



HAL
open science

A Strictly Convex Hull for Computing Proximity Distances With Continuous Gradients

Adrien Escande, Sylvain Miossec, Mehdi Benallegue, Abderrahmane Kheddar

► **To cite this version:**

Adrien Escande, Sylvain Miossec, Mehdi Benallegue, Abderrahmane Kheddar. A Strictly Convex Hull for Computing Proximity Distances With Continuous Gradients. *IEEE Transactions on Robotics*, 2014, 30 (3), pp.666-678. 10.1109/TRO.2013.2296332 . hal-01079783

HAL Id: hal-01079783

<https://hal.science/hal-01079783v1>

Submitted on 4 Nov 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Strictly Convex Hull for Computing Proximity Distances With Continuous Gradients

Adrien Escande, Sylvain Miossec, Mehdi Benallegue, and Abderrahmane Kheddar, *Member, IEEE*

Abstract—We propose a new bounding volume that achieves a tunable strict convexity of a given convex hull. This geometric operator is named sphere-tori-patches bounding volume (STP-BV), which is the acronym for the bounding volume made of patches of spheres and tori. The strict convexity of STP-BV guarantees a unique pair of witness points and at least C^1 continuity of the distance function resulting from a proximity query with another convex shape. Subsequently, the gradient of the distance function is continuous. This is useful for integrating distance as a constraint in robotic motion planners or controllers using smooth optimization techniques. For the sake of completeness, we compare performance in smooth and nonsmooth optimization with examples of growing complexity when involving distance queries between pairs of convex shapes.

Index Terms—Bounding volume, continuous gradients of proximity distances, smooth and nonsmooth optimization, sphere-torus patches, strictly convex hulls.

I. INTRODUCTION

THE distance function and its use in collision-free robotic planning are remarkably addressed by the seminal work of Gilbert and Johnson [1], which provides a thorough analysis of its mathematical properties. In [1], motion planning is formulated as an optimization problem in which collision avoidance is defined as inequality constraints. Indeed, each collision avoidance constraint is written in terms of the Euclidian distance between the robot and the obstacles: this distance must always be above a certain positive threshold.

Manuscript received April 19, 2013; revised November 9, 2013; accepted December 19, 2013. Date of publication January 13, 2014; date of current version June 3, 2014. This paper was recommended for publication by Associate Editor D. Hsu and Editor C. Torras upon evaluation of the reviewers comments. This work was partially supported by the European Commission's Seventh Framework Program as part of the ROBOHOW.COG project under Grant 288533.

A. Escande is with the Centre National de la Recherche Scientifique-National Institute for Advanced Industrial Research and Technology, Intelligent Systems Research Institute, Joint Robotics Laboratory, UMI3218/CRT, Tsukuba 305-8568, Japan (CNRS-AIST JRL UMI3218/CRT), Japan (e-mail: adrien_escande@yahoo.fr).

S. Miossec is with PRISME Laboratory, University of Orléans, Bourges 18020, France (e-mail: Sylvain.Miossec@bourges.univ-orleans.fr).

M. Benallegue is with the Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) - CNRS, Toulouse cedex 4 BP 54200 31031, France (e-mail: mehdi.benallegue@gmail.com).

A. Kheddar is with the CNRS-AIST JRL, and the CNRS-University of Montpellier II, Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier (LIRMM), Montpellier 34095, France (e-mail: kheddar@gmail.com).

This paper has supplementary downloadable multimedia material available at <http://ieeexplore.ieee.org>. The video size is 3 715 701 bytes.

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

Digital Object Identifier 10.1109/TRO.2013.2296332

A particularly important theorem in [1]—that we rediscovered and proved differently in [2]—is that the distance function for a pair of objects is not continuously differentiable unless one of the objects is convex and the other is strictly convex. As we illustrate later, nondifferentiability of the distance function can be a problem when used in smooth optimization (see theoretical explanations in [3]): the optimization solver might not converge in some configurations. Then, why not use nonsmooth optimization solvers? This is an option. However, with respect to smooth optimization routines, nonsmooth ones are less easily available, less complete, and slower. Furthermore, it is worth adding distance constraints in existing schemes that already rely on smooth optimization routines (robotic models are generally smooth), or directly on function derivatives.

To still use smooth methods when they are beneficial, the idea is to enforce the continuous differentiability of the distance function. This is the subject of this paper. In [1], the problem is formulated in two dimensions using a rectangle representing the shape of a 3-degree-of-freedom (DOF) robot moving among fixed two-dimensional (2-D) convex obstacles. To add continuously differentiable collision avoidance constraints between the convex shapes of the robot and the obstacle, Gilbert and Johnson suggested the idea of lightly “bulging” the rectangle approximating the shape of the robot so as to enforce the strict convexity of its shape. *Thus far, the automation of the “bulging” and its extension to the 3-D case have still not been addressed: this is the core contribution of our study.*

We propose a solution for the 3-D case [2]: an automated and robust geometrical operator to construct a strictly convex shape of a given geometrical model. In other words, we present a method to *bulge* (i.e., round or curve) a polyhedral convex hull into a strictly convex hull/shape. Besides, our operator provides tunable bulging, ranging from the sphere bounding volume (i.e., the sphere with minimum radius encapsulating the entire object) to a polyhedral convex hull (i.e., the convex polyhedron with minimum volume that bounds the entire object). Our operator builds like a polyhedral convex hull, except that vertices are patches from small spheres of predefined radius, faces are patches from predefined bigger-radius spheres, and edges are patches from tori connecting adjacent features; therefore, we named it sphere tori patches (STP), and since it is a bounding volume, we added BV, yielding the name STP-BV.

Our second contribution is a study of the impact of STP-BVs when used for distance computation in optimization problems. We take three examples of increasing complexity to study their usefulness in terms of convergence success, speed, and precision. We compare 1) two smooth solvers without STP-BVs distance regularization; 2) the same two solvers with STP-BV;

and 3) a nonsmooth solver without STP-BV. The results show that the second option outperforms the others.

Many robotics problems require distance computation: motion planning, grasping, simulation, collision-avoidance, and design based on collision-free workspace computation. Smooth formulation of these problems could then benefit from the use of STP-BV. Examples of the use of STP-BVs can be found in [4] for planning, in [5] for trajectory optimization, and in [6] and [7] for control.

II. RELATED WORKS

There are many types of strictly convex shapes. Used as bounding volumes (covering at best the robot's links), they would guarantee the differentiability of the distance function. However, they either have a poor volume ratio or are time-consuming when used in distance computations. Spheres, ellipsoids, and superellipsoids are such strictly convex shapes. A single sphere or ellipsoid per link exhibits a weak volume ratio fitting, particularly for relatively thin and long robot links. The use of parts of these shapes to better fit each link increases the number of constraints to be added in the optimization problem. By combining several spheres, k -IOS [8] can fit the object tighter, yet the ratio might be no better than 0.5 in pathological cases. Superellipsoids or even hyperquadrics are also very good candidates, but computing distance from these shapes to others is too time-consuming [9] to be considered in control. Furthermore, the volume ratio fitting for a general convex shape cannot be made arbitrarily close to 1.

Few other papers addressed the nondifferentiability of the distance between convex bodies. Rusaw in [10] used nonsmooth analysis in the context of sensory-based planning. Patel *et al.* [11] used cylinders as BVs to cover a robot and acknowledged problems in the control when pairs of cylinders become parallel. They proposed solving this issue by thresholding the output speed of the robot. Recently, many authors acknowledged this problem when using capsules [12] or convex hulls in optimization problems [13].

To avoid collisions, pseudodistances can also be used, provided they have good differentiability properties. Distance fields such as in [14] are a possible solution. However, they remain computationally expensive as stated in [15] and need to be adapted for a pair of moving objects. Closer to the Euclidean distance, Zhu *et al.* [16] study the differentiability of a family of pseudodistances based on the gauge function of a convex polygon (thus encompassing the L_1 and L_∞ distances). These pseudodistances are differentiable almost everywhere in the same manner as the Euclidean distance. However, the sufficient condition for differentiability is not met in the same cases as those for the Euclidean distance as well as for additional cases (when a face of the Minkowski difference of the two objects is parallel to a face of the polygon defining the pseudodistance). Zhu *et al.* [16] do not provide a way to avoid these cases.

III. MOTIVATION

A. Example of the Distance Gradient's Discontinuity in Two Dimensions

Let us consider the segment depicted in Fig. 1. d is the distance from this segment to the horizontal axis, and we look at the

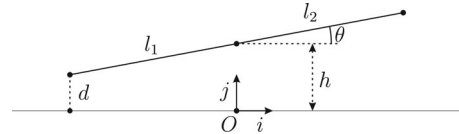


Fig. 1. Segment free to rotate around one of its points. d denotes the distance to the horizontal axis.

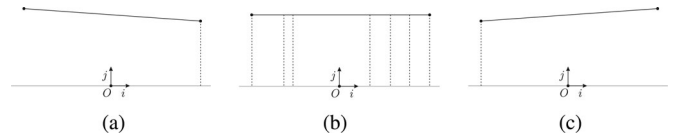


Fig. 2. When θ passes over 0 (b), the witness points jump between the configurations of (a) and (c). (a) $\theta < 0$. (b) Parallel case: $\theta = 0$. (c) $\theta > 0$.

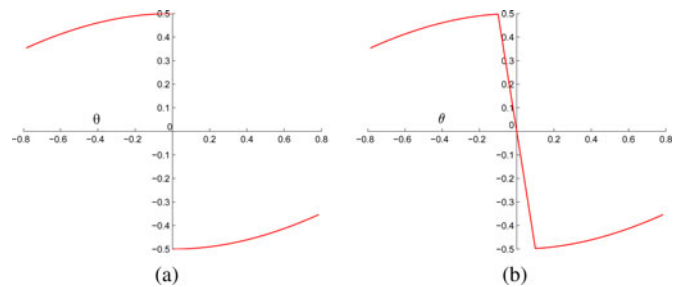


Fig. 3. Gradient $\partial d / \partial \theta$ of the distance between the bar (left) or its bulged version (see Fig. 4) (right) and the horizontal axis. In the first case the gradient is discontinuous at $\theta = 0$, while in the second case it varies quickly but continuously around $\theta = 0$. (a) Gradient discontinuity (see Fig. 1). (b) Regularized gradient (see Fig. 4).

