Eurographics/ IEEE-VGTC Symposium on Visualization 2010
G. Melançon, T. Munzner, and D. Weiskopf
(Guest Editors)

Volume 29 (2010), Number 3

Non-iterative Second-order Approximation of Signed Distance Functions for Any Isosurface Representation

V. Molchanov, P. Rosenthal, L. Linsen

Jacobs University, Bremen, Germany

Abstract

Signed distance functions (SDF) to explicit or implicit surface representations are intensively used in various computer graphics and visualization algorithms. Among others, they are applied to optimize collision detection, are used to reconstruct data fields or surfaces, and, in particular, are an obligatory ingredient for most level set methods. Level set methods are common in scientific visualization to extract surfaces from scalar or vector fields. Usual approaches for the construction of an SDF to a surface are either based on iterative solutions of a special partial differential equation or on marching algorithms involving a polygonization of the surface. We propose a novel method for a non-iterative approximation of an SDF and its derivatives in a vicinity of a manifold. We use a second-order algebraic fitting scheme to ensure high accuracy of the approximation. The manifold is defined (explicitly or implicitly) as an isosurface of a given volumetric scalar field. The field may be given at a set of irregular and unstructured samples. Stability and reliability of the SDF generation is achieved by a proper scaling of weights for the Moving Least Squares approximation, accurate choice of neighbors, and appropriate handling of degenerate cases. We obtain the solution in an explicit form, such that no iterative solving is necessary, which makes our approach fast.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism—

1. Introduction

Geometric object representations in computer graphics and scientific visualization are typically given in the form of curves and surfaces. In scientific visualization, they are frequently used to represent boundary shapes of features such as isocontours of 2D or 3D scalar fields. Therefore, problems on efficient and accurate extraction, generation, reconstruction, and regularization of surfaces are important tasks of modern visualization research.

A representation dual to a surface is its *signed distance function* (SDF). For any regular manifold it is possible to define a scalar function evaluating the distance of a point to the surface and distinguishing by its sign the two parts of the space separated by the manifold. On the other hand, any reasonable SDF can be considered as an implicit description of its zero-level isosurface. Both of these two approaches find a wide field of applications in modern science.

In this paper we propose a fast and accurate approximation of an SDF in a narrow neighborhood of a two-dimensional surface. Our goal is to derive analytical expressions for function values and its derivatives to avoid time-consuming iterative computations. The desired accuracy is obtained by using a second-order scheme.

For our algorithm we consider the most general set-up: Surfaces may be given explicitly in form of points on the surface or implicitly in form of a level set to a scalar field. The scalar field itself is known at a finite set of 3D sample points, which do not need to exhibit any spatial structure, regularity, or connectivity, i.e., the samples may be unstructured and irregularly distributed. This set-up is a generalization of any commonly used surface or scalar field representation. In particular, it includes scalar field representations over structured and unstructured grids. In this paper, we discuss the algorithms in their most general form. We do not discuss possible simplifications of our algorithm or improvement of performance for special cases. One implication of using implicit surface representations is that the surface can

be a manifold of any topology. In particular, it may consist of multiple components. In Section 3, we formally give the problem specification.

The main concepts involved in our algorithm are point-based representation of isosurfaces, localization in a narrow band, *Moving Least Squares* (MLS) approximation, and algebraic surface fitting. Significant efforts are directed to improve stability and reliability of computations. The implementation of these steps is described in Section 4. The individual components of the algorithm are isopoint extraction, narrow band generation, spherical fitting, stability improvement, and SDF approximation.

The main contributions of our paper are: (1) application of algebraic sphere fitting to estimate an SDF to an isosurface, (2) modification of an MLS approach to avoid vanishing smoothing weights and to increase the stability of calculation, (3) two-angles criterion for special case detection (e.g. layer sheet, sharp features), and (4) testing of the method proposed to investigate its accuracy and efficiency. The latter is done in form of numerical tests described in Section 5. Section 6 gives further results and discusses them.

The impact of our work is in any area that involves the use of signed distance functions. A few prominent examples are:

- Distance computations between moving objects: The SDF is exploited to measure distances between objects and to predict how quickly they are approaching each other and when they will collide. Such distance computation and collision detection are of utmost importance in physical simulations and computer-aided manufacturing, but may also be used in scientific visualization.
- Data field and surface reconstruction: When reconstructing a data field using scattered data interpolation or reconstructing a surface from a point-cloud representation, SDFs are frequently been embedded in Voronoi diagram generations (or Delaunay triangulation/tetrahedrization).
- Level-set methods: Level set methods are common in scientific visualization to extract surfaces from scalar or vector fields. Almost all methods incorporate a *re-initialization* of the level set function ϕ to an SDF as a part of the algorithm. When assuming that the underlying function is an SDF, several simplifications and speed-ups of the level set approach can be achieved.

  Then, for numerical accuracy, a level set function must stay well behaved close to the front [PMO*99], i.e. $|\nabla\phi| \approx 1$. Since flat and/or steep regions develop in the advection step, a periodic re-initialization is required to maintain this property.

  Moreover, in the context of a non-morphological approach [OF03], when the gradient of the level set function $|\nabla\phi|$ is substituted by the delta-function $\delta(\phi)$, the propagation of the front is obviously confined by its narrow-band, where $\delta(\phi) > 0$. Without re-initialization to the SDF, the whole process stops.

A more detailed review on related topics is presented in Section 2.

The standard iterative re-initialization algorithm in level set methods is based on the Eikonal equation. It is neither local, nor of a fixed order of accuracy, nor provides any smoothness of the solution. Moreover, it has non-optimal $O(N \log N)$ complexity, may fail for some input function, and requires the same amount of time for each iteration as the proposed direct method for the whole procedure. Thus, our approach has better efficiency and quality characteristics when compared with the standard algorithms.

2. Related Work

Signed distance fields find their applications in a wide range of areas from pure physical and medical simulations to computer graphics and design. Among recent works on the topic we mention collision detection problematic [TKZ*04, BMF05], digital design [FP06], and volume ray tracing [SK00]. Recent GPU implementations (see, for example, [SPG03] and references therein) meet the tremendous demand on fast SDF computation.

