IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON HUMAN-CENTERED SMART SYSTEMS AND TECHNOLOGIES

Received April 26, 2018, accepted May 22, 2018, date of publication May 25, 2018, date of current version June 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2840494

A Novel WebVR-Based Lightweight Framework for Virtual Visualization of Blood Vasculum

CHENXI HUANG[1], WEN ZHOU[2], YISHA LAN[1], FEI CHEN[3], YONGTAO HAO[1], YONGQIANG CHENG[4], AND YONGHONG PENG[5], (Member, IEEE)

[1]Department of Computer Science and Technology, Tongji University, Shanghai 201804, China
[2]School of Software Engineering, Tongji University, Shanghai 200065, China
[3]Department of Cardiology, Shanghai Tongji Hospital, Tongji University, Shanghai 200065, China
[4]School of Engineering and Computer Science, University of Hull, Hull HU6 7RX, U.K.
[5]Faculty of Computer Science, University of Sunderland, St Peter Campus, Sunderland SR6 0DD, U.K.

Corresponding authors: Wen Zhou (zhouwen327@163.com) and Yongtao Hao (hao0yt@163.com)

This work was supported in part by the National Science and Technology Support Program under Grant 2015BAF10B01 and in part by the Natural Science Foundation of China under Grants 81370390, 81500381, and 81201069.

ABSTRACT With the arrival of the Web 2.0 era and the rapid development of virtual reality (VR) technology in recent years, WebVR technology has emerged as the combination of Web 2.0 and VR. Moreover, the concept of ''WebVR + medical science'' is also proposed to advance medical applications. However, due to the limited storage space and low computing capability of Web browsers, it is difficult to achieve real-time rendering of large-scale medical vascular models on the Web, let alone large-scale vascular animation simulations. The framework proposed in this paper can achieve virtual display of the medical blood vasculum, including lightweight processing of the vasculum and virtual realization of blood flow. This innovative framework presents a simulation algorithm for the virtual blood path based on the Catmull–Rom spline. The mechanisms of progressive compression and online recovery of the lightweight vascular structure are further proposed. The experimental results show that our framework has a shorter browser-side response time than existing methods and achieves efficient real-time simulation.

INDEX TERMS WebVR, aortic vascular, Web3D, Catmull-Rom spline, lightweight vascular progressive compression.

I. INTRODUCTION

With the development of virtual reality (VR), an increasing number of industries have begun to develop virtual reality technologies. A multitude of novel application concepts have also arisen, such as virtual communities, virtual architecture and virtual characters, among others [1]. Insiders refer to the year 2016 is referred to as the primary era of virtual reality, and the virtual reality industry is expanding enormously. Of the fourteen important challenges of the twenty-first century listed by the field of human engineering technology, as published by National Academy of Engineering, two of them are closely related to virtual reality, i.e., enhancing the verisimilitude and applicability of virtual reality and reverse engineering of the human brain. In 2016, the market for the global virtual reality industry reached $6.7 billion and is expected to reach $100 billion in 2020. Therefore, virtual reality technology has become one of the most valuable industries. Virtual reality technology, which is also known as ''immersive multimedia'' or ''computer simulation reality'', is considered an important development subject in the twenty-first century and is one of the most paramount technologies that affect the lives of people. Virtual reality is a new technology that combines computer graphics, human-machine interface technology, sensor technology, artificial intelligence and other elements. The traditional human-computer interaction occurs through the mouse and keyboard, whereas the interaction with virtual reality is performed in the most natural manner based on sensors and any object in the virtual environment. The goal of virtual reality technology is to improve the function of human-computer interaction and eventually achieve a realistic experience with respect to vision, olfaction, hearing and touch. Furthermore, a realistic three-dimensional multi-sensory environment can be generated via computers, graphic workstations and other related equipment. As a result, a teaching situation with self-participation is produced, and the environment also gives feedback on the behavior of participants to achieve deep integration and interaction between people and the environment. The important features of virtual

reality are captured by ''3I'', i.e., immersion, interaction and imagination. Immersion refers to a variety of senses, such as vision, hearing, olfaction, taste and touch in the virtual environment, which present a telepresence to the participants. Interaction refers to the human-computer interface and natural feedback of the participants in the virtual environment, and imagination refers to the way in which participants form conceptions of the future with changes in the environmental states and interactive behaviors [2].

With the advent of Web 2.0, the concept of ''Internet plus'' has been proposed, and this suggests the brand-new field of WebVR technology, i.e., internet plus virtual reality technology [3]. The big-data model is transformed by WebVR technology [4] from traditional 4V (volume, variety, velocity and value) to 5V, which contains an additional visualization. Moreover, the novel concept of WebVR plus medical science is proposed because medical science is one of the fields most closely related to human beings. Many problems such as medical disease research and medical teaching can be solved by WebVR plus medical science. Despite the rapid development of WebVR technology, a series of problems urgently need to be solved. For example, WebVR has features of enormous data transmission demand, complex semantics, loose structure and low loading speed over the Internet. Moreover, the rate of WebVR scene operation is low with poor effect. For the web browser, both the speed and effect of online rendering are unsatisfactory.

To address the above problems under the research background of WebVR plus medical science, this paper proposes a novel visual framework in a WebVR plus medical vascular structure to visualize the virtual blood vasculum based on a web browser. This research aims to improve the teaching of medical science in practice. The main requirements that must be achieved for browser-based virtual vascular system are high-level real-time characteristics of the vascular system and a better human machine interaction experience. In this manner, users can freely use the terminal interaction tools, such as the mouse, to interact with the virtual vascular system, including dragging and rotating, with less or no delay. Furthermore, blood flow movement is simulated to represent real scenarios as accurately as possible in a smooth and vivid manner. Because of the inclusion relationship of vessels and blood, the flow of blood cannot be physically seen through the vessels. As a result, to achieve a better visual effect, blood flows close to the vessels in our practical approach, and thus the flow direction of blood can be clearly observed, which contributes to a pleasurable teaching effect.

Our study contains two innovative contributions:

1) A simulation algorithm for the virtual blood path based on the Catmull-Rom spline, resulting in a more vivid effect of the virtual vessels.
2) A lightweight flow treatment that contributes to the improved effect of vascular progressive compression and guarantees the efficiency of data rendering in the browser portal to achieve better visualization.

