

Automated Surface Extraction: Adaptive Remeshing Meets Lagrangian Shrink-Wrapping

Martin Čavarga
Comenius University Bratislava,
Mlynská dolina 5692, 841 04, Bratislava, Slovak Republic,
martin.cavarga@fmph.uniba.sk

Abstract

Fairing methods, frequently used for smoothing noisy features of surfaces, evolve a surface towards a simpler shape. The process of shaping a simple surface into a more complex object requires using a scalar field defined in the ambient space to drive the surface towards a target shape. Practical implementation of such evolution, referred to as Lagrangian Shrink-Wrapping (LSW), on discrete mesh surfaces presents a variety of challenges. Our key innovation lies in the integration of adaptive remeshing and curvature-based feature detection, ensuring mesh quality and proximity to target data. We introduce the Equilateral Triangle Jacobian Condition Number metric for assessing triangle quality, and introduce trilinear interpolation for enhanced surface detailing to improve upon existing implementations. Our approach is tested with point cloud meshing, isosurface extraction, and the elimination of internal mesh data, providing significant improvements in efficiency and accuracy. Moreover we extend the evolution to surfaces with higher genus to shrink-wrap even more complex data.

Keywords

fairing, surface evolution, Lagrangian Shrink-Wrapping, adaptive remeshing, feature detection, mesh quality, point cloud meshing, isosurface extraction, mesh simplification

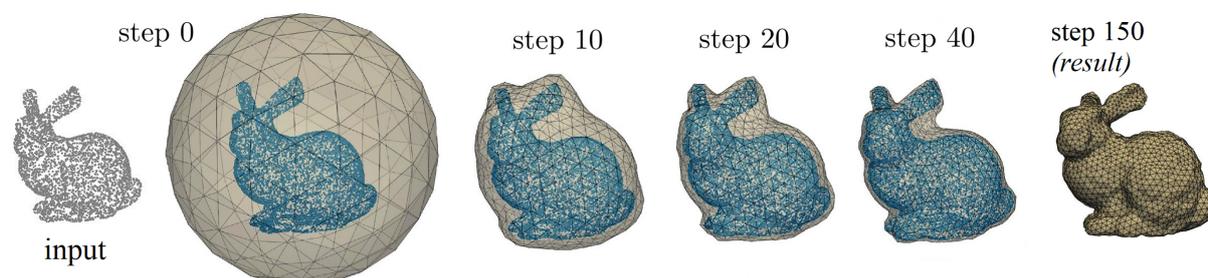


Figure 1: Lagrangian shrink-wrapping (LSW) evolution with the help of *adaptive remeshing* of input point cloud sampling of *bunny* with 12K vertices over 150 steps with an *icosphere* starting surface with subdivision level $s = 2$.

1 INTRODUCTION

Shape extraction from diverse input, including image data, point clouds, or mesh surfaces, is ubiquitous in modern geometry processing. A critical challenge in this domain is maintaining the quality of mesh elements extracted from these representations, which is essential for downstream processing applications such as rendering or numerical simulations. Traditional methods of direct triangulation of point clouds often require point normals, or otherwise struggle to produce watertight manifold surfaces. Unlike direct processing techniques, evolutionary methods make use of principles like diffusion described by partial differential equations which can be equipped with spatial information of input data, such as distance fields, to evolve a simple starting sur-

face with known topological properties, for example, a sphere, into a shape with the desired amount of detail.

1.1 Contributions

This work introduces a robust and stable shrink-wrapping tool designed for voxel, mesh, and point cloud data (see Fig. 1), employing a triangle surface mesh. Our contributions include:

- In contrast with prior approaches, the surface is no longer subject to the stretching of linear discretization elements (mesh triangles), essentially pushing the solution closer to how a smooth surface would evolve. This places our shrink-wrapping algorithm among the state-of-the-art methods which also employ some form of adaptive remeshing.

- Compared to existing implementations, we use trilinear interpolation of (possibly truncated) distance voxel values instead of nearest neighbor for shape precision.
- Extension of the tool’s applicability to initial surfaces constructed from distance contours broadens its utility to higher-genus target data¹.
- The use of customized equilateral triangle quality metric, validating the production of high-quality meshes.
- We adopt a curvature-based feature detection technique, facilitating a more accurate representation of intricate geometric features.

2 RELATED WORK

Our exploration stands on the contributions in significant areas of application: *fairing*, *remeshing*, *image segmentation*, *surface reconstruction*, and *geometry simplification*.

The fairing approaches utilize the key diffusion term in the evolutionary equations. In its simplest form it involves smoothing out all features of an input surface as performed using a semi-implicit formulation of the standard *mean curvature flow* (MCF) by Mikula et al. [2], and minimizing *Willmore energy* by Crane et al. [3]. This leads also towards unknotting tangled higher-genus surfaces by Yu et al. [4] who also considered a point cloud or mesh obstacle repelling an evolving surface. The outputs of such obstacle problem can be stacked on top of each other in the form of *nested cages* studied by Sach et al. [5].

Surface extraction from image data involves augmenting the standard MCF by *advection* which was also explored by Mikula et al. [2] in Section 3.2. These methods date back all the way to perhaps the most widespread brain extraction tomography application called BET2 first published by Smith in 2002 [6]. BET2 extracts a surface from an initial icosphere mesh which evolves towards the boundary of human brain, driven by advection evaluated from the thresholds computed within the histogram of the tomography image.

We put significant emphasis on the development of point cloud reconstruction tools developed by Daniel et al. [7] in collaboration with the authors of [2]. A suitable example of recent evolutionary approaches to surface reconstruction from point cloud data is Point2Mesh which also uses neural networks to evolve convex hull starting surfaces [8].

¹ We consider the genus of a surface envelope of a triangle soup or a point cloud target set in the sense of Hurtado et al. [1].

Since, as we mentioned, evolutionary methods can use simpler initial surfaces, they can also serve as simplification tools for enveloping meshes with undesired internal cavities and non-manifold vertices or edges [1].

This leads us to the use of evolutionary shrink-wrapping for the purposes of *remeshing*, as done by Kobbelt et al. [9], and extended to outward-evolving quad surface patches by Huska et al. [10].

3 METHODOLOGY

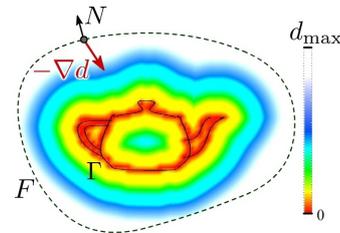


Figure 2: Slice of distance field d to surface Γ Utah Teapot with resolution 120^3 and an evolving surface F driven by fields d and $-\nabla d$. Source: [11].

