Development of Multi-branch T-spline Templates and its Applications in Reverse Engineering

Kritika Joshi[1], Amba D. Bhatt[2]

[1]MNNIT, Allahabad, India, rme1615@mnnit.ac.in
[2]MNNIT, Allahabad, India, abhatt@mnnit.ac.in

Corresponding author: Kritika Joshi, rme1615@mnnit.ac.in

Abstract. T-Spines are being increasingly used for the design and analysis of complex-shaped geometry. It has numerous potential applications in reverse engineering. In this paper, single and multi-branch T-spline templates have been developed for the reconstruction of tree branches. The proposed method uses cross-sections, derived from point cloud data to create T-spline based surface models. We can apply it in reverse engineering for the reconstruction of any single or multi-branch object.

Keywords: Branching, Point cloud data (PCD), Surface reconstruction, T-spline.
DOI: https://doi.org/10.14733/cadaps.2020.487-501

1 INTRODUCTION

Branch surface modeling is a mathematical way of representing objects having one or several nodes and branches (natural or human-made). This term has mainly come in the picture from the ongoing research in the field of the biomedical industry. Its primary domain area includes engineering, medical, and biological science. The construction of a suitable branch surface model helps to analyze the internal branch structure like heart valves, blood vessels, arteries, lung airways, etc. It also has the potential to optimize the surgical procedure, diagnose and improve various common diseases [6, 13]. These applications require digital information (data acquisition) of the physical object. When an object is reconstructed and represented as a digital model, it is known as reverse engineering of the shape [11]. This process of reverse engineering is further evolving by the ability to capture data from the physical world and hence improving the quality of the model.

As the shape and functions of a physical object are closely related, thus drawing out the exact form by considering topological and geometrical aspects is a challenging task. In case of a branched structure, arrangement and positions of the branches decide the complexity of the shape of a tree. However, maintaining geometrical continuity at furcations where topological boundaries are changing is considered as a critical point during its reconstruction [10].

1.1 Previous Related Work

The traditional framework of reverse engineering involves the creation of triangular meshes from unorganized point cloud data followed by data capturing. However, the parametric model provides a better environment for more flexible and smoother features of the surface. Parametric modeling makes use of mathematical techniques such as Bezier, B-spline, NURBS, and T-splines. Unlike B-splines, T-splines allows local refinement while keeping the original geometry and parameterization unchanged. Hence, it is enabled to preserve the shape with relatively less number of control points [12].

The earlier approaches comprise simplification of branching problem through a quadratic mesh and triangular meshes [4],[8]. The mesh-based environment increases the complexity to encode the precise geometric information of the object, hence do not meet continuity requirement. Other methods include branch blending technique, in which Bezier patches have used for stitching of the branch segments [14]. These blending techniques may suffer from self-intersection of cross-sections and mismatching of coordinate frames at junctions. For branch reconstruction, use of parametric surface provides comparatively smoother transitions at the junction without any extra effort for segment blending and hole filling. For example, the creation of disjoint branched surfaces from B-splines [1],[3] differs from previous approaches of the surface stitching. The disjoining methods are capable of creating G1 continuous branched surface. However, the absence of local refinement is one of the major drawbacks while working with B-spline based surface model. The branched surface has to face change in topological boundaries at the junction, which increases the discontinuity over the surface. These boundaries can be detectable by sectional contours for the parametric mapping of the surface. Though the absence of the sectional curve at transition zone makes it difficult to identify the location of the junction point [9]. Ginnis et.al [5] has used the concept of Euclidian Voronoi diagram and T-spline for the interpolation of the data points obtained from the parallel cross-sections. The used approach is good in the sense of smooth surface. However, they lack the flexibility and speed of a single T-spline equation to reconstruct a branched surface.

By considering the numerous applications of reverse engineering for designing complex branch shape, the proposed method is an attempt to develop T-spline templates for the reconstruction of the tree branches. It can work well for missing point cloud data and suitable for both parallel and non-parallel cross-sections. It mainly addressed the sequencing of data points, supported by the concept of Euclidean distance.

1.1.1 *T-spline surface*

A T-spline surface is defined from a two-dimensional parametric template, called T-mesh. It allows local refinement with a smaller number of control points as compared to a non-uniform B-spline surface [2],[12]. Thus, reconstruction from a T-spline surface provides more flexible and realistic shape as compared to other mathematical surfaces (Bezier, B-splines, NURBS). The equation of a T-spline surface is given by:

$$S(u,v) = \frac{\sum w_{ij} P_{ij} B_{ij}(u,v)}{\sum w_{ij}\, B_{ij}(u,v)} \qquad (1.1)$$

Where each control point P_{ij} is associated with weight w_{ij}. For an order k the blending function is given by:

$$B_{ij}(u,v) = \; N_i^{k-1}(u)\, N_j^{k-1}(v)\,, \;\; (u,v) \in D \qquad (1.2)$$

In the above equation, u and v are parametric values, defined by a domain D of horizontal and vertical direction. Each blending function is associated with a knot vector t_u and t_v.

$$t_u = \{u_i\}_{i=1}^{k+1} \quad \text{and} \quad t_v = \{v_j\}_{j=1}^{k+1} \qquad (1.3)$$

A T-mesh is used to assign these knot vectors in both of the parametric directions. For an odd degree, all vertices of T-mesh acts as control points, and for even degree, these points are located at the center of each rectangle of the T-mesh. If m is the multiplicity of a knot value, then blending function is C^{k-m-1} continuous at that location. Thus, by controlling the continuity, blending functions can control the shape of the surface. In this paper, we are employing cubic degree polynomial to maintain C^2 continuity over the surface. For the sake of simplicity and flexibility in design, non-rational surfaces (w_{ij}=1) has used.

2 OVERVIEW OF WORK

The geometric model of the branch surface may contain several adjacent control polygon/s at different layers. These control polygons are represented by some sequence of control vertices. The polygons are linked with a suitable correspondence to create a control polyhedron. Such models either utilize presumed data set or data obtained from the original object in the form of meshes, polygons, pixels, etc. In our previous work, we had performed some experiments on T-meshes to construct bifurcations and multi-furcations from presumed branch data [7]. The basic layout of the template acquired from the prior work has displayed in Fig. 1. In present work, the assumed set of control polygons have replaced by layers of cross-sections obtained from the point cloud of the tree branches. The input sectional curves can have arbitrary alignment from each other.

The output of the current method is a closed branched surface that interpolates the layer-wise control polygon/s through a T-spline template. The branch template has formed by mapping of control points using the sequencing algorithm explained in section 3. The method proceeds in the following steps.

