PATH PLANNING AND TOPOLOGY OPTIMIZATION FOR BIOMIMETIC THREE-DIMENSIONAL BIOPRINTING

by

CAN KÜÇÜKGÜL

Submitted to the Graduate School of Engineering and Natural Sciences

in partial fulfillment of

the requirements for the degree of

Master of Science

Sabancı University

Spring 2013-2014

PATH PLANNING AND TOPOLOGY OPTIMIZATION FOR BIOMIMETIC THREE-DIMENSIONAL BIOPRINTING

APPROVED BY:

Assoc. Prof. Dr. Bahattin Koç..

(Thesis Supervisor)

Assoc. Prof. Dr. Gözde Ünal ....

Assist. Prof. Dr. Murat Kaya...

DATE OF APPROVAL: ....

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Dr. Bahattin Koc for his constant guidance, motivation and patience. I am really thankful for his direction, devotion and encouragement which kept me focused on my work. I believe that I have learned a lot from him.

I would also like to thank my dissertation committee members Dr. Gözde Ünal and Dr. Murat Kaya for their participation, time, and attention to improve my dissertation.

Additionally, I'm also thankful for Dr. Gözde Ünal for sharing the mesh file of the coronary artery in evolution of this work. Alongside, I am grateful to TUBITAK for funding this project (Project # 112M094).

It is a pleasure to thank my research group members Saime Burçe Özler and Forough Hafezi, for their endless understanding and support, which helped me lot to come this far both in academic and social life. I would like to extend my thanks to all the faculty, staff, and students I've been fortunate to encounter in the Department of Industrial Engineering during my graduate studies at Sabancı University.

Finally, I would like to thank my family, for their endless love, care, support and inspiration, without which I would not have accomplished this graduate study.

PATH PLANNING AND TOPOLOGY OPTIMIZATION FOR BIOMIMETIC THREE-DIMENSIONAL BIOPRINTING

Can Küçükgül

Industrial Engineering, Master's Thesis, 2014

Thesis Supervisor: Assoc. Prof. Dr. Bahattin Koç

Keywords: 3D bioprinting; scaffold-free tissue engineering; macro-vascular structures; biomimetic; path planning and optimization; computer-aided bio-manufacturing

Abstract

Tissue engineering is a highly promising multi-disciplinary field for development of biological substitutes to replace or enhance the functions of damaged tissue or organs. Traditionally, highly porous scaffolds have been used for most of the tissue engineering applications. However, the challenges in seeding the cells into a scaffold and possible immunogenic reactions of scaffold materials have led to a new method of bioprinting with live cells. With the recent advancement in bio-additive manufacturing, cells with or without biological active molecules and biomaterials can be bioprinted layer-by-layer to form three-dimensional (3D) tissue constructs.

In this research work, novel biomodeling and path planning methods for bioprinting are proposed so three-dimensional tissue structures could be biomimetically printed with live cells directly from medical images. First, the medical images of the targeted tissue are imaged and segmented to convert computer tomography (CT) or magnetic resonance imaging (MRI) images to a mesh model. For path planning and optimization, the generated mesh models need to be converted to computer-aided (CAD) models. The captured mesh models are converted into smooth parametric surfaces by developed novel biomodeling algorithms. Then, several bioprinting strategies are proposed to bioprint live multi-cellular aggregates using the created computer models. Because mechanically weak cellular aggregates need to be supported perfectly at each layer, several support structure generation algorithms are proposed. The proposed methods are used to make bioprinted cellular aggregates conserve their planned 3D form, while providing sufficient conditions for cell fusion. The proposed algorithms are implemented and several example tissue structures are bioprinted by directly controlling a bioprinter with the generated commands. The results show that multicellular aggregates and their support structures can be bioprinted biomimetically in the form of the biomodeled tissues.

3 BOYUTLU BİYO-BASIM İÇİN BASIM YOLU HESAPLANMASI VE TOPOLOJİ OPTİMİZASYONU

Can Küçükgül

Endüstri Mühendisliği, Yüksek Lisans Tezi, 2014

Tez Danışmanı: Doç. Dr. Bahattin Koç

Anahtar Kelimeler: 3B biyo-basım; iskelesiz doku mühendisliği; makro-vasküler yapılar; biyo-eşlenik; basım yolu optimizasyonu; bilgisayar destekli biyo-üretim

Özet

Doku mühendisliği hastalıklı veya zarar görmüş doku veya organların fonksiyonlarını yeniden sağlamak veya geliştirmek için çalışan, son derece umut verici bir multi-disipliner alandır. Şimdiye kadarki doku mühendisliği çalışmaları, genellikle gözenekli doku iskelelerin geliştirilmesi üzerinde yoğunlaşmıştır. Ancak, doku iskelelerinde kullanılan biyo-malzemelere karşı vücudun vereceği immünojenik reaksiyonlar ve iskelelere hücre ekiminin zorlukları, doğrudan canlı hücrelerin basımı (biyo-basım) yönteminin gerekliliğini ortaya koymuştur. Katmanlı-üretim ve biyo-basım alanlarındaki yenilikçi çalışmalar, canlı hücrelerin diğer biyo-malzemelerle veya tek olarak katman-katman basılarak üç boyutlu doku yapılarının oluşturulabilmesine olanak sağlamaktadır.

Bu araştırmanın amacı, üç boyutlu doku yapılarını biyo-basım yöntemi ile dokunun veya organın anatomik yapısına uygun olarak üretmek için yeni biyo-modelleme ve basım-yolları yöntemleri geliştirmektir. Basılması hedeflenen dokunun anatomik yapısına uygun üretmek için, ilk olarak medikal görüntüleri ağ modeline çevrilerek bilgisayar ortamına aktarılır. Geliştirilen yeni biyo-modelleme metotlarıyla elde edilen bu ağ modeli, basım yolu hesaplamaları ve optimizasyonun yapıla bilinmesi için parametrik yüzey modeline dönüştürülür. Bu modeller kullanılarak, optimum basım yollarının hesaplanması için metotları geliştirilmiştir. Canlı hücreler mekanik açıdan zayıf olduklarından onları basılacak katmanlar boyunca destekleyip bir arada durmalarını sağlayacak destek yapıları geliştirilmiştir. Böylelikle canlı hücreler basıldıkları formu koruyacak ve füzyonları kolaylaşacaktır. Geliştirilen biyo-modelleme ve basım yolu hesaplama algoritmaları ile biyo-yazıcı kontrol edilerek farklı doku yapıları katman-katman canlı hücreler kullanılarak basılmıştır. Elde edilen sonuçlar ile canlı hücreler ve destek yapıları ile basılan doku yapıları, biyo-modellenen dokunun anatomik yapısına birebir benzerlikte üretilebileceği gösterilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction and Literature Review

1.1 Introduction

Maintaining the physical well-being and healthy life is one of the most important elements for an individual. However, malfunctioning or failing organs or tissues hamper one's health greatly. Especially, cardiovascular organ failures are the primary reasons of deaths and they rank among the top ten leading causes of morbidity and mortality [1]. Among several treatment methods, autografts and blood vessel transplantation are the most effective treatments for cardiovascular diseases. However, their use is limited because of the limited numbers of autografts at donor site and the patient's deficient health conditions. Recently, tissue engineering is a highly promising multi-disciplinary field for development of biological substitutes to replace or enhance the functions of defected tissue or organs for treatment of cardiovascular diseases [2, 3, 4].

Early tissue engineering strategies have involved developing a synthetic, biological or composite scaffold and seeding cells into it. There-dimensional (3D) scaffolds aim to take over the role of extracellular matrix (ECM), to supply a suitable environment for cell attachment, proliferation and differentiation and have the same functional role until the cells create their own ECM. With recent advancements in additive or layered-based manufacturing, biofabrication or bioprinting techniques have recently been developed for tissue engineering [5]. It is possible to fabricate tissue scaffolds with precise geometries layer-by-layer according to a computer-aided design model of the respective tissue or body part [6, 7]. Especially, synthetic-biologic hydrogel hybrids with their biochemical and mechanical properties mimicking the native ECM are strong 3D cell culture platforms for cell physiology and tissue printing studies [8]. However, there are very few biomaterials which can effectively mimic the natural ECM environment. Moreover, a scaffold material should maintain integrity of tissue growth, controlled degradation and should be nontoxic and nonimmunogenic [9, 10]. It is also essential to control the micro-architecture of scaffolds. Several researchers have investigated designing functionally gradient porous scaffolds with controllable heterogeneous porous architecture [11, 12].

Because of these challenges and drawbacks of the scaffold-based methods, the recent vascular tissue engineering studies focus on scaffold-free techniques. In scaffold-free tissue engineering, spherical/cylindrical cell aggregates with or without biomaterials are used as building blocks to create 3D tissue constructs.

Scaffold-free tissue engineering approaches are generally based on bioprinting or direct cell writing. There are three main branches describing the variety of technologies of scaffold-free bioprinting, namely, inkjet-based, direct laser writing, and extrusion/deposition based bioprinting [13]. Inkjet-based printing is developed to print bioink, which combines biomaterials and cells in the form of droplets. Its high-throughput efficiency and cost effectiveness make this approach highly versatile [7]. Inkjet based approaches are generally based on two technologies namely continuous inkjet (CIJ), where small droplets with a stable flow made by fluid instability through a nozzle on a passage and drop-on-demand (DOD) inkjet where ink droplets are produced when they need to be deposited [7]. Direct laser printing has an advantage to have high resolution over other bioprinting methods. However it has its own disadvantages like process-induced cell damage and toxic photo-initiator usage [7]. In extrusion-based printing and direct cell writing, strands of biomaterials or living cellular aggregates can be printed continuously layer-by-layer. Moreover, they provide adequate mechanical integrity to fabricate 3D structures [7]. Scaffold-free tissue engineering has got much superiority over scaffold-based one, such as, simple scale-up and automation, vascularization advantage in thick tissues and accurate parallel deposition of various types of cells [7, 14].

In the literature, there have been a few research focusing on scaffold-free tissue engineering of small-diameter, multi-layered, tubular vascular and nerve grafts [4, 9, 13, 14, 15, 16]. Different 3D bioprinting systems have been proposed to fabricate vascular structures. A platform-assisted 3D inkjet bioprinting system was used in order to fabricate NIH 3T3 mouse fibroblast-based tubes with an overhang structure having post-printing cell viability above 82% [14]. Multicellular spherical and cylindrical aggregates have been bioprinted to achieve flexibility in tube diameter and wall thickness and to form branched tubular structures [13, 15, 17]. However, cell aggregates should be perfectly supported by hydrogels for 3D printing. Human embryonic stem cell spheroid aggregates consisting controllable and repeatable sizes are fabricated with a valve-based cell printer [18]. According to that work, the printed stem cells have high viability after printing and are able to differentiate into any of the three germ layers. Nevertheless, the formation of large amounts of spherical aggregates requires a lot of time and the fusion process of the spheroids is completed in 5-7 days [18]. On the other hand, it is possible to fabricate more controlled structure in a short time using cylindrical cell aggregates (bio-ink). Moreover, the fusion of cylindrical bio-inks takes relatively short time (2-4 days) [17].

Even though recent studies in bioprinting have advanced tissue engineering greatly, fabricating complex biological tissues or organ constructs biomimetically has been still lacking. Bottom-up scaffold-free approaches have a great potential to provide the necessary level of flexibility for patient specific, customized tissue or organ biofabrication [19, 20]. However, biomimetic and patient specific computer-aided modeling of tissue or organs including crucial information of tissue/organ’s biological, biophysical, and biochemical properties should be developed [21].

In the literature, several modeling and reconstruction applications on hard tissues and bones has been developed [22, 23]. While mimicking the interior micro architecture of the fabricated tissue, scaffold’s porosity ratio is analyzed in these works. Moreover, a micro channel vascular network of a rat liver is generated, considering vascular design parameters such as branching angles and diameters to reach an anatomically correct representative model [24]. However, path planning and optimization for bioprinting directly from computer models need to be developed to achieve the goal of patient specific, customized organ and body part fabrication.

Additionally, bioprinting has its own challenges such as printing time and cell viability, limited number of biomaterials that can be used and biological and physical constraints. The bioprinting process needs to be performed in as minimum as possible amount of time, so that the cellular aggregates only face stress and lack of medium for a short period and hence, detrimental effects on cell viability can be minimized. Moreover, there are limited numbers of material and biomaterials available to be used in bioprinting, therefore; selecting the most appropriate biomaterial-cell combination for the desired task is very critical. Lastly, sufficient geometric conditions must be satisfied with path planning in order to enable cell fusion for both in between layers and within the layers after bioprinting.

The main goal of this research work is to develop novel computer aided algorithms and strategies to biomodel and generate path planning for 3D bioprinting of blood vessel constructs biomimetically. While generating a path plan for 3D bioprinting, the focus is centralized on obtaining an anatomically correct representative/substitute of the desired vessel. Therefore branching topology and length information for each separate branch of the original vessel needs to be preserved. A macro-vascular model is generated biomimicking real blood vessel models directly from medical images. Main blood vessels that are used throughout this research are descending human abdominal aorta (Figure 1.1(a)) [25] and coronary arteries (Figure 1.1(b)) [26].

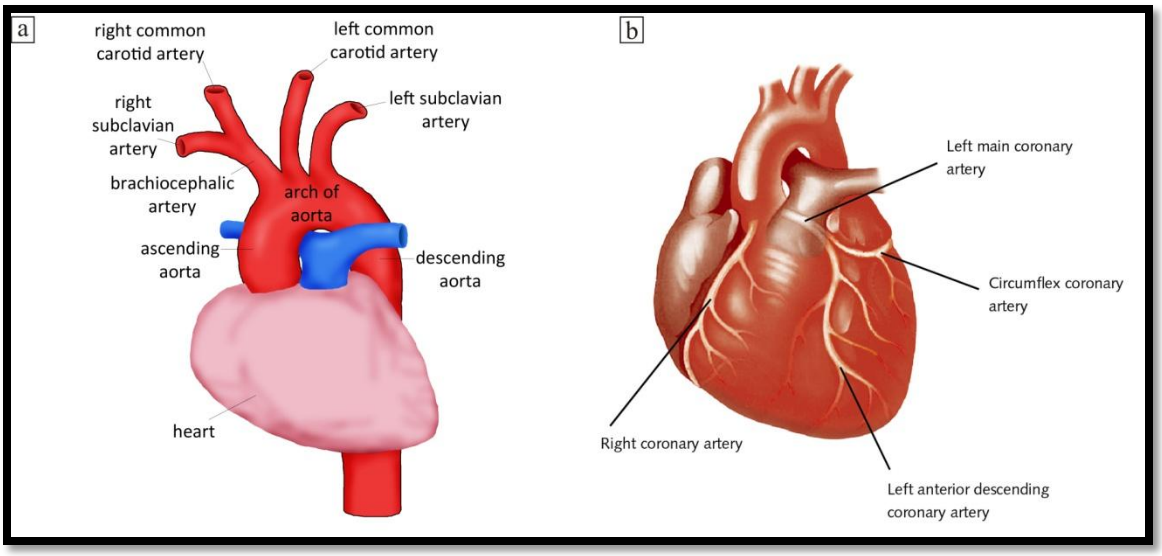
a
right common carotid artery
left common carotid artery
right subclavian artery
left subclavian artery
brachiocephalic artery
arch of aorta
ascending aorta
descending aorta
heart
b
Left main coronary artery
Circumflex coronary artery
Right coronary artery
Left anterior descending coronary artery

Figure 1.1. Representative images of human aorta and coronary arteries [26].

1.2 Organization of Thesis

The presentation of this thesis is organized as follows: 3D imaging and biomodeling is discussed in Chapter 2. Path planning generation for bioprinting is presented in Chapter 3. Implementations and examples of the developed methods are presented in Chapter 4. Then conclusions and future studies are given in Chapter 5.

Chapter 2

3D Imaging and Biomimetic Biomodeling

2.1 3D Imaging of Vascular Constructs

To be able to mimic and 3D bioprint a tissue, the anatomically-correct geometry of the targeted tissue has to be obtained and converted into a computer-aided design (CAD) model. Medical images such as Magnetic Resonance Imaging (MRI) or Computer Tomography (CT) are used for capturing the anatomically correct models of targeted tissue or organs. To capture the 3D geometry of a tissue or organ, the medical images need to be imaged and segmented. For segmentation, the Mimics (Medical Image Segmentation for Engineering on Anatomy) software [27] is used.

To demonstrate the proposed methodology, a part of human abdominal aorta model obtained from a computer tomography scan as shown in Figure 2.1. In Mimics, a part of the vessel is masked from the scan image, which contains the geometrical information of aorta. Then, region growing method is used to capture the 3D geometry of the aorta. Then, the segmented part of the aorta is converted into a 3D mesh model. Figure 2.1 shows the imaging and segmentation steps for capturing the geometry of the abdominal aorta.

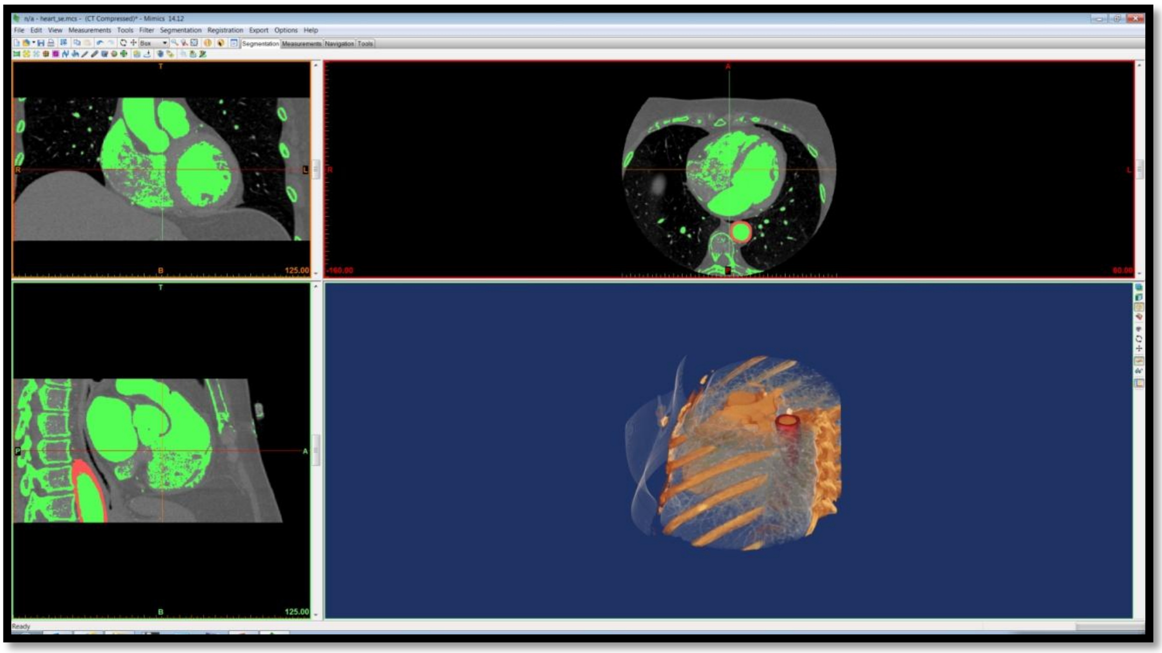

Figure 2.1. Segmentation phase of an aorta vessel, from abdominal region.

The initial geometric information of the model is represented as a mesh model or stereolithography (STL) model. The STL files are generated by tessellating the outside surface of the object with triangles. The STL model of the extracted blood vessel is shown in Figure 2.2.

2.2 Biomodeling of Vascular Constructs

As explained above, an anatomically correct STL model of blood vessels are obtained using the segmentation software. As shown in Figure 2.2, the converted STL models of the vessels are not smooth and approximated with numerous triangular facets. In order to generate bioprinting path planning as well as the topology optimization for bioprinting processes, the resultant STL models need to be represented by parametric surfaces. A novel biomodeling method is developed to convert these mesh (triangular facets) models into smooth parametric surfaces to be used for 3D bioprinting. The parametric representation of vessel models also eliminates the noise stemmed from the previous segmentation phase. First, the section curves are generated from the mesh model. The center points of each section (contour) are then calculated. The generated center points are used for approximation of a centerline curve. Lastly, the NURBS surfaces are generated along the trajectory of the calculated centerline curve.

To identify the boundaries of the STL model of the vessels, edge curves EC = $\{EC_b\}_{b=0..B}$ are defined as the end sections (bottom and top curves), as shown with red and green curves in Figure 2.2 and 2.5. Those curves are basically the starting and the end section curves of the STL models. The first edge curve which is also the first section curve EC_0 is used to initiate the centerline curve extraction process. Since STL models are represented with triangulated surfaces, all faces F = $\{F_n\}_{n\ =\ 1..N}$ have three vertices and each vertex can be a part of several faces as shown in Figure 2.2(b). Initially, all the vertices V = $\{V_m\}_{m\ =\ 1..M}$ are set "*unvisited*" (V_m^{info} = 0). Once the generation of section curves started, they start to surround the mesh surface while heading towards the edge curves one by one. The trajectory that the section curves follow is their marching direction. The "*unvisited*" vertices will be marked as "*visited*" (V_m^{info} = 1) when they contribute to form a section curve with respect to the marching direction.

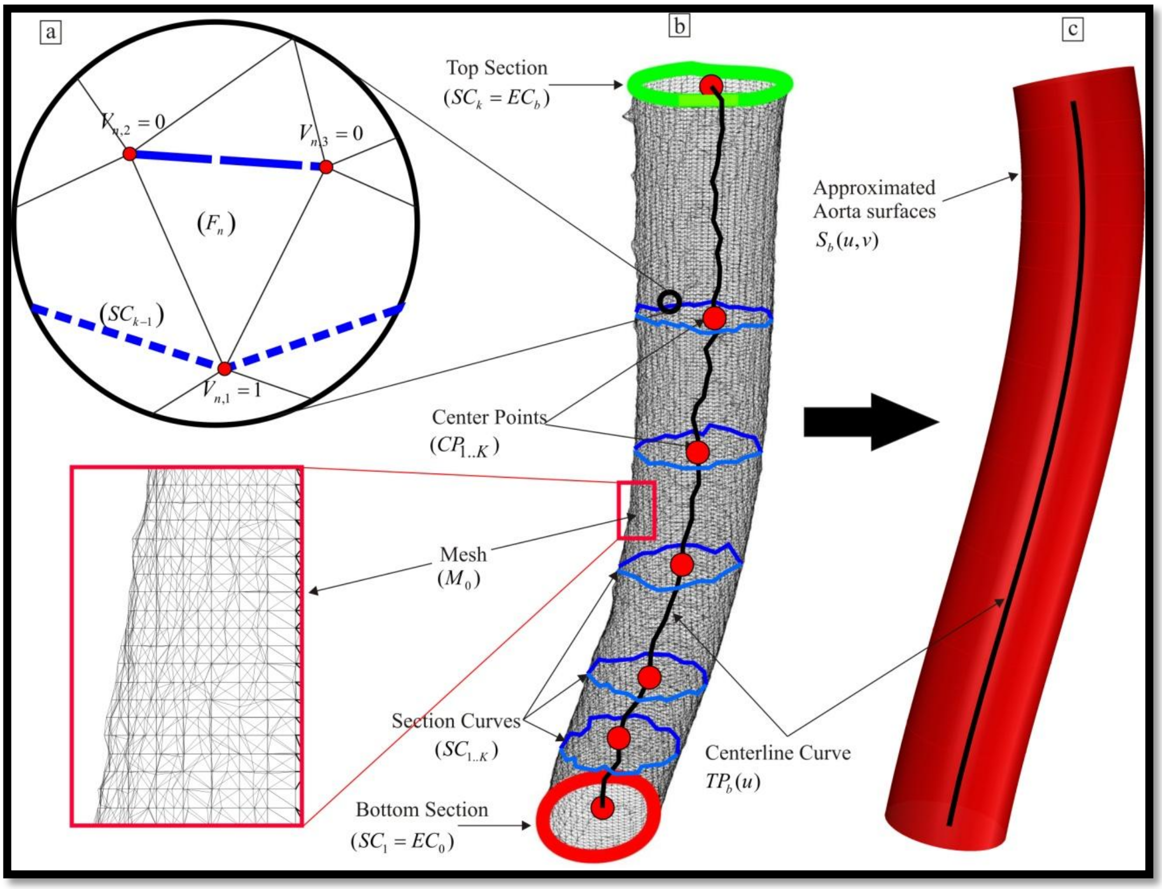

Figure 2.2. (a-b) STL file of the aorta, mesh structure. (b) The modeled aorta's edge curves (green & red) and the initial section curve (red). (b-c) STL (mesh) surface of aorta, the initial center points & smoothed representation of the centerline curve.

The set of sections is represented as SC = $\{SC_k\}_{k=1..K}$ and each section is defined with *l* vertices (points) represented as section curve vertices SCV = $\{V_{k,l}\}$. Since the end points of the face edges from the first section (EC_0) belong to a set of elements of vertices list, the algorithm marks those points as "*visited*" in order to proceed and not to visit those vertices again. Connecting these *l* vertices (points) respectively results in a closed polyline curve, thus we refer the sections as section curves as shown in Figure 2.2(b). Moreover, the set of *n* faces that are connected to each vertex *l*, FV = $\{F_{l,n}\}_{n=1..\mathrm{N}}$ are grouped and constitute the elements of the current face set CFS. As shown in Figure 2.3, the green polyline is the k^{th} section curve and red faces connected to it are the current face set. For each section curve, there will be center points CP = $\{CP_k\}_{k=1..K}$ reflecting the area weight-based center points for that section curve, and respective radius values R = $\{R_k\}_{k=1..K}$ reflecting the radius of the generated sphere of that section curve. The center points and the corresponding radius values are calculated as follows:

$$CP_k = \left(\frac{\sum\limits_{l=1..L} V_{k,l}(x)}{L}, \frac{\sum\limits_{l=1..L} V_{k,l}(y)}{L}, \frac{\sum\limits_{l=1..L} V_{k,l}(z)}{L} \right) \tag{2.1}$$

$$R_k = \underset{l=1..L}{mean}\left(\left\| \overline{V_{k,l}, CP_k} \right\| \right)$$

Where;

$SCV_k = \{V_{k,l}\}_{l=1..L}$ are set of *l* points with x-y-z coordinates, and CP_k and R_k are the center points and radius of k^{th} section curve, respectively.

Connecting center points CP = $\{CP_k\}_{k=1..K}$, along each branch form trajectory curves TC = $\{TC_i\}_{i=1..I}$. Each center point will be a part of at least one (two at joint points) trajectory curve as shown in Figure 2.4. The total number of trajectory curves is at least as number of branches as the algorithm introduces two new trajectory curves at each branching point. A better understanding of the concept is illustrated in Figure 2.5, with five trajectory curves on a three-branched coronary artery model.

Throughout the method, the area weight-based center points and their corresponding radius values are determined according to those vertices in the current section SCV, marking the vertices of SCV as "*visited*" and updating the current face set CFS like in

Figure 2.3. As shown in Figure 2.3, k^{th} section curve is highlighted with green polyline, with having red faces connecting to it as current face set. Then "*unvisited*" vertices of the red current face set generates the new $k+1^{th}$ section curve with a discrete yellow polyline, which renews the current face set with the blue faces. If a joint point is introduced (in branching parts), then the second set of faces is preserved to continue marching for a new trajectory curve at a later stage. Whenever a section curve intersects with one of the edge curves or reaches a joint point, then the current trajectory curve is finalized and the iteration continues with a subsequent trajectory curve. As shown in Figure 2.5(b), after TC_1 reaches to a joint point, TC_2 starts to march through the edge, and when one of the section curves of TC_2 intersects with the edge curve, then marching turn passes to TC_3.

