International Journal of Modeling, Simulation, and Scientific Computing

Enhancement of B-Spline surface rebuilding for deformed CAD model reconstruction

Nessrine ELLOUMI*
SETIT, ISBS, University of Sfax, Tunisia
ellouminessrine@gmail.com

Aicha BEN MAKHLOUF
LATIS, University of Sousse, Tunisia
aychaa.benmakhlouf@gmail.com

Borhen LOUHICHI
Department of Mechanical Engineering, College of Engineering, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 11432, Saudi Arabia
blouhichi@imamu.edu.sa

3D reconstruction is a method used to generate a three-dimensional (3D) model of a real object using a 3D point cloud or a 3D mesh using Computer Aided Design (CAD) software or Reverse Engineering (RE) techniques. The points' coordinates are acquired by digitizing physical items using specialized instruments such as a three-dimensional measuring machine (CMM), a structured light scanner, or a LASER pulse-based 3D scanner. These scanned objects are used in many industries, such as mechanics, aeronautics, vehicle manufacturing, medicine, and art. Within this particular framework, achieving a highly accurate reconstruction of the geometric model is a challenging task that requires a significant investment of time and relies heavily on the use of modeling and reverse engineering approaches. This work introduces a novel method for reconstructing a CAD model by inserting new points into a given distorted mesh. Walton's method is employed for accurate rebuilding to ensure a high level of accuracy in the fitted surfaces. B-Spline surfaces are estimated based on the obtained 3D points, and then the geometric model is rebuilt. In order to verify the performance of the suggested approach, 3D surfaces are reconstructed, employing a 3D reconstruction algorithm to generate an accurate rebuild model. The precision of the suggested method is demonstrated by comparing the results of the geometric model reconstruction using the input mesh with the results obtained after inserting new points.

Keywords: 3D reconstruction; Computer Aided Design (CAD); Reverse Engineering (RE); B-Spline surfaces

1. Introduction

Computer Aided Design (CAD) refers to the use of computer technologies for generating three-dimensional (3D) models of physical objects. Indeed, most 3D models are designed within computer-aided design (CAD) software. In the process of visualization, data interchange, or 3D printing, it is necessary to convert the CAD

model into a 3D mesh consisting of a limited number of vertices and edges. In particular cases, the original model may become lost or distorted. Therefore, a Reverse Engineering (RE) technique [1] is essential in order to reconstruct a new representation from the discrete mesh. This method generates a geometric representation of an existing object. A reconstruction process is necessary to accurately align the input data and estimate the 3D surfaces when rebuilding the CAD model. The surface reconstruction process is crucial in various fields such as data visualization, medical imaging, computer vision, computer animation, computer graphics, digital entertainment, and augmented reality [2,3] [4]. Triangular meshes are commonly used to represent 3D data in various applications, where they describe the surface using 3D vertices and faces. The development of various modeling software, such as SolidWorks, CATIA, and ICEM SURF, has enabled the creation of complex designs. Furthermore, the use of 3D scanners, such as the Leica 3D laser scanner has led to increasing the creation of triangular meshes. However, the generated meshes are not adequate due to the density of the input vertices. Nevertheless, the quantity of these vertices may be inadequate to accurately reconstruct a precise model. In this field, numerous researchers have proposed different rebuilding methods for enhancing the quality of the input mesh. This is achieved by the process of resampling the vertices, regularizing the connectivity, and introducing new vertices. This paper introduces a novel approach for CAD model rebuilding from an input triangular mesh based on B-Spline surface reconstruction. The initial mesh is enhanced by introducing additional points using Walton's technique. Next, the B-Spline surfaces of the geometric model are rebuilt by approximating the associated control points using the Levenberg Marquardt Algorithm. Finally, the CAD model is generated. This paper is structured as follows: The second section presents a comprehensive analysis of CAD model reconstruction and 3D subdivision procedures, which involves the methods for inserting 3D points into the input data. Next, the specific phases of the proposed approach are described. The obtained results are assessed using three quality metrics for different case studies.

2. State of the art

For many years, several researchers have dedicated their efforts for the development of efficient methods to reconstruct 3D models from an input mesh. Louhichi et al. [5]introduced a technique for CAD model reconstruction given distorted mesh. This method involves approximating B-Spline surfaces using the weighted displacement estimation (WDE) method, which addresses the issue of disorganized 3D points. Moreover, Ben Makhlouf et al. [6] presented an alternative approach for reconstructing a CAD model from a distorted mesh using the Levenberg Marquardt Algorithm (LMA). An alternative algorithm [7] has been devised using the Thin Plate Spline (TPS) approach to enhance the positioning of control points for the B-Spline surface reconstruction. Park et al. [8] described a technique for generating computer-aided design (CAD) surfaces based on the designed model using finite element method

(FEM). In order to create a curve-based feature, boundary curves have been identified to rebuild the basic surface. Multiple studies have concentrated on the estimation of a surface by dividing an initial mesh [9] [10]. This subdivision technique is employed to generate additional vertices and faces based on the original data. Consequently, the subdivision scheme generates a new mesh that has a greater number of vertices and polygonal faces compared to the original input mesh. Doo and Sabin [11], as well as Catmull and Clark [12], were the first researchers to introduce subdivision methods within this particular framework. Their proposed methods generate uniform B-Spline surfaces dedicated to quadratic or cubic surfaces. Several other studies have concentrated on the partitioning of triangular and quadrilateral meshes [13]. Charles Loop [14] devised the Loop Subdivision method, which involves recursively subdividing a triangular mesh. This approach relies on the insertion of a new point between two adjacent points in order to partition each triangle into four smaller triangles. The subdivision methods are characterized by their simplicity and their ability to generate a concise representation of the reconstructed surface. Nevertheless, these approaches are limited to accurately reconstructing a precise 3D surface. Other studies have concentrated on estimating the surface by modifying the associated mesh through the use of Delaunay Triangulation (DT) based techniques [15]. These approaches rely on the DT criterion to guarantee the absence of overlapping nodes within the circle/sphere of any triangle/tetrahedron of the input mesh. Delaunay's insertion methods introduce new points into a Delaunay Triangulation (DT) while preserving the Delaunay criterion [16]. Several works investigated this criterion to establish a connection between a group of locations in space and produce a three-dimensional mesh. Liu et al. [17] introduced a highly efficient technique for creating Delaunay meshes by inserting fewer extra points compared to earlier methods [18]. Another method, introduced by Chew [19] and Ruppert [20], involves creating new nodes at the centers of circle/sphere of elements in a predetermined sequence. This approach ensures high quality by generating triangles with a minimum angle bound in the mesh. Rebay [21] has presented the Voronoi-segment point insertion method as an alternative to the circle point insertion strategy. A Voronoi-segment is a line segment that connects the circle centers of two neighboring triangles, or tetrahedral. The added node is placed at a specific location on the Voronoi segment that meets the optimal local size requirements. This technique produces structured meshes consisting of six triangles at each internal node. The limitation of this method lies in the susceptibility of the boundary detectors to external factors and the challenge of devising a strategy to analyze the variations in the generated contour patterns. In their study, [22] the authors introduced the off-center approach as a substitute for centers to enhance the quality of a mesh by adding new 3D points, denoted Steiner points. The benefit of this refinement technique is that it is able to incorporate a few Steiner points when compared to alternative methods. However, this technique is restricted to the two-dimensional domain. Sparse Voronoi refinement (SVR) [23] is a method that has been created