3.1 Surface Evolution

Let X be a Riemannian 2-manifold. For a time interval $[0, T]$, the map $F : [0, T] \times X \rightarrow \mathbb{R}^3$ defines surface evolution in \mathbb{R}^3 , governed by:

$$\partial_t F = v_N + v_T, \quad (1)$$

where v_N and v_T are the normal and tangential components of velocity respectively. To ensure that the decomposition into normal and tangential components at $F(x) \in \mathbb{R}^3$ is well-defined at each point $x \in X$, $F^t = F(t, \cdot)$ must be an immersion of X into \mathbb{R}^3 for all $t \in [0, T]$. Equation (1) is accompanied by an initial immersion $F^0 = F(0, \cdot)$. First-order Laplacian smoothing follows $\partial_t F = \Delta_{g_F} F = -2HN$, where $H = (\kappa_{\min} + \kappa_{\max})/2$ is the mean curvature and N the outward unit normal. As demonstrated in Section 4.2 of [11], numerical simulations using the Laplace-Beltrami formulation with respect to the metric g_F of immersion F converge to the behavior of mean curvature flow (MCF) for the shrinking sphere solution $r(t) = \sqrt{r_0^2 - 4t}$, beginning from radius r_0 .²

Now let $\Gamma \subset \mathbb{R}^3$ be a *target set*, potentially a non-manifold surface (see Fig.2) or a point cloud. Let $d : \mathbb{R}^3 \rightarrow \mathbb{R}$ be the distance field of Γ , taking either the unsigned d^+ or signed d^\pm form, the latter of which distinguishes the interior $\text{Int}(\Gamma)$ from the exterior $\text{Ext}(\Gamma)$ of Γ with negative and positive sign respectively.

The target set Γ generates its distance field d in its ambient space \mathbb{R}^3 , and for this reason, it can affect the evolution of F in the following advection-diffusion model:

$$\partial_t F = \varepsilon \Delta_{g_F} F + \eta N + \rho v_T, \quad F(0, \cdot) = F^0, \quad (2)$$

² This alignment underscores the practical equivalence of the Laplacian smoothing approach to MCF in numerical settings.

where ε, ρ , and η are control functions, namely

$$\varepsilon(d) := C_1(1 - e^{-d^2/C_2}), \quad C_1, C_2 > 0, \quad (3)$$

$$\rho(d) := A(1 - e^{-d^2}), \quad A > 0, \quad (4)$$

$$\eta(d) := D_1 d(-\nabla d \cdot N) - D_2 \sqrt{1 - (\nabla d \cdot N)^2}, \quad (5)$$

$$D_1 > 0, \quad D_2 \geq 0,$$

inspired by [10] using the distance field d to target Γ . With ε we ensure that MCF slows down to zero as F approaches Γ , we also use a similar control function ρ to slow down the effects of tangential redistribution near Γ , and finally η controls the orientation of normal N to surface F (see Fig. 2). To make sure that F does not evolve past Γ under the influence of MCF, we can use modified weight functions ε^+ and η^+ which are non-zero only for positive values $d > 0$.

We also use the *angle-based tangential redistribution* inspired by [10] for computing velocity v_T with preferred weight $\omega = 0.05$ throughout our experiments.

3.2 The Discrete Picture

The evolving surface F is represented by a triangle mesh (K, V) where K is an abstract simplicial complex and $V \subset F[X] \subset \mathbb{R}^3$ the vertex set, such that its *geometric realization* — in the sense of Hoppe et al. [12] — is a compact manifold surface without boundary. Scalar and vector fields (d and $-\nabla d$) are sampled on regular voxel grids $G \subset \mathbb{R}^3$ containing Γ with cell size $c_G > 0$ using *trilinear interpolation*. In our experiments, we compute cell size as $c_G = \beta_{\min}/40$, that is: 40 voxels per minimal dimension of the target set's bounding box.

We use the *icosphere* subdivision surface (see "step 0" in Fig.1) for simulations with spherical starting surfaces, and for another set of simulations we use *isosurfaces* of the distance field $S_{d_0} = \{\mathbf{x} \in \mathbb{R}^3 \mid d(\mathbf{x}) = d_0\}$ reconstructed via the *Marching Cubes* algorithm [13] after which it is processed with a single step of *adaptive remeshing* by Dunyach et al. [14] to achieve more uniform vertex density³. The adaptive remeshing algorithm is then used during evolution when necessary⁴.

Distance field d is computed using a multi-step approach from Section 2.1 in [11] accelerated by spatial data structures like AABB tree and Octree, and using the *Fast-Sweeping* algorithm by Zhao [16]. The gradient ∇d is computed as a central difference of respective neighboring cell values for each cell.

Non-linear parabolic equation 2 is discretized using finite 2-volumes (areas) V_i surrounding mesh vertices

³ The PMP library [15] allows us to use only one iteration of `pmp::Remeshing::adaptive_remeshing` if the resulting mesh quality suffices.

⁴ when the measured polygon quality drops below the desired level.

$F_i, i = 1, \dots, N_V = |V|$ (see Fig. 3). The concept of these, so called, *co-volumes* is explained in more detail by Meyer et al. [17] and Mikula et al. [2]. In particular, the Laplace-Beltrami operator in (2) is discretized using a cotangent scheme when compiling the underlying sparse linear system of dimension $N_V \times N_V$. Analogously to [10], the advection terms in our case $\eta N + \rho v_T$ correspond to the state of the previous time step during evolution, and thus are added to the system's right-hand side.

The linear system is solved for each coordinate of vertex F_i and time step $t \in [0, T]$. Our preferred solver is BiCGStab, but according to [2] the sparse SOR method can also be used.

