

SmoothIsoPoints: Making PDE-based Surface Extraction from Point-based Volume Data Fast

Paul Rosenthal¹, Vladimir Molchanov² and Lars Linsen^{2,3}

¹University of Rostock, Rostock, Germany

²Jacobs University, Bremen, Germany

³Westfälische Wilhelms-Universität Münster, Münster, Germany

Keywords: Surface Extraction, Isosurfaces, Level Sets, Unstructured Point-based Volume Data.

Abstract: PDE-based methods like level-set methods are a valuable and well-established approach in visualization to extract surfaces from volume data. We propose a novel method for the efficient computation of a signed-distance function to a surface in point-cloud representation and embed this method into a framework for PDE-based surface extraction from point-based volume data. This enables us to develop a fast level-set approach for extracting smooth isosurfaces from data with highly varying point density. The level-set method operates just locally in a narrow band around the zero-level set. It relies on the explicit representation of the zero-level set and the fast generation of a signed-distance function to it. A level-set step is executed in the narrow band utilizing the properties and derivatives of the signed-distance function. The zero-level set is extracted after each level-set step using direct isosurface extraction from point-based volume data. In contrast to existing methods for unstructured data which operate on implicit representations, our approach can use any starting surface for the level-set approach. Since for most applications a rough estimate of the desired surface can be obtained quickly, the overall level-set process can be shortened significantly. Additionally, we avoid the computational overhead and numerical difficulties of PDE-based reinitialization. Still, our approach achieves equivalent quality, flexibility, and robustness as existing methods for point-based volume data.

1 INTRODUCTION

Many modern simulation methods generate unstructured point-based volume data, i. e., scalar fields where the data points may have an arbitrary distribution in a 3D space and do not exhibit any connectivity. A major group of such data stems from Lagrangian numerical simulations of natural phenomena such as fluid dynamics. They allow for the reproduction of complex natural phenomena by not only simulating the evolution of data at the sample points but also simulating the flow of the sample points under respective forces. Hence, data points move over time, change their neighborhoods, and are distributed with a highly varying density. Such simulations are typically carried out with millions of particles.

Level-set methods have a large variety of applications. In scientific visualization, they are used to extract boundary surfaces of features from volume data. Typically, the algorithms operate on rectilinear cells and a given initial level-set function is modified to explicitly or implicitly minimize a given energy functional.

An approach generalizing the basic idea of level sets to work directly on unstructured point-based volume data was presented in (Rosenthal and Linsen,

2008b). They do not resample the data over a structured grid using scattered data interpolation, which inevitably introduces resampling errors, which can grow enormously for data sets with highly varying point density, unless the sampling points are chosen very densely which lets the data size explode. unstructured points avoids this source of error.

The level-set approach models the evolution of a surface by applying forces in normal direction. To allow for easy change of the topology of the surface, it is implicitly represented as the zero-level set of a level-set function. The evolution is carried out by transforming the level-set function using an iterative numerical integration scheme, which implicitly transforms the surface. The approach typically induces complex calculations and the evaluation of partial differential equations at each sample location and each iteration step. When dealing with large data sets and small time-integration steps, this can lead to enormous computation times until convergence of the level-set process. The calculations for unstructured point-based data are even more complex than those in the gridded case which was the major drawback of the method by Rosenthal and Linsen. Later, the approach was modified to utilize a narrow-band technique and significantly speed up the computations (Rosenthal

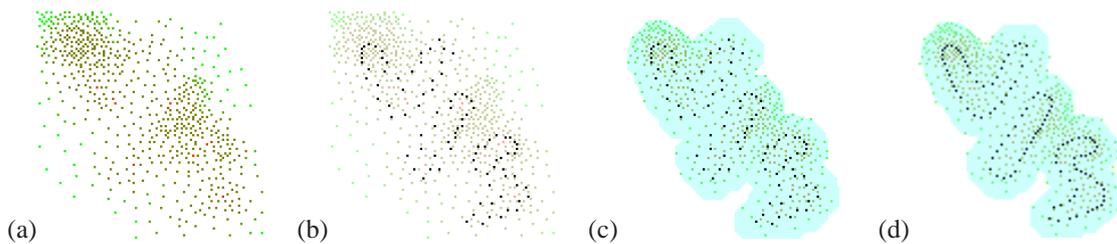


Figure 1: Illustration of our proposed level-set method for extracting smooth isosurfaces from an artificial 2D data set. a) As input data we assume a volumetric scalar data set with unstructured sample positions and no connectivity. Colors indicate function values at the data points. b) Directly extracting isosurfaces from the data may produce bumpy surfaces in regions with highly varying point density. Black dots indicate the isosurface in point-cloud representation. Nevertheless, the isosurface can be used by our method as a good initial guess for the PDE-based extraction of a smooth isosurface. In fact, every closed surface (in point-cloud representation) can be used as initial zero-level set. c) In each step of the level-set iteration a narrow band around the current zero-level set is generated and all data points within the narrow band are reinitialized with an efficient approximation of the signed-distance function to the zero-level set. Colors indicate the signed distance to the zero-level set. d) After reaching a steady state, the zero-level set represents the smooth isosurface.

et al., 2010). However, due to the restriction to start with a predefined signed-distance function as level-set function and the need for a time-consuming numerical reinitialization, the approach is still not able to process data sets with several million data points in a reasonable amount of time.

We propose a level-set method which makes the extraction of smooth isosurfaces from unstructured point-based data sets with several million sample points much faster:

- Similar to previous work, we achieve high accuracy and flexibility by operating only on the unstructured data points. We do not require any scalar field at positions other than the sample points.
- We replace the time-consuming numerical reinitialization of previous work by a level-set function computation on basis of the approximation of a signed-distance function to the respective zero-level set. This procedure increases the accuracy of the whole process, since every level-set step is carried out with a level-set function very close to a signed-distance function. Furthermore, the computation of the approximation is so fast, that the overall computation time of the level-set process drops even if this computation is done in each level-set step.
- We maintain the efficiency of restricting computations to a narrow band around the current zero-level set. In each level-set step, only a small portion of the data points around the current zero-level set is marked to be active.
- We update the narrow band efficiently. The bottleneck of narrow-band methods is often the update of the narrow band, which includes the computation of the level-set function values at

those points that are newly inserted into the narrow band. These points typically have no defined level-set function value. Using our efficient signed-distance function computation for establishing the level-set function, we can quickly determine the required function values.

- We permit any point cloud as initial surface for the level-set process. To initialize the whole level-set process, only a point cloud with surface normals, acting as initial zero-level set, has to be given. The level-set function is computed as approximative signed-distance function to the zero-level set allowing for the computation of the first level-set step. In particular, the initial surface can be an isosurface in point-cloud representation as computed in (Rosenthal and Linsen, 2006). The normals at the points are computed as gradients of the given scalar field.

