SDFace: Boundary-Aware Face Extraction from FM-Generated Implicit 3D Geometries

Hannah Norman (hnorman)

CS468 Final Project Report

Abstract

This project introduces SDFace, a boundary-aware approach to extracting localized surface regions (faces) directly from implicit 3D geometries generated by foundation models, eliminating the need for explicit meshing in finite element analysis (FEA) workflows. Using Signed Distance Functions (SDFs), SDFace detects zero-crossings, clusters surface normals, and refines results with gradient and curvature analysis to extract faces suitable for boundary-specific integrations. Applied to SDFs generated by the Make-a-Shape model, SDFace demonstrates the potential to integrate implicit geometry pipelines with modern FEA workflows for improved efficiency.

1 Background

Finite Element Analysis. Finite Element Analysis (FEA) is a computational technique that is used to approximate solutions to complex physical problems by dividing a domain into smaller, discrete elements. FEA is widely applied in fields such as structural engineering, multiphysics simulations, biomechanics, and aerospace design, where analyzing complex systems under various conditions is critical. The typical FEA workflow involves defining a geometry; discretizing it into a mesh of elements; applying material properties, boundary conditions, and external forces; and solving for variables such as displacements, stresses, or heat transfer. While highly versatile, FEA workflows often rely on explicit meshing, a computationally expensive process that can account for up to 80% of the preparation time.

2 Introduction

Motivation. The integration of foundation models in 3D and 4D scene understanding has transformed how we represent and manipulate geometric data. Recent advancements, such as the Make-a-Shape[2] model, have enabled the generation of compact and flexible implicit representations, like Signed Distance Functions (SDFs), which are well-suited for capturing complex geometries. However, Finite Element Analysis (FEA) relies on localized, surface-specific integrations, but implicit representations lack the tools to extract such surfaces directly, requiring conversion to explicit meshes. This gap raises two key questions:

1. Can we extract localized surface regions (e.g., faces) directly from implicit geometries for boundary-specific integrations in FEA workflows?
2. How can we integrate such a pipeline with modern workflows that leverage foundation model-generated geometric representations?

Related Work. Implicit representations, such as SDFs, have gained prominence as flexible and compact tools for representing complex 3D geometries. Their continuous nature and mathematical simplicity make them ideal for use in applications ranging from generative modeling to physical simulations. Despite these advantages, leveraging implicit representations for localized surface-specific operations remains an open challenge, as there lacks tools for directly extracting distinct regions or features. As far as we are aware, there is no prior work that directly tackles the challenge of extracting localized surface regions from SDFs, especially with the aim of supporting downstream workflows such as Finite Element Analysis (FEA).

Make-a-Shape[2], a recent 3D generative model, demonstrates the potential of diffusion-based methods to produce high-quality voxelized SDFs from multi-view inputs. This approach highlights the power of foundation models in creating detailed and versatile geometric representations but does not address the segmentation or extraction of specific surface regions. Similarly, ClusteringSDF[3] leverages neural implicit surfaces for 3D decomposition, enabling the segmentation of entire scenes into meaningful components. However, its focus is on scene-level reconstruction rather than the precise extraction of localized surface regions from objects. These methods exemplify the strengths of implicit representations while underscoring the knowledge gap that SDFace aims to fill.

Overview. This project addresses these motivating questions and gaps in existing research by introducing SDFace, a pipeline for extracting localized surface regions (faces) directly from implicit geometries represented by SDFs. By eliminating the need for meshing, SDFace streamlines the integration of foundation model-generated geometries into FEA workflows, reducing preparation time and improving efficiency.

Building on course topics such as geometric representations, foundation models for 3D/4D data, and implicit functions in content creation and analysis, this project extends the application of SDFs to boundary-specific operations critical for FEA routines. By addressing these challenges, SDFace not only advances the integration of implicit geometric representations into simulation workflows but also paves the way for faster and more efficient computational pipelines.

3 Method

SDFace is a pipeline designed to extract localized surface regions, or faces, directly from implicit 3D geometries represented as Signed Distance Functions (SDFs). The methodology takes advantage of the compact and continuous properties of voxelized SDFs, generated by the Make-a-Shape foundation model. These representations encode the signed distance of each grid point to the nearest surface, with zero-crossings marking surface boundaries, providing a foundation for accurate and efficient face extraction without relying on explicit meshing.

