Surface Extraction from Multi-field Particle Volume Data Using Multi-dimensional Cluster Visualization

Lars Linsen, Tran Van Long, Paul Rosenthal, and Stephan Rosswog

Abstract—Data sets resulting from physical simulations typically contain a multitude of physical variables. It is, therefore, desirable that visualization methods take into account the entire multi-field volume data rather than concentrating on one variable. We present a visualization approach based on surface extraction from multi-field particle volume data. The surfaces segment the data with respect to the underlying multi-variate function. Decisions on segmentation properties are based on the analysis of the multi-dimensional feature space. The feature space exploration is performed by an automated multi-dimensional hierarchical clustering method, whose resulting density clusters are shown in the form of density level sets in a 3D star coordinate layout. In the star coordinate layout, the user can select clusters of interest. A selected cluster in feature space corresponds to a segmenting surface in object space. Based on the segmentation property induced by the cluster membership, we extract a surface from the volume data. Our driving applications are Smoothed Particle Hydrodynamics (SPH) simulations, where each particle carries multiple properties. The data sets are given in the form of unstructured point-based volume data. We directly extract our surfaces from such data without prior resampling or grid generation. The surface extraction computes individual points on the surface, which is supported by an efficient neighborhood computation. The extracted surface points are rendered using point-based rendering operations. Our approach combines methods in scientific visualization for object-space operations with methods in information visualization for feature-space operations.

Index Terms—Multi-field and multi-variate visualization, isosurfaces and surface extraction, point-based visualization, star coordinates, visualization in astrophysics, particle simulations.

1 Introduction

The Smoothed Particle Hydrodynamics (SPH) method is frequently employed in astrophysics because of its natural adaptivity. Being completely Lagrangian, the interpolation points ("particles") follow the hydrodynamic flow and are thus not constrained by any prescribed grid structure. Since the particle masses are usually kept fixed to exactly conserve mass, this naturally leads to well-sampled high-density regions with the drawback that low-mass density zones are less well sampled by SPH particles.Consequently, SPH simulations often produce highly clustered particle distributions. In Section 2, we give more background information on SPH simulations.

Navratil et al. [31] expressed in their IEEE Visualization 2007 application paper that it would be desirable to have surface extraction techniques that directly operate on cosmological particle-based data, i. e. point-based volume data. Since there was no tool publically available that provided such functionality, they decided to use an inverse-distance-based interpolation over a regular grid prior to applying standard grid-based surface extraction.

The preference to avoid such a resampling is based on the potentially high error introduced when operating on highly adaptive SPH data. An extremely high sampling rate would have to be used to ensure that no information is lost in well-resolved high density areas, i. e. small areas with many particles. With such a high sampling rate one would introduce a lot of redundancy in less densely populated areas and would significantly blow up the amount of data to a hardly manageable size. To avoid such an effect, one would have to use an adaptive resampling, which would no longer allow for the direct application of standard visualization techniques. Instead of trying to deal with such adaptively resampled meshes, it would be more desirable to avoid resampling errors completely by directly operating on the particle data. Hence, we developed visualization techniques that directly operate on point-based volume data with a large number of points.

In most astrophysical simulations the underlying 3D data field is multi-valued and includes several scalar quantities such as velocities, temperature, density, and, sometimes, mass fractions of different chemical species. For the astrophysicists it is of great interest to examine interfaces between different materials, e. g. fuel and ashes, and to observe the propagation of (e. g. ionization or burning) fronts over time. For their visualization it is intuitive to use surface extractions.

The multi-dimensional feature space of the data sets will be analyzed by applying automatic components coupled with intuitive interaction mechanisms operating on appropriate visualization layouts. An automatic multi-dimensional clustering based on density computations is proposed, whose results can be interactively analyzed after projecting it into non-orthogonal coordinates such as star coordinates [23, 41]. The star coordinate layout is optimized to separate projected clusters while maintaining the structure of the individual clusters. Interaction techniques in projected star coordinate space shall allow for selection, refinement, and modification of clusters as well as for reconfiguration of the parameter space and filtering of individual data points. The feature space operations are described in Section 4.

The feature extraction result needs to be visualized in the volumetric object space. We developed a boundary surface extraction method for multi-dimensional features that directly operates on point-based volume data. The boundary surface needs to separate those particles that belong to the cluster from those that do not. Hence, local neighborhood information for particles is required. We developed a *k*d-tree structure with an efficient indexing scheme that supports efficient neighborhood estimation. Surface boundaries are extracted in the form of a set of points located on the surfaces. These point clouds are rendered using point-based surface rendering methods. The object space operations are described in Section 5.

The coupled system of feature-space operations including automatic clustering and interactive visual exploration and the object-space operations including efficient direct surface extraction and rendering makes for a powerful analysis tool of multi-field SPH simulation data. In Section 6, we demonstrate how these techniques can be applied for an intuitive analysis of real data.

2 Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics (SPH) is a completely Lagrangian, meshfree scheme. Originally invented in an astrophysical context [27,

- *All authors are with the School of Engineering and Science, Jacobs University, Bremen, Germany, E-mails: l.linsen@jacobs-university.de, v.tran@jacobs-university.de, p.rosenthal@jacobs-university.de, s.rosswog@jacobs-university.de.*

Manuscript received 31 March 2008; accepted 1 August 2008; posted online 19 October 2008; mailed on 13 October 2008.
For information on obtaining reprints of this article, please send e-mailto:tvcg@computer.org.

29] SPH has by now found its way into a wealth of different applications as diverse as breaking waves on beaches, liquid metal moulding, elasticity and fracture problems, traffic flow simulations and giant impacts on planets, see Monaghan [30] for a recent review.

Apart from its natural adaptivity SPH's main strength is its excellent conservation properties: even in discrete form it conserves mass, energy, linear and angular momentum by construction. For an ideal fluid, no more than a Lagrangian, a prescription on how to obtain the matter density by summation, and the first law of thermodynamics are needed to obtain a completely conservative set of discretized equations.

The main idea is to approximate a quantity f at a point $\vec{r}$ by summing up contributions from the neighboring particles, each weighted by a kernel function W

$$f(\vec{r}) = \sum_b \frac{m_b}{\rho_b} f_b W(\vec{r}-\vec{r}_b, h) \ , \qquad (1)$$

where m_b is the particle mass, ρ_b the density and f_b the function value at the position of particle b. The width of the kernel function W is determined by the so-called smoothing length, h. The SPH prescription for derivatives is to take the exact derivative of the above function approximation $\nabla f(\vec{r}) = \sum_b \frac{m_b}{\rho_b} f_b \nabla W(\vec{r}-\vec{r}_b, h)$.

In astrophysics, the method is mainly used to simulate the three-dimensional Euler equations, often with a large variety of additional physical effects such as radiative transfer [46], magnetic fields [36], or nuclear burning [49]. For the examples given in this paper, we used a cubic spline for kernel W with smoothing length $h = 2$.

