COUPLED SIGNED-DISTANCE FUNCTIONS FOR IMPLICIT SURFACE RECONSTRUCTION

Yuanhao Gong, Grégory Paul, Ivo F. Sbalzarini

MOSAIC Group, Institute of Theoretical Computer Science, ETH Zurich, Switzerland.
SIB Swiss Institute of Bioinformatics, Zurich, Switzerland.

ABSTRACT

We present a coupled signed-distance function method for reconstructing closed implicit surfaces from unstructured point clouds. The method can capture high curvature without the need for adaptive grids and is easy to implement. We present the method, benchmark it on artificial data, and apply it to two biological point data sets from protein surfaces and PALM microscopy.

***Index Terms*—** Surface reconstruction, point cloud, coupled signed-distance function, coupled level set, PALM

1. INTRODUCTION

Surface reconstruction from unstructured point clouds is a fundamental problem in diverse fields such as computer graphics, computer vision, fluid dynamics, molecular modeling, microscopy, etc. The problem is challenging due to the absence of connectivity information between the points, which may lead to topological ambiguities. Moreover, high curvature (sharp corners or edges) and noise in the point positions often complicate the task.

A wealth of methods have been proposed to address these issues; we briefly review them in Sec. 1.1. We then propose a new implicit surface reconstruction method called *coupled signed-distance functions* (cSDF). It is inspired by coupled level-set methods [1, 2, 3], but addresses some of their shortcomings when reconstructing high-curvature regions, while guaranteeing the signed-distance property.

1.1. Prior Works and Motivation

Numerous methods have been proposed for surface reconstruction from unstructured point clouds. This includes approaches based on FFTs (e.g., [4]), Poisson surfaces (e.g., [5]), and moving least squares (e.g., [6]). Level-set methods [7, 8] minimize the p-norm of the distance field $d(\cdot)$ from the implicit surface Γ. The same objective function can also be minimized using Bregman iterations, leading to very efficient algorithms [9]. In this energy-based approach, methods from image segmentation have also been adapted to surface reconstruction [10]. Thanks to the convexity of the energy, fast solvers (e.g., split-Bregman or Primal/Dual) can be used. The regularization term in a segmentation model, however, tends to smooth the result and remove details from the surface. Moreover, the computational cost depends on proper initialization and step-size control. These issues are avoided with cSDF.

Prior work also demonstrated several strategies to save memory and reduce the computational complexity of these algorithms. This includes narrow-band formulations [11], multi-scale methods [12, 13], and DT-grids [14]. The need for computationally expensive level-set re-initialization has been reduced by adding an additional penalty term in the energy [15].

1.2. Properties of cSDF

The contributions of the present method are:

1. cSDF do not require (pseudo-)time evolution of a PDE. This avoids initialization, reinitialization, and the stability condition for time stepping.
2. The result from cSDF is a signed distance function.
3. cSDF requires no estimation of normals. Approaches such as the Poisson surface method require consistent normal estimation. This is avoided in cSDF by a two-layer labeling scheme.
4. cSDF can capture high-curvature regions without requiring adaptive grids.

2. METHOD

Let the point cloud $S = \{x_i : x_i \in \mathbb{R}^n, i = 1, \ldots, N\}$. We first illustrate cSDF in 2D and then apply it to synthetic and real-world data in 3D. The method progresses in three steps as detailed by Algorithms 1 to 3 below. The corresponding steps are illustrated as ①, ②, and ③ in Fig. 1.

2.1. Step 1: Distance Field

For a given point cloud S, we compute the distance field $d(x)$ on a predefined Cartesian grid $G = U \times V$ of uniform resolu-

Funding: Swiss National Science Foundation, grant CRSII3-132396/1.

tion h. This amounts to solving the following Eikonal equation:

$$\begin{cases} \|\nabla d(x)\| = 1 & \forall x \in G \\ \text{s.t. } d(x_i) = 0 & \forall x_i \in S \end{cases} \tag{1}$$

as a boundary-value formulation of the corresponding Hamilton-Jacobi problem.

