Letter

Multi-Joint Active Collision Avoidance for Robot Based on Depth Visual Perception

Hui Li, Xingfang Wang, Xiao Huang, Yifan Ma, and Zhihong Jiang

Dear editor,

Human-robot collaboration is a research topic that has numerous potential applications, such as in smart cities. An important safety consideration in human-robot collaboration is collision avoidance [1]. Many studies have prioritized collision avoidance of the robot end-effector. However, multi-joint (whole-body) collision avoidance is also very important in complex working scenarios. Several studies have achieved multi-joint collision avoidance by estimating the distance between obstacles and control points placed on the robot body, which adds computational effort to the collision-detection process. Therefore, we combine a robot skeleton with depth visual perception to achieve a fast multi-joint collision avoidance response. This method can be divided into collision detection and collision avoidance strategies, as illustrated in Fig. 1. In collision detection, we use a statistical filtering algorithm to denoise visual data, and propose a real-time obstacle distance-estimation algorithm based on a robot skeleton. In the collision avoidance strategy, a multi-joint repulsive force model is established based on the artificial potential field, and a motion control strategy is developed to obtain a collision-free trajectory. Real-time collision avoidance experiments with a 7-DOF collaborative robot are conducted to verify our approach. The experimental results show that our method enables the robot to have the capability of active collision avoidance for both the end-effector and the whole body. The robot can avoid obstacles and complete desired tasks with a smooth trajectory.

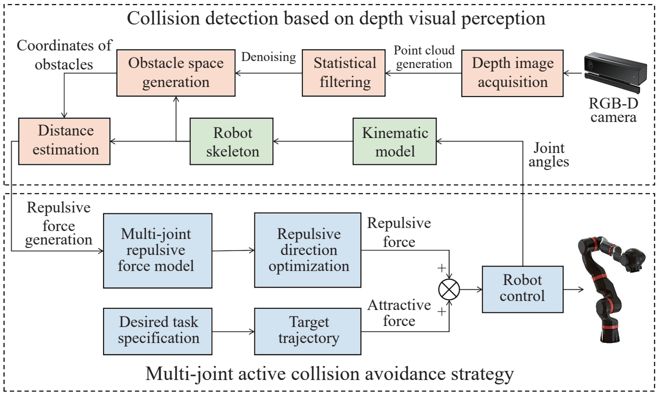
Collision detection based on depth visual perception
Coordinates of obstacles
Obstacle space generation
Denoising
Statistical filtering
Point cloud generation
Depth image acquisition
RGB-D camera
Distance estimation
Robot skeleton
Kinematic model
Joint angles
Repulsive force generation
Multi-joint repulsive force model
Repulsive direction optimization
Repulsive force
+
+
Robot control
Desired task specification
Target trajectory
Attractive force
Multi-joint active collision avoidance strategy

Fig. 1. The system architecture of multi-joint active collision avoidance.

Related work: Collision detection is typically studied as a three-dimensional (3D) visual perception problem [2]. Distance estimation between a robot and obstacles based on visual data is the key to collision detection. In [3]–[5], a 3D depth camera (Microsoft Kinect) was used to obtain the depth data of obstacles and estimate the distance between the obstacles and all control points set on the robot. Based on this distance estimation method, in [6], the mapping relationship between the depth space and Cartesian space was further analyzed, and the data processing was optimized. In [7], the point cloud was used to generate convex hulls of the robot and obstacles; the distance between convex hulls was then estimated to realize collision detection. However, the accuracy of collision detection is affected by visual noise. In [7], the noise was eliminated based on changes in the depth values between two frames. In [4], Kalman filtering was applied for data denoising of the nearest obstacle point.

The Collision avoidance strategy is the application of collision detection results in motion control, which directly determines the collision avoidance ability of the robot. These strategies can be divided into robot end-effector collision avoidance, and multi-joint collision avoidance. Collision avoidance of the end-effector equipped with operating tools was explored in [8], in which the distance from obstacles to the robot was converted into a repulsive force through the artificial potential field, and a collision-free trajectory was generated. Considering that the task priority of the end-effector was higher than that of other joints, the repulsive force was only applied to the end-effector [3]. To achieve multi-joint collision avoidance, other joints were constrained by joint velocities based on theory [9]. In [5], a similar approach was adopted in which Cartesian constraints were applied to other joints.

