Visualizing the Behavior of Higher Dimensional Dynamical Systems

Rainer Wegenkittl, Helwig Löffelmann and Eduard Gröller

Vienna University of Technology, Institute of Computer Graphics *

Abstract

In recent years scientific visualization has been driven by the need to visualize high-dimensional data sets within high-dimensional spaces. However most visualization methods are designed to show only some statistical features of the data set. This paper deals with the visualization of trajectories of high-dimensional dynamical systems which form a L_n^n data set of a smooth n-dimensional flow. Three methods that are based on the idea of parallel coordinates are presented and discussed. Visualizations done with these new methods are shown and an interactive visualization tool for the exploration of high-dimensional dynamical systems is proposed.

1 Introduction

Due to the rapid development in computer technology scientists are able to explore larger and larger data sets. This allows not only to increase the number of samples under investigation, but it also allows to increase the number of different parameters sampled at each location. Multiparameter data collections of high dimension are becoming increasingly common in many scientific disciplines. Nearly every field that uses numerical data (such as applied mathematics, computer sciences, finance, market research, medical sciences, social sciences and a lot of other fields) deal with data sets of this type. The need of visualizing such data structures leads to the development of special visualization and data brushing methods.

Most of these visualization techniques are used to visualize statistical characteristics such as clustering of data or outliers. In section 2 some of these methods are listed and shortly discussed. They are very well suited for discretly sampled multivariate data sets and adjusted to specific requirements of this kind of data, e.g., the detection of correlation. They do not, however, reveal very well the underlying structure when the samples are taken from a smooth, continuos flow in a high-dimensional space. Such data sets can for example result from dynamical systems.

Dynamical systems are used for simulating natural processes by a mathematical model. This model is described by a set of differential equations, one equation for each state variable. The more simple a model is, the less state variables does it need. When a sophisticated, well adapted model is needed, more state variables have to be added forming a high-dimensional dynamical system. These systems have been investigated with the same visualization methods as sampled real world data as described in section 3.

The dynamical behavior of a dynamical system can be described by its topology, which consists of specific features like fixed points, cycles, attractors, repellors, and separatrices. These structures might become visible when a set of trajectories within phase space is shown. As statistical visualization methods are typically not designed to show integral curves within a high-dimensional phase space, we focused our work on this problem. In section 4 several methods for visualizing high-dimensional trajectories are presented. Section 5 shows some results achieved with the described methods. As this is work in progress some extensions and ideas for future work are also discussed.

*Institute of Computer Graphics, Vienna University of Technology, Karlsplatz 13/186/2, A-1040 Vienna, Austria email:{wegenkittl, loeffelmann, groeller}@cg.tuwien.ac.at

2 Visualizing Multidimensional Data

Multiparameter data sets are becoming more and more common in many scientific disciplines. Depending on their origin several different aspects of the data can be distinguished. Data sets can be discrete (like measured data) or continuos (like dynamical systems given by differential equations), they can be spatially coherent (such as medical images) or spatially incoherent (like census data). Data sets may consist of a collection of sampled data. Each sample is an n-dimensional data item. For scalar values n equals 1, if each sample consists of three different variables n equals 3. These n dimensions are called dependent variables. They vary with the location they are sampled at. The m-dimensional space where samples are taken is described by an m-dimensional lattice. The m location variables are called independent. High-dimensional data structures L_n^m may have a large number $n > 3$ of sampled variables, or they can be sampled within a high-dimensional ($m > 3$) space [1].

Visualization methods for high-dimensional data can be used to represent high-dimensional samples on a low-dimensional lattice or they are used to display low-dimensional data structures which are sampled within a high-dimensional lattice. Two important goals of visualization techniques are the identification of individual parameters (what is the value of some data within a specific region), and the detection of regions and correlation of variables (e.g., where do data points of a specific value reside).

Basically five different sets of visualization methods for high-dimensional data can be distinguished. These methods are briefly discussed in the following five subsections. The visualization of high-dimensional data often uses a combination of two or more of these methods, e.g., color coding is used on focused data and assisted by interactive sonification. Most of the listed methods are easy to implement and allow a fast and interactive exploration of data sets. This is very important since interactivity introduces time as a fourth dimension into the space where the data is explored.