Several methods are available to numerically solve this equation, including the Fast Marching Method [11], the Group Marching Method [16], the Fast Sweeping Method (FSM) [17], the Fast Iteration Method [18], and direct Hamilton-Jacobi solvers [19]. Here, we use an extended-window FSM restricted to a narrow band of width b:

$$N_\mathrm{b} = \{x \in G : \exists x_i \in S \; s.t. \; ||x_i - x|| < b\} \,. \tag{2}$$

2.1.1. *Extended-window Fast Sweeping Method*

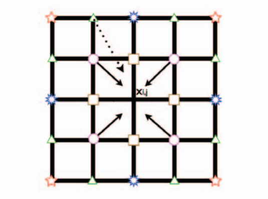
FSM sweeps the grid until convergence, which can be inefficient for points far from the interface. Fast iterative methods relax this by using locks [18]. These locks, however, cause additional serialization. Here, we accelerate FSM by using a larger window size $w > 1$ (see Algorithm 1). Fig. 3(b) shows an example $d(x)$ computed using FSM with $w = 3$. The original FSM [17] is recovered for $w = 1$. While an extended window can significantly reduce the iteration count, the local update cost increases from 2 to $(w + 1)(w + 2)/2 - 1$. We initialize the algorithm with:

$$\begin{cases} d(x) = +\infty & \forall x \in N_\mathrm{b} \setminus S \\ d(x_i) = 0 & \forall x_i \in S \,. \end{cases} \tag{3}$$

Algorithm 1 Extended-window fast sweeping method in 2D

```
1: INPUT: threshold $tol$, window size $w$, $S$, $U$, $V$
2: set $w^{+1} = w + 1$
3: initialize $d_{k+1}(x)$ using Eq. 3
4: define the loop sets
     $\{(i,j) : i = w^{+1} \dots U - w, j = w^{+1} \dots V - w\}$,
     $\{(i,j) : i = U - w \dots w^{+1}, j = w^{+1} \dots V - w\}$,
     $\{(i,j) : i = U - w \dots w^{+1}, j = V - w \dots w^{+1}\}$,
     $\{(i,j) : i = w^{+1} \dots U - w, j = V - w \dots w^{+1}\}$
5: while $\max\{|d_{k+1}(x) - d_k(x)|\} > tol$ do
6:     go through the loop sets and do
       $d_{k+1}(x) = \mathrm{Update}^\mathrm{w}(d_k(x))$
7: end while
8: $d_k(x) = \sqrt{d_{k+1}(x)}$
9: OUTPUT: $d_k(x)$
```

The $\mathrm{Update}^\mathrm{w}$ of FSM [17] is extended to account for all points in a w-neighborhood, as illustrated on the right.



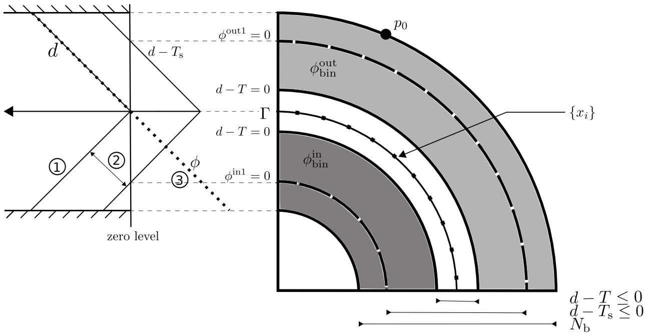


Fig. 1. Illustration of Algorithms 1 to 3

2.2. Step 2: Coupled Signed-Distance Functions

We aim to compute $\phi(x)$, the signed-distance function associated with $d(x)$. The key idea of cSDF is to apply distance-preserving shift transformations to the output of Algorithm 1, thus solving the boundary-value problem in Eq. 1 without (pseudo-)time evolution. Specifically, we shift d by an offset T in order to determine the functions $\phi^\mathrm{in}_\mathrm{bin}$ and $\phi^\mathrm{out}_\mathrm{bin}$ that indicate whether the shifted level set $d - T_\mathrm{s}$ is inside or outside of Γ (see shaded areas in Fig. 1). Algorithm 2 first computes these bands for d shifted down by T_s (i.e., the level sets ϕ^in1 and ϕ^out1) and then for the function $d - T_\mathrm{s}$ shifted up again, yielding ϕ^in2 and ϕ^out2. The threshold T defines the separation between the regions to be labeled.

Algorithm 2 cSDF construction

```
1: INPUT: threshold $T$, $T_\mathrm{s}$, $d(x)$
2: $d_0(x) = d(x) - T$
3: select any point $p_0$ on the outer boundary of the narrow band.
4: starting from $p_0$, label as $\phi^\mathrm{out}_\mathrm{bin}$ the connected component where
   $d_0 > 0$; label the rest of the region where $d_0 > 0$ as $\phi^\mathrm{in}_\mathrm{bin}$.
5: $d_1(x) = d(x) - T_\mathrm{s}$
6: compute $\phi^\mathrm{in1}$ and $\phi^\mathrm{out1}$ using Algorithm 1 on $\phi^\mathrm{in}_\mathrm{bin}$ and $\phi^\mathrm{out}_\mathrm{bin}$,
   respectively, with input $d_1(x)$
7: $\phi^\mathrm{in2} = \phi^\mathrm{in1} - T_\mathrm{s}$, $\phi^\mathrm{out2} = \phi^\mathrm{out1} - T_\mathrm{s}$
8: OUTPUT: $\phi^\mathrm{in2}$ and $\phi^\mathrm{out2}$
```

