



ARTICLE

Interactive Restoration of Three-Dimensional Implicit Surface with Irregular Parts

Jiayu Ren^{1,*}, Yoshihisa Fujita² and Susumu Nakata²

¹Graduate School of Information Science and Engineering, Ritsumeikan University, Shiga, 525-8577, Japan

²College of Information Science and Engineering, Ritsumeikan University, Shiga, 525-8577, Japan

*Corresponding Author: Jiayu Ren. Email: gr0450ek@ed.ritsumei.ac.jp

Received: 08 August 2022 Accepted: 08 November 2022

ABSTRACT

Implicit surface generation based on the interpolation of surface points is one of the well-known modeling methods in the area of computer graphics. Several methods for the implicit surface reconstruction from surface points have been proposed on the basis of radial basis functions, a weighted sum of local functions, splines, wavelets, and combinations of them. However, if the surface points contain errors or are sparsely distributed, irregular components, such as curvature-shaped redundant bulges and unexpectedly generated high-frequency components, are commonly seen. This paper presents a framework for restoring irregular components generated on and around surfaces. Users are assumed to specify local masks that cover irregular components and parameters that determine the degree of restoration. The algorithm in this paper removes the defects based on the user-specific masks and parameters. Experiments have shown that the proposed methods can effectively remove redundant protrusions and jaggy noise.

KEYWORDS

Shape modeling; implicit surfaces; interactive restoration

1 Introduction

Implicit representation is a common method for defining three-dimensional shapes and has advantages in many applications, such as shape modeling in computer graphics, physics-based simulation, material science, and medical imaging. One important application of implicit representation is shape modeling based on constructive solid geometry, skeletal modeling, and blending, among others [1,2]. Three-dimensional shapes can be effectively structured as combinations of primitives using algebraic operations. Another application of implicit representation is the shape modeling of obstacles for particle-based fluid simulation [3,4]. Implicit representation helps to efficiently estimate boundary force to fluid particles when the particles approach obstacles. It is also useful for shape modeling of complex materials, such as composite porous and molecular surfaces [5–7]. In these applications, such structures of complex materials and molecules are generated procedurally and defined as isosurfaces of scalar fields.



An implicit surface reconstruction from a set of surface points is a typical problem in computer graphics and several methods for the automatic generation of scalar fields have been proposed since the 1990s. Section 2 provides a brief review of the implicit surface reconstruction methods. However, as mentioned in several articles [8–12], the quality of reconstructed surfaces is sometimes poor depending on the density and accuracy of the given surface points. Defects, such as missing points, noise, and nonuniform sampling, often result in generated shapes with irregular components, such as redundant protrusions and noisy-like jaggy components. Fig. 1 gives two examples of irregular implicit surfaces.

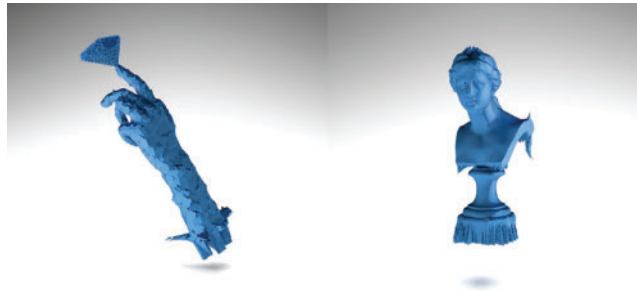


Figure 1: Examples of irregular surfaces generated based on piecewise polynomials [13,14]. Left: surface “Hand” with noisy and protruding components; the point data is provided by Rapidform (INUS Technology, Inc., USA). Right: surface “Venus” with protruding components; the point data is generated using a 3D scanning system

In this paper, we provide solutions for restoring irregular implicit surfaces. We focus on two major surface irregularities: redundant protrusions and jaggy noise. We assume that users specify masks in three-dimensional space and that restoration is performed only in the masked region. If a mask is specified to cover a redundant protrusion on a surface, our method updates the masked region by replacing the long protruding part with a round or flat-shaped surface. Users can control the roundness/flatness of the restored portion of the surface. The new scalar field within the mask is generated as the least-square solution of Laplace’s equation with both Dirichlet and Neumann conditions. The Dirichlet boundary condition is used to guarantee continuity at the border of the mask, while the Neumann boundary condition is used to control the roundness/flatness. Similarly, users can specify the region of restoration of jaggy noise by a mask that covers the region. Furthermore, users can control the degree of noise reduction by a parameter. Noise reduction is performed through low-resolution resampling within the mask. The final scalar field is generated by dense spline interpolation of the entire domain, with the spline ensuring continuity.

2 Related Work

Shape representation can be performed in various ways. Traditionally, shapes can be represented by enumerating small geometric elements and allowing them to cover the surface, a technique known as explicit representation [15,16]. The advantage of this approach is that the surface can be extended arbitrarily based on the users’ needs. Another prospective approach for representing a shape is to find an appropriate field defined by the function $f(\mathbf{x})$, where \mathbf{x} is a point in N -dimensional space, and the shape is considered the zero-value set of the field, when $f(\mathbf{x}) = 0$. This idea is known as implicit representation and it is feasible to generate shapes without having to enumerate elements over the shape. One approach for determining the field $f(\mathbf{x})$ is an implicit surface reconstruction

from a set of oriented surface points. Given a set of surface points, $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$, with normal vectors, $\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_n$, the field is determined so that the interpolation conditions, $f(\mathbf{x}_i) = 0$, and the gradient conditions, $\nabla f(\mathbf{x}_i) \parallel \mathbf{n}_i$, are exactly or approximately satisfied. Fig. 2 shows an example of the field $f(\mathbf{x})$ that satisfies the interpolation conditions. The signs of the field values should be opposite inside and outside the closed shape. The colors red and blue denote positive and negative values, respectively, and the depth of the color is proportional to the absolute value of the field.

