

# The Contour Spectrum\*

Chandrajit L. Bajaj<sup>†</sup> Valerio Pascucci<sup>†</sup> Daniel R. Schikore<sup>‡</sup>

Shastra Lab & Center for Image Analysis and Data Visualization

Department of Computer Sciences

Purdue University

West Lafayette, IN 47907

{bajaj,pascucci,drs}@cs.purdue.edu

## ABSTRACT

We introduce the *contour spectrum*, a user interface component that improves qualitative user interaction and provides real-time exact quantification in the visualization of isocontours. The contour spectrum is a signature consisting of a variety of scalar data and contour attributes, computed over the range of scalar values  $w \in \mathfrak{R}$ . We explore the use of surface area, volume, and gradient integral of the contour that are shown to be univariate B-spline functions of the scalar value  $w$  for multi-dimensional unstructured triangular grids. These quantitative properties are calculated in real-time and presented to the user as a collection of signature graphs (plots of functions of  $w$ ) to assist in selecting relevant isovalues  $w_0$  for informative visualization. For time-varying data, these quantitative properties can also be computed over time, and displayed using a 2D interface, giving the user an overview of the time-varying function, and allowing interaction in both isovalue and timestep. The effectiveness of the current system and potential extensions are discussed.

**Keywords:** Visualization, Scalar Data, User Interfaces, Real-time Quantitative Query

## 1 Introduction

Exploratory visualization is an iterative process, with many visualization parameters to control. Without effective user-interface tools, the visualization user must rely on a-priori knowledge about the data of interest in choosing effective parameters. Informative and effective visualizations often mask the amount of time and effort which was required to create such a successful visualization. Recent approaches in user interfaces for controlling visualization parameters have aspired to provide the user with the tools to more rapidly choose parameters

which result in an effective visualization.

Isocontouring is an example of particular interest with a relatively simple parameter, the isovalue. Because an isocontour effectively shows only a subset of the data, modification of the isovalue and interactive display of one or more isocontours is often necessary to infer global structure of the scalar field from the display of contours. With the increasing size of typical datasets competing against the improved algorithms for computing contours [1, 4, 9, 11, 14], enhanced user interfaces for assisting in isovalue selection can dramatically decrease the cycle from data acquisition to effective visualization.

We present a user interface for isocontouring which presents the user with a collection of data characteristics to aid in the selection of significant isovalues. Characteristics such as surface area, volume, and gradient integral are shown to be univariate B-spline functions of the scalar value  $w \in \mathfrak{R}$  and are computed in real time. Presented with overlapping signature graphs, the user simply clicks in the scalar value dimension to select an isocontour level. Upon selecting an isovalue  $w_0$ , the interface displays the exact values of the characteristic measures (limited, of course, by the accuracy of the given sampled data), giving immediate quantification. For time-varying data, the user may interact in 1D with the scalar value dimension, or in 2D with both scalar value ( $w$ ) and time ( $t$ ) dimensions. In the 2D interface, a single bivariate characteristic function of  $w$  and  $t$  is displayed as a colormap, allowing the user to select significant  $w_0$  and  $t_0$  parameters. Through the display of data characteristics over the entire range of  $w$ , the user gains a sense of the global characteristics of the scalar field, and uses domain-specific knowledge to select isocontours of interest. We also suggest methods in which the computed metric properties can be used to automatically generate a set of significant isovalues  $w_0$ .

\*See also <http://www.cs.purdue.edu/research/shastra>

<sup>†</sup>After Sept 1, 1997: Department of Computer Sciences & TICAM, University of Texas at Austin, Austin, TX 78712.

<sup>‡</sup>Current Address: P.O. Box 808, Mail Stop L-561, Livermore, CA 94550, [schikore@llnl.gov](mailto:schikore@llnl.gov)

## 2 Related Work

Effective visualization interfaces and knowledge-based systems are increasingly important for rapidly creating useful visualizations from a constantly growing supply of data, which is also growing in size and complexity. A related example in scalar field visualization is that of selection of a transfer function, or colormap. He et al. introduce manual techniques for selecting transfer functions based on selective user refinement from an initial palette of functions, as well as automated techniques based on desired characteristics of the resulting images [8]. Characteristics explored include image entropy, image variance, and edge content. Bergman et al. developed rule-based criteria for selecting colormaps, based the spatial frequency of the image and a user specified representation task to determine the type of colormap which would most effectively present the data [3].

