PolarEyez: A Radial Focus+Context Visualization for Multidimensional Functions

Sanjini Jayaraman

Thesis submitted to the faculty of
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
In
Computer Science and Applications

Dr. Chris North, Chair
Dr. Scott McCrickard
Dr. Roger Ehrich

January 6th, 2003
Blacksburg, VA, USA

Keywords: Visualization, Multi-dimensional functions, focus+context, radial

PolarEyez: A Radial Focus+Context Visualization for Multidimensional Functions

Sanjini Jayaraman

(Abstract)

Multi-dimensional functions are characterized by a large number of parameters on which the functional value depends. They are commonly used in engineering problems such as image analysis and system modeling. Multi-dimensional function spaces are very difficult to understand due to their multi-dimensional nature and the presence of a large number of data points in the functional space.

A point called the focal point is selected by the user in the vast multi-dimensional parameter space. Rays called “focal rays” emanate from the focal point in all directions to the boundaries of the functional space. The focal rays contain functional data points. The focal point is mapped onto the center of the visualization with the focal rays arranged radially around it. The degree of detail decreases as we move away from the focal point toward the edges of the visualization in accordance with the focus+context technique. The functional values are mapped onto a color scheme with shades of green representing positive function values, and shades of red representing negative function values. Interactive features such as the ability to change the focal point, highlighting of functional values aid the user in exploring and analyzing the functional space. The algorithm, practical applications of the visualization approach and results of formative evaluation studies are also elaborated in this thesis.

The contributions of this thesis are fourfold, namely, providing an overview of the functional space, equal treatment of all dimensions, improved scalability and a smooth blending of details with the overview.

Acknowledgement

The future belongs to those who believe in the beauty of their dreams.
--Eleanor Roosevelt

This Master’s thesis is an example of the transformation of a dream into reality through dogged pursuit and perseverance. This transformation would not have been possible without the continued support and encouragement of various people.

I am greatly indebted to my advisor Dr. Chris North for introducing me to the exciting and emerging world of visualizations through his course “Information Visualizations”. His course sparked my interest and motivated me to step into the world of visualizations by doing a masters thesis. Dr. North has guided me immensely during the course of my thesis with his motivating words and practical advice. His positive attitude has been very infectious and had a great impact on my continued pursuit in my work.

I would also like to thank all my friends especially Ramesh Vishwanathan and Srinivasan Vasudevan who have stood by me when I was handicapped by bursitis and carpal tunnel syndrome. Without their support the last phase of my thesis work and my defense would not have been possible.

I am grateful to my mother for supporting and encouraging me all through my life. But for her vision I would not have pursued graduate studies. Her guiding words were instrumental in making my IEEE Visualization 2002 presentation (a product of my thesis) a very successful one.

Last but not the least I thank God for guiding me throughout and providing richer experiences for self improvement and awareness. The grace of the Almighty has helped me in acquainting myself with noble people whose virtues has had a indelible positive impact on my life.

Table of Contents

List of Figures

List of Tables

Chapter 1: PROBLEM DESCRIPTION

1.1 Introduction

This chapter introduces the concept of multi-dimensional functions and their importance with a few practical examples. The central theme of this research, the motivation and the challenges are described. A brief idea of the technique developed and an example is also provided at the end of the chapter.

1.2 Multi-dimensional functions

Multi-dimensional functions are of interest to engineers and mathematicians for understanding the behavior of systems. System behavior is represented as a function of several parameters. Formally, the general form of any function is

$$y = f(x_1, x_2, x_3 \ldots\ldots x_n)$$

where, "y" is called the "dependent variable" and $x_{1 \ldots} x_n$ are called the "independent variables". A function expresses the relationship between the dependent and independent variables. Each independent variable constitutes a dimension. Such functions having a number of independent variables are called multi-dimensional functions. In scientific problems, the number of independent variables is large making them very complex to analyze and understand.

Multi-dimensional functions can be expressed as mathematical formulae or as multi-dimensional arrays of functional values. Mathematical formulae express the relationship between the dependent and independent variables as a mathematical expression. Functional values can be obtained by substituting the parameter values in the expression. Formulae are commonly found in mathematical modeling applications, such as network communication models for obtaining throughput. Multi-dimensional arrays of functional values are obtained by discretely sampling the functional space on a uniform grid. Values between the grid points are obtained by interpolation. Such arrays of sampled values are collected from scientific experiments, complex simulations or advanced models, where the formula cannot be explicitly specified.

Several transformations and computations are performed in multi-dimensional functional spaces to understand them. Examples of tasks that are of interest to users are

- finding correlations between the functional value and the variables (e.g. Is the functional value affected by a particular parameter alone? Does a particular parameter have any effect on the function value?)
- identifying outlier functional data points
- local maximum and minimum values
- finding points whose functional value is greater than the desired or expected value
- zeros of a function
- optimum values of parameters
- determining stability of functional values in localized regions

1.3 Examples of multi-dimensional functions

Multi-dimensional functions are commonplace in several engineering disciplines such as aerospace, economics, dynamics, and neural networks. A generic example is a function representing the performance of a system. The independent variables are the factors on which the system performance depends. A study of such a function yields an insight into the effect of various parameters on the performance. Performance tuning can then be hypothesized and studied. Two specific examples of multi-dimensional functions are elaborated below.

1.3.1 Neural Network application

In neural networks, multi-dimensional functions are widely employed for the representation of facts and in the expression of learning functions. Learning functions are utilized by applications for recognizing facts. A common example of a neural networks application is character recognition. The input is a multi-dimensional function representing the bitmap of a handwritten character. The neural network learns the bitmap input forming a learning function (is a function of several variables) in the process. The learning function enables the neural network to identify any hand-written character presented to it at a later stage, the response being the appropriate character value (such as 0-9, A-Z, a-z) [Humphrys97].

1.3.2 Business application

Business applications use multi-dimensional functions for the construction of business models. Business models for trade flows in a company, risk management policies and annual turnover are used for analyzing an organization's performance over a period of time. The ability to identify deficiencies in the company's policies by analyzing the functions in business models is instrumental in the development of better policies and strategies for the company's growth.

An example is the relationship between the quantity of manufactured goods and the quantity of inputs, modeled using production functions. A simple example of a production function is the Cobb Douglas function [Weins02]. A three factor Cobb-Douglas production function is

$$q = f\,(L,K,M) = A * L^{\alpha} * K^{\beta} * M^{\gamma}.$$

where, L is the labor, K is the capital required, M is the amount of materials and supplies and q is the product. α, β, and γ are the parameters of the function estimated using regression analysis. Multi-dimensional functions such as the Cobb-Douglas function needs to be optimized for maximizing the yield of goods with minimum wasteful expenditures.

1.4 Multi-dimensional relations and multi-dimensional functions

A subtle difference exists between multi-dimensional functions and multi-dimensional relations in terms of the distribution of data points. It's important to distinguish between the two since the visualization technique described in this thesis is best suited for functions.

Multi-dimensional relational spaces contain data points spanning a portion of the space. For example Table 1.1 [Goel99] shows data points of parameters for designing an aircraft.

Length	Breadth	Depth	Draft	Speed	Blending Coefficient	Objective Function
300	43	17.65634	11.04311	17.463396	0.214789	0.001055811
300	42.999996	17.829277	10.924515	17.469976	0.000014	0.001054124
300	43	17.635946	11.065382	17.514271	0.242642	0.001056084
300	43	17.766165	10.97506	17.699734	0.079081	0.001054747
300	42.999996	17.792397	10.960772	17.518015	0.051263	0.001054547
300	43	17.440191	11.225469	17.287531	0.522346	0.001059768
300	42.999996	17.767805	10.969849	17.484047	0.078807	0.001054697
299.99997	43	17.612064	11.081265	17.495827	0.275017	0.001056397
299.99997	42.999996	17.448832	11.246042	17.58687	0.533355	0.001059512
300	43	17.836496	10.921103	17.377964	0	0.001054232
300	43	17.556271	11.134746	17.518227	0.359101	0.001057343
300	43	17.840027	10.930996	17.488449	0	0.001054231
300	43	17.60931	11.089222	17.557772	0.283246	0.001056472
299.99963	43	17.834534	10.928561	17.471003	0	0.001054191
300	43	17.828184	10.936685	17.477894	0.00747	0.001054273
299.99997	42.999996	17.830067	10.931367	17.471189	0.00401	0.001054217
300	43	17.621899	11.076528	17.527895	0.262771	0.001056269
300	42.999992	17.540123	11.155281	17.651192	0.392947	0.001057678
300	42.999992	17.507853	11.179898	17.556587	0.437394	0.001058226
300	43	17.648663	11.055993	17.53241	0.228463	0.001055932
299.99997	43	17.355652	11.394067	17.689247	0.741529	0.001062791

Table 1.1: Multi-dimensional relations for an aircraft design

The values assumed by "length" parameter vary between 275 and 350; however the mode is around 300. Very few values occur in the remaining regions of the space. The same is the case for the "breadth" and "depth" parameters whose values are predominantly around 43 and 11 respectively. Values of the objective function are consequently defined only for certain values of the parameters.

In the case of multi-dimensional functions the functional value exists at all points, for all combinations of the values of the variables in the space. For example Table 1.2 shows the values of the variables x_1,x_2,x_3,x_4,x_5 and the corresponding functional value. Parameters x_1-x_5 assume discrete values between 1 and 5. From the table we see that the functional value exists for all possible values of the parameters. In essence the data-density distribution is uniform and dense in the case of functional data spaces as opposed to a non-uniform and sparse distribution in the case of multi-dimensional data spaces.

x1	x2	x3	x4	x5	Functional Value
1	1	1	1	1	-0.45941823
1	1	1	1	2	-0.45941823
1	1	1	1	3	-0.45941823
1	1	1	1	4	-0.45941823
1	1	1	1	5	-0.45941823
1	1	1	2	1	-0.45941823
1	1	1	2	2	-0.45941823
1	1	1	2	3	-0.45941823
1	1	1	2	4	-0.45941823
1	1	1	2	5	0.95286434
1	1	1	3	1	0.95286434
1	1	1	3	2	0.95286434
1	1	1	3	3	0.95286434
1	1	1	3	4	0.95286434
1	1	1	3	5	0.95286434
1	1	1	4	1	0.95286434
1	1	1	4	2	0.95286434
1	1	1	4	3	0.95286434
1	1	1	4	4	1.3026203
1	1	1	4	5	1.3026203
1	1	1	5	1	1.3026203
1	1	1	5	2	1.3026203
1	1	1	5	3	1.3026203
1	1	1	5	4	1.3026203
1	1	1	5	5	1.3026203
1	1	2	1	1	1.3026203
1	1	2	1	2	1.3026203
1	1	2	1	3	1.0980021
1	1	2	1	4	1.0980021

Table 1.2: Multi-dimensional function

1.5 Visualization of 1-dimensional and 2-dimensional functions

It is easy to imagine functional spaces in one and two dimensions [Card99]. Several well-established visual techniques are also available for the visualization of low-dimensional spaces. The simplest and commonly used ones are graphs and plots which use cartesian or polar coordinate axes as frames of references for plotting data points [Fig.1.1]. While graphs and plots are two-dimensional representations, three-dimensional visuals can also be constructed using the principles of graphics to show the orientation of the data points in space.

Graphing techniques can be enhanced using attributes such as color, height, length, displacement of points and size. For example color maps and height fields map the functional values to the color and height attributes, respectively.

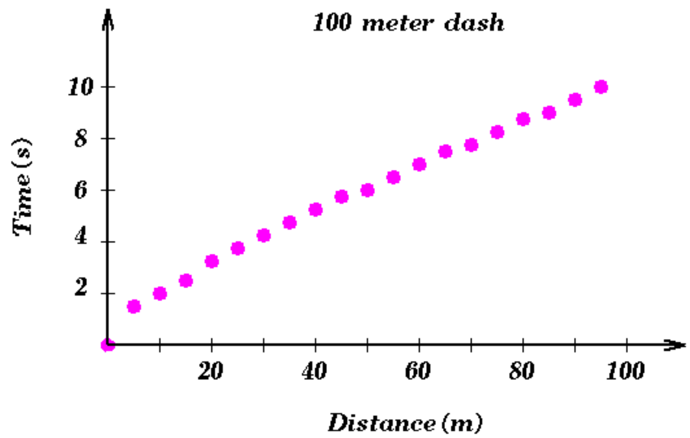
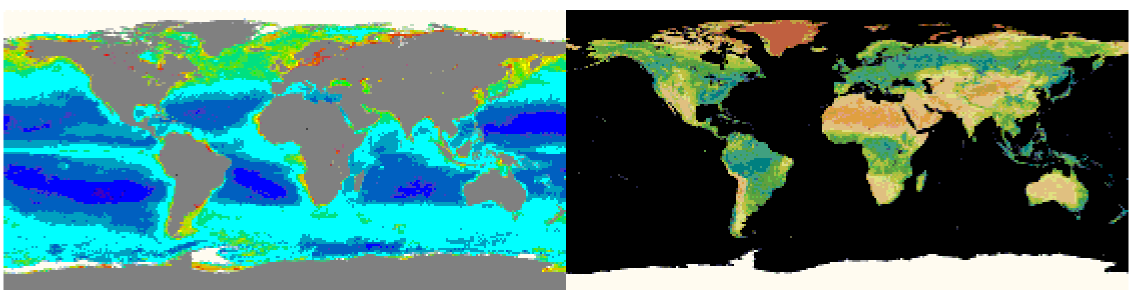
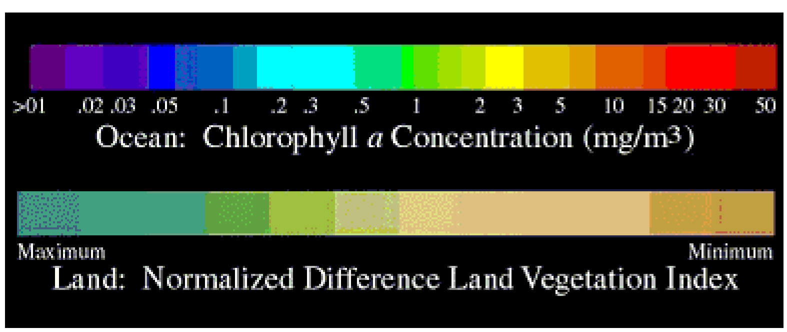


Figure 1.1: Cartesian plot for the one-dimensional function Time = f(Distance)

An example of a color map is shown in Fig.1.2. The mapping of the chlorophyll concentrations and land vegetation index are shown on the world map [Orbital02]. The VIBGYOR (rainbow) color scheme is used in the chlorophyll concentration mapping, with red representing high concentrations, and violet representing low concentrations.



(a) **(b)**



(c)

Figure 1.2: (a)Color Map of Ocean chlorophyll = f (Latitude , Longitude)
(b) Color Map of Land vegetation = f (Latitude , Longitude)
(c) Color scales

In height maps [Roettger98], the functional value is mapped to heights to create topographic contours [Fig 1.3].

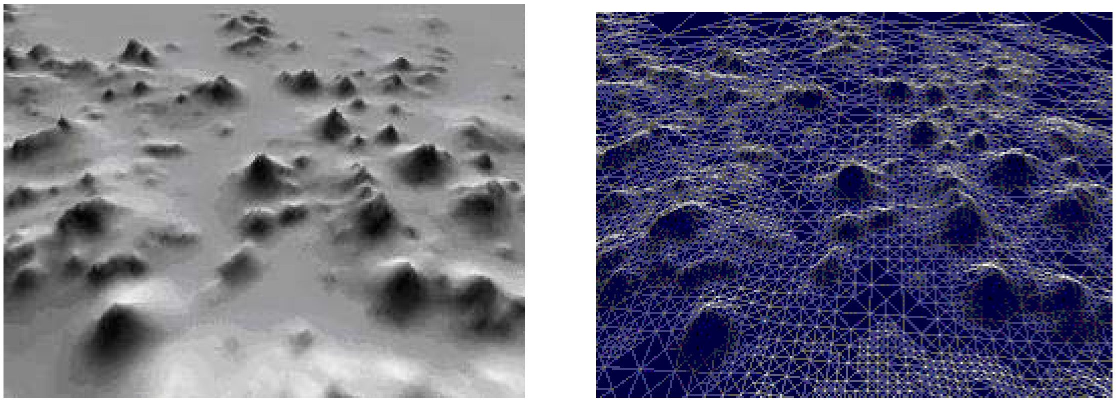


Figure 1.3: Example for 2-D functions represented as height fields

1.6 Motivation

Analysis of multi-dimensional functional spaces is an important and difficult problem in scientific applications. The difficulty arises because an increase in the number of dimensions (greater than 3) makes it difficult to grasp the nature of the space. Several mathematical and graphical approaches are available for performing tasks with multi-dimensional functions. Mathematical approaches help in computing specific values such as gradients and maximum or minimum values of the function. Graphical techniques provide an idea of the spatial orientation of functional values. Attributes such as color and size can be used for showing additional information. Both approaches simplify the problem by using the concept of projections to reduce the number of dimensions analyzed or seen at any point of time.

Projections, used by existing visualization approaches are not very helpful in understanding the nature of the whole space and its properties. Visualization of multi-dimensional functions in-terms of two and three-dimensional plots show only a limited portion of the space to the user. Users have to integrate several plots to obtain a holistic understanding of the space. Hence locating regions of interest becomes tedious and important regions could be missed in the process. The importance of the problem and the shortcomings of current approaches are the motivations for developing a new technique.

1.7 Problem

The problem is to develop a technique for visualizing multi-dimensional functions. The primary objective is to provide an idea of how the space looks like by helping users to answer preliminary questions like “Does the functional space contain predominantly negative or positive values?”, “Are the positive and negative values clustered or scattered in the space?". The overall idea of the space can be provided if all the dimensions are presented equally and simultaneously to the user. Equal treatment of dimensions also avoids the need to mentally integrate several partial views of the space. In the functional domain it's not uncommon to find a large number of dimensions supporting an infinite number of data points. Hence it's very important for the visualization approach to show promises of scalability. A focus into certain portions of the functional space within the global overview would be of interest to the users.

The objectives of the visualization described above can be succinctly summarized as four important goals

1) To obtain an overview of the functional space
2) Uniform treatment of all the dimensions
3) Ability to view the details of a local region within the global overview
4) Scalability

1.8 Challenges

The problem of visualizing multi-dimensional functions is very hard. The complexity is due to the multi-dimensional nature of the data and the presence of a large number of functional points in the space.

Humans live in a three-dimensional world and hence functions with two or three dimensions can be easily imagined. The function can also be graphed and shown in two dimensions. As the number of dimensions increases, it becomes increasingly difficult to imagine the space. Functions are essentially understood in terms of coordinate axes. The orientation of the coordinate axes needs to be visualized which is hard as the number of dimensions increases.

The other challenge is that functional data expressed concisely as mathematical formulae or as multi-dimensional arrays of sampled values contain a large number of data points. In the case of continuous functions the number of functional values is infinite and for discrete functions the number of values is given by $(\text{range per dimension})^{\text{dimension}}$. For example if we take a ten-dimensional space with the parameter values ranging between 1 and 100 discretely, the number of points being handled is approximately 100^{10}.

There are several problems associated with handling many data points. One of them is the storage of data. Computer memory (RAM) is certainly not sufficient and so values need to be stored in the hard disk with a certain number of functional values cached for faster retrieval. Fetching of values from the secondary storage is slower causing rendering delays for the visualization. Another issue is the aggregation of data points to provide a more realistic representation of the space. The greater the extent of aggregation, the greater will be the execution time for generating the visualization which would adversely affect interaction. Hence the algorithms for the visualization should use of minimal number of computations.

1.9 Present approach

The present approach is based upon the concepts of radial arrangement and focus+context. The user selects a point from the vast parameter space called the focal point. The focal point constitutes the point of interest in the space and serves as a starting point for the exploration of the space`. Rays called "focal rays" emanate from the focal point in all directions to the boundaries of the functional space. The focal rays contain functional data points. The focal point is mapped onto the center of the visualization with the focal rays arranged radially around it. The degree of detail decreases as we move away from the focal point toward the edges of the visualization. Thus more details are shown around the focal point (which is the point of interest in the space) without losing the contextual information, in accordance with the focus+context technique. The functional values are color coded with shades of green representing positive function values, and shades of red representing negative function values.

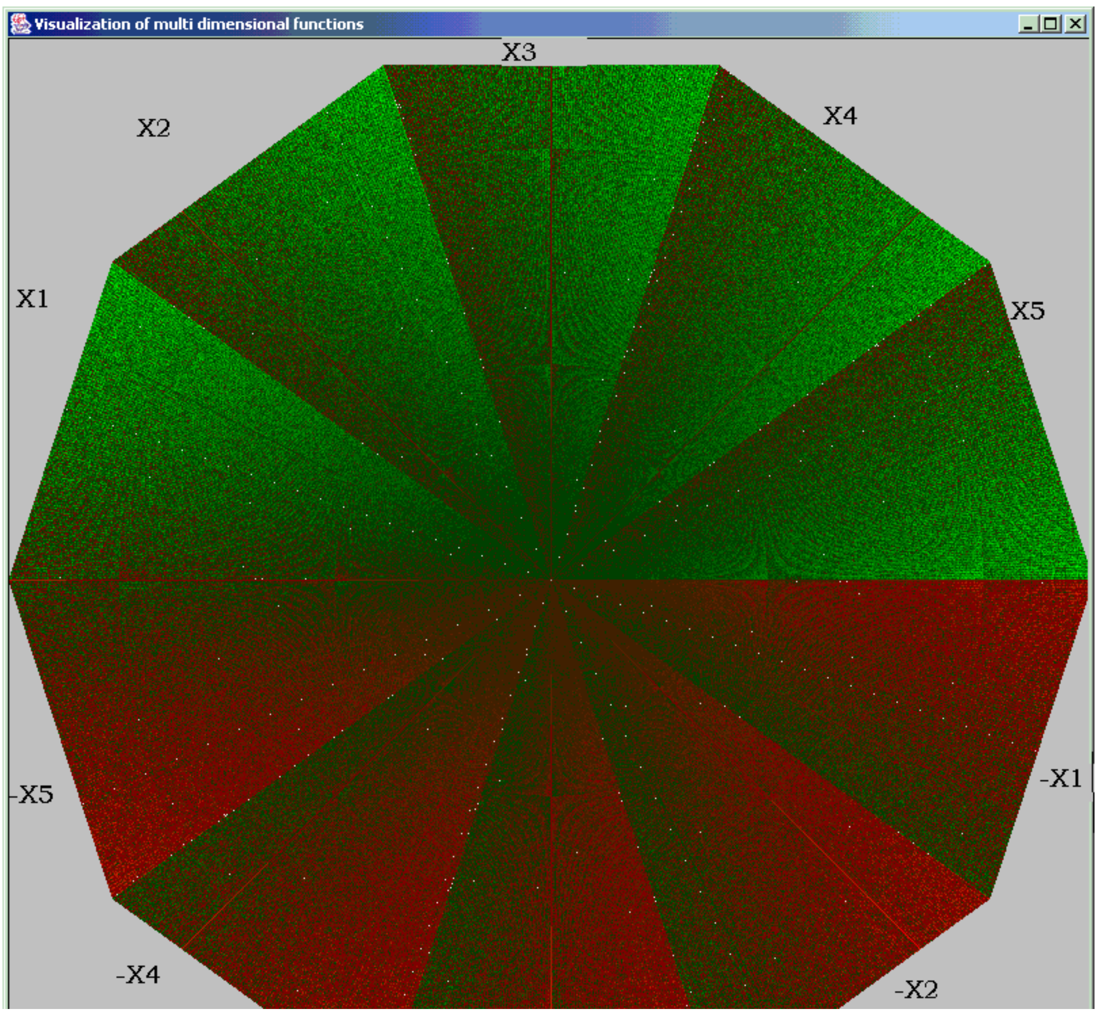


Figure 1.4: Visualization of the $f(x_1,x_2,x_3,x_4,x_5) = x_1+x_2+x_3+x_4+x_5$ function

A simple example is the $f(x_1,x_2,x_3,x_4,x_5) = x_1+x_2+x_3+x_4+x_5$ function [Fig.1.4]. The visualization shows the functional behavior for the entire parameter range from –50 to 50. Every triangular slice shows the trend of the function for a gradual increase in value of a parameter. The functional value also increases gradually as seen from the change in color from red to green. The same trend can be seen for all other parameters in all the slices.

1.10 Overview of thesis

This thesis is divided into eight chapters. Chapter 1 introduces the concept of multi-dimensional functions, the problem being approached, the difference between multi-dimensional functions and relations, involved challenges and a brief description of the approach. Chapter 2 covers related work and the insights gained from these approaches. The concepts behind the technique developed for visualizing multi-dimensional functions are discussed in Chapter 3. Chapter 4 details the algorithm and discusses software related issues. Chapter 5 deals with an investigation of the applications of the visualization technique. Feedback from users is important for improvement and has been obtained from formative evaluation studies. The process and the results are described in Chapter 6. Finally, in Chapter 7 we conclude our findings and outline some future work.

