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

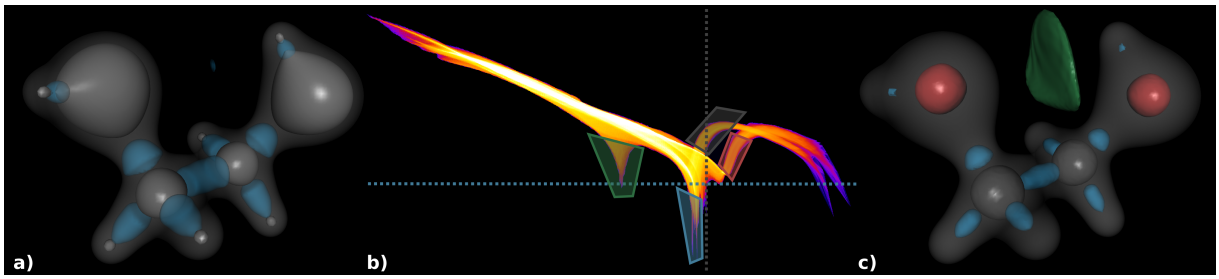

Figure 1: *Fiber Surfaces of electron density and reduced gradient in an ethane-diol molecule: (a) While an isosurface of electron density identifies regions of influence of atoms (grey), it does not distinguish atomic type. An isosurface of reduced gradient identifies bonding interaction sites (blue) 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 multi-variate 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 polylines and 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 the field of scientific visualization has expanded, methods have been developed for direct visualization of scalar and vector fields in particular. However, relatively few methods are known that give effective visualizations for multi-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 composed of individual fibers (the equivalent of contours in multi-fields).

Our contribution is therefore to show:

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

A crucial aspect of this method is that 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 multi-field topological analysis [**?**].

2. Background

Four previous lines of work are relevant to our method: the generalisation of contours / isosurfaces to fibers in multi-fields (Section 3), the use of Marching Cubes to extract isosurfaces and other separating surfaces (Section 2.1), the use of multi-dimensional transfer functions in direct volume rendering (Section 2.3), and methods for multifield visualization (Section 2.2). We will defer discussion of fibers to the next section, and deal with the other related work first.

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 simple 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 for meshes constructed from primitives other than cubes, in particular 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 multifield visualization in $Domf$ are known. 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 [**?**].

Multifields can also be shown as multidimensional histograms, and recent work on continuous scatterplots [**?**] has shown the importance of showing the continuity assumed in mesh representations. Subsequent work has focussed on linear features [LT10] which are now [CD13] known to be related to the topology of the multifield. Further work on multifield topology is ongoing [EH04, EHP08b, NN09], but these methods are complex and not yet fully developed.

2.3. Direct Volume Rendering

Direct volume rendering was initially introduced to provide X-ray-like visualizations of data [**?**] by defining a transfer function to map data values to color and opacity, then integrating light transport along rays through each pixel. Laidlaw [Lai95] constructed multi-dimensional transfer functions over multifields, while others [KKH02, Kin02, KWTM03] showed how to use derived properties such as gradient or curvature to turn scalar fields into multifields. Many variations of this have since been described [KH13],. Of particular interest is the observation that linear features in the range correspond approximately to material boundaries, and interfaces are often designed using polygonal widgets to highlight regions in $Ranf$ [**?**].

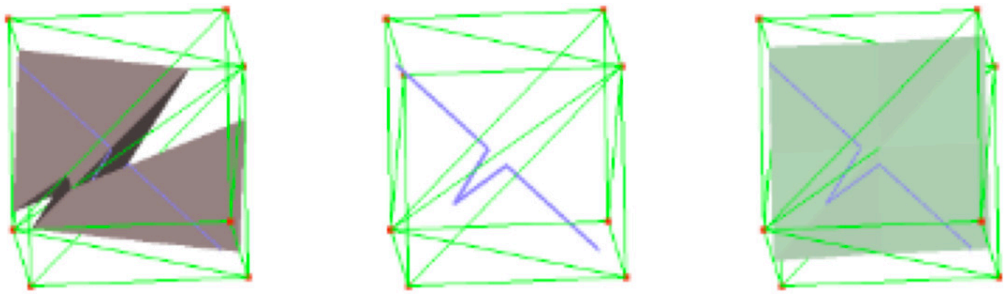

Figure 2: *Example of Fiber Construction. Left: an isosurface of f_1. Centre: a fiber defined by the intersection of isosurfaces. Right: an isosurface of f_2. Both isosurfaces also show the fiber for reference.*

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. And second, it produces an image, not a geometric surface that can also be used for modelling and simulation.