In visualization of isocontours, a simple and common approach is to select a set of isovalues  $w_0$  which are evenly spaced throughout the range  $w$  of the function. It is clear that such a technique is prone to miss features which may be considered important. The ability to select  $w_0$  and view isocontours in real-time allows the user to browse the scalar field for interesting features, however in the absence of additional guidance, the user may only use their prior knowledge of what they expect to see in the data, and query for isocontours in a trial-and-error loop. In the following sections, we introduce a user interface which presents the user with quantitative information defined over the entire range of isovalues. The interface serves the dual purpose of aiding the interactive selection of isovalues and providing real-time feedback of quantitative information about the selected isocontour.

## 3 The Contour Spectrum

The *contour spectrum* consists of computed metrics over the scalar field. On the basis of such metrics we define a set of functions which provide a useful tool to enhance the interactive query of the dataset. One primary advantage of the the contour spectrum interface is that it allows one to display in a 2D image a “global” view of the examined scalar field, independent of its dimension. For example, in the display of a 3D isosurface, one contour component maybe be hidden inside another. If we associate the isocontour display with the contour tree (details follow) it becomes immediately clear that the current isosurface is composed of two components and hence we might need a clipping plane to look inside the current isosurface.

Below we report on several examples of contour measures of general utility. Additional measures may be easily defined to enhance the approach both in general and for application dependent contour features.

### 3.1 Isoline Length and Isosurface Area

In this section we introduce the methodology used for efficient exact quantitative queries over the scalar field. In particular we determine a simple B-spline-based algorithm which allows the exact length (area) computation of an isocontour. The spline approach makes the computed data suitable for direct display in the contour spectrum.

Given a 2D (3D) scalar field we determine the exact length (area) value of any isocontour of height (scalar value)  $w$ . For unstructured triangular meshes these signature univariate functions are B-splines. Spline functions are easily displayed in the contour spectrum without introducing additional approximations (with respect to the given sampled data), and at the same time is used to perform interactive quantitative queries. The method generalizes to meshes of higher dimensions, providing a means for analyzing (with the spectrum) and interactively perform quantitative queries on datasets of any dimension, independently from the ability to display them.

#### 3.1.1 2D Contour Length

The two-dimensional scalar field case is particularly simple and is treated in detail to introduce the general methodology which becomes increasingly useful for field dimensions three or higher.

Consider a 2D scalar field composed of triangles  $t_i$  and vertices  $v_i$  such as the terrain in Figure 1. We build (and display) the spline function  $L(w)$  whose value  $L(w_0)$  is the length of the isocontour of height  $w_0$ .  $L(w)$  can be computed as the sum of all the contributions  $L_i(w)$  given by each cell  $c_i$  to the length of the contours:

$$L(w) = \sum_i L_i(w)$$

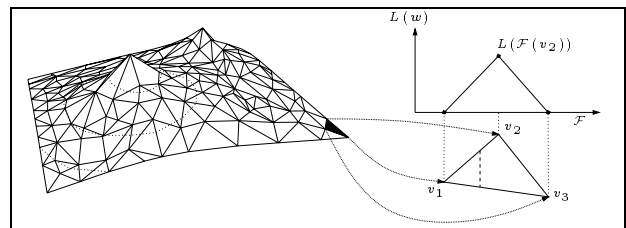


Figure 1: (left) A 2D scalar field displayed as a terrain. (right) The portion of an isocontour contained in a single triangle.