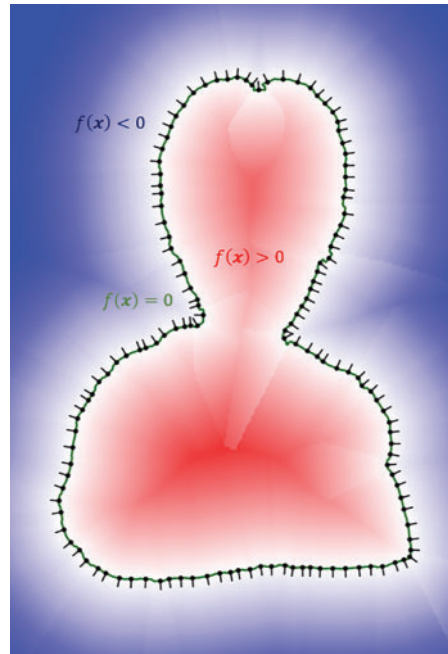


Figure 2: The color map of a field $f(\mathbf{x})$ and the two-dimensional curve defined as $f(\mathbf{x}) = 0$. Constraint points and normal vectors are represented by dots and short line segments respectively. The example is generated using the grid of the polynomial approach [13,14]

Existing work on implicit representations can be divided into two categories, namely, traditional mathematical methods and newly proposed learning-based methods. A typical traditional method is to generate the field $f(\mathbf{x})$ as a linear combination of radial basis functions so that the zero-value set of the field satisfies the interpolation or approximation conditions [17–19]. In the multilevel partition of the unity approach [20], the field is generated as a weighted sum of hierarchical polynomials approximating signed distance functions (SDFs). The hierarchical structure greatly enhanced the capability of numerous input points. Another approach is to use Fourier series [21] and wavelets [22] to generate fields as indicator functions, where the value is one inside the surface and zero outside the surface. The Poisson surface reconstruction [9] replaces the function generation process with a Poisson equation with hierarchical local bases and is noise resistant. The grid of polynomial approach [13,14] generates fields as a set of polynomials that approximate the SDF condition. The simplicity of the function helps in the fast direct rendering of the surfaces. See comprehensive surveys [23,24] for detail.

More recently, deep learning-based approaches have gained more popularity. In those approaches, SDFs [25–28] or occupancy fields [29–32] are learned via machine learning methods. In [25,33], estimated projection vectors are obtained using deep learning methods, such as regression forests [34]. The sign can be calculated using the projection vector and surface normal. Finally, the signed

projection vectors can be used to approximate SDFs. Moreover, the surfaces can be represented with decision boundaries retrieved from neural networks [35]. In this approach, an occupancy network is constructed, and the probability of either taking zero or one at an arbitrary point is output. The occupancy field consists of zeros and ones based on the higher probability at the corresponding location. The shape is represented by the decision boundary that separates zeros and ones. Different forms of the expected shape, such as point clouds, predefined models, or images, can be used as input. A state-of-the-art survey is provided in [36].

Although some methods for creating implicit shapes have been proposed, the problem of irregular parts is a typical issue, as mentioned in Section 1. Our approach is to restore irregular surfaces a posteriori according to the interactive operations of users. There are some methods for modifying implicit surfaces interactively. An implicit surface manipulation method, known as WarpCurves [37], has been proposed. This method allows users to modify surfaces by manipulating user-specific curves on surfaces. Another approach for deforming implicit surfaces is free-form deformation [38]. In this method, the field is updated according to control points or curves specified on surfaces. An interactive manipulation approach based on typical mathematical operations, such as Boolean, blending, twisting, and bending, has also been proposed [39]. Various deformations can be performed as a combination of simple operations and real-time rendering. Our method focuses on restoration of unexpectedly generated shapes and is designed to effectively remove irregular parts, including redundant protrusions and jaggy noise, using simple operations. The restoration level can also be controlled by user-specific parameters. The geometric information that users should provide is only masks that cover the irregular parts. The next section discusses the reasons for each type of issue.

3 The Issue

One of the common challenges in model creation is irregular parts that are unexpectedly generated, including curvature shaped redundant bulges and unexpectedly generated high frequency components. Defects such as missing points or unevenly displaced normal vectors may cause noisy or redundant problems. In this case, the interpolation may be misguided, resulting in either large curvatures or high-frequency components. As a result, redundant protruding parts or noisy parts are produced. Figs. 3 and 4 show some examples.

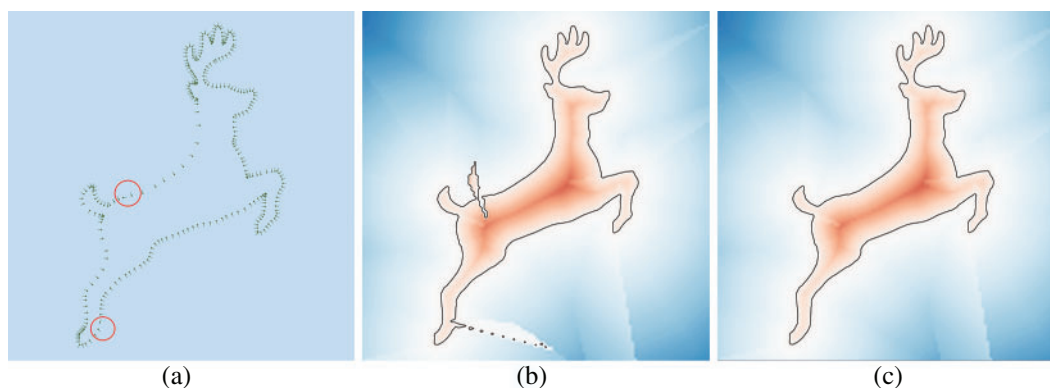


Figure 3: An example of redundant parts caused by the wrong direction of normal vectors, generated using the grid of polynomial approach [13, 14]. Irregular directions of normal vectors are shown in (a). An irregular curve $f(x) = 0$ is shown in (b), and the ideal curve is shown in (c)

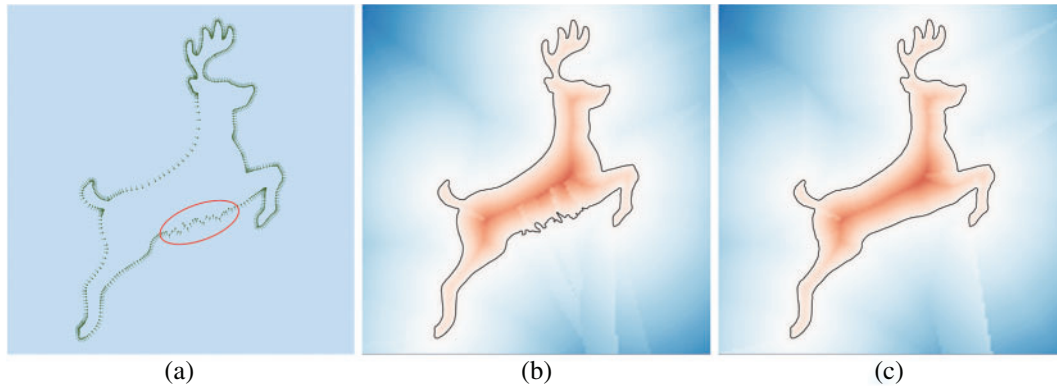


Figure 4: An example of a noisy curve, generated using the grid of polynomial approach [13,14]. Fluctuated distribution of normal vectors is shown in (a); the irregular curve $f(x) = 0$ is shown in (b), and the ideal curve is shown in (c)

