International Journal of Transportation Engineering and Technology
2021; 7(1): 12-18
http://www.sciencepublishinggroup.com/j/ijtet
doi: 10.11648/j.ijtet.20210701.12
ISSN: 2575-1743 (Print); ISSN: 2575-1751 (Online)

Simulated Training System of Ship Anchoring Operation Based on Virtual Reality Technology

Hejun Geng[1], Baowei Wang[2], Deling Wang[1], Xiaobin Jiang[1, *]

[1]Merchant Marine College, Shanghai Maritime University, Shanghai, China

[2]Shanghai Lanbin Petrochemical Equipment Co., Ltd., Shanghai, China

Email address:

genghj@shmtu.edu.cn (Hejun Geng), dlwang@shmtu.edu.cn (Deling Wang), xbjiang@shmtu.edu.cn (Xiaobin Jiang)

[*]Corresponding author

To cite this article:

Hejun Geng, Baowei Wang, Deling Wang, Xiaobin Jiang. Simulated Training System of Ship Anchoring Operation Based on Virtual Reality Technology. *International Journal of Transportation Engineering and Technology*. Vol. 7, No. 1, 2021, pp. 12-18.
doi: 10.11648/j.ijtet.20210701.12

Received: March 6, 2021; **Accepted**: March 17, 2021; **Published**: March 26, 2021

Abstract: Seldom can trainees be allowed to practice anchoring in current teaching and training for crew. It is quite dangerous, difficult and costly to train and teach the trainees of anchoring operation by using real anchoring equipment. The practical operation teaching and training of ship mooring maneuvers is. In order to improve the training efficiency and save the training cost, an idea of anchoring simulation based on the virtual reality technology has emerged. According to our research on the problems existing in the research and development of the simulation training system for ship anchoring operation. The 3D modeling technology is used to build 3D models of anchoring equipment such as windlass and anchor chains, and the 3D visualization and interactive operation of virtual scene of ships are realized by using Unity 3D engine design. Use inverse dynamics method for virtual human action simulation and A* path finding algorithm for path planning is proposed in the paper, as well as a position-based dynamics (PBD) method to solve the problems of low simulation efficiency and fidelity in the simulation of anchor chains. Besides, a bounding box technology and a continuous collision detection method are employed, respectively, to detect collisions between different objects in a scene. Through the simulation system to carry out practical training for crew, the simulation system can meet the training requirements of ship anchoring.

Keywords: Virtual Reality, Ship Anchoring Operation, Position-Based Dynamics, Visualization Simulation

1. Introduction

Anchoring is the most common way to berth ships. Whether to grasp the correct use and maintenance methods of anchoring equipment is directly correlated with the operation safety of ships, and they are the primary professional knowledge and operating skills for the crew of deck departments [1]. Generally, shipping companies do not have real windlasses for teaching and training for crew, restricting crew to operate them proficiently. During actual ship anchoring, accidents (such as anchor loss, chain breaking, and windlass burnout) are often caused by crew's improper operation. Therefore, an immersive and interactive ship anchoring simulation system developed with an advanced virtual reality technology can be used to train crew, help them grasp the process of anchoring operation, and enhance their work efficiency and ability to handle emergencies, so as to ensure the safety of life, property, and environment.

The Manila 2010 Amendment 2A-/1 of the STCW Convention (International Convention on Standards of Training, Certification and Watch keeping for Seafarers) proposed the standard requirements on the correct procedures for anchoring operation [2]. It is necessary to more precisely simulate anchor chains to make the anchoring operation simulation system closer to the real environment and fulfil the requirement of teaching and training as much as possible. Only based on the more optimized three-dimensional visualization effect of the simulation can this be achieved. However, there is little research on ship anchoring operation simulation from the aspect of visualization, especially on the visualization of anchor chains during ship anchoring.

For research on visual simulation of ship handling, please refer to the literature [3-5]. An anchor chain can be simulated

as a flexible object. However, it is not an ordinary rope. Thus, any misoperation during anchoring may cause casualties. Researchers worldwide have investigated the simulation of flexible ropes and proposed the mass point-spring model [6], Crossrat rods method [7], and PBD method [8]. Owing to simplicity, rapidity, controllability, efficiency, and robustness, the PBD method has been widely used in the simulation of ropes, cloth, deformable solids, and fluids in the field of computer graphics [9]. Considering the characteristics of anchor chains in the ship anchoring operation simulation system, the application of the PBD method in it is proposed in this paper..

2. Overall Design

To improve the realism of the simulation system, we designed the system modules from three dimensions (support technology, development tools and display application) based on the actual demand of a training system and provided the technical support for the entire system through the research and development of underlying technologies (such as the virtual human technology, particle system, physics engine, and network technology). Then, three-dimensional models of the bridge, ship windlass, and anchor equipment were established using 3dsMax modelling software, with a Panamax bulk carrier "Changshanhai" as a mother ship. The related functions of the windlass were simulated using a Unity 3D engine, and a three-dimensional virtual scene of the ship was constructed [10]. Through ship anchoring operation training in this scene, the trainees can understand the attributes of anchoring equipment and relevant operating requirements. The system structure is illustrated in Figure 1.

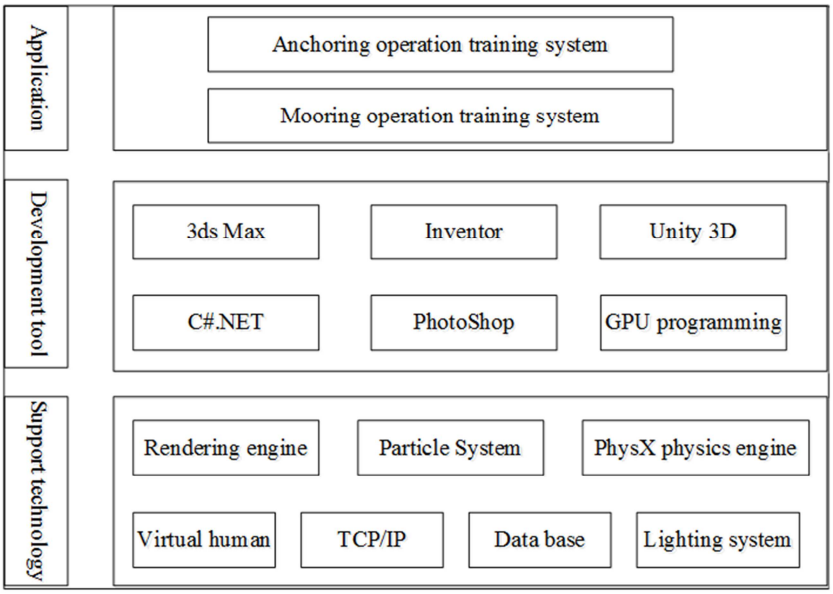


Figure 1. *Overall architecture of the system.*

3. Core Technology

3.1. Virtual Human Technology

The virtual human technology is a comprehensive application of artificial intelligence technology and computer graphics. In this paper, the geometric model of a virtual person was established using bone skinning technology. Besides, the virtual human animation was created by rotating the joints of human bones to preset angle values. This method of determining the positions of bones is called forward dynamics. Inverse dynamics contributes to improving the accuracy of operating positions and the calculation efficiency of virtual human's actions. A virtual human's posture is expressed by an angle θ between the joints and the position x of the endmost node of a bone, expressed as [11]:

$$\theta = f^{-1}(x) \tag{1}$$

The simulation effect of some virtual human actions based on reverse dynamics is presented in Figure 2.

Figure 2. *Virtual human actions.*

In addition to realizing the basic actions of a virtual person, controlling it to reach a preset place is also of equal importance, requiring us to plan its walking path. Considering that an A* algorithm in a heuristic search algorithm is the fastest one to calculate the shortest path by far, it was applied to plan the walking paths of the virtual person. The core of the

A* algorithm is an evaluation function [12].

$$f(n) = g(n) + h(n) \tag{2}$$

Where, $f(n)$ represents the evaluation function from the initial node s to the target one t; $g(n)$ denotes the actual cost from the initial node s to the intermediate one n; $h(n)$ indicates the estimated cost of the shortest path from n to t.

The automatic pathfinding process of the visual person after the A* algorithm was applied in the system is exhibited in Figure 3, where the virtual person walking towards the direction is marked with the red line.

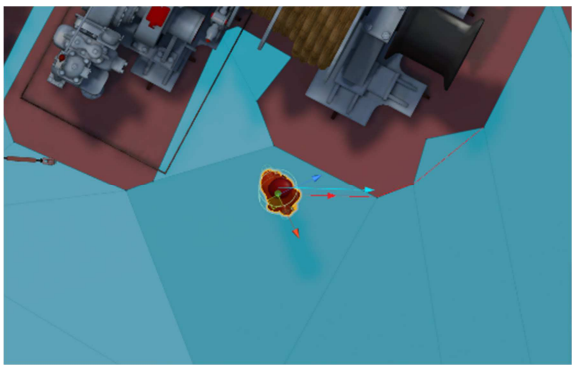


Figure 3. *Virtual human walking along the path.*

3.2. Position-based Dynamics

In the PBD method, an anchor chain was represented by particles coupled with constraints, each of which was iteratively solved in each time step. The simulation and iterative calculation of an anchor chain in the system can be divided into three steps: 1) moving anchor chain particles according to speeds and external forces; 2) moving an anchor chain to meet constraints; 3) performing time integral.

The anchor chain particles were represented by N particles and M constraints. Each particle $i \in [1,..., N]$ consisted of three attributes, including the mass mi, position x_i, and velocity v_i. Besides, each constraint $j \in [1, \ldots, M]$ contained 5 attributes.

(1) Base: n_j,

(2) Scalar constraint function: C_j: $\mathbb{R}^{3n_j} \to \mathbb{R}$,

(3) Index set: $\{i_1, \cdots, i_{nj}\}, i_k \in \{1, \cdots, N\}$,

(4) Stiffness parameter: $k \in \{0 \cdots 1\}$,

(5) Constraint type: $\begin{cases} Unilateral: & C_j(x_{i_1}, \cdots, x_{i_{nj}}) \geq 0 \\ Bilateral: & C_j(x_{i_1}, \cdots, x_{i_{nj}}) = 0 \end{cases}$

The dynamic object simulation process based on the given data and time step Δt is illustrated in Table 1. The position and velocity of an anchor chain were specified in steps (1)-(3). Particle velocity and position were calculated in steps (5)-(6) by applying forward Euler integral to the constraints imposed on the anchor chain particles. The collision constraints were calculated before the iterative calculation (7). The predicted position pi was corrected in steps (8)-(10) to ensure that the particles can meet the M_{coll} external and M internal constraints. Finally, the new position pi and velocity vi were calculated.

Table 1. *Position-based dynamics.*

(1) for all vertices i do
(2) initialize $x_i = x_i^0, v_i = v_i^0, w_i = 1/m_i$
(3) end for
(4) loop
(5) for all vertices i do $v_i \leftarrow v_i + \Delta t w_i f_{ext}(x_i)$
(6) for all vertices i do $p_i \leftarrow x_i + \Delta t v_i$
(7) for all vertices i do generate Collision Constraints ($x_i \to p_i$)
(8) loop solver Iterations times
(9) project Constraints ($C_i, \cdots, C_{M+MColl}, p_i, \cdots, p_N$)
(10) end loop
(11) for all vertices i do
(12) $v_i \leftarrow (p_i - x_i)/\Delta t$
(13) $v_i \leftarrow p_i$
(14) end for
(15) velocity Update ($v_i, \cdots, v_N$)
(16) end loop

In the steps (8)-(10) of the Algorithm 1, the predicted position p_i of the anchor chain particles was corrected to solve M constraint equations and $3N$ unknown position components, enabling the anchor chain particles to meet all the constraints. P was used to represent $\left[p_1^T, \cdots, p_N^T\right]^T$ and the input condition of all constraint functions C_j. The obtained equation to be solved was [9]:

$$\begin{aligned} &C_1(p) = 0 \\ &\cdots \\ &C_M(p) = 0 \end{aligned} \tag{3}$$

A nonlinear Gauss-Seidel iteration method was employed to solve the constraint equation and calculate the estimated positions of the anchor chain particles. Additionally, each constraint function was linearized using the PBD. Based on the separation solving of each constraint equation, the equation generated between each constraint and the positions of all related particles was $C(p)$=0. For the given p, a particle position correction value Δp satisfying the constraint conditions was obtained.

$$C(p + \Delta p) = 0 \tag{4}$$

The above equation was expanded in a Taylor series way

$$C(p + \Delta p) = C(p) + \nabla_p C(p) \cdot \Delta p + O(|\Delta p|)^2 = 0 \tag{5}$$

Higher-order terms were omitted to obtain

$$C(p) + \nabla_p C(p) \cdot \Delta p \approx 0 \tag{6}$$

The point p moved according to a certain movement control equation, and Δp moved according to a constraint C different from the movement direction. Consequently, the movement of Δp should not hinder that of p, that is, the directions of original translation and rotation should be maintained. Therefore, a gradient direction $\nabla pC(p)$, which was perpendicular to the original direction, was selected as the

movement direction of Δp. In the gradient direction, the positions of the particles had the most prominent changes, and they could be stable faster:

$$\Delta p = \lambda \nabla_p C(p) \tag{7}$$

A Lagrangian multiplier λ can be obtained by substituting equation (7) into equation (6).

$$\lambda = -\frac{C(p)}{\left|\nabla_p C(p)\right|^2} \tag{8}$$

The multiplier was substituted into equation (7) to obtain the final equation of the correction vector Δp of a single particle i.

$$\Delta p_i = \lambda w_i \nabla_{pi} C(p) \tag{9}$$

Where, the Lagrangian multiplier $\lambda = -\frac{C(p)}{\sum_j w_j \left|\nabla_{pj} C(p)\right|^2}$ was the same for all particles. If these points had separate masses, the inverse wi=1/mi of them would be corrected by proportional weighting, and the Newton-Raphson calculation process would be extended. After Δp was obtained, the current position was updated to p←p+Δp. Then, a new linear system was generated by evaluating ▽pCj(p) and -Cj(p) in the new position. Afterwards, the process was repeated.

3.3. Collision Detection

Collision detection and response must be considered in the visual simulation. Given the expected effect of the anchor chain movement, the positions of the simulation objects were directly manipulated by projecting the points to the effective positions in the simulation process. This can effectively avoid collisions between the anchor chain and the ship's side, the seabed or itself. In the simulation system, a continuous collision detection method is used to address the collision constraints of anchor chain particles in a scene. In line (7) of Algorithm 1, an additional collision constraint McColl is generated in each time step of the iterative calculation, and the number of collision vertices determines the number of the McColl. In the simulation system, the ray $x_i \rightarrow p_i$ of each vertex, the surface point q_s closest to pi, and the surface normal ns at that position were calculated; the constraint function $C(p)=(p-q_c)\bullet n_c$ was added to the constraint list. The velocity direction of each vertex generating the collision constraint was perpendicular to the collision normal and reflected in its direction. The anchor chain particles could collide with other objects or particles in the scene. These two types of collisions were calculated separately in the system.

Collisions between particles and rigid bodies

Collisions between particles and objects such as decks or seabed can be detected by first determining their contact surfaces and then calculating the normal n of them each. The non-penetration constraints used in the simulation system are [13]

$$C(p) = \mathrm{n}^T p - d_{rest} = 0 \tag{10}$$

Where, d_{rest} represents the distance maintained by the anchor chain particles when they were stationary. The physical properties of the particles such as friction, viscosity, and radiuses were considered in this study to improve the realism of the simulation system. Various collision responses were generated according to the physical characteristics of particles and collided objects.

The collision effect of the material of the anchor chain in consideration is exhibited in Figure 4, where Arrow 1 denotes the velocity of the anchor chain particles during the collision. The particle velocity was divided into two components, along the normal and tangent of the contact surface, respectively. Arrow 2 indicates the friction force exerted on the particles, which eliminates part of the tangential velocity component. Arrow 3 represents the force exerted by the contact surface on the anchor chain particles, and the force offsets their normal velocity. Arrow 4 stands for the viscous force.

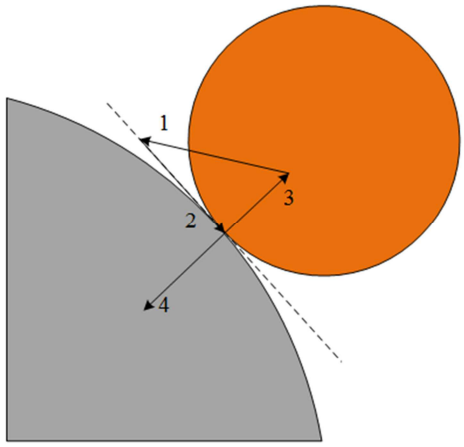


Figure 4. *Collision between particle and rigid body.*

In the calculation of the collision constraint between the anchor chain and rigid body, the thickness of the anchor chain particles was considered to obtain a more realistic simulation effect. The constraint equation is [14]

$$C(q, p_1, p_2, p_3) = (q - p_1) \cdot n - h \tag{11}$$

Where, h is the thickness of the anchor chain, n denotes the normal direction of the triangles p1, p2, and p3. According to Figure 5, the anchor chain particle q keeps above the triangles p1, p2, and p3 at a thickness of h. Considering that different values of h could produce different distances between the anchor chain particles and the contact surfaces, more realistic detention on the collisions between the anchor chain and the rigid body could be achieved by adjusting the thickness of the particles. Figure 6 illustrates the effect of collisions between particles and cubes regardless of particle thickness and particles embedded in the cubes. The thicker the anchor chain particles, the deeper they embedded in the rigid body. The thickness values of the particles from left to right in the figure are 0.08, 0 and -0.05, respectively.

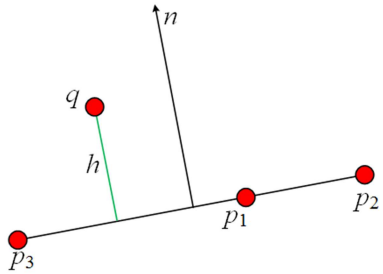
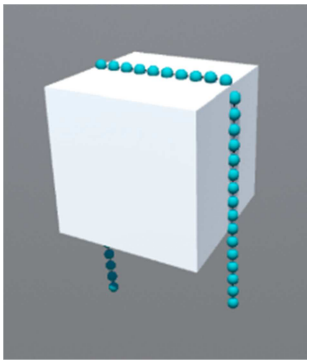
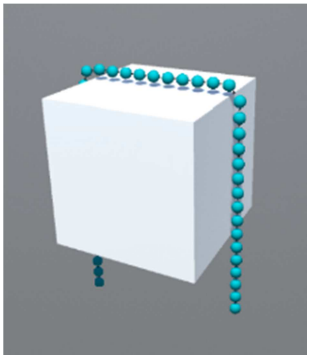


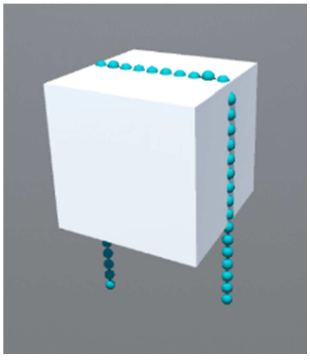
Figure 5. *Collision constraints under the consideration of particle thickness.*



(a) Chain radius=0.08



b) Chain radius=0



(c) Chain radius=-0.05

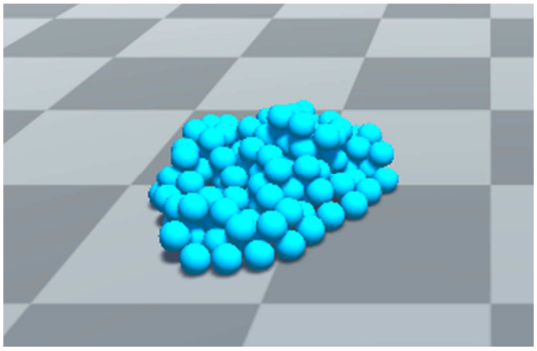
Figure 6. *Collision effect between anchor chain particles and the cubes.*

Collisions between particles

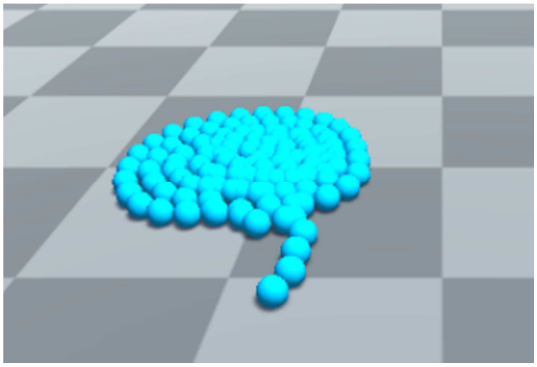
Collisions between particles can be handled by linearizing particles and introducing them into a contact plane, similar to the way of addressing the collisions with the rigid bodies. When the nonlinear characteristics of the collision constraint are maintained, the collision constraint between particles is [15]

$$C(p_i, p_j) = | p_i - p_j | -(r_i - r_j) \geq 0 \tag{12}$$

Where, r_i and r_j represent the radii of the two particles, respectively. The effect of collision detection with or without considering collisions between the particles in the case of the same number of particles is presented in Figure 7. Generally, the detection under the consideration of collisions between particles must be conducted during the simulation of an anchor chain to prevent particles from passing through each other when they are stacked. The detection with considering collisions between particles is shown in Figure 7(a). However, in the case of no stack of anchor chain particles, calculating the collisions between particles will occupy a lot of computer capacity, making the calculation of collisions between particles in this case superfluous. The detection without considering collisions between particles is illustrated in Figure 7(b).



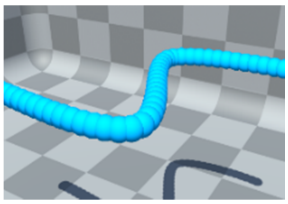
(a) Considering collisions between particles



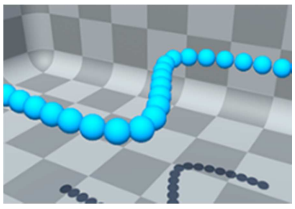
(b) Without considering collisions between particles

Figure 7. *Detection on collisions between particles.*

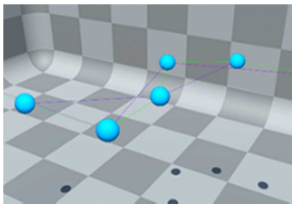
The number of particles is intimately related to the effect of collision detection. In the simulation system, the closeness of an anchor chain can be solved by adjusting the number of particles. The particle spacing was equal to its radius when the relative value of the number of particles was 1. The overlap of the anchor chain particles is exhibited in Figure 8(a). Besides, they just contacted with each other when the relative value of the number of particles was 0.5, as indicated in Figure 8(b). Specifically, the smaller the relative value of the number of particles, the longer the distance between particles, as demonstrated in Figure 8(c).



a) particles number=1



(b) particles number=0.5



(c) particles number=0.1

Figure 8. *Detection on inter-particle collisions in the case of the different numbers of anchor chain particles.*

Collisions between rigid bodies

During virtual roaming interaction, it is necessary to detect possible collisions between moving rigid bodies in a scene. If detection is directly performed on collisions between two objects, the calculation cost is often higher. The consumption of computational capacity will be reduced by calculating geometric intersection after using a collider to detect collisions between collision bodies. The basic collision detection technologies for virtual scenes mainly include axial bounding box (AABB) detection, directional bounding box (OBB) detection, bounding ball detection, and discrete directional polyhedron (k-DOP) detection (Figure 9). For detection on collisions between rigid bodies, five types of colliders provided by the Unity engine were primarily used, including box colliders, ball colliders, capsule colliders, mesh colliders, and role control colliders [16].

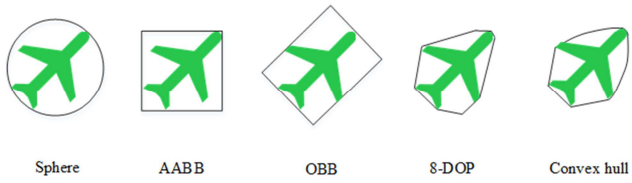


Figure 9. *Bounding box type.*

Figure 10. *Visual effect of the anchoring operation.*

4. Implementation of the System

All the test programs described in this paper ran on Windows 10 and were developed with Unity3d and Visual Studio 2019 software. The hardware environment included an Intel(R) Core i7 CPU and a GT 730 graphics card. The time step provided in the anchoring operation simulation was 2ms, and the operating speed of the system was about 60 fps, satisfying its requirement of real-timing.

The visual effect of the anchoring operation of a ship is shown in Figure 10, in which a, b, and c indicate the full housing, partial drop, and full drop of the anchor chain, respectively; d, e, and f denote the various states of the anchor chain from the beginning of the drop; g, h, and I represent the final effect of the anchoring operation according to the ship's movement. The detection on collisions between the anchor chain and the seabed and the collisions between the chains in the stack were considered in the simulation system.