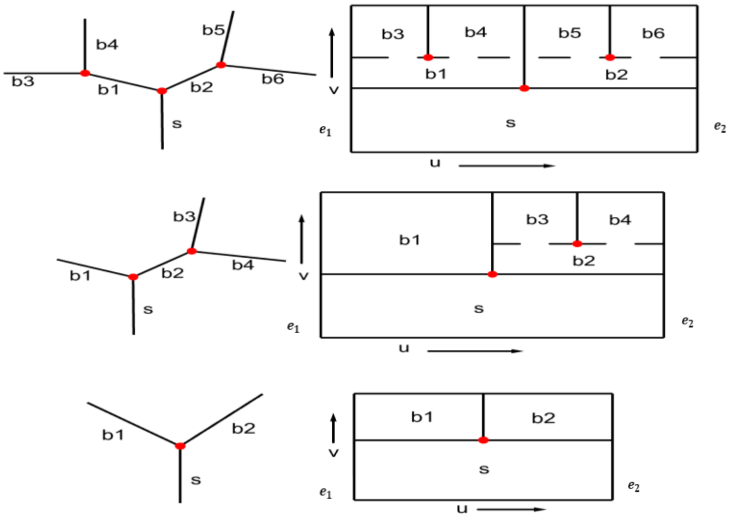


Figure 1: The layout of templates for respective branch structure.

Step 1. Extract the control polygons by dividing the point cloud of the tree branch into individual branch segments. Compile these polygons according to the independent set of branches.

Step 2. Approximation of junction point as its exact position is unknown.

Step 3. Sequencing of the control points, for construction control polyhedron. Each layer of the polyhedron may contain one or more adjacent control polygon/s.

Step 4. Map the layers of control polyhedron onto the template.

Step 5. Calculation of the knot vectors and blending functions using Eqn. 1.3 and Eqn. 1.2.

Step 6. Obtained final T-spline surface using Eqn. 1.1.

The details shall be given in upcoming sections. Our main aim is to maintain a suitable linkage between the adjacent control polygon/s of each layer for capture correct topology. Hence, the focus is placed on the second, third and fourth step respectively.

3 METHODOLOGY

In general, the main components of a branch structure are stem or parent branch, junction or node, children branch, and terminal or leaf branch, respectively. Fig. 2, illustrates the leaf branches (blue color) followed by the parent, junction (red) and children branches. Although these branches evolved from the node or junction but individually, each behaves like a stem. By arranging these independent data set, we can develop a tree branch.

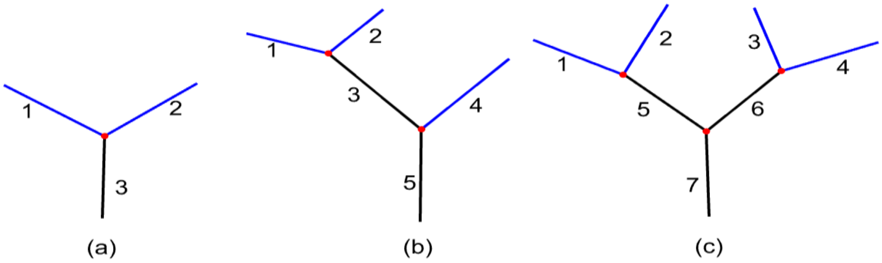


Figure 2: Terminology used for a branch structure: (a) Bifurcation, (b) Asymmetric serial bifurcation, (c) Symmetric serial bifurcation.

The work starts with the collection of point cloud data of the three reference tree branches shown in Fig 3. We have used three different scanners for this purpose. The first tree branch has been captured from a terrestrial laser scanner, FARO Focus3D X330. While the remaining two are scanned from the 3d laser scanner, EinScan-SP, and COMET L3D 5M, respectively.

Initially, the method is applied to the three test shapes drawn out from the first tree branch. These shapes encompass stem part, bifurcating branch, and multi-furcating branch respectively, shown in Fig. 4. After that, the same technique has been implemented on the second and third reference tree branch. Fig. 5 indicates the labeled discrete bifurcating branch set and their cross-sections. Labeling provides an order to store the cross-sections data of these individual branches. Later these sections are arranged through a sequencing algorithm to construct a control polyhedron.

3.1 Sequencing of Control Polygons

The mapping of obtained cross-sections in a T-spline template is done by arranging data points in the form of a control polyhedron. It represents a three-dimensional structure of the desired surface through control points. The cross-sections behave as stacks of control polygons, which are placed one over another as per the labeled branch. Fig. 6, shows stacks of adjacent control polygon/s, and the corresponding rows in the template. For ease of understanding, Fig. 8(a) represents the layered control polygon/s, comprise branch part, junction, and stem. In this context, P_s and P_e indicate the starting and terminal point of the sequenced data, that lies in each layer of the polygon/s.

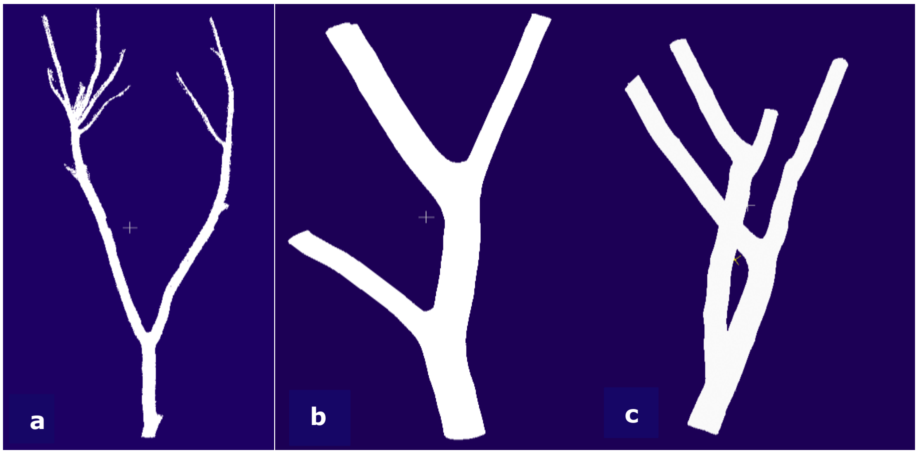


Figure 3: Point cloud data of the three tree branches: (a) First tree branch, (b) Second tree branch, (c) Third tree branch.

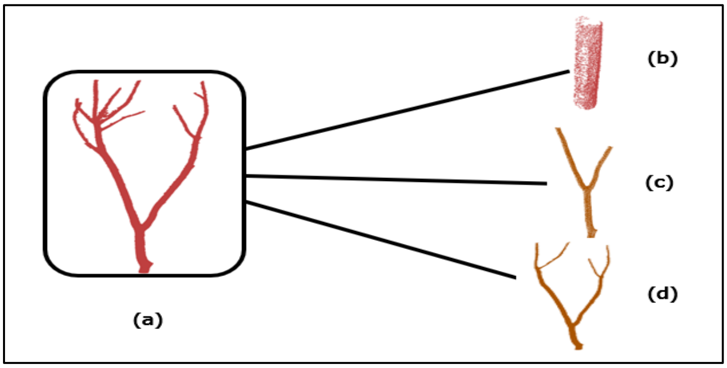


Figure 4: Initial test shapes, taken from Fig. 3(a): (a) Point cloud data of Fig. 3(a), (b) Stem part, (c) Bifurcating branch, (d) Multi-furcating branch.