Thus, we can concentrate on the computation of the generic term  $L_i(w)$  associated with the triangle  $t_i$ , as il-

illustrated in Figure 1. Triangle  $t_i$  has vertices  $v_1, v_2$  and  $v_3$  with height values  $\mathcal{F}(v_1) \leq \mathcal{F}(v_2) \leq \mathcal{F}(v_3)$ . Given the equation  $f(x, y, w) = 0$  of the plane containing  $t_i$ , the value  $L_i(w_0)$  is the length of the intersection between  $t'_i$  (projection of  $t_i$  onto the mesh space) and the 2D line of equation  $f(x, y, w_0) = 0$  (see figure 1). As we change the value of  $w_0$  we obtain the measure of all the slices parallel to the line  $f(x, y, 0) = 0$ . In general it is known from spline theory that given a  $d$ -simplex in  $\mathbb{R}^d$  the function that gives the measure of all the parallel slices of such simplex (that is the measure of the intersection with a set of parallel hyperplanes) is a degree  $d-1$ ,  $C^{d-2}$  continuous, polynomial B-spline function [5]. Scalar fields of arbitrary topology meshes can of course be handled by first generating simplicial approximation of them. Note that in constructing a simplicial approximation for structured and convex cell topology meshes, no new data values are needed for the scalar field.

In the 2D case the B-spline is simply a piecewise linear  $C^0$  function. Hence we need only compute the length of the segment for  $w = \mathcal{F}(v_2)$  and connect it with the other two extremes for which the length is 0.

Note that the B-spline formulation of the length is also useful to automatically handle the eventual degenerate cases. For example a portion of the terrain at height  $w$  can be a flat parallel to the  $x, y$  plane (a lake). In this case there occurs a definition problem, in determining the length of an isocontour which is partially a 1-dimensional curve and partially a 2D surface. The natural solution is to remove the flat region to regularize the dimension of the contour. The consequence is that the function that computes the contour length is only  $C^{-1}$  at the height  $w$ . Using the B-spline approach no special care must be taken for this case since the knot vectors of the flat triangles are  $\mathcal{F}(v_1) = \mathcal{F}(v_2) = \mathcal{F}(v_3)$  resulting in “valid” splines which shrink to a point (as they should be).

### 3.1.2 3D Contour Area

As already pointed out, the above spline function can be computed for simplices of any dimension. For the 3D case of a tetrahedron  $(v_1, v_2, v_3, v_4)$  with scalar function values  $(\mathcal{F}(v_1) \leq \mathcal{F}(v_2) \leq \mathcal{F}(v_3) \leq \mathcal{F}(v_4))$  we have a degree two,  $C^1$  polynomial B-spline (see Figure 2). In this case the determination of the control polygon is as follows:

- First the area  $L(v_2)$  of the section of height  $\mathcal{F}(v_2)$  is computed.
- A straight line from the point  $(\frac{\mathcal{F}(v_1)+\mathcal{F}(v_2)}{2}, 0)$  passes through the point  $(\mathcal{F}(v_2), L(v_2))$  and continues up to the point  $P$  of abscissa  $\frac{\mathcal{F}(v_2)+\mathcal{F}(v_3)}{2}$ .

The point  $P$  is then connected with the point  $(\frac{\mathcal{F}(v_3)+\mathcal{F}(v_4)}{2}, 0)$

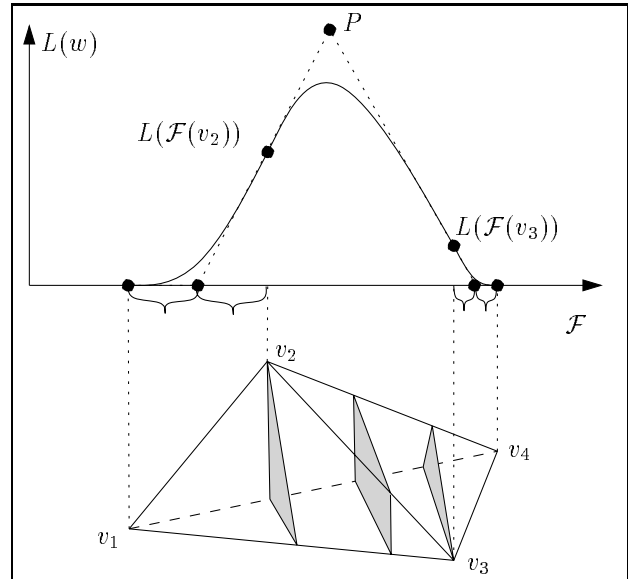


Figure 2: Area computation for the continuous range of isocontours contained in a single tetrahedron.

