Available online at www.sciencedirect.com

SCIENCE DIRECT

Computers & Graphics 28 (2004) 311–325

COMPUTERS & GRAPHICS

www.elsevier.com/locate/cag

Gaining understanding of multivariate and multidimensional data through visualization

Selan dos Santos, Ken Brodlie*

School of Computing, University of Leeds, Leeds LS2 9JT, UK

Abstract

High dimensionality is a major challenge for data visualization. Parameter optimization problems require an understanding of the behaviour of the objective function in the n-dimensional space around the optimum—this is multidimensional visualization and is the traditional domain of scientific visualization. Large data tables require us to understand the relationship between attributes in the table—this is multivariate visualization and is an important aspect of information visualization. Common to both types of 'high-dimensional' visualization is a need to reduce the dimensionality for display. In this paper we present a uniform approach to the filtering of both multidimensional and multivariate data, to allow extraction of data subject to constraints on their position or value within an n-dimensional window, and on choice of dimensions for display. A simple example of understanding the trajectory of solutions from an optimization algorithm is given—this involves a combination of multidimensional and multivariate data.

Keywords: Visualization; Multidimensional; Multivariate; Reference model

1. Introduction

One of the major challenges for visualization is to find effective ways of presenting high-dimensional data, so that insight and knowledge can be gained. It is not an easy problem—the visualization process must somehow convert the high-dimensional data to low-dimensional geometry for display. In this paper, we explore how this dimension reduction can be achieved.

First, however, it is important to define our terminology, since words such as 'dimensionality' are overused in visualization. We shall follow the review paper of [1], and talk in terms of *multidimensional* and *multivariate*. We shall think of an item of data as a sample from a k-variate function $F(X)$ defined over an n-dimensional domain D. Thus $F = (f_1, f_2, \ldots, f_k)$ has k components, and $X = (x_1, x_2, \ldots, x_n)$ is a point in D. We shall allow k to be zero, in which case we just have a point in D, and we allow n to be zero in which case we just have a value of F. We shall talk in terms of *dependent* variables F and *independent* variables X. Statisticians use the corresponding terms *response* variables and *predictor* variables (see, for example, [2, p. 233]).

In this paper, we begin by looking in slightly more detail at examples of multidimensional data—traditionally associated with scientific visualization—and at examples of multivariate data—more associated perhaps with information visualization; and at cases that involve both multidimensional and multivariate data. In all these examples, we restrict attention to numeric, real-valued data—of course this is only a subset (but an important one) of the types of data considered within information visualization particularly. We then revisit the well-established reference model for scientific visualization, extending it in a way that supports high numbers of dimensions and variates, and showing it has some relevance, therefore, to information visualization also. Our aim is to bring the two fields of scientific and information visualization rather closer together, by providing this common framework for numeric data.

From this reference model, we identify a key filtering process which reduces the complexity of the problem. This process involves a pair of operations: one is the definition of a window of interest (in the

*Corresponding author. Tel.: +113-343-5484; fax: +113-343-5468.

E-mail address: kwb@comp.leeds.ac.uk (K. Brodlie).

0097-8493/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2004.03.013

multidimensional case this is a window defining the domain of interest within the n-dimensional space, and in the multivariate case it is a region defining the range of interest of the variates); the other operation either reduces the dimensionality (for multidimensional data) or reduces the number of variates (for multivariate data). We then describe tools which implement these two operations, and present an example where we have found the filter useful. This example is an optimization problem in which the objective function has many parameters, or dimensions, and in which the solution trajectory can be considered as multivariate (each point being an observation, each coordinate of a point being a variable). Thus one example allows us to explore both multidimensional and multivariate data.

2. Multidimensional and multivariate data

2.1. Multidimensional data—scientific visualization

Scientific visualization commonly deals with multidimensional visualization. Usually the visualization is concerned with sample data which is given at specified points within the n-dimensional domain D, and the goal is to recreate from this sampled data an estimate of the underlying entity, $F(X)$, over the entire domain. Interpolation is a key part of this process. In mathematical modelling applications, the model itself may be provided to us, as an approximation to some physical phenomenon that is being investigated. Corresponding data sets may be generated during a pre-processing step by evaluating the model at a set of points in the n-dimensional domain D.

Often the number of dimensions is small—from simple 1D applications such as temperature measured at different times, to 3D applications such as medical imaging, where data is captured within a volume. Standard techniques—contouring in 2D; isosurfacing and volume rendering in 3D—have emerged over the years to handle this sort of data. There is no dimension reduction issue in these applications, since the data and display dimensions essentially match.

Increasingly, however, scientific visualization needs to concern itself with higher-dimensionality problems, such as occur in parameter optimization problems, where we wish to visualize the value of an objective function $F = (f_1)$ in terms of a large number of control parameters, $X = (x_1, x_2, \ldots, x_n)$, say. This is a much harder problem and relatively unexplored.

One suggestion, the Hyperslice method from van Liere and van Wijk [3,4], is to look at all 2D orthogonal subspaces of X, and present a grid of contour maps. Each of these subspaces is a slice of the original data obtained by fixing the value of $(n-2)$ parameters, and varying the remaining two parameters within a specified region. Thus we reduce from one n-dimensional space to m 2D spaces, where $m = n(n-1)/2$. In fact the subspace visualizations are laid out in a symmetric $n \times n$ grid, with the diagonal showing n 1D visualizations, where only one parameter varies (so a line graph is drawn).

2.2. Multivariate data—information visualization

Information visualization commonly deals with multivariate data from application areas such as statistical analysis, stock markets, or earth sciences. In many applications, the data is given in the form of a data table, where each column represents an attribute, and each row represents an observation of these attributes. There are no independent variables here, so we can view n as zero, and see the data as an unordered set of k-tuples with S elements, $F^i = (f^i_1, f^i_2, \ldots, f^i_k), i = 1, 2, \ldots, S$. In fact it is possible to make an alternative, geometric interpretation, and think of these as S points in a k-dimensional space.[1] As mentioned earlier, we restrict our attention in this paper to elements which are numeric and real-valued.

The goal of the visualization, determined by the context of the problem, usually involves the searching for patterns, structure (clusters), trends, behaviour, or correlation among attributes. The resulting information is then fed into the exploratory stage of the knowledge-acquiring process to support the elaboration of hypotheses about the phenomenon responsible for the targeted data.

The number of variates is typically quite large, and so in this field the study of reducing the number of variates (or dimension reduction in the geometric interpretation) is well developed. There are two main approaches. In the first approach, the data is reduced to 2D or 3D by projection—in these low dimensions, a scatter plot can be used. The use of multiple views allows a set of these scatter plots to be produced, in a matrix form similar to the Hyperslice method mentioned above for multidimensional data—see, for example, [5].

The second approach is to use some technique in which a large number of variates can be presented in a low-dimensional display format. Well-known techniques of this form include re-arranging axes to be non-orthogonal (parallel co-ordinates [6], glyphs [7]/icons [8], and Andrews' plots [9]); hierarchical approaches (worlds within worlds [10] or hierarchical axis [11]); or screen-based techniques (pixel-oriented techniques [12], and natural texture mapping [13,14]). These methods all succeed to an extent in presenting large numbers of variates in a single display, but there are eventual limits to what can be handled.

[1] This contributes to the ambiguity of the term 'dimension' in visualization.

2.3. Multidimensional and multivariate data

There are further applications which are both multivariate and multidimensional. For example, in medical imaging we may wish to look at co-registered CT and MR data: here we have two variates defined over a 3D domain. The numbers of variates and dimensions are small in this case, and so it is possible to solve this particular application by extension of existing methods, for example, combining the two variates in some way within the volume rendering process.

Another application is in optimization where in addition to many parameters, there may be several different criteria—thus again, several variates and several dimensions. For discussion of optimization problems with multicriteria objective functions, see for example [15].