Chapter 2: RELATED WORK

2.1 Introduction

Analyzing multi-dimensional functions presents an important and interesting challenge to researchers. Several mathematical and visualization solutions have been put forth. Mathematical approaches deal with the computation of statistics to reveal correlations between dimensions. Visualization approaches provide a graphical representation of the functional space. Users can interact with the visualization to identify functional properties and patterns. A cross section of the different approaches is presented in this chapter.

2.2 Visualization approaches

Information visualization is an innovative approach aimed at providing insight into data through visual means. Information visualization exploits the fact that human vision has the highest bandwidth. Novel and effective representations and metaphors are developed, coupled with interactivity to allow us to perceive more than what can be realized from static tables and datasets.

Users need to have an idea of the mathematical approaches for using them. For example prior mathematical knowledge of the techniques is required for tailoring the inputs and for the interpretation of results. Another limitation is that they assume that the data is valid. Erroneous inputs lead to erroneous outputs. In contrast visualization techniques are not based on such assumptions. However mathematical approaches can be combined with visual techniques as in "Multivariate Visualization Using Metric Scaling" [Wong97] and "Animating Multi-dimensional Scaling To Visualize N-Dimensional Data Sets"[Bentley96]. The next section elaborates the various visualization approaches for multi-dimensional function spaces.

2.2.1 Approaches for visualization of multi-dimensional functions

2.2.1.1 Worlds Within Worlds

"Worlds Within Worlds" [Feiner90] makes use of a 3D visualization technique. Every 3D space is termed a world. The final visualization is comprised of 3D worlds embedded within another 3D world thereby making use of a nested coordinate system. Only the innermost world is visible to the user by default with the outer world dimensions remaining a constant. The constant values used for the outer world dimension is based on the location of the origin of the inner world in the outer world. By moving the origin of the inner world in the outer world, different slices can be seen. Outer worlds can also be viewed and explored by means of the interaction techniques provided.

The visualization of the inner world consists of essentially lines graphs and surface plots [Fig 2.1]. Line graphs present the functional value for the variation of a single parameter value. Surface plots show the functional value for the variation of two or three parameters. The idea behind this is that graphs can be understood easily by users. Novel schemes to map information such as different color schemes and the like are therefore avoided. The plots and graphs show the variation of the function when a certain number of parameters are varied with the others (typically the parameters in the outer worlds) remaining constant and are displayed to the user in the innermost world.

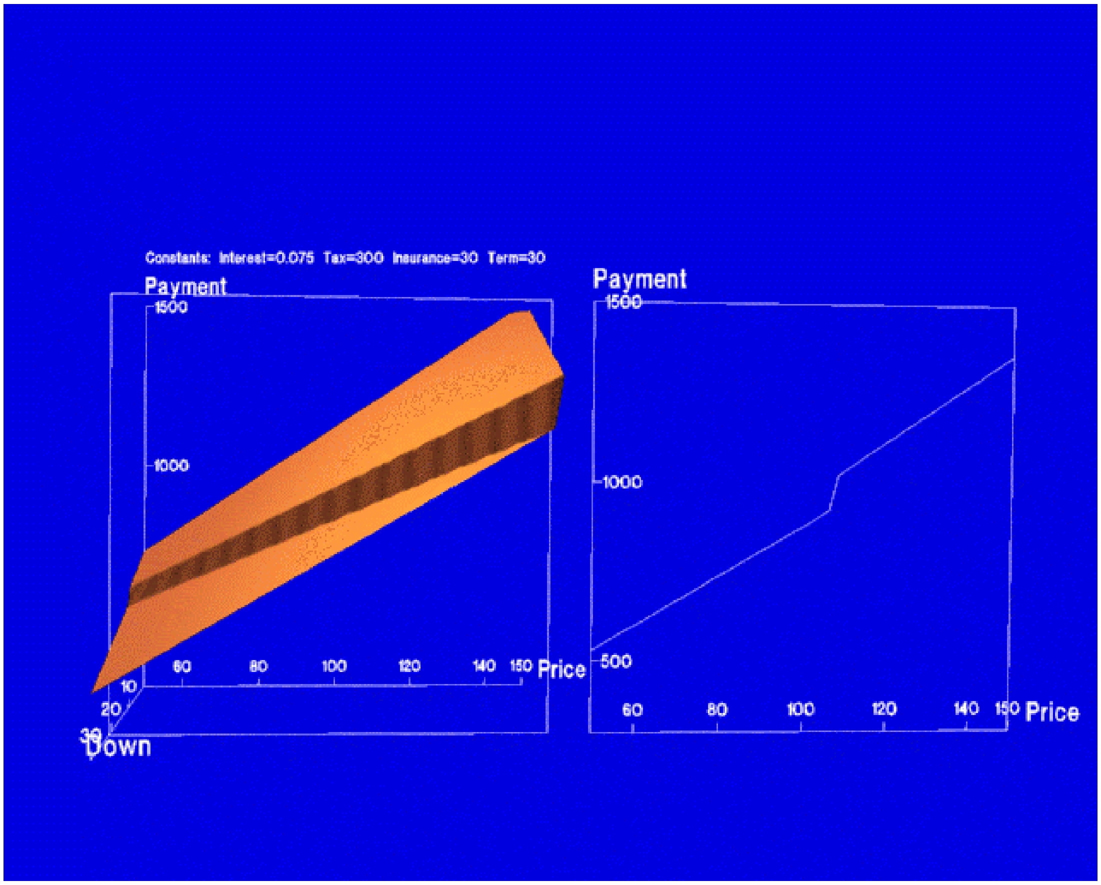


Figure 2.1: Visualization of the inner world consists of line graphs and surface plots

The exploration, directed search and comparison operators are provided for interacting with the visualization. The exploration operator aids in understanding the space by showing the portion of the space of interest to the user [Fig 2.2]. Directed search is provided to help the users refine their search and identify specific regions satisfying their constraints in the context of the space. The comparison task helps in comparing plots with different variables being varied in the nested world.

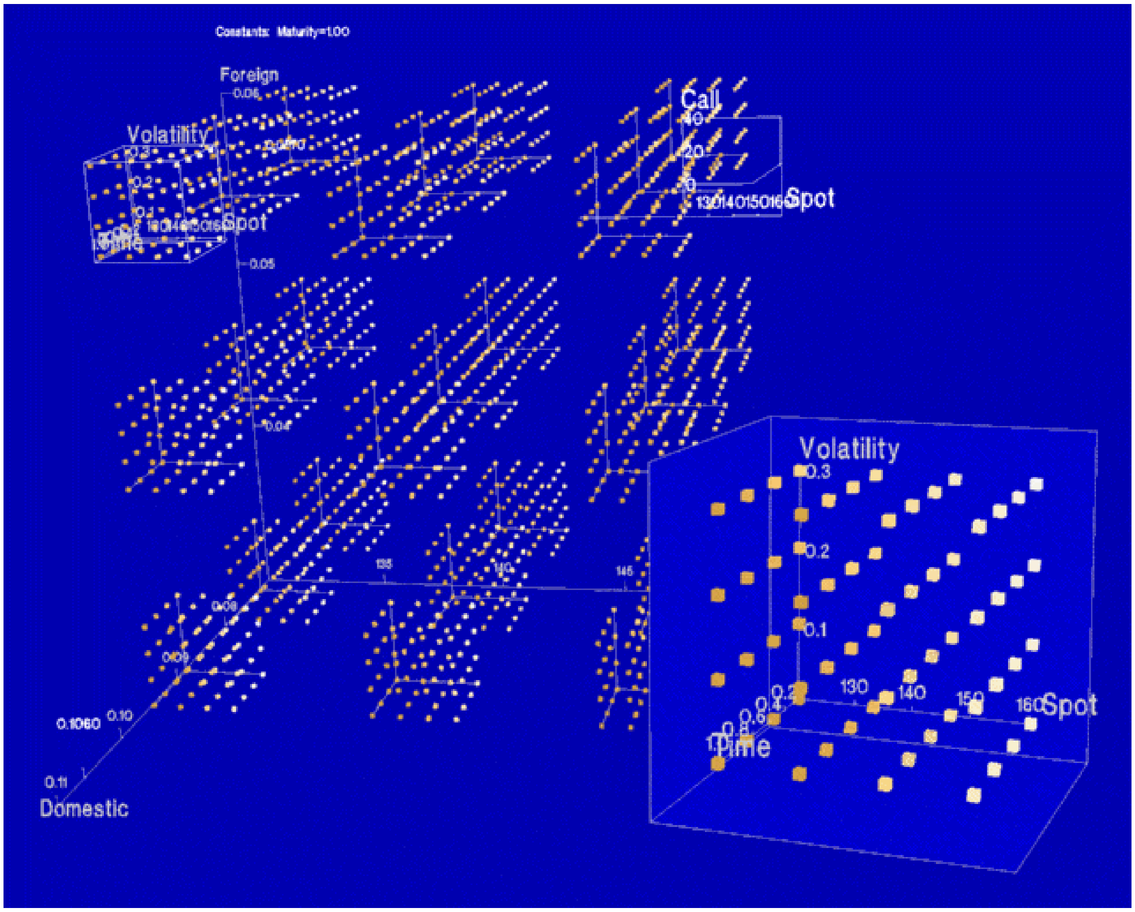


Figure 2.2: Visualization for exploring the functional space. One of the worlds from the overview is magnified using the zoom tool.

The major advantage of this approach is that only a 3D world is shown to the user. Hence, the user is not overwhelmed with a large amount of data. The visualization consists of graphs are easy to understand and interpret. The interaction techniques provided help the user in searching, exploring and comparing the worlds and have been implemented in a very powerful manner.

However, the approach has shortcomings. One of the major impairments is limited scalability. As the number of dimensions increase the number of worlds also increases which translates to increased nesting and lesser clarity. Consequently, many of the worlds are occluded. Functions with fewer than six dimensions (single degree of nesting), which is the case when a two-dimensional world is nested within a three-dimensional world, can be visualized well with Worlds Within Worlds. The range per dimension supported is not limited since scales of the plots can be varied accordingly to accommodate wide parameter ranges. Hence nesting of dimensions may not be preferred for problems which require he user to answer questions like

- What is the nature of the multi-dimensional space?
- What is the best fit for a set of parameters (these parameters may be in different worlds)?

Since a 3D approach is used it may take more time to render these images especially when the number of dimensions increases. Consequently interaction time increases. Equipment which works well for 3D renderings may be required for the visualization to be effective.

This approach makes it difficult to explore and understand the effect of variation of parameters placed in different worlds due to the lack of an overview of the functional space. The dimensions are ordered from the inner to the outer worlds. Hence inner worlds are visible to the user, providing unequal treatment of dimensions. Labeling adds to the clutter on the screen when the number of dimensions increases. When the user wants to zoom into a portion of the space, the zoomed space is placed alongside the main visualization [Fig.2.3]. Such interactive features are not blended with the visualization. This occupies the visualization space and when many parameters need to be compared the number of such windows also increases causing more confusion to the user.

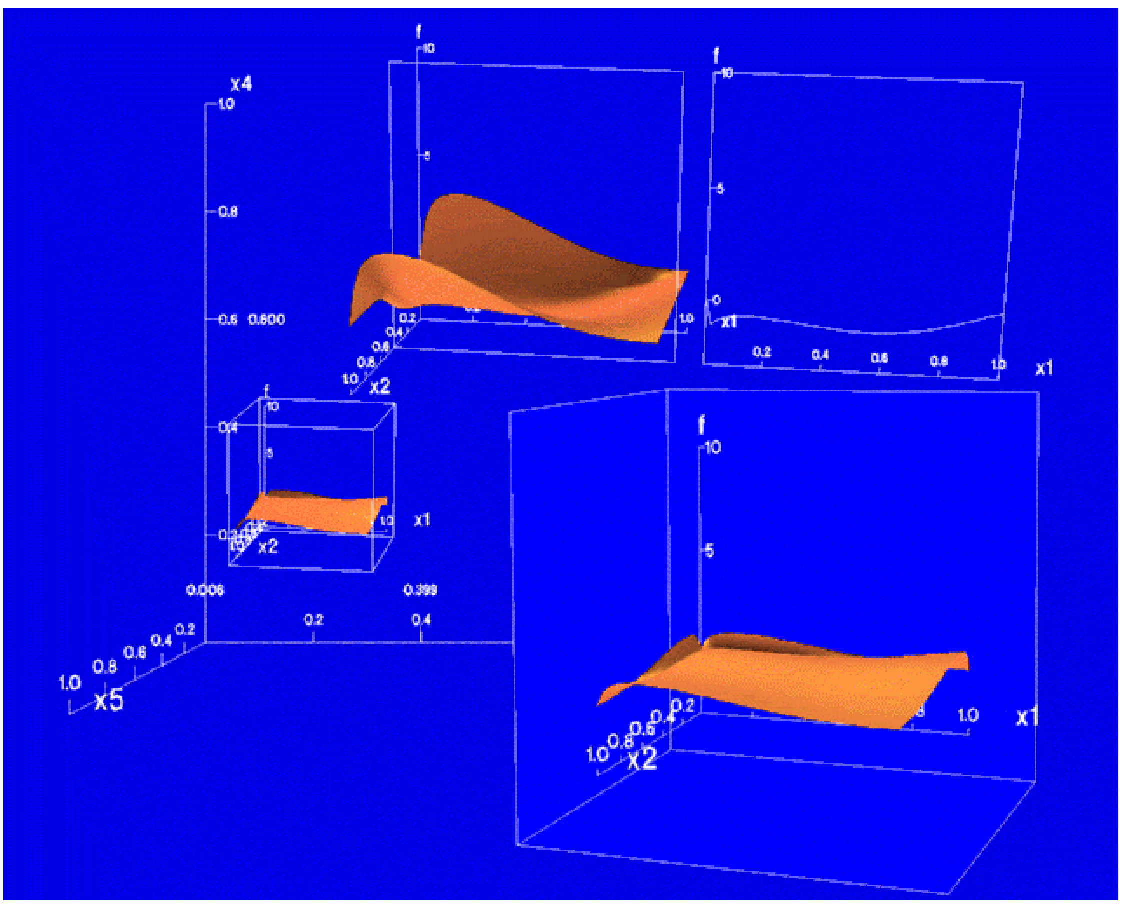


Figure 2.3: Line graph and Zoom tool in use in a world

In summary, this approach is good for exploring spaces with fewer than 6 dimensions. It is more suitable for applications where although there are a large number of parameters, most of them need to be fixed due to user specified constraints. An example is a function for finding the mortgage rates for houses in the United States. The mortgage rates when certain parameters such as location, down payment are fixed, with the remaining parameters varying is of interest to the user. The approach may be expensive if specialized devices are required for its effective exploitation.

2.2.1.2 HyperSlice

The HyperSlice [VanWijk93] approach uses a simple technique of representing the multi-dimensional function as a matrix of orthogonal two-dimensional slices. The matrix shows two-dimensional slices for all pairs of dimensions [Fig 2.4]. Each two- dimensional slice represents the variation of the function when two of the parameters are varied, the rest of the parameters remaining constant. Here the functional values are computed at runtime (computational steering). One of the major ideas used in this visualization is the choice of two-dimensional slices rather than 3D slices as in Worlds Within Worlds because two-dimensional slices are easier to interpret and their rendering takes less time.

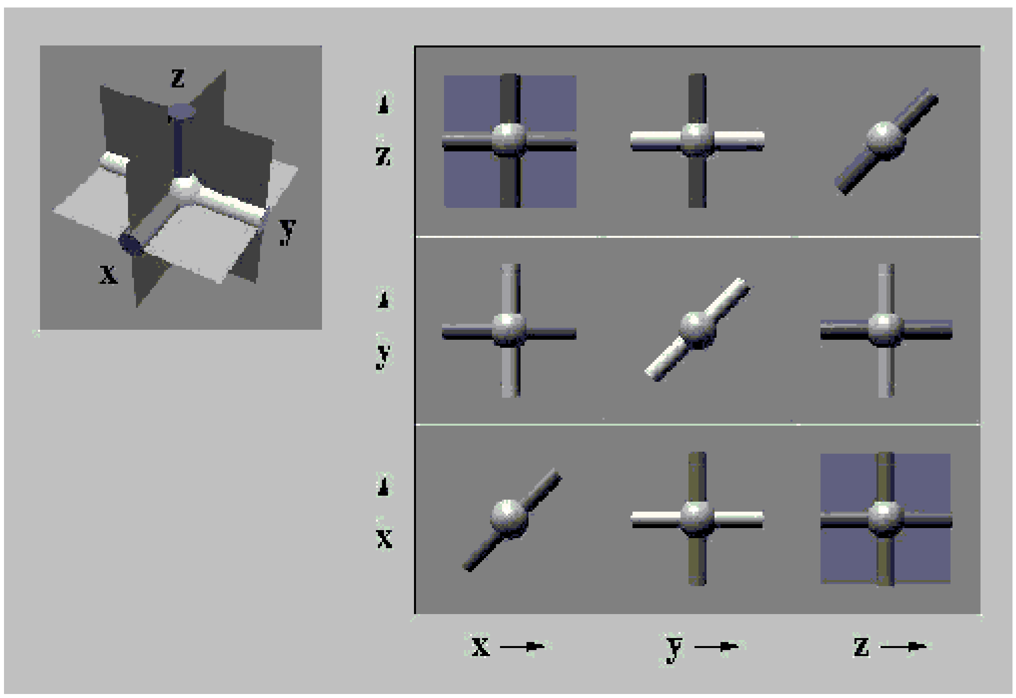


Figure 2.4: Concept of HyperSlice

Automated traversal of the data space is possible through direct interaction manipulation techniques. Every slice displays the function at a user-defined focus which can be altered using the “translation action” to view new regions [Fig 2.5]. The ranges of the parameters in every slice can be altered (scaling effect) to zoom in or zoom out of the space. Rotation around a point is also supported. The above three features are with respect to a point in space.

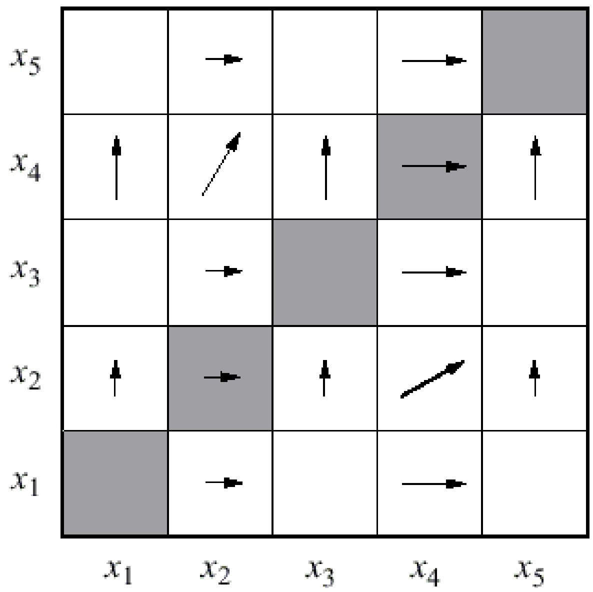


Figure 2.5: Effects of a translating action achieved by dragging a slice

Another interesting set of features which are concerned with a group of points are paths, gradient paths and contours. Paths comprise interesting points defined by the user to enable him to quickly traverse the space [Fig 2.6]. Users can revert to these paths when exploring other parts of the hyperspace for comparison purposes. Additional features to find maxima in the hyperspace (gradient paths) and the indication of all points with a chosen functional value (contouring) are valuable features for exploring and understanding the hyperspace.

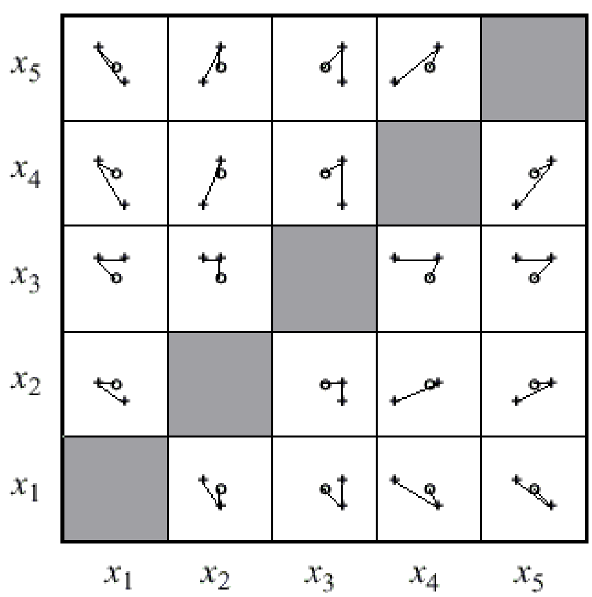


Figure 2.6: User defined path

One of the main advantages of this approach is that the visualization is very compact and hence the available space is utilized to a maximum. All the dimensions are treated equally and hence all the dimensions are seen in the visualization. This makes it easier to compare parameters. The concepts of paths, gradient paths and contouring are essential and novel features. HyperSlice shows improved scalability with respect to Worlds Within Worlds. As the number of dimensions increases the number of scatter plots increases resulting in an increase in the size of the matrix. The size of the matrix, limited by the visualization screen width, is given by

$$\text{Screen width} = \text{Size of matrix} = \text{dimension}^{\text{range of dimension}}$$

Hence functions with 10 parameters can be supported by HyperSlice. Parameter ranges can be accommodated by scaling the plots. However as the number of dimensions increases the matrix size increases affecting the range of the parameters that can be accommodate in the matrix.

The main shortcoming of the approach is that visualization of the space in terms of two-dimensional slices does not provide a good overview. Users need to mentally integrate the slices to obtain an overall picture of the functional space. It also makes it hard for the user to know where to slice to see interesting patterns in the function space.

For certain problems, viewing the function in terms of slices is not very convenient. Slices show the variation of the function with only two parameters. It would be useful if variation of the function with more parameters can be seen in the same slice so that variables which strongly influence the function and such other properties of the space can be identified. Since the plots are arranged symmetrically in a matrix arrangement there would be repetition of plots, which consumes visualization space unnecessarily.

A variance of the HyperSlice approach is the Hyperbox [Alpren91]. The Hyperbox is a two-dimensional depiction of an n-dimensional box [Fig.2.7]. The faces or panels of the box consist of two-dimensional projections unlike Hyperslice where the two-dimensional projections are arranged in a matrix like format. The dimensions can be sliced to show histograms on the panels. This approach also suffers from the drawbacks of projections.

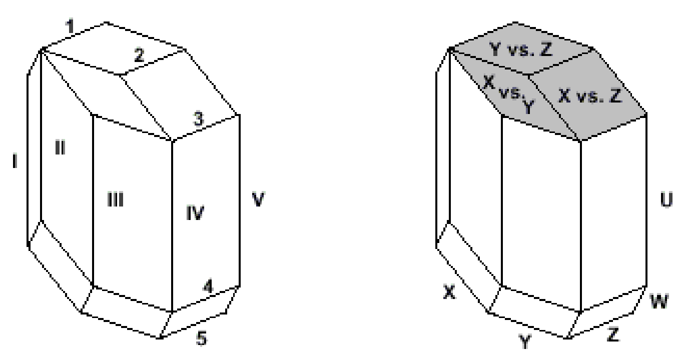


Figure 2.7: Concept of hyperbox

In summary, HyperSlice approach is extremely powerful. It also shows increased scalability when compared to the Worlds within Worlds and provides a number of powerful interaction features for the user.

2.2.1.3 Mihalisin's Approach

This approach uses the concept of hierarchical axes [Mihalisin91]. Each independent variable is sampled regularly in a grid or lattice like fashion and the functional values are computed at these points. All the independent variables and their values are shown in the visualization as well as their corresponding functional values [Fig 2.8(a)]. The visualization of the function $x^2 + u^2 + y^2 + v^2$ is shown in Fig.2.8(b). Additional tools such as the subspace zoom tool and the animation tools are provided to explore and understand the functional space.