to ensure the preservation of the Delaunay criterion and an ideal radius-edge ratio during the remeshing process. The SVR algorithm maintains a constant degree for the majority of vertices in order to provide a consistent processing time. The SVR approach is limited by its long execution time. Another technique of DT refinement, denoted as parallel SVR, has resolved this limitation. It enhances efficiency by employing parallel processing instead of sequential processing [24]. However, this technique is unable to enhance the quality of the mesh, such as by optimizing the minimal/maximal angle and valence. In the context of point insertion, Walton [25] introduced a novel approach for the insertion of additional points into an input mesh. This involves evaluating the mesh's Bezier surface on every triangle of the input mesh. Owen et al. [26] subsequently enhanced this algorithm. Their method assigns a normal vector to each vertex of the triangulation. By using the vertex coordinates and computed normal vectors of each triangle, this technique derives new control points and obtains the inserted 3D points. Also, the advancing front method [27] is an iterative approach for the insertion of Steiner points. This approach calculates the optimal position of the new point on every triangle of the mesh. Consequently, the computed point is inserted, and a local reconnection is applied. The efficiency of the advancing front method has been demonstrated through the use of numerous realistic cases. Nevertheless, a drawback of this approach is the substantial computational expense incurred when performing additional operations to address mesh distortion. To address this limitation, a novel approach has been developed that integrates the Octree and advancing front methods [28] to create a 3D finite element mesh. The combined approach generates meshes of superior quality while requiring fewer processing resources compared to using a single method. An Octree is employed to create a background grid, which is subsequently used to triangulate the model surface. Nevertheless, the problem of the inadequate convergence of these approaches persists. Furthermore, the mentioned approaches do not provide a satisfactory reconstruction quality of the CAD model from the improved meshes. This paper introduces a novel method for initially inserting 3D points into the input mesh and approximating B-Spline surfaces to reconstruct a CAD model with high accuracy.

3. Proposed methodology

To address the challenges of B-Spline surface reconstruction, this paper introduces a novel methodology for reconstructing B-Spline surfaces by inserting new points based on Walton's algorithm. A general method is employed to reconstruct the final CAD model using the extracted deformed mesh information. An overview of the suggested method is presented in the flowchart (Figure 1). To reconstruct the CAD model, the proposed approach involves the extraction of the corresponding three-dimensional points to each face of the given CAD model mesh. Then, new points are inserted using Walton's technique [25]. After that, a CAD face reconstruction step is executed. This process is applied to all CAD model faces:

(1) CAD edge Reconstruction The process of reconstructing CAD edges includes the precise matching of data points to establish the boundaries of the model.
(2) Approximate the B-Spline surface's control points Levenberg Marquardt Algorithm (LMA) is investigated to approximate the B-Spline surface's control points [6].
(3) Reconstructing the bearing surface of the CAD face. The CAD face is redesigned to enhance the bearing surface.
(4) Edge Projection onto Surface An edge projection step onto the surface is needed in order to incorporate loops and create the face.

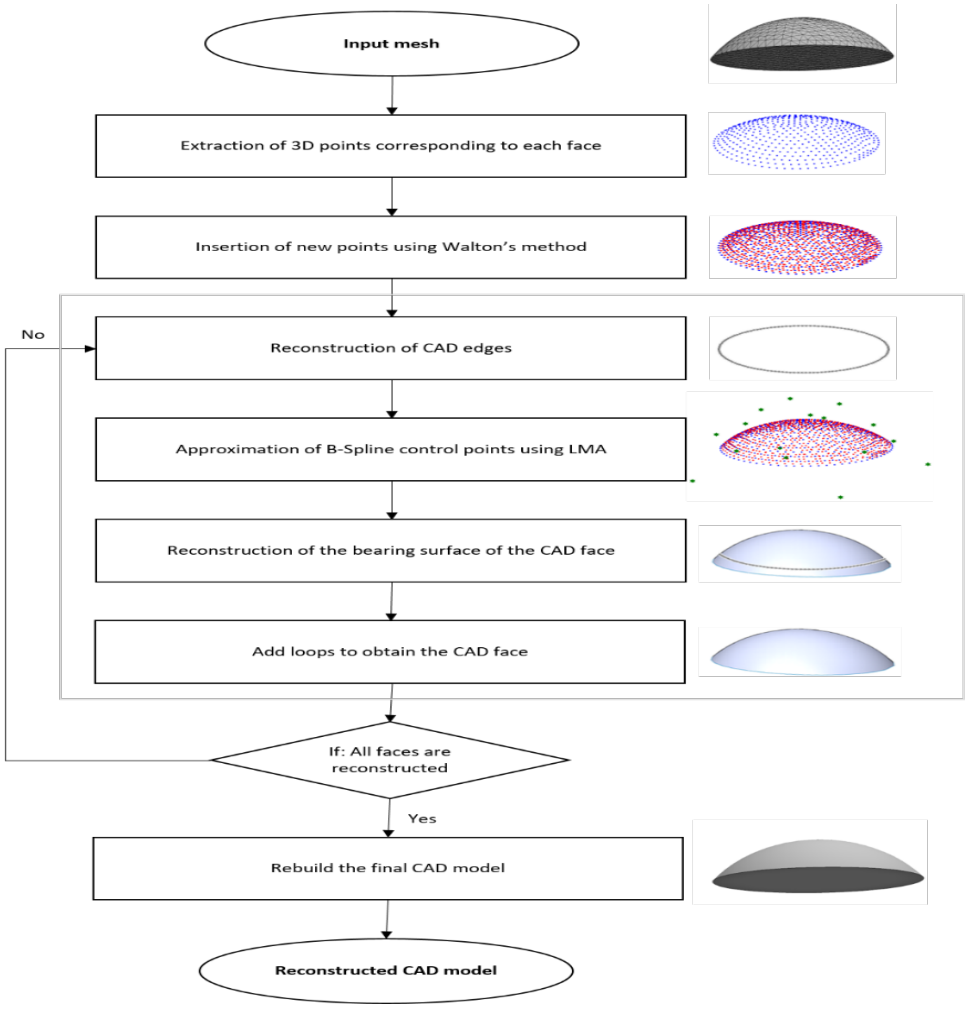

Fig. 1. Overview of the proposed methodology

3.1. *3D point insertion method*

Accurate 3D model reconstruction is a critical task that depends on the modeling capabilities provided by CAD software and requires a high level of interaction between the software and the designer. Tsai et al. [29] presented multiple techniques in order to reconstruct B-Spline surfaces given an input set of points while ensuring C1 continuity. Galvez and Iglesias [30] employed distinct techniques to produce B-Spline surfaces based on 3D input data. Renata et al. [31] present a new approach to reconstructing a triangulation that meets the Delaunay criterion from a 3D point cloud using self-organizing maps. Renata et al. employed the "Particle Swarm Optimization (PSO)" technique to regenerate NURBS surfaces of a specific order based on the input point. These studies demonstrate that the process of reconstructing a three-dimensional surface cannot yield accurate results because of the insufficiency of the methods used for the input data and the insufficient number of data points. The suggested method tries to enhance the input mesh by introducing new points, with the goal of achieving optimal results and reducing the surface reconstruction error. This task is accomplished by augmenting the quantity of points in order to achieve a more accurate representation of the 3D mesh. The quality of reconstruction is directly proportional to the number of points used, resulting in a reduction in 3D reconstruction error. Walton's approach [25] is investigated to insert new 3D points into the input deformed mesh. This approach involves assessing the meshed surface by introducing a new point on each triangle of the given mesh (Figure 2).

