Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-2 (2022) 307–312

A Radial Basis Functions approach to collision avoidance in collaborative tasks

G. Cipriani * M. Bottin * G. Rosati * M. Faccio **

* *University of Padova, Department of Industrial Engineering, Padova, Italy.*
** *University of Padova, Department of Management and Engineering, Vicenza, Italy.*

Abstract: In the last years, collaborative human-robot applications have become more and more appealing thanks to the robot's easiness of programming and the promise of increasing precision and safety. However, by combining two resources (the cobot and the human operator) there is a problem of safety since cobot and human operator have to work in the same workspace. To ensure human safety, the distance between robot and operator must be assessed and the robot must adapt accordingly either by reducing its velocity or by modifying its trajectory.
In this paper, we propose a new online method to adapt the trajectory of the robot to the human movements using a single depth camera. This algorithm eliminates the robot from the scene using a simple calibration process. Then, it interpolates the shared workspace, captured by the depth camera, using Radial Basis Functions (RBFs). The result is a continuous function that is representative of the risk of collision with obstacles on the plane. Its gradient is used as a repulsive potential in the Artificial Potential Field (APF) method to generate the path. This method eliminates the need to calculate the distance between operator and robot since it is intrinsically considered in the potentials. Results shows the validity of the method.

Keywords: Intelligent robotics, cobots, collision avoidance, human-robot collaboration, safety.

1. INTRODUCTION

Nowadays, the demand for high product variety drives the manufacturing market and requests for higher flexibility in production. One trend born to accomplish this task is the human-robot collaboration [IFR (2020)]. The robot is not constrained inside a fenced cage, but it works actively and responsively with the human operator, sharing its workspace. Collaboration can take advantage both of the dexterity and flexibility of human operators, and the repeatability, speed, accuracy, and strength of robots [Matheson et al. (2019)]. As a result, higher agility is achieved, and complex tasks can be completed with higher productivity and lower costs. However, this comes with a decreased throughput when compared to a traditional robotic system. Fortunately, a careful allocation of tasks and increased flexibility can make this aspect negligible [Faccio et al. (2019a), Faccio et al. (2019b)].

Even if the collaboration can be achieved with traditional robots, dedicated robots, called cobots, have been designed. Indeed, they are equipped with sensors and procedures to detect and prevent impacts [Villani et al. (2018)]. Furthermore, they can exhibit mobility through their installation over mobile robots or their ease of installation. In this way, they can be used to assist inexperienced operators or substitute ill workers [Cohen et al. (2021)].

However, the collaboration has some downsides regarding the safety of the operator. Indeed, one of the current trends among researchers is about developing algorithms and devices to protect the operator from the risk of collisions with the robot [Villani et al. (2018), Robla-Gómez et al. (2017)]. Moreover, nowadays, most collaborative tasks involve either coexistence or sequential collaboration to reduce intrinsically that risk [IFR (2020)].

This need arises because the robot is not able to perceive the environment in which it moves. Thus, approaches to increase its perception are fundamental to exploit the benefits of collaboration through obstacle avoidance, predictive control, and task recognition [Antão et al. (2019)]. For example, a robot should be able to distinguish between human interferences due to trajectory error or time sequence error [Boschetti et al. (2021)]. In the first case, the robot can decide whether the interference can cause a collision or not. In the second case, it should redefine its task and warn the operator of his poor performance.

In general, the approaches to collision detection and collision avoidance regard the use of force/torque sensors, vision systems [Robla-Gómez et al. (2017)], or dedicated end-effectors [Tommasino et al. (2021), Kim et al. (2013)].

The different solutions adopted depend on the modality of the collaboration. For example, in [Mariotti et al. (2019)], authors focused on the so-called "Hand Guiding" mode making the most of the estimated forces given by the controller of the robot and the data from a force/torque sensor attached to the end-effector. These forces are used to detect impacts. Furthermore, they can distinguish between intentional and accidental collisions adapting the response to the stimulus.