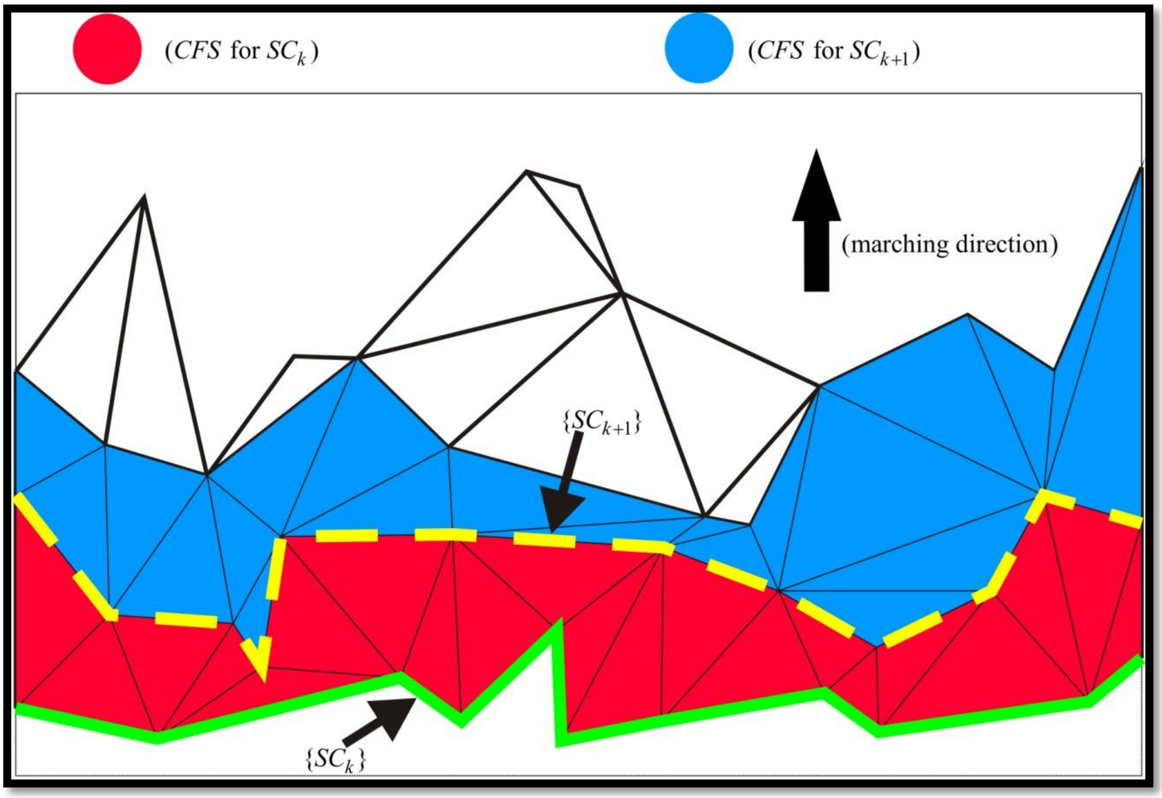

Figure 2.3. Current face set (red), for section curve *k*, and current face set (blue), for section curve *k+1*; with respect to the marching direction.

After the trajectory curves with their respective center points and radius values are calculated and stored dynamically along the marching direction, the parent-child relationships needs to be found to fit surfaces through each branch. In Figure 2.5(b), TC_1 is the parent of both TC_2 and TC_3, while TC_2 and TC_3 is the child of TC_1. If a trajectory curve does not have any child, then this trajectory curve is a leaf curve. If a trajectory curve does not have a parent, then this trajectory curve is a root curve. Starting from the root *TC₁*, each trajectory curve determines its two child (if any) trajectory curves.

When all parent-child pairs are determined, for each parent trajectory curve $TC_{i,}$ the algorithm computes the angles among each parent and two child (*t* & *t-1*) group (3 pairs for a group). In this section, for biomimetic modeling, the branching angles are not needed for surface generation; however in subsequent sections (3.4 & 3.7) branching angles information will be used to generate centerline curves in a particular 2-coordinates plane. Therefore, to convert the sum of the angle pairs for each group to a planar form and span 360 degrees, the algorithm scales the sum of three pair angles to 360 degrees (2Π in total) with the procedure below, as also shown in Figure 2.4 at the joint point:

$$ratio \leftarrow (\ \mathrm{TC}_i^{angle\ ,t} + \mathrm{TC}_i^{angle\ ,t-1} + child\_angle\ )\ /\ 2\Pi$$

$$\mathrm{TC}_i^{angle\ ,t} \leftarrow ratio \times \mathrm{TC}_i^{angle\ ,t}$$

$$\mathrm{TC}_i^{angle\ ,t-1} \leftarrow ratio \times \mathrm{TC}_i^{angle\ ,t-1}$$

Then each trajectory curves' length is determined by connecting its center points through the marching direction.

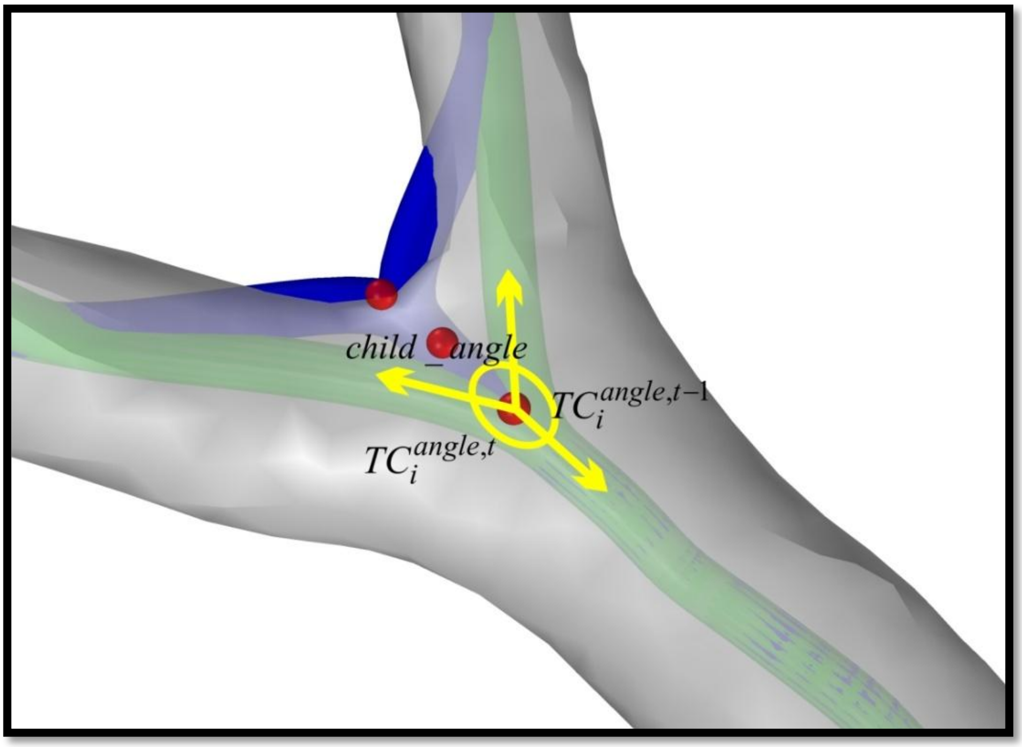

Figure 2.4. Determination of angles between a parent trajectory curve *i*, and its two child.

Moreover, a predetermined number of last center points for each trajectory curve are omitted. Because of the reason that the algorithm examines the surface information of mesh to extract the trajectory curves by using the generated center points, branching parts are realized when they already occurred. This fact causes the approximated trajectory curves to intersect with the mesh surface, as shown with blue pipes in Figure 2.4. Therefore, by omitting a number of center points from the last part (the number is determined by a function of mesh volume and total facets) the trajectory curves lies securely inside of the mesh surface, as shown with green pipes in Figure 2.4. Then, median radius values are determined for each trajectory curve in order to decrease the radius variety before the smooth surface generation, to obtain a finer surface geometry.

After each trajectory curve with their corresponding center points are calculated from each leaf curve to the root, the algorithm links child trajectory curves with its parents and stacks the respective center points and radius values in a topological order. For all branches and for each leaf trajectory curve reaching the root, the algorithm fits a B-spline curve, which will be the centerline for the respective parametric surface. This parametric B-spline centerline curve is defined as [28]:

$$TP_b(u) = \sum_{q=1}^{b_p} N_{q,p}(u) \cdot CP_q \qquad 0 \le u \le 1 \tag{2.2}$$

Where B-spline basis function is;

$$N_{q,0}(u) = \begin{cases} 1 & \text{if } u_q \le u < u_{q+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{q,p}(u) = \frac{u - u_q}{u_{q+p} - u_q} N_{q,p-1}(u) + \frac{u_{q+p+1} - u_q}{u_{q+p+1} - u_{q+1}} N_{q+1,p-1}(u)$$

CP_q's are the control points, and the $N_{q,p}(u)$ are the pth-degree B-spline basis functions as defined above with the knot vector $U=\{u_0,\ldots,u_m\}$ where u_q's be a nondecreasing sequence of real numbers.

Then the parametric B-spline surface(s) of the vessel model are generated using the centerline curve(s) $TP_b(u)$, with respect to the median radius value of relevant center points as shown in Figure 2.5(c). This operation basically sweeps a planar closed curve along the centerline curve. Denote the centerline by $TP_b(u)$ and the planar closed curve by T(v). T(v)'s radius value gets the $\mathrm{TC}_{leaf,b}^{median\_rad}$ value for the starting point, and $\mathrm{TC}_{root}^{median\_rad}$ value for the ending point of the respective centerline curve, and if there are any other trajectory curves linking leaf and the root, their respective $\mathrm{TC}_{linker}^{median\_rad}$ radius values are placed on their middle center point $TC_{linker}^{mid\_CP}$ locations (Figure 2.5(c)). A general form of the swept surface is given by [28]:

$$S_b(u,v) = TP_b(u) + M(u)T(v) \tag{2.3}$$

$$T(v) = \left(r_i^{median\_rad} \cos(v), r_i^{median\_rad} \sin(v) \right)$$

Where; $0 \le u \le 1$ and $0 \le v \le 1$

Where $M(u)$ is a 3x3 matrix incorporating rotation and nonuniform scaling of T(v) as a function of u.

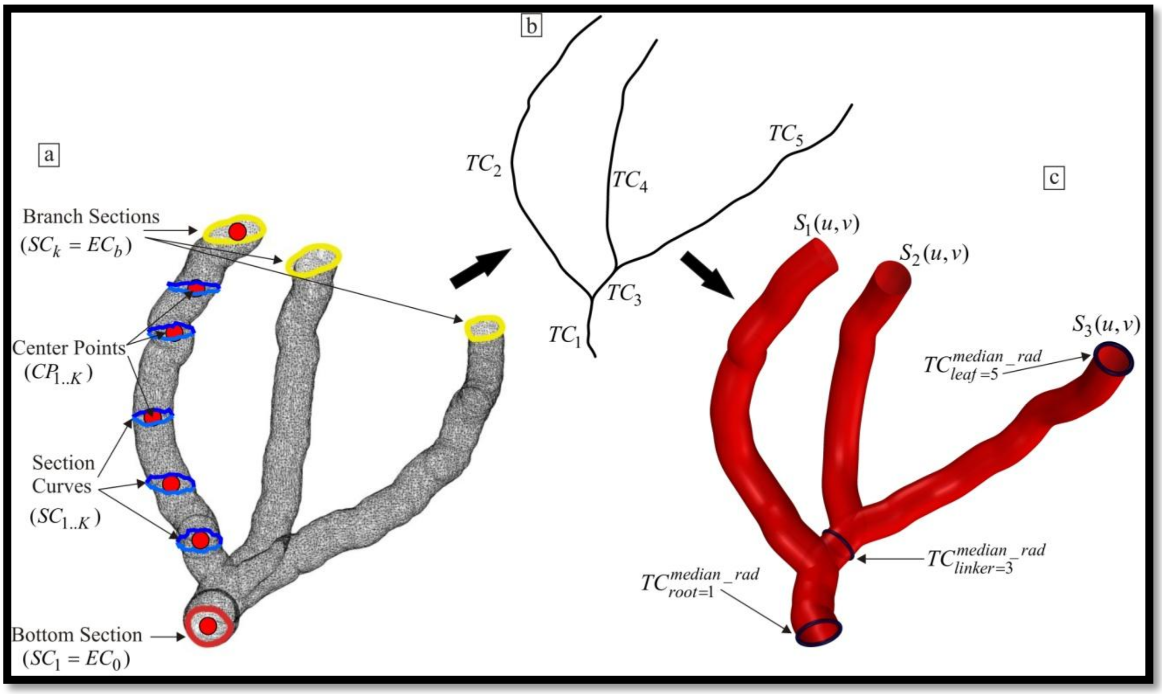

Figure 2.5. (a) Mesh model of a three-branched coronary artery vessel, (b) generated trajectory curves according to the model, (c) generated smooth surfaces with respect to trajectory curves and median radius values.

As explained above, generating smooth parametric surfaces from the mesh model is determined using Algorithm 1 for which its pseudo-code is given below:

```
Algorithm 1.  Biomimetic Smooth Parametric Surface(s) Generation
Input:
$M_0$:                            Mesh model of vessel network
EC = $\{EC_b\}_{b=0..B}$:           a set of edge curves on Mesh
F = $\{F_n\}_{n=1..N}$:             a set of faces on Mesh
V = $\{V_m\}_{m=1..M}$:             a set of vertices on Mesh
Output:
$S_b(u,v)$ :                      generated NURBS Surface(s)
TC:                               trajectory curve array
Start
Initialize $i \leftarrow 1$, $k \leftarrow 1$, $j \leftarrow 0$, $num\_Branches\_Reached \leftarrow 0$, $check\_intersection \leftarrow 0$
Initialize $p \leftarrow 1$, $t \leftarrow 1$, $cubic\_Volume\_of\_Mesh \leftarrow M_0^{volume}$
If ( $N$ / $cubic\_Volume\_of\_Mesh > 1$ ) Then { $num\_Points\_to\_Delete \leftarrow 2$ }
Else { $num\_Points\_to\_Delete \leftarrow 1$ }
For (all $V_m$) { $V_m^{info} \leftarrow 0$ = "not visited" }
$SC_k \leftarrow EC_0$
$SCV_k = \{V_{k,l}\}$ ← a set of $l$ vertices that $k^{th}$ section curve contains/intersects
For (all $V_{k,l}$) {
  $V_{k,l}^{info} \leftarrow 1$ = "visited"
  FV ← FV ∪ $\{F_{l,n}\}$ '// a set of $n$ faces that are connected to vertex $l$
}
```

```
$CFS$ ← {FV} ∪ $CFS$
$TC_i^{CP,p}$ ← $CP_k$ ‘// using Equation(2.1)
$TC_i^{R,p}$ ← $R_k$ ‘// using Equation(2.1)
While ($num\_Branches\_Reached < B$) {
  For ( $n$ = 1 to $size(CFS_0)$ ) {
    If ( $V_{n,1}^{info} + V_{n,2}^{info} + V_{n,3}^{info} == 1$ ) Then {
      If ( $V_{n,1}^{info} == 0$ ) Then { VC ← VC ∪ {$V_{n,1}$} }
      If ( $V_{n,2}^{info} == 0$ ) Then { VC ← VC ∪ {$V_{n,2}$} }
      If ( $V_{n,3}^{info} == 0$ ) Then { VC ← VC ∪ {$V_{n,3}$} }
    }
  }
  If ( $CFS_0^{CP} \neq$ NULL) Then {
    $p$ ← 1, $i$ ← $i$ + 1
    $TC_i^{CP,p}$ ← $CFS_0^{CP}$
    $TC_i^{R,p}$ ← $CFS_0^{R}$
    $CFS_0^{CP}$ ← NULL, $CFS_0^{R}$ ← NULL
  }
  CC ← check curves obtained by connecting vertices of VC in a topological order
  If ( $size$(CC) = 1) Then {
    $k$ ← $k$ + 1, $p$ ← $p$ + 1
    $SC_k$ ← $CC_0$
    $SCV_k$ = {$V_{k,l}$} ← a set of $l$ vertices that $k^{th}$ section curve contains/intersects
    For (all $V_{k,l}$) {
            $V_{k,l}^{info}$ ← 1 = “visited”
            FV ← FV ∪ {$F_{l,n}$} ‘// a set of $n$ faces that are connected to vertex $l$
    }
    For ( $b$ = 1 to $B$) {
      If ( $SCV_k \cap EC_b \neq \emptyset$ ) Then {
        $num\_Branches\_Reached$ ← $num\_Branches\_Reached$ + 1
        $check\_intersection$ ← 1
        $SCV_k$ ← $EC_b$
        $TC_i^{CP,p}$ ← $CP_k$ ‘// using Equation(2.1)
        $TC_i^{R,p}$ ← $R_k$ ‘// using Equation(2.1)
        Delete ← $CFS_0$
      }
    }
    For (all $V_{k,l}$) {
            $V_{k,l}^{info}$ ← 1 = “visited”
            FV ← FV ∪ {$F_{l,n}$} ‘// a set of $n$ faces that are connected to vertex $l$
    }
    If ($check\_intersection$ == 0 ) Then { $CFS_0$ ← {FV} − $CFS_0$ }
  }
  Else {
    $CFS_0^{CP}$ ← $CP_k$
    $CFS_0^{R}$ ← $R_k$
    $k$ ← $k$ + 1, $i$ ← $i$ + 1, $p$ ← 1
    $\mathrm{TC}_i^{CP,p}$ ← $CP_{k-1}$ '// using Equation(2.1)
    $\mathrm{TC}_i^{R,p}$ ← $R_{k-1}$  '// using Equation(2.1)
    $SC_k$ ← $CC_0$
    $SCV_k = \{V_{k,l}\}$ ← a set of $l$ vertices that $k^{th}$ section curve contains/intersects
    **For** (all $V_{k,l}$) {
        $V_{k,l}^{info}$ ← 1 = "*visited*"
        FV ← FV ∪ $\{F_{l,n}\}$ '// a set of $n$ faces that are connected to vertex $l$
    }
    $CFS$ ← ({FV} − $CFS_0$) ∪ $CFS$
  **}**
  **Delete** ← VC
  **Delete** ← CC
**}**
**For** ( $i$ = 1 to $I$) {
  $t$ ← 1
  $\mathrm{TC}_i^{length}$ ← total length of the polyline, composed of connecting $\mathrm{TC}_i^{CP}$'s from the *first_index* through *last_index* in topological order
  **For** ( $j$ = 1 to $I$) {
    **If** ( $i \neq j$) **Then** {
      **If** ( $\mathrm{TC}_i^{CP,last\_index\ (p)}$ == $\mathrm{TC}_j^{CP,first\_index\ (p)}$) **Then** {
        $\mathrm{TC}_i^{child,t}$ ← $TC_j$
        $\mathrm{TC}_j^{parent}$ ← $TC_i$
        $t$ ← $t$ + 1
      }
    }
    **Delete** ← last *num_Points_to_Delete* $\mathrm{TC}_i^{CP}$'s & $\mathrm{TC}_i^{R}$'s
    $\mathrm{TC}_i^{median\_rad}$ ← *median*( all $\mathrm{TC}_i^{R}$)
  }
}
**For** ( $i$ = 1 to $I$) {
  t ← 1
  **If** ( $\mathrm{TC}_i^{child}$ ≠ NULL) **Then** {
    *child_angle* ← *angle*({$\mathrm{TC}_i^{child,t}$, $\mathrm{TC}_i^{child,t+1}$})
    **For** ( $j$ = 1 to $I$) {
      **If** ( $i \neq j$ and $\mathrm{TC}_j^{parent}$ == $\mathrm{TC}_i$) **Then** {
        $\mathrm{TC}_i^{angle,t}$ ← *angle*({ $TC_i$, $TC_j$ })
        $t$ ← $t$ + 1
      }
    }
    *ratio* ← ( $\mathrm{TC}_i^{angle,t}$ + $\mathrm{TC}_i^{angle,t-1}$ + *child_angle* ) / 2Π
    $\mathrm{TC}_i^{angle,t}$ ← *ratio* × $\mathrm{TC}_i^{angle,t}$
    $\mathrm{TC}_i^{angle,t-1}$ ← *ratio* × $\mathrm{TC}_i^{angle,t-1}$
  }
}
$b$ ← 0
```

```
For ( $i$ = 1 to $I$) {
  If ( $\mathrm{TC}_i^{child}$ == NULL) Then {
    $b \leftarrow b + 1$
    While ( $\mathrm{TC}_i^{parent}$ $\neq$ NULL) {
      $TP_b \leftarrow TP_b \cup \{\mathrm{TC}_i^{CP}\}$
      $TR_b \leftarrow TR_b \cup \{\mathrm{TC}_i^{R}\}$
      $\mathrm{TC}_\mathrm{i} \leftarrow \mathrm{TC}_i^{parent}$
    }
    $TP_b(u)$ ← approximate a trajectory curve with the $CP_b$'s using Equation (2.2)
    $S_b(u,v)$ ← build the surface along $TP_b(u)$ with respect to its $CP_b$'s median radius values(s) using Equation (2.3)
  }
}
End
```

Chapter 3

Path Planning for 3D-Bioprinting

Previous works on scaffold-free 3D bioprinting of vascular structures are generally based on simple vertical extrusions [29]. In this chapter, we are bioprinting complex geometries of cellular structures with self supporting hydrogels. Not only because of its complex geometry, but also due to the dynamic structures of both cells and hydrogels, it is challenging to build such structures in 3D. Here, an anatomically correct representation of vessels is aimed; therefore, mechanically-weak cellular aggregates should be supported by hydrogels to have sufficient conditions for cell fusion. In order to mimic the original vessel effectively and to minimize the human interventions, topology optimization is carried out for 3D bioprinting to control the bioprinter directly from generated commands.

3.1 3D Bioprinting System

In this research, the NovoGen MMX^{TM} (Organovo) bioprinter is used for printing three dimensional biomodeled tissues. This automated 3D bioprinter has three stepper motors for X-Y-Z motion as well as two deposition heads to print hydrogel biomaterials and cellular aggregates (bio-ink). The bioprinter has a built-in controller system, where the micro-deposition is maintained throughout the printing with a laser-based calibration. Glass capillaries with 250/500 μm diameter are used as deposition tips for two deposition heads. Those capillaries are both capable of aspirating and dispensing gels or cell aggregates using a metal plunger inside of them which works as a piston moving up and down (Figure 3.1). The bioprinter has heating and cooling chambers with adjustable temperatures, ranging from 25°C to 95°C for heating and 4°C to 25°C for cooling. Thermo reversible hydrogel is placed into the heating chamber to preserve its liquid form in order to be able to aspirate it into the glass capillary. After aspiration, the gel head moves to the cooling chamber and keep the capillary, filled with hydrogel, inside the chamber for a predetermined time in order for hydrogel to become a gel. After this phase change, the gel head can dispense the material with its piston downside movement. The speed of the push down movement is exactly the same with the speed of the capillaries horizontal movement. The cell-deposition head moves the same way except the heating and cooling steps are not used for cellular aggregates (bio-ink).

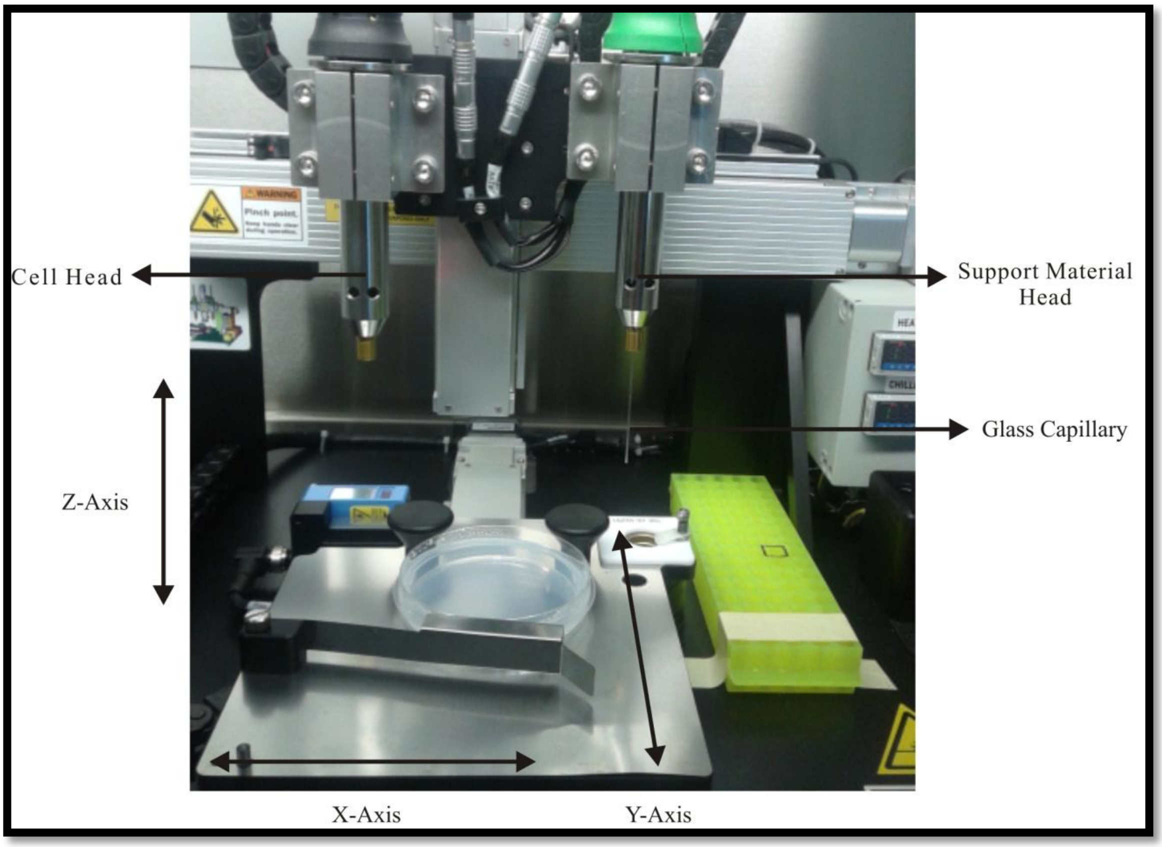

Figure 3.1. NovoGen MMXtm (Organovo) Bioprinter.

Although the bioprinter software has a built-in controller commands, they cannot be used for printing complex structures. One of the biggest drawbacks of the current software is that, it can only dispense fluid or gels in linear-movements. Because of this limitation, paths with curves need to be approximated with short linear segments. A user-generated scripts needs to be developed for controlling the bioprinter directly for printing complex 3D structures. As shown in Figure 3.2, once a planned curve trajectory

$\{C_t\}$ is determined with at most capillary volume $v_{capillary}$ length, a linear interpolation is used on the curve to extract the *n* route points $\{RP_{t,n}\}$ of planned curve $\{C_t\}$ for the bioprinter to follow the linear paths between the points in topological order as shown with red curves (bioprinter curves). Eventually there will be errors between the planned curve trajectory and bioprinter curve trajectory, as bioprinter curve trajectory shortcuts the small arc segments with small linear line segments. The maximum error, maximum linear distance between planned curve and bioprinter curve for a specific arc segment, will be the bioprinting error for the generated model. After series of bioprinting trials, its optimized that, the cylindrical planned trajectory curves are divided to its *n* route points where each linear distance between the consecutive point pairs $\{RP_{t,n},RP_{t,n+1}\}$ of 0.3 mm gives the best result with 3 mm/s deposition speed, in terms of planned shape formation. With these parameters, the bioprinting errors for macro-vascular models are between 0.05 mm minimum and 0.12 mm maximum.