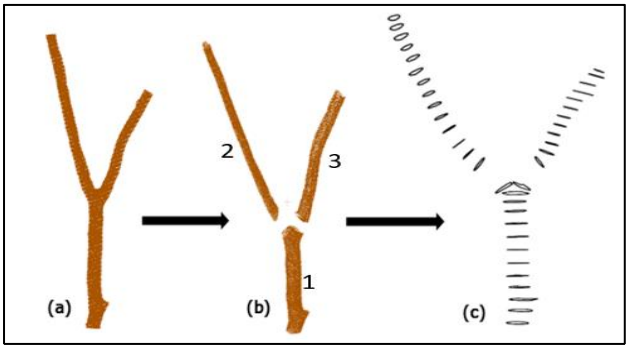


Figure 5: Individual branches and their cross-sections of Fig. 4(c): (a) PCD of Fig. 4(c), (b) A discrete branch set, (c) Cross-sections of the discrete branch set.

The disjoint points, d_1 and d_2 separates the two adjacent control polygon/s. In the multi-branch surface, disjoint points will increase with the number of adjacent branches.

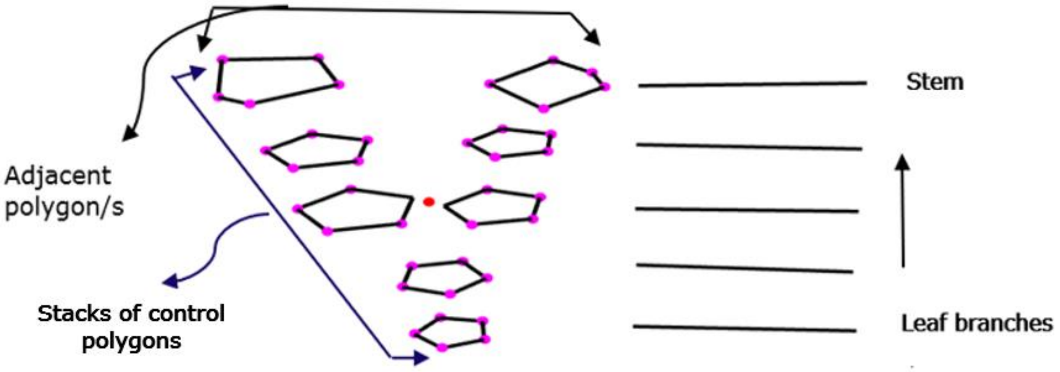


Figure 6: Stacks of control polygon/s, and the corresponding layers/rows in the template.

J is the symbol used for the approximated junction point. The red lines in Fig. 8(a), connects all the d_1 and d_2 at the junction J. We can refer it as disjoint edges. For a branch, P_s and P_e are two extreme points which belong to the farthest control polygons of the respective layer. All these indicative points are shown in Fig. 8(b). Our purpose is to identify the location of these indicative points and arrange them with remaining points to construct a control polyhedron. Since along the disjoint edges and towards the junction, the distance between the two disjoint points is decreased. Hence, in a given set of adjacent control polygons, the disjoint points lie at minimum distance. For control polygons of the first layer, starting (P_s) and end control point (P_e) have been taken at the maximum distance from their disjoint points. While in other subsequent layers they hold the shortest distance from previously stored values of P_s and P_e. The Euclidean distance between two points can estimate the location of these points.

$$E_d(X_1, X_2) = \left|\sqrt{(X_1 - X_2)^2}\right|_3 \tag{3.1}$$

Where, $E_d(X_1, X_2)$ is the Euclidean distance between two points in space.

3.1.1 *Approximation of junction point*

The approximation of the junction is a crucial step for the construction of a branched surface model. The location of the junction or node affects the shape of the surface, as it connects the parent and children branches. In the present method, we have used the two adjacent polygons, which do not meet each other at the junction but situated in the vicinity of it.

In Fig. 7, J$_1$ and J$_2$ are the approximated junction points. According to the first approximation, J$_1$ will lie at the mid-point of d_1 and d_2. While second approximation is the intersection point of the two disjoint edges passing through d_1 and d_2. If the target junction point is T_j, then:

$$\text{Approximation error } \epsilon_i = \left|J_i - T_j\right| \tag{3.2}$$

If m is number of approximations used to estimate the position of junction point, then minimum approximation error ε_m:

$$\epsilon_{\mathrm{m}} = \min\left(\epsilon_{\mathrm{i}}\right) \qquad \forall \ i \in m \tag{3.3}$$

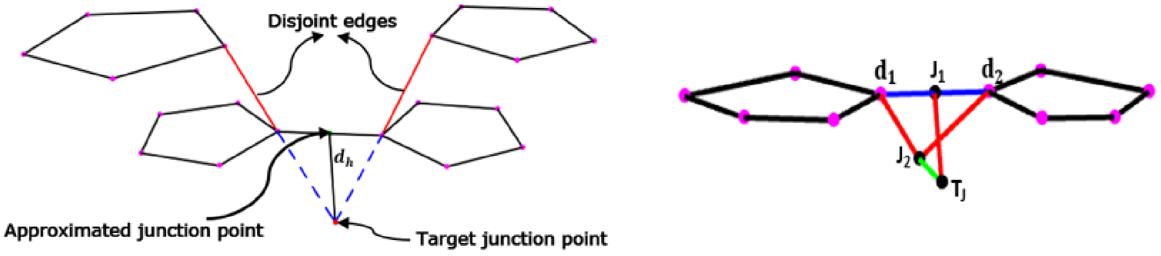


Figure 7: Approximation of the junction point.

Here, for all reference tree branches, junction point will be located corresponding to minimum approximation error.

