Eurographics Conference on Visualization (EuroVis) 2015
H. Carr, K.-L. Ma, and G. Santucci
(Guest Editors)

*Volume 34 (2015), Number 3*

# Fiber Surfaces: Generalizing Isosurfaces to Bivariate Data

Hamish Carr[1], Zhao Geng[1], Julien Tierny[2,3], Amit Chattopadhyay[1] and Aaron Knoll[4]

[1]University of Leeds, UK
[2] Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France
[3] CNRS, UMR 7606, LIP6, F-75005, Paris, France
[4]University of Utah, USA

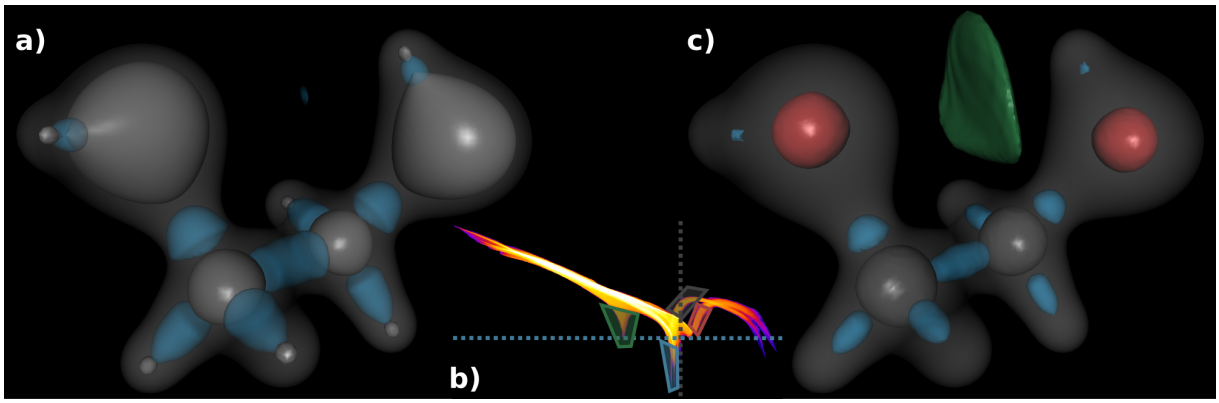


**Figure 1:** *Fiber Surfaces of electron density and reduced gradient in an ethane-diol molecule: (a) While an isosurface of electron density identifies regions of influence of atoms (grey), it does not distinguish atomic type. An isosurface of reduced gradient identifies bonding interaction sites (blue) but does not distinguish non-covalent (top) from covalent bonds (others). (b) Continuous scatter plot (log scale) of electron density and reduced gradient. Isosurfaces and fiber surfaces are shown as dashed lines and polygons respectively. (c) Fiber surfaces distinguish atom types (oxygen in red, carbons in grey) as well as bond types (non-covalent in green, covalent in blue).*

**Abstract**

*Scientific visualization has many effective methods for examining and exploring scalar and vector fields, but rather fewer for bivariate fields. We report the first general purpose approach for the interactive extraction of geometric separating surfaces in bivariate fields. This method is based on* fiber surfaces*: surfaces constructed from sets of* fibers*, the multivariate analogues of isolines. We show simple methods for fiber surface definition and extraction. In particular, we show a simple and efficient fiber surface extraction algorithm based on Marching Cubes. We also show how to construct fiber surfaces interactively with geometric primitives in the range of the function. We then extend this to build user interfaces that generate parameterized families of fiber surfaces with respect to arbitrary polygons. In the special case of isovalue-gradient plots, fiber surfaces capture features geometrically for quantitative analysis that have previously only been analysed visually and qualitatively using multi-dimensional transfer functions in volume rendering. We also demonstrate fiber surface extraction on a variety of bivariate data.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modelling—Curve, surface, solid and object representations

## 1. Introduction

As scientific visualization expanded, methods developed for visualizing scalar and vector fields, but fewer methods for bivariate fields. In particular, methods to extract surfaces representing boundaries are largely unexplored. We introduce a novel method for visualizing bivariate fields based on the generalization of isosurfaces to *fiber surfaces*: surfaces made up from fibers (the equivalent of contours in multi-fields).

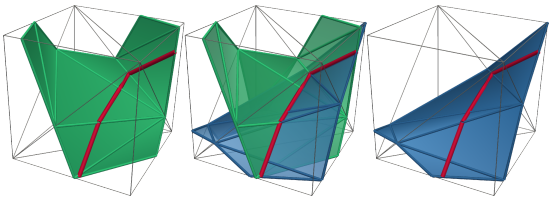

**Figure 2:** *Fiber Construction. Left: isosurface of $f_1$. Centre: fiber defined by intersecting isosurfaces. Right: isosurface of $f_2$. Both isosurfaces show the fiber for reference.*

Our contribution is therefore to show:

1. a general purpose method that produces separating surfaces representing boundaries in bivariate fields,
2. an efficient and simple fiber surface extraction method based on Marching Cubes for any mesh type,
3. simple yet powerful interfaces based on lines and polygons in the range of the field,
4. families of fiber surfaces parametrized by a single variable with respect to polygons
5. a relationship between these families of fiber surfaces and linear combinations of two scalar fields,
6. that fiber surfaces can extract geometric surfaces for features in multi-dimensional transfer functions [KKH02]

Crucially, fiber surfaces are *geometric*: no topological computation is required. However, topological analysis gives further insight and further power, and we will also discuss the relationship with multifield topological analysis [CD14].

## 2. Background

Relevant work includes the use of Marching Cubes to extract isosurfaces and separating surfaces (Section 2.1), methods for multifield visualization (Section 2.2), multi-dimensional transfer functions for direct volume rendering (Section 2.3), and generalizing contours to fibers in multifields (Section 3).

### 2.1. Isosurfaces and Marching Cubes

Given a scalar field $f : \mathbb{R}^3 \to \mathbb{R}$, contours and isosurfaces can be defined mathematically as the inverse image $f^{-1}(h) = \{x \in Domf : f(x) = h\}$ of an *isovalue* $h \in Ranf$. For a simply connected domain, this has the useful property that it separates the domain into pieces: in particular, for many datasets, the isosurface is a closed surface which represents some sort of boundary in the phenomenon under study. Computationally, isosurfaces are approximated with triangles using *Marching Cubes* [LC87]. In this algorithm, the space is subdivided into a grid of cubes with known data values at the grid intersections. For a given isovalue $h$, the algorithm then extracts a surface in each cube in four stages:

I: **Classification:** The data value $f(x)$ at each corner of the cube is compared with the isovalue $h$. If $f(x) > h$, the vertex is classified as "black" (a 1 bit). Otherwise it is classified as "white" (a 0 bit).