Another example from optimization, that we shall pursue later in the paper, is the visualization of trajectories of intermediate estimates of the solution point, as generated by an iterative algorithm. The algorithm generates a sequence of points $\{Y^i\}, i = 1, 2, \ldots, S$, towards the minimum of a function $Q(Y)$, of k variables $y_1, y_2, \ldots, y_k$. The points can be regarded as S items of multivariate data with k attributes, but they are ordered in sequence. However, we also know the value of the objective function Q at each point: visualization of the function is a multidimensional visualization problem. Later in this paper, we explore how to visualize both trajectory and function, as a multivariate and multidimensional visualization problem.

3. Reference model for multivariate, multidimensional visualization

3.1. Haber and McNabb reference model

Reference models are useful in giving us a high-level view of the processes involved in visualization. Early work in scientific visualization benefited from the clarity of thinking in 1990 which underpinned the reference model of Haber and McNabb [16]. This expressed the visualization process as a sequence of steps: *data enrichment*, to prepare the data for visualization; *mapping*, to convert from numerical data to an abstract geometrical representation; and *rendering*, to create an image from the geometry.

This is illustrated in Fig. 1. This model has formed the basis for many scientific visualization systems, such as IRIS Explorer [17] and Open Visualization Data Explorer [18], and toolkits such as VTK [19].

The model was essentially designed for the core scientific visualization applications, involving scalar and vector field data defined over 2D and 3D. In this section we revisit this model: we elaborate the data enrichment step so that it can better describe higher-dimensional visualization problems; and we then show how this same model can effectively describe the multivariate problems of information visualization, and indeed the multivariate, multidimensional problems described in Section 2.3 above.

3.2. Extending the model for multidimensional data visualization

We begin with the case of multidimensional data, that is, data sampled from a function $F(X)$, where $X = (x_1, x_2, \ldots, x_n)$. The visualization mapping and rendering processes are now well understood, but rather less attention has been paid to the data enhancement process. The original intent was that it should be an interpolation process, for example, generating a regular grid of data from a given set of scattered data. In reality, it has often been interpreted as a filtering type process, to select data of interest from a larger initial set.

In our extended model, we replace the data enhancement process with two separate processes: 'Data Analysis' and 'Filtering'. In the data analysis step, the raw data would have associated with it an interpolation function, with the ability to recreate throughout the domain, an estimate of the underlying entity being visualized. One can view this interpolation function

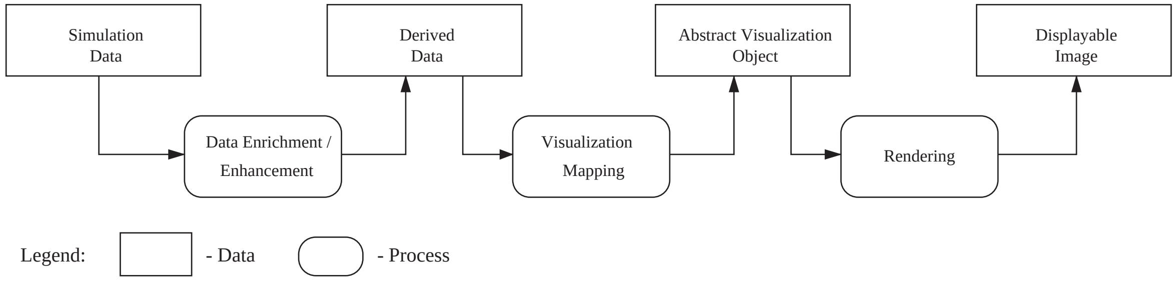

Fig. 1. Haber–McNabb dataflow model for scientific visualization.

being tagged to the data as it passes along the pipeline. The data analysis step can be seen as a pre-processing step—it is possible to return to alter the interpolation, but this is the exception rather than the rule. Since there is little interaction with the user, one can see this as a 'computer-centred' operation.

In the filtering step, we extract the portion of the data we wish to visualize. This involves placing bounds on the domain D. We have found it convenient to see this as a pair of distinct operations: the definition of an n-dimensional window with upper and lower bounds, and n-dimensional focus point within these bounds; together with a constraint term which controls the parameter values within the window—for example, we can reduce the dimension by fixing certain parameters at their focus point values. Thus a slice operation would be seen as both defining a window of interest, and also applying a constraint to specify the slice through the window. The interpolation function created in the data analysis step is used to provide the values of the function on the slice. In contrast with the data analysis step, the filtering process is interactive—the user will typically apply a number of filters in a particular session. Thus filtering can be seen as 'human-centred'.

The extended reference model is shown in Fig. 2. Again we have an overall view as a dataflow pipeline in which one process receives data, operates on it, and passes on the result to another process.

We start with our *Problem Data*. The data passes first through the data analysis step, being converted to *Visualization Data*—i.e., data plus interpolant to allow us to visualize. This passes to the filtering step, which extracts the *Focus Data*.

The third and fourth steps correspond to the mapping and rendering processes of the original Haber–McNabb model. The mapping step takes the Focus Data and creates some geometrical representation, thus generating *Geometry Data*. The rendering step creates *Image Data* for display on a monitor.

3.3. Model for multivariate data visualization

We now revisit this model from a multivariate data viewpoint. Encouragingly, we find that it describes this case quite effectively. The problem data now consists of raw multivariate data $F^i = (f_1^i, f_2^i, \ldots, f_k^i), i = 1, 2, \ldots, S$. The data analysis step is again computer centred and consists of some analysis technique. Two popular ones are Principle Component Analysis, PCA, which projects the data into a lower-dimensional—i.e., lower number of variates—subspace that accounts for most of the variance in the data [20], and MultiDimensional Scaling, (MDS), which uses nonlinear optimization to lay out the observations in a lower-dimensional subspace in such a way that their separation corresponds as closely as possible to their separation in the original higher-dimensional space [21]. Although these techniques are not general means for clustering their outcome can sometimes be useful in identifying clusters and trends in the data.

Both PCA and MDS have the disadvantage, however, that the original set of variates are no longer retained. That is, the data analysis step produces Visualization Data whose variates are not easily interpreted in terms of the variates of the Problem Data. Moreover, in extreme cases clusters could be lost by the dimension reduction process. As an alternative approach, aiming to retain the original variates, Yang et al. [22] proposed the Visual Hierarchical Dimension Reduction (VHDR) approach. Here the variates are placed into clusters and a representative variate is selected (either the 'centre' dimension of the cluster, or a new variate which is an average of those in the cluster). This reduces the complexity of the final display, without destroying the meaning of the variates.

The filtering step takes the multivariate Visualization Data, however produced, and applies a very similar operation to filtering in the multidimensional case. Again we can see the filter as a pair of operations. We

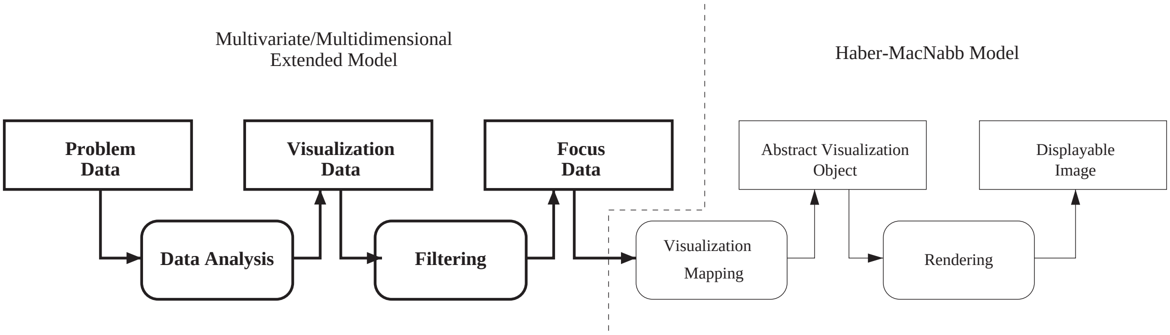

Fig. 2. Extended dataflow model to accommodate multivariate multidimensional visualization. The darker blocks on the left-hand side of the dashed line replace the first three components of the original model (see Fig. 1).

define a window in the value space of the k variates, which we can as before interpret geometrically as a k-dimensional region. This specifies the bounds of interest on the values of the variates. In addition we apply constraints, which in this case is a selection from the k variates (similar to the multidimensional case where we used constraints to identify parameters of interest). In multivariate data visualization, this filtering step of identifying data of interest is often called *brushing* [23].