The main approach of our local level-set method and the structure of the visualization pipeline are explained in Section 3 and depicted in Figure 1: Given a scalar field in the form of point-based volume data (a), one can extract an isosurface in point-cloud representation. When dealing with highly varying point densities, the isosurface may exhibit noise (b). We want to remove the noise using a level set approach that adds a smoothing term to the isosurface extraction functional. One main contribution of this paper is to present a novel and efficient signed distance function computation, which enables us to use the noisy isosurface (b) as an initial surface for the level-set process. All computations are only carried out within a narrow band surrounding the zero-level set (c). Consequently, the level-set process converges after a few fast steps and produces a smooth isosurface (d).

To efficiently store the sample points and allow for fast neighbor queries, we use a spatial decomposition

based on a three-dimensional kd -tree as described in Section 4. In Section 5, the fast extraction of the zero-level set and the determination of the narrow-band points are described. The calculation of the level-set function as approximated signed-distance function to the zero-level set including the calculation of derivatives is explained in Section 6. The actual level-set process applied to the points of the narrow band and considerations about time step restrictions are presented in Section 7. Finally, results and their discussion, including the analysis of computation times and detailed comparisons, are provided in Section 8.

2 RELATED WORK

Visualizing large unstructured point-based volume data sets is challenging, especially when dealing with highly varying sample point densities. The generation of a polyhedrization (Edelsbrunner and Shah, 1992) of the unstructured data points can be very slow and is generally seen as impractical. The most common and well-established way of dealing with unstructured point-based volume data is to resample to a structured grid using scattered data interpolation techniques (Franke and Nielson, 1991; Park et al., 2006; Vuçini et al., 2009). After this preprocessing step, a variety of well-known visualization methods like iso-surface extraction or level-set segmentation can be applied to the gridded data to generate the desired visualizations.

In the majority of cases, level-set methods are executed on regular hexahedral grids that allow for fast discrete derivative computations. Unfortunately, the resampling to a regular grid always introduces inaccuracies, which heavily depend on the resolution of the grid and the point density of the data set. This can lead to enormous interpolation errors when dealing with data sets of highly varying point density and assuming regular grids that fit today’s hardware memory constraints. Using adaptive grids can reduce the error, but the more adaptive it gets, the more complicated the processing becomes. This observation raises the desire to directly operate on unstructured point-based volume data.

The original idea of level sets goes back to Sethian and Osher (Osher and Sethian, 1988; Osher and Fedkiw, 2003), who first describe the evolution of a surface by representing the surface implicitly as the solution of an equation with respect to an underlying scalar field. The solution is found by iteratively applying partial differential equations (PDEs) to the scalar field. Many different approaches exist and the range of application areas is wide. A general framework

for level-set segmentation of a large variety of regular data sets is presented in (Breen et al., 2005). Enright et al. apply a level-set approach to an octree-based adaptive mesh (Enright et al., 2004).

Level-set methods on gridded data nowadays typically operate locally. The introduction of local level-set methods goes back to (Adalsteinsson and Sethian, 1995). Local level-set methods, also called narrow-band methods, have developed rapidly in recent years (van der Laan et al., 2011; Nielsen and Museth, 2006), and have also been remodeled to work on today’s fast and parallel graphics hardware (Lefohn et al., 2004). However, the latter methods highly utilize the structure of regular grids and are not usable for unstructured data.

The only narrow-band approach, directly processing level sets on unstructured point-based volume data without any grid calculation or reconstruction of the scalar field was presented in (Rosenthal et al., 2010). The level-set function is initialized as a signed-distance function at the sample positions in a narrow band around a simple starting surface. The whole level-set evolution including the reinitialization of the narrow-band and level-set function is only based on these locations. Needed derivatives of the level-set function are approximated by a least-squares approach. The required update of the narrow band is done using an approximate signed-distance function and a numerical reinitialization step keeps the level-set function close to a signed-distance function. The approach represents the zero-level set implicitly and the level-set function must be initialized with a rather simple and inflexible signed-distance function.

In contrast to common local level-set methods, we further expand the idea of localization. After each level-set step and subsequent extraction of the zero-level set, the level-set function has to be set to a signed-distance function to ensure good numerical behavior (Peng et al., 1999). The common way of achieving this would be the numerical reinitialization of the level-set function to a signed-distance function (Schwartz and Colella, 2008), which is known to be rather time consuming. The direct construction of a signed-distance function can be achieved by the well-known fast-marching methods (Sethian, 1999; Tsai, 2002) which, however, always need an underlying grid or tessellation. Several papers propose the construction of a signed-distance function based on a moving least-squares fitting (Adamson and Alexa, 2003; Kolluri, 2008; Molchanov et al., 2010). These approaches can also be applied to unstructured point-based data. However, it is stated that they require a certain regular sampling density, which is also demonstrated in Section 8, where we compare our

signed-distance function approximation to those approaches.

To our knowledge, there exists no approach for smooth isosurface extraction which combines the accuracy of directly applying a narrow-band level-set method to unstructured data and explicit construction of the signed-distance function to the zero-level set. We propose such an approach that directly operates on unstructured point-based data. It does not need to reconstruct any scalar field at positions other than the sample points. Furthermore, the proposed method for approximating the signed-distance function is fast and robust against highly varying point densities.

3 GENERAL APPROACH

Let $D \subset \mathbb{R}^3$ be a bounded domain and $M \subset D$ a set of unstructured sample points $\mathbf{x}_i \in M$. Furthermore, let $f_i \in \mathbb{R}$ be the data values at the positions \mathbf{x}_i resulting in an unstructured point-based data set representing a scalar field $f : D \rightarrow \mathbb{R}$. Our goal is to extract an isosurface from this volumetric scalar data set with respect to a given isovalue $f_{\text{iso}} \in \mathbb{R}$. Moreover, the smoothness of the extracted isosurface, i. e. the mean curvature, should be controllable.

We use a level-set approach to manage the trade-off between both goals with the level-set equation

$$\frac{\partial \varphi}{\partial t} = ((1 - \lambda)(f - f_{\text{iso}} - \varphi) + \lambda \kappa_\varphi) \|\nabla \varphi\|. \quad (1)$$

Here $\varphi : \mathbb{R}^3 \times \mathbb{R}_+ \rightarrow \mathbb{R}$ denotes the level-set function, κ_φ the mean curvature of the respective level set, and $\lambda \in [0, 1]$ the smoothness parameter. This level-set equation models smooth isosurface extraction from the given scalar field f with respect to isovalue f_{iso} and smoothness λ .