4 Proposed Methods

This section presents implicit surface restoration methods that remove unexpectedly generated redundant protrusions and jaggy noise. Assuming a scalar field, $f(\mathbf{x})$, is given in a bounding box Ω^{BB} and the surface, $f(\mathbf{x}) = 0$, has some unexpected defects. The goal is to obtain a new scalar field, $g(\mathbf{x})$, defined in Ω^{BB} so that the surface, $g(\mathbf{x}) = 0$, is defect-free. The basic concept of our correction is to replace the field with a new one locally in the region surrounding the defects. Assuming a user-specific region $\Omega^{mask} \subset \Omega^{BB}$ that covers the defect as a 2D illustration is shown in Fig. 5, which shows some examples of a mask at the specified location where the redundant protruding component should be removed. In our approach, the original field is replaced with a new one so that the defects are appropriately removed. The new field is generated in such way that the values are approximately equivalent to the rest of the mask, i.e.,

$$g(\mathbf{x}) \begin{cases} \approx f(\mathbf{x}) & (\mathbf{x} \in \Omega^{BB} \setminus \Omega^{mask}) \\ = h(\mathbf{x}) & (\mathbf{x} \in \Omega^{mask}) \end{cases}, \quad (1)$$

where $h(\mathbf{x})$ is the newly generated field in the mask. Simultaneously, $g(\mathbf{x})$ should be continuous at the boundary of the mask.

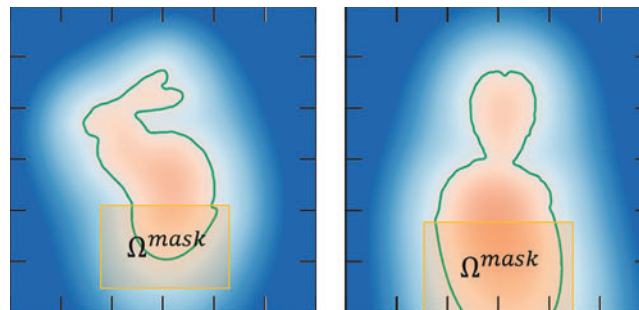


Figure 5: Illustrations of user-interactive specifications. To specify the mask, the user is expected to input the coordinate of the lower-left and the upper-right vertex of the mask

4.1 Correction of Redundant Parts

The strategy of removing redundant parts is illustrated in Fig. 6. Assuming $f(\mathbf{x}) = 0$ denotes an implicit shape with redundant parts. The goal is to define a new field, $g(\mathbf{x})$, as an alternative to the given field, $f(\mathbf{x})$. A standard situation occurs when the protruding part does not exceed the boundary of Ω^{BB} , while a particular situation occurs when the protrusion touches one or more of the boundaries of Ω^{BB} , and the mask touches the corresponding boundary. In this case, special care will be necessary, as discussed later in this section. The left and right illustrations in Fig. 5 show the standard and special situations, respectively, with corresponding mask specifications. The first step is to obtain a local field $h(\mathbf{x})$ ($\mathbf{x} \in \Omega^{\text{mask}}$) that appropriately removes the redundant protrusions. Assuming the rectangular mask is placed around a protrusion so that one edge is penetrated by the base of the protrusion as shown in Fig. 5. In an ideal situation, the new local field should possess the following three properties:

P1: the field is as smooth as possible.

P2: the value is equivalent to $f(\mathbf{x})$ at the boundary of the mask for continuity.

P3: the value decays rapidly from positive to negative at the edge penetrated by the base.

The property P3 is to control the speed of decay. In this approach, it is assumed that some protruded surfaces are unexpectedly generated as the result of limitations on scanning environment. It means that users can specify the flatness of the restored surfaces after removing the protrusions by controlling the speed of the decay.

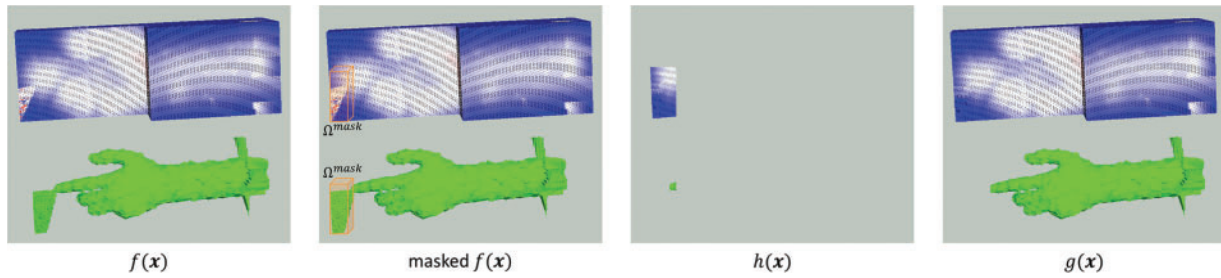


Figure 6: The strategy to remove redundant parts, a cuboid mask has been specified. The cross-section of the fields and the implicit surfaces are shown at the upper and lower part of the figures, respectively

To obtain a field that mostly satisfies the three properties, we propose the following equations:

$$\begin{cases} \min \int_{\Omega^{\text{mask}}} (\Delta h(\mathbf{x}))^2 d\Omega \\ h(\mathbf{x}) = f(\mathbf{x}) \quad (\mathbf{x} \in \partial\Omega^{\text{mask}}), \\ \frac{\partial h(\mathbf{x})}{\partial \mathbf{n}} = \alpha \quad (\mathbf{x} \in E^{\text{mask}}) \end{cases} \quad (2)$$

where $\partial\Omega^{\text{mask}}$ is the boundary of the mask, E^{mask} is the edge penetrated by the base of the protrusion, and α is a user-specific parameter that determines the gradient of the field at the edge. The first, second, and third equations correspond to P1, P2 and P3, respectively. In the first item, the field $h(\mathbf{x})$ is the solution to Laplace's equation if the minimum value is zero. Unfortunately, Laplace's solution cannot be obtained because the second and third conditions are overdetermined. We adopt the least-square problem instead of Laplace's equations to obtain a solution that is as smooth as possible under the overdetermined conditions. The second equation is the condition of continuity, which serves as the Dirichlet boundary condition in the context of solving the partial differential equation. The third one

is the condition that determines the gradient of $h(\mathbf{x})$ at E^{mask} , which serves as the Neumann boundary condition. If α is large, the value rapidly decays toward the negative, and the zero-level set should be close to the edge. The reverse is also true. This means that α plays the role of controlling the roundness of the restored shape after removing the redundant protrusion. Fig. 7 shows the relationship between the parameter α and the effects of restoration of 2D examples.