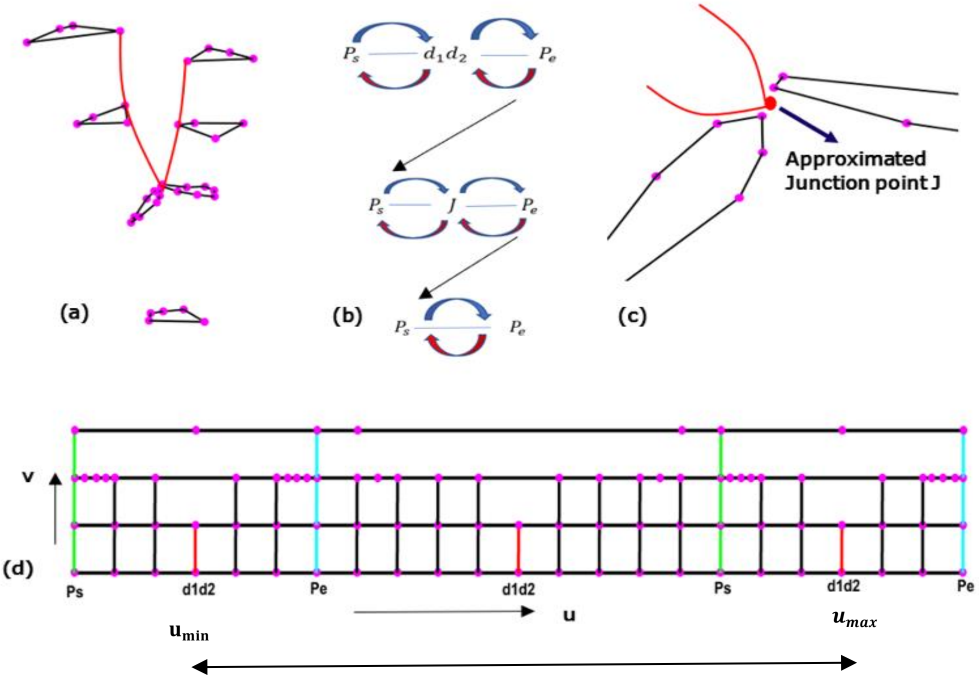
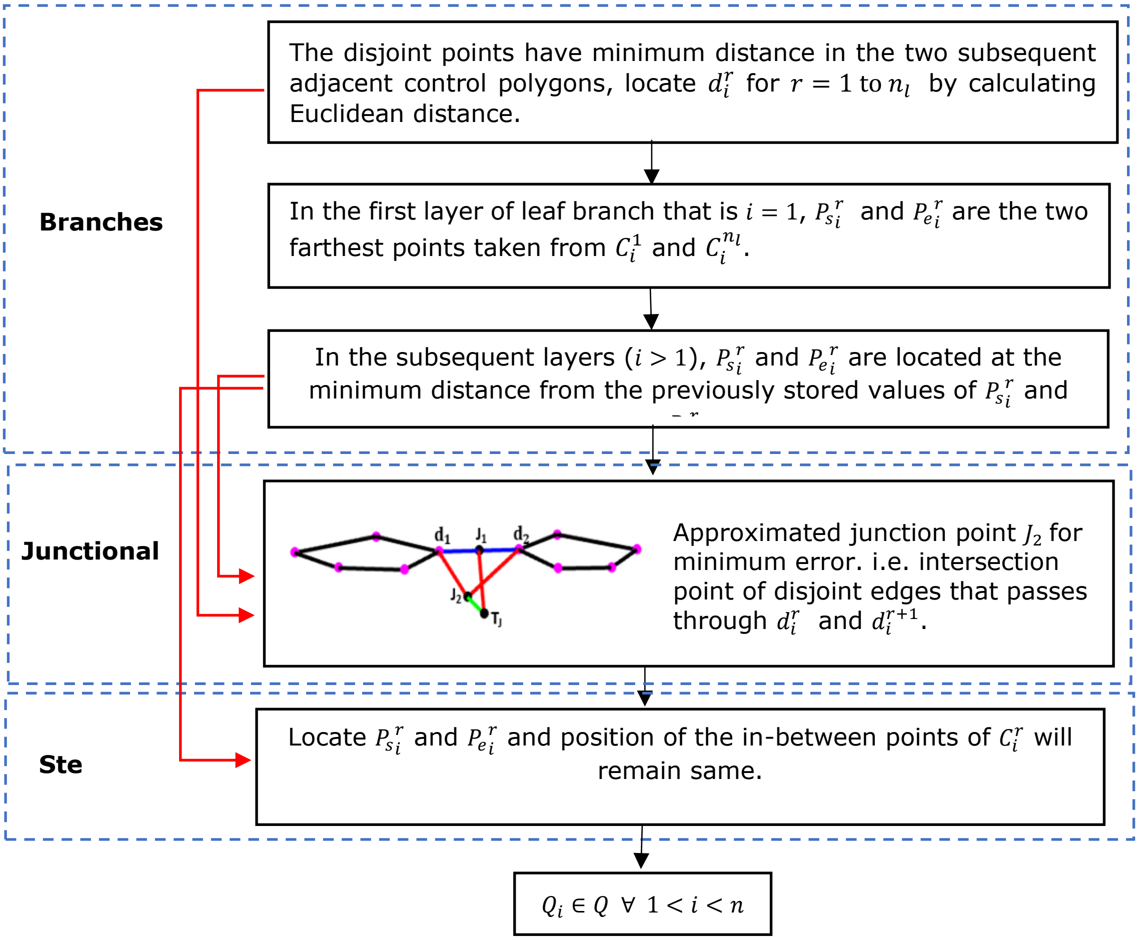


Figure 8: The process diagram of sequencing for a simple set of control polygon/s: (a) Set of control polygons, (b) Path follow up by the sequenced data for Fig. 8(a), (c) Enlarged view Fig. 8(a) at the junction, (d) T-spline template corresponding to the Fig. 8(a).

3.1.2 *Steps of algorithm*

In a fixed plane, for r number of leaf branches present in a tree branch:
C_i^r = r^{th} control polygon/s present in the i^{th} layer/row of the template.
Where, $i = 1, \ldots.n.$ and $r = 1, \ldots.n_l$
P_j^r= Control points in r^{th} control polygon/s. $j = 1, \ldots..N$
Initialize the independent set of control polygons and Q to empty, ∀ $C_i^r \in Q$
% Locate the position of disjoint points ($d_i^r, d_i^{r+1} \ldots\ldots d_i^{n_l}$), $P_{s_i}^{\ r}$ and $P_{e_i}^{\ r}$ where ∀ d_i^r, $P_{s_i}^{\ r}$ and $P_{e_i}^{\ r} \in C_i^r$



Branches

The disjoint points have minimum distance in the two subsequent adjacent control polygons, locate d_i^r for $r = 1$ to n_l by calculating Euclidean distance.

In the first layer of leaf branch that is $i = 1$, $P_{s_i}^r$ and $P_{e_i}^r$ are the two farthest points taken from C_i^1 and $C_i^{n_l}$.

In the subsequent layers $(i > 1)$, $P_{s_i}^r$ and $P_{e_i}^r$ are located at the minimum distance from the previously stored values of $P_{s_i}^r$ and

Junctional

Approximated junction point J_2 for minimum error. i.e. intersection point of disjoint edges that passes through d_i^r and d_i^{r+1}.

Ste

Locate $P_{s_i}^r$ and $P_{e_i}^r$ and position of the in-between points of C_i^r will remain same.

$Q_i \in Q\ \forall\ 1 < i < n$

The algorithm will provide disjoint points (d_i^r), the starting points $(P_{s_i}^r)$, end points $(P_{e_i}^r)$, junction point (J) and a compiled set with remaining points of C_i^r. The data set of the individual layers have stored in Q_i such that $Q_i \in Q$. The data set of Q can be plotted as control polyhedron of the respective tree branch.

3.1.3 *Template mapping*

The reconstructed branch surface is obtained by mapping of the sequenced data set Q_i onto each row of the T-spline template. The control polygon/s of leaf and stem part represent first and the last row of the template respectively. The template displayed in Fig. 8(d) is acquired by assigning sequenced control vertices of Fig. 8(a). The locations of P_s, P_e, and d_1, d_2 possessed a fixed parametric value in the template. It provides a suitable frame for the position of remaining control points. To separate branches, disjoint points d_1 and d_2 shared the same parametric position. The green and blue lines in Fig. 8(d) connects all values of P_s and P_e along with layers. In the template, it covers the forward path first, which includes control vertices lies between P_s to P_e. After that, control points of the reverse path are placed until it reaches to starting point P_s. And then the path of P_s to P_e is replicated to maintain periodicity.

