Mesh Optimization of Vessel Surface Model for Computer-Aided Simulation of Percutaneous Coronary Intervention

Fan Yang, Zeng-Guang Hou, Shao-Hua Mi, Gui-Bin Bian, and Xiao-Liang Xie

Abstract— **Percutaneous coronary intervention is the gold standard to coronary diseases in the past decades due to much less trauma and quick recovery. However, due to the traits of minimal invasiveness, clinicians have to defeat the difficulties in eye-hand coordination during the procedure, which also makes it a non-trivial task in the catheterization lab. The computer-aided surgical simulation is designed to provide a reliable tool for the early stage of the training of the procedure. In this simulation system, the surface model of the vessels contribute the major part in the virtual anatomic environment. On the other hand, heavy interactions between the virtual surgical tools and the model surface occur during the training. In order to achieve acceptable performances, the patient-specific vessel surface model needs further process to adapt to this situation. We proposed in this paper an approach to optimize the meshes that consist the surface model with its application in consideration. The connectivity of the surface model is firstly checked. Next a smooth processing is applied without modifying the geometry of the largest-connected surface. Then the quantities of the polygons consisting the model surface are eliminated both dramatically and appropriately. The resultant surface model is applied in the validation test interacting with the virtual guidewire.**

I. Introduction

Percutaneous coronary intervention (PCI) is the gold standard in fighting the coronary heart diseases, which is one of the most lethal diseases in the modern world [1]. Due to the minimally-invasive traits, this procedure causes smaller incision and much less trauma to the patients than open surgery performed in the past. However, this feature also makes it hard for the learners to get used to it. Besides, after entering the real catheterization labs, the clinicians have to endure the potential occupational hazards year after year [2].

In this context, robot-assisted intravascular system came into play with the aim of changing this situation. With these robots [2], [3], the work load of the cardiologists are dramatically lightened, and the risks of their health are greatly decreased. However, as their ancient manually performed counterparts, the technique of steering the robotic arms in surgery is still hard for the novice.

Traditional paradigm of PCI training strictly followed the physical fashion – performing demonstrative procedure on biological or non-biological models [4], [5]. The former materials, including unclaimed cadavers and live animals, are ethic-disputed. Moreover, the expenditure on the preservation and feeding is high-rising, and the distinction in anatomy between human and animal volunteer is apparent even for the amateur. The latter are rigid and stiff both in touching and looking.

In the case of training of robotic systems, one fact is that the advanced robotic systems can not be applied to the training for the health of the clinicians and the lives of the patients will be at risk. In order to solve this problem, better training vehicles are required. Thanks to the advancement of computer hardware, we could employ the greatly improved computer system as the platform, on which the dedicated computer-aided training simulation can be realized.

Our aim is to implement a computer-aided simulation training system for the minimally-invasive intravascular robotic system built in our lab [6]. The prototype consists of two main components: the display of the 3-D virtual anatomic simulation, and the manipulating hardware interface. In realizing this training platform, the temporal performance of interaction between the anatomic structures in computing environment (e.g., the vascular system) and the virtual surgical tools are undoubtedly of the primary concern. In order to improve this performance, the quantity of the consisting polygons needed to be greatly reduced.

In this paper, we developed an approach to substantially eliminate the total number of the polygonal elements of the vasculature model. As the most popular primitive for the modern computer graphics application, the polygons can be rendered in any standard commercial computer graphics systems. The fact is that in surface rendering, the more precise the results are, the more polygons are required. This makes the rendering of the huge amount of polygons occupy too much computation resources, especially the memories. In the field of medical visualization and simulation, models rendered using surface techniques require even more polygons to capture various of complex and delicate human anatomic structures. The algorithm introduced in our work is an application independent method to eliminate the polygons (especially triangles) by performing local operations [7].

