

MIDSURFER: A PARAMETER-FREE APPROACH FOR MID-SURFACE EXTRACTION FROM SEGMENTED VOLUMETRIC DATA

A PREPRINT

✉ Eva Boneš^{1,2,*}, ✉ Dawar Khan^{1,*}, ✉ Ciril Bohak², ✉ Benjamin A. Barad³,

✉ Danielle A. Grotjahn⁴, ✉ Ivan Viola¹ and ✉ Thomas Theußl¹.

¹King Abdullah University of Science and Technology (KAUST), Saudi Arabia.

E-mail: {eva.bones | dawar.khan | ivan.viola | thomas.theussl} @kaust.edu.sa.

²University of Ljubljana, Slovenia. E-Mail: ciril.bohak@fri.uni-lj.si.

³Oregon Health & Science University, OR, United States. E-mail: barad@ohsu.edu.

⁴The Scripps Research Institute, CA, United States. E-mail: grotjahn@scripps.edu.

*E. Boneš and D. Khan are co-first authors.

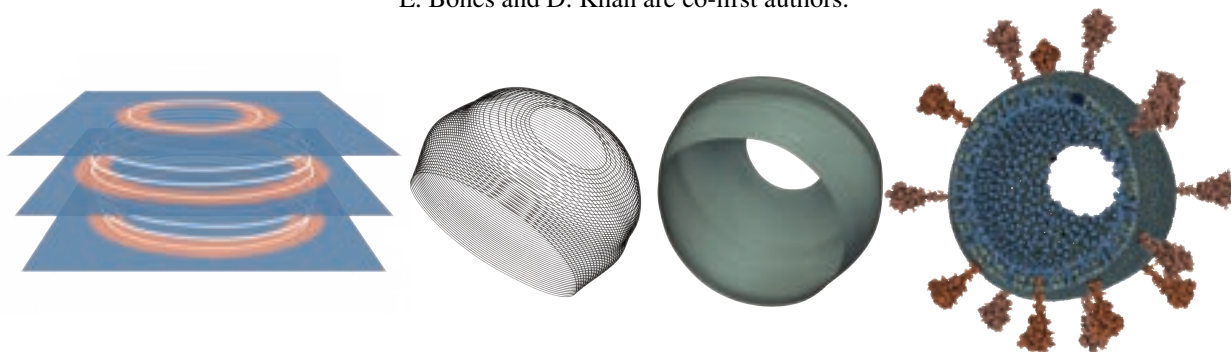


Fig. 1: An overview of the overall method: From left to right: Volumetric segmented data represented in slice-wise view, the extracted mid-polylines from the data, the triangular surface generated from mid-polylines, and a use case of the mid-surface with the modeling results visualized over the extracted mid-surface mesh.

ABSTRACT

In the field of volumetric data processing and analysis, extracting mid-surfaces from thinly bounded compartments is crucial for tasks such as surface area estimation and accurate modeling of biological structures, yet it has lacked a standardized approach. To bridge this gap, we introduce MidSurfer—a novel parameter-free method for extracting mid-surfaces from segmented volumetric data. Our method produces smooth, uniformly triangulated meshes that accurately capture the structural features of interest. 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 across the volume, creating a backbone of regularly distributed points on each slice that represents 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. Our findings demonstrate that this method surpasses previous techniques in versatility, simplicity of use, and accuracy. Our approach is now publicly available as a plugin for ParaView, a widely-used multi-platform tool for data analysis and visualization, and can be found at (<https://github.com/kaust-vislab/MidSurfer>).

Keywords Mid-Surface · Biomedical Visualization · Volumetric Data · Biological Structures · Parameter-Free · Surface Meshing

1 Introduction

In the exploration of biological structures, the visualization and analysis of surfaces play a crucial role in understanding 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.

The mid-surface concept [1] serves as a crucial tool in this context, effectively bridging the divide between raw volumetric data and a meaningful geometric representation of biological structures. Broadly defined, a mid-surface [2, 3] is a surface model that captures the structure’s geometric essence by depicting its median or central layer, which is equidistant from the object’s inner and outer surfaces. This model is invaluable for analyzing thinly bounded compartments, offering a well-balanced and precise surface for subsequent analysis. Notably, the mid-surface, situated between the inner and outer leaflets of the lipid bilayer, is of particular importance for understanding membrane structure and various cellular functions, such as membrane fluidity, protein localization, and cell signaling.

Distinguishing between mid-surfaces, medial surfaces [4], and isosurfaces [5], illustrated in Fig. 2, is crucial, as these concepts, while often related, serve distinct purposes in volumetric data analysis. Mid-surfaces are designed to represent the central geometry of thinly bounded spaces, capturing the essence of an object’s median layer. Medial surfaces, in contrast, focus on the geometric and topological core of a shape, essentially forming its skeleton. Isosurfaces, meanwhile, are defined by extracting surfaces that represent a constant value within a scalar field, a technique commonly employed in volume rendering. Additionally, the concept of outer surfaces [6, 7, 8], including molecular surfaces [9] and membrane surfaces [10], plays a significant role in various analysis scenarios. Among these, mid-surfaces are particularly advantageous for the detailed analysis of biological structures due to their unique ability to accurately represent the median geometry. However, it is noteworthy that, to the best of our knowledge, the scientific literature primarily mentions a single approach [1] for extracting mid-surfaces, which necessitates adjusting multiple parameters to achieve the desired outcome. Other methods for extracting similar surfaces, as discussed in section 2, are either not fit for purpose or fail to ensure the topological integrity of a manifold with a boundary.

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 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 from biological data. Our work addresses this necessity, proposing a novel, efficient method that enhances the geometric representation of biological structures, thereby facilitating a deeper understanding of their complex functions.

The key contributions of the presented method include:

- Introducing a novel, parameter-free, and robust method for extracting the mid-surface from segmented volumetric data.
- Representing the mid-surface in the form of a high-quality triangular surface mesh, while considering the 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 plugin in ParaView¹, an open-source visualization tool, thereby providing accessible functionality to a wide range of users, from novices to experts.

¹<https://www.paraview.org>

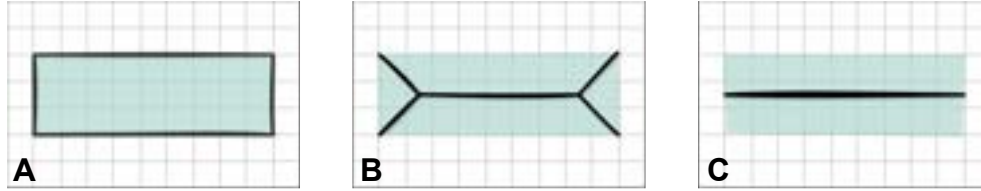


Fig. 2: Comparative illustration of (A) isosurfaces, (B) medial surfaces, and (C) mid-surfaces to emphasize the unique advantages of mid-surfaces in capturing the median geometry. The segmentation is shown in light green color and the respective surfaces with a thick black line.

2 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 by users or algorithmically extracted from reference data, encompassing a wide array of geometric representations, including meshes, point clouds, and volumetric data. Historically, numerous approaches for surface extraction have been developed, primarily influenced by the nature of the input data. However, it’s imperative to approach these methods with flexibility, as input data can often be converted among different formats with high precision. Consequently, we review related works capable of processing various types of input data.

Point cloud data, one of the prevalent forms of representation, poses unique challenges and opportunities for surface reconstruction. The goal is either to fit a surface that encompasses all points or to devise a surface model that optimally represents the point cloud. Pioneering contributions in this area were made by Hoppe *et al.* [11, 12], who introduced innovative techniques for the surface approximation of unorganized point sets. Boissonnat and Cazals [13] offered a method for smooth surface reconstruction from arbitrary point distributions using Natural Neighbour interpolation of the distance function. Approaches to simplify surfaces derived from unstructured point clouds were advanced by Pauly *et al.* [14, 15]. Subsequent research introduced methods for fitting smooth surfaces to arbitrary shapes using moving least squares combined with local mappings or other features [16, 17, 18, 19], with further developments aimed at preserving sharp features [20, 21, 22]. Ohtake *et al.* [23] demonstrated surface fitting excellence using multi-level partitioning of unity implicits. The advent of deep neural networks has also made a significant impact on surface reconstruction, achieving remarkable outcomes. Williams *et al.* [24] explored the use of deep neural networks as a geometric prior in this context, while Boulch and Marlet [25] tackled the scalability challenges inherent in deep learning-based surface reconstruction techniques. Comprehensive reviews of these advancements are documented in survey papers by Berger *et al.* [26] and Huang *et al.* [27].

It is worth noting that many of the above-mentioned methods are adaptable to volumetric data through a straightforward conversion of volumes to point clouds, as illustrated by some approaches, including [23], which demonstrate promising results. However, these techniques are primarily designed for unstructured data, contrasting with the highly regular data characteristic of our use cases. By leveraging the regularity inherent in our datasets, our method achieves greater efficiency and alignment with our specific requirements.

While point cloud data is frequently associated with surface reconstruction, the challenges inherent in working with volumetric data stem from its continuous representation of structures. Unlike point clouds, where the goal is often to reconstruct discrete surfaces or a collection of interconnected surfaces, volumetric data requires an initial determination of what constitutes a surface within the volume. Typically, this classification divides surfaces into three primary categories as illustrated in Fig. 2: (1) *isosurfaces* [5], which delineate boundaries within volumes of uniform value; (2) *medial surfaces* [28] (also known as *medial axes*), representing the structural skeleton; and (3) *mid-surfaces*, situated equidistantly between two boundary surfaces, such as isosurfaces or the edges of a segmented region. Our research primarily addresses the latter category, though we acknowledge key contributions to the former two due to their relevance and impact on our work.