2.1 Attribute Mapping

Attribute mapping is one of the most common methods to visualize high-dimensional data. This method uses one or two-dimensional lattices to define some simple geometric primitives, e.g., contours or planes. The attributes of these geometric primitives can be used to visualize the remaining variables. The most often used attribute is the color of the geometric primitive. Color coding can be employed to display up to three variables. Each of these variables is mapped to one component of the underlying color model. The most common color models are the RGB model and the HLS model. Fig. 1 shows a possible mapping for these two color models.

A major advantage of color coding is the fact that it is very often used (e.g., weather forecast maps). Therefore many users are familiar with this kind of visualization. Another advantage is the easy calculation and interpretation of color coded images. A disadvantage is that colors do not have a unique order, so color coded

RGB
Variable 1⟶Red
Variable 2⟶Green
Variable 3⟶Blue

HLS
Variable 1⟶Hue
Variable 2⟶Saturation
Variable 3⟶Brightness

Figure 1: Attribute mapping with the RGB and the HLS color model

images have to show a color legend to allow an exact interpretation. Another disadvantage is the restriction to encode only three variables. The encoding of all three components of a color model leads to an image where the three different variables are not distinguishable any more. In realistic applications color coding is for example restricted to two variables. Another problem is the fact that about eight percent of the population suffers from some kind of color blindness.

2.2 Geometric Coding

Geometric coding, originally proposed by Pickett [17], is used for displaying high-dimensional data on a low-dimensional lattice by displaying distinct geometric objects within the lattice and mapping the high-dimensional data to some geometric features or attributes of these objects. In the following some of the most well known objects for geometric coding are listed.

Glyphs are often utilized for interactive exploration of data sets. A glyph is a generic term describing a graphical entity whose shape or appearance is modified by mapping data values to some of its graphical attributes. An interactively positioned glyph adapts its appearance according to the underlying data. Variables can be mapped to the length, shape, angle, color and transparency of the glyph. Examples of this kind of visualization are given in [13] and [14].

Icons Another visualization method uses icons as basic primitives. An icon is a generalization of a single pixel to higher dimensions having multiple perceivable features and attributes. The fact that shape and color are perceptually separable features is used for the display of color icons. They merge separable features by using color, shape and texture perception to code multiple variables. In [15] an icon is presented that allows to encode six different parameters by color coding six different lines within a square icon. Since the color coding scheme could code one variable within a color component the method could be even extended to visualize 18 parameters (three parameters for each of the six lines). This number can even be increased by subdividing the length of the lines of the icon and apply different parameters to the line segments. Individual variables are not recognizable any more, but correlation patterns appear.

Chernoff Faces are a well known example to encode multivariate data. The displayed objects are stylized faces and the variables influence the appearance of the shape of the face, the mouth, eyes, nose and eyebrows. Chernoff used this method to visualize a twelve-dimensional data set on a two-dimensional lattice [4]. Due to the fact that the human brain is well skilled in recognizing different faces it should be easy to detect regions of clustered data and outliers with this method.

Data Jacks as used by Cox, Ellson and Olano are three-dimensional shapes with four different limbs. Again the length and the color of these limbs are taken to code up to sixteen parameters.

m-Arm Glyph is another glyph for the visualization of high-dimensional data sets. This two-dimensional structure introduced by Pickett and Grinstein [18] consists of a main axis and m arms attached to it. The length and thickness of each arm and their angle to the main axis encode different parameters.

2.3 Sonification

Sonification is another method of making more than three dimensions perceptible for a researcher. A sound is produced according to the mapped parameters [7]. Variables can be mapped to the loudness, the pitch and even to the orchestration of the sound. One disadvantage of these methods is that the various parameters that characterize a sound influence each other, i.e., a sound at a constant volume but with changing tune is perceived as if the volume changes too. Nevertheless sonification is a good tool for visualizing high-dimensional data sets because it stimulates a different sense organ and thus may avoid overloading the visual system.

2.4 Reduction of Dimension

An obvious way of visualizing high-dimensional data sets is the reduction of dimension. This can be done by either focusing, where only part of the whole data set is shown, or by linking, where some focused parts are linked together to represent the whole data set.

2.4.1 Focusing

Focusing techniques may involve selecting subsets, reduction of dimension by projection, or some more general manipulation of the layout of information on the screen.

Examples for subset selection techniques are panning, zooming [3] and slicing [6]. Reduction of dimension can be achieved by simply projecting high-dimensional spaces along some axes into a low-dimensional space and/or color coding of multiparameter images. Techniques for more general layout manipulation include a variety of techniques for adapting to a user's point of interest such a fisheye views [5] and rooms [11].

