FEATURE ARTICLE

MidSurfer: A Parameter-Free Approach for Mid-Surface Extraction from Segmented Volumetric Data

Eva Boneš[*,1,2], Dawar Khan[*,1,3], Ciril Bohak[2], Benjamin A. Barad[4], Danielle A. Grotjahn[5], Ivan Viola[1], Thomas Theußl[1,6],
[1]*King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia*
[2]*University of Ljubljana, Faculty of Computer and Information Science, Slovenia*
[3] *Department of Information Technology, The University of Haripur, Haripur, Pakistan*
[4]*Oregon Health & Science University, Department of Chemical Physiology and Biochemistry, Portland, OR, USA*
[5]*The Scripps Research Institute, Department of Integrative Structural & Computational Biology, La Jolla, CA, USA*
[6]*Consivi KG, Deutschlandsberg, Austria*
[*]*E. Boneš and D. Khan are Joint first authors with equal contributions.*

Abstract—This paper presents MidSurfer, a novel parameter-free method for extracting mid-surfaces from segmented volumetric data. The method generates uniformly triangulated, smooth meshes that accurately capture structural features. The process begins with the Ridge Field Transformation step that transforms the segmented input data, followed by the Mid-Polyline Extraction Algorithm that works on individual volume slices. Based on the connectivity of components, this step can result in either single or multiple polyline segments that represent the structural features. These segments form a coherent series, creating a backbone of regularly spaced points representing the mid-surface. Subsequently, we employ a Polyline Zipper Algorithm for triangulation that connects these polyline segments across neighboring slices, yielding a detailed triangulated mid-surface mesh. Results show that this method outperforms previous techniques in versatility, simplicity, and accuracy. Our approach is publicly available as a ParaView plugin at https://github.com/kaust-vislab/MidSurfer.

Surface analysis and visualization play an essential role in the exploration of biological structures, aiding in the understanding of complex biological phenomena. In particular, the study of thinly bounded compartments, such as cells and their organelles bounded with membranes, requires precise surface representation for accurate analysis since membranes serve as interfaces for various cellular components, such as cell membranes, protein-protein interfaces, organelle junctions, and cell-extracellular matrix contacts. Membranes govern molecular exchange, cellular communication, and intracellular transport, thereby shaping cellular function and organization. These structures, often explored through volumetric microscopy data (obtained using cryogenic electron tomography (cryo-ET), focused ion beam scanning electron microscopy (FIB-SEM), or some other acquisition method), contain rich details critical for various biological research and applications. However, the extraction and analysis of surfaces from such data pose significant challenges due to the intricate nature of biological structures and the limitations of existing approaches.

In biological research, precise surface representation is critical for analyzing and modeling cellular compartments, necessitating new surface extraction methods. Traditional approaches often fall short in capturing the intricate details required for thorough biological analyses, particularly from volumetric microscopy data. The mid-surface concept, representing the median layer of an object and providing an accurate model for thinly bounded structures, emerges as a promising solution. However, the literature indicates a significant gap: a standardized, parameter-free method for mid-surface extraction is notably absent, with existing methods requiring manual parameter adjustments and often failing to maintain topological integrity. This gap underscores the need for an innovative approach that combines accuracy with usability in extracting mid-surfaces.

XXXX-XXX © 2021 IEEE
Digital Object Identifier 10.1109/XXX.0000.0000000

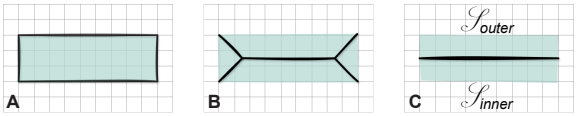

FIGURE 1. Comparative illustration of (A) isosurfaces, (B) medial surfaces, and (C) mid-surfaces, emphasizing mid-surfaces' unique advantages in capturing median geometry. Segmentations are shown in light green and surfaces with thick black lines.

We define the mid-surface ($\mathscr{S}_m$) as a two-dimensional manifold embedded within a three-dimensional structure, equidistant from its inner and outer surfaces (see Fig. 1). These surfaces represent the bounding layers of the structure, where $\mathscr{S}_{\text{inner}}$ denotes the interior boundary and $\mathscr{S}_{\text{outer}}$ denotes the exterior boundary. Mathematically,

$$\mathscr{S}_m = \left\{ \mathbf{p} \in \mathbb{R}^3 \mid \mathrm{d}(\mathbf{p}, \mathscr{S}_{\text{inner}}) = \mathrm{d}(\mathbf{p}, \mathscr{S}_{\text{outer}}) \right\}. \quad (1)$$

In our case, $\mathscr{S}_{\text{inner}}$ and $\mathscr{S}_{\text{outer}}$ denote the inner and outer surfaces bounding the segmented volume. For example, in Fig. 1, $\mathscr{S}_{\text{inner}}$ corresponds to the lower boundary of a rectangle, while $\mathscr{S}_{\text{outer}}$ corresponds to its upper boundary. The mid-surface (C) is represented by a horizontal line equidistant from these boundaries, distinguishing it from *medial surfaces* [1] (B) and *isosurfaces* [2] (A).

In biological research, there is a strong need for parameter-free, easy-to-use algorithms that generate high-quality meshes readily usable in downstream applications. In this paper, we focus on extracting *mid-surface*, in the form of a regular triangular mesh. Our approach adopts a simple, slice-wise strategy that requires no user-defined parameters and produces uniformly sampled meshes. The main objective is to develop a robust, parameter-free method to reconstruct mid-surfaces as high-quality, regular triangular meshes from binary segmentations, particularly in contexts such as cryo-electron tomography, where data are naturally organized as 2D slices. The method integrates smoothing of the Signed Distance Field (SDF), ridge extraction via Hessian analysis in 2D slices, and a novel polyline zippering algorithm for surface reconstruction.

Conceptually, this mid-surface corresponds to the ridge of the SDF derived from the segmented volume. While ridge extraction [3] is a well-studied problem, our approach focuses on extracting this ridge in a way that guarantees regular, uniformly triangulated meshes without requiring parameter tuning, which is important for downstream modeling and analysis. The key contributions (illustrated in Fig. 2) include:

- Introducing a novel, parameter-free, and robust method for extracting the mid-surface from segmented volumetric data.
- Representing the mid-surface as a high-quality triangular surface mesh, while considering standard mesh quality metrics.
- Evaluating the method across multiple datasets, demonstrating significant improvements over existing alternatives. Additionally, showcasing its efficacy in two structural biology use cases, including quantitative analysis with surface morphometrics and mid-surface modeling.
- Making our algorithm publicly available as a ParaView plugin (paraview.org), providing accessible functionality to a wide range of users.

Related Work

In 3D modeling, surfaces serve as foundational structures across various applications, facilitating the attainment of desired final shapes. These surfaces may be manually crafted or algorithmically extracted from reference data, encompassing a wide array of geometric representations, including meshes, point clouds, and volumetric data. While numerous approaches for surface extraction exist, their applicability often extends beyond their primary input data type due to format interconvertibility.

Volumetric data presents unique challenges due to its continuous representation of structures. Unlike point clouds, where the goal is often to reconstruct discrete surfaces, volumetric data requires an initial determination of what constitutes a surface within the volume. As illustrated in Fig. 1, surfaces typically fall into three categories: (1) *isosurfaces* [2], delineating uniform value boundaries; (2) *medial surfaces* [1], representing structural skeletons; and (3) *mid-surfaces*, situated equidistantly between two boundary surfaces. Our research primarily addresses mid-surfaces, though we acknowledge contributions to other categories.

Isosurfaces, while useful for visualizing volumetric data, face limitations in accurately representing complex structures with multiple boundaries. The popular Marching Cubes algorithm suffers from discretization errors and smoothing artifacts, making it unsuitable for mid-surface extraction. Medial surfaces, often depicted through medial meshes, encapsulate core geometric features but prove inadequate for our complex structures due to their typically branched skeletons, failing to provide a practical foundation for mesoscale modeling.

