

Safe Human-Robot Interaction Based on Dynamic Sphere-Swept Line Bounding Volumes

J. A. Corrales*, F. A. Candelas and F. Torres

*Physics, Systems Engineering and Signal Theory Department
University of Alicante
Carretera San Vicente del Raspeig s/n, 03690 San Vicente del Raspeig, Spain*

Emails: jcorrales@ua.es; Francisco.Candelas@ua.es; Fernando.Torres@ua.es

* Corresponding Author: Phone: +34 96 590 34 00 - Ext: 2440; Fax: +34 96 590 97 50

Abstract: This paper presents a geometric representation for human operators and robotic manipulators which cooperate in the development of flexible tasks. The main goal of this representation is the implementation of real-time proximity queries which are used by safety strategies for avoiding dangerous collisions between humans and robotic manipulators. This representation is composed of a set of bounding volumes based on swept-sphere line primitives which encapsulate their links more precisely than previous sphere-based models. The radius of each bounding volume does not only represent the size of the encapsulated link, but it also includes an estimation of its motion. The radii of these dynamic bounding volumes are obtained from an algorithm which computes the linear velocity of each link. This algorithm has been implemented for the development of a safety strategy in a real human-robot interaction task.

Keywords: Human-robot interaction; Bounding volumes; Distance computation; Safety; Collision avoidance

1 Introduction

The cooperation between human and robots in the execution of a common task takes advantage of their complementary features. On one hand, the robot completes those subtasks which are exhausting or dangerous for the human. On the other hand, the human performs the subtasks which cannot be executed by the robot because of their complexity. This synergy enables the development of more flexible and complex tasks which cannot be performed individually by a human or a robot [1]. The current paper focuses on the application of human-robot interaction in the development of industrial tasks. The application of human-robot interaction in industrial environments does not only enable the execution of more complex tasks but it also improves their productivity [2] because the robot and the human operator can work inside the same work cell. Furthermore, safety fences become unnecessary, the robot's work cell can be reduced and this involves a decrease in the workspace costs.

Nevertheless, the safety of human operators has to be always guaranteed when humans and robots share the same workspace. Therefore, a precise localization of the robots and the humans is needed in order to verify that there are no collisions among them. This localization cannot only be reduced to a global positioning in the workspace because close collaborative tasks require a more detailed tracking of their body structure. Thus, a tracking of all their links is needed. In the case of the robot, this tracking is easily performed by reading the joint angles from its controller and applying forward kinematics. In the case of the human operator, a motion capture system should be used. In this paper, the human wears a motion capture suit based on inertial sensors (gyroscopes and accelerometers) whose measurements are combined with those of a UWB localization system in order to obtain precise estimates of the movements of the limbs of the human [3]. An inertial motion capture system has been chosen owing to its

advantages over other motion capture technologies [4]: self-containment, easy installation, no line-of-sight restrictions, no occlusions, precise information for joint rotations and high sampling rates.

The tracking systems described above register the global position of the human and the robot in the environment and the rotation angles of their joints. This localization information is applied to linear skeletons which represent their bodies as kinematic chains composed by links and joints. Although these skeleton representations model correctly their movements; they do not take into consideration the shape and size of the surface of their bodies, which is indispensable for collision avoidance. Therefore, it is necessary to create a model of the surfaces of the bodies of the human operator and the robot. This model should be precise by having a similar shape to the links but it should also be simple enough to compute collision detection on real-time. This paper develops a bounding volume representation for the surface of the human and the robot which fulfills these two requirements: precision and efficiency. In particular, these bounding volumes are based on swept-sphere line primitives, which provide a tighter fit than previous spherical representations [5-8] but with a very similar computational cost.

In addition, this paper presents a novel dynamic bounding volume algorithm which determines the sizes of the bounding volumes depending of the movements of the links which they cover. In particular, the radius of each swept-sphere line is proportional to the linear-velocity of its corresponding link. Thus, the size of the bounding volume does not only models the shape of the link but it also represents the distance that the link will cover between each pair of sampling steps. Therefore, these dynamic bounding volumes are estimations of the motions that the links may have performed in the next sampling step. This fact supposes an improvement with regard to previous collision detection techniques based on static bounding volumes. Static bounding volume algorithms only compute the collision detection at each sampling step without considering what may happen between the current sampling step and the next one.

This paper is organized as follows. Section 2 presents an in-depth study of the main advantages and disadvantages of the previous approaches developed for computing proximity queries in robotic systems. Section 3 explains the bounding volume representation developed and lists its main virtues over the previous approaches. Section 4 describes the algorithm which establishes the radii of the bounding volume depending on the linear velocity which is computed for each link of the body of the robot and the human. Section 5 applies the theoretical concepts of the previous two sections in a real human-robot interaction task where a metallic structure is assembled. Finally, the last section presents the conclusions of this paper and future lines of work which could be considered.

2 Previous Work

Collision detection and minimum distance computation between 3D objects are two important techniques which are used in different applications [9, 10] such as robotic motion planning, computer graphics, computational geometry and simulation. Both techniques can be considered as two different solutions for a common problem, the geometric proximity estimation [5]. Proximity estimation techniques can be classified [11] in discrete or continuous techniques depending on how the trajectories of the objects are represented. Discrete techniques sample the objects' trajectories at fixed time instants while continuous techniques take into account all the configurations of the objects along their complete trajectories. The most commonly used discrete techniques (feature tracking and bounding volume) and continuous techniques (swept-volume intersection and trajectory intersection) are described in the following paragraphs.

Feature tracking techniques apply the proximity queries over geometric features (such as vertices, edges or faces) that form the boundaries of the objects and require that the objects are

modeled as convex polyhedrons. Several algorithms have been developed in the literature in order to track the closest features of the polyhedrons which represent the objects: Lin-Canny [12], Voronoi-Clip (V-Clip) [13], Gilbert-Johnson-Keerthi (GJK) [14] and Hierarchical-Walk (H-Walk) [15]. These algorithms are suitable for objects which have regular geometric shapes because they can be modeled with a few polyhedral components. However, non-convex and irregular objects require polyhedrons composed by a large number of faces in order to approximate their surfaces. In these cases, feature tracking techniques become computationally unaffordable.

Bounding volume techniques define a group of bounding volumes that encapsulate the surface of the objects. Bounding volumes are geometric shapes which have simpler proximity queries than the complex objects they bound. Therefore, bounding volume techniques are more suitable for real-time applications than feature tracking techniques. The more commonly used bounding volumes are: axis-aligned bounding boxes (AABBs) [16], object-oriented bounding boxes (OOBBs) [17], swept-sphere volumes [18], discrete-orientation polytopes (k-DOPs) [19] and spheres [5, 6].

