Published in final edited form as:

Proc SPIE Int Soc Opt Eng. 2016 February 27; 9786: . doi:10.1117/12.2216123.

Rapid Virtual Stenting for Intracranial Aneurysms

Liang Zhao^{1,†}, Danyang Chen^{1,†}, Zihe Chen¹, Xiangyu Wang¹, Nikhil Paliwal¹, Jianping Xiang¹, Hui Meng¹, Jason J. Corso², and Jinhui Xu¹

Liang Zhao: lzha06@buffalo.edu; Danyang Chen: danyangc@buffalo.edu; Zihe Chen: zihechen@buffalo.edu; Xiangyu Wang: xiangyuw@buffalo.edu; Nikhil Paliwal: npaliwal@buffalo.edu; Jianping Xiang: jxiang2@buffalo.edu; Hui Meng: huimeng@buffalo.edu; Jason J. Corso: jjcorso@eecs.umich.edu; Jinhui Xu: jinhui@buffalo.edu

¹University at Buffalo

²University of Michigan

Abstract

The rupture of Intracranial Aneurysms is the most severe form of stroke with high rates of mortality and disability. One of its primary treatments is to use stent or Flow Diverter to divert the blood flow away from the IA in a minimal invasive manner. To optimize such treatments, it is desirable to provide an automatic tool for virtual stenting before its actual implantation. In this paper, we propose a novel method, called ball-sweeping, for rapid virtual stenting. Our method sweeps a maximum inscribed sphere through the aneurysmal region of the vessel and directly generates a stent surface touching the vessel wall without needing to iteratively grow a deformable stent surface. Our resulting stent mesh has guaranteed smoothness and variable pore density to achieve an enhanced occlusion performance. Comparing to existing methods, our technique is computationally much more efficient.

1 INTRODUCTION

Intracranial Aneurysm (IA) is a vascular disease affecting 5% of the US population. The rupture of IA could lead to subarachnoid hemorrhage, the most devastating form of stroke with high rates of mortality and disability. One of the increasingly popular minimal invasive treatments for such a disease is stenting. A stent or Flow Diverter (FD) is a densely braided self-expandable mesh deployed across an aneurysmal region to reduce the inflow. The flow modification induced by stent leads to IA thrombosis, parent vessel reconstruction, and obliteration of the IA. Although stent/FD has considerably high success rates, post-treatment complications, including incomplete or prolonged occlusion and post-FD rupture, are still unresolved challenging issues with this kind of treatments. To further improve such treatments, it is desirable to optimize them by virtually simulating stenting before its actual implantation.

In this paper, we propose a novel method, called *ball-sweeping*, to compute a simplex mesh [4] of a virtual stent¹. Our method is a generalization of the popular plane sweeping method in Computational Geometry. It first sweeps a maximal inscribed sphere through the portion

[†]Equal contribution first authors.

¹We follow the terminology in [4] by calling the quadrilateral mesh as simplex mesh.

of the vessel bearing the aneurysm; the loci of the boundary of the sweeping sphere form a surface (called *sweeping surface*) touching the vessel wall from its interior. Similarly, the locus of the center of the sweeping sphere becomes a centerline of the vessel. Since the maximal inscribed sphere is not unique at a given location of the vessel, our method then uses dynamic programming to produce a centerline with the best smoothness. This enables us to indirectly generate a smooth sweeping surface. Finally, our method generates a simplex mesh of a virtual stent on the sweeping surface. The resulting stent mesh has variable density, depending on its relative location to the aneurysm and the curvature, which helps improving the occlusion performance.

1.1 Previous Work

Several results exist for this problem [10, 3, 12], which adopt one of the following two main approaches. The first approach is based on Finite Element Analysis (FEA) [10]. It uses FEA to model the whole procedure of stent deployment in a patient-specific aneurysm geometry to obtain realistic stent configuration. The second approach is based on the idea of adaptive expansion [3, 12, 9]. It first initializes the stent around the centerline of the vessel, and then repeatedly expands the stent using some pseudo internal and external forces until the stent touches the vessel wall. Despite their various advantages, these methods share two common issues. The first issue is that they all heavily depend on collision detection to ensure that the stent stays inside the vessel wall. This often leads to high time complexity as repeatedly performing collision detection could be rather time consuming. The second issue is that they need an initial configuration or centerline as their input. This could further slow down their execution on raw data sets. Recently, there is also a more efficient method [11], but the stent always occludes to the vessel wall in the method, which is not realistic.