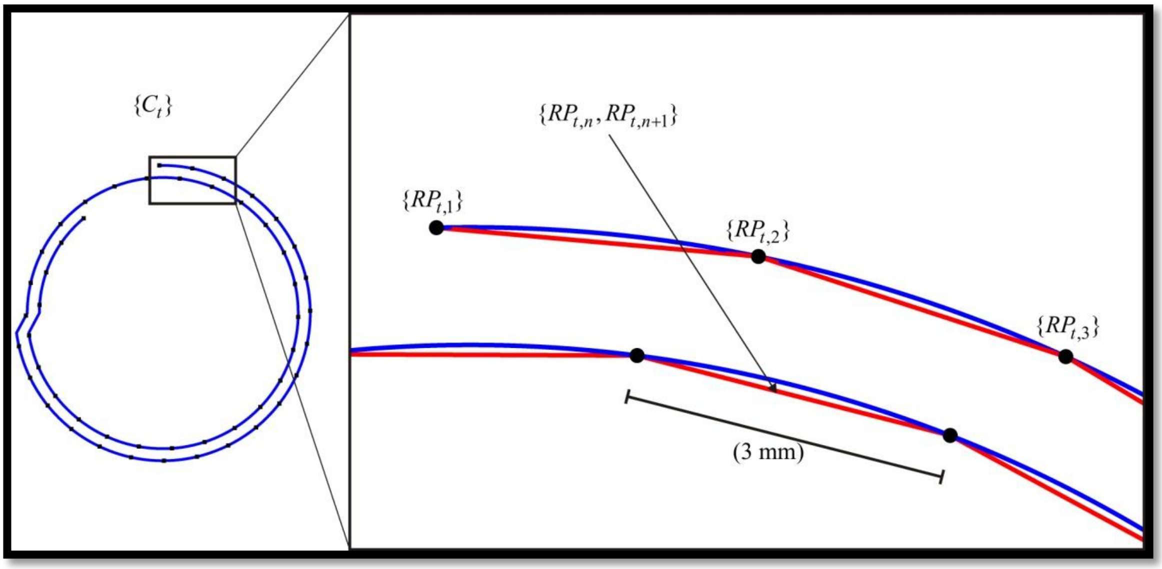

Figure 3.2. Cylindrical trajectories curve that lengths capillary volume is separated to its route points; route points are connected to form the bioprinting path for that curve.

3.2 3D Bioprinting of Biomimetic Aortic Vascular Constructs with Self-Supporting Cells

After the smooth surface model of aorta $S_b(u,v)$ is generated in biomodeling section (Chapter 2), an optimum 3D bioprinting topology needs to be determined in order to obtain an anatomically correct representation of the printed vessel. Path planning for both cellular aggregates and hydrogel support structures is calculated in this section. Both cellular aggregates and support structures are printed by a glass capillary in a gel like form layer by layer to form the 3D tissue construct. Because of the fact that the bioprinted materials are not self-shape conserving, both cells and support structures should accordingly be placed on the valleys of the preceding layer (shown in Figure 3.3) in order to provide cell fusion and structure conservation and most importantly, to reach correct anatomical model of the original vessel.

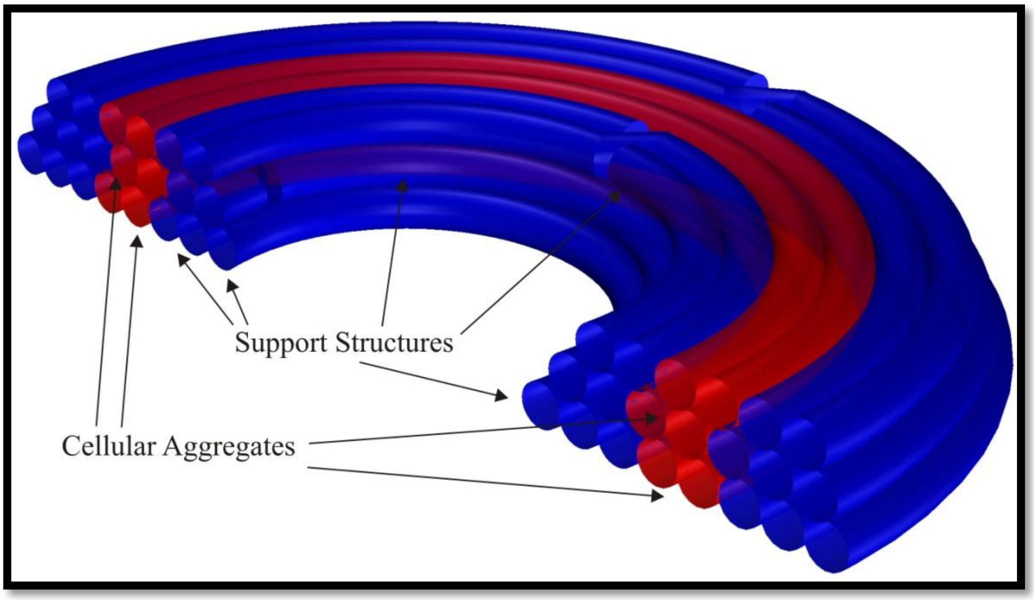

Figure 3.3. Three consecutive example layers showing how support structures (blue) and cellular aggregates (red) are placed on the valleys of the preceding layer.

In the model, the height increments between consecutive layers is slightly less than the diameter of the capillary tubes, the total number of layers (*totalLayers*) is calculated by dividing the surface height to the *height_increment* amount. The vessel's surface representation is then sliced with successive planes which resulted in contour curves $C_{j,0}(t) = \{c_{j,0}\}_{j=1..totalLayers}$ for each layer as shown in Figure 3.4. The number of cylinders for each layer is then determined by the $maxStep_j$ value from $maxStep_j = topSupport + totalLayers - j$ where *topSupport* is the number of support cylinders on top layer that is entered by user. Since $maxStep_j$ variable is dependent to layer number, its value is maximum initially and drops by one at every consecutive layer through top, which provides constant elevation between successive layers.

To conserve the general shape of the vessel on each layer and to prevent the deformation of weak cellular aggregates, each contour curve is offset using the $maxStep_j$ value of the specific layer on *x*,*y*-plane as shown in Figure 3.4. The initial offset amount $O_{j,i} = \{o_{j,i}\}_{i=1..maxStep_j}$ for a layer can be found by the following formula:

$$o_{j,i} = \left(\left(maxStep_j/2\right) - \left(i-1\right)\right) \times d_{capillary} \tag{3.1}$$

Where $d_{capillary}$ is the diameter of the glass capillaries used.

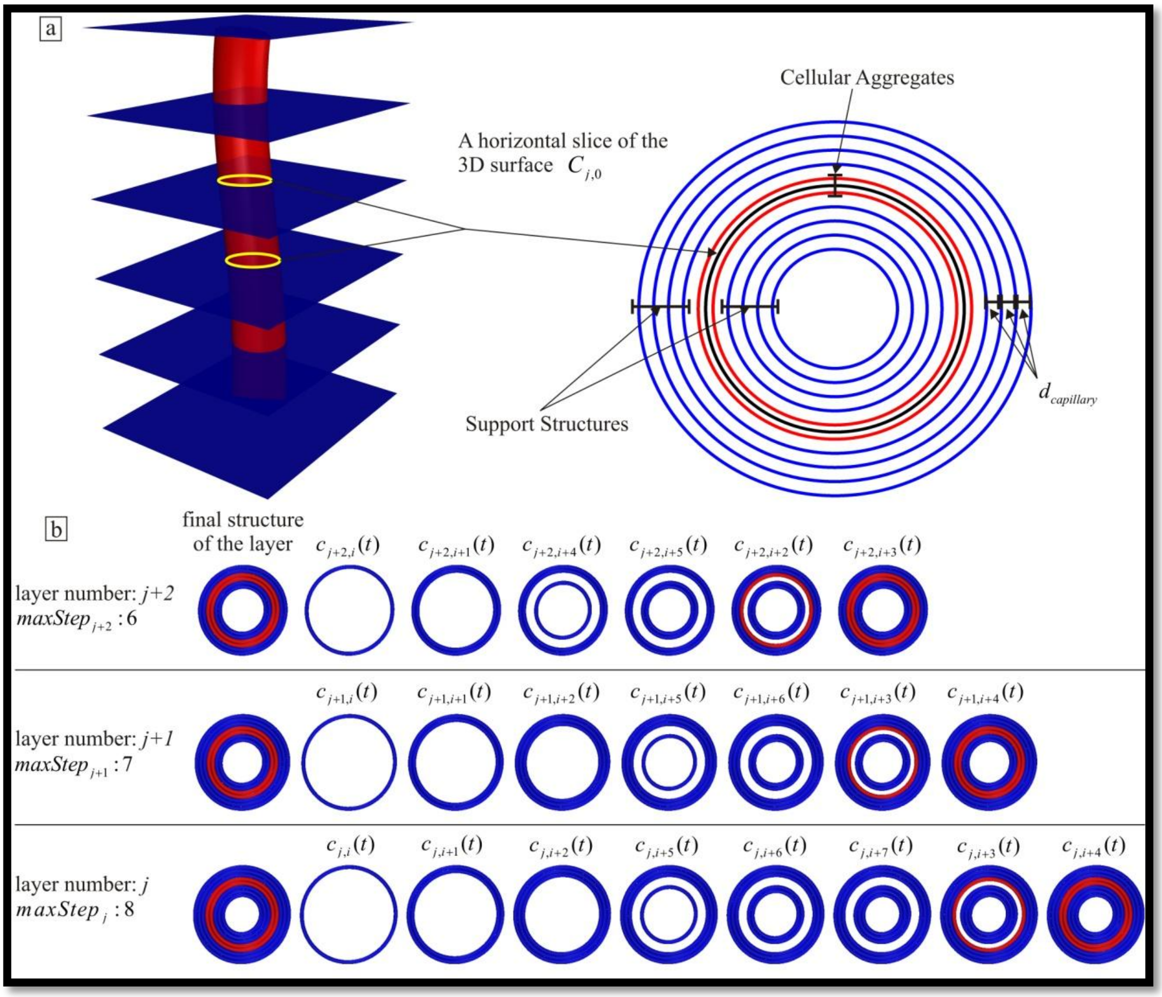

Figure 3.4. (a) The slicing process of a blood vessel and the placement of the support structures & cellular aggregates at j^{th} layer, (b) The bioprinting topology for three example consecutive layers, of both support structures and cellular aggregates.

The initial offset amount for a layer is strictly positive, resulting in exterior offset curves. However, the offset amount is dropped by the capillary diameter for each successive cylinder on that layer. Therefore, after ($maxStep_j/2$) cylinders, the offset amount will become negative resulting in interior offset curves as shown in Figure 3.4. Thus, cellular aggregates are supported by support structures from both inner and outer directions. As $c_{j,0}$(t) defines a contour curve of the surface on a given height and a curve parameter *t*, then the offset curves $c_{j,i}$(t) is calculated with offset amount $o_{j,i}$ as follows:

$$c_{j,i}\left(t\right)=c_{j,0}\left(t\right)+o_{j,i}\overrightarrow{N_{j,i}}\left(t\right)_{i=1..\max Step_j} \tag{3.2}$$

Where;

$\overrightarrow{N_{j,i}}\left(t\right)$ = unit normal vector on curve $c_{j,0}\left(t\right)$ at a parametric location t.

Two center cylinders on a layer (red ones in Figure 3.4), with respect the *maxStep*$_j$ value, is placed as cellular aggregates and the remaining cylinders as support structures in order to effectively mimic the original vessel dimensions and to provide better coverage of cells. Furthermore, at a layer, support structures are printed first, and then the cellular aggregates in order to prevent cell outflow and to preserve anatomically correct shape of the modeled vessel as shown in Figure 3.4. As the $o_{j,i}$'s for the j^{th} layer keeps decreasing by $d_{capillary}$ amount at each increment on *i*, support structures on a layer are printed from the outermost one to the innermost one as shown in Figure 3.4.

After appropriate sections of the cell composition and support structure are determined for each layer, the 3D bioprinting path plan for cell-biomaterial topology is calculated. Then, layer by layer, these cylindrical aggregates of the cell and gels will be printed accordingly a file that is generated by Algorithm 2 using a 3D bioprinter [4]. A cross sectional view of a smooth blood vessel (aorta) model and the surrounding support structures, which are generated by Algorithm 1 and 2, are shown in Figure 3.5. The finalized aorta model is composed of cellular aggregates and support structures that keep the cellular aggregates in its designed shape. Since the cylindrical cell aggregates are lacking in strength than the biomaterial, each layer is perfectly supported for stable cell aggregate printing [4].

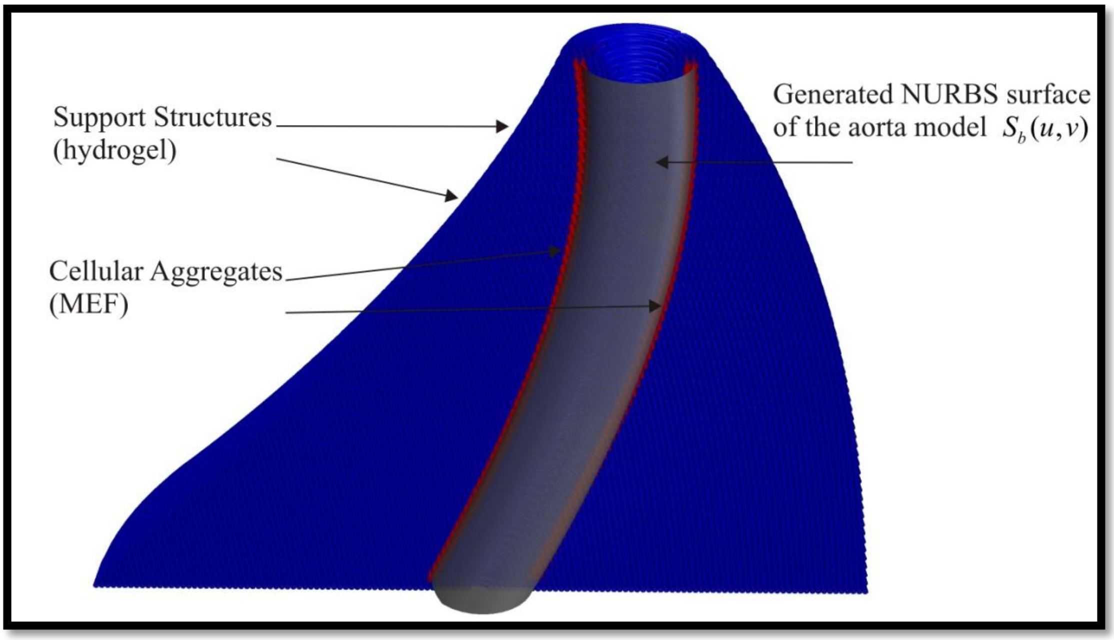

Figure 3.5. Representation of the ‘Self-Supporting’ model, with vessel (grey), cellular aggregates (red) and support structures (blue).

Algorithm 2 presents the calculation of the self-supporting structures and path planning for 3D bioprinting of both cellular aggregates and support structures. The algorithm takes the generated B-spline surface of vessel model and outputs a path plan for bioprinting of anatomically correct vessel model.

```
Algorithm 2.  Self-Supporting Structure Generation
Input:
$S_b(u,v)$:        generated NURBS Surface
$d_{capillary}$:    diameter of the glass capillaries
topSupport:     number of support cylinders on top layer (user input)
Output:
Finalized vascular model, with support structure
Path planning for 3D-Bioprinting (a compatible script file for the 3D-Bioprinter)
Start
Initialize totalLayers←(surfaceHeight/elevate)+1, j← 1, n← 1, i← 1, contourLevel ← 0
Initialize $maxStep_j$← topSupport+totalLayers-j
For ( j = 1 to totalLayers) {
  contourLevel←contourLevel + elevate
  If ($c_{j,0}$← contouring the surface from a given contourLevel, results in a closed curve) Then {
    Initialize $o_{j,i}$← ($maxStep_i$/2)×$d_{capillary}$
    For ( i = 1 to $maxStep_i$) {
        $c_{j,i}$← offset $c_{j,0}$ by $o_{j,i}$ using Equation (3.2)
        Initialize curveLength← length($c_{j,i}$)
        If (curveLength < minSegmentLength) Then {Exit For Loop}
```

```
            If ( $i$ = $maxStep_i$/2 or $i$ = $maxStep_i$/2+1) Then {
            Store $c_{j,i}$ and curveLength in the script file as cellular structure}
            Else{Store $c_{j,i}$ and curveLength in the script file as support structure}
            $o_{j,i}$←calculate $o_{j,i}$ using Equation (3.2)
    }
$maxStep_j$← topSupport + totalLayers – $j$ }
}
End
```

3.3 Zig-Zag Approach for Vertical Path Planning of Vascular Constructs

Similar to Self-Supporting path planning, a zig-zag based path planning is proposed for branched vascular constructs. After obtaining the freeform surface representation of the branched vascular constructs, the final step before the fabrication process is to create a bioprinting path plan for both cellular aggregates and support structures. An algorithm is developed to determine an optimum path-plan for bioprinting of branched structures such as coronary arteries. The proposed algorithm creates a zig-zag pattern path to anatomically mimic the shape of the vessel with cellular aggregates while providing support structure to conserve its form.

The proposed method starts with calculating an invisible bounding box, BB_0 = {bb_0, bb_1, bb_2, bb_3, bb_4, bb_5, bb_6, bb_7} of the generated smooth parametric surfaces $S_b(u,v)$. The edges of the bounding box lie parallel to X-Y-Z-axis. Then those points are offset with a predetermined amount in *x,y*-plane to enlarge planar area, which will be the domain for zig-zag shaped support structures that lie perpendicular to each other for each subsequent layer as shown in Figure 3.6. Then the layer number is determined by dividing the vertical surface length to predetermined distance (*interval*) between the parallel layers and *contourLevel* is set to the bottom plane of the box.

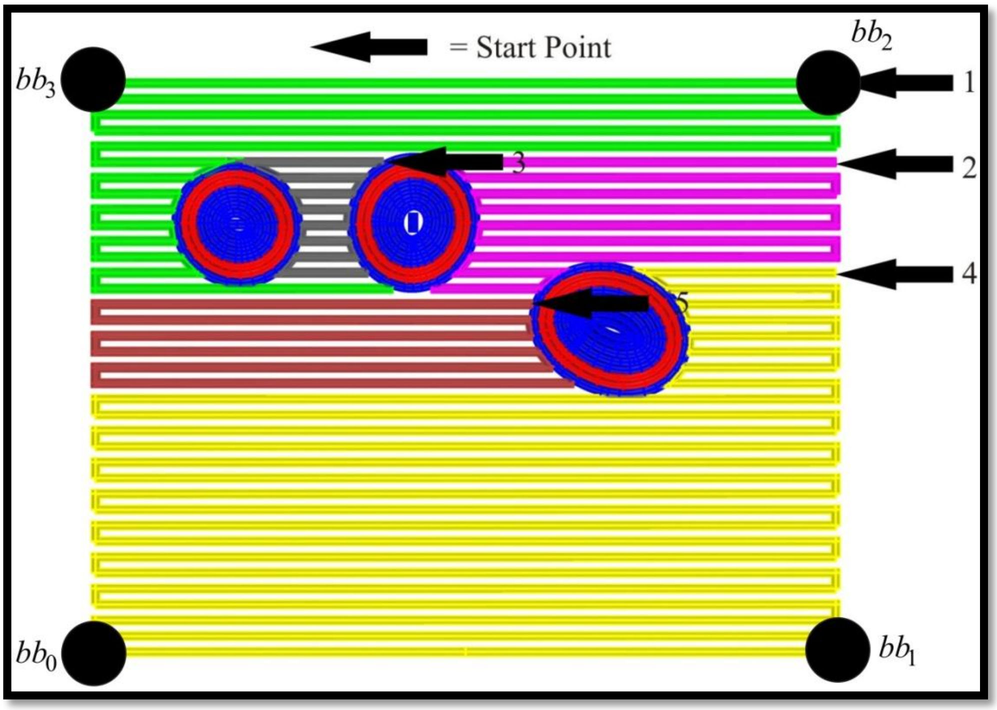

Figure 3.6. The printing order of outer support structures; first green, second magenta, third gray, fourth yellow curves and fifth brown curves.

For each layer, starting from the bottom level, *contourLevel* and duplicate vertices' *z*-coordinates of the enlarged bounding box are increased by *interval* amount to determine the level of the contour for that layer. For each layer *i*, the algorithm introduces at most *B* closed contour curves $CC_i = \{C_{i,b}\}_{b=1..B}$ from the smooth surfaces $S_b(u,v)$. And offset those closed curves with an amount of *offsetAmount* ← (*cellStripe* + *supportStripe / 2*) × $d_{capillary}$ with the following equation:

$$C_{i,b}^{offsetAmout}\left(t\right)=C_{i,b}\left(t\right)+offsetAmount\cdot\overrightarrow{N_{i,b}}\left(t\right)_{i=1..B} \tag{3.3}$$

Where;

$\overrightarrow{N_{i,b}}\left(t\right)$ = unit normal vector on curve $C_{i,b}\left(t\right)$ at a parametric location t

This offset operation results in at most *B* offset contour curves OC_i = $\{C_{i,b}^{offsetAmount}\}_{b=1..B}$. *cellStripe* variable defines the number of cylindrical cellular aggregates to satisfy desired wall thickness, and *supportStripe* variable defines the number of supportive cylinders to conserve cellular aggregates. The gap between OC_i's and CC_i's (part B's in Figure 3.7) implies that the cellular aggregates should be placed to the area between those curves.

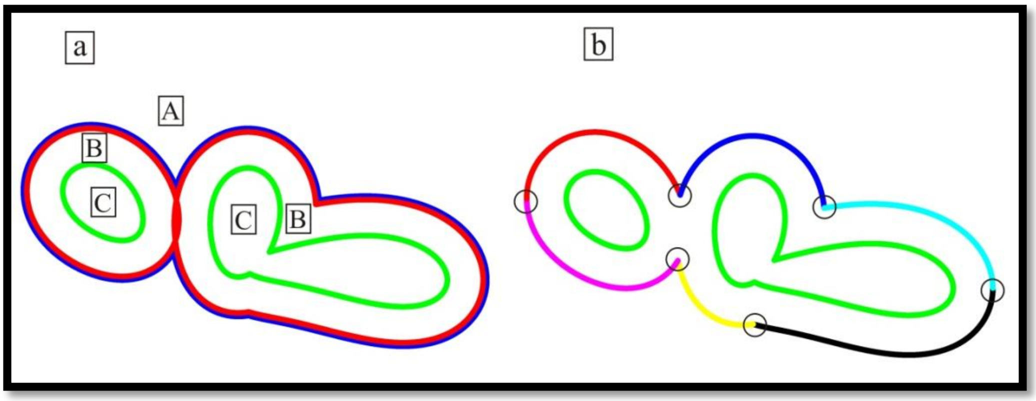

Figure 3.7. (a) Union operation of intersecting OC_i's (shown with red), results in blue curve. Support structure space (A&C), cellular aggregate space (B). (b) Splitting of an even layer OC_i's to its six successor curves (red-blue-cyan-black-yellow-magenta), from its deflection points and greatest & lowest x-coordinate points.

For even numbered layers; offset curves OC_i's are split from the deflection points where G^1 discontinuity occurs - sudden change points in direction of the unit tangent vectors of the respective curve - and min. & max. *x*-coordinate points resulting in new OC_i set (as shown in Figure 3.7(b)).

For odd numbered layers; offset curves OC_i's are split from the deflection points and min. & max. *y*-coordinate points resulting in new OC_i set. Along with the border lines that are placed around the domain of the enlarged bounding box (Figure 3.6), they constitute the layer curve set $LC_{i,k}$, for the respective layer *i*.

After layer curve sets $LC_{i,k}$'s are generated, they crossed with parallel lines (lying perpendicular to *x*,*z*-plane for even numbered, *y*,*z*-plane for odd numbered layers) *contourLevel*. The intersections of layer curves and those parallel lines results in *n* intersection points $P_i = \{P_{i,n}\}$. Each intersection point $P_{i,n}$ has three type of information, which layer curve $LC_{i,k}$ it belongs to $P_{i,n}^{curve\_id}$, which move made last $P_{i,n}^{previousMove}$ and its status $P_{i,n}^{info}$ (whether if its "*visited*", "*not visited*", or "*waiting*") to guide the algorithm to form non-intersecting and non-repeating support cylinders. Then P_i's are sorted in descending order with respect to *x*,*y*,*z*-coordinates for even numbered layers, and sorted in descending order with respect to *y*,*x*,*z*-coordinates for odd numbered layers.

After the point sorting process, the algorithm starts to generate support structures CO_i's, that cover the cellular aggregates from outside in a zig-zag fashion. Therefore, from P_i's, the algorithm starts a dynamic search from the lowest indexed "*not visited*" point for a feasible neighbor point and connect them to form polylines. Searching for a feasible neighbor point procedure for an intersection point $P_{i,n}$, for even numbered layers is as follows:

If $P_{i,n}^{previousMove}$ = 0 **Then** search for a "*non-visited"* point, shares same x-coordinate and a lower y-coordinate, pick the closest one and set its $P_{i,n+1}^{previousMove}$ = 1

If $P_{i,n}^{previousMove}$ = 1 **Then** search for a "*non-visited"* point, lies in the same curve, has a lower x-coordinate, pick the closest one and set its $P_{i,n+1}^{previousMove}$ = 2

If $P_{i,n}^{previousMove}$ = 2 **Then** search for a "*non-visited"* point, shares same x-coordinate and a greater y-coordinate, pick the closest one and set its $P_{i,n+1}^{previousMove}$ = 3

If $P_{i,n}^{previousMove}$ = 3 **Then** search for a "*non-visited"* point, lies in the same curve, has a lower x-coordinate, pick the closest one and set its $P_{i,n+1}^{previousMove}$ = 0

For odd numbered layers, a neighbor point for intersection point $P_{i,n}$ is searched using *previousMove* as follows:

If $P_{i,n}^{pre\ viousMove}$ = 0 **Then** search for a "*non-visited"* point, shares same y-coordinate and a lower x-coordinate, pick the closest one and set its $P_{i,n+1}^{previousMove}$ = 1

If $P_{i,n}^{previousMove}$ = 1 **Then** search for a "*non-visited"* point, lies in the same curve, has a lower y-coordinate, pick the closest one and set its $P_{i,n+1}^{previousMove}$ = 2

If $P_{i,n}^{previousMove}$ = 2 **Then** search for a "*non-visited"* point, shares same y-coordinate and a greater x-coordinate, pick the closest one and set its $P_{i,n+1}^{previousMove}$ = 3

If $P_{i,n}^{previousMove}$ = 3 **Then** search for a "*non-visited"* point, lies in the same curve, has a lower y-coordinate, pick the closest one and set its $P_{i,n+1}^{previousMove}$ = 0

Hence, the search process continues dynamically and a zigzag patterned support curves are generated for subsequent layers.

After that, contour curves CC_i's will be offset inwards and resulting curves to be union to form cellular aggregates CM_i, with respect to *cellStripe* value (the number of cylinders to meet the desired wall thickness). If the offset curves form the inside support curves CI_i until minimum segment length is reached to prevent any self-intersection.