2.3. Step 3: Surface Reconstruction using cSDF

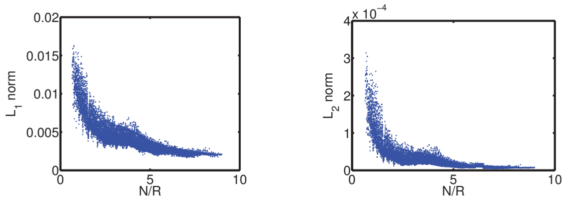
After computing ϕ^in2 and ϕ^out2, a joint estimation of the signed-distance function ϕ of the reconstructed surface Γ is computed from ϕ^in2, ϕ^out2, and $d(x)$ as described in Algorithm 3.

3. RESULTS

We demonstrate cSDF on 2D and 3D benchmarks and show its application to real-world data.

Algorithm 3 Surface reconstruction using cSDF

```
INPUT: $\phi^{\mathrm{in}2}$, $\phi^{\mathrm{out}2}$, $d$
$d^{\mathrm{in}}_{\mathrm{out}} = ||\phi^{\mathrm{in}2}| - |\phi^{\mathrm{out}2}||$,
$d^{\mathrm{in}}_{\mathrm{edge}} = ||\phi^{\mathrm{in}2}| - d|$, $d^{\mathrm{out}}_{\mathrm{edge}} = ||\phi^{\mathrm{out}2}| - d|$
for all $x \in N_b$ do
    $t = \min\{d^{\mathrm{in}}_{\mathrm{out}}, d^{\mathrm{in}}_{\mathrm{edge}}, d^{\mathrm{out}}_{\mathrm{edge}}\}$
    if $d^{\mathrm{in}}_{\mathrm{out}} == t$ then $\phi = (\phi^{\mathrm{in}2} - \phi^{\mathrm{out}2})/2$
    if $d^{\mathrm{in}}_{\mathrm{edge}} == t$ then $\phi = -\phi^{\mathrm{in}2}$
    if $d^{\mathrm{out}}_{\mathrm{edge}} == t$ then $\phi = \phi^{\mathrm{out}2}$
end for
OUTPUT: $\phi$
```



(a) L_1-error vs. N/R (b) L_2-error vs. N/R

Fig. 2. Reconstruction errors for point clouds on circles

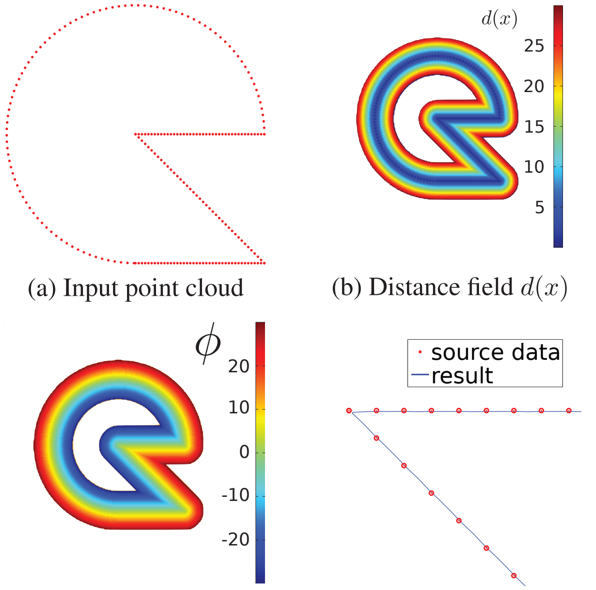
3.1. 2D Benchmarks

We test the accuracy of cSDF by sampling N points uniformly on a circle of radius R and comparing the reconstructed circle to the ground truth for decreasing N. We use a 200^2 grid for all $N \in [45, 360] \times R \in [40, 70]$. We linearly interpolate the resulting ϕ^{c} at each of the original $x_i \in S$. The correct value would be $\phi^{\mathrm{c}} = 0$ for all x_i. We hence compute the overall (reconstruction plus interpolation) mean L_1 and L_2 errors as $\sum_{i=1}^{i=N} |\phi^{\mathrm{c}}(x_i)|/N$ and $\left(\sum_{i=1}^{i=N} \phi^{\mathrm{c}}(x_i)^2/N\right)^{1/2}$, respectively. The result is shown in Fig. 2. The memory usage of cSDF is $3|N_{\mathrm{b}}|$.

