1b shows the nodes that are in the selected region after these operations. Then, by generating streamlines through the selected regions, we get the streamlines through the area where the flow is spiralling, see figure 2.

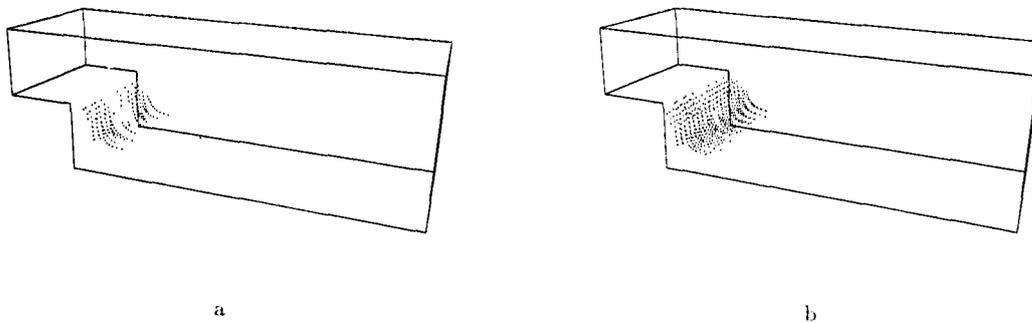


Figure 1. Visualization of selection using dots; a) selection consists of nodes where $|H_n| > .60$; b) selection after removing isolated nodes and expansion

Visualizing the spiralling pattern by manual placement of streamline starting points turned out to be tedious and time consuming. Using the selection criterion described above, generating starting points in the selected area, and tracing streamlines in two directions from these points, the job was quick and easy.

4 Techniques for selective visualization

In the previous section, three stages in selective visualization were mentioned: selection creation, selection processing, and selective visualization mapping. In this section, the purpose of these stages and techniques used in each of these stages are described in more detail.

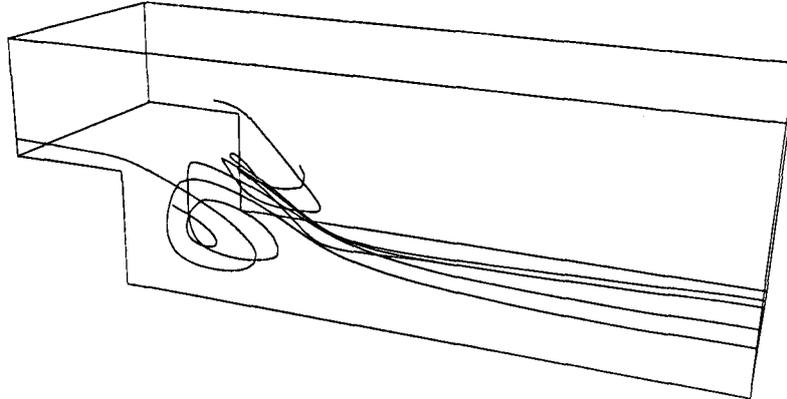


Figure 2. Visualization using streamlines streamlines are generated through the selected region shown in figure 1b

4.1 Selection creation

The first stage in the usage of selections is the creation of a selection: one has to specify which data are interesting. The semantics of a selection, or what it means to say that parts of a data set are interesting, are determined by the application area, the purpose of the research, the phenomena studied, and the personal approach of the user. Our problem here is to provide a general technique that enables users to specify their criteria of interest. Below it will be explained how we represent a selection in a dataset, and the techniques we use to create such a selection, as is described in (Van Walsum 1993).

In our view, selections are those regions in a dataset that are in some way interesting to the user. Therefore, selections can be viewed as interesting parts of it dataset. To represent these interesting parts, one can use all kinds of geometric modelling techniques. It is, however, better to use a representation analogous to the representation of the original data. Therefore we use a set of interesting grid nodes as the representation of interesting parts. Such a set of grid nodes will be represented by a Boolean dataset on the grid.

