2022 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-9620-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICIP46576.2022.9897175

DEFINING POINT CLOUD BOUNDARIES USING PSEUDOPOTENTIAL SCALAR FIELD IMPLICIT SURFACES

Ethan Payne, Amanda S. Fernandez

University of Texas at San Antonio
Department of Computer Science

ABSTRACT

Identifying smooth and meaningful object boundaries of noisy 3D point-clouds presents a challenge. Rather than rely on the points of the cloud itself, we identify a smooth implicit surface to represent the boundary of the cloud. By constructing a scalar field using a semantically-informative pseudopotential function, we take an arbitrary-resolution isosurface and apply standard computer vision morphological transformations and edge detection on 2D slices of the pseudopotential field. When recombined, these slices comprise a new point-cloud representing the 3D boundary of the object as determined by the chosen isosurface. Our method leverages the strength and accessibility of 2D vision tools to identify smooth and semantically significant boundaries of ill-defined 3D objects, and additionally provides a continuous scalar field containing insight regarding the internal structure of the object. Our method enables a powerful and easily implementable pipeline for 3D boundary identification, particularly in domains where natural candidates for pseudopotential functions are already present.

Index Terms— boundary identification, point cloud, pseudopotential functions

1. INTRODUCTION

In domains where 3D point clouds represent samples from continuous underlying objects, there may be a strong motivation to define a smooth object boundary not necessarily limited to the points in the point cloud. Level-set representations and interpolating methods offer a bridge from discrete object representations like point clouds into smooth and continuous 3D surfaces. Non-Uniform Rational B-Spline (NURBS) [1] surface methods interpolate between points within the point cloud to obtain a closed-form continuous definition for the object surface, however these interpolation-based methods will perform poorly for noisy point clouds, and are generally not at all suitable when clouds contain points in the interior volume of the object. While interpolation attempts to explicitly fit a surface to a point cloud, level-set methods like signed distance functions [2] instead define an implicit surface representing the object boundary, along with notions of 'interior' and 'exterior'. While signed distance functions offer an attractive way to represent object boundaries; raw, noisy, and sparse point clouds contain little information from which we might recover such a function. Thus, neither interpolation nor signed distance functions provide a sufficient pipeline to directly define smooth boundaries from arbitrary point clouds.

In this paper we present an approach to defining 3D (and higher-dimensional) objects in terms of smooth implicit surfaces extracted directly from sparse and noisy point clouds. We first introduce a class of functions similar to atomic potentials which we use to construct a 3D scalar field conceptually similar to signed distance functions. By carefully selecting a pseudopotential function to fit the desired domain requirements, we can generate and refine a voxel grid of which we then take an isosurface. At this point, we decompose the grid into slices which we process as 2D images, applying morphological transformations and using the Canny edge detection algorithm to obtain a collection of 2D boundaries. By recombining these 2D boundary slices, we can recover a voxel grid of arbitrary resolution representing the point cloud boundary.

While for many domains and datasets this method may at first prove challenging to employ due to its reliance on an appropriate selection of pseudopotential function, this approach nonetheless offers a fully-explainable pipeline for explicitly extracting smooth boundaries directly from point clouds which would otherwise be unattainable using other state-of-the-art methods.

2. RELATED WORK

Several variants of the boundary identification problem exist as semi-independent research domains, incumbent on the desired context and application. Though the exact approach often differs significantly, these methods are related in the abstract by the use of similar techniques or generation of similar output. While interpolative methods are ultimately still limited by the inherent restrictions of parametric interpola-

This material is based upon work supported by the National Science Foundation under Grant No. 2134237. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

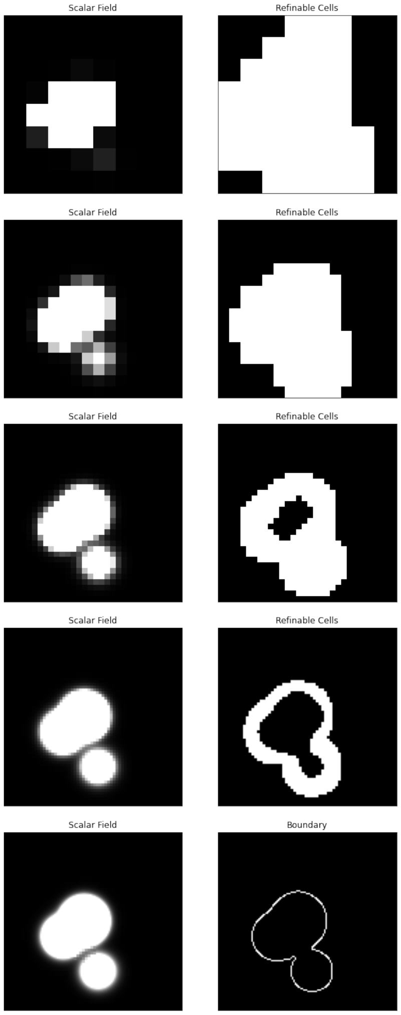

Fig. 1: Computed scalar field slices (left) with voxels marked for refinement (right). The proportion of points to refine increases with the surface area and complexity of the object underlying the point-cloud. The final row shows the computed boundary after refinement has terminated.