3.3 Numerical Stability

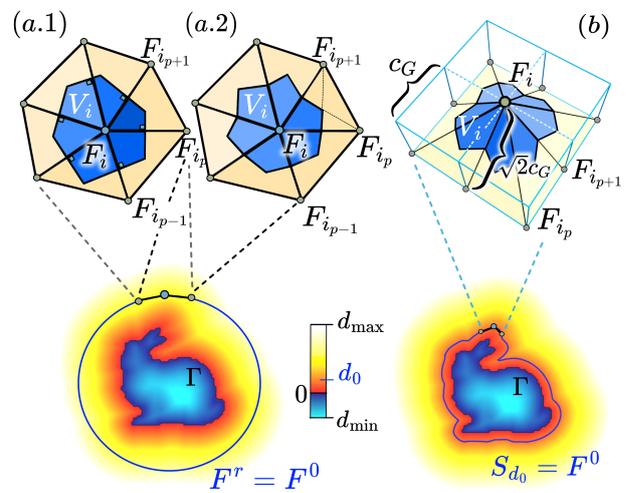


Figure 3: Two distinct types of Laplacian co-volumes evaluated for icosphere starting surface F^r : *Voronoi* (a.1) and *barycentric* (a.2) [17]. The starting surface can also be computed as a level set S_{d_0} of the distance field d where the maximum expected size of the (barycentric) co-volume can be estimated from 4 neighboring voxels with side $c_G > 0$ (b).

The stability of the sparse linear system derived from (2) relies heavily on the spectral radius $\rho(\mathbf{A})$ of the system matrix \mathbf{A} with non-negative diagonal entries. According to Section 3 in [18], the semi-implicit finite volume approach for curves in \mathbb{R}^2 leads to stability constraint $\tau \approx h^2$ where $h > 0$ is a spatial step and $\tau > 0$ a time step. In the co-volume formulation on F this translates to $\tau \approx \mu(V)$ where $\mu(V)$ is the measure (area) of a representative co-volume V . Large deviations result in the formation of singularities⁵ in F . However, variability of measures $\mu(V)$ in unstructured meshes allows only an approximate control ensuring $\rho(\mathbf{A}) \leq 1$.

⁵ At first glance, we consider these singularities qualitatively different from those handled by Kazhdan et al. [19].

The simplest available curvature-preserving heuristic involves uniform scaling the evolving surface F and its corresponding distance field d to target set Γ by factor

$$\phi = (\tau / (\sigma \mu_0(V)))^{1/2}, \quad (6)$$

where $\sigma > 0$ is a, so called, *shrink factor* which accounts for the decrement in average $\mu(V)$. Whenever one requires a result F^t in the original size we simply scale by the inverse of (6). The mean initial co-volume measure $\mu_0(V)$ is estimated from the starting geometry assuming uniform vertex density over the surface. In the case of an icosphere with radius $r > 0$, we put $\mu_0(V) = 4\pi r^2 / N_V^s$, where

$$N_V^s = (N_E^0(4^s - 1) + 3N_V^0) / 3 \quad (7)$$

is the number of vertices under subdivision level $s \geq 0$ with $N_V^0 = 12$ vertices and $N_E^0 = 30$ edges of the base icosahedron. For the proof of formula (7) see Chapter 5 in [20]. With initial isosurface $F^0 = S_{d_0}$, we estimate $\mu_0(V) = c_G^2(4\sqrt{2})/3$ for voxel size $c_G > 0$ (see Fig. 3 (b)).

Throughout our experiments with the shrink-wrapping evolution we keep track of the value $\mu(V)$ per vertex as well as its bounds and average.

3.4 Remeshing and Feature Preservation

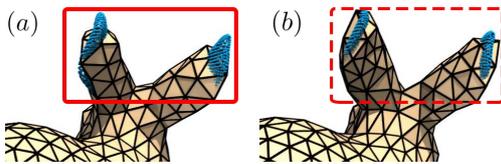


Figure 4: A trade-off between local triangle quality and feature preservation: Results after using cosine-based (a) and curvature-based (b) feature detection. Points of the target set Γ are shown in blue.

Adaptive remeshing [14] introduces deviations from the evolving surface F when applied. This can be mitigated using control functions ε^+ and η^+ with strictly positive support for signed distance fields that can even be shifted by some value $\tilde{d}_0 > 0$ to achieve better results. Sparse point cloud sets Γ , however, do not provide values $d < 0$, and require the use of different techniques for freezing points which should not sink below their respective feature.

In mesh processing, features refer to distinctive subsets of the mesh surfaces, such as sharp corners, creases, and boundaries, which carry important information about the shape of the object being represented. We need the sensitivity to *true features*⁶, and on the other hand avoid marking false positives at *convex-dominant saddle* (CDS) *points* (see Fig.5 (b)), that is: saddle points with much higher positive curvature.

⁶ Stemming from the shape of generating set Γ .

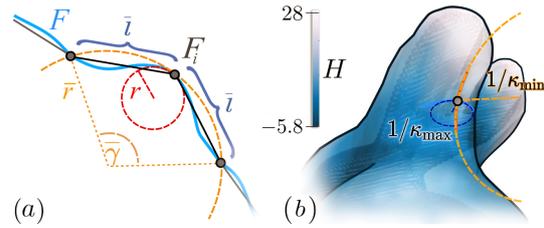


Figure 5: (a): An estimate of mean curvature angle $\bar{\gamma}$ at vertex F_i . (b): The imbalance of principal curvatures κ_{\max} and κ_{\min} at a *convex-dominant saddle vertex*. The color values show mean curvature H .

Since we require an automatic process during surface evolution, we propose a vertex-based detection evaluating the *angle of mean curvature* $\gamma = Hl$ where l is the arc length of smooth surface F . In the discrete setting we evaluate $\bar{\gamma}_i$ at vertex $\bar{F}_i = F_i$ using *neighborhood mean edge length*⁷: $\bar{l} = \frac{1}{m} \sum_{p=1}^m \|F_{i_p} - F_i\|$ where m is the valence of vertex F_i , and the cotangent estimate of mean curvature \bar{H} from [17].

Using $\bar{\gamma}$ we convert mean curvature H to a scale-invariant quantity. Vertices with $\bar{\gamma} = 2\bar{l}\bar{H} < \gamma_{\text{crit}}$ are marked as feature. Unfortunately, this alone leads to a decrease of mesh quality in CDS points – saddle points where $|\kappa_{\max}| < K|\kappa_{\min}|$ for curvature imbalance factor $K > 1$. Such feature elements can be false positives, and we propose not to mark them. Additionally, we should also not mark vertices with valence $m > 6$ because the following adaptive remeshing steps will avoid fixing them.