3 Related Work

3.1 Multi-field Spatial Data Visualization

Most spatial data visualization focus on one parameter, which may be scalar, vector, or tensor-valued. Recently, attempts were made to generalize the visualization methods to multi-dimensional (scalar) volume data that allow for the visual extraction of multi-dimensional features.

Sauber et al. [39] suggested to use multigraphs to generate combinations of multiple scalar fields, where the number of nodes in the graph increase exponentially with the number of dimensions. Similarly, Woodring and Chen [48] allowed for boolean set operations of scalar field visualization. In this context, Akiba and Ma [1] and Blaas et al. [7] were the first who used sophisticated visualization methods and interaction in the multi-dimensional feature space. Akiba and Ma [1] suggested a tri-space visualization that couples parallel coordinates in feature space with volume rendering in object space in addition to one-dimensional plots over time. Blaas et al. [7] use scatter plots in feature space, where the multi-dimensional data is projected into arbitrary planes.

Ivanovska et al. [20] use operations in multi-dimensional feature space, where the feature space is three-dimensional and represented the CIE L*a*b* color space. They couple automatic clustering algorithms based on spatial subdivision of the multi-dimensional feature space with interactions to split and merge clusters to define features of interest. Their idea is similar to ours, but the clustering approaches they applied were the standard ones.

Other approaches are based on statistics rather than interactive visual feature extraction: Jänicke et al. [22] use statistical measurements to detect regions of a certain behavior in multi-dimensional volume data, while Oeltze et al. [32] use correlation and principal component analysis to visualize medical perfusion data.

3.2 Multi-dimensional Non-spatial Data Clustering and Visualization

A traditional way of displaying non-spatial data is the use of scatter diagrams, which do not extend well to more than three dimensions. To display multi-dimensional points in a visual space, non-orthogonal coordinate systems such as parallel coordinates [19, 44] can be used. Parallel vertical lines represent the different dimensions or attributes, and individual records are represented by a polygonal line connecting points on the parallel coordinate lines. A large community is using parallel coordinates on a regular basis. Similarly, Andrews' curves [3] plot each d-dimensional point as a curved line using the function $f(t) = \frac{x_1}{\sqrt{2}} + x_2 \sin(t) + x_3 \cos(t) + \ldots$ where the d-dimensional point is given by x=$(x_1, \ldots, x_d)$. The function is usually plotted in the interval $[-\pi, \pi]$. The striking advantage of these visualizations is that they can represent many dimensions and one can easily understand the relation between the dimensions or attributes. Several extensions have been proposed to alleviate the inherent clutter problem for a large number of records. However, the distribution of clusters in the multi-dimensional space is not visually supported.

The Radviz (Radial visualization) approach [4] is similar in spirit to the parallel coordinates. For d dimensions, the d lines emanate radially from the center of a circle and terminate at the perimeter of the circle at special end points. Similarly, the $2D$ star coordinates [23] arrange the coordinate axes on a circle placed on a two-dimensional plane with axes having their origin at the center of the circle and an arrangement exposing equal angles between adjacent axes. However, a data point x=$(x_1, \ldots, x_d)$ of a d-dimensional data set is mapped to a point using two-dimensional Cartesian coordinates and plotted as a single point. The Viz3D approach [2] extended the Radviz approach by projecting the d-dimensional data into a 3D display space, and Shaik and Yeasin [41] extended the 2D star coordinates to 3D star coordinates.

Wegman and Luo [45] proposed a method for visualizing density for two- and three-dimensional non-spatial data sets. This density function $f(x)$ can be displayed by a series of contours in a plane, where contours of level λ bound the subset $S_\lambda = \{x \in \mathbb{R}^2 : f(x) = \lambda f_{\max}\}$ where $0 < \lambda < 1$ and $f_{\max} = \sup_x f(x)$. A series of contours is represented in a three-dimensional display by depicting each contour of level λ on the plane $z = \lambda f_{\max}$, which leads to a density surface. For the three-dimensional data sets, the respective density surface is in a four-dimensional space such that the direct extension of this rendering method cannot be implemented. The authors proposed to represent series of density surfaces $S_\lambda = \{x \in \mathbb{R}^3 : \ f(x) = \lambda f_{\max}\}$, for distinct levels λ, where again $f_{\max} = \sup_x f(x)$. For high-dimensional data, Scott and Sain [40] proposed an extension that they applied to data sets up to dimensionality of six. The multivariate density function is represented in a three-dimensional space by slices, where several variables are fixed.

In statistics, an attraction region of modes of multivariate density functions is often regarded as a cluster. There are several approaches that report how to identify clusters in large data sets based on computing density. Hinneburg and Keim introduced the DENCLUE approach [17], where high density clusters are identified by determining density attraction. Hinneburg et al. further introduced the HD-Eye system [16] that uses visualization to find the best contracting projection into a one- or two-dimensional space. The data is divided based on a sequence of the best projections determined by the high density clusters. The advantage of this method is that it does not divide regions of high density.

3.3 Unstructured Point-based Data Visualization

Many efficient high-quality visualization methods exist for gridded volume data, where the grid may exhibit a uniform structure or may be formed by connecting arbitrarily distributed vertices like in unstructured tetrahedral meshes. Thus, a common strategy for visualizing point-based volume data is to "convert" them into gridded volume data. Such a conversion can be performed by interpolating values over a uniform structured grid using scattered data interpolation techniques or by connecting the points to form a mesh using tetrahedrization techniques.

Scattered data interpolation has been a research topic for several decades. A large number of scattered data interpolation techniques are available today. Lohda and Franke [25] have broadly classified them in the following way: polynomial-based approaches or piecewise continuous polynomial parametric solutions; algebraic solutions; radial basis function methods; Shepard's methods and its variants; and subdivision-based approaches.

The generation of tetrahedral meshes from point-based volume data also has a long tradition. Du and Wang [14] give an overview over various approaches. Widely accepted are the results given by Delaunay tetrahedrization [13], whose implementation is also included into the Computational Geometry Algorithms Library (CGAL) [10]. More recent approaches try to improve existing Delaunay tetrahedrization algorithms with respect to robustness, quality, and efficiency. Robustness against numerical errors during Delaunay insertion[33] or for boundary recovery [38] is desired. Quality criteria with respect to some design goals are often ensured by post-processing steps [28]. The incremental insertion [15, 8] method is one of the most efficient implementations. Still, computational costs are high. Co et al. [11] presented an approach for isosurface extraction from point-based volume data that uses local Delaunay triangulation, which keeps the number of points for each each Delaunay triangulation step low and thus improves the overall performance.

In the astrophysics community, visualization of slices through the volume, isosurface extraction, direct volume rendering techniques, and particle rendering as color-mapped points are most commonly used [31, 42]. A tool that provides such functionality (except for isosurfaces) is the freely available visualization tool SPLASH [34]. The direct volume rendering is executed by a ray casting approach, where integration along the rays is performed using a kernel-based interpolation with the SPH kernel function $W(x)$. The high adaptivity of the SPH data forces one to use many rays and small step sizes along the rays to not lose details in densely populated regions, which makes this purely software-based direct volume rendering approach slow. Rotation, zooming, and similar desired features cannot be achieved at interactive rates. The visualization tool ParaView Meshless [34] handles SPH data similarly. Navratil et al. [31] apply an inverse-distance-based interpolation for resampling the data to a regular grid prior to isosurface extraction.