Swept volume intersection techniques [20-22] compute the volumes swept by the movements of the objects during the whole time span and test them for overlap. These techniques are continuous because they consider all the possible configurations of the objects and no collision is missed out. However, the computation of these swept volumes is very complex and is computationally expensive.

Trajectory intersection techniques are based on the estimation of the trajectories of the objects and the detection of overlaps among them. The objects' trajectories are estimated by evaluating algebraic polynomials depending on time [23, 24] or by bisecting the objects' paths into segments [25]. The computational cost of these techniques becomes high when the objects' trajectories are too complex because the number of degrees of the polynomials and the number of path bisections have to be increased.

In robotics applications, proximity estimation techniques based on bounding volumes or a combination of another technique with a bounding volume technique are generally applied. For instance, in [7] a set of spheres is used to create a safety strip between the robot and the environment and the distances between these spheres are applied in an optimization method to determine the best configuration of the robot. In [6] a set of dynamic spheres is also applied for collision detection between robot arms and human operators. These spheres are obtained from an approximation of the surface of the object by generalized cylinders and the number of spheres is dynamically changed depending on the possibility of a collision. In [5] a sphere-tree hierarchy is defined in order to approximate the current configuration of the links of a humanoid robot and their range of motion. In [8] an optimization method is implemented to select the direction of the robot's trajectory which minimizes the distance between the robot and its goal and which maximizes the distances between the sphere-based geometric models of the human and the robot.

As shown above, bounding volume techniques are preferred in robotic applications because they present a good relation between their computational complexity and the precision of the provided proximity data for real-time tasks. Feature tracking techniques are not convenient for human-robot interaction tasks because the human body requires complex polyhedrons with many faces to be modeled. Swept-volume intersection and trajectory intersection techniques are also not used because the human body movements involve too complex trajectory representations. Thereby, the inclusion of the human operator in the robotic workspace increases the computational costs of these algorithms and makes them unfeasible for real-time proximity queries in human-robot interaction tasks.

3 Geometric Representation

The structure of the body of the human operator and the robotic manipulator are represented with the same model: a linear skeleton. This representation has been chosen because it is the simplest way to represent the kinematic chain which produces the movements of the human and the robot. The skeleton is composed of a group of bones which identify the links of the robot and the limbs of the human. The bones of the skeleton are connected by nodes which identify the joints of the robot and the human and allow relative motions of neighboring bones. In addition, these nodes are organized in a tree hierarchy which represents the dependency of their motion. Thereby, the motion of a parent node will involve a relative motion of its children nodes because they are physically connected.

Fig. 1 depicts the skeleton of the human operator and the skeleton of the robotic manipulator which has been used in the presented research (a 7 d.o.f Mitsubishi PA-10). The skeleton of the human operator is composed by 18 nodes while the skeleton of the robot is composed by 8 nodes. The positions of the root nodes of these skeletons (hips and base, respectively) represent their global position in the environment and the movements of the rest of the nodes are relative to them.

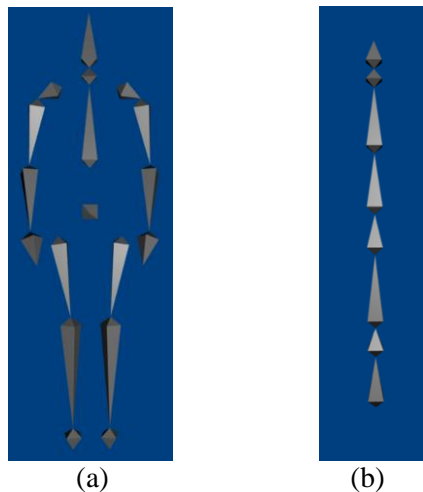


Fig. 1 (a) Skeleton of the human operator and (b) skeleton of the robotic manipulator.

A coordinate system is assigned to each node in order to represent the motion of the corresponding joint. The Y axis of each coordinate system matches the main direction of the bone which is directly connected to the node. As shown above, the pose of each node's coordinate system is relative to the parent's coordinate system. In order to compute the relative transformation matrix between each node and its parent, the axis-position joint representation [26] is used instead of the typical Denavit-Hartenberg notation [27] because the human skeleton includes branching nodes which cannot be modeled with the Denavit-Hartenberg notation. Therefore, each coordinate system i can be expressed relatively to its parent's coordinate system $i-1$ by a transformation matrix ${}^{i-1}T_i$ composed by the rotation of the joint i and the offset between both nodes. While the rotation of the joint is dynamically obtained from the measurements of the inertial motion capture system, the offset between the nodes is a fixed value predefined for each skeleton. By recursively pre-multiplying each transformation matrix by the transformation matrices of its predecessors, the transformation matrix (forward kinematics) between each node and the world coordinate system is obtained.

Each bone of the skeletons of the human operator and the robot manipulator is covered with a bounding volume which models the geometry of the corresponding link. As shown in section 2, previous collision detection systems for robotic applications [5-8] commonly develop

bounding volume strategies based on spheres. This is due to the fact that spheres have a quick overlap test and only their centers need to be transformed during the robot tracking. Nevertheless, spheres do not provide a very tight fit for cylindrical links [28]. Therefore, much more spheres are needed for approximating the surface of these links and thus the computational cost of the sphere-based proximity techniques increases. This paper uses sphere-swept lines (SSLs) as bounding volumes in order to overcome these limitations of the sphere approximations. The memory cost for representing SSL and spheres is very similar because SSLs need three parameters (the ends of a segment and a radius) while spheres need two parameters (the center and a radius). However, the SSL bounding volumes fit better the cylindrical links and thus only one SSL is needed for covering each skeleton's bone. In fact, the ends of the segment of each SSL coincide with the ends of the corresponding bone. In Fig. 2, the SSL bounding volumes which have been used to cover the skeleton of the human and the robot are shown.

The computation of the distance between two SSLs has a constant cost as the computation of the distance between two spheres. Firstly, the distance between the closest points of the two segments of both SSLs is calculated (see [29] for a detailed formulation of the problem). Then, the real distance between the SSLs is obtained by subtracting the radii of the SSLs to this distance.