In summary, we can see that geometric surfaces for multifields are of interest across scientific visualization, with particular value to all of the domains where multi-dimensional transfer functions have already been adopted for direct volume rendering. We therefore turn our attention to generalising isosurfaces to the bivariate case, observing that the development is similar for higher dimensions.

3. Fibers

Given a function $f : \mathbb{R}^d \rightarrow \mathbb{R}^r$, the domain $Dom f = \mathbb{R}^d$ is the set of input values: in scientific visualization, d is most commonly $2, 3$, or 4. In contrast, the range $Ran f = \mathbb{R}^r$ is the set of output values: in scientific visualization, this may be scalar ($d = 1$), bivariate ($d = 2$), vector ($d = r$), with special semantic meaning), or higher dimensions. In this paper, while we focus on the bivariate case, the approach is generally applicable to higher dimensions as well. For convenience, we will assume $d = 3$ unless otherwise specified, and use f_1, f_2 to refer to the two output variables. We have also assumed a simply connected domain, but the definitions and our method apply to more general manifolds as well.

For bivariate (and higher) data, inverse images are well-defined. In general, this inverse image is referred to as a *fiber* [**?**], and is analogous to a contour. In the $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ case, a fiber is defined by a point $h = (h_1, h_2) \in Ran f$, and can be found by intersecting the isosurface of h_1 in f_1 and the isosurface of h_2 in f_2, as shown in Figure 3.

Note that, where a contour is of co-dimension 1 (i.e. one dimension less than the domain itself), this fiber is of co-dimension 2, and is therefore a 1-manifold structure similar to a contour line in a $\mathbb{R}^2 \rightarrow \mathbb{R}^1$ scalar field. This happens because our point $h = (h_1, h_2)$ effectively specifies two parameters, each of which reduces the dimensionality of the data by one: as a result, we reduce the data too much for useful visualization. Moreover, as a result of this, a single fiber cannot separate regions, as shown in Figure 3.

4. Fiber Surfaces

The challenge is to find a generalization of contours to bivariate volumes that produces well-defined surfaces that separate regions. 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 $f^{-1}(h)$. As a result, any path in $Ran f$ will correspond to a set of fibers that varies continuously in the domain, sweeping out a surface or set of surfaces.

We note that 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.

To remain consistent with this usage, we will use *fiber surface* to refer to the inverse image in $Dom f$ of any path $P \in Ran f$, and *fiber surface component* to refer to a single connected component of a fiber surface.

Fiber surfaces have two properties similar to isosurfaces which we will exploit. First, as already noted, each fiber surface component is a continuous surface. Second, if the path P separates $Ran f$ into regions, then the corresponding fiber

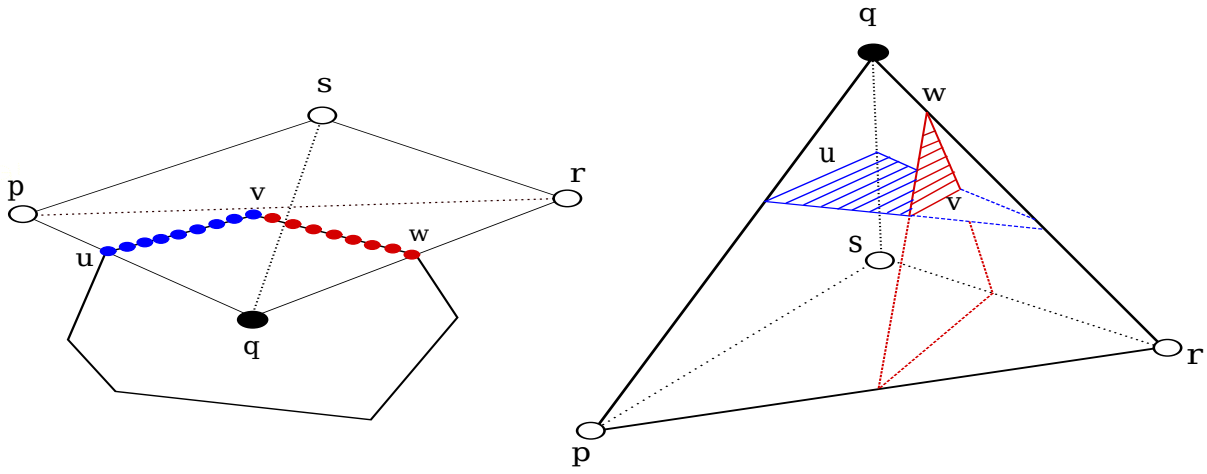

Figure 3: *Fiber Surface of a Polygon. Left: the polygon in the range with the projection of a single tetrahedra. Right: the tetrahedron in the range with the fibers corresponding to edges uv, vw. Each fiber is a point in the range but a line in the domain. Each line in the range corresponds to a plane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

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.