II: **Triangle Topology:** The eight bits are converted into a single-byte integer called the 'case' and used to retrieve triangle topology from a look-up table, with triangle vertices located along edges of the cube.

III: **Vertex Interpolation:** For each triangle vertex, linear interpolation based on the isovalue is applied along the vertex' edge to determine the exact location. Without this stage, the vertices are fixed to grid locations and result in blocky surfaces which are visually displeasing.

IV: **Normal Vectors:** Normal vectors are constructed either as flat normals of the faces, by averaging normals around each vertex, or using central differencing and interpolation to estimate the gradient vector at the vertex.

While Marching Cubes is not perfect, it is the principal method for isosurface extraction due to its simplicity, robustness and ability to represent material boundaries as separating surfaces [NY06, Wen13]. Variants exist in particular for tetrahedra [Blo88], where the surfaces extracted are mathematically correct for the linear interpolant.

### 2.2. Multifield Visualization

Other than reduction to scalar fields or direct volume rendering, few general methods for bivariate visualization in $Domf$ are known, except for the special case of complex-valued fields [WB96], where a complex value was chosen in the range of $f : \mathbb{C}^2 \to \mathbb{C}$, and the corresponding 2-manifold contour in $\mathbb{C}^2$ was constructed. If we treat $\mathbb{C}$ as $\mathbb{R}^2$, $f$ can be restated as $f : \mathbb{R}^4 \to \mathbb{R}^2$, and these complex contours are then fibers of $f$, as described in the next section.

One method that is often used is to classify the data points statistically as "interior" or "exterior" then apply stage II. of Marching Cubes. However, this binary classification makes it difficult to apply stages III. and IV, which are usually resolved with heuristics [HM03, SBSG06].

Multifields can be shown as multidimensional histograms, and recent work on continuous scatterplots [BW08] has shown the importance of the presumed mesh continuity. Subsequent work has focussed on linear features [LT10] which are now [CD14] known to be related to the topology of the multifield. Further work on multifield topology is ongoing [EH04, EHP08, NN11], but these methods are complex and not fully developed.

### 2.3. Direct Volume Rendering

Direct volume rendering visualizes data [Lev88] with a transfer function that maps data values to color and opacity, then integrates light transport along rays for each pixel. Laidlaw [Lai95] first proposed multidimensional transfer functions for MRI data. 2D transfer functions were also proposed to augment univariate data with gradient magnitude [KKH02, Kin02] or curvature [KWTM03]. Early multifield approaches employed Gaussian kernels [KPI*03] or maximum-intensity projection [SR04]. Bivariate classification can also use derived quantities such as heat diffusion [GWK12], or apply clustering of cells [WZL*12] or Morse-Smale analysis [KKH13].

Recent work has built user interfaces for dimensionality reduction and transfer function generation, added scatterplots [Dol07], kernel density estimation [MWCE09, LWT*14], parallel coordinates [ZK10, GXY11], combinations of all three [ZH13], or automatic approaches [GMY11, ZH14]. We refer interested readers to a recent survey [KH13]. For multifield geometric features. Kotava et al. [KKS*12] show that the necessary sampling rate for sharp feature reconstruction depends on the product of frequencies of all component fields convolved with the transfer function, and sample directly in transfer function space to render high-frequency features similar to fiber surfaces.

In these papers, linear range features correspond roughly to material boundaries, and interfaces are thus often designed using polygonal widgets in $Ranf$ [SKK06]. For example, Kotava et al. [KKS*12] volume render fine features in multidimensional transfer functions by sampling in range space. While direct volume rendering is now standard for visualizing scalar fields and multifields, it has two drawbacks. First, even with modern GPUs, rendering is relatively slow, especially for complex high-gradient data. Second, it produces an image, not a geometric surface that can also be used for modelling and simulation.

Geometric surfaces are thus of general interest, with particular value where multi-dimensional transfer functions have already been adopted. We therefore turn our attention to generalizing isosurfaces to the bivariate case, observing that the development is similar for higher dimensions.

## 3. Fibers

For $f : \mathbb{R}^d \to \mathbb{R}^r$, the domain $Domf = \mathbb{R}^d$ is the set of input values, with $d$ commonly $2, 3$, or 4. In contrast, the range $Ranf = \mathbb{R}^r$ is the set of output values: in scientific visualization, this may be scalar ($r = 1$), bivariate ($r = 2$), vector ($r = d$, with special semantic meaning), or higher dimensions. This paper only addresses bivariate data, so we assume $d = 3, r = 2$, and use $f_1, f_2$ to refer to the two output variables. We also assume a simply connected domain. We leave for future work the questions of extra dimensions and more general manifolds.

For bivariate (and higher) data, inverse images of points (*fibers*) are well-defined [Sae04], and are analogous to contours. In $\mathbb{R}^3 \to \mathbb{R}^2$, a fiber is defined by a point $h = (h_1, h_2) \in Ranf$, and can be found by intersecting isosurfaces of $h_1$ in $f_1$ and $h_2$ in $f_2$, as shown in Figure 2.

For a contour of co-dimension 1 (one dimension less than the domain), this fiber is of co-dimension 2, i.e. a 1-manifold structure similar to a contour line. This happens because our point $h = (h_1, h_2)$ specifies two parameters, each of which reduces the dimensionality by one. Thus, a single fiber cannot separate regions, as shown in Figure 2.

## 4. Fiber Surfaces

The challenge is to find a generalization of contours to bivariate volumes that produces well-defined separating surfaces. We do so by constructing surfaces from fibers, and for this a further property of fibers (and contours) is useful: as $h$ varies continuously, so does $f^{-1}(h)$. Thus, any path in $Ranf$ corresponds to a set of fibers that varies continuously in the domain, sweeping out a surface or set of surfaces.

There has been historic confusion between isosurfaces and their connected components (often called contours or isosurface components), since connectivity was not considered when isosurfaces were originally defined. Similarly, a fiber may have multiple components, and the term *fiber component* has recently been introduced to refer to a single connected component of a fiber [SSC*]. For consistency, we therefore use *fiber surface* for the inverse image in $Domf$ of any path $P \in Ranf$, and *fiber surface component* for a single connected component of a fiber surface.