The remainder of the paper is organized as follows. Related work is introduced in Section II. The lightweight virtual framework is proposed in Section III. The experimental results and discussion are presented in Section IV. Section V concludes the paper with plans for future work.

II. RELATED WORK

A. LIGHTWEIGHT TECHNOLOGY

Lightweight technology has been well studied since the 1990s. Hoppe [5] proposed a progressive network method based on mesh collapse, but high-manifold meshes cannot be distinguished by this method, and a long computing time is needed. An elimination algorithm based on an increment of the quadric surface was developed by Garland and Heckbert [6] to achieve lightweight meshes, but this approach also suffers from long computing time. Kettner [7] proposed a new data structure known as the half-edge data structure to achieve lightweight meshes, and this method has been widely used by researchers. The advantage of this structure is that the mutual query is convenient, and the nodes in it occupy a large amount of space. Morigi and Rucci [8] improved the method using a simplified information mechanism for mesh surface evolution and developed a hierarchical mesh simplification algorithm in a Riemannian geometry mesh-surface manifold under the p-Laplacian regularization operation. The method can be adapted to simplify the geometry topology structure, but this approach also takes a long time to calculate. Furthermore, Wen *et al.* [9] proposed an algorithm for lightweight progressive meshes (LPM) that contributes to streaming of mesh transmission and visual effect. Reusable components can be eliminated through reusability detection for components in the meshes under this algorithm. Hence, the mesh data are progressively transferred. However, all of the methods above rely on manual model segmentation, which is highly labor intensive.

In general, Human's observation of objects trends from blurred to clear. The time spent in this process can be completely negligible compared with the time used by network caching and other factors. The basic problem of rendering the level-of-details (LoD) task is a precise process of simplifying the model and making the initial loading model into a rough mesh model in which simplification of raw meshes is indispensable. The mesh-loading simplified level-based bandwidth adaptive technology and LoD was proposed by Zhao *et al.* [10] has and contains inconformity for a large-scale mesh model. Lavoué *et al.* [11] developed a 3D visual model based on WebGL technology to render a large-scale model at the browser end in real time. However, the method only fits for the specific model due to various constraints. In a review by Evans *et al.* [12], the current development of Web3D technology is cast as vibrant and exciting, both in the academic and the wider developer communities. Kronrod and Gotsman [13] developed a type of streaming mesh code that optimizes the geometric component but sacrifices the connectivity of meshes. A method for

streaming mesh code known as GEncoder was proposed by Lewiner *et al.* [14] and is a single-rate and geometric-driven lightweight method. In this method, mesh compression can be conducted for any dimensions, regardless of the mesh topological limitation. However, all of the methods above depend on a mesh prior hypothesis and lead to large effect differences for different types of meshes. Bajaj *et al.* [15] developed a lightweight method based on mesh triangle-zone coding, and Diazgutierrez *et al.* [16] proposed a mesh coding method based on the minimum spanning tree. Moreover, Coors and Rossignac [17] and Gumhold [18] put forward a lightweight method based on an ergodic coding model of a triangular mesh, and triangular-mesh coding based on the covalent bond was proposed by Mamou *et al.* [19]. However, these methods are not oriented to WebVR and take a long time because of the complexities of the model-rendering algorithms.

B. THREE-DIMENSIONAL VISUALIZATION OF AORTIC VASCULUM

Currently, CT and other tomography methods are widely used in clinical diagnosis and treatment to supply sliced 2D images. Because the 2D slice images only express the anatomic information of one section, doctors mostly rely on experience to imagine the shape of the aortic vascular cross-referenced to a number of two-dimensional images, which leads to difficulties in treatment. In recent years, 3D visualization has become a fundamental task in building web applications. A multitude of outstanding research studies have been performed in both applications and theory [20]. The aortic vascular system reconstructed by Mimics, a 3D reconstruction software package, can be viewed and rotated dynamically. In this manner, clinicians can produce a thorough, detailed localization and qualitative and quantitative analysis [21]. Moreover, based on the reconstructed 3D vascular system, preoperative surgical planning and surgical simulation can be conducted, and an accurate model for biomechanical finite element analysis is also possible. The process of 3D aortic vascular reconstruction using Mimics consists of four steps: 1) The image import function of Mimics is used to input the CT sectional images. Four directions (anterior, posterior, left and right) are automatically defined, and the top and bottom directions are manually defined. Moreover, images of the sagittal and coronal planes are automatically produced according to the axial images. 2) A mask is generated by extracting the profiles in the editing operation interface of three views. 3) A hotspot is selected using region growth in the software. In this manner, 2D images can be converted into a 3D model using the Calculate 3D function based on a 3D interpolation method. 4) A 3D visualization model is obtained in STL format [22].

C. PROGRESSIVE TRANSMISSION

In the commonly used client-server data transmission scheme, a client first sends a request to the server side [23]. When the request is recognized, the server side begins to transmit data to the client. In this variety of a traditional spatial data transmission model, diversification in client devices and network equipment (such as portable computers, portable terminals, mobile communication equipment) leads to the inconsistent needs of spatial data. Progressive transformation consists of novel transmission of a spatial data network and is one of the hottest topics in current GIS research. Three hierarchies of spatial data granularity control are proposed in progressive transmission, i.e., factor level, target level and geometric detail level. Based on this foundation, the progressive transmission process of spatial data is discussed in accordance with the priority of spatial scale and semantic level [25]. Furthermore, four strategies are proposed to build a multi-scale space database that includes multilevel-scale display storage, primary-scale change accumulation, key-scale function evolution and primary-scale automatic synthesis. The transmission effectively solves the prominent contradiction between limited network bandwidth and the unlimited growth of spatial data. However, in the client display process, spatial data implements the progressive visualization from the most important to the least important information and from low resolution to high resolution. This approach meets the needs of users for browsing, inquiry and analysis of the data from different angles and different levels.

III. METHODOLOGY

In this section, we propose the data-driven novel framework. The framework consists of three components, as illustrated in Fig. 1: the technology of vascular lightweighting for the vasculum, the blood-path plan based on the Catmull-Rome spline at the server side and the recovery at the browser end. In the section on the vascular lightweight, a method for progressive meshing based on mesh folding is used to compress the data of the vessels. Generally, the information included in the mesh is divided into three categories of geometry, connectivity, and attribute information. The geometry information is defined as the position of each vertex in the mesh in 3D Cartesian space. The connectivity information primarily describes the incidence relationships among mesh elements. The attribute information associates scalar or discrete properties of the mesh elements, such as color, norm, texture coordinates, etc. Considering the methods of the blood-path plan, a method based on splines is developed to make the blood flow smoothly in the vessels and achieve a better virtual effect as well. Furthermore, we transmit data progressively to the browser portal to obtain a flowing effect in the data rendering. As a result, blood can flow fluently following the planned path and a rotation angle can be autonomously defined by judging the appearance characteristics of the vessels. Thus this approach guarantees that the direction of the blood center crosses correctly on the key points and achieves a more vivid effect.

