Discretization-Agnostic Deep Self-Supervised 3D Surface Parameterization

Chandradeep Pokhariya *
chandradeep.pokhariya@research.iiit.ac.in
IIIT Hyderabad
India

Shanthika Naik*
shanthika.naik@research.iiit.ac.in
IIIT Hyderabad
India

Astitva Srivastava
astitva.srivastava@reserarch.iiit.ac.in
IIIT Hyderabad
India

Avinash Sharma
asharma@iiit.ac.in
IIIT Hyderabad
India

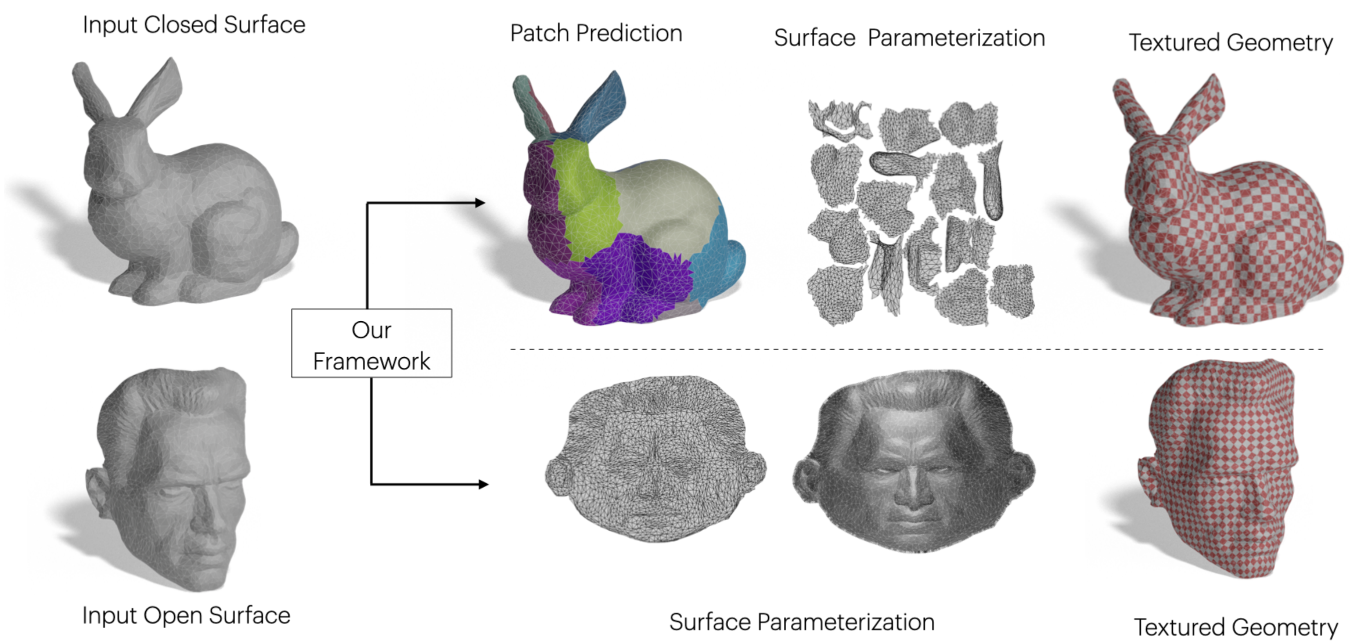

Figure 1: UV parameterization for open and closed surfaces estimated via our proposed framework.

ABSTRACT

We present a novel self-supervised framework for learning the discretization-agnostic surface parameterization of arbitrary 3D objects with both open and closed surfaces. Our framework leverages diffusion-enabled global-to-local shape context for each vertex first to partition the closed surface into multiple patches using the proposed self-supervised PatchNet and subsequently perform independent UV parameterization of these patches by learning forward and backward UV mapping for individual patches. Thus, our framework enables learning a discretization-agnostic parameterization at a lower resolution and then directly inferring the parameterization for a higher-resolution mesh without retraining. We evaluate our framework on multiple 3D objects from the publicly available SHREC [Lian et al. 2011] dataset and report superior/faster UV parameterization over conventional methods.

