

Filling the Signed Distance Field by Fitting Local Quadrics

Takeshi Masuda
Intelligent Systems Institute
National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba, Ibaraki 305-8568 Japan
masuda@ieee.org

Abstract

We propose a method of filling unmeasured regions of shape models integrated from multiple measurements of surface shapes. We use the signed distance field (SDF) as shape representation that contains information of the surface normal along with the signed distance at the closest point on the surface from the sampling point. We solve this problem by iteratively fitting quadratic function to generate smoothly connected SDF. We analyzed the relationship between the quadratic coefficients and the surface curvature, and by using the coefficients, we evenly propagated the SDF so that it satisfies the constraints of the field. The proposed method was tested on synthetic data and real data that was generated by integrating multiple range images.

1 Introduction

In recent years, computing and sensing devices have been rapidly improved their performance and availability. With these devices, many methods for measuring object shapes have been proposed. Every sensing method has measurement limitation which comes from sensing principle and device layout. For example, with active methods that measure shapes by transmitting structured or organized light, it is impossible to measure mirror-like specular surface, black surface and occluded regions by obstacles, because the camera cannot receive the reflection of the transmission. With passive methods like stereoscopy, it is impossible to measure surfaces without texture. Measurable volume is limited by device specification and device layout. Sensors can be placed only in free space, and it is known that there are cases that requires infinitely many measurements to cover the whole surface of an object even though the shape is very simple. The regions whose shape cannot or is hard to be measured is needed to be interpolated in many real applications.

In the field of computer vision, many methods have been proposed to interpolate sparse surface obtained by binocular stereo[1, 2, 3]. These methods were formalized by regu-

larizing the fitting energy with the smoothness energy. The smoothness energy is determined by surface curvatures, and these method generate a reasonable smooth surface whose total curvature is minimized balancing the fitting energy to the input sparse depth map.

Unlike these methods processed on 2.5-D space, there is also an approach of surface reconstruction on 3-D space. Since the most versatile and useful representation of surface is polygon meshes, there are researches that estimate polygon meshes from measured point clouds[4]. It is hard to automatically generate surface that is consistent with topological or geometrical requirements, there are researches to fix the mesh to satisfy requirements[5].

The author has used the regularly sampled signed distance field (SDF) to represent 3-D shapes[6]. The SDF is a scalar field whose value is the signed distance from the object surface. It was originally introduced for integration of a shape model from registered multiple shape measurements. To generate a smooth surface, signed distances of input shapes are averaged at each voxel, and a marching cube algorithm is applied to extract the isosurface as polygon meshes [7, 8, 9]. The surface reconstructed by the marching cube algorithm usually contains topological faults that should be fixed by post-processing. Euclidean distance transformation (EDT) [10] has some similarity with the SDF. Though the SDF uses the signed distance in sub-voxel resolution, the distance of EDT is measured between voxels which is unsigned.

Volumetric integration was originally implemented by occupancy voting on voxels in three classes: 'object' (inside), 'surface', 'space' (outside)[11]. The unmeasured region was automatically extracted as an interface of 'object' and 'space' voxels without 'surface' voxels between them. This technique requires not only registration of range images but also sensor configurations of the projector and camera. A similar technique was applied on surface reconstruction using the SDF [8]. The unmeasured region was simply filled by classifying voxels into three classes: 'unseen' (inside), 'near surface', 'empty' (outside).

The level-set method was introduced to reconstruct

smooth surface from sparse or noisy range images [12]. This method represent the spacial transition of interface about time by differential equations. It was applied to fill holes of the SDF smoothly [13], and to fix inconsistency of the SDF [14]. The size of gradient of the potential field generated by the level-set method is not constant in general, because the speed function is dependent on the curvature. This is different to the SDF whose gradient is the unit surface normal.

The author proposed a unified framework in which the SDF representation was used to solve integration together with registration simultaneously[6]. In the existing researches, these problems have been formulated separately, and the data shapes are preprocessed to remove erroneous measurements, and registered before integrated. Integration incorporated with registration is necessary to detect erroneous measurements or non-overlapping regions by automatic comparison of multiple-views without heuristic criteria and human operations. Polygon meshes useful for rendering and applications are generated from the SDF by a similar method as the marching cubes.

In this paper, we propose an algorithm that interpolate the unmeasured regions in the SDF. The supposed input is the SDF generated by our integration and registration algorithm mentioned above. In the rapid prototyping industry, it is usual that defects are processed by manual operation with interactive GUI, and it is not hard to fill a few flat small holes. Thus, we mainly focus on defects on curved surfaces, and the holes we fill is larger than those in the result of the marching cubes. To smoothly interpolate the SDF, we use quadratic approximation of the SDF. From the quadratic approximation coefficients, we can estimate the differential geometrical properties like surface curvatures. This can fix the inconsistency that remains in the result of our integration method which uses linear approximation of the SDF.

The definitions and basic properties of the SDF are explained in Sec. 2.1. Its differential geometry related the surface curvature is analyzed and estimation of lacked sample by quadratic approximation of SDF is introduced in Sec. 3.1. Finally, we show the experimental results on a synthetic and integrated SDF generated from real range images in Sec. 4.