A. CATMULL-ROM SPLINE-BASED BLOOD PATH PLANNING

1) ACQUISITION OF VIRTUAL BLOOD PATH

To achieve the virtual blood effect, a model of the capsule is used to simulate the vessels. Subsequently, N sample points

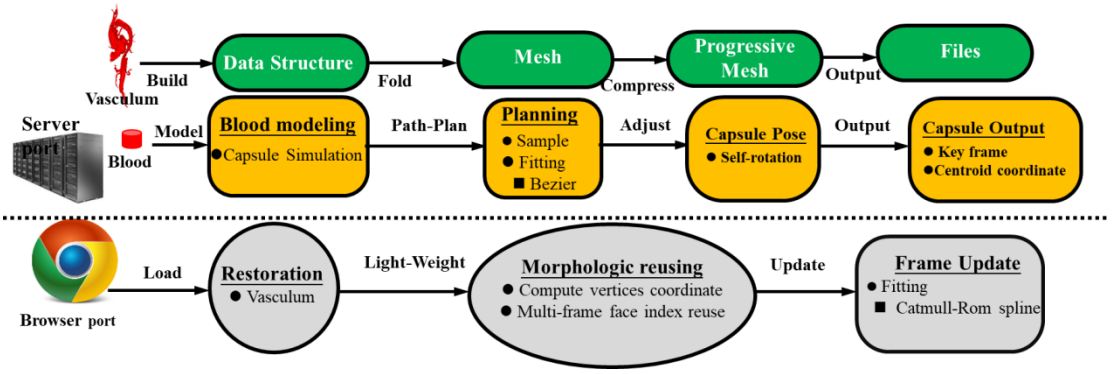

FIGURE 1. Overview of proposed framework.

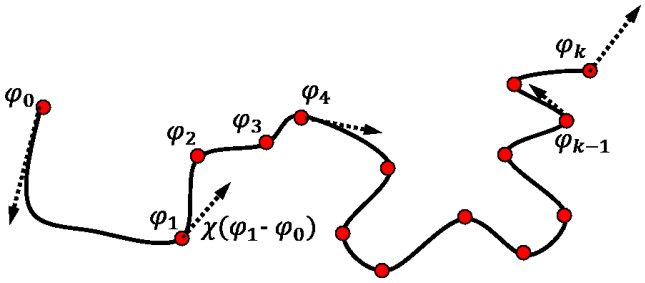

FIGURE 2. Path of virtual blood based on Catmull-Rom spline.

written as $\psi = \{i = (0, N)|\varphi_i \in \psi\}$ are collected randomly and continuously in a mesh. The tangent lines of every sample point are calculated as shown in Fig. 2.

We assume that four sequential sample points φ_{i-2}, φ_{i-1}, φ_i, φ_{i+1} are chosen in a path P(s) of the virtual blood, and the path can be given as follows:

$$P(s) = a_0 + a_1\mu + a_2\mu^2 + a_3\mu^3 = \sum_{i=0}^{3} a_i\mu^i \tag{1}$$

where μ is a variable and a_0, a_1, a_2, a_3 are coefficients. The equations below are obtained by differentiating the sample points in the path P(s).

$$p(0)' = \chi(\varphi_i - \varphi_{i-2}) \tag{2}$$
$$p(1)' = \chi(\varphi_{i+1} - \varphi_{i-1}) \tag{3}$$
$$p(0) = \varphi_{i-1} \tag{4}$$
$$p(1) = \varphi_i \tag{5}$$

The influence of the term χ is described in Fig. 3. The curvature becomes more abrupt as the value of χ increases.

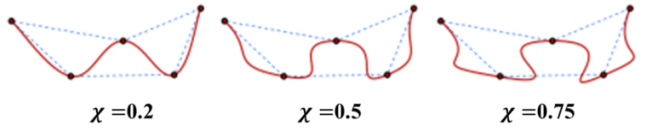

FIGURE 3. Effect of χ.

By comparing (1) with (2), (3), (4) and (5), the following equations can be obtained,

$$a_0 = \varphi_{i-1} \tag{6}$$
$$a_0 + a_1 + a_2 + a_3 = \varphi_i \tag{7}$$
$$a_1 = \chi(\varphi_i - \varphi_{i-2}) \tag{8}$$
$$a_1 + 2a_2 + 3a_3 = \chi(\varphi_{i+1} - \varphi_{i-1}) \tag{9}$$

The path can also be expressed in matrix form as shown below,

$$P(s) = [1 \quad \mu \quad \mu^2 \quad \mu^3] \times \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\chi & 0 & \chi & 0 \\ 2\chi & \chi - 3 & 3 - 2\chi & -\chi \\ -\chi & 2 - \chi & \chi - 2 & \chi \end{bmatrix} \begin{bmatrix} \varphi_{i-2} \\ \varphi_{i-1} \\ \varphi_i \\ \varphi_{i+1} \end{bmatrix} \tag{10}$$

In this manner, a virtual blood path is obtained, and parameter χ is set to 0.5 to obtain a best fitting curve in our work.

2) VIRTUAL VASCULUM WITH ANIMATION

In accordance with the changes in vascular curvature, an automatic rotation of the virtual blood is produced to achieve a more vivid effect. The normal curvature is calculated first.