The input data is the patient-specific surface model of the human vasculature based on the computed tomography angiography (CTA). Firstly, the connectivity of the polygons that consisting the surface is validated thoroughly to include the largest connected polygons that is effective in representing the surface. Secondly, the uneven surfaces are smoothed in order to reduce the unnecessary effects on the following decimation. Finally, numbers of the component polygons of the vessel model are eliminated. The capability of our approach in reducing the amount of polygonal components

This work was supported in part by the National Nature Science Foundation under Grants: 61225017, 61203342, 61203318.

The authors are with State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China fan.yang@ia.ac.cn, hou@compsys.ia.ac.cn

of the vessel model is proved by the experimental results.

The remainder of this paper is organized as follows. Section II outlines the work flow and describes the techniques introduced in this work. Section III describes the experiments and validates the results, ending with a discussion. The final section concludes the whole work and outlooks the future.

II. Methodologies

Some preprocessing steps illustrated in Fig. 1 are necessary before the elimination of the polygons in the surface. Firstly, the validation of the connectivity among the vertices in the image-based model surface are performed to guarantee that the surface to be processed is complete as a whole and all the vertices are connected. Secondly, the smoothing step is introduced in order to depress the bulges and sunken areas in the connected surface. Finally, the polygonal elements in the surface model are selected and part of them are deleted; the "holes" left are patched and the geometry of the model is preserved as complete as possible.

A. Connectivity Validation

Before any processing steps, one routine must be done to ensure that the surface model considered is well-shaped that no extra polygons exist in defining the same piece of surface. This routine is designed to extract the polygons based on the geometric connectivity. Concretely, it can extract the cells (i.e., polygons and/or triangles) that share the common points (i.e., vertices).

B. Surface Smoothing

In the field of computer graphics and imaging, the visualization of curves and surfaces (both will be referred to as *shapes*) both in two- or three-dimensional space are common, especially for the volumetric data from medical field. The computation is in fact the proper approximation of the original shapes by the polygonal curves or polyhedral surfaces. Typically, artifacts at certain level can be found in the resultant approximating shapes (see Fig. 2a), and can be smoothed while maintaining the overall geometry (see Fig. 2b). In this work, one of our objectives is the smoothing or fairing of the model surface acquired by applying the iso-intensity extraction techniques on the raw medical images. A signal processing approach is employed to fulfill this task. The approach is actually a linear low-pass filtering technique [8], which is a low-level processing in the field of digital signal processing. It can be applied to the smoothing process on the curves/surfaces of arbitrary dimension and complexities in geometry. Moreover, no shrinkages will be introduced by the method during the computation.

The method is built upon the concepts of *discrete graph signal* and its three-dimensional counterpart, *discrete surface signal* [9].

By discrete graph signal, it means that some functions defined on the directed graph (*digraph* for short). Corresponding the polyhedral surfaces in consideration, the digraph G can be marked as $G : \{V_g, E\}$, where $V_g = \{1, \ldots, n\}$ is *vertex set* including all the vertices in the graph, labeled from 1 to n; and E is the *edge set* including the edges between any pair of vertices in the set V_g. The so-called discrete graph signal is a n-dimensional vector $x = [x_1, \ldots, x_n]^T$, where the component $x_i, i = 1, \ldots, n$ denotes the signal intensity at the vertex $i \in V_g$.

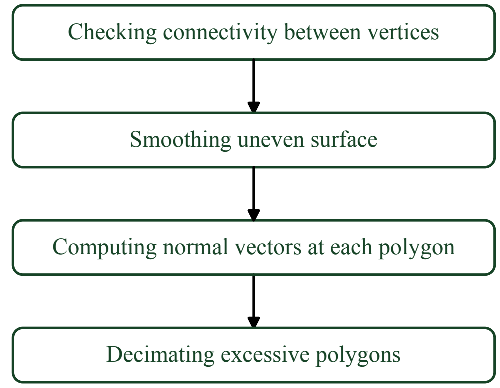
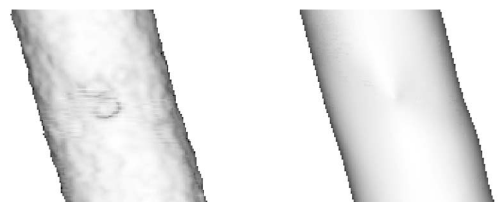


Fig. 1: Overview of the work flow.