10.1016/j.ifacol.2022.04.211

In other modes, a calculation of the distance between the robot and moving obstacles (such as operators) must be assessed [Secil and Ozkan (2022)]. In general, this is performed using vision systems and algorithms optimized to guarantee real-time capabilities. The detection of the human operator is achieved using complex devices, such as infrared cameras and reflective spheres [Amorim et al. (2021)], or RGB-D cameras coupled with dedicated tracking algorithms [Mauro et al. (2018)]. The algorithms measure the distance between robot and operator and decide whether to reduce velocities and stop [Magrini et al. (2020)] or modify the trajectory [Flacco et al. (2012), Scimmi et al. (2021), Bottin et al. (2021b)]. Depth cameras are extensively used for their simplicity and reduced cost.

In this work, we introduce a new algorithm that can simplify the detection of human interferences in the collaboration space by combining Radial Basis Functions (RBFs) and Artificial Potential Field (APF) method. Unlike other methods, this algorithm makes use of a single depth camera to view the scene from above. Then, the scene is elaborated to remove the robot, and interpolated using RBFs to obtain a continuous function. Finally, APF method is adopted to find the trajectory and modify it if operators interfere in the scene during the execution of the task. Such an algorithm can be used in conjunction with a task planning algorithm [Bottin et al. (2021a)].

The paper is organized as follows. In Section 2, the global RBFs will be described in all its components. In Section 3, the theory is applied in a use case to demonstrate its applicability. Finally, conclusions are drawn in Section 4.

2. POTENTIAL FUNCTION

The method used to modify trajectories derived from the APF method used extensively in the field of path planning [Warren (1989)]. The potentials are extrapolated directly from the interpolation of the environment with Radial Basis Functions, first introduced by [Strack and Janković (1999)] and used in many fields of application during the years [Buhmann (2000)]. They can be used to represent a measure of the risk of collisions inside the environment in which the robot moves, for example interpolating the data coming from depth cameras. Indeed, the interpolation of depth images causes the obstacles, such as human operators, to be the maximums of the interpolated functions. Thus, such interpolation can be used so that the robot will deviate while following the gradient of the function. In such way, the robot can avoid the operator without the need of feature recognition.

2.1 Radial Basis Functions

Radial Basis Function (RBF) method is widely used to approximate multivariate functions since it can handle high dimensional problems with scattered data [Buhmann (2000)].

This property made RBFs suitable as neural network models [Broomhead and Lowe (1988)]. In particular, their introduction in this field was driven by their simplicity in modeling nonlinear relationships using a linear combination of the weights of the variables. They are defined as a combination of functions [Buhmann (2000)]:

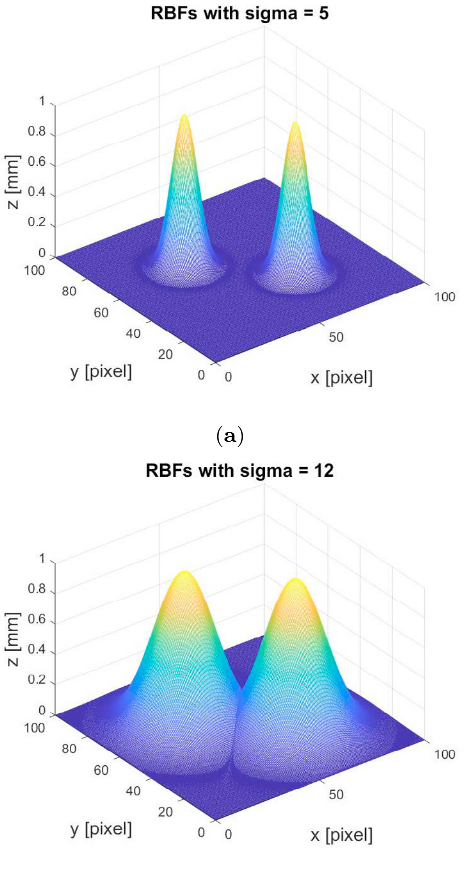
$$s(x) = \sum_{\xi \in \Xi} \lambda_\xi \phi(\|\mathbf{x} - \xi\|) \qquad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$, λ_ξ represents the coefficients of the combination, $\phi(\|\mathbf{x} - \xi\|)$ is the basis and Ξ is the set of distinct points used as centers for the RBFs.