In Fig. 4, the normal vector is changed in the direction of the tangent vector, and the normal curvature is defined as the change rate of the normal vector. Two vector operations, including internal operation and external operation, are developed to calculate the normal curvature. The Jacobian matrix of point P in a curve is given by the following

$$\psi = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial z}{\partial u} & \frac{\partial x}{\partial v} \end{bmatrix} = [\psi_u \quad \psi_v] \tag{11}$$

Internal operation I is mathematically given by the following

$$I = \psi^T\psi = \begin{bmatrix} \psi_u^T\psi_u & \psi_v^T\psi_u \\ \psi_u^T\psi_v & \psi_v^T\psi_v \end{bmatrix} \tag{12}$$

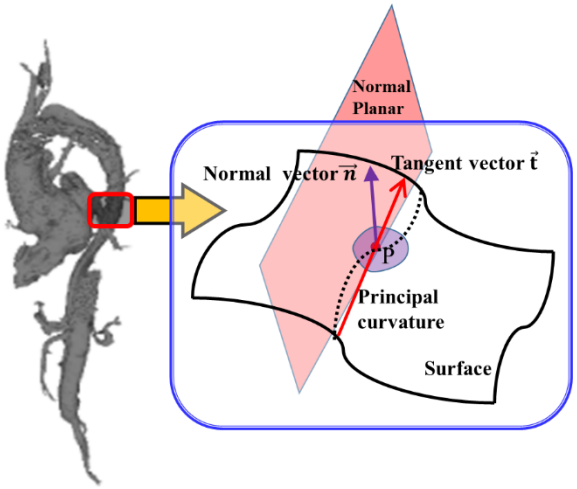

FIGURE 4. Principal curvature of vasculum (the dotted line is the normal curvature).

External operation Θ is defined as shown

$$\Theta = \begin{bmatrix} \frac{\partial^2 \psi}{\partial u^2}\vec{n} & \frac{\partial^2 \psi}{\partial u \partial v}\vec{n} \\ \frac{\partial^2 \psi}{\partial u \partial v}\vec{n} & \frac{\partial^2 \psi}{\partial v^2}\vec{n} \end{bmatrix} \tag{13}$$

The normal line of the normal vector $\vec{t}$ at point P is expressed as follows:

$$\vec{K} = \frac{\vec{t}^T I \vec{t}}{\vec{t}^T \Theta \vec{t}} \tag{14}$$

In the process, $\vec{\Delta}$ is defined as the gravity vector of the virtual blood. To obtain the rotation angles of the blood in three planes, vectors including $\vec{K}$ and $\vec{\Delta}$ are projected into three Cartesian planes, which are the XY plane, XZ plane and YZ plane. The normal vectors of the three planes above are $\vec{n}_{xy} = (0, 0, 1), \vec{n}_{xz} = (0, 1, 0)$ and $\vec{n}_{yz} = (1, 0, 0)$, respectively. Additionally, $\vec{K}_{xy}$ which is the vector after projection of $\vec{K}$, is shown as follows.

$$\vec{K}_{xy} = \vec{K} \times \vec{n}_{xy} \tag{15}$$

Similarly, other vectors such as $\vec{K}_{xz}, \vec{K}_{yz}$ after projection can be calculated in the same way. Furthermore, three projection vectors of $\vec{\Delta}$ are produced as $\vec{\Delta}_{xy}, \vec{\Delta}_{xz}, \vec{\Delta}_{yz}$ in a similar approach.

The rotation angle is given by the following equation

$$\vec{\theta} = \begin{bmatrix} \theta_{xy} \\ \theta_{xz} \\ \theta_{yz} \end{bmatrix} = \begin{bmatrix} \cos^{-1}\left(\frac{\vec{K}_{xy} dot \vec{\Delta}_{xy}}{\left|\vec{K}_{xy}\right| \left|\vec{\Delta}_{xy}\right|}\right) \\ \cos^{-1}\left(\frac{\vec{K}_{xz} dot \vec{\Delta}_{xz}}{\left|\vec{K}_{xz}\right| \left|\vec{\Delta}_{xz}\right|}\right) \\ \cos^{-1}\left(\frac{\vec{K}_{yz} dot \vec{\Delta}_{yz}}{\left|\vec{K}_{yz}\right| \left|\vec{\Delta}_{yz}\right|}\right) \end{bmatrix} \tag{16}$$

Algorithm 1 is a method of the virtual blood-path plan based on the Catmull-Rom spline.

Algorithm 1 Catmull-Rom Spline-Based Blood Path Planning Algorithm

Step 1: Select N sample points ψ from vessels that are in a uniform distribution.
Step 2: Calculate the barycentric coordinates $\vec{\Delta}$ of the virtual vessels.
Step 3: According to (14), $\forall \varphi_i \in \psi$, calculate the normal curvature vector of the point.
Step 4: In accordance with (16), calculate the rotation angle of the blood.
Step 5: Perform an operation of sample fitting for the set of sample points ψ.
Step 6: According to (10), calculate the path in each frame of four continuous sample points $[\varphi_{i-2}, \varphi_{i-1}, \varphi_i, \varphi_{i+1}] \in \Psi$. If $i+1 > |\Psi|$ then end this step, otherwise repeat step 6.

B. LIGHTWEIGHT FOR VASCULUM AND RESTORATION OVER WEB BROWSER

This section discusses the lightweight process of the virtual vessels and the method that uses JavaScript to recover the mesh data after the lightweight process at the browser end. Figure 5 illustrates the operations of collapse and split for the models. The collapse operation shown in Fig. 5(a) is conducted off-line, and the split operation shown in Fig. 5(b) is performed on-line through JavaScript. Because operations such as the pointer are not supported by JavaScript and JavaScript is single-threaded, a JavaScript worker model is used to achieve a multithreading effect. The model is realized by running the rendering and mesh split processes in the foreground. Therefore, the progressive mesh-rendering effect is achieved by reducing the resolution of the original mesh and noting the interrelated details.

1) MESH COLLAPSE

The half-edge data structure proposed by Kettner [7] is adopted in the mesh collapse. By adjusting the pointer of a half-edge data structure, two of the vertices are merged into a new vertex. The index of one of the two folding points becomes the index of the new vertex in the mesh with updated coordinates, which are generally located at the center of the two folding points. As a result, the number of faces and vertices of the entire mesh are reduced. Additionally, a split object set ϕ is accordingly generated for future recovery. Fig. 6 shows one example scenario for the mesh collapse and split object generation process. Five objects are shown in the figure, $\forall S_i \in \phi$. Three objects are vertex indices v_s, v_l, v_r in the split region after the split operation and the original coordinates of the two vertices $\vec{v}_0, \vec{v}_1$ that are merged.

