Bleeding Simulation With Improved Visual Effects for Surgical Simulation Systems

Wen Shi, *Student Member, IEEE*, Peter Xiaoping Liu, *Fellow, IEEE*, and Minhua Zheng, *Member, IEEE*

***Abstract*—In surgical simulation, the Navier–Stokes (N–S) equation is commonly employed to imitate the physical characteristics of bleeding and the smooth particle hydrodynamics (SPHs) algorithm is applied to solve the numerical solution of the N–S equation. However, blood is viscous, incompressible and non-Newtonian fluid whose physical properties cannot be fully incorporated by the simple N–S equation, and the kernel approximation of the SPH algorithm may lead to both edge and volume distortions plus high computational cost. In this paper, both the tension force and the effect of platelets on the viscous force of bleeding particles are incorporated into the N–S equation in order to render more realistic visual effect and biological features of bleeding in surgical simulation. Constant core radius of the kernel function of the SPH algorithm is substituted with a function of particle density, avoiding potential edge distortions in simulating bleeding area. A repulsive force between particles is introduced, which effectively prevents volume distortions. Besides, accelerated search for particles based on the cube mesh improves the computational efficiency. The simulation results show that the presented simulation method leads to smooth bleeding surface and improves the visual effects of edge and volume in comparison with existing methods, and relatively high computational efficiency can be achieved as well.**

***Index Terms*—Bleeding simulation, kernel function, Navier–Stokes (N–S) equation, repulsive force, smooth particle hydrodynamic (SPH) algorithm.**

I. Introduction

VIRTUAL surgery provides an invaluable and promising alternative to conventional surgical training approaches which are mainly based on animals and cadavers [1], [2]. Bleeding simulation plays an important role when surgeons are trained through virtual surgery platforms. Bleeding is inevitable and often associated with the entire surgical process. It has unique physical properties and its kinetic status is diversified. It not only improves the surgeon's ability to deal with unexpected situations but also increases the visual realism of the surgical simulation environment [3], [4].

In the early stages, owing to the limitation of computer performance and the interactivity operation of virtual surgery system, algorithms based on the computational fluid dynamics models were difficult to be applied. Simplified linear models were used instead of complicated nonlinear models to obtain better real-time performance. Çakmak and Kühnapfel [5] adopted texture animation to simulate bleeding on the surface of organs in their virtual laparoscopic surgery system, which showed good visual effect of flow and fine instantaneity. According to the wave equation introduced by Basdogan *et al.* [6] and Kass and Miller [7] presented a surface-flow model. Both Çakmak and Basdogan achieved the characteristics of blood flow through the particle system. However, without the hydrodynamic model, the physical properties of bleeding could not be satisfied.

The Navier–Stokes (N–S) equation describes physical fluids in the form of mathematical equations, and produces practical benefits in many fields of production [8]. Most physical properties of blood conform to the N–S equation. The difficulty of solving the analytical solution immensely reduces computation speed and further restricts the real-time performance of the virtual surgery system. For improving the real-time performance, Stam [9] presented an algorithm to calculate the motion state of liquids, dividing grids in the research area by discrete ways. The physical quantities of each grid vertex were computed by the N–S equation to obtain the motion state of the liquid passing through the grid vertex at the next time. The computation cost of this algorithm is still intensive as a result of dense grids, therefore it can only be used to simulate the bleeding of simple scenes, such as the blood flow in vessels. Rianto and Li [10] adopted rough grids and acceleration by GPU to solve the N–S equation by means of curvilinear interpolation, achieving real-time, and realistic simulation results.

The grid-based algorithm has been extensive applied in computer scenarios, such as the lakes with exquisite ripples since it is efficient in surface rendering. However, the robustness is not ideal in the interactive environment. The smoothed particle hydrodynamics (SPHs) is more suitable for solving the N–S equation in bleeding simulation since it is a grid-free algorithm based on Lagrangian particle formulation. The SPH algorithm greatly reduce the computational complexity of solving the N–S equation under the assurance of fidelity requirement. Müller *et al.* [11] presented an

Manuscript received October 2, 2018; accepted November 17, 2018. Date of publication December 7, 2018; date of current version January 19, 2021. This work was supported in part by the National Science Foundation of China under Grant 61773051, Grant 61761166011, and Grant 51705016, in part by the Beijing Natural Science Foundation under Grant 4172048, and in part by the Fundamental Research Funds for the Central Universities under Grant 2016RC021 and Grant 2017JBZ003. This paper was recommended by Associate Editor L. Wang. *(Corresponding authors: Peter Xiaoping Liu; Minhua Zheng.)*

W. Shi and M. Zheng are with the School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China (e-mail: mhzheng@bjtu.edu.cn).

P. X. Liu is with the School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China, and also with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: xpliu@sce.carleton.ca).

Color versions of one or more of the figures in this paper are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2018.2883406

interactive algorithm based on the SPH algorithm to simulate the bleeding of vessel rupture. They introduced tension when calculating the N–S equation to achieve the smooth and vivid simulated surface. Qin *et al.* [12] studied the mechanical properties of bleeding in the blood vessel by combining the spring mass model with the SPH algorithm and modified the viscosity coefficient of the particles to simulate the non-Newtonian characteristics of the bleeding in solving the N–S equation. With increasing applications of the SPH algorithm, some defects were exposed when solving problems by particle approximation, and different kinds of modifications and variants were presented to improve the defects of initial SPH algorithm. For example, Hu and Adams [13] proposed an SPH algorithm with the angular momentum conservative. Some researchers have analyzed the computational characteristics of the SPH algorithm [14]. Swegle *et al.* [15] identified the unstable tensile problem with the SPH method that can be significant for strength of materials. Monaghan [16]–[18] proposed symmetrization formulations that are good modifications to the previous problems. Liu *et al.* [19], [20] presented a finite particle method, which approximate field variables at a set of arbitrarily admeasured particles by a set of basis functions. Fang *et al.* [21], [22] proposed a regularized Lagrangian finite point method for simulating the incompressible viscous fluids. Other prominent revised methods of the SPH algorithm include the moving least square particle hydrodynamics [23], [24], the integration kernel correction [25], the reproducing kernel particle method [26], [27], and so forth.

The tissues of human body are different from other materials, and they have special physical properties due to their special ingredients. For instance, blood is different from water because of the cells it contains, such as the platelets that cause clotting. In our previous work, we studied the deformation properties of biological tissues [28]–[30] and carried out a preliminary study on the properties of bleeding [31]. Although many kinetic properties of blood conform to the N–S equation, the unique physical properties of blood cannot be fully described by the N–S equation. In order to realize more realistic bleeding simulation, the special biological features of blood need to be considered.

In this paper, we incorporate more physical properties of blood into the simulation of bleeding by improving both the N–S equation and the SPH algorithm. In specific, we introduce both the tension force and the effect of platelets on the viscous force in order to improve visual effect. The constant core radius of the SPH kernel function is replaced with a function of particle density, improving the accuracy and stability of the SPH algorithm. In order to eliminate the volumetric distortion caused by penetrations between particles, we introduce a repulsive force between particles. The resulting bleeding simulation presents substantially improved visual realism without reducing the real-time performance.

The rest of this paper is organized as follows. The principles of the N–S equation and the traditional SPH method are introduced in Section II. We show how the N–S equation and the SPH algorithm are improved in Section III. The usage of repulsive force in solving the volume distortion caused by penetrations between particles is presented in Section IV. The comparison of simulation results is presented in Section V and this paper is concluded in Section VI.

