Active Collision Avoidance for Human-Manipulator Safety

GUANGLONG DU¹, YINHAO LIANG¹, GENGCHENG YAO¹, CHUNQUAN LI², RONIGUES J. MURAT¹, AND HUA YUAN¹

¹School of Information Engineering, South China University of Technology, Guangzhou 510006, China
²School of the Computer Science and Engineering, Nanchang University, Nanchang 330031, China

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

This work was supported in part by the National Natural Science Foundation of China under Grant 61973126, Grant 61863028, Grant 81660299, and Grant 61503177, in part by the Guangdong Natural Science Funds for Distinguished Young Scholar under Grant 2017A030306015, in part by the Pearl River S&T Nova Program of Guangzhou under Grant 201701010059, in part by the Guangdong special projects under Grant 2016TQ03 × 824, in part by the Science and Technology Department of Jiangxi Province of China under Grant 20161ACB21007, Grant 2017BB50071, and Grant 2017BJB202033, in part by the Fundamental Research Funds for Central Universities under Grant 2019ZZD27, in part by the Science and Technology Planning Project of Guangdong Province under Grant 2017B090914002, and in part by the Innovation Team of the Modern Agriculture Industry Technology System in Guangdong Province under Grant 2019KJ139. Nanchang University and South China University of Technology contributed equally.

ABSTRACT This paper proposes a novel method for active collision avoidance to protect the human who enters a robot’s workspace in a human-robot collaborative environment. The proposed method uses a somatosensory sensor to monitor the robot’s workspace and detect anyone attempting to enter it. When someone enters the workspace, a Kinect detects and calculates the position of his or her skeleton points in real-time. However, due to the measurement errors and noise of the device, the tracking error increases over time. Therefore, the proposed method applies an improved particle filter (IPF) to accurately estimate the position of the skeleton points. In order to detect the human-robot collision in real-time, the proposed method uses cylinders to establish the bounding box model for human bones and robots and the human-robot collision is replaced by the collision between the cylinders, greatly improving the efficiency of collision detection. Moreover, taking human safety and productivity into account, the robot velocity control is carried out based on the distance between the robot and human. Then, the proposed method uses a rule-based logic system to analyze human motion so that the robot can take appropriate measures to avoid humans. Finally, the dynamic roadmap (DRM) approach plans new paths in real-time to allow robots to bypass humans. By actively avoiding collisions, the proposed method ensures that the robot will never touch the human body. The significant advantage of the proposed method is that it can detect humans in real-time, analyze their behavior and protect humans without any modification to the robot. The proposed method has been tested in practical applications, and the results show that it can successfully guarantee the safety of people entering the robot’s workspace.

INDEX TERMS Active collision avoidance, dynamic roadmap (DRM), human-robot safety, improved particle filter (IPF).

I. INTRODUCTION

With the development of Industry 4.0, robots tend to be intelligent and collaborative in the future [1]–[3]. The human-robot collaboration will bring wider use of robots and promote robots to become indispensable partners in human life [4]. However, the traditional methods of industrial production could not meet the needs of the enterprise’s production safety. Safety has become one of the most important factors in the future development of human-robot interaction. Although human-robot collaboration allows robots to do broader and more complex tasks, it raises the safety concerns of the robots. However, increasing robot safety often means a compromise in their performance, which requires designers to find a balance between safety and performance. At present, collaborative robots can perceive the environment and change their behavior accordingly. For example, when a robotic arm hits the human arm, the robot can perceive the existence of
the human arm through the force sensor. Then it stops or goes away to protect human safety. This feature limits the performance of collaborative robots, such as speed, load, and so on. Therefore, an effective method is proposed to predict the possible collision between robots and humans, and timely feedback is provided to ensure human safety.

When a robot performs work in an unstructured, dynamic, and populated environment, finding a way to work safely and efficiently becomes more challenging [5]. Safe and efficient human-robot interaction is indispensable to future robotics applications, where humans and robots must collaborate to perform tasks jointly [6]. To meet the strictest safety requirements, robots must have an excellent ability to change motion paths in complex environments. In any human-robot collaborative system or any environment where humans and robots coexist, ensuring human safety at all times is a considerable challenge. In order to help humans in daily life, several studies have been performed to develop robot systems, which include interaction and cooperation with humans [7]. These studies indicated that robots are very useful in a variety of tasks performed in collaboration with humans. For this reason, finding a way to protect human operators in human-robot collaborative system is highly relevant. This not only includes detecting possible collisions in real-time, but also avoiding collision with humans during runtime.

Active collision avoidance in real time consists of three parts [8]: 1) Environment perception; 2) collision avoidance; 3) robot control. Collision avoidance is one of the most important fields in robotics. Many real-time capable planning concepts are based on the potential field methods that were introduced in two previous works. In these methods, virtual attractive and repulsive fields are used to represent targets and obstacles. In this case, the robot is attracted to targets and moves away from obstacles. Real-time planning algorithms are significant to robotic systems [9].

Despite many achievements in robotic safety, most real-time collision avoidance methods are either too expensive or have very limited security in practical applications.

1) Some methods, for example [17], are not practical in the real world because these methods require a dedicated set with built-in sensors, which can hinder the user’s motion and increase cost in practical applications.

2) References [21]–[24] treat the human body as a moving obstacle, so the robot may mistakenly believe that the human body is the operation object and put people at risk. In addition, due to the lack of independent intelligence, the efficiency of collision avoidance is relatively low.

3) Current collaborative robots are relatively expensive, so replacing a large number of traditional robots with cooperative robots is not a realistic task. Moreover, cooperative robots tend to adopt post-collision detection, which inevitably puts people at risk.

To address these issues, this paper proposes an active collision avoidance method for human-robot incorporation. In this proposed method, Kinect is used to detect humans in real-time; instead of particle filter (PF), the improved particle filter (IPF) algorithm for Kinect is applied to estimate the position of the skeleton joint accurately, thus improving the accuracy of human tracking. Moreover, robot velocity control is implemented to ensure the safety of humans while improving production efficiency. A rule-based logic system is developed to enable the collision avoidance system to obtain the most appropriate response strategy according to different behaviors of the human. When humans are immobile, dynamic roadmap (DRM) is used to plan new paths for the robot directly. When humans move slowly, the boundary box is calculated, and the robot bypass this box by using DRM to plan a new path. When humans move too fast, the robot immediately stops. Therfore, the proposed collision avoidance method can take different measures according to different human behaviors for more effectively and safely protecting humans. The main contributions are listed as follow:

1) Cylinders were adopted to build a bounding box model for both the human bones and the robots. Hence, collision detection can be achieved by detecting the relative position between the cylinders instead of the point cloud interaction in the depth image, thereby improving the timeliness and effectiveness of the collision avoidance method.

2) The proposed active collision avoidance method can analyze and predict human behavior and choose the best path for the robot so that the flexibility and intelligence of the robot movement can be improved.