5. Fiber Surfaces of Lines in the Range

The simplest fiber surfaces are defined by lines in the range, expressed in normal form: $\vec{n} \cdot p = c$, where $\vec{n} = (n_1, n_2)$. Here, c measures the perpendicular distance from the origin to the line, scaled by the magnitude of $\vec{n}$. 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 it follows that $n_1 f_1(x) + n_2 f_2(x) = c$ for all such points. Thus, this fiber surface is simply an isosurface at isovalue c of the weighted sum $\vec{n} \cdot f$, i.e. an isosurface of a scalar field derived from the original bivariate field through a linear combination. Doing this at runtime is trivial: we need only pass both vector $\vec{n}$ and isovalue c to Marching Cubes, and compute derived values at vertices on the fly.

If f is defined on a tetrahedron, it then follows that fiber surface of a line is a plane, as shown in Figure **??**. For a cubic mesh with trilinear interpolation, Nielson's cases [**?**] 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 $\vec{n}$ but have different isovalues c. This implies that the corresponding lines int the range are parallel, with c being the distance from the origin to the line. We define the line through the origin to be the reference line, and observe that each other line is at a fixed distance 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 Curves and Polylines

Once we have recognized the fiber surfaces of lines as isosurfaces of derived linear combinations, we turn our attention to the general case: arbitrary curves, polylines and polygons. We will assume initially that we have a separating polygon: i.e. a closed loop of line segments in the range.

Again, we start from the observations that each point on the polygon corresponds to a fiber, and that f is a continuous mapping. As we travel around the polygon, 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 **??**, in which we show a single tetrahedron with the fiber surface defined near a vertex v of a polygon in the range. Each linear segment in the range is a subset of a range line fiber surface, each of which is planar in the tetra-

hedron, 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.

Since the polygon has an inside and outside, mesh vertices whose fibers are inside the polygon are also inside the fiber surface. Correspondingly, vertices whose fibers are outside the polygon 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 exactly if $f(u)$ and $f(v)$ are on opposite sides of the polygon boundary. Thus, iff the line segment $f(u)f(v)$ intersects P at some point $w \in Ranf$, the cube 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, for computing normals of the fiber surface thus approximated, we can again use flat shading or averaged normals. Alternately, we can identify which edge of the polygon was intersected, and use its normal vector $\vec{n}$ to weight the gradient components of f as before.

Algorithm 1 Algorithm for extracting Fiber Surfaces

Require: Function f, Mesh M in $Domf$, Polygon P in $Ranf$

```
for each cell C in mesh M do
  for each vertex V in C do
    if V is in polygon P then
      Classify V as black
    else
      Classify V as white
    end if
  end for
  Compute case MCC 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
```

This leads to the algorithm in Algorithm 1. While this is on principle computable, it is heavily dependent on relatively slow geometric tests. Instead, we return to the observation in 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 polygon is defined by the zero-contour of the polygon's signed distance field. Moreover, given linear interpolation, interpolating in the distance field will compute the same w as before, without requiring us to perform either point in polygon tests or line intersection tests.

Finally, we note that if we compute the distance field for a polygon, and choose the contours at distances other than zero, 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

Once we have understood what fiber surfaces are and how to extract them, the next task is to build a user interface to define them in a simple but meaningful way. In the longer run, we expect user interfaces for fiber surfaces 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 their effectiveness must be deferred to the point at which domain-specific interfaces are constructed. We therefore focus on demonstrating the type of interface is feasible, and therefore restrict our attention to simple proof-of-concept interfaces.

Clearly, one crucial element of a fiber surface interface is to show the range of the function as well as the fiber surfaces in the range, as shown in Figure **??**. 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 [**?**] of the bivariate function f in the range, and superimpose lines and polygons on it. As the continuous scatterplot is fixed for a given dataset, we will assume that this has been precomputed.

The second major decision is whether to use lines, polylines 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 polylines or polygons are the more general case: we therefore an interface using polygons.

Thus, our interface consists primarily of two widgets: a range widget and a domain widget, as previously used by Sakurai et al. [**?**]. In the range widget, we provide the ability to edit a polygon, 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.