In this work, the Radial Basis Functions are built using the Gaussian function. Thus, the basis is described by the following equation:

$$\phi(\|\mathbf{x} - \xi\|) = e^{-\frac{\|\mathbf{x}-\xi\|}{2\sigma^2}} \qquad (2)$$

where $\mathbf{x}$ are the coordinates (x, y) of the points in which evaluate the function and ξ are the coordinates of the centers of the Gaussian functions. σ is canonically the standard deviation. In this case, σ represents a measure of the influence of a single Gaussian over the others, as can be seen in Figure 1. In fact, the higher σ, the wider the spread of the Gaussian function.



(**a**)

(**b**)

Fig. 1. Mutual influence of RBFs depending on σ: (**a**) $\sigma = 5$, (**b**) $\sigma = 12$.

Since the function is constituted of a linear combination of the variables, the weights can be estimated using the least square method. Considering the following simplification:

$$G_i = \phi(\|\mathbf{x} - \xi\|) \qquad (3)$$

the weights are obtained with Equation 4

$$\lambda_\xi = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T f(x,y) = \mathbf{G}^+ f(x,y) \tag{4}$$

where $\mathbf{G}^+$ is the pseudoinverse of the matrix $\mathbf{G}$, and $f(x,y)$ contains the depth image data. The calculation of $\mathbf{G}^+$ is time-consuming, but it can be performed in advance. Thus, weights estimation is reduced to a simple and fast matrix multiplication.

The value of σ is arbitrary; thus, tests were carried on fitting different images to find the best value for σ. For an equally spaced grid of RBFs, the best interpolation results have been achieved when:

$$\sigma = \frac{D_{\xi_i\xi_j}}{2} \tag{5}$$

where $D_{\xi_i\xi_j}$ is the distance between the Gaussian centers.

Finally the interpolated image $f^*(x,y)$ is given by:

$$f^*(x,y) = \mathbf{G} \cdot \lambda_\xi \tag{6}$$

In [Carr et al. (2001)], authors take advantage of RBFs to fit point-clouds representing 3D objects. They obtained smooth surfaces and were also able to repair incomplete meshes. Similarly, in this work, we used RBFs to approximate depth images data to obtain a more regular environment where searching for paths. In Figure 2a, a depth image of the robot is shown. This image was interpolated using a 15 × 15 grid of evenly distributed RBFs on the plane. As can be seen in Figure 2b, the resulting function is regular and continuous. Such function is easier to work with and can be easily added to other functions. Moreover, the calculation of the gradient is straightforward since it is not necessary to calculate the gradient directly from the final function. From Equation 1, the gradient is given by:

$$\frac{\delta f(x,y)}{\delta x} = \frac{\delta \mathbf{G}}{\delta x}\lambda_\xi \tag{7}$$

$$\frac{\delta f(x,y)}{\delta y} = \frac{\delta \mathbf{G}}{\delta y}\lambda_\xi \tag{8}$$