5. Conclusion

In this paper, a simulation training system for ship anchoring operation was established to realize the real-time simulation of the process. Based on the virtual human technology, the virtual human's actions and the pathfinding functions were simulated, the real-time simulation of the anchor chain was realized using the position-based dynamics method, and the collision detection methods corresponding to the different objects in the scene were established. Thus, the problem of penetration between the anchor chain and the deck, ship's side and dock in the scene was solved. The simulation system satisfies the ship anchoring operation's requirement of real-timing. The visual effect of simulation is more realistic and reliable, providing users with a strong sense of immersion. Moreover, applying the established simulation system of anchoring operation to training can not only significantly save costs but also protect personnel safety and lower risks effectively. Furthermore, relevant models and algorithms have been integrated into a 3D simulation platform, contributing to guiding the simulation of ship anchoring operation and the evaluation of crew training. Subsequent research can be carried out for the following aspects: multiple computers are networked to achieve multi-person collaborative training; design and implementation of the automatic evaluation function of the simulation system.

References

[1] Li Wei. Ship Structure and Equipment. in *Dalian: Dalian Maritime University Press,* 2008.

[2] Dong Fang, Mao Hongxin. Interpretation of the Manila Amendments to STCW Convention. in *China Maritime Safety*, 2012, (5), pp. 6-10.

[3] Tao Rui. Zhu Yaohui, Ren Hongxiang, *et al*. Ship fire-fighting training system based on virtual reality technique. in *Journal of Shanghai Maritime University,* 2017, 38 (1), pp. 73-78.

[4] Jiang Xiaobin, Ren Hongxiang, Wang Mingyang, *et al*. Research on cable visualization in ship mooring operation simulation training system. in *J. Huazhong Univ of Sci& Tech. (Natural Science Edition)*. 2019 47 (10), pp. 99-104.

[5] QIU Shaoyang, Ren Hongxiang, Yin Jingang. A free fall lifeboat training system based on virtual reality technology. in *Journal of Dalian Maritime University,* 2017. 43 (3), pp. 14-18.

[6] Liu T, Bargteil A W, O"Brien J F, et al. Fast simulation of mass-spring systems. in *ACM Transactions on Graphics,* 2013, 32 (6), pp. 1-7.

[7] Kugelstadt T, Schömer, Elmar. Position and Orientation Based Cosserat Rods. in *Acm Siggraph/eurographics Symposium on Computer Animation. Eurographics Association,* 2016: 169–178.

[8] Müller M, Heidelberger B, Hennix M, et al. Position based dynamics. in *Journal of Visual Communication & Image Representation,* 2007, 18 (2), pp. 109-118.

[9] Bender, Jan, Müller, Matthias, Macklin Miles, et al. A Survey on Position Based Dynamics. in *Tutorial Proceedings of Eurographics,* 2017.

[10] Guo Chen, Wu Heng, Shi Cheng-jun. *et al*. Overall design and system structure of virtual reality technigue based marine power plant simulator. in *Journal of Dalian Maritime University,* 1999, 25 (2), pp. 67-71.

[11] Jiang Xiaobin, Tan jiawan, Ren Hongxiang. *et al.* Interactive Simulation System of Marine Windlass Based on Virtual Reality Technology. in *Journal of Chongqing Jiaotong University* (*Nature Science*)*,* 2019, 38 (06), pp. 16-20.

[12] He xin, Ren Hongxiang. Construction of Navigation Mesh Based on Improved Triangulation Algorithm. in *Computer Simulation,* 2019, 036 (010), pp. 373-377.

[13] M Müller. Detailed Rigid Body Simulation using Extended Position Based Dynamics. In *Symposium on Computer Animation.* 2020, 39 (8), pp. 101-112

[14] GALVEZ, Javier, et al. A nonsmooth frictional contact formulation for multibody system dynamics. in *International Journal for Numerical Methods in Engineering,* 2020, 121 (16), pp. 3584-3609.

[15] Macklin M, Chentanez N, Kim T Y. Unified particle physics for real-time applications. in *Acm Transactions on Graphics,* 2014, 33 (4), pp. 1-12.

[16] Jiang Dezhi, Yao Wenlong, Zhang Jundong. Development of engine room resource management simulator based on Unity 3D. in *Navigation of China*, 2015, 38 (3), pp. 13-17.