1.2 Our Contribution

Aiming to fix the main issues of the existing methods, our ball sweeping method completely avoids collision detection. It computes some Voronoi vertices to directly generate the center of the maximal inscribed sphere. Also our method does not need to adjust the stent iteratively². These together make our method have the following advantages.

- **Fast:** Since our method does not require collision detection and iteratively modifying the stent, it is considerably faster than existing methods. For example, it takes only a couple of minutes for typical aneurysm data sets, while existing methods could take half hour or even tens of hours.
- **Smooth:** Our resulting stent mesh is guaranteed to be smooth, while none of the existing methods optimizes the smoothness of the stent surface.
- **Realistic:** Our generated stent mesh has variable density at different locations, depending on their distance to the aneurysm and their curvatures. This enables us to achieve better occlusion performance.

²Note that although our approach shares with [1, 6] the idea of using Voronoi diagram. There are major differences. Both [1] and [6] need to generate the whole Voronoi diagram of the vessel. But our approach only needs to generate part of the Voronoi diagram necessary for the ball sweeping, thus achieving better time efficiency.

- **No Initialization:** Our method does not need any initial configuration or centerline as its input. It solves both centerline and stenting problem. We do need to manually decide the stent is deployed in which part of vessel.

2 THE BALL SWEEPING METHOD

In this section, we present our ball sweeping method for virtual stenting.

Let T be a triangular mesh of the vessel wall obtained from the 3D image of the aneurysm. As mentioned earlier, our objective is to obtain a fast virtual stenting method which is capable of producing smooth simplex stent mesh. To reach this goal, we need to avoid the iterative collision detection computation and yet still ensure that the resulting stent mesh is always inside the vessel wall. These are two seemingly contradicting expectations, as the main purpose of the iterative computation is to ensure that the stent stays inside the vessel wall.

To overcome this difficulty, our main idea is to generalize the plane sweeping technique in Computational Geometry [2] to a ball sweeping technique inside the vessel. To shed some light on this, consider placing an elastic ball inside the vessel. To reach a stable status (*i.e.*, with minimum energy), the elastic ball will be maximally inscribed inside the vessel. Now imagine that the elastic ball is dragged (or sweeps) through the region of the vessel containing the aneurysm. At each time point of this sweeping process, the elastic ball is always maximally inscribed inside the vessel wall to minimize its energy. This means that the ball always touches the vessel wall and stays inside the vessel. The loci of each point on the boundary of the ball form a continuous curve. The union of all such curves, one for every point of the boundary, forms a cylinder-like structure maximally touching the vessel wall. The boundary of such a structure is a pipe-like surface, called sweeping surface, touching the vessel wall. If we are able to embed the stent mesh on this surface, then we not only avoid the iterative computation of collision detection, but also ensured the stent mesh staying inside the vessel. See Figure 2 for an example of this sweeping process.

Implementing this approach is actually not easy. This is because sweeping is a continuous process and our computation can only be discrete. To resolve this issue, we notice that the center of the maximal inscribed sweeping ball is either a Voronoi vertex of the medial axis of the triangular vessel mesh or a point on a Voronoi edge [2]. Thus our idea for emulating the continuous sweeping process is to just compute the center of the sweeping ball at those Voronoi vertices, and sweep the ball along the Voronoi edge between each consecutive pair of Voronoi vertices. This means that we need a sweeping direction at each step for determining the next Voronoi vertex. Also due to the existence of aneurysm, we need to avoid the sweeping ball goes inside the aneurysm, and ensure that the resulting stent mesh is smooth. To address all these issues, we propose to use the following main steps for the sweeping process.

1. Compute an initial centerline.
2. Use the initial centerline to generate a real smooth centerline.
3. Generate the simplex mesh of the stent.

2.1 Initial Centerline Estimation

In this section we show how to efficiently obtain an initial centerline. The main purpose of generating an initial centerline is for giving a rough estimation of the sweeping direction. Although we can compute an exact centerline to achieve the same purpose, this will require the information of the exact locations of all vertices (*i.e.*, the global information) of the triangular vessel mesh and perform the computation in the geometric domain, which could be quite time consuming. A more efficient way is to perform the computation in the graph (*i.e.*, the triangular mesh) domain, and use local computation to approximate the real centerline. Particularly, we first determine a guideline and then use it to guide us to find an exact and smooth centerline whose computation depends only on local information. A byproduct of the initial centerline is that it avoids the sweeping ball going into the aneurysm sac.