2 Signed Distance Field

2.1 Definition

Assume that the object surface is an oriented curved surface S , the closest point on the surface from an arbitrary 3-D point p in general position is determined uniquely. Exceptions are the cases that the point p is equidistant to multiple parts of the object surface, for example, at the center

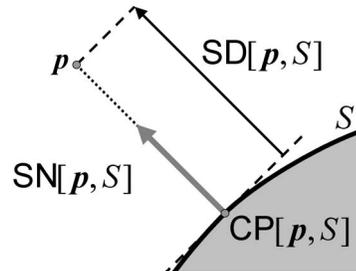


Figure 1: On an oriented 3-D curved surface S , the closest point $CP[p, S]$ is determined uniquely from a 3-D sampling point p at a general position. The closest point is at the foot of the surface normal $SN[p, S]$, and the sign of the distance $SD[p, S]$ is determined by on which side of the surface the sampling point exists.

of a sphere, on the axis of cylinder, on the equidistant plane from two planes, and on the Voronoi boundary in general. The closest point is represented by

$$CP[p, S] = \operatorname{argmin}_{r \in S} \|r - p\|^2. \quad (1)$$

If the surface is smooth and differentiable, the direction of the vector pointing from the closest point $CP[p, S]$ to the sampling point p coincides with the surface normal $SN[p, S]$. The surface normal is normalized such that $\|SN[p, S]\| = 1$, and its direction is taken from inside to outside of the object (Fig. 1). The signed distance is represented by

$$SD[p, S] = SN[p, S]^T (p - CP[p, S]), \quad (2)$$

which forms a scalar field whose gradient is the vector field of the surface normal:

$$SN[p, S] = \operatorname{grad}_p SD[p, S]. \quad (3)$$

We denote an SDF sample about p by

$$SDF[p, S] = \{p, CP[p, S], SN[p, S], SD[p, S]\},$$

and the signed distance at x in the neighborhood of p is linearly approximated by

$$SD[x, S] = SN[p, S]^T (x - p) + SD[p, S]. \quad (4)$$

Hereinafter, as long as we treat only one SDF field sampled at p on the surface S , we shorten the notations $CP[p, S]$, $SN[p, S]$, $SD[p, S]$ by c, n, s respectively.

2.2 Distance between Two SDFs

Suppose that there are two SDFs $SDF[p, A]$ and $SDF[p, B]$ at p . We determine the distance between them by integrating squared difference of linearly extrapolated signed distances within the neighborhood around p . The distance can

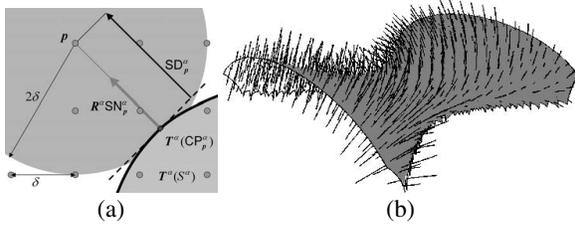


Figure 2: (a) The object surface S is sampled to determine the discrete SDF samples (2-D analogy). (b) Illustration of vectors from $CP[p, T^\alpha(S^\alpha)]$ to p . Closest points found on the boundary are not included.

be simplified by the weighted sum of squared differences of the surface normal and the signed distance as [6]

$$d^2(\text{SDF}[p, A], \text{SDF}[p, B]) = w_n \|\text{SN}[p, A] - \text{SN}[p, B]\|^2 + (\text{SD}[p, A] - \text{SD}[p, B])^2, \quad (5)$$

where w_n is the balancing weight depending on the size and shape of the neighborhood. We take a cube of edge length δ as the neighborhood, and then $w_n = \delta^2/12$.

2.3 Sampling

The SDF is discretized by taking discrete sampling points p . We take a cube enough large to include the object which is bounded by the points $\mathbf{o} = \{o_x, o_y, o_z\}$ and $\{o_x + W, o_y + W, o_z + W\}$. Within this sampling region, each sampling point p is located at a regular lattice point whose interval is δ (Fig. 2). For implementation convenience, we take the size of the sampling region as $W = 2^l \delta$, where l is a positive integer denoting the sampling level. For each integer index $\mathbf{i} = \{i_x, i_y, i_z\}$ ($0 \leq i_x, i_y, i_z \leq 2^l - 1$), the sampling point is located at $p_i = \mathbf{o} + (\mathbf{i} + \{0.5, 0.5, 0.5\})\delta$. The SDF samples whose absolute distance is larger than a thickness threshold ($= 2\delta$ in the implementation) or whose closest points are located on the boundary of the measured region are not included as a valid sample.

2.4 Surface Reconstruction

From the SDF, we generate a triangulated surface model that can be used for practical applications. We consider a cubic cell surrounded by eight sampling points $p_{[i-j]}$ where $\mathbf{j} = \{j_x, j_y, j_z\}$ is an integer vector whose components are 0 or 1, and its center is at $\mathbf{o} + \mathbf{i}\delta$. The signed distance at the corners are extrapolated to the center by 4, and according to the sign of the averaged signed distance at the center, the cell is classified in inside or outside of the object. The bounding faces of the cubes of different signs are extracted (Fig. 3 (a)), and holding the structure, the coordinates of

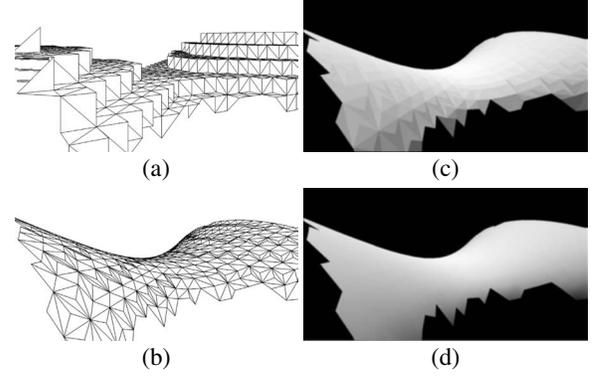


Figure 3: Surface reconstruction steps.