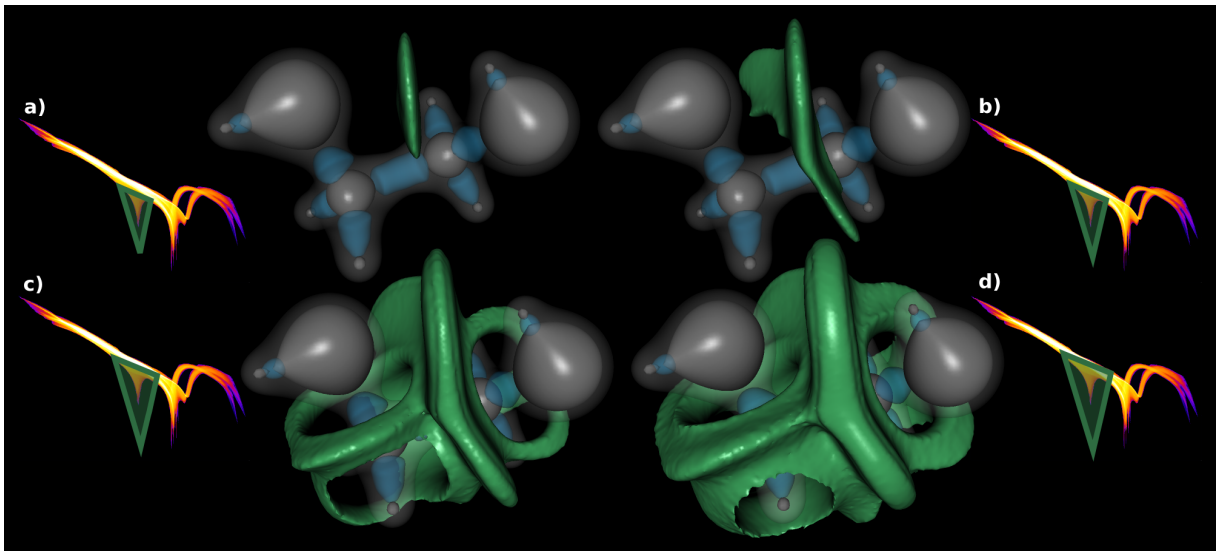

Figure 4: *From left to right: fiber surfaces obtained by increasing the range distance to the initial fiber polygon.*

8. Fiber Surface Examples

Given this user interface, the next step is to apply fiber surfaces to visualization. In this section, we show the general case for applications in chemistry, cosmology and combustion: in the next section we specialize fiber surfaces to material boundaries in multi-dimensional transfer functions.

In (Figure 1), we apply fiber surface extraction to a bivariate data-set representing an ethanediol molecule with the electron density and a derived property known as the reduced gradient [JKMS*10], which is related exponentially to the electron density in regions without chemical interaction. As a result, the continuous scatterplot of the logarithms of these properties appears along an axis (1(b)). Elsewhere, chemical interactions occur where these properties do not have this simple relationship, and in particular, around the atoms (2 Carbon, 2 Oxygen, 6 Hydrogen), the covalent (C-C, C-H, C-O) and non-covalent (O-H) atomic bonds.

In this dataset, an isosurface of electron density can be used to capture regions of influence of the atoms (in grey, 1(a)), but does not distinguish between types of atoms. However, since the different types of atom differ in their reduced gradient, it is simple to select a polygon in the range for each type (1(c), oxygen in red, carbon in grey). Similarly, an isosurface of the reduced gradient (1(b)) captures bonds (blue), but does not distinguish between covalent and non-covalent bonds: using a polygon to select a sub-range of the electron density distinguishes bond types.

We note that Guenther et al. [GABCG*14] used topology to segment the interaction sites, but needed additional processing to distinguish covalent and non-covalent bonds. In contrast, fiber surfaces use polygons in the range to achieve a similar result, but is considerably simpler to implement.

Interestingly, fiber surfaces also revealed other structures, as shown in Figure 4. Here, the downward spike in the continuous scatterplot was selected to investigate chemical reactions off the main line, and a planar interface between the two alcohol (OH) groups was observed. It was then straightforward to explore this interface by choosing different constants with respect to the polygon, as discussed in Section 6. The resulting surfaces, in a) - d), could be interpreted as candidate regions for non-covalent interactions in the presence of perturbation for uncertainty assessments.

Our second example (Figure 5) comes from cosmology, with a bi-variate data-set representing a time-step in a simulation of universe expansion. Here, the first value represents concentrations of matter (white isosurfaces), while the second value represents concentrations of dark matter (grey isosurfaces). Scientifically, the expectation is that high concentrations of dark matter will principally be co-located with high concentrations of matter, but not vice versa. As a result, analysis of this data focusses on “bubbles”, where both properties are locally high and “filaments” with high concentrations of matter but not dark matter, that are expected to connect the bubbles. The relevant thresholds are, however, unknown and are expected to vary across the data set.

