Level Set Grids for Hybrid Manufacturing

Liam White[1,2], Bryan Quaife[2], Michael Borish[1], Cameron Adkins[1], Alex Roschli[1]
[1]Oak Ridge National Laboratory, Manufacturing Science Division
[2]Florida State University, Department of Scientific Computing

This material is based upon work supported by the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, Office of Advanced Manufacturing, under contract number DE-AC05-00OR22725.

Abstract

Hybrid manufacturing merges the design flexibility of additive manufacturing with the precision finishing of subtractive processes. Despite its innovative potential, the sequential nature of current hybrid manufacturing workflows presents significant efficiency challenges. This work introduces the Level Set Grid, a novel representation aimed at enhancing hybrid manufacturing by enabling the parallel execution of additive and subtractive processes.

The Level Set Grid innovatively combines explicit and implicit geometric representations to facilitate the precise modeling of objects and efficient collision detection. By leveraging a level set function, typically a signed distance function, the model offers a robust framework for representing complex geometries. This function's evaluations populate a sparse volumetric grid, blending the accuracy of implicit models with the practicality of explicit representations.

Preliminary findings underscore the model's capability to integrate seamlessly with existing manufacturing protocols, thereby facilitating the generation of additive slicing and subtractive toolpathing. This work outlines the foundational steps toward the model's application in hybrid manufacturing, highlighting its potential to revolutionize the field by improving process efficiency, flexibility, and innovation.

Grid Construction

Using OpenVDB, a Level Set Grid is constructed from a tessellated mesh with a desired voxel size and narrow band width.

A sparse volumetric grid is generated to encompass the mesh (Figure 1). Voxels within the narrow band are classified by ray casting based on their position—inside, outside, or on the object's surface—determined by the number of ray intersections with the mesh (Figure 2).

The fast-marching method computes and stores the shortest distance from voxels to the object surface as signed values—negative for inside and non-negative for outside or on the surface (Figure 2). This signed distance field is refined by solving the Eikonal equation over the grid to accurately represent the object's geometry.

Voxelization involves a trade-off: smaller voxels increase geometric accuracy but require more memory and processing power, a crucial factor in constructing a Level Set Grid.

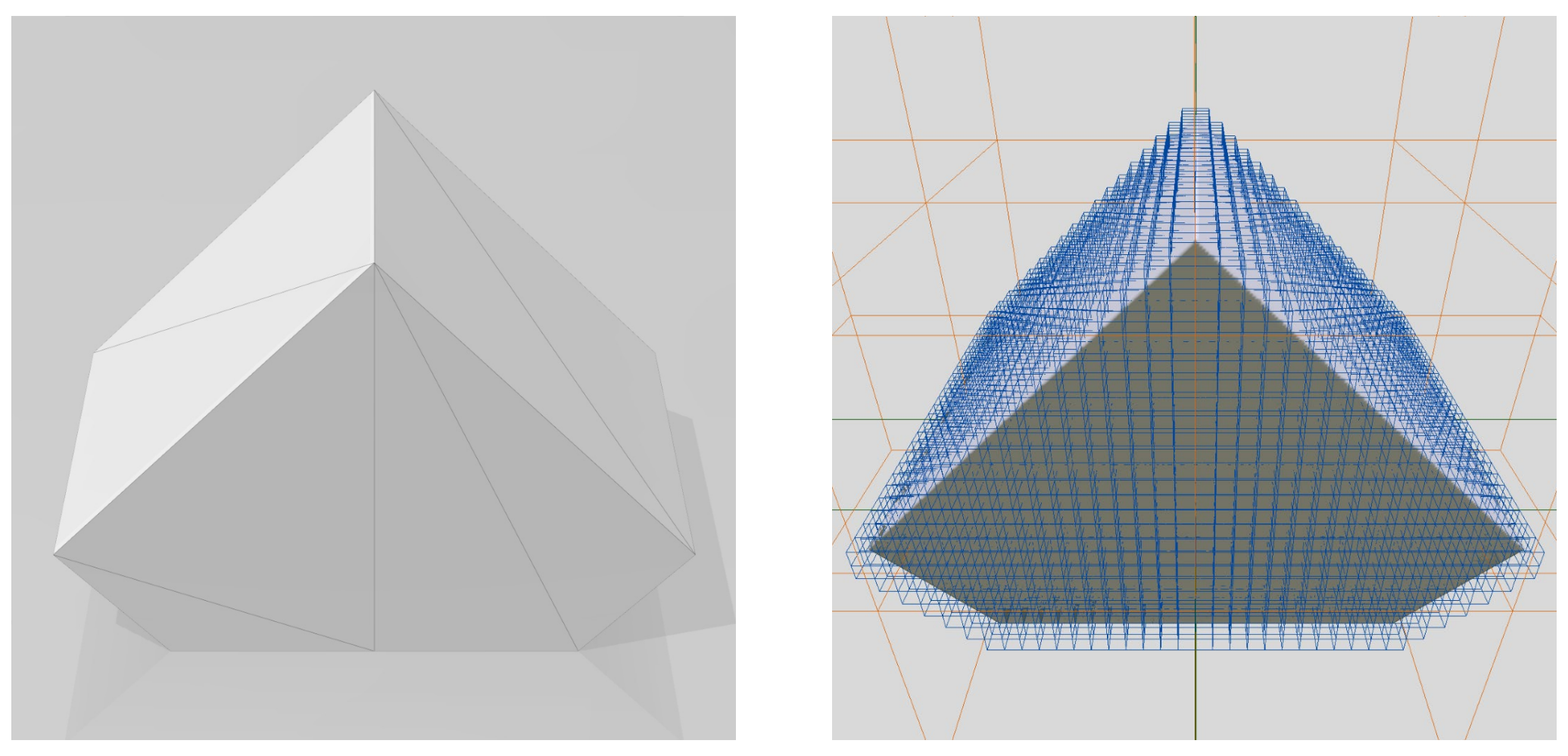

Figure 1. A tessellated mesh representation of an object (left) alongside a voxelized representation of the same object (right).