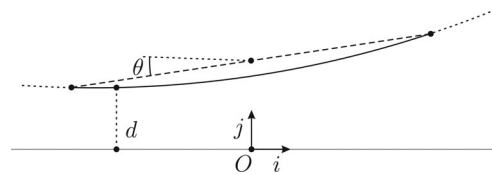


Fig. 4. Segment is replaced by a part of circle with a big radius. Now, when going over $\theta = 0$, the pair of witness points moves continuously.

gradient of d with respect to θ . This gradient [see Fig. 3(a)] is shown to be discontinuous in [10]. The discontinuity occurs when the segment becomes parallel to the horizontal axis: in this case, the witness point of the segment switches instantly between its two endpoints (see Fig. 2). This leads to a gradient discontinuity of d since both endpoints move in an opposite direction when θ changes. Such a point of discontinuity is named *kink point*.

B. Regularization of the gradient

To avoid this gradient discontinuity, the idea is to prevent jumps of the pair of witness points. This case occurs when *flat* parts of both objects' shapes become parallel. We remove the flat parts by bulging (i.e., curving) them. For instance, we replace the segment of our example by a part of a circle having a large radius (see Fig. 4).

The now unique witness point travels continuously between both endpoints and the gradient becomes continuous [see Fig. 3(b)]. This is the main idea behind the bounding volume

TABLE I
MEAN TIME PER ITERATION IN MICROSECONDS (WITH STANDARD DEVIATION)
FOR THE OPTIMIZATION PROBLEM (1)–(2)

SolvOpt	FSQP	NPSOL
1.3 ± 0.1	10.4 ± 0.4	39.3 ± 7.4

that we describe in Section V. Subsequently, bulging an object is called *gradient regularization* or *regularization* for short.

C. Efficiency of the Regularization

Let us consider the following optimization problem:

$$\min_{h, \theta} h^2 + (\theta - \theta_b)^2 \quad (1)$$

$$\text{s.t. } d(h, \theta) \geq d_0 \quad (2)$$

where h denotes the height of a 1-m bar center ($l_1 = l_2 = 0.5$ m in Fig. 1); θ , its orientation; $d_0 > 0$, a margin distance; and θ_b , the target orientation. We can show that this optimization problem converges to the kink point $(h, \theta) = (d_0, 0)$ for $-d_0 l_1 \leq \theta_b \leq d_0 l_2$. In the other cases, the solution is a regular point. We compare several strategies to solve this problem (and the other ones discussed in this paper): using smooth optimization routines with or without regularization, and using nonsmooth routines without regularization. For the former, we use two solvers: FSQP [17] and NPSOL.¹ These are SQP algorithms, a class of routines among the fastest for nonlinear smooth problems, with a superlinear convergence. For the latter, we chose SolvOpt [18]. Analytic center cutting plan methods, used, for example, by the optimization solver OBOE [19], deal with nonsmooth problems as well, but require the optimization problem to be convex, whereas a distance function provides a nonconvex constraint. Each solver has its own parameters for stopping criteria. Thus, we compare them by the number of iterations or computation time needed to obtain a desired precision of the solution (determined analytically when possible or by brute force). We consider two possibilities by choosing θ_b appropriately: convergence to a regular point or to a kink point. Each case study is optimized 100 times with the same random initial conditions, and the results are quantified as mean values. Computations are performed for different regularization radii (when applicable) and different algorithm parameter values. For a fair comparison, the best results in terms of computation time with respect to (wrt) the solution precision are kept. This corresponds to having tuned the solvers to obtain the best of their capabilities for the problem. Results are presented in Fig. 5. Using Table I, one can also obtain the mean results in terms of the computation time. All the computation times given in this paper are obtained on an Intel Core 2 Duo P8600 (3M Cache, 2.40 GHz).

Note the slower convergence to a kink point than to a regular one for all algorithms (except SolvOpt). For NPSOL, regularization is necessary to obtain convergence. For FSQP, surprisingly, convergence is obtained without regularization when using the Armijo line-search option. However, the regularization

improves slightly the convergence time. We observed that regularization allows approximately 20% less collision constraint evaluations. We have no explanation for the good behavior of FSQP for the nonsmooth case. It may be attributed to the special line-search designed to maintain feasible iterates. It would be worth studying this issue from the optimization viewpoint.

Overall, the solver for nonsmooth problems SolvOpt performs worse in terms of the number of iterations but better in terms of computation time than other solvers, because of a smaller computation time per iteration.

These results depend on the optimization problem (see others in Section VIII). We later see that the relative performance of SolvOpt degrades quickly with more complex problems.

This study also suggests an optimal choice of the regularization radius wrt the desired precision [see Fig. 5(c)]. The precision for a given radius depends on the body's shape: bodies with smaller edges and faces give a better precision for the same regularization radius. For a 1-m bar, to obtain a 10^{-n} precision of (h, θ) , the best choice is a circle of radius 10^n m. From the study of the relation between the obtained precision on h and the sphere radius, we observed that the radius providing the fastest convergence is linked to the regularization precision given by Theorem 6.2. Indeed, smaller radii give less precision (bigger bulging), but bigger radii result in stiffer problems, whose convergence is slower. Hence, the best choice of regularization radius is the smallest possible radius that fulfills the desired precision.

IV. DISTANCE CONTINUITY PROPERTIES

Gilbert and Johnson [1] demonstrated the necessity of having one body strictly convex and the other convex to obtain the differentiability of their separating distance function. We recall this theorem and complement it with additional properties.

A. Problem Definition and Notations

Definition 4.1 (Convexity and Strict Convexity): An object O is convex iff $\forall (X, Y) \in O^2$ and $\forall t \in]0, 1[$ the point P defined by $P = (1 - t)X + tY$ lies in O ; it is said to be *strictly convex* if $X \neq Y$ and P lies in $\text{int } O$, the interior of O .

Let δ be the distance between two convex objects O_1 and O_2 . The relative position between the two objects is locally parameterized by $q \in \mathbb{R}^6$.

We call *witness points* a pair of points of $O_1 \times O_2$ that are at the distance δ . Under certain conditions (see later Lemma 4.1), this pair is unique. Thus, we can define $p_{\min}(q) = (p_{1\min}(q), p_{2\min}(q))^T$ as the function that associates the pair of witness points with each q .

Furthermore, the surface of each object can be described by a function of two parameters. Let u be the 4-D vector of these 2×2 parameters, and r_1 and r_2 be these functions for O_1 and O_2 (to simplify, we define r_1 and r_2 as a function from a subset of \mathbb{R}^4 to \mathbb{R}^3 even though both of them are functions from a subset of \mathbb{R}^2 to \mathbb{R}^3).

The witness points being on the objects' surfaces, we define u_{\min} as the function of q that returns the vector u of these points. Let $R(q)$ be the rotation matrix,

¹http://www.sbsi-sol-optimize.com/asp/sol_product_npsol.htm

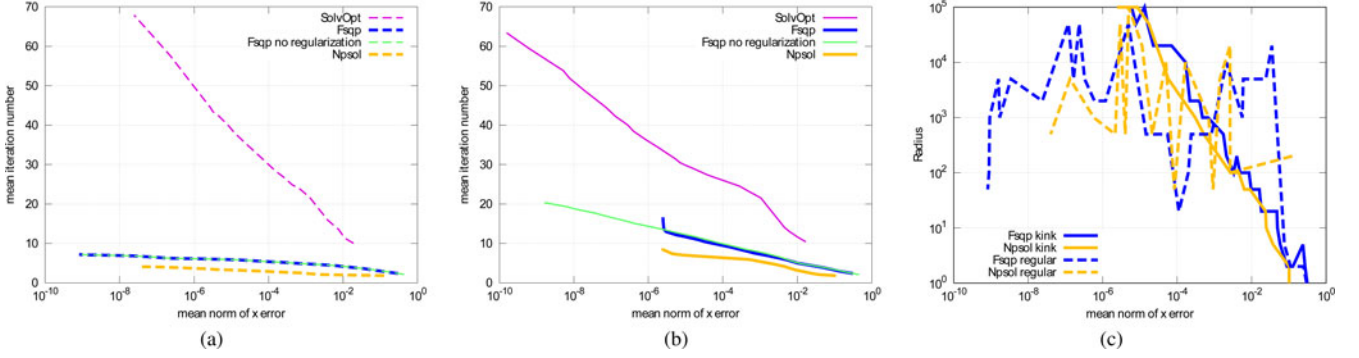


Fig. 5. Comparison of the number of iterations (ordinates) required to achieve a desired precision of the result (abscissa with $x = [h, \theta]$) for the optimization problem (1)–(2) with (a) $d_0 = 0.05$ and $\theta_b = 0.01$ or (b) $\theta_b = 0.02$, and (c) corresponding regularization radius. One can note that when converging to a regular point, FSQP behaves exactly the same with or without regularization. Furthermore, there is no real relation between the optimal radius and the precision when converging to a regular point, which is normal since regularization is not an issue at this point. (a) Convergence to a regular point. (b) Convergence to a kink point. (c) Best radius for a given precision.

and $T(q)$ the translation vector; we have $p_{\min}(q) = (r_1(u_{\min}(q)), R(q)r_2(u_{\min}(q)) + T(q))^T$.

Finally, $f(q, u) = r_1(u) - R(q)r_2(u) - T(q)$ is the distance vector between the two witness points parameterized by u

$$\delta(q) = \min_u \|f(q, u)\| = \|f(q, u_{\min}(q))\|. \quad (3)$$

Note that f is C^∞ wrt q since R and T are rotation and translation, and that if r_1 and r_2 are C^k wrt u , so is f .

B. Strict Convexity of a Body

Gradient discontinuity occurs at a jump of the witness points (see Section III). This happens around a configuration in which the witness pair is not unique, e.g., when an edge is parallel to a face. If one object is strictly convex, parallelism does not happen. The following successive lemmas and theorems demonstrated in [1] link the uniqueness of witness points (hence the continuity of the gradient) with the strict convexity of one object. Thus far, they are valid for $\delta > 0$.

Lemma 4.1 (Uniqueness of witness points): There is a unique pair of witness points if at least one of the bodies is strictly convex (and the other is convex).

Lemma 4.2: The witness points of the minimum distance between two convex bodies are continuous functions of q if at least one of the bodies is strictly convex and the other is convex (u_{\min} and p_{\min} are continuous functions of q).