We are going to make use of the approach by Rosenthal and Linsen [35]. They developed a direct visualization technique that operates on unstructured point-based volume data to extract isosurfaces. The method is based on spatial domain partitioning using a kd-tree and an indexing scheme for efficient neighbor search. Rosenthal and Linsen showed that this direct isosurface extraction significantly outperformed the local triangulation approach by Co et al. [11].

4 Feature-space Operations

Since we are dealing with large amounts of particles for which we compute a number of properties, our feature space is typically multi-dimensional (in the range of up to 10 or maybe 20 dimensions) and our data consist of many d-dimensional points. Multi-dimensional features are given in the form of particles that have similar properties. In feature space this correlates to clusters of points. We provide means to automatically detect clusters in the multi-dimensional feature space. In order to visualize the clusters, they need to be projected into a three-dimensional space, for which we are using 3D star coordinates.

The projection and the star coordinate layout are optimized to minimize the overlap of projected clusters. Using star coordinates, the clusters are related to the input dimensions such that an interactive exploration system provides mechanisms to interactively explore clusters and select the clusters of interest. In the following, we will describe these feature space operations in more detail.

4.1 Automatic Multi-dimensional Cluster Detection

Given the multi-dimensional feature space with a large number of d-dimensional points lying in that feature space, each point corresponds to one sample of the volumetric data field and each dimension represents one data channel (typically one scalar value) stored at that sample. Samples are the positions of the particles, where, in a pre-processing step, we add the contributions of all particles to that sample using Equation 1 to obtain the actual function values at the positions. In order to understand the distribution of the points in feature space, we propose to compute a density function and to determine the number of clusters as well as the high density region of each cluster.

Given a multivariate density function $f(x)$ in d dimensions, modes of $f(x)$ are positions where $f(x)$ has local maxima. Thus, a mode of a given distribution is more dense than its surrounding area. We want to find the attraction regions of modes. To do so, we choose various values for constants λ $(0 < \lambda < \sup_x f(x))$ and consider regions of the particle space where values of $f(x)$ are greater than or equal to λ. The λ-level set of the density function $f(x)$ denotes a set $S(f,\lambda) = \{x \in \mathbb{R}^d : \ f(x) \geq \lambda\}$. The set $S(f,\lambda)$ consists of a number q of connected components $S_i(f,\lambda)$ that are pairwise disjoint. The subsets $S_i(f,\lambda)$ are called λ-density clusters (λ-clusters for short). A cluster can contain one or more modes of the respective density function.

Let the domain of the data set be given in the form of a d-dimensional hypercube, i. e., a d-dimensional bounding box. To derive the density function, we spatially subdivide the domain of the data set into cells of equal shape and size. Thus, the spatial subdivision provides a binning into d-dimensional cells. For each cell we count the number of points lying inside. The multivariate density function $f(x)$ is given by the number of points per cell divided by the cell's area and the overall number of data points. As the area is equal for all cells, the density of each cell is proportional to the number of data points lying inside the cell. The cell should be small enough such that local changes of the density function can be detected but also large enough to contain a large number of points such that averaging among points is effective. Because of the curse of dimensionality, there will be many empty cells. We do not need to store empty cells such that the amount of cells we are storing and dealing with is (significantly) smaller than the number of the d-dimensional points.

The λ-clusters can be computed by detecting regions of connected cells with densities larger than λ. As we identify density with point counts, the densities are integer values. Hence, we start by computing density clusters for $\lambda = 1$. Subsequently, we process each detected λ-cluster individually by iteratively removing those cells with minimum density, where the minimum density increases in steps of 1. If this process causes a cluster to fall into two subclusters, the subclusters represent higher-density clusters within the original cluster. If a cluster does not fall into subclusters during the process, it is a mode cluster. This process generates a hierarchical structure, which is summarized by the high density cluster tree (short: cluster tree). The root of the cluster tree represents all points. Figure 1(a) shows a cluster tree with 4 mode clusters represented by the tree's leaves. Cluster tree visualization provides a method to understand the distribution of data by displaying the attraction regions of modes of the multivariate density function. Each cluster contains at least one mode.

4.2 Projecting Clusters into 3D Star Coordinates

Having computed the d-dimensional high density clusters, we need to project them into a three-dimensional space for visualization purposes. In order to visualize the high density clusters in a way that allows clusters to be correlated with the d dimensions, we need to use a coordinate system that incorporates all d dimensions. Such a coordinate system can be obtained by using star coordinates.

When projecting the d-dimensional high density clusters into a three-dimensional star coordinate representation, clusters should remain clusters. Thus, points that are close to each other in the d-dimensional feature space should not be further apart after projection into the three-dimensional space.

Let O be the origin of the $3D$ star coordinate system and $(a_1,\ldots,a_d)$ be a sequence of d three-dimensional vectors representing the axes. The mapping of a d-dimensional data point $x = (x_1,\ldots,x_d)$ to a three-dimensional data point $\Pi(x)$ is determined by the average sum of vectors a_k of the $3D$ star coordinate system multiplied with its attributes x_k for $k = 1,\ldots,d$, i.e.

$$\Pi(x) = O + \frac{1}{d}\sum_{k=1}^{d} x_k a_k. \tag{2}$$

Since it can be shown that

$$||\Pi(x) - \Pi(y)||_1 \leq ||x-y||_1$$

for any d-dimensional points x and y, the distance of the images of two d-dimensional points is lower than or equal to the distance of

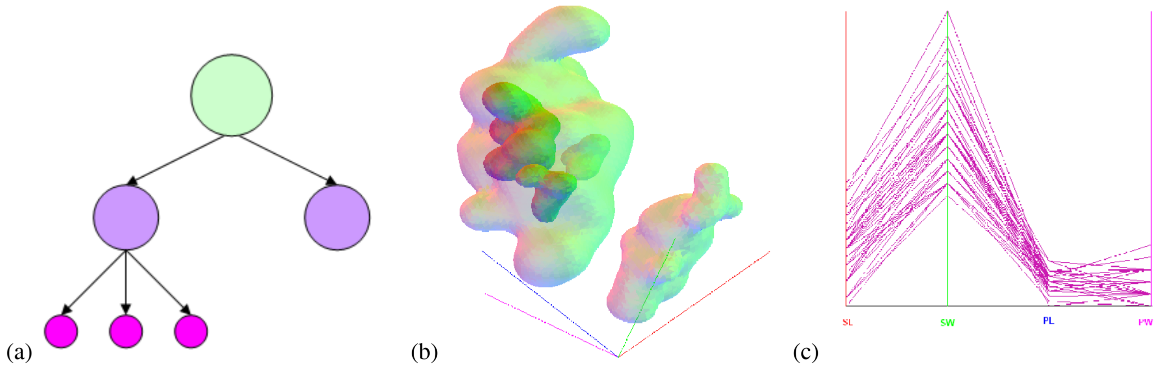

Fig. 1. (a) Cluster tree of density visualization with four modes shown as leaves of the tree. (b,c) Visualization of four-dimensional data set: (b) Nested density cluster visualization based on cluster tree using 3D star coordinates. (c) Right-most cluster in (b) is selected and its homogeneity is evaluated using parallel coordinates. In both pictures the relation between the selected cluster with the dimension can be observed.

the points with respect to the L_1-norm. Therefore, two points in the multi-dimensional space are projected to $3D$ star coordinates preserving the similarity properties of clusters (at least with respect to the L_1-norm). In other words, the mapping of d-dimensional data to a three-dimensional space determined by Equation (2) does not break clusters.