Level set methods introduced by Sethian and Osher [OS88, Set99] describe evolution of a hypersurface implicitly given by a level set function. Initially proposed to simulate a motion of an object, modern methods mostly address capturing of shapes rather than tracking. Level set methods are applied to scalar and vector data [Mun07], regular and non-uniform sample sets [MBZW02], topology optimization in hydrodynamics [DMZ08], flow visualization [WJE00], color image segmentation [PSLF07], edge detection [CV01], and to many other problems. We refer to the article by Whitaker [Whi05] for a state-of-the-art overview and to the book by Osher and Fedkiw [OF03] for a thorough exposition of the main aspects of level set methods.

The auxiliary level set function φ should remain close to an SDF to its zero-level set to ensure stability and accuracy of evolution [PMO*99]. It has two important impacts on the whole procedure. First, the function should be initialized as an SDF. Usually, a spherical ansatz is applied, which is in most cases far from the optimal configuration. Second, a re-initialization is regularly performed to bring φ to an SDF during its evolution. For this purpose the Eikonal equation

$$\frac{\partial \psi}{\partial t} = \operatorname{sgn}(\varphi)\,(1 - |\nabla\psi|) \tag{1}$$

is commonly used. The stationary solution ψ_0 is then taken as a new level set function to continue the process. Solving Equation (1) requires significant computational efforts and requires the user to decide when to launch the re-initialization and which time step to choose. Moreover, there

is no guarantee that the process converges nor that it does not distort the isosurface.

The growing popularity of variational level set formulations gave rise for a new approach proposed by Li et al. [LXGF05]. The authors integrated an additional divergence-like term into the governing level set equation entirely eliminating the re-initialization step. The method has been successfully applied to several problems (see, for instance, [PPS09]) but there are still open questions concerning a safe parameter choice and interaction of the new term with other components of the level set equation.

The idea of reducing the number of computations by localization around the isosurface was developed by Adalsteinsson and Sethian [AS95] leading to the so called *narrow-band approach*. Level set function values are computed and updated only in a vicinity of the isocontour and are reinitialized to SDF values only when the curve (hypersurface) is close to the edge of the band. Peng et al. [PMO*99] proposed a localization of the method by Osher and Sethian and included a proper re-initialization based on a marching strategy to the algorithm. The narrow-band approach elevated level set methods to a new grade of efficiency.

Though most level set models are designed for spatial (regular or adaptive) grids, there exist several mesh-free formulations [ZOMK00]. The main sources of unstructured point-based data are sensor measurements and numerical Lagrangian methods. Performed on sample locations level set evolution requires no computational grid, is free of sample-to-grid interpolation errors and profits from natural adaptivity of data. We refer the reader to visualization of scattered volume data [RL06], image segmentation with geometric deformable models [HCLS05] and PDE-based surface extraction from Smoothed Particle Hydrodynamics simulations [RL08].

MLS is an adequate tool to handle spatially scattered data. It provides a technique to approximate not only a value of the sampled field but also its derivatives of arbitrary order. One can distinguish two basic approaches: a fast and simple one by Nayroles et al. [NTV92] and a more accurate but also more expensive one by Belytschko et al. [BLG94] which involves spatial variation of MLS weights into computation. MLS method was applied to volume ray casting [LGM*08], adaptive integration of scanned data [FCS08], statistical implicit representation [OGG09], and various other surface reconstruction algorithms.

Algebraic surface fitting [GG07, GGG08] is of particular interest to our work. The authors utilize the approach by Pratt [Pra87] for up-sampling of point-based surfaces. The key idea is to use the notion of algebraic distance instead of the Euclidean one. The approach generalizes the work by Kolluri [Kol05], where an SDF approximation appears as a side outcome of the reconstruction routine. We refer the reader to the paper by Cheng et al. [CWL*08] for a recent survey on modern MLS methods.

3. Problem Statement

A smooth scalar field $f(\mathbf{x})$ is sampled on a set of points $\mathbf{x}_i$ arbitrarily distributed in $\mathbb{R}^3$. Let Γ be an isosurface with respect to a given value $f_{\text{iso}} \in \mathbb{R}$, i.e., $\Gamma = \{\mathbf{x} \in \mathbb{R}^3 : \ f(\mathbf{x}) = f_{\text{iso}}\}$. The α-band to the isosurface Γ contains all samples $\mathbf{x}_i$ such that the distance from any sample to Γ is less or equal than α with $\alpha > 0$. A scalar function $\varphi(\mathbf{x})$ is said to be a signed distance function to Γ if and only if

$$\varphi(\mathbf{x})\big|_{\mathbf{x}\in\Gamma} = 0, \qquad |\nabla\varphi(\mathbf{x})| = 1. \tag{2}$$

The problem is to develop an effective computational algorithm for explicit approximation of the signed distance function φ to an isosurface Γ within the α-band of width 2α.

4. Algorithm

We propose to perform the following steps to construct a signed distance field around an implicitly given isosurface:

1. Given a scalar field $f(\mathbf{x})$ on samples $\mathbf{x}_i$, extract a set of isopoints $\mathbf{p}_j$ corresponding to the isovalue f_{iso}; estimate the normals $\mathbf{n}_j$ at the isopoints using a kd-tree and the MLS technique. This step can be skipped if the isosurface is given explicitly.
2. For a given $\alpha > 0$ mark all samples $\mathbf{x}_i$ lying in the α-band of the isosurface; simultaneously establish lists of the closest isopoints for the marked samples.
3. For each sample $\mathbf{y}_k$ in the band take the closest isopoints and use the two-angles criterion to test whether multiple isosurface components lie in the sample's neighborhood.
4. Handle the special case when a sample has approximately the same distance to two different isosurface components; compute the distances to both of them as in the subsequent step, compare the distances found, and restrict further processing to the closer of the two isosurface components.
5. In the regular case:
   - perform a local sphere fitting to reconstruct the part of the isosurface close to $\mathbf{y}_k$;
   - if the sphere degenerates to a plane, compute a distance between the sample and the plane (taking into consideration its orientation);
   - otherwise, if the sphere is not degenerate, use the isopoint normals to analyse its convexity and compute the distance between the sample and the sphere.