For extracting the smooth isosurface, we dynamically modify a given initial surface $\Gamma \subset \mathbb{R}^3$, the zero-level set, with respect to the level-set problem until it reaches a steady state. To facilitate as much flexibility as possible, we assume the initial surface Γ to be a closed surface represented as a point cloud $P \subset \mathbb{R}^3 \times S^2$, where S^2 denotes the two-dimensional sphere in \mathbb{R}^3 . Here, the pairs $(\mathbf{p}_i, \mathbf{n}_i) \in P$ consist of points $\mathbf{p}_i \in \mathbb{R}^3$ with associated surface normals $\mathbf{n}_i \in S^2$. This surface representation is used, as it is the output of the direct isosurface extraction process from point-based volume data (Rosenthal and Linsen, 2006). This is how we typically generate the initial surface. Any other surface representation (e. g., an implicit surface representation or triangular meshes) can be easily converted to the described point-cloud representation.

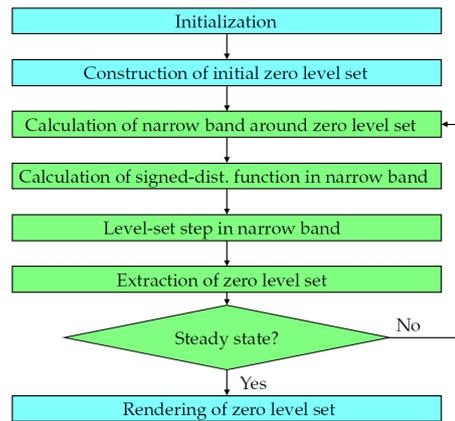


Figure 2: Illustration of the proposed level-set pipeline.

The zero-level set is implicitly represented as $\varphi = 0$ with respect to the level-set function φ which is modified following the level-set equation (1). This task is carried out utilizing a narrow-band level-set approach which only operates on the set of sample points M . The pipeline of this level-set method consists of three main phases: the initialization, the level-set evolution, and the rendering. Its flow chart is depicted in Figure 2.

In the initialization phase, the initial zero-level set is generated by a direct isosurface extraction technique. When applying this approach to other tasks (apart from smooth isosurface extraction), one can start with other appropriate surfaces. Alternatively, one can always start with a sphere lying in the bounding box of the data set. Note, that, when not starting with an initial surface close to the isosurface the final zero-level set can lack surface components of the global solution, similar to most narrow-band approaches. However, such misses can only occur in cases where the narrow band does not catch any part of the respective surface component during the whole level-set process. Moreover, this behavior can be used to extract specific components of a surface.

Furthermore, we build data structures to store and handle neighborhood information for the sample points. The data structures are used to extract points on the zero-level set efficiently, to generate the narrow band around the points of the zero-level set, and to compute the signed-distance function. The data structures we used are efficient implementations of a three-dimensional kd -tree.

In the iterative processing phase, first a narrow band of sample points is generated around the zero-level set. The narrow band consists of all sample points, which have a distance to the isosurface that is smaller than a given width of the narrow band. For all the sample points in the narrow band, the level-set

function ϕ is computed as signed-distance function to the zero-level set using an approximating scheme. The approximation results in an analytic representation of the signed-distance function, which allows for the direct computation of derivatives using analytic methods.

Subsequently, the actual level-set step following the level-set equation (1) is performed at all sample points in the narrow band. Since an explicit Euler time discretization is used for updating the level-set function at the sample points, the time steps are bounded by the Courant-Friedrichs-Lewy (CFL) condition (Courant et al., 1967) to permit numerical stability. In particular, this condition also guarantees that the zero-level set cannot leave the narrow band in a single iteration. In fact, we restrict the level-set step so that the zero-level set does not move more than half of the width of the narrow band.

The zero-level set to the processed level-set function is extracted using direct isosurface extraction (Rosenthal and Linsen, 2006) and the surface normals required are obtained by normalizing the level-set function gradients. The approach computes points on the zero-level set (leading to a point-based surface representation) by linear interpolation between suitable neighbors of data points. This completes one iteration step.

The iterative process is executed until the level-set function reaches a steady state. Note, that apart from the local nature of the approach the resulting smooth isosurface is independent of the starting surface and does only depend on the convergence threshold and the smoothness parameter λ . After convergence, the zero-level set is rendered using point-based rendering techniques (Gross and Pfister, 2007; Rosenthal and Linsen, 2008a).

4 DATA MANAGEMENT

For efficient calculations of neighborhood information and access to the sample and zero-level set points, auxiliary data structures are needed. First, a structure that allows for the efficient extraction of the points of the zero-level set from the sample points and the fast generation of the narrow band is desired. Since the positions of the sample points do not change during the iterative phase, this can be static and, thus, created in a preprocessing step. The unstructured data points are stored using an efficient three-dimensional *kd*-tree implementation. From this *kd*-tree the neighborhood information for extracting the zero-level set is derived in the same way as it was proposed in (Rosenthal and Linsen, 2008b). The determination of all sample

points lying in the narrow band is achieved by utilizing maximum distance queries from the zero-level set points on the *kd*-tree of the sample points.

A second data structure is needed for speeding up the approximation of the signed-distance function to the zero-level set. Note that the first *kd*-tree (containing the sample points) could also have been used for this purpose. However, since distribution and quantity of sample points and points of the zero-level set can differ substantially, it is more efficient to generate in each iteration a new *kd*-tree for the points of the new zero-level set.

5 ZERO-LEVEL SET EXTRACTION AND CREATION OF NARROW BAND

After each level-set step, the zero-level set is extracted in an explicit form, more precisely as a point-cloud surface. This step is equivalent to extracting the isosurface with isovalue 0 from the updated level-set function. For extracting smooth isosurfaces from scalar fields, a good initial guess of the smooth isosurface can be obtained by extracting the respective isosurface from the given scalar field.

Both tasks are done using the direct isosurface extraction approach for unstructured point-based volume data presented in (Rosenthal and Linsen, 2006). This approach utilizes the first *kd*-tree that stores the data samples. An appropriate neighborhood approximating the natural neighborhood is generated for each sample point utilizing an efficient indexing scheme. Points of the zero-level set are directly extracted by linearly interpolating between neighboring samples with different signs of the level-set function. This results in a zero-level set in point-cloud representation with local point density proportional to the local point density of the data set. Surface normals at the zero-level set points are obtained as the normalized approximated gradient of the underlying scalar field using a four-dimensional least-squares approach (Rosenthal and Linsen, 2008b).