Considering the extent of sizes β_{\min} , β_{\max} of axis-aligned bounding box of Γ and box expansion factor $\zeta > 0$ for the distance field d we use empirical estimate

$$r \approx 0.4(\beta_{\min} + (0.5 + \zeta)\beta_{\max}) \quad (8)$$

for the radius of the starting icosphere with subdivision level s . The sizing for remeshing [14] is computed as

$$l_{\min} = 2\lambda_{\min}r \sin(\gamma_{\text{ico}}2^{-(s+1)}), \quad l_{\max} = \lambda_{\max}l_{\min}, \quad (9)$$

where $\gamma_{\text{ico}} = 2\pi/5$ is the angular segmentation of the base icosahedron, and $\lambda_{\max} > \lambda_{\min} > 0$ are controllable scale factors. Throughout experiments in Section 4 we empirically put $\lambda_{\min} = 0.14$, and $\lambda_{\max} = 4$. For the isosurface evolution we change the minimum edge length factor to $\lambda_{\min} = 0.4$ and put

$$l_{\min} = \lambda_{\min}\sqrt{2}c_G, \quad l_{\max} = \lambda_{\max}l_{\min}, \quad (10)$$

where c_G is the cell size of the distance field grid. Combined with error $\varepsilon_{\text{rem}} = \max\{\|F - \bar{F}\|\}$ between linear approximation \bar{F} and surface arc F , we obtain adaptive sizing values for all edges [14] (we set this value to $(l_{\min} + l_{\max})/4$). For our evolving surface, the sizing needs to be adapted to stabilized scale ϕ .

⁷ Also used by [6] to estimate the local radius of curvature for a surface update term.

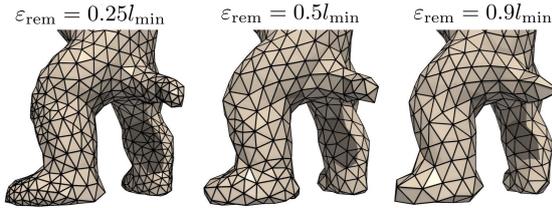


Figure 6: The effect of changing approximation error ϵ_{rem} after 80 time steps for adaptive remeshing as a multiple of minimum edge length $l_{min} = 0.0799$ of *Armadillo* (in stabilized scale $\phi = 0.0261$).

It should also be noted that strong (low- ϵ_{rem}) adaptivity may adversely affect co-volume sizing necessary for stability (see Fig. 6), especially for feature vertices.

To achieve higher level of detail for surface F close to Γ , the mesh sizing parameters λ_{min} and λ_{max} can be decreased for chosen time steps to achieve the effect seen in Fig. 1. The time step size τ must be adjusted accordingly, to ensure stability as described in Section 3.3.

3.5 Triangle Quality Metrics

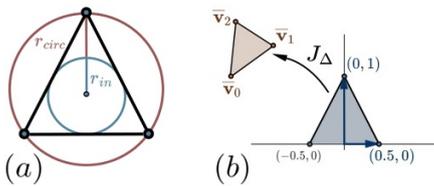


Figure 7: Two quality metrics with preference for equilateral triangles: double inradius over circumradius ratio (a), and Jacobian of the unit equilateral triangle (b) to the planar representation $T = \{v_0, v_1, v_2\}$ in triangle plane: \mathcal{P}_T .

As mentioned in Section 3.3, the semi-implicit formulation of (2) places restrictions on sizes of co-volumes and their uniformity across the surface. The resulting surface F^T can then be used, for example, in mechanical FEM analysis [21]. Most importantly, however, our motivation to ensure that most triangles are as close to equilateral as possible stems from the possibility of fast visualization. The rendering of almost-equilateral triangles can then be accelerated by treating them as discs, i.e.: *fish-scale mesh*⁸ [22]. More information on the quality of linear elements can be found in Section 1.1.2 of [23].

We choose two metrics measuring "equilateralness", namely

1. $2r_{in}/r_{circ} \in [0, 1]$ where r_{in} is the *inradius* and r_{circ} is the *circumradius* of the triangle, implemented in MeshLabTM. For an equilateral triangle $2r_{in} = r_{circ}$.
2. Our customized *condition number* $\kappa(J_{\Delta})$ of the *equilateral triangle Jacobian* $J_{\Delta} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ inspired by [24].

⁸ It is faster to compute disc-ray intersection than triangle-ray intersection.

When expressing the planar vertex position in the respective basis, we normalize the edge lengths with respect to basis vectors, shown in Fig.7, to account for scaling.

3.6 Experimental Setup

We implemented our framework in C++ with the help of the PMP Library⁹ [15] which provides a versatile half-edge (*surface*) mesh representation [26]. As mentioned in Section 3.2, we chose the BiCGSTAB solver with IncompleteLUT preconditioner from the Eigen library¹⁰.

3D visualization of the VTK polydata output is accomplished using ParaView from KitwareTM. Since VTK supports only vertex values, we average metrics from Section 3.5 across triangles adjacent to each vertex.

First, we show the validity of the stabilization heuristic from Section 3.3 by measuring $\mu_{max} = \max_{i=1, \dots, N_V} \mu(V_i)$, $\mu_{min} = \min_{i=1, \dots, N_V} \mu(V_i)$ and the mean $\mu(V) = \frac{1}{N_V} \sum_{i=1}^{N_V} \mu(V_i)$ for all time steps $t \in [0, T]$. We also observe how the use of remeshing reduces the range of values $[\mu_{min}, \mu_{max}]$.

Since our implementation currently supports two types of starting surface — icosphere and isosurface (see Fig.3) — we naturally choose the latter option to shrink-wrap target data with higher genus. To compare the efficiency of shrink-wrapping in other approaches, we select the closest predecessor [7] and the latest work on Repulsive Surfaces by Yu et al. [4]. Since these methods achieve a somewhat "incomplete" wrapping result, we distinguish between two scenarios

- *obstacle*, with $D_2 = 0$,
 - and *full-wrap*, with $D_2 > 0$,
- where D_2 is from the advection control function (5).

We judge the experiments based on two criteria:

1. *Distance-based*: How well the result F^T approximates the target set Γ .
2. *Quality-based*: What is the triangle quality distribution of F^T according to Section 3.5.

The first criterion can be measured by simply sampling d on its domain per vertex as $d|_{F^t}$ using trilinear interpolation, or we can approximate the *Hausdorff distance*:

$$d_H(F^t, \Gamma) := \max \left\{ \sup_{\mathbf{p} \in F^t} d(\mathbf{p}, \Gamma), \sup_{\mathbf{q} \in \Gamma} d(F^t, \mathbf{q}) \right\}, \quad (11)$$

where $d(\mathbf{p}, \Gamma) \approx d|_{F^t}$ and $d(F^t, \mathbf{q})$ is the value of the distance field to surface F^t at a point $\mathbf{q} \in \Gamma$ for all $t \in [0, T]$.