4.1. Isopoints Extraction

Without loss of generality we assume that $f_{\text{iso}} = 0$. If Γ is given implicitly, we first compute a set of isopoints from the unorganized dataset $\{\mathbf{x}_i, f_i\}$. For that purpose we apply the direct isosurface extraction algorithm proposed by Rosenthal and Linsen [RL06]. The key idea is to collect all pairs of neighboring samples $\mathbf{x}_i$ and $\mathbf{x}_j$, such that the associated scalar field values have different sign, i.e., $f_i \cdot f_j < 0$. Then,

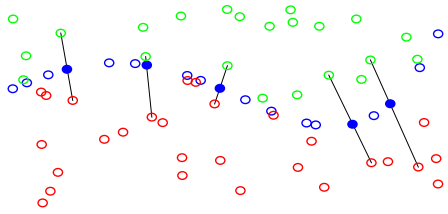

Figure 1: *Direct extraction of isopoints by Rosenthal and Linsen [RL06]. A set of isopoints (blue) is found by means of linear interpolation between neighboring samples with positive (green) and negative (red) scalar values.*

one uses linear interpolation to find an isopoint on the line segment $[\mathbf{x}_i,\ \mathbf{x}_j]$. The idea is illustrated in Figure 1.

4.2. α-band Construction

Having computed the set of isopoints, we mark all samples $\mathbf{x}_i$ within a narrow band of width $2\alpha > 0$ around the isosurface Γ. These are the samples, on which the SDF will be approximated.

For each isopoint $\mathbf{p}_j \in \Gamma$ we find all samples $\mathbf{x}_i$ such that $\|\mathbf{p}_j - \mathbf{x}_i\| < \alpha$. The search is effectively performed using the kd-tree that stores the set of all samples $\{\mathbf{x}_i\}$. Collecting all such neighbors and removing duplicates from the list, we end up with the subset of samples belonging to the narrow band. The approach is illustrated in Figure 2.

For later steps in our processing pipeline, we need to consider the nearest isopoints on surface Γ for each sample in the α-band. For efficiency purposes, we establish those lists of closest isopoints to the samples already during the band construction, as all required information is already given. During construction of the band, we collect all isopoints that have a sample $\mathbf{x}_i$ in their α-neighborhood for the respective

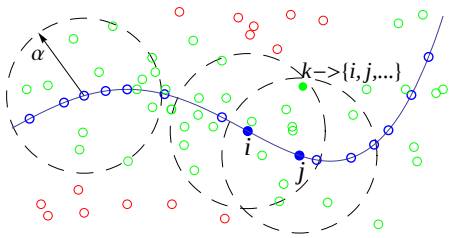

Figure 2: *Narrow band construction on unstructured samples. A circle of radius α around each isopoint (blue) marks samples that belong to the band (green) and neglects the ones that lie outside (red). Simultaneously, we establish a list of closest isopoints for each marked sample.*

sample. Then, we sort the isopoints for each sample $\mathbf{x}_i$ with respect to the distance to $\mathbf{x}_i$.

Altogether, the outcome of the α-band construction step includes a list of sample indices defining which points belong to the band and for each marked sample a sorted list of isopoints lying in its α-neighborhood.

4.3. Spherical Fitting

An accurate computation of the distance between a sample and an isosurface is hard if the isosurface is represented as a sparse set of isopoints. Therefore a (local) reconstruction of the smooth surface is required. In our approach we find an implicit algebraic surface to fit the discrete data which includes isopoint positions and associated normals.

The efficiency of the proposed method is reflected in two characteristics: it is local and direct (i.e., not iterative). The locality is based on an MLS technique for reconstruction of a part of Γ close to the point of interest. The use of algebraic distance allows us to derive the explicit solution of the MLS problem. Additionally, we require the existence of an analytic formula to measure the distance from the reconstructed surface to any point in space. This inevitably restricts the class of solutions but makes the method direct and fast.

We combine *algebraic* and *geometric* approaches in one algorithm [GG07]: the algebraic distance is used for local reconstruction of the isosurface Γ and the Euclidean distance is applied to evaluate the SDF.

Only a few representatives of algebraic surfaces admit analytical evaluation of the signed distance function. Since we aim to avoid any iterative methods, we restrict our consideration to algebraic spheres of the form

$$s(\mathbf{x}) = a_0 + \mathbf{a}\cdot\mathbf{x} + a_4\mathbf{x}\cdot\mathbf{x} \tag{3}$$

with $\mathbf{a} = (a_1, a_2, a_3)$ and $\mathbf{x} = (x_1, x_2, x_3)$. The solution may naturally degenerate to a plane when a_4 vanishes and, therefore, is exact for flat surfaces. However, an extension of the class of fitting surfaces to, for example, ellipsoids is problematic: there exists no analytic formula for the distance to a general second-order manifold. Thus, the class of algebraic spheres is in this sense optimal: it is large enough and still admits direct computation of distances.

We utilize and extend the approach of Guennebaud et al. [GG07, GGG08] to algebraic sphere fitting using positional and derivative constraints. First, we find $m+1$ isopoints nearest to a point of interest $\mathbf{y}$. In most cases we use the list of closest isopoints established during the α-band generation. If the list's length is less than $m+1$, we find further neighbors using the kd-tree that stores the isopoint set Γ.

Let $\mathbf{p}_1, \ldots, \mathbf{p}_{m+1}$ be the $m+1$ closest isopoints ordered such that $\|\mathbf{y}-\mathbf{p}_1\| \le \ldots \le \|\mathbf{y}-\mathbf{p}_{m+1}\|$. We define a parameter h to be equal to the distance from $\mathbf{y}$ to the farthest of

the isopoints, i.e., $h = \|\mathbf{y} - \mathbf{p}_{m+1}\|$. This parameter is used to define the support size of the weighting function

$$\omega_{\mathbf{y}}(\mathbf{p}) = \max\left\{\left(1 - \frac{\|\mathbf{y}-\mathbf{p}\|^2}{h^2}\right)^4, 0\right\}. \quad (4)$$

While the Gaussian kernel is the usual choice for theoretical considerations, the function in Equation (4) is widely used in MLS, as it performs a smooth data approximation and reduces computational efforts due to its compact support and absence of square root or trigonometric function evaluations.

Now we look for the optimal algebraic sphere (3), whose zero-level isosurface $\{\mathbf{x} \in \mathbb{R}^3 : s(\mathbf{x}) = 0\}$ fits positions of isopoints, i.e., $s(\mathbf{p}_j) = 0$, and their normals, i.e., $\nabla s(\mathbf{p}_j) = \mathbf{n}_j$. The best fit is defined by parameters $a_0, \ldots, a_4$ minimizing the cost function

$$E(a_0, \ldots, a_4) = \sum_{j=1}^{m} \omega_j \left[|s(\mathbf{p}_j)|^2 + \beta \|\nabla s(\mathbf{p}_j) - \mathbf{n}_j\|^2\right], \quad (5)$$

where $\omega_j = \omega_{\mathbf{y}}(\mathbf{p}_j)$. The sum has only m terms, since $\omega_{m+1} = 0$ due to Equation (4) and the definition of h. So, the $(m+1)$st isopoint $\mathbf{p}_{m+1}$ is only passively involved in Equation (5). Parameter $\beta \geq 0$ weights the derivative constraints and can be tuned by the user depending on the quality of the underlying data. Based on scaling analysis, Guennebaud et al. [GG07] proposed to choose β to be large compared to the local neighborhood size.