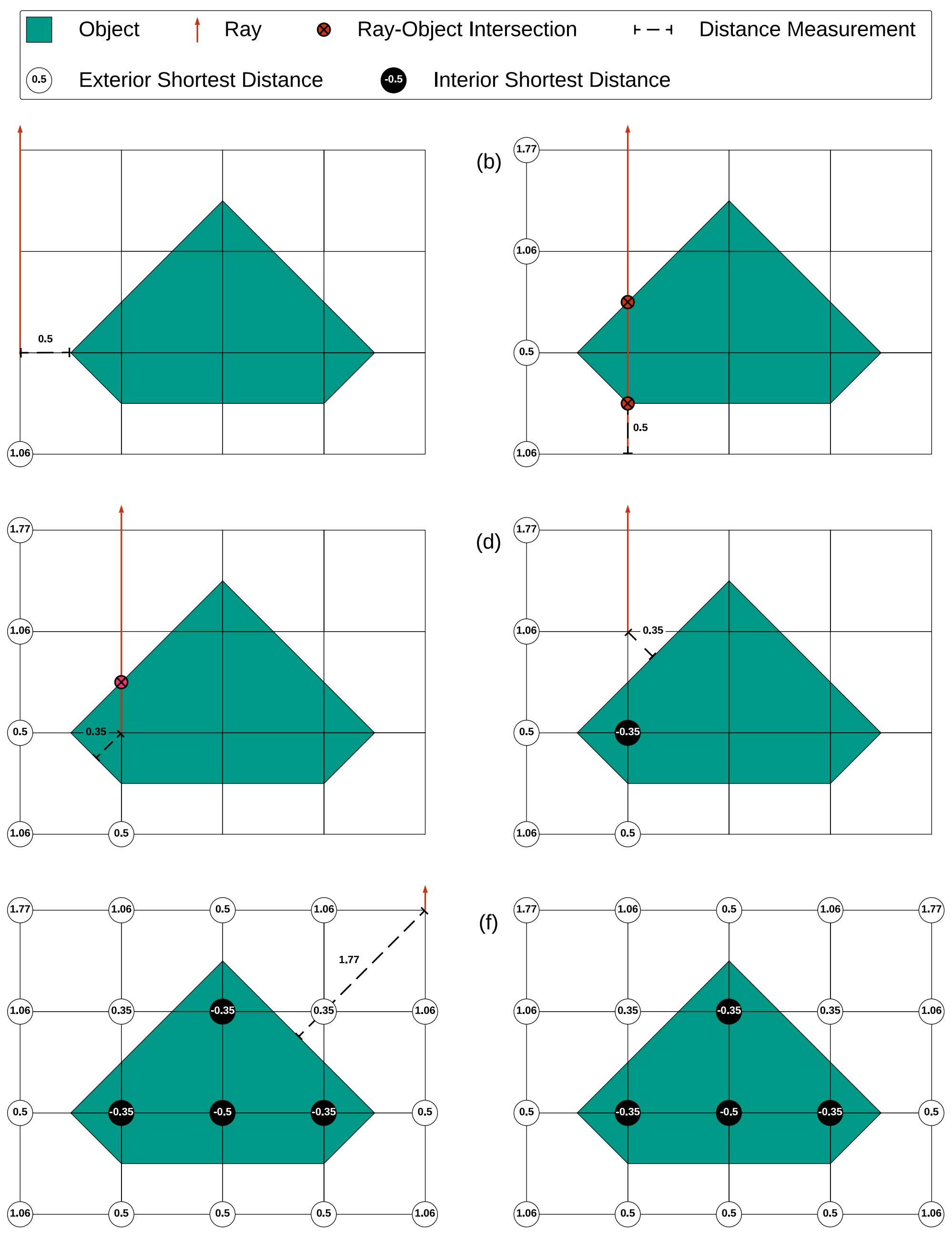

Figure 2. Detailed process of the signed distance field generation for a 2D object. (a-c) Rays cast from voxels intersect the object boundary zero, two, and one times, dictating the sign of the corresponding distance value. (d & e) Iterative steps showcasing the assignment of signed distances to all voxels. (f) The completed signed distance field of the object.

Slicing

Slicing converts 3D models into print-ready layers, involving creating polygonal geometries and their fill paths. With Level Set Grids, this entails two steps: Cross-sectioning with OpenVDB's 'clip' function for a 2D voxel grid, and Contour Extraction using a modified marching squares algorithm for contours. These steps produce polygonal layers for additive manufacturing.

Using OpenVDB's 'clip' function, we obtain a Level Set Grid's planar cross-section by clipping the grid with an axis-aligned bounding box, adjusted to match the target plane. This produces a 2D grid of voxels, from which we extract polygonal contours (Figure 3a).

A modified marching squares algorithm extracts contours from Level Set Grid cross-sections, ensuring consistent orientation for easy conversion to polygons with holes. This process scans 2D grid cells, using a binary index based on node distances to define cell edges (Figure 3b). Bilinear interpolation identifies segment endpoints, maintaining interior object orientation to the right for consistent contour winding (Figure 3c). The algorithm iteratively generates segments, marking visited cells, until a contour closes (Figure 4).