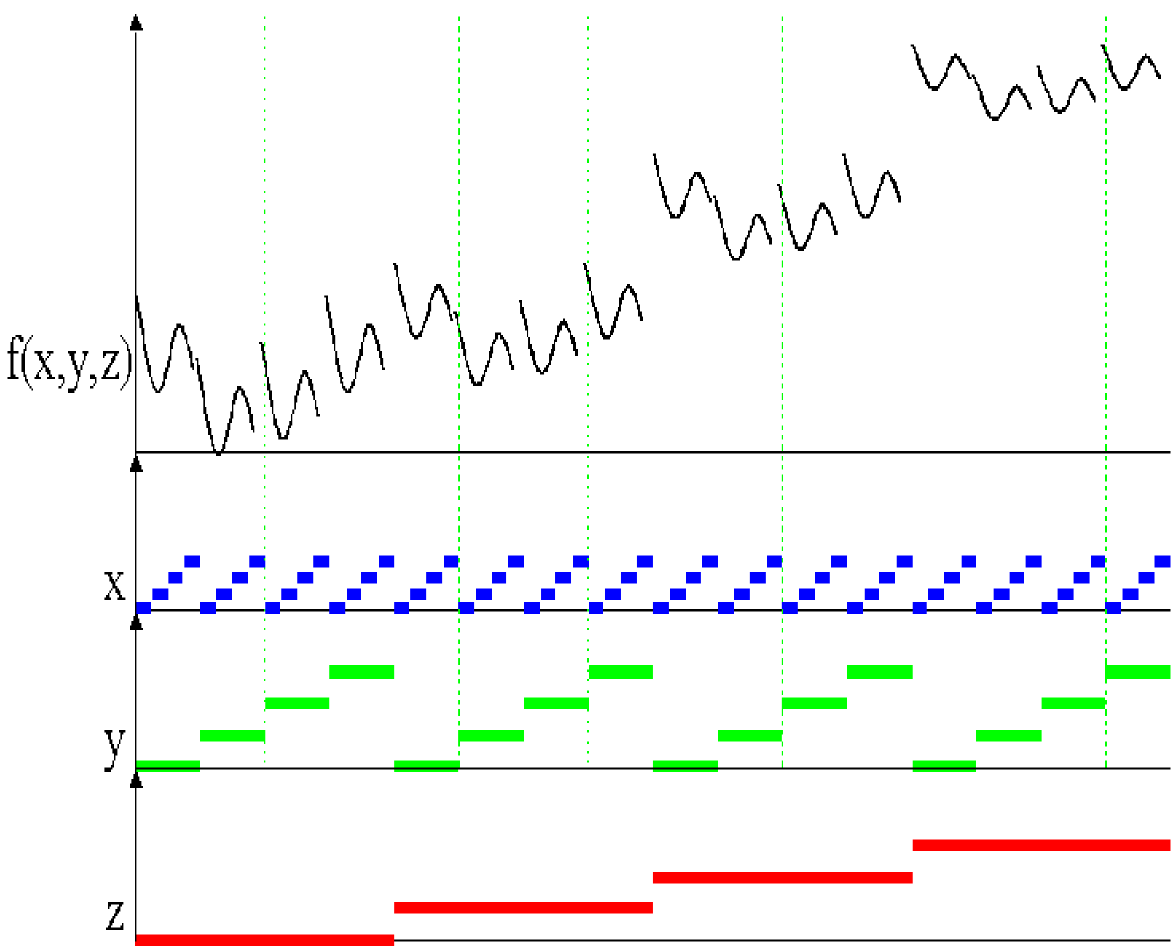


Figure 2.8(a): The concept of hierarchical coordinates

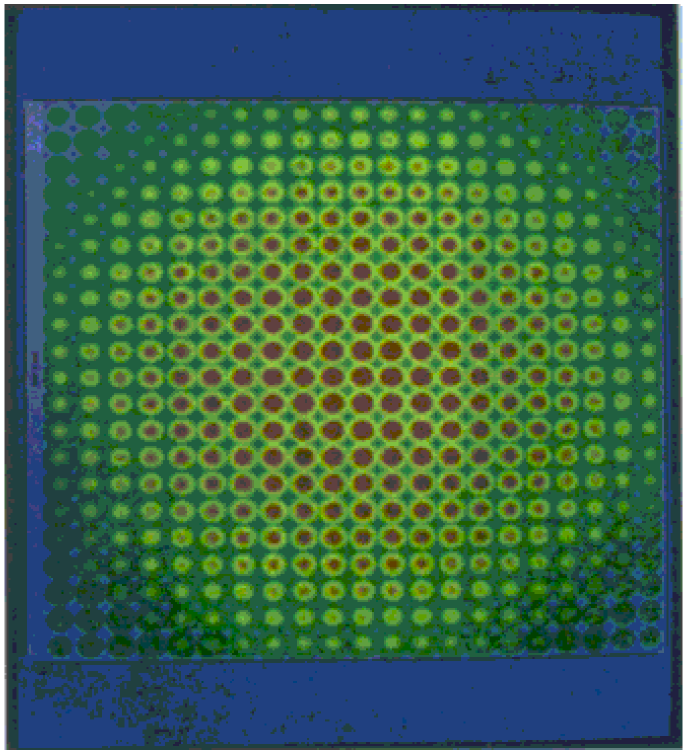


Figure 2.8(b): Visualization of $x^2 + u^2 + y^2 + v^2$

The advantages of this approach are that the values of the independent and dependent variable are shown in the visualization in a very orderly fashion. This helps the user to form an overall idea about the functional space. Details are seen only on demand using the zooming and animation tools. Hence, the approach is very effective in locating maxima and minima.

The drawback of this method is the limited scalability because function values for all combinations of independent variables are shown. The size of the visualization is given by

$$\text{Size of visualization} = \text{Screen width} = \text{dimension}^{\text{range of dimension}}$$

The number of dimensions and the range per dimension affect the size of the visualization. Hence functions with 6 parameters can be supported. The number of dimensions interferes with the scalability of range per dimension. Increase in the range per dimensions results in increased sampling which cannot be accommodated when the number of dimensions increases.

Another major problem is that the resulting visualization is lossy due to the sampling of data. Also if the range of the parameters is very large then the visualization becomes very confusing and it would be difficult to show all the values of the parameters to the user.

Identifying patterns in the space is subject to the order in which the coordinates are nested. For the purpose a permutation tool is also provided. However it is difficult for the user to speculate about the order of nesting the coordinates especially when analyzing functional spaces whose nature is unknown.

In short, this solution is very suitable for functions with a small number of parameters where the range per parameter is also small and constrained to integral values. Applications requiring identification of specific parameter values are well suited for this approach.

2.2.1.4 Projections approach (VMFP) from Multistat Inc.

In this approach the visualization consists of two or three-dimensional graphs obtained by calculating functional values at selected points. The selected points constitute a uniformly distributed sequence (UDS) [Sevastyanov01]. Identification of points for the UDS is achieved by dividing the range of the parameters into n equal parts and selecting the center point of each part. The functional values are computed at the points in the UDS and the resulting data set is divided into two or more non-intersecting subsets using a splitting criterion. Each subset is then color encoded to obtain 2D or 3D graphs [Fig. 2.9]. Features such as brushing and linking [Becker87] are provided such that selection of points in a certain region of space are colored similarly and highlighted in the other charts as well.

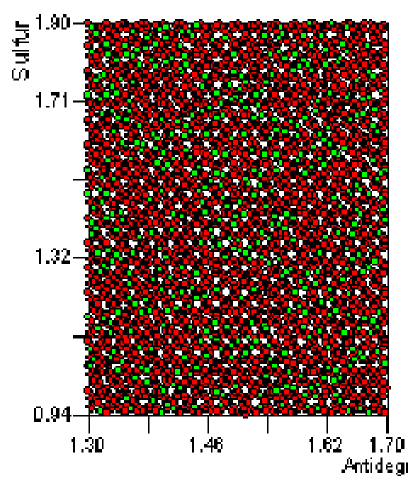


Figure 2.9: The plot between two parameters Sulphur and Antidegradant

It is hard to evaluate the success of this approach because of insufficient information. Choice of splitting criterion is important otherwise patterns cannot be seen. It is also not clear how splitting criteria are selected and whether they are fixed or different depending on the type of the data sets.

Aggregation has not been employed due to pre-calculation of functional values. Hence the resulting visualization may not reveal patterns which occur between selected points in the UDS. Pre-calculation has the demerit of memory requirements. Hence for real valued function spaces with a wide parameter range, the approach becomes comparatively expensive.

Their description suggests different color codes depending on the data set, which may confuse users. A richer set of features may be provided.

The approach is not a very novel technique since it is founded on showing simple graphical projections between variables. It is more suitable when the functional value is dependent on a few parameters.

2.2.1.5 Comparative Evaluation

A comparative evaluation of the above approaches is presented in Table 2.1.

	Worlds Within Worlds	**HyperSlice**	**Mihalisin**
Overview of the space	- due to slicing	- due to slicing	+ due to sampling of the entire space
Uniform treatment of dimensions	- due to nesting of coordinates	+ all dimension shown equally in matrix fashion	- due to nesting of coordinates
Scalability	- ~5D one nested world(2D) within another(3D)	+ ~10D matrix can be big	- ~6D

Table 2.1: Comparative evaluation of the techniques for visualizing multi-dimensional functions

2.2.2 Approaches for visualization of multi-dimensional relations

The visualizations described in this section are well suited for understanding and analyzing multi-dimensional relations (described in Chapter 1) rather than functional data. This is due to functional spaces containing a large number of functional points uniformly distributed throughout the space. Hence, the resulting visualization becomes densely populated and cluttered with significant occlusion of data points.

2.2.2.1 Simple visual techniques

Graphs and charts are the simplest techniques to visualize multi-dimensional data spaces. The graphical approaches can be categorized into Cartesian and Polar coordinate based approaches depending on the type of coordinates used to plot the points. Line graphs use the Cartesian coordinates, where the functional value is plotted on the Y-axis for varying values of the dimensions along the X-axis. The extension of the line graph concept for multiple dimensions results in multiple line graphs [Grinstein01]. A multiple line graph is a single chart showing line graphs between the different parameters and the functional value. The line graphs for different dimensions can be distinguished with different colors and types (continuous, dashed) of lines [Fig. 2.10].

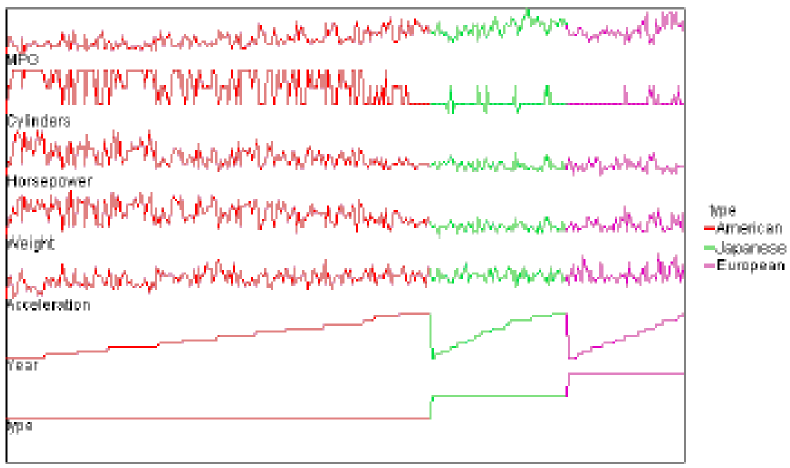


Figure 2.10: Multiple line graph of a classic car data set

Another type of Cartesian plot is the scatter plot which is a point projection of the data onto a 2D or 3D space. Polar charts are circular graphs displaying each point based on its polar coordinates [Fig 2.11] [Grinstein01]. It can be considered as a "wrap around" version of a line graph.

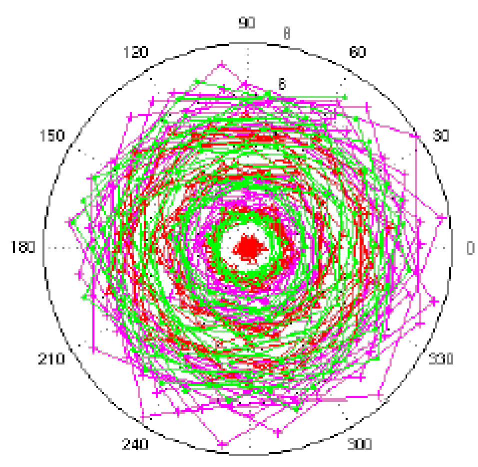


Fig 2.11.Polar chart for a flower data set

The major shortcoming of these approaches is the limited scalability. For example, in multiple line graphs, since the functional value axis is the same for all the graphs, the different line graphs can overlap as the number of parameters increase, causing confusion [Fig 2.12].

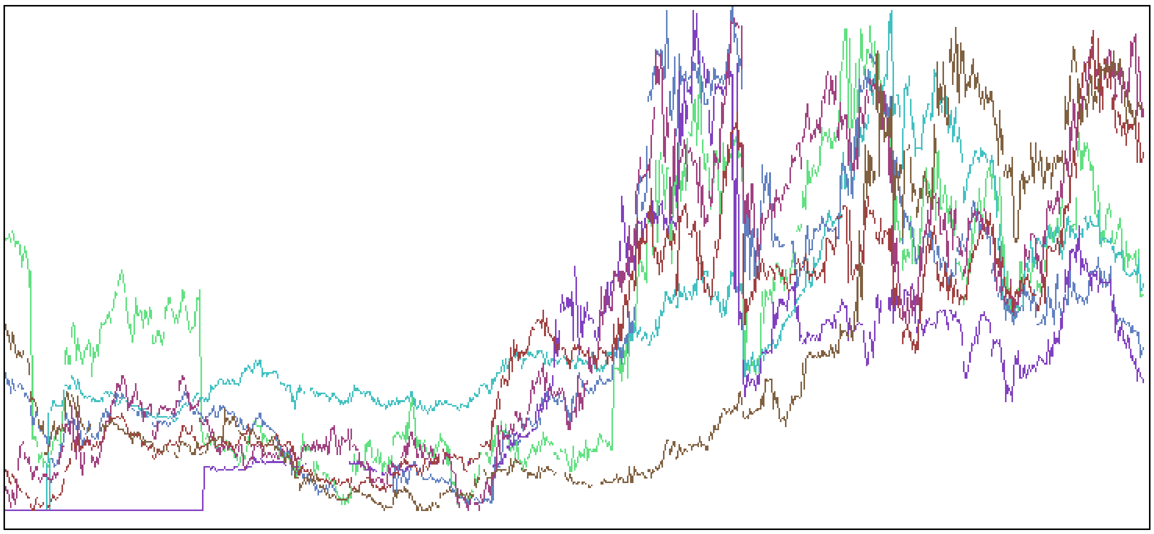


Figure 2.12: Visualization of a 7D data using line graphs. Adjacent line graphs overlap causing confusion.

2.2.2.2 Approaches using coordinate axes representations

Many approaches such as Parallel Coordinates, Star Coordinates and scatter plots are based on the representation of the coordinate axes in the visualization. The orientation of the coordinate axes and representation of data points with respect to the coordinate axes differs depending on the approach.

Scatter plots is a conventional approach used in SpotFire [Ahlberg95] where data points are represented on two-dimensional plots [Fig 2.13]. Plots with different pairs of dimensions are arranged in a matrix like format. Points of interest can be highlighted in any plot. Plots can be linked such that highlighted points in one of the plots can be highlighted in the others using the brushing and linking [Becker87] concept.

Parallel Coordinates [Inselberg97] consists of equally spaced parallel axes with varying scales on each axis. A data point is shown as a series of lines by connecting the points representing individual dimension values on the axes. Star Coordinates [Kandogan00] places the coordinate axes in a circular manner as opposed to parallel coordinates to conserve space [Fig 2.14]. The angles between the axes are equal. Data points are scaled to the length of the axis and represented. Scaling and rotational transformations are available to identify patterns in the data.

The data points become indiscernible when the approaches described in this section are applied to multi-dimensional function spaces. For example, in the case of Parallel

Coordinates the lines are too close to each other forming a thick thread of lines [Fig.2.16]. The SpotFire visualization shows a dense indiscernible blob of points [Fig.2.15].

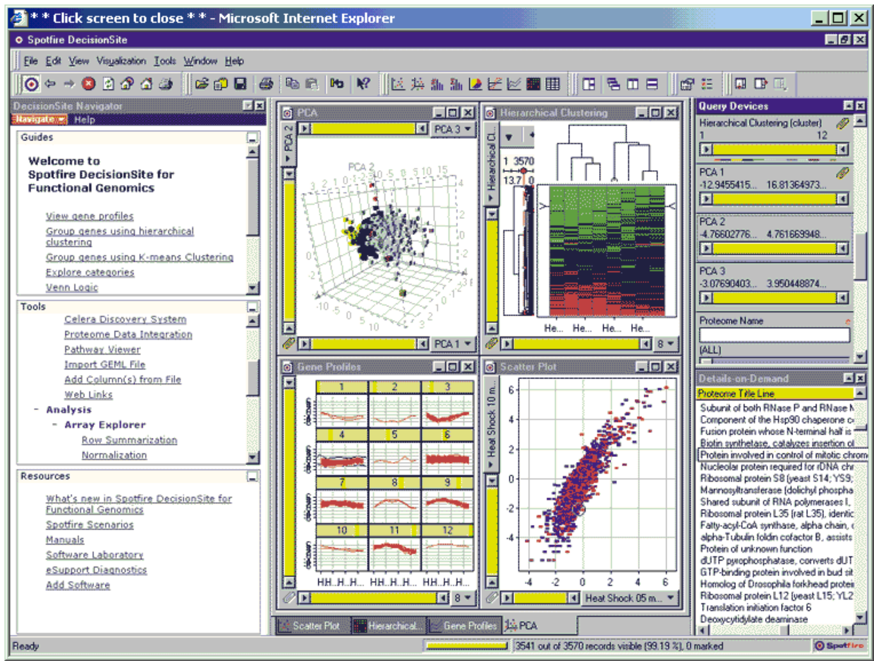


Figure 2.13: Scatter plot visualization of a genomic dataset [SpotFire02]

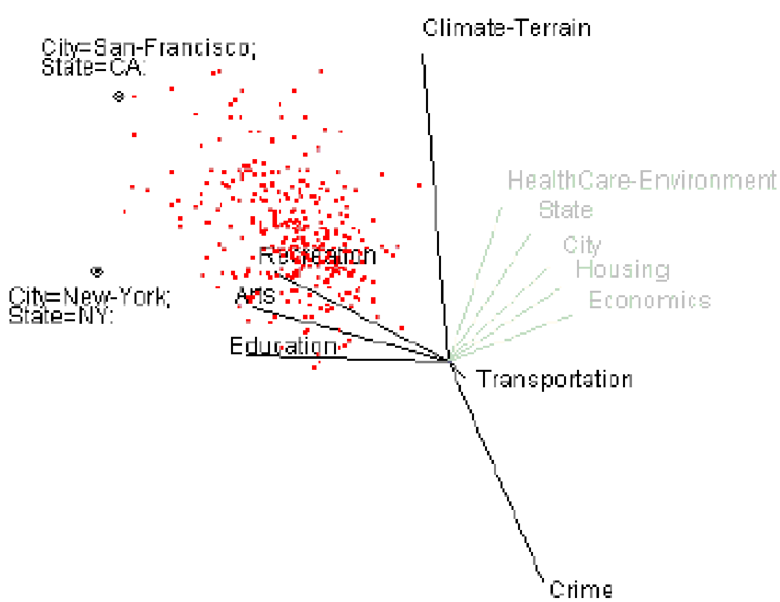


Figure 2.14: Star Coordinates visualization of a multi-dimensional relation

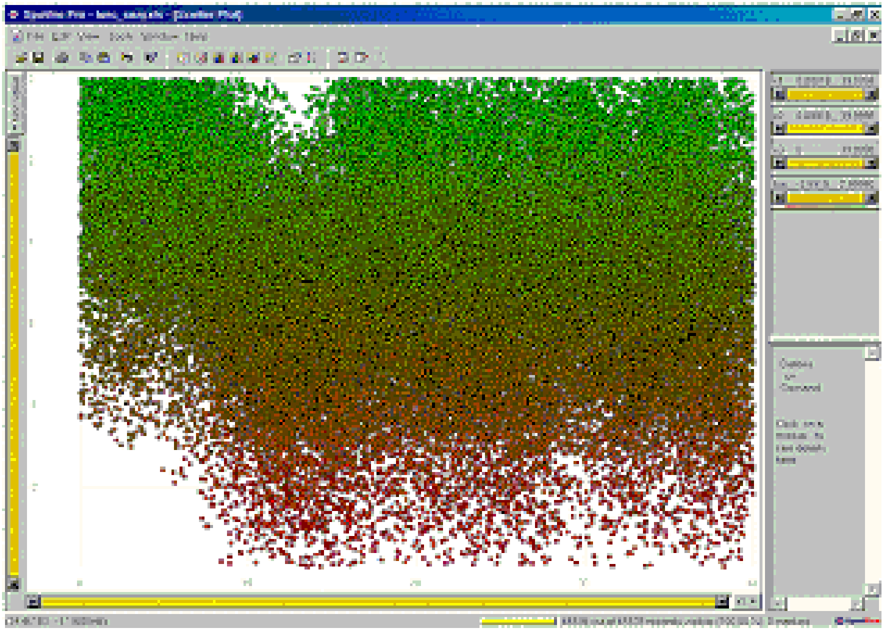


Fig 2.15: SpotFire visualization for $\cos(x^2) + \cos(y^2) + \cos(z^2)$. Variables x,y,z assume values from 0-40. The plot seen in the SpotFire is x vs. functional value. The functional values appear as a massive blob of points.

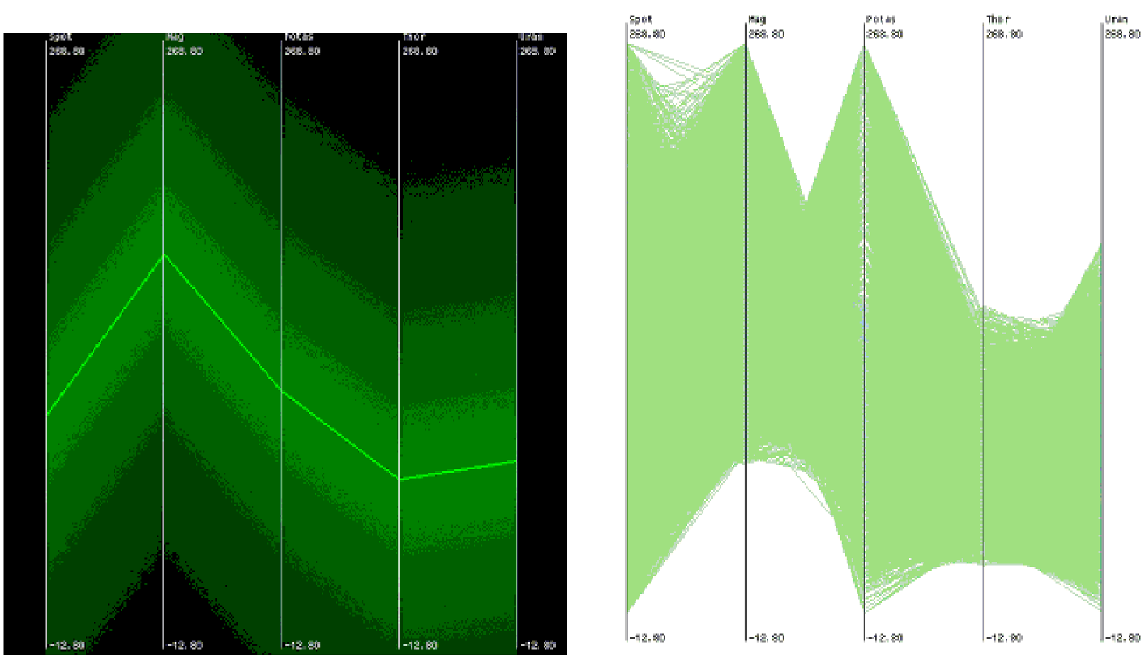


Figure 2.16: Clutter in Parallel Coordinates caused by visualizing multi-dimensional functions

2.2.2.3 Table Lens

Table Lens [Rao94] is a very different representation from the ones described above. Table Lens presents the data graphically in a spread sheet format providing a good overview of the data. Details are obtained on demand by focusing on the rows of interest. Table Lens is also not very effective, and a considerable amount of visualization space is wasted in representing the values of the independent variables which are not of interest [Fig 2.17].

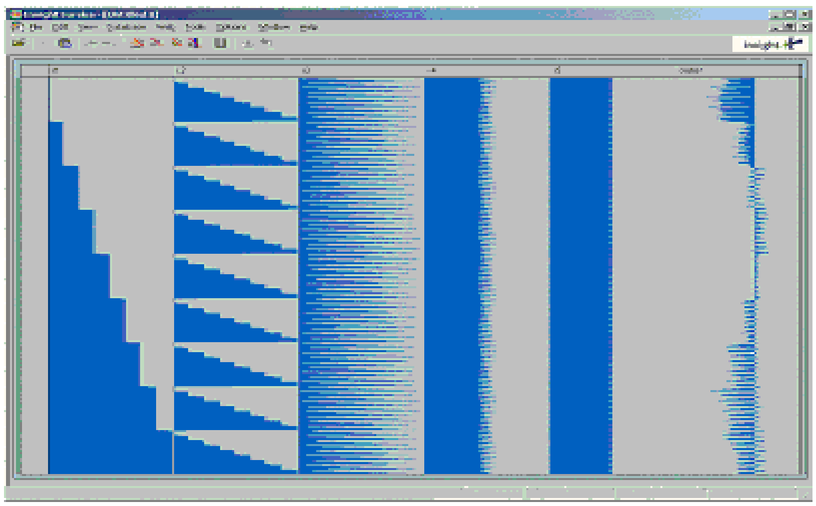


Figure 2.17: TableLens visualization of a 5-dimensional function with 9 samples per dimension. Columns 1-5 represent independent variables; column 6 represents the dependent variable.