3) In the proposed active collision avoidance method, an intelligent system with rule-based logic is proposed for analyzing human behavior so that the system can correctly respond to different human movement speeds, and the DRM was adopted to plan a new path, thereby improving the effectiveness and the independent intelligence of the proposed collision avoidance method.

The remainder of the paper is organized in sections. Section III provides an overview of the paper. Section IV presents details about the modeling of humans and robots, which is the basis of the paper. Section V describes the process of estimating skeleton position using improved PF. Section VI presents fast path planning using DRM. Section VII details robot velocity control and Section VIII describes active collision avoidance using a set of rules. The experiments and results are presented in Section IX. Discussion is given in Section X and the conclusion in Section XI.

II. RELATED WORK
Planning algorithms allow robots to avoid obstacles in real environments. Jiang et al. [10] described a planning algorithm that can adaptively coordinate humans and robots in the hybrid assembly system. Zhang et al. [11] proposed an autonomous path planning method based on improved rapidly-exploring random tree algorithm. Han et al. detected
obstacles in real-time and re-planned the path based on distance calculation and discrete detection [12]. Some studies have assessed obstacle avoidance in dynamic environments [13]. Most of them have treated operators as moving obstacles. This meant that they only need to consider collision avoidance of robots. References [14], [15] computed the distance between the human and the robot by analyzing images of the environment. Most of the collision detection methods cited above require that information about the environment must be available. Such collision avoidance algorithms are primarily based on the distance between the industrial robot and the detected objects in the real environment.

In human-robot collaboration systems, two approaches of avoiding collision have been used for safety: vision-based methods, which analyze motion, color and texture [16], and inertial sensor-based methods, which uses special suits for motion capture [17]. Currently, vision-based methods are a better choice for avoiding collisions in robotic control systems because in this way the operator does not have to wear additional equipment. Meanwhile, the surrounding environmental information, not just the motions of the operator, can be obtained. In addition, with the development of various 3D vision sensing system, some new depth cameras, like the Microsoft Kinect sensor [18], have facilitated the development of a powerful and easy to use sensor system.

In recent years, studies of the vision-based method have focused on increasing the efficiency of collision detection. One method involved a multi-camera system to detect obstacles [19]. Tan and Arai [20] used a triple stereo-vision system to track the movement of a sitting operator who was wearing a color marker. References [21]–[23] implemented collision detection and avoidance based on the depth images. Ahmad and Plapper [24] proposed an automatic path planning algorithm for robot manipulator, which located the unknown obstacles based on the time-of-flight (TOF) sensor.

### III. OVERVIEW

Human safety can be questionable in environments containing working robots. In this paper, an active collision-avoidance method is proposed to protect the humans who enter robot’s workspace. Fig. 1. Shows the flowchart of the proposed method. First, a Kinect, which has two infrared cameras for depth detection and one visual-spectrum camera for visual recognition, is used to detect the skeletons inside humans who enter the workspace. According to the position data of the skeleton joints of the human body from the Kinect, the human breaking into the robot’s workspace can be detected even if the human is static. In this way, the risk that the robot mistakes humans for the object of operation can be reduced, thereby ensuring the safety of the human. After building bounding cylinders for humans and robots, collision detection is achieved by detecting the relative position between the cylinders instead of the point cloud intersections of the depth image, which improves the accuracy and effectiveness of the proposed method.

To improve the precision of the skeleton points detection, an IPF was used instead of the traditional PF for estimation. For the sake of the safety of humans while improving efficiency, the robot velocity control is implemented based on the distance between the robots and humans. Once the robot-to-human distance is less than the safe distance, the speed of the robot begins to linearly reduce relative to the distance. Furthermore, in order to achieve real-time obstacle avoidance, the most commonly used strategy is to repeatedly plan the path, but this will greatly increase the avoidance time and the avoidance number, reducing the intelligence and flexibility of the system. Therefore, a rule-based logic system was developed to analyze the behaviors of the human so that the system could take different measures based on the current motion states of the human. When the human is static, the system plans a new path using DRM. If the human approaches slowly, the system builds a bounding sphere for the human and then plans a new path using DRM. When the human approaches quickly, the robot immediately stops. According to the current motion state of the human, the robot adopts different reaction strategies, thus improving the overall intelligence of the system. In this way, not only can the avoidance time of the robot caused by the repetitive planning path be reduced, but also the required cost can be reduced.

### IV. MODELING OF HUMANS AND ROBOTS

#### A. BUILDING BOUNDING CYLINDERS FOR HUMANS USING KINECT

The human movement tracking approach is driven by two key design goals: computational efficiency and robustness.
Nowadays, various depth cameras are widely used in human motion recognition as the position of the skeletal joints can be estimated by the depth image caught by these cameras. As illustrated in [25], a single input depth image is segmented into dense body parts with different probabilities and labels, which are used to estimate the 3D skeletal joints. In the proposed method, Kinect is used as a depth camera for extracting human motion data considering that the Kinect has a larger infrared sensor size and a wider field of view and produces a depth image with a higher quality relative to other depth cameras [26]. The Kinect also provides an application program interface (API) for human tracking and positioning. When someone enters the robot’s workspace, the API can detect the human according to the characteristics of the three-dimensional data and calculate the location of the human’s skeleton.

As mentioned above, the joints of the skeleton can be located using Kinect. Fig. 2 shows the 15 skeleton joints in the RGB image. The 15 skeleton joint points are numbered from top to bottom and left to right. Cylinders are used to construct the human body based on the volume of the different parts of the body (Fig. 2). Pairs of adjacent points are used to build each cylinder. The distance between the two adjacent points was used to determine the radius and length of the cylinders, so accounting for the sizes of different persons. The cylinders are different for every part of the body. For example, the cylinder for the torso is bigger than the cylinder for the arm. The scaling relationship between the size of a cylinder and the length of the corresponding two adjacent points can be calculated using statistical data.

B. BUILDING BOUNDING CYLINDERS FOR ROBOTS WITH DH PARAMETERS

The standard Denavit-Hartenberg (D-H) [27] convention is the most widely used means of describing robot kinematics [27]. The position of the robot and its tools and their orientation are defined according to the controller conventions. Through consecutive homogeneous transformations from the base coordinate to the robot tool coordinate, the kinematic equation can be defined as follows:

$$T_N^0 = T_N^0(v) = T_1^0T_2^1 \cdots T_N^{N-1} = \prod_{i=1}^{N} T_i^{i-1}$$  \hspace{1cm} (1)

where $T_i^{i-1}$ is the translation matrix from the $i-1$ coordinate to the $i$ coordinate, $N$ is the number of joints (or joints coordinates). The position of each joint $j$ can be calculated using Eq. (1).

$$T_{j}^{N} = \prod_{i=1}^{j} T_{i}^{i-1}$$  \hspace{1cm} (2)

Because the size of each link can be obtained from factory parameters, the radius of the cylinder of each link can be determined. The length of the bounding cylinder of link $j$ is the distance between the position of joint $j - 1$ and $j$. Then, the cylinder of each link can be built after the radius and the length is determined (Fig. 3).