A selection is a discrete set of grid nodes, and strictly speaking it does not represent a continuous region of a dataset. However, we will consider a grid cell as selected if all (or an arbitrary minimum number) of its vertices belong to the selection. Selected regions then can be obtained by joining adjacent selected cells. Although no assumptions are made about the distribution of selected nodes and cells, in practice selections often occur in a limited number of coherent regions.

As selections are the interesting regions in a dataset, selections can be created by evaluating some interestingness criterion for each part of the dataset. Because selections are represented as Boolean datasets, this reduces to evaluation of this criterion for each grid node. For the criterion, we use a general Boolean expression. For example, the expression for figure 1a. is “`selection = hn > .60`”, where `hn` has been precalculated, using the expression “`hn = dot(v, curl(v))/(norm(v)*norm(curl(v)))`”.

The power of a selection technique that uses such Boolean expressions depends on the elements and operators that can be used in the expressions. Apart from standard comparison and Boolean operations, the following elements can be used in our expressions:

- constants and variables: the basic elements of expressions are scalar and vector constants and variables representing the data, grid and index values at grid nodes;
- arithmetic functions: scalar functions (such as sum and product), vector functions (such as dot product, cross product and vector length) and matrix functions (such as transpose, trace, and product), can be used to manipulate data;

- gradient functions: to calculate local derivatives, gradients, curls and divergences;
- statistical functions: to generate threshold values such as minimum maximum and average.

The functions in the expressions represent three classes of functions: point functions (arithmetic functions), using only values at one grid node, local functions (gradient functions), using values at a grid node, and values at neighbouring grid nodes, and global functions (statistical functions), using values of all nodes in the domain.

4.2 Selection processing

The ultimate goal of the visualization process is to produce images that reveal information that is hidden in the data. To enhance the visualization result, we create selections. As these selections might not be optimal, we add techniques to enhance the selections. This section describes some of these selection processing techniques.

In the previous section it was stated that selections are represented as Boolean data. For datasets where the grid has a regular topology this means that selections can be represented as Boolean arrays. If we restrict ourselves for the moment to 3D datasets, we can say that selections are 3D Boolean arrays. If we also disregard the grid positions and assume a regularly spaced grid, a selection is the 3D analogue of a black-and-white image. Thus, techniques analogous to 2D image processing techniques can be applied to selections.

A first technique that can then be applied on selections is clustering: as selected grid nodes are likely to occur in adjacent cells, they can often be aggregated to a number of coherent clusters. No assumptions are made as to the meaning of these clusters or their shapes. Clustering can be combined with a picking technique to select, one or more clusters from the initial selection, or a filtering technique to remove small clusters.

Other well-known image processing techniques are erosion and dilation. These techniques can be used to shrink respectively expand a cluster. In case of erosion, selected nodes at the border of a selected cluster are set to not selected; in case of dilation, not selected nodes at the border of a selected cluster are set to selected. Algorithms can be found in (Serra 1982).

Erosion and dilation operations can be combined. If one applies an erosion, followed by a dilation, one gets an opening. If one applies a dilation, followed by an erosion, one gets a closing. Openings can be used to remove small details from selections, and to separate two clusters if they are connected by only a few nodes. Closing can be used to fill up holes in selections, and to group two clusters that are close to each other.

At the beginning of this section we said we disregard the grid position information, to yield regularly spaced arrays. The techniques described can be easily applied to such arrays. In practice, however, most numerical flow simulations are performed on curvilinear grids of strongly variable cell size and shape. The effect, of disregarding the grid positions (the techniques are applied in index space (i,j,k-space) instead of in physical space) is that the physical distance between neighbouring grid nodes is not accounted for. For the clustering, this means that two selected nodes that are very close together, yet separated by another node, are not joined, while two nodes that are far more apart, yet not separated, are joined. For erosion (or dilation) this means that the measure of shrinking (or expanding) varies over the grid, and depends on the distance between nodes in physical space.