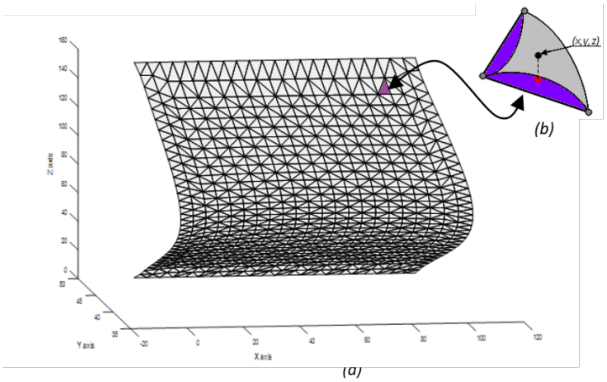

Fig. 2. Surface reconstruction using point insertion (a) Deformed mesh surface (b) Inserted within the chosen triangle.

3.1.1. *Walton's method*

In this paper, the point insertion process of a new point into the triangles of a given deformed mesh is implemented using Walton technique [25]. The approach involves

computing the normal vector at each vertex of the triangulation (Figure 3(a)). The normal vector (Figure 3(b)) is defined by the equation (Equation 1).

In this paper, the point insertion process of a new point into the triangles of a given deformed mesh is implemented using Walton technique [25]. The approach involves computing the normal vector at each vertex of the triangulation (Figure 3(a)). The normal vector (Figure 3(b)) is defined by the equation (Equation 1).

$$N_P = \sum_{i=1}^{n} N_i * w_i a1 \tag{1}$$

Where:
P: is the selected node.
n: is the number of triangles having a same node P.
N_i: represents each normal vector corresponding to the adjacent triangles having the same node P.
w_i : The weighting coefficient for each normal vector Ni is determined by Equation (**??**).

$$w_i = \frac{\alpha_i}{\sum_{i=1}^{n} \alpha_i} a2 \tag{2}$$

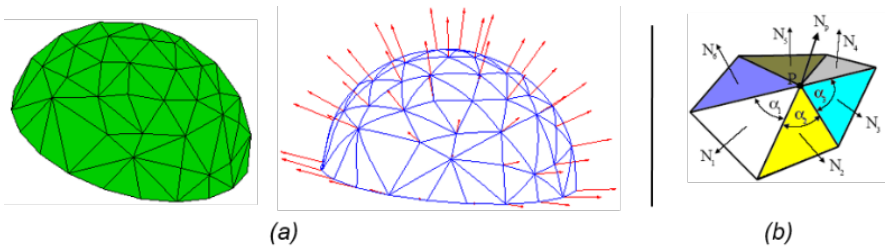

Fig. 3. Normal vector calculation

(a) Normal vectors at all the points of the meshed surface, (b) Normal face.

Given a triangle defined by three points $P0$, $P1$, $P2$ and three vectors $N0$, $N1$, $N2$, the proposed approach quantifies the portion of the reconstructed triangle surface. The calculation of the inserted point on a selected surface is determined as follows (Equation 3).

$$S(u,v,w) = \sum_{i+j+k=4} P_{i,j,k} \frac{4!}{i!j!k!} u^i v^j w^k a3 \tag{3}$$

$$u, v, w \geq 0; u + v + w = 1; i, j, k \geq 0$$

Such as u, v and w: are the 3D point coordinates of the triangle within the given deformed mesh (Figure 4).

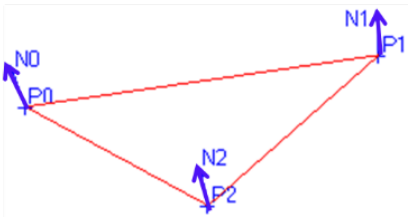

Fig. 4. Determination of a normal vector for the triangle points

$P_{i,j,k}$: Are the control points information of the surface relying on $P0$, $P1$, $P2$, $N0$, $N1$ and $N2$. (Figure 5)

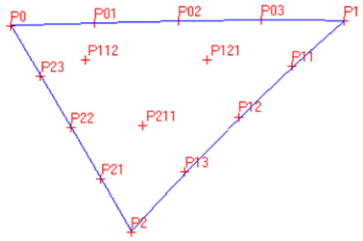

Fig. 5. Control points of the selected triangle

3.1.2. *TriangleEdge Control points*

To introduce a new point on the triangle, it is necessary to calculate the three control points $\mathrm{P}_{(i,j)}$ for each edge of the triangle (Figure 6).

The suggested TriangleEdge is defined by two endpoints, P_i and $\mathrm{P}_{(i+1)}$, as well as two surface normal vectors and two curve tangent vectors, N_i and $\mathrm{N}_{(i+1)}$. The tangents (T_i) are determined by Equation 4:

$$T_i = T_{i+1} = \frac{P_{i+1} - P_i}{||P_{i+1} - P_i||} a4 \tag{4}$$

The control point coordinates of each TriangleEdge(i) are determined using the equations (Equation 5) to (Equation 14) [25].

$$d_i = ||P_{i+1} - P_i|| \tag{5}$$

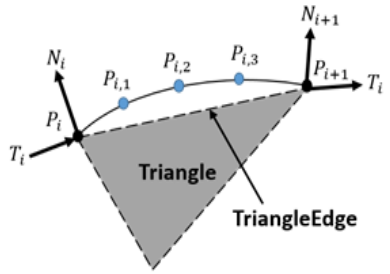

Fig. 6. Control points on TriangleEdge

$$a_i = N_i.N_{i+1} \tag{6}$$

$$a_{i,0} = N_i.T_i \tag{7}$$

$$a_{i,1} = N_{i+1}.T_{i+1} \tag{8}$$

$$\rho_i = \frac{6\left(2a_{i,0} + a_i.a_{i,1}\right)}{4 - a_i^2} \tag{9}$$

$$V_{i,1} = P_i + \frac{d_i\left(6T_i - 2\rho N_i + \sigma N_{i+1}\right)}{18} \tag{10}$$

$$V_{i,2} = P_{i+1} - \frac{d_i\left(6T_{i+1} + \rho N_i + 2\sigma N_{i+1}\right)}{18} \tag{11}$$

$$P_{i,1} = \frac{1}{4}P_i + \frac{3}{4}V_{i,1} \tag{12}$$

$$P_{i,2} = \frac{1}{2}V_{i,1} + \frac{1}{2}V_{i,2} \tag{13}$$

$$P_{i,3} = \frac{3}{4}V_{i,2} + \frac{1}{4}P_{i+1} \tag{14}$$

3.1.3. *Inner Triangle Control points*

The inner control points P_{112}, P_{121} and P_{211} are defined using the control points determined on each edge of the triangle (Figure 5). The control points G_{ij}, G_{ij} where $i = 0, 1, 2$; $j = 0, 1$, are assessed as a function of the parameters $(u,\ v,\ w)$ according to Equation 27 and Equation 28. For each TriangleEdge(i), the following parameters are determined:

$$W_{i,0} = V_{i,1} - P_i \tag{15}$$

$$W_{i,1} = V_{i,2} - V_{i,1} \tag{16}$$

$$W_{i,2} = P_{i+1} - V_{i,2} \tag{17}$$

$$D_{i,0} = P_{i,3} - \frac{1}{2}(Pi, 1 + P_i) \tag{18}$$

$$D_{i,1} = P_{i,1} - \frac{1}{2}(Pi, 1 + P_{i,3}) \tag{19}$$

$$A_{i,0} = \frac{N_i \times W_{i,0}}{||W_{i,0}||} \tag{20}$$

$$A_{i,2} = \frac{N_{i+1} \times W_{i,2}}{||W_{i,2}||} \tag{21}$$

$$A_{i,1} = \frac{A_{i,0} + A_{i,2}}{||A_{i,0} + A_{i,2}||} \tag{22}$$

$$\lambda_{i,0} = \frac{D_{i,0}.W_{i,0}}{W_{i,0}.W_{i,0}} \tag{23}$$

$$\lambda_{i,1} = \frac{D_{i,1}.W_{i,2}}{W_{i,2}.W_{i,2}} \tag{24}$$

$$\mu_{i,0} = D_{i,0}.A_{i,0} \tag{25}$$

$$\mu_{i,1} = D_{i,1}.A_{i,2} \tag{26}$$

The control points $G_{i,0}$, $G_{i,1}$ are defined in relation to TriangleEdge(i).