The second property that our projection from multi-dimensional feature space into three-dimensional star coordinate systems should fulfill is that separate clusters should not be projected into the same region. The projection into star coordinates may cause severe cluttering of clusters when not carefully choosing the axes $(a_1,\dots,a_d)$. To alleviate the problem of overlapping clusters we introduce a method which chooses a "good" coordinate system. Assume that a hierarchy of high density clusters have q mode clusters, which do not contain any higher level densities. Let m_i be the barycenter of the points within the ith cluster, $i=1,\dots,q$. We want to choose a projection that maintains best the distances between clusters. Let $\{v_1,v_2,v_3\}$ be an orthonormal basis of the candidate three-dimensional space of projections. The desired choice of a $3D$ star coordinate layout is to maximize the distance of the q projected barycenters $V^T m_i$ with $V=[v_1,v_2,v_3]^T$, i.e. to maximize the objective function

$$\sum_{i<j} ||V^T m_i - V^T m_j||^2 = \text{trace}(V^T SV)$$

with

$$S = \sum_{i<j} (m_i - m_j)(m_i - m_j)^T \ .$$

Thus, the three vectors v_1, v_2, v_3 are the three unit eigenvectors corresponding to the three largest eigenvalues of matrix S. This step is a principal component analysis (PCA) applied to the barycenters of the clusters. As a result, we choose the d three-dimensional axes of the $3D$ star coordinate system as $a_i = (v_{1i}, v_{2i}, v_{3i})$, $i=1,\dots,d$.

Figure 2(a) shows the distribution of points after projection for an example using synthetic data with $d=20$. The automatic cluster computation extracted eight high density clusters indicated by the colors. The remaining points (colored bluish) represent noise. The eight clusters can clearly be identified as clusters, as they did not break apart or spread during projection. Moreover, clusters can clearly be distinguished not only by color but also by position, as they remained clearly separated during projection and do not overlap.

After having computed the projected clusters, we can display them using star coordinates by rendering a point primitive for each projected data point as shown in Figure 2(a). A less cluttered and more beautiful display would render the boundary of the cluster.

4.3 Nested Cluster Visualization

Considering the cluster that is described by the set of points $\{p_i = (x_i, y_i, z_i) : i = 1,\dots,m\}$ after being projected into the 3D space. In order to compute the boundary of this group of points,we need to have a continuous representation of the group. Therefore, we consider the function $f_h(p) = \sum_{i=1}^{m} K(\frac{p-p_i}{h})$, $p \in \mathbb{R}^3$, where K is a kernel function and h is the bandwidth. Then, we render the boundary set of the points by using the marching cubes algorithm [26] to extract the boundary surface of the set $S(h,c) = \{p \in \mathbb{R}^3 : \ f_h(p) \geq c\}$, where c is an isovalue. We choose parameter h and c to guarantee that $S(h,c)$ is connected and has a volume of minimum extension. The kernel function should be sufficiently smooth and have a small compact support. For example, we can choose $K(p) = (1-||p||^2)^2$ for $||p|| \leq 1$ and $K(p) = 0$ otherwise and the bandwidth h to be equal to the longest length of the minimum spanning tree of these m points. In Figure 2(b) we show the visualization of the clusters in Figure 2(a) by rendering such boundary surfaces, where it can be shown that for the chosen kernel isovalue $c = \frac{9}{16}$ is appropriate.

In order to visualize all clusters of the cluster tree, we render the surfaces in a semi-transparent fashion. The resulting visualization shows sequences of nested surfaces, where the inner surfaces represent higher density levels. Figure 1(b) shows the nested density cluster visualization with respect to the cluster tree in Figure 1(a). The cluster tree has been computed by interactively choosing appropriate λ-values.

Generating all clusters and displaying them in star coordinates allows for further analysis of the detected clusters. The simplest interaction method is to select individual clusters by just clicking at the boundary surface. When a cluster is selected, intra-cluster variability is visualized using parallel coordinates, see Figure 1(b) and (c).

5 Object-space Operations

After having detected multi-dimensional features of interest in the form of density clusters in feature space, we want to extract their boundary in volumetric object space. To do so, we need to map inside-outside properties to the particles in object space. Given the properties at the particle positions, we define a continuous volumetric scalar field, the zero-level set of which represents the features boundary surface. Afterwards, we extract the boundary surface in the form of a point-cloud representation using isosurface extraction. The surface in point cloud representation is being rendered using point-based rendering techniques.

5.1 Deriving a Continuous Inside-outside Scalar Field

The membership to a selected cluster imposes a binary decision onto the particles. Hence, we can assign value 1 to all particles belonging

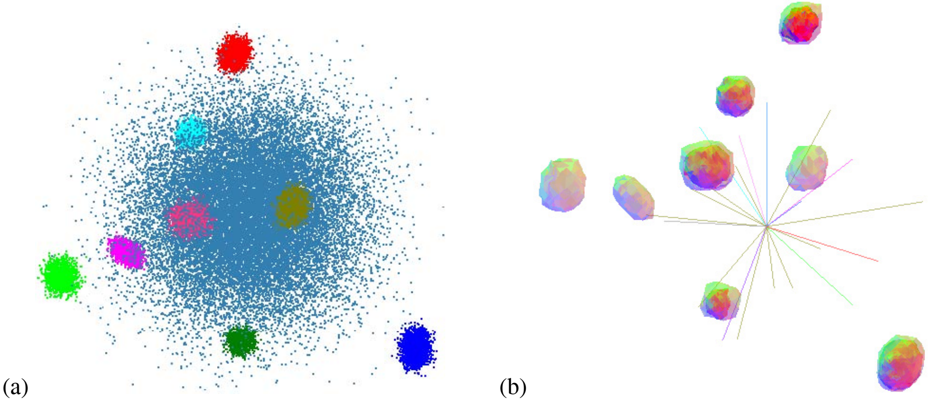

Fig. 2. (a) Visualization of feature space clusters: Extraction of eight clusters in 20-dimensional domain and projection into 3D star coordinate space while maintaining cluster property and avoiding overlap. (b) Density cluster visualization by rendering boundary surfaces.

to the cluster and value -1 to all the others. Now, we would like to extract an isosurface with respect to these assigned values using isovalue 0 following the ideas from Rosenthal and Linsen [35]. Such a direct isosurface extraction from unstructured point-based data requires the derivation of a continuous scalar field encoding inside-outside decisions.