However, robots have various structural size requirements and work environments in different task scenarios. When the robot structure is larger or the robot is mounted on a platform, the robot body is more likely to collide with dynamic obstacles than the end-effector. Therefore, in the face of unknown dynamic obstacles, the robot body, including all joints, needs to have more active collision avoidance capabilities, rather than applying joint constraints.

Collision detection based on depth visual perception: The accuracy and real-time nature of collision detection are the key factors affecting the reliability of collision avoidance motion. Therefore, according to the noise characteristics of the RGB-D camera (Kinect V2), a statistical filter is used to suppress noise to improve the robustness of the point cloud. Distance estimation based on the robot skeleton can obtain the nearest distance from the obstacles to the robot through a rapid response.

1) Depth image acquisition: Depth images are collected by Kinect V2, which can contain depth data in the range of 0.5−4.5 m in front of the camera [10]. The mapping relationship between the depth image coordinate system and camera coordinate system is given by

$$\kappa = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}, \begin{cases} x_c = \dfrac{(p_x - c_x)d_p}{f_x} \\ y_c = \dfrac{(p_y - c_y)d_p}{f_y} \\ z_c = d_p \end{cases} \tag{1}$$

where κ denotes the internal parameter matrix of the camera; f_x, f_y, c_x, and c_y are the internal parameters accordingly. (x_c, y_c, z_c) are the 3D coordinates of a point in the camera coordinate system; (p_x, p_y) are the corresponding pixel coordinates on the depth image; and d_p is the depth value of this point. The depth image is then used to generate the point cloud by the above mapping relationship.

2) Statistical filtering: To improve the computing efficiency of the point cloud data, the data amount of the point cloud is reduced through voxel grid downsampling, as shown in Fig. 2(b).

Through practical experiments, it is found that the depth error increases with a sudden change in the object distance. The noise distribution at the edges of the objects is shown in Fig. 2(b). Thus, a statistical filtering algorithm is used to eliminate sparse noise and outliers in the global point cloud. Assuming that the point-cloud distri-

Corresponding author: Xiao Huang and Zhihong Jiang.

Citation: H. Li, X. F. Wang, X. Huang, Y. F. Ma, and Z. H. Jiang, “Multi-joint active collision avoidance for robot based on depth visual perception,” *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 12, pp. 2186–2189, Dec. 2022.

The authors are with the School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China (e-mail: lihui2011@bit.edu.cn; xingfangw@126.com; 7520200120@bit.edu.cn; 3120200159@bit.edu.cn; jiangzhihong@bit.edu.cn).

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

Digital Object Identifier 10.1109/JAS.2022.105674

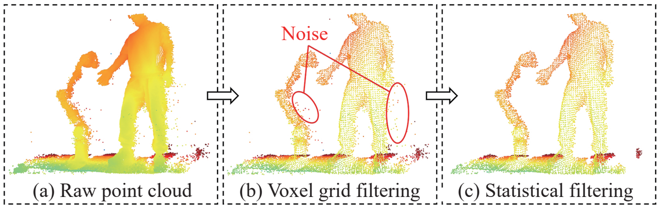

Fig. 2. (a) Point cloud; (b) downsampling; (c) denoising.

bution conforms to a statistical model, the parameters of the statistical model can be obtained by calculating the distance between a given point and its neighborhood point set. Outliers with a certain sparsity will not obey the distribution law of the statistical model; therefore, outliers and other noise can be separated. Suppose there is a point p_i (x, y, z) in the point cloud. d is defined as the mean distance from p_i to the nearest points whose number is m, which obeys a Gaussian distribution $G(d)$ with mean μ_d and variance σ_d. When $d \notin (\mu_d - 3\sigma_d, \mu_d + 3\sigma_d)$, point p_i is considered as an outlier and is removed from the point cloud. The distance d and the distribution $G(d)$ can be calculated as follows:

$$d = \frac{1}{m}\sum_{i=1}^{m}\sqrt{(x-x_i^2)+(y-y_i^2)+(z-z_i^2)}$$

$$G(d) = \frac{1}{\sigma_d\sqrt{2\pi}}e^{-\frac{(d-\mu_d)^2}{2\sigma_d^2}}. \tag{2}$$

The statistical filtering algorithm shows an obvious denoising effect in Fig. 2(c), and avoids the loss of the key feature data of the point cloud.

