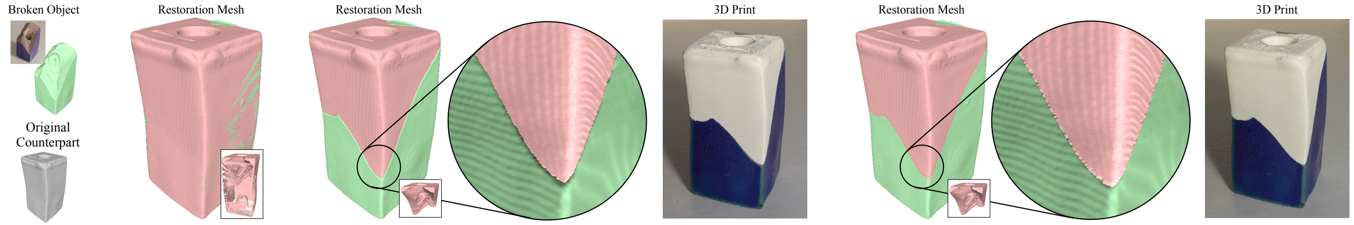
Automated Reconstruction of Smoothly Joining 3D Printed Restorations to Fix Broken Objects

Nikolas Lamb*
Lehigh University
nil518@lehigh.edu

Sean Banerjee
Clarkson University
sbanerje@clarkson.edu

Natasha Kholgade Banerjee
Clarkson University
nbanerje@clarkson.edu



(a) Input 3D Scans (b) Artifacts with Boolean Subtraction (c) Non-Smooth Join with Euclidean Distance Thresholding (d) Smooth Join with Our Approach

Figure 1: (a) Given a 3D scan of a broken object, our approach automatically generates a 3D printed restoration piece to repair the object by using the scan of an original counterpart as a reference. (b) Small-scale differences in geometric structure between the broken and original scans cause traditional volumetric Boolean subtraction methods to generate artifacts such as floating components and thin wide slivers that extend over the surface of the broken object, rendering the restoration piece unprintable. (c) Naïvely thresholding the Euclidean distance introduces a non-smooth join. (d) Our approach ensures smoothness prior to and across the join. Insets show alternate view of restoration meshes.

ABSTRACT

In this work, we provide an approach to automatically reconstruct a 3D printed restoration piece for a broken object from 3D scanned meshes of the broken object and an original counterpart. Our approach provides two contributions to reconstruct a restoration with a smooth join to the broken object, necessary for object functionality such as liquid containment, injury prevention, and visual aesthetic. As our first contribution, we leverage the original counterpart mesh to grow an exterior surface for the restoration piece that approaches the broken object within a small tolerance. As our second contribution, we project the exterior surface boundary onto the broken object to create a fracture surface boundary whose vertices satisfy the constraints of proximity, normal alignment, and tangency to vertices on the exterior surface boundary. Our approach prevents artifacts of volumetric Boolean subtraction, such as floating components and thin long slivers at the join, and avoids ruts at the join region introduced by Euclidean distance thresholding. We show 3D printed restoration results for 14 objects and 3D printable results for 8 objects.

CCS CONCEPTS

• **Computing methodologies** → *Mesh models*; *Shape analysis*.

KEYWORDS

restoration, repair, 3D printed, broken, fracture, join, 3D scan

ACM Reference Format:
Nikolas Lamb, Sean Banerjee, and Natasha Kholgade Banerjee. 2019. Automated Reconstruction of Smoothly Joining 3D Printed Restorations to Fix Broken Objects. In *SCF '19: Symposium on Computational Fabrication (SCF '19), June 16–18, 2019, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3328939.3329005

1 INTRODUCTION

When a household object breaks, most consumers choose to discard the object instead of repairing it, generating waste. 3D printing has become ubiquitous, providing the potential to print restoration pieces for broken objects. However, while 3D modeling tools have become common in the consumer space, manually creating the precise, highly irregular, geometry of a restoration piece at the region of fracture requires 3D modeling expertise, and is out of the scope of an average user. In this work, we provide an approach to automatically reconstruct a 3D printable restoration piece for a broken object from a 3D scanned mesh of the object. To generate a high-fidelity restoration, our approach leverages a 3D mesh of an original complete counterpart that is rigidly aligned to the broken 3D mesh. The 3D mesh of the original may be obtained by scanning objects in a set, e.g., plates in a dish set or salt and pepper shakers, or by using prior scans such as those of a person's teeth.

The principle challenge addressed by our work is that innate imprecisions in 3D scanning technologies and inaccuracies in rigid alignment algorithms introduce small-scale deviations between the

[*] Work was done while author was an undergraduate student at Clarkson University.

SCF '19, June 16–18, 2019, Pittsburgh, PA, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6795-0/19/06...$15.00
https://doi.org/10.1145/3328939.3329005

original and broken 3D meshes. Due these deviations, volumetric Boolean subtraction of the meshes to generate the restoration yields artifacts such as small floating components where the surfaces of the original and broken meshes are similar, and thin slivers that extend from the join of the restoration mesh and the broken mesh as shown in Figure 1(b). One approach to circumvent the artifacts is to merge an exterior surface and a fracture surface obtained by thresholding the Euclidean distance between the vertices of each 3D mesh and their nearest neighbors on the other mesh. However, as shown in Figure 1(c), this approach does not yield a restoration with a smooth join to the broken object, needed for functionality such as avoiding liquid discharge through ruts at the join, preventing injury from abrasion, or maintaining visual aesthetics. To reconstruct a restoration piece with a smooth join as shown in Figure 1(d), we provide two contributions.

1. Our first contribution is to grow an exterior surface for the restoration mesh as shown in Figure 2(c) by iteratively adding vertices from the original mesh until we reach the broken mesh within a tolerance. Our approach preserves mesh smoothness prior to the join by ensuring that the exterior surface vertices are as close to the broken mesh as possible, in contrast to an exterior surface generated by Euclidean distance thresholding where a tolerance needed to eliminate small artifacts may create a surface that is distant from the broken mesh.
2. Our second contribution is to generate a fracture surface boundary as shown in Figure 2(d) by projecting the exterior surface boundary onto the broken mesh such that the projected vertices satisfy the criteria of proximity, normal alignment, and tangency to the exterior surface. Our fracture surface generation approach ensures smoothness across the join between the restoration and broken object.

We show results of using our approach to create 3D printed restoration pieces for 14 objects and 3D printable pieces for 8 objects in Section 8. Our approach now allows the average consumer to reduce household waste and save on the cost of replacing or fixing broken objects. Users no longer need to have 3D modeling experience in order to generate a restoration part, and the broken object does not need to exhibit specific geometric qualities such as symmetry to be repaired. Using our work, kitchen-ware may be restored to be water-tight by providing a small amount of guidance to the user on applying adhesive. Users can repair tools, handrails, and handles such that the objects will stand up to prolonged handling without causing discomfort or injury. Our work enables the automation of repair in domain-specific applications such as restoration of objects of cultural and archaeological significance, generation of dental implants from scans of original and broken teeth, or creation of 3D printed bone replicas for reconstructive surgery.

2 RELATED WORK

Most prior approaches to repair objects using 3D printing focus on domain-specific applications such as generating medical implants, e.g., hips, knees, and dentures [Rengier et al. 2010], and restoration of artifacts of cultural value, e.g., busts [Scopigno et al. 2011], statues [Arbace et al. 2013], a historic porcelain bowl [Antlej et al. 2011], and a fractured silver crown [Seixas et al. 2018]. These approaches require users to manually generate the 3D printed restoration. Flores et al. [2017; 2016] provide a method to reconstruct a human ear using 3D printed reference models manually generated in Blender, and using auricular tracings and a 3D scan of the patient's head as guides in the modeling process. Lee [2013] provides a system to aid dentists in generating 3D printable dentures using a virtual sculpting tool and a 3D scan of a patient's mouth as a reference. While it is possible to include the input of users with domain expertise in fields such as cultural heritage, medicine, or dentistry, it is not feasible for average consumers to have the experience required to manually model the irregular fracture surface generated when everyday objects break.

Some approaches circumvent precisely replicating the fracture surface by altering the broken object. Harrysson et al. [2007] generate custom knee implants that attach to an existing bone structure which has been cut so that the interface is a flat plane, eliminating the need to precisely extract the fracture surface. Zheng et al. [2006] and Gao et al. [2008; 2005] provide a system to automate turbine repair by adding excess material to the damaged turbine and eroding the result until it conforms with a reference 3D model. These approaches do not enable average consumers to repair everyday broken objects made of ceramics, glass, or plastic, as they require specialized mechanical equipment to cut or erode the object, and professional expertise to alter the object and use the equipment. Additionally, the approaches of Zheng et al. and Gao et al. are subtractive, and not suited to the additive manufacturing process underlying 3D printing. Other approaches have used Boolean subtraction to generate restoration models when broken objects exhibit symmetry. Singare et al. [2008] generate restorative prosthetics for individuals with unilateral facial tumors by obtaining a 3D model of the patient's skull, virtually removing the tumor area, reflecting the 3D model over the bisecting plane of symmetry, and computing the Boolean subtraction of the altered half from the unaltered reflected half. They then manually refine the result to derive the restoration part. Fatini et al. [2008] use a similar approach in the restoration of a damaged medieval skull. The approaches of Singare et al. and Fatini et al. do not allow restoration parts to be generated for broken regions which do not have symmetric counterparts on the object, and require that only one side of the object be damaged.

More recently, in the approach of Gregor et al. [2014] and in the follow-up approach of Papaioannou et al. [2017], the authors provide a partially automated pipeline to generate restoration parts for a variety of broken cultural heritage objects with corresponding original objects generated using symmetric reflection [Sipiran et al. 2014]. Similar to the approaches of Sekou et al. and Fatini et al., the approaches of Gregor et al. and Papaioannou et al. fail to generate restoration pieces for broken objects without symmetric counterparts on the original object as we show with the approach of Papaioannou et al. in Section 8. Our approach does not require symmetries to be present on the object. Additionally, these approaches require users to manually flag the fracture surface, whereas our approach automatically extracts the fracture surface.