Again for each cell we obtain a spline function, as illustrated in Figure 2. The sum of the splines associated to each cell is a single spline that gives the contour area for any isovalue.

## 3.2 Inside Area/Volume Computation

Once the length/area function of the isocontours is given the Area/Volume of the region “below” (“above”) the isocontour can be determined by exact integration of the length/area polynomial B-spline function. This gives as a result a new B-spline function in which degree and continuity are increased by one. In this way we can easily plot the area/volume spectrum. The case for the 2D contour is illustrated in Figure 3.

## 3.3 Gradient Integral

While length and area are important metrics to report, in many cases they are not sufficient to guide the user in choosing appropriate isovalues. In many situations the user is interested in finding and displaying prominent surfaces in the data. Toward this end we have designed a metric which is based on the *slope* or *gradient* of the function. The difficulty with the gradient measure is to define it properly, since along a particular contour the gradient of the scalar field is not (usu-

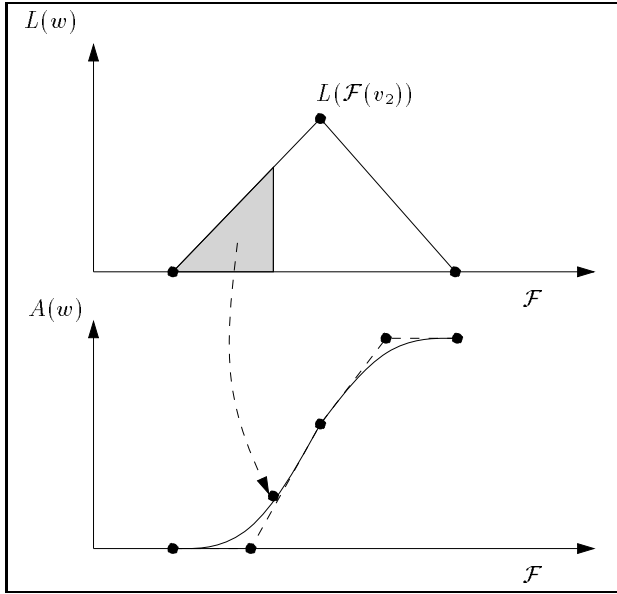


Figure 3: 2D area computation by integration of the length function  $L(w)$ . The shaded region corresponds to the area *less than* or *below* the isovalue. The area *above* the isovalue is computed symmetrically.

ally) constant. To compute a consistent (single valued) gradient function we resort to the spline decomposition of the contour length/area function. For each triangle/tetrahedron of the mesh we have a spline function which gives the length of any contour within that triangle/tetrahedron. Moreover, by piecewise linear approximation, within each triangle/tetrahedron the gradient of the scalar field is constant. Hence to determine the contribution to the gradient function of the contours within a single triangle we just need to multiply the length function by the absolute value of the (constant) gradient. Again the sum of the splines defined in each triangle/tetrahedron gives a single global spline function which defines the gradient integral of any isocontour in the scalar field. Figure 4 shows an MRI scan of a human heart. The maximum of the gradient (marked function plot on bottom figure) corresponds to the isocontour (isocontour on top figure) bounding the relevant portion of the data. Note how the maximum of the contour surface (red function plot) is attained for a lower height value of the field. It captures the noisy part of the data that has a large contour length due to the numerous components.

### 3.4 Real Time Quantitative Queries

The 2D plot of each of the above metrics provides a qualitative understanding of their trend. Once an isocontour is selected the user is usually interested in the