C. COLLISION DETECTION FOR CYLINDERS

Because both the robot and the human are bounded by cylinders, using cylinder-cylinder collision detection can efficiently approximate the collision case between a human and a robot. Given two colliding cylinders $A$ and $B$, the relative positions between them could be classified into three categories according to whether the cylinders intersect or not at their bases, as shown in Fig. 4. The case in which one base of $A$ interact with one base of $B$ is called base-intersection (Fig. 4 (a)). The case in which the two cylinders intersect only at the sides is called lateral-quadrature (Fig. 4 (b)). And the case in which one base of one cylinder intersects with the side of another cylinder is called lateral-heterotrophic (Fig. 4 (c)).

For details about collision detection for cylinder-shaped bodies, please refer to [28].
V. HUMAN TRACKING USING IMPROVED PARTICLE FILTER

A. IMPROVED PF

Although the human skeleton position can be measured by the Kinect, the inherent noises of the sensor can increase over time. To reduce the measurement errors and noise, an improved PF was used to improve the reliability of the proposed methodology. The PF algorithm, which is a state estimator [29], can be used to approximate the posterior with a finite number of state samples that have corresponding normalized weights. At time \( t_k \), the approximation of posterior density can be defined as follows [29]:

\[
p(x_k | z_{1:k}) \approx \sum_{i=1}^{N} w_{i,k} \delta (x_k - x_{i,k})
\]  

where \( x_{i,k} \) is the position state of the \( i \)th particle at time \( t_k \), \( z_{1:k} \) represents the observations, \( N \) is the number of samples, \( w_{i,k} \) is the normalized weight of the \( i \)th particle at time \( t_k \) and \( \delta (\cdot) \) is the Dirac delta function.

As in one previous study [30], [31], an ensemble Kalman filter (EnKF) is used to approximate the posterior probability function (PDF) of state variables \( \{x_{i,0}\}_{i=1}^{N} \). Given the initial ensembles \( \{x_{i,0}\}_{i=1}^{N} \), the forecast ensembles \( \{x^f_{i,k}\}_{i=1}^{N} \) can be calculated as follows:

\[
x^f_{i,k} = f(x^a_{i,k-1}) + w_{i,k-1}, w_{i,k-1} \sim N(0, Q_{k-1})
\]  

where \( w_k \) is the model error and \( Q_{k-1} \) represents the covariance of the model error. The analyzed particles can be calculated as follows:

\[
x^a_{i,k} = x^f_{i,k} + K_k[z_k - h(x^f_{i,k}) + v_{i,k}], v_{i,k} \sim N(0, R_k)
\]  

where \( v_k \) is the observation error, \( K_k \) represents the covariance of the observed error and \( h(\cdot) \) is the measurement matrix. The Kalman gain \( K_k \) can be obtained as follows:

\[
K_k = P_k^f H^T (H P_k^f H^T + R_k)^{-1}
\]

\[
P_k^f H^T = 1/(N-1) \sum_{i=1}^{N} (x^f_{i,k} - \bar{z}_k)[h(x^f_{i,k}) - h(\bar{x}_k)]^T
\]

\[
H P_k^f H^T = 1/(N-1) \sum_{i=1}^{N} [h(x^f_{i,k}) - h(\bar{x}_k)][h(x^f_{i,k}) - h(\bar{x}_k)]^T
\]

\[
x^a_{i,k} = 1/N \sum_{i=1}^{N} x^f_{i,k}
\]  

According to the literature [32], the weight updating formula is calculated as follows:

\[
w_{i,k} = w_{i,k-1} \frac{p(z_k | x_{i,k}) \cdot p(x_{i,k} | x_{i,k-1})}{r(x_{i,k} | x_{i,k-1}, z_k)}
\]  

where \( r(x_{i,k} | x_{i,k-1}, z_k) \) is the importance density. When \( r(x_{i,k} | x_{i,k-1}, z_k) \) is selected as \( p(x_{i,k} | x_{i,k-1}) \), Eq. (7) can be expressed as:

\[
w_{i,k} = w_{i,k-1} p(z_k | x_{i,k})
\]  

To avoid particle degradation, particles must be resampled and then the same weight \( (w_{i,k} = 1/N) \) is assigned to all the particles. A Markov Chain Monte Carlo (MCMC) method [32] is used to improve the diversity of particles after resampling. The transition process is implemented as follows:

\[
p(x_k | z_{1:k})k(x_k^f | x_k)dx_k = p(x_k^f | z_{1:k})
\]  

where \( k(x_k^f | x_k) \) is the Markov transition kernel.

In the Metropolis Hasting algorithm, the resampled particles move to the proposed particle only if \( u \leq a \), which are defined as

\[
a = \min(1, \frac{p(z_k | x_k^f)}{p(z_k | x_k)})
\]

\[
u \sim U[0, 1]
\]  

After the MCMC step, the new particles are more diverse because they have a distribution closer to the posterior PDF.

B. ESTIMATION OF SKELETON POSITION USING IMPROVED PF

In the proposed method, the human skeleton position is measured by the Kinect and an improved PF is used to estimate the position. Each particle position has three \( x_{i,PF} \) states \( (x_{i,PF,k} = [p_x, p_y, p_z]) \). The deterministic input is set to zero since there is no deterministic input to the system. The accumulated position difference of the \( i \)th particle at the \( s \)th position iteration as follows:

\[
JD_{i,s} = \sum_{k=(s-1)M_s+1}^{M_s} \left( p_{p_x,k} - p_{\hat{p}_x,k} \right)^2 + \left( p_{p_y,k} - p_{\hat{p}_y,k} \right)^2
\]

\[
+ \left( p_{p_z,k} - p_{\hat{p}_z,k} \right)^2
\]  

Here, \( M_s = \Delta T_s / t \) and \( p_{\hat{p}_x,k} \) is the position state of the \( i \)th particle and \( p_{\hat{p}_x,k} \) is the position state of the \( i \)th particle estimated by EnKF at time \( k \). Through the \( JD_{i,s} \) values, the weight of each particle would be recalculated in every \( \Delta T_s \) period. Then, \( JD_{i,s} \) should be adopted in the IPF instead of the direct measurements [33]. Therefore, the approximate posterior is represented by:

\[
p \left( x_s | JD_{i,s} \right) \approx \sum_{i=1}^{N} w_{i,s} \delta \left( x_s - x_{i,s} \right)
\]  