Despite these consequences, we think that these techniques still can be applied successfully to curvilinear grids. One reason is that normally the grids are designed for specific flow fields; the grid spacing is dense where one expects the complexity of the flow field to be high. It is thus reasonable to use the same resolution for selection processing techniques. Another reason is that these techniques, though necessary, are not critical in their result. They are used to polish a selection, and not to calculate exact results (as is the case in the visualization mapping algorithms). For the opening and closing techniques, there is an additional reason: the sequential application of the complementary erosion and dilation operations greatly reduces the effects of varying distances in physical space. The

net effect of filling holes, or separating clusters, remains. However, as soon as further information is drawn from the shape of the clusters, above mentioned techniques should be refined to account for the variable node spacing.

4.3 Selective visualization mapping

Selective visualization mapping is the stage where raw data and selections are combined to create visual primitives. This section describes what role selections can play in the mapping stage. Below, we will discuss different uses of selections in the mapping stage. Each can be combined with standard visualization techniques for vector fields, such as arrow plots, streamlines and isosurfaces.

The most straightforward way of using selections is *direct visualization* of a selection. In that case we do not need the original data, the selection and grid data is enough. Direct visualization of a selection can reveal information, especially if physical phenomena can be described by selection expressions. An example is the visualization of structures in turbulent flows, as described by Hunt et al. (Hunt, Wray and Moin 1988). They describe criteria to distinguish different zones in a turbulent flow field. Each of these criteria can easily be transformed into a selection expression. By direct visualization of the selection, these zones can be visualized. Visualization of a selection itself is not difficult; one can create small glyphs at each selected grid node, or generate a surface around a selected region.

Apart from this direct visualization, selections can be used to enhance the visualization of raw data. Selections can be used for focusing: for drawing our attention to specific regions, for segmentation or for further exploration of each segment separately. Alternatively, selections can be used for *clipping* and *highlighting* the visual primitives. By showing only interesting parts or primitives, we can avoid uninteresting data to appear in our images, causing cluttering and blocking the view on the essential parts.

Examples are the visualization of arrows at selected nodes, or the visualization of isosurfaces or streamlines in selected areas.

Another usage of selections is to help the *determination of visualization parameters*. All visualization algorithms need parameters: an isosurface needs an isovalue a streamline needs a starting point, etc. In most visualization systems the values for these parameters must be specified manually, which leads to a trial-and-error visualization approach. It is, however, possible to generate (initial guesses for) these values from the dataset using selections, by performing calculations on a selection, or by just taking samples from a selection. A nice illustration of this is picking starting positions for streamlines from a selected region.

5 Implementation and results

The techniques described in the previous sections have been implemented in C as a set of AVS-modules (AVS 1002) on a Silicon Graphics workstation. New AVS-modules were implemented for selection creation and selection processing. To accommodate selective mapping, some existing mapper modules (streamlines, isosurfaces) had to be rewritten. The most important modules follow below.

We implemented selection creation as one module: *cbs* (content based selection). It has two input ports: an AVS-field (data) and a string (an expression). The output port, a Boolean or red field, is filled by evaluating the expression for each node of the data set.

There are several modules for selection processing: *cluster*, *erode*, *dilate*, *open* and *close*. Each of these modules has a selection as input port, and a selection as output port. The modules implement the corresponding techniques described in section 4.2.

For the selective visualization mapping, we added a *select_sample* module. This module takes a number of samples from a selected region. These samples can then be used as input for standard AVS-modules such as streamlines and hedgehogs.

Additionally, we developed a new *streamline* generator, a new *isosurface* generator, an *index space slicer* and a *selectiota visualizer*. The streamlines can be generated inside, outside or through a selected region, thus implementing the clipping and the parameter extraction function of selections. The streamline algorithm uses a second order Runge Kutta integration, and operates in physical space.

