

A Novel Approach to Triangulate 3D Point Clouds for 3-Axis Sculptured Surfaces Machining

Mohamed Bey

Centre de Développement des
Technologies Avancées (CDTA)
Division Productique et Robotique (DPR)
Baba Hassen, Algiers, Algeria
Université des Sciences et de la
Technologie Houari Boumediene (USTHB)
Faculté de Génie Mécanique & Génie des
Procédés (FGMGP)
Bab Ezzouar, Algiers
Algeria

Krimo Azouaoui

Université des Sciences et de la
Technologie Houari Boumediene (USTHB)
Faculté de Génie Mécanique & Génie des
Procédés (FGMGP)
Bab Ezzouar, Algiers
Algeria

Reverse Engineering (RE) is employed to replicate sculptured surfaces from 3D point clouds. However, reconstructing continuous surfaces remains challenging. To address this, triangulated models are used as an alternative. In this paper, we present a novel and simple approach based on geometric considerations for generating a 3D triangulation from 3D point clouds with fewer triangles while satisfying a predefined accuracy. In this approach, we combine regular and adaptive triangulations and explicitly integrate approximation accuracy, adaptive subdivision, point distributions within triangles, and triangle quality. The approach involves four steps: (1) generating an initial regular 3D triangulation; (2) adaptively subdividing triangles; (3) verifying triangle qualities; and (4) generating the STL model for use in 3-axis machining. The lower number of generated triangles significantly reduces the processing times for optimizing the machining and the costs. We validated the approach across numerous 3D point clouds of complex sculptured surfaces, proving its performance for triangulation accuracy and processing time.

Keywords: Sculptured Surface, 3D Point Cloud, Adaptive Triangulation, STL Model.

1. INTRODUCTION

Parts with sculptured surfaces are widely used in various industries, including mold, die, aerospace, and automotive. They are often manufactured using multi-axis Computer Numerical Control (CNC) milling machines ranging from 3-axis to 5-axis based on surface topology and tool accessibility. Main objectives in machining are excellent surface quality and a minimum machining time [1]. The part shape is obtained in three stages: roughing, semi-finishing, and finishing [2]. The corresponding toolpaths are generated by taking several factors into account, such as the Computer-Aided Design (CAD) part model, machining accuracy, cutting tools, machining strategies, interference and collision problems, and cutting conditions [3]. Toolpaths are generated using either continuous parametric models [4-12] or triangulated models [13-15].

CAD models of sculptured surfaces can be obtained using either Forward Engineering (FE) or Reverse Engineering (RE) techniques. The former is used when surface geometries are relatively simple, whereas the latter is adopted when the CAD model part is unavailable or when part surfaces are highly intricate. The RE technique allows replicating parts from 3D point clouds obtained after scanning them using contact or noncontact measuring devices [16]. The primary challenge lies in reconstructing continuous smooth surfaces using surface fitting approaches [17-20]. However, the-

se approaches are generally complex, and the procedure for surface reconstruction from a massive point cloud is tedious and time-consuming. To overcome this issue, triangulated models are utilized as an alternative.

To reduce costs and product development cycles, the machining of sculptured surfaces must be optimized. For this, a large number of factors must be considered, including production cost, accuracy, optimizing cutting tool paths, minimizing machining time, choosing suitable cutting tools (size, shape, material), avoiding interference and collision, machining strategies, local geometric shapes, cutting conditions, etc. Consequently, generating toolpaths by considering only a subset of constraints is time-consuming and expensive. When machining is based on a point cloud, the relationships between points and necessary geometric properties, such as unit normal vectors and principal curvatures, are unknown. These drawbacks make toolpath generation and machining optimization extremely challenging tasks. Moreover, a high point density is required to represent complex surfaces accurately, which significantly increases difficulty and processing time.

In industrial practice, sculptured surfaces can be trimmed or untrimmed, and highly precise surface representation is not always required. These surfaces are employed for both functional and aesthetic applications. Functional applications include aerodynamic surfaces (turbine blades and impellers), optical components (reflectors), medicinal devices (prosthetics), and manufacturing (molds and dies). In this case, high accuracy is required to meet functional requirements (increase performance, optimize aerodynamics, fluid flow, heat transfer, etc.). Many consumer products have aesthetic applications, such as consumer electronics, electrical casings, automotive dashboards, and packaging. For

Received: August 2025, Accepted: October 2025

Correspondence to: Mohamed Bey, Centre de Développement des Technologies Avancées (CDTA), Baba Hassen Algiers, Algeria.

E-mail: bey_mohamed@yahoo.com

doi: 10.5937/fme2504661B

© Faculty of Mechanical Engineering, Belgrade. All rights reserved

FME Transactions (2025) 53, 661-680 661

these types of products, these surfaces are used to improve their aesthetic appeal, which is a key aspect in customer satisfaction and ergonomics. As a result, they do not need to be extremely accurate, and consequently a lower accuracy is sufficient.

Visibility of sculptured surfaces is a critical factor in various applications, especially in manufacturing. It relates to the ability to see or access a surface from a specific direction without obstruction, which is crucial for machining and inspection. In machining, the visibility of a surface determines whether a cutting tool can reach and process it. A surface is considered visible from a particular direction if a line of sight can be created from that direction to any point on the surface. If the view direction is parallel to the vertical Z-axis (tool axis) and all surface points are visible, these surfaces can be machined on 3-axis CNC machines in a single setup (Figure 1.a). If there exist invisible points (hidden zones), multiple setups are required to machine all surfaces on 3-axis machines. However, all surfaces can be machined in a single setup when using 5-axis machines (Figure 1.b). This work deals with point clouds of sculptured surfaces to be machined on 3-axis Computer Numerical Control (CNC) milling machines.

Based on these practical facts, triangulating point clouds with different accuracies using approximating techniques appears to be a more efficient approach, particularly for sculptured surfaces. So, the fundamental motivation for the research is the development of a novel and simple approach for generating a 3D triangulated model from point clouds with an imposed accuracy that varies depending on the sculptured surface type (functional or aesthetic) instead of using complicated surface fitting techniques. As a result, for the same point cloud, a unique triangulated model is generated for each imposed accuracy. Therefore, the objective of this work is to optimize the sculptured surface machining process and shorten the product development cycle.



a. 03-axis machining.



b. 05-axis machining.

Figure 1. Machining of sculptured surfaces.

In this paper, we present a novel and simple approach based on geometric considerations to generate a 3D triangulation from a 3D unstructured point cloud with fewer triangles while maintaining a predefined accuracy for 3-axis sculptured surface machining using the least squares method. In this approach we combine both

regular and adaptive triangulations to increase its efficiency and explicitly integrate approximation accuracy, adaptive subdivision, point distributions within triangles, and triangle quality. The approach involves four main steps: (1) generating an initial regular 3D triangulation; (2) adaptively and recursively subdividing triangles based on point distribution until the desired accuracy is achieved; (3) verifying triangle qualities to retrieve surface boundaries efficiently; and (4) generating the associated STL model for use in 3-axis machining. The reduced number of generated triangles significantly reduces the memory needed for storing them, processing times to optimize the machining of the sculptured surfaces, and, as a result, costs. We have validated the approach's efficacy for triangulation and processing time on various 3D point clouds of theoretical and real intricate sculptured surfaces.

This work belongs within the first research direction, and its main contributions are outlined as follows:

- Efficient triangulation approach combining regular and adaptive methods.
- Achieving a compromise between triangulation accuracy and processing time.
- Introduction of a novel triangle subdivision strategy based on point distribution within triangles.
- Applicability for both trimmed and untrimmed sculptured surfaces.
- Efficient retrieving of physical part boundaries.
- Introduction of a novel metric for evaluating boundary triangle quality.
- Generating fewer triangles and thus avoiding over-precision for functional and nonfunctional surfaces.
- Production of CNC-machinable triangles.
- Reducing the product development cycle.

The main practical application of this work is the duplication of functional and nonfunctional (aesthetic) sculptured surfaces from 3D unstructured point clouds for milling on 3-axis CNC machines. As a result, the application field is broad, including aerodynamic surfaces, molds, dies, car bodies, toys, and more. The developed software can be integrated into CAM software (Computer-Aided Manufacturing) to directly use the generated STL model in the production and the optimization of the machining.

The paper is structured as follows. Section 2 offers an overview of related works on machining 3D point clouds. Section 3 provides a detailed description of the proposed triangulation approach. Experimental results and discussions are presented in Section 4. Section 5 offers conclusions and future perspectives.

2. RELATED WORKS

To machine parts defined by point clouds, two directions are followed by researchers. The first consists of creating meshes or continuous surfaces before generating toolpaths, whereas the second one involves developing approaches to generate toolpaths directly from point clouds.

A comprehensive survey of different techniques for surface reconstruction using point clouds is provided in

[21]. They are classified based on the types of employed priors, the addressed data imperfection, and the generated output types. They took into account the following: point cloud imperfections (sampling density, noise, outliers, misalignment, and missing data), point cloud inputs (surface normals, scanner information, RGB images), and techniques (smoothness, visibility, planarity, etc.). [22] presented an overview and comparison of learning-based and conventional methods for solving the surface reconstruction problem from point clouds using interpolation and approximation. Under appropriate conditions, learning-based techniques can generate highly detailed surfaces. However, this result requires training on massive sets of sufficiently complex surfaces and related point clouds. Delaunay triangulation, which creates 2D triangles or 3D tetrahedra from point clouds, is employed in several applications, including computer graphics and computational geometry. [23] presented the Delaunay triangulation methods and analyzed their implementations on CPUs, GPUs, and FPGAs. [24] developed the “Delaunay Meshing Network+” (DMNet+) approach to reconstruct surfaces from point clouds using a Graph Neural Network (GNN). [25] presented a voxel structure that reconstructs an initial mesh from a point cloud and optimized it to improve quality. [26] presented the MergeNet approach to generate meshes from sparse point clouds. [27] developed the PPSurf approach to reconstruct surfaces from point clouds. [28] developed an approach for identifying the set of elliptical surface segments in each scanned line of a structured point cloud.

For the first direction, from a regular point cloud, [29] generated 3-axis roughing and finishing toolpaths, and [30] reconstructed a B-Spline surface. In [31–32], to machine parts on five positioned axes milling machines, an unstructured point cloud is approximated by voxels, the minimum part setup number is calculated, and a 3-axis toolpath is generated. From a 2D triangular grid, [33] created a simplified 3D triangulation and generated a 5-axis toolpath. [34–35] selected the optimal ball-end cutter based on Delaunay triangulation. [36] developed an iso-planar piecewise linear toolpath from an irregular point cloud. [37], based on Delaunay triangulation, created regions and determined optimal ball-end cutters. [38], from the point cloud, generated a mesh, segmented it, and created a toolpath with a given scallop height. [39] denoised a point cloud, created an octree model, and generated roughing and finishing toolpaths.

For the second direction, [40] created 3-axis roughing and finishing toolpaths from a regular point cloud. [41] generated a 3-axis toolpath using a parallel plane machining strategy with circular arcs. [42] subdivided the point cloud into areas of varying complexity, selected a ball-end cutter for each area, and generated the respective 3-axis toolpaths. [43] generated a 3-axis toolpath using the Inverse Tool Offset method (ITO). [44] created iso-planar toolpaths with flat-end, ball-end, and fillet-end cutters of various sizes from a 2D grid. [45] generated adaptive toolpaths from point clouds. [46] created 3-axis spline NC paths from point clouds. [47] devised 3-axis toolpaths maintaining a constant scallop height. [48] segmented the point cloud into subregions and generated 3+2-axis machining toolpaths for each

region. [49] created continuous spiral toolpaths for 3-axis milling. [50] created a 3-axis toolpath using a parallel plane machining strategy. [51] generated adaptive 3-axis spiral toolpaths from point clouds.

From the above works, several deficiencies can be identified:

- Generating continuous smooth surfaces is a challenging, time-consuming, and laborious process that necessitates a constant interaction between the software and the designer. As a result, it lacks full automation.
- In Delaunay triangulation, every point of the point cloud corresponds to a triangle's vertex. Thus, a dense point cloud produces a large number of triangles, which increases considerably the amount of memory to store the results and the processing time for optimizing the machining. Besides, Delaunay triangulation is the convex hull of the point cloud, and other approaches must be used to retrieve the surface boundaries.
- Learning-based approaches necessitate training on a wide variety of intricate surfaces, and the results are related to the point cloud density.
- Other specific approaches are very complex to implement.

Therefore, it is crucial to propose a simple approach that overcomes these drawbacks and enables the generation of 3D triangles for trimmed and untrimmed sculptured surfaces with fewer triangles in a reasonable processing time while preserving an imposed accuracy.

In addition, machining accuracy, processing time, and complexity in previous works were all directly related to the adopted approaches' parameters and the point cloud density. However, several important factors were overlooked, such as surface nature (trimmed and untrimmed), model accuracy, and surface functionality. The novelty of our work, situated within the first resection direction, lies in addressing these aspects explicitly by proposing a simple and efficient triangulation of 3D point clouds with a reduced number of triangles for 3-axis sculptured surface machining. Thereafter, machining optimization, which includes cutting tool combination, machining strategies, cutting conditions, etc., becomes simpler than directly considering the point cloud.

3. PROPOSED APPROACH

Given that the approach considers intricate sculptured surfaces, a dense input point cloud is required to retrieve efficiently their geometries. Furthermore, since these surfaces will be machined on vertical 3-axis CNC milling machines, it is assumed that the 3D point cloud is a height map representing both trimmed and untrimmed sculptured surfaces, with the Z-axis being vertical. Figure 2 depicts the overall flowchart of our proposed approach. The subsequent subsections provide details of each step.