Then, the following equation is satisfied:

\[
p \left( x_s | JD_{i,s} \right) \propto p \left( JD_{i,s} | x_s \right) p \left( x_s | x_{s-1} \right) p \left( x_{s-1} | JD_{i,s-1} \right)
\]  

and the weight of the \( i \)th particle [33] can be expressed as:

\[
\omega_{i,s} \propto p \left( JD_{i,s} | x_{i,s} \right) p \left( x_{i,s} | x_{i,s-1} \right) \omega_{i,s-1}
\]  

By picking the important density and resampling, the weight of the \( i \)th particle can be obtained as follows:

\[
\omega_{i,s} \propto p \left( JD_{i,s} | x_{i,PF,s} \right)
\]
VI. FAST PATH PLANNING USING DRM

In dynamic environments, the angles between joints and the degree of rotation will change a lot. The motions of the robot in three-dimensional space are very complex and may be occluded at any time considering that obstacles are mostly dynamic. In this way, obstacle avoidance is necessary for the robot to work in changing environments. In this paper, dynamic roadmap (DRM [34]), an adaptive real-time path planning method that incorporates a pre-processing stage and a planning stage, was adopted for obstacle avoidance. DRM uses pre-computation to increase online efficiency by generating a pre-processed roadmap. For a given time, DRM can quickly determine which parts of the roadmap are blocked before each planning step. An enhanced DRM can be produced through a pre-processing stage and a planning stage [35].

In the preprocessing stage, the first step is to build undirected roadmaps in the q-dimensional configuration space of the robot, in which q is the number of robot joints. The second step is to look for conflict points and edges in the roadmaps. First, the nodes of the roadmaps were created using a uniform sampling method. Then, for each node, according to pseudo-norm space, its k nearest neighbors were identified provided that the segment between the two nodes was collision-free and connected as part of the roadmap. The discrete configurations at roadmap nodes and along edges for self-collision were checked by using a related algorithm. The direct workspace metric was defined as follows:

\[ d_{\infty}(p, q) = \max_{a \in A} \| a(p) - a(q) \| \]  

(16)

Here, A is the set of all reference points on the surface of the robot and a(p) is the location of reference point a in the workspace if the robot is in configuration p. Eq. (16) provides the maximum degree to which any reference point may be displaced between two checked configurations. If the robot is bounded by a convex polyhedron and all of its vertices are selected as reference points, the metric gives the maximum displacement of any point on the robot.

It is to be noted that self-collision points and edges must be removed when creating the roadmap. For this reason, after creating the roadmaps, in order to look for conflict points and edges, the DRM method was used to create a map of the area between the workspace grid and the configuration space roadmap. The planning stage involves the following steps: invalidating blocked parts of the roadmap, connecting start, goal configurations to the roadmap and searching the graph.

VII. ROBOT VELOCITY CONTROL

In a human-robot collaborative environment, human safety can be difficult to guarantee.

In order to ensure the safety of humans while improving the efficiency of the robot in the process of human-robot cooperation, the movement speed of the robot is determined based on the distance between the robot and the human. When the present distance \( S_{\text{present}} \) is less than the safety distance \( S_{\text{safe}} \), the speed of the robot decreases linearly with \( S_{\text{present}} \). When the present distance \( S_{\text{present}} \) is less than the minimum distance \( S_{\text{min}} \), the robot stops working. As shown in Fig. 5, according the distance between the robot and the human, the zone is divided into three regions including the safe zone, the acc/dec zone and the stopping zone.

In the proposed method, the robot takes different speeds based on the distance between the robot and the human. The allowable speed of the robot is defined in the three areas (Fig. 5) as follows [36]:

\[
V_r = \begin{cases} 
V_R & S_{\text{safe}} < S_{\text{present}} \\
S_{\text{present}} < S_{\text{safe}} < S_{\text{min}} \\
0 & S_{\text{present}} < S_{\text{min}}
\end{cases} \tag{17}
\]

The safe distance \( S_{\text{safe}} \) is given by:

\[
S_{\text{safe}} = V_h(T_r + T_b) + V_r T_r + S_h + C + Z \\
T_b = \frac{T_b V_{\text{max}}}{V_{\text{max}} V_r} \tag{18}
\]

where \( V_h \) and \( V_r \) represent the speed of the human and the robot, respectively. \( T_r \), \( T_b \), and \( S_h \) denote the collision avoidance system reaction time, the braking time with respect to \( V_r \) and the braking distance of the robot, respectively. \( C \) and \( Z \) represent the resolution distance of the sensor and the uncertainty distance based on the human and the sensor. \( V_{\text{max}} \) is the maximum speed of the robot under the maximum load. The minimum distance \( S_{\text{min}} \) is defined as:

\[
S_{\text{min}} = V_h T_r + C + Z \quad \text{when } V_r \approx 0 \tag{19}
\]

From Eq. (17), When the distance between the robot and the person is greater than \( S_{\text{safe}} \), the speed of the robot \( V_R \) remains static. When the distance is greater than \( S_{\text{min}} \) and less than \( S_{\text{safe}} \), the speed of the robot mainly depends on the speed of the robot and the person and the braking distance. When the distance is less than \( S_{\text{min}} \), the robot stops moving to ensure safety.

VIII. ACTIVE COLLISION AVOIDANCE USING RULE-BASED LOGIC SYSTEM

Since human behavior is unpredictable, real-time path planning alone cannot achieve intelligent obstacle avoidance. Our
collision avoidance system uses a knowledge-based set of rules for decision making [36]. The knowledge-based contains rules that are usually expressed in the form of “IF A THEN B,” where A is the antecedent conditions and B is the consequences.

In the proposed approach, when operators enter the robot’s workspace, that is, when the distance of the operator from the robot $D_{HR}$ is lower than the dangerous distance $D_{HR_{\text{min}}}$ ($D_{HR_{\text{min}}}$ refers to the maximum working range of the robot and its tool), the collision avoidance system is triggered. According to the distance of the Kinect from the industrial robot, the maximum detection radius $R_{\text{max}}$ of the sensor can be determined. From the above section of the robot velocity control, the safety distance $S_{\text{safe}}$ represents the distance for which the robot can avoid people in time. Let $S_{\text{safe}} = R_{\text{max}}$, and formula (18) can be rewritten as follow:

$$R_{\text{max}} = V_{\text{H_{danger}}} (T_r + T_b) V_{\text{max}} + V_{\text{max}} T_r + S_b + C + Z \quad (20)$$

where $V_{\text{max}}$ represents the maximum speed of the robot under the maximum load, $T_r$, $T_b$, $V_{\text{max}}$, and $S_b$ denote the collision avoidance system reaction time, the braking time with respect to $V_{\text{max}}$ and the braking distance of the robot, respectively. $C$ and $Z$ represent the resolution distance of the sensor and the uncertainty distance based on the human and the sensor. Once the allowable maximum speed $V_{\text{max}}$ of the robot is determined, the allowable maximum speed $V_{\text{H_{danger}}}$ of the people can be obtained. Furthermore, the final speed $V_{\text{H_{danger}}}$ is determined by simulating the robot obstacle avoidance in the simulation environment. To achieve active collision avoidance, a set of rules is formulated in three cases where a human may enter the robot’s workspace:

Case 1: The human is approaching at a fast pace. When the human gets close to the manipulator at the speed $V_{\text{H}} > V_{\text{H_{danger}}}$ m/s (where $V_{\text{H_{danger}}}$ is a dangerous speed), the system cannot ensure the new planned path is safe during the next iteration. The wisest choice is to avoid the human rather than plan a new path right away. When the human gets close to the manipulator in the direction of $\hat{D}_H$, the manipulator should avoid the human in the direction of $\hat{D}_H$ so that they do not collide.