In addition to isosurface and medial-surface approaches, distance/level-set–based medial-axis and centerline methods extract *one-dimensional* curve skeletons (e.g., Wischgoll *et al.* [4]; Telea and Vilanova [5]) for tubular lumina and vascular morphometry. While centerline techniques [4], [5] could, in principle, be adapted toward mid-surface reconstruction, moving from *1D* skeleton extraction to a *two-manifold* surface would require substantial redesign. Such

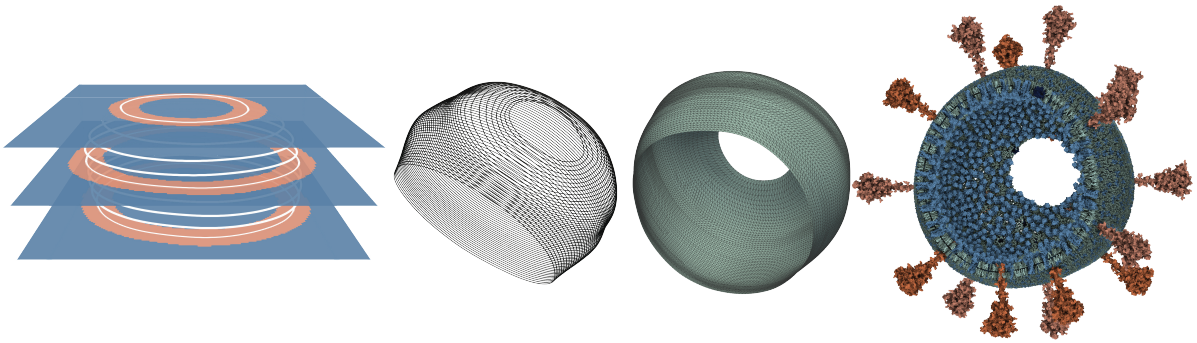

FIGURE 2. An overview of the presented method. From left to right: Volumetric segmented data represented in slice-wise view, the extracted mid-polylines, the triangular surface generated from mid-polylines, and a use case of the mid-surface with modeling results visualized over the extracted mesh.

pipelines typically yield point/voxel skeletons that then need nontrivial, parameterized surface reconstruction; direct triangulation of these point sets often produces irregular meshes unless followed by significant remeshing [6]. By contrast, we use related distance/Hessian cues to recover a *two-manifold mid-surface*—the locus equidistant to a single bilayer's inner and outer leaflets—returned as a uniformly triangulated surface mesh suited to visualization and morphometric surface analysis [7].

Concretely: (1) Wischgoll *et al.* [4] segment vessel walls and derive centerlines for radius/length/branching, i.e., a curve-skeleton output; (2) Telea and Vilanova [5] present a level-set algorithm for robust centerline extraction from volumetric datasets. Both target 1D skeletons and therefore address a different problem than ours. In the same structural-biology context, Barad *et al.* [7] reconstruct membrane *surfaces* from voxel segmentations (e.g., screened Poisson) to enable morphometrics; this pipeline involves parameter choices and mesh quality is also poor. By contrast, MidSurfer directly estimates the mid-surface and outputs a uniformly triangulated two-manifold suitable for quantitative analysis.

Mid-surface representations [8], which conceptually relate to ridge structures in scalar fields [3], are widely applied in engineering, finite element analysis, and structural biology to capture the median layer between boundaries. This trait renders mid-surfaces particularly adept for exploring thinly bounded structures. Barad *et al.*'s approach [7] employs a mid-surface morphometrics pipeline with meshing techniques to quantify various metrics. Their methodology integrates several techniques, including semi-automated segmentation and screened Poisson Surface Reconstruction (PSR) for mesh generation. While PSR effectively transforms membrane voxel segmentation into implicit surface meshes, there's a risk of geometrical fidelity loss unless input parameters are meticulously chosen. Their investigations highlight the demand for more automated, parameter-free methods for mid-surface extraction—our research's primary aim.

General ridge extraction techniques in scalar fields [3] provide a theoretical basis for mid-surface estimation, but they often assume different input data and do not readily produce uniform, high-quality triangular meshes. MemSurfer [9], designed for characterizing curved membranes from molecular simulations, further emphasizes the necessity for accurate mid-surface modeling. Their approach allows for precise calculations of membrane properties such as area per lipid, lipid density, bilayer thickness, and curvature. These calculations represent potential use cases for our approach, which aims to provide more accurate mid-surface representation without extensive parameter tuning.

Overall, these methods are limited by their dependence on user-defined parameters, inconsistent mesh quality, and complex pipelines. We therefore propose a simple, parameter-free approach that generates high-quality mid-surface meshes directly from segmentation data.

Method

Our method processes binary volumetric segmentations to extract mid-surfaces (see Eq. (1)) from segmented structures. The surface is ultimately reconstructed as a regular triangular mesh derived from *mid-polylines*, where an individual mid-polyline is a series of straight line segments lying midway between the inner and outer boundaries of a segmentation. This principle forms the basis of the Mid-Polyline Extraction Algorithm.

Method Overview

Figs. 2 and 3 illustrate the overall pipeline, which consists of three key modules detailed in subsequent subsections. The primary goal is to extract a mid-surface from volumetric

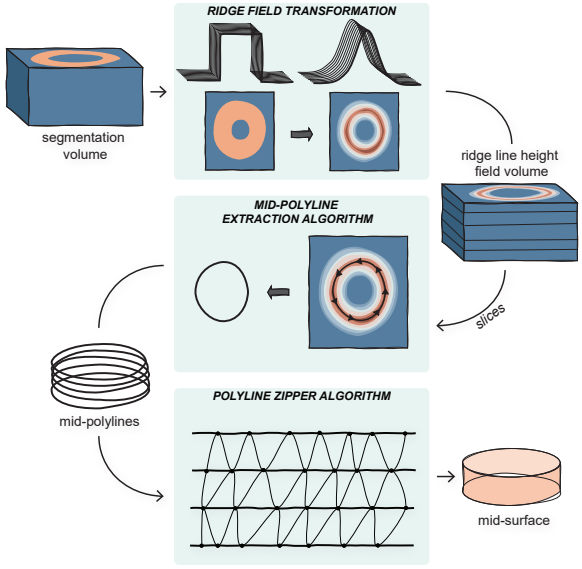

FIGURE 3. Schematic overview of the Mid-Surface Extraction Algorithm. The process takes volumetric segmented data as input and produces a regular triangular mesh of the mid-surface, which can then be used for modeling or other applications (see Fig. 2). The workflow consists of three main steps: Ridge Field Transformation (which converts binary segmentation data into a continuous representation), the Mid-Polyline Extraction Algorithm (which extracts mid-polylines from slices), and the Polyline Zipper Algorithm (generating the final mesh).

data and represent it as a triangular surface mesh. This mesh is constructed by triangulating horizontally aligned mid-polylines derived from segmented volumetric data.

The volumetric data is analyzed slice-by-slice, with each slice yielding mid-polylines at its specific level. Continuous segmentations within a slice produce a single seamless mid-polyline, while discontinuities lead to fragmented mid-polyline segments.

To derive a mid-polyline from a slice, we transform the binary segmentation into a *ridge line height field*, emphasizing symmetry for accurate delineation. Starting from the highest point, the method traces along the ridge using Euler integration and golden section search, continuing until the mid-polyline forms a loop or exits the component's boundary. The algorithm handles multiple connected components within a slice. Mid-polylines correspond to the ridges of the smoothed SDF across slices. These lines from all slices are then connected into a triangular mesh using the Polyline Zipper Algorithm, which systematically connects vertices from adjacent slices to form a high-quality, uniformly triangulated surface mesh representing the mid-surface of the original volumetric data.

Ridge Field Transformation

To successfully extract the mid-polyline in subsequent stages, a smooth height field featuring a central ridge, termed the ridge line height field, is required. This transformation must be symmetric to ensure the ridge is precisely positioned midway between segmentation boundaries and covers the entire segmented area for accurate extraction.

Naively, a method that meets these requirements could involve applying a convolution operator to perform a weighted average of surrounding pixels. This is often achieved through Gaussian smoothing, defined as:

$$G(x,y,z) = \frac{1}{(2\pi\sigma^2)^{3/2}} e^{-\frac{x^2+y^2+z^2}{2\sigma^2}}, \tag{2}$$

where $G(x,y,z)$ represents the Gaussian kernel, σ denotes the standard deviation (or kernel radius) of the Gaussian distribution, and (x,y,z) indicates kernel position. Choosing proper parameters is critical for achieving accurate results. The optimal parameters depend on the width of the segmentation: wider segmentations require larger smoothing values to ensure smooth transitions across the area without leaving constant patches, while narrower segmentations demand smaller values to avoid overly diminishing features within the resulting height field.