*Both authors contributed equally to this research.

SA '22 Technical Communications, December 6–9, 2022, Daegu, Republic of Korea

ACM ISBN 978-1-4503-9465-9/22/12…$15.00
https://doi.org/10.1145/3550340.3564235

CCS CONCEPTS

• **Computing methodologies** → **Parametric curve and surface models**; **Neural networks**.

KEYWORDS

UV parameterization, texture mapping, , neural network, self-supervised learning, surface parameterization.

ACM Reference Format:
Chandradeep Pokhariya, Shanthika Naik, Astitva Srivastava, and Avinash Sharma. 2022. Discretization-Agnostic Deep Self-Supervised 3D Surface Parameterization. In *SIGGRAPH Asia 2022 Technical Communications (SA '22 Technical Communications), December 6–9, 2022, Daegu, Republic of Korea.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3550340.3564235

1 INTRODUCTION

Estimating the UV parameterization of arbitrary 3D surfaces lies at the core of computer graphics and geometry processing domain, with a wide range of applications such as 3D modelling, texture-mapping, remeshing, simulation, etc. Formally, it is defined as the projection of vertices of a tessellated surface (polygon mesh) onto a 2D map (UV plane). Determining the aforementioned mapping is not a trivial task and demands a solution with specific properties. The estimated mapping is expected to be isometric, conformal, and non-overlapping. Existing conventional methods [Lévy et al. 2002; Li et al. 2018; Sander et al. 2001; Sawhney and Crane 2017; Wang et al. 2013] aim to estimate an object-centric mapping with an iterative optimization process, focusing on minimizing an energy function explicitly constructed to retain the desired properties. However, they face scalability issues while dealing with high-resolution object meshes and are also prone to local minima.

With the advent of deep learning, researchers are harnessing the power of neural networks to solve various ill-posed problems, offering tractable solutions. Neural surface parameterization has recently been attempted [Aigerman et al. 2022] but under supervised, data-driven settings, requiring a large amount of training data. Such supervised learning solutions get subjected to data bias and hence suffer from poor generalization to unseen, out-of-distribution samples.

This paper presents a novel, self-supervised framework for learning the discretization-agnostic surface parameterization of arbitrary 3D objects with both open and closed surfaces as shown in Figure 1. First, to handle closed surfaces (e.g., a sphere) or surfaces with regions of extreme extrinsic curvature, we propose a learning-based partitioning of the given surface into multiple open patches, which are independently parameterized. To this end, we employ a self-supervised network that assigns each 3D point of the surface to one of the patches, trained using losses based on local features (such as face-normals) and geodesic relationships within the patch.

Subsequently, we propose to learn the surface parameterization of an arbitrary (open) 3D surface to a UV plane using a *Multi-layer Perceptron* (MLP). More specifically, given a *open* 3D surface (patch), we train the forward MLP to predict per-point UV coordinates independently. In order to ensure a meaningful UV mapping, we enforce cycle-consistency loss between the input and reconstructed surface by learning a backward mapping (UV-to-3D) MLP. Additional losses are employed to achieve desired properties of surface parameterization, i.e., isometric, conformal, and area-preserving. A diffusion process [Sharp et al. 2020] over the mesh provides a multi-scale characterization of the underlying surface, entailing a global-to-local context for each vertex. Hence, the DiffusionNet backbone is used for PatchNet, and similarly, respective features are appended while learning surface parameterization to achieve discretization-agnostic UV mapping. A key advantage of learning a discretization-agnostic parameterization is that we can learn on meshes at a lower resolution and then directly infer the parameterization for high resolution meshes without retraining, as shown in Figure 3.