(a) (b)

Fig. 2: Smoothing the artifacts: (a) the artifacts (the horizontal "steps" on the model surface of the vasculature); (b) the results after the smoothing process.

By discrete surface signal, it means that some functions defined on the polyhedral surfaces. The polyhedral surfaces S can be represented as $S : \{V_s, F\}$, where $V_s = \{1, \ldots, n\}$ is *vertex set* including all the vertices in the polyhedral surfaces, marked from 1 to n; and F is the set including the faces surrounded by the vertices in the set V_s and the edges connecting them. The discrete surface signal is a n-dimensional vector $y = [y_1, \ldots, y_n]^T$, where the component $y_j, j = 1, \ldots, n$ corresponds to the signal value at the vertex $j \in V_s$.

For the smoothing of the planar polygonal curves, the classical Fourier analysis can be applied to decompose the original curves into orthogonal subspaces with distinct frequencies. After that, the components at high frequencies are removed as noises and the ones at low frequencies are left. The idea can be extended to the case of the polyhedral surfaces with arbitrary topology.

For the discrete surface signal y, the weighted Laplacian operators for the vertices in V_s are defined as

$$\Delta y_i = \sum_{j \in i^*} w_{ij} \left(y_j - y_i \right), \tag{1}$$

where w_{ij} is the positive weight for $y_j - y_i$, and for any i, the sum of its weights are always one, $\sum_{j \in i^*} w_{ij} = 1$. The matrix form of (1) is

$$\Delta y = -Ky, \tag{2}$$

where $K = I - W$, with W the matrix of weights and I an identity matrix. Here we define $0 \le \lambda_1 \le \ldots \le \lambda_n \le 2$ as the real eigenvalues of K, and $e_1, \ldots, e_n$ the corresponding unit right eigenvectors.

At this point, the low-pass filtering can be presented as the multiplication of the matrix function $f(K)$ by the input signal y:

$$y' = f(K)y. \tag{3}$$

For any polynomial transfer function, the output signal in (3) can be rewritten as

$$y' = \sum_{i=1}^{n} \rho_i f(k_i) e_i, \tag{4}$$

where ρ_i are the coefficients of the linear combination of the unit right eigenvectors of the input signal $y = \sum_i \rho_i e_i$. In order to realize a low pass filter, $f(k_i)$ is designed to satisfy two conditions: for $k_i \in [0, 2]$, $f(k_i) \approx 1$ for low frequencies; $f(k_i) \approx 0$ for high frequencies.

The low-pass filtering mechanism can be achieved by adjusting the weights in the following polynomial approximation [10]:

$$f_N(k) = w_0 \frac{\theta}{\pi} T_0(1 - k/2) + w_n \sum_n \frac{2\sin(n\theta)}{n\pi} T_n(1 - k/2), \tag{5}$$

where θ is the unique solution of $k = 2(1 - \cos\theta)$ on $[0, \pi/2]$, and $T_0(\cdot)$ and $T_n(\cdot)$ are the Chebyshev polynomials. Here in this work, the weights in (5) is adjusted to form a Hamming window [10], which can be represented as follows:

$$w_n = 0.54 + 0.46\cos(n\pi/(N + 1)). \tag{6}$$

After smoothing the surface, the normal vectors (*normals* in short) are computed for each polygonal mesh.

C. Polygons Decimation

To eliminate the total number of polygons that consisting the model surface, an algorithm for the decimation of triangle meshes is employed in our approach [7]. The algorithm attempts to decimate certain rate of original meshes by firstly deleting the vertices whose coordinates are validated beforehand in the surface and then patching the holes left by creating a new triangulation at the location.

All the vertices are labeled as simple, complex, and corners (see Fig. 3). For simple vertices, two special cases are further specified as boundary vertices, and interior edge vertices.