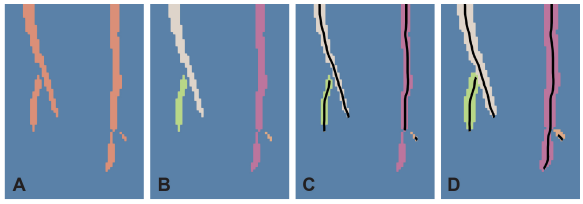

FIGURE 4. Visualization of the connected components identification and mid-polyline tracing on a binary mask slice. (A) The original binary mask. (B) The connected components within the binary mask, each assigned a unique ID. (C) Tracing the mid-polyline within these identified connected components. (D) Tracing the mid-polyline on the dilated connected components, avoiding ending the line prematurely around narrow boundaries.

While Gaussian smoothing provides a simple solution, manually setting parameters across varying segmentation widths introduces challenges. Drawing on the understanding that the transformation should depend on segmentation width and meet requirements for symmetry and a smooth ridge line height field, we utilize a SDF. The SDF assigns distance values to each pixel/voxel $\mathbf{x}$ relative to the nearest boundary:

$$SDF(\mathbf{x}) = sign(\mathbf{x}) \cdot d(\mathbf{x}, \mathscr{S}), \tag{3}$$

where $\mathscr{S}$ presents the surface, $d(\mathbf{x}, \mathscr{S}) = \min_{\mathbf{y}\in\mathscr{S}} \|\mathbf{x}-\mathbf{y}\|$ is the minimal distance from $\mathbf{x}$ to surface $\mathscr{S}$ and $sign(\mathbf{x})$ is

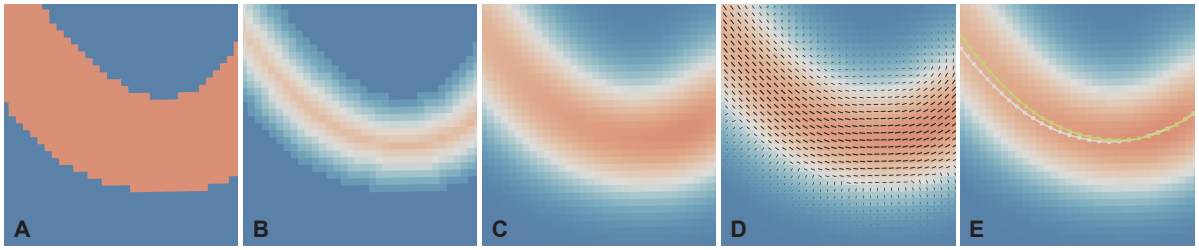

FIGURE 5. Visualization of Ridge Field Transformation and Mid-Polyline Extraction Algorithm on a slice: (A) Initial binary segmentation, showcasing the structure of interest. (B) Signed distance field derived from the initial segmentation. (C) Ridge line height field (a smoothed version of the signed distance field to reduce discretization artifacts and create a stable gradient field for ridge extraction). (D) Visualization of normalized eigenvectors associated with the smallest eigenvalue, represented as a line field. (E) Golden section search optimization: a white polyline depicts a streamline within an eigenvector field drifting from the ridge, while a green line shows the golden section search perpendicular to the eigenvector, ensuring the streamline's alignment with the ridge.

the sign function determining whether the point $\mathbf{x}$ is inside or outside the surface that can be defined as

$$sign(\mathbf{x}) = \begin{cases} +1 & \text{if } \mathbf{x} \text{ is outside } \mathscr{S}, \\ 0 & \text{if } \mathbf{x} \in \mathscr{S}, \\ -1 & \text{if } \mathbf{x} \text{ is inside } \mathscr{S} \end{cases} . \tag{4}$$

In our case of binary segmentation, each pixel or voxel $\mathbf{x}$ inside the segmented mask is assigned the shortest distance to the nearest boundary layer of surface $\mathscr{S}$. This formulation ensures that each point's SDF value encodes its relative position between the two layers, with points lying exactly midway between the boundaries having the highest magnitude. This produces a ridge line height field irrespective of segmentation thickness.

While the SDF accurately identifies segmentation centers, resolution limitations can introduce staircasing artifacts. To counteract this issue while keeping the approach parameter-free and dataset-adaptive, we apply Gaussian smoothing with parameters derived from the SDF itself. Specifically – the standard deviation is set as half of the maximum SDF value ($\sigma = \frac{\text{SDF}_{\max}}{2}$ and the kernel size is computed to fit the Gaussian curve ($radius = 2\sigma + 1$). The adaptive Gaussian kernel used for smoothing is defined as:

$$G_s(x,y,z) = \frac{1}{(2\pi\left(\frac{\text{SDF}_{\max}}{2}\right)^2)^{3/2}} e^{-\frac{x^2+y^2+z^2}{2\left(\frac{\text{SDF}_{\max}}{2}\right)^2}}, \tag{5}$$

and the smoothed SDF is obtained by convolving ($*$) the original field with this kernel:

$$SDF_{\text{smoothed}}(x,y,z) = SDF(x,y,z) * G_s(x,y,z). \tag{6}$$

This process ensures a uniform ridge line height field across slices by performing calculations in 3D rather than slice-by-slice. Both SDF computation and smoothing are applied volumetrically to eliminate staircasing artifacts and maintain smooth transitions between slices.

The slicing direction aligns with the microscopy data orientation based on sample preparation. This ensures slices conform to structures being extracted while leveraging higher resolution in slice planes due to anisotropic resolution caused by cryoET's missing wedge problem [10]. Briefly, this phenomenon arises from limited tilt range during acquisition, resulting in incomplete Fourier space sampling and lower resolution along the z-axis compared to the x and y axes. Although smoothing is performed volumetrically, ridge extraction is applied independently in each 2D slice, reducing the impact of z-blurring on the resulting mid-polylines.

Mid-Polyline Extraction Algorithm

Given a stack of slices with smooth height fields, our process begins by computing a line field in each slice. This involves calculating the Hessian matrix $\mathbf{H}$:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}, \tag{7}$$

where f denotes the smoothed signed distance field restricted to a single 2D slice at a fixed z-plane, and (x,y) are 2D coordinates within that slice, i.e.,

$$f(x,y) = SDF_{\text{smoothed}}(x,y,z_i). \tag{8}$$

The elements of $\mathbf{H}$ are second-order partial derivatives that measure changes in the gradient of f with respect to x and y. These derivatives form the basis for identifying curvature directions.

From these matrices, we identify and extract the eigen-

vector characterized by:

$$\mathbf{H}\mathbf{v} = \lambda \mathbf{v}, \tag{9}$$

where λ represents an eigenvalue, and $\mathbf{v}$ is its corresponding eigenvector. Specifically, we focus on the smallest eigenvalue, $\lambda_{\min}$, and its associated eigenvector, $\mathbf{v}_{\min}$. Here, $\lambda_{\min}$ denotes the eigenvalue with the smallest numerical value (the most negative), which typically corresponds to the direction of principal curvature along the ridge. This eigenvector indicates the direction of minimal curvature within the slice, effectively highlighting the path of least geometric variation. Tracing this direction from any given point on a ridge line ensures adherence to the ridge, thereby maintaining a consistent path. The elements of the Hessian matrix $\mathbf{H}$ are determined by second-order partial derivatives of the height field function f, providing a mathematical framework for identifying curvature directions. The eigenvalues and eigenvectors of $\mathbf{H}$ are computed using Jacobi iteration [11, (Chapter 11.1)].

A ridge aligns with the mid-polyline and delineates the trajectory of the mid-surface. Navigating this ridge through a process informed by $\mathbf{v}_{\min}$ ensures precise alignment of the mid-polyline with the segmented structure's inherent geometry. This principle is foundational for subsequent steps aimed at deriving the mid-surface from an aggregation of mid-polylines.

Upon establishing a vector field of minimal curvature in each slice, the next step involves tracing the mid-polyline. A single slice may feature multiple *hill ranges*, necessitating the identification of connected components within the segmentation (Fig. 4). For each component, tracing begins at the peak of the smoothed height field – the pixel or voxel with the maximal value.

