IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 29, 2018, accepted December 15, 2018, date of publication December 24, 2018,
date of current version January 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2889311

Efficient Calculation of Minimum Distance Between Capsules and Its Use in Robotics

MOHAMMAD SAFEEA[1,2], PEDRO NETO[1], (Member, IEEE), AND RICHARD BEAREE[2], (Member, IEEE)

[1]Department of Mechanical Engineering, University of Coimbra, 3030-788 Coimbra, Portugal
[2]Arts et Métiers ParisTech, 59800 Lille, France

Corresponding author: Pedro Neto (pedro.neto@dem.uc.pt)

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme through the ColRobot project under Grant 688807, in part by the Portugal 2020 project DM4Manufacturing funded by UE/FEDER through the program COMPETE2020 under Grant POCI-01-0145-FEDER-016418, in part by the Portuguese Foundation for Science and Technology under Grant SFRH/BD/131091/2017, and in part by the Project COBOTIS under Grant PTDC/EME-EME/32595/2017.

ABSTRACT The problem of minimum distance calculation between line-segments/capsules, in 3D space, is an important subject in many engineering applications, spanning CAD design, computer graphics, simulation, and robotics. In the latter, the human–robot minimum distance is the main input for collision avoidance/detection algorithms to measure collision imminence. Capsules can be used to represent humans and objects, including robots, in a given dynamic environment. In this scenario, it is important to calculate the minimum distance between capsules efficiently, especially for scenes (situations) that include a high number of capsules. This paper investigates the utilization of QR factorization for performing efficient minimum distance calculation between capsules. The problem is reformulated as a bounded variable optimization in which an affine transformation, deduced from QR factorization, is applied on the region of feasible solutions. A geometrical approach is proposed to calculate the solution, which is achieved by computing the point closest to the origin from the transferred region of feasible solutions. This paper is concluded with numerical tests, showing that the proposed method compares favorably with the most efficient method reported in the literature.

INDEX TERMS Minimum distance, line-segments, capsules, robotics.

I. INTRODUCTION

The subject of minimum distance calculation between line-segments/capsules is important in many areas, for example in CAD design, computer graphics/games and in simulation. In such cases, minimum distance calculations are used to detect any overlap or collision between elements. This subject is also important in robotics for the problem of path planning and safety in human-robot interaction, where the minimum distance is used as a measure of collision imminence. In this scenario, calculating the minimum distance on-line is required for time critical applications such as human-robot collision avoidance and the path planning of robots navigating obstacles towards a goal. Most of collision avoidance methods require the calculation of the minimum distance between robot and surrounding environment (including humans), which are commonly represented by geometric primitives (capsules and/or spheres). By using higher number of geometric primitives the accuracy of representation increases. In such a case, the acquisition of more data from sensors is required, resulting in higher computational cost associated with sensor data processing and minimum distance calculation between the robot and the surrounding humans/objects. Outside computer graphics science the number of studies approaching human and object representation in real environment from real sensor data is very limited. In this context, quite a few methods had been proposed in literature. Ellipses have been utilized to represent the links of a robot while obstacles are represented by spheres [1]. A computationally efficient solution is based on the representation of the robot and obstacles by segments of lines with spheres swept onto them [2]. In [3] and [4] a humanoid robot is represented by capsules, while in [5] robot and human are represented by a collection of spheres. In [6], an industrial manipulator and a human are represented by capsules. A novel method for evaluating the distances between dynamic obstacles using multiple depth cameras is presented in [7].

A depth space oriented discretization of the Cartesian space is introduced, including occluded points. Robot and surrounding environment can be precisely described using mesh representation [8]. This method has the disadvantage of being costly for performing robot-obstacle minimum distance calculations. To speed up the computation, researchers have utilized the power of parallel processing, and GPU processors, to carry out the calculations [8]. A collision avoidance framework for mechanisms with complex geometries was demonstrated in simulation environment [9]. Diverse geometries/shapes have been reported in literature for accurate representation of humans and objects, namely in computer graphics science [10]. However, concerning the application in real environment using real sensors, capsules and spheres continue to be the most common geometry applied. Previous studies in human-robot collision avoidance showed that the error derived from the representation of humans using capsules is relatively small compared to the human-robot minimum distance dimension, so that it can be considered not problematic for the collision avoidance process [11].