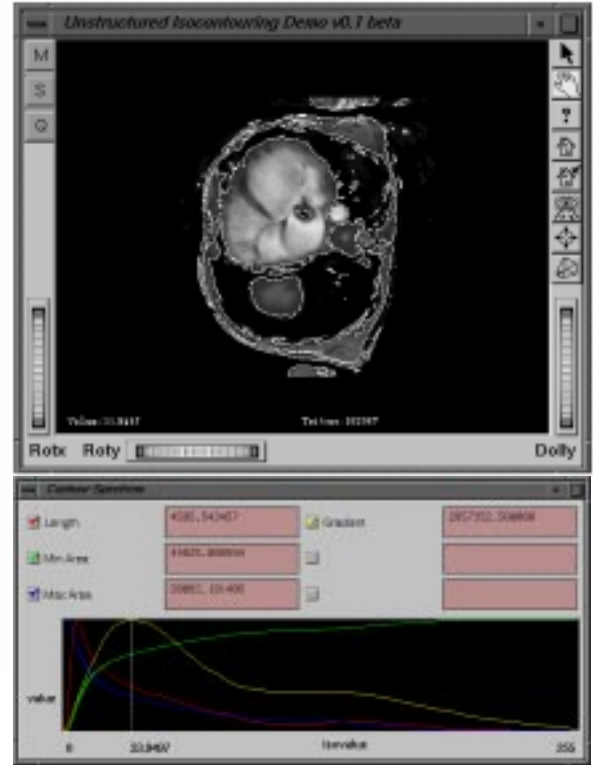


Figure 4: Top: MRI cross section of a human torso displaying the heart. Bottom: the corresponding contour spectrum with isovalue selected at the maximum of the gradient integral.

exact value of each of such metrics. This can be accomplished using the same B-spline representation. Since the B-spline defined above are exact representations of the relative metrics for the given piecewise scalar field we only need to search in the knot vector for the interval in which the selected isovalue lies and evaluate the related portion of spline. This takes, in the worst case,  $O(\log n)$  (the evaluation can be considered  $O(1)$ ) time where  $n$  is the number of different scalar values at the mesh vertices. Note that for the MRI data we have  $n = 256$ . In the general case, if  $n$  is too large we can apply any error bounded reduction scheme to keep  $n$  within an acceptable value. In such cases we will not get exact but error bounded results.

### 3.5 Contour Tree

While the display of contour metrics is both helpful and informative, there is clearly a lack of global structural information in the metrics described. For example, there is no indication of features such as local maxima and minima of the field. For this purpose we introduce the use of the *contour tree* as a tool for assisting the user in interaction with complex scalar fields. A con-

tour tree captures the global changes in contour topology of the scalar field defined on the input the mesh. It has been used before in image processing and GIS research [6, 7, 10, 12, 13]. Another name in use is the *topographic change tree*, and it is related to the *Reeb graph* used in Morse Theory [13]. Note the difference from the topology graph [2], which remains embedded in the mesh space and hence for 3D meshes is not displayed as a 2D graph.

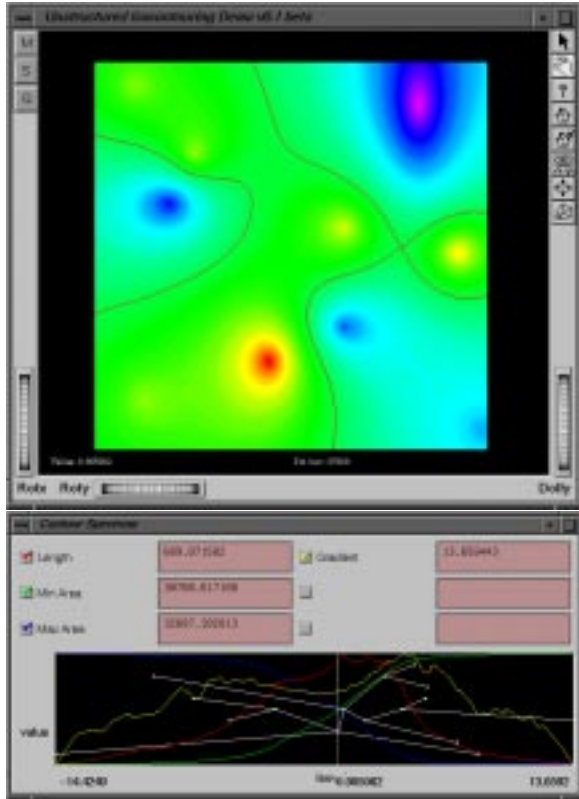


Figure 5: A 2D scalar field (top) with the associated contour spectrum with superimposed *contour tree* (bottom).