3 OVERVIEW

Let a 2-manifold mesh M be represented as $\mathcal{V}_M, \mathcal{F}_M$ such that $\mathcal{V}_M = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 ... \mathbf{v}_n\}$ where $\mathbf{v} \in \mathbb{R}^3$ is an ordered set of vertices

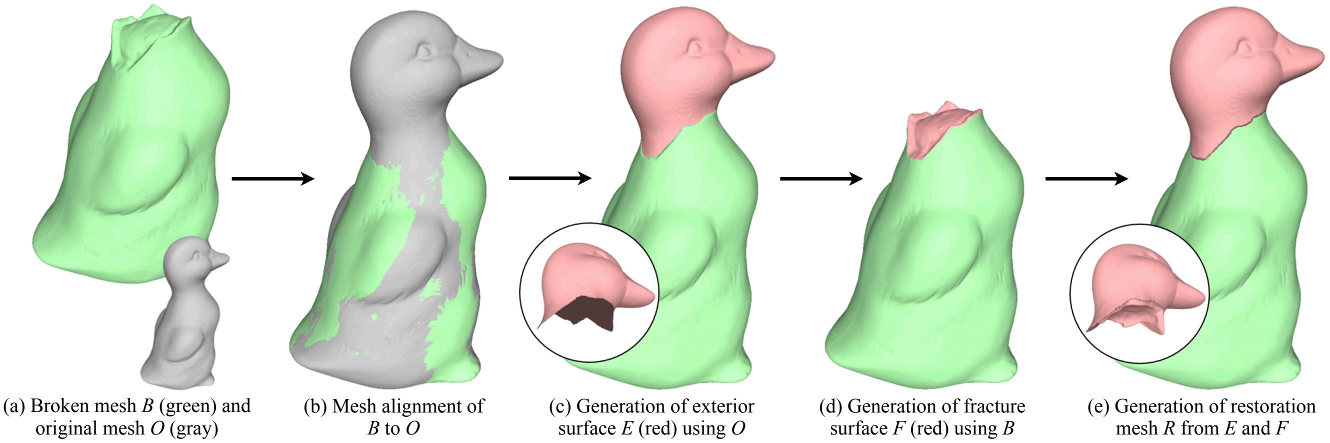

(a) Broken mesh B (green) and original mesh O (gray)
(b) Mesh alignment of B to O
(c) Generation of exterior surface E (red) using O
(d) Generation of fracture surface F (red) using B
(e) Generation of restoration mesh R from E and F

Figure 2: Overview of our approach. (a) Our approach takes as input a mesh B corresponding to the broken object, and a mesh O corresponding to the unbroken original object. (b) We align B to O, and (c) automatically generate the exterior surface E by extracting the vertex in O that is farthest removed from B and iteratively growing the mesh to extract portions of O not present in B. (d) We generate the fracture surface F by projecting the boundary of E onto B. (e) We obtain the restoration mesh R by triangulating the gap between E and F.

and $\mathcal{F}_M = \{(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3), \ldots\}$ is a set of 3-tuples such that each 3-tuple forms a triangle in counter-clockwise order. As shown in Figure 2(a), our approach takes as input two meshes (i) a scanned mesh $B = (\mathcal{V}_B, \mathcal{F}_B)$ corresponding to the broken object and (ii) a mesh $O = (\mathcal{V}_O, \mathcal{F}_O)$ corresponding to the unbroken original object. As shown in Figure 2(b) we clean these inputs so that they are closed 2-manifold meshes and perform a rigid alignment of B to O, as discussed in Section 4. To specify the outer structure of the restoration mesh R, we generate an exterior surface E as shown in Figure 2(c) using an iterative mesh growing approach that extracts regions of O present in R, as discussed in Section 5. To extract the fracture surface F shown in Figure 2(d), we project the boundary of the exterior surface E onto B to compute an ordered cycle on the surface of B, and we identify triangles inside the ordered cycle as being part of the fracture surface, as discussed in Section 6. We generate the restoration mesh R shown in Figure 2(e) by triangulating automatically generated correspondences between the boundaries of E and F, as discussed in Section 7.

The only manual steps in our approach pertain to intuitive tasks that can be performed by average users with minimal instruction. These steps correspond to (i) preparing the object for scanning by filling deep concavities with clay and applying talcum powder as discussed in Subsection 4.1, (ii) cleaning a few small artifacts generated by particulate matter such as dust, dirt, or talcum powder on the scanner surface as discussed in Subsection 4.2, and (iii) specifying the number of parts that have broken off.

4 MESH PRE-PROCESSING

4.1 Object Scanning

To scan a broken object, we use a Shining 3D EinScan-SP desktop laser scanner which scans objects from multiple angles using a turntable. For broken objects that have a hollow interior, we fill the interior with soft modeling clay, as shown in Figure 3(a), to avoid artifacts of 3D scanning highly non-convex regions. We ensure that the fracture region on the object is not altered by our additions. We dust objects that are reflective with talcum powder to ensure that object specularities do not impede scanning. We place the object on the scanner turntable and use the scanner software to reconstruct the object by capturing 20 depth images. We remove the clay and talcum powder from the object immediately after scanning. While the clay we use is non-toxic and can be cleaned when used on everyday objects, it may not be desirable for objects of cultural significance. In this case, the user may resort to one of the various filler materials used in the conservation industry—such as plaster of Paris [Williams 2002]—which can be readily removed either mechanically or by submerging in hot water or a chemical bath.

4.2 Mesh Cleaning

The meshes B and O may contain self-intersections, small artifacts, and mesh holes as shown in Figure 3(b). We split off self-intersections in Autodesk Netfabb using an approach similar to Campen and Kobbelt [2010], and delete all but the largest connected component in both meshes. We visually inspect the meshes for small attached artifacts generated by the presence of particulate matter such as dust, dirt, or talcum powder on the turntable, and remove these artifacts manually by selecting them using a lasso tool in Netfabb and deleting them. We close any holes in the mesh using the approach of Hu et al. [2012] implemented in MeshLab [Cignoni et al. 2008], which ensures that the meshes are closed, 2-manifold surfaces, as shown in Figure 3(c).

4.3 Mesh Alignment

After cleaning the input meshes B and O, we subsample both meshes to 3000 vertices and obtain a rigid transformation which globally aligns B to O using the Super 4PCS algorithm [Mellado et al. 2014]. We fine-tune the alignment by using ICP [Besl and McKay 1992] on the dense vertex sets transformed by the global alignment. To prevent vertices on the two meshes that are distant from each other from influencing the alignment, we re-run ICP to rigidly re-align B to O using only vertices in each mesh that have nearest neighbors in the other mesh closer than a threshold γ, where $\gamma = 3.0$ millimeters. From here on, while $\mathcal{V}_O$ refers to the vertex set of the original mesh without any transformation, $\mathcal{V}_B$ refers to the vertex set of the broken mesh aligned using Super 4PCS and two ICP steps, as shown in Figure 3(d).

5 EXTERIOR SURFACE GENERATION

To obtain the exterior surface $E = (\mathcal{V}_E, \mathcal{F}_E)$, we initialize $\mathcal{V}_E = \{\mathbf{v}\}$ where $\mathbf{v} \in \mathcal{V}_O$ is the farthest vertex from any vertex in $\mathcal{V}_B$. We then use an iterative approach to expand E, as shown in Figure 4(a). In each iteration we compute the current mesh boundary ∂E by obtaining an ordered list of all face edges in E that occur only once. We then extend the mesh boundary by adding to $\mathcal{V}_E$ a set of vertices $\mathcal{V}_{\overline{\mathbf{v}}}$ for each vertex $\overline{\mathbf{v}} \in \partial E$, where $\mathcal{V}_{\overline{\mathbf{v}}} = \mathcal{V}_d \cap \mathcal{N}_O(\overline{\mathbf{v}}) \cap (\mathcal{V}_O \setminus \mathcal{V}_E)$, $\mathcal{V}_d \subset \mathcal{V}_O$ is a set of vertices that are farther than μ millimeters away from any vertex in $\mathcal{V}_B$, and $\mathcal{N}_O(\overline{\mathbf{v}})$ is the 1-ring neighborhood of $\overline{\mathbf{v}}$. In our work we use $\mu = 0.35$ millimeters. We stop adding vertices to $\mathcal{V}_E$ when we reach an iteration where for all vertices $\overline{\mathbf{v}} \in \partial E$, $\mathcal{V}_{\overline{\mathbf{v}}} = \emptyset$. We then set $\mathcal{F}_E \subset \mathcal{F}_O$ such that $\mathcal{F}_E$ includes all faces in $\mathcal{F}_O$ for which every vertex referenced in the face is in $\mathcal{V}_E$. For objects that require multiple restoration parts, i.e., multiple separate connected components, we set $\mathcal{V}_O = \mathcal{V}_O \setminus \mathcal{V}_E$ and generate additional exterior surfaces repeatedly until we have extracted the desired number of connected components, specified as input by the user. We then combine all exterior surfaces to form the final exterior surface E.

If the mesh is not sufficiently dense, E may penetrate B. To avoid penetration, prior to computing E we first estimate the density of the mesh by determining if the length of any edge in O or B is greater than or equal to η, where

$$\eta = \frac{2\mu}{\sqrt{7/3}}, \tag{1}$$

and if so we recursively subsample the faces adjacent to the edge using the approach of Jacobson [2018] until the length of all edges in O and B are shorter than η. We derive η below.

Computation of η. Consider an equilateral triangle $\Delta\mathbf{ABC}$ shown in Figure 4(b), with each side having the same length as the longest edge in the mesh B, and a line segment $\mathbf{PQ}$ representing an edge in the mesh O. Assume that $\mathbf{P}$ and $\mathbf{Q}$ satisfy the condition that every point on the exterior surface is at least μ units away from any point on B. If $\Delta\mathbf{ABC}$ is a triangle of the mesh B, then $\mathbf{P}$ and $\mathbf{Q}$ are each at least μ units from $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$.

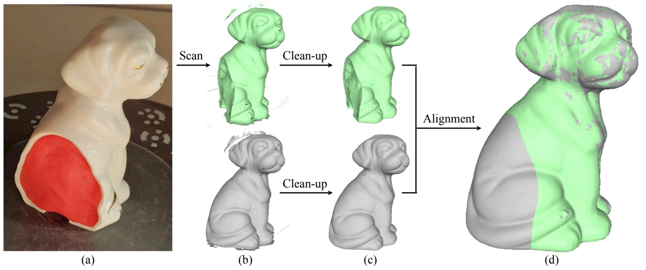


Figure 3: Mesh pre-processing. (a) We fill the interior of the broken object with modeling clay before scanning. (b) Because the broken scan (green) and original scan (gray) can exhibit mesh holes and scanning artifacts, (c) we split off self-intersections and fill holes to generate 2-manifold meshes. (d) We use Super 4PCS [Mellado et al. 2014] and ICP [Besl and McKay 1992] to rigidly align *B* to *O*.