To delete the unimportant vertices, some criteria must be designed and applied to validate the geometry of all potential targets among the model. First of all, the complex vertices are definitely retained. Then the rest are well checked to decide whether delete or not. For the simple vertices that are not located on the boundary or the interior edge, the distance from them to the average plane is the unique principle – if the distance is less than a certain value, it will be preserved;

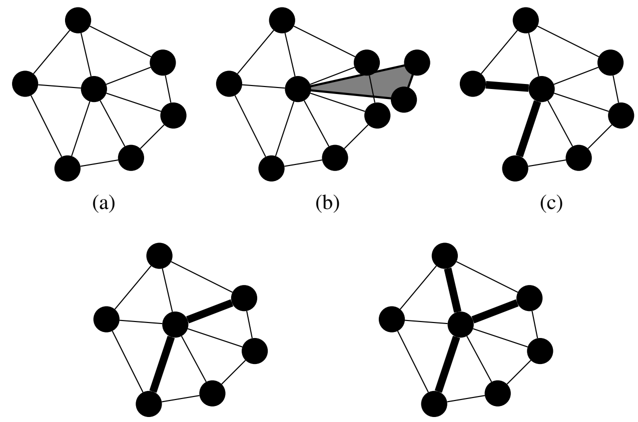


Fig. 3: Five classes of the vertex in the surface. (a) General simple case; (b) Complex case; (c) Boundary case; (d) Interior edge case; (e) Corner case.

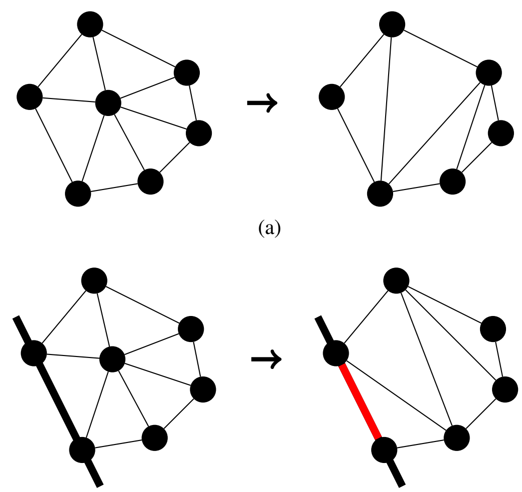


Fig. 4: Reduction of polygons before and after eliminating the vertex. (a) General simple vertex case: number of polygons before reduction: 6, number of polygons after reduction: 4; (b) Boundary vertex case: number of polygons before reduction: 5, number of polygons after reduction: 4. Note that the edge in *red* does not belong to the surface before the computation.

if not, it will be deleted. The vertices on the boundary or the interior edge are checked in the similar way – if the distance is less than a certain value, it will be retained; otherwise, it will be kicked out. The vertices located at corners are usually preserved to maintain the approximation of the original geometry or left the "noises" introduced in the modeling phase to be depressed by using appropriate method.

Once upon the deletion of the vertex and its associated edges, one (in simple case) or two holes (in boundary or interior edge case) are left for patching thus maintaining the surface. In achieving the new patches for these holes, new triangulation at the location is created, which is done before the actual deletion mentioned before.

Comparing the number of triangles before and after this process, two triangles are reduced for the cases of general simple vertices, interior edge vertices, and some corner vertices; whilst one for the cases of vertices located at the boundaries (see Fig. 4).

III. Experiments and Results

A. Data and Experimental Setup

Fig. 5 illustrated the experimental setup of the simulation system for the minimally invasive intravascular procedure (PCI). The user can interact with the simulator (i.e., deliver the virtual guidewire through the virtual blood vessels) by manipulating the mechantronic haptic device. All of our experimental programs ran on a desktop machine equipped with Intel's 2.83GHz Core 2 Quad CPU and 4GB RAM.