Thus, if we choose low isovalues for both properties (5(a)), the matter isosurface has one primary component that contains most of the volume, while the dark matter isosurface has many small connected components. In contrast, at higher isovalues (5(b)), dark matter isosurfaces exhibit less

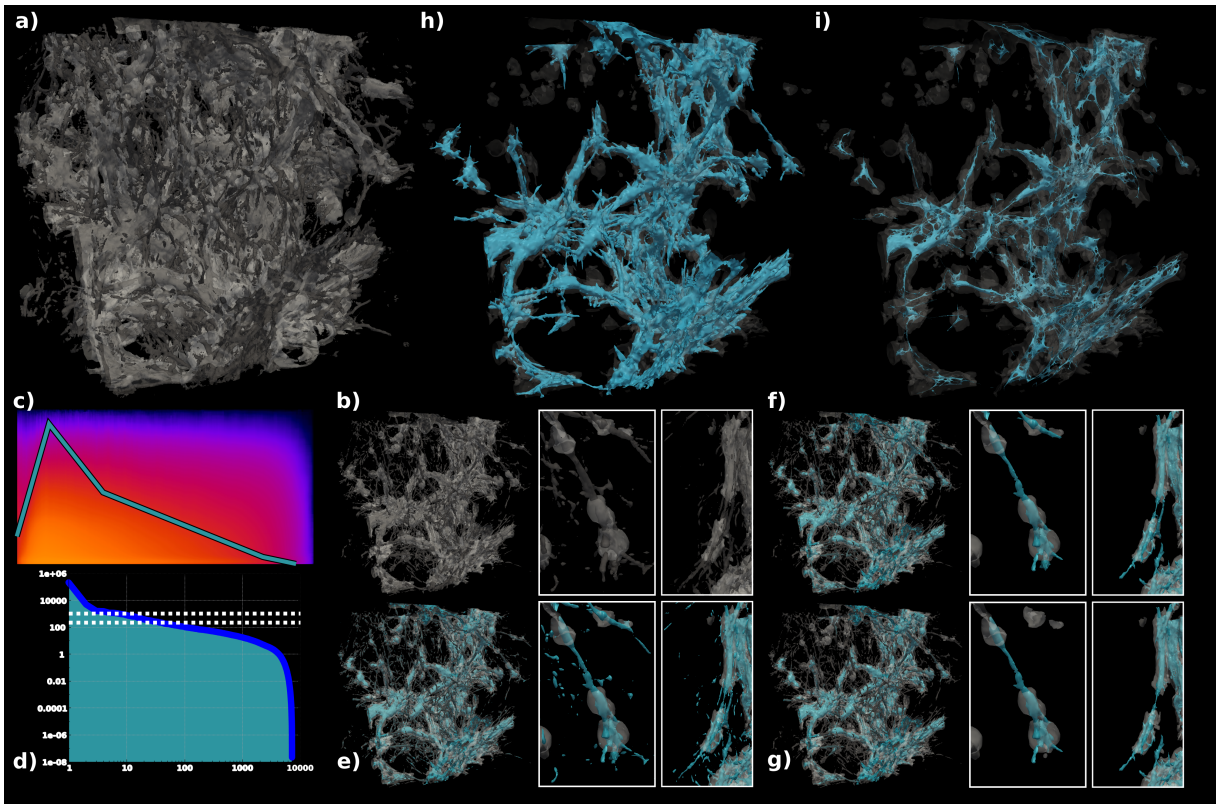

Figure 5: *Fiber Surfaces of a cosmological simulation of universe expansion: (a, b) Isosurfaces of matter (white) and dark matter (grey) concentrations for different pairs of isovalues (a: low, b: high). (c) Continuous scatter plot of matter vs. dark matter distribution. (d) Distribution by volume of fiber surface components. (e) Progressive volume filtering of the fiber surface components (dashed lines in (d)). (f) Fiber surface contraction through smoothing passes to enhance the visual display of bubbles and filaments.*

noise and matter isosurface occupies less space and better reflects the backbone structure of matter which links galaxies throughout the expansion process. However, as shown in zoom-in views, this backbone structure is composed of both isosurfaces (matter and dark matter).

If we use a polygon to select the main feature visible in the continuous scatterplot, (5(c)), the fiber surface clearly connects matter isosurface components through dark matter isosurface components (zoom-in views), hence better capturing the overall backbone structure of the galaxies.

Moreover, since fiber surfaces are explicit geometry, we filter the individual connected components by their contained volumes (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(e)). 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(f)). In the process, features that look very like the expected bubbles and filaments start appearing (5(f)).

In our third example, we look at two properties in a combustion simulation: temperature and temperature gradient magnitude (Figure 6). Here, features are often 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.