To find such an initial centerline, we first compute a shortest path SP as the first estimation of the centerline on the triangular mesh (vessel) from one end of the vessel to the other. This implicitly ensures that the centerline does not go into the aneurysm sac (see Figure 1b). Then for each vertex of SP , we compute a shortest path ring (on the triangular mesh) around the vessel (see Figure 1c). The center of this ring is an initial position of the centerline, and the union of all these center points forms the initial (or the second estimation of the) centerline. Computing the initial centerline in this way is fast as its computation is purely on the triangular mesh and uses only local information.

2.1.1 Shortest Path Between Two Ends of the Vessel—To compute the shortest path SP , we add a dummy vertex s (or t) at each end of the vessel, and connect an edge from each vertex at the end of the triangular mesh T to the corresponding dummy vertex with 0 edge weight. Then, a shortest path SP between s and t is computed in T using Dijkstra algorithm [5], where the weight of each edge of T is its simply length (see Figure 1b).

2.1.2 Shortest Path Ring—To compute a shortest path ring $Ring_v$ for each vertex v of SP , we first cut the triangular mesh T along the shortest path SP , and duplicate every vertex of SP so that there is a copy on each side of the cut. This opens up the vessel wall and we denote the new mesh as T' . Then the shortest path ring $Ring_v$ of v is obtained by computing the shortest path between v and its copy v' in the T' using Dijkstra algorithm. By the nature of Dijkstra algorithm, we know that it stops once v' is reached. Thus it is a local computation. After obtaining $Ring_v$, we compute the mean point of all vertices of $Ring_v$ as its approximate center. The set of these centers is denoted as M_0 .

2.1.3 Virtual Ring and Sweeping Direction—After generating the set M_0 of center points of the shortest path rings, we can connect every consecutive pair of them to form an approximate centerline. However, it is possible that the two consecutive center points m_{p1} and m_{p2} may be rather far away from each other. This could significantly affect the quality of the centerline. To resolve this issue, we can add some virtual rings between m_{p1} and m_{p2} in the following way. We first sample cnt_1 points $S_{p1} = \{s_{i,1} | i = 1 \dots cnt_1\}$ from $Ring_{p1}$ and cnt_1 points $S_{p2} = \{s_{i,2} | i = 1 \dots cnt_1\}$ from $Ring_{p2}$. Then, we compute a shortest path $SP_{s_{i,1}s_{i,2}}$ from $s_{i,1}$ to $s_{i,2}$ along the vessel, and sample cnt_2 points on each shortest path with equal

distance between consecutive points (denoted as $Sample_i$). Let m_j be the mean point of the j th points in each $Sample_i$, and M_1 be their union. Then, an improved initial centerline can be obtained by connecting consecutive pair of points in $M = M_0 \cup M_1$.

For each $m_j \in M$, we compute a sweeping direction d_j for the sweeping ball. To obtain a robust direction for m_j , we first compute a cubic spline interpolation for points in M along the vessel [8], and then compute the gradient of the spline curve at m_j as the sweeping direction d_j .

2.2 Generating a Real Smooth Centerline

With the initial centerline and sweeping direction, one may tempt to generate a real centerline by detecting the Voronoi vertex closest to these center points in M . However, since the centerline is not unique, the resulting centerline may contain spikes, especially at the region near the aneurysm (see Figure 1a), which could affect the flow diverting ability of the stent. To resolve this issue, we need to find a centerline with the best smoothness by performing a global optimization on all possible centerlines. Below, we first identify a set of candidate Voronoi vertices for each center point in M , and then use dynamic programming to determine the best centerline.

2.2.1 Identifying Voronoi Vertex Candidates for Each Center Point—To

identify the set of Voronoi vertices near a center point, we first compute the Voronoi diagram $VD(T)$ of all vertices in T^3 . Then, for each $m_j \in M$, we identify as its candidates a set $C_j = \{c_{ij} | j = 1 \dots n_1\}$ of Voronoi vertices of $VD(T)$ whose orthogonal projection distance to m_j along the sweeping direction d_j is within a given threshold distance to m_j .