Theorem 4.3: The minimum distance between two convex bodies is a C^1 function of q if at least one of the bodies is strictly convex and the other is convex.

If the surfaces of these objects have additional continuity properties and $\frac{\partial^2 f^2}{\partial u^2}$ is invertible (which is always the case but for pathological cases), the distance function becomes smoother.

Lemma 4.4: If the surfaces of both bodies are C^k , with $k \geq 2$, and one body is strictly convex, then the witness points are C^{k-1} functions of q .

Proof: Let u_0 be the coordinates of the witness points at q_0 . We have $\left(\frac{\partial f}{\partial u}(q_0, u_0)\right)^T f(q_0, u_0) = 0$ [optimality condition of

the minimization problem in (3)] which can be rewritten as $\frac{\partial f^2}{\partial u}(q_0, u_0) = 0$.

Let $F(q, u) = \frac{\partial f^2}{\partial u}(q, u)$. F is C^{k-1} , $F(q_0, u_0) = 0$ and $\frac{\partial F}{\partial u}(q_0, u_0)$ is invertible. Thus, u is locally a C^{k-1} function of q (implicit functions theorem). Subsequently, u_{\min} is a C^{k-1} function of q . ■

Theorem 4.5: If the surfaces of both bodies are C^k , with $k \geq 2$, and one body is strictly convex, then the minimum distance between them is C^k .

Proof: Let us derive $\delta(q) = \|f(q, u_{\min}(q))\|$ wrt q

$$\begin{aligned} \frac{\partial \delta}{\partial q}(q) &= \left(\frac{\partial f}{\partial q} + \frac{\partial u_{\min}}{\partial q} \cdot \frac{\partial f}{\partial u} \right)^T \frac{f}{\|f\|} \\ &= \left(\frac{\partial f}{\partial q}(q, u_{\min}(q)) \right)^T \frac{f(q, u_{\min}(q))}{\|f(q, u_{\min}(q))\|} \end{aligned} \quad (4)$$

since $\left(\frac{\partial f}{\partial u}(q, u_{\min}(q))\right)^T f(q, u_{\min}(q)) = 0$.

Using Lemma 4.4, the demonstration is straightforward. ■

To sum up, the continuity properties of the minimum distance function with respect to the relative position of two convex bodies are as follows.

- 1) The distance is always C^0 (even with no convexity assumptions).
- 2) It is piecewise C^1 with simple convexity. Discontinuities of the gradient arise when faces or edges are parallel (non-strict convexity).
- 3) Strict convexity of a body and C^0 surfaces ensures the C^1 property of the distance. Having C^1 surfaces does not improve the continuity of the distance as compared to having C^0 surfaces,
- 4) Additional C^k property of both surfaces yields C^k smoothness of the distance if $\frac{\partial^2 f^2}{\partial u^2}$ is invertible.

These properties are also true locally. In particular, if witness points move on the interior of C^∞ surfaces, one at least being strictly convex, the distance is C^∞ . This is the case when considering the distance between two spherical parts.



Fig. 6. For two objects in interpenetration (one in the plain line, one in the dash line), a degenerate case where there are an infinity of witness pairs, some of which are depicted in gray.

C. Interpenetration Case

The C^n property for $n > 0$ cannot be reached everywhere in the interpenetration case; the distance minimization problem is not convex anymore. Indeed, in some configurations there are several (up to infinite, see Fig. 6) pairs of witness points. Jumps between witness pairs are thus inevitable resulting in gradient discontinuities. However, we can retain the results of the aforementioned theorems for a subset of penetration cases. But first we need to ensure the continuity properties between the penetration and the nonpenetration case. We define the penetration distance as the opposite of the distance between the pair of points verifying the optimality condition $(\frac{\partial f}{\partial u})^T f = 0$ while being at the minimal distance among the possible pairs with opposite normal vectors (which is the same definition as in the nonpenetration case). This definition is equivalent to the minimum separating translation distance (see [20, Ch. 2]). Jumps between pairs do not happen if the penetration depth is less than twice the minimal radius of curvature of the penetrating parts of the objects. Under this assumption of “slight” penetration, the previous results hold; if not we may then have gradient discontinuities. In the worst case, we may encounter the discontinuities for $\delta = 0$ when at least one witness point is on a vertex and the other on an edge. These are cases of bifurcation of the number of witness pairs. However, these discontinuities will be avoided when a security margin is considered for collision avoidance [1], which is the case for our bounding volume (see the next section). Therefore, discontinuities will only occur in strict penetration cases, which are repulsive situations: the gradient points away from them. Hence, they should not be a problem in most applications where penetration can occur (like in optimization starting with an unfeasible point).

V. SPHERE-TORUS-PATCH BOUNDING VOLUMES

A. Definition:

We consider a point cloud \mathcal{P} , and, for $r \geq 0$, the set \mathcal{P}_r of (closed) balls $\mathfrak{B}(P, r)$ centered at each $P \in \mathcal{P}$ and with radius r ($\mathcal{P}_0 \equiv \mathcal{P}$). Let $\mathcal{B}_{R,r}(\mathcal{P})$ be the set of all balls with radius R that encloses every ball of \mathcal{P}_r ($R - r$ must be at least the radius of the smallest sphere containing \mathcal{P}).

Definition 5.1 (STP-BV) We name STP-BV of \mathcal{P} for the radii R , and r the intersection of all the balls of radius R containing the balls of radius r centered at each point of \mathcal{P}

$$\text{STP}_{R,r}(\mathcal{P}) = \bigcap_{B \in \mathcal{B}_{R,r}(\mathcal{P})} B. \quad (5)$$

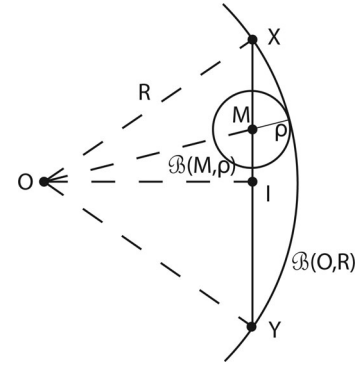


Fig. 7. Figure for Lemma 5.1.

Henceforth, we ignore the special case where there is a single point in \mathcal{P} and $r = 0$ (therefore, the STP-BV has a nonempty interior).

B. Strict Convexity

We now show that an STP-BV is strictly convex, which is not straightforward. Indeed, the intersection of an infinite number of convex volumes is always convex, but the intersection of an infinite number of strictly convex volumes is not necessarily strictly convex. We prove with the following lemma and theorem that the strict convexity is retained in the case of STP-BV. The lemma shows that for a point strictly on a segment, a minimal distance to the boundary of any ball with radius R containing this segment can be guaranteed.

Lemma 5.1: Let X and Y be two points such that $X \neq Y$ and $XY \leq 2R$. Let $t \in]0, 1[$ and define $M = (1 - t)X + tY$ and $\rho = R - \sqrt{R^2 - t(1 - t)XY^2}$. Then, for any ball $\mathfrak{B}(O, R)$ that contains X and Y , $\text{int } \mathfrak{B}(M, \rho) \subset \text{int } \mathfrak{B}(O, R)$ and $\text{int } \mathfrak{B}(M, \rho)$ is not empty.

Proof: O being not fixed, the longest distance between O and M is achieved when O is such that X and Y are both on the surface of $\mathfrak{B}(O, R)$: $OX = OY = R$.² We consider this case and we denote the middle point of the segment $[X, Y]$ as I (see Fig. 7). The triangle OIX is orthogonal at I so that $OI^2 = R^2 - \frac{1}{4}XY^2$. Then considering the triangle OIM , orthogonal at I , we have $OM^2 = R^2 - t(1 - t)XY^2$. We have $t(1 - t)XY^2 > 0$ since $t \in]0, 1[$ and $X \neq Y$, so that $OM < R$.

In the general case (i.e., $OX \leq R, OY \leq R$), we have $OM \leq \sqrt{R^2 - t(1 - t)XY^2} < R$ so that whatever O, M is at least at a distance $\rho > 0$ of the boundary of $\mathfrak{B}(O, R)$. ■

Theorem 5.2 (Strict Convexity of STP-BV): $\text{STP}_{R,r}(\mathcal{P})$ is strictly convex.

Proof: Let us show that for any $X \neq Y$ of the STP-BV and any $t \in]0, 1[$, $(1 - t)X + tY \in \text{int } \text{STP}_{R,r}(\mathcal{P})$. Let $X \neq Y$. If $X, Y \in \text{STP}_{R,r}(\mathcal{P})$, then for every B of $\mathcal{B}_{R,r}(\mathcal{P})$, $X, Y \in B$. In particular, $XY \leq 2R$. We define t, M , and ρ as in Lemma 5.1. Then, $\forall B \in \mathcal{B}_{R,r}(\mathcal{P})$, $\text{int } \mathfrak{B}(M, \rho) \subset \text{int } B$; thus $\text{int } \mathfrak{B}(M, \rho) \subseteq \text{int } \text{STP}_{R,r}(\mathcal{P})$ so that $M \in \text{int } \text{STP}_{R,r}(\mathcal{P})$. ■

²This is quite intuitive, but it can be proven by writing the maximization of OM wrt O , with the constraints $OX \leq R$ and $OY \leq R$.

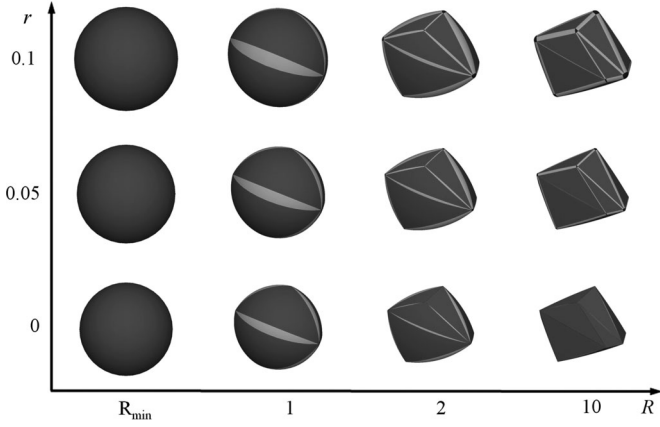


Fig. 8. STP-BV bulge tuning with the two radii (R, r) . The parts of big spheres are depicted in dark gray, the tori in lighter gray, and the small spheres in black. R_{\min} denotes the radius of the smallest bounding sphere. In this example, it is $0.87 + r$. For $(R, r) \rightarrow (\infty, 0)$, the STP-BV tends to the convex hull of the original polyhedron (bottom right). An increase in r has a dilating effect, whereas a decrease in R from the convex hull $(\infty, 0)$ to the enclosing sphere $(R_{\min}, 0)$ achieves continuous bulging.

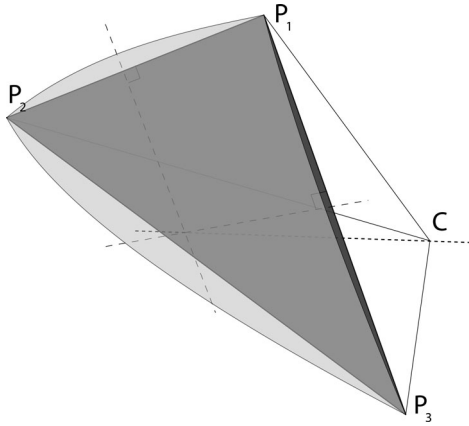


Fig. 9. Sphere construction. C is equidistant from each P_i , which means its projection is the circumcenter of the face (at the intersection of the edge); perpendicular bisectors are depicted by thin dash lines.

C. Surface

Here, we describe the surface of an STP-BV. The following theorem helps us reducing the amount of work.

Theorem 5.3 (Dilation): $STP_{R,r}(\mathcal{P})$ is the dilation of $STP_{R-r,0}(\mathcal{P})$ by a ball of radius r .

Proof: This stems trivially from the fact that if X is in a ball of radius $R - r$, then $\mathfrak{B}(X, r)$ is included in the ball having the same center, and radius R , and conversely. ■

For presenting how to build an STP-BV surface, we take $r = 0$. For the sake of simplicity, we write R instead of $R - r$.

A ball $B = \mathfrak{B}(C, R) \in \mathcal{B}_{R,0}(\mathcal{P})$ really limits the STP-BV if it is tangent to one or more points of \mathcal{P} . The following three cases are observed.

- 1) B is tangent to three or more points P_i (see Fig. 9). The part of the corresponding sphere, which is inside the cone defined by the vectors CP_i and with apex C , is a part of the boundary of the STP-BV.

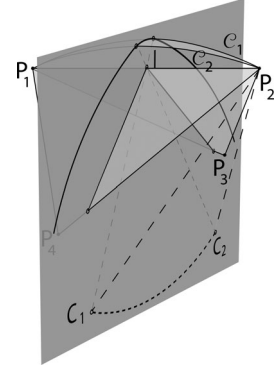


Fig. 10. Torus construction. The gray plane indicates the median plane of the edge P_1P_2 , on which are depicted its intersection with the incident faces, the corresponding spheres, and the torus in between. The circles C_1 and C_2 denote the limits between the torus and the spheres (not depicted here) and generate this torus by rotation around (P_1P_2) .

- 2) B is tangent to two points P_1 and P_2 . The geodesic arc on B from P_1 to P_2 is part of the boundary of the STP-BV. Let B_1 be a ball tangent to P_1, P_2 and an additional point P_3 . All the balls of $\mathcal{B}_{R,r}(\mathcal{P})$ tangent to P_1 and P_2 are obtained by rotating B_1 around P_1P_2 and away from P_3 until it becomes tangent to a new point P_4 . The corresponding geodesic arcs form a part of a torus (see Fig. 10), whose limits are the planes $C_1P_1P_2$ and $C_2P_1P_2$, where C_1 denotes the center of B_1 and C_2 denotes the center of the ball tangent to P_1, P_2 , and P_4 .
- 3) B is tangent to only one point. It limits the STP-BV only at this point.

The surface of the STP-BV is thus made of parts of spheres and tori, hence the name sphere-torus-patch bounding volume. The sphere parts can be seen as the “faces” of the volume and the tori as its “edges.” The vertices are the points of \mathcal{P} that define the axes of the tori. When $r > 0$, the vertices become parts of small spheres with radius r ; see Fig. 8.

By definition, $STP_{R,0}(\mathcal{P})$ is the smallest bounding volume built with balls of radius R . $\mathcal{H}(\mathcal{P})$ being the convex hull of \mathcal{P} , we have $\mathcal{H}(\mathcal{P}) \subset STP_{R,0}(\mathcal{P})$. The parameter R denotes the maximal radius of curvature of the STP-BV and controls the boundary on the maximal margin between the STP-BV and $\mathcal{H}(\mathcal{P})$ (see Theorem 6.2). When R tends to infinity, $STP_{R,0}(\mathcal{P})$ tends to $\mathcal{H}(\mathcal{P})$. The STP-BV can thus be made arbitrarily close to a convex polyhedron. When used, the parameter r gives the minimal distance between $STP_{R,r}(\mathcal{P})$ and $\mathcal{H}(\mathcal{P})$. It is used as a security margin. Both R and r are set by the user.

VI. GEOMETRIC CONSTRUCTIONS

The vertices and torus axes of $STP_{R,0}(\mathcal{P})$ define a polyhedron $\mathcal{H}_R(\mathcal{P})$ (usually but not necessarily convex). To each $B \in \mathcal{B}_{R,0}(\mathcal{P})$ tangent to three or more points of \mathcal{P} corresponds a polygonal face of $\mathcal{H}_R(\mathcal{P})$ defined by these points (the face may not be planar in the rare case of nonplanar cospherical points). If a face has more than three points, we divide it into triangles. The

new edges correspond to flat tori. Obviously, $\mathcal{H}_R(\mathcal{P}) \subseteq \mathcal{H}(\mathcal{P})$. We show how to build spheres and tori from edges and faces and prove some properties.

A. Sphere Construction

Let us consider a triangular face of $\mathcal{H}_R(\mathcal{P})$ (see Fig. 9). P_1, P_2 , and P_3 are its vertices given counterclockwise around the outer normal.

We want to build the (unique) sphere of center C and radius R that goes through the three vertices of the face, and is *above* the face (direction given by the outer normal).

Let $u = P_1P_2, v = P_1P_3, c = P_1C$, and $w = u \times v$. w is collinear to the outer normal and points in the same direction.

C needs to be equidistant to the points P_i and is thus on the median planes of the edges. We then solve the following system to find the coordinates of C in the frame (P_1, u, v, w) :

$$\begin{cases} u \cdot c = u^2/2 \text{ (median plan of } P_1P_2) \\ v \cdot c = v^2/2 \text{ (median plan of } P_1P_3) \\ c^2 = R^2 \\ c \cdot w < 0 \text{ (inner solution).} \end{cases} \quad (6)$$

Writing $c = \alpha u + \beta v + \gamma w$, we get from the two first equations

$$\alpha = \frac{u^2v^2 - u \cdot v \cdot v^2}{2(u \times v)^2}, \quad \beta = \frac{u^2v^2 - u \cdot v \cdot u^2}{2(u \times v)^2}.$$

These are the coordinates of the circumcenter of the triangle. Replacing α and β in the third equation leads to an equation of degree 2 in γ . Only one of its two solutions complies with the fourth equation: $\gamma = -\sqrt{\frac{R^2 - \alpha^2u^2 - \beta^2v^2 - 2\alpha\beta u \cdot v}{(u \times v)^2}}$.

Of the sphere centered on this point C , we only keep the part inside the cone defined by vectors CP_i and whose apex is C .

B. Torus Construction

We obtain the torus above one edge of $\mathcal{H}_R(\mathcal{P})$ by rotating the sphere of an incident face around this edge and keeping the resulting inner volume. The construction, depicted in Fig. 10, is based on the following result.

Lemma 6.1: The distance between the center of a sphere corresponding to a face and the median point I of one of its edges depends only of the length l of this edge and is $\sqrt{R^2 - \frac{l^2}{4}}$.

Proof: With the notation of Fig. 10, it is the direct result of the pythagorean theorem written for the triangle (IC_1P_2) . ■

The centers C_1 and C_2 of the two spheres corresponding to the two incident faces of the edge that we consider are thus on a same circle with center I and radius $\sqrt{R^2 - \frac{l^2}{4}}$.

We consider the circle \mathcal{C}_1 of center C_1 and radius R in the plane defined by C_1 and the edge. By construction, this circle coincides with the sphere centered in C_1 .

By making this circle revolve around the edge until it coincides with the circle \mathcal{C}_2 of same radius centered in C_2 and in the plane defined by C_2 and the edge, we obtain the part of torus we need. Note that the torus does not have the usual shape of a donut since its usual small radius is bigger than the (usual) big one. The part that we consider here is on the inner side of

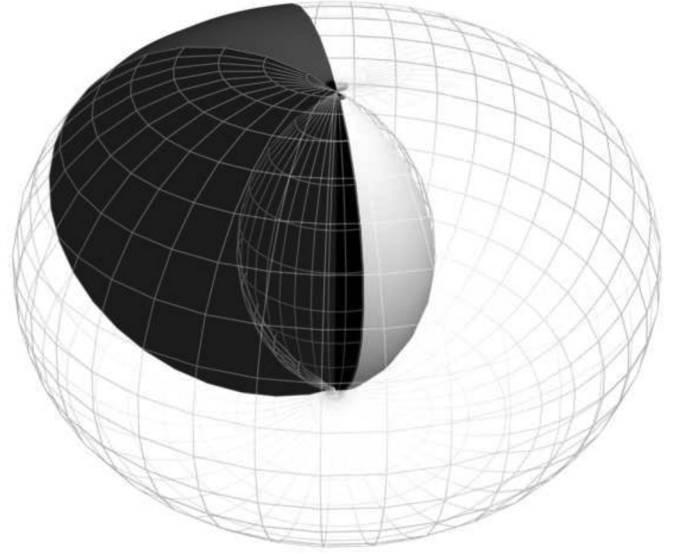


Fig. 11. Black and white parts are the portions of torus that we consider; the white surface is the part used in the STP-BV.

the entire torus, as shown in white in Fig. 11. The torus and the spheres coincide in their delimiting planes and are perpendicular to these planes. Therefore, the torus is tangent to both spheres, and the junction between the torus and each sphere is then C^1 .

C. Properties

Theorem 6.2 (Maximal margin): If a denotes the length of the longest edge of the polyhedron, then the maximal margin between $\mathcal{H}_R(\mathcal{P})$ and $\text{STP}_{R,0}(\mathcal{P})$ is $R - \sqrt{R^2 - \frac{a^2}{12}}$.