The resulting Focus Data then passes to the Mapping step, which applies a suitable technique for multivariate visualization such as those described in Section 2.2 above. Note that in the case of projection techniques, such as scatter plot matrices, we can see these as requiring (for each scatter plot) a filter which extracts a given two variates from the set of k. For the other methods, such as parallel coordinates, a filter may not be required—although even these methods can sometimes benefit from a reduction in the number of variates. The final Rendering step is as before.

For data which is both multidimensional and multivariate, we can use exactly the same model. The filtering step now applies a filter first to the multidimensional aspect of the data, and then to the multivariate aspect, using the approaches described above. Indeed the filters can be applied in either order. Please refer to Table 1 for a summary of how these two operations relate to multidimensional and multivariate data.

Thus the reference model of Fig. 2 can provide a high-level view of the visualization process for multidimensional and multivariate data, and thus helps us to see scientific visualization and (at least part of) information visualization in a common framework. It is worth noting at this point the Data State Reference Model of Chi [24]—this builds on earlier work to develop a taxonomy of information visualization. This similarly suggests a pipeline of processes, as in the Haber and McNabb model, but the transformation steps are slightly different from those we are suggesting here, being driven mainly by information visualization, and targeted at a wider class of data.

Table 1
Listing some techniques associated with the Data analysis & filtering steps for multidimensional and multivariate cases

Data type	Data analysis	Filtering
Multidimensional	Interpolation	Window on domain D, Selection of dimensions
Multivariate	PCA, MDS, VHDR	Window on variate space, Selection of variates

4. The filter process for multidimensional and multivariate data

4.1. Filtering multidimensional data

To recap, the filter process for multidimensional data defines a window in n-dimensional space, defines a focus point within the window, and applies a constraint—in our work here, this constraint identifies those parameters which are to be fixed at their focus point values and those parameters which are allowed to vary within the window.

We have found that this functionality can be achieved using a set of three tools: one defines the window, the second specifies the dimensions, and the third extracts the specified data from the original Visualization Data, outputting results as Focus Data. A schematic of this is shown in Fig. 3. (The Visualization Data can be regarded as datapoints in n-dimensional space, together with an interpolation function capable of returning a value of the function at any point in the space.) We have implemented the tools as modules in IRIS Explorer [17].

The window definition tool is called an *n-dimensional Window* and we show its user interface in Fig. 4. From its input data, it recognises the number of dimensions, and lays these out as vertices of an n-sided polygon as shown. Each spoke from centre to a vertex acts as a means of specifying the extent of the domain, and the focus point, for that dimension. In the figure, the end-points of the domain are shown as cyan and red circles and the focus point within that domain is marked as yellow. By moving the circles along the spoke, the user can apply different bounds, and define different focus points. Changing these will generate different Focus Data. In the left image, the bounds of the window are the full extent of the data, but in the right image, the user has defined subranges in three of the four dimensions.

The dimension specification tool is called an *Interaction Graph* and we show its user interface in Fig. 5. Again from the input data, the number of dimensions are recognised, and these are laid out as vertices of a polygon, maintaining the metaphor of the *n-dimensional Window* definition tool. The overall appearance of the tool resembles that of a 2D fully connected graph in which a single vertex defines a 1D space, an edge defines a 2D subspace, a triangle corresponds to a 3D subspace, and so on up to the whole space (represented by a polygon connecting all the vertices). Note that all the possible subspaces are encoded in this representation. Hence, the tool may be regarded as a visual retrieval tool in the sense that it allows the user to create and manipulate as many subspaces as necessary to form a mental model of the complex data. A similar approach in the information visualization field called *InfoCrystal* was proposed in [25] to visualize and query all the possible relationships among N concepts.

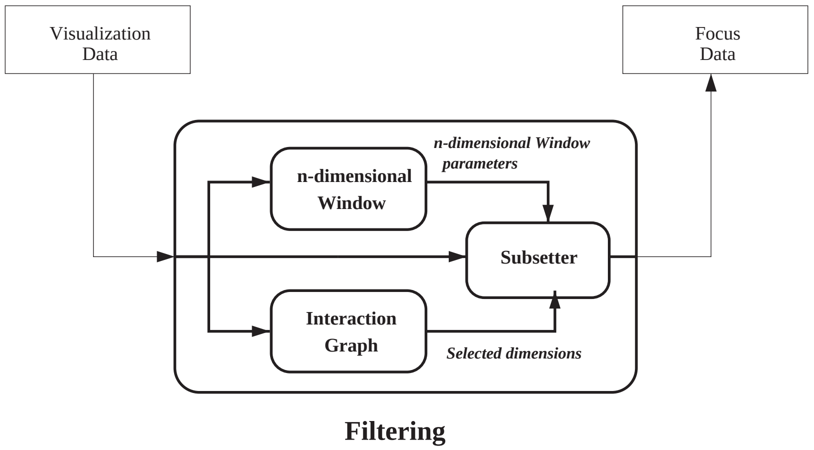

Fig. 3. Acquiring the focus data from the visualization data by applying a filtering process. The filtering process is expanded to show its three component operations: window definition (performed by the *n-dimensional Window* module), dimension specification (performed by the *Interaction Graph* module), and extracting the corresponding subset of the Visualization Data (performed by the *Subsetter* module).

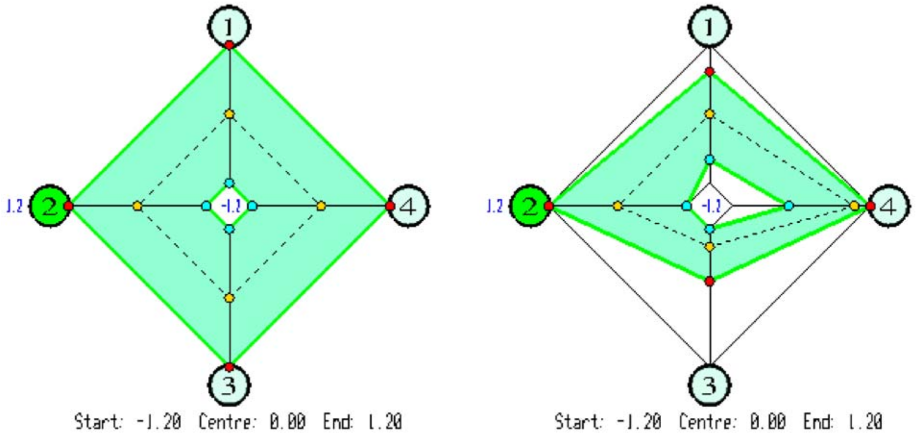
1
1.2
2
-1.2
4
3
Start: -1.20 Centre: 0.00 End: 1.20
1
1.2
2
-1.2
4
3
Start: -1.20 Centre: 0.00 End: 1.20

Fig. 4. User interface for the *n-dimensional Window* definition tool for a 4-dimensional case. The picture on the left shows the starting configuration for the tool, having the range for each dimension (numbered from 1 to 4) set to cover the whole data set. The picture on the right shows the same tool after some interaction has been done, in particular on the ranges of dimensions 1, 3 and 4. Also notice that the focus point (represented by a dashed polyline connecting the yellow circles inside the green region) has also been changed to a different position in the 4-dimensional space. The text at the bottom of each picture shows three numeric values related to the dimension 2, the currently selected dimension (which is indicated by a darker background on the dimension indicator circle): *Start*, the lower limit for the extent of the domain in that dimension; *Centre*, the current value for that component of the *n*-dimensional focus point, and *End*, the upper limit for the extent of the domain in that dimension. By dragging the corresponding circles the user can change those values.