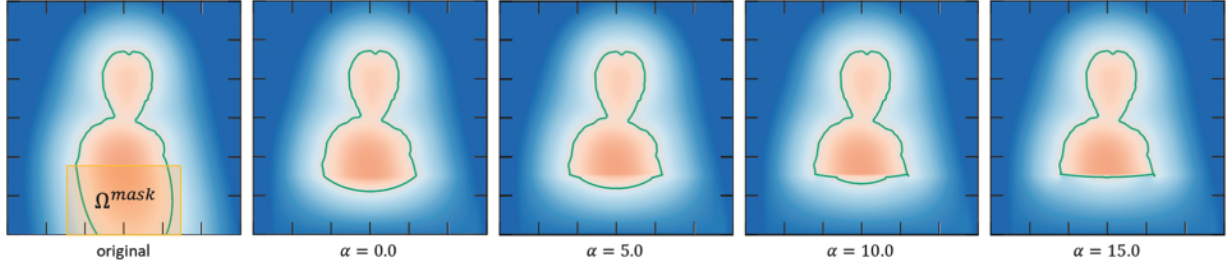


Figure 7: Different roundness of restored shapes obtained by setting different gradients of the field at the top edge of the mask. The examples are generated via Turk's method [17]

In some special situations, the surface unexpectedly extends toward the border of Ω^{BB} , and the mask should touch the border, as shown in Fig. 5 (right). In this case, the least-square problem lacks Dirichlet boundary condition at the border. Therefore, the function value should be determined along the border before solving the equations. A natural approach will be used to determine the values as the linear interpolation of an edge of $f(\mathbf{x}^{\text{left}})$ and $f(\mathbf{x}^{\text{right}})$, where \mathbf{x}^{left} and $\mathbf{x}^{\text{right}}$ are the ends of the edge at the border. In 3D situations, assuming the bottom surface takes Dirichlet boundary condition, it is to be determined by the mean value of the interpolation of the edge of $f(\mathbf{x}^{\text{left}})$ and $f(\mathbf{x}^{\text{right}})$ and, the interpolation of the edge of $f(\mathbf{x}^{\text{front}})$ and $f(\mathbf{x}^{\text{rear}})$.

For practical computations, the above-mentioned formulation should be appropriately discretized. We apply a finite difference scheme to obtain the discrete linear least-square equations. Assuming the domain Ω^{BB} is divided into $L \times M \times N$ uniform cells whose centers are \mathbf{x}_{ijk} ($i = 1, 2, \dots, L, j = 1, 2, \dots, M, k = 1, 2, \dots, N$) and a mask Ω^{mask} is placed so that the protrusion to be removed penetrates the top edge of the mask. Let I_{MIN} and I_{MAX} be the minimum and maximum indices of the discrete points within the mask Ω^{mask} along the x -axis, respectively. Let J_{MIN} and J_{MAX} be the minimum and maximum indices of the discrete points within the mask Ω^{mask} along the y -axis, respectively. Let K_{MIN} and K_{MAX} be the minimum and maximum indices of the discrete points within the mask Ω^{mask} along the z -axis, respectively. The goal is to obtain the discrete values of the restored function, $g(\mathbf{x})$, i.e.,

$$g_{ijk} = \begin{cases} f_{ijk} & (\mathbf{x}_{ijk} \notin \Omega^{\text{mask}}) \\ h_{ijk} & (\mathbf{x}_{ijk} \in \Omega^{\text{mask}}) \end{cases}, \quad (3)$$

where $f_{ijk} = f(\mathbf{x}_{ijk})$. At this moment, we have m unknown variables, where $m = (I_{\text{MAX}} - I_{\text{MIN}} + 1) \times (J_{\text{MAX}} - J_{\text{MIN}} + 1) \times (K_{\text{MAX}} - K_{\text{MIN}} + 1)$ and we will determine the unknowns h_{ijk} using the Eq. (2). With the abovementioned discretization, (2) can be discretized using a standard finite difference scheme as the following equations:

$$\begin{cases} \min_{h_{ijk}} \sum_{i=I_{\min}}^{I_{\max}} \sum_{j=J_{\min}}^{J_{\max}} \sum_{k=K_{\min}}^{K_{\max}} \left(\frac{h_{i-1,j,k} + h_{i+1,j,k} + h_{i,j-1,k} + h_{i,j+1,k} + h_{i,j,k-1} + h_{i,j,k+1}}{6} \right)^2 \\ h_{ijk} = f_{ijk} \quad (i \in \{I_{\min} - 1, I_{\max} + 1\} \text{ or } j \in \{J_{\min} - 1, J_{\max} + 1\} \text{ or } k \in \{K_{\min} - 1, K_{\max} + 1\}). \\ h_{ijK_{\max}} = f_{ijk} - \alpha d \quad (i = I_{\min}, I_{\min} + 1, \dots, I_{\max}, \quad j = J_{\min}, J_{\min} + 1, \dots, J_{\max}) \end{cases} \quad (4)$$

where the notation d is the distance between the centers of neighboring cells. The discrete values at the top row in the mask, $h_{ijK_{\max}}$ ($I_{\min} \leq i \leq I_{\max}; J_{\min} \leq j \leq J_{\max}$), can be determined only by the third equation. Therefore, we have n remaining unknowns where $n = (I_{\max} - I_{\min} + 1) \times (J_{\max} - J_{\min} + 1) \times (K_{\max} - K_{\min})$, and the discrete Eq. (4) leads to an overdetermined linear least-square system:

$$\min_{\mathbf{h}} \|\mathbf{A}\mathbf{h} - \mathbf{b}\|^2 \quad (\mathbf{A} \in \mathbb{R}^{m \times n}), \quad (5)$$

where \mathbf{h} is the vector of n unknowns and \mathbf{b} is the vector of m constants corresponding to m equations. Note that $m > n$.

The uniqueness of the least-square solution is critical. As a result, the solution is guaranteed to be unique, as described below. According to Eq. (4), the matrix \mathbf{A} contains n rows that are identical to Laplace's equation with the Dirichlet condition on the whole boundary, which has a unique solution. This means that the rank of the matrix is n . In other words, the matrix is full rank. Based on the basic properties of linear algebra, the overdetermined least-square problem with a full rank matrix has a unique solution. This means that although there are different least-square solvers, they all produce the same solution if they are accurate.

Another issue is the choice of least-square solvers. Typically, singular value decompositions (SVD), QR and conjugate gradient least squares (CGLS) are used to solve linear least square problems [40]. In this study, the matrix is sparse and might be quite large depending on the size of the mask and the resolution of the discretization. The algorithms of SVD and QR use dense matrices and are inappropriate for users because they require a significant amount of computational time and memory space. However, the CGLS matches our case because the algorithm is designed for sparse and large matrices. Therefore, we adopt CGLS as the least-square solver throughout this paper.