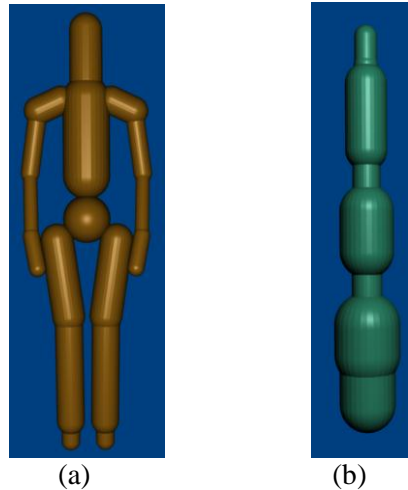


Fig. 2 (a) SSL bounding volumes for the human's skeleton and (b) SSL bounding volumes for the robot's skeleton.

4 Dynamic Bounding Volume Algorithm

4.1 General Description of the Algorithm

Static bounding volumes have a constant size which is pre-established beforehand and it does not change during the execution of the proximity queries. However, in the case of human-robot interaction, these volumes should change depending on the movements of the human operator and the robot in order to develop a safer interaction. One of the main contributions of this paper is the development of dynamic bounding volumes whose sizes change depending on the linear velocity of the associated link. In particular, the radius $radius_i(t)$ of the SSL which encapsulate the link i of the skeleton has two components: a constant component $radius_surface_i$ which includes the maximum radius of the surface of the link and a variable component which is proportional to the linear velocity of the link at each time instant t :

$$radius_i(t) = radius_surface_i + |{}^i v(t)| \cdot \Delta t \quad (1)$$

The second term of Eq. 1 predicts the distance that the link will cover during the sampling period Δt . Therefore, the radius of the SSL does not only represent the geometry of the corresponding link but also an approximation of its movement between each pair of samples.

The velocities of all the links of the skeletons of the human operator and the robotic manipulator must be calculated in order to compute the radius of each SSL at each sampling instant. In order to compute these radii and then calculate the minimum distance between the human and the robot, the algorithm summarized in Fig. 3 is implemented and executed at a constant time interval Δt . Firstly, the measurements of the joint angles are obtained. The robot's joint angles are read from the robot's controller while the human's joint angles are obtained from an inertial motion capture suit which is worn by the human operator. Next, the angular rate of each joint is estimated from the variation of the joint angles. The linear velocity of each link is also calculated from the joint angular rates in order to update the radius of each bounding volume according to Eq. 1. The position of the ends of each SSL is updated by applying forward-kinematics. Finally, the minimum distance between the bounding volumes of the human and the bounding volumes of the robot is computed. This value is obtained by calculating the distance between each pair of SSLs and selecting the minimum one. In the following sections, the joint angular rate estimation step and the link linear velocity computation step of this algorithm are described in detail.

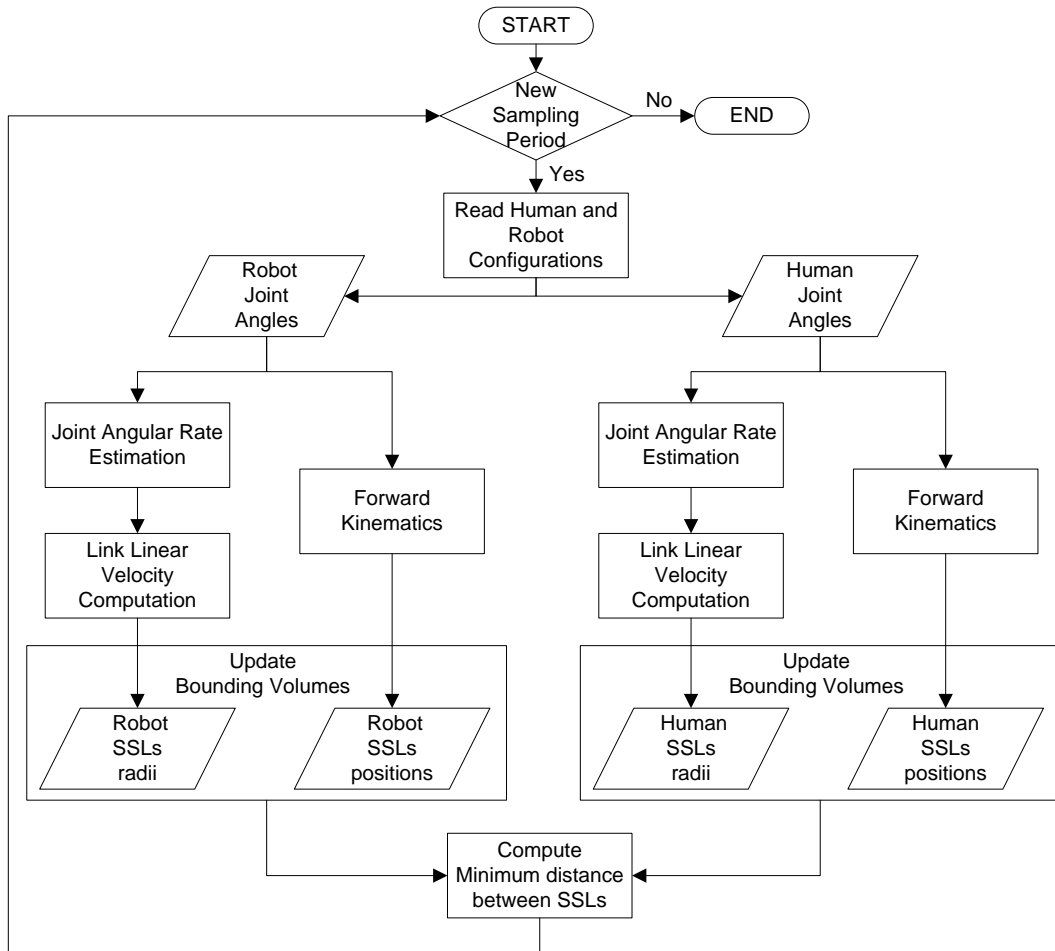


Fig. 3 Diagram of the dynamic bounding volume algorithm based on the linear velocities of the links.

4.2 Joint Angular Rate Estimation

After reading the angle values of each joint of the human operator and the robot, the current angular rate of each joint should be computed. This value identifies the instantaneous velocity with which the joint angles change through time. A first approach for obtaining the angular rate of each joint i is the finite difference method (FDM) which uses the difference between the current joint angle $\alpha_i(t)$ and the previous one $\alpha_i(t-1)$:

$$\dot{\alpha}_i(t) = \frac{\alpha_i(t) - \alpha_i(t-1)}{\Delta t} \quad (2)$$

However, this method is not convenient when the sampling period Δt is small because the errors in the angle measurements are amplified and become significant [30].