⁹ The library is inspired by the book with the same name [25].

¹⁰ PMP also uses Eigen internally for matrix and vector representations.

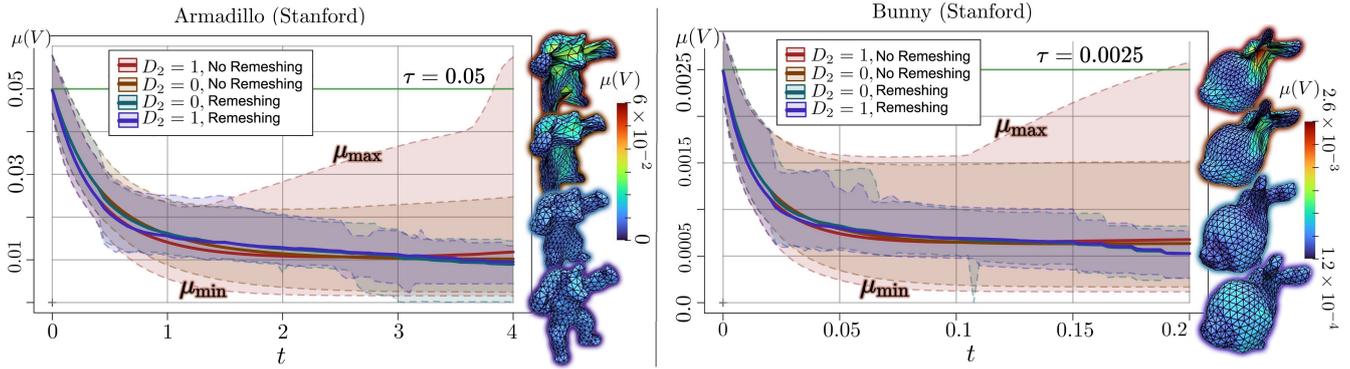


Figure 8: The range of co volume measures $[\mu_{\min}, \mu_{\max}]$ tested for two target meshes (Armadillo and Bunny), for four different settings (of constant D_2 and with or without adaptive remeshing) with resulting meshes $F^{80\tau}$ on the right.

Note that compared to [27] and [28] we choose to utilize trilinear interpolation within already computed distance fields over regular grids.

Furthermore, we divide our experiments into the *comparative* and *demo* categories, with the latter being focused solely on mesh simplification.

It can be observed that evolutionary algorithms (mentioned in Section 2) have a large variety of parameters to fine-tune. We also deem our overall surface extraction algorithm to have too many parameters to cover in this paper. We shall henceforth only mention some of them while, regarding the rest, we refer to our source code on GitHub [29] which will also contain the parameter settings which will yield the comparative, as well as, "most representative" results shown in Section 4.

4 RESULTS

4.1 Stability

Since scaling (6) can be weighed by shrink factor $\sigma > 0$, we first assume $\sigma = 1$, and observe the values of co-volume measures $\mu(V)$ for each vertex during $N_t = 80$ steps. From measures shown, for example, in Fig. 8 we deduce $\mu(V) \approx \tau/5$ most of the time, so we put $\sigma = 1/5$. It is evident that without remeshing, values $\mu(V)$ fluctuate even above the time step size τ .

4.2 LSW for Surface Reconstruction

For the comparative evaluation we choose the Bunny mesh uniformly sampled¹¹ to 12K vertices. Comparing the resulting values in three distinct histogram evaluations for the final time step (see Fig. 9), yields a clear improvement of our method when compared with that of Daniel et al. [7] in terms of both face quality metrics due to remeshing. As mentioned in Section 3.6, it was also important to see how far away the resulting surface

stays from the target point cloud Γ . Clearly, the distribution of distances is smallest for our *full wrap* configuration with the maximum fit. However, in order to improve the effectiveness of feature detection (see Section 3.4) for point cloud targets, we chose value $\tilde{d}_0 = \frac{3\sqrt{3}}{4}c_G$ which is the 1.5-multiple of the voxel diagonal half-length, which is evident in the shift of the histogram of $d|_F$ when compared to the results from [7] and [4] (see Fig. 9).

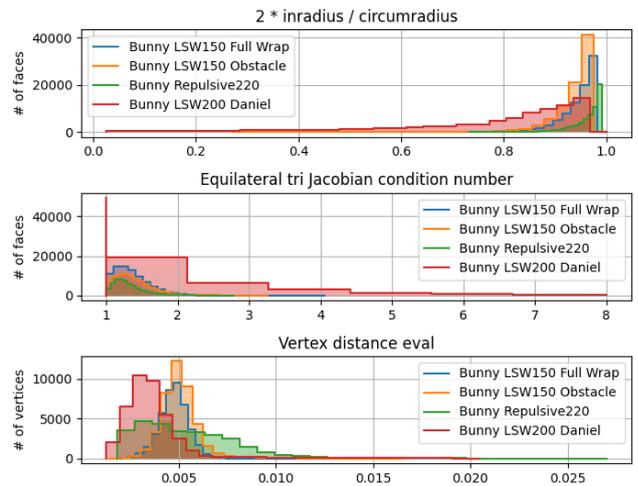


Figure 9: Comparison of four different LSW experiments via histograms of per-face (see Section 3.5) and per-vertex metric $d|_F$ for the last time step.

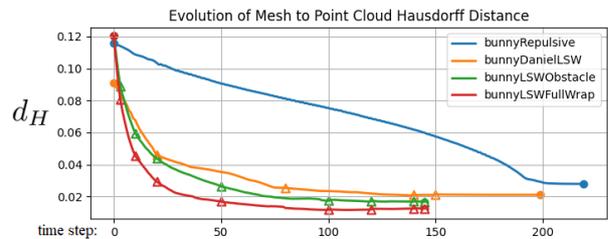


Figure 10: Evaluation of (11) throughout evolution.

¹¹ From the uniform distribution over all mesh vertex indices from the original file.