2.4.2 Linking

One consequence of focusing is that each view will only convey partial information about the data. This can be compensated by linking several focused visualizations. Linking can be done by sequencing several visualizations over time (guided tour) or by showing them in parallel simultaneously. The parallel visualization can be done in separate windows as for example with the well known scattered data plots. It can also be done within one single image by using parallel coordinates [12] (see section 2.5) or dimensional stacking [2]. Dimensional stacking is a recursive projection method, where two dimensions are mapped on the horizontal and vertical axis, creating a discrete grid. Within each cell of this grid the process is applied again with the next two dimensions that have not been used so far. This process continues recursively until all dimensions are assigned. A new coordinate system is positioned at a fixed location within the previous coordinate system. Another form of linking is the use of hierarchical axis [16] where n independent variables are hierarchically stacked on the x-axis and the analysis of one dependent variable is done on the y-axis. The pixel constraint on the x-axis is overcome by a set of hierarchical (color coded) symbols that represent the data not only at data points, but along whole data lines (or subspaces).

2.5 Parallel Coordinates

Parallel coordinates [12] represent dimensions on parallel axes. All axes are arranged orthogonal to a horizontal line uniformly spaced on the display. Each point of the data set corresponds to a polyline that intersects the parallel coordinate axes at the coordinate values of the data point. This method allows the detection of special characteristics of the data under investigation by looking at the patterns that are produced by the polylines. If all variables reside, for example, on a line in n-space, then all polylines will intersect each other at specific points between the (vertical) parallel coordinate axes. Thus a line within n-space can be visualized by a set of points between the parallel coordinate axes. This gives a duality between points and lines which is an interesting feature of parallel coordinates. By interactively brushing through the data set statistical characteristics like outliers and clusters can be recognized very easily.

3 High Dimensional Dynamical Systems and Visualization

A lot of natural phenomenona can be approximately described by differential equations. Scientists in many fields, e.g., in chemistry, physics, biology, economy, medical research and other fields, investigate models called dynamical systems. Each differential equation describes the change of one state variable, thus a set of n differential equations defines the behavior of n state variables describing a n-dimensional dynamical system. For each "sampled" set of state variables the differential equations give an n-dimensional vector describing the direction of the flow at the specific point. Thus the discretized flow described by n differential equations forms a vector field of dimension n, where each vector itself is of dimension n. This can be interpreted as an L_n^n data set (see section 2).

The main difference to measured "real world" data is that a dynamical system typically describes a (maybe) complex but smooth flow. The behavior of this flow is entirely determined by its topology. Visualization methods have to reveal this information to the scientist. Figure 2 (similar to [19]) shows the visualization methods described in section 2 and some advantages and disadvantages. As can be seen all the methods focus on the representation of statistical quantities and do not give any information on the underlying flow topology.

An important question that arises is what kind of information has to be shown to give an insight into the topology of a dynamical system. For the interpretation of the behavior of the system each point within n-space can not be investigated by itself but has to be seen in respect with its neighborhood. This is due to the fact, that the data is derived from a continuos flow field. There are two basic approaches concerning this task:

1. The neighboring information can be calculated from the vector field, e.g., by interpreting the Jacobian matrix, and the derived (for example scalar) data is displayed in n-space. The derived data has not to be a scalar, but can be of dimension m itself. The directional information at each point in n-space may be projected to an m-dimensional data object that describes some local features of the system. An example is given by [14], who use a glyph for displaying topological information of the flow such as convergence, shear, vorticity and curvature.

2. A direct global visualization of the flow can be done by starting short integral curves, so called trajectories, which follow the flow, at the nodes of an n-dimensional regular grid. The detection of topological features such as separatrices can be done visually by interpreting the flow directions of the trajectories. Unfortunately displaying n-dimensional trajectories is a non-trivial problem. This problem will be approached in section 4.

Displaying an n-dimensional trajectory is an important task to allow a direct global visualization of the behavior of a dynamical system. One possible way for doing this uses the fact that an n-dimensional directional vector can be described by $n-1$ angles. Since the direction changes smoothly along a trajectory these angles can also be used to describe the behavior of a trajectory thus reducing the dimension by one. This would allow us to display four-dimensional trajectories in three-space. The investigation of topological structures such as two-dimensional and three-dimensional manifolds in four-space has already been done by Hanson (see for example [9]).