Figure 3 shows a synthetic example with sharp corners to illustrate cSDF's capability of representing them without introducing excessive surface smoothing.

3.2. 3D Benchmarks

We benchmark cSDF in 3D by using the vertices of the triangulated surfaces of the well-known computer graphics models "Armadillo" and "Buddha" as input point clouds. The number of points for each model, the CPU time for cSDF reconstruction of the implicit surface representation, and the resulting errors against the known ground truth at the vertex positions are given in Table 1. The code is implemented in C and run on a 2 GHz Intel Core i7. Figure 4 shows the resulting reconstructions and close-ups with the input point cloud overlaid to demonstrate the method's capability of representing high-curvature regions ("Armadillo" claws) without grid refinement (Fig. 4(b) and (d)).



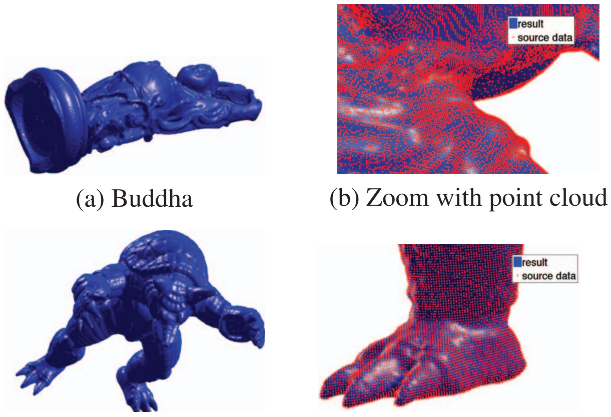
(a) Input point cloud (b) Distance field $d(x)$

(c) Result ϕ (d) Zoom at a sharp corner

Fig. 3. 2D benchmark example with sharp corners

Surface	# Points	Grid	CPU time	L_1 — error	L_2 — error
Armadillo	172 974	175 × 208 × 159	39.9 s	0.0523	0.0755
Buddha	543 652	82 × 199 × 82	32.6 s	0.0673	0.1122
Protein	3 511	66 × 66 × 66	2.9 s	–	–
PALM	582	400 × 400	0.0056 s	–	–

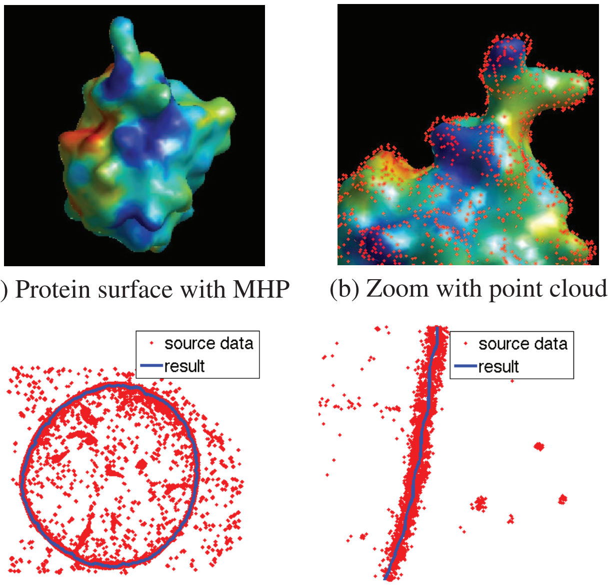
Table 1. cSDF benchmarks ($b = 6$ in all cases).



(a) Buddha (b) Zoom with point cloud

(c) Armadillo (d) Zoom with point cloud

Fig. 4. Model surface reconstruction using cSDF



(a) Protein surface with MHP (b) Zoom with point cloud

(c) 2D PALM data (d) Zoom with point cloud

Fig. 5. Biological surface reconstruction using cSDF

3.3. Applications in Bioimaging

We apply cSDF to two point data sets from biology. The first data set comprises 3D positions of atoms in a protein conformation obtained from molecular-dynamics simulations[1]. We use cSDF to reconstruct the molecular surface of the protein and locally shade it according to the Molecular Hydrophobicity Potential (MHP). The result is shown in Fig. 5(a) and (b). The second case considers a 2D PALM super-resolution image. PALM intrinsically produces point clouds and cSDF can be used to reconstruct the imaged geometry (Fig. 5(c) and (d)). The PALM image shows a cell nucleus with the lamin proteins of the nuclear lamina fluorescently labeled[2]. Prior to cSDF reconstruction, the image is pre-processed to remove outlier points. Identification of objects from the reconstructed surface is a post-processing step, done here to retain only the largest connected component.