Case 2: The human is slowly approaching the robot. When the human gets close to the manipulator at the speed ($0 < V_{\text{H}} < V_{\text{H_{danger}}}$ m/s), the system predicts the motion trail of the human and plan a new path to avoid the human using DRM. Because of the randomness of human movement, the system calculates a bounding sphere that should contain all possible motion trails in a period (including cases in which the human suddenly stops). Then the robot should plan a new path to avoid the calculated bounding sphere rather than the human during that period. When the human suddenly accelerated ($V_{\text{H}} > V_{\text{H_{danger}}}$ m/s), the system should respond as in case 1.

Case 3: The human is immobile. When the human stands near the robot, the system judges whether the human impedes the robot’s movements. If so, the system plans a new path using DRM to avoid him or her. In this case, the robot does not need to avoid a bounding sphere. Instead, the system can plan a shorter, more efficient path. When the human suddenly moves at the speed $V_{\text{H}} > V_{\text{H_{danger}}}$ m/s, the system should respond as in case 1. Otherwise, the system should respond as in case 2 when the speed $0 < V_{\text{H}} < V_{\text{H_{danger}}}$ m/s.

The rules can be expressed as follows:

Rule 1: IF ($V_{\text{H}} > V_{\text{H_{danger}}}$) & ($D_{HR} < D_{HR_{\text{min}}}$), THEN $P_R = P_R + (V_{\text{H_{danger}}} - \dot{V}_{\text{H}})$.

Rule 2: IF ($0 < V_{\text{H}} < V_{\text{H_{danger}}}$) & ($D_{HR} < D_{HR_{\text{min}}}$), THEN $P_R = P_{\text{New, Bounding}}$.

Rule 3: IF ($V_{\text{H}} = 0$) & ($D_{HR} < D_{HR_{\text{min}}}$), THEN $P_R = P_{\text{New, Static}}$.

where $P_R$ is the next position of the robot. $P_{\text{New, Bounding}}$ is the new plan calculated by DRM. With this plan, the robot can avoid the bounding sphere. $P_{\text{New, Static}}$ is the new plan calculated by DRM, with which the robot can avoid the human.

**IX. EXPERIMENT**

**A. EXPERIMENTAL ENVIRONMENT**

A series of experiments were performed to assess the effectiveness of the proposed method. The time requirement and efficiency of the proposed method were compared to those of previous methods [19] and [21]. During the experiments, the proposed active collision avoidance system was executed on an eight-core CPU, of which four were for the visualization and the robot movement control and the other four were for the complex position calculation and planning path.

In the proposed method, a GOOGOL GRB3016 robot was used to finish the experiments. The link parameters in D-H model of the robot are listed in Table 1. In the robot control system, a working path was designed, and the robot was made to move along this path repetitively by using the reverse kinematics algorithm [37]. An emulation scene including the robot manipulator model and the human model was built into the robot control system. In order to locate the 3D positions of the human and the robot, a position measurement system with a Kinect and a calibration board was designed. The calibration board was tightly attached to the robot near its base and the Kinect was firmly fixed to a tripod, which was placed 2.1 meters away horizontally, 1.6 meters away vertically and 1.6 meters height relative to the robot base (see Fig. 6). The resolution of the Kinect depth images was 640 × 480 and the capture frequency was 30 Hz. In order to determine the position of the robot and the human, the angle not need to avoid a bounding sphere. Instead, the system can plan a shorter, more efficient path. When the human suddenly moves at the speed $v_{H-} > v_{H_{\text{danger}}}$ m/s, the system should respond as in case 1. Otherwise, the system should respond as in case 2 when the speed $0 < v_{H} < v_{H_{\text{danger}}}$ m/s.

The rules can be expressed as follows:

Rule 1: IF ($v_{H-} > v_{H_{\text{danger}}}$) & ($D_{HR} < D_{HR_{\text{min}}}$), THEN $P_R = P_R + (v_{H_{\text{danger}}} - v_{H})$.

Rule 2: IF ($0 < v_{H} < v_{H_{\text{danger}}}$) & ($D_{HR} < D_{HR_{\text{min}}}$), THEN $P_R = P_{\text{New, Bounding}}$.

Rule 3: IF ($v_{H} = 0$) & ($D_{HR} < D_{HR_{\text{min}}}$), THEN $P_R = P_{\text{New, Static}}$.

TABLE 1. Link parameters in D-H model for the GOOGOL GRB3016 robot.

| Joints | D-H | $a$(mm) | $\alpha$(rad) | $d$(mm) | $\theta$(rad) |
|--------|-----|---------|---------------|---------|---------------|
| 1      | 2   | 150     | –\pi/2       | 250     | 0             |
| 2      | 3   | 570     | –\pi         | 0       | –\pi/2        |
| 3      | 4   | 150     | \pi/2        | 0       | 0             |
| 5      | 6   | 0       | –\pi/2       | 650     | 0             |
of the Kinect was adjusted before the experiments, so that the human skeleton and the calibration board can be captured in real-time. Suppose that \((x_k, y_k, z_k)\), \((x_b, y_b, z_b)\) and \((x_c, y_c, z_c)\) are the frames of the Kinect, robot base and the calibration board respectively. The relative location between the robot base frame \(X_bY_bZ_b\) and the calibration target frame \(X_cY_cZ_c\) could be measured offline using a ruler. A method described in previous work [38] is used to calculate the relative location between the calibration target frame \(X_cY_cZ_c\) and the Kinect frame \(X_kY_kZ_k\), with respect to the robot base frame \(X_bY_bZ_b\), could be determined using a calibration target (Fig. 7). The transformation from the robot base to the calibration board should be measured and determined before any experiments begin. During the experiments, the positions of the human were obtained by the depth information from the Kinect. By capturing the images of the calibration board with the RGB camera in the Kinect, the position of the robot could be determined by detecting the corners of the calibration board.