We wanted to find more general methods for visualizing n-dimensional trajectories with $n \geq 4$. In the next section three methods are described for addressing this problem. Two of these methods are extensions to Inselbergs parallel coordinates technique [12]. The third method uses projection and a new type of linking. Theoretically all these methods are not restricted to a specific maximum number of displayed dimensions.

4 Our Approach Towards Visualizing High Dimensional Dynamical Systems

4.1 Extruded Parallel Coordinates

Extruded parallel coordinates are based on parallel coordinates. With parallel coordinates a trajectory is sampled during its evolution at discrete points in time $\{x(t_0), x(t_1), x(t_2), \ldots\}$ and its coordinates are inserted as polylines in a parallel coordinate system (see left side of Fig. 3). Instead of using the same coordinate system for each sample we now move the parallel coordinate system along the third spatial axis. The polylines of the samples can be viewed as cross sections of a moving plane with a complex surface which defines the trajectory. The right side of Fig. 3 shows this surface and the moving parallel coordinate system at the end of the surface.

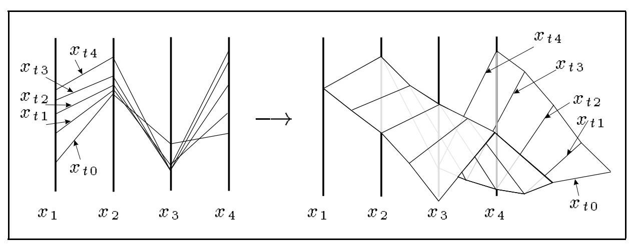

Figure 3: A discrete sampled trajectory in parallel coordinates (left) and a three-dimensional extruded surface defining the same trajectory (right)

The geometry of the surface can be generated and modified fast and easily allowing an interactive exploration of trajectories in the dynamical system. All exploration methods used by parallel coordinates can be used as well since rotating the surface and parallel projecting it reveals exactly the parallel coordinate representation. This can be used as a starting point for the exploration of the trajectory. Clustering and correlation can be visually detected (see [12]). Rotating the surface a little bit reveals the evolution of the trajectory over time without any animation methods that would have to be used for parallel coordinates.

Convergence or divergence can be observed by varying the starting coordinates of the trajectory slightly. The changing shape of the surface shows for each dimension if its attracted or repelled by some topological structure. If for one dimension a whole interval is used for the starting points of trajectories the surface expands to a volume which again can be used for a structural analysis of the underlying dynamical system.

	Attibute Mapping	Glyphs	parallel Coordinates	Reduction of Dimension
dimension	$n \leq 3$	$n > 3$	$n > 3$	$n \geq 3$
identification of individual parameter	only for one Variable	only for few glyphs	good for low quantity of data	depends on visualization method
discontinuities, trends, outliers	good	good	good	good
local extrema	possible (depends on color coding scheme)	possible (e.g., shape coding)	good	possible
correlations	possible	possible	good (for neighbouring axes)	

Figure 2: Methods for the Visualization of High Dimensional Data (similar to [19])

4.2 Linking with Wings

This idea is based on a new method of linking data. Two arbitrary dimensions of the high-dimensional system are selected and displayed as a two-dimensional trajectory within a base plane (the high-dimensional trajectory is projected into a two-dimensional subspace). The third dimension (along the z-axis) can now be used to display a third variable over the base trajectory. if the resulting three-dimensional trajectory is connected with the base trajectory this connection can be thought of as a wing on the base trajectory. This wing can be tilted at each point within a plane normal to the base trajectory. When different tilting angles are used several additional dimensions can be linked to the base trajectory on separate wings (see Fig. 4).

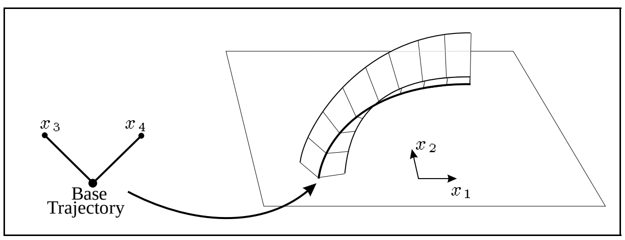

Figure 4: Two-dimensional base trajectory with two wings for the third and fourth dimension linked to it

Theoretically any number of wings can be added to display high-dimensional trajectories. As the number of wings increases occlusion might become a severe problem. To avoid this the wings can be rendered transparently with opaque trajectories at the top. Again this method is easy to implement and fast allowing its use within an interactive exploration tool. Such a system could for example allow to animate a flight along the base trajectory, where the change in the variables linked to wings can be seen easily.