Starting from the initial point $\mathbf{r}_i$, we compute the next point $\mathbf{r}_{i+1} = (x_{i+1}, y_{i+1})$ along the mid-polyline using Euler integration:

$$\mathbf{r}_{i+1} = \mathbf{r}_i + h \cdot \mathbf{v}_i, \tag{10}$$

where $h = \sqrt{2} \times \text{spacing}$ defines the integration step size, and $\mathbf{v}_i = (x_{v_i}, y_{v_i})$ represents the direction at $\mathbf{r}_i$, corresponding to the eigenvector with the minimal eigenvalue (Eq. (9)). While Euler integration is computationally efficient, we refine each step using the golden section search method [11, (Chapter 10.1)], ensuring alignment with ridge peaks (Fig. 5 (F)). Existing methods for ridge extraction may also be applicable; however, we chose this parameter-free tracing approach because it yields nearly equidistant points (calculated via Eq. (10)) along each mid-polyline, which facilitates uniform mesh generation and integrates easily into our 2D slice-based workflow.

Tracing of the mid-polyline continues until it either completes a loop or exits the connected component's boundary. If a point exits the connected component without forming a loop, tracing proceeds in the opposite direction from the starting point to ensure the entire polyline is traced.

A crucial aspect of this process is treating the vector field as a line field, as mid-polylines lack inherent directionality. Vectors are adjusted (flipped) when their orientation deviates by more than 90° from the preceding vector to maintain continuity [12].

In cases where boundaries are very thin, the next point in the tracing process may fall outside the boundary, prematurely ending the process. To address this, morphological dilation is applied to connected components, increasing their width by one pixel. This ensures a more accurate determination of whether an exit is genuine or caused by thin boundaries (Fig. 4). Importantly, the dilated slice is used solely for termination checks and does not affect segmentation morphology or field calculations. This strategy ensures robust continuation of tracing in thin regions while preserving nearly equidistant sampling and avoiding the need for additional heuristics.

For intricate topologies or suboptimal segmentations—such as branching structures uncommon in biological data but frequent in low-resolution datasets—an additional verification phase ensures comprehensive tracing. This phase identifies untraced segments within a connected component, treating them as independent components and re-initiating mid-polyline extraction for each.

Once all points within a single connected component are traced, the procedure advances to subsequent components. This process continues slice by slice until all components are addressed. After extracting mid-polylines across all slices, the workflow transitions to triangulating these polylines in the next phase of our method (Fig. 5).

Polyline Zipper Algorithm

We developed a triangulation method named the Polyline Zipper Algorithm to transform extracted mid-polylines into a mid-surface mesh. This algorithm utilizes polylines stacked on parallel planes at equidistant levels, where each horizontal level corresponds to a slice of volumetric data. At each level, the mid-polyline comprises either a single continuous line or multiple line segments. Vertices and edges from adjacent slices act as *teeth* in the mesh generation process, akin to a zipper.

In our zipper-like approach, vertices from adjacent slices are systematically connected, traversing neighboring polylines and triangulating edges from both sides to form triangle strips. This ensures comprehensive connectivity across the resulting triangular surface mesh.

Let $S = \{S_i\}_{i=1}^{n}$ represent the set of mid-polylines for the n slices. The edges contributing to zipper connectivity fall into two categories: valid pairs and non-pair edges. Two edges, $e_i \in S_i$ and $e_j \in S_{i+1}$, form a valid pair if the Euclidean distance between their centers, $d(e_i, e_j)$, is

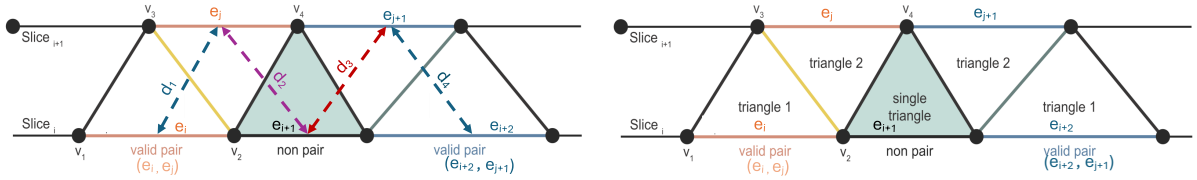

FIGURE 6. Zipper approach for connecting edges from two neighboring slices. Left: Edge pairing step based on nearest distances. Dashed arrows indicate computed distances (d_1, d_2, d_3, d_4), with inequalities ($d_1 < d_2$ and $d_4 < d_3$) illustrating how valid pairs are selected according to shortest distances. Right: Valid pairs, exemplified by edges $e_i \in \text{Slice}_i$ and $e_j \in \text{Slice}_{i+1}$, form two triangles (shown in white). The green-colored triangle represents a single triangle formed by a non-pair edge.

shorter than the distance between e_i and any other edge $e_k \in S_{i+1}$, and shorter than the distance between e_j and any other edge $e_m \in S_i$. Mathematically, this is expressed as:

$$d(e_i, e_j) < d(e_i, e_k), \quad \forall e_k \in S_{i+1}, \tag{11}$$

$$d(e_i, e_j) < d(e_m, e_j), \quad \forall e_m \in S_i. \tag{12}$$

If e_i is closest to $e_j \in S_{i+1}$ but e_j has another edge $e_m \in S_i$ that is closer than e_i, then e_i is classified as a non-pair edge. In other words, if only Eq. (11) is satisfied but not Eq. (12), then $e_i \in S_i$ is a non-pair edge. Similarly, if only Eq. (12) is satisfied but not Eq. (11), then $e_j \in S_{i+1}$ is considered a non-pair edge.

The four vertices v_1, v_2, v_3, and v_4 of each valid pair of edges are connected to form two triangles (Fig. 6). To ensure higher-quality triangles, the diagonal edge is established between vertices v_1 and v_3, or between v_2 and v_3, prioritizing vertices with larger interior angles. For example, in Fig. 6 (left valid pair), $m\angle v_1 v_2 v_4$ is greater than $m\angle v_3 v_1 v_2$, so v_2 and v_3 are diagonally connected.

For non-pair edges $e_i \in S_i$ (Fig. 6), the valid vertex from the nearest edge $e_j \in S_{i+1}$ is connected to form a single triangle. Non-pair edges may arise when the number of vertices between adjacent slices differs. By flagging edges as paired or non-paired using Eq. (11) and Eq. (12), and carefully handling both types, the method ensures robust connectivity even when multiple mid-polylines exist on the same slice.

Additionally, the algorithm accounts for slice distances and unconnected regions adjacent to zipper-like connectivity. These unconnected regions intentionally preserve holes in the original data, ensuring the mesh's integrity and fidelity to the segmented structure.

Triangle strip generation progresses iteratively between consecutive polylines across all slices. Once completed, the segmented dataset yields a complete mid-surface mesh. The mesh generation process, from input mid-polylines to the final surface mesh, is illustrated in Fig. 7.

Unlike reconstruction algorithms such as simple and scalable surface reconstruction (SSSR) [13] or Delaunay mesh reconstruction, our approach directly connects extracted points at their precise locations, mitigating the risk of losing the original input geometry's fidelity. Additionally, our direct triangulation method handles gaps in the data without requiring extra input parameters, setting it apart from conventional mesh reconstruction techniques. The output mesh meets targeted accuracy requirements while demonstrating superior quality and readiness for refinement processes such as smoothing, simplification, or mesh quality improvements based on application needs.

The mid-polylines provided to the Polyline Zipper Algorithm are labeled with slice numbers during the Mid-Polyline Extraction Algorithm. This labeling, combined with the pairing strategy defined by Eq. (11) and Eq. (12), ensures that each edge $e_i \in S_i$ is connected to its nearest valid pair $e_j \in S_{i+1}$. This ensures robust connectivity even with branches in the mid-surface. For non-pair edges, the distance between adjacent slices serves as a threshold to create edges or preserve holes in the surface.

The Mid-Polyline Extraction Algorithm also generates points with (nearly) uniform distribution horizontally and vertically. The step size between adjacent points on a polyline is adjusted relative to vertical slice spacing to produce nearly equilateral triangles. However, inward bending of the mid-surface across slices may reduce uniformity near boundaries (Fig. 2). While bending effects are typically minimal and progressive, increasing vertex counts for uniformity may impact computational efficiency in terms of time and memory.