After path planning and topology optimization is finalized, first the support structures and then the cellular aggregates are bioprinted at each layer. Figure 3.8 shows branching steps of the coronary artery model with three consecutive layers, along with their contour curves (shown with blue curves at the bottom part); offset curves (shown with black curves at the bottom part); and the deflection points (red circles at the bottom part) for each layer.

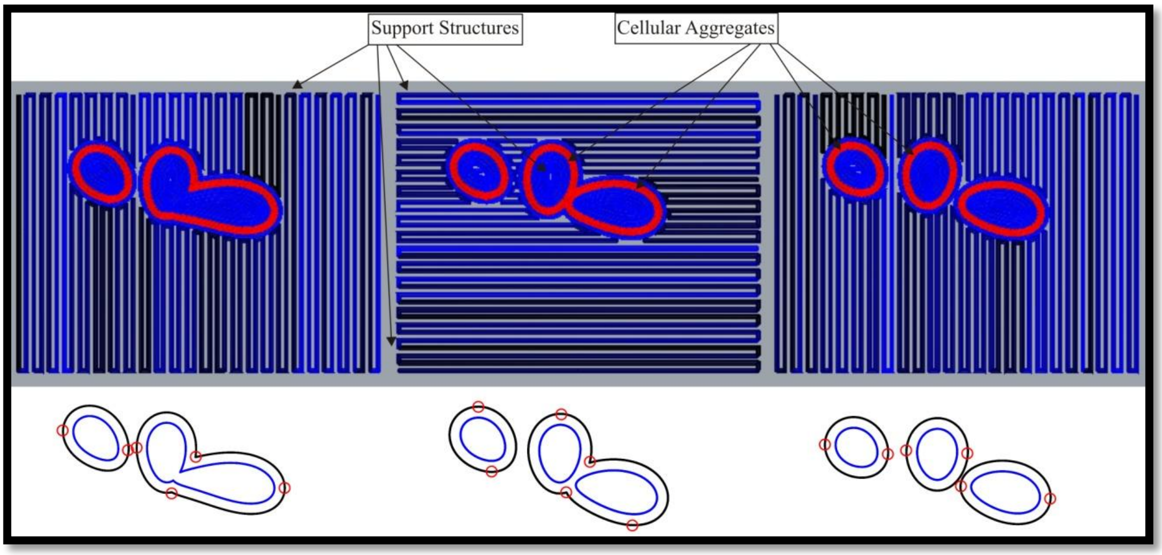

Figure 3.8. Example layers of a coronary artery, showing the joint locations of the branched vessel model (Figure 2.5).

For this method, the pseudo-code of Algorithm 3 is given below.

<u>Algorithm 3.</u> Zig-Zag Support Structure Generation

Input:

$S_b(u,v)$: generated NURBS Surface(s)

$d_{capillary}$: diameter of the glass capillaries

$v_{capillary}$: max. volume of the glass capillaries

cellStripe: the number of cylindrical cellular aggregates to satisfy desired wall thickness

supportStripe: the number of supportive cylinders to conserve cellular aggregates
interval: distance between planar layers
gap: distance between subsequent linear support cylinders
enlarge: enlargement amount of the cutting plane

Output:

Finalized vascular model, with support structure
Path planning for 3D-Bioprinting (a compatible script file for the 3D-Bioprinter)

```
Start
$\{bb_0, bb_1, bb_2, bb_3, bb_4, bb_5, bb_6, bb_7\} \leftarrow \text{BoundingBox}(S_b(u,v))$
$lb \leftarrow bb_0,\ rb \leftarrow bb_1,\ rt \leftarrow bb_2,\ lt \leftarrow bb_3$
$lb' \leftarrow (bb_0(x) - enlarge,\ bb_0(y) - enlarge,\ bb_0(z)),\ rb' \leftarrow (bb_1(x) + enlarge,\ bb_1(y) - enlarge,\ bb_1(z))$
$rt' \leftarrow (bb_2(x) + enlarge,\ bb_2(y) + enlarge,\ bb_2(z)),\ lt' \leftarrow (bb_3(x) - enlarge,\ bb_3(y) + enlarge,\ bb_3(z))$
Initialize $totalLayers \leftarrow floor(distance(bb_3,bb_0)\ /\ interval)+1,\ j \leftarrow 1,\ n \leftarrow 1,\ i \leftarrow 1$
Initialize $contourLevel \leftarrow plane(\{lb', rb', rt', lt'\})$ = “0”
For ( $i$ = 1 to $totalLayers$) {
  $contourLevel \leftarrow contourLevel + interval$
  $lb'(z) \leftarrow lb'(z) + interval,\ rb'(z) \leftarrow rb'(z) + interval$
  $rt'(z) \leftarrow rt'(z) + interval,\ lt'(z) \leftarrow lt'(z) + interval$
  $offsetAmount \leftarrow (cellStripe + supportStripe\ /\ 2) \times d_{capillary}$
  For ( all $S_b(u,v)$) {
    $C_{i,b} \leftarrow contour(S_b(u,v))_{contourLevel}$   ‘// contour curves at respective contour level
    $C^o_{i,b} \leftarrow offset(C_{i,b})_{offsetAmount}$   ‘// offset curves with respective offset amount
    $CC_i = \{C_{i,b}\} \leftarrow$ a set of $b$ closed contour curves that $i^{th}$ layer contains
    $OC_i \leftarrow OC_i \cup \{C^o_{i,b}\}$
  }
  If ( $i$ == EVEN) Then {
    $OC_i \leftarrow split(OC_i)_{intersection,minX,maxX}$
    $line1 \leftarrow addLine(\{lb', rb'\})$
    $line2 \leftarrow addLine(\{lt', rt'\})$
    $LC_{i,k} \leftarrow OC_i \cup line1 \cup line2$
    $l \leftarrow lb'$
    For ( $j = lt'(x)$ to $rt'(z)$) Step $gap$ {
      $line \leftarrow addLine(\{j,l\})$
      For ( all $LC_{i,k}$) {
        If ( $LC_{i,k} \cap line \neq$ NULL) Then {
          $P_{i,n} \leftarrow LC_{i,k} \cap line$
          $P^{info}_{i,n} \leftarrow$ “not visited” = 0
          $P^{curve\_id}_{i,n} \leftarrow k$
          $P_i \leftarrow P_i \cup \{P_{i,n}\}$
        }
      }
      $l(x) \leftarrow l(x) + gap$
    }
    $P_i \leftarrow sortPoints(P_i)_{x,y,z}$
    $curveLength \leftarrow 0$
    $polyLine \leftarrow$ NULL
    While ( $min(P_i^{info})$ == 0) {
      If ($P_{i,n} \leftarrow polyLine^{endpoint}$); If Not {
      $P_{i,n} \leftarrow$ get the $minimum\_indexed$ $P_{i,n}^{info}$ == 2 = "waiting" point; If Not {
      $P_{i,n} \leftarrow$ get the $minimum\_indexed$ $P_{i,n}^{info}$ == 0 = "not visited" point }
      $P_{i,n+1} \leftarrow$ find the best appropriate neighbor point according to previousMove; If Not {
        $CO_i \leftarrow CO_i \cup polyLine$
        $P_{i,n}^{info} \leftarrow$ "visited" = 1
        $polyLine \leftarrow$ NULL
        return_to_start_of_the_loop }
        $line \leftarrow addLine(\{P_{i,n}, P_{i,n+1}\})$
      If ( $polyLine^{length} + line^{length} \leq v_{capillary}$) Then {
        $polyLine \leftarrow polyLine \cup line$
        $P_{i,n}^{info} \leftarrow$ "visited" = 1
      }
      Else {
        $P_{i,n}^{info} \leftarrow$ "waiting" = 2
      }
    }
  }
  Else {
    $OC_i \leftarrow split(OC_i)_{intersection,minY,maxY}$
    $line1 \leftarrow addLine(\{lt', lb'\})$
    $line2 \leftarrow addLine(\{rt', rb'\})$
    $LC_{i,k} \leftarrow OC_i \cup line1 \cup line2$
    $l \leftarrow lb'$
    For ( $j = lb'(y)$ to $lt'(y)$) Step $gap$ {
      $line \leftarrow addLine(\{j,l\})$
      For ( all $LC_{i,k}$) {
        If ( $LC_{i,k} \cap line \neq$ NULL) Then {
          $P_{i,n} \leftarrow LC_{i,k} \cap line$
          $P_{i,n}^{info} \leftarrow$ "not visited" = 0
          $P_{i,n}^{curve\_id} \leftarrow k$
          $P_i \leftarrow P_i \cup \{P_{i,n}\}$
        }
      }
      $l(y) \leftarrow l(y) + gap$
    }
    $P_i \leftarrow sortPoints(P_i)_{y,x,z}$
    $curveLength \leftarrow 0$
    $polyLine \leftarrow$ NULL
    While ( $min(P_i^{info})$ == 0) {
      If ($P_{i,n} \leftarrow polyLine^{endpoint}$); If Not {
      $P_{i,n} \leftarrow$ get the $minimum\_indexed$ $P_{i,n}^{info}$ == 2 = "waiting" point; If Not {
      $P_{i,n} \leftarrow$ get the $minimum\_indexed$ $P_{i,n}^{info}$ == 0 = "not visited" point }
      $P_{i,n+1}$ ← find the best appropriate neighbor point according to *previousMove;* **If**
**Not** {
      $CO_i \leftarrow CO_i \cup polyLine$
      $P_{i,n}^{info}$ ← "*visited*" = 1
        *polyLine* ← NULL
        return_to_start_of_the_loop }
        $line \leftarrow addLine(\{P_{i,n}, P_{i,n+1}\})$
      **If** ( $polyLine^{length} + line^{length} \leq v_{capillary}$) **Then** {
        $polyLine \leftarrow polyLine \cup line$
        $P_{i,n}^{info}$ ← "*visited*" = 1
      }
      **Else** {
        $P_{i,n}^{info}$ ← "*waiting*" = 2
      }
    }
  }
  **For** ( all $CC_i$) { $CC_{i,b}^{check} \leftarrow 0$ }
  $offsetAmount \leftarrow (cellStripe + supportStripe - 1/2) \times d_{capillary}$
  **While** ( $min(CC_i^{check}) == 0$ ) {
    **For** ( $b$ = 1 to $B$) {
      **If** ( $CC_{i,b}^{check} \leftarrow 0$ ) **Then** {
       $crv \leftarrow offset(C_{i,b})_{\text{offsetAmount}}$
       **If** ( $crv^{length} \leq min\_Segment\_Length$ ) **Then** { $CC_{i,b}^{check} \leftarrow 1$ }
       **If** ( $0 \leq offsetAmount < (cellStripe + supportStripe - 1/2) \times d_{capillary}$) **Then** {
        $CM_i \leftarrow CM_i \cup crv$ }
       **Else** { $CI_i \leftarrow CI_i \cup crv$ }
      }
    }
    $offsetAmount \leftarrow offsetAmount - d_{capillary}$
  }
Send_to_Bioprinter( { $CO_i$, $CI_i$, $CM_i$} )
}
**End**
```

3.4 Transforming Biomodeled Smooth Parametric Surfaces to a Vertical Form

The proposed Self-Support and Zig-Zag methods presented in Section 3.2 and 3.3 could result in excessive use of support material and cells, as well as increase in the total printing time which could negatively affect cell viability. Since the printed tissue constructs are flexible in nature, they can be twisted to a degree and keep the desired shape. Therefore, we designed a new surface representation that elongates vertically in *z*-direction by preserving the original lengths and branching topology of the vessel.

The main idea here is to convert the components of the trajectory curve set TC$_i$ = $\{TC_i\}_{i=1..I}$ to linear form and build smooth surfaces along those curve $S_i(u,v)$, with respect to each of their median radius $\mathrm{TC}_i^{median\_rad}$ values. Each trajectory curve, other that the root (*TC*$_1$), has a branching part $\mathrm{TC}_i^{branc\,hLine}$ (to get separated from the direction of its parent curve) and a linear part TC_i^{rest} , if $\mathrm{TC}_i^{lengt\,h}$ is greater than *branchLine*, as shown in Figure 3.9. *minDistance* (the minimum linear distance between two parallel branches) and *branchingAngle* (the separation angle of two branches) values are determined by the user and those two parameters, together with the *maxRadius* (the maximum median radius of the trajectory curves other than the root) value, determines the length of the *branchLine* by:

(3.4)

$$minDistance + 2 \times maxRadius > 2 \times branchLine \times Cos(\Pi/2 - (branchingAngle\ /\ 2))$$

For the trajectory curves whose length $\mathrm{TC}_i^{lengt\,h}$ is shorter than *branchLine*, their length is adjusted to *branchLine* so that the minimum distance between the surfaces is satisfied and any possible intersection of the branches is avoided. Hence, no other trajectory curves median radius value $\{TC_i^{median\_rad}\}_{i=2..I}$ is greater than *maxRadius* as shown in Figure 3.10(a).

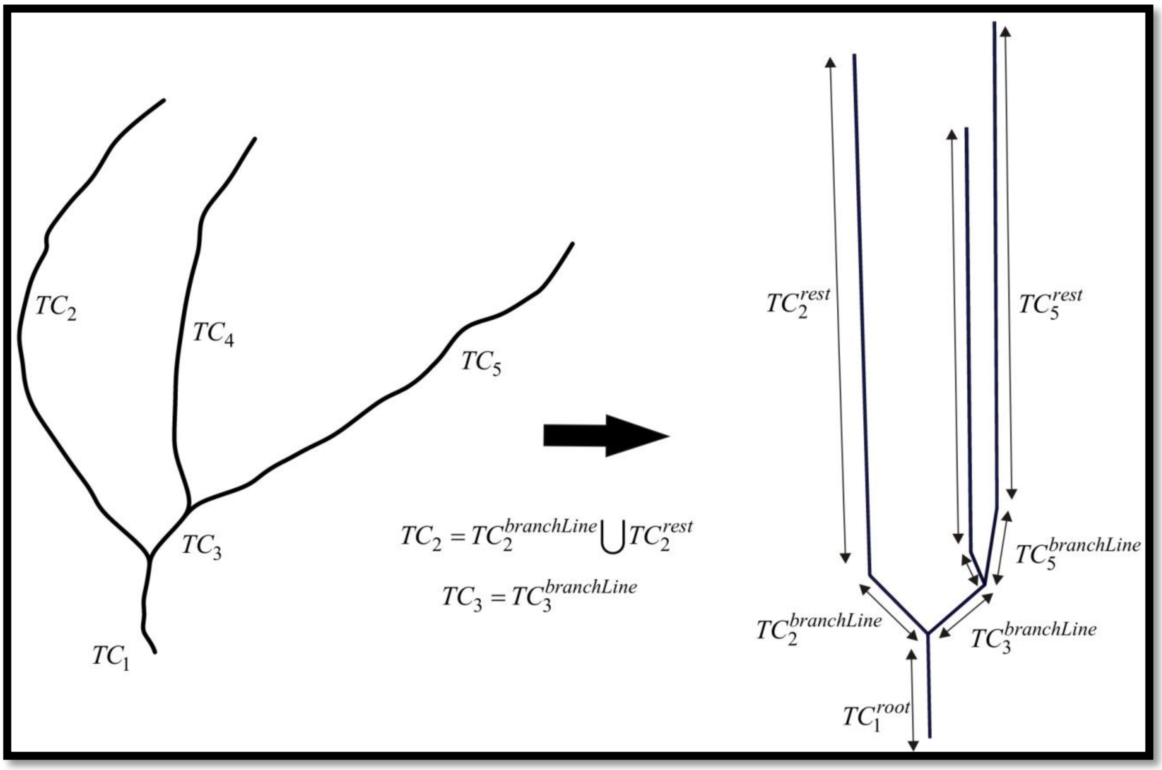

Figure 3.9. Converting the biomodeled trajectory curves to a vertical pattern, with respect to each trajectory curves original lengths.

First, the root (TC_1) curve approximates a linear centerline curve $line_1$, lies in z-plane with its length equal to$\mathrm{TC}_1^{lengt\ h}$, then the surface $S_1(u,v)$ along $line_1$ with respect to its radius $TC_1^{median\ \_rad}$ is fitted with the following equation:

$$S_1(u,v) = line_1(u) + M(u)T(v) \tag{3.5}$$

$$T(v) = \left(TC_1^{median\_rad}\cos(v), TC_1^{median\_rad}\sin(v)\right)$$

$$0 \le u \le 1 \text{ and } 0 \le v \le 1$$

Where $M(u)$ is a 3x3 matrix incorporating rotation and nonuniform scaling of T(v) as a function of u.

After that, for each trajectory curve $\{TC_i\}_{i=2..I}$, *branchLine* ($\mathrm{TC}_i^{branc\ hLine}$) and *rest* ($\mathrm{TC}_i^{rest}$) parts are generated vertically with respect to the respective trajectory curves' length and *branchingAngle*, as shown in Figure 3.9. The key point of this centerline curve generation is that each parent-child pair lies in perpendicular planes, therefore any possible intersection between the branches are eliminated. Lastly, the smooth surface(s) $S_i(u,v)$ are built along centerline curve(s) with respect to its median radius $TC_i^{median\_rad}$, as shown in Figure 3.10(b) showing a branched vessel model with three branches and five trajectory curves.

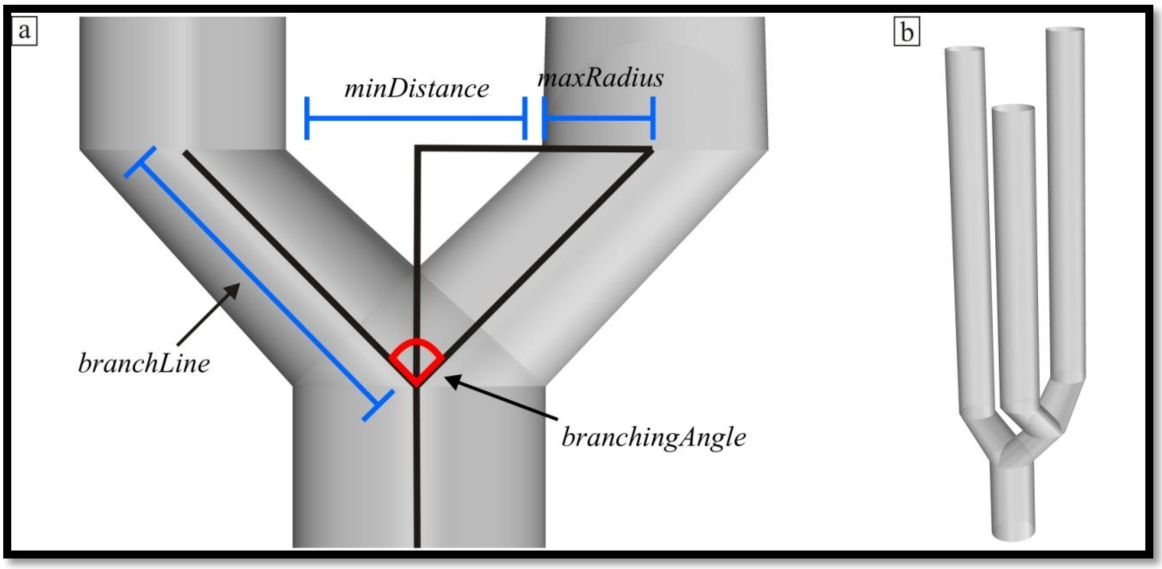

Figure 3.10. (a) The organization of the branching line with respect to the branching angle, minimum distance between the branches and maximum branch radius; vertical surface representation of a coronary artery, (b) coronary artery model with three branches and five trajectory curves.

The pseudo-code for Algorithm 4 is given below.

Algorithm 4. Smooth Parametric Surface Generation for Vertical 3D-Bioprinting

Input:

TC:	trajectory curve array
minDistance:	the minimum planar distance between parallel branch surfaces
branchingAngle:	the planar angle between all branch pairs
pS:	the predefined spot for starting point of trajectory curves & smooth surfaces
branchLine:	the minimum length of a branching line
increment:	increment amount for the length of branching lines, in case they fall short

Output:

$S_i(u,v)$:	generated NURBS Surface(s)

```
Start
Initialize $maxBranches \leftarrow 2$, $i \leftarrow 1$, $constructionPlane \leftarrow (x,y)$, $t \leftarrow 1$
$maxRadius \leftarrow max(TC_i^{median\_rad})_{i = 2..I}$
While ( $minDistance + maxBranches \times maxRadius > maxBranches \times branchLine \times Cos(\Pi/2 - (branchingAngle / 2))$ ) {
  $branchLine \leftarrow branchLine + increment$
}
```

```
$destination_i \leftarrow (pS\ (x), pS\ (y), pS\ (z) + \mathrm{TC}_i^{lengt\ h})$
$line_i \leftarrow addLine(\{pS, destination_i\})$
$circle_i^1 \leftarrow addCircle(line_i^{startPoint}, TC_i^{median\ \_rad})_{constructionPlane}$
$circle_i^2 \leftarrow addCircle(line_i^{endPoint}, TC_i^{median\ \_rad})_{constructionPlane}$
$TC_i^{surface} \leftarrow loftSurface(\{circle_i^1, line_i, circle_i^2\})$
$\mathrm{S}_i(u,v) \leftarrow TC_i^{surface}$
$(\mathrm{TC}_i^{child,t})^{startPoint} \leftarrow line_i^{endPoint}, (\mathrm{TC}_i^{child,t+1})^{startPoint} \leftarrow line_i^{endPoint}$
$(\mathrm{TC}_i^{child,t})^{startCircle} \leftarrow circle_i^2, (\mathrm{TC}_i^{child,t+1})^{startCircle} \leftarrow circle_i^2$
$(\mathrm{TC}_i^{child,t})^{plane} \leftarrow$ "1" $= (x,z), (\mathrm{TC}_i^{child,t+1})^{plane} \leftarrow$ "1" $= (x,z)$
$(\mathrm{TC}_i^{child,t})^{angle} \leftarrow (\Pi + branchingAngle / 2), (\mathrm{TC}_i^{child,t+1})^{angle} \leftarrow (\Pi - branchingAngle / 2)$
**For** ( $i = 1$ to $I$) {
  **If** ($\mathrm{TC}_i^{parent} \neq \mathrm{NULL}$) **Then** {
    **If** ($\mathrm{TC}_i^{plane} ==$ "1" $= (x,z)$) **Then** {
      **If** ($\mathrm{TC}_i^{angle} < \Pi$) **Then** {$destination_i \leftarrow (\mathrm{TC}_i^{startPoint}\ (x) + (branchingAngle / 2) \times branchLine, \mathrm{TC}_i^{startPoint}\ (y), \mathrm{TC}_i^{startPoint}\ (z) + Sin(\Pi/2 - (branchingAngle / 2)) \times branchLine)$}
      **If** ($\mathrm{TC}_i^{angle} > \Pi$) **Then** {$destination_i \leftarrow (\mathrm{TC}_i^{startPoint}\ (x) - (branchingAngle / 2) \times branchLine, \mathrm{TC}_i^{startPoint}\ (y), \mathrm{TC}_i^{startPoint}\ (z) + Sin(\Pi/2 - (branchingAngle / 2)) \times branchLine)$}
      $line_i \leftarrow addLine(\{pS, destination_i\})$
      $circle_i^1 \leftarrow addCircle(line_i^{endPoint}, TC_i^{median\ \_rad})_{constructionPlane}$
    }
    **Else** {
      **If** ( $\mathrm{TC}_i^{angle} < \Pi$) **Then** {$destination_i \leftarrow ( \mathrm{TC}_i^{startPoint}\ (x), \mathrm{TC}_i^{startPoint}\ (y) + (branchingAngle / 2) \times branchLine, \mathrm{TC}_i^{startPoint}\ (z) + Sin(\Pi/2 - (branchingAngle / 2)) \times branchLine)$}
      **If** ( $\mathrm{TC}_i^{angle} > \Pi$) **Then** {$destination_i \leftarrow ( \mathrm{TC}_i^{startPoint}\ (x), \mathrm{TC}_i^{startPoint}\ (y) - (branchingAngle / 2) \times branchLine, \mathrm{TC}_i^{startPoint}\ (z) + Sin(\Pi/2 - (branchingAngle / 2)) \times branchLine)$}
      $line_i \leftarrow addLine(\{pS, destination_i\})$
      $circle_i^1 \leftarrow addCircle(line_i^{endPoint}, TC_i^{median\ \_rad})_{constructionPlane}$
    }
    **If** ( $\mathrm{TC}_i^{lengt\ h} \leq branchLine$ ) **Then** {
      $TC_i^{surface} \leftarrow loftSurface(\{\mathrm{TC}_i^{startCircle}, line_i, circle_i^1\})$
      $\mathrm{S}_i(u,v) \leftarrow TC_i^{surface}$
    }
    **Else** {
      $TC_i^{surface\ ,t} \leftarrow loftSurface(\{\mathrm{TC}_i^{startCircle}, line_i, circle_i^1\})$
      $destination_i^2 \leftarrow ((line_i)^{endPoint}(x), (line_i)^{endPoint}(y), (line_i)^{endPoint}(z) + \mathrm{TC}_i^{lengt\ h} - branchLine)$
      $line_i^2 \leftarrow addLine(\{(line_i)^{endPoint}, destination_i^2\})$
      $circle_i^2 \leftarrow addCircle(line_i^{endPoint}, TC_i^{median\ \_rad})_{constructionPlane}$
      $TC_i^{surface,t+1} \leftarrow loftSurface(\{circle_i^1, line_i^2, circle_i^2\})$
      $S_i(u,v) \leftarrow TC_i^{surface,t} \cup TC_i^{surface,t+1}$
    }
  }
}
End
```

3.5 Path planning with Self-Supporting Method for Branched Vascular Constructs

After obtaining the vertically biomodeled smooth parametric surfaces with Algorithm 4, path planning and topology optimization needs to be carried out to 3D bioprint the model. In this part, a novel method is proposed for 3D bioprinting path planning which is also capable of printing branched structures. The main idea of self-supporting method for branched vascular constructs is to use least amount of material to cover and support the cellular aggregates. Hence, the method also optimizes the duration of the printing process to improve the cell viability.

The support structures are printed in a circular form in this approach, therefore the curves that are generated by contouring the whole surface representation at each layer *k*, $CC_k = \{C_{k,n}\}_{k=1..K}$ need to be offset inwards and outwards to form the support wall. Because of weak mechanical properties of hydrogel support material, they cannot keep their 3D form if printed on top of each other. Therefore, each support piece at each layer *k*+1 needs to be deposited on to the valleys of the support material at the preceding layer *k*. For each layer, Algorithm 5 computes the border curves $BC_k = \{BC_{k,n}\}_{n=1..N}$ and total number of cylindrical support structure and bio-ink pieces ($maxStep_k$) for that layer. Border curves represent the largest boundaries of outward support structure for that layer. As can be seen in Figure 3.11, the border curves of the $k+1^{th}$ layer (BC_{k+1}) are shown with black curves. To determine the border curves for the k^{th} layer, the contour curves CC_k are iteratively offset outwards with the capillary diameter $d_{capillary}$ increments in the offset amount (blue curves in Figure 3.11), once offset curves are large enough to strictly enclose the border curves of the $k+1^{th}$ layer, then they set as the border curves of their layer. Moreover, the total number of increments in offset amount to generate offset curves will be the total number of cylindrical support structure and the bio-ink pieces ($maxStep_k$) for that layer, as shown with blue curves in Figure 3.11.