2) MESH RECOVERY BASED ON JAVASCRIPT

In the browser on the rendering end, triangles of the model mesh are constantly split to recover the mesh. The process of mesh recovery is shown in Fig. 7. First, the folded data are rendered by the main program on the client side. All triangles

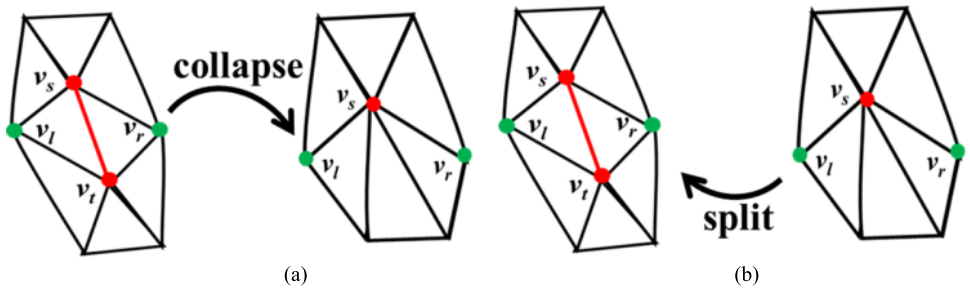

FIGURE 5. Operation of mesh collapse and split. (a) Triangles collapse. (b) Triangle split.

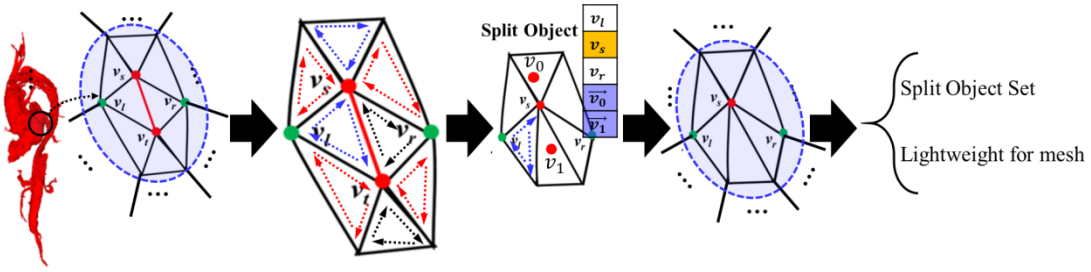

FIGURE 6. Overview of collapse.

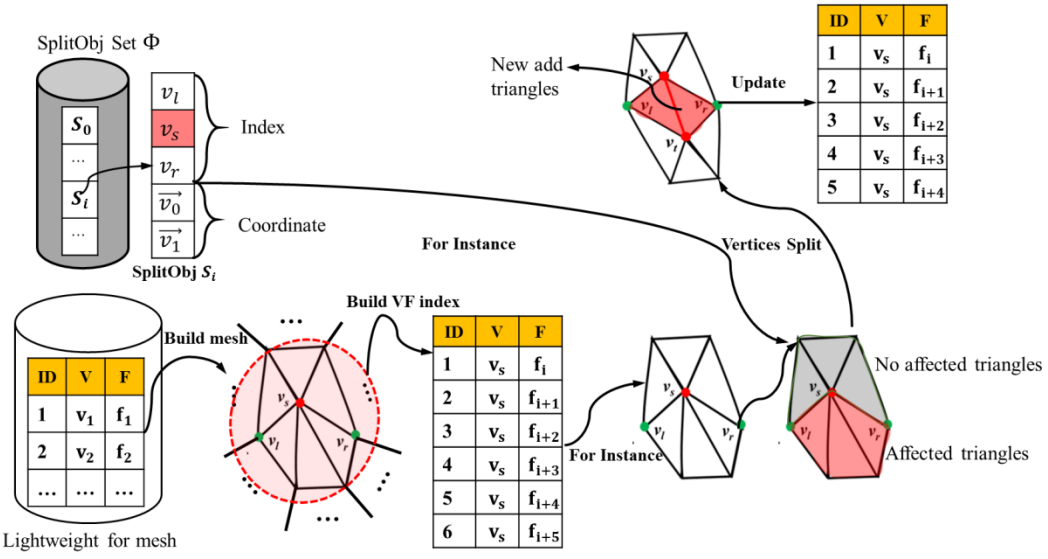

FIGURE 7. Restoration of mesh based on JavaScript.

of the vertex can be obtained by setting up a point-surface index table. The split objects, including the index of the point v_s, its neighboring vertices v_l and v_r and the target coordinates $\vec{v}_0$, $\vec{v}_1$ for the new vertices v_s, v_t to be generated, are transmitted to the JavaScript worker thread by the main progress. The worker thread splits the mesh vertices and feeds back the point-surface index table to the main program after the split. Finally, the entire mesh is updated by the main process. The process is iterated, and the mesh with fine details is progressively displayed. Because the browsers receive and render data progressively, the demands for processing and caching operations are dramatically reduced during the process. To a certain extent, the proposed algorithm relieves the processing pressure on the browser.

Algorithm 2 describes the details of the mesh split. To list and differentiate the affected triangles from unaffected ones, we apply clockwise rules to order the triangle indices and determine those that are affected.

3) VIRTUAL BLOOD ANIMATION

To vividly simulate the blood flow, we use a capsular model. The rotation angles of the virtual blood vessels can be

Algorithm 2 Mesh Split

Initialization: With the point-surface mesh table Θ, $\forall$ split objects $S_i = \{v_s, v_l, v_r, \vec{v}_0, \vec{v}_1\} \in \phi$, $\kappa \leftarrow v_l$
Step 1: Do while (True)
Step 2: For i in Θ
Step 3: Surface index $f \leftarrow \Theta[i]$
Step 4: Obtain three vertices sets according to the surface index
Step 5: For j in [0,4)
Step 6: If vertex f[j] is equal to the v_s in S_i then
Step 7: Break;
Step 8: End if
Step 9: End for
Step 10: If $f[(j+2)\%3] == \kappa$ then
Step 11: Remove ith surface index from Θ
Step 12: $\kappa \leftarrow f[(j+2)\%3]$
Step 13: End if
Step 14: End for
Step 15: If $v_r == \kappa$ then break; End If
Step 16: End while
Step 17: $\kappa \leftarrow v_r$
Step 18: Repeat step 2-16 with variables v_l and v_r
Step 19: Add two new triangles
Step 20: Update the coordinates of v_s, v_t in two triangles by $\vec{v}_0$, $\vec{v}_1$

calculated using (16) on the Cartesian plane. Each of the virtual capsules of the blood rotates via the Euler rotation formula and adaptively follows the calculated capsular path by changing its posture. The Euler rotations of the capsular model on the three Cartesian planes are required to change the posture. Assuming that the rotation angles of the three Cartesian planes are α, β, γ for the XY, XZ and YZ planes, respectively, the rotation formula can be expressed as shown below,

$$R_{xy} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{17}$$

$$R_{xz} = \begin{bmatrix} -\sin\beta & 0 & \cos\beta \\ 0 & 1 & 0 \\ \cos\beta & 0 & \sin\beta \end{bmatrix} \tag{18}$$

$$R_{yz} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix} \tag{19}$$

IV. RESULTS AND DISCUSSIONS

The experiments are conducted with the following settings: Intel Core I7 processor, 8 GB RAM, and NVIDIA GTX1060 graphic card running on Windows 10 operating system. The network bandwidth of the sever side is 4 Gbps. The offline program is coded in C++, and the online program is written in JavaScript using the three.js library.