Each level-set step is bounded by the CFL-condition, allowing for stability of the whole level-set process. Thus, the movement of the zero-level set is also bounded for each level-set iteration and it is possible to restrict the level-set process to a narrow band around the zero-level set without losing flexibility or distorting the result. For generating the narrow band, all sample points having a smaller distance to the zero-level set than the width of the band are marked as belonging to the band. As the zero-level set

is extracted in a point cloud representation, we define the distance of a sample point to the zero-level set as the distance to the nearest point of the zero-level set. For each zero-level set point all sample points with distance smaller than the band's width are efficiently obtained by a nearest-neighbor query to the kd -tree storing the sample points.

The width of this narrow band depends on the distribution of the sample points, i. e., it is data-dependent. To capture the entire area surrounding the zero-level set, the width of the narrow band has to be greater than the maximum distance of neighboring zero-level set points. In fact, we use twice that distance to ensure that the zero-level set is kept in the narrow band. In all our experiments, it was sufficient to choose one global value for the band's width for the whole local level-set process, even for data sets with highly varying point densities. Nevertheless, the performance of the approach can be tuned further by dynamically adjusting the band's width with respect to the CFL-condition. This condition restricts the time step of each level-set step and thus the movement of the zero-level set. Choosing the width of the narrow band to the double of the maximum distance zero-level set points move is more than sufficient. In particular, when the level-set process is close to convergence, the zero-level set moves only slowly and a very small narrow band can be chosen.

6 SIGNED-DISTANCE FUNCTION COMPUTATION

Let $P \subset \mathbb{R}^3 \times S^2$ be a set of pairs $(\mathbf{p}_i, \mathbf{n}_i)$, where $\mathbf{p}_i \in \mathbb{R}^3$ is a point on a two-dimensional surface in space and $\mathbf{n}_i \in S^2$ is the associated surface normal. Our goal is to approximate the signed-distance function to the surface represented by P . More precisely, we want to construct a smooth scalar field $\varphi: \mathbb{R}^3 \rightarrow \mathbb{R}$ with $\varphi(\mathbf{p}_i) = 0$ for all points \mathbf{p}_i on the surface and $\|\nabla\varphi(\mathbf{x})\| \approx 1$ for all points $\mathbf{x} \in \mathbb{R}^3$.

Following the lines of the whole level-set method, the approximation should not depend on an underlying scalar field and should be robust against highly varying point density of the point cloud. The main idea comes from the observation that the signed-distance function can be approximated accurately on the line $\gamma_i(t) = \mathbf{p}_i + t \cdot \mathbf{n}_i$ near each surface point by setting $\varphi(\gamma_i(t)) = t$. In a small neighborhood around a line this measurement is still accurate and we can approximate

$$\varphi(\mathbf{x}) \approx f_i(\mathbf{x}) := \langle \mathbf{x} - \mathbf{p}_i, \mathbf{n}_i \rangle, \quad (2)$$

i. e., we use the signed distance between the projec-

tion of \mathbf{x} to γ_i and the point \mathbf{p}_i . Here $\langle \cdot, \cdot \rangle$ denotes the scalar product.

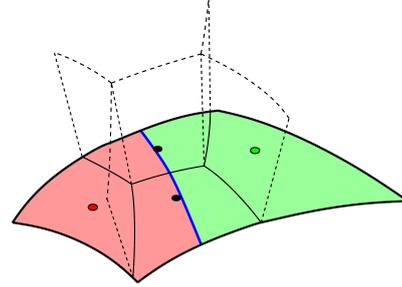


Figure 3: For the points of the surface the hypothetical two-dimensional Voronoi diagram, depicted by the solid lines, induces truncated cones in space, depicted by the dashed lines. The blue line indicates the curve on the surface where the third-nearest point and the fourth-nearest point, i. e., the red and the green point, respectively, are equally distant.

This approximation should be done for all points \mathbf{x} for which \mathbf{p}_i is the nearest neighbor, i. e., for all points lying in the truncated cones induced by a hypothetical two-dimensional Voronoi diagram on the surface, as illustrated by Figure 3. Just applying this approximation in the respective cones would result in discontinuities in the resulting scalar field at the borders of the truncated cones. Hence, the respective functions f_i should be blended to produce a smooth scalar field.

Although theoretically a large number of Voronoi cells can meet at a vertex, in practice - when using floating point representations of finite precision - one can assume that at most three cells meet exactly at the vertices of a 2D Voronoi diagram. These vertices result in edges between the associated truncated cones, as illustrated in Figure 3. Consequently, the blending between different cells has to handle the transition between the three nearest neighbors to a particular point \mathbf{x} , i. e., the signed-distance approximation has to be symmetric in the three nearest neighbors of \mathbf{x} .

Additional care to avoid discontinuities has to be taken at the positions where the third-nearest neighbor to \mathbf{x} changes to a point which was not considered before, as depicted by the blue line in Figure 3. More precisely, the contribution of the third-nearest point to φ should vanish when the distances of \mathbf{x} to the third- and fourth-nearest points are equal.

We implement these considerations by defining an approximation to the signed-distance function as follows: Let $\mathbf{x} \in \mathbb{R}^3$ and \mathbf{p}_i be the i th-nearest point of P to \mathbf{x} . We define the signed-distance function at \mathbf{x} to P by

$$\varphi = \frac{\lambda_1 d_2 d_3 \cdot f_1 + \lambda_2 d_1 d_3 \cdot f_2 + \lambda_3 d_1 d_2 \cdot f_3}{\lambda_1 d_2 d_3 + \lambda_2 d_1 d_3 + \lambda_3 d_1 d_2}, \quad (3)$$

i. e., φ is the convex combination of the local signed-

distance functions

$$f_i = f_i(\mathbf{x}) := \langle \mathbf{x} - \mathbf{p}_i, \mathbf{n}_i \rangle \quad (4)$$

with weight functions defined by

$$d_i = d_i(\mathbf{x}) := \|\mathbf{x} - (\mathbf{p}_i + f_i(\mathbf{x}) \cdot \mathbf{n}_i)\| \quad (5)$$

$$\lambda_i = \lambda_i(\mathbf{x}) := l_4(\mathbf{x}) - l_i(\mathbf{x}) \quad (6)$$

$$l_i = l_i(\mathbf{x}) := \|\mathbf{x} - \mathbf{p}_i\| . \quad (7)$$

The continuous change of the signed-distance function between the three nearest surface points is accomplished by the functions $d_i(\mathbf{x})$, which calculate the orthogonal distance of the point \mathbf{x} to the line induced by the i th-nearest surface point and its surface normal. The functions $\lambda_i(\mathbf{x})$ represent the differences of the distances from the point \mathbf{x} to the fourth-nearest surface point and from the point \mathbf{x} to the i th-nearest surface point. These functions $\lambda_i(\mathbf{x})$ assure a continuous behavior of the signed-distance function if the third- and fourth-nearest neighbor of \mathbf{x} are having similar distances to \mathbf{x} . In this region, indicated by the blue line in Figure 3, the order of the third- and fourth-nearest neighbor might switch. An illustration of the functions is given in Figure 4.