To determine the border curves and the total number of cylindrical support structures and bio-ink pieces for each layer, a top-down approach needs to be developed. Therefore, the algorithm uses the user defined variable for determining the border curves and number of support cylinders for the top layer. Here, *topSupport* is a user variable for the number of support cylindrical pieces to enclose each *n* contour curve $C_{K,n}$ at the top layer. From that information, $maxStep_K$ will be equal to *topSupport* and the border curves for that layer can be found:

$$offsetAmount \leftarrow (topSupport\,/\,2) \times d_{capillary} \tag{3.6}$$

$$C_{K,n}^{offsetAmout}(t) = C_{K,n}(t) + offsetAmount \cdot \overrightarrow{N_{K,n}}(t)_{n=1..N}$$

Where;

$\overrightarrow{N_{K,n}}(t)$ = unit normal vector on curve $C_{K,n}(t)$ at a parametric location t

$BC_K = \{C_{K,n}^{offsetAmount}\ \}$

The algorithm then starts iterating downwards through the layers and determines the border curves and the total number of cylindrical support structures and bio-ink pieces for each layer, with the methodology explained above.

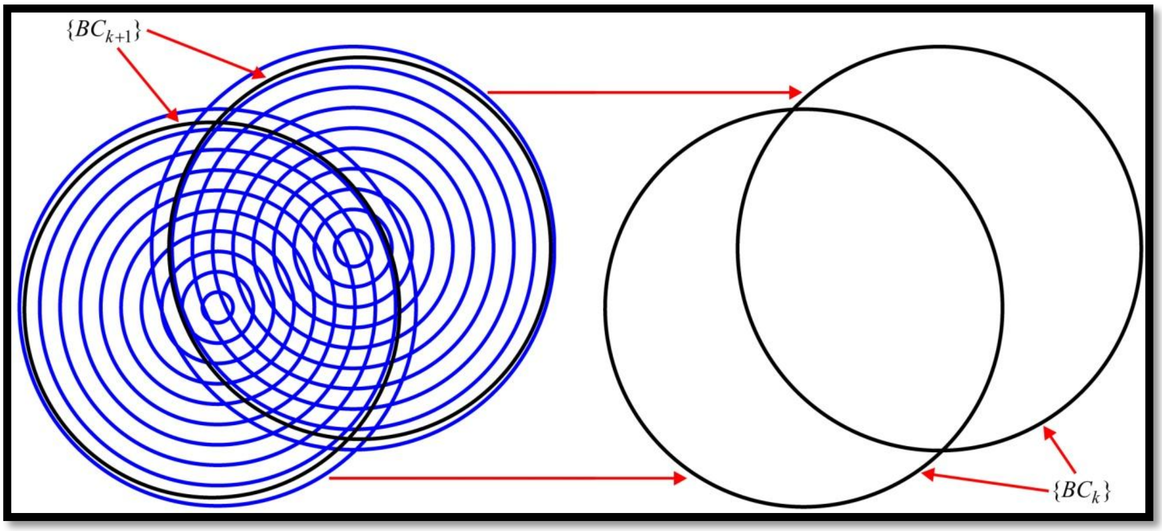

Figure 3.11. Determination of border curves for layer *k*, using the border curve information of layer *k+1*.

Once the total number of cylindrical support structures and bio-ink pieces are determined for each layer, from bottom to top layer, contour curves CC_k are offset with sequentially decreasing offset amounts (*o*) with $maxStep_k$ times resulting in offset curves $C^{o}_{k,n}$. According to the *cellStripe* value (the number of bio-ink cylinders to satisfy the wall thickness of the vessel) and the distance between the offset curves and the contour curves, the types of the offset curves $C^{o}_{k,n}$ are determined, whether they are outwards or inwards support cylinders {CO_k, CI_k} (support structures), or bio-ink cylinders middle {CM_k} as shown in Figure 3.12. As shown in Figure 3.12, the contour curves (shown with black curves) represents the inner boundaries of the biomodeled vessel, the closest *cellStripe* number (2 in this specific example) of cylindrical curves in outward direction represents the bio-ink cylinders middle (shown with red curves). The rest of the cylinders are grouped as support cylinders outwards and inwards (shown with blue) according to their orientation based on their corresponding contour curves. Every cylinder curve in each of the three sets {CO_k, CI_k, CM_k}, starting from the broadest curve for each set, are then linked together if they can connect each other with a line segment (Figure 3.12), if that line segment satisfies the following conditions:

- The line segment must be linear.
- The line segment must not intersect any of the other curves belonging to any of these sets.
- The line segments' lengths must be at most equals to $d_{capillary}$.

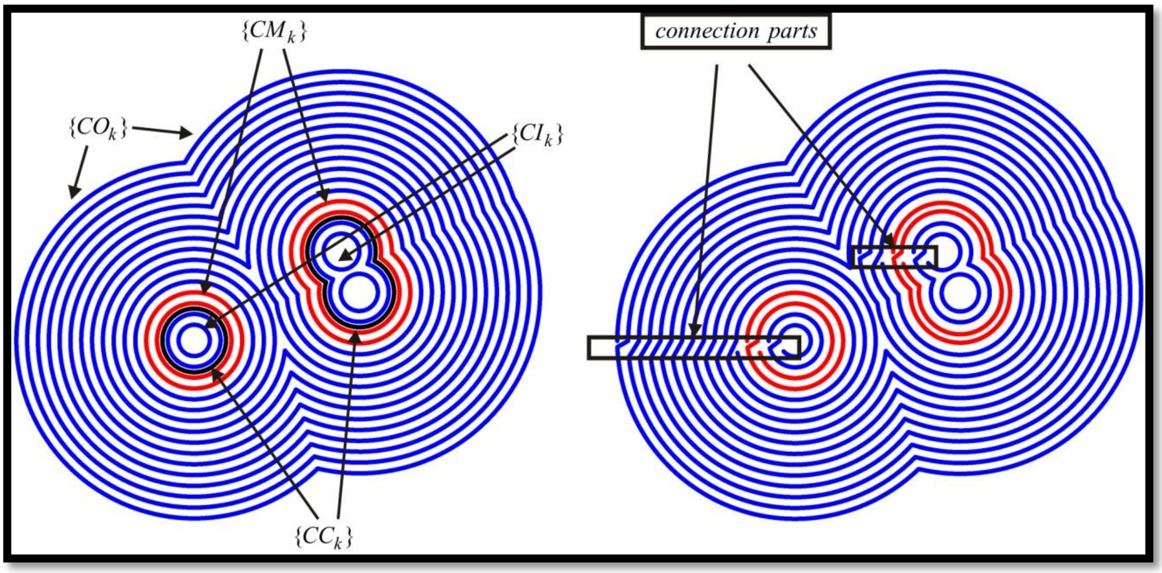

Figure 3.12. Linking procedure for supportive cylinders inwards, bio-ink cylinders middle and supportive cylinders outwards.

At a layer, the support cylinders outwards and inwards are printed first, and then the bio-ink aggregates are deposited along the valleys of the support structure. A representative path planning example of a three branched, eight trajectory curved surface model is shown in Figure 3.13.

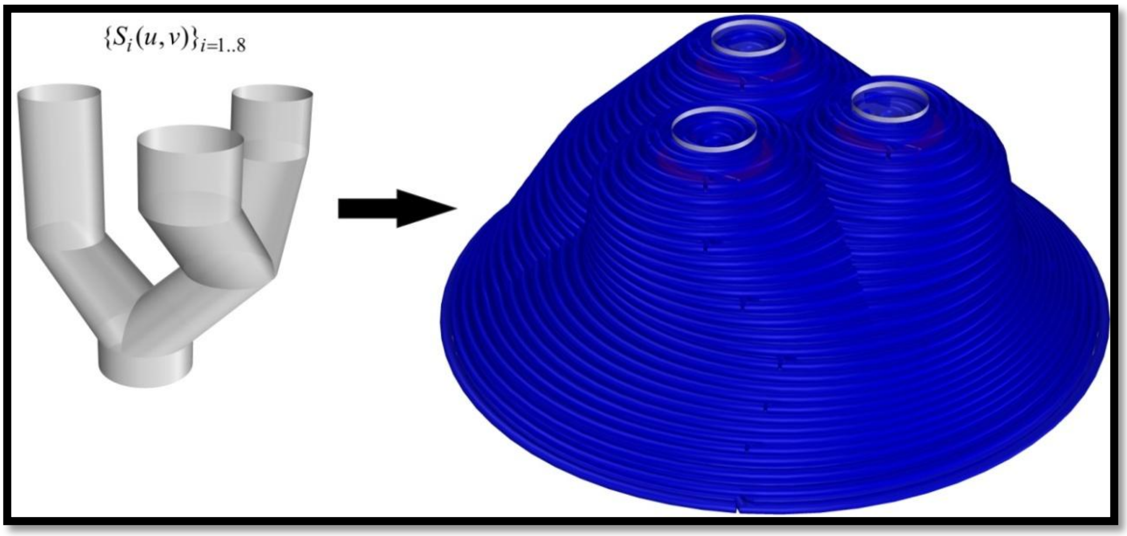

Figure 3.13. A representative path planning example of a three branched, eight trajectory curved surface model.

The pseudo-code of the Algorithm 5 is given below.

Algorithm 5. Vertical 3D-Bioprinting Self-Supporting Model

Input:

$S_i(u,v)$: generated NURBS Surface(s)

$d_{capillary}$: diameter of the glass capillaries

topSupport: the number of supportive cylinders at top layer

$v_{capillary}$: max. volume of the glass capillaries

cellStripe: the number of cylindrical cellular aggregates to satisfy desired wall thickness

interval: distance between layers

Output:

Finalized vascular model, with support structure

Path planning for 3D-Bioprinting (a compatible script file for the 3D-Bioprinter)

```
Start
{ $bb_0, bb_1, bb_2, bb_3, bb_4, bb_5, bb_6, bb_7$} $\leftarrow$ BoundingBox($S_b(u,v)$)
$lb \leftarrow bb_4$, $rb \leftarrow bb_5$, $rt \leftarrow bb_6$, $lt \leftarrow bb_7$
Initialize $totalLayers \leftarrow floor(distance(bb_3,bb_0)\ /\ interval)+1$, $j \leftarrow 1$, $k \leftarrow totalLayers$, $i \leftarrow 1$
Initialize $contourLevel \leftarrow plane(\{lb, rb, rt, lt\})$ = "$k$"
$contourLevel \leftarrow contourLevel - interval$
$lb(z) \leftarrow lb(z) - interval$, $rb(z) \leftarrow rb(z) - interval$, $rt(z) \leftarrow rt(z) - interval$, $lt(z) \leftarrow lt(z) - interval$
$offsetAmount \leftarrow (topSupport\ /\ 2) \times d_{capillary}$
For ( all $S_i(u,v)$) {
  $C_{k,n} \leftarrow contour(S_i(u,v))_{contourLevel}$
  $C^{o}_{k,n} \leftarrow offset(C_{k,n})_{\mathrm{offsetAmount}}$
  $CC_k = \{C_{k,n}\}$ ← a set of $n$ closed contour curves that $k^{th}$ layer contains
  $OC_k \leftarrow OC_k \cup \{C^{o}_{k,n}\}$
  $BC_k \leftarrow OC_k$
  $maxStep_k \leftarrow topSupport$
}
For ( $k$ = 1 to $K$) {
  $contourLevel \leftarrow contourLevel - interval$
  $lb(z) \leftarrow lb(z) - interval, rb(z) \leftarrow rb(z) - interval$
  $rt(z) \leftarrow rt(z) - interval, lt(z) \leftarrow lt(z) - interval$
  $j \leftarrow K - k$
  $BC^{j}_{j+1} \leftarrow transport(BC_{j+1})_{contourLevel}$
  $maxStep_j \leftarrow 1$
  $offsetAmount \leftarrow (maxStep_j / 2) \times d_{capillary}$
  For ( all $S_i(u,v)$) {
    $C_{j,n} \leftarrow contour(S_i(u,v))_{contourLevel}$
    $C^{o}_{j,n} \leftarrow offset(C_{j,n})_{\text{offsetAmount}}$
    $CC_j = \{C_{j,n}\}$ ← a set of $n$ closed contour curves that $j^{th}$ layer contains
    $OC_j \leftarrow OC_j \cup \{C^{o}_{j,n}\}$
  }
  $BC_j \leftarrow OC_i$
  While ($BC^{j}_{j+1} \supseteq BC_j$) {
     $maxStep_j \leftarrow maxStep_j + 1$
     $offsetAmount \leftarrow (maxStep_j / 2) \times d_{capillary}$
     For ( all $S_i(u,v)$) {
       $C_{j,n} \leftarrow contour(S_i(u,v))_{contourLevel}$
       $C^{o}_{j,n} \leftarrow offset(C_{j,n})_{\text{offsetAmount}}$
       $CC_j = \{C_{j,n}\}$ ← a set of $n$ closed contour curves that $j^{th}$ layer contains
       $OC_j \leftarrow OC_j \cup \{C^{o}_{j,n}\}$
       $BC_j \leftarrow OC_i$
     }
   }
  $maxStep_j \leftarrow 2 \times maxStep_j$
}
$contourLevel \leftarrow plane(\{lb, rb, rt, lt\})$ = "0"
For ( $k$ = 1 to $K$) {
  $contourLevel \leftarrow contourLevel + interval$
  $lb(z) \leftarrow lb(z) + interval, rb(z) \leftarrow rb(z) + interval$
  $rt(z) \leftarrow rt(z) + interval, lt(z) \leftarrow lt(z) + interval$
  $offsetAmount \leftarrow (maxStep_k / 2) \times d_{capillary}$
  For ( all $S_i(u,v)$) {
    $C_{k,n} \leftarrow contour(S_i(u,v))_{contourLevel}$
    $CC_k = \{C_{k,n}\}$ ← a set of $n$ closed contour curves that $k^{th}$ layer contains
   }
   For ( all $CC_k$) { $CC^{check}_{k,n} \leftarrow 0$ }
   For ( $j$ = 1 to $maxStep_k$ ) {
     For ( $n$ = 1 to $num\_Contour\_Curves$) {
      If ($CC^{check}_{k,n} \leftarrow 0$ ) Then {
          $C^o_{k,n}$ ← $offset(C_{k,n})_{\text{offsetAmount}}$
          $OC_k$ ← $OC_k \cup \{C^o_{k,n}\}$
          **If** ( $(C^o_{k,n})^{length} \leq min\_Segment\_Length$ ) **Then** { $CC^{check}_{k,n}$ ← 1 }
          **If** (($j \geq maxStep_k/2 - cellStripe$) **and** ($j \leq maxStep_k/2$)) **Then** { $CM_k$ ← $CM_k \cup C^o_{k,n}$ }
          **ElseIf** ($j < maxStep_k/2 - cellStripe$) **Then** { $CO_k$ ← $CO_k \cup C^o_{k,n}$ }
          **Else** { $CI_k$ ← $CI_k \cup C^o_{k,n}$ }
        }
      }
      $offsetAmount$ ← $offsetAmount - d_{capillary}$
    }
}
Connect( { $CO_k$, $CI_k$, $CM_k$} )
Split( { $CO_k$, $CI_k$, $CM_k$}, $v_{capillary}$ )
Send_to_Bioprinter( { $CO_k$, $CI_k$, $CM_k$} )
**End**
```

3.6 Path Planning with Hybrid Method for Branched Vascular Constructs

Figure 3.13 shows that Self-Supporting methods implementation with vertically biomodeled smooth surfaces neither reduce the supporting material usage and duration of the bioprinting process, as it utilizes an excessive number of outwards supporting cylinders. To effectively reduce the printing time and material use, a hybrid method is proposed where self-supporting and zig-zag methods are combined.

The proposed hybrid method is mainly constructed over self-supporting method as two methodologies show great similarities. By altering the self-supporting method's border curve constraint with a more relaxed constraint and by introducing a new path planning pattern, the algorithm reduces the printing duration and material use significantly (as shown in Figure 4.10).

In self-supporting method, for any subsequent layer, whenever the boundaries of the offset contour curves OC_k of the k^{th} layer strictly encloses the largest outward boundaries of the border curves of the upper layer BC_{k+1}, then those OC_k curves are set as the border curves BC_k of k^{th} layer. However, in this hybrid method, the largest outward boundaries of the offset contour curves of the k^{th} layer do not have to enclose the boundaries of the border curves BC_{k+1} of the upper layer. This constraint conversion makes $BC_k = BC_{k+1}$ possible for layers of contouring the same vertical extruded surface group.

Once BC_k's are set for each layer, from bottom to top layer, the algorithm checks for the odd numbered layers k that satisfies $BC_k = BC_{k+1}$ property. If the border curves BC_k's are set as $curveOuts_k$, the outward support border cylinders of bio-ink cylinders are set as $curveIns_k$, and the hollow area between the $\{curveIns_k, curveOuts_k\}$ need to be filled by custom zig-zag pattern that is generated by Algorithm 6.

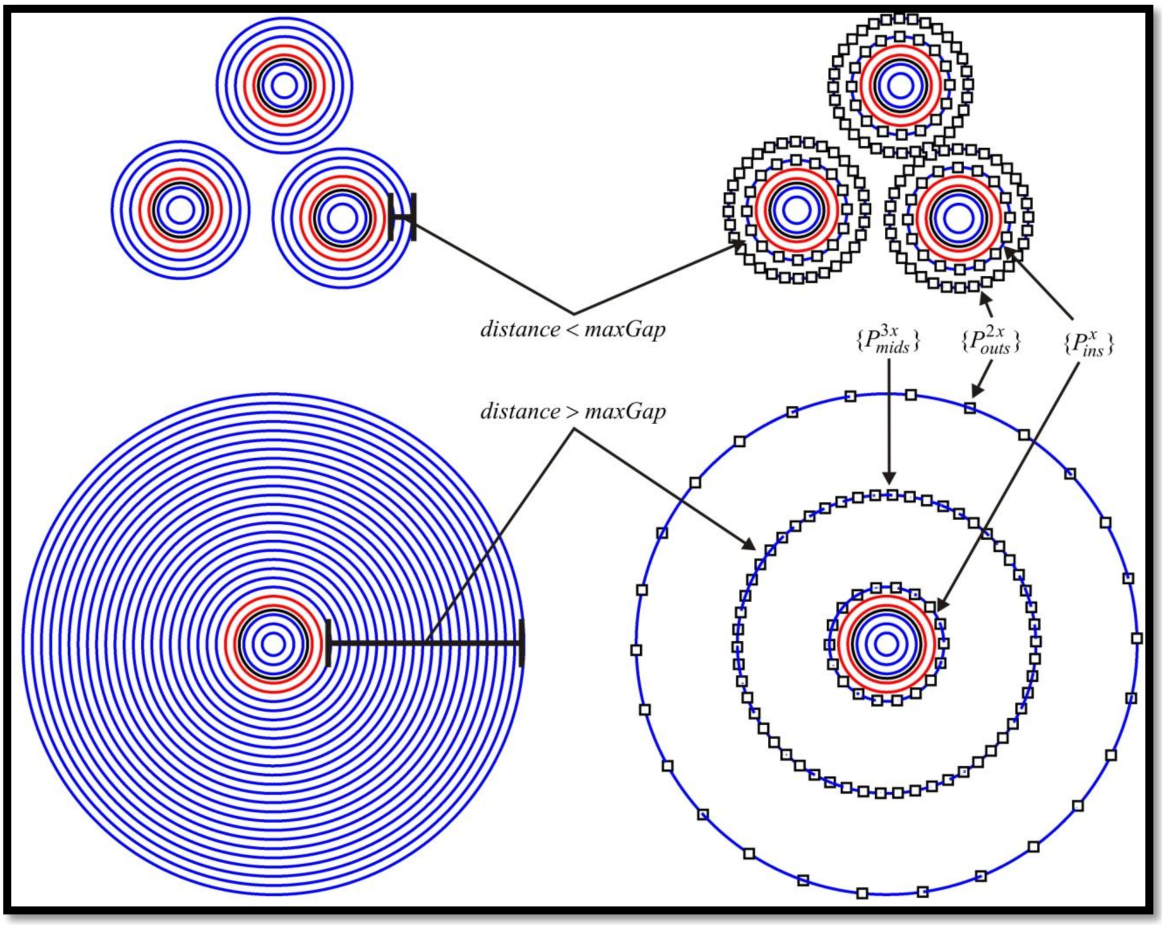

Figure 3.14. Traveling point extraction process for large and small distance curves.

If the linear distance between $curveIns_k$ and $curveOuts_k$ pairs is greater than *maxGap* (user defined distance) value, then the algorithm approximates another set of curves $curveMids_k$ that pass just from the middle of $\{curveIns_k, curveOuts_k\}$ as in Figure 3.14. Division of each curve set $\{curveIns_k, curveMids_k, curveOuts_k\}$ to *x*, *3x* and *2x* points respectively, results in relative sized point sets for each curve set $\{P^{x}_{ins}, P^{3x}_{mids}, P^{2x}_{outs}\}^{x=1..X}$ (Figure 3.14). For instance, P^{2x}_{outs} represents the *2x* division points of $curveOuts_k$. By traveling among the division points in a planned way as shown in Figure 3.15, a zig-zag patterned support structure is generated for odd layers satisfying $BC_k = BC_{k+1}$. This zig-zag patterned support structure will be the k^{th} layers support cylinders outwards (CO_k) set, as shown in Figure 3.15. The rest of the methodology is completely identical with the self-supporting approach explained above.

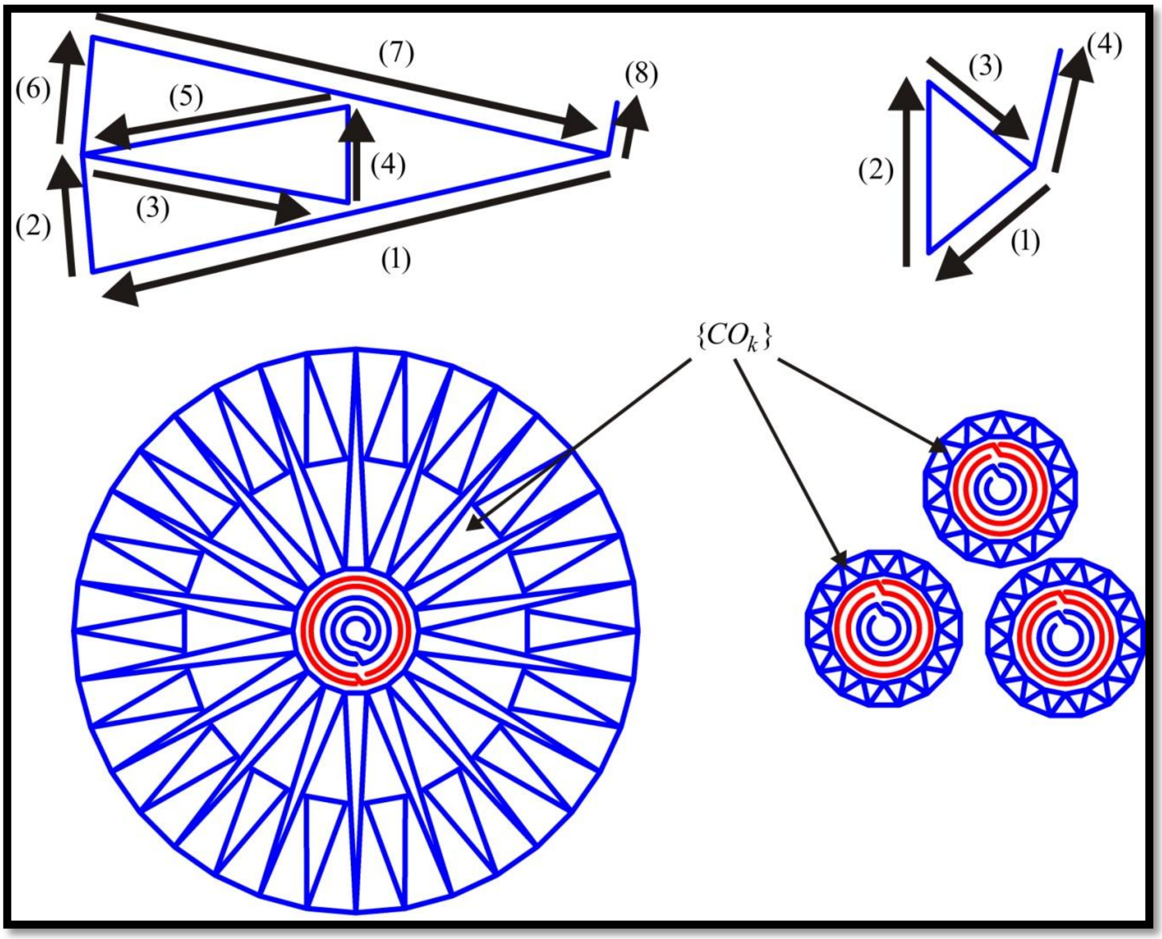

Figure 3.15. The zig-zag pattern outer support structure CO_k generation for odd numbered layers, for both large and small distance curves.

The proposed hybrid method reduces the material and time for subsequent vertical layers, and hence for the whole bioprinting process.

The pseudo-code of the Algorithm 6 is given below.