2.2.2.4 Attribute Explorer

Attribute Explorer [Tweedie96] is another artifact for examining multi-attribute data. It supports the tasks of making choices from a population of multi-attribute items. Scales with histograms are assigned to every attribute of the data set. Users can interact with the scales by selecting values for particular attributes. Selecting values for one attribute highlights those items on the other scales. When selecting three or more attributes, values of a particular attribute can be locked. Once a subpopulation of items of interest has been identified individual lines running across all the attribute scales can be used for inspecting them.

Attribute Explorer is not well suited for the domain of multi-dimensional functions because histograms for single valued functional data are of no practical use. Interaction features do not provide for a good exploration of the space. For example, identification of maxima and minima is not well supported. The approach would not scale well, especially when there are several parameters with a very wide range of values.

2.2.2.5 RadViz

The RadViz display [Hoffman99] technique uses a non-linear mapping to map n-dimensional data points on to 2D space [Fig. 2.18]. The n points constituting dimensions are arranged around the perimeter of a circle with equal spacing between the points. A set of n springs are fixed to one end of each of these points with the other end attached to a puck. The spring constants are dependant on the magnitude of the data points. The data point is plotted at the equilibrium position of the springs when the puck is released. At the equilibrium position the sum of all the spring forces equals zero [ChungWong97].

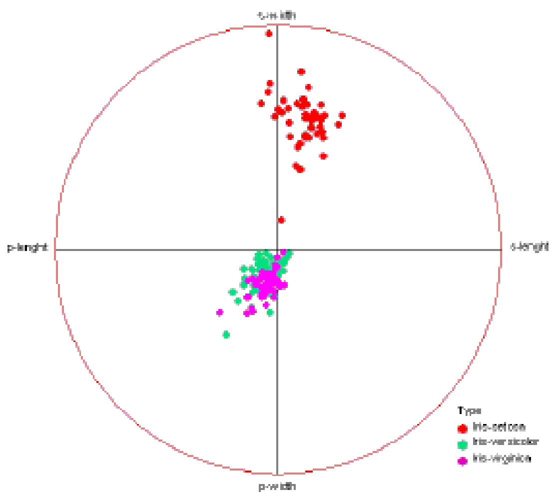


Figure 2.18: RadViz visualization of a flower data set

The RadViz approach is powerful for visualizing correlations between the dimensions. Clusters can be easily identified. However the approach does not provide insights into the distribution of the data within a dimension.

2.3 Mathematical Approaches

A branch of mathematics known as multivariate statistics consists of many techniques for identifying relationships and patterns in the multi-dimensional data space. The approach taken by the multivariate statistical techniques is to reduce the number of variables necessary for summarizing the data. Lower dimensional spaces can be analyzed easily with graphical techniques. Some of the techniques available are

- Multi-dimensional scaling
- Cluster Analysis
- Principal Component Analysis

Multi-dimensional scaling (MDS) is a set of data analysis techniques which seek to portray the similarities between dimensions in a low-dimensional space so that their relationships can be examined both visually and numerically [Davison92]. The fundamental goals of MDS are

- reducing number of dimensions

- providing spatial representations for the data

The inputs are required to be in the form of interrelationships like distances, measures, and proximity. MDS procedures use metrics (weighted Euclidean or Minkowski or Manhattan) to compute the similarities or dissimilarities between dimensions. Results are reflected in the spatial representations where points close to each other have the strongest possible relation to the similarities among the pairs of dimensions. That is, two similar dimensions are represented by two points that are close together, and two dissimilar objects are represented by two points that are far apart. For example Fig 2.19 shows the MDS based on the flying mileages between ten American cities. The MDS map reveals the relative locations of the cities in the United States.

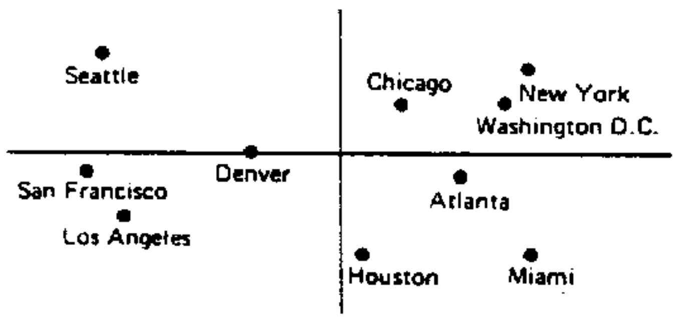


Figure 2.19 MDS of flying mileages between 10 American cities.

Cluster Analysis is another multivariate analysis technique that attempts to organize information about variables so that clusters (relatively homogeneous groups) can be formed [Aldenderfer84]. The inputs are proximity measures similar to that of MDS. The four basic steps in cluster analysis are data collection and selection of the variables for analysis, generation of a similarity matrix, decision about number of clusters and interpretation, validation of cluster solution. The output of the analysis is a tree diagram showing clusters of the dimensions. The tree diagram helps in selection of variables of importance from portions of the tree where the cluster structures are stable for a long distance.

Principal component analysis (PCA) is a mathematical procedure that transforms a number of possibly correlated variables in to a smaller number of uncorrelated variables called principle components [ComponentAnalysis03]. The mathematical technique used in PCA is eigen analysis. The input to PCA is a matrix of multi-dimensional values. The matrix is standardized to construct a square symmetric matrix. The square symmetric matrix is solved for eigenvalues and eigenvectors. The eigenvector associated with the largest eigenvalue has the same direction as the first principal component. The eigenvector associated with the second largest eigenvalue determines the direction of the second principal component. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible.

2.4 Lessons Learned

Every approach for visualization of multi-dimensional spaces is suitable for specific applications and for specific types of patterns (e.g. identifying clusters). Nevertheless a survey of the different approaches provides valuable insight on shortcomings and the missing features in different approaches. Some of them are as follows

- A visual overview of the space is needed

- Scalability especially beyond six dimensions is necessary

- Uniform treatment of dimensions is required

- Point aggregation is desirable. Aggregation of points refers to grouping several functional values and representing them as a single pixel on the display. It is important for scalability and for making the visualization more realistic.

- Computation of functional points should be done on the fly. To depict functional values they may be pre-computed or computed on the fly (computational steering). Pre-computation is expensive in terms of memory, and functional values between pre-computed points need to be interpolated. Comparatively, computational steering is a better choice but harder to implement. It would be helpful to implement computational steering in the visualization algorithm to provide for scalability.

- Two-dimensional plots and graphs should be used. Two-dimensional plots and graphs are the easiest graphical presentations for users to comprehend. These can be considered for providing detailed views of the functional space to the user.

2.5 Our Contributions

Most of the techniques for visualizing multi-dimensional functions have two commonalities. They employ reduction in dimensionality through slicing techniques, and the functional value is color encoded and represented. Slicing techniques are important for seeing portions of the space but are not effective enough to provide an overview of the space. If slicing and projections are not provided for as in Mihalisin, scalability becomes an issue.

Our approach overcomes these drawbacks. It is a novel technique with an emphasis on providing an overview of the space. The overview helps users in grasping the nature of the space and also displays all the dimensions equally. Details of a local region selected, by the user is blended in the overview. Unlike tools for multi-dimensional relations the approach promises scalability in terms of the number of dimensions and the range per dimension. A simple aggregation scheme has been attempted to make the visualization realistic.

Chapter 3: CONCEPTUAL MAPPING

3.1 Introduction

This chapter introduces the fundamental concepts of the PolarEyez visualization namely focal point, focal rays and the focus+context technique. The concepts required in the generation of the visualization such as selection of functional data points, radial arrangement of points, hypercube, faces, paths are also described.

3.2 Functional space

Functional spaces being considered for our approach are continuous or discrete. They can be represented either as mathematical formulae or as discretely sampled values as described in Chapter 1. The range for the dimensions of the functional space is specified by the user with the maximum and minimum values. Hence the space is bounded and is considered as an n-dimensional hypercube. The challenge is to map an n-D space to 2-D space for the visual display.

3.3 The approach

PolarEyez is a new focus+context visualization for multi-dimensional functions that uniformly integrates all dimensions into a single view. The visualization arranges the dimensions in a radial fashion around a polar focus point [Figure 3.1], and maps function value to a color scale at every point in the space. For 2-dimensional functions, it is identical to a heat map. However, instead of adding views of 2D slices, additional dimensions are added directly into the radial arrangement. This provides an integrated overview of the space from the perspective of a navigable focus point within the space.
The PolarEyez approach conceptually supports any number of dimensions and arbitrary range bounds on each dimension. For simplicity, the following mappings will be described for 3D functions defined within a bounding cube. The concepts scale up naturally, and examples with greater dimensionality are described later [Jayaraman02].

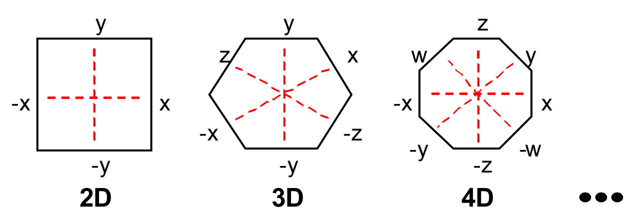


Figure 3.1: Basic spatial arrangement in PolarEyez with increasing number of dimensions.

3.4 Conceptual Mapping

3.4.1 Focal point

Functional spaces are vast and so a point or region has to be selected to start the exploration of the space. A point called the focal point is selected in the multi-dimensional function space which acts as the starting point. By default the focal point is set at the center of the function space. The function value is sampled at this point, mapped to a color scale, and displayed. The focal point is represented in the center of the visualization thereby serving as a reference point with respect to which the entire visualization is constructed.

The focal point also helps the user in navigating through the space by serving as a point of interest. The user's focus would be in certain regions of the space at the outset which would subsequently change on identifying other regions of interest. This shift in focus is translated as the selection of another focal point in the functional space. The visualization is accordingly reoriented to reflect the shift in interest. Fig 3.2 shows change in the focal point can be used for navigation [Jayaraman02].

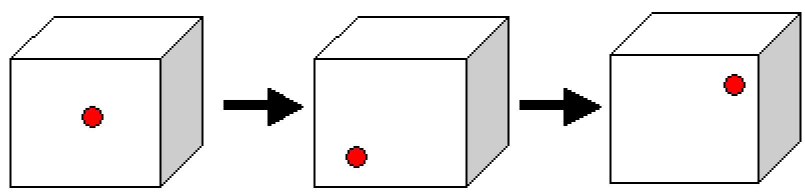


Figure 3.2: Navigating the space by changing the focal point.

3.4.2 Focal rays

Rays emanate from the focal point in all directions and extend to the boundaries of the functional space. The focal rays are dense and help encompass all data points in the space. The data points on the rays are sampled and their functional values are color coded and plotted on the visualization.

A radial arrangement technique is employed for arranging the focal rays around the focal point in the visualization (Figure 3.3). Essentially, the rays in the multi-dimensional space are rotated around the focal point onto a 2D plane of the visualization. The end points of the rays in the functional space are distributed in a grid like pattern on the cube bounding the functional space. Enough rays are chosen such that the visualization is completely filled with colored rays. Hence, each pixel on the perimeter of the visualization is the destination point of some ray emanating from the focal point [Jayaraman02].

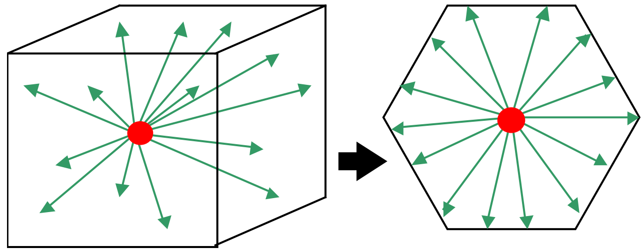
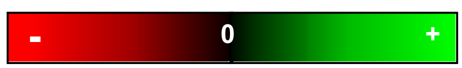


Figure 3.3: Focal point with rays in all directions map from function space to visualization space.

3.4.3 Coloring scheme

The color attribute has been employed for representing the functional values of the data points on the visualization. Shades of green whose RGB values vary between (0,0,0) to (0,255,0) represent the positive function values. Shades of red whose RGB values vary between (0,0,0) to (255,0,0) represent the negative function values [Figure 3.4a].The points at which the functional value is undefined and the points outside the bounded space are represented in white. The focal ray consists of different colors [Figure 3.4b] depending on the function values assumed by the data points.



(a)

(b)

Figure 3.4: (a) Approximate color scale for mapping function values. (b) Color coded focal ray.

3.4.4 Techniques for ordering the rays

The two main challenges confronted in mapping the focal rays to the visualization are

- Identifying rays
- Ordering the rays around the focal point.

Two heuristics, faces and paths are used for the purpose and are elaborated below.

3.4.5 Faces

The functional space is bounded, as mentioned earlier by an n-dimensional cube. The focal rays strike the faces of the cube. The first heuristic groups rays according to the face of the bounding cube that they intersect. The group of rays striking a face of the cube defines a pyramid whose base is a face of the cube, and whose pinnacle is the focal point. The pyramid of rays is represented in two dimensions (visualization) as a triangular pie-slice [Figure 3.5a]. The exterior edge of the pie slice is flat to represent the corresponding face of the cube. The different triangular slices constitute a polygon. So, in the case of a 3-D function, the bounding cube has six faces and so the visualization is a hexagon with each face of the hexagon representing a face of the cube. Generally, an n-D function is mapped on to a 2n sided regular polygon.

Grouping rays according to the face of the bounding cube helps in grouping them with respect to their rotationally nearest primary axis. So, the pie slices and their faces are identified by the primary dimension that they bound. For example, in a 3-D function, f(x,y,z), pairs of opposite faces of the bounding cube would correspond to the "x", "y" or "z" dimension. The value of the primary dimension increases or decreases on moving from the focal point to the cube face. The opposite cube faces are displayed as pie slices on opposite sides of the focal point in the visualization. One of these pie-slices represents the functional variation for increasing values of the primary dimension (+x) it represents, while the other shows the converse (-x).

Hence the heuristic of faces helps in an unbiased mapping of all the dimensions in the visualization.

3.4.6 Paths

The second heuristic orders rays within each pie slice according to their destination points on the cube face. A path is defined on the cube face that linearly orders the ray destination points along the hexagon face. There are many possible alternatives for this path. Our current approach uses a straightforward scan-line path across the cube face (Figure 3.5b). The ray at the center of the cube face, and parallel to the primary dimension of the face, is located in the center of the pie slice. A spiral path, a potential alternative, orders rays according to increasing angle from the primary dimension of the face. Note that in the general case a face of a d-dimensional hyper-cube is itself a (d-1)-dimensional hyper-cube.

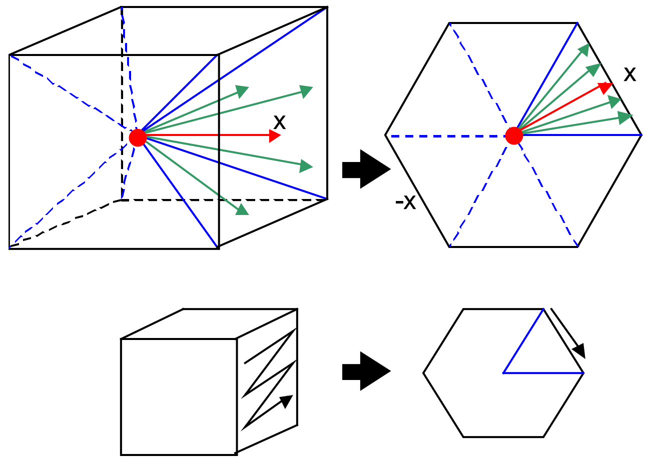


Figure 3.5: Rays are (a) grouped by face, and (b) ordered within a face by a scan-line path.

3.4.7 Generalized conceptual mapping

To generalize beyond 3D functions, the mappings from multi-dimensional function space to 2D screen space are summarized as (with colored reference to Figure 3.5):

- Focal point maps to focal point (in Figure 3.5, red).
- Ray maps to ray (green).
- Hyper-cube maps to a regular polygon (black).
- Hyper-cube face maps to polygon face (black).
- Hyper-pyramid maps to pie slice (blue).
- Function value maps to color.

3.4.8 Focus+context

Focus+Context (F+C) [Spe01] technique makes it possible to show targeted information and details to the user. It aids in the integration of the overview and the detailed view of the functional space smoothly into the visualization.

While exploring functional spaces, users typically select portions of the space for viewing and understanding in greater detail. The center point of such regions of interest is the focal point. Regions near the focal point are given more importance and displayed in greater detail compared to the rest of the space. The context can be shown in terms of a less detailed view of the rest of the function space. This helps in indicating where the regions of interest lie in the space. Mathematically, the extent of details shown in the visualization is inversely proportional to the distance from the focal point (center of the visualization). Thus, the details and the contextual overview are blended smoothly using the focus+context technique.

3.4.9 Concept of aggregation

Since function data may exist between the rays in the functional space, it is necessary to aggregate this data onto the nearest ray so that all data is represented in the visualization, making it more realistic. That is, each ray aggregates (e.g. averages) data around it. Hence, each ray in function space is actually a narrow pyramid. Together, these narrow pyramids completely fill the entire cube. In visualization space, each ray is actually a narrow triangular pie slice.

This aggregation, combined with the radial layout, creates the smooth focus+context [Spence01] effect. This enables users to view details of a localized region without losing the context of the overall functional space. Less aggregation (more detail) occurs near the focal point, and more aggregation (less detail) occurs distant from the focal point. Furthermore, aggregation increases smoothly with increase in distance from the focus, creating a smooth transition between detail and context.

Proof of this concept is helpful. Intuitively, the surface of a cube of width w in function space is visually encoded on the perimeter of a hexagon of width w in 2D visualization space (technically, a linear mapping on the width occurs to map distance in function space to distance in screen space). As width w increases, the surface area of the cube increases more rapidly than the perimeter length of the hexagon. Hence, as width w increases (and therefore distance from the center focal point increases), more area in the function space must be aggregated and encoded into relatively less perimeter length in visualization space (Figure 3.6).

Mathematically, a cube with radius r from the focal point has width w=2r. The surface area of the cube of width w is $6w^2$. In the general case, the hyper-surface area A of the n-dimensional hyper-cube of width w is the total volume of its 2n hyper-faces, each of which is an (n-1)-dimensional hyper-cube with volume w^{n-1}. So $A = 2nw^{n-1}$. The data contained in the surface area A is encoded in the visualization on the perimeter of the corresponding polygon of radius r and width w. The perimeter length L of the polygon of width w can be approximated by the circumference of a circle of diameter w, or $L = \pi w$. Hence, the aggregation factor, which measures the amount of data encoded per unit screen space, is:

$$\text{Aggregation} = A/L = 2nw^{n-1}/\,\pi w \approx nw^{n-2} \approx n2^n\, r^{n-2}$$

Therefore, for greater than 2 dimensions, the aggregation factor is a function of the distance r from the focal point. Aggregation increases exponentially as the distance from the focal point increases. Note that for n=2, aggregation is constant (e.g. a heat map) [Jayaraman02].

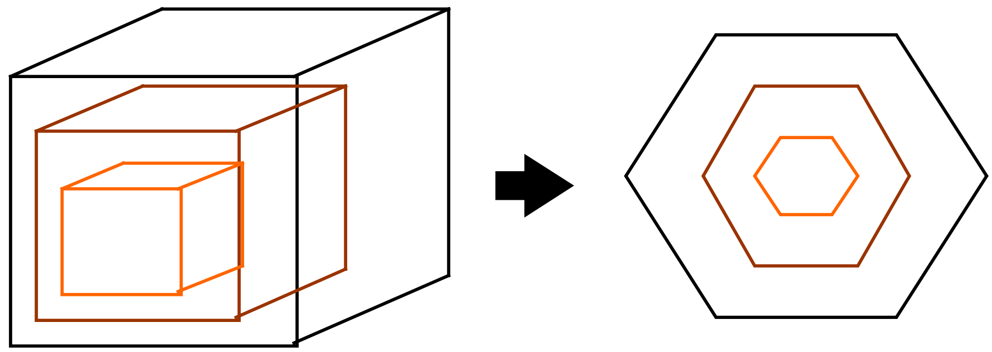


Figure 3.6: Surfaces of concentric cubes map to perimeters of concentric polygons. Outer cubes require more aggregation.

3.5 Interactive exploration

User interaction enhances the merits of the visualization. For example, users are allowed to select and navigate portions of the space. Based on user inputs, the visualization is re-oriented. The feedback guides future exploration decisions and helps uncover functional properties. The various features enabling user interaction are elaborated below.

3.5.1 Location awareness

As users move the mouse over the visualization, a tool tip shows the coordinates of the point in multi-dimensional space and the corresponding function value. Location awareness provides access to detailed information when required (details-on-demand).

3.5.2 Navigation by change of focal point

Users can navigate the space by altering the coordinates of the focal point within the multi-dimensional function space. This is achieved by entering the desired coordinates or by clicking on a point in the display to make it the new focal point. The visualization places the new focal point at the center, and reorients the polar overview from the perspective of the new focal point. This enables users to view more details of a desired region using the focus+context technique. Navigating the focal point nearer to the edge of the bounding cube will cause some rays to be shorter than others. Rays emanating toward a nearer face will naturally intersect the bounding cube sooner, and causes portions of some pie slices to appear truncated. This helps users recognize nearness to the boundaries, and orient themselves within the space [Figure 3.7].

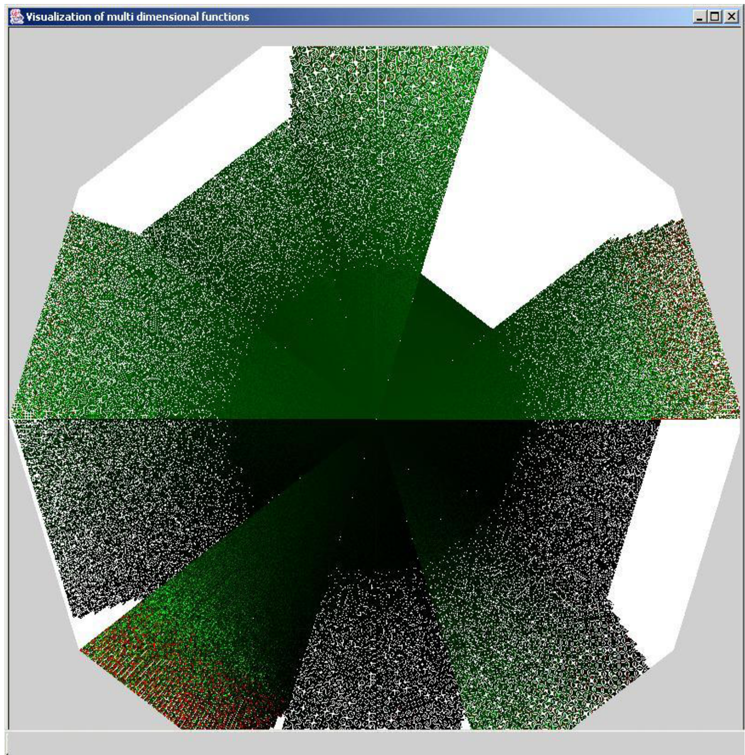


Figure 3.7: Shorter focal rays at the edges of the space

3.5.3 Filtering of dimensions

Users can also filter out uninteresting pie slices from the overview. This provides more detail to pie slices of interest by evenly distributing extra angular screen space to these slices. This enables users to focus on interesting regions, or eliminate independent variables that have little effect on function behavior. This feature could be extended to enable users to independently control the angular space devoted to each pie slice by directly resizing them.

3.5.4 Highlighting

Querying is supported by means of highlighting functional data points. Users can enter values or use dynamic sliders [Ahlberg94] for specifying functional and parameter values. Points satisfying the range criteria are highlighted in bright yellow [Figure 3.8]. For example, users can analyze whether desired functional points occur in clusters or are spread throughout the space.