In this study, the human and the robot are represented by capsules. Capsules are considered a good geometric primitive to represent a human, as in Figure 1 (A), in which L'Uomo Vitruviano from Leonardo da Vinci is represented by 10 capsules. The human body can be represented roughly by a single capsule, Figure 1 (B). In this scenario the arms can extend out of the capsule volume for some configurations. Figure 1 (C) shows a human represented by 3 capsules. In Figure 1 (D) the human is represented by 5 capsules, 2 capsules in each arm and 1 single capsule for the torso and head. A relatively precise representation of the human hand and forearm by 21 capsules is shown in Figure 1 (G).

A robot can roughly be represented by 2 capsules, Figure 1 (E), or by 3 capsules representing the main robot links (KUKA iiwa with 7 DOF), Figure 1 (F). The pose of the capsules covering the robot is obtained from forward kinematics calculation using the measured robot joint angles.

In [12], it is proposed an algebraic method for minimum distance computation between two capsules. Another method for computing the minimum distance between cylinders with flat ends was proposed in [13]. Nevertheless, the aforementioned methods are lengthy because they consider the different configurations in which two capsules might collide with each other. A method to determine the minimum distance between multiple known (geometry, position, orientation and configuration) and multiple unknown objects within a camera image is presented in [14]. The distance is estimated by searching for the largest expansion radius where the projected model does not intersect the object areas classified as unknown. A novel approach to approximate the minimum distance between robot links and obstacles is proposed in [15]. Obstacles are represented by a bounding box, modeled as cylinders and boxes. Each part of the robot arm is subdivided into an optimal number of spheres that encompass the initial volume. The minimum distance between the robot and the objects is approximated by the minimum distance between the obstacle bounding box and the spheres. An algorithm for computing the minimum translational distance based on the Gilbert-Johnson-Keerthi algorithm between two spherically extended polytopes is introduced in [16]. A well-known methodology for efficiently computing the segment to segment (capsules) distance which is considered the most efficient method in literature concerning computational efficiency is detailed in [17, pp. 417–418].