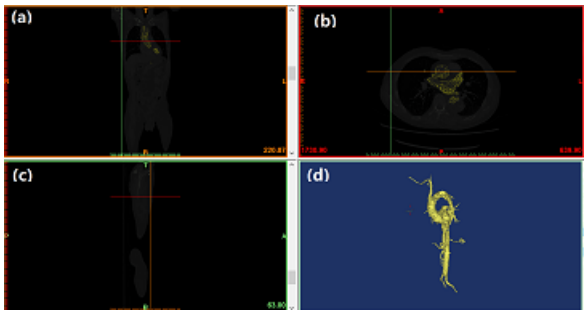

FIGURE 8. A contrasting experiment with CT images and 3D visualization image.

A. SUBJECTIVE EVALUATIONS OF EXPERIMENTAL RESULTS

As shown in Fig. 8, the intuitive display of the aorta vessels can be visualized using a 3D visualization model based on CT. Fig.8 (a) to (c) show CT images from different angles, and Fig.8 (d) is a final result of volume visualization. The figure presents the process of using two-dimensional images to reconstruct a three-dimensional model. Many manipulations such as zoom, rotation and hyalinization can be conducted through the model of the three-dimensional visualized image. Furthermore, an omni-directional, multi-angle and multi-level view is used to clarify the characteristics of the organization structure.

In teaching settings, realistic scenes and dynamic interactive settings are created and supplied by virtual reality technology. Moreover, the learners show higher learning motivation and participation with the use of virtual reality technology. More specifically, traditional medical teaching is not able to display the shape of the blood vessels in a comprehensive and realistic manner such that students do not have a detailed understanding of the blood vasculum. Virtual reality technology supports the occurrence of situational learning, which can offer feedback from multi-channels (such as hearing, vision, touch, etc.) to aid the learners in transferring the learning in a virtual situation into real life and meet the needs of situational learning. Currently, this three-dimensional animation technology has great potential for use in situational teaching methods in which the blood vasculum is shown in the form of three-dimensional animation to promote better engagement of student learning activities. The novel framework we proposed can be used in practice to improve teaching.

B. OBJECTIVE EVALUATIONS OF EXPERIMENTAL RESULTS

To verify the effectiveness of the proposed framework, the performance is evaluated against two indicators, the fps (frames per second) and the time to produce animations. The fps refers to the maximum frame rates that can be achieved. In our work, the time is calculated from when a request is sent at the browser to when an animation is loaded at the webpage end. A timer is placed at the webpage end to achieve an accurate timing effect.

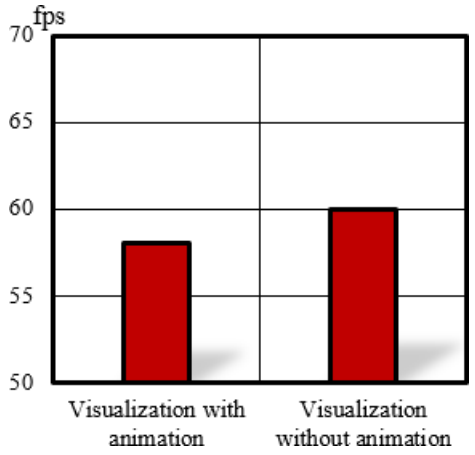

FIGURE 9. A contrasting experiment on vascular visualization in fps based on the browser. The results of visualization with animation and visualization without animation are included.

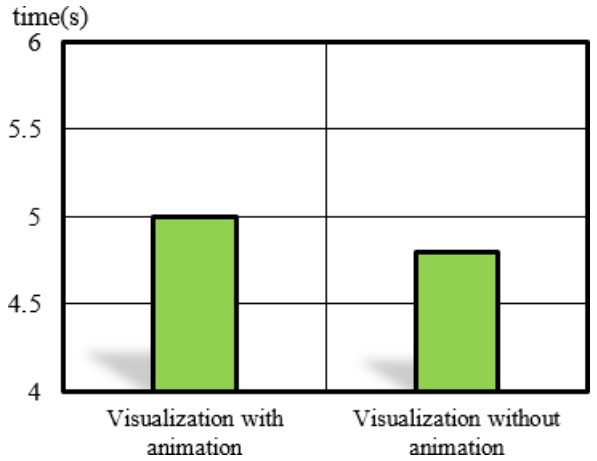

FIGURE 10. A contrasting experiment on vascular visualization and time based on the browser. The results of visualization with animation and visualization without animation are shown.

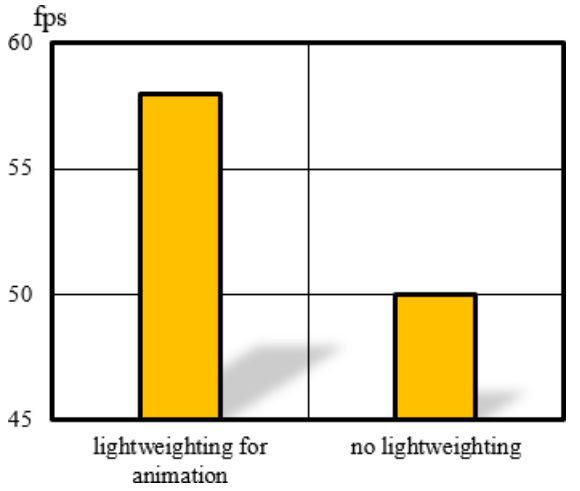

FIGURE 11. A contrasting experiment of vascular visualization in fps based on the browser. The results of lightweighting for animation and no lightweighting are included.