3) Obstacle space generation: For fear of taking the robot as an obstacle, it is essential to remove robots from the point cloud and retain only the pose information of obstacles, that is, to form an obstacle space. To achieve this goal, we form envelopes around the robot according to the robot's pose state and then remove them. In this study, the robot is a TCR-7 collaborative robot, which is a 7-DOF robot developed by our team [11]. As robot joints J_{2-4} and J_{4-7} have coaxiality in structure, a two-link skeleton can be formed based on joints J_2, J_4, and J_7, as shown in Fig. 3(a). Furthermore, collision detection between the robot and obstacles is achieved based on a two-link robot skeleton.

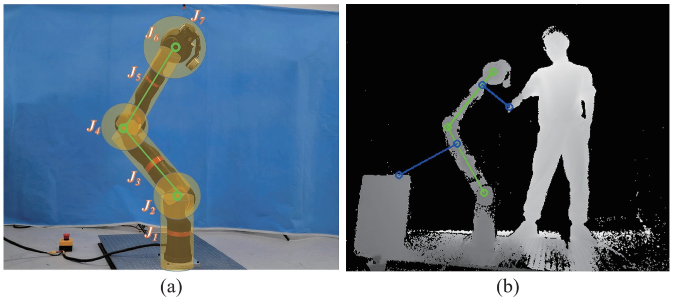

Fig. 3. (a) TCR-7 robot skeleton (green) and envelopes (orange); (b) Distance estimation experiment; the blue line is the closest distance between the robot skeleton and obstacles.

4) Distance estimation: Consider an obstacle point O in obstacle space, whose Cartesian coordinates are $O_C = (x_O, y_O, z_O)$. Owing to the similarity of the robot skeleton structure, only the first link of the robot skeleton in Fig. 3(a) is considered here, that is, the robot structure from joints 2 to 4. The Cartesian coordinates of joints 2 and 4 are $J_{2C} = (x_{J2}, y_{J2}, z_{J2})$ and $J_{4C} = (x_{J4}, y_{J4}, z_{J4})$.

To obtain the distance $\|\boldsymbol{OP}\|$ from the obstacle point to the robot skeleton, the first step is to ensure the nearest position P. Because J_2J_4 has a finite length, we also need to determine whether P is on J_2J_4 using the following:

$$\sigma = \frac{\boldsymbol{J_2O}\cdot\boldsymbol{J_2J_4}}{\boldsymbol{J_2J_4}\cdot\boldsymbol{J_2J_4}} = \frac{\|\boldsymbol{J_2O}\|}{\|\boldsymbol{J_2J_4}\|}\cos(\theta)$$

$$\|\boldsymbol{OP}\| = \begin{cases} \|\boldsymbol{J_2O}\|, & \sigma \le 0, \\ \dfrac{\boldsymbol{J_2O}\times\boldsymbol{J_2J_4}}{\|\boldsymbol{J_2J_4}\|}, & 0 < \sigma < 1 \\ \|\boldsymbol{OJ_4}\|, & \sigma \ge 1 \end{cases} \tag{3}$$

where θ is the angle between vectors $\boldsymbol{J_2O}$ and $\boldsymbol{J_2J_4}$, andσ is the ratio between the projection of $\|\boldsymbol{J_2O}\|$ in vectors $\boldsymbol{J_2J_4}$ and $\|\boldsymbol{J_2J_4}\|$. When $\sigma \le 0$, $\theta > 90°$, obstacle point O is to the left of J_2J_4, and point P is J_2, which is the leftmost point. When $0 < \sigma < 1$, $0 < \theta < 90°$, the position of O is as shown in Fig. 4; P is within J_2J_4, and the relationship satisfies the equation $P = J_2 + \sigma\boldsymbol{J_2J_4}$. When $\sigma \ge 1$, O is to the right of J_2J_4, and P is J_4, which is the rightmost point. In practice, the distance between the robot and obstacle point should also consider the structural size of the robot. Therefore, the envelope shown in Fig. 3(a) is used to replace the robot structure. The actual distance between the robot and the obstacle point can be obtained by

$$\|\boldsymbol{OP}_e\| = \|\boldsymbol{OP}\| - R \tag{4}$$

where P_e is the corresponding position of point P on the surface of the robot envelope, and R is the envelope radius.