4. CONCLUSIONS

We have presented coupled signed-distance functions (cSDF) for implicit surface reconstruction from unstructured point clouds. The method is based on applying distance-preserving shift transformations to the Euclidean distance map of the point data. The result is a signed-distance function, dispensing with the need for re-initialization. We have demonstrated the method in both 2D and 3D, benchmarked its computational cost, and illustrated its capability to represent high curvature without adaptive grids. We also demonstrated the applicability of the method to 2D and 3D real-world data. We currently use a single level function and can hence not represent intersecting surfaces.

5. REFERENCES

[1] M. Sussman, "A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles," *J. Comput. Phys.*, vol. 187, no. 1, pp. 110–136, 2003.

[2] N. Paragios and R. Deriche, "Coupled geodesic active regions for image segmentation: A level set approach," in *Proc. Europ. Conf. Computer Vision*. 2000, pp. 224–240, Springer.

[3] X. Zeng, L. H. Staib, R. T. Schultz, and J. S. Duncan, "Segmentation and measurement of the cortex from 3D MR images using coupled-surfaces propagation," *IEEE Trans. Medical Imaging*, vol. 18, no. 10, pp. 927–937, 1999.

[4] M. Kazhdan, "Reconstruction of solid models from oriented point sets," in *Proc. 3rd Eurographics Symp. Geometry Process.*, 2005, pp. 73–es.

[5] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. 4th Eurographics Symp. Geometry Process.*, 2006, pp. 61–70.

[6] C. A. Öztireli, G. Guennebaud, and M. H. Gross, "Feature preserving point set surfaces based on non-linear kernel regression," *Comput. Graph. Forum*, vol. 28, no. 2, pp. 493–501, 2009.

[7] H. K. Zhao, S. Osher, B. Merriman, and M. Kang, "Implicit and nonparametric shape reconstruction from unorganized data using a variational level set method," *Computer Vision and Image Understanding*, vol. 80, no. 3, pp. 295–314, 2000.

[8] H. K. Zhao, S. Osher, and R. Fedkiw, "Fast surface reconstruction using the level set method," in *Variational and Level Set Methods in Computer Vision*. IEEE, 2001, pp. 194–201.

[9] J. Ye, X. Bresson, T. Goldstein, and S. Osher, "A fast variational method for surface reconstruction from sets of scattered points," UCLA CAM report, UCLA, 2010.

[10] T. Goldstein, X. Bresson, and S. Osher, "Geometric applications of the split Bregman method: Segmentation and surface reconstruction," *J. Sci. Comput.*, vol. 45, no. 1, pp. 272–293, 2010.

[11] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 93, no. 4, pp. 1591, 1996.

[12] I. Tobor, P. Reuter, and C. Schlick, "Multi-scale reconstruction of implicit surfaces with attributes from large unorganized point sets," in *Proc. Shape Modeling Appl.* IEEE, 2004, pp. 19–30.

[13] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H. P. Seidel, "Multi-level partition of unity implicits," *ACM Trans. Graphics*, vol. 22, no. 3, pp. 463–470, 2003.

[14] M. B. Nielsen and K. Museth, "Dynamic Tubular Grid: An efficient data structure and algorithms for high resolution level sets," *J. Sci. Comput.*, vol. 26, no. 3, pp. 261–299, 2006.

[15] C. Li, C. Xu, C. Gui, and M. D. Fox, "Level set evolution without re-initialization: a new variational formulation," in *Proc. Conf. Computer Vision & Pattern Recognition (CVPR)*, Washington, DC, USA, 2005, IEEE, pp. 430–436.

[16] S. Kim, "An $O(N)$ level set method for Eikonal equations," *SIAM J. Sci. Comput.*, vol. 22, no. 6, pp. 2178–2193, 2001.

[17] H. Zhao, "A fast sweeping method for Eikonal equations," *Math. Comput.*, vol. 74, no. 250, pp. 603–628, 2005.

[18] W. K. Jeong and R. T. Whitaker, "A fast iterative method for Eikonal equations," *SIAM J. Sci. Comput.*, vol. 30, no. 5, pp. 2512–2534, 2008.

[19] M. Sussman and E. Fatemi, "An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow," *SIAM J. Sci. Comput.*, vol. 20, no. 4, pp. 1165–1191, 1999.

[1] Data courtesy of Dr. Anton Polyansky, Zagrovic group, MFPL, Vienna.

[2] Data courtesy of Dr. Jonas Ries, Ewers group, ETH Zurich.