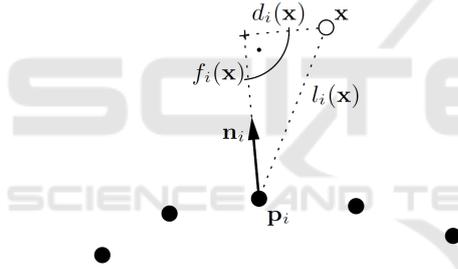


Figure 4: Illustration of the functions used to approximate the signed-distance function. For a point \mathbf{x} and each point \mathbf{p}_i , $f_i(\mathbf{x})$ denotes the distance between \mathbf{p}_i and the base point of \mathbf{x} on γ_i , $d_i(\mathbf{x})$ denotes the distance of the same base point to \mathbf{x} , and $l_i(\mathbf{x})$ is the distance between \mathbf{x} and \mathbf{p}_i .

If the unlikely case appears that more than three of the nearest neighbors have very similar distances to \mathbf{x} , the λ_i nearly vanish and the signed-distance function would get very unstable. In our implementation, we detect such cases with a threshold. For ($\lambda_1 > 0.0001$) the standard procedure for computing φ is used. For ($\lambda_1 < 0.00001$) we compute φ by setting $\lambda_i = 1$. In between we use a linear interpolation between the two approaches.

To achieve a signed-distance function which is differentiable everywhere, the weight functions d_i and λ_i can be smoothed with any standard kernel function. In our implementation, we applied

$$p: [0, 1] \rightarrow [0, 1], \quad p(t) := 6t^5 - 15t^4 + 10t^3 \quad (8)$$

which is the lowest-order polynomial with vanishing first- and second-order derivatives at 0 and 1.

The blending with this polynomial produces C^2 -continuous transitions. The approximated signed-distance function has the following properties:

1. $\varphi(\mathbf{x}) = f_1(\mathbf{x})$ for all points $\mathbf{x} \in \gamma_1$ with nearest neighbor \mathbf{p}_1 .
2. The change between cells of the Voronoi diagram leads to a continuous change in φ , since φ is symmetric.
3. Changing third- and fourth-nearest neighbor leads to a continuous change in φ , since $\lambda_3 = 0$.

In contrast to several related approaches, the proposed signed-distance approximation is very robust against non-uniformly sampled point clouds. This results in the minimization of artifacts even when dealing with data sets of highly varying point density. We demonstrate this property with the help of artificial and real-world data sets in Section 8.

Derivatives Calculation

During the level-set process, not only the constructed signed-distance function but typically also derivatives of the function are used. Since the signed-distance function is described analytically, all desired derivatives can be computed. We list the first derivatives of the functions used:

$$\nabla \lambda_i(\mathbf{x}) = \frac{(\mathbf{x} - \mathbf{p}_i)}{l_i(\mathbf{x})} - \frac{(\mathbf{x} - \mathbf{p}_4)}{l_4(\mathbf{x})} \quad (9)$$

$$\nabla d_i(\mathbf{x}) = \frac{\mathbf{x} - \mathbf{p}_i - f_i(\mathbf{x}) \cdot \mathbf{n}_i}{d_i(\mathbf{x})} \quad (10)$$

$$\nabla f_i(\mathbf{x}) = \mathbf{n}_i \quad (11)$$

Also higher-order derivatives can be easily derived. In particular the curvature of a level set is given by

$$\kappa_\varphi = \nabla \cdot \frac{\nabla \varphi}{\|\nabla \varphi\|} = \frac{1}{\|\nabla \varphi\|} (\nabla \cdot \nabla \varphi - 1), \quad (12)$$

which can be explicitly computed from the level-set function, i. e., from the signed-distance function approximation.

To compute the signed-distance function values and derivatives efficiently, a three-dimensional k d-tree is built for the zero-level set points of each time step. Utilizing this spatial space partitioning, the four nearest zero-level set points are computed for each sample point in the current narrow band.

7 LEVEL-SET PROPAGATION

The propagation of the level-set function is performed following the level-set equation (1), which models the

smooth isosurface extraction as combination of hyperbolic advection to the data scalar field and mean curvature flow. This is done using an explicit time discretization scheme of order one. For this level-set process only the level-set function values and derivatives obtained at the sample points are used. Furthermore, no information about the scalar field other than at the sample points in the narrow band is needed.

Since the level-set function is explicitly reset to a signed-distance function in each level-set step, the level-set propagation does not have to maintain this property to produce high-quality results. However, the used time step Δt has to be chosen carefully to keep the zero-level set always in the narrow band and fulfill the CFL-condition. In fact, meeting the CFL-condition described in (Rosenthal and Linsen, 2008b), i. e.

$$\Delta t \left(\frac{(1-\lambda)|f - f_{\text{iso}} - \varphi|}{d_{\text{min}} \|\nabla\varphi\|} \|\text{div}\varphi\| + \frac{6\lambda}{d_{\text{min}}^2} \right) < 1 \quad (13)$$

with

$$\|\text{div}\varphi\| := \left(\left\| \frac{\partial\varphi}{\partial x_1} \right\| + \left\| \frac{\partial\varphi}{\partial x_2} \right\| + \left\| \frac{\partial\varphi}{\partial x_3} \right\| \right) \quad (14)$$

and d_{min} denoting the distance to the nearest neighbor, is also sufficient to keep the zero-level set within the narrow band during computations. This was already observed in (Peng et al., 1999).

8 RESULTS AND DISCUSSION

The presented approach was applied to a variety of data sets to verify our method and evaluate it in terms of accuracy and speed. All computation times were measured on a single 2.66GHz XEON processor.

First we have tested our proposed signed-distance function approximation approach in terms of accuracy, since this is a crucial part of our level-set pipeline. We compared our approach with two very related and recently presented methods (Adamson and Alexa, 2003; Molchanov et al., 2010). For this purpose, we have applied all three methods to an artificial two-dimensional point cloud, which represents the curve of a limaçon of Pascal (Lawrence, 1972) sampled at 1,000 random positions. Visualizations of the respective isolines of the signed-distance function approximations are shown in Figure 5. Following suggestions of the respective papers, both competing methods are applied with 15 neighbors for interpolation. They require a relatively uniform sampled point cloud and result in enormous errors in the approximation causing deformed isolines. Our method is robust against highly varying point densities and does not exhibit such artifacts. Still, computations of our method

are faster since only a fixed number of four neighbors is used. For a surface with 100,000 points and 240,000 distance computations the method of Adamson and Alexa and the method of Molchanov et al. both need 112 seconds, in comparison to 103 seconds with our method.