$$G_{i,0} = \frac{1}{2}(P_{i,1} + P_{i,2}) + \frac{2}{3}\lambda_{i,0}W_{i,1} + \frac{1}{3}\lambda_{i,1}W_{i,0} + \frac{2}{3}\mu_{i,0}A_{i,1} + \frac{1}{3}\mu_{i,1}A_{i,0} \tag{27}$$

$$G_{i,1} = \frac{1}{2}(P_{i,2} + P_{i,3}) + \frac{1}{3}\lambda_{i,0}W_{i,1} + \frac{2}{3}\lambda_{i,1}W_{i,0} + \frac{1}{3}\mu_{i,0}A_{i,1} + \frac{2}{3}\mu_{i,1}A_{i,0} \tag{28}$$

The inner control points of the triangle $P_{1,1,2}$, $P_{1,2,1}$ and $P_{2,1,1}$ are calculated as following:

$$P_{1,1,2} = \frac{1}{u+v}(uG_{2,2} + vG_{0,1}) \tag{29}$$

$$P_{1,2,1} = \frac{1}{w+u}(wG_{0,2} + uG_{1,1}) \tag{30}$$

$$P_{2,1,1} = \frac{1}{v+w}(vG_{1,2} + wG_{2,1}) \tag{31}$$

Finally, the coordinates of the inserted point at parameter (u, v, w) are calculated using equation 3 and the control points specified in equations (Equation 12) to (Equation 14) and (Equation 28) to (Equation 31).

3.2. *CAD Edge Reconstruction*

In this phase, the CAD edges are calculated by reconstructing B-Spline curves. For a given set of n points X_k and a degree p of the B-Spline curve, each point on the curve is interpolated (Equation 32).

$$X_k = C\left(\underline{u}_k\right) = \sum_{i=0}^{n} N_{ip}(\underline{u}_k)P_i \tag{32}$$

Where:

P_i are the curve control points. N_i : represent the B-Spline basis functions. $\underline{u}_k$: are the parameters of each point on the curve ($0 \leq \underline{u}_k \leq 1$). To calculate B-Spline curve parameters $\underline{u}_k$, the centripetal method is used [29] [30]. This method is described by the equations (Equation 33) and (Equation 34).

$$\underline{u}_0 = 0,\ \underline{u}_n = 0,\ \ \underline{u}_k = \underline{u}_{k-1} + \frac{\sqrt{|Q_k - Q_{k-1}|}}{d},\ \ \{k = 1 \ldots n-1\} \tag{33}$$

$$d = \sum_{k=1}^{n} \sqrt{|Q_k - Q_{k-1}|} \tag{34}$$

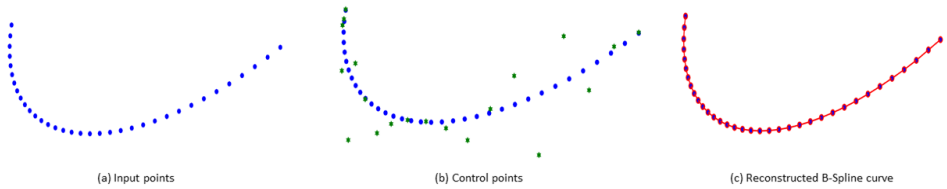

Fig. 7. Curve interpolation using the centripetal method

Following that, the averaging approach [30] [31] is used to calculate the knot vector based on the obtained parameters.

$$(u_0 = \cdots = u_p = 0,\ u_{m-p} = \cdots = u_m = 1),\quad u_{j+p} = \frac{1}{p}\sum_{i=j}^{j+p-1} \underline{u}_i, (j = 1 \ldots n-p) \tag{35}$$

The resulting knot vector has been investigated to calculate the B-Spline basis functions; the control point coordinates of the B-Spline curve are obtained using (Equation 32).

3.3. *CAD faces reconstruction*

In this stage, the optimization process of Levenberg Marquardt Algorithm is used to rebuild the bearing surface of every CAD face. To approximate each B-Spline surface, the obtained set of points, which includes the deformed mesh points and the newly inserted points, is investigated [32]. The control points' coordinates are determined by optimizing the objective function that represents the non-linear least squares problem [33]. Then, B-Spline surface [34] is determined by the degrees p and q in the u-direction and v-direction, respectively, as well as the knot vectors and the set of control points. Equation 36 is used to find the coordinates of every point on the B-Spline surface.

$$S(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) P_{i,j} \tag{36}$$

Where: u, v: are the parameters of the 3D point, referring to its coordinates in the parametric space. $N_{i,p}$, $N_{j,q}$: are the B-Spline basis functions.

$P_{i,j}$: represent the control points of the B-Spline surface.

To approximate the above control points, the following algorithm is developed.

Algorithm 1. Pseudocode of the surface rebuilding algorithm

Input: M: The set of updated 3D points; N: number of updated points **Output**: The reconstructed B-Spline surface

1. Begin
1. Calculate the infinite average plane $PL_{average}$ of the given input points.2. **For** i = 1: N
Project the 3D point M(i) on $PL_{average}$
End For
3. Define the 2D bounding box corresponding to the projected points.
4. **For** i = 1: N
Calculate the parameter (u,v) of each projected point.
End For
5. Create a P_0 vector that contains the control points that correspond to the chosen beginning surface.
6. Construct the B-Spline surface S_0 using the control points P_0.
Initialize the damping parameter $\lambda \leftarrow 0.0001$
Repeat

$$\lambda \leftarrow \lambda * 0.04$$

For i = 1: N
Find the distance $d(i)$ from the point $M(i)$ to the surface S_0.
End For
Compute F_0 the matrix that contains the gradient of $d(i)$ in each row.

$$U \leftarrow F_0^T F$$

$$V \leftarrow F_0^T . \, distance(M, S_0)$$

$J_0 \leftarrow \sum distance(M, S_0)^{\mathbf{2}}$
Repeat

$$\lambda \leftarrow \lambda * 5$$

$$H \leftarrow U + \lambda(I + Diagonal\ (U_{11}, U_{22}, \ldots, U_{nn})$$

Solve the system $H.X = -V$

$$P_{new} \leftarrow P_0 + X$$

Calculate the B-Spline surface S_{new} that is determined by the new control points P_{new}

$$J_{new} \leftarrow \sum distance(M, S_{new})^{\mathbf{2}}$$

If J converge
Normalize P_{new}

$$P_0 \leftarrow P_{new}$$

$$S_0 \leftarrow S_{new}$$

Return (P_0, S_{new})

End If
Until ($J_{new} < J_0$) **Or** (an iteration limit is reached)
If ($J_{new} < J_0$) ***then***

$$P_0 \leftarrow P_{new}$$

$$S_0 \leftarrow S_{new}$$

End If
Until (an iteration limit is reached)
2. Return the reconstructed B-Spline surface S_0
3. End

By limiting each reconstructed B-Spline surface to the corresponding reconstructed CAD edges described in Section 3.3, the CAD faces are determined, and the final CAD model is rebuilt applying the general algorithm described in the flowchart (Figure 1).