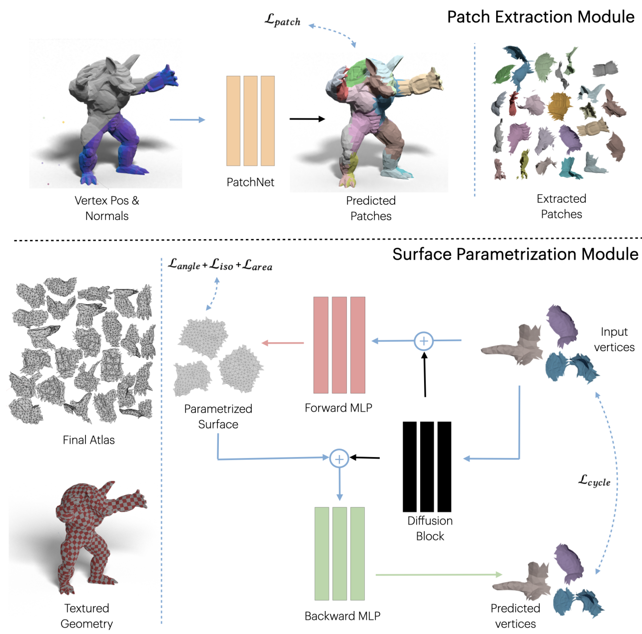

Figure 2: The outline of proposed framework.

2 METHOD

We now describe the proposed framework in detail. The input to our framework is a mesh $\mathcal{M} = \{\mathcal{V}, \mathcal{F}, \mathcal{N}_V\}$, where $\mathcal{V}$, $\mathcal{F}$ and $\mathcal{N}_\mathcal{V}$ are the sets of vertex positions, faces and vertex-normals respectively. Our framework consists of two modules: (i) Patch extraction module and (ii) Surface parameterization module.

2.1 Patch Extraction Module

Handling surfaces with regions of high extrinsic curvature or closed topology requires the 3D manifold to be partitioned into multiple open patches to minimize distortion and overlap. Each patch is defined as $\mathcal{P}_k = \{\mathcal{V}_k, \mathcal{F}_k, \mathcal{N}_{Fk}\}$ ($k = 1, 2, \dots K$), where $\mathcal{V}_k \subseteq \mathcal{V}$ is the set of vertices belonging to $\mathcal{P}_k$. $\mathcal{F}_k \subseteq \mathcal{F}$ is the set of faces defined on $\mathcal{V}_k$ and $\mathcal{N}_{Fk} \subseteq \mathcal{N}_F$ is the associated set of face-normals. We propose PatchNet with parameters ϕ_{patch}, which learns to assign each vertex of $\mathcal{M}$ to one of the K patches, as shown in Figure 2. Here, K is a controllable parameter and can vary based on the acceptable amount of distortion in the input mesh. To learn the parameters ϕ_{patch}, we minimize the following cosine similarity constraint on the estimated patches:

$$\mathcal{L}_{cos} = \sum_{k=1}^{K} \frac{1}{|\mathcal{F}_k|} \left[1 - \left(\sum_{i,j \in \mathcal{F}_k} (\hat{n_i}^T \hat{n_j}) \right) \right]^2 \quad (1)$$

where $i, j \in \mathcal{F}_k$ are the pair of faces with unit normal vectors $\hat{n}_i, \hat{n_j} \in \mathcal{N}_{Fk}$, respectively, and $|\mathcal{F}_k|$ is the number of faces in that patch. The above constraint has the effect of producing locally flat

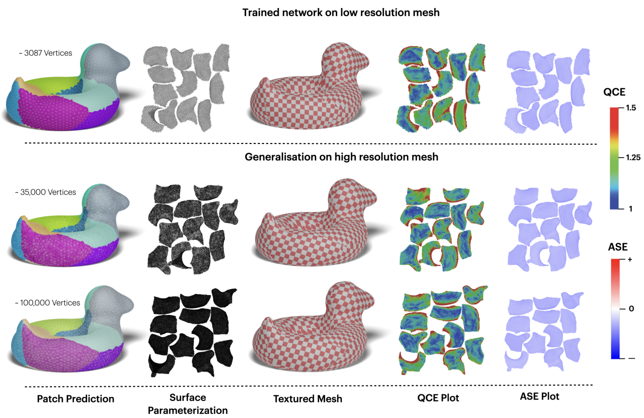

Figure 3: Discretization-agnostic UV parameterization.