The isosurface algorithm generates a surface that is clipped against, a selected region. For the algorithm we use a tetrahedrization of the grid cells. In this way we could avoid some problems occurring in the hexahedral version of the Marching Cubes algorithm (Lorensen and Cline 1987), at the expense of approximately twice as many triangles. The isosurface generator implements the clipping function of a selection.

The index space slicer uses clustering information. For each cluster, the centroid (in index space) is calculated. This cluster centre is then used to position three orthogonal cutting planes through the clusters. The index space slicer thus implements both clipping and parameter extraction.

The selection visualizer generates dots or crosses at the nodes of the selection. These dots or crosses can be coloured with the value of a scalar variable, to visualize a scalar variable in a selection.

The above described modules were used to visualize a direct simulation of a pipe flow. At the mechanical engineering department of the Delft, University of Technology, drag reduction in turbulent flow by polymer addition is studied (Den Toonder, Nieuwstadt and Kuiken 1994). Criteria are being developed to specify regions with large magnitude of strain and elongational type of flow, because these are the regions in which the polymer additives are assumed to be stretched by the flow, and become active in influencing the flow, leading globally to drag reduction. We used selective visualization to compare such a criterion with two of its components.

The original flow simulation is done on a Cray, and consists of several time steps of a flow in a pipe on a 3D cylindrical grid of more than three million nodes. For our visualization, we cut a region of approximately fourteen thousand nodes containing velocity data from one time step. Before the data could be visualized, we had to calculate the following derived data values (where u is the velocity, $\text{tr}(M)$ is the trace and $\det(M)$ is the determinant of matrix M):

$$\begin{aligned} L &= \nabla u \\ D &= \frac{1}{2} (L + L^T) \\ II_D &= \frac{1}{2} \text{tr}(D \cdot D) \\ III_D &= \det(D) \\ R &= \frac{3III_D}{II_D} \\ R_2 &= \frac{3III_D}{II_D^{\frac{3}{2}}} \end{aligned}$$

II_D contains information on the magnitude of strain, R_2 contains information on the type of flow and R , called the "elongation parameter", contains mixed information on both magnitude of strain and type of flow. The goal of the visualization was to check whether R can be used as a criterion for the relevant regions, or whether both II_D and R_2 should be used separately.

To visualize the relationship between II_D and R_2 , we selected a region with elongational type of flow ($|R_2| > 1$). The nodes in this region were coloured with the value of II_D , where we used a rainbow-like colourmap, with blue for low values of II_D and red for high values of II_D . The resulting image is shown in figure 3.*

To compare this result with the value for R , we added an isosurface at high values for R , i.e. $|R| = 80$. The result is shown in figure 4.*

The regions with high magnitude of strain and elongational type of flow are the regions where the selected nodes are red. From the images it is clear that regions with high value for R include these

* See page C-539 for Figures 3 and 4.

regions. To validate this result, we also selected those regions where R is small, and both R_2 and II_D are large, i.e. those regions where $|R| < 80$ and $|R_2| > 1$ and $II_D > 7500$. This selection was empty, there were no nodes that satisfied the condition.

Another view is presented in figure 5.* There a cluster is chosen from the selected nodes where $|R| > 80$. In this cluster, R_2 is visualized using coloured crosses at the nodes, and II_D is visualized using three slices through the cluster. The crosses and slices are coloured using a standard rainbow-like colourmap.

From the visualizations, it was concluded that R is a reasonable criterion to identify regions with large stretching, and may be used in computations instead of R_2 and II_D separately. This considerably simplifies the fluid flow computations.

6 Conclusions and future research

The selective visualization approach presented improves both the efficiency and the effectiveness of the visualization process. The use of selections allows the user to create images that contain only those data that are relevant, or interesting; the image thus contains less data and more information and therefore is more effective. The efficiency of the visualization process is increased, because selective visualization allows users to explicitly search for certain features in the dataset, thereby avoiding the tedious process of manually probing a dataset.