II. Navier–Stokes Equation and Smooth Particle Hydrodynamics

A. Navier–Stokes Equation

The N–S equation combines momentum conservation, mass conservation, and energy conservation to describe fluids [32]. Newton's second law is the fundamental theoretical basis of the N–S equation, that is, $F = ma$.

The forces received by each fluid element is mainly composed of two types, i.e., the external forces determined by environmental factors and the internal forces affected by its own properties. The strain variables are easier to get than the complex stress calculations. Therefore, the magnitude of stress can be obtained through the relationship between stress and strain. Then the necessary physical characteristics can be computed through the continuity equation.

Taking an example of forces in a single direction, such as the X direction, the total force of the fluid element is as follows:

$$F = \left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z}\right) dxdydz + \rho X dxdydz \quad (1)$$

where σ_{xx} is the compressive stress, τ_{yx} and τ_{zx} are the shear stresses, and X is the mass force.

Substituting (1) into the formula $F = ma$

$$\left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z}\right) dxdydz + \rho X dxdydz = \frac{du_x}{dt}\rho dxdydz \quad (2)$$

where u_x is the strain.

Simplifying (2)

$$\rho\frac{du_x}{dt} = \left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z}\right) + \rho X. \quad (3)$$

The same as the X direction, in the directions of Y and Z

$$\rho\frac{du_y}{dt} = \left(\frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{zy}}{\partial z}\right) + \rho Y \quad (4)$$

$$\rho\frac{du_z}{dt} = \left(\frac{\partial \sigma_{zz}}{\partial z} + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y}\right) + \rho Z. \quad (5)$$

The transformation between strain u and shear stress τ is as follows:

$$\tau_{xy} = \tau_{yx} = \mu\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right). \quad (6)$$

The transformation between strain u and normal stress σ is as follows:

$$\sigma_{xx} = -p + 2\mu\frac{\partial u_x}{\partial x} - \frac{2}{3}\mu\nabla'\vec{u}. \quad (7)$$

Merging (3), (6), and (7), getting the following vector formula:

$$\rho\frac{d\vec{u}}{dt} = -\nabla p + \mu\nabla^2\vec{u} + \frac{1}{3}\mu\nabla\left(\nabla'\vec{u}\right) + \rho\vec{F}. \quad (8)$$

When it is used to calculate the incompressible fluid, i.e., $\nabla'\vec{u} = 0$, the N–S equation can be expressed as

$$\rho\frac{d\vec{u}}{dt} = -\nabla p + \mu\nabla^2\vec{u} + \rho\vec{F}. \quad (9)$$

B. Smooth Particle Hydrodynamics

SPHs is a Lagrange particle algorithm which is used to simulate fluid flow. It is considered to be the earliest gridless algorithm. The SPH algorithm was initially presented by Gingold and Monaghan [33] and Lucy [34]. It became an algorithm for simulating fluids and was later extended to simulate incompressible flows such as virtual bleeding.

The fundamental of the SPH algorithm is to use a smooth function to approximate the field of variables. The particle approximation method is used to approximate every variable in the series of partial differential equations, and a series of ordinary differential equations is obtained. The ordinary differential equations are solved by using explicit integral.

The derivation of the SPH algorithm is as follows:

$$f(x) = \int_{\Omega} f(x')\delta(x - x')dx' \tag{10}$$

where function δ is expressed as

$$\delta(x - x') = \begin{cases} \infty & x = x' \\ 0 & x \neq x'. \end{cases} \tag{11}$$

Since $f(x)$ is defined and continuous in the interpolated region Ω, (10) can be established accurately.

When $W(x - x', h)$ is used to replace $\delta(x - x')$, $f(x)$ can be changed to

$$f(x) = \int_{\Omega} f(x')W(x - x', h)dx' \tag{12}$$

where $W(x - x', h)$ is defined as the *kernel function*, and h is the *core radius*. The value of h determines the size of support domain.

When h approaches 0, the kernel function are equal to the delta function, that is, $W(x - x', h) = \delta(x - x')$, otherwise, we can used the following equation to approximate their relationship:

$$< f(x) >= \int_{\Omega} f(x')W(x - x', h)dx'. \tag{13}$$

The SPH algorithm is a pure Lagrange particle method without grid generation, and it is mainly based on interpolation. The partial differential equation is transformed into an integral equation by the kernel approximation method, and then the continuous integral equation is transformed into a discrete equation by the particle approximation method, as follows:

$$f(x) \approx \sum_{j=1}^{N} f(x_j)W(x_i - x_j, h_i)\Delta V_j = \sum_{j=1}^{N} \frac{m_j}{\rho_j} f(x_j)W_{ij}. \tag{14}$$

III. Modifying the N–S Equation and the SPH Algorithm

As mentioned before, it is extremely demanding to obtain the analytic solution of the partial differential equations except for some simple cases. Using the SPH algorithm in obtaining the numerical solution can increase the speed of computation. In this process, the accuracy of the solution and the simulation realism are directly determined by the physical characteristics of blood and the alternative of the kernel function.

On the right side of the N–S equation, i.e., (9), we need to calculate three forces: 1) the pressure ∇p which is mostly caused by jacent bleeding particles; 2) the viscous force $\mu\nabla^2\overrightarrow{u}$ affected by platelets, which increase with time and speed up the coagulation of blood [35]; and 3) the external force $\rho\overrightarrow{F}$ which is mainly effected by gravity and surgical operation and is allowed to be directly replaced into the formula. When the particles get into the support domain of each other, the pressure will be generated by interaction between the particles [36]. Surface shrinkage caused by the tension force owing to the attraction between particles is also relatively obvious since the viscosity of blood is much stronger than that of general fluid [37]. In this paper, both the effects of tension force and platelets are considered to improve the physical characteristics of bleeding simulation.

A. Pressure and Tension

The location of particle i should be determined before calculating the pressure. If it is near the bleeding surface, both pressure, and tension need to be calculated. Inside the bleeding area, the tension is trivial and can be ignored, and the particle is only subjected to pressure.

The acceleration of particle i generated by the pressure can be written as

$$a_i^p = \frac{f_i^p}{\rho_i} = -\frac{1}{\rho_i}\sum_j m_j \frac{p_j}{\rho_j}\nabla W(r_i - r_j, h) \tag{15}$$

where j represents the particles in the support domain of particle i.

The mass of each bleeding particle is assumed to be equal to m. According to (15), the interaction forces computed between contiguous particles are unequal, i.e., the pressure of particle i for particle j is not the same as the pressure of particle j for particle i, which does not follow the Newton's third law. Therefore, it is necessary to revise (15) as follows:

$$a_i^p = \frac{f_i^p}{\rho_i} = -m\sum_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2}\right)\nabla W(r_i - r_j, h) \tag{16}$$

or

$$a_i^p = \frac{f_i^p}{\rho_i} = -m\sum_j \left(\frac{p_i + p_j}{2\rho_i\rho_j}\right)\nabla W(r_i - r_j, h). \tag{17}$$

Because of the high computational complexity of (16), (17) is a better choice due to its relatively high efficiency. Moreover, because of the lack of adjacent particles, minimal density will occur on the edge particles. Using (17) to compute the density is more accurate.