Figure 5 shows a 2D scalar field along with its associated contour tree. For each edge in the contour tree there is a connected component of an isocontour in the scalar field. If, while varying the isovalue, two contour components merge together we have in the contour tree two edges that join. Similarly, if an isocontour splits in two or more components we will have in the contour tree an edge that splits in two or more edges. Moreover the comparison between the contour tree and the spectrum may aid in the selection of interesting contours. Typically an isovalue that has a contour tree with many edges but a relatively small overall contour length/area corresponds to a noisy region. Symmetrically a single component of large length/area correspond to a well defined featured of the scalar field. Computation of the contour tree is

discussed in greater detail in [14].

## 4 User Interface

The user interface for presenting the contour spectrum takes on two forms. For static data, a window presents a selected subset of the plots of the computed signature functions. The horizontal axis represents the isovalue dimension. The vertical axis represents the range of each function, all of which are normalized for overlapping display. See Figure 6 for an example. The user may select a subrange of the isovalues for display in order to enhance the local detail in the computed metrics. Vertical bars represent the current isovalues, which the user may change with a familiar click-and-drag operation.

With time-varying data, it is desirable that the user have the ability to quickly browse all parameters of the visualization. In this case we use the vertical dimension of the interface as an index into the timestep of the data. Of course, while we use *time* here as an example, other parameters may be varied similarly, such as input parameters to a numerical simulation. Using this interface, each point in the 2D display maps to a number of functions. We selectively display one function at a time by pseudocoloring of the function values over the 2D grid, as shown in Figure 7.

## 5 Rule-based Contouring

An interesting and promising pursuit is to develop techniques which strategically choose a set of key isovalues which convey the data most clearly.

An important caveat to rule-based contouring is that users familiar with a particular isovalue selection mechanism, such as the selection of  $n$  evenly spaced isovalues, may easily misinterpret the display of a number of contours which are irregularly scattered throughout the range of the function.

The contour spectrum allows the development of an adaptive ability to capture the “interesting” features of a dataset. Figure 8 shows the scalar field obtained as CT scan of an engine. The main component of the engine can be easily determined by selecting the maximum of the gradient integral. Of course this remains simply an aid in the interactive querying stage of the dataset, as the concept of “interesting” feature of a scalar field remains highly dependent on the type of dataset we are dealing with.

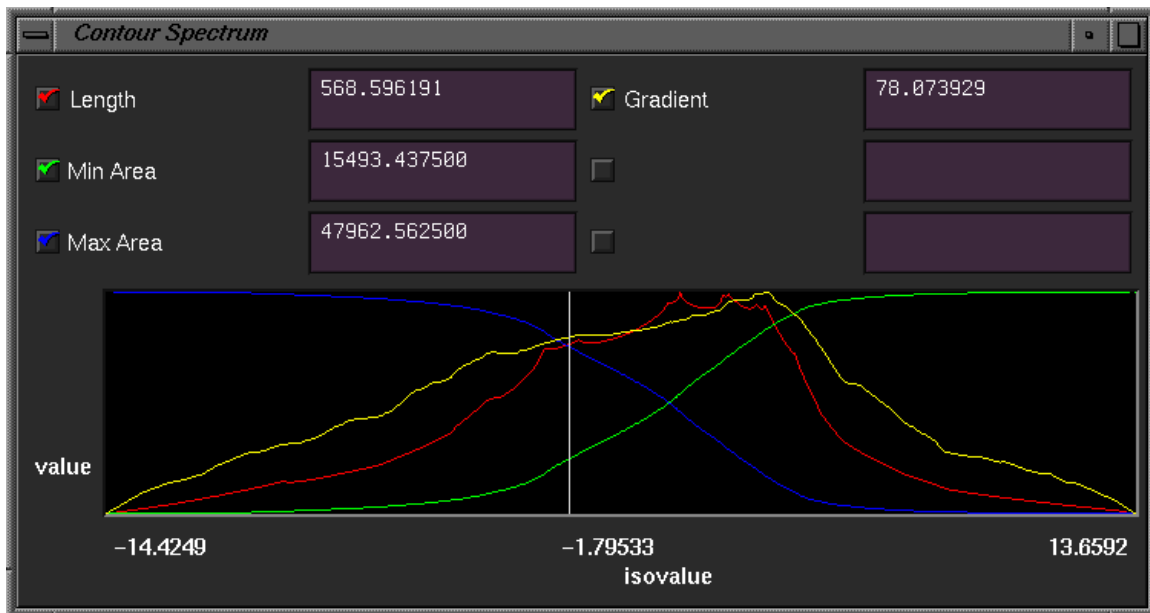


Figure 6: Contour Spectrum Interface showing univariate Quantitative Signature functions.