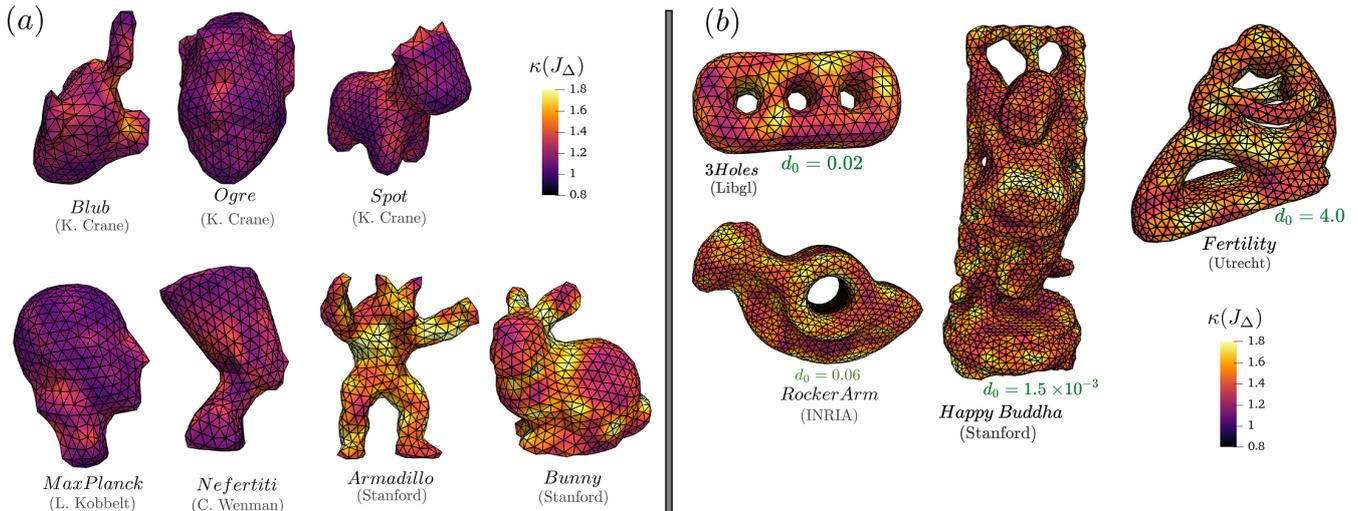


Figure 11: Evaluation of $\kappa(J_\Delta)$ (see Section 3.5) on the shrink-wrapping results from *icosphere* setup (a), and the higher-genus *isosurface* starting surface with given isolevel values $d_0 > 0$.

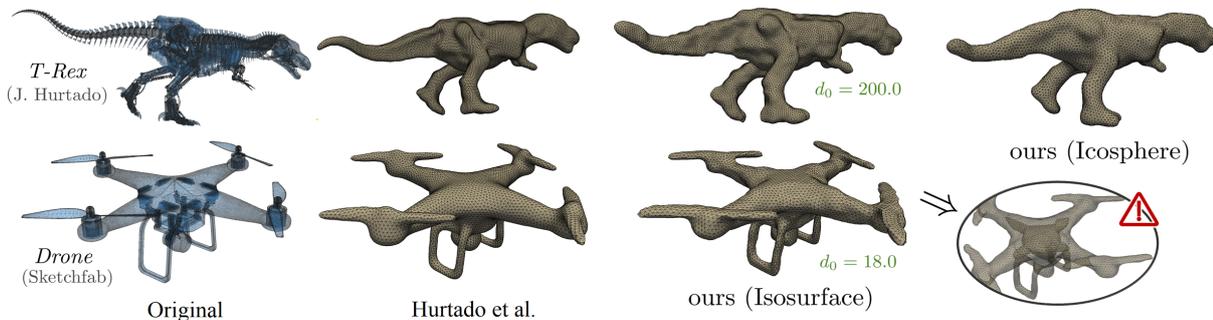


Figure 12: Comparison of our isosurface setup with the "Enveloping CAD" approach by Hurtado et al.[1].

When examining Hausdorff distance (11), we observe fast and stable convergence of F towards the target point cloud Γ in both our Obstacle and FullWrap configurations. Due to the lack of the corrective term with coefficient D_2 in (5), the implementation by Daniel et al. [7] hangs without reaching the surface with significant oscillations. The completely different global method by Yu et al. [4] finds it difficult to converge to the target altogether even after more time steps (see Fig. 10).

We also mark the time steps when mesh sizing changes with symbol Δ . In case of [7] it takes place of a 4:1 triangle subdivision, whereas we use a decay factor of 0.7 for λ_{\min} (see Section 3.4) which is, analogously to [7], applied to time step size τ to ensure stability. In the case of our implementation and that of Yu [4], remeshing takes place whenever necessary for all time steps.

4.3 High-Quality Simplification

Treating our model as an enveloping tool for mesh simplification presents its own benefits and challenges. Most importantly, the simpler topology of the starting

icosphere surface F^0 yields higher simplification ratio considering meshes with internal simplices, that is: those embedded inside the enveloping boundary of the surface, as in Hurtado et al. [1]. On the flip side, the shrink-wrapping of datasets with higher-genus enveloping boundaries leads to self-intersections of F . Without the ability to fix self-intersections by incrementing the surface genus, we are bound to the genus of the starting surface F^0 . That being said, if we construct F^0 from a higher-genus isosurface of the target set distance field d , we are bound to keep internal cavities if the data already has some larger than voxel size $c_G > 0$ (see Fig. 12).

In this section, we also present quality assessment via $\kappa(J_\Delta)$ of the resulting surfaces F^T for a wide variety of datasets as shown in Fig. 11. Even with a few iterations of adaptive remeshing whenever the range of $\kappa(J_\Delta)$ per triangle exceeds our chosen range in our experiments: $[1, 1.5]$ (which slightly exceeds that recommended in [24]), we are able to maintain the quality of almost all faces for all time steps. For meshes with significant features, such as *Armadillo*, *Bunny*, and the higher-genus

isosurface evolution datasets (Fig. 11 (b)) the use of curvature-based feature detection as described in Section 3.4 inhibits adaptive remeshing on vertices marked as feature.

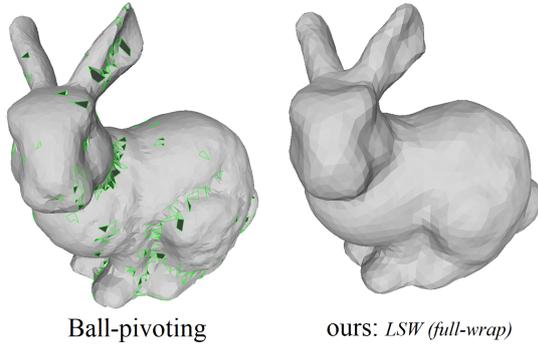


Figure 13: Comparison of point cloud reconstruction results from the Ball-pivoting algorithm and our full-wrap LSW with sizing 0.005. Boundary edges of the non-watertight resulting surface are highlighted in green.