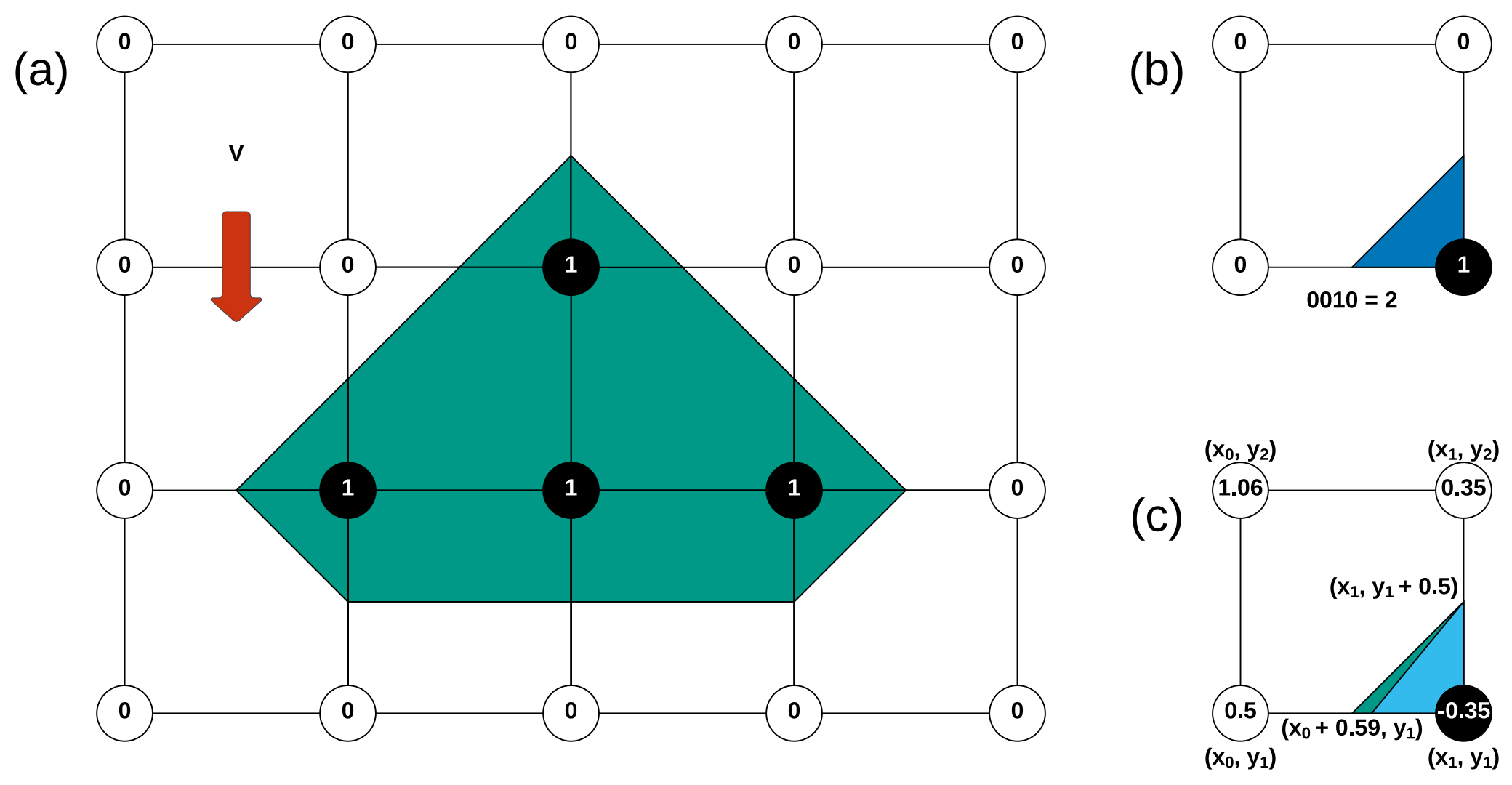

Figure 3. The contour extraction process of a Level Set Grid cross-section. (a) Linear scan (red arrow) of 2D grid cells with generated binary indices. (b) Reference to the pre-compiled lookup table for edge representation based on binary indices. (c) Determination of the segment's endpoints using bilinear interpolation (cyan triangle).