Therefore, an adaptive windowing technique [30] has been implemented to overcome these limitations in the computation of the joint angular rates. This technique involves the calculation of the angular range in a window of previous angle measurements. The size of this window is selected adaptively depending on the magnitude of the velocity. On one hand, the window size should be small when the velocity is high in order to maintain reliable estimates. On the other hand, when the velocity is low, the window should be large in order to obtain more precise estimates. In particular, the adaptive windowing technique uses a constant value ε_i which identifies the maximum allowable error between the current angle measurements α_i in the window and the estimates of these angles $\hat{\alpha}_i$ calculated from the angular rate in the window. In the developed system, this parameter ε_i is chosen according to the magnitude of the maximum acceleration of each joint. The size n of the window is the maximum value $n = \max\{1, 2, 3, \dots\}$ that verifies the following equation:

$$|\alpha_i(t-i) - \hat{\alpha}_i(t-i)| \leq \varepsilon_i \quad \forall i \in \{1, 2, \dots, n\} \quad (3)$$

The previous angle estimate $\hat{\alpha}_i(t-i)$ is computed with the following equation:

$$\hat{\alpha}_i(t-i) = \alpha_i(t) - \frac{i(\alpha_i(t) - \alpha_i(t-n))}{n} \quad (4)$$

Once the size n of the window is established, the angular rate in this window can be obtained with the following equation:

$$\dot{\alpha}_i(t) = \frac{\alpha_i(t) - \alpha_i(t-n)}{n\Delta t} \quad (5)$$

A comparison between the angular rates obtained with the finite difference method (Eq. 2) and with the adaptive windowing method (Eq. 5) is shown in Fig. 4. While the finite difference method presents small peaks due to the measurement noise, the adaptive windowing is able to smooth these peaks by estimating the angular rate in a dynamic window.

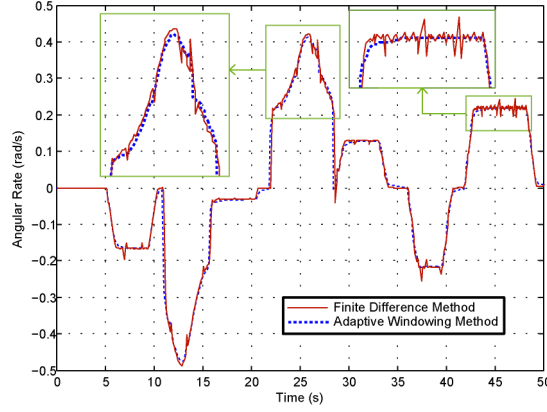


Fig. 4 Comparison of the angular rate of the first joint of the robotic manipulator in a human-robot interaction task computed by the finite difference method and the adaptive windowing method (with $\varepsilon = 0.0052$ rad).

4.3 Link Linear Velocity Computation

The next step of the dynamic bounding volume algorithm after the estimation of the angular rates of the joints is the calculation of the linear velocity of each link. First of all, the angular rate of each joint i (node of the skeleton) should be represented in the coordinate system attached to this joint. However, the angular rates $\dot{\phi}_i(t) = \begin{bmatrix} \dot{\alpha}_i(t) \\ \dot{\beta}_i(t) \\ \dot{\gamma}_i(t) \end{bmatrix}$ estimated in section 4.2 are not represented in this coordinate system because they are obtained from Euler angles in ZXY order. Therefore, a transformation matrix $E_{ZXY}(\phi_i(t))$ between both coordinate systems is needed. The columns of this matrix are obtained by representing the relative rotation axes of the Euler angles $\begin{bmatrix} \hat{Z}_{Euler} \\ \hat{X}_{Euler} \\ \hat{Y}_{Euler} \end{bmatrix}$ with regard to the absolute axes of the coordinate system of joint i :

$$E_{ZXY}(\phi_i(t)) = \begin{bmatrix} \hat{Z}_{Euler} \\ Rot_Z(\alpha_i(t)) \cdot \hat{X}_{Euler} \\ Rot_Z(\alpha_i(t)) \cdot Rot_X(\beta_i(t)) \cdot \hat{Y}_{Euler} \end{bmatrix} \quad (6)$$

The final form of the transformation matrix $E_{ZXY}(\phi_i(t))$ would be the following one:

$$E_{ZXY}(\phi_i(t)) = \begin{bmatrix} 0 & \cos(\alpha_i(t)) & -\sin(\alpha_i(t))\cos(\beta_i(t)) \\ 0 & \sin(\alpha_i(t)) & \cos(\alpha_i(t))\cos(\beta_i(t)) \\ 1 & 0 & \sin(\beta_i(t)) \end{bmatrix} \quad (7)$$

The angular rate of the joint i in its coordinate system is obtained by applying the transformation matrix $E_{ZXY}(\phi_i(t))$ to the Euler angular rate $\dot{\phi}_i(t)$:

$$\dot{\theta}_i(t) = E_{ZXY}(\phi_i(t)) \cdot \dot{\phi}_i(t) \quad (8)$$

As the joints of the skeleton are part of a kinematic chain, the angular velocity of the coordinate system of a joint i does not only depend on its angular rate but also it depends on the angular velocity of the previous joints (i.e. the predecessor nodes in the skeleton's hierarchy). Therefore, for calculating recursively the angular velocity of each coordinate system i , the

angular velocity ${}^{i-1}\omega_{i-1}(t)$ of its parent $i-1$ in addition to its own angular rate $\dot{\theta}_i(t)$ are considered:

$${}^i\omega_i(t) = {}^iRot_{i-1}(t) \cdot {}^{i-1}\omega_{i-1}(t) + \dot{\theta}_i(t) \quad (9)$$

The matrix ${}^iRot_{i-1}(t)$ represents the rotation transformation between the coordinate system $i-1$ and the coordinate system i . It is obtained by multiplying the rotation matrices corresponding to the rotation angles of joint i :

$${}^iRot_{i-1}(t) = Rot_z(\alpha_i(t)) \cdot Rot_x(\beta_i(t)) \cdot Rot_y(\gamma_i(t)) \quad (10)$$

The linear velocity of each node of the skeleton is computed similarly. In particular, the linear velocity ${}^i v_i(t)$ of the coordinate system i is composed by the linear velocity ${}^{i-1}v_{i-1}(t)$ of its parent $i-1$ plus the linear velocity caused by the angular velocity of its parent $i-1$:

$${}^i v_i(t) = {}^iRot_{i-1}(t) \cdot ({}^{i-1}v_{i-1}(t) + {}^{i-1}\omega_{i-1}(t) \times {}^{i-1}P_i) \quad (11)$$

Where ${}^{i-1}P_i$ is the offset vector between the coordinate systems $i-1$ and i .