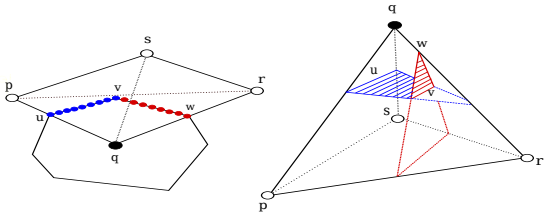


**Figure 3:** *Fiber Surface of a Polygon. Left: the polygon in the range with the projection of a single tetrahedra. Right: the tetrahedron in the range with the fibers corresponding to edges uv, vw. Each fiber is a point in the range but a line in the domain. Each line in the range corresponds to a plane in the tetrahedron.*

Fiber surfaces have two properties similar to isosurfaces. First, each fiber surface component is a continuous surface. Second, if the path $P$ separates $Ranf$ into regions, then the corresponding fiber surface $f^{-1}(P)$ is a separating surface in $Domf$. To see that this is true, consider two points $p, q \in Ranf$ that are separated by $P$. Claim: all paths $Q \in Domf$ between $f^{-1}(p)$ and $f^{-1}(q)$ intersect $f^{-1}(P)$. To see this, we observe that each point $r \in Q$ belongs to some

fiber $f^{-1}(h_r) : h_r \in Ranf$. Moreover, since $f$ is a continuous mapping, the set $h_r : r \in Q$ must be continuous in $Ranf$, and therefore forms a path $f(Q)$ in $Ranf$, although some $h_r$ may be repeated. But $P$ separates $p$ and $q$, so any such path $f(Q)$ must cross $P$: it then follows that $Q$ must cross $f^{-1}(P)$, as required. Moreover, if $Q$ does not touch the boundary of $Domf$, then it must be a closed 2-manifold surface.

## 5. Fiber Surfaces of Lines in the Range

The simplest fiber surfaces are given by lines in the range in normal form: $\vec{n} \cdot p = c$, where $\vec{n} = (n_1, n_2)$. Here, $c$ is the perpendicular distance from the origin to the line, scaled by $\|\vec{n}\|$. For any point $p = (h_1, h_2)$ on this line, $n_1 h_1 + n_2 h_2 = c$. Now, given any point $x$ on the fiber $f^{-1}(p)$ in the domain, $(f_1(x), f_2(x)) = p = (h_1, h_2)$, and $n_1 f_1(x) + n_2 f_2(x) = c$ for all such points. Thus, this fiber surface is the isosurface at $c$ of the weighted sum $\vec{n} \cdot f$, i.e. an isosurface of a derived scalar field. Doing this at runtime is easy: we pass $\vec{n}$ and $c$ to Marching Cubes, and compute derived values at vertices on the fly. Trivially, an isosurface of one variable is a fiber surface defined by a vertical or horizontal line.

If $f$ is defined on a tetrahedron, the fiber surface of a line is a plane (Figure 3). For a cubic mesh with trilinear interpolation, Nielson's cases [Nie03] can be used. We also observe that for a derived field $g = \vec{n} \cdot f$, $\nabla g = n_1 \nabla f_1 + n_2 \nabla f_2$ - i.e. we can compute normal vectors either with the gradient of the derived field $g$, or by linear combination of the gradient vectors of the components of $f$. Alternately, as with conventional isosurfaces, we can either use flat-shading of each triangle, or average normal vectors around each vertex.

All isosurfaces of $\vec{n} \cdot f$ share the normal vector $\vec{n}$ with different isovalues $c$. Therefore their range lines are parallel, with $c$ the distance from the origin to the line. We define the line through the origin to be the reference, and observe that the others are at fixed distances from it: i.e. parallel lines are contours of the distance field of the reference line. We will use this to generalize beyond lines to arbitrary curves.

## 6. Fiber Surfaces of Polygons

We now consider the general case of fiber surface defined by arbitrary curves, polylines or polygons. Since curves and polylines can be approximated with polygons, we assume a separating polygon: i.e. a closed loop of range line segments, which we call the *fiber surface control polygon* or *FSCP*.

Again, we start from the observations that each point on the FSCP corresponds to a fiber, and that $f$ is a continuous mapping. As we travel around the FSCP, the continuity of $f$ implies that while a fiber may separate into components or join, the fibers themselves deform continuously into each other, thus sweeping out a set of continuous surfaces in the domain. In the ideal case, we would extract the fiber surface exactly, but in practice this is more difficult than it sounds. Consider Figure 3, in which we show a single tetrahedron with the fiber surface defined near a vertex $v$ of the FSCP. Each linear segment in the range is a subset of a range line fiber surface, each of which is planar in the tetrahedron, and the planes meet along the fiber for $v$. Thus, having anything other than a straight line gives potentially arbitrarily complex geometry in each tetrahedron in the mesh.

An important property of isosurfaces is that they separate an inside from an outside. This property also carries over to fiber surfaces. Consider a path $P_d$ from the inside of a closed fiber surface to its outside. Each point on this path belongs to a fiber, and this fiber is defined by a point in the range. Thus, the projection of $P_d$ is also a continuous path $P_r$, but in the range. Moreover, there is some point $p_d$ where $P_d$ crosses the fiber surface, and it's fiber is defined by point $p_r = f(p_d)$ in the range. Removing the fiber of $p_r$ then disconnects the path $P_d$ in the domain, and removing the point $p_r$ disconnects the path $P_r$ in the range. It then follows that any set whose removal disconnects the range of $f$ must disconnect the domain of $f$, and the result follows.

Strictly speaking, it is possible for the inside of the FSCP to map to the outside of the fiber surface. In practice, as long as we are prepared to render both sides of the surface, this is unimportant. Algorithmically, mesh vertices whose fibers are inside the FSCP are also inside the fiber surface. Correspondingly, vertices whose fibers are outside the FSCP are also outside the fiber surface. But this is the same as stage I of Marching Cubes, substituting a point-in-polygon test for the isovalue comparison. Stage II is then performed as usual, using the standard lookup tables.

Stages III and IV are trickier, as we no longer have an isovalue for interpolation. Observe however that vertices of the cube have locations in the range: i.e. vertex $u$ of the cube will map to $f(u)$, and an edge $uv$ will intersect the fiber surface if and only if $f(u)$ and $f(v)$ are on opposite sides of the FSCP $P$. Thus, the line segment $f(u)f(v)$ intersects $P$ at some point $w \in Ranf$, and therefore the edge $uv$ must intersect the fiber surface. Moreover, if we parametrize $f(u)f(v)$ with a parameter $t$, we can compute the point $e = u + t(v-u)$ at which the fiber surface intersects the cube edge.

Finally, normals of these approximate fiber surfaces can use flat shading or averaged normals. Alternately, we can identify which edge of $P$ was intersected, and use its normal vector $\vec{n}$ to weight the gradient components of $f$ as before.

This leads to Algorithm 1. While this can be implemented as it stands, it depends on slow geometric tests, so we recall from Section 5 that fiber surfaces from parallel lines correspond to isosurfaces of the distance field contours of a reference line in the range. Generalizing this, the fiber surface for a given FSCP is defined by the zero-contour of the FSCP's signed distance field. Moreover, given linear interpolation, interpolating in the distance field will compute the same $w$ as before, without requiring either point in polygon tests or line intersection tests.