2.2.2 Computing a Smooth Centerline—To obtain a smooth centerline $\{c_1, \dots, c_n\}$,

we use bending energy to measure the smoothness of our centerline. Since the centerline is actually a polygonal curve, we can simplify the bending energy as the summation of the square of all angle changes along the centerline.

$$E_{\text{bend}} = \sum_{i=2}^{n-1} \theta_i^2, \quad (1)$$

where θ_i is the angle formed by the two consecutive line segment intersecting at c_i .

In Eq (1), each θ_i depends only on three points c_{i-1} , c_i and c_{i+1} . It is equivalent to computing an MAP (Maximum a Posteriori) estimation for a 2nd-order Hidden Markov Model. Below we propose a dynamic programming method for this problem, following the spirit of Extended Viterbi Algorithm [7].

We minimize the following objective function by choosing one point $c_i \in C_j$ with corresponding radius r_i .

³Note that we use the Voronoi diagram of the vertices rather than the medial axis of the triangular mesh T for better time efficiency, since medial axis construction could be much slower and instable. Our experimental result shows that the relative error of the two types of Voronoi vertices (on the radius of maximal inscribed sphere) is no more than 2%.

$$\min_{c_i \in C_i} w_1 \sum_{i=2}^n (r_i - r_{i-1})^2 + w_2 \sum_{i=2}^n \|c_i - c_{i-1}\|^2 + w_3 \sum_{i=1}^n (-r_i) + \sum_{i=2}^{n-1} \theta_i^2, \quad (2)$$

where $\|\cdot\|$ is the Euclidean norm. The first three terms are added to further improve the smoothness of the centerline. Term $\sum_{i=2}^n (r_i - r_{i-1})^2$ is for demanding that the radius of the maximal inscribed ball does not have rapid change at consecutive centers. This will implicitly ensure the smoothness of the stent mesh. Term $\sum_{i=2}^n \|c_i - c_{i-1}\|^2$ requires the distances between consecutive centers roughly the same. This will make the distribution of $\{c_i\}$ uniform along the centerline for better evenly distributed stent mesh. Term $\sum_{i=1}^n (-r_i)$ is for maximizing the total radius so as to ensure that resulting stent mesh touches the vessel wall. We solve the above optimization problem using the following Dynamic Programming method.

Let U, E be two arrays of matrices with size $n_i \times n_{i+1}$ for their i -th element $U[i]$ and $E[i]$. First, U is initialized as follows. $U[i][k, l]$

$$= \begin{cases} w_1(r_{ik} - r_{i+1,l})^2 + w_2\|c_{ik} - c_{i+1,l}\|^2 + w_3(-r_{ik}) + 0.5w_3(-r_{i+1,l}), & \text{if } i=1, \\ w_1(r_{ik} - r_{i+1,l})^2 + w_2\|c_{ik} - c_{i+1,l}\|^2 + 0.5w_3(-r_{ik}) + 0.5w_3(-r_{i+1,l}), & \text{if } 1 < i < n-1, \\ w_1(r_{ik} - r_{i+1,l})^2 + w_2\|c_{ik} - c_{i+1,l}\|^2 + 0.5w_3(-r_{ik}) + w_3(-r_{i+1,l}), & \text{if } i=n-1. \end{cases} \quad (3)$$

Then we compute E in the following way. $E[i][k, l]$

$$= \begin{cases} U[1][k, l], & \text{if } i=1 \\ U[i][k, l] + \min_j \{E[i-1][j, k] + (\theta(c_{i-1,j}, c_{ik}, c_{i+1,l}))^2\}, & \text{otherwise,} \end{cases} \quad (4)$$

where $\theta(c_{i-1,j}, c_{ik}, c_{i+1,l})$ is the angle formed by points $c_{i-1,j}, c_{ik}, c_{i+1,l}$. The minimum of Eq (2) is obtained by $\min_{i,j} E[n-1][i, j]$.

2.3 Generating the Stent Mesh

With the determined smooth centerline, we can now generate the stent mesh. We expect that the density of the stent mesh is higher in those regions either near the aneurysm or having a larger curvature. To achieve this goal, we add additional center points along the centerline in the region near the aneurysm or those having a larger curvature.

After we have the final center points, the stent mesh is computed by sampling points on a circle orthogonal to d_i with corresponding radius⁴. Then we connect the corresponding points on the consecutive circles to get the stent mesh.