5 DISCUSSION AND FUTURE WORK

Our scale-based heuristic shows stable results with icosphere starting surfaces F^0 (see Section 4.1). However, adjusting parameters is necessary for isosurface F^0 to ensure stability. The LSW method excels in providing the best fit for full-wrap configurations, surpassing other surface evolution techniques in Hausdorff distance evaluation, demonstrating LSW's ability to closely approximate target meshes without sacrificing surface integrity.

When compared to shrink-wrapping techniques like Yu et al. [4], our approach offers competitive face quality and significantly better fit via distance-based evaluation, particularly excelling in the full-wrap configuration (see Section 4.2).

In addition to the results presented in Section 4, we must also qualitatively compare our reconstruction approach to the *Ball-Pivoting Algorithm* (BPA) [30] which, unlike the widely-used *Poisson Reconstruction* [31], does not require normals for the input point cloud (see Fig. 13). Unlike BPA, which can introduce holes, LSW with the full-wrap configuration stands out in producing watertight surface reconstruction without the need for point normals. It should be noted, that unlike BPA, shrink-wrapping approaches merely approximate the original points.

As a mesh simplification tool, our model offers both benefits and challenges. The use of an icosphere F^r as a starting surface simplifies topologies effectively, yet produces self-intersections for higher-genus target data Γ . This stems from the fact that the *comparison principle* (see Huisken [32]) no longer holds for the forced MCF (2). The use of isosurface S_{d_0} with isolevel $d_0 > 0$,

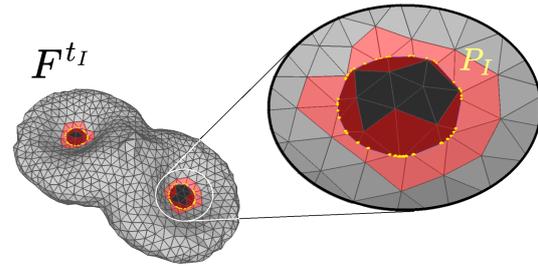


Figure 14: The formation of self-intersection polylines P_t at time $t_t > 0$ when shrink-wrapping a double torus. Faces intersecting at least one other face from surface F^{t_t} are highlighted in red.

on the other hand, does not solve this problem for general target data Γ because it preserves undesired cavities (see Fig. 12).

In our future work we focus on solving the issue of compatibility with higher-genus data by removing surface patches with inverted normals, and properly connecting the remaining faces (see Fig. 14). This will automatically increment the genus of the evolving surface F whenever self-intersections are detected at time $t_t > 0$. This approach is the time-inverse of the surgery technique by Kovács [33], but requires only combinatorial adjustment when the self-intersection polyline P_t is computed.

We shall also examine the *stopping criterion* for LSW to achieve full versatility. Evolution can, for example, terminate when at least 98% of the points no longer move, or change their distance d to Γ . Such criteria need to be formulated for general target data with arbitrary genus and features.

Furthermore, our feature detection techniques are insufficient for very sharp features in Γ (see the *T-Rex* dataset in Fig. 12), and require further inquiry with various projection-based techniques.

When it comes to the applications, we consider using our method for progressively streaming very large mesh files only by passing well-sampled subset of its vertex table and then reconstructing the sample as a point cloud. Feature-sensitive stochastic sampling has been done by Li et al. [34], yet may benefit from our approach which does not require normals in the input data Γ . We propose that this progressive vertex-focused file parsing provides reasonable preview for very large data, often acquired by modern 3D scans.

6 CONCLUSION

In this work, we have introduced a novel approach to automated surface extraction that significantly enhances the efficiency and accuracy of shaping complex objects from simple surfaces. By integrating adaptive remeshing and curvature-based feature detection into the process of Lagrangian Shrink-Wrapping (LSW), we have shown that our method not only maintains high

mesh quality but also ensures proximity to target data across various configurations.

Our findings demonstrate that the scale-based heuristic provides a stable foundation for the evolution of icosphere starting surfaces, albeit requiring nuanced parameter adjustments for other surfaces like isosurfaces. Notably, our method achieves superior surface fitting in "full-wrap" configurations, outperforming existing evolutionary methods when assessed using Hausdorff distance. This precision in approximation without sacrificing surface integrity highlights the potential of our approach for applications requiring accurate and high-quality surface reconstructions.

Moreover, our method presents a competitive alternative to existing surface reconstruction and mesh simplification techniques, such as those proposed by Yu et al. [4], especially in terms of face quality and the preservation of intricate geometric features. The ability of LSW to produce watertight tessellations without the need for normals in the input data, unlike the Ball-Pivoting Algorithm, further underscores its utility for seamless model generation from point cloud data.

As an enveloping tool for mesh simplification, our approach exhibits a comparable simplification ratio for meshes with internal simplices, aligning with the observations of Hurtado et al. [1]. However, we also recognize the challenges posed by datasets with higher-genus enveloping boundaries introducing self-intersections during evolution, which will be the focus of our future work.

Looking forward, we aim to address the limitations encountered with higher-genus data and explore the development of more robust feature detection techniques to capture sharper features accurately. The potential application of our method for progressive streaming of large mesh files presents an exciting avenue for research, promising a significant impact on the field of geometry processing and beyond.

ACKNOWLEDGMENTS

We extend our gratitude to Patrik Daniel and Mariana Sarkoci Remesikova, the authors of [7], for the source code of their application, as well as to Jan J. Hurtado, the co-author of [1], for providing the executable of the article's application (with the T-Rex mesh included). We also express our appreciation to Keenan Crane and Chris Yu, the co-authors of [4], and Robert Bohdal, for their assistance in providing experimental results from the *Repulsive Surfaces* framework.

Special thanks are extended to the creators of the test meshes used in our studies: the Stanford 3D Model Repository (Armadillo, Bunny, Happy Buddha), Keenan Crane (Blub, Ogre, Spot), teams of INRIA, Libgl, Utrecht (RockerArm, 3Holes, Fertility), Leif

Kobbelt and the team at RWTH Aachen (Max Planck), and Cosmo Wenman (Nefertiti), for generously sharing their high-quality models. Additionally, we acknowledge the development teams of the PMP library led by Daniel Seiger and Mario Botsch, and the Eigen Library, for their indispensable tools that facilitated our research.

Our work has also benefited from the use of other datasets not explicitly mentioned earlier (e.g.: Drone from Sketchfab), and the use of ParaView from Kitware™ and MeshLab from ISTI™ for visualization. We recognize the contributions of these and all other data providers who have indirectly supported our research.