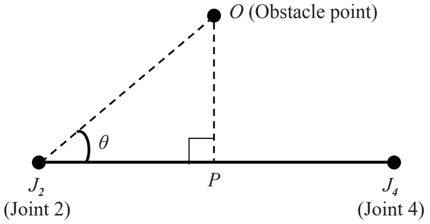

Fig. 4. Distance estimation. J_2J_4 is the first link of the robot skeleton.

The experiment setup is illustrated in Fig. 3(b). The distance estimation method can obtain the nearest obstacle distance, nearest obstacle point, and most dangerous position on the robot structure, which is the key input information in the collision avoidance strategy.

Multi-joint active collision avoidance strategy: The repulsive force based on the artificial potential field causes the robot to move away from the obstacles. To quickly avoid unknown dynamic obstacles, we propose a novel multi-joint repulsive force model. This model can translate this tendency into active collision-avoidance ability.

1) Multi-joint repulsive force model: With the advantages of simple principles and high reliability, the artificial potential field has been applied to the robot control [12]. The field is composed of repulsive forces that keep the robot away from obstacles, and attractive forces that make the robot close to the target position. Compared with the method of applying a repulsive force only at the end-effector in previous studies [3], [5], we propose a method of applying repulsive forces to multiple robot joints.

From (3) and (4), the repulsive force model is defined as follows:

$$\boldsymbol{F} = v\frac{\boldsymbol{OP}}{\|\boldsymbol{OP}\|} \tag{5}$$

where v is the amplitude of the repulsive force, defined as an inverted sigmoid function [3] of the distance $\|\boldsymbol{OP}_e\|$ between the robot and obstacles.

$$v = \frac{V_{\max}}{1 + e^{(\|\boldsymbol{OP_e}\|(2/\rho)-1)\alpha}} \tag{6}$$

where $V_{\max}$ is the maximum value of v, ρ is the dangerous distance threshold, and α is the shape factor.

In general, the repulsive force acts on the position closest to the obstacle, namely point P. As the obstacle move around the robot, the position of point P changes. However, it is found through experiments that the continuous change of repulsive force position causes incoherence of robot motion and affects the effect of robot collision avoidance. Therefore, by transforming the repulsive force, it is applied only at the positions of the three joints (J_2, J_4, and J_7), as shown in Fig. 5.

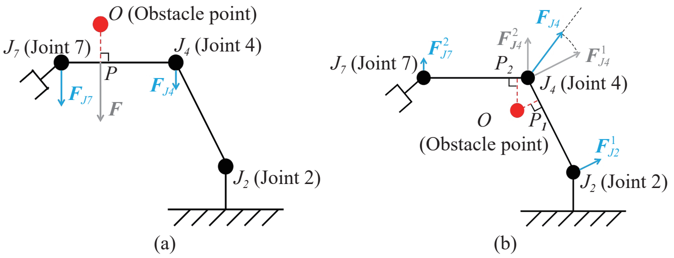

Fig. 5. Schematic of the multi-joint repulsive force model. (a) The obstacle point affects only the J_4J_7 part of the robot skeleton; (b) The obstacle point affects the whole robot skeleton; the blue arrow is the repulsion force finally applied to the robot.