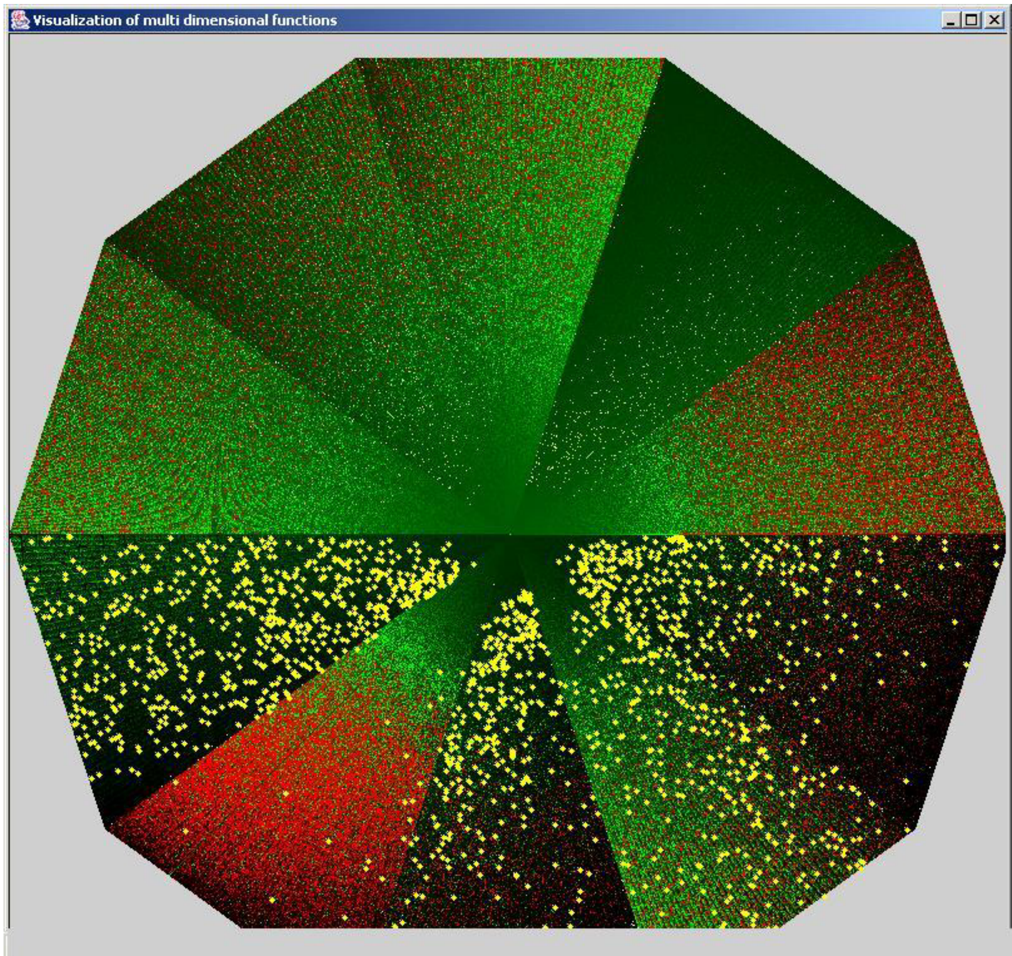


Figure 3.8: Highlighting functional points

3.5.5 Detailed Views

Coupling the PolarEyez overview with detailed views makes the tool powerful. Two types of detailed views that can be provided are described below.

Users select regions for detailed examination in the overview. The detailed view is a miniature PolarEyez visualization, with the focal point being the center of the selected region. An option for varying the aggregation factor is provided giving the user the flexibility to alter the degree of zooming. By manipulating the aggregation factor and shortening the size of the selected region the number of points shown in the detailed view can be adjusted. A variation of this scheme is to provide detailed view in the overview itself. The PolarEyez view can be generated with the same function with reduced parameter ranges. This approach has the advantage of having less windows cluttering the screen at the expense of the original larger overview of the functional space.

Another approach for detailed views is to couple the PolarEyez overview with the Hyperslice [VanWijk93] tool. Such detailed views reveal the variation between two parameters at a time. World within world is also a good alternative but coupling with PolarEyez is expensive in terms of windows and screen space required.

Use of the two detailed views is explained with an example. Let us consider a five dimensional function f(a,b,c,d,e) with the parameters ranging between 0 and 10. The functional space is predominantly negative except when $5<a<7$. The regions with positive functional values can be easily identified from the overview. This region can be selected with the mouse for further examination using the PolarEyez type detailed view. The detailed view can reveal the values taken by the different dimensions. For example, a particular dimension say b, has the value 8 for all points in the selected region. Since two of the parameters have a fixed value in the selected regions, the user may want to examine the variation of the functional value for the remaining dimensions. The Hyperslice tool type of detailed view can help in seeing the desired functional variation.

Chapter 4: ALGORITHM AND SOFTWARE ARCHITECTURE

4.1 Introduction

This chapter deals with the software implementation of the concepts elaborated in the previous chapter. Two algorithms employed for generating the visualization and the tradeoffs between them are first described. The first algorithm attempts to identify functional points and then map them onto the visualization while the second selects pixels on the visualization and identifies the corresponding functional point in the space for representation. An overview of the software architecture along with the various modules is also provided. Aggregation is very important especially in high dimensional spaces and is complex to implement. The various implementation strategies for aggregation are discussed. The chapter concludes with the issues confronted in the software development process.

4.2 Algorithm 1: Functional space to visual display mapping

The algorithm maps the functional values of the points on the focal rays in the functional space onto the visualization [Shastri02]. The visualization is a circle divided into several segments. The number of slices in the visualization is twice the number of dimensions with each slice spanning 360/2*n (n being the number of dimensions) degrees.

The focal point is mapped onto the center of the visualization screen with the points on the focal rays arranged around it. For mapping the points on the focal rays, the number of focal rays striking a face of the cube is computed. The face of a hypercube is divided into a grid containing L number of grid cells where L is given by

$$L = r * (360/2*n),$$

n being the number of dimensions and 'r' being the distance from the pixel to the focal point in the visualization. The functional points on the rays striking the particular grid surface are aggregated and represented on the visualization. A green-red coloring scheme with green shades representing positive values and red shades representing negative functional values is used during the mapping.

A variant of the approach, the visualization display to space mapping algorithm, was also attempted and described in the following section.

4.3 Algorithm 2: Visual display to functional space mapping

The fundamental idea behind the algorithm is to calculate the functional point corresponding to every pixel in the visualization. The functional value and the corresponding color value is subsequently computed and plotted on the screen. Aggregation is achieved by averaging several data values that are closer to the computed point on the focal ray and representing the average value on the pixel. The concepts involved in the mapping of an n-dimensional space to a 2-dimensional space are elaborated below.

4.3.1 Pixel Attributes

Every pixel on the visual display is associated with two attributes computed by the algorithm. The two attributes are used to select the hypercube and location of the functional point on the corresponding hypercube. The two attributes are

- The distance of the pixel from the center of the visualization, r
- The angle spanned by the pixel with the horizontal, θ

The center of the visualization represents the focal point. Hence the distance r, yields the distance of the pixel point from the focal point in the visualization. The distance is computed by using the distance formula

$$r = \sqrt{((i-x)^2 + (j-y)^2)}$$

where x and y are the position coordinates of the pixel.

The angle θ varies between 0 and θ_{max} (the maximum angle allocated for a slice). The maximum angle of any slice is given by

$$\theta_{max} = 360/2*n$$

n being the number of dimensions. The angle is computed using the tangent of the angle formed by the focal point, the pixel and the horizontal.

4.3.2 Mapping Pixel Attributes to Functional Coordinates

The essence of computing data points in the functional space lies in translating the pixel attributes to functional coordinates. This is achieved in two stages. First the distance attribute of the pixel is mapped to functional coordinates using the following equation

$$r/R * \text{ dimension range,}$$

r being the distance of the pixel from the focal point and R is the maximum size of the visualization. The translation of the distance attribute helps in identifying the hypercube to be considered for obtaining the other functional coordinates. The hypercube face considered for computation is based upon the slice in which the pixel lies.

A path (scan line path) is traced out on every one of the cube faces as discussed earlier. The angle attribute is used to locate the position of the data point on the path. The scan line path forms a straight line when stretched out. The angle attribute is converted to a distance traveled on the path. The maximum value of the angle θ_{max} covers the entire length of the path. So $\theta < \theta_{max}$ maps onto a distance less than the maximum length of the path. For intermediate values of the angle the position of the functional point is computed using

$$\theta * (\text{maximum length of the path on the particular cube face}) / \theta_{max}$$

Using the dimensions of the scan line the distance is repeatedly divided to provide the position of the functional point on the path. The length of the scan line path varies depending on the size of the cube. Cube sizes are smaller near the focal point and increases as the distance from the focal point increases.

4.3.3 Alignment of functional points on the focal ray

Functional points on a focal ray should appear similarly when represented in the visualization. In order to ensure alignment of points similar triangle concepts are used. For any pixel the coordinates of the functional coordinates on the outermost cube are first obtained using the technique described in the previous section. The coordinate values are then reduced to the cube under consideration using the concept of similar triangles. In Figure. 4.1 triangles FDE and FD'E' are similar. First DE is calculated and then D'E' is computed using the proportional property of similar triangles.

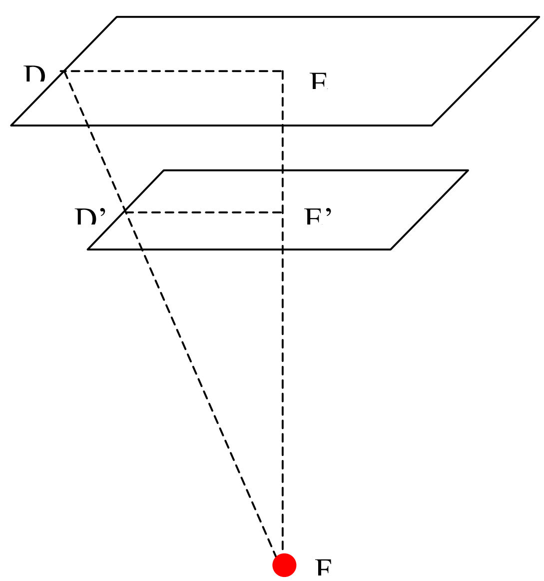


Figure 4.1: Similar triangles formed by the cube faces and the focal point

4.3.4 Functional value computation

After the computation of the functional coordinates in the multi-dimensional space the functional value is computed using the mathematical formula or array of values provided by the user. The functional value is subsequently mapped on to the color scale and the pixel is plotted on the screen

The various steps in the algorithm is provided below. Fig 4.2 illustrates the algorithm.

For each pixel *p(x,y)*:

1. Map pixel *p* to region *r* in function space:
    a. Map pixel *p* to point *pt* in function space:
        i. Determine pie slice *s* containing pixel *p* using angles w.r.t. focal point
        ii. Determine concentric polygon face containing *p* using radius w.r.t. focal point
        iii. Determine concentric cube face represented by polygon face using radius and slice *s*
        iv. Calculate point *pt* of pixel *p* on cube face using scan-line path:
            - Calculate fraction of distance of *p* along polygon face, using angles
            - Map to distance along path on cube face
            - Follow distance on cube face to locate *pt*
    b. Approximate region *r* around point *pt*:
        i. Map pixel width to fraction of path length using approximate number of pixels on polygon face
        ii. Include that fraction of path around *pt* in *r*
        iii. Map pixel width to radial depth of cube face using approximate number of pixels on pie radius and dimension range.
        iv. Include that depth in *r*
2. Compute aggregate function value *v* for region *r*:
    a. If sampled array data: average the samples contained in region or interpolate samples around region *r*
    b. If math formula: sample and average the center and corners of region *r*
3. Map value *v* to color *c* using color scale
4. Plot color *c* in pixel *p*
5. Repeat for mirror pixel *p2* on opposite side of focal point, reusing *pt* and *r* calculations from step (1) where possible.

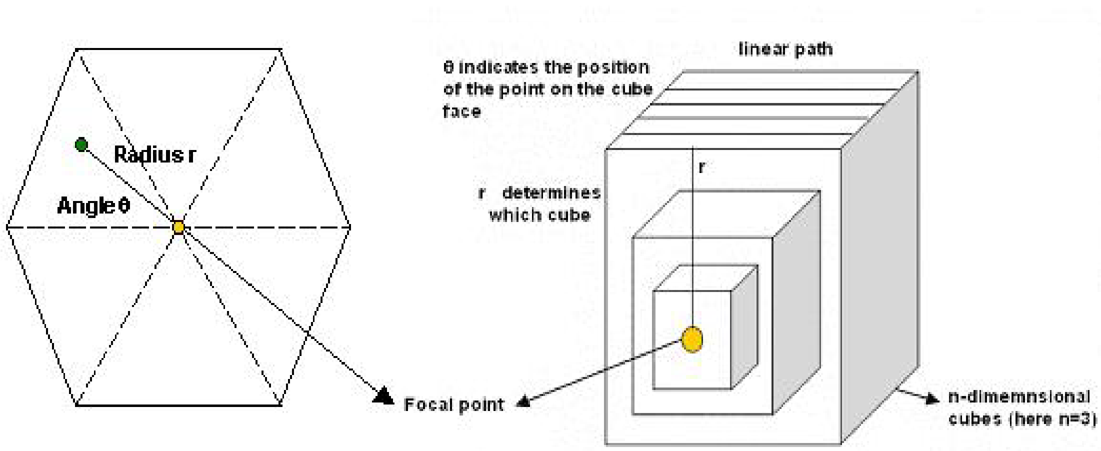


Figure 4.2 Illustration of the visual display to functional space mapping algorithm

4.4 Tradeoffs between the two algorithms

The major issue with the functional space to visual display algorithm is the difficulty in ensuring the alignment of points along the focal ray which is not a trivial problem to solve. The selection of points on a grid in the cube face also poses problems. Grid bounds need to be stored for computation of functional points in the grid. The above shortcomings are easier to overcome in the visual display to functional space mapping. However aggregation is made more complex due to the row-wise pixel scanning. Adjacent pixels map onto different hypercubes and hence calculated values need to be stored for every pixel for performing aggregation. The visual display to functional space mapping is preferred due to the reduced complexity of implementation.

4.5 Software architecture

The visual display to functional space algorithm has been implemented in Java and the resulting visualization tool is called PolarEyez. The software development is based on a modular structure delineating the modules with differing functionalities. The three main modules are

1. Functional Input module
2. Visualization Generator module
3. Interactive features module

The modules have been developed as Java packages [Sun02] to promote code readability and easy extension of the visualization software. For example users can add interaction functions if desired without having to tamper much with the code blocks pertaining to the visualization generation. Every module is composed of a primary object encapsulating the essential functionality of the module. Every object contains several functions for facilitating the interaction between modules.

The choice of Java for software development is influenced by the support for graphical procedures and GUI components. The availability of API's and inbuilt functions for generating graphics reduced programming complexity and time. Added to the graphical support are several functions for performing trigonometrical computations. Also, Java lends itself well for a modular development using the object oriented concepts.

4.5.1 Functional Input Module

This module is responsible for the user inputs namely the function and its approximate maximum and minimum values. This module computes the number of dimensions based on the inputs and communicates these values to the visualization generator module.

4.5.2 Visualization Generator Module

This module is responsible for generating the entire visualization. This involves the computations and the graphical functions necessary for the display. The computations involved are

- Calculating number of slices and the angle spanned by them
- Computation of radius and the angle spanned by a pixel
- Computation of the functional coordinates
- Computation of RGB color values for the functional value
- Graphic display of the functional values at the appropriate point in the visualization

This module contains a number of functions, which output the coordinates and other necessary values to the interaction features module.

4.5.3 Interactive Features Module

This module contains functions implementing the interactive features. This module closely interacts with the visualization generator to pass on user inputs (e.g. functional value range for highlighting points) for generating the visualization afresh. Specifically the interactions supported are

- Change of focal point
- Coordinate values on the visualization
- Highlighting of values
- Filtering of slices

The Figure 4.3 shows the modular organization and the interaction between them.

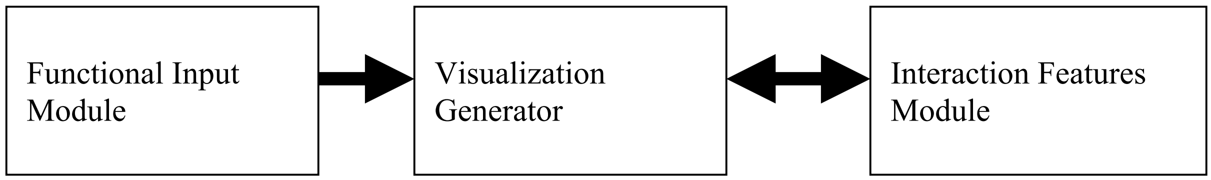


Figure 4.3: Modules comprising the PolarEyez visualization

4.6 Implementation of aggregation

Aggregation is the averaging of several functional values so that more points are represented on the visualization. Due to the limited amount of visual display space and the large amount of functional points present aggregation becomes important for increasing the realistic nature of the visualization. Aggregation can be performed along a ray or path or along both.

4.6.1 Aggregation along a focal ray

Several points lying on the same focal ray between two successively mapped points are not accounted for. All these points can be aggregated on a pixel. Pixels representing points along a particular focal ray possess the same value for the angle attribute. Distance and angle attribute values for every pixel are stored in arrays in memory. The pixel with the same angle attribute value in a slice is found out by searching the array. Hypercubes between the functional points corresponding to the two pixels are used for computing functional points along the focal ray. The resolution for considering hypercubes depends on the aggregation factor.

4.6.2 Aggregation along the path on the cube face

Two pixels mapping onto points on the same face of the hypercube do not map onto successive points in the functional space. The intermediate points in the functional space can be aggregated. The aggregation in this case occurs along the scan line path on a face of the cube and can be implemented in any of the following two ways.

Integration can yield better results for aggregation of points. The integral of a function is the area under the curve of the function. Several numerical methods such as Trapezoidal rule and Simpsons rule exist for finding the definite integral of the function. The approach is to subdivide the area under the curve into infinitely small rectangles and obtain the area of each. The sum of the areas of the rectangles provides the area under the curve of the function. The integration process yields the sum of all the functional values between two points on the face of the hypercube. The average value is obtained by diving the result of integration by the total number of data points.

Interpolation techniques can be used for obtaining more number of points for increasing the aggregation factor. These techniques are useful especially in the case of multi-dimensional array of sampled values.

4.6.3 Effects of aggregation

Aggregation tends to smooth out values. This raises the doubt of missing important functional values such as spikes. To understand the effects of aggregation let us consider a function whose value lies between -100 and 100. Table 4.1 shows an example where two spikes (one negative and one positive) occur within the set of aggregated points. The effects of the two spikes are not reflected in the aggregated result.

Number of values aggregated	Functional values	Aggregated value
8	**-90,**12,12,13,14,**90**,12,12	9.375

Table 4.1: Factors influencing aggregation – multiple peak values

Table 4.2 depicts the effect of increasing the number of aggregated points where 45 is a local maximum. The functional values near the local maximum are much lower (around 12). As more number of points are aggregated the effect of local maximum becomes indiscernible. Hence increasing the aggregation affects the revelation of patterns in the overview.

Number of values aggregated	Functional values	Aggregated value
6	12,12,12,13,**45**,14	18
16	12,12,12,13,14,**45**,12,12,11.8,11.9,12.4,11.1,11.3,9,10.5,10	13.75

Table 4.2: Factors influencing aggregation – number of aggregated points

4.7 Implementation Status

Table 4.3 shows the implementation status of the concepts elaborated in Chapter 3.

Feature	Implementation status
Location awareness using tool tips	✓
Change of focal point	✓
Highlighting points	✓
Detailed view	X
Filtering of dimensions	X
Aggregation	X

Table 4.3: Status of implementation of the concepts

4.8 Software Issues

4.8.1 Execution time

Execution time is affected by the computations for mapping pixel attributes to functional domain and in fetching functional values from secondary storage when specified as a multi-dimensional array of points. Fetching functional values from secondary storage becomes necessary when the sampling of the functional space is fine grained and/or when the number of dimensions is high. The retrieval time can be reduced to an extent by caching related functional points (based on areas selected by the user in the visualization) in every slice in memory. However when users select very different points in the visualization during interaction caching of values loses its advantage.

Computations are simple but several such calculations are made in every slice. All the calculations deal with real numbers (double variables in Java) to maintain precision. An effort is made to minimize redundant calculations by using a mirroring technique. Specifically, the computations made for a pixel in a slice, such as angle and distance of the pixel from the focal point are used for pixel the diametrically opposite slice. This limits the number of calculations and speeds up the generation of the visualization. However aggregation of functional values increases the execution time since functional value computations need to be performed for all the surrounding points.

Figure 4.3 is the graph of the execution time for the $\Sigma\cos(x^2)$ function (range –20 to 20) visualization (without aggregation) for varying number of dimensions. The visualization was run on a Pentium 500Mhz PC with 128 MB RAM. Increase in execution time with an increase in the number of dimension can be clearly seen.

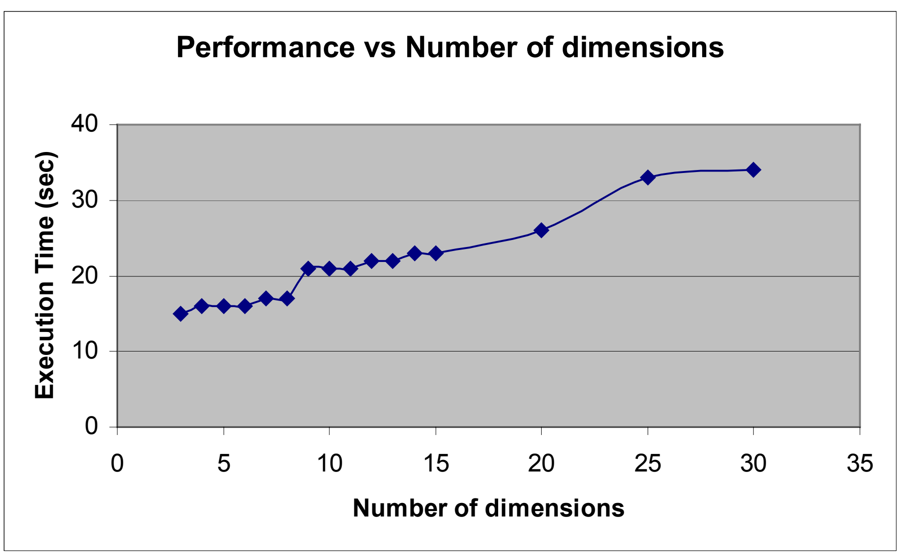


Figure 4.3: Performance vs Number of dimensions plot

4.8.2 Aggregation challenges

Aggregation is performed be averaging the functional values near the computed point in the functional space. Adjacent pixels on the display map onto different hypercubes and even possibly different cube faces due to the slightly varying distance values from the focal point. Hence, pixels mapping onto the same hypercube and face need to be kept track of. Arrays are required to store information regarding previously visited functional points on the scan line path. Since memory storage is limited, minimum amount of details pertaining to a pixel need to be stored (such as the distance of the pixel).

An important issue is determining the number of points to be aggregated. The greater the extent of aggregation greater is the number of points accounted for in the visualization. Given the tradeoffs between visualization generation time and accuracy perfect aggregation is not possible. Hence the visualization is lossy to an extent.

4.8.3 Response delay

Response delay is the time taken for repainting the visualization based on user selection. Response delay needs to be as minimum as possible to avoid users becoming impatient and irritated. Sine functional points are computed on the fly the delay is affected by the number of computations required by the mapping algorithm. Every user interaction, such as change of focal point and highlighting cause a repetition of computations. An increase in the extent of aggregation affects the response delay aversely.

Response delay is currently minimized by storing functional values and coordinates using arrays in memory. As the number of dimensions increases the amount of memory required increases and places a demand on the resources.

4.8.4 Memory requirements

Computer memory is a limited and warrants careful usage. Storage of values in memory can be used for reducing execution time and response delays during interaction. However due to the nature of the functional domain the number of data points to be handled is large. The amount of memory required cannot be provided for unless graphical workstations are used. Hence memory should be used wisely by storing essential values and removing those which have been made use of. Secondary memory can also be used for storing values such that the execution time of the program is not affected adversely. Effective memory usage is thus a major challenge in the development of the algorithm.

Chapter 5: APPLICATIONS

5.1 Function Examples

5.1.1 Analysis of the Cosine Function

The visualization of the function $\cos(x^2) + \cos(y^2) + \cos(z^2)$ is presented [Fig 5.1]. A simple function is taken and its properties are explained in detail so that readers will be able to relate the properties of the function with the visualization.