The Marching Cubes algorithm [29], celebrated for its simplicity and efficiency, remains the quintessential method for isosurface extraction, leveraging linear interpolation along voxel edges. Subsequent attempts to resolve its ambiguities or improve upon its drawbacks, such as those by [30, 31], have not supplanted it, often due to increased complexity and reduced performance. Efforts to generate closed and oriented surfaces through valid triangulation include Guzetic *et al.*’s [32] tetrahedral decomposition-based method. Livnat *et al.* [33] proposed a near-optimal isosurface extraction technique for both structured and unstructured grids using span space representation. Wood *et al.* [34] introduced an approach tailored for distance volumes, beginning with a coarse mesh extraction that is refined via a

multiscale force-based solver to achieve a semi-regular mesh with adaptive geometric sampling. To mitigate volume sampling aliasing, Kobbelt *et al.* [35] developed a feature-sensitive surface extraction method that preserves the simplicity of Marching Cubes. Ju *et al.* [36] presented an efficient, real-time capable octree-based approach utilizing Hermite contouring, which does not necessitate the explicit identification and processing of specific features. Addressing the precision required for medical surface extractions, Kim *et al.* [37] demonstrated an approach using a Laplacian map, particularly for extracting cortical surfaces in the human brain. The advent of differentiable methods for isosurface extraction, aimed at facilitating rapid optimization with physically informed neural networks, led to novel approaches like that of Remelli *et al.* [38], which employs Deep Signed Distance Functions for explicit surface mesh representation, demonstrated in single-view reconstructions and physically-driven shape optimization.

Although the methods described above do not directly address our specific challenges, theoretically, our problem could be reformulated as an isosurface extraction task in some specific scenarios. However, this is not applicable to many of our cases due to the complexity of the structures we aim to address.

Medial surfaces, often depicted through medial meshes [39], encapsulate an object’s core geometric and topological features [40]. Yet, in the context of our work, medial surfaces prove inadequate due to the complexity of the structures we examine. The skeletons of these complex structures are typically branched and fail to provide a practical foundation for mesoscale modeling.

This limitation underscores the necessity for a mid-surface representation [2], a concept widely applied in fields such as engineering and finite element analysis [41], as well as structural biology, as highlighted by Barad *et al.* [1]. In contrast to isosurfaces, which excel in depicting uniform structures by showcasing constant values or medial surfaces that reduce shapes to their skeletal essence, mid-surfaces uniquely capture the median layer situated between two boundaries. This distinct trait renders mid-surfaces particularly adept for the detailed geometric exploration of thinly bounded structures.

Barad *et al.*’s approach employs a mid-surface morphometrics pipeline alongside meshing techniques to model and quantify various metrics, including curvature and distances within and between surfaces, in addition to orientation differences. Their methodology integrates several existing techniques, each with specific input parameters to yield optimal outcomes. They adopted a semi-automated algorithm [42] for segmenting cellular membranes from cryo-ET data, followed by the use of screened Poisson Surface Reconstruction (PSR) [43] for mesh generation. While PSR effectively transforms membrane voxel segmentations or point clouds into implicit surface meshes, there’s a notable risk of geometrical fidelity loss and the obliteration of sharp features and voids unless input parameters are meticulously chosen. Despite these challenges, the quantitative findings of Barad *et al.* provide a robust solution to a complex problem. Nonetheless, the quest for optimal parameter settings remains daunting. Their investigations further highlight the demand for more automated, parameter-free, and structurally reliable methods for mid-surface extraction—our research’s primary aim.

In one phase of our method, we employ a technique to seamlessly merge parallel contours into a unified mesh representation, a process known as *zippering*, the Zipper algorithm, or contour stitching. Various strategies for addressing this challenge in a broader context have been documented in prior works [5, 44, 45]. Our approach is particularly influenced by the method developed by Turk and Levoy [46], who presented the method for the creation of zippered polygon meshes. Their algorithm focuses on amalgamating multiple mesh fragments obtained from different viewpoints into a cohesive mesh that represents a singular object. Unlike their mesh-to-mesh algorithm, our implementation is tailored to converting stacked polylines into mesh structures. Furthermore, our method innovatively incorporates new triangles to facilitate connections between these structures—a modification not explicitly required in Turk and Levoy’s original formulation. It is also worth noting the existence of Zipper [47], which introduces a mesh data structure aimed at efficient mesh representation. However, this does not directly pertain to the mesh generation.

Another phase of our method extracts mid-polylines from individual slices, a problem that is closely related to ridge extraction as presented by Furst *et al.* [48] and by Furst and Pizer with Marching Ridges [49], where ridges are defined as level sets of first derivatives of the input function. This has been subsequently used, e.g., by Sadlo and Peikert [50] for visualizing Lagrangian coherent structures extracted as ridges from FTLE fields, and Kindlmann *et al.* [51] for identifying white matter structures in Diffusion Tensor MRI.

3 Method

Our method processes binary segmentations of volumetric data to derive mid-surfaces from individually segmented structures. Two critical definitions underpin our approach:

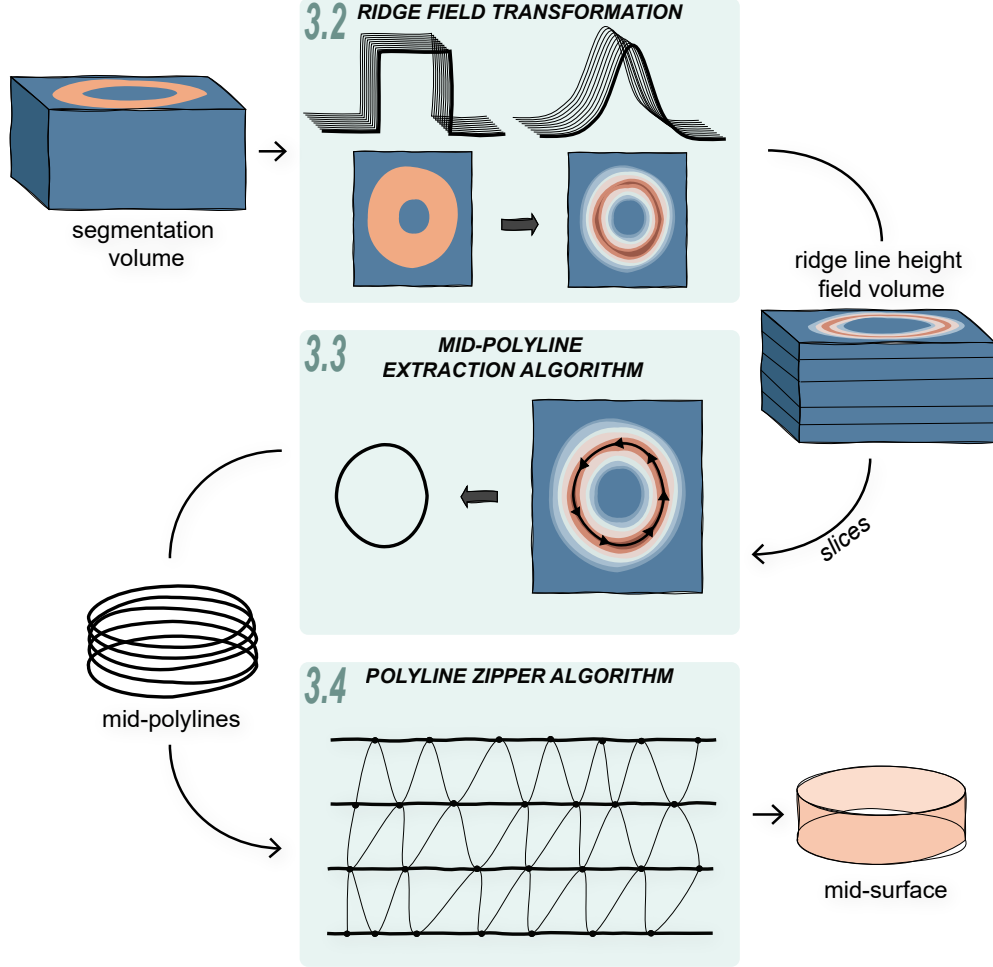


Fig. 3: Schematic overview of the Mid-surface Extraction Algorithm, showing steps from Ridge Field Transformation (Sec. 3.2) that transforms the binary segmentation data; through the Mid-Polyline Extraction Algorithm (Sec. 3.3 that extracts the mid-polylines from slices; to the final mesh generation using the Polyline Zipper Algorithm (Sec. 3.4).

Mid-polyline is defined as a series of straight line segments that lie precisely midway between the inner and outer boundaries of a segmentation, confined to a single slice. This principle underlies the Mid-polyline Extraction Algorithm detailed in Sec. 3.3, serving as its foundational concept.

Mid-surface is conceptualized as a two-dimensional manifold centrally embedded within a three-dimensional structure, equidistant from the structure’s inner and outer surfaces. The transformation from mid-polylines to cohesive mid-surfaces is achieved through our Polyline Zipper Algorithm, streamlining the construction of the desired surfaces.