A dimension is selected by mouse click; in the top pictures of the Fig. 5, dimension 1 has been selected. This allows parameter 1 to vary within its bounds, while the other parameters, all unselected at present, remain fixed at their focus point values. Thus the output will be effectively a 1D line graph. A further selection will open the filter to a second parameter, giving a 2D field that can be visualized using a contour map or surface view, as shown in the middle pictures. A third selection will give a 3D field that could be isosurfaced, or volume rendered, as shown in the bottom pictures. Notice that lines joining selected vertices are thickened, and the vertices are highlighted.

The behaviour of the filter has a degree of continuity in the following sense. If we have a 3D field, but toggle off one of the parameters, we create a 2D field which is a

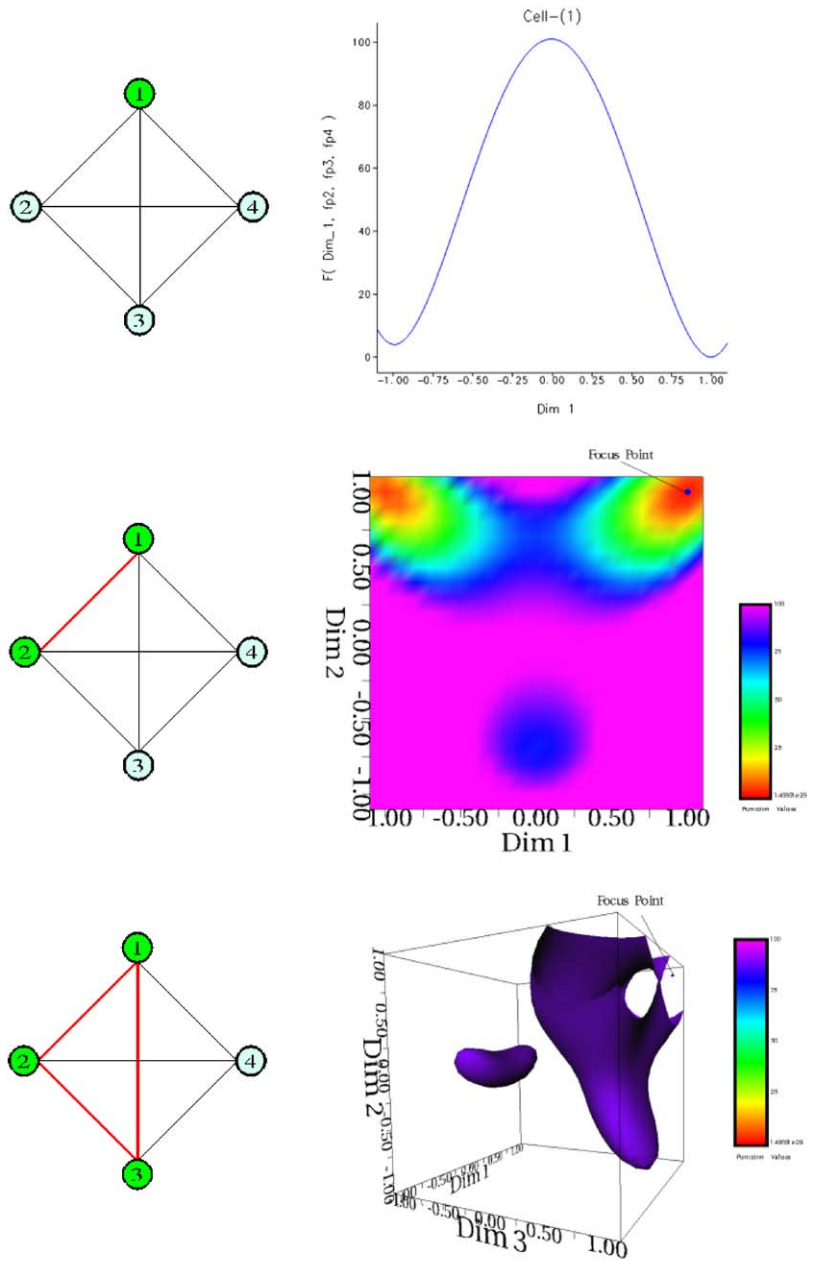

Fig. 5. User interface for the *Interaction Graph* tool along with several visualizations. The sequence of pictures depicts the action of progressively investigating the 4D space. The left column contains a sequence of *Interaction Graphs* at different stages of the investigation process. The right column contains the corresponding data visualizations. The top row shows the *Interaction Graph* with only one dimension selected (dimension 1) and a line graph corresponding to the visualization of the data having that dimension free to vary over its range and the other three dimensions fixed to the values of the focus point. The middle row indicates that the user has selected a second dimension (dimension 2) and the visualization has been changed from the 1D line-graph to a 2D coloured field. In this case the dimensions 1 and 2 are free and dimensions 3 and 4 are the values of the focus point. The bottom row shows that currently 3 dimensions have been selected and a typical 3D visualization method, namely isosurfacing, has been applied to the 3D output data.

slice through the earlier 3D space, at the focus point value of the toggled parameter—the interaction graph is shown in the top images of Fig. 6. Selecting now a fourth parameter, we move into a new 3D space which contains that 2D field as a slice—see the lower images. An example in the next section should make this clearer.

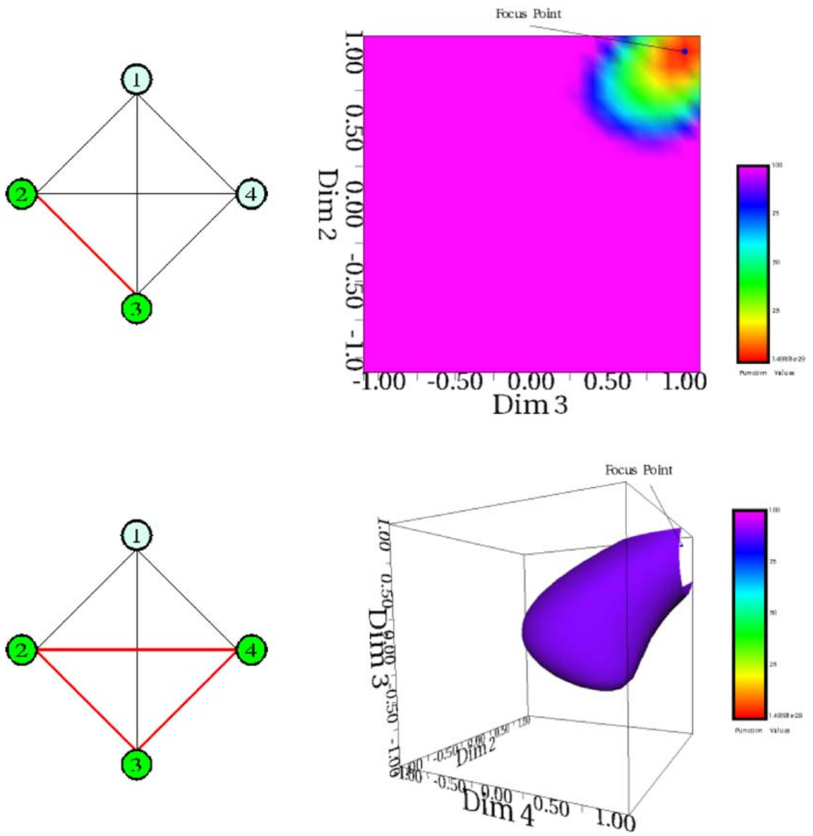

Fig. 6. User interface for the *Interaction Graph* tool along with several visualizations. The sequence of pictures depicts a further investigation of the 4D space around the focus point, started in Fig. 5. The top row corresponds to the *Interaction Graph* after the dimension 1 has been deselected and a corresponding 2D coloured field generated. The bottom row shows the *Interaction Graph* after the dimension 4 has been selected, which returns the output to a 3D visualization but now of a different 3D subspace (2-3-4-space).

The final module, the *Subsetter*, simply takes the description of the filter given by the *n-dimensional Window* and *Interaction Graph*, and extracts the corresponding Focus Data. In terms of IRIS Explorer, the modules receive lattice data of any dimension as input. The resulting output is a 1D, 2D or 3D lattice for input to a corresponding mapping module in IRIS Explorer. Indeed we can generate 4D data by selecting a further parameter as time dimension, and animating through a set of 3D spaces, the fourth parameter stepping through its range of values. An earlier version of the *Interaction Graph* was described in [26].