Proof: Since each edge of $\mathcal{H}_R(\mathcal{P})$ is the revolution axis of its associated torus, the maximal margin is achieved in its median plane, and is equal to $R - \sqrt{R^2 - \frac{l^2}{4}}$, l being the length of the edge.

For a face, the following two cases are observed with respect to the position of the circumcenter H :

- 1) H is outside of the face, and then, the maximal margin is achieved on the longest edge of this face;
- 2) H is inside the face. In this case, for a fixed longest length of the edges, the maximal margin is achieved when the face is equilateral and is $R - \sqrt{R^2 - \frac{a^2}{12}}$. ■

Since $\mathcal{H}_R(\mathcal{P}) \subseteq \mathcal{H}(\mathcal{P})$, this is also the maximal margin to $\mathcal{H}(\mathcal{P})$.

Theorem 6.3 (Continuity property of STP-BV): When $r > 0$, the surface of the STP-BV is C^1 .

Proof: (sketch) For $r = 0$, the STP-BV is convex and its surface is C^1 everywhere but at its vertices. A dilation of $r > 0$ makes it round at the vertices, thus C^1 everywhere. ■

An STP-BV is even piecewise C^∞ since the tori and the spheres are C^∞ surfaces. However, as stated in Theorem 4.5, there is no advantage in having C^1 surfaces instead of C^0 with respect to the distance continuity.

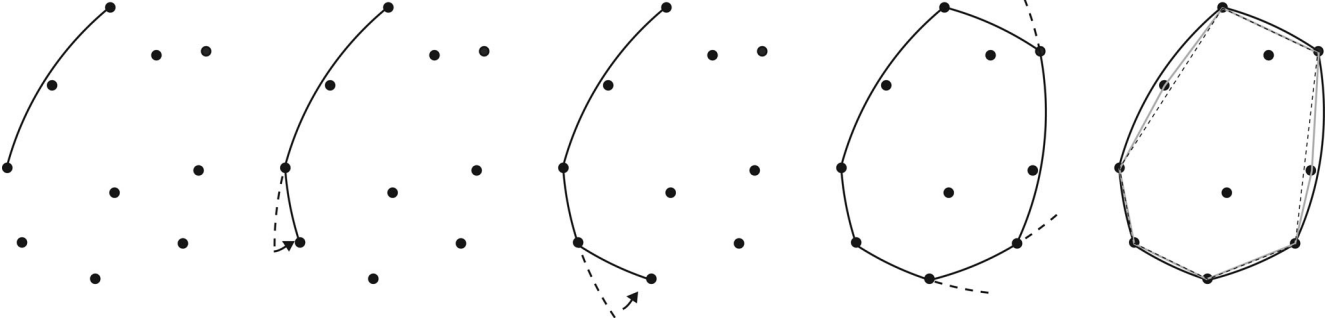


Fig. 12. Construction of the STP-BV of a cloud \mathcal{P} in a 2-D case by rotation of an initial circle (left) around the points as it encounters them. Some points of the convex hull $\mathcal{H}(\mathcal{P})$ (gray polygon) are dismissed due to the curvature of the circle. The STP-BV polygon $\mathcal{H}_R(\mathcal{P})$ is depicted by a thin dashed line (right).

Algorithm: STP-BV construction: Pseudo-code

Data: point cloud \mathcal{P} , value of R

Result: the STP-BV polyhedron $\mathcal{H}_R(\mathcal{P})$

- v, v_1, v_2 and v_3 are vertices

- c, c_1 and c_2 are sphere centers

- $next$: is a list of sets $(v_1, v_2, v_3, (v, a))$, sorted by increasing value of a .

-output: a structure describing $\mathcal{H}_R(\mathcal{P})$

BuildVolume(\mathcal{P}, R)

begin

$init(next, output)$

while no $next.isEmpty()$ **do**

$(v_1, v_2, v_3, (v, a)) \leftarrow next.pop()$

$output.addFace(v_1, v_2, v_3)$

if $next.contains(v_1, v_2)$ **then**

$next.delete(v_1, v_2)$

else

$next.insert(v_1, v_2, v_3, nextPoint(v_1, v_2, v_3))$

if $next.contains(v_2, v_3)$ **then**

$next.delete(v_2, v_3)$

else

$next.insert(v_2, v_3, v_1, nextPoint(v_2, v_3, v_1))$

return output

Algorithm: intermediate functions: Pseudo-code

init($next, output$)

begin

for each $(v_1, v_2, v_3) \in \mathcal{P}^3$ with $v_i \neq v_j$ **do**

$c \leftarrow sphere(v_1, v_2, v_3, R)$

if $\mathcal{P} \subseteq \mathcal{B}(c, R)$ **then**

$output.addFace(v_1, v_2, v_3)$

$next.insert(v_1, v_2, v_3, nextPoint(v_1, v_2, v_3))$

$next.insert(v_2, v_3, v_1, nextPoint(v_2, v_3, v_1))$

$next.insert(v_3, v_1, v_2, nextPoint(v_3, v_1, v_2))$

return SUCCESS

return FAIL

nextPoint(v_1, v_2, v_3)

begin

for each $v \in \mathcal{P} - \{v_1, v_2\}$ **do**

$c_2 \leftarrow sphere(v_1, v_2, v, R)$

if $\mathcal{P} \subseteq \mathcal{B}(c_2, R)$ **then**

if $v = v_3$ **then**

return (v, π)

else

$c_1 \leftarrow sphere(v_1, v_2, v_3, R)$

return $(v, angle(c_1, c_2, v_1, v_2))$

VII. SPHERE-TORI-PATCHES BOUNDING VOLUME CONSTRUCTION

There is by definition a one-to-one relation between the spheres and tori of the $STP_{R,0}(\mathcal{P})$ and the faces and edges of $\mathcal{H}_R(\mathcal{P})$. The construction of $\mathcal{H}_R(\mathcal{P})$ from \mathcal{P} is described in Algorithm 1. It is mostly a *gift wrapping algorithm* with spheres instead of planes (see Fig 12)³; we find an initial face whose associated sphere contains \mathcal{P} . We then rotate the sphere around the edges of the face until it reaches new points of \mathcal{P} . This defines new faces and new edges around which to rotate.

The *init* function (Algorithm 2) finds an initial face. Since we can perform only one rotation at a time, we maintain a list of rotations to be completed. Maintaining this list *next* is the main task of Algorithm 1. For a reason we provide later, the smallest of the rotations in this list is performed first. An element of *next* is a set $(v_1, v_2, v_3, (v_4, a))$ encoding that the (positive) rotation of the sphere associated with the face $v_1 v_2 v_3$ around the edge $v_1 v_2$ reaches the point v_4 after a rotation of angle a (in Fig. 10, a is the angle between IC_1 and IC_2).

The list *next* is ordered according to the angles a . It supports the classical operations *pop* and *insert*. The operation *contains*(v_1, v_2) returns true if any element of *next* begins by (v_1, v_2) or (v_2, v_1) . *delete*(v_1, v_2) removes such an element. *output* is a structure describing a polyhedron (sets of vertices, edges, and faces). The function *addFace*(v_1, v_2, v_3) updates it by adding the face $v_1 v_2 v_3$ (described counterclockwise) and the corresponding edges and vertices, if not already present. *sphere*(v_1, v_2, v_3, R) returns the center of the sphere with radius R as computed in Section VI-A. *angle*(c_1, c_2, v_1, v_2) returns the angle between ic_1 and ic_2 , where i is the middle point of $v_1 v_2$; *nextPoint*(v_1, v_2, v_3) looks for the first point v encountered while rotating the sphere associated with $v_1 v_2 v_3$ around $v_1 v_2$, and returns it along with the rotation angle made to reach it. v denotes the only point for which the sphere associated with $v_1 v_2 v_3$ contains \mathcal{P} , but when there are cospherical points, in which case any of these points will do. The special case where $v = v_3$ is discussed later.

At each iteration, Algorithm 1 processes a new face $v_1 v_2 v_3$ obtained by turning around $v_1 v_2$. For the two edges $v_1 v$ and $v v_2$ the following two cases can be observed.

³See also the supplementary multimedia material.

- 1) The edge was encountered before when processing a face f . There is no need to rotate around this edge anymore because rotating the sphere associated with f around the edge would give $v_1 v v_2$ and vice versa. The rotation is removed from *next*.
- 2) The edge was not encountered before. We need to perform the rotation around it later so it is added to *next*.

The output of Algorithm 1 is $\mathcal{H}_R(\mathcal{P})$ from which, with the computations described in the previous section, we can build the boundary of $\text{STP}_{R,0}(\mathcal{P})$ (spheres and tori with their limits) and thus any $\text{STP}_{R+r,r}(\mathcal{P})$.

The reason for choosing the smallest rotations first is to ensure the robustness of the algorithm wrt numerical errors in the case of polygonal faces whose vertices are on the same sphere (cospherical). In this case, all vertices should be reached at the same time when turning around an edge of this face, and thus, the vertex with the lowest index is selected. However, this is sometimes not the case because of numerical rounding errors. Thus, different vertices can be chosen when we reach the face by turning around different edges, resulting most of the times in overlapping triangles, which results in an algorithm failure. To avoid this, we force the algorithm to finish covering a polygonal face that it already began, by choosing to turn around the edge with the lowest rotation. This avoids us setting a threshold that defines when points are cospherical and searching for cospherical points each time there is a rotation around an edge.

In special cases, it can happen that the rotation of the face $v_1 v_2 v_3$ around $v_1 v_2$ stops at $v = v_3$. For example, this occurs when \mathcal{P} has only three points, but it can happen with more.⁴ In this case, after turning around $v_1 v_2$, the algorithm will remove the rotations around $v_1 v_3$ and $v_2 v_3$ from *next*, while one of them might still be needed. To avoid this, *nextPoint* forces the angle to π when detecting the case, to ensure that the corresponding rotation will be the last one processed.

Finally, *init* fails in two cases: 1) if R is smaller than the radius of \mathcal{P} ($\text{STP}_{R,0}(\mathcal{P})$ does not exist), and 2) in the case of thin long clouds, where no ball of $\mathcal{B}_{R,0}(\mathcal{P})$ is tangent to three points of \mathcal{P} . In the latter case, $\text{STP}_{R,0}(\mathcal{P})$ is the torus whose axis is defined by the two furthest points of \mathcal{P} .