In Fig. 5 (a), the repulsive force $\boldsymbol{F}$ is divided into $\boldsymbol{F_{J7}}$ and $\boldsymbol{F_{J4}}$, which act on joints 7 and 4, respectively. The amplitude and relation between $\boldsymbol{F_{J7}}$ and $\boldsymbol{F_{J4}}$ are as follows:

$$\begin{cases} \boldsymbol{F_{J4}} = \frac{\|\boldsymbol{PJ_7}\|}{\|\boldsymbol{J_4J_7}\|}\boldsymbol{F},\ \boldsymbol{F_{J7}} = \frac{\|\boldsymbol{J_4P}\|}{\|\boldsymbol{J_4J_7}\|}\boldsymbol{F} \\ \boldsymbol{F_{J4}}+\boldsymbol{F_{J7}}=\boldsymbol{F}. \end{cases} \tag{7}$$

The repulsive force is inversely proportional to the distance from point P to the joint position. As shown in Fig. 5(a), P is closer to joint 7, indicating that the obstacle point is closer to joint 7. Thus, a greater repulsive force must be applied at the joint location. Because the distance from the obstacle to $\boldsymbol{J_2J_4}$ is greater than the dangerous distance threshold ρ, the repulsive force is not defined for $\boldsymbol{J_2J_4}$.

In Fig. 5(b), the distances from the obstacles to $\boldsymbol{J_2J_4}$ and $\boldsymbol{J_4J_7}$ are both less than the threshold ρ. Repulsive forces are defined on both $\boldsymbol{J_2J_4}$ and $\boldsymbol{J_4J_7}$, namely, $\boldsymbol{F^1}$ and $\boldsymbol{F^2}$, respectively. It should be noted that joint 4 is the joint in the middle of the robot skeleton; therefore, two forces are applied to joint 4, namely, $\boldsymbol{F^1_{J4}}$ of $\boldsymbol{F^1}$, and $\boldsymbol{F^2_{J4}}$ of $\boldsymbol{F^2}$. If the combined force of $\boldsymbol{F^1_{J4}}$ and $\boldsymbol{F^2_{J4}}$ is regarded as the repulsive force $\boldsymbol{F_{J4}}$ directly, then the value of $\boldsymbol{F_{J4}}$ will be affected by the $\boldsymbol{F^1_{J4}}$ and $\boldsymbol{F^2_{J4}}$ directions. For example, the approximate or opposite directions of $\boldsymbol{F^1_{J4}}$ and $\boldsymbol{F^2_{J4}}$ lead to a sudden increase or decrease in the value of $\boldsymbol{F_{J4}}$. This phenomenon causes the robot motion speed to change rapidly, and may even make the robot unable to complete the avoidance action.

$$\boldsymbol{F_{J4}} = \frac{\boldsymbol{F^1{}_{J4}}+\boldsymbol{F^2{}_{J4}}}{\left\|\boldsymbol{F^1{}_{J4}}+\boldsymbol{F^2{}_{J4}}\right\|}\max(\left\|\boldsymbol{F^1{}_{J4}}\right\|,\left\|\boldsymbol{F^2{}_{J4}}\right\|). \tag{8}$$

To offset this negative influence, the $\boldsymbol{F_{J4}}$ direction is determined by the $\boldsymbol{F^1_{J4}}$ and $\boldsymbol{F^2_{J4}}$ resultant force direction. Its value is the maximum of $\|\boldsymbol{F^1_{J4}}\|$ and $\|\boldsymbol{F^2_{J4}}\|$. This solution causes the value and direction of $\boldsymbol{F_{J4}}$ to change continuously, ensuring the smoothness of the robot motion to a certain extent.

2) Repulsive direction optimization: According to the obstacle point cloud in Fig. 2, an actual obstacle is composed of multiple obstacle points. If only the obstacle point nearest to the robot is considered, then the nearest obstacle point is

$$\boldsymbol{O_{min}} = \arg\min_{O\in S}\|\boldsymbol{OP_e}\|. \tag{9}$$

The term S is the dangerous obstacle area, which is defined by the robot's position and the dangerous distance threshold ρ. The repulsive force can then be changed from that in (5) to

$$\boldsymbol{F_{Omin}} = v(\boldsymbol{O_{min}P_e})\frac{\boldsymbol{O_{min}P}}{\|\boldsymbol{O_{min}P}\|} \tag{10}$$

In the process of multi-joint collision avoidance, considering only the nearest obstacle point cannot make the robot deal with a situation of being surrounded by obstacles. In Fig. 6(a), point P is assumed to be the joint center of the robot (such as J_2, J_4, or J_7). The two sides are obstacles A and B, respectively. Considering only the nearest obstacle O_{min}, a repulsive force to the left is generated at point P. The force is unable to move away from the obstacles, and may even collide with obstacle A.