By implementing the tools as IRIS Explorer modules, we gain access to the data analysis, visualization mapping and rendering facilities already developed for that system.

4.2. Filtering multivariate data

Exactly the same interface can be used to filter multivariate data. The *n-dimensional Window* now acts to restrict the range of values of the variates. The *Interaction Graph* selects the variates of interest as in the multidimensional case by clicking on the variate numbers at the vertices of the polygon. Thus we can select a 2D projection for display as a scatter plot, multiple 2D projections for a matrix of scatter plots, or a 3D projection for display as a 3D scatter plot. Again one can traverse smoothly between different projections. An example of this is given in the next section. The *Subsetter* operation then extracts the Focus Data for input to the next stage of the pipeline.

Thus Filtering is applied in a consistent way to both multidimensional and multivariate data.

5. Applications of the filter process for multidimensional and multivariate data

5.1. Exploring a multidimensional function

In order to demonstrate the filter tool, we illustrate its use to explore a well known function of four variables from the optimization world, the chained Rosenbrock function [27]. This is a generalisation of the original Rosenbrock function (the case $n = 2$), which has an interesting banana-shaped valley—the shape of the function is shown in Fig. 7. The four-dimensional generalisation, however, is much harder to envisage.

The function is defined by the following expression:

$$F(x) = \sum_{i=2}^{n} [100(x_{i-1}^2 - x_i)^2 + (x_{i-1} - 1)^2].$$

We choose a four-dimensional example for ease of presentation, but the idea scales up to higher dimensions. Indeed in the concluding chapter we extend to the six-dimensional case.

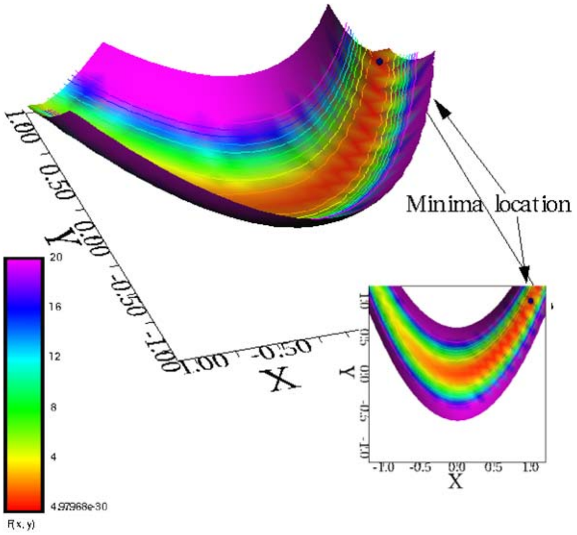

Fig. 7. The main picture shows a 3D view of the famous Rosenbrock's Banana-shaped Valley having its height and colour associated with the function value. The small picture on the bottom-left corner is a top view visualization of the same function. The *minimum* location is depicted by a black dot indicated by the arrows.

It is easy to determine the minimum point of (1, 1, 1, 1), almost by inspection, with corresponding minimum value of zero—but what is the behaviour of the function near the minimum? This is the sort of sensitivity analysis question that is increasingly important in optimization problems.

In the following sequence of pictures, the *Interaction Graph* has been used to study the behaviour of the 4D function near the minimum point (1, 1, 1, 1). An *n-dimensional Window* specification has been applied to restrict data to a region near the minimum, and specify the focus point as (1, 1, 1, 1). An exploration sequence is generated by selecting different dimensions with the *Interaction Graph*, using the sequence described in the previous section in Figs. 5 and 6. The IRIS Explorer dataflow pipeline, or map, is shown in Fig. 8, reflecting the structure of our reference model.

The corresponding visualizations are shown in Fig. 9. The top left image shows a graph with parameter 1 allowed to vary; next we allow parameter 2 to vary and display as a contour plot—see top right image and notice the low values in the neighbourhood of $(-1.0, 1.0)$ and $(1.0, 1.0)$ in these two dimensions, with an indication also of a low area around $(0.0, -0.6)$. (Parameters 3 and 4 are fixed at their focus values, namely 1.0.)

In the third image (middle left), we have added parameter 3 and isosurfaced at a value of 10. Again a sense of minima near $(-1.0, 1.0, 1.0)$ and (1.0, 1.0, 1.0) is gained. In the image middle right, we combine the view in dimensions (1,2,3) with slice planes in dimensions (1,2)—as already seen in step 2 of the exploration at top right—and in dimensions (2,3). The (2,3) slice passes through the larger of the two isosurface volumes, and does not show any other minima. This seems an interesting direction to pursue. Therefore, we toggle parameter 1 to give a slice through the space at the focus point of parameter 1, and this is shown lower left, confirming the view we had from the previous image. Finally, selecting parameter 4 allows us to enter the 3D

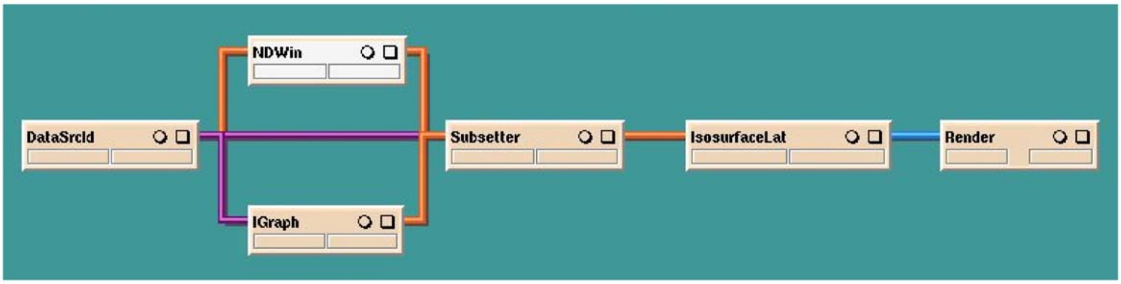

Fig. 8. IRIS Explorer map reflecting the basic structure of the proposed reference model. The DataSrcId module reads in the information on the data such as dimensionality, type (multidimensional or multivariate) and ranges for each dimension. The NDWin module corresponds to the *n-dimensional Window* tool of the model. The IGraph module corresponds to the *Interaction Graph* tool of the model. The Subsetter module implements the *Subsetter* concept of the model. The Focus Data is then sent downstream to an isosurfacing module to be displayed by the Render module.