All restored discrete values in Eq. (3) can be obtained by solving the least-square problem. Finally, the restored field, $g(\mathbf{x})$, can be obtained as the standard spline interpolation [13,14] of the discrete values g_{ijk} , i.e.,

$$g(\mathbf{x}) = \sum_i \sum_j \sum_k c_{ijk} B_i(x) B_j(y) B_k(z) \quad (\mathbf{x} = (x, y, z) \in \Omega^{\text{BB}}), \quad (6)$$

where $B_i(t)$ is a quadratic B-spline function with uniform intervals and c_{ijk} are the spline coefficients. Note that spline interpolation is not the only option and there are other ways like tri-linear interpolation and radial basis functions. We applied spline interpolation because the scalar values are given at uniform grid points and a smooth interpolation can be performed using an efficient scheme described in [13]. In conclusion, Fig. 8 shows the complete procedure of removing redundant parts. For more concise presentation of discrete field f_{ijk} and g_{ijk} , the illustrations are given in 2D form.

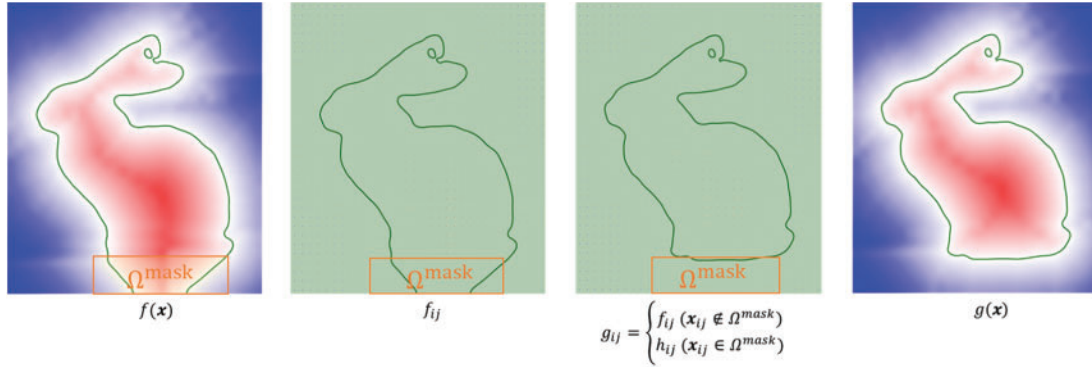


Figure 8: Illustration of the proposed algorithm of removing redundant parts

4.2 Correction of Noisy Parts

Assuming the shape, $f(\mathbf{x}) = 0$, contains partially generated jaggy noise and a user-specific mask, Ω^{mask} , covering the noise. The goal is to define a new field, $g(\mathbf{x})$, as Eq. (1). In this situation, the values of $f(\mathbf{x})$ vary slightly about the noisy surface. Our main idea is to smooth out the small fluctuations in the user-specific mask via coarse sampling. To obtain the final function, $g(\mathbf{x})$, which is continuous and smooth along the boundary of Ω^{mask} , we perform fine sampling and spline interpolation over the entire domain. The following is the mathematical formulation of the idea. Let $(m^{\text{mask}}, n^{\text{mask}})$ be a user-specific sampling resolution of the mask, $\mathbf{x}_{ijk}^{\text{mask}}$ be the sampling points, and $f_{ijk}^{\text{mask}} = f(\mathbf{x}_{ijk}^{\text{mask}})$. We define the restored function, $h(\mathbf{x})$ ($\mathbf{x} \in \Omega^{\text{mask}}$), as the standard spline interpolation of f_{ijk}^{mask} , i.e.,

$$h(\mathbf{x}) = \sum_i \sum_j \sum_k c_{ijk}^{\text{mask}} B_i(x) B_j(y) B_k(z) \quad (\mathbf{x} = (x, y, z) \in \Omega^{\text{mask}}), \quad (7)$$

where $B_i(t)$ is the quadratic B-spline with uniform intervals and c_{ijk}^{mask} are the coefficients of the spline interpolation.

Currently, we have $h(\mathbf{x})$ and $f(\mathbf{x})$ inside and outside the mask, respectively. The two functions have gaps along the boundary of Ω^{mask} , as shown in Fig. 9, and they must be smoothly connected. Let \mathbf{x}_{ijk} be fine uniform sampling points over Ω^{BB} . We then define the final function, $g(\mathbf{x})$, as the spline interpolation of the fine sampling of the two functions, i.e.,

$$g_{ij} = \begin{cases} f(\mathbf{x}_{ijk}) & (\mathbf{x}_{ijk} \in \Omega^{\text{BB}} \setminus \Omega^{\text{mask}}) \\ h(\mathbf{x}_{ijk}) & (\mathbf{x}_{ijk} \in \Omega^{\text{mask}}) \end{cases}, \quad (8)$$

$$g(\mathbf{x}) = \sum_i \sum_j \sum_k c_{ijk} B_i(x) B_j(y) B_k(z) \quad (\mathbf{x} = (x, y, z) \in \Omega^{\text{BB}}), \quad (9)$$

where c_{ijk} are the coefficients of the spline interpolation of g_{ijk} . Fig. 10 shows the procedure of de-noise. Smoothing should be applied only to the mask while the detailed shape outside the mask keeps the original one. The smoothing intensity is controlled by the parameters, m^{mask} and n^{mask} . Fig. 11 shows the relationship between the parameters and the smoothing effects. For more concise illustration, they are given in 1D or 2D form.