In particular, consider the case where $\mathbf{P}$ and $\mathbf{Q}$ are exactly μ units away from $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, and the length a of a side of $\Delta\mathbf{ABC}$ is such that it satisfies the triangle inequality over $\Delta\mathbf{PBI}$, i.e., $||\mathbf{PI}|| + ||\mathbf{BI}|| \geq ||\mathbf{PB}||$ in Figure 4(b). If $\mathbf{PQ}$ intersects $\Delta\mathbf{ABC}$, then the line segment $\mathbf{PQ}$ is perpendicular to $\Delta\mathbf{ABC}$, passes through the centroid of $\Delta\mathbf{ABC}$, and is bisected by $\Delta\mathbf{ABC}$. The length of $\mathbf{PQ}$, i.e. η, is given as

$$\eta = 2||\mathbf{PI}|| = 2\sqrt{||\mathbf{PB}||^2 - ||\mathbf{BI}||^2}, \tag{2}$$

since $\Delta\mathbf{PBI}$ is a right triangle. As $\mathbf{BI}$ is the hypotenuse of $\Delta\mathbf{IBN}$, and $\angle\mathbf{IBN}$ is $\pi/6$, the length of $\mathbf{BI}$ can be found as

$$||\mathbf{BI}|| = \frac{||\mathbf{BN}||}{\cos(\pi/6)} = \frac{a/2}{\sqrt{3}/2} = \frac{a}{\sqrt{3}}. \tag{3}$$

Equation (2) can now be written as

$$\eta = 2\sqrt{\mu^2 - \frac{a^2}{3}}. \tag{4}$$

In Theorem A.1 in the appendix, we show that $\Delta\mathbf{ABC}$ being equilateral with side length a equal to the longest edge in B provides the smallest length for $\mathbf{PQ}$ in comparison to any triangle in B. In Theorem A.2, we show that $\mathbf{PQ}$ is the smallest possible line segment from O that intersects $\Delta\mathbf{ABC}$. If it can be ensured that line segments in O are smaller than $\mathbf{PQ}$, then edges in E grown from O will not penetrate triangles in B. By performing the above analysis for a line segment $\mathbf{PQ}$ in B and a triangle $\mathbf{ABC}$ in O, it can be similarly ensured that if line segments in B are smaller than $\mathbf{PQ}$, then triangles in E grown from O will not be penetrated by line segments in B.

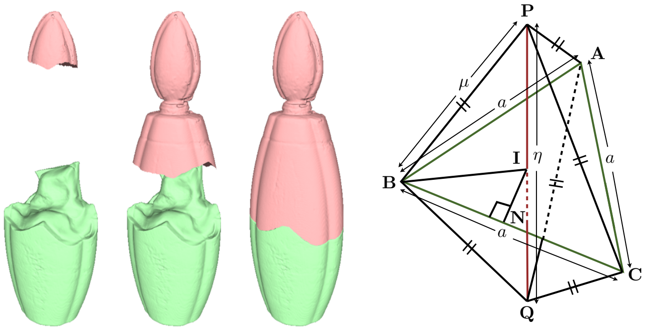
If the length of a does not satisfy the triangle inequality for $\Delta\mathbf{PBI}$, i.e., if $||\mathbf{PI}|| + ||\mathbf{BI}|| < ||\mathbf{PB}||$, then the above analysis fails and line segments smaller than $\mathbf{PQ}$ in O will intersect triangles in B. This sets an upper bound $a_{\max}$ on the longest edge a of B, such that at the upper bound, $||\mathbf{PI}|| + ||\mathbf{BI}|| = ||\mathbf{PB}||$, and $||\mathbf{BI}|| = \frac{a_{\max}}{\sqrt{3}}$. Given that $||\mathbf{PI}|| = \frac{\eta}{2}$ and $||\mathbf{PB}|| = \mu$, $a_{\max}$ can be obtained as

$$a_{\max} = \sqrt{3}\,(||\mathbf{PB}|| - ||\mathbf{PI}||) = \sqrt{3}\left(\mu - \frac{\eta}{2}\right). \tag{5}$$

The same upper bound exists on the longest edge of O. However, if $a = a_{\max}$, the line segment $\mathbf{PQ}$ collapses to a point, i.e., $||\mathbf{PQ}|| = 0$, and $||\mathbf{PI}|| = 0$, or $a_{\max}$ is given as

$$a_{\max} = \sqrt{3}\mu. \tag{6}$$

Thus in practice, a should be smaller than $a_{\max}$. In this work, we subsample triangles in the meshes B and O to have a small maximum edge length a. To ensure that the relative sizes of triangles in B and O are balanced with respect to each other, we set a to be the same



(a) Round 100, round 300, and final round 459 of exterior mesh growth

(b) Computation of η

Figure 4: (a) Starting with the vertex in $\mathcal{V}_O$ that is farthest from any vertex in $\mathcal{V}_B$, we grow the exterior mesh by adding vertices from $\mathcal{V}_O$ to the set $\mathcal{V}_E$ if they are in the one-ring of any vertex in ∂E, do not already belong to $\mathcal{V}_E$, and are distanced at least μ from any vertex in $\mathcal{V}_B$. We iteratively add vertices until E reaches B as shown in the last image. (b) In Equation (1), we determine the maximum value η that the longest edge PQ in O can take in order to avoid penetrating the triangle ΔABC as shown by treating ΔABC as equilateral.

as η, the length of **PQ**. Setting $a = \eta$ in Equation (4) and solving for η we obtain the expression in Equation (1). We subsample any triangle in B or O that has an edge larger than η, until the resulting triangles have an edge length lesser than η. Note that setting $a = \eta$ ensures that $a < a_{\max}$.

6 FRACTURE SURFACE GENERATION

In this section we identify the fracture surface of the restoration mesh $F = (\mathcal{V}_F, \mathcal{F}_F)$, shown in Figure 2(d). To ensure an accurate fracture surface, we project ∂E onto B by minimizing an objective function discussed in Section 6.1. Since ∂E projected onto B may not contain a complete list of connected vertices on B, we determine a complete edge boundary by identifying a cycle on B, and use the handedness of B to identify and extract F as triangles on the left of the projected boundary, as discussed in Section 6.2.

6.1 Boundary Projection

To bridge the gap between meshes E and B, we project ∂E onto B such that the restoration mesh exhibits a strong cusp where it interfaces with B, as shown in Figure 5(d), creating a smooth connection between the two meshes. For each vertex $\mathbf{v} \in \partial E$ we choose a vertex $\overline{\mathbf{v}} \in \mathcal{V}_B$ from the k nearest neighbors in $\mathcal{V}_B$ to $\mathbf{v}$ such that $\overline{\mathbf{v}}$ minimizes the energy function

$$f(\overline{\mathbf{v}}) = \lambda_1 f_1(\overline{\mathbf{v}}) + \lambda_2 f_2(\overline{\mathbf{v}}) + \lambda_3 f_3(\overline{\mathbf{v}}). \tag{7}$$

As shown in Figure 5(b), the term $f_1(\overline{\mathbf{v}})$ in Equation (7) ensures that $\overline{\mathbf{v}}$ is as close as possible to $\mathbf{v}$ and is given as

$$f_1(\overline{\mathbf{v}}) = \|\overline{\mathbf{v}} - \mathbf{v}\|. \tag{8}$$

As shown in Figure 5(c), the term $f_2(\overline{\mathbf{v}})$ in Equation (7) ensures that the normals $\mathbf{n_v}$ and $\overline{\mathbf{n_v}}$ are aligned, and is given as

$$f_2(\overline{\mathbf{v}}) = -\min(\mu, 1)(\mathbf{n_v} \cdot \overline{\mathbf{n_v}}). \tag{9}$$

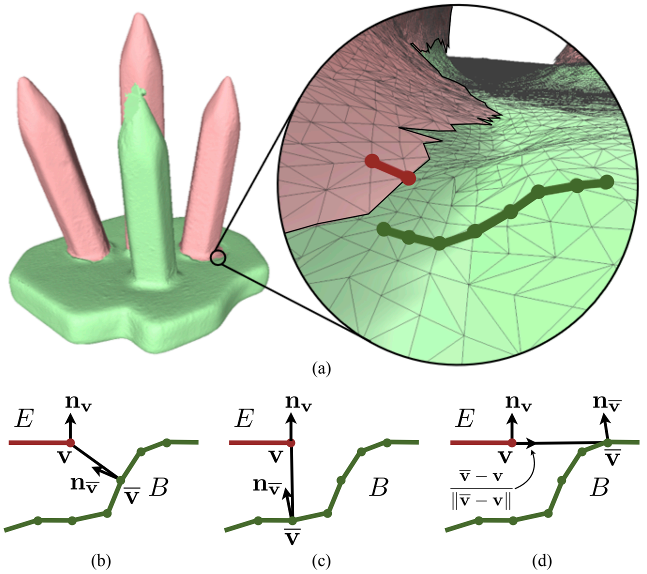

(a)
E
$\mathbf{n_v}$
$\mathbf{v}$
$\mathbf{n}_{\overline{\mathbf{v}}}$
$\overline{\mathbf{v}}$
B
(b)
E
$\mathbf{n_v}$
$\mathbf{v}$
$\mathbf{n}_{\overline{\mathbf{v}}}$
$\overline{\mathbf{v}}$
B
(c)
E
$\mathbf{n_v}$
$\mathbf{n}_{\overline{\mathbf{v}}}$
$\mathbf{v}$
$\overline{\mathbf{v}}$
$\frac{\overline{\mathbf{v}} - \mathbf{v}}{\|\overline{\mathbf{v}} - \mathbf{v}\|}$
B
(d)

Figure 5: Effect of minimizing select terms in Equation (7) on boundary projection. (a) If we choose a $\overline{\mathbf{v}}$ that minimizes only $f_1(\overline{\mathbf{v}})$, v connects to the closest vertex in B, resulting in a furrowed join. (b) If we choose a $\overline{\mathbf{v}}$ that minimizes $\lambda_1 f_1(\overline{\mathbf{v}}) + \lambda_2 f_2(\overline{\mathbf{v}})$, v connects to the closest vertex with a similar normal, also resulting in a furrowed join. (c) Using our approach, we choose a $\overline{\mathbf{v}}$ by minimizing $\lambda_1 f_1(\overline{\mathbf{v}}) + \lambda_2 f_2(\overline{\mathbf{v}}) + \lambda_3 f_3(\overline{\mathbf{v}})$ which selects a vertex with a similar normal such that a vector drawn from v to $\overline{\mathbf{v}}$ is tangent to the normal at v, resulting in a smooth join between meshes.