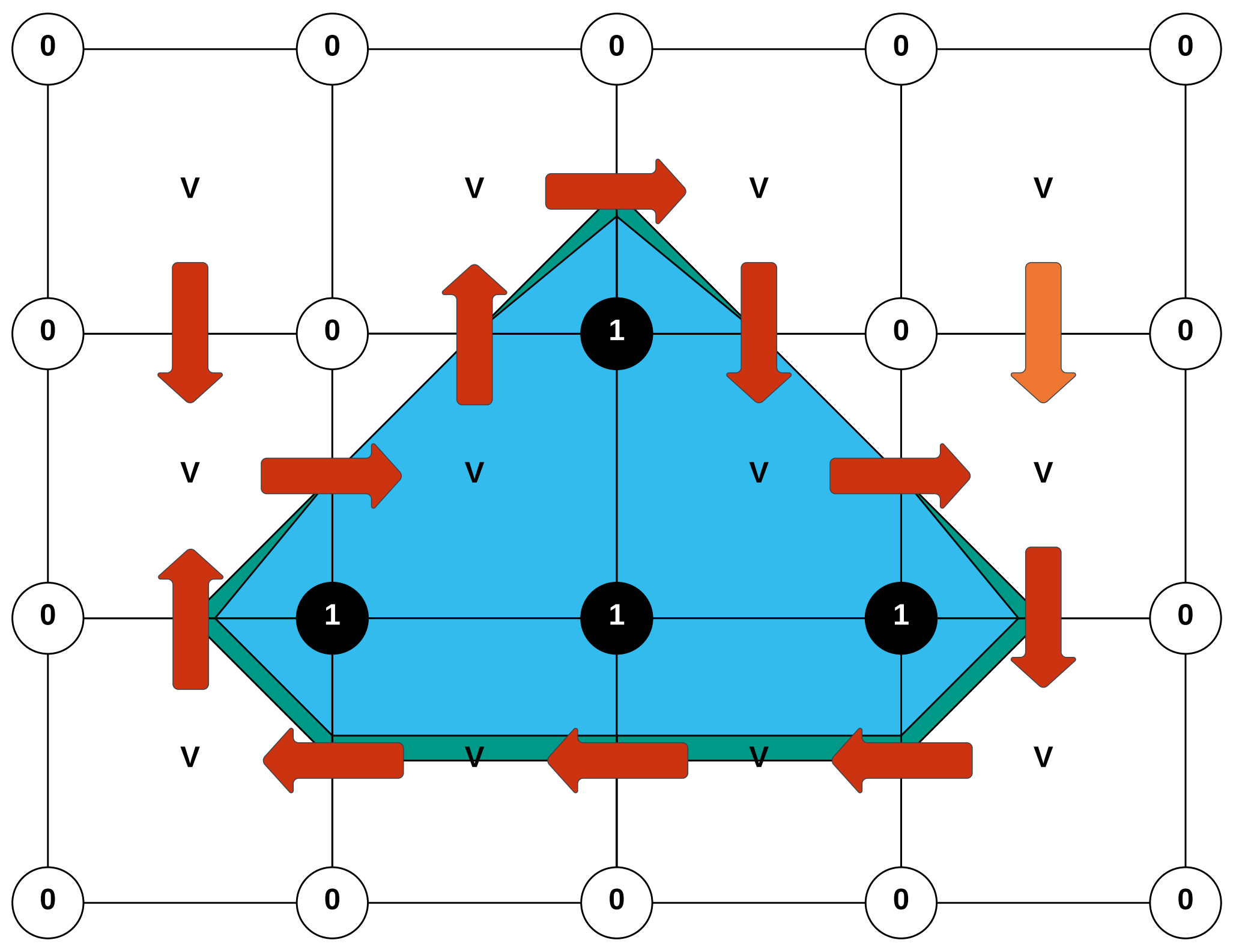

Figure 4. Schematic overview of the grid walk procedure for contour extraction. Beginning with the first segment, the walk (red arrows) progresses through adjacent grid cells, accumulating segment endpoints for the current contour. The process is terminated upon revisiting a 'visited' cell ('V'), indicating a completed contour. The algorithm then proceeds to identify potential additional contours (orange arrow), while avoiding previously visited grid components.

Surface Pathing

The slicing algorithm provides a basis for generating subtractive surface paths, using sliced contours to guide the manufacturing tool across the object's surface smoothly.

The process calculates surface normals at each contour point by evaluating the gradient of the signed distance field in the Level Set Grid, using central differences to identify the direction of maximum field increase (Figure 5a).

After calculating the surface normal, it's matched against a tool orientation vector and an angular range to determine its fit for the surface path, ensuring alignment with the tool's preferred orientation and tilt limits (Figure 5b).

Surface normals and orientation constraints are iteratively applied to contour points, creating a path that aligns with the object's geometry and manufacturing requirements (Figure 5c). This path guides the subtractive tool for precise material removal. The quality of this pathing—and the final object's surface—depends on the slicing accuracy and the Level Set Grid's resolution, emphasizing the importance of choosing the right grid resolution.