4. EXPERIMENTS

4.1. *Surface quality evaluation*

To demonstrate the efficacy of the approach, the following metrics are employed to assess the precision of the surface reconstruction process: the Maximum Root Mean Square Error (MRMS), the Hausdorff distance (Hd), and the average distance.

4.1.1. *MRMS metric*

The Maximum Root Mean Square Error metric [35] quantifies the spatial difference between the points on the original surface S and the reconstructed surface S'(Equation 37).

$$MRMS = max(d_{RMS}(S, S'),\ d_{RMS}(S', S)) \tag{37}$$

d_{RMS} represent the average difference between two surfaces S and S' (Equation 38).

$$d_{RMS}\left(S, S'\right) = \sqrt{\frac{1}{|S|} \iint_{p \in S} d\left(p, S'\right)^2 dS} \tag{38}$$

Where $d\left(p, S'\right)$ represents the minimum Euclidean distance between two points p and p' along the surfaces S and S', respectively (Equation 39).

$$d\left(p, S'\right) = ||p - p'||_2 \tag{39}$$

where:

p: is a 3D point in S.
p': is a 3D point in S'.
$|S|$: is the area of the surface S.
$||p - p'||_2$: The Euclidean distance between the two points p and p'.

4.1.2. *Hausdorff distance (Hd)*

The Hausdorff distance (Hd) metric [36] measures the geometric error between two surfaces S and S' which represent the original and rebuilt surfaces, respectively. Hausdorff distance is defined by Equation 40.

$$Hd\left(S, S'\right) = \left(d\left(S, S'\right), d\left(S', S\right)\right) \tag{40}$$

The Hausdorff distance $d\left(S, S'\right)$ is calculated using Equation 41.

$$d\left(S, S'\right) = d\left(p\ ,\ S'\right) \tag{41}$$

Where: $d\left(p, S'\right)$ represents the minimum Euclidean distance between the points p and p' along the surfaces, S' respectively.

$$d\left(p, S'\right) = ||p\ -\ p'||_2 \tag{42}$$

4.1.3. *Average distance*

The average distance [6] is a geometric measure that quantifies the surface reconstruction error. It is calculated by finding the mean of the Euclidean distances between two different sets of points: the original surface S and the rebuilt surface S'. Euclidian distance between each point p in the surface S and the point p' in the surface S' is computed using (Equation 43).

$$d\left(p, p'\right) = ||p - p'||_2 \tag{43}$$

4.2. *Reconstruction of the B-Spline surfaces*

To validate the proposed method, the reconstruction error is measured by comparing the fitted B-Spline surface given the initial set of extracted points with the surface reconstructed from the refreshed points using Walton's algorithm.

An example of a spherical surface reconstruction is illustrated in (Figure 8). The original surface contains a total of 537 points. Following the phase of inserting points, the 3D point number has increased to 1537.

The difference between the surface reconstruction based on the starting point and the obtained point is highlighted in (Table 1).

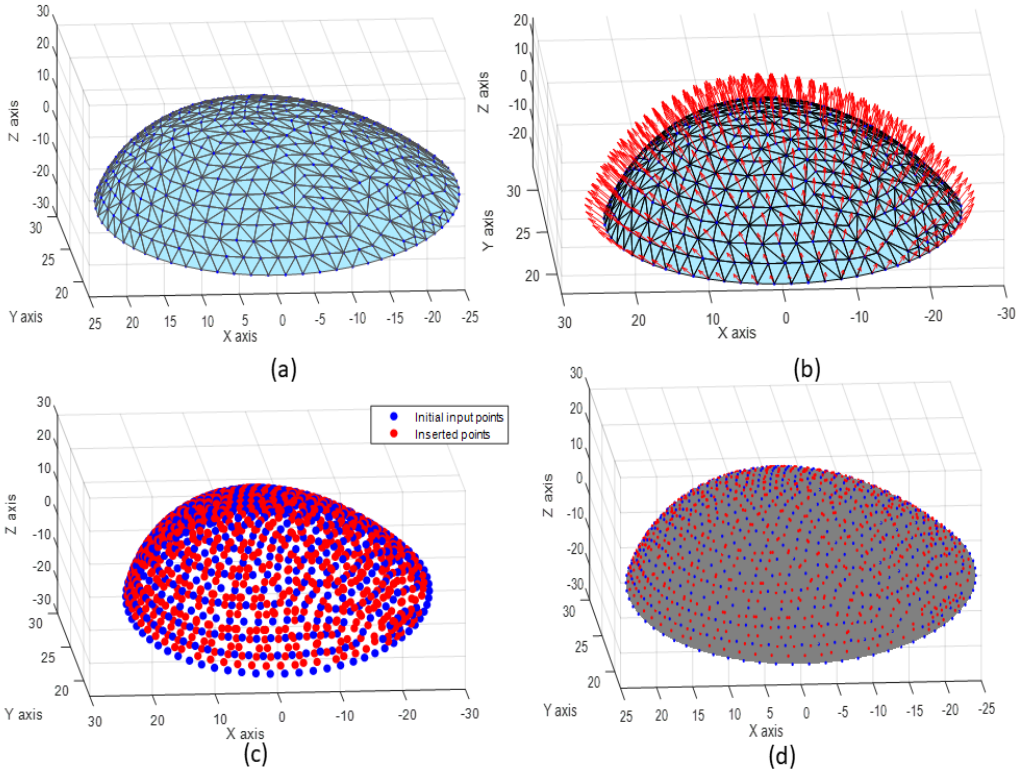

Fig. 8. Reconstruction of the spherical surface (a) Input mesh (b) Normal vectors (c) (Blue) Initial points and (Red) inserted point (d) B-Spline surface after reconstruction.

Table 1. Spherical surface reconstruction results before and after point insertion method.

	Points Number	Average distance	Hausdorff distance	MRMS
The rebuilt surface using **LMA [6] before points insertion**	537	$3{,}538*10^{-6}$	0,0206	0,0035
The rebuilt surface after points insertion using **Loop Subdivision method** [14]	2073	$3{,}779*10^{-5}$	0,0542	0,0308
The rebuilt surface with point insertion method using **the proposed approach**	1537	$4{,}373*10^{-7}$	0,0181	$5{,}953*10^{-4}$

Surface rebuilding error computed through the average distances (Table 1) indicates that the surface reconstruction error from the input data is greater than the error after incorporating new points using the proposed methodology. In addition, the results demonstrate that Walton's method is more efficient than the Loop subdivision method when it is tested to update the number of points. Figure 9 illustrates an additional instance of B-Spline surface reconstruction.

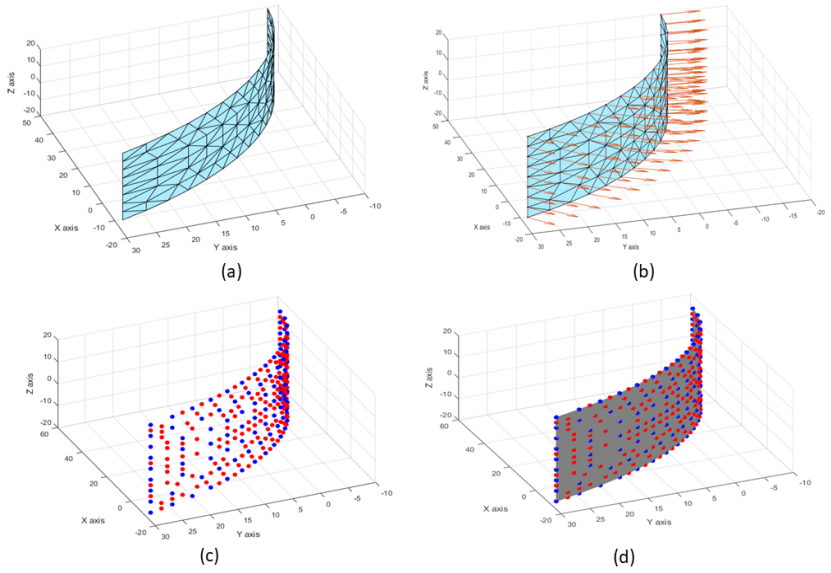

Fig. 9. B-Spline surface rebuilt (a) Input mesh (b) Calculated normal (c) Initial 3D points (blue) and added points (Red) (d) Rebuilt surface.