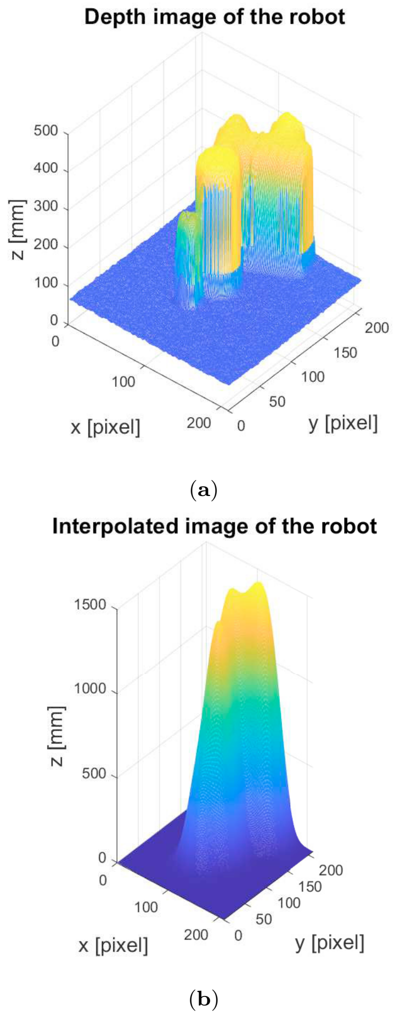
Thus, it is only necessary to know the gradient formulation of a simple gaussian and to calculate the weights with Equation 4. This simplicity makes it useful for the APF application exposed in the next section where gradients are used to modify the trajectory.

2.2 *Robot removal from depth images*

Independently of the vision system adopted, the robot will always appear in the images. Most of the works track the robot or use its kinematics to understand where it is to calculate its distance from the operator [Secil and Ozkan (2022), Mauro et al. (2018)].

In [Magrini et al. (2020)], the robot is removed from the scene using its augmented 3D model to eliminate possible disturbances or false alarms. As a result, the processed depth image data contains only obstacles.

Following [Magrini et al. (2020)], the captured depth image is processed by means of a calibration method: the robot moves on the plane following a grid disposed all over the collaborative space. In the meanwhile, the camera takes snapshots of its movements, as in Figure 2a. The position of the robot at every snapshot is read from the robot controller. After this process, the images are interpolated using RBFs as shown in Figure 2b. The interpolation generates an augmented representation of the robot that is useful to be more conservative during the elimination phase. Then, the images are binarized, and the MATLAB function *regionprops* is used to find the pixels occupied by the robot. Finally, a matrix is populated with the image's pixels that must be deleted at every position of the robot in the collaboration space. Such calibration method is easy to implement and is convenient when a reliable CAD model of the robot is not available. It is worth to notice that by using a CAD model the calibration could be performed more accurately.

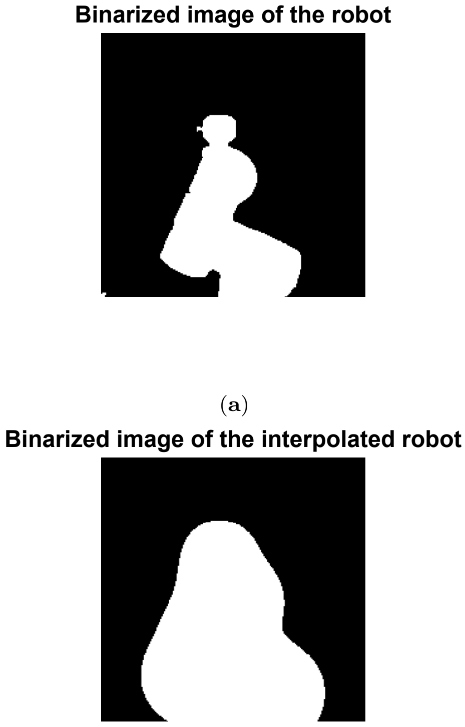


(**a**)

(**b**)

Fig. 2. Example of interpolation of the depth image of the robot using RBFs: (**a**) original depth image filtered to reduce noise, (**b**) interpolated function using 15x15 RBFs grid.

Figures 3a and 3b show the binarized images. The threshold for the binarization is selected experimentally searching for a good approximation of the robot. The threshold corresponds to the height at which the interpolated function is cut. Therefore, higher thresholds mean greater heights for the cut, and the final result is nearer to the original image. This iterative process is simplified since the


Binarized image of the robot
Binarized image of the interpolated robot

(a)

(b)

Fig. 3. Example of binarization: (**a**) original depth image, (**b**) interpolated image with a threshold of 100 mm.