In addition, we compared the approximated and exact signed-distance function values for our method numerically. For this purpose, we sampled the curve with different numbers of points and approximated the signed-distance function at one million randomly distributed sample points in a band around the curve. At the same sample points of the band, we computed the exact signed-distance function values using the analytic expression of a limaçon. The obtained numerical differences are shown in Table 1. The relative error of our approach is reciprocal to the number of sample points on the curve. For the number of surface points we encountered in our real-world examples (about 10^4), we have a maximum error of 0.23% and a relative error of 0.053%. All our experiments showed, that this is sufficiently low to allow for accurate level-set computations.

Table 1: Error analysis for signed-distance function approximation to the limaçon of Pascal. For different number of random sample points on the curve, the relative average error and the relative maximum error is given when comparing the approximated with the exact signed-distance function values at one million points in a band around the curve.

# points	rel. average error	rel. maximum error
10^3	$5.30 \cdot 10^{-3}$	$2.21 \cdot 10^{-2}$
10^4	$5.34 \cdot 10^{-4}$	$2.30 \cdot 10^{-3}$
10^5	$5.47 \cdot 10^{-5}$	$3.18 \cdot 10^{-4}$
10^6	$5.56 \cdot 10^{-6}$	$3.54 \cdot 10^{-5}$
10^7	$5.48 \cdot 10^{-7}$	$4.66 \cdot 10^{-6}$

For the analysis of performance of our level-set pipeline, we applied it to an unstructured point-based volume data set with 16 million randomly distributed sample points. The data set was generated by resampling the regular Hydrogen data set (courtesy of SFB 382 University Tübingen) of size $128 \times 128 \times 128$ to the random positions. (For all data sets that are resampled from regular data to a random distribution, we never make use of the original regular structure.) An illustration of the evolution process for this data set and a sphere as initial surface is shown in Figure 6. The whole level-set process for extracting a smooth isosurface was performed in 18 minutes, including preprocessing. In comparison, we achieved a computation time of 36 minutes with the method from (Rosenthal et al., 2010). When starting with the (non-smooth) isosurface as a first guess our computation time drops to 22 seconds and we obtain a

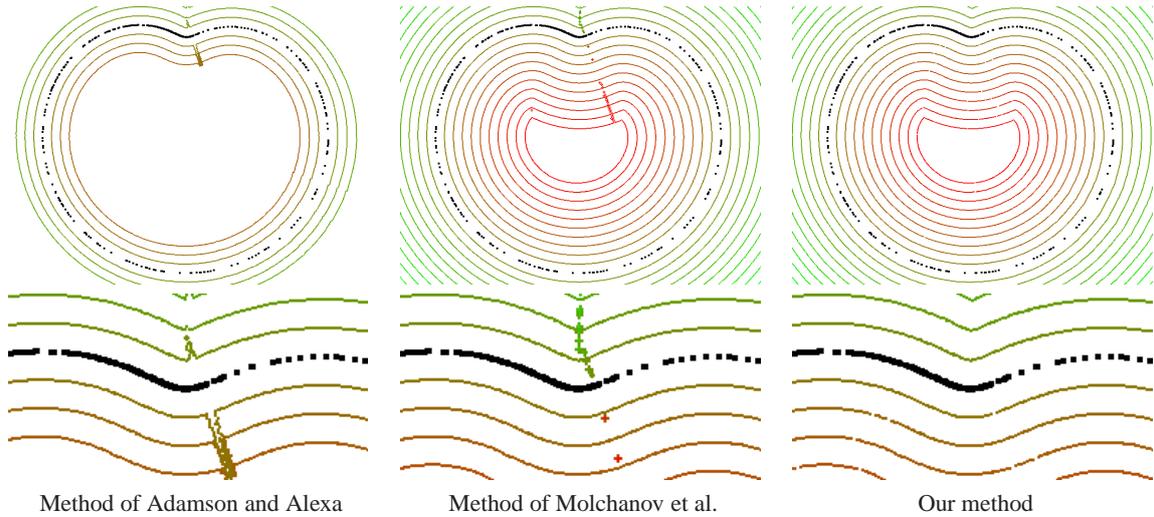


Figure 5: Qualitative comparison of our method with two competing methods for computing the signed-distance field of a limaçon of Pascal, non-uniformly sampled at 1,000 positions. In the upper row, the points on the limaçon are colored black. Additionally, isolines to the respective signed-distance approximation are drawn, color-coded with respect to the isovalue. The close-up views in the lower row show errors for the competing methods at the region of highly varying point density.

speed-up of two orders of magnitude. Starting with the isosurface is not possible when using the approach from (Rosenthal et al., 2010).

Table 2: Computation times for the iterative level-set phase of a data set with 16 million sample points and 50,000 zero-level set points. For different sizes of the narrow band (nb), we list the number of sample points in the narrow band, the computation time for determining the points in the narrow band, and the computation time for the level-set (ls) step.

# points in nb	nb comp.	ls step
200k	0.4 s	3.0 s
400k	2.0 s	5.1 s
800k	16.3 s	17.1 s
1,600k	67.5 s	70.6 s

We analyzed the computation time for the iterative phase with respect to different widths of the narrow band, i. e., for different numbers of narrow-band points. For the Hydrogen data with 16 million sample points the times are given in Table 2. Here, the computation time for the narrow-band creation includes creation of a kd -tree for the zero-level set points, the marking of all data points closer to the zero-level set points than the width of the band, and the calculation of the signed-distance function values and derivatives at data points in the narrow band. The times for the level-set step include the update of the level-set function values following the level-set equation, the extraction of the new zero-level set, as well as the calculation of surface normals for the zero-level set points. It is favorable to choose the width of the band with respect to the CFL-condition to achieve an optimal

balance between large time step and small number of processed sample points.

Finally, we applied our local level-set method to real-world unstructured point-based volume data sets with point densities varying in two orders of magnitude. These highly varying point densities would lead to enormous interpolation errors, when not operating directly on the sample point locations. The first data set stems from an astrophysical particle simulation using smoothed particle hydrodynamics. In the simulation, two stars orbiting each other are represented by a set of particles. During the simulation, the stars exchange gas through a jet between them and get distorted by the jet and gravity forces. In Figure 7(a), we visualize a snapshot of the simulation at a late point in time, when one star (the one in the upper left) is already distorted, while the other star (the one in the lower right) was not yet affected significantly. We compare our approach with direct isosurface extraction on this data set with 2,600,000 sample points and extract surfaces with respect to gas density. The whole computation time for 10 level-set steps was 42 seconds. In contrast to direct isosurface extraction, the smooth surface generated by our level-set approach does not exhibit any outliers.