patches. However, geodesically far-apart triangles with high cosine similarity may be assigned to the same patch, which is undesirable. To circumvent such disjoint assignments, we minimize the following additional constraint:

$$\mathcal{L}_{geo} = \sum_{k=1}^{K} \frac{1}{|\mathcal{P}_k|} \left(\sum_{i,j \in \mathcal{V}_k} g(i,j) \right) \tag{2}$$

where $g(i,j)$ denotes the geodesic distance between the pair of vertices i & j within the patch and $|\mathcal{P}_k|$ is the number of vertices in that patch. We model PatchNet using DiffusionNet [Sharp et al. 2020] architecture to achieve multi-scale characterization of the underlying surface, entailing a global-to-local context for all the vertices. Input to PatchNet is the vertices $\mathcal{V}$ and vertex-normals $\mathcal{N}_V$, and the output is the predicted assignment probability for all the vertices to each of the K patches. Subsequently, per-face probabilities are obtained by taking the mean probabilities of the corresponding face vertices. We further consolidate the per-face probabilities by taking an average over neighboring faces, and then each face is assigned to the patch with the highest probability. Note that the whole mesh can be considered as a single patch in the case of a open surface with extrinsic curvature of low variability. The combined objective function for patch extraction becomes $\mathcal{L}_{patch} = \lambda_{cos}\mathcal{L}_{cos} + \lambda_{geo}\mathcal{L}_{geo}$.

2.2 Surface Parameterization Module

Each patch $\mathcal{P}_k = \{\mathcal{V}_k, \mathcal{F}_k, \mathcal{N}_k\}$ is treated as a separate open surface and is independently parameterized. Let $f : \mathbb{R}^3 \to \mathbb{R}^2$ be the mapping of each vertex $v \in \mathcal{V}_k$ to a 2D point u on the UV plane. We propose to represent f using a *forward* mapping network MLP_f with learnable parameters ϕ_f. First, the set of vertices $\mathcal{V}_k$ for the given patch is passed to the diffusion block to get a global shape encoding $\psi \in \mathbb{R}^{128}$. Per-vertex input given to MLP_f is $z \in \mathbb{R}^{131}$ (v concatenated with ψ) and the output is $u \in \mathbb{R}^2$ (UV coordinate), i.e. $u = MLP_f(z)$. Since we do not have corresponding ground truth UV coordinates, we resort to a self-supervised cycle-consistency loss. We employ another $MLP_{f^{-1}}$ with learnable parameters $\phi_{f^{-1}}$ to represent the *backward* mapping $f^{-1} : \mathbb{R}^2 \to \mathbb{R}^3$. $MLP_{f^{-1}}$ takes u as input and predicts its corresponding 3D position, which ideally should match with the input vertex position v . We enforce this consistency by minimizing the following cycle loss:

$$\mathcal{L}_{cycle} = \frac{1}{|\mathcal{V}_k|} \sum_{v \in \mathcal{V}_k} \left(v - MLP_{f^{-1}}(u) \right)^2 \tag{3}$$

Note that due to presence of non-linear activation functions in MLP_f and $MLP_{f^{-1}}$, the condition $\phi_f . \phi_{f^{-1}} = I$ need not hold. Per-vertex prediction can be noisy, resulting in an irregular UV space. Conditioning the MLPs with the diffusion-based global shape-encoding ψ regularizes the UV prediction and improves the output of $MLP_{f^{-1}}$. We further add losses to enforce desired properties of surface parameterization, namely, L_{iso} provides isometric behaviour, L_{angle} preserves angles of the faces and L_{area} preserves face-area (neglecting uniform scaling). The final objective function for surface parameterization is given follows:

$$\mathcal{L}_{uv} = \lambda_1 \mathcal{L}_{cycle} + \lambda_2 \mathcal{L}_{iso} + \lambda_3 \mathcal{L}_{angle} + \lambda_4 \mathcal{L}_{area}. \tag{4}$$

Please refer to our supplementary for description of other loss terms.

3 RESULTS & EVALUATION

We compute Quasi-Conformal Error (QCE) and Area Scale Error (ASE) on the final texture atlas for quantitative and qualitative evaluation. *Please refer to the supplementary for their description.*

We provide a **qualitative comparision** of our framework with BFF[Sawhney and Crane 2017] and OptCuts[Li et al. 2018] in Figure 4. As shown, our framework performs on par with these methods on varying geometrical shapes.