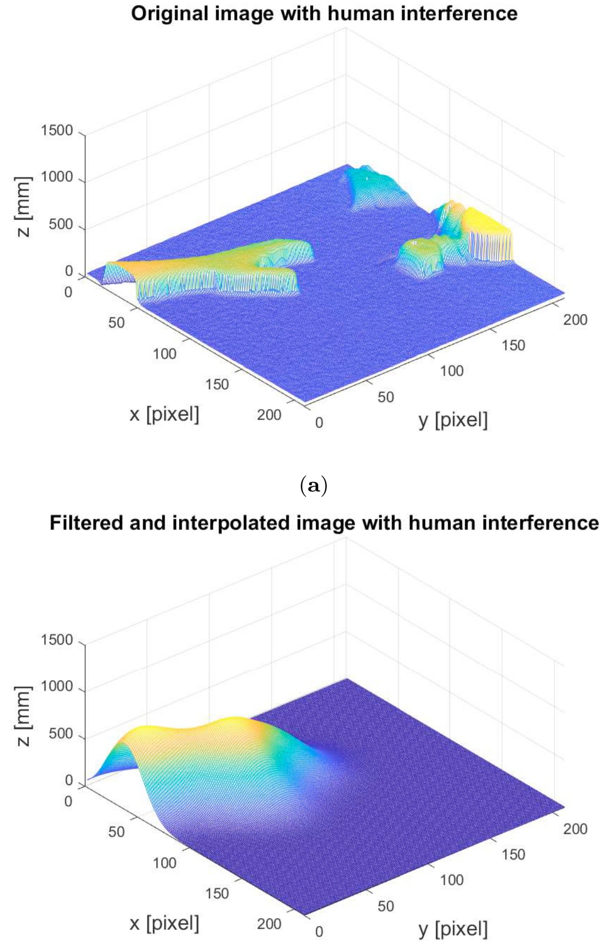
threshold selected for one position shows good consistency for all the other positions of the robot. It is clear how the interpolation augments the size of the robot. Furthermore, the augmentation simplify the calibration process, since it is not necessary to have a snapshot of the robot in every single point of the plane.

In Figure 4, the result of the filtering is proposed. As can be seen in Figure 4a, the operator approaches the collaboration space. The image shown in 4b is the one used for future operations.

3. TRAJECTORY GENERATION

In this phase, we make the assumption that robot and operator face each other during the task. This is typical if we consider the case of limited collaborative space. In such a way, all the possible collisions between the arm of the operator and the robot links are avoided. Thus, only the wrist is expected to collide with the operator. Future work will address this limitation to expand the model.

After removing the robot, the filtered image can be analyzed to find the optimal path for the robot. The method implemented is based on the Artificial Potential Field (APF) method. Thus, an attractive force drives the robot to the target, and a repulsive one makes it divert to avoid obstacles.



(a)

(b)

Fig. 4. Example of filtering: (**a**) original depth image, with the operator hand on the left and the robot on the right; (**b**) filtered and interpolated image, where the robot has been completely removed.

The attractive force is constant and is formulated as [Siciliano et al. (2008)]:

$$F_a = k_a \frac{\mathbf{e}(\mathbf{p})}{\|\mathbf{e}(\mathbf{p})\|} \tag{9}$$

where k_a is a constant, and $\mathbf{e}(\mathbf{p}) = \mathbf{p}_{goal} - \mathbf{p}$ is the distance between the current position of the robot ($\mathbf{p}$) and the target ($\mathbf{p}_{goal}$).

The repulsive force is derived from the gradient of the interpolated image. Moreover, the gradient can be exploited to create a vortex field. This potential forces the robot to go around the obstacle (i.e., the operator) improving the final path, and limiting the risk of being stuck in local minima.

The repulsive force is defined as:

$$F_r = \pm k_r \begin{bmatrix} \dfrac{\delta f^*}{\delta y} \\ -\dfrac{\delta f^*}{\delta x} \end{bmatrix} \tag{10}$$

where the sign decides the direction of rotation, k_r is a constant, and f^* is the result of the depth image interpolation and filtering to remove the robot.

It is worth noticing that since the operator comes from the outside of the space, only one of the two directions of the vortex (clockwise or counterclockwise) can be used. The right direction depends strictly on the direction of motion of the robot, as can be seen in the example in Figure 5.

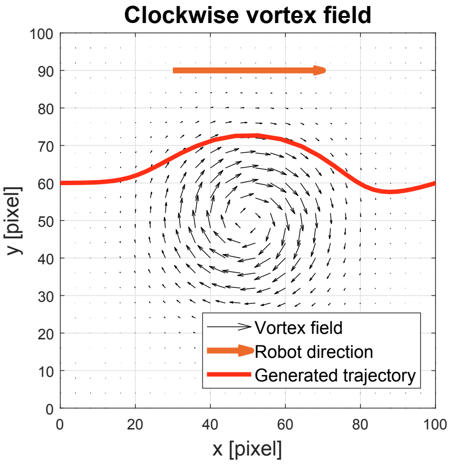


Fig. 5. Example of a trajectory generated using APF and vortexes.

Finally, the total force acting on the robot is :

$$F_t(\mathbf{p}) = F_a(\mathbf{p}) + F_r(\mathbf{p}) \tag{11}$$

This force can be used in different ways to modify the trajectory [Siciliano et al. (2008)]. In this work, it is used as cartesian speed:

$$\dot{\mathbf{v}} = F_t(\mathbf{p}) \tag{12}$$

The path is generated using the gradient descent method:

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \dot{\mathbf{v}}\Delta t \tag{13}$$

where Δt is the time between two path computations.

In Figure 6, a typical path generated is showed. It is clear how the robot is able to divert from the straight trajectory to compensate for human interferences in the scene.

To modify the maximum distance from the operator, the path can be adjusted in two ways:

(1) tuning the coefficients of the potentials;
(2) increasing the σ of the RBFs (without modifying the weights).

The second way shows another advantage of the use of RBFs. Once the weights have been calculated, they can be assigned to another grid of RBFs to generate a function that is always a good approximation of the system, but can have different σ. The final result is tighter or wider than the original as shown in Figure 7.

Finally, a fundamental parameter analyzed to ensure that the algorithm has adequate capabilities for online control is the time necessary to compute a new command for the robot. This parameter was measured experimentally using a Techman TM5-700 and a Kinect camera. MATLAB is used to control both. The images were interpolated using a 15x15 Gaussian grid. The tests returned a mean total time of $0.0806 \pm 0.0080s$, of which the algorithm used only $0.0223 \pm 0.0041s$ to calculate the new pose, and the rest to take the image from the camera and the actual position from the robot. Thus, this particular system can perform online with an update frequency of 10 Hz.

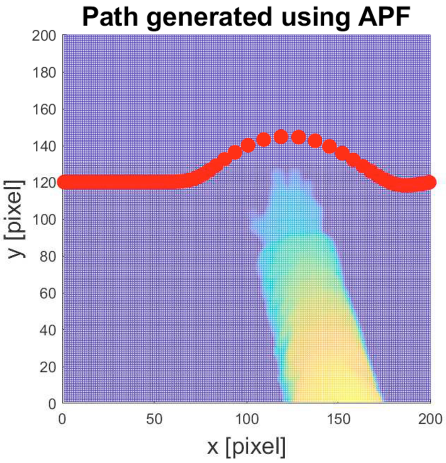


Fig. 6. Path generated with APF method showing the deviation in correspondence to human interference. The red dots represent the robot tool center point.