We use the SPH kernel to generate the scalar field. To each particle we apply the SPH kernel according to Equation 1, where the function value f_b assigned to particle b is 1 for particles inside the cluster and -1 otherwise. We neglect masses and densities and, instead, apply a normalization of the weights. This defines a smoothly varying function value, where the smoothing lengths h taken from the SPH simulation ensures that the sign of the values at the particle positions themselves do not change. We compute the values at the particle positions and proceed with the direct isosurface extraction method described below.

5.2 Direct Isosurface Extraction

Let $f : \mathbb{R}^3 \to \mathbb{R}$ be a trivariate scalar function, whose values are given for a large, finite set of samples $(\mathbf{x}_i, f(\mathbf{x}_i))$. The sample positions $\mathbf{x}_i \in \mathbb{R}^3$ are not arranged in a structured way, nor is any connectivity or neighborhood information known for the sample locations. Our goal is to extract an isosurface $f(\mathbf{x}) = v_{iso}$ with respect to any real isovalue v_{iso} out of the range of function f.

Isosurface extraction over discrete structures is typically performed in two steps. First, a number of points $\mathbf{p}_k \in \mathbb{R}^3$ on the isosurface are computed, i. e. $f(\mathbf{p}_k) = v_{iso}$. For this purpose, an interpolation scheme is applied to the function values $f(\mathbf{x}_i)$ in order to locally reconstruct a continuous scalar field between the given discrete sample positions $\mathbf{x}_i$. We refer to the points $\mathbf{p}_k$ on the isosurface as *isopoints*.

In a second step, some kind of neighborhood information for the isopoints is generated, which is used to render the isosurface. When the samples are arranged on a structured grid, the neighborhood information can be retrieved from the structure of the grid. Typically, polygonal meshes are generated and rendered.

Our idea for isopoint computation from scattered volume data is based on linear interpolation between pairs of samples with close positions $\mathbf{x}_i$ and $\mathbf{x}_j$. The inspiration for this approach is given by isopoint computation using the marching tetrahedra algorithm after partitioning the domain via Delaunay tetrahedrization. The analogy is illustrated in Figure 3 for the 2D case.

We want to adopt this isopoint computation method for our purposes while avoiding the computation of the Delaunay tetrahedrization. Thus, we need to estimate the natural neighbors for each sample position $\mathbf{x}_i$. Since the exact natural neighbors can only be determined via the expensive Voronoi diagram computation, we approximate its result using a spatial decomposition [35]. Note that replacing the natural neighbors by the nearest neighbors would fail in case of varying density distributions of the samples. The kd-tree data structure [6] is known to be a data structure with well-balanced trade-off between flexibility and efficiency. Operations on a kd-tree are fast, yet it is robust against varying density distribution and clustering of sample positions.

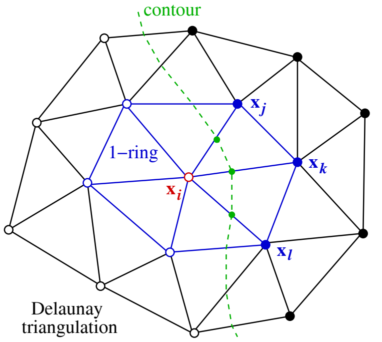

Fig. 3. Isopoint computation for scattered data via Delaunay triangulation and linear interpolation along Delaunay edges. For sample position $\mathbf{x}_i$ the incident edges to sample locations $\mathbf{x}_j$, $\mathbf{x}_k$, and $\mathbf{x}_l$ intersect the contour.

To perform a fast exploration of the kd-tree, we introduce an indexing scheme that, beside saving storage space, allows us to determine neighbors using bitwise operations on the index. We determine a small number of potential candidates for our neighbors based on the neighborhoods in the kd-tree and reject some of them using an angle criterion. Exploiting this neighborhood information, we select those pairs of neighbors that are separated by the isosurface. We compute the isopoint by linearly interpolating between the two samples. More details on this fast direct isosurface extraction method is given by Rosenthal and Linsen [35].

The result of the isopoint computation is an isosurface in point-cloud representation, i.e., it is represented by a set of points lying on the surface, see Figure 4(b). In a final step, we render the isosurface using point-based rendering techniques, see Figure 4(c).

5.3 Splat-based Raytracing

For point-based rendering of the extracted surfaces we apply a raytracing approach to splats, which are computed from the surface points. We follow the approach by Linsen et al. [24].

Let P be a point cloud representing a surface and consisting of n points $\mathbf{p}_1, \dots, \mathbf{p}_n \in \mathbb{R}^3$. We generate m splats $S_1, \dots, S_m$ that cover the entire surface represented by point cloud P. For each of these splats we are computing its radius $r_i \in \mathbb{R}$, $i = 1, \dots, m$, and a normal field $\mathbf{n}_i(u, v)$, $i = 1, \dots, m$, where $(u, v) \in [-1, 1] \times [-1, 1]$ with $u^2 + v^2 \leq 1$ describes a local parameterization of the splat.

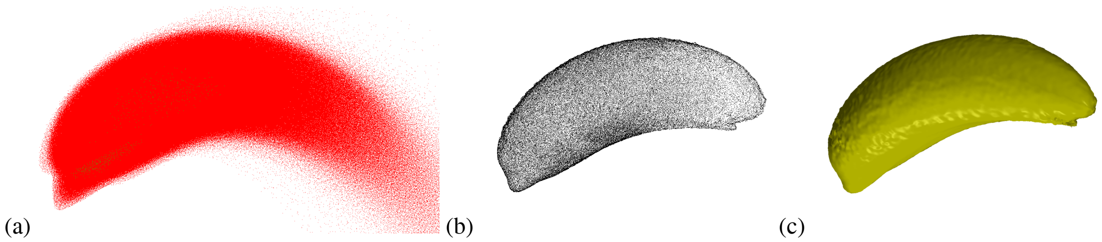

Fig. 4. (a) Particle distribution of SPH simulation representing point-based volume data. (b) Isopoints extracted from the point-based volume data form a point-based surface representation. (c) Point-based surface rendering.

In order to generate a smooth-looking visualization of a surface with a piece-wise constant representation, we need to smoothly (e. g. linearly) interpolate the normals over the surface before locally applying the light and shading model. Since we do not have connectivity information for our splats, we cannot interpolate between the normals of neighboring splats. Instead, we need to generate a linearly changing normal field $\mathbf{n}_i(u,v)$ within each splat, cf. [9].