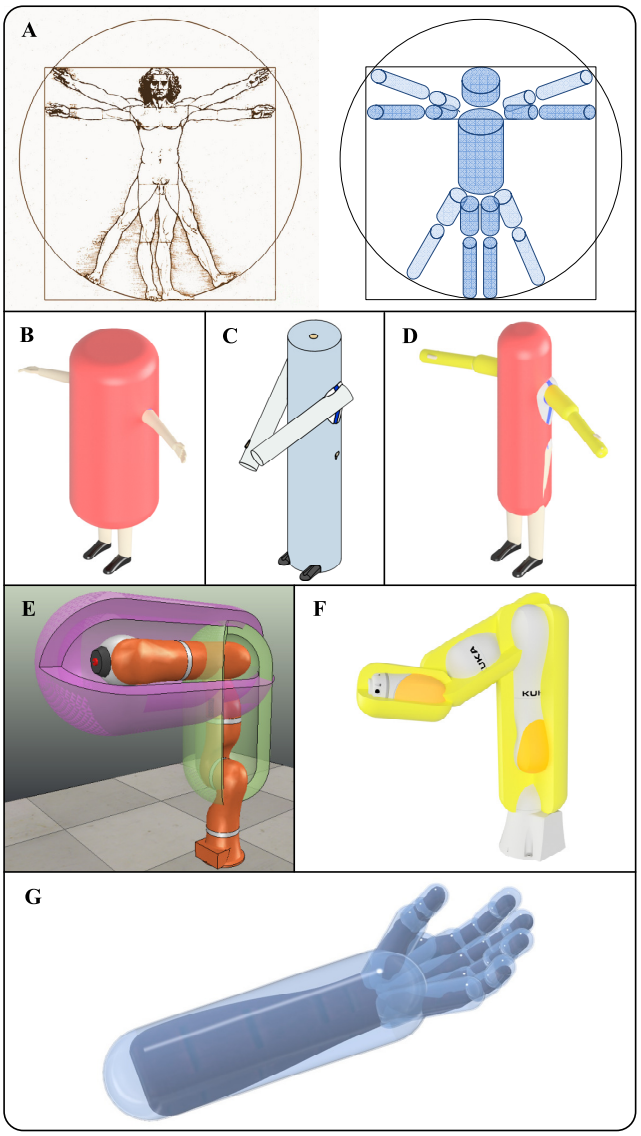

FIGURE 1. (A) L'Uomo Vitruviano from Leonardo da Vinci and its representation by 10 capsules, (B) human represented by 1 capsule, (C) human represented by 3 capsules, (D) human represented by 5 capsules, (E) robot represented by 2 capsules, (F) robot represented by 3 capsules and (G) human hand and forearm are represented by 21 capsules.

In this study, Section II presents the proposed QR-based capsule-capsule minimum distance method. Experiments and results are reported in Section III, both qualitative and quantitative. Finally, the conclusion is in Section IV. The Media materials contain the detailed deduction of the proposed QR method and running code.

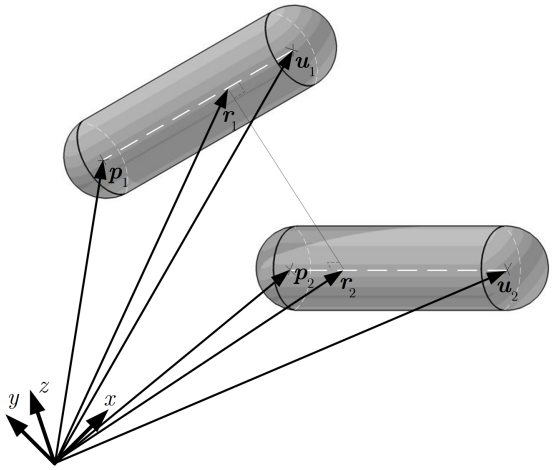

FIGURE 2. Minimum distance between two capsules.

II. MINIMUM DISTANCE: CAPSULE-CAPSULE

A. FORMULATION

Owing to their geometry, calculating the minimum distance between two capsules can be reduced to the calculation of the minimum distance between two line-segments at the capsules axes. In Figure 2 it is shown two line-segments representing the axes of two capsules and their associated common normal. Each capsule can be defined by two vectors (at the beginning and end of the capsule's axis-segment) and a radius ρ. Let's designate the position vectors defining the end points of the axis-segment of a capsule by $\boldsymbol{p}_1$ and $\boldsymbol{u}_1$ (capsule 1), and $\boldsymbol{p}_2$ and $\boldsymbol{u}_2$ (capsule 2). Then, we can define two vectors $\boldsymbol{s}_1 = \boldsymbol{u}_1 - \boldsymbol{p_1}$ and $\boldsymbol{s}_2 = \boldsymbol{u_2} - \boldsymbol{p_2}$.

Two points of interest, one in each axis segment of a capsule, are considered. Those points are represented relative to the base frame by two vectors, $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$:

$$\boldsymbol{r_1} = \boldsymbol{p}_1 + \boldsymbol{s_1}\lambda_1 \tag{1}$$

$$\boldsymbol{r_2} = \boldsymbol{p}_2 + \boldsymbol{s_2}\lambda_2 \tag{2}$$

where λ_1 and λ_2 are scalar parameters. Each parameter has a value in the range from zero to one when the point it represents is confined in between the two ends of the axis-segment of the capsule.