The goal is to allow the robot to avoid the human who has entered its workspace. In the experiments, to ensure the safety of the human, the robot velocity control is introduced, and the movement speed of the robot is determined based on the distance between the robot and the human. When the distance is less than the minimum distance \(S_{min}\), the robot would stop working. Moreover, in order to further protect the human from the accidental injuries in the event of an experimental failure in the experiments, we set the force threshold to 50 Newton which is less than the maximum permissible forces of various part of the body as illustrated in the Collaborative Robot Collision Standard ISO/TS 15066. When the detected force is greater than the set value, the robot stops working. In this way, when incidental contact between robots and humans occurs, contact shall not result in pain or injury. The parameters used for the experiment were as follows: \(v_{H_{danger}} = 0.2 \text{ m/s}, D_{HR_{min}} = 0.2 \text{ m}\).

During the experiments, the robot moved all the way along the designed path and the position of the robot EE with respect to the Kinect frame was calculated by the robot control system in real-time. When a human enters its workspace, the skeleton of the person was detected immediately by the Kinect. With the captured human skeleton, the position of the human related to the Kinect frame and the speed at which the human moves could be determined by the robot control system. Once the system determined that the human was approaching the robot, the active collision-avoidance system performed collision avoidance according to the speed of the human’s movements. Five testers were invited to carry out the experiments to verify the efficiency of the proposed method. The testers are aged from 20 to 26, four males and one female. Note that the testers were asked to approach the robot at different speeds. Firstly, the testers moved toward the robot at a relatively high speed which was more than 0.2 \text{ m/s}. Then, the testers approached the robot at a speed between 0 and 0.2 \text{ m/s}. Finally, the tester moves away from the robot. Each experiment was repeated three times, for a total of 15 times.

### B. EXPERIMENTAL RESULTS

In the experiment, the testers attempted to interrupt the robot’s work. Fig. 8 (a) shows the experimental scene. In order to monitor the robot’s environment, a three-dimensional virtual scene was constructed as shown in Fig. 8 (b). During the initialization of the virtual scene, there was only one virtual robot. When the Kinect sensor detected a human, the sensor calculated the position of his or her skeleton in real environment and then a 3D skeleton and a 2D skeleton were drawn in the virtual scene (Fig. 8 (b)) and on the 2D image (Fig. 8 (c)), respectively.

Fig. 9 shows the experimental procedure. The human tried to approach the robot, then broke into its planned path, and finally left it. The avoidance component of the robot was
equal to the approach component of the human, as shown in Fig. 10. The pink dotted line is the movement trajectory of the human, the blue solid line is the new path of the robot, and the red line is the old path of the robot. In the beginning, as the human was approaching the robot quickly, the system found that his or her speed exceeded $v_{H,\text{danger}}$. Then the robot tried to avoid the human directly. When the human slowed down, the system calculated a bounding sphere to plan a new path for the robot.

In the experiment, the proposed method was compared with method [19] and method [21]. When a human is approaching the robot at a faster speed, the robot moved to avoid him directly, while in these previous methods, the robot realizes obstacle avoidance by repeatedly planning a new path. When the speed of the human slows down, the robot avoids him or her by avoiding the calculated boundary sphere, while in methods [19] and [21], the robot still avoids humans by repeatedly planning new paths. Fig. 12 shows the comparison of the proposed method and other methods in the robot moving trajectory.

Fig. 13 shows the comparative results in X, Y, Z. Compared with methods [19] and [21], in the X direction, it is obvious that the path of the proposed method is most close to the default trajectory, which is nearly the same as the default trajectory. And in the Y and Z direction, though three methods’ paths all have clear differences with the default trajectory from 7th sec to the 10th sec, we can find that the difference

![Figure 9](image1.png)  
**FIGURE 9.** Experimental procedure: (a) The human approached the robot. (b) The human broke into the planned path of the robot. (c) The robot avoided human. (d) The human left.

![Figure 10](image2.png)  
**FIGURE 10.** Three-dimensional trajectory.

![Figure 11](image3.png)  
**FIGURE 11.** Velocity curve of the human and robot in the direction of the connections between the human and the robot.

![Figure 12](image4.png)  
**FIGURE 12.** Robot moving trajectory.

![Figure 13](image5.png)  
**FIGURE 13.** Comparative results in X, Y, Z.

3.95 s ($P_{R2}$). Then, when the human began to move away from the robot, the robot returned to its previous path ($P_{R3}$).

In the experiment, the proposed method was compared with method [19] and method [21]. When a human is approaching the robot at a faster speed, the robot moved to avoid him directly, while in these previous methods, the robot realizes obstacle avoidance by repeatedly planning a new path. When the speed of the human slows down, the robot avoids him or her by avoiding the calculated boundary sphere, while in methods [19] and [21], the robot still avoids humans by repeatedly planning new paths. Fig. 12 shows the comparison of the proposed method and other methods in the robot moving trajectory.
between the proposed method and default path is minimum. In general, the path of the proposed method is most close to the default trajectory in a three-dimension space. It is because that the proposed method can correctly respond to different human movement speeds by analyzing the human’s behaviors and choose the best path for the robot so that the flexibility and intelligence of the robot movement can be improved.

Fig. 14 shows the comparisons of the proposed method, method [19] and method [21] in terms of the avoidance time, the number of avoidance events and the increased route. Table 2 lists the statistics for the data in Fig. 14, including the average, minimum and maximum values of the five testers using the three methods. In the proposed method, the system had to move to avoid the human 3.4 times on average, which collectively took 4.76s on average. In the method [19], because the system planned a new path for the robot repeatedly, the robot had to move 8.4 times on average to avoid the human and spent more time doing it, at an average of 11.52s. In the method [21], the average number of avoidance events and average avoidance time are 5.2 times and 6.72s, respectively. Because the robot moved in a new path to avoid the human, the route was longer than the initial one. In the current method, the path is increased by an average of 783.2 mm, and the paths that were built by method [19] and method [21] are 1173.2 mm and 937.4 mm, respectively.

Moreover, a serial of statistical tests is conducted. The statistical test data of the paired t-test are shown in Table 3. All the two-tail p-values are less than alpha level (0.05), which indicated that the proposed method has better performance than method [19] and method [21] significantly in our experiments.

### Table 2. Comparisons of the proposed method, method [19] and method [21].

|                | Avoidance time (s) | Number of avoidance events | Increased route (mm) |
|----------------|--------------------|-----------------------------|----------------------|
| Proposed method| Minimum 4.4        | 2                           | 757                  |
|                | Maximum 5.3        | 4                           | 808                  |
|                | Average 4.7        | 3.3                         | 785.3                |
| Method [19]    | Minimum 10.3       | 7                           | 1102                 |
|                | Maximum 12.5       | 10                          | 1263                 |
|                | Average 11.5       | 8.4                         | 1175.1               |
| Method [21]    | Minimum 6.1        | 4                           | 870                  |
|                | Maximum 7.4        | 7                           | 1008                 |
|                | Average 6.7        | 5.5                         | 935.7                |

### Table 3. Statistical test of the proposed method, method [19] and method [21].

|                | Two-tail p-value   | Avoidance time (s) | Number of avoidance events | Increased route (mm) |
|----------------|--------------------|--------------------|-----------------------------|----------------------|
| Proposed method vs Method [19] | 6.24344E-14 | 2.72322E-09 | 1.24749E-13 |
| Proposed method vs Method [21] | 1.08384E-12 | 3.27587E-07 | 3.20317E-09 |

When the human slowed down, the proposed method calculated a bounding sphere for the robot so that it could avoid the human quickly, which saved a considerable amount of
time. However, the previous methods constantly calculated a new path for the robot so that the robot could move around the human. Although the increase in the length of the route was less pronounced using the previous methods, avoidance time and avoidance events increased significantly. It can be concluded that the proposed rule-based logic system can formulate reasonable response strategies according to different behaviors of people. In addition, the establishment of a cylindrical bounding box model for human bones and robots helps to improve the timeliness and effectiveness of the collision avoidance system.