As shown in Figure 5(d), the term $f_3(\overline{\mathbf{v}})$ in Equation (7) ensures that a vector drawn from $\mathbf{v}$ to $\overline{\mathbf{v}}$ is approximately tangent to $\mathbf{n_v}$, and is given as

$$f_3(\overline{\mathbf{v}}) = \min(\mu, 1)\left(\mathbf{n_v} \cdot \frac{\overline{\mathbf{v}} - \mathbf{v}}{\|\overline{\mathbf{v}} - \mathbf{v}\|}\right). \tag{10}$$

We collect these vertices into an ordered list of projected vertices $\mathcal{V}_p$. In our work we use $k = 100$, $\lambda_1 = 1.0$, $\lambda_2 = 4.0$, and $\lambda_3 = 4.0$. Figure 6(a) shows the vertices selected on B for a portion of ∂E.

To generate a cycle with no repetitions from the disconnected vertices in $\mathcal{V}_p$, we iterate over successive vertex pairs in $\mathcal{V}_p$ and use Dijkstra's algorithm [1959] to generate a path between each vertex pair. We insert the path between the current vertices in $\mathcal{V}_p$, as shown in Figure 6(b). We link successive vertices until the last vertex in $\mathcal{V}_p$ is linked to the first, completing the cycle. We then check the cycle for repeated vertices, which may generate self-intersecting geometry. For all vertices $\mathbf{v} \in \mathcal{V}_p$, we check if $\mathbf{v}$ occurs in the previous n vertices, where $n = \min\left(\rho, \left\lfloor \frac{|\mathcal{V}_p|}{2} \right\rfloor\right)$, and $|\mathcal{V}_p|$ refers to the count of $\mathcal{V}_p$. If we find an occurrence, we delete all the vertices in $\mathcal{V}_p$ that lie between the first and current occurrence of $\mathbf{v}$. In our work we use $\rho = 50$ to prevent collapse of the entire cycle. The dark green edges in Figure 6(c) represent the resulting fracture surface boundary extracted after eliminating cycle repetitions.

6.2 Extracting Fracture Mesh

The cycle $\mathcal{V}_p$ divides B into several connected components, some of which correspond to the exterior surface of B and some of which correspond to the fracture surface F. Because the order of the projected cycle $\mathcal{V}_p$ coincides with the order of ∂E, vertices in the fracture surface lie on the left side of the projected cycle. To generate F, we extract the boundary of F from B by exploiting the right-handed referencing of faces in B. We set $\mathcal{F}_F$ to all faces $\mathbf{f} \in \mathcal{F}_B$ such that if there exist any two successive vertices referenced in $\mathbf{f}$ which are equal to any two successive vertices in $\mathcal{V}_p$, $\mathbf{f}$ is added to $\mathcal{F}_F$. We then compute the interior of the fracture surface by removing the projected edge vertices from B, so that $\mathcal{V}_B = \mathcal{V}_B / \mathcal{V}_p$. We compute the connected components in B and for each component $C = (\mathcal{V}_C, \mathcal{F}_C)$, we add C to F if $\mathcal{V}_F \cap \mathcal{V}_C \neq \emptyset$. To ensure correct mesh orientation, we reverse the order of vertex references in $\mathcal{F}_F$. The opaque green faces in Figure 6(c) represent the extracted fracture surface.

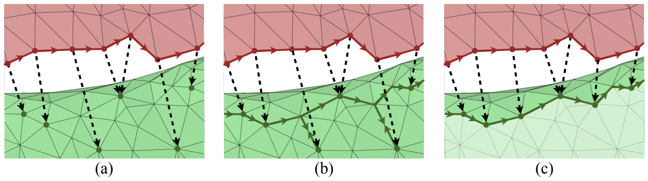


Figure 6: Extracting fracture mesh. (a) For each vertex in ∂E, shown in red, we select a corresponding vertex on B using the energy function described in Equation (7). The generated vertices on B, shown in green and pointed to with a dotted arrow, are aggregated into $\mathcal{V}_p$, and (b) connected using Dijkstra's algorithm. (c) To eliminate self-intersecting geometry, we search for and eliminate repetitions. We extract the fracture surface shown by the opaque green faces by exploiting the right-hand referencing of faces in B.

7 FORMING RESTORATION MESH

To generate the restoration mesh $R = (\mathcal{V}_R, \mathcal{F}_R)$, we combine the exterior mesh E generated in Section 5 and the fractured mesh F generated in Section 6, such that $\mathcal{V}_R = \mathcal{V}_E \cup \mathcal{V}_F$ and $\mathcal{F}_R = \mathcal{F}_E \cup \mathcal{F}_F$. To determine correspondences for triangulation between E and F, for each vertex in ∂E we find the nearest neighbor in $\mathcal{V}_p$, and store these correspondences in a new ordered list $\mathcal{V}_c$. To ensure that faces generated during triangulation do not overlap, we split $\mathcal{V}_c$ into two partial lists such that the first list spans from the first through the p^{th} vertex where $p = \left\lfloor \frac{|\mathcal{V}_c|}{2} \right\rfloor$ and the second list spans from the $(p+1)^{\text{th}}$ vertex through the last vertex. For each partial list we iterate over the vertices, and for each vertex we check if that vertex has occurred previously in the partial set. If it has, we overwrite all the vertices between the first occurrence and the current vertex with the current vertex. We then recombine the two partial lists and repeat this overlap elimination on the vertices between and including the p_1^{th} and p_2^{th} vertices, where $p_1 = \left\lfloor \frac{p}{2} \right\rfloor$ and $p_2 = \left\lfloor \frac{3p}{2} \right\rfloor$. To triangulate the gap between E and F in R, we iterate over the vertices in $\mathcal{V}_c$ using an iterator i and the vertices in $\mathcal{V}_p$ using an iterator j. For a given iteration with $\mathbf{v} \in \partial E$ and $\overline{\mathbf{v}} \in \mathcal{V}_p$, if $\mathcal{V}_c(i+1) = \mathcal{V}_p(j)$ then we create a new face $(\mathbf{v_{i+1}}, \mathbf{v_i}, \overline{\mathbf{v_j}})$, append this face to $\mathcal{F}_R$, and increment i, else we create a new face $(\overline{\mathbf{v_j}}, \overline{\mathbf{v_{j+1}}}, \mathbf{v_i})$, append this face to $\mathcal{F}_R$, and increment j. We continue to generate faces until i is $|\mathcal{V}_c| - 1$ and j is $|\mathcal{V}_p| - 1$. Self-intersections may occur during the triangulation between E and F if the energy function described in Equation (7) selects vertices on B that are behind the cusp of the fracture. Since the geometry beyond the intersection of R and B does not impact the smoothness of the join, we use Autodesk Netfabb's version of the approach of Campen and Kobbelt [2010] to remove self-intersections.

8 RESULTS

We generate restoration parts for 22 objects discussed in Tables 1 and 2, and shown in Figures 7 to 10. Our chosen objects represent functional items, such as the cups and candle holder, and ornamental items, such as the Mario and Winnie the Pooh figurines. We select objects that are either sold as sets, mass produced and sold as individual items, or hand crafted and sold in pairs. The candle holder, flower bowl, and flower were cast objects for which we obtained and scanned exact replicas. For the watering can shaped salt shaker, barn, and milk jug, we purchased a set of two similar objects, broke one of the two, and used the unbroken object to generate the original mesh. The owl cup and candle stand were hand-made in sets of two, which we similarly exploited to generate repairs. In this case, since the items were handmade, it is likely that minor variations exist between the original and broken object. For some objects, like the owl cup in Figure 7, our scanner was unable to capture the inside of the object, and provided an open mesh. We obtained a closed mesh for these objects by using hole filling in MeshLab to close the top with a plane surface.

We remove any biases in the generation of our broken objects by striking each object at a random location. After scanning, the original meshes vary in size from 0.2 million vertices to 1.2 million vertices, and the broken meshes range from 0.1 million to 1.1 million vertices. We implemented our algorithm in MATLAB R2018b and ran our experiments on a desktop computer with an Intel i7-7700k processor and 32 GB of RAM, running Microsoft Windows 10. We provide timing results for our algorithm devoid of the mesh preprocessing steps for all objects in Table 2. Restoration pieces were printed using PLA plastic on one of three 3D printers— a FlashForge Creator Pro, a Qidi Tech, and an Ultimaker 3. Each restoration part took between 2 and 4 hours to print.

Figures 7 and 8 show 3D printed results for 14 objects, and Figure 9 shows 3D printable results for 5 objects generated using our approach. Our restoration pieces allow a user to ergonomically use a cup even after it has been broken, as shown with the owl cup in Figure 7, or display a set of candles even after one or more candle holders have been broken, as shown with the candle stand in Figure 8. The tight join of our restoration part enables it to support parts of the broken object even without adhesive, as shown in Figure 7 with the leg of the big elephant and in Figure 8 with the candle holder. In addition, our approach can repair intricate items, like the Mario figurine in Figure 7 and the Winnie the Pooh figurine in Figure 8, such that they are still easily recognizable. As shown by the timing results in Table 2, the restoration piece can be computed

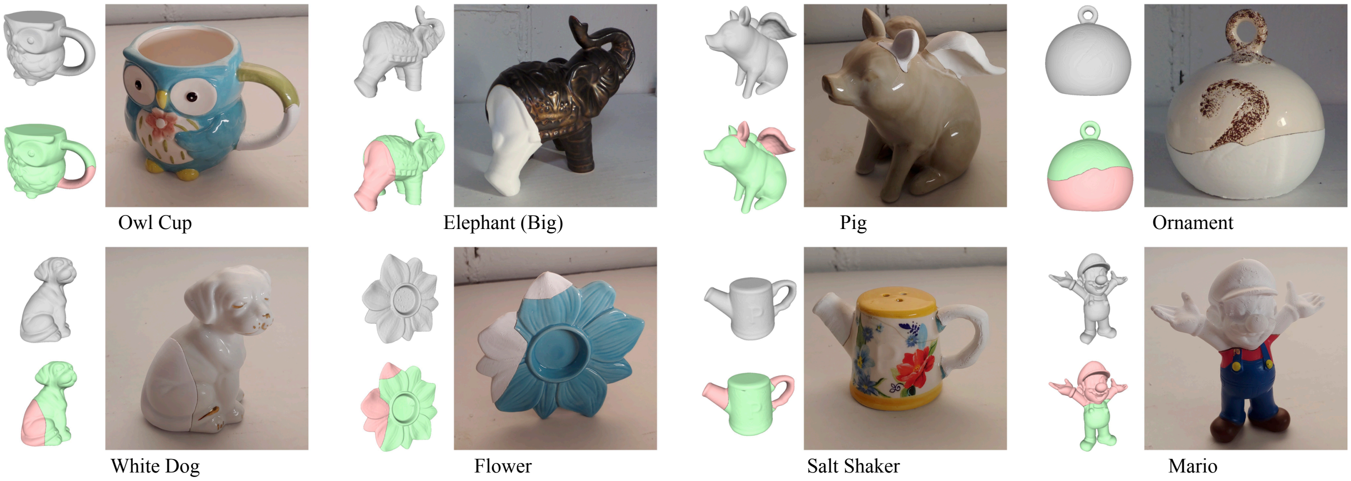


Figure 7: Original scanned mesh (gray), broken mesh (green) with restoration mesh (red), and 3D printed restoration piece joined to the broken piece for an owl cup, a big elephant, a pig, an ornament, a white dog, a flower, a watering can-shaped salt shaker, and a Mario figurine.