The problem of calculating the minimum distance between the two capsules renders to a minimization problem:

$$min(\Phi) = min(|\boldsymbol{p}_2 + \boldsymbol{s_2}\lambda_2 - (\boldsymbol{p}_1 + \boldsymbol{s_1}\lambda_1)| - \rho_1 - \rho_2) \tag{3}$$

where ρ_1 and ρ_2 are the radii of the capsules. Giving that ρ_1 and ρ_2 are constants, then the minimization problem can be reformulated:

$$min(|\triangle \boldsymbol{r}|) = min(|\boldsymbol{p}_2 + \boldsymbol{s_2}\lambda_2 - (\boldsymbol{p}_1 + \boldsymbol{s_1}\lambda_1)|) \tag{4}$$

where $\triangle \boldsymbol{r} = \boldsymbol{r_2} - \boldsymbol{r}_1$. We can rewrite the optimization function in the following equivalent quadratic form:

$$min(f) = min((\mathbf{A}\boldsymbol{x} + \boldsymbol{y})^{\mathrm{T}}(\mathbf{A}\boldsymbol{x} + \boldsymbol{y})) \tag{5}$$

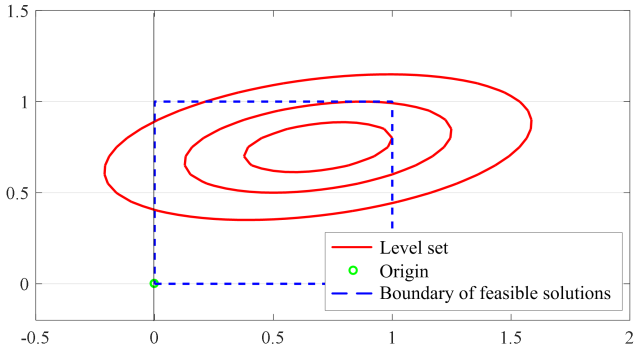

FIGURE 3. Region of feasible solutions of optimization problem (5).

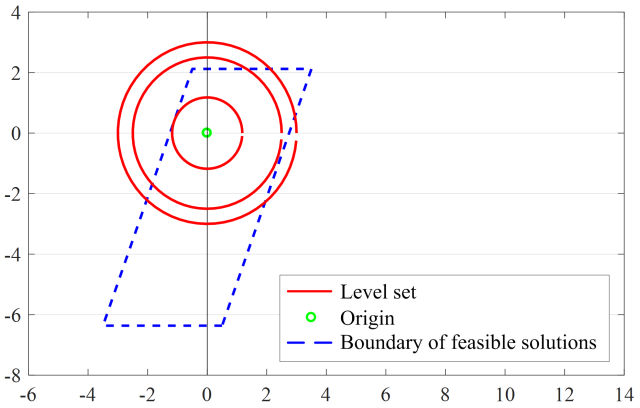

FIGURE 4. Modified optimization problem (7).

where matrix $\mathbf{A} = \begin{bmatrix}\boldsymbol{s_2} & -\boldsymbol{s_1}\end{bmatrix}$ and vector $\boldsymbol{y} = \boldsymbol{p_2} - \boldsymbol{p_1}$. The problem can be viewed as minimizing (5), subject to the constrains $0 < x_1 < 1$ and $0 < x_2 < 1$ (x_1 and x_2 are the components of the vector $\boldsymbol{x}$). Figure 3 shows the level sets and the region of feasible solutions of the optimization problem (5).

B. QR FACTORIZATION

The function f can be reformulated by performing **QR** factorization on matrix $\mathbf{A}$ and fixing. Then, the optimization problem in (5) is equivalent to:

$$min(f_1) = min((\mathbf{R}\boldsymbol{x} + \mathbf{Q}^{\mathbf{T}}\boldsymbol{y})^{\mathrm{T}}(\mathbf{R}\boldsymbol{x} + \mathbf{Q}^{\mathbf{T}}\boldsymbol{y})) \tag{6}$$

where $\mathbf{Q}$ is a 3×2 matrix whose column vectors are of unit length and mutually orthogonal, and matrix $\mathbf{R}$ is a 2×2 upper triangular. By performing a variable change the optimization problem becomes:

$$min(f_1) = min(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}) \tag{7}$$

where $\boldsymbol{u}$ is given by:

$$\boldsymbol{u} = (\mathbf{R}\boldsymbol{x} + \mathbf{Q}^{\mathbf{T}}\boldsymbol{y}) \tag{8}$$

The modified optimization problem (7) is shown in Figure 4. We notice that after performing the transformation described in (8) the elliptical level sets of the cost function are transformed into circles and the rectangular

Algorithm 1: Minimum distance calculation

Input : $\mathcal{R}$ region of feasible solutions
$(\mathbf{Q}, \mathbf{R})$ facorization matrices of $\mathbf{A}$
Output : $\boldsymbol{u_{min}}$ vector of coefficients
01 : Transform $\mathcal{R}$ using the function $f(x) = (\mathbf{R}x + \mathbf{Q}^{\mathrm{T}}y)$
02 : **If** Origin is inside $\mathcal{R}$ **then**
03 : $\boldsymbol{u_{min}} \leftarrow [0, 0]^{\mathrm{T}}$
04 : **else**
05 : **for** each boundary segment of $\mathcal{R}$ **do**
06 : $\boldsymbol{c} \leftarrow$ point of segment closest to origin
07 : **If** first iteration **then**
08 : $\boldsymbol{u_{min}} \leftarrow \boldsymbol{c}$
09 : **else**
10 : **If** norm($\boldsymbol{u_{min}}$) > norm($\boldsymbol{c}$) **then**
11 : $\boldsymbol{u_{min}} \leftarrow \boldsymbol{c}$
12 : **end if**
13 : **end if**
14 : **end for**
15 : **end if**

region of feasible solutions is transformed into a parallelogram. The solution to the modified optimization problem (7) is reduced to finding that point of the parallelogram region closest to the origin, $\boldsymbol{u_{min}}$, efficiently calculated in 2D space, Algorithm 1. Finally, the minimum distance between two capsules is calculated from:

$$d_{min} = \sqrt{\boldsymbol{u}_{min}^{\mathrm{T}}\boldsymbol{u}_{min} + \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{y}^{\mathrm{T}}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\boldsymbol{y}} - \rho_1 - \rho_2 \qquad (9)$$

For a set of n line-segments/capsules it can be noticed that:

1) Minimum distance calculations shall be performed mutually between any two capsules of the set, resulting in $O\left(n^2\right)$ complexity;
2) QR factorization of matrix $\mathbf{A}_i$ associated with a subset i of two capsules can be enhanced for efficiency since different $\mathbf{A}_i$ have shared columns in their structure;
3) The vector $\boldsymbol{u_{min}}$ is calculated in two dimensional space, while other algorithms calculate $\boldsymbol{x_{min}}$ in the three dimensional space;
4) Vector operations in $\Re^2$ are less costly than operations in $\Re^3$;
5) We propose $\boldsymbol{u_{min}}$ in $\Re^2$ and take advantage of the fact that the area of feasible solutions is a parallelogram.

A document with the full detail of the proposed QR method, including comparison with the method in [17] and C++ running code is in Media materials.

III. EXPERIMENTS AND RESULTS

Performance was evaluated by comparing the proposed QR method with the method in [17] to compute the segment to segment minimum distance. Three comparison criteria were considered: (1) computational complexity, (2) execution time using C++ and (3) numerical precision. The results for the computational complexity of the algorithms (number of floating point operations – addition, multiplication, division and square root for QR) are in Table 1. For the second comparison criteria, a set of line-segments was randomly generated and the minimum distance (squared) between each two line-segments of the set was calculated using the proposed QR method and the method in [17]. By implementing the algorithms in C++, results indicate that in terms of execution time the proposed QR method performed about 10% faster than the method in [17], Figure 5 and Figure 6.