This linear velocity corresponds to the coordinate system i which is situated at the beginning of the link i . The linear velocity ${}^i v_{i+1}(t)$ at the end of the link (where the child coordinate system $i+1$ is situated) has also to be calculated. This velocity not only contains the linear velocity of the beginning of the link (caused by the predecessors' movements) but also the linear velocity generated at the end of the link by the angular velocity of the joint:

$${}^i v_{i+1}(t) = {}^i v_i(t) + {}^i\omega_i(t) \times {}^i P_{i+1} \quad (12)$$

The linear velocity with maximum norm between the two ends of each link i (${}^i v_i(t)$ and ${}^i v_{i+1}(t)$) is used as the linear velocity ${}^i v(t)$ of that link and is applied in Eq. 1 in order to determine the radius of the corresponding SSL at each time step.

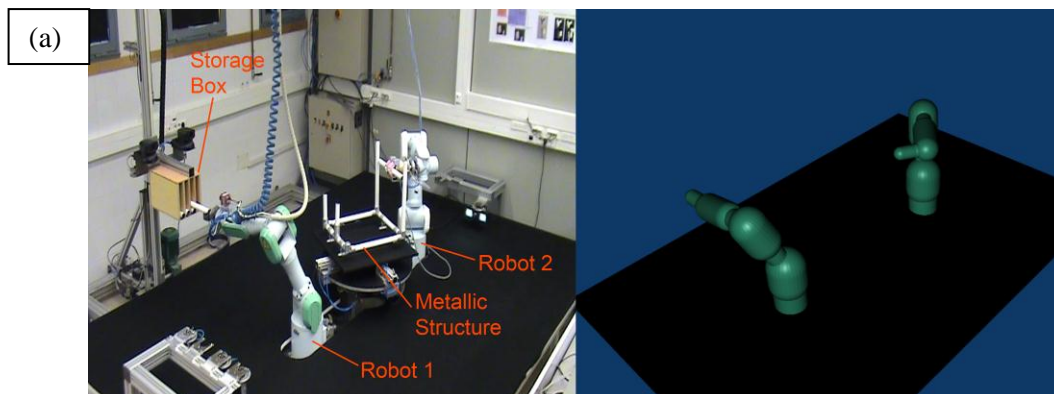
5 Experimental Results

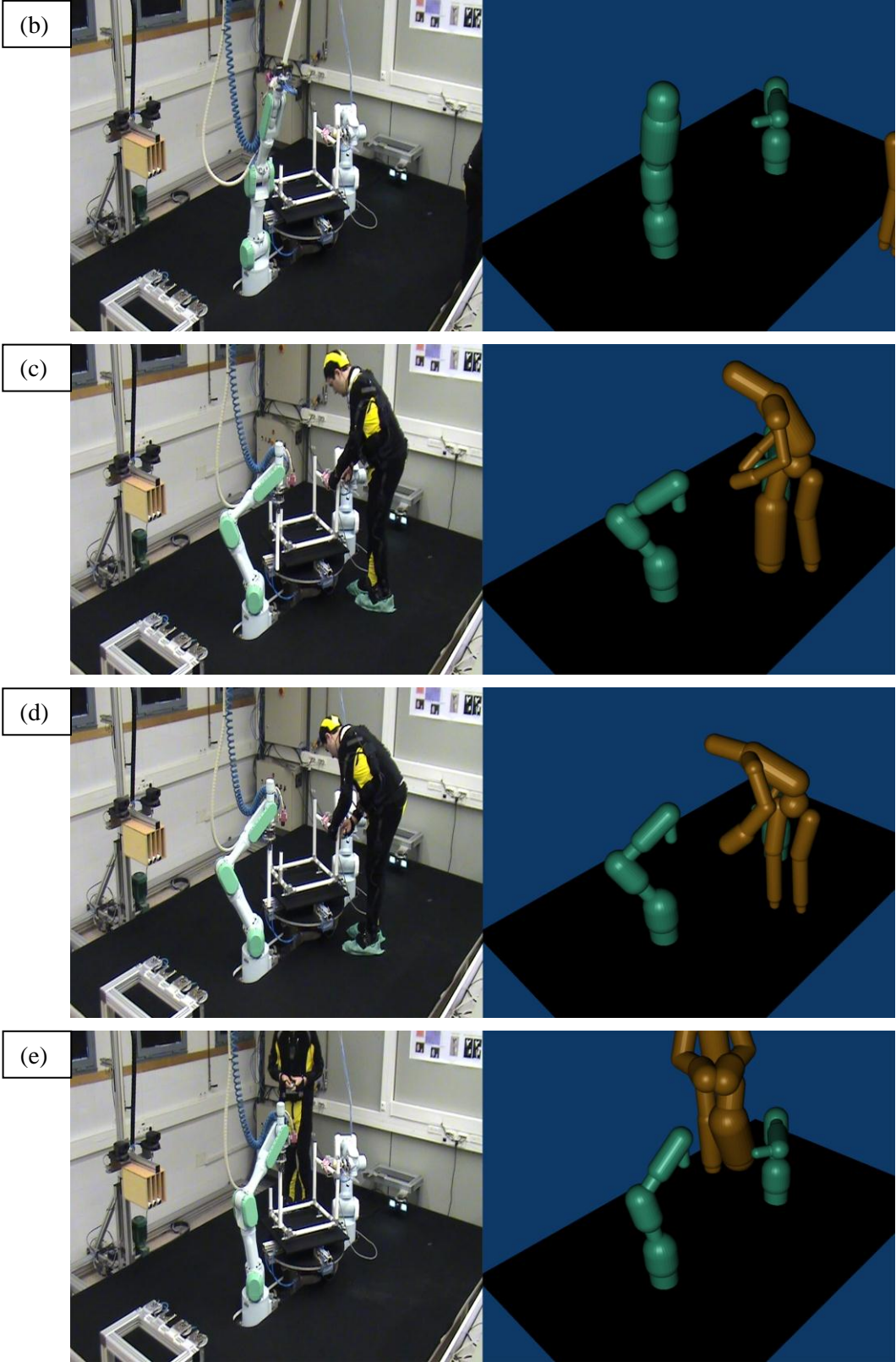
The dynamic bounding volume algorithm described in section 4 has been applied in the development of a safe human-robot interaction task. In particular, this task consists in the assembly of a metallic structure where two 7 d.o.f robotic manipulators (Mitsubishi PA-10) and a human operator cooperate. As the human operator enters the manipulators' workspaces, his/her safety has to be guaranteed by monitoring the movements of the human and the manipulators. On one hand, the human operator wears an inertial motion capture suit (Animazoo GypsyGyro-18) and a UWB localization tag (Ubisense) whose localization measurements are combined with a Kalman filter [3] in order to track all the movements of her/his limbs. On the other hand, the joint angles of the robotic manipulators are read from the controllers of the robots. All these joint angle measurements are treated by the dynamic bounding volume algorithm in order to compute the minimum distance between the bounding volumes which cover the skeletons of the human operators and the manipulators. If the distance between two bounding volumes is smaller than a safety threshold (0.5m in this task), the manipulator will stop its normal behavior and will remain still until the distance between the