Apart from these two effects, selective visualization increases the class of problems that can be solved by visualization. Conditional visualizations, like the ones presented in the examples, are not possible using standard visualization techniques.

In our future research, we will investigate other roles of selections than those described in this paper: instead of using information from the selections to enhance the visualization mapping, we will try to visualize selections, and information from selections (such as aggregate data values) directly, by mapping this information onto icons. Such icons display information in a way that is rather different, from common presentation techniques in scientific visualization.

Acknowledgements

The data for the flow in a backward facing step geometry is courtesy Technical Mathematics Department of Delft, University of Technology. The data for the pipe flow is courtesy Mechanical Engineering Department, of Delft University of Technology. We are much indebted to Jack van Wijk for his critical comments and Jaap den Toonder for his cooperation in the pipe flow example. We also thank Erik Jansen, Guus Segal and Deborah Silver for their comments on earlier versions of this paper.

References

- Advanced Visual Systems Inc. (1992), *AVS User's Guide, Release 4*.
- Buning, P. G. (1989), Numerical algorithms in CFD post-processing in 'Lecture Series 1989-07', von Karman Institute for Fluid Dynamics.
- Haber, R. B. and McNabb, D. A. (1990), Visualization idioms: A conceptual model for scientific visualization systems, in G. A.I. Nielson, B. Shriver and L. J. Rosenblum, eds, 'Visualization in Scientific Computing', IEEE Computer Society Press, pp. 74-92.
- Helman, J. L. and Hesslink, L. (1991), 'Visualizing vector field topology in fluid flows' *IEEE Computer Graphics and Applications* 11(3), 36-46.

* See page C-539 for Figure 5.

- Hunt, J. C. R. , Wray, A. A. and Moin, P. (1988), Eddies, streams and convergences zones in turbulent flows, in 'Center for Turbulence Research Proceedings of Summer Program' , pp. 193-208.
- Lorensen, W. E. and Cline, H. E. (1987), 'Marching cubes: a high resolution 3D surface construction algorithm', *Computer Graphics* 21(4), 163-169.
- Pagendarm, H.-G. and Seitz, B. (1993), An algorithm for detection and visualization of discontinuities in scientific data fields applied to flow data with shock waves, in P. Palamidese, ed., 'Scientific Visualization: Advanced Software Techniques', Ellis Horwood Limited, pp. 161-177.
- Post, F. H. and Walsum, T. van (1993), Fluid flow visualization, in H. Hagen, H. Müller and G. M. Nielson, eds, 'Focus on scientific visualization', Springer-Verlag, pp. 1-40.
- Serra, J. (1982), *Image analysis and mathematical morphology* Academic Press.
- Silver, D. , Gao, M. and Zabusky, N. (1991), Visualizing causal effects in 4D space-time vector fields, in G. M. Nielson and L. Rosenblum, eds, 'Proceedings Visualization '91', IEEE Computer Society Press, pp. 12-16.
- Silver, D. and Zabusky, N. J. (1993), 'Quantifying visualizations for reduced modeling in nonlinear science: Extracting structures from data sets', *Journal of Visual Communication and Image Representation* 4(1), 44-61.
- Toonder, J. M. J. den , Nieuwstadt, F. T. M. and Kuiken, G. D. C. (1994), 'The role of elongational viscosity in the mechanism of drag reduction by polymer additives', *Submitted to Applied Scientific Research*.
- Villasenor, J . and Vincent, A. (1992) , 'An algorithm for space recognition and time tracking of vorticity tubes in turbulence', *CVGIP: Image Understanding* 55(1) , 27-35.
- Walsum, T. van (1993), 'Content-based grid node selection for vector field visualization', Presented at, the 4th Eurographics Workshop on Visualization in Scientific Computing, Abingdon, UK.