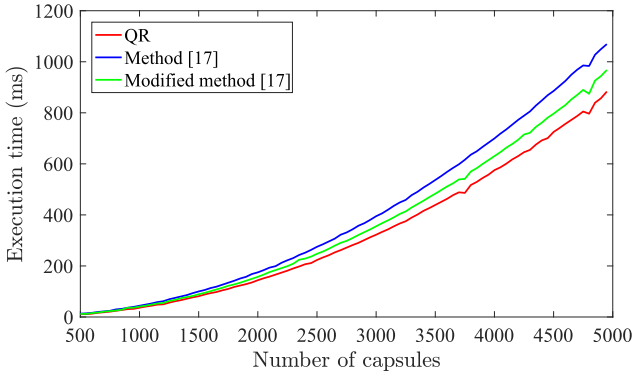

FIGURE 5. Execution time comparison for the proposed QR method, the method in [17] and the modified method in [17] as a function of number of line-segments/capsules of the set. Algorithms implemented in C++.

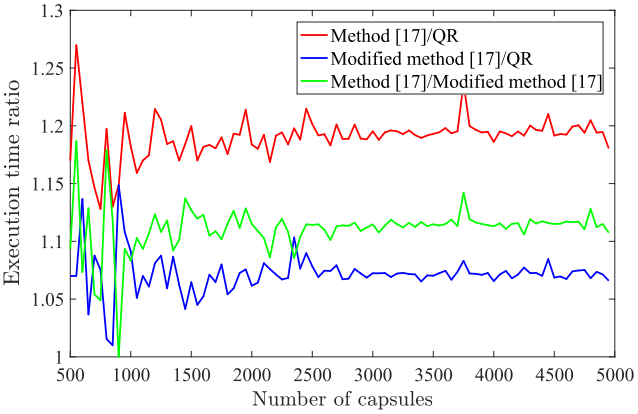

FIGURE 6. Execution time ratio (Method [17]/QR), (Modified method [17]/QR) and (Method [17]/Modified method [17]) as a function of number of line-segments/capsules of the set. Algorithms implemented in C++.

We noticed that method [17] can be modified by promoting some of the operations from $O(n^2)$ to $O(n)$. The results of those operations can be stored in memory and used later in the $O(n^2)$ part of the algorithm. In such a case, the $O(n^2)$ computational complexity of method [17] is reduced as shown in Table 1. Nevertheless, the proposed QR method is still more efficient in terms of execution time as shown in Figure 5 and Figure 6. The modified method in [17] is detailed in the Media materials.

Considering the third comparison criteria, numerical precision, a group of 5000 line-segments/capsules has been generated randomly in 3D space. The (x, y, z) coordinates of the start and end point of each segment are in the

TABLE 1. Computational complexity for the method in [17], the modified method in [17] and the proposed QR method.

Method	Route	Computational complexity
Method in [17]	Least expensive	$\frac{56}{2}n^2 - \frac{56}{2}n$
Method in [17]	Most expensive	$\frac{66}{2}n^2 - \frac{66}{2}n$
Modified method in [17]	Least expensive	$\frac{40}{2}n^2 - \frac{24}{2}n$
Modified method in [17]	Most expensive	$\frac{50}{2}n^2 - \frac{34}{2}n$
Proposed QR method	Least expensive	$\frac{29}{2}n^2 - \frac{5}{2}n$
Proposed QR method	Most expensive	$\frac{52}{2}n^2 - \frac{28}{2}n$