The wings can also be textured with a grid texture allowing an exact measurement of the wing dimensions. To overcome the problem of occlusion texture can be used to modulate the transparency of the wings. Negative values of variables displayed on wings can be shown by expanding the wing to the opposite side of the base trajectory or by using an additive offset for each wing so that the minimum of each variable is mapped to zero. For this approach the zero line has to be encoded on the wing. This can again be done by using some specific texture.

When a four-dimensional trajectory has to be displayed, the angles of the wings can be chosen to be $\frac{\pi}{2}$ and $-\frac{\pi}{2}$. In this case wings lie within the base plane. The trajectory is shown in a two-dimensional image without any projectional distortions .

Self intersection of the wings can be a problem. The size of the wings should be chosen to be rather small with respect to the size of the base trajectories. This is required to avoid massive occlusion. Furthermore the angles of the wings must not be too big. This ensures that the occurrance of self intersections of the wings within regions where the base trajectory exhibits big curvature is not a severe problem.

4.3 Three-dimensional Parallel Coordinates

Three-dimensional parallel coordinates are again based on the parallel coordinate method. As described in section 2.5 the basic idea of parallel coordinates is to depict each variable on a one-dimensional space. All these one-dimensional spaces are put together within a two-dimensional space and linked with one-dimensional polylines. All information is packed in the two-dimensional space. Since the visualization of three-dimensional structures poses no problem we increased each dimension of the parallel coordinate method. The basic information now resides in separate two-dimensional spaces (planes) where two dimensional trajectories are shown. These planes are combined within three-space and linked by surfaces which connect the separated projections of trajectories (see left part of figure 5).

The positioning of the planes is more flexible in comparison to the parallel lines of the parallel coordinate method. The planes can be moved and rotated within three-space to avoid occlusion in different regions of the structure. The right side of Fig. 5 shows two coinciding planes, where the connecting surface is bended normal to the planes to give a better overview of the linking.

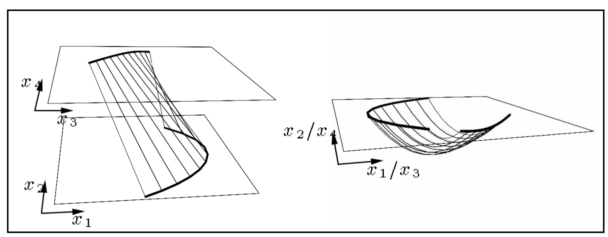

Figure 5: Linking parallel planes instead of lines extends the idea of parallel coordinates by one dimension (left); these parallel planes can coincide, with three-dimensional linking surfaces (right)

Placing the planes orthogonal to each other as shown on the left side of Fig. 6 is another possibility of showing the structure of the linkage of the separated trajectories. All these arrangements can be stacked to allow the representation of even more dimensions. An example is given in the right side of Fig. 6, where an eight-dimensional trajectory is shown. Since the planes and linking surfaces are rendered transparently or are approximated with lines, no massive occlusion occurs (notice that all four separate projections of the trajectory can be seen easily).

If the structure of the trajectory is more complex, as for example in the case of a trajectory of a chaotic attractor, the resulting visualization can be rather crowded. Since the rendering of the presented structures is fast, interactive brushing methods can be used to overcome this problem. For instance the linking surface can be rendered transparently, and only a small temporal interval of the linkage is rendered opaquely. When the interval is moved corresponding parts of the trajectories can then be detected (see Fig. 12).

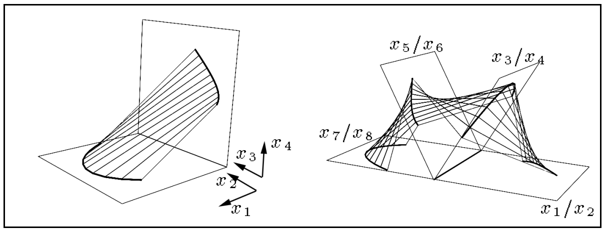

Figure 6: Planes might be orthogonal (left) and stacking allows an arbitrary number of dimensions (right)

When within one dimension a whole interval is chosen as initial region, the linking surfaces expand to linking volumes, whose changing thickness reveals convergent and divergent regions of the trajectory. Again this interval can be chosen interactively for each dimension allowing a quick exploration of the behavior of the dynamical system.