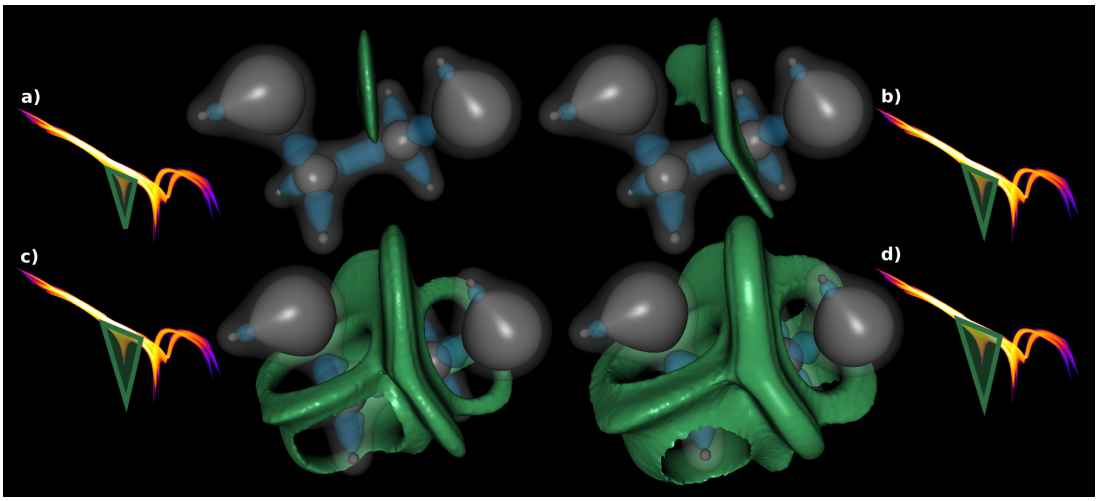


**Figure 4:** *Fiber surfaces obtained by increasing the range distance to the initial fiber surface control polygon, from (a) to (d).*

**Algorithm 1** Algorithm for extracting Fiber Surfaces

**Require:** Function $f$, Mesh M in $Dom f$, Polygon P in $Ran f$
  **for** each cell C in mesh M **do**
    **for** each vertex V in C **do**
      **if** f(V) is in polygon P **then**
        Classify V as black
      **else**
        Classify V as white
      **end if**
    **end for**
    Compute Marching Cubes (MC) case from vertex classification
    **for** Each triangle T in MC case **do**
      **for** Each cell edge $u, v$ intersected by MC case **do**
        Find intersection $w$ of line segment $f(u), f(v)$ and polygon $P$
        Find parameter $t$ on $f(u)f(v)$ for $w$
        Interpolate vertex $e = u + t(v - u)$
        Interpolate normal vector at $e$
      **end for**
    **end for**
  **end for**

Finally, if we compute the distance field for a FSCP, and choose contours at non-zero distances, we obtain a family of fiber surfaces that nest properly inside each other as isosurfaces do, and that can be parametrized by the distance value.

## 7. User Interfaces

Given the above definition of fiber surfaces, the next task is to build a user interface to define them. In the longer run, we expect user interfaces to be at least as rich as those for multi-dimensional transfer functions. Moreover, user interfaces for fiber surfaces will ultimately be application dependent, so evaluation of effectiveness must be deferred until domain-specific interfaces are constructed. We therefore restrict our attention to simple proof-of-concept interfaces.

Clearly, a fiber surface interface must show the range of the function as well as the fiber surfaces in the domain. Moreover, it is desirable to show some information in the range that helps the user understand *where* to place a line, curve or polyline. We therefore show the continuous scatterplot [BW08] of $f$ in the range, and superimpose lines and polygons on it. As the continuous scatterplot is fixed for a given dataset, we assume it has been precomputed.

The second major decision is whether to use lines or polygons to define fiber surfaces. Lines can be defined in several ways: as a pair of points that can be manipulated, as a dial for normal vector $\vec{n}$ and a slider for constant $c$, or as a single point $p$ that defines vector $\vec{n} = p - O$ from the origin constant $c = \vec{n} \cdot p$. We show some of these possibilities in the accompanying video, but observe that polygons are the more general case, and assume only polygons.

Thus, our interface consists primarily of two widgets: a range widget and a domain widget, as previously used by Sakurai et al. [SSC*]. In the range widget, we provide the ability to edit the FSCP, while in the domain widget, we show the corresponding fiber surface. In addition to this, we provide a number of utility widgets to control what is visualized at any point, and in particular, a slider for setting the distance field parameter for families of fiber surfaces.

## 8. Fiber Surface Examples

Given this user interface, we now show examples drawn from chemistry, cosmology and combustion.