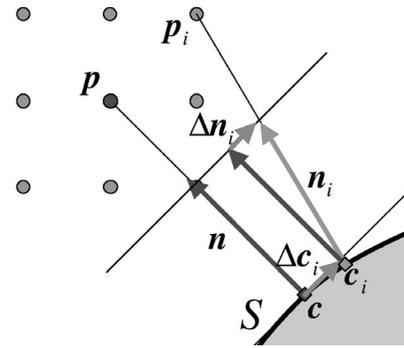


Figure 4: Curvature estimation from the neighborhood of a sampling point p .

the vertices are mapped from p to the corresponding closest point CP_p (Figs. 3 (b), (c)).

This modeling method directly uses SDF properties, and interpolation of the vertex coordinates is not necessary unlike the marching cube algorithm [8, 7, 9]. The surface normal SN_p is also attached to each vertex, which make the surface rendering smoother and quicker (Fig. 3 (d)).

3 Differential Geometry of SDF

3.1 Surface Curvature

We can determine the surface curvatures from the SDF shape representation[15]. Let p be the central sampling point at which we are determining the curvatures, and c and n be the closest point and surface normal at p respectively. Let c_i and n_i be the closest point and the surface normal at a sampling point $p_i = p + s_i$ in the neighborhood of p . The tangent elements of these displacements according to

the transition from the center is represented by

$$\begin{aligned}\Delta_n \mathbf{c} &= P_n \cdot \Delta \mathbf{c} = (\Delta_n c_x, \Delta_n c_y, \Delta_n c_z), \\ \Delta_n \mathbf{n} &= P_n \cdot \Delta \mathbf{n} = (\Delta_n n_x, \Delta_n n_y, \Delta_n n_z),\end{aligned}$$

where $P_n = I - \mathbf{n} \cdot \mathbf{n}^T$ is the projection matrix in the direction of the surface normal \mathbf{n} (Fig. 4).

Gaussian curvature is known as “the total spread of normal directions per unit surface area” [16]. This means that curvature information can be extracted from the relationship between the transitions of closest points and surface normals. Assuming that the relationship is approximated by a symmetric matrix \mathbf{K} , which has 6 independent components $k_{xx}, k_{yy}, k_{zz}, k_{yz}, k_{zx}, k_{xy}$, we can establish equations for the transitions of closest points and surface normals in the neighborhood of \mathbf{p} .

$$\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \Delta_n c_{ix} & 0 & 0 & 0 & \Delta_n c_{iz} & \Delta_n c_{iy} \\ 0 & \Delta_n c_{iy} & 0 & \Delta_n c_{iz} & 0 & \Delta_n c_{ix} \\ 0 & 0 & \Delta_n c_{iz} & \Delta_n c_{iy} & \Delta_n c_{ix} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \cdot \begin{bmatrix} k_{xx} \\ k_{yy} \\ k_{zz} \\ k_{yz} \\ k_{zx} \\ k_{xy} \end{bmatrix} = \begin{bmatrix} \vdots \\ \Delta_n n_{ix} \\ \Delta_n n_{iy} \\ \Delta_n n_{iz} \\ \vdots \end{bmatrix} \quad (6)$$

These equations can be solved by the singular value decomposition to obtain the least square solution of \mathbf{K} , which we denote $\hat{\mathbf{K}}$.

Because the matrix $\hat{\mathbf{K}}$ is real symmetric, it can be decomposed with real eigenvalues by

$$\hat{\mathbf{K}} = \mathbf{U} \cdot \begin{bmatrix} \kappa_1 & 0 & 0 \\ 0 & \kappa_2 & 0 \\ 0 & 0 & \kappa_3 \end{bmatrix} \cdot \mathbf{U}^T, \quad (7)$$

where $\mathbf{U} = [\mathbf{w}_1 | \mathbf{w}_2 | \mathbf{w}_3]$ is an orthogonal matrix. The transitions $\Delta_n \mathbf{c}_i$ and $\Delta_n \mathbf{n}_i$ are on the tangent plane, there is an orthogonal base that satisfies $\mathbf{w}_3 = \pm \mathbf{n}$. The corresponding eigenvalue κ_3 can take any value, and it is possible to set $\kappa_3 = 0$.

This computation is equivalent to project \mathbf{K} by multiplying the projection matrix P_n . The matrix $\hat{\mathbf{K}}$ is normalized to obtain $\bar{\mathbf{K}}$ by

$$\bar{\mathbf{K}} = \hat{\mathbf{K}} \cdot P_n, \quad (8)$$

and the principle curvatures can be obtained from the relationship

$$\text{tr}(\bar{\mathbf{K}}) = \kappa_1 + \kappa_2, \quad \|\bar{\mathbf{K}}\|_F^2 = \kappa_1^2 + \kappa_2^2, \quad (9)$$

where $\|\cdot\|_F$ signifies Frobenius norm of a matrix. This method can be applied on shape representations other than the SDF as long as pairs of $\Delta_n \mathbf{c}_i$ and $\Delta_n \mathbf{n}_i$ are available.