9. Material Boundaries in Acquired Data

As observed in Section 2.3, multi-dimensional transfer functions are frequently applied to the combination of isovalue and gradient magnitude to highlight material boundaries for direct volume rendering [KKH02]. Given the shared depen-

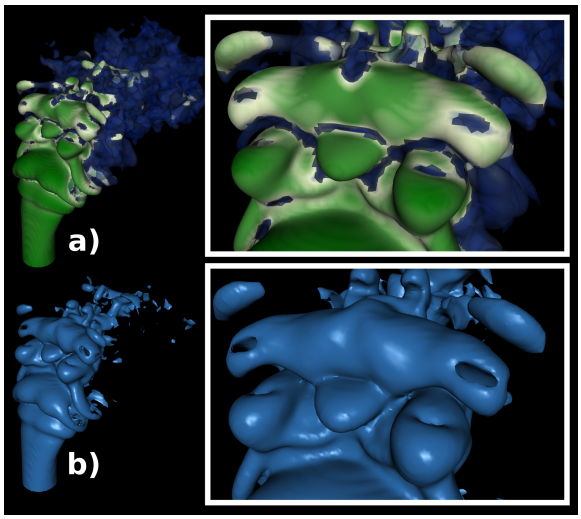

Figure 6: *Extracting the core of a burning flame: (a) Isotherm thresholded (opaque triangles) on temperature gradient magnitude(color map). (b) Extracting the same feature with fiber surfaces guarantees closed 2-manifolds.*

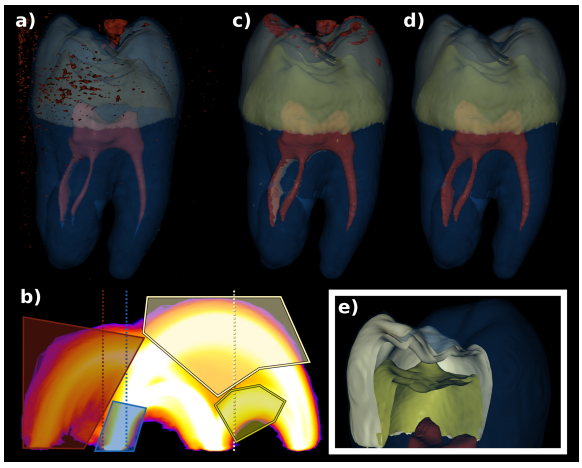

Figure 7: *Fiber surfaces for material boundaries in a tooth CT-scan: (a) User selected isosurfaces. (b) Continuous scatter plot of isovalue vs. gradient magnitude with user selected isovalues (dashed lines) and polygons. (c) Fiber surfaces of the selected polygons. (d) Fiber surfaces after connected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

dence on regions in the continuous scatterplot, the immediate question is whether fiber surfaces can extract the corresponding surfaces geometrically. In this section, we show that we can do exactly this with fiber 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.

Figure 8 shows another data-set (the engine) commonly used with multi-dimensional transfer functions. While the continuous scatterplot exhibits clear arc-shaped features (a), the pre-images of their center-lines yield noisy fiber surfaces (b). This is due to the fact that for this data set, most of the volume is air. Thus, these arcs represent air points and not material boundaries. Using the distance field of these polyg-

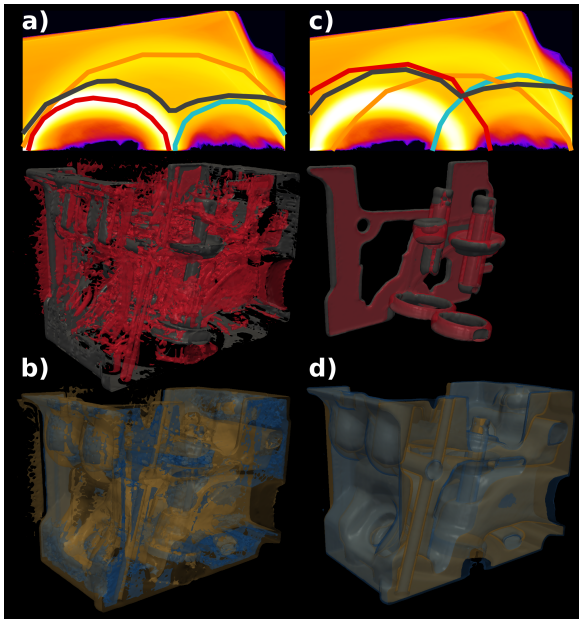

Figure 8: *Fiber surfaces of a CT scan of an engine: (a) Feature-driven user selection of polygons in continuous scatterplot. (b) Corresponding fiber surfaces. (c) Range-space distance field exploration based on the initial polygons. (d) Corresponding fiber surfaces.*