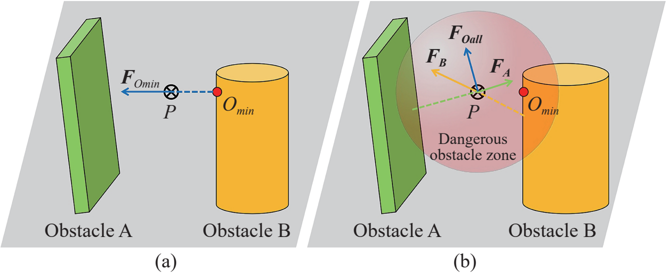

Fig. 6. Repulsive force (blue arrow) calculation method. (a) Only the nearest obstacle point is considered; (b) All obstacle points in the dangerous obstacle space (red transparent sphere) are considered.

Therefore, all obstacle points in dangerous obstacle space S need to be calculated, as shown in Fig. 6(b) and expressed by (11). It can be seen that the direction of $\boldsymbol{F_{Oall}}$ is jointly determined by all obstacle points in space S, and the correct collision avoidance direction is generated, as shown in Fig. 6(b). The terms $\boldsymbol{F_A}$ and $\boldsymbol{F_B}$ are the resultant repulsion forces generated at the dangerous obstacle points in obstacles A and B, respectively. They satisfy the relationship $\boldsymbol{F_S} = \boldsymbol{F_A} + \boldsymbol{F_B}$. The amplitude of $\boldsymbol{F_{Oall}}$ is determined only by the nearest obstacle point. If all dangerous obstacle points are used to calculate the amplitude, then the magnitude will be affected by the number of obstacle points, which may cause the robot to perform an incorrect collision-avoidance action.

$$\begin{cases} \boldsymbol{F_s} = \sum_{O\in S}\boldsymbol{F}(O) \\ \boldsymbol{F_{Oall}} = v(\boldsymbol{O_{min}P_e})\frac{\boldsymbol{F_s}}{\|\boldsymbol{F_s}\|}. \end{cases} \tag{11}$$

3) Robot motion control: The motion control framework of the robot is illustrated in Fig. 7. Its motion states include the desired task motion and collision-avoidance motion. First, the safety state of the robot is determined by the collision-detection result. If the robot is not safe, it enters the collision avoidance motion state and outputs the repulsive force $\boldsymbol{F_{Oall}}$. Otherwise, the robot enters the desired task motion state and outputs attractive force $\boldsymbol{F_{Att}}$. The definition of attractive force is similar to the mathematical model of repulsive force in (5) and (6); however, the difference is that the attractive force is to estimate the distance of the robot to the target position.

The dynamic calculation then converts the repulsive and attractive forces into joint torques, which in turn translate into joint acceleration. In the process of switching between two motion states, the change in forces is discontinuous, which may lead to abrupt changes in joint acceleration. Therefore, the Savitzky-Golay filter based on the least-squares fitting method [13] is used to smooth the joint acceleration. Finally, the smooth-filtered acceleration is converted into joint velocity through integration. Robot motion is controlled based on the joint velocity level.

Experiments: Collision-avoidance experiments and their results are used to analyze and evaluate our proposed method. The experimental setup includes a TCR-7 collaborative robot and its controller, a Kinect V2 camera, and a computer for real-time data processing (Ubuntu 18.04 system). The repulsive force parameters in (6) were set as follows: $V_{\max} = 60.0$ N, $\rho = 0.30$ m, and $\alpha = 6.0$.

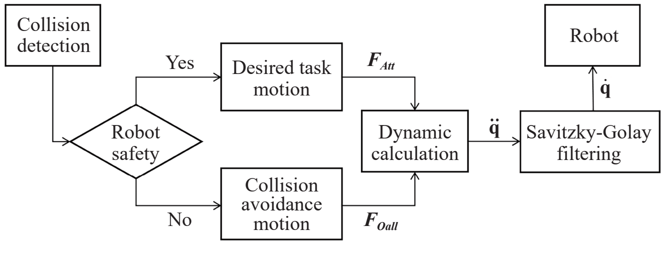

Fig. 7. Robot motion control framework.