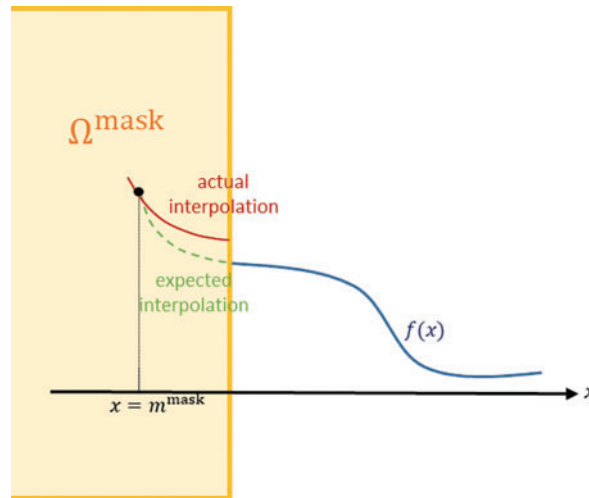


Figure 9: Illustration of the discontinuous gap at $\partial\Omega^{\text{mask}}$

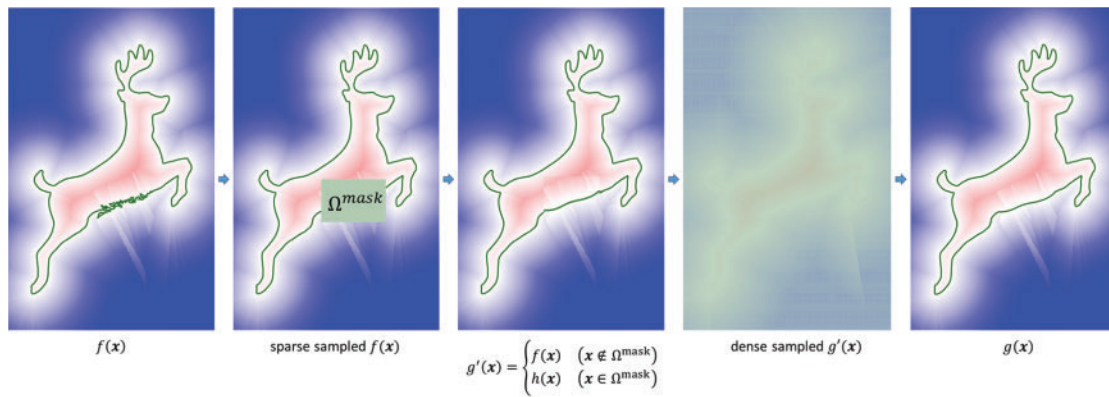


Figure 10: Illustration of the proposed algorithm of noise reduction

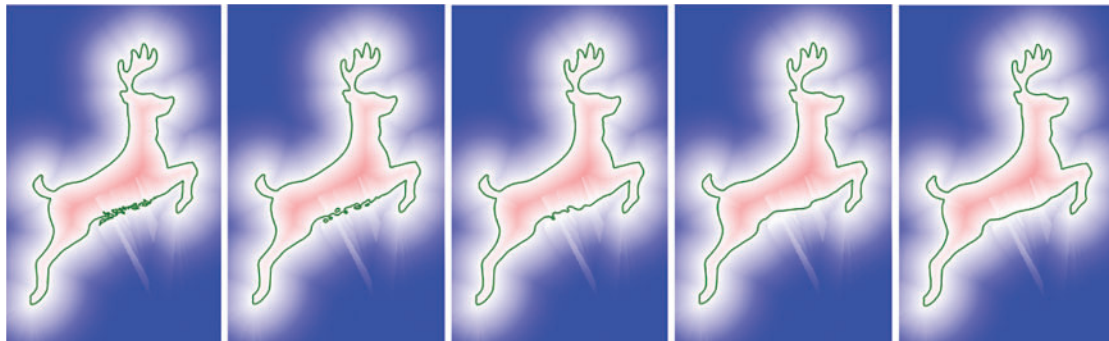


Figure 11: Examples of de-noise processes with different sparse sampling resolutions. From left to right: noised curve, de-noised curve applied 32×32 , 16×16 , 8×8 and, 4×4 discrete samples, respectively, in sparse sampling

5 Experiments

This section presents some experimental examples to evaluate the proposed methods. Three-dimensional restoration examples are provided, using methods discussed in Section 4 and extended to three-dimensional formulations. Fig. 12 shows the data points of “Venus,” “Nefertiti,” and “Happy Buddha.” The implicit surfaces are generated using the grid of polynomial approach [13,16].



Figure 12: Test examples of a surface with irregular parts with respective mask specification. Left: surface “Venus” with the redundant part. Middle: Surface “Nefertiti” with noise part; the data point set is provided by Rapidform (INUS Technology, Inc., USA) [41]. Right: surface “Happy Buddha” with noisy parts around the entire domain; the data point set is provided by Stanford 3D Scanning Repository

Fig. 13 shows restoration examples of removing redundant parts. The redundant part is considered as the protruding part at the bottom of the statue “Venus,” and the user-specified mask covers this area. With different settings of the gradient α at the top surface of the specified mask, different shapes of restoration can be obtained. A flat shape can be obtained if a larger number of the gradient α is used, as shown in Fig. 13 (left), whereas a shape with larger roundness can be obtained if the gradient α is getting smaller, as shown in Fig. 13 (right).

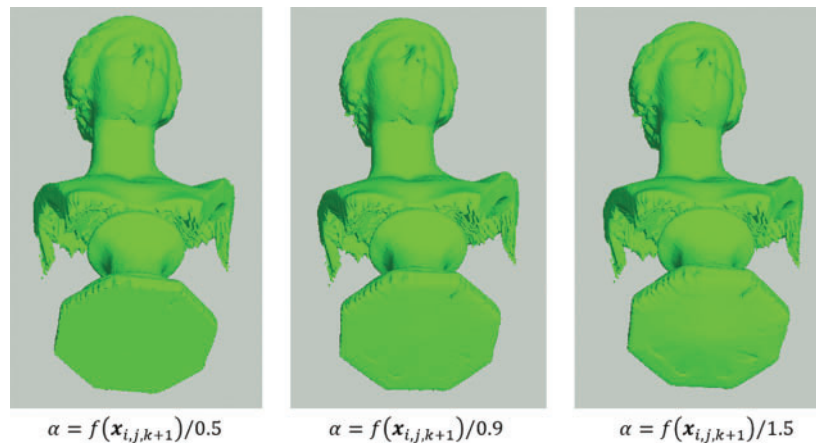


Figure 13: Examples of removing redundant parts

Figs. 14 and 15 show restoration examples of partial and overall de-noise, respectively. The resolution of coarse sampling influences the intensity of de-noise, and setting a smaller number of sampling density results in a larger intensity of de-noise. However, the proposed method works well

only when there are only a few smaller noisy parts, as shown in Fig. 14, in which case only a smaller mask is required to make corrections. If a de-noise process across the entire domain is required, the proposed method is less robust because it is difficult to control the intensity of de-noise to provide a well-balanced setting capable of removing noisy parts while preserving other details. As a result, there is a trade-off between the details of restoration and the intensity of de-noise. This limitation is vital in the case shown in Fig. 15.