In Table 1, we provide a **quantitative comparison** of our framework with BFF[Sawhney and Crane 2017] and OptCuts[Li et al. 2018] on a few classes of SHREC [Lian et al. 2011] dataset using QCE and ASE metric. We train our network on 16 meshes for each mentioned class and compute errors on 4 test sample meshes. Please note that, instead of purely object-centric learning, we compare on a category-specific generalized network, and our performance is comparable to other object-centric methods. Such generalization can be attributed to intrinsic characterization encoded in diffusion features used in our surface parametrization module.

Figure 3 shows the **discretization-agnostic learning** capability of our framework. We train on a mesh with only $\sim 3K$ vertices and directly infer at high resolutions ($\sim 35K$ and $\sim 100K$ vertices). Please note that the error values for high-resolution meshes stay close to the low-resolution mesh, as observed in the error plots.

Table 1: Comparison of QCE and ASE metrics with BFF [Sawhney and Crane 2017] and OptCuts [Li et al. 2018] on SHREC dataset.

	BFF (Linux)		OptCuts		Ours	
Class	QCE↓	|ASE|↓	QCE↓	|ASE|↓	QCE↓	|ASE|↓
Laptop	1.046	2.052	**1.045**	**2.005**	1.196	2.420
Pliers	**1.112**	1.909	1.128	**1.391**	1.274	2.895
Rabbit	**1.132**	2.116	1.160	2.062	1.183	**0.992**
Scissors	1.156	1.456	**1.122**	**1.276**	1.261	2.728
Bird	2.130	**1.103**	**1.129**	1.928	1.262	1.996