3.1 Method Overview

Fig. 3 illustrates the overall pipeline, encompassing three key modules detailed in the subsequent subsections. The overarching goal is to accurately extract a mid-surface from volumetric data and represent it as a triangular surface mesh. This mesh is constructed by triangulating a set of horizontally aligned mid-polylines (Sec. 3.4). These mid-polylines are derived from segmented volumetric data (Sec. 3.3). The volumetric data is analyzed slice-by-slice, with each slice yielding mid-polylines at its specific level. When the segmentation within a slice is continuous, it results in a single, seamless mid-polyline. Conversely, discontinuities in the segmentation lead to fragmented mid-polyline segments at that level. To derive a mid-polyline from a given slice, we first convert the binary, cliff-like segmentation into a gradient, hill-range-like height field with a ridge in the center that we term a *ridge line height field* (Sec. 3.2). This conversion process emphasizes symmetry to guarantee the central ridge line accurately delineates the mid-polyline.

The symmetry of the transformation ensures the extracted ridge line precisely embodies the mid-polyline’s path. We initiate the extraction process by identifying the highest point on the ridge line height field, which serves as our starting position. From there, the method proceeds in the ridge’s direction, which aligns with the direction of minimal curvature. This determination is based on the analysis of the Hessian matrix, which compiles the second-order derivatives for each pixel within the slice. By calculating the matrix’s eigenvectors and eigenvalues, we ascertain the primary curvatures, guiding our extraction process along the minimal curvature direction.

3.2 Ridge Field Transformation

To successfully extract the mid-polyline in the following stages, it is imperative to generate a smooth height field featuring a central ridge, known as a ridge line height field. This transformation must be symmetric, ensuring the ridge is precisely positioned midway between the segmentation boundaries. Additionally, it should seamlessly cover the entire breadth of the segmented area, facilitating an accurate and comprehensive extraction of the mid-polyline.

Naïvely, a method that meets these requirements could be a convolution operator that performs a weighted average of surrounding pixels. This is often achieved through Gaussian smoothing, defined as:

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

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) indicate kernel position. Choosing the proper parameters, particularly the standard deviation, is critical for achieving accurate results. The optimal parameters depend on the width of the segmentation. For wider segmentations, larger smoothing values are required to ensure a smooth transition across the entire area, preventing leaving any constant patches. Conversely, narrower segmentations demand smaller values to avoid overly diminishing the features within the resulting height field.

The challenge of manually setting the accurate parameters—and the need to apply a single parameter set across the entire volume, which might contain segmentations of varying widths—prompted us to explore a different approach. Drawing on the understanding that the transformation should depend on the segmentation’s width and the previous requirements for symmetry and a smooth ridge line height field, we utilize a Signed Distance Field (SDF). SDF assigns a distance value to each pixel/voxel relative to the nearest boundary. Mathematically, the SDF can be defined as follows:

$$\text{SDF}(x, y, z) = \min(d_{\text{in}}(x, y, z), d_{\text{out}}(x, y, z)), \quad (2)$$

where, $d_{\text{in}}(x, y, z)$ represents the distance to the inner boundary, while $d_{\text{out}}(x, y, z)$ represents the distance to the outer boundary. This formulation ensures that points lying exactly at the midpoint between the two boundaries have the highest values. Consequently, it yields a ridge line height field where the highest points are located in the middle between the two boundaries, irrespective of their distance (i.e., the thickness of the segmentation).

The SDF accurately identifies the center of the segmentation. However, the resolution of the original segmentation influences the resulting SDF and can introduce a staircasing effect. To counteract this issue, we apply smoothing to the SDF using the Gaussian kernel. However, to keep the approach parameter-free and adapt to the specific dataset in use, the smoothing parameters for this step are also derived from the SDF, where the standard deviation is half of the SDF maximum value $\sigma = \frac{\text{SDF}_{\text{max}}}{2}$ and the size of the kernel is calculated accordingly to reasonably fit the Gaussian curve $\text{radius} = 2\sigma + 1$. Gaussian kernel $G_s(x, y, z)$ in our case is defined as:

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

The final smoothed SDF is obtained by convolving, denoted by $*$, the original SDF (Eq. (2)) with the Gaussian smoothing (Eq. (3)):

$$\text{SDF}_{\text{smoothed}}(x, y, z) = \text{SDF}(x, y, z) * G_s(x, y, z). \quad (4)$$

The intended outcome of this process is a uniform ridge line height field across slices. To achieve this, we apply the procedure to the entire volume, ensuring consistent transitions between slices and eliminating the staircasing effect typically observed in slice-based 2D processing. This means that both the calculation of the SDF and its subsequent smoothing are performed in 3D.

For slicing, we consider the axial orientation of the microscopy data, which comes in a stacked format with an up-and-down direction based on sample preparation. This inherent directionality guides our slicing direction to ensure it aligns with the natural structure and preparation of the sample. The axial slices are known to have the highest resolution due to the limited range of the tilting process during the microscopy data acquisition.

3.3 Mid-Polyline Extraction Algorithm

Having a stack of slices (each in 2D) with a smooth height field, our process begins by computing a line field in each slice. This involves calculating the Hessian matrix \mathbf{H} , which is defined as

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}, \quad (5)$$

where f represents the height field function, and the elements of the Hessian matrix represent the function’s second-order partial derivatives with respect to x and y , measuring the rate of change of the gradient of the field, providing the basis for this calculation.

From these matrices, we identify and extract the eigenvector characterized by

$$\mathbf{H}\mathbf{v} = \lambda\mathbf{v}, \quad (6)$$

corresponding to the smallest eigenvalue denoted by λ_{\min} , with \mathbf{v}_{\min} representing the corresponding eigenvector. 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 guarantees adherence to the ridge, thereby maintaining a consistent path. The eigenvalues and eigenvectors of the Hessian matrix \mathbf{H} are determined by Jacobi iteration [53, Chapter 11.1], ensuring a precise mathematical framework for navigating the ridge line height field.

A ridge aligns with the mid-polyline, delineating the trajectory of the mid-surface. Thus, navigating this ridge through a process informed by the calculated vectors ensures the mid-polyline’s precise alignment with the segmented structure’s inherent geometry. This principle is crucial, laying the groundwork 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, our next step involves the meticulous tracing of the mid-polyline. It is vital to acknowledge that a single slice might feature multiple *hill ranges* rather than a single one. This complexity necessitates the delineation of connected components within the segmentation, as demonstrated in Fig. 4. For each distinct component, the initiation point for mid-polyline tracing is identified by pinpointing the peak of the smoothed height field—the pixel or voxel exhibiting the maximal value.

With a starting point selected and set as the current point \mathbf{r}_i , we perform a computationally cheap Euler integration step to determine the next point $\mathbf{r}_{i+1} = (x_{i+1}, y_{i+1})$ along our mid-polyline calculated as

$$\mathbf{r}_{i+1} = \mathbf{r}_i + h \cdot \mathbf{v}_i, \quad (7)$$

where $h = \sqrt{2} \times \text{spacing}$ is the integration step, *spacing* denotes the distance between slices, effectively defining the separation between mid-polylines, and $\mathbf{v}_i = (x_{v_i}, y_{v_i})$ is the direction of the mid-polyline at the point \mathbf{r}_i , corresponding to the eigenvector with the minimal eigenvalue (see Eq. (6)). We can avoid more accurate but computationally expensive integration methods since at each step of the integration, the position of the computed point is fine-tuned using the golden section search method [53, Chapter 10.1]. If the pixel that we are on is not the pixel with the maximum value

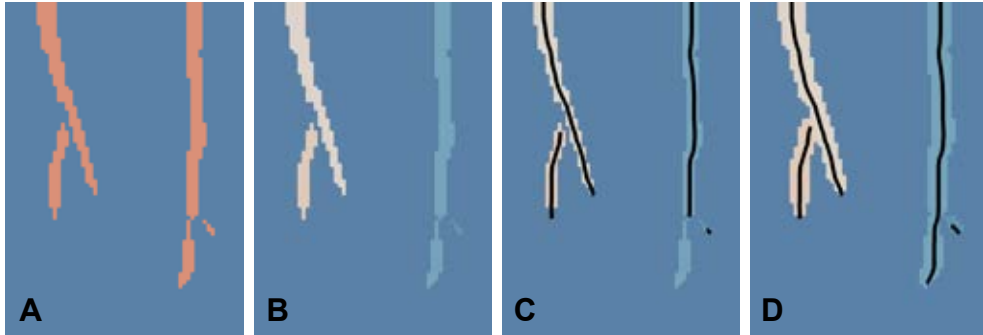


Fig. 4: Visualization of the connected components identification and mid-polyline tracing on a binary mask slice. (A) The original binary mask. (B) The connected components within the binary mask, each assigned a unique ID. (C) Tracing the mid-polyline within these identified connected components. (D)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

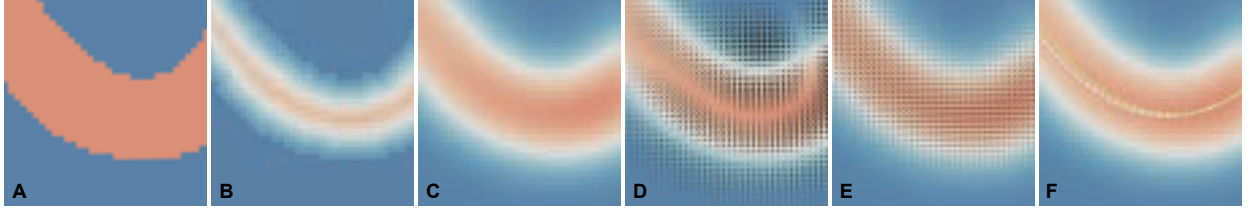


Fig. 5: Visualization of Ridge Field Transformation and Mid-Polyline Extraction Algorithm on a slice: (A) Initial binary segmentation, showcasing the structure of interest. (B) Signed distance field derived from the initial segmentation. (C) Ridge line height field (a smoothed version of the signed distance field to ensure a consistent non-zero derivative across all pixels). (D) Curvature tensor field visualization, derived from the ridge line height field, using superquadric tensor glyphs [52]. (E) Visualization of normalized eigenvectors associated with the smallest eigenvalue, represented as a line field. (F) 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.

along the maximal principal direction (perpendicular to the current direction of the line field), we move our point accordingly, as shown in Fig. 5 (F).