Finally, we would like to thank the anonymous reviewers for analyzing our work meticulously.

7 REFERENCES

- [1] Jan Hurtado Jauregui, Anselmo Montenegro, Marcelo Gattass, Felipe Carvalho, and Alberto Raposo. Enveloping cad models for visualization and interaction in xr applications. *Eng. Comput.*, 38(1):781–799, 2022.
- [2] Karol Mikula, Mariana Remešiková, Peter Sarkoci, and Daniel Ševčovič. Manifold evolution with tangential redistribution of points. *SIAM J. Sci. Comput.*, 36:A1384–A1414, 2014.
- [3] Keenan Crane, Ulrich Pinkall, and Peter Schröder. Robust fairing via conformal curvature flow. *ACM Trans. on Graphics (TOG)*, 32(4):1–10, 2013.
- [4] Chris Yu, Caleb Brakensiek, Henrik Schumacher, and Keenan Crane. Repulsive surfaces. *ACM Trans. Graph.*, 40(6), 2021.
- [5] Leonardo Sacht, Etienne Vouga, and Alec Jacobson. Nested cages. *ACM Trans. Graph.*, 34:1–14, 2015.
- [6] Stephen Smith. Fast robust automated brain extraction. *Human brain mapping*, 17:143–55, 2002.
- [7] Patrik Daniel, Matej Medl'a, Karol Mikula, and Mariana Remešiková. Reconstruction of surfaces from point clouds using a lagrangian surface evolution model. page 589–600, 2015.
- [8] Rana Hanocka, Gal Metzer, Raja Giryes, and Daniel Cohen-Or. Point2mesh: A self-prior for deformable meshes. *ACM Trans. Graph.*, 39(4):126:1–126:12, 2020.
- [9] Leif Kobbelt, Jens Vorsatz, Ulf Labsik, and Hans-Peter Seidel. A shrink wrapping approach to remeshing polygonal surfaces. *Comput. Graph. Forum*, 18:119–130, 2000.
- [10] Martin Huska, Matej Medl'a, Karol Mikula, and Serena Morigi. Lagrangian evolution approach

- to surface-patch quadrangulation. *Appl. Math.*, 66:1–43, 2021.
- [11] Martin Čavarga. Advection-driven shrink-wrapping of triangulated surfaces. In *Proc. of the 26th Central European Seminar on Computer Graphics : CESC*, page 95–104, 2022.
- [12] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Mesh optimization. In *Proc. of the 20th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '93*, page 19–26, 1993.
- [13] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998.
- [14] Marion Donyach, David Vanderhaeghe, Loïc Barthe, and Mario Botsch. Adaptive remeshing for real-time mesh deformation. In *Eurographics 2013 - Short Papers*, page 29–32, 2013.
- [15] Daniel Sieger and Mario Botsch. The polygon mesh processing library. <http://www.pmp-library.org>, 2019.
- [16] Hongkai Zhao. A fast sweeping method for eikonal equations. *Math. Comput.*, 74:603–627, 2005.
- [17] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan Barr. Discrete differential-geometry operators for triangulated 2-manifolds. page 37–57, 2003.
- [18] Karol Mikula, Daniel Ševčovič, and Martin Balázovjeh. A simple, fast and stabilized flowing finite volume method for solving general curve evolution equations. *Communications in Computational Physics*, 7:195–211, 2010.
- [19] Michael Kazhdan, Jake Solomon, and Mirela Ben-Chen. Can mean-curvature flow be modified to be non-singular? In *Computer Graphics Forum*, volume 31, pages 1745–1754. Wiley Online Library, 2012.
- [20] Martin Čavarga. Mesh primitive counting formulas for subdivision surfaces. In *Proc. of the 9th Slovak-Czech Conference on Geometry and Graphics 2023*, pages 67–76, 2023.
- [21] Dan Neumayer, Madhukar Chatiri, and Matthias Höermann. Drop test simulation of a cooker including foam packaging and pre-stressed plastic foil wrapping. 2006.
- [22] Radim Sara and Ruzena Bajcsy. Fish-scales: Representing fuzzy manifolds. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 811–817. IEEE, 1998.
- [23] S.W. Cheng, T.K. Dey, and J. Shewchuk. *De-launay Mesh Generation*. Chapman & Hall/CRC Computer and Information Science Series. CRC Press, 2016.
- [24] Sandia National Laboratories. Metrics for triangular elements. https://www.sandia.gov/files/cubit/15.3/help_manual/WebHelp/mesh_generation/mesh_quality_assessment/triangular_metrics.htm, 2017.
- [25] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. *Polygon Mesh Processing*. AK Peters / CRC Press, 2010.
- [26] Daniel Sieger and Mario Botsch. Design, implementation, and evaluation of the surface_mesh data structure. In *Proc. of the 20th International Meshing Roundtable*, pages 533–550, 2012.
- [27] Dejun Zhang, Fazhi He, Soonhung Han, Lu Zou, Yiqi Wu, and Yilin Chen. An efficient approach to directly compute the exact hausdorff distance for 3d point sets. *Integrated Computer-Aided Engineering*, 24:261–277, 2017.
- [28] Michael Bartoň, Iddo Hanniel, Gershon Elber, and Myung-Soo Kim. Precise hausdorff distance computation between polygonal meshes. *CAGD*, 27(8):580–591, 2010.
- [29] Martin Čavarga. Lsw mesh flow. <https://github.com/MCInversion/LSWMeshFlow>, 2022.
- [30] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE trans. on visualization and computer graphics*, 5(4):349–359, 1999.
- [31] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In Alla Sheffer and Konrad Polthier, editors, *Proc. of the 4th Eurographics Symposium on Geometry Processing, Cagliari, Sardinia, Italy, June 26-28, 2006*, volume 256 of *ACM International Conference Proceeding Series*, pages 61–70, 2006.
- [32] Gerhard Huisken. Flow by mean curvature of convex surfaces into spheres. *J. Differential Geom.*, 20:237–266, 1984.
- [33] Balázs Kovács. Numerical surgery for mean curvature flow of surfaces. *SIAM J. Sci. Comput.*, 46(2):A645–A669, 2024.
- [34] Yuanqi Li, Jianwei Guo, Xinran Yang, Shun Liu, Jie Guo, Xiaopeng Zhang, and Yanwen Guo. Deep point cloud simplification for high-quality surface reconstruction. *ArXiv*, abs/2203.09088, 2022.