human and the robot is again greater than the safety threshold. This is the implemented safety strategy that verifies that there are no robot caused collisions between the manipulators and the human operator. The dynamic bounding volume algorithm has been implemented in a PC with an Intel Core 2 processor and 2GB of RAM and it has mean execution time of 0.0625s per frame.

In this task, the robotic manipulators are used to handle the tubes while the human operator screws the connectors which fix the new tubes to the structure. The collaboration of the human operator is needed because the screwing of the connectors is a difficult task to be performed by two manipulators. While the human operator collaborates with the manipulators, the safety strategy described in the previous paragraph is executed in order to guarantee the safety of the human. This human-robot interaction task can be divided in the following phases:

- Phase 1 (Figs. 5.a and 5.b): One of the manipulators extracts one metallic tube from a storage box in order to insert it in one of the free connectors of the metallic structure. The 3D representation of Fig. 5.b shows how the bounding volumes of this manipulator grow in size while it is moving quickly. The other manipulator handles a tube at a fixed position where the human operator will screw the connectors. This manipulator will not move during all the execution of the task and thus the safety strategy will not be applied to it because it does not perform dangerous movements for the human. Meanwhile, the human operator prepares the connectors for the metallic structure outside the manipulators' workspace.
- Phase 2 (Figs. 5.c and 5.d): The human operator enters in the manipulators' workspace in order to screw the first connector to the tube which is held by the second manipulator. The 3D representations of Figs. 5.c and 5.d show how the bounding volumes of the human operator change their size depending on their linear velocity. Meanwhile, the first manipulator continues approaching the metallic structure in order to insert the tube which has been extracted from the storage box in the previous phase. When the distance between the first manipulator and the human operator is smaller than the safety threshold (0.5m), this manipulator stops its motion (Fig. 5.d) in order to avoid any collision.
- Phase 3 (Figs. 5e. and 5f.): After having screwed the first connector to the tube, the human operator picks up the other connector and walks around the workspace until reaching the other end of the tube (Fig. 5.e). The first manipulator resumes its motion when the human goes away and the distance between them is greater than the safety threshold. This manipulator continues performing its task and inserts the tube it is carrying in the structure. Meanwhile, the human operator screws the second connector at the other end of the tube in order to fix it to the structure (Fig. 5.f). In this phase, the safety behavior is not activated because the distance threshold is not exceeded.





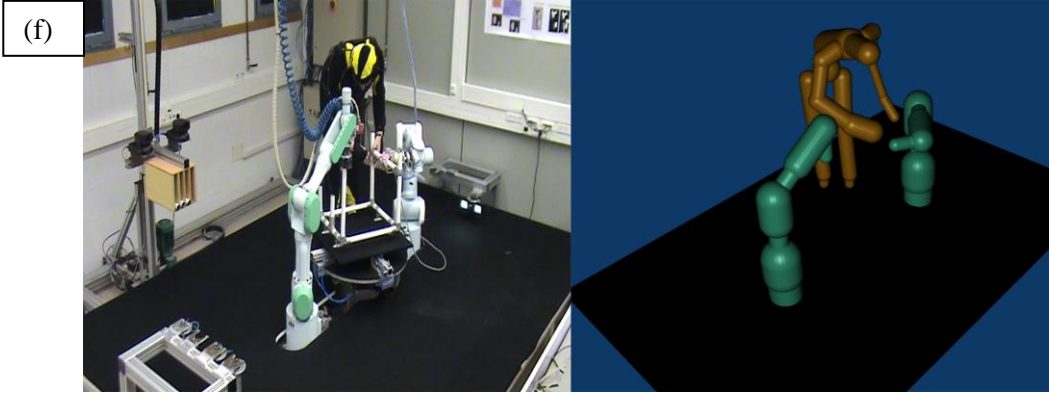


Fig. 5 Assembly task of a metallic structure based on the cooperation of two robotic manipulators and a human operator. Each photograph shows a phase in the assembly task. The 3D models on the right of each photograph depict the corresponding dynamic volumes (SSLs) of the human operator and the two manipulators.

The bounding volumes of the first manipulator and the human operator change their sizes during the execution of the task depending on the linear velocity of each link. The 3D representations of the bounding volumes in Fig. 5 show these changes at some concrete frames. This figure also shows the correspondence in each frame between the pose of the bounding volumes depicted on the right and the real pose of the photograph on the left. Nevertheless, there is a small difference between the real pose and the pose of the bounding volumes of the legs of the human in Figs. 5.c and 5.d because of the accumulation of some drift in the rotational measurements of the hips' node of the inertial motion capture system. This error is corrected in the subsequent frames when new measurements are registered and processed by the inertial motion capture system.

Figs. 6 and 7 depict a more detailed evolution of the radii of the bounding volumes. Only the changes of several bounding volumes have been represented for the sake of clarity. In particular, Fig. 6 shows the changes in the radii of the three last links of the first robotic manipulator. The plots in Fig. 7 show the changes in the radii of the left foot and the right hand of the human operator. These are the links with more radius variations because they are the terminal nodes of the skeleton with the maximum linear velocities. Phase 1 of the task last from 0s to 35s, phase 2 from 36s to 67s and phase 3 from 68s to 100s. The period in phase 2 when the manipulator performs the safety strategy can be identified in Fig. 6 because the radii of its bounding volumes do not change between 53s and 67s.