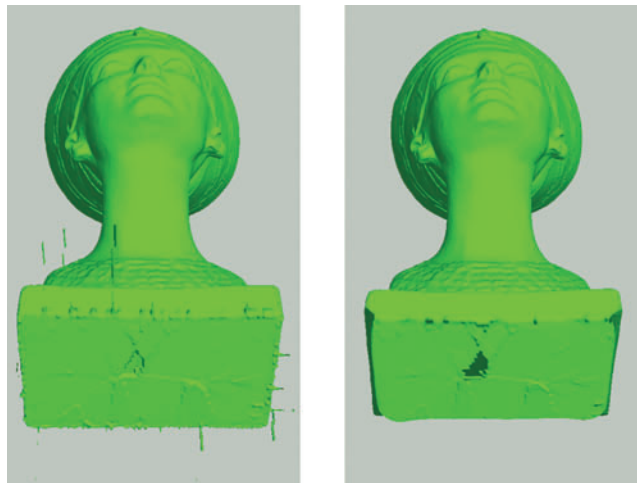


Figure 14: Examples of removing noisy components around a small region at the bottom of the statue. The original and restored surfaces are given by the left and right figures, respectively. The density of sparse sampling is (30, 30, 30)

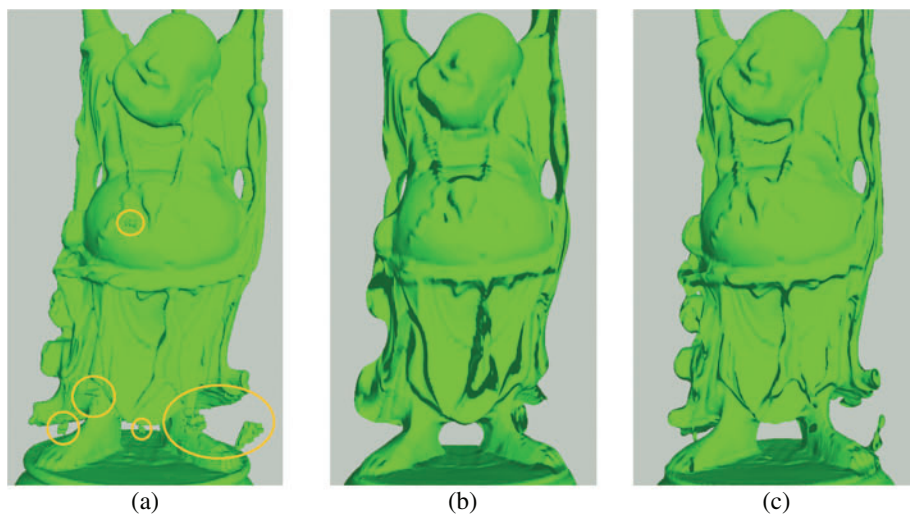


Figure 15: Examples of the overall processing of de-noise. The original surface is shown in (a). Less-detailed with higher de-noise intensity is given in (b), after setting coarse sampling resolution as (20, 20, 20). Noisy parts have not been completely removed in (c), after setting coarse sampling resolution as (40, 40, 40), but overall details have been well-preserved

6 Conclusion and Future Work

This paper presents many methods to correct three-dimensional implicit surfaces with irregular components, such as redundant protruding parts and jaggy noise. Sections 1 and 2 introduce implicit surfaces and some state-of-the-art generation methods. Section 3 offers observations of irregular parts and their respective causes. Section 4 introduces the proposed correction methods. Experiments in Section 5 demonstrates that the proposed approaches can produce effective corrections, subject to some limitations that will be described further below.

In most situations, users have little knowledge regarding the properties of the field $f(\mathbf{x})$. Good restoration requires the use of appropriate parameters, such as α and the coarse sampling resolution. On the parameter α , the value should be determined depending on the users' requirements. Although visualizing the restored shape will help users to control the parameter, such interface is not developed at this moment and this is an important future work. The same problem applies to the parameter of sampling resolution. In addition, developing 3D GUIs to set masks visually will strongly support the users and this is considered as another important future work. For other limitations, the trade-off between the intensity of noise reduction and the level of restored details discussed in Section 5 has theoretically narrowed down the effectiveness of the proposed method.

Moreover, the proposed methods output the final result, $g(\mathbf{x})$, interpolated by spline, in Eqs. (6) and (9). Such an operation will make the result differ from the original field $f(\mathbf{x})$ outside the mask if the original field is generated by other methods. However, in general, such a difference is minimal and we can consider $g(\mathbf{x})$ as a decent approximation of the original field $f(\mathbf{x})$.

Funding Statement: This work was supported by JSPS KAKENHI Grant No. 21K11928.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Schmidt, R., Wyvill, B., Galin, E. (2005). Interactive implicit modeling with hierarchical spatial caching. *International Conference on Shape Modeling and Applications 2005 (SMI' 05)*, pp. 104–113. <https://doi.org/10.1109/SMI.2005.25>
2. Marschner, S., Shirley, P., Ashikhmin, M., Gleicher, M., Hoffman, N. et al. (2015). *Fundamentals of computer graphics*. 4th edition. USA: A K Peters/CRC Press.
3. Kanetsuki, Y., Nakata, S. (2015). Moving particle semi-implicit method for fluid simulation with implicitly defined deforming obstacles. *Journal of Advanced Simulation in Science and Engineering*, 2(1), 63–75. <https://doi.org/10.15748/jasse.2.63>
4. Kanetsuki, Y., Nakata, S. (2020). Artificial force free boundaries: Particle-based fluid simulation with implicit surfaces. In: *Computational and experimental simulations in engineering*, pp. 207–222. Springer, Cham, Switzerland. https://doi.org/10.1007/978-3-030-27053-7_20
5. Sonon, B., François, B., Massart, T. J. (2015). An advanced approach for the generation of complex cellular material representative volume elements using distance fields and level sets. *Computational Mechanics*, 56(2), 221–242. <https://doi.org/10.1007/s00466-015-1168-8>
6. Yang, N., Wang, S., Gao, L., Men, Y., Zhang, C. (2017). Building implicit-surface-based composite porous architectures. *Composite Structures*, 173(4), 35–43. <https://doi.org/10.1016/j.compstruct.2017.04.004>
7. Parulek, J., Viola, I. (2012). Implicit representation of molecular surfaces. *2012 IEEE Pacific Visualization Symposium*, pp. 217–224. Songdo.