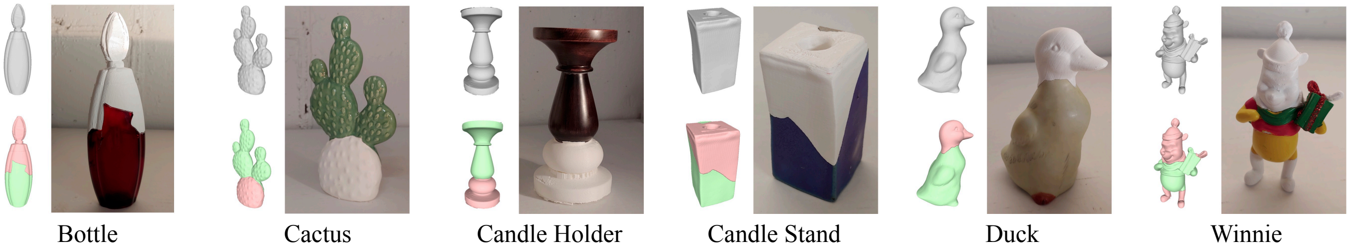


Figure 8: Original scanned mesh (gray), broken mesh (green) with restoration mesh (red), and 3D printed restoration piece joined to the broken piece for a bottle, a cactus, a candle holder, a candle stand, a duck, and a Winnie the Pooh figurine.

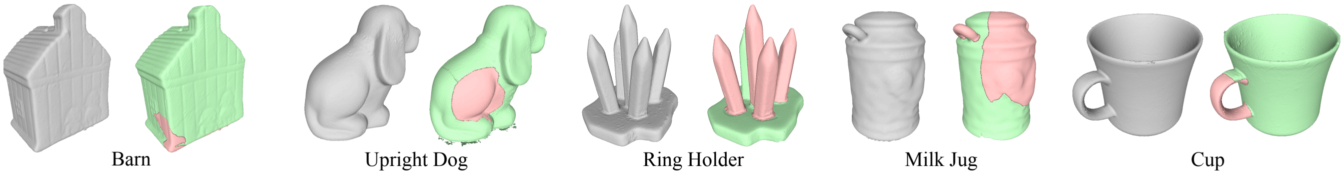


Figure 9: Original (gray), broken (green) and restoration (red) meshes for a variety of objects.

quickly, so that the main bottleneck in performing the repair is in scanning the object and printing the restoration.

For most of our objects, scanning and alignment imperfections fit within the tolerance of $\mu = 0.35$ millimeters. However, the bowl and frog cup on the top of Figure 10(a) show large alignment imperfections in the indicated regions. For the bowl, the original counterpart had to be supported by clay for proper scanning, while for the frog cup, the scanner does not accurately model the deep concavities on the inside. The elephant at the bottom of Figure 10(a) has large scanning imprecisions, since a number of fractured pieces were joined together when the elephant broke, causing higher deviations in the geometry from the original structure. As a result, choosing $\mu = 0.35$ millimeters leads to slivers that extend onto the broken object. While still 3D printable, the restoration piece would not have a tight fit with the broken object as shown in Figure 10(b). In this case, the user has the option of increasing the tolerance of μ. Figure 10(c) shows results for the bowl, the frog cup, and the elephant with a tolerance of $\mu = 1.5$ millimeters, in which case the restoration mesh is more plausible.

8.1 Evaluation

To determine the accuracy of each tested approach, we provide quantitative results in Table 1, where we report the error as the average Euclidean distance between vertices in R for various approaches to the nearest neighbors in $O \cup B$. To provide an intuitive comparison of error, we eliminate faces in R that come from O or B, providing quantitative results only for new geometry in R generated by the approach being tested.

8.1.1 Comparison to naïve Boolean subtraction. As shown in Figure 11(a), we compare our approach to volumetric Boolean subtraction of B from O performed in Autodesk Maya. This approach creates long thin mesh slivers where the surfaces of the original and broken meshes are similar, as shown in Figure 11(a), which would prevent these pieces from being physically attached to the broken object if the object were printed. However, this approach also generates an excessive number of connected components, as shown in Table 2, which makes the 3D prints functionally unusable.

8.1.2 Comparison to altered Boolean subtraction. As shown in Figure 11(d), we compare our approach to an altered volumetric Boolean subtraction of B from O performed in Autodesk Netfabb. In this approach, we move every vertex in O with a nearest neighbor in B that is closer than μ to its corresponding nearest neighbor in B, where $\mu = 0.35$ millimeters. While this approach eliminates most surface artifacts and slivers produced by naïve Boolean subtraction, some artifacts still remain as shown in Figure 11(d), and the restoration join is not smooth.

8.1.3 Comparison to naïve Euclidean distance thresholding. As shown in Figure 11(c), we compare our approach to a naïve Euclidean distance thresholding. In this approach, we keep vertices in O if their nearest neighbors in B are farther away than μ and vice versa for vertices in B. As in prior cases, we use a value of $\mu = 0.35$ millimeters. For the result in Figure 1, we manually discarded artifacts that arise due to the use of a small threshold, and triangulated between the surface boundaries using the approach described in Section 7. However this method is infeasible for the average consumer as it requires mesh editing expertise. To automatically remove artifacts, we increase the distance for which vertices are discarded in both meshes until only the desired number of connected components remain. This approach eliminates surface artifacts, but fails to generate a restoration with a smooth join, as shown in Figure 11(c).

8.1.4 Comparison to Papaioannou et al. We compare our approach to the work of Papaioannou et al. [2017] in Figure 11(b). We implemented their approach in MATLAB R2018b, and we set their vertex exclusion parameter for the exterior surface σ to 0.35 millimeters, and for the fracture surface ρ to 3.0 millimeters. Their approach is specifically suited to broken objects that have been completed using symmetries so that the complete object has the same geometry in the available region as the broken object. In our case, where the geometry of the original counterpart shows small-scale differences from the broken objects, their approach retains large regions of B that do not belong to the fractured surface, often preventing the mesh from being 3D printed. It does not generate a smooth join between the exterior and fracture surface and generates holes on the exterior region as shown in Figure 11(b).

8.1.5 Variations of our approach. To demonstrate the effect of the individual terms in the energy function described in Equation (7), we generate results using only the first term in Equation (7) as shown in Figure 11(e), using the first and second terms in in Equation (7) as shown in Figure 11(f), and using our approach with all terms in Equation (7) as shown in Figure 11(g). As shown in Figure 11(e), boundary projection using the first term generates a similar result to that of altered Boolean subtraction at the join region. It can be seen in Figure 11(f) that addition of the second term improves the smoothness of the join with objects for which the edges of the fracture region fall directly below the exterior region and mimic the orientation of the fractured region. Finally, our approach with the full optimization function described in Equation (7) demonstrates the smoothest join for all objects tested, as shown in Figure 11(g).

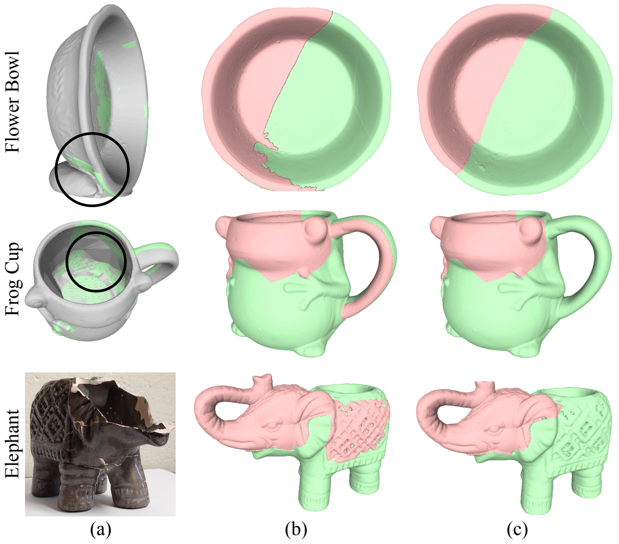


Figure 10: (a) The bowl and frog cup show alignment imperfections due to the use of clay to support the bowl and inaccurate scanning of the interior for the frog cup, while joining multiple fractured pieces creates scanning imprecisions for the elephant. (b) Using a small value of μ at 0.35 creates slivers that extend over the broken region. (c) By changing the value of μ to 1.5, we obtain a more plausible result.

Table 1 shows that adding each term in Equation (7) progressively reduces the error, with the lowest error obtained using three terms. In the table, the terms Naïve Bool, Altered Bool, and Naïve Thresh refer to naïve Boolean subtraction, altered Boolean subtraction, and naïve Euclidean distance thresholding respectively. Our results using f_1 alone, f_1 and f_2, and all three terms in Equation (7), show lower errors compared to altered Boolean subtraction, naïve Euclidean distance thresholding, and the approach of Papaioannou et al. [2017]. The error generated by naïve Boolean subtraction is low relative to our work, as volumetric Boolean subtraction only generates geometry that is co-planar with faces in $O \cup B$. However, as discussed in Subsection 8.1.1, the excessive number of connected components and slivers produced by the method as shown in Table 2 makes it unusable in obtaining a 3D printable restoration.

8.2 Statistical Testing

We perform a one-sided paired Wilcoxon signed-rank test to determine if the distances generated by our approached are significantly

Table 1: Quantitative comparison of our approach to naïve Boolean subtraction, altered Boolean subtraction, naïve Euclidean distance thresholding, and the work of Papaioannou et al. [2017]. As the error, we report the average distance between vertices for new geometry generated in the restoration piece.