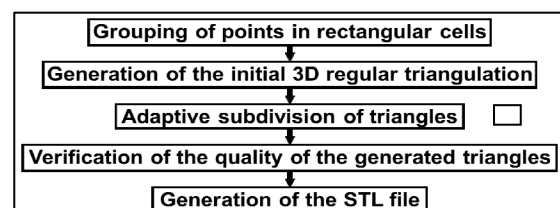


Figure 2. General flowchart of the proposed triangulation approach.

3.1 Grouping of points in rectangular cells

Since the point cloud is dense and the approach requires point assignment to triangles, it is necessary to reduce the processing time for this recurrent procedure. To address this problem, the first step of the approach involves grouping points into planar rectangular cells. Using point coordinates, raw part limit points (P_{min} and P_{max}) and minimum dimensions (length L , width l , and height H) are computed (Figure 3.a). Cell creation requires specifying the number of cells, n_x and n_y , along the X-axis and Y-axis, respectively (Figure 3.b). Thereafter, the limit points P_{low} and P_{top} of each cell on the XY plane are determined (Figure 3.b). Next, the point cloud is projected onto the XY plane, and the points belonging to each cell are identified (Figure 3.b). Finally, cell limits along the Z-axis are calculated, resulting in 3D cells (Figure 3.c).

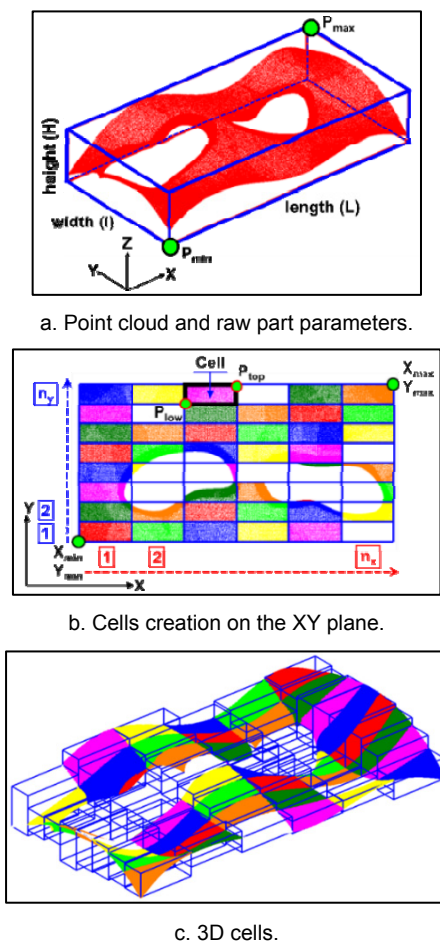


Figure 3. 3D rectangular cells.

3.2 Generation of the initial 3D regular triangulation

The proposed approach begins with the generation of an initial 3D regular triangulation, which is then refined via adaptive subdivision. It is established through the following steps:

3.3 Generation of the initial 2D planar regular triangulation

The generation of the initial 2D planar regular triangulation passes by the following steps (Figure 4):

- Specify the initial lengths of the triangle's segments, L_x and L_y , along the X-axis and Y-axis, respectively.
- Calculate the associated numbers of segments $Nseg_x$ and $Nseg_y$ along the X-axis and Y-axis, respectively.
- Generate the initial 2D planar regular triangulation on the XY plane.
- Project the point cloud onto the XY plane and determine for each triangle its points based on cell points.
- Identify the neighbors V_1 , V_2 , and V_3 for each triangle T . Neighbors are important in adaptive triangulation since they help to reduce the processing time.

It is important to note that the initial lengths of the triangle segments are related to the point cloud geometric complexity. The more the geometry is complex, the more the initial lengths must be small. The raw part's dimensions and the initial lengths of the triangle segments along the X-axis and Y-axis are used to calculate the values of $Nseg_x$ and $Nseg_y$ (Figure 4). The number of segments $Nseg_x$ ($Nseg_y$) is calculated by dividing the length L (width l) of the raw part by the initial triangle segment's length L_x (L_y) along the X-axis (Y-axis). Consequently, the shorter the initial lengths of the triangle segments, the greater the number of them. Thus, the approach processing time is directly related to these values. As a result, large values reduce the number of triangles required for adaptive subdivision, which decreases processing time and considerably increases the triangle number.

To test if a point belongs to a triangle, several methods can be used, including barycentric coordinates, triangle area, oriented angles, crossing tests, and cross product [52]. In this work, the cross product is used.

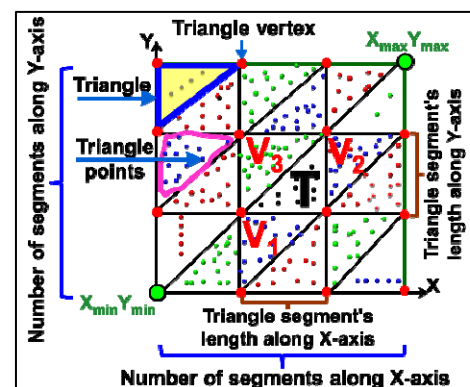


Figure 4. Generation of the initial 2D planar regular triangulation.

3.4 Filtering of 2D planar triangles and vertices

The proposed approach is based on using the least squares method to approximate triangle points by a non-vertical plane with a specified accuracy. Thus, a triangle is filtered out, and it is ignored in subsequent steps if it encloses less than three (3) points or all its projected points on the XY plane are aligned along the same straight line (Figure 5). Next, if all common triangles to a vertex are filtered out, then the vertex itself is filtered out (Figure 5).

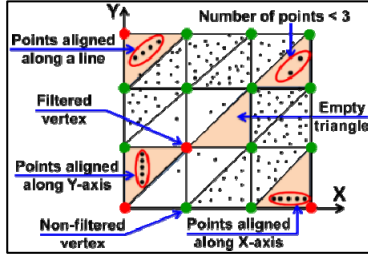


Figure 5. Filtering of triangles and vertices.

3.5 Conversion of 2D regular triangulation to 3D regular triangulation

In this step, only non-filtered 2D planar triangles are considered. So far, 2D triangles have been created on the XY plane, and to convert them to 3D triangles, triangle vertices' Z coordinates are calculated in two stages. The approximation of points by a set of triangles is adopted to take account of the noise included in the point cloud due to the scanning operation of the real object. Thus, in the first stage, the 3D points of each triangle are approximated by a non-vertical plane using the least squares method (Figure 6.a). The plane equation is:

$$Z = A \cdot x + B \cdot y + C \quad (1)$$

where the coefficients A, B, and C are given by:

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{i=N} X_i^2 & \sum_{i=0}^{i=N} X_i Y_i & \sum_{i=0}^{i=N} X_i \\ \sum_{i=0}^{i=N} X_i Y_i & \sum_{i=0}^{i=N} Y_i^2 & \sum_{i=0}^{i=N} Y_i \\ \sum_{i=0}^{i=N} X_i & \sum_{i=0}^{i=N} Y_i & N \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=0}^{i=N} X_i Z_i \\ \sum_{i=0}^{i=N} Y_i Z_i \\ \sum_{i=0}^{i=N} Z_i \end{bmatrix} \quad (2)$$

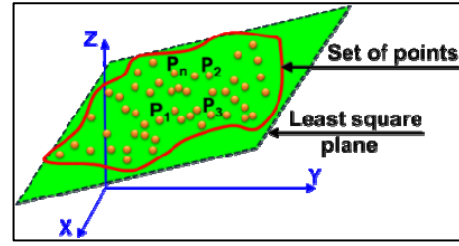
where N is the number of triangle points and X_i , Y_i , and Z_i are their coordinates (Figure 6.a).

Equation (1) indicates that it cannot be applied to a triangle with its points belonging to a vertical plane. This issue is due to the limits of the least squares method.

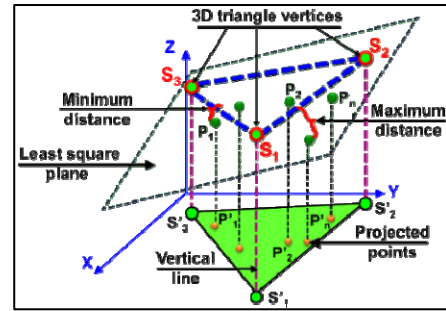
In the second stage, the vertices' Z coordinates are calculated. For each vertex of the 2D triangle, the intersection point between the vertical line passing through

this 2D vertex and the calculated plane returns its corresponding 3D vertex (Figure 6.b). Next, the maximum "error" between the calculated plane and triangle points is calculated for each triangle (Figure 6.b).

In this approach, if a vertex is shared by "n" triangles, its Z coordinate will have "n" values because the computation is done for each triangle without considering the shared triangles. From all Z values, the maximum Z coordinate (Z_{max}), minimum Z coordinate (Z_{min}), and average Z coordinate ($Z_{average}$) are calculated. This causes a geometric discontinuity (gap) between triangles at this vertex (Figure 7.a). To ensure C^0 continuity between the created 3D triangles, the 2D segment shared by two neighboring planar triangles must be shared after conversion to a 3D segment.



a. Approximating a set of points by a plane.



b. Conversion of a 2D triangle to a 3D triangle and corresponding errors.

Figure 6. Conversion of a 2D triangle to a 3D triangle.

Such a coincidence is ensured if each 3D segment's vertex coordinates have unique values. For this, a technique is proposed in which unique coordinate values are assigned to each vertex shared by multiple triangles. Once 3D triangles are created, three possibilities are examined for each vertex, considering its shared triangles (Figure 7.b):

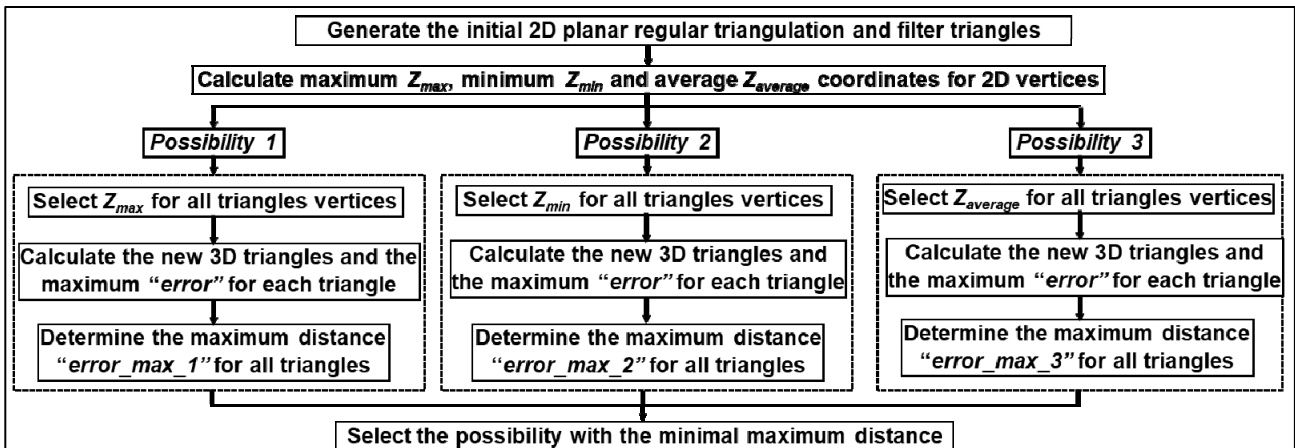


Figure 8. General flowchart for determining which possibility produces the best initial 3D regular triangulation.

- **Possibility 1:** The vertex's Z coordinate equals the maximum Z coordinate, Z_{max} .
- **Possibility 2:** The vertex's Z coordinate equals the minimum Z coordinate, Z_{min} .
- **Possibility 3:** The vertex's Z coordinate equals the average Z coordinate, $Z_{average}$.

Figure 7.c shows the scenario for the first possibility. Figure 8 shows the general flowchart of the procedure for selecting the possibility that generates the best first approximation of the point cloud. If the resultant error is smaller than the specified accuracy, the 3D triangles fulfill the required accuracy and thus remain unchanged, leading to the end of the process. However, if the error exceeds the specified accuracy, triangles are recursively and adaptively subdivided until the accuracy is fulfilled.

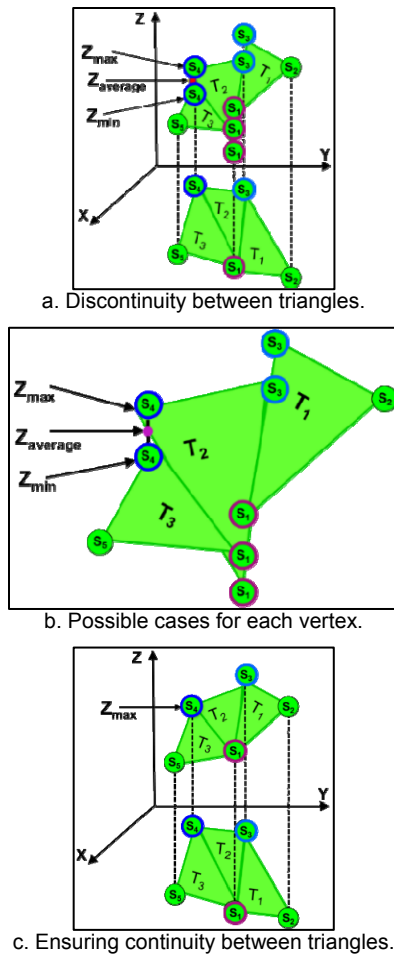


Figure 7. Continuity between triangles.