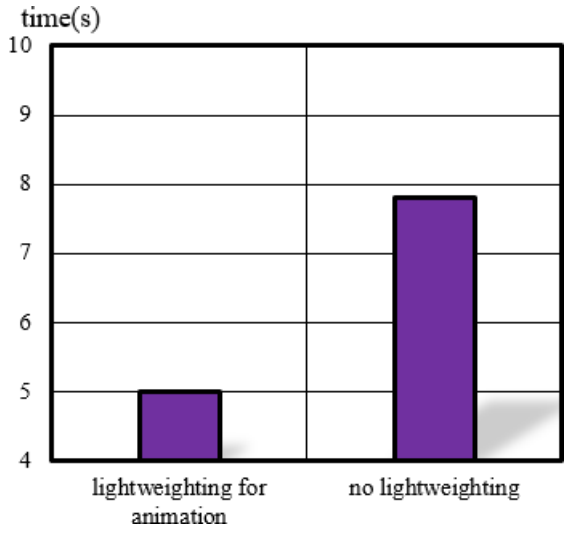

FIGURE 12. A contrasting experiment of vascular visualization in time based on the browser. The results of lightweighting for animation and no lightweighting are included.

Figure 9 shows a comparison of the vascular visualization with animation and without animation in fps based on the browser. Figure 10 displays a contrasting result of vascular visualization in time based on the browser, which includes visualization with animation and visualization without animation. It can be observed from Fig. 9 and Fig. 10 that a subtle difference exists between two methods in terms of fps and time. This result shows that the animation task in our method does not have a great impact on vascular visualization performance, but it greatly improves the user experience and enriches the vascular visualization.

Fig. 11 and Fig. 12 show contrasting experiments on vascular visualization in fps and time, respectively, based on the browser. Both figures contain the results from lightweighting for animation and from no lightweighting. The results show that the method of lightweight animation can increase the fps to a maximum, reduce the processing time required by the browser, and hence improve the user experience.

Fig. 13 shows the final rendering of the vascular vessels. Fig. 13(1) to (4) show a series of operations via vascular visualization based on the browser. The user can interact with any vessel by dragging, rotating and zooming. The blood capsule marked in red has a simulated circular animation motion.

C. DISCUSSION

The proposed framework includes three components namely, lightweight vasculum, Catmull-Rom spline-based blood-path plan and recovery and rendering in the client-side web browser. A mesh folding-based progressive approach is used in the lightweight vasculum. This method can reduce the amount of data that a browser needs for rendering and enables progressive rendering at the browser side. The blood-path

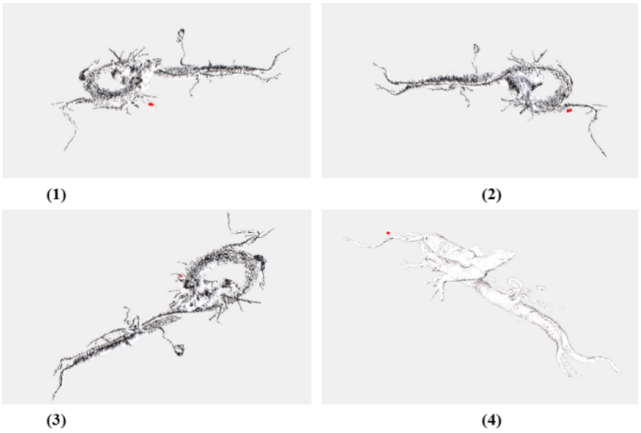

FIGURE 13. Final effects of vascular visualization based on a browser representing four different states.

plan is formed based on the Catmull-Rom spline to output key frames and centroid coordinates. Moreover, progressive transmission is used to prioritize the transmission such that the user waiting time can be shortened.

Via progressive transmission, users can quickly obtain a lower resolution and sketched expression of the vasculum at the beginning. With the accumulation of low-level detail data transmitted to the client side, the sketched expression is dynamically enriched with higher resolution and finer expression. This type of visual expression of the gradual process from sketched to fine can satisfactorily conform to our understanding of the spatial information of users and can also facilitate information navigation. Moreover, the needs of users related to cognition of spatial information can be satisfied by this type of expression such that a more comprehensive and meticulous understanding of the virtual blood vasculum is produced.

As a collection of simulations, computer graphics and human-computer interface technologies, virtual reality offers new opportunities and breaks through the restrictions implied by traditional medical education. This application is rewarding for innovating educational conceptions, expanding education space, improving teaching quality and efficiency, and conserving consumables. As a result, virtual reality technology is expected to prompt the reform and development of medical education in the near future.

V. CONCLUSION

This paper proposes a novel WebVR progressive mesh visualization framework for virtual blood vasculum. Due to limited computing capability, network bandwidth and caching space of browsers, achieving a realistic visualization effect is an arduous process with conventional methods. Our approach uses a lightweight treatment for the blood vasculum such that vessels can be represented and loaded progressively at the browser side. With reduced one-off rendering data, users can experience a real-time response with the most paramount information and gradually access finer details to ensure a good user experience. Furthermore, a spine algorithm for blood-path planning is developed to animate the fluent and smooth movement of blood in the vessels. The validity and outstanding performance of our framework were verified by experiments to vividly display the morphology of the virtual blood vasculum on the web-browser side.