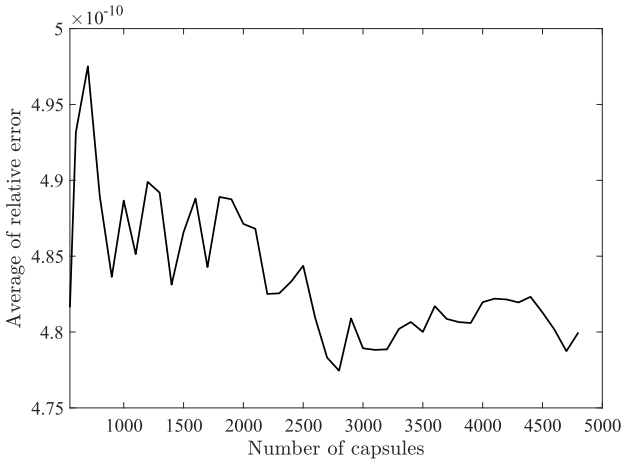

FIGURE 7. Average relative error between the method in [17] and the proposed QR method as a function of the number of line-segments/capsules of the set.

range $[-100, 100]$. The proposed QR method and the method in [17] are used to calculate the minimum distance between each couple of segments from the set. For each couple of segments, the relative error is defined as the ratio of the absolute value of difference between calculations using the two methods:

$$e = \frac{2|d_{min}^{qr} - d_{min}^{s}|}{d_{min}^{qr} + d_{min}^{s}} \tag{10}$$

where d_{min}^{qr} is the minimum distance calculated using the proposed QR method and d_{min}^{s} is the minimum distance calculated using the method in [17]. Experimental tests resulted in a maximum value of the relative error of $1.059e^{-8}$, a minimum value of the relative error of $1.11e^{-16}$ and an average error of $4.87e^{-10}$. These values demonstrate that the error is negligible. The same test has been repeated for different groups of line-segments/capsules with different number of elements, Figure 7.

IV. CONCLUSION

In this study we proposed a novel method based on QR factorization for performing minimum distance calculations for a set of line-segments/capsules. Capsules demonstrated to be good solution to represent humans and objects in real environment having data from real sensors as input. Experimental results indicate that the proposed solution is more efficient than the existing most efficient method in literature. Such efficiency was measured in computationally complexity (reduced number of floating point operations), execution time (about 10% better) and numerical precision (the error is negligible).