Algorithm 6. Vertical 3D-Bioprinting Hybrid Model

Input:

$S_i(u,v)$:	generated NURBS Surface(s)
$d_{capillary}$:	diameter of the glass capillaries
topSupport:	the number of supportive cylinders at top layer
$v_{capillary}$:	max. volume of the glass capillaries
cellStripe:	the number of cylindrical cellular aggregates to satisfy desired wall thickness
interval:	distance between layers
maxGap:	critical distance between contour curves

Output:

Finalized vascular model, with support structure
Path planning for 3D-Bioprinting (a compatible script file for the 3D-Bioprinter)

```
Start
{ $bb_0, bb_1, bb_2, bb_3, bb_4, bb_5, bb_6, bb_7$} ← BoundingBox($S_b(u,v)$)
$lb$ ← $bb_4$, $rb$ ← $bb_5$, $rt$ ← $bb_6$, $lt$ ← $bb_7$
Initialize $totalLayers$ ← $floor(distance(bb_3,bb_0)$ / $interval$)+1, $j$ ← 1, $k$ ← $totalLayers$, $i$ ← 1
Initialize $contourLevel$ ← $plane(\{lb, rb, rt, lt\})$ = "$k$"
$contourLevel$ ← $contourLevel$ – $interval$
$lb(z)$ ← $lb(z)$ – $interval$, $rb(z)$ ← $rb(z)$ – $interval$, $rt(z)$ ← $rt(z)$ – $interval$, $lt(z)$ ← $lt(z)$ – $interval$
$offsetAmount$ ← ($topSupport$ / 2) × $d_{capillary}$
For ( all $S_i(u,v)$) {
  $C_{k,n}$ ← $contour(S_i(u,v))_{contourLevel}$
  $C^o_{k,n}$ ← $offset(C_{k,n})_{\text{offsetAmount}}$
  $CC_k = \{C_{k,n}\}$ ← a set of $n$ closed contour curves that $k^{th}$ layer contains
  $OC_k$ ← $OC_k \cup \{C^o_{k,n}\}$
  $BC_k$ ← $OC_k$
  $maxStep_k$ ← $topSupport$
}
For ( $k$ = 1 to $K$) {
  $contourLevel$ ← $contourLevel$ – $interval$
  $lb(z)$ ← $lb(z)$ – $interval$, $rb(z)$ ← $rb(z)$ – $interval$
  $rt(z)$ ← $rt(z)$ – $interval$, $lt(z)$ ← $lt(z)$ – $interval$
   $j$ ← $K – k$
  $BC^j_{j+1}$ ← $transport(BC_{j+1})_{contourLevel}$
  $maxStep_j$ ← 1
  $offsetAmount$ ← ($maxStep_j$ / 2) × $d_{capillary}$
  For ( all $S_i(u,v)$) {
    $C_{j,n}$ ← $contour(S_i(u,v))_{contourLevel}$
    $C^o_{j,n}$ ← $offset(C_{j,n})_{\text{offsetAmount}}$
     $CC_j = \{C_{j,n}\}$ ← a set of $n$ closed contour curves that $j^{th}$ layer contains
     $OC_j$ ← $OC_j \cup \{C^o_{j,n}\}$
  }
  $borderCurves_j$ ← $OC_i$
  While ($BC^j_{j+1} \supset BC_j$) {
    $maxStep_j$ ← $maxStep_j$ + 1
    $offsetAmount$ ← ($maxStep_j$ / 2) × $d_{capillary}$
    For ( all $S_i(u,v)$) {
      $C_{j,n}$ ← $contour(S_i(u,v))_{contourLevel}$
      $C^o_{j,n}$ ← $offset(C_{j,n})_{\text{offsetAmount}}$
       $CC_j = \{C_{j,n}\}$ ← a set of $n$ closed contour curves that $j^{th}$ layer contains
      $OC_j$ ← $OC_j \cup \{C^o_{j,n}\}$
       $BC_j$ ← $OC_i$
    }
   }
  $maxStep_j$ ← 2 × $maxStep_j$
```

```
}
$contourLevel \leftarrow plane(\{lb, rb, rt, lt\})$ = "0"
For ( $k$ = 1 to $totalLayers$) {
  $contourLevel \leftarrow contourLevel + interval$
  $lb(z) \leftarrow lb(z) + interval$, $rb(z) \leftarrow rb(z) + interval$
  $rt(z) \leftarrow rt(z) + interval$, $lt(z) \leftarrow lt(z) + interval$
  $offsetAmount \leftarrow (maxStep_k / 2) \times d_{capillary}$
  For ( all $S_i(u,v)$) {
    $C_{k,n} \leftarrow contour(S_i(u,v))_{contourLevel}$
    $CC_k = \{C_{k,n}\}$ ← a set of $n$ closed contour curves that $k^{th}$ layer contains
   }
   If (($BC_k == BC_{k+1}$) and ($k$ == ODD)) Then {
      $amount \leftarrow (cellStripe + ½) \times d_{capillary}$
      $curveIns_k \leftarrow offset(CC_k)_{amount}$
      $curveOuts_k \leftarrow BC_k$
      If ($maxStep_k / 2 \times d_{capillary} > maxGap$) Then {
        $amount \leftarrow maxStep_k / 4 \times d_{capillary}$
        $curveMids_k \leftarrow offset(curveIns_k)_{amount}$
        $x \leftarrow avg(curveIns^{length})$
        $P_{ins}^{x} \leftarrow divide(curveIns_k)_x$
        $P_{mids}^{3x} \leftarrow divide(curveMids_k)_{3x}$
        $P_{outs}^{2x} \leftarrow divide(curveOuts_k)_{2x}$
        For ($x$ = 0 to $X$) { '// For all $CC_k$
          If ( $x$ == 0) Then {
            $PP_k \leftarrow PP_k \cup \{P_{ins}^{0} \cup P_{outs}^{last\_index} \cup P_{outs}^{0} \cup P_{mids}^{last\_index} \cup P_{mids}^{1} \cup P_{outs}^{0} \cup P_{outs}^{1} \cup P_{ins}^{0}\}$
          }
          ElseIf ( ($x > 0$) and ($x < X$)) Then {
            $PP_k \leftarrow PP_k \cup \{P_{ins}^{k} \cup P_{outs}^{2k-1} \cup P_{outs}^{2k} \cup P_{mids}^{3k-1} \cup P_{mids}^{3k+1} \cup P_{outs}^{2k} \cup P_{outs}^{2k+1} \cup P_{ins}^{k}\}$
          }
          Else { $CO_k \leftarrow polyLine(PP_k)$}
        }
      }
      Else {
        $x \leftarrow avg(curveIns^{length})$
        $P_{ins}^{x} \leftarrow divide(curveIns_k)_x$
        $P_{outs}^{2x} \leftarrow divide(curveOuts_k)_{2x}$
        For ($x$ = 0 to $X$) { '// For all $CC_k$
          If ( $x$ == 0) Then {
            $PP_k \leftarrow PP_k \cup \{P_{ins}^{0} \cup P_{outs}^{last\_index} \cup P_{outs}^{1} \cup P_{ins}^{0}\}$
          }
          ElseIf ( ($x > 0$) and ($x < X$)) Then {
            $PP_k \leftarrow PP_k \cup \{P_{ins}^{k} \cup P_{outs}^{2k-1} \cup P_{outs}^{2k+1} \cup P_{ins}^{k}\}$
          }
          Else { $CO_k \leftarrow polyLine(PP_k)$}
        }
      }
    For ( all $CC_k$) { $CC_{k,n}^{check}$ ← 0 }
    For ( $j$ = $maxStep_k/2 - cellStripe$ to $maxStep_k$ ) {
    For ( $n$ = 1 to $num\_Contour\_Curves$) {
      If ($CC_{k,n}^{check}$ ← 0 ) Then {
        $C_{k,n}^{o}$ ← $offset(C_{k,n})_{\text{offsetAmount}}$
        $OC_k$ ← $OC_k \cup \{C_{k,n}^{o}\}$
        If ( $(C_{k,n}^{o})^{length} \leq min\_Segment\_Length$ ) Then { $CC_{k,n}^{check}$ ← 1 }
        If (($j \geq maxStep_k/2 - cellStripe$) and ($j \leq maxStep_k/2$)) Then { $CM_k$ ← $CM_k \cup C_{k,n}^{o}$}
        Else { $CI_k$ ← $CI_k \cup C_{k,n}^{o}$}
      }
     }
     $offsetAmount$ ← $offsetAmount - d_{capillary}$
    }
   }
  Else {
   For ( all $CC_k$) { $CC_{k,n}^{check}$ ← 0 }
    For ( $j$ = 1 to $maxStep_k$ ) {
    For ( $n$ = 1 to $num\_Contour\_Curves$) {
      If ($CC_{k,n}^{check}$ ← 0 ) Then {
        $C_{k,n}^{o}$ ← $offset(C_{k,n})_{\text{offsetAmount}}$
        $OC_k$ ← $OC_k \cup \{C_{k,n}^{o}\}$
        If ( $(C_{k,n}^{o})^{length} \leq min\_Segment\_Length$ ) Then { $CC_{k,n}^{check}$ ← 1 }
        If (($j \geq maxStep_k/2 - cellStripe$) and ($j \leq maxStep_k/2$)) Then { $CM_k$ ← $CM_k \cup C_{k,n}^{o}$}
        ElseIf ($j < maxStep_k/2 - cellStripe$) Then { $CO_k$ ← $CO_k \cup C_{k,n}^{o}$}
        Else { $CI_k$ ← $CI_k \cup C_{k,n}^{o}$}
      }
     }
     $offsetAmount$ ← $offsetAmount - d_{capillary}$
    }
  }
}
Connect( { $CO_k$, $CI_k$, $CM_k$} )
Split( { $CO_k$, $CI_k$, $CM_k$}, $v_{capillary}$ )
Send_to_Bioprinter( { $CO_k$, $CI_k$, $CM_k$} )
End
```

3.7 Generating Horizontal Centerline Curves to Guide Path Planning of Horizontal Branched Vascular Construct Printing

As an alternative approach to vertical bioprinting of biomimetic vascular constructs, a horizontal path planning methodology is proposed in Sections 3.7 and 3.8. To optimize this bioprinting horizontal path planning, the centerline curves must be generated to guide the bioprinter to follow the trajectories for creating the vascular constructs.

The steps for the proposed horizontal methods follow the same way as the vertical surface generation method (Section 3.4) but in the horizontal *x*,*y*-plane. Starting from a root trajectory curve (TC_1), each trajectory curve $\{TC_i\}_{i=1..I}$, is duplicated from the end point of its parent curve. An example is shown in Figure 3.16 for a five branched vascular structure with nine trajectory curves.

After each trajectory curve is determined, from each leaf curve (trajectory curves without any child, a total *b* branches in the structure) to the root, algorithm connects child trajectory curves with its parents through the root. This connected curve $mainRoads_b$ is set as the centerline for the respective branch *b* (as shown in Figure 3.16).

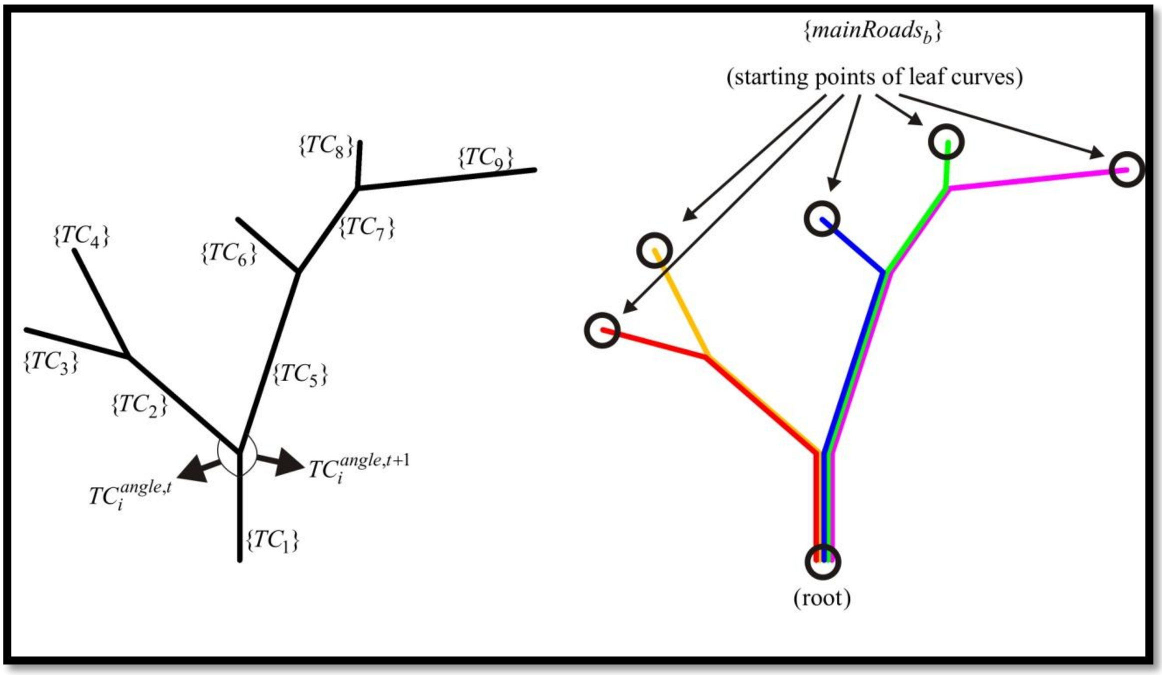

Figure 3.16. Approximation of centerline curves using trajectory curve information and the angles between parent-child trajectory curve pairs; and generation of branch number main roads from leaves to the root.

This main roads set represents the branch curves, each of them starting from a root curve. For path planning purpose, the main road curve set needs to be sorted in the clockwise direction where the base is the root curve. However, simple Euclidean coordinate system information is not enough to determine the order of the branches.

Therefore, each curve in *mainRoads* set is offset in counterclockwise direction and the total intersection number with the *mainRoads* set is determined as the sorted rank of that specific $mainRoads_b$ branch as shown in Figure 3.17.

Lastly, each consecutive main road pair, $\{mainRoads_b, mainRoads_{b+1}\}$, intersected and combined from the intersection point to create the trajectory curves to be used in Section 3.8 to organize path planning, the set of $\{sortedRoads_b\}_{b=0..B}$ constitutes each bridge curve from $mainRoads_b$ to $mainRoads_{b+1}$, including $sortedRoads_0 = mainRoads_1$. Figure 3.17 shows the sorted roads set (shown with blue curves) for a five branched structure, at the same time $sortedRoads_2$ is highlighted with red arrow (also implies the printing direction).

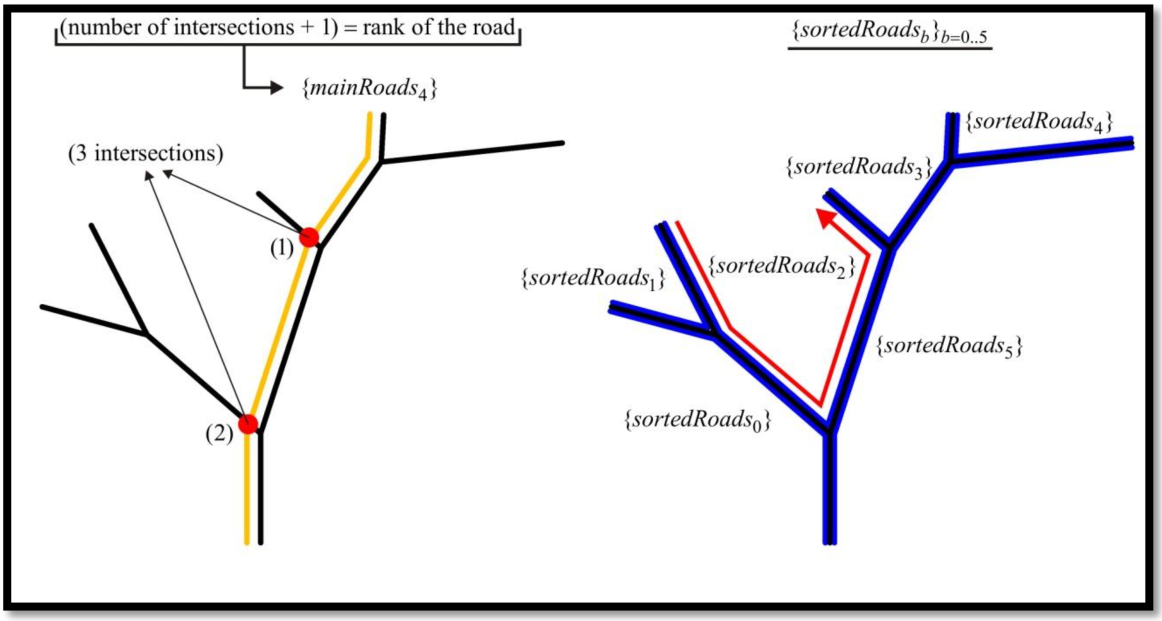

Figure 3.17. Determining the ranks of the main roads and generating arranged sorted roads for 3D printing.

The pseudo-code of the Algorithm 7 is given below.

Algorithm 7. Centerline Curves Generation for Horizontal 3D-Bioprinting

```
Input:
TC:                 trajectory curve array
pS:                 the predefined spot for starting point of trajectory curves & smooth surfaces
Output:
sortedRoads_b:      updated trajectory curve array
Start
Initialize maxBranches ← 2, i ← 1, constructionPlane ← (x,y), t ← 1, intersection ← 0
totalRadius ← $sum(\mathrm{TC}_i^{median\_rad})_{i = 1..I}$
```

```
$cpCount \leftarrow count(\mathrm{TC}_i^{CP})_{i = 1..I}$
$meanRadius \leftarrow totalRadius \,/\, cpCount$
$destination_i \leftarrow (pS\ (x),\ pS\ (y) + \mathrm{TC}_i^{lengt\ h},\ pS\ (z))$
$line_i \leftarrow addLine(\{pS,\ destination_i\})$
$\mathrm{TC}_i^{curve\_id} \leftarrow line_i$
$(\mathrm{TC}_i^{child,t})^{startPoint} \leftarrow line_i^{endPoint}$ , $(\mathrm{TC}_i^{child,t+1})^{startPoint} \leftarrow line_i^{endPoint}$
$(\mathrm{TC}_i^{child,t})^{side} \leftarrow$ 1 = “left”, $(\mathrm{TC}_i^{child,t+1})^{side} \leftarrow$ 2 = “right”
**For** ( $i$ = 1 to $I$) {
  **If** ($\mathrm{TC}_i^{parent} \neq$ NULL) **Then** {
    **If** ($\mathrm{TC}_i^{side}$ == 1 = “left”) **Then** {
      $destination_i \leftarrow findDestination(\{\ \mathrm{TC}_i^{parent,curve\_id},\ \mathrm{TC}_i^{startPoint},\ 2\Pi - \mathrm{TC}_i^{angle}\ \})_{constructionPlane}$
      $line_i \leftarrow addLine(\{pS,\ destination_i\})$
      $\mathrm{TC}_i^{curve\_id} \leftarrow line_i$
      **If** ($\mathrm{TC}_i^{child} \neq$ NULL) **Then** {
        $(\mathrm{TC}_i^{child,t})^{startPoint} \leftarrow line_i^{endPoint}$ , $(\mathrm{TC}_i^{child,t+1})^{startPoint} \leftarrow line_i^{endPoint}$
        $(\mathrm{TC}_i^{child,t})^{side} \leftarrow$ 1 = “left”, $(\mathrm{TC}_i^{child,t+1})^{side} \leftarrow$ 2 = “right”
      }
    }
    **Else** {
      $destination_i \leftarrow findDestination(\{\ \mathrm{TC}_i^{parent,curve\_id},\ \mathrm{TC}_i^{startPoint},\ 2\Pi - \mathrm{TC}_i^{angle}\ \})_{constructionPlane}$
      $line_i \leftarrow addLine(\{pS,\ destination_i\})$
      $\mathrm{TC}_i^{curve\_id} \leftarrow line_i$
      **If** ($\mathrm{TC}_i^{child} \neq$ NULL) **Then** {
        $(\mathrm{TC}_i^{child,t})^{startPoint} \leftarrow line_i^{endPoint}$ , $(\mathrm{TC}_i^{child,t+1})^{startPoint} \leftarrow line_i^{endPoint}$
        $(\mathrm{TC}_i^{child,t})^{side} \leftarrow$ 1 = “left”, $(\mathrm{TC}_i^{child,t+1})^{side} \leftarrow$ 2 = “right”
      }
    }
  }
}
**For** ( $i$ = 1 to $B$) {
  **If** ($\mathrm{TC}_i^{child}$ == NULL) **Then** {
    $mainRoads_i \leftarrow \mathrm{TC}_i^{curve\_id}$
  }
}
**For** ( $i$ = 1 to $B$) {
  **While** ($mainRoads_i^{startPoint} \neq pS$) **Then** {
    $mainRoads_i \leftarrow mainRoads_i \cup mainRoads_i^{parent}$
  }
}
$sortedRoads_0 \leftarrow$ NULL
**For** ( $i$ = 1 to $B$) {
  $intersection \leftarrow 1$
  $testCurve \leftarrow offset(mainRoads_i)_{0.1}$
  For ( $j$ = 1 to $B$) {
    If (($testCurve \cap mainRoads_j$) ≠ NULL ) Then {
      $intersection \leftarrow intersection + 1$
    }
  }
  $sortedRoads_{intersection} \leftarrow mainRoads_i$
  $dummy \leftarrow sortedRoads_{intersection}^{endPoint}$
  $sortedRoads_{intersection}^{endPoint} \leftarrow sortedRoads_{intersection}^{startPoint}$
  $sortedRoads_{intersection}^{startPoint} \leftarrow dummy$
}
$sortedRoads_0 \leftarrow sortedRoads_1$
For ( $i$ = 1 to $B - 1$) {
   $pt \leftarrow firstIntersectionPoint(\{sortedRoads_i, sortedRoads_{i+1}\})$
  $line1 \leftarrow addSubCurve(\{sortedRoads_i^{startPoint}, pt\})$
  $line2 \leftarrow addSubCurve(\{pt, sortedRoads_{i+1}^{startPoint}\})$
  $sortedRoads_i \leftarrow line1 \cup line2$
}
End
```

3.8 Path Planning for Horizontal Vascular Construct Printing

After obtaining the trajectory centerline curve set {*sortedRoads*$_b$} from Algorithm 7, the path planning for bioprinting is determined. Since bioprinting is limited to a single plane, the radius of the vascular construct must be constant throughout. Moreover, the path planning is also limited to the capillary diameter $d_{capillary}$, as the parallel horizontal sequence of support structure and bio-ink pieces must be exactly differ by $d_{capillary}$ (Figure 3.18). Therefore, mean radius of the model, *meanRadius*, is calculated and converted to a cylindrical quantity *vesselRadius* by dividing it to *elevate* (distance between layers) amount. As also can be seen from Figure 3.18, the total cylinder numbers of support structures and bio-ink's at each layer, $layerLength_i$, is decreasing by one at every successive layer for each branch, so that each cylinder can lie through the valleys that have been formed by the previous layer.

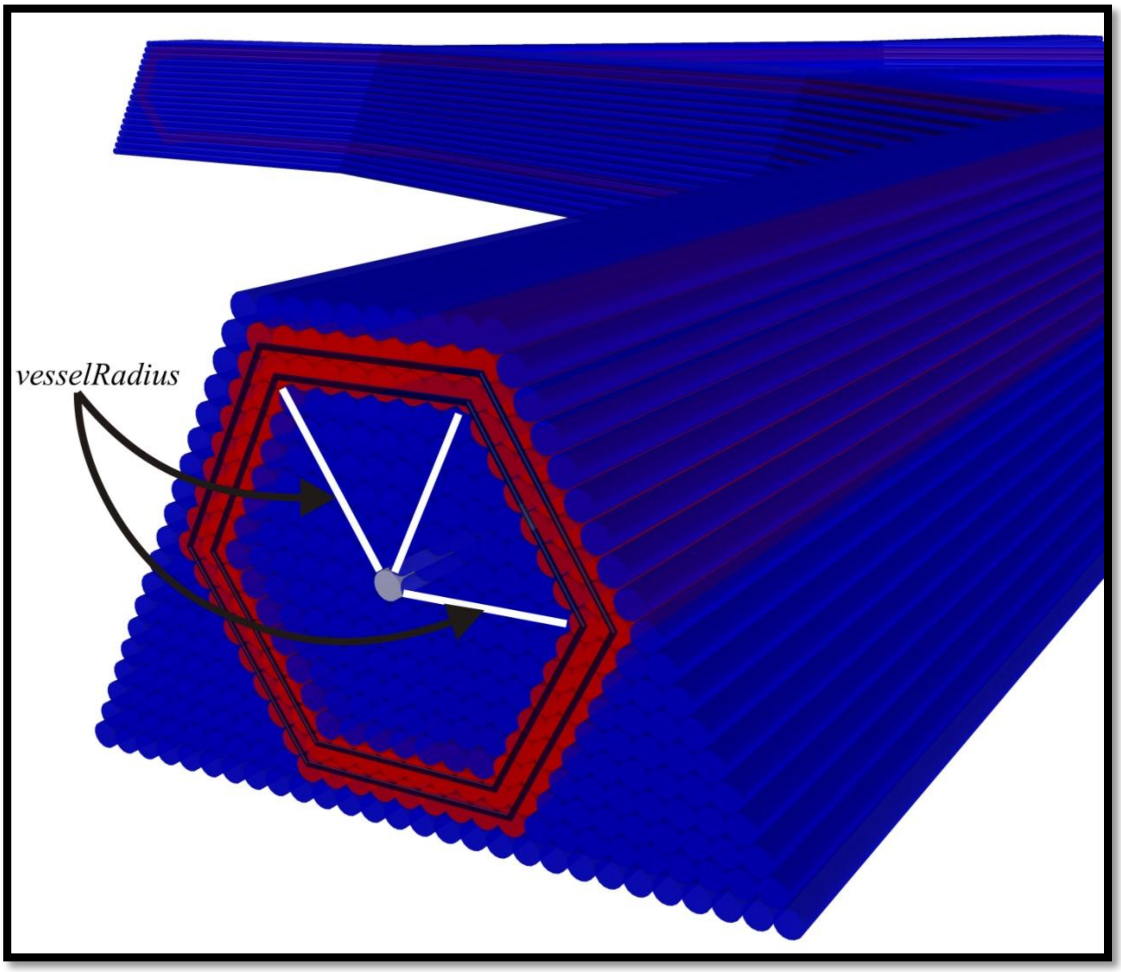

Figure 3.18. The organization of support structures (blue) and cellular aggregates (red) with respect to the vessel models radius, in horizontal path planning.

Using the number of support structure layers at top and bottom and the *cellStripe* value (the number of bio-ink cylinders to mimic natural wall thickness), the total number of layers, $i=1..I$, is determined. Then, at start points of each $sortedRoads_b$ curve, *cellStripe* numbers of six-edged polygons PG = $\{PG_{b,cellStripe}\}_{b=1..B}$ are placed, with the radius of *vesselRadius*, as shown for a five branched vascular structure in Figure 3.19.

For each layer, the curves $sortedRoads_b$ are transported vertically in an order from $sortedRoads_1$ to $sortedRoads_B$ to the level of that specific layer. Then each $sortedRoads_b$ curves are offset in *x-y* plane with offset amounts, incrementing with capillary diameter $d_{capillary}$ amount, in clockwise and counterclockwise directions. If those offset curves intersects with polygons $PG_{b,cellStripe}$ for that branch, then offset curves are set as cylindrical bio-ink pieces CM_b. If there is no intersection, then the offset curves are set as cylindrical support structures CO_b (in Figure 3.19, blue cylinders represents the support structures, while red ones are representing bio-ink).

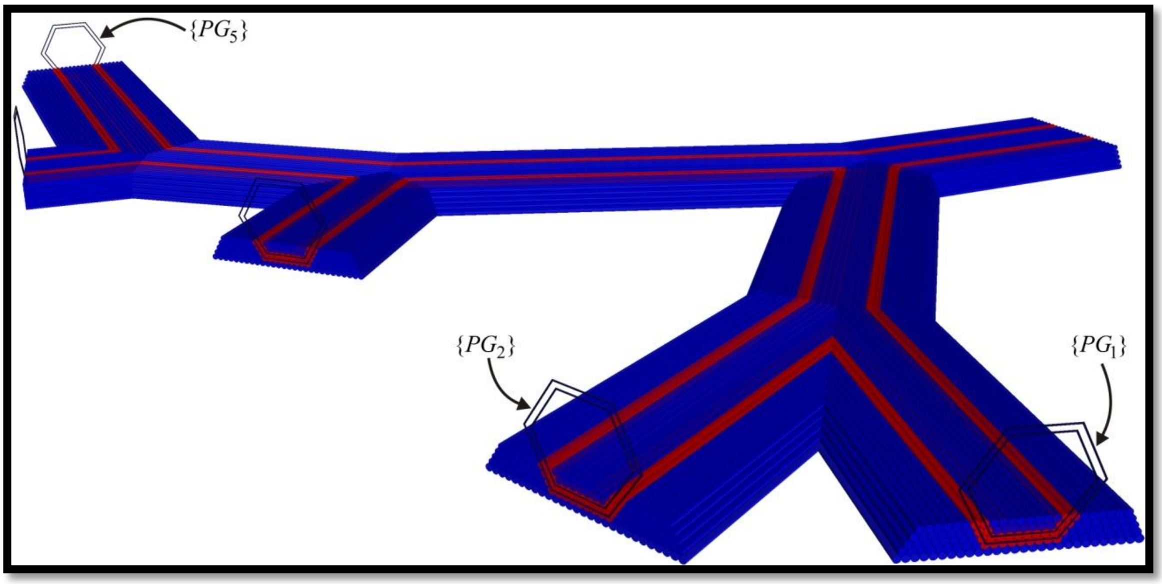

Figure 3.19. Placements of polygons in order to classify support structures and cellular aggregates for horizontal path planning.

At a layer, the support cylinders are printed first, then the bio-ink aggregates are deposited along the valleys that support structures are formed as shown in Figure 3.20.