The direct isosurface extraction suffers from severe incorrect outliers, caused by the very different gradient magnitudes in the data set. This is also the reason why we did not choose the isosurface as initial zero-level set for our local level-set method. The errors in the position of the isopoints can cause severe errors in the surface normal approximation and lead to

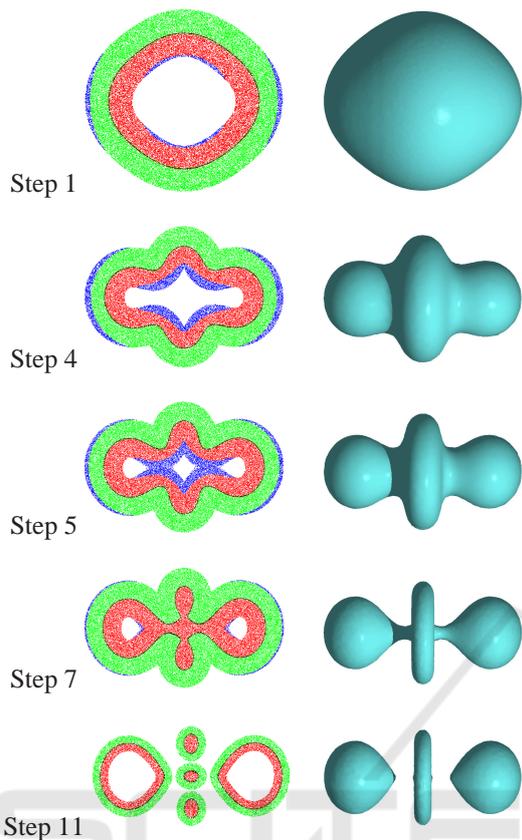


Figure 6: Evolution of the zero-level set for normal advection to the Hydrogen data set with 16 million sample points. For each time step, a splat-based ray tracing of the zero-level set is shown on the right-hand side. On the left-hand side, a point rendering of a slab of the data set is shown illustrating the narrow band. The surface points extracted are colored black, points with negative level-set function value are colored red, and points with positive level-set function value are colored green. Points which newly entered the narrow band are colored blue. All data points not belonging to the narrow band are not rendered. Note the easy change of topology of the zero-level set and dynamic adjustment of the narrow band. If the iterative process is close to steady state, a much smaller width of the narrow band can be used.

an unusable signed-distance function approximation. Instead, there are two different options for initializing such tricky cases. If the approximate shape of the final surface is known the zero-level set can be initialized as an approximation to the final surface, as in the presented case, where the zero-level set was initialized as two spheres centered at the stars’ barycenters. Alternatively, one can extract the zero-level set as iso-surface from a smoothed version of the original data set. In most cases, the band around this initial surface will still catch all parts of the final smooth isosurface and ensure a correct level-set propagation, which is then carried out again with the original data.

The second data set is the White Dwarf data set with 500,000 sample points already used by Rosenthal et al. It models a small star that is torn apart by the strong gravity forces of a black hole. We used the same parameters and initial surface as for their method and achieved an overall computation time of 182 seconds, which still is two times faster compared to 366 seconds of the competing method (Rosenthal et al., 2010). In terms of quality, the results are comparable. A rendering of the extracted zero-level set and a slab of the narrow band is shown in Figure 7(b). The maximum distance between the points of the two resulting surfaces was of the same order as the stopping criterion used for both level-set processes.

All experiments have shown, that the proposed method for approximating the signed-distance function produces results of high quality. The local level-set method using explicit zero-level set representation is significantly faster than competing methods. Compared to the previous local level-set approach (Rosenthal et al., 2010) the proposed method speeds up each level-set step by reducing the number of needed neighbors per point and omitting the numerical reinitialization. Additionally, it facilitates the usage of any initial zero-level set, significantly reducing computation times further. A comprehensive performance overview of our approach for the various data sets including comparisons to the prior narrow-band technique is given in Table 3. Note the remarkable speed up when using the proposed method with a good initial guess as starting surface, compared to initialization with a sphere required by competing methods with implicit zero-level set representation.

Still, experiments have shown that the proposed method is, in terms of quality, equivalent to the previous global and local methods by Rosenthal et al. and it has successfully been applied to real-world data. Moreover and in contrast to the prior local level-set approach (Rosenthal et al., 2010), it is able to extract objects with complex topology very fast by using an initial surface close to the expected final result, such as the (non-smooth) isosurface.

9 CONCLUSION

We have presented a local level-set method that combines the accuracy of directly applying level sets to unstructured data with the fast level-set computation using narrow bands and an explicitly extracted zero-level set. This option enables us to use any surface as initial zero-level set and gives an additional speed-up when using good initial guesses for the final surface. We directly operate on the unstructured data

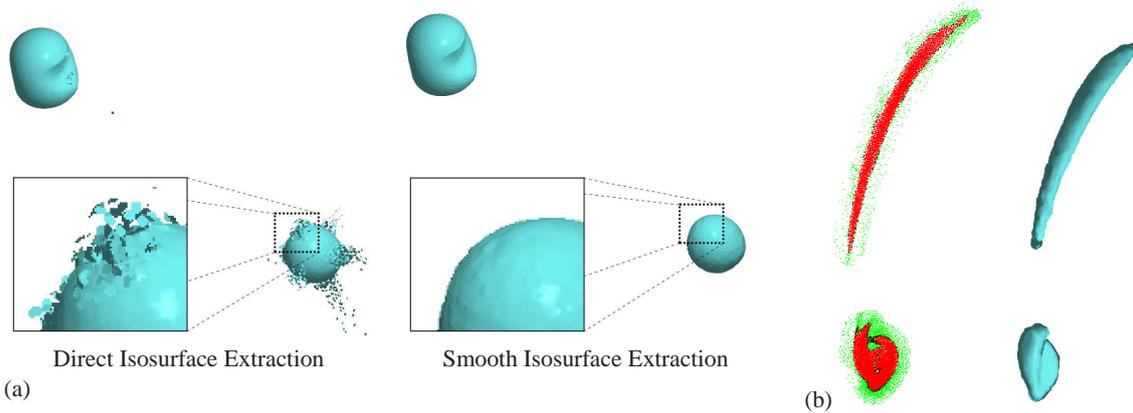


Figure 7: (a) Comparison between direct isosurface extraction on the left-hand side and smooth isosurface extraction on the right-hand side for a stars simulation data set with 2.6 million sample points. Our level-set method effectively eliminates the noise in the surface. (b) Illustration of the zero-level set for normal advection to the White Dwarf data set with 500,000 sample points. On the left-hand side, a point rendering of a slab of the data set is shown illustrating the narrow band. A splat-based ray tracing of the zero-level set after convergence of the level-set process is shown on the right-hand side.