tion, Leal et al. [3] successfully apply dimensionality reduction and point-cloud regularization to extend the viability of NURBS fitting to a wider class of objects. On the other hand, while deep learning methods lack a certain degree of explainability and interpretability for defining object boundaries, they do have the benefit of yielding high quality performance even from noisy and incomplete point clouds. By utilizing a deep learning-based approach within a framework of signed distance functions (SDF), Ma et al. [4] learn an SDF from a point cloud, 'pulling' the surrounding space onto the surface defining the object. Park et al. [5] also similarly learn SDFs from point clouds, using an encoder-decoder architecture and a probabilistic approach to predict the value of the signed distance function at a given location. Towards the task of part-boundary detection rather than whole-object boundary detection, Loizou et al. [6] take inspiration from advances in 2D image contour detection and use a graph convolutional network to locate internal boundaries between object parts. Abello et al. [7] use 2D boundary identification techniques even more directly towards an application on point clouds by backprojecting a non-learning-based 2D image segmentation result onto a 3D scene. Finally, while entirely lacking a smooth or direct approach to object boundaries, Xiang et al. [8] use their Snowflake Point Deconvolution with a skip-transformer to fill in and complete sparse point clouds. While some of the methods mentioned above operate in terms of part-boundaries, point cloud segmentation, or point cloud completion rather than holistic boundary identification, underlying all of these tasks is the concept of a well-defined 3D object boundary.

3. PROPOSED APPROACH

Our approach consists of three distinct steps: (1) selecting a pseudopotential function appropriate for the given boundary-identification task, (2) computing the scalar field corresponding to the selected pseudopotential, and (3) processing the scalar field voxel grid using established vision techniques.

3.1. Pseudopotential Functions

Fundamental to the viability of our method is the selection of an appropriate pseudopotential function from which a scalar field can be computed. For a point cloud $P = \{p_0, p_1, ...p_n\}, \quad p_i \in \mathbb{R}^3$ of n 3D points, and a location $x \in \mathbb{R}^3$, we define the pseudopotential f in Equation 1:

$$f(x, p_i) \coloneqq f_0(\|x - p_i\|) = f_0(r) \tag{1}$$

Where f_0 is a function of distance r from the center of a voxel cell x and a point p_i in the point cloud, chosen for its semantic relation to the desired boundary-identification application. Some examples of choices for f_0 are the following:

$$f_0(r) = \begin{cases} 1 & r \leq \alpha \in \mathbb{R}^+ \\ 0 & r > \alpha \end{cases} \tag{2}$$

$$f_0(r) = r \tag{3}$$

$$f_0(r) = \frac{1}{r} \tag{4}$$

$$f_0(r) = \left(\frac{\sigma}{r}\right)^{12}, \quad \sigma \in \mathbb{R} \tag{5}$$

$$f_0(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right], \quad \varepsilon, \sigma \in \mathbb{R} \tag{6}$$

Equation 2 represents an indicator function with a sharp cutoff at a radius α, Equation 3 scales linearly with distance, Equation 4 scales inversely with distance, Equation

Algorithm 1 Voxel-Grid Refinement

```
$P \leftarrow$ point cloud
$V_0 \leftarrow$ initial voxel grid
$n \leftarrow$ max number of refinement iterations
$i \leftarrow 0$
for $x \in V_0$ do
    compute $F(x, P)$
    if $x$ lies on the border of the isosurface threshold then    ▷ simplified for brevity
        mark $x$ and neighbors for refinement
    end if
end for
while number of cells marked for refinement is nonzero and $i \leq n$ do
    $V_i \leftarrow$ voxel grid of greater resolution
    for $x \in V_i$ do
        if $x$ marked for refinement then    ▷ must check 'parent' of $x$ from previous resolution level
            recompute $F(x, P)$
        end if
    end for
    $i \leftarrow i + 1$
end while
```

5 adds a scaling parameter σ and a raises the radial effect to a large power, and Equation 6 is the Lennard-Jones potential, a commonly used approximation for intermolecular interactions. Different parameterizations and formulations of pseudopotential functions will profoundly affect the nature of the computed point-cloud boundary, so the selection of this function is the most sensitive step of our method.