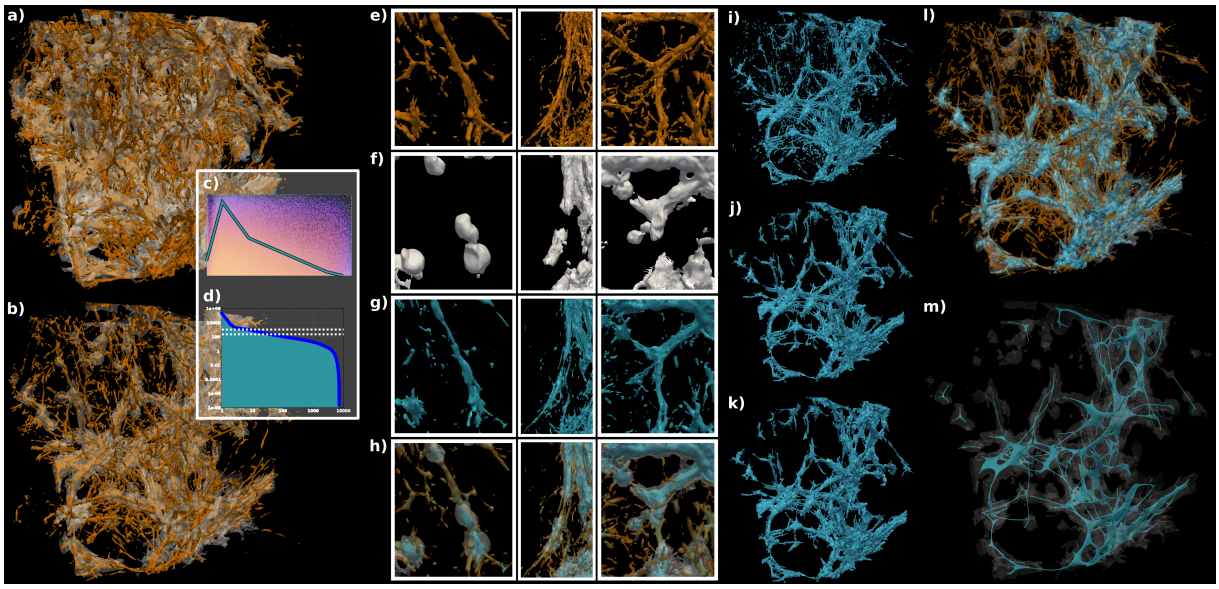


**Figure 5:** *Fiber Surfaces of a cosmological simulation of universe expansion: (a, b) Isosurfaces of matter (white) and dark matter (orange) concentrations for different pairs of isovalues (a: low, b: high). (c) Discrete scatter plot superimposed on the continuous one (matter vs. dark matter distribution). (d) Distribution by surface area of fiber surface components. (e)-(g) Zoom-in views for dark matter, matter and fiber surface respectively. (h) Composite zoom-in views for the 3 surfaces emphasizing their nesting relation. (i) Fiber surface for the polygon shown in (c). (j)-(k) Progressive simplification of the fiber surface components by surface area (dashed lines in (d)). (l) Composite view of dark matter (orange), matter (white) and fiber surface (blue). (m) Fiber surface contraction through smoothing passes to enhance the visual display of bubbles and filaments.*

**Chemistry:** In Figure 1, we extract fiber surfaces from bivariate data for (i) the electron density and (ii) the reduced gradient of an ethanediol molecule. Here, the electron density is the distribution function of electrons, with local maxima near atoms, while the reduced gradient [JKMS*10] is the deviation in atomic densities due to molecular interactions, and has local minima near atoms and covalent or non-covalent interactions. Together, these functions are used by chemists to isolate regions of influence of atoms and regions of molecular interactions. Tools such as NCI plots [CGJK*11], use sequences of isosurfaces and thresholds to isolate such features. Isosurfaces of electron density capture regions of atomic influence (grey, 1(a)), but cannot distinguish between atom types. Isosurfaces of reduced gradient capture molecular interactions (blue, 1(a)), but do not distinguish covalent from non-covalent bonds.

Electron density and reduced gradient are related exponentially in regions where no chemical interaction occurs, i.e. on the main separating axis of the continuous scatterplot (bright colors, 1(b)). Regions away from this axis are thus presumed to be significant in analysing chemical interactions. In our example, the regions of interest occur near the atoms (2 C, 2 O, 6 H), and near covalent (C-C, C-H, C-O) and non-covalent (O-H) atomic bonds.

These features can be selected visually in the continuous scatterplot (1(b)). Since different atom types have different reduced gradients, they have distinct polygons above the main axis (1(c), oxygen in red, carbon in grey). Similarly, covalent and non-covalent bonds differ in electron density, clearly distinguishable as the polygons below the main axis (1(d), covalent bonds in blue, non-covalent bonds in green).

While topology-based approaches [GABCG*14] semi-automatically segment these regions using the join tree of the reduced gradient and the Morse-Smale complex of a function derived from electron density, our approach is simpler. We leave automatic detection of chemical features in range space for the future, as it is likely to depend on topological analysis of fiber surfaces. Moreover, in contrast to semi-automatic methods, our approach allows interactive exploration of the features in range space, as seen in Figure 4. Here, the downward spike in the continuous scatter plot was selected to investigate non-covalent interactions (planar interface between the two alcohol (OH) groups). Finally, our user interface allows interactive exploration of the parameterized family of fiber surfaces induced by this polygon, by choosing different distance constants, as discussed in Section 6. The resulting surfaces, in a) - d), could be interpreted as candidate regions for non-covalent interactions.

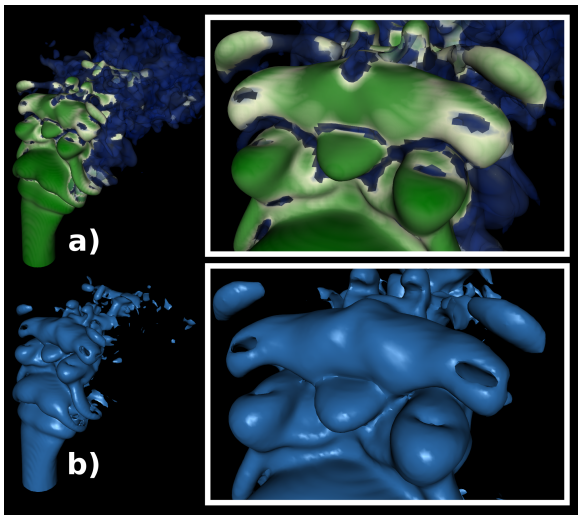


**Figure 6:** *Extracting the core of a burning flame: (a) Isotherm thresholded (opaque triangles) on temperature gradient magnitude (color map). (b) Extracting the same feature with fiber surfaces guarantees closed 2-manifolds.*

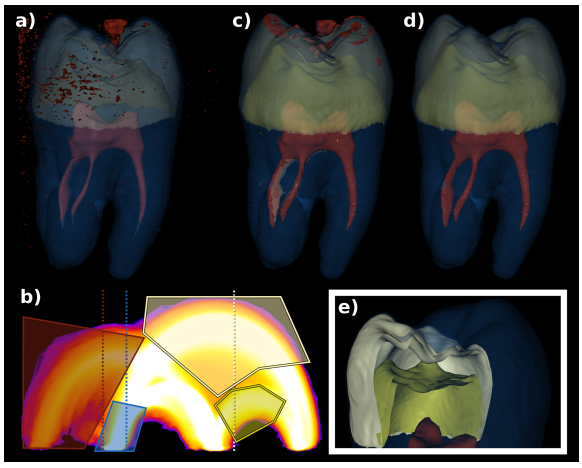


**Figure 7:** *Fiber surfaces for material boundaries in a tooth CT-scan: (a) User selected isosurfaces. (b) Continuous scatter plot of isovalue vs. gradient magnitude with user selected isovalues (dashed lines) and polygons. (c) Fiber surfaces of the selected polygons. (d) Fiber surfaces after connected component filtering. (e) Cut-away view of the fiber surfaces.*

**Cosmology:** Our second example (Figure 5) is from cosmology, with bivariate data representing a time-step in a simulation of universe expansion. Here, the first value represents concentrations of matter (white isosurfaces), while the second represents concentrations of dark matter (orange isosurfaces). Scientifically, the expectation is that high concentrations of matter will be co-located with high concentrations of dark matter, but not vice versa. Analysis thus focuses on "bubbles", where both properties are locally high, and "filaments" that connect the bubbles by thin threads with high concentrations of dark matter only. The thresholds are, however, unknown and are expected to vary across the data set.