Table 2 presents the measured errors for the surface reconstruction achieved from the initial input points as well as from the obtained set of points using both the Loop Subdivision method and the proposed approach.

Table 2. Surface reconstruction results before and after the points' insertion method

	Points Number	Average distance	Hausdorff distance	MRMS
The rebuilt surface using **LMA [6] before points insertion**	103	$1{,}815*10^{-5}$	0,2638	0,0468
The rebuilt surface after points insertion using **Loop Subdivision method** [14]	365	$5{,}355*10^{-5}$	0,1179	0,0488
The rebuilt surface with point insertion method using **the proposed approach**	263	$5{,}242*10^{-6}$	0,0380	0,0056

The acquired results highlighted in (Table 1) and (Table 2) prove that the number of points influences the surface reconstruction process. In the case of the Loop subdivision method [14], the point insertion technique increases the error rate of the surface rebuilding process. This method is not efficient for the B-Spline surface rebuilding. However, the obtained results by the developed approach are convenient:

the reconstruction error after the points' insertion step using Walton's technique is lower than the reconstruction error given the input points.

4.3. *CAD model reconstruction: case studies*

To validate the proposed approach, four CAD models are designed in the SolidWorks environment. Then, different deformations are produced by external forces / pressures to generate more complex shapes. The figure (Figure 10) presents the first CAD model reconstruction with a simple topology (rectangular model). To rebuild the deformed CAD model, the algorithm described in Figure 1 is applied. During the CAD face reconstruction, a B-Spline surface is approximated, the corresponding edges are determined, and the final loops are added.

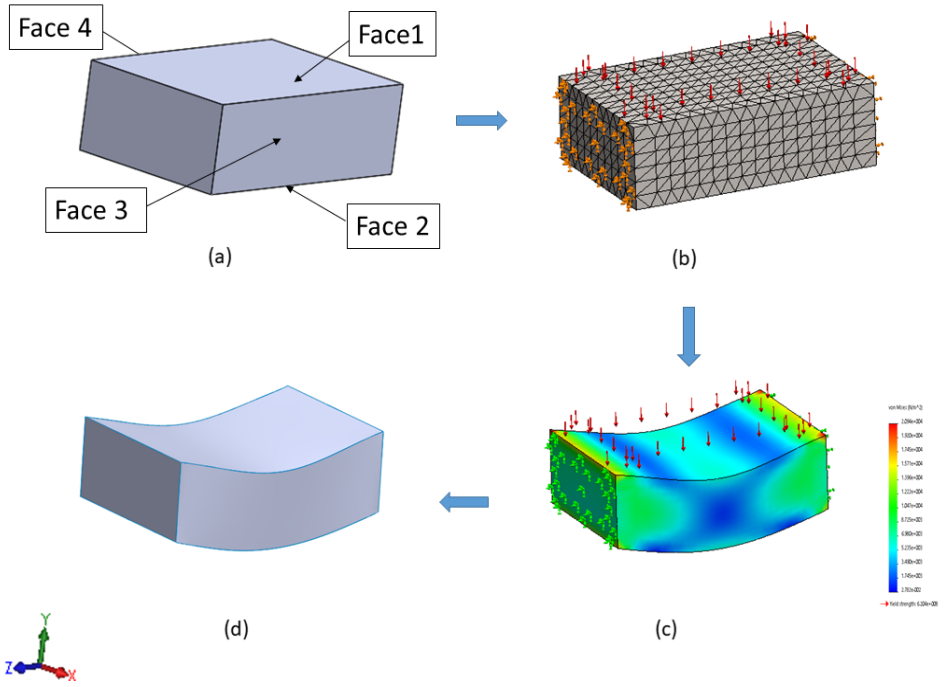

Fig. 10. Reconstruction of the rectangular model. (a) Initial model (b) extracted mesh (c) Finite Element results of the deformed model (d) Reconstructed deformed model.

The following tables (Table 3), (Table 4), (Table 5) and (Table 6) present the results of four deformed faces corresponding to the deformed rectangular model (Figure 10).

The second example (Figure 11), presents a second model containing five faces. The selected face in (Figure 11) is chosen to evaluate the developed algorithm.

Table 3. Surface reconstruction results before and after points insertion (Face 1)

	Points Number	Average distance	Hausdorff distance	MRMS
The rebuilt surface using **LMA [6] before points insertion**	765	$1{,}414*10^{-5}$	0,1634	0,0267
The rebuilt surface after points insertion using **Loop Subdivision method** [14]	2945	$1{,}309*10^{-5}$	0,6992	0,0743
The rebuilt surface with point insertion method using **the proposed approach**	2181	$2{,}966*10^{-6}$	0,0289	0,0048

Table 4. Surface reconstruction results before and after points insertion (Face 2)

	Points Number	Average distance	Hausdorff distance	MRMS
The rebuilt surface using **LMA [6] before points insertion**	765	$1{,}310*10^{-5}$	0,3025	0,0250
The rebuilt surface after points insertion using **Loop Subdivision method** [14]	2945	$3{,}650*10^{-6}$	0,3356	0,0121
The rebuilt surface with point insertion method using **the proposed approach**	2181	$2{,}957*10^{-6}$	0,0975	0,0078

Table 5. Surface reconstruction results before and after points insertion (Face 3)

	Points Number	Average distance	Hausdorff distance	MRMS
The rebuilt surface using **LMA [6] before points insertion**	435	$3{,}374*10^{-6}$	0,0290	0,0048
The rebuilt surface after points insertion using **Loop Subdivision method** [14]	1645	$9{,}280*10^{-6}$	0,2805	0,0138
The rebuilt surface with point insertion method using **the proposed approach**	1211	$2{,}661*10^{-6}$	0,0181	0,0035

The figure (Figure 12) presents the third model rebuild. The reconstruction results of the selected face are highlighted in Table 8.

The fourth example (Figure 13) presents the reconstruction of a more complex model: the crankshaft model.

To evaluate the reconstruction results of the selected face (Figure 13) the reconstruction error of the B-Spline surface from the initial points is calculated. After the insertion of new points using the Loop Subdivision method and the proposed approach, the reconstruction results are measured (Table 9).