## 6 Conclusions

In addition to increasing user interaction, quantitative interfaces for visualization are a first step to developing the ability to automatically select visualization parameters for effective visualizations. While certain general isovalue selection techniques are discussed here, we propose that application specific rules for isovalue selection based on metric properties be developed. In particular we are exploring the use of 2D vector field displays of vector signature functions as well as automatic generation of significant parameter values of the underlying dataset.

The measure of visualization effectiveness is the amount of insight gained by the user. For automated visualization and parameter selection to become viable and effective, it will be necessary for visualization users to understand the implications of the parameter selection techniques which have been applied.

## Acknowledgments

This research was supported in part by AFOSR grant F49620-94-10080 and ARO grant DAAH04-95-1-0008.

## References

- [1] Chandrajit L. Bajaj, Valerio Pascucci, and Daniel R. Schikore. Fast isocontouring for improved interactivity. In *Proceedings of 1996 Symposium on Volume Visualization*, pages 39–46, October 1996.
- [2] Chandrajit L. Bajaj and Daniel R. Schikore. Visualization of scalar topology for structural enhancement. Technical Report CSD-TR-96-006, Department of Computer Sciences, Purdue University, West Lafayette, IN, 1996.
- [3] L. Bergman, B. Rogowitz, and L. Treinish. A rule-based tool for assisting colormap selection. In G. M. Nielson and D. Silver, editors, *Visualization '95 Proceedings*, pages 118–125, October 1995.
- [4] Paolo Cignoni, Claudio Montani, Enrico Puppo, and Roberto Scopigno. Speeding up isosurface extraction using interval trees. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):158–170, 1997.
- [5] Carl de Boor. *A Practical Guide to Splines*. Springer-Verlag, 1978.
- [6] H. Freeman and S.P. Morse. On searching a contour map for a given terrain profile. *Journal of the Franklin Institute*, 248:1–25, 1967.
- [7] C. Gold and S. Cormack. Spatially ordered networks and topographic reconstructions. In *Proc. 2nd Int. Sympos. Spatial Data Handling*, pages 74–85, 1986.
- [8] Taosong He, Lichan Hong, Arie Kaufman, and Hanspeter Pfister. Generation of transfer functions with stochastic search techniques. In *Visualization '96 Proceedings*, pages 227–234, October 1996.