3.6 Adaptive subdivision of triangles

Adaptive triangle subdivision is used only when the initial 3D regular triangulation fails to satisfy the predefined accuracy "Precision". Its steps are outlined in the subsequent subsections.

3.7 Determining the triangle points' distribution relative to specific points

Given the unstructured nature of the point cloud, the initial 3D regular triangles may have irregular point distributions. This fact must be addressed explicitly. As

a result, the points distribution within each triangle is assessed using six specific points (Figure 9.a):

- The triangle vertices S_1 , S_2 , and S_3 .
- The midpoints M_1 , M_2 , and M_3 of the triangle's sides $[S_1S_2]$, $[S_2S_3]$, and $[S_1S_3]$.

The points closest to each specific point are identified from the projected triangle points onto the XY plane (Figure 9.b). To quantify the distribution of triangle points relative to these specific points, six Boolean variables are defined for each triangle:

- $close_{S_1}$, $close_{S_2}$, and $close_{S_3}$: Boolean variables associated with triangle vertices S_1 , S_2 , and S_3 , respectively.
- $close_{M_1}$, $close_{M_2}$, and $close_{M_3}$: Boolean variables associated with triangle sides' middle points M_1 , M_2 , and M_3 , respectively.

Boolean variables are introduced to check if there is a subset of triangle points close to a specific point or not. If it exists, then the Boolean variable is true; otherwise, it is false.

As shown in Figure 9.a, P_{S_1} , P_{S_2} , and P_{S_3} are the closest points to vertices S_1 , S_2 , and S_3 , respectively, while P_{M_1} , P_{M_2} , and P_{M_3} are the closest points to midpoints M_1 , M_2 , and M_3 , respectively. Then, the smallest distance associated with each specific point is calculated.

Two more parameters are included for adaptive subdivision to deal with triangles with nonhomogeneous point distribution:

➤ $area_min_tr$: this parameter limits the minimum triangle area to activate the triangle subdivision process, preventing the creation of extremely small triangles. The lower its value, the more the total triangle number increases.

➤ $percent_subd$: this parameter is introduced to quantify how far triangle points are from specific points to start adaptive subdivision to eliminate empty zones from the triangle (Figure 9.c and Figure 9.d). It helps in effectively capturing point cloud boundaries and ensuring homogeneous point distribution within triangles. A lower value of $percent_subd$ leads to more homogeneously distributed points within triangles, better recovery of boundaries, and an increase in the total triangle number. To achieve this objective, six threshold distances are assigned to each triangle using triangle side lengths:

1. Three threshold distances associated with triangle vertices (Figure 9.c):

$$\begin{cases} threshold_{S_1} = percent_{subdiv} \cdot \max(S_1S_2, S_1S_3) \\ threshold_{S_2} = percent_{subdiv} \cdot \max(S_2S_1, S_2S_3) \\ threshold_{S_3} = percent_{subdiv} \cdot \max(S_3S_1, S_3S_2) \end{cases} \quad (3)$$

2. Three threshold distances associated with midpoints of the triangle's sides (Figure 9.d):

$$\begin{cases} threshold_{M_1} = percent_{subdiv} \cdot S_1S_2 \\ threshold_{M_2} = percent_{subdiv} \cdot S_2S_3 \\ threshold_{M_3} = percent_{subdiv} \cdot S_1S_3 \end{cases} \quad (4)$$

where S_iS_j is the length of the triangle segment defined by the projected vertices S_i and S_j on the XY plane.

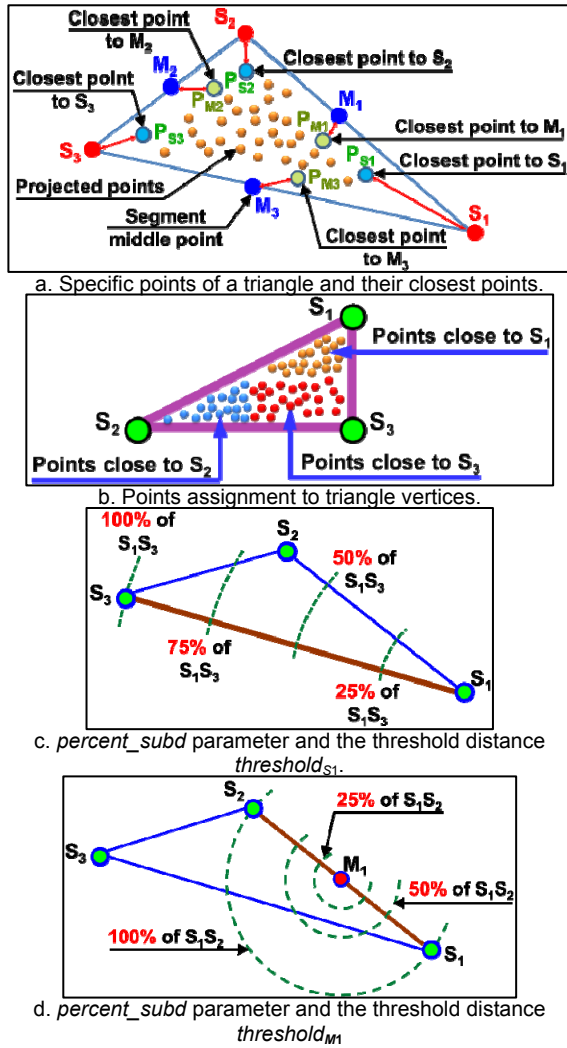


Figure 9. Points distribution within a triangle and threshold distances.

In the proposed approach, the six Boolean variables are adjusted based on the six threshold distances and the distances of the closest points to the specific points, as follows:

$$\left\{ \begin{array}{l} \text{if } (S_1 P_{S1} > threshold_{S1}) \\ \text{then } close_S1 = \text{false}; \text{ else } close_S1 = \text{true}; \\ \text{if } (S_2 P_{S2} > threshold_{S2}) \\ \text{then } close_S2 = \text{false}; \text{ else } close_S2 = \text{true}; \\ \text{if } (S_3 P_{S3} > threshold_{S3}) \\ \text{then } close_S3 = \text{false}; \text{ else } close_S3 = \text{true}; \end{array} \right. \quad (5)$$

$$\left\{ \begin{array}{l} \text{if } (M_1 P_{M1} > threshold_{S1}) \\ \text{then } close_M1 = \text{false}; \text{ else } close_M1 = \text{true}; \\ \text{if } (M_2 P_{M2} > threshold_{S2}) \\ \text{then } close_M2 = \text{false}; \text{ else } close_M2 = \text{true}; \\ \text{if } (M_3 P_{M3} > threshold_{S3}) \\ \text{then } close_M3 = \text{false}; \text{ else } close_M3 = \text{true}; \end{array} \right. \quad (6)$$

These adjustments are made to handle in the same manner the situations in which there are no points close to a specified point or the closest point is further away than the associated threshold distance. Consequently, a triangle is subdivided if at least one Boolean variable is

true. For this, the following steps are performed for each generated 3D triangle:

- Identify triangle points, approximate them with a plane, and calculate the maximum "error".
- Project triangle vertices and its points onto the XY plane.
- Determine midpoints M_1, M_2 , and M_3 of the triangle's sides and the projected triangle area, $area_tr$.
- Determine the sets of points close to triangle vertices, closest points, and minimum distances.
- Calculate the Boolean variables $close_S1$, $close_S2$, and $close_S3$.
- Calculate the threshold distances $threshold_{S1}$, $threshold_{S2}$, and $threshold_{S3}$ using equation (3).
- Adjust the Boolean variables $close_S1$, $close_S2$, and $close_S3$ using equation (5).
- Find point sets close to midpoints, including closest points and minimum distances.
- Determine the Boolean variables $close_M1$, $close_M2$, and $close_M3$.
- Calculate the threshold distances, $threshold_{M1}$, $threshold_{M2}$, and $threshold_{M3}$ using equation (4).
- Adjust the Boolean variables $close_M1$, $close_M2$, and $close_M3$ using equation (6).
- Subdivide the triangle if one of the following conditions is satisfied:
 - ❖ $area_tr \geq area_min_tr$ and ($close_S1 = \text{false}$ or $close_S2 = \text{false}$ or $close_S3 = \text{false}$ or $close_M1 = \text{false}$ or $close_M2 = \text{false}$ or $close_M3 = \text{false}$).
 - ❖ $error > Precision$.

3.8 Subdivision schemas of a triangle

Based on Boolean variables and the required accuracy $Precision$, the present work addresses nine cases for adaptive and recursive triangle subdivision:

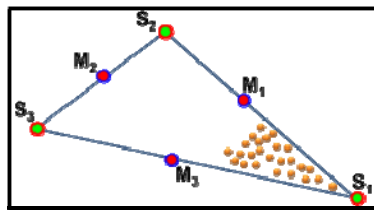
- **Case 1:** $close_S1 = \text{true}$, $close_S2 = \text{false}$, and $close_S3 = \text{false}$: all triangle points are only close to the vertex S_1 .
- **Case 2:** $close_S1 = \text{false}$, $close_S2 = \text{true}$, and $close_S3 = \text{false}$: all triangle points are only close to the vertex S_2 .
- **Case 3:** $close_S1 = \text{false}$, $close_S2 = \text{false}$, and $close_S3 = \text{true}$: all triangle points are only close to the vertex S_3 .
- **Case 4:** $close_S1 = \text{false}$, $close_S2 = \text{true}$, and $close_S3 = \text{true}$: all triangle points are far only from the vertex S_1 .
- **Case 5:** $close_S1 = \text{true}$, $close_S2 = \text{false}$, and $close_S3 = \text{true}$: all triangle points are far only from the vertex S_2 .
- **Case 6:** $close_S1 = \text{true}$, $close_S2 = \text{true}$, and $close_S3 = \text{false}$: all triangle points are far only from the vertex S_3 .
- **Case 7:** $close_M1 = \text{false}$, or $close_M2 = \text{false}$, or $close_M3 = \text{false}$: all triangle points are far only from M_1, M_2 , or M_3 .
- **Case 8:** $close_S1 = \text{false}$, $close_S2 = \text{false}$, and $close_S3 = \text{false}$: all triangle points are far from the vertices S_1, S_2 , and S_3 .

- **Case 9:** $error > Precision$: the triangle does not fulfill the required accuracy.

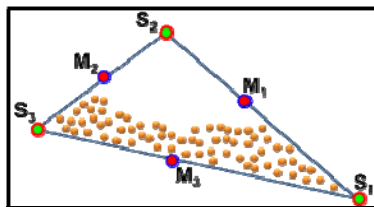
For the subdivision process, these cases are considered in the aforementioned order for each triangle. The goal of the initial eight cases is the generation of a homogeneous point distribution within the triangles and the accurate capture of point cloud boundaries, while the aim of the ninth case is to guarantee that the triangle satisfies the specified accuracy. These nine cases are classified into five categories, each with its own subdivision scheme:

- **First category:** it is associated with cases 1, 2, and 3 (Figure 10.a).
- **Second category:** it is associated with cases 4, 5, and 6 (Figure 10.b).
- **Third category:** it is associated with case 7 (Figure 10.c).
- **Fourth category:** it is associated with case 8 (Figure 10.d).
- **Fifth category:** it is associated with case 9.

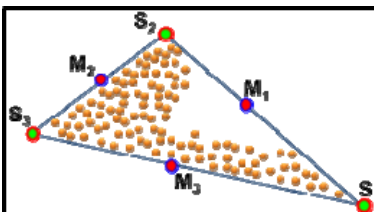
It is important to emphasize that all possible scenarios must be examined in each category. The first four categories typically arise at the point cloud boundary or with low point cloud density. For each triangle, the categories are examined sequentially in the abovementioned order. The proposed subdivision schema for each category is described below.



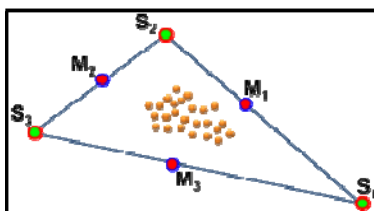
a. First category.



b. Second category.



c. Third category.



d. Fourth category.

Figure 10. Point distribution categories.

1) Subdivision schema for the first category: Its steps are described for case 1 (Figure 11.a):

- Identify the farthest point P from vertex S_1 and calculate the distance R between them.
- Determine the bisector (Δ) of the angle $[S_2S_1S_3]$.
- Find the intersection point S of the bisector (Δ) and the circle of radius R centered on S_1 .
- Determine the straight line (Σ) passing through the point S and perpendicular to the bisector (Δ).
- Calculate the intersection point C_1 of the straight line (Σ) and the segment $[S_1S_3]$.
- Calculate the intersection point C_2 of the straight line (Σ) and the segment $[S_1S_2]$.
- Subdivide the triangle T into three triangles: T_{new} , T_1 , and T_2 .
- Assign original triangle points to triangle T_{new} and filter triangles T_1 and T_2 .
- Subdivide neighbor triangle $V1$ into two triangles and neighbor triangle $V2$ into two triangles.
- Assign original triangles' points to the four new triangles and determine filtered triangles.
- Update neighbor triangles.
- Determine point distribution and Boolean variables for the modified triangles.
- Determine the approximate planes for the modified triangles.