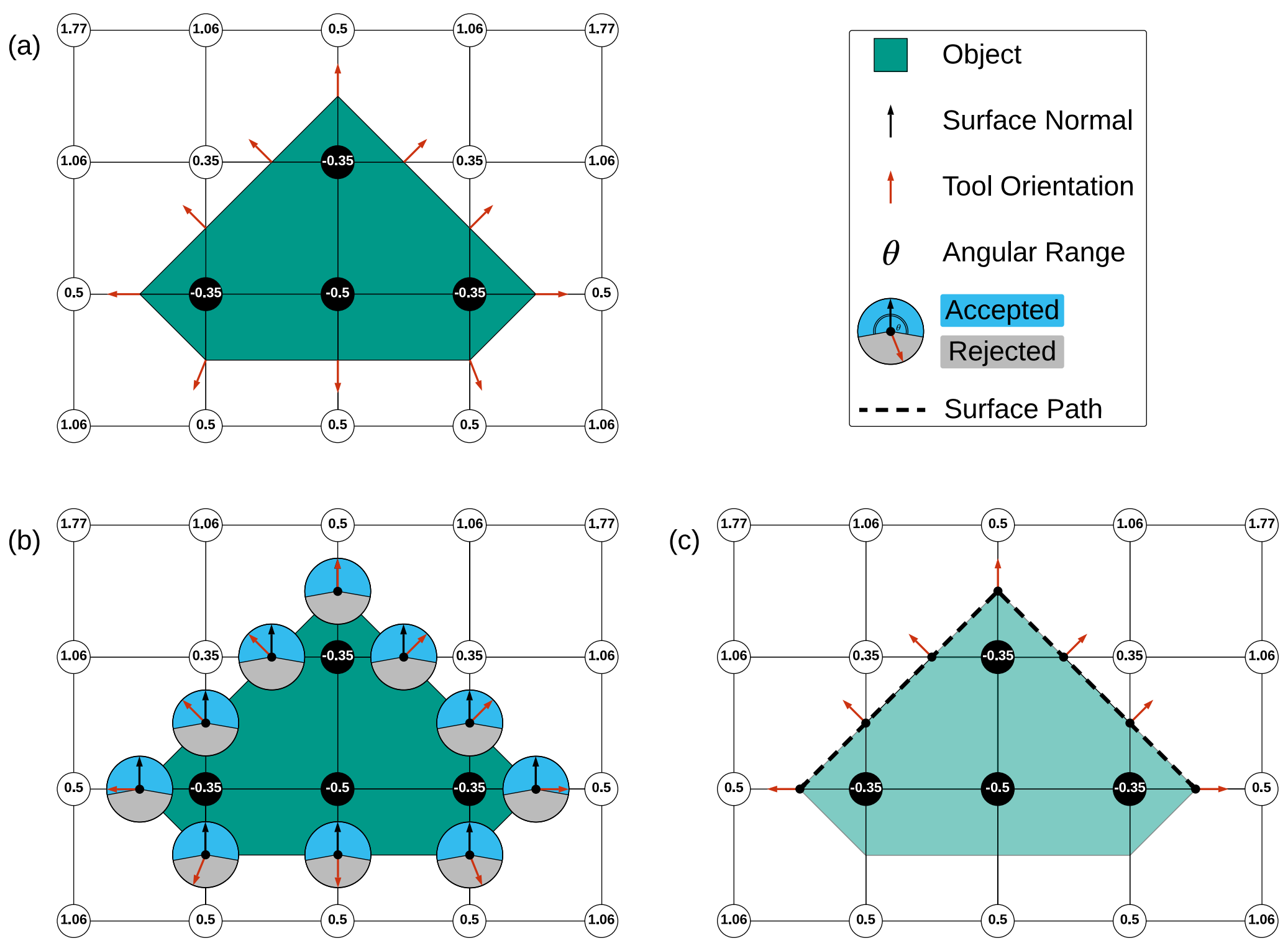

Figure 5. Workflow of surface pathing generation. (a) Graphical depiction of the surface normal computation using the gradient of the signed distance field. (b) Illustration of the orientation constraint evaluation, comparing the computed surface normal with the predefined tool orientation vector and acceptable angular range. (c) Resultant surface pathing based on the computed surface normal vectors and orientation constraints.

Results

To validate Level Set Grids for hybrid manufacturing, experiments were conducted on a complex, high-curvature object measuring 48mm x 48mm x 23mm. This object was converted from an STL mesh to a Level Set Grid, utilizing a voxel size of 0.2mm and a narrow band width of 1 voxel, showcasing precise curvature representation as depicted in Figure 6.

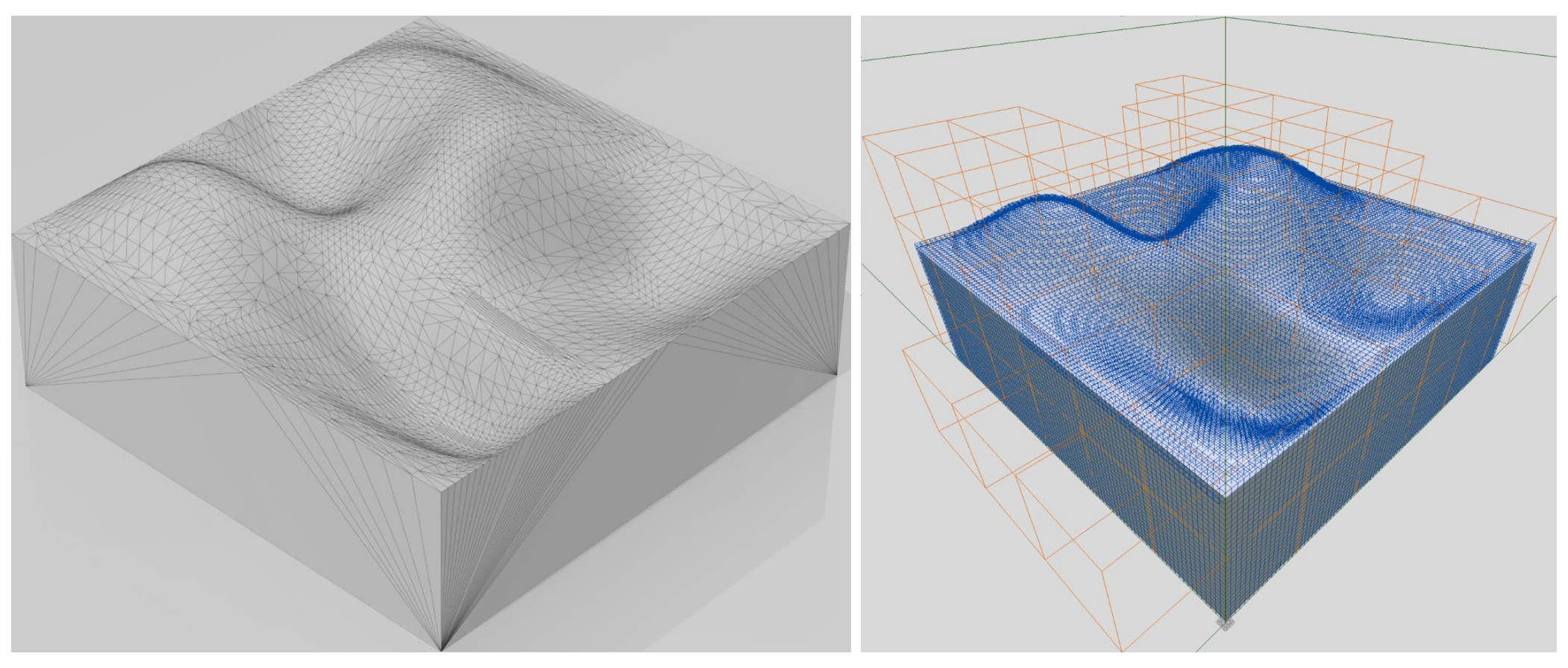

Figure 6. Visualization of the triangulated mesh representation of the object (left) alongside its corresponding Level Set Grid representation (right).