8. Liu, Y., Song, Y., Yang, Z., Deng, J. (2017). Implicit surface reconstruction with total variation regularization. *Computer Aided Geometric Design*, 52(5), 135–153. <https://doi.org/10.1016/j.cagd.2017.02.005>
9. Kazhdan, M., Bolitho, M., Hoppe, H. (2006). Passion surface reconstruction. *Symposium on Geometry Processing*, pp. 61–70. Cagliari. <https://doi.org/10.2312/SGP/SGP06/061-070>
10. Wang, J., Yu, Z., Zhu, W., Cao, J. (2013). Feature-preserving surface reconstruction from unoriented, noisy point data. *Computer Graphics Forum*, 32(1), 164–176. <https://doi.org/10.1111/cgf.12006>
11. Liu, S., Wang, C. C. L. (2012). Quasi-interpolation for surface reconstruction from scattered data with radial basis function. *Computer Aided Geometric Design*, 27(7), 435–447. <https://doi.org/10.1016/j.cagd.2012.03.011>
12. Drake, K. P., Fuselier, E. J., Wright, G. B. (2022). Implicit surface reconstruction with a curl-free radial basis function partition of unity method. arXiv: 2101.05940.
13. Nakata, S., Aoyama, S., Makino, R., Hasegawa, K., Tanaka, S. (2012). Real-time isosurface rendering of smooth fields. *Journal of Visualization*, 15(2), 179–187. <https://doi.org/10.1007/s12650-011-0119-5>
14. Itoh, T., Nakata, S. (2015). Fast generation of smooth implicit surface based on piecewise polynomial. *Computer Modeling in Engineering & Sciences*, 107(3), 187–199. <https://doi.org/10.3970/cmcs.2015.107.187>
15. Vemuri, B., Bolle, R. (1991). On three-dimensional surface reconstruction methods. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(1), 1–13. <https://doi.org/10.1109/34.67626>
16. Poursaeed, O., Fisher, M., Aigerman, N., Kim, V. G. (2020). Coupling explicit and implicit surface representations for generative 3D modeling. In: *Lecture notes in computer science*, vol. 12355, pp. 667–683. Springer, Cham, Switzerland. https://doi.org/10.1007/978-3-030-58607-2_39
17. Turk, G., O'Brien, J. F. (2002). Modelling with implicit surfaces that interpolate. *ACM Transactions on Graphics*, 21(4), 855–873. <https://doi.org/10.1145/571647.571650>
18. Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R. et al. (2001). Reconstruction and representation of 3D objects with radial basis functions. *Proceedings of the 28th annual conference on Computer Graphics and interactive techniques (SIGGRAPH '01)*, pp. 67–76. Los Angeles.
19. Zeng, Y., Zhu, Y. (2022). Implicit surface reconstruction based on a new interpolation/approximation radial basis function. *Computer Aided Geometric Design*, 92(4), 102062. <https://doi.org/10.1016/j.cagd.2021.102062>
20. Ohtake, Y., Alexander, B., Alexa, M., Turk, G., Seidel, H. P. (2003). Multi-level partition of unity implicits. *ACM Transactions on Graphics*, 22(3), 463–470. <https://doi.org/10.1145/882262.882293>
21. Kazhdan, M. (2005). Reconstruction of solid models from oriented point sets. *Proceedings of the Third Eurographics Symposium on Geometry Processing (SIGGRAPH '05)*, pp. 73–82. Vienna.
22. Manson, J., Petrova, G., Schaefer, S. (2008). Streaming surface reconstruction using wavelets. *Computer Graphics Forum*, 27(5), 1411–1420. <https://doi.org/10.1111/j.1467-8659.2008.01281.x>
23. Berger, M., Levine, J. A., Nonato, L. G., Taubin, G., Silva, C. T. (2013). A benchmark for surface reconstruction. *ACM Transaction on Graphics*, 32(2), 1–17. <https://doi.org/10.1145/2451236.2451246>
24. Berger, M., Tagliasacchi, A., Seversky, L. M., Alliez, P., Guennebaud, G. et al. (2017). A survey of surface reconstruction from point clouds. *Computer Graphics Forum*, 36(1), 301–329. <https://doi.org/10.1111/cgf.12802>
25. Erler, P., Guerrero, P., Ohrhallinger, S., Mitra, N. J., Wimmer, M. (2020). Points2Surf: Learning implicit surfaces from point clouds. *European Conference on Computer Vision*, pp. 108–124. Glasgow.
26. Liu, S., Guo, H., Pan, H., Wang, P., Tong, S. et al. (2021). Deep implicit moving least-squares functions for 3D reconstruction. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1788–1797.
27. Ma, B., Han, Z., Liu, Y., Zwicker, M. (2021). Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. *Proceedings of the 38th International Conference on Machine Learning*, pp. 7246–7257.

28. Ma, B., Liu, Y., Han, Z. (2022). Reconstructing surfaces for sparse point clouds with on-surface priors. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6315–6325. New Orleans.
29. Jia, M., Kyan, M. (2010). Learning occupancy function from point clouds for surface reconstruction. arXiv: 11378.
30. Martel, J. N. P., Lindell, D. B., Lin, C. Z., Chan, E. R., Monteiro, M. et al. (2021). ACORN: Adaptive coordinate networks for neural scene representation. arXiv: 2105.02788.
31. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A. (2020). Convolutional occupancy networks. *European Conference on Computer Vision*, pp. 523–540. Glasgow.
32. Takikawa, T., Litalien, J., Yin, K., Kreis, K., Loop, C. et al. (2021). Neural geometric level of detail: Real-time rendering with implicit 3D shapes. arXiv: 2101.10994.
33. Ladický, L., Saurer, O., Jeong, S., Maninchedda, F., Pollefeys, M. (2017). From point clouds to mesh using regression. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3913–3922. Venice.
34. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
35. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A. (2019). Occupancy networks: Learning 3D reconstruction in function space. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4460–4470. Long Beach.
36. Huang, Z., Wen, Y., Wang, Z., Ren, J., Jia, K. (2022). Surface reconstruction from point clouds: A survey and a benchmark. arXiv: 2205.02413.
37. Sugihara, M., Wyvill, B., Schmidt, R. (2010). WarpCurves: A tool for explicit manipulation of implicit surfaces. *Computers & Graphics*, 34(4), 282–291. <https://doi.org/10.1016/j.cag.2010.03.008>
38. Eyiyurekli, M., Breen, D. (2010). Interactive free-form level-set surface-editing operators. *Computers & Graphics*, 34(5), 621–638. <https://doi.org/10.1016/j.cag.2010.06.006>
39. Reiner, T., Mückl, G., Dachsbacher, C. (2011). Interactive modeling of implicit surfaces using a direct visualization approach with signed distance functions. *Computers & Graphics*, 35(3), 596–603. <https://doi.org/10.1016/j.cag.2011.03.010>
40. Ascher, U. M., Greif, C. (2011). *A first course in numerical methods*. USA: Society for Industrial and Applied Mathematics.
41. Kim, D. B., Kang, E. C., Lee, K. H., Pajarola, R. B. (2006). Framework for adaptive sampling of point-based surfaces using geometry and color attributes. *Proceedings of the International Conference on Computational Science*, pp. 371–374. Reading, UK.