Table 3: Performance comparison for the presented data sets. For each data set the number of points, the used isovalue, and the used smoothness parameter is given. Additionally, the initial surface, the needed number of level-set steps, and the overall computation time, including the generation of the initial zero-level set, is given for our approach and for the method proposed by Rosenthal et al., followed by our achieved speed-up.

data set	# points	f_{iso}	λ	our method			(Rosenthal et al., 2010)			speed-up
				init. s.	#steps	time	init. s.	#steps	time	
Hydrogen	16.0M	30	0.05	isosurface	3	22 sec	sphere	19	36 min	98.1
Engine	16.0M	70	0.10	isosurface	10	59 sec	sphere	27	47 min	47.7
2 stars	2.6M	0.01	0.10	2 spheres	10	42 sec	sphere	-	-	-
Wh. Dwarf	0.5M	0.00006	0.10	sphere	37	182 sec	sphere	39	366 sec	2.0

and do not need to process data at any position other than the sample points. For each iteration, we build a narrow band of sample points around the zero-level set, compute the level-set function at these points as a signed-distance function to the zero-level set, process the level-set function in the narrow band with respect to a given level-set equation, and extract the new zero-level set. As we use a signed-distance function for the level-set function, we avoid numerical reinitialization steps, which further reduces computation times. After convergence, the final zero-level set is visualized using point-based rendering techniques.

Similar to most narrow-band, thus local, techniques our method is only able to extract surfaces which are captured by the computation area during the level-set process. However, this covering can be guaranteed when using the directly extracted isosurface as starting surface for the level-set processing. We have compared our results with the ones obtained by Rosenthal et al. and achieved significant speed-ups. Still, the proposed method achieves the same results in terms of quality. Our method is capable of robustly processing data sets with several million sample points and highly varying point density.

ACKNOWLEDGMENTS

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under project grant LI 1530/6-2.

REFERENCES

- Adalsteinsson, D. and Sethian, J. A. (1995). A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118(2):269–277.
- Adamson, A. and Alexa, M. (2003). Approximating and intersecting surfaces from points. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 230–239, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.
- Breen, D., Whitaker, R., Museth, K., and Zhukov, L. (2005). Level set segmentation of biological volume datasets. In Suri, J. S., Wilson, D. L., and Laxminarayan, S., editors, *Handbook of Biomedical Image Analysis, Volume I: Segmentation Models, Part A*, pages 415–478, New York, NY, USA. Kluwer.

- Courant, R., Friedrichs, K. O., and Lewy, H. (1967). On the partial difference equations of mathematical physics. *IBM Journal of Research and Development*, 11(2):215–234.
- Edelsbrunner, H. and Shah, N. R. (1992). Incremental topological flipping works for regular triangulations. In *SCG '92: Proceedings of the eighth annual symposium on Computational geometry*, pages 43–52, New York, NY, USA. ACM.
- Enright, D., Losasso, F., and Fedkiw, R. (2004). A fast and accurate semi-lagrangian particle level set method. *Computers and Structures*, 83(6–7):479–490.
- Franke, R. and Nielson, G. M. (1991). Scattered data interpolation: A tutorial and survey. In Hagen, H. and Roller, D., editors, *Geometric Modeling: Methods and Applications*, pages 131–160. Springer, New York, NY, USA.
- Gross, M. and Pfister, H. (2007). *Point-Based Graphics*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Kolluri, R. (2008). Provably good moving least squares. *ACM Trans. Algorithms*, 4(2):1–25.
- Lawrence, J. D. (1972). *A Catalog of Special Plane Curves*. Dover Publications.
- Lefohn, A. E., Kniss, J. M., Hansen, C. D., and Whitaker, R. T. (2004). A streaming narrow-band algorithm: Interactive computation and visualization of level sets. *IEEE Transactions on Visualization and Computer Graphics*, 10(4):422–433.
- Molchanov, V., Rosenthal, P., and Linsen, L. (2010). Non-iterative second-order approximation of signed distance function for any isosurface representation. *Computer Graphics Forum*, 29(3):1211–1220.
- Nielsen, M. B. and Museth, K. (2006). Dynamic tubular grid: An efficient data structure and algorithms for high resolution level sets. *J. Sci. Comput.*, 26:261–299.
- Osher, S. and Fedkiw, R. (2003). *Level set methods and dynamic implicit surfaces*. Springer, New York, NY, USA.
- Osher, S. and Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49.
- Park, S. W., Linsen, L., Kreylos, O., Owens, J. D., and Hamann, B. (2006). Discrete Sibson interpolation. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):243–253.
- Peng, D., Merriman, B., Osher, S., Zhao, H., and Kang, M. (1999). A PDE-based fast local level set method. *Journal of Computational Physics*, 155(2):410–438.
- Rosenthal, P. and Linsen, L. (2006). Direct isosurface extraction from scattered volume data. In Santos, B. S., Ertl, T., and Joy, K. I., editors, *EuroVis06: Proceedings of the Eurographics/IEEE-VGTC Symposium on Visualization*, pages 99–106, Aire-la-Ville, Switzerland. Eurographics Association.
- Rosenthal, P. and Linsen, L. (2008a). Image-space point cloud rendering. In *Proceedings of Computer Graphics International*, pages 136–143.
- Rosenthal, P. and Linsen, L. (2008b). Smooth surface extraction from unstructured point-based volume data using PDEs. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1531–1546.
- Rosenthal, P., Molchanov, V., and Linsen, L. (2010). A narrow band level set method for surface extraction from unstructured point-based volume data. In Skala, V., editor, *Proceedings of WSCG, The 18th International Conference on Computer Graphics, Visualization and Computer Vision*, pages 73–80, Plzen, Czech Republic. UNION Agency – Science Press.
- Schwartz, P. and Colella, P. (2008). A second-order accurate method for solving the eikonal equation. *Proceedings in Applied Mathematics and Mechanics*, 7(1).
- Sethian, J. A. (1999). *Level Set Methods and Fast Marching Methods*. Cambridge University Press, Cambridge, UK, 2nd edition.
- Tsai, Y.-h. R. (2002). Rapid and accurate computation of the distance function using grids. *Journal of Computational Physics*, 178(1):175–195.
- van der Laan, W. J., Jalba, A. C., and Roerdink, J. B. T. M. (2011). A memory and computation efficient sparse level-set method. *Journal of Scientific Computing*, 46(2):243–264.
- Vučini, E., Möller, T., and Gröller, M. E. (2009). On visualization and reconstruction from non-uniform point sets using b-splines. *Computer Graphics Forum*, 28(3):1007–1014.