Remark (computation time): For the polyhedrons that we usually consider (few hundred vertices), the construction time is around 1 s. The full STP-BV model of the robot discussed in Section VIII-B, is obtained in approximately 30 s.

VIII. APPLICATIONS

Distance computation algorithms can be categorized as feature-based and simplex-based. In [2], we implemented a featured-based distance computation for the STP-BV based on V-Clip [21]. Later (see [22]) we implemented a simplex-based method based on the GJK algorithm [23] [24] and its adaptation for the case of penetration distance [25]. Here, we use the latter solution because the penetration depth is necessary in optimization algorithms working with nonfeasible iterates (such as most of the gradient-based algorithms, like NPSOL). In a nutshell,

⁴When for three points v_1, v_2 , and v_3 both spheres built on $v_1 v_2 v_3$ and $v_1 v_3 v_2$ contain \mathcal{P} .

TABLE II
MEAN TIME PER ITERATION IN MICROSECOND (WITH STANDARD DEVIATION)
FOR THE OPTIMIZATION PROBLEM (7)–(8)

	SolvOpt	FSQP	FSQP no regul.	NPSOL
Reg.	35.9 ± 8.0	37.4 ± 1.8	33.0 ± 0.5	69.3 ± 27.6
Kink	39.6 ± 10.4	55.4 ± 4.8	60.5 ± 11.6	94.6 ± 30.4

the integration of the STP-BV in a GJK scheme boils down to writing its support function s , i.e., the function that maps a vector v to the extremal point p of the STP-BV in the direction of v . Let f be a part of a torus or sphere defining the surface of the STP-BV. All the normal vectors to f form a cone \mathcal{N}_f . We have $s(v) = s_f(v)$ with s_f the support function of the torus or sphere supporting f such as $v \in \mathcal{N}_f$ (see [22] for more details). The support function is computed in $\mathcal{O}(\sqrt{n})$, where n denotes the number of vertices in $\mathcal{H}_R(\mathcal{P})$. However, it comes close to $\mathcal{O}(1)$ when there is a temporal coherency. Our STP-BV library⁵ consists of two parts: the STP-BV construction from a point cloud, and the distance computation, which can be used like any classical distance computation algorithms. One can express collision avoidance constraints by requiring that the returned distance is positive, or enforce contacts by imposing that the distance must be equal to zero.

A. Cube Problem

We study a generalization of the problem (1)–(2) to the 3-D case: minimizing the distance of a six-DOF cube to a reference configuration while avoiding a collision constraint with a fixed bigger cube (see Fig. 13). The fixed big cube is modeled as a polyhedron and the small one as a polyhedron or its STP-BV regularization. The optimization problem solved is as follows:

$$\min_x \frac{\|C\|^2}{2} - \text{tr}(R(\psi_0, \theta_0, \phi_0)R(\psi, \theta, \phi)^T) \quad (7)$$

$$\text{s.t. } d(x) \geq d_0 \quad (8)$$

where $x = [C_x, C_y, C_z, \psi, \theta, \phi]$ with $C = [C_x, C_y, C_z]^T$ as the cube center, and ψ, θ, ϕ denote the roll, pitch, and yaw angles respectively, of the cube orientation, ψ_0, θ_0, ϕ_0 denote the reference orientation. $R(\cdot, \cdot, \cdot)$ denotes a rotation matrix, $\text{tr}(\cdot)$ indicates for the trace, and $d(\cdot)$ denotes the distance between the small and the big cubes. The minimum of $-\text{tr}(R(\psi_0, \theta_0, \phi_0)R(\psi, \theta, \phi)^T)$ is obtained when both rotation matrices are equal, which gives a value of -3 .

The computations performed are the same as with the bar problem, and we chose $(\psi_0, \theta_0, \phi_0)$ to make the optimization converge either to a regular point (see Fig. 13, first case) or to a kink point (see Fig. 13, third case). Similarly to the bar problem, despite the kink points of the problem being a subset of measure zero in the parameter space, the optimization will converge to them for regions of the $(\psi_0, \theta_0, \phi_0)$ space that have nonzero measure. The probability of converging to a kink point is thus not zero. Results are given in Fig. 14 and Table II. We did not run NPSOL without regularization because of the poor convergence obtained in the bar problem. One can observe that

⁵STP-BV library can be provided by authors upon request.

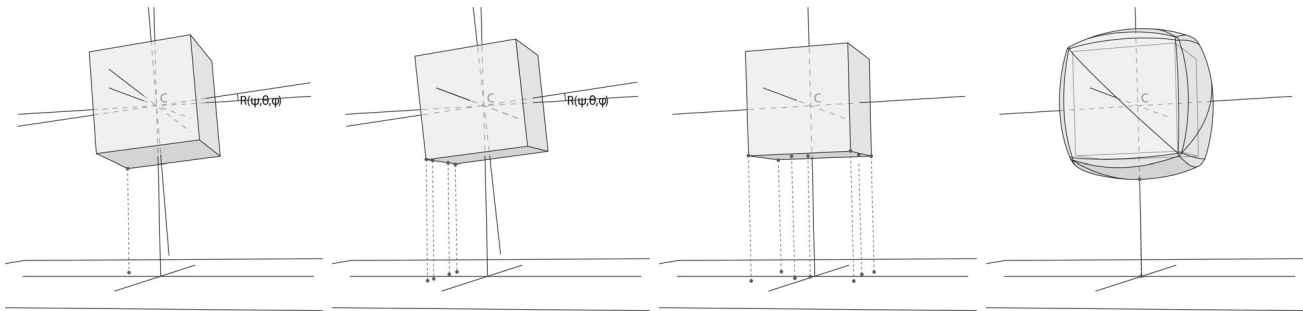


Fig. 13. Cube problem. From left to right: (a) case with only one pair of witness points, (b) one edge parallel to a face of the big cube, inducing an infinity of witness pairs, (c) parallel faces, and (d) the STP-BV of the cube in the same configuration as (c), yielding a single pair of witness points.

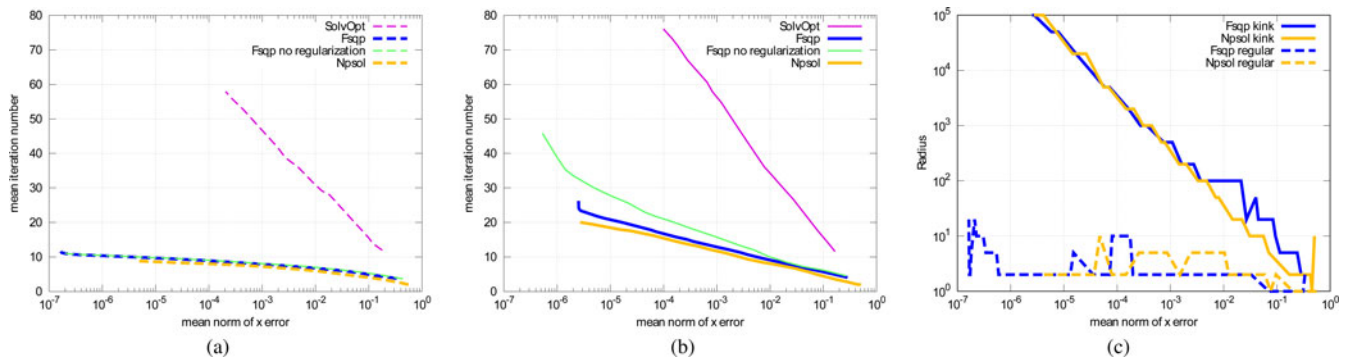


Fig. 14. Comparison of the computation time (ordinate) to achieve a desired precision of the result (abscissa) for the optimization problem (7) and (8), and the corresponding optimal regularization radius. FSQP behaves the same with or without regularization when converging to a regular point. (a) Convergence to a regular point. (b) Convergence to a kink point. (c) Best radius for a given precision.

convergence is faster to a regular point. In terms of the computation time, SQP algorithms are faster than SolvOpt, whereas in terms of the number of iterations, SQP algorithms are considerably better. In terms of the computation time, the fastest convergence to a kink point is obtained with FSQP using the STP-BV, followed by the use of NPSOL with the STP-BV for high precision and FSQP with the cube (i.e., not using the STP-BV) for low precision; in terms of the number of iterations, the fastest algorithm is NPSOL, followed by FSQP with the STP-BV, and FSQP without the STP-BV. At a regular point, in terms of the computation time, the fastest is FSQP without the STP-BV, closely followed by FSQP with the STP-BV and NPSOL; in terms of the number of iterations NPSOL is better, followed by FSQP. A notable difference with the bar case is that the use of the STP-BV significantly improves the convergence speed of FSQP that can however correctly converge to a kink point. As in the bar case, the best choice of radius is the smallest one that gives the required precision.

B. Collision-Free Humanoid Posture Generation

Our study is originally motivated by the integration of collision avoidance in a planning algorithm presented in [4], part of which consists of generating postures. We show results from our posture generator using the STP-BV for collision avoidance. In brief, this posture generator performs an optimization under constraints such as static equilibrium, required body positions (for example, robot–environment contacts, described as

matches between frames of the robot and frames of the environment), and collision avoidance. The variable is the configuration vector. The criterion to be minimized can be any user-defined smooth function. In the following scenarios, we use the distance to a human-like upright position.

The problem of collision avoidance in posture generation was addressed in [26] and [27]. Abdel-Malek *et al.* [26] generated postures from an optimization formulation. Spheres are used as coverage to define the collision avoidance constraints. Their method requires a considerable number of spheres to cover the virtual avatar and hence a considerable number of pairwise constraints. Peinado *et al.* [27] generated postures from inverse kinematics and prioritized tasks, but this method is not smooth with all observers' primitives; moreover, observers need to be specified by the user and are context specific. There are other approaches to the aforementioned problem such as in [28] where avoidance is made through potential fields.

The model of each HRP-2 body contains between 50 and 800 vertices. Parameters of the STP-BV are as follows: $r = 2$ cm, and R is fixed between 1 m and 10^5 m. The environment is modeled with polyhedrons, as only one body of a pair needs to be strictly convex for the continuity of the distance gradient. Body pairs that need to be checked for self-collision have been studied in [29] and [30]. In the latter, lookup tables are used for dealing with composed joints; hence, the safety margins of the bounding volumes do not restrain the movement possibilities. We focused on this point too, but since we need continuous gradients for all constraints, such a method was not possible.