	Barn	Bottle	Cactus	Candle Stand	Candle Holder	Cup	Duck	Elephant	Elephant (Big)	Flower	Flower Bowl
Naïve Bool	0.0226	0.0271	0.0264	0.0026	0.0293	0.0198	0.0219	0.0020	0.0263	0.0229	0.0018
Altered Bool	0.0578	0.0596	0.0578	0.0692	0.0651	0.0582	0.0526	0.0652	0.0597	0.0525	0.0575
Naïve Thresh	0.2895	0.1851	0.0838	0.2239	0.5410	0.0946	0.1328	0.1540	0.2996	0.1591	0.3537
Papaioannou et al. [2017]	0.0365	0.9042	1.3417	2.9806	0.0549	0.0374	0.5258	0.5049	0.1740	0.0338	0.1481
Ours f_1 only	0.0475	0.0442	0.0427	0.0454	0.0495	0.0453	0.0463	0.0518	0.0466	0.0471	0.0478
Ours f_1, f_2	0.0401	0.0380	0.0331	0.0461	0.0354	0.0452	0.0308	0.0531	0.0249	0.0282	0.0436
Ours f_1, f_2, f_3	0.0279	0.0254	0.0228	0.0323	0.0240	0.0282	0.0193	0.0377	0.0193	0.0229	0.0338
	Frog Cup	Mario	Milk Jug	Ornament	Owl Cup	Pig	Ring Holder	Salt Shaker	Upright Dog	White Dog	Winnie
Naïve Bool	0.0272	0.0215	0.0224	0.0218	0.0043	0.0261	0.0189	0.0107	0.0093	0.0221	0.0217
Altered Bool	0.0651	0.0480	0.0536	0.0616	0.0786	0.0604	0.0477	0.0605	0.0519	0.0518	0.0543
Naïve Thresh	0.1878	0.1315	0.2353	0.1617	0.1635	0.1194	0.1107	0.1232	0.3930	0.1545	0.1772
Papaioannou et al. [2017]	0.0611	0.0803	0.4033	0.9954	0.0361	0.4408	0.2164	0.1426	0.9717	0.6812	0.0365
Ours f_1 only	0.0549	0.0319	0.0405	0.0456	0.0491	0.0484	0.0526	0.0501	0.0370	0.0412	0.0452
Ours f_1, f_2	0.0482	0.0267	0.0255	0.0326	0.0377	0.0291	0.0520	0.0362	0.0320	0.0261	0.0294
Ours f_1, f_2, f_3	0.0357	0.0192	0.0187	0.0276	0.0239	0.0214	0.0285	0.0255	0.0216	0.0188	0.0224

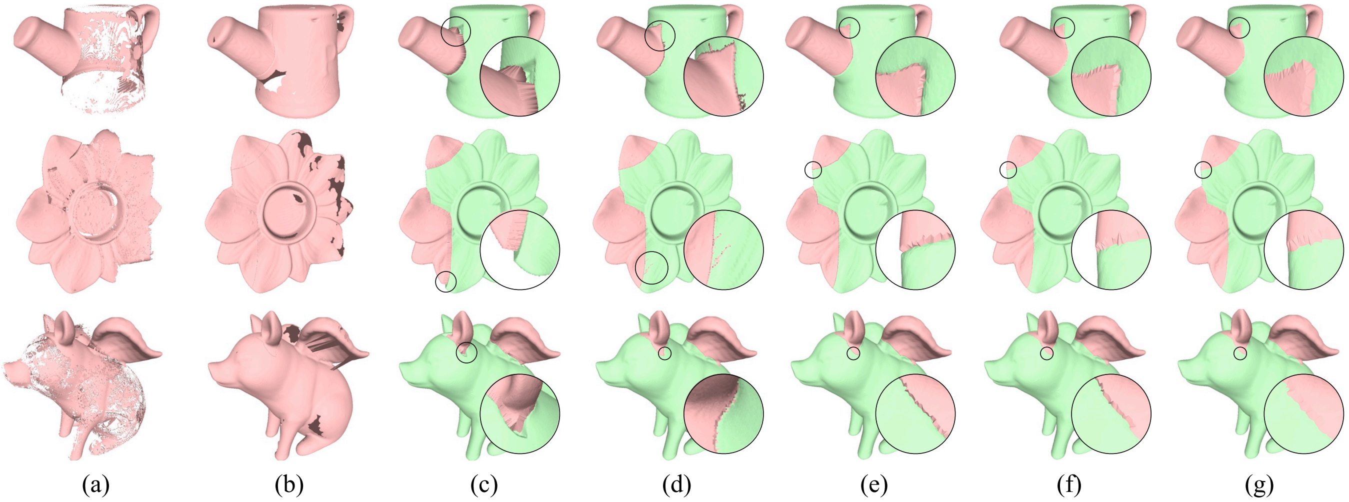


Figure 11: Restoration reconstruction results using (a) naïve Boolean subtraction, (b) Papaioannou et al. [2017], (c) naïve Euclidean distance thresholding, (d) altered Boolean subtraction, (e) our approach using f_1, (f) our approach using f_1 and f_2, and (g) our complete approach using f_1, f_2, and f_3.

lower than those generated by using naïve Boolean subtraction, altered Boolean subtraction, naïve Euclidean distance thresholding, and the approach of Papaioannou, et al. [2017]. A lower distance indicates that the restoration piece is more closely aligned with the original mesh, thereby creating a better fit. Since we compare our approach to four other approaches, our statistical tests are performed multiple times. Hence, we apply a Bonferroni correction and reject the null hypothesis if $p < 0.05/4$, i.e., if $p < 0.0125$. We state the null and alternate hypotheses as follows:

Null: There is no difference in the distance value between our approach and [comparative] approach.
Alternate: The distance value obtained by our approach is less than [comparative] approach.

Here [comparative] represents one of naïve Boolean subtraction, altered Boolean subtraction, naïve Euclidean distance thresholding, or the approach of Papaioannou, et al. [2017]. Our approach is significantly better than altered Boolean subtraction, naïve Euclidean distance thresholding, and the approach of Papaioannou, et al. [2017]

Table 2: Number of connected components for naïve Boolean subtraction and our approach, and time to generate restoration piece for our approach.

	Connected Components		Timing (s)
	Naïve Bool	Ours	Ours
Barn	182	1	71.11
Bottle	269	1	35.32
Cactus	2079	1	66.97
Candle Stand	339	1	113.11
Candle Holder	754	1	33.82
Cup	501	1	138.33
Duck	490	1	29.70
Elephant	101	1	317.59
Elephant (Big)	9424	1	135.25
Flower	439	2	74.43
Flower Bowl	297	1	183.21
Frog Cup	3171	1	322.61
Mario	618	1	11.75
Milk Jug	981	2	56.51
Ornament	603	1	149.96
Owl Cup	969	1	67.21
Pig	13316	2	113.04
Ring Holder	695	3	152.24
Salt Shaker	895	2	18.91
Upright Dog	564	1	48.79
White Dog	1635	1	33.64
Winnie	457	4	19.18

and we strongly reject the null hypothesis in all three cases with $p < 1 \times 10^{-4}$. When comparing our approach and naïve Boolean subtraction, we fail to reject the null hypothesis, and conclude that the distances generated by both approaches are not significantly different. This is to be expected as naïve Boolean subtraction provides a geometrically precise volumetric difference which retains artifacts with small mesh imprecisions, while our approach seeks to obtain a difference mesh for physical 3D printing.

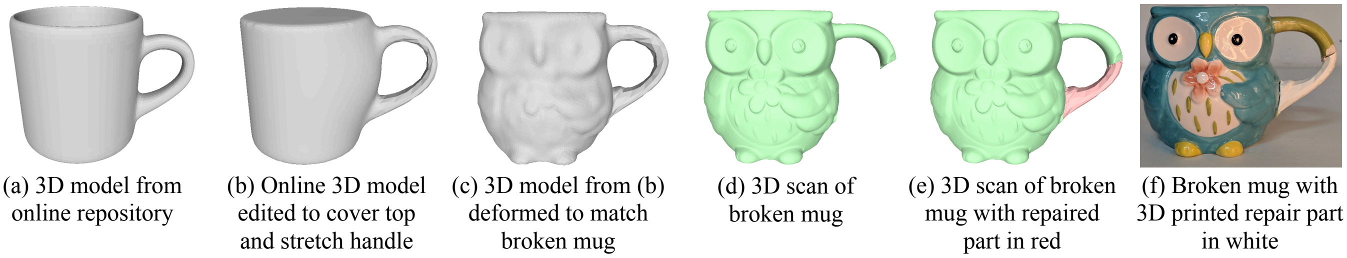
9 DISCUSSION

In this paper, we present an approach to automatically obtain restoration pieces for broken objects from a 3D scan of the object given an original counterpart scan for reference. Our approach ensures a smooth join between the restoration and the broken object by generating an exterior surface for the restoration that approaches the broken object as close as possible within a tolerance, and by projecting the exterior surface boundary onto the broken object mesh to generate a fracture surface that is nearly tangential to the exterior surface. We demonstrate the effectiveness of our approach on 22 objects with diverse sizes and geometries. The only manual steps in our approach are restricted to object setup and pre-processing, and to specifying the number of parts that have broken off, tasks that are relatively intuitive to the average user.

One limitation is that the triangulation approach to join the exterior and fracture surfaces discussed in Section 7 may fail if the broken object consists of two separate connected components set up such that the exterior surface needs to join both of them creating a hole. Another limitation is that our work assumes that the internal structure of the restored object is solid, as opposed to having a shell structure as is common in many ceramic objects. In most cases, the solid restoration does not impede functionality, however, in the case of the watering-can based salt shaker, the restoration would not contain the holes needed to release the salt. Additionally, solid 3D restorations consume more material than a restoration with a shell structure. In future work, we will investigate creating restorations with alternative geometries and topologies that preserve functionality and make optimal use of 3D print material. This may be done, for instance, by providing an interface that enables a user to mark toplogical holes in the mesh. A third concern is that current technologies limit the ability to reproduce the external colors or textures of the object for a visually seamless join. In fields such as archaeology, it is necessary to keep the restoration visually distinct from the original broken object for reversibility and ethical attribution of work [International Council of Museums 2017]. However, for common-use objects, it would be interesting to investigate techniques that enable automated non-planar 2D printing, through, for instance, decals or spray painting.