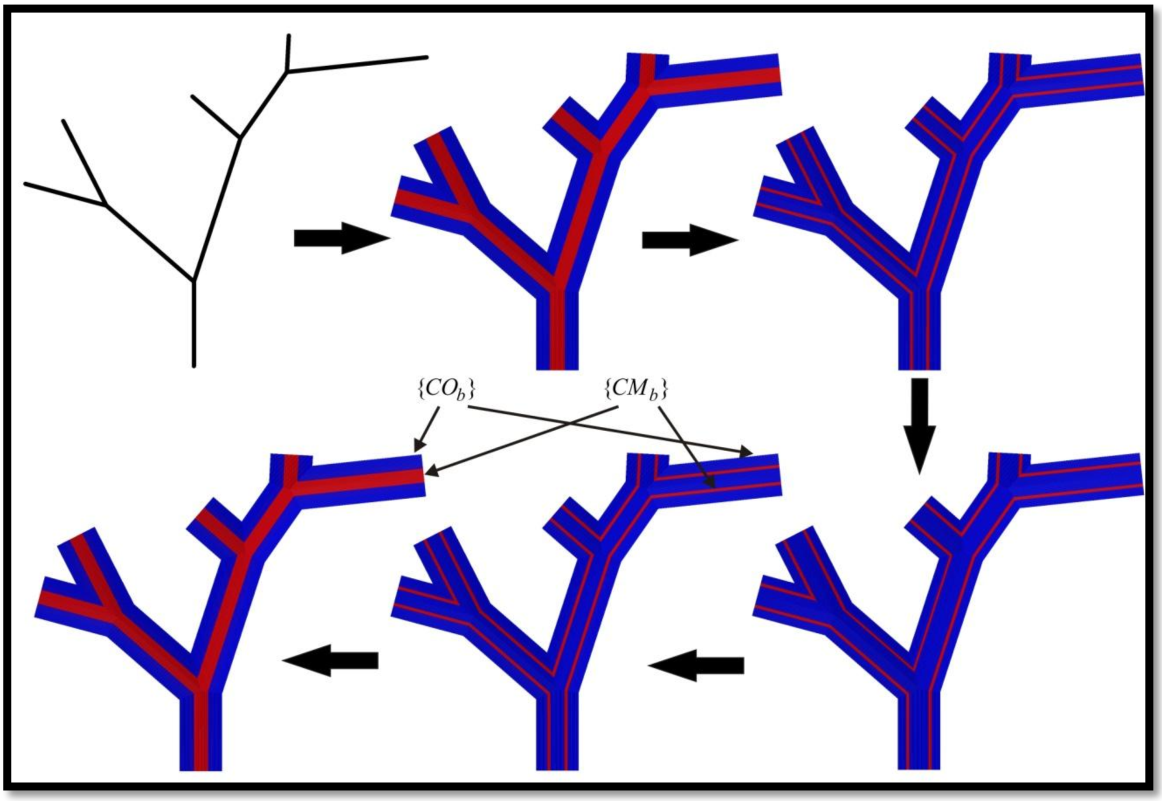

Figure 3.20. Five example layers from a five branched vascular construct to express the path planning topology for horizontal printing, blue cylinders represents the support structures, while red ones are representing bio-ink.

The pseudo-code of the Algorithm 8 is given below.

Algorithm 8. Horizontal 3D-Bioprinting Self-Supporting Model

Input:

$\{sortedRoads_b\}_{b=0..B}$:	generated centerline trajectory curve array
$d_{capillary}$:	diameter of the glass capillaries
meanRadius:	average radius value of the vascular structure
support:	the total number of supportive layers (both on top and bottom)
$v_{capillary}$:	max. volume of the glass capillaries
cellStripe:	the number of cylindrical cellular aggregates to satisfy desired wall thickness
elevate:	distance between layers

Output:

Finalized vascular model, with support structure

Path planning for 3D-Bioprinting (a compatible script file for the 3D-Bioprinter)

```
Start
$vesselRadius \leftarrow floor(meanRadius / elevate)$ ‘// radius in terms of cylinders
Initialize $totalLayers \leftarrow (2 \times (vesselRadius - 1) + 1) + cellStripe \times 2 + support$
Initialize $baseLength \leftarrow vesselRadius + cellStripe + totalLayers$
Initialize $j \leftarrow 1$, $k \leftarrow 0$, $i \leftarrow 1$, $layerLength_i \leftarrow baseLength$
For ( $i = 2$ to $I$) {
  $layerLength_i \leftarrow baseLength + 1 - i$
```

```
}
For ( i = 1 to I) {
  For ( k = 0 to B) {
    pt ← $sortedRoads_k^{startPoint}$
    For ( l = 1 to cellStripe) {
      $pg_k^l$ ← addPolygon({((vesselRadius + 1 –  l) × $d_{capillary}$), pt, 6})
      $PG_k$ ← $PG_k \cup pg_k^l$
    }
    curve ← move({$sortedRoads_k$, (– (((totalLayers – i) / 2) – i) })$_z$
    If ( k == 0) Then {
      For ( j == 1 to ceil($layerLength_i$ / 2)) {
        If (($layerLength_i$ / 2) == INT) Then {
          offsetAmount ← ( j – 1) ×  $d_{capillary}$
          crv ← offset(curve)$_{offsetAmount}$
        }
        Else {
          offsetAmount ← ( j – 0.5) ×  $d_{capillary}$
          crv ← offset(curve)$_{offsetAmount}$
        }
        If ( (crv ∩ $PG_k$) == NULL) Then { $CO_k$ ← $CO_k$ ∪ crv }
        If ( (crv ∩ $PG_k$) ≠ NULL) Then { $CM_k$ ← $CM_k$ ∪ crv }
      }
    }
    Else {
      For ( j == 1 to floor($layerLength_i$ / 2)) {
        If (($layerLength_i$ / 2) == INT) Then {
          offsetAmount ← ( j – 1) ×  $d_{capillary}$
          crv ← offset(curve)$_{offsetAmount}$
        }
        Else {
          offsetAmount ← ( j – 0.5) ×  $d_{capillary}$
          crv ← offset(curve)$_{offsetAmount}$
        }
        If ( (crv ∩ $PG_k$) == NULL) Then { $CO_k$ ← $CO_k$ ∪ crv }
        If ( (crv ∩ $PG_k$) ≠ NULL) Then { $CM_k$ ← $CM_k$ ∪ crv }
      }
    }
    If ( ($layerLength_i$ == ODD) and (k ≠ 0)) Then {
      pt ← firstIntersectionPoint({curve,{$CO_k$, $CM_k$}})
     crv ← addSubCurve({$curve^{startPoint}$, pt})
     If ( (crv ∩ $PG_k$) == NULL) Then { $CO_k$ ← $CO_k$ ∪ crv }
     If ( (crv ∩ $PG_k$) ≠ NULL) Then { $CM_k$ ← $CM_k$ ∪ crv }
    }
  }
  Connect( { $CO_k$, $CM_k$} )
  Split( { $CO_k$, $CM_k$}, $v_{capillary}$ )
  Send_to_Bioprinter( { $CO_k$, $CM_k$} )
}
End
```

Chapter 4

Implementations and Examples

4.1 Material (Hydrogel) & Bio-ink Preparation

A bio-inert, thermo-reversible hydrogel called NovoGel was used as a support material for 3D printing of the developed models. The preparation of 2% (w/v) NovoGel (Organovo) was carried out with phosphate buffered saline (PBS: Thermo Scientific Hyclone 1X) with Ca^{2+} and Mg^{2+} salts. The solution was mixed with magnetic stirrer and it was kept in microwave for 1 minute on high power settings. Then, the solution was located in a water bath set at 70°C. NovoGel solution was autoclaved following standard liquid sterilization procedures [4].

The 3D bioprinting requires a uniformly flat surface. After the sterilizing a 2 % Agarose solution with PBS (Thermo Scientific Hyclone 1X), 20 mL agarose solution is transferred using a pipette onto a petri dish bottom covering the entire dish surface. In compliance with aseptic techniques, the sterilized mold was slowly put down onto the agarose inside the petri dish. The mold was carefully taken away from the petri dish after the agarose solution became completely gel [4]. During the material preparation, adequate sterilization rules are followed against any contamination.

For Bio-ink preparation [20], immortalized MEF cells were cultured in 15 cm-diameter culture dishes. Cells were detached from the culture plate using two different approaches. Cells were either detached using 0.1% trypsin (Biological Industries, Israel) for 10 minutes (Exp1), or 2.5 mM EGTA (ethylene glycol-bis(2-aminoethyl ether)-N,N,N´,N) (Idranal VI, Fluka, Germany) in PBS [20]. Following detachment, trypsin or EGTA was neutralized using serum containing medium. Following detachment, cells were centrifuged at 200 x g and supernatant was discarded. The cell pellet was resuspended to obtain 10 x 10^6 cells / 20 ml medium incubated at 37°C in 15 ml-conical tubes under rotation (PTR-30 Grant-Bio rotator, U.K). Following pelleting, cells were resuspended in 1 ml medium and transferred into 1.5 ml Eppendorf tubes and centrifuged again (1000 x g). Then, the cell pellets were drawn into capillary tubes. Following incubation of cells in capillaries at indicated times in a 50 ml-falcon tube containing culture medium inside tissue culture incubator. Following incubation, cells inside the capillary tubes were extruded into cylindrical grooves on agarose gel (2% in PBS). Then, plates were covered with culture medium and put into the incubator until cylindrical bioinks are formed. Cylindrical bioink MEF cells were drawn back into capillary tubes and bioprinting was performed using the 3D bio-printer.

For continuous bioprinting, cells were centrifuged at 200 x g. The pellet was resuspended to have 20 x 10^6 cells / 6.5 ml and transferred into 15 ml-conical tubes. Following rotation at 37°C, cells were pelleted and transferred into Eppendorf tubes. Cell pellets in 1.5 ml Eppendorf tubes (60 x 10^6 cells in total) were transferred into capillaries by continuous bioprinting [4].

4.2 Accuracy Results of Biomimetic Biomodeling Phase

To highlight the proposed biomimetic biomodeling methods capabilities, three different mesh models of blood vessels are used, as shown in Figure 4.1. First two of them are three branched and the last one is five branched. According to the visual observations from Figure 4.1, the generated parametric smooth surfaces shapes mostly mimic the natural vessel geometries of the mesh models.

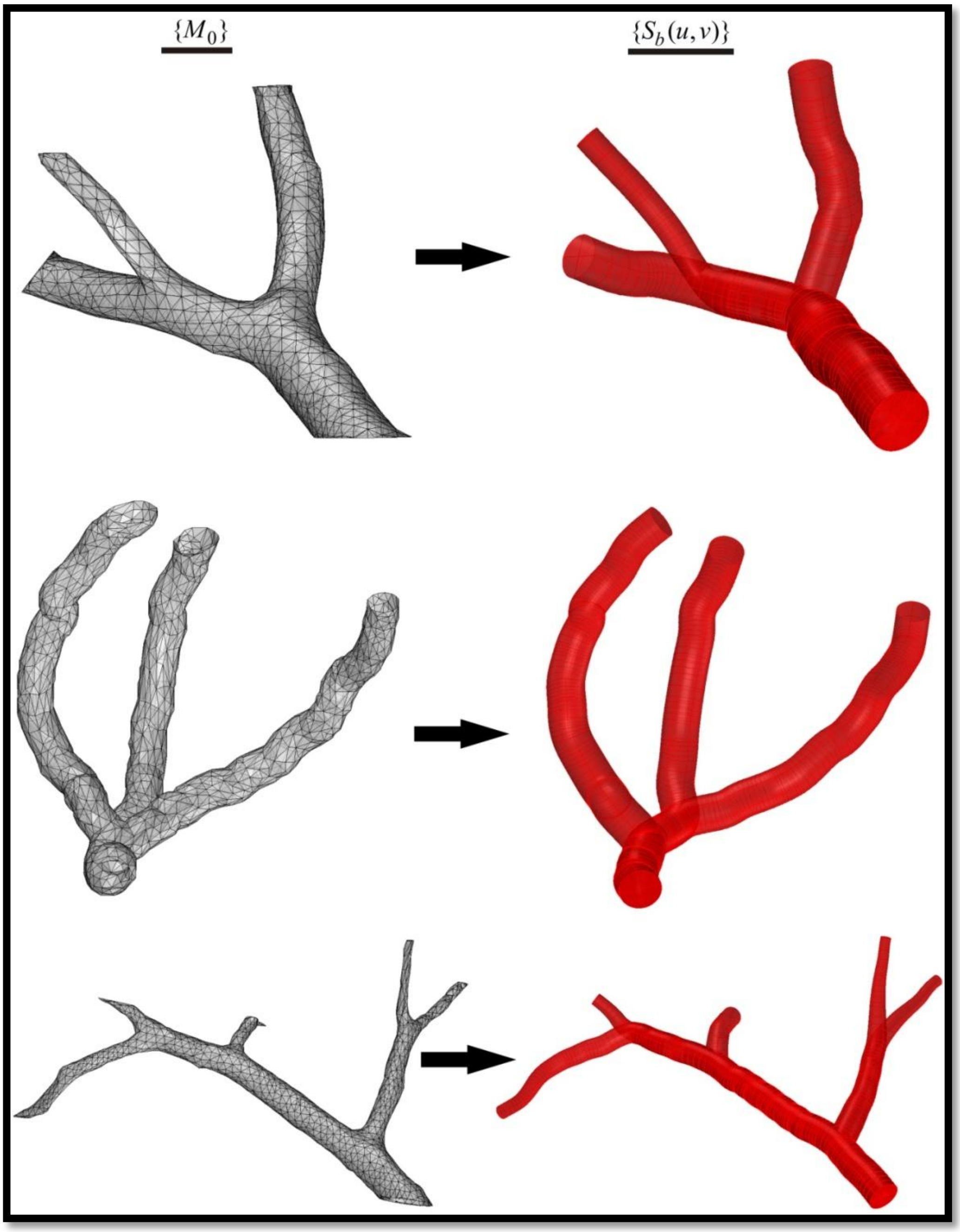

Figure 4.1. Biomodeling results of branched vascular constructs, from mesh models to smooth surfaces.

To conduct quantification analysis of the errors, contour curves are obtained with periodical increments for abdominal aorta's mesh model and abdominal aorta's smooth surface model (shown in Figure 4.2).

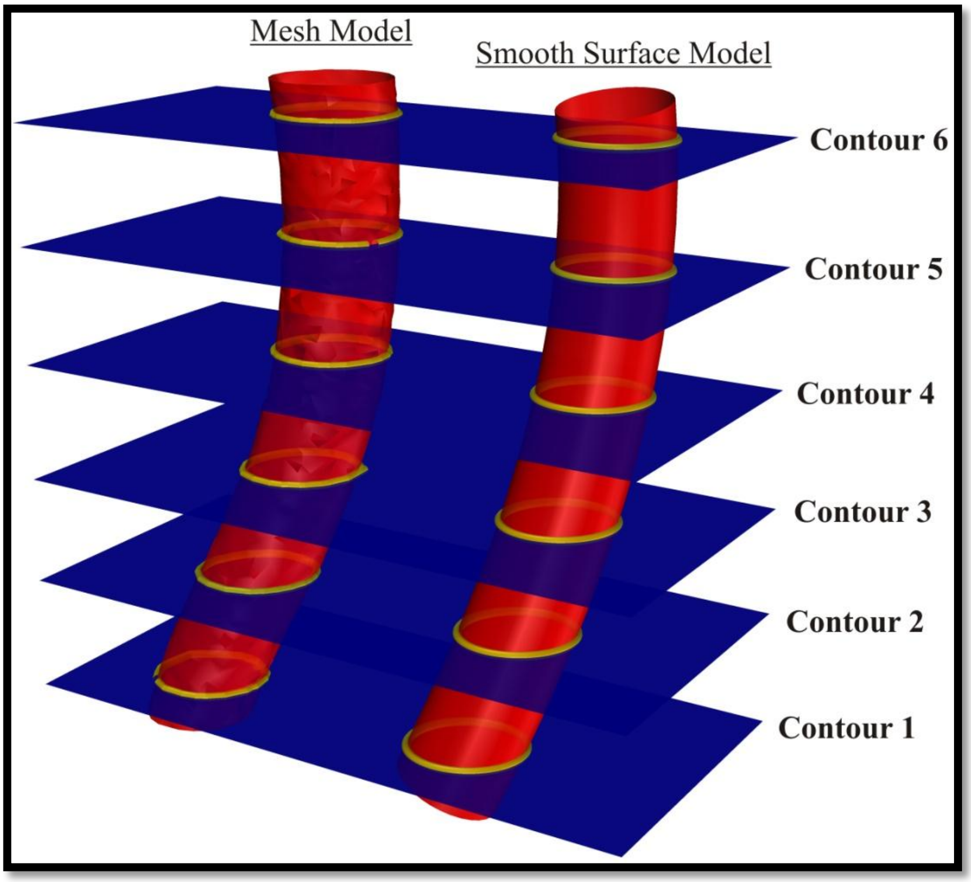

Figure 4.2. Contouring operations for both Mesh and Smooth Surface Model.

Then minimum, maximum and average difference (distance) of mesh contour curves and surface contour curves at all contour layers are found. Those values are the errors of biomodeling algorithm for the abdominal aorta model. The results of the comparisons are shown in Table 4.1:

Error / Contour #	Min. Error (mm)	Max. Error (mm)	Avg. Error (mm)
Contour 1	0.06	0.91	0.52
Contour 2	0.045	0.84	0.4
Contour 3	0.02	0.26	0.11
Contour 4	0.01	0.29	0.14
Contour 5	0.05	0.72	0.43
Contour 6	0.063	0.88	0.65

Table 4.1. Six countours errors of the biomimetic biomodeling phase for abdominal aorta model.

Observing the quantification of the errors, as ultimate goal is to generate path planning for bioprinting of vascular constructs in a reasonable mimicked way, those errors are negligible as abdominal aorta models, which is used in this work, diameter is around 9 mm.

Another set of analysis are done for evaluating the surface smoothness of the generated smooth surface model. Since the mesh model generated by the segmentation software includes a lot of noise and errors, approximation with a smooth surface eliminates surface roughness and errors. As shown in Figure 4.3, the continuity of the black and white stripes from top to bottom of the aorta's smooth surface model indicates smooth connection, tangency, and curvature match through the domain of the surface. Therefore, the results of smoothness of biomodeling phase are highly satisfying.

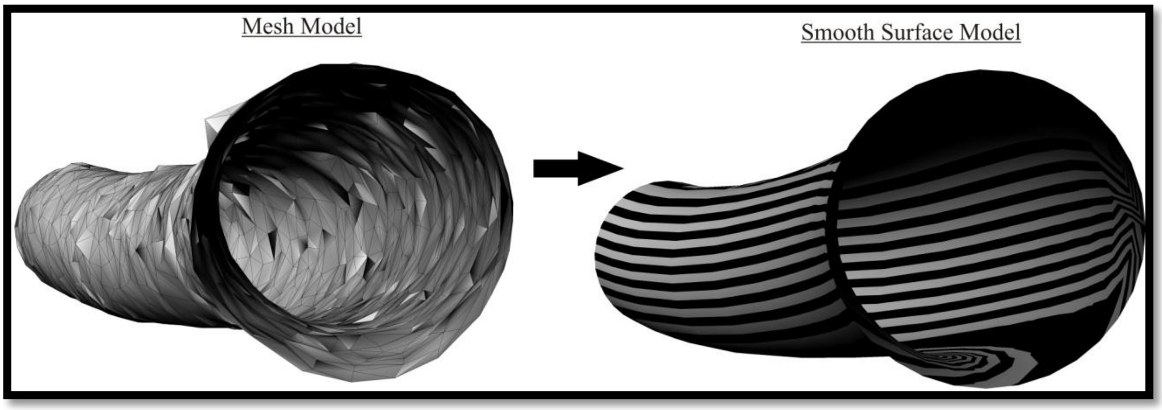

Figure 4.3. Smoothness analysis for the mesh model and smooth surface model.

4.3 Path Planning and Bioprinting Examples

Various path planning examples and their bioprinting outcomes are listed for each five main path planning and topology optimization methods explained before. The proposed techniques have been implemented with Rhinoceros 4.0 [30], using Rhino Script and Visual Basic programming languages. Observing the shape formations of the bioprinted structures, it can be stated that path planning methodologies are highly effective in generating biomimetic representatives of the blood vessels. In other words, the results show that multicellular aggregates and their support structures can be bioprinted biomimetically to form the biomodeled tissues.

First, an eight-layered aortic vascular construct piece is biomodeled, with 9 millimeters in diameter and 3.5 millimeters long [20], as shown in Figure 4.4, and it is bioprinted, using MEF cells as bio-ink (Figure 4.5), layer-by-layer with self-supporting method that is proposed in Section 3.2. This method is developed for non-branched vascular constructs.

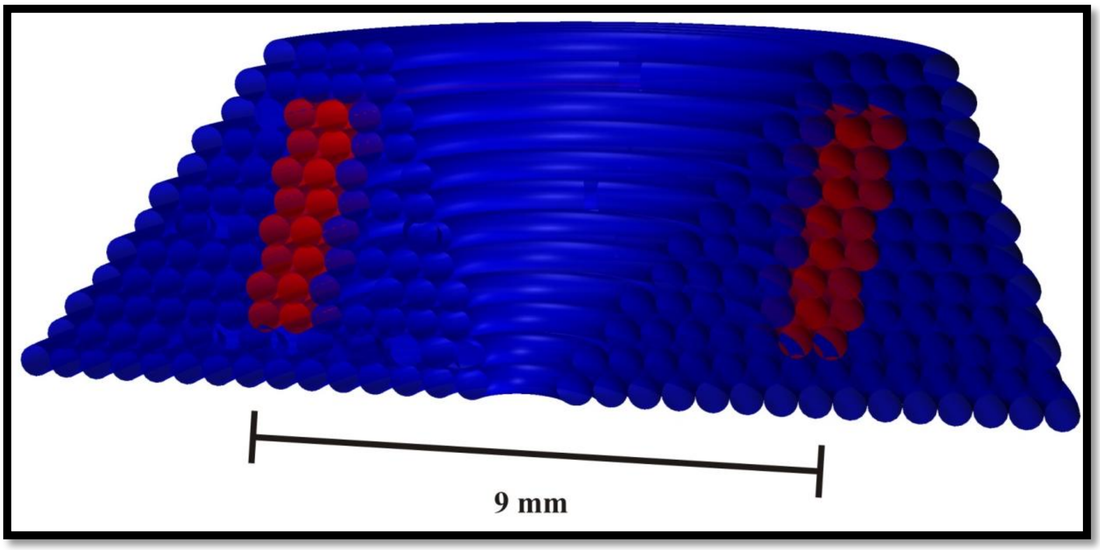

Figure 4.4. The cross sectional path planning view of the aortic model with support structures (blue) generated with respect to the self-supporting model and cellular aggregates (red).

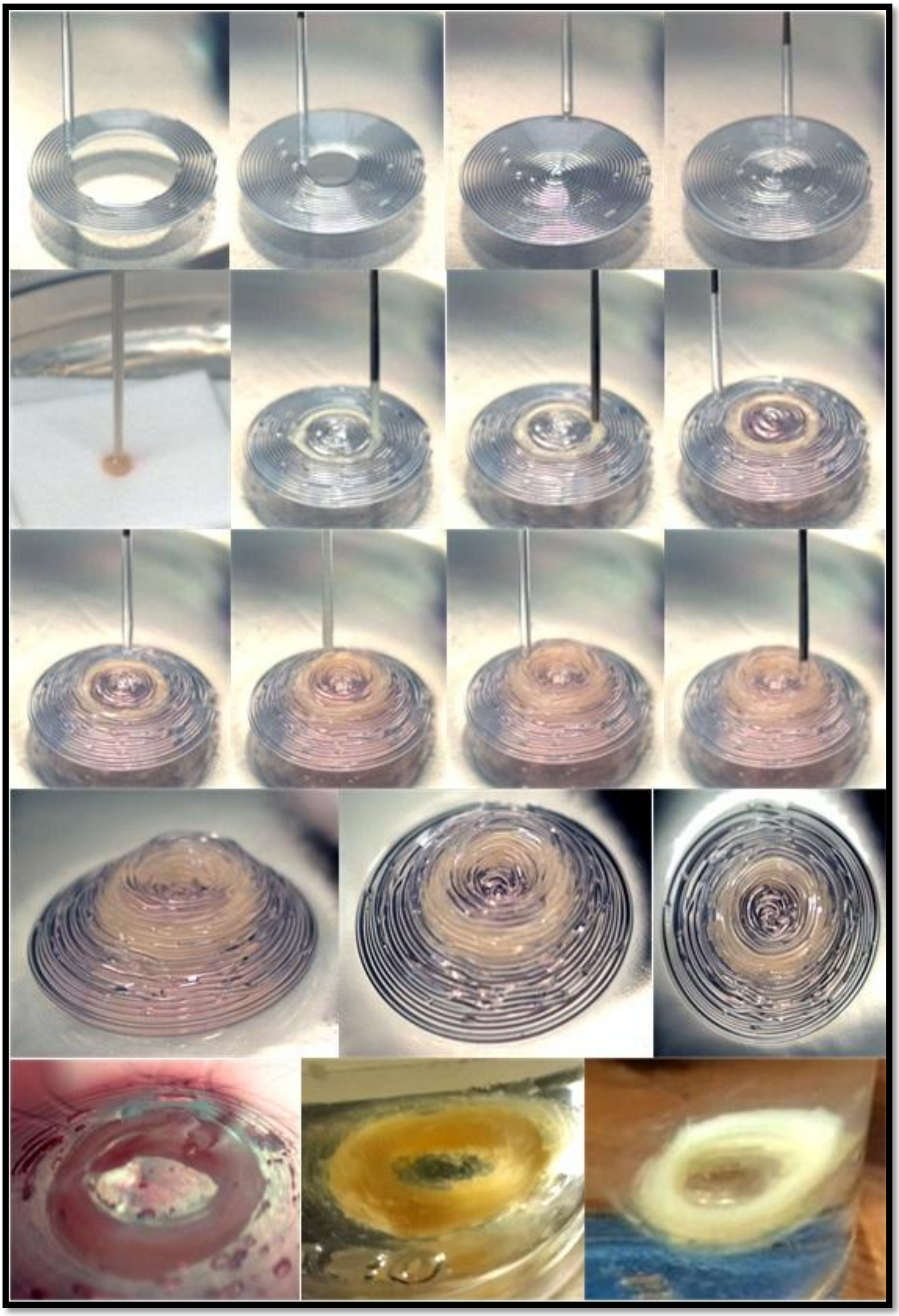

Figure 4.5. 3D printed MEF cell aggregates of originally mimicked aorta model with self-supporting path plan.

Three consecutive layers of the branching part of biomimetically biomodeled coronary artery (Figure 4.6), is printed using hydrogel and red colored hydrogel pair (replicating bio-ink) as shown in Figure 4.7.

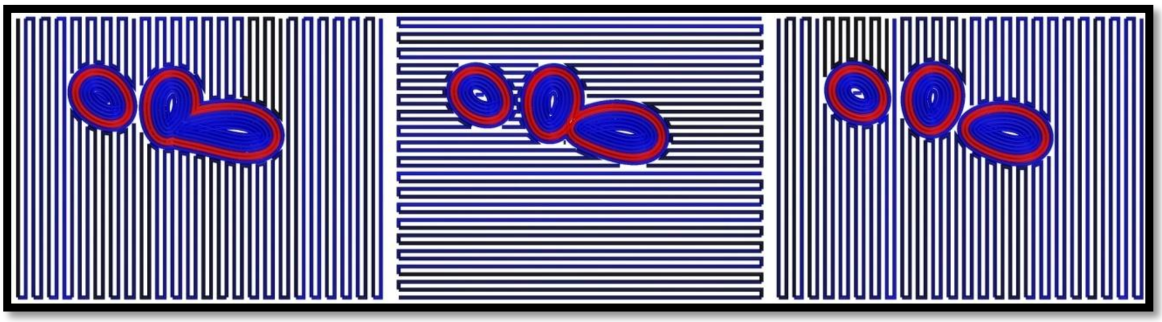

Figure 4.6. Path planning of a coronary artery model for three consequent layers, with zig-zag method.