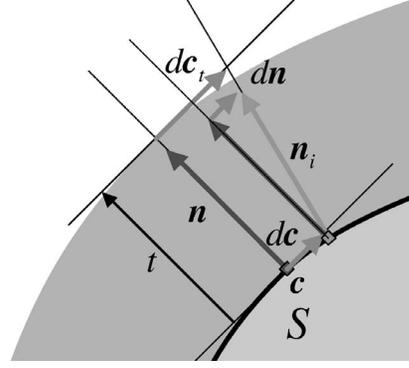


Figure 5: Surface element on the parallel surface.

3.2 Parallel Surface

Assume that the surface element on the object surface around the closest point \mathbf{c} is represented by

$$d\mathbf{c} = \theta^1 \mathbf{w}_1 + \theta^2 \mathbf{w}_2, \quad (10)$$

where \mathbf{w}_1 and \mathbf{w}_2 are the principal directions. The transition of surface normal is represented by using the principal curvatures by

$$d\mathbf{n} = \kappa_1 \theta^1 \mathbf{w}_1 + \kappa_2 \theta^2 \mathbf{w}_2. \quad (11)$$

The closest point on the parallel surface whose distance is t from the object surface is $\mathbf{c}_t = \mathbf{c} + t\mathbf{n}$ (Fig. 5), and its transition is represented by

$$d\mathbf{c}_t = d\mathbf{c} + t d\mathbf{n} = (1 + t\kappa_1) \theta^1 \mathbf{w}_1 + (1 + t\kappa_2) \theta^2 \mathbf{w}_2. \quad (12)$$

The surface element of the parallel surface is given by

$$(1 + t\kappa_1)(1 + t\kappa_2) d\theta^1 d\theta^2. \quad (13)$$

Comparing coefficients of Eqs. (11) and (12), the principal curvatures on the parallel surface are given by

$$\kappa_1(t) = \frac{\kappa_1}{1 + t\kappa_1}, \quad \kappa_2(t) = \frac{\kappa_2}{1 + t\kappa_2}. \quad (14)$$

The volume on the object surface element within the thickness $\pm T$ is given by

$$\int_{-T}^T (1 + t\kappa_1) \cdot (1 + t\kappa_2) dt = 2T + \frac{2K}{3} T^3, \quad (15)$$

if the radii of curvature are larger than T . This result signifies that the surface density of the closest point is dependent on the Gaussian curvature if the sampling points distribute evenly in 3-D space. Compared to the flat plane, the relative surface density change is $\pm 1/3$ when the thickness $T = 2\delta$ and $K = \pm 1/(2\delta)^2$.

3.3 Quadratic Approximation of the SDF

We have shown the linear approximation within the neighborhood of a sampling point in Sec. 2.1, and that the surface curvature can be estimated by linear approximation of the relationship between the transitions of closest points and surface normals. To estimate the SDF properties at a lacked sampling point smoothly, it is necessary to approximate the SDF by a quadratic function.

Assume that the SDF is approximated by a quadratic function about the transition of the sampling point $\Delta\mathbf{p}_i$ by

$$s(\Delta\mathbf{p}; \mathbf{H}, \mathbf{n}, s) = \frac{1}{2} \Delta\mathbf{p}^T \mathbf{H} \Delta\mathbf{p} + \mathbf{n}^T \Delta\mathbf{p} + s, \quad (16)$$

and the surface normal field is linearly approximated by

$$\mathbf{n}(\Delta\mathbf{p}; \mathbf{H}, \mathbf{n}) = \mathbf{H} \Delta\mathbf{p} + \mathbf{n}, \quad (17)$$

because the surface normal field is the gradient of the SDF as Eq. (3). The Hessian matrix \mathbf{H} is symmetric because the SDF is a scalar potential and the surface normal field is irrotational. We denote the quadratic approximation coefficients $\mathbf{H}, \mathbf{n}, s$ about \mathbf{p} by QSDF(\mathbf{p}).

The quadratic approximation coefficients QSDF(\mathbf{p}) are estimated by minimizing following criteria,

$$E(\mathbf{p}; \text{QSDF}(\mathbf{p})) = \sum_{\mathbf{p}_i \in N(\mathbf{p})} w_i d^2(\text{SDF}[\mathbf{p}_i, S], \text{QSDF}(\mathbf{p}_i)), \quad (19)$$

where $d^2(\cdot)$ is the distance between two SDFs given by Eq. (6), w_i is the weighting value, $\text{SDF}[\mathbf{p}_i, S]$ is the SDF sample, and $\text{QSDF}(\mathbf{p}_i)$ is the interpolated SDF by the quadratic approximation. Likewise Eq. (6), we can establish equations about 10 coefficients, including 6 independent components of symmetric matrix \mathbf{H} , surface normal \mathbf{n} and signed distance s , for each sampling point as Eq. (18), where $w_{ni} = w_i w_n$. These equations can also be solved by the singular value decomposition.