The vasculature surface models were generated from the original CTA images [11]. For the sake of simplicity, only a segment of the abdominal aorta (consists of $74,307$ polygons) is adopted in this paper. The approach can be ported in order to process the model surfaces of the whole vasculature related to the PCI procedure.

B. Validating Connectivity Between Vertices

The polygonal elements in the patient-specific surface model of the human vessels need to be checked such that the connectivity between any pair of vertices are validated. After doing this, the integration of the model surface is guaranteed that the excessive polygons are kicked off. Table I shows that the total number of the polygonal surfaces were not changed, which implies that the given model (i.e., the local one) was already the largest-connected region in the model before the validation process.

C. Smoothing Model Surface

Since the unavoidable noises introduced in the phase of image acquisition and processing, uneven surfaces may be found in the surface rendered based on the imagery. The mess of artifacts and crusts may induce the problems in the smoothness of the delivery of the virtual surgical tools, e.g., guidewires, catheters, etc. To avoid this situation, a thorough smoothing process is needed on the inner surface of the model, which is equivalent to the smoothing on the model surface per se.

The algorithm utilized in this processing task is implemented based on the low-pass filtering principles, with two parameters to be tuned in control of the smoothing effect on the objects. One of the parameters regulates the number of iterations the algorithm should take. It is equivalent to the order of the polynomial that approximating the windowed sinc function given in (5). The other parameter is used to configure the bandwidth of this so-called low-pass filter.

In this work, several parameter sets were given with the aim of searching for the one that can generate the most acceptable results. Judging from Fig. 6, the models on the top row (see Figures 6a, 6b, and 6c) smoothed with the bandwidth 0.1 are still fairly rough compared to the input, while the ones on the bottom row (see Figures 6d, 6e, and 6f) with the bandwidth 0.01 are much better. With the increase of the number of iterations, the smoothing effect became more apparent, compared with the column of results from the one on its left. At this stage of processing, we chose the parameter set which was used to generated the results depicted in Fig. 6f.

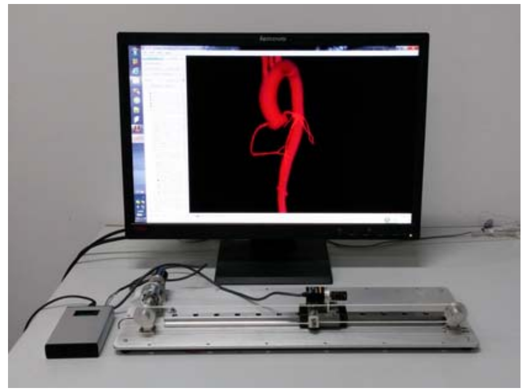


Fig. 5: Experimental setup of the 3-D simulation system for the PCI procedure.

TABLE I: No. of polygons before/after the validation

	No. of polygons
Before validation	$74,307$
After validation	$74,307$

D. Elimination of Component Polygons

The elimination of polygon elements of the model surface was conducted after the smoothing computation terminated demonstrated in the last section.

For the input surface model, over 74k polygons were required to model the vessel segment by using Marching Cubes method [12]. One can imagine that the overall number of the polygons needed to model the whole aorta and the trees of the coronary arteries. Rendering the large amount of data will surely exhaust most memory on any machine with typical hardware configuration, thus no sufficient space could be spared for the interaction and other computation. To facilitate this situation, we adopted the decimation algorithm introduced in Section II-C with multiple sets of parameters in a series of experiments.