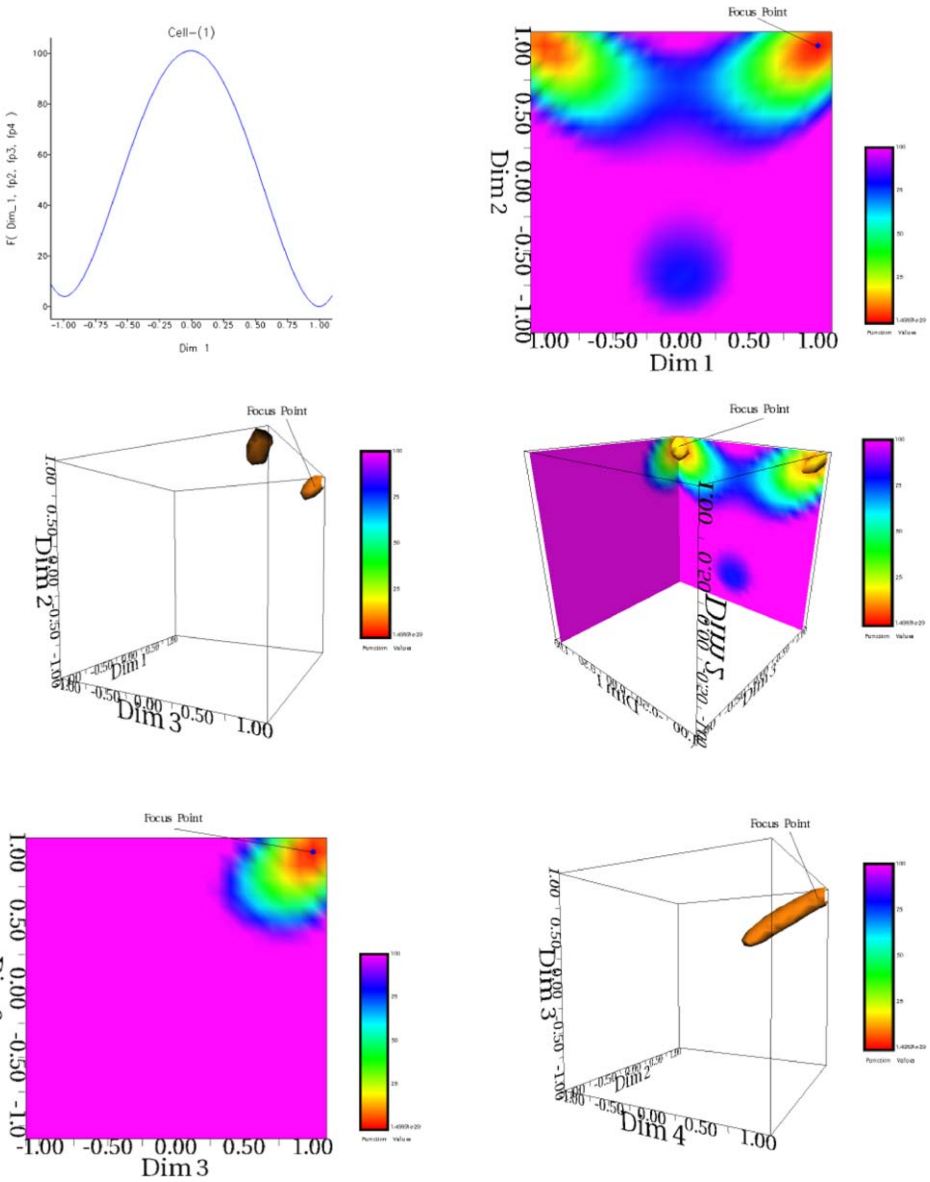

Fig. 9. Progressive exploration of the Rosenbrock function in 4D. These pictures are visualizations with different dimensionality and combination of dimension, taken from the same focus point located at (1,1,1,1). From top to bottom, left to right: Cell-(1), 1D line graph, dimension 1 is free to vary over its range; Cell-(1,2), a 2D coloured map with colours assigned according to the function values, dimensions 1 and 2 are free; Cell-(1,2,3), an isosurface of value 10, dimensions 1, 2 and 3 are free; Cell-(1,2,3), isosurface of value 10, combined with Cell-(1,2) and Cell-(2,3) which are slices of the Cell-(1,2,3); Cell-(2,3), a 2D coloured map, dimensions 2 and 3 are free; Cell-(2,3,4), an isosurface of value 10, dimensions 2, 3 and 4 are free.

space of parameters 2, 3 and 4. Bottom right shows an isosurface of value 10 in the (2,3,4)-space, indicating a tube structure containing the low values of the function.

We can proceed in this way, touring through the 4D space. We could switch to inspect around the traditional starting point for optimization codes, namely $(-1.2, 1.0, -1.2, 1.0)$, by manipulating the bounds for the 4D window on the *n-dimensional Window* tool.

5.2. *Exploring an optimization trajectory*

In this section, we extend the example above to show how we can filter both multidimensional and multivariate data with the same tool. We are interested in seeing the trajectory of successive approximations to the minimum generated by a popular optimization technique, namely the Nelder and Mead simplex method [28].

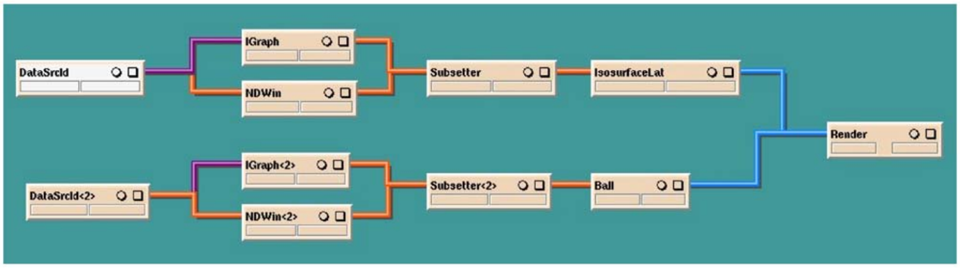

Fig. 10. IRIS Explorer map showing two distinct pipelines, the top one for the multidimensional data (Rosenbrock function in 4D) and the bottom one for the multivariate data (the optimization trajectory). Both pipelines make use of the same type of modules for filtering the data. At the end the output of both pipelines are combined into a single visualization.

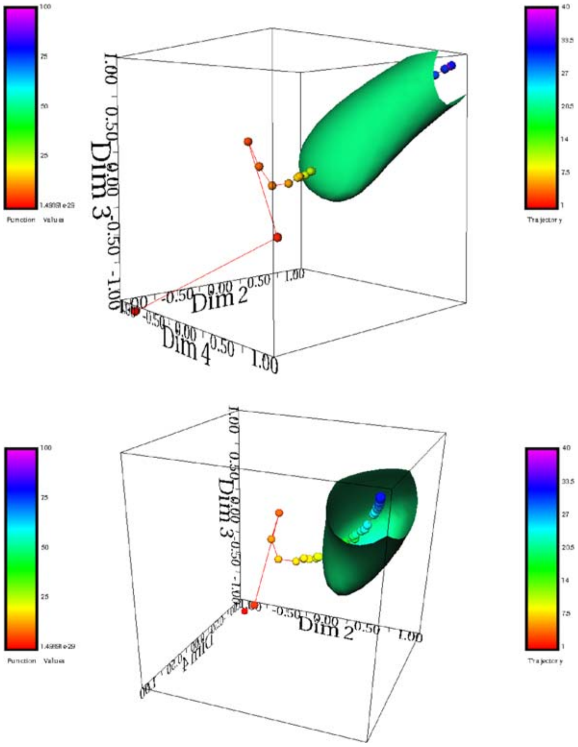

Fig. 11. Combining the visualization of the 4D Rosenbrock function (multidimensional data) with the successive approximations to the minimum generated by the simplex method (multivariate data). Both pictures are of the same subspace Cell-(2,3,4), but taken at distinct viewpoints. The function is represented by an isosurface of value 50 and the optimization trajectory is represented by a sequence of balls connected by line segments. Each ball represents one step of the optimization algorithm, and its colour is associated with the order of the step in the sequence. Each ball's position in 3D space is obtained by considering only the second, third and fourth coordinates. Note that the last step is at the minimum, located at (1,1,1,1).

We can regard the approximations as an ordered sequence of multivariate data items, to be displayed as a scatter plot. Therefore, the data set has four variates (the coordinates of a point in four-dimensional space) and the number of observations is equal to the number of intermediate steps generated by the algorithm until it reaches the minimum. As before, we treat the visualization of the function itself as a multidimensional problem.

We create an IRIS Explorer dataflow pipeline as shown in Fig. 10: notice that the pipeline has two inputs, multivariate data representing the trajectory, and multidimensional data consisting of the function values on a grid.

Fig. 11 shows one stage of the investigation. It shows two views of the 3D space comprising dimensions (2,3,4), corresponding to the final image in Fig. 9 in the previous section. In addition to the visualization of the function, we have a second branch of the visualization pipeline, which inputs as Visualization Data the simplex method trajectory as 4-variate data, and applies a similar filter to generate a set of 3-variate data representing the trajectory. These are mapped as 3D scatter plots and the geometry from the isosurface and scatter plot mapping techniques are merged into a single Render process. The two images represent the 3D visualization taken from different viewpoints. We can see the trajectory of the optimization algorithm as it enters within the isosurface of value 50, and progresses within that isosurface towards the minimum.