As seen in Fig. 1, there is a relationship $\mathbf{c} = \mathbf{p} - s \cdot \mathbf{n}$, and by differentiating this equation by \mathbf{p} , we obtain

$$\frac{\partial \mathbf{c}}{\partial \mathbf{p}} = \mathbf{I} - \frac{\partial s}{\partial \mathbf{p}} \cdot \mathbf{n}^T - s \cdot \frac{\partial \mathbf{n}}{\partial \mathbf{p}} = P_n - s \cdot \mathbf{H}. \quad (20)$$

As explained in Sec. 3.1, $\mathbf{K} = \partial \mathbf{n} / \partial \mathbf{c}$ and $P_n \mathbf{K} = \mathbf{K}$, then

$$\mathbf{H} = \mathbf{K}(\mathbf{I} + s\mathbf{K})^{-1}. \quad (21)$$

By using Eqs. (7) and (14), the matrix \mathbf{H} is decomposed by

$$\mathbf{H} = \mathbf{U} \begin{bmatrix} \kappa_1(s) & 0 & 0 \\ 0 & \kappa_1(s) & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{U}^{-1}. \quad (22)$$

This signifies that the matrix \mathbf{H} is symmetric of rank 2, and whose eigenvalues are the principal curvatures on the parallel surface. This result means that from the matrix \mathbf{H} and the signed distance s , we can estimate the surface curvatures.

We denote the solution obtained from Eq. (18) by $\hat{\mathbf{H}}, \hat{\mathbf{n}}, \hat{s}$, and to satisfy the constraints mentioned above, we normalize these estimates. First, the length of the surface normal vector is normalized by

$$\bar{\mathbf{n}} = \frac{\hat{\mathbf{n}}}{\|\hat{\mathbf{n}}\|}. \quad (23)$$

Then, the Hessian matrix \mathbf{H} is projected on the tangent plane by

$$\bar{\mathbf{H}} = P_{\bar{\mathbf{n}}} \hat{\mathbf{H}} P_{\bar{\mathbf{n}}}, \quad (24)$$

where $P_{\bar{\mathbf{n}}} = \mathbf{I} - \bar{\mathbf{n}} \cdot \bar{\mathbf{n}}^T$. Finally, the signed distance is adjusted by

$$\bar{s} = \frac{\sum_{\mathbf{p}_i \in N(\mathbf{p})} w_i (s_i - (\frac{1}{2} \Delta\mathbf{p}^T \bar{\mathbf{H}} \Delta\mathbf{p} + \bar{\mathbf{n}}^T \Delta\mathbf{p}))}{\sum_{\mathbf{p}_i \in N(\mathbf{p})} w_i}. \quad (25)$$

The new SDF at \mathbf{p} is determined by $\overline{\text{SDF}}[\mathbf{p}] = \{\mathbf{p}, \mathbf{p} - s\bar{\mathbf{n}}, \bar{s}\}$. By fitting QSDF(\mathbf{p}) on the neighboring SDFs around the central sampling point \mathbf{p} , we can determine the new SDF samples at \mathbf{p} even when the SDF samples are lacked at \mathbf{p} . We estimate the new value by solving linear equations, which does not require preparing different forms of filters for various types of data missing like [2].

4 Experiments

4.1 Implementation

The proposed method was implemented by the following algorithm:

- 1: Load the input SDF: **org**.
- 2: Initialize the output SDF: **ext** \leftarrow **org**.
- 3: **for** $i = 0$ to N_{iter} **do**
- 4: Dilate: **ext**⁺ \leftarrow **ext** \oplus $\mathbf{w}(\Delta\mathbf{p}_i)$.
- 5: Update: **ext** \leftarrow $\{\overline{\text{SDF}}[\mathbf{p}, \text{org} \cup \text{ext}] \mid \mathbf{p} \in \text{ext}^+\}$.
- 6: Trim: **ext** \leftarrow $\{\text{ext} \mid |s| \leq T\}$
- 7: **enddo**

Once initialized, the input SDF **org** is not changed, and the interpolated shape **ext** is the output. In Step 4, **ext** is extended by dilating with the neighborhood mask to determine the region for computation. In Step 5, for each \mathbf{p} within the dilated region, fit QSDF to **org** and **ext** in the neighborhood, and the new SDF parameters in **ext** are determined. The new SDF can be computed only when enough samples (10 for 3-d cases) are found within the neighborhood. If samples at the same location are found in both **org** and **ext**,

$$\begin{bmatrix} \vdots & \vdots \\ \sqrt{w_{ni}}\Delta p_{ix} & 0 & 0 & 0 & \sqrt{w_{ni}}\Delta p_{iz} & \sqrt{w_{ni}}\Delta p_{iy} & \sqrt{w_{ni}} & 0 & 0 & 0 \\ 0 & \sqrt{w_{ni}}\Delta p_{iy} & 0 & \sqrt{w_{ni}}\Delta p_{iz} & 0 & \sqrt{w_{ni}}\Delta p_{ix} & 0 & \sqrt{w_{ni}} & 0 & 0 \\ 0 & 0 & \sqrt{w_{ni}}\Delta p_{iz} & \sqrt{w_{ni}}\Delta p_{iy} & \sqrt{w_{ni}}\Delta p_{ix} & 0 & 0 & 0 & \sqrt{w_{ni}} & 0 \\ \frac{\sqrt{w_i}}{2}\Delta p_{ix}^2 & \frac{\sqrt{w_i}}{2}\Delta p_{iy}^2 & \frac{\sqrt{w_i}}{2}\Delta p_{iz}^2 & \sqrt{w_i}\Delta p_{iy}\Delta p_{iz} & \sqrt{w_i}\Delta p_{iz}\Delta p_{ix} & \sqrt{w_i}\Delta p_{ix}\Delta p_{iy} & \sqrt{w_i}\Delta p_{ix} & \sqrt{w_i}\Delta p_{iy} & \sqrt{w_i}\Delta p_{iz} & \sqrt{w_i} \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} h_{xx} \\ h_{yy} \\ h_{zz} \\ h_{yz} \\ h_{zx} \\ h_{xy} \\ n_x \\ n_y \\ n_z \\ n_s \end{bmatrix} = \begin{bmatrix} \vdots \\ \sqrt{w_{ni}}n_{ix} \\ \sqrt{w_{ni}}n_{iy} \\ \sqrt{w_{ni}}n_{iz} \\ \sqrt{w_i}s_i \\ \vdots \end{bmatrix} \quad (18)$$