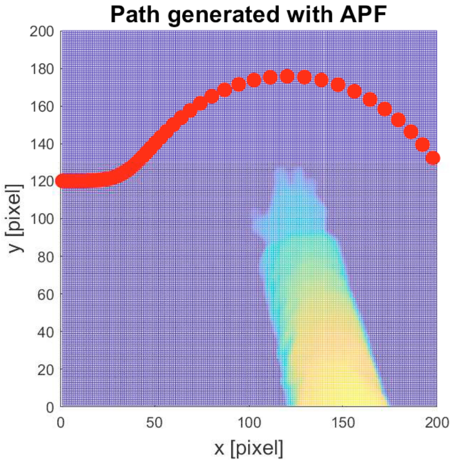


Fig. 7. Path generated with APF method using an augmented value for σ ($\sigma' = 1.5\sigma$)

4. CONCLUSIONS

In this paper, a novel approach to achieve human detection and avoidance in collaborative tasks was presented. The algorithm can detect the presence of the human operator in the workspace and modify the robot trajectory accordingly, avoiding potential collisions. This algorithm can be used online at a frequency dependent on the robot and the camera. The tests showed a computation time of 22 ms to calculate a new pose, whereas about 60 ms were used to take the photo and the data from the robot.

The algorithm mixes Radial Basis Functions to interpolate the scene, and Artificial Potential Fields to generate the path. In particular, RBFs create a continuous function on which the gradient is calculated; such a gradient, moreover, is used as a repulsive potential for APFs that automatically generate robot trajectory. This algorithm is innovative since it requires uniquely a single depth camera to work. Thus, it does not require either complex elaborations on multiple images, or complex setups. It requires the classical calibration of the camera, and a second simple calibration in which the robot moves over the space. The second calibration is necessary to remove the robot from future images. It is worth stating that using a single depth camera presents some limitations. For example, the algorithm is very conservative since it can not consider the third dimension. However, the only safe case not considered is when the operator executes actions above the robot. Indeed, possible worst-case scenarios can take place when the operator is working beneath the robot, such as the one in which the robot is handling a sharp object. Unfortunately, it is impossible to overcome this limitation using a single depth camera. The operator can move one arm over the robot and the other beneath it at the same time. Thus, the safety of the operator is not guaranteed in any case.

Future works will regard limitations and the elimination of the assumption of face-to-face task execution. Indeed, the robot overall encumbrance have been ignored.

REFERENCES

Amorim, A., Guimares, D., Mendona, T., Neto, P., Costa, P., and Moreira, A.P. (2021). Robust human position estimation in cooperative robotic cells. *Robotics and Computer-Integrated Manufacturing*, 67, 102035. doi: https://doi.org/10.1016/j.rcim.2020.102035.

Antão, L., Reis, J., and Gonçalves, G. (2019). Voxel-based space monitoring in human-robot collaboration environments. In *2019 24th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*, 552–559. doi:10.1109/ETFA.2019.8869240.

Boschetti, G., Faccio, M., and Minto, R. (2021). Control model for collaborative manufacturing: An integrated opened framework for human-robot collaboration. In V. Niola and A. Gasparetto (eds.), *Advances in Italian Mechanism Science*, 403–413. Springer International Publishing, Cham.

Bottin, M., Rosati, G., and Boschetti, G. (2021a). Working cycle sequence optimization for industrial robots. *Mechanisms and Machine Science*, 91, 228–236. doi: 10.1007/978-3-030-55807-9_26.

Bottin, M., Rosati, G., and Cipriani, G. (2021b). Iterative path planning of a serial manipulator in a cluttered known environment. *Mechanisms and Machine Science*, 91, 237–244. doi:10.1007/978-3-030-55807-9_27.

Broomhead, D.S. and Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Syst.*, 2.

Buhmann, M.D. (2000). Radial basis functions. *Acta Numerica*, 9, 1–38. doi:10.1017/S0962492900000015.

Carr, J.C., Beatson, R.K., Cherrie, J.B., Mitchell, T.J., Fright, W.R., McCallum, B.C., and Evans, T.R. (2001). Reconstruction and representation of 3d objects with radial basis functions. In *Computer Graphics (SIGGRAPH '01 Conf. Proc.), pages 67–76. ACM SIGGRAPH*, 67–76. Springer.