The pressure of each particle can be obtained before calculating the acceleration. The formula of liquid state is as follows:

$$p = k\rho \tag{18}$$

where k is a constant, which is determined by the temperature. In liquid simulation, the formula of liquid state revised by Desbrun and Gascuel [38] is generally used

$$p = k(\rho - \rho_0) \tag{19}$$

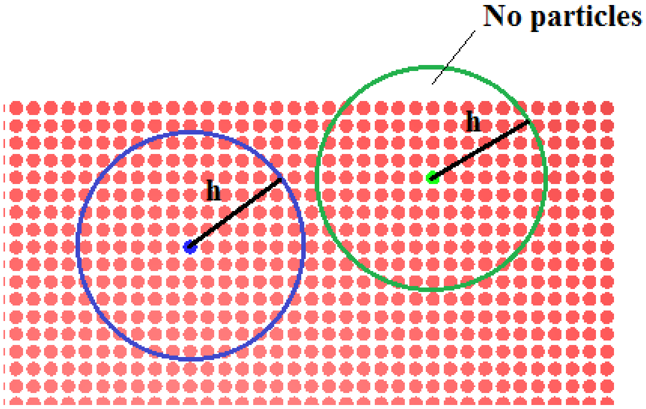


Fig. 1. 2-D schematic of the location of bleeding particles.

where ρ_0 represents static density. It is not related to the magnitude of the pressure since it is a constant. Moreover, ρ_0 makes the simulation result more stable by affecting the gradient of the field obtained in the SPH method.

The density is computed by the SPH algorithm as

$$\rho_i = m\sum_j \frac{\rho_j}{\rho_j} W(r_i - r_j, h) = m\sum_j W(r_i - r_j, h). \quad (20)$$

In terms of the particles near the bleeding surface, both the pressure and the tension are supposed to be considered [39]. Fig. 1 shows a two-dimensional (2-D) diagram of bleeding particles near the bleeding surface and inside the bleeding volume. When the support domain of a bleeding particle is filled with particles, it is considered that this bleeding particle is inside the bleeding volume, otherwise is near the bleeding surface. The blue circle and the green circle represent the support domain of blue particle and green particle, respectively. The blue particle is inside the bleeding volume, and the green particle is near the bleeding surface, which is not full of bleeding particles in its support domain. We adopt the concept of Colorfield [40] to calculate the tension of the bleeding particles. The Colorfield is calculated as follows:

$$C(r_i) = \sum_j m_j \frac{A_j}{\rho_j} W(r_i - r_j, h)$$
$$A_j = \begin{cases} 1 & \text{Particles exist} \\ 0 & \text{No particles} \end{cases} \quad (21)$$

where A_j is used to determine whether there are particles in the support domain of particle i. When there is a particle j in the support domain of particle i, the value of A_j is one, otherwise, the value of A_j is zero. When the particle i is inside the bleeding volume, such as the blue particle shown in Fig. 1, the values of A_j of its surrounding particles are one, and the Colorfield of the surrounding particles offset against each other since it has been assumed that the masses of bleeding particles are equal, so the tension of a particle inside the bleeding volume is zero. If the particle is near the bleeding surface, the Colorfield outside is zero but the Colorfield inside is not, and the tension affects this particle.

The gradient $\vec{n}$ is the normal of the particle surface, which is defined as $\vec{n} = \nabla C$. The surface curvature can be calculated as $\kappa = [(-\nabla^2 C)/(|\ \vec{n}\ |)]$. Then the tension of a particle is defined as

$$f^t = \varepsilon\kappa\vec{n} = -\varepsilon\nabla^2 C\frac{\vec{n}}{|\ \vec{n}\ |} \quad (22)$$

where ε is a constant used to normalize the calculation. The acceleration of particle i generated by tension can be expressed as follows:

$$a_i^t = \frac{f_i^t}{\rho_i} = -\varepsilon\nabla^2 C\frac{\vec{n}}{\rho_i\,|\ \vec{n}\ |}. \quad (23)$$

B. Viscous Force

The acceleration generated by the viscous force of particle i is as follows:

$$a_i^v = \frac{f_i^v}{\rho_i} = \frac{\mu\nabla^2 v(r_i)}{\rho_i} = \mu \cdot \frac{1}{\rho_i}\sum_j m_j \frac{v_j}{\rho_j}\nabla^2 W(r_i - r_j, h). \quad (24)$$

Similar to the calculation of pressure in Section III-A, the viscous force calculated according to (24) does not confirm to the Newton's third law. We assume that all the particles have equal mass m, (24) can be transformed into

$$a_i^v = \mu m\sum_j \frac{v_j - v_i}{\rho_i\rho_j}\nabla^2 W(r_i - r_j, h). \quad (25)$$

Previous work [40] considered that the viscous force was merely determined by the particle velocity, but under the action of platelets, the rate of blood coagulation will increase with time [41], [42]. According to our method, platelet-related parameter P is introduced into the viscosity coefficient μ to simulate the effect of platelets [31], which is defined as

$$\mu = b\exp(-aP_i), a > 0, b > 0 \quad (26)$$

where a and b are constant parameters.

When the size of P_i is reduced with time, the viscosity coefficient μ increases.

The size of P_i over time is computed as follows:

$$\frac{dP}{dt} = k\nabla^2 P$$
$$\nabla^2 P(r_i) = \sum_j m_j \frac{P_j - P_i}{\rho_j}\nabla^2 W(r_i - r_j, h) \quad (27)$$

where k is the transfer parameter, and P_i and P_j are the initial platelet characteristic of particle i and particle j, respectively.

Finally, the accelerations caused by the above-mentioned forces, i.e., pressure, tension, viscous force, and gravity, are summed to solve the total acceleration, as follows:

$$a_i = \frac{f_i^p + f_i^t + f_i^v + f_i^g}{\rho_i}. \quad (28)$$

C. Choice of Kernel Function

The accuracy of the SPH algorithm is mainly determined by appropriate kernel functions, which also have impact on stability and computational efficiency.

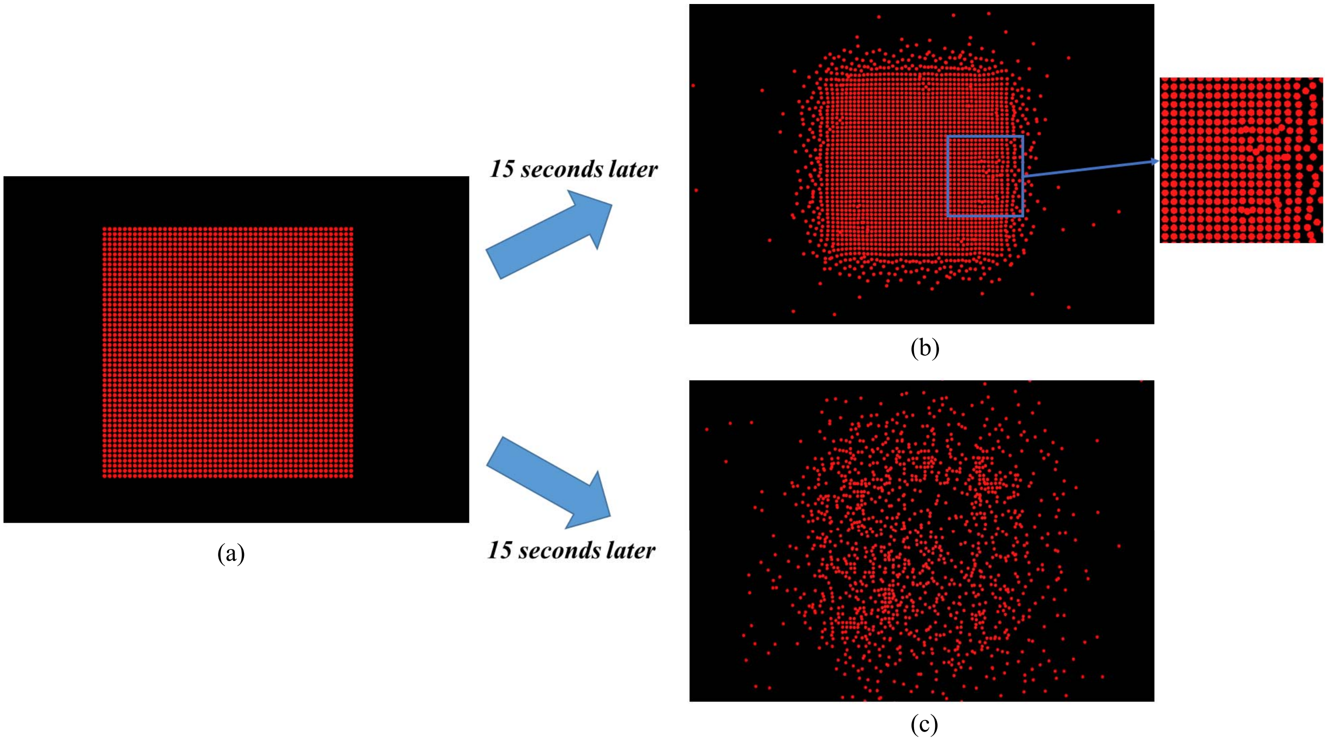


Fig. 2. Influence of the repulsion force on a particle system. (a) Particle system in initial state. (b) Particle system without binding force between particles. (c) Particle system after introducing the repulsive force between particles.

In the computation of density in (20), we use the poly6 as the kernel function [40], which is defined as

$$W_{\text{poly6}}(r, h) = \frac{315}{64\pi h^9}\begin{cases} \left(h^2 - r^2\right)^3 & 0 \leq r \leq h \\ 0 & \text{others.} \end{cases} \tag{29}$$

We choose the poly6 kernel function, and it avoids the calculation of square roots because the power of r is two, which can greatly improve the computational efficiency.

If the poly6 kernel function is applied to compute the pressure in (17), the gradient of the kernel function approaches to zero if the distance between the particles is too close, resulting in an aggregation phenomenon. Therefore, the spiky kernel function [36] is adopted instead when calculating the pressure. The spiky kernel function computes the repulsive force when particles approach to each other, which can avoid aggregation. The spiky kernel function is defined as follows:

$$W_{\text{spiky}}(r, h) = \frac{15}{\pi h^6}\begin{cases} (h - r)^3 & 0 \leq r \leq h \\ 0 & \text{others.} \end{cases} \tag{30}$$

Its gradient is calculated as

$$\text{grad}\big(W_{\text{spiky}}(r, h)\big) = \frac{45}{\pi h^6}\begin{cases} (h - r)^2 \hat{r} & 0 \leq r \leq h \\ 0 & \text{others.} \end{cases} \tag{31}$$

The viscosity kernel function [40] is adopted in the calculation of viscous force in (25). Viscous force impedes the motion of particles and reduces the kinematic velocity of bleeding particles. If the poly6 or the spiky is applied, it may increases the velocity of the particles, which contradicts the characteristic of viscous force since the Laplace operator is negative in a certain range. Therefore, the viscosity kernel function is used in the calculation of viscous force

$$W_{\text{viscosity}}(r, h) = \frac{15}{2\pi h^3}\begin{cases} -\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1 & 0 \leq r \leq h \\ 0 & \text{others.} \end{cases} \tag{32}$$

Its Laplace operator is calculated as

$$\text{Laplacian}\big(W_{\text{viscosity}}(r, h)\big) = \frac{45}{\pi h^6}(h - r). \tag{33}$$

This Laplacian operator is non-negative, and conforms to the characteristics of viscous force.

Since the physical characteristics of the tension is analogous to those of the pressure, we use the spiky kernel function as well for the calculation of tension.

In summary, when calculating the properties of particles, such as density, the poly6 kernel function is used. When computing the interaction forces, the spiky kernel function is used. When computing the viscous force and the platelet property, the viscosity kernel function is used.

The core radius h is considered to be constant in the classical SPH algorithm. However, during the bleeding process, as the distance r between the particles increases, the density ρ of the particles decreases. This leads to the distortion phenomenon that when r is close to h, the pressure and viscous forces between the particles tend to be very close to zero. In our method, we change the core radius h to be a function of the density of particles

$$h_{i+1} = \lambda \times (\frac{m_i}{\rho_i})^{1/d} \tag{34}$$

where λ is a constant, m is the mass of particle i, ρ is the density of particle i, and d is the dimension.

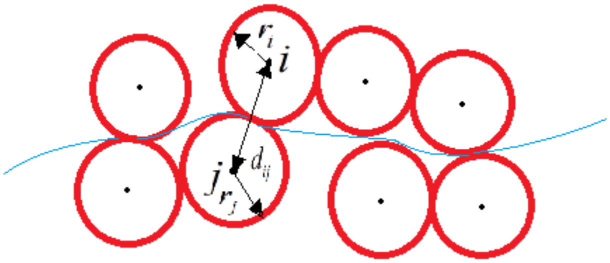


Fig. 3. Contact and penetration between particles.

TABLE I
Algorithm of Bleeding Simulation

```
Generate bleeding particles
Generate cube mesh
for all bleeding particles $i$ do
  for all bleeding particles $j$ in the 27 cube mesh around $i$ do
    if the distance between $i$ and $j$ less than the core radius do
      if $pe$ greater than one do
        Calculate the Repulsive Force
      else do
        Calculate the Pressure and Tension
        Calculate the Viscous Force caused by the platelets
      end if
    end if
  end for
  Compute and assemble distributed force
  Calculate the displacement and the position of the next moment
end for
Render
```

IV. Introducing Repulsive Force Between Particles

Another improvement of our method compared with other work is that we introduce an additional repulsive force between particles to the particle system. In the SPH algorithm, the blood is replaced by a series of particles, which possess bleeding properties and interact with each other within the range controlled by the kernel function. Although solving the N–S equation by the SPH algorithm can greatly improve the computational efficiency and has good accuracy, the N–S equation is used to describe the conservation of momentum of viscous incompressible fluid rather than particle interactions. When using the SPH to solve the N–S equation, we use the spiky to compute the pressure, which can avoid clustering. However, when a few particles move fast, the repulsive force provided by the spiky kernel function is not enough to impede the penetrations between particles, which leads to a phenomenon called *volume distortion* [43]. Another repulsion force should be considered when particles contact or permeate with each other to avoid this phenomenon.

The influence of the repulsion force between particles on the particle system is explained in Fig. 2. The 2-D particle system shown in Fig. 2(a) is influenced by the forces according to the N–S equation, which can be solved by the SPH algorithm. The initial state of the particle system is stationary. The time step size is 0.0005 and the radius of the particles is 3. After 15 s, the particles interact with each other and move. The particles on the edge of the system diffuse, while the internal particles are stationary due to the balanced forces, as shown in Fig. 2(b). When a few high speed particles move inside the particle system, they pass through the particle system and overlap with the internal particles without affecting the motion of them. This phenomenon is volume distortion.

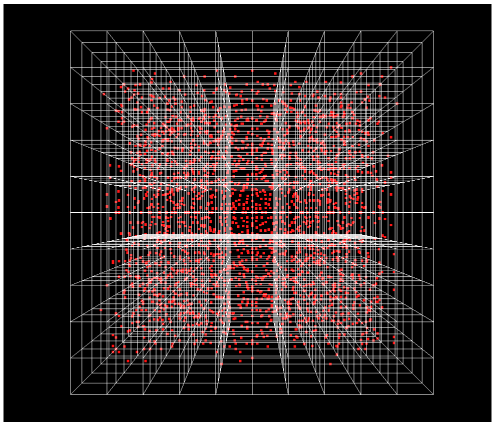


Fig. 4. Generate cube mesh to improve the search efficiency.

In order to solve this problem, another repulsive force between the particles have to be considered. In the SPH algorithm, a particle represents a finite volume in continuum scale. Using each particle to represent an atom or a molecule in nano-scale is quite similar to the classic molecular dynamics method. When the particles interact with each other, a highly peaked repulsive viscous force is applied to the particles that are approaching and tending to penetrate each other. As shown in Fig. 3, the contact and penetration between two particles can be identified by

$$\mathrm{pe} = \frac{(r_i + r_j)}{d_{ij}} \geq 1 \tag{35}$$

where r_i and r_j are the radius of particle i and particle j, respectively, and d_{ij} is the distance between particle i and particle j.

When the contact or penetration between particles is detected, the repulsive force needs to be applied. We choose the penalty force in Lennard–Jones form. Lennard–Jones form is a function for computing the effect of potential energy between two molecules [43]. It is widely used because of its simple analytical form, especially for describing the intermolecular interactions of inert gases.

In our method, the Lennard–Jones form penalty force is applied on the two approaching bleeding particles as follows:

$$f_{ij} = \begin{cases} \bar{D}(\mathrm{pe}^{n_1} - \mathrm{pe}^{n_2})/d_{ij}^2, & \mathrm{pe} \geq 1 \\ 0, & \mathrm{pe} \leq 1 \end{cases} \tag{36}$$

where the parameters $\bar{D}$, n_1, and n_2 are usually taken as 10^5, 6, and 4, respectively. When the repulsive force is applied, after 15 s, the resulting particle system is shown in Fig. 2(c). It can be seen that when the particles collide, the structure of the particle system changes, and the internal particles of the system also move. The volume distortion phenomenon can be avoided.

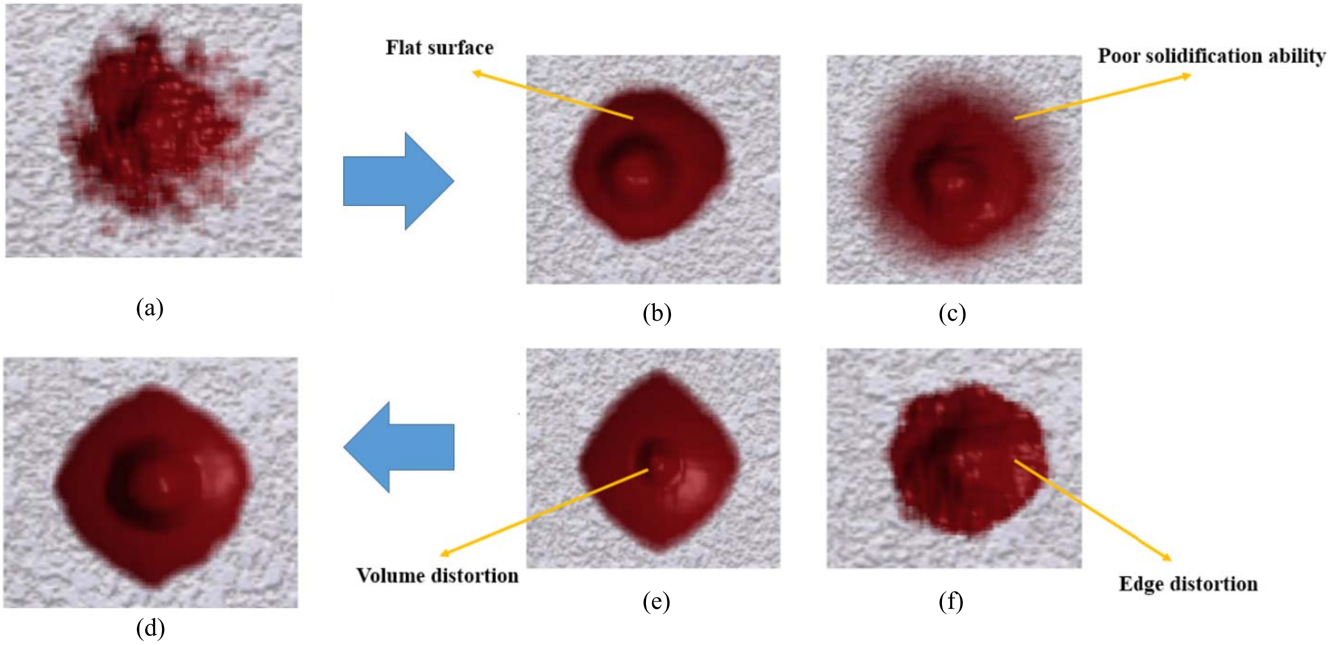


Fig. 5. Comparison of different improvements on the visual effect of bleeding simulation with our algorithm. The number of particles in each figure is 20 000.

V. Bleeding Simulation Results

The bleeding simulation used Visual Studio 2013, OpenGL, and C++ programming.

A. Simulation Algorithm

The procedure includes creating particles, searching for particles in the support domain, calculating forces and displacements, and rendering. The execution of the proposed algorithm is outlined in Table I.

When solving the N–S equation with the SPH algorithm, the search for a particle's neighboring particles in its support domain is the most computationally demanding step. In our algorithm, we divide the particle system into cube mesh at step 2, as shown in Fig. 4, which can substantially reduce the search range and helps to determine whether the particles contact. The mesh size is equal to the core radius of the SPH.

All the particles in the support domain of a certain particle i lie in the 27 cube mesh around this particle. If a particle j is in the support domain of the particle i, it should be judged whether particle j penetrate particle i by pe in (35), and whether the repulsive force needs to be applied.

B. Simulation Results

1) Step-by-Step Improvements: We simulated the bleeding in the case of a round wound, in which the particles gush from the middle and spread around. The simulation was carried out for different improvements, and the effects of each improvement on the simulation results are shown in Fig. 5 (the number of particles is 20 000).

Fig. 5(a) shows a classical physical model of bleeding simulation [11], [12], that is, the SPH method to solve the N–S equation without improvements. This method cannot fully describe the physical properties of bleeding. From the visual effects of the simulation, it can be seen that it is inconsistent with the real bleeding. Each improvement effect of the physical model of the bleeding simulation is shown in Fig. 5(b)–(e).

Tension force is the force acting on the surface of the liquid to reduce the surface area of the liquid. The surface of a diffused liquid will shrink into a sphere under the action of tension force. When the tension force is not incorporated into the bleeding simulation, as shown in Fig. 5(b), the blood diffuses around and does not shrink. Therefore, the surface of the blood is not spherical and the reflected light cannot be seen.

The coagulation speed of blood increases under the effect of platelets. The platelets makes the viscous force of blood increases with time. When the influence of platelets is not considered, as shown in Fig. 5(c), the viscous force of blood is only related to the velocity of the bleeding particles. As the number of bleeding particles increases, the bleeding surface becomes unsmooth and discrete.

The penetration of particles happens when the N–S equation is solved by the SPH method, which leads to volume distortion. In Fig. 5(d), it can be seen that bleeding particles permeate with each other when the repulsion force absent. Consequently, the gush of blood is obviously narrowed at the source of the bleeding. The Fig. 5(d) also shows that the particles around the bleeding source sink down, and the blood volume is visually thinner than that of the real situation.

As shown in Fig. 5(e), the edge of the blood is serrated, and the smoothness of the bleeding surface is poor. If the core radius h of the kernel function is constant, the edge distortion occurs. The reason is that the density ρ of the particles becomes smaller as the distance r between the particles increases during the bleeding process. When r is close to h, the pressure and viscous forces between the particles tend to be very close to zero.

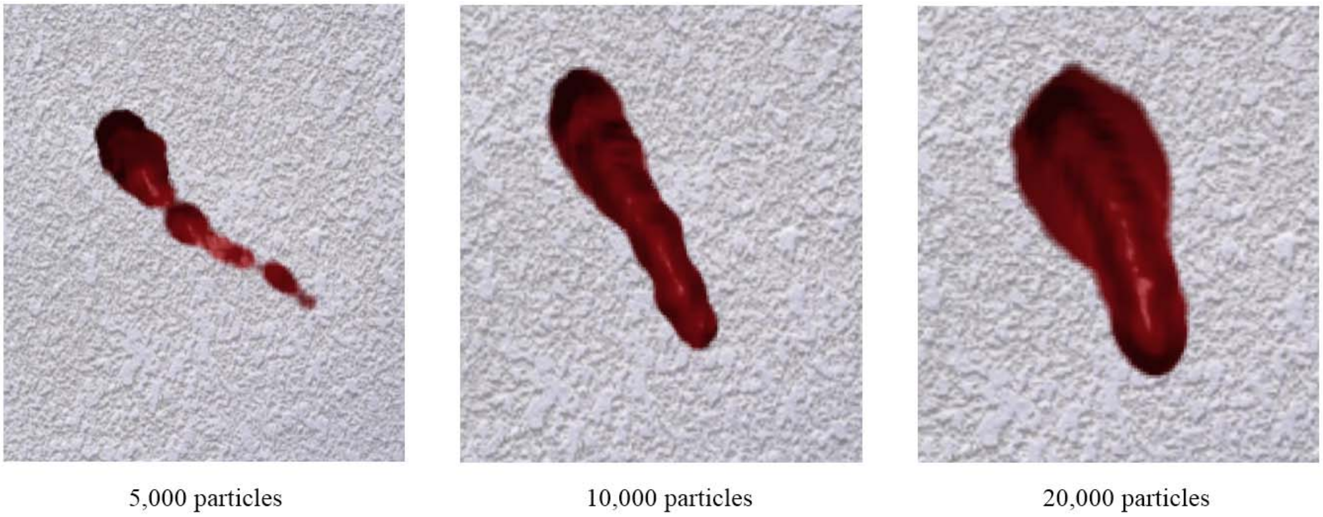


Fig. 6. Simulation of dynamic bleeding process with our improvements.

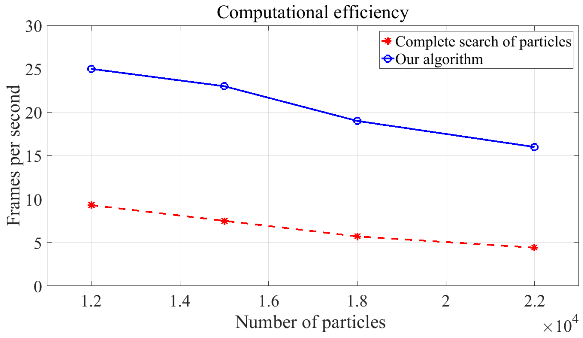


Fig. 7. Computational efficiency of our algorithm.

Bleeding simulation using our method is shown in Fig. 5(f). In comparison with conventional simulation methods, the visual effect of simulation is much improved. Overall, it can be seen from the comparison that the bleeding surface of the simulation using our method is smoother and more realistic according to light reflection because the introduction of tension force and platelets. The gush of blood and the thickness of the blood do not distort under the effect of repulsion force. The problem of edge distortion is also solved.

2) Dynamic Bleeding Process: Bleeding simulation in our method is adjustable in terms of position and shape to match different wounds. Fig. 6 shows the simulation of the dynamic bleeding process. As the number of particles increases, the volume increases, and the blood flows around.

In order to make the visual effects of virtual surgery smooth, the frame per second (FPS) of the simulation needs to be kept above 20. However, the FPS of the bleeding simulation is inversely proportional to the number of bleeding particles, i.e., as the number of bleeding particles increases, the FPS will decrease. Although we did not use GPU acceleration in bleeding simulation, our algorithm can still achieve good computational efficiency. As shown in Table II and Fig. 7, when the number of bleeding particles is 12 000, the FPS can be maintained at 25. Without using the cube mesh to accelerate the search for neighboring particles, the FPS is only 9.3.

TABLE II
RELATIONSHIP BETWEEN THE NUMBER OF BLEEDING PARTICLES AND THE FPS

Number of	**FPS**	
bleeding particles	Complete search of particles	Our algorithm
12,000	9.3	25
15,000	7.5	23
18,000	5.7	19
22,000	4.4	16

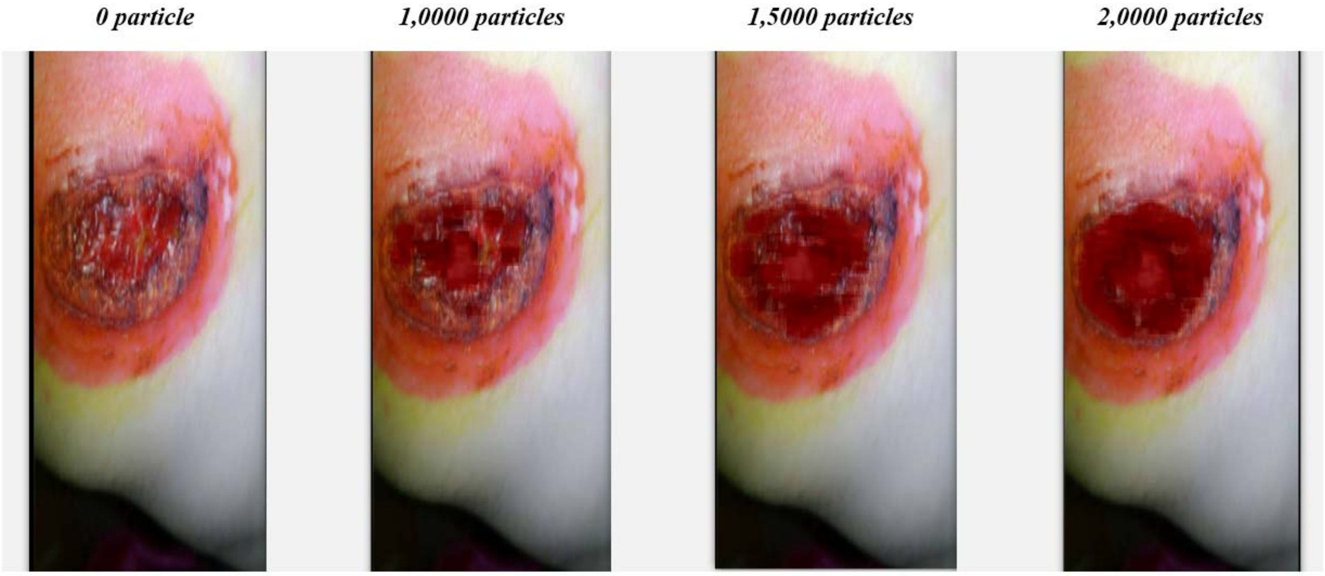
3) Combination With Real Wounds: We applied the bleeding simulation to a real wound as shown in Fig. 8. There was no bleeding on the original wound. As the number of particles increases, the blood exudes from the wound gradually. When the number of particles is 20 000, the range of blood covers the whole wound.

The presented method was also implemented to simulate bleeding phenomena in a surgery simulation system developed by our research group [44], [45], as shown in Fig. 9.

VI. CONCLUSION

In this paper, the problem of bleeding simulation has been dealt with, in particular to improve the visual realism. Specifically, more physical properties of bleeding have been incorporated in order to overcome the existing problems associated with the N–S equation. As mentioned in previous sections, although the N–S equation can be used to perform kinetic analysis for bleeding and the SPH method can give the approximate solution of the N–S equation in a discrete form, the problems associated are as follows.

1) Blood is viscous, incompressible and non-Newtonian fluid so that some of its unique physical properties cannot be fully described by the N–S equation. The bleeding simulation using the traditional N–S equation is anamorphic in the diffusion process and the surface smoothness is poor.
2) When the SPH method is used to solve the N–S equation, constant core radius of the SPH leads to edge distortions in simulating the bleeding area.



0 particle *1,0000 particles* *1,5000 particles* *2,0000 particles*

Fig. 8. Bleeding simulation on a round wound.

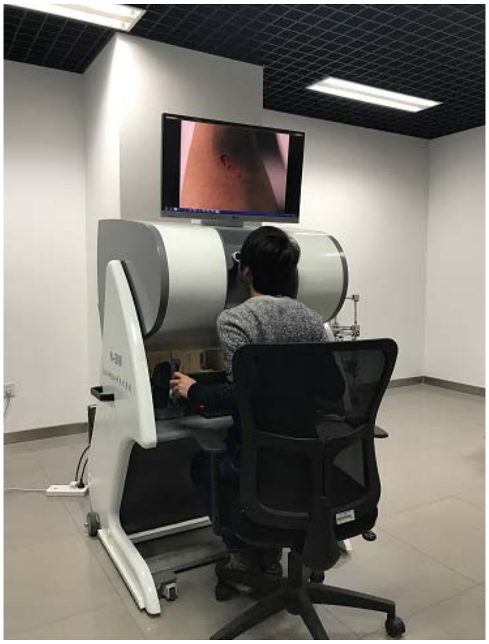


Fig. 9. Neurosurgery simulation system.

3) The repulsive force provided by the traditional SPH method is not enough to prevent penetrations between particles, which lead to volume distortion in simulation.
4) The computational efficiency of simulation is low.

In order to solve these problems, we have taken the following measures.

1) During the process when each of the stress items in the N–S equation is solved by the SPH method, we improved the representation of tension force and introduced the effects of platelets on the viscous force to the calculation.
2) The nature of the kernel function in the SPH method was studied. The constant core radius of the kernel function was changed to be a function of the density of particles, which solved the problem of edge distortion in the bleeding simulation.
3) Additional repulsive force was introduced when the penetration of particles was detected, which solved the problem of volume distortion.
4) The computational efficiency of bleeding simulation was greatly improved by using the search method for neighboring particles based on the cube mesh.

The bleeding simulation with our method can achieve good visual realism and computational efficiency, and can adapt to different shapes of bleeding wounds. In the future, we will explore to further improve the computational efficiency with GPU acceleration.

References

[1] Q. Zhao, "Summary of virtual reality," *Sci. China F Inf. Sci.*, vol. 39, no. 1, pp. 2–46, 2009.

[2] F. Hu *et al.*, "Cyberphysical system with virtual reality for intelligent motion recognition and training," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 2, pp. 347–363, Feb. 2017.

[3] J. Lu, T. Zhang, F. Hu, and Q. Hao, "Preprocessing design in pyroelectric infrared sensor-based human-tracking system: On sensor selection and calibration," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 2, pp. 263–275, Feb. 2017.

[4] Q. Sun, F. Hu, and Q. Hao, "Mobile target scenario recognition via low-cost pyroelectric sensing system: Toward a context-enhanced accurate identification," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 3, pp. 375–384, Mar. 2014.

[5] H. K. Çakmak and U. Kühnapfel, "Animation and simulation techniques for VR-training systems in endoscopic surgery," in *Proc. Eurograph. Workshop Animation Simulat. (EGCAS)*, 2000, pp. 173–185.

[6] C. Basdogan, C. H. Ho, and M. A. Srinivasan, "Simulation of tissue cutting and bleeding for laparoscopic surgery using auxiliary surface," *Stud. Health Technol. Informat.*, vol. 62, no. 62, pp. 38–44, 1999.

[7] M. Kass and G. Miller, "Rapid, stable fluid dynamics for computer graphics," *ACM SIGGRAPH Comput. Graph.*, vol. 24, no. 4, pp. 49–57, 1990.

[8] R. Temam, *Navier–Stokes Equations*, vol. 2. Amsterdam, The Netherlands: North-Holland, 1984.

[9] J. Stam, "Stable fluids," in *Proc. 26th Annu. Conf. Comput. Graph. Interact. Techn.*, 1999, pp. 121–128.

[10] S. Rianto and L. Li, "Fluid dynamic visualisations of cuttings-bleeding for virtual reality heart beating surgery simulation," in *Proc. 33rd Australasian Conf. Comput. Sci.*, vol. 102, 2010, pp. 53–60.

[11] M. Müller, S. Schirm, and M. Teschner, "Interactive blood simulation for virtual surgery based on smoothed particle hydrodynamics," *Technol. Health Care*, vol. 12, no. 1, pp. 25–31, 2004.

[12] J. Qin, W.-M. Pang, B. P. Nguyen, D. Ni, and C.-K. Chui, "Particle-based simulation of blood flow and vessel wall interactions in virtual surgery," in *Proc. 2010 Symp. Inf. Commun. Technol.*, 2010, pp. 128–133.

[13] X. Y. Hu and N. A. Adams, "Angular-momentum conservative smoothed particle dynamics for incompressible viscous flows," *Phys. Fluids*, vol. 18, no. 10, 2006, Art. no. 101702.

[14] M. B. Liu and G. R. Liu, "Smoothed particle hydrodynamics (SPH): An overview and recent developments," *Archives Comput. Methods Eng.*, vol. 17, no. 1, pp. 25–76, 2010.

[15] J. W. Swegle, D. L. Hicks, and S. W. Attaway, "Smoothed particle hydrodynamics stability analysis," *J. Comput. Phys.*, vol. 116, no. 1, pp. 123–134, 1995.

[16] J. J. Monaghan, "Why particle methods work," *SIAM J. Sci. Statist. Comput.*, vol. 3, no. 4, pp. 422–433, 1982.

[17] J. Monaghan, "Particle methods for hydrodynamics," *Comput. Phys. Rep.*, vol. 3, no. 2, pp. 71–124, 1985.

[18] J. J. Monaghan, "Smoothed particle hydrodynamics," *Annu. Rev. Astron. Astrophys.*, vol. 30, no. 1, pp. 543–574, 1992.

[19] M. B. Liu, W. P. Xie, and G. R. Liu, "Modeling incompressible flows using a finite particle method," *Appl. Math. Modell.*, vol. 29, no. 12, pp. 1252–1270, 2005.

[20] M. B. Liu and G. R. Liu, "Restoring particle consistency in smoothed particle hydrodynamics," *Appl. Numer. Math.*, vol. 56, no. 1, pp. 19–36, 2006.

[21] J. Fang and A. Parriaux, "A regularized Lagrangian finite point method for the simulation of incompressible viscous flows," *J. Comput. Phys.*, vol. 227, no. 20, pp. 8894–8908, 2008.

[22] J. Fang, A. Parriaux, M. Rentschler, and C. Ancey, "Improved SPH methods for simulating free surface flows of viscous fluids," *Appl. Numer. Math.*, vol. 59, no. 2, pp. 251–271, 2009.

[23] G. A. Dilts, "Moving-least-squares-particle hydrodynamics—I. Consistency and stability," *Int. J. Numer. Methods Eng.*, vol. 44, no. 8, pp. 1115–1155, 1999.

[24] G. A. Dilts, "Moving least-squares particle hydrodynamics II: Conservation and boundaries," *Int. J. Numer. Methods Eng.*, vol. 48, no. 10, pp. 1503–1524, 2000.

[25] J. Bonet and S. Kulasegaram, "Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations," *Int. J. Numer. Methods Eng.*, vol. 47, no. 6, pp. 1189–1214, 2000.

[26] J.-S. Chen, C. Pan, C.-T. Wu, and W. K. Liu, "Reproducing kernel particle methods for large deformation analysis of non-linear structures," *Comput. Methods Appl. Mech. Eng.*, vol. 139, nos. 1–4, pp. 195–227, 1996.

[27] W. K. Liu *et al.*, "Overview and applications of the reproducing kernel particle methods," *Archives Comput. Methods Eng.*, vol. 3, no. 1, pp. 3–80, 1996.

[28] 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, Nov. 2017.

[29] Y. Zou and P. X. Liu, "A new deformation simulation algorithm for elastic-plastic objects based on splat primitives," *Comput. Biol. Med.*, vol. 83, pp. 84–93, Apr. 2017.

[30] Q. Q. Cheng, P. X. Liu, P. H. Lai, and Y. N. Zou, "An interactive meshless cutting model for nonlinear viscoelastic soft tissue in surgical simulators," *IEEE Access*, vol. 5, pp. 16359–16371, 2017.

[31] W. Shi, M. Zheng, and P. X. Liu, "Virtual surgical bleeding simulation with Navier–Stokes equation and modified smooth particle hydrodynamics method," in *Proc. IEEE Int. Conf. Inf. Autom. (ICIA)*, 2017, pp. 276–281.

[32] R. Temam, *Navier–Stokes Equations*, vol. 2. Amsterdam, The Netherlands: North-Holland, 1984.

[33] R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics: Theory and application to non-spherical stars," *Monthly Notices Roy. Astron. Soc.*, vol. 181, no. 3, pp. 375–389, 1977.

[34] L. B. Lucy, "A numerical approach to the testing of the fission hypothesis," *Astron. J.*, vol. 82, pp. 1013–1024, Dec. 1977.

[35] C. Pozrikidis, *Numerical Computation in Science and Engineering*, vol. 6. New York, NY, USA: Oxford Univ. Press, 1998.

[36] W.-M. Pang *et al.*, "Orthopedics surgery trainer with PPU-accelerated blood and tissue simulation," in *Proc. Med. Image Comput. Comput. Assist. Intervention (MICCAI)*, 2007, pp. 842–849.

[37] S. Clavet, P. Beaudoin, and P. Poulin, "Particle-based viscoelastic fluid simulation," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation*, 2005, pp. 219–228.

[38] M. Desbrun and M.-P. Gascuel, "Smoothed particles: A new paradigm for animating highly deformable bodies," in *Proc. Comput. Animation Simulat.*, 1996, pp. 61–76.

[39] R. Bridson, *Fluid Simulation for Computer Graphics*. Natick, MA, USA: CRC Press, 2015.

[40] M. Müller, D. Charypar, and M. Gross, "Particle-based fluid simulation for interactive applications," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation*, 2003, pp. 154–159.

[41] K. Xu, Y.-S. Xiong, K. Tan, and G.-Y. Guo, "A model of bended bloodstream with small amount bleeding," *J. Nat. Univ. Defense Technol.*, vol. 26, no. 5, pp. 70–73, 2004.

[42] P. Oppenheimer, A. Gupta, S. Weghorst, R. Sweet, and J. Porter, "The representation of blood flow in endourologic surgical simulations," *Stud. Health Technol. Informat.*, vol. 81, pp. 365–371, Jan. 2001.

[43] M. B. Liu, G. R. Liu, K. Y. Lam, and Z. Zong, "Smoothed particle hydrodynamics for numerical simulation of underwater explosion," *Comput. Mech.*, vol. 30, no. 2, pp. 106–118, 2003.

[44] Y. Zou and P. X. Liu, "A high-resolution model for soft tissue deformation based on point primitives," *Comput. Methods Programs Biomed.*, vol. 148, pp. 113–121, Sep. 2017.

[45] Y. Zou, P. X. Liu, C. Yang, C. Li, and Q. Cheng, "Collision detection for virtual environment using particle swarm optimization with adaptive cauchy mutation," *Cluster Comput.*, vol. 20, no. 2, pp. 1765–1774, 2017.

Wen Shi (S'17) received the B.Sc. degree in mechanical engineering and automation and M.Sc. degree in mechatronics from Beijing Jiaotong University, Beijing, China, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree in mechatronics from Beijing Jiaotong University.

His current research interests include surgery simulation and virtual reality.

Peter Xiaoping Liu (SM'07–F'19) received the B.Sc. degree in mechanical engineering and M.Sc. degree in measurement and control from Northern Jiaotong University, Beijing, China, in 1992 and 1995, respectively, and the Ph.D. degree in electrical and computer engineering 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. He is also with the School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing, as an Adjunct Professor. He has published over 300 research articles. His current research interests include interactive networked systems and teleoperation, haptics, surgical simulation, robotics, and system control.

Dr. Liu has served as an Associate Editor for several journals, including the IEEE/ASME TRANSACTIONS ON MECHATRONICS, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and IEEE ACCESS. He is a licensed member of the Professional Engineers of Ontario and a fellow of the Engineering Institute of Canada.

Minhua Zheng (M'15) received the B.Sc. degree in measurement control and information technology from Beihang University, Beijing, China, in 2010 and the Ph.D. degree in electronic engineering from the Chinese University of Hong Kong, Hong Kong, in 2015.

She is currently with the School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing, as a Lecturer. Her current research interests include robotics and intelligent systems, human–robot interaction, social robotics, and virtual surgery systems.