The minimization problem

$$s_{\text{opt}}(\mathbf{x}; a_0, \ldots, a_4) = \arg\min E \quad (6)$$

has the following explicit solution [GGG08]

$$a_4 = \frac{\beta}{2} \frac{\sum \omega_j \mathbf{p}_j \cdot \mathbf{n}_j - \sum \omega_j \mathbf{p}_j \cdot \sum \omega_j \mathbf{n}_j / \Omega}{\sum \omega_j \mathbf{p}_j \cdot \mathbf{p}_j - \sum \omega_j \mathbf{p}_j \cdot \sum \omega_j \mathbf{p}_j / \Omega}, \quad (7a)$$

$$\mathbf{a} = \sum \omega_j \mathbf{n}_j / \Omega - 2a_4 \sum \omega_j \mathbf{p}_j / \Omega, \quad (7b)$$

$$a_0 = \mathbf{a} \cdot \sum \omega_j \mathbf{p}_j / \Omega - a_4 \sum \omega_j \mathbf{p}_j \cdot \mathbf{p}_j / \Omega, \quad (7c)$$

where $\Omega = \sum \omega_j$.

4.4. Stability Improvement

When testing spherical approximation (7) in our algorithm, we found that our computations tend to lose stability and accuracy in certain cases. After a closer look to the formulae and the analysis of possible issues we distinguished three sources of errors: degeneration of weights ω_j, working in absolute coordinates, and inappropriate choice of closest isopoints in case of multiple isosurface components. We have improved the approach to eliminate or alleviate the issues.

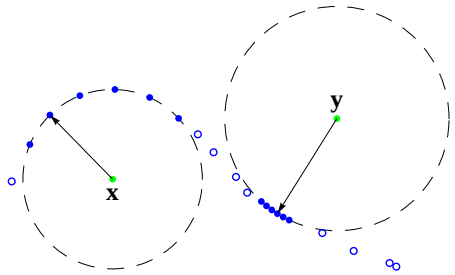

Figure 3: *Vanishing MLS weights due to small differences in distances to neighbors. The nearest neighbors of the sample* $\mathbf{x}$ *lie on a perfect sphere; the isosurface close to the sample* $\mathbf{y}$ *is oversampled.*

Weights. Vanishing smoothing weights ω_j may dramatically affect the stability of calculation, since it leads to large errors in expression (7). The weights degenerate in the following situations:

- $\mathbf{y}$ is far from Γ (not relevant for the α-band approach);
- Γ is oversampled;
- all neighbors of $\mathbf{y}$ are almost equidistant.

In all these cases $\|\mathbf{y} - \mathbf{p}_1\| / \|\mathbf{y} - \mathbf{p}_{m+1}\| \approx 1$ (see Figure3).

Increasing the number $m+1$ of the isopoints considered can help at the cost of additional computational time when Γ is oversampled. The equidistant neighbors' case requires a special processing: trivially, $\mathbf{x}$ is the center of the fitting sphere with the radius equal to $\|\mathbf{x} - \mathbf{p}_1\|$ (see Figure 3).

The general solution we propose to avoid the weight's degeneration is as follows: It is obvious that multiplication of the cost function E with an arbitrary constant positive number does not affect the solution of Equation (6). We redefine the weights involved in expression (5) to be $\widetilde{\omega}_j = \omega_j / \omega_1$, which implies $\widetilde{\omega}_1 = 1$. This normalization can be incorporated into the expression for the weight function (4) such that no new errors appear due to a division of too small quantities.

Coordinate system. It is common for MLS-type methods to work in relative and not in absolute coordinate systems to improve the stability of computation. We utilize this idea and rewrite Equations (7) to the form

$$a_4 = \frac{\beta}{2} \frac{\sum \widetilde{\omega}_j \widetilde{\mathbf{p}}_j \cdot \mathbf{n}_j - \sum \widetilde{\omega}_j \widetilde{\mathbf{p}}_j \cdot \sum \widetilde{\omega}_j \mathbf{n}_j / \widetilde{\Omega}}{\sum \widetilde{\omega}_j \widetilde{\mathbf{p}}_j \cdot \widetilde{\mathbf{p}}_j - \sum \widetilde{\omega}_j \widetilde{\mathbf{p}}_j \cdot \sum \widetilde{\omega}_j \widetilde{\mathbf{p}}_j / \widetilde{\Omega}}, \quad (8a)$$

$$\mathbf{a} = \sum \widetilde{\omega}_j \mathbf{n}_j / \widetilde{\Omega} - 2a_4 \sum \widetilde{\omega}_j \widetilde{\mathbf{p}}_j / \widetilde{\Omega}, \quad (8b)$$

$$a_0 = \mathbf{a} \cdot \sum \widetilde{\omega}_j \widetilde{\mathbf{p}}_j / \widetilde{\Omega} - a_4 \sum \widetilde{\omega}_j \widetilde{\mathbf{p}}_j \cdot \widetilde{\mathbf{p}}_j / \widetilde{\Omega}, \quad (8c)$$

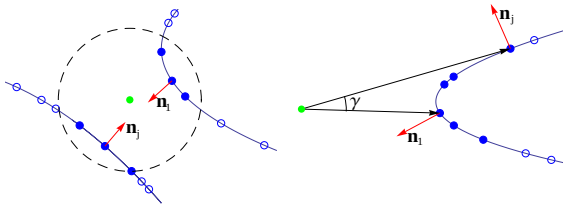

Figure 4: *Accurate choice of neighbors. Sheet separation (left) is characterized by large angle between normals of neighbors (red vectors). To distinguish sharp features (right) we use an additional criterion. Neighbor isopoints of a sample (green disc) have normals with large angle discrepancy (red arrows). However we do not treat it as sheet layer, since the angle between the pointing vectors γ is small.*

where $\widetilde{\mathbf{p}}_j = \mathbf{p}_j - \mathbf{y}$ and $\widetilde{\Omega} = \sum \widetilde{\omega}_j$. Thus, we shift the origin to the position of $\mathbf{y}$.