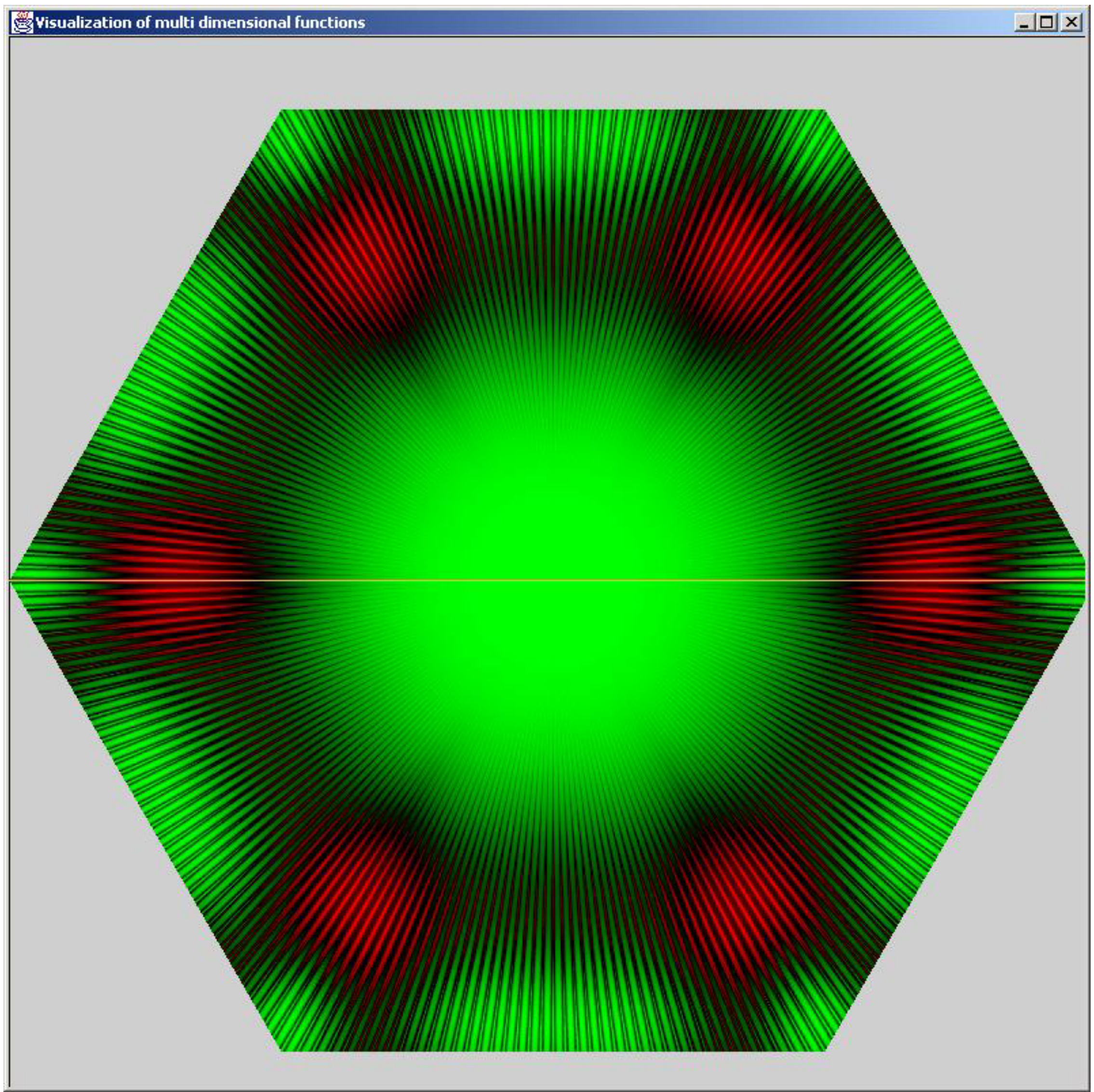


Figure 5.1: $\Sigma\cos(x^2)$ function in three dimensions (range –20 to 20)

The range of the parameters is –20 to 20 on all the dimensions. The properties of the function as seen from the visualization are as follows

- The functional space consists of both positive and negative values
- The focal point is at (0,0,0) and the space around the focal point consists of very high functional values (indicated by the bright green color). The cosine function has the high value of 1 at x = 0
- The function is symmetric because interchanging the dimensions does not alter the functional value. In the visualization all the triangular slices appear similar revealing the symmetricity of the function
- cos(θ) = cos(-θ). This property is verified because the opposite slices are similar. In opposite slices one of them depict positive values of a variable while the other depicts the negative values
- Alternating nature of the cosine function is exhibited due to the alternating patterns of green and red along any ray

The alternating patterns of red and green are evident when the range of the parameters is increased to –50 to 50 [Fig 5.2].

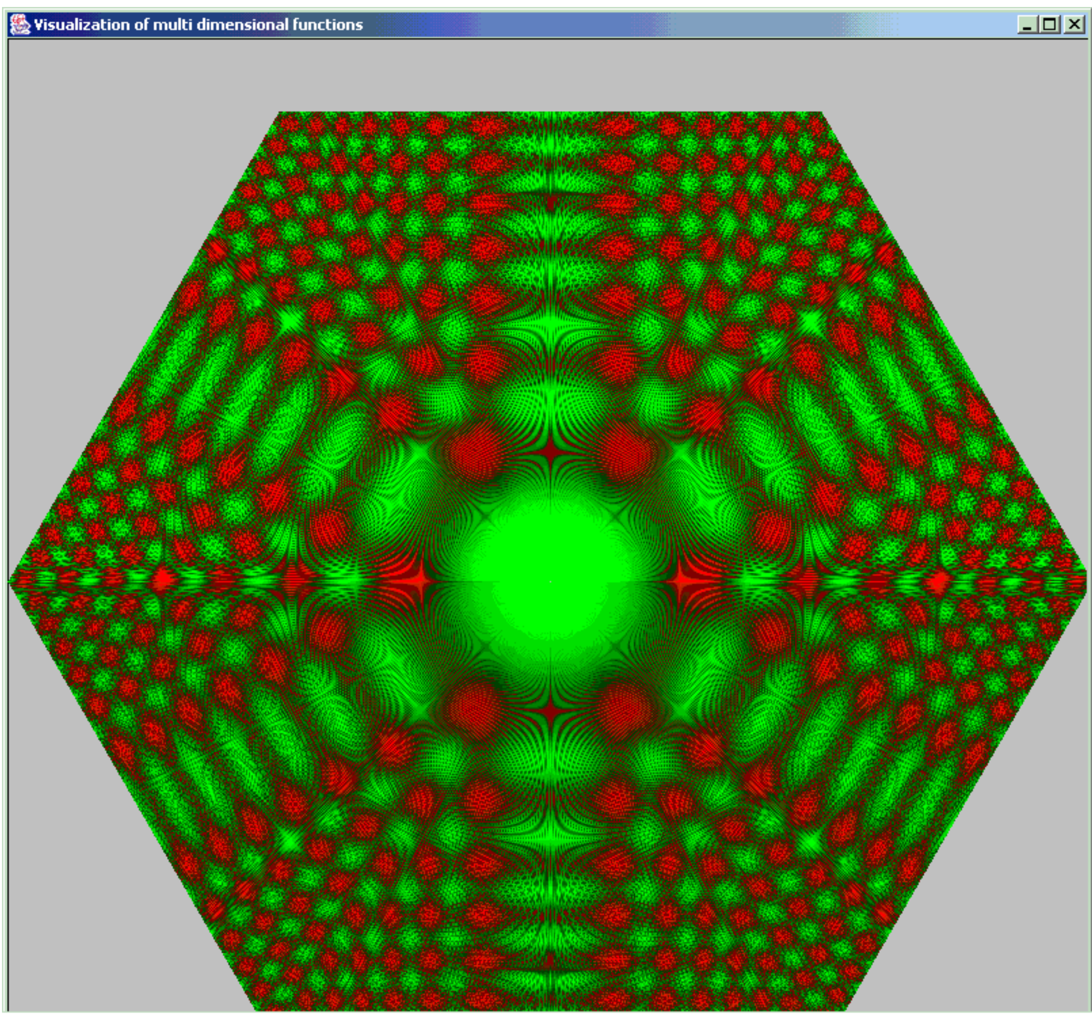


Figure 5.2: Σcos(x^2) function in three dimensions (range –50 to 50)

The visualization for the 5D,10D and 12D $\Sigma\cos(x^2)$ function is shown below [Fig 5.3,5.4] Increasing the number of dimensions results in the slices becoming thinner (the red regions appear smaller and squashed).

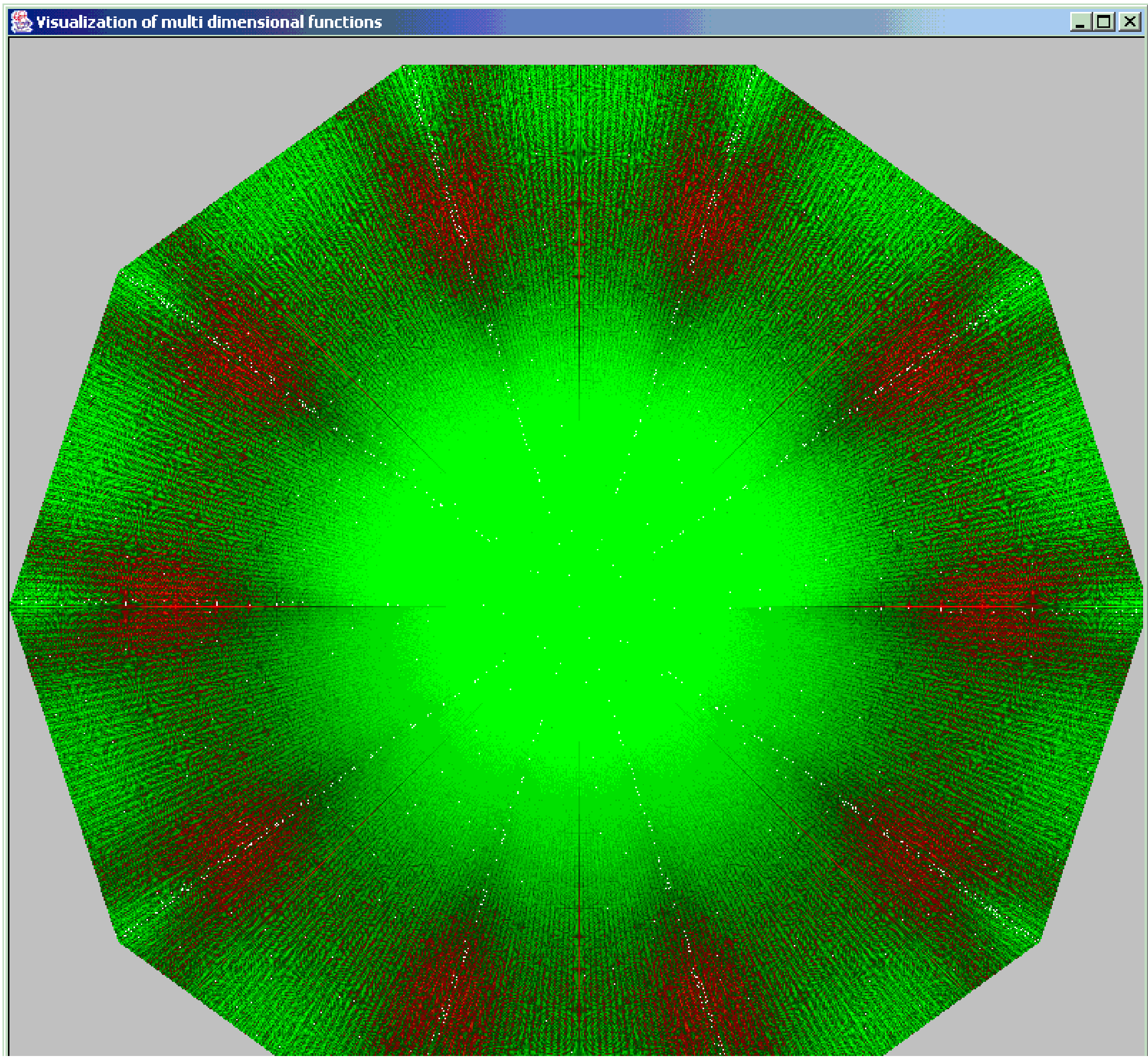


Figure 5.3: $\Sigma\cos(x^2)$ function in five dimensions (range –20 to 20)

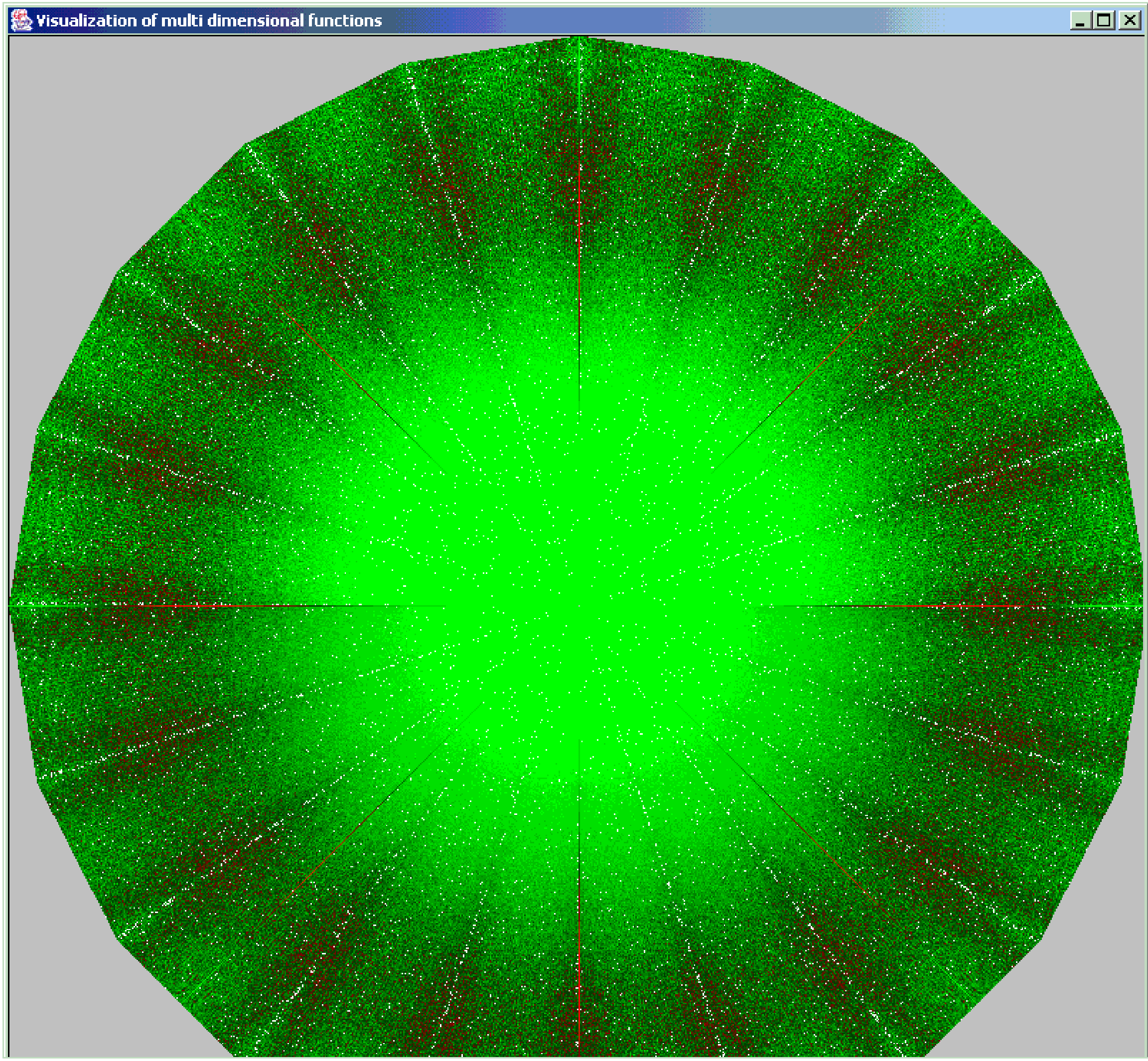


Figure 5.4: $\Sigma\cos(x^2)$ function in ten dimensions (range –20 to 20)

5.1.2 Analysis of dataset

A dataset containing functioning values for varying values of four parameters was generated using ‘Matlab’. The dataset can be considered to emulate a system’s behavior. The range of the parameters is between 1 and 10 with approximate maximum and minimum as 212 and -95 respectively. The system behaves in a different way for different ranges of the individual parameters. The aim is to understand the optimum range of each parameter that yields the desired performance. The red regions are indicative of non-acceptable performance while the green regions symbolize acceptable system performance. The overview [Fig 5.5] provides a good estimate of regions where the system shows the desired performance and reveals the following

- Tooltip feature reveals that the functional space consists of predominantly unacceptable performance values.

- Parameter1 is a very influential parameter for acceptable performance values. This can be deduced because all the other slices predominantly contain shades of red.
- The acceptable system values arise when parameter1 values range from 7-10
- The acceptable performance region is fairly wide in terms of the number of values providing good tolerance values while designing the system

Acceptable values based on the constraints of the other parameters can be seen using the highlighting feature.

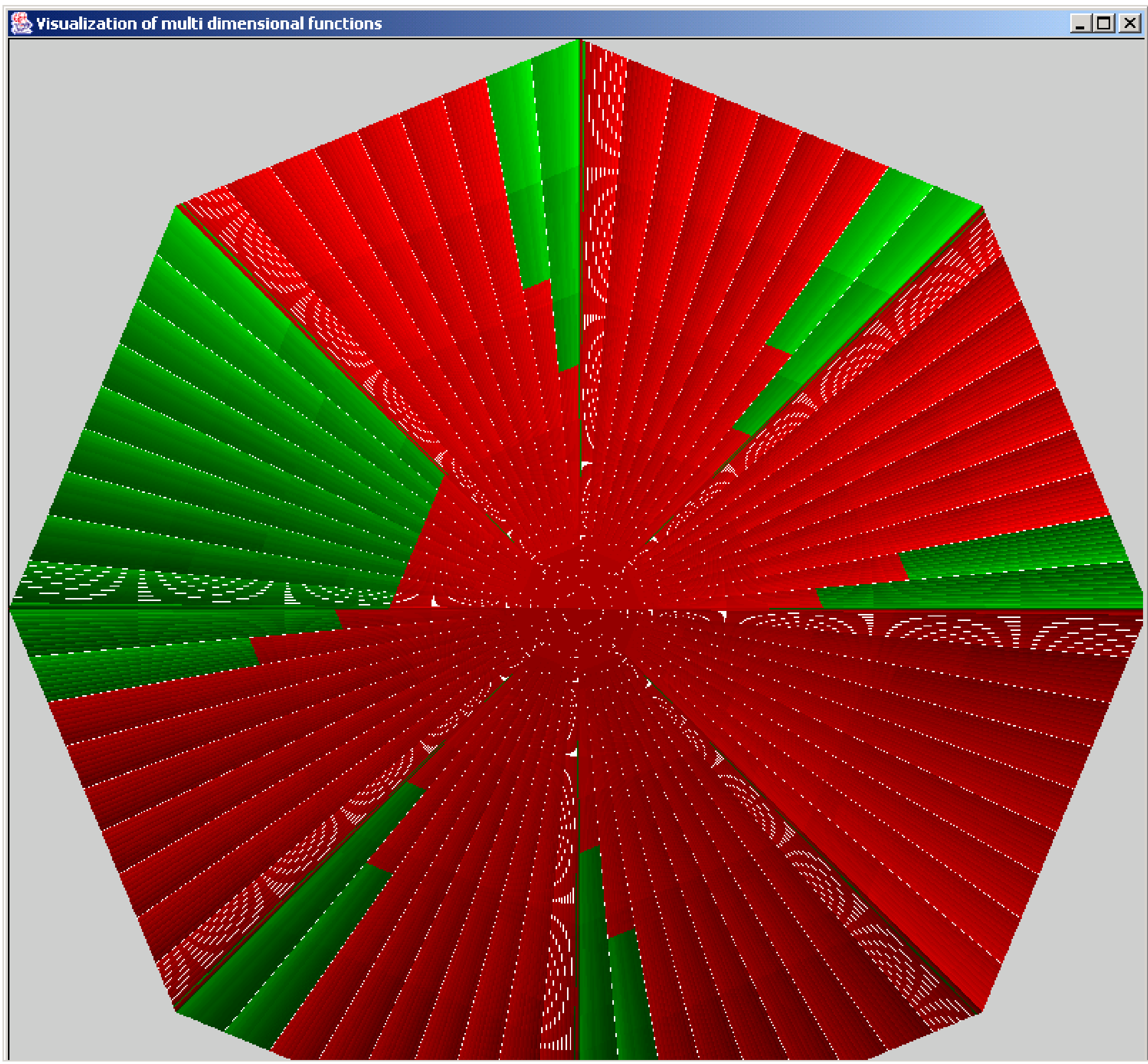


Figure 5.5 :Overview of the space for the dataset generated using MATLAB

5.1.3 Physics application

The collision of two particles is given by the equation

$$V = m * u * \sin(\Theta) \;/\; [\, M * \sin(\Phi + \Theta)]$$

where V is the velocity of the target particle, m is the mass of colliding particle, u is the velocity of colliding particle, M is the mass of the stationary particle, Θ is the angle of deflection of colliding particle and Φ is the angle of deflection of target particle. The collision scenario between two particles is shown in Fig 5.6.

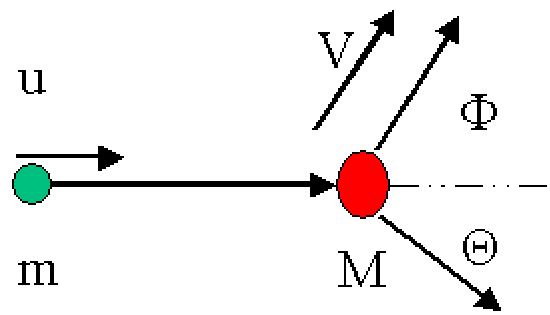


Figure 5.6: Collision between two particles

The overview of the functional space is shown is Fig 5.7. The overview reveals the effect of the various parameters on the final velocity of the stationary particle.

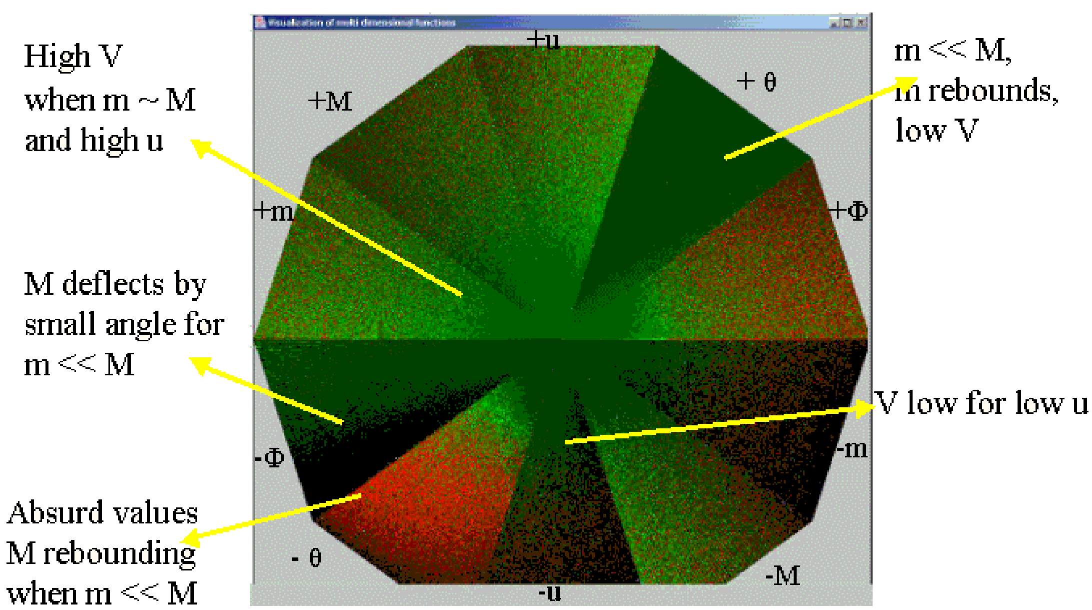


Figure 5.7 Overview of functional space

For example high initial velocity of the colliding particle with m ~ M or m >> m causes high final velocities [Fig 5.7]. Tolerance ranges for the colliding masses can be estimated from the overview for designing experiments.

5.2 Future Applications

The PolarEyez solution can be applied practically to a couple of interesting applications in biology and economics to provide an understanding of the multi-dimensional space. However at this stage more features described in the earlier chapters need to be implemented and incorporated into the visualization software. Hence the applications and the promises of PolarEyez are described in this section.

5.2.1 Bifurcation theory

Bifurcation is a qualitative change in a system caused by the variation of some physical parameter,λ [Seydel99]. The system can be characterized by a mathematical model. The behavior of the system varies drastically at the critical point λ_0, which is the point at which the system undergoes bifurcation. For example in a system for values of $\lambda < \lambda_0$ the system exhibits symmetric behavior and for $\lambda > \lambda_0$ the system shows asymmetric behavior.