X. DISCUSSIONS

For the sake of safe interaction, human beings and robots are usually required to work in separate spaces. In current industrial applications that lack sensor surveillance, robot work cells must be static. If a human enters the robot’s work cell, some additional strategies must be used to ensure the safety of both the human and the robot. Existing applications for collaboration between robots and humans are relatively common. These applications allow robots to work in less space in situations in which there are no physical barriers or in which no humans are involved. In complex environments, humans may be mistaken for operation objects, like workpieces. Robots’ lack of independent intelligence can place humans in danger. This paper uses Kinect to detect humans who entered the robot’s workspace and the IPF is applied to improve the accuracy of estimating the skeleton position. By building bounding cylinders for the human and the robot, the collision detection between the human and the robot can be achieved by detecting the relative position of the cylinders, which can improve the effectiveness and accuracy of the system. In order to make a reasonable reaction strategy for the obstacle avoidance system based on human behaviors, a rule-based logic system is proposed, and DRM is applied to plan a new path to avoid humans. When humans are immobile, DRM is used to plan a new path for the robot directly. When the humans are moving slowly, a bounding sphere is calculated, and the robot bypasses it by planning a new path using DRM. When the human is moving at a fast pace, the robot stops working immediately. By taking different measures, the system was found to actively protect humans.

In future work, voice control will be added to the system so that the robot can talk to any human who enters its workspace. The robot can be made more clearly aware of humans’ intentions. This may make avoidance more efficient.

XI. CONCLUSION

This paper proposes an active collision avoidance method for robots using the Kinect sensor and the IPF. This proposed method can effectively detect human breaking into the workspace of the robots. In the proposed method, the collision detection is achieved by detecting the relative position of the bounding cylinders of the human and the robot. To take different reaction strategies according to human movement, a rule-based logic system was proposed to analyze the behavior of the human and DRM was used to plan a new path for the robot to avoid the human. Finally, experiments were performed to show the effectiveness of the proposed algorithm. Experimental results demonstrated that this proposed method could be applied to practical applications such as protecting the behaviors of humans who enter a robot’s workspace in a human-robot collaborative environment.

REFERENCES

[1] S. Robla-Gomez, V. M. Becerra, J. R. Llata, E. Gonzalez-Sarabia, C. Torre-Ferrero, and J. Perez-Oria, “Working together: A review on safe human-robot collaboration in industrial environments,” IEEE Access, vol. 5, p. 26754–26773, 2017.
[2] A. Vysocky and P. Novak, “Human-robot collaboration in industry,” MM Sci. J., vol. 9, no. 2, pp. 903–906, 2016.
[3] M. Bortolini, E. Ferrari, M. GAMBERI, F. PILATI, and M. Faccio, “Assembly system design in the industry 4.0 era: A general framework,” IFAC-PapersOnLine, vol. 50, no. 1, pp. 5700–5705, Jul. 2017.
[4] A. Corrales Paredes, M. Malfaz, and M. A. Salichs, “Signage system for the navigation of autonomous robots in indoor environments,” IEEE Trans Ind. Informat., vol. 10, no. 1, pp. 680–688, Feb. 2014.
[5] T. Nozaki, T. Mizoguchi, Y. Saito, D. Yashiro, and K. Ohnishi, “Recognition of grasping motion based on modal space haptic information using DP pattern-matching algorithm,” IEEE Trans Ind. Informat., vol. 9, no. 4, pp. 2043–2051, Nov. 2013.
[6] L. Wang, B. Schmidt, and A. Y. C. Nee, “Vision-guided active collision avoidance for human-robot collaborations,” Manuf. Lett., vol. 1, no. 1, pp. 5–8, Oct. 2013.
[7] K. Takemura, K. Takahashi, J. Takamatsu, and T. Ogasawara, “Estimating 3-D Point-Of-Regard in a real environment using a head-mounted eye-tracking system,” IEEE Trans. Human-Mach. Syst., vol. 44, no. 4, pp. 531–536, Aug. 2014.
[8] F. Flacco, T. Kröger, A. De Luca, and O. Khatib, “A depth space approach to human-robot collision avoidance,” in Proc. IEEE Int. Conf. Robot. Autom., Saint Paul, MN, USA, May 2012, pp. 338–345.
[9] C.-H. Chen, T.-K. Liu, and J.-H. Chou, “A novel crowding genetic algorithm and its applications to manufacturing robots,” IEEE Trans Ind. Informat., vol. 10, no. 3, pp. 1705–1716, Aug. 2014.
[10] X. Jiang, L.-K. Merhi, and C. Menon, “Force exertion affects grasp classification using force myography,” IEEE Trans. Human-Mach. Syst., vol. 48, no. 2, pp. 219–226, Apr. 2018.
[11] H. Zhang, Y. Wang, J. Zheng, and J. Yu, “Path planning of industrial robot based on improved RRT algorithm in complex environments,” IEEE Access, vol. 6, pp. 53296–53306, 2018.
[12] D. Han, H. Nie, J. Chen, and M. Chen, “Dynamic obstacle avoidance for manipulators using distance calculation and discrete detection,” Robot. Comput. Integr. Manuf., vol. 49, pp. 98–104, Feb. 2018.
[13] X. Wu and S. Ma, “Neurally controlled steering for collision-free behavior of a snake robot,” IEEE Trans. Control Syst. Technol., vol. 21, no. 6, pp. 2443–2449, Nov. 2013.
[14] A. Dietrich, T. Wimbock, A. Albu-Schaffer, and G. Hirzinger, “Integration of reactive, torque-based self-collision avoidance into a task hierarchy,” IEEE Trans. Robot., vol. 28, no. 6, pp. 1278–1293, Dec. 2012.
[15] Y. Hu, W. Zhao, and L. Wang, “Vision-based target tracking and collision avoidance for two autonomous robotic fish,” IEEE Trans. Ind. Electron., vol. 65, no. 5, pp. 1401–1410, May 2018.
[16] J. Krüger, B. Nickolay, P. Heyer, and G. Seliger, “Image based 3D surveillance for flexible Man-Robot-Cooperation,” CIRP Ann., vol. 54, no. 1, pp. 19–22, 2005.
[17] J. A. Corrales, F. A. Candelas, and F. Torres, “Safe human–robot interaction based on dynamic sphere-swept line bounding volumes,” Robot. Comput. Integr. Manuf., vol. 27, no. 1, pp. 177–185, Feb. 2011.
[18] Microsoft. (2014). Kinect for Windows. Accessed: Jun. 6, 2019. [Online]. Available: https://developer.microsoft.com/en-us/windows/kinect
[19] C. Morato, K. N. KIPS, B. Zhao, and S. K. Gupta, “Toward safe human robot collaboration by using multiple kinects based real-time human tracking,” J. Comput. Inf. Sci. Eng., vol. 14, no. 1, pp. 98–106, Mar. 2014.
[20] J. T. C. Tan and T. Arai, “Triple stereo vision system for safety monitoring of human-robot collaboration in cellular manufacturing,” in Proc. IEEE Int. Symp. Assem. Manuf. (ISAM), Tampere, Finland, May 2011.
REFERENCES