In the slicing trial, we extracted cross-sections from the Level Set Grid at 5-voxel intervals (1cm) along the z-axis and then isolated the polygonal contours. Figure 7 illustrates that these contours closely match the object's intricate curvature, highlighting the Level Set Grid's exceptional ability to capture detailed geometric features.

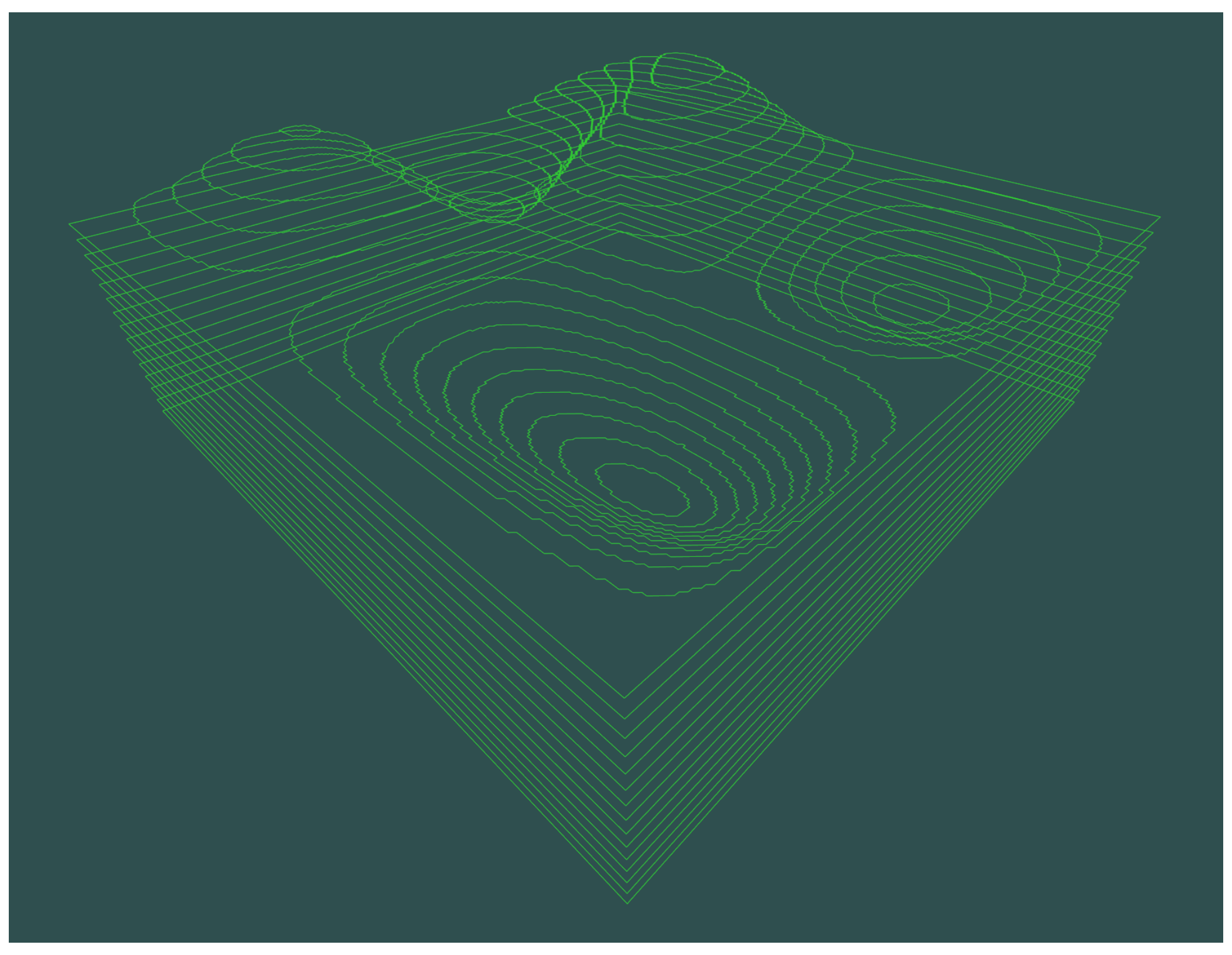

Figure 7. Extracted polygonal layer contours from the object's Level Set Grid, obtained by slicing the grid every 5 voxels (equivalent to 1cm) along the z-axis.

Demonstrating surface pathing involved selecting the object's top surface for toolpath generation with a tool orientation vector $O_t = (0, 0, 1)$ and an angular range of $\theta = 89°$. Contours were extracted every 5 voxels (1cm) along the y-axis, as shown in Figure 8, with surface normals computed from the Level Set Grid's gradient (Figure 9). These normals were then assessed against the tool orientation vector and angular range for inclusion in the pathing, resulting in a surface path that precisely mirrors the object's top curvature, illustrated in Figure 10.

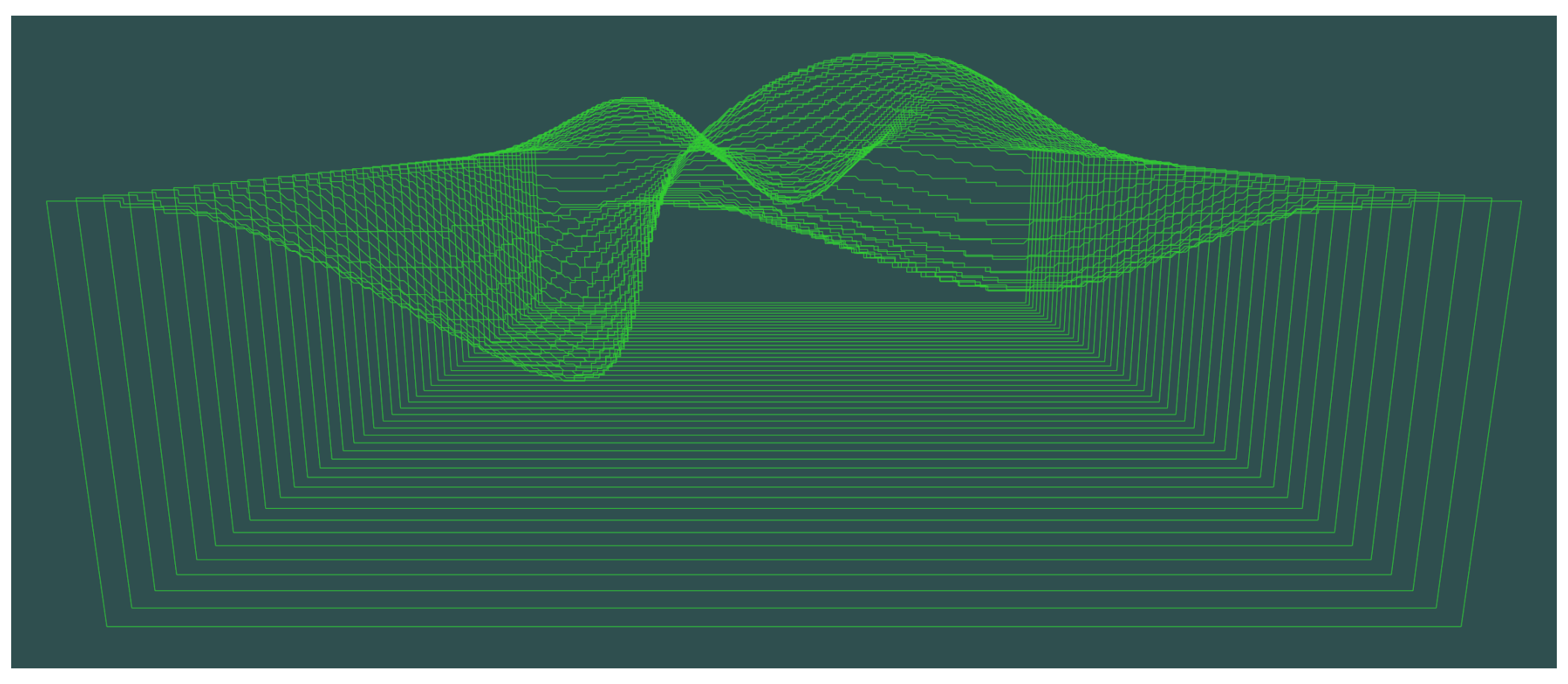

Figure 8. Extracted polygonal layer contours from the object's Level Set Grid, obtained by slicing the grid every 5 voxels (equivalent to 1cm) along the y-axis.