The raytracing algorithm is carried out in a traditional way [5, 47, 43] using a recursive procedure and is based on ray-splat intersection. The splat-ray intersection process is sped up using spatial partitioning, for example by octrees, and a sequence of nested checks with no false negatives and few false positives. The surface normal $\mathbf{n}$ at an intersection point is computed by a weighted average of the normals $\mathbf{n}_i$ at the respective locally parameterized positions (u_i, v_i) of the splats' intersection points. This weighted averaging leads to continuously varying normals on the surface.

6 Results and Discussion

The main data for this project come from simulations of the disruption of white dwarf stars by the tidal forces of a black hole. The disruption of the white dwarf *outside* the event horizon of the black hole is only possible for black holes of moderate masses ($< 10^5$ M$_\odot$), otherwise the white dwarf is swallowed as a whole before disruption. It is exactly this property that offers the unique possibility to probe the existence of the much-debated, but yet undiscovered class of intermediate-mass black holes ($100 < \mathrm{M} < 10^5$ M$_\odot$) by predicting the signatures (gravitational waves, electromagnetic display in the optical and X-ray waveband and nucleosynthetic yields) that this type of tidal disruption would have. These predictions will soon become testable by new experiments such as LISA [12] and LSST [21]. The corresponding calculations are typical examples of astrophysical multi-scale (the time scales of different physical processes vary by more than 10 orders of magnitude) and multi-physics (gravity, hydrodynamics, nuclear reactions) simulations.

The simulations are typically performed with 500,000 to 5,000,000 SPH particles. For each particle a nuclear reaction network consisting of seven (quasi-)nuclei [18] is implicitly integrated along the hydrodynamical flow to correctly account for the feedback of the nuclear energy generation onto the gas dynamics[37]. Therefore, for each particle the information of the mass fraction of seven chemical elements (or groups of elements) needs to stored: Helium, Carbon, Oxygen, Neon, Magnesium, Silicon, and Iron.

For feature space exploration we consider the seven-dimensional feature space with about 500,000 points and determine the density clusters. We divide each dimension of data space into $N = 10$ steps. We applied the automatic clustering to time slice $t = 150.93s$ after simulation start when the disrupted white dwarf is stretched and first material is being accreted onto the black hole. The cluster analysis using 3D star coordinates reveals mainly three cluster modes, see Figure 5 (left). The clusters are projected into optimized 3D star coordinates maintaining their structure and avoiding overlap. From the total of $502,218$ particles, we obtain a mode cluster with 105,413 particles and another cluster with $364,930$ particles, which includes two mode clusters with $260,604$ and $34,314$ particles, respectively. Particles that do not belong to any cluster, i.e., isolated particles, are regarded as noise. The clusters correspond to Helium (red), Carbon (green), and Silicon (black), which are the dominant elements of the disrupted remnant.

The initial 0.2 M$_\odot$ white dwarf was made entirely of Helium and so the remaining Helium after the disruption (49.3%) is mainly the unburnt remaining nuclear fuel from the regions that were never heated strongly enough to undergo nuclear burning. This material forms an outer mantle around the processed core of the debris, a moderate Helium fraction (~ 0.2) from incomplete nuclear burning is still present in the debris core. Inside this Helium shell there is a very thin shell made of material with a Carbon mass fraction in excess of 0.3, the total carbon mass being 4.1% of the initial white dwarf mass. The core of the debris which was always at the highest densities and which was heated to highest temperatures ($T > 10^9$ K) is dominated by freshly produced Silicon (mass fraction ≈ 0.8, 46.4% of the initial stellar mass). These physical results are consistent with and supported by the mode analysis in the star coordinates: the dominant cluster is highly dependent on Helium, the second most important mode cluster is highly dependent on Silicon, and the third mode cluster (Carbon) lies between them. The last two clusters are more closely related, which is indicated by the lower density cluster that includes them. The physical reason for this observation is that both of the latter elements have been produced in the regions of the star that have undergone nuclear processing, while the Helium is mainly from cool and nuclearly unprocessed surface layers of the white dwarf.

Figure 5 (right) shows an earlier time slice $t = 13.7s$, which is at the end of the nuclear burning phase. During this stage a slightly larger Oxygen fraction existed which was subsequently consumed in the nuclear reactions leading to the build-up of the Silicon core.

Having explored the distribution of the clusters in feature space, one is interested in seeing which spatial areas correlate with the clusters. Figures 6(a)-(b) and Figures 6(e)-(f) show the particle distribution in object space for time slices 150.93s and 13.7s, respectively. The colors encode the three mode clusters we have detected: The green particles belong to the "Silicon cluster", the blue particles belong to the "Helium cluster", and the red particles belong to the intermediate cluster. The distribution gives the impression that Silicon is highly condensed in the core and is surrounded by a highly condensed Helium region. This assumption gets affirmed when looking into the boundary surfaces that are extracted from the data. Figures 6(c) and (g) show the Silicon cluster at time steps 150.93s and 13.7s, respectively, and Figures 6(d) and (h) show the Helium cluster at time steps 150.93s and 13.7s, respectively. The renderings are obtained using the splat-based raytracing approach. We observe that the Helium cluster boundary surface is not closed, which causes problems for a splat-based rendering, as boundaries exhibit rendering artifacts.

A large set of calculations shows the nuclear energy release triggered by the tidal compression can lead to thermonuclear explosions of white dwarfs of the full mass range between 0.2 M$_\odot$ and the Chandrasekhar mass, provided that the white dwarf penetrates deep enough into the tidal radius of the black hole. Such thermonuclear explosions have a very different signature from standard type Ia Supernovae. If these events do exist, future supernova surveys such as LSST should detect them. An underluminous thermonuclear explosion followed by an X-ray flare of a few months length would be a compelling testimony for the existence of intermediate mass black holes.

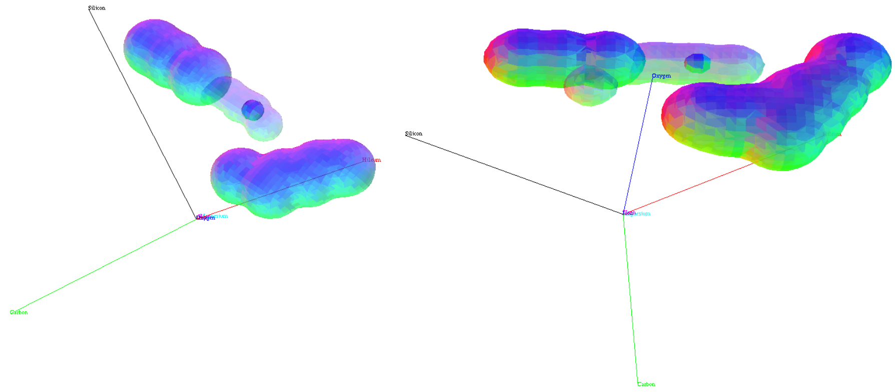

Fig. 5. Seven-dimensional feature space exploration of a multi-field smoothed particle hydrodynamics simulations using a visualization of nested clusters in 3D optimized star coordinates. *Left:* time slice $t = 150.93s$. *Right:* time slice $t = 13.7s$.