Experimental Results

In this section, we present the outcomes of applying our approach across various datasets and scenarios, demonstrating its versatility and efficacy in mid-surface extraction and analysis. Our method's performance is first evaluated through a series of surface generation examples, where we present its capabilities and compare them against those of traditional approaches. Next, we analyze the mesh quality visually and numerically and compare our mesh quality

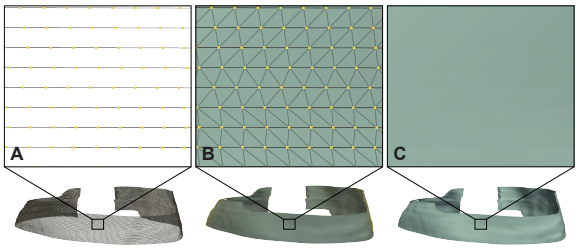

FIGURE 7. Mesh generation results. (A) Mid-polylines and the points extracted from slices. (B) The triangle mesh, a result of the triangulation process with new edges connecting points between the mid-polylines. (C) The final surface mesh without visible edges to improve visual clarity.

with the existing method. Finally, we illustrate the practical applications of our results in two specific domains: the modeling of biological structures and the estimation of membrane curvature from microscopic data. The proposed algorithm is implemented as a ParaView plugin in C++ and tested on Intel(R) Xeon(R) Gold 6230R CPU 2×2.10 GHz with 256 GB RAM and 64 bit Windows 10 operating system.

For our surface generation experiments, we use a well-documented segmentation dataset provided by Klein *et al.* [14], showing the structural details of SARS-CoV-2. This dataset is particularly useful for showcasing our results because it includes segmentation of the virus and its host cells, which feature membranes of notable thickness. The data was initially segmented through automated processes and then refined manually, providing us with high-quality, reliable data for our initial experiments. We use another dataset from Barad *et al.* [7], enabling direct comparison with their methodology. This dataset consists of segmentation volumes highlighting the mitochondrial structure and contains two distinct labels: the inner and outer mitochondrial membranes. These segmentations were derived semi-automatically [15] from specimens prepared using cryo-FIB milling and scanned using cryo-ET. This dataset, despite its complexity beyond our initial experiments, offers a tangible real-world application scenario for our algorithm, demonstrating its applicability.

Surface Generation Results

The final outcomes of our algorithm are triangular surface meshes. Therefore, to evaluate our algorithm, we used different types of data to observe the visual results of our method and confirm its robustness. First, we examine the applicability of two existing methods for surface extraction. As described in related work, various methods exist for generating surface meshes, including medial and

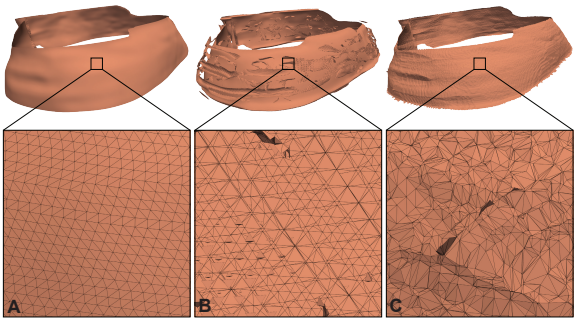

FIGURE 8. Evaluation of different surface types highlighting their limitations for our application. (A) The mid-surface generated by our method, showing a regular, uniformly triangulated mesh. (B) The ridge surface extracted using the VCG ParaView Plugin [16], which produces a similar median geometry but results in uneven point sampling and irregular triangles. (C) The medial surface computed by VoxelCores [17], illustrating common topological artifacts and inconsistent resolution in thin regions.

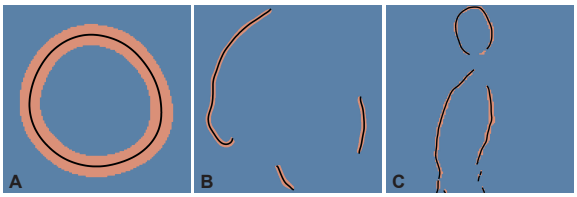

FIGURE 9. Performance of the Mid-Polyline Extraction Algorithm across slices featuring varied topologies. (A) A circular segmentation, illustrating the algorithm's efficacy in linking connected loops. (B) A slice containing multiple components, resulting in the generation of several mid-polylines. (C) A complex segmentation with notably thinner boundaries. The segmentations were sourced from datasets provided by Klein *et al.* [14] (A, B) and Barad *et al.* [7] (C).

isosurfaces. However, they are not directly applicable to our target objectives. To highlight these differences and shortcomings, we tested two existing approaches, namely the ParaView plugin [16] for ridge surface extraction and the VoxelCores method [17] for the medial surface extraction. The results are shown in Fig. 8, where we can see that the mid-surface generated by our method has no specific issue. In contrast, both of the previous methods introduced unnecessary additional surface portions and redundant mesh elements in the output mesh. This underscores the necessity and significance of our algorithm in providing robust results.

Next, we evaluated the robustness of our Mid-Polyline Extraction Algorithm. Figure 9 shows the results of applying the Mid-Polyline Extraction Algorithm to slices from

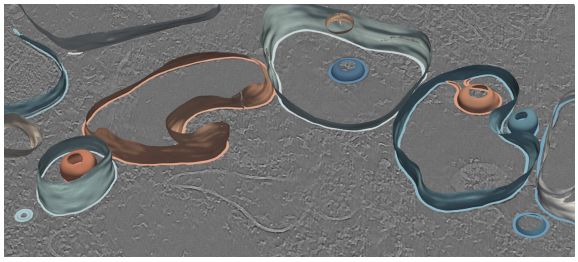

FIGURE 10. Mid-surfaces generated from the SARS-CoV-2 dataset [14] overlaid on the original tomogram slice and the segmentation mask.

diverse datasets. These results demonstrate that despite variations in segmentation thickness and complexity, our algorithm consistently performs well.

We also evaluated our method on various types of data to see the overall performance from input volumetric data to the final surface mesh. Figs. 2 and 7 show the results of two instances of the SARS-CoV-2 dataset [14], along with the intermediate results of mid-polylines. The surface meshes generated from the overall dataset [14] are presented in Fig. 10. We can see a slice of the segmentation and the corresponding extracted mid-surface meshes. The results contain 25 isolated objects. Most of these objects in the segmented input data are without missing information, as exemplified by the three models shown in Figs. 2, 7 and 8. However, for some objects, the input segmentation has holes in the surfaces, which are preserved in the output. These holes might be true gaps in the surface or missing information due to noise in the data. In either case, our algorithm generates surfaces based on the input data without adding new details, preserving the holes. To further demonstrate the accuracy of the resulting mid-surfaces, we have conducted an evaluation on a synthetic dataset, which can be found in the supplemental material.

Assessment of Mesh Quality

Mesh quality is essential for various applications. In this section, we evaluate the quality of the generated meshes and compare their visual appearance and quantitative metrics with existing methods. First, we describe the mesh quality metrics used for our analysis, and then we present the results.

Mesh quality metrics used:

Following standard quality metrics from the meshing domain [6], we evaluated our mesh using the *triangle quality (Q)*, whose value varies from 0 (poor-quality) to 1 (good quality) and is calculated as:

$$Q = \frac{6}{\sqrt{3}} \frac{A}{(p \times h)}, \tag{13}$$

were, A is the area of triangle p is its half-perimeter, and h is the length of its longest edge. The angles of the triangles are also crucial for assessing quality. Small and large angles can indicate poor quality. Therefore, we computed the *percentage of angles less than* 30° and *greater than* 120°. Similarly, we determined the *average value of the minimal angles* $\overline{\theta}_{\text{min}}$ in all triangles. Valence refers to the number of adjacent edges to a vertex. While 6 is considered the optimal valence, we noted the *percentage of vertices with valence numbers 5, 6, or 7* V_{567}, which is an important metric [18]. The optimal valence meshes are called regular meshes that are easily remeshed to improve other quality metrics.

Mesh Quality Results:

Table 1 presents the numerical mesh quality results, demonstrating that our meshes are highly regular, with the proportion of V_{567} vertices ranging from 89% to 97%. The ratio of poor-quality triangles (low Q values, triangles with very small or very large angles) is minimal. For instance, Q_{avg} is between 0.78 and 0.83, and $\overline{\theta_{min}}$ is consistently above 41°. Across all metrics, our method outperforms the PSR$*$ method in terms of mesh quality. In terms of processing time, our method is slightly slower than PSR$*$ but remains within an acceptable range. For simpler datasets (see the last three rows in Table 1), it performs efficiently, while for more complex datasets, the runtime increases. Notably, the time required for mid-polyline generation is higher than for meshing itself. However, since our method is inherently parallelizable, this aspect will be addressed in planned future work.