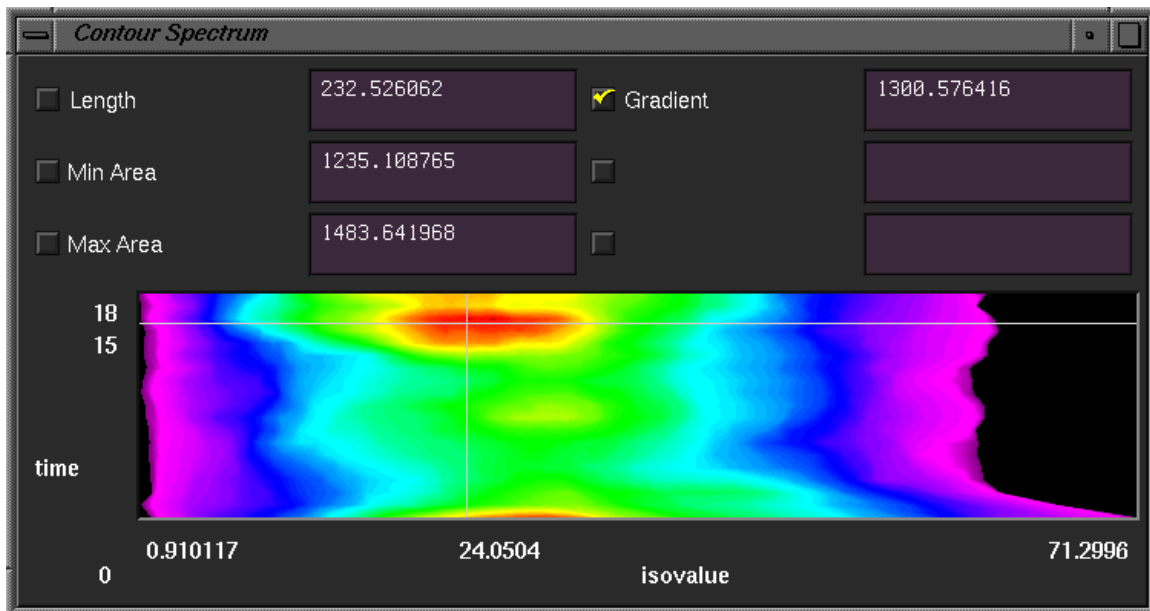


Figure 7: Contour Spectrum Interface showing Bivariate Signature functions via color maps.

- [9] Takayuki Itoh, Yasushi Yamaguchi, and Koji Koyamada. Volume thinning for automatic isosurface propagation. In *Visualization '96 Proceedings*, pages 303–310, October 1996.
- [10] I.S. Kweon and T. Kanade. Extracting topographic terrain features from elevation maps. *CVGIP: Image Understanding*, 59:171–182, 1994.
- [11] Y. Livnat, H. W. Shen, and C. R. Johnson. A near optimal isosurface extraction algorithm for unstructured grids. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):73–84, 1996.
- [12] J.K. Sircar and J.A. Cerbrian. Application of image processing techniques to the automated labelling of raster digitized contours. In *Proc. 2nd Int. Symp. on Spatial Data Handling*, pages 171–184, 1986.
- [13] S. Takahashi, T. Ikeda, Y. Shinagawa, T.L. Kunii, and M. Ueda. Algorithms for extracting correct critical points and constructing topological graphs from discrete geographical elevation data. 14:C-181–C-192, 1995.
- [14] Marc van Kreveld, Rene van Oostrum, Chandrajit L. Bajaj, Valerio Pascucci, and Daniel R. Schikore. Contour trees and small seed sets for isosurface traversal. In *13th ACM Symposium on Computational Geometry*, pages 212–220. ACM, 1997.

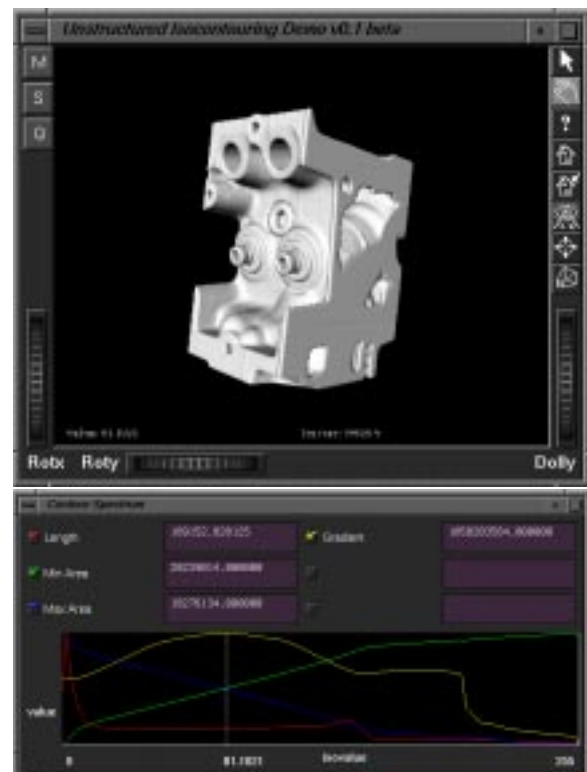


Figure 8: Isocontour of a CT scan of an engine (top) automatically selected at the isovalue corresponding to the maximum of the weighted gradient spectrum (bottom).