The repetitions of control vertices have been done at disjoint points, junction points and for the closing of the surface. So here, we are employing some relationships introduced in previous work [7]. For even order (k=4), multiplicity of control vertices and knot values at disjoints is given by:

$$m = \frac{k}{2} + 1$$

For multi-furcation, the total number of disjoining points (N_{dk}) could be determined using n_{dk} and number of control polygon employed in the first layer of control polyhedron (n_b).

$$N_{dk} = n_{dk}(n_b - 1)$$

$$n_{dk} = 2 + \frac{k-2}{2}$$

Where, n_{dk} is the number of disjoining points for bifurcation. For a given parametric domain in u and v direction (Fig. 8(d)), the final T-spline surface can be obtained by Eqn. 1.1.

4 RESULTS

In this section, we are exhibiting the results obtained by implementing the above method in three reference tree branches.

4.1 Reconstruction of the Three Test Shapes

All the initial shapes mentioned in Fig. 4(b), Fig. 4(c), and Fig. 4(d) are addressed in this part. These shapes are point cloud data of stem, bifurcation, and a multi-furcating branch.

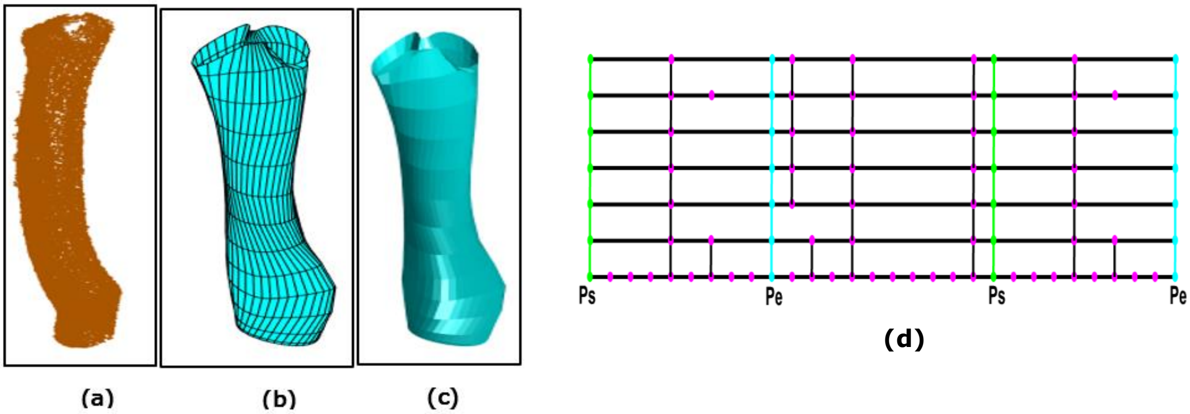


Figure 9: Reconstruction of the stem part shown in Fig. 4(b): (a) Point cloud data of the tree stem, (b) Resulting parametric surface, (c) Resulting surface with flat shading, (d) The template used for its reconstruction.

The surface depicted in Fig. 9(b), is obtained by T-spline fitting through the template shown in Fig. 9(d). This shape has no adjacent branches for disjoint points. The seven rows (i=7) are corresponding to layers of control polygons. The parametric position of P_s and P_e are constant throughout the template. Except for the first layer, other layers have single control polygon. For the first row, r has two control polygons, and thus n_l=2. All control vertices between P_s to P_e and P_e to P_s, are placed according to the sequencing algorithm.

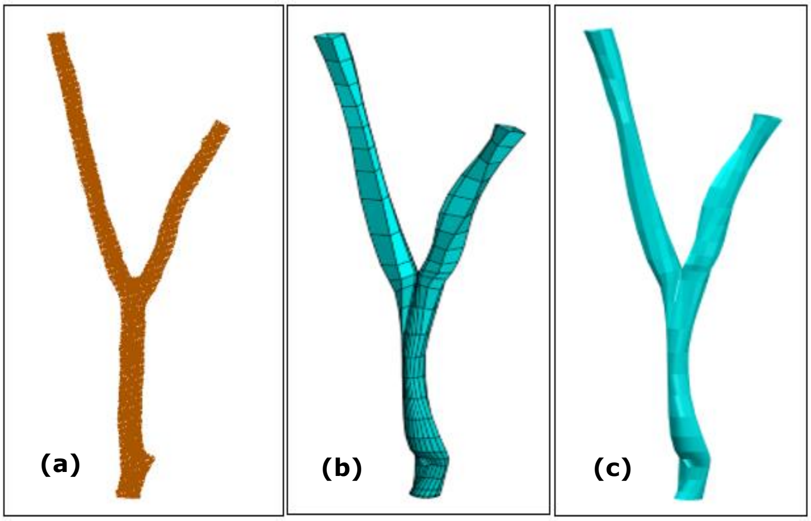


Figure 10: Reconstructed bifurcating branch: (a) Point cloud data of the branch, (b) Resulting parametric surface, (c) Flat shaded surface.

Fig. 10 and Fig. 11, shows the resulting surface of the bifurcating and multi-furcating branch shape respectively. In Fig. 11, corresponding to four leaf-branches, n_l =4. i.e. the first row (i=1) of the template will map four control polygons ($C_1^1, C_1^2, C_1^3, C_1^4$) with six disjoint points. While for the stem of the branch, i=n will be the last row of the template. The number of control points may vary for all r.

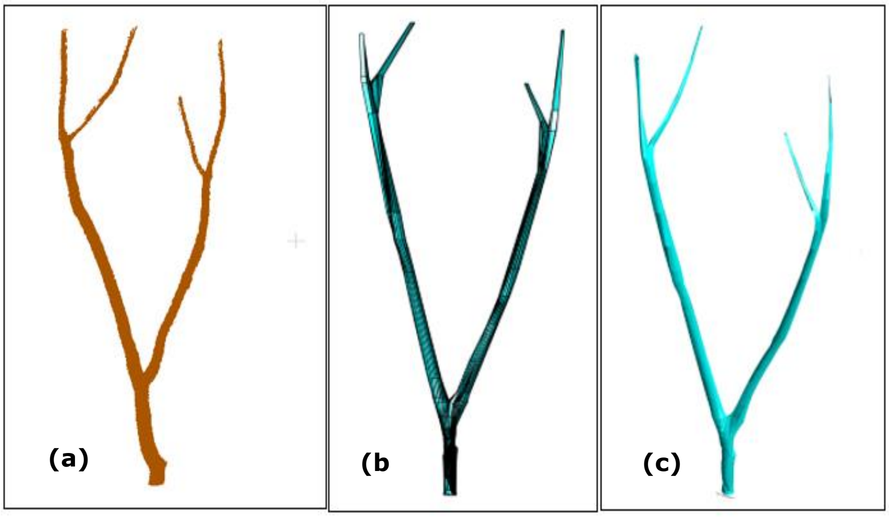


Figure 11: Reconstructed multi-branch: (a) Point cloud data of the branch, (b) Resulting parametric surface, (c) Flat shaded surface.