Due to the dependence of our work on small as opposed to large-scale differences between the original and broken meshes, our work is not directly applicable to creating smooth joins using publicly available 3D models whose structure may differ significantly from the broken object's original counterpart, or using algorithms that provide complete object proxies [Hermoza and Sipiran 2018; Pauly et al. 2005; Rock et al. 2015; Stutz and Geiger 2018] that do not preserve low-level fidelity with respect to the broken object. We demonstrate a functional restoration result for the owl cup in Figure 12(e) with a corresponding 3D print in Figure 12(f) obtained by deforming a publicly available cup model in Figure 12(a) to match the broken cup in Figure 12(d). The deformation was performed using geometry constraints such as vertex proximity and mesh smoothness using the approach of Sorkine and Alexa [2004]. The deformed cup is shown in Figure 12(c). However, due to the deviation in overall metric structure of the online cup from the owl cup, we had to edit it manually prior to deformation by stretching out the cup handle to match the owl cup handle as shown in Figure 12(b). Additionally, due to the use of a purely proximity-based approach as the deformation constraint, the restoration does not form a smooth join with the upper part of the handle. As part of future work, we are interested in incorporating the criteria of normal alignment and tangency directly into the deformation operation to create a result with a smooth join, and in performing hierarchical deformation to account for large-scale deviations. This will enable us to integrate our work with 3D model completion or retrieval algorithms to provide restoration pieces without the need for the original counterpart as input.

As shown in Figure 13, the precision of the join between the restoration piece and the broken object is impacted by physical constraints of 3D printing, such as the uncertainty induced by the nozzle diameter which prevents a precise connection at acute angles, and the presence of support material on the fracture surface, which may not be perfectly eliminated. As part of future work, we are interested in incorporating physical constraints into the reconstruction of the 3D restoration by factoring in the nozzle diameter,



(a) 3D model from online repository (b) Online 3D model edited to cover top and stretch handle (c) 3D model from (b) deformed to match broken mug (d) 3D scan of broken mug (e) 3D scan of broken mug with repaired part in red (f) Broken mug with 3D printed repair part in white

Figure 12: In the absence of the original unbroken object, we use a 3D model obtained from an online repository and deform it to fit the structure of the original object. 3D model from (a) obtained from TurboSquid under the Royalty Free License, with credit attributed to TurboSquid user *ead12*.

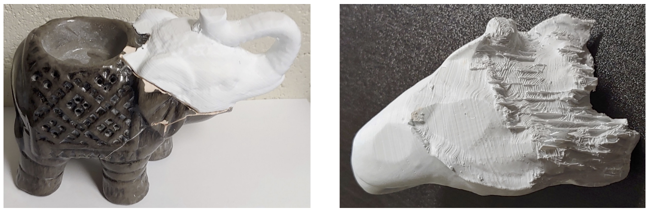


Figure 13: The 3D print of the elephant head does not make a perfect join with the elephant (left) due to uncertainties introduced by the nozzle diameter, and the presence of support structure remnants on the fracture surface (right).

generating joinable 3D printed parts that can be connected to form large restoration pieces that cannot be printed intact on consumer printers, and automatically recommending optimal configurations for 3D printing a restoration piece by orienting the piece to avoid support structures on the fracture surface. Connecting our restoration reconstruction approach with automated recommendation for 3D printing parts will enable an average user to have a fully automated end-to-end system for restoring their broken objects.

ACKNOWLEDGMENTS

This work was partially supported by NSF grant #1730183.

REFERENCES

Kaja Antlej, Miran Eric, Mojca Šavnik, Bernarda Županek, Janja Slabe, and Borut Battestin. 2011. Combining 3D technologies in the field of cultural heritage: three case studies. In *Proceedings of the Conference VAST: The 12th International Symposium on Virtual Reality, Archaeology and Cultural Heritage–Short and Project Papers (Prato 2011)*. The Eurographics Association, 1–4.

Lucia Arbace, Elisabetta Sonnino, Marco Callieri, Matteo Dellepiane, Matteo Fabbri, Antonio Iaccarino Idelson, and Roberto Scopigno. 2013. Innovative uses of 3D digital technologies to assist the restoration of a fragmented terracotta statue. *Journal of Cultural Heritage* 14, 4 (2013), 332–345.

Paul J. Besl and Neil D. McKay. 1992. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 2 (Feb. 1992), 239–256. https://doi.org/10.1109/34.121791

Marcel Campen and Leif Kobbelt. 2010. Exact and robust (self-) intersections for polygonal meshes.. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, The Eurographics Association and Blackwell Publishing Ltd, Malden, MA, 397–406.

Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. 2008. Meshlab: an open-source mesh processing tool.. In *Eurographics Italian chapter conference*, Vol. 2008. The Eurographics Association, Salerno, Italy, 129–136.

Edsger W Dijkstra. 1959. A note on two problems in connexion with graphs. *Numerische mathematik* 1, 1 (1959), 269–271.

Massimiliano Fantini, Francesca de Crescenzio, Franco Persiani, Stefano Benazzi, and Giorgio Gruppioni. 2008. 3D restitution, restoration and prototyping of a medieval damaged skull. *Rapid Prototyping Journal* 14, 5 (2008), 318–324.

Roberto L Flores, Hannah Liss, Samuel Raffaelli, Aiza Humayun, Kimberly S Khouri, Paulo G Coelho, and Lukasz Witek. 2017. The technique for 3D printing patient-specific models for auricular reconstruction. *Journal of Cranio-Maxillofacial Surgery* 45, 6 (2017), 937–943.

Jian Gao, Xin Chen, Oguzhan Yilmaz, and Nabil Gindy. 2008. An integrated adaptive repair solution for complex aerospace components through geometry reconstruction. *The International Journal of Advanced Manufacturing Technology* 36, 11-12 (2008), 1170–1179.

Jian Gao, Janet Folkes, Oguzhan Yilmaz, and Nabil Gindy. 2005. Investigation of a 3D nonâĂŘcontact measurement based blade repair integration system. *Aircraft Engineering and Aerospace Technology* 77, 1 (2005), 34–41. https://doi.org/10.1108/00022660510576028

Robert Gregor, Ivan Sipiran, Georgios Papaioannou, Tobias Schreck, Anthousis Andreadis, and Pavlos Mavridis. 2014. Towards Automated 3D Reconstruction of Defective Cultural Heritage Objects.. In *Proceedings of the Eurographics Workshop on Graphics and Cultural Heritage (GCH '14)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 135–144. https://doi.org/10.2312/gch.20141311

Ola LA Harrysson, Yasser A Hosni, and Jamal F Nayfeh. 2007. Custom-designed orthopedic implants evaluated using finite element analysis of patient-specific computed tomography data: femoral-component case study. *BMC musculoskeletal disorders* 8, 1 (2007), 91.

Renato Hermoza and Ivan Sipiran. 2018. 3D Reconstruction of Incomplete Archaeological Objects Using a Generative Adversarial Network. In *Proceedings of Computer Graphics International 2018 (CGI 2018)*. ACM, New York, NY, USA, 5–11.

Ping Hu, Changsheng Wang, Baojun Li, and Mingzeng Liu. 2012. Filling Holes in Triangular Meshes in Engineering. *JSW* 7, 1 (2012), 141–148.

International Council of Museums. 2017. *ICOM Code of Ethics for Museums.* International Council of Museums, Paris, France. 30 pages.

Alec Jacobson et al. 2018. gptoolbox: Geometry Processing Toolbox. http://github.com/alecjacobson/gptoolbox.

Jane Lee. 2013. 5 Ways Smithsonian Uses 3-D Scanning to Open Up History. http://news.nationalgeographic.com/news/2013/09/130904-3d-printing-smithsonian-whale-skeleton-technology-science/. *National Geographic* 9 (2013), 1.

Nicolas Mellado, Dror Aiger, and Niloy J Mitra. 2014. Super 4pcs fast global pointcloud registration via smart indexing.. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, The Eurographics Association and Blackwell Publishing Ltd, Malden, MA, 205–215.

Georgios Papaioannou, Tobias Schreck, Anthousis Andreadis, Pavlos Mavridis, Robert Gregor, Ivan Sipiran, and Konstantinos Vardis. 2017. From reassembly to object completion: A complete systems pipeline. *Journal on Computing and Cultural Heritage (JOCCH)* 10, 2 (2017), 8.

Mark Pauly, Niloy J. Mitra, Joachim Giesen, Markus Gross, and Leonidas J. Guibas. 2005. Example-based 3D Scan Completion. In *Proceedings of the Third Eurographics Symposium on Geometry Processing (SGP '05)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 23–32.

Fabian Rengier, Amit Mehndiratta, Hendrik Von Tengg-Kobligk, Christian M Zechmann, Roland Unterhinninghofen, H-U Kauczor, and Frederik L Giesel. 2010. 3D printing based on imaging data: review of medical applications. *International journal of computer assisted radiology and surgery* 5, 4 (2010), 335–341.

Jason Rock, Tanmay Gupta, Justin Thorsen, JunYoung Gwak, Daeyun Shin, and Derek Hoiem. 2015. Completing 3d object shape from one depth image.. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Piscataway, NJ, 2484–2493.

Roberto Scopigno, Marco Callieri, Paolo Cignoni, Massimiliano Corsini, Matteo Dellepiane, Federico Ponchio, and Guido Ranzuglia. 2011. 3D models for cultural heritage: beyond plain visualization. *Computer* 44, 7 (2011), 48–55.

Maria Luiza Seixas, Paulo Santos Assis, João Cura DâĂŽArs Figueiredo Jr, Maria Aparecida Pinto, and Daniella Gualberto Caldeira Paula. 2018. The use of rapid prototyping in the joining of fractured historical silver object. *Rapid Prototyping Journal* 24, 3 (2018), 532–538.

Sekou Singare, Yaxiong Liu, Dichen Li, Bingheng Lu, and Sanhu He. 2008. Individually prefabricated prosthesis for maxilla reconstruction. *Journal of Prosthodontics* 17, 2 (2008), 135–140.

Ivan Sipiran, Robert Gregor, and Tobias Schreck. 2014. Approximate Symmetry Detection in Partial 3D Meshes. *Comput. Graph. Forum* 33, 7 (Oct. 2014), 131–140.

Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. 2004. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. ACM, New York, NY, 175–184.

David Stutz and Andreas Geiger. 2018. Learning 3d shape completion from laser scan data with weak supervision.. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Piscataway, NJ, 1955–1964.

Nigel Williams. 2002. *Porcelain: repair and restoration, a handbook*. University of Pennsylvania Press, Philadelphia, PA.

Lukasz Witek, Kimberly S Khouri, Paulo G Coelho, and Roberto L Flores. 2016. Patient-specific 3D models for autogenous ear reconstruction. *Plastic and Reconstructive Surgery–Global Open* 4, 10 (2016), e1093.