In this work, the termination conditions of the adopted algorithm were set to 10%, 50%, 75%, 90%, and 99%, respectively. By termination condition, which is equivalent to the reduction rate, we mean that the rate of the total number of the polygonal elements deleted with respect to the total number of the polygonal elements in the input model surface. This can be written as follows

$$\text{Reduction Rate } (\%) = \frac{\text{No. of } \textit{deleted} \text{ Polygons}}{\text{No. of input Polygons}}. \tag{7}$$

Table II illustrates the effects of the decimation algorithm. Fig. 7 depicts the visualization results of the decimation with

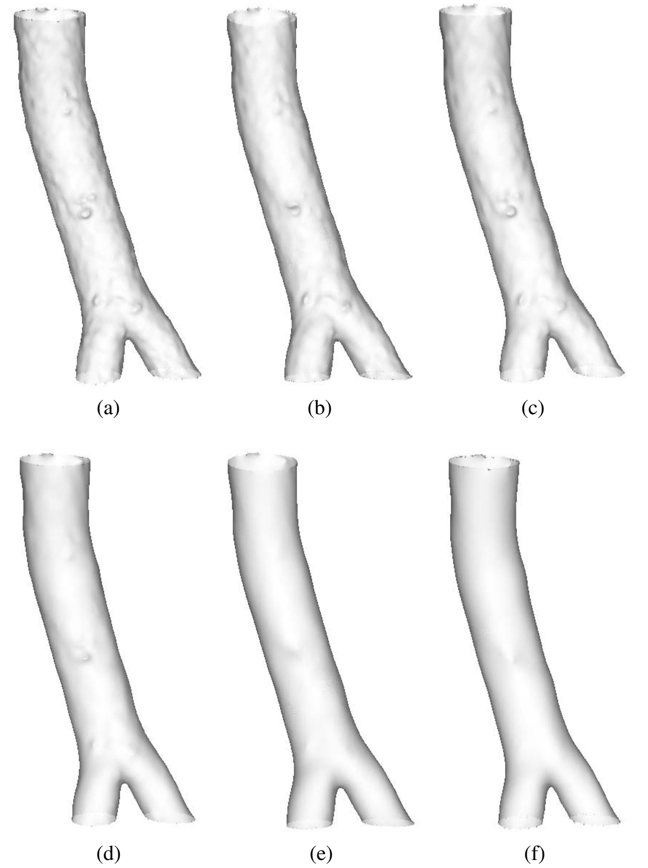


Fig. 6: Smoothing effects by applying different parameters: top row illustrates the results with pass band $= 0.1$, and the numbers of iterations are: 30, 60, and 100, respectively; bottom row illustrates the results with pass band $= 0.01$, and the numbers of iterations are: 30, 60, and 100, respectively.

typical parameters. Observing these results, the models in Figures 7b and 7c contain acceptable amounts of polygons. However, the visualization in Fig. 7c unveiled obvious alterations on the geometry of the model.

E. Validating Decimated Surfaces

We ran several trials to validate the interaction performance between the decimated models of the blood vessels and the virtual guidewire [13] in the simulation environment. Table III illustrates the relation between the reduction rates and the frames per second (fps) during the simulation of the decimated models represented in this work. Table IV illustrates the relation between the reduction rates and fps during the simulation of the decimated models of the whole blood vasculature for the 3-D simulation of the PCI procedure, i.e., the aorta plus the coronary arteries [11], [14]. By "Initial fps", we mean that the fps value when the virtual guidewire delivered into the vessel model. By "Average fps", we mean that the average fps value of the fps values at the five specific locations on the path along the lumen of the vessel model towards the given location in the coronary arteries. By "Success", we mean that whether the virtual guidewire

TABLE II: No. of polygons at different reduction rates

Reduction Rate (%)	10	50	75	90	99
No. of Polygons	$66,875$	$37,153$	$18,576$	$7,430$	743