3 RESULTS AND CFD ANALYSIS

To validate our ball sweeping method, we conduct two experiments. The first experiment is to use a computational flow dynamics simulation tool to simulate the blood flow and examine the occlusion performance of our generated virtual stent meshes for two real Aneurysm data sets. We deploy a commercial flow diverter (Pipeline Embolization Device, Covidien, Irvine, CA) on two representative patient-specific aneurysm geometries using our method (Aneurysm I and II). Aneurysm I is a side-wall aneurysm located at the internal carotid artery, and Aneurysm II is a fusiform aneurysm located at the basilar artery. The results of the deployment of the simplex mesh are shown in Figure 3a and 3c, respectively. We can see from the figure that the result shows good wall apposition of the simplex mesh. After deployment we extract 48 wires from the simplex mesh, and sweep the wires to give them 3D structure. These 3D wire structures are discretized into a stereolithographic format, used as input to flow simulation. We then perform computational fluid dynamic simulations on the “treated” aneurysms to analyze post-treatment hemodynamics. Figure 3b and 3d show the velocity streamlines for both aneurysm geometries. The streamlines show that post flow diverter treatment significantly reduce the flow of blood into the aneurysmal sac.

The second experiment is for running time comparison between our method and two representative methods, the finite element method [10] and the adaptive expansion method [9]. We implement both methods and compare them with our method on the above two aneurysm data sets using MATLAB. Table 1 shows the comparison result. From the table, it is clear that our method is significantly faster than the two existing methods. Note that for a fair comparison, we do not consider the method in [11] as it is not realistic.

4 CONCLUSION

In this paper, we propose a novel method called ball sweeping for virtual stenting. Our method is based on computational geometry technique and avoids the time-consuming iterative collision detection computation. Comparing to existing methods, our technique is faster, yields smooth and density-variable stent mesh, and needs no initialization (such as centerline) on the data set. Experimental results (based on computational flow dynamics simulator) show that the stent meshes generated by our method have good occlusion performance.

Acknowledgments

This research was partially supported by NIH through grant NIH 1R01NS091075.

REFERENCES

1. Antiga L, et al. VMTK: Vascular Modeling Toolkit. <http://www.vmtk.org>.
2. Berg, M.; Cheong, O.; Kreveld, M.; Overmars, M. Computational Geometry: Algorithms and Applications. Springer; 2008.

⁴If the curvature is too large, it is hard to keep the points on the circle in the same order of the centers. But as stated in the previous paragraph, when the curvature is too large, we sample more points on the centerline. Therefore, the directions of two consecutive centers do not vary much. This is very helpful to keep the order of the points on the the circles.

3. Cebal JR, Lohner R. Efficient simulation of blood flow past complex endovascular devices using an adaptive embedding technique. *IEEE Trans. Med. Imag.* 2005; 24(4)
4. Delingette H. General Object Reconstruction Based on Simplex Meshes. *International Journal of Computer Vision.* 1999; 32(2)
5. Dijkstra EW. A note on two problems in connexion with graphs. *Numerische Mathematik.* 1
6. Ford M, et al. An objective approach to digital removal of saccular aneurysms: Technique and applications. *Brit. J. Radiol.* 2009; 82
7. He Y. Extended Viterbi algorithm for second order hidden Markov process. 9th International Conference on Pattern Recognition. 1988; 2
8. Lee ETY. Choosing nodes in parametric curve interpolation. *Computer-Aided Design.* 1989; 21(6)
9. Larrabide I, et al. Fast virtual deployment of self-expandable stents: Method and in vitro evaluation for intracranial aneurysmal stenting. *Med. Image Anal.* 2010
10. Ma D, et al. Computer modeling of deployment and mechanical expansion of neurovascular flow diverter in patient-specific intracranial aneurysms. *Journal of Biomechanics.* 2012; 45
11. Paliwal N, et al. Fast Virtual Stenting With Vessel-Specific Initialization and Collision Detection. *ASME.* 2014
12. Valencia LF, Montagnat J, Orkisz M. 3D Graphical models for vascularstent pose simulation. *Innovations Technol. Biol. Med.* 2007; 28(2)

(a) The unsmooth centerline and stent

(b) The shortest path between two ends.

(c) Example of one guiding point.

Figure 1.
Some illustrations for this method.

Figure 2.
The Ball sweeping process.

Figure 3.
Aneurysm II

Table 1

The running time comparison

Method	Time
Finite Element Method	100 hours
Adaptive Expansion	27.5 minutes
Ball sweeping	134/451 seconds

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript