IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 15, 2018, accepted November 20, 2018, date of publication November 29, 2018,
date of current version December 27, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2883679

A Surface Mass-Spring Model With New Flexion Springs and Collision Detection Algorithms Based on Volume Structure for Real-Time Soft-Tissue Deformation Interaction

CHUNQUAN LI[1,2], JIAJUN DING[1], ZHICHAO HONG[1], YUCHENG PAN[1], AND PETER X. LIU[1,2], (Senior Member, IEEE)

[1]School of Information Engineering, Nanchang University, Nanchang 330029, China
[2]Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada

Corresponding author: Chunquan Li (lichunquan@ncu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61503177, Grant 81660299, and Grant 61863028, in part by the China Scholarship Council under the State Scholarship Fund under Grant CSC 201606825041, in part by the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences under Grant Y6S9011F58, in part by the Science and Technology Department of Jiangxi Province of China under Grant 20161ACB21007, Grant 20171BBE50071, and Grant 20171BAB202033, and in part by the Education Department of Jiangxi Province of China under Grant GJJ14228 and Grant GJJ150197.

ABSTRACT A critical problem associated with surgical simulation is balancing deformation accuracy with a real-time performance. Although the canonical surface mass-spring model (MSM) can provide an excellent real-time performance, it fails to provide effective shape restoration behavior when generating large deformations. This significantly influences its deformation accuracy. To address this problem, this paper proposes a modified surface MSM. In the proposed MSM, a new flexion spring is first developed to oppose bending based on the included angle between the initial position vector and the deformational position vector, improving the shape restoration performance, and enhance the deformational accuracy of MSM; then, a new type of surface triangular topological unit is developed for enhancing the computational efficiency and better adapting to the different topological soft tissue deformational models. In addition, to further improve the accuracy of deformational interactions between the soft tissue and surgical instruments, we also propose two new collision detection algorithms. One is the discrete collision detection with the volumetric structure (DCDVS), applying a volumetric structure to extend the effective range of collision detection and the other is the hybrid collision detection with the volumetric structure (HCDVS), introducing the interpolation techniques of the continuous collision detection to DCDVS. Experimental results show that the proposed MSM with the DCDVS or the HCDVS can achieve accurate and stable shape restoration and show the real-time interactive capability in the virtual artery vessel and heart compared with the canonical surface MSM and new volume MSM.

INDEX TERMS Mass-spring model (MSM), flexion spring, shape restoration, collision detection, soft-tissue deformation, surgical simulation.

I. INTRODUCTION

Surgical simulation is a sophisticated virtual reality technology that simulates real surgical processes for training medical students or surgeons in the medical field by using computers [1]. It allows medical students and surgeons to improve their operating skills in subtle surgical procedures without potentially risking a patient's life. In addition, it can also save training time and enhance training efficiency for medical students and surgeons [2]. Thus, many virtual surgery simulators have been developed for some special operations such as neurosurgery [3], [4], laparoscopic cholecystectomy [5] and lung-related surgeries [6]. However, a crucial issue associated with surgical simulation is how to simulate realistic interaction behavior between soft tissue and surgical instruments such as realistic deformation interaction from visualization and haptic perception [7], [8].

To obtain this goal, some physics-based elastic deformation models have been used instead of geometry-based models in surgical simulation to improve visualization and haptic perception fidelity [9] because physics-based models that consider continuum mechanics can achieve accurate simulation by using numerical methods to solve the partial differential equations (PDEs). Currently, various physics-based elastic deformation models are employed to simulate the mechanical behavior of soft tissues. These models are generally divided into two categories: the finite element method (FEM) and the mass-spring model (MSM).

FEM is a continuum-based method and is relatively accurate since it has a powerful biomechanical relevance. However, its high computational cost makes real-time interaction prohibitive. To obtain real-time capabilities, linear FEMs with condensation and precomputation techniques are employed to rapidly compute the strains in the elastic deformation models for surgical simulation [10]–[15]. However, it is difficult for linear FEMs to simulate the large deformations and rotations in the virtual surgical simulation. To overcome these difficulties, Georgii and Westermann [16] proposed the corotated FEM technique, which can divide the displacement of an element into two parts: the rigid solid rotation and the deformation. Some similar approaches were applied in [17] and [18]. However, the corotated FEM technique still retains linear elastic behavior, which may cause the rare reality for soft tissue simulation. To achieve accuracies for large strains or rotations, Miller *et al.* [19] proposed combining explicit time integration with FEM to eliminate solving iterative equations during the time step procedure. Sangpradit *et al.* [20] presented a rolling finite element model (RFEM) for simulating rolling tissue indentation in a relatively short time. Tang and Wan [21] proposed a strain-based constraint FEM to efficiently simulate nonlinear homogeneous soft tissues in an interactive simulation. Although the abovementioned FEMs have achieved some improvements in computational accuracy and speed, they can be used for real-time interaction in only relatively small and simple models.

MSM is another physics-based elastic deformation model based on Hooke's law and is discretized into many mass points interconnected to each other with massless springs. Although there is the main limitation for the MSM in biomechanical accuracy, its simple mathematical frame, high versatility for topological changes, and computational frame are well suited for parallel computing and are all appealing advantages over FEMs in surgical simulation [29]–[31]. In addition, to improve the biomechanical accuracy of the MSM, a number of variant MSMs have been developed, mainly focusing on designing or optimizing their shape and topology, describing the realistic mechanical behavior of actual soft tissue including nonlinear elasticity, and balancing real-time computation and accuracy for surgical simulation.

Initially, the canonical surface MSM was mainly used for simulating woven fabrics such as for animating cloth objects [22]. The canonical surface MSM consists of a mesh with many mass points, and each mass point is linked to its neighbors by massless springs with a natural length that is not equal to zero; these springs can be divided into structural springs, shear springs, and flexion springs [22]. Subsequently, the canonical surface MSM is introduced or improved to simulate the deformable surface tissues in surgical simulation, such as the craniofacial or facial part [23], [24] and virtual severed blood vessels [25]. Nedel and Thalmann [26] simulated muscle tissue using a surface MSM with additional angular springs to preserve the muscle volume shape, but the additional angular springs could effortlessly influence the material properties, potentially resulting in a stiffer model. To improve the realism and real-time performance of the virtual surgical simulation, Bao *et al.* [27] proposed a dynamic surface MSM, using the virtual spring to simulate the volumetric data of soft tissue. However, their model did not consider the problem that the linear elastic model may lead to a super-elasticity effect. Wang *et al.* [28] presented a modified surface MSM, introducing a rigid core and new springs to preserve the volumetric and shape information of soft tissue. In addition, the super-elasticity influence of linear elastic models was also corrected by using inverse dynamics. However, its simulation precision needs to be further enhanced owing to the rigid core. Although the abovementioned MSMs based on surface mesh have a simple structure and high computational speed, their ability to simulate soft tissue, including simulating nonlinear elasticity and shape restoration, needs to be further improved. In particular, when the surface MSMs generate large deformation, it is hard for surface MSMs to restore the original shape.

Furthermore, the volumetric mesh MSM was developed to improve the nonlinear and shape restoration for simulating soft tissue deformation. The topological structure of the volumetric mesh MSM is composed of many tetrahedral or cubic units, each of which is in turn composed of springs and mass points. Mollemans *et al.* [29] proposed a tetrahedral MSM. However, this technique is not suitable for real-time simulation due to its high computation cost. San-Vicente *et al.* [30] proposed a cubic MSM with linear springs to approximately represent the complex nonlinear behavior of living tissues, in which the spring orientations were categorized into vertices, diagonal and internal diagonal. Duan *et al.* [31] introduced novel constraints on positions in a tetrahedral MSM to simulate nonlinearity and shape restoration in order to realistically simulate incompressibility. However, compared with surface MSMs, However, both the cubic and tetrahedral MSM has too many springs and thus still have a high computational cost, especially in simulating complex soft tissue models.

Unlike FEMs, which are formed according to the elasticity theory, MSMs rely on the structure of the spring mesh, and it is difficult to achieve suitable spring stiffness and damping. Therefore, Lloyd *et al.* [32] developed a method to identify the isotropic linear spring parameters with triangular, rectangular and tetrahedral meshes. Moreover, some investigations also are performed for improving the time integration techniques of mass-spring systems. For instance,

Liu *et al.* [33] developed a new numerical approach based on block coordinate descent for implicit Euler time step of the mass-spring system, which can approximate the solution in a limited amount of computational time.

In addition, since most biological soft tissues are characteristic by a nonlinear mechanical response, some researchers have attempted to introduce nonlinear springs into the MSM to achieve more realistic simulations. Cooper and Maddock [34] developed nonlinear springs with a polynomial formulation. Teschner *et al.* [35] presented a multilayered spring configuration with several polynomial coefficients to simulate soft tissue deformation during surgical procedures. Basafa *et al.* [36] presented a nonlinear MSM with its parameters tuned to simulate the nonlinear and viscoelastic behaviors of typical living soft tissues. Huangfu *et al.* [37] developed a tetrahedral MSM with nonlinear spring equations by using the dynamic local simulation solution algorithm based on the modified explicit Euler integration to solve the deformation equation. Omar *et al.* [38] developed an MSM by using the stiffness variation in a conical spring to simulate the nonlinear deformation of soft tissues. However, it is more difficult for MSMs with nonlinear springs to accurately estimate their parameters, which greatly restricts their flexibility in the virtual surgical simulation.

Some other methods have also been employed to simulate deformations of soft tissue. James and Pai [11] proposed the boundary element method (BEM) to simulate deformation. Owing to the hypothesis of isotropic linear elasticity for the deformation model, this method may result in unrealistic deformation effects in a simulation with large deformation. Ahmadian and Nikooyan [39] reported a method utilizing the empirical formulas derived from practical experiments for the deformation of soft tissue, in which the deformation behaviors of the different soft tissues are described using different mathematical formulations with the optimization technique. However, its deformational accuracy greatly depends on the experimental results from soft tissues. Furthermore, the experimental results from soft tissues also greatly constrain its extensibility. Horton *et al.* [40] developed a meshless method on the basis of the total Lagrangian formulation to simulate the deformation of very soft tissue. Due to the explicit time integration with small time steps being employed to stabilize the simulation, this approach has some limitations in real-time simulations. Liu *et al.* [8] presented a hybrid method, combining the advantages of the MSM and FEM. However, this method has a high computational cost for solving continuum constitutive equations.

For a surgical simulation system, the elastic deformation model should simultaneously achieve both accurate deformational behavior and good real-time interaction. However, the various elastic deformation models presented above demonstrate that simultaneously achieving both goals is difficult. For instance, the canonical surface MSM [22] has a simple structure and high computational speed but has difficulty restoring its original shape when large deformation is generated. This result occurs because the canonical surface MSM failed to oppose bending in terms of the included angle between the initial position vector and deformational position vector. Therefore, the key issue for developing an elastic model is maintaining a reasonable balance between accurate deformational behavior and real-time performance in a surgical simulation.

In addition, the accuracy of the deformational interaction behavior also relies on the interactive collision detection in the surgical simulation because a single missed collision might result in an unacceptable simulation or remarkable distortion [41], [42]. Most of the earlier work in interactive collision detection was based on discrete collision detection (DCD) algorithms [43]–[46]. Since the DCD algorithms are only executed at a discrete interval of the simulation, they have the low computational cost for interactive collisions. However, owing to checking for collisions at a discrete instance, the DCD algorithms may lead to missed collisions with fast movements or very large deformations [56]. To address these problems, the continuous collision detection (CCD) algorithms have been developed. By employing continuous paths and then detect them for overlaps, CCD algorithms provides the simulation of the motion between the discrete time steps. These algorithms involve interactive algorithm for rigid models [47], articulated model [48], [49], and deformable models [50], [51]. Nevertheless, CCD algorithms need to provide the higher computational cost than DCD. To reduce the computational cost, various acceleration algorithms are designed. Some of them contain eliminating redundant elements [52], normal cone culling [53], and coplanarity-based culling [54]. However, such acceleration algorithms for the CCD algorithm are mainly limited to the polygonal models of surface meshes [56]. To accelerate the continuous collision detection on volume meshes, Sud *et al.* [55] developed a fast proximity computation among multiple deformable models by using voronoi-based culling with N-body distance query. Tang *et al.* [56] proposed new culling algorithm, which contains a continuous separating axis test and eliminating redundant elementary tests between the characteristic of volume elements to execute fast and robust continuous collision checking between deforming volume meshes. However, the CCD algorithms are unable to provide enough high real-time interactive performance when the scale of deformable models are larger [57]. Moreover, Faure *et al.* [58] and Allard *et al.* [59] developed some image-space algorithms based on GPU rasterization techniques for interactive collision detection. However, these algorithms can cause missed collisions due to the discrete image-space representation. In addition, Bounding volume hierarchies (BVHs) have been extensively applied to improve the performance of collision detection between rigid and deformable models. Some typical BVH approaches are axis aligned bounding box (AABB) trees [60], sphere trees [61], oriented bounding box (OBB) trees [62], and discrete orientation polytopes (DOPs) [63]. These typical collision detection algorithms have provided effective detection of deformational bodies. However, these typical BVH approaches need to continuously

update their bounding volume for the large deformable models, leading to the high computational cost. Consequently, it still remains some challenges for the collision detection algorithms to provide both the detection accuracy and high real-time interactive performance for the large deformable models.

In this paper, we focus on the shape restoration problem and real-time interaction of the surface MSM to achieve a balance between real-time performance and accurate deformational behavior in a surgical simulation. Specially, the shape restoration performance denotes that after the deformations of MSM are over, the MSM can restore the degree of its initial shape that it has before it never generates deformation. It is a great crucial issue for a surface MSM to offer an excellent shape restoration capability because the shape restoration has a great impact on its deformation accuracy. In addition, to achieve accurate real-time interactions in the surgical simulation, effective collision detection algorithms are indispensable for the surface MSM. Since the DCD algorithms have low computational costs and provide great real-time interactive collision detections, it generally is used to provide the collision detection for the surgical simulation. However, the DCD algorithms cause missed or inaccurate detections with fast movements, great large deformations, or surgical with very small sizes. Although CCD algorithms can provide accurate collision detection by using continuous interpolations, these algorithms have high computational costs and are usually used to detect high-speed collision such as the collision detection for a car crash or high-speed bullet [56]. Specially, surgical operations are not particularly fast with only local deformations or the fast movements for only the local soft tissue. Therefore, it is another great crucial issue to develop effective collision algorithms that can provide an effective balance between the real-time interaction and detection accuracy in the surgical simulation based on the characteristics of the surgical operations.

To address these problems mentioned above, first, instead of using the volumetric MSM, we propose a modified surface MSM to simulate the volumetric tissue for real-time surgical simulation. In our MSM, a new type of surface triangular topological unit is developed to decrease the computational burden and to better adapt to different topological soft tissue models. Subsequently, to improve the deformational accuracy of our MSM in the surgical simulation, a new type of flexion spring is developed in our surface triangular topological unit, which is capable of opposing bending based on the included angle between the initial position vector and deformational position vector. Specifically, the new flexion spring can greatly improve the shape restoration for our MSM so that the deformational accuracy of our surface MSM can achieve substantial improvement. Finally, we developed two new collision detection algorithms. One is the discrete collision detection based on the volumetric structure (DCDVS), which uses the volumetric structure rather than the surface structure to extend the effective range of collision detection and improve the detection accuracy. The other is a hybrid collision detection based on the volumetric structure (HCDVS), which combines the advantages of the CCD and the proposed DCDVS algorithm to achieve an effective balance between the detection accuracy and real-time interactive performance. Both the algorithms can improve the accuracy of interactions between the surgical instrument and our MSM as well as enable our surface MSM to accurately detect surgical instruments with different sizes. Moreover, experiments on the virtual vessel and heart soft tissue are executed to verify that our proposed surface MSM combined with the volumetric structure interactive algorithm can achieve accurate and stable shape restoration performance as well as real-time interactive capability in surgical simulation.

This paper is arranged as follows: the proposed surface MSM is illustrated in Section II and two different collision detection algorithms are demonstrated in Section III. In Section IV, the performance of the proposed surface MSM and two collision detection algorithms is experimented compared with other state-of-art MSMs and collision detection algorithms. Discussions and conclusions with future works on our algorithms are detailed in section V and Section VI, respectively.

II. PROPOSED MSM ALGORITHM

Generally, MSMs are composed of various topological mesh units, and each mesh unit is composed of mass points and spring units. According to the topological structure of the meshes located on a closed surface or throughout the entire volume space of the soft tissue model, MSMs can be divided into surface and volumetric models. Compared with volumetric MSMs, surface MSMs have the advantages of a simple structure and high computational speed. However, when large deformations are generated in surface MSMs, it is difficult for surface MSMs to achieve good shape restoration. To achieve a balance between accuracy and computational speed in virtual surgical simulation, the proposed MSM including the topological unit and new flexion spring, is developed.

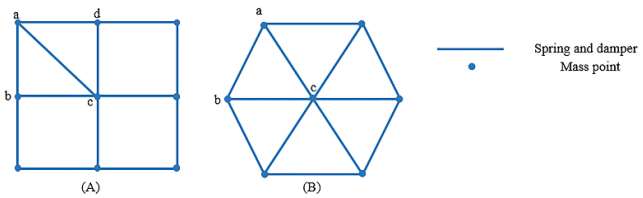

FIGURE 1. Polygons with different topologies can be built with triangular topological units. (A) Quadrilateral *abcd* can consist of triangles *abc* and *acd*. (B) A hexagon can also consist of many triangles, including triangle *abc*.

A. TRIANGULAR TOPOLOGICAL UNIT

It is generally accepted that geometric models in computer graphics are composed of triangles. In addition, the triangular topological structure is the basic unit for describing a plane. For instance, as shown in Fig. 1A, quadrilateral *abcd* can consist of triangles *abc* and *acd*. In the same way, the topology

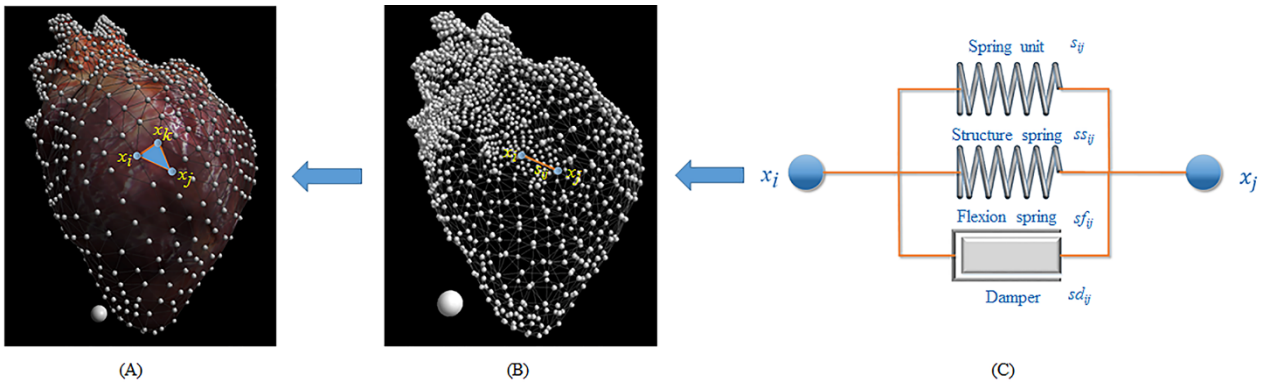

FIGURE 2. Implementation process of the surface MSM based on the virtual heart model. (A) The surface MSM consists of many triangular topological units, and each triangular topological unit consists of three mass points and three spring units, such as the triangle unit $x_i x_j x_k$. (B) The inside structure of the surface MSM including mass points and spring units, such as the mass points x_i and x_j, and the spring unit s_{ij}. (C) The spring unit s_{ij} consists of the structure spring ss_{ij}, damper d_{ij}, and new flexion spring sf_{ij}.

of other polygons can consist of a series of triangles, such as the hexagon in Fig. 1B. Therefore, the proposed surface MSM adopts the triangular topological structure as the basic topological unit to better match different geometric topological structures.

As shown in Fig. 2, taking the irregular virtual heart model for example, we thoroughly demonstrate how to build our surface MSM using triangular topological units. Figs. 2A and 2B show that our surface MSM consists of many triangular topological units; each triangular topological unit consists of three mass points and three spring units; each spring unit connects two adjacent mass points and consists of a structure spring, flexion spring and damper. These springs all follow Hooke's law. For instance, as shown in Figs. 2A and 2B, each of the three blue points from the triangular topological unit $x_i x_j x_k$ represents a mass point of the surface MSM, and each of the orange lines from the triangular topological unit $x_i x_j x_k$ represents a spring unit, such as the spring unit s_{ij}. Furthermore, Fig. 2C shows that the spring unit s_{ij} is composed of the structure spring ss_{ij}, damper sd_{ij}, and new flexion spring sf_{ij}

Without loss of generality, suppose that the virtual heart is discretized into a set of mass points N and a set of spring units M. For two arbitrary adjacent mass points x_i and x_j, $i, j = 1, \ldots, N$ and $i \neq j$, they can be connected by the spring unit $s_{ij} \in M$, which is composed of the structure spring ss_{ij}, damper sd_{ij}, and new flexion spring sf_{ij}. The structure spring ss_{ij} and damper sd_{ij} are the same configurations as those of the canonical surface MSM [22]. According to Hooke's law, the structure spring force is computed at time t as follows:

$$f_{ij}^{s}(t) = k_{ij}^{s}\left[\left|p_j(t) - p_i(t)\right| - l_{ij}(0)\right] \cdot \frac{p_j(t) - p_i(t)}{\left|p_j(t) - p_i(t)\right|}. \quad (1)$$

Here, $f_{ij}^{s}(t)$ is the spring force of the structure spring ss_{ij} acting on the mass point x_i; ss_{ij} is allowed to connect mass points x_i and x_j;k_{ij}^{s} is the elastic coefficient of ss_{ij}; p_i and p_j are the positions of the mass points x_i and x_j, respectively; and $l_{ij}(0)$ is the initial distance between x_i and x_j; $\left|p_j(t) - p_i(t)\right|$ denotes the absolute value or modulus of the vector $p_j(t) - p_i(t)$. Similarly, we define the damping force at time t as follows:

$$f_{ij}^{d}(t) = u_{ij}^{d} \cdot \frac{\partial\left[\left|p_j(t) - p_i(t)\right| - l_{ij}(0)\right]}{\partial t} \cdot \frac{p_j(t) - p_i(t)}{\left|p_j(t) - p_i(t)\right|} \quad (2)$$

where $f_{ij}^{d}(t)$ is the damping force of the damper sd_{ij}, describing the viscous force acting on x_i; the damper sd_{ij} is used to connects x_i and x_j, and $u_{ij}^{d}$is the damping coefficient of sd_{ij}.

B. NEW FLEXION SPRING

Furthermore, it is a significantly important problem for a surface MSM to provide a good shape restoration performance because the shape restoration performance has a great influence on the deformation accuracy of the MSM. Since the canonical surface MSM never has the accurate shape restoration performance, it cannot provide an accurate deformation. To achieve an accurate shape restoration performance, a new type of flexion spring is developed to replace the shearing and flexion spring from the canonical MSM [22], which can greatly improve the deformational accuracy of the surface MSM. Before demonstrating the new flexion spring in detail, let us first consider the disadvantage of the canonical surface MSM [22] involving the shape restoration.

1) DISADVANTAGE OF THE SHAPE RESTORATION FROM THE CANONICAL SURFACE MSM

As shown in Fig. 3, to facilitate understanding, we assume that the mass points x_i and x_k are fixed and that x_j can move in the xy plane and is adjacent to points x_i and x_k. In addition, Fig. 3B shows that the initial equilibrium position of the mass point x_j is described as $p_j(0)$. Fig. 3C shows that the mass point x_j moves to position $p_j(t)$ at time t owing to external forces acting on it. After that time, the mass point x_j does not stop moving until its kinetic energy is exhausted. In this case, the mass point x_j might stop at another equilibrium position $p_j(t_e)$ at time t_e, as shown in Fig. 3D. In other words, the mass point x_j may be unable to return to the initial position $p_j(0)$.

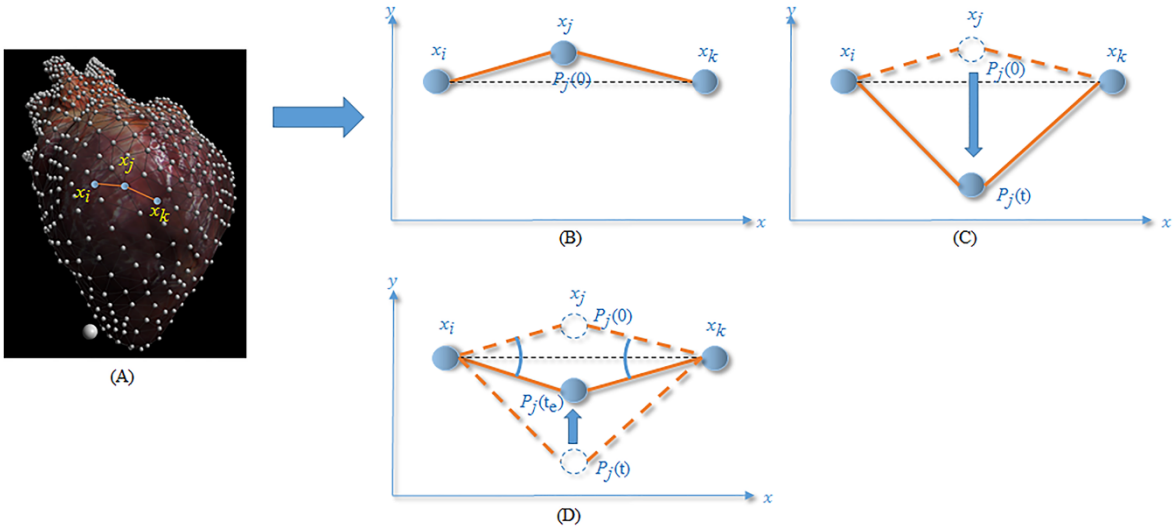

FIGURE 3. Disadvantage of the shape restoration from the canonical surface MSM. (A) Mass points x_i, x_k, and x_j are adjacent to each other on the surface MSM based on the heart model. (B) Assume that the mass points x_i and x_k are fixed and that the mass point x_j can move in the xy plane and has an initial position of $p_j(0)$. (C) The mass point x_j moves to position $p_j(t)$ at time t. (D) The mass point x_j stops at another equilibrium position $p_j(t_e)$ at time t_e.

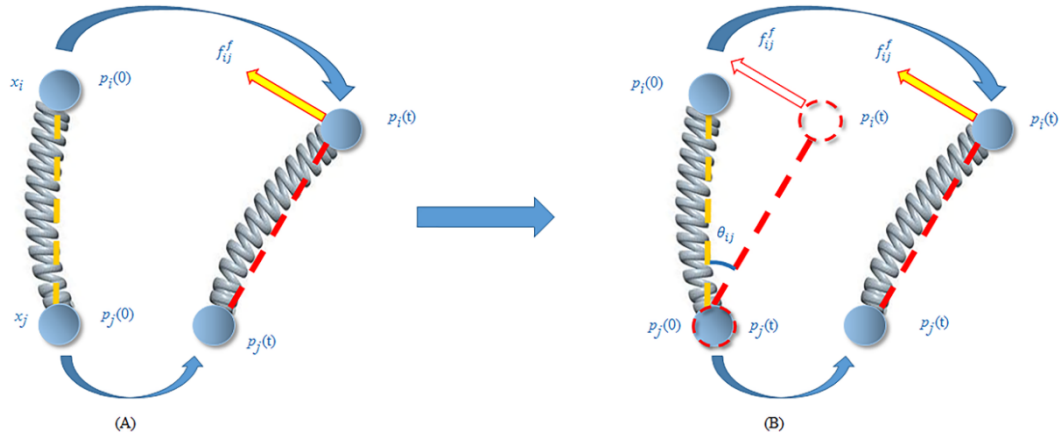

FIGURE 4. Schematic diagram of the new flexion spring. (A) The initial positions of the mass points x_i and x_j are $p_i(0)$ and $p_j(0)$, respectively; mass points x_i and x_j are located at positions $p_i(t)$ and $p_j(t)$ at time t, respectively. The spring force f_{ij}^f can enable the mass point x_i to return to the original position $p_i(0)$ from the current position $p_i(t)$. (B) The included angle θ_{ij} between vectors $p_j(0) - p_i(0)$ and $p_j(t) - p_i(t)$ is not equal to zero.

Similarly, it is easy to understand that if the mass points x_i and x_k are not fixed, the mass point x_j might also be unable to return to its initial position $p_j(0)$. Therefore, the canonical surface MSM might not recover its original shape with deformation, especially large deformation.

2) PROPOSED FLEXION SPRING

To overcome the abovementioned problem, we propose a new flexion spring, which is inspired from such a fact that most biological tissues can oppose bending according to the included angle between initial position vector and deformational position vector, so they can enable their connecting mass points to return to their original position from the deformational position with the external force being removed from the biological tissues. The principle of the proposed flexion spring is described as follows.

As shown in Fig. 4, $p_i(0)$ and $p_j(0)$ are the original positions of the adjacent two mass points x_i and x_j, respectively. Due to the forces acting on the mass points x_i and x_j, these points can move to positions $p_i(t)$ and $p_j(t)$ from positions $p_i(0)$ and $p_j(0)$ at time t, respectively. It can be observed in Fig. 4(B) that the included angle θ_{ij} between vectors $p_j(0) - p_i(0)$ and $p_j(t) - p_i(t)$ is not equal to zero when position $p_j(t)$ from the vector $p_j(t) - p_i(t)$ is translated to position $p_j(0)$. At this point, since the flexion spring sf_{ij} acts on the mass point x_i, the corresponding spring force f_{ij}^f tries to enable the mass point x_i to return to the original position $p_i(0)$ from the current position $p_i(t)$.

Similarly, the flexion spring sf_{ij} acting on point x_j is the same resistance as that acting on point x_i. Therefore, when the external forces exerted on the soft tissue model are removed, the forces f_{ij}^f and f_{ji}^f acting on points x_i and x_j can enable the positons $p_i(t)$ and $p_j(t)$ to return back to the original positions $p_i(0)$ and $p_j(0)$, respectively. In other words, along with the external forces being removed, the new flexion spring sf_{ij} can cause the vector $p_j(t) - p_i(t)$ to return its initial state $p_j(0) - p_i(0)$ by enabling the included angle θ_{ij} between the vectors $p_j(0) - p_i(0)$ and $p_j(t) - p_i(t)$ to be equal to zero. In the same way, each new flexion spring from our MSM has the same effect as the flexion spring sf_{ij}. Because these flexion springs are capable of opposing bending according to the included angle between initial position vector and deformational position vector, they can enable their connecting mass points to return to their original position from the deformational position with the external force being removed from the MSM. Hence, this model can efficiently improve the deformational accuracy and achieve high shape restoration for our MSM. According to the above principle, the new flexion spring force f_{fij} at time t can be computed as follows:

$$f_{ij}^f(t) = k_{ij}^f \cdot \frac{\theta_{ij}}{|p_j(0) - p_i(0)|} \cdot \frac{d_{ij}}{|d_{ij}|} \tag{3}$$

where k_{ij}^f is the elastic coefficient of the flexion spring sf_{ij}; θ_{ij} is the included angle between vectors $p_j(0) - p_i(0)$ and $p_j(t) - p_i(t)$; $|p_j(0) - p_i(0)|$ is the absolute value or modulus of the vector $p_j(0) - p_i(0)$; d_{ij} is the directional vector of the flexion spring force $f_{ij}^f(t)$ as follows:

$$d_{ij} = \frac{[p_j(t) - p_i(t)] \otimes [p_j(0) - p_i(0)] \otimes [p_j(t) - p_i(t)]}{|[p_j(t) - p_i(t)] \otimes [p_j(0) - p_i(0)] \otimes [p_j(t) - p_i(t)]|} \tag{4}$$

where the directional vector d_{ij} is in the plane where the cross product ($\otimes$) between vectors $p_j(0) - p_i(0)$ and $p_j(t) - p_i(t)$ is located and is also perpendicular to the vector $p_j(t) - p_i(t)$, its direction being from the vector $p_j(t) - p_i(t)$ to $p_j(0) - p_i(0)$. Note that the included angle θ_{ij} is defined as

$$\theta_{ij} = \arccos\left\{ t\frac{[p_j(t) - p_i(t)] \odot [p_j(0) - p_i(0)]}{|p_j(t) - p_i(t)| \cdot |p_j(0) - p_i(0)|} \right\} \tag{5}$$

where $\odot$ represents dot multiplication or inner product between vectors $p_j(t) - p_i(t)$ and $p_j(0) - p_i(0)$; $|p_j(t) - p_i(t)|$ and $|p_j(0) - p_i(0)|$ denote the absolute value or modulus of vectors $p_j(t) - p_i(t)$ and $p_j(0) - p_i(0)$, respectively.

C. DEFORMATIONAL COMPUTING FOR PROPOSED SURFACE MSM

Suppose that mass point x_i is adjacent to mass point x_j with the spring unit s_{ij}. Then, according to formulas (1), (2), and (3), the internal force $f_{ij}^{in}(t)$ of the spring unit s_{ij} acting on mass point $x_i(t)$ at time t is described as follows:

$$f_{ij}^{in}(t) = f_{ij}^s(t) + f_{ij}^f(t) + f_{ij}^d(t). \tag{6}$$

Similarly, mass point x_i is linked with n mass points through n spring units. Therefore, n spring units acting on mass point x_i can generate the total internal force $f_i^{in}(t)$, which can be computed at time t as follows:

$$f_i^{in}(t) = \sum_{j=1}^{n} f_{ij}^{in}(t) = \sum_{j=1}^{n} \left(f_{ij}^s(t) + f_{ij}^f(t) + f_{ij}^d(t) \right). \tag{7}$$

For any mass point x_i, $i = 1, \ldots, N$, there is a mass $m_i \in R$, an external force $f_i^{ext}(t) \in \mathbb{R}^3$ and an internal force $f_i^{in}(t) \in \mathbb{R}^3$ applied on point m_i. The external force $f_i^{ext}(t)$ acting on point m_i includes gravity and air friction. According to Newton's law, the dynamic equation of any mass point x_i from our surface MSM can be given at time t as follows:

$$m_i a_i(t) = f_i^{ext}(t) + f_i^{in}(t) \tag{8}$$

where $a_i(t) \in \mathbb{R}^3$ is the acceleration of mass point x_i at time t and can be computed as

$$a_i(t) = \frac{f_i^{ext}(t) + f_i^{in}(t)}{m_i}. \tag{9}$$

Suppose the time interval is Δt; then, the position $p_i(t + \Delta t)$ is

$$p_i(t + \Delta t) = p_i(t) + v_i(t) \cdot \Delta t \tag{10}$$

and the velocity $v_i(t + \Delta t)$ of the mass point x_i at time $t + \Delta t$ can be given as

$$v_i(t + \Delta t) = v_i(t) + a_i(t) \cdot \Delta t. \tag{11}$$

Note that $v_i(t) \in \mathbb{R}^3$ and $p_i(t) \in \mathbb{R}^3$ are the velocity and position of mass point x_i at time t, respectively.

The deformational computing steps for our surface MSM are summarized as follows. Suppose that the initial time is t_0. Initialize the position $p_i(t_0)$, velocity $v_i(t_0)$, and mass m_i for each mass point x_i, $i = 1, \ldots, N$ from our MSM. The internal force $f_i^{in}(t_0)$ and external force $f_i^{ext}(t_0)$ for each mass point x_i can be obtained according to formulas (7) and (8) with $p_i(t_0)$, $v_i(t_0)$, and m_i. For each mass point x_i, $i = 1, \ldots, N$, the corresponding position p_i and velocity v_i can be iteratively updated in the next time step $t + \Delta t$ using formulas (9), (10) and (11). Therefore, the deformation of the proposed surface MSM can be dynamically simulated.

III. PROPOSED COLLISION DETECTION ALGORITHMS

The DCD algorithms check for collisions at a discrete instance, resulting in missed collisions with fast movements or very large deformations [56]. This greatly influences the accuracy of interactions between the surgical instrument and deformational model and restricts the interactive capability between surgical instruments with different sizes and the deformational model. To address these problems, we develop two new collision detection algorithms: the DCDVS and HCDVS algorithm. The former utilizes the volumetric structure to extend the effective collision range of the DCD algorithm, causing an improvement of detection accuracy. The latter combines the DCDVS algorithm with the interpolation

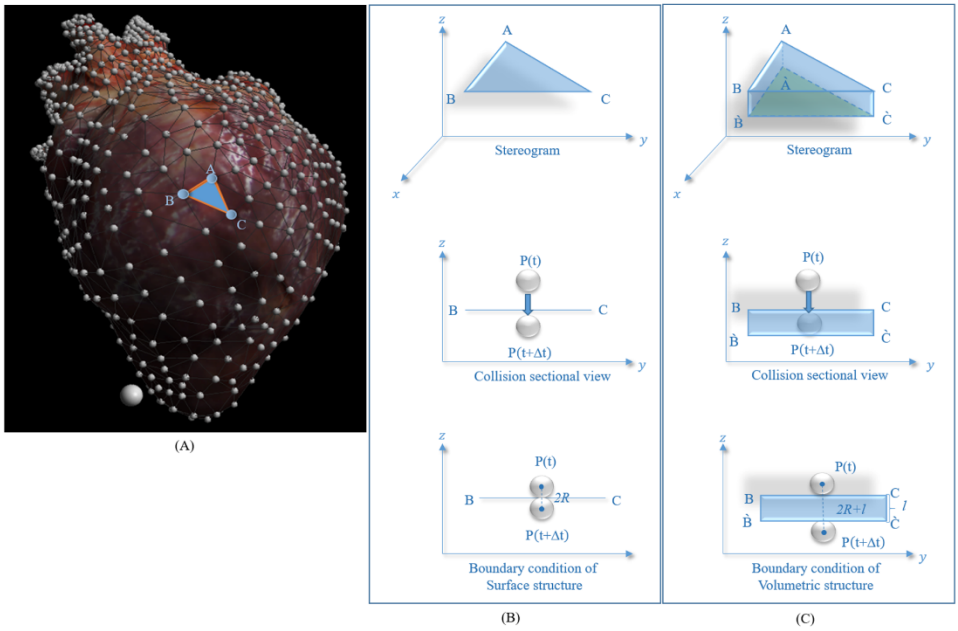

FIGURE 5. Differences between the DCD and DCDVS algorithms. (A) The surface MSM of the virtual heart is composed of k triangles (triangular topological units), such as triangle ABC; a white ball o is used to represent a surgical instrument. (B) $P(t)$ and $P(t+1)$ represent the positions of the surgical instrument o at time t and $t+1$, respectively; line BC represents the sectional view of triangle ABC; $2R$ represents the effective range (boundary condition) of the DCD algorithm in which the surgical instrument can collide with triangle ABC. (C) Plane BC $\grave{B}\grave{C}$ represents the sectional view of the triangular prism ABC $\grave{A}\grave{B}\grave{C}$; $2R+l$ represents the effective range (boundary condition) of the DCDVS algorithm in which the surgical instrument can collide with triangular prism ABC $\grave{A}\grave{B}\grave{C}$.

technique of CCD algorithm [56] to achieve an effective balance between the detection accuracy and real-time interactive performance. The two algorithms are demonstrated as follows.

A. PROPOSED DCDVS ALGORITHM

Before demonstrating the DCDVS algorithm, let us first consider the weakness of the DCD algorithm caused by the discrete computing behavior of a computer.

1) WEAKNESS OF THE DCD ALGORITHM

As shown in Fig. 5, the surface MSM of the virtual heart is composed of k triangles (triangular topological units), and each of the k triangles such as triangle ABC is described as the triangle $tr_i, i = 1, \ldots, k$. For the sake of simplicity, a white ball o represents a surgical instrument, and its center represents the position of the surgical instrument.

Furthermore, Fig. 5B shows that the surgical instrument o is located at the position $P(t)$ at time t before it interacts with triangle ABC. The surgical instrument o (the white ball) moves towards triangle ABC and gradually closer to it. In the third subgraph of Fig. 5B from up to down, when the surgical instrument (the white ball) is above the triangle ABC and just touches it, the distance between the center of the white ball and the triangle ABC is equal to R. Afterwards, the white ball continues to interact with the triangle ABC until it is below the triangle ABC. When the surgical instrument (the white ball) is below the triangle ABC and just touches it, the distance between them is also equal to R. As the white ball continues to move down, the triangle ABC is unable to touch the triangle ABC. From the above analysis, $2R$ indicates an effective collision or interactive range between the surgical instrument (the white ball) and the triangle ABC. Specially, once the surgical instrument o (the white ball) moves towards triangle ABC sufficiently quickly, the surgical instrument o passed through triangle ABC and was then located at the opposite position $P(t+\Delta t)$ of position $P(t)$, without intersecting or contacting triangle ABC at time $t+\Delta t$. The DCD algorithm can execute effective collision detection only at each interval Δt due to the discrete computing behavior of a computer. Consequently, the DCD algorithm is unable to detect collisions between the surgical instrument o and triangle ABC at time $t+\Delta t$ even if there was a real collision between time t and $t+\Delta t$.

In this case, to effectively detect the collision between the surgical instrument o and triangle ABC with the DCD algorithm, the average moving speed of the surgical instrument o should be less than or equal to the maximum speed value V_{max}, which is formulated as follows:

$$V_{mean} \leq V_{max} \tag{12}$$

where V_{mean} is defined as the average moving speed of the surgical instrument o; the maximum speed value V_{max} can be computed as follows:

$$V_{max} = \frac{2R}{\Delta t} \tag{13}$$

where R is the radius of the surgical instrument o and $2R$ represents the effective collision range (boundary condition) of the DCD algorithm in which the surgical instrument can collide with triangle ABC, as shown in Fig. 5B. Formula (13) clearly shows that the maximum speed V_{max} depends on the radius R of the surgical instrument and time interval Δt. In general, the time interval Δt is fixed, depending on the computer. However, the radius R of the surgical instrument is variable depending on the design requirement. Once its radius R becomes sufficiently short, the corresponding V_{max} of the surgical instrument o (the white ball o) will become very slow according to formula (13), which greatly limits the moving speed of the surgical instrument. Therefore, the surgical instrument must move slowly. Otherwise, the interaction between the MSM and the surgical instrument may not be successfully detected by the DCD algorithm at the time interval Δt. This may result in some incorrect interactive results in the surgical simulation. Hence, the size of the radius R need to be extended to ensure the validity and reliability of the DCD algorithm in the surgical simulation.

However, this extension to the radius R can greatly affect the performance of virtual surgical simulation. For instance, it is generally necessary for virtual surgery to simulate some operating behaviors between surgical instruments with different radii and soft tissue. Therefore, enlarging the radius R to improve the DCD algorithm is not very rational in some simulation scenes.

2) PROPOSED DCDVS ALGORITHM

To address the abovementioned problem, we develop the discrete collision detection with volume structure (DCDVS) algorithm, which adopts the triangular prism volume structure, instead of the triangular surface structure. The triangular prism volume structure extends the effective range of collision detection, which can improve the validity and reliability of collision detection between instruments with different radii and virtual soft tissue.

As illustrated in Fig. 5C and Fig. 6, the triangular prism $ABC\,\grave{A}\grave{B}\grave{C}$, which is the volumetric structure for improving collision detection, is constructed by translating triangle ABC with length l along the opposite direction of the normal vector $\vec{n}$ of triangle ABC. Specially, a normal vector is an object such as a line or vector that is perpendicular to a given object. For a triangle, a surface normal vector can be calculated as the vector cross product of two (non-parallel) edges of the polygon. Clearly, the triangle $\grave{A}\grave{B}\grave{C}$ is completely parallel and equal to triangle ABC. Since the surgical instrument o interacts with triangular prism ABC $\grave{A}\grave{B}\grave{C}$ instead of triangle ABC, the effective range of collision detection is extended from $2R$ (shown in Fig. 5B) to $2R + l$ (shown in Fig. 5C). In this case, the maximum speed V_{max} that the proposed DCDVS algorithm can detect at the time interval Δt is also effectively increased as follows:

$$V_{max} = \frac{2R + l}{\Delta t} \tag{14}$$

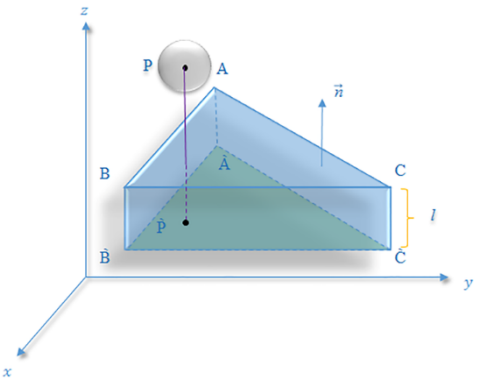

FIGURE 6. Schematic diagram of volume structure collision detection. Vector $\vec{n}$ is the normal vector of triangle ABC; triangular prism ABC $\grave{A}\grave{B}\grave{C}$ is the volumetric structure for improving collision detection; l is the volumetric structure distance of triangular prism ABC $\grave{A}\grave{B}\grave{C}$ from triangle ABC to triangle $\grave{A}\grave{B}\grave{C}$; $\grave{P}$ is the projection positon of position P on triangle $\grave{A}\grave{B}\grave{C}$.

where l is the volumetric structure distance from triangle ABC to triangle $\grave{A}\grave{B}\grave{C}$. Compared with that in formula (13), the maximum speed in formula (14) relies on both the radius R and distance l. If the distance l is assigned to a reasonable range, the radius R of the surgical instrument o can be reduced to zero. This result indicates that the size of surgical instruments need not be extended to a certain range to ensure the validity and reliability of collision detection. Therefore, the DCDVS algorithm can adapt to surgical instruments with different sizes, which can be reasonably regulated to match corresponding demands in surgical simulation.

The implementation process of the DCDVS algorithm can be demonstrated with two steps:

Step (1) The first step is to determine whether a collision has occurred between the virtual surgical instrument and the virtual soft tissue. Suppose that $P = (x_0, y_0, z_0)$ is the current position of the virtual surgical instrument; then, the plane equation of triangle ABC can be defined as

$$\alpha x + \beta y + \tau z + \rho = 0 \tag{15}$$

where α, β, τ and ρ are the plane equation coefficients. Similarly, the plane equations of triangle $\grave{A}\grave{B}\grave{C}$ can be defined as

$$\alpha_1 x + \beta_1 y + \tau_1 z + \rho_1 = 0 \tag{16}$$

where α_1, β_1, τ_1 and ρ_1 are the plane equation coefficients.

Furthermore, the distance L_1 from position $P = (x_0, y_0, z_0)$ to triangle ABC can be computed as

$$L_1 = \frac{|\alpha x_0 + \beta y_0 + \tau z_0 + \rho|}{\sqrt{\alpha^2 + \beta^2 + \tau^2}}. \tag{17}$$

The distance L_2 from $P = (x_0, y_0, z_0)$ to triangle $\grave{A}\grave{B}\grave{C}$ can be computed as

$$L_2 = \frac{|\alpha_1 x_0 + \beta_1 y_0 + \tau_1 z_0 + \rho_1|}{\sqrt{\alpha_1^2 + \beta_1^2 + \tau_1^2}}. \tag{18}$$

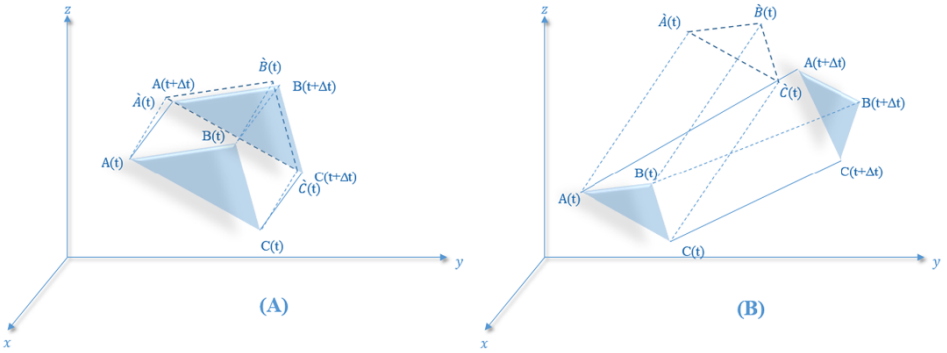

FIGURE 7. Limitation of DCDVS algorithm in the large enough deformation. (A) When the MSM generating a small deformation, the moving trajectory from A(t)B(t)C(t) to A($t+\Delta t$)B($t+\Delta t$)C($t+\Delta t$) is similar to that from the triangle A(t)B(t)C(t) to $\grave{A}\,(t+\Delta t)\,\grave{B}\,(t+\Delta t)\,\grave{C}\,(t+\Delta t)$. (B) When the MSM generating a large enough deformation, the moving trajectory from A(t)B(t)C(t) to A($t+\Delta t$)B($t+\Delta t$)C($t+\Delta t$) is great different from the trajectory from the triangle A(t)B(t)C(t) to $\grave{A}\,(t+\Delta t)\,\grave{B}\,(t+\Delta t)\,\grave{C}\,(t+\Delta t)$. In this case, it is unavailable for the DCDVS to extend the distance l of the triangular prism ABC$\grave{A}\grave{B}\grave{C}$ for improving the detection accuracy.

If $L_1 < 2R + l$ and $L_2 < 2R + l$ are true, the second step will continue. Otherwise, collision does not occur between the surgical instrument and triangle ABC.

Step (2) Suppose that $\grave{P}$ is the projection position of position $P = (x_0, y_0, z_0)$ on triangle $\grave{A}\grave{B}\grave{C}$, as shown in Fig. 6. If position $\grave{P}$ is inside triangle $\grave{A}\grave{B}\grave{C}$, then collision can be determined to occur between the surgical instrument and triangle ABC. Otherwise, collision does not occur.

B. PROPOSED HCDVS ALGORITHM

1) LIMITATION OF THE PROPOSED DCDVS ALGORITHM

As shown in Fig. 7A, the proposed MSM provides a deformation within a certain range so that the triangle $A(t)B(t)C(t)$ at the time t can be approximately parallel to the triangle $A(t+\Delta t)B(t+\Delta t)C(t+\Delta t)$ at the time $t+\Delta t$. In other words, the distance between $A(t)$ and $A(t+\Delta t)$ is approximately equal to either the distance between $B(t)$ and $B(t+\Delta t)$, or the distance between $C(t)$ and $C(t+\Delta t)$. In this case, we can observe from the Fig. 7A that the moving trajectory from $A(t)B(t)C(t)$ to $A(t+\Delta t)B(t+\Delta t)C(t+\Delta t)$ is similar to that from the triangle $A(t)B(t)C(t)$ to $\grave{A}\,(t+\Delta t)\,\grave{B}\,(t+\Delta t)\,\grave{C}\,(t+\Delta t)$ (the distance l *of* the triangular prism $ABC\,\grave{A}\grave{B}\grave{C}$ *in DCDVS*). Therefore, it is reasonable for the DCDVS algorithm to use the triangular prism $ABC\,\grave{A}\grave{B}\grave{C}$ with the distance l to improve the detection accuracy.

However, as shown in Fig. 7B, when the proposed MSM generates a large enough deformation, the triangle ABC at the time t may not be parallel to the triangle ABC at the time $t+\Delta t$. In other words, the distance between $A(t)$ and $A(t+\Delta t)$ may be neither equal to the distance between $B(t)$ and $B(t+\Delta t)$ nor the distance between $C(t)$ and $C(t+\Delta t)$. Fig. 7B shows that the moving trajectory from $A(t)B(t)C(t)$ to $A(t+\Delta t)B(t+\Delta t)C(t+\Delta t)$ is great different from the trajectory from the triangle $A(t)B(t)C(t)$ to $\grave{A}\,(t+\Delta t)\,\grave{B}\,(t+\Delta t)\,\grave{C}\,(t+\Delta t)$. In this case, it is not appropriate for the DCDVS to extend the distance l of the triangular prism $ABC\,\grave{A}\grave{B}\grave{C}$ for increasing the V_{max} in formula (14) and improving the detection accuracy.

2) PROPOSED HCDVS ALGORITHM

The CCD algorithms generally adopt linear interpolation techniques between discrete vertex positions of the volume meshes and detect collisions among swept volumes by using elementary detection [56]. These CCD algorithms offer accurate and robust collision detection effects by calculating the first time of interaction, prohibiting inter-penetrations and detecting collisions even between fast moving objects. Inspired by the CCD algorithms, we propose a hybrid collision detection algorithm with the volume structure (HCDVS) by introducing the interpolation techniques of the CCD algorithm into DCDVS to overcome this drawback of the DCDVS algorithm. For the sake of simplicity, the HCDVS algorithm is demonstrated based on the aforesaid DCDVS algorithm. Therefore, the proposed MSM still uses the virtual heart shown in Fig.5. Without loss of generality, the triangle ABC is also still used to describe the fundamental principle of the HCDVS algorithm. Since time interval Δt is extremely short, the moving speed of the triangle ABC can be approximated as constant.

As shown in Fig. 8A, the points A, B, and C on the triangle ABC are described as $A(t)$, $B(t)$, and $C(t)$ at the time t, respectively. Similarly, these points are described as $A(t+\Delta t)$, $B(t+\Delta t)$, and $C(t+\Delta t)$ at the time $t+\Delta t$, respectively. The positions of the vertex points $A(t)$, $B(t)$, and $C(t)$ can be described as $p_A(t)$, $p_B(t)$, and $p_C(t)$ at the time t, respectively. If the surgical instrument does not contact the triangle ABC from the time t to $t+\Delta t$, the positions of the vertex points $A(t+\Delta t)$, $B(t+\Delta t)$, and $C(t+\Delta t)$ will be updated as $p_A(t+\Delta t)$, $p_B(t+\Delta t)$, and $p_C(t+\Delta t)$ at the time $t+\Delta t$ by the formula (10), respectively. To determine whether a collision occurs between the surgical instrument and the triangle ABC, a simple way is that the triangle ABC is used to conduct interpolations between the triangle $A(t)B(t)C(t)$ and

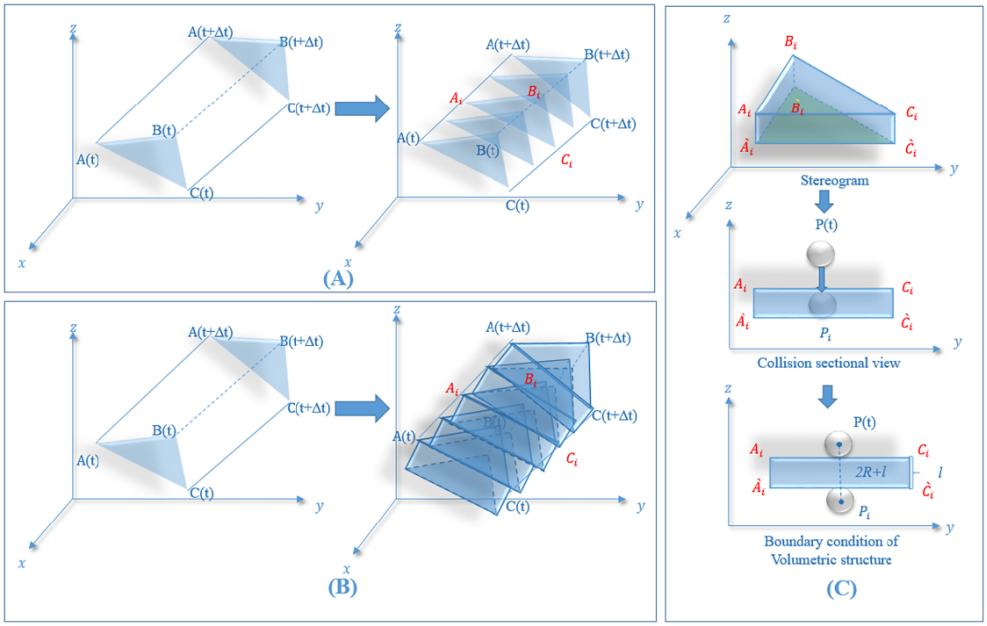

FIGURE 8. Schematic diagram of HCDVS algorithm. (A) points A, B, and C on the triangle ABC are described as A(t), B(t), and C(t) at the time t, respectively; points A, B, and C on the triangle ABC are described as A($t+\Delta t$), B($t+\Delta t$), and C($t+\Delta t$) at the time $t+\Delta t$, respectively; the triangle ABC has not thickness so that some gaps between the triangle A(t)B(t)C(t) and A($t+\Delta t$)B($t+\Delta t$)C($t+\Delta t$) may never be interpolated. (B) The triangular prism from the DCDVS algorithm is employed to replace the triangle ABC to perform the interpolation between the triangle A(t)B(t)C(t) and A($t+\Delta t$)B($t+\Delta t$)C($t+\Delta t$); the triangular prism $A_iB_iC_i\grave{A}_i\grave{B}_i\grave{C}_i$ is used for the ith interpolation. (C) In the ith interpolation, the surgical instrument o interacts with the triangular prism $A_iB_iC_i\grave{A}_i\grave{B}_i\grave{C}_i$; plane $A_iC_i\grave{A}_i\grave{C}_i$ represents the sectional view of the triangular prism $A_iB_iC_i\grave{A}_i\grave{B}_i\grave{C}_i$; l is the volumetric structure distance of triangular prism$A_iB_iC_i\grave{A}_i\grave{B}_i\grave{C}_i$ from the triangle $A_iB_iC_i$ to $\grave{A}_i\grave{B}_i\grave{C}_i$; $2R+l$ represents the effective range in which the surgical instrument can collide with triangular prism $A_iB_iC_i\grave{A}_i\grave{B}_i\grave{C}_i$ in the ith interpolation.

$A(t+\Delta t)B(t+\Delta t)C(t+\Delta t)$. Unfortunately, the triangle ABC has not thickness so that some gaps between the triangle $A(t)B(t)C(t)$ and $A(t+\Delta t)B(t+\Delta t)C(t+\Delta t)$ may never be interpolated in Fig. 8A. This may lead to some missed detections between these gaps. From the reason mentioned above, the triangular prism from the DCDVS algorithm is employed to replace the triangle ABC to perform the interpolation between the triangle $A(t)B(t)C(t)$ and $A(t+\Delta t)$ $B(t+\Delta t)C(t+\Delta t)$.

As illustrated in Fig. 8B, assume that the total interpolation number between the triangle $A(t)B(t)C(t)$ and $A(t+\Delta t)B(t+\Delta t)C(t+\Delta t)$ is equal to Υ. For the ith interpolation, $i=0,1,\ldots,\Upsilon$, the triangular prism is expressed as $A_iB_iC_i\grave{A}_i\grave{B}_i\grave{C}_i$. Similar to the DCDVS algorithm, the triangular prism $A_iB_iC_i\grave{A}_i\grave{B}_i\grave{C}_i$ is also constructed by translating the triangle $A_iB_iC_i$ with the length l along the opposite direction of the normal vector of triangle $A_iB_iC_i$.

In addition, the total number of the interpolation is defined as

$$\Upsilon = \left\lceil \frac{L_{max}}{l} \right\rceil. \tag{19}$$

Here, Υ is the total number of the interpolation; l is the volume structure distance of the triangular prism $A_iB_iC_i\grave{A}_i\grave{B}_i\grave{C}_i$ from triangle $A_iB_iC_i$ to triangle $\grave{A}_i\grave{B}_i\grave{C}_i$ shown in Fig. 8B and Fig. 8C; "$\lceil\ \rceil$" can round values to the nearest integer toward positive infinity; L_{max} is defined as

$$L_{max} = \max\{L_A, L_B, L_C\} \tag{20}$$

where $L_A = |p_A(t+\Delta t) - p_A(t)|$, $L_B = |p_B(t+\Delta t) - p_B(t)|$, and $L_C = |p_C(t+\Delta t) - p_C(t)|$ represent the distances between $A(t)$ and $A(t+\Delta t)$, $B(t)$ and $B(t+\Delta t)$, and $C(t)$ and $C(t+\Delta t)$, respectively; L_{max} is the maximum value of distances $|p_A(t+\Delta t) - p_A(t)|$, $|p_B(t+\Delta t) - p_B(t)|$, and $|p_C(t+\Delta t) - p_C(t)|$.

According to the interpolation number Υ, the time interval Δt between the time t and $t+\Delta t$ is divided into Υ interpolation time subintervals. For the ith interpolation, $i=0,1,\ldots,\Upsilon$, the corresponding time subinterval is described as t_i. Specially, t_0 and t_Υ are equal to t and $t+\Delta t$, respectively. Then, we can achieve the positions of the vertex point A_i, B_i, C_i, $\grave{A}_i$, $\grave{B}_i$, and $\grave{C}_i$ on each triangular prism $A_iB_iC_i\grave{A}_i\grave{B}_i\grave{C}_i$ at the time subinterval t_i, $i=0,1,\ldots,\Upsilon$ as follows:

$$p_A(t_i) = p_A(t) + \frac{i}{\Upsilon}\cdot[p_A(t+\Delta t) - p_A(t)] \tag{21}$$

$$p_B(t_i) = p_B(t) + \frac{i}{\Upsilon}\cdot[p_B(t+\Delta t) - p_B(t)] \tag{22}$$

$$p_C(t_i) = p_C(t) + \frac{i}{\Upsilon}\cdot[p_C(t+\Delta t) - p_C(t)] \tag{23}$$

$$p_{\grave{A}}(t_i) = p_A(t_i) - \vec{n}_i \cdot l \tag{24}$$

$$p_{\grave{B}}(t_i) = p_B(t_i) - \vec{n}_i \cdot l \tag{25}$$

and

$$p_{\grave{C}}(t_i) = p_C(t_i) - \vec{n}_i \cdot l \tag{26}$$

respectively. Here, $p_A(t_i)$, $p_{Bi}(t_i)$, $p_C(t_i)$, $p_{\grave{A}}(t_i)$, $p_{\grave{B}}(t_i)$, and $p_{\grave{C}}(t_i)$ represent the positions of the vertex points A_i, B_i, C_i, $\grave{A}_i$, $\grave{B}_i$, and $\grave{C}_i$ at the time subinterval t_i, respectively; $\vec{n}_i$ is the unit normal vector of the triangle $A_iB_iC_i$ as follows:

$$\vec{n}_i = \frac{[p_{Ai}(t_i) - p_{Bi}(t_i)] \otimes [p_{Ci}(t_i) - p_{Bi}(t_i)]}{|[p_{Ai}(t_i) - p_{Bi}(t_i)] \otimes [p_{Ci}(t_i) - p_{Bi}(t_i)]|} \tag{27}$$

where $\otimes$ represents the cross product between vectors $p_{Ai}(t_i) - p_{Bi}(t_i)$ and $p_{Ci}(t_i) - p_{Bi}(t_i)$; $|[p_{Ai}(t_i) - p_{Bi}(t_i)] \otimes [p_{Ci}(t_i) - p_{Bi}(t_i)]|$ means the absolute value or modulus of $[p_{Ai}(t_i) - p_{Bi}(t_i)] \otimes [p_{Ci}(t_i) - p_{Bi}(t_i)]$.

Suppose that the virtual surgical instrument is located at the position $P(t)$ at time t and $P(t + \Delta t)$ at time $t + \Delta t$, resectively. Then, for the ith interpolation, $i = 0, 1, \ldots, \Upsilon$, the position $P_i(t_i)$ of the virtual surgical instrument is computed as

$$P_i(t_i) = P(t) + \frac{i}{\Upsilon}[P(t + \Delta t) - P(t)] \tag{28}$$

As shown in Fig. 8C, for the ith interpolation, $i = 0, 1, \ldots, \Upsilon$, the surgical instrument o interacts with the triangular prism $A_iB_iC_i\grave{A}_i\grave{B}_i\grave{C}_i$; plane $A_iC_i\grave{A}_i\grave{C}_i$ represents the sectional view of the triangular prism $A_iB_iC_i\grave{A}_i\grave{B}_i\grave{C}_i$; l is the volumetric structure distance of triangular prism $A_iB_iC_i\grave{A}_i\grave{B}_i\grave{C}_i$ from triangle $A_iB_iC_i$ to $\grave{A}_i\grave{B}_i\grave{C}_i$; $2R + l$ represents the effective range in which the surgical instrument can collide with triangular prism $A_iB_iC_i\grave{A}_i\grave{B}_i\grave{C}_i$ in the ith interpolation. Similar to the effective rang of collision detection in DCDVS, the range in HCDVS can also be computed as $2R+l$ in the ith interpolation. Therefore, the maximum speed V_{max} of the surgical instrument o in the ith interpolation is computed according to the formula (14) in the DCDVS algorithm.

Furthermore, if the HCDVS algorithm conducts the interpolation number of Υ at the time interval Δt, the effective rang of a collision detection will be equal to $2R + \Upsilon l$. In this case, the proposed HCDVS algorithm can detect the maximum speed V_{max} that the surgical instrument provides at the time interval Δt as

$$V_{max} = \frac{2R + \Upsilon l}{\Delta t}. \tag{29}$$

In other words, when the surgical instrument interacts with our MSM at the speed V_{max}, the HCDVS algorithm can detect accurately the collision between the surgical instrument and MSM. Therefore, by introducing the interpolation technique from the CCD algorithm, the HCDVS algorithm achieves the higher speed of the collision detection between our MSM and the surgical instrument compared with the DCDVS algorithm. In particular, the maximum average moving speed of the triangle ABC at the time interval Δt is approximately equal to $V_{\max\_mean} = L_{max}/\Delta t$. According to formulas (19) and (29), $V_{\max\_mean}$ is less than V_{max}. This indicates that the HCDVS algorithm does not miss the collision detection between the proposed MSM and the surgical instrument.

Without loss of generality, we use the triangle ABC as any triangle of our MSM. For the triangle ABC, the detection process of the HCDVS algorithm is summarized as follows:

Step (1) The number of interpolations can be achieved according to formula (19). Suppose that the maximum interpolation number is equal to Υ.

Step (2) For the ith interpolation, $i = 0, 1, \ldots, \Upsilon$, the triangular prism $A_iB_iC_i\grave{A}_i\grave{B}_i\grave{C}_i$ is constructed based on the formulas (21)-(27). Suppose that $P_i = (x_i, y_i, z_i)$ is the current position of the virtual surgical instrument that computed according to the formula (28). Distances L_{i1} and L_{i2} are used to describe the distances from the position $p_i = (x_i, y_i, z_i)$ to the triangle $A_iB_iC_i$ and $\grave{A}_i\grave{B}_i\grave{C}_i$, respectively. Similar to L_1 and L_2 of formulas (18) and (19), L_{i1} and L_{i2} are computed as

$$L_{i1} = \frac{|\alpha_i x_i + \beta_i y_i + \tau_i z_i + \rho_i|}{\sqrt{\alpha_i^2 + \beta_i^2 + \tau_i^2}} \tag{30}$$

and

$$L_{i2} = \frac{\left|\grave{\alpha}_i x_i + \grave{\beta}_i y_i + \grave{\tau}_i z_i + \grave{\rho}_i\right|}{\sqrt{\grave{\alpha}_i^2 + \grave{\beta}_i^2 + \grave{\tau}_i^2}} \tag{31}$$

where α_i, β_i, τ_i, and ρ_i are the coefficients of the plane equation of the triangle $A_iB_iC_i$; $\grave{\alpha}_i$, $\grave{\beta}_i$, $\grave{\tau}_i$, and $\grave{\rho}_i$ are the coefficients of the plane of the triangle $\grave{A}_i\grave{B}_i\grave{C}_i$. Note that the plane equation of the triangle $A_iB_iC_i$ and $\grave{A}_i\grave{B}_i\grave{C}_i$, can be defined as

$$\alpha_i x + \beta_i y + \tau_i z + \rho_i = 0 \tag{32}$$

and

$$\grave{\alpha}_i x + \grave{\beta}_i y + \grave{\tau}_i z + \grave{\rho}_i = 0 (33) \tag{33}$$

respectively.

Particularly, If $L_{i1} \leq 2R + l$ and $L_{i2} \leq 2R + l$ are true, the fourth step will continue, indicating that the surgical instrument intersects or is inside the triangular prism $A_iB_iC_i\grave{A}_i\grave{B}_i\grave{C}_i$ in the ith interpolation; otherwise, if $i \leq \Upsilon$ is true, the second step will continue, indicating that the surgical instrument never interacts the triangular prism $A_iB_iC_i\grave{A}_i\grave{B}_i\grave{C}_i$ in the ith interpolation.

Step (3) The interaction between the surgical instrument and triangle ABC never occurs for all the Υ interpolations. The fifth step will continue.

Step (4) The HCDVS algorithm has successfully detected the collision between the surgical instrument and triangle ABC.

Step (5) The HCDVS algorithm is terminated.

IV. HYBRID POSITION CONSTRAINT RESPONSE METHOD

After performing the abovementioned collision detection, a hybrid position constraint response (HPCR) method is developed to match the DCDVS or HCDVS algorithm and carry out the corresponding deformation response computation.

A. HPCR METHOD FOR DCDVS

The principle of this method is that while the surgical instrument interacts with the abovementioned triangle tr_i, $i = 1, \ldots, k$, each of the triangles can follow this surgical instrument to move together. This method can improve the deformational effect of the soft tissue in surgical simulation.

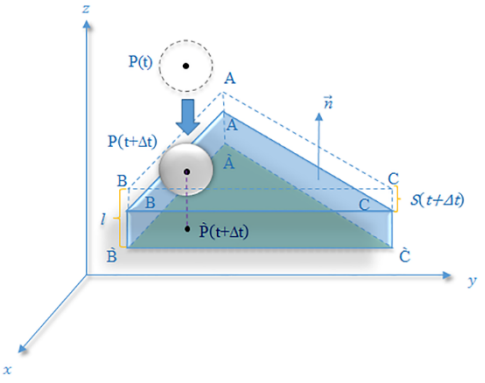

FIGURE 9. Schematic diagram of the HPCR method for the DCDVS algorithm. Position $P(t)$ is the position of the surgical instrument at time t; $P(t+\Delta t)$ is the position of the virtual surgical instrument at time $t+\Delta t$; $\vec{n}$ is the normal vector of triangle ABC; $\grave{P}$ is the projection of position P on triangle $\grave{A}\grave{B}\grave{C}$; l is the volumetric structure distance of triangular prism ABC $\grave{A}\grave{B}\grave{C}$ from triangle ABC to triangle $\grave{A}\grave{B}\grave{C}$; $S(t+\Delta t)$ is the movement distance of triangle ABC following the surgical instrument (white ball) from time t to $t+\Delta t$.

As shown in Fig. 9, the surgical instrument is located at position $P(t)$ at time t and is not in contact with triangle ABC. The normal vector of triangle ABC is $\vec{n}$. When the surgical instrument interacts with triangle ABC at the next time step $t + \Delta t$, triangle ABC can be tangent to the surgical instrument, enabling triangle ABC to follow the movement of the surgical instrument. Suppose that $P(t + \Delta t) = (x_0, y_0, z_0)$ is the position of the virtual surgical instrument at time $t + \Delta t$. Then, position $P(t + \Delta t)$ is projected to triangle $\grave{A}\grave{B}\grave{C}$ as position $\grave{P}(t + \Delta t)$. According to formula (18), the distance from position $P(t + \Delta t)$ to $\grave{P}(t + \Delta t)$ can be achieved as

$$Dis(t + \Delta t) = L_2 = \frac{|\alpha_1 x_0 + \beta_1 y_0 + \tau_1 z_0 + \rho_1|}{\sqrt{\alpha_1^2 + \beta_1^2 + \tau_1^2}} \quad (34)$$

where $Dis(t + \Delta t)$ is the distance from $P(t + \Delta t)$ to $\grave{P}(t + \Delta t)$. In this case, the movement distance $S(t + \Delta t)$ of triangle ABC following the surgical instrument (white ball) from time t to $t + \Delta t$ is described as

$$S(t + \Delta t) = R + l - Dis(t + \Delta t) \quad (35)$$

where R is the radius of the surgical instrument (white ball) and l represents the volumetric structure distance from triangle ABC to triangle $\grave{A}\grave{B}\grave{C}$. The movement direction of triangle ABC is opposite to its normal vector as follows:

$$d(t + \Delta t) = -\vec{n}. \quad (36)$$

In addition, suppose that any mass point x_i from our MSM is the intersection point of m triangles. After m triangles have collided with the surgical instrument at time t, their moving distance and direction at the next time step $t + \Delta t$ can be expressed as $S_1(t + \Delta t), S_2(t + \Delta t) \cdots, S_m(t + \Delta t)$ and $d_1(t + \Delta t), d_2(t + \Delta t), \cdots, d_m(t + \Delta t)$, respectively. Hence, at time $t + \Delta t$, the position $p_i(t + \Delta t)$ of the mass point x_i can be achieved as follows:

$$P_i(t + \Delta t) = P_i(t) + \frac{\sum_{i=1}^{m} S_i(t + \Delta t)}{n} \cdot \frac{\sum_{i=1}^{m} d_i(t + \Delta t)}{\left|\sum_{i=1}^{m} d_i(t + \Delta t)\right|} \quad (37)$$

where $\sum_{i=1}^{m} S_i(t + \Delta t)/n$ is the average moving distance of m triangles and $\sum_{i=1}^{m} d_i(t + \Delta t)/\left|\sum_{i=1}^{m} d_i(t + \Delta t)\right|$ is the average direction of m triangles In this way, as each mass point from our is computed using our HPCR method, the deformational response of our MSM can be simulated after collision detection occurs.

B. HPCR METHOD FOR HCDVS

A hybrid position constraint response (HPCR) method is proposed to match the HCDVS algorithm and conduct the corresponding deformation response computation as follows.

As shown in Fig. 10A, the surgical instrument is located at position $P(t)$ at time t and is not in contact with triangle ABC. The normal vector of triangle ABC is $\vec{n}$. The points A, B, and C on the triangle ABC are described as $A(t)$, $B(t)$, and $C(t)$ at the time t, respectively. When the surgical instrument does not collide the triangle ABC, the points A, B, and C can be described as $A(t + \Delta t)$, $B(t + \Delta t)$, and $C(t + \Delta t)$ at the time $t + \Delta t$. In this case, the positions of these points can be computed by formula (10). To detect an interaction between the surgical instrument and the triangle ABC, the HCDVS algorithm uses the triangular prism $ABC\grave{A}\grave{B}\grave{C}$ with the distance l to perform the interpolations of the volume structure between triangles $A(t)B(t)C(t)$ and $A(t + \Delta t)B(t + \Delta t)C(t + \Delta t)$. According to formula (19), the interpolation number can be computed as Υ. For each interpolation $i = 0, 1, \ldots, \Upsilon$, the triangular prism $ABC\grave{A}\grave{B}\grave{C}$ corresponds to the triangular prism $A(t_i)B(t_i)C(t_i)\grave{A}(t_i)\grave{B}(t_i)\grave{C}(t_i)$.

As shown in Fig. 10B, we assume that the HCDVS algorithm detects an interaction between the surgical instrument and the triangular prism $A(t_c)B(t_c)C(t_c)\grave{A}(t_c)\grave{B}(t_c)\grave{C}(t_c)$ in the cth interpolation. Here, t_c belongs to $[t, t + \Delta t]$, representing the first time of interaction between the surgical instrument and the triangular prism $A(t_c)B(t_c)C(t_c)\grave{A}(t_c)\grave{B}(t_c)\grave{C}(t_c)$. The normal vector of triangle $A(t_c)B(t_c)C(t_c)$ is $\vec{n}_c$. The surgical instrument is located at position $P(t_c)$

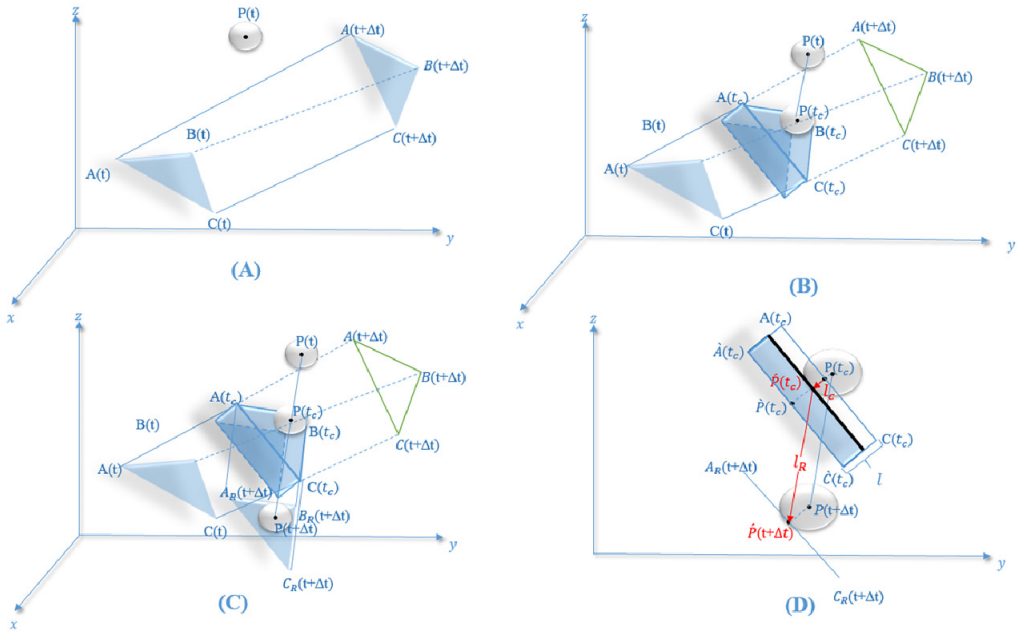

FIGURE 10. Schematic diagram of the HPCR method for the HCDVS algorithm. (A) $P(t)$ is the position of the surgical instrument at time t; A(t), B(t), and C(t) are the vertex points of the triangle ABC at the time t, respectively; A($t+\Delta t$), B($t+\Delta t$), and C($t+\Delta t$) are the vertex points of the triangle ABC at the time $t+\Delta t$. (B) $P(t_c)$ and $A(t_c)\,B(t_c)\,C(t_c)\,\grave{A}(t_c)\,\grave{B}(t_c)\,\grave{C}(t_c)$ are the position of the surgical instrument and the triangular prism in the cth interpolation, respectively. (C) The surgical instrument (white ball) enables the triangle ABC to move from $A(t_c)\,B(t_c)\,C(t_c)$ at time t_c to $A_R(t+\Delta t)\,B_R(t+\Delta t)\,C_R(t+\Delta t)$ at the time $t+\Delta t$. (D) $P(t_c)$ and $P(t+\Delta t)$ is the position of the virtual surgical instrument at time t_c and $t+\Delta t$, repectively; $\dot{P}(t_c)$ is the projection position of the position $P(t_c)$ on $\grave{A}(t_c)\,\grave{B}(t_c)\,\grave{C}(t_c)$; $\dot{P}(t_c)$ is the projection position of the position $P(t_c)$ on the black line described as a plane that is parallel to $A(t_c)\,B(t_c)\,C(t_c)$ or $\grave{A}(t_c)\,\grave{B}(t_c)\,\grave{C}(t_c)$; $A(t_c)\,C(t_c)\,\grave{A}(t_c)\,\grave{C}(t_c)$ represents the sectional view of the triangular prism $A(t_c)\,B(t_c)\,C(t_c)\,\grave{A}(t_c)\,\grave{B}(t_c)\,\grave{C}(t_c)$; $A(t_c)\,C(t_c)$ and $\grave{A}(t_c)\,\grave{C}(t_c)$ represents the sectional view of the triangle $(t_c)\,B(t_c)\,C(t_c)$ and $\grave{A}(t_c)\,\grave{B}(t_c)\,\grave{C}(t_c)$, respectively; $\dot{P}(t+\Delta t)$ is the projection position of the position $P(t+\Delta t)$ on line A_RC_R that represents the sectional view of the triangle $A_RB_RC_R$; l represents the volumetric structure distance from the triangular $A(t_c)\,B(t_c)\,C(t_c)$ to $\grave{A}(t_c)\,\grave{B}(t_c)\,\grave{C}(t_c)$; l_c and l_R represents the moving distance from the time t_c and $t+\Delta t$.

and in contact with the triangular prism $A(t_c)\,B(t_c)\,C(t_c)\,\grave{A}(t_c)\,\grave{B}(t_c)\,\grave{C}(t_c)$ at time t_c.

Afterwards, Fig. 10C shows that owing to the interaction between the surgical instrument and the triangular prism $A(t_c)\,B(t_c)\,C(t_c)\,\grave{A}(t_c)\,\grave{B}(t_c)\,\grave{C}(t_c)$ at the time t_c, the triangle ABC is described as $A(t_c)\,B(t_c)\,C(t_c)$, and it is tangent to the surgical instrument. This enables the triangle ABC to follow the movement of the surgical instrument instead of following the formula (10) of the proposed MSM. From this reason, the triangle ABC is always tangent to the surgical instrument, and follow its movements from the time t_c to the time $t+\Delta t$. In Fig. 10C, the triangle ABC is described as $A_R(t+\Delta t)\,B_R(t+\Delta t)\,C_R(t+\Delta t)$ at the time $t+\Delta t$. This indicates that the surgical instrument enables the triangle ABC to move from $A(t_c)\,B(t_c)\,C(t_c)$ to $A_R(t+\Delta t)\,B_R(t+\Delta t)\,C_R(t+\Delta t)$ at the time $t+\Delta t$.

Furthermore, Fig. 10D is the projection drawing of Fig. 10C. The plane $A(t_c)\,C(t_c)\,\grave{A}(t_c)\,C(t_c)$ represents the sectional view of the triangular prism $A(t_c)\,B(t_c)\,C(t_c)\,\grave{A}(t_c)\,\grave{B}(t_c)\,\grave{C}(t_c)$ shown in Fig. 10D. The line $A(t_c)\,C(t_c)$ and $\grave{A}(t_c)\,C(t_c)$ represents the sectional view of the triangle $(t_c)\,B(t_c)\,C(t_c)$ and $\grave{A}(t_c)\,\grave{B}(t_c)\,\grave{C}(t_c)$, respectively. Suppose that $P(t_c) = (x_c, y_c, z_c)$ is the position of the virtual surgical instrument at time t_c; $\dot{P}(t_c)$ is the projection position of the position $P(t_c) = (x_c, y_c, z_c)$ on triangle $\grave{A}(t_c)\,\grave{B}(t_c)\,\grave{C}(t_c)$ at time t_c. According to the formula (18) or (33), the distance from position $P(t_c)$ to $\dot{P}(t_c)$ can be achieved as

$$Dis(t_c) = \frac{|\alpha_c x_c + \beta_c y_c + \tau_c z_c + \rho_c|}{\sqrt{\alpha_c^2 + \beta_c^2 + \tau_c^2}} \tag{38}$$

where α_c, β_c, τ_c, and ρ_c denotes the parameters of the plane equation of triangle $A(t_c)\,B(t_c)\,C(t_c)$. Particularly, the position $\dot{P}(t_c)$ is the projection position of the position $P(t_c)$ on the plane that is described as the black line in Fig. 10D and parallel to the triangle $(t_c)\,B(t_c)\,C(t_c)$ or $\grave{A}(t_c)\,\grave{B}(t_c)\,\grave{C}(t_c)$. To ensure the triangle $A(t_c)\,B(t_c)\,C(t_c)$ to be tangent to the surgical instrument, the triangle $A(t_c)\,B(t_c)\,C(t_c)$ is translated to the plane described by the black line shown in Fig. 10D. The translation distance l_c is computed as follows:

$$l_c = R + l - Dis(t_c) \tag{39}$$

where R the radius of the surgical instrument (the white ball o); l represents the volumetric structure distance from the triangular $A(t_c)\,B(t_c)\,C(t_c)$ to $\grave{A}(t_c)\,\grave{B}(t_c)\,\grave{C}(t_c)$. The movement direction of the triangle $A(t_c)\,B(t_c)\,C(t_c)$ is opposite to its normal vector as follows:

$$d_c = \vec{n}_c. \tag{40}$$

Fig. 10C and Fig. 10D show that the surgical instrument moves to the position $P(t+\Delta t)$ at the time $t+\Delta t$; the triangle ABC follows the surgical instrument and moves to $A_RB_RC_R$ at the time $t+\Delta t$. In this case, Fig. 10D shows that the movement distance of the triangle ABC from the black line to A_RC_R is computed as

$$l_R = |P(t+\Delta t) - P(t_c)|. \tag{41}$$

Here, A_RC_R represents the sectional view of the triangle $A_RB_RC_R$. In other words, the distance l_R is equal to $\left|\acute{P}(t+\Delta t) - \acute{P}(t_c)\right|$; $\acute{P}(t+\Delta t)$ denotes the tangent point between the surgical instrument (the white ball o) and the triangle $A_RB_RC_R$. The direction of the triangle $A_RB_RC_R$ is defined as

$$d_R = \frac{P(t+\Delta t) - P(t_c)}{|P(t+\Delta t) - P(t_c)|} \tag{42}$$

Finally, from the time t_c to $t+\Delta t$, the total movement distance of triangle ABC is computed as

$$S(t+\Delta t) = |l_c d_c + l_R d_R| \tag{43}$$

where $l_c \odot d_c$ describes a dot multiplication or inner product between vectors l_c and d_c; $l_R \odot d_R$ represents a dot multiplication or inner product between vectors l_R and d_R. The corresponding direction is described as

$$d(t+\Delta t) = \frac{l_c d_c + l_R d_R}{|l_c d_c + l_R d_R|}. \tag{44}$$

Suppose that any mass point x_i from our MSM is the intersection point of m triangles. After m triangles have collided with the surgical instrument at time t_c, their moving distance and direction at the next time step $t+\Delta t$ can be expressed as $S_1(t+\Delta t), S_2(t+\Delta t)\cdots, S_m(t+\Delta t)$ and $d_1(t+\Delta t), d_2(t+\Delta t), \cdots, d_m(t+\Delta t)$, respectively. Hence, at time $t+\Delta t$, the position $p_{x_i}(t+\Delta t)$ of the mass point x_i can be achieved as follows:

$$p_{x_i}(t+\Delta t) = p_{x_i}(t_c) + \frac{\sum_{i=1}^{m} S_i(t+\Delta t)}{m} \cdot \frac{\sum_{i=1}^{m} d_i(t+\Delta t)}{\left|\sum_{i=1}^{m} d_i(t+\Delta t)\right|} \tag{45}$$

where $p_{x_i}(t_c)$ is computed according to (21), (22) or (23); $\sum_{i=1}^{m} S_i(t+\Delta t)/m$ is the average moving distance of m triangles and $\sum_{i=1}^{m} d_i(t+\Delta t)/\left|\sum_{i=1}^{n} d_i(t+\Delta t)\right|$ is the average direction of m triangles. In this way, as each mass point from the proposed MSM is computed using our HPCR method for the HCDVS algorithm, the deformational response of our MSM can be simulated after collision detection occurs.

V. EXPERIMENTAL COMPARISONS

A. EXPERIMENTAL INSTRUMENT AND PARAMETER SETTING

Specifically, several evaluations are conducted for our surface MSM: shape restorations are compared between our proposed MSM, the canonical surface MSM [22] and the new volume MSM [31]; real-time tests are performed for comparisons between our MSM, the canonical surface MSM [22] and the new volume MSM [31]; some collision detection experiments provide comparisons between the proposed DCDVS, proposed HCDVS, DCD [46], and CCD [56] with volume the structure interpolation based on our MSM. Some general experimental parameters are described as follows. However, since the above different experiments have to utilize the corresponding experimental parameters, other experimental parameters will be described in the corresponding experiment parts.

FIGURE 11. Experimental instruments for evaluating the proposed MSM with DCDVS and HCDVS.

As shown in Fig. 11, the proposed methods are implemented and verified in C++ and OpenGL on PC with an Intel(R) Core (TM) i7-6700 CPU at 3.40 GHz, Intel(R) HD Graphics GPU, 8.00 GB RAM, and WIN10 64-bit operating system. The PHANToM Omni haptic device is used to model the virtual surgical instrument.

Furthermore, the geometric models of the artery vessel model and heart model for visual rendering are all OBJ files, which include position coordinates, normal vectors and texture coordinates from the geometric models. The physical models of the artery vessel and heart model are OFF files, which include only the position coordinates of the mass points from the compared three MSMs. The parameters of our in this paper are set as follows: each mass point $m_i = 0.5$, the elastic coefficient of each flexion spring $k_i^f = 0.1$ N/mm, the elastic coefficient of each structure spring $k_i^s = 0.05$ N/mm, and the damping coefficient $u_i^d = 0.0012$ N/(mm/s). Similarly, the parameters of the canonical MSM [22] and the new volume MSM [31] are set as follows:

each mass point $m_i = 0.5$, the elastic coefficients of each structure spring $k_i^s = 0.05$ N/mm, the elastic coefficients of each angle spring $k_i^a = 0.05$ N/mm, the elastic coefficient of each shear spring $k_i^{sh} = 0.05$ N/mm, and the damping coefficient $u_i^d = 0.0012$ N/(mm/s). All the experiments are displacement-driven simulations with a time interval of $\Delta t = 0.01$.

B. EXPERIMENTAL RESULTS

1) COMPARISON OF SHAPE RESTORATION

The deformational offset rate γ, the average steady error ΔP, the steady offset standard deviation δ, the steady offset rate γ_∞, and the steady state transition time t_e are used to evaluate the shape restoration of these compared MSMs.

First, the deformational offset rate γ is employed to describe the amount of deformation for the virtual artery vessel, defined as

$$\gamma = \frac{|P(t_0) - P_{max}|}{|D|} \tag{46}$$

where $P(t_0)$ is the average original position of maximum deformation of the mass point pulled and pushed by the surgical instrument; t_0 indicates the time that the virtual artery vessel does not generate deformation despite this mass point being touched by the surgical instrument; P_{max} is the average maximum deformation position of the mass point pulled and pushed by the surgical instrument; and D is the inside diameter of the artery vessel.

Second, the average steady offset ΔP is applied to evaluate the accuracy of the shape restoration based on the same deformational offset rate γ, defined as

$$\Delta P = \frac{\sum_{i=1}^{n} |P_i(t_0) - P_i(t_\infty)|}{n} \tag{47}$$

where $P_i(t_0)$ is the average original position of any mass point x_i from the MSM at time t_0; $P_i(t_\infty)$ denotes the average steady position of any mass point x_i after the deformation of virtual soft tissue ends; t_∞ denotes the average time that the deformation of the virtual soft tissue ends; and n is the number of mass points from the virtual soft tissue.

Third, we also use the steady offset standard deviation δ to evaluate the stability of the shape restoration, defined as

$$\delta = \sqrt{\frac{\sum_{i=1}^{n} (|P_i(t_0) - P_i(t_\infty)| - \Delta P)^2}{n-1}} \tag{48}$$

Fourth, the steady offset rate γ_∞ is defined as the ratio of the average steady offset ΔP to the deformational offset rate γ as follows:

$$\gamma_\infty = \frac{\Delta P}{\gamma} \tag{49}$$

Finally, the average steady state transition time t_e, which denotes how much time is spent in restoring the deformational offset rate of the maximum deformation position of the virtual soft tissue from ΔP to $0.1\Delta P$ on average, is used to describe the convergence performance of the shape restoration.

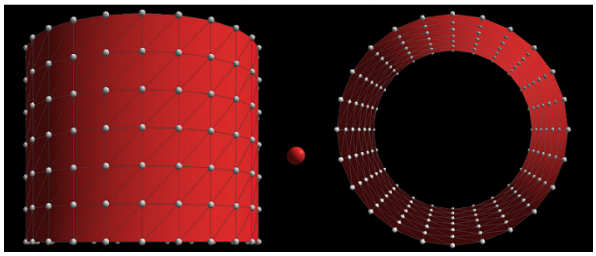

FIGURE 12. Virtual artery vessel model.

As shown in Fig. 12, we first use the canonical surface MSM [22] and the proposed MSM to simulate the deformation of the artery vessel and evaluate the quality of their shape restoration. The mesh model of the artery vessel consists of 378 points and 720 triangles, and the inside diameter of the artery vessel is set to D=35 mm.

The specific experimental process of the shape restoration for two surface MSMs is described as follows: First, we enable the virtual artery vessel to maintain the same deformational offset rate γ by manipulating the haptic device to control the virtual surgical instrument (the red sphere) shown in Fig. 12. Second, we remove the red sphere on the virtual artery vessel at time t_0. Finally, we record the average steady offset ΔP, the steady offset standard deviation δ, the steady offset rate γ_∞, and the steady state transition time t_e for our MSM and the canonical MSM. In addition, we adopt two different operation modes, push mode and pull mode, to operate the virtual artery vessel for testing the shape restoration of the two MSMs at the same deformational offset rate γ. Moreover, two different deformational offset rates, $\gamma = 0.3$D and $\gamma = 0.1$D, are employed to illustrate the volumetric deformation restoring performance in the large deformation and small deformation of the virtual artery vessel, respectively. For our MSM and the canonical surface MSM, the indexes Δp, δ, γ_∞, and t_e are tested, repeated 30 times and averaged at two different offset rates of $\gamma = 0.3$D and $\gamma = 0.1$D with two different operation modes (push mode and pull mode). The corresponding results are shown in Table 1.

Table 1 shows that in the push mode with deformational offset rates $\gamma = 0.1$D and $\gamma = 0.3$D, our MSM has average offset errors of $\Delta P = 0.0287$D and $\Delta P = 0.0089$D, respectively. However, for the canonical surface MSM, the average offset errors are $\Delta P = 0.1104$ D and $\Delta P = 0.0248$D, respectively. Similarly, the corresponding steady offset standard deviations are $\delta = 0.0018$ D and $\delta = 0.0004$ D, respectively, for our MSM, versus $\delta = 0.0234$ D and $\delta = 0.0063$ D for the canonical surface MSM. Furthermore, as for the push mode with the aforesaid two offset rates, the steady offset rates of our MSM are $\gamma_\infty = 9.57\%$ and $\gamma_\infty = 8.90\%$, whereas those of the canonical surface MSM are $\gamma_\infty = 36.8\%$ and $\gamma_\infty = 24.8\%$, respectively. These results show that the average offsets, steady offset standard deviations, and steady offset rates of our MSM are smaller than those of the canonical surface

TABLE 1. Comparisons of shape restoration between our MSM and the canonical MSM. Here, γ is deformational offset rate; ΔP is average steady offset; δ is steady offset standard deviation; γ_∞ is steady offset rate; t_e is steady state transition time.

Model	Operation mode	γ	ΔP	δ	γ_∞	t_e
Our surface MSM	Push	0.3D	0.0287D	0.0018D	9.57%	3.82s
Canonical surface MSM	Push	0.3D	0.1104D	0.0234D	36.80%	10.78s
Our surface MSM	Push	0.1D	0.0089D	0.0004D	8.90%	2.87s
Canonical surface MSM	Push	0.1D	0.0248D	0.0063D	24.8%	6.43s
Our surface MSM	Pull	0.3D	0.0302D	0.0023D	10.07%	3.71s
Canonical surface MSM	Pull	0.3D	0.0982D	0.0202D	32.73%	10.57s
Our surface MSM	Pull	0.1D	0.0092D	0.0007D	9.20%	2.94s
Canonical surface MSM	Pull	0.1D	0.0177D	0.0048D	17.7%	5.98s

MSM in the push mode with different deformational offset rates. This indicates that our proposed MSM provides the better performance of the shape restoration compared with the canonical surface MSM in push mode.

Similarly, the average offset errors of our MSM in the pull mode with the deformational offset rates of $\gamma = 0.3$D and $\gamma = 0.1$D are $\Delta P = 0.0302$D and $\Delta P = 0.0092$D, respectively. However, the average offset errors of the canonical surface MSM are $\Delta P = 0.0982$D and $\Delta P = 0.0177$D, respectively. Subsequently, in the pull mode with the two different offset rates, the corresponding steady offset standard deviations are $\delta = 0.0023$ D and $\delta = 0.0007$ D, respectively, for our MSM, versus $\delta = 0.0202$ D and $\delta = 0.0063$ D for the canonical surface MSM, respectively. Furthermore, the steady offset rates of our MSM are $\gamma_\infty = 10.07\%$ and $\gamma_\infty = 9.20\%$, whereas those of the canonical surface MSM are $\gamma_\infty = 32.73\%$ and $\gamma_\infty = 17.7\%$ in the push mode, respectively. The results indicate that the average offset errors and steady offset rates of our MSM are also smaller than those of the canonical surface MSM in the pull mode with the different deformational offset rates. This also confirms that our proposed MSM has the more performance of the shape restoration than the canonical surface MSM in pull mode.

For the canonical surface MSM, the steady offset rate s of $\gamma = 0.3$D and $\gamma = 0.1$D show large fluctuations. For instance, in the push mode with $\gamma = 0.3$D, the steady offset rate of the canonical surface MSM is equal to $\gamma_\infty = 36.8\%$; however, in the push mode with $\gamma = 0.1$D, its average offset error is $\gamma_\infty = 24.80\%$. For the canonical surface MSM, $\gamma_\infty = 36.8\%$ is 1.48 times more than $\gamma_\infty = 24.80\%$. Conversely, for our MSM, the steady offset rates for $\gamma = 0.3$D and $\gamma = 0.1$D are approximately equal, indicating that the shape restoration of our MSM has more stability than that of the canonical surface MSM in deformational simulation of the virtual artery vessel.

Table 1 also compares the steady state transition time t_e at different deformation conditions for our MSM and the canonical surface MSM. It can be observed that the steady state transitions of the canonical surface MSM is $t_e = 10.78$ s, $t_e = 6.43$ s, $t_e = 10.57$ s, and $t_e = 5.98$ s in different operating modes with different values of γ; however, those of our MSM are $t_e = 3.82$ s, $t_e = 2.87$ s, $t_e = 3.71$ s, and $t_e = 2.94$ s. These results indicate that our MSM has a shorter steady state transition than the canonical surface MSM.

To intuitively compare the convergence performance of the shape restoration for the above two MSMs, the convergence processes of the maximum deformation position of the soft tissue in push mode with $\gamma = 0.3$D from the first two rows of Table 1 are displayed in Fig. 13, where the horizontal axis represents time and the vertical axis is the deformational offset from 0.3D to 0. It can be clearly observed that it takes approximately 4 s for our MSM to reach steady sate with a small average offset steady error; however, the canonical surface MSM takes approximately 11 s with the larger average steady offset. Since the convergence tendencies of the two MSMs in different modes with $\gamma = 0.3$D and $\gamma = 0.1$D in Table 1 are similar to the convergence tendencies in Fig. 13, for brevity, only the convergence tendencies of the two MSMs in push mode with $\gamma = 0.3$D are shown in Fig. 13.

Furthermore, Fig. 14 shows the front views of the virtual artery vessel to visually compare the shape restoration of our MSM and canonical surface MSM in push mode with $\gamma = 0.1D$; Fig. 14A describes the initial deformation state, the deformational process, and the final deformation of the canonical surface MSM from left to right. Similarly, Fig. 14B describes the initial deformation state, the deformational process, and the final deformation of our MSM from left to right. It can be observed that the shape restoration performance of our MSM is superior to that of the canonical surface MSM. To more clearly display the difference in the shape restoration between the canonical surface MSM and our MSM, we convert the front views of the virtual artery vessel in Fig. 14 to the corresponding plane views shown in Fig. 15. This figure further shows that compared with the canonical surface MSM, our MSM has better shape restoration performance in the steady state.

It should be noted that the shape restoration effects of our MSM and the canonical MSM with other various modes and conditions from Table 1 are similar to those shown in Figs. 14 and 15. Hence, for the sake of simplicity, we provide only the visual shape restoration of two MSMs in push mode with $\gamma = 0.1$D for simulating the virtual artery vessel in Figs. 14 and 15.

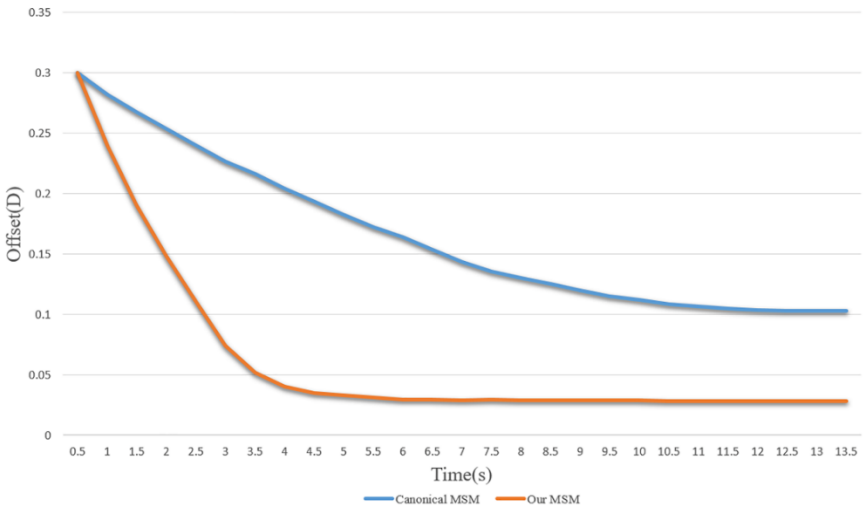

FIGURE 13. Comparison of the convergence performance of the shape restoration for our MSM and canonical surface MSM in push mode with $\gamma = 0.3D$. The horizontal axis represents restoration time, and the vertical axis is the deformational offset from 0.3D to 0.

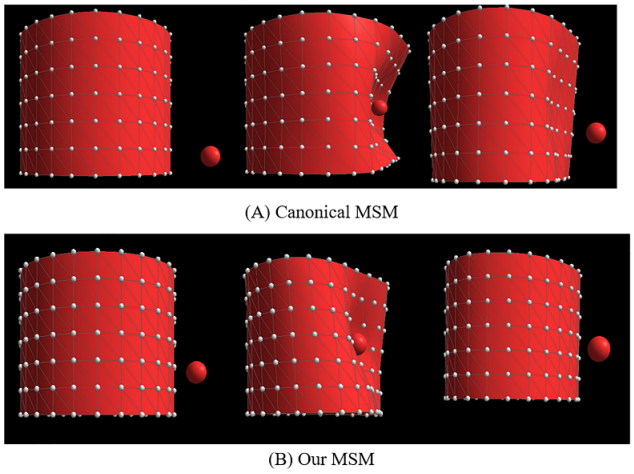

FIGURE 14. Front views of the virtual artery vessel for comparing the shape restorations between the canonical MSM and our MSM. (A) Initial deformation state, deformational process, and final deformation of the virtual artery vessel simulated by the canonical MSM are described from left to right. (B) Initial deformation state, deformational process, and final deformation of the virtual artery vessel simulated by our MSM are described from left to right.

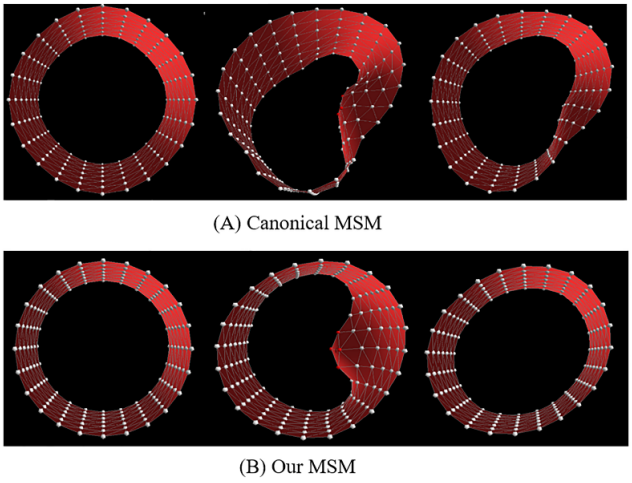

FIGURE 15. Plane views of the virtual artery vessel for comparing the shape restorations between the canonical MSM and our MSM. (A) Initial deformation state, deformational process, and final deformation of the virtual artery vessel simulated by the canonical MSM are described from left to right. (B) Initial deformation state, deformational process, and final deformation of the virtual artery vessel simulated by our MSM are described from left to right.

In order to further evaluate the shape restoration, the proposed surface MSM is compared with a new volume MSM proposed in reference [31] by using a virtual heart with complex topological structure in Fig. 16. It can be observed that the mesh model of the virtual heart simulated by the volume MSM includes 632 points and 724 tetrahedrons; the mesh model of the virtual heart simulated by the proposed MSM contains 252 points and 500 triangles. Similarly, we also first select all points on the virtual heart and then record their average initial positions of all points before the virtual heart generate a deformation; afterwards, record the average maximum deformation positions of all points; finally, record their average final positions of all points after the deformation process of the virtual heart stops. Based on the average maximum deformation positions and the average errors between the initial and final positions, we can evaluate the shape restoration of the proposed MSM and volume MSM that are used to simulate the virtual heart.

Fig. 17A shows the initial, maximum deformation, and final positions of all the points on the virtual heart simulated by the volume MSM from left to right. Similarly, Fig. 17B displays the initial, maximum deformation, and final positions on the virtual heart concerning our proposed MSM from left to right. Specially, in Fig. 17, the blue points mean that the surgical instrument does not touch them. Conversely, the red points denote that the surgical instrument collides

TABLE 2. Comparisons of shape restoration between our MSM and volume MSM. $\Delta P^i_{\max}$ is average maximum position offset; ΔP^i is average final position offset; δ^i is average final position offset standard deviation of ΔP^i; γ^i_∞ is steady offset rate; t^i_e is average steady state transition.

Model	Index	$\Delta P^i_{\max}$	ΔP^i	δ^i	γ^i_∞	t^i_e
Volume MSM	1	5.24	0.81	0.044	15.46%	3.80
Volume MSM	2	5.19	0.75	0.041	14.45%	3.89
Volume MSM	3	5.27	0.87	0.048	16.51%	3.86
Volume MSM	4	1.34	0.21	0.013	15.67%	1.24
Volume MSM	5	1.45	0.27	0.014	18.62%	1.26
Volume MSM	6	1.56	0.32	0.016	20.51%	1.19
Volume MSM	7	1.29	0.19	0.009	14.73%	1.28
Volume MSM	8	1.26	0.15	0.012	11.91%	1.18
Volume MSM	9	1.21	0.17	0.008	14.05%	1.24
Volume MSM	10	1.27	0.16	0.017	12.60%	1.25
Volume MSM	11	1.38	0.26	0.016	18.84%	1.26
Volume MSM	12	1.39	0.23	0.005	16.55%	1.26
Our MSM	1	5.83	0.54	0.032	9.26%	3.62
Our MSM	2	5.78	0.49	0.027	8.48%	3.48
Our MSM	3	5.87	0.53	0.034	9.03%	3.52
Our MSM	4	2.16	0.18	0.009	8.33%	1.25
Our MSM	5	2.48	0.21	0.011	8.47%	1.32
Our MSM	6	2.72	0.23	0.013	8.46%	1.24
Our MSM	7	2.01	0.16	0.008	7.96%	1.27
Our MSM	8	2.03	0.13	0.007	6.40%	1.22
Our MSM	9	1.98	0.14	0.007	7.07%	1.35
Our MSM	10	1.84	0.13	0.006	7.07%	1.28
Our MSM	11	2.54	0.19	0.010	7.48%	1.28
Our MSM	12	2.32	0.18	0.009	7.76%	1.22

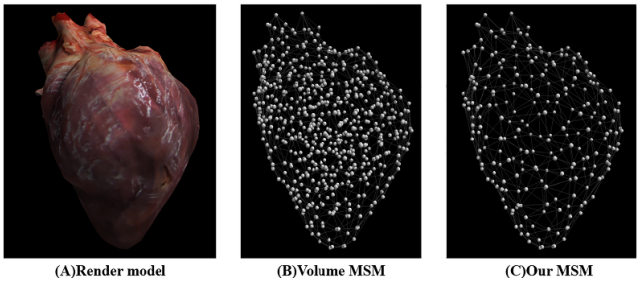

FIGURE 16. Virtual heart model. (A) Virtual heart with complex topological structure. (B) Mesh model of the virtual heart simulated by the volume MSM with 632 points and 724 tetrahedrons. (C) Mesh model of the virtual heart simulated by our proposed MSM with 252 points and 500 triangles.

them. Furthermore, it can be observed from Fig. 17 that both the volume MSM and proposed MSM provide a good shape restoration performance for the virtual heart simulated by them after the deformations of the virtual hearts are ended.

Without loss of generality, as shown in Fig. 17C, we select 12 points from the virtual heart simulated by the volume MSM and the proposed MSM to provide a quantitative evaluation for the shape restoration of the two MSMs. As listed in Table 2, $\Delta P^i_{\max}$ denotes the average maximum position offset between the average initial and the average maximum deformation of each point on the virtual heart position; ΔP^i signifies the average final position offset between the average initial and average final position of each point on the virtual heart; δ^i corresponds to the average final position offset standard deviation of ΔP^i; the steady offset rate γ^i_∞ is defined as the ratio of the average steady offset ΔP to the average maximum position offset $\Delta P^i_{\max}$; t^i_e represents the average steady state transition which denotes how much time each point on the virtual heart spends in restoring its average deformational offset from ΔP^i to $0.1\Delta P^i$ on average, indicating the convergence performance of the shape restoration. Note that the smaller indicators ΔP^i, δ^i, and t^i_e are, the better the shape restoration performance of the corresponding MSM is.

From Table 2, we can observe that the average maximum position offsets of points 1, 5 and 6 are $\Delta P^1_{\max}$=5.83, $\Delta P^5_{\max}$=2.48, and $\Delta P^6_{\max}$=2.72, respectively, for the proposed MSM, versus $\Delta P^1_{\max}$ = 5.24, $\Delta P^5_{\max}$ = 1.45, and $\Delta P^6_{\max}$=1.56, for the volume MSM. Obviously, the points 1, 5, and 6 on the proposed MSM have the larger offsets than those on the volume MSM. Furthermore, the points 1, 5, and 6 on the proposed MSM provide the average final position offsets of ΔP^1=0.54, ΔP^5=0.21, ΔP^6=0.23, respectively, whereas the corresponding points on the volume MSM are ΔP^1=0.81, ΔP^5=0.27, ΔP^6=0.32, respectively. Similarly, the average final position offset standard deviation of the points 1, 5 and 6 on our MSM are δ^i=0.032, δ^i=0.011 and δ^i =0.0013, respectively. However, the average offset errors of the points 1, 5 and 6 on the canonical MSM are δ^i=0.044, δ^i=0.014 and δ^i=0.0016. Moreover, the steady offset rate of the points 1, 5 and 6 are γ^1_∞=9.26%, γ^5_∞= 8.47%, and γ^6_∞= 8.46%, respectively, for the proposed MSM, versus γ^1_∞=15.46%, γ^5_∞=18.62%, and γ^6_∞=20.51%, for the volume MSM. In brief, although each point of the 12 points on the proposed MSM has the larger average maximum position offsets compared with the corresponding point on the volume MSM, each point on the proposed MSM has better shape restoration performance than the corresponding point on the volume MSM. This indicates that the proposed MSM provides the better shape restoration performance compared the volume MSM.

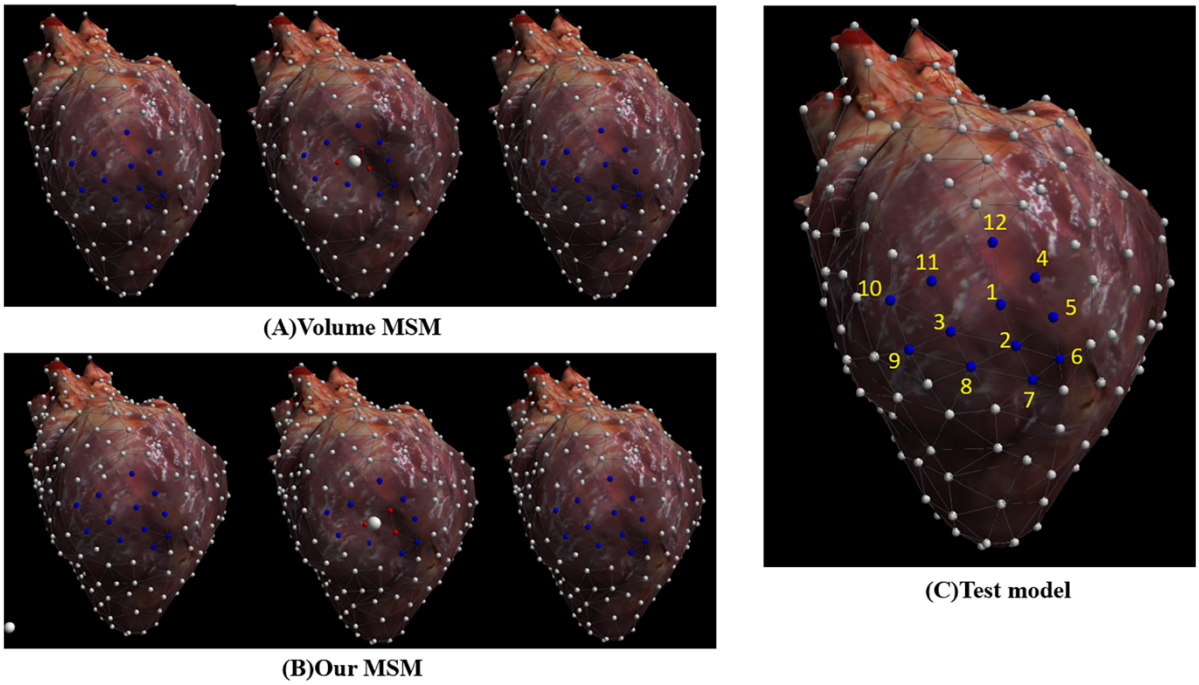

FIGURE 17. Visual effects of the virtual heart for comparing the shape restorations between the volume MSM and our MSM. (A) Initial positions, maximum deformation positions, and final positions on the virtual heart simulated by the volume MSM are described from left to right. (B) Initial positions, maximum deformation positions, and final positions on the virtual heart simulated by our MSM from left to right. (C) Twelves points from the virtual heart simulated by the volume MSM and the proposed MSM is used to evaluate the shape restoration of the two MSMs.

In addition, Table 2 shows that the average steady state transition of each point on the proposed MSM is similar to that of each point on the volume MSM. For instance, the average steady state transitions of points 1, 5 and 6 are t_e^1=3.62, t_e^5=1.32, and t_e^6=1.24, respectively, for the proposed MSM, versus t_e^1=3.80, t_e^5=1.26, and t_e^6=1.19, for the volume MSM. This confirms that the proposed MSM and the volume MSM has the similar average steady state transition.

The following observations can be made about the shape restoration of our MSM: the comparisons between our MSM and the canonical surface MSM simulation show that our MSM can achieve shape restoration with good accuracy and fast convergence in both small and large deformational conditions with different topological models. Furthermore, our MSM can provide slightly better shape restoration performance compared with the volume MSM.

2) REAL-TIME INTERACTION WITH VIRTUAL INSTRUMENTS

To evaluate the computational efficiency of our MSM, frames per second (FPS) is usually applied as the evaluation criterion in real-time interaction simulations. In principle, the higher the FPS is, the better the visual effect. However, a high FPS also lead to a large computational cost. Generally, 24 FPS is essential for achieving a satisfactory degree of visual realism [9]. To compare the FPS of the proposed MSM and the canonical MSM, we use the simple virtual artery vessel model, the mesh model of which consists of 378 points and 720 triangles.

Table 3 shows the results of the real-time interaction between the virtual instrument and the virtual artery vessel model simulated by the proposed and the canonical MSM, respectively. For canonical MSM, the minimum FPS, maximum FPS, and average FPS are 229 FPS, 239 FPS, and 233.5 FPS, respectively. However, for our MSM, the minimum FPS, maximum FPS, and average FPS are 214 FPS, 222 FPS, and 217.5 FPS, respectively. These results demonstrate that our MSM is only slightly slower than the canonical MSM in the real-time interactive simulation. This increase occurs because the new flexion springs in our MSM require a slightly higher computational cost for the shape restoration. Although the canonical MSM has higher average FPS than our MSM in real-time interactive simulation, the average FPS from our MSM is 217.5 FPS, which is far higher than the 24 FPS that is essential for achieving a satisfactory degree of visual realism [9].

Afterwards, we also compare the real-time interaction of our MSM with that of the volume MSM by using a virtual heart model. When the virtual heart is simulated by the volume MSM, the mesh model of the virtual heart includes two parts. One is a surface mesh model, consisting of 252 points and 500 triangles. The other is an internal mesh model, containing 380 points and 224 triangles. The third row of Table 3 shows that for the volume MSM, its minimum, maximum, and average FPS are 175 FPS, 180 FPS, and 177.2 FPS, respectively. On the other hand, we use a mesh model of the virtual heart that are simulated by our surface MSM.

TABLE 3. Comparisons of real-time interaction between our MSM, canonical MSM, and volume MSM. Here, Min (FPS), minimum frames per second; Max (FPS), maximum frames per second; Avg (FPS), average frames per second; FPS, frames per second.

Type	External Size of Mesh Model	Internal size of Mesh Model	Physical Model	Min (FPS)	Max (FPS)	Avg (FPS)
Virtual artery vessel	378 points 720 triangles	0 point 0 triangle	Canonical MSM	229	239	233.5
Virtual artery vessel	378 points 720 triangles	0 point 0 triangle	Our MSM	214	222	217.5
Virtual heart	252 points 500 triangles	380 points 224 triangles	Volume MSM	175	180	177.2
Virtual heart	252 points 500 triangles	0 points 0 triangles	Our MSM	212	228	217.9
Virtual heart	1529 points 3054 triangles	0 points 0 triangles	Our MSM	141	163	151.35

The mesh model is only a surface mesh model, including 252 points and 500 triangles. The fourth row of Table 3 shows that for our MSM, its minimum, maximum, and average FPS are 212 FPS, 228 FPS, and 217.9 FPS. The results indicate that our MSM has the higher FPS compared with the volume MSM in real-time interactive performance. This is due to the fact that our MSM has only the external points and triangles that are the same as those of the volume MSM, however the volume MSM not only the external points and triangles, but also the internal points and triangles. Thus, our MSM has the better real-time interactive performance compared with the volume MSM. This indicates that our MSM can be used to replace the volume MSM for improving computational efficiency.

Furthermore, our MSM is further evaluated by using a complex mesh model of the virtual heart. The mesh model consists of 1529 points and 3054 triangles. The fifth row of Table 3 shows that the minimum, maximum, and average FPS of our MSM are 141 FPS, 163 FPS, and 151.3 5 FPS, respectively. The average FPS from our MSM simulating the complex heart is still able to achieve 151.35 FPS, which is still far higher than the baseline of 24 FPS and enough to reach the requirement of real-time interaction in surgical simulation [9].

In addition, the video screenshots of the real-time interaction process are shown in Fig. 18, where the virtual surgical instrument interacting with the different parts of the virtual heart simulated by our MSM displays good visual effects in the real-time interaction. The abovementioned results further indicate that our MSM has good interaction performance for real-time surgical simulation.

It should be noted that the mesh model of the virtual artery vessel and virtual heart simulated by our MSM consists of 378 points and 720 triangles, and 252 points and 500 triangles, respectively. Since the virtual artery vessel has more points and triangles, it should have the lower minimum FPS, maximum FPS, and average FPS than the virtual heart. However, Table 3 shows that the virtual artery vessel and the virtual heart are similar in the minimum FPS, maximum FPS, and average FPS. This is due to this fact that the virtual heart uses a more complex visual rendering model than the virtual artery vessel.

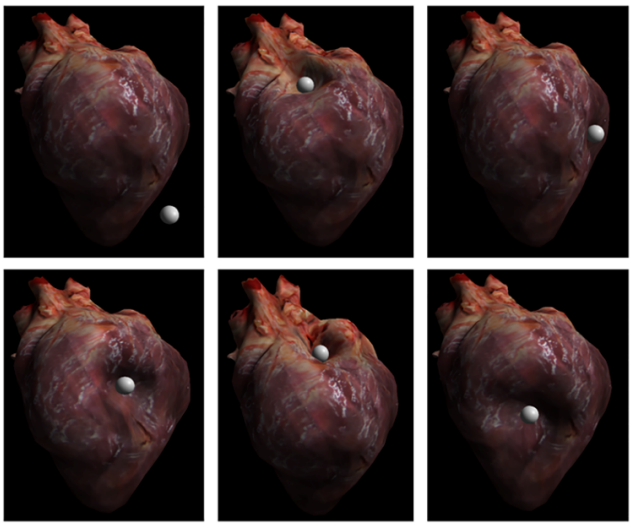

FIGURE 18. Real-time deformational interaction between the surgical instrument and the virtual heart simulated by our MSM.

3) EXPERIMENT FOR COLLISION DETECTION

To evaluate the performance of the proposed DCDVS and HCDVS algorithm, both the DCD and CCD algorithm [56] are used to compare with them. Specially, to achieve a fair comparison between the CCD [56] and the proposed HCDVs, the CCD algorithm [56] also adopts the volume structure techniques of the DCDVS or HCDVS algorithm to conduct continuous interpolation. In addition, we utilize virtual surgical instruments with different sizes to interact with the virtual heart in real time. For the sake of simplicity, the virtual surgical instrument is modeled using white spheres with two different radii of R=1.0 and R=0.1; the mesh model of the virtual heart consists of 1529 points and 3054 triangles. Multiple collision detection experiments are conducted to compare the four collision algorithms [46]. Each algorithm is executed in 30 independent runs with two different surgical

TABLE 4. Comparisons between the CCD, HCDVS, DCDVS, and DCD algorithm. R is the radius of virtual surgical instrument; V_{max} is the maximum speed of virtual surgical instrument; "\" denotes that V_{max} is linked with the interpolation number and can be computed according to formula (29).

Algorithm	R	v_{max}	Velocity Rang	Time (ms)	Success Rate
CCD	1.0	1800	[0, 200]	2.59	100%
Our HCDVS	1.0	\	[0, 200]	1.18	100%
Our DCDVS	1.0	600	[0, 200]	1.16	100%
DCD	1.0	200	[0, 200]	1.12	87.3%
CCD	1.0	1600	[200, 600]	2.62	100%
Our HCDVS	1.0	\	[200,600]	1.20	100%
Our DCDVS	1.0	600	[200, 600]	1.17	93.1%
DCD	1.0	200	[200, 600]	1.15	0%
CCD	1.0	1600	[600, 1562]	2.47	100%
Our HCDVS	1.0	\	[600, 1562]	1.52	100%
Our DCDVS	1.0	600	[600, 1562]	1.17	3.3%
DCD	1.0	200	[600,1562]	1.16	0%
CCD	0.1	1620	[0, 20]	2.52	100%
Our HCDVS	0.1	\	[0, 20]	1.16	100%
Our DCDVS	0.1	420	[0, 20]	1.16	100%
DCD	0.1	20	[0, 20]	1.15	83.4%
CCD	0.1	1620	[20, 420]	2.64	100%
Our HCDVS	0.1	\	[20, 420]	1.21	100%
Our DCDVS	0.1	420	[20, 420]	1.12	94.3%
DCD	0.1	20	[20, 420]	1.12	0%
CCD	0.1	1620	[420, 1562]	2.68	96.7%
Our HCDVS	0.1	\	[420, 1562]	1.67	96.6%
Our DCDVS	0.1	420	[420, 1562]	1.19	6.7%
DCD	0.1	20	[420, 1562]	1.15	0%

instrument sizes, R = 1.0 and R = 0.1 in different speed ranges. The volumetric structure distance is set to l=4.0. The time interval for each collision detection is set to Δt=0.01.

First, we utilize virtual surgical instruments with a radius of R=1.0 to interact with the virtual heart in real time to evaluate the success rate of collision detection for the CCD, HCDVS, DCDVS, and DCD algorithm in different velocity ranges. The corresponding results are given in Table 4. The first four rows of Table 4 show that the detection success rates of the four algorithms are equal to 100%, 100%, 100%, and 87.3%, respectively, when the speed range of the surgical instrument is [0, 200]. All the four algorithms can successfully detect interactions between the surgical instrument and the virtual heart, respectively. However, the DCD algorithm obtains the lowest success rate in velocity range [0, 200].

Afterwards, as the speed range of the surgical instrument varies from [0, 200] to [200, 600], both the CCD and HCDVS algorithm still obtain 100% success rate for checking the collision. The success rate of DCDVS is able to reach 93.1%. However, the DCD algorithm is unable to achieve the available collision detection. Compared with the DCD algorithm, the DCDVS algorithm uses a volume structure to extend the collision range to increase the detection success rate.

Furthermore, as the speed range of the surgical instrument rises to [600, 1562] from [200, 600], both the CCD and HCDVS algorithm are always able to provide 100% success rates. However, the DCDVS and DCD algorithm offer 3.3% and 0% success rates, respectively. Owing to the utilization of the interpolation technique with the volume structure for the CCD and HCDVS, they can achieve accurate and robust collision detection in the high-less speed range [600, 1562]. On the other hand, since the interactive speed rises to [600, 1562], the DCDVS algorithm can hardly detect a collision effectively in spite of the extended collision range.

Subsequently, to verify the adaptability of the CDVS algorithm for detecting surgical instruments with different sizes, the radius of the surgical instrument is decreased to R=0.1. The corresponding results are displayed in Table 4, which shows that the detection success rates of the CCD, HCDVS, and DCDVS algorithm reaches 100% within the speed range of [0, 20], respectively; however, the detection success rate of the DCD algorithm reaches 83.4% within the same speed range. As the speed of the surgical instrument increases from [0, 20] to [20, 420] with a constant R = 0.1, Table 4 shows that the success rates of the CCD and HCDVS algorithm are also maintained at 100%, respectively. The success rate of the DCDVS algorithm has dropped slightly to 94.3%. Unfortunately, the DCD algorithm is unable to achieve successful detection. The DCDVS has introduced a volume structure to extend the range of the collision detection so that it can provide an effective detection success rate. We further increase the speed range to [420, 1562] from [20, 420]. It can be observed from the Table 4 that the success rates of the CCD algorithm and the proposed HCDVS algorithm have been slightly reduced to 96.7% and 96.6%, respectively. Nevertheless, the DCDVS and DCD algorithm provide only 6.7% and 0% success rates, respectively.

In addition, to evaluate computational costs for the four collision detection algorithms, we record the average detection time these algorithms spend in detecting a collision between the virtual heart and virtual surgical instrument with different radii. It can be observed from the Table 4 that for the different speed ranges with different radii of the surgical instrument, the CCD and DCD algorithm always provide the longest and shortest detection time among all the four algorithms, respectively. Interestingly, the average detection time of the proposed HCDVS algorithm is similar in speed ranges [0, 200] and [200, 600] when the radius of the surgical

instrument is equal to 1.0. However, the proposed HCDVS algorithm provides longer average detection time in speed range [600, 1562] than those in speed ranges [0, 200] and [200, 600]. Similarly, we also observe that the proposed HCDVS algorithm is also similar in speed ranges [0, 200] and [200, 600] when the radius of the surgical instrument is equal to 1.0. Also, the proposed HCDVS has longer average detection time in [600, 1562] than in [0, 200] and [200, 600]. In addition, the DCDVS algorithm has the second shortest average detection time among all the four algorithms.

The following observations can be made about the collision detection performance of the four algorithms: The comparisons between the CCD, the HCDVS, DCDVS, and DCD algorithm show that the CCD and HCDVS algorithm can accurately and effectively detect the collision between the virtual heart and surgical instrument with different sizes in different speed ranges. On the other hand, the DCDVS algorithm also achieves accurate and effective collision detection on the condition R = 1.0 and speed range [0, 600], or R=0.1 and speed range [0, 420]. The DCD algorithm achieves a relatively accurate and effective collision detection on the condition R = 1.0 and speed range [0, 200], or R=0.1 and speed range [0, 20]. From the above analysis, the CCD and proposed HCDVS algorithm are restrained by neither the sizes nor the speed ranges of the surgical instruments. The CCD and HCDVS algorithm have good adaptability to surgical instruments with different sizes in the surgical simulation. Specially, by using interpolation techniques, the CCD algorithm always obtains the accuracy and robust collision detection between the virtual heart and surgical instrument with different sizes in different speed ranges. However, the interpolation techniques also increase the computational cost of the collision detection. Compared with the CCD algorithm, the proposed HCDVS algorithm adopts the volume structure to perform the dynamic interpolations. When the detected MSM moves fast, the HCDVS provides more interpolations; otherwise, the HCDVS provides fewer interpolations. Therefore, the HCDVS has a lower computational cost compared with the CCD algorithm.

But it is worth noting that surgeons' actions are very smooth and slow when they perform the surgical operation. This indicates that the virtual surgical simulation does not need to supply the fast collision detection under the normal surgical circumstance. Therefore, our proposed DCDVS can also use to detect the collisions in surgical simulation.

VI. DISCUSSION

To achieve an effective balance between deformation accuracy and real-time performance in surgical simulation, an appropriate deformable physical model is required. Owing to the simple structure, the canonical surface MSM can provide an excellent real-time performance. However, the canonical surface MSM failed to provide a good shape restoration performance when it generating large deformations. This has a significant influence on the deformation accuracy of the canonical surface MSM. To improve the shape restoration behavior, we develop a new surface MSM that has the following characteristics and advantages: (1) A new type of flexion spring opposes bending in terms of the included angle between the initial position vector and the deformational position vector; therefore, this flexion spring enables our surface MSM to accurately and effectively achieve the shape restoration effect of the volumetric MSM when our surface MSM generating large deformations. (2) Our MSM has good real-time interactive capability because it uses the surface topology instead of the volumetric mesh MSM to simulate soft tissue. (3) Our MSM has good adaptability to various topological models using surface triangular topological units, which can be used to establish other topological units, such as rectangular or hexagonal topological units. It is should be noted that the shape restoration performance of the MSM has great impacts on its deformation accuracy. If the MSM is unable to achieve the good shape restoration performance, it cannot provide accurate deformation. Interestingly, the proposed surface MSM overcomes the fundamental drawback that the surface canonical MSM cannot provide an accurate shape restoration performance. Therefore, we can enhance the deformable accuracy of the MSM by improving the shape restoration. This also indicates that our proposed surface MSM has opportunities to replace the volume MSM to achieve a reasonable balance between the real-time interactive performance and good deformation effects in the surgical simulation.

The proposed surface MSM, however, has some limitations that should be further investigated: (1) The accurate deformation of the MSM not only depends on its shape restoration, but also on its mass, spring stiffness and damping coefficients. More specially, the mass, spring stiffness and damping coefficients of the MSM should be linked with the material parameters of real soft tissue, such as Young's modulus and Poisson's ration. In this way, we can obtain more accurate model parameters from the material parameters of the real soft tissue according to biomechanical test results. Since we do not obtain Young's modulus and Poisson's ration of the real heart tissue in time, the model parameters of our MSM have been manually set and regulated until the model achieves a good visual appearance. Therefore, some further investigations on the proposed MSM need to be done in the future. First, we will plan to measure and achieve the material parameters of the real hear tissue such as Young's modulus and Poisson's ration. Then, we will utilize these material parameters to determine the mass, spring stiffness and damping coefficients of the proposed MSM. It is should be noted that it is a challenging problem for the proposed surface MSM to discern its mass, spring stiffness and damping coefficients because it is hard to provide a reasonable match between the coefficients of the proposed surface MSM and the material parameters of the real volumetric tissue. This is to require our further investigations. For instance, we can improve the complex match between these coefficients and material parameters by trying to some new swarm intelligence algorithms [64], [65]. (2) This paper did not

elaborate on the relationships between various reaction forces and different deformation behaviors, such as nonlinearity and viscoelasticity. If the model parameters of our MSM can be reasonably regulated, our MSM can accurately describe these relations between various forces and deformational behaviors. Thus, the abovementioned issues will be investigated in future work. (3) In addition, a reasonable combination between the proposed surface MSM and the volume MSM may achieve a promising balance between the real-time interactive performance and good deformation effects in the surgical simulation. Finally, we will realize applications in the surgical simulation by using the proposed MSM.

In addition, the interactive accuracy also relies on the interactive collision in the surgical simulation because missed detections may lead to unacceptable simulation or remarkable distortion results. The DCD algorithms have low computational costs and provide great real-time interactive collision detections. Consequently, it is used to conduct interactive detections between the surgical instrument and virtual soft tissue in surgical simulation. However, it may miss detections with fast movements, great large deformations, or surgical with very small sizes in surgical simulation. The CCD algorithms can offer accurate collision detection by using continuous interpolations. However, they have high computational costs in the real-time interactive detection. To achieve a reasonable compromise between the real-time interaction and detection accuracy, we propose two new collision detection algorithms: DCDVS and HCDVS algorithm. First, the features and advantages of the DCDVS algorithm are demonstrated as follows: (1) The DCDVS algorithm can accurately and effectively detect relatively fast collision behavior between surgical instruments with different sizes and our MSM by extending the effective range of collision detection. (2) The CDVS algorithm does not need to limit the sizes of surgical instruments to a certain range to ensure the validity and reliability of the collision detection between the surgical instrument and our MSM; the fundamental reason for these advantages is that our algorithm utilizes volumetric structure detection instead of face structure detection, which enables the effective detection range to be extended from the length $2R$ to the length $2R + l$; this extension can enable the maximum velocity of the surgical instrument to increase from $V_{max} = 2R/\Delta t$ to $V_{max} = (2R + l)/\Delta t$. Subsequently, the characteristics and advantages of the HCDVS algorithm are demonstrated as follows: (1) The HCDVS algorithm can accurately and effectively detect great fast collision behaviors between surgical instruments with different sizes and our MSM by combining the interpolation technique of the CCD algorithm with the CDVS algorithm. (2) Compared with the CCD algorithm, the HCDVS algorithm adopts the volume structure to perform the dynamic interpolations. When the detected MSM move fast, the HCDVS provide more interpolations; otherwise, the HCDVS provide less interpolations. Therefore, the HCDVS have a lower computational cost compared with the CCD algorithm. In addition, since the DCDVS and HCDVS algorithm can effectively improve the detection accuracy between the virtual surgical instrument and our MSM, the DCDVS and HCDVS algorithm also further improves the interactive accuracy in surgical simulation. It should be noticed that the DCDVS fails to provide accurate detection between the surgical instrument and virtual heart, when there are fast interactions between them. In fact, surgeons' operations are relatively smooth and slow when they perform the surgical operation in most surgical procedures. This indicates that the virtual surgical simulation does not need to provide the fast collision detection under the normal surgical circumstance. Therefore, the proposed DCDVS algorithm can also use to detect the collisions between the surgical instrument and virtual heart, decreasing computational costs and improving the real-time interaction in surgical simulation.

Similarly, some further investigations will be done for the proposed collision detection algorithm: (1) The proposed HCDVS algorithm uses volume structures to replace the triangles to conduct dynamic interpolations. Actually, the volume structure distance l is linked with the interpolation number, and also influences the computational cost and detection accuracy of the HCDVS algorithm. In the future, we will consider how to establish effective relations between the volume structure distance and the interpolation number to improve the performance and adaptability for the HCDVS algorithm. (2) This paper aims to solve the collision between the surgical instrument and the proposed MSM algorithm. Next, we will further consider the tissue-tissue collisions for the proposed algorithm in the surgical simulation. (3) We will combine our MSM and collision detection algorithm with other MSMs to further implement a more effective balance between the interactive accuracy and the real-time performance in surgical simulation.

VII. CONCLUSION

In this paper, we present a new surface MSM for improving the shape restoration performance and real-time interactive performance. Since the new flexion spring is developed in our MSM, our MSM can achieve accurate shape restoration in the deformational simulation. Since surface triangular topological units are used in our MSM, the model has good adaptability and real-time interactive capability for simulating different mesh topological models. Since the DCDVS and HCDVS algorithm are developed for our MSM, our MSM can achieve accuracy and fast collision between surgical instruments with different sizes and our MSM.

REFERENCES

[1] C. Basdogan, M. Sedef, M. Harders, and S. Wesarg, "VR-based simulators for training in minimally invasive surgery," *IEEE Comput. Graph. Appl.*, vol. 27, no. 2, pp. 54–66, Mar. 2007.

[2] W. Shen and W.-Q. Zeng, "Virtual Reality technology in modern medicine," in *Proc. Int. Conf. Audio Lang. Image Process.*, 2010, pp. 557–561.

[3] S. Delorme, D. Laroche, R. DiRaddo, and F. Del Maestro, "NeuroTouch: A physics-based virtual simulator for cranial microneurosurgery training," *Neurosurgery*, vol. 71, pp. 32–42, Sep. 2012.

[4] Y. Zou, P. X. Liu, Q. Cheng, P. Lai, and C. Li, "A new deformation model of biological tissue for surgery simulation," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3494–3503, 2017.
[5] R. Aggarwal, P. Crochet, A. Dias, A. Misra, P. Ziprin, and A. Darzi, "Development of a virtual reality training curriculum for laparoscopic cholecystectomy," *Brit. J. Surg.*, vol. 96, no. 9, pp. 1086–1093, 2009.
[6] A. P. Santhanam, F. G. Hamza-Lup, and J. P. Rolland, "Simulating 3-D lung dynamics using a programmable graphics processing unit," *IEEE Trans. Inf. Technol. Biomed.*, vol. 11, no. 5, pp. 497–506, Sep. 2007.
[7] S. Cotin, H. Delingette, and N. Ayache, "Real-time elastic deformations of soft tissues for surgery simulation," *IEEE Trans. Vis. Comput. Graphics*, vol. 5, no. 1, pp. 62–73, Jan. 1999.
[8] X. P. Liu, S. Xu, H. Zhang, and L. Hu, "A new hybrid soft tissue model for visio-haptic simulation," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 11, pp. 3570–3581, Nov. 2011.
[9] A. Nealen, M. MÃijller, R. Keiser, E. Boxerman, and M. Carlson, "Physically based deformable models in computer graphics," *Comput. Graph. Forum.*, vol. 25, no. 4, pp. 809–836, 2006.
[10] M. Bro-Nielsen and S. Cotin, "Real-time volumetric deformable models for surgery simulation using finite elements and condensation," in *Proc. Int. Conf. Vis. Biomed. Comput.*, vol. 15, 1996, pp. 535–540.
[11] D. L. James and D. K. Pai, "ArtDefo: Accurate real time deformable objects," in *Proc. Conf. Comput. Graph. Interact. Techn.*, 1999, pp. 65–72.
[12] J. Berkley *et al.*, "Creating fast finite element models from medical images," *Stud. Health Technol. Inform.*, vol. 70, pp. 26–32, Jan. 2000.
[13] G. Debunne, M. Desbrun, M.-P. Cani, and A. H. Barr, "Dynamic real-time deformations using space & time adaptive sampling," in *Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, vol. 83, 2001, pp. 31–36.
[14] W. Wu and P. A. Heng, "A hybrid condensed finite element model with GPU acceleration for interactive 3D soft tissue cutting," *Comput. Animation Virtual Worlds*, vol. 15, nos. 3–4, pp. 219–227, 2004
[15] B. Lee, D. C. Popescu, B. Joshi, and S. Ourselin, "Efficient topology modification and deformation for finite element models using condensation," *Stud. Health Technol. Inform.*, vol. 119, pp. 299–304, Jan. 2006.
[16] J. Georgii and R. Westermann, "Corotated finite elements made fast and stable," in *Proc. Workshop Virtual Reality Interact. Phys. Simulations*, 2013, pp. 11–19.
[17] M. MÃijller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler, "Stable real-time deformations," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation (SCA)*, 2002, pp. 49–54.
[18] M. Hauth and W. Strasser, "Corotational simulation of deformable solids," *J. WSCG*, vol. 12, nos. 1–3, pp. 137–145, 2004.
[19] K. Miller, G. Joldes, D. Lance, and A. Wittek, "Total lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation," *Commun. Numer. Methods Eng.*, vol. 23, no. 2, pp. 121–134, 2007.
[20] K. Sangpradit, H. Liu, P. Dasgupta, K. Althoefer, and L. D. Seneviratne, "Finite-element modeling of soft tissue rolling indentation," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 12, pp. 3319–3327, Dec. 2011.
[21] W. Tang and T. R. Wan, "Constraint-based soft tissue simulation for virtual surgical training," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 11, pp. 2698–2706, Nov. 2014.
[22] X. Provot, "Deformation constraints in a mass-spring model to describe rigid cloth behavior," *Integr. VlSI J.*, vol. 23, no. 19, pp. 147–154, 1995.
[23] Y. Lee, D. Terzopoulos, and K. Waters, "Realistic modeling for facial animation," in *Proc. Conf. Comput. Graph. Interact. Techn.*, 1995, pp. 55–62.
[24] E. Keeve, S. Girod, and B. Girod, "Craniofacial surgery simulation," in *Proc. Int. Conf. Vis. Biomed. Comput.*, vol. 1131, 1996, pp. 541–546.
[25] J. Brown, S. Sorkin, J.-C. Latombe, K. Montgomery, and M. Stephanides, "Algorithmic tools for real-time microsurgery simulation," *Med. Image Anal.*, vol. 6, no. 3, pp. 289–300, 2002.
[26] L. P. Nedel and D. Thalmann, "Real time muscle deformations using mass-spring systems," in *Proc. Comput. Graph. Int.*, 1998, pp. 156–165.
[27] C.-B. Bao, B. L. Wang, L. Zhuo, and C. Ming, "A dynamic mass spring model for simulation of soft tissue deformation," *Chin. J. Syst. Simul.*, vol. 18, no. 4, pp. 846–847, 2006.
[28] Y. Wang, Y. Xiong, K. Xu, K. Tan, and G. Guo, "An improved mass-spring model using surface mesh," *Chin. J. Comput.-Aided Des. Comput. Graph.*, vol. 19, no. 2, pp. 168–171, 2007.
[29] W. Mollemans, F. Schutyser, J. Van Cleynenbreugel, and P. Suetens, "Fast soft tissue deformation with tetrahedral mass spring model for maxillofacial surgery planning systems," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI*, 2004, pp. 371–379.
[30] G. San-Vicente, I. Aguinaga, and J. T. Celigueta, "Cubical mass-spring model design based on a tensile deformation test and nonlinear material model," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 2, pp. 228–241, Feb. 2012.
[31] Y. Duan *et al.*, "Volume preserved mass–spring model with novel constraints for soft tissue deformation," *IEEE J. Biomed. Health Inform.*, vol. 20, no. 1, pp. 268–280, Jan. 2016.
[32] B. A. Lloyd, G. Székely, and M. Harders, "Identification of spring parameters for deformable object simulation," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 5, pp. 1081–1094, Sep. 2007.
[33] T. Liu, A. W. Bargteil, J. F. O'Brien, and L. Kavan, "Fast simulation of mass-spring systems," *ACM Trans. Graph.*, vol. 32, no. 6, 2013, Art. no. 214.
[34] L. Cooper and S. Maddock, "Preventing collapse within mass-spring-damper models of deformable objects," in *Proc. Int. Conf. Central Eur. Comput. Graph. Vis.*, 1997, pp. 70–78.
[35] M. Teschner, S. Girod, and B. Girod, "Direct computation of nonlinear soft-tissue deformation," in *Proc. Vis., Modeling, Vis.*, 2000, pp. 22–24.
[36] E. Basafa, F. Farahmand, and G. Vossoughi, "A non-linear mass-spring model for more realistic and efficient simulation of soft tissues surgery," *Stud. Health Technol. Inform.*, vol. 132, no. 1, pp. 23–25, 2008.
[37] Z. Huangfu, L. Yan, and X. Liu, "An improved mass-spring model for simulation of soft tissue deformation," *J. Inf. Comput. Sci.*, vol. 10, no. 17, pp. 5551–5558, 2013.
[38] N. Omar, Y. Zhong, R. N. Jazar, A. Subic, J. Smith, and B. Shirinzadeh, "Soft tissue modelling with conical springs," *Bio-Med. Mater. Eng.*, vol. 26, no. 1, pp. S207–S214, 2015.
[39] M. T. Ahmadian and A. A. Nikooyan, "Modeling and prediction of soft tissue directional stiffness using *in-vitro* force," *Int. J. Sci. Res.*, vol. 16, pp. 385–389, 2006.
[40] A. Horton, A. Wittek, G. R. Joldes, and K. Miller, "A meshless total lagrangian explicit dynamics algorithm for surgical simulation," *Int. J. Numer. Methods Biomed. Eng.*, vol. 26, no. 8, pp. 977–998, 2010.
[41] D. Baraff, A. Witkin, and M. Kass, "Untangling cloth," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 862–870, 2003.
[42] G. van den Bergen, "Efficient collision detection of complex deformable models using AABB trees," *J. Graph. Tools.*, vol. 2, no. 4, pp. 1–13, 1997.
[43] D. Baraff, "Curved surfaces and coherence for non-penetrating rigid body simulation," in *Proc. ACM SIGGRAPH Comput. Graph.*, vol. 24, no. 4, Sep. 1990, pp. 19–28.
[44] J. D. Cohen, M. C. Lin, D. Manocha, and M. Ponamgi, "I-COLLIDE: An interactive and exact collision detection system for large-scale environments," in *Proc. Symp. Interact. 3D Graph.*, vol. 170, 1995, pp. 189–196.
[45] M. C. Lin and S. Gottschalk, "Collision detection between geometric models: A survey," in *Proc. 8th IMA Conf. Math. Surf.*, Southend-on-Sea, U.K., 1998, pp. 37–56.
[46] P. Shirley, M. Ashikhmin, and S. Marschner, *Fundamentals of Computer Graphics*, 3rd ed. Boca Raton, FL, USA: CRC Press, 2009.
[47] S. Redon, A. Kheddar, and S. Coquillart, "Fast continuous collision detection between rigid bodies," *Eurograph. Comput. Graph. Forum*, vol. 21, no. 3, pp. 279–287, 2002.
[48] S. Redon, Y. J. Kim, M. C. Lin, and D. Manocha, "Fast continuous collision detection for articulated models," in *Proc. ACM Symp. Solid Modeling Appl.*, 2004, pp. 145–156.
[49] X. Zhang, S. Redon, M. Lee, and Y. J. Kim, "Continuous collision detection for articulated models using Taylor models and temporal culling," *ACM Trans. Graph.*, vol. 26, no. 3, 2007, Art. no. 15.
[50] N. K. Govindaraju *et al.*, "Interactive collision detection between deformable models using chromatic decomposition," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 991–999, 2005.
[51] W. S.-K. Wong and G. Baciu, "Dynamic interaction between deformable surfaces and nonsmooth objects," *IEEE Trans. Vis. Comput. Graphics*, vol. 11, no. 3, pp. 329–340, 2005.
[52] S. Curtis, R. Tamstorf, and D. Manocha, "Fast collision detection for deformable models using representative-triangles," in *Proc. Symp. Interact. 3D Graph. Games*, 2008, pp. 61–69.
[53] M. Tang, S. Curtis, S. E. Yoon, and D. Manocha, "ICCD: Interactive continuous collision detection between deformable models using connectivity-based culling," *IEEE Trans. Vis. Comput. Graphics*, vol. 15, no. 4, pp. 544–557, Jul. 2009.
[54] M. Tang, D. Manocha, and R. Tong, "Fast continuous collision detection using deforming non-penetration filters," in *Proc. ACM SIGGRAPH Symp. Interact. 3D Graph. Games (I3D)*, New York, NY, USA, 2010, pp. 7–13.

[55] A. Sud, N. Govindaraju, R. Gayle, I. Kabul, and D. Manocha, "Fast proximity computation among deformable models using discrete Voronoi diagrams," in *Proc. ACM SIGGRAPH*, 2006, pp. 1144–1153.
[56] M. Tang, D. Manocha, S.-E. Yoon, P. Du, J.-P. Heo, and R.-F. Tong, "VolCCD: Fast continuous collision culling between deforming volume meshes," *ACM Trans. Graph.*, vol. 30, no. 5, pp. 111:1–111:15, 2011.
[57] A. Fukuhara *et al.*, "Proposition and evaluation of a collision detection method for real time surgery simulation of opening a brain fissure," *ROBOMECH J.*, vol. 1, no. 1, pp. 1–14, 2014.
[58] J. Allard, F. Faure, H. Courtecuisse, F. Falipou, C. Duriez, and P. G. Kry, "Volume contact constraints at arbitrary resolution," in *Proc. ACM SIGGRAPH*, 2010, pp. 1–10.
[59] F. Faure, J. Allard, F. Falipou, and S. Barbier, "Image-based collision detection and response between arbitrary volumetric objects," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation*, Cham, Switzerland, 2008, pp. 155–162.
[60] I. J. Palmer and R. L. Grimsdale, "Collision detection for animation using sphere-trees," *Comput. Graph. Forum.*, vol. 14, no. 2, pp. 105–116, 2010.
[61] S. Gottschalk, M. C. Lin, and D. Manocha, "OBBTree: A hierarchical structure for rapid interference detection," in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, 1996, pp. 171–180.
[62] J. T. Klosowski, M. Held, J. S. B. Mitchell, H. Sowizral, and K. Zikan, "Efficient collision detection using bounding volume hierarchies of k-DOPs," *IEEE Trans. Vis. Comput. Graph.*, vol. 4, no. 1, pp. 21–36, Jan. 1998.
[63] M. Hutter and A. Fuhrmann, "Optimized continuous collision detection for deformable triangle meshes," *J. WSCG*, vol. 15, nos. 1–3, pp. 25–32, 2008.
[64] C. Li, Z. Song, J. Fan, Q. Cheng, and P. X. Liu, "A brain storm optimization with multi-information interactions for global optimization problems," *IEEE Access*, vol. 6, pp. 19304–19323, 2018.
[65] Z. Song, J. Peng, C. Li, and P. X. Liu, "A simple brain storm optimization algorithm with a periodic quantum learning strategy," *IEEE Access*, vol. 6, pp. 19968–19983, 2018.

CHUNQUAN LI received the B.Sc., M.Sc., and Ph.D. degrees from Nanchang University, Nanchang, China, in 2002, 2007, and 2015, respectively. He has been with the School of Information Engineering, Nanchang University, since 2002, where he is currently an Associate Professor and a Young Scholar of Ganjiang River. He has published over 30 research articles.

He is also a Visiting Professor with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada. His current interests include virtual surgery simulation, robotics, haptics, computing intelligence, and their applications to biomedical engineering.

JIAJUN DING received the B.Sc. and M.Sc. degrees from Nanchang University, Nanchang, China, in 2014 and 2017, respectively. His current research interests include virtual surgical simulation and haptics.

ZHICHAO HONG received the B.Sc. degree from Hangzhou Dianzi University, Nanchang, China, in 2015. He is currently pursuing the M.Sc. degree in information and communication engineering with Nanchang University, Nanchang, Jiangxi, China.

His current research interests include virtual surgical simulation and haptics.

YUCHENG PAN received the B.Sc. degree from Nanchang University, Nanchang, Jiangxi, China, in 2017, where he is currently pursuing the M. Sc. degree in information and communication engineering.

His current research interests include swarm particle swarm optimization, genetic algorithms, and other computational intelligence techniques.

PETER X. LIU (M'02–SM'07) received the B.Sc. and M.Sc. degrees from Northern Jiaotong University, Beijing, China, in 1992 and 1995, respectively, and the Ph.D. degree from the University of Alberta, Edmonton, AB, Canada, in 2002.

He has been with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada, since 2002, where he is currently a Professor and the Canada Research Chair. He has published over 280 research articles. His current interests include computing intelligence, interactive networked systems and teleoperation, haptics, micro-manipulation, robotics, intelligent systems, context-aware intelligent networks, and their applications to biomedical engineering.

Dr. Liu was a recipient of the 2007 Carleton Research Achievement Award, the 2006 Province of Ontario Early Researcher Award, the 2006 Carty Research Fellowship, the Best Conference Paper Award of the 2006 IEEE International Conference on Mechatronics and Automation, and the 2003 Province of Ontario Distinguished Researcher Award. He serves as an Associate Editor for several journals, including the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE/ASME TRANSACTIONS ON MECHATRONICS, the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and the IEEE ACCESS. He is a licensed member of the Professional Engineers of Ontario and a fellow of the Engineering Institute of Canada.