4.2 Reconstruction of the Second Tree Branch

The resulting parametric surface illustrated in Fig. 12(b), is obtained by interpolating all the sequenced data in the template given by Fig. 12(d). In the reference tree branch, we can see three leaf-branches, followed by one intermediate branch and stem. In the first layer/row there are three adjacent control polygons. The number of control polygon will remain same until it crosses first junction J_1. After that, each row will possess two control polygons, until it reaches the stem part (top row of the template).

Here, the first control polygon has been taken out from the left leaf-branch (see Fig. 12(a)). However, we can also initiate from its right side. The point P_s lies in polygons of left leaf branch and stem (green line in the template). And point P_e belongs to the polygons of right-side branches and stem. Here, we have four disjoints edges corresponding to two junction points. The first disjoint edge is obtained by connecting d_1's (disjoint points) of all layers, likewise others too. The first red line (longer one) shown in Fig. 12(d) separates the two edges through d_1 and d_2 for J_2. While the second one (shorter one) indicates the bifurcation through d_3 and d_4 for junction J_1. The referred template, showing only a few control points. However, other points have placed in between P_s to P_e and P_e to P_s, according to their sequence.

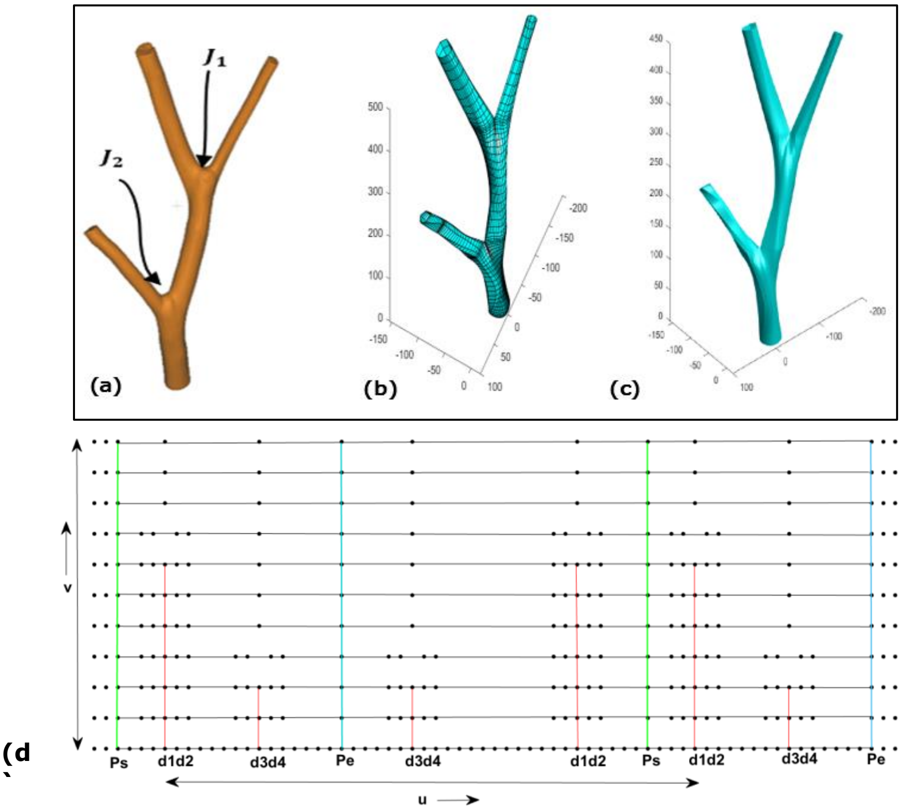


Figure 12: Reconstructed multi-branch and corresponding template layout: (a) The reference tree branch, (b) Resulting parametric surface model, (c) Surface model with flat shading, (d) Template layout used for the branch surface.

4.3 Reconstruction of the Third Tree Branch

Fig. 13(b), is a parametric surface model obtained from the reference branch shown in Fig. 13(a). The interpolation of the sequenced points has been done through the T-spline template given by Fig. 13(d). This tree branch comprises four leaf-branches and six disjoint edges (by joining all d_1, d_2, d_3, d_4, d_5, d_6 along each layer) associated with three junction points (J_1,J_2 ,J_3). Corresponding to four-leaf branches, the first row of the template will hold data points of four control polygons. While moving towards v-direction, the number of polygons will decrease. After crossing J_2 (i=12), the last three rows will map the control polygons of the stem part. The starting and end edges are depicted by the green and blue color in the template.