3.2. Computation of the Scalar Field

With an appropriate pseudopotential function selected, we embed the point cloud within a voxel grid of desired resolution and compute the value of the function with respect to each point in the point-cloud at each cell of this grid. A natural means to aggregate the effects of individual points is to sum the respective pseudopoentials yielding $F(x, P)$ defined below in Equation 7:

$$F(x, P) \coloneqq \sum_{p_i \in P} f(x, p_i) \tag{7}$$

However, this is not the only possible aggregation method. For different boundary behavior, we might for example instead take the product, (Equation 8), compute a p-norm (Equation 9), take the inverse of an aggregated pseudopotential (Equation 10), or aggregate only the effects of the k-nearest-neighbors of x within the point cloud (Eq. 11).

$$F(x, P) \coloneqq \prod_{p_i \in P} f(x, p_i) \tag{8}$$

$$F(x, P) \coloneqq \left[\sum_{p_i \in P} f(x, p_i)^{\mathrm{p}} \right]^{\frac{1}{\mathrm{p}}}, \quad \mathrm{p} \geq 1 \tag{9}$$

$$F(x, P) \coloneqq \frac{1}{\sum_{p_i \in P} f(x, p_i)} \tag{10}$$

$$F(x, P) \coloneqq \sum_{p_i \in \mathrm{KNN}(x, P)} f(x, p_i) \tag{11}$$

This process is easily the most computationally intensive step of our approach, as for a cloud of n points, embedded in a $m_0 \times m_1 \times m_2$ grid, the number of function calls to f using Equation 7 is $O(n * m_0 * m_1 * m_2)$. Several techniques can be employed to reduce this complexity, including adaptive refinement of the grid resolution (Algorithm 1, Figure 1), cutoffs for long-range effects (Equation 12), and thresholding for voxel cells once the value of the pseudopotential reaches a given cutoff (Equation 13).

$$\begin{gathered} F_\alpha(x, P) \coloneqq F(x, P_\alpha), \\ P_\alpha = \{p_i : \|x - p_i\| < \alpha \in \mathbb{R}^+\} \end{gathered} \tag{12}$$

$$F_\beta(x, P) \coloneqq \min \begin{cases} F(x, P) \\ \beta \in \mathbb{R} \end{cases} \tag{13}$$

We can visualize the impact of modifying the f and F in Figure 2.

It should be noted that each of these time-saving techniques sacrifices some information of the final computed scalar field, so the decision whether or not to apply any of these complexity reduction methods must be made with careful consideration of the required boundary semantics. Importantly, we can obtain a boundary of arbitrary resolution and smoothness by modifying the resolution of the final voxel-grid. We can visualize this effect in Figure 3.

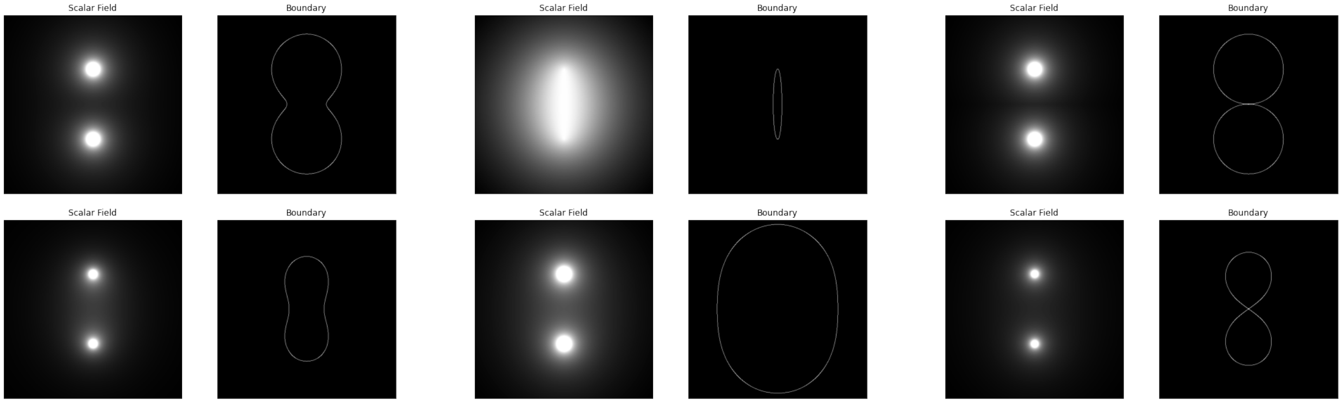

Fig. 2: Scalar fields and boundaries of a cloud of 2 points corresponding to various choices for pseudopotential f and F.