5 Results

All three techniques described in section 4 have been implemented into a rendering system. This prototype implementation has been used to test the methods for robustness and expressiveness. It turned out, that occlusion is a problem for still images, but adding transparency and using animation help to overcome this disadvantage. The methods should work fine within an interactive exploration tool. The development of such a tool is currently done.

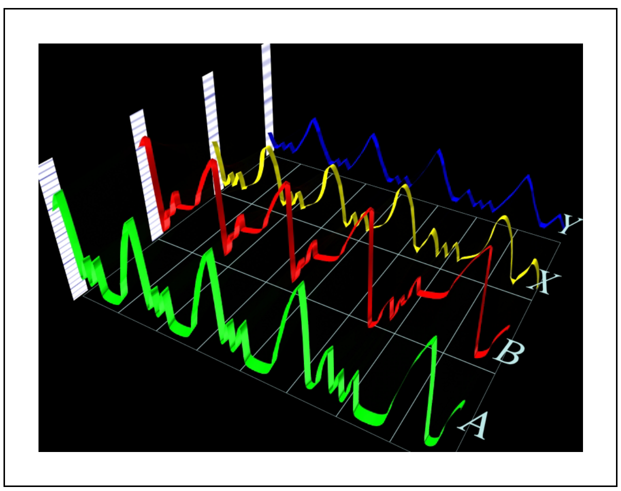

Figure 7: Mixed-Mode oscillations as formed by the peroxidase-oxidase model

Despite the fact that interactivity is necessary to use the full potential of the presented techniques the following images give a flavor of the different visualization methods. Figure 7 shows a trajectory of a four-dimensional dynamical system, i.e., the peroxidase-oxidase reaction model [10], as extruded parallel coordinates. The chemical model shows mixed-mode oscillations (oscillations with alternating amplitudes) in all variables, which is clearly visible with extruded parallel coordinates. The use of parallel coordinates would have needed animation techniques to show the mixed-mode behavior of the model.

Figure 8 shows a trajectory of a chaotic attractor (five-dimensional system). Again extruded parallel coordinates are used

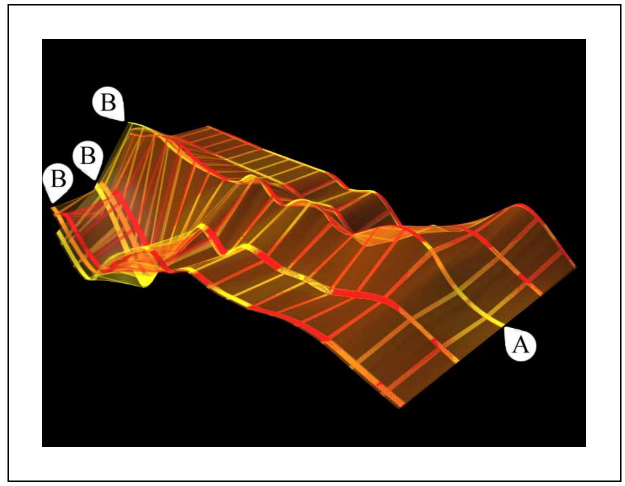

Figure 8: Small jitter at the starting point (A) of a chaotic attractor yields large effects after some time period (B)

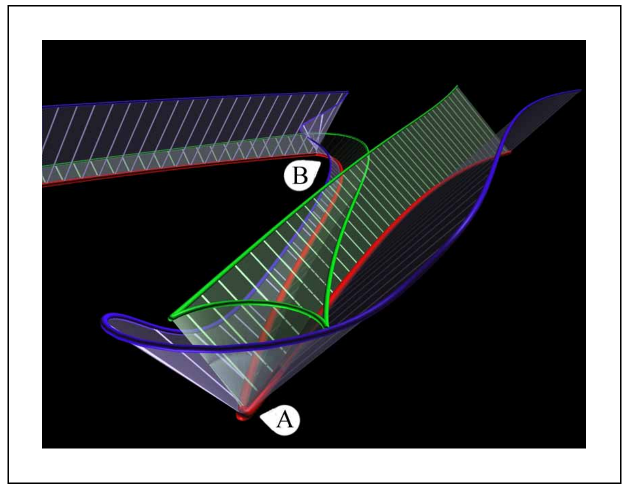

Figure 9: Self intersecting wings due to high curvature of the base trajectory

for the visualization. To show the chaotic behavior of the attractor, the starting point of the third dimension is slightly jittered (point A) and three different trajectories (shown with different colors) are superimposed. The tiny differences of the starting coordinates produced large changes after a few time steps of integration (points B). Interestingly these differences are noticeable only in the first three dimensions so far (integration over a longer time interval showed diverging behavior in the fourth and fifth dimension also).