For comparison, we look into the mesh quality metrics of the existing recently proposed method, namely the PSR-based approach developed by Barad *et al.* [7]. In their method, the segmentation volumes are converted into unoriented point clouds, with each voxel's center corresponding to a point in the cloud. Normal vectors are estimated for each point based on neighboring points, followed by the application of a single smoothing iteration. The oriented point cloud is then used to generate a surface mesh using the screened Poisson algorithm, with reconstruction parameters such as depth, interpolation weight, and the minimum number of samples needing to be manually set for each dataset. The resulting mesh is refined by removing triangles that extend beyond the segmented region and simplifying the surface to ensure it has no more than 150,000 triangles. For more detailed information on this method, we encourage readers to refer to the original paper.

Our results show a significant improvement. Similarly, the visual results of the meshes are shown in Fig. 11, where, if we look into the zoom view, we can see a significant improvement in mesh regularity and angle quality over the existing method. The generation of a high-quality mesh is

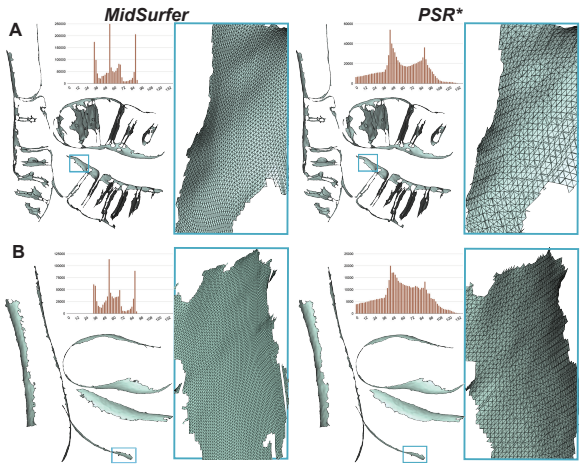

FIGURE 11. Triangulation results: Left: Ours, Right: PSR*, the PSR based approach developed by Barad *et al.* [7]. The top row shows TE7-IMM, and the bottom row depicts TE7-OMM. The quantitative results are shown in Table 1. The hole in the MidSurfer results (bottom-left) persists from the input data, which is refilled by the PSR* (bottom-right).

TABLE 1. Quantitative results of mesh quality (PSR* [7] vs. Ours). Q represents triangle quality calculated via Eq. (13). $\overline{\theta_{min}}$ is the average of the minimal angles of all triangles, and V_{567} represents the percentage of regular vertices (having valences 5, 6, or 7). Additionally, the percentages of small and large angles are also shown. The time (in Ours result) is the time taken for mid-polylines generation + the time taken for meshing. PSR* time is the total time taken.

Method	Model	#Vertices	Q_{min}	Q_{avg}	$\overline{\theta_{min}}$	$\theta < 30°$	$\theta > 120°$	V_{567}	Time (sec.)
PSR*	Fig. 11 (A)	165808	0.00	0.60	31.69	14.00 %	0.77 %	67 %	172
Ours	Fig. 11 (A)	289088	0.03	0.78	41.26	0.02 %	0.01 %	89 %	332 + 130
PSR*	Fig. 11 (B)	87423	0.00	0.60	31.0	14.88 %	0.91 %	70 %	92
Ours	Fig. 11 (B)	150234	0.00	0.79	42.57	0.05 %	0.02 %	92 %	283 + 40
Ours	Fig. 7	69939	0.01	0.83	44.56	0.03%	0.01 %	97 %	14 + 12
Ours	Fig. 13 (A)	10409	0.23	0.82	43.93	1.09 %	0.00 %	97 %	2 + 1
Ours	Fig. 13 (B)	39796	0.04	0.81	44.0	0.02 %	0.01 %	97 %	4 + 4

ensured by generating a highly regular pattern of vertices in the mid-polylines and then triangulating these lines using a quality-aware method by the Polyline Zipper Algorithm. In addition, Table 1 includes a time analysis, indicating that our method demonstrates longer processing times compared to that of Barad, primarily due to the higher number of vertices involved. Still, the times are on the same order of magnitude and remain within acceptable limits. Additionally, our method has a high potential for acceleration through parallelism. The higher number of vertices is due to ensuring high accuracy by considering the same resolution as that of the input data, which defines the inter-slice spacing, thereby affecting the step size in point generation.

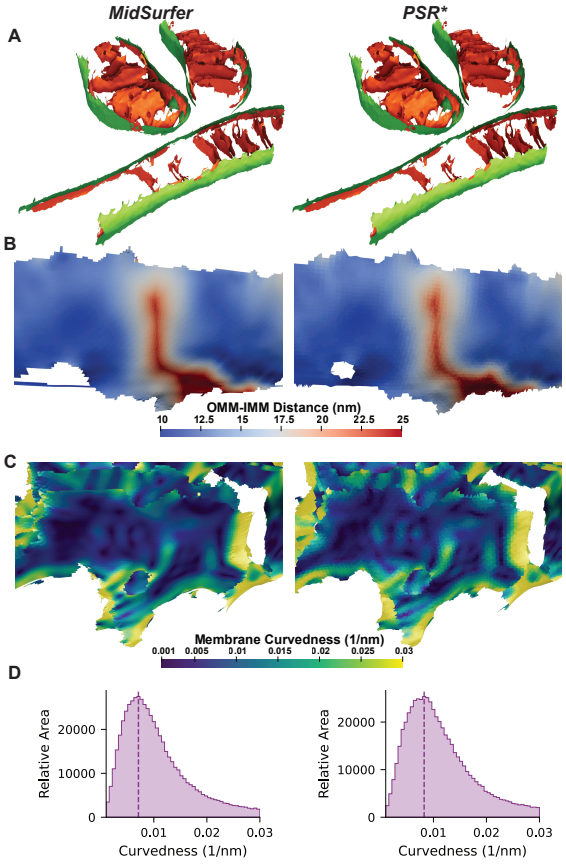

FIGURE 12. Morphometric comparison of membrane surfaces generated with MidSurfer vs. Screened PSR. (A) Surface meshes were generated from segmentations depicting mitochondrial inner and outer membranes observed by cryo-ET, then quantified with the Surface Morphometrics toolkit [7]. Visual comparison shows similar results overall, with improved smoothness of quantification for (B) inter-membrane distance and (C) membrane curvedness, demonstrating the effectiveness of MidSurfer surfaces for downstream quantitative analysis. (D) Histogram analysis of the surface measurements shows comparable collective quantification of curvedness but with a smaller high-curvature tail as a result of reducing artifactual high-curvature measurements. PSR* is the PSR based approach developed by Barad *et al.* [7].

Use Case Applications

Mid-surface, once extracted, can be utilized in many applications. In this section, we showcase two use cases of the generated mid-surfaces obtained from cryo-ET experiments.

Surface morphometrics:

The Surface Morphometrics toolkit [7] was recently developed to quantify the ultrastructure of biological membranes, including membrane curvature, orientation, and

inter-membrane spacing, using mid-surface models. For this purpose, they utilized the screened PSR method to generate mid-surface meshes, which the toolkit [7] then uses for various statistical analyses. Here, we aim to apply our mid-surface meshes for these quantifications and compare them with the method proposed by Barad *et al.* [7].

Figure 12 illustrates the results of these quantifications for mid-surfaces generated using MidSurfer compared to those generated by the Screened PSR-based method [7]. The results are comparable both visually and statistically, with MidSurfer largely reproducing similar results. This underscores that the meshes produced by MidSurfer yield results comparable to those of a recently proposed method. The consistent triangle size and high degree of smoothness slightly reduced high curvature measurements introduced by quantization artifacts in flat membrane segments and generally improved visualization smoothness.

It is important to note that achieving results similar to those presented in Fig. 12 using the PSR-based method [7] requires expert users to fine-tune the mid-surface extraction parameters. These results were achieved after the authors of the technique invested time in fine-tuning mid-surface extraction parameters to achieve the best possible results. It is highly unlikely that infrequent or new users would achieve similar accuracy. In contrast, MidSurfer employs a parameter-free approach, providing results with a single click. For such users, our approach is likely to yield superior mid-surface extraction compared to manually tuning the Barad *et al.* toolkit.