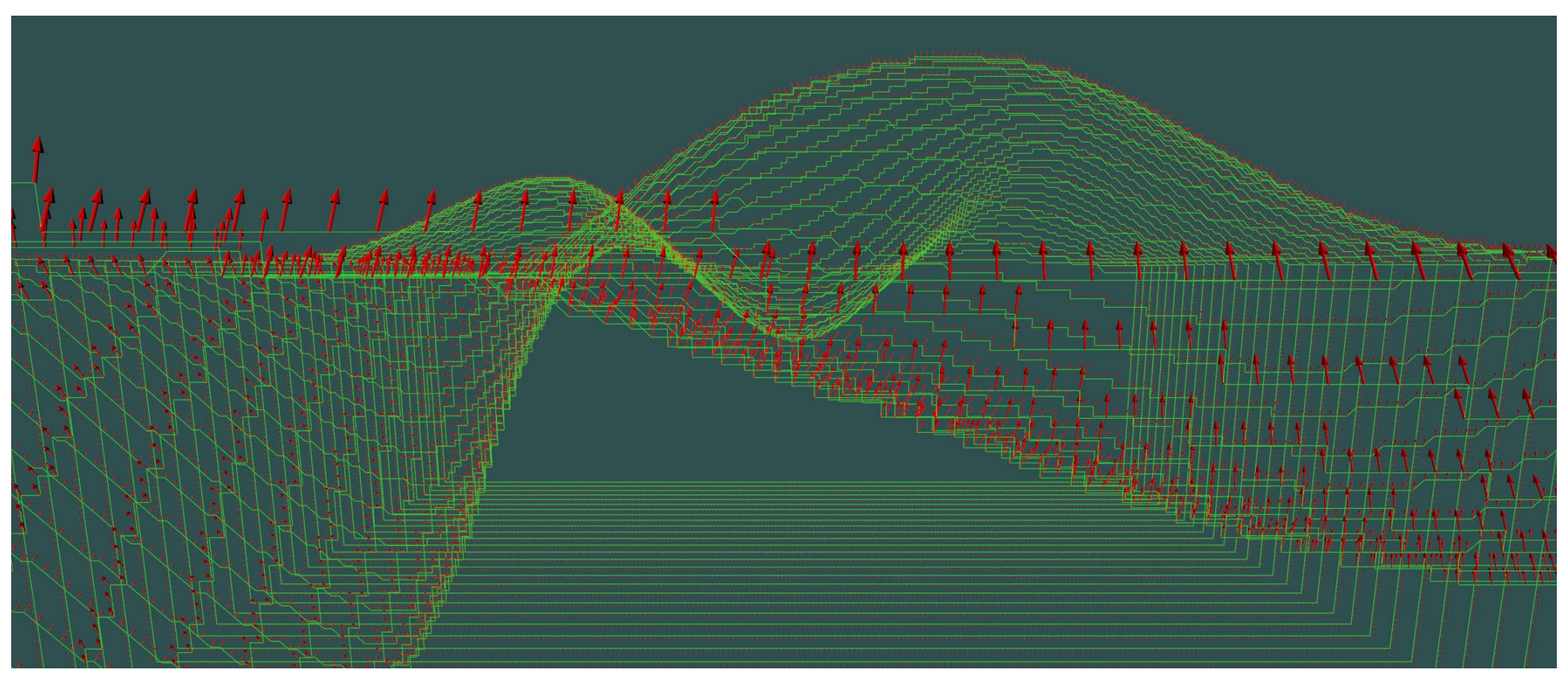

Figure 9. Computed surface normal vectors (red unit vectors) for each point in the contour, based on the gradient of the signed distance field in the object's Level Set Grid.

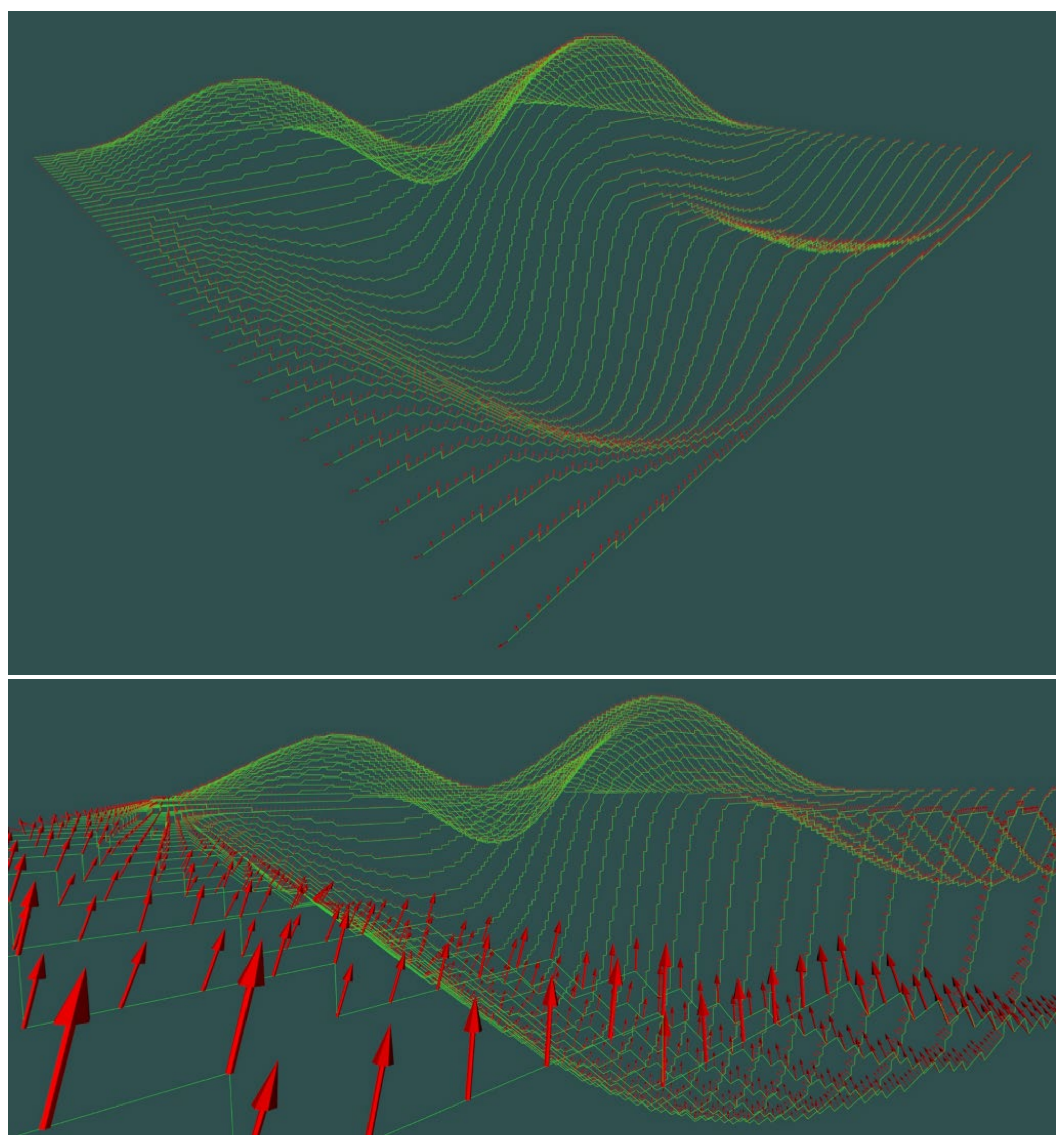

Figure 10. Results showcasing the generated surface pathing composed of points with surface normal vectors within the acceptable angular range when compared to the tool orientation vector.

Figure 10 reveals that surface pathing quality depends on the slicing accuracy, controlled by the Level Set Grid's resolution. Optimal grid resolution is crucial for superior surface quality in the final object. The grid's voxel size must be finer than the manufacturing system's positional accuracy. When high resolution poses computational challenges, polyline simplification may improve pathing accuracy.

Conclusion

This study introduces the Level Set Grid, a hybrid model blending explicit and implicit forms to enhance additive and subtractive process parallelization in hybrid manufacturing. It highlights precise geometry modeling and efficient collision detection. Results demonstrate its effective integration into additive slicing and subtractive pathing, accurately capturing and constructing object geometries. While marking progress, this research is a foundation for further exploration, aiming to fully integrate the Level Set Grid into manufacturing workflows and optimize its performance across various applications.