In Fig. 9 wings are applied to a base trajectory of the four-dimensional wonderland model [8], which describes the interactions of population, economy, environmental health and pollution. Rather big wings with larger angles are used intentionally to show the artifacts due to the intersecting wings at the cusp of the base trajectory (point A). In spite of the self intersections of the wings the overall behavior of the system is visible: The green trajectory declines constantly along the whole trajectory, whereas the blue trajectory (describing the environmental health) collapses at point A and regenerates at point B (here the wings intersect again).

Figure 10 shows a hedge-hog visualization of a four-dimensional data set. On a regular four-dimensional grid flow directions are depicted. The visualization shows a cyclic behavior in the first and second dimension, whereas the third dimension is attracted and the fourth is repelled by the origin. The cyclic behavior in the ground plane is additionally visualized by using oriented line integral con-

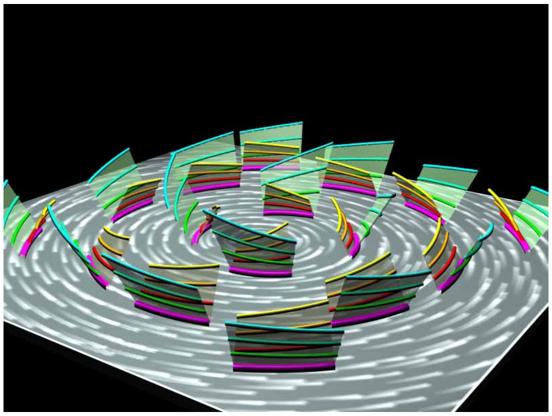

Figure 10: Hedge-hog visualization of a four-dimensional data set

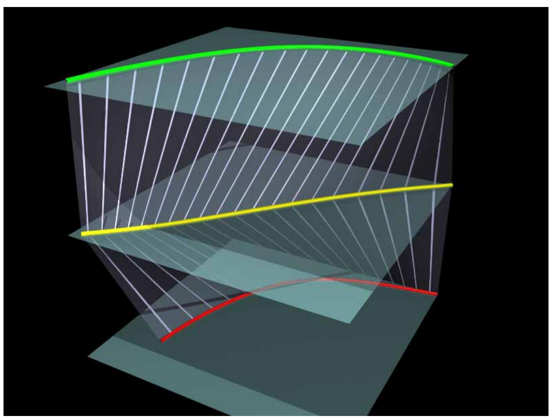

Figure 11: A six-dimensional stacked predator-prey model with simple linkage

volution [20].

In Fig. 11 a six-dimensional predator prey system is stacked with the extended parallel coordinate method. Due to the simple shape of the separated trajectories, the linking surfaces can be seen easily and so the whole dynamics of that specific trajectory is visible. Fig. 12 on the other hand shows a trajectory of a complex dynamical system derived from a Lorenz system. Here the structure of the linkage can not be perceived easily. So a small temporal interval has been highlighted on the linking surface. This interval can be animated or interactively moved forward and backward to reveal the structure of the linkage.

6 Conclusion and Future Work

This paper presents various ideas to visualize trajectories of higher dimensional dynamical systems. These techniques are: an extrusion of parallel coordinates to the third dimension, a two-dimensional projection with a new type of linking dimensions on a wing of a trajectory, and the extension of the parallel coordinate technique by using "parallel" planes and three dimensional surfaces for the linkage. Although a prototype implementation shows some interesting results further research is still necessary to gain experience in applying these techniques to large real world data. Since all methods are easy to implement and allow a fast calculation of visualizations they meet the basic requirements for the implementation within an interactive exploration tool. In the process of implementing this software some data brushing methods can be adapted to work with the presented methods.

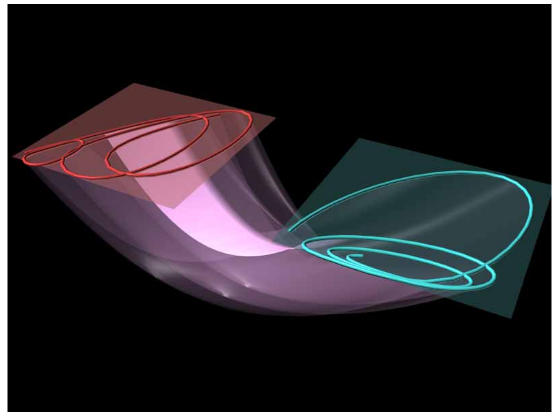

Figure 12: Complex linkage with highlighted time interval for the trajectory of a chaotic attractor

Another interesting phenomenon we like to explore in the future is the structure of the linkage surface. When using parallel coordinates, some structures that are formed by the polylines give insight to the structure of the original data. Since the methods introduced in this paper are extensions to the parallel coordinates technique, comparable structures may also occur and give insight to the structure of the dynamical system. Volume visualization methods and cross section methods will be used to extract these structure out of the surfaces linking the separated projections of trajectories.