1) Experiment 1 (End-effector collision avoidance): In the experimental scheme, the end-effector needs to avoid dynamic obstacles, and quickly return to the target position after avoiding the collision. Fig. 8 shows the results of Experiment 1. In the face of the two hands approaching from above and below, the end-effector generates a reasonable repulsive force according to the multi-obstacle visual data.

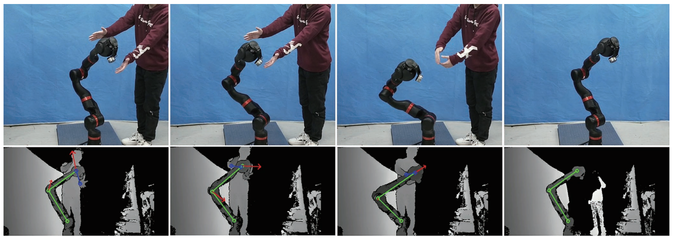

Fig. 8. Experiment 1: End-effector collision avoidance. The red arrows are repulsive forces, which represent the motion trend of the robot.

The change in the closest distance between the robot end-effector and the obstacles is shown in Fig. 9(a). When the closest distance is less than ρ, the robot avoids dangerous obstacles. The experimental results show that the closest distance is always greater than the distance limit of 150 mm (red dotted line). The minimum distance is 0.192 m, which occurs at 0.73 s.

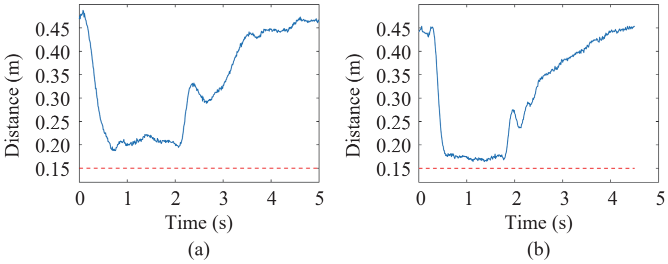

Fig. 9. The closest distance between robot and obstacles. (a) and (b) correspond to Experiments 1 and 2 respectively; the red dotted line is the distance limit.

The position trajectory of the robot end-effector in Experiment 1 is shown in Fig. 10(a). The gradual change in the trajectory color and arrow represents the changing trend of the end-effector position over time. After collision avoidance, the end-effector can accurately return to the original target position; the position error is 1.603 mm.

2) Experiment 2 (Multi-joint collision avoidance): Experiment 2 demonstrates the collision avoidance effect of our multi-joint active method on the robot's whole body. When the obstacles approached, the robot suspended the desired task motion and performed a collision avoidance motion. When the robot completed collision avoidance, the end-effector returned to the pause position and continued to perform the task along the desired trajectory. Fig. 10(b) shows the trajectory of the end-effector position in Fig. 11. The maximum position error between the end-effector and the desired trajectory when performing the desired task is 2.589 mm. In Fig. 9(b), the minimum distance between the robot's whole body and the obstacles is 0.165 m, which occurs at 1.15 s.

Conclusions: We propose a multi-joint active collision avoidance method based on depth visual perception, that makes the robot's whole body has a rapid response and smooth motion in unknown dynamic obstacles environment. The core of this method is an innovative model that combines depth vision and a robot skeleton to generate active repulsion forces for robot joints, whose reliability is verified by the experiments. But now the method does not consider the optimization of motion planning, which is the direction of our future research.

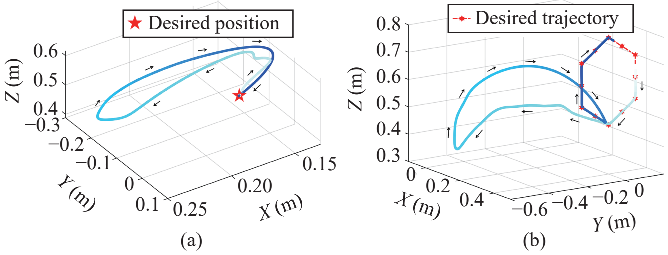

Fig. 10. The trajectory of robot end-effector. (a) Results of Experiment 1; the end-effector task is to reach the desired position; (b) Results of Experiment 2; the end-effector task is to draw a hexagonal trajectory.