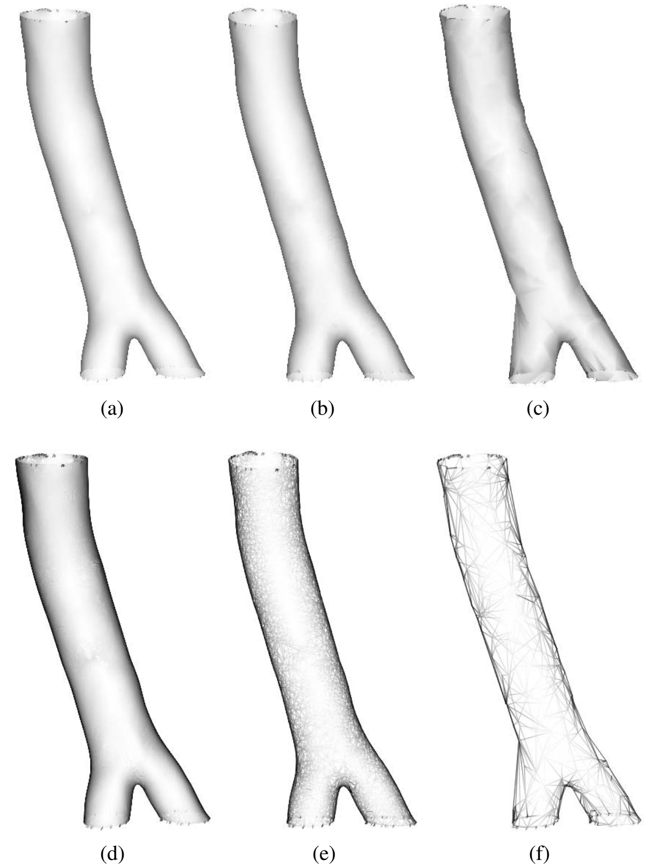


Fig. 7: Decimation effects by applying different parameters: top row illustrates the results with reduction rates of 10%, 90%, and 99%; bottom row illustrates the wire frame representation of the corresponding results depicted above.

had been delivered to the given location in the coronary arteries. Judging from these results, the model (consisting the aorta and the coronary arteries) with 99% of its polygons decimated were used in the simulation.

F. Discussions

All the programs ran in the experiments were written in C++. Some functioning modules were implemented based on the Visualization Toolkit (VTK) [15], an open source effort implementing various of algorithms from both the general and special fields of computer graphics.

In validating the connectivity of the adjacent points among the surface model, the algorithm was configured to extract the largest-connected areas in this work.

To depress the noises introduced by image processing, a surface smoothing algorithm was employed by slightly altering the locations of the vertices of the meshes. In smoothing the surfaces, two distinct parameters were the

TABLE III: Reduction rates vs. fps — Case I

Reduction Rate (%)	10	50	75	90	99
Initial fps	30	35	38	41	42
Average fps	6	10	15	18	30
Success? (Yes/No)	No	No	Yes	Yes	Yes

TABLE IV: Reduction rates vs. fps — Case II

Reduction Rate (%)	90	95	99
Initial fps	20	32	31
Average fps	4	6	11
Success? (Yes/No)	No	No	Yes

key switches for the algorithm, among which the width of the pass band of the low-pass filter and the number of the iterations to be performed were determined. Between the number of iterations and the smoothing effects on the surface, a positive correlation was demonstrated. Meanwhile, the width of the pass band and the smoothing effects implied a negative correlation.

In decimating the polygons, the problem is: on the one hand, the higher rate of reduction is preferred for the interactive simulation; on the other hand, the over high rate of reduction may lead to heavy tortures on the model surface thus the geometry is doomed to be destroyed. This should be taken into consideration when selecting the parameter for the decimation algorithm. In this work, we have tested with different rates and successfully determined the one that was acceptable. The output of the decimation process are the progressive meshes [16]. And the resultant meshes had well preserved the original geometry of the model.

In validating the performance of the interaction, the fps greater than 10 is sufficient for the prototype. In this paper, the surface model representing the aorta and the coronary arteries for the future simulation study was decimated 99% of its original polygons, achieving an fps of 11 (> 10) on average.

IV. Conclusions and Future Work

In the context of simulating the intravascular procedure, the interactive performance between the virtual surgical tools (i.e., catheters or guidewires) and the vasculature is one of the most important benchmarks. The amount of the data used to model the surfaces of the related vasculature is among the main factors. In this paper, we developed an approach to address this problem. The idea is to eliminate the polygons consisting the model surface without destroying the geometry of the vasculature. The data used in the experiments were the patient-specific vessel model generated in our previous work [11], [14].