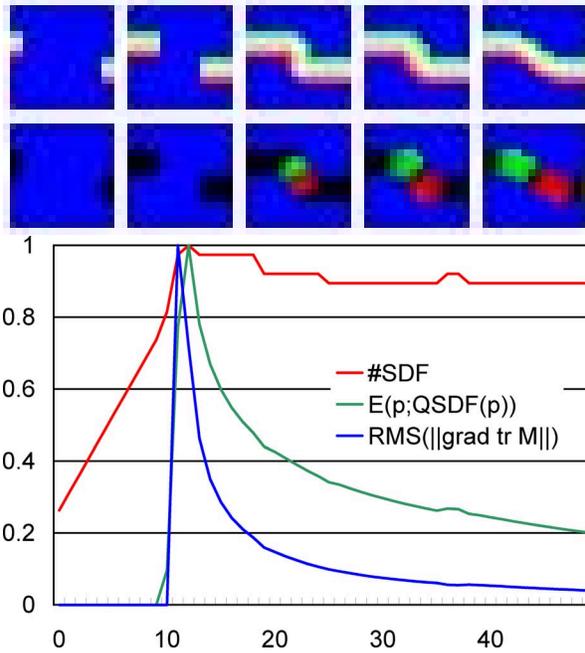


Figure 6: Experiment on a 2-D case of 16×16 pixels. Top left is the SDF map of the initial shape **org** with two short segments on the edge. From left to right, propagated shapes **ext** after 6, 12, 18 and 50 iterations are shown. Corresponding curvature map is shown below each SDF map. Bottom: plot of the number of samples, fitting error and total curvature to the number of iterations. Each plot is normalized by the maximum value.

the sample from **org** is prioritized to **ext**. This is to keep the propagated shape **ext** not travel away from the initial shape **org**. In Step 6, samples whose absolute distance is larger than $T(= 2\delta)$ is trimmed away to maintain the computation near the surface and to reduce the computational cost. By applying one iteration, the boundary is extended by about one voxel width, and holes are gradually filled by propagated SDF.

4.2 Two-dimensional Case

The propagation process in 2-D case is illustrated in Fig. 6. It is observed that, after two propagated edges meet together, the procedure tends to even the curvature. When we use a flat weighting values $w(\Delta p_i) = 1$, the pseudo-inverse of the matrix of the left part of Eq. 18 can be qualitatively expanded by

$$\begin{aligned} \hat{H}_{xx} &= a_{xxs} D_{xx} * S + a_{xxs} D_x * N_x + a_{xxy} D_y * N_y \\ \hat{H}_{yy} &= a_{yys} D_{yy} * S + a_{yys} D_x * N_x + a_{yyx} D_y * N_y \\ \hat{H}_{xy} &= a_{xys} D_{xy} * S + a_{yys} D_y * N_x + a_{yyx} D_x * N_y \\ \hat{N}_x &= a_{xs} D_x * S + a_{xx} 1 * N_x \\ \hat{N}_y &= a_{ys} D_y * S + a_{yy} 1 * N_y \\ \hat{S} &= a_s W * S - a_{sx} D_x * N_x - a_{sy} D_y * N_y, \end{aligned}$$

where $S, N_x, N_y, \hat{H}_{xx}, \hat{H}_{yy}, \hat{H}_{xy}, \hat{N}_x, \hat{N}_y, \hat{S}$ are the image representation of $s_i, n_{ix}, n_{iy}, h_{xx}, h_{yy}, h_{xy}, n_x, n_y, s$ of Eq. 18 respectively, and $*$ represents image filtering with differential filters: $D_{xx}, D_{yy}, D_{xy}, D_x, D_y, 1$: the flat average and W : weighted smoothing filter. These equations can be understood by that $\hat{H}_{xx}, \hat{H}_{yy}$ and \hat{H}_{xy} are determined by a linear combination of differentials of S, N_x and N_y ; \hat{N}_x and \hat{N}_y are the sum of differential of S and the local average of N_x and N_y ; and \hat{S} is determined by the average of linearly extrapolated S . Since the SDF data is not completely occupied, the coefficients and operators can be different from point to point. Because the coefficients are basically determined by linear combination of S, N_x and N_y , the estimates are affected by the input errors by almost linear manner.

4.3 Three-dimensional Results

We applied the proposed method on a synthetic half ellipsoid (Fig. 7). The object surface was sampled by level $l = 6$, and the neighborhood $w(\Delta p_i)$ was Gaussian distribution whose standard deviation was δ in the 32-neighborhood, or $\|\Delta p_i\| \leq 2\delta$. We set the weighting value by

$$w_i = w(\Delta p_i) \cdot \max(1 + s\kappa_1, 0) \cdot \max(1 + s\kappa_2, 0) \quad (26)$$

in the experiments. The surface was generated from the output SDF **ext** by the reconstruction method explained in Sec. 2.4.

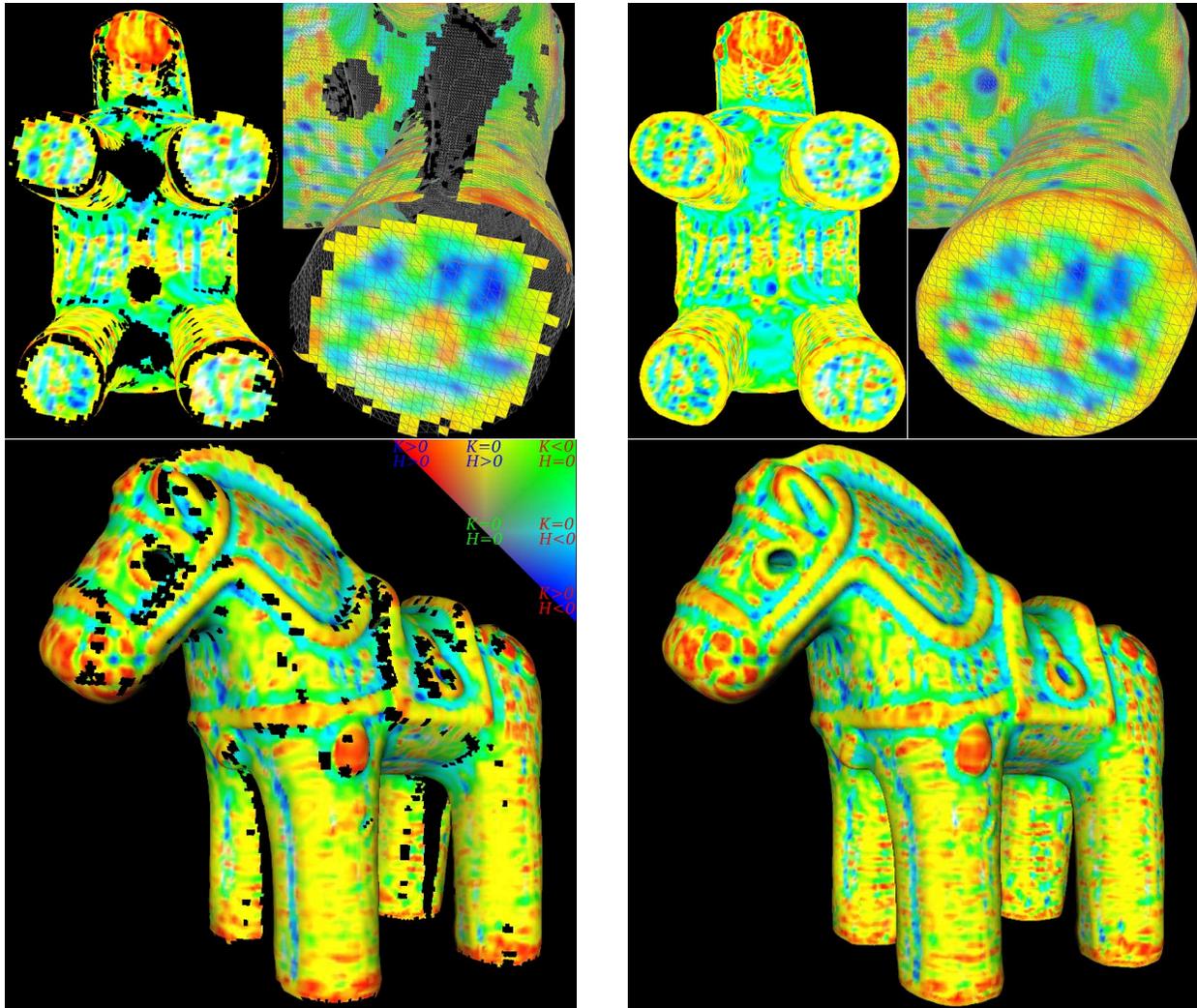


Figure 8: Experiment on the horse figure data. The initial SDF (left) was smoothly interpolated by the proposed method (right). Color corresponds the surface curvature estimated by the quadratic approximation of the SDF.

Without knowledge of the global shape, the rest half was smoothly filled by iterative local process. The filled part is smoothly connected to the original shape, but there is no guarantee that its shape is an ellipsoid. Plots of fitting error and number of sampling points used for computation shows that they increased until the hole was closed by about 40 iterations, and then they gradually decreased to converge.

The proposed method was tested on the SDF generated by our integration method [6] from the range images of a horse figure at the sampling level $l = 8$. The initial SDF was generated intendedly without linear SDF extrapolation which is included in our integration method, and there are many holes including large holes between the legs which

was left unmeasured due to measurement difficulty. After 20 iterations, the holes were completed by a continuous surface. It can be observed that the generated surface is connected to the initial surface continuously by means of curvature also. Computation time was about 1 hour on 1.7GHz CPU.