Neighbors. If an isosurface Γ has multiple components, the correct choice of neighboring isopoints may fail. The set of the $m+1$ closest isopoints may contain points from different parts of the isosurface. Obviously, this constellation leads to undesired spherical fitting (see Figure 4 (left)) and consequently to wrong signed distance evaluation.

If the above-mentioned situation occurs, there exists an index $2 \le j \le (m+1)$ such that $\mathbf{n}_1 \cdot \mathbf{n}_j < \delta$ for some threshold $-1 \le \delta \le 1$. In our tests we use $\delta = 0$. In fact, since the underlying scalar field $f(\mathbf{x})$ is assumed to be continuous, the normals of all components of Γ should be uniformly point to the inside of the volumes bounded by the isosurfaces or to the outside. Thus, we perform a simple test on the orientations of the normals to check whether $\mathbf{y}$ lies between multiple isosurface components. If the test is positive, we adjust the selection of the isopoints by only taking points belonging to one of the isosurface components.

An additional issue that needs to be resolved is that of distinguishing sharp features (see Figure 4 (right)). In addition to the angle between the normals at the isopoints, we consider the angle γ between vectors pointing from the sample to the isopoints. Although the former angle may be large, the latter angle is small. Thus, all neighbors should be considered as belonging to the same isosurface component.

If $\mathbf{n}_1 \cdot \mathbf{n}_j < \delta$ for some j and γ is large, we execute two different algebraic sphere fittings. For the first one, we use $m+1$ nearest neighbors of $\mathbf{p}_1$ (including the isopoint itself), whereas for the second fitting, we use the $m+1$ nearest neighbors of $\mathbf{p}_j$. Both approximations will be used to estimate the distances to the point $\mathbf{y}$ in the next step. The minimal distance with a correct sign will be accepted as the value of SDF.

4.5. SDF Approximation

Plane. The algebraic sphere fitting can degenerate to a plane fitting when a_4 vanishes. In this case we can simply compute the distance between the sample $\mathbf{y}$ and the plane. This is done by

$$\varphi(\mathbf{y}) = \frac{a_0}{\sqrt{\mathbf{a} \cdot \mathbf{a}}}. \tag{9}$$

Recall that the coefficients a_i are found in the relative system of coordinates, where $\mathbf{y} = \mathbf{0}$. Since the surface part close to $\mathbf{y}$ is (nearly) flat, we can estimate the gradient and the curvature of the SDF by

$$\nabla\varphi(\mathbf{y}) = \mathbf{a}, \qquad \triangle\varphi(\mathbf{y}) = 0.$$

Sphere. As soon as the best fitting sphere is found, it is not difficult to compute the distance between $\mathbf{y}$ and $\{\mathbf{x} \in \mathbb{R}^3 : s_{\text{opt}}(\mathbf{x}) = 0\}$. For that purpose we use the following formulae for the center $\mathbf{c}$ and the radius r of the sphere:

$$\mathbf{c} = -\frac{1}{2a_4}\mathbf{a}, \qquad r = \sqrt{\frac{\mathbf{c} \cdot \mathbf{c}}{4a_4^2} - \frac{a_0}{a_4}}. \tag{10}$$

Then the unsigned distance is equal to $\|\mathbf{y} - \mathbf{c}\| - r$. However, for the *signed* distance we should take into account the sign of the free coefficient a_0. Finally, using $\mathbf{y} = \mathbf{0}$ we derive the SDF approximation expression

$$\varphi(\mathbf{y}) = \operatorname{sgn}(a_0)\left(\sqrt{\mathbf{c} \cdot \mathbf{c}} - r\right). \tag{11}$$

The gradient and the Laplacian of φ are estimated as

$$\nabla\varphi(\mathbf{y}) = -\frac{\operatorname{sgn}(a_0)}{\sqrt{\mathbf{c} \cdot \mathbf{c}}}\mathbf{c}, \qquad \triangle\varphi(\mathbf{y}) = \frac{\operatorname{sgn}(a_0)}{r}.$$

Note that for the expression above we considered the weights $\widetilde{\omega}_j$ to be constant. The approach by Belytschko et al. [BLG94] differs from one by Nayroles et al. [NTV92] exactly in accurate handling of spatial derivatives of the weights.

In Figure 5 the shortest paths from samples $\mathbf{x}$ and $\mathbf{y}$ are shown as black arrows. Formula (11) fails when applied to samples $\mathbf{z}$ and $\mathbf{v}$, since the red vectors do not point to the isosurface. Thus, for samples lying on the concave side of the isosurface at a larger distance to the isosurface we can not apply Formula (11). Instead, we use the planar approximation

$$\varphi(\mathbf{y}) = -\sum \widetilde{\omega}_j \widetilde{\mathbf{p}}_j \cdot \sum \widetilde{\omega}_j \mathbf{n}_j. \tag{12}$$

The position and orientation of the plane is given by the MLS *plane* fitting. Consequently, we get the gradient and the curvature as

$$\nabla\varphi(\mathbf{y}) = \sum \widetilde{\omega}_j \mathbf{n}_j, \qquad \triangle\varphi(\mathbf{y}) = 0.$$

There is no guarantee that the SDF approximation φ remains smooth when moving from $\mathbf{x}$ to $\mathbf{y}$. Therefore, we restrict the size of the α-band by maximal curvature radius of the isosurface Γ.

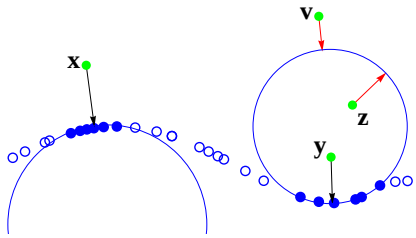

Figure 5: *Calculation of a distance to the isosurface reconstructed by means of non-degenerate spheres. Distances for the samples* **x** *and* **y** *are found correctly. To estimate the SDF at positions* **z** *and* **v** *a planar approximation is used instead.*

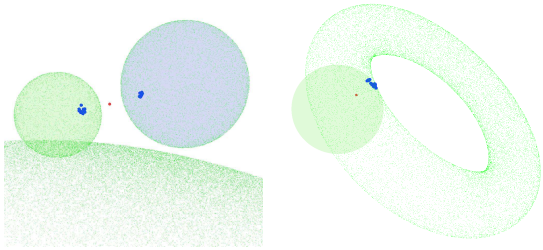

Figure 6: *Left: a sample (red) lying in a layer between two spheres and a plane has its nearest neighbors (blue) belonging to different objects (test 2). Thus, two fitting spheres were generated (light blue and light green) to detect the closest isosurface. Right: Fitting a sphere for a torus represented as a set of randomly distributed points (test 3). The sphere (light green) is found using positions and normals of isopoints (blue) close to the point of interest (red).*

5. Numerical Tests

All numerical tests presented in this section were performed on a PC with an Intel Xeon 3.20GHz processor.