[21] P. Zhang, P. Jin, G. Du, and X. Liu, “Ensuring safety in human-robot coexisting environment based on two-level protection,” Ind. Robot. Int. J., vol. 43, no. 3, pp. 264–273, May 2016.

[22] M. Fischer and D. Henrich, “3D collision detection for industrial robots and unknown obstacles using multiple depth images,” in Advances in Robotics Research. Berlin, Germany: Springer, 2009, pp. 111–122.

[23] A. Mohammed, B. Schmidt, and L. Wang, “Active collision avoidance for human–robot collaboration driven by vision sensors,” Int. J. Comput. Intell. Manuf., vol. 30, no. 9, pp. 970–980, 2016.

[24] R. Ahmad and P. Plapper, “Safe and automated assembly process using vision assisted robot manipulator,” Procedia CIRP, vol. 41, pp. 771–776, 2016.

[25] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, “Real-time human pose recognition in parts from single depth images,” Commun. ACM, vol. 56, no. 1, pp. 116–124, Jan. 2013.

[26] V. Villena-Martínez, A. Fuster-Guilló, J. Azorín-López, M. Saval-Calvo, J. Mora-Pascual, J. García-Rodríguez, and A. García-Garcia, “A quantitative comparison of calibration methods for RGB-D sensors using different technologies,” Sensors, vol. 17, no. 2, p. 243, 2017.

[27] R. He, Y. Zhao, S. Yang, and S. Yang, “Kinematic-parameter identification for serial-robot calibration based on POE formula,” IEEE Trans. Robot., vol. 26, no. 3, pp. 411–423, Jun. 2010.

[28] S. Cheng and Y. Feng, “Fast collision detection algorithm of cylinders based on generatrices,” J. Jilin Univ., Sci. Ed., vol. 53, no. 2, pp. 291–296, Feb. 2015.

[29] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” IEEE Trans. Signal Process., vol. 50, no. 2, pp. 174–188, 2002.

[30] H. Bi, J. Ma, and F. Wang, “An improved particle filter algorithm based on ensemble Kalman filter and Markov chain Monte Carlo method,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 6, no. 2, pp. 447–459, Feb. 2015.

[31] G. Evensen, “Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics,” J. Geophys. Res., vol. 99, no. C5, pp. 10143–10162, 1994.

[32] C. Andrieu, A. Doucet, and R. Holenstein, “Particle Markov chain Monte Carlo methods,” J. Roy. Stat. Soc. B. Stat. Methodol., vol. 72, no. 3, pp. 269–342, Jun. 2010.

[33] S.-H.-P. Won, W. W. Melek, and F. Golnaraghi, “A Kalman/particle filter-based position and orientation estimation method using a position sensor/inertial measurement unit hybrid system,” IEEE Trans. Ind. Electron., vol. 57, no. 5, pp. 1787–1798, May 2010.

[34] P. Leven and S. Hutchinson, “A framework for real-time path planning in changing environments,” Int. J. Robot. Res., vol. 21, no. 12, pp. 999–1030, Dec. 2002.

[35] T. Kunz, U. Reiser, M. Stilman, and A. Verl, “Real-time path planning for a robot arm in changing environments,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Taipei, Taiwan, Oct. 2010, pp. 18–22.

[36] H. Shin, K. Seo, and S. Rhim, “Allowable maximum safe velocity control on human-robot distance for collaborative robot,” in Proc. 15th Int. Conf. Ubiquitous Robots (UR), Honolulu, HI, USA, Jun. 2018.

[37] G. Antonelli, S. Chiaverini, and G. Fusco, “A new on-line algorithm for inverse kinematics of robot manipulators ensuring path tracking capability under joint limits,” IEEE Trans. Robot. Autom., vol. 19, no. 1, pp. 162–167, Feb. 2003.

[38] G. Du and P. Zhang, “Online robot calibration based on vision measurement,” Robot. Comput.-Integr. Manuf., vol. 29, no. 6, pp. 484–492, Dec. 2013.

GUANGLONG DU received the Ph.D. degree in computer application technology from the South China University of Technology, Guangzhou, China, in 2013. He is currently an Associate Professor with the School of Computer Science and Engineering, South China University of Technology. His research interests include intelligent robotics, human–computer interaction, artificial intelligence, and machine vision.

YINHAO LIANG received the B.S. degree from the Guangdong University of Technology. He is currently pursuing the master’s degree with the School of Computer Science and Engineering, South China University of Technology. His research interests include robot teaching, machine vision, and human–robot interaction.

GENGCHENG YAO received the B.S. degree in mechanical and electronic engineering from the South China University of Technology, Guangzhou, China, in 2019, where he is currently pursuing the M.S. degree in computer science. His research interest includes the analysis of virtual reality and human–robot interaction.

CHUNQUAN LI received the B.Sc., M.Sc., and Ph.D. degrees from Nanchang University, Nanchang, China, in 2002, 2007, and 2015, respectively.

He has been with the School of Information Engineering, Nanchang University, since 2002, where he is currently a Professor and a Young Scholar of Ganjiang River. He has published over 30 research articles. He is also a Visiting Professor with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada. His current interests include computing intelligence, haptics, virtual surgery simulation, and robotics and their applications to biomedical engineering.

RONIGUES J. MURAT received the B.S. degree in electronic and information engineering from the Dalian University of Technology, in 2019. He is currently pursuing the M.E. degree in computer science with the South China University of Technology. His research interests include intelligent robotics, artificial intelligence, and machine vision.

HUA YUAN received the B.S. degree from Harbin Engineering University, Harbin, China, and the M.S. and Ph.D. degrees from Sichuan University, Chengdu, China. She is currently an Associate Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. Her research interests include image processing, video communication, big data processing, and the next-generation network architecture.