For low isovalues (5(a)), the matter isosurface has one component containing most of the volume, while the dark matter isosurface has many small components. At higher isovalues (5(b)), dark matter isosurfaces exhibit less noise, while matter isosurfaces occupy less space, better showing the backbone structure of matter which links galaxies during expansion. However, as shown in zoom-in views, this backbone structure involves both matter and dark matter.

If we use a polygon to select the main feature visible in the scatter plot, (5(c)), the corresponding fiber surface captures the shape of the bubbles of matter isosurfaces (zoom-in views 5(e)-(g)) while still connecting these bubbles through dark matter isosurface components, hence better capturing the overall backbone structure of the galaxies. This is emphasized in Figures 5(h) and 5(l), which illustrate the nesting relation of these three surfaces: the fiber surface appears in light blue when it encompasses the dark matter isosurface, and in light green in the opposite configuration. We do not claim fiber surfaces are better than isosurfaces at comparing matter and dark matter. Existing visualizations of such data commonly employ side-by-side volume rendering with separate 1D classification [IWPN]. In discussions with cosmologists, we found that they frequently use joint histograms to analyze multifield data; fiber surfaces may prove helpful to domain experts in showing the combined geometry of both fields, in the context of multivariate scatterplots.

Moreover, since fiber surfaces are explicit geometry, we filter the individual connected components by their surface areas (5(d)). Doing so shows the predominance of a few very large components, allowing us to select a filter threshold that limits the initial self occlusion of the fiber surface (5(i)-(k)). Moreover, thanks to the explicit representation of fiber surfaces, the visualization of the backbone structure of the galaxies can be enhanced by contracting the fiber surface by a smoothing procedure with a large number of iterations (5(m)). In the process, features that look very like the expected bubbles and filaments start appearing (5(m)).

**Combustion:** In our third example, we look at two properties in a combustion simulation: temperature and temperature gradient magnitude (Figure 6). Here, features are found by extracting an isotherm, then using a threshold of the gradient magnitude to identify regions of rapid change, but this leads to a non-manifold surface with holes in it (a). In contrast, choosing a fiber surface guarantees closed 2-manifold surfaces that satisfy the same constraint (b), improving our ability to do geometric and numerical post-processing.

**Table 1:** *Runtime Statistics : All timings were measured in seconds and performed on a 3.06GHz Mac Pro (2012) with 64GB memory and a Radeon 5870 1GB video card, running OSX 10.8.5, and using VTK 6.1.0.*

| Data | Resolution | Polygon Edge Nr | Distance Field Time | Geom Extract. Time | Geom. Size | Frame Rate |
|---|---|---|---|---|---|---|
| tooth | 103 * 94 * 161 | 11 | 6.91308s | 1.58783s | 150,800 | 17.85 |
| | | 39 | 13.7605s | 2.05943s | 603,732 | 4.68 |
| enzo | 256*256*256 | 8 | 63.8571s | 16.8751s | 1,748,868 | 1.44 |
| | | 17 | 67.4505s | 24.2812s | 7,446,081 | 0.38 |
| combustion | 170*160*140 | 12 | 16.5805s | 3.94118s | 531,554 | 5.57 |
| | | 30 | 18.7608s | 4.21217s | 711,794 | 4.09 |
| ethaneDiol | 115*116*134 | 9 | 6.84636s | 1.66709s | 4,948 | 150.29 |
| | | 15 | 7.4306s | 1.71076s | 8,140 | 217.95 |

## 9. Material Boundaries in Acquired Data

As seen in Section 2.3, multi-dimensional transfer functions often use isovalue and gradient magnitude to highlight material boundaries in DVR [KKH02]. Given the shared dependence on the continuous scatterplot, the question is whether fiber surfaces can extract corresponding geometric surfaces.

In our first example (Figure 7), we use the same CT scan of a tooth used by Kniss et al. [KKH02]. While isosurfaces can be chosen in (a) to segment the pulp (red), the dentin (blue) and the enamel (white), this does not isolate the boundary between the dentin and the enamel. However, the continuous scatterplot (b) exhibits clear features corresponding to material boundaries which we select with polygons. Note that the isosurface corresponding to the enamel boundary (white, (a)) spans two distinct features, which can be isolated with separate polygons (white and yellow, (b)).

While these segmentations are not new, they have been used previously to generate images, not geometric surfaces. In contrast, fiber surfaces are geometric, allowing faster rendering as well as reducing noise and removing occluding features (c), (d). Finally, since we define fiber surfaces with respect to the distance field of the polygon, we can choose other fiber surfaces nested inside (or outside) the original selection, and derive further information geometrically.

## 10. Comparisons

In previous sections, we showed how fiber surfaces extract meaningful bivariate features. We now compare them to DVR. Figure 8 compares fiber surface visualization to the volume rendering method of Kotava et al. [KKS*12]. In addition to filtering out small features by size to reduce occlusion, fiber surfaces render at much higher frame rates than DVR, at the expense of additional preprocessing time.

## 11. Implementation

The Fiber Surface algorithm was implemented as a filter for Visualization Toolkit (VTK) 6.1.0, with a user interface based on Qt 4.8. As proof-of-concept, our initial implementation favours simplicity and generality over performance. For convenience and runtime performance, we compute a distance field explicitly as a two-dimensional array and look-up, but apply no other optimizations.

We also report some performance statistics. Since range polygons capture surface structures in the domain, the performance of computation and rendering is directly related to the shape and position of polygons. Since polygon shape can be defined arbitrarily, we have simply chosen two polygons for each data set with different shape complexity.

Table 1 gives these statistics, including the resolution of each data set, the number of edges in the polygon, the computation time for the distance field of the polygon, the computation time to extract the fiber surface, the number of triangles in the fiber surface, and the frame rate reported by vtkRenderer. In each case, the continuous scatter plot was pre-computed as a texture in a using the original authors' implementation [BW08], which took from a few seconds to up to 8 minutes (enzo data-set).

Table 1 shows that the computation is dominated by the distance field computation and fiber surface extraction, and we intend to optimize these stages in future. Equally, while the frame rates drop below 1fps for enzo (the cosmology data set), this occurs when we hit 7.5 million triangles, and we expect that a more recent video card would improve this.

## 12. Conclusions & Future Work

We have generalized isosurfaces to bivariate data to produce well-defined geometric surfaces that isolate regions corresponding to features identified in multi-dimensional transfer functions, and that are easy to implement. Since we generalize from isosurfaces, many of the well-known optimizations for Marching Cubes should apply, and we intend to do so. In particular, we expect that this method will be embarassingly parallelizable, as once the polygon or line is chosen, extraction in each cell is independent.

It is clear that trivariate functions could use three-dimensional range widgets, but this will be more complex, and we leave this for future work. Beyond trivariate, we expect to develop constructs similar to fiber surfaces, but these are more likely to be useful analytically than visually.