REFERENCES

[1] S. I. Choi and B. K. Kim, "Obstacle avoidance control for redundant manipulators using collidability measure," *Robotica*, vol. 18, no. 2, pp. 143–151, 2000.
[2] P. Bosscher and D. Hedman, "Real-time collision avoidance algorithm for robotic manipulators," *Ind. Robot, Int. J.*, vol. 38, no. 2, pp. 186–197, 2011.
[3] A. De Santis, A. Albu-Schaffer, C. Ott, B. Siciliano, and G. Hirzinger, "The skeleton algorithm for self-collision avoidance of a humanoid manipulator," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Sep. 2007, pp. 1–6.
[4] C. Fang, A. Rocchi, E. M. Hoffman, N. G. Tsagarakis, and D. G. Caldwell, "Efficient self-collision avoidance based on focus of interest for humanoid robots," in *Proc. IEEE-RAS 15th Int. Conf. Humanoid Robots (Humanoids)*, Nov. 2015, pp. 1060–1066.
[5] L. Balan and G. M. Bone, "Real-time 3D collision avoidance method for safe human and robot coexistence," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 276–282.
[6] C. Liu and M. Tomizuka, "Algorithmic safety measures for intelligent industrial co-robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 3095–3102.
[7] F. Fabrizio and A. De Luca, "Real-time computation of distance to dynamic obstacles with multiple depth sensors," *IEEE Robot. Autom. Lett.*, vol. 2, no. 1, pp. 56–63, Jan. 2017.
[8] K. B. Kaldestad, S. Haddadin, R. Belder, G. Hovland, and D. A. Anisi, "Collision avoidance with potential fields based on parallel processing of 3D-point cloud data on the GPU," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 3250–3257.
[9] M. Sagardia, A. M. Turrillas, and T. Hulin, "Realtime collision avoidance for mechanisms with complex geometries," in *Proc. IEEE Conf. Virtual Reality 3D User Inter. (VR)*, Mar. 2018, p. 1.
[10] R. Hu, M. Savva, and O. van Kaick, "Functionality representations and applications for shape analysis," *Comput. Graph. Forum*, vol. 37, no. 2, pp. 603–624, 2018.
[11] M. Safeea and P. Neto, "Human-robot collision avoidance for industrial robots: A V-REP based solution," in *Proc. 25th ISPE Int. Conf. Transdisciplinary Eng. Transdisciplinary Eng. Methods Social Innov. Ind.* Amsterdam, Netherlands: IOS Press, 2018.
[12] P. Ennen, D. Ewert, D. Schilberg, and S. Jeschke, "Efficient collision avoidance for industrial manipulators with overlapping workspaces," *Procedia CIRP*, vol. 20, pp. 62–66, 2014.
[13] D. Biermann, R. Joliet, and T. Michelitsch, "Fast distance computation between cylinders for the design of mold temperature control systems," TU Dortmund Univ., Dortmund, Germany, Tech. Rep., 2008, doi: 10.17877/DE290R-8711.
[14] S. Kuhn and D. Henrich, "Fast vision-based minimum distance determination between known and unkown objects," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2007, pp. 2186–2191.
[15] S. Tarbouriech and W. Suleiman, "On bisection continuous collision checking method: Spherical joints and minimum distance to obstacles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 7613–7619.
[16] E. J. Bernabeu and J. Tornero, "Hough transform for distance computation and collision avoidance," *IEEE Trans. Robot. Autom.*, vol. 18, no. 3, pp. 393–398, Jun. 2002.
[17] P. J. Schneider and D. H. Eberly, *Geometric Tools for Computer Graphics*. San Francisco, CA, USA: Morgan Kaufmann, 2003, pp. 417–418.

MOHAMMAD SAFEEA received the M.S. degree in mechanical engineering from the University of Coimbra, Portugal, in 2016. He is currently pursuing the Ph.D. degree in robotics with the University of Coimbra and ENSAM ParisTech, Lille, France.

During his career, he has been involved in several projects, where he developed several software solutions for robotics, control, and interfacing with hardware and sensors. His main research topics are robot dynamics, robot control, and the issues of safety for human–robot collaboration.

PEDRO NETO received the Ph.D. degree in mechanical engineering (robotics) from the University of Coimbra, in 2012. He is currently a Professor Auxiliar with the Department of Mechanical Engineering, University of Coimbra. He is a supervisor for several M.Sc. and Ph.D. students and Postdoctoral researchers. He created the Collaborative Robotics Laboratory (CoRLuc), in 2016, with an investment of more than 1 million Euros. CoRLuc research activities focus in collaborative robotics, human–robot interaction, and the application of robots in advanced manufacturing activities in the context of industry 4.0. He has authored more than 100 publications, including books, book chapters, international journal papers, conference proceedings, reports, and demonstration videos. He coordinates several research projects at the University of Coimbra, including flagship projects supported by the European Commission under Horizon 2020 Framework. He served on the scientific committees of several conferences and has been a member of the IEEE Factory Automation technical committee, since 2016.

RICHARD BEAREE received the M.Eng. degree in mechanical engineering from Lille University, in 2001, the M.S. degree in automatic control, in 2002, and the Ph.D. degree in automatic control from the French engineering school ENSAM ParisTech, in 2005. He is currently a Professor in robotics with the LISPEN Laboratory, ENSAM ParisTech. He is also the Head of the Specialized Master Program "Expert in collaborative robotics for Industry of the Future." His current research interests include robot localization, computer vision, trajectory planning, trajectory generation, and vibration control with applications to industrial robotics systems.