First the connectivity between the adjacent vertices was validated and the unconnected vertices were removed. Then the noisy surfaces introduced from the image acquisition and processing were smoothed with the least effects on the model's geometry. Finally the decimation pass was introduced to reduce the number of the polygon components at the desired reduction rate. In reducing the polygonal elements, not only did we consider the number of the polygons that had been removed, but also the visualization of the resultant models. Validation illustrated the effectiveness of our approach in improving the performance of the interaction with the virtual guidewire.

Our future work will be the visualization of other human organs (e.g., heart, skeletons, etc.) appeared in the field of the real surgeries on the one hand, and the physical modeling of the virtual human organs on the other.

References

[1] World Health Organization, "The top 10 causes of death." [Online]. Available: http://is.gd/Mr3Ktz

[2] N. R. Smilowitz and G. Weisz, "Robotic-assisted angioplasty: current status and future possibilities," *Current Cardiology Reports*, vol. 14, pp. 642–646, 2012.

[3] R. Beyar, L. Gruberg, D. Deleanu *et al.*, "Remote-control percutaneous coronary interventions: Concept, validation, and first-in-humans pilot clinical trial," *Journal of the American College of Cardiology*, vol. 47, no. 2, pp. 296–300, Jan 2006.

[4] A. Lunderquist, K. Ivancev, S. Wallace *et al.*, "The acquisition of skills in interventional radiology by supervised training on animal models: A three-year multicenter experience," *CardioVascular and Interventional Radiology*, vol. 18, no. 4, pp. 209–211, 1995.

[5] T. Mori, N. Hatano, S. Maruyama *et al.*, "Significance of 'hands-on training' in laparoscopic surgery," *Surgical Endoscopy*, vol. 12, no. 3, pp. 256–260, 1998.

[6] C. Ji, Z.-G. Hou, and X.-L. Xie, "An image-based guidewire navigation method for robot-assisted intravascular interventions," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, 2011, pp. 6680–6685.

[7] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen, "Decimation of triangle meshes," in *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '92. New York, NY, USA: ACM, 1992, pp. 65–70.

[8] G. Taubin, "Curve and surface smoothing without shrinkage," in *Computer Vision, 1995. Proceedings., Fifth International Conference on*, 1995, pp. 852–857.

[9] ——, "A signal processing approach to fair surface design," in *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '95. New York, NY, USA: ACM, 1995, pp. 351–358.

[10] G. Taubin, T. Zhang, and G. Golub, "Optimal surface smoothing as filter design," in *Computer Vision ECCV '96*, ser. Lecture Notes in Computer Science, B. Buxton and R. Cipolla, Eds. Springer Berlin Heidelberg, 1996, vol. 1064, pp. 283–292.

[11] F. Yang, Z.-G. Hou, and S.-H. Mi, "Lumen segmentation of abdominal aorta using geodesic active contours for robotic intravascular surgical simulator," State Key Lab. of Manage. & Control for Complex Syst., Inst. of Autom., Beijing, China, Tech. Rep., Aug. 2013.

[12] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 163–169, Aug. 1987.

[13] S.-H. Mi, Z.-G. Hou, F. Yang *et al.*, "Multi-body mass-spring model for virtual reality training simulators based on a robotic guide wire operating system," in *Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on*, 2013, pp. 2031–2036.

[14] F. Yang, Z.-G. Hou, S.-H. Mi *et al.*, "3D modeling of coronary arteries based on tubular-enhanced CURVES segmented regions for robotic surgical simulation," in *Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on*, 2013, pp. 2013–2018.

[15] W. Schroeder, L. Avila, and W. Hoffman, "Visualizing with VTK: a tutorial," *Computer Graphics and Applications, IEEE*, vol. 20, no. 5, pp. 20–27, 2000.

[16] H. Hoppe, "Progressive meshes," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '96. New York, NY, USA: ACM, 1996, pp. 99–108.