2) Subdivision schema for the second category: its steps are described for case 4 (Figure 11.b):

- Find the closest point P to the vertex S_1 and calculate the distance R between them.
- Calculate the intersection point C_1 of the segment $[S_1S_3]$ and the circle of radius R centered on S_1 .
- Determine the intersection point C_2 of the segment $[S_1S_2]$ and the circle with radius R centered on S_1 .
- Generate three triangles from the triangle T : T_{new} , T_1 , and T_2 .
- Filter the triangle T_2 and assign original triangle points to triangles T_{new} and T_1 .
- Subdivide neighbor triangle $V1$ into two triangles and neighbor triangle $V2$ into two triangles.
- Assign original triangles' points to the four new triangles and determine filtered triangles.
- Update neighbor triangles.
- Determine point distribution and Boolean variables for the modified triangles.
- Determine the approximate planes for the modified triangles.

3) Subdivision schema for the third category: its steps are described for case 7 (Figure 11.c):

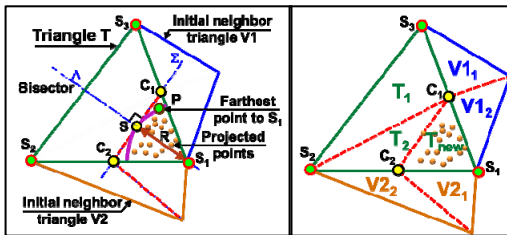
- Subdivide the triangle T at the point M_1 into two triangles: T_{new} and T_1 .
- Assign original triangle points to the two new triangles and determine filtered triangles.
- Subdivide neighbor triangle V into two triangles: V_1 and V_2 .
- Determine triangles' points for the two new triangles, V_1 and V_2 , and determine filtered triangles.
- Update neighbor triangles.

- Determine point distribution and Boolean variables for the modified triangles.
- Determine the approximate planes for the modified triangles.

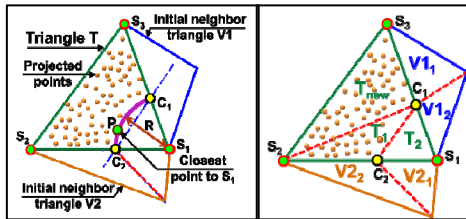
4) Subdivision schema for the fourth category: its aim is to remove the triangular zone that is farthest from a given vertex using the following steps (Figure 11.d):

- Determine the closest point P_1 to vertex S_1 and the distance P_1S_1 .
- Determine the closest point P_2 to vertex S_2 and the distance P_2S_2 .
- Determine the closest point P_3 to vertex S_3 and the distance P_3S_3 .
- If the distance P_1S_1 is the largest, apply the subdivision schema for the second category by considering vertex S_1 .
- If the distance P_2S_2 is the largest, apply the subdivision schema for the second category by considering vertex S_2 .
- If the distance P_3S_3 is the largest, apply the subdivision schema for the second category considering vertex S_3 .

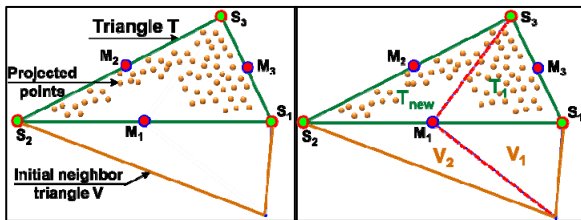
5) Subdivision schema for the fifth category: its steps are similar to those of the subdivision schema for the third category, with the exception that the subdivision is done relative to the middle point of the longest triangle side. This way allows avoiding creating triangles with a large ratio of side lengths (Figure 11.c).



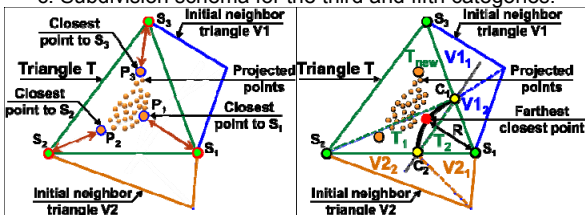
a. Subdivision schema for the first category.



b. Subdivision schema for the second category.



c. Subdivision schema for the third and fifth categories.



d. Subdivision schema for the fourth category.

Figure 11. Triangle subdivision schemas before and after subdivision.

3.9 Verification of triangle quality and generation of the STL file

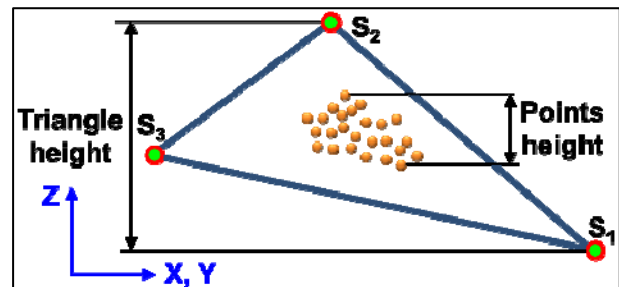
Since the triangular mesh is an approximation of the object's real geometry, its density and quality have significant impacts on accuracy. So, mesh quality assessment is essential for ensuring accuracy. For this, after generating the triangular mesh, its quality must be assessed using various mesh quality metrics (aspect ratio, minimum angle, triangle size, Jacobian ratio, orthogonality, curvature, dihedral angle, etc.) [53]. These metrics are used to identify and fix possible mesh problems. Triangles must be as close to equilateral as possible to avoid elongated or thin triangles in the mesh.

The experimental validation of the presented approach shows that for specific conditions of point distribution within a boundary triangle, its height along the Z-axis increases considerably. To address this problem, a new mesh quality metric is proposed, which is the ratio of the triangle height T_{height} and the triangle points height P_{height} along the Z-axis (Figure 12.a). The T_{height}/P_{height} ratio is high for the following cases (Figure 12.b):

- Triangle points are very close to one vertex.
- Triangle points are very close to only one triangle side.
- Triangle points are almost aligned.

The 3D triangles generated for the above cases will have a significant triangle height and be almost vertical. They are classified as “poor” triangles if their T_{height}/P_{height} ratio is greater than a specified threshold ratio $ratio_{height}$. According to the conducted tests, the maximum value of $ratio_{height}$ is 10 for a good triangulation. Triangles of poor quality produce tool movements without effective machining, which increases the total toolpath length. Consequently, it is indispensable to filter these triangles to address this problem and to reduce the total toolpath length. The classification of the generated triangles is done from the following steps:

- Verify if the triangle is a boundary triangle.
- If it is, perform the following steps:
 - Calculate the T_{height}/P_{height} ratio and compare it to a specified threshold ratio $ratio_{height}$.
 - ❖ If $T_{height}/P_{height} > ratio_{height}$, then the triangle quality is “poor” and is filtered. Next, the filtered triangles’ neighbors are updated and become boundary triangles (Figure 13).
 - ❖ Otherwise, the triangle quality is “good”, and thus it is not filtered.



a. Heights of triangle and its points.

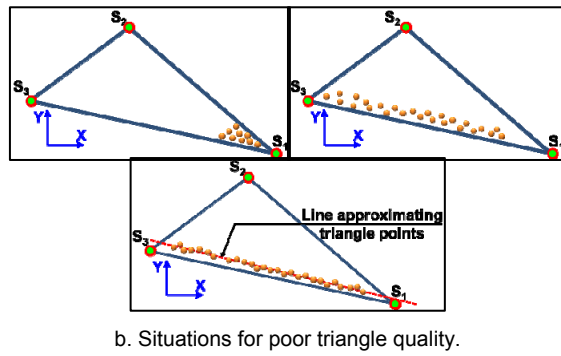


Figure 12. Triangle quality.

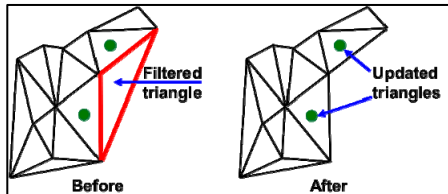


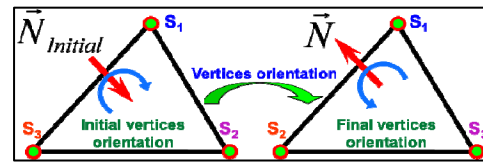
Figure 13. Filtering poor boundary triangle and triangulation update.

Finally, the “good” triangles are stored as an STL file and used to generate 3-axis toolpaths (Figure 14.a). For this, the triangle vertices are oriented so that the Z component of the triangle unit normal vector $\vec{N}$ becomes positive (Figure 14.b).

The roughing, semi-finishing, and finishing toolpaths are generated from the STL file by considering the selected machining strategies, chosen cutters, and the geometric properties of the triangles [54-55]. For the inishing operation, the toolpath is constructed based on the intersection segments between the machining planes associated with the selected machining strategy and the triangles.

facet normal	N_x	N_y	N_z
outer loop			
vertex	X_1	Y_1	Z_1
vertex	X_2	Y_2	Z_2
vertex	X_3	Y_3	Z_3
endloop			
endfacet			

a. STL file structure.



b. Vertices orientation.

Figure 14. STL file.

3.10 Global process flowchart

Table 1 and Figure 15 show the parameters used in the approach with their typical values and the global process, respectively. The stages of the approach are :

Stage 1: Select the best initial approximation and generate the initial 3D regular triangulation.

• **Stage 2:** Advances to stage 4 if the *Error* is less than the accuracy (*Precision*). If not, begin the process of adaptive subdivision. In this stage, each triangle is considered independently from all triangles and subdivided until it satisfies the accuracy (*Precision*). The subdivision process for all triangles is described below:

For each triangle:

If ($area_tr \geq area_min_tr$), then identify the case of the triangle from Case 1 to Case 8.

If (triangle error > *Precision*), then the triangle is in Case 9.

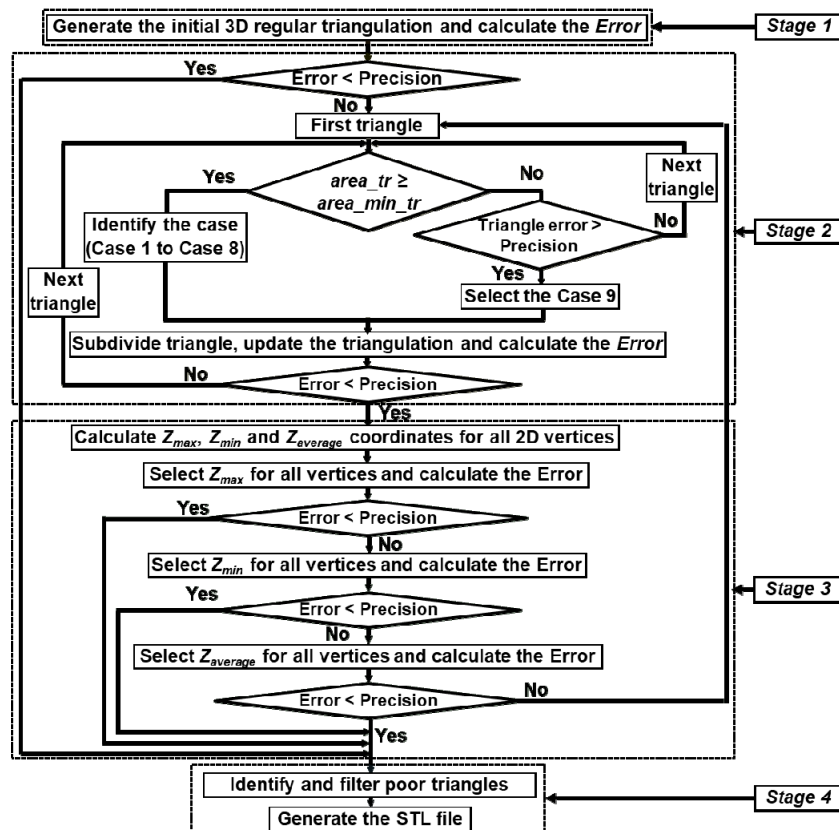


Figure 15. Flowchart of the overall process of the proposed approach.

For the selected case, do the following steps:

- Subdivide the triangle and calculate associated approximation errors.
- Subdivide its neighbors and calculate associated approximation errors.
- Update triangles' neighbors.
- Calculate the Boolean variables and the threshold distances for these created and modified triangles.
- Adjust Boolean variables.
- Calculate the maximum error *Error* for all independent triangles.

If ($Error < Precision$) then stop the subdivision process.

Else go to the next triangle.

• **Stage 3:** Calculate Z_{max} , Z_{min} , and $Z_{average}$ coordinates for unfiltered vertices and update 3D triangles with one of these possibilities. Stop the subdivision process and pass to stage 4 if the *Error* for one possibility is less than the accuracy (*Precision*). Otherwise, go to stage 2 to refine triangulation.

• **Stage 4:** Filter poor triangles and generate the STL file.

Table 1. Parameters of the approach and their typical values.

Parameters	Typical values
n_x : number of cells along X-axis	100 to 200
n_y : number of cells along Y-axis	100 to 200
L_x : initial lengths of the triangle's segments along X-axis	depends on the surface geometry: 1 mm to 50 mm
L_y : initial lengths of the triangle's segments along Y-axis	depends on the surface geometry: 1 mm to 50 mm
$Nseg_x$: number of segments along X-axis	equal to L/L_x
$Nseg_y$: number of segments along Y-axis	equal to L/L_y
$area_min_tr$: minimum triangle area to homogenize point distribution	0.1 mm^2 to 0.05 mm^2
$percent_subd$: quantify how far triangle points are from a specific point	20% to 30%
<i>Precision</i> : accuracy of the triangles' approximation	depends on the surface functionality: 0.5 mm to 0.005 mm
T_{height}/P_{height} : metric for filtering triangles	less than 10

4. VALIDATION RESULTS AND DISCUSSIONS

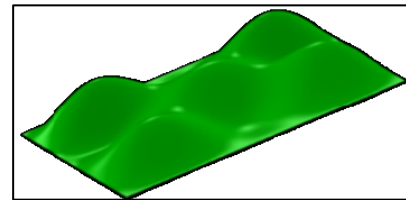
The proposed approach is implemented in a Windows object-oriented environment using the C++ programming language and the OpenGL graphics library. The developed software deals with both trimmed and untrimmed sculptured surfaces defined by a 3D point cloud for 3-axis machining. To demonstrate its efficacy, we perform validations first on theoretical sculptured surfaces and then on real, intricate ones. The calculations were performed on a workstation with the following specifications: an Intel Core i9-10900K CPU

3.70 GHz, 32.0 GB of RAM, and an NVIDIA GeForce RTX 3070 Graphics Card, running Windows 10.

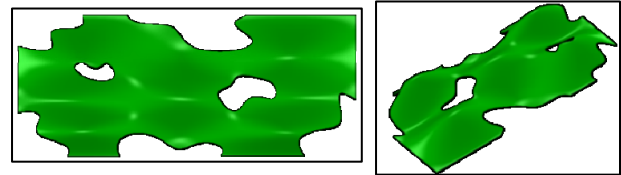
For the validation, the parameters to set are the numbers of cells n_x and n_y along the X-axis and Y-axis, the number of segments $Nseg_x$ and $Nseg_y$ along the X-axis and Y-axis, the minimum area of a triangle $area_min_tr$, the parameter of subdivision $percent_subd$, and the imposed accuracy *Precision*. In the conducted validations, for each value of $Nseg_x$ and $Nseg_y$, various precision levels *Precision*, are selected, starting from low accuracy to high accuracy.

4.1 Theoretical sculptured surfaces

The first validations are performed on two theoretical sculptured surfaces with different geometric complexity. One is untrimmed, while the other is trimmed (Figure 16). The latter has a largely more complicated geometry with two holes and five intricate cutouts at its borders. Their CAD models are randomly sampled by 3D points (Figure 17). Table 2 gives point cloud parameters.

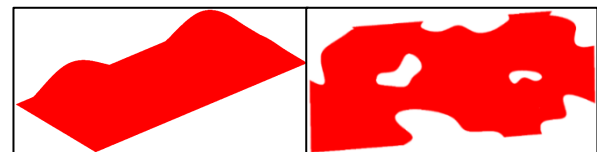


a. First surface.



b. Second surface with a projection onto the XY plane and a 3D view.

Figure 16. CAD models of the considered sculptured surfaces.



a. First point cloud.

b. Second point cloud.

Figure 17. Point clouds of the considered sculptured surfaces.

Table 2. Parameters of the point clouds.

	Number of points	Raw part dimensions
First point cloud	13 936 807	89 mm × 40 mm × 13 mm
Second point cloud	18 743 100	178 mm × 80 mm × 26 mm

For these surfaces, the numbers of cells n_x and n_y along the X-axis and Y-axis are set to 200 and 100, respectively. For all tests, $area_min_tr$ and $percent_subd$ are set to 0.05 mm^2 and 30%, respectively.

First point cloud:

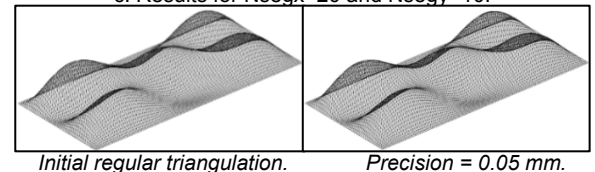
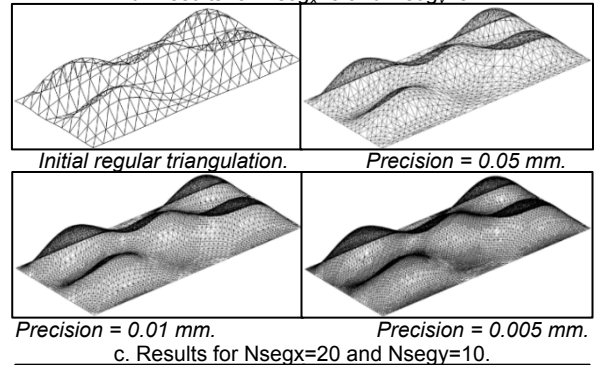
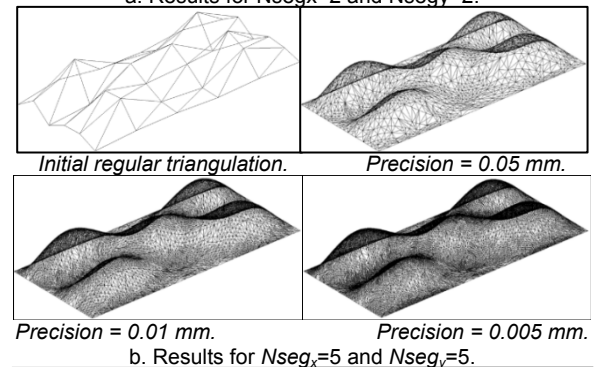
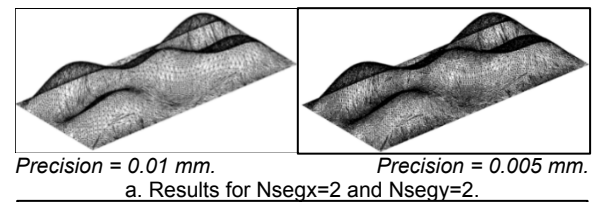
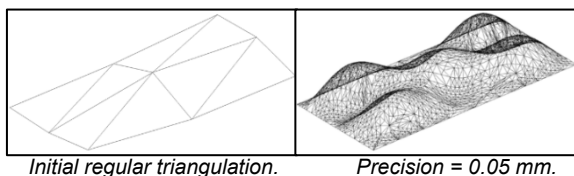
Table 3. Results for the first point cloud.

	Case 1	Case 2	Case 3	Case 4
	$Nseg_x=2$ ($L_x=44.5$ mm), $Nseg_y=2$ ($L_y=20$ mm)	$Nseg_x=5$ ($L_x=17.8$ mm), $Nseg_y=5$ ($L_y=8$ mm)	$Nseg_x=20$ ($L_x=4.45$ mm), $Nseg_y=10$ ($L_y=4$ mm)	$Nseg_x=100$ ($L_x=0.89$ mm), $Nseg_y=50$ ($L_y=0.8$ mm)
Regular triangulation				
Error (mm)	9.45864	3.61489	1.16397	0.07866
Number of triangles and vertices	8 9	50 36	400 231	10000 5151
Ratio T_{height}/P_{height}	3.47	4.89	4.71	0.94
Processing time	7 s	7 s	7 s	8 s
Adaptive triangulation with Precision = 0.05 mm				
Error (mm)	0.04985	0.04992	0.04990	0.04912
Number of triangles and vertices	6195 3143	5923 3009	4992 2561	10190 5246
Ratio T_{height}/P_{height}	2.88	2.12	2.27	0.94
Processing time	55 s	39 s	21 s	1 s
Adaptive triangulation with Precision = 0.01 mm				
Error (mm)	0.00999	0.00999	0.00999	0.00998
Number of triangles and vertices	29093 14639	28889 14528	24026 12160	23090 11730
Ratio T_{height}/P_{height}	1.44	1.18	1.01	0.94
Processing time	1 min 14 s	58 s	37 s	10 s
Adaptive triangulation with Precision = 0.005 mm				
Error (mm)	0.00500	0.00500	0.00500	0.00499
Number of triangles and vertices	58108 29212	55641 27956	47874 24136	47567 24036
Ratio T_{height}/P_{height}	1.47	1.15	0.58	0.94
Processing time	1 min 42 s	1 min 20 s	53 s	27 s

Table 3 and Figure 18 present the results for various values of $Nseg_x$ (L_x) and $Nseg_y$ (L_y). The results clearly show that even with a low number of segments, the sculptured surface shape is accurately recovered. The T_{height}/P_{height} ratio values are low (less than 2.88) for all cases, indicating that the produced triangles are of good quality. Thus, an STL file can be created for use in 3-axis machining.

For an imposed accuracy *Precision*, adaptive triangulation processing time decreases as the number of initial segments increases. Hence, this number is critical, and it is recommended to start the process with a judicious choice of it to reduce the processing time. Moreover, the results reveal a significant reduction in the number of triangles' vertices, which exceeds 99% compared to the initial point number. Even with a reduced number of triangles, the required accuracy is satisfactory, proving the effectiveness of the approach.

The approach is compared to regular triangulation for an accuracy of 0.005 mm. Using the trial and error method, the accuracy of a regular triangulation is satisfied when the numbers of segments $Nseg_x$ and $Nseg_y$ are 500 and 250, respectively. The resulting regular triangulation consists of 250 000 triangles and 125 751 vertices, with a T_{height}/P_{height} ratio of 0.18. Table 4 presents a quantitative comparison of regular and adaptive triangulations. The results clearly show that the proposed approach reduces the number of triangles and vertices by more than 76%.



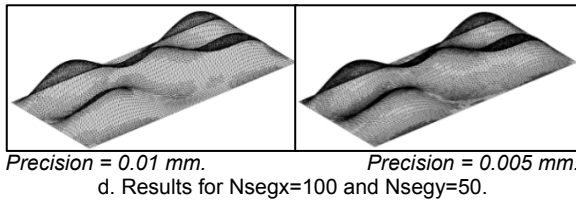


Figure 18. Results for the first point cloud.

Table 4. Comparison of regular and adaptive triangulations for the first point cloud.

	Case 1	Case 2	Case 3	Case 4
% reduction in the triangles number	76.76%	77.43%	80.85%	80.97%
% reduction in the number of vertices	76.77%	77.77%	80.81%	80.89%

This efficiency is due to the adaptive creation of triangles based on the geometric complexity of the original sculptured surfaces, as opposed to regular triangulation.

Second point cloud:

Before presenting the results for the second point cloud, it is imperative to examine how the parameters *area_min_tr* and *percent_subd* affect the surface's boundary recovery accuracy and the total number of triangles.

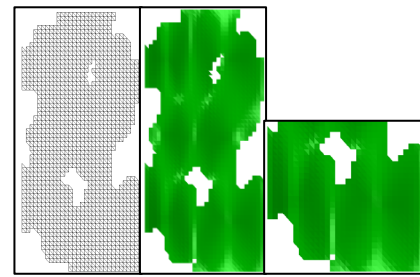
- **Impact of the parameter *area_min_tr*:** to study the effect of this parameter, the values of *Precision*, *percent_subd*, *Nsegx*, and *Nsegy* are set equal to 0.05 mm, 30%, 60, and 30, respectively. The results for different *percent_subd* values are shown in Table 5 and Figure 19. In light of the obtained results, the surface boundaries are better recovered, and the number of triangles generated increases the smaller the parameter *area_min_tr*.

Table 5. Impact of the parameter *area_min_tr* on the total number of triangles and vertices.

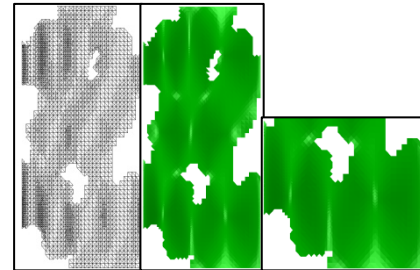
	Triangles number	Vertices number
Regular triangulation	2956	1606
<i>area_min_tr</i> =4mm ²	8173	4270
<i>area_min_tr</i> =2mm ²	8433	4477
<i>area_min_tr</i> =1mm ²	8586	4615
<i>area_min_tr</i> =0.2mm ²	8904	4915
<i>area_min_tr</i> =0.05mm ²	9448	5428

- **Impact of the parameter *percent_subd*:** to examine the effect of this parameter, the values of *Precision*, *area_min_tr*, *Nsegx*, and *Nsegy* are set to 0.05 mm, 0.05 mm², 60, and 30, respectively. The results for various *percent_subd* values are shown in Table 6 and Figure 20. According to the results, the surface boundaries are better recovered, and the number of triangles generated increases the smaller the parameter *percent_subd*.

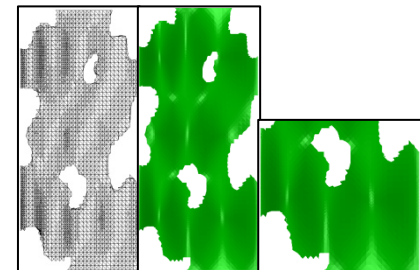
Based on the results of the previous analyses, small values of the parameters *area_min_tr* and *percent_subd* should be used to improve the accuracy of the recovery of the surfaces' boundaries. According to the set of conducted validations, the typical values for *percent_subd* and *area_min_tr* vary from 20% to 30% and 0.05 mm² to 0.1 mm², respectively. These ranges allow us to obtain a compromise between accuracy and processing time.



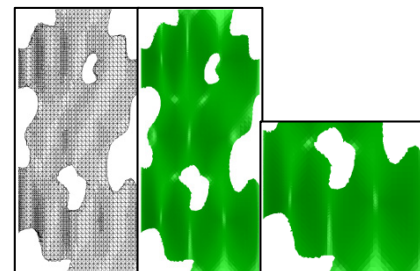
a. Regular triangulation.



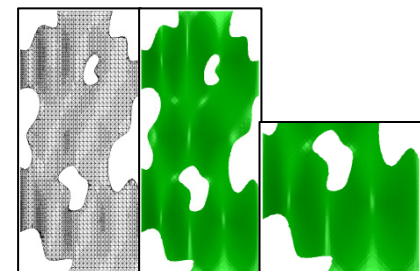
b. *area_min_tr* = 4 mm².



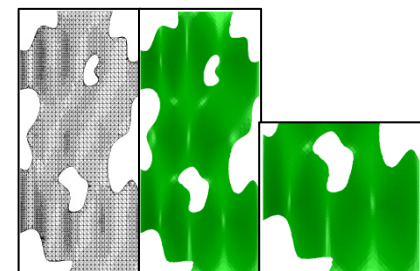
c. *area_min_tr* = 2 mm².



d. *area_min_tr* = 1 mm².



e. *area_min_tr* = 0.2 mm².

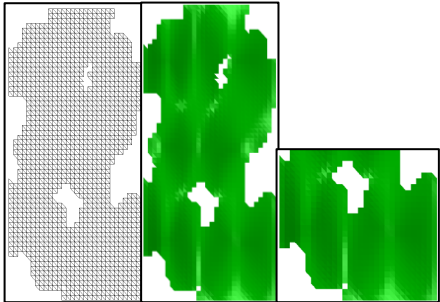


f. *area_min_tr* = 0.05 mm².

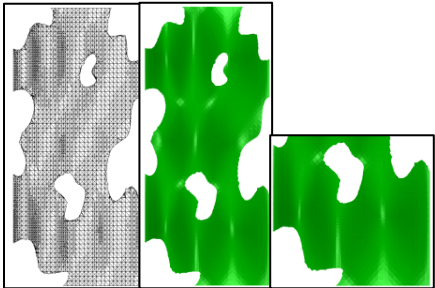
Figure 19. Impact of the parameter *area_min_tr* on the surface's boundary recovery.

Table 6. Impact of the parameter *percent_subd* on the total number of triangles and vertices.

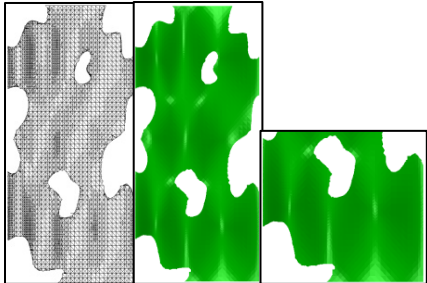
	Triangles number	Vertices number
Regular triangulation	2956	1606
<i>percent_subd</i> =60%	8972	4996
<i>percent_subd</i> =50%	9030	5050
<i>percent_subd</i> =40%	9044	5068
<i>percent_subd</i> =30%	9448	5428
<i>percent_subd</i> =20%	9978	5895



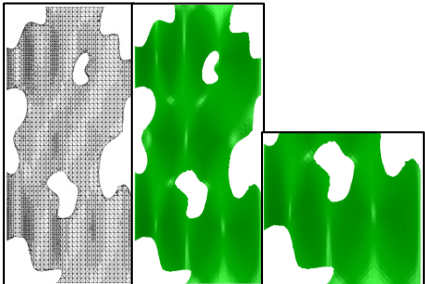
a. Regular triangulation.



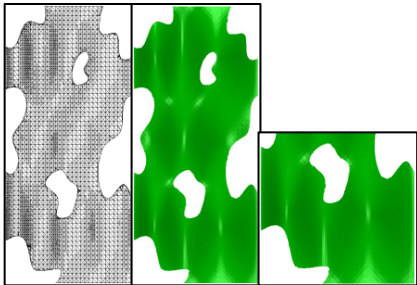
b. percent_subd = 60%.



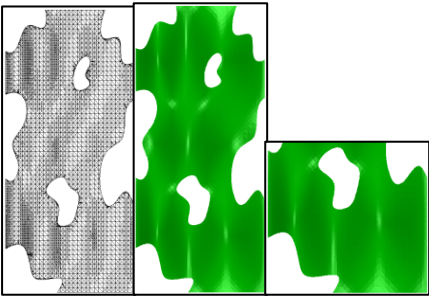
c. percent_subd = 50%.



d. percent_subd = 40%.



e. percent_subd = 30%.

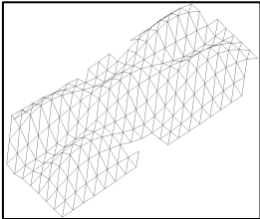


f. percent_subd = 20%.

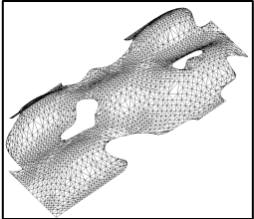
Figure 20. Impact of the parameter *percent_subd* on the surface's boundary recovery.

Table 7 and Figure 21 show the results for different values of $Nseg_x (L_x)$ and $Nseg_y (L_y)$. In Figure 21.a, the initial regular triangulation doesn't contain the two holes because the initial numbers of segments, $Nseg_x$ and $Nseg_y$, are too coarse. However, the final results, which were obtained for various accuracies, all properly exclude the triangulation's holes. The mechanism responsible for these results is the adaptive subdivision, which considers the different cases for each triangle and its parameters. This result demonstrates that even if the proposed approach starts with a small number of segments, it can effectively identify and recover the sculptured surface shape, its boundaries, and hole boundaries. The more the number of segments increases, the closer the holes and the cutouts are to the theoretical ones. Furthermore, as the precision decreases, triangles are highly subdivided in areas of large curvature variations, and the surface's boundaries are well identified and recovered. This demonstrates that the approach can deal with extremely intricate, trimmed, sculptured surfaces.

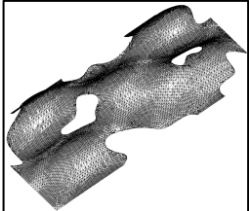
For all considered cases, the T_{height}/P_{height} ratio is low (less than 4.97), indicating that triangles are of good quality. The reduction percentage between the initial point number and the generated vertices number exceeds 99%. Even with a low number of triangles, the required precision is satisfied, demonstrating the effectiveness of the approach. It is important to notice that for the same imposed accuracy, the processing time for the adaptive triangulation decreases as the number of initial segments increases. Consequently, this number is critical and must be chosen carefully.



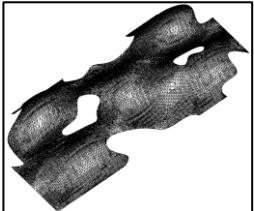
Initial regular triangulation.



Precision = 0.05 mm.



Precision = 0.01 mm.



Precision = 0.005 mm.

a. Results for $Nseg_x=20$ and $Nseg_y=10$.

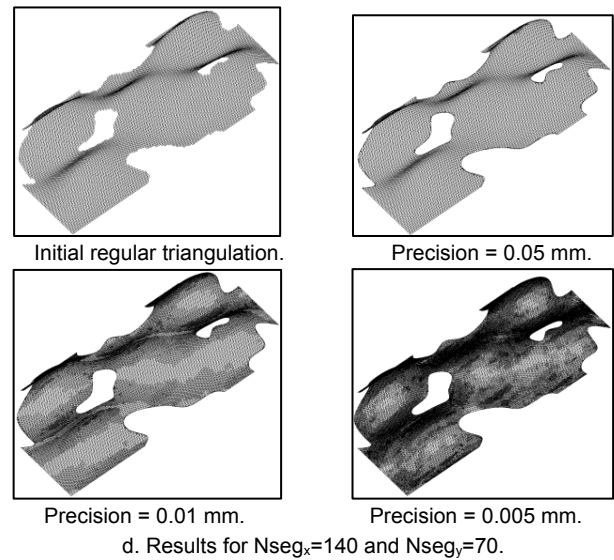
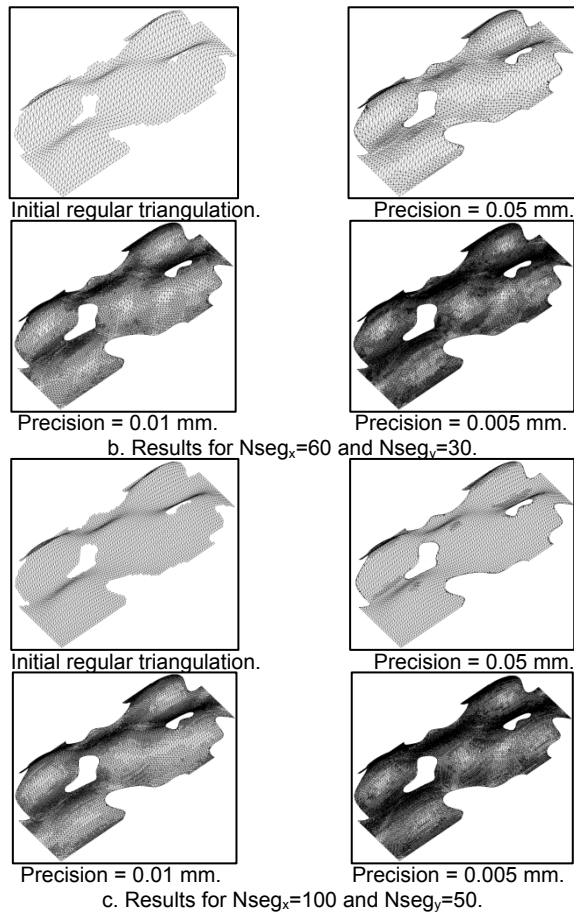


Figure 21. Results for the second point cloud.

A comparison is done with regular triangulation for an accuracy of 0.005 mm. After trial and error, the accuracy of a regular triangulation is satisfied when the numbers of segments $Nseg_x$ and $Nseg_y$ are 700 and 350, respectively. The resulting triangulation consists of 377 231 triangles and 190 351 vertices, with a T_{height}/P_{height} ratio of 0.64. Table 8 gives a quantitative comparison between regular and adaptive triangulations. The results show that the reduction of the number of triangles and vertices exceeds 72%.

Table 7. Results for the second point cloud.

	Case 1 $Nseg_x=20$ ($L_x=8.9$ mm), $Nseg_y=10$ ($L_y=8$ mm)	Case 2 $Nseg_x=60$ ($L_x=2.96$ mm), $Nseg_y=30$ ($L_y=2.66$ mm)	Case 3 $Nseg_x=100$ ($L_x=1.78$ mm), $Nseg_y=50$ ($L_y=1.6$ mm)	Case 4 $Nseg_x=140$ ($L_x=1.27$ mm), $Nseg_y=70$ ($L_y=1.14$ mm)
Regular triangulation				
Error (mm)	2.8001	0.41111	0.21895	0.08962
Number of triangles and vertices	356 212	2956 1606	7969 4203	15440 8028
Ratio T_{height}/P_{height}	7.86	7.52	4.34	3.10
Processing time	9 s	10 s	10 s	11 s
Adaptive triangulation with Precision = 0.05 mm				
Error (mm)	0.04983	0.04997	0.04975	0.04881
Number of triangles and vertices	9810 5612	9448 5428	10743 6156	17101 9446
Ratio T_{height}/P_{height}	4.97	4.91	3.95	2.52
Processing time	43 s	18 s	7 s	3 s
Adaptive triangulation with Precision = 0.01 mm				
Error (mm)	0.00999	0.00999	0.00999	0.00999
Number of triangles and vertices	44220 23174	42974 22438	43700 22871	41250 21686
Ratio T_{height}/P_{height}	3.96	3.81	2.30	2.44
Processing time	1 min 14 s	52 s	37 s	26 s
Adaptive triangulation with Precision = 0.005 mm				
Error (mm)	0.00499	0.00500	0.00499	0.00500
Number of triangles and vertices	103595 53130	100224 51332	99332 50814	98439 50396
Ratio T_{height}/P_{height}	3.96	2.80	1.99	2.44
Processing time	2 min 41 s	2 min 18 s	1 min 55 s	1 min 39 s

Table 8. Comparison of the regular and adaptive triangulations for the second point cloud.

	Case 1	Case 2	Case 3	Case 4
% reduction in the triangles number	72.54%	73.43%	73.97%	73.90%
%reduction in the number of vertices	72.09%	73.03%	73.31%	73.52%

Table 9. Results for the real point cloud.

	Case 1	Case 2	Case 3	Case 4
	$N_{seg_x}=25$ ($L_x=8$ mm), $N_{seg_y}=25$ ($L_y=8$ mm)	$N_{seg_x}=80$ ($L_x=2.5$ mm), $N_{seg_y}=80$ ($L_y=2.5$ mm)	$N_{seg_x}=100$ ($L_x=2$ mm), $N_{seg_y}=100$ ($L_y=2$ mm)	$N_{seg_x}=200$ ($L_x=1$ mm), $N_{seg_y}=200$ ($L_y=1$ mm)
Regular triangulation				
Error (mm)	3.7353	0.98070	0.73055	0.48852
Number of triangles and vertices	1049 577	10106 5234	15699 8076	61911 31422
Ratio T_{height}/P_{height}	37.77	2743.07	102.96	9.84
Processing time	14 s	14 s	15 s	19 s
Adaptive triangulation with Precision = 0.5 mm				
Error (mm)	0.49986	0.49997	0.49661	0.48852
Number of triangles and vertices	7319 5275	13429 8273	18969 11110	61911 31422
Ratio T_{height}/P_{height}	5.25	4.96	4.96	9.84
Processing time (s)	28 s	7 s	5 s	0
Adaptive triangulation with Precision = 0.1 mm				
Error (mm)	0.09998	0.09999	0.09999	0.09994
Number of triangles and vertices	30203 17170	29932 16941	32375 18239	68919 36528
Ratio T_{height}/P_{height}	4.97	4.73	4.85	4.95
Processing time	1 min 9 s	24 s	22 s	19 s
Adaptive triangulation with Precision = 0.05 mm				
Error (mm)	0.04999	0.04999	0.04999	0.04999
Number of triangles and vertices	60988 33113	57441 31022	58432 31632	83125 43727
Ratio T_{height}/P_{height}	4.95	4.09	4.81	2.75
Processing time	1 min 46 s	55 s	52 s	43 s

4.2 Real sculptured surfaces

The second validation is conducted on a real object, a “mask” composed of highly intricate sculptured surfaces (Figure 22.a). The scanning is performed using a measuring arm, “ROMER”, with 07 degrees of freedom from Hexagon Metrology equipped with a laser scanner (Figure 22.b). Figure 22.c and Figure 22.d represent the scanning operation of the mask and the acquired point cloud, respectively. Because of the mask's intricate geometry, the point cloud is dense to efficiently retrieve all its details. The acquired point cloud is composed of 25618568 points, and its raw part dimensions are 200 mm × 200 mm × 43 mm.



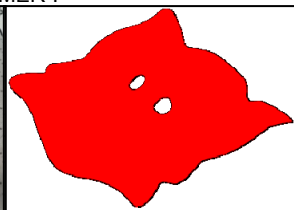
a. Mask.



b. Measuring arm “ROMER”.



c. Laser scanning of the mask.



d. Acquired point cloud.

Figure 22. Point cloud of the highly complex real part “mask”.

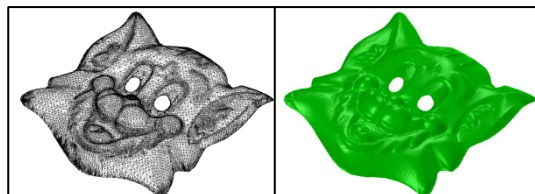
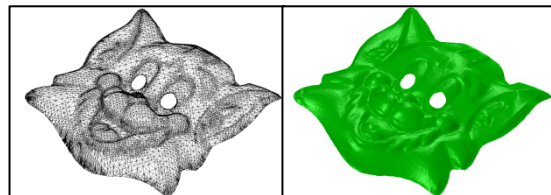
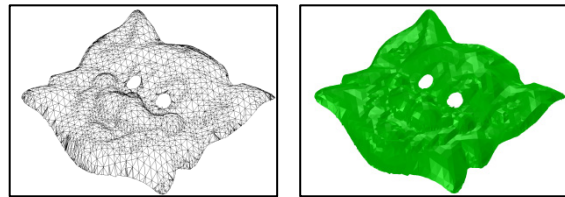
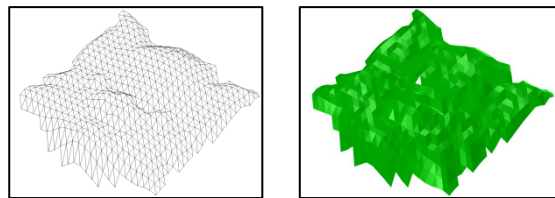
For this case, the numbers of cells n_x and n_y along the X-axis and Y-axis are set to 100 and 100, respectively. The values of $area_min_tr$ and $percent$

$_{subd}$ for all tests are set to 0.05 mm² and 20%, respectively, based on the prior analysis of their effects on the surface's boundary recovery.

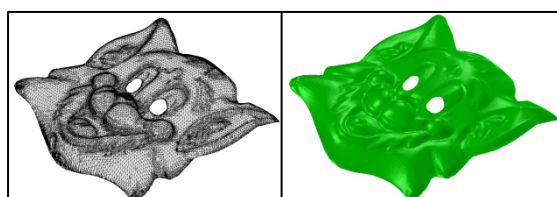
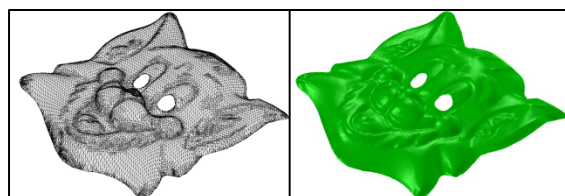
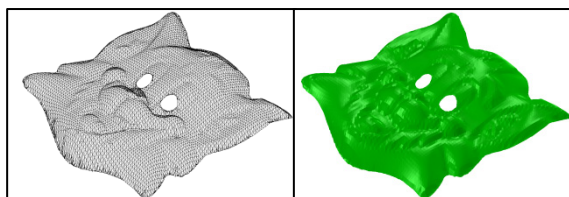
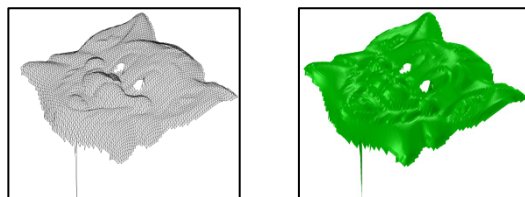
Table 9 and Figure 23 show the results for different values of N_{seg_x} (L_x) and N_{seg_y} (L_y). In Figure 23.a, the initial regular triangulation doesn't contain any detail of the mask since the initial numbers of segments, N_{seg_x} and N_{seg_y} , are too coarse. However, all of the details (eyes, ears, nose, etc.) are correctly recovered in the final results, which were obtained for various accuracies. This result illustrates that the proposed approach can effectively identify and recover the sculptured surface shape, its boundaries, and hole boundaries even if it begins with a reduced number of segments. Furthermore, as the precision decreases, triangles are highly subdivided in areas of large curvature variations, and the surface's boundaries are well identified and recovered.

In Figure 23.b, the T_{height}/P_{height} ratio is very high (2743.07) for the initial regular triangulation because one boundary triangle is almost vertical due to the point distribution inside it. However, once the adaptive subdivision is launched for various accuracies, this ratio decreases below 4.96, indicating that the resulting triangles are of high quality.

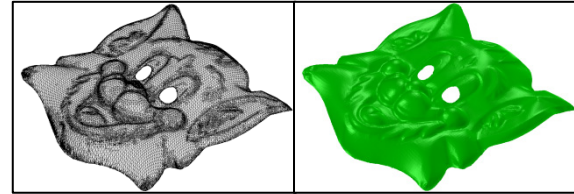
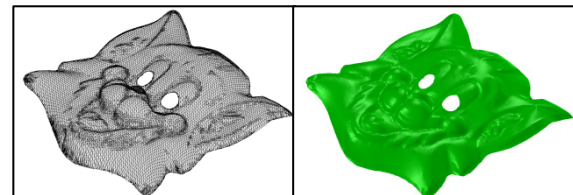
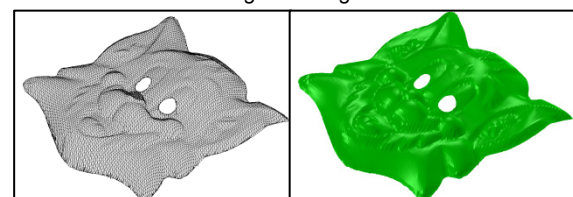
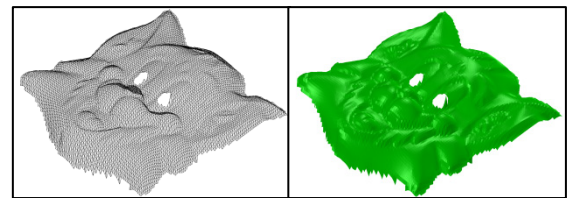
It is worth noting that as the number of initial segments increases, the initial regular triangulation is closer to the real shape of the mask, and the processing time for adaptive triangulation decreases. According to the obtained results, the mask's sculptured surface shape, borders, and hole boundaries are well identified and recovered. Besides, in regions with high variations of curvatures, triangles are subdivided recursively to be able to retrieve the small details. This achievement shows the approach's efficacy and proves its capability of handling extremely intricate surfaces.



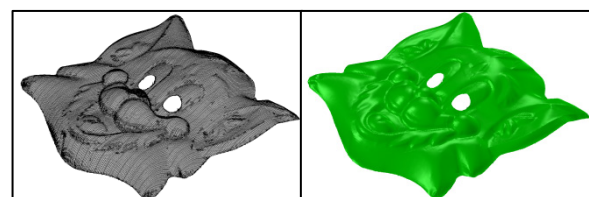
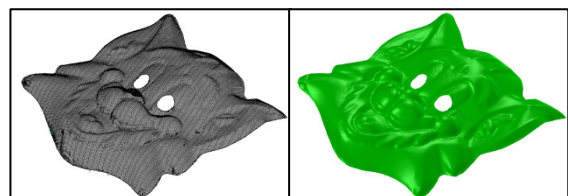
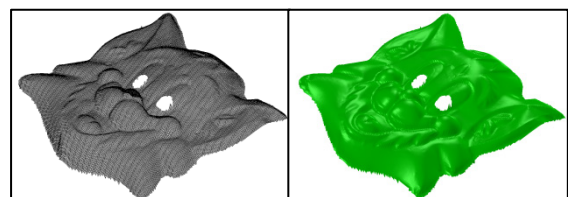
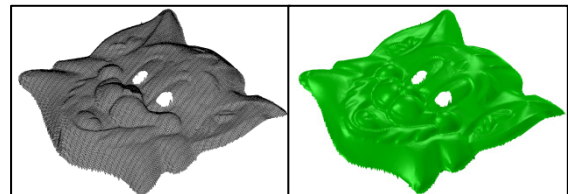
a. Results for $N_{seg_x}=25$ and $N_{seg_y}=25$.



b. Results for $N_{seg_x}=80$ and $N_{seg_y}=80$.



c. Results for $N_{seg_x}=100$ and $N_{seg_y}=100$.



d. Results for $N_{seg_x}=200$ and $N_{seg_y}=200$.

Figure 23. Results for the point cloud of the mask.

5. CONCLUSION

In this paper, we have presented a novel and efficient approach for generating a 3D triangulation using the least squares method with a reduced number of triangles while maintaining predefined accuracy from a 3D point cloud for 3-axis sculptured surface machining. It combines both regular and adaptive triangulations to maximize efficiency. Besides, it explicitly integrates approximation accuracy, adaptive subdivision, point distributions within triangles, and triangle quality. By analyzing the point cloud distribution within the triangle relative to specific key points, the triangle is adaptively and recursively subdivided using various proposed subdivision schemes. For the adaptive subdivision, triangles are generated based on the original sculptured surfaces' geometric complexity.

We have validated the approach's efficiency on many 3D point clouds representing intricate theoretical and real trimmed and untrimmed sculptured surfaces, proving its performance for both triangulation and processing time. The primary advantage of the approach lies in significantly reducing the number of the generated triangles, thereby significantly reducing the reproduction cycle of parts with sculptured surfaces and hence the costs, including cutting tool selection, machining strategy choice, cutting conditions adaptation, and toolpath generation.

However, we must point out two limitations of our approach. Firstly, it cannot address vertical regions due to inherent issues with the least squares method. Consequently, points on vertical planes must be filtered out before proceeding with the approach. Secondly, the point cloud must be dense to recover efficiently the surface and its boundaries.

As an immediate research perspective, it is intended to use the generated triangles for optimizing sculptured surface machining processes. To attain this objective, several research directions can be considered:

- Segmenting the triangles into zones based on the local shapes of their vertices, and then identifying the optimal shapes and sizes of the cutters.
- Selecting and combining the appropriate machining strategies.
- Identifying the best combination of ball-end cutters to minimize the total machining time.
- Combining all forms of cutters (ball-end cutters, flat-end cutters, and fillet-end cutters) by taking into account local shapes of the generated triangles.
- Adapting the cutting parameters (feedrate, spindle speed) to minimize vibrations and to improve surface finish.

ACKNOWLEDGMENTS

We extend our sincere gratitude to Mr. Messaoud BENGHERABI and Mr. Mustapha AOUACHE of the CDTA research center (Algiers, Algeria) for their invaluable assistance, comments, and recommendations, which have significantly enhanced the paper quality.

REFERENCES

- [1] Redonnet, J. M., Gamboa, V. A., Traslosheros M. A., Segonds S.: Optimization of free-form surface machining using parallel planes strategy and torus milling cutter, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, Vol. 232, No. 2, pp. 240-250, 2018.
- [2] Stratogiannis, F. I., Galanis, N. I., Karkalos, N. E., Markopoulos, A. P.: Optimization of the manufacturing strategy, machining conditions, and finishing of a radial impeller, Machines, Vol. 8, No.1, 2020.
- [3] Rahul, A. M., Gupta, T.V.K., Ramkumar, J.: A comprehensive review of free-form surface milling—Advances over a decade, Journal of Manufacturing Processes, Vol. 62, pp. 132-167, 2021.
- [4] Choi, B.K., Jun, C.S.: Ball-end cutter interference avoidance in NC machining of sculptured surfaces, Computer-Aided Design, Vol. 21, No. 6, pp. 371-378, 1989.
- [5] Choi, B. K., Lee, C. S., Hwang, J. S., Jun, C. S.: Compound surface modeling and machining, Computer-Aided Design, Vol. 20, No. 3, pp. 127-136, 1988.
- [6] Duc, E.: *Usinage de formes gauches: contribution à l'amélioration de la qualité des trajectoires d'usinage*. PhD Thesis. Ecole Normale Supérieure de Cachan, France, 1998.
- [7] Yang, D. C., Han, Z.: Interference detection and optimal tool selection in 3-axis NC machining of free-form surfaces, Computer-Aided Design, Vol. 31, No. 5, pp. 303-315, 1999.
- [8] Park, S. C.: Tool path generation for Z-constant contour machining, Computer-Aided Design, Vol. 35, No. 1, pp. 27-36, 2003.
- [9] Lasemi, A., Xue, D. and Gu, P.: Recent development in CNC machining of freeform surfaces: a state-of-the-art review, Computer-Aided Design, Vol. 42, No. 7, pp. 641-654, 2010.
- [10] Globocki-Lakic, G., Sredanovic, B., Jotic, G., Gotovac, S.: A Comparative Analysis of Milling Strategies of Complex Geometry Surfaces, FME Transactions, Vol. 50, No. 4, pp. 623-634, 2022.
- [11] Hendriko, H.: A hybrid Analytical and Discrete Based Methodology to Calculate Path Scallop of Helical Toroidal Cutter in Five-Axis Milling, FME Transactions, Vol. 46, No. 4, pp. 552-559, 2018.
- [12] Mladenovic, G. M., Tanovic, L. M., Ehmann, F. K.: Tool Path Generation for Milling of Free Form Surfaces With Feedrate Scheduling, FME Transactions, Vol. 43, No. 1, pp. 9-15, 2015.
- [13] Duvedi, R. K., Bedi, S., Batish, A., Mann, S.: A multipoint method for 5-axis machining of triangulated surface models, Computer-Aided Design, Vol. 52, pp. 17-26, 2014.
- [14] Park, S.C.: Sculptured surface machining using triangular mesh slicing, Computer-Aided Design, Vol. 36, pp. 279-288, 2004.

- [15] Daoshan O. Y. & Hsi-Yung. F.: Machining triangular mesh surfaces via mesh offset based tool paths, *Computer-Aided Design and Applications*, Vol. 5, No.1-4, pp. 254-265, 2008.
- [16] Raja, V., Fernandes, K. J.: *Reverse engineering: an industrial perspective*, Springer, 2008.
- [17] Chang, Q., Ma, W. and Deng, C.: Constrained least square progressive and iterative approximation (CLSPIA) for B-spline curve and surface fitting. *Visual Computer*, Vol. 40, pp. 4427-4439, 2024.
- [18] Yao, Z., Hu, Q.: Accelerated local progressive-iterative approximation methods for curve and surface fitting, *Visual Computer*, Vol. 41, pp. 5979-5993, 2025.
- [19] Li, P., Song, D., Li, J., Zhou, L., Ma, J.: A method for fast reconstruction of closed section workpiece surface using point clouds, *Advances in Mechanical Engineering*, Vol. 16, No. 2, pp. 1-13, 2024.
- [20] Peng, Y. H., Yin. Z. W.: Direct tool path regeneration for physical object modification from digitized points in reverse engineering. *Int J AdvManufTechnol*, Vol. 33, pp. 1204-1211, 2007.
- [21] Berger, M., Tagliasacchi, A., Seversky, L., Alliez, P., Guennebaud, G., Levine, J., Sharf, A., Silva, C.: A Survey of Surface Reconstruction from Point Clouds, *Computer Graphics Forum*, Vol. 36, No. 1, pp. 301-329, 2016.
- [22] Sulzer, R., Marlet, R., Vallet, B., Landrieu, L.: A Survey and Benchmark of Automatic Surface Reconstruction From Point Clouds, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Vol. 47, No. 3, pp. 2000-2019, 2025.
- [23] Elshakhs, Y. S., Deliparaschos, K. M., Charalambous, T. Oliva, G. and Zolotas, A.: A Comprehensive Survey on Delaunay Triangulation: Applications, Algorithms, and Implementations Over CPUs, GPUs, and FPGAs, *IEEE Access*, Vol. 12, pp. 12562-12585, 2024.
- [24] Zhang, C., Tao, W.: Learning Meshing from Delaunay Triangulation for 3D Shape Representation, *International Journal of Computer Vision*, Vol. 133, pp. 3413-3436, 2025.
- [25] Lv, C., Lin, W. and Zhao, B.: Voxel Structure-Based Mesh Reconstruction From a 3D Point Cloud, *IEEE Transactions on Multimedia*, Vol. 24, pp. 1815-1829, 2022.
- [26] Wang, W., Deng, Y., Li, Z., Liu, Y., Lei, N.: MergeNet: Explicit Mesh Reconstruction from Sparse Point Clouds via Edge Prediction, *IEEE International Conference on Multimedia and Expo (ICME)*, Niagara Falls, ON, Canada, pp. 1-6, 2024.
- [27] Erler, P., Fuentes-Perez, L., Hermosilla, P., Guerrero, P., Pajarola, R., Wimmer, M.: PPSurf: Combining Patches and Point Convolutions for Detailed Surface Reconstruction, *Computer Graphics Forum*, Vol. 43: No. 1, pp. 1-12, 2024.
- [28] Marković, V.; Jakovljević, Ž.: Recognition of one class of surfaces from structured point cloud, *FME Transactions*, Vol. 45, No. 4, pp. 481-490, 2017.
- [29] Lin, A. C., Liu, H. T.: Automatic generation of NC cutter path from massive data points, *Computer-Aided Design*, Vol. 30, pp. 77-98, 1998.
- [30] Choi, J. W., Hurl, S. M. and Lee, S. H.: Free-form surface generation from measuring points using laser scanner, *International Journal of the Korean Society of Precision Engineering*, Vol. 3, No. 4, pp. 15-23, 2002.
- [31] Breteau, P., Thiébaud, F., Bourdand, P., Lartigue, C.: A First approach for rapid copying of free form surfaces in 5 axis machining, *International Conference on Integrated Design and Manufacturing in Mechanical Engineering (IDMME)*, Grenoble, France, pp. 1-12, 2006.
- [32] Makki. M., Lartigue. C., Tournier. C., Thiébaud. F.: Direct duplication of physical models in discrete 5-axis machining, *Virtual and Physical Prototyping*, Vol. 3, No. 2, pp. 93-103, 2008.
- [33] Chui, K. L., Chiu, W.K. and Yu, K. M.: Direct 5-axis tool-path generation from point cloud input using 3D biarc fitting, *Robotics and Computer-Integrated Manufacturing*, Vol. 40, pp. 270-286, 2008.
- [34] OuYang, D., Van Nest, B.A., Feng, H. Y.: Automatic ball-end milling tool selection from 3d point cloud data, *International Conference on Flexible Automation and Intelligent Manufacturing*, Toronto, Canada, pp. 253-260, 2004.
- [35] OuYang, D., Van Nest, B. A., Feng, H. Y.: Determining gouge free ball-end mills for 3D surface machining from point cloud data, *Robotics and Computer Integrated Manufacturing*, Vol. 21, No. 4-5, pp. 338-345, 2005.
- [36] Feng, H. Y., Teng, Z.: Iso-planar piecewise linear NC tool path generation from discrete measured data points, *Computer-Aided Design*, Vol. 37, pp. 55-64, 2005.
- [37] Bey, M., Azouaoui. K., Boutassouna. M.: Sculptured surfaces subdivision from 3d cloud of points and association of optimum ball cutters, *Procedia CIRP*, Vol. 77, pp. 582-585, 2018.
- [38] Shen, L. Y. et al.: A framework from point clouds to workpieces, *Visual Computing for Industry, Biomedicine, and Art*, Vol. 5, No. 21, pp. 1-18, 2022.
- [39] Liao, J., Huang, Z.: Data model-based toolpath generation techniques for CNC milling machines, *Frontiers in Mechanical Engineering*, Vol. 10, pp. 1-12, 2024.
- [40] Park, S. C., Chung, Y. C.: Tool path generation from measured data, *Computer-Aided Design*, Vol. 35, pp. 467-475, 2003.
- [41] Chui. K. L., Yu. K. M., Lee. T.C.: Direct tool-path generation from massive point input. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Vol. 216, pp. 199-206, 2002.
- [42] Teng, Z., Feng, H. Y. and Azeem. A.: Generating efficient tool paths from point cloud data via

machining area segmentation, *Int J AdvManuf Technol*, Vol. 30, pp. 254-260, 2006.

- [43] Kayal. P.: Inverse offset method for adaptive cutter path generation from point-based surface, *Int J CAD/CAM*, Vol. 7, No. 1, pp. 21-30, 2007.
- [44] Yau. H. T. and Hsu. C. Y.: Generating NC tool paths from random scanned data using point-based models, *Int J AdvManufTechnol*, Vol. 41, pp. 897-907, 2009.
- [45] Yingjie. Z., Liling. G.: Adaptive tool-path generation on point-sampled surfaces, *Precis Eng*, Vol. 35, pp. 591-601, 2011.
- [46] Liu. Y., Xia. S., Qian. X.: Direct NC path generation from discrete points to continuous spline paths, *Journal of Computing and Information Science in Engineering*, Vol. 12, No. 3, 2012.
- [47] Liu. W., Zhou. L. S. and An. L. L.: Constant scallop-height tool path generation for three-axis discrete data points machining, *Int J AdvManufTechnol*, Vol. 63, pp. 137-146, 2012.
- [48] Liu. X., Li. Y. and Li. Q.: A region-based 3 + 2-axis machining toolpath generation method for freeform surface, *Int J AdvManufTechnol*, Vol. 97, pp. 1149-1163, 2018.
- [49] Xu. J., Xu. L. and Sun. Y.: A method of generating spiral tool path for direct three-axis computer numerical control machining of measured cloud of point, *J ComputInfSciEng*, Vol. 19, No. 4, 2019.
- [50] Dhanda. M. and Pande. S. S.: Adaptive tool path planning strategy for freeform surface machining using point cloud, *Computer-Aided Design and Applications*, Vol. 16, No.2, pp. 289-307, 2019.
- [51] Mandeep. D., Aman. K. and Pande. S. S.: Pande. Adaptive spiral tool path generation for computer numerical control machining using point cloud, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, Vol. 235, No. 22, pp. 1-17, 2021.
- [52] Ericson. C.: *Real-time collision detection*, CRC Press, 2005.
- [53] Sorgente, T., Biasotti, S., Manzini, G., Spagnuolo, M.: A Survey of Indicators for Mesh Quality Assessment, *Computer Graphics Forum*, Vol. 42, No. 2, pp. 461-483, 2023.
- [54] Bey, M., Bendifallah, H., Tchantchane, Z., Driouech, A, LarbiCherif, D.: 3-Axis Roughing of Complex Parts Using Plunge Milling Strategy, *CongrèsAlgérien de Mécanique, (CAM'2019), Ghardaïa, Algeria*, 2019.
- [55] Bey, M. et al.: Finishing of Sculptured Surfaces by Combining Ball End Tools and Steepest Ascent Machining Strategy, *CongrèsAlgérien de Mécanique, (CAM'2013), Mascara, Algeria*, 2013.

NOMENCLATURE

n_x	number of cells along X-axis
n_y	number of cells along Y-axis
L_x	initial lengths of the triangle's segments along X-axis
L_y	initial lengths of the triangle's segments along Y-axis
$Nseg_x$	number of segments along X-axis
$Nseg_y$	number of segments along Y-axis
$area_min_tr$	minimum triangle area to homogenize point distribution
$percent_subd$	quantify how far triangle points are from a specific point
$Precision$	accuracy of the triangles' approximation
T_{height}/P_{height}	metric for filtering triangles

НОВИ ПРИСТУП ТРИАНГУЛАЦИЈИ 3Д ОБЛАКА ТАЧАКА ЗА ОБРАДУ СКУЛПТУРАЛНИХ ПОВРШИНА НА ТРООСНИМ CNC МАШИНАМА

М. Беј. К. Азуан

Реверзибилни инжењеринг (РЕ) представља кључни метод за дигиталну реконструкцију и репликацију скулптуралних површина на основу 3Д облака тачака. Ипак, креирање непрекидних и геометријски тачних површинских модела из дискретних података и даље је значајан изазов у пракси. Као ефикасна алтернатива, употреба триангулисаних мрежа омогућава робустан и рачунски прихватљив приступ реконструкцији оваквих површина. У овом раду представљен је нов, геометријски заснован метод за генерисање 3Д триангулације из облака тачака, при чему се постиже редукован број троуглова уз контролисану грешку апроксимације. Предложени алгоритам комбинује регуларну и адаптивну триангулацију, уз експлицитно укључивање критеријума као што су локална прецизност апроксимације, адаптивна деоба елемената, расподела тачака унутар троуглова и евалуација квалитета мреже. Методологија се састоји из четири фазе: Генерисање почетне регуларне 3Д триангулације; Адаптивна деоба троуглова на основу геометријске сложености локалне површине; Евалуација и побољшање квалитета генерисаних елемената; Извоз STL modela pogodnog za troosnu CNC obradu. Примена предложеног приступа на различитим скуповима 3Д података скулптуралних површина показала је значајно смањење укупног броја троуглова, као и оптимизацију времена припреме обраде и CNC генерисања путања алата, уз задржавање високе прецизности модела. Резултати потврђују практичну ефикасност алгоритма у контексту дигиталне реконструкције и САМ припреме комплексних површина.