Tracing of the mid-polyline continues until it either completes a loop or exits the connected component’s boundary. When a point exits the connected component and does not make a loop, tracing also proceeds in the opposite direction from the starting point to trace the entire polyline.

A crucial aspect of the tracing process is the treatment of the vector field as a line field, given that our mid-polyline lacks inherent direction. Consequently, vectors are devoid of inherent directionality and are flipped as necessary to ensure continuity. Specifically, we adjust vectors if their orientation deviates by more than 90° from the preceding vector [54].

In cases where the boundary is very thin, the next point in the tracing process might fall outside the boundary, prematurely ending the process. To address this, morphological dilation is applied to the connected components, increasing their width by one pixel. This process enables a more accurate determination of whether an exit from the boundary is genuine or a result of its thinness, as demonstrated in Fig. 4. It is important to note that the dilated slice is used solely to check for the line termination, thus avoiding errors when the boundary is thin. It is not used to calculate the field and, thereby, does not change the morphological characteristics of the segmentation.

In scenarios characterized by intricate topologies or suboptimal segmentations—such as branching structures uncommon in biological data but frequently encountered in low-resolution datasets—an additional verification phase is implemented upon the completion of tracing. This phase assesses whether the tracing comprehensively covered the connected component or if specific sections, like branches, were inadvertently missed. Detected untraced segments are then regarded as independent connected components, prompting a re-initiation of the mid-polyline extraction process for each, employing the same methodology as initially outlined.

Upon thoroughly connecting all points within a single connected component on the slice, the procedure advances to the next component. This progression continues slice by slice, ensuring all components within a slice are addressed before moving to the next. Following the extraction of mid-polylines from every slice, the workflow progresses to the subsequent phase of our method, namely, the triangulation of these extracted mid-polylines (Sec. 3.4). The overall process from segmentation to the extraction of the mid-polyline is illustrated in Fig. 5.

3.4 Polyline Zipper Algorithm

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

In our zipper-like approach, vertices from adjacent slices are systematically connected, traversing across the two neighboring polylines and triangulating edges from both sides, resulting in the formation of triangle strips. This method seamlessly connects all horizontal polylines via vertical edges, ensuring comprehensive connectivity across the resulting triangular surface mesh.

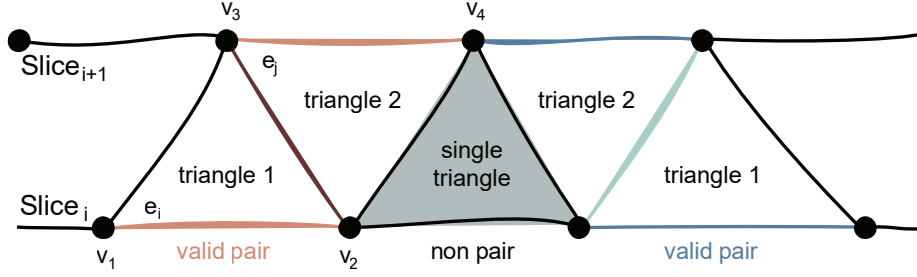


Fig. 6: Zipper approach for connecting edges from two neighboring slices. Valid pairs, exemplified by edges $e_i \in \text{Slice}_i$ and $e_n \in \text{Slice}_{i+1}$, form two triangles (Left). The green-colored triangle represents a single triangle formed by a non-pair edge (center). On the right, the connection of another valid pair of edges is depicted.

The edges contributing to zipper connectivity fall into two categories. 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 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, a valid pair is defined by satisfying the following two conditions:

$$d(e_i, e_j) < d(e_i, e_k), \quad \forall e_k \in S_{i+1}, \quad (8)$$

$$d(e_i, e_j) < d(e_m, e_j), \quad \forall e_m \in S_i. \quad (9)$$

If e_i is found to be closest to $e_j \in S_{i+1}$, but e_j already has another edge $e_m \in S_i$ that is closer than e_i , then e_i is considered a non-pair edge. In other words, if the condition of Eq. (8) is satisfied but not that of Eq. (9), then $e_i \in S_i$ is said to be a non-pair edge. Similarly, if Eq. (9) is satisfied but not Eq. (8), then $e_j \in S_{i+1}$ is a non-pair edge.

The four vertices of v_1, v_2, v_3 , and v_4 of each valid pair of edges are connected to form two triangles (see 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. 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 the non-pair edge $e_i \in S_i$, the valid vertex from the nearest edge $e_j \in S_{i+1}$ is connected to form a single triangle. Moreover, the method takes into account the distance between slices and the absence of edges adjacent to the zipper-like connectivity, which aids in the identification and preservation of holes within the mesh. These unconnected regions are intentional, as they accurately represent voids in the original data, ensuring the mesh’s integrity and fidelity to the segmented structure.

Triangle strip generation progresses iteratively between two consecutive polylines across the full slice stack. Once done, the segmented dataset yields a complete mid-surface mesh. The process of the mesh generation from the input stack of mid-polylines to the final surface mesh is presented in Fig. 9.

Unlike reconstruction algorithms such as simple and scalable surface reconstruction (SSSR) [55] or Delaunay mesh reconstruction [56], our approach directly connects the previously extracted points in their precise locations, thereby mitigating the risk of losing the original input geometry’s fidelity. Moreover, our direct triangulation method distinctively handles potential gaps in the data without the need for specifying extra input parameters, setting it apart from conventional mesh reconstruction techniques. The output mesh not only meets the targeted accuracy requirements but also demonstrates superior mesh quality. Furthermore, this mesh is ready for subsequent refinement processes, including smoothing, simplification, or improvements in mesh quality, according to the specific demands of the application.

4 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 in Sec. 4.1, where we present its capabilities and compare them against those of traditional approaches. Next, we analyzed the mesh quality visually and numerically and compared our mesh quality with the existing method in Sec. 4.2. 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, presented in Sec. 4.3. The proposed algorithm is implemented as ParaView [57] plugin in C++ and tested on Intel(R) Xeon(R) Gold 6230R CPU 2×2.10 GHz with 156 GB RAM and 64 bit Windows 10 operating system.

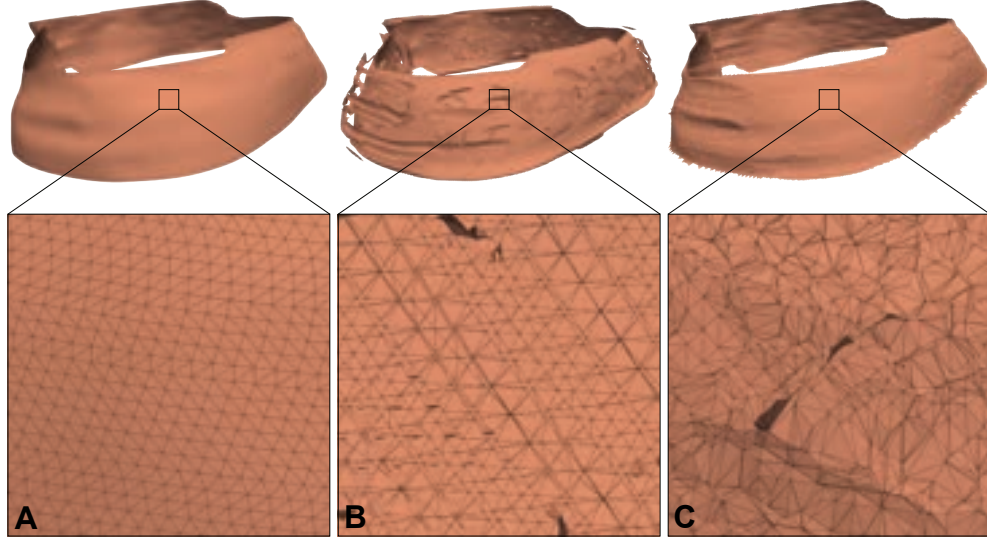


Fig. 7: Evaluation of different surface types highlighting their limitations for our application. (A) The mid-surface as a result of our method. (B) The ridge surface, generated with the VCG ParaView Plugin [59]. (C) The medial surface as produced by the VoxelCores [40].

For our surface generation experiments, we used a well-documented segmentation dataset provided by Klein *et al.* [58], 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 used another dataset from Barad *et al.* [1], 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 [42] 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.