The SDFace pipeline begins by detecting surface boundaries through zero-crossings in the SDF representation, which form the base representation of the geometry's surface. To facilitate the extraction of localized regions, surface normals are computed at boundary points. This was achieved by manually modifying the internal Make-a-Shape model to output an additional voxelized grid of distance gradient values along each axis, providing the necessary data for accurate normal computation. The computed normals are then grouped into continuous faces using the k-means clustering algorithm. The number of clusters, corresponding to the number of faces, is currently user-defined, offering flexibility but introducing dependency on user input.

To enhance the quality of extracted faces, the initial clustering results are refined through gradient and curvature analysis. Gradient analysis sharpens the boundaries of each face, ensuring accurate segmentation of surfaces with fine geometric detail. Curvature analysis further aids in distinguishing between sharp edges and continuous curved surfaces, addressing challenges such as fragmented extractions in regions of high curvature or noise.

The pipeline proceeds with a structured workflow: raw geometries are sourced from Blender defaults and the ShapeNetCore[1] v2 dataset. Screenshots of each geometry at various angles are taken to create multi-view image collections. These inputs are fed into the Make-a-Shape model to produce high-quality voxelized SDF representations. The SDFs are then processed by the SDFace program to segment faces on the geometry and produce a labeled surface mask. Finally, the segmented SDFs are converted into explicit meshes using the Marching Cubes algorithm, allowing for visualization in standard OBJ format.

4 Experiments

To evaluate the performance of SDFace, the pipeline was tested on both simple and complex geometries. The input geometries include basic primitives, specifically a cube, cylinder, and sphere, as well as more complex, real-world shapes, including a bottle and table. Notably, the complex geometries can be understood as combinations of simpler components, providing a natural extension for assessing the pipeline's capabilities. Figures 1 and 2 showcase the experimental results.

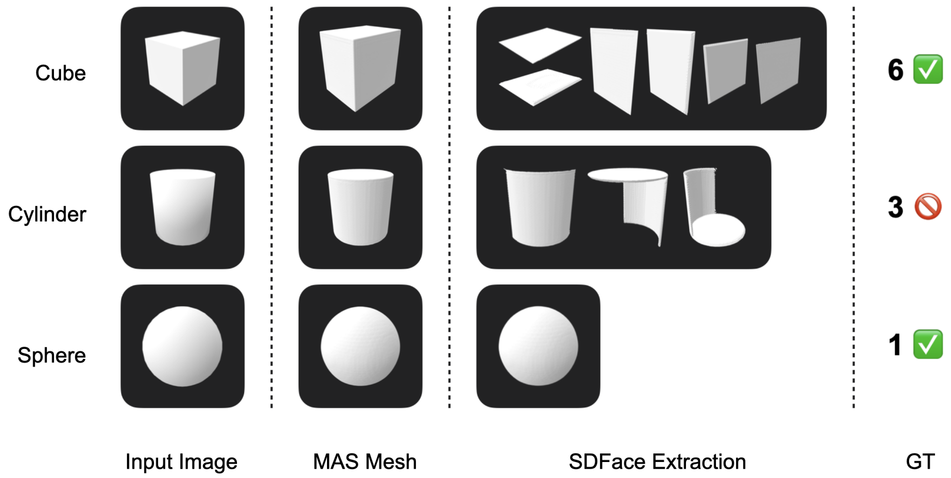

Figure 1: SDFace extraction results with basic geometries.