Modeling:

Mid-surface extraction plays a crucial role in biological modeling and molecular dynamics (MD) simulations, where accuracy and realism are paramount. These simulations aim to mimic the behavior of biomolecules and their interactions in a computational environment, providing invaluable insights into biological processes at the molecular level. A key challenge in MD simulations is accurately representing lipid bilayer membranes, essential for cell structure and function. Here, we highlight a use case of mid-surface extraction in biological modeling using MesoCraft, a mesoscale modeling tool [19]. We demonstrate the applied utility of mid-surface extraction within the domain of biological modeling, focusing on the SARS-CoV-2 virion. As illustrated in Fig. 13, we leverage the dataset provided by Klein *et al.* [14] for the generation of detailed viral surface depictions. The advantage of mid-surface extraction is evident in the positioning of various components on the virion's surface mesh, notably the lipids constituting the lipid bilayer membrane. It effectively avoids artifacts like membrane shrinking or bloating, ensuring faithful representations of biological membranes' behavior. Such accurate modeling was not achievable with other alternatives, such as using the inner and outer surfaces of the particle membrane.

Discussion

We present a technique for mid-surface extraction from volumetric microscopy data of biological structures, addressing a gap in current methodologies. Unlike previous methods, which often require extensive manual tuning of parameters, our findings demonstrate that our approach can accurately extract the mid-surface from segmentation data of various topologies without any user-defined parameterization. For reproducibility and expert control, the open-source plugin exposes the automatically selected internal scales (e.g., the smoothing σ derived from $SDF_{\max}$) under *Advanced Settings*, while keeping the default pipeline parameter-free for typical use.

MidSurfer is designed for vesicular, shell-like structures in structural biology (e.g., cryo-ET) and is not evaluated on medical/vascular data. Centerline/medial-axis techniques such as Wischgoll *et al.* [4] and Telea and Vilanova [5] target *one-dimensional* curve skeletons of tubular lumina; in contrast, we estimate a *mid-surface* embedded between paired membranes. This distinction of 1D curve skeleton vs. mid-surface implies different algorithmic goals and outputs. A practical implication of our design is the uniformly triangulated surface mesh, which benefits downstream tasks such as FEM, tetrahedral meshing, and morphometric analysis, where mesh regularity strongly affects stability and accuracy [6].

Our research further solidifies the choice of mid-surfaces over more commonly utilized surface representations, a conclusion we show in Fig. 8. The direct comparison conducted on the same dataset reveals that both isosurfaces and medial surfaces exhibit critical limitations that undermine their utility for our scenario. Notably, the definitions of isosurfaces and medial surfaces invariably lead to the generation of artifacts that persist despite exhaustive manual adjustments of parameters. Furthermore, the generation of irregular meshes using these methods necessitates subsequent remeshing to attain a usable form for further analysis, presenting a significant procedural inefficiency. These findings strongly support the use of the mid-surface approach that not only captures the complex geometries of biological structures but also avoids the common problems associated with more traditional surface representations.

The MidSurfer method relies on the input of accurately segmented volumetric data. This prerequisite, while ensuring high precision in mid-surface extraction, makes our method dependent on segmentation quality. In cases of suboptimal segmentation, particularly with thin boundaries (e.g., missing labeled voxels, unconnected components where there should be continuity or unexpected branches

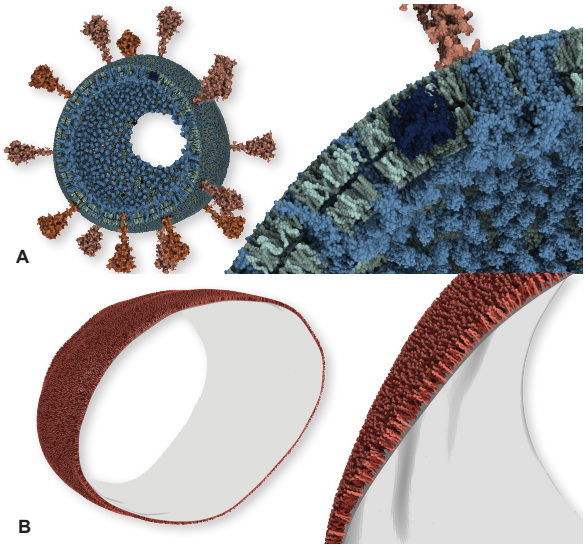

FIGURE 13. Modeling results with MesoCraft [19]. (A) SARS-CoV-2 virion with the lipids of the lipid bilayer membrane placed on both sides of the extracted mid-surface. (B) Lipids of the lipid bilayer membrane populated only on the outer side of the surface.

from otherwise smooth boundaries), the efficacy of our approach might be affected. Addressing this issue could entail developing a sophisticated preprocessing stage, but currently, this adjustment is left to the discretion of users based on their specific needs. In the context of branches, our slice-adjacent pairing is effective on our datasets; however, highly branched or vascular-like structures may require more topology-aware matching, which we leave for future work.

Additionally, our method in its current form does not cater to the extraction of mid-surfaces from capped structures, i.e., segmentations that feature disk-like shapes on the lower or upper bounds without an inner boundary within a single slice. Defining a mid-polyline in these cases becomes challenging due to the lack of information from adjacent slices, leading to uncertainty in determining the correct direction for the mid-surface. Initially, our focus was on microscopic data, which usually do not include these kinds of topologies. This is because the structures we are interested in often extend beyond a single volume's thickness, resulting in slices that contain only the interior regions of interest (e.g., mitochondria) without the end caps. Consequently, we did not prioritize addressing this specific limitation. However, our current method automatically identifies and bypasses these structures. Future efforts could delve into multi-dimensional slicing techniques, using data from orthogonal planes to enhance the extraction process.

In a few cases, MidSurfer failed to reconstruct sections of the surface at the edges of mitochondrial segmentations (Fig. 12), where the segmentations were thinner; however, these edge regions are routinely ignored during quantification due to reduced image and segmentation quality caused by the missing wedge [20]. These results broadly show that MidSurfer models are appropriate for detailed quantitative analysis of biological membrane ultrastructure.

In the context of meshing, we aim for high-quality results through uniform meshing, which requires a higher density of elements (see Table 1) and is therefore computationally expensive. Sharp regions, in particular, demand a denser distribution of elements to capture geometry accurately. However, uniform meshing applies the same density even to simpler areas, introducing unnecessary computational overhead. Additionally, mesh quality may degrade in regions with sharp boundaries where surface curvature increases abruptly. While the uniform mesh provides high quality and consistency, it may have limitations in cases requiring variable resolution. In future work, adopting adaptive meshing strategies could be a valuable direction to further improve surface detail.

One of the key strengths of our mid-surface extraction algorithm is its potential for parallelization. While the current implementation processes slices sequentially, the algorithm is designed so that each slice can be handled independently, making it straightforward to parallelize. This will significantly speed up processing, which is particularly important for the large volumetric datasets often encountered in biological research. Parallelization is one of several advantages of using a 2D approach, particularly in the context of cryo-ET data. The 2D slicing approach allows for computational efficiency and practical implementation while still producing accurate mid-surface reconstructions. Additionally, given the anisotropic resolution of cryo-ET data, where the z-axis typically provides the highest resolution due to the effects of the missing wedge, working slice by slice aligns naturally with the data characteristics. Together, the efficiency, accuracy, and scalability of the 2D approach make it an ideal choice for our method.

Conclusion

In this work, we introduced and formally defined a type of surface, denoted as mid-surface, which is an essential structure for modeling and analysis of 3D microscopy data in structural biology. The concept of mid-surfaces has been notably absent in the visual computing literature until recently, when domain scientists developed a highly effective workflow for mid-surface extraction [7], utilizing screened Poisson Surface Reconstruction. Unlike their bespoke workflow, our approach is grounded in geometric principles and crucially eliminates the need for parametric tuning. It achieves a quality of output comparable to that of the

previous method under optimal parameter configurations. The mesh produced by our Polyline Zipper Algorithm is characterized by its nearly equilateral triangles, rendering it highly suitable for a range of analytical and processing tasks. While it might be possible to derive mid-surfaces by processing medial surfaces (e.g., pruning branches) or through ridge extraction [3] (e.g., removing disconnected components), and subsequently applying smoothing and remeshing techniques, such an approach would be significantly more laborious in terms of parameter optimization and less efficient compared to our streamlined one-click solution.