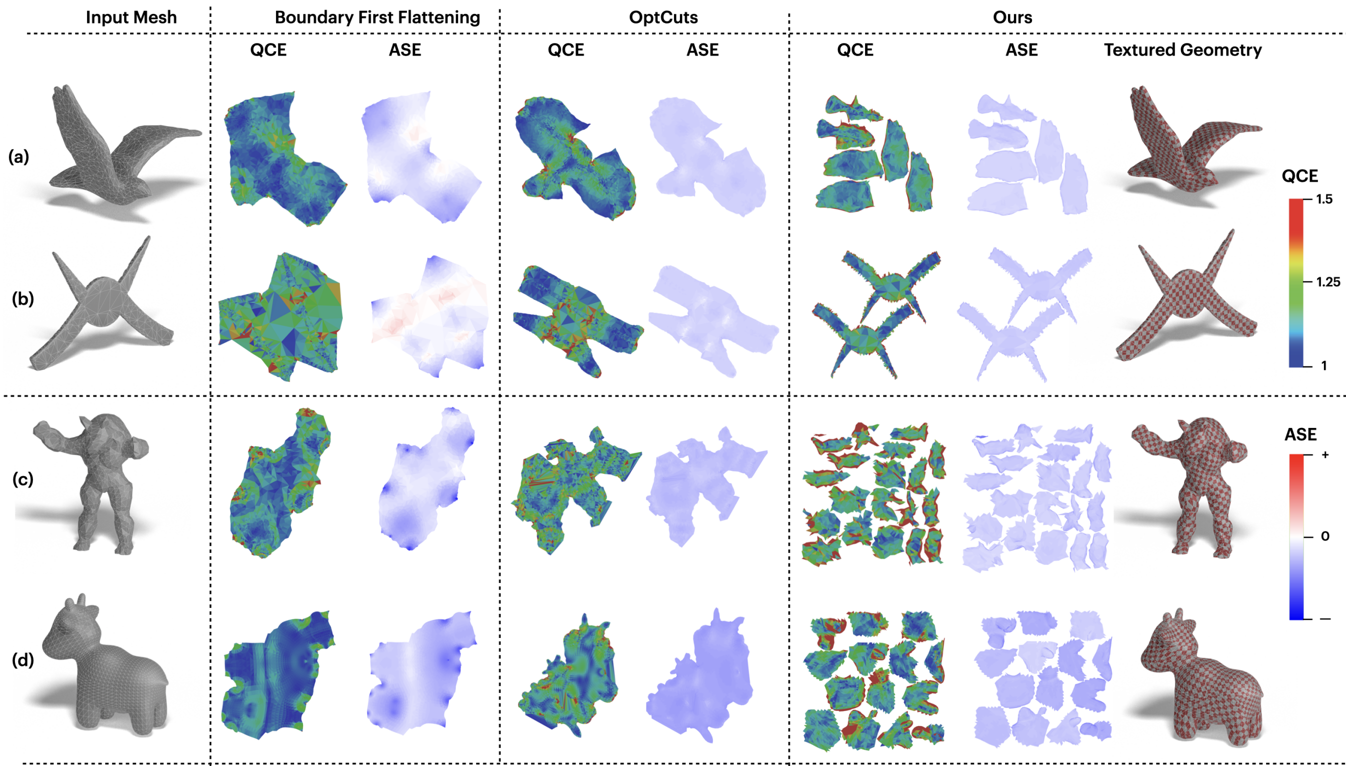

Figure 4: Comparison of error plots for QCE and ASE with other methods. First two categories (a) Bird, (b) Pliers are taken from SHREC dataset; (c) Armadillo & (d) Spot.

Table 2: Comparison of computation time for Stanford's Armadillo.

Resolution	BFF (Linux)	OptCuts	Ours
$30K$	17.41 sec	$>$ 10 min	**2.92 sec**
$100K$	61.04 sec	$>$ 10 min	**5.02 sec**

More importantly, discretization-agnostic learning allows us to reduce the computation time significantly compared to other methods. Specifically, we train our method on the decimated Stanford's Armadillo mesh with $\sim 2K$ vertices and compare our computation time with other two methods at higher resolution ($\sim 30K$ and $\sim 100K$) as shown in Table 2.

Please refer to our supplementary for implementation details and detailed ablative studies.

4 CONCLUSION

We proposed a novel self-supervised learning based framework for surface parameterization of open and closed surfaces. Our framework enables discretization-agnostic learning, significantly improving our inference time performance on high-resolution meshes.

ACKNOWLEDGMENTS

We thank Dhawal Sirikonda for helping us with the visualization of the QCE error metric.

REFERENCES

Noam Aigerman, Kunal Gupta, Vladimir G. Kim, Siddhartha Chaudhuri, Jun Saito, and Thibault Groueix. 2022. Neural Jacobian Fields: Learning Intrinsic Mappings of Arbitrary Meshes.

Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. 2002. Least Squares Conformal Maps for Automatic Texture Atlas Generation. *ACM Trans. Graph.* (2002).

Minchen Li, Danny M. Kaufman, Vladimir G. Kim, Justin Solomon, and Alla Sheffer. 2018. OptCuts: Joint Optimization of Surface Cuts and Parameterization. *ACM Transactions on Graphics* 37, 6 (2018).

Zhouhui Lian, Afzal Godil, Benjamin Bustos, Mohamed Daoudi, Jeroen Hermans, Shun Kawamura, Yukinori Kurita, Guillaume Lavoué, Hien Nguyen, Ryutarou Ohbuchi, Yuki Ohkita, Yuya Ohishi, Fatih Porikli, Martin Reuter, Ivan Sipiran, Dirk Smeets, Paul Suetens, Hedi Tabia, and Dirk Vandermeulen. 2011. SHREC '11 Track: Shape Retrieval on Non-rigid 3D Watertight Meshes. *Eurographics Workshop on 3D Object Retrieval* (01 2011), 79–88.

Pedro V. Sander, John Snyder, Steven J. Gortler, and Hugues Hoppe. 2001. Texture Mapping Progressive Meshes. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01).* Association for Computing Machinery, New York, NY, USA, 409–416.

Rohan Sawhney and Keenan Crane. 2017. Boundary First Flattening. *ACM Trans. Graph.* (2017).

Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. 2020. DiffusionNet: Discretization Agnostic Learning on Surfaces. (2020).

He Wang, Kirill A. Sidorov, Peter Sandilands, and Taku Komura. 2013. Harmonic Parameterization by Electrostatics. *ACM Trans. Graph.* (2013).