4.1 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 Sec. 2, various methods exist for generating surface meshes, including medial and 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 [59] for ridge surface extraction and the VoxelCores method [40] for the medial surface extraction. The results are shown in Fig. 7, 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. Fig. 8 shows the results of applying the Mid-Polyline Extraction Algorithm to slices from 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. 1 and 9 show the results of two instances of the SARS-CoV-2 dataset [58], along with the intermediate results of mid-polylines. The surface meshes generated from the overall dataset [58] 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. 1, 7 and 9. 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

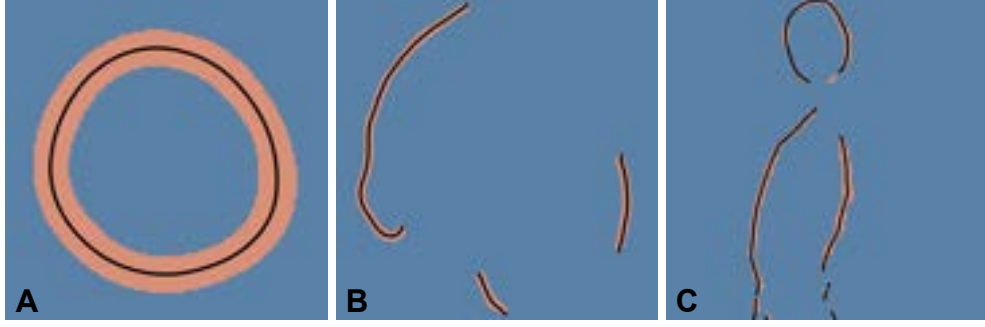


Fig. 8: Performance of the Mid-Polyline Extraction Algorithm across slices featuring varied topologies. (A) A circular segmentation, illustrating the algorithm’s efficacy in linking connected loops. (B) A slice containing multiple components, resulting in the generation of several mid-polylines. (C) A complex segmentation with notably thinner boundaries. The segmentations were sourced from datasets provided by Klein *et al.* [58] (A, B) and Barad *et al.* [1] (C).

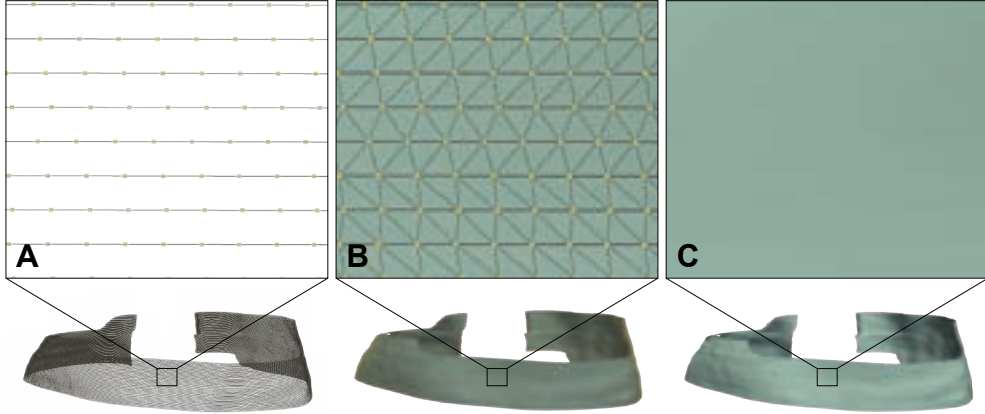


Fig. 9: Mesh generation results. (A) Mid-polylines and the points extracted from slices. (B) The triangle mesh, a result of the triangulation process with new edges connecting points between the mid-polylines. (C) The final surface mesh without visible edges to show visual clarity.

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.

4.2 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 [60], we evaluated our mesh using the *triangle quality* (Q), whose value varies from 0 (poor-quality) to 1 (good quality) and is calculated as:

$$Q = \frac{6}{\sqrt{3}} \frac{A}{(p \times h)}, \quad (10)$$

where, 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 θ_{\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 [61]. The optimal valence meshes are called regular meshes that are easily remeshed to improve other quality metrics.

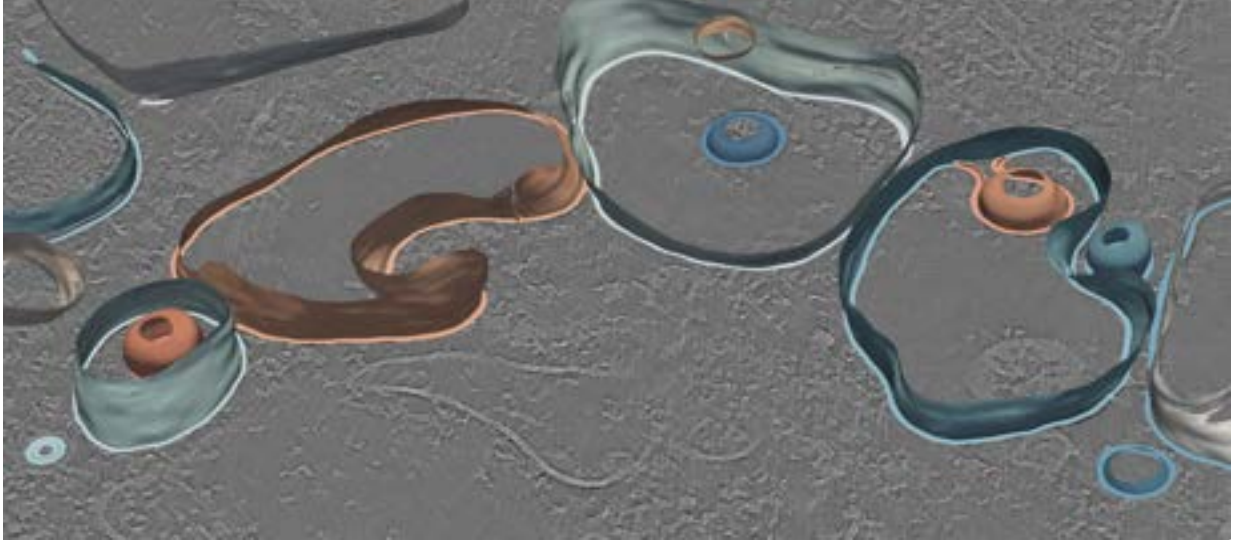


Fig. 10: Mid-surfaces generated from the SARS-CoV-2 dataset [58] overlaid on the original tomogram slice and the segmentation mask.

Mesh quality results: Tab. 1 presents the numerical mesh quality results, which show our meshes are highly regular with a higher ratio of v_{567} vertices. The ratio of poor triangles (lower value of Q , triangles with small or large angles) is also very small. Let’s look into the mesh quality metrics of the existing recently proposed method in the same domain for mid-surface meshes i.e., the PSR [43] based approach developed by Barad *et al.* [1]. 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 [1]. The generation of a high-quality mesh is 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, Tab. 1 includes a time analysis, indicating that our method demonstrates longer processing times compared to that of Barad *et al.* [1], 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.

Table 1: Quantitative results of mesh quality (PSR* [1] vs. Ours). Q represents triangle quality calculated via Eq. (10). θ_{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}	θ_{min}	$\theta < 30^\circ$	$\theta > 120^\circ$	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. 9	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 %	0.0 %	97 %	2 + 1
Ours	Fig. 13 (B)	39796	0.04	0.81	44.0	0.02 %	0.01 %	97 %	4 + 4

4.3 Use Case Applications

Mid-surface, once extracted, can be utilized in many important 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 [1] was recently developed to quantify the ultrastructure of biological membranes, including membrane curvature, orientation, and inter-membrane spacing, using mid-surface

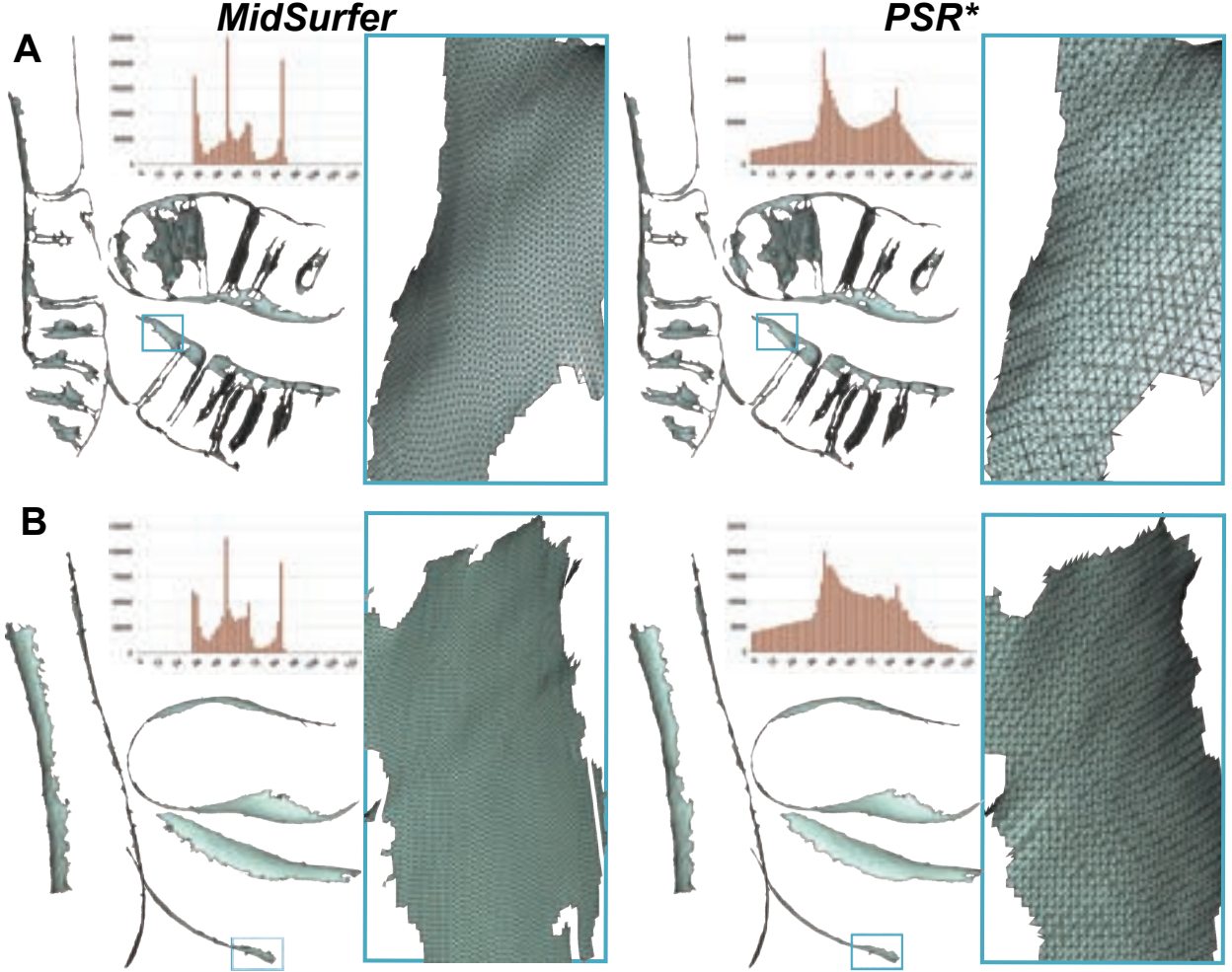


Fig. 11: Triangulation results: Left: Ours, Right: *PSR**, the PSR [43] based approach developed by Barad *et al.* [1]. The top row shows TE7-IMM, and the bottom row depicts TE7-OMM. In the angle histogram, the x-axis indicates angle measurements in degrees, while the y-axis shows the frequency of angles for each value. We observe that our method yields a higher ratio of angles near 60 degrees. The quantitative results are shown in Tab. 1.