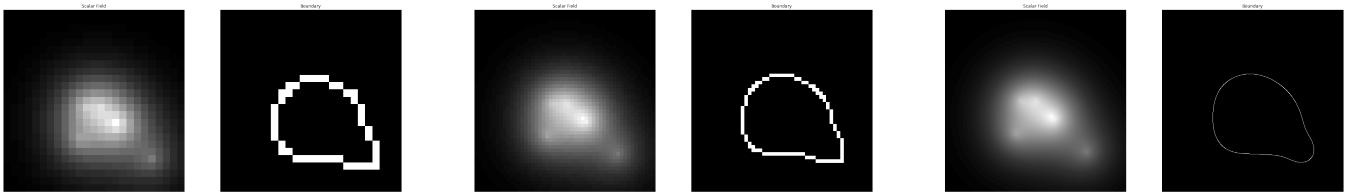

Fig. 3: A higher voxel-grid resolution yields a more precise boundary.

3.3. Slice-Wise Boundary Identification

Once the scalar field has been computed, we can identify the boundary corresponding to each slice of the 3D voxel grid. We first threshold the scalar field at a value consistent with the semantics of the final desired boundary. For each slice of the voxel grid along a given axis, we then may choose to perform morphological image processing transformations (e.g. erode, dilate, open, close, fill) to remove undesired features, such as internal voids and surface discontinuities. Which morphological transformations are used will depend heavily on the desired boundary semantics. For example, for point-clouds sampled only on the surface of an object, the resulting interior voids should be filled as these boundaries are not meaningful.

After the application of any necessary morphological transformations, we identify a 2D boundary for each slice using the Canny edge detection algorithm. By recombining the individual Canny edges, we obtain a grid of boundary voxels. Since the decomposition of the voxel grid into slices along a single axis introduces a bias depending on which axis is selected, we can repeat this slice-wise process along the remaining two axes, and aggregate the results using a union or intersection. We must also note that similar and potentially superior results might be attained using a fully-3D implementation of the Canny algorithm, averting the need for decomposition into slices, but currently available Canny implementations only support 2D input data.

4. CONCLUSIONS

While the pseudopotential-based boundary definition proposed in this work is certainly applicable in areas such as particle dynamics simulation and astrophysical analysis, further investigation is needed for additional boundary localization tasks. For example, evaluation of the performance of this technique on standard point cloud datasets, and use of this approach to augment existing techniques such as deep learning and interpolation-based methods. For domains where ground-truth boundary datasets are available, we might also apply deep learning techniques to identify optimal pseudopotential functions and hyperparameters. We may also be able to take advantage of the graph-based techniques developed for the optimization of the conceptually-similar Mumford-Shaw problem [9] to replace our relatively naive voxel grid refinement algorithm, and future work should consider to what extent our method is related to the Mumford-Shaw problem. Furthermore, this work demonstrates that the problem of point cloud boundary detection represents a strong motivation for developing fully 3D (and higher dimensional) implementations of corresponding 2D image morphological transformations and the Canny edge detection algorithm.

This class of pseudopotential functions and their associated isosurfaces, in concert with morphological image transformations and the Canny edge detection algorithm, present a unique method for conceiving smooth boundaries of noisy point clouds. Our approach offers a novel and broadly applicable framework for boundary-identification problems.

5. REFERENCES

[1] Gerald E. Farin, *NURB curves and surfaces: from projective geometry to practical use*, AK Peters, Ltd., 1995. 1

[2] Mikaël Rousson and Nikos Paragios, "Shape priors for level set representations," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 05 2002, vol. 2351, pp. 78–92. 1

[3] Nallig Leal, Esmeide Leal, and John Branch, "Automatic construction of nurbs surfaces from unorganized points," *DYNA*, vol. 78, pp. 133–141, 04 2011. 2

[4] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker, "Neural-pull: Learning signed distance function from point clouds by learning to pull space onto surface," in *International Conference on Machine Learning*. PMLR, 2021, pp. 7246–7257. 2

[5] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 165–174. 2

[6] Marios Loizou, Melinos Averkiou, and Evangelos Kalogerakis, "Learning part boundaries from 3d point clouds," in *Computer Graphics Forum*. Wiley Online Library, 2020, vol. 39, pp. 183–195. 2

[7] Montiel Abello, Joshua G Mangelson, and Michael Kaess, "A graph-based method for joint instance segmentation of point clouds and image sequences," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9565–9571. 2

[8] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han, "Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5499–5509. 2

[9] Noha El-Zehiry, Steve Xu, Prasanna Sahoo, and Adel Elmaghraby, "Graph cut optimization for the mumford-shah model," pp. 182–187, 01 2007. 4