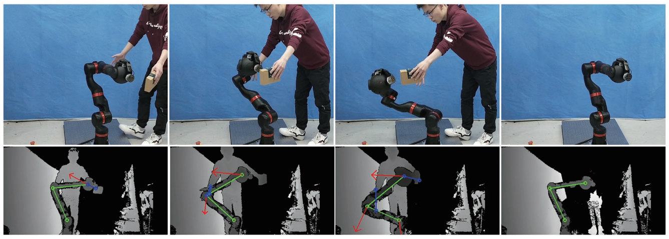

Fig. 11. Experiment 2: Multi-joint collision avoidance.

Acknowledgments: This work was supported in part by the National Key Research and Development Program of China (2018YFB1305300), the China Postdoctoral Science Foundation (2020TQ0039, 2021M700425), the National Natural Science Foundation of China (61733001, 62103054, U2013602, 61873039, U1913211, U1713215).

References

[1] H. Wei, Z. Li, and C. L. P. Chen, " A survey of human-centered intelligent robots: Issues and challenges," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 4, pp. 602–609, Oct. 2017.

[2] A. Zacharaki, I. Kostavelis, *et al.*, " Safety bounds in human robot interaction: A survey," *Saf. Sci.*, vol. 127, no. 104667, Jul. 2020.

[3] F. Flacco, T. Kröger, A. De Luca, *et al.*, " A depth space approach to human-robot collision avoidance," in *Proc. IEEE Int. Conf. Rob. Autom.*, 2012, pp. 338–345.

[4] H. Nascimento, "Collision avoidance interaction between human and a hidden robot based on Kinect and robot data fusion," *IEEE Robot. Autom. Lett.*, vol. 6, no. 1, pp. 88–94, Jan. 2021.

[5] J. H. Chen and K. T. Song, "Collision-free motion planning for human-robot collaborative safety under Cartesian constraint," in *Proc. IEEE Int. Conf. Rob. Autom.*, 2018, pp. 4348–4354.

[6] F. Flacco, T. Kröger, A. De Luca, *et al.*, " A depth space approach for evaluating distance to objects," *J. Intell. Rob. Syst.*, vol. 80, no. 1, pp. 7–22, Dec. 2015.

[7] M. Dennis, K. Tobias, *et al.*, " A constraint-based approach for human-robot collision avoidance," *Adv. Rob.*, vol. 34, no. 5, pp. 265–281, Mar. 2020.

[8] Y. H. Lee and K. T. Song, "Real-time obstacle avoidance with a virtual torque approach for a robotic tool in the end-effector," in *Proc. IEEE Int. Conf. Rob. Autom.*, 2021, pp. 8436–8442.

[9] F. Flacco, A. De Luca, and O. Khatib, " Control of redundant robots under hard joint constraints: Saturation in the null space," *IEEE Trans. Rob.*, vol. 31, no. 3, pp. 637–654, Jun. 2015.

[10] O Wasenmüller and D Stricker, "Comparison of Kinect v1 and v2 depth images in terms of accuracy and precision," in *Proc. Asian Conf. Comput. Vis. Workshop*, 2016, pp. 1–12.

[11] Z. Sun, *et al.*, " BIT-DMR: A humanoid dual-arm mobile robot for complex rescue operations," *IEEE Robot. Autom. Lett.* , vol. 7, no. 2, pp. 802–809, Apr. 2022.

[12] Y. Yang, L. Liao, H. Yang, and S. Li, "An optimal control strategy for multi-UAVs target tracking and cooperative competition," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 12, pp. 1931–1947, Dec. 2021.

[13] R. W. Schafer, "What is a Savitzky-Golay filter," *IEEE Signal Process. Mag.*, vol. 28, no. 4, pp. 111–117, Jul. 2011.