Cohen, Y., Shoval, S., Faccio, M., and Minto, R. (2021). Deploying cobots in collaborative systems: major considerations and productivity analysis. *Int. J. of Production Research*, 1–17.

Faccio, M., Bottin, M., and Rosati, G. (2019a). Collaborative and traditional robotic assembly: a comparison model. *The Int. J. of Advanced Manufacturing Technology*, 102, 1355–1372.

Faccio, M., Minto, R., Rosati, G., and Bottin, M. (2019b). The influence of the product characteristics on human-robot collaboration: a model for the performance of collaborative robotic assembly. *The Int. J. of Advanced Manufacturing Technology*, 106, 2317–2331.

Flacco, F., Kröger, T., De Luca, A., and Khatib, O. (2012). A depth space approach to human-robot collision avoidance. In *2012 IEEE Int. Conf. on Robotics and Automation*, 338–345. doi:10.1109/ICRA.2012.6225245.

IFR (2020). Top trends robotics 2020. `https://ifr.org/ifr-press-releases/news/top-trends-robotics-2020`.

Kim, S., Laschi, C., and Trimmer, B. (2013). Soft robotics: a bioinspired evolution in robotics. *Trends in Biotechnology*, 31(5), 287–294. doi:https://doi.org/10.1016/j.tibtech.2013.03.002.

Magrini, E., Ferraguti, F., Ronga, A.J., Pini, F., De Luca, A., and Leali, F. (2020). Human-robot coexistence and interaction in open industrial cells. *Robotics and Computer-Integrated Manufacturing*, 61, 101846. doi: https://doi.org/10.1016/j.rcim.2019.101846.

Mariotti, E., Magrini, E., and Luca, A.D. (2019). Admittance control for human-robot interaction using an industrial robot equipped with a f/t sensor. In *2019 Int. Conf. on Robotics and Automation (ICRA)*, 6130–6136. doi:10.1109/ICRA.2019.8793657.

Matheson, E., Minto, R., Zampieri, E.G.G., Faccio, M., and Rosati, G. (2019). Human–robot collaboration in manufacturing applications: A review. *Robotics*, 8(4). doi:10.3390/robotics8040100.

Mauro, S., Scimmi, L.S., and Pastorelli, S. (2018). Collision avoidance system for collaborative robotics. In C. Ferraresi and G. Quaglia (eds.), *Advances in Service and Industrial Robotics*, 344–352. Springer International Publishing, Cham.

Robla-Gómez, S., Becerra, V.M., Llata, J.R., González-Sarabia, E., Torre-Ferrero, C., and Pérez-Oria, J. (2017). Working together: A review on safe human-robot collaboration in industrial environments. *IEEE Access*, 5, 26754–26773. doi:10.1109/ACCESS.2017.2773127.

Scimmi, L.S., Melchiorre, M., Troise, M., Mauro, S., and Pastorelli, S. (2021). A practical and effective layout for a safe human-robot collaborative assembly task. *Applied Sciences*, 11(4). doi:10.3390/app11041763.

Secil, S. and Ozkan, M. (2022). Minimum distance calculation using skeletal tracking for safe human-robot interaction. *Robotics and Computer-Integrated Manufacturing*, 73, 102253.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2008). Robotics: Modelling, planning and control.

Strack, O. and Janković, I. (1999). A multi-quadric area-sink for analytic element modeling of groundwater flow. *J. of Hydrology*, 226(3-4), 188–196. doi:10.1016/S0022-1694(99)00147-X.

Tommasino, D., Bottin, M., Cipriani, G., Doria, A., and Rosati, G. (2021). Development and validation of an end-effector for mitigation of collisions. *J. of Mechanical Design*, 1–33. doi:https://doi.org/10.1115/1.4052443.

Villani, V., Pini, F., Leali, F., and Secchi, C. (2018). Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, 55, 248–266.

Warren, C. (1989). Global path planning using artificial potential fields. In *Proceedings, 1989 Int. Conf. on Robotics and Automation*, 316–321 vol.1.