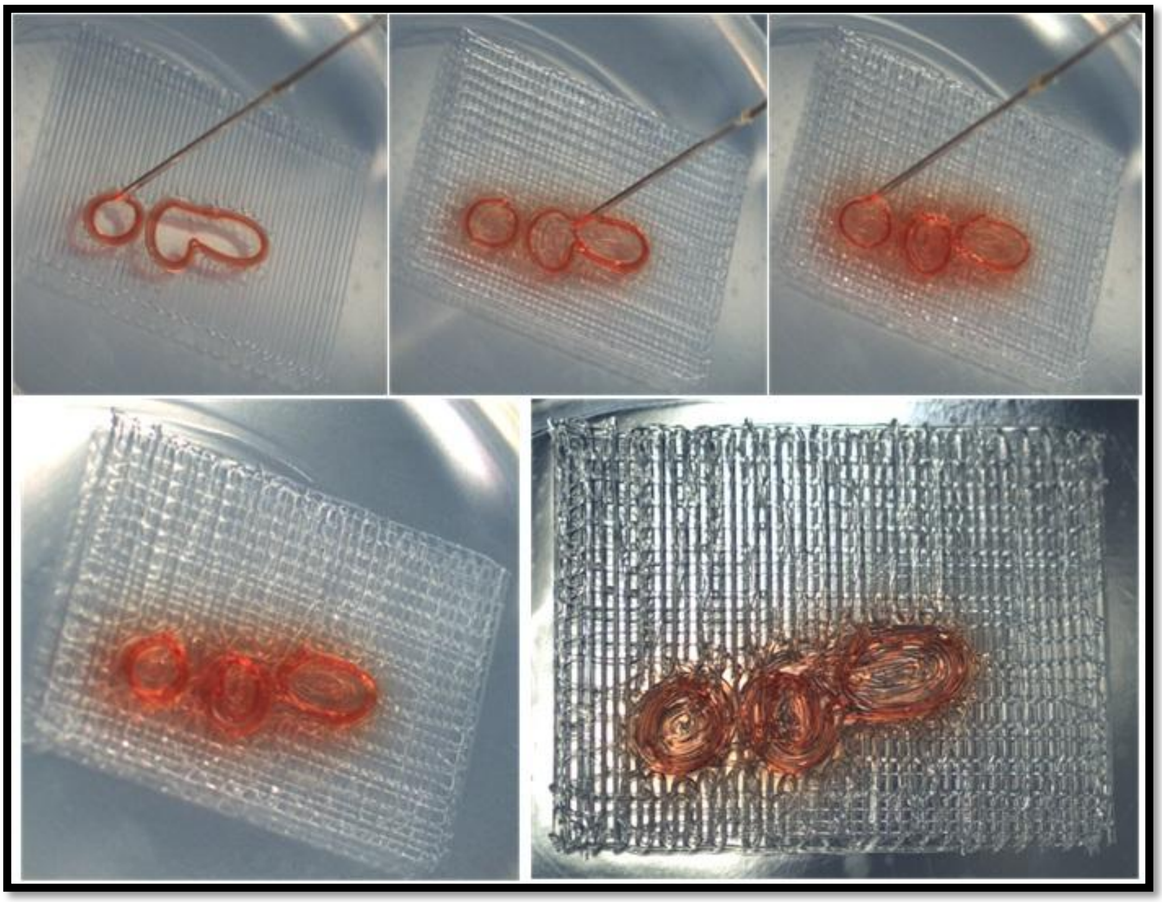

Figure 4.7. 3D printed layers of the coronary artery model, with zig-zag method.

To demonstrate further improvements and its capabilities in printing branched vascular structures with self-supporting method, an eight-layered, 3.5 millimeter diameters each and 3.5 millimeters long two- branched structure is modeled (Figure 4.8), and printed with hydrogels as supportive structure and MEF cells as bio-ink, as shown in Figure 4.9.

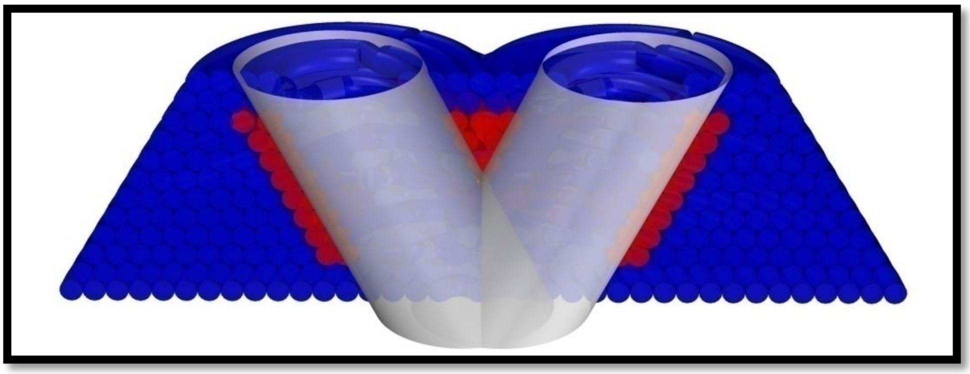

Figure 4.8. The cross sectional path planning view of the branched vascular model (grey) with support structures (blue) generated with respect to the self-supporting model and cellular aggregates (red).

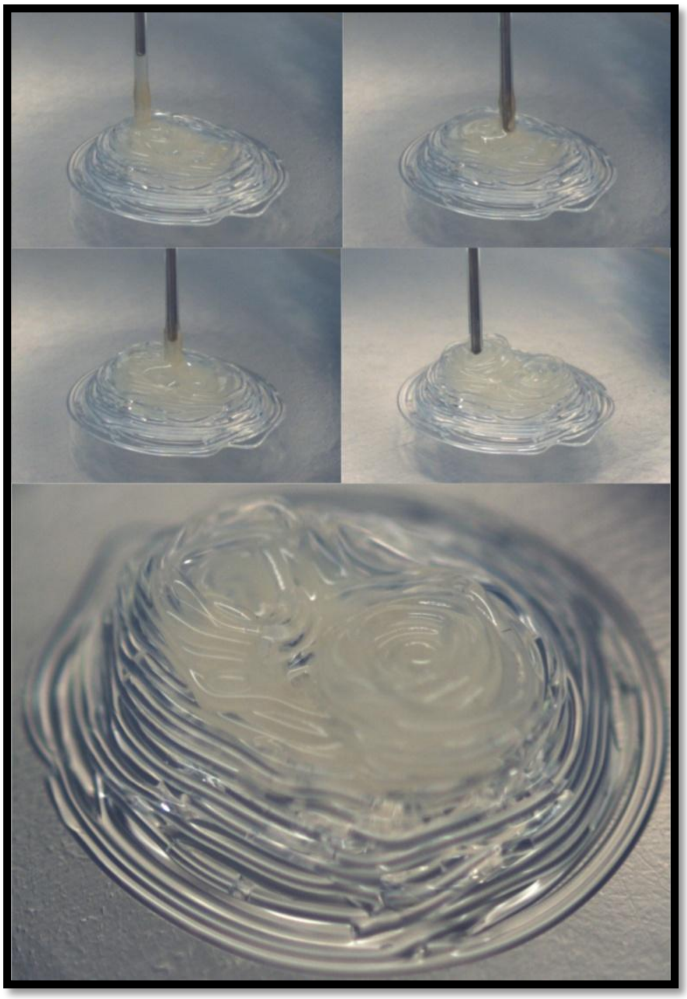

Figure 4.9. 3D printed MEF cell aggregates of branched vascular model in self-supporting model.

The fourth path planning example group is generated with hybrid method, as discussed in Section 3.6; this method is developed to overcome the time issue that exists in self-supporting method and provides material and time gain for consequent vertical layers, as can be observed by the vertical parts of the biomodeled vascular construct and their support structure width in Figure 4.10. Even though, hybrid method provides 40% time gain over self-supporting method. Moreover, in Figure 4.11, a simple vertical extruded surface is modeled with 4 millimeters diameter and 2 centimeters long, and printed with hydrogels.

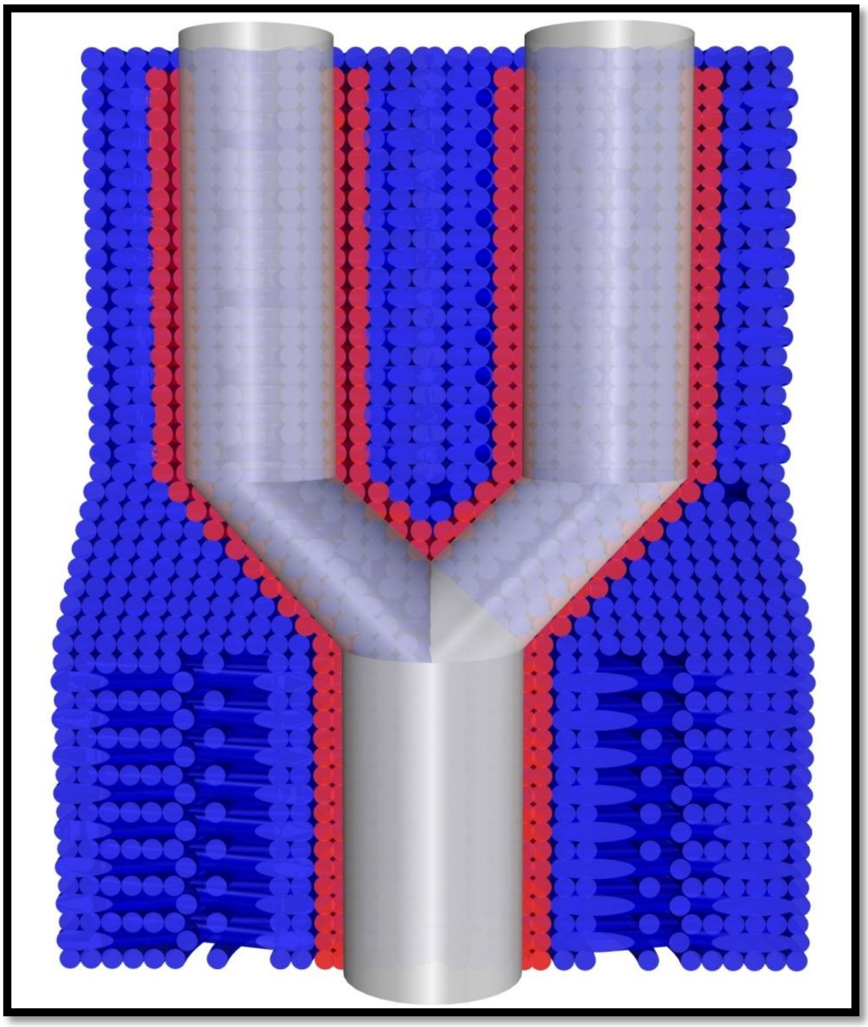

Figure 4.10. The cross sectional path planning view of the branched vascular model (grey) with support structures (blue) generated with respect to the hybrid printing model and cellular aggregates (red).

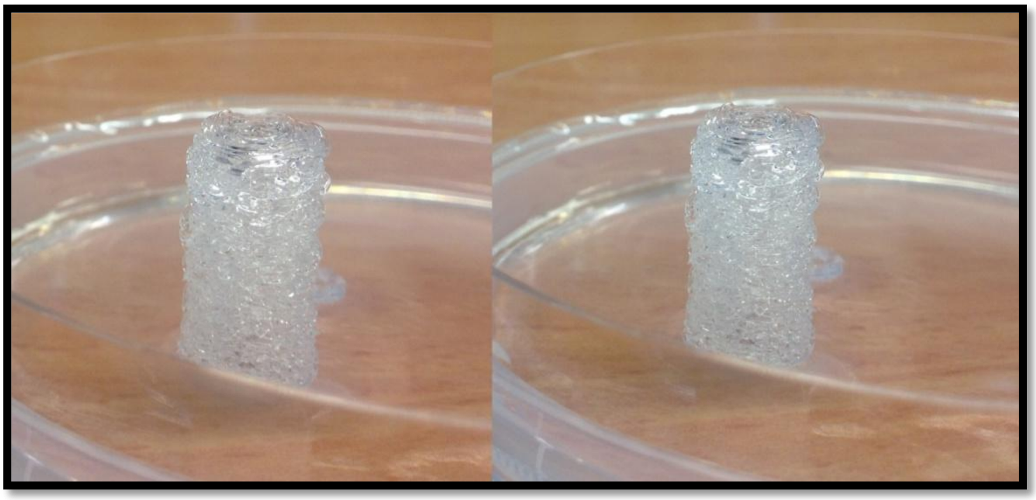

Figure 4.11. 3D printed non-branched vertical vascular construct with hybrid printing method.

Lastly, a seven-layered “Y” shape two-branched vascular construct is modeled with 1.2 millimeters radius (Figure 4.12). The horizontal path planning method is used for that example, while printing is done with blue colored, red colored and plain hydrogels, as shown in Figure 4.13. This approach is suitable for long and highly branched vessels, as bioprinting those kind of structures in vertical way results in both material and time waste, which is an issue for cell viability. Horizontal printing approach provides nearly 30% time gain over hybrid method for the same example model.

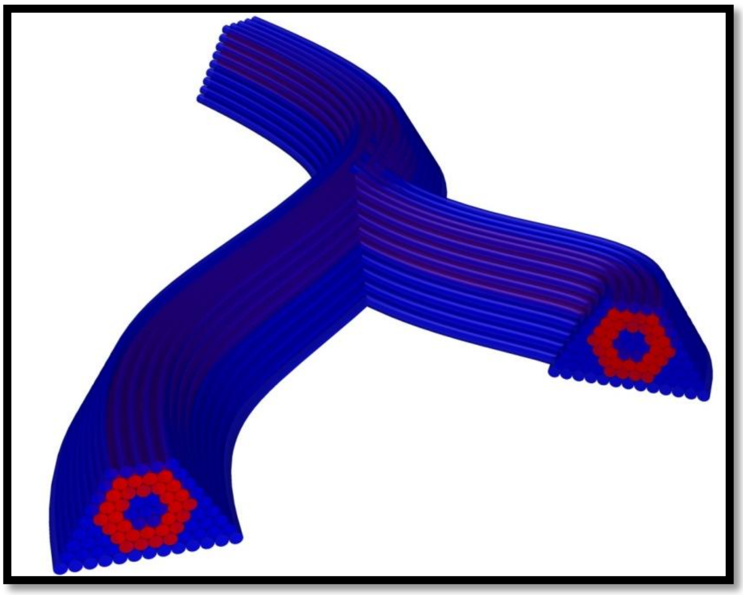

Figure 4.12. The path planning of a two branched vascular model with support structures (blue) generated with respect to the horizontal printing model and cellular aggregates (red).

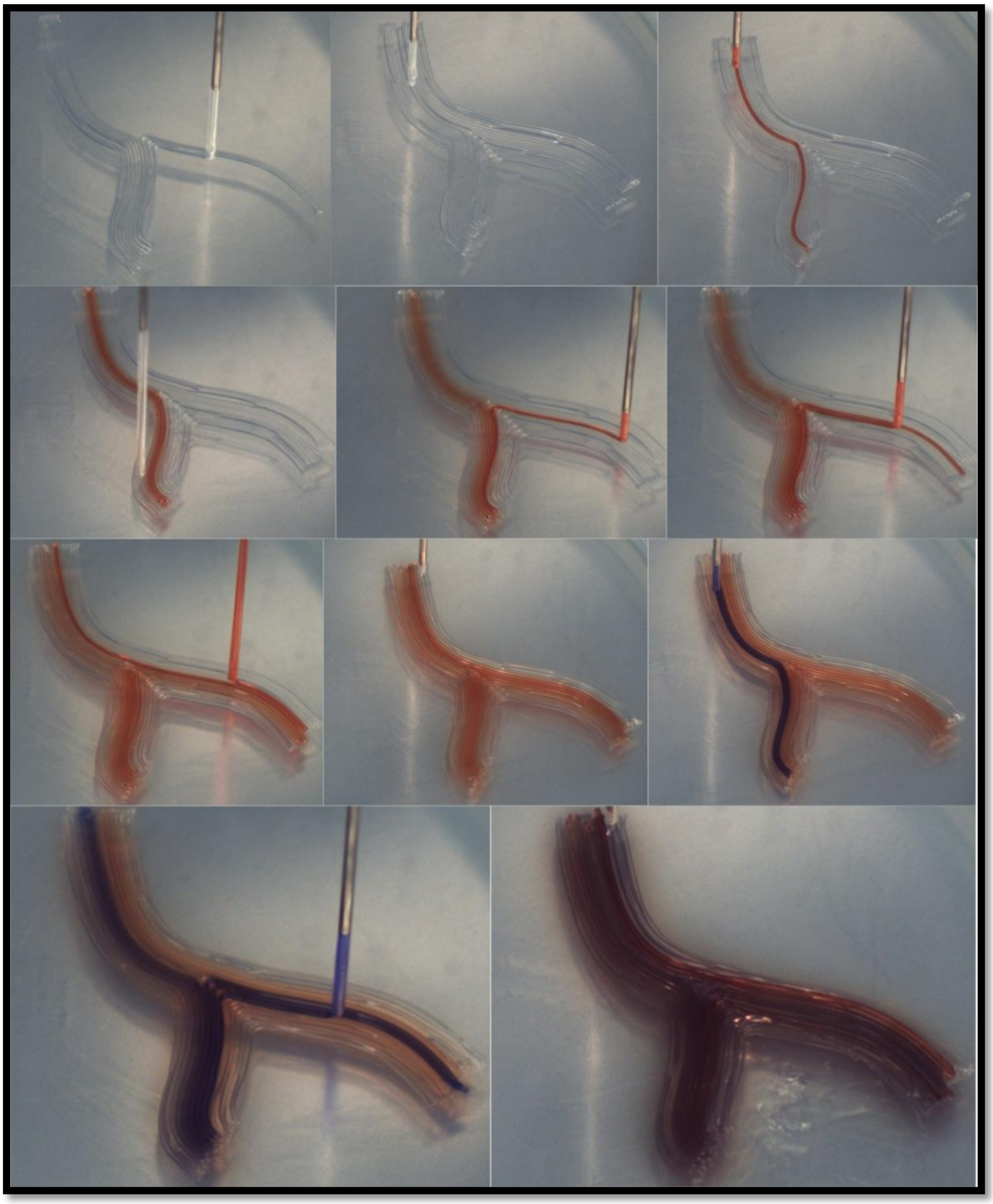

Figure 4.13. 3D printed two branched vascular construct with horizontal printing method. Colored hydrogels represent the cellular aggregates.

Chapter 5

Conclusions and Future Study

Increasing organ failures and cardiovascular diseases in recent years require alternative treatment procedures. Tissue engineering is one of the promising alternatives for development of biological substitutes. Especially, with recent advancements in bioprinting, 3D tissue constructs can be printed layer-by-layer using live cells and biomaterials.

In this thesis, novel biomodeling and path planning for 3D bioprinting are proposed. Scaffold-free macro-vascular structures are biomimetically printed with live cells and support biomaterials directly with the developed path planning algorithms. First, medical images of the desired blood vessels are segmented and stored as a STL (mesh) file to effectively mimic anatomic information of the blood vessel. Then, those mesh models are converted into smooth parametric surfaces by developed novel biomodeling algorithms. In order to bioprint anatomically correct vascular structures, mechanically weak cellular aggregates should be supported perfectly at each layer. Therefore, several support structure generation algorithms are also developed, namely zig-zag, self-supporting, hybrid and vertical bioprinting. Those methodologies use directly the biomimetically biomodeled surface representation of the desired blood vessels. However, as the horizontal boundaries and the area that the model occupies cannot be controlled by biomimetic biomodeling, material use and therefore the printing time increases for each layer. In order to favor cell viability, the total printing time needs to be optimized to minimum level. Moreover, the generated path plans may not guarantee cell contact between layers, as the slopes of the biomodeled structures or the separation angles of branches might exceed the maximum allowable slope for cell fusion between vertical layers. To overcome those drawbacks two new path planning methodologies are developed while preserving the original branching topology, separation angles and length information of the desired blood vessel. The self-supporting and hybrid methods are combined to bioprint cellular aggregates in vertical direction, by orienting radial group of cellular aggregates on top of each other for each layer. The motivation of this vertical bioprinting approach is that vascular cells lies in radial direction to form the natural blood vessels with ring like cellular structures. The second path planning methodology aims to generate horizontal path planning based on native vessels branching topology and length information. This approach is suitable for long and highly branched vessels. Overall, the main aim of these algorithms is to make bioprinted cellular aggregates conserve their 3D forms according to the planned model, while providing sufficient conditions for cell fusion. The bioprinting results with various size, diameter and branched vascular models show that multicellular aggregates as well as their support structures can be bioprinted layer-by-layer to form anatomically correct substitutes of the biomodeled tissues.

After the bioprinting process, the printed structure needs to be matured in an incubator with an appropriate medium up to 7-10 days. Then, the structure is carefully separated and cleaned from its supportive hydrogel walls and then placed into a bioreactor. The transfer to bioreactor enables printed cells of the vessel construct to fuse and further maturation with the flow medium. This engineered bioreactor mimics the biologically active environment for the bioprinted blood vessel and improves its mechanical strength over time. Moreover, it increases the burst pressure of the bioengineered vessel while providing satisfactory suturability level. For the future study, a bioreactor can be developed for further maturation of the printed vacular constructs. A continuous deposition bioprinter can also be developed to bioprint all the support biomaterials and cellular aggregates once at a layer. Therefore, with this enhancement, bioprinting of complex tissues and even organs could be possible with the proposed path planning algorithms.

Bibliography

[1] Nemeno-Guanzon JG, Lee S, Berg JR, Jo YH, Yeo JE, Nam BM, Koh Y-G, Lee JI. 2012. Trends in tissue engineering for blood vessels. J. Biomed. Biotechnol. 2012:14.

[2] Langer R, Vacanti JP. Tissue engineering. Science. 1993;260(0036-8075 (Print)):920-6.

[3] Pollok JM, Vacanti JP. Tissue engineering. Semin Pediatr Surg. 1996(1055-8586 (Print)):191-6.

[4] Kucukgul C, Ozler B, Karakas HE, Gozuacik D, Koc B. 2013. 3D Hybrid Bioprinting of Macrovascular Structures. Procedia Engineering 59(0):183-192.

[5] Wüst S, Müller R, Hofmann S. Controlled Positioning of Cells in Biomaterials—Approaches Towards 3D Tissue Printing. Journal of Functional Biomaterials. 2011;2(3):119-54.

[6] Melchels FPW, Domingos MAN, Klein TJ, Malda J, Bartolo PJ, Hutmacher DW. Additive manufacturing of tissues and organs. Progress in Polymer Science. 2012;37(8):1079-104.

[7] Koç B, Hafezi F, Ozler SB, Kucukgul C. Bioprinting-Application of Additive Manufacturing in Medicine. In: Bandyopadhyay A, Bose S, editors. Additive Manufacturing: CRC Press; in press.

[8] Tibbitt MW, Anseth KS. 2009. Hydrogels as extracellular matrix mimics for 3D cell culture. Biotechnology and Bioengineering 103(4):655-663.

[9] Nakayama K. In Vitro Biofabrication of Tissues and Organs. In: Forgacs G, Sun W, editors. Biofabrication: Micro- and Nano-fabrication, Printing, Patterning and Assemblies: Elsevier; 2013. p. 1-16.

[10] Matsuda N, Shimizu T, Yamato M, Okano T. Tissue Engineering Based on Cell Sheet Technology. Advanced Materials. 2007;19(20):3089-99.

[11] Khoda AKM, Ozbolat IT, Koc B. 2011. A functionally gradient variational porosity architecture for hollowed scaffolds fabrication. Biofabrication 3(3):034106.

[12] Khoda AKM, Ozbolat IT, Koc B. 2013. Designing heterogeneous porous tissue scaffolds for additive manufacturing processes. Computer-Aided Design 45(12):1507-1523.

[13] Ozbolat IT, Yin Y. Bioprinting Toward Organ Fabrication: Challenges and Future Trends. Biomedical Engineering, IEEE Transactions on. 2013;60(3):691-9.

[14] Xu C, Chai W, Huang Y, Markwald RR. Scaffold-free inkjet printing of three-dimensional zigzag cellular tubes. Biotechnology and Bioengineering. 2012;109(12):3152-60.

[15] Marga F, Jakab K, Khatiwala C, Shepherd B, Dorfman S, Hubbard B, Colbert S, Gabor F. 2012. Toward engineering functional organ modules by additive manufacturing. Biofabrication 4(2):022001.

[16] Billiet T, Vandenhaute M, Schelfhout J, Van Vlierberghe S, Dubruel P. A review of trends and limitations in hydrogel-rapid prototyping for tissue engineering. Biomaterials. 2012;33(26):6020-41.

[17] Norotte C, Marga FS, Niklason LE, Forgacs G. 2009. Scaffold-free vascular tissue engineering using bioprinting. Biomaterials 30(30):5910-5917.

[18] Faulkner-Jones A, Greenhough S, King JA, Gardner J, Courtney A, Shu W. 2013. Development of a valve-based cell printer for the formation of human embryonic stem cell spheroid aggregates. Biofabrication 5(1):015013.

[19] Mironov V, Kasyanov V, Markwald RR. Organ printing: from bioprinter to organ biofabrication line. Curr Opin Biotechnol. 2011;22(5):667-73.

[20] Kucukgul Can, Ozler Saime Burce, Inci İlyas, Irmak Ster, Gozuacik Devrim, Taralp Alpay, et al. 3D Bioprinting of Biomimetic Aortic Vascular Constructs with Self-Supporting Cells. Bioengineering and Biotechnology. submitted in 2014.

[21] Sun W, Starly B, Nam J, Darling A. Bio-CAD modeling and its applications in computer-aided tissue engineering. Computer-Aided Design. 2005;37(11):1097-114.

[22] Chen Z, Su Z, Ma S, Wu X, Luo Z. Biomimetic modeling and three-dimension reconstruction of the artificial bone. Comput Methods Programs Biomed. 2007;88(2):123-30.

[23] Wu S, Liu X, Yeung KWK, Liu C, Yang X. Biomimetic porous scaffolds for bone tissue engineering. Materials Science and Engineering: R: Reports. 2014;80(0):1-36.

[24] Li X, He J, Liu Y, Zhao Q, Wu W, Li D, et al. Biomaterial Scaffolds with Biomimetic Fluidic Channels for Hepatocyte Culture. Journal of Bionic Engineering. 2013;10(1):57-64.

[25] Human Coronary Artery [30.08.2014]. Available from: http://www.promocell.com/fileadmin/promocell/Kapitelbilder/Human_Coronary_Artery_Smooth_Muscle_Cells_2.jpg.

[26] Human Body Anatomy [30.08.2014]. Available from: http://www.cea1.com/anatomy-sistems/aorta-and-arteries/.

[27] http://www.materialise.com/mimics, *Mimics*. 2008.

[28] Piegl L, Tiller W. 1997. The NURBS Book: Springer. 646 p.

[29] Christopher MO, Francoise M, Gabor F, Cheryl MH. 2013. Biofabrication and testing of a fully cellular nerve graft. Biofabrication 5(4):045007.

[30] Rhinoceros 4.0, Robert McNeel & Associates, Seattle., http://www.rhino3d.com.