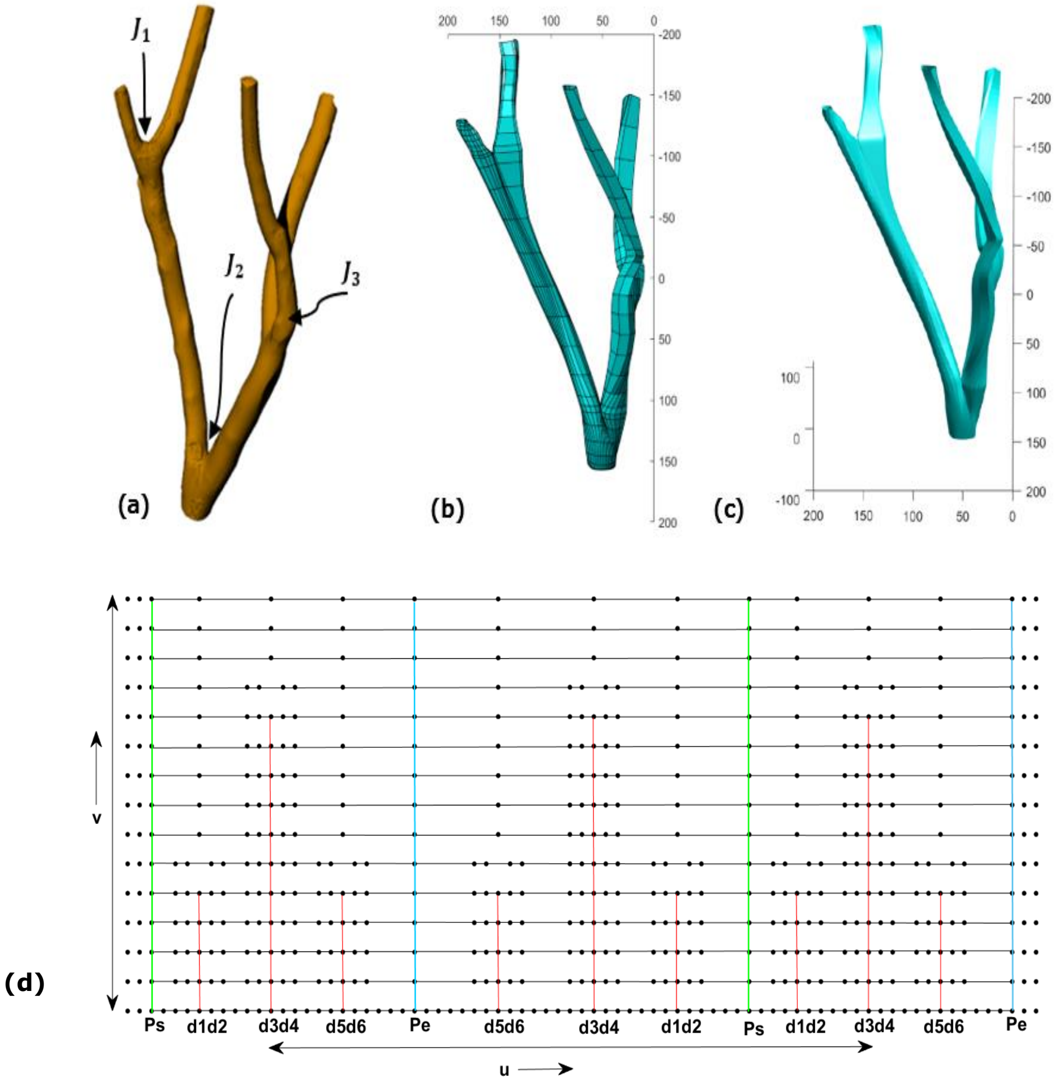


Figure 13: Reconstructed multi-branch and corresponding template layout: (a) The original tree branch, (b) Resulting parametric surface model, (c) Surface model with flat shading, (d) Template layout used for the branch surface.

5 DISCUSSIONS

The purpose of this work to develop a T-spline template for the reconstruction of multi-branch surface models. Therefore, the method has been implemented on three tree branches. To achieve this, we made a sequenced database from an independent set of control polygons so that we can impose it onto a template. The algorithm is based on distance measure criteria to locate the position of control points in 3d space. The misalignment of parametric lines can be avoided by estimating the shortest projected distance from the previously stored values.

5.1 Variations in Average Diameters

The obtained surface models have been tested to estimate diametric variation between the model and reference branch. These surface models are shown in Fig. 12(c) and Fig. 13(c). The variation has been plotted against the interval marked on each branch, displayed in Fig. 15(a) and Fig. 16(a).

In the first graph, there are total 22 marked points along the length of the branch. For understanding, Fig. 15(c) indicates few widely spaced mark points. Fig. 15(b) depicts relative error at the marked points. Positive error at some of the marked points specifies the larger diameter of the surface model as compared to the reference tree branch. From the error plot, it can be seen that deviation is maximum at point 11 (below the junction J_2).

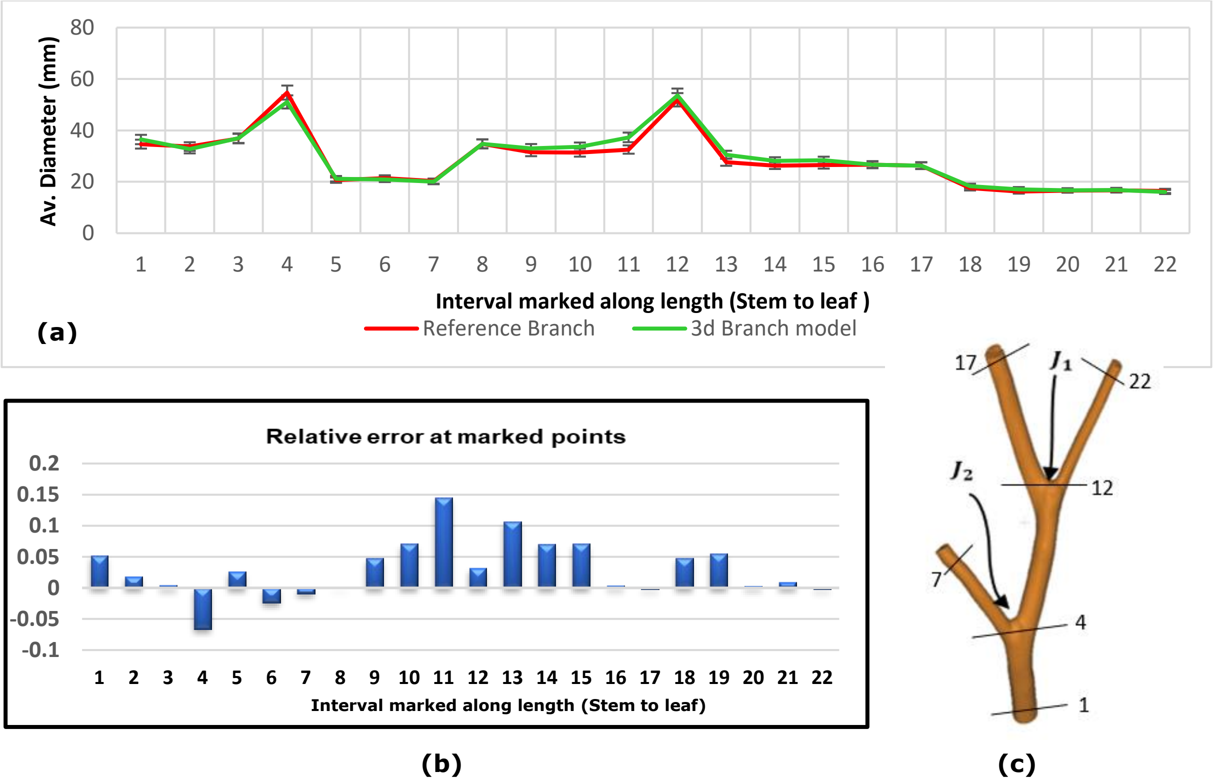


Figure 15: Diametric variation for Fig. 12(c): (a) Comparison of average diameter against the marked interval on tree branch, (b) Relative error at the marked points, (c) Reference tree branch with few markings.

The diametric variation of another branch surface model (Fig. 13(c)) is given by Fig. 16. The graph is plotted against the 32 marked points in the tree branch and few of them are visible in Fig. 16(c).

We can see the dominance of positive error points, with its maximum value at 13. At these points, the model size is greater than the reference tree branch.