5.1. Accuracy and Efficiency

We have performed several tests to investigate the accuracy and the efficiency of the proposed SDF approximation with respect to the initial isosurface profile, the quality of the isosurface discretization, the number of samples at which an SDF is computed, and the number of neighbors taken into account in the fitting procedure.

First, we compare the accuracy and efficiency of the SDF evaluation of three methods: the naive closest neighbor distance

$$\varphi_0(\mathbf{y}) = (\mathbf{y}-\mathbf{p}_1)\cdot\mathbf{n}_1,$$

an SDF similar to one used in Kolluri's surface up-sampling algorithm [Kol05]

$$\varphi_1(\mathbf{y}) = \sum_{j=1}^{20} \omega_j(\mathbf{y})\ (\mathbf{y}-\mathbf{p}_j)\cdot\mathbf{n}_j \Big/ \sum_{k=1}^{20} \omega_k(\mathbf{y}),$$

(here we use our weighting function (4)), and the proposed approach with $m = 24$. It is worth to mention that all three methods converge to the correct SDF when the number of isopoints grows to infinity.

For our analysis, we constructed SDF approximations in a neighborhood of the following surfaces:

1. A perfect sphere of radius 30 discretized with 10000 points uniformly distributed over the surface. The α-band has the width of 20 length units and contains about $630k$ samples.
2. A plane and two perfect spheres of radii 30 and 20 length units discretized with $10k$ and $5k$ uniformly sampled points, correspondingly. The α-band with $\alpha = 14$ consists of $860k$ points and covers the space between all three objects.
3. A torus with outer radius of 40 and inner radius of 10 length units. The surface is represented as $10k$ uniformly distributed points. The SDFs are approximated in a band with $\alpha = 5$ containing $570k$ samples.

Since the exact SDFs for all these examples are given by analytical formulae, it is possible to compute the average and maximum error of the SDF approximation for each of the methods. Two examples of fitting a sphere for a torus and a multi-object example are presented in Figure 6. Numerical results are presented in Table 1.

The entries in Table 1 show that our method is exact for planar and spherical surfaces and properly treats multi-component objects. We expect our method to be one order of magnitude more accurate for arbitrary isosurfaces. Most of the computational time is spent to find the nearest isopoints, which we can deduce from the comparison of t and τ. Thus, the proposed non-iterative SDF approximation is comparable in efficiency with other explicit methods.

To estimate the dependence of the performance on the number of samples N and in the number of neighbors m we run a series of tests varying these parameters. The results presented in Table 2 show that the computational time t depends linearly on N and slightly superlinearly on m.

5.2. SDF for Explicit Isosurfaces

In Figures 7 and 8 several level surfaces of SDF are shown. The SDFs are constructed in vicinities of manifolds explicitly given by a number of isopoints together with their normals. For our tests we used datasets representing a skeleton hand ($327k$ surface points, courtesy of Stereolithography Archive, Clemson University), a bunny ($35k$ surface points, courtesy of the Stanford University Computer Graphics Laboratory) and a dragon ($437k$ surface points, courtesy of the Stanford University Computer Graphics Laboratory). The

test 1	φ_0	φ_1	φ
avrε	$7.7 \cdot 10^{-3}$	0.0271	$2.2 \cdot 10^{-15}$
maxε	0.1422	0.1925	$1.4 \cdot 10^{-14}$
t [s]	0.3	1.13	2.03
τ [s]	0.3	0.57	1.2
test 2	φ_0	φ_1	φ
avrε	$1.9 \cdot 10^{-3}$	$6.6 \cdot 10^{-3}$	$5.2 \cdot 10^{-14}$
maxε	0.1094	0.1345	$3.4 \cdot 10^{-12}$
t [s]	0.34	1.47	2.73
τ [s]	0.34	0.72	1.66
test 3	φ_0	φ_1	φ
avrε	0.0141	0.0488	$2.8 \cdot 10^{-3}$
maxε	0.4157	0.3577	0.0905
t [s]	0.69	3.69	5.62
τ [s]	0.69	0.86	1.49

Table 1: *Accuracy and efficiency of SDF approximations by naive approach φ_0, weighted sum formula φ_1, and our proposed algorithm φ. Average and maximal error ε is measured for three test isosurfaces with uniformly sampled points: a sphere (test 1), two spheres and a plane (test 2) and a torus (test 3). The efficiency is characterized by the time of SDF computation (1) with the lists of closest isopoints from the α-band construction: t, and (2) with the pre-computed 24 closest isopoints for all α-band samples: τ.*

	$N = 150k$	$N = 300k$	$N = 600k$
$m = 10$	$0.42s$	$0.84s$	$1.71s$
$m = 20$	$0.67s$	$1.31s$	$2.59s$
$m = 30$	$1.33s$	$2.45s$	$4.79s$
$m = 40$	$2.22s$	$4.67s$	$8.75s$

Table 2: *Computational time required for the SDF approximation depending on sample number N and number of neighboring isopoints m (the α-band construction time is not accounted). The test is performed for $40k$ isopoints sampled uniformly on a perfect torus.*

images were generated with image-space post-processing of point set data to fill holes in the rendering. The results show a high-quality SDF construction close to convex parts of the initial isosurface; concave regions and sharp features cause minor not visible irregularities in the signed distance field.

5.3. SDF for Implicit Isosurfaces

If a volumetric scalar field is given in form of scattered samples, we first apply the direct isopoints extraction algorithm as described in Section 4.1. The extracted isopoints determine the position of the isosurface and can be used for a sampling of additional points with no function values in the α-band around Γ. When the SDF has been approximated onto the dense point cloud in the α-band, we extract an isosurface due to the SDF scalar field. The discretization of Γ has a better quality and can subsequently been used for point-based visualization of the manifold. In Figure 9, we show the results of dense isopoint set extraction on the example of White Dwarf dataset with $500k$ points. The data come from a Smoothed Particle Hydrodynamics astrophysical simulation.