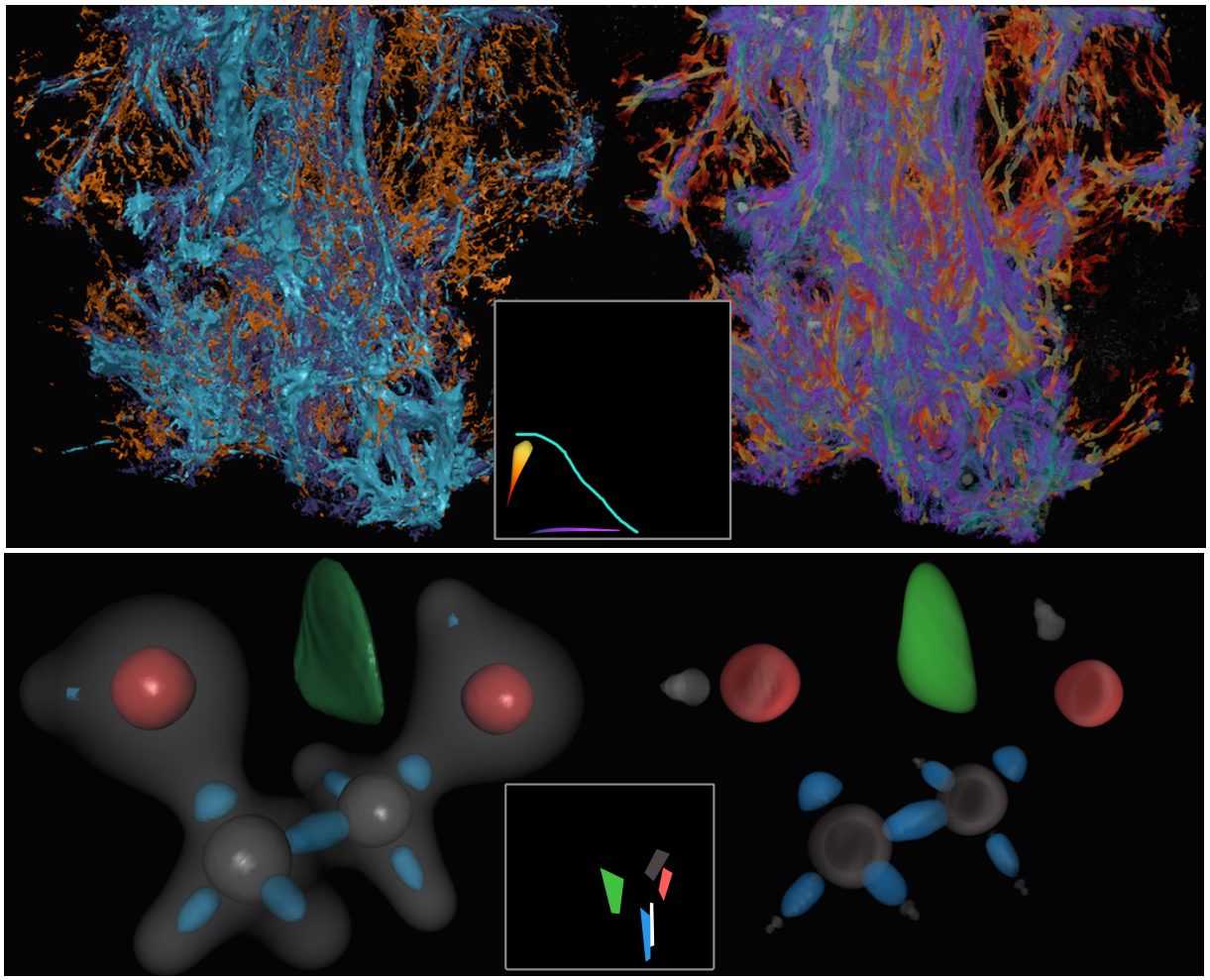

Figure 9: *Fiber surfaces compared with multivariate volume rendering. Top left bottom right: tooth, enzo, and ethane diol datasets volume render at 30, 15 and 6 fps, respectively at 1 MP, optimized with peak finding [KKS*12] to identify the fiber surface boundaries (this gives a roughly 2–4x performance improvement over naive volume rendering). Fiber surfacing exhibits better contrast, enables geometric analyses and better rendering performance via rasterization.*

onal boundaries (c), however, makes it easy to extract sharp boundaries between the distinct metals of the engine (d).

9.1. Comparisons

In the previous sections, we showed how to use fiber surfaces to extract meaningful bivariate features. We now turn our attention to how they compare with direct volume rendering. Figure 9 compares fiber surface visualization with the volume rendering method by 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 direct volume rendering, although at the expense of additional preprocessing time.

10. Implementation

The Fiber Surface algorithm has been implemented as a filter for the Visualization Toolkit (VTK), version 6.1.0, with a user interface designed using Qt 4.8. As a 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. We have not implemented any other optimizations.

In this section, we also report the performance statistics of our Fiber Surface implementation. Given the fact that polygon defined in the range will capture surface structures in the domain, the performance of computation and rendering is

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

Data	Resolution	Polygon Edge Nr	Distance Field Time	Geom Extract. Time	Geom. Size	Geom. Render Time
tooth	103 * 94 * 161	11	6.91308s	1.58783s	150800	0.05603s
		39	13.7605s	2.05943s	603732	0.213837s
toy	128 * 128 * 128	8	8.53906s	2.08904s	170704	0.063076s
		28	10.5034s	1.98428s	19792	0.00845408s
engine	256*256*110	16	31.0106s	7.73815s	916388	0.305289s
		6	28.8783s	6.85951s	19958	0.029253s
enzo	256*256*256	8	63.8571s	16.8751s	1748868	0.694394s
		17	67.4505s	24.2812s	7446081	2.63s
combustion	170*160*140	12	16.5805s	3.94118s	531554	0.179521s
		30	18.7608s	4.21217s	711794	0.244361s
ethaneDiol	115*116*134	9	6.84636s	1.66709s	4948	0.00665402s
		15	7.4306s	1.71076s	8140	0.00458813s

directly related to the shape and position of polygons. Considering polygon shape can be defined arbitrarily, in this run of performance testing we have chosen two polygons each data set with different shape complexity. Testing has been carried out on both synthetic datasets (allowing for verification of the Fiber Suface implementation), and real data. Table 1 gives the dataset statistics and runtime results. These include: the resolution of each data set; the number of edges in the polygon; the computation time to generate distance field to the polygon over the range; the computation time to extract the fiber surface; the number of triangles in the fiber surface; and the time for rendering the geometry.

Table 1 shows large fraction of timings are covered by distance field computation and fiber surface extraction. With respect to distance field computation, we could notice that it is not much affected by the complexity of polygon in most of the data sets. On the other hand, the fiber surface extraction is clearly depend on the scale of the output geometry.

11. Conclusions & Future Work

We have shown a generalization of isosurfaces to bivariate data that produces well-defined geometric surfaces, that isolates regions with respect to both properties, that is easy to implement and whose surfaces correspond to features identified in multi-dimensional transfer functions.

Given the relationship between our method and isosurfaces, many of the well-known optimizations for Marching Cubes should be transferable, and we intend to do so. In particular, we expect that this method will be embarassingly parallelizable, as once the polygon or line has been chosen, the extraction can be distributed to as many cores as are available.

We observe that, as with isosurfaces, multiple fiber surface components are defined by a given primitive. Therefore, the logical next step is to use topological methods [**?**] to track, analyse and filter fiber surface components. As it happens, this work is already underway [**?**]. We felt that it was important, however, to demonstrate the purely geometric extraction of fiber surfaces before proceeding to this step, as topological methods are known to be more complex to implement, and should be avoided where simpler methods produce the same result.

References

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

[CD13] CARR H., DUKE D.: Joint Contour Net. *IEEE Transactions on Visualization and Computer Graphics* (2013), xxx–xxx. accepted. 2

[EH04] EDELSBRUNNER H., HARER J.: Jacobi Sets of Multiple Morse Functions. *In Foundations of Computational Matematics, Minneapolis, 2002* (2004), 37–57. Cambridge Univ. Press, 2004. 2

[EHP08] EDELSBRUNNER H., HARER J., PATEL A. K.: Reeb Spaces of Piecewise Linear Mappings. In *SoCG* (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 and Computer Graphics (Proc. of IEEE VIS)* (2014). 6

[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. *Visualization and Computer Graphics, IEEE Transactions on 19*, 3 (2013), 495–513. 2

[Kin02] KINDLMANN G.: Transfer Functions in Direct Volume Rendering: Design, Interface, Interaction. *Course notes of ACM SIGGRAPH* (2002). 2

[KKH02] KNISS J., KINDLMANN G., HANSEN C.: Multidimensional Transfer Functions for Interactive Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics 8*, 3 (2002), 270–285. 2, 7, 8

[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 *Pacific Visualization Symposium (PacificVis), 2012 IEEE* (2012), IEEE, pp. 161–168. 9

[KWTM03] KINDLMANN G., WHITAKER R., TASDIZEN T., MOLLER T.: Curvature-based Transfer Functions for Direct Volume Rendering: Methods and Applications. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)* (2003), IEEE Computer Society, p. 67. 2

[Lai95] LAIDLAW D.: *Geometric Model Extraction from Magnetic Resonance Volume Data*. PhD thesis, California Institute of Technology, 1995. 2

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

[LT10] LEHMANN D. J., THEISEL H.: Discontinuities in continuous scatterplots. *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS* (2010). 2

[NN09] NAGARAJ S., NATARAJAN V.: Simplification of jacobi sets. In *TopoInVis* (2009). 2

[NY06] NEWMAN T. S., YI H.: A survey of the marching cubes algorithm. 854–879. 2

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