Table 6. Reconstruction results of the fourth face before and after points insertion (Face 4)

	Points Number	Average distance	Hausdorff distance	MRMS
The rebuilt surface using **LMA [6] before points insertion**	435	$5{,}353*10^{-6}$	0,0407	0,0070
The rebuilt surface after points insertion using **Loop Subdivision method** [14]	1645	$1{,}041*10^{-5}$	0,0535	0,0128
The rebuilt surface with point insertion method using **the proposed approach**	1211	$3{,}006*10^{-6}$	0,0217	0,0042

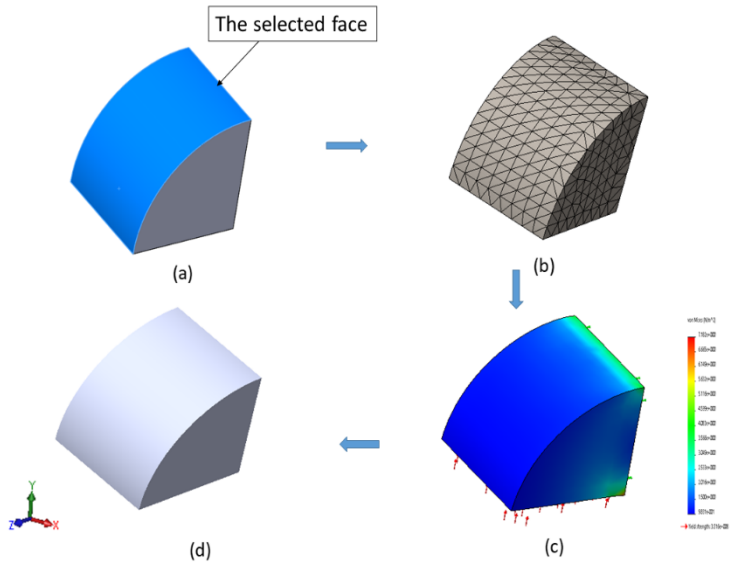

Fig. 11. CAD model reconstruction. selected face of the initial model; (b) extracted mesh; (c) Finite element results of the deformed model; (d) reconstructed deformed model.

Table 7. Reconstruction results of the selected face before and after points insertion

	Points Number	Average distance	Hausdorff distance	MRMS
The rebuilt surface using **LMA [6] before points insertion**	123	$6{,}238*10^{-6}$	0,0121	0,0050
The rebuilt surface after points insertion using **Loop Subdivision method** [14]	445	$1{,}969*10^{-5}$	0,0473	0,0158
The rebuilt surface with point insertion method using **the proposed approach**	323	$1{,}352*10^{-6}$	0,0082	0,0014

The results demonstrate that the suggested method reduces reconstruction error compared to two previous methods: the first uses the initial extracted points,

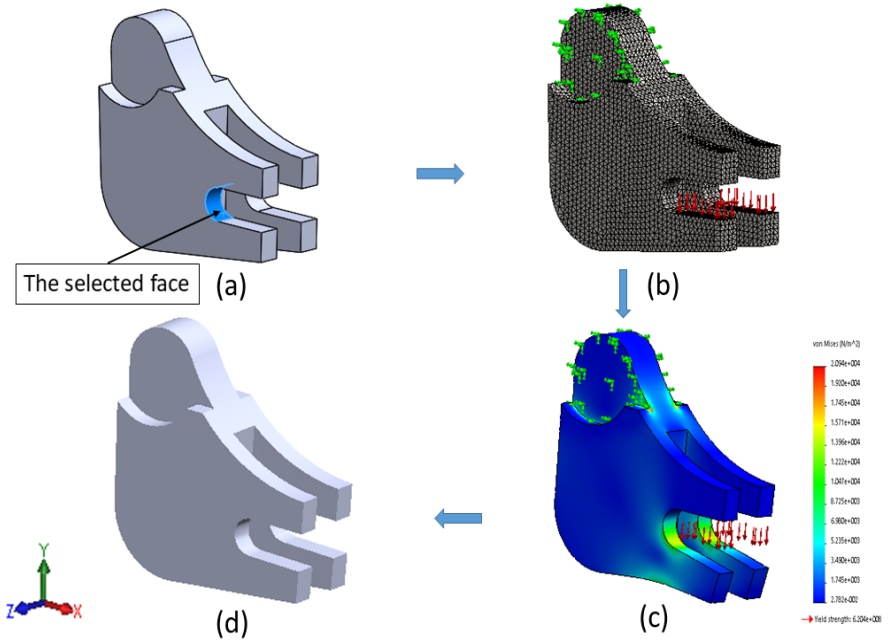

Fig. 12. CAD model reconstruction. (a) selected face of the initial model (b) extracted mesh (c) Finite Element results of the deformed model (d) reconstructed deformed model.

Table 8. Reconstruction results of the selected face before and after points insertion

	Points Number	Average distance	Hausdorff distance	MRMS
The rebuilt surface using **LMA [6] before points insertion**	199	$2{,}063*10^{-5}$	0,2156	0,0210
The rebuilt surface after points insertion using **Loop Subdivision method** [14]	733	$2{,}298*10^{-5}$	0,2960	0,0209
The rebuilt surface with point insertion method using **the proposed approach**	535	$8{,}696*10^{-6}$	0,1853	0,0122

and the second inserts new points using the Loop Subdivision method [14]. This is executed for each selected face after reconstruction. The suggested method enhances surface reconstruction quality by inserting additional points using Walton's technique and approximating the B-Spline surface using LMA.

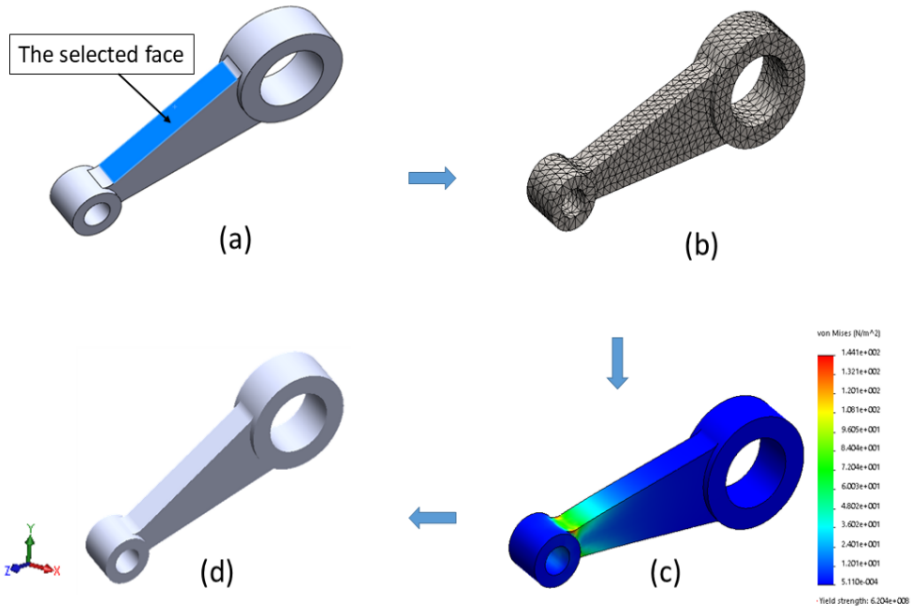

Fig. 13. Reconstruction of the crankshaft model. (a) Selected face of the initial model (b) extracted mesh (c) Finite Element results of the deformed model (d) reconstructed deformed model.