In terms of computation times, the clustering step is the most intense one. For the given examples, it took about $330s$. The subsequent projection and surface extraction in feature space took about $1s$. The rendering is interactive. The generation of a smooth inside-outside field was not optimized and took about $16s$, whereas the actual isosurface extraction took only $3.5s$. The splat-based raytracing took about $50s$ when generating a high-resolution output of 1200×1200 pixels. The memory consumption is linear both in the number of particles and in the number of dimensions. This is achieved by performing one-dimensional binning in feature space and removing empty cells before continuing with binning in the subsequent dimension.

7 Conclusion

We have presented an approach to visualize multi-field smoothed particle hydrodynamics data. It consists of feature space and corresponding object space operations. In feature space, we compute a density function, which we use for automatic detection of hierarchies of high density clusters. These clusters are projected into a 3D star coordinate space. The projection is defined such that it optimizes cluster distribution in terms of overlap, compactness, and shape. A nested level set visualization for the high density area with respect to different density levels allows for an interactive exploration of the hierarchical clusters and to correlate the clusters to the original dimensions. The detected clusters can also be displayed in volumetric object space. Their boundary surfaces in object space are extracted using direct isosurface extraction from unstructured point-based volume data, where the underlying scalar field is obtained by applying the SPH kernel to the binary inside-outside encoding of particles with respect to the cluster of interest. The extracted surface in the form of a point cloud is rendered using a splat-based raytracing technique.

Acknowledgements

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under project grant LI-1530/6-1.

References

[1] H. Akiba and K.-L. Ma. A tri-space visualization interface for analyzing time-varying multivariate volume data. In *In Proceedings of Eurographics/IEEE VGTC Symposium on Visualization*, pages 115–122, May 2007.

[2] O. Almir and C. Maria. Viz3d: Effective exploratory visualization of large multidimensional data sets. *Computer Graphics and Image Processing, XVII Brazilian Symposium on SIBGRAPI*, pages 340–347, 2004.

[3] D. Andrews. Plots of high-dimensional data. *Biometrics*, 28:125–136, 1972.

[4] M. Ankerst, D. Keim, and H. Kriegel. Circle segments: A technique for visually exploring large multidimensional data sets. *IEEE Visualization Proceedings, Hot topic session, San Francisco, CA*, 1996.

[5] A. Appel. Some techniques for shading mashine rendering of solids. In *Proceedings of the Spring Joint Computer Conference*, pages 37–45, 1968.

[6] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975.

[7] J. Blaas, C. P. Botha, and F. H. Post. Interactive visualization of multifield medical data using linked physical and feature-space views. In *EuroVis*, pages 123–130, 2007.

[8] H. Borouchaki, F. Hecht, E. Saltel, and P. George. Reasonably efficient delaunay based mesh generator in 3 dimensions. In *4th International Meshing Roundtable*, pages 3–14. Sandia National Laboratories, 1995.

[9] M. Botsch, M. Spernat, and L. Kobbelt. Phong splatting. In *Eurographics Symposium on Point-Based Graphics*, pages 25–32, 2004.

[10] Computational geometry algorithms library (CGAL). http://www.cgal.org/.

[11] C. S. Co and K. I. Joy. Isosurface Generation for Large-Scale Scattered Data Visualization. In G. Greiner, J. Hornegger, H. Niemann, and M. Stamminger, editors, *Proceedings of Vision, Modeling, and Visualization 2005*, pages 233–240. Akademische Verlagsgesellschaft Aka GmbH, 2005.

[12] K. Danzmann. LISA Mission Overview. *Advances in Space Research*, 25:1129–1136, 2000.

[13] B. N. Delaunay. Sur la sphere vide. *Bull. Acad. Sci. USSR*, 7:793–800, 1934.

[14] Q. Du and D. Wang. Recent progress in robust and quality delaunay mesh generation. *J. Comput. Appl. Math.*, 195(1):8–23, 2006.

[15] P. L. George, F. Hecht, and E. Saltel. Automatic mesh generator with specified boundary. *Comput. Methods Appl. Mech. Eng.*, 92(3):269–288, 1991.

[16] A. Hinneburg and D. Keim. An efficient approach to clustering in large multimedia databases with noise. *Proc. Int. Conf. Knowledge Discovery and data mining*, pages 58–65, 1998.

[17] A. Hinneburg, D. Keim, and M. Wayryniuk. Hd-eye: Visual mining of high-dimensional data. *IEEE Computer Graphics and Applicatons*, pages 22–31, 1999.

[18] W. R. Hix, A. M. Khokhlov, J. C. Wheeler, and F.-K. Thielemann. The Quasi-Equilibrium-reduced alpha -Network. *Astrophysical Journal*, 503:332–+, Aug. 1998.

[19] A. Inselberg. The phane with parallel coordinates. *Visual Computer*, 1:69–97, 1985.

[20] T. Ivanovska and L. Linsen. A user-friendly tool for semi-automated segmentation and surface extraction from color volume data using geometric feature space operations. In L. Linsen, H. Hagen, and B. Hamann, editors, *Visualization in Medicine and Life Sciences*, pages 153–170. Springer-Verlag, Heidelberg, Germany, 2007.

[21] Z. Ivezic, J. A. Tyson, R. Allsman, J. Andrew, R. Angel, and for the LSST Collaboration. LSST: from Science Drivers to Reference Design and Anticipated Data Products. *ArXiv e-prints*, 805, May 2008.

[22] H. Jänicke, A. Wiebel, G. Scheuermann, and W. Kollmann. Multifield visualization using local statistical complexity. *IEEE Transaction on Vi-*