models. For this purpose, they utilized the screened PSR method [43] to generate mid-surface meshes, which the toolkit [1] 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.* [1].

Fig. 12 illustrates the results of these quantifications for mid-surfaces generated using MidSurfer compared to those generated by the Screened PSR-based method [43, 1]. 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 [1] requires expert users to fine-tune the mid-surface extraction parameters. These results were achieved after the authors of the technique [1] 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 [1].

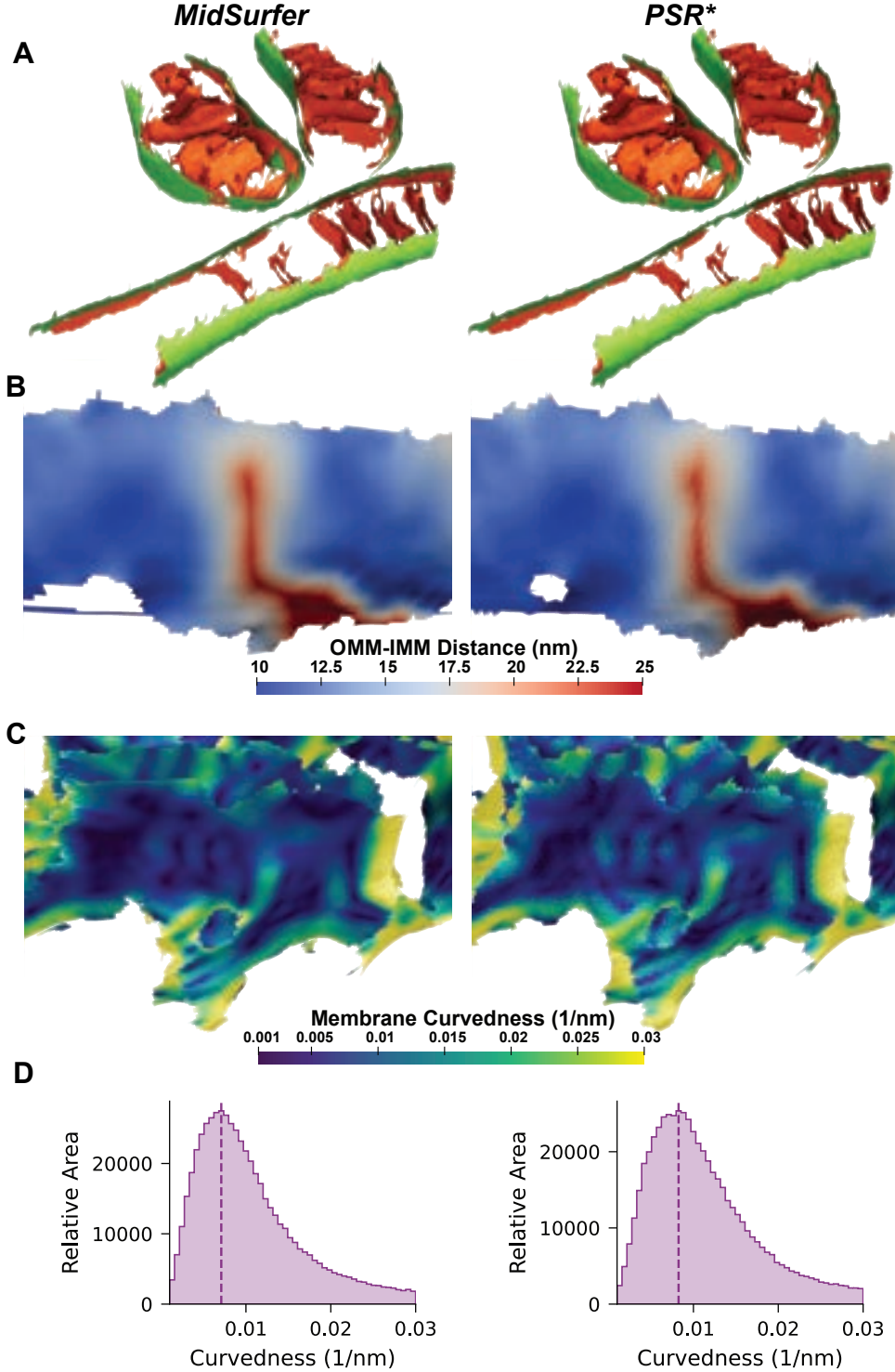


Fig. 12: Morphometric comparison of membrane surfaces generated with MidSurfer vs. Screened PSR. (A) Surface meshes were generated from segmentations depicting mitochondrial inner and outer membranes observed by cryo-ET, then quantified with the Surface Morphometrics toolkit [1]. Visual comparison shows 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 [43] based approach developed by Barad *et al.* [1].

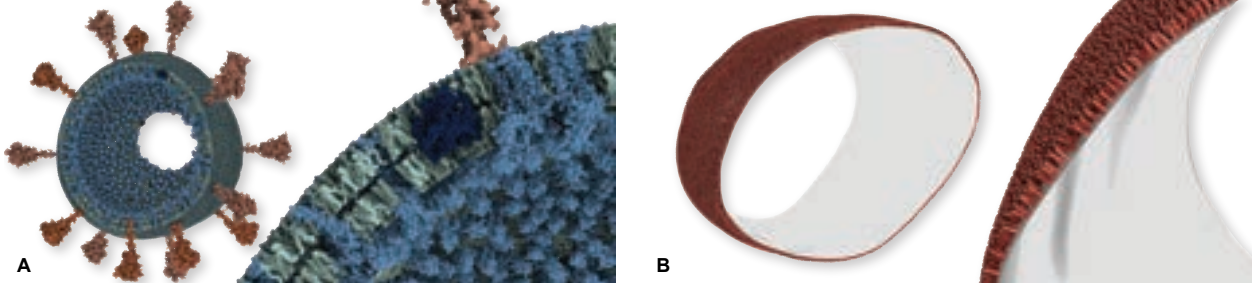


Fig. 13: Modeling results with MesoCraft [62]. (A) SARS-CoV-2 virion with the lipids of the lipid bilayer membrane placed on both sides of the extracted mid-surface. (B) Lipids of the lipid bilayer membrane only populated on the outer side of the surface.

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 [62]. 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.* [58] 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.

5 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.

Our research further solidifies the choice of mid-surfaces over more commonly utilized surface representations, a conclusion we show in Fig. 7. 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, these methods’ generation of irregular meshes 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.

As demonstrated in Fig. 8, our method for mid-polyline extraction shows consistent robustness across various datasets, with segmentation thickness and topological complexity being the primary variables. The key to achieving robustness despite varying segmentation thickness lies in our approach of employing the Signed Distance Field (SDF) for obtaining the ridge line height field as opposed to other smoothing functions.

Our approach to triangulation enables us to generate well-formed triangles consistently since it progresses through the volume slice by slice, yielding the resulting triangles with the same height. By choosing an appropriate size for the integration step in mid-polyline extraction, we can achieve nearly equilateral triangles. Furthermore, by imposing a deliberate limitation on the edge lengths within our triangulation process, we adeptly avoid the unintentional closure of gaps present in the original segmentation data. These gaps, often critical for the structural and functional understanding of biological systems, are preserved, thereby maintaining the correctness of the biological model.

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 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.

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 [8]. These results broadly show that MidSurfer models are appropriate for detailed quantitative analysis of biological membrane ultrastructure.

One of the key features integrated into our mid-surface extraction algorithm by design is its inherent capability for parallelization. While the current implementation operates sequentially, the algorithm’s structure allows for the future development of a parallelized version that is expected to accelerate processing speeds significantly. This design consideration is vital for efficiently handling the extensive volumetric datasets commonly encountered in biological research and, with it, enhancing the algorithm’s practicality for research applications. We have developed and released a plugin to improve the method’s accessibility and usability. This plugin, which integrates with ParaView [57]—a widely-used platform for data analysis and visualization—makes it easier for researchers to incorporate our mid-surface extraction technique into their existing workflows. By making our algorithm available as a plugin, we eliminate the need for users to navigate through multiple toolboxes, thereby simplifying the research process. This effort to enhance accessibility is aimed at encouraging broader adoption of our method across different scientific fields, extending its applicability beyond its initial focus on biological analysis and modeling.