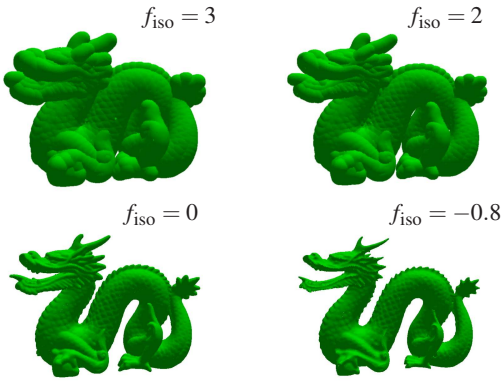

Figure 7: *Several isosurfaces of the SDF field constructed for the dragon dataset with $437k$ surface points.*

6. Results, Discussion, and Conclusion

The proposed method for non-iterative approximation of SDF to arbitrary isosurface is based on the MLS fitting of algebraic sphere using positional and derivative constraints. The class of fitting surfaces includes non-degenerate and degenerate spheres (planes) that allows for a proper handling of both curved and (nearly) flat surfaces. Since the Euclidean distance from a spatial point to an algebraic sphere can be computed exactly, the SDF to the isosurface can be approximated fast and accurately. The smoothness of the approximating field is controlled by the MLS weighting function in a vicinity of the isosurface. A sufficiently large safe width α of the vicinity can be determined by the radius of the maximal curvature on the manifold. The method properly handles surfaces with many components by correct choice of closest isopoints. The dependence of efficiency of the algorithm on the number of samples N and the number of closest isopoints m is linear and slightly superlinear, correspondingly.

To fairly compare the proposed method for a non-iterative SDF approximation with methods based on Equation (1) is difficult. The efficiency of iterative methods depends on deviation of a given function from the desired SDF. Depending on that deviation, a φ may be found in a few iterations, it may require many iterations, or the algorithm may fail. The explicit approach takes approximately the same amount of computational time as one iteration with respect to Equation (1) and allows us to simultaneously approximate the derivatives of the SDF. In the context of level-set methods,

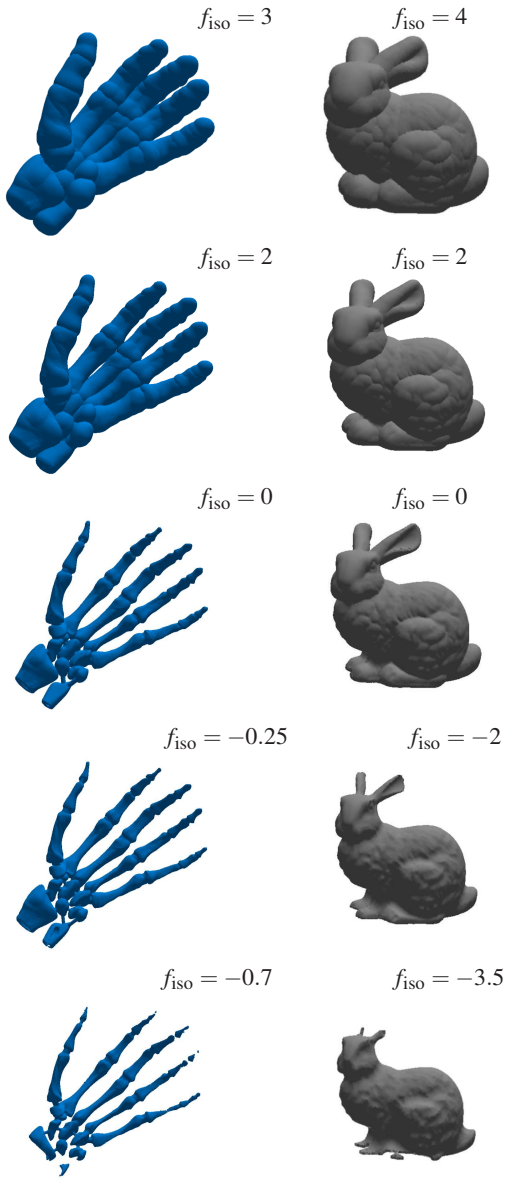

Figure 8: *Several isosurfaces of the SDF fields constructed for the skeleton hand (327k surface points) and the bunny (35k surface points) datasets.*

values of $\nabla\varphi$ are needed for re-initialization (as well as for most governing level-set equations), which makes our explicit method significantly more efficient.

The choice of the support size h in our algorithm is adaptive: h equals to the distance from a sample to its $(m+1)$st closest isopoint. In contrast to the constant h in various surface reconstruction algorithms [GG07], it bounds the number of operations when Γ is oversampled and ensures a stable computation near undersampled regions.

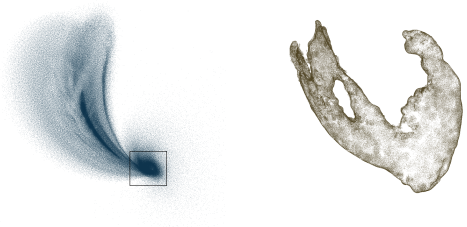

Figure 9: *Left: The White Dwarf dataset with* 500*k data points. Right: The samples on the isosurface with* $f_{\text{iso}} = 0.0001$ *extracted from the signed distance field approximated in a dense* α*-band with about* 5*M samples. (Data set courtesy of Stephan Rosswog, Jacobs University)*

The use of the proposed SDF construction yields the following improvements in well-known techniques:

- It is possible to initialize a level-set function as an accurate SDF to an almost arbitrary initial surface. Definitely, this saves a lot of computational efforts and time if the surface is chosen according to a priori knowledge about the object to be extracted.
- Improvement in construction of Voronoi diagram and Delaunay triangulation/tetrahedrization of point sets can be performed via preliminary reconstruction of signed distance field.

7. Future Work

The proposed method can be enhanced in multiple directions.

- Point density weights (as in [GGG08]) can be used to overcome irregularities in isosurface discretization. Alternatively an explicit regularization of isopoints positions can be fulfilled.
- Improvement of the algorithm for more accurate fitting in the presence of sharp features is desired.
- Optimization of the number of neighbors for the MLS procedure and thresholds for the angle criteria can affect the quality of the SDF approximation. These parameters should be adaptive with respect to local curvature of the isosurface and density of isopoints in the neighborhood.
- Since an algebraic sphere is not able to fit other second-order surfaces exactly, a cylinder fitting could be easily implemented to enrich the class of fitting surfaces.
- Tests on accuracy of gradient and curvature estimation should be performed. Accurate consideration of spatial derivatives of MLS weights is required.