This visualization technique allows the algorithm developer to understand the way in which the algorithm converges to the solution, within the high-dimensional space. For example, switching to the (1,3,4)-space, and looking again at isosurface value 50, we get the images in Fig. 12. Notice there are two areas of low function value (corresponding to arms of the '*n*-dimensional banana') but the algorithm is correctly following the downhill path to the region of lowest value.

6. Conclusions and future work

In this paper we have revisited the influential reference model for visualization, proposed by Haber and

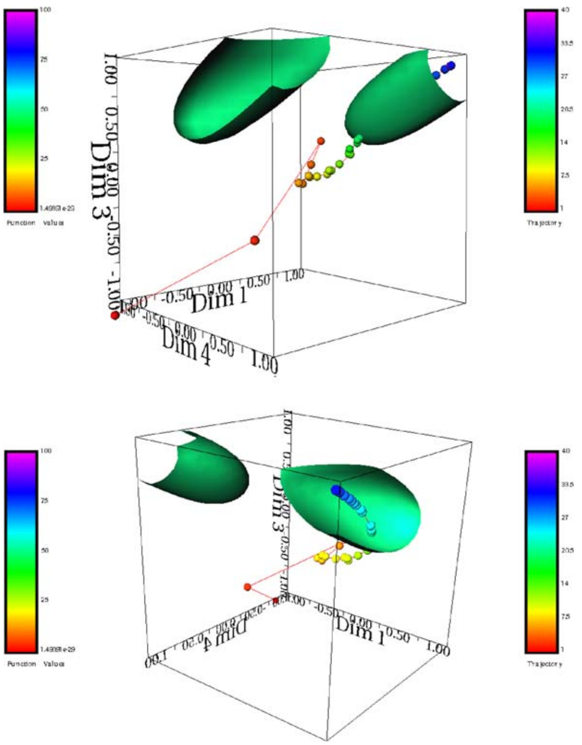

Fig. 12. Further visualization of the 4D Rosenbrock function (multidimensional data) with the successive approximations to the minimum generated by the simplex method (multivariate data). This figure shows the Cell-(1,3,4) with two regions of low value (contrast with Cell-(2,3,4) in Fig. 11 where there is only one region).

McNabb over a decade ago, from the perspective of multidimensional and multivariate visualization. This has allowed us to refine the data enhancement step into a pair of processes: data analysis and filtering. The filter step itself is separated into two further processes, one defining an *n*-dimensional window within the space, the other making a selection of dimensions (or variates) for display. This extended reference model has then been used as the basis of new filter modules which can be used in a dataflow visualization environment. A key aspect of the work is the uniform treatment of both multi-dimensional and multivariate data.

We have demonstrated the approach on the visualization of a popular multidimensional function in optimization, and a popular optimization algorithm generating a multivariate trajectory. Through visualization, we have gained an understanding both of the function, and the algorithm. Although for simplicity the example used is only 4D, the approach scales to between 10 and 20 dimensions (and we show below an example in six dimensions). For any higher dimensionality, we would expect a data analysis technique to be applied as a preprocess to identify the key dimensions for more interactive visual exploration.

The work has been implemented in terms of the visualization system, IRIS Explorer, as three new modules. By integrating into an existing environment (rather than building our own standalone tool), we immediately gain access to the rich functionality of that system.

The paradigm described in this paper is one of sequential exploration of the high-dimensional data sets, through successive low-dimensional subspaces, with a smooth transition between these subspaces. We are now extending this to allow multiple filters or concurrent views, where we retain views of where we have visited in our previous explorations. The model of Fig. 3 extends to that shown in Fig. 13, where the Filtering step now accommodates multiple *Interaction Graph* selections of the dimensions, called 'cells', and a *Workspace Manager* generalises the *Subsetter* process to maintain a record of all cells (i.e., subspaces) that have been selected. Thus the output from the *Workspace Manager* is now an *array* of Focus Data.

A nice feature of this approach is that dynamic changes to the *n*-dimensional window are propagated to all elements of this array, and so all visualizations generated from the array are dynamically changed. The experience is of walking through the *n*-dimensional space (by moving the focus point for example) and seeing the effect in all the subspaces previously created—corresponding to looking around in different directions in the *n*-dimensional world. These dynamic changes are a nice application of the 'Snap' visualization concept

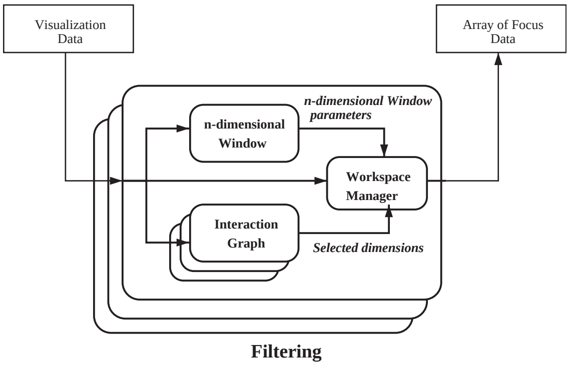

Fig. 13. An extended version of the filtering model to allow multiple filters or concurrent views corresponding to the visited locations in the *n*-dimensional space. The filtering process also accommodates multiple *Interaction Graph* selections of the dimensions ('cells') which correspond to the concept of subspaces. The *Workspace Manager* maintains a record of all cells that have been created and updates them whenever a change of the focus point or dimensional range has been made via the *n-dimensional Window* module. The final output is now an *array* of Focus Data.