Table 9. Selected face reconstruction results before and after points insertion

	Points Number	Average distance	Hausdorff distance	MRMS
The rebuilt surface using **LMA [6] before points insertion**	339	$2{,}341*10^{-7}$	$9{,}228*10^{-4}$	$2{,}777*10^{-4}$
The rebuilt surface after points insertion using **Loop Subdivision method** [14]	1261	$5{,}440*10^{-7}$	0,0043	$8{,}910*10^{-4}$
The rebuilt surface with point insertion method using **the proposed approach**	847	$2{,}011*10^{-7}$	$6{,}542*10^{-4}$	$2{,}415*10^{-4}$

5. Conclusion

In this paper, an original method based on the combination of Walton's algorithm to insert new points into an input mesh and the Levenberg Marquardt Algorithm to approximate B-Spline surfaces is proposed. Several steps are followed in order to obtain the rebuilt CAD model. The first step is to get the starting points from the mesh surfaces. The next step is to use Walton's algorithm to insert new points. Then, in order to restore the CAD faces, the surfaces are rebuilt using LMA and

the loops are added. The quality of the rebuilt surfaces is improved by the proposed method, which is based on inserting additional points. The proposed approach's robustness is validated through the rebuilding of several models. Accuracy in surface reconstruction is evaluated using three geometric metrics: MRMS, Hd, and average distance. Surface reconstruction using the suggested method consistently outperforms surface reconstruction starting from the original points or using a different subdivision method (the Loop subdivision method) to insert new points. The implemented approach's performance is proven by comparing these results with other existing methods.

References

1. Bénière R., Subsol G., Gesquière G., Le Breton F. and Puech W., A comprehensive process of reverse engineering from 3d meshes to cad models, *Computer-Aided Design* **45**(11):1382–1393, 2013.
2. Khan D., Yan D.-M., Gui S., Lu B. and Zhang X., Molecular surface remeshing with local region refinement, *International journal of molecular sciences* **19**(5):1383, 2018.
3. Ma Z. and Liu S., A review of 3d reconstruction techniques in civil engineering and their applications, *Advanced Engineering Informatics* **37**:163–174, 2018.
4. Fathi H., Dai F. and Lourakis M., Automated as-built 3d reconstruction of civil infrastructure using computer vision: Achievements, opportunities, and challenges, *Advanced Engineering Informatics* **29**(2):149–161, 2015.
5. Louhichi B., Abenhaim G. N. and Tahan A. S., Cad/cae integration: updating the cad model after a fem analysis, *The International Journal of Advanced Manufacturing Technology* **76**:391–400, 2015.
6. Ben Makhlouf A., Louhichi B., Mahjoub M. A. and Deneux D., Reconstruction of a cad model from the deformed mesh using b-spline surfaces, *International Journal of Computer Integrated Manufacturing* **32**(7):669–681, 2019.
7. Makhlouf A., Louhichi B., Deneux D. and Mahjoub M. A., Reconstruction of the cad model using tps surface, *2019 23rd International Conference Information Visualisation (IV)*, IEEE, pp. 417–424, 2019.
8. Park J. M., Lee B. C., Chae S. W. and Kwon K. Y., Surface reconstruction from fe mesh model, *Journal of Computational Design and Engineering* **6**(2):197–208, 2019.
9. Ren B. and Hagiwara I., Composite freeform surface reconstruction using recursive interpolating subdivision scheme, *Computers in Industry* **50**(3):265–275, 2003.
10. Yang X., Surface interpolation of meshes by geometric subdivision, *Computer-Aided Design* **37**(5):497–508, 2005.
11. Bølviken H. S., Bersvendsen J., Orderud F., Snare S. R., Brekke P. and Samset E., Two methods for modifed doo–sabin modeling of nonsmooth surfaces—applied to right ventricle modeling, *Journal of Medical Imaging* **7**(6):067001–067001, 2020.
12. Bandara K. and Cirak F., Isogeometric shape optimisation of shell structures using multiresolution subdivision surfaces, *Computer-Aided Design* **95**:62–71, 2018.
13. Alam M. N. and Li X., Non-uniform doo-sabin subdivision surface via eigen polygon, *Journal of Systems Science and Complexity* **34**:3–20, 2021.
14. Li G., Ren C., Zhang J. and Ma W., Approximation of loop subdivision surfaces for fast rendering, *IEEE transactions on visualization and computer graphics* **17**(4):500–514, 2010.
15. Delaunay B., Sur la sphère vide. a la mémoire de georges vorono, *Bulletin de*

l'Académie des Sciences de l'URSS Classe des sciences mathématiques et naturelles **6**:793.
16. Cheng S.-W., Dey T. K. and Shewchuk J., *Delaunay mesh generation*, CRC Press, 2012.
17. Liu Y.-J., Xu C.-X., Fan D. and He Y., Efficient construction and simplification of delaunay meshes, *ACM Transactions on Graphics (TOG)* **34**(6):1–13, 2015.
18. Dyer R., Zhang H. and Möller T., Delaunay mesh construction , 2007.
19. Chew L. P., *Guaranteed-quality triangular meshes*, *Tech Rep*, Cornell University, 1989.
20. Ruppert J., A new and simple algorithm for quality 2-dimensional mesh generation, *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, pp. 83–92, 1993.
21. Rebay S., Efficient unstructured mesh generation by means of delaunay triangulation and bowyer-watson algorithm, *Journal of computational physics* **106**(1):125–138, 1993.
22. Üngör A., Off-centers: A new type of steiner points for computing size-optimal quality-guaranteed delaunay triangulations, *Latin American Symposium on Theoretical Informatics*, Springer, pp. 152–161, 2004.
23. Hudson B., Miller G. and Phillips T., Sparse voronoi refinement, *Proceedings of the 15th International Meshing Roundtable*, Springer, pp. 339–356, 2006.
24. Hudson B., Miller G. L. and Phillips T., Sparse parallel delaunay mesh refinement, *Proceedings of the nineteenth annual ACM symposium on Parallel algorithms and architectures*, pp. 339–347, 2007.
25. Walton D. J. and Meek D. S., A triangular g1 patch from boundary curves, *Computer-Aided Design* **28**(2):113–123, 1996.
26. Owen S. J. and White D. R., Mesh-based geometry: A systematic approach to constructing geometry from a finite element mesh., *IMR*, Citeseer, 2001.
27. Marcum D. L. and Weatherill N. P., Unstructured grid generation using iterative point insertion and local reconnection, *AIAA journal* **33**(9):1619–1625, 1995.
28. Xie Q. and Geng G., 3d mesh generation based on improved advancing front method, *2017 2nd International Conference on Modelling, Simulation and Applied Mathematics (MSAM2017)*, Atlantis Press, pp. 147–151, 2017.
29. Rajamohan G., Selection of parameterization method for fitting of freeform curves using uniformly spaced data, *Trends in Manufacturing and Engineering Management: Select Proceedings of ICMechD 2019* pp. 943–950, 2021.
30. Balta C., Öztürk S., Kuncan M. and Kandilli I., Dynamic centripetal parameterization method for b-spline curve interpolation, *IEEE Access* **8**:589–598, 2019.
31. Ammad M. and Ramli A., Cubic b-spline curve interpolation with arbitrary derivatives on its data points, *2019 23rd International Conference in Information Visualization–Part II*, IEEE, pp. 156–159, 2019.
32. Gálvez A. and Iglesias A., Particle swarm optimization for non-uniform rational b-spline surface reconstruction from clouds of 3d data points, *Information Sciences* **192**:174–192, 2012.
33. Shakarji C. M., Least-squares fitting algorithms of the nist algorithm testing system, *Journal of research of the National Institute of Standards and Technology* **103**(6):633, 1998.
34. Piegl L. and Tiller W., *The NURBS book*, Springer Science & Business Media, 2012.
35. Niu Y., Zhong Y., Guo W., Shi Y. and Chen P., 2d and 3d image quality assessment: A survey of metrics and challenges, *IEEE Access* **7**:782–801, 2018.
36. Elloumi N., Kacem H. L. H., Dey N., Ashour A. S. and Bouhlel M. S., Perceptual metrics quality: Comparative study for 3d static meshes, *International Journal of*

Service Science, Management, Engineering, and Technology (IJSSMET) **8**(1):63–80, 2017.