Acknowledgements

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

References

[AS95] ADALSTEINSSON D., SETHIAN J. A.: A fast level set method for propagating interfaces. *Journal of Computational Physics 118* (1995), 269–277.

[BLG94] BELYTSCHKO T., LU Y. Y., GU L.: Element-free Galerkin methods. *Int. J. Numer. Meth. Engng 37* (1994), 229–256.

[BMF05] BRIDSON R., MARINO S., FEDKIW R.: Simulation of clothing with folds and wrinkles. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses* (New York, NY, USA, 2005), ACM, p. 3.

[CV01] CHAN T. F., VESE L. A.: Active contours without edges. *Image Processing, IEEE Transactions on 10*, 2 (2001), 266–277.

[CWL*08] CHENG Z.-Q., WANG Y.-Z., LI B., XU K., DANG G., JIN S.-Y.: A survey of methods for moving least squares surfaces. *Proceedings of Point Based Graphics* (2008).

[DMZ08] DUAN X.-B., MA Y.-C., ZHANG R.: Shape-topology optimization for Navier-Stokes problem using variational level set method. *J. Comput. Appl. Math. 222*, 2 (2008), 487–499.

[FCS08] FIORIN V., CIGNONI P., SCOPIGNO R.: Practical and robust MLS-based integration of scanned data, 2008.

[FP06] FRISKEN S. F., PERRY R. N.: Designing with distance fields. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses* (New York, NY, USA, 2006), ACM, pp. 60–66.

[GG07] GUENNEBAUD G., GROSS M.: Algebraic point set surfaces. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers* (New York, NY, USA, 2007), ACM, p. 23.

[GGG08] GUENNEBAUD G., GERMANN M., GROSS M. H.: Dynamic sampling and rendering of algebraic point set surfaces. *Comput. Graph. Forum 27*, 2 (2008), 653–662.

[HCLS05] HO H. P., CHEN Y., LIU H., SHI P.: Point-based geometric deformable models for medical image segmentation. In *MICCAI* (2005), pp. 278–285.

[Kol05] KOLLURI R.: Provably good moving least squares. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms* (Aug. 2005), pp. 1008–1018.

[LGM*08] LEDERGERBER C., GUENNEBAUD G., MEYER M., BÄCHER M., PFISTER H.: Volume MLS ray casting. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1372–1379.

[LXGF05] LI C., XU C., GUI C., FOX M. D.: Level set evolution without re-initialization: a new variational formulation. pp. 430–436 vol. 1.

[MBZW02] MUSETH K., BREEN D. E., ZHUKOV L., WHITAKER R. T.: Level-set segmentation from multiple non-uniform volume datasets. *Visualization Conference, IEEE* (2002).

[Mun07] Curve/surface representation and evolution using vector level sets with application to the shape-based segmentation problem. *IEEE Trans. Pattern Anal. Mach. Intell. 29*, 6 (2007), 945–958. Abd El Munim, Hossam E. and Farag, Aly A.

[NTV92] NAYROLES B., TOUZOT G., VILLON P.: Generalizing the finite element method: Diffuse approximation and diffuse elements. *Comput. Mech. 10* (1992), 307–318.

[OF03] OSHER S., FEDKIW R.: *Level set methods and dynamic implicit surfaces*. Springer, 2003.

[OGG09] OZTIRELI C., GUENNEBAUD G., GROSS M.: Feature preserving point set surfaces based on non-linear kernel regression, 2009.

[OS88] OSHER S., SETHIAN J. A.: Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys. 79*, 1 (1988), 12–49.

[PMO*99] PENG D., MERRIMAN B., OSHER S., ZHAO H., KANG M.: A PDE-based fast local level set method. *J. Comput. Phys. 155*, 2 (November 1999), 410–438.

[PPS09] PENG Y., PI L., SHEN C.: A semi-automatic method for burn scar delineation using a modified Chan-Vese model. *Comput. Geosci. 35*, 2 (2009), 183–190.

[Pra87] PRATT V.: Direct least-squares fitting of algebraic surfaces. *SIGGRAPH Comput. Graph. 21*, 4 (1987), 145–152.

[PSLF07] PI L., SHEN C., LI F., FAN J.: A variational formulation for segmenting desired objects in color images. *Image Vision Comput. 25*, 9 (2007), 1414–1421.

[RL06] ROSENTHAL P., LINSEN L.: Direct isosurface extraction from scattered volume data. In *Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization* (2006), pp. 99–106.

[RL08] ROSENTHAL P., LINSEN L.: Smooth surface extraction from unstructured point-based volume data using PDEs. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1531–1546.

[Set99] SETHIAN J. A.: *Level Set Methods and Fast Marching Methods*, second edition ed. Cambridge University Press, Cambridge, UK, 1999.

[SK00] SRAMEK M., KAUFMAN A.: Fast ray-tracing of rectilinear volume data using distance transforms. *IEEE Transactions on Visualization and Computer Graphics 6*, 3 (2000), 236–252.

[SPG03] SIGG C., PEIKERT R., GROSS M.: Signed distance transform using graphics hardware. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)* (Washington, DC, USA, 2003), IEEE Computer Society, p. 12.

[TKZ*04] TESCHNER M., KIMMERLE S., ZACHMANN G., HEIDELBERGER B., RAGHUPATHI L., FUHRMANN A., CANI M.-P., FAURE F., MAGNETAT-THALMANN N., STRASSER W.: Collision detection for deformable objects, 2004.

[Whi05] WHITAKER R.: Isosurfaces and level-sets. In *The Visualization Handbook*, Hansen C., Johnson C., (Eds.). Elsevier, 2005, pp. 97–123.

[WJE00] WESTERMANN R., JOHNSON C., ERTL T.: A level-set method for flow visualization. In *VIS '00: Proceedings of the conference on Visualization '00* (Los Alamitos, CA, USA, 2000), IEEE Computer Society Press, pp. 147–154.

[ZOMK00] ZHAO H.-K., OSHER S., MERRIMAN B., KANG M.: Implicit and non-parametric shape reconstruction from unorganized points using variational level set method. *Computer Vision and Image Understanding 80* (2000), 295–319.