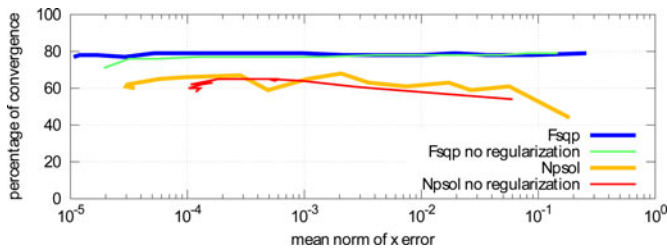


Fig. 15. *Enter-car scenario*. Percentage of convergence (ordinate) achieved for a desired precision of the result (abscissa).

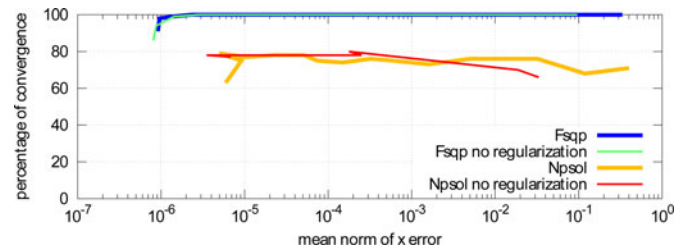


Fig. 17. *Reaching scenario*. Percentage of convergence (ordinate) achieved for a desired precision of the result (abscissa).

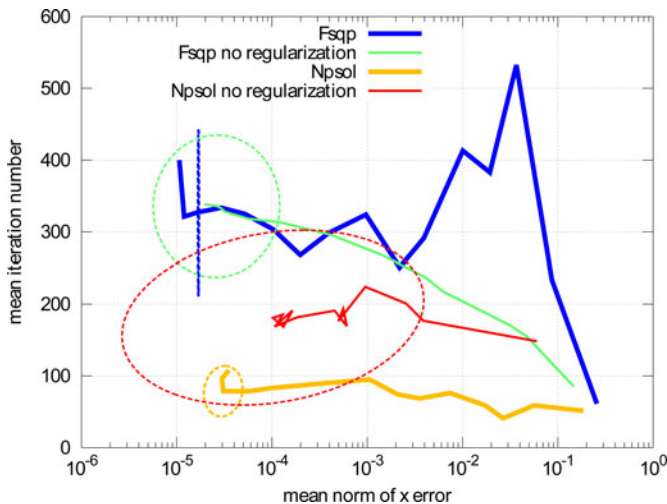


Fig. 16. *Enter-car scenario*. Comparison of the mean iteration number (ordinate) to achieve a desired precision of the result (abscissa). Confidence ellipses represent the repartition of solutions around an arbitrary mean point.

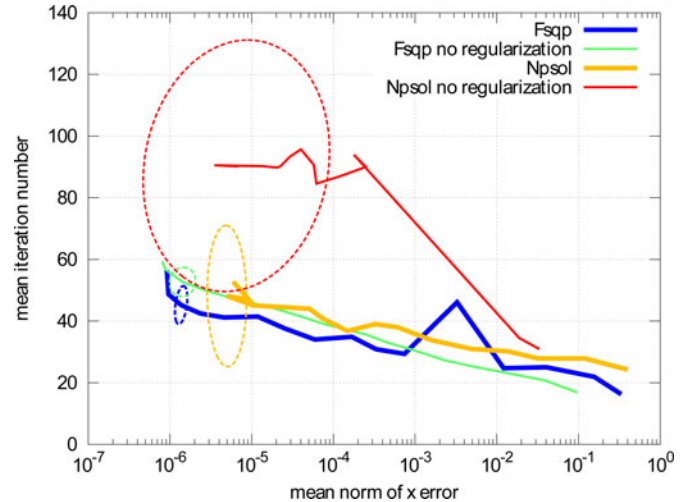


Fig. 18. *Reaching scenario*. Comparison of the mean iteration number (ordinate) to achieve a desired precision of the result (abscissa). Confidence ellipses represent the repartition of solutions around a mean point.

Thus, we used specific analytical functions of the joints values to prevent collision around the hip, waist, neck, and shoulder joints. These functions were obtained either geometrically or by experimentations on a real HRP-2 robot. There are 117 autocollision constraints. Eight of them are analytical, the others being computed with the STP-BVs. We apply this posture generator to two scenarios (see also [2]):

Enter-car scenario: The robot has to place its right foot inside a car, on the floor in front of the driver seat, and its left foot on the floor outside of the car. The robot must be stable with only the left foot contact. Nineteen pairs are checked for collisions with the environment. The scenario is illustrated in Fig. 19.

Reaching scenario: HRP-2 attempts to reach an object near its waist. Both feet are placed on the ground and the left-hand position and orientation are fixed around the object. This scenario was designed to exhibit a nonsmooth self-collision between the forearm and the flat part of the waist, as can be seen in the right picture of Fig. 20. For these two scenarios, we modified our protocol; we identify beforehand the log-linear relations between the parameters of the different algorithms and the radii. Each parameter p_i is chosen to follow a law $p_i(R) = 10^{a_i \log_{10} R + b_i}$, where the (a_i, b_i) have been identified to obtain the best iteration-precision ratio for a set of scenarios. We then run both scenarios above for R varying from 1 to 10⁵; each time the parameters are given by the laws. Figs. 15–18

TABLE III
MEAN TIME PER ITERATION IN MILLISECOND (WITH STANDARD DEVIATION)
FOR *ENTER-CAR* AND *REACHING* SCENARIOS

	FSQP	FSQP no regul.	NPSOL	NPSOL no regul.
Enter-car	508 ± 81	497 ± 103	369 ± 85	218 ± 17
Reaching	304 ± 32	231 ± 8	415 ± 120	267 ± 96

can then be seen as parametric functions of R . We proceed this way to get closer to real-condition uses; we do not take the results for the best parameters and radii tuned for each particular scenario—such a tuning cannot be considered when generating thousands of different postures, e.g., in our case of planning. Each scenario is running with the same parameters, chosen to have the best compromise between the computation time and the expected precision of the solution. The initial postures are randomly chosen around the straight posture within a range of 4° for the angles and 6.5 cm for the root coordinates.

Figs. 15–18 and Table III present the computation results for the *Enter-car* and *Reaching* scenarios. In each case, the results are given with and without the use of the STP-BV using NPSOL and FSQP (we did not run SolvOpt because of its poor results in the cube case). The displayed results are the mean values of 100 runs with random initial conditions, for each value of R . Despite the short range of initial values, the algorithms sometimes converge to different local minima that we identified

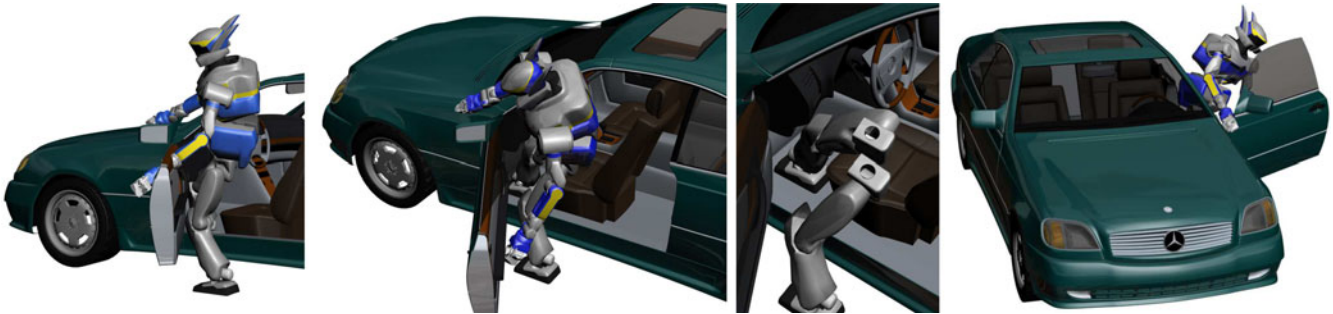


Fig. 19. Enter-car scenario. From left to right: without collision avoidance constraints, successful collision-free posture from three different view points.

manually (for example, the left arm above the car door for the *Enter-car* scenario, instead of being along it as seen in Fig. 19). For each of them, we compute the best parameter vector x_{ref} over an extensive series of runs. The convergence precision is then measured as the distance between the obtained parameter vector x of a run and its closest local minimum. At the optimum of the *Enter-car* scenario, six collision constraints with the environment are active. Only one of them is nonsmooth at the solution. At the optimum of the *Reaching* scenario, one collision constraint is active (between the waist and the left arm) which is nonsmooth at the solution. The convergence ratios are presented in Figs. 15 and 17, and for the cases which converged, Figs. 16 and 18 illustrate the convergence speeds.

For both scenarios, FSQP allows a better convergence ratio than NPSOL. Nonconvergences are due to the impossibility of finding a feasible solution. On average, for NPSOL, the use of STP-BV decreases the number of iterations by approximately 50% and the computation time by approximately 25%. On average, for FSQP, the use of the STP-BV does not significantly improve the number of iterations, and the computation time is slightly slower, but it allows reaching better mean precisions. For both FSQP and NPSOL, the confidence ellipses are smaller with the STP-BV, representing a better precision reliability of the obtained solutions. The use of the STP-BV significantly improves the convergence speed of NPSOL because a nonsmooth collision is active at the solution, which leads to convergence problems without STP-BV. However, as compared to the cube problem for which no convergence is obtained without STP-BV, here, convergence is obtained because the nonsmoothness is relatively not dominant; indeed, the size of the face causing gradient discontinuity is considerably smaller than the cube's. For FSQP, we do not have explanations why the nonsmoothness has so little impact on the convergence speed, except that it might have something to do with the feasibility of the iterates. Our interpretation of the nonconvergence due to unfeasibility with NPSOL is that the nonlinear problems that we attempt to solve are highly nonconvex, particularly because of the collision constraints. The reason why FSQP accommodates better with this nonconvexity is because it keeps feasible iterates, what may help to avoid unfeasible local minima in which NPSOL falls. Another consequence of the nonconvexity of the problem is that in approximately 10–40% of the convergence cases with NPSOL, the convergence was obtained in other local minima. Even if the problems

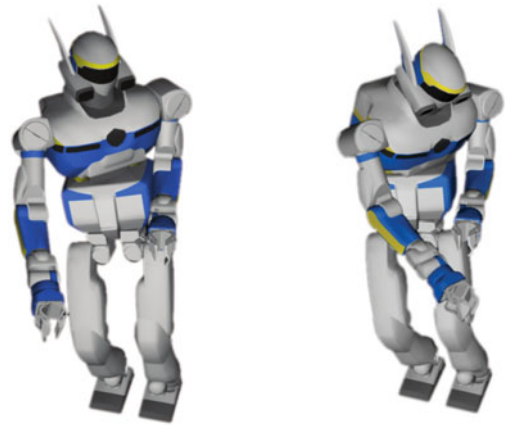


Fig. 20. Reaching scenario. (Left) Posture obtained without collision avoidance constraints. (Right) Successful collision-free posture.

considered are nonconvex, we do not address the problem of a global minimum search in this paper.