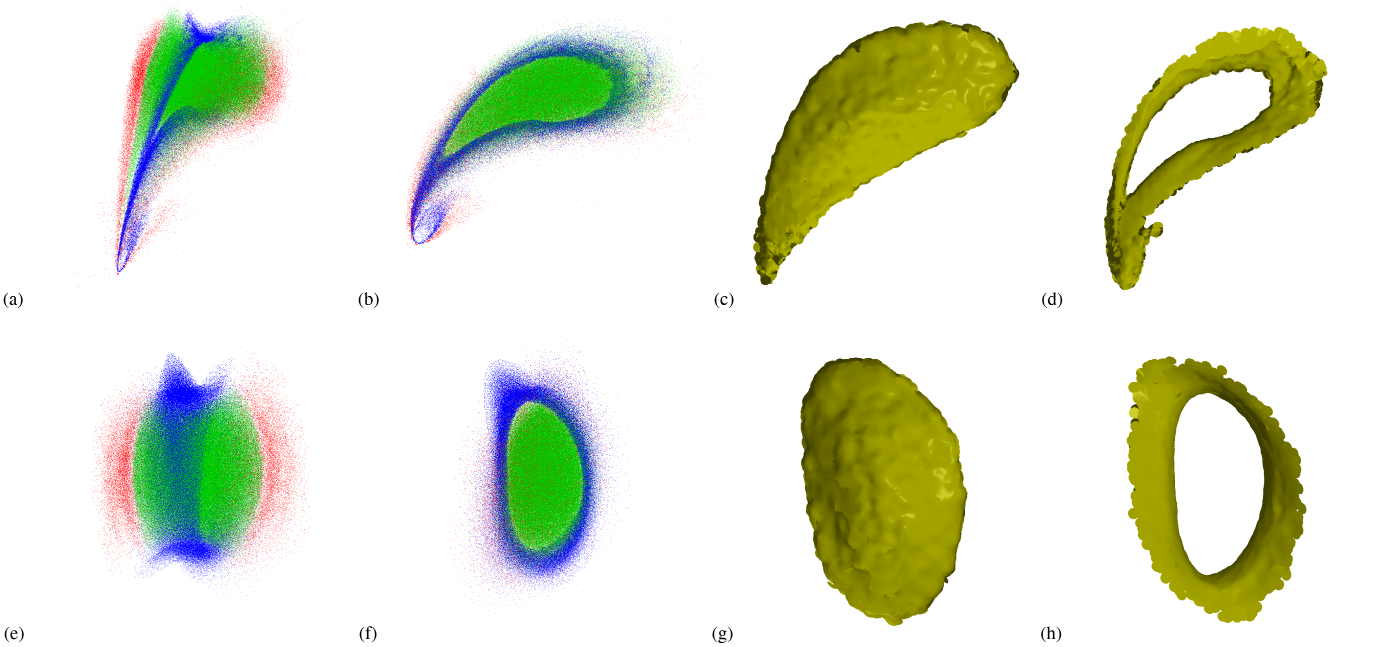

Fig. 6. Object space visualization of cluster distribution and their boundary surfaces for a multi-field smoothed particle hydrodynamics simulations of time slices $t = 150.93s$ (upper row) and $t = 13.7s$ (lower row). *Left and middle left:* Two perspectives of the distribution of the three mode clusters. *Middle right and right:* Extracted boundary surfaces for green and blue clusters, respectively.

sualization and Computer Graphics, 13(6):1384–1391, 2007.

[23] E. Kandogan. Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. *Proc. ACM Int. Conf. Knowledge Discovery and Data Mining*, pages 107–116, 2001.

[24] L. Linsen, K. Müller, and P. Rosenthal. Splat-based ray tracing of point clouds. *Journal of WSCG*, 15(1–3):51–58, 2007.

[25] S. K. Lodha and R. Franke. Scattered data techniques for surfaces. In *Proceedings of Dagstuhl Conference on Scientific Visualization*, pages 182–222. IEEE Computer Society Pres, 1999.

[26] W. Lorensen and H. Cline. Marching cubes: a high resolution 3d surface construction algorithm. *Computer Graphics*, 21:163–169, 1987.

[27] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, 82:1013–1024, 1977.

[28] P. Maur and I. Kolingerová. Post-optimization of delaunay tetrahedrization. In *SCCG '01: Proceedings of the 17th Spring conference on Computer graphics*, page 31, Washington, DC, USA, 2001. IEEE Computer Society.

[29] J. Monaghan. *Monthly Notices of the Royal Astronomical Society*, 181:375, 1977.

[30] J. J. Monaghan. Smoothed particle hydrodynamics. *Reports of Progress in Physics*, 68:1703–1759, Aug. 2005.

[31] P. A. Navrtil, J. L. Johnson, and V. Bromm. Visualization of cosmological particle-based datasets. In *IEEE Visualization*, 2007. to appear.

[32] S. Oeltze, H. Doleisch, H. Hauser, P. Muigg, and B. Preim. Interactive visual analysis of perfusion data. *IEEE Transaction on Visualization and Computer Graphics*, 13(6):1392–1399, 2007.

[33] S. E. Pav and N. J. Walkington. Robust three dimensional delaunay refinement. In *13th International Meshing Roundtable*, pages 145–156. Sandia National Laboratories, SAND 2004-3765C, 2004.

[34] D. Price. SPLASH. http://arxiv.org/abs/0709.0832.

[35] P. Rosenthal and L. Linsen. Direct isosurface extraction from scattered volume data. In B. S. Santos, T. Ertl, and K. I. Joy, editors, *Eurographics / IEEE VGTC Symposium on Visualization - EuroVis 2006*, pages 99–106,367, 2006.

[36] S. Rosswog and D. Price. Magma: a magnetohydrodynamics code for merger applications. *Monthly Notices of the Royal Astronomical Society*, 379:915 – 931, 2007.

[37] S. Rosswog, E. Ramirez-Ruiz, W. R. Hix, and M. Dan. Simulating black hole white dwarf encounters. *Computer Physics Communications*, 179:184–189, July 2008.

[38] N. S. Sapidis and R. Perucchio. Domain delaunay tetrahedrization of arbitrarily shaped curved polyhedra defined in a solid modeling system. In *SMA '91: Proceedings of the first ACM symposium on Solid modeling foundations and CAD/CAM applications*, pages 465–480, New York, NY, USA, 1991. ACM Press.

[39] N. Sauber, H. Theisel, and H.-P. Seidel. Multifield-graphs: An approach to visualizing correlations in multifield scalar data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):917–924, 2006.

[40] D. Scott and S. Sain. *Multidimensional Density Estimation, in Handbook of Statistics, Vol 23: Data Mining and Computational Statistics, Edited by C.R. Rao and E.J. Wegman*. Elsevier, Amsterdam, 2004.

[41] J. Shaik and M. Yeasin. Visualization of high dimensional data using an automated 3d star coordinate system. *International Joint Conference on Neural Networks*, pages 1339–1346, 2006.

[42] R. Walker, P. Kenny, and J. Miao. Visualization of Smoothed Particle Hydrodynamics for Astrophysics. In L. Lever and M. McDerby, editors, *Theory and Practice of Computer Graphics 2005*, pages 133–138, University of Kent, UK, June 2005. Eurographics Association. (Electronic version http://diglib.eg.org).

[43] A. Watt. *3D Computer Graphics*. Pearson - Addison Wesley, 3 edition, 2000.

[44] E. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 21:664–675, 1990.

[45] E. Wegman and Q. Luo. On methods of computer graphics for visualizing densities. *Journal of Computational and Graphics Statistics*, 11:137–162, 2002.

[46] S. C. Whitehouse, M. R. Bate, and J. J. Monaghan. A faster algorithm for smoothed particle hydrodynamics with radiative transfer in the flux-limited diffusion approximation. *Monthly Notices of the Royal Astronomical Society*, 364:1367–1377, Dec. 2005.

[47] T. Whitted. An improve illumination model for shaded display. *Communications of ACM*, 23(6):343–349, 1980.

[48] J. Woodring and H.-W. Shen. Multi-variate, time varying, and comparative visualization with contextual cues. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):909–916, 2006.

[49] S. Yoon, P. Podsiadlowski, and S. Rosswog. Remnant evolution after a carbon-oxygen white dwarf merger. *Monthly Notices of the Royal Astronomical Society, in press*, 2007.