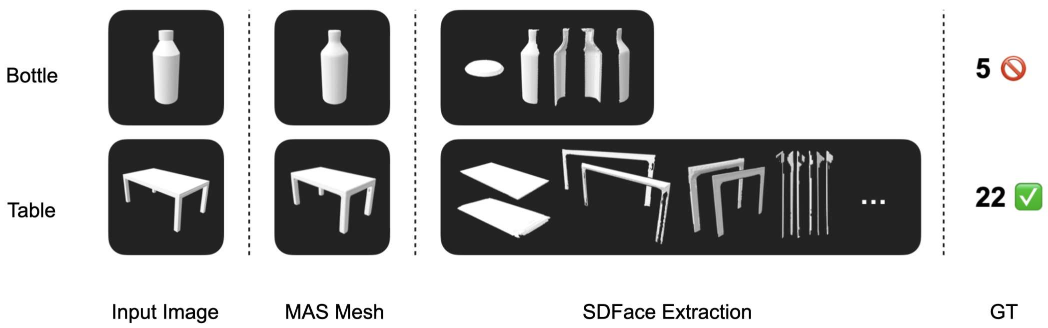

Figure 2: SDFace extraction results with complex geometries.

For simple geometries with piecewise planar surfaces, SDFace demonstrated precise boundary detection and segmentation. For instance, the algorithm accurately extracted all six planar faces of a cube, maintaining sharp, well-defined boundaries. The cylinder's flat end caps were also segmented moderately well, though they were grouped with portions of the curved lateral surface. Despite its continuous curvature, the sphere did not pose significant challenges. This outcome is largely attributed to the user-defined number of clusters, which aligned with the ground truth face count, allowing the algorithm to segment the surface cleanly, as expected.

When applied to complex geometries, the pipeline demonstrated scalability and some generalization capabilities. The bottle, composed of a cylindrical body and circular cap and base, produced results similar to those of the basic cylinder. While the base was accurately segmented, the body and cap were divided into four vertical segments instead of horizontal regions that better represent their continuous curvature. For the table geometry, the pipeline performed well overall, effectively identifying and segmenting the planar tabletop and the four rectangular legs. However, the table, being the noisiest input geometry in the dataset, exhibited several artifacts, particularly among smaller regions such as the faces on a leg. These results highlight the need for further refinement to improve segmentation accuracy in noisy or irregular inputs.

5 Conclusions

This project introduces SDFace, a boundary-aware pipeline for extracting localized surface regions directly from implicit 3D geometries represented as Signed Distance Functions (SDFs). By bypassing the computationally expensive meshing process, SDFace provides a streamlined and efficient approach for integrating implicit representations into workflows like Finite Element Analysis (FEA). The pipeline successfully demonstrates its ability to handle a range of geometries, from simple primitives to complex, real-world objects, validating its applicability across different levels of geometric complexity.

Limitations. Despite its strengths, the pipeline has several limitations. The reliance on a user-defined parameter—the number of clusters for k-means clustering—introduces a degree of manual intervention that may limit full automation. Additionally, the handling of high-curvature regions and transitions between distinct surface areas can lead to artifacts or over-segmentation, limiting the precision of extracted faces. These challenges highlight the need for more adaptive and robust clustering techniques. The current approach also assumes relatively clean and noise-free input SDFs, which may not generalize well to noisy or incomplete data that may be generated by 3d foundation models, like Make-a-Shape.

Future Work. Adaptive clustering methods that dynamically determine the number of faces could reduce user dependency and improve segmentation accuracy. Integrating machine learning models trained on diverse geometries offers a promising direction to enhance generalization, automation, and robustness. Refinement techniques, such as advanced gradient and curvature analysis, could better handle complex surfaces and reduce artifacts. Expanding the pipeline to support noisy or incomplete SDFs would further broaden its applicability to real-world tasks.

By automating surface extraction, the SDFace pipeline highlights its potential to streamline workflows and seamlessly integrate with modern 3D foundation models. Despite remaining challenges, this work establishes a strong foundation for future advancements in the field.

References

[1] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[2] Ka-Hei Hui, Aditya Sanghi, Arianna Rampini, Kamal Rahimi Malekshan, Zhengzhe Liu, Hooman Shayani, and Chi-Wing Fu. Make-a-shape: a ten-million-scale 3d shape model. *arXiv preprint*, arXiv:2401.11067, 2024.

[3] Tianhao Wu, Chuanxia Zheng, Tat-Jen Cham, and Qianyi Wu. Clusteringsdf: Self-organized neural implicit surfaces for 3d decomposition. *arXiv preprint*, arXiv:2403.14619, 2024.

Appendix

A Github Repository

Project page: `https://github.com/hnorm0629/cs468-final-project`.