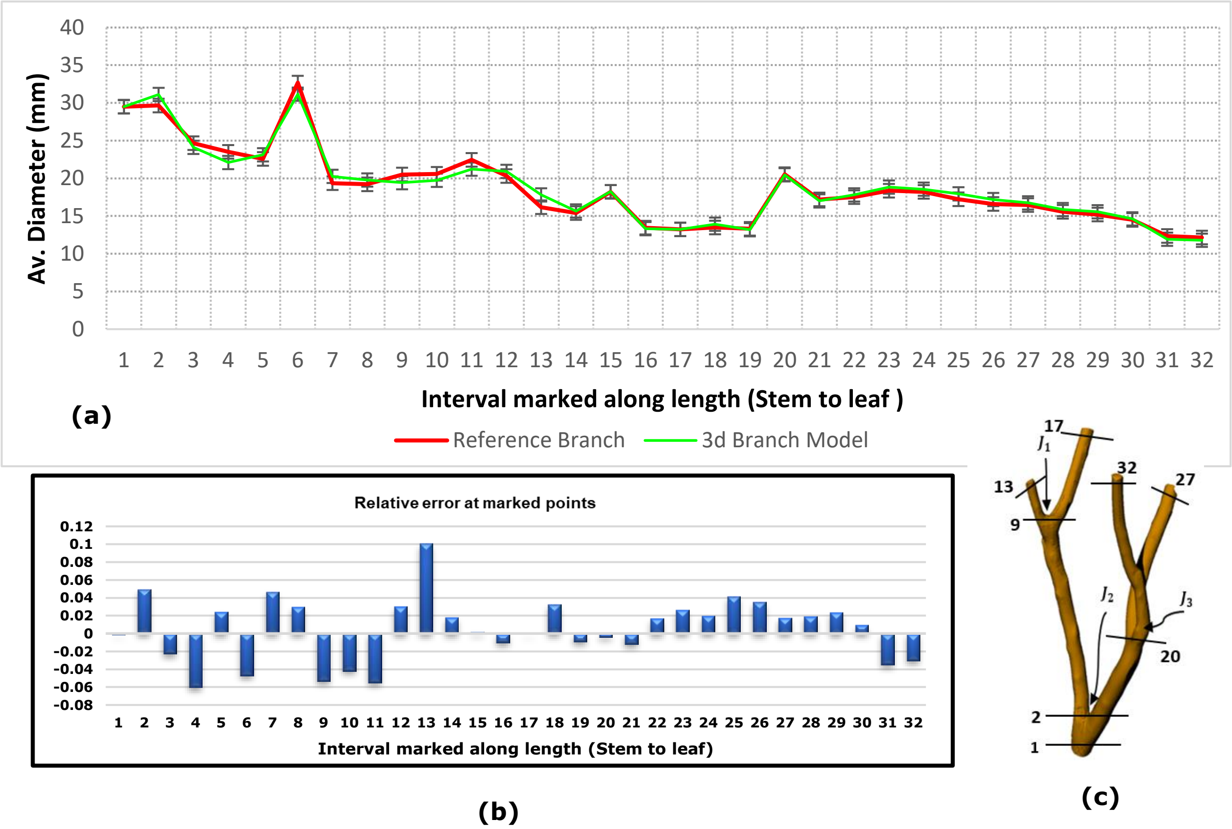


Figure 16: Diametric variation for Fig. 13(c): (a) Comparison of average diameter against the marked interval on tree branch, (b) Relative error at the marked points, (c) Reference tree branch with few markings.

The possible cause of diametric variation could be approximated location of the junction point/s. The error induced at furcations can be minimized by optimizing the relative position of the junction point. The number of control vertices in control polygon, the relative distance between the successive polygon/s, are some of the adjustable parameters, which affects the shape variation of the surface.

6 CONCLUSION AND FUTURE WORK

In this work, T-spline templates have been developed for reconstruction of the branch surface. The method addressed the sequencing of data points and its mapping over the template. It is capable to interpolate non-parallel contours of point cloud data. Therefore, it can handle irregular and asymmetric profiles of a branch shape. Some of its reverse engineering-based applications include the design of lung airways, arteries, engine manifolds, etc.

The technique has given us encouraging results as it can capture the topology well. However, experimental studies are required to minimize the geometrical error. The work is still under progress for optimal approximation of junction point and complete process of surface reconstruction. The future work includes error analysis to optimize the shape of the obtained surface model.

Kritika Joshi, https://orcid.org/0000-0002-1622-8785
Amba D. Bhatt, https://orcid.org/0000-0003-0022-1930

REFERENCES

[1] Asthana, V.; Bhatt, A.-D.: G1 Continuous bifurcating and multi-bifurcating surface generation with B-spline, Computer-Aided Design & Applications, 14(1), 2017, 95-106. https://doi.org/10.1080/16864360.2016.1199759

[2] Bazilevs, Y.; Calo, V.-M.; Cottrell, J.-A.; Evans, J.-A.; Hughes, T.-J.-R.; Lipton, S.; Scott. M.-A.; Sederberg, T.-W.: Isogeometric analysis using T-splines, Computer Methods in Applied Mechanics and Engineering, 199(5-8), 2010, 229-263. https://doi.org/10.1016/j.cma.2009.02.036

[3] Bhatt, A.-D.; Goel, A.; Gupta, U.; Awasthi, S.: Reconstruction of branched surfaces: experiments with disjoint B-spline surfaces, Computer-Aided Design & Applications, 11(1), 2014, 76-85. https://doi.org/10.1080/16864360.2014.949577

[4] Felkel, P.; Wegenkittl, R; Buhler, K.: Surface models of tube trees, Computer Graphics international, 2004, 70-77.

[5] Ginnis, A.-I.; Kostas, K.-V.; Kaklis, P.D.: Construction of smooth branching surface using T-splines, Computer-Aided Design, 92, 2017, 22-32. https://doi.org/10.1016/j.cad.2017.06.001

[6] Jewkes, R.; Burton, H.-E.; Espino, D. M.: Towards additive manufacturing of functional, spline-based morphometric models of healty and diseased coronary arteries: in vitro proof-of- concept using a porcine template, Journal of Functional Biomaterials, 9(1), 2018, 15. https://doi.org/10.3390/jfb9010015

[7] Joshi. K.; Bhatt. A.-D.: Experiments with T-mesh for constructing bifurcation and multi-furcation using periodic knot vectors, Computer-Aided Design and Applications, 16(2), 2019, 382-395. https://doi.org/10.14733/cadaps.2019.382-395

[8] Klein, R.; Schilling, A.; Strasser, W.: Reconstruction and simplification of surfaces from contours, Graphical Models, 62(6), 2000, 429-443. https://doi.org/10.1006/gmod.2000.0530

[9] Kresslein, J.; Haghighi, P.; Park, J.; Ramnath, S.; Sutradhar, A.; Shah, J.-J.: Automated cross-sectional shape recovery of 3D branching structures from point cloud, Journal of Computational Design and Engineering, 5, 2018, 368-378. https://doi.org/10.1016/j.jcde.2017.11.010

[10] Meyers, D.; Skinner, S.; Sloan, K.: Surfaces from contours, ACM Transactions on Graphics, 11(3), 1992, 228-258. https://doi.org/10.1145/130881.131213

[11] Reverse Engineering- Recent Advances and Applications, InTech, Croatia, 2012.

[12] Sederberg, T.-W.; Zheng, J.; Bakenov, A.; Nasri, A.: N. L.: T-splines and T-NURCCs, ACM Transactions on Graphics, 22(3), 2003, 477-484. https://doi.org/10.1145/882262.882295

[13] Urick, B.; Sanders, T. -M.; Hossain, S.-S.; Zhang, Y. -J.; Hughes, T. -J. -R.: Review of patient-specific vascular modeling: template based isogeometric framework and the case for CAD, Archives of Computational Methods in Engineering, 26(2), 2019, 381-404. https://doi.org/10.1007/s11831-017-9246-z

[14] Ye, X.; Cai, Y. Y.; Chui, C.; Anderson, J. H.: Constructive modelling of G1 bifurcation, Computer Aided Geometric Design, 19(7), 2002, 513–531. https://doi.org/10.1016/S0167-8396(02)00131-0