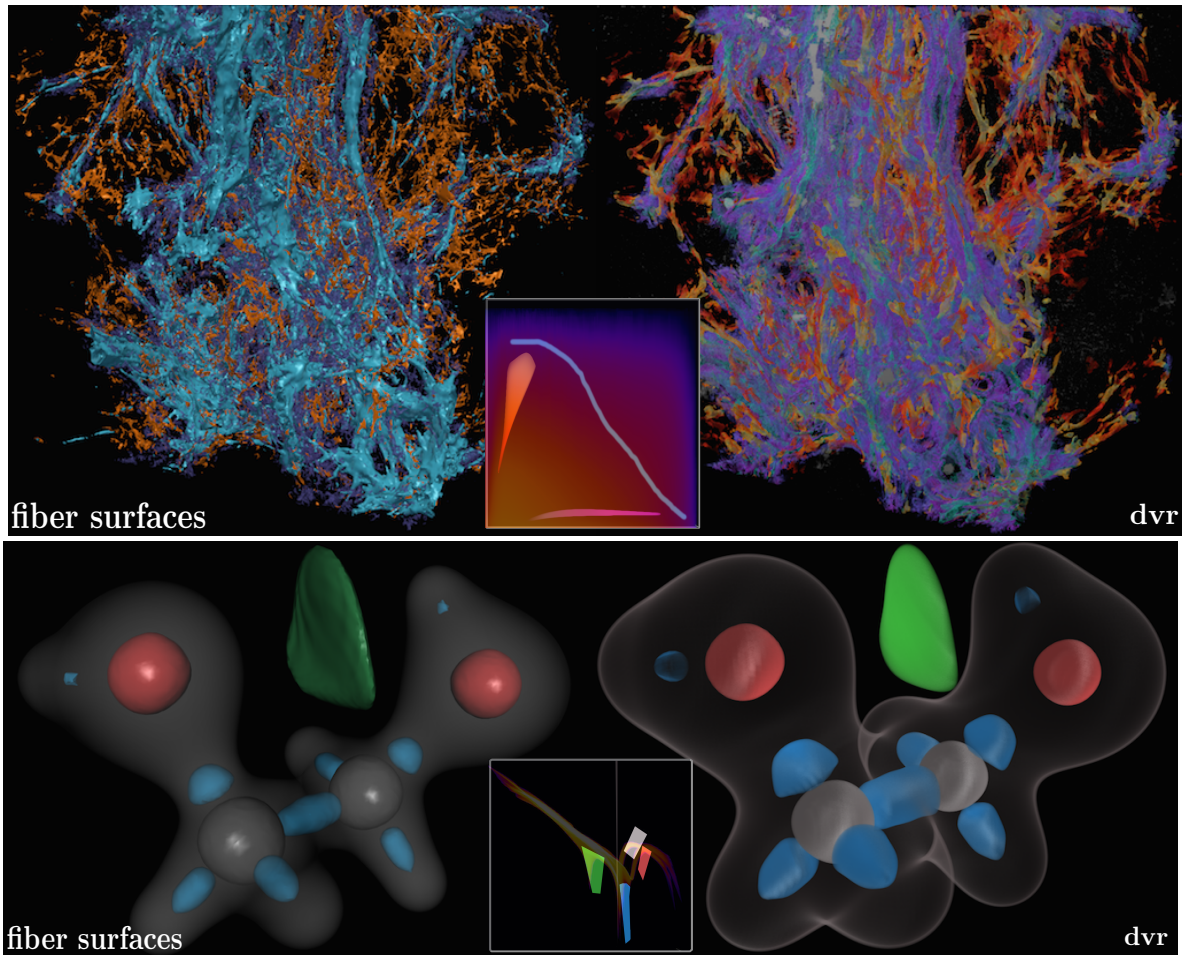


**Figure 8:** *Fiber surfaces compared with direct volume rendering. Top: cosmology (enzo) simulation. Bottom: chemistry (ethane-diol) simulation. Direct volume rendering was optimized with peak finding [KKS*12] to identify the fiber surface boundaries (this gives a roughly 2–4x performance improvement over naive volume rendering). Fiber surfacing exhibits better contrast, enables geometric analyses and better rendering performance via rasterization.*

As with isosurfaces, topological analysis can be applied to fiber surfaces: this work is already underway [CD14]. We felt it was important, however, to report the purely geometric solution first, as it is simpler to understand and implement.

## 13. Acknowledgements

Acknowledgements are due to EPSRC grant EP/J013072/1 for funding this work at Leeds, to the grants LABEX Calsimlab ANR-11-LABX-0037-01 and DGA DT-SCAT-DA-IDF FD1300034MNRBC at UMPC, and to grant NSF CISE ACI-0904631 at Utah.

## References

[Blo88] BLOOMENTHAL J.: Polygonization of implicit surfaces. *Computer Aided Geometric Design* (1988), 341–355. 2

[BW08] BACHTHALER S., WEISKOPF D.: Continuous Scatterplots. *IEEE Transactions on Visualization and Computer Graphics (Proc. of IEEE VIS)* (2008). 2, 5, 8

[CD14] CARR H., DUKE D.: Joint Contour Nets. *IEEE Transactions on Visualization & Computer Graphics 20*, 8 (2014), 1100–1113. 2, 9

[CGJK*11] CONTRERAS-GARCÍA J., JOHNSON E. R., KEINAN S., CHAUDRET R., PIQUEMAL J.-P., BERATAN D. N., YANG W.: NCIPLOT: A Program for Plotting Noncovalent Interaction Regions. *Journal of Chemical Theory and Computation 7*, 3 (2011). 6

[Dol07] DOLEISCH H.: SimVis: Interactive visual analysis of large and time-dependent 3D simulation data. In *39th IEEE Conference on Winter Simulation* (2007), pp. 712–720. 3

[EH04] EDELSBRUNNER H., HARER J.: *Jacobi Sets of Multiple Morse Functions*. Cambridge University Press, 2004, pp. 37–57. 2

[EHP08] EDELSBRUNNER H., HARER J., PATEL A. K.: Reeb Spaces of Piecewise Linear Mappings. In *ACM Symposium on Computational Geometry* (2008). 2

[GABCG*14] GUENTHER D., ALVAREZ-BOTO R., CONTRERAS-GARCIA J., PIQUEMAL J., TIERNY J.: Characterizing molecular interactions in chemical systems. *IEEE Transactions on Visualization & Computer Graphics 20*, 12 (2014), 2476–2485. 6

[GMY11] GUO H., MAO N., YUAN X.: Wysiwyg (what you see is what you get) volume visualization. *IEEE Transactions on Visualization & Computer Graphics 17*, 12 (2011), 2106–2114. 3