The parameter-free mid-surface reconstructions we generate are able to recapitulate the results obtained by the previous best-in-class screened PSR approach with parameters optimized by domain experts. Bypassing the need for parameter optimization will increase the accessibility of midsurface-based analysis approaches for microscopists collecting and segmenting biological membranes since poor parameter selection in previous approaches can result in surfaces that are not suitable for quantification. Beyond accessibility, the increased robustness of the results with variables such as segmentation thickness will improve the degree of automation of pipelines using membrane mid-surface models as a critical step for analysis.

6 Conclusion

In this work, we have 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 [1], utilizing screened Poisson Surface Reconstruction [43]. 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 [1] 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 through the processing of medial surfaces—by pruning branches—or ridge surfaces—by removing disconnected components—and then applying smoothing and remeshing techniques, such an approach would be significantly more laborious and less efficient compared to our streamlined *one-click* solution.

Admittedly, the success of our method is contingent 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 extraction 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—our cutting-edge mid-surface extraction algorithm—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.

Acknowledgment

This research was supported by King Abdullah University of Science and Technology (KAUST) (BAS/1/1680-01-01), the KAUST Visualization Core Lab, and the Nadia’s Gift Foundation Innovator Award from the Damon Runyon Cancer Foundation (DRR-65-21 to D.A.G.).

References

- [1] Benjamin Asher Barad, Michaela Medina, Daniel Fuentes, R. Luke Wiseman, and Danielle Ann Grotjahn. Quantifying organellar ultrastructure in cryo-electron tomography using a surface morphometrics pipeline. *The Journal of Cell Biology*, 222, 2023. doi:10.1083/jcb.202204093.
- [2] Mohsen Rezayat. Midsurface abstraction from 3D solid models: general theory and applications. *Computer-Aided Design*, 28(11):905–915, 1996.
- [3] Yogesh H Kulkarni, Anil Sahasrabudhe, and Mukund Kale. Leveraging feature generalization and decomposition to compute a well-connected midsurface. *Engineering with Computers*, 33(1):159–170, 2017.
- [4] Mark A Price and Cecil G Armstrong. Hexahedral mesh generation by medial surface subdivision: Part II. Solids with flat and concave edges. *International Journal for Numerical Methods in Engineering*, 40(1):111–136, 1997.
- [5] E. Keppel. Approximating Complex Surfaces by Triangulation of Contour Lines. *IBM Journal of Research and Development*, 19(1):2–11, 1975. doi:10.1147/rd.191.0002.
- [6] Xin Feng, Kelin Xia, Y. Tong, and Guowei Wei. Geometric modeling of subcellular structures, organelles, and multiprotein complexes. *International Journal for Numerical Methods in Biomedical Engineering*, 28, 2012. URL <https://api.semanticscholar.org/CorpusID:15547735>.
- [7] Norbert Lindow, Daniel Baum, and Hans-Christian Hege. Ligand Excluded Surface: A New Type of Molecular Surface. *IEEE Trans. on Vis. and Comp. Graphics*, 20(12):2486–2495, 2014. doi:10.1109/TVCG.2014.2346404.
- [8] Maria Salfer, Javier Collado, Wolfgang Baumeister, Rubén Fernández-Busnadiego, and Antonio Martínez-Sánchez. Reliable estimation of membrane curvature for cryo-electron tomography. *PLOS Computational Biology*, 16:e1007962, 08 2020. doi:10.1371/journal.pcbi.1007962.
- [9] Tiantian Liu, Minxin Chen, and Benzhuo Lu. Parameterization for molecular Gaussian surface and a comparison study of surface mesh generation. *Journal of Molecular Modeling*, 21(5):113, April 2015. ISSN 0948-5023. doi:10.1007/s00894-015-2654-9. URL <https://doi.org/10.1007/s00894-015-2654-9>.
- [10] Mohsen Sadeghi, Thomas R. Weikl, and Frank Noé. Particle-based membrane model for mesoscopic simulation of cellular dynamics. *The Journal of Chemical Physics*, 148(4):044901, 01 2018. ISSN 0021-9606. doi:10.1063/1.5009107. URL <https://doi.org/10.1063/1.5009107>.
- [11] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 26(2):71–78, jul 1992. ISSN 0097-8930. doi:10.1145/142920.134011. URL <https://doi.org/10.1145/142920.134011>.
- [12] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Mesh optimization. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 19–26, 1993.
- [13] Jean-Daniel Boissonnat and Frédéric Cazals. Smooth surface reconstruction via natural neighbour interpolation of distance functions. In *Proceedings of the sixteenth annual symposium on Computational geometry*, pages 223–232, 2000.
- [14] M. Pauly, M. Gross, and L.P. Kobbelt. Efficient simplification of point-sampled surfaces. In *IEEE Visualization, 2002.*, pages 163–170, 2002. doi:10.1109/VISUAL.2002.1183771.
- [15] Mark Pauly, Richard Keiser, Leif P. Kobbelt, and Markus Gross. Shape modeling with point-sampled geometry. *ACM Trans. Graph.*, 22(3):641–650, jul 2003. ISSN 0730-0301. doi:10.1145/882262.882319. URL <https://doi.org/10.1145/882262.882319>.
- [16] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C.T. Silva. Point set surfaces. In *IEEE Visualization, 2001.*, pages 21–29, 537, 2001. doi:10.1109/VISUAL.2001.964489.
- [17] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C.T. Silva. Computing and rendering point set surfaces. *IEEE Trans. on Vis. and Comp. Graphics*, 9(1):3–15, 2003. doi:10.1109/TVCG.2003.1175093.

- [18] David Levin. Mesh-independent surface interpolation. In *Geometric modeling for scientific visualization*, pages 37–49. Springer, 2004.
- [19] Nina Amenta and Yong Joo Kil. Defining point-set surfaces. *ACM Trans. Graph.*, 23(3):264–270, aug 2004. ISSN 0730-0301. doi:10.1145/1015706.1015713. URL <https://doi.org/10.1145/1015706.1015713>.
- [20] Shachar Fleishman, Daniel Cohen-Or, and Cláudio T. Silva. Robust moving least-squares fitting with sharp features. *ACM Trans. Graph.*, 24(3):544–552, jul 2005. ISSN 0730-0301. doi:10.1145/1073204.1073227. URL <https://doi.org/10.1145/1073204.1073227>.
- [21] Gaël Guennebaud and Markus Gross. Algebraic point set surfaces. *ACM Trans. Graph.*, 26(3):23–es, jul 2007. ISSN 0730-0301. doi:10.1145/1276377.1276406. URL <https://doi.org/10.1145/1276377.1276406>.
- [22] A. C. Öztireli, G. Guennebaud, and M. Gross. Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression. *Computer Graphics Forum*, 28(2):493–501, 2009. doi:<https://doi.org/10.1111/j.1467-8659.2009.01388.x>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2009.01388.x>.
- [23] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-level partition of unity implicits. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH ’05, page 173–es, New York, NY, USA, 2005. ISBN 9781450378338. doi:10.1145/1198555.1198649. URL <https://doi.org/10.1145/1198555.1198649>.
- [24] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep Geometric Prior for Surface Reconstruction. In *IEEE CVPR*, June 2019.
- [25] Alexandre Boulch and Renaud Marlet. POCO: Point Convolution for Surface Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6302–6314, June 2022.
- [26] Matthew Berger, Andrea Tagliasacchi, Lee M. Seversky, Pierre Alliez, Gaël Guennebaud, Joshua A. Levine, Andrei Sharf, and Claudio T. Silva. A Survey of Surface Reconstruction from Point Clouds. *Computer Graphics Forum*, 36(1):301–329, 2017. doi:<https://doi.org/10.1111/cgf.12802>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12802>.
- [27] Zhangjin Huang, Yuxin Wen, Zihao Wang, Jinjuan Ren, and Kui Jia. Surface Reconstruction from Point Clouds: A Survey and a Benchmark, 2022.
- [28] Harry Blum. A Transformation for Extracting New Descriptors of Shape. In Weiant Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, 1967.
- [29] William E. Lorensen and Harvey E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *ACM Trans. Graphics*, 21(4):163–169, August 1987. ISSN 0097-8930. doi:10.1145/37402.37422. URL <https://doi.org/10.1145/37402.37422>.
- [30] Akio Doi and Akio Koide. An efficient method of triangulating equi-valued surfaces by using tetrahedral cells. *IEICE TRANSACTIONS on Information and Systems*, 74(1):214–224, 1991.
- [31] G.M. Treece, R.W. Prager, and A.H. Gee. Regularised marching tetrahedra: improved iso-surface extraction. *Computers & Graphics*, 23(4):583–598, 1999. ISSN 0097-8493. doi:[https://doi.org/10.1016/S0097-8493\(99\)00076-X](https://doi.org/10.1016/S0097-8493(99)00076-X). URL <https://www.sciencedirect.com/science/article/pii/S009784939900076X>.
- [32] A. Gueziec and R. Hummel. Exploiting triangulated surface extraction using tetrahedral decomposition. *IEEE Trans. on Vis. and Comp. Graphics*, 1(4):328–342, 1995. doi:10.1109/2945.485620.
- [33] Y. Livnat, Han-Wei Shen, and C.R. Johnson. A near optimal isosurface extraction algorithm using the span space. *IEEE Trans. on Vis. and Comp. Graphics*, 2(1):73–84, 1996. doi:10.1109/2945.489388.
- [34] Z.J. Wood, M. Desbrun, P. Schroder, and D. Breen. Semi-regular mesh extraction from volumes. In *Proceedings Visualization 2000. VIS 2000 (Cat. No.00CH37145)*, pages 275–282, 2000. doi:10.1109/VISUAL.2000.885705.
- [35] Leif P. Kobbelt, Mario Botsch, Ulrich Schwanke, and Hans-Peter Seidel. Feature sensitive surface extraction from volume data. In *Proc. ACM SIGGRAPH*, SIGGRAPH ’01, page 57–66, New York, NY, USA, 2001. ISBN 158113374X. doi:10.1145/383259.383265. URL <https://doi.org/10.1145/383259.383265>.
- [36] Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. Dual contouring of hermite data. *ACM Trans. Graph.*, 21(3):339–346, jul 2002. ISSN 0730-0301. doi:10.1145/566654.566586. URL <https://doi.org/10.1145/566654.566586>.
- [37] June Sic Kim, Vivek Singh, Jun Ki Lee, Jason Lerch, Yasser Ad-Dab’bagh, David MacDonald, Jong Min Lee, Sun I. Kim, and Alan C. Evans. Automated 3-D extraction and evaluation of the inner and outer cortical surfaces using a Laplacian map and partial volume effect classification. *NeuroImage*, 27(1):210–221, 2005. ISSN 1053-8119. doi:<https://doi.org/10.1016/j.neuroimage.2005.03.036>. URL <https://www.sciencedirect.com/science/article/pii/S1053811905002235>.