5 Conclusion

We have developed a method for completing the lack of the shape model by fitting local quadratic function on the SDF shape representation. We have analyzed the relationship between the fitting coefficients and the surface curvatures, and

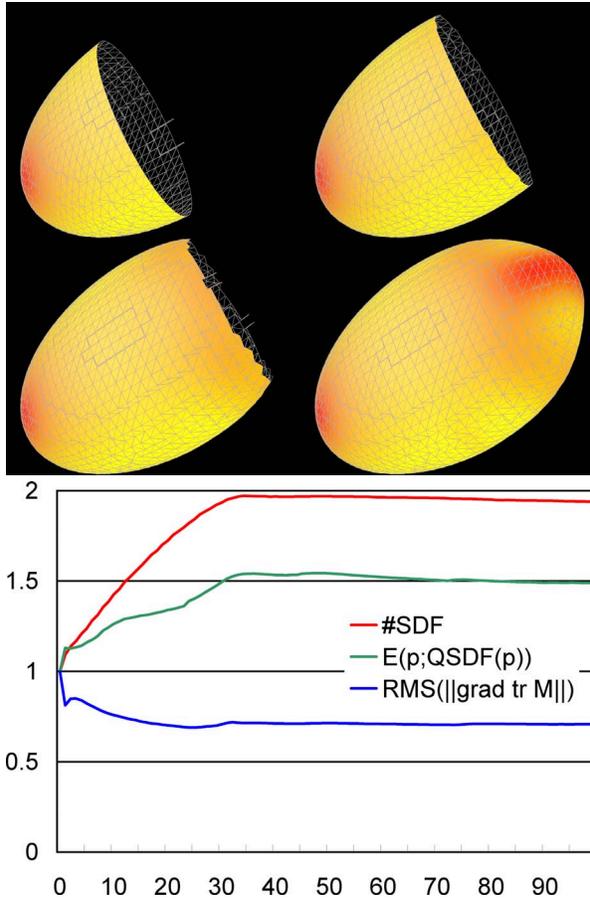


Figure 7: Result on a half ellipsoid as the initial state and the extended shapes after 10, 20 and 50 iterations. Plot of the number of samples, fitting errors and total curvature are plotted to the number of iterations. Each plot is normalized by the initial value.

the filled SDF is consistent to not only the signed distance but also the surface normal of the original SDF. This method can be implemented by multi-resolution strategy with multiple locality scales for fast processing on large holes. Robust statistics can also be introduced for outlier discrimination. Stable determination of convergence and propagation stabilization is currently on-going work. Some GUI should be necessary for real applications on complicated shapes.

References

[1] W. E. L. Grimson, "An implementation of a computational theory of visual surface interpolation," *Computer Vision, Graphics, and Image Processing*, vol. 22, pp. 39–69, 1983.

[2] Demetri Terzopoulos, "The computation of visible-surface representations," *IEEE Trans. PAMI*, vol. 10, no. 4, pp. 417–438, Jul. 1988.

[3] Richard Szeliski, "Fast surface interpolation using hierarchical basis functions," *IEEE Trans. PAMI*, vol. 12, no. 6, pp. 513–528, Jun. 1990.

[4] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Proc. SIGGRAPH92*, 1992, vol. 26, pp. 71–78.

[5] T. K. Dey and S. Goswami, "Tight cocone: A watertight surface reconstructor," *Journal of Computing and Information Science in Engineering*, vol. 3, pp. 302–307, 2003.

[6] Takeshi Masuda, "Registration and integration of multiple range images by matching signed distance fields for object shape modeling," *Computer Vision and Image Understanding*, vol. 87, pp. 51–65, 2003.

[7] A. Hilton, A. J. Stoddart, J. Illingworth, and T. Winderatt, "Reliable surface reconstruction from multiple range images," in *Proc. ECCV96*, 1996, pp. 117–126.

[8] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proc. SIGGRAPH96*, 1996, pp. 303–312.

[9] M. D. Wheeler, Y. Sato, and K. Ikeuchi, "Consensus surfaces for modeling 3D objects from multiple range images," in *Proc. ICCV97*, 1997, pp. 917–924.

[10] Toyofumi Saito and Jun-ichiro, "New algorithms for euclidean distance transformation of an n-dimensional digitized picture with applications," *Pattern Recognition*, vol. 27, no. 11, pp. 1551–1565, 1994.

[11] Y. Sakaguchi, H. Kato, K. Sato, and S. Inokuchi, "Acquisition of entire surface data based on fusion of range data," *Trans. IEICE*, vol. E-74, no. 10, pp. 3417–3422, 1991.

[12] Ross T. Whitaker, "A level-set approach to 3d reconstruction from range data," *International Journal of Computer Vision*, vol. 29, no. 3, pp. 203–231, Oct. 1998.

[13] James Davis, Stephen R. Marschner, Matt Garr, and Marc Levoy, "Filling holes in complex surfaces using volumetric diffusion," in *Proc. 3DPVT2002*, 2002, pp. 428–437.

[14] Ryusuke Sagawa and Katsushi Ikeuchi, "Taking consensus of signed distance field for complementing unobservable surface," in *Proc. 3DIM2003*, 2003, pp. 410–417.

[15] Takeshi Masuda, "Surface curvature estimation from the signed distance field," in *Proc. 3DIM2003*, 2003, pp. 361–368.

[16] Jan J. Koenderink, *Solid Shapes*, The MIT Press, 1990.