[GWK12] GURIJALA K. C., WANG L., KAUFMAN A.: Cumulative heat diffusion using volume gradient operator for volume analysis. *Visualization and Computer Graphics, IEEE Transactions on 18*, 12 (2012), 2069–2077. 3

[GXY11] GUO H., XIAO H., YUAN X.: Multi-dimensional transfer function design based on flexible dimension projection embedded in parallel coordinates. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE* (2011), IEEE, pp. 19–26. 3

[HM03] HUANG R., MA K.-L.: RGVis: region growing based techniques for volume visualization. In *11th Pacific Conference on Computer Graphics and Applications* (2003), pp. 355–363. 2

[IWPN] INSLEY J., WAGNER R., PAPKA M. E., NORMAN M.: Project Stargate Application Driver (http://www.sdsc.edu/Gallery/vd_galaxymodeling.html). 7

[JKMS*10] JOHNSON E., KEINAN S., MORI-SANCHEZ P., CONTRERAS-GARCIA J. COHEN A., YANG W.: Revealing noncovalent interactions. *Journal of the American Chemical Society* (2010). 6

[KH13] KEHRER J., HAUSER H.: Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE Transactions on Visualization & Computer Graphics 19*, 3 (2013), 495–513. 3

[Kin02] KINDLMANN G.: Transfer Functions in Direct Volume Rendering: Design, Interface, Interaction. *Course notes of ACM SIGGRAPH* (2002). 3

[KKH02] KNISS J., KINDLMANN G., HANSEN C.: Multidimensional Transfer Functions for Interactive Volume Rendering. *IEEE Transactions on Visualization & Computer Graphics 8*, 3 (2002), 270–285. 2, 3, 8

[KKH13] KOTAVA N., KNOLL A., HAGEN H.: Morse–smale decomposition of multivariate transfer function space for separably-sampled volume rendering. *Computer Aided Geometric Design 30*, 6 (2013), 549–556. 3

[KKS*12] KOTAVA N., KNOLL A., SCHOTT M., GARTH C., TRICOCHE X., KESSLER C., COHEN E., HANSEN C. D., PAPKA M. E., HAGEN H.: Volume rendering with multidimensional peak finding. In *IEEE Pacific Visualization Symposium* (2012), pp. 161–168. 3, 8, 9

[KPI*03] KNISS J., PREMOZE S., IKITS M., LEFOHN A., HANSEN C., PRAUN E.: Gaussian transfer functions for multi-field volume visualization. In *Proceedings of IEEE Visualization 2003* (October 2003), pp. 497–504. 3

[KWTM03] KINDLMANN G., WHITAKER R., TASDIZEN T., MOLLER T.: Curvature-based Transfer Functions for Direct Volume Rendering: Methods and Applications. In *IEEE Visualization* (2003), p. 67. 3

[Lai95] LAIDLAW D.: *Geometric Model Extraction from Magnetic Resonance Volume Data*. PhD thesis, California Institute of Technology, 1995. 3

[LC87] LORENSEN W., CLINE H.: Marching cubes: a high resolution 3d surface construction algorithm. *ACM Computer Graphics 21* (1987), 163–169. 2

[Lev88] LEVOY M.: Volume Rendering: Display of Surfaces from Volume Data. *IEEE Computer Graphics and Applications 8*, 3 (1988), 29–37. 3

[LT10] LEHMANN D. J., THEISEL H.: Discontinuities in continuous scatterplots. *IEEE Transactions on Visualization & Computer Graphics 16*, 6 (2010), 1291–1300. 2

[LWT*14] LIU S., WANG B., THIAGARAJAN J. J., BREMER P.-T., PASCUCCI V.: Multivariate volume visualization through dynamic projections. In *IEEE Large Scale Data Analysis and Visualization*. 2014. 3

[MWCE09] MACIEJEWSKI R., WOO I., CHEN W., EBERT D. S.: Structuring feature space: A non-parametric method for volumetric transfer function generation. *IEEE Transactions on Visualization & Computer Graphics 15*, 6 (2009), 1473–1480. 3

[Nie03] NIELSON G. M.: On Marching Cubes. *IEEE Transactions on Visualization and Computer Graphics 9*, 3 (2003), 283–297. 4

[NN11] NAGARAJ S., NATARAJAN V.: Simplification of jacobi sets. In *Topological Methods in Data Analysis and Visualization*. Springer, 2011, pp. 91–102. 2

[NY06] NEWMAN T. S., YI H.: A survey of the marching cubes algorithm. *Computers & Graphics 30*, 5 (2006), 854–879. 2

[Sae04] SAEKI O.: *Topology of Singular Fibers of Differentiable Maps*. No. 1854 in Lecture Notes in Mathematics. Springer, 2004. 3

[SBSG06] SEREDA P., BARTROLI A., SERLIE I., GERRITSEN F.: Visualization of boundaries in volumetric data sets using LH histograms. *IEEE Transactions on Visualization & Computer Graphics 12*, 2 (March 2006), 208–218. 2

[SKK06] SALAMA C., KELLER M., KOHLMANN P.: High-Level User Interfaces for Transfer Function Design with Semantics. *IEEE Transactions on Visualization & Computer Graphics 12*, 5 (Sept 2006), 1021–1028. 3

[SR04] SCHULZE J., RICE A.: Real-time volume rendering of four channel data sets. In *IEEE Visualization* (2004), IEEE Computer Society, pp. 598–34. 3

[SSC*] SAKURAI D., SAEKI O., CARR H., WU H.-Y., YAMAMOTO T., DUKE D., TAKAHASHI S.: Interactive visualisation for fiber singularities of multivariate functions. In submission. 3, 5

[WB96] WEIGLE C., BANKS D.: Complex-valued contour meshing. In *IEEE Visualization* (1996). 2

[Wen13] WENGER R.: *Isosurfaces: Geometry, Topology, and Algorithms*. A K Peters/CRC Press, 2013. 2

[WZL*12] WANG Y., ZHANG J., LEHMANN D. J., THEISEL H., CHI X.: Automating Transfer Function Design with Valley Cell-Based Clustering of 2D Density Plots. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 1295–1304. 3

[ZH13] ZHOU L., HANSEN C.: Transfer function design based on user selected samples for intuitive multivariate volume exploration. In *IEEE Pacific Visualization Symposium (PacificVis) 2013* (2013), IEEE, pp. 73–80. 3

[ZH14] ZHOU L., HANSEN C.: GuideME: Slice-guided Semiautomatic Multivariate Exploration of Volumes. *Computer Graphics Forum 33*, 3 (2014), 151–160. 3

[ZK10] ZHAO X., KAUFMAN A.: Multi-dimensional reduction and transfer function design using parallel coordinates. In *IEEE/EG Volume Graphics* (2010), pp. 69–76. 3