A simple example is a wooden board supported at both ends with a load on top of it. The board will retain its material properties (such as stiffness *K*) as long as the load is below a threshold value. For small variations of the load the board may undergo small perturbations, but for large variations the board gets deformed. This threshold value is important as it controls the transition from a stable to an unstable state. In this example the load is the crucial parameter of the system and is treated as the bifurcation parameter λ. The load λ and stiffness *K* are examples of parameters.

The outcome of any experiment or any event is controlled by parameters. Varying a parameter can result in a transition from a quantitative change to a qualitative change. The practical problem is to control the state of a system by finding parameters such that the state fulfills the requirements. Table 6.1 gives a few examples of systems and the corresponding λ parameter.

System	**Bifurcation parameter λ**
Combustion	Temperature
Nerve impulse	Generating potential
Vibration of an engine	Frequency

Table 5.1: Examples of systems and their corresponding λ parameter

Patterns of inputs can affect a system causing a qualitative change. Tasks that are of importance in bifurcation study are

- Effect of different input patterns on system behavior. For example some patterns may be more robust and harder to disturb than another pattern that may be highly sensitive and easy to excite
- Identifying stable and unstable states
- Values of parameters for which the system is most sensitive

PolarEyez can be used in this application in the following way. As described earlier a system is characterized by multiple parameters. Solving the first differential of the function yields all the zeros of the system. The system undergoes a qualitative change at the zeros. An important task is to find the zeros which are responsible for a qualitative change in the system. In particular the value of the bifurcation parameter that causes the change is required. The overview of the functional space can be obtained using PolarEyez is obtained by supplying the function and the range of the individual parameters. The overview helps in detecting variation in the system behavior. Next focal point can be moved to the zeros of the function, producing a localized detailed view (focus+context) of the space. The new view is explored to identify abnormal system variations and also the value of the bifurcation parameter using the highlighting and mouse move over features. Similarly the focal point can be changed to other zeros of the function to identify the best bifurcation point.

5.2.2 Rubber Formulation application

Models can be generated from experimental data through mathematical procedures such as regression analysis. One such application is in the tire industry formulation of rubber is an important task since it affects the mechanical properties of rubber [Sevastyanov01]. The properties of rubber can be improved by varying the quantities of the ingredients namely Sulphur(S), Antidegradant(A), Filler(F) and Oil(O). Two of the mechanical properties Flex and Hardness depend on the ingredients according to the following equation:

$$\text{Flex} = 0.668 - 0.0014 * F^2 / O - 0.0147 * S / A$$

$$\text{Hardness} = 56.58 + 0.063 * F^2 / O - 44.6 * O / (S * A * F) + (0.0017 / S) * A * O^2 - 0.006 * S / (F * O^2) - 0.0052719 * F / S^3$$

The ranges of the parameters can obtained from experimental data.

The experimental data can be fed to the PolarEyez visualization. The properties flex or hardness are the functional values. The overview shows how the ingredients affect the flex or hardness property. Typical tasks that can be accomplished with PolarEyez are

- Which of the ingredients has the maximum effect on flex/hardness?
- What combination of ingredients yield acceptable values of flex/hardness?
- What are the tolerance ranges of the ingredients that yield optimum values of flex/hardness?

5.2.3 Applications in Economics

Several models in economics modeling business relationships consist of multi-dimensional functions. An example is the Black-Scholes formula [Scholes03] which is used for pricing financial options. A stock option is a contract that gives a person the right, (not the obligation) to buy or sell a stock at a pre-specified price (the exercise price) for a pre-specified time, that is, until the option expires. If the option gives the right to buy shares of a stock, it is a call option. If the option gives you the right to sell shares of a stock, it is a put option. Exactly how much should be paid for these contracts is determined using the Black-Scholes Formula. The formula relates five parameters to the price of a call option as follows

$$C = S * d^{-t} * N(x) - K * r^{-t} * N * (x - \sigma * \sqrt{t})$$

where $x = \log(s * d^{-t} / (K * r^{-t})) / \sigma * \sqrt{t} + ½ * \sigma * \sqrt{t}$

S is the currently underlying asset price (in dollars)
K is the strike price (in dollars)
t is the current time-to expiration (in years)
r is the riskless return (annualized)
d is the payout return (annualized)
σ is the underlying asset volatility (annualized)

Other examples [Business03] in economics are as follows:

- Econometric models are non-linear multi-dimensional equations dependent on variables such as inflation, GDP growth, productivity growth, population growth, consumer spending, unemployment rate, and time of year.
- In the oil industry the outputs of refining processes are described by multi-dimensional equations with independent variables such as temperature, pressure, and an array of chemical concentrations. Efficient refining management requires managers to understand the relative costs, benefits, and risks of alternate production strategies.
- In finance valuation models for derivative financial instruments are in the form of non-linear multi-dimensional equations.

5.2.4 Other Applications

Several other applications [Business03] involving multi-dimensional functions are as follows

- Modeling propagation of weather patterns in meteorology
- In medicine biological functions such as blood pressure are non-linearly dependent on multiple factors such as temperature, age, time of day, and chemical concentrations in the blood.
- All non-trivial sociological models of group behavior are dependent on a multitude of variables such as gross population, population density, age, and education.
- In chemistry the behavior and products of chemical reactions are dependent on many variables such as temperature, pressure, and chemical concentrations.

- In physics many phenomena are described by non-linear equations having more than two independent variables. For example, the strength of a magnetic field at the center of a solenoid is non-linearly dependent on the number of turns in the solenoid, its diameter, and the amperage applied to the wire.

Chapter 6: FORMATIVE EVALUATION AND RESULTS

6.1 Introduction

Eliciting feedback from the users through usability studies is an important task for the success of a visualization approach. Feedback helps in understanding the effectiveness of the mapping strategy from a n-dimensional space onto a two-dimensional space and the course of future developments for refining the approach. Informal studies were conducted during the course of development to understand the users viewpoint. A formal usability study was conducted at the end of the first phase of development using PolarEyez. The objectives of the study, the methodology of evaluation and the feedback obtained are elaborated in this chapter. A comparative study with another tool developed by Mihalisin et. al. is also described.

6.2 Objectives of the study

Informal studies during the development indicated the need for additional features. At the end of the software development of PolarEyez a formal usability study was designed. The fundamental objective of the study was to understand the efficacy of the mapping and the other features that were required to make PolarEyez more applicable in the users' areas of interest. The emphasis of the study was in learning about user's expectations rather than evaluation. The feedback would help guide future research towards improvement of the tool.

The study was aimed at addressing the following:

- How effective and easy is the mapping to understand?
- Is an overview of the space being provided?
- Are all dimensions are being represented equally?
- Is the approach scalable

Tasks were developed to estimate how well the claims were satisfied.

The audience for the tool is technical people who often use multi-dimensional functions for analysis. Hence, the participants for the study were selected from the Engineering and Mathematics departments.

6.3 The study

The study was divided into seven sections. The seven sections were:

- User Demographics.
- Demo.
- Task Set based on a known function.
- Task Set based on an unknown function.
- Quantitative rating based on Sections 3 and 4.
- Comparison of PolarEyez with Mihalisin.

- Overall impression of the tool.

The questions in every section were a mixture of open ended questions as well as quantitative rating questions. The Likert scale (range: 1 – 7) was chosen for the rating questions with 1 representing a low rating and 7 a very high rating.

Section 1 was targeted at user demographics. It was also used to understand the extent to which users had used multi-dimensional functions and the kind of tasks they had done while analyzing such functions. We also wanted to know the existing tools that they had used and the missing features in them. The answers would reveal what kind of features we would need to incorporate in our tool based on the type of tasks that users performed with multi-dimensional functions. The questions in the user demographics section are:

- How have you used multi-dimensional functions?
- Two practical areas where you have used multi dimensional functions?
- How often have you used multi-dimensional function?
  Often / at times / rarely
- Have you had any problems analyzing multi-dimensional functions? If yes, what are they?
- Please list the tools that you use(d) for analyzing or visualizing multi-dimensional functions.
- State the most useful features, and missing features you would like.
- In general, what is your opinion about visualizing multi-dimensional functions?

In Section 2 the mapping strategy and coloring scheme for the visualization was explained using a series of powerpoint slides. A demo of the visualization was also shown to the user using the simple function $\sum x^2$ in five dimensions. The properties of the function were explained using the visualization. The choice of the function was a simple one so that users would not be influenced by the demo given to answer the questions in the remaining sections.

In Section 3 the visualization of the $\sum \cos(x^2)$ function in 3, 5 and 10 dimensions was shown to the user and were asked to answer the following questions using the various features of the tool. Users were not aware of the function entered while answering these questions.

1) What is the maximum value of the function? What is the minimum value of the function (you can use the navigation features)

2) a) Do you see any patterns? What kind of patterns do you see?
   b) do you think the patterns repeat?

3) a) Do you see any similarity between the opposite slices
   b) Do you think adjacent slices of the visualization are similar / do a pair of opposite slices have any similarity with any other pairs.

After answering the above questions the function was revealed to the user. They were asked to find out properties of the function that could be deciphered from the visualization. The rationale behind this section is that the function being a simple one, users would be able to think of its properties and understand the visualization better. Also, users take time in getting acquainted with the system. This section helps users in familiarizing themselves with the system before moving on to sections with unknown functions. The last part of this section pertained to gauging the scalability of the function. Users were asked to rate how well they saw the properties of the function in the 5D and 10D case.

Section 4 pertains to the visualization of an unknown function. In this case the sampled function values are the inputs to the tool rather than the function itself. Users were asked to observe the visualization and asked to answer questions regarding what they observed.

Section 5 consisted of a number of rating questions related to the features and the effectiveness of the visualization based on the visualization in the previous two sections. Users were asked to add comments and the basis of their choices as well.

In Section 6 a tool developed by Mihalisin et.al. for visualizing multi-dimensional functions was explained to the user. The users were asked to experiment with the tool so that they could become comfortable in using it. The tool based on Mihalisin's approach was written by Dr. Chris North in Visual Basic [Fig 6.1]. The tool supported only the tool tip feature though the approach supported features such as the permute tool and detailed views. Therefore users were given the same tasks as given for PolarEyez in Section 5. For evaluating the two approaches participants required observation of the visualization and usage of just the tool tip feature. Another reason for not involving very elaborate features for the tasks was to focus on the comparison between the two approaches.

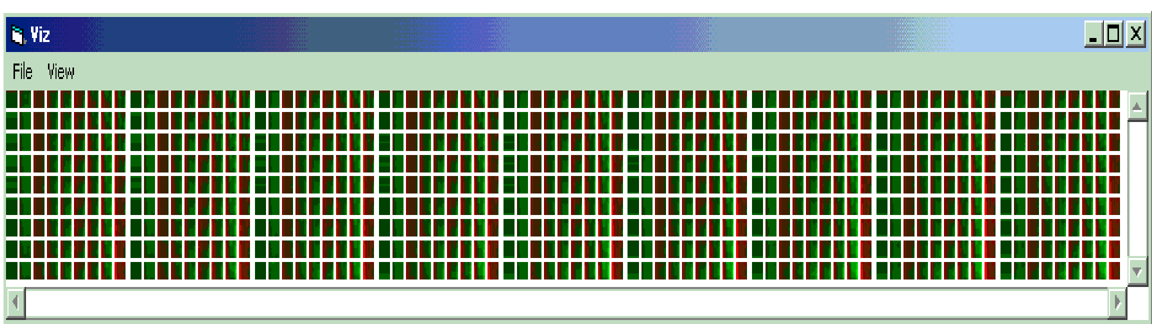


Figure 6.1 Visualization of a 5 D data set with parameters ranging between 1 and 9.

Finally, the users were asked to rate PolarEyez and Mihalisin's tool for overview of the space, extent to see details and ease of identifying properties of the function. The aim of showing another approach was to obtain unbiased feedback and to provide a reference point for comparison.

In the last Section, users were asked to comment on the usefulness of the tool. They were asked open ended questions to gauge how well they liked the tool.

The study requires 3-5 subjects since the objective was to get constructive feedback and suggestions from the users. The users were required to have prior knowledge and experience in using multi-dimensional functions.

6.4 Results

The feedback from the users provided valuable insight on the approach and missing features. Users were comfortable in understanding the PolarEyez as well as the Mihalisin's approach.

We restricted the number of candidates to three because of the duration of the study (about two hours) and also because of the limited availability of subjects with a knowledge of the problem domain. The three participants were either Masters or Doctoral students from the mathematics and engineering departments. All the users had often used multi-dimensional functions in their area of specialization. They felt that visualizing multi-dimensional functions is a difficult yet interesting topic. The participant from the mathematics department used multi-dimensional functions for biological modeling and the other two participants used them for image processing and analysis. The users had used other tools such as MATLAB, Gnuplot, Candys, XPPAUT and CONTENT for analyzing multi-dimensional function spaces and so they were able to grasp the need for PolarEyez and its capabilities.

6.4.1 Positive Feedback

The study evoked a positive response about the conceptual mapping and the interactive features of the tool from the participants. They considered PolarEyez as a prospective tool for analysis in their area of specialization. Their impressions are reflected in their ratings. The pros of the tool according to participants are discussed below.

- **Mapping Strategy**
  The mapping strategy used was clear to the participants and they were able to relate it to the visualization. For example in Part 3 of the study users were able to decipher the patterns in the visualization around different focal points.

- **Overview**
  Participants showed enthusiasm in guessing an unknown function ($\sum\cos(x^2)$ in three dimensions) based on the overview from the visualization. Participants were pretty successful in their attempt and were able to identify the presence of the cosine component. The overview also helped in identifying the properties of the $\sum\cos(x^2)$ function such as symmetry, sinusoidal nature and even property. Examination of individual slices in Part 4 of the study helped the participants in locating regions with

predominant positive or negative values around the maxima and minima of the function respectively.

- **Interactive Features**
  Participants were comfortable in using the features for navigating in the function space. The mouse moveover feature was employed to identify maximum and minimum values of the function. The highlighting feature helped in locating regions of interest satisfying a range of functional values. The ratings confirmed the usefulness of the features (mouse mover, highlighting, change of focal point) [Table 6.1]. The ratings were obtained after showing the visualization of a known and an unknown function.

Properties	Average Rating (1-7)
Highlighting	7
Change of focal point	6.3
Tooltip	7
Overview of the functional space	6.3
Equal representation of all dimensions	7
Ease of understanding the mapping approach	6.7
Effectiveness of the approach	6.3

Table 6.1: Average rating of the approach and the features of PolarEyez

6.4.2 Constructive Criticism

The study confirmed the need to incorporate additional features as indicated by informal feedback. The suggestions of the participants are discussed below.

- **Support for detailed view**
  PolarEyez provided a good overview of the functional space by which regions of interest can be identified by the participants. However there was no support for detailed views for further exploration. When showed the $\sum \cos(x^2)$ function in 5 and 10 dimensions users felt that from the 5D visualization the properties of the function could be deciphered but it was a bit difficult in the 10D case without the support of detailed views. A suggestion for detailed view was to expand selected pie slices by filtering other slices for a detailed view.

- **Color Scheme**
  One of the participant felt the need for a color scheme having increased number of color shades so that fine variations in the functional value can be discerned. The participant also recommended a provision for personalized coloring schemes. This was motivated by the fact that another tool MATLAB used red for positive functional values and green for negative values in contrast to PolarEyez.

- **Other Suggestions**
  Users also wanted a quantitative estimate of the number of points being highlighted. One of the users felt that the visualization was kind of noisy and distracting and so wanted to try out a different mapping scheme. Users wanted to select the point in the visualization to change the focal point. Also, the need for parameter based highlighting to locate points in the visualization was felt.

6.4.3 Comparison of Mihalisin and PolarEyez

The participants felt that Mihalisin helped in identifying specific values such as the maximum and minimum values of the unknown data set with lesser time than for PolarEyez. Mihalisin fared better in this regard because all combination of parameter values were arranged distinctively in a matrix like format. For locating maxima and minima PolarEyez required users to mouse over the visualization to find out functional values.

On the whole participants were not happy with the Mihalisin's visualization. They felt that PolarEyez provided a more compact and yet informative overview whereas the Mihalisin's representation was too clustered to identify patterns. Moreover, the revelation of patterns is possible only upon rearranging the parameters. Identifying dependencies of the functional value on the dimensions was not easy. Functional properties could not be deciphered from the Mihalisin's approach. It was felt that all dimensions are not given equal weights because the outer dimensions were more pronounced in the visualization as compared to the nested dimensions. Users identified that Mihalisin was not a scalable solution.

PolarEyez evoked a better response than the Mihalisin's approach. Table 6.2 shows the average rating for both the approaches with respect to the other approach.

	PolarEyez (1-7)	**Mihalisin (1-7)**
Clarity of the overview	6.7	4.3
Ability to see details	6.3	4.3
Ease of identifying patters/properties of the function	7	4.7

Table 6.2: Ratings for PolarEyez and the Mihalisin's approach

6.5 Conclusion

From the study we learnt that users required tools primarily to help them analyze multi-dimensional function spaces. The typical tasks that users performed were to find local maximum and minimums, locating points with a certain functional value such as zeros of a function and identifying optimal values for parameters. The opinion of the participants was that the overview provided by PolarEyez was powerful and helped in broadly locating regions of interest. The change of focal point and highlighting features further helped them in narrowing in on specific functional points. However, specific areas could not be explored further and a solution to their problem could not be arrived at, due to the absence of a detailed view. In summary PolarEyez showed promise in supporting user-specified tasks.

Chapter 7: CONCLUSIONS AND FUTURE WORK

7.1 Introduction

Every visualization technique is well suited for providing an insight for certain types of problems. The problems best suited for visualization with PolarEyez and its pros and cons are discussed in this chapter. Suggestions for further improvement based on its shortcomings are also provided. The chapter concludes with reiteration of the salient features of the technique and the contributions of the approach to research.

7.2 Pros

The major advantage of this visualization approach is an integrated overview of the entire functional space on all dimensions simultaneously. The overview helps users identify regions of high or low function values, frequency of particular values, clusters, etc. This relieves users from mentally integrating separate 2D slices in short-term memory as required by previous approaches. The polar nature of the overview enables users to view patterns in function value proceeding away from the focal point in all directions, and estimate distances to interesting phenomena. For example Fig 7.1 is the visualization of a four dimensional function $f(x_1,x_2,x_3,x_4)$. The problem is to identify regions with positive functional values for designing a system. The overview clearly reveals that positive functional values occur only when $x_1>=7$ and $x_1<=9$. Users can explore identified regions subsequently.

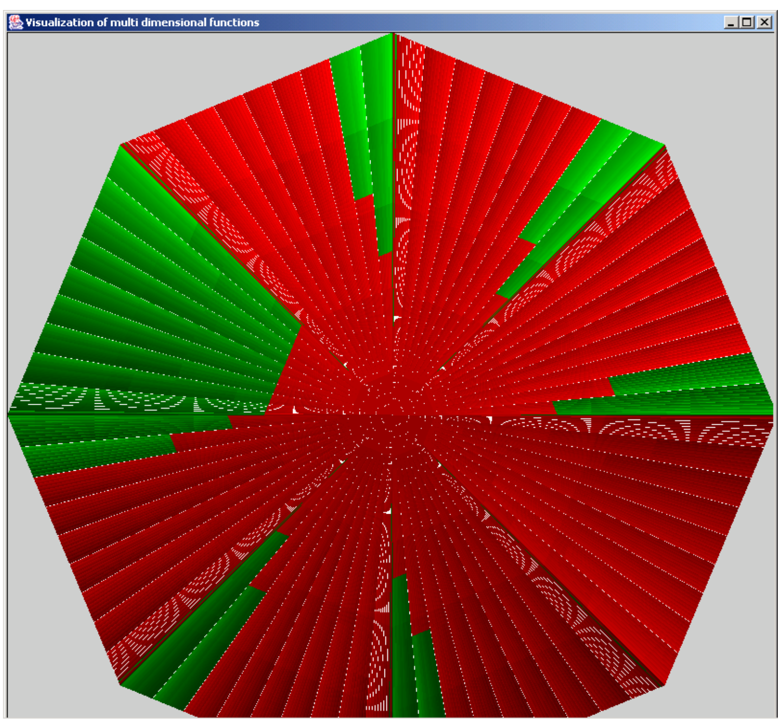


Figure 7.1 Visualization for a four dimensional function $f(x_1,x_2,x_3,x_4)$

Representing the overview from the perspective of a point in the space is somewhat egocentric and natural for users to grasp, as confirmed by informal feedback from test users. Interestingly, the flat edges of the polygon pie slices are helpful for orienting users and provide a natural analogy to the squares and cubes of heat maps. We had previously explored a completely circular layout, which caused confusion for users.

Another advantage is scalability. This approach scales up smoothly. Each additional dimension simply adds two more slices to the pie, gradually narrowing all slices. It scales to approximately 10 to 20 dimensions (20 to 40 pie slices). Beyond that, very narrow pie slices can still provide some useful information. Figure 5.1, 5.3, 5.4 shows the $\Sigma\cos(x_i^2)$ in 3,5 and 10 dimensions. As the number of dimensions increases the approach shows graceful degradation by the slices becoming narrower. For 2D functions, this approach reduces cleanly to a simple heat map. Furthermore, the number of dimensions d scales independently of the bounding range r of the dimensions. This is because dimensions and range are mapped to separable dimensions in the visualization. Dimensions are represented circumferentially, and dimension range is represented radially. The range for each dimension receives a screen-width (e.g. 1000 pixels) of space regardless of number of dimensions.

7.3 Cons

Every visualization approach possesses shortcomings. The various problems of the PolarEyez approach are described below.

- **Multi-valued functions**

The PolarEyez approach is not well suited for multi-valued functions because of the usage of a one-one mapping of color to functional value. When a functional data point possesses more than one unique functional value ambiguity arises as to which one of the functional values need to be considered. Multiple PolarEyez views, each showing one functional value can be used to solve this problem.

- **Slicing problems**

PolarEyez does not lend itself well for problems requiring certain parameters to be kept a constant with the remaining ones varying. This is especially the case with optimization problems. The overview provided shows the variation of all the dimensions simultaneously. Interactive features also do not provide for keeping certain parameters constant. Coupling of 2D plots as explained in section 3.5.5 can provide a possible solution.

- **Variation of parameters**

The overview shown by PolarEyez shows the variation of all parameters with one parameter varying linearly in every slice. However in any particular slice the variation of the remaining parameters is unpredictable. More order can be provided for in depicting the variation of the other parameters in a slice by improving the mapping strategy. Alternate face paths (such as spirals) can be investigated.

- **Coloring Scheme**

The present coloring scheme poses several problems. The current scheme does not distinguish functional values well as the range of functional values increases. This occurs because a wide range of functional values are mapped onto 256 shades of red or green. For example let us assume the functional value varying between 0 and 500. The positive functional values (500 values considering discrete functional values) are mapped to 256 shades of green. Approximately two points map to every shade of green if the functional values are discrete. However more number of functional values map to a single color value when the functional value is real valued (as when computing from the function). As the range of the functional values increases the problem becomes more severe.

Distribution of functional values also has a bearing on the visualization. For example considering a function with positive functional values and range of values between 0 and 1000, if the concentration of values is near the minimum value then the visualization appears black revealing no significant patterns.

Fig 7.1 shows the visualization of a function whose maximum and minimum values are 12053 and –66115 respectively. Since most of the functional values are in the range of 1000 to 5000 the visualization appears dark and patterns cannot be noticed easily.

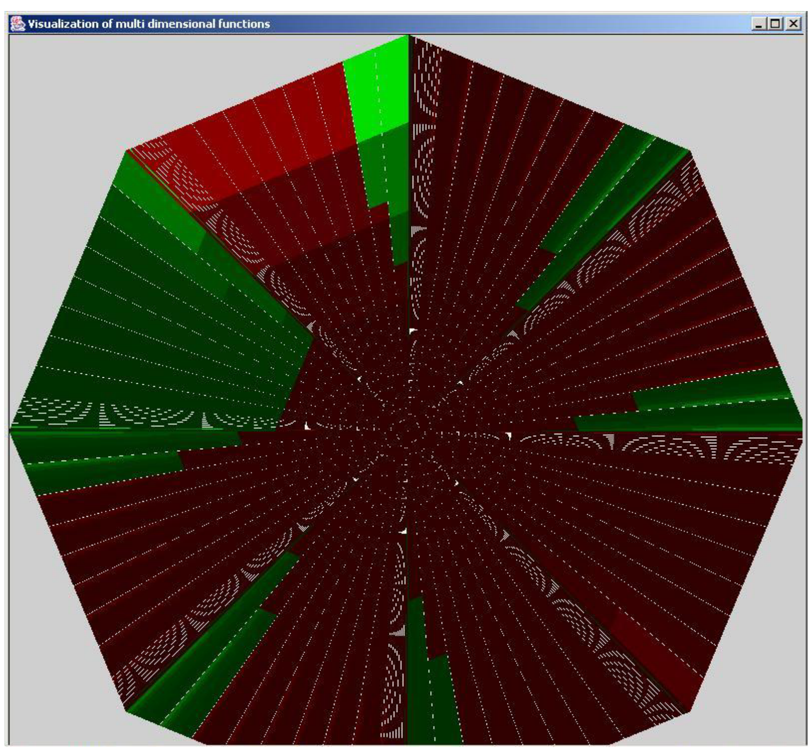


Figure 7.2: Visualization for a function with maximum and minimum as 12053 and –66115 respectively.