IX. CONCLUSION

We propose a new bounding volume operator to enforce the continuity of the gradient of the distance function. The main idea is to ensure the strict convexity of the bounding envelopes of the objects, which are computed offline. We proposed such a bounding volume made as a patch of parts of spheres and tori, the STP-BV, whose parameter choice is a tradeoff between the tightness of the volume and the regularization of the gradient of the distance. We showed that this regularization together with classical smooth optimization is an interesting alternative to the direct use of a nonsmooth approach, through academic cases and a collision-free (including self-collision) optimization-based humanoid posture generation for HRP-2.

REFERENCES

- [1] E. G. Gilbert and D. W. Johnson, "Distance functions and their application to robot path planning in the presence of obstacles," *IEEE J. Robot. Autom.*, vol. RA-1, no. 1, pp. 21–30, Mar. 1985.
- [2] A. Escande, S. Miossec, and A. Kheddar, "Continuous gradient proximity distances for humanoids free-collision optimized-postures generation," in *Proc. IEEE/RAS Int. Conf. Human. Robots*, Pittsburgh, PA, USA, Nov. 20–Dec. 1, 2007, pp. 188–195.
- [3] F. Bonnans, C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal, *Numerical optimization- Theoretical and Practical Aspects*. New York, NY, USA: Springer-Verlag, Sep. 2002.

- [4] A. Escande, A. Kheddar, and S. Miossec, "Planning contact points for humanoid robots," *Robot. Autom. Syst.*, vol. 61, pp. 428–442, May 2013.
- [5] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar, "Generation of whole-body optimal dynamic multi-contact motions," *Int. J. Robot. Res.*, vol. 32, pp. 1104–1119, 2013.
- [6] S. M. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Auton. Robots*, vol. 32, no. 4, pp. 433–454, May 2012.
- [7] O. Stasse, A. Escande, N. Mansard, S. Miossec, P. Evrard, and A. Kheddar, "Real-time (self)-collision avoidance task on a HRP-2 humanoid robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 3200–3205.
- [8] X. Zhang and Y. J. Kim, "*k*-IOS: Intersection of spheres for efficient proximity query," in *Proc. IEEE Int. Conf. Robot. Autom.*, Saint Paul, MN, USA, May 14–18, 2012, pp. 354–359.
- [9] N. Chakraborty, J. Peng, S. Akella, and J. Mitchell, "Proximity queries between convex objects: An interior point approach for implicit surfaces," in *Proc. 2006 IEEE Int. Conf. Robot. Autom.*, Orlando, FL, USA, May 2006, pp. 1910–1916.
- [10] S. Rusaw, "Sensor-based motion planning in SE(2) and SE(3) via nonsmooth analysis," Oxford University Computing Laboratory, Oxford, U.K. Tech. Rep. PRG-RR-01-13, Sep. 27, 2001.
- [11] R. Patel, F. Shadpey, F. Ranjbaran, and J. Angeles, "A collision-avoidance scheme for redundant manipulators: Theory and experiments," *J. Robot. Syst.*, vol. 22, no. 12, pp. 737–757, 2005.
- [12] A. Dietrich, T. Wimböck, A. Albu-Schäffer, and G. Hirzinger, "Integration of reactive, torque-based self-collision avoidance into a task hierarchy," *IEEE Trans. Robot.*, vol. 28, no. 6, pp. 1278–1293, Dec. 2012.
- [13] R. Lampariello, D. Nguyen-Tuong, C. Castellini, G. Hirzinger, and J. Peters, "Trajectory planning for optimal robot catching in real-time," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 3–9, 2011, pp. 3719–3726.
- [14] V. Shapiro and I. Tsukanov, "Implicit functions with guaranteed differential properties," in *Proc. ACM Symp. Solid Model. Appl.*, 1999, pp. 258–269.
- [15] A. Fuhrmann, G. Sobottka, and C. Gross, "Distance fields for rapid collision detection in physically based modeling," in *Proc. Int. Conf. Comput. Graph. Vis.*, 2003, pp. 58–65.
- [16] X. Zhu, H. Ding, and S.-K. Tso, "A pseudodistance function and its application," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 344–352, Apr. 2004.
- [17] C. T. Lawrence and A. L. Tits, "A computationally efficient feasible sequential quadratic programming algorithm," *SIAM J. Optimiz.*, vol. 11, pp. 1092–1118, 2001.
- [18] F. Kappel and A. V. Kuntsevich, "An implementation of Shor's r -algorithm," *Comput. Optim. Appl.*, vol. 15, pp. 193–205, 2000.
- [19] Y. Nesterov and J. Vial, "Homogeneous analytic center cutting plane methods for convex problems and variational inequalities," *SIAM J. Optim.*, vol. 9, no. 3, pp. 707–728, 1999.
- [20] G. van den Bergen, *Collision Detection in Interactive 3-Environments* (The Morgan Kaufmann Series in Interactive 3D Technology), D. H. Eberly, Ed. San Mateo, CA, USA: Morgan Kaufmann, 2004.
- [21] B. Mirtich, "V-Clip: Fast and robust polyhedral collision detection," *ACM Trans. Graph.*, vol. 17, no. 3, pp. 177–208, 1998.
- [22] M. Benallegue, A. Escande, S. Miossec, and A. Kheddar, "Fast C^1 proximity queries using support mapping of sphere-torus-patches bounding volumes," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 483–486.
- [23] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE J. Robot. Autom.*, vol. 4, no. 2, pp. 193–203, Apr. 1988.
- [24] C. J. Ong and E. G. Gilbert, "Fast versions of the gilbert-johnson-keerthi distance algorithm: Additional results and comparisons," *IEEE Trans. Robot.*, vol. 17, no. 4, pp. 531–539, Aug. 2001.
- [25] C. J. Ong and E. G. Gilbert, "Growth distances: New measures for object separation and penetration," *IEEE Trans. Robot. Autom.*, vol. 12, no. 6, pp. 888–903, Dec. 1996.
- [26] K. Abdel-Malek, J. Yang, T. Marler, S. Beck, A. Mathai, X. Zhou, A. Patrick, and J. Arora, "Towards a new generation of virtual humans," *Int. J. Human Factors Model. Simul.*, vol. 1, no. 1, pp. 2–39, 2006.
- [27] M. Peinado, R. Boulic, B. Le Calenne, and D. Méziat, "Progressive cartesian inequality constraints for the inverse kinematic control of articulated chains," presented at the *EuroGraphics*, Dublin, Ireland, 2005.
- [28] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *Int. J. Human. Robot.*, vol. 2, no. 4, pp. 505–518, 2005.
- [29] J. Kuffner, K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, "Self-collision detection and prevention for humanoid robots,"

in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington DC, USA, May 2002, pp. 2265–2270.

- [30] K. Okada and M. Inaba, "A hybrid approach to practical self collision detection system of humanoid robot," in *Proc. IEEE/RSJ Int. Conf. Robots Intell. Syst.*, 2006, pp. 3952–3957.



Adrien Escande received the M.S. degree from École des Mines de Paris, Paris, France, in 2005 and the Ph.D. degree in robotics from Université d'Évry Val-d'Essonne, Évry, France, in 2008, after spending three years with the CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/CRT, Tsukuba, Japan.

He then worked as a Research Scientist with CEALIST, Fontenay-aux-Roses, France, until the end of 2012 and is now back at JRL. His current research interests include whole-body planning and control for humanoid robots and mathematical optimization for

robotics.



Sylvain Miossec received the M.Sc. and Ph.D. degree from École Centrale de Nantes, Nantes, France, in 2001 and 2004, respectively.

He was a Postdoctoral Researcher, with a 2-year JSPS Fellowship, at the CNRS-AIST Joint Robotics Laboratory, UMI3218/ and a Research Associate with the Centre National de la Recherche Scientifique (CNRS) for another two years. His research interests include biped robots, humanoid robots, optimal motion, walking control and stability, multibody simulation, and optimal design of robots.



Mehdi Benallegue received the ingénieur degree from the Institut National d'Informatique (INI), Algeria, in 2007, the M.Sc. degree from the University of Paris 7, Paris, France, in 2008, and the Ph.D. degree from the Université Montpellier 2, Montpellier, France, in 2011. He conducted his Ph.D. research with the CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/CRT and at the INRIA Grenoble.

He is currently a Research Associate with the Laboratory for Analysis and Architecture of Systems, Toulouse, France. His research interests include computational geometry, humanoid robots, locomotion, imitation, learning, and state observers.



Abderrahmane Kheddar (M'12) received the BSCS degree from the Institut National d'Informatique (INI), Algeria, and the M.Sc. and Ph.D. degrees in robotics from the University of Pierre et Marie Curie, Paris 6, Paris, France.

He is currently Directeur de Recherche with French National Centre for Scientific Research and the Director of the CNRS-AIST Joint Robotic Laboratory, UMI3218/CRT, Tsukuba, Japan. He is also leading the Interactive Digital Human team at CNRS-UM2 LIRMM, Montpellier, France. His research interests include haptics, humanoids, and thought-based control.

Dr. Kheddar served on the editorial board of the IEEE TRANSACTIONS ON HAPTICS from 2007 to 2010. He is presently an Editor of the IEEE TRANSACTIONS ON ROBOTICS and an Associate Editor of the *Journal of Intelligent and Robotic Systems*, *Presence*, and *Frontiers in Bionics*. He is a member of the the IEEE Robotics and Automation Society.