References

[1] Bergeron, G. Grienstein, *A Reference Model for the Visualization of Multi-Dimensional Data*, Proceedings of Eurographics '89, 1989.

[2] J. LeBlanc, M. O. Ward, N. Wittels, *Exploring N-Dimensional Databases*, IEEE Visualization '90 Proceedings, pages 230–239, 1990.

[3] A. Buja, C. Hurley, J. A. McDonald, *Elements of a viewing pipeline for Data Analysis*, Dynamic Graphics for Statistics, Wadsworth and Brooks/Colr, Belmont, CA, 1988.

[4] H. Chernoff, *The use of faces to represent points in k-dimensional space graphically*, Journal of the American Statistical Association, 68, pages 361–368, 1993.

[5] G. W. Furnas, *Generalized Fisheye Views*, Proceedings of CHI '86, Human Factors in Computing Systems, New York, 1986

[6] G. W. Furnas, *Dimensionality Constraints on Projection and Section Views of High Dimensional Loci*, Computer Science and Statistics: Proceedings of the 20 Symposium on the Interface, Washington, D.C., 1988

[7] G. Grienstein, S. Smith, *The perceptualization of scientific data*, Proceedings of SPIE '90, Santa Clara, CA, 1990.

[8] E. Gröller, R. Wegenkittl, A. Milik, A. Prskawetz, G. Feichtinger, W. C. Sanderson, *The Geometry of Wonderland*, Chaos, Solitons & Fractals, 7(12), pages 1989–2006, 1996.

[9] A. J. Hanson, R. A. Cross, *Interactive Visualization Methods for Four Dimensions*, IEEE Visualization '93 Proceedings, pages 196–203, 1993.

[10] T. Hauck, F. W. Schneider, *Mixed-Mode and Quasiperiodic Oscillations in the Peroxidase-Oxidase Reaction*, The Journal of Physical Chemistry, 97(2), pages 391–397, 1993.

[11] D. A. Henderson, S. K. Card, *Rooms: The Use of Multiple Virtual Workspaces to Reduce Space Contention in a Window-Based Graphical User Interface*, ACM Transactions on Graphics, Vol. 5(3), pages 211–243, July 1986.

[12] A. Inselberg, B. Dimsdale, *Parallel Coordinates: a Tool for Visualizing Multidimensional Geometry*, Visualization '90 Proceedings, pages 361–378, 1990.

[13] G. D. Kerlick, *Moving Iconic Objects in Scientific Visualization*, IEEE Visualization '90 Proceedings, pages 124–129, 1990.

[14] W. C. de Leeuw, J. J. van Wijk, *A Probe for Local Flow Field Visualization*, IEEE Visualization '93 Proceedings, pages 39–45, 1993.

[15] H. Levkowitz, *Color Icons: Merging Color and Texture Perception for Integrated Visualization of Multiple Parameters*, IEEE Visualization '91 Proceedings, pages 164–170, 1991.

[16] T. Mihalisin, J. Timlin, J. Schwegler, *Visualization and Analysis of Multi-variate Data: A Technique for All Fields*, IEEE Visualization '91 Proceedings, pages 171–178, 1991.

[17] R. M. Pickett, *Visual Analyses of Texture in the Detection and Recognition of Objects*, Picture Processing and Psycho-Pictories, Academic Press, New York, 1970.

[18] R. M. Pickett, G. Grienstein, *Iconographic Displays for Visualizing Multidimensional Data*, Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, China, pages 514–519, 1988.

[19] H. Schumann, *Visualisierung wissenschaftlicher Daten*, lecture notes, University of Rostock, 1993.

[20] R. Wegenkittl, E. Gröller, W. Purgathofer, *Animating Flow Fields: Rendering of Oriented Line Integral Convolution*, Proceedings of Computer Animation '97, IEEE Computer Society, pages 15–21, 1997.