Fig 7.3 is another example of a function for the calculating the vibration amplitude of hammer foundations in civil engineering. The functional value ranges from 100 to 5000. The slices show positive shades of green with no gradients in color. Hence identification of regions satisfying constraints from the overview is difficult.

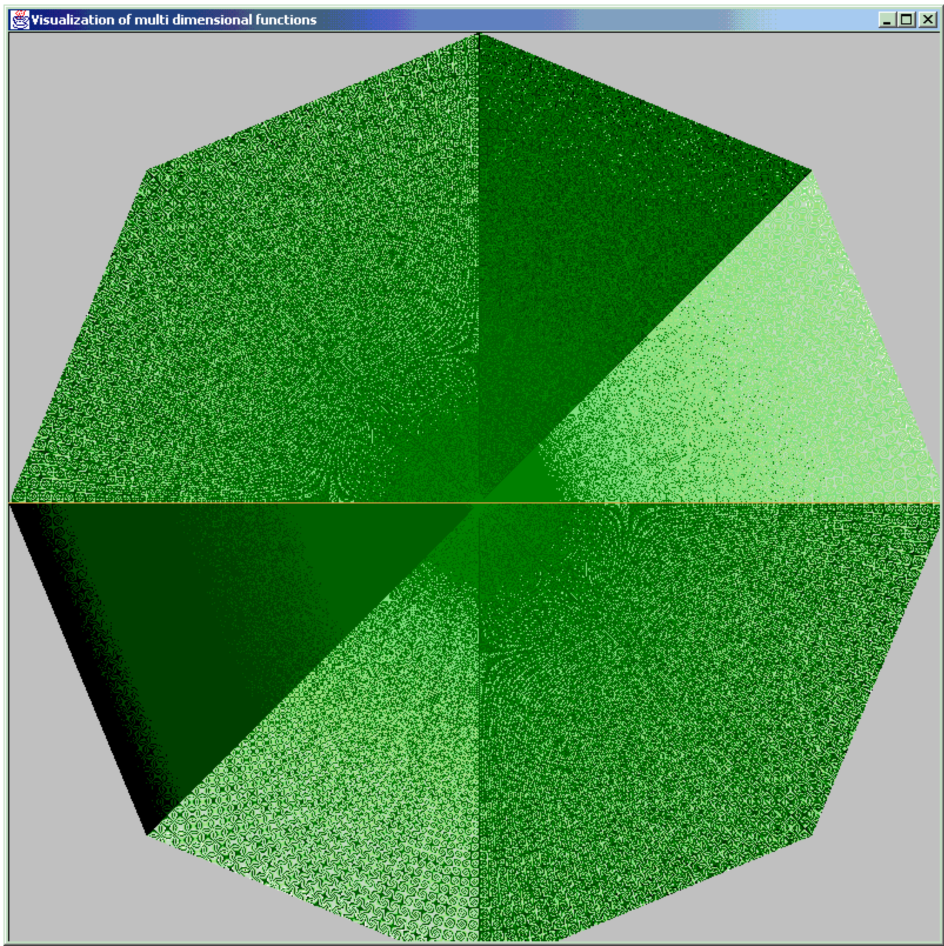


Figure 7.3: Visualization of the vibration amplitude of hammer foundations

Color gradients would be helpful in distinguishing between ranges of functional values. The present coloring scheme can be expanded to accommodate more shades of green and red. The functional values can be divided with each range of values being mapped onto a set of green/red color shades. Another solution is to assign colors based on the distribution of functional values. For example in the case of a functional space shown in Fig 7.1, more colors can be assigned for representing functional values between 1000 and 5000 in comparison to the rest of the functional values.

Another issue stems from the difference in coloring schemes among the different tools used for understanding and graphing multi-dimensional functions. Tools such as MATLAB use the exact opposite color scheme. Forcing users to adjust to the coloring schemes of different tools may cause confusion and may not be preferable. Hence users should be provided with the capability to select color schemes most comfortable to them.

7.4 Future Work

Improvement of any visualization tool is an evolutionary process with alternating stages of user studies and development. The current approach used by PolarEyez evolved from a radically different approach in which the focal point is stretched out into a horizontal line, and the rays are organized along the line pointing vertically was explored earlier [Fig 7.4] [Chennupati01]. This significantly expanded the detail of the focus region, at the expense of the extremities. However, users were too confused by this mapping, and could not overcome the belief that it was depicting 2D slices.

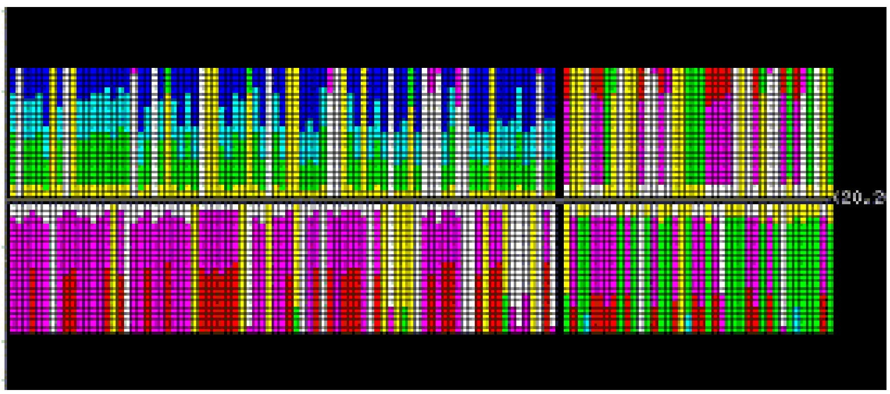


Figure 7.4: Stretching the focal point into a horizontal line, with rays extending vertically, to exaggerate the focus+context effect.

Hence an alternate approach of arranging the focal rays in a radial layout around the focal point was attempted. The focal point was also expanded in a fisheye-like [Keahey98] manner to show more details to the user. This alternate approach has been implemented in PolarEyez. The informal and formative user evaluations conducted on PolarEyez has provided valuable guidelines for future work. Possible future developments are outlined below.

1. Aggregation
   Aggregation needs to be implemented in PolarEyez. The current aggregation technique is a rudimentary one. More efficient and improved aggregation algorithms can be looked into as well.

2. Detailed views
   The focus + context approach provides a detail view within the overview with the degree of details fixed and limited. Hence a more detailed view is called for exploring functional regions of importance. It would be interesting to use PolarEyez as an overview for controlling separate 2D slices in an overview + detail fashion.

3. Extension of the functional domain

The types of functions that can be currently visualized are limited. More complex functions with differential and integral components can be included.

4. Variations in the mapping technique
   At present a linear scan line path is employed while mapping the pixel attributes to the functional coordinates. Other paths such as spiral path etc could replace the scan line path to make the visualization more intuitive to the user. Other ray arrangement techniques apart from the radial arrangement used here can be explored. Alternate paths and ray arrangements could reduce aggregation issues and make the process easier.

5. Multiple foci
   Another interesting feature would be to see multiple foci in the visualization so that views of different functional regions can be obtained.

6. Support for querying
   Complex querying features (for example highlighting regions containing largest number of points with a certain functional value) can be introduced to help users start the exploration of the space.

7. Other features
   Filtering of dimension by removing unwanted slices can be implemented. Statistical estimates such as percentage of negative and positive values, mode of the values in a particular region, can be presented to the user.

7.5 Conclusions

This research work contributes a novel layout and navigation strategy for visualizing multi-dimensional functions. A polar focal point with rays emanating in all directions compress the multi-dimensional space into a circular layout. The approach is advantageous for identifying regions for further exploration and scrutiny by providing an overview of the functional space. Other tools such as World Within Worlds, HyperSlice and Matlab are helpful in providing detailed views of the regions selected in the overivew.

The key characteristics of the approaches are:

- Provides an integrated overview of the entire bounded function space, from the natural perspective of a point within space.
- Treats all dimensions uniformly, without employing conventional slicing schemes, enabling visualization of variation in all dimensions simultaneously.
- Provides focus + context, with smooth seamless transition from detail to overview.
- Smoothly scales up to 10-20 dimensions, with approximately 1000 pixel range on each dimension.

APPENDIX A: USAGE OF POLAREYEZ SOFTWARE

A.1 Installation of Java

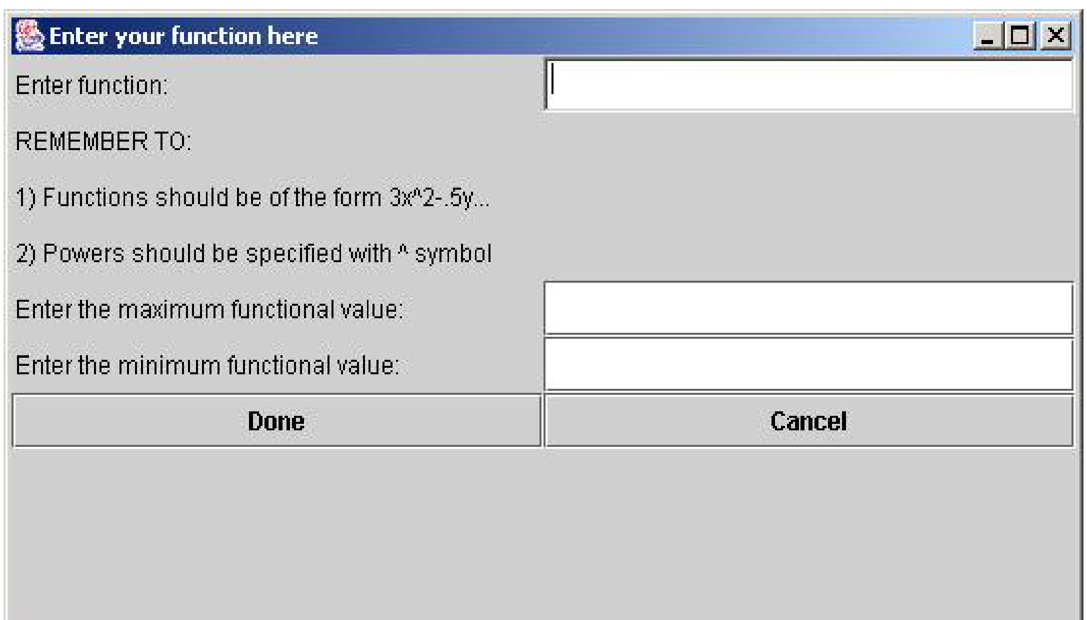
Java version 1.3 is required for running the PolarEyez application. All the PolarEyez software needs placed in a directory. The path of the corresponding directory needs to be provided in the java classpath.

A.2 Compilation and execution of the PolarEyez software

The software is compiled by compiling only the features.java file followed by the execution of the file. The remaining files are packages and are called by the features.java file.
Command for compilation: javac features.java
Command for execution: java features

A.3 Inputs

The inputs to the application are the function and its approximate maximum and minimum values.



A.4 Interaction

The visualization corresponding to the function is generated. Separate windows are provided for each one of the interaction features for navigating the functional space.

APPENDIX B : QUESTIONNAIRE USED FOR USER STUDY

Part 1: User Demographics

1. Name

2. Education / Job

3. How have you used multi-dimensional functions?

4. Do you know two practical areas where you have used multi dimensional functions?

5. A) How often have you used multi-dimensional function?

   Often / at times / rarely

6. Have you had any problems analyzing multi-dimensional functions? If yes, what are they?

7. Please list the tools that you use(d) for analyzing or visualizing multi-dimensional functions.

8. State the most useful features, and missing features you would like.

10. In general, what is your opinion about visualizing multi-dimensional functions?

Part 2: Demo

The mapping strategy is explained to the user with a series of powerpoint slides.
A demo of the tool with an example is then given

Part 3: Task Set 1

Please look at the visualization and answer the following questions. You are free to use the interactive features of the tool to answer any of those questions.

1. What is the maximum value of the function? What is the minimum value of the function (you can use the navigation features)

2. A) Do you see any patterns? What kind of patterns do you see?

   B) do you think the patterns repeat?

3. A) Do you see any similarity between the opposite slices

   B) do you think adjacent slices of the visualization are similar (OR) do a pair of opposite slices have any similarity with any other pairs.

4. Navigate the focal point to (13,0,0).
   a. Do you see the positive slice of parameter 1 being shortened?

   b. Move the focal point to (0,13,0). Does the same effect occur for parameter 2?

Please stop here for a moment before answering the next question for some additional information

5. List the properties of the function you notice from the visualization?

Please stop here for a moment before answering the next question for some additional information

6. Are you able to see the same properties as seen for 3-dimensional case for a)5D case b)10D case. Please rate on a scale of 1 (same properties cannot be deciphered) - 7 (the same set of properties are clearly seen as in the 3D case)

Part 4: Task Set 2

Please look at the visualization and answer the following questions. You are free to use the interactive features of the tool to answer any of those questions.

1. What is the maximum and minimum value of the function

2. Which triangular slice(s)has predominant positive values.

3. Which triangular slice(s)has predominant negative values

4. If the answer to the above question is more than one, can you decide on only one slice which has the most number of negative values.

5. Which parameter contributes to maximum number of negative values and for which value of this parameter

6. Does the functional space consist of predominantly positive or negative values.

Part 5: User Feedback based on the two visualizations shown

Having seen the two visualizations shown earlier answer the following questions. You are free to provide any comments for any of the questions

1. Features:
    a) How useful do you think is the change of focal point (scale 1-poor,7-good)

    b) How useful do you think is the highlighting feature (scale 1-poor,7-good)

    c) How useful do you think is the mouse moveover feature (scale 1-poor,7-good)

2. Do you think a good overview of the space is obtained from the visualization (scale 1-poor,7-good)

3. Do you think that all variables are represented equally on the visualization (scale 1-poor,7-good)

4. How easy is the mapping to understand? (rate on a scale 1-very tough,7-easy)

5. Were you able to see the properties you wanted to see in the case of the cos(x^2) function

6. How well do you rate the tool in revealing the properties that you expected in the case of the cos(x^2) function? (scale 1-poor,7-good)

7. How effective do you think the tool is on the basis of the two visualizations shown (1 poor –7 good)

Part 6: Tool A and Tool B

Please use the visualization provided to answer the following questions

1. What is the maximum and minimum value of the function

2. Which parameter contributes to maximum number of negative values and for which value of this parameter

3. Does the functional space consist of predominantly positive or negative values.

For each of the questions below rate using the scale 1 (poor) – 7(good)

	Tool A	Tool B
Overview of the functional space		
Extent to see details		
Ease of identifying patterns /properties of the function		

4. Any feedback regarding the pros and cons of the two tools

Final set of questions

1. What kind of tasks do you typically want a tool for visualizing multi-dimensional functions to do?

2. Would you or others like to use this tool in your area of specialization? (1 never –7 definitely)

3. Overall opinion of Polareyez please rate on a scale of 1-poor, 7-good

References:

[Ahlberg94] Ahlberg, C., Shneiderman, B., "Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays", *Proc. ACM SIGCHI 1994*, pp. 313-317, (1994).

[Ahlberg95] Ahlberg, C., Wistrand, E., "IVEE: An Information Visualization and Exploration Environment", *Proc. IEEE Information Visualization Symposium '95*, pp. 66-73, (1995).

[Aldenderfer84] Aldenderfer, M.S., and Blashenfield, R.K., "Cluster analysis", Sage Publications (1984).

[Alpren91] Alpern.,B., Carter., L., "Hyperbox. In Gregory M. Nielson and Larry Rosenblum", editors, *Proceedings of IEEE Visualization '91*, pages 133–139, San Diego, California, October 1991.

[Becker87] Becker, R., Cleveland, W., "Brushing scatterplots", *Technometrics*, 29(2): 127-142, (1987).

[Bentley96] Bentley, C.L., and Ward, M.O., "Animating multi-dimensional scaling to visualize n-dimensional data sets," *IEEE Information Visualization '96*, pp. 72-73, October, 1996.

[Business03] Business Animation, [WWW Document] http://www.businessanimation.com/ASP/WhitePaper.htm#_Definition_of_terms, (accessed January 2003)

[Card99] Card, S., Mackinlay, J., Shneiderman, B., *Information Visualization: Using Vision to Think*, Morgan Kaufmann, (1999).

[Chennupati01] Chennupati., B., Pande., A., Vatrapu., R., Hanumolu., V., North., C., " A Focus+Context Approach for Visualizing Multi-dimensional Functions", Virginia Tech, 2001.

[ChungWong97] P. C. Wong and R. D. Bergeron. 30 Years of Multi-dimensional Multivariate Visualization, pages 3-33. IEEE Computer Society Press., Los Alamitos, CA, 1997

[ComponentAnalysis03] Institute of Phonetics Sciences of the University of Amsterdam, [WWW Document], http://www.fon.hum.uva.nl/praat/manual/Principal_component_analysis.html, (accessed January 2003)

[Davison92] Davison, M.L, "Multi-dimensional scaling", Krieger Publishing (1992).

[Feiner90] Feiner, S., Beshers, C., "Worlds within Worlds: Metaphors for Exploring n-Dimensional Virtual Worlds", *Proc. ACM UIST '90*, pp. 76-93, (1990)

[Goel99] Goel, A., http://csgrad.cs.vt.edu/~agoel/parallel_coordinates/stf/table9.stf

[Grinstein01] Grinstein., G., Trutschl., M., Cvek., U., "High-Dimensional Visualizations", 7th Data Mining Conference-KDD 2001, San Francisco, California, 2001.

[Hoffman99] P. Hoffman and G. Grinstein, "Dimensional Anchors: A Graphic Primitive for Multi-dimensional Multivariate Information Visualizations," presented at NPIV '99 (Workshop on New Paradigms in Information Visualization and Manipulation), 1999.

[Humphrys97] Humphrys, M. (1997). http://www.computing.dcu.ie/~humphrys/ai.html

[Inselberg97] Inselberg, A., "Multi-dimensional Detective", *Proc. IEEE Information Visualization Symposium'97,* pp. 100-107, (1997).

[Jayaraman02] Jayaraman,S., North, C., "A Radial Focus+Context Visualization for Multi-dimensional Functions", *Proc. IEEE Visualization Symposium'02,* pp. 443-450, (2002).

[Kandogan00] Kandogan, E., "Star Coordinates: a Multi-dimensional Visualization Technique with Uniform Treatment of Dimensions", *LBHT Proc. IEEE Information Visualization Symposium 2000*, pp. 9-12, (2000).

[Keahey98] Keahey, A., "The generalized detail-in-context problem", *Proc. IEEE Symposium on Information Visualization*, IEEE, (October 1998).

[Mihalisin91] Mihalisin, T. Timlin, J., and Schegeler, J., "Visualizing multivariate Functions, Data, and Distributions", *IEEE Computer Graphics and Applications*, 11(13), pp. 28-35, (1991).

[Orbital02] Orbital Imaging Corporation, [WWW Document], http://www.orbimage.com/news/img_week/image07.html, (accessed December 2002)

[Rao94] Rao, R., Card, S., "Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information" *Proc. ACM CHI'94*, pp. 318-322, (1994).

[Roettger98] Roettger, S., Heidrich, W., Slusallek, P., Seidel, H.. "Real-Time Generation of Continuous Levels of Detail for Height Fields" In V. Skala, editor, *Proc. WSCG '98*, pages 315-322, (1998.)
http://wwwvis.informatik.uni-stuttgart.de/~roettger/html/Main/terrain.html

[Scholes03] Black Scholes Formula, [WWW Document], http://in-the-money.com/glossarynet/Black-Sc.htm, (accessed January 2003)

[Sevastyanov01] Sevastyanov, V. " New Method for Multi-dimensional Functions", Multistat, (2001).

[Seydel99] Seydel, R., "World of Bifurcation. Online Collection and Tutorials of Nonlinear Phenomena", *WOB99*, (1999)

[Shastri02] Shastri., P., Kulkarni., M., Mahajan., R., North., C.," Visualization of Multi-dimensional Functions", Virginia Tech, 2002.

[Spence01] Spence, R., *Information Visualization*, Addison-Wesley, (2001).

[Spotfire02] SpotFire, http://www.spotfire.com/products/fg.asp (accessed 2002)

[Sun02] Sun's Java 2 SDK, Standard Edition Documentation, [WWW document], http://java.sun.com/j2se/1.4.1/docs/index.html, (accessed January-May 2002).

[Twedie96] Tweedie, L., Spence, R., Dawkes, H., Su. H., "Externalizing Abstract Mathematical Models", *Proc ACM CHI'96*, pp. 406-402, (1996).

[VanWijk93] Van Wijk, J., Van Liere, R., "HyperSlice: Visualization of Scalar Functions of Many Variables", *Proc. IEEE Visualization '93*, pp. 119-125, (1993).

[Weins02] Weins, G. Elmer, (2002). http://www.egwald.com/economics/productionfunctions.php

[Williamson92] Williamson,C. and Shneiderman B. The Dynamic HomeFinder: Evaluating Dynamic Queries in a Real-Estate Information Exploration System. In Proc of ACM SIGIR 92 (June 1992), 338 -- 346

[Wong97] Wong.,P.C., Bergeron.,D., "Multivariate Visualization Using Metric Scaling", *Proc. IEEE Visualization '97*, October 1997, pp. 111-118.

Resume

SANJINI JAYARAMAN
sjayaram@vt.edu
http://csgrad.cs.vt.edu/~sjayaram

EDUCATION:

M.S. Computer Science, Expected December 2002
Virginia Tech, Blacksburg, Virginia. GPA: 3.7/4.00
B.E. Computer Science, May 2000
University of Madras, India. Overall Percentage: 86%, **University rank holder**.

EXPERIENCE:

- **Intern, Black and Decker, Corporate Headquarters,Towson,MD** May 2002-Aug 2002
  Worked in a team to develop an E-business solution. Responsibilities include requirement analysis, design of visualization software for viewing the tables, construction of pivot table reports and multi-dimensional cubes using Microsoft SQL Server and Analysis Services.
- **Research Assistant,** Center for Transportation Research, Aug 2001-May 2002
  Developed web applications in Cold Fusion utilizing the SQL Server databases.
- **System Administrator,** Vice President of Student Affairs - Virginia Tech , June -Aug 2001
  Developed and maintained the department website. Designed tables for a database containing student records and also developed it's web interface using **Javascripts.**
- **Teaching Assistant,** Dept. of Computer Science, Aug 2000 – May 2001.
  Debugging C++ data structure projects, designed test files for project evaluation.

PUBLICATIONS:

- Jayaraman, S., North, C., "A Radial Focus+Context Visualization of multidimensional functions," *Proceedings of **IEEE** Visualization,* October 2002.
- Published an evaluation report in **ACM SIGCHI** on current visualization projects.

PROJECTS:

- Designed and implemented a **Java** based software (*Polareyez*) for the visualization of large multi dimensional functional and data spaces using *focus+context* approach (Master's Thesis).
- Studied the following **wireless technologies** and conducted performance tests
  a) **IEEE 802.11b** WLAN's medium access protocol (CSMA/CA) in the DCF mode.
  b) **MANET** routing protocols (DSR and TORA) under varying loads and rate of mobility.
  c) Radio resource management subsystem in GSM systems, with an emphasis on handovers
- Designed a distributed file system and implemented two **protocols** (write share, update timeout) in **Java** for reducing consistency related communication in such systems.
- Designed and developed a tool in **Java** for **visualizing databases** based on *Snap* architecture.
- Designed a new model based on set theory for personalizing web based social networks. The project involved analysis and evaluation of the *PIPE* personalization approach.
- Conducted requirements analysis for the development of a Calendar Management System.
- Developed an **object oriented** design for simulating mobile agents and created test scenarios.
- Developed a web resource on **Virtual Private Networks** covering the Protocols, Hardware & Software resources and Encryption schemes. http://www.ari.vt.edu/ece5516/vpn/index.htm
- Designed and implemented an **Active Network** application called Virtual Exhibition in **Java**.
- Designed and implemented a text editor in **C**.

COMPUTER SKILLS:

Languages : C, C++, Java, Intel 8085/8086 Assembly language programming
Operating Systems : UNIX, Windows 95/98

Databases	: Oracle, SQL, SQL Server, Analysis Services
Others	: Cold Fusion ,HTML, JavaScript, Visio, VSE, Latex

ACTIVITIES AND HONORS:

- **Merit Scholarship** from the Dept. of Computer Science during under-graduation.
- **Member of the Editorial Board** of the monthly and annual school magazine for four years.
- Won prizes at collegiate paper presentations and athletic events.