Jianming Zheng, Zhongguo Li, and Xi Chen. 2006. Worn area modeling for automating the repair of turbine blades. *The International Journal of Advanced Manufacturing Technology* 29, 9-10 (2006), 1062–1067.

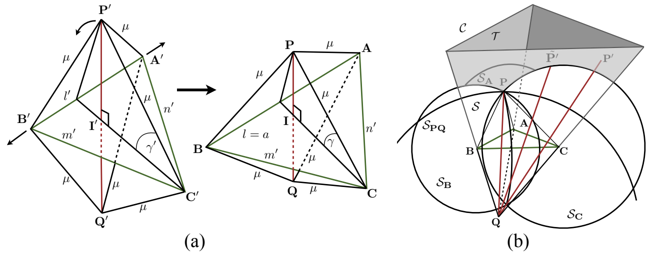
A PROOFS FOR COMPUTATION OF η

Theorem A.1. *Given a line segment* $\mathbf{PQ}$ *from mesh* O *intersecting with a triangle* $\Delta\mathbf{ABC}$ *from mesh* B*, such that* $\mathbf{P}$ *and* $\mathbf{Q}$ *are each exactly* μ *units away from the vertices* $\mathbf{A}$, $\mathbf{B}$, *and* $\mathbf{C}$. *The length of* $\mathbf{PQ}$ *is the smallest possible if* $\Delta\mathbf{ABC}$ *is an equilateral triangle with side length being* a*, the length of the longest edge in* O.

Proof: Consider an initial scalene triangle $\Delta\mathbf{A'B'C'}$ on the left of Figure 14(a), and an initial line segment $\mathbf{P'Q'}$, such that $\mathbf{P'}$ and $\mathbf{Q'}$ are each exactly μ units away from vertices $\mathbf{A'}$, $\mathbf{B'}$, and $\mathbf{C'}$. $\mathbf{P'Q'}$ is then perpendicular to and bisected by $\Delta\mathbf{ABC}$. The proof can be intuited using a paper-unfolding approach. If the edge $\mathbf{A'B'}$ in Figure 14(a) is changed by pulling points $\mathbf{A'}$ and $\mathbf{B'}$ to points $\mathbf{A}$ and $\mathbf{B}$ in the diagram on the right, then in order to preserve the edge lengths of $\mathbf{P'A'}$, $\mathbf{P'B'}$, and $\mathbf{P'C'}$ to be μ, the two triangles $\mathbf{P'A'C'}$ and $\mathbf{P'B'C'}$ need to be unfolded out along the crease $\mathbf{P'C'}$. This will induce the point $\mathbf{P'}$ to move downward along the curved arrow in Figure 14(a), causing the angle γ' to reduce to γ, i.e., $\gamma < \gamma'$. $\mathbf{P'I'} = \mu \sin\gamma'$, and $\mathbf{PI} = \mu\sin\gamma$.

It can be shown that γ' is acute by observing that $\angle\mathbf{P'C'B'}$ and $\angle\mathbf{P'A'C'}$ are both acute angles since triangles $\mathbf{P'B'C'}$ and $\mathbf{P'A'C'}$ are isosceles about $\mathbf{P'}$, and that a cross-sectional triangle of the tetrahedron $\mathbf{P'A'C'B'}$ containing $\mathbf{P'C'}$ will have the angle at $\mathbf{C'}$ be lesser than the greater of $\angle\mathbf{P'C'B'}$ and $\angle\mathbf{P'A'C'}$. In particular, the angle, i.e., γ' will be the lowest when the cross-section contains $\mathbf{P'I'}$. Similarly γ is also acute. For acute angles γ and γ', if $\gamma < \gamma'$, then $\sin\gamma < \sin\gamma'$. Therefore $||\mathbf{PI}|| < ||\mathbf{P'I'}||$. Keeping m' and n' constant, $||\mathbf{PI}||$ is least when $\mathbf{AB}$ attains a length $l = a$.

A similar analysis with the tetrahedron $\mathbf{Q'A'B'C'}$ shows that $||\mathbf{QI}||$ is least when $\mathbf{AB}$ attains a length of $l = a$, so that $\mathbf{PQ}$ will have the least length for $\mathbf{AB}$ having length a. After changing l' to $l = a$, the stretching can be repeated on m' and then on n' to show that the length of $\mathbf{PI}$ attains the least value for $\mathbf{AB}$, $\mathbf{BC}$, and $\mathbf{CA}$ each having lengths a, or $\Delta\mathbf{ABC}$ being an equilateral triangle with side length a.



(a) (b)

Figure 14: (a) Pulling apart points A′ and B′ causes the point P′ to come closer to the triangle ABC so that it is closest when ABC is an equilateral triangle with side length being the longest edge in O. (b) For triangle ABC, PQ shown is smaller than any line segment P′ with point P′ in the interior or boundary of the shaded region devoid of P. PQ is smallest when the sides PA, PB, and PC attain their least value μ.

Theorem A.2. *Given a line segment PQ from mesh O intersecting with an equilateral triangle ΔABC from mesh B, such that P and Q are each at least μ units away from the vertices A, B, and C. The length of PQ is the smallest if P and Q are each exactly μ units away from vertices A, B, and C.*

Proof: For the equilateral triangle $\Delta\mathbf{ABC}$, consider the line segment $\mathbf{PQ}$ that intersects $\Delta\mathbf{ABC}$, and where $\mathbf{P}$ and $\mathbf{Q}$ are each at least μ units away from $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, i.e., $||\mathbf{PA}|| \geq \mu$, $||\mathbf{PB}|| \geq \mu$, and $||\mathbf{PC}|| \geq \mu$. Then, $\mathbf{P}$ is the intersection point of the three closed spheres $\mathcal{S}_\mathbf{A}$, $\mathcal{S}_\mathbf{B}$, and $\mathcal{S}_\mathbf{C}$, whose radii are $||\mathbf{PA}||$, $||\mathbf{PB}||$, and $||\mathbf{PC}||$ respectively. In order for the line segment $\mathbf{PQ}$ to intersect the triangle $\Delta\mathbf{ABC}$, it needs to be confined in the pyramidal cone $\mathcal{C}$.

Consider a point $\mathbf{P'}$ chosen in the space $\mathcal{T} = \mathcal{C} \setminus (\mathcal{S}_\mathbf{A} \cup \mathcal{S}_\mathbf{B} \cup \mathcal{S}_\mathbf{C})$. This space is the interior of the shaded region shown in Figure 14(b). For this point, $||\mathbf{P'A}|| > ||\mathbf{PA}||$, $||\mathbf{P'B}|| > ||\mathbf{PB}||$, and $||\mathbf{P'C}|| > ||\mathbf{PC}||$. The line segment $\mathbf{P'Q}$ lies outside the region of intersection of the cone $\mathcal{C}$ and the sphere $\mathcal{S}_\mathbf{PQ}$ traced by $\mathbf{P}$ about $\mathbf{Q}$, i.e., outside of $\mathcal{S}_\mathbf{PQ} \cap \mathcal{C}$. So $||\mathbf{P'Q}|| < ||\mathbf{PQ}||$.

Consider a second point $\tilde{\mathbf{P'}}$ chosen on $((\delta\mathcal{S}_\mathbf{A} \cup \delta\mathcal{S}_\mathbf{B} \cup \delta\mathcal{S}_\mathbf{C}) \cap \mathcal{C}) \setminus \{\mathbf{P}\}$, which represents the boundary of the shaded region at the surface of the spheres devoid of $\mathbf{P}$. In Figure 14(b) $\tilde{\mathbf{P'}}$ is confined to the boundary of the sphere $\mathcal{S}_\mathbf{C}$, i.e., to $(\delta\mathcal{S}_\mathbf{C} \setminus (\delta\mathcal{S}_\mathbf{A} \cup \delta\mathcal{S}_\mathbf{B})) \cap \mathcal{C}$, so that $||\tilde{\mathbf{P'}}\mathbf{C}|| = ||\mathbf{PC}||$. To reach $\tilde{\mathbf{P'}}$ from $\mathbf{A}$ or $\mathbf{B}$, one needs to step out of at least one of $\mathcal{S}_\mathbf{A}$ or $\mathcal{S}_\mathbf{B}$, i.e., at least one of the following conditions must hold: $||\tilde{\mathbf{P'}}\mathbf{A}|| > ||\mathbf{PA}||$ or $||\tilde{\mathbf{P'}}\mathbf{B}|| > ||\mathbf{PB}||$. The line segment $\tilde{\mathbf{P'}}\mathbf{Q}$ lies outside $\mathcal{S}_\mathbf{PQ} \cap \mathcal{C}$. So $||\tilde{\mathbf{P'}}\mathbf{Q}|| < ||\mathbf{PQ}||$. The same effect can be seen for the choice of a point on the boundaries $(\delta\mathcal{S}_\mathbf{A} \setminus (\delta\mathcal{S}_\mathbf{B} \cup \delta\mathcal{S}_\mathbf{C})) \cap \mathcal{C}$, and $(\delta\mathcal{S}_\mathbf{B} \setminus (\delta\mathcal{S}_\mathbf{A} \cup \delta\mathcal{S}_\mathbf{C})) \cap \mathcal{C}$.

Thus, $||\mathbf{P'Q}|| > ||\mathbf{PQ}||$ if any one of the following hold: $||\mathbf{P'A}|| > ||\mathbf{PB}||$, $||\mathbf{P'B}|| > ||\mathbf{PB}||$, or $||\mathbf{P'C}|| > ||\mathbf{PC}||$. Since the lower bound on the distance of $\mathbf{P}$ from any vertex in the mesh B is μ, the smallest values that $||\mathbf{PA}||$, $||\mathbf{PB}||$, or $||\mathbf{PC}||$ can take is μ. Therefore, any vertex $\mathbf{P'}$ that is further than μ units from at least one of the vertices $\mathbf{A}$, $\mathbf{B}$, or $\mathbf{C}$ is further away from $\mathbf{Q}$ than $\mathbf{P}$ is from $\mathbf{Q}$. One can do a similar analysis to see that any vertex $\mathbf{Q'}$ that is further than μ units from at least one of the vertices $\mathbf{A}$, $\mathbf{B}$, or $\mathbf{C}$ is further away from $\mathbf{P}$ than $\mathbf{Q}$ is from $\mathbf{P}$. Thus $\mathbf{PQ}$ is the smallest length line segment that intersects $\Delta\mathbf{ABC}$.