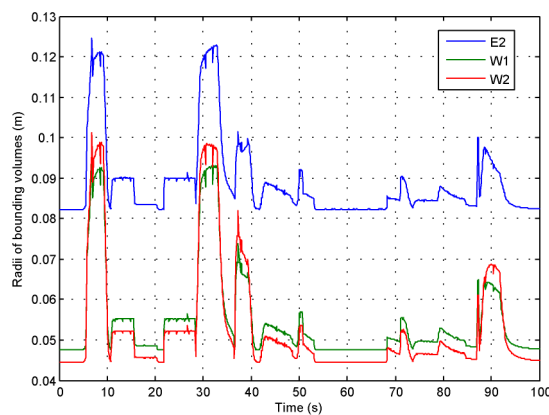


Fig. 6 Evolution of the radii of the bounding volumes which cover the last three links (E2, W1 and W2) of the first robotic manipulator.

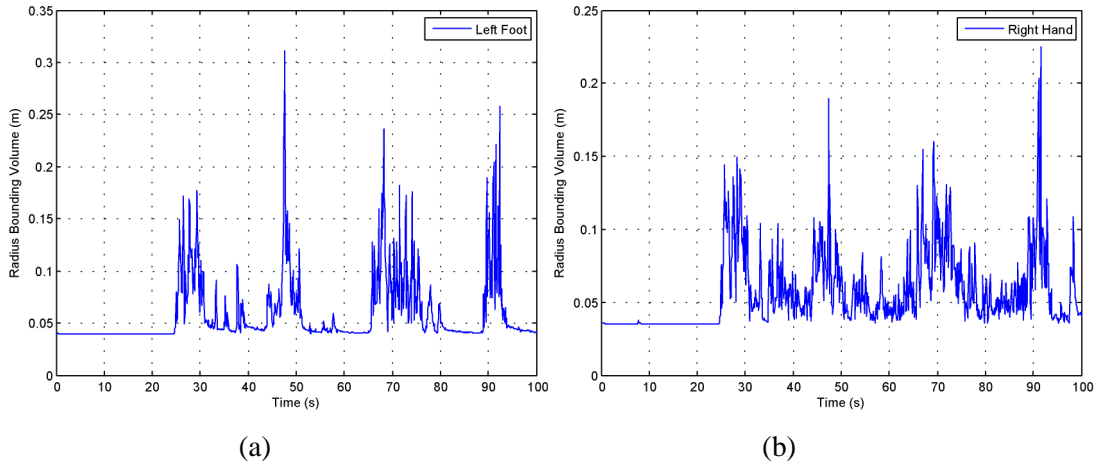


Fig. 7 Evolution of the radii of the bounding volumes which cover (a) the left foot and (b) the right hand of the human operator.

As stated in section 3, the geometric representation of the human and the robot with SSLs bounding volumes involves an improvement with regard to previous human-robot interaction systems [6-8] which use approaches based on spheres. In general, SSLs provide a better adaptation to the shape of human limbs and robotic links surfaces than spheres. This fact is demonstrated in Table 1, where the real volumes of the links of the PA10 robot are compared with the volumes of the SSLs and spheres bounding volumes. Thus, the geometric representation of the robot links based on spheres requires on average 64% of empty volume more than the representation based on SSLs. Because of this drawback, many approaches based on spheres use more than one sphere to cover each link more precisely. For instance, a minimum of 68 spheres are used for the human and 35 spheres for the robot in [6] while 46 spheres are required for the human and 41 spheres for the robot in [8]. The geometric representation based on SSLs described in this paper requires only 18 SSLs for the human and 8 SSLs for the robot. This reduction in the number of bounding volumes involves a reduction in the number of pairwise distance tests: 144 tests in comparison with 2380 tests in [6] and 1886 in [8]. Thereby, the performance of the minimum distance computation algorithm is improved with the SSL representation.

Table 1. Comparison of the space usage (volume) of the bounding volumes based on SSLs and spheres with regard to the real volumes of the links of the PA10 robot.

Robot Link	Real Volume (cm ³)	SSL Volume (cm ³)	Sphere Volume (cm ³)	Ratio Sphere-SSL Volume (%)
Base	7,798.50	13,550.42	15,250.58	11.15
S1	2,964.80	7,241.83	16,926.54	57.22
S2	4,590.20	6,410.59	31,176.00	79.42
S3	3,266.50	7,519.02	15,003.56	49.89
E1	1,806.20	2,579.58	16,115.26	83.99
E2	1,715.30	4,071.64	18,908.17	78.47
W1	720.08	808.64	7,470.30	89.18

6 Conclusions

This paper presents a geometric representation of the human body and the structure of robotic manipulators for the development of safe human-robot interaction tasks in industrial environments. This representation is composed of two main elements for each actor (robot or human): a skeleton and a set of bounding volumes. The skeleton models the kinematic structure of the actor by a hierarchy of linear links and rotational joints. Bounding volumes are required to cover the surface of each link of this skeleton. Typical sphere-based solutions are not convenient because a high number of small spheres is required in order to cover each link. The swept-sphere line representation described in this paper does not have this problem because these bounding volumes adapt better to the cylindrical shape of the majority of the links in robotic manipulators and human beings. In this representation, only one bounding volume per link is enough to cover with sufficient precision the surface of the links of the robot and the human operator.

The radii of these bounding volumes change during the development of the human-robot task depending on the linear velocity of the link which is encapsulated by each bounding volume. Thereby, the radius of each bounding volume does not only include the maximum size of the encapsulated link, but also the estimation of the distance which can be covered by the link during each sampling period. This paper describes in detail the novel algorithm which has been implemented in order to compute the dynamic radii of these bounding volumes. This dynamic bounding volume algorithm has been applied in the execution of a real task where a human operator and two robotic manipulators have cooperated in the assembly of a metallic structure. The dynamic bounding volume algorithm is implemented in order to calculate the minimum distance between the human operator and the robotic manipulator which is moving and activates a safety strategy if this distance is smaller than a pre-established threshold.

In future work, a hierarchy of bounding volumes could be developed. For instance, a set of AABBs which surround the current swept-sphere lines could be added. Therefore, only the swept-sphere lines which are inside the closest AABBs will be verified for the proximity queries and the number of pair-wise tests will be substantially reduced. This will facilitate the real-time implementation of the method. More complex safety strategies which involve the modification of the robot's trajectory on real-time could be also considered.

Acknowledgments

This work is funded by the Spanish Ministry of Education and the Spanish Ministry of Science and Innovation through the projects DPI2005-06222 and DPI2008-02647 and the grant AP2005-1458.