Admittedly, the success of our method depends upon the quality of the initial segmentation. Moving forward, we aim to develop a mid-surface extraction workflow that facilitates quick iterations between segmentation and extraction, allowing for visual assessments and iterative refinements to enhance the accuracy. Both the mid-polyline extraction and the zipping algorithms present opportunities for parallelization, suggesting potential for their development into high-performance techniques that deliver immediate results.

As a forthcoming development, MidSurfer will be integrated into surface morphometrics workflows within the realm of structural biology, promising substantial contributions to the field. With the algorithm accessible as a ParaView plugin, we anticipate its adoption across new application areas, solidifying its position as a go-to method for mid-surface extraction in diverse disciplines engaged in volume data analysis.

REFERENCES

1. H. Blum, “A Transformation for Extracting New Descriptors of Shape,” in *Models for the Perception of Speech and Visual Form*. MIT Press, 1967, pp. 362–380.
2. E. Keppel, “Approximating Complex Surfaces by Triangulation of Contour Lines,” *IBM J. Res. Dev.*, vol. 19, no. 1, pp. 2–11, 1975.
3. T. Lindeberg, “Edge detection and ridge detection with automatic scale selection,” in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1996, pp. 465–470.
4. T. Wischgoll, J. S. Choy, E. L. Ritman, and G. S. Kassab, “Validation of image-based method for extraction of coronary morphometry,” *Annals of Biomedical Engineering*, vol. 36, no. 3, pp. 356–368, 2008. [Online]. Available: https://doi.org/10.1007/s10439-008-9443-x
5. A. Telea and A. Vilanova, “A robust level-set algorithm for centerline extraction,” in *Proceedings of the Symposium on Data Visualisation 2003*, ser. VISSYM ’03. Goslar, DEU: Eurographics Association, 2003, p. 185–194.
6. D. Khan, A. Plopski, Y. Fujimoto, M. Kanbara, G. Jabeen, Y. J. Zhang, X. Zhang, and H. Kato, “Surface Remeshing: A Systematic Literature Review of Methods and Research Directions,” *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 3, pp. 1680–1713, 2022.
7. B. A. Barad, M. Medina, D. Fuentes, R. L. Wiseman, and D. A. Grotjahn, “Quantifying Organellar Ultrastructure in Cryo-Electron Tomography Using a Surface Morphometrics Pipeline,” *J. Cell Biol.*, vol. 222, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:256844441
8. M. Rezayat, “Midsurface Abstraction From 3D Solid Models: General Theory and Applications,” *Comput.-Aided Design*, vol. 28, no. 11, pp. 905–915, 1996.
9. H. Bhatia, H. I. Ingólfsson, T. S. Carpenter, F. C. Lightstone, and P.-T. Bremer, “MemSurfer: A Tool for Robust Computation and Characterization of Curved Membranes,” *J. Chem. Theory Comput.*, vol. 15, no. 11, pp. 6411–6421, Nov 2019.
10. E. Moebel and C. Kervrann, “A Monte Carlo framework for missing wedge restoration and noise removal in cryo-electron tomography,” *Journal of Structural Biology: X*, vol. 4, p. 100013, 2020.
11. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*. Cambridge University Press, 1992.
12. W. Benger and H.-C. Hege, “Strategies for Direct Visualization of Second-Rank Tensor Fields,” in *Visualization and Processing of Tensor Fields*. Springer Berlin Heidelberg, 2006, pp. 191–214.
13. D. Boltcheva and B. Lévy, “Simple and Scalable Surface Reconstruction,” LORIA - Université de Lorraine; INRIA Nancy, Research Report, July 2016. [Online]. Available: https://hal.science/hal-01349023
14. S. Klein, M. Cortese, S. L. Winter, M. Wachsmuth-Melm, C. J. Neufeldt, B. Cerikan, M. L. Stanifer, S. Boulant, R. Bartenschlager, and P. Chlanda, “SARS-CoV-2 Structure and Replication Characterized by In Situ Cryo-Electron Tomography,” *Nat. Commun.*, vol. 11, no. 1, p. 5885, 2020.
15. A. Martinez-Sanchez, I. Garcia, S. Asano, V. Lucic, and J.-J. Fernandez, “Robust Membrane Detection Based on Tensor Voting for Electron Tomography,” *J. Struct. Biol.*, vol. 186, no. 1, pp. 49–61, 2014.
16. F. Sadlo, P. Hausner, L. Hofmann, P. Jung, G. Karch, R. Peikert, L. Pilz, M. Roth, and K. Sdeo, “VCG ParaView Plugins,” https://vcg.iwr.uni-heidelberg.de/plugins/, March 2019.
17. Y. Yan, D. Letscher, and T. Ju, “Voxel Cores: Efficient,

Robust, and Provably Good Approximation of 3D Medial Axes," *ACM Trans. Graph.*, vol. 37, no. 4, July 2018.

18. N. Aghdaii, H. Younesy, and H. Zhang, "5-6-7 Meshes: Remeshing and Analysis," *Comput. & Graphics*, vol. 36, no. 8, pp. 1072–1083, 2012.
19. N. Nguyen, O. Strnad, T. Klein, D. Luo, R. Alharbi, P. Wonka, M. Maritan, P. Mindek, L. Autin, D. S. Goodsell, and I. Viola, "Modeling in the Time of COVID-19: Statistical and Rule-Based Mesoscale Models," *IEEE Trans. Vis. Comput. Graph.*, 2020.
20. M. Salfer, J. Collado, W. Baumeister, R. Fernández-Busnadiego, and A. Martínez-Sánchez, "Reliable Estimation of Membrane Curvature for Cryo-Electron Tomography," *PLoS Comput. Biol.*, vol. 16, p. e1007962, August 2020.

Eva Boneš is a Ph.D. student at the University of Ljubljana, Faculty of Computer and Information Science. Her research interests include graphics, visualization, and nanoscale biomedical data. She received her master's degree in Computer and Information Science from the University of Ljubljana. She is currently a researcher in the Laboratory for Graphics and Multimedia in Ljubljana. Contact her at eva.bones@fri.uni-lj.si.

Dawar Khan is a Postdoctoral Fellow at King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia. His research interests include computer graphics, visualization, LLMs, VLMs, HCI, and augmented reality. He received his Ph.D. in Computer Science from the Institute of Automation, University of Chinese Academy of Sciences. Previously, he served as an Assistant Professor at the Nara Institute of Science and Technology in Japan and the University of Haripur in Pakistan. Contact him at dawar.khan@kaust.edu.sa.

Ciril Bohak is an Assistant Professor at the Faculty of Computer and Information Science at the University of Ljubljana, Ljubljana, Slovenia. His research interests include computer graphics, scientific visualization, and human-computer interaction. Bohak received his Ph.D. in Computer Science from the University of Ljubljana. He is the Chair of the Slovenian IEEE Computer Chapter. Contact him at ciril.bohak@fri.uni-lj.si.

Benjamin A. Barad is an Assistant Professor at Oregon Health & Science University (OHSU), Portland, Oregon, USA and is leading the Barad Lab at OHSU. His research interests include CryoET, computational tools, and cellular remodeling in response to intracellular bacteria. Barad received his Ph.D. in Biophysics from UCSF after graduating from Stanford University. He is a recipient of the American Cancer Society Postdoctoral Fellowship. Contact him at barad@ohsu.edu.

Danielle A. Grotjahn is an Assistant Professor at The Scripps Research Institute, La Jolla, California, USA and is leading the Grotjahn Lab at Scripps. Her research interests include mitochondrial dynamics under stress, advanced imaging techniques, and cellular biology. Grotjahn received her Ph.D. in Biophysics from The Scripps Research Institute. She is a Pew Scholar in Biomedical Sciences Award recipient. Contact her at grotjahn@scripps.edu.

Ivan Viola is a Professor at King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia and is leading the Nanovisualization research group. His research interests include interactive molecular visualization, data reconstruction, interpretation, representation, modeling, and rendering. Viola received his Ph.D. in Visualization from TU Wien in Austria. In 2013, he received a WWTF grant to establish a research group at TU Wien. Contact him at ivan.viola@kaust.edu.sa.

Thomas Theußl is with Consivi KG in Deutschlandsberg, Austria, which he founded in 2025, offering consulting, support, and development in scientific visualization. His research interests include visualization techniques, mathematical foundations of visualization, and applications of visualization. Theußl received his Master's degree in Computer Science from TU Wien in Austria. He has published tutorials on ParaView and Avizo on YouTube. Contact him at thomas.theussl@consivi.com.