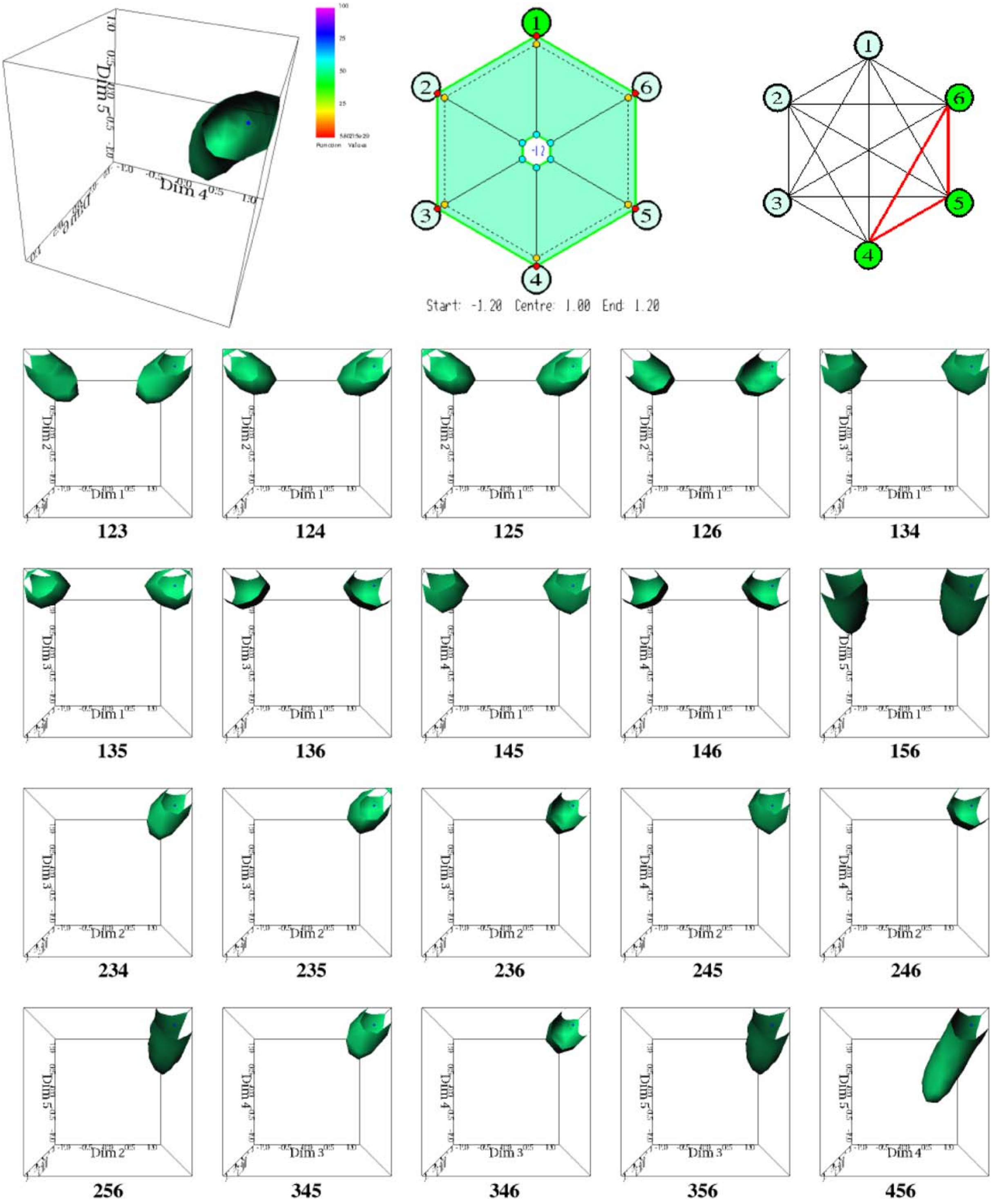

Fig. 14. Showing a visualization of the 6D Rosenbrock function (multidimensional data) through all possible combinations of 3 dimensions, making a total of twenty distinct 3D subspaces. The three pictures on the top, starting from left are: visualization of a chosen subspace, Cell-(4,5,6); the *n-dimensional Window* definition tool set for the 6-dimensional case; and the corresponding *Interaction Graph* having the dimensions 4, 5 and 6 selected. Just below them we have all the subspaces in a subsampled version to allow an overall view of the function. All the subspaces are obtained using the focus point (1,1,1,1,1,1).

introduced by North and Shneiderman [29]. Multiple filter processes (see Fig. 13) allow the behaviour in multiple *n*-dimensional windows to be studied simultaneously, and has the advantage of allowing us to keep track of all visited locations in *n*-dimensional space.

When one looks at a set of multiple views, new insights become possible. In Fig. 14, we look again at Rosenbrock's function, but this time in six dimensions (to illustrate also the way the approach scales to higher dimensions). In the upper part of the figure, we show the visualization in the (4,5,6)-subspace, the *n-dimensional Window* selection tool within the 6D space and the *Interaction Graph* selecting the dimensions 4,5 and 6. However, in the lower part we show all possible 3D subspaces—20 in all. An interesting phenomenon is immediately visible—in all subspaces involving dimension 1, we see a second isosurface (at the other end of the 'banana'!) where the first coordinate is close to -1. In all other subspaces, the ten which do not involve dimension 1, there is just the single isosurface, enclosing the minimum point. By making dynamic changes to the *n*-dimensional Window, and with the 'Snap' concept, we can start to explore this phenomenon in more detail.

Acknowledgements

We would like to thank Dr Ian Coope (University of Canterbury, New Zealand) for his help and suggestions for visualizing optimization trajectories. Thanks also to Dr Chris Goodyer (University of Leeds) for his help.

We acknowledge the financial assistance of the Brazilian Ministry of Education Agency's CAPES scholarship to Selan dos Santos.

References

[1] Wong PC, Bergeron RD. Scientific visualization—Overviews, methodologies and techniques. Silver Spring, MD: IEEE Computer Society Press; 1997. Years of multidimensional multivariate visualization, p. 3–33 [chapter 30].

[2] Martinez W, Martinez A. Computational statistics handbook with MATLAB. London: Chapman & Hall; 2002.

[3] van Liere R, van Wijk JJ. Visualization of multidimensional scalar functions using hyperslice. CWI Quarterly 1994;7(2):147–58.

[4] van Wijk JJ, van Liere R. Hyperslice—visualization of scalar function of many variables. In: Proceeding of the IEEE Conference on Visualization (Visualization '93), 1993. p. 119–25.

[5] Cleveland WS, Visualizing data. Hobart Press, New Jersey, USA, 1993.

[6] Inselberg A, Dimsdale B. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In: Proceedings of the First IEEE Conference on Visualization, 1990. p. 361–70.

[7] Chernoff H. The use of faces to represent points in k-dimensional space graphically. Journal of the American Statistical Association 1973;68:361–8.

[8] Pickett RM, Grinstein GG. Iconographic displays for visualizing multidimensional data. In: Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics, vol. 1, 1988. p. 514–9.

[9] Andrews DF. Plots of high-dimensional data. Biometrics 1972;29:125–36.

[10] Feiner SK, Beshers C. Worlds within worlds: metaphors for exploring n-dimensional virtual worlds. In: Proceedings of the Third Annual ACM SIGGRAPH Symposium on User Interface Software and Technology, 1990. p. 76–83.

[11] Mihalisin T, Timlin J, Schwegler J. Visualizing multivariate functions, data, and distributions. IEEE Computer Graphics and Applications 1991;11(3):28–35.

[12] Keim DA. Designing pixel-oriented visualization techniques: theory and applications. IEEE Transactions on Visualization and Computer Graphics 2000;6(1):59–78.

[13] Interrante V. Harnessing natural textures for multi-variate visualization. IEEE Computer Graphics and Applications 2000;20(6):6–11.

[14] Healey CG, Enns JT. Large datasets at a glance: combining textures and colors in scientific visualization. IEEE Transactions on Visualization and Computer Graphics 1999;5(2):145–67.

[15] Osyczka A. Multicriterion optimization for engineering design. Design optimization. New York: Academic Press; 1985, p. 193–227.

[16] Haber RB, McNabb DA. Visualization idioms: a conceptual model for scientific visualization systems. In: Nielson GM, Shriver B, Rosenblum LJ, editors. Visualization in scientific computing. Silver Spring, MD: IEEE Computer Society Press; 1990. p. p74–93.

[17] NAG IRIS Explorer. Web site, http://www.nag.co.uk (2003).

[18] Open Visualization Data Explorer. Web site, http://www.opendx.org/ (2003).

[19] Schroeder W, Martin K, Lorensen B. The visualization toolkit an object-oriented approach to 3D graphics, 3rd ed. Kitware Inc.; 2003: http://public.kitware.com/VTK/.

[20] Jackson JE. A user's guide to principal components. Wiley series in probability and mathematical statistics. New York: Wiley; 1991.

[21] Mead A. Review of the development of multidimensional scaling methods. Statistician 1992;41(1):27–39.

[22] Yang J, Ward MO, Rundensteiner EA, Huang S. Visual hierarchical dimension reduction for exploration of high dimensional datasets. In: Bonneau GP, Hahmann S, Hansen C, editors. Proceedings of the Joint Eurographics/IEEE TVCG Symposium on Data Visualization 2003. New York: IEEE Press/ACM Press; 2003. p. p19–28.

[23] Becker R, Cleveland W. Brushing scatterplots. Technometrics 1987;29(2):127–42.

[24] Chi EH, A taxonomy of visualization techniques using the data state reference model. In: Proceedings of Symposium on Information Visualization. New York: IEEE Press; 2000. p. 69–75.

[25] Spoerri A. Infocrystal: a visual tool for information retrieval and management. In: Proceedings of the Second International Conference on Information and Knowledge Management (CIKM), Washington, DC, United States, 1993. p. 11–20.

[26] dos Santos SR, Brodlie KW. Visualizing and investigating multidimensional functions. In: Ebert D, Brunet P, Navazo I, Editors. Proceedings of the Joint Eurographics/IEEE TVCG Symposium on Data Visualization 2002. Barcelona, Spain: IEEE Press/ACM Press; 2002. p. 173–81,276.

[27] Conn AR, Gould NIM, Toint P. Testing a class of methods for solving minimization problems with simple bounds on the variables. Mathematics of Computation 1988;50:399–430.

[28] Nelder JA, Mead R. A simplex method for function minimization. Computer Journal 1965;7(4):308–13.

[29] North C, Shneiderman B. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. In: Proceedings of Advanced Visual Interfaces 2000, 2000. p. 128–35.