- [38] Edoardo Remelli, Artem Lukoianov, Stephan Richter, Benoit Guillard, Timur Bagautdinov, Pierre Baque, and Pascal Fua. MeshSDF: Differentiable Iso-Surface Extraction. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22468–22478. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/fe40fb944ee700392ed51bfe84dd4e3d-Paper.pdf.
- [39] Hau Kit Yong, Leran Wang, David JJ Toal, Andy J Keane, and Felix Stanley. Multi-fidelity Kriging-assisted structural optimization of whole engine models employing medial meshes. *Structural and Multidisciplinary Optimization*, 60:1209–1226, 2019.
- [40] Yajie Yan, David Letscher, and Tao Ju. Voxel cores: efficient, robust, and provably good approximation of 3D medial axes. *ACM Trans. Graph.*, 37(4), jul 2018. ISSN 0730-0301. doi:10.1145/3197517.3201396. URL <https://doi.org/10.1145/3197517.3201396>.
- [41] Yoonhwan Woo. Abstraction of mid-surfaces from solid models of thin-walled parts: A divide-and-conquer approach. *Computer-Aided Design*, 47:1–11, 2014. ISSN 0010-4485. doi:<https://doi.org/10.1016/j.cad.2013.08.010>. URL <https://www.sciencedirect.com/science/article/pii/S0010448513001504>.
- [42] Antonio Martinez-Sanchez, Inmaculada Garcia, Shoh Asano, Vladan Lucic, and Jose-Jesus Fernandez. Robust membrane detection based on tensor voting for electron tomography. *Journal of Structural Biology*, 186(1):49–61, 2014. ISSN 1047-8477. doi:<https://doi.org/10.1016/j.jsb.2014.02.015>. URL <https://www.sciencedirect.com/science/article/pii/S1047847714000495>.
- [43] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3), jul 2013. ISSN 0730-0301. doi:10.1145/2487228.2487237. URL <https://doi.org/10.1145/2487228.2487237>.
- [44] H. Fuchs, Z. M. Kedem, and S. P. Uselton. Optimal surface reconstruction from planar contours. *Commun. ACM*, 20(10):693–702, oct 1977. ISSN 0001-0782. doi:10.1145/359842.359846. URL <https://doi.org/10.1145/359842.359846>.
- [45] H. N. Christiansen and T. W. Sederberg. Conversion of complex contour line definitions into polygonal element mosaics. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 12(3):187–192, aug 1978. ISSN 0097-8930. doi:10.1145/965139.807388. URL <https://doi.org/10.1145/965139.807388>.
- [46] Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 311–318, 1994.
- [47] Topraj Gurung, Mark Luffel, Peter Lindstrom, and Jarek Rossignac. Zipper: A compact connectivity data structure for triangle meshes. *Computer-Aided Design*, 45(2):262–269, 2013. ISSN 0010-4485. doi:<https://doi.org/10.1016/j.cad.2012.10.009>. URL <https://www.sciencedirect.com/science/article/pii/S001044851200214X>. Solid and Physical Modeling 2012.
- [48] J.D. Furst, S.M. Pizer, and D.H. Eberly. Marching cores: a method for extracting cores from 3d medical images. In *Proceedings of the Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 124–130, 1996. doi:10.1109/MMBIA.1996.534064.
- [49] Jacob D Furst and Stephen M Pizer. Marching ridges. In *SIP*, volume 1, pages 22–26, 2001.
- [50] Filip Sadlo and Ronald Peikert. Visualizing Lagrangian Coherent Structures and Comparison to Vector Field Topology. In Hans-Christian Hege, Konrad Polthier, and Gerik Scheuermann, editors, *Topology-Based Methods in Visualization II*, pages 15–29. Springer Berlin Heidelberg, 2009. ISBN 978-3-540-88606-8. doi:10.1007/978-3-540-88606-8_2. URL https://doi.org/10.1007/978-3-540-88606-8_2.
- [51] Gordon Kindlmann, Xavier Tricoche, and Carl-Fredrik Westin. Anisotropy Creases Delineate White Matter Structure in Diffusion Tensor MRI. In Rasmus Larsen, Mads Nielsen, and Jon Sporring, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006*, pages 126–133. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-44708-5.
- [52] Gordon Kindlmann. Superquadric Tensor Glyphs. In Oliver Deussen, Charles Hansen, Daniel Keim, and Dietmar Saupe, editors, *Eurographics / IEEE VGTC Symposium on Visualization*. The Eurographics Association, 2004. ISBN 3-905673-07-X. doi:10.2312/VisSym/VisSym04/147-154.
- [53] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C (2nd ed.): the art of scientific computing*. Cambridge University Press, USA, 1992. ISBN 0521431085.
- [54] Werner Benger and Hans-Christian Hege. Strategies for Direct Visualization of Second-Rank Tensor Fields. In Joachim Weickert and Hans Hagen, editors, *Visualization and Processing of Tensor Fields*, pages 191–214. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-31272-7. doi:10.1007/3-540-31272-2_11. URL https://doi.org/10.1007/3-540-31272-2_11.

- [55] Dobrina Boltcheva and Bruno Lévy. Simple and Scalable Surface Reconstruction. Research report, LORIA - Université de Lorraine ; INRIA Nancy, July 2016. URL <https://hal.science/hal-01349023>.
- [56] Wenyong Gong, Yong-Jin Liu, Kai Tang, and Tieru Wu. Approximate Delaunay mesh reconstruction and quality estimation from point samples. *Journal of Computational and Applied Mathematics*, 274:23–34, 2015. ISSN 0377-0427. doi:<https://doi.org/10.1016/j.cam.2014.07.001>. URL <https://www.sciencedirect.com/science/article/pii/S0377042714003082>.
- [57] James Ahrens, Berk Geveci, and Charles Law. ParaView: An End-User Tool for Large Data Visualization. In *Visualization Handbook*. Elsevier, 2005. ISBN 978-0123875822.
- [58] Steffen Klein, Mirko Cortese, Sophie L. Winter, Moritz Wachsmuth-Melm, Christopher J. Neufeldt, Berati Cerikan, Megan L. Stanifer, Steeve Boulant, Ralf Bartenschlager, and Petr Chlanda. SARS-CoV-2 structure and replication characterized by in situ cryo-electron tomography. *Nature Communications*, 11(1):5885, 2020. ISSN 2041-1723. doi:[10.1038/s41467-020-19619-7](https://doi.org/10.1038/s41467-020-19619-7). URL <https://doi.org/10.1038/s41467-020-19619-7>.
- [59] Filip Sadlo, Philip Hausner, Lutz Hofmann, Philipp Jung, Grzegorz Karch, Ronald Peikert, Lukas Pilz, Martin Roth, and Kai Sdeo. VCG ParaView plugins. <https://vcg.iwr.uni-heidelberg.de/plugins/>, March 2019.
- [60] Dawar Khan, Alexander Plopski, Yuichiro Fujimoto, Masayuki Kanbara, Gul Jabeen, Yongjie Jessica Zhang, Xiaopeng Zhang, and Hirokazu Kato. Surface Remeshing: A Systematic Literature Review of Methods and Research Directions. *IEEE Trans. on Vis. and Comp. Graphics*, 28(3):1680–1713, 2022.
- [61] Nima Aghdaii, Hamid Younesy, and Hao Zhang. 5-6-7 Meshes: Remeshing and analysis. *Computers & Graphics*, 36(8):1072–1083, 2012. ISSN 0097-8493. doi:<https://doi.org/10.1016/j.cag.2012.09.005>. URL <https://www.sciencedirect.com/science/article/pii/S0097849312001574>. Graphics Interaction Virtual Environments and Applications 2012.
- [62] Ngan Nguyen, Ondrej Strnad, Tobias Klein, Deng Luo, Ruwayda Alharbi, Peter Wonka, Martina Maritan, Peter Mindek, Ludovic Autin, David S. Goodsell, and Ivan Viola. Modeling in the Time of COVID-19: Statistical and Rule-based Mesoscale Models. *IEEE Trans. on Vis. and Comp. Graphics*, 2020. doi:[10.1109/TVCG.2020.3030415](https://doi.org/10.1109/TVCG.2020.3030415).