References

- [1] S. A. Green, M. Billingham, X. Chen, J. G. Chase, Human-robot collaboration: A literature review and augmented reality approach in design, *Int. J. Adv. Robot. Syst.* 5 (2008), 1-18.
- [2] D. Ebert, T. Komuro, A. Namiki, M. Ishikawa, Safe human-robot-coexistence: Emergency-stop using a high-speed vision-chip, in: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada, 2005, pp. 1821-1826.

- [3] J. A. Corrales, F. A. Candelas, F. Torres, Hybrid tracking of human operators using IMU/UWB data fusion by a Kalman filter, in: 3rd ACM/IEEE International Conference on Human-Robot Interaction, Amsterdam, The Netherlands, 2008, pp. 193-200.
- [4] G. Welch, E. Foxlin, Motion tracking: No silver bullet, but a respectable arsenal, *IEEE Comput. Graph. Appl.* 22 (2002), 24-38.
- [5] K. Steinbach, J. Kuffner, T. Asfour, R. Dillmann, Efficient collision and self-collision detection for humanoids based on sphere trees hierarchies, in: 6th IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS'06), Genoa, Italy, 2006, pp. 560-566.
- [6] B. Martinez-Salvador, M. Perez-Francisco, A. P. Del Pobil, Collision detection between robot arms and people, *J. Intell. Robot. Syst.* 38 (2003), 105-119.
- [7] J. Xu, D. K. Liu, G. Fang, An efficient method for collision detection and distance queries in a robotic bridge maintenance system, in: T.-J. Tarn, S.-B. Chen, C. Zhou (Eds), *Robotic Welding, Intelligence and Automation*, Springer Berlin Heidelberg, New York, 2007, pp. 71-82.
- [8] L. Balan, G. M. Bone, Real-time 3D collision avoidance method for safe human and robot coexistence, in: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 276-282.
- [9] P. Jimenez, F. Thomas, C. Torras, 3D collision detection: A survey, *Comput. Graph.* 25 (2001), 269-285.
- [10] S. Kockara, T. Halic, K. Iqbal, C. Bayrak, R. Rowe, Collision detection: A survey, in: *IEEE International Conference on Systems, Man and Cybernetics*, Montreal, QC, Canada, 2007, pp. 4046-4051.
- [11] S. Redon, M. C. Lin, D. Manocha, Y. J. Kim, Fast continuous collision detection for articulated models, *J. Comput. Inf. Sci. Eng.* 5 (2005), 126-137.
- [12] M. C. Lin, J. F. Canny, A fast algorithm for incremental distance calculation, in: *IEEE International Conference on Robotics and Automation*, Sacramento, Ca, USA, 1991, pp. 1008-1014.
- [13] B. Mirtich, V-Clip: Fast and robust polyhedral collision detection, *ACM Trans. Graph.* 17 (1998), 177-208.
- [14] C. J. Ong, E. G. Gilbert, Fast versions of the Gilbert-Johnson-Keerthi distance algorithm: Additional results and comparisons, *IEEE Trans. Robot. Autom.* 17 (2001), 531-539.
- [15] L. J. Guibas, D. Hsu, L. Zhang, A hierarchical method for real-time distance computation among moving convex bodies, *Comput. Geom.-Theory Appl.* 15 (2000), 51-68.
- [16] B. Gino van den, Efficient collision detection of complex deformable models using AABB trees, *J. Graph. Tools* 2 (1997), 1-13.
- [17] S. Gottschalk, M. C. Lin, D. Manocha, OBBTree: A hierarchical structure for rapid interference detection, in: *ACM SIGGRAPH Conference on Computer Graphics*, New Orleans, LA, USA, 1996, pp. 171-180.
- [18] E. Larsen, S. Gottschalk, M. C. Lin, D. Manocha, Fast distance queries with rectangular swept sphere volumes, in: *IEEE International Conference on Robotics and Automation*, San Francisco, CA, USA, 2000, pp. 3719-3726.

- [19] J. T. Klosowski, M. Held, J. S. B. Mitchell, H. Sowizral, K. Zikan, Efficient collision detection using bounding volume hierarchies of k-DOPs, *IEEE Trans. Vis. Comput. Graph.* 4 (1998), 21-36.
- [20] S. Cameron, Collision detection by 4-dimensional intersection testing, *IEEE Trans. Robot. Autom.* 6 (1990), 291-302.
- [21] K. Abdel-Malek, J. Z. Yang, Sweeping of an object held by a robotic end-effector, *Robot. Comput.-Integr. Manuf.* 21 (2005), 159-173.
- [22] B. Y. Maiteh, D. Blackmore, L. Abdel-Malek, M. C. Leu, Swept-volume computation for machining simulation and Virtual Reality application, *J. Mater. Process. Manuf. Sci.* 7 (1999), 380-390.
- [23] A. Schweikard, Polynomial-time collision detection for manipulator paths specified by joint motions, *IEEE Trans. Robot. Autom.* 7 (1991), 865-870.
- [24] S. Redon, A. Kheddar, S. Coquillart, An algebraic solution to the problem of collision detection for rigid polyhedral objects, in: *IEEE International Conference on Robotics and Automation*, San Francisco, CA, USA, 2000, pp. 3733-3738.
- [25] F. Schwarzer, M. Saba, J. C. Latombe, Adaptive dynamic collision checking for single and multiple articulated robots in complex environments, *IEEE Trans. Robot.* 21 (2005), 338-353.
- [26] K. Sims, D. Zeltzer, A figure editor and gait controller for task level animation, in: *SIGGRAPH Course Notes (Synthetic Actors: The Impact of Artificial Intelligence and Robotics on Animation)*, Atlanta, Georgia, USA, 1988, pp. 164-181.
- [27] J. Devanit, R. S. Hartenberg, A kinematic notation for lower-pair mechanisms based on matrices, *ASME J. Appl. Mech.* 22 (1955), 215-221.
- [28] V. Macagon, B. Wuensche, Efficient collision detection for skeletally animated models in interactive environments, in: *Image and Vision Computing New Zealand Conference*, Palmerston North, New Zealand, 2003, pp. 378-383.
- [29] C. Ericson, *Real-time collision detection*, Elsevier, San Francisco, 2005.
- [30] F. Janabi-Sharifi, V. Hayward, C. S. J. Chen, Discrete-time adaptive windowing for velocity estimation, *IEEE Trans. Control Syst. Technol.* 8 (2000), 1003-1009.