REFERENCES

[1] J. M. Erdman, "Reference in a 3-D virtual world: Preliminary observations on library outreach in 'second life,'" *Reference Librarian*, vol. 47, no. 2, pp. 29–39, 2009.
[2] C. LaGuardia, "Virtual dreams give way to digital reality," *Library J.*, vol. 120, pp. 42–44, Oct. 1995.
[3] K. Ogatal *et al.*, "Visualization of three-dimensional cardiac electrical excitation using standard heart model and anterior and posterior magneto-cardiogram," *Int. J. Cardiovascular Imag.*, vol. 22, pp. 581–593, Jun. 2006.
[4] M. Zhang, Z. Lv, X. Zhang, G. Chen, and K. Zhang, "Research and application of the 3D virtual community based on WEBVR and RIA," *Comput. Inf. Sci.*, vol. 2, no. 1, pp. 84–89, 2009.
[5] H. Hoppe, "Progressive meshes," in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn.*, 1996, pp. 99–108.
[6] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proc. 24th Annu. Conf. Comput. Graph. Interact. Techn.*, 1997, pp. 209–216.
[7] L. Kettner, "Using generic programming for designing a data structure for polyhedral surfaces," *Comput. Geometry*, vol. 13, no. 1, pp. 65–90, May 1999.
[8] S. Morigi and M. Rucci, "Multilevel mesh simplification," *Vis. Comput.*, vol. 30, pp. 479–492, May 2014.
[9] L. Wen, N. Xie, and J. Jia, "Fast accessing Web3D contents using lightweight progressive meshes," *Comput. Animation Virtual Worlds*, vol. 27, pp. 466–483, Sep. 2016.
[10] S. Zhao, W T. Ooi, A. Carlier, G. Morin, and V. Charvillat, "3D mesh preview streaming," in *Proc. ACM Multimedia Syst. Conf. Ser.*, 2013, pp. 178–189.
[11] G. Lavoué, L. Chevalier, and F. Dupont, "Streaming compressed 3D data on the Web using JavaScript and WebGL," in *Proc. 18th Int. Conf. 3D Web Technol.*, 2013, pp. 19–27.
[12] A. Evans, M. Romeo, A. Bahrehmand, J. Agenjo, and J. Blat, "3D graphics on the Web: A survey," *Comput. Graph.*, vol. 41, pp. 43–46, Jun. 2014.
[13] B. Kronrod and C. Gotsman, "Optimized compression of triangle mesh geometry using prediction trees," in *Proc. Int. Symp. 3D Data Process. Vis. Transmiss.*, 2002, pp. 602–608.
[14] T. Lewiner, M. Craizer, H. Lopes, S. Pesco, L. Velho, and E. Medeiros, "GEncode: Geometry-driven compression for general meshes," *Comput. Graph. Forum*, vol. 25, no. 4, pp. 685–695, 2006.
[15] C. L. Bajaj, V. Pascucci, and G. Zhuang, "Progressive compression and transmission of arbitrary triangular meshes," in *Proc. IEEE Vis.*, Oct. 1999, pp. 307–537.
[16] P. Diazgutierrez *et al.*, "Hierarchyless simplification, stripification and compression of triangulated two-manifolds," *Comput. Graph. Forum*, vol. 24, no. 3, pp. 457–467, 2005.
[17] V. Coors and J. Rossignac, "Delphi: Geometry-based connectivity prediction in triangle mesh compression," *Vis. Comput.*, vol. 20, pp. 507–520, Nov. 2004.
[18] S. Gumhold, "Optimizing Markov models with applications to triangular connectivity coding," in *Proc. Symp. Discrete Algorithms*, 2005, pp. 331–338.
[19] K. Mamou, T. Zaharia, and F. Prêteux, "TFAN: A low complexity 3D mesh compression algorithm," *Comput. Animation Virtual Worlds*, vol. 20, pp. 343–354, Jun. 2009.
[20] M. Okazaki, S. Takayama, A. Seki, H. Ikegami, and T. Nakamura, "Posterolateral rotatory instability of the elbow with insufficient coronoid process of the ulna: A report of 3 patients," *J. Hand Surg.*, vol. 32, pp. 236–239, Feb. 2007.
[21] J. Sanehez-Sotelo, S. W. O'Driscoll, and B. F. Morrey, "Medial oblique compression fracture of the coronoid process of the ulna," *J. Shoulder Elbow Surg.*, vol. 14, pp. 60–64, Jan. 2005.
[22] W. Jiao *et al.*, "Application of MIMICS software to 3D reconstruction of medical image," *Chin. Med. Equip. J.*, vol. 36, pp. 115–118, 2015.

[23] M. Bertolotto and M. J. Egenhofer, "Progressive transmission of vector map data over the world wide Web," *Geoinformatica*, vol. 5, pp. 345–373, Dec. 2001.
[24] H. Han, V. Tao, and H. Wu, "Progressive vector data transmission," presented at the 6th Assoc. Geograph. Inf. Lab. Eur., 2003.
[25] T. Ai, Z. Li, and Y. Liu, "Progressive transmission of vector data based on changes accumulation model," presented at the 11th Int. Symp. Spatial Data Handling, 2004.

CHENXI HUANG received the B.Sc. degree in computer science from Tongji University, Shanghai, China, in 2015, where he is currently pursuing the Ph.D. degree in computer science. His research interests include image processing, image reconstruction, data fusion, 3-D visualization, and machine learning.

WEN ZHOU is currently pursuing the Ph.D. degree with the School of Software Engineering, Tongji University. His research interests include Web3-D, sketch-based retrieval, and machine learning.

YISHA LAN is currently pursuing the B.Sc. degree with Tongji University, Shanghai, China. Her research interests include image processing, reconstruction, and 3-D visualization.

FEI CHEN received the Ph.D. and M.D. degrees from the Medical College, Zhejiang University, China. He is currently a Fellow Doctor with Tong Hospital, Tong University, Shanghai. His research interests include the pathophysiological proceeding of atherosclerosis, interventional treatment of CHD, image reconstruction of coronary stent, and 3-D visualization of coronary stent.

YONGTAO HAO received the Ph.D. degree in mechanical engineering from Shanghai Jiaotong University. He is currently a Professor with the College of Electronics and Information Engineering, Tongji University. He has authored 40 articles and holds six patents. His research interests include intelligent design, machine learning, data mining, Industry 4.0, integrated knowledge, and virtual reality. He is a Senior Member of the Council for the Chinese Mechanical Engineering Society.

YONGQIANG CHENG is currently a Lecturer with the School of Engineering and Computer Science, University of Hull, U.K. His research interests include digital healthcare technologies, embedded systems, control theory and applications, artificial intelligence, and data mining.

YONGHONG PENG (M'02) is currently a Professor of data science (Chair) and the Head of data science research with the University of Sunderland, U.K. His research areas include data science, machine learning, data mining, and artificial intelligence. He is a member of the Data Mining and Big Data Analytics Technical Committee of the IEEE Computational Intelligence Society and the Chair of the Big Data Task Force. He is also a Founding Member of the Technical Committee on Big Data of the IEEE Communications Society and an Advisory Board Member of the IEEE Special Interest Group on Big Data for Cyber Security and Privacy. He is an Associate Editor of the IEEE Transactions on Big Data and an Academic Editor of the *PeerJ* and the *PeerJ Computer Science*.